



ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

[ΗΛΕΚΤΡΟΝΙΚΗ ΔΙΑΚΥΒΕΡΝΗΣΗ]

Διαμόρφωση Διαδικτυακού Μαθήματος στην Τεχνητή
Νοημοσύνη για επαγγελματίες Τ.Π.Ε.
Ανάπτυξη Εκπαιδευτικού Υλικού

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΟΥ

Τριανταφύλλου Βασίλη

Επιβλέπων : Αλεξόπουλος Χαράλαμπος

Μέλη εξεταστικής επιτροπής:

Λουκής Ευριπίδης

Σαράντης Δημήτρης

Σάμος, [Απρίλιος 2024]

Η σελίδα αυτή είναι σκόπιμα λευκή.

Πρόλογος

Στη διάρκεια της ιστορίας, η κοινωνίες προοδεύουν συνεχώς δημιουργώντας νέες δυνατότητες προς όφελος της ανθρωπότητας. Σε τρεις περιπτώσεις, αυτές οι πρόοδοι και οι κοινωνικές αλλαγές αποδείχθηκαν τόσο σημαντικές ώστε η ανθρωπότητα τις θεώρησε βιομηχανικές επαναστάσεις που προκάλεσαν αλλαγές στον τρόπο με τον οποίο σκεφτόμαστε, λειτουργούμε και αλληλεπιδρούμε. Στον 21ο αιώνα, ένα τεράστιο μωσαϊκό τομέων κάνουν μια νέα επαναστατική αλλαγή που ονομάζεται 4η Βιομηχανική Επανάσταση (4IR). Μερικοί από τους τομείς αυτούς είναι τα κυβερνο-φυσικά συστήματα, το διαδίκτυο των πραγμάτων, η ρομποτική, η γνωσιακή πληροφορική, η γενετική μηχανική, η νανοτεχνολογία, βιοτεχνολογία, οι κβαντικοί υπολογιστές και η τεχνητή νοημοσύνη. Την ώρα που όλα γύρω μας γίνονται «έξυπνα», μένει ένα πολύ σημαντικό συστατικό να ακολουθήσει: Τα στελέχη, το ανθρώπινο δυναμικό της αγοράς εργασίας. Η μετάβαση είναι αργή και απαιτείται συγκροτημένη παρέμβαση μέσω ενεργητικών πολιτικών, εκπαίδευση και δια βίου μάθηση. Προς την κατεύθυνση αυτή σημαντικό ρόλο διαδραματίζουν τα Μαζικά Ανοικτά Διαδικτυακά Μαθήματα (MOOCs). Κατατάσσονται στη μη τυπική εκπαίδευση, στηρίζονται στο μοντέλο της ελεύθερης και ανοικτής πρόσβασης στη γνώση και για το λόγο αυτό, προσελκύουν έναν τεράστιο αριθμό εκπαιδευομένων από όλο τον κόσμο. Υπό το πρίσμα αυτό και θέλοντας να συμβάλουμε προς την κατεύθυνση αυτή, στην παρούσα διπλωματική δημιουργήσαμε μια σειρά μαθημάτων τεχνητής νοημοσύνης για επαγγελματίες πληροφορικής.

Ευχαριστίες

Η συγγραφή της παρούσας διπλωματικής εργασίας είναι το αποτέλεσμα συστηματικής δουλειάς στο Πανεπιστήμιο Αιγαίου. Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή, κύριο Αλεξόπουλο Χαράλαμπο, επίκουρο καθηγητή του Πανεπιστημίου, για τη συνεργασία και την πολύτιμη συμβολή του στην ολοκλήρωσή της. Με το ευρύ επιστημονικό του πεδίο υπήρξε εμπνευστής και καθοδηγητής μου, συμβάλλοντας καθοριστικά στη διεύρυνση των γνώσεων μου και στην ολοκλήρωση της διπλωματικής εργασίας. Η καθοδήγησή του και η εμπιστοσύνη που μου έδειξε ήταν στοιχεία βαρύνουσας σημασίας για την ευόδωση της προσπάθειας και την περάτωση της εργασίας αυτής. Ένα μεγάλο ευχαριστώ οφείλω στους οικείους μου, οι οποίοι στάθηκαν αρωγοί σε όλη την πορεία των σπουδών μου και συμπαραστάθηκαν ηθικά σε κάθε μου βήμα με αγάπη, υπομονή και συνεχή ενθάρρυνση. Τους ευχαριστώ.

© [2024]

του

[ΤΡΙΑΝΤΑΦΥΛΛΟΥ ΒΑΣΙΛΗ]

Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων

ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

Η σελίδα αυτή είναι σκόπιμα λευκή.

Πίνακας περιεχομένων

Ακρωνύμια	8
Περίληψη	20
Εισαγωγή	24

Μέρος 1ο Θεωρητική προσέγγιση

1. Ηλεκτρονική Διακυβέρνηση και Τεχνητή Νοημοσύνη	27
1.1 Η Ηλεκτρονική Διακυβέρνηση	27
1.2 Είδη πληροφοριακών συστημάτων Ηλεκτρονικής Διακυβέρνησης	29
1.3 Τεχνολογίες που μεταμορφώνουν τις κυβερνήσεις	30
1.4 Τεχνητή Νοημοσύνη	31
2. Ψηφιακός Μετασχηματισμός	33
2.1 Δείκτης Ψηφιακής Οικονομίας και Κοινωνίας (DESI)	34
2.2 Στόχοι και επιδιώξεις της πολιτικής	35
2.3 Ψηφιακές Δεξιότητες και Τεχνητή νοημοσύνη	38
3. Διά βίου Μάθηση	40
3.1 Μαζικά Ανοικτά Διαδικτυακά Μαθήματα (MOOCs)	40
3.2 Πλατφόρμες φιλοξενίας μαθημάτων MOOCs	40
3.2.1 EdX	43
3.2.2 Coursera	43
3.2.3 Udacity	44
3.2.4 Udemy	45
3.2.5 FutureLearn / United Kingdom	45
3.2.6 France Université Numérique (FUN) / France	45
3.2.7 European Multiple MOOC Aggregator	45
3.2.8 K-MOOC/Korea	46
3.2.9 XuetangX/China	46
3.2.10 ZhihuiShu/ China	46
3.3 MOOCs στην Ελλάδα	47
3.3.1 mooc.edu.gr	48
3.3.2 Opencourses	48
3.3.3 Coursity	48
3.3.4 Mathesis	49
3.3.5 Ψηφιακή Ακαδημία Πολιτών	50
3.4 Η τεχνητή νοημοσύνη στις πλατφόρμες MOOC	51
3.5 Τεχνητή νοημοσύνη στις Ελληνικές πλατφόρμες	53
3.6 Το MOOC που δημιουργήσαμε	54
3.7 Γιατί επαγγελματίες ΤΠΕ	54
3.8 Οι ενότητες που δημιουργήσαμε (Syllabus)	57

Μέρος 2ο Αναλυτική περιγραφή ενοτήτων

Ενότητα 1: Θεμέλια της τεχνητής νοημοσύνης	60
Μάθημα 1.1: Πεδίο εφαρμογής της τεχνητής νοημοσύνης	60
Θέματα προς συζήτηση.....	60
Σημειώσεις διάλεξης	60
Τι είναι η Τεχνητή Νοημοσύνη;	60
Τομείς της Τεχνητής Νοημοσύνης	61
Επίλυση προβλημάτων	61
Αναπαράσταση γνώσης και συλλογιστική	62
Μηχανική μάθηση	63
Επεξεργασία φυσικής γλώσσας	63
Αντίληψη	64
Ρομποτική.....	64
Περιπτώσεις χρήσης	65
ΡΟΜΠΟΤΙΚΕΣ ΗΛΕΚΤΡΙΚΕΣ ΣΚΟΥΠΕΣ 48.....	65
Αναφορές	67
Μάθημα 1.2: Επίλυση προβλημάτων με αλγόριθμους αναζήτησης	67
Θέματα προς συζήτηση	67
Σημειώσεις διάλεξης	67
Τι είναι πρόβλημα;	67
Αναζήτηση χώρου καταστάσεων	69
Αναζήτηση χώρου λύσεων	70
Περιπτώσεις χρήσης 57	74
ΣΥΣΤΗΜΑΤΑ ΠΛΟΗΓΗΣΗΣ ΑΥΤΟΚΙΝΗΤΩΝ 57	74
Αναφορές 58	75
Μάθημα 1.3: Αναπαράσταση γνώσης.....	76
Θέματα προς συζήτηση	76
Σημειώσεις διάλεξης	76
Εισαγωγή	76
Προτασιακή και κατηγορηματική λογική	77
Συστήματα κανόνων παραγωγής.....	78
Δομημένες αναπαραστάσεις	79
Περιπτώσεις χρήσης 57	82
ΟΝΤΟΛΟΓΙΑ BBC	82
Αναφορές	83
Μάθημα 1.4: Μηχανική μάθηση	85
Θέματα προς συζήτηση	85
Σημειώσεις διάλεξης	85

Μεθοδοι εποπτευόμενης μάθησης	86
Δέντρα Αποφάσεων	87
Naive Bayes	89
Περιπτώσεις χρήσης	91
NAIVE BAYES	91
Αναφορές	92
Μάθημα 1.5: Εφαρμογές της Τεχνητής Νοημοσύνης	93
Θέματα προς συζήτηση	93
Σημειώσεις διάλεξης	93
Προσωπικοί βοηθοί.....	93
Συστήματα συστάσεων	94
Αυτόνομη οδήγηση	94
Ευφυές λιανικό εμπόριο	95
Παιχνίδια υπολογιστών.....	95
Ρομποτικές συσκευές και έξυπνες συσκευές	95
Industry 4.0 (Βιομηχανία 4.0)	96
Επιτήρηση και ασφάλεια	97
Συστήματα βασισμένα στη γνώση	97
Γλώσσες για τεχνητή νοημοσύνη	97
Περιπτώσεις χρήσης	98
AMAZON GO APP	98
Αναφορές	99
Μάθημα 1.6: Ηθικές επιπτώσεις της τεχνητής νοημοσύνης.....	100
Θέματα προς συζήτηση	100
Σημειώσεις διάλεξης	100
Κίνητρα	100
Αλγοριθμική δικαιοσύνη και αλγοριθμική μεροληψία	101
Ιδιωτικότητα και καπιταλισμός της επιτήρησης.....	102
Αλγοριθμική λογοδοσία	103
Σύνθεση	103
Περιπτώσεις χρήσης	104
"ΣΕΞΙΣΤΙΚΟ" ΕΡΓΑΛΕΙΟ ΠΡΟΣΛΗΨΗΣ	104
Αναφορές	104
 Ενότητα 2: Εισαγωγή στη Μηχανική Μάθηση	 106
Μάθημα 2.1: Εισαγωγή στη Μηχανική Μάθηση	106
Θέματα προς συζήτηση.....	106
Σημειώσεις διάλεξης	106
Εισαγωγή	106

Θεμέλια της μηχανικής μάθησης με επίβλεψη	107
Αξιολόγηση της επιβλεπόμενης μηχανικής μάθησης.....	112
Περιπτώσεις χρήσης	112
ΟΠΤΙΚΗ ΑΝΑΓΝΩΡΙΣΗ ΧΑΡΑΚΤΗΡΩΝ	112
Αναφορές	113
Μάθημα 2.2: Γλώσσες και πόροι	114
Θέματα προς συζήτηση.....	114
Σημειώσεις διάλεξης	114
Εισαγωγή	114
Βιβλιοθήκες για χειρισμό δεδομένων.....	115
Numpy	115
Pandas	116
Βιβλιοθήκες για τη διαδικασία μηχανικής μάθησης	117
Άλλες βιβλιοθήκες και εργαλεία.....	119
Περιπτώσεις χρήσης	119
ΟΠΤΙΚΗ ΑΝΑΓΝΩΡΙΣΗ ΧΑΡΑΚΤΗΡΩΝ	119
Αναφορές	127
Μάθημα 2.3: Μετασχηματισμός και οπτικοποίηση δεδομένων.....	128
Θέματα προς συζήτηση.....	128
Σημειώσεις διάλεξης	128
Εισαγωγή	128
Πολυδιάστατη κλιμάκωση	130
Τοπική γραμμική ενσωμάτωση	131
t-στοχαστική ενσωμάτωση γείτονα	132
Επιλέγοντας έναν αλγόριθμο	133
Περιπτώσεις χρήσης	133
ΤΙΜΕΣ ΣΠΙΤΙΩΝ ΣΤΗΝ ΚΑΛΙΦΟΡΝΙΑ	133
Αναφορές	137
Μάθημα 2.4: Γραμμικές μέθοδοι για επιβλεπόμενη μάθηση	139
Θέματα προς συζήτηση.....	139
Σημειώσεις διάλεξης	139
Εισαγωγή	139
Γραμμικά μοντέλα για παλινδρόμηση	139
Δέντρα αποφάσεων και παλινδρόμησης.....	142
k – κοντινότεροι γείτονες	143
Naive Bayes - Ναίβ Μπέης	144
Επιλέγοντας έναν αλγόριθμο	145
Περιπτώσεις χρήσης	145
ΤΙΜΕΣ ΣΠΙΤΙΩΝ ΣΤΗΝ ΚΑΛΙΦΟΡΝΙΑ	145
ΟΙΝΟΠΑΡΑΓΩΓΟΙ	150
Αναφορές	150

Μάθημα 2.5: Μη γραμμικές μέθοδοι για επιβλεπόμενη μάθηση	156
Θέματα προς συζήτηση.....	156
Σημειώσεις διάλεξης	156
Εισαγωγή	156
Μηχανές διανυσμάτων υποστήριξης.....	156
Πολυστρωματικό perceptron	159
Μοντέλο συνόλου	159
Επιλέγοντας έναν αλγόριθμο	160
Περιπτώσεις χρήσης	162
ΤΙΜΕΣ ΣΠΙΤΙΩΝ ΣΤΗΝ ΚΑΛΙΦΟΡΝΙΑ	162
ΟΙΝΟΠΑΡΑΓΩΓΟΙ	165
Αναφορές	169
Μάθημα 2.6: Μάθηση χωρίς επίβλεψη	170
Θέματα προς συζήτηση.....	170
Σημειώσεις διάλεξης	170
Εισαγωγή	170
Ιεραρχική ομαδοποίηση.....	171
K-Means.....	172
Μοντέλα μίξης Gauss	173
Αξιολόγηση της ποιότητας της ομαδοποίησης.....	174
Περιπτώσεις χρήσης	175
ΟΙΝΟΠΑΡΑΓΩΓΟΙ	175
Αναφορές	179
Ενότητα 3: Νευρωνικά δίκτυα και βαθιά μάθηση για την όραση	180
Μάθημα 3.1: Εγκέφαλος & Νευρωνικά δίκτυα	180
Θέματα προς συζήτηση.....	180
Σημειώσεις διάλεξης	180
Εισαγωγή	180
Σειριακά υπολογιστικά και βιολογικά νευρικά συστήματα	180
Στοιχεία τεχνητών νευρωνικών δικτύων και η λειτουργία τους	181
Επισκόπηση των κοινών αρχιτεκτονικών	183
Περιπτώσεις χρήσης	185
ΠΡΟΕΤΟΙΜΑΣΙΑ ΥΠΟΛΟΓΙΣΤΗ ΓΙΑ ΤΗΝ ΚΩΔΙΚΟΠΟΙΗΣΗ ΑΛΓΟΡΙΘΜΟΥ	
ΒΑΣΙΣΜΕΝΟΥ ΣΕ ANNs ΧΡΗΣΙΜΟΠΟΙΩΝΤΑΣ ΤΗΝ PYTHON	185
Αναφορές	189
Μάθημα 3.2: Απλά Perceptrons και μάθηση με επίβλεψη.....	189
Θέματα προς συζήτηση.....	189
Σημειώσεις διάλεξης	189
Απλά perceptrons	189
Μάθηση με επίβλεψη	190

Εκπαιδευτική φάση 1	190
Εκπαιδευτική φάση 2	191
Λειτουργία απλών perceptrons	193
Απλό perceptron έναντι perceptrons ενός στρώματος	194
Περιπτώσεις χρήσης	195
ΕΚΠΑΙΔΕΥΣΗ ΑΠΛΟΥ PERCEPTRON	195
Μάθημα 3.3: Πολυστρωματικά Perceptrons και Keras	198
Θέματα προς συζήτηση.....	198
Σημειώσεις διάλεξης	198
Βαθιά νευρωνικά δίκτυα (DNN)	198
Βαθιά μάθηση	198
Αλγόριθμος οπισθοδιάδοσης.....	198
Συναρτήσεις	191
Αλγόριθμοι Επικλινής καθόδου	200
Αλγόριθμος προσαρμοστικής κλίσης	202
Μέση τετραγωνική διάδοση (Root mean square propagation RMSprop)	202
Πολυστρωματικά perceptrons (MLPs)	202
MLP για την ταξινόμηση πολλαπλών κατηγοριών Iris	202
Περιπτώσεις χρήσης	205
ΕΚΠΑΙΔΕΥΣΗ ΑΠΛΟΥ PERCEPTRON	205
Μάθημα 3.4: Βαθιά μάθηση για ταξινόμηση εικόνων.....	207
Θέματα προς συζήτηση.....	207
Σημειώσεις διάλεξης	207
Όραση υπολογιστών (CV).....	207
Συνελκτικά νευρωνικά δίκτυα (CNN)	208
CNN για την αναγνώριση χειρόγραφων ψηφίων	209
LeNet5.....	209
Σύνολο δεδομένων MNIST	209
Περιπτώσεις χρήσης	211
ΕΚΠΑΙΔΕΥΣΗ LENET5	211
Μάθημα 3.5: Διαφορετικά CNN για την ταξινόμηση εικόνων	215
Θέματα προς συζήτηση.....	215
Σημειώσεις διάλεξης	215
Ζητήματα και όρια του CNN.....	215
Υπολειμματικά δίκτυα (ResNet).....	215
CNNs με βάση την περιοχή (R-CNNs).....	215
Προ-εκπαιδευμένο μοντέλο Keras	218
Βάση ImageNet.....	218
ResNet50	219
Περιπτώσεις χρήσης	220
VGG16 ΠΡΟ-ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΡΥΘΜΙΣΗ	220

Μάθημα 3.6: Εντοπισμός αντικειμένων Μοντέλο YOLO_v3	222
Θέματα προς συζήτηση.....	222
Σημειώσεις διάλεξης	222
Εντοπισμός αντικειμένων.....	222
YOLOv1	223
Πλεονεκτήματα - Μειονεκτήματα YOLOv1	225
YOLOv2	226
YOLOv3	226
Περιπτώσεις χρήσης	226
ΔΙΑΧΕΙΡΙΣΗ ΕΡΓΟΥ DNNs ΓΙΑ ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ	226
 Ενότητα 4: Βαθιά μάθηση για επεξεργασία φυσικής γλώσσας και ανάλυση	
μεγάλων δεδομένων	229
 Μάθημα 4.1: Ενσωμάτωση λέξεων και ταξινόμηση κειμένου	229
Θέματα προς συζήτηση.....	229
Σημειώσεις διάλεξης	229
Εισαγωγή στο NLP	229
Ταξινόμηση κειμένου	230
Συλλογή, Προεπεξεργασία, Ενσωμάτωση λέξεων	236
Κατασκευή μοντέλου, Εκπαίδευση,	241
Δοκιμή, αξιολόγηση, προσαρμογή.....	243
Αποθήκευση, χρήση	248
Προσωπική Εργασία	255
Περιπτώσεις χρήσης	257
ΤΟ ΤΜΗΜΑ ΑΛΛΗΛΟΓΡΑΦΙΑΣ ΤΟΥ GMAIL	257
Αναφορές.....	260
Μάθημα 4.2: Νευρωνικά δίκτυα για NLP και βιβλιοθήκες	260
Θέματα προς συζήτηση.....	260
Σημειώσεις διάλεξης	260
Βιβλιοθήκες Scikit-learn, Keras και Tensorflow	260
Συνελικτικό νευρωνικό δίκτυο	265
Επαναλαμβανόμενο νευρωνικό δίκτυο	269
Παραλλαγές RNN (LSTM, GRU)	269
Μηχανισμός προσοχής.....	278
Ενισχυτική μάθηση.....	280
Μάθηση χωρίς επίβλεψη	289
Περιπτώσεις χρήσης	291
ΠΑΙΧΝΙΔΙΑ ΚΕΙΜΕΝΟΥ	291
ΕΝΕΡΓΗ ΑΝΑΔΙΑΤΥΠΩΣΗ ΕΡΩΤΗΣΕΩΝ	292
Μάθημα 4.3: Νέες προσεγγίσεις, εφαρμογές, ανοικτά προβλήματα.....	294

Θέματα προς συζήτηση.....	294
Σημειώσεις διάλεξης	294
NLM με βάση το χαρακτήρα	295
Βασισμένο σε Νευρωνικό Μοντέλο Γλώσσας	270
Μετασχηματιστές, BERT και ELMo.....	305
Εφαρμογές επεξεργασίας φυσικής γλώσσας	303
Ανοικτά προβλήματα.....	311
Περιπτώσεις χρήσης	312
ΕΞΥΠΗΡΕΤΗΣΗ ΠΕΛΑΤΩΝ	312
ΠΑΡΑΚΟΛΟΥΘΗΣΗ ΤΗΣ ΦΗΜΗΣ.....	312
ΠΛΗΡΟΦΟΡΗΣΗ ΑΓΟΡΑΣ.....	313
ΚΑΝΟΝΙΣΤΙΚΗ ΣΥΜΜΟΡΦΩΣΗ	313
ΥΓΕΙΟΝΟΜΙΚΗ ΠΕΡΙΘΑΛΨΗ	314
Μάθημα 4.4: Προβλήματα μεγάλων δεδομένων, τεχνικές, Hadoop	315
Θέματα προς συζήτηση.....	315
Σημειώσεις διάλεξης	315
Εισαγωγή στα μεγάλα δεδομένα και την ανάλυση μεγάλων δεδομένων	315
Ανάλυση μεγάλων δεδομένων.....	317
Αλυσίδα αξίας μεγάλων (μαζικών) δεδομένων	318
Επισκόπηση Map reduce.....	319
Πλαίσιο Hadoop	319
Από το Hadoop στο Spark.....	322
Περιπτώσεις χρήσης	322
ΑΡΧΙΤΕΚΤΟΝΙΚΕΣ ΔΕΔΟΜΕΝΩΝ BACKEND ΚΑΙ Ο ΡΟΛΟΣ ΤΟΥ HADOOP	322
Αναφορές	325
Μάθημα 4.5: Μεγάλα δεδομένα Hadoop και Spark	325
Θέματα προς συζήτηση.....	325
Σημειώσεις διάλεξης	325
Τα κύρια συστατικά του Spark	325
GraphX.....	332
Περιπτώσεις χρήσης	332
ΕΤΑΙΡΕΙΑ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ	332
Αναφορές	336
Μάθημα 4.6: Μεγάλα δεδομένα Ανάλυση, Οπτικοποίηση, Εφαρμογές	338
Θέματα προς συζήτηση.....	338
Σημειώσεις διάλεξης	338
Ανάλυση μεγάλων δεδομένων στο Cloud.....	339
Databricks	342
Spark ML.....	344
Περιπτώσεις χρήσης	345
TENSORFLOW	345

Αναφορές	346
-----------------------	------------

Βιβλιογραφία

1. Bughin, J., Seong, J., Manyika, J., Chui, M. and Joshi, R. (2018), Notes from the AI frontier: Modeling the impact of AI on the world economy. McKinsey Global Institute. New York City, United States
2. Putra, A.B.N.R, Mukhadis, A., Soraya, D.U., &, and Dzakiya, N. 2019. The Innovation of Hybrid Learning through Live 24-Hour Streaming Personal in the Learning Process in Higher Education of Vocational in the Era of Industrial Revolution 4.0. In 2nd International Conference on Vocational Education and Training
3. Dawes, S. (2002). The future of e-government. Retrieved from http://www.ctg.albany.edu/publications/reports/future_of_egov/future_of_egov.pdf.
4. McIver, W.J., & Elmagarmid, A. (2002). Advances In Digital Government: Technology, Human Factors, and Policy. Kluwer Academic Publishers.
5. McIver, W.J., & Elmagarmid, A. (2002). Advances In Digital Government: Technology, Human Factors, and Policy. Kluwer Academic Publishers.
6. Field, T. (Ed.). (2003) OECD e-Government Studies the e-Government Imperative. OECD Publishing.
7. Grönlund, A. & Horan, T.A. (2005). Introducing e-gov: history, definitions, and issues. Communications of the Association for Information Systems, 15 (1), 39.
8. Guida, J. & Crow, M. (2009). E-government and e-governance. ICT4D. Information and Communication Technology for Development, 283-320.
9. Carter, L. & Bélanger, F. (2005). The utilization of e-government services: citizen trust, innovation and acceptance factors. Information Systems Journal, 15(1), 5-25.
10. Lfstedt, U. (2007). E-Government services in local governments a study of development in Swedish municipalities. Journal of Organisational Transformation and Social Change, 4 (2), 157-176.
11. McIver, W.J., & Elmagarmid, A. (2002). Advances In Digital Government: Technology, Human Factors, and Policy. Kluwer Academic Publishers
12. Bannister, F. & Connolly, R. (2012a). Defining e-governance. E-Service Journal: A Journal of Electronic Services in the Public and Private Sectors, 8 (2), 3-25.

13. Bannister, F., & Connolly, R. (2012b). Forward to the past: Lessons for the future of e-government from the story so far. *Information Polity*, 17, 211-226.
14. Kim, J.H. (2016). A study on the effect of financial inclusion on the relationship between income inequality and economic growth. *Emerging Markets Finance and Trade*, 52 (2), 498-512.
15. McClure, D. L. (2000). Statement of David L. McClure, U.S. General Accounting Office, before the Subcommittee on Government Management, Information and Technology, Committee on Government Reform, House of Representatives. Retrieved from <http://www.gao.gov.com>
16. Heeks, R. (2004). *Understanding E-Governance for Development*, i-Government Working Paper Series, Vol 11, Manchester: Institute For Development Policy And Management.
17. Layne, K. & Lee, J. (2001). Developing a Fully Functional EGovernment: A Four Stage Model. *Government Information Quarterly*, 18(2)
18. Sharma, S. & Gupta, J. (2002). Transforming to EGovernment: A Framework", European Conference on E-Government (ECEG)
19. Eggers, W. D., & Bellman, J. (2015). Digital government transformation. The journey to government's digital future. Deloitte University Press. Retrieved from <https://www2.deloitte.com/za/en/insights/topics/digital-transformation/digital-transformation-in-government.html>.
20. Bannister, F., & Connolly, R. (2012b). Forward to the past: Lessons for the future of e-government from the story so far. *Information Polity*, 17, 211-226.
21. Linders, D., Wilson, S. C., & Bertot, J. C. (2012). Open government as a vehicle for government transformation. In *Public Sector Transformation through E-Government* (pp. 20-35). Routledge
22. Boughzala, I.; Janssen, M. & Assar, S. (2015). E-government 2.0: Back to reality, a 2.0 application to vet. In *Case Studies in e-Government 2.0* (pp. 1-14). Springer, Cham.
23. Dash, S. & Pani, S. K. (2016). E-Governance paradigm using cloud infrastructure: benefits and challenges. *Procedia Computer Science*, 85, 843-855
24. Holgate, R. (2018). Hype Cycle for Digital Government Technology 2018. Gartner. Retrieved from <https://www.gartner.com/doc/3884179/hype-cycle-digital-government-technology>
25. Annoni, A., Benczur, P., Bertoldi, P., Delipetrev, P., De Prato, G., Feijoo, C., Fernandez Macias, E., Gomez, E., Iglesias, M., Junklewitz, H., López Cobo, M., Martens, B., Nascimento, S., Nativi, S., Polvora, A., Sanchez, I., Tolan, S., Tuomi, I., & Alujevic, L. V. (2018). Artificial Intelligence: A European Perspective. Joint Research Centre, European Commission. <https://doi.org/10.2760/11251>
26. Janssen, M.; Charalabidis, Y. & Krcmar, H. (2017). Open data, information processing and datification of government. In *Proceedings of the 50th Hawaii International Conference on System Sciences*; Marjanovic, O., & Cezec-Kecmanovic, D. (2017). Exploring the tension between transparency and datification effects of open government IS through the lens of Complex Adaptive Systems. *The Journal of Strategic Information Systems*

27. Ανακοίνωση της Επιτροπής προς το Ευρωπαϊκό Κοινοβούλιο, το Ευρωπαϊκό Συμβούλιο, το Συμβούλιο, την Ευρωπαϊκή Οικονομική και Κοινωνική Επιτροπή και την Επιτροπή των Περιφερειών Τεχνητή νοημοσύνη για την Ευρώπη, Βρυξέλλες
28. Wirtz, B. W.; Weyerer, J. C., & Geyer, C. (2018). Artificial Intelligence and the Public Sector—Applications and Challenges. *International Journal of Public Administration*, 1–20. <https://doi.org/10.1080/01900692.2018.1498103>.
29. Chou, J. S.; Lin, C-W.; Pham, A.-D. & Shao, J.-Y. (2015). Optimized artificial intelligence models for predicting project award price. *Automation in Construction*, 54, 106–115. <https://doi.org/10.1016/j.autcon.2015.02.006>.
30. Pan, Y.; Tian, Y.; Liu, X.; Gu, D. & Hua, G. (2016). Urban Big Data and the Development of City Intelligence. *Engineering*, 2(2), 171–178. <https://doi.org/10.1016/J.ENG.2016.02.003>
31. Sun, T. Q., & Medaglia, R. (2019). Mapping the Challenges of Artificial Intelligence in the Public Sector: Evidence from Public Healthcare. *Government Information Quarterly*, 36(2), 368-383. <https://doi.org/10.1016/j.giq.2018.09.008>
32. Tinholt, D.; Carrara, W. & van der Linden, N. (2017). Unleashing the potential of Artificial Intelligence in the Public Sector. Capgemini Consulting. Retrieved from <https://www.capgemini.com/consulting/wp-content/uploads/sites/30/2017/10/ai-in-public-sector.pdf>.
33. González-Pérez, L. I., & Ramírez-Montoya, M. S. (2022). Components of Education 4.0 in 21st century skills frameworks: systematic review. *Sustainability*, 14(3), 1493. doi: <https://doi.org/10.3390/su14031493>
34. Αραβαντινού-Φατώρου, Κ., & Καλογρίδη, Σ. (2021). Η Εκπαίδευση 4.0 στα επιμορφωτικά προγράμματα των εκπαιδευτικών Δευτεροβάθμιας Εκπαίδευσης. Στο Α. Λιοναράκης (Επιμ.) Πρακτικά 11ου Συνεδρίου για την Ανοικτή & εξ Αποστάσεως Εκπαίδευση, Τόμος 11, Μέρος 1Α' (σελ. 54-68). e-Publisher: EKT. <http://dx.doi.org/10.12681/icodl.3549>
35. Πριμικήρη, Α. (2020). Εκπαίδευση 4.0 Μάθηση και Δεξιότητες του 21ου αι. Αθήνα: Εκδόσεις Παπαδόπουλος.
36. Vuorikari, R., Kluzer, S. & Punie, Y. (2022). DigComp 2.2: The Digital Competence Framework for Citizens-With new examples of knowledge, skills and attitudes (Vol. UR 31006 EN). Publications Office of the European Union
37. Righi, R., López-Cobo, M., & Alaveras, G. (2020). Academic offer of advanced digital skills in 2019-20: international comparison: focus on artificial intelligence, high performance computing, cybersecurity, and data science. Retrieved April 18, 2022, from <https://data.europa.eu/doi/10.2760/225355>
38. Καραλής, Θ. (2013). Κίνητρα και Εμπόδια για τη Συμμετοχή των Ενηλίκων στη Διά Βίου Εκπαίδευση. Αθήνα: ΙΝΕ ΓΣΕΕ & ΙΜΕ ΓΣΕΒΕΕ.
39. Hoy, M. B. (2014). MOOCs 101: an introduction to massive open online courses. *Medical reference services quarterly*, 33(1), 85-91.
40. Karnouskos, S., & Holmlund, M. (2014). Impact of Massive Open Online Courses (MOOCs) on Employee Competencies and Innovation. Blekinge Institute of Technology

41. P. Gómez, E. Trigo, R. Moreno, A. Partial Evaluation of European Deposit Insurance Scheme funding based on risk análisis Int. Rev. Econ. Financ., 78 (2022), pp. 234-247, 10.1016/j.iref.2021.11.013
42. Mortensen, E.Ø.; De Notaris, C.; Peixoto, L.; Olesen, J.E.; Rasmussen, J. Short-term cover crop carbon inputs to soil as affected by long-term cropping system management and soil fertility. Agric. Ecosyst. Environ. 2021, 311, 107339.
43. Li, X.; Tan, A.; Chen, K.; Pan, Y.; Gentry, T.; Dou, F. Effect of cover crop type and application rate on soil nitrogen mineralization and availability in organic rice production. Sustainability 2021, 13, 2866.
44. Ruipérez-Valiente, J. A. (2021). Supplementary materials for paper: Global and regional MOOC providers: Differences in learners' demographics, preferences, and perceptions. <https://doi.org/10.17605/OSF.IO/MBWDX>
45. Βλαχαβάς, Ι., Κεφαλάς, Π., Βασιλειάδης, Ν., Κόκκορας, Φ., Σακελλαρίου, Η. (2011), Τεχνητή Νοημοσύνη, Γ' Έκδοση, Πανεπιστήμιο Μακεδονίας
46. EU (2020b). Σχέδιο δράσης για την ψηφιακή εκπαίδευση 2021-2027. Ανακτήθηκε Απρίλιο 18, 2022, από <https://eur-lex.europa.eu/legal-content/EL/TXT/PDF/?uri=CELEX:52020DC0624&from=EN>
47. EU (2021). Greece in the Digital Economy and Society Index. Retrieved April 18, 2022, from <https://digitalstrategy.ec.europa.eu/en/policies/desi-greece>
48. Αποστολάκης Ι., Λουκής Ε., Χάλαρης Ι., (2008), «Ηλεκτρονική Δημόσια Διοίκηση, Οργάνωση, Τεχνολογία και Εφαρμογές», Εκδόσεις: Παπαζήση

Ακρωνύμια

PCA	Principal Component Analysis	Ανάλυση κύριων συνιστωσών
MDS	Multidimensional Scaling	Πολυδιάστατη κλιμάκωση
LLE	Locally Linear Embedding	Τοπικά γραμμική ενσωμάτωση
t-SNE	t- Stocasting Neighbor Embedding	t- στοχαστική ενσωμάτωση γειτόνων
YoLo	You Only Look Once	Κοιτάξεις μόνο μια φορά
IoU	Intersection over Union	Λόγος της τιμής προς την ένωση
CNN	Convolutional Neural Networks	Συνελκτικά νευρωνικά δίκτυα
R-CNN	Region-Based CNN	Βασισμένα στην περιοχή CNN
CV	Computer Vision	Όραση Υπολογιστών
MLP	Multilayer Perceptron	Πολυστρωματικό perceptron
ResNet	Residual Network	Διαφορικό δίκτυο
FPN	Feature Pyramid Network	Δίκτυο χαρακτηριστικών πυραμίδας
DNN	Deep Neural Networks	Βαθιά νευρωνικά δίκτυα
MSE	Mean Squared Error	Μέσο τετραγωνικό σφάλμα
MSLE	Mean Squared Logarithmic Error	Μέσο τετραγωνικό λογαριθμικό σφάλμα
MAE	Mean Absolute Error	Μέσο απόλυτο σφάλμα
ANN	Artificial Neural Networks	Τεχνητά νευρωνικά δίκτυα
NLP	Natural Language Processing	Επεξεργασία φυσικής γλώσσας
CBOW	Continuous bag of words	
GloVe	Global Vectors for Word Representation	Παγκόσμια διανύσματα για αναπαράσταση λέξεων
CPU	Central Processing Unit	Κεντρική Μονάδα Επεξεργασίας
GPU	Graphic Processing Unit	Μονάδα επεξεργασίας γραφικών
TPU	Tensor Processing Unit	Μονάδες επεξεργασίας τανυστών
NLTK	Natural Language Toolkit	Εργαλειοθήκη φυσικής γλώσσας
POS	Part of Speech	Μέρος του λόγου
ReLU	Rectified Linear Unit	
RNN	Recurrent neural networks	Επαναλαμβανόμενα νευρωνικά δίκτυα
BPTT	Back propagation Through Time	Αντίστροφη μετάδοση λάθους μέσα στο χρόνο
LSTM	Long Short-Term Memory	Μακροχρόνια βραχυπρόθεσμη μνήμη
BRNNs	Bidirectional RNNs	Αμφίδρομο RNN
GRU	Gated Recurrent Unit	Μονάδα αναδρομικής πύλης
RL	Reinforcement Learning	Ενισχυτική μάθηση

DQN	Deep Q Networks (Q-value συνάρτηση)	
BiDAF	Bi-Directional Attention Flow	
C2W	Character to word	Χαρακτήρας σε λέξη
ELMo	Embedding from Language Model	Ενσωμάτωση από το γλωσσικό μοντέλο
biLM	Bidirectional language model	Αμφίδρομο γλωσσικό μοντέλο
ESIM	Enhanced Sequential Inference model	Ενισχυμένο μοντέλο διαδοχικών συμπερασμάτων
SRL	Semantic role labeling	Σημασιολογική σήμανση ρόλων
CRF	Conditional random field	Τυχαίο πεδίο υπό όρους
BCN	Bi-attentive classification network	Δίκτυο ταξινόμησης διπλής προσοχής
BERT	Bidirectional Encoder Representations from Transformers	Αναπαραστάσεις κωδικοποιητή διπλής κατεύθυνσης από μετασχηματιστές
NER	Named entity recognition	Αναγνώριση ονομαστικών οντοτήτων
BD	Big data	Μεγάλα Δεδομένα
BI	Business Intelligence	Επιχειρηματική ευφυΐα
RDBMS	Relational Data Base Management System	Σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων
HDFS	Hadoop Distributed File System	Κατανεμημένο σύστημα αρχείων Hadoop
YARN	Yet Another Resource Negotiator	Άλλος ένας διαπραγματευτής πόρων
RDD	Resilient Distributed Datasets	Ευέλικτα κατανεμημένα σύνολα δεδομένων
IAAS	Infrastructure as a service	Υποδομές ως υπηρεσία
SAAS	Software as a Service	Λογισμικό ως Υπηρεσία
PAAS	Platform as a Service	Πλατφόρμα ως υπηρεσία
GCP	Google Cloud Platform	Πλατφόρμα Cloud της Google
AWS	Amazon Web Services	Υπηρεσίες Web της Amazon

Περίληψη

Σήμερα, η ΤΝ συνδυάζει μια τεράστια ποικιλία επιμέρους πεδίων, τα οποία καλύπτουν ένα φάσμα που ξεκινά από γενικούς τομείς, όπως η μάθηση και η αντίληψη, και φτάνει σε συγκεκριμένες εργασίες όπως το σκάκι, η απόδειξη μαθηματικών θεωρημάτων, η συγγραφή ποίησης, και η διάγνωση ασθενειών. Η ΤΝ συστηματοποιεί και αυτοματοποιεί τις διανοητικές εργασίες, γι' αυτό και μπορεί να έχει εφαρμογή σε οποιαδήποτε σφαίρα της ανθρώπινης διανοητικής δραστηριότητας. Με αυτή την έννοια, είναι πραγματικά ένα οικουμενικό πεδίο.

Η τεχνητή νοημοσύνη ως τεχνολογία που προκαλεί ανατροπές, έχει τη δυνατότητα να μεταμορφώσει τη διακυβέρνηση. Μπορεί να χρησιμοποιηθεί ως εργαλείο που προσφέρει εξατομικευμένη εμπειρία παροχής υπηρεσιών, που βελτιώνει την αποτελεσματικότητα των back-end διαδικασιών, που ενισχύει τη συμμόρφωση με την πολιτική και βοηθάει στον εντοπισμό της απάτης.

Πολλές κυβερνήσεις θεωρούν την ΤΝ ως στρατηγικό πόρο για την ενίσχυση της ανταγωνιστικότητας της χώρας και την τόνωση της οικονομικής ανάπτυξης. Σύμφωνα με έκθεση της McKinsey & Company¹ του 2018 η τεχνητή νοημοσύνη μπορεί δυνητικά να συμβάλει 13 τρισεκατομμύρια δολάρια στην παγκόσμια οικονομία έως το 2030. Ωστόσο στις περισσότερες χώρες, η έλλειψη ευαισθητοποίησης σχετικά με τις δυνατότητες της τεχνητής νοημοσύνης, η χαμηλή διαθεσιμότητα ή ποιότητα των δεδομένων αλλά και η χαμηλή πρόσβαση σε ψηφιακές δεξιότητες αποτελούν σημαντικά εμπόδια για την υιοθέτηση της τεχνητής νοημοσύνης.

Συμμαντικό ρόλο στην ανάπτυξη ψηφιακών δεξιοτήτων μπορούν να διαδραματίσουν τα μαζικά ανοικτά διαδικτυακά μαθήματα (MOOCs). Τα MOOCs έχουν αλλάξει δραστικά τον τρόπο με τον οποίο μαθαίνουμε και διδάσκουμε. Ο κύριος στόχος των MOOCs είναι να παρέχουν νέες ευκαιρίες σε έναν τεράστιο αριθμό εκπαιδευομένων να παρακολουθήσουν δωρεάν διαδικτυακά μαθήματα από οπουδήποτε στον κόσμο. Τα MOOCs έχουν μοναδικά χαρακτηριστικά που τα καθιστούν ένα αποτελεσματικό μοντέλο τεχνολογικά ενισχυμένης μάθησης στην τριτοβάθμια εκπαίδευση και όχι μόνο.

Η εργασία αυτή περιλαμβάνει την ανάπτυξη ενός διαδικτυακού εκπαιδευτικού προγράμματος (MOOC) για την τεχνητή νοημοσύνη στην ελληνική γλώσσα και απευθύνεται σε επαγγελματίες της πληροφορικής. Το εκπαιδευτικό πρόγραμμα καλύπτει από τα θεμέλια της τεχνητής νοημοσύνης και τη μηχανική μάθηση μέχρι εξειδικευμένα θέματα όπως τα νευρονικά δίκτυα και τη βαθιά μάθηση για την όραση και την επεξεργασία φυσικής γλώσσας. Περιλαμβάνει συγκεκριμένα εργαλεία και παραδείγματα. Το εκπαιδευτικό υλικό χωρίζεται σε 4 ενότητες με 6 υποενότητες η κάθε μια. Η κάθε υποενότητα θεωρητικό μέρος, παρουσίαση, περιπτώσεις χρήσης, πρακτικές ασκήσεις και ερωταπαντήσεις. Στο τέλος κάθε ενότητας υπάρχουν ασκήσεις αυτοαξιολόγησης.

Λέξεις - Κλειδιά: Τεχνητή νοημοσύνη (artificial intelligence), Ηλεκτρονική διακυβέρνηση, Ανοικτό διαδικτυακό μάθημα (Mooc), Μηχανική μάθηση,

Abstract

Today, AI combines a huge variety of sub-fields, covering a spectrum ranging from general areas such as learning and cognition to specific tasks such as chess, proving mathematical theorems, writing poetry, and diagnosing diseases. AI systematises and automates mental tasks and can therefore be applied to any sphere of human mental activity. In this sense, it is truly a universal field

As a disruptive technology, AI has the potential to transform governance. It can be used as a tool that delivers a personalised service experience, that improves the efficiency of back-end processes, that enhances compliance with policy and helps detect fraud.

Many governments see AI as a strategic resource to enhance the country's competitiveness and stimulate economic growth. According to a 2018 McKinsey & Company report, AI can potentially contribute \$13 trillion to the global economy by 2010. However, in most countries, lack of awareness about the potential of AI, low availability or quality of data and low access to digital skills are significant barriers to AI adoption.

Massive open online courses (MOOCs) can play an important role in nurturing digital skills. MOOCs have drastically changed the way we learn and teach. The main goal of MOOCs is to provide new opportunities for a huge number of learners to take free online courses from anywhere in the world. MOOCs have unique features that make them an effective model for technology-enhanced learning in higher education and beyond.

This paper involves the development of an online training course (MOOC) on artificial intelligence in Greek. The online training program covers the fundamental areas of research and practice by introducing the learner to the world of artificial intelligence. The training material is divided into 4 modules with 6 sub-modules each. Each sub-module includes a theoretical part, presentation, use cases, practical exercises and question and answer sessions. At the end of each module there are self-assessment exercises.

Keywords: Artificial intelligence, E-government, Open Online Course (Mooc), Machine learning,

Εισαγωγή

Ο ρόλος της τεχνολογίας στον δημόσιο τομέα συνεχίζει να εξελίσσεται. Δεδομένου του αυξανόμενου όγκου δεδομένων που είναι διαθέσιμα στους οργανισμούς, η χρήση της τεχνητής νοημοσύνης (AI) είναι επιτακτική ανάγκη για την αποτελεσματική, αποδοτική και ηθική χρήση των κυβερνητικών πόρων. Προωθείται αφενός από τις τεχνικές εξελίξεις σε διάφορους τομείς, όπως η μηχανική μάθηση, τα νευρωνικά δίκτυα και η βαθιά μάθηση, και αφετέρου από τις οικονομικές δυνάμεις, καθώς πολλές κυβερνήσεις συνεχίζουν να προσπαθούν να παρέχουν όλο και περισσότερες υπηρεσίες με όλο και λιγότερους πόρους. Η τεχνητή νοημοσύνη προσφέρει τεράστιες δυνατότητες για τη μείωση του κόστους (για παράδειγμα) της παροχής εξατομικευμένων και προσαρμοσμένων υπηρεσιών στους πολίτες.

Η τεχνητή νοημοσύνη αναπτύσσεται εδώ και χρόνια, αλλά εισέρχεται σε ένα κρίσιμο στάδιο της ανάπτυξης και της υιοθέτησής της. Κατά την τελευταία δεκαετία συνέπεσε μια σειρά παραγόντων - συγκεκριμένα, οι βελτιωμένες τεχνικές όπως η βαθιά μάθηση και η αύξηση των διαθέσιμων δεδομένων και της επεξεργαστικής ισχύος των υπολογιστών - που επιτρέπουν την πολύ πιο εκτεταμένη ανάπτυξη αυτής της τεχνολογίας. Αυτό συνεπάγεται πλήθος ευκαιριών, αλλά και κινδύνων και προκλήσεων, και ο τρόπος με τον οποίο θα επιλέξουμε να ανταποκριθούμε σε αυτές, θα έχει εκτεταμένες επιπτώσεις για πολλά χρόνια στο μέλλον. Τόσο το κοινό όσο και οι υπεύθυνοι χάραξης πολιτικής έχουν την ευθύνη να κατανοήσουν τις δυνατότητες και τους περιορισμούς αυτής της τεχνολογίας, καθώς γίνεται όλο και περισσότερο μέρος της καθημερινής μας ζωής. Αυτό απαιτεί επίγνωση του πότε και πού αναπτύσσεται αυτή η τεχνολογία.

Η τεχνητή νοημοσύνη θα έχει σημαντικές επιπτώσεις στους τρόπους με τους οποίους η κοινωνία ζει και εργάζεται. Η τεχνητή νοημοσύνη μπορεί να επιταχύνει την ψηφιακή αναστάτωση στην αγορά εργασίας. Πολλές θέσεις εργασίας θα ενισχυθούν από την ΤΝ, πολλές θα εξαφανιστούν και πολλές νέες, άγνωστες ακόμη θέσεις εργασίας, θα δημιουργηθούν. Μια σημαντική κυβερνητική επένδυση σε **δεξιότητες και κατάρτιση** είναι επιτακτική ανάγκη, εάν αυτή η αναστάτωση πρόκειται να αντιμετωπιστεί με επιτυχία και προς όφελος του εργαζόμενου πληθυσμού και της εθνικής αύξησης της παραγωγικότητας. Αυτή η ανάπτυξη δεν είναι εγγυημένη: πρέπει να γίνει περισσότερη δουλειά για να εξεταστεί πώς μπορεί να χρησιμοποιηθεί η τεχνητή νοημοσύνη για την αύξηση της παραγωγικότητας και δεν πρέπει να θεωρηθεί ως γενική πανάκεια για τα ευρύτερα οικονομικά ζητήματα.

Σύμφωνα με το ομόνυμο Ινστιτούτο UNESCO, η Δια Βίου Μάθηση ενσωματώνει δύο έννοιες: μάθηση και ζωή. Ο συνδυασμός υπονοεί ότι η μάθηση εκτείνεται σε όλες τις διαφορετικές φάσεις της ζωής και της ανάπτυξης ενός ατόμου, όχι μόνο σε περιβάλλοντα τυπικής εκπαίδευσης και κατάρτισης αλλά και σε διάφορους χώρους μάθησης μέσω πολλών μορφών και μέσων. Επομένως, η μάθηση δεν πρέπει να περιοριστεί σε μερικά χρόνια τυπικής εκπαίδευσης. Η δια βίου μάθηση περιλαμβάνει πέντε βασικά στοιχεία: 1)

εκπαίδευση για όλες τις ηλικιακές ομάδες, 2) όλα τα επίπεδα εκπαίδευσης, 3) όλους τους τρόπους μάθησης (επίσημες, μη τυπικές και άτυπες), 4) όλους τους χώρους και τις σφαίρες μάθησης, 5) και ποικιλία σκοπών (Ινστιτούτο UNESCO για τη Δια Βίου Μάθηση, 2022).

Ένα από τα εργαλεία της δια βίου μάθησης αποτελεί το εκπαιδευτικό μοντέλο **MOOC** (Massive Open Online Courses) είναι μια εξαιρετική επιλογή. Πρόκειται για διαδικτυακά μαθήματα περιεχομένου υψηλής ποιότητας με δωρεάν ή χαμηλού κόστους πρόσβαση . Ονομάζονται «μαζικά» λόγω του μεγάλου αριθμού εγγεγραμμένων φοιτητών και «ανοιχτά» επειδή δεν υπάρχουν προϋποθέσεις εγγραφής και κατάρτισης και έχουν τη δυνατότητα απόκτησης πιστοποιητικού.

Πρόσφατες έρευνες αγοράς δείχνουν ότι η ζήτηση για δεξιότητες ΤΝ έχει σχεδόν τριπλασιαστεί τα τελευταία 3 χρόνια και ο αριθμός των σχετικών θέσεων εργασίας έχει αυξηθεί κατά 119%. Οι εργοδότες, ωστόσο, δυσκολεύονται να βρουν υποψηφίους με τον κατάλληλο συνδυασμό δεξιοτήτων. Πέρα από την αυξημένη ζήτηση, το κενό ενισχύεται από την έλλειψη Επαγγελματικής εκπαίδευσης και κατάρτισης (ΕΕΚ) για την απόκτηση σχετικών δεξιοτήτων, δεδομένου ότι η ΤΝ αποτελεί επί του παρόντος αντικείμενο εξειδίκευσης ΤΠΕ που προσφέρεται κυρίως στο υψηλότερο επίπεδο της τριτοβάθμιας εκπαίδευσης.

Υπό αυτό το πρίσμα στο δεύτερο μέρος της διπλωματικής εργασίας δημιουργούμε MOOC με στόχο να είναι σύγχρονο, αυτόνομο, σπονδυλωτό μάθημα για επαγγελματίες ΤΠΕ, οι οποίοι πρέπει να βελτιώσουν τις δεξιότητες, τις γνώσεις και τις ικανότητές τους σε τεχνολογίες ΤΝ και πρακτικές εφαρμογές. Οι επαγγελματίες ΤΠΕ που θα παρακολουθήσουν αυτό το μάθημα θα αποκτήσουν και θα αναπτύξουν τις δεξιότητες που σχετίζονται με την ΤΝ - μαζί με δεξιότητες επίλυσης προβλημάτων, διαχείρισης και σχετικές με τον πελάτη (οριζόντιες) δεξιότητες - που απαιτούνται για να ανταποκριθούν στις σύγχρονες απαιτήσεις του εργασιακού χώρου και να επιτύχουν σε μια ανταγωνιστική αγορά εργασίας. Με την ολοκλήρωση του μαθήματος οι εκπαιδευόμενοι θα είναι σε θέση να κατανοούν τα θεωρητικά θεμέλια και τις πρακτικές εφαρμογές της τεχνητής νοημοσύνης και να αναπτύξουν σχετικές δεξιότητες. Αναλυτικά θα μπορούν να κατανοήσουν τη σχέση μεταξύ της τεχνητής νοημοσύνης και της μηχανικής μάθησης, τις κοινωνικές επιπτώσεις της τεχνητής νοημοσύνης, να αποφασίζουν πώς να επιλέγουν τον κατάλληλο αλγόριθμο μηχανικής μάθησης, να εφαρμόζουν γραμμικές και μη γραμμικές μεθόδους ML σε προβλήματα ταξινόμησης και παλινδρόμησης, να γνωρίζουν τα βασικά στοιχεία των νευρωνικών δικτύων και τον τρόπο κατασκευής τους, να εφαρμόζουν την ανάλυση μεγάλων δεδομένων σε μεγάλα σύνολα δεδομένων, να κατανοούν τις δυνατότητες, τις προκλήσεις και τις συνέπειες της βαθιάς μάθησης και τέλος να εφαρμόζουν βαθιά μάθηση σε προβλήματα φυσικής γλώσσας .

Μέρος 1^ο

Θεωρητική προσέγγιση

1. Ηλεκτρονική Διακυβέρνηση και Τεχνητή Νοημοσύνη

1.1 Η Ηλεκτρονική Διακυβέρνηση

Οι Τεχνολογίες Πληροφοριών και Επικοινωνιών (ΤΠΕ) εξελίσσονται με ταχείς ρυθμούς και αποτελούν σήμερα μέρος του καθημερινού βίου πολλών ανθρώπων ενώ επηρεάζουν σημαντικά τον τρόπο με τον οποίο αυτοί επικοινωνούν, εργάζονται και ζουν. Οι ΤΠΕ συνεισφέρουν στη δημιουργία, διαχείριση και εξαγωγή σωστών πληροφοριών, ενισχύοντας έτσι την ικανότητα λήψης ενημερωμένων αποφάσεων. Η πληροφορία αναδεικνύεται ως σημαντικό αγαθό στην εποχή της παγκοσμιοποίησης, για την ανάπτυξη και την ανταγωνιστικότητα των οργανισμών. Σε αυτήν την τεχνολογική εποχή, οι κυβερνήσεις χρειάζεται να εφαρμόσουν την ανάπτυξη χρησιμοποιώντας πληροφορίες τεχνολογίας για την επιτάχυνση της διαδικασίας ανάπτυξης. Η τεχνολογία των πληροφοριών και επικοινωνιών συνεισφέρει στην αποτελεσματικότητα, στην ταχύτητα διακίνησης πληροφοριών, στην ενίσχυση της επικοινωνίας και συνεργασίας, στην παγκόσμια εμβέλεια και διαφάνεια. Η σημερινή εποχή επισημαίνει τη σημασία της συνδυασμένης και αποτελεσματικής χρήσης της Τεχνολογίας Πληροφορικής και Επικοινωνιών (ΤΠΕ) στον τομέα της διακυβέρνησης. Επομένως, σε αυτήν την εποχή της αυτονομίας, για την εφαρμογή μιας καλής διακυβέρνησης, μία από τις προϋποθέσεις είναι η χρήση των πληροφοριών και της τεχνολογίας των επικοινωνιών, ή αν θέλουμε να το διατυπώσουμε με ένα άλλον όρο που είναι πιο δημοφιλής, η ηλεκτρονική διακυβέρνηση (Putra et al, 2018²).

Υπάρχει μια πληθώρα ορισμών για την Ηλεκτρονική Διακυβέρνηση. Είναι δύσκολο να δοθεί ένας συγκεκριμένος, αλλά μετά από τη βιβλιογραφική επισκόπηση ορισμών της ηλεκτρονικής διακυβέρνησης παρατίθενται ορισμένοι ενδεικτικοί ορισμοί:

- Ηλεκτρονική διακυβέρνηση είναι η χρήση τεχνολογίας πληροφοριών για την υποστήριξη κυβερνητικών επιχειρήσεων, συμμετοχή πολιτών και παροχή κρατικών υπηρεσιών. Αποτελεί ένα ευρύ ορισμό που ενσωματώνει τις τέσσερις βασικές διαστάσεις αντικατοπτρίζει τις λειτουργίες της κυβέρνησης, δηλαδή ηλεκτρονικές υπηρεσίες, ηλεκτρονική δημοκρατία, ηλεκτρονικό εμπόριο και ηλεκτρονική διαχείριση (Dawes, 2002³).
- Η ψηφιακή διακυβέρνηση (Digitalgovernment DG) ή η ηλεκτρονική διακυβέρνηση μπορεί επίσης να οριστεί ως «Η διαδικασία χρήσης πληροφοριών και τεχνολογιών επικοινωνίας που επιτρέπουν την πολιτική και πολιτική συμπεριφορά της κυβέρνησης» (McIver & Elmagarmid, 2002⁴).
- Σύμφωνα με την Ευρωπαϊκή Επιτροπή (2003) «η ηλεκτρονική διακυβέρνηση (eGovernment) ορίζεται ως η χρήση των τεχνολογιών των πληροφοριών και των επικοινωνιών στις δημόσιες διοικήσεις σε συνδυασμό με οργανωτικές αλλαγές και νέες

δεξιότητες του προσωπικού. Σκοπός είναι η βελτίωση των δημόσιων υπηρεσιών, καθώς και η ενίσχυση των δημοκρατικών διαδικασιών και των διαδικασιών στήριξης των δημόσιων πολιτικών» (Europa, 2003)¹. Επιπλέον « η ηλεκτρονική διακυβέρνηση περιλαμβάνει κάτι περισσότερο από τα εργαλεία: περιλαμβάνει την επανεξέταση των οργανισμών και των διαδικασιών και την αλλαγή της συμπεριφοράς, έτσι ώστε οι δημόσιες υπηρεσίες να παρέχονται πιο αποτελεσματικά στους ανθρώπους. Η ηλεκτρονική διακυβέρνηση, που εφαρμόζεται σωστά, δίνει τη δυνατότητα στους πολίτες, τις επιχειρήσεις και τους οργανισμούς να πραγματοποιούν τις αλληλεπιδράσεις τους με την κυβέρνηση ευκολότερα, ταχύτερα και με χαμηλότερο κόστος.»²

- Το Ινστιτούτο Ηλεκτρονικής Διακυβέρνησης (2004) «Η ηλεκτρονική διακυβέρνηση περιλαμβάνει νέα κανάλια πρόσβασης στην κυβέρνηση, νέα στυλ ηγεσίας, νέες μέθοδοι συναλλαγών επιχειρήσεων και νέα συστήματα οργάνωσης και παροχή πληροφοριών και υπηρεσιών. Το δυναμικό του για την ενίσχυση της διαδικασίας διακυβέρνησης είναι ανυπολόγιστο».
- Ο Όμιλος της Παγκόσμιας Τράπεζας (World Bank Group, 2015) αναφέρεται στον όρο «Ηλεκτρονική Διακυβέρνηση» ως τη χρήση τεχνολογιών πληροφοριών από κυβερνητικούς φορείς (όπως Δίκτυα ευρείας περιοχής, Διαδίκτυο και υπολογιστές κινητής τηλεφωνίας) που έχουν τη δυνατότητα να μεταμορφώσουν τις σχέσεις με τους πολίτες, τις επιχειρήσεις και άλλα κυβερνητικά όπλα. Αυτές οι τεχνολογίες μπορούν να εξυπηρετήσουν μια ποικιλία διαφορετικών σκοπών: καλύτερη παροχή κρατικών υπηρεσιών στους πολίτες, βελτιωμένη αλληλεπίδραση με επιχειρήσεις και βιομηχανία, ενδυνάμωση των πολιτών μέσω πρόσβασης σε πληροφορίες ή πιο αποτελεσματική κυβερνητική διαχείριση. Τα οφέλη που προκύπτουν μπορεί να είναι λιγότερη διαφθορά, αυξημένη διαφάνεια, μεγαλύτερη ευκολία, αύξηση εσόδων και/ή μείωση κόστους.».

Η ηλεκτρονική διακυβέρνηση συνήθως εκλαμβάνεται ως η χρήση τεχνολογιών πληροφοριών και επικοινωνιών (ΤΠΕ) από τις κυβερνήσεις σε συνδυασμό με οργανωτική αλλαγή για τη βελτίωση των δομών και των κυβερνητικών δραστηριοτήτων (Field, 2003). Επίσης, η εφαρμογή της ηλεκτρονικής διακυβέρνησης αναμένεται να βοηθήσει τις κυβερνήσεις να παρέχουν υπηρεσίες και να μεταμορφώσουν τις σχέσεις με τους πολίτες, τις επιχειρήσεις και τα άλλα εργαλεία της κυβέρνησης

¹http://publications.europa.eu/resource/cellar/fe5a4ba6-3042-4699-b769-cbff829d909f.0003.02/DOC_2#KEYTERMS

²<https://digital-strategy.ec.europa.eu/el/policies/egovernment>

(Grönlund & Horan, 2005; Guida & Crow, 2009). Η ηλεκτρονική διακυβέρνηση εξακολουθεί να υιοθετείται από την παγκόσμια κοινότητα ως η πιο δημόσια διαδικτυακή μετάβαση υπηρεσιών. Η πρόοδος στις ΤΠΕ επέτρεψε την παράδοση νέων τύπων των κυβερνητικών υπηρεσιών, μέσω ποικίλων ψηφιακών καναλιών, όπως email, smartphones, tablet και έξυπνες κάρτες. Κεντρική θέση στην ηλεκτρονική διακυβέρνηση είναι η ικανότητα παροχής πληροφοριών και υπηρεσιών για την υποστήριξη των επιχειρήσεων και των πολιτών της ευρύτερης κοινότητας, την εξοικονόμηση χρόνου και τη μείωση κόστους (Carter & Bélanger, 2005; Lfstedt, 2007). Πρόσφατη βιβλιογραφία έχει δείξει ότι η ηλεκτρονική διακυβέρνηση μπορεί να είναι επωφελής για την οικονομία και τη μείωση της φτώχειας και της ανισότητας, την εικονική εκπαίδευση και την ταχεία διάδοση ιατρικών πληροφοριών. Ειδικότερα, η ηλεκτρονική διακυβέρνηση προωθεί την οικονομική ανάπτυξη περιορίζοντας τις δυνατότητες των αυθαίρετων και διεφθαρμένων πράξεων και έχει άμεση επιρροή κατά της διαφθοράς στο δημόσιο τομέα (Bannister & Connolly, 2012a; Elbahnasawy, 2014; Kim, 2016).

1.2 Είδη πληροφοριακών συστημάτων Ηλεκτρονικής Διακυβέρνησης

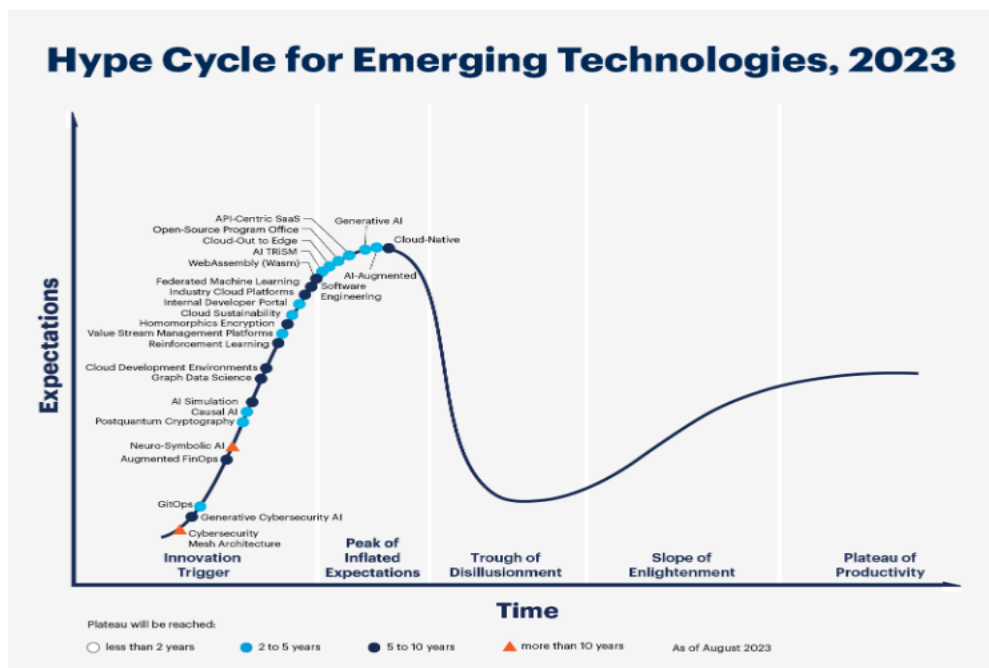
Η ηλεκτρονική διακυβέρνηση εστιάζει σε μεγάλο βαθμό στις ψηφιακές υπηρεσίες και σε διάφορα πρόσθετα μέσα επικοινωνίας μεταξύ κυβερνήσεων κρατών, επιχειρήσεων και πολιτών (McClure, 2000). Ως αποτέλεσμα, η εξέλιξη των συστημάτων ηλεκτρονικής διακυβέρνησης τα τελευταία χρόνια έχει επιτρέψει στις κυβερνήσεις να παρέχουν τις διοικητικές τους υπηρεσίες διαδικτυακά, εστιάζοντας στους πολίτες (Heeks, 2004)¹⁶. Τέσσερις κύριοι τύποι συστημάτων ηλεκτρονικής διακυβέρνησης έχουν περιγραφεί από τους Layne&Lee (2001)¹⁷ και Sharma&Gupta (2002) ως εξής:

- Κυβέρνηση προς Πολίτη- Government-to-Citizen (G2C): αυτό το σύστημα παρέχει πληροφορίες και βασικές υπηρεσίες στους πολίτες, όπως για παράδειγμα ανανεώσεις αδειών, εκδόσεις πιστοποιητικών γέννησης, θανάτου, γάμου και άλλες υπηρεσίες.
- Κυβέρνηση προς Υπάλληλο - Government-to-Employee (G2E): αυτό το σύστημα παρέχει εξειδικευμένες υπηρεσίες που αφορούν τους δημόσιους υπαλλήλους και μόνο.
- Κυβέρνηση προς Επιχείρηση - Government-to-Business (G2B): αυτό το σύστημα επιτρέπει επιχειρηματικές συναλλαγές μεταξύ κυβέρνησης και επιχειρήσεων.

- Κυβέρνηση προς Κυβέρνηση- GovernmenttoGovernment (G2G): αυτό το σύστημα μπορεί να χρησιμοποιηθεί σε τοπικό, εθνικό αλλά και διεθνές επίπεδο.

1.3 Τεχνολογίες που μεταμορφώνουν τις κυβερνήσεις

Η εφαρμογή ψηφιακών καινοτομιών στις κυβερνητικές διαδικασίες, συμπεριλαμβανομένης της διακυβέρνησης, της πολιτικής και των υπηρεσιών, αντανakλά τις γενικές τάσεις στην τεχνολογία, τη δημόσια διοίκηση και τη σχέση κυβέρνησης-πολίτη κατά ένα συγκεκριμένο χρονικό διάστημα (Eggers&Bellman, 2015)¹⁹. Πλαίσια όπως ο κύκλος Hype του Gartner (εισήχθη το 1995) απεικονίζουν τον τρόπο υιοθέτησης τεχνολογιών από κυβερνητικούς οργανισμούς, από τον αρχικό ενθουσιασμό μέχρι μια περίοδο απογοήτευσης και την τελική κατανόηση της σημασίας και του ρόλου της τεχνολογίας στην κυβέρνηση, καθώς και την παραγωγική της χρήση (Linden&Fenn, 2003). Ο κύκλος Hype έχει εφαρμοστεί από διάφορους ερευνητές για να εξηγήσει την ψηφιακή καινοτομία στην κυβέρνηση (Bannister&Connolly, 2012b)²⁰ και να διερευνήσει συγκεκριμένες πρωτοβουλίες όπως οι τρόποι αλληλεπίδρασης κυβέρνησης - πολίτη με δυνατότητα ΤΠΕ (Schellong, 2009), η ανοιχτή κυβέρνηση στις ΗΠΑ (Linders, Wilson&Bertot, 2012), η ηλεκτρονική διακυβέρνηση 2.0 (Boughzala, Janssen&Assar, 2015) και cloudcomputing σε e-Government (Dash&Pani, 2016)²³, μεταξύ άλλων. Η έκδοση του κύκλου Hype του 2018 για τεχνολογία ψηφιακής κυβέρνησης (Holgate, 2018)²⁴ αναφέρεται σε τεχνολογίες που βασίζονται σε τεχνητή νοημοσύνη (AI), Διαδίκτυο των Πραγμάτων (IoT) και Blockchain, μεταξύ άλλων.



Εικόνα 1

Πρόσφατες μελέτες έχουν επικεντρωθεί σημαντικά σε αυτές τις τεχνολογίες ως κύριους παράγοντες αλλαγής για την κυβέρνηση και τη διακυβέρνηση (Engin & Treleaven, 2019). Η υιοθέτηση νέων αναδυόμενων τεχνολογιών αποτελεί απλώς το ξεκίνημα για τη μετάβαση από την Ηλεκτρονική Διακυβέρνηση στον Ψηφιακό Μετασχηματισμό της Κυβέρνησης. Ωστόσο, αυτές οι τεχνολογίες δεν πρέπει να εξετάζονται μεμονωμένα αλλά και σε συνδυασμό με άλλους παράγοντες και τα ιδιαίτερα χαρακτηριστικά τους. Ο πλήρης μετασχηματισμός συχνά συμβαίνει όταν διάφορες τεχνολογίες και καινοτομίες συνδυάζονται, καθώς η αξία του ψηφιακού μετασχηματισμού αφορά περισσότερο τον τρόπο συνεργασίας των κυβερνήσεων με τους πολίτες τους παρά τα εργαλεία που χρησιμοποιούνται. Αυτό επιτρέπεται από την αυξανόμενη διαθεσιμότητα προσωπικής τεχνολογίας και την αυξανόμενη ποσότητα δεδομένων (Eggers & Bellman, 2015).

Ορισμένες από τις κύριες τεχνολογίες που διαδραματίζουν σημαντικό ρόλο στον ψηφιακό μετασχηματισμό της κυβέρνησης περιλαμβάνουν:

- Τεχνητή Νοημοσύνη (AI)
- Συμπεριφορικές και προγνωστικές αναλύσεις
- Ρομποτική και αυτοματοποίηση
- Διαδίκτυο των Πραγμάτων (IoT)
- Γεω-χωρική και εκμετάλλευση δεδομένων τοποθεσίας
- Blockchain - Κατανεμημένος λογιστικός κατάλογος
- Ανοικτά κυβερνητικά δεδομένα και διεπαφές εφαρμογών προγραμματισμού

1.4 Τεχνητή Νοημοσύνη

Η τεχνητή νοημοσύνη (TN) αναφέρεται σε μια ευρεία γκάμα μηχανών ή αλγορίθμων ικανών να παρατηρήσουν το περιβάλλον τους, να μαθαίνουν και να λαμβάνουν έξυπνες αποφάσεις με βάση τις αποκτημένες γνώσεις και εμπειρίας. Για να λειτουργήσουν αποτελεσματικά, οι υπολογιστές χρειάζονται δύο πράγματα τουλάχιστον: μεγάλο όγκο σχετικών δεδομένων και συγκεκριμένους κανόνες για την επεξεργασία αυτών των δεδομένων. Αυτός ο ευρύς ορισμός της τεχνητής νοημοσύνης περιλαμβάνει διάφορες τεχνολογίες, όπως η μηχανική μάθηση, η οποία βελτιώνει την απόδοσή της έχοντας στην διάθεσή της όσο το δυνατόν περισσότερα δεδομένα, η βαθιά μάθηση, η αναλυτική πρόβλεψη, η όραση υπολογιστών και η επεξεργασία φυσικής γλώσσας, μεταξύ άλλων (Annoni, et al., 2018). Αν και ο όρος "τεχνητή νοημοσύνη" υπάρχει από τη δεκαετία του 1950, τα τελευταία χρόνια υπάρχει έντονο ενδιαφέρον, λόγω της αυξανόμενης ποσότητας δεδομένων που συλλέγονται καθημερινά μέσω του Διαδικτύου, των κοινωνικών δικτύων, των τηλεπικοινωνιών,

των ψηφιακών φωτογραφιών, των οικονομικών πλατφορμών και του IoT. Σύμφωνα με εκτιμήσεις, το 2017 παράχθηκαν 2,5 quintillion bytes δεδομένων καθημερινά. Η δημιουργία δεδομένων γίνεται μία κύρια δραστηριότητα των δημόσιων οργανισμών, με την υποστήριξη νέων τεχνολογιών όπως του Δυναδικτού των Πραγμάτων. Η αναλυτική χρήση αυτών των δεδομένων αναμένεται να βελτιώσει τη διαδικασία λήψης αποφάσεων και τον σχεδιασμό πολιτικών, ιδιαίτερα με την εφαρμογή της τεχνητής νοημοσύνης (Janssen, Charalabidis&Krcmar, 2017)²⁶.

Επομένως, η τεχνητή νοημοσύνη κερδίζει ολοένα αυξανόμενο ενδιαφέρον από τους υπεύθυνους πολιτικής. Η Ευρωπαϊκή Επιτροπή προβλέπει ότι η τεχνητή νοημοσύνη μπορεί να έχει σημαντικές επιδράσεις στη ζωή των πολιτών της ΕΕ και να προσφέρει σημαντικά οφέλη στην κοινωνία και την οικονομία, μέσω βελτίωσης της υγειονομικής περίθαλψης, αποτελεσματικότερης δημόσιας διοίκησης, ασφαλέστερων μεταφορών, πιο ανταγωνιστικής βιομηχανίας και βιώσιμης γεωργίας (Ευρωπαϊκή Επιτροπή, 2018α). Έχει προωθήσει μια σειρά πρωτοβουλιών πολιτικής με σκοπό την πρόοδο σε αυτόν τον τομέα, μεταξύ άλλων της επικοινωνίας "Τεχνητή Νοημοσύνη για την Ευρώπη" (Ευρωπαϊκή Επιτροπή, 2018β)²⁷, της δήλωσης συνεργασίας για την AI (Ευρωπαϊκή Επιτροπή, 2018γ) και του συντονισμένου σχεδίου δράσης για την ανάπτυξη της AI στην ΕΕ (Ευρωπαϊκή Επιτροπή, 2018δ). Επίσης, ο ΟΟΣΑ εργάζεται στην ανάλυση και τη μέτρηση της τεχνητής νοημοσύνης. Έχει δημιουργήσει ένα παρατηρητήριο πολιτικής και έχει συγκαλέσει ομάδα εμπειρογνομόνων για την τεχνητή νοημοσύνη - όλα αυτά με στόχο την υποστήριξη εφαρμογών αυτής της τεχνολογίας, συμπεριλαμβανομένου του δημόσιου τομέα (OECD, n.d.a). Επιπλέον, ο ΟΗΕ άνοιξε το Κέντρο UNICRI για Τεχνητή Νοημοσύνη και Ρομποτική, με στόχο να λειτουργήσει ως διεθνής πόρος σε θέματα που σχετίζονται με την τεχνητή νοημοσύνη και τη ρομποτική (UNICRI, n.d.). Σύμφωνα με την ανασκόπηση των Wirtz, Weyerer&Geyer (2018), η βιβλιογραφία σχετικά με την τεχνητή νοημοσύνη στον δημόσιο τομέα επικεντρώνεται σε πέντε μεγάλες κατηγορίες:

1. Κρατική υπηρεσία τεχνητής νοημοσύνης,
2. Εργασιακό και κοινωνικό περιβάλλον επηρεασμένο από την τεχνητή νοημοσύνη,
3. Δημόσια τάξη καινομοθεσία που αφορούν την τεχνητή νοημοσύνη,
4. Ηθική τεχνητής νοημοσύνης και
5. Πολιτική κυβέρνησης για την τεχνητή νοημοσύνη.

Πρόσφατες μελέτες έχουν εξετάσει μια ποικιλία εφαρμογών της τεχνητής νοημοσύνης στον δημόσιο τομέα, από μοντέλα για την πρόβλεψη των τιμών για έργα δημόσιας υποδομής (Chou, Lin, Pham&Shao, 2015)²⁹ ή τη χρήση των δεδομένων της πόλης ως εργαλείο πολιτικής για την προώθηση των στόχων της αστικής ανάπτυξης (Pan, Tian, Liu, Gu&Hua, 2016)³⁰, έως την εφαρμογή της τεχνητής νοημοσύνης στην ιατρική διάγνωση και θεραπεία, τις δυνατότητες της τεχνητής

νοημοσύνης να μεταμορφώσει το κυβερνητικό εργατικό δυναμικό (Sun, T.Q., & Medaglia, 2019)³¹ και πολλές άλλες περιπτώσεις.

Παρά τα αυξανόμενα παραδείγματα χρήσης και την εξέλιξη της συζήτησης σχετικά με τα πλεονεκτήματα και τους κινδύνους της υιοθέτησης της τεχνητής νοημοσύνης στον δημόσιο τομέα, υπάρχουν περιορισμένα δεδομένα για τις πραγματικές επιπτώσεις (Ηνωμένα Έθνη, 2018). Το προηγούμενο δεν προκαλεί εντύπωση δεδομένου ότι οι υφιστάμενες πρακτικές και λύσεις τεχνητής νοημοσύνης στον δημόσιο τομέα βρίσκονται στα αρχικά στάδια της αυτοματοποίησης διαδικασιών και της προβλεπτικής ανάλυσης (Tinholt, Carrara&vanderLinden, 2017)³².

2. Ψηφιακός Μετασχηματισμός

Η τέταρτη Βιομηχανική Επανάσταση (4IR) είναι σε πλήρη εξέλιξη και χαρακτηρίζεται από την ευρεία χρήση αναδυόμενων τεχνολογιών, οι οποίες δημιουργούν νέες προκλήσεις και ευκαιρίες. Σημαντικές τεχνολογίες σε αυτό το πλαίσιο περιλαμβάνουν τη ρομποτική, τη νανοτεχνολογία, την επεξεργασία του ανθρώπινου γονιδιώματος, τις νευροεπιστήμες, την εικονική πραγματικότητα (VR), την επαυξημένη πραγματικότητα (AR), την τρισδιάστατη εκτύπωση, το Διαδίκτυο των Πραγμάτων (IoT), τα κυβερνοφυσικά συστήματα (Cyber-Physical Systems), τη μηχανική μάθηση και την ανάλυση μεγάλων δεδομένων, την επεξεργασία φυσικής γλώσσας, το Υπολογιστικό Νέφος (CloudComputing), και τις τεχνολογίες blockchain, μεταξύ άλλων. Αυτές οι τεχνολογίες μετασχηματίζουν την κοινωνία, την οικονομία και την εκπαίδευση, ενώ θολώνουν τα όρια μεταξύ της φυσικής, ψηφιακής και βιολογικής σφαίρας.

Σύμφωνα με τον Schwab (2016), η 4IR σηματοδοτείται από τη συνδυασμένη εφαρμογή αυτών των τεχνολογιών, με στόχο να διευκολύνει την ανθρώπινη δραστηριότητα με ταχύτητα και κλιμάκωση. Ωστόσο, αυτή η εξέλιξη συνεπάγεται τόσο δυνητικούς κινδύνους όσο και ευκαιρίες, δημιουργεί νέες γνώσεις και αναδιαμορφώνει τις θέσεις εργασίας και τις δεξιότητες. Έτσι, αναδύεται η ανάγκη μετάβασης από παραδοσιακούς τρόπους λειτουργίας σε νέες μορφές οργάνωσης της ανθρώπινης δραστηριότητας, μια διαδικασία που περιγράφεται ως "Ψηφιακός Μετασχηματισμός" και στον τομέα της εκπαίδευσης ως "Εκπαίδευση 4.0", προσαρμοσμένη στις απαιτήσεις της 4IR. (Gonzalez-Perez&Ramirez-Montoya, 2022)³³.

Σύμφωνα με τους Αραβαντινού - Φατώρου και Καλογρίδη (2021)³⁴, η Εκπαίδευση 4.0 εισάγει την έννοια της επιχειρηματικότητας και των δεξιοτήτων, οι οποίες αποτελούν το προφίλ του πολίτη της νέας ψηφιακής εποχής. Βασικά στοιχεία αυτής της εκπαίδευσης είναι η εισαγωγή νέων αντικειμένων στα αναλυτικά προγράμματα, η ανάπτυξη ικανοτήτων και δεξιοτήτων, η

επιχειρηματικότητα, η εξ αποστάσεως εκπαίδευση, η διά βίου μάθηση, καθώς και η ανατροφοδότηση και η αξιολόγηση. Η Εκπαίδευση 4.0 δεν αντιπροσωπεύει μια εξέλιξη της εκπαίδευσης αυτή καθαυτή, αλλά είναι μάλλον ένας συμβολισμός για να τονιστούν οι αλλαγές στην εκπαίδευση που σχετίζονται με την 4IR. Αυτές οι αλλαγές περιλαμβάνουν μια εκπαίδευση που καθοδηγείται από τη ζήτηση, βασίζεται στις δεξιότητες, ενσωματώνει τεχνολογίες σύμφωνα με τη νέα πραγματικότητα και υποστηρίζεται από τη διά βίου μάθηση, με έμφαση στη συναισθηματική νοημοσύνη.

Ο όρος "μετασχηματισμός" αναφέρεται σε μια ριζοσπαστική αλλαγή ή μεταβολή, που δημιουργεί μια νέα και βελτιωμένη μορφή λειτουργίας ή δομής. Αυτή η διαδικασία απαιτεί ένα εύλογο χρονικό διάστημα για την εφαρμογή της αλλαγής και της καινοτομίας (Πριμικήρη, 2020)³⁵. Ο ψηφιακός μετασχηματισμός αφορά τις διαδικασίες μέσω των οποίων μια επιχείρηση ή ένας οργανισμός, είτε δημόσιος είτε ιδιωτικός, προσαρμόζεται και αναπτύσσεται για να εκμεταλλευτεί τα πλεονεκτήματα που προσφέρουν οι νέες τεχνολογίες. Πρόκειται για μια συνεχή διαδικασία που δημιουργεί ευκαιρίες, αλλά απαιτεί εκπαίδευση και υποδομές.

2.1 Δείκτης Ψηφιακής Οικονομίας και Κοινωνίας (DESI)

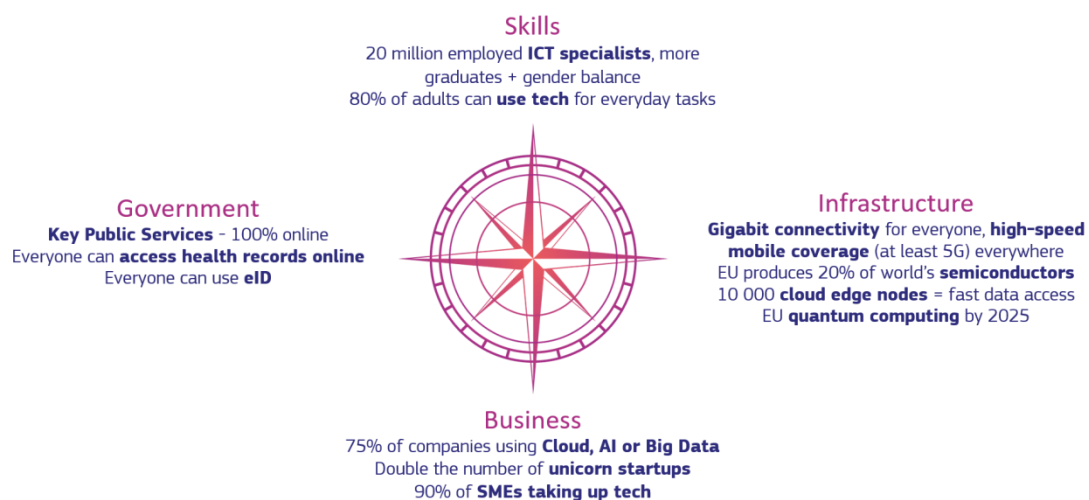
Ο δείκτης Ψηφιακής Οικονομίας και Κοινωνίας (DESI) έχει αποτελέσει σημαντικό εργαλείο για την παρακολούθηση της ψηφιακής προόδου των κρατών μελών της Ευρωπαϊκής Ένωσης από το 2014 και έπειτα. Η Ευρωπαϊκή Επιτροπή δημοσιεύει ετήσιες εκθέσεις DESI που παρέχουν αναλυτικές πληροφορίες σχετικά με την κατάσταση της ψηφιακής οικονομίας και κοινωνίας σε κάθε χώρα μέλος. Οι εκθέσεις περιλαμβάνουν αξιολογήσεις για τους τομείς προτεραιότητας και θεματικά κεφάλαια που αναλύουν την εξέλιξη της ψηφιακής πολιτικής σε επίπεδο ΕΕ. Ο δείκτης DESI κατατάσσει τα κράτη μέλη με βάση το επίπεδο της ψηφιοποίησής τους και αναλύει τη σχετική πρόοδο τους τα τελευταία χρόνια, λαμβάνοντας υπόψη το αρχικό σημείο εκκίνησής τους. Επίσης, προσαρμόζεται σε σχέση με τους στόχους που καθορίζονται στο πρόγραμμα πολιτικής για την πορεία προς την ψηφιακή δεκαετία.

Τα βασικά σημεία του προγράμματος πολιτικής της ψηφιακής δεκαετίας περιλαμβάνουν την ψηφιακή εκπαίδευση, την ασφάλεια και βιωσιμότητα των ψηφιακών υποδομών, τον ψηφιακό μετασχηματισμό των επιχειρήσεων και τη ψηφιοποίηση των δημόσιων υπηρεσιών. Αυτοί οι στόχοι και οι επιδιώξεις θέτουν τη βάση για μια ολοκληρωμένη και βιώσιμη ψηφιακή μετασχηματική διαδικασία σε όλους τους τομείς της οικονομίας.

2.2 Στόχοι και επιδιώξεις³ της πολιτικής

Το πρόγραμμα πολιτικής για την ψηφιακή δεκαετία θέτει συγκεκριμένους στόχους για τη μελλοντική ανάπτυξη της ψηφιακής κοινωνίας και οικονομίας. Συνοψίζοντας τους κύριους στόχους αυτούς σε 4 σημεία έχουμε τα παρακάτω:

1. **Εκπαίδευση και κατάρτιση:** Η δημιουργία ενός πληθυσμού με ψηφιακές δεξιότητες είναι ζωτικής σημασίας για την αντιμετώπιση των προκλήσεων του ψηφιακού κόσμου. Αυτό περιλαμβάνει επίσης και την κατάρτιση υψηλά εξειδικευμένων επαγγελματιών στον ψηφιακό τομέα.
2. **Ασφάλεια και βιωσιμότητα των ψηφιακών υποδομών:** Η ανάπτυξη και η διατήρηση ψηφιακών υποδομών που είναι ασφαλείς και βιώσιμες είναι ουσιώδους σημασίας για την ασφάλεια και τη σταθερότητα του ψηφιακού περιβάλλοντος.
3. **Ψηφιακός μετασχηματισμός των επιχειρήσεων:** Οι επιχειρήσεις πρέπει να προσαρμόσουν τις δραστηριότητές τους στον ψηφιακό κόσμο, ενσωματώνοντας ψηφιακές τεχνολογίες και διαδικασίες για τη βελτίωση της απόδοσης και της ανταγωνιστικότητάς τους.
4. **Ψηφιοποίηση των δημόσιων υπηρεσιών:** Η ψηφιοποίηση των δημόσιων υπηρεσιών επιτρέπει την αποτελεσματική παροχή υπηρεσιών στους πολίτες και τις επιχειρήσεις, μειώνοντας το χρόνο και το κόστος για όλους τους ενδιαφερόμενους.



Εικόνα 2

Παράλληλα με τους στόχους, οι επιδιώξεις της Ψηφιακής Δεκαετίας θα διασφαλίσουν ότι ο ψηφιακός μετασχηματισμός στην Ευρώπη θα ωφελήσει όλους τους ανθρώπους. Οι επιδιώξεις μπορούν να συνοψιστούν ως εξής:

1. **Ασφαλής ψηφιακός κόσμος:** Ο στόχος είναι να δημιουργηθεί ένα περιβάλλον στο διαδίκτυο που είναι ασφαλές και προστατευμένο για τους χρήστες.

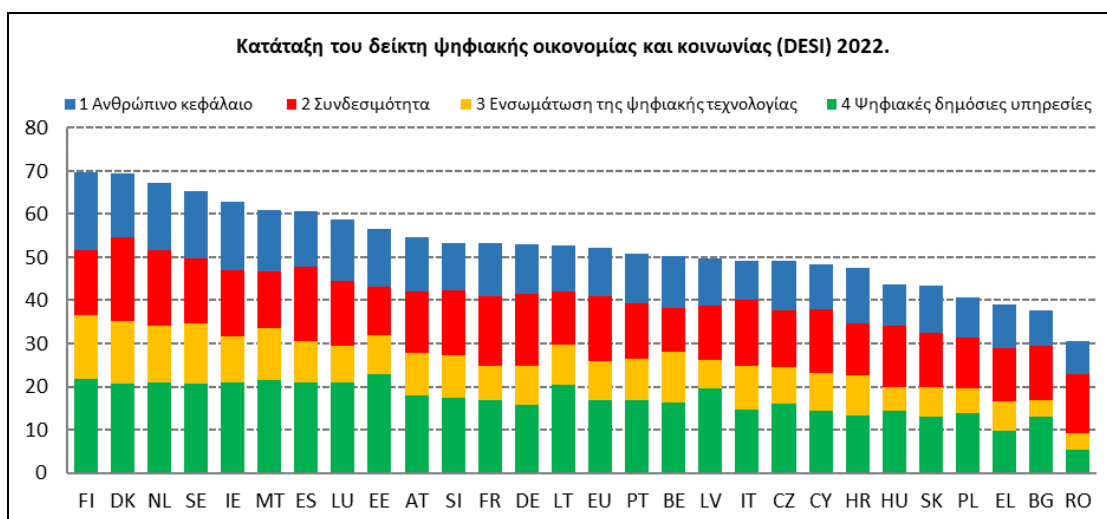
³https://digital-strategy.ec.europa.eu/en/policies/europes-digital-decade#tab_1

2. **Πρόσβαση σε ψηφιακές ευκαιρίες για όλους:** Κάθε άτομο, ανεξαρτήτως θέσης ή κοινωνικής ομάδας, πρέπει να έχει τη δυνατότητα να επωφεληθεί από τις ευκαιρίες που παρέχει ο ψηφιακός κόσμος.
3. **Πρόσβαση σε δεδομένα για μικρές επιχειρήσεις και βιομηχανία:** Η διαθεσιμότητα δεδομένων για μικρές επιχειρήσεις και βιομηχανίες είναι κρίσιμη για την ανάπτυξη και την καινοτομία.
4. **Πρόσβαση στην ψηφιακή τεχνολογία για νεοφυείς επιχειρήσεις και ΜΜΕ:** Να διασφαλιστεί ότι νέες και μικρές επιχειρήσεις έχουν πρόσβαση στα εργαλεία και την τεχνολογία που χρειάζονται για την επιτυχία τους.
5. **Συνεργασία μεταξύ καινοτόμων υποδομών:** Επιτρέπει την αξιοποίηση συνεργιών και τη δημιουργία νέων ευκαιριών.
6. **Ανταγωνισμός με δίκαιους όρους:** Να διασφαλιστεί ότι ο ανταγωνισμός στον ψηφιακό κόσμο είναι δίκαιος και ισότιμος για όλους.
7. **Δημόσιες υπηρεσίες προσιτές στο διαδίκτυο:** Η διασφάλιση ότι οι δημόσιες υπηρεσίες είναι προσιτές και εύκολα προσβάσιμες μέσω του διαδικτύου.
8. **Έρευνα και καινοτομία με βιωσιμότητα:** Η έρευνα και η καινοτομία πρέπει να επικεντρώνονται στη βιωσιμότητα και στον κοινωνικό αντίκτυπο.
9. **Ασφάλεια στον κυβερνοχώρο:** Η ασφάλεια στον κυβερνοχώρο είναι κρίσιμη για την προστασία των ατόμων και των επιχειρήσεων από κυβερνοεπιθέσεις.

Τα κράτη μέλη θα υποβάλουν έκθεση στην Επιτροπή σχετικά με τις προγραμματισμένες δράσεις που υποστηρίζουν τους καθορισμένους στόχους.

Οι δράσεις και οι πολιτικές της Επιτροπής στον ψηφιακό τομέα καθοδηγούνται ήδη από αυτούς τους στόχους και τις αρχές. Καθώς θα έχουμε όλο και περισσότερες καινοτομίες, το πλαίσιο για την ψηφιακή δεκαετία θα διασφαλίσει ότι το ευρωπαϊκό όραμα για τον ψηφιακό μετασχηματισμό θα είναι σαφές και θα υποστηρίζεται ευρέως από όλες τις μελλοντικές δράσεις σε ολόκληρη την Ευρώπη.

Η Ελλάδα κατατάσσεται 25η μεταξύ των 27 κρατών μελών της ΕΕ στην έκδοση του δείκτη ψηφιακής οικονομίας και κοινωνίας (DESI) για το 2022. Ωστόσο, συνολικά, η Ελλάδα σημείωσε ικανοποιητική πρόοδο τα τελευταία έτη συγκριτικά με άλλα κράτη μέλη της ΕΕ, το οποίο σημαίνει ότι η Ελλάδα καλύπτει τη διαφορά.



Εικόνα 3

ΔΕΙΚΤΗΣ DESI 2022	Ελλάδα		ΕΕ
	κατάταξη	βαθμολογία	βαθμολογία
	25	38,9	52,3

Στο πλαίσιο της Βίβλου Ψηφιακού Μετασχηματισμού, η τεχνητή νοημοσύνη (TN) είναι ένας σημαντικός τομέας δράσης. Παρά τη σημαντική αυτή προοπτική, η εθνική στρατηγική της Ελλάδας για την τεχνητή νοημοσύνη εξακολουθεί να είναι στο στάδιο της προετοιμασίας. Τον Δεκέμβριο του 2021, ο «Αρχιμήδης» (προϋπολογισμός: 21εκ. ευρώ), ιδρύθηκε ως μονάδα του ερευνητικού κέντρου καινοτομίας «Αθηνά», του μοναδικού ερευνητικού κέντρου στην Ελλάδα που επικεντρώνεται αποκλειστικά στις νέες τεχνολογίες διαχείρισης πληροφορίας, υπολογιστικών μεθόδων και επικοινωνιών, με κύριο στόχο να αποτελέσει πύλη εισόδου για τους κορυφαίους επιστήμονες στον τομέα της τεχνητής νοημοσύνης. Αυτή η ερευνητική μονάδα συνεργάζεται τόσο με ελληνικά όσο και με ξένα πανεπιστήμια, δημιουργώντας ευκαιρίες συνεργασίας και ανταλλαγής ιδεών για την εκπαίδευση, την έρευνα και την καινοτομία στους τομείς της Τεχνητής Νοημοσύνης, της Επιστήμης Δεδομένων και των Αλγορίθμων, μεταξύ κορυφαίων ακαδημαϊκών και νέων επιστημόνων στην Ελλάδα και διεθνώς. Διοργανώνει συστηματικά επιμορφωτικά και εκπαιδευτικά προγράμματα, όπου επιτυχημένοι επιστήμονες από το εξωτερικό και την Ελλάδα μοιράζονται τις εξειδικευμένες γνώσεις τους με το οικοσύστημα του «Αρχιμήδη». Επίσης, προωθεί τη συνεργασία με το αναπτυσσόμενο οικοσύστημα των νεοφυών επιχειρήσεων στην Ελλάδα, ενθαρρύνοντας τη μεταφορά των αποτελεσμάτων της έρευνας στην αγορά.

Το Εθνικό Δίκτυο Υποδομών Τεχνολογίας και Έρευνας (ΕΔΥΤΕ, GRNET) παρέχει υπολογιστική υψηλών επιδόσεων (HPC) μέσω του συστήματος πληροφοριών για την έρευνα (ARIS) από το 2015, με την πρόσφατη δρομολόγηση διαδικασίας επέκτασης του συστήματος (από το 2021 με προϋπολογισμό: 23 εκ. ευρώ) σε συνεργασία με την επιχείρηση EuroHPC. Η βελτίωση του εθνικού κέντρου υπολογιστικής υψηλών επιδόσεων Τεχνητής Νοημοσύνης λαμβάνει επίσης υποστήριξη

από αντίστοιχα κέντρα υπολογιστικής υψηλών επιδόσεων σε διάφορες χώρες των Βαλκανίων και της Κύπρου. Η συνολική επένδυση σε υλικό (hardware) θα στηρίξει την εγκατάσταση μηχανήματος 30PFloP+ στην περιφέρεια Αττικής. Αυτό αποτελεί ένα σημαντικό βήμα για την ενίσχυση της υπολογιστικής υποδομής στην Ελλάδα. Η συνεργασία με ελληνικά και ξένα πανεπιστήμια, καθώς και η στενή συνεργασία με τα ακαδημαϊκά ιδρύματα, ενισχύει την τεχνολογική καινοτομία στην Ελλάδα. Τέλος, η συνεργασία με τις ΜΜΕ και τη βιομηχανία βοηθάει στη διάδοση των επιστημονικών εξελίξεων στην ευρύτερη οικονομία και ενισχύει την ανταγωνιστικότητα του τομέα.

	ΔΕΙΚΤΗΣ DESI 2020	Ελλάδα ΔΕΙΚΤΗΣ DESI 2021	ΔΕΙΚΤΗΣ DESI 2022	ΕΕ ΔΕΙΚΤΗΣ DESI 2022
3α1 ΜΜΕ με τουλάχιστον βασικό επίπεδο ψηφιακής έντασης % των ΜΜΕ	ά.α.	ά.α.	39 % 2021	55 % 2021
3β1 Ηλεκτρονική ανταλλαγή πληροφοριών % των επιχειρήσεων	38 % 2019	38 % 2019	35 % 2021	38 % 2021
3β2 Μέσα κοινωνικής δικτύωσης % των επιχειρήσεων	19 % 2019	19 % 2019	29 % 2021	29 % 2021
3β3 Μαζικά δεδομένα % των επιχειρήσεων	13 % 2018	13 % 2020	13 % 2020	14 % 2020
3β4 Υπολογιστικό νέφος % των επιχειρήσεων	ά.α.	ά.α.	17 % 2021	34 % 2021
3β5 Τεχνητή νοημοσύνη % των επιχειρήσεων	ά.α.	ά.α.	4 % 2021	8 % 2021
3β6 ΤΠΕ για την περιβαλλοντική βιωσιμότητα % των επιχειρήσεων που έχουν μεσαία/υψηλή ένταση πράσινης δράσης μέσω ΤΠΕ	ά.α.	65 % 2021	65 % 2021	66 % 2021
3β7 Ηλεκτρονικά τιμολόγια % των επιχειρήσεων	9 % 2018	ά.α. 2020	ά.α. 2020	32 % 2020
3γ1 ΜΜΕ που πραγματοποιούν πωλήσεις μέσω διαδικτύου % των ΜΜΕ	9 % 2019	ά.α. 2020	20 % 2021	18 % 2021
3γ2 Κύκλος εργασιών ηλεκτρονικού εμπορίου % του κύκλου εργασιών των ΜΜΕ	4 % 2019	ά.α. 2020	11 % 2021	12 % 2021
3γ3 Διασυννοριακές ηλεκτρονικές πωλήσεις % των ΜΜΕ	4 % 2019	4 % 2019	7 % 2021	9 % 2021

Εικόνα 4

3 Ενσωμάτωση της ψηφιακής τεχνολογίας	Ελλάδα		ΕΕ
	κατάταξη	βαθμολογία	βαθμολογία
	ΔΕΙΚΤΗΣ DESI 2022	22	26,6

2.3 Ψηφιακές Δεξιότητες και Τεχνητή νοημοσύνη

⁴Σύμφωνα με τη Βίβλο Ψηφιακού Μετασχηματισμού 2020-25 του Υπουργείου Ψηφιακής Διακυβέρνησης, η Τέταρτη Βιομηχανική Επανάσταση προκαλεί και προσκαλεί τη χώρα σε μια πλήρη συνεργασία με τις ψηφιακές τεχνολογίες. Αυτό επιτρέπει στους ανθρώπους να

⁴ [ETΠΕ \(etpe.gr\)](https://etpe.gr)

προλαμβάνουν τις ταχύτητες με τις οποίες οι μηχανές εξελίσσονται, θέτοντας εγκαίρως το πλαίσιο των ηθικών αξιών, των αρχών και των κανόνων. Αυτό το πλαίσιο είναι απαραίτητο για να διασφαλιστεί η ανθρώπινη υπόσταση και αξιοπρέπεια, καθώς και για τη διατήρηση των δημοκρατικών θεσμών της χώρας. Η γνώση, η εκπαίδευση και η ανάπτυξη των ψηφιακών ικανοτήτων και δεξιοτήτων αναδεικνύονται ως η κινητήρια δύναμη πίσω από τη δυνατότητα των πολιτών να ασκούν ενεργό ρόλο και να συμβάλλουν στο πλαίσιο υιοθέτησης και αντιμετώπισης των αλλαγών. Επιπλέον, η Ευρωπαϊκή Ένωση, μέσω του Ευρωπαϊκού Σχεδίου Δράσης για την Ψηφιακή Εκπαίδευση 2021-27, επισημαίνει επειγόντως αυτήν την πτυχή μέσω της Στρατηγικής Προτεραιότητας 2, η οποία επικεντρώνεται στην ενίσχυση των ψηφιακών δεξιοτήτων και ικανοτήτων για την προώθηση του ψηφιακού μετασχηματισμού.

Σύμφωνα με το Cedefop (2008), η έννοια της "δεξιότητας" αφορά την ικανότητα εκτέλεσης εργασιών και επίλυσης προβλημάτων, ενώ η έννοια της "ικανότητας" αφορά την αποτελεσματική εφαρμογή μαθησιακών αποτελεσμάτων σε ένα συγκεκριμένο πλαίσιο, όπως εκπαίδευση, εργασία ή προσωπική ανάπτυξη. Ο ψηφιακός γραμματισμός, ή ψηφιακή ικανότητα, αναπαριστά το σύνολο των ικανοτήτων και δεξιοτήτων που απαιτούνται στην σύγχρονη κοινωνία, όπου η επικοινωνία και η τεχνολογική πρόοδος είναι κυρίαρχα χαρακτηριστικά. Αυτές οι ικανότητες και δεξιότητες εφαρμόζονται σε διάφορους τομείς της κοινωνικής ζωής, όπως η εκπαίδευση, η εργασία, ο ελεύθερος χρόνος και άλλες πτυχές της καθημερινής ζωής (Sofos, 2011·Σοφός, 2014).

Στην αναφορά της Ευρωπαϊκής Ένωσης σχετικά με την δια βίου μάθηση (EU, 2019), η ψηφιακή ικανότητα (digital competence) του πολίτη αναφέρεται στη δεξιότητα κριτικής και υπεύθυνης χρήσης των ψηφιακών τεχνολογιών σε διάφορους τομείς, όπως η μάθηση, η εργασία και η συμμετοχή στην κοινωνία και είναι μια παό τις οκτώ βασικές ικανότητες. Συγκεκριμένα, περιλαμβάνει δεξιότητες όπως ο πληροφοριακός γραμματισμός, η επικοινωνία και η συνεργασία, η κατανόηση των μέσων επικοινωνίας, η δημιουργία ψηφιακού περιεχομένου και ο προγραμματισμός, καθώς και θέματα ασφάλειας και πνευματικής ιδιοκτησίας. Αυτές οι δεξιότητες επιτρέπουν στα άτομα να συμμετέχουν ενεργά στην κοινωνία και να επιτυγχάνουν προσωπικούς, κοινωνικούς ή επαγγελματικούς στόχους. Περιλαμβάνουν την ικανότητα χρήσης, πρόσβασης, αξιολόγησης και δημιουργίας ψηφιακού περιεχομένου, καθώς και τη διαχείριση και προστασία πληροφοριών και ψηφιακών ταυτοτήτων. Επιπλέον, αναγνωρίζουν και αλληλεπιδρούν με διάφορες τεχνολογίες, όπως λογισμικό, συσκευές, τεχνητή νοημοσύνη και ρομπότ.

Στο Ευρωπαϊκό Πλαίσιο DigComp 2.2 (Vuorikari, Kluzer&Runie, 2022)³⁶, η έννοια της ψηφιακής ικανότητας αναλύεται σε πέντε βασικές περιοχές, κάθε μία από τις οποίες αναλύεται σε διάφορες επιμέρους ικανότητες και επίπεδα επίδοσης: πληροφοριακός γραμματισμός, επικοινωνία και συνεργασία, δημιουργία ψηφιακού περιεχομένου, ασφάλεια και επίλυση προβλημάτων. Με εστίαση στον τομέα της τεχνητής νοημοσύνης (TN) και της ανάλυσης δεδομένων ως βασικού πυλώνα της 4IR και του ψηφιακού μετασχηματισμού, έρευνα της Ε.Ε. (Righi, López-Cobo&Alaveras, 2020)³⁷ ανέδειξε ως βασικό σύνολο προηγμένων ψηφιακών δεξιοτήτων τη ρομποτική και τον αυτοματισμό, τη μηχανική μάθηση, την ανάλυση μεγάλων δεδομένων, την ηθική και δεοντολογία της Τεχνητής Νοημοσύνης, καθώς και την κυβερνοασφάλεια σε επίπεδο συστημάτων και προσωπικών δεδομένων.

3. ⁵Διά βίου Μάθηση

Όπως αναδεικνύεται στα έγγραφα της Ευρωπαϊκής Ένωσης, οι τεχνολογικές μεταβολές και οι οικονομικές αλλαγές επηρεάζουν τις απαιτούμενες γνώσεις, τη νοοτροπία, συμπεριφορά και τις αξίες των πολιτών. Είναι αναγκαίο να αναζητούν συνεχώς νέες γνώσεις, δεξιότητες και ικανότητες προκειμένου να αναβαθμιστούν ή να επανεκταχθούν στην αγορά εργασίας. Σε αυτήν την εκπαιδευτική λογική, οι παραδοσιακές βαθμίδες εκπαίδευσης σταδιακά ενσωματώνονται, διαμορφώνοντας ένα εκπαιδευτικό περιβάλλον όπου η μάθηση συνδέεται με την εργασία και υπογραμμίζεται η σημασία της δεξιότητας και της κατάρτισης (Καλεράντε και Ελευθεράκης, 2014). Επιπλέον, η έννοια της Δια Βίου Μάθησης περιλαμβάνει όλες τις εκπαιδευτικές δραστηριότητες ανεξαρτήτως τύπου, βαθμίδας και περιεχομένου, οι οποίες λαμβάνουν χώρα σε διάφορα εκπαιδευτικά πλαίσια (τυπικά, μη τυπικά, άτυπα) και συμμετέχουν άτομα κάθε ηλικίας και επιπέδου παιδείας, σε διάφορα στάδια της ζωής τους (Καραλής, 2013). Από την άλλη πλευρά, οι έννοιες της Δια Βίου Μάθησης και της Δια Βίου Εκπαίδευσης διαφοροποιούνται, σύμφωνα με τον Καραλή (2013), με τη δεύτερη να αφορά σε ένα οργανωμένο/θεσμικό εκπαιδευτικό πλαίσιο, ενώ η πρώτη θεωρείται ως ένα ιδεώδες ή μία ατομική ευθύνη και δικαίωμα, βάσει της οποίας πρέπει να υπάρχουν πολιτικές που θα εξασφαλίζουν τις απαραίτητες προϋποθέσεις για την ανάπτυξή της, όπως ευκαιρίες, κίνητρα και δυνατότητες για μάθηση.

3.1 Μαζικά Ανοικτά Διαδικτυακά Μαθήματα (MOOCs)

Τα Μαζικά Ανοικτά Διαδικτυακά Μαθήματα (MOOCs) εμφανίστηκαν για πρώτη φορά το 2008. Σήμερα αναπτύσσονται, κυρίως, από μεγάλα τριτοβάθμια ιδρύματα και παρακολουθούνται από χιλιάδες άτομα, απ' όλο τον κόσμο. Όσοι εγγράφονται σε κάποιο πρόγραμμα δεν πληρώνουν δίδακτρα, ούτε απαιτούνται κάποια κριτήρια για να τα παρακολουθήσουν ακόμα και αν προτείνεται από τον δημιουργό τους η κατοχή συγκεκριμένων γνώσεων και δεξιοτήτων για να μπορεί να γίνει κατανοητό το περιεχόμενο τους. Το εκπαιδευτικό υλικό παρέχεται μέσω σύντομων βίντεο, παρουσιάσεων ή άλλων ψηφιακών μέσων (Hoy, 2014)³⁹. Αυτό αναρτάται σε online πλατφόρμες, όπου οι ενδιαφερόμενοι μπορούν να αναζητήσουν, να εγγραφούν σε μαθήματα ή ακόμη και να δημιουργήσουν δικά τους μαθήματα. Για την αξιολόγηση των συμμετεχόντων, ανατίθενται εργασίες που αξιολογούνται από άλλους συμμετέχοντες, εκπαιδευτικούς ή από αυτοματοποιημένα κομμάτια. Οι συμμετέχοντες λαμβάνουν δωρεάν ένα ηλεκτρονικό πιστοποιητικό μετά την επιτυχή ολοκλήρωση του προγράμματος, ενώ μπορούν επίσης να αποκτήσουν επίσημο πιστοποιητικό με την καταβολή ενός αντιτίμου και την συμμετοχή τους σε εξετάσεις (Karnouskos & Holmlund, 2014)⁴⁰.

3.2 Πλατφόρμες φιλοξενίας μαθημάτων MOOCs

Αν και τα πρώτα MOOCs δεν χρησιμοποιούσαν κάποιο συγκεκριμένο χώρο φιλοξενίας, σήμερα, παράλληλα με την αύξηση του ενδιαφέροντος γι' αυτά, έχουν δημιουργηθεί διάφορες online

⁵ [ETΠΕ \(etpe.gr\)](http://etpe.gr), σελ.5

πλατφόρμες, που διαφέρουν μεταξύ τους ως προς το σκοπό για τον οποίο χρησιμοποιούν τα κέρδη τους. Κατά συνέπεια, χωρίζονται σε κερδοσκοπικές, που σκοπό έχουν να αυξήσουν τα εισοδήματα των ιδιοκτητών και των μετόχων τους και σε μη κερδοσκοπικές, των οποίων επιδίωξη είναι να χρησιμοποιούν τα έσοδά τους για να πετύχουν το σκοπό ή την αποστολή για την οποία δημιουργήθηκαν (Hansmann, 1980). Οι πιο γνωστές κερδοσκοπικές πλατφόρμες είναι η Coursera, η Udacity και η Udemy, ενώ οι πιο γνωστές μη κερδοσκοπικές, η Edx, η Khan Academy και η FutureLearn.

Σε ευρωπαϊκό επίπεδο, η Ευρωπαϊκή Επιτροπή δημιούργησε το portalOpenEducation Europa, τον Σεπτέμβριο του 2013, ως μια πύλη για τους ευρωπαϊκούς Ανοικτούς Εκπαιδευτικούς Πόρους (OER). Σκοπός της είναι να συνδέσει του πόρους ανοικτής εκπαίδευσης από όλα τα κράτη της Ευρωπαϊκής Ένωσης, και να τους διαθέσει – δωρεάν – σε κάθε ενδιαφερόμενο, φοιτητή, ερευνητή και καθηγητή. Σήμερα, φιλοξενεί - επιπλέον - ηλεκτρονικές σειρές μαθημάτων και MOOCs από κορυφαία ευρωπαϊκά ιδρύματα (OpenEducation Europa, 2016).

Περισσότερα από 1.200 πανεπιστήμια σε όλο τον κόσμο έχουν ξεκινήσει δωρεάν διαδικτυακά μαθήματα. Εκτός από τις μεγαλύτερες παγκόσμιες πλατφόρμες MOOC (Coursera, edX, FutureLearn), πολλές εθνικές κυβερνήσεις σε όλο τον κόσμο έχουν ξεκινήσει τις δικές τους πλατφόρμες MOOC ανά χώρα, όπως η Ινδία, η Ιταλία, το Ισραήλ, το Μεξικό και η Ταϊλάνδη.

Μετά από μια δεκαετία εκλαΐκευσης, το 2021, πάνω από 220 εκατομμύρια φοιτητές είχαν εγγραφεί σε τουλάχιστον ένα μάθημα σε μία από αυτές τις πλατφόρμες, και 40 εκατομμύρια μόνο το 2021 (εξαιρουμένης της Κίνας). Τα MOOCs και οι πλατφόρμες MOOC εξακολουθούν να αναπτύσσονται, ακόμη και μετά το τρελό "Έτος του MOOC" που προκλήθηκε από την πανδημία και τους ταξιδιωτικούς περιορισμούς.

Οι πάροχοι MOOC προσπάθησαν να επωφεληθούν από αυτή τη δυναμική επεκτείνοντας τα μαθήματά τους (Gomez et al., 2022)⁴¹. Στους αγγλόφωνους παρόχους MOOC προστίθενται τώρα διάφοροι περιφερειακοί πάροχοι MOOC που εμφανίστηκαν τα τελευταία χρόνια, εστιάζοντας στην ανάπτυξη μαθησιακού υλικού προσαρμοσμένου στις ανάγκες των "τοπικών" χρηστών όσον αφορά τη γλώσσα και το περιεχόμενο. Παραδείγματα είναι ο λατινοαμερικανικός πάροχος MiriadaX, ο κινεζικός XuentangX, ο ιταλικός Federica.EU, ο γερμανικός OpenHPI κ.λπ. Οι σημαντικότεροι από αυτούς, όσον αφορά τον αριθμό των εγγεγραμμένων σπουδαστών, είναι οι Coursera (Πανεπιστήμιο του Στάνφορντ), edXTM (MIT και Χάρβαρντ), UdacityTM (Πανεπιστήμιο του Στάνφορντ) και FutureLearn (Ανοικτό Πανεπιστήμιο) (De Notaris et al., 2021; Li et al., 2022)⁴³. Αρκετές περιφερειακές πρωτοβουλίες MOOC χρησιμοποιούν το πρότυπο Open edX - την πρωτοβουλία ανοικτού κώδικα του edX - ως το λογισμικό για την υλοποίηση MOOCs (Ruiz-Valiente et al., 2020)⁴⁴. Οι Ruiz-Valiente et al. (2022) κατατάσσουν τους παρόχους MOOC σε παγκόσμιο, περιφερειακό και τοπικό επίπεδο με βάση το εύρος των μαθημάτων που προσφέρουν:

- Παγκόσμια εμβέλεια: Πάροχοι όπως οι MITx, HarvardX, FutureLearn, CU Boulder, openSAP, και OpenWHO προσφέρουν μαθήματα κυρίως στα αγγλικά σε παγκόσμιο και διεθνές κοινό.
- Μικτή εμβέλεια: Πάροχοι που προσφέρουν μαθήματα στην τοπική γλώσσα και στα αγγλικά. Ως εκ τούτου, έχουν περιφερειακή εστίαση. Επιτρέπουν σε εκπαιδευόμενους από όλο τον κόσμο να παρακολουθήσουν κάποια από τα μαθήματά τους. Τα μαθήματα

μπορεί να φιλοξενοούνται στην πλατφόρμα τους ή σε συνεργασία με μια παγκόσμια πλατφόρμα όπως το edX ή το Coursera.

- Περιφερειακή εμβέλεια: Οι πάροχοι που διαχειρίζονται την πλατφόρμα MOOC προσφέρουν μαθήματα κυρίως σε μια εθνική γλώσσα. Επικεντρώνονται σε έναν εθνικό, περιφερειακό ή τοπικό πληθυσμό. Στην κατηγορία αυτή περιλαμβάνονται πάροχοι όπως οι UPVx, Edraak και XuetangX.

Ωστόσο, μια προσπάθεια κατηγοριοποίησης των πιο δημοφιλών πλατφορμών MOOC ανά γεωγραφική περιοχή έγινε από τους ίδιους ερευνητές (Shah et al., 2022).

Πλατφόρμες MOOC από τη Βόρεια Αμερική & τις Ηνωμένες Πολιτείες
Coursera / United States
edX / United States
Udacity / United States
Canvas Network/ United States
Kadenze / United States
Stanford Lagunita / United States
Complexity Explorer / United States
MéxicoX / Mexico
Πλατφόρμες MOOC από την Ευρώπη και το Ηνωμένο Βασίλειο
FutureLearn / United Kingdom
France UniversitéNumérique (FUN) / France
Miríadax / Spain
EduOpen / Italy
Federica Web Learning / Italy
European Multiple MOOC Aggregator (EMMA) / Europe
OpenHPI / Germany
MOOC.fi / Finland
Prometheus / Ukraine
Open Education (openedu.ru) / Russia
Πλατφόρμες MOOC από την Ασία (εκτός από τις κινεζικές πλατφόρμες MOOC)
SWAYAM / India
NPTEL / India
JMOOC / Japan
gacco / Japan

OpenLearning / Japan
K-MOOC / Korea
ThaiMOOC / Thailand
Edraak (Arabic) / Jordan
Campus-II / Israel

Πλατφόρμες MOOC κινεζικής γλώσσας
XuetangX / China
Chinese University MOOC / China
Zhihuishu / China
CNMOOC / China
Xue Yin Online / China
Open Education (openedu.tw) / Taiwan
eWant — education you want / Taiwan

Πίνακας 1

3.2.1 edX

Το edX, που δημιουργήθηκε ως κοινοπραξία μεταξύ του Χάρβαρντ και του MIT το 2012 και ήταν μη κερδοσκοπικό. Αλλά το 2021, η edX εξαγοράστηκε από την 2U έναντι 800 εκατ. δολαρίων. Η edX είναι μία από τις μεγαλύτερες πλατφόρμες MOOC στον κόσμο, με περισσότερους από 46 εκατομμύρια μαθητές. Η πλατφόρμα προσφέρει πάνω από 4.000 μαθήματα και μπορεί να υπερηφανεύεται για 160 συνεργαζόμενα ιδρύματα. Καλύπτει δημοφιλή θέματα όπως η επιστήμη των δεδομένων και οι ανθρωπιστικές επιστήμες. Η πλατφόρμα προσφέρει επίσης μαθήματα πληροφορικής σε Python, front-end και full-stack development και κυβερνοασφάλεια.

Επιπλέον, η EdX προσφέρει διάφορα προγράμματα πιστοποιητικών, συμπεριλαμβανομένων 56 MicroMasters και 14 MicroBachelors (τα οποία προσφέρουν μια διαδρομή για την απόκτηση πιστωτικών μονάδων), 52 XSeries και 381 επαγγελματικά πιστοποιητικά. Όπως το Coursera, έτσι και το edX έχει ξεκινήσει πολλαπλά διαδικτυακά προγράμματα σπουδών.

Το edX χρησιμοποιεί ένα σύστημα μάθησης ανοικτού κώδικα που επιτρέπει στους τεχνολόγους και τους εκπαιδευτικούς να εμπλουτίζουν ελεύθερα τα MOOCs τους, προσθέτοντας εργαλεία που υποστηρίζουν τις συγκεκριμένες ανάγκες των φοιτητών τους. Εκτός από τα επαγγελματικά πιστοποιητικά, οι εκπαιδευόμενοι μπορούν να κερδίσουν πανεπιστημιακές μονάδες μέσω των προγραμμάτων MicroBachelors και MicroMasters. Το edX παρέχει επίσης πλήρεις διαδικτυακούς μεταπτυχιακούς τίτλους σπουδών, συμπεριλαμβανομένων των μεταπτυχιακών τίτλων για τη διαχείριση της εφοδιαστικής αλυσίδας, τις επιστήμες διατροφής και το μάρκετινγκ.

3.2.2 Coursera

Το Coursera ιδρύθηκε το 2012 από δύο καθηγητές του Στάνφορντ και είναι ένας από τους μεγαλύτερους παρόχους μαθημάτων μαζικής ανοικτής πρόσβασης στον κόσμο. Η πλατφόρμα

συνεργάζεται με 190 εταιρείες και πανεπιστήμια για την παροχή πλήρως εξ αποστάσεως και αυτορυθμιζόμενων ευκαιριών μάθησης που οδηγούν σε ψηφιακά πιστοποιητικά σε προπτυχιακό, μεταπτυχιακό και μεταπτυχιακό επίπεδο. Το Coursera μπορεί να υπερηφανεύεται για το υψηλό ποσοστό ικανοποίησης, με το 87% των χρηστών να αναφέρουν οφέλη για την καριέρα τους.

Ο κατάλογος περιλαμβάνει πάνω από 3.900 μαθήματα και ειδικότητες που καλύπτουν τομείς όπως οι επιχειρήσεις, η επιστήμη των υπολογιστών, οι φυσικές επιστήμες και η μηχανική, οι τέχνες και οι ανθρωπιστικές επιστήμες και η εκμάθηση γλωσσών. Το Coursera προσφέρει επίσης προσιτά online πτυχία μέσω των συνεργαζόμενων σχολών του. Οι φοιτητές μπορούν να αποκτήσουν μεταπτυχιακό τίτλο σπουδών στη λογιστική από το Πανεπιστήμιο του Ιλινόις ή παγκόσμιο μεταπτυχιακό τίτλο σπουδών στη δημόσια υγεία από το Imperial College του Λονδίνου.

Με πάνω από 118 εκατομμύρια εκπαιδευόμενους και πάνω από 500 εκατομμύρια δολάρια σε χρηματοδότηση, το Coursera είναι η μεγαλύτερη πλατφόρμα MOOC στον κόσμο. Επί του παρόντος, η Coursera προσφέρει πάνω από 10.000 μαθήματα από περισσότερους από 300 συνεργάτες. Πέρα από τα μεμονωμένα μαθήματα, το Coursera προσφέρει τα δικά του πιστοποιητικά, γνωστά ως Specialization. Ο κατάλογος του Coursera απαριθμεί επί του παρόντος περισσότερες από 840 εξειδικεύσεις. Η πλατφόρμα MOOC φιλοξενεί επίσης 26 πλήρως διαδικτυακά μεταπτυχιακά σε τομείς όπως η επιστήμη των υπολογιστών, η επιστήμη των δεδομένων, οι επιχειρήσεις και η δημόσια υγεία. Το Coursera for Business καθιστά τα διαδικτυακά μαθήματα διαθέσιμα σε πελάτες επιχειρήσεων.

3.2.3 Udacity

Το Udacity είναι μια κερδοσκοπική πλατφόρμα MOOC που επικεντρώνεται στην ανάπτυξη καριέρας μέσω τεχνικών και επαγγελματικών διαδικτυακών μαθημάτων. Τα θέματα καλύπτουν έξι τομείς σπουδών, οι οποίοι περιλαμβάνουν την επιστήμη των δεδομένων, το cloud computing, τα αυτόνομα συστήματα και την τεχνητή νοημοσύνη. Οι σπουδαστές μπορούν επίσης να παρακολουθήσουν μαθήματα προγραμματισμού και ανάπτυξης σε C++, Blockchain και Android developer.

Επιπλέον, οι εργαζόμενοι επαγγελματίες μπορούν να ολοκληρώσουν MOOCs που τους βοηθούν να αποκτήσουν δεξιότητες διαχείρισης προϊόντων και ανάλυσης μάρκετινγκ. Πέρα από τις ποικίλες προσφορές μαθημάτων, το Udacity παρέχει ολοκληρωμένες υπηρεσίες σταδιοδρομίας, όπως εξατομικευμένη καθοδήγηση για την εύρεση εργασίας, καθοδήγηση για τη σύνταξη βιογραφικού σημειώματος και βέλτιστες πρακτικές στο LinkedIn. Το πρόγραμμα Udacity Talent Program επιτρέπει στους χρήστες να δημιουργήσουν λεπτομερή προφίλ και να συνδεθούν με σημαντικούς εργοδότες όπως η Google και η Mercedes-Benz.

Το Udacity έχει απομακρυνθεί από την προηγούμενη ταυτότητά του ως πλατφόρμα MOOC. Ήταν η πρώτη από τις αρχικές πλατφόρμες MOOC που έφτασε σε καθεστώς μονόκερου (> 1 δισεκατομμύριο δολάρια αποτίμηση) και συνεργάζεται με εταιρείες τεχνολογίας για τη δημιουργία Nanodegrees που εκπαιδεύουν τους φοιτητές για θέσεις εργασίας με επίκεντρο την τεχνολογία. Τα Nanodegrees κοστίζουν περίπου 1.000 δολάρια και η ολοκλήρωσή τους μπορεί να διαρκέσει μερικούς μήνες. Παρόλο που το Udacity έχει σταματήσει σε μεγάλο βαθμό να εκδίδει νέα δωρεάν μαθήματα, εξακολουθεί να έχει στον κατάλογό του ορισμένα μαθήματα που είναι δωρεάν για έλεγχο.

3.2.4 Udemy

Ως ο μεγαλύτερος πάροχος διαδικτυακής μάθησης, το Udemy προσφέρει πάνω από 150.000 μαθήματα σε 65 γλώσσες. Ενώ οι σπουδαστές μπορούν να παρακολουθήσουν μια ποικιλία δωρεάν μαθημάτων, πολλά MOOCs απαιτούν χρέωση. Πληρώνοντας για premium περιεχόμενο, οι χρήστες αποκτούν επίσης πρόσβαση σε λειτουργίες όπως η άμεση ανταλλαγή μηνυμάτων, οι ερωτήσεις και απαντήσεις και τα πιστοποιητικά ολοκλήρωσης.

Τα θέματα καλύπτουν 11 ευρείες κατηγορίες, όπως παραγωγικότητα γραφείου, υγεία και γυμναστική και φωτογραφία. Οι σπουδαστές μπορούν επίσης να ολοκληρώσουν μαθήματα χρηματοοικονομικής και λογιστικής, να μάθουν τα βασικά στοιχεία του Bitcoin και του blockchain ή να αναπτύξουν δεξιότητες ανάλυσης της παγκόσμιας αγοράς. Επιπλέον, το Udemy παρέχει περιεχόμενο προσωπικής ανάπτυξης που επιτρέπει στους χρήστες να διαχειρίζονται το άγχος, να βελτιώνουν την αυτοεκτίμησή τους και να καλλιεργούν ουσιαστικές σχέσεις.

3.2.5 FutureLearn / United Kingdom

Το FutureLearn είναι μια πλατφόρμα MOOC με έδρα το Ηνωμένο Βασίλειο και 19 εκατομμύρια χρήστες. Το Ανοικτό Πανεπιστήμιο εγκαινίασε το FutureLearn στα τέλη του 2012. Αργότερα έγινε κοινή ιδιοκτησία του Ανοικτού Πανεπιστημίου και του ομίλου SEEK Group με έδρα την Αυστραλία. Και πιο πρόσφατα, εξαγοράστηκε από την GUS μετά από μια ταραχώδη χρονιά. Οι περισσότεροι πανεπιστημιακοί εταίροι του βρίσκονται στο Ηνωμένο Βασίλειο και την Ευρώπη, αλλά έχει επίσης μερικούς πανεπιστημιακούς εταίρους σε άλλες χώρες, συμπεριλαμβανομένων των Ηνωμένων Πολιτειών, της Αυστραλίας και της Νότιας Κορέας.

Η FutureLearn προσφέρει το δικό της πρόγραμμα πιστοποιήσεων, το οποίο είναι γνωστό ως FutureLearnPrograms. Το 2021, η FutureLearn εγκαινίασε έναν νέο τύπο μικροπιστοποιητικών, το ExpertTrack.

3.2.6 FranceUniversité Numérique (FUN) / France

Το FUN είναι η επίσημη πλατφόρμα MOOC της Γαλλίας. Το γαλλικό Υπουργείο Ανώτατης Εκπαίδευσης ξεκίνησε το έργο τον Ιούλιο του 2013. Το FUN έχει περισσότερους από 100 εταίρους μεταξύ των ιδρυμάτων τριτοβάθμιας εκπαίδευσης που δημιουργούν MOOCs, συμπεριλαμβανομένων τριών πανεπιστημίων στο Βέλγιο, ενός στην Ελβετία και δύο στην Τυνησία. Το FUN διαθέτει περισσότερα από 600 MOOCs και έχει περισσότερους από δύο εκατομμύρια εγγεγραμμένους φοιτητές.

3.2.7 European Multiple MOOC Aggregator

Το European Multiple MOOC Aggregator, που ονομάζεται εν συντομία EMMA, είναι μια πιλοτική δράση διάρκειας 30 μηνών που υποστηρίζεται από την Ευρωπαϊκή Ένωση. Στόχος της είναι να αναδείξει την αριστεία σε καινοτόμες διδακτικές μεθοδολογίες και μαθησιακές προσεγγίσεις μέσω της πιλοτικής εφαρμογής MOOCs μεγάλης κλίμακας σε διάφορα γνωστικά αντικείμενα. Το EMMA παρέχει ένα σύστημα για την παροχή δωρεάν, ανοικτών, διαδικτυακών μαθημάτων σε πολλές γλώσσες από διάφορα ευρωπαϊκά πανεπιστήμια, ώστε να συμβάλει στη διατήρηση της πλούσιας

πολιτιστικής, εκπαιδευτικής και γλωσσικής κληρονομιάς της Ευρώπης και να προωθήσει την πραγματική διαπολιτισμική και πολύγλωσση μάθηση.

3.2.8 K-MOOC

Το K-MOOC, ή Κορεατικό MOOC, είναι η επίσημη πλατφόρμα MOOC της Κορέας. Η πλατφόρμα MOOC ξεκίνησε με πρωτοβουλία του Υπουργείου Παιδείας και λειτούργησε για πρώτη φορά το 2015. Επί του παρόντος, απαριθμεί πάνω από 2000 μαθήματα από 170 εταίρους. Πάνω από 3 εκατομμύρια εκπαιδευόμενοι έχουν εγγραφεί στην πλατφόρμα MOOC.

3.2.9 XuetangX / China

Το XuetangX είναι η πρώτη και μεγαλύτερη πλατφόρμα MOOC της Κίνας. Ιδρύθηκε το 2013 από το Πανεπιστήμιο Tsinghua υπό την εποπτεία του Υπουργείου Παιδείας Έρευνας της Κίνας. Είναι ίσως η ταχύτερα αναπτυσσόμενη πλατφόρμα MOOC. Όταν η ClassCentral πήρε συνέντευξη από τον πρόεδρο του διοικητικού συμβουλίου του XuetangX τον Οκτώβριο του 2016, η πλατφόρμα MOOC είχε πέντε εκατομμύρια χρήστες. Μέχρι το τέλος των 8ων γενεθλίων της στις 10 Οκτωβρίου 2021, η XuetangX είχε ξεπεράσει τα 80 εκατομμύρια εγγεγραμμένους χρήστες.

Η πλατφόρμα MOOC χτίστηκε αρχικά πάνω σε μια έντονα προσαρμοσμένη έκδοση του OpenedX, αλλά απομακρύνθηκε από αυτό, επανασχεδίασε την πλατφόρμα και έγινε διεθνής στις αρχές του 2020. Στην πλατφόρμα MOOC προσφέρονται πάνω από 7000 δωρεάν μαθήματα, bootcamps και προγράμματα πιστοποίησης.

Το XuetangX διαθέτει επίσης ένα προϊόν συστήματος διαχείρισης μάθησης (LMS) στο cloud, το οποίο χρησιμοποιείται από πανεπιστήμια σε όλη την Κίνα και το οποίο είχαν χρησιμοποιήσει περισσότεροι από 48 εκατομμύρια Κινέζοι φοιτητές μέχρι το τέλος του 2020.

3.2.10 ZhihuiShu

Το ZhihuiShu, που μεταφράζεται ως "Δέντρο Σοφίας", ξεκίνησε τον Δεκέμβριο του 2012. Η ιδιαιτερότητά του είναι ότι επιτρέπει στους πανεπιστημιακούς εταίρους του να μοιράζονται μεταξύ τους μαθήματα με πιστωτικές μονάδες, διευκολύνοντας την ακαδημαϊκή κινητικότητα σε όλη την Κίνα.

Από τα 14.100 μαθήματα που προσφέρονται στην πλατφόρμα, τα 10.500 (δηλαδή το 75%) είναι μαθήματα με πιστωτικές μονάδες μόνο για φοιτητές εντός της πανεπιστημιούπολης. Η πλατφόρμα προσφέρει 45 μαθήματα με δωρεάν ψηφιακά πιστοποιητικά. Και υπάρχουν επίσης διαθέσιμα 18 προγράμματα μικροπιστοποιητικών.

3.3 Τα MOOCs στην Ελλάδα

Όπως έχουμε αναφέρει τα MOOCS (Massive Online Open Courses), ή αλλιώς «Μαζικά Ανοιχτά Διαδικτυακά Μαθήματα», η πρόσβαση είναι ελεύθερη, χωρίς περιορισμούς λόγω τόπου διαμονής, ηλικίας, ή οικονομικής κατάστασης. Κατά συνέπεια έχουμε πρόσβαση στις πλατφόρμες που αναφέραμε από οπουδήποτε άρα και από τη χώρα μας. Παρόλα αυτά έχουν γίνει αντίστοιχες προσπάθειες και σε μας. Οι βασικές ελληνικές πλατφόρμες παροχής MOOCs είναι η Opencourses, η Coursity και η Mathesis ενώ τελευταία έχει δημιουργηθεί και η mooc.edu.gr κυρίως για εκπαιδευτικούς αλλά και κάθε ενδιαφερόμενο.

3.3.1 mooc.edu.gr

Σύμφωνα με την ιστοσελίδα τους, η πλατφόρμα mooc.edu.gr είναι η επίσημη διαδικτυακή πλατφόρμα του Υπουργείου Παιδείας και Θρησκευμάτων (ΥΠΑΙΘ) για τη φιλοξενία και διάθεση Μαζικών Ανοιχτών Διαδικτυακών Μαθημάτων (MOOCs – Massive Online Open Courses) σε εκπαιδευτικούς της Πρωτοβάθμιας και Δευτεροβάθμιας Εκπαίδευσης, αλλά και κάθε ενδιαφερόμενο/η.

Η πλατφόρμα mooc.edu.gr σχεδιάστηκε, αναπτύχθηκε και υποστηρίζεται από το Ινστιτούτο Τεχνολογίας Υπολογιστών και Εκδόσεων «ΔΙΟΦΑΝΤΟΣ» (ITYE) και, ειδικότερα, από τη Διεύθυνση Στρατηγικής και Ψηφιακού Εκπαιδευτικού Υλικού (ΔΙΣ). Είναι βασισμένη στην πλατφόρμα ανοιχτού κώδικα Open edX (<https://open.edx.org>). Φιλοξενείται σε υποδομή του Εθνικού Δικτύου Υποδομών Τεχνολογίας και Έρευνας Α.Ε. (ΕΔΥΤΕ).

Η πλατφόρμα mooc.edu.gr αναπτύχθηκε αρχικά για να καλύψει την ανάγκη για ανάρτηση και διάθεση στην εκπαιδευτική κοινότητα των MOOCs που προβλέπονταν να αναπτυχθούν από το European SchoolNet (EUN) (<http://www.eun.org>), σε συνεργασία με το Υπουργείο Παιδείας και Θρησκευμάτων, στο πλαίσιο του έργου SRSS II.

Στη συνέχεια, η πλατφόρμα mooc.edu.gr θα συνεχίσει να καλύπτει επιμορφωτικές ανάγκες εκπαιδευτικών και κάθε ενδιαφερόμενου/ης, με την ανάρτηση και διάθεση στην εκπαιδευτική κοινότητα -και ευρύτερα στην κοινωνία- μίας ευρείας γκάμας MOOCs. Αυτά θα αναπτύσσονται υπό την εποπτεία του ΥΠΑΙΘ, σε συνεργασία με το ΙΕΠ, το οποίο θα έχει τον συντονισμό και την παρακολούθηση, καθώς και την επιστημονική και παιδαγωγική υποστήριξη των επιμορφωτικών προγραμμάτων, και το ΙΤΥΕ, το οποίο θα έχει την ευθύνη της λειτουργίας της πλατφόρμας και την τεχνική υποστήριξη της εφαρμογής των επιμορφωτικών προγραμμάτων.

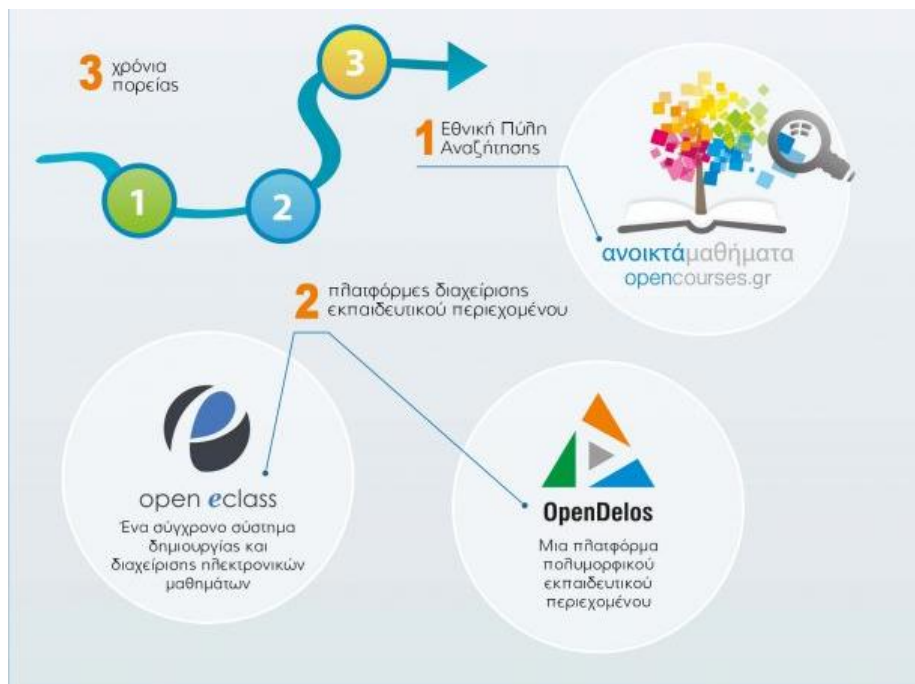
Τα Ανοιχτά Διαδικτυακά Μαθήματα (MOOCs) της πλατφόρμας mooc.edu.gr θα προσφέρονται δωρεάν σε όλους. Θα αφορούν σε μια ευρεία γκάμα θεμάτων. Θα δίνουν έμφαση στην ανάπτυξη και καλλιέργεια δεξιοτήτων, σε καινοτόμες παιδαγωγικές έννοιες, στρατηγικές, εργαλεία για τη διδασκαλία και τη μάθηση, καθώς και πολλά άλλα. Μέσα από την πλατφόρμα mooc.edu.gr ενθαρρύνεται η ανταλλαγή απόψεων και προτάσεων και υποστηρίζεται η αλληλεπίδραση με μια μεγάλη κοινότητα εκπαιδευτικών.

Εκτός του ότι προσφέρονται δωρεάν – με εξαίρεση συγκεκριμένες περιπτώσεις, όπου σε δεύτερο χρόνο ενδέχεται να υπάρξει κάποιο κόστος - τα οφέλη των MOOCS είναι τα εξής:

3.3.2 Opencourses – (Ιστοσελίδα: Opencourses.gr)

Η πλατφόρμα Ανοικτά Ακαδημαϊκά Μαθήματα (Opencourses.gr), σύμφωνα με τις πληροφορίες που έχει αναρτήσει στην επίσημη ιστοσελίδα της παρέχει μαθήματα προπτυχιακού ή μεταπτυχιακού προγράμματος σπουδών, τα οποία διδάσκονται σε ελληνικά Πανεπιστήμια και ΑΤΕΙ και έχουν διαμορφωθεί ψηφιακά, ώστε να διατίθενται στο Διαδίκτυο με άδειες ανοικτής χρήσης πνευματικών δικαιωμάτων. Τα συγκεκριμένα μαθήματα είναι δωρεάν στο σύνολο τους και δεν δίνεται η δυνατότητα χορήγησης πιστοποιητικών ολοκλήρωσης.

Ο στόχος της πλατφόρμας, όπως διατυπώνεται από τους διαχειριστές της, είναι: «η καλλιέργεια και η ενίσχυση της κουλτούρας αξιοποίησης των νέων δυνατοτήτων που παρέχουν οι Τεχνολογίες Πληροφορικής και Επικοινωνιών (ΤΠΕ) για την προσφορά ψηφιακού εκπαιδευτικού περιεχομένου υψηλής



ποιότητας – το οποίο, βασισμένο σε διεθνείς πρακτικές και πρότυπα, δύναται να αναβαθμίσει την εκπαιδευτική διαδικασία, να αναδείξει τον κοινωνικό ρόλο των ΑΕΙ και να προβάλλει διεθνώς το παραγόμενο διδακτικό έργο τους».

Τα μαθήματα μπορούν να αξιοποιηθούν από φοιτητές/ές για επανάληψη ή προετοιμασία για το δια ζώσης μάθημα της σχολής τους, από μη φοιτητές/ριες και οποιοδήποτε άτομο θέλει να διευρύνει τις γνώσεις του και από μαθητές/ριες, ώστε να βοηθηθούν για την επιλογή σχολής και επαγγελματικού προσανατολισμού. Στα στοιχεία που παρέχονται στην πλατφόρμα, φαίνεται πως παρέχονται 4.234 μαθήματα, τα οποία προέρχονται από 26 ακαδημαϊκά ιδρύματα.

3.3.3 Coursity – (Ιστοσελίδα: Coursity.gr)

Στην ιστοσελίδα του coursity διαβάζουμε είναι μία καινοτόμα πλατφόρμα, που φιλοξενεί σεμινάρια τα οποία πραγματοποιούνται διαδικτυακά με τη μορφή βιντεοσκοπημένων μαθημάτων. Τα διαδικτυακά σεμινάρια καλύπτουν ένα μεγάλο εύρος επιστημονικών πεδίων και γνωστικών αντικειμένων. Η διάρκεια των περισσότερων σεμιναρίων είναι 4-8 εβδομάδες. Παράλληλα,

προσφέρονται σεμινάρια ετήσιας διάρκειας. Οι επιστημονικά υπεύθυνοι και οι διδάσκοντες είναι αναγνωρισμένοι καθηγητές ελληνικών Πανεπιστημίων.

Η επιτυχής παρακολούθηση των σεμιναρίων οδηγεί στη λήψη Πιστοποιητικού Επιμόρφωσης από τα Κέντρα Επιμόρφωσης και Δια Βίου Μάθησης Ελληνικών Πανεπιστημίων. Το Coursity δίνει την ευκαιρία σε όλους για πρόσβαση σε υψηλής ποιότητας μάθηση. Προσφέρει τη δυνατότητα της δωρεάν παρακολούθησης πολλών μαθημάτων.

Το Coursity προσφέρει τα σεμινάρια με έναν καινοτόμο τρόπο:

- Με μαγνητοσκοπημένες βιντεοδιαλέξεις και όχι με παραδοσιακές σημειώσεις, ώστε να αποφεύγεται ο αποπροσανατολισμός.
- Με διαδραστικό υλικό ασκήσεων και εργασιών, ώστε να παρέχονται κίνητρα για μάθηση και επιτυχή ολοκλήρωση.
- Σε ένα φιλικό περιβάλλον που στηρίζεται στις πιο καινοτόμες και εξελιγμένες τεχνολογίες μάθησης.
- Με μικρό και προσιτό κόστος, με στόχο να δώσει την ευκαιρία σε περισσότερους ανθρώπους για πρόσβαση σε υψηλής ποιότητας μάθηση.
- Με λειτουργία φόρουμ, για αλληλεπίδραση μεταξύ των συμμετεχόντων και των διδασκόντων και τη δημιουργία ενεργών κοινοτήτων μάθησης.
- Με παρακολούθηση του σεμιναρίου αποκλειστικά από απόσταση, από οποιοδήποτε χώρο επιθυμεί ο συμμετέχων, σε οποιοδήποτε χρόνο, σύμφωνα με τις απαιτήσεις του. Οι βιντεοδιαλέξεις, και το υλικό των ασκήσεων και εργασιών, καθώς και το φόρουμ, είναι διαθέσιμα 24 ώρες το εικοσιτετράωρο.
- Με συνεχή και άμεση εκπαιδευτική και τεχνική υποστήριξη.

Στο τέλος κάθε ενότητας/εβδομάδας ο εκπαιδευόμενος χρειάζεται να έχει ολοκληρώσει ορισμένες ασκήσεις (ερωτήσεις κλειστού τύπου ή σύντομης απάντησης) και να έχει εκπονήσει μια σύντομη δραστηριότητα (mini-project). Με την ολοκλήρωση όλων των ενοτήτων, δηλαδή στο τέλος του σεμιναρίου, ο εκπαιδευόμενος χρειάζεται ακόμη να εκπονήσει την τελική εργασία (project).

Τα σεμινάρια απευθύνονται σε εκπαιδευτικούς πρωτοβάθμιας και δευτεροβάθμιας εκπαίδευσης, σε Προπτυχιακούς φοιτητές, μεταπτυχιακούς φοιτητές και υποψηφίους διδάκτορες, σε Επαγγελματίες και σε όλους όσους θέλουν να εμπλουτίσουν τις γνώσεις τους και να έχουν πρόσβαση στη μάθηση, μέσω σεμιναρίων υψηλής ποιότητας, που πραγματοποιούνται με μία καινοτόμα διαδικασία σε πλατφόρμα τελευταίας τεχνολογίας.

Στην Coursity υπάρχουν δεκάδες μαθήματα, τα οποία παρέχονται κυρίως από το Πανεπιστήμιο Ιωαννίνων και το Ελληνικό Ανοιχτό Πανεπιστήμιο (Ε.Α.Π.).

3.3.4 Mathesis

Το Κέντρο Ανοικτών Διαδικτυακών Μαθημάτων *Mathesis* (Mathesis.cup.gr), ιδρύθηκε το 2015 —με πρωτοβουλία και κύρια ευθύνη του Στέφανου Τραχανά— ως ένα ιδιαίτερο τμήμα των Πανεπιστημιακών Εκδόσεων Κρήτης (ΠΕΚ) με αποκλειστικό σκοπό τη δημιουργία και δωρεάν προσφορά στους φοιτητές, τους επαγγελματίες επιστήμονες και το ευρύτερο κοινό, διαδικτυακών μαθημάτων στο επίπεδο των καλύτερων διεθνών προτύπων.

Σύμφωνα με τη ιστοσελίδα mathesis.cup.gr η χώρα μας δεν επιτρέπεται να μείνει έξω από τις επαναστατικές αλλαγές που συντελούνται παγκοσμίως σε αυτό το θέμα και κρίναμε ότι ήταν υποχρέωσή μας, ως μη κερδοσκοπικού οργανισμού που υπηρετεί δημόσιους σκοπούς και αξίες, να δημιουργήσουμε μέσα στις ΠΕΚ μια εθνική εστία διαδικτυακών μαθημάτων που θα παρακολουθεί τις διεθνείς εξελίξεις και θα προσπαθεί να συμβαδίζει με αυτές. Τα διαδικτυακά μαθήματα δεν είναι παρά μια νέα εκδοχή πανεπιστημιακών συγγραμμάτων — βιβλιομαθήματα, είναι ο δικός μας όρος γι' αυτά— τα οποία, επομένως, απαιτούν έναν ειδικευμένο φορέα με ακόμα υψηλότερο επίπεδο επαγγελματισμού και εμπειρογνωμοσύνης από ό,τι στο παραδοσιακό έντυπο βιβλίο. Έτσι, το Mathesis προέκυψε φυσιολογικά ως η επέκταση των ΠΕΚ στον χώρο των Μαζικών Ανοικτών Διαδικτυακών Μαθημάτων (ΜΑΔΙΜ ή MOOCs = Massive Open Online Courses).

Ένα ιδιαίτερο χαρακτηριστικό των διαδικτυακών μαθημάτων είναι ότι η ποιότητά τους δεν είναι μόνο ζήτημα περιεχομένου και διδακτικής ικανότητας του δημιουργού τους. Εξαρτάται καίρια και από το τεχνολογικό επίπεδο της πλατφόρμας που τα διαχειρίζεται. Στην πραγματικότητα η πλατφόρμα καθ' εαυτή —προϊόν και η ίδια μιας γόνιμης (και διαρκούς) αλληλεπίδρασης δημιουργών περιεχομένου και τεχνικών πληροφορικής— αποτελεί μια «εργαλειοθήκη» εκπαιδευτικών τεχνικών και εφαρμογών που είναι πέρα από τις δυνατότητες κάθε μεμονωμένου παρόχου διαδικτυακών μαθημάτων να τις αναπτύξει.

Όμως μια πλατφόρμα με την τεχνολογία, τη λειτουργικότητα και την αισθητική του OpenEdX θέτει από μόνη της υψηλές απαιτήσεις εκπαιδευτικής ποιότητας από τα μαθήματα που φιλοξενεί. Έτσι, η πρόκληση για το Mathesis είναι να εξασφαλίσει —επιλέγοντας τους κατάλληλους διδάσκοντες— ότι η ποιότητα των «βιβλιομαθημάτων» του δεν θα είναι κατώτερη του «περιβάλλοντος» που τα φιλοξενεί.

3.3.5 Ψηφιακή Ακαδημία Πολιτών - (Ιστοσελίδα: Nationaldigitalacademy.gov.gr)

Η πλατφόρμα «Ψηφιακή Ακαδημία Πολιτών» έχει δημιουργηθεί από το υπουργείο ψηφιακής διακυβέρνησης της Ελλάδας και παρέχει ένα ευρύ φάσμα δωρεάν ανοικτών μαθημάτων, τα οποία στοχεύουν κυρίως στην ενίσχυση των ψηφιακών ικανοτήτων των πολιτών και τη βελτίωση του ψηφιακού γραμματισμού. Στην πλατφόρμα υπάρχει και το ευρωπαϊκό εργαλείο αυτοαξιολόγησης των ψηφιακών ικανοτήτων, καθώς και η δυνατότητα εξατομικευμένων προτάσεων παρακολούθησης μαθημάτων σύμφωνα με τις υπάρχουσες γνώσεις και ανάγκες του/της κάθε ενδιαφερόμενου/ης.

3.4 Η τεχνητή νοημοσύνη στις πλατφόρμες MOOC

Στις πλατφόρμες MOOC υπάρχει ευρεία γκάμα μαθημάτων AI που διδάσκονται από ειδικούς στον τομέα, καλύπτοντας θέματα όπως η μηχανική μάθηση, η βαθιά μάθηση, η επεξεργασία φυσικής γλώσσας και άλλα. Οι εκπαιδευόμενοι μπορούν να επιλέξουν μαθήματα σύμφωνα με τους στόχους τους, το επίπεδο των γνώσεών τους. Οι δεξιότητες που θα αποκτήσουν είναι πολύτιμες στην αγορά εργασίας.

Με συστηματική αναζήτηση διαπιστώσαμε ότι υπάρχουν πολλά μαθήματα Τεχνητής Νοημοσύνης (TN) που προσφέρονται παγκοσμίως, από διαδικτυακές πλατφόρμες. Χρησιμοποιώντας τη λειτουργία αναζήτησης στις πλατφόρμες που προαναφέραμε, με λέξεις-κλειδιά όπως Artificial Intelligence, AI, Machine Learning και Deep Learning πήραμε μια λίστα μαθημάτων και προγραμμάτων που προσφέρονται. Τα μαθήματα μπορεί να προέρχονται και από κορυφαία πανεπιστήμια και εταιρείες τεχνολογίας, όπως το MIT, το Harvard, το Columbia και την IBM, και καλύπτουν από εισαγωγικά θέματα μέχρι προχωρημένες εφαρμογές. Στον παρακάτω πίνακα συγκεντρώσαμε τους αριθμούς μαθημάτων που παρέχονται καθώς και τον αριθμό μαθημάτων σχετικά με τεχνητή νοημοσύνη.

Ο αριθμός των μαθημάτων γενικά αλλά και συγκεκριμένα μαθημάτων τεχνητής νοημοσύνης είναι δύσκολο να προσδιοριστεί με ακρίβεια, καθώς ο αριθμός μπορεί να αλλάζει διαρκώς. Ο παρακάτω πίνακας αποτυπώνει την εικόνα όπως αυτή παρουσιάζεται τον Δεκέμβριο του 2023.

Πλατφόρμα	Συνολικός Αριθμός μαθημάτων	Τεχνητή νοημοσύνη Μηχανική μάθηση
Edx	4.422	1500
Coursera	7000+	2000
Udacity	350	86
Udemy	210.000	1700
CanvasNetwork	613	0
Kadenze	100	4
futurelearn	3671	19
FranceUniversitéNumérique FUN	703	6
Miríadax	687	2
edupen	401	0
Federica Web Learning	480	6
European Multiple MOOC Aggregator called EMMA	67	0
openHPI	159	25
Mooc.fi	23	2
Prometheus/ukraine	300	4
Open Education/ Russia	1272	5
SWAYAM / India	2748	10
K-Mooc Korea	2429	37

thaiMOOC/ Thailand	707	5
Edraak / Jordan	104	2
XuetangX / China	8861	34
Chinese University MOOC	1000	35

Πίνακας 2

Αυτή η λίστα περιλαμβάνει τόσο μεμονωμένα μαθήματα όσο και προγράμματα ειδίκευσης που περιλαμβάνουν πολλαπλά μαθήματα και συνήθως ολοκληρώνονται με ένα τελικό έργο ή πιστοποιητικό.

Από τα πιο δημοφιλή μαθήματα σύμφωνα με τις εγγραφές σε αυτά αλλά και σχετικά με τα αντικείμενα που διαπραγματευτήκαμε στο δικό μας μάθημα επιγραμματικά μπορούμε να αναφέρουμε τα εξής:

edX	• «Artificial Intelligenc» by Columbia University • «Machine Learning» by Columbia University • «Deep Learning» by IBM • «Introduction to Artificial Intelligence (AI)» by Microsoft • «Applied AI with Deep Learning» by IBM
Coursera	• «Machine Learning» by Stanford University • «AI For Everyone» by DeepLearning.AI • «Deep Learning Specialization» by DeepLearning.AI • «IBM AI Engineering Professional Certificate» • «AI for Medicine Specialization» by DeepLearning.AI • «Natural Language Processing Specialization» by DeepLearning.AI
Udacity	• «Artificial Intelligence Nanodegree Program» : Αυτό το ολοκληρωμένο πρόγραμμα περιλαμβάνει ενότητες σχετικά με τη μηχανική μάθηση, την πιθανολογική συλλογιστική, την όραση υπολογιστών και την επεξεργασία φυσικής γλώσσας. • «AI for Business Leaders» : Αυτό το μάθημα απευθύνεται σε επαγγελματίες που επιθυμούν να ενσωματώσουν την ΤΝ στις επιχειρηματικές τους στρατηγικές. Επικεντρώνεται στην κατανόηση των εννοιών και των

	εφαρμογών της ΤΝ στο επιχειρηματικό πλαίσιο • «Εισαγωγή στο TensorFlow για τεχνητή νοημοσύνη, μηχανική μάθηση και βαθιά μάθηση»: TensorFlow, ένα δημοφιλές πλαίσιο για μηχανική μάθηση και βαθιά μάθηση.
--	---

Πίνακας 3

3.5 Τεχνητή νοημοσύνη στις Ελληνικές πλατφόρμες.

Τα μαθήματα τεχνητής νοημοσύνης που παρέχουν οι ελληνικές πλατφόρμες MOOC είναι πολύ περιορισμένα. Αναζητήσαμε μαθήματα σε όλες τις πλατφόρμες που αναφέραμε παραπάνω και παρουσιάζουμε τα αποτελέσματα αναλυτικά. Ορισμένα από αυτά είναι μαθήματα προπτυχιακού επιπέδου που διδάσκονται σε κάποιο εξάμηνο σε τμήματα Πανεπιστημίου και έχουν ανεβεί και στην πλατφόρμα μαζικών ανοικτών διαδικτυακών μαθημάτων.

Mathesis.cup.gr	1. Επιστήμη Δεδομένων και Εφαρμοσμένη Μηχανική Μάθηση με Python (Ε)
Opencourses	1. Opencourse Τεχνητή Νοημοσύνη (Μηχανικών Πληροφορικής και Τηλεπικοινωνιών, Πανεπιστήμιο Δυτικής Μακεδονίας, Κωνσταντίνος Στεργίου – Προπτυχιακό) [Παρουσιάσεις σε PDF] 2. Τεχνητή Νοημοσύνη (Μηχανικών Πληροφορικής, ΤΕΙ Αν. Μακεδονίας και Θράκης, Βασίλειος Καμπουρλάζος – Προπτυχιακό) [Δεν λειτουργεί ο δικτυακός τόπος] 3. Τεχνητή Νοημοσύνη (Εφαρμοσμένης Πληροφορικής, Πανεπιστήμιο Μακεδονίας, Ιωάννης Ρεφανίδης – Προπτυχιακό) [βίντεο διαφάνειες ασκήσεις] 4. Τεχνητή Νοημοσύνη (Κάτια-Λήδα Κερμανίδου Προπτυχιακό Τμήμα Πληροφορικής, Ιόνιο Πανεπιστήμιο) [Διαλέξεις (δεν δουλεύουν) Παρουσιάσεις] 5. Τεχνητή Νοημοσύνη (ΙΩΑΝΝΗΣ ΧΑΤΖΗΛΥΓΕΡΟΥΔΗΣ - Προπτυχιακό - (Α-) ΜΗΧΑΝΙΚΩΝ Η/Υ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ, Πανεπιστήμιο Πατρών) [κλειδωμένο]

	6. Τεχνητή Νοημοσύνη Ι (Νίκος Φακωτάκης, Κυριάκος Σγάρμπας, Κώστας Μουστάκας, Παύλος Πέππας - Προπτυχιακό Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών, Πανεπιστήμιο Πατρών [Παρουσιάσεις βιντεοδιαλέξεις..] 7. Τεχνητή Νοημοσύνη ΙΙ (Νίκος Φακωτάκης, - Προπτυχιακό Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών, Πανεπιστήμιο Πατρών [Παρουσιάσεις βιντεοδιαλέξεις..]
--	---

Πίνακας 4

3.6 Το MOOC που δημιουργήσαμε

Το μάθημα που δημιουργήσαμε, αποτελείται από τέσσερις ενότητες κάθε μία από τις οποίες διαθέτει έξι μαθήματα. Το κάθε μάθημα αποτελείται από σημειώσεις διάλεξης, παρουσίαση, ερωτήσεις – απαντήσεις, περιπτώσεις χρήσης, ασκήσεις πρακτικής και ερωτήσεις πολλαπλών επιλογών. Στο δεύτερο μέρος της διπλωματικής εργασίας έχουν αναπτυχθεί αναλυτικά οι σημειώσεις διάλεξης και οι περιπτώσεις χρήσεις. Με εργαλείο το moodle, ένα πακέτο λογισμικού για τη διεξαγωγή ηλεκτρονικών μαθημάτων μέσω Διαδικτύου, που προσφέρει ολοκληρωμένες υπηρεσίες ασύγχρονης τηλεκπαίδευσης, όλο το υλικό υπάρχει σε δικτυακό τόπο του Πανεπιστημίου.

3.6.1 Γιατί για Επαγγελματίες ΤΠΕ

Με την Πρόβλεψη Δεξιοτήτων του Cedefop μπορούμε να φανταστούμε πώς θα είναι ο κόσμος της εργασίας σε 10 χρόνια από τώρα προβλέποντας τις μελλοντικές τάσεις της απασχόλησης.

Ο τομέας της Πληροφορίας και της Επικοινωνίας (ΤΠΕ) είναι ένας από τους πιο ταχέως εξελισσόμενους τομείς στην ΕΕ. Οι επαγγελματίες ΤΠΕ, που εργάζονται κυρίως στον τομέα των ΤΠΕ, αντιπροσώπευαν περίπου το 2% της συνολικής απασχόλησης στην ΕΕ το 2022. Οι επαγγελματίες ΤΠΕ αναπτύσσουν και βελτιώνουν συστήματα τεχνολογίας πληροφοριών, υλικό, λογισμικό και ψηφιακές βάσεις δεδομένων, παρέχουν συμβουλές σε άλλους επαγγελματίες σχετικά με τον τύπο εφαρμογών ΤΠΕ που είναι πιο κατάλληλες για τους σκοπούς τους και υποστηρίζουν τις επιχειρήσεις για να διασφαλίσουν την ακεραιότητα και την ασφάλεια των δεδομένων. Οι θέσεις εργασίας σε αυτήν την ομάδα περιλαμβάνουν αναλυτές συστημάτων, προγραμματιστές λογισμικού, ιστού και πολυμέσων, προγραμματιστές εφαρμογών, διαχειριστές βάσεων δεδομένων και ειδικούς δικτύων υπολογιστών. Αυτοί οι εργαζόμενοι είναι σημαντικοί για τον ψηφιακό

μετασχηματισμό που λαμβάνει χώρα αυτή τη στιγμή σε αρκετούς ευρωπαϊκούς οικονομικούς τομείς.

Βασικά στοιχεία

- Περίπου 4,6 εκατομμύρια άνθρωποι απασχολήθηκαν ως επαγγελματίες ΤΠΕ το 2022.
- Μεταξύ 2012 και 2022 η απασχόληση για επαγγελματίες ΤΠΕ αυξήθηκε κατά 90%.
- Μεταξύ 2019 και 2020 και παρά το οικονομικό lockdown που σημειώθηκε σε ολόκληρη την ΕΕ, δημιουργήθηκαν σχεδόν 300 χιλιάδες νέες θέσεις εργασίας για επαγγελματίες ΤΠΕ. Η απασχόληση αυξήθηκε περαιτέρω το επόμενο έτος και μέχρι το τέλος του 2022 ήταν 1 εκατομμύριο εργαζόμενοι πάνω από το επίπεδο του 2019.
- Οι περισσότεροι επαγγελματίες ΤΠΕ – 54% το 2021 – απασχολούνται στον τομέα των ΤΠΕ.
- Οι περισσότεροι επαγγελματίες ΤΠΕ (80%) έχουν επιτύχει επίπεδο προσόντων ISCED 5 ή μεγαλύτερο το 2021, ισοδύναμο με το πρώτο στάδιο της τριτοβάθμιας εκπαίδευσης. Το επίπεδο προσόντων του επαγγέλματος δεν αναμένεται να αλλάξει κατά την περίοδο έως το 2035.
- Οι επαγγελματίες ΤΠΕ είναι κυρίως άνδρες (82% το 2021).
- Η απασχόληση επαγγελματιών ΤΠΕ προβλέπεται να αυξηθεί κατά 30% μεταξύ 2022 και 2035.

Οι εξελίξεις στις ψηφιακές τεχνολογίες, η διάδοσή τους σε πολλούς και διαφορετικούς οικονομικούς τομείς και η αλληλεπίδραση μεταξύ της ψηφιακής και της πράσινης μετάβασης θα οδηγήσουν σε αλλαγές στις δεξιότητες που απαιτούνται από τους επαγγελματίες ΤΠΕ στο μέλλον. Οι περισσότεροι από τους μελλοντικούς επαγγελματίες ΤΠΕ θα χρειαστούν διεπιστημονικές δεξιότητες σε σχήμα T.

Παράγοντες αλλαγής που θα επηρεάσουν τις δεξιότητες που απαιτούνται

- **Εξελίξεις στην τεχνολογία και στις αξιακές αλυσίδες:** Καθώς η ΤΠΕ ενσωματώνεται όλο και περισσότερο στις οικονομικές δραστηριότητες, έχουν αναπτυχθεί και εξακολουθούν να αναπτύσσονται πολυάριθμες εφαρμογές λογισμικού. Οι εξελίξεις στην τεχνολογία και στις αξιακές αλυσίδες θα οδηγήσουν ενδεχομένως σε μετατόπιση του κέντρου βάρους από τις τεχνικές δεξιότητες ΤΠΕ σε ειδικές ανά τομέα γνώσεις και σε μη τεχνικές δεξιότητες, όπως η διαχείριση και ο σχεδιασμός.
- **Περαιτέρω ψηφιοποίηση της οικονομίας:** Η ψηφιοποίηση θα αυξήσει σημαντικά τη ζήτηση ατόμων που θα γνωρίζουν σε βάθος τον συγκεκριμένο τομέα και τα οποία θα είναι ικανά να αναπτύσσουν αποτελεσματικές και εξατομικευμένες λύσεις ΤΠΕ για κάθε επιχείρηση ή οργανισμό.

- **Αύξηση εξωπορισμού τεχνικών δεξιοτήτων ΤΠΕ σε πιο φτηνές αγορές εκτός της ΕΕ:** Οι επαγγελματίες της ΕΕ θα πρέπει να διαθέτουν δεξιότητες σε διάφορους τομείς, στο πλαίσιο της ΤΠΕ, όπως είναι η διαχείριση μιας αλυσίδας εφοδιασμού.
- **Ολοένα και πιο ισχυροί υπολογιστές:** Οι διαρκώς ισχυρότεροι ηλεκτρονικοί υπολογιστές θα οδηγήσουν σε αύξηση της ποσότητας και της ποικιλίας των δεδομένων που παράγονται. Η τάση για «μαζικά δεδομένα» θα οδηγήσει σε ζήτηση για υψηλές δεξιότητες ανάλυσης δεδομένων και δεξιότητες στην κλιμακοποίηση και τη διαχείριση δεδομένων για επιχειρήσεις. Ως αποτέλεσμα, αναμένεται να αναδυθούν νέα επαγγέλματα (π.χ. επιστήμονες δεδομένων, διαχειριστές δεδομένων και υπεύθυνοι δεδομένων).
- **Η μετάβαση προς το υπολογιστικό νέφος:** Η τάση αυτή αναμένεται να αυξηθεί με ταχύ ρυθμό τόσο για τις επιχειρήσεις όσο και για τους καταναλωτές, μειώνοντας τη ζήτηση για τεχνικές γνώσεις από την πλευρά των χρηστών, καθώς οι υπηρεσίες θα ανατίθενται σε εξωτερικούς παρόχους υπολογιστικού νέφους. Αυτό σημαίνει ότι οι επιχειρήσεις θα χρειάζονται δεξιότητες για την ενοποίηση των υπηρεσιών, τη διαχείριση υπηρεσιών, τον σχεδιασμό και τη διαχείριση υπολογιστικών νεφών και τη δημιουργία και βελτιστοποίηση των κέντρων δεδομένων υπολογιστικού νέφους.
- **Αυτοματοποίηση:** Καθώς η έρευνα και οι επενδύσεις της βιομηχανίας στην αυτοματοποίηση και στην ανάπτυξη κόμβων έξυπνων οικοσυστημάτων αυξάνονται, θα αυξηθεί και η ζήτηση εξειδίκευσης στον τομέα του λογισμικού και του υλικού με υψηλού επιπέδου γνώσεις αριθμητισμού και πεδίου. Οι επαγγελματίες με τις εν λόγω δεξιότητες θα είναι πολύτιμοι τόσο στις καθιερωμένες επιχειρήσεις που θα επιδιώκουν να ενισχύσουν τη θέση τους στην αγορά όσο και στις νεοφυείς επιχειρήσεις που θα επιδιώκουν να βρουν τη δική τους θέση στην υφιστάμενη αγορά.
- **Η ανάπτυξη του Διαδικτύου των Πραγμάτων (IoT):** Η εν λόγω ανάπτυξη θα αυξήσει τη ζήτηση για δεξιότητες και επαγγέλματα σε σχέση με την αρχιτεκτονική και τον σχεδιασμό, τη γνώση και τις δεξιότητες στον χειρισμό διαφοροποιημένων συστημάτων και την κατανόηση της τυποποίησης και της διαλειτουργικότητας μεταξύ συνδεδεμένων (και προς σύνδεση) συστημάτων. Επίσης, θα αυξηθεί η ζήτηση τεχνογνωσίας στα δίκτυα του Διαδικτύου των Πραγμάτων, καθώς και για δεξιότητες στη διαχείριση των παραμέτρων πολλαπλών δικτύων που αποτελούν μέρος τους.
- **Διασύνδεση χάρη σε «έξυπνα συστήματα»:** Καθώς οι διάφορες συνιστώσες των υποδομών ΤΠΕ θα διασυνδέονται όλο και περισσότερο μεταξύ τους με την ανάπτυξη των «έξυπνων συστημάτων», οι απειλές εγκληματικότητας και τρομοκρατίας στον κυβερνοχώρο θα επεκταθούν. Κατά συνέπεια, προβλέπεται αυξημένη ζήτηση για δεξιότητες στον τομέα της επιστήμης και της ανάλυσης δεδομένων, σε συνδυασμό με επιχειρηματικές γνώσεις. Η ζήτηση για δεξιότητες στον τομέα της κυβερνοασφάλειας όσον αφορά αμφότερα τα συστήματα λογισμικού και υλικού θα αυξηθεί και οι επαγγελματίες θα χρειαστεί ενδεχομένως να διαθέτουν υψηλού επιπέδου προσόντα για να μπορούν να ανταποκριθούν στις απαιτήσεις των διασυνδεδεμένων «έξυπνων» συστημάτων υποδομών του μέλλοντος.

Οι ανάγκες δεξιοτήτων των επαγγελματιών των ΤΠΕ επικεντρώνονται στην καινοτομία, τον αλφαριθμητισμό και τις μαθηματικές εργασίες. Δεν αποτελεί έκπληξη το γεγονός ότι οι ψηφιακές δεξιότητες – τόσο εργαλεία γραφείου όσο και εξειδικευμένο λογισμικό και βάσεις δεδομένων – έχουν τη μεγαλύτερη ζήτηση σε σύγκριση με τον μέσο όρο όλων των επαγγελματιών. Ανάλυση OJA του Cedefop δείχνει ότι η SQL, η Javascript, η PHP και η Python είναι οι πιο απαιτητικές γλώσσες προγραμματισμού και υποστηρίζει επίσης ότι μεταξύ των επαγγελματιών θέσεων εργασίας, το 15% των OJA αναφέρουν τουλάχιστον μία δεξιότητα τεχνητής νοημοσύνης.

Η αλλαγή των αναγκών σε δεξιότητες υποδηλώνει επίσης μεγαλύτερη ανάγκη για περαιτέρω κατάρτιση – τόσο οι συνολικές όσο και οι ψηφιακές ανάγκες αναβάθμισης δεξιοτήτων είναι υψηλότερες από τον μέσο όρο όλων των επαγγελματιών.

Πρόσφατες έρευνες δείχνουν ότι η ζήτηση για δεξιότητες Τεχνητής Νοημοσύνης έχει σχεδόν τριπλασιαστεί ενώ ο αριθμός των συναφών θέσεων εργασίας έχει αυξηθεί κατά 119%. Οι εργοδότες, ωστόσο, συναντούν πρωτοφανείς δυσκολίες να βρουν υποψήφιους με τον κατάλληλο συνδυασμό δεξιοτήτων στο υφιστάμενο εργατικό δυναμικό. Πέρα από τη ζήτηση, το κενό ενισχύεται περαιτέρω από την ελλιπή ανάπτυξη σχετικών δεξιοτήτων από την Επαγγελματική Εκπαίδευση και Κατάρτιση, δεδομένου ότι η Τεχνητή Νοημοσύνη αποτελεί κατά κύριο λόγο αντικείμενο εξειδίκευσης των επαγγελματιών ΤΠΕ που προσφέρεται στο υψηλότερο επίπεδο της τριτοβάθμιας εκπαίδευσης. Επομένως καθίσταται απαραίτητη η αναβάθμιση και βελτίωση των συστημάτων Επαγγελματικής Εκπαίδευσης και Κατάρτισης που απευθύνονται τόσο σε επαγγελματίες (συνεχιζόμενη κατάρτιση) όσο και σπουδαστές (αρχική εκπαίδευσης), ώστε οι υπάρχοντες αλλά και μελλοντικοί επαγγελματίες του κλάδου των ΤΠΕ να αποκτήσουν και να αναπτύξουν τις απαιτούμενες δεξιότητες Τεχνητής Νοημοσύνης και να μπορούν να ανταπεξέλθουν στις σύγχρονες απαιτήσεις της αγοράς εργασίας.

3.7 Οι ενότητες που δημιουργήσαμε (Syllabus)

Στην πρώτη ενότητα μάθησης “Θεμέλια της Τεχνητής Νοημοσύνης” παρουσιάζεται ένας γενικός ορισμός του τι είναι η τεχνητή νοημοσύνη, οι θεμελιώδεις τομείς που αποτελούν τη βάση όλων των εφαρμογών και των αλγορίθμων της, καθώς και μια προοπτική των ευρύτερων τομέων εφαρμογής.

Η μηχανική μάθηση είναι ο τομέας της τεχνητής νοημοσύνης που έχει μεγαλύτερο αντίκτυπο στην τρέχουσα τεχνολογία. Οι τεχνικές της επιτρέπουν την επίλυση πολύπλοκων προβλημάτων με τη χρήση δεδομένων χωρίς το βάρος της ανάπτυξης πολύπλοκων προγραμμάτων με το χέρι. Βοηθούν στον εντοπισμό κάθε είδους μοτίβων που μπορούν να χρησιμοποιηθούν για τη λήψη καλύτερων αποφάσεων. Οι εφαρμογές της μπορούν να βρεθούν σε όλα τα είδη βιομηχανιών και τομέων, από την αυτοματοποίηση στη μεταποίηση, τις τράπεζες, την υγειονομική περίθαλψη ή τις οικιακές συσκευές μέχρι

ολόκληρους κλάδους όπως η Επιστήμη Δεδομένων, όπου παίζει κεντρικό ρόλο στην ανάλυση δεδομένων για την απόκτηση μοτίβων και μοντέλων που χρησιμοποιούνται για τη δημιουργία νέων προϊόντων ή τη λήψη επιχειρηματικών αποφάσεων.

Στη δεύτερη ενότητα μάθησης “Εισαγωγή στη μηχανική μάθηση” παρουσιάζονται οι πιο συχνά χρησιμοποιούμενοι αλγόριθμοι μηχανικής μάθησης που χρησιμοποιούνται για την ανάπτυξη εφαρμογών μηχανικής μάθησης, καθώς και η διαδικασία που περνά από τα δεδομένα ενός προβλήματος σε ένα μοντέλο που επιλύει μια συγκεκριμένη εργασία, από τον προσδιορισμό του τύπου αλγορίθμου μηχανικής μάθησης που είναι κατάλληλος για μια εφαρμογή, την προεπεξεργασία και την οπτικοποίηση των δεδομένων έως τον συντονισμό των μεθόδων μηχανικής μάθησης για τη δημιουργία ενός μοντέλου και την αξιολόγηση και σύγκριση των αποτελεσμάτων ενός μοντέλου.

Πέρα από μια εισαγωγή στην Τεχνητή Νοημοσύνη και τη Μηχανική Μάθηση που υποχρεωτικά έπρεπε να συμπεριληφθούν στα μαθήματα, υπάρχουν δύο επιπλέον ενότητες. Αυτές είναι τα Νευρωνικά δίκτυα και βαθιά μάθηση για την όραση και η Βαθιά μάθηση για επεξεργασία φυσικής γλώσσας & ανάλυση μεγάλων δεδομένων.

Τα τεχνητά νευρωνικά δίκτυα (ANN) και η εξέλιξή τους, τα βαθιά νευρωνικά δίκτυα (DNN), αποτελούν τη βάση των σύγχρονων εφαρμογών τεχνητής νοημοσύνης. Αυτοί οι αλγόριθμοι μηχανικής μάθησης φέρνουν επανάσταση στον τρόπο με τον οποίο πραγματοποιούμε όχι μόνο τις επιστημονικές και βιομηχανικές διαδικασίες αλλά και τις καθημερινές μας δραστηριότητες, με ουσιαστικές παγκόσμιες συνέπειες σε οικονομικό και κοινωνικό επίπεδο. Η ενότητα αυτή ξεκινά εξηγώντας πώς τα βιολογικά νευρικά συστήματα ενέπνευσαν την ανάπτυξη των ANN και παρουσιάζει την εξέλιξη των αρχιτεκτονικών τους σε σχέση με τις κύριες τεχνικές και τα καθήκοντα εφαρμογής, από τα απλά perceptrons και την επιβλεπόμενη μάθηση, μέχρι τα πολυεπίπεδα perceptrons και το Keras, μέχρι τα διάφορα Convolutional Neural Networks για την ταξινόμηση εικόνων. Ο εκπαιδευόμενος θα είναι σε θέση να χρησιμοποιήσει και να επανεκπαιδεύσει διαφορετικά ενσωματωμένα CNN της Keras και να υλοποιήσει ένα έργο εντοπισμού αντικειμένων σε πραγματικό χρόνο.

Η τέταρτη μαθησιακή ενότητα είναι αφιερωμένη στην Επεξεργασία Φυσικής Γλώσσας Natural Language Processing (NLP) και στην Ανάλυση Μεγάλων Δεδομένων, δύο από τους πιο ανεπτυγμένους και αναπτυσσόμενους τομείς εφαρμογής της Τεχνητής Νοημοσύνης (TN). Όσον αφορά τη (NLP) (μαθήματα 1-3), η ενότητα αυτή παρέχει στον εκπαιδευόμενο τις απαραίτητες γνώσεις σχετικά με τις προσεγγίσεις και τους αλγορίθμους, καθώς και

δεξιότητες για τη χρήση διαφόρων βιβλιοθηκών (NLP) για την υλοποίηση εφαρμογών (NLP) (π.χ., ενσωμάτωση κειμένου για επεξεργασία με νευρωνικά δίκτυα, γλωσσική ταξινόμηση, προχωρημένα θέματα NLP) και για την επιλογή και εφαρμογή του κατάλληλου μοντέλου και εργαλείων NLP με βάση την προς επίλυση εργασία. Όσον αφορά την Ανάλυση Μεγάλων Δεδομένων (μαθήματα 4-6), αυτή η Ενότητα παρουσιάζει τις βασικές τεχνικές των Μεγάλων Δεδομένων, τα κύρια εργαλεία λογισμικού, όπως το Hadoop και το Spark, και εργαλεία για υπολογιστικό νέφος. Ο εκπαιδευόμενος εφοδιάζεται με την απαραίτητη τεχνογνωσία για την αποθήκευση, τον χειρισμό και την οπτικοποίηση μαζικών συνόλων δεδομένων και για την εκτέλεση εργασιών Ανάλυσης Μεγάλων Δεδομένων στο νέφος (π.χ. παλινδρόμηση, ταξινόμηση και ομαδοποίηση) με εργαλεία όπως το PySpark.

Μέρος 2^ο

Αναλυτική περιγραφή ενοτήτων

Ενότητα 1: Θεμέλια της τεχνητής νοημοσύνης

Μάθημα 1.1: Πεδίο εφαρμογής της τεχνητής νοημοσύνης

Θέματα προς συζήτηση

- Τι είναι η Τεχνητή Νοημοσύνη;
- Τομείς της Τεχνητής Νοημοσύνης
 - Θεμελιώδεις τομείς
 - Εξειδικευμένοι τομείς

Σημειώσεις διάλεξης

- Τι είναι η Τεχνητή Νοημοσύνη;

Η νοημοσύνη είναι ένα από τα πιο σημαντικά ανθρώπινα χαρακτηριστικά μας. Είμαστε σε θέση να επιλύουμε έναν τεράστιο αριθμό διαφορετικών εργασιών, είμαστε σε θέση να φανταζόμαστε και να δημιουργούμε νέα πράγματα, είμαστε σε θέση να μαθαίνουμε και να προσαρμοζόμαστε σε κάθε νέα κατάσταση που αντιμετωπίζουμε. Δεν αποτελεί έκπληξη το γεγονός ότι μεταξύ των πραγμάτων που φανταζόμαστε είναι το πώς θα δημιουργήσουμε μια μηχανή που θα είναι ικανή να διαθέτει νοημοσύνη. Δεδομένου ότι δεν υπάρχει ένας αποδεκτός ορισμός της ανθρώπινης νοημοσύνης, κυρίως επειδή περιλαμβάνει τόσα πολλά διαφορετικά πράγματα, είναι επίσης δύσκολο να ορίσουμε τι είναι η τεχνητή νοημοσύνη. Κατά τη διάρκεια της ιστορίας του πεδίου, έχουν εμφανιστεί πολλοί ορισμοί με διαφορετικά οράματα για το τι πρέπει να είναι μια ευφυής μηχανή.

Ορισμένοι ορισμοί επικεντρώνονται στο ανθρώπινο μοντέλο και ορίζουν ένα πρόγραμμα ικανό για τεχνητή νοημοσύνη ως κάτι που μπορεί να αναπαράγει ό,τι κάνουν οι άνθρωποι. Υπάρχουν ορισμοί με διαφορετική εστίαση και εμβέλεια, για ορισμένους ορισμούς το σημαντικό είναι να αναπαράγει τις ανθρώπινες νοητικές ικανότητες (δηλαδή τη σκέψη), όπως η συλλογιστική, η επίλυση προβλημάτων ή η μάθηση, για άλλους ορισμούς μια νοημοσύνη πρέπει να αναπαράγει όλες τις παρατηρήσιμες συμπεριφορές (δηλαδή να ενεργεί) και να είναι δυσδιάκριτη από έναν άνθρωπο.

Αυτή η μίμηση όλων των ανθρώπινων δεξιοτήτων είναι το επίκεντρο της ιδέας του τεστ Τούρινγκ. Μια ιδέα που έχει γίνει δημοφιλής από τότε που η τεχνητή νοημοσύνη μπήκε διάχυτα στη ζωή μας. Η ιδέα είναι ότι για να διαπιστωθεί αν ένα πρόγραμμα είναι πραγματικά ευφυές πρέπει να περάσει ένα παιχνίδι μίμησης. Μια ομάδα ανθρώπινων κριτών θα υποβάλει ένα σύνολο ερωτήσεων που θα απαντηθούν από το πρόγραμμα και έναν άνθρωπο. Αν οι κριτές δεν μπορούν να καταλάβουν ποιες απαντήσεις προέρχονται από το πρόγραμμα και ποιες από τον άνθρωπο, τότε το πρόγραμμα έχει περάσει το τεστ. Ορισμένες παραλλαγές του τεστ περιλαμβάνουν πιο αντιληπτικές εργασίες όπως η όραση ή η ομιλία.

Μια λιγότερο ανθρωποκεντρική προσέγγιση της τεχνητής νοημοσύνης την ορίζει από την άποψη του ορθολογισμού. Ένα ευφυές πρόγραμμα θα πρέπει να ενεργεί για να επιτύχει το καλύτερο δυνατό αποτέλεσμα στο έργο που του έχει ανατεθεί (δηλαδή να ενεργεί ορθολογικά). Αυτό το βέλτιστο αποτέλεσμα εξαρτάται από το περιβάλλον και τη δυσκολία της εκάστοτε εργασίας. Αυτή είναι η προσέγγιση του ορθολογικού πράκτορα. Αυτός ο ορισμός ισχυρίζεται ότι είναι πιο γενικός από το να εστιάζει στους νόμους που ορίζουν την ορθολογική σκέψη (δηλ. να σκέφτεται ορθολογικά). Αυτοί οι νόμοι μπορούν να οριστούν μαθηματικά χρησιμοποιώντας τη λογική.

Ο ορισμός του ορθολογικού πράκτορα περιλαμβάνει τη λογική θεώρηση ως έναν πιθανό τρόπο επίτευξης ορθολογικής συμπεριφοράς, αλλά περιλαμβάνει και άλλους τρόπους λήψης αποφάσεων που καθιστούν τις επιλογές του πράκτορα πιο αποτελεσματικές. Αυτή η προσέγγιση είναι πιο γενική και περιλαμβάνει επίσης όλους τους τομείς που απαιτούνται για να περάσει το τεστ Turing.

• Τομείς της Τεχνητής Νοημοσύνης

Η τεχνητή νοημοσύνη έχει επικεντρωθεί σε όλα τα διαφορετικά χαρακτηριστικά που θεωρούνται μέρος της ευφυούς ανθρώπινης συμπεριφοράς. Η επίλυση προβλημάτων, η αναπαράσταση της γνώσης και η μάθηση μπορούν να θεωρηθούν θεμελιώδεις τομείς που βρίσκονται στο κέντρο της εξέλιξης άλλων τομέων της τεχνητής νοημοσύνης.

i. Επίλυση προβλημάτων

Η επίλυση προβλημάτων είναι ένας από τους βασικούς τομείς της τεχνητής νοημοσύνης. Αυτός ο τομέας περιλαμβάνει όλα τα θέματα σχετικά με την απόκτηση λύσης για ένα πρόβλημα που ορίζεται με ένα σύνολο χαρακτηριστικών και με τη χρήση ενός συνόλου διαθέσιμων ενεργειών, εξερευνώντας τις διάφορες δυνατότητές του με τη χρήση μιας στρατηγικής αναζήτησης και γνώσης σχετικά με το πεδίο του προβλήματος.

Υπάρχουν αλγόριθμοι που μπορούν να εκτελέσουν αυτό το έργο χρησιμοποιώντας μια στρατηγική εξαντλητικής αναζήτησης, απαριθμώντας όλες τις πιθανότητες σε ένα συστηματικό τρόπο. Προφανώς, αυτό δεν θα ήταν πολύ έξυπνο και είναι επίσης ανέφικτο για πραγματικές εφαρμογές. Η ευφυής προσέγγιση θα ήταν να χρησιμοποιείται η γνώση σχετικά με το πρόβλημα για να λαμβάνονται αποφάσεις σχετικά με το τι πρέπει να διερευνηθεί πρώτα. Αυτή είναι μια πιο ευφυής προσέγγιση.

Όλοι αυτοί οι αλγόριθμοι αντλούν την έμπνευσή τους από διαφορετικούς τομείς. Κάποιοι είναι πιο αλγοριθμικοί, αλλά περιλαμβάνουν στην απόφασή τους τη λήψη αποφάσεων τη γνώση του τομέα από το πρόβλημα ("ευρετικοί"). Για παράδειγμα, αλγόριθμοι όπως ο A* ή το hill climbing. Άλλοι είναι εμπνευσμένοι από βιολογικές αναλογίες όπως η φυσική εξέλιξη ή η συμπεριφορά των κοινωνικών εντόμων και των ζώων. Για παράδειγμα, οι γενετικοί αλγόριθμοι ορίζουν ένα πρόβλημα ως ένα άτομο που μπορεί να ανταλλάσσει μέρος των πληροφοριών του με άλλα για να λάβει καλύτερες λύσεις χρησιμοποιώντας το κανόνες της γενετικής

ii. Αναπαράσταση γνώσης και συλλογιστική

Η γνώση είναι αυτό που καθιστά δυνατή τη λήψη αποφάσεων, οπότε αποτελεί επίσης θεμελιώδη τομέα της τεχνητής νοημοσύνης. Δεν γνωρίζουμε την πραγματική αναπαράσταση που χρησιμοποιεί ο εγκέφαλός μας για τη συσσώρευση όλων των πληροφοριών που αποκτούμε κατά τη διάρκεια της ζωής μας, αλλά διάφοροι τομείς των μαθηματικών μας παρέχουν μοντέλα που μπορούν να χρησιμοποιηθούν για την ανάπτυξη ευφυών συστημάτων. Η αναπαράσταση της γνώσης περιλαμβάνει φορμαλισμούς για την αναπαράσταση της πραγματικότητας σε έναν υπολογιστή, αλλά και όλους τους μηχανισμούς που επιτρέπουν τη συλλογιστική προκειμένου να να αποκτήσουμε περισσότερη γνώση που συνεπάγεται η αναπαριστώμενη γνώση.

Οι περισσότεροι φορμαλισμοί που χρησιμοποιούνται στην τεχνητή νοημοσύνη προέρχονται από τη μαθηματική λογική. Ο τομέας αυτός παρέχει ένα αξιόπιστο σύνολο μεθόδων που επιτρέπουν τη συλλογιστική και την εξαγωγή συμπερασμάτων. Παρέχει επίσης συγκεκριμένες αναπαραστάσεις για όλα τα είδη γνώσης που μπορεί να είναι απαραίτητες για μια εφαρμογή

Ένα από τα μειονεκτήματα των λογικών αναπαραστάσεων είναι ότι είναι συνήθως δύσκολο να τυποποιηθεί η γνώση που εκφράζεται σε φυσική γλώσσα σε τόσο άκαμπτη αναπαράσταση. Για να γίνει ευκολότερη η μετάφραση της γνώσης σε μια αναπαράσταση που μπορεί να χρησιμοποιηθεί από ένα πρόγραμμα, έχουν οριστεί διάφοροι φορμαλισμοί που βασίζονται σε γραφικές αναπαραστάσεις, όπως για παράδειγμα πλαίσια ή σημασιολογικά δίκτυα. Αυτό έχει επεκτείνει το πεδίο αυτό και σε άλλους τομείς όπως ο Παγκόσμιος Ιστός με έργα όπως ο Σημασιολογικός Ιστός που φιλοδοξεί να συμπεριλάβει τυπικούς ορισμούς σε όλες τις ιστοσελίδες, ώστε να μπορούν να χρησιμοποιηθούν και από προγράμματα. Αυτό έχει επίσης αυξήσει τη σημασία των τομέων που σχετίζονται με την αναπαράσταση γνώσης, όπως η μηχανική οντολογιών.

Υπάρχει κάποια γνώση που δεν μπορεί να αναπαρασταθεί με τη χρήση λογικού φορμαλισμού. Η λογική έχει μια αντίληψη της γνώσης μερικές φορές πολύ άκαμπτη, επειδή κάτι πρέπει να είναι πάντα αληθές ή ψευδές. Ένας τομέας της αναπαράστασης της γνώσης είναι αφιερωμένος στην αναπαράσταση της αβεβαιότητας στη γνώση και στον μηχανισμό συλλογισμού. Για παράδειγμα, τα δίκτυα του Μπέιζ, που εισάγουν στατιστικές κατανομές και την εξαγωγή συμπερασμάτων του Μπέιζ στην αναπαράσταση των γεγονότων και των μεταξύ τους σχέσεων.

Οι γνώσεις αυτές είναι ιδιαίτερα σημαντικές στον τομέα των συστημάτων που βασίζονται στη γνώση (που ονομάζονται επίσης Έμπειρα Συστήματα). Συστήματα επίλυσης προβλημάτων με μεγάλη έμφαση στην αναπαράσταση γνώσεων και τη συλλογιστική.

iii. Μηχανική μάθηση

Η μηχανική μάθηση μπορεί να οριστεί ως η αυτόνομη απόκτηση γνώσης μέσω της χρήσης προγραμμάτων υπολογιστών. Επιτρέπει προγράμματα να προσαρμόζονται σε νέες καταστάσεις ή να βελτιώνουν την απόδοσή τους. Βοηθά επίσης στην επίλυση ενός θεμελιώδους προβλήματος στην επιστήμη των υπολογιστών, υπάρχουν ορισμένες εργασίες που είναι πολύ δύσκολο να προγραμματιστούν με το χέρι και είναι ευκολότερο να φτιάξουμε προγράμματα που μαθαίνουν πώς να τις επιλύσουν από το μηδέν.

Η μηχανική μάθηση είναι ένας ευρύς τομέας που έχει πολλές εφαρμογές. Μπορούμε να μάθουμε με επίβλεψη, δηλαδή να υπάρχει ένας δάσκαλος που λέει στο πρόγραμμα ποια είναι η λύση για ένα συγκεκριμένο πρόβλημα, και αυτό χρησιμοποιεί έναν αλγόριθμο μάθησης για να αποκτήσει μια αναπαράσταση ικανή να αναγνωρίζει όχι μόνο αυτό το πρόβλημα αλλά και άλλα παρόμοια προβλήματα. Μπορούμε επίσης να μάθουμε χωρίς επίβλεψη, προσπαθώντας να βγάλουμε νόημα από τα δεδομένα, βρίσκοντας μοτίβα, αναζητώντας ποια πράγματα είναι παρόμοια και ποια πράγματα είναι διαφορετικά.

Οι επιβλεπόμενοι αλγόριθμοι στη μηχανική μάθηση έχουν διαφορετικές αρχές και περιλαμβάνουν άλλους τομείς όπως η στατιστική ή η μαθηματική ανάλυση. Για παράδειγμα, μια εργασία θα μπορούσε να περιλαμβάνει την εξεύρεση μιας συνάρτησης που προβλέπει μια συνεχή τιμή δεδομένου των χαρακτηριστικών του προβλήματος, η οποία μπορεί να επιλυθεί με τη χρήση παλινδρόμησης ή προσαρμογής συναρτήσεων. Άλλες εργασίες περιλαμβάνουν την ταξινόμηση προβλημάτων σε διαφορετικές κατηγορίες, οπότε κάθε πρόβλημα φέρει ετικέτα και χρειαζόμαστε ένα σύνολο κανόνων που επιτρέπουν τον προσδιορισμό της σωστής κλάσης. Πιο συγκεκριμένοι εποπτευόμενοι αλγόριθμοι μάθησης είναι σε θέση να μαθαίνουν μια εργασία με δοκιμή και σφάλμα, ακόμη και on-line ή χρησιμοποιώντας προσομοιώσεις. Αλγόριθμοι χωρίς επίβλεψη επιτρέπουν την κατανόηση των δεδομένων αναζητώντας κρυμμένα μοτίβα. Μπορούμε να χρησιμοποιήσουμε αλγορίθμους που μπορούν να ανακαλύψουν τις πιθανές ετικέτες των δεδομένων, ομαδοποιώντας τις με βάση την ομοιότητά τους.

iv. Επεξεργασία φυσικής γλώσσας

Μια από τις ικανότητες που μας κάνουν ανθρώπους είναι η ικανότητα να επικοινωνούμε μεταξύ μας χρησιμοποιώντας μια συμβολική γλώσσα. Πρόκειται για έναν εξειδικευμένο τομέα που αντλεί τις γνώσεις του από τη γλωσσολογία. Ο στόχος αυτού του τομέα είναι η αυτοματοποίηση όλων των εργασιών που περιλαμβάνουν την χρήση της φυσικής γλώσσας. Αυτό περιλαμβάνει, για παράδειγμα, την εξαγωγή γνώσης, την παραγωγή γλώσσας, την απάντηση ερωτήσεων ή τη μηχανική μετάφραση

Η κατανόηση της φυσικής γλώσσας δεν είναι εύκολη υπόθεση. Η βασική επεξεργασία του γραπτού κειμένου αποτελείται από πολλά βήματα όπως η ανάλυση του κειμένου για την εξαγωγή των συστατικών του, η γραμματική ανάλυση, ο προσδιορισμός της κατηγορίας για κάθε λέξη, η συντακτική ανάλυση των προτάσεων ή η σημασιολογική τους ανάλυση και ο μετασχηματισμός τους σε ένα φορμαλισμό αναπαράστασης γνώσης. Όλες οι διαδικασίες περιλαμβάνουν πολλαπλές εργασίες που περιλαμβάνουν αλγορίθμους αναζήτησης, αναπαράστασης και συλλογιστικής της γνώσης και μηχανική μάθηση

Για παράδειγμα, η ταξινόμηση των λέξεων μιας πρότασης σε γραμματικές κατηγορίες (Part of Speech Tagging) μπορεί να γίνει με τη δημιουργία ενός προγράμματος για την εργασία αυτή, χρησιμοποιώντας γλωσσολογική γνώση. Αυτό το πρόβλημα περιλαμβάνει μια ομάδα ατόμων για κάθε πιθανή γλώσσα, ή το έργο θα μπορούσε να μαθευτεί από παραδείγματα προτάσεων που έχουν ήδη επισημανθεί. Άλλη εφαρμογή είναι τα συστήματα διαλόγου, πρόκειται για

προγράμματα που μπορούν να έχουν μια συνομιλία. Αυτό περιλαμβάνει όχι μόνο την κατανόηση προτάσεων, αλλά και τη δημιουργία απαντήσεων ανάλογα με τα λεγόμενα. Μια συνομιλία δεν είναι ένα σύνολο από ζεύγη προτάσεων, αλλά ένας συνδεδεμένος και συνεκτικός λόγος. Αυτό σημαίνει ότι πρέπει να αποθηκεύεται μια ακριβής αναπαράσταση των θεμάτων της συζήτησης και να παρακολουθείται ποια στοιχεία αναφέρονται σε κάθε πρόταση. Αυτό συνεπάγεται αναζήτηση και συλλογισμό προκειμένου να αντιστοιχίζονται τα λεγόμενα με όλες τις προτάσεις της συζήτησης. Η επεξεργασία φυσικής γλώσσας θα αποκτήσει σημαντικό ρόλο στο μέλλον, επειδή βοηθά στην πιο φυσική αλληλεπίδραση με ευφυή προγράμματα. Οι διεπαφές που βασίζονται σε αυτές τις τεχνολογίες θα αντικαταστήσουν άλλα είδη διεπαφών σε πολλές μελλοντικές εφαρμογές.

v. Αντίληψη

Στην προσπάθεια δημιουργίας ευφυών συστημάτων, ο τελικός στόχος είναι να μπορέσουμε να αναπτύξουμε αυτά τα προγράμματα στον πραγματικό κόσμο. Αυτό σημαίνει ότι θα διαθέτουν αισθητήρες που θα επιτρέπουν τη συλλογή πληροφοριών από το περιβάλλον προκειμένου να δημιουργηθεί μια εσωτερική αναπαράσταση και να λαμβάνουν αποφάσεις.

Υπάρχουν πολλά είδη αισθητήρων που μπορούν να χρησιμοποιηθούν, αλλά μια από τις πιο δύσκολες λειτουργίες που μπορούμε να φανταστούμε είναι η αναγνώριση των πληροφοριών που περιέχονται σε μια εικόνα. Το πρόβλημα αυτό αντιμετωπίζεται στον τομέα της όρασης υπολογιστών. Υπάρχουν πολλές εργασίες που μπορούν να επιλυθούν με την όραση υπολογιστών, από την αναγνώριση συγκεκριμένων στοιχείων, όπως για παράδειγμα ανθρώπων, μέχρι την αναγνώριση και κατηγοριοποίηση όλων των αντικειμένων που υπάρχουν σε μια εικόνα.

Η δημιουργία ενός συστήματος που είναι σε θέση να αναγνωρίζει τι υπάρχει σε μια εικόνα δεν είναι μια εργασία που μπορεί να γίνει χειροκίνητα. Αυτό σημαίνει ότι πρέπει να εφαρμοστούν τεχνικές μηχανικής μάθησης για να διδαχθεί το σύστημα. Οι πρόσφατες εξελίξεις στα νευρωνικά δίκτυα έχουν θέσει αυτές τις τεχνικές μπροστά στον αγώνα για τη δημιουργία συστημάτων που μπορούν να επιλύουν οπτικές εργασίες.

Η αύξηση της υπολογιστικής ισχύος χάρη στο εξειδικευμένο υλικό, όπως οι Μονάδες Επεξεργασίας Γραφικών (GPUs), έχει οδηγήσει τώρα σε πιο σύνθετες εφαρμογές της όρασης υπολογιστών. Δεν περιοριζόμαστε στην αναγνώριση πληροφοριών από μια φωτογραφία αλλά μπορούμε να χρησιμοποιήσουμε και βίντεο, επιτρέποντας εφαρμογές όπως η παρακολούθηση αντικειμένων (π.χ. επιτήρηση), η αυτόνομη οδήγηση ή η αναγνώριση χειρονομιών.

Ο τομέας αυτός περιλαμβάνει επίσης την αναγνώριση ομιλίας, που συνήθως περιλαμβάνει τη μετάφραση του ηχητικού σήματος σε κείμενο, η οποία επίσης επιλύεται με τη χρήση μοντέλων μηχανικής μάθησης. Τα σήματα από άλλου είδους αισθητήρες μπορούν επίσης να υποβληθούν σε επεξεργασία και ανάλυση για την εξαγωγή και να ανιχνεύσουν μοτίβα, όπως τα λείζερ, οι αισθητήρες υπερύθρων ή υπερήχων.

vi. Ρομποτική

Η αντίληψη είναι ένα από τα στοιχεία που κάνουν τα ρομπότ να γνωρίζουν το περιβάλλον τους, αλλά υπάρχουν πολλές αποφάσεις που υπερβαίνουν την αντίληψη. Από την εποχή που τα πρώτα ρομπότ αναπτύχθηκαν σε εργοστάσια για να αυτοματοποιήσουν τη διαδικασία κατασκευής αυτοκινήτων έχουν εξελιχθεί πολλά πράγματα σε αυτόν τον τομέα.

Τα ρομπότ δεν περιορίζονται πλέον σε ένα στενό περιβάλλον, όπου σχεδόν όλα βρίσκονται σε μια συγκεκριμένη θέση και είναι τέλεια χρονικά προγραμματισμένα. Πρέπει να αναπτύσσουν τα καθήκοντά τους σε μεταβαλλόμενα και αβέβαια περιβάλλοντα. Αυτό σημαίνει ότι πολλές από τις εργασίες που αναπτύσσουν δεν μπορούν να προγραμματιστούν εκ των προτέρων ή τουλάχιστον πρέπει να συνθέσουν ένα σύνολο πρωτόγωνων λειτουργιών προκειμένου να επιτύχουν ορισμένους στόχους.

Αυτό καθιστά την επίλυση προβλημάτων και την αναζήτηση, με τη σύνδεσή τους με τον αυτόματο σχεδιασμό, ένα από τα στοιχεία του προγραμματισμού τους. Οι πιο δύσκολες εργασίες στη ρομποτική, οι οποίες είναι πολύ δύσκολες για να προγραμματιστούν χειροκίνητα και οι οποίες περιλαμβάνουν πιο σύνθετες διαδικασίες πρέπει να μαθαίνονται χρησιμοποιώντας προσομοιωμένα περιβάλλοντα ή πραγματικές αλληλεπιδράσεις, εφαρμόζοντας αλγόριθμους μηχανικής μάθησης όπως αυτοί που αναπτύχθηκαν στην ενισχυτική μάθηση.

Έχουμε ήδη ρομπότ που μπορούν να περπατήσουν, να τρέξουν ή να πηδήξουν. Ρομπότ που μπορούν να αλληλεπιδρούν μέσα σε περιορισμένα περιβάλλοντα με πιο σύνθετους τρόπους, όπως το άνοιγμα θυρών ή το περπάτημα σε σκάλες. Η αντίληψή μας για το τι είναι ρομπότ έχει επίσης επεκταθεί σε μια μεγάλη ποικιλία μηχανών, από αυτά που μπορούν να παραλάβουν και να μεταφέρουν προϊόντα σε μια αποθήκη παιχνιδιών που μπορούν να αλληλεπιδρούν με τα παιδιά και να δείχνουν διαθέσεις ή συναισθήματα ή να απαντούν στις ερωτήσεις μας, αυτοματοποιημένες ηλεκτρικές σκούπες που μπορούν να σκουπίζουν τα πατώματά μας μέχρι μη επανδρωμένα εναέρια οχήματα που μπορούν να κάνουν επιτήρηση. Όλες αυτές οι εργασίες περιλαμβάνουν ευφυείς συμπεριφορές που σχετίζονται με γνωστικές ικανότητες, όπως η χαρτογράφηση του περιβάλλοντος, η επίλυση προβλημάτων με στόχευση, η αναγνώριση προβλημάτων και η συλλογιστική για τους άλλους.

Η δυνατότητα ύπαρξης πολλαπλών ρομπότ έθεσε επίσης το πρόβλημα του συντονισμού, της συνεργασίας και του ανταγωνισμού. Αυτό ανοίγει τη ρομποτική σε πιο σύνθετους τομείς της τεχνητής νοημοσύνης, όπως η κατανεμημένη τεχνητή νοημοσύνη, όπου δεν υπάρχει μόνο ένας πράκτορας που επιλύει μια εργασία, με ικανότητες όπως η επικοινωνία ή η διαπραγμάτευση με τη χρήση της θεωρίας παιγνίων.

Περιπτώσεις χρήσης

ΡΟΜΠΟΤΙΚΕΣ ΗΛΕΚΤΡΙΚΕΣ ΣΚΟΥΠΕΣ

Οι ρομποτικές συσκευές είναι ο τέλειος συνδυασμός πολλαπλών τεχνικών τεχνητής νοημοσύνης. Ένα παράδειγμα είναι οι ρομποτικές ηλεκτρικές σκούπες. Οι εργασίες που εκτελούν αυτά τα ρομπότ είναι φαινομενικά απλές, αλλά χρειάζονται πολύπλοκους αλγόριθμους τεχνητής νοημοσύνης. Ένα από τα πράγματα που πρέπει να κάνουν αυτά τα ρομπότ είναι να χαρτογραφούν

τα δωμάτια και να βρίσκουν τη θέση τους μέσα σε αυτά. Αυτό μπορεί να γίνει χρησιμοποιώντας διάφορους διαφορετικούς αισθητήρες. Μια δυνατότητα είναι να λαμβάνονται φωτογραφίες του δωματίου και να εξάγονται συγκεκριμένα χαρακτηριστικά που επιτρέπουν στο ρομπότ να βρίσκει σημεία που μπορούν να αναγνωριστούν, όπως για παράδειγμα τραπέζια, καρέκλες ή καναπέδες. Αυτή η εργασία χρειάζεται αλγορίθμους υπολογιστικής όρασης για την εξαγωγή των χαρακτηριστικών από τις εικόνες. Αυτό θα χρησιμοποιηθεί για να αποφασιστεί ποια χαρακτηριστικά αντιστοιχούν σε ορόσημα και τα οποία θα αποθηκευτούν για να δημιουργηθεί ένας χάρτης για το δωμάτιο ή τα δωμάτια που έχουν επισκεφθεί κατά τη διάρκεια του καθαρισμού. Το ρομπότ επαναλαμβάνει αυτή την εργασία περιοδικά για να ενημερώνει τον χάρτη και να ενημερώνει τη θέση του. Το ρομπότ πρέπει επίσης να αποφασίσει πότε έχει περάσει από το ένα δωμάτιο στο άλλο.

Άλλοι αισθητήρες μπορούν να χρησιμοποιηθούν για την ανίχνευση της εγγύτητας με άλλα αντικείμενα που δεν μπορούν να ανιχνευθούν με τη χρήση της όρασης. Τα δωμάτια δεν παραμένουν στατικά και τα αντικείμενα που δεν ανιχνεύονται με τη χρήση όρασης πρέπει επίσης να αποφεύγονται. Μερικές φορές πρέπει να ανιχνεύονται και να αποφεύγονται ακόμη και κινούμενα αντικείμενα, όπως για παράδειγμα κατοικίδια ζώα. Αυτοί οι αισθητήρες παρέχουν συνεχώς πληροφορίες στο ρομπότ, το οποίο πρέπει να αποφασίσει το σχέδιο που απαιτείται για την αποφυγή του αντικειμένου. Άλλοι αισθητήρες ανιχνεύουν επίσης την επιφάνεια που καθαρίζεται, για να αποφασίσουν για παράδειγμα ότι βρίσκεται πάνω από ένα χαλί και ότι οι κινήσεις και ο καθαρισμός πρέπει να είναι διαφορετικοί ή ότι το ρομπότ πρόκειται να κινηθεί προς ένα χαλί, οπότε η ταχύτητα που απαιτείται για να ανέβει στο χαλί πρέπει να αυξηθεί. Πρέπει επίσης να ανιχνεύονται πιο συγκεκριμένα πράγματα, για παράδειγμα ότι το ρομπότ βρίσκεται μπροστά από μια σκάλα και ότι η κίνηση προς τα εμπρός σημαίνει πτώση προς τα κάτω.

Χρησιμοποιώντας τις πληροφορίες εντοπισμού πρέπει επίσης να ληφθεί ένα σχέδιο για τον καθαρισμό των δωματίων. Αυτό χρειάζεται αλγορίθμους σχεδιασμού που θα αποφασίσουν μια τροχιά μέσα σε κάθε δωμάτιο για να διασφαλίσουν ότι καθαρίζεται ολόκληρη η περιοχή ή ότι ειδικά σημεία μέσα στο δωμάτιο πρέπει να προσεγγιστούν με διαφορετικό τρόπο, όπως κάτω από έναν καναπέ, ή ότι οι γωνίες του δωματίου χρειάζονται διαφορετική ταχύτητα ή περισσότερο χρόνο για να καθαριστούν. Όλες αυτές οι αποφάσεις συνδυάζονται με τον έλεγχο των ενεργοποιητών του ρομπότ, των τροχών και των στοιχείων καθαρισμού, λαμβάνοντας αποφάσεις για τον τρόπο ελιγμών μέσα στο δωμάτιο ή τη διέλευση από το ένα δωμάτιο στο άλλο. Το σχέδιο καθαρισμού πρέπει επίσης να λαμβάνει υπόψη την εσωτερική κατάσταση του ρομπότ, όπως αν είναι γεμάτο, οπότε δεν μπορεί να καθαρίσει άλλο, μετακινούμενο ενδεχομένως προς το σταθμό βάσης για να πετάξει όλη τη βρωμιά που έχει συλλέξει, ή ότι η μπαταρία είναι χαμηλή, οπότε πρέπει επίσης να μετακινηθεί προς το σταθμό βάσης για να φορτίσει πριν συνεχίσει την εργασία καθαρισμού. Όλες αυτές οι εργασίες σχεδιασμού χρειάζονται κάποιες τεχνικές αναπαράστασης γνώσης, ώστε ορισμένες αποφάσεις να βασίζονται σε κανόνες και συλλογισμούς.

Οι σύγχρονες ρομποτικές ηλεκτρικές σκούπες εξελίσσονται γρήγορα, οπότε εκτός από την εισαγωγή πιο σύγχρονων τεχνικών τεχνητής νοημοσύνης, προχωρούν και σε άλλες παρόμοιες εργασίες, όπως για παράδειγμα το κούρεμα του γκαζόν ή ο καθαρισμός της πισίνας.

<https://www.youtube.com/watch?v=YVG5--9Sjro>

Αναφορές

- **Artificial intelligence: a modern approach** Russell, S.J., Norvig, P, Prentice-Hall, (τέταρτη έκδοση) 2020. ISBN: 9781292153964 (κεφάλαιο 1)
- **Τεχνητή νοημοσύνη: μια νέα σύνθεση** Nilsson, N.J., Morgan Kaufmann Publishers, 1998. ISBN: 1558604677 (κεφάλαιο 1)
- **Τεχνητή νοημοσύνη: δομές και στρατηγικές για την επίλυση σύνθετων προβλημάτων** - Luger, G.F, Pearson Education: Addison Wesley, 2009. ISBN: 9780321545893 (κεφάλαια 1 και 16)

Μάθημα 1.2: Επίλυση προβλημάτων με αλγόριθμους αναζήτησης

Θέματα προς συζήτηση

- Εισαγωγή στην επίλυση προβλημάτων
- Αναζήτηση στο χώρο των καταστάσεων
- Αναζήτηση χώρου λύσεων
- Προβλήματα ικανοποίησης περιορισμών

Σημειώσεις διάλεξης

i. Τι είναι πρόβλημα;

Μπορούμε να ορίσουμε την επίλυση προβλήματος ως τη διαδικασία που, ξεκινώντας από ένα σύνολο αρχικών στοιχείων και χρησιμοποιώντας ένα σύνολο επιλεγμένων ενεργειών, είναι σε θέση να καθορίσει την ακολουθία βημάτων που οδηγεί στη λύση του προβλήματος. Αυτό μπορεί να είναι ένα σύνολο ενεργειών που αποκτούν ένα σύνολο στοιχείων που κατέχουν ορισμένες ιδιότητες ή τον τρόπο με τον οποίο συνδυάζονται τα στοιχεία ώστε να κατέχουν ορισμένους περιορισμούς.

Προφανώς, υπάρχουν πολλά είδη προβλημάτων, αλλά όλα μοιράζονται ένα σύνολο κοινών χαρακτηριστικών που τους επιτρέπουν να ταξινομηθούν και να δομηθούν. Επομένως, είναι εύλογη η ιδέα να υπάρχουν αλγόριθμοι ικανοί να επιλύουν προβλήματα αυτόματα, αν αναπαραστήσουμε επαρκώς τα χαρακτηριστικά τους.

Για την αυτόματη επίλυση προβλημάτων, είναι πρώτα απαραίτητο να μπορούμε να αναπαραστήσουμε τα χαρακτηριστικά των προβλημάτων με έναν τυπικό και δομημένο τρόπο. Αυτό σημαίνει ότι πρέπει να ορίσουμε μια κοινή γλώσσα για τον ορισμό των προβλημάτων. Δεύτερον, πρέπει να ορίσουμε αλγόριθμους που αντιπροσωπεύουν στρατηγικές ικανές να βρουν

τη λύση ενός προβλήματος, που χρησιμοποιούν την κοινή αναπαράσταση και που εγγυώνται μέχρι ένα ορισμένο βαθμό ότι θα βρεθεί η λύση που θέλουμε.

Αν συνοψίσουμε τα χαρακτηριστικά που περιγράφουν ένα πρόβλημα, μπορούμε να εντοπίσουμε τα ακόλουθα στοιχεία:

- Ένα σημείο εκκίνησης, τα στοιχεία που καθορίζουν τα χαρακτηριστικά του προβλήματος.
- Ένας στόχος προς επίτευξη, αυτό που θέλουμε να επιτύχουμε με την επίλυσή του.
- Ενέργειες/λειτουργίες που έχουμε στη διάθεσή μας για την επίλυση του προβλήματος, εργαλεία χειρισμού των στοιχείων που περιγράφουν την κατάσταση του προβλήματος.
- Περιορισμοί σχετικά με τον στόχο, ποια χαρακτηριστικά πρέπει να έχει η λύση
- Στοιχεία που σχετίζονται με το πρόβλημα που ορίζεται από το συγκεκριμένο πεδίο, γνώσεις που έχουμε από το πεδίο που μπορούν να μας βοηθήσουν να λύσουμε το πρόβλημα με αποτελεσματικό τρόπο.

ii. Αναζήτηση χώρου καταστάσεων

Η πιο γενική προσέγγιση για την αναπαράσταση ενός προβλήματος είναι ο λεγόμενος χώρος καταστάσεων. Πρέπει να υποθέσουμε ότι μπορούμε να περιγράψουμε το πρόβλημα από τα στοιχεία που εμπλέκονται σε αυτό και τις σχέσεις του. Σε κάθε βήμα της επίλυσής του, τα στοιχεία αυτά θα έχουν συγκεκριμένα χαρακτηριστικά και σχέσεις που θα αρκούν για να καθορίσουν το σημείο της επίλυσης στο οποίο βρισκόμαστε.

Ονομάζουμε κατάσταση την αναπαράσταση των στοιχείων που περιγράφουν το πρόβλημα σε μια δεδομένη στιγμή. Θα διακρίνουμε σε κάθε πρόβλημα δύο ειδικές καταστάσεις, την αρχική κατάσταση (σημείο εκκίνησης) και την τελική κατάσταση (στόχος του προβλήματος).

Για να μετακινηθούμε μεταξύ των διαφορετικών καταστάσεων, θα χρειαστούμε αυτό που θα ονομάσουμε τελεστή μετασχηματισμού (ή δράση). Ένας τελεστής είναι μια συνάρτηση μετασχηματισμού στην αναπαράσταση μιας κατάστασης που τη μετατρέπει σε μια άλλη κατάσταση. Οι τελεστές θα ορίζουν μια σχέση προσβασιμότητας μεταξύ των καταστάσεων.

Τα στοιχεία που ορίζουν έναν τελεστή είναι τα εξής:

1. Προϋποθέσεις εφαρμογής, οι προϋποθέσεις που πρέπει να πληροί μια κατάσταση για να την εφαρμόσουμε.
2. Συνάρτηση μετασχηματισμού, ο μετασχηματισμός που πρέπει να εφαρμοστεί στην κατάσταση για να επιτευχθεί μια διάδοχη κατάσταση της τρέχουσας κατάστασης.

Το σύνολο των καταστάσεων ενός προβλήματος και η σχέση προσβασιμότητάς τους με δεδομένους τους τελεστές ορίζουν αυτό που ονομάζεται χώρος καταστάσεων. Αυτός αντιπροσωπεύει όλες τις διαδρομές μεταξύ όλων των πιθανών καταστάσεων ενός προβλήματος. Θα μπορούσε να εξομοιωθεί με έναν οδικό χάρτη του προβλήματος. Η λύση του προβλήματός μας βρίσκεται εντός αυτού του χάρτη, απλώς πρέπει να βρούμε το σωστό μονοπάτι.

Το τελευταίο στοιχείο που απομένει να οριστεί είναι η λύση. Θα την ορίσουμε με δύο τρόπους, όπως η ακολουθία βημάτων που οδηγεί από την αρχική κατάσταση στην κατάσταση στόχου

(ακολουθία τελεστών) ή η κατάσταση στόχου του προβλήματος. Υπάρχουν προβλήματα στα οποία η ακολουθία τελεστών είναι ο στόχος της λύσης. Σε αυτά, σε γενικές γραμμές, γνωρίζουμε τη λύση, αλλά δεν ξέρουμε πώς να την δημιουργήσουμε. Υπάρχουν επίσης προβλήματα στα οποία θέλουμε να ξέρουμε αν είναι δυνατόν να συνδυάσουμε τα στοιχεία του προβλήματος έτσι ώστε αυτό να πληροί ορισμένους περιορισμούς, αλλά ο τρόπος επίτευξής του δεν είναι σημαντικός.

Η επίλυση ενός προβλήματος που ορίζεται ως πρόβλημα χώρου καταστάσεων περνάει μέσα από την εξερεύνησή του χρησιμοποιώντας μια συστηματική στρατηγική. Πρέπει να ξεκινήσουμε από την αρχική κατάσταση που ορίζει το πρόβλημα, αξιολογώντας κάθε πιθανό βήμα και προχωρώντας μέχρι να βρούμε μια τελική κατάσταση. Στη χειρότερη περίπτωση, θα πρέπει να εξερευνήσουμε όλα τα πιθανά μονοπάτια από την αρχική κατάσταση μέχρι να φτάσουμε σε μια κατάσταση που να πληροί τις προϋποθέσεις της τελικής κατάστασης.

Ο τρόπος για να μειώσουμε τον χρόνο αναζήτησης σε κάτι λογικό είναι να προσπαθήσουμε να χρησιμοποιήσουμε τη γνώση σχετικά με το πρόβλημα που θέλουμε να λύσουμε μέσα στις λειτουργίες του αλγορίθμου αναζήτησης. Η κύρια δυσκολία είναι ότι αυτή η γνώση θα είναι ξεχωριστή για κάθε τομέα, επομένως δεν θα μπορεί να μεταφερθεί σε άλλους. Θα είναι μια λιγότερο γενική λύση, αλλά θα κερδίσουμε σε αποτελεσματικότητα.

Για να αναζητήσουμε τη βέλτιστη λύση, χρειαζόμαστε κάποιο μέτρο του κόστους απόκτησής της. Γενικά, θα έχουμε ένα κόστος που σχετίζεται με τους τελεστές που μας επιτρέπουν να μετακινηθούμε από τη μία κατάσταση στην άλλη, οπότε το κόστος αυτό θα έχει θεμελιώδη ρόλο στον υπολογισμό του κόστους της λύσης. Θα μετρήσουμε το κόστος των ενεργειών για το μονοπάτι που μας οδηγεί από την αρχική κατάσταση του προβλήματος στην τελική κατάσταση. Δεν μπορούν να τεθούν όλα τα προβλήματα με αυτόν τον τρόπο, όπως θα δούμε στις επόμενες ενότητες.

Δεδομένου ότι ο γράφος δημιουργείται καθώς τον εξερευνούμε, μπορούμε να δούμε ότι έχουμε δύο στοιχεία που παρεμβαίνουν στο κόστος της διαδρομής προς τη λύση που αναζητούμε. Πρώτον, θα έχουμε το κόστος του ήδη διανυθέντος μονοπατιού, το οποίο μπορούμε να υπολογίσουμε απλά προσθέτοντας το κόστος των τελεστών που εφαρμόστηκαν από την αρχική κατάσταση στον τρέχοντα κόμβο. Δεύτερον, θα έχουμε το πιο δύσκολο να προσδιοριστεί κόστος, δηλαδή αυτό για το μονοπάτι που πρέπει να ακολουθήσουμε ακόμη για να φτάσουμε στην τελική κατάσταση. Εφόσον δεν το γνωρίζουμε, θα πρέπει να χρησιμοποιήσουμε τη γνώση που έχουμε για το πρόβλημα για να λάβουμε μια προσέγγιση.

Προφανώς, η ποιότητα αυτής της γνώσης που μας επιτρέπει να προβλέψουμε το μελλοντικό κόστος θα κάνει την αναζήτησή μας περισσότερο ή λιγότερο επιτυχημένη. Αν η γνώση μας ήταν τέλεια, θα μπορούσαμε να προχωρήσουμε γρήγορα προς τον στόχο, απορρίπτοντας όλους τους δρόμους με μεγαλύτερο κόστος, στην περίπτωση αυτή, θα βρίσκαμε τη λύση μας σε γραμμικό χρόνο. Στον αντίποδα, αν είχαμε πλήρη άγνοια, θα έπρεπε να εξερευνήσουμε πολλούς δρόμους πριν βρούμε τη βέλτιστη λύση, όλους στη χειρότερη περίπτωση.

Επομένως, θα είναι θεμελιώδεις καθήκον για κάθε πρόβλημα, να ληφθεί μια συνάρτηση που υπολογίζει το μελλοντικό κόστος από μια κατάσταση στην κατάσταση λύσης. Όσο περισσότερο μπορούμε να την προσαρμόσουμε στο πραγματικό κόστος, τόσο καλύτερα θα λειτουργούν οι

αλγόριθμοι. Αυτή η συνάρτηση ονομάζεται ευρετική συνάρτηση. Οι αλγόριθμοι που χρησιμοποιούν αυτή την ευρετική πληροφορία για να καθοδηγήσουν την εξερεύνησή τους ονομάζονται ευρετικοί αλγόριθμοι αναζήτησης.

Μια πρώτη στρατηγική που μπορεί να μας έρθει στο μυαλό είναι να χρησιμοποιήσουμε την εκτίμηση του μελλοντικού κόστους από μια κατάσταση για να αποφασίσουμε ποιες καταστάσεις θα εξερευνήσουμε στη συνέχεια. Με αυτόν τον τρόπο, θα υποθέσουμε ότι οι καταστάσεις που είναι φαινομενικά πιο κοντά στη λύση θα είναι μέρος της διαδρομής προς τη βέλτιστη λύση και επομένως θα τη βρούμε νωρίτερα αν τις εξερευνήσουμε πρώτες. Αυτή θα ήταν μια καλή απόφαση αν γνωρίζαμε ότι οι πληροφορίες που διαθέτουμε είναι τέλειες. Συνήθως, αυτό δεν ισχύει.

Ο αλγόριθμος A^* είναι ένας ευρετικός αλγόριθμος αναζήτησης που χρησιμοποιεί αυτή την ιδέα. Αντί να χρησιμοποιεί μόνο την τιμή του μελλοντικού κόστους από μια κατάσταση προς τη λύση για να διατάξει τα πιθανά μονοπάτια προς εξερεύνηση, υπολογίζει το άθροισμα αυτού του κόστους και το κόστος του μονοπατιού από την αρχική κατάσταση στην τρέχουσα κατάσταση. Επομένως, σε κάθε κατάσταση που δεν έχει ακόμη εξερευνηθεί θα ανατεθεί μια τιμή ως συνάρτηση αυτών των δύο κοστών. Αυτός ο αλγόριθμος εγγυάται την επίτευξη της βέλτιστης λύσης όταν η συνάρτηση υποεκτιμά πάντα την πραγματική αξία του εναπομείναντος μονοπατιού. Όσο πιο κοντά είναι η τιμή στην πραγματική τιμή, τόσο πιο γρήγορα ο αλγόριθμος θα βρει τη βέλτιστη λύση. Εάν αυτή η συνθήκη δεν ισχύει, δεν υπάρχει καμία εγγύηση για την ποιότητα της λύσης. Το κύριο μειονέκτημα αυτού του αλγορίθμου είναι ότι αν η ευρετική συνάρτηση για το πρόβλημα απέχει πολύ από την πραγματική τιμή, ο αριθμός των μονοπατιών που αποθηκεύονται για μια πιθανή μελλοντική εξερεύνηση μπορεί να αυξηθεί εκθετικά, οπότε θα υπερβεί τυχόν περιορισμούς μνήμης για δύσκολα προβλήματα.

Ο αλγόριθμος IDA^* χρησιμοποιεί μια διαφορετική στρατηγική που επιτρέπει την επίλυση προβλημάτων με όχι τόσο καλή ευρετική. Εξερευνά το γράφημα αναζήτησης με βάση το επίπεδο χρησιμοποιώντας μια στρατηγική με βάση το βάθος, έτσι ώστε οι περιορισμοί μνήμης να είναι πάντα γραμμικοί με το βάθος του μονοπατιού που εξερευνάται. Το βάθος αντιστοιχεί σε αυτόν τον αλγόριθμο στην τιμή του εκτιμώμενου κόστους για το εξερευνημένο μονοπάτι.

iii. Αναζήτηση χώρου λύσεων

Οι αλγόριθμοι που είδαμε στην προηγούμενη ενότητα μας εξυπηρετούν κάθε φορά που μπορούμε να θέσουμε το πρόβλημα ως αναζήτηση μιας διαδρομής σε έναν χώρο καταστάσεων. Μπορεί όμως να συναντήσουμε προβλήματα στα οποία αυτή η προσέγγιση είναι πολύ τεχνητή, αφού δεν έχουμε πραγματικά τελεστές αλλαγής κατάστασης ή δεν υπάρχει πραγματικά ένας ορισμός του κόστους που να σχετίζεται με τους τελεστές του προβλήματος. Επίσης, υπάρχουν προβλήματα όπου η εύρεση της βέλτιστης λύσης είναι εκτός κάθε δυνατότητας, αφού το μέγεθος του χώρου αναζήτησης είναι πολύ μεγάλο. Επομένως, πρέπει να συμβιβαστούμε με μια λύση που μπορούμε να θεωρήσουμε καλή.

Η προσέγγιση για την επίλυση αυτών των προβλημάτων είναι παρόμοια με το χώρο καταστάσεων, έχουμε τα ακόλουθα στοιχεία:

1. Μια αρχική κατάσταση, η οποία σε αυτή την περίπτωση θα είναι μια πλήρης λύση. Αυτό θα μας παρουσιάσει το πρόσθετο πρόβλημα του πώς θα την βρούμε με σχετικά χαμηλό χρονικό κόστος.
2. Ένα σύνολο τελεστών, οι οποίοι σε αυτή την περίπτωση θα λειτουργούν με πλήρεις λύσεις και δεν θα έχουν σχετικό κόστος.
3. Μια συνάρτηση ποιότητας, η οποία θα μετράει πόσο καλή είναι μια λύση και θα μας επιτρέπει να καθοδηγούμε την αναζήτηση.

Το να γνωρίζουμε πότε έχουμε ολοκληρώσει την εξερεύνηση εξαρτάται αποκλειστικά από τη στρατηγική του αλγορίθμου, ο οποίος θα πρέπει να αποφασίσει πότε δεν είναι πλέον δυνατό να βρούμε μια καλύτερη λύση. Επομένως, δεν θα έχουμε μια καθορισμένη τελική κατάσταση. Οι αλγόριθμοι που λειτουργούν με αυτή την αναπαράσταση ονομάζονται αλγόριθμοι τοπικής αναζήτησης.

Μεταξύ αυτών των αλγορίθμων, ο πιο συχνά χρησιμοποιούμενος είναι η λεγόμενη αναζήτηση Hill-Climbing. Υπάρχουν διάφορες παραλλαγές αυτού του αλγορίθμου που έχουν τα πλεονεκτήματα και τα μειονεκτήματά τους. Για παράδειγμα, η απλή αναρρίχηση λόφου συνίσταται στο να επιλέγουμε πάντα τον πρώτο φορέα που αντιπροσωπεύει βελτίωση σε σχέση με την τρέχουσα λύση, οπότε δεν εξερευνούμε όλους τους προσβάσιμους γείτονες, εξοικονομώντας την εξερεύνηση ενός συγκεκριμένου αριθμού απογόνων. Το πλεονέκτημα είναι ότι είναι ταχύτερο από τον έλεγχο όλων των δυνατοτήτων, το μειονέκτημα είναι ότι υπάρχει μεγαλύτερη πιθανότητα να μην φτάσουμε στις καλύτερες λύσεις.

Ο αλγόριθμος που χρησιμοποιείται ευρύτερα είναι ο αλγόριθμος αναρρίχησης λόφου με την απότομη άνοδο. Αυτή η παραλλαγή επεκτείνει όλους τους πιθανούς απογόνους ενός κόμβου και επιλέγει αυτόν που αντιπροσωπεύει τη μεγαλύτερη βελτίωση σε σχέση με την τρέχουσα λύση. Με αυτή τη στρατηγική, υποθέτουμε ότι η καλύτερη λύση θα βρεθεί μέσω του διαδόχου που έχει τη μεγαλύτερη διαφορά ποιότητας με την τρέχουσα λύση, ακολουθώντας μια άπληστη πολιτική. Όπως θα δούμε αργότερα, αυτή η στρατηγική μπορεί να είναι επικίνδυνη.

Το κύριο πρόβλημα αυτού του αλγορίθμου απορρέει από τα πραγματικά χαρακτηριστικά της ευρετικής συνάρτησης που χρησιμοποιείται για την καθοδήγηση της αναζήτησης. Πρέπει να λάβουμε υπόψη ότι ο αλγόριθμος θα σταματήσει την αναζήτηση όταν δεν υπάρχει καλύτερη λύση από την τρέχουσα. Δεδομένου ότι δεν διατηρούμε μνήμη της διαδρομής που ακολουθήσαμε, θα είναι αδύνατο να επανεξετάσουμε τις αποφάσεις μας, οπότε αν έχουμε κάνει κάποιο λάθος, η λύση στην οποία θα καταλήξουμε πιθανόν να μην είναι η παγκόσμια βέλτιστη.

Αυτή η περίπτωση είναι η συνήθης, καθώς είναι πιθανό η ευρετική συνάρτηση που χρησιμοποιούμε να έχει πολλά τοπικά βέλτιστα και, ανάλογα με το από πού ξεκινάμε την αναζήτηση, θα καταλήξουμε να βρούμε το ένα ή το άλλο. Η περίπτωση αυτή μπορεί να βελτιωθεί εκτελώντας τον αλγόριθμο αρκετές φορές από διαφορετικά τυχαία επιλεγμένα σημεία και κρατώντας την καλύτερη λύση. Αυτή η στρατηγική ονομάζεται αναρρίχηση λόφου με τυχαία επανεκκίνηση. Πολλές φορές, αυτή η στρατηγική είναι πιο αποτελεσματική από άλλες πιο σύνθετες μεθόδους και είναι αρκετά φθηνή. Η μόνη επιπλοκή είναι η δυνατότητα δημιουργίας των διαφορετικών αρχικών λύσεων αναζήτησης, δεδομένου ότι μερικές φορές το πρόβλημα δεν επιτρέπει την εύκολη εύρεση αρχικών λύσεων.

Η στρατηγική αναζήτησης του αλγορίθμου αναρρίχησης λόφου και των παραλλαγών του είναι η απλούστερη που μπορεί να χρησιμοποιηθεί. Το μεγαλύτερο πρόβλημα είναι ότι είναι υπερβολικά αισιόδοξη και υποθέτει ότι υπάρχει μια λίγο πολύ άμεση (πάντα ανοδική) διαδρομή προς τα τοπικά βέλτιστα. Σε δύσκολα προβλήματα, αυτό δεν είναι απαραίτητο να ισχύει και απαιτούνται λίγο πιο εξαντλητικές στρατηγικές, οι οποίες εξερευνούν περισσότερους κόμβους από το άμεσο μονοπάτι και γνωρίζουν πώς να εκμεταλλευτούν τις πληροφορίες που βρίσκουμε στην πορεία.

Υπάρχουν διάφορες στρατηγικές που μπορούμε να προσθέσουμε στη βασική στρατηγική αναρρίχησης λόφων, οι οποίες μπορούν να βελτιώσουν σημαντικά τα αποτελέσματά της. Η πρώτη είναι να εξερευνήσουμε όχι μόνο τους καλύτερους γείτονες, αλλά και άλλους που μπορεί να είναι χειρότεροι, αλλά που μπορεί να βρίσκονται πίσω από ένα μονοπάτι που είναι καλύτερο από αυτό που μπορούμε να βρούμε αν ακολουθήσουμε απευθείας την ανοδική κλίση της ευρετικής συνάρτησης. Για να αποφασίσουμε ποιους χειρότερους κόμβους θα επισκεφτούμε χρειαζόμαστε μια στρατηγική απόφασης, μια δυνατότητα είναι αυτή που προτείνεται από τον αλγόριθμο προσομοιωμένης ανόπτωσης, ο οποίος χρησιμοποιεί μια αναλογία της φυσικής για να αποφασίσει πότε θα εξερευνήσει λύσεις που είναι χειρότερες από την τρέχουσα.

Πολλές από τις μεθόδους που ακολουθούν αυτή τη στρατηγική είναι εμπνευσμένες από βιολογικές ή φυσικές αναλογίες. Οι πιο γνωστοί είναι οι εξελικτικοί αλγόριθμοι, στους οποίους περιλαμβάνονται οι γενετικοί αλγόριθμοι, αλλά και άλλοι αλγόριθμοι που βασίζονται σε έννοιες εμπνευσμένες από την εξέλιξη των ειδών και την οικολογία. Μια άλλη οικογένεια αλγορίθμων που χρησιμοποιεί επίσης μια στρατηγική πληθυσμών είναι η λεγόμενη νοημοσύνη σμήνους. Αυτή περιλαμβάνει αλγορίθμους που είναι εμπνευσμένοι από τη βιολογία, όπως οι αλγόριθμοι βελτιστοποίησης σμήνους σωματιδίων. Η εφαρμογή όλων αυτών των αλγορίθμων έχει επεκταθεί πέρα από τα προβλήματα τεχνητής νοημοσύνης και εφαρμόζεται σε κάθε πρόβλημα που περιλαμβάνει τη βελτιστοποίηση μιας συνάρτησης και δεν μπορεί να επιλυθεί με κλασικές μεθόδους. Στην πραγματικότητα, όλοι αυτοί οι αλγόριθμοι περιλαμβάνονται σε έναν τομέα της επιστήμης των υπολογιστών που είναι γνωστός ως μεταερευνητικοί.

Προβλήματα ικανοποίησης περιορισμών

Υπάρχουν συγκεκριμένα προβλήματα που, λόγω των ιδιοτήτων τους, μπορούν να επιλυθούν ευκολότερα με αλγορίθμους προσαρμοσμένους σε αυτά παρά με γενικούς αλγορίθμους. Αυτή είναι η περίπτωση των προβλημάτων ικανοποίησης περιορισμών. Τα χαρακτηριστικά που ορίζουν ένα πρόβλημα ικανοποίησης περιορισμών είναι τα εξής:

- Το πρόβλημα μπορεί να αναπαρασταθεί με ένα σύνολο μεταβλητών.
- Κάθε μεταβλητή έχει ένα πεδίο τιμών (ένα πεπερασμένο σύνολο διακριτών τιμών ή διαστημάτων συνεχών τιμών)
- Περιορισμοί συνέπειας μεταξύ των μεταβλητών (δυναδικοί, ν-αριαίοι)
- Η λύση είναι μια ανάθεση τιμών σε αυτές τις μεταβλητές

Αυτό σημαίνει ότι κάθε πρόβλημα ικανοποίησης περιορισμών μπορεί να οριστεί από μια τριάδα στοιχείων $P = [X, D, C]$, όπου X είναι ένας κατάλογος μεταβλητών, D είναι ένας κατάλογος των

περιοχών που αντιστοιχούν στις μεταβλητές και C είναι ένας κατάλογος σχέσεων μεταξύ συνόλων μεταβλητών που καθορίζουν τους περιορισμούς που πρέπει να τηρεί το πρόβλημα.

Το ενδιαφέρον αυτού του τύπου προβλημάτων πηγάζει από τη στενή σχέση του με άλλες περιοχές, όπως η ικανοποιησιμότητα, η σχεσιακή άλγεβρα, η θεωρία βάσεων δεδομένων, η όραση ή οι γράφοι και ότι πολλά πραγματικά προβλήματα μπορούν να τεθούν ως προβλήματα ικανοποίησης περιορισμών, όπως ο προγραμματισμός εργασιών, η εφοδιαστική (τοποθεσία ή κατανομή πόρων, ανθρώπων, οχημάτων...), η διαχείριση δικτύων (ηλεκτρισμός, επικοινωνίες, ...), ο σχεδιασμός/διαμόρφωση, η αναγνώριση μορφής εικόνας, η συλλογιστική (χρονική, χωρική...). Οι εφαρμογές αυτές καθιστούν την περιοχή αυτή πολύ ενδιαφέρουσα από πρακτική άποψη.

Το πρόβλημα ικανοποίησης περιορισμών μπορεί να οριστεί ως γενικό πρόβλημα αναζήτησης. Εφόσον μια λύση είναι μια ανάθεση τιμών στις μεταβλητές του προβλήματος, θα ήταν απαραίτητο να απαριθμηθούν με κάποιο τρόπο όλες οι πιθανότητες μέχρι να βρεθεί μια που να ικανοποιεί τους περιορισμούς. Προφανώς, αυτή η προσέγγιση είναι ανέφικτη, δεδομένου ότι το σύνολο των πιθανών αναθέσεων προς διερεύνηση είναι πολύ μεγάλο.

Η πρώτη προσέγγιση θα βασίζεται σε έναν αλγόριθμο τυφλής αναζήτησης. Θα ορίσουμε το πρόβλημα έτσι ώστε να μπορούμε να εξερευνήσουμε το σύνολο των πιθανών αναθέσεων τιμών στις μεταβλητές με μια στρατηγική αναζήτησης σε βάθος. Μπορούμε να καθορίσουμε μια σταθερή σειρά για τις μεταβλητές και να εξερευνήσουμε συστηματικά το χώρο των μερικών λύσεων, αναθέτοντας διαδοχικά τιμές στις μεταβλητές με τη σειρά που έχουμε ορίσει. Τη στιγμή που μια μερική λύση παραβιάζει οποιονδήποτε από τους περιορισμούς του προβλήματος, γνωρίζουμε ότι δεν θα αποτελέσει μέρος οποιασδήποτε πλήρους λύσης, οπότε θα επαναλάβουμε τις προηγούμενες αναθέσεις που έχουμε κάνει. Αυτός ο τύπος αναζήτησης ονομάζεται χρονολογική αναζήτηση με οπισθοδρόμηση (backtracking search).

Άλλοι τρόποι για την αντιμετώπιση αυτού του τύπου προβλημάτων είναι οι λεγόμενες τεχνικές διάδοσης περιορισμών. Αυτές αποσκοπούν στη μείωση του αριθμού των συνδυασμών που πρέπει να ελεγχθούν, απορρίπτοντας όλους εκείνους που περιέχουν αναθέσεις τιμών που γνωρίζουμε εκ των προτέρων ότι δεν μπορούν να εμφανιστούν σε καμία λύση. Οι τεχνικές αυτές έχουν αναπτυχθεί από ευρετικές τεχνικές που προέρχονται από ιδιότητες των γραφημάτων περιορισμών. Οι πιο συχνά χρησιμοποιούμενες είναι αυτές που προέρχονται από την ιδιότητα που ονομάζεται k -συνέπεια. Η ιδιότητα αυτή μπορεί να οριστεί για οποιαδήποτε τιμή του k , αλλά συνήθως χρησιμοποιείται η περίπτωση του k ίσου με 2, η οποία στην περίπτωση αυτή ονομάζεται arc-consistency. Θα λέμε ότι ένα γράφημα περιορισμών είναι arc-consistency, αν για κάθε ζεύγος μεταβλητών στο γράφημα, για κάθε μία από τις τιμές της μίας μεταβλητής, μπορούμε να βρούμε μια άλλη τιμή της άλλης μεταβλητής που δεν παραβιάζει τους μεταξύ τους περιορισμούς.

Υπάρχουν διαφορετικοί αλγόριθμοι που συνδυάζουν την αναζήτηση και τη διάδοση, ανάλογα με το είδος της συνέπειας που αξιολογείται σε κάθε βήμα αναζήτησης. Ο απλούστερος αλγόριθμος ονομάζεται έλεγχος προς τα εμπρός. Αυτός ο αλγόριθμος προσθέτει έναν έλεγχο συνέπειας τόξου μεταξύ των μελλοντικών μεταβλητών και της τρέχουσας μεταβλητής για κάθε ανάθεση. Αυτός ο έλεγχος συνέπειας τόξου έχει ως αποτέλεσμα να εξαλείφονται όλες οι τιμές των μελλοντικών

μεταβλητών που δεν είναι συνεπείς με την ανάθεση στην τρέχουσα μεταβλητή. Εάν κάποιο πεδίο οποιασδήποτε μελλοντικής μεταβλητής μείνει κενό, μπορούμε να επανεξετάσουμε την τρέχουσα ανάθεση ή να επιστρέψουμε πίσω.

Περιπτώσεις χρήσης

ΣΥΣΤΗΜΑΤΑ ΠΛΟΗΓΗΣΗΣ ΑΥΤΟΚΙΝΗΤΩΝ

Οι ρομποτικές συσκευές είναι ο τέλειος συνδυασμός πολλαπλών τεχνικών τεχνητής νοημοσύνης. Ένα παράδειγμα είναι οι ρομποτικές ηλεκτρικές σκούπες. Οι εργασίες που εκτελούν αυτά τα ρομπότ είναι φαινομενικά απλές, αλλά χρειάζονται πολύπλοκους αλγορίθμους τεχνητής νοημοσύνης. Ένα από τα πράγματα που πρέπει να κάνουν αυτά τα ρομπότ είναι να χαρτογραφούν τα δωμάτια και να βρίσκουν τη θέση τους μέσα σε αυτά. Αυτό μπορεί να γίνει χρησιμοποιώντας διάφορους διαφορετικούς αισθητήρες. Μια δυνατότητα είναι να λαμβάνονται φωτογραφίες του δωματίου και να εξάγονται συγκεκριμένα χαρακτηριστικά που επιτρέπουν στο ρομπότ να βρίσκει ορόσημα που μπορούν να αναγνωριστούν, όπως για παράδειγμα τραπέζια, καρέκλες ή καναπέδες. Αυτή η εργασία χρειάζεται αλγορίθμους υπολογιστικής όρασης για την εξαγωγή των χαρακτηριστικών από τις εικόνες. Αυτό θα χρησιμοποιηθεί για να αποφασιστεί ποια χαρακτηριστικά αντιστοιχούν σε ορόσημα και τα οποία θα αποθηκευτούν για να δημιουργηθεί ένας χάρτης για το δωμάτιο ή τα δωμάτια που έχουν επισκεφθεί κατά τη διάρκεια του καθαρισμού. Το ρομπότ επαναλαμβάνει αυτή την εργασία περιοδικά για να ενημερώνει τον χάρτη και να ενημερώνει τη θέση του. Το ρομπότ πρέπει επίσης να αποφασίσει πότε έχει περάσει από το ένα δωμάτιο στο άλλο.

Άλλοι αισθητήρες μπορούν να χρησιμοποιηθούν για την ανίχνευση της εγγύτητας με άλλα αντικείμενα που δεν μπορούν να ανιχνευθούν με τη χρήση της όρασης. Τα δωμάτια δεν παραμένουν στατικά και τα αντικείμενα που δεν ανιχνεύονται με τη χρήση όρασης πρέπει επίσης να αποφεύγονται. Μερικές φορές πρέπει να ανιχνεύονται και να αποφεύγονται ακόμη και κινούμενα αντικείμενα, όπως για παράδειγμα κατοικίδια ζώα. Αυτοί οι αισθητήρες παρέχουν συνεχώς πληροφορίες στο ρομπότ, το οποίο πρέπει να αποφασίσει το σχέδιο που απαιτείται για την αποφυγή του αντικειμένου. Άλλοι αισθητήρες ανιχνεύουν επίσης την επιφάνεια που καθαρίζεται, για να αποφασίσουν για παράδειγμα ότι βρίσκεται πάνω από ένα χαλί και ότι οι κινήσεις και ο καθαρισμός πρέπει να είναι διαφορετικοί ή ότι το ρομπότ πρόκειται να κινηθεί προς ένα χαλί, οπότε η ταχύτητα που απαιτείται για να ανέβει στο χαλί πρέπει να αυξηθεί. Πρέπει επίσης να ανιχνεύονται πιο συγκεκριμένα πράγματα, για παράδειγμα ότι το ρομπότ βρίσκεται μπροστά από μια σκάλα και ότι η κίνηση προς τα εμπρός σημαίνει πτώση προς τα κάτω.

Χρησιμοποιώντας τις πληροφορίες εντοπισμού πρέπει επίσης να ληφθεί ένα σχέδιο για τον καθαρισμό των δωματίων. Αυτό χρειάζεται αλγορίθμους σχεδιασμού που θα αποφασίσουν μια τροχιά μέσα σε κάθε δωμάτιο για να διασφαλίσουν ότι καθαρίζεται ολόκληρη η περιοχή ή ότι ειδικά σημεία μέσα στο δωμάτιο πρέπει να προσεγγιστούν με διαφορετικό τρόπο, όπως κάτω από έναν καναπέ, ή ότι οι γωνίες του δωματίου χρειάζονται διαφορετική ταχύτητα ή περισσότερο χρόνο για να καθαριστούν. Όλες αυτές οι αποφάσεις συνδυάζονται με τον έλεγχο των ενεργοποιητών του ρομπότ, των τροχών και των στοιχείων καθαρισμού, λαμβάνοντας αποφάσεις για τον τρόπο ελιγμών μέσα στο δωμάτιο ή τη διέλευση από το ένα δωμάτιο στο άλλο. Το σχέδιο καθαρισμού πρέπει επίσης να λαμβάνει υπόψη την εσωτερική κατάσταση του ρομπότ, όπως αν

είναι γεμάτο, οπότε δεν μπορεί να καθαρίσει άλλο, μετακινούμενο ενδεχομένως προς το σταθμό βάσης για να πετάξει όλη τη βρωμιά που έχει συλλέξει, ή ότι η μπαταρία είναι χαμηλή, οπότε πρέπει επίσης να μετακινηθεί προς το σταθμό βάσης για να φορτίσει πριν συνεχίσει την εργασία καθαρισμού. Όλες αυτές οι εργασίες σχεδιασμού χρειάζονται κάποιες τεχνικές αναπαράστασης γνώσης, ώστε ορισμένες αποφάσεις να βασίζονται σε κανόνες και συλλογισμούς.

Οι σύγχρονες ρομποτικές ηλεκτρικές σκούπες εξελίσσονται γρήγορα, οπότε εκτός από την εισαγωγή πιο σύγχρονων τεχνικών τεχνητής νοημοσύνης, προχωρούν και σε άλλες παρόμοιες εργασίες, όπως για παράδειγμα το κούρεμα του γκαζόν ή ο καθαρισμός της πισίνας.

<https://www.youtube.com/watch?v=YVG5--9Sjro>

Αναφορές

- Artificial intelligence: a modern approach-Russell, S.J., Norvig, P, Prentice-Hall, (τέταρτη έκδοση) 2020. ISBN: 9781292153964 (Κεφάλαια 3, 4 και 6)Τεχνητή νοημοσύνη: μια νέα σύνθεση-Nilsson, N.J, Morgan Kaufmann Publishers, 1998. ISBN: 1558604677 (Κεφάλαια 7, 8, 9 και 11)
- Τεχνητή νοημοσύνη: δομές και στρατηγικές για την επίλυση σύνθετων προβλημάτων-Luger, G.F, Pearson Education: Addison Wesley, 2009. ISBN: 9780321545893 (Κεφάλαια 3, 4)
- Ευρετική αναζήτηση: Schrödl, S., Elsevier, 2012. ISBN 978-0-12-372512-7

Μάθημα 1.3: Αναπαράσταση γνώσης

Θέματα προς συζήτηση

- Εισαγωγή στην αναπαράσταση γνώσης
- Μαθηματική λογική
- Συστήματα κανόνων παραγωγής
- Δομημένες αναπαραστάσεις
- Οντολογίες
- Αβέβαιη γνώση και συλλογισμός

Σημειώσεις διάλεξης

- **Εισαγωγή**

Στο προηγούμενο μάθημα, μόλις μιλήσαμε για αλγόριθμους που είναι σε θέση να επιλύουν προβλήματα εξερευνώντας το χώρο των δυνατοτήτων. Αυτοί οι αλγόριθμοι χρησιμοποιούν ελάχιστες πληροφορίες για να βρουν μια λύση. Όχι μόνο όσον αφορά τα χαρακτηριστικά που χρησιμοποιούνται για την αναπαράσταση του προβλήματος αλλά και τη γνώση που εμπλέκεται στη λήψη αποφάσεων κατά τη διάρκεια της εξερεύνησης. Οι ευρετικές συναρτήσεις που χρησιμοποιούνται για την αξιολόγηση κάθε εναλλακτικής λύσης του προβλήματος τις αντιμετωπίζουν όλες με τον ίδιο τρόπο. Προφανώς, κατά τη διάρκεια της επίλυσης ενός προβλήματος, οι καταστάσεις που διερευνούμε δεν χρειάζεται πάντα να αξιολογούνται εξίσου, ούτε όλες οι πληροφορίες που χρησιμοποιούμε έχουν πάντα την ίδια σημασία.

Δεν μπορούμε να πάρουμε αυτές τις αποφάσεις αν δεν έχουμε βαθιά κατανόηση των χαρακτηριστικών του προβλήματος και του τρόπου με τον οποίο μπορούν να μας βοηθήσουν να πάρουμε αποφάσεις. Αυτό μας οδηγεί στο συμπέρασμα ότι σε κάθε σύνθετο πρόβλημα στην Τεχνητή Νοημοσύνη, πρέπει να αναρωτηθούμε τι γνώσεις μπορεί να χρειαστούμε για την επίλυσή του και πώς θα πρέπει να τις χρησιμοποιήσουμε για να κάνουμε την επίλυσή του πιο αποτελεσματική.

Η ανάγκη εισαγωγής περισσότερης γνώσης στην επίλυση προβλημάτων μας οδηγεί στο να θέσουμε στον εαυτό μας δύο ερωτήματα. Πρώτον, πώς να επιλέξουμε τον φορμαλισμό αναπαράστασης που μας επιτρέπει να μεταφράσουμε εύκολα τον πραγματικό κόσμο σε μια αναπαράσταση. Δεύτερον, πώς πρέπει να είναι αυτή η αναπαράσταση ώστε να χρησιμοποιείται αποτελεσματικά. Αυτή η γνώση θα πρέπει να χρησιμοποιηθεί στη διαδικασία λήψης αποφάσεων, θα πρέπει να αξιολογήσουμε πότε είναι απαραίτητο να τη χρησιμοποιήσουμε και τι μπορούμε να αποκομίσουμε από αυτήν.

Στο σημείο αυτό, πρέπει να κάνουμε διάκριση μεταξύ πληροφοριών και γνώσεων. Θα αποκαλούμε πληροφορία το σύνολο των βασικών δεδομένων, χωρίς ερμηνεία, που λαμβάνονται ως είσοδος του συστήματος. Τα δεδομένα αυτά δεν μας λένε τίποτα αν δεν τα συνδέσουμε με τη σημασία τους στο πρόβλημα. Γνώση θα ονομάσουμε τα δεδομένα πρώτης τάξης, τα οποία μοντελοποιούν την εμπειρογνωμοσύνη ενός συγκεκριμένου τομέα με δομημένο τρόπο, ή τα οποία λαμβάνονται από την ερμηνεία των βασικών δεδομένων.

Ως μηχανισμός αναπαράστασης της γνώσης, θα μιλήσουμε για σχήματα αναπαράστασης γνώσης, τα οποία θα αποτελέσουν ένα μέσο για την κωδικοποίηση της πραγματικότητας σε έναν υπολογιστή. Από την άποψη του υπολογιστή, ένα σχήμα αναπαράστασης μπορεί να περιγραφεί ως συνδυασμός ενός στατικού μέρους, δομών δεδομένων που κωδικοποιούν το πρόβλημα που επιλύει ο πράκτορας και ενός δυναμικού μέρους, δομών δεδομένων που αποθηκεύουν τη γνώση σχετικά με το περιβάλλον στο οποίο επιλύεται το πρόβλημα και διαδικασιών που χειρίζονται τις δομές.

Η γνώση που μπορούμε να αναπαραστήσουμε σε έναν τομέα μπορεί να χωριστεί σε δύο τύπους. Υπάρχει η δηλωτική γνώση που είναι μια γνώση που αναπαρίσταται ανεξάρτητα από τη χρήση της, δηλαδή δεν επιβάλλει συγκεκριμένο μηχανισμό συλλογισμού. Επομένως, μπορούμε να αποφασίσουμε πώς θα τη χρησιμοποιήσουμε ανάλογα με το πρόβλημα που πρόκειται να λύσουμε με αυτήν. Οι μηχανισμοί συλλογισμού που χρησιμοποιούνται σε αυτόν τον τύπο γνώσης είναι γενικοί και η αποτελεσματικότητα στη χρήση τους θα εξαρτηθεί από την ενσωμάτωση γνώσης ελέγχου που επιτρέπει την εστίαση της χρήσης της. Οι τύποι δηλωτικής γνώσης είναι η σχεσιακή γνώση, η οποία αναπαριστά τον τρόπο με τον οποίο διαφορετικές έννοιες σχετίζονται μεταξύ τους και τις ιδιότητες που μπορούν να προκύψουν μέσω αυτών των σχέσεων, η κληρονομική γνώση, η οποία καθορίζει τις δομικές σχέσεις μεταξύ των εννοιών, έτσι ώστε να μπορούμε να ορίσουμε ιδιότητες γενίκευσης/ειδίκευσης, συμπερίληψης ή κληρονομικότητας ιδιοτήτων και η επαγωγική γνώση, η οποία καθορίζει τον τρόπο με τον οποίο ιδιότητες και γεγονότα μπορούν να προκύψουν από άλλες μέσω σχέσεων συναγωγής.

Ο δεύτερος τύπος γνώσης είναι η διαδικαστική γνώση. Αυτή υποδεικνύει ρητά τον τρόπο χρήσης της, οπότε ο μηχανισμός συλλογισμού συνδέεται με την αναπαράσταση, με το πλεονέκτημα ότι είναι πιο αποτελεσματική η χρήση της.

- **Προτασιακή και κατηγορηματική λογική**

Η λογική είναι ένα από τα θεμέλια της αναπαράστασης και της συλλογιστικής στην τεχνητή νοημοσύνη και οι περισσότεροι φορμαλισμοί που χρησιμοποιούνται σε εφαρμογές που δεν περιλαμβάνουν αβεβαιότητα είναι μια περιορισμένη υπογλώσσα της συμβολικής λογικής. Μέσω αυτής της γλώσσας, μπορούμε να εκφράσουμε προτάσεις που συσχετίζουν οποιαδήποτε ατομική δηλωτική φράση μέσω των λογικών συνδέσμων και της σχετικής σημασιολογίας τους. Αυτή η σημασιολογία ορίζεται από αυτό που ονομάζεται θεωρία μοντέλων, η οποία καθορίζει τη σχέση μεταξύ των τύπων και των τιμών της αλήθειας και του ψεύδους, είναι αυτό που ονομάζεται ερμηνεία. Κάθε συνδυασμός σύνδεσμος έχει τον δικό του πίνακα αλήθειας που καθορίζει πώς συνδυάζονται οι τιμές αλήθειας των ατόμων για να προκύψει η τιμή αλήθειας του τύπου.

Από τις δηλώσεις που εκφράζονται μέσω αυτής της γλώσσας, μπορούν να προκύψουν νέες δηλώσεις που συνάγονται από αυτές. Για το σκοπό αυτό, υπάρχουν διάφορες μεθοδολογίες επικύρωσης συλλογισμών που καθορίζουν τα βήματα για την επίλυση αυτών των ερωτημάτων. Υπάρχουν αρκετές μεθοδολογίες επικύρωσης συλλογισμών και μπορούν να χωριστούν σε δύο ομάδες. Αυτές που βασίζονται στη σημασιολογία επιδιώκουν να αποδείξουν ότι τα μοντέλα που κάνουν ένα σύνολο δηλώσεων αληθές περιλαμβάνουν τα μοντέλα που κάνουν αληθές το συμπέρασμα που θέλουμε να αποδείξουμε. Αυτές που βασίζονται στο συντακτικό εκμεταλλεύονται την αλγεβρική δομή της γλώσσας της λογικής (άλγεβρα Boole) για να αποδείξουν ότι η δήλωση αληθούς/λάθους είναι προσβάσιμη χρησιμοποιώντας τις προϋποθέσεις και το συμπέρασμα ενός συλλογισμού.

- **Συστήματα κανόνων παραγωγής**

Η προτασιακή λογική μπορεί επίσης να θεωρηθεί ως μηχανισμός περιγραφής αλγορίθμων. Μπορούμε να περιγράψουμε ένα πρόβλημα σαν να ήταν μια διαδικασία συλλογισμού, μια απόδειξη. Τα κατηγορήματα που χρησιμοποιούμε αναλύουν το πρόβλημα σε απλούστερα προβλήματα και ορίζουν μια ολική ή μερική σειρά μεταξύ τους, επιτρέποντας τη δημιουργία συστημάτων που εκτελούν την περιγραφή ενός προβλήματος για την επίλυσή του. Σε αυτό το είδος αναπαράστασης, χρησιμοποιούμε ένα περιορισμένο σύνολο εκφράσεων που μπορούν να εκφραστούν χρησιμοποιώντας τη γλώσσα της λογικής. Επιτρέπονται μόνο καθολικά ποσοτικοποιημένοι λογικοί τύποι που μπορούν να μετατραπούν σε διαζευκτικές προτάσεις. Αυτός ο τύπος μορφών αντιστοιχεί σε αυτό που θα ονομάσουμε κανόνες παραγωγής.

Χρησιμοποιώντας αυτόν τον φορμαλισμό, μπορούμε να περιγράψουμε τον τρόπο επίλυσης ενός προβλήματος, δηλώνοντας το σύνολο των κανόνων από τους οποίους μπορεί να προκύψει μια λύση και χρησιμοποιώντας τη λογική. Η διαφορά μεταξύ των κανόνων παραγωγής και άλλων διαδικαστικών φορμαλισμών (όπως οι προστακτικές γλώσσες προγραμματισμού) είναι ότι δεν εξηγούμε άμεσα τον αλγόριθμο που πρέπει να χρησιμοποιηθεί για την εύρεση της λύσης του προβλήματος, αλλά υποδεικνύουμε τις συνθήκες που πρέπει να πληροί. Ένας μηχανισμός συλλογισμού είναι υπεύθυνος για την εύρεσή της συνδυάζοντας τα βήματα που υποδεικνύονται από τους κανόνες.

Αυτός ο τρόπος περιγραφής των διαδικασιών εμπίπτει στις λεγόμενες δηλωτικές γλώσσες. Το πλεονέκτημα της δήλωσης των συνθηκών που πληρούν οι λύσεις είναι ότι δεν περιοριζόμαστε σε μία μόνο ακολουθία εκτέλεσης. Ένα σύνολο κανόνων μπορεί να περιγράφει πολλούς διαφορετικούς αλγορίθμους που μοιράζονται μέρος των προδιαγραφών τους. Αυτού του είδους οι γλώσσες επιτρέπουν επίσης την έκφραση των προγραμμάτων σε υψηλότερο επίπεδο αφαίρεσης, αφού δεν χρειάζεται να ορίσουμε ακριβώς τον τρόπο με τον οποίο πρέπει να επιτευχθεί η λύση, παρά μόνο τις συνθήκες της. Η εύρεση της λύσης που πληροί τις συνθήκες ανατίθεται στον μηχανισμό συλλογισμού που χρησιμοποιούμε.

Για να θέσουμε αυτόν τον φορμαλισμό σε προοπτική, μπορούμε να κάνουμε την αναλογία με τους αλγορίθμους αναζήτησης που είδαμε στο προηγούμενο μάθημα. Οι κανόνες είναι ισοδύναμοι με τους τελεστές αναζήτησης και ο αλγόριθμος αναζήτησης που χρησιμοποιείται είναι ο μηχανισμός επικύρωσης. Ένας σχετικά μικρός αριθμός τελεστών επιτρέπει τη δημιουργία λύσεων για πολύ

διαφορετικά προβλήματα, οι οποίες δημιουργούνται όλες από το συνδυασμό αυτών των κανόνων. Το σύνολο των κανόνων ονομάζεται βάση γνώσης. Ένα πρόβλημα ορίζεται με τη χρήση γεγονότων. Τα γεγονότα αυτά θα είναι η περιγραφή των χαρακτηριστικών και των συνθηκών του προβλήματος και θα αποτελούνται από πρωταρχικά κατηγορήματα, συναρτήσεις, σχέσεις και σταθερές που ορίζονται στο πρόβλημα.

Ο τρόπος με τον οποίο θα προσδιορίσουμε αν υπάρχει και ποια είναι η απάντηση στο πρόβλημα, που θέτουμε, θα εξαρτηθεί από τη μέθοδο συλλογισμού που χρησιμοποιούμε για να επικυρώσουμε το πρόβλημα χρησιμοποιώντας τη βάση γνώσης και τη βάση γεγονότων. Αυτός ο μηχανισμός θα υλοποιηθεί μέσω αυτού που θα ονομάσουμε μηχανή εξαγωγής συμπερασμάτων.

- **Δομημένες αναπαραστάσεις**

Η γλώσσα της λογικής θα πρέπει να μας επιτρέπει να αναπαραστήσουμε οποιαδήποτε γνώση που θα θέλαμε να χρησιμοποιήσουμε σε ένα πρόβλημα. Ωστόσο, η ικανότητά τους να εκφράζουν τη γνώση παρεμποδίζεται από τη δυσκολία χρήσης της τυπικής τους γλώσσας. Θα μπορούσαμε να συγκρίνουμε αυτή τη δυσκολία με τη διαφορά μεταξύ του προγραμματισμού σε κώδικα μηχανής ή της χρήσης μιας γλώσσας προγραμματισμού υψηλού επιπέδου. Η γλώσσα της λογικής είναι πολύ χαμηλού επιπέδου για να επιτρέψει την τυποποίηση των μεγάλων ποσοτήτων γνώσης που απαιτούνται για τις πρακτικές εφαρμογές. Αυτός είναι ο λόγος για τον οποίο δημιουργήθηκαν οι δομημένες αναπαραστάσεις ως γλώσσες που είναι πιο κοντά στον τρόπο με τον οποίο συνηθίζουμε να χρησιμοποιούμε και να περιγράφουμε τη γνώση.

Τα σημασιολογικά δίκτυα είναι ένας φορμαλισμός για την αναπαράσταση της γνώσης όπου η γνώση αναπαρίσταται ως γράφος. Οι κόμβοι αντιστοιχούν σε έννοιες και τα τόξα σε σχέσεις μεταξύ εννοιών. Η αιτιολόγηση για τη χρήση αυτού του τύπου αναπαράστασης για την τυποποίηση της γνώσης προέρχεται από τη γνωστική ψυχολογία, η οποία είναι ο τομέας όπου τα σημασιολογικά δίκτυα ορίστηκαν για πρώτη φορά. Σύμφωνα με αυτόν τον τομέα, η αναπαράσταση που χρησιμοποιούν οι άνθρωποι για να αποθηκεύουν και να ανακτούν τη γνώση βασίζεται στη συσχέτιση μεταξύ των εννοιών μέσω των σχέσεών τους. Τα πλαίσια εξελίχθηκαν από τα σημασιολογικά δίκτυα. Δομούν τον φορμαλισμό των σημασιολογικών δικτύων και επιτρέπουν τον λεπτομερή ορισμό της συμπεριφοράς των στοιχείων που ορίζονται. Από τη μία πλευρά, μια έννοια (πλαίσιο) είναι μια συλλογή χαρακτηριστικών που την ορίζουν και αυτά έχουν μια σειρά από χαρακτηριστικά και περιορισμούς που καθορίζουν τη σημασιολογία της. Μια σχέση συνδέει τις έννοιες μεταξύ τους και έχει επίσης μια σειρά χαρακτηριστικών που καθορίζουν τη συμπεριφορά της. Η σημασιολογία αυτών των φορμαλισμών αντιστοιχεί σε ορισμένες περιορισμένες υπογλώσσες της λογικής.

Οι γλώσσες πλαισίου έχουν εξελιχθεί σε αυτό που είναι γνωστό ως γλώσσες οντολογίας. Μια οντολογία μπορεί να οριστεί επίσημα ως "ένας κατάλογος των τύπων των πραγμάτων που υποθέτουμε ότι υπάρχουν σε ένα πεδίο D από τη σκοπιά κάποιου που χρησιμοποιεί μια γλώσσα L με σκοπό να μιλήσει για το D. Τα στοιχεία μιας οντολογίας αντιπροσωπεύουν κατηγορήματα, σταθερές, έννοιες και σχέσεις που ανήκουν σε μια γλώσσα L όταν χρησιμοποιείται για να επικοινωνήσει πληροφορίες σχετικά με το D".

Οι οντολογίες απέκτησαν σημασία λόγω του έργου της Κοινοπραξίας του Παγκόσμιου Ιστού (W3C), του Σημασιολογικού Ιστού. Το έργο αυτό φιλοδοξεί να καταστήσει όλη τη γνώση στο WWW διαθέσιμη σε οποιαδήποτε εφαρμογή υπολογιστή σε μορφή αναγνώσιμη από μηχανήματα. Ο στόχος είναι να σχολιαστούν όλες οι υπάρχουσες ιστοσελίδες χρησιμοποιώντας γλώσσες οντολογιών και να δημιουργηθούν πρότυπες οντολογίες και αποθήκες γνώσης. Το έργο αυτό σχετίζεται με την πρωτοβουλία Open Linked Data για τη διάθεση προς δημόσια χρήση όλης της γνώσης που παράγουν οι οργανισμοί. Το W3C έχει τυποποιήσει διάφορες γλώσσες για την αναπαράσταση γνώσης όπως οι RDF, RDFS και OWL και διάφορα λεξιλόγια οντολογιών για κοινούς τομείς.

Στην Τεχνητή Νοημοσύνη, μια οντολογία θα είναι μια ρητή τυπική περιγραφή των εννοιών ενός τομέα (κλάσεις). Αυτές οι κλάσεις θα περιγράφονται από ιδιότητες που θα αντιπροσωπεύουν τα χαρακτηριστικά, τις ιδιότητες και τις σχέσεις των κλάσεων. Επιπλέον, αυτά τα χαρακτηριστικά θα έχουν περιορισμούς (τύπος, **καρτελικότητα**, ...). Τέλος, θα έχουμε περιπτώσεις (αναγνωρίσιμα στοιχεία) που θα αποτελούν τα συγκεκριμένα άτομα που αναπαριστά η οντολογία.

Η ανάπτυξη μιας οντολογίας απαιτεί τον ορισμό των κλάσεων που συνθέτουν το πεδίο, την οργάνωση των κλάσεων σε μια ταξινομική ιεραρχία σύμφωνα με τις συγγενειές τους, τον ορισμό των ιδιοτήτων κάθε κλάσης και την υπόδειξη των περιορισμών των τιμών τους και την ανάθεση τιμών στις ιδιότητες για τη δημιουργία στιγμιότυπων. Ο τρόπος με τον οποίο γίνεται αυτή η διαδικασία περιγράφεται σε διάφορες μεθοδολογίες ανάπτυξης οντολογιών που υπάρχουν στη βιβλιογραφία για την αναπαράσταση γνώσης. Οι μεθοδολογίες αυτές είναι αρκετά πολύπλοκες, αφού από μόνη της η ανάπτυξη μιας οντολογίας για έναν πραγματικό τομέα είναι ένα πολύπλοκο έργο. Μια απλή διαδικασία για τη δημιουργία μιας οντολογίας ακολουθεί τα εξής βήματα:

1. Καθορισμός του τομέα και της εμβέλειας της οντολογίας
2. Εξέταση της επαναχρησιμοποίησης υφιστάμενων οντολογιών
3. Καταγραφή των σημαντικών όρων της οντολογίας
4. Ορισμός των κλάσεων και της ιεραρχίας τους
5. Ορισμός των ιδιοτήτων των κλάσεων
6. Ορισμός των χαρακτηριστικών των ιδιοτήτων
7. Δημιουργία παραδειγμάτων

Αβέβαιη γνώση και συλλογιστική

Γενικά, η γνώση που χρησιμοποιείται από τις εφαρμογές τεχνητής νοημοσύνης και ιδίως από τα συστήματα βασισμένα στη γνώση δεν είναι πάντα ακριβής και πλήρης. Στην πράξη, αντιμετωπίζουμε προβλήματα όπως:

- Η αναπαράσταση της γνώσης για την κάλυψη όλων των γεγονότων που σχετίζονται με ένα πρόβλημα είναι πολύ δύσκολη.
- Για ορισμένους τομείς, όλα τα γεγονότα και οι κανόνες που είναι απαραίτητοι για την επίλυση του προβλήματος δεν ορίζονται με ακρίβεια.
- Υπάρχουν προβλήματα στα οποία, ακόμη και αν έχουμε τους κανόνες για την επίλυσή τους, δεν έχουμε όλες τις απαραίτητες πληροφορίες για την εφαρμογή τους.

Αυτό σημαίνει ότι για να επιχειρηματολογήσουμε σε αυτούς τους τομείς θα πρέπει να χρησιμοποιήσουμε πιο ισχυρά εργαλεία από αυτά που παρέχει η κλασική λογική, η οποία θα μας επέτρεπε να εργαστούμε μόνο με γνώση που μπορούμε να ορίσουμε ως αληθή ή ψευδή.

Στην πραγματικότητα, αυτό δεν είναι παράλογο, δεδομένου ότι οι άνθρωποι μπορούν να λαμβάνουν αποφάσεις όταν έρχονται αντιμέτωποι με ελλειπείς ή ανακριβείς πληροφορίες και ότι η ανακρίβεια ή η έλλειψη γνώσης δεν τους εμποδίζει να βγάλουν συμπεράσματα. Είναι σαφές ότι αν θέλουμε τα ευφυή προγράμματα να μιμηθούν την ικανότητα των ανθρώπων, πρέπει να τα εξοπλίσουμε με μηχανισμούς που είναι ικανοί να αντιμετωπίσουν αυτό το πρόβλημα.

Η αντιμετώπιση αυτού του προβλήματος έχει οδηγήσει στην ανάπτυξη μιας σειράς λογικών και μοντέλων που προσπαθούν να αντιμετωπίσουν το πρόβλημα της ελλιπούς και ανακριβούς γνώσης από διαφορετικές οπτικές γωνίες και έτσι να μοντελοποιήσουν τις διαδικασίες συλλογισμού που εφαρμόζουν οι άνθρωποι. Υπάρχουν πολλές προτάσεις που έχουν αναπτυχθεί καθ' όλη τη διάρκεια της εξέλιξης της τεχνητής νοημοσύνης, εμείς θα επικεντρωθούμε μόνο στα πιθανοτικά μοντέλα και συγκεκριμένα στα Μπεϋζιανά Δίκτυα.

Τα πιθανοτικά μοντέλα βασίζονται όλα στα στοιχεία της θεωρίας πιθανοτήτων. Οι κατανομές πιθανοτήτων μοντελοποιούν την πεποίθηση σχετικά με την αλήθεια ή το ψεύδος γεγονότων. Μπορούμε να αποδώσουμε τιμές πιθανότητας στα γεγονότα που έχουμε και να τις χρησιμοποιήσουμε για να σκεφτούμε σχετικά με τη βεβαιότητά τους. Κάθε γεγονός θα έχει μια σχετική πιθανότητα από μόνο του ή θα προκύπτει από την πιθανότητα άλλων γεγονότων. Αυτές οι πιθανότητες θα μας επιτρέψουν να λάβουμε αποφάσεις. Αυτή η λήψη αποφάσεων δεν είναι στατική, η πιθανότητα ενός γεγονότος μπορεί να τροποποιηθεί από την παρατήρηση και την τροποποίηση της πίστης σε άλλα συναφή γεγονότα.

Πριν μιλήσουμε για το πώς να μοντελοποιήσουμε τη συλλογιστική με τη χρήση πιθανοτήτων, πρέπει να επανεξετάσουμε ορισμένες βασικές έννοιες της θεωρίας πιθανοτήτων. Το θεμελιώδες στοιχείο της θεωρίας πιθανοτήτων είναι η τυχαία μεταβλητή. Μια τυχαία μεταβλητή έχει ένα πεδίο τιμών. Πιθανές τιμές που μπορεί να πάρει και στις οποίες αναθέτουμε μια κατανομή πιθανότητας. Μπορούμε να έχουμε τυχαίες μεταβλητές boolean, διακριτές ή συνεχείς. Στην πράξη, όλα τα σχήματα αναπαράστασης γνώσης βασίζονται στη λογική, οπότε θα χρησιμοποιήσουμε λογικούς τύπους ως βασικό στοιχείο αναπαράστασης. Με αυτόν τον τρόπο, θα ορίσουμε ως λογική πρόταση κάθε τύπο ή κατηγορημα, αφού αυτά είναι τα πρωταρχικά στοιχεία της αναπαράστασής μας. Μια λογική πρόταση θα έχει μια σχετική τυχαία μεταβλητή που θα δείχνει το βαθμό πίστης της.

Μια τυχαία μεταβλητή θα έχει μια σχετική κατανομή πιθανότητας. Ο τρόπος έκφρασης αυτής της κατανομής πιθανότητας εξαρτάται από τον τύπο της τυχαίας μεταβλητής (διακριτή ή συνεχή). Σε κάθε πρόβλημα, θα έχουμε διαφορετικές λογικές προτάσεις που θα παρεμβαίνουν σε μια απόφαση- επομένως, θα πρέπει να περιγράψουμε πώς είναι οι τυχαίες μεταβλητές που περιγράφουν αυτές τις προτάσεις και πώς επηρεάζουν τα συμπεράσματα που προκύπτουν από αυτές τις προτάσεις. Η ένωση των τυχαίων μεταβλητών μπορεί να περιγραφεί με τη χρήση μιας κοινής κατανομής πιθανοτήτων. Αυτός θα είναι ο μηχανισμός που θα μας επιτρέψει να

περιγράψουμε τον τρόπο με τον οποίο μπορούμε να συλλογιστόμαστε χρησιμοποιώντας πιθανότητες.

Γενικά, δεν σχετίζονται μεταξύ τους όλες οι προτάσεις που εμφανίζονται σε ένα πρόβλημα. Στην πραγματικότητα, για κάθε πρόταση της εξαρτημένης λογικής, μπορούμε να προσδιορίσουμε μόνο ένα υποσύνολο προτάσεων που τις επηρεάζουν, ενώ οι υπόλοιπες είναι άσχετες με την εξαγωγή συμπερασμάτων για τις πιθανότητές τους. Θα ονομάσουμε αυτή την ιδιότητα πιθανολογική ανεξαρτησία.

Τα δίκτυα Μπέιζ είναι ένας φορμαλισμός που επιτρέπει την αναπαράσταση των σχέσεων μεταξύ ενός συνόλου ανεξάρτητων τυχαίων μεταβλητών. Ένα δίκτυο Μπέιζ είναι ένας άκυκλος κατευθυνόμενος γράφος που περιέχει πιθανοτικές πληροφορίες στους κόμβους του, υποδεικνύοντας την επιρροή που έχουν σε έναν κόμβο οι γονείς του στο γράφο. Το σύνολο των πιθανοτήτων που αναπαρίσταται στο δίκτυο Μπέιζ περιγράφει την κοινή κατανομή πιθανοτήτων όλων των μεταβλητών, αυτό σημαίνει ότι δεν είναι απαραίτητος ένας πλήρης πίνακας που περιγράφει την επιρροή μεταξύ όλων αυτών.

Περιπτώσεις χρήσης

ΟΝΤΟΛΟΓΙΑ BBC

Το BBC διαθέτει μεγάλη ποσότητα περιεχομένου που δημιουργείται και διανέμεται παγκοσμίως. Αυτό το περιεχόμενο πρέπει να οργανωθεί για να είναι χρήσιμο, ώστε να μπορεί να εξυπηρετηθεί απευθείας ως περιεχόμενο ιστού ή να χρησιμοποιηθεί για τη δημιουργία νέου περιεχομένου για διαφορετικούς σκοπούς. Για την επίλυση αυτού του προβλήματος, το BBC έχει ορίσει διάφορες οντολογίες που εξειδικεύονται σε διάφορα θέματα, όπως αθλητισμός, φαγητό, πολιτική, άγρια φύση, δημιουργικά έργα και γενικές οντολογίες που επιτρέπουν την περιγραφή εννοιών του υπερτομέα, όπως πρόσωπα, γεγονότα, οργανισμοί, τόποι ή θέματα.

Αυτές οι οντολογίες επιτρέπουν στο BBC, για παράδειγμα, να οργανώσει το περιεχόμενο που εμφανίζεται στον ιστότοπό του. Μπορούν να ορίζουν τις ειδήσεις χρησιμοποιώντας όρους της οντολογίας και να συνδέουν τις πληροφορίες με τα στοιχεία που περιγράφονται, όπως τα πρόσωπα, οι οργανισμοί ή οι τόποι που εμφανίζονται στο κείμενο, άλλες ειδήσεις ή άλλα γεγονότα. Σε αυτές τις πληροφορίες μπορεί να πλοηγηθεί κανείς για να λάβει περισσότερες πληροφορίες που σχετίζονται με την τρέχουσα είδηση.

Το BBC έχει επίσης ορίσει μια οντολογία που ορίζει τα στοιχεία στα εθνικά προγράμματα σπουδών του Ηνωμένου Βασιλείου. Αυτό επιτρέπει στους μαθητές και τους εκπαιδευτικούς να έχουν πρόσβαση σε πόρους που έχει ετοιμάσει το BBC, όπως βίντεο και μαθήματα και να τα εξερευνήσουν, να τα συσχετίσουν και να τα ανακαλύψουν σύμφωνα με τα προγράμματα σπουδών. Η οντολογία ταξινομεί το περιεχόμενο ανάλογα με το επίπεδο, το θέμα και τον τομέα μελέτης. Το πλεονέκτημα αυτής της οντολογίας είναι ότι έχει αναπτυχθεί χρησιμοποιώντας τα πρότυπα του Σημαιολογικού Ιστού που ορίζονται από την Κοινοπραξία του Παγκόσμιου Ιστού (W3C) και χρησιμοποιώντας άλλες ήδη αναπτυγμένες οντολογίες του Σημαιολογικού Ιστού. Αυτό επιτρέπει τη σύνδεση των όρων τους με άλλους πόρους που εμπλουτίζουν τις πληροφορίες που μπορούν να συσχετιστούν και να προσπελαστούν.

Οντολογίες του BBC: <https://www.bbc.co.uk/ontologies/bbc>

Οντολογία προγραμμάτων σπουδών του BBC: <https://www.bbc.co.uk/ontologies/curriculum>

Οντολογία τροφίμων του BBC: <https://www.bbc.co.uk/ontologies/fo>

Αναφορές

- Artificialintelligence: a modern approach Russell,S.J.; Norvig,P, Prentice-Hall, (τέταρτη έκδοση) 2020. ISBN: 9781292153964 (κεφάλαια7-9,12-14)
- Τεχνητή νοημοσύνη: μια νέα σύνθεση Nilsson,N.J, Morgan Kaufmann Publishers, 1998.ISBN: 1558604677 (κεφάλαια12-19)
- Τεχνητή νοημοσύνη: δομές και στρατηγικές για την επίλυση σύνθετων προβλημάτων-Luger, G.F,Pearson Education: Addison Wesley, 2009. ISBN: 9780321545893 (κεφάλαια2,7-9).
- Ontologydevelopment101: Ένας οδηγός για τη δημιουργία της πρώτης σας οντολογίας. Noy,N.F.,&McGuinness,D.L.(2001).

Μάθημα 1.4: Μηχανική μάθηση

Θέματα προς συζήτηση

- Εισαγωγή στη μηχανική μάθηση
- Μάθηση με επίβλεψη
- Δέντρα αποφάσεων
- Naive Bayes

Σημειώσεις διάλεξης

- Εισαγωγή

Όλες οι μέθοδοι για τις οποίες μιλήσαμε στα προηγούμενα μαθήματα μπορούν να χρησιμοποιηθούν για την υλοποίηση εφαρμογών που χρειάζονται νοημοσύνη για την επίλυση των εργασιών τους. Επικεντρώνονται βασικά στη συμπερίληψη της γνώσης του τομέα για την ευκολότερη επίλυση προβλημάτων, αποφεύγοντας το υπολογιστικό κόστος που συνεπάγεται η χρήση μόνο γενικών αλγορίθμων. Ένας από τους περιορισμούς που μπορούμε να δούμε σε αυτές τις εφαρμογές είναι ότι δεν μπορούν να λύσουν προβλήματα για τα οποία δεν έχουν προγραμματιστεί. Τα όριά τους βρίσκονται στη γνώση που έχουμε συμπεριλάβει. Αν θέλουμε να έχουμε εφαρμογές που μπορούμε πραγματικά να θεωρήσουμε ευφυείς με την ευρύτερη έννοια, θα πρέπει να είναι σε θέση να επιτύχουν πράγματα που ξεπερνούν αυτό το όριο. Στην πραγματικότητα, μια εφαρμογή θα φαινόταν πιο ευφυής αν είναι σε θέση να προσαρμόζεται και να αποκτά νέες γνώσεις για να μπορεί να λύνει περισσότερα προβλήματα.

Όταν σχεδιάζουμε μια εφαρμογή Τεχνητής Νοημοσύνης (ή άλλου είδους εφαρμογές) για την επίλυση σύνθετων προβλημάτων, συνήθως πρέπει να αντιμετωπίσουμε περιορισμούς, όπως το ότι δεν μπορούμε να προβλέψουμε όλες τις πιθανές καταστάσεις που μπορεί να συμβούν, δεν μπορούμε να προβλέψουμε μελλοντικές αλλαγές στο περιβάλλον όπου αναπτύσσεται η εφαρμογή ή δεν μπορούμε να δημιουργήσουμε ένα πρόγραμμα για μια εργασία επειδή είναι πολύ περίπλοκη για να προγραμματιστεί με το χέρι. Η μηχανική μάθηση είναι ο τομέας της τεχνητής νοημοσύνης που παρέχει τις τεχνικές για να γίνει αυτό. Η ιδέα είναι να αναπτυχθούν μέθοδοι ικανές να αυξήσουν τις ικανότητες των εφαρμογών τεχνητής νοημοσύνης, όπως τα συστήματα βασισμένα στη γνώση, η επεξεργασία φυσικής γλώσσας ή η ρομποτική, ώστε να είναι πιο ευέλικτες και αποτελεσματικές. Επί του παρόντος, η μηχανική μάθηση εφαρμόζεται για κάθε εργασία που περιλαμβάνει τη χρήση δεδομένων για την επίλυση προβλημάτων. Αυτό συμβαίνει επειδή είναι πιο αποτελεσματική η εκμάθηση ενός μοντέλου από τα δεδομένα και η ενσωμάτωσή του σε άλλες εργασίες.

Υπάρχουν διάφοροι τρόποι ταξινόμησης των μεθόδων μηχανικής μάθησης. Ο πιο απλός τρόπος είναι να προσέξουμε την ανατροφοδότηση που χρησιμοποιεί ο αλγόριθμος για την εκτέλεση της μάθησης. Έχουμε τρεις διαφορετικές κατηγορίες αλγορίθμων μηχανικής μάθησης:

- Μάθηση χωρίς επίβλεψη, όπου δεν υπάρχουν πληροφορίες σχετικά με τον στόχο που θέλουμε να επιτύχουμε, οπότε πρέπει να ανακαλύψουμε τα μοτίβα που έχουν νόημα για τα δεδομένα.
- Μάθηση με επίβλεψη, όπου κάθε παράδειγμα συνδέεται με μια έξοδο και ο στόχος είναι να μάθουμε μια αναπαράσταση ή μια συνάρτηση ικανή να αντιστοιχίσει τα παραδείγματα στην έξοδο.
- Ενισχυτική μάθηση, όπου μια εργασία επιλύεται με δοκιμή και σφάλμα χρησιμοποιώντας ως ανατροφοδότηση μια σειρά από ανταμοιβές ή τιμωρίες ανάλογα με την αποτελεσματικότητα της τρέχουσας ικανότητας του μοντέλου για την επίλυση της εργασίας.

Μια άλλη ταξινόμηση γίνεται ανάλογα με την αναπαράσταση των δεδομένων που χρησιμοποιούνται για την εκμάθηση ενός μοντέλου. Υπάρχουν αλγόριθμοι που είναι σε θέση να μαθαίνουν από λογικές αναπαραστάσεις ή που χρησιμοποιούν μια αναπαράσταση βασισμένη σε ζεύγη χαρακτηριστικών-τιμών. Αυτές οι τελευταίες αναπαραστάσεις είναι οι πιο συχνά χρησιμοποιούμενες σε τρέχουσες εφαρμογές.

Μια τελευταία ταξινόμηση αφορά το αν η μάθηση γίνεται επαγωγικά, δηλαδή χρησιμοποιούμε συγκεκριμένα παραδείγματα και λαμβάνουμε μια αναπαράσταση/συνάρτηση γενικεύοντας τα δεδομένα. Αυτό το είδος μάθησης δεν είναι ορθό από την άποψη της τυπικής λογικής, διότι υποθέτουμε ότι ένα περιορισμένο σύνολο παραδειγμάτων είναι αρκετό για την εξαγωγή γενικών συμπερασμάτων. Αυτό σημαίνει ότι η αναπαράσταση που μαθαίνουμε μπορεί να αποδειχθεί ψευδής απλά και μόνο με την εύρεση ενός αντιπαραδείγματος. Αν και, αυτό είναι το είδος μάθησης που εκτελούν όλοι οι αλγόριθμοι με επίβλεψη και επιτρέπει την εκμάθηση χρήσιμων μοντέλων για πολλές εργασίες.

Τα μοντέλα μπορούν επίσης να μαθαίνουν επαγωγικά από μια λογική αναπαράσταση, πηγαίνοντας από γενικούς κανόνες και λαμβάνοντας τι άλλοι λογικοί τύποι συνεπάγονται. Αυτό το είδος μάθησης επιτρέπει να γίνει σαφής η γνώση, η οποία μπορεί στη συνέχεια να χρησιμοποιηθεί συμπληρωματικά με τη γενική γνώση από την οποία έχει προέλθει. Για παράδειγμα, υπάρχουν απλοί κανόνες που εξηγούν πώς παίζεται το σκάκι και όλες οι πιθανές παρτίδες σκακιού μπορούν να προκύψουν από αυτούς τους κανόνες, αλλά είναι χρήσιμο να εξαχθούν κάποια συγκεκριμένα σύνολα κινήσεων που μπορούν να εφαρμοστούν σε ένα παιχνίδι, όπως για παράδειγμα τα ανοίγματα των παρτίδων.

i. Μέθοδοι εποπτευόμενης μάθησης

Η επιβλεπόμενη μάθηση είναι μια εργασία μηχανικής μάθησης που έχει ως είσοδο ένα σύνολο παραδειγμάτων που περιγράφονται από ένα σύνολο χαρακτηριστικών. Κάθε παράδειγμα έχει συσχετισμένη μια έξοδο που είναι η σωστή απάντηση για το πρόβλημα που αντιπροσωπεύει. Το μοντέλο που θα εκπαιδευτεί θα συσχετίσει τα παραδείγματα με την έξοδό του αποκτώντας μια γενικότερη αναπαράσταση.

Τα δεδομένα μπορούν να αναπαρασταθούν με διάφορους τρόπους. Από το πρόβλημα του τομέα αφαιρείται ένα σύνολο σχετικών χαρακτηριστικών και επιλέγεται ένας κατάλληλος τύπος δεδομένων. Συνήθως τα χαρακτηριστικά μπορούν να είναι συνεχή (πραγματικοί αριθμοί) ή

διακριτά (απαρίθμηση τιμών με ή χωρίς σειρά). Κάθε παράδειγμα θα αποτελείται από έναν κατάλογο των τιμών που αντιστοιχούν σε κάθε επιλεγμένο χαρακτηριστικό.

Για την έξοδο, μπορούμε να έχουμε συνεχή ή διακριτή έξοδο. Προβλήματα με συνεχείς τιμές αντιστοιχούν στο έργο που ονομάζεται παλινδρόμηση, προβλήματα με διακριτή έξοδο αντιστοιχούν στο έργο της ταξινόμησης. Η επιλογή του ενός ή του άλλου εξαρτάται από το πρόβλημα και τον τομέα. Οι περισσότεροι αλγόριθμοι μηχανικής μάθησης μπορούν να επιλύσουν και τα δύο προβλήματα.

Το μοντέλο μπορεί να αναπαρασταθεί με διάφορους τρόπους. Μπορεί να είναι ένα σύνολο κανόνων, ένας συνδυασμός συναρτήσεων κατανομής πιθανοτήτων, ένας συνδυασμός σύνθετων συναρτήσεων... Διαφορετικές αναπαραστάσεις συνδέονται με διαφορετικούς αλγορίθμους μάθησης με επίβλεψη. Ανάλογα με τα χαρακτηριστικά του τομέα, ορισμένα μοντέλα θα προτιμηθούν από άλλα. Ορισμένα μοντέλα μπορούν να εξηγήσουν τις αποφάσεις τους εάν βασίζονται σε κανόνες ή υπολογίζουν τη συνάφεια κάθε χαρακτηριστικού εισόδου στην απόφαση της εξόδου. Ορισμένοι αλγόριθμοι ταξινόμησης δίνουν απλώς μια μοναδική απάντηση, ενώ άλλοι είναι σε θέση να δώσουν μια βεβαιότητα σχετικά με την απάντηση.

- **Δέντρα Αποφάσεων**

Μπορούμε να εκπαιδεύσουμε μια συνάρτηση που συσχετίζει ένα διάνυσμα παραδειγμάτων χαρακτηριστικών-τιμών με μια έξοδο, αναζητώντας το ελάχιστο σύνολο ερωτήσεων που πρέπει να θέσουμε προκειμένου να διαχωρίσουμε κάθε παράδειγμα από άλλα παραδείγματα ανάλογα με την έξοδό τους. Αυτό το σύνολο ερωτήσεων θα είναι το μοντέλο που αναπαριστά τα παραδείγματα. Ο στόχος είναι να αποκτήσουμε μια γενική αναπαράσταση που να είναι σε θέση να διαχωρίζει τα παραδείγματα ανάλογα με την ετικέτα τους για την εργασία ταξινόμησης ή ανάλογα με την τιμή τους για την εργασία παλινδρόμησης.

Η αναπαράσταση του μοντέλου είναι ένα δέντρο που οργανώνει τις αποφάσεις/ερωτήματα που πρέπει να δοκιμαστούν στα παραδείγματα εισόδου προκειμένου να ληφθεί η έξοδός τους. Κάθε απάντηση οδηγεί σε διαφορετικό κλάδο του δέντρου μέχρι να φτάσουμε σε ένα φύλλο. Τα φύλλα του δέντρου θα υπολογίζουν την έξοδο, είτε πρόκειται για μια ετικέτα είτε για μια πραγματική τιμή.

Αυτή η αναπαράσταση είναι ισοδύναμη με έναν λογικό τύπο σε Disjunctive Normal Form (DNF), δηλαδή ένα σύνολο από συζευκτικούς τύπους που συνδυάζονται με διαζεύξεις. Αυτό μπορεί επίσης να γίνει κατανοητό ως ένα σύνολο εναλλακτικών κανόνων που μπορούν να ελεγχθούν για να προκύψει η έξοδος.

Για να βρούμε το ελάχιστο δέντρο που εξηγεί ένα σύνολο δεδομένων θα πρέπει να εξερευνήσουμε όλους τους δυνατούς τύπους DNF που μπορούν να συνθέσουν τα χαρακτηριστικά και τις τιμές των δεδομένων. Ο αριθμός των πιθανών τύπων αυξάνεται εκθετικά με τον αριθμό των χαρακτηριστικών, οπότε θα πρέπει να χρησιμοποιήσουμε μια προσεγγιστική στρατηγική. Επειδή αυτό το πρόβλημα μπορεί να αναπαρασταθεί ως πρόβλημα αναζήτησης όπως αυτά που

καλύφθηκαν στο μάθημα 1.2, θα εξερευνήσουμε τα πιθανά δέντρα χρησιμοποιώντας μια ευρετική συνάρτηση και μια στρατηγική αναζήτησης.

Ο χώρος αναζήτησης θα είναι τα πιθανά δέντρα που μπορούμε να κατασκευάσουμε με τα χαρακτηριστικά. Θα εξερευνήσουμε αυτόν τον χώρο δουλεύοντας με μερικές λύσεις, ξεκινώντας με ένα άδριο δέντρο και προσθέτοντας περισσότερες αποφάσεις που θα βελτιώσουν μια ευρετική συνάρτηση που λειτουργεί με παρόμοιο τρόπο με τον αλγόριθμο αναρρίχησης λόφου.

Η ευρετική συνάρτηση θα στρέψει την αναζήτηση προς το δέντρο αποφάσεων που περιλαμβάνει τον ελάχιστο αριθμό ερωτήσεων. Η ελαχιστοποίηση του μεγέθους του μοντέλου που μαθαίνεται είναι μια συνήθης προκατάληψη στους αλγορίθμους μηχανικής μάθησης. Η αιτιολόγηση για τα δέντρα αποφάσεων είναι ότι ελαχιστοποιώντας τον αριθμό των ερωτήσεων, μειώνουμε την πιθανότητα να συμπεριλάβουμε ψευδείς ερωτήσεις που είναι πολύ συγκεκριμένες για τα δεδομένα που χρησιμοποιούμε για τη δημιουργία του μοντέλου. Η εκμάθηση πολύ συγκεκριμένων αναπαραστάσεων είναι γνωστή ως υπερπροσαρμογή (overfitting). Αυτό θα βοηθήσει στην απόκτηση μιας πιο γενικής αναπαράστασης, ώστε να εφαρμόζεται σε νέα παραδείγματα και όχι μόνο σε αυτά που έχουμε χρησιμοποιήσει. Αυτή η μεροληψία είναι επίσης γνωστή ως ξυράφι του Όκαμ, που μπορεί να διατυπωθεί ως εξής: "Η πολυμορφία δεν θα έπρεπε ποτέ να τίθεται χωρίς ανάγκη".

Υπάρχουν διάφορες παραλλαγές των αλγορίθμων δέντρων απόφασης. Εμείς θα εξηγήσουμε τον πιο γενικό αλγόριθμο που αποτελεί την κανονική υλοποίηση για τις περισσότερες βιβλιοθήκες μηχανικής μάθησης που μπορούν να χρησιμοποιηθούν για αυτή την εργασία. Τα δέντρα απόφασης μπορούν να χρησιμοποιηθούν για την επίλυση εργασιών ταξινόμησης και παλινδρόμησης, εμείς θα επικεντρωθούμε μόνο στην ταξινόμηση. Ο αλγόριθμος μπορεί να οριστεί αναδρομικά, ξεκινά με ένα κενό δέντρο και προσθέτει μια απόφαση σε κάθε βήμα με τον ακόλουθο τρόπο:

1. Εάν όλα τα τρέχοντα παραδείγματα έχουν μόνο μία ετικέτα, προσθέστε ένα φύλλο στο δέντρο με την ετικέτα αυτή.
2. Εάν τα τρέχοντα παραδείγματα έχουν διαφορετικές ετικέτες:
 - 2.1. Αξιολογήστε τα χαρακτηριστικά των τρεχόντων παραδειγμάτων και υπολογίστε τη σημασία τους για το διαχωρισμό των παραδειγμάτων ανάλογα με την ετικέτα τους.
 - 2.2. Επιλέξτε το καλύτερο χαρακτηριστικό
 - 2.3. Διαχωρίστε τα παραδείγματα σύμφωνα με τις τιμές του επιλεγμένου χαρακτηριστικού
 - 2.4. Εφαρμόστε αναδρομικά τον αλγόριθμο σε κάθε διαμέριση και λάβετε τα υποδέντρα που ταξινομούν την διαμέριση
 - 2.5. Δημιουργήστε έναν κόμβο απόφασης για το χαρακτηριστικό όπου κάθε τιμή έχει ως υποδέντρο το αποτέλεσμα της αναδρομικής κλήσης

Όπως βλέπουμε, ο αλγόριθμος λειτουργεί διαιρετικά, προσθέτοντας σε κάθε βήμα έναν νέο κόμβο στο δέντρο και χωρίζοντας τα δεδομένα σύμφωνα με την απόφαση που προστίθεται στο δέντρο.

Το βασικό σημείο αυτού του αλγορίθμου είναι το πώς επιλέγουμε το καλύτερο χαρακτηριστικό όταν πρέπει να προσθέσουμε μια νέα απόφαση στο δέντρο. Υπάρχουν διάφορες ευρετικές μέθοδοι που μπορούν να εφαρμοστούν για να μετρηθεί πώς ένα χαρακτηριστικό θα μειώσει τον αριθμό των αποφάσεων που θα χρειαστούν για την ταξινόμηση ενός συνόλου παραδειγμάτων.

Υπάρχουν βασικά δύο συναρτήσεις που χρησιμοποιούνται ως ευριστικά κέρδος πληροφορίας και ο δείκτης Gini. Θα εξηγήσουμε πώς λειτουργεί η πρώτη.

Η απόκτηση πληροφοριών βασίζεται στη θεωρία της πληροφορίας. Ο τομέας αυτός μελετά μεταξύ άλλων την κωδικοποίηση μηνυμάτων. Στην περίπτωση μας η κατασκευή ενός δέντρου είναι ουσιαστικά η απόκτηση του ελάχιστου κώδικα που μπορεί να αναπαραστήσει τα δεδομένα που έχουμε. Μπορούμε επίσης να το δούμε ως μια εργασία συμπίεσης. Θέλουμε να συμπίεσουμε την πληροφορία των δεδομένων, ώστε να μπορούμε να τα μεταδώσουμε όσο το δυνατόν φθηνότερα.

Η θεωρία της πληροφορίας μπορεί να υπολογίσει πόσο μακρύς πρέπει να είναι ένας κώδικας για να μεταδώσει ένα σύνολο μηνυμάτων ανάλογα με το ποιες είναι οι μεταξύ τους πιθανότητες. Ένα μήνυμα που είναι πιο συχνό μπορεί να κωδικοποιηθεί με έναν πιο σύντομο κώδικα, ώστε να χρειάζεται λιγότερη προσπάθεια για να μεταδοθεί και μπορούμε να κρατήσουμε μεγαλύτερους κώδικες για πιο σπάνια μηνύματα.

Μπορούμε να υπολογίσουμε τον αριθμό των bits που απαιτούνται για την κωδικοποίηση ενός συνόλου μηνυμάτων σύμφωνα με τις πιθανότητές τους χρησιμοποιώντας τη συνάρτηση εντροπίας (H), δεδομένου ενός συνόλου μηνυμάτων $M=\{m_1, \dots, m_n\}$ και τις πιθανότητές τους $P(M) = \{P(m_1), \dots, P(m_n)\}$ μπορούμε να υπολογίσουμε:

$$H(M | A) = \sum_{a \in A} P(M | A=a) H(M | A=a)$$

Μπορούμε να θεωρήσουμε ότι οι ετικέτες που έχουμε στα δεδομένα μας αντιστοιχούν στα μηνύματα και οι πιθανότητές τους στη συχνότητα που έχουν. Για να πάρουμε μια απόφαση πρέπει να υπολογίσουμε την υπό συνθήκη εντροπία σε σχέση με τις τιμές για κάθε ένα από τα χαρακτηριστικά που περιγράφουν τα παραδείγματα:

$$H(M) = - \sum_{i=1}^n P(m_i) * \log(P(m_i))$$

Όπου $P(M|A=a)$ είναι η πιθανότητα ένα παράδειγμα να έχει την τιμή a στο χαρακτηριστικό A και $H(M|A=a)$ είναι η εντροπία των παραδειγμάτων που έχουν την τιμή a στο χαρακτηριστικό A . Ουσιαστικά ο σταθμισμένος μέσος όρος της εντροπίας των δεδομένων αν διαιρεθεί με τις τιμές ενός χαρακτηριστικού. Όσο πιο κοντά είναι αυτή η τιμή στην εντροπία, τόσο λιγότερο σημαντικό είναι το χαρακτηριστικό, αυτό λογίζεται ως κέρδος πληροφορίας:

$$IG(M|A)=H(M) - H(M|A)$$

Έτσι, σε κάθε σημείο απόφασης, πρέπει να υπολογίζουμε τη συνάρτηση IG και να επιλέγουμε το χαρακτηριστικό με την υψηλότερη τιμή.

- **Naive Bayes (Ναίβ Μπέης)**

Τα δέντρα αποφάσεων υποθέτουν ότι το σύνολο των εννοιών που θέλουμε να μάθουμε μπορεί να επιμεριστεί τέλεια με ένα σύνολο κανόνων. Μερικές φορές τα όρια μεταξύ των εννοιών είναι

ασαφή και θέλουμε ένα μοντέλο που να μπορεί να αναπαραστήσει αυτή την κατάσταση. Υποθέτουμε ότι η συσχέτιση μεταξύ εισόδου και εξόδου είναι επίσης αβέβαιη και θέλουμε να είμαστε σε θέση να ποσοτικοποιήσουμε αυτή την αβεβαιότητα.

Προκειμένου να μοντελοποιήσουμε την αβεβαιότητα στην περιγραφή της έννοιας και στις αποφάσεις, μπορούμε να χρησιμοποιήσουμε κατανομή πιθανοτήτων για να αναπαραστήσουμε τα χαρακτηριστικά των δεδομένων εισόδου και της συσχέτισής τους. Πρόκειται για την ίδια έννοια που χρησιμοποιήθηκε στο μάθημα 1.3, όταν εξηγήθηκαν οι μέθοδοι για αβέβαιη γνώση και συλλογισμό. Σε αυτή την περίπτωση, αντί να ορίζονται χειροκίνητα τα μοντέλα, αυτά προκύπτουν από τα δεδομένα.

Για την εκμάθηση του μοντέλου από τα δεδομένα, πρέπει να αποφασίσουμε ποια κατανομή πιθανότητας αντιπροσωπεύει καλύτερα κάθε χαρακτηριστικό, επιλέγοντας μεταξύ των υφιστάμενων κατανομών και λαμβάνοντας υπόψη αν τα δεδομένα είναι διακριτά ή συνεχή και τη συμπεριφορά των τιμών τους. Ο μαθησιακός στόχος είναι η απόκτηση μιας κατανομής πιθανοτήτων που περιγράφει τις αλληλεπιδράσεις μεταξύ των χαρακτηριστικών και της εξόδου.

Όπως εξηγήσαμε στο μάθημα 1.3, είναι πολύπλοκο να υπολογίσουμε ένα μοντέλο για όλες τις πιθανές αλληλεπιδράσεις μεταξύ των χαρακτηριστικών, οπότε πρέπει να υποθέσουμε κάποιες σχέσεις ανεξαρτησίας που βοηθούν να διασπάσουμε την κοινή κατανομή πιθανοτήτων σε παράγοντες που μπορούν να εκτιμηθούν ανεξάρτητα και στη συνέχεια να συνδυαστούν για τη λήψη αποφάσεων. Υπάρχουν διαφορετικές υποθέσεις που μπορούμε να υιοθετήσουμε και διαφορετικές πολυπλοκότητες για το μοντέλο που πρέπει να εκτιμηθεί. Ο φορμαλισμός των Μπεϋζιανών δικτύων μπορεί να μαθευτεί από τα δεδομένα με κάποιες εγγυήσεις για το υπολογιστικό κόστος, αν ο γράφος που θέλουμε να μάθουμε έχει ορισμένους περιορισμούς. Το ένα άκρο είναι το μοντέλο που ονομάζεται αφελές Bayes. Πρόκειται για ένα μοντέλο που υποθέτει ότι όλες οι μεταβλητές που περιγράφουν την είσοδο είναι ανεξάρτητες μεταξύ τους και η μόνη εξάρτηση του γράφου είναι με την έξοδο. Αυτή η υπόθεση δεν ισχύει πάντα, στα περισσότερα προβλήματα υπάρχουν κάποιες αλληλεπιδράσεις μεταξύ των χαρακτηριστικών εισόδου, αλλά αν αυτή η εξάρτηση δεν είναι ισχυρή, αυτό το μοντέλο μπορεί να έχει καλά αποτελέσματα.

Αυτό μειώνει το κόστος εκμάθησης ενός απλού μοντέλου bayes για να εκτιμήσει ανεξάρτητα για κάθε μεταβλητή τη σχέση εξάρτησής του με την έξοδο. Αυτή η εκτίμηση εξαρτάται από τον τύπο της κατανομής πιθανοτήτων που χρησιμοποιείται για την περιγραφή των μεταβλητών. Υποθέτοντας διακριτές μεταβλητές και διακριτή έξοδο, μπορούν να μοντελοποιηθούν χρησιμοποιώντας πολυωνυμικές κατανομές που υπολογίζουν τον πίνακα ενδεχομένων για κάθε ζεύγος μεταβλητής-εξόδου. Αυτό σημαίνει ότι απλώς υπολογίζονται οι πιθανότητες για κάθε συνδυασμό τιμών μεταβλητής-εξόδου και αποθηκεύεται το αποτέλεσμα για την πραγματοποίηση προβλέψεων.

Όπως και στα δίκτυα Bayes, η εξαγωγή συμπερασμάτων μπορεί να γίνει με την εφαρμογή του θεωρήματος Bayes, όπου η έξοδος εξαρτάται από την είσοδο που πρόκειται να προβλεφθεί. Η εξαρτημένη πιθανότητα των δεδομένων ως προς την έξοδο αντιστοιχεί στην κοινή κατανομή πιθανότητας που υπολογίζεται ως το γινόμενο των εκτιμώμενων πιθανοτήτων για κάθε χαρακτηριστικό $P(Ea|O)$ δεδομένου ότι θεωρείται ότι όλα είναι ανεξάρτητα. Οι εκ των προτέρων

πιθανότητες της εξόδου $P(O)$ είναι η εκτίμηση των πιθανοτήτων των τιμών της εξόδου από τα δεδομένα. Το συμπέρασμα μπορεί να υπολογιστεί ως εξής:

$$P(O | E) = P(E | O)P(O) = \prod_a P(E_a | O)P(O)$$

Περιπτώσεις χρήσης

NAIVE BAYES

Έχουμε συνηθίσει να στέλνουμε και να λαμβάνουμε ηλεκτρονικό ταχυδρομείο και η εμπειρία μας λέει ότι κάθε μέρα θα λαμβάνουμε αρκετά ηλεκτρονικά μηνύματα με παράξενες προτάσεις για θαυματοουργά προϊόντα, ευκαιρίες για πλουτισμό ή ανεπιθύμητες διαφημίσεις. Έχουμε επίσης συνηθίσει να επισημαίνουμε αυτές τις επικοινωνίες στην εφαρμογή ανάγνωσης ηλεκτρονικού ταχυδρομείου και σταδιακά βλέπουμε ότι παρόμοια μηνύματα ηλεκτρονικού ταχυδρομείου ταξινομούνται ως ανεπιθύμητα και μεταφέρονται σε έναν συγκεκριμένο φάκελο.

Προφανώς, κάθε άτομο λαμβάνει διαφορετικά μηνύματα ηλεκτρονικού ταχυδρομείου, οπότε είναι αδύνατο να προγραμματίσουμε χειροκίνητα το έργο της ταξινόμησης των μηνυμάτων spam. Αυτό είναι ένα έργο που μπορούμε να λύσουμε μόνο με τη χρήση μηχανικής μάθησης, ώστε να αποκτήσουμε ένα μοντέλο που μπορεί να προσαρμοστεί στις ειδικές συνθήκες κάθε χρήστη. Έτσι, όταν χαρακτηρίζουμε αυτά τα μηνύματα ηλεκτρονικού ταχυδρομείου, εκπαιδεύουμε έναν αλγόριθμο μηχανικής μάθησης. Λέμε ακριβώς στον αλγόριθμο τι είναι μέλος της κατηγορίας spam και υποθέτουμε ότι ό,τι δεν έχει επισημανθεί ανήκει στην κατηγορία no spam.

Μπορούμε να εφαρμόσουμε διαφορετικούς αλγόριθμους μηχανικής μάθησης σε αυτό το πρόβλημα, αλλά χρειαζόμαστε έναν αλγόριθμο που δεν χρειάζεται να εκπαιδεύεται από την αρχή κάθε φορά που λαμβάνουμε ένα νέο μήνυμα ηλεκτρονικού ταχυδρομείου και που κάνει τις απλούστερες υποθέσεις για τα δεδομένα που πρόκειται να χρησιμοποιήσουμε. Αυτή είναι μια τέλεια εφαρμογή για πιθανοτικά μοντέλα και συγκεκριμένα για τον αλγόριθμο Naive Bayes.

Ο Naive Bayes υποθέτει ότι τα χαρακτηριστικά που περιγράφουν τα δεδομένα είναι ανεξάρτητα. Αυτό δεν είναι πραγματικά αληθές γενικά, αλλά τα μηνύματα spam θα έχουν πολύ συγκεκριμένα χαρακτηριστικά, αρκετά διαφορετικά από τα νόμιμα μηνύματα που θα είναι αρκετά για να έχουμε καλούς ταξινομητές.

Μια συγκεκριμένη εργασία που πρέπει να γίνει όταν πρέπει να χρησιμοποιηθεί ένας αλγόριθμος μηχανικής μάθησης, είναι η επιλογή των χαρακτηριστικών που θα περιγράφουν τα δεδομένα. Στην προκειμένη περίπτωση το έργο αυτό είναι δύσκολο επειδή τα μηνύματα ηλεκτρονικού ταχυδρομείου είναι κείμενο. Κάθε ένα από αυτά έχει διαφορετικό σύνολο λέξεων και ο αριθμός των λέξεων μπορεί να διαφέρει πολύ. Στην περίπτωση αυτή χρησιμοποιείται μια ειδική αναπαράσταση για τον ορισμό μέρους των χαρακτηριστικών σε ένα ηλεκτρονικό ταχυδρομείο. Ονομάζεται αναπαράσταση σακούλας λέξεων. Αυτή η αναπαράσταση υποθέτει ότι υπάρχει ένα λεξιλόγιο λέξεων και ότι ένα ηλεκτρονικό ταχυδρομείο αναπαρίσταται από ένα σύνολο μετρήσεων των λέξεων από το λεξιλόγιο. Άλλα χαρακτηριστικά μπορεί επίσης να είναι ενδιαφέροντα για τη διάκριση των νόμιμων και των ανεπιθύμητων μηνυμάτων πέρα από τις λέξεις.

Βίντεο για το Naive Bayes που εξηγεί την εφαρμογή ταξινόμησης Spam Mail

<https://www.youtube.com/watch?v=Q8lOVip5YUw>

Αναφορές

- Artificial intelligence: a modern approach Russell, S. J. ; Norvig, P, Prentice-Hall, (fourth edition) 2020. ISBN: 9781292153964 (chapters18,20)
- Artificial intelligence: structures and strategies for complex problem solving -Luger, G. F, Pearson Education: Addison Wesley, 2009. ISBN: 9780321545893 (chapter10)

Μάθημα 1.5: Εφαρμογές της Τεχνητής Νοημοσύνης

Θέματα προς συζήτηση

- Εφαρμογές της Τεχνητής Νοημοσύνης
- Γλώσσες για την τεχνητή νοημοσύνη

Σημειώσεις διάλεξης

- **Προσωπικοί βοηθοί**

Έχουμε ήδη συνηθίσει να έχουμε έναν προσωπικό βοηθό στο τηλέφωνό μας ή σε μια συσκευή στο σπίτι μας. Αυτοί οι βοηθοί έχουν εξελιχθεί από τα απλά ρομπότ συνομιλίας που ξεκίνησαν στα σπάργανα της τεχνητής νοημοσύνης.

Παρουσιάζουν διαφορετικές δυνατότητες, από το να διαθέτουν διεπαφή αναγνώρισης φωνής έως διεπαφές κειμένου. Οι τομείς της επεξεργασίας φυσικής γλώσσας και της αντίληψης αποτελούν μέρος του προγραμματισμού τους.

Δεν υπάρχει ακόμη μεγάλη πολυπλοκότητα στις διεπαφές κειμένου που καθοδηγούνται κυρίως από λέξεις-κλειδιά και απλούς κανόνες, αλλά χρησιμοποιούνται μοντέλα μηχανικής μάθησης για να αναγνωρίζουν τις προθέσεις του χρήστη και να μπορούν να δουλεύουν με ένα ευρύτερο φάσμα εκφράσεων που κάνουν την αλληλεπίδραση πιο φυσική. Τα μοντέλα για την αναγνώριση φωνής λαμβάνονται με τη χρήση τεχνικών μηχανικής μάθησης, όπου τα νευρωνικά δίκτυα έχουν δείξει την υπεροχή τους χάρη σε μεγάλα σύνολα δεδομένων ηχογραφήσεων φωνής με τις απομαγνητοφωνήσεις τους.

Το φάσμα των ενεργειών που μπορούν να εκτελέσουν αυτοί οι βοηθοί αυξάνεται επίσης, συμπεριλαμβανομένων άλλων εργασιών φυσικής γλώσσας, όπως η απάντηση ερωτήσεων αναγνωρίζοντας αυτή την πρόθεση του χρήστη και μετατρέποντάς την σε ερώτημα κειμένου σε μια μηχανή αναζήτησης. Αυτές οι εργασίες εκτελούνται καλύτερα σε εξειδικευμένους τομείς, όπως για παράδειγμα στις κρατήσεις ταξιδιών ή πτήσεων ή στον προγραμματισμό συναντήσεων, όπου οι προθέσεις του χρήστη και οι πιθανές αλληλεπιδράσεις είναι πιο περιορισμένες.

Μπορούν επίσης να βοηθήσουν στην αλληλεπίδραση με άλλες συσκευές, χρησιμεύοντας ως διεπαφή φωνητικών εντολών και χρησιμοποιώντας σύνολα κανόνων που αναπαρίστανται ως διαδικαστική γνώση που περιγράφεται σε απλές γλώσσες αναπαράστασης γνώσης.

Αυτός ο τομέας έχει φέρει ένα μεγάλο σύνολο εργαλείων που βοηθούν στην ανάπτυξη αυτών των βοηθών, εκδημοκρατίζοντας τη χρήση τους. Αυτό δημιούργησε επίσης μια αναπτυσσόμενη αγορά για αυτά τα εργαλεία, στην οποία κυριαρχούν όλο και περισσότερο οι μεγάλες εταιρείες τεχνολογίας.

- **Συστήματα συστάσεων**

Έχουμε ήδη συνηθίσει να λαμβάνουμε συστάσεις από υπηρεσίες που αποτελούν πλέον μέρος της ζωής μας, όπως ηλεκτρονικά καταστήματα ή υπηρεσίες ροής και ειδήσεων. Το περιεχόμενο μπορεί να είναι σχεδόν οτιδήποτε, από βιβλία, μουσική ή ταινίες μέχρι πιο σύνθετα πράγματα όπως άρθρα ειδήσεων, φίλοι ή διαφημίσεις.

Αυτά τα συστήματα συλλέγουν πληροφορίες σχετικά με τη συμπεριφορά των χρηστών τους και τα σχόλιά τους για να εξατομικεύσουν την εμπειρία τους, κάνοντας συστάσεις που μπορούν να αυξήσουν τα έσοδα ή να βελτιώσουν την αντίληψη της υπηρεσίας.

Τα θεμέλια αυτών των συστάσεων βρίσκονται σε μεθόδους μηχανικής μάθησης που συσχετίζουν τη συμπεριφορά του χρήστη με τα προσωπικά του χαρακτηριστικά. Υπάρχουν διάφοροι εξειδικευμένοι επιμέρους τομείς στα συστήματα συστάσεων, δεδομένης της ποικιλίας του περιεχομένου που μπορεί να συστηθεί και του διαφορετικού είδους ανατροφοδότησης που μπορεί να ληφθεί από τον χρήστη. Ένας από αυτούς τους τομείς είναι το συνεργατικό φιλτράρισμα, το οποίο χρησιμοποιεί τις πληροφορίες από παρόμοιους χρήστες για τον υπολογισμό των συστάσεων σε μια εφαρμογή της αρχής της σοφίας του πλήθους. Η επεξεργασία φυσικής γλώσσας χρησιμοποιείται επίσης σε αυτά τα συστήματα προκειμένου να εξαχθούν πληροφορίες από το περιεχόμενο (π.χ. ειδήσεις) ή να ανιχνευθεί το συναίσθημα του χρήστη (π.χ. κριτικές) που μπορεί να χρησιμοποιηθεί για να προστεθούν περισσότερες πληροφορίες για τον υπολογισμό των συστάσεων.

- **Αυτόνομη οδήγηση**

Ο απώτερος στόχος της τεχνητής νοημοσύνης είναι να είναι σε θέση να αναπτύξει αυτόνομους πράκτορες που θα είναι σε θέση να επιλύουν πολύπλοκα προβλήματα στον πραγματικό κόσμο. Η αυτόνομη οδήγηση μοιάζει με το καλύτερο παράδειγμα αυτού του στόχου.

Πρόκειται για μια εφαρμογή που εξελίσσει όλους τους τομείς της τεχνητής νοημοσύνης στο επόμενο επίπεδο, εμπλέκοντας την αντίληψη με τους πιο σύνθετους τρόπους, λόγω της ανάγκης αναγνώρισης σε πραγματικό χρόνο κάθε στοιχείου που υπάρχει στο δρόμο και στο περιβάλλον του. Το σύστημα πρέπει επίσης να αναπαριστά εσωτερικά τις πληροφορίες που εξάγει από την αντίληψη, χρησιμοποιώντας γλώσσες αναπαράστασης γνώσης και εκτελώντας αυτόματη συλλογιστική για τη λήψη αποφάσεων. Όλα τα μοντέλα που χρησιμοποιούνται για την επεξεργασία των πληροφοριών που επεξεργάζονται οι αισθητήρες του συστήματος πρέπει να εκπαιδεύονται χρησιμοποιώντας τεχνικές μηχανικής μάθησης που θα εξάγουν τα χαρακτηριστικά που χρησιμοποιούνται στις αποφάσεις. Τέλος, το σύστημα πρέπει να ενεργεί ως συνέπεια των αποφάσεων που θα περιλαμβάνουν το χειρισμό των ενεργοποιητών του αυτοκινήτου.

Καθώς η τεχνολογία στην αυτόνομη οδήγηση εξελίσσεται, ο συντονισμός και η διαπραγμάτευση μεταξύ των οχημάτων ή ακόμη και άλλων στοιχείων που βρίσκονται στο δρόμο, όπως τα φανάρια ή άλλες οδικές πινακίδες, θα περιλαμβάνει τεχνικές από την κατανομημένη τεχνητή νοημοσύνη.

- **Ευφυές λιανικό εμπόριο**

Ως επέκταση των συστάσεων από τα ηλεκτρονικά καταστήματα, οι μηχανισμοί τεχνητής νοημοσύνης μπορούν επίσης να αναπτυχθούν σε φυσικά καταστήματα, προκειμένου να ανιχνεύουν και να παρακολουθούν τους χρήστες μέσω καμερών ή κινητών τηλεφώνων. Όπως και

στο αντίστοιχο διαδικτυακό, στόχος είναι η παρατήρηση της συμπεριφοράς των πελατών και η αλληλεπίδραση μαζί τους με τη χρήση εφαρμογών για κινητά τηλέφωνα.

Υπάρχουν πολλές τεχνικές που μπορούν να εφαρμοστούν από τον τομέα της μηχανικής μάθησης για την ανακάλυψη μοτίβων και την παρακολούθηση της συμπεριφοράς των πελατών κατά τη διάρκεια του χρόνου. Οι πιο σύνθετες εφαρμογές αφορούν την πλήρη αυτοματοποίηση της αγοραστικής εμπειρίας μέσα στο κατάστημα, την παρακολούθηση των κινήσεων του χρήστη, την ανίχνευση των αλληλεπιδράσεων με τα προϊόντα στο κατάστημα, την αναγνώριση όταν κάτι έχει επιλεγεί και την αυτόματη χρέωση του χρήστη για την τελική αγορά μετά την έξοδο από το κατάστημα. Αυτό περιλαμβάνει σύνθετες εργασίες μηχανικής μάθησης, όρασης υπολογιστών, αναπαράστασης γνώσης και συλλογισμού.

- **Παιχνίδια υπολογιστών**

Τα ηλεκτρονικά παιχνίδια αποτελούν το τέλειο περιβάλλον για την εφαρμογή της τεχνητής νοημοσύνης, και ορισμένα παιχνίδια έχουν αποτελέσει από την αρχή του κλάδου τον πήχη για την επίτευξη της τεχνητής νοημοσύνης. Το σκάκι και το Go αποτέλεσαν τεράστια ορόσημα. Και τα δύο έχουν καταρριφθεί τα τελευταία χρόνια.

Υπάρχουν διαφορετικές τεχνικές που εφαρμόζονται στα ηλεκτρονικά παιχνίδια ανάλογα με τα χαρακτηριστικά του παιχνιδιού. Ορισμένα παιχνίδια χρειάζονται έναν αντίπαλο υπολογιστή ικανό να παίξει σε ένα επίπεδο αρκετά καλό ώστε να είναι ενδιαφέρον για τον χρήστη. Οι περισσότεροι από αυτούς τους παίκτες μπορούν να προγραμματιστούν με τη χρήση αναζήτησης με αντιπαλότητα. Ορισμένα παιχνίδια μπορούν να αναπαρασταθούν με απλούς κανόνες που μπορούν να περιγραφούν με χρήση διαδικαστικής γνώσης και να εκτελεστούν με χρήση απλών διαδικασιών συλλογισμού. Ορισμένα περιβάλλοντα απαιτούν από τον παίκτη υπολογιστή να προσαρμόζεται στον ανθρώπινο παίκτη ή να αναπτύσσει το έργο του σε ένα περιβάλλον που καθιστά δύσκολο τον προγραμματισμό του με απλούς κανόνες, οπότε μια συμπεριφορά πρέπει να μαθευτεί με τη χρήση ενισχυτικής μάθησης.

Τα πιο σύνθετα παιχνίδια κάνουν χρήση τεχνικών επεξεργασίας φυσικής γλώσσας για διεπαφές που βασίζονται σε διαλόγους, ή παράγουν νέο περιεχόμενο διαδικαστικά χρησιμοποιώντας αλγόριθμους επίλυσης προβλημάτων που βασίζονται σε προβλήματα ικανοποίησης περιορισμών ή γενετικούς αλγόριθμους.

- **Ρομποτικές συσκευές και έξυπνες συσκευές**

Έχουμε πλέον στο σπίτι μας ρομποτικές συσκευές που διαθέτουν αισθητήρες και μηχανισμούς ενεργοποίησης. Από τις απλές συσκευές του διαδικτύου των πραγμάτων που ελέγχουν για παράδειγμα τη μονάδα κλιματισμού μας, μέχρι την ηλεκτρική σκούπα που σκουπίζει το πάτωμα αυτόνομα.

Δεν έχουμε συνηθίσει να σκεφτόμαστε ότι οι περισσότερες από τις συσκευές μας παίρνουν τις δικές τους αποφάσεις, αφού τις θεωρούμε πολύ απλά πράγματα, αλλά βασίζονται σε πολύπλοκες τεχνολογίες. Για παράδειγμα, τα πλυντήρια ρούχων είναι σε θέση να αποφασίζουν για την ποσότητα νερού που χρειάζονται τα ρούχα μας ανιχνεύοντας πόσο βρώμικα είναι, ή να ελέγχουν την ταχύτητα περιστροφής του βαρελιού για την ελαχιστοποίηση της κατανάλωσης ενέργειας χρησιμοποιώντας συλλογισμούς που βασίζονται στην ασαφή λογική. Επίσης, δεν έχουμε

συνηθίσει να θεωρούμε την κάμερά μας ρομπότ, αλλά είναι σε θέση να χρησιμοποιεί τους αισθητήρες της για να ανιχνεύει στοιχεία στην εικόνα, όπως πρόσωπα, χρησιμοποιώντας όραση υπολογιστή ή να αποφασίζει ποια αντικείμενα στην εικόνα πρέπει να έχουν την εστίαση του φακού.

Οι ρομποτικές ηλεκτρικές σκούπες πρέπει να χρησιμοποιούν πιο σύνθετες τεχνικές, πρέπει να χαρτογραφούν το περιβάλλον τους από τις πληροφορίες των αισθητήρων τους και να σχεδιάζουν τη διαδρομή που επιτρέπει τον αποτελεσματικότερο καθαρισμό και να αποφασίζουν τη συμπεριφορά των ενεργοποιητών τους προκειμένου να καθαρίζουν αποτελεσματικά.

- **Industry 4.0 (Βιομηχανία 4.0)**

Η εισαγωγή των ρομπότ στη μεταποίηση είναι πλέον μια παλιά είδηση, αλλά υπάρχουν και άλλες πιθανές εφαρμογές. Μπορούμε να δούμε μια μονάδα παραγωγής ως ένα πολύ μεγάλο ρομπότ με πολλούς αισθητήρες που επιτρέπουν την παρακολούθηση όλων των δραστηριοτήτων του, με επίσης μεγάλο αριθμό μηχανισμών ενεργοποίησης που εκτελούν την εργασία κατασκευής. Αυτό σημαίνει ότι όλες οι τεχνικές που χρησιμοποιούνται στη ρομποτική μπορούν επίσης να εφαρμοστούν στην κατασκευή, συμπεριλαμβανομένης της όρασης υπολογιστών, της μηχανικής μάθησης, του σχεδιασμού και της συλλογιστικής.

Τα επιμέρους στοιχεία του εργοστασίου μπορούν επίσης να θεωρηθούν ως πολλαπλοί πράκτορες που χρειάζονται συντονισμό και να επιλύσουν τον ανταγωνισμό για τους πόρους, απαιτώντας επίσης τεχνικές από την κατανοημένη τεχνητή νοημοσύνη.

Η διαχείριση και ο προγραμματισμός της αλυσίδας διανομής και των αποθεμάτων επεκτείνουν τη ρομποτική άποψη πέρα από το εργοστάσιο σε όλη την αλυσίδα εφοδιασμού. Αυτό προσθέτει τη δυνατότητα χρήσης αυτόματου σχεδιασμού και συλλογισμού, συμπεριλαμβανομένης της χρήσης τεχνικών μηχανικής μάθησης για την ανίχνευση και εκμάθηση προτύπων από την παρατηρούμενη συμπεριφορά.

Μια τελευταία εφαρμογή στη μεταποίηση, αλλά και σε άλλους τομείς της βιομηχανίας, είναι η δυνατότητα εκτέλεσης προληπτικής συντήρησης με τη χρήση τεχνικών μηχανικής μάθησης ικανών να προβλέπουν το χρόνο και το είδος των βλαβών των στοιχείων του εργοστασίου.

- **Επιτήρηση και ασφάλεια**

Πρόκειται για πολύ γενικούς όρους που περιλαμβάνουν από την ασφάλεια στον κυβερνοχώρο ή τη βιομετρία μέχρι τη βιντεοεπιτήρηση. Όλοι αυτοί οι τομείς επωφελούνται από τις τεχνικές μηχανικής μάθησης και την όραση υπολογιστών.

Η βιομετρία βασίζεται συνήθως στην αναγνώριση ενός ατόμου σε πληροφορίες που αντιστοιχούν σε εικόνες ή φωνή. Αυτά τα συστήματα αναγνώρισης πρέπει να εκπαιδεύονται χρησιμοποιώντας αλγόριθμους μηχανικής μάθησης από δεδομένα. Η βιντεοεπιτήρηση μπορεί να αυτοματοποιηθεί για την παρακολούθηση και την αναγνώριση διαφόρων ειδών αντικειμένων με τη χρήση της όρασης υπολογιστή, από αριθμούς πινακίδων ή ανθρώπους έως πιθανές απειλές σε ευαίσθητα περιβάλλοντα.

Η ασφάλεια στον κυβερνοχώρο χρησιμοποιεί επίσης τη μηχανική μάθηση για να ανιχνεύσει ανώμαλες συμπεριφορές σε υπολογιστές ή δίκτυα που θα μπορούσαν να προκληθούν από ιούς υπολογιστών, κακόβουλο λογισμικό ή μη εξουσιοδοτημένες προσβάσεις. Περισσότερο προς την πλευρά της επιστημονικής φαντασίας, η προγνωστική αστυνόμευση χρησιμοποιεί αλγόριθμους

μηχανικής μάθησης πρόβλεψης για την πρόληψη του εγκλήματος με την οργάνωση των περιπολιών της αστυνομίας μέσα σε μια γειτονιά ή μια πόλη.

- **Συστήματα βασισμένα στη γνώση**

Πρόκειται για μια από τις πρώτες εφαρμογές της τεχνητής νοημοσύνης. Όταν άρχισαν να επεκτείνονται πέρα από τον ακαδημαϊκό τομέα, ήταν για την ανάπτυξη συστημάτων ικανών να λαμβάνουν αποφάσεις για προβλήματα σε εξαιρετικά εξειδικευμένους τομείς στο ίδιο επίπεδο με έναν άνθρωπο εμπειρογνώμονα, όπως για παράδειγμα η ιατρική διάγνωση ή η αναζήτηση πετρελαίου. Ονομάστηκαν Έμπειρα ή Εξειδικευμένα συστήματα.

Σήμερα τα συστήματα αυτά ονομάζονται συστήματα βασισμένα στη γνώση. Η κατασκευή αυτών των συστημάτων απαιτεί μια πολύπλοκη προσπάθεια αναπαράστασης της γνώσης, για την τυποποίηση των πληροφοριών του εξειδικευμένου τομέα, τη χρήση πολλαπλών μηχανισμών συλλογιστικής, όπως κανόνες παραγωγής ή προσεγγιστικές μεθόδους συλλογιστικής, όπως τα δίκτυα Bayes. Για ορισμένους τομείς, τμήματα της γνώσης μπορούν επίσης να ληφθούν με τη χρήση τεχνικών μηχανικής μάθησης που παράγουν μοντέλα τα οποία μπορούν να προστεθούν στη διαδικασία συλλογισμού.

Είναι αδύνατο να απαριθμήσει κανείς τον αριθμό των συστημάτων βασισμένων στη γνώση που χρησιμοποιούνται σήμερα για την αυτοματοποίηση διαδικασιών που ήταν αδύνατο ή πολύ δύσκολο να αυτοματοποιηθούν, ή για την παροχή συμβουλών στους ανθρώπους ώστε να μπορούν να λαμβάνουν καλύτερες αποφάσεις.

- **Γλώσσες για τεχνητή νοημοσύνη**

Η εκβιομηχάνιση της Τεχνητής Νοημοσύνης είχε ως αποτέλεσμα την υιοθέτηση διαφορετικών γλωσσών και τεχνολογιών για την ανάπτυξη εφαρμογών. Διαφορετικοί υποτομείς της τεχνητής νοημοσύνης έχουν ιδιαίτερες ανάγκες και άλλοι χρησιμοποιούν γλώσσες που βρίσκονται στην επικρατούσα τάση για την ανάπτυξη λογισμικού.

Η ανάπτυξη και η παραγωγή είναι ξεχωριστά πράγματα. Όλες οι εφαρμογές τεχνητής νοημοσύνης χρειάζονται μια φάση πρωτοτυποποίησης, πριν τα αποτελέσματα περάσουν στη φάση όπου μπορούν να ενσωματωθούν σε μια εφαρμογή που απευθύνεται στην αγορά. Αυτό σημαίνει συνήθως ότι σε αυτές τις φάσεις χρησιμοποιούνται διαφορετικές γλώσσες και τεχνολογίες.

Πολλές εξειδικευμένες γλώσσες στην ιστορία της τεχνητής νοημοσύνης αναπτύχθηκαν ταιριάζοντας με τις ανάγκες συγκεκριμένων τομέων ή με παραδείγματα προγραμματισμού που διευκόλυναν την ανάπτυξη ευφύων συστημάτων. Υπάρχουν γλώσσες με μακρά ιστορία, όπως η PROLOG, η οποία βασίζεται στη λογική και την επίλυση προβλημάτων όπως η απόδειξη θεωρημάτων, και η LISP, η οποία είναι μια λειτουργική γλώσσα που διευκολύνει την εργασία με συμβολικές αναπαραστάσεις και όχι με αριθμητικούς υπολογισμούς. Αυτές οι γλώσσες χρησιμοποιούνται ακόμη, αλλά δεν είναι κυρίαρχες στην τρέχουσα επανάσταση στις εφαρμογές τεχνητής νοημοσύνης

Περιπτώσεις χρήσης

AMAZON GO APP

Η αυτοματοποίηση του τρόπου με τον οποίο κάνετε τα καθημερινά σας ψώνια ήταν ένα από τα όνειρα των μεγάλων αλυσίδων σούπερ μάρκετ. Η επίλυση αυτού του προβλήματος θα μειώσει τον χρόνο για την αγορά, εξαλείφοντας τις μεγάλες ουρές στο ταμείο και θα μειώσει τις απώλειες λόγω κλοπών. Η ιδέα αυτή έχει αναπτυχθεί σε ορισμένα καταστήματα σούπερ μάρκετ από την Amazon με την ονομασία Amazon Go. Τα καταστήματα αυτά είναι γεμάτα με αισθητήρες και κάμερες που είναι σε θέση να παρακολουθούν τους χρήστες στο εσωτερικό τους και να ανιχνεύουν τη συμπεριφορά τους. Η σχετική τεχνολογία είναι παρόμοια με αυτή που απαιτείται για την ανάπτυξη αυτόνομων οχημάτων.

Το πρώτο πρόβλημα που πρέπει να επιλυθεί είναι η λήψη πληροφοριών από τους χρήστες που μπορούν να χρησιμοποιηθούν για την παρακολούθησή τους μέσα στο κατάστημα. Κάθε χρήστης πρέπει να κάνει check in με την εφαρμογή Amazon Go. Είναι εκείνη τη στιγμή που το σύστημα πρέπει να αναγνωρίσει τον χρήστη, λαμβάνοντας διάφορες εικόνες όπου να εξάγει από ένα σύνολο χαρακτηριστικών. Αυτό σημαίνει ότι πρέπει να τμηματοποιήσει ένα άτομο από αυτές τις εικόνες, να λάβει διάφορες μετρήσεις και να τις συσχετίσει με τον χρήστη. Λάβετε υπόψη ότι αυτές οι μετρήσεις πρέπει να είναι ανεκτικές σε πολλά πράγματα, ιδίως στα ρούχα, που δυσχεραίνουν τη λήψη ορισμένων από τις μετρήσεις. Η τμηματοποίηση εικόνων με χρήση υπολογιστικής όρασης θα είναι απαραίτητη, αλλά θα πρέπει να χρησιμοποιηθούν πολλές εικόνες για να μειωθεί η αβεβαιότητα. Αυτή η εργασία πρέπει να μαθευτεί από παραδείγματα, διότι δεν μπορεί να προγραμματιστεί χειροκίνητα. Αυτό σημαίνει ότι τα μοντέλα για την επίλυση αυτής της εξαγωγής χρειάζονται μηχανική μάθηση. Πρέπει επίσης να εφαρμοστεί κάποια συλλογιστική με χρήση της αβεβαιότητας για να συνδυαστούν διάφορες αποφάσεις. Αυτά τα χαρακτηριστικά θα χρησιμεύσουν ως μοντέλο του χρήστη.

Αυτές οι πληροφορίες θα χρησιμοποιηθούν για την αναγνώριση του χρήστη μέσα στο κατάστημα. Οι κάμερες και οι αισθητήρες επιλέγουν πολλούς χρήστες ταυτόχρονα, οπότε τα μοντέλα που πρέπει να χρησιμοποιηθούν πρέπει να εξάγουν τις πληροφορίες από όλους τους χρήστες ταυτόχρονα και να τις συγκρίνουν με τις πληροφορίες που εξάγονται όταν ο χρήστης εισέρχεται στο κατάστημα. Κάθε πληροφορία πρέπει να διαχωρίζεται για κάθε χρήστη και να ταιριάζει, ώστε το σύστημα να μπορεί να συνθέσει πού βρίσκεται ο κάθε χρήστης στο κατάστημα. Οι κάμερες δεν μπορούν να τοποθετηθούν σε οποιοδήποτε σημείο, οι χρήστες πρέπει να ανιχνεύονται από διαφορετικές γωνίες, πρέπει να λαμβάνονται υπόψη όλα τα στοιχεία του καταστήματος και δεν θα ταιριάζουν απόλυτα όλες οι πληροφορίες που εξάγονται με τις αρχικές πληροφορίες για τον χρήστη. Αυτό σημαίνει επίσης ότι πρέπει να εφαρμοστούν αλγόριθμοι υπολογιστικής όρασης που μαθαίνονται με τη χρήση μηχανικής μάθησης και οι αποφάσεις τους να συγκεντρώνονται με τη χρήση συλλογισμού με αβέβαιες πληροφορίες.

Το πιο πολύπλοκο έργο είναι να ανιχνεύσει πότε ένας χρήστης επιλέγει ένα προϊόν. Τα περισσότερα προϊόντα στο κατάστημα είναι επισημασμένα με μοτίβα ώστε να μπορούν να ανιχνευθούν και να αναγνωριστούν, και επίσης υπάρχουν αισθητήρες που ανιχνεύουν πότε ένα προϊόν σηκώνεται, αλλά το να αποφασιστεί πότε ένας χρήστης έχει πραγματικά σηκώσει κάτι είναι πολύπλοκο. Υπάρχουν πολλαπλές περιστάσεις που πρέπει να ληφθούν υπόψη. Ένας χρήστης μπορεί, για παράδειγμα, να πάρει κάτι και στη συνέχεια να αλλάξει γνώμη, τοποθετώντας το προϊόν ξανά στη θέση του (ή όχι). Το προϊόν μπορεί να είναι κρυμμένο από την κάμερα επειδή βρίσκεται πολύ κοντά στον χρήστη ... Το να αποφασίσει κανείς πότε πραγματικά αγοράζει κάτι είναι δύσκολο.

Τέλος, το σύστημα πρέπει να ανιχνεύει πότε ο χρήστης φεύγει από το κατάστημα, συγκεντρώνοντας όλες τις συμπεριφορές που ανιχνεύονται από τον χρήστη και αποφασίζοντας τον κατάλογο των προϊόντων που θα χρεωθούν.

Amazon Go

<https://youtu.be/NrmMk1Myrxc>

Αναφορές

- ArtificialIntelligence: 101 Things You Must Know Today About Our Future-Lasse Rouhiainen, Create Space Independent Publishing Platform, 2018. ISBN: 978-1982048808

Μάθημα 1.6: Ηθικές επιπτώσεις της τεχνητής νοημοσύνης

Θέματα προς συζήτηση

- Ιδιωτικότητα
- Αλγοριθμική δικαιοσύνη
- Λογοδοσία
- Σύνεση

Σημειώσεις διάλεξης

- **Κίνητρα**

Τα ανθρώπινα όντα έχουν αναπτύξει ηθικούς κανόνες που διαμορφώνουν την κοινωνία μας. Ένας πιθανός ορισμός της ηθικής είναι ένα σύνολο κανόνων που εμείς, ως κοινωνία, αποφασίζουμε ότι είναι καλό να ακολουθούμε εθελοντικά, επειδή κάνουν τον κόσμο μας καλύτερο. Πολλοί από αυτούς τους κανόνες είναι αρκετά παλιοί και οικουμενικοί, όπως η δέσμευσή μας να μη σκοτώνουμε άλλους ανθρώπους. Άλλοι είναι σχετικά νέοι, όπως η αντίθεση στη δουλεία. Σε πολλές περιπτώσεις οι ηθικοί κανόνες καταλήγουν να αντικατοπτρίζονται σε εθνικούς νόμους. Ωστόσο, είναι σημαντικό να σημειωθεί ότι οι ηθικοί κανόνες δεν χρειάζεται να είναι πολύ ακριβείς, καθώς πολύ συχνά αποτελούν απλώς κατευθυντήριες γραμμές συμπεριφοράς (π.χ. είναι ανήθικο να λέμε ψέματα στους φίλους μας). Επίσης, τους ακολουθούμε οικειοθελώς, πιθανότατα επειδή θέλουμε να αισθανόμαστε καλά και να προβάλλουμε μια καλή εικόνα του εαυτού μας. Από την άλλη πλευρά, οι νόμοι πρέπει να είναι πολύ πιο ακριβείς προκειμένου να ελαχιστοποιούνται οι διαφορετικές ερμηνείες. Επίσης, ο νόμος επιβάλλεται και πρέπει να τον ακολουθούμε ανεξάρτητα από το αν μας αρέσει ή όχι, γιατί διαφορετικά θα τιμωρηθούμε.

Ένα ηθικό δίλημμα είναι μια κατάσταση στην οποία πρέπει να γίνει μια επιλογή μεταξύ δύο τρόπων δράσης, καθένας από τους οποίους συνεπάγεται την παραβίαση μιας ηθικής αρχής ή προκαλεί κάποια βλάβη. Είναι σημαντικό να έχουμε κάποια ηθικά πρότυπα για να λάβουμε την καταλληλότερη απόφαση.

Οι τεχνολογικές εξελίξεις δημιουργούν συχνά νέες ηθικά ευαίσθητες καταστάσεις. Επειδή οι νόμοι χρειάζονται κάποιο χρόνο για να προσαρμοστούν στα μεταβαλλόμενα περιβάλλοντα, είναι σημαντικό να υπάρχει ηθική καθοδήγηση. Κατά πάσα πιθανότητα, αυτό που υπαγορεύει η ηθική θα αντικατοπτρίζεται αργότερα στο νόμο.

Η τεχνητή νοημοσύνη επηρεάζει δραματικά τη ζωή μας και υπάρχει μια κοινωνική αξίωση αυτή η πρόοδος να γίνεται ακολουθώντας αυστηρές ηθικές αρχές. Ορισμένες εταιρείες τεχνολογίας προσλαμβάνουν ηθικολόγους για να τις βοηθήσουν στις επιχειρηματικές τους διαδικασίες, οι ερευνητές της τεχνητής νοημοσύνης σηκώνουν κάποιες σημαίες σχετικά με τον τρόπο χειρισμού των επερχόμενων τεχνολογιών και οι ευαισθητοποιημένοι πολίτες πιέζουν τους πολιτικούς να τηρούν και να προστατεύουν τα πολιτικά δικαιώματα. Συνολικά, οι ηθικές πτυχές της τεχνητής νοημοσύνης αποτελούν επί του παρόντος μια ανοιχτή συναρπαστική συζήτηση.

Το μάθημα αυτό απευθύνεται σε επαγγελματίες ΤΠΕ και μπορεί να πιστεύουν ότι η δεοντολογία δεν τους αφορά. Οι διευθύνοντες σύμβουλοι και άλλα στελέχη είναι εκείνοι που λαμβάνουν στρατηγικές αποφάσεις και εκείνοι που θα πρέπει να ανησυχούν για την ηθική αντιμετώπιση των πραγμάτων. Αυτό μπορεί να ισχύει εν μέρει, αλλά οι επαγγελματίες ΤΠΕ δεν μπορούν να αποφύγουν το μερίδιο ευθύνης που τους αναλογεί. Σύμφωνα με τα λόγια της Cherri M. Pancake, της προέδρου της ACM:

"Οι επαγγελματίες της πληροφορικής είναι η πρώτη γραμμή άμυνας ενάντια στην κακή χρήση της τεχνολογίας. Η συλλογική μας κατανόηση των υπολογιστικών συστημάτων μας φέρνει σε θέση να προστατεύουμε ευαίσθητες πληροφορίες και να διασφαλίζουμε ότι τα συστήματα ενσωματώνονται με τρόπους που είναι κατάλληλοι, ασφαλείς και αξιόπιστοι. Η κοινωνία πρέπει να είναι σίγουρη ότι είμαστε προσηλωμένοι στην ηθική συμπεριφορά ως θεμέλιο της εργασίας μας. Η ανάγκη αυτή έχει καταστεί προσωπική ευθύνη κάθε επαγγελματία του κλάδου μας"

Σε αυτό το μάθημα θα αναφέρουμε ορισμένα θέματα ηθικού ενδιαφέροντος. Για μια πιο εμπεριστατωμένη ανάλυση σας προτρέπουμε να επισκεφθείτε την ιστοσελίδα "Η ηθική της τεχνητής νοημοσύνης: Issues and Initiatives", ένα έγγραφο της Υπηρεσίας Ερευνών του Ευρωπαϊκού Κοινοβουλίου.

- **Αλγοριθμική δικαιοσύνη και αλγοριθμική μεροληψία**

Αναπτύσσεται όλο και περισσότερο λογισμικό για την καθοδήγηση της λήψης αποφάσεων. Η μηχανική μάθηση χρησιμοποιείται για την εκμάθηση αποτελεσμάτων από προηγούμενες περιπτώσεις και υποθέτει ότι οι νέες περιπτώσεις θα παρουσιάζουν παρόμοια μοτίβα. Σε πολλούς τομείς αυτοί οι αλγόριθμοι εκτελούνται με ελάχιστη ή καθόλου ανθρώπινη κατανόηση και η ποιότητα του αποτελέσματός τους εξαρτάται με πολύ ασαφή τρόπο από τις αποφάσεις σχεδιασμού που λαμβάνουν οι μηχανικοί λογισμικού ή από την ποιότητα των δεδομένων που χρησιμοποιούνται για την εκπαίδευση του αλγορίθμου.

Αυτοί οι αλγόριθμοι για τη λήψη αποφάσεων μπορεί να έχουν σημαντικό αντίκτυπο στους ανθρώπους, όπως ο καθορισμός της αποδοχής τραπεζικών δανείων, η χορήγηση αποφυλακίσεων με εγγύηση, η επιλογή θέσεων εργασίας κ.λπ.

Είναι γνωστό ότι οι άνθρωποι έχουν προκαταλήψεις που επηρεάζουν τη λήψη των αποφάσεών μας και περιμένουμε από αυτούς που λαμβάνουν αποφάσεις να είναι σε θέση να είναι αντικειμενικοί. Τυπικές μεταβλητές προκατάληψης είναι η φυλή, το φύλο, ο σεξουαλικός προσανατολισμός και, ανάλογα με το πλαίσιο, επίσης η ηλικία, η εθνικότητα κ.λπ.

Λέμε ότι ένας αλγόριθμος είναι δίκαιος ή αμερόληπτος εάν αντιμετωπίζει πανομοιότυπα δύο διαφορετικά άτομα που διαφέρουν μόνο σε μια μεταβλητή προκατάληψης. Όταν οι αλγόριθμοι τεχνητής νοημοσύνης άρχισαν να χρησιμοποιούνται για την καθοδήγηση της λήψης αποφάσεων, γενικά πιστευόταν ότι θα ήταν δίκαιοι, επειδή εκπαιδεύονται με ψυχρά δεδομένα και δεν θα έπρεπε να υποφέρουν από την κοινωνική μας προκατάληψη. Ωστόσο, δεν άργησε να παρατηρηθεί ότι αυτό δεν ίσχυε. Ορισμένα γνωστά παραδείγματα περιλαμβάνουν ρατσιστές εισηγητές σωφρονιστικών παροχών ή σεξιστές εισηγητές υποψηφίων για κενές θέσεις εργασίας.

Οι πρώτες προσπάθειες επίλυσης του προβλήματος αφορούσαν απλώς την εξάλειψη των κρίσιμων μεταβλητών, αλλά αποδείχθηκε ότι οι αλγόριθμοι μπορούσαν να συμπεράνουν αυτές τις μεταβλητές με μεγάλη ακρίβεια χρησιμοποιώντας άλλες φαινομενικά αβλαβείς μεταβλητές (π.χ.,

σε ορισμένες περιπτώσεις μπορεί κανείς να μαντέψει με μεγάλη ακρίβεια τη φυλή ή τη θρησκεία ενός ατόμου από τον ταχυδρομικό του κώδικα).

Ορισμένοι ερευνητές υποστηρίζουν ότι η προκατάληψη μπορεί να προέρχεται από τους προγραμματιστές, οι οποίοι μολύνουν ακούσια τον αλγόριθμο μέσω σχεδιαστικών αποφάσεων με τις δικές τους προκαταλήψεις. Άλλες φορές είναι η κακή ποιότητα των δεδομένων που χρησιμοποιούνται για την εκπαίδευση του αλγορίθμου που προκαλεί την προκατάληψη. Επειδή πολύ συχνά τα δεδομένα εξάγονται από το Διαδίκτυο και ο όγκος των διαθέσιμων δεδομένων είναι τεράστιος, οι μηχανικοί τείνουν να πιστεύουν ότι είναι μια καλή αναπαράσταση της πραγματικότητας, αλλά είναι γνωστό ότι το περιεχόμενο του Διαδικτύου είναι ιδιαίτερα μεροληπτικό ως προς την ηλικία, το φύλο, τη φυλή και πολλές άλλες ευαίσθητες μεταβλητές.

Πολλοί ερευνητές προσπαθούν εντατικά να δημιουργήσουν τις θεωρητικές βάσεις προκειμένου να εγγυηθούν αμερόληπτους αλγορίθμους, αλλά υπάρχουν ακόμη πολλές δυσκολίες που πρέπει να επιλυθούν. Εν τω μεταξύ, πρέπει να έχουμε επίγνωση του προβλήματος και να κάνουμε ό,τι μπορούμε για να το ελαχιστοποιήσουμε.

• **Ιδιωτικότητα και καπιταλισμός της επιτήρησης**

Καθημερινά αλληλεπιδρούμε όλο και περισσότερο με τον παγκόσμιο ιστό, και αυτή η αλληλεπίδραση αφήνει τεράστιες ποσότητες προσωπικών πληροφοριών που μπορούν εύκολα να συγκεντρωθούν από εταιρείες τεχνολογίας. Το έξυπνο ηχείο σας μπορεί να γνωρίζει αν έχετε αμυγδαλίτιδα, το ρομπότ καθαρισμού σας μπορεί να γνωρίζει αν χρειάζεστε έναν καναπέ, η έξυπνη τηλεόρασή σας μπορεί να γνωρίζει αν είστε λυπημένος, ο διαχειριστής των φωτογραφιών σας μπορεί να γνωρίζει αν η τηλεόραση στο σαλόνι σας είναι παλιά, και αυτά είναι απλά παραδείγματα. Από τη διασταύρωση δεδομένων και τη χρήση αλγορίθμων τεχνητής νοημοσύνης, είναι εύλογο ότι κάποια εταιρεία έχει ένα αρχείο για εσάς που περιέχει με μεγάλη ακρίβεια την ηλικία σας, τον σεξουαλικό σας προσανατολισμό, τις πολιτικές σας ιδέες, ακόμη και το ψυχολογικό σας προφίλ.

Ενώ οι περισσότεροι άνθρωποι πιστεύουν ότι είμαστε οι πελάτες των εταιρειών τεχνολογίας των οποίων τα προϊόντα χρησιμοποιούμε (όπως τα μέσα κοινωνικής δικτύωσης, τα προγράμματα περιήγησης στο διαδίκτυο, οι εφαρμογές κινητών τηλεφώνων), κάποιοι έχουν αρχίσει να υποστηρίζουν ότι δεν είμαστε οι πελάτες αλλά το εμπόρευμα με το οποίο αυτές οι εταιρείες βγάζουν κέρδος. Στην πραγματικότητα, ορισμένοι συγγραφείς υποστηρίζουν ότι τα προσωπικά δεδομένα έχουν γίνει ο πιο πολύτιμος πόρος στη γη.

Η Shoshana Zuboff επινόησε τον όρο καπιταλισμός επιτήρησης για να περιγράψει αυτό το νέο σενάριο με τρία βασικά συστατικά: επιθετική συλλογή δεδομένων με ελάχιστη ευαισθητοποίηση των χρηστών, αλγόριθμοι τεχνητής νοημοσύνης ικανοί να εκμεταλλευτούν αυτά τα δεδομένα και να μειώσουν την αβεβαιότητα ανακαλύπτοντας τις πιθανότητες της μελλοντικής συμπεριφοράς των ανθρώπων, οικοσύστημα εταιρειών πρόθυμων να δομήσουν και να αξιοποιήσουν αυτά τα δεδομένα.

Υπάρχουν πολλές ηθικά αμφισβητήσιμες πτυχές πίσω από αυτό το νέο επιχειρηματικό μοντέλο, αλλά θα αναφέρουμε δύο:

Δεν γνωρίζουν πολλοί άνθρωποι τον όγκο των δεδομένων που δίνουν και τον όγκο των δεδομένων που μπορούν να εξαχθούν (με μεγάλη ακρίβεια) για αυτούς. Επίσης, έχουμε πολύ λίγες γνώσεις σχετικά με την "αξία" των δεδομένων μας, οπότε δεν υπάρχει συσχέτιση μεταξύ της αξίας της υπηρεσίας που λαμβάνει ο χρήστης και της αξίας των δεδομένων που δίνει ως αντάλλαγμα.

Τα δεδομένα των χρηστών χρησιμοποιούνται κυρίως για διαφημιστικούς σκοπούς και η διαφήμιση αφορά ουσιαστικά τη χειραγώγηση της συμπεριφοράς μας. Ενώ η διαφήμιση μας συνοδεύει εδώ και αιώνες και τις περισσότερες φορές δεν τη θεωρούμε ανήθικη, πρέπει να γνωρίζουμε ότι στην ιδιαίτερα διασυνδεδεμένη κοινωνία μας δίνουμε στις εταιρείες πολύ περισσότερα πλεονεκτήματα για αυτή τη χειραγώγηση. Δεν είναι στο ίδιο ηθικό επίπεδο να δείχνεις προσθήκες αθλητικού εξοπλισμού σε κάποιον που ο αλγόριθμος ΤΝ πιστεύει ότι είναι λάτρης του αθλητισμού, όπως το να διαφημίζεις ένα δίκτυο τυχερών παιχνιδιών σε κάποιον που ο αλγόριθμος ΤΝ πιστεύει ότι έχει οικονομικά προβλήματα.

- **Αλγοριθμική λογοδοσία**

Οι αλγόριθμοι τεχνητής νοημοσύνης χρησιμοποιούνται όλο και περισσότερο για την καθοδήγηση της λήψης αποφάσεων και οι αποφάσεις αυτές έχουν αντίκτυπο στη ζωή μας. Ως εκ τούτου, είναι λογικό να περιμένουμε κάποιο επίπεδο λογοδοσίας για τις αποφάσεις αυτές. Εάν μια εταιρεία, ας πούμε μια τράπεζα, λάβει μια απόφαση, όπως η απόρριψη ενός αιτήματος δανείου, ο πελάτης έχει το δικαίωμα να έχει κάποιου είδους εξήγηση, καθώς διαφορετικά η απόφαση μπορεί να φανεί αυθαίρετη. Επίσης, θα πρέπει να είναι δυνατόν να θεωρηθεί η εταιρεία υπεύθυνη για τις αποφάσεις που λαμβάνονται από τους αλγορίθμους της με τον ίδιο τρόπο που είναι υπεύθυνη για τις αποφάσεις που λαμβάνονται από τους υπαλλήλους της. Προκειμένου να επιτευχθεί αυτό, οι επαγγελματίες της ΤΝ εργάζονται για το πώς θα επιβληθούν οι ακόλουθες αρχές:

- **Επεξήγηση.** Οι αλγόριθμοι τεχνητής νοημοσύνης θα πρέπει να παρέχουν κάποιο είδος υποστήριξης για τις αποφάσεις τους. Η υποστήριξη πρέπει να είναι κατανοητή από τους ανθρώπους χωρίς να χρειάζεται να γνωρίζουν τις λεπτομέρειες του αλγορίθμου.
- **Διαφάνεια.** Οι εταιρείες δεν μπορούν να κρύβονται πίσω από την πολυπλοκότητα ή την ιδιόκτητη φύση του αλγορίθμου για να αποφύγουν την παροχή εξηγήσεων.
- **Έλεγχος.** Οι αλγοριθμικές τεχνικές, η ακρίβεια, η ευρωστία θα πρέπει να ελέγχονται από ανεξάρτητο τρίτο μέρος. Αυτό απαιτεί πρότυπα ποιότητας και μεθοδολογίας

- **Σύνεση**

Δεν υπάρχει καμία αμφιβολία σχετικά με την τεράστια επιτυχία των συστημάτων τεχνητής νοημοσύνης σε διάφορους τομείς και τον τρόπο με τον οποίο αυτή η επιτυχία έχει δημοσιευθεί στα μέσα μαζικής ενημέρωσης. Εκτός από τα πολλά άρθρα που αναφέρουν την επιτυχηθείσα επιτυχία, πολλά άλλα εικάζουν για τη μελλοντική επιτυχία. Οι εταιρείες και οι ερευνητές μπορεί να μπουκ στον πειρασμό να υπερβάλουν σχετικά. Για τις εταιρείες είναι καλό μάρκετινγκ να εξηγούν τα καταπληκτικά χαρακτηριστικά των σημερινών προϊόντων τους και τα ακόμη πιο καταπληκτικά χαρακτηριστικά των προϊόντων που πρόκειται να έρθουν σύντομα. Για τους ερευνητές είναι μια καλή στρατηγική για να εξασφαλίσουν τη χρηματοδότηση του έργου τους: η επιτυχία του παρελθόντος δίνει αξιοπιστία για τη μελλοντική επιτυχία. Εξάλλου, επειδή το ευρύ κοινό δεν κατανοεί την επιστημονική και τεχνική γραφή, είναι δελεαστικό να χρησιμοποιούνται υπεραπλουστευτικές μεταφορές που πολύ συχνά φαίνονται υπερβολικά αισιόδοξες.

Για το λόγο αυτό, ορισμένοι ερευνητές και επαγγελματίες της ΤΝ συνιστούν να καταβάλλεται επιπλέον προσπάθεια για να γνωστοποιούνται επίσης οι περιορισμοί της τεχνολογίας. Τα σημερινά συστήματα ΤΝ μπορεί να λειτουργούν καλά στην πλειονότητα των περιπτώσεων, αλλά

οι χρήστες πρέπει να γνωρίζουν ότι η πλήρης ορθότητά τους δεν έχει ελεγχθεί πλήρως. Επίσης, πολλά θεμελιώδη προβλήματα της τεχνητής νοημοσύνης δεν έχουν ακόμη επιλυθεί και θα απαιτήσουν ριζικές ανακαλύψεις, ενώ ίσως ορισμένα ζητήματα να μην επιλυθούν ποτέ. Αυτό το αίτημα για σύνεση εμφανίζεται στη Διακήρυξη της Βαρκελώνης για την ορθή ανάπτυξη και χρήση της τεχνητής νοημοσύνης.

Ως παράδειγμα, αναλογιστείτε την αναγνώριση προσώπου, η οποία είναι μια από τις πιο επιτυχημένες εφαρμογές της τεχνητής νοημοσύνης. Μια πρόσφατη μελέτη της αμερικανικής κυβέρνησης ανέλυσε σχεδόν 200 αλγόριθμους και παρατήρησε ότι έχουν χειρότερες επιδόσεις σε μη λευκά πρόσωπα. Η επίγνωση αυτού του περιορισμού είναι σημαντική, δεδομένου ότι τέτοιοι αλγόριθμοι μπορούν να χρησιμοποιηθούν για την επιβολή του νόμου, τον έλεγχο των συνόρων και άλλες εφαρμογές σε ολόκληρη την κοινωνία.

Ένα άλλο παράδειγμα είναι τα αυτοκινούμενα αυτοκίνητα. Το 2015 ο Guardian προέβλεψε 10 εκατομμύρια αυτοοδηγούμενα αυτοκίνητα στους δρόμους μέχρι το 2020, Αλλά το έτος αυτό έφτασε - και τα αυτοοδηγούμενα αυτοκίνητα δεν είναι. Υπάρχουν πολλά τεχνικά και νομικά προβλήματα που πρέπει ακόμη να επιλυθούν. Επομένως, πότε θα κυκλοφορήσουν στους δρόμους μας; Ορισμένοι ειδικοί λένε ότι αρκετά σύντομα, ενώ άλλοι λένε ότι ποτέ.

Περιπτώσεις χρήσης

"ΣΕΞΙΣΤΙΚΟ" ΕΡΓΑΛΕΙΟ ΠΡΟΣΛΗΨΗΣ ΑΙ

Το 2014, η Amazon ανέπτυξε ένα σύστημα τεχνητής νοημοσύνης για να εξετάζει τα βιογραφικά των υποψηφίων για εργασία και να επιλέγει αυτόματα τους καλύτερους υποψηφίους. Το σύστημα κατέτασσε τους υποψηφίους από ένα έως πέντε αστέρια. Το 2015 η εταιρεία συνειδητοποίησε ότι το σύστημα δεν κατέτασσε με ουδέτερο ως προς το φύλο τρόπο.

Ο λόγος ήταν ότι το σύστημα είχε εκπαιδευτεί να παρατηρεί μοτίβα με τους υπαλλήλους κατά την τελευταία 10ετία. Τα περισσότερα προέρχονταν από άνδρες, δεδομένου ότι ο κλάδος της τεχνολογίας κυριαρχείται σε μεγάλο βαθμό από άνδρες. Έτσι, το σύστημα έμαθε ότι οι άνδρες υποψήφιοι ήταν καλύτεροι και τιμωρούσε τα βιογραφικά που περιείχαν τη λέξη "γυναίκες". Ακόμη και αν το φύλο δεν περιλαμβανόταν ρητά στα δεδομένα, μπορούσε εύκολα να συναχθεί από πολλά βιογραφικά. Για παράδειγμα, ο αλγόριθμος κατέληγε να τιμωρεί τους υποψηφίους που ανέφεραν ότι συμμετείχαν σε λέσχες μαγειρικής (κυρίως γυναίκες) ή ότι σπούδασαν σε κολέγια μόνο για γυναίκες. Το 2016 η εταιρεία εγκατέλειψε το έργο επειδή δεν μπορούσε εύκολα να διορθώσει την προκατάληψή του!

Για περισσότερες πληροφορίες:

<https://www.reuters.com/article/us-amazon-com-jobs-automation-insight-idUSKCN1MK08G>

Αναφορές

- Ethics of Artificial Intelligence and Robotics – Vincent Müller, The Stanford Encyclopedia of Philosophy, Summer 2020

- The Ethics of Artificial Intelligence- European Parliament, 2020, ISBN: 978-92-846-5799-5 (www.europarl.europa.eu)
- Policy and Investment Recommendations for Trustworthy Artificial Intelligence ec.europa.eu/digital-single_market/en/news/policy-and-investment-recommendations-trustworthy-artificial-intelligence
- Machine Bias (<https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing>)
- Explainer: what is surveillance capitalism and how it shapes our economy (<https://theconversation.com/explainer-what-is-surveillance-capitalism-and-how-does-it-shape-our-economy-119158>)
- Its 2020: where are our self-driving cars? (<https://www.vox.com/future-perfect/2020/2/14/21063487/self-driving-cars-autonomous-vehicles-waymo-cruise-uber>)
- Barcelona Declaration for the proper development and usage of Artificial Intelligence (www.iiia.csic.es/barcelonadeclaration)
- IEEE STANDARDS ASSOCIATION, Ethically Aligned Design: standards.ieee.org/industryconnections/ec/autonomoussystem.html.
- A US government study confirms most face recognition systems are racist. MIT Technology Review, dec 2020 (<https://www.technologyreview.com/2019/12/20/79/ai-face-recognition-racist-us-government-nist-study/>)

Ενότητα 2: Εισαγωγή στη Μηχανική Μάθηση

Μάθημα 2.1: Εισαγωγή στη Μηχανική Μάθηση

Θέματα προς συζήτηση

- Εισαγωγή
- Θεμέλια της επιβλεπόμενης μηχανικής μάθησης
- Αξιολόγηση της επιβλεπόμενης μηχανικής μάθησης

Σημειώσεις διάλεξης

- **Εισαγωγή**

Για να λύσουμε μια εργασία με έναν υπολογιστή, συνήθως γράφουμε έναν αλγόριθμο που καθορίζει όλα τα βήματα που πρέπει να γίνουν για την επίλυσή της. Όσο πιο δύσκολη είναι η εργασία, τόσο λιγότερο εφικτό είναι να προγραμματιστεί ένας τέτοιος αλγόριθμος, και στο ακραίο, υπάρχουν εργασίες όπου είναι τόσο δύσκολες που είναι σχεδόν αδύνατο να τις κάνουμε σωστά. Για αυτές τις εργασίες, μπορούμε να καταφύγουμε στη χρήση δεδομένων αντί της κατανόησης και της γνώσης μας για το πρόβλημα. Μπορούμε να συλλέξουμε εκατοντάδες ή χιλιάδες παραδείγματα της εργασίας και να ορίσουμε έναν αλγόριθμο ικανό να επεξεργαστεί αυτά τα δεδομένα και να βρει τον αλγόριθμο για την εργασία από μόνος του, ένα μοντέλο. Ίσως το μοντέλο να μην είναι τέλει και μερικές φορές να μην είμαστε σε θέση να το κατανοήσουμε, αλλά θα είναι μια χρήσιμη προσέγγιση που μπορούμε να εφαρμόσουμε στην εργασία.

Αυτή η κατάσταση συμβαίνει όλο και πιο συχνά. Δεν μπορούμε να γράψουμε ένα πρόγραμμα ικανό να αναγνωρίσει ένα πρόσωπο, αλλά μπορούμε να συλλέξουμε πολλές εικόνες και να πούμε σε έναν υπολογιστή πού βρίσκεται το πρόσωπο στην εικόνα. Είναι δύσκολο για έναν αναλυτή να αποφασίσει ποια χαρακτηριστικά κάνουν έναν πελάτη πιο πρόθυμο να αγοράσει ένα προϊόν, αλλά έχουμε εκατομμύρια αρχεία αγορών που ένας αλγόριθμος μπορεί να εξερευνήσει για να βρει μοτίβα που θα μπορούσαν να αποτελέσουν χρήσιμο μυρμήγκι που ο αναλυτής μπορεί να ερμηνεύσει και να κατανοήσει.

Υπάρχουν πολλά διαφορετικά προβλήματα που μπορούν να επιλυθούν με τη χρήση της μηχανικής μάθησης, αυτό σημαίνει ότι υπάρχουν αλγόριθμοι για διαφορετικούς σκοπούς. Μπορούμε να ταξινομήσουμε τους τομείς της μηχανικής μάθησης με διαφορετικούς τρόπους:

- **Μάθηση με επίβλεψη:** Αυτοί οι αλγόριθμοι υποθέτουν ότι έχουμε δεδομένα για την εργασία μας και κάθε παράδειγμα εισόδου έχει μια έξοδο που αντιστοιχεί στη σωστή απάντηση. Ο στόχος είναι να βρεθεί ένα μοντέλο, που περιγράφεται με τη χρήση ενός συγκεκριμένου φορμαλισμού, ικανό να γενικεύει από τα παραδείγματα και να συσχετίζει σωστά την είσοδο και την έξοδο. Η επιτυχία θα μετρηθεί από την ικανότητα να κάνει προβλέψεις για παραδείγματα που δεν έχουν χρησιμοποιηθεί για τη δημιουργία του μοντέλου. Υπάρχουν δύο πιθανές εργασίες σε αυτόν τον τομέα. Μπορούμε να έχουμε ως έξοδο ένα περιορισμένο σύνολο τιμών, που θα ονομάσουμε ετικέτες, πρόκειται για

ταξινόμηση, ή μπορούμε να έχουμε ως έξοδο έναν πραγματικό αριθμό, πρόκειται για παλινδρόμηση.

- **Μάθηση χωρίς επίβλεψη:** Αυτοί οι αλγόριθμοι δεν έχουν μια έξοδο που πρέπει να προβλεφθεί. Χρησιμοποιούν μόνο τις πληροφορίες από τα παραδείγματα εισόδου και ο στόχος είναι να ανακαλύψουν μοτίβα που σχετίζονται με την ομοιότητα ή τη διαφορετικότητα των τιμών τους. Υπάρχουν διάφορες εργασίες, η πιο συνηθισμένη είναι η ομαδοποίηση, η οποία αποκτά μια κατανομή των δεδομένων εισόδου, συγκεντρώνοντας παραδείγματα που έχουν υψηλή ομοιότητα και διαχωρίζοντας τα δεδομένα σε ομάδες που είναι αρκετά διαφορετικές ώστε να αντιστοιχούν σε διαφορετικά πρότυπα.
- **Μάθηση με ημιεπίβλεψη:** Αυτοί οι αλγόριθμοι χρησιμοποιούν ένα μείγμα επισημειωμένων και μη επισημειωμένων δεδομένων. Η ύπαρξη επισημειωμένων δεδομένων είναι δαπανηρή και χρονοβόρα και μερικές φορές τα μη επιβλεπόμενα δεδομένα μπορούν να αξιοποιηθούν για τη βελτίωση της επιβλεπόμενης μάθησης με τη χρήση ενός επιβλεπόμενου μοντέλου για την επισήμανση των μη επιβλεπόμενων δεδομένων και την επανεκπαίδευση του μοντέλου ή με τη χρήση επιβλεπόμενων πληροφοριών για τη βελτίωση των εργασιών ομαδοποίησης.
- **Ενισχυτική μάθηση:** Αυτοί οι αλγόριθμοι είναι εξειδικευμένοι στον τομέα της επίλυσης προβλημάτων. Ένας πράκτορας επιλύει ένα έργο σε ένα περιβάλλον χρησιμοποιώντας ένα σύνολο ενεργειών, για παράδειγμα ένα ρομπότ που μαθαίνει να πλοηγείται μέσα σε ένα δωμάτιο ή που πρέπει να μάθει να μαζεύει και να μετακινεί αντικείμενα. Το έργο επιλύεται με δοκιμή και σφάλμα και το περιβάλλον δίνει ανατροφοδότηση στον αλγόριθμο ανάλογα με την απόδοσή του, μια θετική ή αρνητική ανταμοιβή. Ο αλγόριθμος διαδίδει την ανατροφοδότηση στις αποφάσεις προκειμένου να τις βελτιώσει στο μέλλον.

Όλοι αυτοί οι αλγόριθμοι είναι **επαγωγικοί**, δηλαδή εργάζονται από το ειδικό (τα δεδομένα) στο γενικό (το μοντέλο), επιτυγχάνοντας μια αναπαράσταση που γενικεύει τις πληροφορίες εισόδου. Αυτό σημαίνει ότι ένα μοντέλο μπορεί να ακυρωθεί με τη χρήση αντιπαραδειγμάτων. Δεδομένου ότι δεν μπορούμε να χρησιμοποιήσουμε όλα τα πιθανά δεδομένα, το μοντέλο είναι ελλιπές και προσεγγίζει μόνο την πραγματικότητα. Υπάρχουν **αλγόριθμοι απαγωγικής (αφαιρετικής) μάθησης**. Αυτοί χρησιμοποιούν ένα μοντέλο της εργασίας και μαθαίνουν πώς να το βελτιώσουν κάνοντάς το πιο αποτελεσματικό. Υπάρχουν κάποιοι εξειδικευμένοι τομείς όπου χρησιμοποιούνται αυτά τα μοντέλα, αλλά οι επαγωγικοί αλγόριθμοι είναι αυτοί που χρησιμοποιούνται στις περισσότερες τρέχουσες εφαρμογές.

Μια διαφορετική ταξινόμηση είναι με βάση τον τρόπο με τον οποίο το μοντέλο επιλύει την εργασία. Υπάρχουν **διακριτικοί αλγόριθμοι**, που μαθαίνουν μόνο ένα μοντέλο που είναι σε θέση να διαχωρίσει τα διάφορα προβλήματα της εργασίας, για παράδειγμα, πού είναι το όριο που διαχωρίζει τις διάφορες ετικέτες στην επιβλεπόμενη μάθηση και **γενετικοί (genetic) αλγόριθμοι** που είναι σε θέση να αποκτήσουν ένα μοντέλο που περιγράφει τα διάφορα προβλήματα. Αυτοί οι τελευταίοι αλγόριθμοι βασίζονται συνήθως σε πιθανοτικά μοντέλα.

• Θεμέλια της μηχανικής μάθησης με επίβλεψη

Η μηχανική μάθηση μπορεί να προσεγγιστεί από διαφορετικές οπτικές γωνίες, θα επικεντρωθούμε μόνο σε ένα υποσύνολο αλγορίθμων επαγωγικής μάθησης με επίβλεψη. Θα υποθέσουμε ότι η μάθηση θα πραγματοποιηθεί από παραδείγματα που περιγράφονται από ένα σύνολο χαρακτηριστικών. Αυτό σημαίνει ότι κάθε παράδειγμα θα είναι μια λίστα που για κάθε

χαρακτηριστικό έχει μια τιμή από ένα πιθανό σύνολο (ζεύγη χαρακτηριστικών-τιμών). Τα χαρακτηριστικά μπορεί να είναι συνεχή (πραγματικές τιμές) ή διακριτά (διατεταγμένες ή μη διατεταγμένες ετικέτες). Κάθε παράδειγμα έχει συσχετισμένη μια έξοδο που μπορεί να είναι από ένα σύνολο διακριτών τιμών (ετικέτα), ο στόχος είναι η ταξινόμηση ή μια πραγματική τιμή, ο στόχος είναι η παλινδρόμηση.

Κάθε αλγόριθμος μηχανικής μάθησης θα ορίζει μια γλώσσα αναπαράστασης. Αυτός είναι ο τρόπος με τον οποίο το μοντέλο θα περιγράφει την εργασία. Μπορεί να είναι για παράδειγμα ένα σύνολο λογικών τύπων με ορισμένους περιορισμούς, μπορεί να είναι μια οικογένεια συναρτήσεων (π.χ. πολυώνυμα), ή μπορεί να είναι ένα μείγμα κατανομής πιθανοτήτων. Καθορίζοντας τη γλώσσα αναπαράστασης, θα καθορίσουμε τα πιθανά μοντέλα που μπορούν να αναπαραστήσουν τα δεδομένα και τη συσχέτισή τους με την έξοδο.

Μπορούμε να διατυπώσουμε την εποπτευόμενη επαγωγική μάθηση ως πρόβλημα αναζήτησης. Θέλουμε να βρούμε στο χώρο όλων των πιθανών αναπαραστάσεων που η γλώσσα μας επιτρέπει μια συγκεκριμένη αναπαράσταση (ένα μοντέλο) ικανή να κάνει καλές προβλέψεις με βάση τα δεδομένα εισόδου και την έξοδό τους. Το πόσο καλή είναι η υπόθεση θα πρέπει να μετρηθεί ως το πόσο κοντά είναι οι έξοδοι του μοντέλου στις πραγματικές εξόδους των δεδομένων. Θα το μετρήσουμε αυτό ως μια συνάρτηση που ονομάζεται συνάρτηση απωλειών (ή συνάρτηση κόστους), η οποία μετρά το κόστος του λάθους. Έτσι μπορούμε να ορίσουμε επίσημα ένα πρόβλημα μηχανικής μάθησης ως εξής:

- Δεδομένου ενός συνόλου παραδειγμάτων που περιγράφονται από ζεύγη εισόδου-εξόδου που δημιουργούνται από μια άγνωστη συνάρτηση $y = f(x)$, όπου μπορούμε να αποδώσουμε μια πιθανότητα σε κάθε ζεύγος $P(x,y)$
- Δεδομένου ενός χώρου υποθέσεων $H=\{h(x;\alpha)|\alpha\in\Pi\}$, όπου Π είναι ένα σύνολο παραμέτρων
- και μια συνάρτηση απώλειας $L(y,h(x;\alpha))$, το κόστος ενός σφάλματος
- να ανακαλύψετε μια συνάρτηση h_α που προσεγγίζει την f όσο το δυνατόν καλύτερα

Κάθε μέθοδος μηχανικής μάθησης θα επιλύσει αυτό το πρόβλημα χρησιμοποιώντας διαφορετικό χώρο υποθέσεων και διαφορετική στρατηγική αναζήτησης. Στόχος είναι η ελαχιστοποίηση του εμπειρικού σφάλματος που διαπράττει το μοντέλο. Το πραγματικό σφάλμα ενός μοντέλου h μπορεί να υπολογιστεί ως εξής:

$$E_{true}(h_a) = \int L(y, h_a(x)) P(x, y) dx dy$$

Δηλαδή, το μέσο σφάλμα για όλα τα παραδείγματα με δεδομένη την πιθανότητά τους στο πρόβλημα. Επειδή θα εργαστούμε μόνο με ένα δείγμα όλων των πιθανών παραδειγμάτων, μπορούμε να υπολογίσουμε το εμπειρικό σφάλμα ως εξής:

$$E_{emp}(h_a) = \frac{1}{n} \sum_{(x,y)} L(y, h_a(x))$$

Ο στόχος είναι η επίλυση:

$$h'_a = \arg \min_{h_a \in H} E_{emp}(h_a)$$

Αυτό είναι που ονομάζεται εμπειρική ελαχιστοποίηση του κινδύνου.

Η συνάρτηση απώλειας μπορεί να είναι διαφορετική ανάλογα με τον στόχο μάθησης. Για την ταξινόμηση, η συνήθης είναι η απώλεια 0/1, η οποία προσθέτει στο κόστος την τιμή 1 εάν η έξοδος του μοντέλου διαφέρει από τη σωστή έξοδο και 0 εάν είναι η ίδια. Για την παλινδρόμηση, μπορεί να χρησιμοποιηθεί η απώλεια τετραγωνικού σφάλματος, δηλαδή το άθροισμα των τετραγωνικών διαφορών μεταξύ της εξόδου του μοντέλου και της πραγματικής εξόδου των δεδομένων και η απώλεια απόλυτου σφάλματος που υπολογίζει την απόλυτη τιμή της διαφοράς. Υπάρχουν περισσότερες συναρτήσεις απωλειών και μερικές φορές η επιλογή εξαρτάται από τη διατύπωση του αλγορίθμου.

Η επιλογή του χώρου των υποθέσεων H εξαρτάται επίσης από το πρόβλημα. Όσο πιο πολύπλοκη είναι η συνάρτηση που θέλουμε να μάθουμε τόσο περισσότερες δυνατότητες πρέπει να παρέχει αυτός ο χώρος αναπαράστασης. Αυτό μπορεί να σημαίνει ότι το μέγεθος της υπόθεσης μπορεί να είναι πολύ μεγάλο για να διερευνηθεί ή πολύ απλό για να προσεγγίσει σωστά το πρόβλημα. Η ελαχιστοποίηση του εμπειρικού σφάλματος θα μπορούσε επίσης να είναι ανεπαρκής για να γίνει μια αποτελεσματική αναζήτηση, οπότε οι αλγόριθμοι χρησιμοποιούν προκαταλήψεις ή μειώνουν τον χώρο αναζήτησης επιβάλλοντας ορισμένες ιδιότητες στη συνάρτηση που πρέπει να μάθουμε. Η συνήθης μεροληψία είναι η αναζήτηση του απλούστερου μοντέλου που ταιριάζει στα δεδομένα, μετρώντας την πολυπλοκότητα ως προς το πόσες παράμετροι απαιτούνται για τον ορισμό του μοντέλου ή το μέγεθος των τιμών τους. Αυτή η ανάγκη επιβολής μιας προκατάληψης για τη μείωση του υπολογιστικού κόστους οδηγεί στο θεώρημα **no free lunch** στη μηχανική μάθηση. Κάθε αλγόριθμος είναι κατάλληλος για μια σειρά διαφορετικών προβλημάτων, αλλά δεν υπάρχει ένας καλύτερος αλγόριθμος, ένας που να αποδίδει τέλεια για κάθε πιθανό πρόβλημα.

Δύο προβλήματα παρουσιάζονται κατά την εκμάθηση ενός μοντέλου από δεδομένα. Μπορούμε να έχουμε ένα μοντέλο που δεν είναι σε θέση να προσαρμόσει τα δεδομένα επειδή η καλύτερη υπόθεση είναι πολύ αδύναμη για την πολυπλοκότητα των δεδομένων, αυτό ονομάζεται **υποπροσαρμογή (underfitting)**, δεν μπορούμε να μειώσουμε το σφάλμα επειδή δεν μπορούμε να προσεγγίσουμε την πραγματική συνάρτηση αρκετά καλά. Μπορούμε επίσης να έχουμε και την αντίθετη περίπτωση, μπορούμε να έχουμε πολύ χαμηλό σφάλμα επειδή το μοντέλο που έχουμε βρει ταιριάζει πολύ καλά στα δεδομένα, αυτό ονομάζεται **υπερπροσαρμογή (overfitting)**.

• Αξιολόγηση της επιβλεπόμενης μηχανικής μάθησης

Η διαδικασία εκμάθησης ενός μοντέλου από δεδομένα με τη χρήση ενός συγκεκριμένου αλγορίθμου μηχανικής μάθησης ξεκινά με την επιλογή ενός δείγματος από το πρόβλημα και στη συνέχεια με τη χρήση του αλγορίθμου (διαδικασία εκπαίδευσης). Αυτό θα οδηγήσει σε μοντέλα που μπορούμε να χρησιμοποιήσουμε για τη μέτρηση του σφάλματος. Δεδομένου ότι χρησιμοποιούμε ένα δείγμα από όλα τα πιθανά παραδείγματα του προβλήματος, δεν μπορούμε

να χρησιμοποιήσουμε ολόκληρο το δείγμα για να αξιολογήσουμε πόσο καλό είναι το μοντέλο που μάθαμε. Πρέπει να ακολουθήσουμε μια ακριβή πειραματική διαδικασία για να έχουμε μια δίκαιη μέτρηση του σφάλματος και να έχουμε κάποιες εγγυήσεις ότι η ακρίβεια του μοντέλου θα είναι αντιπροσωπευτική αυτού που θα έχουμε όταν το χρησιμοποιήσουμε με νέα δεδομένα. Αυτό μειώνει επίσης το πρόβλημα της υπερβολικής προσαρμογής στα δεδομένα.

Η τυπική διαδικασία για την εκπαίδευση μοντέλων μηχανικής μάθησης από δεδομένα ξεκινά με το διαχωρισμό των δεδομένων σε τρία δείγματα:

- Τα δεδομένα εκπαίδευσης θα είναι το μεγαλύτερο υποδείγμα και θα χρησιμοποιηθούν για την εκπαίδευση του μοντέλου.
- Τα δεδομένα επικύρωσης θα χρησιμοποιηθούν για τη ρύθμιση των υπερπαραμέτρων του αλγορίθμου μηχανικής μάθησης, οι υπερπαραμέτροι αυτές είναι τα χαρακτηριστικά που χρησιμοποιεί ο αλγόριθμος για την προκατάληψη του μοντέλου, όπως για παράδειγμα πόσο πολύπλοκος είναι ο χώρος υποθέσεων. Τα δεδομένα αυτά δεν χρησιμοποιούνται για εκπαίδευση.
- Τα δεδομένα δοκιμής θα χρησιμοποιηθούν για τον υπολογισμό του τελικού σφάλματος, του σφάλματος γενίκευσης, του σφάλματος που υποτίθεται ότι θα κάνει το μοντέλο με νέα δεδομένα.

Η διαδικασία για την εκτίμηση των υπερπαραμέτρων του μοντέλου χρησιμοποιώντας τα δεδομένα εκπαίδευσης και επικύρωσης μπορεί να γίνει με διαφορετικούς τρόπους ανάλογα με τον όγκο των διαθέσιμων δεδομένων.

Μια απλή διαδικασία είναι να γίνει ο διαχωρισμός μία φορά χρησιμοποιώντας μια συγκεκριμένη αναλογία, όπως 80%/10%/10%. Αυτό συνήθως οδηγεί σε πολύ αισιόδοξο σφάλμα. Μια στρατηγική επαναλαμβανόμενου διαχωρισμού των δεδομένων εκπαίδευσης και επικύρωσης είναι συνήθως η καλύτερη επιλογή.

Η στρατηγική **leave one out** ξεκινά με τη διαίρεση των δεδομένων σε δεδομένα εκπαίδευσης/επιβεβαίωσης και δεδομένα δοκιμής με μια συγκεκριμένη αναλογία. Η εκπαίδευση συνεχίζεται με την εκμάθηση ενός μοντέλου χρησιμοποιώντας όλα τα δεδομένα εκπαίδευσης/επιβεβαίωσης εκτός από ένα παράδειγμα, το οποίο χρησιμοποιείται για τον υπολογισμό του σφάλματος του μοντέλου. Αυτό επαναλαμβάνεται για όλα τα παραδείγματα. Το μέσο εκτιμώμενο σφάλμα χρησιμοποιείται για τη ρύθμιση των υπερπαραμέτρων και για την εκτίμηση του σφάλματος για ένα μοντέλο με όλα τα δεδομένα. Αυτό είναι δαπανηρό, επειδή το μοντέλο πρέπει να εκπαιδευτεί τόσες φορές όσα και τα παραδείγματα στα δεδομένα εκπαίδευσης/επιβεβαίωσης, αλλά είναι δυνατό για ορισμένους αλγόριθμους να το υπολογίσουν χωρίς να κατασκευάσουν όλα τα μοντέλα. Αυτή η μέθοδος είναι χρήσιμη όταν το μέγεθος των δεδομένων δείγματος που είναι διαθέσιμα για εκμάθηση είναι μικρό. Το τελικό σφάλμα υπολογίζεται χρησιμοποιώντας τα δεδομένα δοκιμής.

Η μέθοδος διασταυρούμενης επικύρωσης **k-fold (k-fold cross validation)** είναι παρόμοια. Στην περίπτωση αυτή τα δεδομένα εκπαίδευσης/επιβεβαίωσης χωρίζονται σε k ξεχωριστά υποδείγματα (folds), συνήθως 10. Το μοντέλο κατασκευάζεται χρησιμοποιώντας $k-1$ αναδιπλώσεις και δοκιμάζεται σε αυτή που δεν χρησιμοποιείται. Αυτό επαναλαμβάνεται k φορές. Το εκτιμώμενο σφάλμα για το μοντέλο με όλα τα δεδομένα είναι το μέσο σφάλμα από όλα τα μοντέλα. Αυτό το

σφάλμα χρησιμοποιείται για τη ρύθμιση των παραμέτρων υπερβολής. Όταν το έργο είναι η ταξινόμηση, οι αναδιπλώσεις επιλέγονται έτσι ώστε η κατανομή των ετικετών να είναι η ίδια για κάθε αναδίπλωση, αυτό ονομάζεται στρωματοποιημένη διασταυρούμενη επικύρωση k-πτυχών (stratified k-fold cross validation). Το τελικό σφάλμα υπολογίζεται με τα δεδομένα δοκιμής.

Το σφάλμα για τα μοντέλα παλινδρόμησης μπορεί να υπολογιστεί χρησιμοποιώντας διάφορες συναρτήσεις. Οι συνήθεις είναι το **μέσο τετραγωνικό σφάλμα** (MSE) που είναι ο μέσος όρος των τετραγωνικών διαφορών μεταξύ της πραγματικής και της προβλεπόμενης τιμής και το **μέσο απόλυτο σφάλμα** (MAE) που είναι ο μέσος όρος της απόλυτης τιμής των διαφορών μεταξύ της πραγματικής και της προβλεπόμενης τιμής.

Το σφάλμα για την ταξινόμηση είναι συνήθως η απώλεια 0/1 (**0/1 Loss**). Αυτό το σφάλμα μπορεί να μετρηθεί ως, ο αριθμός των λανθασμένων προβλεπόμενων εξόδων διαιρεμένος με το συνολικό αριθμό παραδειγμάτων, ή η ακρίβεια ως, ο αριθμός των σωστών προβλεπόμενων εξόδων διαιρεμένος με το συνολικό αριθμό παραδειγμάτων. Τα αποτελέσματα αναπαρίστανται συνήθως ως πίνακας σύγχυσης, ένας πίνακας ενδεχομένων που μετρά τις αναθέσεις των παραδειγμάτων στις προβλεπόμενες και σωστές κλάσεις. Για την περίπτωση της δυαδικής ταξινόμησης (2 κλάσεις, θετική κλάση και αρνητική κλάση) υπάρχει ειδική ορολογία για τις συναρτήσεις που μπορούν να μετρήσουν την απόδοση ενός ταξινομητή.

	Προβλεπόμενη κλάση		
Αληθινή κλάση	Θετικά	Αρνητικά	Σύνολο
Θετική	Αληθινά θετικά (tp)	Ψευδώς αρνητικά (fn)	p
Αρνητική	Ψευδώς θετικά (fp)	Αληθινά αρνητικά (tn)	n
Σύνολο	p'	n'	N

Τα πραγματικά θετικά παραδείγματα είναι αυτά που ανήκουν στη θετική κατηγορία και προβλέπονται ως θετικά, τα ψευδώς θετικά παραδείγματα προβλέπονται ως θετική κατηγορία αλλά στην πραγματικότητα ανήκουν στην αρνητική κατηγορία κ.ο.κ. Το σφάλμα μπορεί να υπολογιστεί ως $(fp+fn)/N$, η ακρίβεια ως $(tp+tn)/N$. Μπορούμε να υπολογίσουμε το αληθώς θετικό ποσοστό ως tp/p και το ψευδώς θετικό ποσοστό ως fp/p . Δύο μέτρα που χρησιμοποιούνται για την αξιολόγηση της ποιότητας του ταξινομητή για κάθε κλάση είναι η ακρίβεια, δηλαδή ο αριθμός των παραδειγμάτων που έχουν προβλεφθεί σωστά για μια κλάση σε σχέση με όλα τα παραδείγματα που έχουν προβλεφθεί $(tp/tp+fp)$ ως αυτή την κλάση και η ανάκληση, ο λόγος των παραδειγμάτων μιας κλάσης που έχουν προβλεφθεί σωστά $(tp/tp+fn)$. Συνήθως, ένας καλός ταξινομητής χρειάζεται μια ισορροπία αυτών των μέτρων για όλες τις κλάσεις. Ένας συνδυασμός αυτών των μέτρων που δείχνει την ισορροπία τους είναι το F1-score που υπολογίζεται ως $2 * (precision * recall) / (precision + recall)$

Περιπτώσεις χρήσης

ΟΠΤΙΚΗ ΑΝΑΓΝΩΡΙΣΗ ΧΑΡΑΚΤΗΡΩΝ

Η οπτική αναγνώριση χαρακτήρων (OCR) είναι μια τεχνολογία που επιτρέπει τη σάρωση μιας σελίδας κειμένου και τη μετατροπή της σε επεξεργάσιμο έγγραφο. Η εργασία αυτή μπορεί να είναι περισσότερο ή λιγότερο δύσκολη εάν περιορίζεται μόνο στην αναγνώριση γραμματοσειρών χαρακτήρων υπολογιστή ή ο στόχος είναι να μπορεί να αναγνωρίζει και χειρόγραφους χαρακτήρες. Ανεξάρτητα από τους περιορισμούς, είναι προφανές ότι είναι αδύνατο να γραφτεί χειροκίνητα ένα πρόγραμμα ικανό να λάβει υπόψη όλες τις πιθανές γραμματοσειρές που μπορούν να χρησιμοποιηθούν σε ένα έγγραφο και επίσης να λάβει υπόψη όλα τα πιθανά πράγματα που μπορεί να συμβούν κατά τη σάρωση μιας σελίδας με κείμενο, όπως εικόνες χαμηλής ποιότητας, σκόνη στην επιφάνεια σάρωσης ή κεκλιμένους χαρακτήρες.

Είναι σαφές ότι αυτό είναι ένα έργο που μπορεί να επιλυθεί μόνο με τη χρήση μηχανικής μάθησης. Υπάρχουν διάφορα μοντέλα που πρέπει να μαθευτούν από τα δεδομένα για να είναι επιτυχής αυτή η εργασία. Αν περιορίσουμε τη δυσκολία μόνο στην αναγνώριση γραμματοσειρών χαρακτήρων υπολογιστών, πρέπει πρώτα να είμαστε σε θέση να αναγνωρίσουμε ποια είναι η γραμματοσειρά που χρησιμοποιείται στο έγγραφο, ώστε να μπορούμε να χρησιμοποιήσουμε ένα συγκεκριμένο μοντέλο για τους συγκεκριμένους χαρακτήρες. Αυτό σημαίνει ότι πρέπει να μάθουμε έναν ταξινομητή για γραμματοσειρές χρησιμοποιώντας χιλιάδες παραδείγματα για κάθε γραμματοσειρά. Για έναν συγκεκριμένο χαρακτήρα, χρειαζόμαστε επίσης ειδικά μοντέλα που πρέπει να εκπαιδευτούν επίσης από χιλιάδες παραδείγματα. Αυτά τα παραδείγματα δεν μπορεί να είναι τέλεια τυπωμένοι χαρακτήρες, πρέπει να λάβουμε υπόψη μας διαφορετικές γωνίες για τον χαρακτήρα και πιθανό θόρυβο, ώστε να μπορούν να αναγνωριστούν σε πολλές καταστάσεις.

Ακόμη και όταν οι χαρακτήρες αναγνωρίζονται, θα εμφανιστούν ορισμένα σφάλματα. Μπορούμε να χρησιμοποιήσουμε απλές τεχνικές για να διορθώσουμε ορισμένα από τα λάθη, όπως η χρήση ενός λεξικού και η αντιστοίχιση των αναγνωρισμένων λέξεων, όπως κάνουν ορισμένα λογισμικά ορθογραφίας, ή μπορούμε επίσης να χρησιμοποιήσουμε τη μηχανική μάθηση, ώστε να μπορούμε να κάνουμε προβλέψεις σχετικά με το ποια είναι η πιο πιθανή λέξη, χρησιμοποιώντας όχι μόνο τους χαρακτήρες που αναγνωρίζονται, αλλά και τις γύρω λέξεις, ώστε να μπορούμε να χρησιμοποιήσουμε το πλαίσιο για να ξεκαθαρίσουμε το λάθος.

Μπορούν να επιλυθούν πιο σύνθετες εργασίες σε εκτυπωμένα έγγραφα, όπως η αναγνώριση εικόνων και πινάκων, ώστε να μπορούν να προστεθούν στο έγγραφο όπως ήταν στο πρωτότυπο. Αυτό σημαίνει ότι πρέπει να υπάρχουν μοντέλα ικανά να αναγνωρίζουν αυτά τα στοιχεία σε μια σελίδα και να εξάγουν συγκεκριμένα χαρακτηριστικά που καθιστούν δυνατή τη σωστή ανακατασκευή τους.

Η αναγνώριση γραφής είναι πιο δύσκολη, τα γενικά μοντέλα μπορούν να μάθουν με συγκεκριμένους περιορισμούς, όπως για την αναγνώριση ταχυδρομικών κωδικών σε γράμματα, υποθέτοντας ότι μπορούμε να έχουμε μόνο αριθμούς. Μερικές φορές η αναγνώριση γραφικών χαρακτήρων απαιτεί την εκμάθηση ενός μοντέλου για κάθε άτομο. Αυτό επιλύεται σε πολλές εφαρμογές που εκτελούν αυτό το έργο με εκπαίδευση για ένα χρονικό διάστημα από το συγκεκριμένο άτομο που χρησιμοποιεί την εφαρμογή, ώστε το πρόγραμμα να μπορεί να εξατομικευτεί στον συγκεκριμένο γραφικό χαρακτήρα του χρήστη. Με τη διαθεσιμότητα όλο και

περισσότερων δεδομένων, ορισμένες εφαρμογές είναι σε θέση να αναγνωρίζουν ένα ευρύ φάσμα διαφορετικών γραφικών χαρακτήρων, αρκεί ο γραφικός χαρακτήρας να είναι αρκετά σαφής.

Μπορείτε να ερευνήσετε την ιστορία της αναγνώρισης OCR, είναι μια πολύ παλιά φιλοδοξία στην τεχνητή νοημοσύνη. Μπορείτε επίσης να αναζητήσετε διάφορες εφαρμογές που είναι σε θέση να κάνουν αναγνώριση OCR και γραφής και να ελέγξετε ποιες είναι οι δυνατότητές τους.

Google Jamboard

Αναφορές

- Artificial Intelligence: 101 Things You Must Know Today About Our Future-Lasse Rouhiainen, Create Space Independent Publishing Platform, 2018. ISBN: 978-1982048808

Μάθημα 2.2: Γλώσσες και πόροι

Θέματα προς συζήτηση

- Εισαγωγή
- Βιβλιοθήκες για χειρισμό δεδομένων
- Βιβλιοθήκες για μηχανική μάθηση
- Βιβλιοθήκες για οπτικοποίηση

Σημειώσεις διάλεξης

- Εισαγωγή

Η γλώσσα *rython* είναι μια γλώσσα υψηλού επιπέδου που επικεντρώνεται στην ποιότητα, την παραγωγικότητα και τη φορητότητα του λογισμικού. Είναι μια γλώσσα με χαμηλή δυσκολία εκμάθησης που διαθέτει μεγάλο αριθμό βιβλιοθηκών που επιτρέπουν την ανάπτυξη λογισμικού για σχεδόν κάθε σκοπό. Συντηρείται καλά και υποστηρίζεται από μια τεράστια κοινότητα.

Είναι μια διερμηνευμένη γλώσσα, οπότε το κύριο μειονέκτημα προέρχεται από την ταχύτητά της. Αυτό δεν αποτελεί πραγματικά πρόβλημα, δεδομένου ότι αυτή η γλώσσα που μπορεί να διασυνδεθεί με βιβλιοθήκες γραμμένες σε άλλες γλώσσες, όπως η C ή η C++, και οι περισσότερες από τις βιβλιοθήκες που χρησιμοποιούνται για κρίσιμες για το χρόνο εφαρμογές αναπτύσσονται σε αυτές τις γλώσσες, καθιστώντας την *rython* μια συγκολλητική γλώσσα που διευκολύνει την πρωτοτυποποίηση και την ανάπτυξη εφαρμογών, αξιοποιώντας άλλους πόρους.

Υπάρχουν και άλλες γλώσσες ανοικτού κώδικα, όπως για παράδειγμα η R, που παρέχουν επίσης βιβλιοθήκες και εργαλεία για τη μηχανική μάθηση. Επί του παρόντος υπάρχει διχασμός μεταξύ της κοινότητας της μηχανικής μάθησης όσον αφορά την επιλογή μεταξύ αυτών των δύο γλωσσών. Υπάρχει επίσης σύγκλιση μεταξύ των διαθέσιμων προϊόντων και για τα δύο οικοσυστήματα, οπότε στο μέλλον θα είναι απλώς θέμα προτίμησης. Η R έχει πιο απότομη καμπύλη εκμάθησης και η ανάπτυξη πολλών έργων μπορεί να σημαίνει τη χρήση πολλών διαφορετικών βιβλιοθηκών. Υπάρχουν επίσης εργαλεία και βιβλιοθήκες για άλλες κοινές γλώσσες όπως η *java* και η C++. Έχουν λιγότερο εκτεταμένες κοινότητες, αλλά χρησιμοποιούνται επίσης για την κατασκευή πρωτοτύπων και την ανάπτυξη μοντέλων μηχανικής μάθησης.

Για αυτό το μάθημα και το υπόλοιπο της ενότητας θεωρείται ότι υπάρχει μια ελάχιστη γνώση της γλώσσας *rython*, οπότε θα επικεντρωθούμε στο συγκεκριμένο οικοσύστημα βιβλιοθηκών που χρησιμοποιούνται για αριθμητικούς υπολογισμούς, αλγορίθμους μηχανικής μάθησης, πειραματισμό και οπτικοποίηση και παρουσίαση των αποτελεσμάτων.

Από όλες τις βιβλιοθήκες που χρησιμοποιούνται για την ανάπτυξη εφαρμογών μηχανικής μάθησης, θα επικεντρωθούμε στις πιο συχνά χρησιμοποιούμενες. Οι βιβλιοθήκες μπορούν να χωριστούν ανάλογα με την εστίασή τους σε:

- βιβλιοθήκες για την επεξεργασία δεδομένων, κυρίως για την ανάγνωση δεδομένων, την επιλογή και τον μετασχηματισμό δεδομένων και τη σύνοψη των πληροφοριών τους (numpy, pandas)
- βιβλιοθήκες για μηχανική μάθηση, κυρίως αφιερωμένες στην υλοποίηση των κύριων αλγορίθμων μηχανικής μάθησης, στην προηγμένη επεξεργασία και μετατροπή δεδομένων, καθώς και στον πειραματισμό και την επικύρωση μοντέλων (scikit-learn, stats-models).
- βιβλιοθήκες για οπτικοποίηση, κατά βάση αφιερωμένες για βασική και προηγμένη οπτικοποίηση δεδομένων και ανάλυση και παρουσίαση αποτελεσμάτων (matplotlib, seaborn)

Μια πλήρης παρουσίαση αυτών των βιβλιοθηκών δεν είναι στο πλαίσιο αυτού του μαθήματος και θα δοθεί μόνο μια γενική ανασκόπηση των αλγορίθμων και των εργαλείων που παρέχουν. Τα διάφορα στοιχεία που θα χρειαστούν θα παρουσιαστούν στις περιπτώσεις χρήσης και στις πρακτικές ασκήσεις των μαθημάτων αυτής της ενότητας. Όλες οι βιβλιοθήκες διαθέτουν εκτεταμένη τεκμηρίωση που διατίθεται ελεύθερα στο διαδίκτυο.

• Βιβλιοθήκες για χειρισμό δεδομένων

Η επιλογή, η φόρτωση και ο μετασχηματισμός των δεδομένων είναι η πρώτη εργασία σε κάθε έργο μηχανικής μάθησης. Είναι σημαντικό να χρησιμοποιείτε τα εργαλεία και τις βιβλιοθήκες που διευκολύνουν αυτές τις εργασίες. Για τους σκοπούς αυτούς χρησιμοποιούνται δύο κύριες βιβλιοθήκες στο οικοσύστημα μηχανικής μάθησης της python. Η πρώτη είναι η **numpy**, η οποία είναι η βιβλιοθήκη της python για αριθμητικούς υπολογισμούς. Πρόκειται για μια γενική βιβλιοθήκη που παρέχει δομές δεδομένων ισοδύναμες με συστοιχίες και πίνακες και όλες τις συναρτήσεις που επιτρέπουν τον χειρισμό τους. Η δεύτερη είναι η **pandas**, η οποία είναι μια βιβλιοθήκη εξειδικευμένη στον χειρισμό δεδομένων που επιτρέπει συγκεκριμένες λειτουργίες για την επεξεργασία δεδομένων και τη στατιστική ανάλυση.

i. Numpy

Η Numpy (www.numpy.org) είναι μια βιβλιοθήκη που ειδικεύεται στους επιστημονικούς υπολογισμούς. Αυτή η βιβλιοθήκη χρησιμοποιείται για κάθε τομέα της επιστήμης που χρειάζεται να χρησιμοποιεί αριθμητικούς αλγορίθμους, όπως είναι η περίπτωση της μηχανικής μάθησης. Έχει υλοποιήσει έναν μεγάλο αριθμό συναρτήσεων και αλγορίθμων, συμπεριλαμβανομένων των στοιχείων που χρησιμοποιούνται στους αλγορίθμους μηχανικής μάθησης, επιτρέποντας την ανάπτυξη αυτών των αλγορίθμων από τις πρώτες αρχές με τη χρήση μεθόδων γραμμικής άλγεβρας και στατιστικών συναρτήσεων. Δεδομένου ότι οι αλγόριθμοι αυτοί έχουν ήδη υλοποιηθεί σε άλλες βιβλιοθήκες δεν χρησιμοποιείται γενικά για το σκοπό αυτό, εκτός αν ο στόχος είναι η ανάπτυξη νέων αλγορίθμων.

Το κεντρικό στοιχείο της βιβλιοθήκης είναι ο πίνακας numpy (**numpy.array**). Πρόκειται για μια πολυδιάστατη δομή δεδομένων που μπορεί να χρησιμοποιηθεί για την αποθήκευση και τον χειρισμό δεδομένων. Οι πίνακες Numpy μπορούν να είναι **ομοιογενείς**, όλα τα στοιχεία του ίδιου τύπου, ή **ετερογενείς**, αλλά οι πράξεις συνήθως αφορούν αριθμητικούς τύπους δεδομένων.

Επιτρέπονται όλοι οι τυπικοί τύποι δεδομένων, από boolean δεδομένα μέχρι ακέραιους, float ή μιγαδικούς αριθμούς.

Αυτή η δομή δεδομένων μπορεί να δημιουργηθεί με τη μετατροπή λιστών της `python`, με τη φόρτωση δεδομένων από ένα αρχείο χρησιμοποιώντας διάφορες συναρτήσεις εισόδου/εξόδου που επιτρέπουν τη χρήση διαφόρων μορφών αρχείων δεδομένων ή με τη χρήση κάποιας από τις διάφορες μεθόδους για τη δημιουργία ειδικών πινάκων (π.χ. `numpy.zeros`). Οι πίνακες μπορούν να στοιβάζονται με διάφορους τρόπους για τη δημιουργία νέων πινάκων από υπάρχοντες πίνακες και μπορούν να μετασχηματίζονται αναδιατάσσοντας τις διαστάσεις τους. Η δεικτοδότηση των πινάκων χρησιμοποιεί την ίδια σύνταξη που χρησιμοποιεί η `python` για τις λίστες, αλλά επιτρέπει επίσης την επιλογή στοιχείων με τη χρήση boolean συνθηκών.

Οι πίνακες επιτρέπουν τις συνήθεις αριθμητικές πράξεις άθροισης, αφαίρεσης, γινομένου και διαίρεσης με κλιμακωτούς αριθμούς και πράξεις γραμμικής άλγεβρας μεταξύ πινάκων, όπως εσωτερικό και εξωτερικό γινόμενο ή πολλαπλασιασμό διανυσματικών πινάκων. Αυτό περιλαμβάνει επίσης άλλες πράξεις γραμμικής άλγεβρας όπως η αντιμετάθεση, ο υπολογισμός του προσδιοριστή, η αντιστροφή πινάκων ή η επίλυση συστημάτων γραμμικών εξισώσεων.

Το εύρος των λειτουργιών αυτής της βιβλιοθήκης είναι μεγάλο και περιλαμβάνει ένα ευρύ φάσμα στατιστικών λειτουργιών για τη δημιουργία δεδομένων σύμφωνα με κατανομές, περιγραφική στατιστική ή στατιστικό έλεγχο. Διαθέτει επίσης έναν μεγάλο κατάλογο μαθηματικών συναρτήσεων, συμπεριλαμβανομένων όλων των τριαδικών συναρτήσεων. Η βιβλιοθήκη αυτή επεκτείνεται από τη βιβλιοθήκη `scipy` που περιλαμβάνει πιο εξειδικευμένα εργαλεία και περισσότερες στατιστικές και βελτιστοποιητικές συναρτήσεις.

ii. Pandas

Το Pandas (pandas.pydata.org) είναι μια βιβλιοθήκη ανάλυσης και χειρισμού δεδομένων που είναι λιγότερο προσανατολισμένη σε αριθμητικούς υπολογισμούς από ό,τι το `numpy`. Βασίζεται σε δύο δομές δεδομένων που εξειδικεύονται σε σειρές δεδομένων (**`pandas.Series`**) και πίνακες δεδομένων (**`pandas.DataFrame`**) που μοιάζουν με τις δομές δεδομένων που χρησιμοποιούνται σε άλλες γλώσσες όπως η R ή σε στατιστικά λογισμικά όπως το SPSS ή το SAS. Επιτρέπει τη φόρτωση δεδομένων από μεγάλο αριθμό μορφών και πηγών δεδομένων, συμπεριλαμβανομένων των πιο ευρέως χρησιμοποιούμενων στην επιστήμη των δεδομένων. Οι δομές δεδομένων της `Python` μπορούν επίσης να μετατραπούν σε δομές δεδομένων της `pandas`.

Οι σειρές και οι πίνακες δεδομένων του Pandas μπορούν να είναι ετερογενών τύπων. Το εύρος των τύπων δεδομένων που μπορούν να χρησιμοποιηθούν σε αυτές τις δομές υπερβαίνει αυτούς που επιτρέπει η `numpy`. Περιλαμβάνει την επεξεργασία κειμένου, τύπους δεδομένων χρόνου και τη χρήση κατηγορικών δεδομένων. Επίσης, οι δυνατότητες ευρετηρίασης επεκτείνουν τη `numpy`, επιτρέποντας την αναφορά σε στήλες και γραμμές με βάση το όνομα και την εκτέλεση πιο σύνθετων ερωτημάτων. Το Pandas επιτρέπει τον μετασχηματισμό και τον χειρισμό των δομών δεδομένων του με τη χρήση λειτουργιών όπως οι βάσεις δεδομένων, με `join`, `merge` και `concatenation`, `pivoting` πινάκων δεδομένων και ομαδοποίηση τιμών μεταξύ άλλων.

Διαθέτει μεγάλο αριθμό συναρτήσεων για χειρισμό και μετασχηματισμό δεδομένων και διευκολύνει ορισμένες από τις συνήθεις εργασίες που γίνονται στην προεπεξεργασία δεδομένων, όπως η επανακωδικοποίηση, ο υπολογισμός ελλειπών τιμών ή ο μετασχηματισμός με χρήση αυθαίρετων συναρτήσεων. Περιλαμβάνει επίσης εξειδικευμένες συναρτήσεις για το χειρισμό χρονοσειρών. Οι συναρτήσεις NumPy μπορούν να χρησιμοποιηθούν για το μετασχηματισμό των δομών δεδομένων και ένας τύπος δομής δεδομένων μπορεί να μετασχηματιστεί σε έναν άλλο. Αυτή η βιβλιοθήκη περιλαμβάνει επίσης ορισμένα ενσωματωμένα εργαλεία οπτικοποίησης για την εκτέλεση γραφικών παραστάσεων των δεδομένων.

- **Βιβλιοθήκες για τη διαδικασία μηχανικής μάθησης**

Αυτές οι βιβλιοθήκες υποστηρίζουν την πλήρη διαδικασία μηχανικής μάθησης, από τα δεδομένα μέχρι τα τελικά μοντέλα. Η **Scikit-learn** είναι η πιο γενική βιβλιοθήκη και περιλαμβάνει όλα τα εργαλεία για την πλήρη διαδικασία και τους πιο ευρέως χρησιμοποιούμενους αλγορίθμους μηχανικής μάθησης. Η **Stats-model** έχει πιο εξειδικευμένα μοντέλα που σχετίζονται περισσότερο με τη στατιστική ανάλυση και μοντελοποίηση.

- i. **Scikit-learn**

Το Scikit-learn (scikit-learn.org) είναι η κύρια βιβλιοθήκη για τη μηχανική μάθηση με χρήση της python. Περιλαμβάνει όλους τους πιο ευρέως χρησιμοποιούμενους αλγορίθμους μηχανικής μάθησης και όλα τα εργαλεία για τον μετασχηματισμό δεδομένων και την εκπαίδευση μοντέλων. Αυτή η βιβλιοθήκη χρησιμοποιεί άμεσα τις δομές δεδομένων της numpy και μπορεί επίσης να χρησιμοποιήσει τις δομές δεδομένων της pandas με κάποιους περιορισμούς.

Η βιβλιοθήκη είναι οργανωμένη ανάλογα με τον τύπο του αλγορίθμου και το μέρος της διαδικασίας ανάπτυξης του μοντέλου μηχανικής μάθησης. Ακολουθώντας την τυπική διαδικασία μηχανικής μάθησης τα πρώτα στοιχεία της βιβλιοθήκης είναι οι συναρτήσεις για τη φόρτωση και τη δημιουργία δεδομένων. Οι δυνατότητες αυτής της βιβλιοθήκης για τη φόρτωση δεδομένων είναι περιορισμένες, αλλά οι numpy και pandas μπορούν να χρησιμοποιηθούν για το σκοπό αυτό. Διαθέτει συναρτήσεις για τη φόρτωση συνόλων δεδομένων από αποθετήρια ή τη δημιουργία δεδομένων με συγκεκριμένα χαρακτηριστικά.

Το δεύτερο βήμα στη διαδικασία δημιουργίας μοντέλων μηχανικής μάθησης είναι η προεπεξεργασία των δεδομένων. Αυτή η βιβλιοθήκη διαθέτει εργαλεία για την εκτέλεση ενός μεγάλου εύρους μετασχηματισμών και κανονικοποιήσεων στα δεδομένα, τη συμπλήρωση ελλειπών τιμών και τη δημιουργία χαρακτηριστικών για εξειδικευμένα δεδομένα όπως εικόνες ή κείμενο.

Το επόμενο βήμα αντιστοιχεί στη μείωση της διαστατικότητας και τον μετασχηματισμό, συμπεριλαμβανομένων αλγορίθμων αποσύνθεσης πινάκων και μεθόδων γραμμικής και μη γραμμικής μείωσης της διαστατικότητας για τον μετασχηματισμό του συνόλου δεδομένων ή για την οπτικοποίηση. Συμπληρωματικά, διαθέτει αλγορίθμους για την επιλογή χαρακτηριστικών.

Για τη διαδικασία εκπαίδευσης των μοντέλων μηχανικής μάθησης παρέχει αλγορίθμους βελτιστοποίησης υπερπαραμέτρων με διάφορες στρατηγικές που μπορούν να διερευνήσουν τις πιθανές τιμές που έχει ένας συγκεκριμένος αλγόριθμος μηχανικής μάθησης, προκειμένου να τον συντονίσουν και να επιτύχουν την καλύτερη απόδοση. Διαθέτει διάφορες στρατηγικές για την επικύρωση μοντέλων, συμπεριλαμβανομένων αυτών που εξηγήθηκαν στο προηγούμενο μάθημα. Τέλος, διαθέτει διαφορετικές μετρικές για την εκτίμηση του σφάλματος των μοντέλων που προκύπτουν με τη χρήση επιβλεπόμενων και μη επιβλεπόμενων αλγορίθμων μηχανικής μάθησης.

Για τις μεθόδους μηχανικής μάθησης, ταξινομούνται ανάλογα με τα χαρακτηριστικά του μοντέλου που μαθαίνουν. Περιλαμβάνει για την επιβλεπόμενη μάθηση: γραμμικά μοντέλα, πιθανοτικά μοντέλα (naive bayes), μοντέλα πλησιέστερου γείτονα, νευρωνικά δίκτυα, μηχανές διανυσμάτων υποστήριξης, δέντρα αποφάσεων και μοντέλα συνόλου. Για τη μάθηση χωρίς επίβλεψη περιλαμβάνει ένα ευρύ φάσμα αλγορίθμων. Ένα πλεονέκτημα αυτής της βιβλιοθήκης είναι ότι όλοι οι αλγόριθμοι χρησιμοποιούν την ίδια διεπαφή και ότι άλλες βιβλιοθήκες που επεκτείνουν τις δυνατότητες της scikit-learn συνήθως ακολουθούν αυτή την ομοιογένεια.

ii. Στατιστικά μοντέλα

Το Statsmodels (statsmodels.org) είναι μια βιβλιοθήκη για την εκτίμηση στατιστικών μοντέλων και τον στατιστικό έλεγχο. Περιλαμβάνει ορισμένους από τους αλγορίθμους του scikit-learn με πιο εξελιγμένες επιλογές. Συγκεκριμένα, περιλαμβάνει ένα μεγαλύτερο εύρος μοντέλων γραμμικής παλινδρόμησης, επιτρέποντας περισσότερες στατιστικές κατανομές, συμπεριλαμβανομένων ισχυρών μεθόδων για παλινδρόμηση και παλινδρόμηση με διακριτές μεταβλητές. Η βιβλιοθήκη είναι περισσότερο προσανατολισμένη στη στατιστική, οπότε περιλαμβάνει πιο εξειδικευμένους αλγορίθμους που χρησιμοποιούνται επίσης για εφαρμογές μηχανικής μάθησης και ανάλυσης δεδομένων, όπως μοντέλα για ανάλυση χρονοσειρών, μη παραμετρική στατιστική ή ανάλυση επιβίωσης. Επεκτείνει επίσης τις δυνατότητες στατιστικών δοκιμών της numpy.

iii. Βιβλιοθήκες για οπτικοποίηση

Υπάρχουν πολλές βιβλιοθήκες και εργαλεία για την οπτικοποίηση δεδομένων, αλλά η rython διαθέτει ορισμένες βιβλιοθήκες που είναι ενσωματωμένες με τις άλλες βιβλιοθήκες του οικοσυστήματος ανάλυσης δεδομένων και μηχανικής μάθησης. Η **Matplotlib** είναι πιο χαμηλού επιπέδου και πιο ευέλικτη. Το **Seaborn** διευκολύνει περισσότερο τη δημιουργία απλών και προηγμένων οπτικοποιήσεων.

iv. Matplotlib

Η Matplotlib (matplotlib.org) είναι μια βιβλιοθήκη για τη σχεδίαση και την οπτικοποίηση δεδομένων. Είναι μια πολύ ευέλικτη βιβλιοθήκη και επιτρέπει τις πιο συνηθισμένες γραφικές παραστάσεις που χρησιμοποιούνται για την ανάλυση και την παρουσίαση αποτελεσμάτων. Αυτό περιλαμβάνει 2D και 3D διαγράμματα δεδομένων, ιστογράμματα, διαγράμματα πυκνότητας, boxplots, barplots για να αναφέρουμε μερικά. Είναι ιδιαίτερα παραμετροποιήσιμη και μπορεί μερικές φορές να είναι δύσκολη η χρήση της κατά τη δημιουργία σύνθετων οπτικοποιήσεων

δεδομένων. Το Pandas χρησιμοποιεί αυτή τη βιβλιοθήκη στις λειτουργίες οπτικοποίησης μειώνοντας την πολυπλοκότητα της χρήσης για τα πιο συνηθισμένα γραφήματα.

v. Seaborn

Το Seaborn (seaborn.pydata.org) είναι μια βιβλιοθήκη βασισμένη στην matplotlib που μειώνει την πολυπλοκότητα της αναπαράστασης δεδομένων, παρέχοντας απλά και εξειδικευμένα διαγράμματα με απλούστερη διεπαφή. Είναι επίσης ιδιαίτερα παραμετροποιήσιμη, αλλά μειώνει το χρόνο για τη δημιουργία γραφημάτων και παρέχει με πολύ χρήσιμες και σύνθετες απεικονίσεις.

• Άλλες βιβλιοθήκες και εργαλεία

Το οικοσύστημα της επιστήμης των δεδομένων και της μηχανικής μάθησης της python αναπτύσσεται καθημερινά, παρέχοντας νέα εργαλεία στις τρεις πτυχές που αναφέραμε στην αρχή του μαθήματος. Όσον αφορά τις δομές δεδομένων για τον χειρισμό δεδομένων, οι ανάγκες για την εργασία με μεγάλες δομές δεδομένων και τη χρήση κατανεμημένων υπολογιστών καλύπτονται από βιβλιοθήκες που έχουν κατασκευαστεί πάνω ή αναπαράγουν τις διεπαφές και τις λειτουργίες των numpy και pandas όπως οι xarray και dask.

Από την πλευρά των αλγορίθμων μηχανικής μάθησης, οι μεγάλες εταιρείες του διαδικτύου έχουν επικεντρωθεί στην κατασκευή βιβλιοθηκών που εξειδικεύονται στα νευρωνικά δίκτυα και τη βαθιά μάθηση. Υπάρχουν πολλές βιβλιοθήκες για να διαλέξετε από τις οποίες μπορείτε να ξεπεράσετε αυτό που μπορεί να κάνει το scikit-learn σε αυτόν τον τομέα. Για παράδειγμα, οι Tensorflow και Keras από την Google ή το PyTorch από το Facebook είναι οι πιο συχνά χρησιμοποιούμενες βιβλιοθήκες για εφαρμογές που βασίζονται σε νευρωνικά δίκτυα. Άλλα μοντέλα μηχανικής μάθησης έχουν επίσης πιο εξειδικευμένες βιβλιοθήκες, για παράδειγμα για τα μοντέλα Bayes, με βιβλιοθήκες όπως το Stan ή το PyMC3.

Για την οπτικοποίηση, μπορούν επίσης να χρησιμοποιηθούν βιβλιοθήκες με πιο διαδραστικές δυνατότητες για πιο ελκυστική παρουσίαση των αποτελεσμάτων, όπως οι Bokeh, Plotly ή Dash.

Για την κατασκευή πρωτοτύπων, την εκτέλεση και την παρουσίαση των αποτελεσμάτων των πειραμάτων, υπάρχουν επίσης ορισμένα εργαλεία που χρησιμοποιούνται ευρέως, όπως για παράδειγμα τα σημειωματάρια jupyter.

Περιπτώσεις χρήσης

ΤΙΜΕΣ ΚΑΤΟΙΚΙΩΝ ΣΤΗΝ ΚΑΛΙΦΟΡΝΙΑ

Θα εξερευνήσουμε ένα γνωστό σύνολο δεδομένων που περιλαμβάνεται στη βιβλιοθήκη scikit-learn σχετικά με την πρόβλεψη της τιμής των κατοικιών στην Καλιφόρνια. Για αυτή την περίπτωση χρήσης, θα πραγματοποιήσουμε κάποια εξερεύνηση και μετασχηματισμό δεδομένων χρησιμοποιώντας τα pandas, matplotlib και seaborn. Για την αναπαραγωγή αυτής της περίπτωσης χρήσης, είναι προτιμότερο να αντιγράψετε τον κώδικα σε ένα σημειωματάριο jupyter, θα βρείτε

οδηγίες σχετικά με τον τρόπο εγκατάστασης και εκτέλεσής του σε αυτή την ιστοσελίδα <https://jupyter.org/>. Εναλλακτικά, μπορεί επίσης να χρησιμοποιηθεί ένας διερμηνέας ipython.

[Κατεβάστε τον κώδικα εδώ](#)

ΒΗΜΑ 1: Θα ξεκινήσουμε την εισαγωγή των βιβλιοθηκών που θα χρησιμοποιήσουμε.

```
import sklearn.datasets as dsets
```

```
import seaborn as sn
```

```
import matplotlib.pyplot as plt
```

```
import numpy as np
```

```
import pandas as pd
```

ΒΗΜΑ 2: Φορτώνουμε το σύνολο δεδομένων και χρησιμοποιούμε μια παράμετρο της συνάρτησης που μας επιτρέπει να λάβουμε τα δεδομένα ως Pandas DataFrame. Μπορούν επίσης να φορτωθούν ως πίνακας NumPy, αλλά το pandas θα μας επιτρέψει να χρησιμοποιήσουμε απευθείας όλες τις συναρτήσεις του για την εξερεύνηση και τον μετασχηματισμό δεδομένων.

Το αποτέλεσμα της συνάρτησης είναι ένα λεξικό που έχει στο κλειδί 'frame' το πλαίσιο δεδομένων pandas με όλα τα δεδομένα.

```
hdata= dsets.fetch_california_housing(as_frame=True)
```

Μπορούμε να λάβουμε τις διαστάσεις του πλαισίου δεδομένων

```
hdata['frame'].shape
```

```
(20640, 9)
```

Θα αναθέσουμε το DataFrame στο λεξικό σε μια μεταβλητή για να συντομεύσουμε το όνομα.

Η μέθοδος head επιτρέπει την απόκτηση των πρώτων στοιχείων του DataFrame. Μπορούμε να δούμε τα χαρακτηριστικά που έχουν χρησιμοποιηθεί.

```
hframe = hdata['frame']
```

```
hframe.head()
```

	MedInc	House Age	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude	MedHouseVal
0	8.3252	41.0	6.984127	1.023810	322.0	2.555556	37.88	-122.23	4.526
1	8.3014	21.0	6.238137	0.971880	2401.0	2.109842	37.86	-122.22	3.585
2	7.2574	52.0	8.288136	1.073446	496.0	2.802260	37.85	-122.24	3.521
3	5.6431	52.0	5.817352	1.073059	558.0	2.547945	37.85	-122.25	3.413
4	3.8462	52.0	6.281853	1.081081	565.0	2.181467	37.85	-122.25	3.422

Αυτό το σύνολο δεδομένων προέκυψε από την απογραφή των ΗΠΑ του 1990, χρησιμοποιώντας μία γραμμή ανά ομάδα τετραγώνων απογραφής. Μια ομάδα οικοδομικών τετραγώνων είναι η μικρότερη γεωγραφική μονάδα για την οποία το Γραφείο Απογραφής των ΗΠΑ δημοσιεύει δειγματοληπτικά στοιχεία (μια ομάδα οικοδομικών τετραγώνων έχει συνήθως πληθυσμό 600 έως 3.000 ατόμων).

Υπάρχουν οκτώ χαρακτηριστικά για κάθε δείγμα και ένα επιπλέον χαρακτηριστικό που είναι η έξοδος για την πρόβλεψη:

- MedInc - διάμεσο εισόδημα στο μπλοκ
- HouseAge - διάμεση ηλικία κατοικίας στο τετράγωνο
- AveRooms - μέσος αριθμός δωματίων
- AveBedrms - μέσος αριθμός υπνοδωματίων
- Population - πληθυσμός οικοδομικού τετραγώνου
- AveOccup - μέση πληρότητα κατοικίας
- Latitude house - γεωγραφικό πλάτος οικοδομικού τετραγώνου
- Longitude house - γεωγραφικό μήκος οικοδομικού τετραγώνου

Η μεταβλητή-στόχος είναι η μέση τιμή κατοικίας για τις συνοικίες της Καλιφόρνια MedHouseVals.

ΒΗΜΑ 3: Μπορούμε να ξεκινήσουμε αναζητώντας τα στατιστικά στοιχεία των δεδομένων χρησιμοποιώντας τη συνάρτηση DataFrame describe

```
hframe.describe(include='all')
```

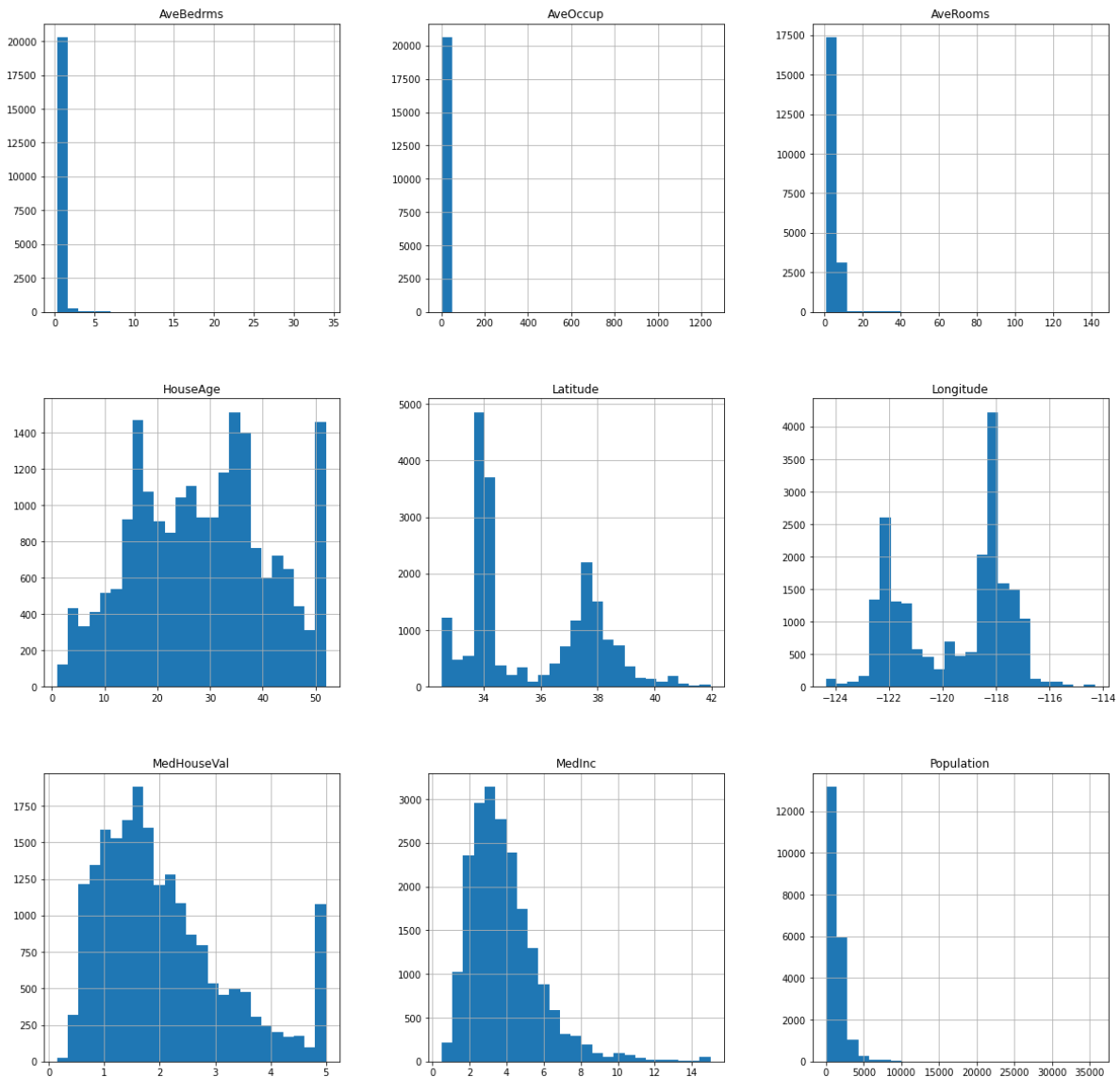
	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude	MedHouseVal
count	20640	20640	20640	20640	20640	20640	20640	20640	20640
mean	3.870671	28.639486	5.429000	1.096675	1425.476744	3.070655	35.631861	-119.569704	2.068558
std	1.899822	12.585558	2.474173	0.473911	1132.462122	10.386050	2.135952	2.003532	1.153956
min	0.499900	1	0.846154	0.333333	3	0.692308	32.540	-124.350	0.149990
25%	2.563400	18	4.440716	1.006079	787	2.429741	33.930	-121.80	1.1960
50%	3.534800	29	5.229129	1.048780	1166	2.818116	34.260	-118.490	1.7970
75%	4.743250	37	6.052381	1.099526	1725	3.282261	37.710	-118.010	2.647250
max	15.000100	52	141.909091	34.066667	35682	1243.333333	41.950	-114.310	5.000010

ΒΗΜΑ 4: Μπορούμε επίσης να απεικονίσουμε τα ιστογράμματα των μεταβλητών για να ελέγξουμε την κατανομή τους

```
hframe.hist(bins=25, figsize=(20,20))
```

```
plt.show()
```

Μπορούμε να παρατηρήσουμε ότι υπάρχουν κάποιες λοξές κατανομές με λίγες μόνο πολύ μεγάλες τιμές, μπορούμε να δημιουργήσουμε επιπλέον στήλες μετασχηματίζοντας τα αρχικά



δεδομένα εφαρμόζοντας ένα λογαριθμικό μετασχηματισμό στις πιο λοξές

```
hframe['logAveBedrms'] = np.log(hframe['AveBedrms'])
```

```
hframe['logAveOccup'] = np.log(hframe['AveOccup'])
```

```
hframe['logAveRooms'] = np.log(hframe['AveRooms'])
```

Μπορούμε να σχεδιάσουμε τις νέες μεταβλητές για να δούμε αν φαίνονται καλύτερα

```
hframe.loc[:,['logAveBedrms','logAveOccup', 'logAveRooms' ]].hist(bins=25, figsize=(10,10))
```

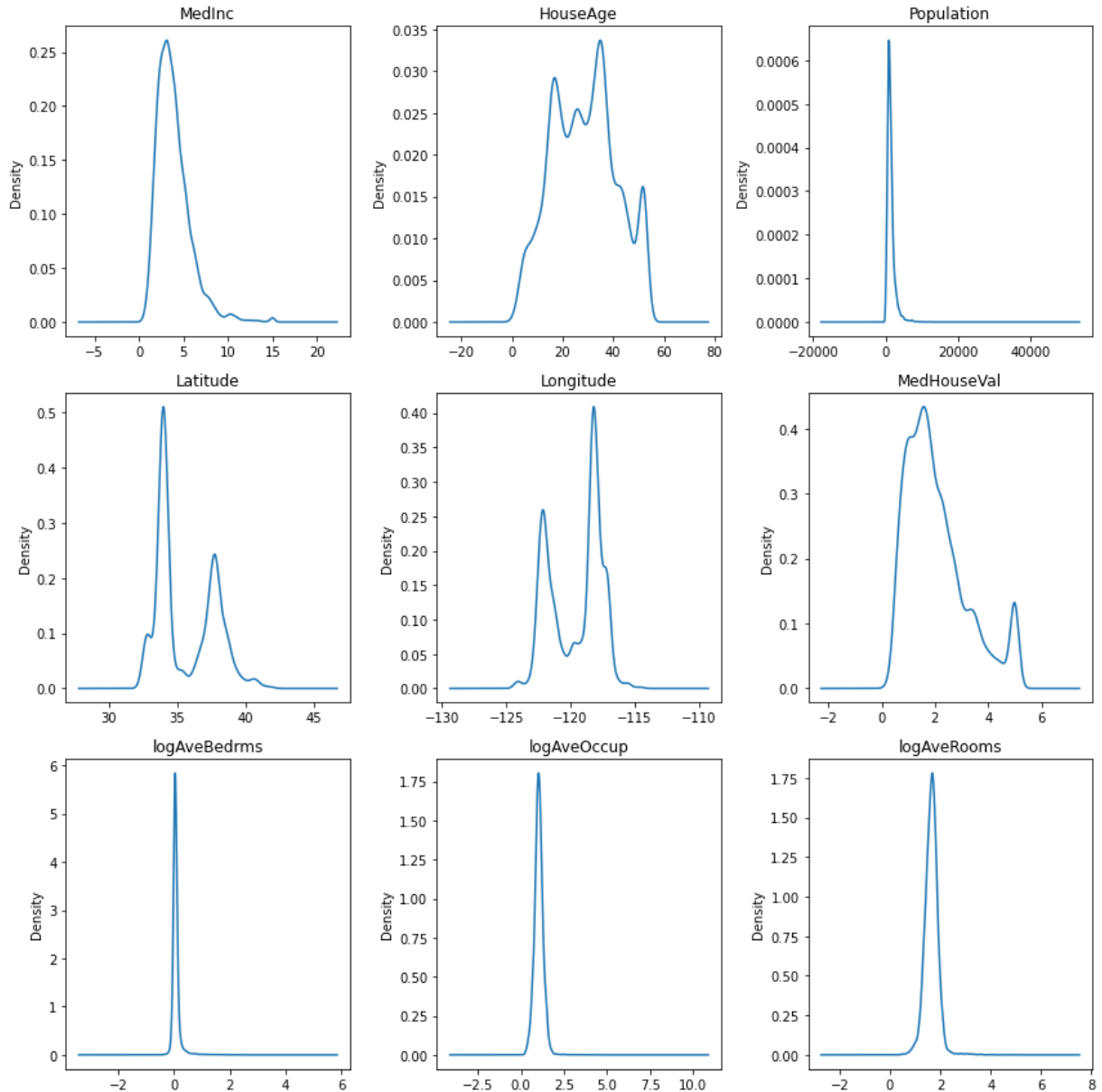
```
plt.show()
```

ΒΗΜΑ 5: Μπορούμε επίσης να σχεδιάσουμε ένα διάγραμμα πυκνότητας πυρήνα για να προσεγγίσουμε καλύτερα την κατανομή πιθανότητας κάθε μεταβλητής

```
ig = plt.figure(figsize=(12,12))
```



```
for i,col in enumerate(['MedInc', 'HouseAge', 'Population', 'Latitude', 'Longitude', 'MedHouseVal',
                        'logAveBedrms', 'logAveOccup',
                        'logAveRooms']):
```



```
    ax = fig.add_subplot(3, 3, i+1)
    hframe[col].plot.kde()
    plt.title(col)
    fig.tight_layout()

plt.show()
```

Μπορούμε να παρατηρήσουμε ότι υπάρχει μια κορυφή στο τέλος της κατανομής πιθανοτήτων της μεταβλητής HouseAge, πρόκειται για σπίτια ηλικίας άνω των 50 ετών που μπορεί να

συμπεριφέρονται διαφορετικά από τα υπόλοιπα σπίτια, θα μπορούσαμε να επιλέξουμε μόνο τα δεδομένα με ηλικία μικρότερη από αυτή.

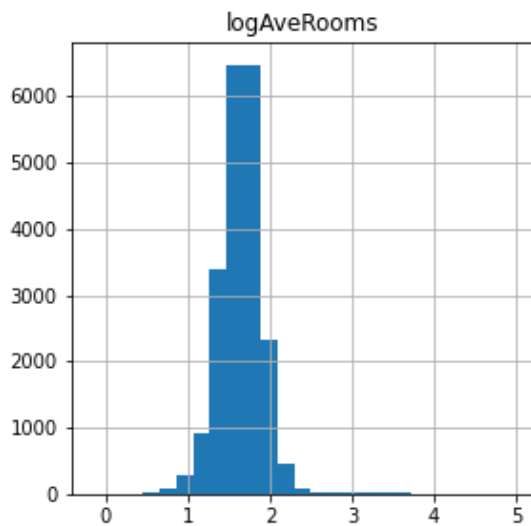
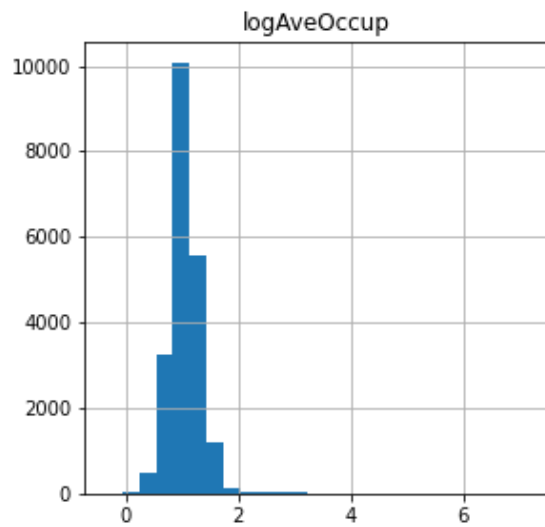
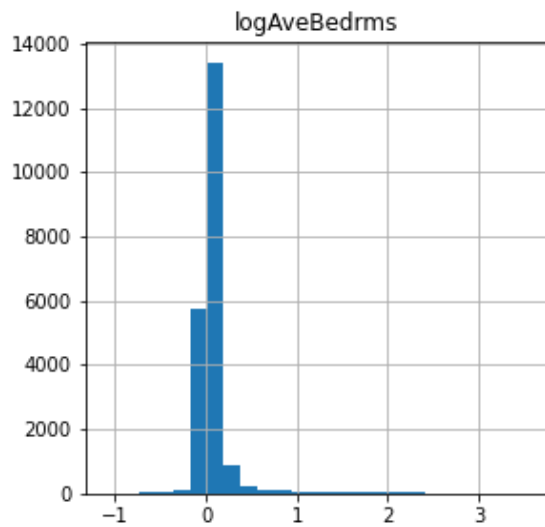
```
hframeL50 = hframe[hframe['HouseAge']<50]
```

```
hframeL50.shape
```

```
(19183, 12)
```

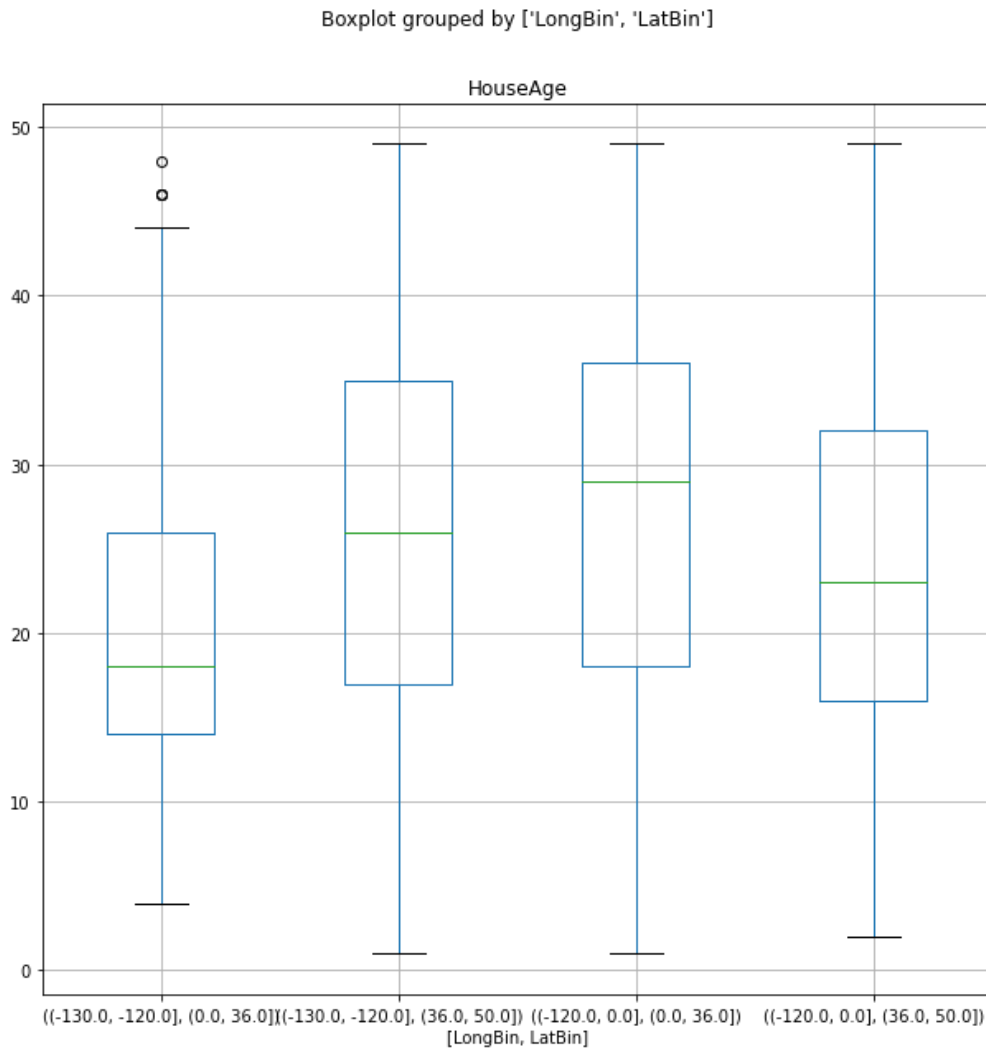
```
hframeL50['HouseAge'].plot.kde()
```

```
plt.show()
```



Translated with DeepL.com (free version)

Έτσι εξαλείφονται περίπου 1500 παραδείγματα και εξαλείφεται αυτή η κορυφή.



Μπορούμε επίσης να παρατηρήσουμε ότι το γεωγραφικό μήκος και το γεωγραφικό πλάτος παρουσιάζουν δύο ξεχωριστές κορυφές, μπορούμε να διακριτοποιήσουμε αυτές τις μεταβλητές και να τις μετατρέψουμε σε κατηγορικές μεταβλητές

```
bins = pd.IntervalIndex.from_tuples([(-130.0, -120.0), (-120.0, 0)])
```

```
hframeL50['LongBin'] = pd.cut(hframeL50['Longitude'], bins)
```

```
bins = pd.IntervalIndex.from_tuples([(0.0, 36.0), (36.0, 50)])
```

```
hframeL50['LatBin'] = pd.cut(hframeL50['Latitude'], bins)
```

Translated with DeepL.com (free version)

ΒΗΜΑ 6: Τώρα μπορούμε να χρησιμοποιήσουμε τις διακριτοποιημένες τιμές για να παρατηρήσουμε την επιρροή σε άλλες μεταβλητές

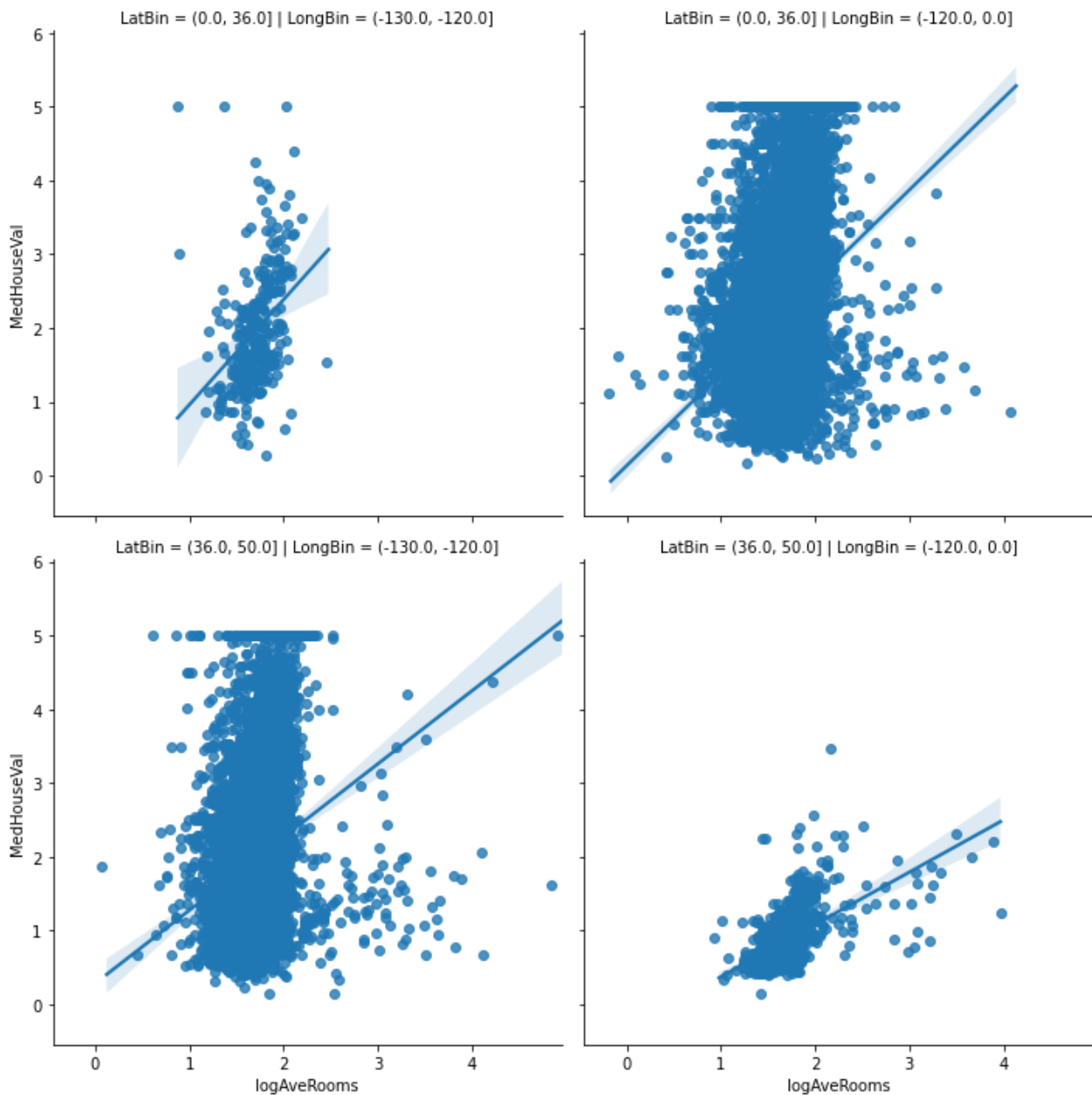
```
hframeL50.boxplot(column='HouseAge',by=['LongBin','LatBin'],figsize=(10,10))
```

```
plt.show()
```

Μπορούμε επίσης να παρατηρήσουμε τις διασταυρούμενες σχέσεις μεταξύ των μεταβλητών των παραδειγμάτων και της μεταβλητής εξόδου from pandas.plotting import scatter_matrix

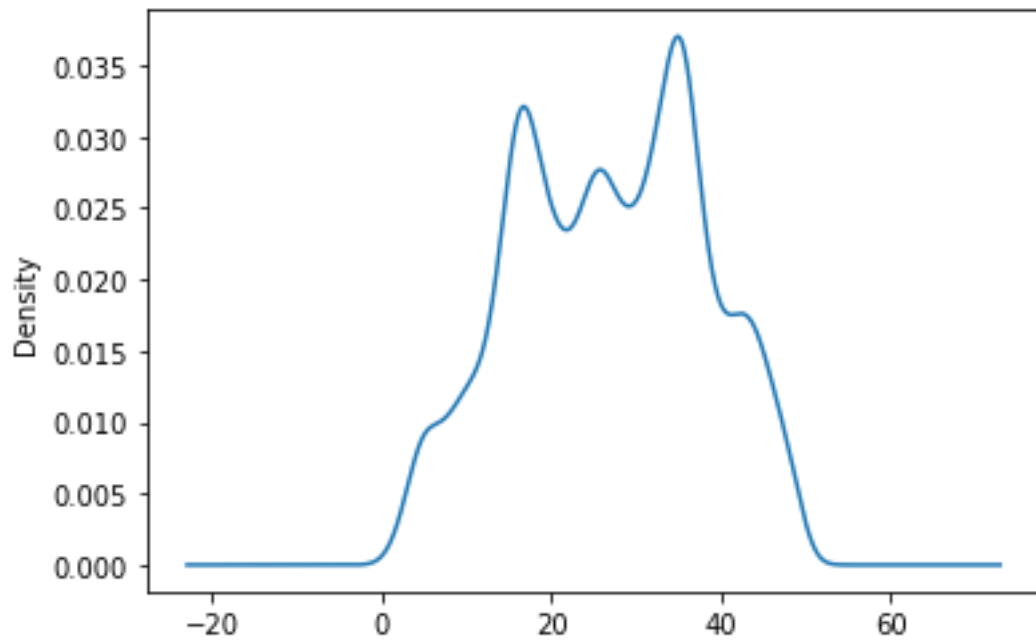
```
scatter_matrix(hframeL50.loc[:,['HouseAge','logAveBedrms','MedHouseVal']],  
alpha=0.2, figsize=(12, 12), diagonal='kde', marker='.')  
plt.show()
```

Μπορούμε να χρησιμοποιήσουμε τις διακριτοποιημένες μεταβλητές για να αναλύσουμε τη σχέση με τη μεταβλητή εξόδου



```
sn.lmplot(x="logAveRooms", y="MedHouseVal", row="LatBin", col="LongBin", data=hframeL50)  
plt.show()
```

Με τις πληροφορίες που αποκτήσαμε οπτικοποιώντας τα δεδομένα, μπορούμε να προχωρήσουμε στη δοκιμή κάποιων πιο σύνθετων μετασχηματισμών και αλγορίθμων μηχανικής μάθησης.



Αναφορές

- Python machine learning Wei-Meng, Lee, Wiley, 2019. ISBN: 978-1-119- 54563-7 (chapters 1 to 5)
- Introduction to Machine Learning with Python: A Guide for Data Scientists, Andreas C. Muller, O'Reilly media, 2016. ISBN: 9781449369415 (chapter 1)
- Python for data analysis, Wes MacKynney, O'Reilly media, 2013, ISBN: 978-1-449-31979-3

Μάθημα 2.3: Μετασχηματισμός και οπτικοποίηση δεδομένων

Θέματα προς συζήτηση

- Εισαγωγή
- Ανάλυση κύριων συνιστωσών
- Πολυδιάστατη κλιμάκωση
- Τοπικά γραμμική ενσωμάτωση
- Ενσωμάτωση t-στοχαστικού γείτονα

Σημειώσεις διάλεξης

Εισαγωγή

Πριν από την εφαρμογή ενός συγκεκριμένου αλγορίθμου μηχανικής μάθησης σε ένα σύνολο δεδομένων, μπορούν να πραγματοποιηθούν διάφορα βήματα προεπεξεργασίας. Ο πρώτος στόχος των βημάτων προεπεξεργασίας είναι η διασφάλιση της ποιότητας των δεδομένων με τη μείωση των θορυβωδών και άσχετων πληροφοριών που μπορεί να περιέχουν. Ο δεύτερος στόχος είναι η μείωση του μεγέθους του συνόλου δεδομένων, οπότε μειώνεται και το υπολογιστικό κόστος της εργασίας ανακάλυψης. Αυτή η διαδικασία μείωσης μπορεί επίσης να έχει ως σκοπό την οπτικοποίηση των δεδομένων για να βοηθήσει τη διαδικασία μάθησης.

Υπάρχουν δύο διαστάσεις που πρέπει να ληφθούν υπόψη κατά τη μείωση του μεγέθους ενός συνόλου δεδομένων. Η πρώτη είναι ο αριθμός των παραδειγμάτων. Το πρόβλημα αυτό μπορεί να αντιμετωπιστεί με τεχνικές δειγματοληψίας, όταν είναι σαφές ότι ένα μικρότερο υποσύνολο των δεδομένων περιέχει τις ίδιες πληροφορίες με το σύνολο των δεδομένων. Η μείωση της διαστατικότητας αντιμετωπίζει τον αριθμό των χαρακτηριστικών του συνόλου δεδομένων μετασχηματίζοντας τα δεδομένα από την αρχική τους αναπαράσταση σε μια αναπαράσταση με μειωμένο σύνολο χαρακτηριστικών. Οι τεχνικές αυτές προέρχονται από πολύ διαφορετικούς τομείς, αλλά ο κοινός στόχος είναι να ληφθεί ένα νέο σύνολο δεδομένων που διατηρεί, μέχρι ενός επιπέδου, την αρχική δομή των δεδομένων, ώστε η ανάλυσή του να καταλήγει στα ίδια ή ισοδύναμα πρότυπα που υπάρχουν στα αρχικά δεδομένα.

Αυτές οι τεχνικές μπορούν επίσης να χρησιμοποιηθούν για την εξαγωγή χαρακτηριστικών. Χαρακτηριστικά που συνοψίζουν τις σχέσεις μεταξύ διαφόρων μεταβλητών με γραμμικό ή μη γραμμικό τρόπο. Δεδομένου ότι θέλουμε μοντέλα που να χρησιμοποιούν τις λιγότερες δυνατές πληροφορίες, το να έχουμε διαθέσιμα αυτά τα παραγόμενα χαρακτηριστικά μπορεί να βοηθήσει στην εύρεση μοντέλων που χρειάζονται λιγότερα χαρακτηριστικά για να εξηγήσουν τα δεδομένα.

Ένα άλλο πρόβλημα που μειώνουν αυτές οι τεχνικές είναι αυτό που ονομάζεται κατάρα της διαστατικότητας. Ένας μεγάλος αριθμός μεθόδων στη μηχανική μάθηση βασίζεται στην

ομοιότητα μεταξύ των διανυσμάτων των χαρακτηριστικών που περιγράφουν τα παραδείγματα. Αυτή η ομοιότητα χάνει το νόημά της όσο περισσότερες διαστάσεις έχουν τα δεδομένα, επειδή για χώρους υψηλής διαστατικότητας όλα έχουν την τάση να έχουν την ίδια ομοιότητα. Επίσης, ο αριθμός των παραδειγμάτων για να είναι δυνατή η γενίκευση με εγγυήσεις πρέπει να αυξάνεται εκθετικά με τον αριθμό των διαστάσεων. Αυτό σημαίνει ότι η μείωση του αριθμού των χαρακτηριστικών που χρησιμοποιούνται στα δεδομένα μπορεί να βοηθήσει.

Υπάρχουν διάφοροι αλγόριθμοι για τη μείωση της διάστασης, σε αυτό το μάθημα θα επικεντρωθούμε σε τέσσερις, την Ανάλυση Κύριων Συνιστωσών (PCA), την Πολυδιάστατη Κλιμάκωση (MDS), την Τοπική Γραμμική Ενσωμάτωση (LLE) και την Ενσωμάτωση t-Stochastic Neighbor (t-SNE). Αυτοί οι αλγόριθμοι χρησιμοποιούνται ευρέως σε πρακτικές εφαρμογές και βρίσκονται στις περισσότερες βιβλιοθήκες μηχανικής μάθησης και στα περισσότερα πακέτα λογισμικού που χρησιμοποιούνται από τη βιομηχανία.

Ανάλυση κύριων συνιστωσών

Η Ανάλυση Κύριων Συνιστωσών (PCA) είναι μία από τις πιο δημοφιλείς γραμμικές μεθόδους μείωσης διαστάσεων. Είναι μη εποπτευόμενη, αυτό σημαίνει ότι δεν χρησιμοποιεί πληροφορίες εξόδου από το σύνολο δεδομένων για την εκτέλεση του μετασχηματισμού. Είναι γραμμική μέθοδος, επομένως τα νέα χαρακτηριστικά θα είναι γραμμικός συνδυασμός των αρχικών χαρακτηριστικών. Ο μετασχηματισμός που υπολογίζεται οδηγεί σε ένα σύνολο ορθογώνιων διαστάσεων που αντιπροσωπεύουν όλη τη διακύμανση του συνόλου δεδομένων. Είναι σύνηθες λίγες μόνο από αυτές τις διαστάσεις να κατέχουν το μεγαλύτερο μέρος της διακύμανσης, πράγμα που σημαίνει ότι με αυτό μόνο το υποσύνολο θα πρέπει να είναι αρκετό για να ανακαλύψουμε τα μοτίβα στα δεδομένα. Αυτό επιτρέπει επίσης την οπτικοποίηση της δομής των δεδομένων, εάν δύο ή τρεις διαστάσεις είναι αρκετές για την αναπαράστασή τους.

Η PCA ενδείκνυται καλύτερα για χαρακτηριστικά που ακολουθούν γκαουσιανή κατανομή. Το νέο σύνολο χαρακτηριστικών, που ονομάζονται κύριες συνιστώσες, είναι ασυσχέτιστα (ορθογώνια), έτσι ώστε το καθένα εξ αυτών να εξηγεί ένα μέρος της συνολικής διακύμανσης. Η μέθοδος μπορεί να περιγραφεί γεωμετρικά ως μια περιστροφή/κλιμάκωση των αξόνων των δεδομένων, αναζητώντας την προβολή όπου η διακύμανση των δεδομένων μπορεί να εξηγηθεί από τις λιγότερες μεταβλητές.

Ο αλγόριθμος για τον υπολογισμό του μετασχηματισμού PCA μπορεί να προκύψει με διάφορους τρόπους. Μπορεί να περιγραφεί ως ένα πρόβλημα βελτιστοποίησης που επιλύεται για κάθε συνιστώσα, όπου για κάθε επανάληψη μεγιστοποιείται η διακύμανση που εξηγείται από την τρέχουσα συνιστώσα. Η προβολή ενός συνόλου δεδομένων x σύμφωνα με ένα διάνυσμα κατεύθυνσης w μπορεί να υπολογιστεί ως εξής:

$$z = w'x$$

όπου w' είναι ο μετασχηματισμός του διανύσματος w . Για τον υπολογισμό της πρώτης διανυσματικής συνιστώσας w_1 θέλουμε την κατεύθυνση που κάνει πιο εμφανείς τις διαφορές μεταξύ των δεδομένων, την κατεύθυνση της μέγιστης διακύμανσης. Για να έχουμε μοναδική λύση για το διάνυσμα αυτό χρειαζόμαστε τον περιορισμό ότι η νόρμα του διανύσματος πρέπει να είναι 1. Αναζητούμε:

$$z_1 = w_1' x$$

Για ένα σύνολο δεδομένων με συνδιακύμανση $\text{Cov}(x) = \Sigma$, μπορούμε να ορίσουμε τη διακύμανση της προβολής που λαμβάνεται από το w_1 ως εξής:

$$\text{Var}(z_1) = w_1' \Sigma w_1$$

Θέλουμε να βρούμε το διάνυσμα που μεγιστοποιεί αυτή την τιμή με την προϋπόθεση $w_1' w_1 = 1$. Αυτό ισχύει αν w_1 είναι το ιδιοδιάνυσμα του Σ με τη μεγαλύτερη ιδιοτιμή. Το δεύτερο διάνυσμα w_2 πρέπει επίσης να έχει μοναδιαία νόρμα και να είναι ορθογώνιο με το w_1 , οπότε είναι ασυσχέτιστο. Αυτό σημαίνει ότι αυτό το διάνυσμα πρέπει να είναι το ιδιοδιάνυσμα του Σ με τη δεύτερη μεγαλύτερη τιμή. Κατόπιν αυτού μπορούμε να υπολογίσουμε όλα τα διανύσματα που αναπαράγουν τη διακύμανση των αρχικών δεδομένων και είναι όλα ορθογώνια μεταξύ τους. Μπορούμε να υπολογίσουμε τον μετασχηματισμό απλά υπολογίζοντας την εκτίμηση της συνδιακύμανσης του πίνακα από τα δεδομένα και στη συνέχεια εφαρμόζοντας ιδιοδιαστολή σε αυτόν τον πίνακα. Τα ιδιοδιανύσματα είναι το διάνυσμα προβολής, οι κύριες συνιστώσες.

Κάθε συνιστώσα εξηγεί ένα μέρος της διακύμανσης και κάθε μία εξηγεί ένα μικρότερο μέρος από την προηγούμενη. Μπορούμε να απεικονίσουμε τα δεδομένα εάν οι δύο ή τρεις πρώτες συνιστώσες εξηγούν ένα μεγάλο μέρος της διακύμανσης των δεδομένων. Εάν τα δεδομένα είναι σε μεγάλο βαθμό συσχετισμένα, θα χρειαστούμε μόνο μερικές από αυτές. Αυτό σημαίνει ότι δεδομένου ενός κατωφλίου μπορούμε να κρατήσουμε μόνο τις συνιστώσες που προσθέτουν σε αυτό το κατώφλι, απορρίπτοντας τις υπόλοιπες. Εναλλακτικά, το ποσοστό της διακύμανσης που εξηγείται από κάθε συνιστώσα μπορεί να απεικονιστεί σε διάγραμμα για να ληφθεί η απόφαση, το διάγραμμα αυτό ονομάζεται διάγραμμα scree. Μπορούμε να αποφασίσουμε τον αριθμό των συνιστωσών αναζητώντας ένα άλμα της διακύμανσης μεταξύ δύο διαδοχικών κύριων συνιστωσών σε αυτό το διάγραμμα.

Υπάρχουν κάποια πράγματα που πρέπει να έχουμε κατά νου χρησιμοποιώντας την PCA. Πρόκειται για έναν γραμμικό μετασχηματισμό, οπότε αν υπάρχουν πιο σύνθετες σχέσεις μεταξύ των χαρακτηριστικών, αυτός ο μετασχηματισμός δεν θα είναι σε θέση να τις ανιχνεύσει. Τα δεδομένα θα πρέπει να είναι κατά προτίμηση κατανομημένα με γκαουσιανή κατανομή. Η μέθοδος λειτουργεί αν δεν ισχύει αυτή η υπόθεση, αλλά όσο λιγότερο γκαουσιανά είναι τα δεδομένα τόσο λιγότερο αντιπροσωπευτικά είναι τα αποτελέσματα. Τα δεδομένα θα πρέπει να σταδιοποιηθούν πριν από την εφαρμογή της PCA, δηλαδή να αφαιρεθεί ο μέσος όρος και να διαιρεθεί με την τυπική απόκλιση για κάθε μεταβλητή. Αυτός ο μετασχηματισμός χρησιμοποιεί όλα τα χαρακτηριστικά για κάθε συνιστώσα, οπότε μπορεί να είναι δύσκολη η ερμηνεία τους. Τα βάρη που συνδυάζουν τα αρχικά χαρακτηριστικά μπορεί να περιλαμβάνουν θετικά και αρνητικά βάρη και τα τελευταία μπορεί να μην έχουν νόημα για ορισμένους τομείς, αν θέλουμε να ερμηνεύσουμε τις συνιστώσες στον αρχικό χώρο (π.χ. τομείς κειμένου).

Πολυδιάστατη κλιμάκωση

Με την πολυδιάστατη κλιμάκωση (MDS) επιτυγχάνεται ένας μετασχηματισμός που προβάλλει απευθείας από τον αρχικό αριθμό διαστάσεων σε έναν χώρο χαμηλότερων διαστάσεων, όπου το μέγεθος αυτού του χώρου είναι μια παράμετρος εισόδου. Αυτό σημαίνει ότι πρέπει να επιλέξουμε τον αριθμό των διαστάσεων που θα λάβουμε πριν από τον υπολογισμό του μετασχηματισμού. Αυτός ο μετασχηματισμός προσπαθεί να διατηρήσει τις αποστάσεις μεταξύ κάθε ζεύγους

παραδειγμάτων σε ένα σύνολο δεδομένων, λαμβάνοντας ένα νέο σύνολο διαστάσεων όπου οι συντεταγμένες που αποδίδονται στα παραδείγματα προσεγγίζουν τις αρχικές αποστάσεις

Υπάρχουν διάφορες παραλλαγές αυτού του αλγορίθμου ανάλογα με τον τύπο των αποστάσεων που χρησιμοποιούνται ή αν ο μετασχηματισμός είναι ή όχι γραμμικός. Ο κλασικός MDS προϋποθέτει ευκλείδειες αποστάσεις και μπορεί να υπολογιστεί με παρόμοιο τρόπο όπως η PCA. Υποθέτοντας ότι d_{ij} είναι η ευκλείδεια απόσταση μεταξύ των παραδειγμάτων x_i και x_j μπορούμε να ακολουθήσουμε τον αλγόριθμο:

1. Υπολογίστε τον πίνακα A ως τον πίνακα με στοιχεία $-1/2 * d_{ij}^2$.
2. Υπολογίστε τον πίνακα B ως πίνακα με στοιχεία $\{a_{ij} - a_{i.} - a_{.j} + a_{..}\}$ όπου $a_{i.}$ είναι ο μέσος όρος όλων των αποστάσεων όλων των a_{ij} σε όλο το j , και ούτω καθεξής
3. Βρείτε τις p μεγαλύτερες ιδιοτιμές του B με τα ιδιοδιανύσματά τους
4. Οι συντεταγμένες των παραδειγμάτων στο μετασχηματισμένο χώρο είναι οι γραμμές του πίνακα των ιδιοδιανυσμάτων κλιμακωμένες με την τετραγωνική ρίζα των ιδιοτιμών.

Το σφάλμα του μετασχηματισμού μπορεί να υπολογιστεί ως η διαφορά μεταξύ των αρχικών αποστάσεων d_{ij} και των νέων αποστάσεων $d^{\wedge}_{ij}$ χρησιμοποιώντας τη συνάρτηση τάσης:

$$stress = \frac{\sqrt{\sum (d_{ij} - d^{\wedge}_{ij})^2}}{\sum d_{ij}^2}$$

Η τιμή του p πρέπει να επιλέγεται έτσι ώστε το μέτρο αυτό να είναι χαμηλό. Λογικά θα αυξάνεται καθώς μειώνουμε τον αριθμό των διαστάσεων, αλλά όπως και με την PCA μπορούμε να εντοπίσουμε το σημείο κατά το οποίο η τιμή αυτή έχει μεγάλη αύξηση, υποδεικνύοντας ότι έχουμε μειώσει πάρα πολλές διαστάσεις. Τα δεδομένα θα πρέπει να τυποποιηθούν όπως και στην PCA.

Άλλες παραλλαγές αυτής της μεθόδου είναι η μετρική πολυδιάστατη κλιμάκωση που γενικεύεται σε άλλες αποστάσεις και η μη μετρική πολυδιάστατη κλιμάκωση που χρησιμοποιεί τις αποστάσεις ως κατάταξη και δεν υποθέτει γραμμική σχέση μεταξύ των χαρακτηριστικών εισόδου.

Ένα μειονέκτημα αυτής της μεθόδου είναι ότι δεν επεκτείνεται σε μεγάλα σύνολα δεδομένων, επειδή χρησιμοποιεί τον πίνακα αποστάσεων που αυξάνει το μέγεθός του τετραγωνικά με τον αριθμό των παραδειγμάτων.

Ένα από τα κύρια πλεονεκτήματα αυτής της μεθόδου είναι ότι λειτουργεί απευθείας με τις αποστάσεις, οπότε τα αρχικά δεδομένα δεν χρειάζεται να έχουν ένα σύνολο συντεταγμένων. Αυτό σημαίνει ότι μπορούμε να μετασχηματίσουμε οποιοδήποτε είδος δεδομένων όπου μπορεί να υπολογιστεί μια συνάρτηση απόστασης μεταξύ παραδειγμάτων για να λάβουμε μια αναπαράσταση που μπορεί να οπτικοποιηθεί.

Τοπική γραμμική ενσωμάτωση

Η τοπική γραμμική ενσωμάτωση εκτελεί έναν μη γραμμικό μετασχηματισμό που ανακτά μια συνολική δομή διατηρώντας την τοπική δομή των δεδομένων. Υποθέτει ότι κάθε παράδειγμα

μπορεί να ανακατασκευαστεί από ένα γραμμικό συνδυασμό των γειτονικών του. Η μέθοδος υπολογίζει για κάθε παράδειγμα ένα σύνολο βαρών ανακατασκευής από τους k-κοντινότερους γείτονές του ή τους γείτονες που είναι πιο κοντά από ένα κατώφλι. Από αυτά τα βάρη υπολογίζεται για κάθε παράδειγμα ένα νέο σύνολο συντεταγμένων σε ένα χώρο χαμηλότερης διάστασης, διατηρώντας τα υπολογισμένα βάρη ανακατασκευής.

Ο αλγόριθμος αυτής της μεθόδου έχει ως εξής:

$$\min_{w_{ij}} \left\| x_i - \sum_{k \in N(i)} w_{ik} x_k \right\|^2 \text{ subject to } \sum_{j \in N(i)} w_{ij} = 1$$

- Υπολογισμός για κάθε παράδειγμα του καταλόγου των k πλησιέστερων γειτόνων στον αρχικό χώρο N(i)
- Προσέγγιση κάθε παραδείγματος με γραμμικό συνδυασμό των γειτόνων του ελαχιστοποιώντας:
- Διατηρώντας τα βάρη αμετάβλητα, βρείτε ένα σύνολο συντεταγμένων σε ένα χώρο χαμηλότερης διάστασης που ελαχιστοποιεί τη συνάρτηση

$$\sum_{i=1}^n \left\| y_i - \sum_{k \in N(i)} w_{ik} y_k \right\|^2$$

Το πρόβλημα αυτό μπορεί να επιλυθεί με ιδιοδιαστολή, όπου οι συντεταγμένες των νέων σημείων αντιστοιχούν στις γραμμές των ιδιοδιανυσμάτων των χαμηλότερων ιδιοτιμών. Ο αριθμός των διαστάσεων που μπορούν να προκύψουν με τη μέθοδο αυτή για ένα σύνολο δεδομένων πρέπει να είναι μικρότερος ή ίσος από τον αριθμό των χρησιμοποιούμενων γειτόνων μείον ένα. Στην πράξη για να έχουμε έναν καλό μετασχηματισμό απαιτείται κάποια μεγαλύτερη διάσταση από την ελάχιστη.

t-στοχαστική ενσωμάτωση γείτονα

Πρόκειται για έναν αλγόριθμο που έχει κερδίσει δημοτικότητα πρόσφατα. Είναι ένας μη γραμμικός μετασχηματισμός που χρησιμοποιείται ως μέθοδος απεικόνισης δεδομένων λόγω της ικανότητάς του να διαχωρίζει πολύπλοκα μοτίβα σε χαμηλές διαστάσεις. Υποθέτει ότι οι αποστάσεις μεταξύ των παραδειγμάτων μπορούν να εκφραστούν ως κατανομή πιθανότητας, συνήθως γκαουσιανή. Ο στόχος είναι να βρεθεί ένα νέο σύνολο συντεταγμένων που διατηρεί την κατανομή των αποστάσεων σε ένα νέο χώρο χαμηλής διάστασης, συνήθως 2D. Η ομοιότητα μεταξύ των κατανομών στον αρχικό και στο νέο χώρο συγκρίνεται με τη χρήση μιας συνάρτησης της θεωρίας της πληροφορίας που μετρά πόσο κοντά είναι δύο κατανομές πιθανοτήτων, τη συνάρτηση απόκλισης Kullback-Leiber. Ο βασικός αλγόριθμος για αυτόν τον μετασχηματισμό έχει τα ακόλουθα βήματα:

1. Για κάθε παράδειγμα του συνόλου δεδομένων, η απόσταση από τα υπόλοιπα παραδείγματα υπολογίζεται και κανονικοποιείται έτσι ώστε όλες οι αποστάσεις για το παράδειγμα να έχουν άθροισμα ένα
2. Υπολογίζεται ένας πίνακας ομοιότητας που αντιπροσωπεύει την κατανομή των παραδειγμάτων
3. Τα παραδείγματα προβάλλονται σε έναν χώρο με τον αριθμό διαστάσεων-στόχο, οι αποστάσεις τους σε αυτόν τον νέο χώρο υπολογίζονται και κανονικοποιούνται όπως προηγουμένως, ώστε να προκύψει ένας νέος πίνακας ομοιότητας.
4. Υπολογίζεται η απόκλιση Kullback-Leiber μεταξύ αυτών των δύο πινάκων.
5. Οι συντεταγμένες των παραδειγμάτων στο νέο χώρο μετακινούνται έτσι ώστε να βελτιωθεί η απόκλιση
6. Εάν υπάρχει βελτίωση, η διαδικασία επαναλαμβάνεται από το βήμα 4

Το κύριο μειονέκτημα αυτού του αλγορίθμου είναι ότι το αποτέλεσμα εξαρτάται από την αρχική προβολή και ορισμένες πρόσθετες παράμετροι πρέπει να ρυθμιστούν πειραματικά. Ο αλγόριθμος μπορεί να αρχικοποιηθεί με τα αποτελέσματα της PCA για να γίνει το αποτέλεσμα πιο κοντά στις πραγματικές θέσεις των δεδομένων στον πραγματικό χώρο. Ως περιορισμός, δεν λαμβάνουμε έναν πίνακα προβολής, παρά μόνο τις συντεταγμένες των δεδομένων, οπότε δεν μπορούμε να εφαρμόσουμε τον μετασχηματισμό σε νέα δεδομένα. Επίσης, ο υπολογισμός του είναι δαπανηρός, για να γίνει αποδοτικός μπορούν να χρησιμοποιηθούν κάποιες προσεγγίσεις που περιορίζουν τις προβολές σε 2D ή 3D.

Επιλέγοντας έναν αλγόριθμο

Αυτοί οι τέσσερις αλγόριθμοι επιλύουν το ίδιο πρόβλημα, αλλά κάνουν διαφορετικές υποθέσεις για τα δεδομένα. Η ανάλυση κύριων συνιστωσών είναι η πιο δημοφιλής μέθοδος, δεδομένου ότι δεν είναι υπολογιστικά δαπανηρή, αλλά υποθέτει ότι τα χαρακτηριστικά είναι κατανομημένα με γκαουσιανή κατανομή και οι μεταξύ τους εξαρτήσεις είναι γραμμικές. Λειτουργεί ακόμη και όταν οι εν λόγω υποθέσεις δεν είναι απολύτως αληθείς, αλλά τα αποτελέσματά δεν θα είναι βέλτιστα. Η πολυδιάστατη κλιμάκωση δεν κάνει καμία υπόθεση σχετικά με την κατανομή των δεδομένων, αλλά υποθέτει επίσης ότι οι εξαρτήσεις μεταξύ των χαρακτηριστικών είναι γραμμικές. Η τοπική γραμμική ενσωμάτωση και η t-στοχαστική ενσωμάτωση γειτόνων θα είναι μια καλή επιλογή για δεδομένα με μη γραμμικές εξαρτήσεις. Προφανώς είναι αδύνατο να γνωρίζουμε τι είδους εξαρτήσεις έχουν τα χαρακτηριστικά ενός προβλήματος χωρίς να αναλύσουμε και να δοκιμάσουμε τα αποτελέσματα, οπότε η διαδικασία είναι συνήθως να υποθέσουμε γραμμικότητα και να δοκιμάσουμε την απόδοση των μετασχηματισμένων δεδομένων. Δεδομένου ότι αυτό είναι το βήμα για την απόκτηση ενός μοντέλου για ένα πρόβλημα, θα πρέπει να αναθεωρηθεί αφού το μοντέλο προκύψει με αλγορίθμους μάθησης και μετρηθεί η απόδοση.

Περυπτώσεις χρήσης

ΤΙΜΕΣ ΣΠΙΤΙΩΝ ΣΤΗΝ ΚΑΛΙΦΟΡΝΙΑ

Σε αυτή την περίπτωση χρήσης, θα συνεχίσουμε την εξερεύνηση του συνόλου δεδομένων για τις τιμές των κατοικιών στην Καλιφόρνια. Θα χρησιμοποιήσουμε τις συναρτήσεις scikit-learn για τη

μείωση της διαστατικότητας που υλοποιούν τις μεθόδους που εξηγούνται σε αυτό το μάθημα. Κατά την εφαρμογή αλγορίθμων μείωσης διαστάσεων και απεικόνισης πρέπει να έχουμε επίγνωση του στόχου που έχουμε με τη χρήση τους. Αν θέλουμε να οπτικοποιήσουμε τα δεδομένα για να ελέγξουμε αν υπάρχουν μοτίβα στα δεδομένα θα πρέπει να σχεδιάσουμε τα αποτελέσματα με διαφορετικές επιλογές και να πάρουμε αποφάσεις από την οπτικοποίηση. Αυτό είναι πολύ υποκειμενικό και απαιτεί συγκεκριμένες γνώσεις σχετικά με τον τομέα του προβλήματος. Όταν ο στόχος είναι η μείωση της διαστατικότητας των δεδομένων, ο μόνος τρόπος ελέγχου της απόφασης είναι η δημιουργία ενός μοντέλου με τα μετασχηματισμένα δεδομένα και η αντιμετώπιση της διαδικασίας μείωσης της διαστατικότητας ως πρόσθετης υπερπαραμέτρου του αλγορίθμου μάθησης.

ΒΗΜΑ 1: Θα ξεκινήσουμε φορτώνοντας αυτά τα δεδομένα, αλλά σε αυτή την περίπτωση, θα το κάνουμε ως πίνακες numpy που είναι η αναπαράσταση που χρησιμοποιεί εσωτερικά το scikit-learn. Θα εισάγουμε πρώτα τις βιβλιοθήκες και τις μεθόδους που θα χρειαστούμε:

```
import sklearn.datasets as datasets
import matplotlib.pyplot as plt
import numpy as np
from sklearn.decomposition import PCA
from sklearn.manifold import LocallyLinearEmbedding, TSNE, MDS
```

ΒΗΜΑ 2: Φορτώνουμε τα δεδομένα, σε αυτή την περίπτωση, έχουμε επίσης ένα λεξικό όπου το κλειδί 'data' έχει το σύνολο δεδομένων:

```
hdata= datasets.fetch_california_housing()
μπορούμε να εξετάσουμε τις διαστάσεις του πίνακα δεδομένων:
hdata['data'].shape
(20640, 8)
```

Έτσι, έχουμε τα ίδια δεδομένα με πριν, σε αυτή την περίπτωση, δεν θα χρειαστούμε την έξοδο για κάθε παράδειγμα, δεδομένου ότι όλες οι μέθοδοι που θα χρησιμοποιήσουμε είναι μη επιβλεπόμενες.

Μπορούμε να αναθέσουμε τα δεδομένα σε μια νέα μεταβλητή, ώστε το όνομά της να είναι συντομότερο:

```
hmatrix = hdata['data']
```

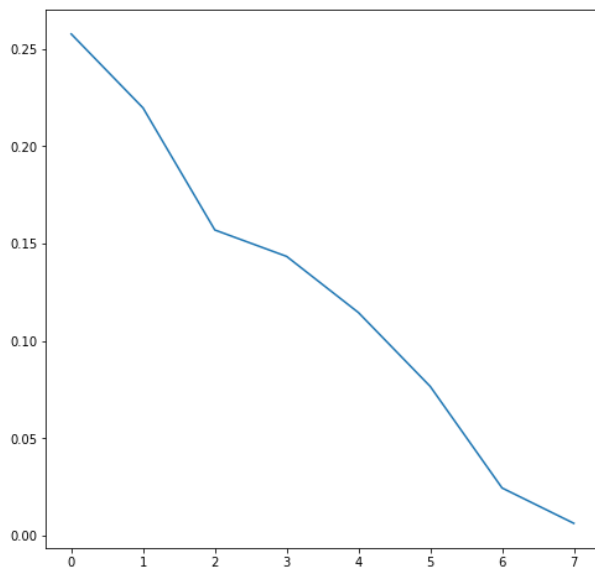
και πραγματοποιούμε τον ίδιο μετασχηματισμό στις μεταβλητές 'AveBedrms', 'AveOccur' και 'AveRooms' όπως κάναμε προηγουμένως για να λάβουμε δεδομένα που έχουν μια πιο γκαουσιανή συμπεριφορά, δεδομένου ότι πρόκειται να εφαρμόσουμε την PCA, η οποία υποθέτει ότι όλες οι μεταβλητές ακολουθούν γκαουσιανές κατανομές. Οι μεταβλητές αυτές αντιστοιχούν στις στήλες 2, 3 και 5.

```
hmatrix[:,2] = np.log(hmatrix[:,2])
hmatrix[:,3] = np.log(hmatrix[:,3])
hmatrix[:,5] = np.log(hmatrix[:,5])
```

ΒΗΜΑ 3: Θα ξεκινήσουμε εκτελώντας μια Ανάλυση Κύριων Συνιστωσών. Πριν το κάνουμε αυτό, μία από τις απαιτήσεις είναι ότι τα δεδομένα πρέπει να είναι τυποποιημένα, πράγμα που σημαίνει ότι τα δεδομένα πρέπει να κεντραριστούν αφαιρώντας τον μέσο όρο και η τυπική απόκλιση των στηλών πρέπει να είναι ένα, οπότε πρέπει να διαιρέσουμε με την εκτιμώμενη τυπική απόκλιση. Αυτό μπορεί να γίνει χρησιμοποιώντας τις συναρτήσεις κλιμάκωσης του scikit-learn, συγκεκριμένα τη συνάρτηση StandardScaler. Εισάγουμε τη συνάρτηση και μετασχηματίζουμε τα δεδομένα:

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
hmatrix = scaler.fit_transform(hmatrix)
Τώρα μπορούμε να εφαρμόσουμε την PCA στα δεδομένα:
pca = PCA()
pcamatrix = pca.fit_transform(hmatrix)
Αυτές είναι οι κύριες συνιστώσες που έχουν υπολογιστεί:
pca.components_
```

```
array([[ -0.06412208, -0.05343653, -0.24177695, -0.16986846, 0.2079705 , 0.22981868, -
0.64257474, 0.62988162],
[0.47962059, -0.39842007, 0.64197149, 0.36724259, 0.11806445, 0.01823021, -0.13135655,
0.18084503],
[-0.02733057, -0.43908822, -0.12435569, -0.24424168, 0.6794295 , 0.37417135, 0.25783847, -
0.25144964],
[-0.66895165, -0.25869378, -0.07818724, 0.65846182, 0.1258653 , -0.13376868, 0.03798847,
0.10352177],
```



```
[-0.11760047, 0.38732401, 0.20479682,
0.2531201, -0.12598973, 0.84200338,
0.06611194, -0.03041105],
[0.11103558, 0.64788025, 0.0831763 ,
0.20602098, 0.6670638, -0.26117326, -
0.06373337, -0.03621899],
[0.50900152, -0.06936251, -0.65648795,
0.48484576, -0.06438472, 0.10846745, -
0.10949145, -0.20531805],
[0.17991072, 0.06407173, -0.16733939,
0.03823327, 0.01869822, -0.00253558,
0.69390075, 0.67246442]]])
```

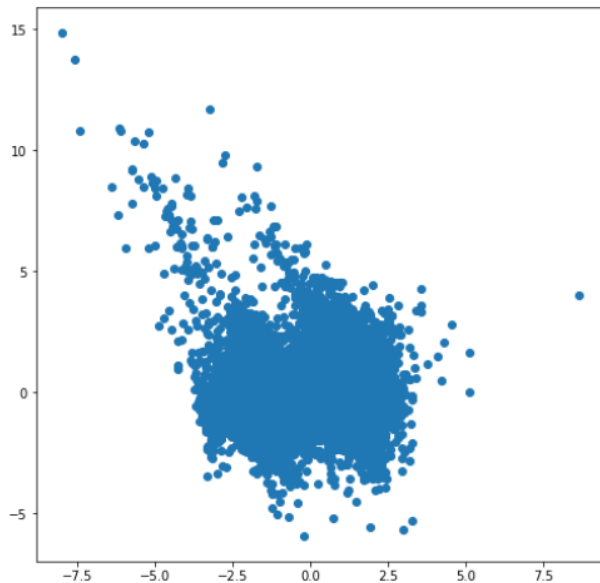
Αυτό είναι το ποσοστό της διακύμανσης που εξηγεί κάθε συνιστώσα:

```
pca.explained_variance_ratio_
array([0.25758417, 0.21963809, 0.15699766, 0.14343577, 0.11460901, 0.07669597, 0.02459801,
0.00644132])
```

Μπορούμε να κάνουμε ένα scree plot για να αποφασίσουμε πόσες μεταβλητές είναι αρκετές για την αναπαράσταση των δεδομένων:

```
fig = plt.figure(figsize=(8,8))
```

```
plt.plot(pca.explained_variance_ratio_)
```



```
plt.show()
```

Όπως βλέπουμε από τις 8 συνιστώσες, οι 6 εξηγούν το 97% της διακύμανσης, ενώ επιλέγοντας 5 έχουμε σχεδόν το 90%. Δυστυχώς, δεν ισχύει ότι μόνο δύο ή τρεις συνιστώσες εξηγούν όλη τη διακύμανση, οπότε η απεικόνιση μόνο των δύο για παράδειγμα θα μας δώσει μια μερική εικόνα της δομής τους, αλλά μπορούμε να σχεδιάσουμε το σύνολο δεδομένων με τις δύο πρώτες συνιστώσες για να δούμε αν υπάρχουν κάποια μοτίβα που φαίνονται ενδιαφέροντα.

```
fig = plt.figure(figsize=(8,8))
```

```
plt.scatter(pcamatrix[:, 0], pcamatrix[:, 1])
```

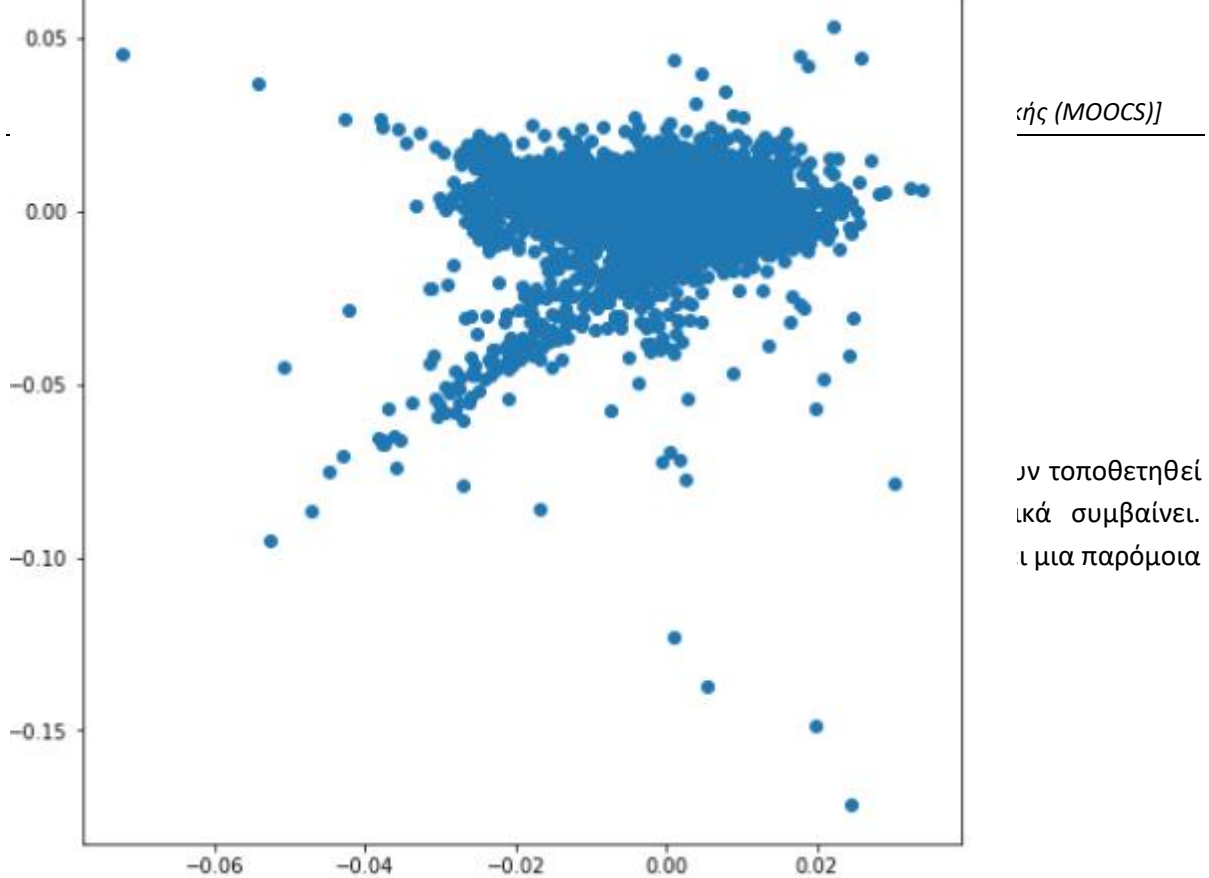
```
plt.show()
```

Μπορούμε να παρατηρήσουμε δύο επικαλυπτόμενες ομάδες. Αυτό μας δίνει μια ένδειξη ότι υπάρχουν κάποιες διακριτές ομάδες στα δεδομένα. Στην περίπτωση χρήσης του προηγούμενου μαθήματος διακριτοποιήσαμε τις μεταβλητές γεωγραφικό μήκος και γεωγραφικό πλάτος επειδή παρουσίαζαν δύο διακριτές κορυφές, αυτές οι δύο συνιστώσες δίνουν ένα πειστικό επιχείρημα για να το κάνουμε αυτό, επειδή πιθανώς αυτές οι δύο ομάδες αντιστοιχούν σε διαφορετικές συμπεριφορές ανάλογα με τις συντεταγμένες των σπιτιών.

Θα μπορούσαμε να προχωρήσουμε στη χρήση της πολυδιάστατης κλιμάκωσης, αλλά ένα από τα προβλήματά της είναι ότι δεν μπορεί να κλιμακωθεί σε μεγάλο αριθμό παραδειγμάτων λόγω του ότι υπολογίζεται με βάση τον πίνακα αποστάσεων των δεδομένων. Στην προκειμένη περίπτωση έχουμε περισσότερα από 20.000 παραδείγματα και η εφαρμογή αυτής της μεθόδου θα απαιτούσε μεγάλο όγκο μνήμης και υπολογιστικού χρόνου.

Θα προχωρήσουμε στην τοπική γραμμική ενσωμάτωση (Locally Linear Embedding) για να εκτελέσουμε έναν μη γραμμικό μετασχηματισμό και να δούμε αν η διαστατικότητα μπορεί να μειωθεί καλύτερα με αυτόν τον τρόπο. Θα πρέπει να διερευνήσουμε τον αριθμό των γειτόνων που χρησιμοποιούνται στον μετασχηματισμό για να δούμε τι επίδραση έχει στην αναπαράσταση έχοντας κατά νου ότι, όσο μεγαλύτερος είναι ο αριθμός των γειτόνων, τόσο λιγότερο μη γραμμικός είναι ο μετασχηματισμός:

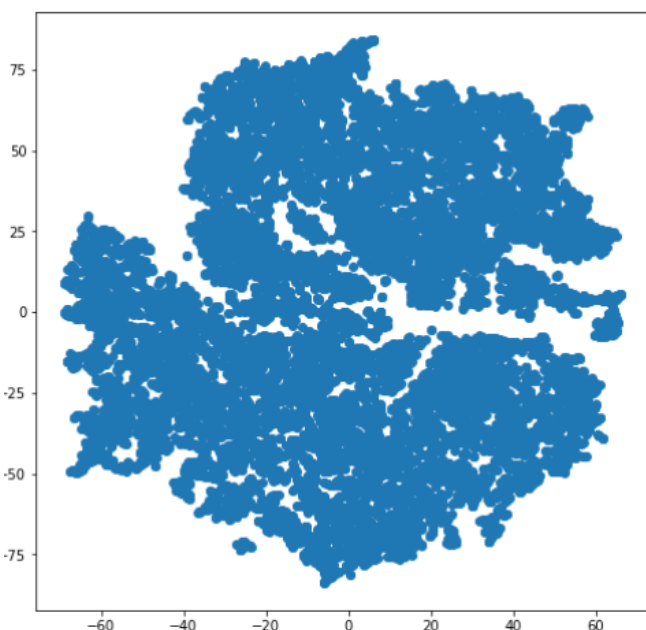
```
LLE = LocallyLinearEmbedding(n_components=2,n_neighbors=11)
```

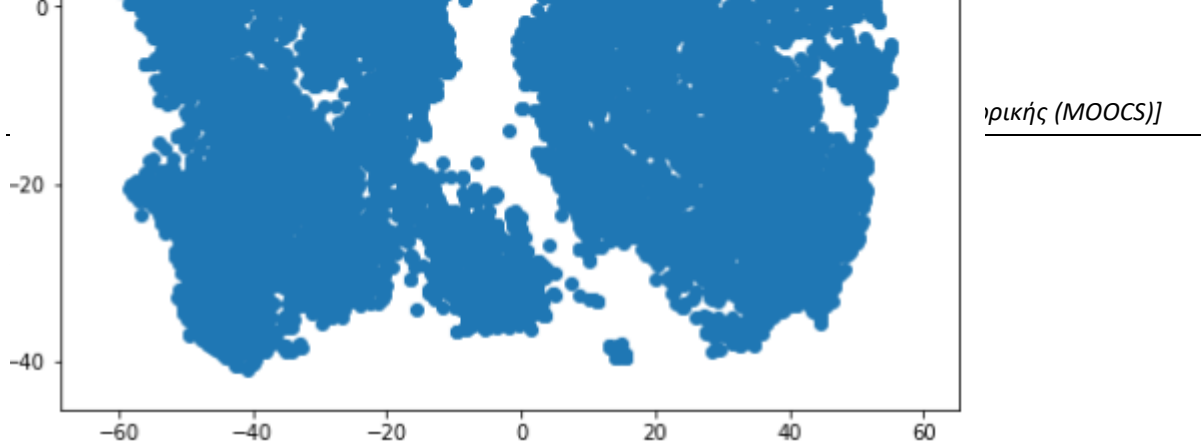


ΒΗΜΑ 4: Τέλος, εκτελούμε έναν μετασχηματισμό με τη χρήση της t-Stochastic Neighbour Embedding χρησιμοποιώντας τις προεπιλεγμένες παραμέτρους, αλλά χρησιμοποιώντας το αποτέλεσμα της PCA ως αρχικοποίηση. Δεδομένου ότι χρησιμοποιούμε τις προεπιλεγμένες παραμέτρους, ο μετασχηματισμός περιορίζεται σε 2D/3D. Μπορούμε να μετασχηματίσουμε σε 2 διαστάσεις όπως κάναμε και προηγουμένως:

```
tsne = TSNE(n_components=2, init='pca')
tsnmatrix = tsne.fit_transform(hmatrix)
fig = plt.figure(figsize=(8,8))
plt.scatter(tsnmatrix[:, 0], tsnmatrix[:, 1])
plt.show()
```

Μπορούμε επίσης να δούμε το διαχωρισμό των δύο ομάδων, αλλά υπάρχει μεγαλύτερη εσωτερική δομή που θα μπορούσε να σημαίνει πιο συγκεκριμένες ομάδες στα δεδομένα. Μπορούμε να τροποποιήσουμε τις προεπιλεγμένες παραμέτρους εξερευνώντας για παράδειγμα την παράμετρο perplexity που σχετίζεται με τον αριθμό των γειτόνων που χρησιμοποιούνται στο μετασχηματισμό και για μεγαλύτερα σύνολα δεδομένων χρειάζεται μεγαλύτερη τιμή.





Τώρα μπορούμε να δούμε έναν μεγαλύτερο διαχωρισμό και μια πιο σύνθετη δομή στα δεδομένα που θα μπορούσε να σημαίνει συγκεκριμένες συμπεριφορές.

Αναφορές

- **Introduction to machine learning** Alpaydin, E, The MIT Press, 2014. ISBN: 9780262028189 (chapter 6)
- **Pattern recognition and machine learning** Bishop, C.M, Springer, 2006. ISBN: 0387310738 (chapter 12)

Μάθημα 2.4: Γραμμικές μέθοδοι για επιβλεπόμενη μάθηση

Θέματα προς συζήτηση

- Εισαγωγή
- Γραμμικά μοντέλα παλινδρόμησης
- Δέντρα αποφάσεων και παλινδρόμησης
- k-κοντινότεροι γείτονες
- Naive bayes

Σημειώσεις διάλεξης

Εισαγωγή

Οι αλγόριθμοι με γραμμική εποπτεία μαθαίνουν μοντέλα που επιτρέπουν μόνο γραμμικούς συνδυασμούς των χαρακτηριστικών των παραδειγμάτων εισόδου. Αυτό δεν σημαίνει ότι δεν επιτρέπονται γραμμικοί μετασχηματισμοί των χαρακτηριστικών. Ο μόνος περιορισμός είναι ότι το μοντέλο θα τα συνδυάζει μόνο γραμμικά κατά τη διάρκεια της μάθησης. Η προσθήκη μη γραμμικών μετασχηματισμών των χαρακτηριστικών εισόδου αυξάνει την πολυπλοκότητα της γλώσσας που μπορούν να χρησιμοποιήσουν τα μοντέλα και μπορεί να τα κάνει πιο ευέλικτα προκειμένου να βρουν μια καλή γενίκευση των παραδειγμάτων εισόδου. Τα γραμμικά μοντέλα είναι υπολογιστικά φθηνότερα από τα μη γραμμικά μοντέλα και για ορισμένους αλγορίθμους μπορεί να ληφθεί μια ερμηνεία των αποτελεσμάτων, δίνοντας κάποιες πληροφορίες σχετικά με τον τρόπο με τον οποίο υπολογίζονται οι αποφάσεις στο μοντέλο.

Από όλους τους αλγορίθμους με γραμμική εποπτεία θα επικεντρωθούμε στα γραμμικά μοντέλα για την παλινδρόμηση, τα δέντρα αποφάσεων, τους k-κοντινότερους γείτονες και το naive bayes. Αυτοί είναι οι συνηθέως αλγόριθμοι που χρησιμοποιούνται σε πρακτικές εφαρμογές και μπορούν να βρεθούν στις περισσότερες βιβλιοθήκες μηχανικής μάθησης και πακέτα λογισμικού που χρησιμοποιούνται από τη βιομηχανία.

Γραμμικά μοντέλα για παλινδρόμηση

Ένα πρόβλημα παλινδρόμησης είναι μια εργασία με επίβλεψη, όπου το μοντέλο προς εκμάθηση πρέπει να συνδέει τις τιμές των χαρακτηριστικών των παραδειγμάτων εισόδου με μια συνεχή έξοδο. Υπάρχουν πολλοί τρόποι για την απόκτηση αυτών των μοντέλων, ο απλούστερος είναι να υποθέσουμε ότι η έξοδος είναι ένας σταθμισμένος γραμμικός συνδυασμός των μεταβλητών εισόδου. Ο στόχος είναι να μάθουμε τα βάρη που ελαχιστοποιούν το σφάλμα που διαπράττει αυτός ο συνδυασμός προβλέποντας τα παραδείγματα που έχουμε και τα μελλοντικά παραδείγματα που θα πρέπει να προβλέψουμε:

$$f(x; w) = w_0 + \sum_{i=1}^p w_i x_i$$

Όπου p είναι ο αριθμός των χαρακτηριστικών και ο όρος w_0 ονομάζεται ανεξάρτητος όρος και επιτρέπει στο μοντέλο να κινείται στο χώρο των εισόδων, ώστε να μην είναι κεντραρισμένο στην αρχή των συντεταγμένων. Αυτό το μοντέλο είναι γνωστό ως γραμμική παλινδρόμηση. Μπορούμε να γενικεύσουμε αυτό το μοντέλο επεκτείνοντας την αναπαράσταση που μπορεί να χρησιμοποιηθεί εκτελώντας έναν αριθμό σταθερών μη γραμμικών μετασχηματισμών στα δεδομένα εισόδου, ορίζοντας το μοντέλο ως εξής:

$$f(x; w) = \sum_{i=1}^p w_i \phi_i(x)$$

Όπου η $\Phi_i(x)$ εκτελεί έναν μη γραμμικό μετασχηματισμό των δεδομένων με $\Phi_0(x) = 1$, οπότε μπορούμε να εισάγουμε τον ανεξάρτητο όρο στην εξίσωση. Αυτός ο μετασχηματισμός μπορεί να αντιστοιχεί σε μια προεπεξεργασία των χαρακτηριστικών των δεδομένων σε ορισμένες εφαρμογές ή σε κάποιου είδους διαδικασία εξαγωγής χαρακτηριστικών, όπου εκτελούνται κάποιοι υπολογισμοί από τα χαρακτηριστικά εισόδου για να προκύψουν πιο χρήσιμα χαρακτηριστικά των δεδομένων για την εφαρμογή μας.

Ο συνήθης τρόπος εκμάθησης των παραμέτρων για τον γραμμικό συνδυασμό είναι η ελαχιστοποίηση των ελαχίστων τετραγώνων. Για να ελαχιστοποιηθεί το άθροισμα του τετραγώνου της διαφοράς μεταξύ της εξόδου του μοντέλου και των πραγματικών εξόδων των δεδομένων, δηλαδή το άθροισμα των τετραγώνων των υπολοίπων, ό,τι δεν εξηγείται από το μοντέλο:

$$SSR(w) = \sum_{i=1}^N (y_i - f(x_i; w))^2 = \sum_{i=1}^N \left(y_i - w_0 - \sum_{j=1}^p w_j x_{ij} \right)^2$$

Για να βρούμε τα βάρη που ελαχιστοποιούν αυτή τη συνάρτηση πρέπει να ξαναγράψουμε αυτή την εξίσωση σε συμβολισμό πινάκων ως εξής:

$$SSR(w) = (y - Xw)^T (y - Xw)$$

Πρόκειται για μια τετραγωνική συνάρτηση με $p+1$ παράμετρο που μπορούμε να λύσουμε εφαρμόζοντας παραγώγους για να βρούμε το ελάχιστο της συνάρτησης, λαμβάνοντας την εξίσωση που έχει μοναδική λύση:

$$\hat{w} = (X^T X)^{-1} X^T y$$

Έτσι, για οποιοδήποτε σύνολο παραδειγμάτων εισόδου, μπορούμε να βρούμε τις καλύτερες παραμέτρους με την έννοια ότι ελαχιστοποιούν τη διαφορά μεταξύ της προβλεπόμενης τιμής και

της πραγματικής τιμής. Ένα μειονέκτημα είναι ότι αν υπάρχουν νέα δεδομένα, τα βάρη πρέπει να υπολογιστούν ξανά.

Ένα πρόβλημα αυτού του μοντέλου είναι ότι, ανάλογα με τον όγκο των δεδομένων που διαθέτουμε και τον αριθμό των χαρακτηριστικών, η λύση που λαμβάνουμε μπορεί να είναι υπερβολικά προσαρμοσμένη. Αυτό είναι ένα πρόβλημα που περιγράψαμε στο πρώτο μάθημα, που συνεπάγεται ότι το μοντέλο που μαθαίνεται είναι πολύ συγκεκριμένο για τα δεδομένα που χρησιμοποιήσαμε στην εκπαίδευση του μοντέλου, οπότε το σφάλμα που παρατηρούμε με νέα δεδομένα είναι μεγαλύτερο από αυτό που μετρήσαμε για τα δεδομένα εκπαίδευσης. Αυτό μπορεί να συμβεί επειδή η γλώσσα που χρησιμοποιεί το μοντέλο μας μπορεί να περιγράψει περισσότερες υποθέσεις από αυτές που χρειάζονται τα δεδομένα και, προκειμένου να επιτευχθεί το ελάχιστο σφάλμα, έχει εξειδικευτεί πάρα πολύ. Σε αυτό το μοντέλο, η πολυπλοκότητα εμφανίζεται στον αριθμό των χαρακτηριστικών που χρησιμοποιούνται για τον υπολογισμό του συνδυασμού και στο μέγεθος των βαρών. Τα πολύ μεγάλα βάρη μπορούν να εισάγουν μεγάλες διακυμάνσεις στο συνδυασμό που μπορούν να καταστήσουν τη συνάρτηση πολύ περίπλοκη.

Η λύση για αυτό το πρόβλημα είναι αυτό που ονομάζεται κανονικοποίηση. Πρόκειται για έναν πρόσθετο όρο που προστίθεται στη συνάρτηση-στόχο που βελτιστοποιούμε (τετραγωνικό άθροισμα των υπολοίπων) προκειμένου να προτιμήσουμε ένα σύνολο βαρών που μειώνουν την πολυπλοκότητα της συνάρτησης που μαθαίνεται. Πρόκειται για μια τιμωρία στην τιμή της συνάρτησης-στόχου που αυξάνει την τιμή της όταν το μοντέλο είναι υπερβολικά πολύπλοκο. Υπάρχουν δύο κανονικοποιήσεις που χρησιμοποιούνται συνήθως σε μοντέλα γραμμικής παλινδρόμησης: η τιμωρία με την απόλυτη τιμή των βαρών του συνδυασμού και η τιμωρία με το άθροισμα του τετραγώνου των βαρών του συνδυασμού. Αυτή η τιμωρία έχει μια υπερπαράμετρο λ που ελέγχει πόσο θέλουμε να τιμωρήσουμε τα βάρη. Αυτή πρέπει να συντονίζεται κατά τη φάση της εκπαίδευσης, ώστε να λαμβάνουμε ένα μοντέλο που έχει καλή γενίκευση. Όσο μεγαλύτερη είναι αυτή η τιμή, τόσο μεγαλύτερη επίδραση έχει αυτή η τιμωρία. Όταν δημιουργείται το μοντέλο, αυτή η υπερπαράμετρος πρέπει να συντονίζεται χρησιμοποιώντας για παράδειγμα διασταυρωμένη επικύρωση.

Δέντρα αποφάσεων και παλινδρόμησης

Εισάγαμε τα δέντρα αποφάσεων στην πρώτη μονάδα μάθησης. Πρόκειται για μοντέλα που διαιρούν αναδρομικά τα παραδείγματα δημιουργώντας ένα σύνολο αποφάσεων που επιτρέπουν την ταξινόμησή τους. Αυτό σημαίνει ότι προϋποθέτει ότι η συνάρτηση εξόδου μπορεί να αναλυθεί σε έναν συνδυασμό απλών συναρτήσεων. Ένα από τα πλεονεκτήματα αυτού του αλγορίθμου είναι ότι το μοντέλο που προκύπτει μπορεί να ερμηνευτεί, αντίθετα το μοντέλο που προκύπτει μπορεί να αλλάξει πολύ ανάλογα με το δείγμα που χρησιμοποιείται για την εκπαίδευση.

Προηγουμένως μιλήσαμε μόνο για τη χρήση των δέντρων απόφασης για ταξινόμηση με χρήση διακριτών χαρακτηριστικών. Τα δέντρα αποφάσεων ενσωματώνουν συνεχή χαρακτηριστικά αναζητώντας μια δυαδική κατάτμηση των τιμών ενός χαρακτηριστικού που μεγιστοποιεί τα κριτήρια επιλογής που χρησιμοποιεί ο αλγόριθμος του δέντρου αποφάσεων. Στην περίπτωση του κέρδους πληροφορίας αναζητά έναν τρόπο διαίρεσης των τιμών του χαρακτηριστικού έτσι ώστε η τιμή αυτή να είναι η μέγιστη όλων των δυνατών διαμερισμάτων. Η διαδικασία διατάσσει τις τιμές του χαρακτηριστικού των παραδειγμάτων σε έναν κόμβο και υπολογίζει το κέρδος πληροφορίας για τις $n-1$ πιθανές κατατμήσεις (με n τον αριθμό των παραδειγμάτων). Διατηρεί την καλύτερη και η τιμή αυτή συγκρίνεται με το κέρδος πληροφορίας των υπόλοιπων χαρακτηριστικών στον κόμβο

του δέντρου. Εάν επιλεγεί μια συνεχής τιμή για έναν κόμβο απόφασης, τα δεδομένα διαιρούνται σύμφωνα με την κατάτμηση που έχει υπολογιστεί, δημιουργώντας μια απόφαση για τους δύο κλάδους που θα προστεθούν στο δέντρο, έναν για τα παραδείγματα που έχουν τιμή μικρότερη από την τιμή της κατάτμησης και έναν για τις τιμές που είναι μεγαλύτερες.

Τα δέντρα αποφάσεων μπορούν επίσης να χρησιμοποιηθούν για παλινδρόμηση, στην περίπτωση αυτή ονομάζονται δέντρα παλινδρόμησης. Το κύριο πλεονέκτημα των δέντρων παλινδρόμησης σε σχέση με τη γραμμική παλινδρόμηση είναι ότι αντί για μια μοναδική συνάρτηση που συνδυάζει όλα τα χαρακτηριστικά εισόδου, η συνάρτηση εξόδου μπορεί να διαιρεθεί σε διάφορες συναρτήσεις ώστε να προσαρμόζονται καλύτερα τα διάφορα μέρη της. Στην περίπτωση της παλινδρόμησης, το κριτήριο για την επιλογή ενός χαρακτηριστικού για έναν κόμβο απόφασης είναι το άθροισμα των τετραγωνικών καταλοίπων, όπως και στα γραμμικά μοντέλα παλινδρόμησης.

Η δημιουργία του δέντρου γίνεται με παρόμοιο τρόπο όπως και για τα δέντρα απόφασης. Για έναν διαχωρισμένο κόμβο του δέντρου, υπολογίζονται διαφορετικές παλινδρομήσεις για κάθε χαρακτηριστικό χρησιμοποιώντας τον μέσο όρο της τιμής εξόδου των παραδειγμάτων του. Κάθε χαρακτηριστικό θα δημιουργήσει ένα σύνολο παλινδρομήσεων ανάλογα με τις τιμές του. Τα διακριτά χαρακτηριστικά θα δημιουργήσουν τόσες παλινδρομήσεις σε έναν κόμβο όσες και οι τιμές που έχει, ενώ οι συνεχείς τιμές θα αναζητήσουν τη δυαδική διάσπαση που ελαχιστοποιεί το σφάλμα. Το χαρακτηριστικό που θα επιλεγεί θα είναι εκείνο που έχει το ελάχιστο άθροισμα τετραγωνικών υπολοίπων και θα δημιουργήσει το επόμενο επίπεδο του δέντρου σε αυτόν τον κόμβο.

Η κύρια διαφορά με τη γραμμική παλινδρόμηση είναι ότι δεν έχουμε ένα σύνολο βαρών αλλά μια κατάτμηση της συνάρτησης εξόδου. Κάθε φύλλο του δέντρου έχει ως πρόβλεψη τον μέσο όρο των τιμών εξόδου των παραδειγμάτων που ταξινομούνται στο συγκεκριμένο φύλλο. Αυτό περιορίζει τις πιθανές συναρτήσεις που μπορεί να υπολογίσει ένα δέντρο σε βηματικές συναρτήσεις, αλλά μειώνεται το υπολογιστικό κόστος. Υπάρχουν παραλλαγές των δέντρων παλινδρόμησης που επιτρέπουν το συνδυασμό διαφορετικών χαρακτηριστικών σε έναν κόμβο και την ύπαρξη πιο σύνθετων μοντέλων στα φύλλα, αλλά με αύξηση του υπολογιστικού κόστους.

Όπως και τα μοντέλα γραμμικής παλινδρόμησης, τα δέντρα αποφάσεων για ταξινόμηση και παλινδρόμηση μπορούν επίσης να προσαρμόζονται υπερβολικά στα δεδομένα, επειδή ένα δέντρο μπορεί να δημιουργεί κόμβους μέχρι να βρεθεί μόνο ένα παράδειγμα σε ένα φύλλο. Αυτό σημαίνει ότι μπορούμε εύκολα να απομνημονεύσουμε τα δεδομένα εκπαίδευσης μειώνοντας το σφάλμα στο μηδέν, αλλά με μεγάλο σφάλμα για την πρόβλεψη νέων παραδειγμάτων. Τα δέντρα απόφασης και τα δέντρα παλινδρόμησης διαθέτουν επίσης υπερπαραμέτρους που επιτρέπουν τη μείωση της πιθανότητας υπερπροσαρμογής. Στην περίπτωση αυτή η πολυπλοκότητα του μοντέλου αντιστοιχεί στον αριθμό των κόμβων που έχει. Τυπικά, μπορούμε να ελέγξουμε σε ένα μοντέλο μεταξύ άλλων: το μέγιστο βάθος του δέντρου, τον ελάχιστο αριθμό παραδειγμάτων που μπορεί να έχει ένα φύλλο, τον ελάχιστο αριθμό παραδειγμάτων προκειμένου να διαχωριστεί ένας κόμβος, τον αριθμό των χαρακτηριστικών που θα χρησιμοποιηθούν για τον διαχωρισμό ενός κόμβου. Αυτές οι υπερπαραμέτρους περιορίζουν τα πιθανά δέντρα που μπορούν να κατασκευαστούν, αλλά αυξάνουν την ικανότητα γενίκευσής τους. Κατά τη δημιουργία του μοντέλου, οι εν λόγω υπερπαραμέτροι πρέπει να συντονίζονται με τη χρήση, για παράδειγμα, διασταυρούμενης επικύρωσης.

Ένα μειονέκτημα αυτού του μοντέλου είναι ότι πρέπει να εκπαιδευτεί από το μηδέν εάν υπάρχουν νέα δεδομένα.

k – κοντινότεροι γείτονες

Το K-Nearest neighbors είναι ένα μοντέλο που μπορεί να χρησιμοποιηθεί για ταξινόμηση και παλινδρόμηση. Είναι ένα μη παραμετρικό μοντέλο, δηλαδή δεν υποθέτει τίποτα για τη συνάρτηση εξόδου και βασίζεται απευθείας στα δεδομένα προκειμένου να κάνει προβλέψεις. Η διαδικασία εκπαίδευσης είναι απλή, επειδή χρειάζεται μόνο να απομνημονεύσει τα δεδομένα. Ο μόνος υπολογισμός που απαιτείται είναι κατά τη στιγμή της πρόβλεψης, όταν τα κατάλληλα παραδείγματα πρέπει να ανακτηθούν και να συνδυαστούν για την απόκτηση της εξόδου για ένα νέο παράδειγμα. Συνοπτικά, υπολογίζει μια τοπική προσέγγιση της τοπικής συνάρτησης χρησιμοποιώντας τα πλησιέστερα παραδείγματα στο σημείο πρόβλεψης.

Προκειμένου να γίνουν προβλέψεις, πρέπει να αποφασιστούν τρία στοιχεία που θα αποτελέσουν τις υπερπαραμέτρους αυτού του μοντέλου. Πρώτον, πώς θα συγκρίνουμε τα παραδείγματα μεταξύ τους. Αυτό σημαίνει να αποφασίσουμε μια συνάρτηση ομοιότητας ή απόστασης που θα χρησιμοποιηθεί για να ανακτήσουμε τα πιο κοντινά παραδείγματα σε αυτό που θέλουμε να προβλέψουμε. Διαφορετικές συναρτήσεις απόστασης θα οδηγήσουν σε διαφορετικές προβλέψεις. Ορισμένες κοινές επιλογές είναι οι συναρτήσεις απόστασης Ευκλείδειας και Μανχάταν, αλλά η επιλογή θα εξαρτηθεί από τα χαρακτηριστικά των δεδομένων. Το δεύτερο στοιχείο που πρέπει να αποφασίσουμε είναι πόσα παραδείγματα θα χρησιμοποιηθούν για τον υπολογισμό της εξόδου, αυτό είναι το k στο όνομα του μοντέλου. Η τελική απόφαση είναι ο τρόπος με τον οποίο θα συνδυαστούν τα παραδείγματα για να παραχθεί η έξοδος, μια συνάρτηση συνδυασμού. Αυτή θα είναι διαφορετική για την ταξινόμηση και την παλινδρόμηση. Συνήθως, για την ταξινόμηση τα παραδείγματα ψηφίζουν για να πάρουν την απόφαση, οπότε η πλειοψηφία αποφασίζει την ταξινόμηση. Για την παλινδρόμηση μπορεί να χρησιμοποιηθεί ο μέσος όρος των προβλέψεων όπως στα δέντρα παλινδρόμησης. Μια άλλη επιλογή είναι να συνδυάσετε τις εξόδους αναθέτοντας ένα βάρος που αντιστοιχεί στην απόσταση του παραδείγματος από τα ανακτημένα παραδείγματα. Όλες αυτές οι αποφάσεις πρέπει να επιλέγονται χρησιμοποιώντας για παράδειγμα διασταυρούμενη επικύρωση.

Αυτό το μοντέλο επηρεάζεται σε μεγάλο βαθμό από την κατάρτα της διαστατικότητας, καθώς εργάζεται με αποστάσεις/ομοιότητες. Αυτό σημαίνει ότι θα έχει προβλήματα στην επίτευξη καλών προβλέψεων όταν ο αριθμός των διαστάσεων εισόδου είναι πολύ μεγάλος. Επίσης, μπορεί να είναι υπολογιστικά ακριβό εάν τα δεδομένα εκπαίδευσης είναι μεγάλα, επειδή ο χρόνος πρόβλεψης αυξάνεται γραμμικά με τον αριθμό των παραδειγμάτων. Ένα πλεονέκτημα αυτού του μοντέλου είναι ότι μπορούν να προστεθούν νέα δεδομένα αφού, δεν απαιτείται νέα προσαρμογή των παραμέτρων.

Naive Bayes - Ναιβ Μπέης

Το μοντέλο αυτό παρουσιάστηκε στην προηγούμενη ενότητα μάθησης. Η αρχή του μοντέλου είναι να υποθέσουμε ότι όλα τα χαρακτηριστικά εισόδου είναι ανεξάρτητα μεταξύ τους και ότι η πρόβλεψη μπορεί να υπολογιστεί ως το γινόμενο των υπό συνθήκη πιθανοτήτων κάθε κλάσης δεδομένων των τιμών των χαρακτηριστικών ενός νέου παραδείγματος. Στην περίπτωση διακριτών χαρακτηριστικών, μπορούμε να εκτιμήσουμε την κατανομή των χαρακτηριστικών ως πολυωνυμικές κατανομές, όπως εξηγήθηκε στην προηγούμενη ενότητα. Όταν έχουμε συνεχή χαρακτηριστικά, θα πρέπει να επιλέξουμε μια κατανομή πιθανοτήτων που ταιριάζει στο χαρακτηριστικό, για παράδειγμα την κατανομή Gauss. Πρέπει απλώς να εκτιμήσουμε τις παραμέτρους της κατανομής χρησιμοποιώντας τα δεδομένα εισόδου.

Η εκτίμηση της πιθανότητας για ένα νέο παράδειγμα που έχει συνεχείς κατανομές είναι παρόμοια με τη διακριτή περίπτωση, έχουμε το γινόμενο των πιθανοτήτων των τιμών των χαρακτηριστικών. Το κύριο πρόβλημα είναι ότι η τιμή που θα λάβουμε για κάθε χαρακτηριστικό θα είναι πολύ μικρή και τα γινόμενα θα είναι ακόμη μικρότερα. Ο τρόπος για να αποφύγουμε αυτό το πρόβλημα είναι να εργαστούμε με τον λογάριθμο της συνάρτησης πιθανότητας. Στην περίπτωση αυτή, το γινόμενο θα μετατραπεί σε άθροισμα λογαρίθμων. Δεδομένου ότι οι πιθανότητες είναι τιμές μικρότερες της μονάδας, όλες οι τιμές θα είναι αρνητικές, οπότε η απόφαση θα αντιστοιχεί στον μεγαλύτερο αριθμό. Οι τιμές μπορούν επίσης να κανονικοποιηθούν ώστε το άθροισμα να είναι ένα, οπότε προκύπτει μια σωστή πιθανότητα. Στην περίπτωση που έχουμε ένα μείγμα συνεχών και διακριτών κατανομών πιθανοτήτων όλες οι πιθανότητες μπορούν να μετασχηματιστούν με τη χρήση λογαρίθμων.

Για αυτό το μοντέλο οι υπερπαραμέτροι είναι οι κατανομές πιθανότητας που χρησιμοποιούνται για τα χαρακτηριστικά. Αυτές μπορούν να επιλεγούν από γνωστές κατανομές πιθανοτήτων. Η διασταυρούμενη επικύρωση μπορεί να χρησιμοποιηθεί για να ληφθεί η απόφαση σχετικά με την εν λόγω κατανομή. Ένα πλεονέκτημα αυτού του τρόπου είναι ότι μπορεί να εκπαιδευτεί σταδιακά εάν υπάρχουν νέα δεδομένα.

Επιλέγοντας έναν αλγόριθμο

Υπάρχουν δύο διαφορετικές εργασίες που επιλύονται από αυτούς τους αλγορίθμους, η ταξινόμηση και η παλινδρόμηση. Και οι τέσσερις αλγόριθμοι αποτελούν τις λιγότερο δαπανηρές επιλογές για τις εργασίες αυτές και είναι οι πρώτοι που χρησιμοποιούνται σε πρακτικές εφαρμογές. Για την παλινδρόμηση, η γραμμική παλινδρόμηση και οι k-κοντινότεροι γείτονες είναι καλές επιλογές. Για μικρά σύνολα δεδομένων και απλές εργασίες η γραμμική παλινδρόμηση θα δώσει καλά αποτελέσματα ειδικά όταν χρησιμοποιείται κανονικοποίηση και αν εισάγονται μη γραμμικοί μετασχηματισμοί όταν χρειάζεται, αλλά αυτό χρειάζεται καλή γνώση του προβλήματος. Ο k-κοντινότερος γείτονας θα είναι μια καλή επιλογή εάν το σύνολο δεδομένων είναι μεγάλο. Για την ταξινόμηση, ο naive bayes είναι ο απλούστερος αλγόριθμος και μπορεί να χρησιμοποιηθεί ως βασική γραμμή. Οι υποθέσεις του είναι πολύ ισχυρές, δεν υπάρχει εξάρτηση μεταξύ των χαρακτηριστικών, αλλά πολλές φορές λειτουργεί καλά ακόμη και όταν δεν ισχύουν, όταν υπάρχουν αρκετά δεδομένα για το έργο. Ο K nearest neighbors θα λειτουργήσει καλά με μεγάλα σύνολα δεδομένων και πολύπλοκα προβλήματα. Τα δέντρα αποφάσεων έχουν το πλεονέκτημα ότι είναι ερμηνεύσιμα και λειτουργούν ανεξάρτητα από το μέγεθος του συνόλου δεδομένων.

Προφανώς, η τελική επιλογή του καλύτερου αλγορίθμου για ένα πρόβλημα εξαρτάται από την απόδοση του μοντέλου που προκύπτει από τα δεδομένα. Είναι δύσκολο να ληφθούν αποφάσεις χωρίς την εφαρμογή των αλγορίθμων και τη ρύθμιση των παραμέτρων τους ώστε να επιτευχθεί το καλύτερο αποτέλεσμα.

Περιπτώσεις χρήσης

TIMES ΣΠΙΤΙΩΝ ΣΤΗΝ ΚΑΛΙΦΟΡΝΙΑ

Θα συνεχίσουμε την εξερεύνηση του συνόλου δεδομένων για τις τιμές των κατοικιών στην Καλιφόρνια. Πρόκειται για ένα πρόβλημα παλινδρόμησης, δεδομένου ότι η σχετική έξοδος είναι συνεχής. Μπορούμε να χρησιμοποιήσουμε για το έργο αυτό μοντέλα γραμμικής παλινδρόμησης, k-κοντινότερων γειτόνων και δέντρων παλινδρόμησης. Για την εκπαίδευση ενός μοντέλου μηχανικής μάθησης πρέπει να προχωρήσουμε συστηματικά. Πρέπει να χωρίσουμε τα δεδομένα σε σύνολα εκπαίδευσης και δοκιμής και να εκτιμήσουμε το σφάλμα χρησιμοποιώντας μια διαδικασία επικύρωσης. Ο πιο συνηθισμένος τρόπος είναι η χρήση της διασταυρούμενης επικύρωσης. Στην περίπτωση χρήσης αλγορίθμων που διαθέτουν υπερπαραμέτρους, αυτές πρέπει επίσης να προσαρμοστούν χρησιμοποιώντας μια διαδικασία επικύρωσης. Η διερεύνηση των υπερπαραμέτρων μπορεί να γίνει με διάφορους τρόπους, για παράδειγμα, εξαντλητικά, λαμβάνοντας ένα μοντέλο για κάθε πιθανό συνδυασμό ενός εύρους τιμών των παραμέτρων του αλγορίθμου, αυτό είναι γνωστό ως αναζήτηση πλέγματος. Η τυχαία αναζήτηση μπορεί να εφαρμοστεί όταν ο αριθμός των συνδυασμών που πρέπει να ελεγχθούν είναι πολύ μεγάλος. Η τελική απόφαση για το ποιο μοντέλο ή ποιοι αλγόριθμοι είναι οι καλύτεροι θα πρέπει να σχετίζεται με το μέτρο απόδοσης που έχει επιλεγεί για την προς επίλυση εργασία (ταξινόμηση/παλινδρόμηση), φροντίζοντας ώστε η απόδοση για τα δεδομένα εκπαίδευσης να είναι παρόμοια με εκείνη από τα δεδομένα δοκιμής, ώστε να μην υπάρχει υπερπροσαρμογή.

ΒΗΜΑ 1: Θα αρχίσουμε να φορτώνουμε τα δεδομένα και να μετασχηματίζουμε τις μεταβλητές "AveBedrms", "AveOccur" και "AveRooms" όπως κάναμε προηγουμένως για να λάβουμε δεδομένα που έχουν πιο γκαουσιανή συμπεριφορά:

```
import sklearn.datasets as dsets
import numpy as np
hframe= dsets.fetch_california_housing(as_frame=True)['frame']
hdata = hframe.loc[:, ['MedInc', 'HouseAge', 'Population','Latitude','Longitude']].copy()
hdata['logAveBedrms'] = np.log(hframe['AveBedrms'])
hdata['logAveOccur'] = np.log(hframe['AveOccur'])
hdata['logAveRooms'] = np.log(hframe['AveRooms'])
```

ΒΗΜΑ 2: Πρέπει να χωρίσουμε τα δεδομένα σε ένα σύνολο εκπαίδευσης και ένα σύνολο δοκιμής. Σε αυτή την περίπτωση χρησιμοποιώντας την προεπιλογή ο διαχωρισμός θα είναι 75%-25%,

μπορούμε να ορίσουμε τη γεννήτρια τυχαίων αριθμών που χρησιμοποιείται για τη δημιουργία των δειγμάτων, ώστε τα αποτελέσματα να είναι αναπαραγώγιμα:

```
from sklearn.model_selection import train_test_split
h_train, h_test, t_train, t_test = train_test_split(hdata, htarget, random_state=0)
```

ΒΗΜΑ 3: Κανονικοποιούμε επίσης τα δεδομένα ώστε να έχουν μέσο όρο μηδέν και μοναδιαία τυπική απόκλιση. Για να κανονικοποιήσουμε τα δεδομένα, η συνάρτηση κανονικοποίησης πρέπει να προσαρμοστεί στα δεδομένα εκπαίδευσης και στη συνέχεια η κανονικοποίηση πρέπει να εφαρμοστεί στα δεδομένα εκπαίδευσης και δοκιμής.

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
h_train = scaler.fit_transform(h_train)
h_test = scaler.transform(h_test)
```

ΒΗΜΑ 4: Ξεκινάμε με ένα μοντέλο γραμμικής παλινδρόμησης. Αυτό το μοντέλο δεν έχει υπερπαραμέτρους, οπότε μπορούμε να εφαρμόσουμε άμεσα τη διασταυρούμενη επικύρωση για να εκτιμήσουμε το σφάλμα του μοντέλου που προκύπτει με τα δεδομένα εκπαίδευσης. Εφαρμόζουμε 10 φορές διασταυρούμενη επικύρωση και χρησιμοποιούμε το μέσο τετραγωνικό σφάλμα ως μέτρο απόδοσης. Από προεπιλογή το scikit learn χρησιμοποιεί το R2 (συντελεστής προσδιορισμού) ως μέτρο απόδοσης, υποδεικνύουμε στη συνάρτηση cross-validation το μέτρο που θέλουμε να χρησιμοποιήσουμε.

```
from sklearn.model_selection import cross_validate
from sklearn.linear_model import LinearRegression
lreg = LinearRegression()
cv_results = cross_validate(lreg, h_train, t_train, cv=10, scoring=make_scorer(mean_squared_error,
greater_is_better=False))
```

Αυτό επιστρέφει ένα λεξικό με τα αποτελέσματα, όπου στο κλειδί 'test_score' έχουμε το μέσο τετραγωνικό σφάλμα κάθε διπλώματος επικύρωσης (το σφάλμα του μοντέλου που εκπαιδεύτηκε με 9 διπλώματα και δοκιμάστηκε στο υπόλοιπο διπλάσιο). Υπολογίζουμε το μέσο όρο αυτών των τιμών για να έχουμε το εκτιμώμενο σφάλμα του μοντέλου χρησιμοποιώντας όλα τα δεδομένα.

```
np.mean(cv_results['test_score'])
-0.44320720946809944
```

ΒΗΜΑ 5: Μπορούμε τώρα να εκπαιδεύσουμε ένα μοντέλο με όλα τα δεδομένα και να υπολογίσουμε το σφάλμα:


```
reg.fit(h_train, t_train)
mean_squared_error(t_test, lreg.predict(h_test))
0.4641412688207677
```

Βλέπουμε ότι το σφάλμα είναι παρόμοιο, άρα τα παραδείγματα που έχουμε στην εκπαίδευση και τη δοκιμή έχουν παρόμοια συμπεριφορά.

ΒΗΜΑ 6: Μπορούμε να εκπαιδεύσουμε κανονικοποιημένες εκδόσεις της γραμμικής παλινδρόμησης για να ελέγξουμε αν αυτό επιτρέπει τη μείωση του σφάλματος που θα μπορούσε να οφείλεται σε υπερβολική προσαρμογή, αλλά στην περίπτωσή μας το σύνολο δεδομένων είναι μεγάλο, οπότε η πιθανότητα αυτού είναι μικρή. Πρώτα πρέπει να ρυθμίσουμε την παράμετρο κανονικοποίησης, αυτό πρέπει να γίνει με τη χρήση διασταυρούμενης επικύρωσης.

```
from sklearn.linear_model import Ridge, RidgeCV, Lasso, LassoCV
lambdas = np.linspace(1, 2, num=100)
model_ridgeCV = RidgeCV(alphas=lambdas, gcv_mode='auto')
model_ridgeCV.fit(h_train, t_train)
model_ridgeCV.alpha_
1.5151515151515151
```

Αυτή είναι η καλύτερη παράμετρος για την εξομάλυνση της γραμμικής παλινδρόμησης με τη χρήση της παλινδρόμησης Ridge.

ΒΗΜΑ 7: Μπορούμε τώρα να υπολογίσουμε την παλινδρόμηση και να εκτιμήσουμε το σφάλμα του μοντέλου όπως κάναμε προηγουμένως χρησιμοποιώντας επίσης διασταυρωμένη επικύρωση:

```
ridge = Ridge(alpha=model_ridgeCV.alpha_)
cv_results = cross_validate(ridge, h_train, t_train,
cv=10, scoring=make_scorer(mean_squared_error, greater_is_better=False))
np.mean(cv_results['test_score'])
-0.4432071503685144
```

Αυτό είναι σχεδόν πανομοιότυπο με την παλινδρόμηση χωρίς κανονικοποίηση. Μπορούμε επίσης να δοκιμάσουμε την παλινδρόμηση Lasso:

```
lambdas = np.linspace(0.00001, 1, num=1000)
model_LassoCV = LassoCV(alphas=lambdas)
model_LassoCV.fit(h_train, t_train)
model_LassoCV.alpha_
lasso = Lasso(alpha=model_LassoCV.alpha_)
```

```
cv_results = cross_validate(lasso, h_train, t_train,  
cv=10,scoring=make_scorer(mean_squared_error, greater_is_better=False))  
np.mean(cv_results['test_score'])  
0.46413881276008334
```

Αυτό είναι επίσης σχεδόν πανομοιότυπο, οπότε η κανονικοποίηση δεν δημιουργεί καλύτερο μοντέλο.

ΒΗΜΑ 8: Μπορούμε τώρα να προχωρήσουμε στα δέντρα παλινδρόμησης. Σε αυτή την περίπτωση, έχουμε αρκετές υπερπαραμέτρους, θα εξερευνήσουμε μόνο τον ελάχιστο αριθμό κόμβων σε ένα φύλλο και το μέγιστο βάθος. Σε αυτή την περίπτωση, θα εκτελέσουμε μια αναζήτηση πλέγματος δημιουργώντας όλα τα μοντέλα που αντιστοιχούν στο συνδυασμό μιας λίστας τιμών για κάθε παράμετρο και θα εκτιμήσουμε το σφάλμα του καλύτερου μοντέλου χρησιμοποιώντας διασταυρωμένη επικύρωση. Το καλύτερο μοντέλο αποθηκεύεται από τη συνάρτηση αναζήτησης πλέγματος με διασταυρούμενη επικύρωση.

```
from sklearn.model_selection import GridSearchCV  
param = {'min_samples_leaf':[1, 2, 3, 5, 10, 25], 'max_depth':[5,10,15, 20, 25]}  
dtr = DecisionTreeRegressor()  
gstree = GridSearchCV(dtr,param,cv=10,scoring=make_scorer(mean_squared_error,  
greater_is_better=False))  
gstree.fit(h_train, h_train)
```

ΒΗΜΑ 9: Το χαρακτηριστικό `best_estimator_` περιέχει το μοντέλο με την καλύτερη απόδοση, τώρα εφαρμόζουμε διασταυρωμένη επικύρωση για να υπολογίσουμε την απόδοσή του:

```
cv_results = cross_validate(gstree.best_estimator_, h_train, t_train,  
cv=10,scoring=make_scorer(mean_squared_error, greater_is_better=False))  
np.mean(cv_results['test_score'])  
-0.47847059063842073
```

Αυτό είναι χειρότερο από την απόδοση που επιτυγχάνεται με τη χρήση γραμμικής παλινδρόμησης.

ΒΗΜΑ 10: Μπορούμε να υπολογίσουμε το σφάλμα για τα δεδομένα δοκιμής εκπαιδευόντας ένα δέντρο παλινδρόμησης με όλα τα δεδομένα εκπαίδευσης:

```
gstree.best_estimator_.fit(h_train,t_train)  
mean_squared_error(t_test,gstree.best_estimator_.predict(h_test))  
0.5115134750811695
```

Αυτό έχει παρόμοια απόδοση όπως εκτιμάται με τη διασταυρούμενη επικύρωση.

ΒΗΜΑ 11: Τέλος, θα εκπαιδεύσουμε έναν k-κοντινότερο γείτονα για παλινδρόμηση. Σε αυτή την περίπτωση έχουμε επίσης υπερπαραμέτρους, θα ρυθμίσουμε τον αριθμό των γειτόνων, τη στάθμιση των γειτόνων κατά την πραγματοποίηση των προβλέψεων και το μέγεθος των φύλλων της δομής δεδομένων που χρησιμοποιείται για την ευρετηρίαση και αναζήτηση των γειτόνων.

```
from sklearn.neighbors import KNeighborsRegressor
param = {'n_neighbors':[5,7,15, 23, 37], 'weights':['distance', 'uniform'],'leaf_size':[5,10, 20, 30]}
knn = KNeighborsRegressor(n_neighbors=11, weights='distance')
knn = GridSearchCV(knn,param,cv=10,scoring=make_scorer(mean_squared_error,
greater_is_better=False))
knn.fit(h_train, h_train)
```

Αυτό επιστρέφει το μοντέλο με τον καλύτερο συνδυασμό παραμέτρων

ΒΗΜΑ 12: Τώρα εκτιμούμε το σφάλμα με διασταυρούμενη επικύρωση:

```
cv_results = cross_validate(knn.best_estimator_, h_train, t_train,
cv=10,scoring=make_scorer(mean_squared_error, greater_is_better=False))
np.mean(cv_results['test_score'])
-0.36212272456634353
```

Αυτή η απόδοση είναι καλύτερη από τους άλλους δύο αλγορίθμους.

Τέλος, μπορούμε να ελέγξουμε το σφάλμα με τα δεδομένα δοκιμής

```
knn.best_estimator_.fit(h_train,t_train)
mean_squared_error(t_test,knn.best_estimator_.predict(h_test))
0.36468014275597765
```

Αυτό συνάδει με το σφάλμα διασταυρούμενης επικύρωσης.

Μπορούμε λοιπόν να συμπεράνουμε ότι το καλύτερο μοντέλο για αυτό το σύνολο δεδομένων από τα τρία μοντέλα που δοκιμάσαμε είναι το k-Nearest Neighbours που έχει την καλύτερη απόδοση μετρούμενη από το μέσο τετραγωνικό σφάλμα.

ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 2: ΟΙΝΟΠΑΡΑΓΩΓΟΙ

Για την περίπτωση αυτή, θα εργαστούμε με το σύνολο δεδομένων για το κρασί που περιλαμβάνεται επίσης στο scikit-learn. Πρόκειται για μια εργασία ταξινόμησης που συνίσταται στην ταξινόμηση διαφορετικών κρασιών σε τρεις κλάσεις που αντιστοιχούν σε τρεις διαφορετικούς καλλιεργητές στην ίδια περιοχή της Ιταλίας. Κάθε κρασί περιγράφεται από τα αποτελέσματα ενός συνόλου χημικών αναλύσεων:

Αλκοόλ

Μηλικό οξύ

Τέφρα

Αλκαλικότητα της τέφρας

Μαγνήσιο

Ολικές φαινόλες

Φλαβονοειδή

Μη φλαβονοειδείς φαινόλες

Προανθοκυανίνες

Ένταση χρώματος

Απόχρωση

OD280/OD315 αραιωμένων οίνων

Προλίνη

Όλα αυτά είναι συνεχή χαρακτηριστικά.

Για την εκπαίδευση ενός μοντέλου μηχανικής μάθησης πρέπει να προχωρήσουμε συστηματικά. Πρέπει να χωρίσουμε τα δεδομένα σε σύνολο εκπαίδευσης και δοκιμής και να εκτιμήσουμε το σφάλμα χρησιμοποιώντας μια διαδικασία επικύρωσης. Ο πιο συνηθισμένος τρόπος είναι η χρήση της διασταυρούμενης επικύρωσης. Στην περίπτωση χρήσης αλγορίθμων που διαθέτουν υπερπαραμέτρους πρέπει να προσαρμοστούν και αυτές με τη χρήση μιας διαδικασίας επικύρωσης. Η διερεύνηση των υπερπαραμέτρων μπορεί να γίνει με διάφορους τρόπους, για παράδειγμα, εξαντλητικά, λαμβάνοντας ένα μοντέλο για κάθε πιθανό συνδυασμό ενός εύρους τιμών των παραμέτρων του αλγορίθμου, αυτό είναι γνωστό ως αναζήτηση πλέγματος. Η τυχαία αναζήτηση μπορεί να εφαρμοστεί όταν ο αριθμός των συνδυασμών που πρέπει να ελεγχθούν είναι πολύ μεγάλος. Η τελική απόφαση για το ποιο μοντέλο ή ποιοι αλγόριθμοι είναι οι καλύτεροι θα πρέπει να σχετίζεται με το μέτρο απόδοσης που έχει επιλεγεί για την προς επίλυση εργασία (ταξινόμηση/ παλινδρόμηση), φροντίζοντας ώστε η απόδοση για τα δεδομένα εκπαίδευσης να είναι παρόμοια με εκείνη από τα δεδομένα δοκιμής, ώστε να μην υπάρχει υπερπροσαρμογή.

Δεδομένου ότι πρόκειται για εργασία ταξινόμησης μπορούμε να χρησιμοποιήσουμε ένα δέντρο απόφασης, k-κοντινότερους γείτονες και naïve Bayes.

Κατεβάστε τον κώδικα [εδώ](#)

ΒΗΜΑ 1: Θα ξεκινήσουμε με τη φόρτωση των δεδομένων. Σε αυτή την περίπτωση αυτό που θα κάνουμε είναι να τα φορτώσουμε ως δύο πίνακες numpy, έναν για τα δεδομένα εισόδου και έναν για τα δεδομένα εξόδου:

```
import sklearn.datasets as datasets
X,y = datasets.load_wine(return_X_y=True)
```

Πρέπει να χωρίσουμε τα δεδομένα σε ένα σύνολο εκπαίδευσης και ένα σύνολο δοκιμής, θα χρησιμοποιήσουμε το προεπιλεγμένο 75%-25%. Θα κανονικοποιήσουμε επίσης τις τιμές των χαρακτηριστικών, μπορούμε να ορίσουμε τη γεννήτρια τυχαίων αριθμών που χρησιμοποιείται για τη δημιουργία των δειγμάτων, ώστε τα αποτελέσματα να είναι αναπαραγώγιμα:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, random_state=0)
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

ΒΗΜΑ 2: Θα ξεκινήσουμε με τα δέντρα απόφασης, Σε αυτή την περίπτωση, έχουμε αρκετές υπερπαραμέτρους, θα εξερευνήσουμε μόνο τον ελάχιστο αριθμό κόμβων σε ένα φύλλο και το μέγιστο βάθος. Σε αυτή την περίπτωση θα εκτελέσουμε μια αναζήτηση πλέγματος δημιουργώντας όλα τα μοντέλα που αντιστοιχούν στο συνδυασμό μιας λίστας τιμών για κάθε παράμετρο και θα εκτιμήσουμε την απόδοση του μοντέλου χρησιμοποιώντας διασταυρωμένη επικύρωση. Αυτό γίνεται από το scikit learn αυτόματα, παρέχοντας μετά την εκπαίδευση με το καλύτερο μοντέλο από όλα όσα έχουν δοκιμαστεί. Στην περίπτωση της ταξινόμησης, η απόδοση αντιστοιχεί στην ακρίβεια του μοντέλου.

```
from sklearn.model_selection import cross_validate
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import accuracy_score
from sklearn.tree import DecisionTreeClassifier
param = {'min_samples_leaf':[2, 3, 5, 10, 25], 'max_depth':[5,10,15, 20, 25]}
dtr = DecisionTreeClassifier()
gstr = GridSearchCV(dtr,param,cv=10)
gstr.fit(X_train, y_train)
```

Αυτό θα μας δώσει τον καλύτερο συνδυασμό υπερπαραμέτρων σύμφωνα με την εκτιμώμενη ακρίβεια από τη διασταυρούμενη επικύρωση. Μπορούμε να εκπαιδεύσουμε αυτό το μοντέλο και να υπολογίσουμε το εκτιμώμενο σφάλμα:

```
cv_results = cross_validate(gstree.best_estimator_, X_train, y_train, cv=10)
np.mean(cv_results['test_score'])
0.8961538461538462
```

Και τέλος, η ακρίβεια με τα δεδομένα δοκιμής:

```
gstree.best_estimator_.fit(X_train,y_train)
accuracy_score(y_test,gstree.best_estimator_.predict(X_test))
0.9333333333333333
```

ΒΗΜΑ 3: Το δεύτερο μοντέλο προς δοκιμή είναι το μοντέλο k-κοντινότερων γειτόνων. Θα προχωρήσουμε με τον ίδιο τρόπο. Πρώτα θα ρυθμίσουμε τις υπερπαραμέτρους, θα ρυθμίσουμε τον αριθμό των γειτόνων, τη στάθμιση των γειτόνων κατά την πραγματοποίηση προβλέψεων και το μέγεθος των φύλλων της δομής δεδομένων που χρησιμοποιείται για την ευρετηρίαση και αναζήτηση των γειτόνων.

```
from sklearn.neighbors import KNeighborsClassifier
param = {'n_neighbors':[3, 5, 7, 11, 15, 23, 37], 'weights':['distance', 'uniform'], 'leaf_size':[5, 10, 20, 30]}
knn = KNeighborsClassifier(n_neighbors=11, weights='distance')
knn = GridSearchCV(knn,param,cv=10)
knn.fit(X_train, y_train)
```

Αυτό μας δίνει τις καλύτερες υπερπαραμέτρους σύμφωνα με την ακρίβεια που υπολογίζεται από τη διασταυρούμενη επικύρωση. Μπορούμε να υπολογίσουμε την ακρίβεια του μοντέλου με διασταυρωμένη επικύρωση:

```
cv_results = cross_validate(knn.best_estimator_, X_train, y_train, cv=10)
np.mean(cv_results['test_score'])
0.9637362637362639
```

Και μπορούμε να υπολογίσουμε την ακρίβεια για τα δεδομένα δοκιμής:

```
knn.best_estimator_.fit(X_train,y_train)
accuracy_score(y_test,knn.best_estimator_.predict(X_test))
0.9777777777777777
```

Αυτό συνάδει με την ακρίβεια της διασταυρούμενης επικύρωσης και είναι ελαφρώς καλύτερο από την ακρίβεια του μοντέλου δέντρων απόφασης.

ΒΗΜΑ 4: Για το αφελές μοντέλο Bayes, έχουμε συνεχή δεδομένα, θα μπορούσαμε να υποθέσουμε ότι όλα τα χαρακτηριστικά κατανέμονται ως Γκαουσιανοί, οπότε μπορούμε να χρησιμοποιήσουμε το γκαουσιανό αφελές μοντέλο Bayes. Αυτό το μοντέλο δεν έχει υπερπαραμέτρους, οπότε μπορούμε να εκτιμήσουμε απευθείας την ακρίβεια διασταυρούμενης επικύρωσης:

```
from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()
cv_results = cross_validate(gnb, X_train, y_train, cv=10)
np.mean(cv_results['test_score'])
0.970329670329670
Και η ακρίβεια για τα δεδομένα δοκιμής:
gnb.fit(X_train, y_train)
accuracy_score(y_test, gnb.predict(X_test))
0.9333333333333333
```

Σε αυτή την περίπτωση, οι k-κοντινότεροι γείτονες επιστρέφουν το μοντέλο με το καλύτερο αποτέλεσμα για τα δεδομένα δοκιμής, αλλά πλησιάζουν με το μοντέλο naive Bayes για την

	precision	recall	f1-score	support
0	1.00	1.00	1.00	16
1	0.95	1.00	0.98	20
2	1.00	0.89	0.94	9
accuracy			0.98	45
macro avg	0.98	0.96	0.97	45
weighted avg	0.98	0.98	0.98	45

ακρίβεια της διασταυρούμενης επικύρωσης. Μπορούμε να χρησιμοποιήσουμε άλλα μέτρα και να επιθεωρήσουμε τα αποτελέσματα ανά κλάση για να δούμε αν υπάρχουν διαφορές.

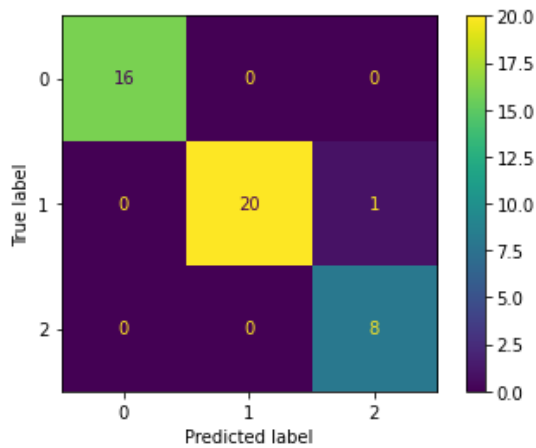
ΒΗΜΑ 5: Τέλος, μπορούμε να προχωρήσουμε με τις αναφορές ταξινόμησης για κάθε μοντέλο:

```
from sklearn.metrics import classification_report
```

	precision	recall	f1-score	support
0	0.88	1.00	0.93	14
1	0.95	0.91	0.93	22
2	1.00	0.89	0.94	9
accuracy			0.93	45
macro avg	0.94	0.93	0.93	45
weighted avg	0.94	0.93	0.93	45

```
print(classification_report(knn.best_estimator_.predict(X_test), y_test))
```

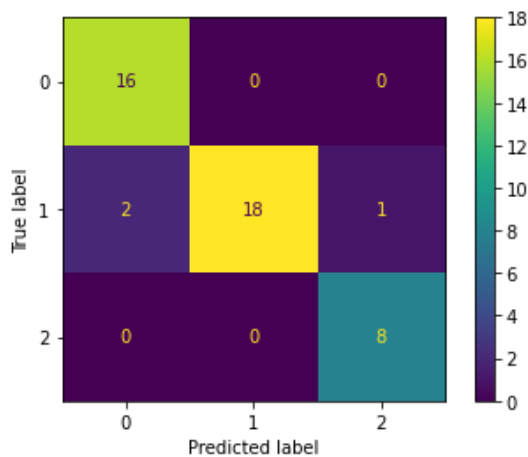
```
print(classification_report(gnb.predict(X_test),y_test))
```



Βλέπουμε ότι και οι δύο ταξινομητές έχουν διαφορά σε μία από τις κλάσεις, οι k-κοντινότεροι γείτονες ταξινομούν τέλεια τις κλάσεις 0 και 2 και ο αφελής Bayes ταξινομεί τέλεια τις κλάσεις 1 και 2. Μπορούμε να εξετάσουμε τον πίνακα σύγχυσης για να δούμε πού βρίσκεται η διαφορά.

Αυτός είναι ο πίνακας σύγχυσης για τους k-κοντινότερους γείτονες

Και αυτός είναι ο πίνακας σύγχυσης για gaussian naive bayes:



Βλέπουμε ότι υπάρχουν κάποια προβλήματα για την κατηγορία 1 και για τα δύο μοντέλα, αλλά το αφελής Bayes έχει περισσότερα προβλήματα με την κατηγορία αυτή. Μπορούμε να δούμε ότι η συνολική απόδοση για το k-nearest neighbours είναι ελαφρώς καλύτερη με υψηλότερη μέση απόδοση και σταθμισμένη με το μέγεθος των κλάσεων σύμφωνα με την έκθεση ταξινόμησης.

Αναφορές

- Introduction to machine learning Alpaydin, E, The MIT Press, 2014. ISBN: 9780262028189 (chapter 3, 8, 9)
- The elements of statistical learning: data mining, inference, and prediction Hastie, T.; Tibshirani, R.; Friedman, J, Springer, 2009. ISBN: 9780387848570 (chapter 3, 9, 13)
- Pattern recognition and machine learning Bishop, C.M, Springer, 2006. ISBN: 0387310738 (chapter 2,3)

Μάθημα 2.5: Μη γραμμικές μέθοδοι για επιβλεπόμενη μάθηση

Θέματα προς συζήτηση

- Εισαγωγή
- Μηχανές διανυσματικής υποστήριξης
- Πολυστρωματικό Perceptron
- Συγκροτήματα μοντέλων

Σημειώσεις διάλεξης

Εισαγωγή

Οι μη γραμμικές μέθοδοι επεκτείνουν τη γλώσσα που μπορεί να χρησιμοποιηθεί για τον ορισμό της σχέσης μεταξύ των χαρακτηριστικών που περιγράφουν τα δεδομένα εισόδου και την έξοδό τους. Αυτά τα μοντέλα υποθέτουν ότι υπάρχουν πολύπλοκες αλληλεπιδράσεις μεταξύ των χαρακτηριστικών των δεδομένων που πρέπει να μοντελοποιηθούν προκειμένου να καθοριστεί η έξοδος. Υπάρχουν πολλοί πιθανοί τρόποι μετασχηματισμού και συνδυασμού των δεδομένων με πολύπλοκους τρόπους, οπότε αυτό ανοίγει ένα μεγάλο εύρος επιλογών που καθιστά τη διαδικασία μάθησης πιο δαπανηρή. Δεν μπορούμε να εγγυηθούμε ότι θα λαμβάνουμε πάντα ακριβείς και μοναδικές λύσεις για το μοντέλο που θέλουμε να μάθουμε και πρέπει να επιλύονται σύνθετα προβλήματα βελτιστοποίησης αντί να εφαρμόζονται εύκολες στην κατανόηση και απλές διαδικασίες. Αυτό σημαίνει επίσης μεγαλύτερη εξάρτηση των αποτελεσμάτων όχι μόνο από τα δεδομένα που χρησιμοποιούμε, αλλά και από τις αρχικές συνθήκες των αλγορίθμων που χρησιμοποιούνται για την εκμάθηση των μοντέλων.

Από όλους τους μη γραμμικούς αλγορίθμους με επίβλεψη θα επικεντρωθούμε στις μηχανές διανυσμάτων υποστήριξης, στις πολυεπίπεδες perceptrons και στα σύνολα μοντέλων. Αυτοί είναι οι συνήθεις αλγόριθμοι που χρησιμοποιούνται σε πρακτικές εφαρμογές και μπορούν να βρεθούν στις περισσότερες βιβλιοθήκες μηχανικής μάθησης και πακέτα λογισμικού που χρησιμοποιούνται από τη βιομηχανία.

Μηχανές διανυσμάτων υποστήριξης

Οι μηχανές διανυσμάτων υποστήριξης (SVM) είναι μοντέλα διάκρισης που μπορούν να χρησιμοποιηθούν για εργασίες ταξινόμησης και παλινδρόμησης. Για τα προβλήματα ταξινόμησης ο στόχος είναι να βρεθεί το όριο απόφασης που διαχωρίζει καλύτερα τα παραδείγματα στις κλάσεις εξόδου, δηλαδή που ελαχιστοποιεί το σφάλμα ταξινόμησης. Για την παλινδρόμηση ο στόχος είναι να βρεθεί ο στενότερος σωλήνας που περιέχει όλα τα παραδείγματα, δηλαδή που ελαχιστοποιεί τη διαφορά μεταξύ της εξόδου του μοντέλου και της πραγματικής εξόδου.

Για την εργασία ταξινόμησης οι SVM υπολογίζουν αυτό που ονομάζεται διαχωριστής μέγιστου περιθωρίου. Αυτό είναι το επίπεδο που διαχωρίζει όλα τα παραδείγματα και έχει τη μεγαλύτερη δυνατή απόσταση από τα δεδομένα των κλάσεων. Αυτό έχει ως αποτέλεσμα, αντί να ελαχιστοποιεί το αναμενόμενο εμπειρικό σφάλμα (λανθασμένες ταξινομήσεις των δεδομένων) να ελαχιστοποιεί το αναμενόμενο σφάλμα γενίκευσης. Αυτό αντιστοιχεί στο αναμενόμενο σφάλμα των παραδειγμάτων που δεν έχουν δει ακόμη. Αυτό σημαίνει ότι, για να βρούμε το διαχωριστικό μεταξύ των κλάσεων, πρέπει να βρούμε τα πλησιέστερα παραδείγματα από τις διάφορες κλάσεις στα δεδομένα και να τοποθετήσουμε ένα επίπεδο έτσι ώστε να βρίσκεται στην ίδια απόσταση από αυτά, ορίζοντας με αυτόν τον τρόπο το όριο μεταξύ των κλάσεων. Αυτό το επίπεδο θα είναι δυνατό για προβλήματα που είναι γραμμικά διαχωρίσιμα, όπου ένα επίπεδο μπορεί να ορίσει το όριο. Μπορούμε να επεκτείνουμε το μοντέλο έτσι ώστε να επιτρέπει μη γραμμικά όρια ή που να επιτρέπει περιορισμένο αριθμό σφαλμάτων στην ταξινόμηση.

Θα ορίσουμε τον αλγόριθμο για την εκμάθηση αυτού του επιπέδου για την περίπτωση δύο κλάσεων, αλλά μπορεί να γενικευτεί για την περίπτωση πολλών κλάσεων. Για να υπολογίσουμε την απόσταση των παραδειγμάτων από ένα επίπεδο w μπορούμε να χρησιμοποιήσουμε την εξίσωση:

$$Y = \frac{w^T x_i + w_0}{\|w\|}$$

Διαιρούμε με τη νόρμα του διανύσματος w , ώστε η απόσταση μεταξύ του επιπέδου και των πλησιέστερων παραδειγμάτων να είναι +1 ή -1 ανάλογα με την πλευρά του επιπέδου στην οποία βρίσκεται το παράδειγμα. Αυτό καθιστά επίσης τη λύση του επιπέδου μοναδική. Αυτό καθιστά το μέγεθος του διαχωρισμού (περιθώριο) μεταξύ του επιπέδου και των παραδειγμάτων $2/\|w\|$. Έτσι, η μεγιστοποίηση του περιθωρίου είναι το ίδιο με την ελαχιστοποίηση της νόρμας των βαρών του επιπέδου.

Αυτό που πραγματικά διακρίνει τα παραδείγματα από διαφορετικές κατηγορίες είναι το πρόσημο της απόστασης από το επίπεδο. Αν αποδώσουμε στα παραδείγματα αριθμητικές ετικέτες (y) για τις κλάσεις, που αντιστοιχούν σε +1 και -1, έχουμε ότι ισχύει το όριο απόφασης για την ταξινόμηση των παραδειγμάτων:

$$y_i(w^T x_i + w_0) \geq 1, \forall i$$

Ο ορισμός αυτός επιτρέπει τον ορισμό του προβλήματος της εύρεσης του επιπέδου που μεγιστοποιεί την απόσταση μεταξύ των παραδειγμάτων των συνόρων ως πρόβλημα βελτιστοποίησης. Ο χειρισμός αυτού του προβλήματος οδηγεί σε μια εναλλακτική διατύπωση που ορίζει τον υπολογισμό αυτού του επιπέδου ως σταθμισμένο συνδυασμό των παραδειγμάτων με αυτόν τον τρόπο:

$$\arg \max_a \sum_{i=1}^N a_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N a_i y_i y_j x_j^T x_j, \text{ st } \sum_{i=1}^N a_i y_i = 0, a_i \geq 0 \forall i$$

Αυτή η βελτιστοποίηση αντιστοιχεί σε ένα πρόβλημα τετραγωνικού προγραμματισμού που μπορεί να επιλυθεί αποτελεσματικά. Για την πλειονότητα των βαρών (α) η τιμή τους θα είναι μηδέν. Εκείνα που έχουν τιμή μεγαλύτερη του μηδενός ονομάζονται διανύσματα υποστήριξης και είναι τα μόνα παραδείγματα που χρειάζονται για να οριστεί το επίπεδο που διαχωρίζει τις κλάσεις. Η εξίσωση του επιπέδου μπορεί να προκύψει ως το άθροισμα των παραδειγμάτων πολλαπλασιασμένο με αυτά τα βάρη.

Προκειμένου να αντιμετωπιστούν μη διαχωρίσιμες κλάσεις και να διατηρηθεί ένα γραμμικό επίπεδο που ορίζεται από τα χαρακτηριστικά εισόδου, μπορεί να προστεθεί ένας πρόσθετος όρος στο πρόβλημα βελτιστοποίησης. Σε αυτή την περίπτωση προσθέτουμε μια πρόσθετη μεταβλητή για κάθε παράδειγμα, η οποία επιτρέπει τη μείωση της απόστασης μεταξύ των παραδειγμάτων και του επιπέδου, ώστε να μπορεί να είναι μικρότερη από ένα. Αυτό ονομάζεται μαλακό περιθώριο και επιτρέπει ορισμένες λανθασμένες ταξινομήσεις. Το συνολικό επίπεδο σφάλματος που επιτρέπουμε μπορεί να προστεθεί ως υπερπαράμετρος πολλαπλασιάζοντας το άθροισμα των τιμών αυτών των μεταβλητών (το σφάλμα) με έναν παράγοντα που αυξάνει ή μειώνει την ανοχή.

Μέχρι τώρα χρησιμοποιήσαμε μόνο γραμμικούς συνδυασμούς των χαρακτηριστικών εισόδου για τον υπολογισμό του επιπέδου που διαχωρίζει τις κλάσεις. Όπως είδαμε στη γραμμική παλινδρόμηση, μπορούμε να συνδυάσουμε και να μετασχηματίσουμε μη γραμμικά αυτά τα χαρακτηριστικά, αλλά τα SVMs επιτρέπουν να το κάνουμε με πιο κομψό τρόπο. Η ιδέα είναι να μετατρέψουμε τα δεδομένα εισόδου σε έναν νέο χώρο με μεγαλύτερο αριθμό διαστάσεων, χρησιμοποιώντας μη γραμμικούς συνδυασμούς και μετασχηματισμούς. Ο στόχος είναι να έχουμε έναν νέο χώρο όπου τα παραδείγματα είναι γραμμικά διαχωρίσιμα. Αυτός ο νέος χώρος ονομάζεται χώρος χαρακτηριστικών. Ο ρητός υπολογισμός αυτού του μετασχηματισμού θα μπορούσε να είναι δαπανηρός, υποθέτοντας ότι ο αριθμός των νέων διαστάσεων που πρέπει να προστεθούν, ώστε να μπορέσουμε να διαχωρίσουμε τις κλάσεις, θα μπορούσε να είναι μεγάλος, ακόμη και άπειρος. Προκειμένου να απλοποιηθεί αυτός ο υπολογισμός, το κλειδί είναι να θεωρήσουμε ότι το πρόβλημα που βελτιστοποιείται βασίζεται στο εσωτερικό γινόμενο μεταξύ των παραδειγμάτων ($x_i^T x_j$) που αντιστοιχεί στον υπολογισμό μιας απόστασης μεταξύ των παραδειγμάτων στο χώρο που ορίζεται από τα χαρακτηριστικά. Οι αποστάσεις σε διαφορετικούς χώρους χαρακτηριστικών μπορούν να οριστούν με τη χρήση συναρτήσεων που ονομάζονται πυρήνες ($K(x,y)$). Ένας πυρήνας υπολογίζει ένα εσωτερικό γινόμενο για έναν δεδομένο χώρο εισόδου χωρίς να χρειάζεται να εκφράζονται ρητά όλες οι διαστάσεις, υπολογίζοντας με αυτόν τον τρόπο την απόσταση μεταξύ των παραδειγμάτων. Οι ιδιότητες που πρέπει να πληροί μια συνάρτηση για να είναι πυρήνας είναι ότι ο πίνακας που ορίζεται από τον υπολογισμό της στα παραδείγματα πρέπει να είναι συμμετρικός και θετικά ημιτελής. Όλα αυτά σημαίνουν ότι ο μη γραμμικός μετασχηματισμός των χαρακτηριστικών εισόδου $\Phi(x_i)$ και το γινόμενό τους μπορούν να αντικατασταθούν από:

$$\Phi(x_i)^T \Phi(x_j) = K(x_i, x_j)$$

Πολλές συναρτήσεις μπορούν να χρησιμοποιηθούν ως πυρήνες, αλλά οι πιο συνηθισμένοι είναι ο πολυωνυμικός πυρήνας $K(x,y) = (1+x^T y)^p$ όπου p είναι ο βαθμός του πολυωνύμου και ο γκαουσιανός πυρήνας γνωστός και ως πυρήνας ακτινικής βάσης $K(x,y) = \exp(-||x-y||^2/s^2)$ που ορίζει έναν σφαιρικό πυρήνα όπου s είναι μια υπερπαράμετρος που ορίζει την ακτίνα της σφαίρας.

Πολυστρωματικό perceptron

Τα perceptrons πολλαπλών στρωμάτων είναι μοντέλα διάκρισης που μπορούν να χρησιμοποιηθούν για ταξινόμηση και παλινδρόμηση. Τα μοντέλα αυτά περιλαμβάνονται στην περιοχή των Τεχνητών Νευρωνικών Δικτύων και μπορούν να θεωρηθούν το απλούστερο μοντέλο αυτού του τύπου. Τα νευρωνικά δίκτυα είναι εμπνευσμένα από τα κύτταρα του εγκεφάλου υπό την έννοια ότι αποτελούνται από μονάδες που εκτελούν απλούς υπολογισμούς και συνδέονται μεταξύ τους προκειμένου να μπορούν να αναπαραστήσουν σύνθετες λειτουργίες. Για ένα perceptron, μια μονάδα έχει συνδέσεις εισόδου με άλλες μονάδες. Οι τιμές που λαμβάνονται στις συνδέσεις εισόδου σταθμίζονται και συνδυάζονται γραμμικά. Η μονάδα έχει μια έξοδο που υπολογίζεται εφαρμόζοντας μια συνάρτηση στο γραμμικό συνδυασμό των εισόδων. Η συνάρτηση αυτή ονομάζεται συνάρτηση ενεργοποίησης. Στη συνέχεια, η έξοδος συνδέεται με άλλες μονάδες. Ο τύπος της συνάρτησης ενεργοποίησης που χρησιμοποιείται καθορίζει τι είδους συναρτήσεις μπορούν να μάθουν. Μπορούμε να ορίσουμε μια μονάδα ως μια συνάρτηση:

$$f(x) = g\left(b + \sum_{i=1}^p w_i x_i\right)$$

όπου g είναι η συνάρτηση ενεργοποίησης, w_i είναι τα βάρη του γραμμικού συνδυασμού και b είναι ο ανεξάρτητος όρος, γνωστός και ως μεροληψία. Η χρήση της ταυτότητας ως συνάρτηση ενεργοποίησης το καθιστά γραμμικό μοντέλο, ισοδύναμο με άλλα μοντέλα που χρησιμοποιούν ένα επίπεδο για να ορίσουν τα όρια μεταξύ των κλάσεων για την ταξινόμηση ή ισοδύναμο με τη γραμμική παλινδρόμηση κατά την επίλυση της παλινδρόμησης.

Ένα πολυστρωματικό perceptron είναι οργανωμένο σε επίπεδα. Υπάρχει ένα στρώμα εισόδου που λαμβάνει τις τιμές από τα χαρακτηριστικά των δεδομένων, ένα σύνολο κρυφών στρωμάτων που μετασχηματίζουν διαδοχικά τις τιμές που λαμβάνονται από το αμέσως προηγούμενο στρώμα και ένα στρώμα εξόδου που επιστρέφει το αποτέλεσμα του υπολογισμού. Κάθε στρώμα αποτελείται από έναν αριθμό μονάδων και κάθε μία συνδέεται με όλες τις μονάδες του επόμενου στρώματος. Όλα αυτά τα στοιχεία σχηματίζουν ένα άκυκλο κατευθυνόμενο γράφημα, ένα νευρωνικό δίκτυο, το οποίο καθορίζει τον τρόπο με τον οποίο εκτελούνται οι υπολογισμοί προκειμένου να προκύψει μια έξοδος από τις τιμές εισόδου. Ένα μοντέλο που έχει μόνο ένα στρώμα εισόδου και ένα στρώμα εξόδου ονομάζεται perceptron, αν υπάρχουν κρυφά στρώματα τότε πρόκειται για perceptron πολλαπλών στρωμάτων.

Αυτό το μοντέλο είναι πολύ ευέλικτο και διαθέτει πολλές υπερπαραμέτρους που πρέπει να ρυθμιστούν, δηλαδή τον αριθμό των στρωμάτων, τον αριθμό των μονάδων ανά στρώμα και τις συναρτήσεις ενεργοποίησης των μονάδων. Ο αριθμός των μονάδων του στρώματος εξόδου εξαρτάται από την εργασία. Για ταξινόμηση μπορεί να χρησιμοποιηθεί μια μονάδα για κάθε κλάση, επίσης μια μονάδα είναι αρκετή εάν πρόκειται για δυαδικό πρόβλημα ταξινόμησης που θεωρεί το πρόσημο της εξόδου ως κλάση. Για παλινδρόμηση αρκεί μια μονάδα αν πρόκειται για μονοδιάστατη έξοδο.

Τα βάρη που χρησιμοποιούνται στους γραμμικούς συνδυασμούς των μονάδων είναι οι παράμετροι που μαθαίνονται κατά την εκπαίδευση του μοντέλου. Μπορούμε να σκεφτούμε ένα νευρωνικό δίκτυο ως μια σύνθεση συναρτήσεων που χρησιμοποιούμε για να εκτελέσουμε μη γραμμική παλινδρόμηση. Χρησιμοποιούμε μια συνάρτηση απώλειας (για παράδειγμα το μέσο

τετραγωνικό σφάλμα) προκειμένου να μετρήσουμε την απόδοση του δικτύου (το σφάλμα). Θέλουμε να ελαχιστοποιήσουμε το σφάλμα δεδομένων των βαρών του δικτύου. Αυτό το πρόβλημα βελτιστοποίησης μπορεί να επιλυθεί με αλγόριθμους βαθμωτής καθόδου, εκτελώντας μια αναζήτηση στο χώρο των βαρών προκειμένου να ελαχιστοποιηθεί το σφάλμα. Εάν η συνάρτηση ενεργοποίησης που χρησιμοποιείται στις μονάδες του δικτύου και η συνάρτηση απώλειας είναι διαφοροποιήσιμες, μπορούμε να υπολογίσουμε αναλυτικά την κατεύθυνση που μεταβάλλει τα βάρη ώστε να ελαχιστοποιηθεί το σφάλμα. Το πρόβλημα αυτό είναι απλό για το perceptron ενός στρώματος, δεδομένου ότι η είσοδος συνδέεται απευθείας με την έξοδο. Για το πολυεπίπεδο perceptron, κάθε στρώμα εκτελεί έναν υπολογισμό που τροφοδοτείται στο επόμενο στρώμα, οπότε κάθε στρώμα έχει το δικό του σφάλμα που πρέπει να υπολογιστεί προκειμένου να προκύψει το σφάλμα των συνδυασμένων στρωμάτων.

Ο αλγόριθμος που χρησιμοποιείται για την εκμάθηση των βαρών του πολυεπίπεδου perceptron ονομάζεται αλγόριθμος backpropagation. Ο αλγόριθμος αυτός λειτουργεί σε δύο βήματα. Το πρώτο βήμα προωθεί τις τιμές των παραδειγμάτων μέσω του δικτύου για να ληφθούν οι τιμές εξόδου. Το δεύτερο βήμα αντιστρέφει τον υπολογισμό, το σφάλμα διαδίδεται προς τα πίσω και κάθε επίπεδο λαμβάνει το σφάλμα που υπολογίζεται από το επόμενο επίπεδο. Τα βάρη των μονάδων υπολογίζονται εκ νέου έτσι ώστε το σφάλμα να μειωθεί χρησιμοποιώντας την κλίση της συνάρτησης που υπολογίζεται στο επίπεδο. Η αλλαγή αυτή υπολογίζεται ως μια μικρή αύξηση των τιμών των βαρών προς την κατεύθυνση της κλίσης της συνάρτησης σφάλματος. Αυτή η αύξηση είναι γνωστή ως ρυθμός μάθησης και επηρεάζει το πόσο γρήγορα προσαρμόζονται τα βάρη στο σφάλμα. Αυτή είναι μια άλλη υπερπαραμέτρος του μοντέλου. Ένας μεγάλος ρυθμός μάθησης μπορεί να σημαίνει κακό αποτέλεσμα, ένας πολύ χαμηλός ρυθμός μάθησης μπορεί να σημαίνει αύξηση του υπολογιστικού χρόνου που απαιτείται για την επίτευξη των βέλτιστων βαρών. Η διαδικασία αυτή επαναλαμβάνεται αρκετές φορές μέχρι τη σύγκλιση.

Αυτή η διαδικασία μπορεί να υπολογιστεί σε απευθείας σύνδεση, ώστε κάθε παράδειγμα να αλλάζει τα βάρη του δικτύου, ή σε παρτίδες, ώστε πολλά παραδείγματα να διαδίδονται μέσω του δικτύου και τα σφάλματα να υπολογίζονται κατά μέσο όρο. Στην περίπτωση που έχουμε μεγάλο αριθμό παραδειγμάτων και πολλά στρώματα και μονάδες, ο υπολογισμός αυτός είναι πολύ δαπανηρός, οπότε αντί να χρησιμοποιούνται όλα τα παραδείγματα για τον υπολογισμό σε κάθε επανάληψη, χρησιμοποιούνται διαδοχικά τυχαίες παρτίδες για τον υπολογισμό των ενημερώσεων αυτό είναι γνωστό ως στοχαστική κάθοδος κλίσης.

Η βελτιστοποίηση των πολυστρωματικών perceptrons παρουσιάζει πολλά προβλήματα. Η συνάρτηση απώλειας που υπολογίζεται είναι εξαιρετικά μη γραμμική, οπότε υπάρχουν πολλές τοπικές βέλτιστες τιμές. Τα τελικά βάρη εξαρτώνται σε μεγάλο βαθμό από τα αρχικά βάρη που ξεκινούν την αναζήτηση και από τη σειρά επεξεργασίας των παραδειγμάτων. Ο μεγάλος αριθμός υπερπαραμέτρων αυτού του μοντέλου καθιστά επίσης εύκολη την υπερπροσαρμογή των δεδομένων. Μια συνήθης στρατηγική που εφαρμόζεται για τη μείωση αυτού του προβλήματος είναι η διακοπή της βελτιστοποίησης πριν από την εύρεση του βέλτιστου, αυτό ονομάζεται πρόωπη διακοπή. Η στρατηγική αυτή βοηθάει στην επίτευξη αποτελεσμάτων με καλύτερη γενίκευση.

Μοντέλο συνόλου

Δεδομένου ότι δεν υπάρχει κανένας αλγόριθμος μηχανικής μάθησης που να είναι ανώτερος για όλα τα προβλήματα, μια δυνατότητα είναι ο συνδυασμός των αποφάσεων πολλών μοντέλων για

τη βελτίωση των επιμέρους αποτελεσμάτων. Αυτός είναι ο στόχος της μάθησης συνόλου. Η ιδέα είναι ότι συνδυάζοντας μοντέλα που έχουν μεγάλη μεταβλητότητα θα έχουμε μοντέλα που θα αποδίδουν καλύτερα για ορισμένα παραδείγματα από ό,τι για άλλα, αλλά ο συνδυασμός θα αντισταθμίζει τα επιμέρους σφάλματα.

Υπάρχουν πολλοί τρόποι για την ομαδοποίηση μοντέλων, από τη μέση τιμή των αποτελεσμάτων ή τη σταθμισμένη ψηφοφορία έως τη χρήση ενός άλλου μοντέλου μηχανικής μάθησης για να μάθει ποιος είναι ο καλύτερος συνδυασμός για ένα σύνολο μοντέλων. Από όλες αυτές τις δυνατότητες το σύνολο των δέντρων απόφασης έχει αποδειχθεί πολύ επιτυχημένο στην πράξη. Αυτό προέρχεται από την υψηλή μεταβλητότητα που έχει αυτό το μοντέλο όταν εκπαιδεύεται με διαφορετικά δείγματα δεδομένων. Το ensemble των δέντρων απόφασης προσθέτει μια νέα υπερπαραμέτρο σε αυτό το μοντέλο, τον αριθμό των μοντέλων που θα χρησιμοποιηθούν για τη δημιουργία του ensemble.

Υπάρχουν διάφορες στρατηγικές που χρησιμοποιούνται συνήθως για την ομαδοποίηση των δέντρων απόφασης:

- **Boosting trees**, πρόκειται για μια επαναληπτική στρατηγική που λαμβάνει μια ακολουθία δέντρων απόφασης και ένα σύνολο βαρών για τις αποφάσεις τους. Κάθε δέντρο υπολογίζει το σφάλμα και σταθμίζει εκ νέου τα παραδείγματα προκειμένου να μάθει το επόμενο δέντρο. Τα βάρη είναι ανάλογα με τη συμβολή των παραδειγμάτων στο σφάλμα του δέντρου, ώστε το επόμενο να μπορεί να εξειδικευτεί στα παραδείγματα που έχουν ταξινομηθεί λανθασμένα.
- **Δέντρα Gradient boosting**, πρόκειται επίσης για μια επαναληπτική στρατηγική που αποτελεί γενίκευση του boosting. Για αυτό το μοντέλο όλες οι εργασίες επιλύονται ως εργασίες παλινδρόμησης χρησιμοποιώντας μια διαφορίσιμη συνάρτηση απώλειας. Κάθε μοντέλο μαθαίνει το υπολειπόμενο σφάλμα του προηγούμενου μοντέλου, οπότε η συνάρτηση εξόδου μαθαίνεται σταδιακά.
- **Bagging trees**, η στρατηγική αυτή δημιουργεί πολλά τυχαία δείγματα με αντικατάσταση των δεδομένων και εκπαιδεύει ένα μοντέλο δέντρου για κάθε δείγμα. Αυτό προάγει τη μεταβλητότητα των μοντέλων. Οι αποφάσεις λαμβάνονται με ψηφοφορία πλειοψηφίας ή με μέσο όρο.
- **Random forest**, αυτή η στρατηγική βασίζεται στη χρήση μόνο ενός υποσυνόλου των χαρακτηριστικών εισόδου για τον υπολογισμό της διάσπασης στους κόμβους. Οι αποφάσεις λαμβάνονται με ψηφοφορία πλειοψηφίας ή με μέσο όρο μοντέλου.

Επιλέγοντας έναν αλγόριθμο

Όλοι οι αλγόριθμοι αυτού του μαθήματος είναι πιο δαπανηροί υπολογιστικά από εκείνους του προηγούμενου μαθήματος, αλλά συνήθως μπορούν να επιτύχουν καλύτερα αποτελέσματα επειδή μπορούν να ανακαλύψουν και να μοντελοποιήσουν ευκολότερα μη γραμμικές εξαρτήσεις στα δεδομένα. Χρησιμοποιούνται όταν η προς επίλυση εργασία είναι πολύ σύνθετη και δεν μπορεί να επιλυθεί με ακρίβεια από απλούστερα μοντέλα. Οι μηχανές διανυσμάτων υποστήριξης είναι οι πιο εύκολες στη χρήση, δεδομένου ότι έχουν λιγότερες υπερπαραμέτρους, και μπορούν να θεωρηθούν ως βελτίωση των γραμμικών μοντέλων δεδομένου ότι εισάγουν αυτόματα μη γραμμικούς μετασχηματισμούς των χαρακτηριστικών. Τα Multi-Layer Perceptrons είναι πιο δύσκολο να ρυθμιστούν, με μεγάλο αριθμό υπερπαραμέτρων, και όχι πολύ διαισθητικά. Το κύριο πλεονέκτημά τους είναι ότι μπορούν να επιτύχουν πιο ακριβή μοντέλα αν συντονιστούν σωστά, αλλά αυτό απαιτεί σκληρή δουλειά. Τα σύνολα μοντέλων, ιδίως εκείνα που χρησιμοποιούν δέντρα

απόφασης/ παλινδρόμησης, είναι πολύ ανταγωνιστικά στους περισσότερους τομείς. Έχουν περισσότερες παραμέτρους προς ρύθμιση, αλλά είναι πιο διαισθητικά και ευκολότερα κατανοητά. Το κύριο πλεονέκτημα είναι ότι είναι σε θέση να δώσουν τουλάχιστον κάποιες πληροφορίες σχετικά με τη συνάφεια των χαρακτηριστικών στις αποφάσεις του μοντέλου μπροστά στην προσέγγιση του μαύρου κουτιού των άλλων δύο αλγορίθμων.

Προφανώς, η τελική επιλογή του καλύτερου αλγορίθμου για ένα πρόβλημα εξαρτάται από την απόδοση του μοντέλου που προκύπτει από τα δεδομένα. Είναι δύσκολο να ληφθούν αποφάσεις χωρίς την εφαρμογή των αλγορίθμων και τη ρύθμιση των παραμέτρων τους ώστε να επιτευχθεί το καλύτερο αποτέλεσμα.

Περιπτώσεις χρήσης

ΤΙΜΕΣ ΣΠΙΤΙΩΝ ΣΤΗΝ ΚΑΛΙΦΟΡΝΙΑ

Θα συνεχίσουμε την εξερεύνηση του συνόλου δεδομένων για τις τιμές των κατοικιών στην Καλιφόρνια. Πρόκειται για ένα πρόβλημα παλινδρόμησης, δεδομένου ότι η σχετική έξοδος είναι συνεχής. Μπορούμε να χρησιμοποιήσουμε για το έργο αυτό μηχανή διανυσμάτων υποστήριξης παλινδρόμησης, πολυστρωματικό perceptron παλινδρόμησης και τυχαίο δάσος παλινδρόμησης. Και πάλι, όπως αναφέραμε στις περιπτώσεις χρήσης του προηγούμενου μαθήματος, για την εκπαίδευση ενός μοντέλου μηχανικής μάθησης πρέπει να προχωρήσουμε συστηματικά. Πρέπει να χωρίσουμε τα δεδομένα σε σύνολο εκπαίδευσης και σε σύνολο δοκιμής και να εκτιμήσουμε το σφάλμα χρησιμοποιώντας μια διαδικασία επικύρωσης. Ο πιο συνηθισμένος τρόπος είναι η χρήση της διασταυρούμενης επικύρωσης. Στην περίπτωση χρήσης αλγορίθμων που έχουν υπερπαραμέτρους πρέπει να προσαρμοστούν και αυτές με τη χρήση μιας διαδικασίας επικύρωσης. Η τελική απόφαση για το ποιο μοντέλο ή ποιοι αλγόριθμοι είναι οι καλύτεροι θα πρέπει να σχετίζεται με το μέτρο απόδοσης που έχει επιλεγεί για την προς επίλυση εργασία (ταξινόμηση/ παλινδρόμηση), φροντίζοντας ώστε η απόδοση για τα δεδομένα εκπαίδευσης να είναι παρόμοια με εκείνη από τα δεδομένα δοκιμής, ώστε να μην υπάρχει υπερπροσαρμογή.

ΒΗΜΑ 1: Θα επαναλάβουμε την προεπεξεργασία που κάναμε στο προηγούμενο μάθημα, ώστε να μπορέσουμε να συγκρίνουμε τα αποτελέσματα:

```
import sklearn.datasets as dsets
import numpy as np
hframe= dsets.fetch_california_housing(as_frame=True)['frame']
hdata = hframe.loc[:, ['MedInc', 'HouseAge', 'Population','Latitude','Longitude']].copy()
hdata['logAveBedrms'] = np.log(hframe['AveBedrms'])
hdata['logAveOccup'] = np.log(hframe['AveOccup'])
hdata['logAveRooms'] = np.log(hframe['AveRooms'])
```

Πρέπει να χωρίσουμε τα δεδομένα σε ένα σύνολο εκπαίδευσης και ένα σύνολο δοκιμής. Σε αυτή την περίπτωση, το μέγεθος του συνόλου δεδομένων καθιστά υπολογιστικά δαπανηρή τη χρήση ενός μεγάλου δείγματος για εκπαίδευση, επειδή ορισμένα από τα μοντέλα που χρησιμοποιούμε

δεν είναι επεκτάσιμα και χρειάζονται πολύ χρόνο για την εκπαίδευση. Για το λόγο αυτό θα αλλάξουμε τον προεπιλεγμένο διαχωρισμό σε 50%-50%, μπορούμε να ορίσουμε τη γεννήτρια τυχαίων αριθμών που χρησιμοποιείται για τη δημιουργία των δειγμάτων ώστε τα αποτελέσματα να είναι αναπαραγώγιμα:

```
X_train, X_test, y_train, y_test = train_test_split(hdata, htarget, test_size=0.5, random_state=0)
```

Κανονικοποιούμε επίσης τα δεδομένα ώστε να έχουν μέσο όρο μηδέν και μοναδιαία τυπική απόκλιση. Για να κανονικοποιήσουμε τα δεδομένα, η συνάρτηση κανονικοποίησης πρέπει να προσαρμοστεί στα δεδομένα εκπαίδευσης και στη συνέχεια η κανονικοποίηση πρέπει να εφαρμοστεί στα δεδομένα εκπαίδευσης και δοκιμής.

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

ΒΗΜΑ 2: Ξεκινάμε με τη μηχανή διανυσμάτων υποστήριξης για παλινδρόμηση. Αυτό το μοντέλο έχει αρκετές υπερπαραμέτρους, θα συντονίσουμε μόνο το βάρος που τιμωρεί το σφάλμα του περιθωρίου (C), αλλά θα μπορούσαμε επίσης να πειραματιστούμε με διαφορετικούς πυρήνες για να μετασχηματίσουμε μη γραμμικά τα δεδομένα. Ο προεπιλεγμένος πυρήνας που χρησιμοποιείται είναι ο γκαουσιανός πυρήνας που είναι σε θέση να προσεγγίσει οποιαδήποτε συνάρτηση. Σε αυτή την περίπτωση θα εκτελέσουμε μια αναζήτηση πλέγματος δημιουργώντας όλα τα μοντέλα που αντιστοιχούν σε έναν κατάλογο τιμών για την παράμετρο κόστους. Θα εκτιμήσουμε το σφάλμα του καλύτερου μοντέλου χρησιμοποιώντας διασταυρούμενη επικύρωση. Το καλύτερο μοντέλο αποθηκεύεται από τη συνάρτηση αναζήτησης πλέγματος με διασταυρωμένη επικύρωση. Από προεπιλογή το scikit learn χρησιμοποιεί το R2 (συντελεστής προσδιορισμού) ως μέτρο απόδοσης, υποδεικνύουμε στη συνάρτηση διασταυρούμενης επικύρωσης το μέτρο που θέλουμε να χρησιμοποιήσουμε, στην προκειμένη περίπτωση το μέσο τετραγωνικό σφάλμα.

```
from sklearn.model_selection import cross_validate
from sklearn.model_selection import GridSearchCV
from sklearn.svm import SVR
param = {'C':[0.1,1, 10, 100]}
svr = SVR()
svrreg = GridSearchCV(svr,param,cv=10, scoring=make_scorer(mean_squared_error,
greater_is_better=False))
svrreg.fit(X_train, y_train)
```

Το χαρακτηριστικό `best_estimator_` περιέχει το μοντέλο με την καλύτερη απόδοση, εφαρμόζουμε τώρα διασταυρωμένη επικύρωση για να υπολογίσουμε την απόδοσή του:

```
cv_results = cross_validate(svrreg.best_estimator_, X_train, y_train, cv=10,
scoring=make_scorer(mean_squared_error, greater_is_better=False))
np.mean(cv_results['test_score'])
-0.3003301672855393
```

Μπορούμε να υπολογίσουμε το σφάλμα για τα δεδομένα δοκιμής εκπαιδεύοντας ένα SVM παλινδρόμησης με όλα τα δεδομένα εκπαίδευσης:

```
svrreg.best_estimator_.fit(X_train,y_train)
mean_squared_error(y_test,svrreg.best_estimator_.predict(X_test))
0.2961355753962943
```

Βλέπουμε ότι και τα δύο σφάλματα είναι παρόμοια.

ΒΗΜΑ 3: Τώρα θα εκπαιδεύσουμε ένα πολυεπίπεδο perceptron για παλινδρόμηση. Σε αυτή την περίπτωση έχουμε επίσης πολλές υπερπαραμέτρους προς ρύθμιση, η εξερεύνησή τους θα έπαιρνε πολύ χρόνο. Θα περιοριστούμε σε ένα perceptron με ένα κρυφό στρώμα δοκιμάζοντας διαφορετικό αριθμό μονάδων για το στρώμα αυτό και τον αριθμό των επαναλήψεων που εκτελεί η οπισθοδιάδοση για να ελέγξουμε αν ο πρόωρος τερματισμός της βελτιστοποίησης οδηγεί σε καλύτερη γενίκευση. Χρησιμοποιούμε διασταυρωμένη επικυρωμένη αναζήτηση πλέγματος με μικρό αριθμό δυνατοτήτων λόγω του μεγέθους του συνόλου δεδομένων:

```
from sklearn.neural_network import MLPRegressor
param = {'hidden_layer_sizes':[(100), (250)], 'max_iter':[500, 1000]}
mlp = MLPRegressor()
mlpreg = GridSearchCV(mlp,param,cv=10, scoring=make_scorer(mean_squared_error,
greater_is_better=False))
mlpreg.fit(X_train, y_train)
```

Εφαρμόζουμε τώρα τη διασταυρούμενη επικύρωση για να υπολογίσουμε την απόδοσή της:

```
cv_results = cross_validate(mlpreg.best_estimator_, X_train, y_train, cv=10,
scoring=make_scorer(mean_squared_error, greater_is_better=False))
np.mean(cv_results['test_score'])
-0.2855341914582281
```

Μπορούμε να υπολογίσουμε το σφάλμα για τα δεδομένα δοκιμής που εκπαιδεύονται με όλα τα δεδομένα εκπαίδευσης:

```
mlpreg.best_estimator_.fit(X_train,y_train)
mean_squared_error(y_test,mlpreg.best_estimator_.predict(X_test))
0.2807308880038008
```

Τα αποτελέσματα για την εκπαίδευση και τη δοκιμή είναι συνεπή και ελαφρώς καλύτερα από ό,τι για το SVM.

ΒΗΜΑ 4: Τέλος, χρησιμοποιούμε ένα σύνολο μοντέλων. Υπάρχουν πολλά σύνολα που μπορούμε να χρησιμοποιήσουμε, θα περιορίσουμε τα πειράματα στο τυχαίο δάσος παλινδρόμησης, αλλά θα μπορούσαν να εφαρμοστούν και άλλα σύνολα. Σε αυτή την περίπτωση θα διερευνήσουμε μόνο τον αριθμό των μοντέλων για το ensemble ως υπερπαραμέτρο με τη χρήση cross validated grid search:

```
from sklearn.ensemble import RandomForestRegressor
param = {'n_estimators':[10,50, 100, 200]}
rf = RandomForestRegressor()
rftree = GridSearchCV(rf,param,cv=10, scoring=make_scorer(mean_squared_error,
greater_is_better=False))
rftree.fit(X_train, y_train)
```

Υπολογίζουμε το διασταυρωμένο επικυρωμένο σφάλμα του καλύτερου μοντέλου:

```
cv_results = cross_validate(rftree.best_estimator_, X_train, y_train, cv=10,
scoring=make_scorer(mean_squared_error, greater_is_better=False))
np.mean(cv_results['test_score'])
-0.27048298764933293
```

Τώρα υπολογίζουμε το σφάλμα με τα δεδομένα δοκιμής:

```
rftree.best_estimator_.fit(X_train,y_train)
0.2783036884495705
```

Το σφάλμα είναι επίσης συνεπές για την εκπαίδευση και τη δοκιμή.

Αυτό το τελευταίο μοντέλο είναι ελαφρώς καλύτερο από τα δύο πρώτα. Μπορούμε επίσης να παρατηρήσουμε ότι χρησιμοποιώντας αυτά τα πιο σύνθετα μοντέλα έχουμε βελτιωθεί σε σχέση με τα γραμμικά μοντέλα που χρησιμοποιήσαμε στο τελευταίο μάθημα.

ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 2: ΟΙΝΟΠΑΡΑΓΩΓΟΙ

Σε αυτή την περίπτωση θα συνεχίσουμε να εργαζόμαστε με το σύνολο δεδομένων για το κρασί από το προηγούμενο μάθημα. Όπως εξηγήσαμε, πρόκειται για μια εργασία ταξινόμησης που συνίσταται στην ταξινόμηση διαφορετικών κρασιών σε τρεις κλάσεις που αντιστοιχούν σε τρεις διαφορετικούς καλλιεργητές στην ίδια περιοχή της Ιταλίας. Κάθε κρασί περιγράφεται από τα αποτελέσματα ενός συνόλου χημικών αναλύσεων:

- Αλκοόλ
- Μηλικό οξύ
- Τέφρα
- Αλκαλικότητα της τέφρας
- Μαγνήσιο
- Ολικές φαινόλες
- Φλαβονοειδή
- Μη φλαβονοειδείς φαινόλες
- Προανθοκυανίνες
- Ένταση χρώματος
- Απόχρωση
- OD280/OD315 αραιωμένων οίνων
- Προλίνη

Όλα αυτά είναι συνεχή χαρακτηριστικά.

Και πάλι, όπως αναφέραμε στις περιπτώσεις χρήσης του προηγούμενου μαθήματος, για την εκπαίδευση ενός μοντέλου μηχανικής μάθησης πρέπει να προχωρήσουμε συστηματικά. Πρέπει να χωρίσουμε τα δεδομένα σε σύνολο εκπαίδευσης και σε σύνολο δοκιμής και να εκτιμήσουμε το σφάλμα χρησιμοποιώντας μια διαδικασία επικύρωσης. Ο πιο συνηθισμένος τρόπος είναι η χρήση της διασταυρούμενης επικύρωσης. Στην περίπτωση χρήσης αλγορίθμων που έχουν υπερπαραμέτρους πρέπει να προσαρμοστούν και αυτές με τη χρήση μιας διαδικασίας επικύρωσης. Η τελική απόφαση για το ποιο μοντέλο ή ποιοι αλγόριθμοι είναι οι καλύτεροι θα πρέπει να σχετίζεται με το μέτρο απόδοσης που έχει επιλεγεί για την προς επίλυση εργασία (ταξινόμηση/ παλινδρόμηση), φροντίζοντας ώστε η απόδοση για τα δεδομένα εκπαίδευσης να είναι παρόμοια με εκείνη από τα δεδομένα δοκιμής, ώστε να μην υπάρχει υπερπροσαρμογή.

Δεδομένου ότι πρόκειται για εργασία ταξινόμησης μπορούμε να χρησιμοποιήσουμε μια μηχανή διανυσμάτων υποστήριξης, έναν πολυστρωματικό perceptron και έναν ταξινομητή τυχαίου δάσους. Θα ξεκινήσουμε με τη φόρτωση των δεδομένων. Σε αυτή την περίπτωση θα το κάνουμε ως δύο πίνακες numpy, έναν για τα δεδομένα εισόδου και έναν για τα δεδομένα εξόδου:

```
import sklearn.datasets as dsets  
X,y = dsets.load_wine(return_X_y=True)
```

Πρέπει να χωρίσουμε τα δεδομένα σε εκπαίδευση και δοκιμή, θα χρησιμοποιήσουμε το προεπιλεγμένο 75%-25%. Πρόκειται για ένα μικρό σύνολο δεδομένων, ώστε να μην έχουμε υπολογιστικά προβλήματα. Θα κανονικοποιήσουμε επίσης τις τιμές των χαρακτηριστικών,

μπορούμε να ορίσουμε τη γεννήτρια τυχαίων αριθμών που χρησιμοποιείται για τη δημιουργία των δειγμάτων ώστε τα αποτελέσματα να είναι αναπαραγώγιμα:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0)
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

Βήμα 1: Θα ξεκινήσουμε με μια μηχανή διανυσμάτων υποστήριξης για ταξινόμηση διερευνώντας ως υπερπαραμέτρους την παράμετρο κόστους και τον πυρήνα που χρησιμοποιείται για τον μη γραμμικό μετασχηματισμό των δεδομένων με τον γκαουσιανό πυρήνα και τον πολυωνυμικό πυρήνα χρησιμοποιώντας τον προεπιλεγμένο βαθμό για το πολυώνυμο, δηλαδή 3.

```
from sklearn.model_selection import cross_validate
from sklearn.model_selection import GridSearchCV
from sklearn.svm import SVC
param = {'C':[0.1,1, 10, 100, 1000], 'kernel':['rbf', 'poly']}
svc = SVC()
svccl = GridSearchCV(svc,param,cv=10)
svccl.fit(X_train, y_train)
```

Αυτό θα μας δώσει τον καλύτερο συνδυασμό των παραμέτρων hyper σύμφωνα με την εκτιμώμενη ακρίβεια από τη διασταυρούμενη επικύρωση. Μπορούμε να εκπαιδεύσουμε αυτό το μοντέλο και να υπολογίσουμε την εκτιμώμενη ακρίβεια:

```
cv_results = cross_validate(svccl.best_estimator_, X_train, y_train, cv=10)
np.mean(cv_results['test_score'])
0.9780219780219781
Και τέλος, η ακρίβεια με τα δεδομένα δοκιμής:
svccl.best_estimator_.fit(X_train,y_train)
accuracy_score(y_test,svccl.best_estimator_.predict(X_test))
1.0
```

Βήμα 2: Το δεύτερο μοντέλο θα είναι ένας πολυστρωματικός ταξινομητής perceptron. Για την περίπτωση αυτή, θα επιλέξουμε ένα perceptron με ένα μόνο κρυφό στρώμα. Θα διερευνήσουμε ως υπερπαραμέτρους τον αριθμό των μονάδων στο κρυφό στρώμα, τη συνάρτηση ενεργοποίησης και τον αριθμό των επαναλήψεων που χρησιμοποιούνται στη διαδικασία βελτιστοποίησης. Για τις συναρτήσεις ενεργοποίησης θα χρησιμοποιήσουμε τη συνάρτηση relu και τη συνάρτηση logistic.

```
from sklearn.neural_network import MLPClassifier

param = {'hidden_layer_sizes':[(10), (100), (200)], 'activation':['relu', 'logistic'], 'max_iter':[500, 1000]}

mlp = MLPClassifier()
mlpcl = GridSearchCV(mlp,param,cv=10)
mlpcl.fit(X_train, y_train)
```

Αυτό θα μας δώσει τον καλύτερο συνδυασμό υπερπαραμέτρων σύμφωνα με την εκτιμώμενη ακρίβεια από τη διασταυρούμενη επικύρωση. Μπορούμε να εκπαιδεύσουμε αυτό το μοντέλο και να υπολογίσουμε την εκτιμώμενη ακρίβεια:

```
cv_results = cross_validate(mlpcl.best_estimator_, X_train, y_train, cv=10)
np.mean(cv_results['test_score'])
0.9637362637362639
```

Και τέλος, η ακρίβεια με τα δεδομένα δοκιμής:

```
mlpcl.best_estimator_.fit(X_train,y_train)
accuracy_score(y_test,mlpcl.predict(X_test))
0.9777777777777777
```

Βήμα 3: Τέλος, θα χρησιμοποιήσουμε ένα σύνολο μοντέλων. Υπάρχουν πολλά σύνολα που μπορούμε να χρησιμοποιήσουμε, θα περιορίσουμε τα πειράματα στο τυχαίο δάσος, αλλά θα μπορούσαν να εφαρμοστούν και άλλα σύνολα. Σε αυτή την περίπτωση θα διερευνήσουμε μόνο τον αριθμό των μοντέλων για το ensemble ως υπερπαραμέτρο με τη χρήση cross validated grid search:

```
from sklearn.ensemble import RandomForestClassifier

param = {'n_estimators':[10,50, 100, 200]}

rf = RandomForestClassifier()
rftree = GridSearchCV(rf,param,cv=10)
rftree.fit(X_train, y_train)
```

Αυτό θα μας δώσει τον καλύτερο συνδυασμό υπερπαραμέτρων σύμφωνα με την εκτιμώμενη ακρίβεια από τη διασταυρούμενη επικύρωση. Μπορούμε να εκπαιδεύσουμε αυτό το μοντέλο και να υπολογίσουμε το εκτιμώμενο σφάλμα:

```
cv_results = cross_validate(rftree.best_estimator_, X_train, y_train, cv=10)
np.mean(cv_results['test_score'])
0.9780219780219781
```

Και τέλος, η ακρίβεια με τα δεδομένα δοκιμής:

```
rftree.best_estimator_.fit(X_train,y_train)
accuracy_score(y_test,rftree.best_estimator_.predict(X_test))
0.9555555555555556
```

Συγκρίνοντας τα τρία μοντέλα βλέπουμε ότι οι διασταυρωμένες επικυρωμένες ακρίβειες είναι πολύ παρόμοιες. Για την ακρίβεια του συνόλου δοκιμής μπορούμε να δούμε ότι η μηχανή διανυσμάτων υποστήριξης έχει τέλειο σκορ. Η ρύθμιση περισσότερων υπερπαραμέτρων πιθανόν να οδηγήσει σε τέλεια μοντέλα για όλους τους αλγορίθμους. Μπορούμε επίσης να παρατηρήσουμε ότι χρησιμοποιώντας αυτά τα πιο σύνθετα μοντέλα έχουμε βελτιωθεί σε σχέση με τα γραμμικά μοντέλα που χρησιμοποιήσαμε στο προηγούμενο μάθημα.

Αναφορές

- Introduction to machine learning Alpaydin, E, The MIT Press, 2014. ISBN: 9780262028189 (chapter 11, 13, 17)
- The elements of statistical learning: data mining, inference, and prediction Hastie, T.; Tibshirani, R.; Friedman, J, Springer, 2009. ISBN: 9780387848570 (chapter 10, 11, 12)
- Pattern recognition and machine learning Bishop, C.M, Springer, 2006. ISBN: 0387310738 (chapter 5, 6, 14)

Μάθημα 2.6: Μάθηση χωρίς επίβλεψη

Θέματα προς συζήτηση

- Εισαγωγή
- Ιεραρχική ομαδοποίηση
- Κ-Μέσοι
- Μοντέλα Γκαουσιανών μιγμάτων
- Ποιοτική Αξιολόγηση της συσταδοποίησης

Σημειώσεις διάλεξης

Εισαγωγή

Οι αλγόριθμοι μη επιβλεπόμενης μηχανικής μάθησης χρησιμοποιούνται για την ανάλυση και τη διερεύνηση δεδομένων όταν δεν υπάρχει σαφής στόχος ή εργασία. Ο σκοπός είναι να εντοπιστούν μοτίβα που μπορεί να είναι χρήσιμα ή που εξηγούν κάποια από τη συμπεριφορά που παρατηρείται στα δεδομένα. Υπάρχουν ορισμένα καθήκοντα που μπορούν να επιλυθούν χωρίς επίβλεψη, αλλά αυτό που είναι πιο συνηθισμένο είναι η ανακάλυψη ομάδων στα δεδομένα που παρουσιάζουν κάποια συνοχή και έχουν μια ξεχωριστή συμπεριφορά. Αυτή η εργασία είναι γνωστή ως ομαδοποίηση.

Η ομαδοποίηση μπορεί να περιγραφεί ως ένα κακοθεωρημένο πρόβλημα, με την έννοια ότι δεν υπάρχει μοναδική λύση, επειδή διαφορετικές καταταμίσεις ενός συνόλου δεδομένων μπορεί να είναι σωστές ανάλογα με τους στόχους ή την προοπτική της εκάστοτε εργασίας. Η εργασία ομαδοποίησης μπορεί να οριστεί ως μια διαδικασία που, χρησιμοποιώντας τις εγγενείς ιδιότητες ενός συνόλου δεδομένων, αποκαλύπτει ένα σύνολο καταταμίσεων που αντιπροσωπεύει την εγγενή δομή του. Πρόκειται, επομένως, για μια εργασία που βασίζεται στα πρότυπα που προκύπτουν από τις τιμές των χαρακτηριστικών που περιγράφουν το σύνολο δεδομένων. Ο στόχος της εργασίας θα είναι είτε η απόκτηση μιας αναπαράστασης που περιγράφει ένα μη επισημασμένο σύνολο δεδομένων (σύνοψη) είτε η ανακάλυψη εννοιών στο εσωτερικό των δεδομένων (κατανόηση) που αναπαρίστανται από τις συστάδες.

Ένα βασικό στοιχείο της ομαδοποίησης είναι ο τρόπος με τον οποίο συγκρίνονται τα παραδείγματα μεταξύ τους. Οι αλγόριθμοι υποθέτουν ότι τα δεδομένα ορίζονται από ένα σύνολο συντεταγμένων που μπορούν να χρησιμοποιηθούν για τη μέτρηση της ομοιότητας ή της απόστασης μεταξύ τους. Αυτό θα χρησιμοποιηθεί για να αποφασιστεί ποια παραδείγματα ομαδοποιούνται επειδή μοιράζονται κοινά στοιχεία και ποιες διαφορετικές ομάδες υπάρχουν στα δεδομένα.

Υπάρχουν πολλοί αλγόριθμοι ομαδοποίησης, θα επικεντρωθούμε στην ιεραρχική ομαδοποίηση, τον k-means και τα μοντέλα μίξης του Γκαουσιανού. Αυτοί είναι οι συνήθεις αλγόριθμοι που χρησιμοποιούνται σε πρακτικές εφαρμογές λόγω της γενικότητάς τους και μπορούν να βρεθούν

στις περισσότερες βιβλιοθήκες μηχανικής μάθησης και πακέτα λογισμικού που χρησιμοποιούνται από τη βιομηχανία.

- **Ιεραρχική ομαδοποίηση**

Η ιεραρχική ομαδοποίηση αποκτά μια κατάτμηση ενός συνόλου δεδομένων δημιουργώντας μια ακολουθία εμφωλευμένων κατατμήσεων που συνδέουν όλα τα παραδείγματα ενός συνόλου δεδομένων. Αυτές οι κατατμήσεις δημιουργούνται σύμφωνα με την ομοιότητα ή την απόσταση μεταξύ των παραδειγμάτων. Οι ιεραρχικές μέθοδοι χρησιμοποιούν δύο αλγοριθμικές στρατηγικές, τη διαιρετική και τη συσσωρευτική, αλλά συνήθως προτιμάται η δεύτερη, επειδή είναι πιο αποτελεσματική. Οι συσσωρευτικές στρατηγικές συγχωνεύουν επαναληπτικά το πιο συγγενικό ζεύγος κατατμήσεων σύμφωνα με ένα μέτρο ομοιότητας/απόστασης μέχρι να υπάρχει μόνο μία κατάτμηση.

Αυτές οι μέθοδοι βασίζονται σε μια συνάρτηση απόστασης/ομοιότητας που συγκρίνει κατατμήσεις και παραδείγματα. Αρχικά, οι τιμές αυτές για κάθε ζεύγος παραδειγμάτων αποθηκεύονται σε έναν πίνακα τον οποίο ο αλγόριθμος ενημερώνει κατά τη διάρκεια της διαδικασίας ομαδοποίησης. Αυτό καθιστά το υπολογιστικό κόστος αυτών των αλγορίθμων τουλάχιστον $O(n^2)$ σε χρόνο και χώρο ακόμη και πριν από την έναρξη της διαδικασίας κατάτμησης.

Η στρατηγική που χρησιμοποιείται από ορισμένους αλγορίθμους θεωρεί αυτόν τον πίνακα ως ένα γράφημα που δημιουργείται προσθέτοντας σε κάθε επανάληψη νέες ακμές με αύξουσα/κατιούσα σειρά μήκους. Στην περίπτωση αυτή, ένα κριτήριο καθορίζει πότε η προσθήκη μιας νέας ακμής οδηγεί σε μια νέα συστάδα. Αυτοί είναι αλγόριθμοι βασισμένοι σε γράφο. Άλλοι αλγόριθμοι μειώνουν τον πίνακα απόστασης/ομοιότητας σε κάθε επανάληψη συγχωνεύοντας δύο ομάδες, διαγράφοντας αυτές τις ομάδες από τον πίνακα και στη συνέχεια προσθέτοντας τη νέα συγχωνευμένη ομάδα. Πρόκειται για αλγεβρικούς αλγορίθμους που βασίζονται στην άλγεβρα πινάκων.

Οι αλγόριθμοι που βασίζονται σε γραφήματα υποθέτουν ότι τα παραδείγματα σχηματίζουν ένα πλήρως συνδεδεμένο γράφημα, το μήκος των ακμών ορίζεται από τη συνάρτηση ομοιότητας/απόστασης και υπάρχει ένα κριτήριο συνδεσιμότητας που χρησιμοποιείται για να καθορίσει πότε δύο παραδείγματα ανήκουν στην ίδια ομάδα. Τα πιο συνηθισμένα είναι, το κριτήριο απλής σύνδεσης, που ορίζει ότι δύο παραδείγματα ανήκουν στην ίδια ομάδα όταν προστίθεται μια νέα ακμή στο γράφο, εάν συνδέει παραδείγματα από ασύζευκτες ομάδες, και το κριτήριο πλήρους σύνδεσης που θεωρεί ότι δύο παραδείγματα ανήκουν στην ίδια ομάδα όταν αυτά και όλα τα παραδείγματα που συνδέονται με αυτά σχηματίζουν μια κλίκα στο γράφο όταν προστίθεται η νέα ακμή. Και τα δύο κριτήρια έχουν διαφορετική προκατάληψη, η απλή σύνδεση επιτρέπει την ανακάλυψη επιμήκων συστάδων και η πλήρης σύνδεση προτιμά τις σφαιρικές συστάδες.

Οι αλγόριθμοι που βασίζονται στην άλγεβρα πινάκων λειτουργούν με παρόμοιο τρόπο. Η διαδικασία ξεκινά με τον πλήρη πίνακα αποστάσεων που υπολογίζεται από τα παραδείγματα με δεδομένη μια συνάρτηση απόστασης/ομοιότητας. Σε κάθε βήμα της διαδικασίας, βρίσκει πρώτα τα δύο πιο παρόμοια παραδείγματα/συστάδες για να συγχωνευθούν σε μια νέα ομάδα. Τα αρχικά παραδείγματα/συστάδες διαγράφονται και η νέα ομάδα προστίθεται στον πίνακα. Στη συνέχεια, υπολογίζονται οι αποστάσεις από τη συγχωνευμένη ομάδα προς τις υπόλοιπες ομάδες. Για το σκοπό αυτό μπορούν να επιλεγούν διαφορετικά κριτήρια.

Οι αποστάσεις αυτής της νέας ομάδας από τις άλλες ομάδες είναι ένας συνδυασμός των αποστάσεων από τις δύο συγχωνευμένες ομάδες. Δημοφιλείς επιλογές για αυτόν τον συνδυασμό είναι η μέγιστη, η ελάχιστη και η μέση τιμή αυτών των αποστάσεων. Το μέγεθος της ομάδας ή η διακύμανσή της μπορεί επίσης να χρησιμοποιηθεί για τη στάθμιση του συνδυασμού. Ορισμένες από αυτές τις επιλογές αναπαράγουν τα κριτήρια που χρησιμοποιούνται από αλγορίθμους βασισμένους σε γράφους, για παράδειγμα η απλή σύνδεση ισοδυναμεί με αντικατάσταση των αποστάσεων με την ελάχιστη απόσταση μεταξύ των παραδειγμάτων από τη νέα ομάδα και των παραδειγμάτων στις άλλες ομάδες, η επιλογή της μεγαλύτερης απόστασης ισοδυναμεί με πλήρη σύνδεση. Η χρήση του μέσου όρου ως κριτήριο οδηγεί στη μέση σύνδεση και αν η διακύμανση των ομάδων χρησιμοποιείται επίσης ως βάρη, αυτό ονομάζεται σύνδεση Ward.

Το κύριο μειονέκτημα αυτών των αλγορίθμων είναι το υπολογιστικό τους κόστος. Ο πίνακας αποστάσεων πρέπει να αποθηκευτεί και αυτό κλιμακώνεται τετραγωνικά με τον αριθμό των παραδειγμάτων. Επίσης, το υπολογιστικό κόστος αυτών των αλγορίθμων είναι κυβικό με τον αριθμό των παραδειγμάτων στη γενική περίπτωση και αυτό μπορεί να μειωθεί σε ορισμένες ειδικές περιπτώσεις σε τετραγωνικό.

Το αποτέλεσμα αυτών των αλγορίθμων είναι αυτό που ονομάζεται δένδρογραμμα. Πρόκειται για ένα δυαδικό δέντρο που δείχνει πώς συγχωνεύτηκαν τα παραδείγματα και οι ομάδες και περιέχει πληροφορίες σχετικά με την απόσταση των συγχωνεύσεων. Αυτό επιτρέπει την οπτικοποίηση των σχέσεων μεταξύ παραδειγμάτων και ομάδων, αλλά δεν είναι πολύ πρακτικό για μεγάλα σύνολα δεδομένων. Προκειμένου να προκύψει μια κατάτμηση των δεδομένων, το δένδρογραμμα πρέπει να κοπεί σε ένα συγκεκριμένο επίπεδο που αντιστοιχεί στις ομάδες των δεδομένων με τη χρήση κάποιων κριτηρίων, όπως για παράδειγμα η σχετική διαφορά μεταξύ της απόστασης των συγχωνεύσεων στην κορυφή της ιεραρχίας ή ένα μέτρο ποιότητας της ομαδοποίησης.

- ***K-Means***

Ο αλγόριθμος K-means είναι ένας αλγόριθμος ομαδοποίησης με όρους, το αποτέλεσμα του αλγορίθμου είναι άμεσα ένα σύνολο ομάδων σε αντίθεση με τους ιεραρχικούς αλγορίθμους. Πρόκειται επίσης για έναν αλγόριθμο που βασίζεται σε πρωτότυπα, αυτό σημαίνει ότι χρησιμοποιείται μια περιγραφή των ομάδων προκειμένου να υπολογιστεί η κατάτμηση και είναι επίσης ένα αποτέλεσμα του αλγορίθμου εκτός από την κατάτμηση. Πρόκειται για έναν πολύ δημοφιλή αλγόριθμο που χρησιμοποιείται πρώτος όταν πρέπει να επιλυθεί μια εργασία ομαδοποίησης. Το πλεονέκτημά του είναι ότι είναι υπολογιστικά φθηνός σε σύγκριση με άλλες εναλλακτικές λύσεις και έχει λογική απόδοση.

Αυτός ο αλγόριθμος υποθέτει ότι οι συστάδες ορίζονται από το κεντροειδές τους (το πρωτότυπο) και έχουν σφαιρικό σχήμα. Η προσαρμογή αυτών των σφαιρών γίνεται με την ελαχιστοποίηση των αποστάσεων των παραδειγμάτων από αυτά τα κεντροειδή. Τα παραδείγματα αντιστοιχίζονται μόνο σε μία συστάδα- αυτό είναι γνωστό ως σκληρή ομαδοποίηση ή σκληρή διαμέριση. Μπορούν να χρησιμοποιηθούν διαφορετικά κριτήρια βελτιστοποίησης για την απόκτηση της διαμέρισης, αλλά το πιο συνηθισμένο είναι η ελαχιστοποίηση του αθροίσματος της τετραγωνικής ευκλείδειας απόστασης των παραδειγμάτων που αντιστοιχίζονται σε μια συστάδα και του κεντροειδούς της συστάδας. Το πρόβλημα μπορεί να τυποποιηθεί ως εξής:

$$\text{minimize}_c \sum_{c_j} \sum_{x_i \in C} \|x_i - \mu_j\|_2^2$$

Όπου μ_i είναι το πρωτότυπο της συστάδας (το κεντροειδές). Αυτό το πρόβλημα ελαχιστοποίησης επιλύεται επαναληπτικά χρησιμοποιώντας έναν αλγόριθμο αναρρίχησης λόφου, έχοντας ως παράμετρο το k . Αυτό σημαίνει ότι πρέπει να αποφασίσουμε εκ των προτέρων πόσες συστάδες θέλουμε από τα δεδομένα.

Ο αλγόριθμος για τον υπολογισμό της διαμέρισης k -means έχει ως εξής:

1. Επιλέξτε k παραδείγματα ως αρχικά πρωτότυπα των k διαμερισμάτων
2. Επαναλάβετε όσο υπάρχουν αλλαγές (ή μετά από έναν αριθμό επαναλήψεων)
 - a. Επαναπροσδιορίστε τα παραδείγματα δεδομένων των πρωτοτύπων
 - b. Επαναυπολογισμός των πρωτοτύπων δεδομένων των τρεχουσών αναθέσεων παραδειγμάτων

Το κόστος του υπολογισμού εξαρτάται από τον αριθμό των επαναλήψεων που απαιτούνται μέχρι τη σύγκλιση, όπου κάθε επανάληψη υπολογίζει $N \cdot k$ αποστάσεις.

Ένα από τα μειονεκτήματα αυτού του αλγορίθμου είναι ότι η τελική λύση εξαρτάται από τα αρχικά πρωτότυπα, δεδομένου ότι θα υπάρχουν πολλά τοπικά ελάχιστα. Μια συνήθης διαδικασία είναι η εκτέλεση του αλγορίθμου αρκετές φορές και η διατήρηση της καλύτερης λύσης. Μια εναλλακτική λύση είναι η χρήση μιας πιο προηγμένης αρχικοποίησης. Για παράδειγμα, συνήθως εφαρμόζεται η στρατηγική K -means++. Αυτή συνίσταται στην επαναληπτική επιλογή των αρχικών πρωτοτύπων με τη χρήση μιας τυχαιοποιημένης διαδικασίας, όπου κάθε νέο πρωτότυπο επιλέγεται με πιθανότητα ανάλογη της τετραγωνικής απόστασης από το πλησιέστερο πρωτότυπο που έχει ήδη επιλεγεί.

Ένα άλλο μειονέκτημα του αλγορίθμου είναι ότι είναι ευαίσθητος σε συστάδες στα δεδομένα με διαφορετικά μεγέθη και πυκνότητες, επιστρέφοντας φτωχά αποτελέσματα σε αυτές τις περιπτώσεις. Αυτό οφείλεται στο γεγονός ότι ο αλγόριθμος μεροληπτεί προς την αναζήτηση σφαιρικών συστάδων που είναι συμπαγείς και διαχωρισμένες.

Το πιο πρακτικό ζήτημα αυτού του αλγορίθμου είναι ο τρόπος επιλογής του αριθμού των συστάδων. Αυτό είναι ένα πρόβλημα για πολλούς άλλους αλγορίθμους συσταδοποίησης που χρειάζονται τον αριθμό των διαμερισμάτων και το οποίο επιλύεται διερευνώντας ένα εύρος πιθανών τιμών και χρησιμοποιώντας ένα κριτήριο ποιότητας συσταδοποίησης. Θα μιλήσουμε γι' αυτά στο τέλος του μαθήματος.

• Μοντέλα μίξης Gauss

Τα μοντέλα μίξης Gauss είναι ένας αλγόριθμος ομαδοποίησης βασισμένος στο πρωτότυπο, ο οποίος υποθέτει ότι τα δεδομένα μπορούν να προσαρμοστούν σε ένα μείγμα κατανομών πιθανότητας. Το σχήμα των συστάδων εξαρτάται από τη συγκεκριμένη κατανομή πιθανότητας που χρησιμοποιείται. Μια συνήθης επιλογή είναι η γκαουσιανή κατανομή. Στην περίπτωση αυτή, ανάλογα με την επιλογή των παραμέτρων που χρησιμοποιούνται για τη μοντελοποίηση, το σχήμα των συστάδων που προσαρμόζονται αντιστοιχεί από αυθαίρετα προσανατολισμένα ελλειψοειδή έως σφαιρικά σχήματα. Η προσαρμογή του μοντέλου στα δεδομένα μπορεί να εκφραστεί γενικά ως εξής

$$P(x | \theta) = \sum_{i=1}^K w_i P(x | \theta_i)$$

Με K τον αριθμό των συστάδων, θ_i τις παραμέτρους των κατανομών και με το άθροισμα όλων των w_i ίσο με 1. Αυτό το μοντέλο προσαρμόζεται συνήθως χρησιμοποιώντας τον αλγόριθμο Expectation-Maximization (EM), αναθέτοντας σε κάθε παράδειγμα μια πιθανότητα για κάθε συστάδα, πράγμα που σημαίνει ότι λαμβάνουμε μια μαλακή ομαδοποίηση των δεδομένων, όλα τα παραδείγματα ανήκουν σε όλες τις συστάδες με έναν βαθμό.

Για την εκμάθηση αυτού του μοντέλου, ο στόχος είναι να εκτιμηθούν οι παράμετροι των κατανομών που περιγράφουν κάθε κλάση. Ο αλγόριθμος Expectation Maximization μεγιστοποιεί την πιθανότητα της κατανομής ως προς το σύνολο δεδομένων επαναληπτικά, ξεκινώντας με μια αρχική εκτίμηση των παραμέτρων. Εκτελεί επανειλημμένα δύο βήματα μέχρι τη σύγκλιση, ακολουθώντας τον αλγόριθμο:

- **Προσδοκία:** μια συνάρτηση που αποδίδει σε κάθε παράδειγμα έναν βαθμό συμμετοχής σε όλες τις K κατανομές πιθανότητας, ο οποίος υπολογίζεται χρησιμοποιώντας την τρέχουσα εκτίμηση των παραμέτρων του μοντέλου.
- **Μεγιστοποίηση:** οι παράμετροι των κατανομών επανεκτιμώνται χρησιμοποιώντας ως βάρη τα μέλη που προέκυψαν από το προηγούμενο βήμα για τη μεγιστοποίηση των μελών των παραδειγμάτων.

Το αποτέλεσμα του αλγορίθμου είναι ένα σύνολο πιθανοτήτων για κάθε παράδειγμα και οι εκτιμώμενες παράμετροι για τις κατανομές που μπορούν να χρησιμοποιηθούν ως πρωτότυπα για τις συστάδες.

Κάθε συγκεκριμένο μοντέλο έχει ένα σύνολο παραμέτρων ανάλογα με την επιλεγμένη κατανομή πιθανότητας, οι οποίες πρέπει να εκτιμηθούν με τη χρήση του αλγορίθμου EM. Για την περίπτωση των γκαουσιανών κατανομών, οι παράμετροι αντιστοιχούν στη μέση τιμή και στην τυπική απόκλιση. Ανάλογα με τις υποθέσεις σχετικά με τα χαρακτηριστικά των δεδομένων μπορούμε να προσαρμόσουμε διαφορετικά μοντέλα με διαφορετικό υπολογιστικό κόστος. Αν υποθέσουμε ότι όλα τα χαρακτηριστικά είναι ανεξάρτητα και έχουν την ίδια τυπική απόκλιση, προσαρμόζουμε ένα μοντέλο που αποτελείται από k σφαίρες όπως στον k -means, που αποτελεί μια ειδική περίπτωση αυτού του αλγορίθμου. Αν υποθέσουμε ότι κάθε χαρακτηριστικό έχει τη δική του τυπική απόκλιση αλλά δεν υπάρχουν αλληλεπιδράσεις (δεν υπάρχουν διασταυρούμενες συσχετίσεις), αυξάνουμε την ευελιξία του μοντέλου για να προσαρμόσουμε k ελλειψοειδή ευθυγραμμισμένα με τους άξονες συντεταγμένων. Η προσθήκη των συνδιακυμάνσεων μεταξύ των χαρακτηριστικών, υποθέτοντας όλες τις πιθανές αλληλεπιδράσεις, επιτρέπει την περιστροφή των ελλειψοειδών σε αυθαίρετες κατευθύνσεις για καλύτερη προσαρμογή στα δεδομένα. Κάθε παραδοχή αυξάνει το υπολογιστικό κόστος της εκτίμησης.

• Αξιολόγηση της ποιότητας της ομαδοποίησης

Δεδομένου ότι δεν έχουμε ένα πραγματικό αποτέλεσμα που μπορούμε να χρησιμοποιήσουμε για να το συγκρίνουμε με τα αποτελέσματα των αλγορίθμων μάθησης χωρίς επίβλεψη, χρειαζόμαστε κάποια εναλλακτικά μέτρα που μπορούν να μας δώσουν μια ιδέα για το πόσο καλή είναι η διαμέριση που έχουμε επιτύχει. Πρέπει επίσης να προσδιορίσουμε τον καλύτερο αριθμό ομάδων που χρειάζεται ο αλγόριθμος ως παράμετρο για την κατάτμηση των δεδομένων.

Υπάρχουν πολλά μέτρα ποιότητας για την ομαδοποίηση που επικεντρώνονται στην κύρια προκατάληψη αυτών των αλγορίθμων. Βασικά, μετράνε πόσο συμπαγείς είναι οι συστάδες (πόσο παρόμοια είναι τα παραδείγματα μέσα στη συστάδα) και πόσο διαχωρισμένες είναι οι συστάδες μεταξύ τους (πόσο ανόμοια είναι τα παραδείγματα σε διαφορετικές συστάδες). Δεν υπάρχει ένα αποδεκτό μέτρο που να λειτουργεί σε κάθε περίπτωση, αλλά μερικά από τα πιο συχνά χρησιμοποιούμενα περιλαμβάνουν τις βαθμολογίες Calinski-Harabasz, Davis-Bouldin και Silhouette. Αυτά τα μέτρα μπορούν να χρησιμοποιηθούν για τη σύγκριση συστάδων και επίσης για την αξιολόγηση του βέλτιστου αριθμού συστάδων. Αυτά τα μέτρα ποιότητας μπορούν να απεικονιστούν για διαφορετικό αριθμό συστάδων για να εντοπιστεί ένα βέλτιστο ή ένα σημείο όπου η προσθήκη περισσότερων συστάδων δεν αυξάνει σημαντικά την ποιότητα.

- **Επιλέγοντας έναν αλγόριθμο**

Και οι τρεις αλγόριθμοι επιτελούν το ίδιο έργο, την κατάτμηση των δεδομένων. Δεδομένου ότι ο στόχος είναι η εξερεύνηση των δεδομένων, δεν υπάρχει κανένα συγκεκριμένο κριτήριο για την επιλογή ενός αλγορίθμου έναντι άλλων πέραν της γνώσης και της διαίσθησης σχετικά με τη συμπεριφορά των δεδομένων. Η ιεραρχική συσταδοποίηση αλλάζει την υπόθεση με τα κριτήρια συνδεσιμότητας που χρησιμοποιούνται, από σφαιρικές συστάδες σε πιο κακοσχηματισμένες συστάδες. Το κύριο μειονέκτημα αυτών των αλγορίθμων είναι το υπολογιστικό τους κόστος, οπότε είναι δύσκολο να χρησιμοποιηθούν για μεγάλα σύνολα δεδομένων. Ο αλγόριθμος K-means είναι η πιο δημοφιλής επιλογή δεδομένου ότι είναι εύκολος στη χρήση, είναι υπολογιστικά φθηνός και υποθέτει μόνο ότι οι ομάδες στα δεδομένα διαχωρίζονται αρκετά, ώστε να μπορούν να ανακαλυφθούν δεδομένου του σωστού αριθμού συστάδων. Αυτή είναι συνήθως η πρώτη επιλογή κατά τη διερεύνηση ενός συνόλου δεδομένων. Τα μοντέλα μίξης Gauss κάνουν συγκεκριμένες υποθέσεις σχετικά με τη συμπεριφορά των δεδομένων και όταν αυτές είναι σωστές θα επιστρέψουν καλύτερα μοντέλα, αλλά με υψηλότερο υπολογιστικό κόστος.

Προφανώς, η τελική επιλογή του καλύτερου αλγορίθμου για ένα πρόβλημα εξαρτάται από τα μέτρα απόδοσης που λαμβάνονται από τις κατατμήσεις των δεδομένων. Είναι δύσκολο να ληφθούν αποφάσεις χωρίς την εφαρμογή των αλγορίθμων και τη ρύθμιση των παραμέτρων τους ώστε να επιτευχθεί το καλύτερο αποτέλεσμα.

Περιπτώσεις χρήσης

ΟΙΝΟΠΑΡΑΓΩΓΟΙ

Σε αυτή την περίπτωση θα συνεχίσουμε να εργαζόμαστε με το σύνολο δεδομένων για το κρασί από το προηγούμενο μάθημα. Όπως εξηγήσαμε, πρόκειται για μια εργασία ταξινόμησης που συνίσταται στην ταξινόμηση διαφορετικών κρασιών σε τρεις κλάσεις που αντιστοιχούν σε τρεις διαφορετικούς καλλιεργητές στην ίδια περιοχή της Ιταλίας. Κάθε κρασί περιγράφεται από τα αποτελέσματα ενός συνόλου χημικών αναλύσεων:

- Αλκοόλ
- Μηλικό οξύ
- Τέφρα
- Αλκαλικότητα της τέφρας

- Μαγνήσιο
- Ολικές φαινόλες
- Φλαβονοειδή
- Μη φλαβονοειδείς φαινόλες
- Προανθοκυανίνες
- Ένταση χρώματος
- Απόχρωση
- OD280/OD315 αραιωμένων οίνων
- Προλίνη

Όλα αυτά είναι συνεχή χαρακτηριστικά.

Οι μη επιβλεπόμενες μέθοδοι μπορούν επίσης να προσεγγιστούν συστηματικά. Όλοι οι αλγόριθμοι που παρουσιάζονται στο μάθημα επιλύουν το ίδιο έργο, την κατάτμηση ενός συνόλου δεδομένων. Δεν υπάρχει σωστός ή λανθασμένος τρόπος απόκτησης μιας διαμέρισης και τα μέτρα ποιότητας που εξηγήσαμε προϋποθέτουν μια συγκεκριμένη θεώρηση μιας καλής διαμέρισης, συμπαγείς και διαχωρισμένες ομάδες. Αυτά τα μέτρα είναι ο τρόπος αξιολόγησης της ποιότητας που επιτυγχάνεται από τις μεθόδους που παρουσιάζονται, δεδομένου ότι αυτό είναι που επιδιώκουν. Τα μέτρα αυτά θα πρέπει να αποτελούν τον οδηγό για τη ρύθμιση των παραμέτρων των αλγορίθμων, των μεθόδων αυτών και τη βάση για την επιλογή του καλύτερου διαμερίσματος. Η επικύρωση των αποτελεσμάτων θα πρέπει να περιλαμβάνει, όταν είναι δυνατόν, την οπτικοποίηση των τελικών διαμερισμάτων με τη χρήση αλγορίθμων οπτικοποίησης όπως αυτοί που παρουσιάζονται στο μάθημα 3.

Δεδομένου ότι πρόκειται για μια εργασία ταξινόμησης, γνωρίζουμε ότι υπάρχουν τουλάχιστον τρεις διαφορετικές ομάδες στα δεδομένα, μπορούμε να το ελέγξουμε αυτό χρησιμοποιώντας διάφορους αλγορίθμους ομαδοποίησης και αξιολογώντας τον αριθμό των ομάδων που ταιριάζουν καλύτερα στα δεδομένα.

ΒΗΜΑ 1: Συνεχίζουμε να φορτώνουμε και να κανονικοποιούμε τα δεδομένα. Σε αυτή την περίπτωση θα χρησιμοποιήσουμε όλα τα δεδομένα και δεν χρειάζονται οι ετικέτες ταξινόμησης:

```
import sklearn.datasets as dsets
```

```
X, _ = dsets.load_wine(return_X_y=True)
```

```
scaler = StandardScaler()
```

```
X = scaler.fit_transform(X)
```

ΒΗΜΑ 2: Μπορούμε να ξεκινήσουμε με την ιεραρχική συσταδοποίηση, υπάρχουν διάφορα κριτήρια που μπορούν να χρησιμοποιηθούν για αυτόν τον αλγόριθμο, θα δοκιμάσουμε τα κριτήρια της πλήρους και της διασύνδεσης των θαλάμων και θα ελέγξουμε χρησιμοποιώντας το σκορ Calinski-Harabasz ποιος αριθμός συστάδων επιστρέφει το καλύτερο σκορ. Αυτή η βαθμολογία μεγιστοποιείται από τον βέλτιστο αριθμό συστάδων, μπορούμε να ελέγξουμε ένα εύρος τιμών από 2 έως 6:

```
from sklearn.cluster import AgglomerativeClustering
```

```
from sklearn.metrics import silhouette_score, calinski_harabasz_score
```

```
HCH = {'ward':[], 'complete':[]}  
for nc in range(2,7):  
    aggcl = AgglomerativeClustering(n_clusters=nc,linkage='ward')  
    labh = aggcl.fit_predict(X)  
    HCH['ward'].append(calinski_harabasz_score(X, labh))  
    aggcl = AgglomerativeClustering(n_clusters=nc,linkage='complete')  
    labh = aggcl.fit_predict(X)  
    HCH['complete'].append(calinski_harabasz_score(X, labh))  
for nc, w, s in zip(range(2,7), HCH['ward'], HCH['complete']):  
    print(f'Nclusters = {nc} - Ward: {w} - Complete: {s}')  
Nclusters = 2 - Ward: 65.36083820586111 - Complete: 36.31083356649205  
Nclusters = 3 - Ward: 67.6474675044098 - Complete: 48.98983018628942  
Nclusters = 4 - Ward: 51.46414629882825 - Complete: 44.720175786324646  
Nclusters = 5 - Ward: 43.67927204722388 - Πλήρης: 42.766683884872386  
Nclusters = 6 - Ward: 39.12896379154367 - Πλήρης: 36.3851782385074  
Βλέπουμε ότι το κριτήριο σύνδεσης Ward επιστρέφει καλύτερες τιμές και ότι η βαθμολογία  
μεγιστοποιείται για 3 συστάδες.
```

ΒΗΜΑ 3: Μπορούμε τώρα να χρησιμοποιήσουμε την ίδια διαδικασία για τον αλγόριθμο k-means:

```
from sklearn.cluster import Kmeans  
KCH = {'kmeans':[]}  
for nc in range(2,7):  
    km = KMeans(n_clusters=nc)  
    labk = km.fit_predict(X)  
    KCH['kmeans'].append(calinski_harabasz_score(X, labk))  
for nc, k in zip(range(2,7), KCH['kmeans']):  
    print(f'Nclusters = {nc} - K-means: {k}')  
Nclusters = 2 - K-means: 69.48646422404195  
Nclusters = 3 - K-means: 70.9400080031512  
Nclusters = 4 - K-means: 55.211159837292456  
Nclusters = 5 - K-means: 47.58076963201465  
Nclusters = 6 - K-means: 41.762114363881686  
Υπάρχει επίσης συμφωνία για το ότι τρεις είναι ο καλύτερος αριθμός συστάδων.
```

ΒΗΜΑ 4: Για τα Gaussian Mixture Models πρέπει να αποφασίσουμε τη σχέση μεταξύ των χαρακτηριστικών, θα δοκιμάσουμε να υποθέσουμε ότι όλες οι μεταβλητές είναι ανεξάρτητες με τη

δική τους διακύμανση και ότι όλες οι μεταβλητές σχετίζονται μεταξύ τους, οπότε θα εκτιμηθεί ένας πλήρης πίνακας συνδιακύμανσης:

```
GCH = {'diag':[], 'full':[]}
```

```
for nc in range(2,7):
```

```
    gmm = GaussianMixture(n_components=nc, covariance_type='full')
```

```
    labg = gmm.fit_predict(X)
```

```
    GCH['full'].append(calinski_harabasz_score(X, labg))
```

```
    gmm = GaussianMixture(n_components=nc, covariance_type='diag')
```

```
    labg = gmm.fit_predict(X)
```

```
    GCH['diag'].append(calinski_harabasz_score(X, labg))
```

```
for nc, f, d in zip(range(2,7), GCH['full'], GCH['diag']):
```

```
    print(f'Nclusters = {nc} - Full: {f} - Diag: {d}')
```

```
Nclusters = 2 - Full: 68.40347775923414 - Diag: 67.86952858824861
```

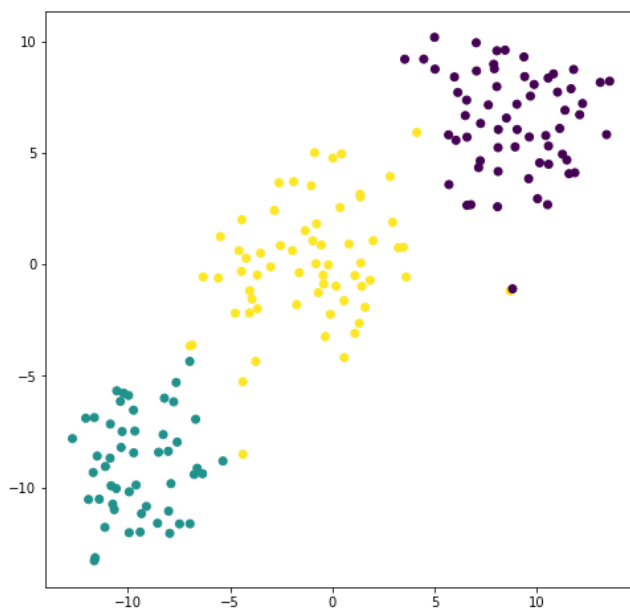
```
Nclusters = 3 - Full: 70.71682878884306 - Diag: 69.21965987528539
```

```
Nclusters = 4 - Full: 52.08910657146111 - Diag: 52.87514202419738
```

```
Nclusters = 5 - Full: 46.33856945827704 - Diag: 44.25939973425076
```

```
Nclusters = 6 - Full: 39.04167891992955 - Diag: 36.452035126270516
```

Για την περίπτωση αυτή, η διαφορά μεταξύ της εκτίμησης μόνο των αποκλίσεων και της πλήρους συνδιακύμανσης είναι πολύ μικρή και επίσης τρεις είναι ο καλύτερος αριθμός συστάδων για αυτόν τον αλγόριθμο. Από τα αποτελέσματα των τριών αλγορίθμων, φαίνεται ότι η διαμέριση που προκύπτει από τον k-means με τρεις συστάδες είναι η καλύτερη με μικρή διαφορά.



```
plt.show()
```

ΒΗΜΑ 5: Μπορούμε να απεικονίσουμε την κατάτμηση χρησιμοποιώντας έναν αλγόριθμο μείωσης διαστάσεων, για παράδειγμα τον t-SNE, και να σχεδιάσουμε τις αναθέσεις για να ελέγξουμε αν πρόκειται για μια συνεκτική κατάτμηση:

```
from sklearn.manifold import TSNE
```

```
tsne = TSNE(n_components=2, init='pca')
```

```
tsnmatrix = tsne.fit_transform(X)
```

```
km = KMeans(n_clusters=3)
```

```
labk = km.fit_predict(X)
```

```
fig = plt.figure(figsize=(8,8))
```

```
plt.scatter(tsnmatrix[:, 0], tsnmatrix[:, 1], c=labk)
```


Μπορούμε να δούμε ότι η προβολή δείχνει τρεις ομάδες παραδειγμάτων και ότι οι αναθέσεις σε συστάδες είναι πολύ συνεκτικές με αυτές τις τρεις ομάδες.

Αναφορές

- Introduction to machine learning Alpaydin, E, The MIT Press, 2014. ISBN: 9780262028189 (chapter 7)
- The elements of statistical learning: data mining, inference, and prediction Hastie, T.; Tibshirani, R.; Friedman, J, Springer, 2009. ISBN: 9780387848570 (chapter 14)
- Pattern recognition and machine learning Bishop, C.M, Springer, 2006. ISBN: 0387310738 (chapter 9)

Ενότητα 3: Νευρωνικά δίκτυα και βαθιά μάθηση για την όραση

Μαθημα 3.1: Εγκέφαλος & Νευρωνικά δίκτυα

Θέματα προς συζήτηση

- Σειριακά υπολογιστικά και βιολογικά νευρικά συστήματα
- Στοιχεία τεχνητών νευρωνικών δικτύων και η λειτουργία τους
- Επισκόπηση των κοινών αρχιτεκτονικών

Σημειώσεις διάλεξης

Σειριακοί υπολογιστές. Όταν κάποιος μιλάει για σημερινούς υπολογιστές, αναφέρεται σε μηχανές που κατασκευάστηκαν με βάση την αρχιτεκτονική υλικού που επινόησε ο Von Neumann γύρω στο 1945. Αυτός ο τύπος αρχιτεκτονικής εξελίχθηκε με την πάροδο του χρόνου, αλλά πάντα αποτελείται από:

- Υπολογιστική μονάδα ή επεξεργαστής (CPU) που εκτελεί σειριακές πράξεις σε δεδομένα,
- Διαφορετικοί τύποι μνημών με συγκεκριμένες φυσικές διευθύνσεις:
 - Μία που περιέχει τις οδηγίες που απαιτούνται για την εκτέλεση των πράξεων
 - Μια προσωρινή που περιέχει τα δεδομένα πάνω στα οποία πρέπει να εκτελεστούν οι πράξεις αυτές και τα αποτελέσματα που προκύπτουν
 - Μια μόνιμη μνήμη όπου φορτώνονται και αποθηκεύονται δεδομένα

Βιολογικά νευρικά συστήματα. Σε αντίθεση με τους σειριακούς υπολογιστές, ο ανθρώπινος εγκέφαλος αποτελείται από ένα δίκτυο περίπου 100 δισεκατομμυρίων νευρώνων. Κάθε νευρώνας εκτελεί τη λειτουργία του στοιχείου επεξεργασίας και έχει συνδέσεις με περίπου 10.000 νευρώνες. Κάθε νευρώνας έχει ένα χαρακτηριστικό κατώφλι ενεργοποίησης και, όταν το άθροισμα των λαμβανόμενων σημάτων υπερβαίνει το κατώφλι ενεργοποίησης του νευρώνα, ο νευρώνας στέλνει ένα νέο σήμα στον επόμενο νευρώνα. Αυτές οι δομές δικτύου επιτρέπουν την παράλληλη και καταναεμημένη επεξεργασία των σημάτων. Το νευρωνικό δίκτυο αναλύει τα σήματα και τα επεξεργάζεται αξιοποιώντας όλους τους κόμβους προκειμένου να παρέχει μια συγκεκριμένη απόκριση που σχετίζεται με τα σήματα εισόδου.

Βιολογικοί νευρώνες. Όταν κάποιος μιλάει για βιολογικούς νευρώνες με απλουστευμένο τρόπο, αναφέρεται σε ένα κύτταρο με τρεις περιοχές:

- Οι δενδρίτες αντιπροσωπεύουν την περιοχή εισόδου του νευρώνα και λαμβάνουν ηλεκτρικά σήματα από άλλους νευρώνες ή από το εξωτερικό,
- Το κυτταρικό σώμα ή σόμα είναι η περιοχή ενσωμάτωσης του σήματος. Ενσωματώνει τα σήματα που λαμβάνονται στους δενδρίτες,
- Ο άξονας είναι η περιοχή εξόδου που στέλνει ηλεκτρικά σήματα εάν η ολοκληρωμένη τιμή των λαμβανόμενων σημάτων υπερβαίνει το κατώφλι του νευρώνα.

Τύποι βιολογικών νευρώνων. Οι αισθητήριοι νευρώνες λαμβάνουν σήματα από το εξωτερικό περιβάλλον. Προκαλούν τη διάδοση των σημάτων εντός του νευρωνικού δικτύου. Το σήμα εξαπλώνεται μέχρι τους κινητικούς νευρώνες στο τέλος του νευρικού συστήματος. Οι κινητικοί

νευρώνες ελέγχουν τη σύσπαση των μυών και των αδένων μας, δημιουργώντας μια συγκεκριμένη απόκριση στο σήμα που προκάλεσε τη διάδοση του σήματος. Μεταξύ των αισθητηριακών νευρώνων και των κινητικών νευρώνων, υπάρχει ένας άλλος τύπος νευρώνων: οι συνειρμικοί νευρώνες ή ενδονευρώνες και δεν επικοινωνούν προς τα έξω. Αυτοί είναι οι πιο συνηθισμένοι νευρώνες μέσα στο νευρικό σύστημα. Επιτρέπουν μια κατανεμημένη και παράλληλη επεξεργασία των σημάτων, εξασφαλίζοντας την επιλογή της σωστής απόκρισης που σχετίζεται με ένα συγκεκριμένο ερέθισμα.

Βιολογικές συνάψεις. Οι συνάψεις είναι εξειδικευμένες βιολογικές δομές που αποτελούν τα σημεία σύνδεσης μεταξύ διαφορετικών νευρώνων. Έχουν θεμελιώδη ρόλο στη διαμόρφωση της έντασης των σημάτων που ρέουν από τον προσυναπτικό νευρώνα στον μετασυναπτικό. Οι συνάψεις μπορεί να είναι διεγερτικού τύπου όταν μεταδίδουν σήματα που ευνοούν την ενεργοποίηση του επόμενου νευρώνα ή ανασταλτικού τύπου όταν δυσχεραίνουν την ενεργοποίηση του μετασυναπτικού νευρώνα. Οι συνάψεις αλλάζουν τη δύναμή τους με την οποία μεταδίδουν τα σήματα με την πάροδο του χρόνου. Η διαδικασία αυτή οφείλεται στη συνεχή μάθηση που δημιουργείται από την αλληλεπίδραση με το περιβάλλον. Δεν υπάρχει ένα συγκεκριμένο τμήμα του νευρικού συστήματος που έχει σκοπό την αποθήκευση αναμνήσεων αλλά όλες οι συνάψεις ολόκληρου του νευρικού συστήματος αποθηκεύουν κατανεμημένες αναμνήσεις.

Διαφορές μεταξύ σειριακών υπολογιστών και βιολογικών νευρωνικών συστημάτων. Συνοπτικά, οι κύριες διαφορές μεταξύ ενός σειριακού υπολογιστή και του ανθρώπινου εγκεφάλου είναι:

- *Τύπος επεξεργασίας:* Στους σειριακούς υπολογιστές, οι επεξεργαστές εφαρμόζουν σειριακές λογικομαθηματικές πράξεις. Στα βιολογικά νευρικά συστήματα, η επεξεργασία πραγματοποιείται με παράλληλο τρόπο μεταξύ όλων των νευρώνων που αποτελούν το δίκτυο. Κάθε νευρώνας συμβάλλει ταυτόχρονα στην επεξεργασία του σήματος.
- *Θέση των μνημών:* Στους σειριακούς υπολογιστές, οι μνήμες έχουν συγκεκριμένη φυσική μοναδική διεύθυνση σε προσωρινές ή μόνιμες μνήμες. Στα βιολογικά νευρικά συστήματα, οι συνάψεις ολόκληρων νευρωνικών δικτύων περιέχουν όλες τις κατανεμημένες μνήμες που έχουν αποκτηθεί από την εμπειρία.
- *Αυτόνομη μάθηση:* Σε αντίθεση με τους σειριακούς υπολογιστές, τα νευρικά συστήματα δεν χρειάζεται να προγραμματιστούν για την επίλυση μιας συγκεκριμένης εργασίας, αλλά μαθαίνουν αυτόνομα με βάση την εμπειρία τους. Οι διαδικασίες μάθησης τροποποιούν τη δύναμη με την οποία οι συνάψεις μεταδίδουν σήματα μεταξύ ενός νευρώνα και του επόμενου, προκειμένου να παραχθεί η συγκεκριμένη απόκριση σε μια συγκεκριμένη είσοδο

Τεχνητά νευρωνικά δίκτυα (ANN). Τα ANN είναι συστήματα επεξεργασίας πληροφοριών εμπνευσμένα από τα βιολογικά νευρικά συστήματα. Τα ANNs επεξεργάζονται δεδομένα με παράλληλο και κατανεμημένο τρόπο. Αποτελούνται από ένα δίκτυο στοιχειωδών μονάδων που ονομάζονται τεχνητοί νευρώνες. Οι μονάδες μπορούν να συνδεθούν με διαφορετικό τρόπο ανάλογα με τη συγκεκριμένη αρχιτεκτονική του δικτύου που οργανώνεται σε διαφορετικά επίπεδα:

- **Στρώμα εισόδου:** Ορισμένοι τεχνητοί νευρώνες των ANN αντιπροσωπεύουν το κανάλι εισόδου για τα δεδομένα. Οι νευρωνικές μονάδες που λαμβάνουν δεδομένα ονομάζονται μονάδες εισόδου. Το στρώμα εισόδου επιτρέπει στα ANN να "αντιλαμβάνονται" τι υπάρχει έξω από το δίκτυο, προσομοιώνοντας τη δραστηριότητα των αισθητήριων βιολογικών νευρώνων.

- **Στρώμα εξόδου:** Άλλες μονάδες του δικτύου ονομάζονται νευρώνες εξόδου και έχουν ως σκοπό την προσομοίωση των λειτουργιών των βιολογικών κινητικών νευρώνων. Δίνουν μια συγκεκριμένη απόκριση στο σήμα εισόδου που ενεργοποιεί την εξάπλωση του δικτύου.
- **Κρυφά στρώματα:** Μεταξύ των στρωμάτων εισόδου και εξόδου μπορεί να υπάρχουν ενδιάμεσα στρώματα που ονομάζονται κρυφά στρώματα. Δεν επικοινωνούν με το εξωτερικό του δικτύου. Οι νευρώνες που βρίσκονται σε αυτά τα στρώματα ονομάζονται κρυφές μονάδες και προσομοιώνουν τις λειτουργίες των βιολογικών ενδονευρώνων.

Τεχνητοί νευρώνες. Ξεκινώντας από τα χαρακτηριστικά του βιολογικού νευρώνα το 1947, οι Warren McCulloch και Walter Pitts επινόησαν το πρώτο μοντέλο τεχνητού νευρώνα για να εξηγήσουν τη λειτουργία των νευρικών κυττάρων. Κάθε τεχνητός νευρώνας έχει την απλή λειτουργία να λαμβάνει σήματα και να τα αθροίζει για να προσομοιώνει τις δραστηριότητες των δενδριτών. Ο νευρώνας γίνεται ενεργός εάν το άθροισμα των σημάτων υπερβαίνει το κατώφλι του νευρώνα σε έναν μηχανισμό παρόμοιο με το βιολογικό σόμα. Ο ακόλουθος τύπος συνοψίζει τη δραστηριότητα ενός τεχνητού νευρώνα:

$$y = \Phi (\sum w * x - \theta)$$

όπου x αντιπροσωπεύει το διάνυσμα των εισόδων που λαμβάνει ο νευρώνας, w είναι το διάνυσμα των βαρών των συνάψεων από τις οποίες ο νευρώνας λαμβάνει το διάνυσμα των εισόδων, θ είναι το χαρακτηριστικό κατώφλι ενεργοποίησης του νευρώνα και Φ αντιπροσωπεύει τη συνάρτηση ενεργοποίησης του νευρώνα, η οποία μπορεί να λάβει διάφορες μορφές προσομοιώνοντας τη δραστηριότητα του βιολογικού άξονα. Οι νευρώνες του δικτύου συνδέονται με τεχνητές συνάψεις.

Τεχνητές συνάψεις. Τα σημεία σύνδεσης μεταξύ των τεχνητών νευρώνων ονομάζονται τεχνητές συνάψεις ή βάρη συνάψεων και προσομοιώνουν τη δραστηριότητα των βιολογικών συνάψεων. Δρουν μεταβάλλοντας την ένταση του σήματος που μεταδίδεται από τον προσυναπτικό νευρώνα στον μετασυναπτικό. Το διάνυσμα των βαρών των συνάψεων περιέχει πραγματικούς αριθμούς και συνήθως μπορεί να πάρει τιμές μεταξύ -1 και 1, αλλά σε συγκεκριμένες περιπτώσεις ξεπερνά αυτές τις τιμές. Η τιμή 1 σημαίνει ότι η σύναψη μεταδίδει ένα σήμα με την πλήρη έντασή του, το 0,5 δείχνει ότι το σήμα περνάει με τη μισή του ένταση και το 0 δείχνει ότι το σήμα δεν μεταδίδεται μεταξύ του προηγούμενου και του επόμενου νευρώνα. Όταν μια σύναψη παίρνει αρνητικές τιμές, υποδηλώνει ότι η σύναψη είναι ανασταλτική και καταστέλλει την ενεργοποίηση του επόμενου νευρώνα. Στα μη εκπαιδευμένα νευρωνικά δίκτυα, τα βάρη των συνάψεων είναι μηδενικά ή σχεδόν μηδενικά, ομοίως με αυτό που παρατηρείται στα βιολογικά νευρικά συστήματα των νεογνών. Σε αυτή την κατάσταση, τα σήματα δεν μπορούν να εξαπλωθούν στο εσωτερικό του δικτύου και απαιτείται μια διαδικασία εκπαίδευσης για να παραχθεί η σωστή απάντηση στη συγκεκριμένη εργασία που σχετίζεται με τα δεδομένα εισόδου.

Λειτουργίες ενεργοποίησης. Προσομοιώστε τη συχνότητα ενεργοποίησης των βιολογικών νευρώνων. Η επιλογή της συνάρτησης εξαρτάται από την προς επίλυση εργασία:

- **Συνάρτηση Heaviside ή βηματική συνάρτηση:** η έξοδος αυτής της συνάρτησης είναι δυαδική: 0 εάν το άθροισμα των εισόδων δεν υπερβαίνει το κατώφλι ενεργοποίησης - 1 εάν το άθροισμα υπερβαίνει το κατώφλι.
- **Σιγμοειδής:** η έξοδος αυτής της συνάρτησης επιτρέπει στον νευρώνα να έχει συνεχή απόκριση εντός του εύρους (0,1). Η απόκριση του νευρώνα αυξάνεται αναλογικά καθώς αυξάνεται το άθροισμα των εισόδων και αντίστροφα.

- *Rectified linear unit (ReLU)*: αυτή η συνάρτηση λαμβάνει το μέγιστο μεταξύ του 0 και του τετραγωνικού γινομένου των εισόδων. Δεν περιορίζει την έξοδο του νευρώνα στο 1 αλλά έχει γραμμική σχέση.

Νευρώνας κατωφλίου και προκατάληψης. Η λήψη των εισόδων ενός τεχνητού νευρώνα έχει εθιστικές ιδιότητες. Για το λόγο αυτό, ο τύπος δραστηριότητας τεχνητού νευρώνα παραλείπει τα κατώφλια του νευρώνα (θ) αντικαθιστώντας τα με την προσθήκη ενός νευρώνα εισόδου στο δίκτυο. Αυτός ο νευρώνας συνδέεται με όλους τους νευρώνες του δικτύου και έχει την ονομασία "νευρώνας προκατάληψης". Ο νευρώνας προκατάληψης έχει σταθερή ενεργοποίηση στην τιμή 1. Τα συναπτικά βάρη των συνδέσεων του ονομάζονται προκατάληψη και αντιπροσωπεύουν το κατώφλι θ κάθε νευρώνα που απαρτίζει το δίκτυο. Τα συναπτικά βάρη των συνδέσεων μεροληψίας μπορούν να αλλάξουν κατά τη διάρκεια της διαδικασίας μάθησης όπως και τα τυπικά συναπτικά βάρη. Αυτός ο μηχανισμός επιτρέπει στο δίκτυο να προσαρμόζει αυτόματα το κατώφλι ενεργοποίησης κάθε νευρώνα.

Φάση μάθησης. Όπως και άλλοι αλγόριθμοι μηχανικής μάθησης, οι ANN χρειάζονται μια φάση εκπαίδευσης για να μάθουν έναν πιθανό μηχανισμό για την επίλυση μιας συγκεκριμένης εργασίας. Κατά τη διάρκεια της φάσης εκπαίδευσης, τα ANN αλληλεπιδρούν με δεδομένα εκπαίδευσης που υποκρύπτουν την προς επίλυση εργασία, με παρόμοιο τρόπο με αυτό που συμβαίνει στα βιολογικά νευρικά συστήματα όταν αποκτούν εμπειρία σε μια συγκεκριμένη εργασία. Για το λόγο αυτό, τα δεδομένα που χρησιμοποιούνται στη φάση εκπαίδευσης ονομάζονται δεδομένα εκπαίδευσης ή σύνολο εκπαίδευσης. Τα ANN αλλάζουν τις τιμές των συναπτικών βαρών τους αλληλεπιδρώντας με το σύνολο εκπαίδευσης αναδρομικά, προκειμένου να ελαχιστοποιήσουν το σφάλμα που παρουσιάζουν κατά την εκτέλεση της εργασίας. Ο τρόπος με τον οποίο το ANN υπολογίζει το σφάλμα εξαρτάται από μια συγκεκριμένη συνάρτηση που ονομάζεται συνάρτηση LOSS.

Μαθησιακά παραδείγματα. Υπάρχουν τρία διαφορετικά παραδείγματα μάθησης που δρουν στα συναπτικά βάρη με βάση διαφορετικούς κανόνες:

- *Μάθηση με επίβλεψη*: τα δεδομένα εκπαίδευσης περιέχουν την απάντηση στην εργασία. Κατά την αλληλεπίδραση με τα δεδομένα εκπαίδευσης, το ANN παρουσιάζει σφάλμα στη λύση. Τα συναπτικά βάρη τροποποιούνται για να μειωθεί το σφάλμα μεταξύ της πραγματικής λύσης και της προβλεπόμενης λύσης.
- *Μάθηση χωρίς επίβλεψη*: τα δεδομένα εκπαίδευσης δεν περιέχουν την απάντηση στην εργασία. Το ANN χρησιμοποιεί τους χρόνους καθυστέρησης μεταξύ των ενεργοποιήσεων των νευρώνων για να τροποποιήσει τα συναπτικά του βάρη.
- *Ενισχυτική μάθηση*: Τα δεδομένα εκπαίδευσης δεν περιέχουν την απάντηση στην εργασία. Στο ANN προστίθενται ενισχυτικές συνθήκες κατά τη φάση κωδικοποίησης. Το δίκτυο διερευνά όλες τις πιθανές εξόδους του. Όταν το δίκτυο λαμβάνει ενίσχυση, τα συναπτικά του βάρη αλλάζουν ώστε να ευνοείται η επανάληψη αυτής της εξόδου

Παρά τις διαφορές τους, το αποτέλεσμα των διαφορετικών διαδικασιών μάθησης μπορεί να γενικευτεί σύμφωνα με τον ακόλουθο τύπο:

$$w_t = w_{t-1} + \eta \Delta w_t$$

όπου w_{t-1} αντιπροσωπεύει τα συναπτικά βάρη του δικτύου στον προηγούμενο κύκλο μάθησης (η γνώση που αποκτήθηκε προηγουμένως)- Δw_t είναι η διαφορά στα συναπτικά βάρη που δημιουργούνται από τον τρέχοντα κύκλο μάθησης (η γνώση που αποκτήθηκε σε αυτόν τον κύκλο μάθησης). Το Δw_t εξαρτάται από τον συγκεκριμένο μηχανισμό μάθησης που χρησιμοποιείται για την εκπαίδευση των ANN- η ($0 \leq \eta \leq 1$) είναι ο ρυθμός μάθησης του δικτύου και αντιπροσωπεύει τη συναπτική πλαστικότητα: πόσο εύκολα το δίκτυο τροποποιεί τα βάρη του κατά τη διαδικασία μάθησης- w_t είναι η τιμή των συναπτικών βαρών στο τέλος αυτού του κύκλου μάθησης.

Γενικές ιδιότητες των ANN. Οι ANN έχουν ορισμένα συναρπαστικά χαρακτηριστικά σε τομείς έρευνας ή εφαρμογής. Ορισμένες ιδιότητες εξαρτώνται από τη συγκεκριμένη αρχιτεκτονική δικτύου, αλλά άλλες είναι γενικές σε όλους τους τύπους αρχιτεκτονικής:

- **Ανθεκτικότητα:** Οι ANN είναι ανθεκτικοί στο θόρυβο και δίνουν τη σωστή απάντηση ακόμη και αν η είσοδος περιέχει θόρυβο ή αν διακοπούν οι εσωτερικές συνδέσεις του δικτύου (κάποιοι από τους τεχνητούς νευρώνες απενεργοποιούνται).
- **Ευελιξία:** Η ίδια αρχιτεκτονική ANN μπορεί να χρησιμοποιηθεί για την επίλυση διαφορετικών εργασιών. Ένα δίκτυο μπορεί να εκτελεί διαφορετικές λειτουργίες ανάλογα με τα δεδομένα που χρησιμοποιούνται κατά τη φάση της εκπαίδευσης.
- **Γενίκευση:** Αφού εκπαιδευτεί για την επίλυση μιας εργασίας με το συγκεκριμένο σύνολο εκπαίδευσης, το ίδιο νευρωνικό δίκτυο μπορεί να επιλύσει την ίδια εργασία χρησιμοποιώντας δεδομένα που δεν περιέχονται στο σύνολο εκπαίδευσης.

Διαφορετικές αρχιτεκτονικές ANNs. Οι διαφορετικές αρχιτεκτονικές ANNs μπορούν να χωριστούν σε δύο μεγάλες ομάδες ανάλογα με τη σύνδεση μεταξύ των νευρώνων που απαρτίζουν το δίκτυο. Στα feedforward ANNs, το σήμα διαδίδεται μονόδρομα από το στρώμα εισόδου στο στρώμα εξόδου χωρίς ποτέ να επιστρέφει πίσω. Τα feedforward ANNs μπορούν να επιλύσουν εργασίες που περιλαμβάνουν εργασίες ταξινόμησης ή παλινδρόμησης ή να εξαγάγουν σχετικές κρυφές πληροφορίες που περιέχονται στις εισόδους. Στα αναδρομικά ANN, το σήμα διαδίδεται με αμφίδρομο τρόπο επιστρέφοντας στους τεχνητούς νευρώνες που βρίσκονται ανάντη στα δίκτυα. Τα αναδρομικά ANN μπορούν να χρησιμοποιηθούν για την αποθήκευση εισόδων στη μνήμη ή για την ανακατασκευή κατεστραμμένων δεδομένων.

Perceptrons ενός στρώματος. Είναι η πρώτη αρχιτεκτονική ANN που σχεδιάστηκε από τον Frank Rosenblatt το 1958. Πρόκειται για ένα ANN με πρόωση. Μόνο το στρώμα εισόδου και το στρώμα εξόδου αποτελούν αυτή την απλή αρχιτεκτονική. Κάθε νευρώνας που συνθέτει το στρώμα εισόδου έχει μονόδρομες συνδέσεις με όλους τους νευρώνες του στρώματος εξόδου (αρχιτεκτονική feedforward). Μπορεί να χρησιμοποιηθεί για την επίλυση εργασιών ταξινόμησης και γραμμικής παλινδρόμησης εφόσον τα δεδομένα είναι γραμμικά διαχωρίσιμα. Οι μονάδες εξόδου μπορούν να χρησιμοποιούν διαφορετικές συναρτήσεις ενεργοποίησης οι οποίες επιλέγονται με βάση τις εργασίες που πρέπει να επιλύσει το ANN. Τα perceptrons χρησιμοποιούν διαδικασίες μάθησης με επίβλεψη για την ενημέρωση των συναπτικών τους βαρών. Στο επόμενο μάθημα εξετάζεται λεπτομερώς αυτή η αρχιτεκτονική δικτύου και υλοποιείται με τη χρήση pythοn για την επίλυση εργασιών δυαδικής ταξινόμησης.

Αυτοοργανωμένοι χάρτες: αποτελείται από ένα στρώμα εισόδου και ένα στρώμα εξόδου, όπου κάθε νευρώνας στέλνει σήματα σε όλους τους νευρώνες του δικτύου. Τα συναπτικά βάρη είναι σταθερά και δεν απαιτούν διαδικασία εκπαίδευσης. Ένας μηχανισμός που παρατηρείται στα

βιολογικά νευρικά συστήματα, ο οποίος ονομάζεται πλευρική αναστολή, ενέπνευσε τις συνδέσεις μεταξύ των μονάδων του στρώματος εξόδου. Δρα με βάση τον ανταγωνισμό μεταξύ των νευρώνων του στρώματος εξόδου. Όταν μια συγκεκριμένη είσοδος εξαπλώνεται στο δίκτυο ενεργοποιεί συγκεκριμένους νευρώνες στο στρώμα εξόδου, ενώ άλλοι νευρώνες προκύπτουν ανασταλμένοι. Όταν παρόμοιες εισροές εξαπλώνονται στο δίκτυο, οι κοντινοί νευρώνες στο στρώμα εξόδου οδηγούνται σε παρόμοια ενεργοποίηση. Αυτή η ιδιότητα είναι χρήσιμη για την επίλυση εργασιών ομαδοποίησης δεδομένων

Δίκτυο Hopfield. Πρόκειται για μια αναδρομική αρχιτεκτονική ANN και αποτελείται από μια σειρά μονάδων που συνδέονται αμφίδρομα. Κάθε μονάδα ανήκει τόσο στο στρώμα εισόδου όσο και στο στρώμα εξόδου. Κάθε μονάδα χρησιμοποιεί μια συνάρτηση ενεργοποίησης Heaviside και εναλλάσσεται μεταξύ της ενεργής (1) και της ανενεργής (0) κατάστασης. Όταν η είσοδος εξαπλώνεται στο δίκτυο, οι μονάδες περνούν μεταξύ των ενεργών και ανενεργών καταστάσεων έως ότου το δίκτυο φτάσει σε ένα σημείο ισορροπίας. Το σημείο ισορροπίας αντιπροσωπεύει μια συγκεκριμένη νευρωνική κωδικοποίηση για την είσοδο. Ένας παρόμοιος μηχανισμός έχει παρατηρηθεί στους βιολογικούς νευρώνες που αποτελούν τον υπόκαμπο, την περιοχή του εγκεφάλου που σχετίζεται με τις επεισοδιακές αναμνήσεις. Τα δίκτυα Hopfield χρησιμοποιούνται για την αποκατάσταση κατεστραμμένων δεδομένων ή για την αποθήκευση δεδομένων.

Μηχανή Boltzmann: Αυτή η αρχιτεκτονική ANN είναι παρόμοια με το δίκτυο Hopfield, αλλά έχει επίσης κρυφές μονάδες που δεν επικοινωνούν με το εξωτερικό. Όπως και τα δίκτυα Hopfield, οι μηχανές Boltzmann μπορούν να αποκαταστήσουν κατεστραμμένα δεδομένα ή να τα αποθηκεύσουν, αλλά οι κρυφές μονάδες επιτρέπουν την αποθήκευση και την επισκευή πιο σύνθετων εισόδων.

Περιπτώσεις χρήσης

ΠΡΟΕΤΟΙΜΑΣΙΑ ΥΠΟΛΟΓΙΣΤΗ ΓΙΑ ΤΗΝ ΚΩΔΙΚΟΠΟΙΗΣΗ ΑΛΓΟΡΙΘΜΟΥ ΒΑΣΙΣΜΕΝΟΥ ΣΕ ANNs ΧΡΗΣΙΜΟΠΟΙΩΝΤΑΣ ΤΗΝ PYTHON

σε αυτή την περίπτωση χρήσης, θα προετοιμάσουμε τον υπολογιστή σας για την κωδικοποίηση αλγορίθμων που βασίζονται σε ANNs χρησιμοποιώντας python:

- Εγκατάσταση του Anaconda/Python
- Εγκατάσταση της βιβλιοθήκης Keras
- Εγκατάσταση της βιβλιοθήκης OpenCV

Βήμα 1: Εγκατάσταση του Anaconda/Python

Μια από τις γλώσσες κωδικοποίησης που χρησιμοποιούνται συχνά για την κωδικοποίηση ANN και άλλων αλγορίθμων ML είναι η Python. Διαθέτει μια σειρά πλεονεκτημάτων κατάλληλων για τον σκοπό αυτό:

- Γλώσσα προγραμματισμού υψηλού επιπέδου: Επιτρέπει επίσης στους χρήστες που δεν είναι πολύ έμπειροι στην επιστήμη των υπολογιστών να μάθουν να προγραμματίζουν διαισθητικά και γρήγορα. Αυτό το χαρακτηριστικό είναι σημαντικό για την ανάπτυξη ANN λόγω της ισχυρής διεπιστημονικότητας που απαιτείται. Χρησιμοποιώντας την Python μπορούν επίσης βιολόγοι, ψυχολόγοι και γνωστικοί επιστήμονες να μάθουν να

αναπτύσσουν εφαρμογές χωρίς να είναι αναγκασμένοι να μάθουν πολύπλοκες έννοιες της πληροφορικής.

- Αναγνωσιμότητα του κώδικα: Οι κώδικες που είναι γραμμένοι σε Python είναι εύκολα αναγνώσιμοι, επιτρέποντας την αποτελεσματική εργασία σε μια ομάδα με υψηλή διεπιστημονικότητα.
- Εξαιρετικά ενσωματώσιμος: Μπορεί να εκτελεστεί σε όλα τα σύγχρονα λειτουργικά συστήματα και επιτρέπει την ενσωμάτωση στον κώδικα που αναπτύσσεται με άλλους αλγορίθμους που έχουν αναπτυχθεί σε διαφορετικές γλώσσες προγραμματισμού
- Διαθεσιμότητα βιβλιοθηκών ανοικτού κώδικα: Διαθέτει αρκετές εύκολα χρησιμοποιήσιμες βιβλιοθήκες χάρη στην πολύ λεπτομερή τεκμηρίωση. Περιέχουν αντικείμενα και συναρτήσεις κατάλληλες για την ταχεία ανάπτυξη εφαρμογών τεχνητής νοημοσύνης.

Για να ξεκινήσετε, [εγκαταστήστε το Anaconda](#), μια πλατφόρμα ανοικτού κώδικα που χρησιμοποιείται για την εκτέλεση της επιστήμης δεδομένων και της μηχανικής μάθησης με Python χρησιμοποιώντας μεμονωμένους υπολογιστές. Σχεδιάστηκε για να ασκεί σόλο και επιτρέπει να χειρίζεται χιλιάδες πακέτα και βιβλιοθήκες ανοικτού κώδικα. Το Anaconda περιέχει διάφορα χρήσιμα λογισμικά που μπορούν να διαχειριστούν μέσω του Anaconda Navigator. Σε αυτή την ενότητα εκμάθησης, οι κώδικες εκτελούνται στο Spyder, ένα περιβάλλον για την ανάπτυξη επιστημονικών εφαρμογών Python. Αποτελεί ένα προηγμένο IDE που περιλαμβάνει προηγμένες λειτουργίες γραφής, αποσφαλμάτωσης και διαδραστικής δοκιμής. Το Jupyter είναι ένα διαδραστικό διαδικτυακό περιβάλλον. Επιτρέπει στους χρήστες να γράφουν και να επεξεργάζονται έγγραφα, συμπεριλαμβανομένης της ανάλυσης δεδομένων. Επιπλέον, το Anaconda προσφέρει τη δυνατότητα χρήσης μιας ειδικής γραμμής εντολών Anaconda. Η γραμμή εντολών του Anaconda είναι ακριβώς όπως η γραμμή εντολών, αλλά εξασφαλίζει ότι μπορείτε να χρησιμοποιήσετε τις εντολές Anaconda και conda από τη γραμμή εντολών, χωρίς να χρειάζεται να αλλάξετε καταλόγους ή διαδρομές. Η προτροπή Anaconda είναι βολική για τη διαχείριση περιβαλλόντων, πακέτων και βιβλιοθηκών.

Βήμα 2: Εγκατάσταση της βιβλιοθήκης Keras

Τώρα εγκαταστήστε την πρώτη βιβλιοθήκη ανοικτού κώδικα: Keras. Η Keras αντιπροσωπεύει ένα υψηλού επιπέδου API που βασίζεται σε διαφορετικά backends για άλλες βιβλιοθήκες χαμηλού επιπέδου, όπως η Theano, το Microsoft Cognitive Toolkit (CNTK) και η TensorFlow. Επιτρέπει την ταχεία δημιουργία πρωτοτύπων ANN και DNN. Τα κύρια χαρακτηριστικά του Keras είναι τα εξής:

- Αρθρωτότητα, όλες οι ενότητες της Keras μπορούν να συνδυαστούν
- Επεκτασιμότητα, μπορούν να προστεθούν νέες ενότητες.

Το Keras μπορεί να χρησιμοποιήσει GPUs της NVIDIA για να εκτελέσει τους υπολογισμούς με υψηλή υπολογιστική ισχύ. Τεκμηρίωση: <https://keras.io/>

Βήμα 3: Εγκατάσταση της εργαλειοθήκης CUDA

Εάν ο υπολογιστής διαθέτει [GPU NVIDIA](#), μεταβείτε στον ιστότοπο της NVIDIA για να κατεβάσετε και να εγκαταστήσετε την εργαλειοθήκη CUDA, ένα περιβάλλον ανάπτυξης για τη δημιουργία εφαρμογών υψηλής απόδοσης με επιτάχυνση GPU.

Βήμα 4: Εγκατάσταση του cuDNN

Κατεβάστε τη [βιβλιοθήκη βαθιών νευρωνικών δικτύων NVIDIA CUDA](#) (cuDNN), μια βιβλιοθήκη με GPU-επιταχυνόμενη βιβλιοθήκη πρωτογενών στοιχείων για βαθιά νευρωνικά δίκτυα. Προσθέστε το CUDA στις διαδρομές περιβάλλοντος των windows.

Βήμα 5: Εγκατάσταση Keras

Τώρα είμαστε έτοιμοι να εγκαταστήσουμε το Keras σε ένα ειδικό περιβάλλον anaconda. Τα ειδικά περιβάλλοντα αποφεύγουν τα προβλήματα συμβατότητας μεταξύ διαφορετικών πακέτων. Δημιουργήστε ένα αποκλειστικό περιβάλλον και εγκαταστήστε το Keras εκτελώντας την ακόλουθη εντολή στην προτροπή του anaconda:

- `conda create -n keras python=3.6 numpy scipy keras-gpu (GPU-version)`
- `conda install -c conda-forge keras (CPU-version)`

Βήμα 6: Επαλήθευση της εγκατάστασης και της έκδοσης του Keras

Για να επαληθεύσετε την εγκατάσταση και την έκδοση του Keras. Εγκαταστήστε το Spyder στο ειδικό περιβάλλον Keras και εκκινήστε αυτό το IDE python εκτελώντας τις ακόλουθες εντολές:

- `conda install Spyder`
- `spyder`

Τώρα εκτελέστε αυτόν τον κώδικα στην κονσόλα spyder για να εισαγάγετε τη βιβλιοθήκη Keras και να επαληθεύσετε την έκδοση του Keras:

- `import keras`
- `keras.__version__`

Βήμα 7: Εγκατάσταση του OpenCV

Η εγκατάσταση της βιβλιοθήκης Keras είναι πιο περίπλοκη από τις τυπικές βιβλιοθήκες python, καθώς απαιτείται η εγκατάσταση του περιβάλλοντος CUDA και του. Ας προχωρήσουμε τώρα με την εγκατάσταση της βιβλιοθήκης OpenCV-python. Αυτή η βιβλιοθήκη επιτρέπει τη χρήση αλγορίθμων υπολογιστικής όρασης (CV) για την εργασία με εικόνες με διαφορετικούς τρόπους. Για να την εγκαταστήσετε, μεταβείτε στην ιστοσελίδα της Anaconda και αναζητήστε την εντολή που πρέπει να εκτελέσετε στην προτροπή anaconda. Στη συνέχεια ενεργοποιήστε το περιβάλλον Keras και εκτελέστε την εντολή εγκατάστασης:

1. `conda install -c conda-forge opencv`

Βήμα 8: Επαλήθευση της εγκατάστασης του OpenCV

Εκτελέστε αυτόν τον κώδικα στην κονσόλα spyder για να εισαγάγετε το OpenCV και να επαληθεύσετε την έκδοση της βιβλιοθήκης:

1. `import opencv`

opencv.__version__

Αναφορές

- Artificial Intelligence: 101 Things You Must Know Today About Our Future-Lasse Rouhiainen, Create Space Independent Publishing Platform, 2018. ISBN: 978-1982048808

Μάθημα 3.2: Απλά Perceptrons και μάθηση με επίβλεψη

Θέματα προς συζήτηση

1. Αρχιτεκτονική και λειτουργία απλών perceptrons
2. Κωδικοποίηση απλών perceptrons για την επίλυση εργασιών δυαδικής ταξινόμησης AND/OR
3. Perceptrons ενός στρώματος

Σημειώσεις διάλεξης

Απλά perceptrons. Το απλό perceptron που επινοήθηκε από τον Rosenblatt το 1958 είναι η πρώτη αρχιτεκτονική ANN που αναπτύχθηκε. Πρόκειται για ένα feedforward δίκτυο που αποτελείται από ένα στρώμα N νευρώνων εισόδου που συνδέονται μονόδρομα με τον ένα μόνο νευρώνα στο στρώμα εξόδου. Ο νευρώνας εξόδου χρησιμοποιεί τη συνάρτηση Heaviside ως συνάρτηση ενεργοποίησης (δυαδική έξοδος $[0,1]$). Ορίζεται ως "απλή" επειδή είναι η πιο απλή αρχιτεκτονική ANNs που μπορεί να αναπτυχθεί. Η έξοδος του απλού perceptron παίρνει δυαδικές τιμές $[0,1]$. Αυτό το ANN μπορεί να χρησιμοποιηθεί για να προβλέψει αν ένα συγκεκριμένο δεδομένο ανήκει σε μια δεδομένη κλάση σε ένα έργο δυαδικής ταξινόμησης. Ο νευρώνας εξόδου που συνθέτει το στρώμα εξόδου παίρνει την τιμή μηδέν εάν το γινόμενο με τελείες μεταξύ του διανύσματος εισόδου και του διανύσματος των συναπτικών βαρών δεν υπερβαίνει το κατώφλι ενεργοποίησης του νευρώνα εξόδου, ενώ όταν το γινόμενο τελείας υπερβαίνει το κατώφλι ενεργοποίησης του νευρώνα εξόδου, η έξοδος παίρνει την τιμή ένα.

Εργασία δυαδικής ταξινόμησης. Η δυαδική ταξινόμηση συνίσταται στην ταξινόμηση της υπαγωγής ενός στοιχείου δεδομένων σε μια συγκεκριμένη κατηγορία στοιχείων. Τυπικές εργασίες δυαδικής ταξινόμησης περιλαμβάνουν ιατρικές εξετάσεις για να προσδιοριστεί αν ένα συγκεκριμένο υποκείμενο έχει ή όχι μια συγκεκριμένη ασθένεια- έλεγχο ποιότητας για να προβλεφθεί αν μια συγκεκριμένη προδιαγραφή δεδομένων είναι σωστή- συστήματα προειδοποίησης για να ελεγχθεί αν τα δεδομένα μιας συγκεκριμένης διαδικασίας φτάνουν σε κρίσιμες τιμές και άλλες εργασίες. Στις εργασίες δυαδικής ταξινόμησης υπάρχουν δύο διαφορετικοί τύποι σφαλμάτων: στα ψευδώς αρνητικά (ανάκληση), το σφάλμα συνίσταται στη μη αναγνώριση της υπαγωγής των δεδομένων στην κλάση ενδιαφέροντος. Αντίθετα, στα ψευδώς θετικά (precision), το σφάλμα συνίσταται στην εσφαλμένη αναγνώριση της υπαγωγής των δεδομένων στην κλάση ενδιαφέροντος.

Αξιολόγηση της ταξινόμησης. Είναι δυνατή η αξιολόγηση της ακρίβειας που επιτυγχάνεται στην εργασία ταξινόμησης με τη χρήση συναρτήσεων της ενότητας sklearn.metrics. Η μέθοδος classification_report() εκτυπώνει έναν πίνακα που περιέχει πληροφορίες σχετικά με την ακρίβεια ταξινόμησης όσον αφορά την ακρίβεια και την ανάκληση. Η μέθοδος confusion_matrix() εκτυπώνει έναν πίνακα με δύο διαστάσεις: πραγματική κλάση και προβλεπόμενη κλάση. Είναι

χρήσιμο να έχετε μια σαφή ιδέα για τα ψευδώς θετικά και τα ψευδώς αρνητικά σε εργασίες ταξινόμησης.

Μάθηση με επίβλεψη. Όπως και άλλα παραδείγματα μάθησης, ο μηχανισμός μάθησης με επίβλεψη δρα τροποποιώντας τις συνάψεις που συνδέουν όλους τους νευρώνες του δικτύου. Η διαδικασία μάθησης με επίβλεψη στοχεύει στην προσομοίωση μιας βιολογικής διαδικασίας μάθησης που βλέπει την παρουσία ενός εξωτερικού δασκάλου. Ο δάσκαλος βοηθά στην επίλυση μιας εργασίας παρέχοντας την ακριβή απάντηση στην εργασία. Γνωρίζοντας την απάντηση στην εργασία, ο μαθητής αναγνωρίζει τα λάθη του και τροποποιεί τις απαντήσεις του με την πάροδο του χρόνου, μειώνοντας τα εμφανιζόμενα λάθη. Θα μπορούσαμε να πούμε ότι μέσω μιας διαδικασίας μάθησης με επίβλεψη, κάποιος μπορεί να μάθει τη λειτουργία που διέπει την επίλυση μιας δεδομένης εργασίας ξεκινώντας από τη λύση που παρέχεται εξωτερικά. Τα ANN που εκμεταλλεύονται ένα μηχανισμό μάθησης με επίβλεψη αλληλεπιδρούν με ένα σύνολο εκπαίδευσης με ετικέτες που περιέχει τη λύση της εργασίας (ετικέτα), προκειμένου να αναπαράγουν τη μαθηματική συνάρτηση που διέπει το πρόβλημα.

Σετ εκπαίδευσης με ετικέτες. Σε ένα ANN που αξιοποιεί μια διαδικασία μάθησης με επίβλεψη για να μάθει τον μηχανισμό που διέπει την επίλυση μιας συγκεκριμένης εργασίας, ο εξωτερικός δάσκαλος παρέχει τη λύση της εργασίας (ετικέτες) στο σύνολο εκπαίδευσης. Το σύνολο εκπαίδευσης με ετικέτες που χρησιμοποιείται στην επιβλεπόμενη μάθηση περιέχει μια σειρά αριθμητικών παραδειγμάτων που περιλαμβάνουν τη σχετική ετικέτα. Κάθε παράδειγμα εκπαίδευσης στο σύνολο εκπαίδευσης αποτελείται από ένα ζεύγος αριθμητικών διανυσμάτων:

- Διάνυσμα εισόδου: ένα διάνυσμα (x) όπου τα στοιχεία του αντιπροσωπεύουν τις διάφορες τιμές που εισέρχονται στο επίπεδο εισόδου
- Διάνυσμα επιθυμητής απόκρισης: ένα διάνυσμα (d) που περιέχει την επιθυμητή έξοδο κάθε νευρώνα στο στρώμα εξόδου.

Το σύνολο εκπαίδευσης με ετικέτες θα εξαπλωθεί στο ANN επανειλημμένα, προκειμένου να μειωθεί το σφάλμα που παρουσιάζει το ANN για την επίλυση της εργασίας. Τα διάφορα παραδείγματα που περιέχονται στο σύνολο εκπαίδευσης με ετικέτες διαχέονται στο δίκτυο αναδρομικά δημιουργώντας αρκετές ενημερώσεις βαρών. Η αναδρομική ιδιότητα της μάθησης αποτρέπει τη διαγραφή της γνώσης που έχει αποκτηθεί προηγουμένως από τη νέα γνώση.

Επανακλιμάκωση δεδομένων. Τα ANN απαιτούν προεπεξεργασμένα δεδομένα για να λειτουργήσουν. Όλα τα δεδομένα εισόδου του ANN πρέπει να αναβαθμιστούν στην περιοχή [0-1]. Το σύνολο εκπαίδευσης με ετικέτες μπορεί εύκολα να αναβαθμιστεί σε αυτό το εύρος χρησιμοποιώντας την κλάση `MinMaxScaler` στην ενότητα `sklearn.preprocessing` που περιλαμβάνεται στο πακέτο `Sklearn`. (Τεκμηρίωση)

Εκπαιδευτική φάση 1. Όταν ένα μη εκπαιδευμένο απλό perceptron ξεκινά τη διαδικασία μάθησης με επίβλεψη, οι τιμές των συναπτικών βαρών του τίθενται στο μηδέν ή σε τιμές πολύ κοντά στο μηδέν, συνήθως στο εύρος (-0,1 / 0,1). Όταν το διάνυσμα εισόδου εξαπλωθεί στο δίκτυο οδηγεί σε συγκεκριμένη ενεργοποίηση των νευρώνων εξόδου του. Το διάνυσμα που περιέχει την ενεργοποίηση των νευρώνων εξόδου συγκρίνεται με το επιθυμητό διάνυσμα εξόδου που ανήκει στο ίδιο παράδειγμα εκπαίδευσης του διανύσματος εισόδου. Εάν δεν υπάρχει διαφορά με το

επιθυμητό διάνυσμα εξόδου ($y=d$), τα συναπτικά βάρη δεν αλλάζουν. Εάν υπάρχει διαφορά μεταξύ της επιθυμητής εξόδου και της εξόδου του δικτύου ($y \neq d$), τα συναπτικά βάρη αλλάζουν.

Κανόνας Δέλτα. Η παρουσία σφάλματος συνεπάγεται την τροποποίηση των συναπτικών βαρών σύμφωνα με τον ακόλουθο μαθηματικό κανόνα που ονομάζεται κανόνας δέλτα:

$$\Delta w_t = (d_t - y_t) x_t$$

Η προσαρμογή των συναπτικών βαρών (Δw_t) ισούται με το γινόμενο με τελείες μεταξύ του σφάλματος (της διαφοράς μεταξύ του επιθυμητού διανύσματος εξόδου d_t και του διανύσματος εξόδου y_t) και του διανύσματος εισόδου (x_t). Η τιμή που προκύπτει μπορεί να προστεθεί στην τιμή των συναπτικών βαρών όπως είδαμε στο προηγούμενο μάθημα:

$$w_t = w_{t-1} + \eta \Delta w_t$$

Όπου η είναι ο ρυθμός μάθησης του δικτύου. Οι τιμές αυτής της υπερπαραμέτρου του δικτύου διαμορφώνονται χειροκίνητα στο εύρος $[0-1]$. Υψηλότεροι ρυθμοί μάθησης οδηγούν σε ταχεία αλλαγή των συναπτικών βαρών

Εκπαιδευτική φάση 2. Όταν όλα τα παραδείγματα εκπαίδευσης που περιέχονται στο σύνολο εκπαίδευσης εξαπλωθούν στο νευρωνικό δίκτυο, μια εποχή εκπαίδευσης τελειώνει. Καθώς προχωρούν οι εποχές εκπαίδευσης, ο αριθμός των σφαλμάτων που πραγματοποιεί το νευρωνικό δίκτυο μειώνεται σταδιακά. Η διαδικασία εκπαίδευσης ολοκληρώνεται όταν το δίκτυο δεν κάνει πλέον λάθη στις απαντήσεις που δίνονται στο διάνυσμα εισόδου σε μια εποχή εκπαίδευσης.

Απλός ψευδοκώδικας perceptron. Αυτός είναι ο γενικός ψευδοκώδικας για την έναρξη και την εκπαίδευση ενός απλού perceptron με N μονάδες εισόδου και μία μονάδα εξόδου.

1. Init synaptic weights (θυμηθείτε το βάρος του νευρώνα μεροληψίας)
2. Set learning rate ($0 \leq \eta \leq 1$)
3. while $y \neq d \rightarrow b$ ενώ η ενεργοποίηση του νευρώνα εξόδου διαφέρει από την επιθυμητή απόκριση
 - a. if $\sum w_t x_t > 0 \rightarrow$ γινόμενο με τελείες (συναπτικά βάρη, είσοδος)
 $y_t = 1 \rightarrow$ ενεργοποίηση νευρώνα εξόδου
else
 $y_t = 0 \rightarrow$ ενεργοποίηση νευρώνα εξόδου
 - b. if $y_t = d_t \rightarrow$ εάν ενεργοποίηση του νευρώνα εξόδου = επιθυμητή έξοδος
 $\Delta w_t = 0 \rightarrow$ δεν αλλάζουν τα συναπτικά βάρη
else
 $\Delta w_t = (d_t - y_t) x_t \rightarrow$ υπολογισμός διανύσματος δέλτα συναπτικών βαρών

c. $w_t = w_{t-1} + \Delta w_t \rightarrow$ ενημέρωση συναπτικών βαρών

AND απλό perceptron. Αυτό το απλό perceptron μαθαίνει πώς να εκτελεί τη λογική συνάρτηση AND. Στην περίπτωση αυτή, δύο νευρώνες εισόδου αποτελούν το επίπεδο εισόδου του απλού perceptron. Κάθε νευρώνας εισόδου λαμβάνει δυαδικές τιμές (0-1). Μόνο μία μονάδα εξόδου συνθέτει το στρώμα εξόδου. Στο perceptron AND, η μονάδα εξόδου ενεργοποιείται όταν και οι δύο μονάδες εισόδου είναι ενεργές. Σε όλες τις άλλες περιπτώσεις, η μονάδα εξόδου είναι ανενεργή.

OR απλό perceptron. Η ιδιότητα της ευελιξίας των ANN επιτρέπει στην ίδια αρχιτεκτονική να εκτελεί διαφορετικές λειτουργίες και η λειτουργία που μαθαίνεται εξαρτάται από το σύνολο εκπαίδευσης. Η ίδια αρχιτεκτονική AND perceptron μπορεί να αναλύσει την είσοδο εκτελώντας επίσης τη συνάρτηση boolean OR εάν εκπαιδευτεί με διαφορετικό σύνολο εκπαίδευσης. Στο perceptron OR, η μονάδα εξόδου ενεργοποιείται όταν τουλάχιστον μία μονάδα εισόδου είναι ενεργή. Όταν και οι δύο μονάδες εισόδου είναι ανενεργές, η μονάδα εξόδου είναι επίσης ανενεργή.

Δημιουργία set εκπαίδευσης AND/OR. Κωδικοποίηση σε Python, αρχίστε να αναπτύσσετε ένα πρώτο απλό perceptron από το μηδέν. Τα απλά σύνολα εκπαίδευσης AND/OR perceptron κωδικοποιούνται γρήγορα χρησιμοποιώντας το πακέτο NumPy. Το NumPy είναι το βασικό πακέτο για επιστημονικούς υπολογισμούς στην Python. Περιέχει πολλές χρήσιμες συναρτήσεις για τον υπολογισμό πινάκων και γενικά για τη γραμμική άλγεβρα. Ενεργοποιήστε δύο πίνακες numpy που περιέχουν τα σύνολα εκπαίδευσης AND/OR. Κάθε σειρά είναι ένα παράδειγμα εκπαίδευσης. Η πρώτη και η δεύτερη στήλη αντιπροσωπεύουν τις τιμές εισόδου, ενώ η τρίτη στήλη είναι η ενεργοποίηση του νευρώνα προκατάληψης. Η τελευταία στήλη περιέχει τις ετικέτες των παραδειγμάτων και αντιστοιχεί στην επιθυμητή έξοδο του δικτύου. Τα σύνολα εκπαίδευσης AND και OR διαφέρουν μόνο ως προς την επιθυμητή έξοδο του δικτύου.

Ενεργοποίηση απλού perceptron AND/OR. Ενεργοποίηση της αρχιτεκτονικής του απλού perceptron, των παραμέτρων και του συνόλου εκπαίδευσης

1. Έναρξη AND/OR perceptron.

- Ενεργοποιήστε τα βάρη των συνάψεων του AND/OR perceptron. μην ξεχάσετε να προσθέσετε το βάρος του νευρώνα προκατάληψης.
- Ορίστε το ρυθμό μάθησης του perceptron
- Ορίστε τον μέγιστο αριθμό εποχών εκπαίδευσης

2. Ορίστε τη συνάρτηση Heaviside

3. Επιλογή συνόλου εκπαίδευσης AND/OR

- λήψη διανυσμάτων εισόδου
- λήψη του επιθυμητού διανύσματος εξόδου

Εκπαίδευση απλού perceptron AND/OR. Στη φάση εκπαίδευσης, το απλό perceptron αλληλεπιδρά με το σύνολο εκπαίδευσης AND/OR για να ενημερώσει τα βάρη του και να μειώσει τα σφάλματά του.

1. Λίστα σφαλμάτων σε αρχικές εποχές εκπαίδευσης
2. Έναρξη βρόχου εποχών εκπαίδευσης:
 - a. Ενεργοποίηση του μετρητή σφαλμάτων της τρέχουσας εποχής εκπαίδευσης
 - b. Έναρξη βρόχου εκπαιδευτικών παραδειγμάτων
 - i. Απλή εξάπλωση perceptron
 - ii. Απλή εκπαίδευση perceptron
 - a. Αύξηση του τρέχοντος μετρητή εποχών
 - b. Υπολογισμός του δέλτα των συναπτικών βαρών
 - c. Ενημέρωση βαρών
 - iii. Εκτύπωση αποτελεσμάτων παραδείγματος εκπαίδευσης
 - c. Αποθήκευση του μετρητή σφάλματος της τρέχουσας εποχής εκπαίδευσης
 - d. Διακοπή της εκπαίδευσης εάν δεν υπάρχει σφάλμα κατά την τρέχουσα εποχή εκπαίδευσης

Αποτελέσματα εκπαίδευσης AND/OR perceptron. Στην περίπτωση αυτή, το απλό perceptron που εκπαιδεύτηκε με το σύνολο εκπαίδευσης AND δεν παρουσιάζει σφάλματα σε ολόκληρη την εποχή εκπαίδευσης μετά από πέντε εποχές εκπαίδευσης. Η ίδια αρχιτεκτονική απλού perceptron που εκπαιδεύτηκε με το σύνολο εκπαίδευσης OR δεν παρουσιάζει σφάλματα σε ολόκληρη την εποχή εκπαίδευσης μετά από πέντε εποχές εκπαίδευσης.

Λειτουργία απλών perceptrons. Τα απλά perceptrons AND/OR χωρίζουν όλα τα παραδείγματα εκπαίδευσης σε δύο κλάσεις. Η μία από αυτές τις κλάσεις ενεργοποιεί τη μονάδα εξόδου, ενώ η άλλη κλάση δεν την ενεργοποιεί. Σε μια δισδιάστατη είσοδο, μια μονοδιάστατη γραμμή αρκεί για να χωρίσει τις δύο κλάσεις:

$$ax + by + c = 0$$

Τα perceptrons AND/OR στο πλαίσιο μιας εποπτευόμενης διαδικασίας μάθησης βρίσκουν τη γραμμή a , b , c , τις παραμέτρους ενημερώνοντας τα συναπτικά τους βάρη:

$$w_1x_1 + w_2x_2 + w_3 = 0$$

Μόλις βρεθούν οι παράμετροι της γραμμής, το απλό perceptron μπορεί να ταξινομήσει την είσοδο υπολογίζοντας σε ποια πλευρά της γραμμής βρίσκεται η είσοδος.

$$w_1x_1 + w_2x_2 + w_3 > 0$$

Πολυδιάστατες εισοδοί με περισσότερες από δύο διαστάσεις (n-διάστατες) μπορούν να χωριστούν σχεδιάζοντας ένα υπερεπίπεδο (n-1) διαστάσεων:

$$w_1x_1 + \dots + w_{n-1}x_{n-1} + w_n = 0$$

Δεν μπορούν να διαχωριστούν όλες οι εισοδοί με γραμμές ή υπερεπίπεδα. Το όριο του απλού perceptron και γενικά των perceptron ενός στρώματος: η είσοδος πρέπει να είναι γραμμικά διαχωρίσιμη.

Σχεδίαση της γραμμής διαίρεσης perceptron. Ενεργοποιήστε τη γραφική παράσταση διαίρεσης και στη συνέχεια σχεδιάστε το διάνυσμα εισόδου ως σημεία. Για να σχεδιάσετε τη γραμμή διαίρεσης, υπολογίστε τις παραμέτρους m/c της γραμμής, ξεκινώντας από:

$$1) ax + by + c = 0$$

$$2) y = (-a/b)x + (-c/b)$$

τότε αντικαταστήστε τις παραμέτρους a, b, c με το διάνυσμα των συναπτικών βαρών ($a = w_1$, $b = w_2$, $c = w_3$)

$$1) w_1x + w_2y + w_3 = 0$$

$$2) y = (-w_1 / w_2)x + (-w_3 / w_2)$$

Απλό perceptron έναντι perceptrons ενός στρώματος. Τα απλά perceptrons ανήκουν στην κατηγορία των perceptrons ενός στρώματος, καθώς έχουν μόνο ένα στρώμα συνδεδεμένο με τις μονάδες εισόδου. Σε αντίθεση με τα απλά perceptrons, άλλα perceptrons ενός στρώματος μπορούν να χρησιμοποιούν συνεχείς και διαφορίσιμες συναρτήσεις ενεργοποίησης (Sigmoid, ReLU) και μπορούν να έχουν περισσότερες από μία μονάδες εξόδου. Η έξοδός τους δεν είναι δυαδική (0 ή 1) αλλά εμπίπτει σε ένα συνεχές εύρος που εξαρτάται από τη συνάρτηση ενεργοποίησης που χρησιμοποιείται. Τα perceptrons ενός στρώματος μπορούν να επιλύσουν εργασίες ταξινόμησης πολλαπλών κατηγοριών εφόσον τα δεδομένα είναι γραμμικά διαχωρίσιμα και εργασίες γραμμικής παλινδρόμησης.

Εργασία ταξινόμησης πολλαπλών κατηγοριών. Η ταξινόμηση πολλαπλών κατηγοριών συνίσταται στην ταξινόμηση της υπαγωγής ενός στοιχείου δεδομένων σε διαφορετικές κατηγορίες στοιχείων. Οι τυπικές εργασίες ταξινόμησης πολλαπλών κλάσεων είναι παρόμοιες με τις εργασίες δυαδικής ταξινόμησης, αλλά μπορούν να ταξινομήσουν περισσότερες από μία κλάσεις. Οι τυπικές εργασίες ταξινόμησης πολλαπλών κλάσεων περιλαμβάνουν:

- ιατρικές εξετάσεις για να προσδιοριστεί αν ένα συγκεκριμένο υποκείμενο έχει ή όχι διάφορες ασθένειες
- έλεγχος ποιότητας για να προβλεφθεί εάν μια συγκεκριμένη προδιαγραφή δεδομένων είναι σωστή,
- συστήματα προειδοποίησης για να ελέγξουν αν τα δεδομένα μιας συγκεκριμένης διαδικασίας φτάνουν σε κρίσιμες τιμές και πολλές άλλες εργασίες.

Όπως και στις εργασίες δυαδικής ταξινόμησης, υπάρχουν δύο διαφορετικοί τύποι σφαλμάτων: στα ψευδώς αρνητικά σφάλματα (ανάκληση), το σφάλμα συνίσταται στη μη αναγνώριση της υπαγωγής των δεδομένων στην κλάση ενδιαφέροντος. Αντίθετα, στα ψευδώς θετικά (precision), το σφάλμα συνίσταται στην εσφαλμένη αναγνώριση της υπαγωγής των δεδομένων στην κλάση ενδιαφέροντος.

Εργασίες γραμμικής παλινδρόμησης. Στις εργασίες παλινδρόμησης, το perceptron πρέπει να ποσοτικοποιήσει τη σχέση μεταξύ της ανεξάρτητης μεταβλητής και της εξαρτημένης μεταβλητής. Η εργασία γραμμικής παλινδρόμησης είναι η απλούστερη εργασία παλινδρόμησης, εφόσον υπάρχει γραμμική σχέση μεταξύ των δύο μεταβλητών.

Στην πραγματική ζωή, οι εργασίες γραμμικής παλινδρόμησης εκτελούνται σε:

- Επιχειρήσεις όταν προβλέπουν τα έσοδα ξεκινώντας από τις τιμές των δαπανών
- Ιατρική όταν προβλέπεται η επίδραση ενός φαρμάκου ξεκινώντας από τη συγκέντρωση
- Γυμναστική όταν προβλέπουν τις καταναλισκόμενες θερμίδες ξεκινώντας από τις ώρες σωματικής δραστηριότητας

Περιπτώσεις χρήσης

ΕΚΠΑΙΔΕΥΣΗ ΑΠΛΟΥ PERCEPTRON

Αυτή η περίπτωση χρήσης θα εξηγήσει πώς μπορεί να εκπαιδευτεί ένα απλό perceptron χρησιμοποιώντας ένα σύνολο δεδομένων που βασίζεται σε πραγματικά δεδομένα. Συγκεκριμένα, τα απλά perceptrons θα εκπαιδευτούν χρησιμοποιώντας ένα σύνολο εκπαίδευσης που περιέχει δεδομένα σχετικά με τα λουλούδια με σκοπό την ταξινόμηση των ειδών. Το perceptron που θα προκύψει από την εκπαίδευση θα μπορούσε να χρησιμοποιηθεί για την ανάπτυξη μιας καινοτόμου εφαρμογής για τις επιχειρήσεις ανθοκομίας.

[ΚΑΤΕΒΑΣΤΕ ΤΟΝ ΚΩΔΙΚΑ ΕΔΩ](#)

Βήμα 1: Φόρτωση συνόλου δεδομένων Iris

Η ενότητα Scikit-learn.datasets περιέχει διάφορα σύνολα δεδομένων που χρησιμοποιούνται για την εκπαίδευση αλγορίθμων ML. Για αυτή την περίπτωση χρήσης φορτώστε το σύνολο δεδομένων Iris. Αυτό το σύνολο δεδομένων περιέχει δεδομένα τριών διαφορετικών τύπων Iris: Setosa, Versicolour και Virginica. Το σύνολο δεδομένων Iris αποθηκεύει μια numpy.ndarray 150x4 που περιέχει δεδομένα 150 διαφορετικών λουλουδιών Iris με τέσσερα διαφορετικά αριθμητικά χαρακτηριστικά των λουλουδιών: μήκος και πλάτος σέπαλου, μήκος και πλάτος πετάλου.

Βήμα 2: Ανάλυση του συνόλου δεδομένων Iris

Το απλό perceptron θα εκπαιδευτεί να αναγνωρίζει την Iris Setosa με βάση δύο διαφορετικά χαρακτηριστικά: το μήκος και το πλάτος του σέπαλου. Το μήκος και το πλάτος του σέπαλου αντιστοιχούν στην πρώτη και τη δεύτερη στήλη των iris.data. Οι κλάσεις Iris αντιστοιχούν στη λίστα iris.target. Τα ονόματα των ειδών Iris αντιστοιχούν στο αρχείο iris.target_names. Για να διακρίνετε μόνο την Iris Setosa από τις άλλες κλάσεις, μετατρέψτε την κλάση 2 στην κλάση 1 (0 = Setosa, 1 = άλλα είδη). Η σχεδίαση του συνόλου δεδομένων για την ανάλυση της κατανομής των δεδομένων

μπορεί να βοηθήσει στην επιλογή της καλύτερης αρχιτεκτονικής ANNs για την εκτέλεση της εργασίας. Η Iris Setosa μπορεί να διαχωριστεί με μια απλή γραμμή από τα άλλα είδη Iris και, στη συνέχεια, είναι δυνατόν να χρησιμοποιηθεί ένα απλό perceptron για την εκτέλεση δυαδικής ταξινόμησης.

Βήμα 3: Init απλή αρχιτεκτονική perceptron

Αφού επιλεγεί η απλή αρχιτεκτονική perceptron, πρέπει να αρχικοποιηθεί. Ξεκινήστε να αρχικοποιείτε τα βάρη των συνάψεων του perceptron και ορίστε τον ρυθμό μάθησης, τον μέγιστο αριθμό εποχών εκπαίδευσης και ορίστε τη συνάρτηση βήματος. Τώρα το μη εκπαιδευμένο απλό perceptron είναι έτοιμο να αλληλεπιδράσει με το επεξεργασμένο σύνολο δεδομένων Iris.

Βήμα 4: Προεπεξεργασία συνόλου δεδομένων Iris

Τα απλά perceptrons και γενικά όλα τα ANN χρειάζονται είσοδο κλιμακωτή στο εύρος(0-1). Είναι δυνατή η κλιμάκωση της εισόδου σε αυτό το εύρος με τη χρήση της κλάσης MinMaxScaler της μονάδας προεπεξεργασίας του πακέτου sklearn. Μετά την αλλαγή της κλίμακας των αριθμητικών τιμών των δεδομένων, είναι δυνατόν να προστεθεί η στήλη που περιέχει την ενεργοποίηση του νευρώνα προκατάληψης που λαμβάνει τα διανύσματα εισόδου. Επιπλέον, η αναγνώριση του είδους Iris Setosa από άλλα είδη Iris είναι μια δυαδική εργασία ταξινόμησης για την οποία είναι απαραίτητη η σύγκλιση των ετικετών (0 = Iris Setosa, 1 = άλλα είδη Iris). Μέσω της συνάρτησης train_test_split στην ενότητα sklearn.model_selection, είναι δυνατή η πραγματοποίηση υποδιαίρεσης σε ένα σύνολο εκπαίδευσης και δοκιμής γρήγορα, τυχαία και ταυτόχρονα τόσο στα διανύσματα εισόδου όσο και στα επιθυμητά διανύσματα εξόδου, διατηρώντας τη σχέση μεταξύ των διανυσμάτων στα παραδείγματα εκπαίδευσης. Είναι δυνατό να καθορίσετε την κατάσταση random_state στο train_test_split για να αναπαράγετε τα αποτελέσματα.

Βήμα 5: Εκπαίδευση απλού perceptron Iris

Στη φάση της εκπαίδευσης, το απλό perceptron αλληλεπιδρά με το σύνολο εκπαίδευσης της ίριδας για να ενημερώσει τα βάρη του και να μειώσει το σφάλμα που παρουσιάζεται για να ανιχνεύσει τα σφάλματα Iris Setosa.

1. Init training epochs errors list
2. Start training epochs loop:
 1. Simple perceptron spreading
 2. Simple perceptron training
 - 1.Increment current epoch counter
 - 2.Compute delta of synaptic weights
 - 3.Update weights
 3. Print training example results
 4. Init current training epoch error counter
 5. Start training examples loop
 6. Store current training epoch error counter
 7. Stop training if no error during current training epoch

Βήμα 6: Αξιολογήστε την εκπαίδευση Iris

Η προβολή της τάσης των σφαλμάτων κατά τη φάση εκμάθησης είναι ένας εξαιρετικός τρόπος για τον έλεγχο των προβλημάτων που μπορεί να παρεμποδίζουν την ανάπτυξη ενός δικτύου ικανού να επιλύει το πρόβλημα με υψηλά επίπεδα ακρίβειας και με την ιδιότητα της γενίκευσης. Σε αυτή την περίπτωση, το απλό perceptron δεν παρουσιάζει κανένα σφάλμα στην αναγνώριση της Iris Setosa από άλλα είδη Iris σε μία εποχή εκπαίδευσης. Η γραμμή που προκύπτει με τη χρήση των συναπτικών βαρών του perceptron διαιρεί σωστά το σύνολο εκπαίδευσης διαχωρίζοντας το Iris Setosa από τα άλλα είδη Iris. Τα αποτελέσματα μπορεί να είναι διαφορετικά εάν δεν καθοριστεί η τυχαία κατάσταση στο `train_test_split`.

Βήμα 7: Δοκιμή Iris απλού perceptron

Μετά την αξιολόγηση της διαδικασίας εκπαίδευσης, το εκπαιδευμένο απλό perceptron είναι έτοιμο να αξιολογηθεί στο σύνολο δοκιμών. Αρχικοποιήστε μια λίστα και συμπληρώστε την με τις προβλέψεις του απλού perceptron. Χρησιμοποιήστε τις μεθόδους `confusion_matrix()` και `classification_report()` της ενότητας `sklearn.metrics` για να αξιολογήσετε την ακρίβεια ταξινόμησης των προβλέψεων.

Βήμα 8: Αποθήκευση του Iris perceptron

Αφού επαληθευτεί η ακρίβεια του δικτύου στην αναπαραγωγή μιας συγκεκριμένης λειτουργίας που ανταποκρίνεται στις ανάγκες του έργου, είναι δυνατόν να αποθηκευτεί η αξία των συναπτικών βαρών για μεταγενέστερη επαναχρησιμοποίηση ως ολοκληρωμένη λειτουργία σε μια συγκεκριμένη εφαρμογή που θα μπορούσε να καινοτομήσει τις διαδικασίες που διέπουν τις ανθοκομικές δραστηριότητες. Είναι επίσης δυνατή η αποθήκευση των τιμών που παράγονται από τις φάσεις εκμάθησης και δοκιμής, προκειμένου να συγκριθούν τα αποτελέσματα που προκύπτουν με τα αποτελέσματα άλλων υπολογιστικών μοντέλων.

Μάθημα 3.3: Πολυστρωματικά Perceptrons και Keras

Θέματα προς συζήτηση

1. Στοιχεία των βαθιών νευρωνικών δικτύων και της βαθιάς μάθησης
2. Κωδικοποίηση και εκπαίδευση ενός πολυστρωματικού perceptron με τη χρήση του Keras

Σημειώσεις διάλεξης

Βαθιά νευρωνικά δίκτυα (DNN). Η προσθήκη ενός ή περισσότερων κρυφών στρωμάτων μεταξύ των στρωμάτων εισόδου και εξόδου αντιπροσωπεύει την εξέλιξη των ANN σε βαθιά νευρωνικά δίκτυα (DNN). Εκτός από την υπέρβαση των περιορισμών των νευρωνικών δικτύων ενός στρώματος όταν εκτελούν εργασίες ταξινόμησης σε μη γραμμικά διαχωρίσιμα δεδομένα και μη γραμμικής παλινδρόμησης, τα DNN αποκτούν την ιδιότητα της αυτόνομης εξαγωγής χαρακτηριστικών διαφορετικών επιπέδων από την είσοδο.

Βαθιά μάθηση. Η προσθήκη κρυφών επιπέδων στις αρχιτεκτονικές δικτύων καθιστά τη διαδικασία μάθησης πιο περίπλοκη. Το σφάλμα που παρουσιάζουν τα DNN ή η Απώλεια δεν εξαρτάται αποκλειστικά από το σύνολο των πράξεων που εκτελούνται από ένα μόνο στρώμα, αλλά από το σύνολο των πράξεων που εκτελούνται από τα διάφορα στρώματα που απαρτίζουν το δίκτυο. Για την τροποποίηση των συναπτικών βαρών κατά τη φάση της μάθησης είναι απαραίτητο να υπολογιστεί η συνιστώσα του σφάλματος που οφείλεται σε κάθε στρώμα χρησιμοποιώντας έναν ειδικό αλγόριθμο: τον αλγόριθμο οπισθοδιάδοσης. Τα DNN μπορούν να επιλύσουν διαφορετικές εργασίες που απαιτούν διαφορετικές συναρτήσεις απώλειας για τον υπολογισμό της απώλειας που παρουσιάζεται στην επίλυση. Επιπλέον, κατά τη φάση της μάθησης, η τροποποίηση των συναπτικών βαρών βασίζεται σε αλγορίθμους αριθμητικής βελτιστοποίησης ή βελτιστοποιητές που έχουν ως σκοπό την ελαχιστοποίηση της εξόδου της συνάρτησης Loss.

Αλγόριθμος οπισθοδιάδοσης. Ο αλγόριθμος οπισθοδιάδοσης, που παρουσιάστηκε επίσημα από τους Rumelhart et al. το 1986, επιτρέπει τον υπολογισμό της συνιστώσας της Απώλειας που οφείλεται σε κάθε στρώμα που απαρτίζει τα DNN. Όταν μια είσοδος (x_t) εξαπλώνεται στο δίκτυο κατά τη διάρκεια της διαδικασίας μάθησης, το δίκτυο παρουσιάζει ένα σφάλμα ή Απώλεια (L) μεταξύ του επιθυμητού διανύσματος εξόδου (dt) και του διανύσματος εξόδου του (yt). Ο αλγόριθμος οπισθοδιάδοσης επιτρέπει τον υπολογισμό της κλίσης της Απώλειας (∇L) σε σχέση με κάθε συναπτικό βάρος στο DNN. Ο αλγόριθμος αυτός δρα με την αναδρομή της Απώλειας που παρουσιάζουν τα DNN από το στρώμα εξόδου στο στρώμα εισόδου μέσω του κανόνα της αλυσίδας, ενός τύπου για τον υπολογισμό της παραγώγου σύνθετων συναρτήσεων.

Συναρτήσεις απώλειας. Οι συναρτήσεις απώλειας είναι απλές ή σύνθετες κυρτές συναρτήσεις που χρησιμοποιούνται στη βαθιά μάθηση για τον υπολογισμό του σφάλματος κατά την επίλυση μιας συγκεκριμένης εργασίας. Υπάρχει πάντα ένα απόλυτο ελάχιστο και μπορεί να υπάρχουν και

τοπικά ελάχιστα. Οι συναρτήσεις αυτές μπορούν να χωριστούν ανάλογα με την εργασία στην οποία χρησιμοποιούνται για τον υπολογισμό του σφάλματος:

- Συναρτήσεις απώλειας παλινδρόμησης
- Συναρτήσεις απωλειών δυαδικής ταξινόμησης

Συναρτήσεις απώλειας παλινδρόμησης: Αυτή η ομάδα συναρτήσεων απωλειών υπολογίζει το σφάλμα ως τη διαφορά μεταξύ των πραγματικών και των προβλεπόμενων τιμών μιας εξαρτημένης μεταβλητής κατά την εκτέλεση μιας παλινδρόμησης:

Μέσο τετραγωνικό σφάλμα (MSE): Υπολογίζει το σφάλμα ως τη μέση τετραγωνική διαφορά μεταξύ των πραγματικών και των προβλεπόμενων τιμών. Χρησιμοποιείται όταν η επιθυμητή έξοδος είναι κανονικά κατανομημένη και είναι ζωτικής σημασίας να τιμωρούνται τα μεγαλύτερα σφάλματα περισσότερο από τα μικρά σφάλματα λόγω του ότι οι στόχοι έχουν μικρό εύρος:

$$MSE = \frac{1}{N} \sum_{t=1}^N (d_t - y_t)^2$$

Μέσο τετραγωνικό λογαριθμικό σφάλμα (MSLE): Υπολογισμός του σφάλματος ως ο μέσος όρος των λόγων μεταξύ των λογαρίθμων των πραγματικών και των προβλεπόμενων τιμών. Χρησιμοποιείται όταν η επιθυμητή έξοδος είναι κανονικά κατανομημένη και δεν είναι απαραίτητο να τιμωρούνται τα μεγαλύτερα σφάλματα περισσότερο από τα μικρά σφάλματα λόγω του ότι οι στόχοι έχουν ευρύ φάσμα.

$$MSLE = \frac{1}{N} \sum_{t=1}^N (\log(d_t + 1) - \log(y_t + 1))^2$$

Μέσο απόλυτο σφάλμα (MAE): Υπολογίζει το σφάλμα ως το μέσο όρο των απόλυτων διαφορών μεταξύ πραγματικών και προβλεπόμενων τιμών. Χρησιμοποιείται όταν η επιθυμητή έξοδος έχει πολυτροπική κατανομή ή για την καταστολή της συνεισφοράς των ακραίων τιμών:

$$MAE = \frac{1}{N} \sum_{t=1}^N |d_t - y_t|$$

Μέσο απόλυτο ποσοστιαίο σφάλμα (MAPE): Υπολογίζει το σφάλμα ως το μέσο όρο των απόλυτων ποσοστιαίων διαφορών μεταξύ πραγματικών και προβλεπόμενων τιμών. Χρησιμοποιείται όταν η επιθυμητή έξοδος δεν έχει ακρότατα ή μηδενικά. Τα σφάλματα κανονικοποιούνται ως προς τις πραγματικές τιμές:

$$MAPE = \frac{1}{N} \sum_{t=1}^N \frac{|d_t - y_t|}{d_t}$$

Συναρτήσεις απώλειας ταξινόμησης: Αυτή η κατηγορία συναρτήσεων απωλειών υπολογίζει το σφάλμα μεταξύ της πραγματικής και της προβλεπόμενης κλάσης σε εργασίες ταξινόμησης:

Δυαδική διασταυρούμενη εντροπία (BCE): Αυτή η συνάρτηση απώλειας είναι η προεπιλεγμένη συνάρτηση απώλειας για προβλήματα δυαδικής ταξινόμησης. Η BCE αυξάνει το σκορ απώλειας

εάν η προβλεπόμενη κλάση αποκλίνει από την πραγματική κλάση και δίνει πληροφορίες σχετικά με την πιθανότητα εξόδου. Απαιτεί σιγμοειδές ως συνάρτηση ενεργοποίησης του στρώματος εξόδου:

$$BCE = d_t \cdot \log y_t + (1 - d_t) \cdot \log (1 - y_t)$$

Τετραγωνισμένη άρθρωση (SH): Αυτή η συνάρτηση απώλειας εγγυάται το μέγιστο περιθώριο στο όριο ταξινόμησης μεταξύ των σημείων δεδομένων, αλλά δεν παρέχει πληροφορίες σχετικά με τις πιθανότητες εξόδου. Απαιτεί υπερβολική εφασπτομένη ως συναρτήσεις ενεργοποίησης του τελευταίου στρώματος:

$$SH = \sum_{I=0}^N (\max(0, 1 - d_t \cdot y_t)^2)$$

Κατηγορική διασταυρούμενη εντροπία (CCE): Αυτή η συνάρτηση απώλειας είναι η προεπιλεγμένη απώλεια για προβλήματα ταξινόμησης πολλαπλών κατηγοριών. Η κατηγορική διασταυρούμενη εντροπία υπολογίζει το σφάλμα ως τη διαφορά μεταξύ δύο κατανομών πιθανότητας. Απαιτεί πολλές μονάδες εξόδου ίσες με τον αριθμό των αναγνωρίσιμων κλάσεων και softmax ως συνάρτηση ενεργοποίησης του τελευταίου στρώματος. Η softmax είναι απαραίτητη για την κανονικοποίηση του διανύσματος εξόδου σε μια κατανομή πιθανότητας των αναγνωρίσιμων κλάσεων

$$CCE = \sum_{I=1}^{N^{classes}} (d_t - \log y_t)$$

Βελτιστοποιητές. Οι βελτιστοποιητές είναι μια κατηγορία επαναληπτικών αριθμητικών αλγορίθμων που χρησιμοποιούνται για την εύρεση των βέλτιστων τιμών των συναπτικών βαρών (σύγκλιση) και ανήκουν στην κατηγορία των αλγορίθμων βαθμωτής καθόδου. Υπάρχουν διάφοροι τύποι βελτιστοποιητών:

- Στοχαστική Επικλινής Κάθοδος (SGD) (Stochastic gradient descent)
- Προσαρμοστικός αλγόριθμος κλίσης (AdaGrad) (Adaptive gradient algorithm)
- Διάδοση μέσου τετραγωνικού συντελεστή (RMSProp) (Root mean square propagation)
- Προσαρμοστική εκτίμηση στιγμής (AdaM) (Adaptive moment estimation)
- ...

Αλγόριθμοι Επικλινής καθόδου: Οι αλγόριθμοι επικλινής καθόδου είναι επαναληπτικοί αλγόριθμοι πρώτης τάξης που ξεκινώντας από την κλίση της συνάρτησης μπορούν να βρουν ένα ελάχιστο σε μια διαφορίσιμη συνάρτηση (σύγκλιση). Οι αλγόριθμοι αυτοί δρουν επαναληπτικά μεταβάλλοντας την ανεξάρτητη μεταβλητή προς την αντίθετη κατεύθυνση από την κλίση της εξαρτημένης μεταβλητής. Όταν χρησιμοποιούνται στον τομέα της βαθιάς μάθησης, οι αλγόριθμοι βαθμωτής καθόδου έχουν αρκετές παραμέτρους:

- Ρυθμός εκμάθησης
- Μέγεθος παρτίδας

- Ορμή

Ρυθμός εκμάθησης. Μια από τις βασικές παραμέτρους όλων των βελτιστοποιητών είναι ο ρυθμός μάθησης. Ο ρυθμός μάθησης καθορίζει το εύρος της μεταβολής που αναλαμβάνουν οι τιμές των συναπτικών βαρών σε κάθε ενημέρωση των βαρών. Χαμηλότερη είναι η τιμή του ρυθμού μάθησης, χαμηλότερη είναι η ταχύτητα σύγκλισης και τα DNN μπορεί να σταθεροποιηθούν σε ένα τοπικό ελάχιστο της συνάρτησης απώλειας. Αντίθετα, χρησιμοποιώντας έναν πολύ υψηλό ρυθμό μάθησης, το δίκτυο μπορεί να συγκλίνει σε ελάχιστα της συνάρτησης απωλειών. Μια καλή πρακτική είναι η υιοθέτηση ενός προσαρμοστικού ρυθμού μάθησης με υψηλές τιμές στα αρχικά στάδια της φάσης μάθησης και χαμηλές τιμές στα τελευταία στάδια. Ο ρυθμός μάθησης μπορεί να διαμορφωθεί χειροκίνητα ή αυτόματα

Μέγεθος παρτίδας. Κάθε φορά που το δίκτυο ενημερώνει τα συναπτικά του βάρη ορίζει ένα βήμα εκπαίδευσης. Το μέγεθος παρτίδας καθορίζει τον αριθμό των παραδειγμάτων εκπαίδευσης που χρησιμοποιούνται για τον υπολογισμό της κλίσης της συνάρτησης απώλειας σε κάθε βήμα εκπαίδευσης. Διαφορετικό μέγεθος παρτίδας έχει διαφορετικά αποτελέσματα στη φάση μάθησης:

- Διαδικτυακή Επικλινής Κάθοδος(Online gradient descent): Ένα μόνο παράδειγμα εξαπλώνεται σε όλο το δίκτυο. Το σφάλμα που παρουσιάζεται αντιπροσωπεύει την κινητήρια δύναμη για την ενημέρωση των συναπτικών βαρών. Το σφάλμα κάθε παραδείγματος συμβάλλει εξίσου στην ενημέρωση των βαρών (μέγεθος παρτίδας = 1).
- Mini παρτίδα Επικλινής Καθόδου(Mini batch gradient descent): Το σύνολο εκπαίδευσης χωρίζεται σε υποσύνολα η παραδειγμάτων εκπαίδευσης. Όλα τα παραδείγματα που περιέχονται σε ένα υποσύνολο εξαπλώνονται στο δίκτυο. Το μέσο σφάλμα που παρουσιάζει ένα υποσύνολο αποτελεί την κινητήρια δύναμη για την ενημέρωση των συναπτικών βαρών. Το σφάλμα κάθε παραδείγματος συμβάλλει λιγότερο στην ενημέρωση των βαρών όσο αυξάνεται το μέγεθος του υποσυνόλου (μέγεθος παρτίδας = n).
- Κλιμακωτή Επικλινής Κάθοδος(Batch gradient descent): Όλα τα παραδείγματα εκπαίδευσης που περιέχονται στο σύνολο εκπαίδευσης κατανέμονται στο δίκτυο. Το μέσο σφάλμα ολόκληρου του συνόλου εκπαίδευσης αποτελεί την κινητήρια δύναμη για την ενημέρωση των συναπτικών βαρών. (μέγεθος παρτίδας = N)

Τα μικρά μεγέθη παρτίδων καθιστούν τη φάση μάθησης θορυβώδη και απαιτούν χαμηλούς ρυθμούς μάθησης, αλλά ευνοούν την ιδιότητα γενίκευσης του δικτύου. Από την άλλη πλευρά, τα μεγάλα μεγέθη παρτίδων καθιστούν τη διαδικασία μάθησης πιο καθαρή επιτρέποντας υψηλότερους ρυθμούς μάθησης, αλλά το δίκτυο θα μπορούσε να εξειδικευτεί υπερβολικά στο σύνολο εκπαίδευσης, χάνοντας την ικανότητα γενίκευσης. Επιπλέον, τα πολύ μεγάλα μεγέθη παρτίδων μπορεί να προκαλέσουν κορεσμό της μνήμης όταν χρησιμοποιούνται μεγάλα σύνολα εκπαίδευσης.

Ορμή. Μια μέθοδος βελτιστοποίησης για την επιτάχυνση της καθόδου κλίσης προς τη σχετική κατεύθυνση. Η ορμή αυξάνεται εάν η κλίση έχει το διάνυσμα κατεύθυνσής της και μειώνεται εάν η κλίση έχει άλλες κατευθύνσεις. Η ορμή αποτρέπει τη σταθεροποίηση σε τοπικά ελάχιστα και μειώνει την ταλάντωση στη διαδικασία εκπαίδευσης. Η ορμή Nesterov είναι ένας ιδιαίτερος τύπος ορμής, στον οποίο η κατεύθυνση της ορμής αλλάζει δυναμικά κατά τη διάρκεια του βήματος εκπαίδευσης, εάν η κατεύθυνση της ορμής είναι λανθασμένη ή υπερβάλλει.

Στοχαστική Επικλινής Κάθοδος (Stochastic gradient descent SGD). Αυτός ο αλγόριθμος βελτιστοποίησης προσεγγίζει την πραγματική κλίση της συνάρτησης απώλειας, η οποία εξαρτάται από ολόκληρο το σύνολο εκπαίδευσης. Στην SGD η προσέγγιση της κλίσης εξαρτάται από μεμονωμένα ή λίγα παραδείγματα εκπαίδευσης:

$$w_{t+1} = w_t - \nabla L(w)$$

όπου w_{t+1} και w_t είναι αντίστοιχα οι καθαρές και τρέχουσες τιμές των συναπτικών βαρών, η είναι ο ρυθμός μάθησης και $\nabla L(w)$ είναι η κλίση της συνάρτησης απώλειας. Η ενημέρωση των βαρών χρησιμοποιώντας μόνο ένα παράδειγμα εκπαίδευσης προκαλεί θορυβώδη μεταβολή στη συνάρτηση Απώλειας κατά τη φάση της εκπαίδευσης και απαιτεί χαμηλούς ρυθμούς μάθησης. Σε κάθε εποχή εκπαίδευσης, το σύνολο εκπαίδευσης ανακατεύεται για να αφήσει την εξάπλωση των παραδειγμάτων εκπαίδευσης σε διαφορετική σειρά. Το ανακάτεμα αποτρέπει τη σταθεροποίηση σε τοπικά ελάχιστα.

Αλγόριθμος προσαρμοστικής κλίσης (Adagrad Adaptive gradient algorithm): Αυτός ο αλγόριθμος βελτιστοποίησης αλλάζει προσαρμοστικά το ρυθμό μάθησης κατά τη διάρκεια της φάσης εκπαίδευσης. Σε αντίθεση με τον SGD, όπου όλα τα συναπτικά βάρη ενημερώνονται ταυτόχρονα με τον ίδιο ρυθμό μάθησης, ο Adagrad χρησιμοποιεί συγκεκριμένους ρυθμούς μάθησης για κάθε συναπτικό βάρος που αλλάζουν σε κάθε βήμα εκπαίδευσης:

$$w_{t+1} = w_t - \frac{n}{\sqrt{G_t}} \nabla L(w)$$

Ο όρος G αντιπροσωπεύει έναν παράγοντα κλιμάκωσης για ένα μεμονωμένο συναπτικό βάρος και αποσκοπεί στην αναπροσαρμογή του ρυθμού μάθησης και εξαρτάται από τις παρελθοντικές κλίσεις. Καθώς προχωρά η φάση εκπαίδευσης, ο όρος G αυξάνεται, μειώνοντας τον ρυθμό μάθησης ώστε να ευνοείται η σύγκλιση.

Μέση τετραγωνική διάδοση (Root mean square propagation RMSprop): RMSProp είναι ένας αλγόριθμος προσαρμοστικού ρυθμού μάθησης. Προσπαθεί να λύσει τη μείωση του μονοτονικού ρυθμού μάθησης του Adagrad, εισάγοντας έναν κινητό μέσο όρο (KM) της τετραγωνικής κλίσης ειδικά για κάθε συναπτικό βάρος:

$$w_{t+1} = w_t - \frac{n}{\sqrt{MA}} \nabla L(w)_t$$

Προσαρμοστική εκτίμηση ροπής (Adaptive moment estimation AdaM): Είναι ένας από τους πιο δημοφιλείς αλγορίθμους βελτιστοποίησης:

$$w_{t+1} = w_t - \frac{n}{\sqrt{MA}}$$

όπου m αντιπροσωπεύει τον εκθετικό κινητό μέσο όρο της δεύτερης στιγμής της κλίσης. Αυτός ο αλγόριθμος έχει τα πλεονεκτήματα του SGD με ορμή σε συνδυασμό με τον προσαρμοστικό ρυθμό μάθησης του RMSProp. Λειτουργεί καλά με μικρό μέγεθος παρτίδας έχει μικρή χρήση μνήμης.

Πολυστρωματικά perceptrons (MLPs). Η προσθήκη ενός ή περισσότερων κρυφών στρωμάτων μεταξύ των μονάδων εισόδου και εξόδου ενός απλού perceptron επιτρέπει την ανάπτυξη πολυεπίπεδων perceptrons (MLPs). Αυτά τα απλά νευρωνικά δίκτυα βαθιά τροφοδοσίας (DNNs) μπορούν να επιλύσουν εργασίες ταξινόμησης που ξεκινούν από μη γραμμικά διαχωριστικά δεδομένα και εργασίες μη γραμμικής παλινδρόμησης. Ένα παράδειγμα μάθησης με επίβλεψη διέπει τη φάση εκπαίδευσης, που απαιτεί ένα σύνολο με ετικέτες. Ο αριθμός των μονάδων εξόδου εξαρτάται από την εργασία που στοχεύει να επιλύσει το MLP:

- Παλινδρόμηση: MLP έχει μόνο μία μονάδα εξόδου με γραμμική συνάρτηση ενεργοποίησης στο στρώμα εξόδου. Η τιμή της εξόδου της μονάδας αντιπροσωπεύει την προβλεπόμενη τιμή της εξαρτημένης μεταβλητής ξεκινώντας από τις ανεξάρτητες μεταβλητές.
- Δυαδική ταξινόμηση: Το MLP έχει μόνο μία μονάδα εξόδου με συνάρτηση ενεργοποίησης Heaviside. Η μονάδα εξόδου λαμβάνει δυαδικές τιμές (0,1). Η τιμή της μονάδας εξόδου αντιπροσωπεύει αν μια είσοδος ανήκει σε μια συγκεκριμένη κλάση.
- Ταξινόμηση πολλαπλών κατηγοριών: Το MLP διαθέτει πολλές μονάδες εξόδου με συνάρτηση ενεργοποίησης softmax. Ο αριθμός των μονάδων εξόδου εξαρτάται από τον αριθμό των κλάσεων που μπορεί να αναγνωρίσει το δίκτυο. Οι τιμές των μονάδων εξόδου κανονικοποιούνται σε μια κατανομή πιθανότητας μεταξύ των αναγνωρίσιμων κλάσεων.

MLP για την ταξινόμηση πολλαπλών κατηγοριών Iris Σε αυτό το μέρος του μαθήματος, θα υλοποιηθεί ένα MLP στο Keras για να εκτελέσει μια ταξινόμηση πολλαπλών κατηγοριών. Το σύνολο δεδομένων iris της ενότητας sklearn.datasets περιέχει δεδομένα από τρία είδη Iris: Setosa, Versicolour και Virginica αποθηκευμένα σε μια numpy.ndarray 150x4 (150 διαφορετικά άνθη ίριδας, 4 διαφορετικά αριθμητικά χαρακτηριστικά: μήκος και πλάτος σέπαλου, μήκος και πλάτος πετάλου). Το στρώμα εισόδου του MLP έχει τέσσερις μονάδες εισόδου (5 με την προκατάληψη) και 3 μονάδες εξόδου με ένα κρυφό στρώμα οκτώ νευρώνων.

1. **Λήψη και προεπεξεργασία δεδομένων.** Το MLP θα εκπαιδευτεί χρησιμοποιώντας και τα τέσσερα χαρακτηριστικά Iris: μήκος και πλάτος σέπαλου, μήκος και πλάτος πετάλου. Οι κλάσεις ίριδας αντιστοιχούν στον κατάλογο iris.target. Τα ονόματα των κλάσεων Iris βρίσκονται στον κατάλογο iris.target_name. Όπως τα απλά perceptrons, έτσι και τα DNN χρειάζονται είσοδο κλιμακωτή στο εύρος(0-1). Είναι δυνατή η αναβαθμολόγηση της εισόδου σε αυτό το εύρος χρησιμοποιώντας την κλάση MinMaxScaler στην ενότητα sklearn.preprocessing. Η Keras προσθέτει αυτόματα τον νευρώνα μεροληψίας. Η κατηγορηματική κλάση ίριδας πρέπει να μετατραπεί σε μορφή κωδικοποίησης one-hot για να ληφθεί το επιθυμητό διάνυσμα εξόδου χρησιμοποιώντας τη μέθοδο to_categorical στο module keras.utils. Παραδείγματα μετατροπής one-hot encode: μετατροπή κλάσης "0" = [1,0,0] - η πρώτη μονάδα εξόδου γίνεται ενεργή- μετατροπή κλάσης "1" = [0,1,0] - η δεύτερη μονάδα εξόδου γίνεται ενεργή, κ.λπ. Μέσω της συνάρτησης train_test_split στην ενότητα sklearn.model_selection, είναι δυνατή η πραγματοποίηση υποδιαίρεσης σε ένα σύνολο εκπαίδευσης και δοκιμής γρήγορα, τυχαία και ταυτόχρονα τόσο στα διανύσματα εισόδου όσο και στα επιθυμητά διανύσματα εξόδου, διατηρώντας τη σχέση μεταξύ των διανυσμάτων στα παραδείγματα εκπαίδευσης. Είναι δυνατό να καθορίσετε την κατάσταση random_state στο train_test_split για να αναπαράγετε τα αποτελέσματα.

2. **Αρχιτεκτονική MLP.** Αρχικοποίηση μιας αρχιτεκτονικής MLP με ένα κρυφό στρώμα οκτώ νευρώνων. `models.Sequential()` επιτρέπει την αρχικοποίηση ενός νέου μοντέλου όπου κάθε στρώμα προστίθεται διαδοχικά στο προηγούμενο με τη χρήση της `model.add()`. `layers.Dense()`: είναι μια κατηγορία στρώματος DNNs όπου κάθε νευρώνας έχει συνδέσεις με όλους τους νευρώνες του προηγούμενου στρώματος. Το κρυφό πυκνό στρώμα έχει ως συνάρτηση ενεργοποίησης την ReLU. Το στρώμα εξόδου είναι ένα πυκνό στρώμα και έχει Softmax ως συνάρτηση ενεργοποίησης. Ο αριθμός των μονάδων εξόδου αντιστοιχεί στον αριθμό των κλάσεων που μπορεί να αναγνωρίσει το δίκτυο.
3. **Μεταγλώττιση και εκπαίδευση του MLP.** `model.compile()` διαμορφώνει το μοντέλο για τη φάση εκπαίδευσης. Οι συναρτήσεις απώλειας, οι βελτιστοποιητές (με ρυθμό μάθησης) και οι μετρικές ακρίβειας μπορούν εύκολα να διαμορφωθούν σε αυτό το βήμα. `model.summary()` εκτύπωση στην κονσόλα της αρχιτεκτονικής του δικτύου με τον αριθμό των παραμέτρων που μπορούν να εκπαιδευτούν. `model.fit()` έναρξη της φάσης εκπαίδευσης. Οι εποχές εκπαίδευσης, το μέγεθος της παρτίδας, τα παραδείγματα εκπαίδευσης και τα παραδείγματα επικύρωσης μπορούν να οριστούν εδώ. Τα δεδομένα επικύρωσης αντιπροσωπεύουν ένα μέρος των δεδομένων εκπαίδευσης που χρησιμοποιούνται για τη δοκιμή του δικτύου κατά τη φάση της εκπαίδευσης. Είναι χρήσιμο να επισημανθεί τότε το δίκτυο χάνει την ιδιότητα γενίκευσης (overfitting). Συνήθως, η επικύρωση αντιπροσωπεύει το 20% των δεδομένων εκπαίδευσης. Στην περίπτωση αυτή, τα δεδομένα δοκιμής χρησιμοποιούνται ως δεδομένα επικύρωσης
4. **Αξιολόγηση της εκπαίδευσης.** Σχεδιασμός του ιστορικού εκπαίδευσης του μοντέλου για εντοπισμό τυχόν προβλημάτων που εμφανίστηκαν κατά τη διαδικασία εκμάθησης και βελτιστοποίηση αλλάζοντας την αρχιτεκτονική του δικτύου (νευρώνες, στρώμα κ.λπ.), τις παραμέτρους (ρυθμός εκμάθησης, μέγεθος παρτίδας, ορμή κ.λπ.) και τη διαμόρφωση (βελτιστοποιητής συνάρτησης απώλειας) ή το σύνολο εκπαίδευσης.
5. **Δοκιμή Iris MLP.** Δοκιμάστε το μοντέλο χρησιμοποιώντας το σύνολο δοκιμών. Για την αξιολόγηση της απόδοσης του μοντέλου, δοκιμάζουμε το μοντέλο χρησιμοποιώντας την `model.predict()` στο σετ ελέγχου και λαμβάνουμε λίστα με τις προβλέψεις του MLP. Μετατρέπουμε τις προβλέψεις σε κατηγορικές τιμές από την κωδικοποίηση ενός σημείου (`np.argmax()`) και, στη συνέχεια, εκτυπώνουμε την αναφορά ταξινόμησης και τον πίνακα σύγχυσης. Η `model.evaluate()` επιστρέφει τις τιμές των μετρικών απώλειας και ακρίβειας στο σύνολο δοκιμής που χρησιμοποιήθηκε για την αξιολόγηση του MLP. Το συγκεκριμένο MLP πέτυχε 100% ακρίβεια στην πρόβλεψη όλων των ειδών ίριδας. Το συγκεκριμένο MLP πέτυχε 100% ακρίβεια στην πρόβλεψη όλων των ειδών ίριδας. Αυτό το MLP της ίριδας δεν χρειάζεται περαιτέρω βελτιστοποίηση και είναι έτοιμο να ενσωματωθεί σε μια συγκεκριμένη εφαρμογή. Εάν το μοντέλο δεν επιτυγχάνει τον στόχο του έργου, βελτιστοποιήστε το αλλάζοντας την αρχιτεκτονική, τη διαμόρφωση, τις παραμέτρους ή τα δεδομένα εκπαίδευσης.
6. **Αποθήκευση του MLP.** Τώρα το μοντέλο είναι έτοιμο για μετέπειτα επαναχρησιμοποίηση. Με τη χρήση του πακέτου `randas` αποθηκεύεται το ιστορικό εκπαίδευσης σε ένα αρχείο `csv`. Χρησιμοποιώντας το `json` αποθηκεύετε την αρχιτεκτονική του δικτύου σε ένα αρχείο `json`. Τα εκπαιδευμένα βάρη του μοντέλου μπορούν να αποθηκευτούν σε ένα αρχείο `h5` χρησιμοποιώντας την ενσωματωμένη μέθοδο στην κλάση `keras.model`

Περιπτώσεις χρήσης

ΕΚΠΑΙΔΕΥΣΗ MLP

Αυτή η περίπτωση χρήσης θα εξηγήσει πώς να ξεκινήσετε και να εκπαιδεύσετε ένα MLP χρησιμοποιώντας ένα σύνολο δεδομένων που βασίζεται σε πραγματικά δεδομένα για την επίλυση μιας εργασίας παλινδρόμησης. Συγκεκριμένα, το MLP θα εκπαιδευτεί χρησιμοποιώντας ένα σύνολο εκπαίδευσης που περιέχει δεδομένα σχετικά με διαφορετικά χαρακτηριστικά σπιτιών και τη σχετική τιμολόγηση. Το MLP που θα προκύψει από την εκπαίδευση θα μπορούσε να χρησιμοποιηθεί για την ανάπτυξη μιας καινοτόμου εφαρμογής για την αγορά ακινήτων.

[ΚΑΤΕΒΑΣΤΕ ΤΟΝ ΚΩΔΙΚΑ ΕΔΩ](#)

Βήμα 1: Φόρτωση συνόλου δεδομένων τιμών κατοικιών της Βοστώνης

Η ενότητα `Scikit-learn.datasets` περιέχει διάφορα σύνολα δεδομένων που χρησιμοποιούνται για την εκπαίδευση αλγορίθμων ML. Για αυτή την περίπτωση χρήσης φορτώστε το σύνολο δεδομένων για κατοικίες της Βοστώνης. Αυτό το σύνολο δεδομένων περιέχει δεδομένα σχετικά με τα χαρακτηριστικά και τις τιμές των κατοικιών στην αγορά ακινήτων της Βοστώνης. Το Boston housing περιέχει 506 παραδείγματα εκπαίδευσης τιμών κατοικιών που σχετίζονται με 13 διαφορετικά χαρακτηριστικά των κατοικιών στην πόλη της Βοστώνης.

Βήμα 2: Ανάλυση του συνόλου δεδομένων για τις τιμές της Βοστώνης

Για την εκπαίδευση του MLP για την πρόβλεψη των τιμών των κατοικιών στη Βοστώνη χρησιμοποιούνται όλα τα διαφορετικά χαρακτηριστικά του συνόλου δεδομένων: Μεταβλητή Charles river, Συγκέντρωση οξειδίου του αζώτου, Ιδιοκτησία πριν από το 1940, Απόσταση από κέντρα απασχόλησης, Προσβασιμότητα σε αυτοκινητόδρομους, Πλήρεις αξίες φόρου ακίνητης περιουσίας, Αναλογία μαθητών/δασκάλων, Ποσοστό μαύρων, Κατάσταση του πληθυσμού, Μέσες τιμές ιδιοκατοίκησης.

Βήμα 3: Αρχικοποίηση MLP αρχιτεκτονικής

Αρχικοποιήστε ένα MLP με ένα στρώμα εισόδου (13 είσοδοι) και ένα κρυφό στρώμα οκτώ μονάδων με συνάρτηση ενεργοποίησης ReLU και στρώμα εξόδου με μία μονάδα εξόδου με γραμμική συνάρτηση ενεργοποίησης. Η `Models.Sequential()` επιτρέπει την αρχικοποίηση ενός νέου μοντέλου. Κάθε στρώμα προστίθεται διαδοχικά στο προηγούμενο. `model.add()`: προσθήκη ενός νέου στρώματος (στρώμα εξόδου) στο μοντέλο. `layers.Dense()`: ένα στρώμα όπου κάθε νευρώνας έχει συνδέσεις με όλους τους νευρώνες του προηγούμενου στρώματος.

Βήμα 4: Προεπεξεργασία συνόλου δεδομένων Iris

Όπως τα ANN, έτσι και τα DNN χρειάζονται είσοδο κλιμακωτή στο εύρος (0-1). Είναι δυνατή η κλιμάκωση της εισόδου σε αυτό το εύρος με τη χρήση της κλάσης `MinMaxScaler` της μονάδας προεπεξεργασίας του πακέτου `sklearn`. Μέσω της συνάρτησης `train_test_split` στην ενότητα `sklearn.model_selection`, είναι δυνατή η πραγματοποίηση υποδιαίρεσης σε ένα σύνολο εκπαίδευσης και δοκιμής γρήγορα, τυχαία και ταυτόχρονα τόσο στα διανύσματα εισόδου όσο και στα επιθυμητά διανύσματα εξόδου, διατηρώντας τη σχέση μεταξύ των διανυσμάτων στα παραδείγματα εκπαίδευσης. Είναι δυνατό να καθορίσετε την κατάσταση `random_state` στο `train_test_split` για να αναπαράγετε τα αποτελέσματα.

Βήμα 5: Μεταγλώττιση και εκπαίδευση MLP

Η `model.compile()` διαμορφώνει το μοντέλο για τη φάση εκπαίδευσης. Οι συναρτήσεις απώλειας, οι βελτιστοποιητές (με ρυθμό μάθησης) και οι μετρικές ακρίβειας μπορούν εύκολα να διαμορφωθούν σε αυτό το βήμα. `model.summary()` εκτύπωση στην κονσόλα της αρχιτεκτονικής του δικτύου με τον αριθμό των παραμέτρων που μπορούν να εκπαιδευτούν. `model.fit()` έναρξη της φάσης εκπαίδευσης. Οι εποχές εκπαίδευσης, το μέγεθος της παρτίδας, τα παραδείγματα εκπαίδευσης και τα παραδείγματα επικύρωσης μπορούν να οριστούν εδώ. Τα δεδομένα επικύρωσης αντιπροσωπεύουν ένα μέρος των δεδομένων εκπαίδευσης που χρησιμοποιούνται για τη δοκιμή του δικτύου κατά τη φάση της εκπαίδευσης. Είναι χρήσιμο να επισημανθεί τότε το δίκτυο χάνει την ιδιότητα γενίκευσης (overfitting). Συνήθως, η επικύρωση αντιπροσωπεύει το 20% των δεδομένων εκπαίδευσης. Στην περίπτωση αυτή, τα δεδομένα δοκιμής χρησιμοποιούνται ως δεδομένα επικύρωσης.

Βήμα 6: Αξιολόγηση της κατάρτισης

Σχεδιάστε το ιστορικό εκπαίδευσης του μοντέλου για να εντοπίσετε τυχόν προβλήματα που προέκυψαν κατά τη διαδικασία εκμάθησης και να το βελτιστοποιήσετε αλλάζοντας την αρχιτεκτονική του δικτύου (νευρώνες, στρώμα κ.λπ.), τις παραμέτρους (ρυθμός εκμάθησης, μέγεθος παρτίδας, ορμή κ.λπ.) και τη διαμόρφωση (βελτιστοποιητής συνάρτησης απώλειας) ή το σύνολο εκπαίδευσης.

Βήμα 7: Δοκιμή MLP

`model.evaluate()` υπολογίζει την απόδοση του μοντέλου ως απόλυτο ποσοστιαίο σφάλμα των δοκιμαστικών προβλέψεων.

Βήμα 8: Αποθήκευση MLP

Τώρα το μοντέλο είναι έτοιμο για μεταγενέστερη επαναχρησιμοποίηση σε εφαρμογές της αγοράς ακινήτων. Η χρήση του πακέτου `randas` αποθηκεύει το ιστορικό εκπαίδευσης σε ένα αρχείο `csv`. Χρησιμοποιώντας το `json` αποθηκεύει την αρχιτεκτονική του δικτύου σε ένα αρχείο `json`. Τα εκπαιδευμένα βάρη μπορούν να αποθηκευτούν σε ένα αρχείο `h5` χρησιμοποιώντας την ενσωματωμένη μέθοδο στην κλάση `keras.model`.

Μάθημα 3.4: Βαθιά μάθηση για ταξινόμηση εικόνων

Θέματα προς συζήτηση

1. Συνελκτικά νευρωνικά δίκτυα (CNN)
2. CNNs για την αναγνώριση χειρόγραφων ψηφίων

Σημειώσεις διάλεξης

Όραση υπολογιστών (CV). Ένα από τα πεδία εφαρμογών όπου η τεχνητή νοημοσύνη σημειώνει μεγάλη επιτυχία είναι αναμφίβολα η όραση υπολογιστών. Αυτό το επιστημονικό πεδίο αξιοποιεί ένα σύνολο ισχυρά διεπιστημονικών τεχνικών που επιτρέπουν στους υπολογιστές να εκτελούν υψηλού επιπέδου επεξεργασία εικόνων και βίντεο-καρέ, τυπικές λειτουργίες των βιολογικών οπτικών συστημάτων.

Δουλειά που έχει επιτευχθεί. Μέσω της χρήσης καμερών είναι δυνατή η λήψη εικόνων από το περιβάλλον. Επί του παρόντος, οι τεχνικές υπολογιστικής όρασης επιτρέπουν την ανάλυση εικόνων με διαφορετικές λειτουργίες:

- Ταξινόμηση εικόνας: Σε αυτήν την εργασία, οι υπολογιστές πρέπει να καταλάβουν εάν μια εικόνα περιέχει ένα συγκεκριμένο αντικείμενο χωρίς να δίνουν πληροφορίες σχετικά με τη θέση του μέσα στην εικόνα ή ποια είναι τα pixel της εικόνας που τη συνθέτουν
- Ανίχνευση αντικειμένου: Σε αυτήν την εργασία, οι υπολογιστές πρέπει να προβλέψουν τις συντεταγμένες κάθε ορθογωνίου (οριοθετημένο πλαίσιο) που περιέχει το αντικείμενο ενδιαφέροντος.
- Τμηματοποίηση αντικειμένων: Σε αυτήν την εργασία, εκτός από τα πλαίσια οριοθέτησης, οι υπολογιστές πρέπει να προβλέψουν τη μάσκα: το σύνολο όλων των pixel που ανήκουν στο αντικείμενο ενδιαφέροντος

Ανθρώπινο οπτικό σύστημα. Το ανθρώπινο οπτικό σύστημα εκτελεί όλες τις εργασίες που επιτυγχάνονται από την όραση των υπολογιστών. Το σήμα διαδίδεται και συγκλίνει στις διάφορες βιολογικές δομές που συνθέτουν το σύστημα:

- Μάτια: αποκτούν οπτικές πληροφορίες από το περιβάλλον (ισοδύναμο με κάμερα υπολογιστή).
- LGN και οπτικός φλοιός: εξαγωγή χαρακτηριστικών της εικόνας (αντίθεση, κορεσμός, φωτεινότητα, περιγράμματα, γραμμές, διαφορές στην υφή)
- Ανώτερος φλοιός: συνδυάζει χαρακτηριστικά για την αναγνώριση αντικειμένων

Το ανθρώπινο οπτικό σύστημα αποτέλεσε πηγή έμπνευσης για την αρχιτεκτονική των νευρωνικών δικτύων με συνελκτικό τρόπο (CNN). Αυτά τα DNN αντιπροσωπεύουν τις πιο σύγχρονες αρχιτεκτονικές που χρησιμοποιούνται για την ανάλυση εικόνων.

Συνελικτικά νευρωνικά δίκτυα (CNN). Η αρχιτεκτονική του βιολογικού οπτικού συστήματος και η σύγκλιση των σημάτων που παρατηρείται σε αυτά τα νευρωνικά συστήματα ενέπνευσαν την αρχιτεκτονική των CNN. Σε αντίθεση με τα MLP, όπου όλοι οι νευρώνες ενός επιπέδου συνδέονται με όλους τους νευρώνες που αποτελούν το προηγούμενο επίπεδο (fully connected layers, FC). Τα CNN αποτελούνται από δύο διαφορετικές περιοχές: την περιοχή συνελίξεων και την περιοχή ταξινόμησης. Η περιοχή συνελίξεων περιέχει δύο νέους τύπους στρώματων: τα στρώματα συνελίξεων και τα στρώματα συγκέντρωσης. Τα στρώματα συνελικτικού τύπου εκτελούν αυτόματη εξαγωγή των χαρακτηριστικών που περιέχονται στις εικόνες, ενώ τα στρώματα συγκέντρωσης επιτρέπουν τη μείωση του μεγέθους του σήματος. Ένα MLP αντιπροσωπεύει την περιοχή ταξινόμησης. Ταξινομεί τα χαρακτηριστικά που εξάγονται από την περιοχή συμβολής για την αναγνώριση της παρουσίας ενός αντικειμένου εντός της εικόνας.

CNNs είσοδος: εικόνες. Το στρώμα εισόδου των CNNs λαμβάνει τις εικόνες ως πίνακα με τις ακόλουθες διαστάσεις: [πλάτος, ύψος, βάθος]. Το πλάτος και το ύψος αντιπροσωπεύουν την ανάλυση της εικόνας και συνήθως έχουν τις ίδιες τιμές (τετραγωνισμένη εικόνα). Το βάθος αντιπροσωπεύει τον αριθμό των χρωμάτων της εικόνας. Η διάσταση της εικόνας και κατά συνέπεια το στρώμα εισόδου πρέπει να διαιρείται πολλές φορές με το 2. (Οι συνήθεις αριθμοί περιλαμβάνουν 32, 64, 96, 224, 384, 512 και 1024).

Περιοχή συνένωσης: στρώματα συνένωσης. Τα στρώματα συνελικτικής ανάλυσης εξάγουν τα κρυμμένα χαρακτηριστικά που περιέχονται στις εικόνες, συνελικτικά σήματα μεταξύ διαφορετικών στρώματων. Κάθε συνελικτικό στρώμα χρησιμοποιεί ένα σύνολο φίλτρων (ή πυρήνες ή ανιχνευτές χαρακτηριστικών). Κάθε φίλτρο υπολογίζει ένα τετραγωνικό γινόμενο μεταξύ των συναπτικών του βαρών και μιας μικρής περιοχής του σήματος εισόδου (δεκτικό πεδίο του φίλτρου). Επιπλέον, κάθε φίλτρο ολισθαίνει στην είσοδο κατά μια συγκεκριμένη τιμή που ονομάζεται stride. Το βάθος καθορίζει τον αριθμό των φίλτρων που χρησιμοποιούνται για τη συνέλιξη του σήματος. Ο χάρτης χαρακτηριστικών (ή χάρτης ενεργοποίησης ή όγκος εξόδου) είναι η έξοδος από κάθε φίλτρο ολίσθησης πάνω στην εικόνα. Ο αριθμός των χαρτών χαρακτηριστικών που εξάγονται από την είσοδο εξαρτάται από τον αριθμό των φίλτρων που χρησιμοποιούνται για τη συνέλιξη της εισόδου. Για να αποφευχθεί η απώλεια της πληροφορίας που περιέχεται στα όρια της εισόδου, συνηθίζεται η συμπλήρωση μηδενικών: προσθήκη μηδενικών γύρω από τα άκρα της εικόνας. Η συνάρτηση ενεργοποίησης που δρα στους χάρτες χαρακτηριστικών που παράγονται από τη συνέλιξη είναι η διορθωμένη γραμμική μονάδα (ReLU).

Γραμμικής διορθωτική μονάδα Rectified linear unit (ReLU). Συνάρτηση ενεργοποίησης που εφαρμόζεται σε χάρτες χαρακτηριστικών. Η ReLU μετατρέπει τις αρνητικές τιμές σε μηδενικές, ενώ οι θετικές τιμές προκαλούν γραμμική ενεργοποίηση. Συνήθως δεν έχει διαμορφώσιμες παραμέτρους. Σε σύγκριση με τις σιγμοειδείς συναρτήσεις επιτρέπουν ταχύτερη και αποτελεσματική εκπαίδευση με μεγάλα σύνολα δεδομένων. Διαφορετικές παραλλαγές:

- Θορυβώδεις ReLUs → επιτυχία RBM
- Διαρρέουσες ReLUs → μικρή θετική κλίση
- Παραμετρικές ReLUs → τροποποιήσιμη θετική κλίση προσθέτοντας παραμέτρους

Η ReLU έχει διαφορετικά πλεονεκτήματα σε σύγκριση με τις άλλες συναρτήσεις ενεργοποίησης, επειδή έχει μεγαλύτερη βιολογική αληθοφάνεια προσομοιώνοντας την οπτική δραστηριότητα και επιτρέπει καλύτερη διάδοση της κλίσης.

Σπειροειδής περιοχή Convolutional region: Pooling layers.: Συγκεντρωτικά στρώματα. Τα στρώματα συγκέντρωσης μειώνουν τη χωρική διάσταση της εισόδου (υποδειγματοληψία), καθιστώντας τον υπολογιστικό υπολογισμό ελαφρύτερο και μειώνοντας τα προβλήματα υπερπροσαρμογής. Παρεμβάλλονται περιοδικά μεταξύ των στρωμάτων συμβολής. Δεν είναι υποχρεωτικά, καθώς μπορεί να προκαλέσουν απώλεια πληροφοριών. Τα στρώματα συγκέντρωσης δεν έχουν μηδενική επικάλυψη αλλά έχουν φίλτρα και stride (συνήθως 2x2, τα μεγαλύτερα φίλτρα μειώνουν τις επιδόσεις). Διαφορετικές παραλλαγές:

- Max pooling: επιστρέφει τη μέγιστη τιμή όλων των τιμών που περιέχονται στο φίλτρο (χρησιμοποιείται περισσότερο)
- Avg pooling: επιστρέφει τη μέση τιμή όλων των τιμών που περιέχονται στο φίλτρο.

Περιοχή ταξινόμησης Classification region: MLP. Οι χάρτες χαρακτηριστικών που υπολογίζονται από την περιοχή συμβολής ισοπεδώνονται σε ένα γραμμικό διάνυσμα. Αυτό το σήμα αποτελεί την είσοδο για την επόμενη περιοχή των CNN: ένα MLP με το πλήρως συνδεδεμένο στρώμα (FC layer). Κάθε νευρώνας σε ένα στρώμα του MLP έχει σύνδεση με όλες τις μονάδες του προηγούμενου στρώματος. Ο αριθμός των μονάδων εξόδου είναι ίσος με τον αριθμό των κλάσεων αντικειμένων που μπορεί να αναγνωρίσει το δίκτυο. Οι μονάδες εξόδου χρησιμοποιούν τη softmax ως συνάρτηση ενεργοποίησης για την κανονικοποίηση της εξόδου σε κατανομή πιθανότητας (εργασίες ταξινόμησης πολλαπλών κατηγοριών) ή συνάρτηση Heaviside (εργασία δυαδικής ταξινόμησης).

CNN για την αναγνώριση χειρόγραφων ψηφίων. Αυτό το μέρος του μαθήματος εξηγεί πώς να χρησιμοποιήσετε το Keras για να ξεκινήσετε και να εκπαιδεύσετε ένα CNN για την αναγνώριση χειρόγραφων ψηφίων σε εικόνες.

LeNet5. Το LeNet5 ήταν ένα από τα πρώτα CNN που επινοήθηκε από τους Yann LeCun et al. το 1998 για την αναγνώριση χειρόγραφων ψηφίων. Αποτελείται από επτά διαφορετικά επίπεδα: Τα C1 και C3 είναι συνελκτικά στρώματα (6 φίλτρα = [5,5]), τα S2 και S4 είναι στρώματα συγκέντρωσης του μέσου όρου, ενώ τα F5, F6, F7 είναι τα πλήρως συνδεδεμένα στρώματα και αντιπροσωπεύουν την περιοχή ταξινόμησης. Στην είσοδο των στρωμάτων F5 και F6 εφαρμόζεται διακοπή. Το dropout θέτει τυχαία ένα ποσοστό κλάσματος των μονάδων εισόδου στο 0 μειώνοντας την υπερπροσαρμογή. Το F7 είναι το στρώμα εξόδου και έχει αριθμό μονάδων εξόδου ίσο με τις αναγνωρίσιμες κλάσεις.

Σύνολο δεδομένων MNIST. Το MNIST αντιπροσωπεύει ένα σύνολο δεδομένων με εικόνες που περιέχουν χειρόγραφα ψηφία. Αποτελεί υποσύνολο του μεγαλύτερου συνόλου δεδομένων NIST. Κάθε ψηφίο έχει κανονικοποιηθεί ως προς το μέγεθος και έχει κεντραριστεί σε μια εικόνα σταθερού μεγέθους (28x28x1). Αυτό το σύνολο δεδομένων αποτελεί σημείο αναφοράς για την αξιολόγηση της απόδοσης διαφορετικών αρχιτεκτονικών DNN και άλλων αλγορίθμων ML. Η

ενότητα datasets της Keras περιέχει το σύνολο δεδομένων MNIST. Η τεκμηρίωση του MNIST μπορεί να θεμελιωθεί εδώ: <http://yann.lecun.com/exdb/mnist/>

1. **Φόρτωση MNIST.** Εισάγετε το MNIST από την ενότητα Keras.datasets και δημιουργήστε το σύνολο εκπαίδευσης του δικτύου. `mnist.load_data()` σας επιτρέπει να θέσετε σε λειτουργία το σύνολο εκπαίδευσης και το σύνολο δοκιμής. Το `Train_images` περιέχει 60000 NumPy πίνακες που σχετίζονται με τις εικόνες των χειρόγραφων ψηφίων για την εκπαίδευση του δικτύου. Το `Train_labels` περιέχει την ετικέτα κάθε εικόνας και αντιπροσωπεύει την επιθυμητή έξοδο του δικτύου κατά τη φάση της εκπαίδευσης. Το `Test_images` περιέχει 10000 NumPy πίνακες που σχετίζονται με τις εικόνες των χειρόγραφων ψηφίων για τη δοκιμή του δικτύου. Το `Train_labels` περιέχει την ετικέτα κάθε εικόνας και αντιπροσωπεύει τη σωστή απάντηση κατά τη φάση δοκιμής.
2. **Ανάλυση του MNIST.** Χρησιμοποιώντας το `matplotlib` παρουσιάζει τις εικόνες MNIST που περιέχουν τα χειρόγραφα ψηφία και τις σχετικές ετικέτες.
3. **Εκκίνηση της περιοχής συνελκτικού δικτύου LeNet5.** Ξεκινήστε να δημιουργείτε τη συνελκτική περιοχή της αρχιτεκτονικής LeNet5: δύο συνελκτικά στρώματα και δύο στρώματα `max-pooling` για την εξαγωγή των χαρακτηριστικών. `model.Sequential()` επιτρέπει την αρχικοποίηση μιας νέας λειτουργίας όπου κάθε στρώμα προστίθεται διαδοχικά. `model.add()` επιτρέπει την προσθήκη ενός νέου στρώματος στο μοντέλο. Η ενότητα `Keras.layer` περιέχει διάφορα προκατασκευασμένα στρώματα ANN. `layers.Conv2D()` είναι ένα δισδιάστατο συνελκτικό στρώμα το οποίο μπορεί να αρχικοποιηθεί με διαφορετικές παραμέτρους. Τα συγκεφαλαιωτικά στρώματα χρησιμοποιούν την ReLU ως συνάρτηση ενεργοποίησης. `layers.MaxPooling2D()` είναι ένα στρώμα μέγιστης συγκέντρωσης το οποίο μπορεί να αρχικοποιηθεί με διαφορετικές παραμέτρους.
4. **Έναρξη της περιοχής ταξινόμησης LeNet5.** Τώρα προσθέστε την περιοχή ταξινόμησης. Αρχικά, η `layer.flatten()` επιτρέπει τη μετατροπή ενός δισδιάστατου πίνακα σε ένα γραμμικό διάνυσμα το οποίο μπορεί να χρησιμοποιηθεί ως είσοδος για το πρώτο στρώμα FC. Επιπλέον, προσθέστε ένα στρώμα εγκατάλειψης χρησιμοποιώντας τη μέθοδο `layer.dropout()`. Το `dropout` θέτει τυχαία ένα κλάσμα της εισόδου στο μηδέν. Τώρα προσθέστε το πρώτο στρώμα FC χρησιμοποιώντας τη μέθοδο `layers.Dense()`. Προσθέστε ένα ακόμη `dropout` και στη συνέχεια προσθέστε το δεύτερο και το τρίτο στρώμα στρώματος FC (στρώμα εξόδου). Το πρώτο και το δεύτερο FC-στρώμα έχουν ReLU ως συνάρτηση ενεργοποίησης ενώ το τρίτο FC-στρώμα είναι το στρώμα εξόδου και έχει Softmax ως συνάρτηση ενεργοποίησης (ταξινόμηση πολλαπλών κατηγοριών). Ο αριθμός των μονάδων εξόδου καθορίζει πόσα διαφορετικά αντικείμενα (κλάσεις) μπορεί να αναγνωρίσει το LeNet5.
5. **Επεξεργασία MNIST.** Οι εικόνες πρέπει να αναβαθμιστούν από το εύρος (0-255) στο εύρος (0,1). Οι ετικέτες των εικόνων πρέπει να μετατραπούν από τις κατηγορηματικές κλάσεις σε μορφή κωδικοποίησης ενός σημείου. Το MNIST περιέχει δέκα διαφορετικά χειρόγραφα ψηφία. Ένα CNN ικανό να αναγνωρίσει όλες τις κλάσεις στο σύνολο δεδομένων MNIST έχει δέκα νευρώνες εξόδου, έναν για κάθε ψηφίο. Παράδειγμα: μετατροπή ετικέτας "5" = [0,0,0,0,0,0,0,1,0,0,0,0,0], η έκτη μονάδα εξόδου γίνεται ενεργή.

6. Μεταγλώττιση και εκπαίδευση του LeNet5. `model.compile()` ρυθμίστε το μοντέλο για τη φάση εκπαίδευσης. Οι βελτιστοποιητές, οι συναρτήσεις απωλειών και οι μετρικές εκπαίδευσης μπορούν να επιλεγούν προ-κατασκευασμένες μονάδες Keras ή μπορούν να προσαρμοστούν. `model.summary()` εκτύπωση στην οθόνη μιας σύνοψης της αρχιτεκτονικής του μοντέλου. `model.fit()` έναρξη της φάσης εκπαίδευσης. Εδώ μπορούν να οριστούν η εποχή εκπαίδευσης και το μέγεθος δέσμης (αριθμός εικόνων που λαμβάνονται ταυτόχρονα από την CPU ή την GPU). Ο διαχωρισμός επικύρωσης χωρίζει τα δεδομένα εκπαίδευσης σε δύο υποσύνολα:
 - i. Δεδομένα εκπαίδευσης: χρησιμοποιούνται για την εκπαίδευση του δικτύου.
 - ii. Δεδομένα επικύρωσης: χρησιμοποιούνται για τη δοκιμή του μοντέλου κατά τη φάση της εκπαίδευσης. Χρήσιμο για την επισήμανση προβλημάτων υπερπροσαρμογής. Συνήθως, η επικύρωση αντιπροσωπεύει το 20% των δεδομένων εκπαίδευσης.
7. **Αξιολόγηση της εκπαίδευσης.** Σχεδίαση του ιστορικό εκπαίδευσης του μοντέλου σε σχέση με την εκπαίδευση και την επικύρωση (ακρίβεια και απώλεια).
8. **Δοκιμή MLP.** Δοκιμή του μοντέλου χρησιμοποιώντας το σύνολο δοκιμών. Για να αξιολογήσουμε την απόδοση του μοντέλου, δοκιμάζουμε το μοντέλο χρησιμοποιώντας την `model.predict()` στο σύνολο δοκιμών και λάμβάνουμε μια λίστα με τις προβλέψεις του LeNet5. `np.argmax()` Μετατρέπουμε την έξοδο σε κατηγορικές τιμές από την κωδικοποίηση one-hot () και στη συνέχεια εκτυπώστε την αναφορά ταξινόμησης και τον πίνακα σύγχυσης. Η `model.evaluate()` επιστρέφει τις τιμές των μετρικών απώλειας και ακρίβειας στο σύνολο δοκιμών που χρησιμοποιήθηκε για την αξιολόγηση του LeNet5. Το LeNet5 πέτυχε ακρίβεια 98% στην πρόβλεψη των ψηφίων στο σύνολο δεδομένων MNIST.
9. **Αποθήκευση Lenet5.** Αφού το μοντέλο εκπαιδευτεί για την αναγνώριση χειρόγραφων ψηφίων, μπορεί να αποθηκευτεί για μελλοντική χρήση σε έργα που περιλαμβάνουν ανάγνωση χειρόγραφων ψηφίων. Η `model.to_json()` επιτρέπει τη μετατροπή της αρχιτεκτονικής του μοντέλου σε μορφή .json, ενώ η `model.save_weights()` επιτρέπει την αποθήκευση των συναπτικών βαρών που προέκυψαν μετά τη διαδικασία εκπαίδευσης σε μορφή h5.
10. **Φόρτωση LeNet5.** Το προ-εκπαιδευμένο μοντέλο μπορεί να χρησιμοποιηθεί σε άλλα έργα ως συνάρτηση ικανή να επεξεργάζεται τις εισόδους που της παρέχονται. `model_from_json()` επιτρέπει τη φόρτωση προ-εκπαιδευμένων αρχιτεκτονικών μοντέλων αποθηκευμένων σε μορφή json. Η `model.load_weights()` επιτρέπει τη χρήση των βαρών στο μοντέλο που είναι αποτέλεσμα της εκπαίδευσης με ένα συγκεκριμένο σύνολο δεδομένων.

Περιπτώσεις χρήσης

ΕΚΠΑΙΔΕΥΣΗ LENET5

Σε αυτή την περίπτωση χρήσης, το LeNet5 θα εκπαιδευτεί χρησιμοποιώντας ένα άλλο σύνολο δεδομένων έγχρωμων εικόνων που χρησιμοποιείται ως σημείο αναφοράς αξιολόγησης: Cifar10. Το αποτέλεσμα είναι ένα CNN ικανό να αναγνωρίζει τα διάφορα αντικείμενα ενδιαφέροντος που

περιέχονται στο Cifar10. Το εκπαιδευμένο LeNet5 θα μπορούσε να επαναχρησιμοποιηθεί για την ανάπτυξη μιας εφαρμογής που ταξινομεί διάφορες εικόνες.

ΚΑΤΕΒΑΣΤΕ ΤΟΝ ΚΩΔΙΚΑ ΕΛΩ

Βήμα 1: Cifar10

Το LeNet5 θα εκπαιδευτεί και θα αξιολογηθεί χρησιμοποιώντας το cifar10, ένα σύνολο δεδομένων με ετικέτες από μικρές έγχρωμες εικόνες που περιέχεται στην ενότητα Keras.datasets. Το cifar10 περιέχει 50.000 εικόνες εκπαίδευσης [32,32,3] και 10.000 εικόνες δοκιμής [32,32,3]. Οι κλάσεις είναι εντελώς αμοιβαία αποκλειόμενες χωρίς να υπάρχει επικάλυψη μεταξύ των κλάσεων. Το Cifar10 χρησιμοποιείται συνήθως ως σημείο αναφοράς για την αξιολόγηση διαφόρων μοντέλων <https://www.cs.toronto.edu/~kriz/cifar.html>

Βήμα 2: Φόρτωση cifar10

Εισαγωγή του συνόλου δεδομένων μικρών εικόνων cifar10 από την ενότητα keras.datasets. `cifar10.load_data()` επιστρέφει :50000 [32x32x3] εικόνες εκπαίδευσης επισημασμένες σε 10 κλάσεις αντικειμένων. 10000 [32x32x3] εικόνες δοκιμής επισημασμένες στις ίδιες κλάσεις αντικειμένων. `cifar10_class_names` περιέχει όλα τα ονόματα των κλάσεων με δείκτη όπως στο σύνολο δεδομένων. [αεροπλάνο, αυτοκίνητο, πουλί, γάτα, ελάφι, σκύλος, βάτραχος, άλογο, πλοίο, φορτηγό].

Βήμα 3: Ανάλυση Cifar10

Σχεδιάστε μερικές εικόνες εκπαίδευσης με σχετικές ετικέτες.

Βήμα 4: Αρχικοποίηση συνελικτικής περιοχής LeNet5

Ξεκινήστε να δημιουργείτε τη συνελικτική περιοχή της αρχιτεκτονικής LeNet5: δύο συνελικτικά στρώματα και δύο στρώματα max-pooling για την εξαγωγή των χαρακτηριστικών. `model.Sequential()` επιτρέπει την αρχικοποίηση μιας νέας λειτουργίας όπου κάθε στρώμα προστίθεται διαδοχικά. `model.add()` επιτρέπει την προσθήκη ενός νέου στρώματος στο μοντέλο. Η ενότητα Keras.layer περιέχει διάφορα προ-δημιουργημένα επίπεδα ANN. `layers.Conv2D()` είναι ένα δισδιάστατο συγκεχυμένο επίπεδο που μπορεί να αρχικοποιηθεί με διαφορετικές παραμέτρους. Τα συνελικτικά στρώματα χρησιμοποιούν την ReLU ως συνάρτηση ενεργοποίησης. `layers.MaxPooling2D()` είναι ένα στρώμα μέγιστης συγκέντρωσης που μπορεί να αρχικοποιηθεί με διαφορετικές παραμέτρους.

Βήμα 5: Αρχικοποίηση περιοχής ταξινόμησης LeNet5

Τώρα προσθέστε την περιοχή ταξινόμησης. Αρχικά, η `layer.Flatten()` επιτρέπει τη μετατροπή ενός δισδιάστατου πίνακα σε ένα γραμμικό διάνυσμα το οποίο μπορεί να χρησιμοποιηθεί ως είσοδος για το πρώτο στρώμα FC. Επιπλέον, προσθέστε ένα στρώμα απόρριψης χρησιμοποιώντας τη μέθοδο `layer.Dropout()`. Το dropout θέτει τυχαία ένα κλάσμα της εισόδου στο μηδέν. Τώρα προσθέστε το πρώτο στρώμα FC χρησιμοποιώντας τη μέθοδο `layers.Dense()`. Προσθέστε ένα

ακόμη dropout και στη συνέχεια προσθέστε το δεύτερο και το τρίτο στρώμα FC (στρώμα εξόδου). Το πρώτο και το δεύτερο FC-στρώμα έχουν ReLU ως συνάρτηση ενεργοποίησης ενώ το τρίτο FC-στρώμα είναι το στρώμα εξόδου και έχει Softmax ως συνάρτηση ενεργοποίησης (ταξινόμηση πολλαπλών κατηγοριών). Ο αριθμός των μονάδων εξόδου καθορίζει πόσα διαφορετικά αντικείμενα (κλάσεις) μπορεί να αναγνωρίσει το LeNet5.

Βήμα 6: Προεπεξεργασία cifar10

Επανακλιμάκωση των τιμών του πίνακα εισόδου από το εύρος [0,255] στο εύρος [0,1]. Μετατρέψτε τις ετικέτες των εικόνων στην επιθυμητή έξοδο του στρώματος εξόδου του CNN (κωδικοποίηση με ένα σημείο). Το Cifar10 περιέχει δέκα διαφορετικές κλάσεις αντικειμένων. το Lenet5 έχει δέκα νευρώνες εξόδου, έναν για κάθε κλάση. παράδειγμα: κλάση 1 = "αεροπλάνο", μετατροπή = [1,0,0,0,0,0,0,0,0,0], η πρώτη μονάδα εξόδου γίνεται ενεργή.

Βήμα 7: Μεταγλώττιση και εκπαίδευση του LeNet5

model.compile() διαμορφώνει το μοντέλο για τη φάση εκπαίδευσης. Οι βελτιστοποιητές, οι συναρτήσεις απωλειών και οι μετρικές εκπαίδευσης μπορούν να επιλεγούν προ-κατασκευασμένες μονάδες Keras ή να προσαρμοστούν. model.summary() εκτυπώνει στην οθόνη μια σύνοψη της αρχιτεκτονικής του μοντέλου. model.fit() ξεκινά τη φάση εκπαίδευσης. Εδώ μπορούν να οριστούν η εποχή εκπαίδευσης και το μέγεθος παρτίδας (αριθμός εικόνων που λαμβάνονται ταυτόχρονα από την CPU ή την GPU). Ο διαχωρισμός επικύρωσης χωρίζει τα δεδομένα εκπαίδευσης σε δύο υποσύνολα:

- Δεδομένα εκπαίδευσης: χρησιμοποιούνται για την εκπαίδευση του δικτύου.
- Δεδομένα επικύρωσης: χρησιμοποιούνται για τη δοκιμή του μοντέλου κατά τη φάση της εκπαίδευσης. Χρήσιμο για την επισήμανση προβλημάτων υπερπροσαρμογής. Συνήθως, η επικύρωση αντιπροσωπεύει το 20% των δεδομένων εκπαίδευσης.

Βήμα 8: Αξιολόγηση της εκπαίδευσης

Σχεδιάστε το ιστορικό εκπαίδευσης του μοντέλου όσον αφορά την εκπαίδευση και την επικύρωση (ακρίβεια και απώλεια). Το μοντέλο δεν παρουσιάζει προβλήματα υπερπροσαρμογής

Βήμα 9: Δοκιμή LeNet5-Cifar10

Δοκιμάστε το μοντέλο χρησιμοποιώντας το σύνολο δοκιμών. Για να αξιολογήσετε την απόδοση του μοντέλου, δοκιμάστε το μοντέλο χρησιμοποιώντας την model.predict() στο σύνολο δοκιμών και λάβετε μια λίστα με τις προβλέψεις του LeNet5. np.argmax() Μετατρέψτε την έξοδο σε κατηγορικές τιμές από την κωδικοποίηση one-hot () και στη συνέχεια εκτυπώστε την αναφορά ταξινόμησης και τον πίνακα σύγχυσης. model.evaluate() Επιστρέφει τις τιμές των μετρικών απώλειας και ακρίβειας στο σύνολο δοκιμών που χρησιμοποιήθηκε για την αξιολόγηση του LeNet5. Το Lenet5 πέτυχε 67,91% ακρίβεια στην πρόβλεψη των κλάσεων αντικειμένων στο σύνολο δεδομένων Cifar10.

Βήμα 10: Αποθήκευση LeNet5

Μόλις το μοντέλο εκπαιδευτεί για την αναγνώριση των κλάσεων αντικειμένων Cifar10, μπορεί να αποθηκευτεί για μελλοντική χρήση σε έργα που περιλαμβάνουν την αναγνώριση αυτών. Η `model.to_json()` επιτρέπει τη μετατροπή της αρχιτεκτονικής του μοντέλου σε μορφή `.json`, ενώ η `model.save_weights()` επιτρέπει την αποθήκευση των συναπτικών βαρών που προέκυψαν μετά τη διαδικασία εκπαίδευσης σε μορφή `h5`.

Βήμα 11: Φόρτωση του LeNet5

Το προ-εκπαιδευμένο μοντέλο μπορεί να χρησιμοποιηθεί σε άλλα έργα ως συνάρτηση ικανή να επεξεργάζεται τις εισόδους που της παρέχονται. `model_from_json()` επιτρέπει τη φόρτωση προ-εκπαιδευμένων αρχιτεκτονικών μοντέλων αποθηκευμένων σε μορφή `json`. `model.load_weights()` επιτρέπει τη χρήση των βαρών στο μοντέλο που είναι αποτέλεσμα της εκπαίδευσης με ένα συγκεκριμένο σύνολο δεδομένων.

Βήμα 11: Επαναχρησιμοποίηση LeNet5

`model.predict()` αναλύει μια εικόνα για να προβλέψει τις προβλέψεις αν μια εικόνα περιέχει τις δέκα κλάσεις αντικειμένων Cifar10.

Μάθημα 3.5: Διαφορετικά CNN για την ταξινόμηση εικόνων

Θέματα προς συζήτηση

- Εξέλιξη των CNNs
- Πώς να χρησιμοποιήσετε το προ-εκπαιδευμένο μοντέλο Keras

Σημειώσεις διάλεξης

Ζητήματα και όρια του CNN. Τα αντικείμενα που περιέχονται στις εικόνες έχουν διαφορετικές σχετικές διαστάσεις. Η διάσταση εξαρτάται από την απόσταση που έχουν από την κάμερα που κατέγραψε την εικόνα. Τα CNN απαιτούν μεγάλο αριθμό φίλτρων για την ολοκληρωμένη ανάλυση της εικόνας, καθιστώντας τον υπολογιστικό υπολογισμό πολύ ακριβό. Επιπλέον, όπως όλα τα DNN, τα CNN αξιοποιούν αλγόριθμους βελτιστοποίησης με βάση τη διαβάθμιση για την ενημέρωση των συναπτικών βαρών τους. Αυτοί οι αλγόριθμοι βελτιστοποίησης εγείρουν το ζήτημα του προβλήματος της εξαφανιζόμενης κλίσης που εμποδίζει περαιτέρω βελτιώσεις κατά τη διαδικασία μάθησης. Ένα ακόμη ζήτημα είναι ότι τα κλασικά CNN είναι σε θέση να αναγνωρίζουν μόνο την παρουσία ενός συγκεκριμένου αντικειμένου μέσα σε μια εικόνα, αλλά δεν είναι σε θέση να προβλέψουν τις συντεταγμένες του αντικειμένου στην εικόνα ή να δώσουν πληροφορίες σχετικά με τα εικονοστοιχεία που ανήκουν στο αντικείμενο.

Υπολειμματικά δίκτυα (ResNet). Οι λειτουργικές μονάδες στον ανθρώπινο εγκεφαλικό φλοιό, οι στήλες των πυραμιδικών νευρώνων, ενέπνευσαν την εξέλιξη της αρχιτεκτονικής των συνελκτικών δικτύων: τα υπολειμματικά δίκτυα ή ResNet. Τα ResNet εκμεταλλεύονται "συνδέσεις παράλειψης" ή "συντομεύσεις", επιτρέποντας την επικοινωνία μεταξύ μη συνεχόμενων στρωμάτων. Τα σημεία από τα οποία ξεκινούν οι συνδέσεις παράκαμψης και οι τερματισμοί τους ορίζουν ένα μπλοκ στρωμάτων ως υπολειμματικό μπλοκ. Τα υπόλοιπα μπλοκ μπορούν να έχουν διαφορετικές δομές (δύο ή τρία παραλειπόμενα στρώματα) και επιτρέπουν στο δίκτυο να μειώσει το πρόβλημα της εξαφανιζόμενης κλίσης.

CNNs με βάση την περιοχή (R-CNNs). Προκειμένου να μειωθεί ο αριθμός των φίλτρων που χρησιμοποιούνται από τα CNN και, επομένως, να μειωθεί η υπολογιστική πολυπλοκότητα, τα R-CNNs αξιοποιούν έναν αλγόριθμο που χρησιμοποιείται συνήθως στην όραση υπολογιστών: τον αλγόριθμο επιλεκτικής αναζήτησης (SS). Αυτός ο αλγόριθμος προτείνει πολλές περιοχές ενδιαφέροντος (RoI) με βάση την ομοιότητα και τις διαφορές υφής. Στη συνέχεια, τα RoI εξαπλώνονται στη συνελκτική περιοχή για την εξαγωγή κρυφών χαρακτηριστικών. Μια μηχανή διανυσμάτων υποστήριξης (SVM) αντικαθιστά την περιοχή ταξινόμησης των κλασικών CNN για την ταξινόμηση των εξαχθέντων χαρακτηριστικών. Παράλληλα με το SVM, ένας γραμμικός παλινδρομητής επεξεργάζεται τις συντεταγμένες των προτεινόμενων RoIs για να πάρει τις συντεταγμένες του ορθογωνίου με την ελάχιστη περιοχή που περιέχει το αντικείμενο ενδιαφέροντος (bounding box). Τα R-CNN ξεπερνούν το όριο των κλασικών CNN στον εντοπισμό αντικειμένων στο εσωτερικό των εικόνων, αλλά η εκπαίδευσή τους είναι αργή, καθώς απαιτείται η εκπαίδευση τριών διαφορετικών μοντέλων: CNN, SVM και γραμμικό παλινδρομητή.

Επιλεκτική αναζήτηση. Ο αλγόριθμος επιλεκτικής αναζήτησης τμηματοποιεί την εικόνα σε πολλές περιοχές που χωρίζονται με βάση την ομοιότητα (χρώμα, υφή, μέγεθος). Μια προσέγγιση "από κάτω προς τα πάνω" συγχωνεύει τη μικρή τμηματοποιημένη περιοχή σε μεγαλύτερες περιοχές, οι οποίες οδηγούν σε διαφορετικά RoI. Οι επικαλυπτόμενες RoIs (υψηλή τομή IoU πάνω από την ένωση) απορρίπτονται με βάση μια χειροκίνητα διαμορφωμένη παράμετρο.

Γρήγορο R-CNN. Το Fast R-CNN είναι μια εξέλιξη του R-CNN. Η συνελικτική περιοχή ενεργεί πρώτα σε ολόκληρη την εικόνα εξάγοντας τον χάρτη χαρακτηριστικών. Στη συνέχεια, η επιλεκτική αναζήτηση προτείνει τα RoIs ξεκινώντας από τον χάρτη χαρακτηριστικών και όχι από την αρχική εικόνα. Μια σειρά από στρώματα συγκέντρωσης, που ονομάζονται στρώματα συγκέντρωσης RoI, μειώνουν τη διάσταση του προτεινόμενου RoI. Στο τέλος, ένα MLP με το στρώμα FC ταξινομεί τα προτεινόμενα RoIs και, παράλληλα, ένας γραμμικός παλινδρομητής υπολογίζει τις συντεταγμένες του οριοθετημένου πλαισίου. Τα γρήγορα R-CNN είναι ταχύτερα από τα R-CNN, αλλά η επιλεκτική αναζήτηση παραμένει το σημείο συμφόρησης αυτών των νευρωνικών δικτύων. Η SS επιβραδύνει τα γρήγορα R-CNN και δεν έχει υψηλή ακρίβεια στη διάκριση αντικειμένων με παρόμοια υφή ή με καθορισμένες ακμές.

Ταχύτερη R-CNN. Το στρώμα πρότασης περιοχής (RPL) αντικαθιστά τον αλγόριθμο επιλεκτικής αναζήτησης. Πρόκειται για ένα στρώμα τεχνητών νευρώνων που συνδέονται άμεσα με τη συνελικτική περιοχή. Ολόκληρη η δομή (περιοχή συνέλιξης + στρώμα πρότασης περιοχής) ονομάζεται δίκτυο πρότασης περιοχής (RPN) και έχει ως στόχο να προτείνει RoIs με βάση ένα σύστημα αγκύρωσης. Μια άγκυρα είναι ένας υποψήφιος RoI που πρέπει να βελτιωθεί εάν περιέχει αντικείμενα. Οι άγκυρες που πέρασαν την επιλογή για να γίνουν RoI επεξεργάζονται από τα στρώματα συγκέντρωσης RoI για να ταξινομηθούν στη συνέχεια. Η αντικατάσταση του αλγορίθμου επιλεκτικής αναζήτησης καθιστά το Faster R-CNN ταχύτερο από το Fast R-CNN, αλλά εξακολουθεί να έχει το όριο της μη αναγνώρισης των εικονοστοιχείων που ανήκουν στο αντικείμενο.

Άγκυρες (Anchors). Η άγκυρα είναι ένα κουτί. Οι άγκυρες τοποθετούνται σε προεπιλεγμένες θέσεις σε όλους τους χάρτες χαρακτηριστικών. Στην τυπική διαμόρφωση, υπάρχουν 9 άγκυρες σε κάθε θέση (χωρισμένες σε τρεις ομάδες: μεγάλες, μεσαίες, μικρές) με διαφορετικό μέγεθος και σχετική διάσταση. Ο αριθμός των αγκυρών είναι ελάχιστα μικρότερος από το συνδυασμό των φίλτρων ολίσθησης. Το RPN επεξεργάζεται την περιοχή των εικόνων που βρίσκεται μέσα στην άγκυρα για να υπολογίσει την πιθανότητα να είναι φόντο ή αντικείμενα.

Μάσκα R-CNN. Αυτή η πιο σύνθετη αρχιτεκτονική CNN μπορεί να προβλέψει τη μάσκα: το σύνολο όλων των εικονοστοιχείων που ανήκουν σε συγκεκριμένα αντικείμενα. Σε σύγκριση με το Faster R-CNN, έχει τρεις κρίσιμες αλλαγές:

- Δίκτυο πυραμίδας χαρακτηριστικών (FPN): Αντικαθιστά τη συνελικτική περιοχή
- RoI Ευθυγράμμιση στρωμάτων: Αντικατάσταση RoI, Συγκέντρωση στρωμάτων
- Κλάδος μάσκας: παράλληλα με τον ταξινομητή

Δίκτυο πυραμίδας χαρακτηριστικών (FPN). Το RPN αντικαθιστά τη συνελικτική περιοχή του ResNet. Το FPN αποτελείται από:

- Ένας κλάδος από κάτω προς τα πάνω (C στρώματα): αντιπροσωπεύεται από ένα ResNet που μειώνει κατά το ήμισυ το μέγεθος των χαρτών χαρακτηριστικών σε κάθε συνελικτικό στρώμα.
- Ένας κλάδος από πάνω προς τα κάτω (M στρώματα): λαμβάνει από τα στρώματα του ResNet και διπλασιάζει το μέγεθος του σήματος σε κάθε στρώμα μέσω της αναβάθμισης του κοντινότερου γείτονα.
- Ο κλάδος εξόδου (στρώματα P): ισοδυναμεί με χάρτες χαρακτηριστικών. Αναλύεται από το RPL για την πρόβλεψη του RoI.

Στρώμα ευθυγράμμισης RoI. (RoI-Align layer) Τα στρώματα ευθυγράμμισης RoI αντικαθιστούν τα στρώματα συγκέντρωσης RoI. Η έξοδος του στρώματος RoIpooling είναι ελαφρώς εσφαλμένη από την αρχική εικόνα λόγω προβλημάτων στρογγυλοποίησης συντεταγμένων, αλλά η διαδικασία τμηματοποίησης εικόνας απαιτεί εξειδίκευση σε επίπεδο εικονοστοιχείου. Το RoI-Align εκμεταλλεύεται τη διγραμμική παρεμβολή για την ανάκτηση της αρχικής θέσης εικονοστοιχείου.

Mask Branch. Ο κλάδος ταξινόμησης διαθέτει έναν παράλληλο κλάδο: ένα πλήρως συνελικτικό δίκτυο (FCN), το οποίο επεξεργάζεται την έξοδο RoI-Align για να δημιουργήσει τη μάσκα: τις συντεταγμένες των εικονοστοιχείων που ανήκουν στο αντικείμενο. Η έξοδος είναι ένας πίνακας με την ίδια διάσταση της εικόνας. Ο πίνακας περιέχει μονάδες σε όλες τις θέσεις όπου το εικονοστοιχείο ανήκει στο αντικείμενο και μηδενικά αλλού.

Προ-εκπαιδευμένο μοντέλο Keras. Η ενότητα Keras.application περιέχει διάφορες αρχιτεκτονικές CNNs που έχουν ήδη εκπαιδευτεί χρησιμοποιώντας τεράστια σύνολα δεδομένων εικόνων. Η χρήση μαζικών συνόλων δεδομένων είναι απαραίτητη για την εκπαίδευση πολύ βαθιών συνελικτικών δικτύων και την αποφυγή προβλημάτων υπερπροσαρμογής κατά τη φάση της εκπαίδευσης. Η μαζική εκπαίδευση είναι μια παρατεταμένη και υπολογιστικά δαπανηρή διαδικασία. Επιπλέον, η συλλογή ενός συνόλου δεδομένων εκατομμυρίων εικόνων είναι μια πολύ δαπανηρή και περίπλοκη διαδικασία.

Βάρη ImageNet. Τα προ-εκπαιδευμένα CNN της Keras περιέχουν τον πίνακα των συναπτικών βαρών που είναι το αποτέλεσμα μιας διαδικασίας εκπαίδευσης που πραγματοποιήθηκε με μαζικά σύνολα δεδομένων. Το ImageNet (image-net.org) είναι ένα σύνολο δεδομένων εικόνων που περιέχει περισσότερες από 1,2 εκατομμύρια εικόνες εκπαίδευσης που ανήκουν σε 1000 διαφορετικές κλάσεις οργανωμένες ιεραρχικά. Το ImageNet αποτελεί το σημείο αναφοράς στον διαγωνισμό: ImageNet Large Scale Visual Recognition Challenge (ILSVRC), έναν διαγωνισμό για αλγόριθμους που ταξινομούν σκηνές και ανιχνεύουν σωστά αντικείμενα.

VGG16. Το VGG16 είναι ένα από τα καλύτερα μοντέλα CNN που προτάθηκαν στο ILSVRC το 2014. Η είσοδος του δικτύου έχει μέγεθος (224,224,3). Το φίλτρο ολίσθησης είναι ένας μικρός πίνακας (3,3) ως δεκτικό πεδίο. Η πτυχωτή περιοχή αποτελείται από πέντε διαφορετικά μπλοκ δύο ή τριών πτυχωτών στρωμάτων (C-layer). Κάθε μπλοκ καταλήγει σε ένα στρώμα συγκέντρωσης (P-layer). ([C1,C2,P3] - [C4,C5,P6] [C7,C8,C9,P10] - [C11,C12,C13,P14] - [C15,C16,C17,P18]. Η περιοχή ταξινόμησης έχει τρία στρώματα FC ([FC1,FC2, output]). Το στρώμα εξόδου έχει 1000 μονάδες εξόδου εάν το VGG16 φορτώνει βάρη ImageNet. Η ανεκπαιδευτή έκδοση είναι επίπονη για να εκπαιδευτεί και απαιτεί υψηλή υπολογιστική ισχύ.

Εισαγωγή VGG16. Η VGG16 μπορεί εύκολα να εισαχθεί χρησιμοποιώντας την ενότητα `Keras.applications`. Το εισαγόμενο δίκτυο χρησιμοποιεί ήδη τα βάρη που προέκυψαν από την εκπαίδευση με το ImageNet.

Προεπεξεργασία εικόνων. Η ενότητα `Keras.preprocessing.image` περιέχει συναρτήσεις για τη φόρτωση εικόνων και τη μετατροπή τους σε `np.array` συγκεκριμένου μεγέθους που μπορεί να επεξεργαστεί από τη μέθοδο `VGG16.preprocess_input`.

Πρόβλεψη κατηγορίας εικόνας. Τώρα το προ-εκπαιδευμένο μοντέλο VGG16 μπορεί να χρησιμοποιηθεί για την πρόβλεψη της κλάσης του αντικειμένου που περιέχεται στην εικόνα χρησιμοποιώντας τη μέθοδο `model.predict(image)`. Το VGG16 διαθέτει μια ενσωματωμένη μέθοδο για την αποκωδικοποίηση της κωδικοποιημένης με `énahot` εξόδου του σε ονόματα κλάσεων του ImageNet.

ResNet50. Ένα βαθύ υπολειμματικό δίκτυο με 50 επίπεδα που εκπαιδεύτηκε με το σύνολο δεδομένων ImageNet. Το ResNet50 μπορεί εύκολα να εισαχθεί χρησιμοποιώντας την ενότητα `Keras.applications`. Το εισαγόμενο δίκτυο χρησιμοποιεί ήδη τα βάρη που προέκυψαν από την εκπαίδευση με το ImageNet.

Προεπεξεργασία εικόνων. Η ενότητα `Keras.preprocessing.image` σας επιτρέπει να χρησιμοποιείτε συναρτήσεις για τη φόρτωση εικόνων και τη μετατροπή τους σε `np.array` που μπορούν να επεξεργαστούν από τις μεθόδους προεπεξεργασίας που περιλαμβάνονται στο μοντέλο ResNet50.

Πρόβλεψη κατηγορίας εικόνας. Τώρα το προεκπαιδευμένο μοντέλο ResNet50 μπορεί να χρησιμοποιηθεί για την πρόβλεψη της κλάσης του αντικειμένου που περιέχεται στην εικόνα χρησιμοποιώντας τη μέθοδο `model.predict(image)`. Το ResNet50 διαθέτει μια ενσωματωμένη μέθοδο για την αποκωδικοποίηση της κωδικοποιημένης με `énahot` εξόδου του σε ονόματα κλάσεων του ImageNet.

Προσαρμογή μοντέλων Keras. Τα προ-εκπαιδευμένα μοντέλα Keras ImageNet είναι αναμφίβολα χρήσιμα αν κάποιος θέλει να λύσει γρήγορα ένα πρόβλημα ταξινόμησης. Ωστόσο, θα μπορούσε να συμβεί ότι η κλάση του αντικειμένου που πρέπει να αναγνωριστεί δεν περιλαμβάνεται μεταξύ εκείνων που περιλαμβάνονται στο τεράστιο σύνολο δεδομένων ή ότι η ακρίβεια που επιτυγχάνεται από το μοντέλο δεν είναι αρκετή για τον τελικό σκοπό του έργου. Είναι δυνατή η προσαρμογή των προ-εκπαιδευμένων μοντέλων Keras μέσω μιας προσαρμοσμένης διαδικασίας μάθησης που βασίζεται σε ένα εξατομικευμένο σύνολο δεδομένων εικόνων. Η εξατομικευμένη εκπαίδευση μπορεί να πραγματοποιηθεί με δύο διαφορετικούς τρόπους:

- Πλήρης εκπαίδευση: Εάν το σύνολο εικόνων εκπαίδευσης είναι αρκετά μεγάλο, το μοντέλο μπορεί να επανεκπαιδευτεί πλήρως από την αρχή για να αποκτήσει ένα αρκετά ακριβές δίκτυο που να ικανοποιεί τους σκοπούς του έργου.
- Μάθηση μεταφοράς: Εάν το σύνολο των εικόνων εκπαίδευσης δεν είναι αρκετά μεγάλο, είναι δυνατόν να επανεκπαιδευτεί εν μέρει το μοντέλο διατηρώντας τα πλεονεκτήματα της εκπαίδευσης στο ImageNet. Αυτή η διαδικασία, που ορίζεται ως μάθηση μεταφοράς, μπορεί να πραγματοποιηθεί με διάφορους τρόπους:
 - Αντικαταστήστε και εκπαιδεύστε την περιοχή ταξινόμησης.
 - Εκπαιδεύστε εκ νέου τα τελευταία στρώματα συνελκτικού συστήματος.

- Πάγωμα, προ-εκπαίδευση και λεπτομερής ρύθμιση.

Αντικατάσταση και εκπαίδευση της περιοχής ταξινόμησης. Το στρώμα εξόδου στο προ-εκπαιδευμένο μοντέλο Keras ImageNet έχει 1000 μονάδες εξόδου που αντιστοιχούν σε κλάσεις αντικειμένων ImageNet. Μια προσαρμοσμένη περιοχή ταξινόμησης με ένα νέο στρώμα εξόδου μπορεί να αντικαταστήσει την αρχική περιοχή ταξινόμησης του μοντέλου. Το στρώμα εξόδου πρέπει να έχει κάποιες μονάδες που αντιστοιχούν στον αριθμό των κλάσεων αντικειμένων στο προσαρμοσμένο σύνολο δεδομένων. Κατά τη διάρκεια της διαδικασίας εκπαίδευσης, είναι δυνατό να παγώσουν όλα τα βάρη του δικτύου εκτός από εκείνα της περιοχής ταξινόμησης και να εκπαιδευτούν μόνο αυτά. Το νεοεκπαιδευμένο μοντέλο διατηρεί την ικανότητα εξαγωγής χαρακτηριστικών που αποκτήθηκε από την εκπαίδευση με ένα τεράστιο σύνολο δεδομένων, αλλά η περιοχή ταξινόμησης μπορεί να αναγνωρίσει τα εξαγόμενα χαρακτηριστικά για να αναγνωρίσει νέες κλάσεις. Η αντικατάσταση και η εκπαίδευση της περιοχής ταξινόμησης είναι η σωστή λύση όταν τα εξαγόμενα χαρακτηριστικά είναι παρόμοια με τα χαρακτηριστικά των νέων αναγνωρίσιμων κλάσεων.

Επανεκπαίδευση των τελευταίων στρωμάτων συνελίξεων. Το πρώτο στρώμα της περιοχής συνελίξεων εξάγει χαρακτηριστικά χαμηλού επιπέδου, όπως γραμμές και αντιθέσεις που είναι κοινά σε όλα τα αντικείμενα. Αντίθετα, τα τελευταία στρώματα συνελικτικής περιοχής εξάγουν χαρακτηριστικά υψηλού επιπέδου που σχετίζονται με συγκεκριμένα αντικείμενα. Αυτός ο τρόπος εκμάθησης μεταφοράς συνίσταται στην αντικατάσταση της περιοχής ταξινόμησης και την εκπαίδευσή της, αλλά τα τελευταία στρώματα συνελικτικής συνέλιξης δεν παγώνουν κατά τη διάρκεια της διαδικασίας εκπαίδευσης. Αυτοί οι μηχανισμοί επιτρέπουν τη διατήρηση των πλεονεκτημάτων στην εξαγωγή χαρακτηριστικών χαμηλού επιπέδου, αλλά επιτρέπουν την εκμάθηση του τρόπου εξαγωγής νέων χαρακτηριστικών υψηλού επιπέδου. Η επανεκπαίδευση των τελευταίων στρωμάτων συνελικτικής ανάλυσης είναι η σωστή λύση όταν τα χαρακτηριστικά υψηλού επιπέδου του νέου αναγνωρίσιμου αντικειμένου δεν είναι παρόμοια με τα αναγνωρίσιμα αντικείμενα στο τεράστιο σύνολο δεδομένων.

Πάγωμα, προ-εκπαίδευση και τελειοποίηση. Αυτός ο τρόπος εκμάθησης μεταφοράς αποτελείται από δύο βήματα:

1. Πάγωμα και προ-εκπαίδευση. Μια νέα προσαρμοσμένη περιοχή ταξινόμησης αντικαθιστά την αρχική. Τα στρώματα συνελικτικού συστήματος παγώνουν κατά τη διάρκεια μιας πρώτης διαδικασίας εκπαίδευσης με την έναρξη του προσαρμοσμένου συνόλου δεδομένων. Τα βάρη που λαμβάνονται με αυτόν τον τρόπο αποθηκεύονται από τη χρήση τους στη δεύτερη φάση εκπαίδευσης.
2. Λεπτομερής ρύθμιση. Το μοντέλο και τα βάρη που προέκυψαν κατά την προ-εκπαίδευση υποβάλλονται σε μια δεύτερη διαδικασία εκπαίδευσης με χαμηλότερο ρυθμό μάθησης. Κατά τη λεπτή ρύθμιση, τα βάρη των στρωμάτων συνελικτικής ανάλυσης δεν είναι παγωμένα και ενημερώνονται με την εκπαίδευση.

Το πάγωμα, η προ-εκπαίδευση και η λεπτομερής ρύθμιση του μοντέλου είναι η σωστή λύση όταν το προσαρμοσμένο σύνολο δεδομένων έχει μικρή διάσταση και οι εικόνες είναι πολύ

διαφορετικές από τις εικόνες του μαζικού συνόλου δεδομένων. Η χρήση του προσαρμοστικού βελτιστοποιητή ρυθμού μάθησης δεν είναι καλή για τη διαδικασία λεπτής ρύθμισης.

Περιπτώσεις χρήσης

VGG16 ΠΡΟ-ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΡΥΘΜΙΣΗ

Σε αυτή την περίπτωση χρήσης, το προ-εκπαιδευμένο μοντέλο VGG16 Keras θα υποβληθεί σε διαδικασία παγώματος, προ-εκπαίδευσης και εκλέπτυνσης μάθησης μεταφοράς. Στόχος είναι η ταξινόμηση εικόνων που αναπαριστούν γάτες και σκύλους με υψηλή ακρίβεια.

Μεταφορά μάθησης στην ταξινόμηση γάτας και σκύλου.

- Εισαγωγή VGG16.
- Φόρτωση συνόλου δεδομένων.
- Προσθήκη μιας προσαρμοσμένης περιοχής ταξινόμησης.
- Ορίστε τον εκπαιδευμένο φάκελο VGG 16 και ορίστε τα callbacks.
- Προεκπαίδευση του VGG16.
- Φορτώστε το raw-tuned VGG16.
- Fine-tune VGG16.
- Δοκιμή του fine-tuned VGG16.

Σύνολο δεδομένων για γάτες και σκύλους. Αυτό το σύνολο δεδομένων αποτελείται από 25.000 εικόνες γατών και σκύλων: 12.500 γάτες και 12.500 σκύλους. Το Kaggle χρησιμοποίησε αυτό το σύνολο δεδομένων ως σημείο αναφοράς για μια πρόκληση. Μπορείτε να κατεβάσετε το σύνολο δεδομένων και την άδεια χρήσης εδώ. Οι εικόνες κάθε κλάσης έχουν ονόματα από το 0 έως το 12499. Αφού κατεβάσετε το σύνολο δεδομένων, χωρίστε το σε σύνολο εκπαίδευσης και σε σύνολο δοκιμής. Στη συνέχεια, δημιουργήστε δύο υποφάκελους: train και test. Στον υποφάκελο test μετακινήστε τις εικόνες από 0 έως 499 και των δύο κλάσεων. Στον υποφάκελο train μετακινήστε τις εικόνες από 500 έως 12499 και για τις δύο κλάσεις.

[ΚΑΤΕΒΑΣΤΕ ΤΟΝ ΚΩΔΙΚΑ ΕΔΩ](#)

Βήμα 1: Εισαγωγή VGG16

Ορίστε τις υπερπαραμέτρους και στη συνέχεια εισαγάγετε το VGG16 από την ενότητα keras.applications. Μην συμπεριλάβετε την περιοχή ταξινόμησης και παγώστε τα στρώματα συνελκτικής ανάλυσης.

Βήμα 2: Φόρτωση συνόλου δεδομένων

Ορίστε το φάκελο train και test του συνόλου δεδομένων που περιέχει τις εικόνες. Στη συνέχεια, ενεργοποιήστε τις γεννήτριες δεδομένων. Οι γεννήτριες δεδομένων επιτρέπουν τη σταδιακή φόρτωση δεδομένων από το σκληρό δίσκο χωρίς κορεσμό της μνήμης. Επιπλέον, οι γεννήτριες προ-επεξεργάζονται αυτόματα τα δεδομένα, αναβαθμίζοντάς τα στο εύρος [0-1] και μετατρέποντας την ετικέτα στο επιθυμητό διάνυσμα εξόδου. Ένα άλλο συναρπαστικό χαρακτηριστικό των γεννητριών δεδομένων είναι η δυνατότητα αύξησης των δεδομένων

εκπαίδευσης με την περιστροφή των εικόνων ή με την αλλαγή του μεγέθους τους (αύξηση δεδομένων).

Βήμα 3: Προσθέστε ένα προσαρμοσμένο μπλοκ ταξινόμησης

Πρώτον, δημιουργήστε τη νέα προσαρμοσμένη περιοχή ταξινόμησης με ένα στρώμα FC, ένα στρώμα συγκέντρωσης μέσου όρου και διακοπή. Το στρώμα εξόδου έχει δύο μονάδες εξόδου, τον ίδιο αριθμό αναγνωρίσιμων κλάσεων. Όταν η περιοχή ταξινόμησης είναι έτοιμη να ενωθεί με την περιοχή συνελίξεων και να εκτυπωθεί η νέα αρχιτεκτονική του μοντέλου.

Βήμα 4: Ορίστε τον εκπαιδευμένο φάκελο VGG και τα callbacks

Ορίστε το φάκελο του εκπαιδευμένου μοντέλου και τη διαδρομή του βάρους. Τα keras.callbacks είναι χρήσιμα βοηθητικά προγράμματα στη διαδικασία εκπαίδευσης.

- Checkpoint: αυτόματη αποθήκευση των βαρών στο τέλος κάθε εποχής. Επιτρέπει να ελέγξετε αν τα νέα βάρη έχουν καλύτερη απώλεια επικύρωσης από τα αποθηκευμένα βάρη πριν τα αποθηκεύσετε
- Early stopping: κλείνει αυτόματα τη φάση εκπαίδευσης εάν η απώλεια επικύρωσης δεν βελτιωθεί σε ένα ελάχιστο δέλτα για έναν ορισμένο αριθμό εποχών.

Βήμα 5: Προ-εκπαίδευση

Ξεκινήστε την πρώτη εκπαίδευση VGG16 με το σύνολο δεδομένων για γάτες και σκύλους χρησιμοποιώντας υψηλό ρυθμό μάθησης.

Βήμα 6: Φόρτωση raw-tuned VGG16

Εκκινήστε ξανά το VGG16, αλλά αυτή τη φορά μην φορτώσετε τα βάρη του ImageNet, αλλά φορτώστε τα βάρη που είναι αποτελέσματα της προ-εκπαίδευσης με το σύνολο δεδομένων για γάτες και σκύλους.

Βήμα 7: Fine-tune

- Ξεκινήστε τη δεύτερη εκπαίδευση VGG16 με το σύνολο δεδομένων για γάτες και σκύλους με χαμηλότερο ρυθμό μάθησης.

Βήμα 8: Δοκιμή γάτας και σκύλου VGG16

Αξιολογήστε τη γάτα και το σκύλο fine-tuned VGG16: 95,5% ακρίβεια - 0.058 Απώλεια. Το λεπτομερώς συντονισμένο μοντέλο παρουσιάζει πραγματικά υψηλότερη ακρίβεια σε σύγκριση με το ακατέργαστο συντονισμένο μοντέλο.

Μάθημα 3.6: Εντοπισμός αντικειμένων Μοντέλο YOLO_v3

Θέματα προς συζήτηση

- Το μοντέλο YOLO
- Εξέλιξη του μοντέλου YOLO: v2 και v3

Σημειώσεις διάλεξης

Εντοπισμός αντικειμένων. Στον εντοπισμό αντικειμένων, για την αναγνώριση και τον εντοπισμό ενός συγκεκριμένου αντικειμένου, ένα μοντέλο πρέπει να είναι σε θέση να προβλέψει τις συντεταγμένες του οριοθετημένου πλαισίου, του ορθογωνίου με το ελάχιστο εμβαδόν που περικλείει το αντικείμενο. Οι συντεταγμένες του οριοθετημένου πλαισίου αντιπροσωπεύονται από το κέντρο του ορθογωνίου, το ύψος και το πλάτος του. Η έξοδος ενός μοντέλου εντοπισμού πρέπει να περιλαμβάνει αυτές τις τιμές στην πιθανότητα κατανομής μεταξύ των ανιχνεύσιμων κλάσεων.

Μοντέλο κοιτάς μόνο μια φορά (YOLO). Το YOLO είναι ένα από τα πιο αποδοτικά μοντέλα CNN που χρησιμοποιούνται σε διαδικασίες εντοπισμού αντικειμένων σε πραγματικό χρόνο. Τα R-CNNs χρησιμοποιούν τον αλγόριθμο επιλεκτικής αναζήτησης ή το επίπεδο πρότασης περιοχής για να προτείνουν διαφορετικές περιοχές ενδιαφέροντος που αναλύονται ξεχωριστά. Διαφορετικά, το YOLO αναλύει ολόκληρη την εικόνα ταυτόχρονα για να προβλέψει πολλαπλά υποθετικά πλαίσια οριοθέτησης. Οι συσκευές στη βάση του μοντέλου YOLO είναι οι εξής:

- Πλέγμα Κελιών (Grid cells)
- Πλαίσια οριοθέτησης (Bounding boxes)
- Μη μέγιστη Επικάλυψη (Non-max suppression)

Πλέγμα Κελιών. Το YOLO διαιρεί την εικόνα σε διαφορετικά κελιά εφαρμόζοντας ένα πλέγμα. Κάθε κελί πλέγματος προβλέπει την παρουσία N πιθανών οριοθετημένων κουτιών (που ρυθμίζονται χειροκίνητα) με τις συντεταγμένες τους και την εμπιστοσύνη του κουτιού. Ο αριθμός των υποθετικών οριοθετημένων κουτιών, N ορίζει τον αριθμό των αντικειμένων που μπορούν να ανιχνευθούν σε κάθε κελί. Η εμπιστοσύνη πλαισίου αντικατοπτρίζει πόσο σίγουρο είναι το μοντέλο για την αναγνώριση ενός γενικού αντικειμένου που περιέχεται σε ένα μόνο κελί.

Πλαίσια οριοθέτησης. Κάθε πιθανό πλαίσιο οριοθέτησης προβλέπει ένα διάνυσμα εξόδου με διαφορετικά στοιχεία:

$$y = [p, \text{bbx}, \text{bby}, \text{bbh}, \text{bbw}, c1, c..., cN]$$

όπου p είναι η αξιοπιστία αντικειμενικότητας ή η αξιοπιστία του πλαισίου: η πιθανότητα ότι το οριοθετημένο πλαίσιο περιλαμβάνει ένα αντικείμενο ενδιαφέροντος: bbx , bby , bbh , bbw είναι οι συντεταγμένες του οριοθετημένου πλαισίου: κέντρο, ύψος και πλάτος: $c1$, $c...$, cN είναι η αξιοπιστία κλάσης: η κατανομή πιθανότητας μεταξύ των διάφορων κλάσεων αντικειμένων που μπορεί να αναγνωρίσει το μοντέλο. Εάν ένα κύτταρο προβλέπει περισσότερα από ένα οριοθετημένα πλαίσια, συνενώνει τα διανύσματα εξόδου τους

Μη μέγιστη Επικάλυψη. Δεδομένου ότι διαφορετικά οριοθετημένα πλαίσια μπορούν να εντοπίζουν το ίδιο αντικείμενο, είναι απαραίτητο να εξαλειφθούν τα αντίγραφα διατηρώντας κάθε άλλο οριοθετημένο πλαίσιο που περιέχει ένα επιπλέον επικαλυπτόμενο αντικείμενο. Για το σκοπό αυτό, το YOLO εκμεταλλεύεται τον αλγόριθμο καταστολής μη μέγιστων τιμών:

1. Εξαλείψτε όλα τα οριοθετημένα πλαίσια που έχουν εμπιστοσύνη πλαισίου $< 0,6$.
2. Από τα εναπομείναντα bounding boxes, επιλέξτε αυτό με την υψηλότερη εμπιστοσύνη box confidence.
3. Συγκρίνετε την επικάλυψη μεταξύ των εναπομεινάντων bounding boxes υπολογίζοντας τον δείκτη διασταύρωσης πάνω από την ένωση (IoU)

Intersection over Union (IoU): ο λόγος μεταξύ της περιοχής τομής και της περιοχής ένωσης μεταξύ δύο οριοθετημένων πλαισίων:

1. Υπολογίστε τις συντεταγμένες του πάνω αριστερού και του κάτω δεξιού σημείου των πλαισίων οριοθέτησης.
2. Υπολογίστε τις συντεταγμένες του ορθογωνίου που προκύπτει από την τομή των δύο πλαισίων οριοθέτησης.
3. Υπολογίστε το εμβαδόν της τομής.
4. Αθροίστε το εμβαδόν και των δύο πλαισίων οριοθέτησης και αφαιρέστε το εμβαδόν τομής.
5. Υπολογίστε τον λόγο μεταξύ των εμβαδών ένωσης και τομής

YOLOv1. Το YOLOv1 είναι το πρώτο μοντέλο YOLO που αναπτύχθηκε. Αποτελείται από μια συνελκτική περιοχή 24 επιπέδων. Η περιοχή ταξινόμησης αποτελείται από δύο πλήρως συνδεδεμένα στρώματα για την ταξινόμηση αντικειμένων και τη γραμμική παλινδρόμηση των οριοθετημένων πλαισίων. Το στρώμα εξόδου χρησιμοποιεί μια softmax ως συνάρτηση ενεργοποίησης, ενώ τα στρώματα συνελκτικής σύνδεσης χρησιμοποιούν το leaky ReLU (εξέλιξη του ReLU). Έχει αναπτυχθεί μια ρηχότερη έκδοση του YOLOv1, το "μικροσκοπικό YOLO". Διαθέτει μόνο εννέα συνελκτικά στρώματα. Το "Tiny YOLO" είναι ακόμη πιο γρήγορο από το YOLOv1, αλλά είναι λιγότερο ακριβές από το αρχικό μοντέλο.

Συναρτήσεις απωλειών YOLOv1. Ένα σύστημα κελιών πλέγματος που προβλέπει πολλαπλά πλαίσια οριοθέτησης για κάθε κελί διασφαλίζει ότι το YOLOv1 δεν κάνει μόνο λάθη στην αναγνώριση των κλάσεων διαφορετικών αντικειμένων (εμπιστοσύνη κλάσεων). Το YOLOv1 κάνει επίσης σφάλματα στην πρόβλεψη της παρουσίας ή της απουσίας αντικειμένων εντός του οριοθετημένου πλαισίου (εμπιστοσύνη πλαισίου) και στην πρόβλεψη των συντεταγμένων του. Αυτοί οι τρεις διαφορετικοί τύποι σφαλμάτων απαιτούν ότι η συνολική συνάρτηση απωλειών του YOLOv1 είναι το άθροισμα τριών επιμέρους συναρτήσεων απωλειών:

- Απώλεια εντοπισμού: Υπολογίζει το σφάλμα στην πρόβλεψη των συντεταγμένων του οριοθετημένου πλαισίου.
- Απώλεια ταξινόμησης: Υπολογίζει το σφάλμα στην πρόβλεψη της κλάσης του αντικειμένου.
- Απώλεια εμπιστοσύνης: Υπολογίζει το σφάλμα κατά την πρόβλεψη της εμπιστοσύνης του πλαισίου.

Κατά την ανάλυση εικόνων, είναι πολύ πιο πιθανό ένα πλαίσιο οριοθέτησης να μην περιέχει ένα αντικείμενο παρά να περιέχει. Η ανισορροπία που δημιουργείται προκαλεί υπερένταση της κλίσης

που προκαλεί αστάθεια του YOLOv1 κατά τη φάση της μάθησης. Οι λ_{coord} και λ_{noobj} είναι παράμετροι που ρυθμίζονται χειροκίνητα και εισάγονται για την εξισορρόπηση της μάθησης μεταξύ φόντου και αντικειμένων. (Στο έγγραφο YOLO παίρνουν τις τιμές 5 και 0,5 αντίστοιχα).

$$\begin{aligned} & \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\ & + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\ & + \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} (C_i - \hat{C}_i)^2 \\ & + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{noobj}} (C_i - \hat{C}_i)^2 \\ & + \sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2 \end{aligned}$$

Απώλεια εντοπισμού. Υπολογίζει το τετραγωνικό σφάλμα μεταξύ των πραγματικών συντεταγμένων του πλαισίου οριοθέτησης και εκείνων που προβλέπει το μοντέλο YOLOv1:

$$\begin{aligned} & \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\ & + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \end{aligned}$$

Απώλεια ταξινόμησης. Υπολογίζει την απώλεια που δημιουργείται μεταξύ της πραγματικής κλάσης και της προβλεπόμενης κλάσης αντικειμένου:

$$+ \sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$$

Το πρώτο άθροισμα έχει τιμή ένα αν ένα αντικείμενο βρίσκεται μέσα στο κελί i και μηδέν όταν το κελί i δεν περιέχει κάποιο αντικείμενο. Το δεύτερο άθροισμα είναι το τετραγωνικό σφάλμα των υπό συνθήκη πιθανοτήτων για κάθε κλάση.

Απώλεια εμπιστοσύνης στο κουτί. Υπολογίζει το τετραγωνικό σφάλμα μεταξύ της πραγματικής εμπιστοσύνης του κουτιού ($C = 1$ εάν υπάρχει αντικείμενο μέσα στο κελί, 0 διαφορετικά) και της προβλεπόμενης εμπιστοσύνης του κουτιού:

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} (C_i - \hat{C}_i)^2$$

$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{noobj}} (C_i - \hat{C}_i)^2$$

Η πρώτη σειρά υπολογίζει το σφάλμα εάν υπάρχει αντικείμενο εντός του οριοθετημένου πλαισίου, ενώ η δεύτερη σειρά υπολογίζει το σφάλμα εάν δεν υπάρχει αντικείμενο εντός του οριοθετημένου πλαισίου.

Πλεονεκτήματα YOLOv1. Εάν εκπαιδευτεί χρησιμοποιώντας το σύνολο δεδομένων ImageNet και συγκριθεί με ταχύτερα R-CNN, το YOLOv1 είναι ταχύτερο από άλλα CNN και επιτρέπει την ανάλυση ολόκληρης της εικόνας, μαθαίνοντας να αναγνωρίζει το χαρακτηριστικό του φόντου (λιγότερα ψευδώς θετικά αποτελέσματα). Επιπλέον, το YOLO μαθαίνει γενικεύσιμες αναπαραστάσεις αντικειμένων και, όταν το YOLO συγκρίνεται με άλλα CNN, επιτυγχάνει καλύτερα αποτελέσματα στην αναγνώριση αντικειμένων που περιέχονται σε μη πραγματικές εικόνες, όπως τα έργα τέχνης.

Μειονεκτήματα του YOLOv1. Σε σύγκριση με άλλα CNN, το YOLO έχει χαμηλότερο ποσοστό ακρίβειας, ιδίως στην αναγνώριση μικρών αντικειμένων. Επιπλέον, το YOLO παρουσιάζει προβλήματα στον εντοπισμό ομάδων ίδιων αντικειμένων, όπως σμήνη πουλιών ή μάζες ανθρώπων. Ένα άλλο πρόβλημα στην ακρίβεια του YOLO είναι η αναγνώριση αντικειμένων με μεταβαλλόμενη αναλογία διαστάσεων.

YOLOv2: βελτιώσεις ακρίβειας. Τα χαρακτηριστικά αυτά οδηγούν σε αύξηση της ακρίβειας:

- Ομαλοποίηση παρτίδας: Η προσθήκη αυτής της διαδικασίας σε όλα τα στρώματα του δικτύου προωθεί τη σύγκλιση κατά τη φάση εκμάθησης.
- Ταξινομητής υψηλής ανάλυσης: Ο YOLOv1 εκπαιδεύτηκε χρησιμοποιώντας εικόνες ImageNet (ανάλυση = 224,224,3). Ο YOLOv2 λαμβάνει έναν πιο πυκνό πίνακα εισόδου (448,448,4). Χρειάστηκε να επανεκπαιδευτεί το πρώτο συγκεκριμένο στρώμα με μια διαδικασία λεπτομερούς ρύθμισης.
- Συνελικτικό με πλαίσια άγκυρας: Το YOLOv2 χρησιμοποιεί ένα σύστημα κουτιών αγκύρωσης παρόμοιο με το ταχύτερο R-CNN. Οι διαστάσεις των αγκυρών επιλέγονται με το χέρι και μπορούν να αναφέρονται στην αναλογία διαστάσεων πραγματικών αντικειμένων. Στην αρχή της φάσης εκμάθησης, μειώνει την απώλεια εντοπισμού σε σύγκριση με τη δημιουργία των bounding boxes σε εντελώς τυχαία.
- Συστάδα διαστάσεων. Το K-means βοηθά στην πρόβλεψη του αριθμού των πιθανών αντικειμένων που περιέχονται στην εικόνα. Χρήσιμο για τον εντοπισμό ομάδων αντικειμένων.
- Άμεση πρόβλεψη θέσης: Περιορίζει τη μετατόπιση των πιθανών συντεταγμένων των bounding boxes.
- Λεπτόκοκκα χαρακτηριστικά. Η συνέλιξη μειώνει τη χωρική διάσταση του σήματος και η ανίχνευση μικρών αντικειμένων γίνεται δύσκολη. Για την επίλυση αυτού του προβλήματος, το YOLOv2 χρησιμοποιεί το passthrough. Μέσω αυτής της προσέγγισης, τα

χαρακτηριστικά που εξάγονται σε διαφορετικά επίπεδα συνέλιξης συνυπάρχουν στα επόμενα στρώματα του δικτύου.

- Εκπαίδευση πολλαπλών κλιμάκων. Η ανάλυση των εικόνων εκπαίδευσης αναβαθμίζεται σε διαφορετικά πολλαπλάσια του 32.

YOLOv2: βελτιώσεις ταχύτητας. Η αρχιτεκτονική DarkNet-19, που προτείνεται στο μοντέλο YOLOv2 μειώνει το απαιτούμενο υπολογιστικό κόστος. Το GoogleNet αντικαθιστά την αρχιτεκτονική της συνελικτικής περιοχής. Ένα συνελικτικό στρώμα που χρησιμοποιεί φίλτρα σχήματος 1x1 και ακολουθείται από μια συγκέντρωση του παγκόσμιου μέσου όρου και το στρώμα εξόδου αντικαθιστά την περιοχή ταξινόμησης

YOLOv2: Βελτιώσεις ταξινόμησης. Συνδυάζοντας πολλαπλά σύνολα δεδομένων εκπαίδευσης, είναι δυνατή η βελτίωση των λεπτομερειών ταξινόμησης. Το YOLOv2 εκτελεί ιεραρχική ταξινόμηση. Το ImageNet περιέχει εικόνες πολλών κλάσεων αντικειμένων που είναι αμοιβαία αποκλειόμενες (πάνω από 22k), ενώ το σύνολο δεδομένων COCO περιέχει πληροφορίες για περίπου 90 ανώτερες ιεραρχικές κλάσεις. Χρησιμοποιώντας το WordNet, μια ιεραρχικά οργανωμένη βάση δεδομένων λέξεων (ένα μπουλντόγκ είναι ένας σκύλος), είναι δυνατόν να συνδυαστούν οι ετικέτες και των δύο συνόλων δεδομένων, λαμβάνοντας ένα ιεραρχικό δέντρο όλων των κλάσεων (WordTree). Οι ληφθείσες ετικέτες οδηγούν σε οργανωμένα ιεραρχικά επίπεδα. Οι ετικέτες μεταξύ των διαφορετικών επιπέδων δεν αποκλείουν η μία την άλλη (ένα μπουλντόγκ μπορεί να είναι σκύλος), ενώ οι ετικέτες του ίδιου επιπέδου αποκλείουν η μία την άλλη (μια γάτα δεν μπορεί να είναι σκύλος).

YOLOv3. Τέσσερα νέα χαρακτηριστικά οδηγούν στη βελτίωση του YOLOv2:

- Προβλέψεις οριοθετημένου πλαισίου: Ένας λογιστικός παλινδρομητής υπολογίζει την εμπιστοσύνη του κουτιού. Εάν το προβλεπόμενο πλαίσιο οριοθέτησης επικαλύπτει πλήρως το πραγματικό πλαίσιο οριοθέτησης, η τιμή αυτή θα πρέπει να είναι ένα. Η πρόβλεψη θα αγνοηθεί εάν η τιμή αυτή πέσει κάτω από ένα ορισμένο όριο. Τα bounding boxes που δεν περιέχουν αντικείμενα δεν έχουν απώλεια ταξινόμησης και απώλεια εντοπισμού.
- Πρόβλεψη σε όλες τις κλίμακες: Διάφοροι ανεξάρτητοι λογιστικοί ταξινομητές αντικαθιστούν τον softmax στο στρώμα εξόδου. Ο ταξινομητής επιτρέπει στο YOLOv3 να αναγνωρίζει κλάσεις αντικειμένων που δεν είναι αμοιβαία αποκλειόμενες (ένα μπουλντόγκ είναι σκύλος). Η απώλεια ταξινόμησης δεν υπολογίζει το σφάλμα ως τετραγωνικό σφάλμα των υπό συνθήκη πιθανοτήτων αλλά ως δυαδική διασταυρούμενη εντροπία.
- Πρόβλεψη κλάσεων: Το YOLOv3 εξάγει κρυμμένα χαρακτηριστικά σε διαφορετικές κλίμακες εκμεταλλευόμενο έναν παρόμοιο μηχανισμό των δικτύων πυραμίδας χαρακτηριστικών (FPN).
- Εξαγωγέας χαρακτηριστικών: Η αρχιτεκτονική των 53 συνελικτικών στρωμάτων DarkNet53 αντικαθιστά την αρχιτεκτονική DarkNet-19.

Περιπτώσεις χρήσης

ΔΙΑΧΕΙΡΙΣΗ ΕΡΓΟΥ DNNs ΓΙΑ ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ

Σε αυτή την περίπτωση χρήσης, θα ασχοληθούμε με τον τρόπο διαχείρισης ενός υποθετικού έργου του DNN που στοχεύει στην ανάπτυξη μιας εφαρμογής. Η εφαρμογή θα είναι σε θέση να παρακολουθεί την οδική κυκλοφορία σε πραγματικό χρόνο μέσω της χρήσης δημόσιων καμερών. Η ανάπτυξη της εφαρμογής περιλαμβάνει διάφορα στάδια.

Βήμα 1: Συλλογή συνόλου δεδομένων

Η συλλογή ενός συνόλου δεδομένων με επισημειωμένες εικόνες που περιέχουν διαφορετικούς τύπους οχημάτων είναι το πρώτο βήμα για την ανάπτυξη του μοντέλου DNN. Υπάρχουν δύο διαφορετικοί τρόποι συλλογής του συνόλου δεδομένων :

- [Αναζήτηση και λήψη ενός προκατασκευασμένου συνόλου σχολιασμένων δεδομένων στο διαδίκτυο.](#) Συνήθως, αυτά τα είδη συνόλων δεδομένων διαθέτουν άδεια χρήσης. Πρέπει να ελέγξουμε στην άδεια αν μπορούμε να χρησιμοποιήσουμε το σύνολο δεδομένων για τον σκοπό μας. Τα σύνολα δεδομένων που συλλέγονται για εκπαιδευτικούς σκοπούς δεν παρέχουν μια αποτελεσματική διαδικασία μάθησης όσο ένα σύνολο δεδομένων που συλλέγεται ρητά για τον αλγόριθμο. Οι εικόνες που πρέπει να αναλύσει το μοντέλο είναι διαφορετικές από τις εικόνες για τις οποίες εκπαιδεύτηκε το μοντέλο.
- Συλλογή ενός συνόλου δεδομένων αφιερωμένου στον σκοπό. Μπορούμε να συλλέξουμε εικόνες από κάμερες οδών για να δημιουργήσουμε ένα προσαρμοσμένο σύνολο δεδομένων. Οι εικόνες που συλλέγονται πρέπει να επισημανθούν με χειροκίνητη ανίχνευση των bounding boxes που περικλείουν τα οχήματα που περιέχονται στη φωτογραφία. Διατίθενται δωρεάν διαδικτυακά προγράμματα για την επισήμανση εικόνων με διαφορετικές μορφές (<https://github.com/tzutalin/labelimg>, <http://www.robots.ox.ac.uk/~vgg/software/via/>). Αυτός ο τύπος συνόλου δεδομένων εκπαίδευσης επιτρέπει μια πιο αποτελεσματική διαδικασία μάθησης. Οι εικόνες που πρέπει να αναλύσει το μοντέλο είναι οι ίδιες με τις οποίες εκπαιδεύτηκε. Το σχολιασμένο σύνολο δεδομένων θα αναλυθεί για να επαληθευτεί η δίκαιη κατανομή όλων των κατηγοριών οχημάτων που το δίκτυο θα είναι σε θέση να αναγνωρίσει.

Το σύνολο δεδομένων πρέπει να χωριστεί σε δύο υποσύνολα: το σύνολο εκπαίδευσης και το σύνολο δοκιμής. Κάθε υποσύνολο πρέπει να έχει βέλτιστη κατανομή των διαφόρων κατηγοριών οχημάτων.

Βήμα 2: Επιλογή μοντέλου

Η επιλογή του μοντέλου πρέπει να γίνει με βάση τις απαιτήσεις της εφαρμογής μας. Τα R-CNN απαιτούν πολύ μεγάλη υπολογιστική ισχύ για να έχουν επαρκή ταχύτητα για την ανάπτυξη της εφαρμογής μας. Το μοντέλο YOLOv3 είναι πιο κατάλληλο για τον σκοπό μας χάρη στην ικανότητά του να αναλύει εικόνες σε πραγματικό χρόνο. Το YOLOv3 είναι διαθέσιμο στο GitHub και μπορείτε να το εγκαταστήσετε σαν πακέτο `anaconda` μέσω `prompt`. Το αποθετήριο GitHub διαθέτει επίσης άδεια χρήσης όπως και το εκπαιδευτικό σύνολο που διατίθεται στο διαδίκτυο.

Βήμα 3: Εκπαίδευση μοντέλου

Μέσω μηχανισμών μάθησης μεταφοράς, είναι δυνατόν να επανεκπαιδευτεί το μοντέλο YOLOv3 για να αναγνωρίζει ειδικά το σύνολο δεδομένων εικόνων μας. Αυτή η διαδικασία αυξάνει την ακρίβεια του YOLOv3 στην πρόβλεψη της θέσης των οχημάτων. Σε περίπτωση που το αρχικό μας μοντέλο δεν ήταν σε θέση να προβλέψει τη θέση του συγκεκριμένου αντικειμένου που μας

ενδιαφέρει, μια διαδικασία επανεκπαίδευσης επιτρέπει στο μοντέλο να αποκτήσει αυτή την ικανότητα, διατηρώντας παράλληλα τα πλεονεκτήματα της προηγούμενης εκπαίδευσης με ένα τεράστιο σύνολο δεδομένων εικόνων.

Βήμα 4: Δοκιμή μοντέλου

Το μοντέλο αλληλεπιδρά με το σύνολο δεδομένων δοκιμής. Εάν η ακρίβεια του μοντέλου ανταποκρίνεται στις ανάγκες μας, μπορούμε να προχωρήσουμε στο επόμενο στάδιο. Διαφορετικά, εάν η ακρίβεια δεν είναι επαρκής, είναι δυνατόν να επιστρέψουμε σε ένα από τα προηγούμενα στάδια:

- Σύνολο εκπαίδευσης: Είναι δυνατή η αύξηση του συνόλου εκπαίδευσης μετά την παρατήρηση των σφαλμάτων που κάνει το μοντέλο, προκειμένου να αυξηθούν τα παραδείγματα δεδομένων στα οποία το μοντέλο αποτυγχάνει.
- Δράση επί του μοντέλου: Αλλαγή του μοντέλου ή ελαφρά τροποποίηση της αρχιτεκτονικής του με επανεκπαίδευσή του με διαδικασίες εκμάθησης μεταφοράς.
- Βελτιστοποίηση της διαδικασίας μάθησης: Το μοντέλο μπορεί να υπερπροσαρμοστεί ή να παραμείνει κολλημένο σε ένα ελάχιστο δωμάτιο κατά την κάθοδο της κλίσης. Σε αυτή την περίπτωση, είναι απαραίτητο να επαναλάβετε τη διαδικασία εκπαίδευσης μεταβάλλοντας τις παραμέτρους αυτής της διαδικασίας για να μεγιστοποιήσετε την ακρίβεια που επιτυγχάνεται στο σύνολο επικύρωσης..

Βήμα 5: Ενσωμάτωση μοντέλου

Το δοκιμασμένο μοντέλο που ανταποκρίνεται στις απαιτήσεις μας μπορεί να εντοπίσει τα οχήματα που υπάρχουν στη ροή βίντεο που αποστέλλεται από μια κάμερα κυκλοφορίας σε πραγματικό χρόνο και μπορεί να χρησιμοποιηθεί για την ανάπτυξη μιας εφαρμογής που αποσκοπεί στη βελτίωση της καθημερινής ζωής. Κατά τη φάση ανάπτυξης και δοκιμής της εφαρμογής, το μοντέλο μπορεί να εξακολουθεί να είναι ακατάλληλο και μπορεί να απαιτηθεί η επιστροφή στα προηγούμενα βήματα.

Ενότητα 4: Βαθιά μάθηση για επεξεργασία φυσικής γλώσσας και ανάλυση μεγάλων δεδομένων

Μάθημα 4.1: Ενσωμάτωση λέξεων και ταξινόμηση κειμένου

Θέματα προς συζήτηση

- Εισαγωγή στο NLP και ιστορικό:
 - Τι είναι
 - Προηγούμενες προσεγγίσεις
- Ταξινόμηση κειμένου:
 - Συλλογή και επισήμανση δεδομένων
 - Προεπεξεργασία
 - Ενσωμάτωση λέξεων
 - Τι είναι αυτό
 - Ανάλυση και χρήση
 - Κατασκευή μοντέλου
 - Εκπαίδευση
 - Δοκιμή, αξιολόγηση, προσαρμογή
 - Αποθήκευση, χρήση

Σημειώσεις διάλεξης



Όπως είπαμε, η Επεξεργασία Φυσικής Γλώσσας αναφέρεται σε κάθε τεχνική που αποσκοπεί στην επεξεργασία της γλώσσας και η οποία, σε ορισμένες περιπτώσεις, φαίνεται να προσομοιώνει κάποια κατανόηση της ίδιας της γλώσσας. Ειδικότερα, μας ενδιαφέρουν οι μη τετριμμένες εργασίες, οι οποίες θα απαιτούσαν μια κατανόηση του ίδιου του κειμένου για να εκτελεστούν αποτελεσματικά.

Αναφερόμαστε σε εργασίες όπως:

- Ταξινόμηση κειμένων
- Ανάλυση συναισθήματος
- Ανίχνευση spam
- Εξαγωγή πληροφοριών
- Σύνοψη κειμένου
- Απάντηση ερωτήσεων
- Υπότιτλοι
- Μετάφραση
- Διάλογος και εικονικοί βοηθοί
- Παραγωγή ομιλίας

Εισαγωγή στην επεξεργασία φυσικής γλώσσας (NPL) Υπόβαθρο

Οι πρώτες μελέτες για την NLP ξεκινούν από την αυγή της επιστήμης των υπολογιστών.

Συμβολικό επίπεδο.

Προβλήματα:

- σημασία του πλαισίου,
- δυσκολία κατανόησης όσων δεν λέγονται,
- επιπλοκές,
- αμφιβολίες,
- ειρωνεία και σαρκασμός,
- tricky names
- μη τυποποιημένη γλώσσα

Στη συνέχεια, προσεγγίσεις που βασίζονται σε τεχνικές μηχανικής μάθησης: SVM, λογιστική παλινδρόμηση και τυχαίο δάσος, χρησιμοποιώντας διάσπαρτα διανύσματα ως δεδομένα.

Οι πρώτες μελέτες για την NLP ξεκινούν από την αυγή της επιστήμης των υπολογιστών. Οι προσεγγίσεις αυτές εστίαζαν στο ίδιο το συμβολικό επίπεδο και βασίζονταν σε κανόνες που προσπαθούσαν να υλοποιήσουν τη λογική και τη γνώση της γλώσσας ξεκινώντας από τον προσδιορισμό του ρόλου των διαφόρων μερών του λόγου. Οι κανόνες προσδιορίζονταν βάσει γλωσσικών μελετών και κωδικοποιούνταν με το χέρι, με βάση συνδυασμούς λέξεων ή άλλα χαρακτηριστικά.

Η ακρίβεια μπορεί να είναι υψηλή εάν οι κανόνες βελτιωθούν προσεκτικά από εμπειρογνώμονα. Αλλά η δημιουργία και η συντήρηση αυτών των κανόνων είναι δαπανηρή

Η φάση αυτή εξέπνευσε λόγω της αυξανόμενης πολυπλοκότητας του προβλήματος, ιδίως: της σημασίας του πλαισίου, της δυσκολίας κατανόησης όσων δεν λέγονται, των συνεπαγωγών, των αμφισημιών, της ειρωνείας, του σαρκασμού, των περιέργων ονομάτων και της μη τυποποιημένης γλώσσας. Τόσες πολλές πτυχές που για έναν άνθρωπο είναι προφανείς, επειδή υπάρχει κατανόηση πίσω από αυτές, αλλά που δεν μπορούν να μετατραπούν σε αναπαραγωγίμες λειτουργίες με τις μηχανές, που δεν έχουν ακόμη ισχυρή τεχνητή νοημοσύνη.

Υπήρξε λοιπόν μια εξέλιξη σε διαδοχικά κύματα, έτσι ώστε ένας αρχικός ενθουσιασμός, που γεννήθηκε από τη διαθεσιμότητα νέων υποσχόμενων τεχνικών, να ακολουθήσει μια περίοδος δυσπιστίας ως προς τη δυνατότητα επίλυσης ορισμένων προβλημάτων. Σύμφωνα με αυτό το σχήμα, όταν οι τεχνικές μηχανικής μάθησης άρχισαν να δίνουν καλά αποτελέσματα σε άλλους τομείς, αυτές χρησιμοποιήθηκαν και στον NLP, ιδίως η SVM, η λογιστική παλινδρόμηση και το τυχαίο δάσος, χρησιμοποιώντας διάσπαρτα διανύσματα ως δεδομένα, που αποτελούνται από δεκάδες ή εκατοντάδες σε μέγεθος, όπως θα δούμε αναλυτικότερα σύντομα.

Αν και τα αποτελέσματα μπορούν ήδη να ληφθούν με αυτόν τον τρόπο, αυτά μπορούν να βελτιωθούν με την ανάπτυξη ειδικών τεχνικών για αυτόν τον τύπο δεδομένων, ιδίως εκείνων που καταφέρνουν να συλλάβουν ορισμένες από τις ιδιαιτερότητές τους, όπως η αλληλουχία.

Θα επικεντρωθούμε στις πιο πρόσφατες προσεγγίσεις βαθιάς μάθησης. Συγκεκριμένα, σε αυτό το μάθημα θα δούμε ταξινομητές κειμένου.



Στην εργασία ταξινόμησης κειμένου, έχουμε ένα κείμενο t και ένα ορισμένο σύνολο κατηγοριών $C = \{c_1, c_2, c_3, \dots, c_n\}$ και πρέπει να καθορίσουμε την κατηγορία c στην οποία ανήκει το κείμενο t .

ceCwith



Επιβλεπόμενη μηχανική μάθηση με νευρωνικό δίκτυο

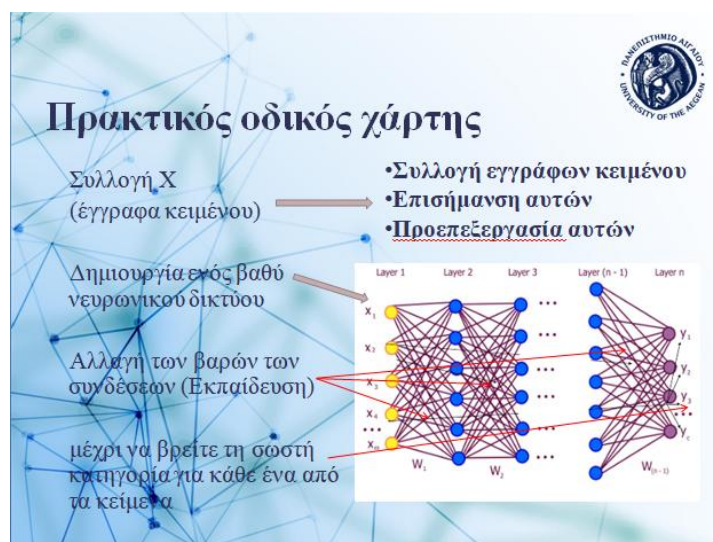
Έχουμε ορισμένα παραδείγματα $X = \{x_1, x_2, x_3, \dots, x_n\}$ (έγγραφα κειμένου)

τα επιθυμητά αποτελέσματα $Y = \{y_1, y_2, y_3, \dots, y_n\}$ (οι ετικέτες στις κατηγορίες μας)

$$Y = f(X)$$

πρέπει να βρούμε το f (ένα βαθύ νευρωνικό δίκτυο και τις συνδέσεις του) έτσι ώστε οι απαντήσεις να είναι ακριβώς αυτές που θέλουμε

Βρισκόμαστε στον τομέα της μηχανικής μάθησης με επίβλεψη. Επομένως, η κατασκευή ενός ταξινομητή με ένα νευρωνικό δίκτυο περιλαμβάνει την εκπαίδευση αυτού του δικτύου ώστε να ανταποκρίνεται σε παραδείγματα σύμφωνα με προκαθορισμένες απαντήσεις. Έτσι, όπως και στην περίπτωση άλλων τύπων ταξινομητών, μπορεί να συνοψιστεί στο ότι έχουμε έναν ορισμένο αριθμό παραδειγμάτων (X) των οποίων γνωρίζουμε τις απαντήσεις (y) που θα πρέπει να μάθει το δίκτυο. Η εκπαίδευση, και σε αυτή την περίπτωση, συνίσταται στην τροποποίηση των βαρών των συνδέσεων μέχρις ότου προτείνοντας τα παραδείγματα που έχουμε στη διάθεσή μας να μην παίρνουμε τις επιθυμητές απαντήσεις και, κυρίως, να βεβαιωθούμε ότι το μοντέλο αυτό γενικεύει, ώστε να δίνει σωστές απαντήσεις και με παραδείγματα που δεν έχει δει προηγουμένως.



Έτσι, σε μια πρώτη προσέγγιση, τα βήματά μας για να φτιάξουμε έναν ταξινομητή είναι τα εξής: λαμβάνουμε το X (τότε τα παραδείγματα των κειμένων μας), για καθένα από τα οποία ορίζεται η υπαγωγή του σε μια συγκεκριμένη κατηγορία, στη συνέχεια ορίζεται το y του.

Στη συνέχεια δημιουργούμε ένα βαθύ νευρωνικό δίκτυο που, ξεκινώντας από αυτά τα δεδομένα, παράγει τις αναμενόμενες ετικέτες.

Τέλος, εκπαιδεύουμε το δίκτυό μας, το οποίο συνίσταται στην τροποποίηση των βαρών των συνδέσεων μέχρι να λάβουμε τα επιθυμητά αποτελέσματα.

Τώρα θα δούμε όλα αυτά τα βήματα με περισσότερες λεπτομέρειες.

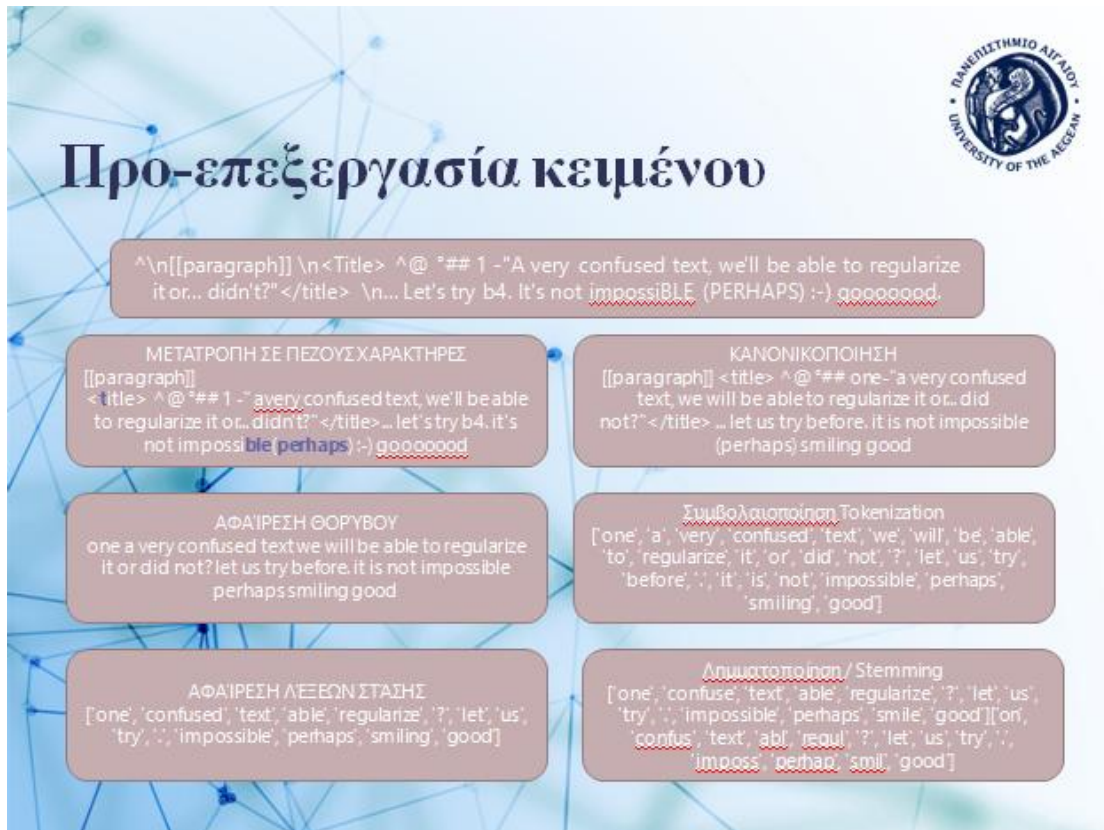


Έτσι, τα δεδομένα μας είναι κείμενα.

Υπάρχουν διάφοροι τρόποι για να τα αποκτήσουμε, τους οποίους δεν θα αναλύσουμε γιατί ξεφεύγουν από το πλαίσιο αυτού του μαθήματος και θα έπαιρναν πολύ χρόνο, αλλά θα τους απαριθμήσουμε. Στην πιο τυχερή, αλλά σπάνια, περίπτωση, μπορεί να τα έχουμε ήδη έτοιμα επειδή μας τα παρέχουν ή επειδή βρίσκουμε ένα έτοιμο σύνολο δεδομένων που μας ταιριάζει

Ανίχνευση στο διαδίκτυο: Αναζήτηση κειμένων στο Διαδίκτυο, με τη χρήση μιας αυτόματης διαδικασίας που μπορούμε επίσης να γράψουμε χάρη στις βιβλιοθήκες που απλοποιούν τη δημιουργία τους.

Ανάθεση του κατάλληλου για εμάς συνόλου δεδομένων σε μια εξειδικευμένη εταιρεία
Χειροκίνητη επιλογή κειμένων, μέσω φίλτρων, ερωτημάτων από σύνολα δεδομένων, επιλογής βιβλίων ή συμμετοχής πολλών ατόμων (επί πληρωμή ή με αντάλλαγμα κάποια δωρεάν υπηρεσία).



Όποιος κι αν είναι ο τρόπος με τον οποίο αποκτήσαμε το σύνολο δεδομένων μας, πρέπει να εκτελέσουμε κάποιες λειτουργίες σε αυτά τα κείμενα, οι οποίες ονομάζονται: προεπεξεργασία. Ανάλογα με τις ανάγκες μας και το είδος και την ποιότητα των κειμένων από τα οποία ξεκινάμε, μπορούμε να επιλέξουμε να εκτελέσουμε κάποιες λειτουργίες ή κάποιες άλλες.

Τώρα τις βλέπουμε όλες γρήγορα, στη συνέχεια θα επανέλθουμε λίγο πιο αναλυτικά βλέποντας επίσης πώς να τις κάνουμε. Η πρώτη είναι απλά να βάλουμε τα πάντα με **πεζά γράμματα**, ώστε να μη θεωρούμε μια λέξη διαφορετική όταν αρχίζει με κεφαλαίο γράμμα επειδή ακολουθεί μία τελεία. Στην πραγματικότητα, πρέπει να μειώσουμε όσο το δυνατόν περισσότερο τον αριθμό των διαφορετικών λέξεων που θα έχουμε να αντιμετωπίσουμε, χωρίς να χάσουμε πληροφορίες.

Το δεύτερο που βλέπουμε είναι η **κανονικοποίηση**. Η οποία στην πραγματικότητα αποτελείται από πολλές λειτουργίες και συνήθως, λανθασμένα, δεν λαμβάνεται ευρέως υπόψη. Πιθανώς επειδή ορισμένες από τις πράξεις από τις οποίες αποτελείται δεν είναι τυποποιημένες. Ωστόσο, η κανονικοποίηση περιλαμβάνει την αντικατάσταση των συνεπτυγμένων μορφών με το εκτεταμένο ισοδύναμό τους, την αντικατάσταση των συντομογραφιών, των συντμήσεων της καθομιλουμένης και της φωνητικής ορθογραφίας, με την αντίστοιχη έκφραση, τη διόρθωση των λαθών και την αναγωγή στην κανονική μορφή επίσης της φωνητικής ορθογραφίας και της συναισθηματικής έμφασης. Οι αριθμοί αντικαθίστανται επίσης με το αντίστοιχο κείμενό τους.

Η φάση της **αφαίρεσης του θορύβου** περιλαμβάνει, στην πραγματικότητα, την εξάλειψη των μη αλφαβητικών χαρακτήρων, συμπεριλαμβανομένων ή όχι των σημείων στίξης, ή μέρους αυτών. Στην περίπτωση του παραδείγματος, αφήνονται τα σημεία, συμπεριλαμβανομένων των ερωτηματικών και των θαυμαστικών.

Με το **tokenization** μετατρέπουμε το κείμενο σε μια λίστα λέξεων. Η **αφαίρεση των stop words** αφορά την αφαίρεση πολύ κοινών λέξεων, οι οποίες επομένως δεν συμβάλλουν στο να χαρακτηρίσουμε ένα κείμενο έναντι ενός άλλου. Έτσι, η αφαίρεσή τους σας επιτρέπει να έχετε να

ασχοληθείτε με λιγότερα στοιχεία χωρίς να χάσετε πληροφορίες. Τέλος, έχουμε τη **λημματοποίηση ή το stemming**. Οι δύο αυτές λειτουργίες είναι εναλλακτικές. Η πρώτη αφαιρεί όλες τις κλίσεις της λέξης και την επαναφέρει στη βασική της μορφή, η δεύτερη την οδηγεί στη ρίζα της.

Κωδικοποίηση προεπεξεργασίας κειμένου



ΜΕΤΑΤΡΟΠΗ ΣΕ ΠΕΖΟΥΣ ΧΑΡΑΚΤΗΡΕΣ <pre style="margin: 10px 0;">Text = text.lower()</pre>	ΚΑΝΟΝΙΚΟΠΟΙΗΣΗ <pre style="margin: 10px 0;">def normalization(text): text = text.replace(":-)", "smiling") text = text.replace("b4", "before") text = text.replace("otw", "on the way") text = re.sub("good", "good", text) text = text.replace("2maro", "tomorrow") text = cntrctfix(text) text = number2text(text) return text</pre>
ΑΦΑΙΡΕΣΗ ΘΟΡΥΒΟΥ <pre style="margin: 10px 0;">def noise_removal(text): text = re.sub("[\]\[\]*\]", "" ,text) text = re.sub("<\/A>[a-z]*>","",text) text = re.sub("\.", "" ,text) text = text.replace("...", "") ... text = " ".join(text.split())</pre>	Συμβολαιοποίηση/ Tokenization <pre style="margin: 10px 0;">text = text.split() import nltk text = nltk.word_tokenize(text) from keras.preprocessing.text import text_to_word_sequence text = text_to_word_sequence(text)</pre>
ΑΦΑΙΡΕΣΗ ΛΕΞΕΩΝ ΣΤΑΣΗΣ <pre style="margin: 10px 0;">from nltk.corpus import stopwords def stopwords_remove(text): new_words = [] for word in text: if word not in stopwords.words('english'): new_words.append(word) return new_words</pre>	Λημματοποίηση/ Stemming <pre style="margin: 10px 0;">from nltk.stem import WordNetLemmatizer def lemmatize(text): lemmatizer = WordNetLemmatizer() lemmas = [] for word in text: lemma = lemmatizer.lemmatize(word, pos='v') lemmas.append(lemma) return lemmas</pre>

Αναφέρουμε πώς να κάνετε αυτές τις λειτουργίες.

To lowercase, δεν χρειάζεται επεξήγηση και μπορεί να επιτευχθεί με την απλή κλήση της μεθόδου `lower()`.

Η κανονικοποίηση θα μπορούσε να ρυθμιστεί με αυτόν τον τρόπο, σας καλώ, ως άσκηση, να ολοκληρώσετε τη λειτουργία προσθέτοντας και άλλες συρρικνωμένες μορφές, συντμήσεις της καθομιλουμένης και φωνητική ορθογραφία. Επιπλέον, οι δύο τελευταίες γραμμές υπενθυμίζουν δύο άλλες συναρτήσεις που θα πρέπει να γράψετε. Οι εισαγωγές που θα δείτε στη συνέχεια αφορούν δύο βιβλιοθήκες που στην πραγματικότητα κάνουν όλη τη δουλειά. Η πρώτη μπορεί να χρησιμοποιηθεί για τη μετατροπή εκφράσεων όπως "you've" σε "you have" ή "he's" σε "he is". Η δεύτερη είναι για να μετατρέπει τους γραπτούς αριθμούς σε αριθμούς στο κειμενικό τους ισοδύναμο.

Η αφαίρεση του θορύβου μπορεί να επιτευχθεί με μια σειρά από κανονικές εκφράσεις και αντικαταστάσεις. Μπορούμε να βάλουμε τις πιο συχνές, αυτές που μας έρχονται στο μυαλό, αυτές που εντοπίζουμε στα κείμενά μας και σταδιακά θα αυξήσουμε τις διορθώσεις που θα μπορεί να κάνει η λειτουργία μας. Στο παράδειγμα, η πρώτη κανονική έκφραση μας επιτρέπει να εξαλείψουμε τις λέξεις που περικλείονται σε διπλές αγκύλες, αν είχαμε αυτή την ανάγκη, επειδή

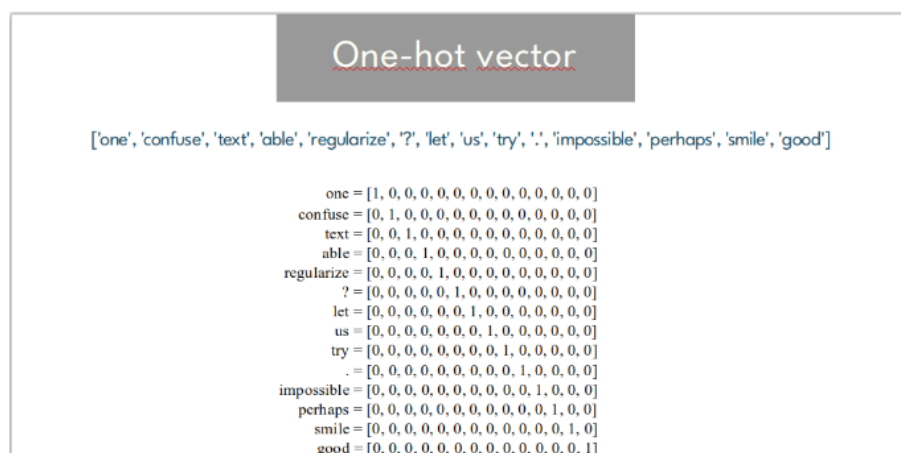
γνωρίζουμε ότι στο κείμενο που έχουμε στη διάθεσή μας υπάρχουν μέρη που γίνονται έτσι ώστε να μην έχουν να κάνουν με το πραγματικό κείμενο είναι ακριβώς. Η δεύτερη αφορά την αφαίρεση των ετικετών. Στην πραγματικότητα, αν κατεβάσαμε κείμενο από το διαδίκτυο χρησιμοποιώντας μια βιβλιοθήκη όπως η BeautifulSoup, έχει και αυτή μεθόδους όπως η get_text η οποία ήδη επιστρέφει το κείμενο χωρίς τις ετικέτες. Ήθελα όμως να βάλω αυτές τις δύο συγκεκριμένες διορθώσεις απλά για να δείξω πώς κατά περίπτωση μπορούμε να προσαρμόσουμε αυτή τη συνάρτηση ανάλογα με τις ανάγκες μας. Θα συμπληρωθεί και αυτή, προσθέτοντας τουλάχιστον τις πιο συχνές αντικαταστάσεις, ιδίως αυτές που θα γίνουν απαραίτητες στο κείμενό μας.

Όσον αφορά την **κωδικοποίηση**, σας παρουσιάζω τρεις μεθόδους (και μπορείτε να βρείτε και άλλες). Από την πιο άμεση διάσπαση, τη χρήση της βιβλιοθήκης nltk έως αυτή που υπάρχει στη βιβλιοθήκη keras. Η πρώτη είναι στην πραγματικότητα μη ενδεδειγμένη, καθώς απλά θα διαχωρίσει τις ομάδες των γραμμάτων με βάση την παρουσία κενών, οπότε πράγματα που συνήθως συναντάμε σε ένα κείμενο, όπως μια λέξη με ένα κόμμα συνημμένο, θα τα θεωρήσει ένα ενιαίο token, στο οποίο δεν είναι εντάξει, προφανώς.

Μπορούμε να κάνουμε την αφαίρεση των **stop words** χρησιμοποιώντας το σώμα του nltk, όπως βλέπουμε σε αυτή τη συνάρτηση.

Μπορούμε επίσης να κάνουμε τη **λημματοποίηση** χρησιμοποιώντας μια κλάση από τη βιβλιοθήκη της nltk με τον τρόπο που βλέπουμε σε αυτή τη συνάρτηση.

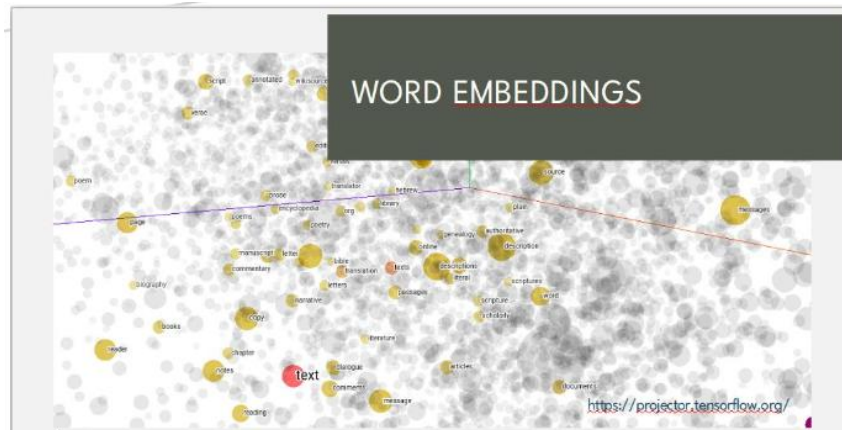
Έχουμε παραθέσει το **stemming** για λόγους πληρότητας, αλλά για όσα θα δούμε αργότερα δεν θα είναι απαραίτητο.



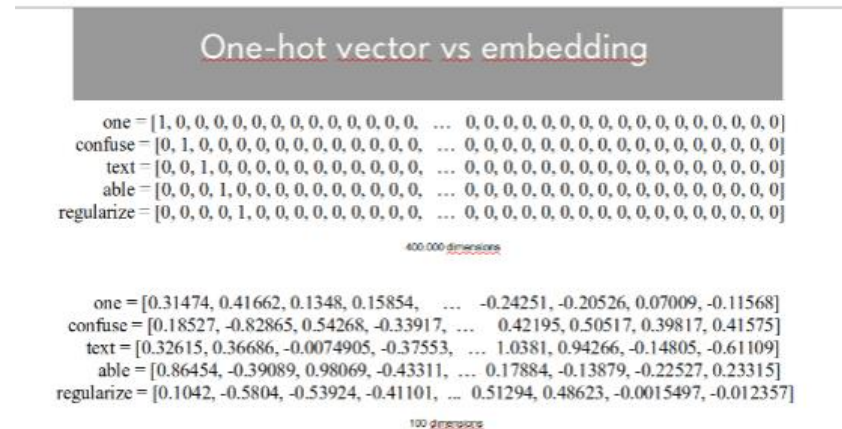
Ωστόσο, για να δημιουργηθεί ένα νευρωνικό δίκτυο, η είσοδος πρέπει να αποτελείται από αριθμούς και όχι από κείμενο. Επομένως, πρέπει να χρησιμοποιήσουμε κάποιον τρόπο για να μετατρέψουμε τη λίστα λέξεων της προεπεξεργασίας σε αριθμούς. Για μεγάλο χρονικό διάστημα αυτό γινόταν με τη μετατροπή των λέξεων σε 1-hot vector. Διανύσματα των οποίων τα στοιχεία είναι όλα μηδενικά εκτός από ένα που είναι σημαντικό.

Εδώ βλέπουμε πώς θα μπορούσαμε να έχουμε κάνει με τη λίστα που πήραμε νωρίτερα από την προεπεξεργασία μας. Προφανώς, όμως, ένα πραγματικό σύνολο δεδομένων που θα χρησιμοποιηθεί για μια πραγματική εργασία δεν θα αποτελείται μόνο από 14 στοιχεία. Πιθανότατα θα είχαμε να κάνουμε με ένα λεξιλόγιο εκατοντάδων χιλιάδων στοιχείων. Επομένως, τα διανύσματά μας θα πρέπει επίσης να έχουν μέγεθος εκατοντάδων χιλιάδων, και το ανώτερο στρώμα του δικτύου μας θα πρέπει επίσης να είναι τόσο μεγάλο.

Αυτός ήταν ο τρόπος που χρησιμοποιήθηκε περισσότερο.



Μέχρι την ανάπτυξη μιας νέας τεχνικής: της ενσωμάτωσης λέξεων.

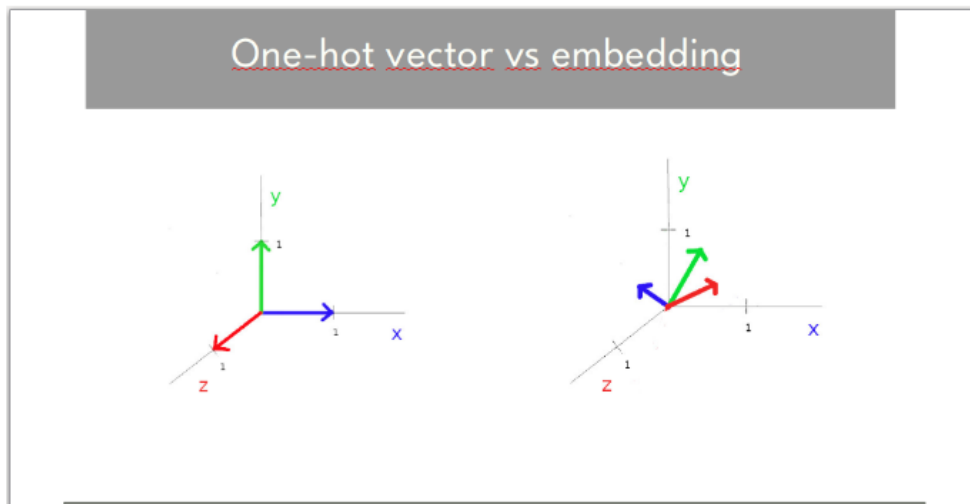


Μπορούμε να δούμε τις ενσωματώσεις, από μαθηματική άποψη, ως μια μεταφορά από ένα χώρο με σημαντικό αριθμό διαστάσεων (για παράδειγμα 100.000) σε έναν χώρο με πολύ μικρότερο αριθμό διαστάσεων (για παράδειγμα 200). Μια μαθηματική ενσωμάτωση από έναν χώρο με πολλές διαστάσεις ανά λέξη σε έναν συνεχή διανυσματικό χώρο με πολύ μικρότερη διάσταση.

Για παράδειγμα, αν κάνουμε τις πρώτες λέξεις της προεπεξεργασίας του πριν και φανταστούμε ότι αποτελούν μέρος ενός λεξικού, ας πούμε, 400.000 λέξεων, θα πρέπει να τις αναπαραστήσουμε με διανύσματα ενός χτυπήματος των 400.000 στοιχείων. Εναλλακτικά, με την ενσωμάτωση, μπορούμε να τις αναπαραστήσουμε με διανύσματα μόνο 100, 200 ή 300 διαστάσεων, όπως αυτά που βλέπουμε παρακάτω.

Αλλά πώς περνάτε από τα διανύσματα ενός φωτός σε συνεχή διανύσματα;

Και τι αντιπροσωπεύουν αυτοί οι αριθμοί;

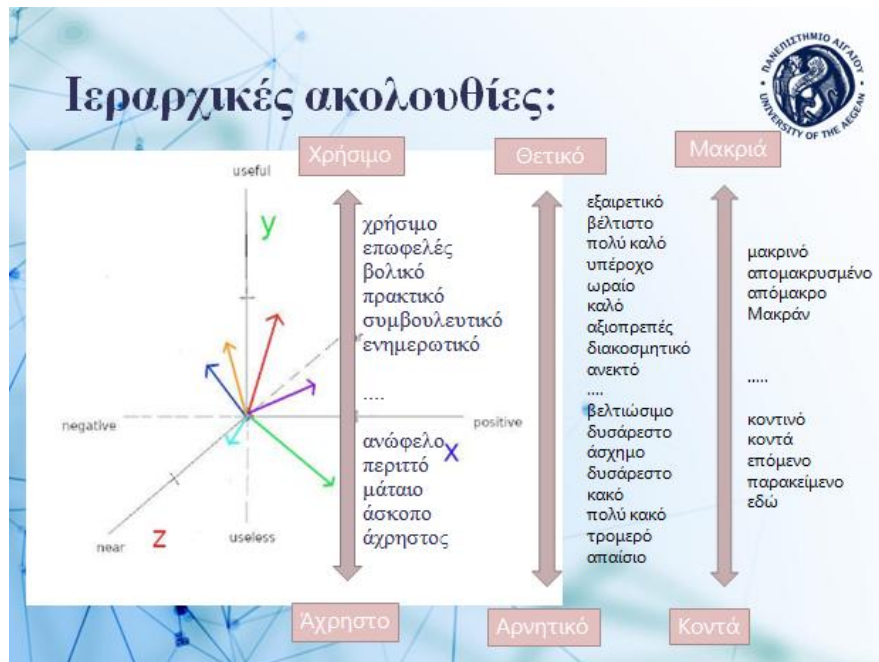


Το πλεονέκτημα δεν είναι μόνο ότι έχουμε περάσει σε έναν πολύ μικρότερο αριθμό και συνεπώς έχουμε να κάνουμε με έναν πιο εύχρηστο πίνακα, αλλά κυρίως ότι έχουμε ενσωματώσει τη σημασιολογία σε αυτά τα διανύσματα, τουλάχιστον εν μέρει. Στην πραγματικότητα, στην περίπτωση των διάσπαρτων διανυσμάτων κάθε διάνυσμα έχει την ίδια απόσταση μεταξύ τους ($\sqrt{2}$), ενώ στην περίπτωση των πυκνών διανυσμάτων οι αποστάσεις είναι διαφορετικές και μετά από μια καλή εκπαίδευση αντικατοπτρίζουν τη σημασιολογική εγγύτητα αυτών. Στην περίπτωση των 1-hot διανυσμάτων, δεν είχαμε πραγματική εγγύτητα και απόσταση μεταξύ των λέξεων, τα διαφορετικά διανύσματα ήταν κατηγορίες από μόνα τους, στα ενσωματωμένα διανύσματα έχουμε ένα συνεχές μακρύ δεκάδες ή εκατοντάδες διαφορετικά μεγέθη που κάνουν τις αποστάσεις μεταξύ διαφορετικών και νοηματικών λέξεων, οπότε εκπαιδεύοντας το δίκτυο σε συγκεκριμένες λέξεις, προετοιμάζουμε και το έδαφος για μια γενίκευση στις γειτονικές.



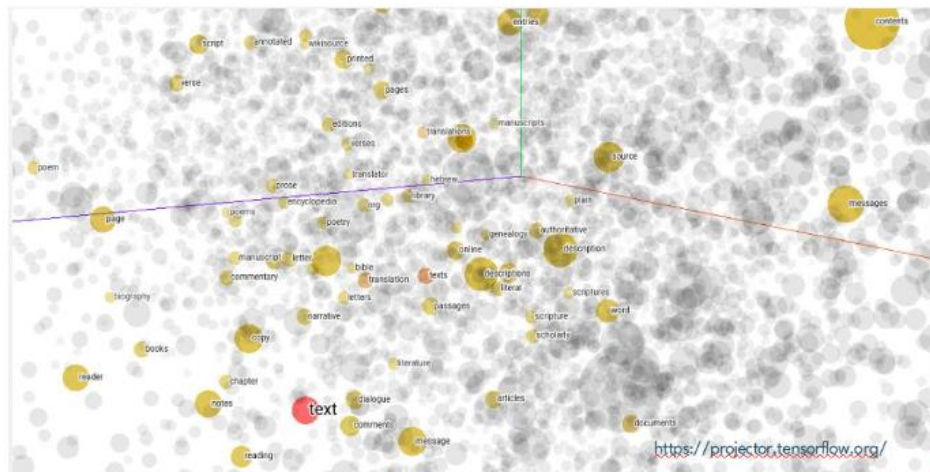
Αν θέλαμε να κάνουμε μόνο ανάλυση συναισθήματος, θα μπορούσαμε να ταξινομήσουμε κάθε λέξη κατά μήκος ενός άξονα που θα μας έλεγε αν μια συγκεκριμένη λέξη εκφράζει περισσότερο θετικό ή αρνητικό συναίσθημα. Έτσι θα μπορούσαμε να βάλουμε το *abominable* στο μέγιστο της αρνητικότητας και το *excellent* στο μέγιστο της θετικότητας, λίγο πιο κάτω θα βάζαμε το *good*, ακόμα πιο κάτω το *decent* και ούτω καθεξής...

Αλλά αυτή θα ήταν μια διάσταση, ίσως θα μπορούσε να είναι μια χαρά για το σκοπό αυτό, θα μπορούσαμε να ταξινομήσουμε όλες τις λέξεις που έχουν μια σαφή θετική ή αρνητική χροιά και να αγνοήσουμε όλες τις άλλες, ή να τις αποτιμήσουμε στο μηδέν σε αυτόν τον άξονα.



Θα μπορούσαμε να ταξινομήσουμε τις λέξεις ανάλογα με το αν υποδηλώνουν μεγαλύτερη ή μικρότερη χρησιμότητα, ή μεγαλύτερη ή μικρότερη εγγύτητα, ή σύμφωνα με άλλες διαστάσεις. Ωστόσο, αυτό θα ήταν μια πολύ απαιτητική εργασία και μπορεί να μην αντικατοπτρίζει το πραγματικό μέγεθος των λέξεων.

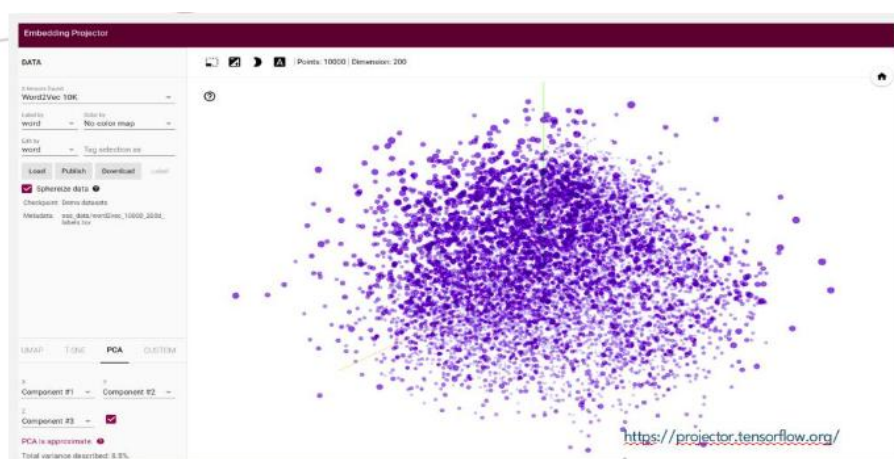
Φανταστείτε να μπορείτε να ταξινομήσετε τις λέξεις σύμφωνα με δεκάδες ή εκατοντάδες διαστάσεις αυτόματα. Θα μπορούσαμε να έχουμε τις λέξεις κατανεμημένες στον πολυδιάστατο χώρο με τρόπο που να αντικατοπτρίζει την ομοιότητα, την εναλλαξιμότητα μεταξύ των ίδιων των λέξεων. Αυτό που ξέρω ότι βρέθηκε σε πολύ κοντινά μέρη θα ήταν στην πραγματικότητα σχεδόν ισοδύναμο σύμφωνα με όλες αυτές τις διαστάσεις που έχουμε εξετάσει.



Εν τέλει, η ενσωμάτωση είναι ακριβώς αυτό, μια διάταξη σε σχέση με διαφορετικές διαστάσεις που καταφέρνει να συλλάβει περισσότερα χαρακτηριστικά. Ωστόσο, δεν πρέπει να αναζητήσουμε ρητά αυτά τα χαρακτηριστικά, θα είναι η ίδια η εκπαίδευση που θα προκαλέσει τη διάταξη των λέξεων με τέτοιο τρόπο ώστε παρόμοιες λέξεις να βρίσκονται κοντά σύμφωνα με κάποιες διαστάσεις.

Αυτό που βλέπουμε είναι μια προσπάθεια να δείξουμε την κατανομή ορισμένων λέξεων στο χώρο των ενσωματώσεων. Με την επανειλημμένη εφαρμογή της PCA, οι διαστάσεις έχουν μειωθεί σε 3. Κάθε σημείο (γκρι, κίτρινο ή κόκκινο) αντιπροσωπεύει μια λέξη. Εδώ βλέπουμε τη μεγέθυνση ενός τμήματος αυτού του χώρου στο οποίο εξηγούνται ορισμένες λέξεις που σχετίζονται με το κείμενο της λέξης.

Προφανώς, η σύνθλιψη των 200 αρχικών διαστάσεων στις 3 τελικές σημαίνει ότι χάνεται η αντίληψη της εγγύτητας κάποιων και άλλοι που δεν φαίνονται τόσο κοντά φαίνονται να είναι κοντά, αλλά μας επιτρέπει να καταλάβουμε πώς γίνεται.



Σας προσκαλώ να πάτε στη διεύθυνση που αναφέρεται (<https://projector.tensorflow.org/>) και να εξερευνήσετε ανεξάρτητα

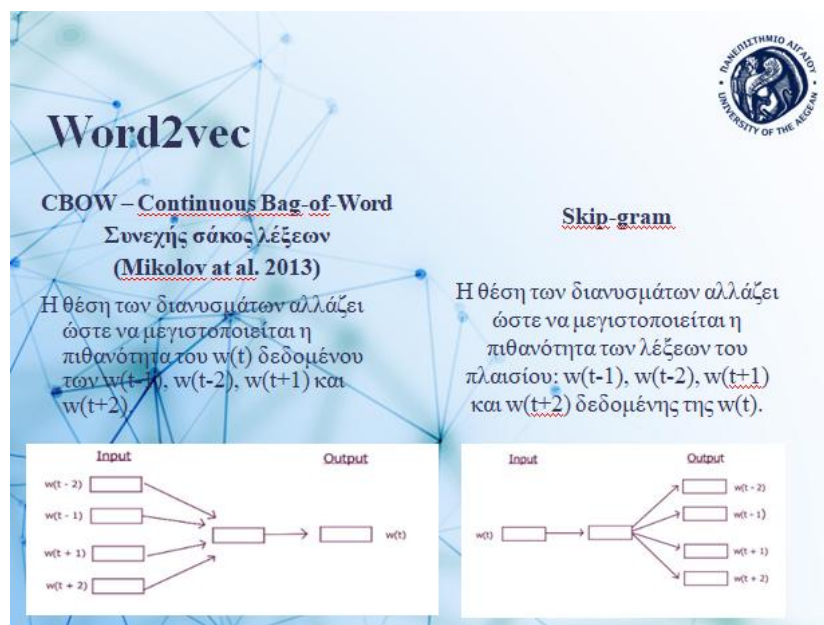


Τώρα έχουμε υποθέσει μόνο τι είναι η ενσωμάτωση.

Για να την κατανοήσουμε βαθύτερα πρέπει να κάνουμε δύο πράγματα:

- 1) να μάθουμε κάτι για το πώς γίνεται
- 2) να κάνουμε κάποιες εξερευνήσεις, ακόμα και να γράψουμε κάποιο κώδικα

Θα αναφέρουμε λοιπόν τρεις από τις κύριες τεχνικές και στη συνέχεια θα δούμε πώς να το φορτώσουμε και να υπολογίσουμε την εγγύτητα μεταξύ των διανυσμάτων που το συνθέτουν.



Η πρώτη μέθοδος που βλέπουμε, αναπτύχθηκε το 2013 από τον Mikolov και άλλους. Ξεκινούν με τυχαία τοποθέτηση των διανυσμάτων που αντιπροσωπεύουν τις λέξεις στον διανυσματικό χώρο. Σε αυτό το σημείο, αναλύοντας τεράστιες ποσότητες κειμένου, ο αλγόριθμος πρέπει να προβλέψει μια λέξη δεδομένου του πλαισίου της. Το πλαίσιο, σε αυτή την περίπτωση, δίνεται από ένα συγκεκριμένο παράθυρο λέξεων που προηγούνται και ακολουθούν τη δική μας, ας πούμε, για απλούστευση, ότι είναι δύο. Στην προκειμένη περίπτωση, λοιπόν, οι δύο λέξεις που προηγούνται και οι δύο λέξεις που ακολουθούν. Η ομοιότητα μεταξύ των διανυσμάτων που περιβάλλουν τη λέξη και της ίδιας της λέξης χρησιμοποιείται για τον υπολογισμό της πιθανότητας ότι η λέξη στο

κέντρο είναι αυτή. Τα διανύσματα προσαρμόζονται έτσι ώστε να μεγιστοποιείται αυτή η πιθανότητα.

Η δεύτερη μέθοδος κάνει το αντίθετο: δεδομένης μιας λέξης, η θέση των διανυσμάτων που αναπαριστούν τις λέξεις, αλλάζει ώστε να μεγιστοποιείται η πιθανότητα εμφάνισης λέξεων συμφραζομένων



GloVe

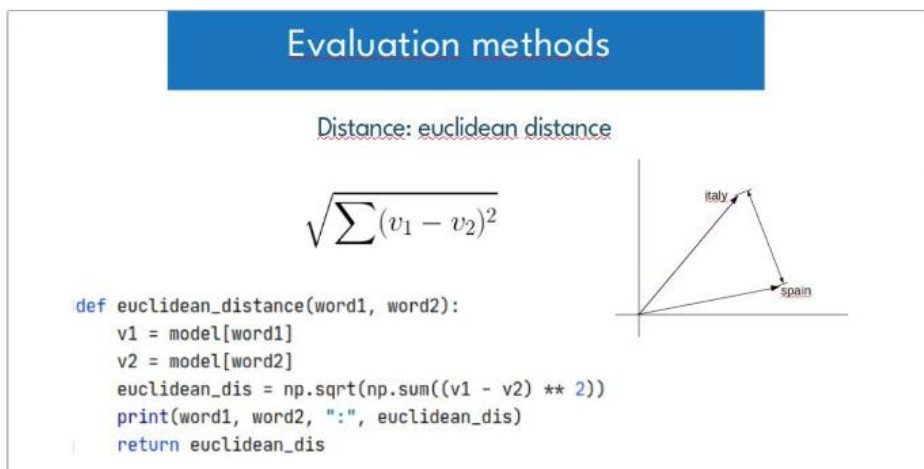
(Jeffrey Pennington, Richard Socher και Christopher D. Manning 2014)

Σταθμισμένα ελάχιστα τετράγωνα

Εκπαίδευση σε παγκόσμιες μετρήσεις συν-εμφάνισης λέξης-λέξης

Ελαχιστοποιεί τη διαφορά μεταξύ του τετραγωνικού γινομένου του διανύσματος μιας λέξης και της λέξης του πλαισίου της

Ένα χρόνο αργότερα, από τη δημοσίευση των μεθόδων που χρησιμοποιούνται από το Word2Vec, δημοσιεύθηκε μια μελέτη από ορισμένους ακαδημαϊκούς του Πανεπιστημίου του Στάνφορντ, οι οποίοι είχαν χρησιμοποιήσει μια άλλη μέθοδο για τη βαθμονόμηση των διανυσμάτων που αντιπροσωπεύουν τις λέξεις. Το GloVe αποτελείται από ένα σταθμισμένο μοντέλο ελαχίστων τετραγώνων που εκπαιδεύεται σε παγκόσμιες μετρήσεις συνύπαρξης λέξεων-λέξεων. Το GloVe ελαχιστοποιεί τη διαφορά μεταξύ του dotproduct των ενσωματώσεων μιας λέξης και της λέξης του περιβάλλοντός της.

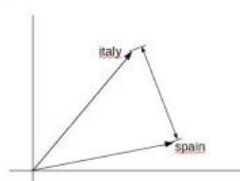


Evaluation methods

Distance: euclidean distance

$$\sqrt{\sum (v_1 - v_2)^2}$$

```
def euclidean_distance(word1, word2):  
    v1 = model[word1]  
    v2 = model[word2]  
    euclidean_dis = np.sqrt(np.sum((v1 - v2) ** 2))  
    print(word1, word2, ":", euclidean_dis)  
    return euclidean_dis
```



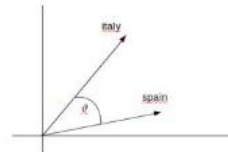
Ο πιο προφανής τρόπος μέτρησης της απόστασης μεταξύ δύο διανυσμάτων είναι ο υπολογισμός της ευκλείδειας απόστασής τους.

Το μέτρο αυτό θα είναι πολύ μικρότερο όσο πιο κοντά είναι τα διανύσματα. Μπορούμε να γράψουμε μια συνάρτηση που υπολογίζει αυτή την απόσταση, με τον τρόπο που βλέπουμε σε αυτό το κομμάτι κώδικα, όπου έχουμε ότι το μοντέλο είναι ένα λεξικό που επιστρέφει το διάνυσμα σε σχέση με τη λέξη που περνάει ως όρισμα.

Evaluation methods

Distance: angle between pairs of word vectors

$$\cos(\vartheta) = \frac{v_1 \cdot v_2}{\|v_1\|_2 \|v_2\|_2}$$



```
def cos_similarity(word1, word2):
    v1 = model[word1]
    v2 = model[word2]
    cos = np.dot(v1, v2) / (np.sqrt(np.dot(v1, v1)) * np.sqrt(np.dot(v2, v2)))
    print(word1, word2, ":", cos)
    return cos
```

Ένας άλλος τρόπος μέτρησης της απόστασης μεταξύ δύο διανυσμάτων είναι η μέτρηση του συνημιτόνου της γωνίας τους. Ο τύπος που μπορούμε να χρησιμοποιήσουμε είναι αυτός που βλέπουμε και μια πιθανή εφαρμογή του οποίου επίσης βλέπουμε.

Αν κάνουμε δοκιμές αντιλαμβανόμαστε ότι αυτό το μέτρο αποτυπώνει πολύ καλύτερα τη σημασιολογική εγγύτητα.



Μέθοδοι αξιολόγησης

Λεκτικές Αναλογίες

Ο βασιλιάς είναι για τη βασίλισσα όπως ο άνδρας για τη γυναίκα
 το μακρύ είναι για το κοντό όπως το υψηλό για το χαμηλό
 το πράσινο είναι για το κόκκινο όπως το μωβ για το κίτρινο
 η Ιταλία είναι για τη Ρώμη όπως η Ισπανία για τη Μαδρίτη

Υπάρχει όμως ένα άλλο μέτρο που μπορούμε να χρησιμοποιήσουμε και το οποίο είναι ακόμη πιο εκπληκτικό: μπορούμε να κάνουμε αναλογίες όπως αυτές!

Τώρα θα δούμε στην πράξη πώς μπορούμε να φτάσουμε σε αυτά τα αποτελέσματα.

Μια ενσωμάτωση θα είναι τόσο καλύτερη όσο περισσότερο ανταποκρίνεται σωστά σε αυτές τις αναλογίες.

Έτσι, θα μπορούσαμε επίσης να υποβάλουμε τις ίδιες αναλογίες σε διαφορετικές ενσωματώσεις και να αξιολογήσουμε ποια ανταποκρίνεται καλύτερα.

Μέθοδοι αξιολόγησης

Μέθοδοι αξιολόγησης στην πράξη

Περίγραμμα

- Δημιουργία της ενσωμάτωσης ή εύρεση μιας προ-εκπαιδευμένης
- Δημιουργία ενός λεξικού με τη λέξη ως κλειδί και το διάνυσμα ως τιμή
- Γράψιμο των συναρτήσεων ή χρήση μιας βιβλιοθήκης που τις διαθέτει
- Διασκεδάζουμε ελέγχοντας τις σχέσεις που γνωρίζουμε και ανακαλύπτοντας νέες

Τώρα θα δούμε πώς να εφαρμόσουμε αυτά τα μέτρα στην πράξη. Τα βήματα που θα ακολουθήσουμε θα είναι αυτά που παρατίθενται: Δημιουργία ενσωμάτωσης ή εύρεση μιας προ-εκπαιδευμένης- Δημιουργία λεξικού με τη λέξη ως κλειδί και το διάνυσμα ως τιμή- Γράψτε τις συναρτήσεις ή χρησιμοποιήστε μια βιβλιοθήκη που τις διαθέτει- Ελέγξτε τις σχέσεις που γνωρίζουμε και ανακαλύψτε νέες.

GloVe: Global Vectors for Word Representation

Jeffrey Pennington, Richard Socher, Christopher D. Manning

Introduction

GloVe is an unsupervised learning algorithm for obtaining vector representations for words. Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space.

Getting started (Code download)

- Download the latest [latest code](#) (licensed under the [Apache License, Version 2.0](#)). Look for "Clone or download"
- Unpack the files: `unzip master.zip`
- Compile the source: `cd GloVe-master && make`
- Run the demo script: `./demo.sh`
- Consult the included README for further usage details, or ask a [question](#)

Download pre-trained word vectors

- Pre-trained word vectors. This data is made available under the [Public Domain Dedication and License](#) v1.0 whose full text can be found at: <http://www.opendatacommons.org/licenses/pddl/1.0/>
 - [Wikipedia 2014 + Gigaword 5](#) (6B tokens, 400K vocab, uncased, 50d, 100d, 200d, & 300d vectors, 822 MB download): [glove.6B.zip](#)
 - Common Crawl (42B tokens, 19M vocab, uncased, 300d vectors, 175 GB download): [glove.42B.300d.zip](#)
 - Common Crawl (840B tokens, 2.2M vocab, cased, 300d vectors, 2.03 GB download): [glove.840B.300d.zip](#)
 - Twitter (2B tweets, 27B tokens, 1.2M vocab, uncased, 25d, 50d, 100d, & 200d vectors, 1.42 GB download): [glove.twitter.27B.zip](#)
- Ruby [script](#) for preprocessing Twitter data

(<https://nlp.stanford.edu/projects/glove/>)

Εάν χρησιμοποιήσουμε προ-εκπαιδευμένες ενσωματώσεις, είμαστε σε θέση να εισάγουμε στο δίκτυό μας ένα μέρος της πληροφορίας των λέξεων που μπορεί να συναχθεί από τη χρήση τους σε εκατομμύρια κείμενα που αναλύθηκαν για την εκπαίδευσή τους.

Στον ιστότοπο του Πανεπιστημίου του Στάνφορντ μπορούμε να βρούμε το GloVe.

Βλέπουμε ότι προσφέρουν 10 διαφορετικές ενσωματώσεις, από μία των 25 διαστάσεων έως μία των 300 διαστάσεων. Ωστόσο, η 25 έχει εκπαιδευτεί σε 2 δισεκατομμύρια λέξεις και έχει λεξιλόγιο 1,2 εκατομμυρίων λέξεων. Έτσι, αν είχαμε χρησιμοποιήσει διανύσματα ενός χτυπήματος, αυτά θα ήταν 1.200.000 διαστάσεις, αντ' αυτού μπορούμε να χρησιμοποιήσουμε διανύσματα μόνο 25 διαστάσεων και επιπλέον αυτά αντιπροσωπεύουν σημεία σε έναν 25-διάστατο χώρο που είναι πιο κοντά ή πιο μακριά σύμφωνα με κάποιες διαστάσεις ανάλογα με την πραγματική ομοιότητα όσον αφορά κάποιες πτυχές αυτών των λέξεων.

Το 300 διαστάσεων έχει εκπαιδευτεί σε 840 δισεκατομμύρια λέξεις και περιέχει 2,2 εκατομμύρια λέξεις!

Create a dictionary with the word as a key and the vector as a value

```
of -0.1529 -0.24279 0.89837 0.16996 0.53516 0.48784 -0.58826 -0.17982 -1.3581 0.42541 0.15377 0.24215 0.13474 0.41193 0.67843 -0.56418 0.42985 -0.012183 -0.
to -0.1897 0.85824 0.19884 -0.849184 -0.889737 0.21886 -0.54952 0.898377 -0.20135 0.34241 -0.892677 0.161 -0.13268 -0.2816 0.18737 -0.42959 0.96839 0.13972
... 0.89837 0.16996 0.53516 0.48784 -0.58826 -0.17982 -1.3581 0.42541 0.15377 0.24215 0.13474 0.41193 0.67843 -0.56418 0.42985 -0.012183 -0.
import numpy as np # The only necessary import is that of numpy, because the vectors will be saved in its format

MY_EMBEDDING_PATH = "../Embeddings/en/glove/" # Modify with the path in which we put the embeddings files
# In this case the path is relative, but an absolute path is also fine

def get_embeddings_dict(dim_embeddings: int) -> dict:
    """ Load the embeddings into memory
    storing it in a dictionary (embeddings_index)
    where each word corresponds to a vector that represents it """

    embeddings_index = {} # we create the empty dictionary that we will fill with the words and their vectors

    f = open(MY_EMBEDDING_PATH + "glove.6B."+str(dim_embeddings)+"d.txt", "r") # create a text flow in reading mode

    for line in f: # we read each line with a for on the stream
        values = line.split() # let's divide the row into its elements
        word = values[0] # the first element is the word
        coefficients = np.asarray(values[1:], dtype='float32') # the next elements are the values of the vector
        embeddings_index[word] = coefficients # let's add a key value pair to the dictionary
    f.close() # we close the stream
    return embeddings_index # return the embeddings dictionary
```

Στην κορυφή βλέπουμε μερικές γραμμές του αρχείου κειμένου που περιέχει την ενσωμάτωση. Κάθε γραμμή είναι περίπου μια λέξη. Το πρώτο στοιχείο είναι η λέξη και όλα τα επόμενα στοιχεία είναι οι συντελεστές του διανύσματος σε σχέση με αυτή τη λέξη. Πρέπει να δημιουργήσουμε ένα λεξικό στο οποίο το κλειδί είναι η λέξη και η τιμή το διάνυσμά της. Ξεκινώντας από αυτό το αρχείο κειμένου, πρέπει να δημιουργήσουμε ένα λεξικό στο οποίο το κλειδί είναι η λέξη και η τιμή το διάνυσμά της.

Ας κάνουμε την εισαγωγή της numpy που θα χρειαστούμε επειδή θα δημιουργήσουμε τα διανύσματα στη μορφή της numpy.

Δηλώνουμε μια σταθερά με τη διαδρομή του φακέλου στον οποίο τοποθετούμε τις ενσωματώσεις μας. Στο παράδειγμα έχουμε μια σχετική διαδρομή, σε ένα φάκελο στο ίδιο επίπεδο με αυτόν στον οποίο βρίσκεται το αρχείο pytho, οπότε ανεβαίνουμε ένα επίπεδο με την άνω τελεία και μετά

κατεβαίνουμε στο φάκελο 'Embeddings' και τους υποφακέλους του μέχρι αυτόν που περιέχει το αρχείο κειμένου με τα embeddings.

Δηλώνουμε τη συνάρτησή μας, ορίζουμε ως παράμετρο το μέγεθος των embeddings, με αυτόν τον τρόπο αν στο φάκελο υπάρχουν αρκετά που διαφέρουν μόνο στο μέγεθος μπορούμε να χρησιμοποιήσουμε την ίδια συνάρτηση για να φορτώσουμε το ένα ή το άλλο. Εναλλακτικά θα μπορούσαμε να ορίσουμε το πλήρες όνομα του αρχείου ως παράμετρο, ή ακόμα και τη διαδρομή που περιλαμβάνει το όνομα του αρχείου.

Δημιουργούμε το κενό λεξικό που θα γεμίσουμε με τις λέξεις και τα διανύσματά τους.

Δημιουργούμε μια περίπτωση του αντικειμένου που θα διαχειρίζεται τη ροή για την πρόσβαση στο αρχείο κειμένου, με την παράμετρο 'r', ανοίγουμε τη ροή για ανάγνωση.

Διαβάζουμε όλες τις γραμμές του αρχείου με έναν βρόχο for, ο οποίος σε κάθε βήμα θα τοποθετεί μια γραμμή του αρχείου στη μεταβλητή 'line'.

Ας χωρίσουμε τη γραμμή στα στοιχεία της με το split που, χωρίς παραμέτρους, θα χρησιμοποιήσει το κενό ως διαχωριστικό.

Το πρώτο στοιχείο θα είναι η λέξη, το επόμενο το διάνυσμα numpy. Ας προσθέσουμε ένα ζεύγος τιμών κλειδιών στο λεξικό.

Στο τέλος του κύκλου, κλείνουμε τη ροή και επιστρέφουμε το λεξικό των ενσωματώσεων.

Write the functions or use a library that has them

```
model = get_embeddings_dict(100) # Let's instantiate the dictionary with the previous function

def euclidean_distance(word1, word2):
    v1 = model[word1] # we create the vectors related to the words passed as parameters
    v2 = model[word2]
    euclidean_dis = np.sqrt(np.sum((v1 - v2) ** 2)) # The formula of Euclidean distance
    print(word1, word2, ":", euclidean_dis) # Print the results
    return euclidean_dis

def cos_similarity(word1, word2):
    v1 = model[word1] # we create the vectors related to the words passed as parameters
    v2 = model[word2]
    cos = np.dot(v1, v2) / (np.sqrt(np.dot(v1, v1)) * np.sqrt(np.dot(v2, v2))) # The cosine formula
    print(word1, word2, ":", cos) # Print the results
    return cos
```

Έχουμε ήδη δει αυτές τις συναρτήσεις όταν είδαμε μαθηματικούς τύπους για τη μέτρηση της απόστασης μεταξύ διανυσμάτων. Τώρα ας τις δούμε πιο αναλυτικά, αν και είναι πολύ απλές.

Δημιουργούμε τα διανύσματα που σχετίζονται με τις λέξεις που περνούν ως παράμετροι.

Υλοποιούμε τον τύπο της ευκλείδειας απόστασης αξιοποιώντας τις συναρτήσεις με τα διανύσματα numpy.

Η άλλη συνάρτηση είναι ανάλογη με μόνη διαφορά τον τύπο.

Having fun checking relationships we know and discovering new ones

```
euclidean_distance("bee", "ant")
euclidean_distance("venice", "florence")
euclidean_distance("spain", "italy")
euclidean_distance("boy", "girl")

cos_similarity("bee", "ant")
cos_similarity("venice", "florence")
cos_similarity("spain", "italy")
cos_similarity("boy", "girl")
```

bee ant	: 5.718665
venice florence	: 4.210178
spain italy	: 4.2352805
boy girl	: 2.3141153
bee ant	: 0.45103332
venice florence	: 0.68192357
spain italy	: 0.7746187
boy girl	: 0.91757303

Καλούμε τις συναρτήσεις μας δοκιμάζοντας ζεύγη λέξεων που γνωρίζουμε ήδη ότι είναι περισσότερο ή λιγότερο παρόμοιες ή τις χρησιμοποιούμε απλώς για να δούμε πόσο κοντά είναι δύο λέξεις.

Όπως αναφέραμε προηγουμένως, η ευκλείδεια απόσταση είναι μικρότερη όσο πιο κοντά είναι οι λέξεις, μέχρι και μηδενική αν ο υπολογισμός γίνεται στην ίδια τη λέξη. Το συνημίτονο των δύο διανυσμάτων αντίθετα αυξάνεται με την αύξηση της ομοιότητας, οπότε θα έχουμε ένα υπολογίζοντας το συνημίτονο του ίδιου διανύσματος, αφού σε αυτή την περίπτωση η γωνία θα είναι 0°

Use Gensim library for analogies

```
# imports from gensim
from gensim.models import KeyedVectors
from gensim.scripts.glove2word2vec import glove2word2vec

MY_EMBEDDING_PATH = "/home/user/pythonprojects/Embeddings/en/glove/" # Change with the absolute path in which you put the embeddings files
EMBEDDING_DIM = 100

glove_file_path = MY_EMBEDDING_PATH + "glove.6B."+str(EMBEDDING_DIM)+"d.txt" # the path of glove embedding
word2vec_glove_file_path = MY_EMBEDDING_PATH + "glove.6B."+str(EMBEDDING_DIM)+"d.word2vec.txt" # the path in which to store the new format
glove2word2vec(glove_file_path, word2vec_glove_file_path) # convert the glove format in word2vec format and save it

model = KeyedVectors.load_word2vec_format(word2vec_glove_file_path) # load embedding vectors

print("man:", model.most_similar("man")) # prints the names of the
print("boy:", model.most_similar("boy"))
print("sea:", model.most_similar("sea"))
print(model.most_similar(positive=["woman", "king"], negative=["man"])) # prints the names of the words whose vectors are closer to woman and king, but far from man
man : king = woman : [['queen', 0.7498541283687483], ('monarch', 0.6843388928839551), ('throne', 0.6755735874176025), ('daughter', 0.6594556578853181), ('princess', 0.6528534793799438),

def analogy(mw1, pw1, pw2):
    result = model.most_similar(positive=[pw1, pw2], negative=[mw1]) # gets the names of the words whose vectors are closer to pw1 and pw2, but far from mw1
    print(mw1, ":", pw1, ":", pw2, ":", result) # prints the results
    return result[0][0]

analogy("man", "king", "woman")
analogy("man", "woman", "king")
analogy("beautiful", "ugly", "good")
analogy("italy", "rome", "spain")

man : king = woman : [['queen', 0.7498541283687483], ('monarch', 0.6843388928839551), ('throne', 0.6755735874176025), ('daughter', 0.6594556578853181), ('princess', 0.6528534793799438), ('per
man : woman = king : [['queen', 0.7498541283687483], ('monarch', 0.6843388928839551), ('throne', 0.6755735874176025), ('daughter', 0.6594556578853181), ('th
beautiful : ugly = good : [['bad', 0.7296661139534546], ('worse', 0.6887085095895359), ('naughty', 0.667994499286543), ('thing', 0.6592233188999756), ('thi
italy : rome = spain : [['madrid', 0.7835128474235935], ('seville', 0.6557359899388123), ('paris', 0.6558979614257812), ('barcelona', 0.617169976234436),
```

Εισάγουμε από το gensim τις κλάσεις που θα χρειαστούμε και που θα δούμε αργότερα.

Ορίζουμε τις σταθερές μας για το μονοπάτι της ενσωμάτωσης και για τη διάσταση της ενσωμάτωσης.

Χρησιμοποιούμε το σενάριο υποστήριξης του gensim 'glove2word2vec' για να μετατρέψουμε το αρχείο από τη μορφή glove στη μορφή word2vec. Το χρειαζόμαστε αυτό επειδή είναι η μορφή που περιμένει το gensim στις κλάσεις και τις συναρτήσεις του. Αυτό το σενάριο αποθηκεύει ένα νέο αρχείο στη μορφή word2vec και με την επόμενη εντολή το ξαναφορτώνουμε. Έτσι αργότερα θα

μπορούσαμε να παραλείψουμε αυτό το βήμα μετατροπής και να χρησιμοποιήσουμε απευθείας αυτό που μόλις αποθηκεύτηκε.

Δημιουργήστε μια περίπτωση της κλάσης 'gensim.models.keyedvectors.Word2Vec KeyedVectors' που περιέχει τις πληροφορίες των διανυσμάτων και διάφορες μεθόδους για τη λειτουργία τους.

Με τη μέθοδο `most_similar` βρίσκουμε όλα τα διανύσματα που μοιάζουν περισσότερο με το υποδεικνυόμενο. Βλέπουμε ότι στην περίπτωση του αγοριού παίρνουμε το διάνυσμα κορίτσι ως το πιο κοντινό διάνυσμα και η ενδεικνυόμενη τιμή είναι ακριβώς ίση με το συνημίτονο που υπολογίσαμε μεταξύ αγοριού και κοριτσιού με τη συνάρτηση που γράψαμε.

Η συνάρτηση αυτή μπορεί να χρησιμοποιηθεί και με άλλους τρόπους, υποδεικνύοντας διανύσματα που θέλουμε να είναι "θετικοί" γείτονες και διανύσματα που πρέπει να είναι "αρνητικοί" γείτονες. Με αυτόν τον τρόπο κάνουμε μια πολύ πιο σύνθετη λειτουργία με εκπληκτικά αποτελέσματα. Είναι σαν να προσθέτουμε στη 'γυναίκα' τη βασιλεία και να αφαιρούμε την αρρενωπότητα. Και τι θα περιμέναμε ως αποτέλεσμα μιας τέτοιας λειτουργίας;

Το αποτέλεσμα είναι ακριβώς αυτό που περιμέναμε και βλέπουμε επίσης περαιτέρω λιγότερο σχετικά αποτελέσματα.

Μπορούμε επίσης να εκφράσουμε αυτό το ίδιο αποτέλεσμα ως αναλογία, στην πραγματικότητα είναι σαν να λέμε ότι ο άνδρας είναι βασιλιάς όπως η γυναίκα είναι βασίλισσα. Μπορούμε λοιπόν να γράψουμε μια συνάρτηση που κάνει αναλογίες χρησιμοποιώντας τη συνάρτηση '`most_similar`' με αυτόν τον τρόπο.

Μπορείτε να δοκιμάσετε και άλλες και θα δείτε ότι η απάντηση είναι σχεδόν πάντα η αναμενόμενη



Ταξινόμηση κειμένου

Πού βρισκόμαστε:

- Εισαγωγή στο NLP και Ιστορικό:
 - Τι είναι
 - Προηγούμενες προσεγγίσεις
- Ταξινόμηση κειμένου:
 - Συλλογή και επισήμανση δεδομένων
 - Προεπεξεργασία
 - Ενσωμάτωση λέξεων
 - Τι είναι αυτό
 - Ανάλυση και χρήση
 - Κατασκευή μοντέλου
 - Εκπαίδευση
 - Δοκιμή, αξιολόγηση, προσαρμογή
 - Αποθήκευση, χρήση

Ας συνεχίσουμε το θέμα της ταξινόμησης των κειμένων.

Αναφέραμε τον τρόπο δημιουργίας ενός συνόλου δεδομένων. Είδαμε τις διάφορες διαδικασίες προεπεξεργασίας που μπορούν να γίνουν. Σε αυτό το σημείο είχαμε ότι η είσοδός μας ήταν λέξεις που έπρεπε να μετασχηματιστούν με αποτελεσματικό τρόπο ώστε να χρησιμοποιηθούν στη βαθιά μάθηση. Είδαμε ότι αρχικά χρησιμοποιήθηκαν διανύσματα one-hot και είδαμε τα πλεονεκτήματα της ενσωμάτωσης σε σχέση με αυτά. Τώρα ας ξεκινήσουμε από την υλοποίηση του μοντέλου μας.



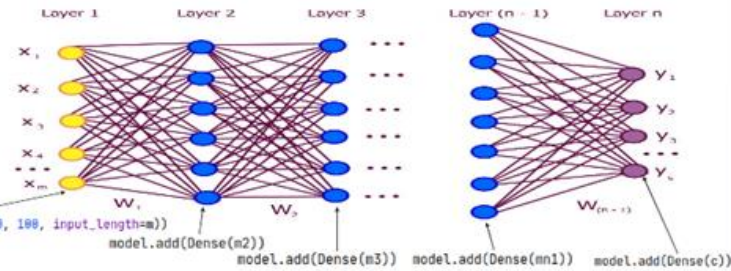
Ταξινόμηση κειμένου

Εχουμε ένα κείμενο t ένα ορισμένο σύνολο κατηγοριών $C = \{c_1, c_2, c_3, \dots, c_n\}$ πρέπει να καθορίσουμε την κατηγορία c_n στην οποία ανήκει το t

```

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Embedding

model = Sequential()
model.add(Embedding(50000, 100, input_length=m))
model.add(Dense(m2))
model.add(Dense(m3))
model.add(Dense(mn1))
model.add(Dense(c))
                    
```



Επομένως το πρώτο στρώμα θα είναι τύπου ενσωμάτωσης. Οι περισσότερες βιβλιοθήκες διαθέτουν ήδη ένα τέτοιο στρώμα. Αν χρησιμοποιήσουμε την keras, όπως θα κάνουμε στις καθοδηγούμενες ασκήσεις μας, το μοντέλο μπορεί να δημιουργηθεί με την ενσάρκωση ενός αντικειμένου τύπου Sequential και την κλήση της μεθόδου add στην οποία περνάμε τον τύπο του στρώματος που θέλουμε να προσθέσουμε.

Όταν προσθέτουμε το στρώμα Ενσωμάτωσης καθορίζουμε επίσης το μέγεθος του λεξικού (+1), καθώς αυτό καθορίζει μία από τις διαστάσεις του στρώματος και το μήκος των διανυσμάτων, επιπλέον πρέπει να καθορίσουμε το μήκος της ακολουθίας, δηλαδή από πόσες λέξεις αποτελούνται τα έγγραφά μας.

Στη συνέχεια, προσθέτουμε τα άλλα επίπεδα που μπορούμε να φτιάξουμε με το μέγεθος που θεωρούμε καταλληλότερο, μέχρι το τελευταίο που θα είναι ακριβώς ο αριθμός των κατηγοριών μεταξύ των οποίων το δίκτυο πρέπει να κάνει διάκριση. Εκτός από την περίπτωση στην οποία πρέπει να κάνει διάκριση μεταξύ δύο, στην οποία το τελευταίο στρώμα θα αποτελείται από έναν μόνο νευρώνα.



Ταξινόμηση κειμένου - Κατασκευή μοντέλου

```

model = Sequential() # Let's instantiate Sequential
model.add(Embedding(len(tokenizer.word_index) + 1, DIM_EMBEDDING, input_length=INPUT_LENGTH, trainable=True, name="Embedding_Layer"))
model.add(Dense(m1, name="Layer1")) # We add the various layers
model.add(Dense(m2, name="Layer2")) # We also set a name
model.add(Flatten(name="No_Layer")) # With the Flatten layer we go from a two-dimensional layer to a one-dimensional layer
model.add(Dense(m3, name="Layer3"))
model.add(Dense(mn1, name="Layer_n-1"))
model.add(Dense(1, activation='sigmoid', name="Output_Layer")) # Composed of a single neuron because we have to discriminate only be
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['acc']) # We create the model defining: optimizer, loss and met
print(model.summary())

```

Layer (type)	Output Shape	Param #
Embedding_Layer (Embedding)	(None, 25, 100)	524600
Layer1 (Dense)	(None, 25, 150)	15150
Layer2 (Dense)	(None, 25, 120)	19520
No_Layer (Flatten)	(None, 5200)	0
Layer3 (Dense)	(None, 80)	256000

Ας δούμε ένα παράδειγμα δημιουργίας και εκπαίδευσης ενός βαθύ νευρωνικού δικτύου

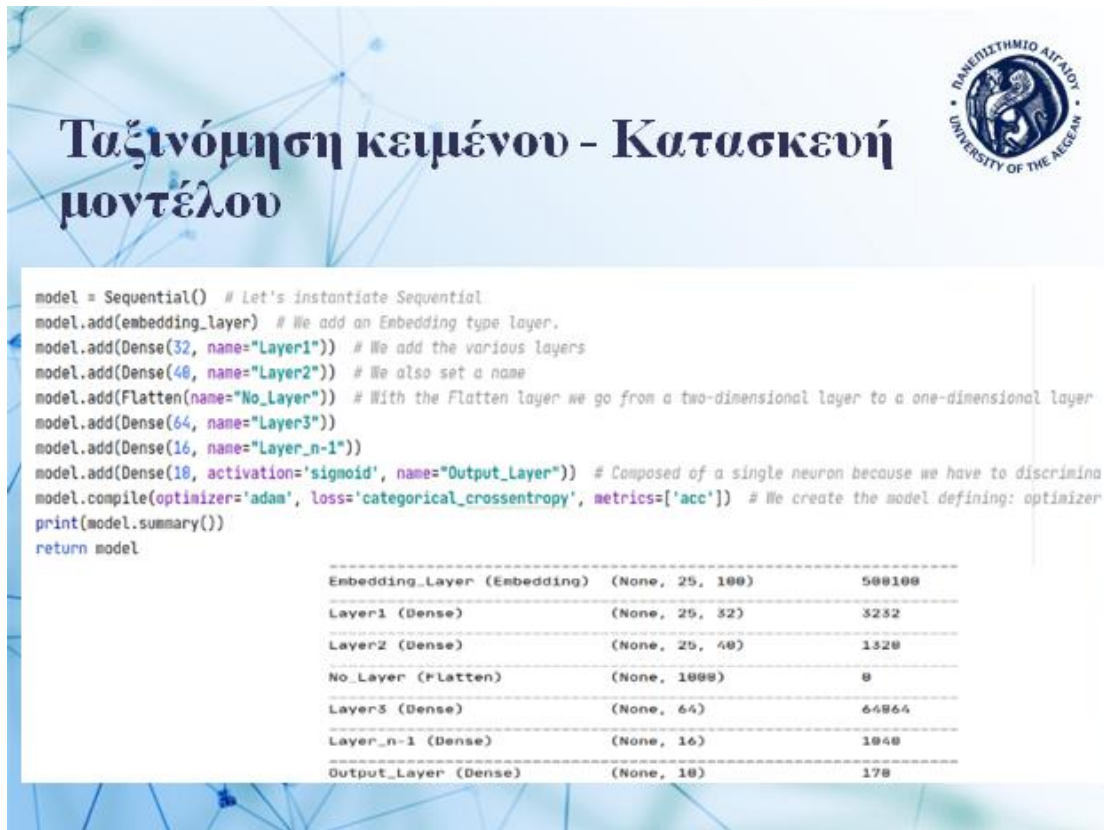
Πρώτα απ' όλα, ας ενσαρκώσουμε την Sequential. Στη συνέχεια, προσθέτουμε ένα επίπεδο τύπου Embedding. Πρέπει να ορίσουμε το μέγεθός του, το ένα δίνεται από τον αριθμό των λέξεων του λεξικού, δηλαδή από τον αριθμό των λέξεων που έχουμε αποφασίσει να χρησιμοποιήσουμε λαμβάνοντας υπόψη κάποιους παράγοντες. Πρώτα απ' όλα, η χρήση πολλών συνεπάγεται υψηλότερο κόστος επεξεργασίας και υπάρχει ο κίνδυνος προσθήκης θορύβου. Προσθέτοντας πολύ λίγες, υπάρχει κίνδυνος να παραλείψουμε πάρα πολλές λέξεις από τα έγγραφα που θα υποβάλουμε. Ένας καλός συμβιβασμός, τουλάχιστον προς το παρόν, θα μπορούσε να είναι να πάρουμε όλες τις λέξεις που βρίσκονται στα σύνολα δεδομένων που χρησιμοποιούμε, απορρίπτοντας ενδεχομένως τις λιγότερο συχνές, αν αυτές υπερβαίνουν έναν ορισμένο αριθμό. Η δεύτερη παράμετρος δηλώνει το μέγιστο μήκος των κειμένων. Η επόμενη παράμετρος δηλώνει αν θέλουμε τα διανύσματα που το αποτελούν και που αντιπροσωπεύουν τις λέξεις να αλλάζουν ή όχι κατά τη διάρκεια της εκπαίδευσης. Στην προκειμένη περίπτωση, δεδομένου ότι δεν χρησιμοποιούμε προ-εκπαιδευμένη ενσωμάτωση, δεν θα είχε νόημα να θέσουμε αυτή την παράμετρο σε false.

Στη συνέχεια προσθέτουμε τα διάφορα επίπεδα, όπως είδαμε στην προηγούμενη εικόνα.

Με το επίπεδο Flatten μεταβαίνουμε από ένα δισδιάστατο επίπεδο σε ένα μονοδιάστατο, όπως μπορούμε να δούμε ξεκάθαρα από τη συνοπτική εκτύπωση.

Το τελικό στρώμα θα αποτελείται από έναν μόνο νευρώνα καθώς πρέπει να διακρίνουμε μόνο μεταξύ δύο κατηγοριών: θετική ή αρνητική, οι οποίες θα κωδικοποιούνται από την κατάσταση απόκρισης 1 ή 0 αυτού του στρώματος.

Με τη μεταγλώττιση δημιουργούμε το μοντέλο που ορίζει, τον βελτιστοποιητή, την απώλεια και τη μετρική για την αξιολόγηση, η οποία επιβάλλεται με το `binary_crossentropy` για την κατηγοριοποίηση μεταξύ δύο κατηγοριών.



Ταξινόμηση κειμένου - Κατασκευή μοντέλου

```

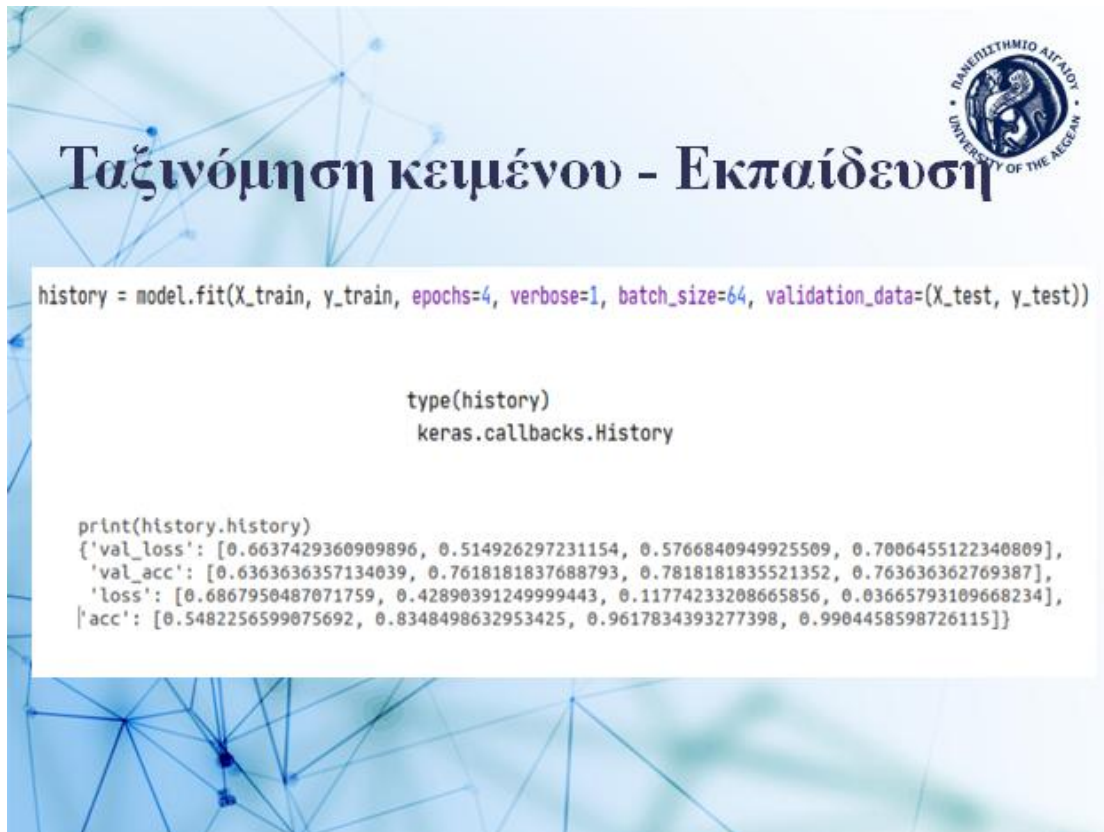
model = Sequential() # Let's instantiate Sequential
model.add(embedding_layer) # We add an Embedding type layer.
model.add(Dense(32, name="Layer1")) # We add the various layers
model.add(Dense(48, name="Layer2")) # We also set a name
model.add(Flatten(name="No_Layer")) # With the Flatten layer we go from a two-dimensional layer to a one-dimensional layer
model.add(Dense(64, name="Layer3"))
model.add(Dense(16, name="Layer_n-1"))
model.add(Dense(18, activation='sigmoid', name="Output_Layer")) # Composed of a single neuron because we have to discrimina
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['acc']) # We create the model defining: optimizer
print(model.summary())
return model
    
```

Layer (type)	Output Shape	Param #
Embedding_Layer (Embedding)	(None, 25, 100)	500100
Layer1 (Dense)	(None, 25, 32)	3232
Layer2 (Dense)	(None, 25, 48)	1520
No_Layer (Flatten)	(None, 1000)	0
Layer3 (Dense)	(None, 64)	64064
Layer_n-1 (Dense)	(None, 16)	1040
Output_Layer (Dense)	(None, 18)	178

Αν έπρεπε να κατηγοριοποιήσουμε μεταξύ πολλαπλών κατηγοριών, θα έπρεπε να ορίσουμε μια διαφορετική απώλεια. Στην περίπτωση που βλέπουμε εδώ (στην οποία το δίκτυο πρέπει να διακρίνει μεταξύ 10 κατηγοριών) ορίσαμε ως απώλεια την `categorical_crossentropy`.

Επομένως, το τελικό στρώμα θα αποτελείται από 10 νευρώνες και οι ετικέτες πρέπει να αποτελούνται από διανύσματα 10 στοιχείων με ένα χτύπημα.

Ας επιστρέψουμε όμως στην προηγούμενη περίπτωση και ας προχωρήσουμε παρακάτω.



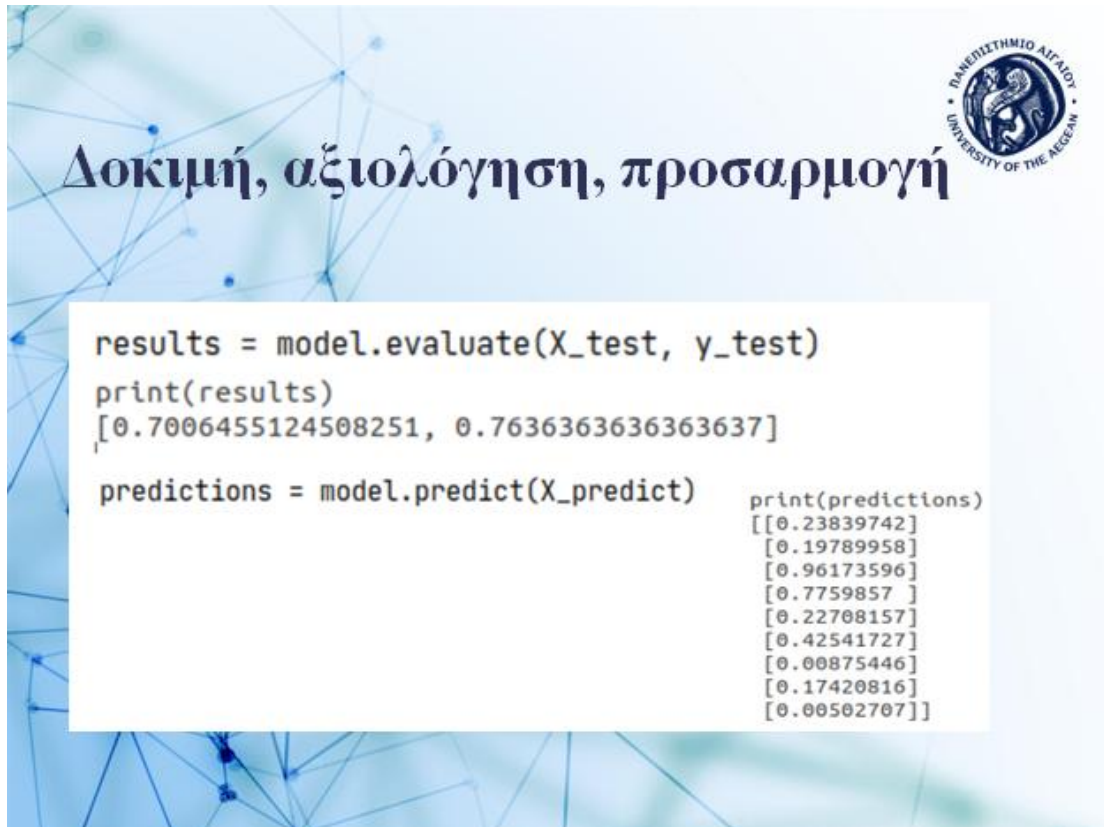
Αφού συναρμολογήσουμε και καταρτίσουμε το μοντέλο, μπορούμε να προχωρήσουμε στη φάση της εκπαίδευσης. Εδώ βλέπουμε ένα παράδειγμα της δήλωσης fit που στο keras μας επιτρέπει να την πραγματοποιήσουμε. Στο παράδειγμα του περνάμε 2 σύνολα δεδομένων με αντίστοιχες ετικέτες. Το πρώτο ζεύγος, X_train και y_train είναι τα παραδείγματα (και οι σχετικές ετικέτες) που θα χρησιμοποιήσουμε για την εκπαίδευση, το άλλο (X_test και y_test), το οποίο είναι προαιρετικό και το οποίο πρέπει να περάσουμε ως πλειάδα για την παράμετρο validation_data, είναι τα παραδείγματα που θα χρησιμοποιεί το δίκτυο σε κάθε εποχή για να επικυρώσει το δίκτυο, δηλαδή να ελέγξει πώς θα συμπεριφερθεί με παραδείγματα που δεν έχει δει ποτέ, στα οποία δεν είχε εκπαιδευτεί. Η παράμετρος epochs δηλώνει πόσες εποχές θα διαρκέσει η εκπαίδευση. Το Verbose υποδεικνύει το επιθυμητό επίπεδο πληροφοριών, το 1 είναι ένας καλός συμβιβασμός, μας ενημερώνει για την εξέλιξη της εκπαίδευσης χωρίς να μας κατακλύζει με υπερβολικό όγκο πληροφοριών.

Τέλος, η παράμετρος batch_size υποδεικνύει το πλήθος των παραδειγμάτων που θα υποβληθούν σε παράλληλη επεξεργασία, στο ίδιο βήμα.

Η μέθοδος fit επιστρέφει ένα αντικείμενο τύπου keras.callbacks.History το οποίο περιέχει διάφορες πληροφορίες σχετικά με την εξέλιξη της εκπαίδευσης.

Συγκεκριμένα, το πεδίο history του περιέχει, στην περίπτωση του παραδείγματος, τις μετρήσεις της ακρίβειας και της απώλειας κατά τη διάρκεια των τεσσάρων εποχών τόσο στο trainset όσο και στο testset.

Τα δεδομένα αυτά μπορούν επίσης να χρησιμοποιηθούν για την εκτύπωση γραφικών παραστάσεων ή για τον υπολογισμό άλλων μετρήσεων.



Αξιολογώντας τα αποτελέσματα, θα μπορούσατε να αποφασίσετε να κάνετε κάποιες αλλαγές στο μοντέλο, σε κάποιες μεταπαραμέτρους ή στον αριθμό των εποχών. Μόνο στο τέλος θα κάνουμε μια τελική δοκιμή εντελώς ανεξάρτητη από το train.

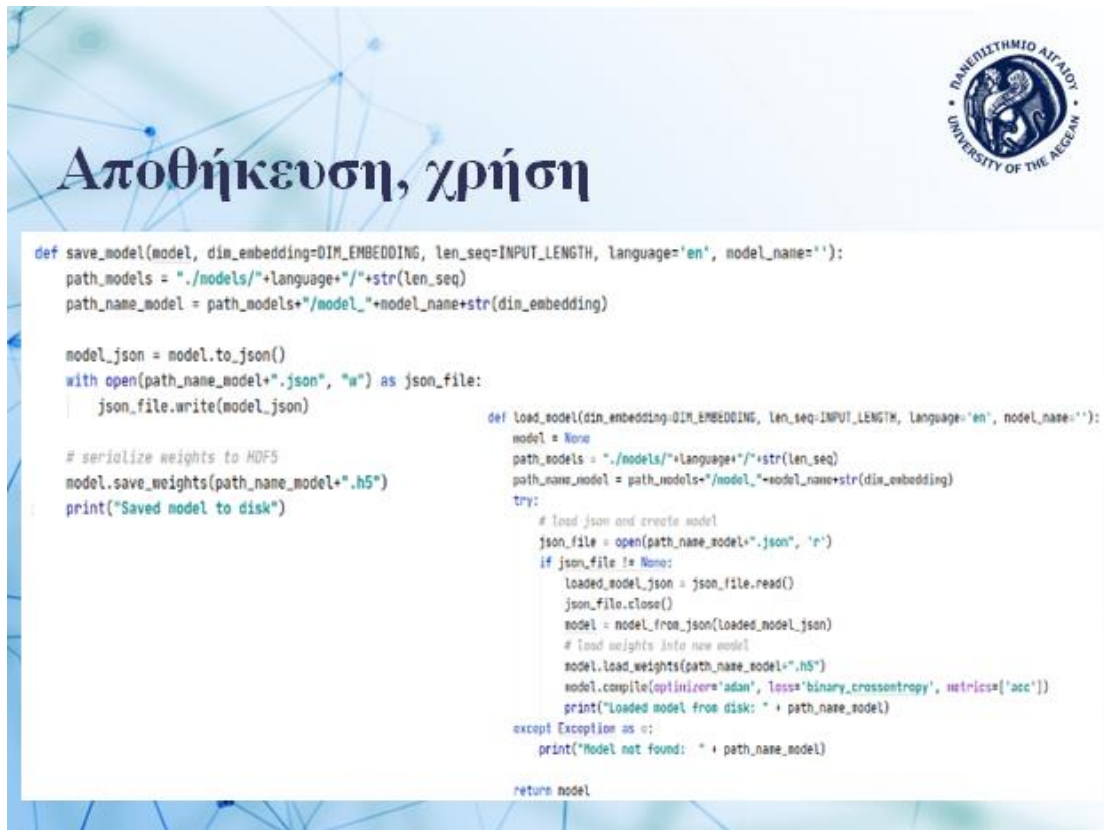
Με τη μέθοδο `evaluate`, το μοντέλο δοκιμάζεται σε νέα δεδομένα που θα πρέπει να είναι ένα διαφορετικό σύνολο, ούτε το train ούτε το validation set. Το αποτέλεσμα είναι μια λίστα που περιέχει το μέσο όρο της ακρίβειας για κάθε μια από τις κατηγορίες, στην προκειμένη περίπτωση την ακρίβεια στις αρνητικές και τις θετικές αξιολογήσεις.

Σε αυτό το σημείο το μοντέλο μπορεί πραγματικά να χρησιμοποιηθεί, οπότε θα του περάσουμε στοιχεία των οποίων δεν γνωρίζουμε την ταξινόμηση. Για να το κάνουμε αυτό χρησιμοποιούμε τη μέθοδο `predict`. Τα δεδομένα που περνούν στην `predict` πρέπει να υποστούν προεπεξεργασία, να μετατραπούν σε αριθμούς και να συμπληρωθούν ακριβώς όπως αυτά που χρησιμοποιήθηκαν κατά την εκπαίδευση.

Το αποτέλεσμα είναι μια `numpy ndarray` που περιέχει τις εξόδους που προέκυψαν με κάθε ένα από τα παραδείγματα. Στην περίπτωσή μας η έξοδος είναι αυτή του τελευταίου νευρώνα, οπότε αποτελείται μόνο από έναν αριθμό για το παράδειγμα. Καθορίζεται ένα κατώφλι για να καθοριστεί αν το συγκεκριμένο στοιχείο έχει βαθμολογηθεί στη μία ή στην άλλη κατηγορία. Το κατώφλι θα μπορούσε λογικά να είναι στο κέντρο, επομένως, με βάση αυτά τα αποτελέσματα, φαίνεται ότι μόνο η τρίτη και η τέταρτη πρόταση που υποβλήθηκαν στο μοντέλο μας ήταν θετικές αξιολογήσεις, ενώ όλες οι άλλες ήταν αρνητικές.

Αν και, στη συγκεκριμένη περίπτωση, δεδομένου ότι πρόκειται για ανάλυση συναισθήματος, η αξιολόγηση μπορεί να μην είναι απαραίτητα σαφής, θα μπορούσαμε να θεωρήσουμε αυτές τις ποσοτικές εκροές, ένα μέτρο του πόσο θετικό είναι το συναισθημα, έτσι ώστε ορισμένες να είναι ιδιαίτερα θετικές και άλλες αρνητικές, άλλες, όπως η έκτη, παρά το γεγονός ότι περιλαμβάνεται

στις αρνητικές αξιολογήσεις, έχοντας λάβει μια εκροή 0,42 θα μπορούσε να θεωρηθεί σχεδόν ουδέτερη, ή σε κάθε περίπτωση όχι ιδιαίτερα αρνητική.



Για να χρησιμοποιήσουμε το μοντέλο σε άλλες περιπτώσεις, για παράδειγμα στο πλαίσιο μιας εφαρμογής, πρέπει να το αποθηκεύσουμε στο τέλος της εκπαίδευσης και να το επαναφορτώσουμε όταν είναι απαραίτητο.

Εδώ βλέπουμε δύο απλές συναρτήσεις χρήσιμες για το σκοπό αυτό.

Μας απασχολεί η μέθοδος αποθήκευσης που πρέπει να γίνει σε δύο αρχεία, ένα που περιέχει τη δομή του μοντέλου αποθηκευμένη ως json και ένα που περιέχει τα βάρη του μοντέλου.

Για να το ξαναφορτώσουμε με τον ίδιο τρόπο πρέπει να διαβάσουμε και τα δύο αρχεία, με το πρώτο παίρνουμε τη δομή και με το δεύτερο τα βάρη. Πριν χρησιμοποιηθεί, το μοντέλο που λαμβάνεται με αυτόν τον τρόπο πρέπει να μεταγλωττιστεί, όπως έγινε κατά τη δημιουργία του μοντέλου.

Εργασία



Προσωπική εργασία

Πραγματοποίηση ταξινόμησης συναισθημάτων

Χρησιμοποιήστε αυτό το σύνολο δεδομένων:

<https://archive.ics.uci.edu/ml/machine-learning-databases/00331/sentiment%20labelled%20sentences.zip>

Προεπεξεργασία
Tokenizer (fit_on_text και text_to_sequences)
pad_sequences
train_test_split
Δημιουργία μοντέλου
Εκπαίδευση
Αποθήκευση μοντέλου

Τώρα θα πρέπει να είστε σε θέση να φτιάξετε έναν ταξινομητή. Αναφέραμε τα βήματα από θεωρητική άποψη και είδαμε πώς να φτιάχνουμε τα διάφορα μέρη, οπότε αρκεί να τα συναρμολογήσετε όλα μαζί.

Το σύνολο δεδομένων μπορεί να μεταφορτωθεί από τον ιστότοπο:

<https://archive.ics.uci.edu/ml/machine-learning-databases/00331/sentiment%20labelled%20sentences.zip>

Μέσα στο zip υπάρχουν τρία αρχεία στα οποία κάθε γραμμή αποτελείται από μια σύντομη πρόταση και την ετικέτα της που χωρίζεται από το κείμενο με ένα tab. Οι ετικέτες είναι 0 για αρνητικό συναίσθημα και 1 για θετικό. Τα τρία αρχεία θα πρέπει κατά προτίμηση να είναι ενοποιημένα, το σημαντικό είναι να έχουμε μια λίστα που να περιέχει τις προτάσεις και μια που να περιέχει τις ετικέτες.

Ορισμένες από τις λειτουργίες προεπεξεργασίας που είδαμε θα εφαρμοστούν στον κατάλογο των κειμένων

Με τον Tokenizer της βιβλιοθήκης Keras δημιουργήστε το λεξικό μέσω του fit_on_texts, μετατρέψτε τα στοιχεία σε ακέραια tokens με το texts_to_sequences και γεμίστε τα μέσω της συνάρτησης του Keras pad_sequences. Με την train_test_split της sklearn, χωρίζουμε το αρχικό σύνολο δεδομένων σε ένα μέρος που θα χρησιμοποιηθεί για την εκπαίδευση και ένα μέρος για την επικύρωση.

Δημιουργήστε το μοντέλο.

Εκκινήστε την εκπαίδευση με, προσαρμόστε και αποθηκεύστε το.

Στο τέλος σας καλώ να δοκιμάσετε μια παραλλαγή του μοντέλου, χρησιμοποιώντας το GloVe αντί για την κενή ενσωμάτωση (είδαμε πώς να το φορτώσετε). Στη συνέχεια, ξεκινήστε την εκπαίδευση με το νέο μοντέλο και δείτε αν υπάρχουν διαφορές. Ποιο έδωσε καλύτερα αποτελέσματα;

Όλος ο κώδικας που είδαμε στις διαφάνειες και η πλήρης λύση της προσωπικής εργασίας που προτάθηκε στην προηγούμενη διαφάνεια βρίσκονται στα αρχεία που εμφανίζονται στη διαφάνεια, αλλά το τελευταίο θα πρέπει να το δείτε μόνο αφού κάνετε την εργασία ανεξάρτητα!

Μια συμβουλή για όταν κάνετε την προ-εκπαιδευμένη έκδοση της ενσωμάτωσης: αφού την ανεβάσετε σε ένα λεξικό με μια συνάρτηση παρόμοια με αυτή που είδατε, θα πρέπει να χρησιμοποιήσετε αυτό το λεξικό λέξεων - διανυσμάτων για να δημιουργήσετε έναν πίνακα διανυσμάτων που πρέπει να περάσει στην παράμετρο `weights` του στρώματος `Embedding`. Αυτός ο πίνακας πρέπει να έχει τον ίδιο αριθμό διανυσμάτων του λεξικού δείκτη - λέξης που παράγεται από τον `Tokenizer` και ο δείκτης τους πρέπει να συμπίπτει με αυτόν του λεξικού που παράγεται από τον `Tokenizer`, αν υπήρχαν κάποιες λέξεις που υπάρχουν στο λεξικό, αλλά όχι στην ενσωμάτωση, ένα διάνυσμα που αποτελείται μόνο από μηδενικά. Η πρώτη παράμετρος της δημιουργίας του στρώματος `Embedding` είναι ακριβώς το μέγεθος αυτού του λεξικού συν ένα, όπως είδαμε ήδη στη δημιουργία του κενού, οι παράμετροι που πρέπει να περάσουμε θα είναι επομένως οι ίδιες με το κενό, συν τα βάρη των παραμέτρων στα οποία θα περάσουμε αυτόν τον πίνακα διανυσμάτων που αντιπροσωπεύουν τις λέξεις που είναι ταξινομημένες όπως το λεξικό που παράγεται από τον `Tokenizer`.

Στα παρακάτω αρχεία μπορείτε να βρείτε τον πλήρη κώδικα για την εκτέλεση των όσων εξηγήθηκαν, συμπεριλαμβανομένων των τμημάτων του κώδικα που παρουσιάζονται στις διαφάνειες:

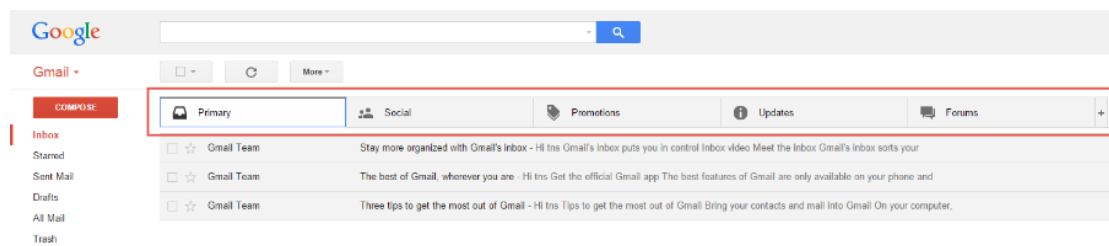
```
Aris_Lesson_4_1_CodeUsedInLesson_Preprocessing.py
Aris_Lesson_4_1_CodeUsedInLesson_VectorsMeasures.py
Aris_Lesson_4_1_CodeusedInLesson_VectorsMeasuresGensim.py
Aris_Lesson_4_1_Exercise1Solution_TextClassificationSentimentAnalysis.py
Aris_Lesson_4_1_Exercise2Solution_TextClassificationCategorical.py
```

Τα δύο τελευταία αρχεία θα πρέπει να ανοίγονται μόνο μετά την εκτέλεση των πρακτικών ασκήσεων, δεδομένου ότι ο κώδικας που περιέχεται σε αυτά είναι η λύση.

Περιπτώσεις χρήσης

ΤΟ ΤΜΗΜΑ ΑΛΛΗΛΟΓΡΑΦΙΑΣ ΤΟΥ GMAIL

Το Gmail χωρίζει τα λαμβανόμενα μηνύματα ηλεκτρονικού ταχυδρομείου σε τέσσερις κύριες κατηγορίες: και ενημερώσεις και αναφέρει ανεξάρτητα ορισμένα μηνύματα ηλεκτρονικού ταχυδρομείου ως σημαντικά με βάση τη συμπεριφορά του χρήστη. Επιπλέον, όπως κάνει πλέον



κάθ
ε
πάρ
οχος
αλλ
ηλογ

ραφίας, μετακινεί αυτόματα τα μηνύματα που αναγνωρίζει ως διαφημιστικά και ανεπιθύμητα ή ακόμη και επικίνδυνα στο φάκελο Spam.

Οι χρησιμοποιούμενες τεχνολογίες έχουν αλλάξει με την πάροδο του χρόνου και έχουν σταδιακά βελτιωθεί τόσο με την αξιοποίηση των εξελίξεων που έχουν σημειωθεί στον τομέα αυτό όσο και με τη λήψη αντιμέτρων για τα προβλήματα που εντοπίστηκαν με την πάροδο του χρόνου.

Για παράδειγμα, οι περισσότεροι από αυτούς που στέλνουν μηνύματα ηλεκτρονικού ταχυδρομείου που καταλήγουν στον φάκελο ανεπιθύμητων μηνυμάτων προσπάθησαν να αποφύγουν τις λέξεις που τους έκαναν αναγνωρίσιμους, οπότε τα φίλτρα έπρεπε να προσαρμοστούν χρησιμοποιώντας διαφορετικούς τρόπους για να τους αναγνωρίσουν.

Αρχικά, το φίλτρο spam βασιζόταν σχεδόν αποκλειστικά σε μαύρες λίστες αποστολέων και σε λέξεις ή εκφράσεις που χαρακτήριζαν αμέσως έναν συγκεκριμένο τύπο ηλεκτρονικού ταχυδρομείου, στη συνέχεια, τα κριτήρια αυτά, τα οποία εξακολουθούν να παίζουν ακόμη ρόλο, έγιναν λιγότερο καθοριστικά.

Επί του παρόντος, τα κριτήρια που χρησιμοποιούνται από τα φίλτρα είναι σε μεγάλο βαθμό προσαρμοσμένα και συνδέονται με τις συνήθειες των μεμονωμένων χρηστών, οπότε ορισμένα μηνύματα που είναι spam για κάποιους μπορεί να μην είναι spam για άλλους. Αυτό που μοιάζει περισσότερο με αυτό που θεωρούνταν προηγουμένως spam από έναν χρήστη, πιθανότατα θα θεωρηθεί spam στο γραμματοκιβώτιό του, ενώ μπορεί να μην είναι σε κάποιο άλλο.

Δεν γνωρίζουμε τις λεπτομέρειες της κατασκευής αυτών των φίλτρων και μπορούμε να υποθέσουμε ότι σήμερα πιθανώς χρησιμοποιούν και κάτι πιο εξελιγμένο, αλλά θα μπορούσαμε να φανταστούμε την υλοποίηση κάτι ανάλογου με τα εργαλεία που γνωρίζουμε μέχρι σήμερα.

Αν δημιουργούσαμε έναν ταξινομητή που διαχωρίζει τα ανεπιθύμητα μηνύματα, τα ενοχλητικά που πρέπει να καταλήξουν στα ανεπιθύμητα μηνύματα από αυτά που επηρεάζουν τον χρήστη και ήμασταν μια εταιρεία όπως η Google, θα είχαμε έναν τεράστιο όγκο μηνυμάτων ηλεκτρονικού ταχυδρομείου που θα είχαν ήδη κατηγοριοποιηθεί ως ανεπιθύμητα ή μη, οπότε η εκπαίδευση του δικτύου θα μπορούσε να πραγματοποιηθεί με τον τρόπο που είδαμε με έναν δυαδικό ταξινομητή. Όλα όσα κάναμε στο παράδειγμα της ανάλυσης συναισθήματος θα μπορούσαν να είναι μια χαρά, εκτός ίσως από το ότι η επιλογή των επιπέδων του δικτύου θα μπορούσε να είναι διαφορετική. Επομένως, θα ξεκινούσαμε από μια προ-εκπαιδευμένη ενσωμάτωση που θα οριζόταν ως

εκπαιδεύσιμη, ώστε να τροποποιήσουμε τις θέσεις των διανυσμάτων έτσι ώστε να προσαρμοστούν ακόμη καλύτερα στο πεδίο του προβλήματός μας.

Μόλις ολοκληρωθεί η εκπαίδευση και αποθηκευτεί το μοντέλο, αυτό θα μπορούσε να χρησιμοποιηθεί για να γίνει αυτή η ταξινόμηση, αλλά δεν θα μπορούσε να χρησιμοποιηθεί με αυτή την αρχική εκπαίδευση για ένα συγκεκριμένο χρονικό διάστημα μόνο.

Η εκπαίδευση αυτή θα ανανεώνεται περιοδικά, προσθέτοντας νέο υλικό που θα έχει κατηγοριοποιηθεί εν τω μεταξύ από τους ίδιους τους χρήστες. Πράγματι, οι χρήστες συνεχίζουν να στέλνουν spam ακόμη και μηνύματα που ο ταξινομητής δεν είχε θεωρήσει ως τέτοια και, αντίθετα, μπορεί να συμβεί να ανακτηθούν από το spam μηνύματα για τα οποία ενδιαφέρεται ο χρήστης και τα οποία είχαν κατηγοριοποιηθεί εσφαλμένα ως spam. Με αυτόν τον τρόπο, το μοντέλο είναι σε θέση να ανταποκρίνεται επαρκώς στα νέα τεχνάσματα που υιοθετούν όσοι στέλνουν ανεπιθύμητα μηνύματα για να μην ανακαλυφθούν και με κάθε νέα εκπαίδευση πλησιάζει περισσότερο στη συμπεριφορά που θα είχαν υιοθετήσει οι χρήστες αν η επιλογή είχε γίνει αυτοπροσώπως πηγαίνοντας να αξιολογήσει κάθε μήνυμα.

Όταν μετακινείτε ρητά μηνύματα ηλεκτρονικού ταχυδρομείου στο φάκελο spam ή τα αναφέρετε ρητά ως τέτοια, συμβάλλετε στο σύνολο εκπαίδευσης του συνολικού φίλτρου και ειδικότερα στην τελειοποίηση του δικού σας.

Πώς νομίζετε ότι η Google διαχειρίζεται αυτές τις διαφορές που μπορεί να προκύψουν μεταξύ των κατηγοριοποιήσεων των ανθρώπων; Πώς καταφέρνει να επιλέγει spam για όλους αλλά και να προσαρμόζεται στις προσωπικές ιδιαιτερότητες;

Το φιλτράρισμα spam είναι ιδιαίτερα χρήσιμο, αλλά και επικίνδυνο, μπορεί να καταλήξουμε με μηνύματα ηλεκτρονικού ταχυδρομείου που ο χρήστης θα ήθελε να λάβει, ίσως απλώς ένα μήνυμα ηλεκτρονικού ταχυδρομείου που περιμένει. Ποια συστήματα πιστεύετε ότι χρησιμοποιεί ήδη σήμερα η Google για να αποφύγει αυτόν τον κίνδυνο; Πώς πιστεύετε ότι θα μπορούσαν να βελτιωθούν περαιτέρω;

Μπορούμε να φανταστούμε ότι όσον αφορά τον διαχωρισμό των μηνυμάτων αλληλογραφίας στις τέσσερις κατηγορίες: κύρια, κοινωνικά, προσφορές και ενημερώσεις, θα μπορούσαν να ισχύουν σκέψεις παρόμοιες με αυτές που αφορούν την επιλογή των ανεπιθύμητων μηνυμάτων.

Ποιες ιδιαιτερότητες πιστεύετε ότι θα είχε αυτή η ταξινόμηση; Πώς θα πρέπει να γίνει λεπτομερώς;

Αναφορές

Ενσωμάτωση

- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. ICLR Workshop, 2013
- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In EMNLP
- <https://projector.tensorflow.org/Embedding>

Ταξινόμηση κειμένου

- <https://freecontent.manning.com/deep-learning-for-text/>
- Deep Learning with Python-Chollet, F., Manning Pubns Co, 2017. ISBN-10: 9781617294433
- Natural Language Processing with Python-Bird, S.; Klein E.; Loper E., O'Reilly Media, 2009. ISBN-10: 0596516495
- https://www.tensorflow.org/hub/tutorials/tf2_text_classification
- <https://www.kdnuggets.com/2019/03/building-nlp-classifiers-cheaply-transfer-learning-weak-supervision.html>Text classification
- <https://www.katacoda.com/basiefusinska/courses/nlp-with-python>
- <https://www.kaggle.com/learn/natural-language-processing>

Μάθημα 4.2: Νευρωνικά δίκτυα για NLP και βιβλιοθήκες

Θέματα προς συζήτηση

- Βιβλιοθήκες Scikit-learn, Keras και Tensorflow
- Συνελικτικό νευρωνικό δίκτυο
- Επαναλαμβανόμενο νευρωνικό δίκτυο
- Παραλλαγές RNN (LSTM, GRU)
- Μηχανισμός προσοχής
- Ενισχυτική μάθηση
- Μάθηση χωρίς επίβλεψη

Σημειώσεις διάλεξης

Scikit-learn, Keras and TensorFlow



Το **Scikit-learn** είναι μια βιβλιοθήκη μηχανικής μάθησης ανοικτού κώδικα που βασίζεται στις NumPy, SciPy και matplotlib. Διαθέτει πολλά βοηθητικά προγράμματα για γενική προ- και μετα-επεξεργασία δεδομένων. Είναι μια βιβλιοθήκη στην Python που χρησιμοποιείται για την κατασκευή παραδοσιακών μοντέλων. Το Scikit-learn υποστηρίζει τις ακόλουθες κύριες εργασίες:

- μετασχηματιστές δεδομένων προεπεξεργασίας
- διάφορα εργαλεία για την προσαρμογή μοντέλων
- αγωγοί (pipelines)
- μάθηση με επίβλεψη και χωρίς επίβλεψη
- επιλογή και αξιολόγηση μοντέλων
- και πολλά άλλα βοηθητικά προγράμματα

Παρέχει επίσης δεκάδες ενσωματωμένους αλγόριθμους και μοντέλα μηχανικής μάθησης. Εν συντομία, το Scikit-learn είναι μια γενική βιβλιοθήκη μηχανικής μάθησης που βασίζεται στη NumPy.

Η **Tensorflow** είναι η πιο διάσημη βιβλιοθήκη παραγωγής για μοντέλα βαθιάς μάθησης. Προσφέρει αυτόματη διαφοροποίηση για την ομαλή εκτέλεση backpropagation, επιτρέποντάς σας να δημιουργήσετε οποιοδήποτε μοντέλο μηχανικής μάθησης.

Είναι μια ολοκληρωμένη πλατφόρμα ανοικτού κώδικα για μηχανική μάθηση. Διαθέτει ένα ολοκληρωμένο, ευέλικτο οικοσύστημα εργαλείων, βιβλιοθηκών και πόρων της κοινότητας που επιτρέπει στους ερευνητές να προωθήσουν την τελευταία λέξη της τεχνολογίας στην ML και στους προγραμματιστές να κατασκευάσουν και να αναπτύξουν εύκολα εφαρμογές που υποστηρίζονται από την ML.

- Εύκολη κατασκευή μοντέλων
- Στιβαρή παραγωγή ML οπουδήποτε
- Ισχυρός πειραματισμός για έρευνα

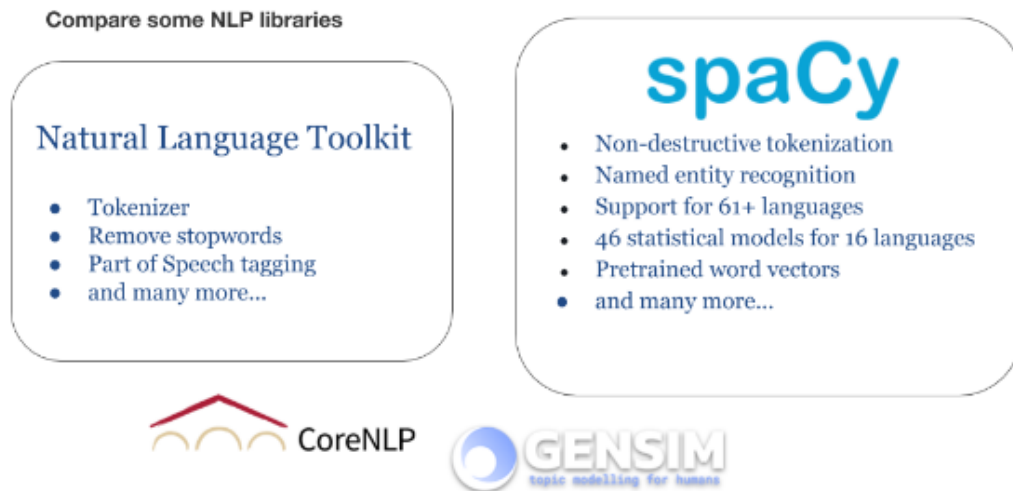
Βασικά, το Tensorflow παίρνει το όνομά του από τη "ροή των τανυστών", όπου ένας τανυστής μπορεί να αναπαρασταθεί ως ένας πολυδιάστατος πίνακας αριθμών. Η "ροή" αναφέρεται στις πράξεις μεταξύ των τανυστών.

Η δύναμη του Tensorflow όσον αφορά την ικανότητα απόδοσης είναι ότι μπορεί να λειτουργήσει σε CPU (Central Processing Unit), τη μονάδα επεξεργασίας γενικής χρήσης, αλλά και σε GPU (Graphic Processing Unit) και TPU (Tensor Processing Unit), ειδική μονάδα επεξεργασίας προσαρμοσμένη για τη λειτουργία τανυστών και προβλήματα βαθιάς μάθησης, εξαιρετικά ταχύτερα από τις άλλες.

Το **Keras** είναι ένα API υψηλού επιπέδου και σήμερα αποτελεί μέρος του Tensorflow. Είναι φιλικό προς το χρήστη και βοηθά στη γρήγορη κατασκευή και δοκιμή ενός νευρωνικού δικτύου με ελάχιστες γραμμές κώδικα. Όπως η κατασκευή απλών ή πολύπλοκων νευρωνικών δικτύων μέσα σε λίγα λεπτά. Είναι σπονδυλωτό, αφού τα πάντα στο Keras μπορούν να αναπαρασταθούν ως ενότητες.

Χρησιμοποιείται για γρήγορη κατασκευή πρωτοτύπων, έρευνα τελευταίας τεχνολογίας και παραγωγή, με τρία βασικά πλεονεκτήματα

- **Φιλικό προς το χρήστη**
Το Keras διαθέτει μια απλή, συνεπή διεπαφή βελτιστοποιημένη για κοινές περιπτώσεις χρήσης. Παρέχει σαφή και αξιοποιήσιμη ανατροφοδότηση για τα σφάλματα του χρήστη.
- **Αρθρωτό και σύνθετο**
Τα μοντέλα της Keras κατασκευάζονται με τη σύνδεση διαμορφώσιμων δομικών στοιχείων μεταξύ τους, με λίγους περιορισμούς.
- **Εύκολη επέκταση**
Επιτρέπει τη γραφή προσαρμοσμένων δομικών στοιχείων για να εκφράσουμε νέες ιδέες για την έρευνα. Δημιουργεί νέα επίπεδα, μετρικές, συναρτήσεις απώλειας και αναπτύξτε μοντέλα τελευταίας τεχνολογίας



Το Natural Language Toolkit (NLTK) είναι μια πλατφόρμα για τη δημιουργία προγραμμάτων Python που εργάζονται με δεδομένα ανθρώπινης γλώσσας. Παρέχει εύχρηστες διεπαφές σε πάνω από 50 σώματα κειμένων και λεξιλογικούς πόρους, όπως το WordNet, το Web Text Corpus, το NPS Chat, το SemCor, το FrameNet, μαζί με μια σουίτα βιβλιοθηκών επεξεργασίας κειμένου για ταξινόμηση, tokenization, stemming, tagging, parsing και σημασιολογική συλλογιστική και wrappers για βιβλιοθήκες NLP βιομηχανικής ισχύος.

Ας δούμε τις κύριες λειτουργίες του NLTK:

- **Τι είναι ο tokenizer;**

Ένας tokenizer είναι μια συνάρτηση NLP που μπορεί να σπάσει ένα συγκεκριμένο στοιχείο σε επιμέρους στοιχεία (αν είναι δυνατόν) σύμφωνα με ένα σύνολο δεδομένων κανόνων. Για παράδειγμα, οι tokenizers προτάσεων χρησιμοποιούνται για να σπάσουν μια συλλογή κειμένου σε προτάσεις, ενώ οι tokenizers λέξεων χρησιμοποιούνται για να σπάσουν συλλογές κειμένου σε λέξεις.

- **Τι είναι τα stopwords;**

Τα stopwords είναι λέξεις που δεν έχουν μεγάλη σημασία για την ανάλυση του κειμένου. Οι λέξεις αυτές χρησιμοποιούνται μόνο για να καλύψουν το κενό μεταξύ των λέξεων.

- **Τι είναι η επισήμανση μέρους του λόγου (POS);**

Πρόκειται για την ταξινόμηση των λέξεων σε μέρη του λόγου μιας γλώσσας που έχει οριστεί. Για παράδειγμα, μια συγκεκριμένη λέξη μπορεί να είναι ουσιαστικό ή ρήμα ή κάτι άλλο. Αυτό το χαρακτηριστικό είναι πολύ ισχυρό και βοηθάει μαζικά στην περαιτέρω ανάλυση.

Το SPACY αναγνωρίζεται συχνά ως μια επαγγελματικής ποιότητας βιβλιοθήκη Python για προηγμένη επεξεργασία φυσικής γλώσσας, το spaCy υπερέρχει στην εργασία με απίστευτα μεγάλης κλίμακας εργασίες εξόρυξης πληροφοριών.

Πρόσφατα κυκλοφόρησε η έκδοση 3.0, σύμφωνα με τον ιστότοπο, η οποία διαθέτει όλες τις νέες γραμμές επικοινωνίες σωληνώσεις που βασίζονται σε μετασχηματιστές, οι οποίες φέρνουν την ακρίβεια του spaCy στην τρέχουσα τεχνολογία. Μπορείτε να χρησιμοποιήσετε οποιονδήποτε προ-εκπαιδευμένο μετασχηματιστή για να εκπαιδεύσετε τις δικές σας σωληνώσεις και ακόμη και να μοιραστείτε έναν μετασχηματιστή μεταξύ πολλαπλών στοιχείων με μάθηση πολλαπλών εργασιών. Η εκπαίδευση είναι πλέον πλήρως παραμετροποιήσιμη και επεκτάσιμη και μπορείτε να ορίσετε τα δικά σας προσαρμοσμένα μοντέλα χρησιμοποιώντας PyTorch, TensorFlow και άλλα πλαίσια. Το νέο σύστημα έργων spaCy σας επιτρέπει να περιγράφετε ολόκληρες ροές εργασίας από άκρο σε άκρο σε ένα μόνο αρχείο, παρέχοντάς σας μια εύκολη διαδρομή από το πρωτότυπο στην παραγωγή και διευκολύνοντας την κλωνοποίηση και προσαρμογή έργων βέλτιστης πρακτικής για τις δικές σας περιπτώσεις χρήσης.

Το Gensim είναι μια βιβλιοθήκη Python σχεδιασμένη ειδικά για " μοντελοποίηση θεμάτων, ευρετηρίαση εγγράφων και ανάκτηση ομοιοτήτων με μεγάλα σώματα κειμένων. " Όλοι οι αλγόριθμοι στο Gensim είναι ανεξάρτητοι από τη μνήμη, σε σχέση με το μέγεθος του σώματος, και ως εκ τούτου, μπορεί να επεξεργαστεί είσοδο μεγαλύτερη από τη μνήμη RAM.

Το Stanford CoreNLP περιλαμβάνει μια σειρά εργαλείων τεχνολογίας ανθρώπινης γλώσσας. Στόχος του είναι να καταστήσει την εφαρμογή της γλωσσολογικής εργαλείων γλωσσικής ανάλυσης σε ένα κομμάτι κειμένου εύκολη και αποτελεσματική. Με το CoreNLP, μπορείτε να εξάγετε όλα τα είδη ιδιοτήτων κειμένου (όπως ονομαστικές οντότητες αναγνώριση ονομάτων, επισήμανση μέρους του λόγου κ.λπ.) με λίγες μόνο γραμμές κώδικα.

Υπάρχουν πολλές περισσότερες βιβλιοθήκες NLP και μερικές φορές με επικαλυπτόμενες λειτουργίες, ούτως ή άλλως κάθε μία έχει χαρακτηριστικά που υποδηλώνουν μια συγκεκριμένη εφαρμογή: εύκολη στη χρήση έναντι επαγγελματικών λειτουργιών, μνήμη έναντι ταχύτητας. Η χρήση μιας συγκεκριμένης βιβλιοθήκης πρέπει να επικυρωθεί ανάλογα με τις λειτουργίες, τις εφαρμογές και το περιβάλλον (υλικό και λογισμικό) που χρησιμοποιείται για την εκτέλεση.

Ας δούμε τώρα βασικές ασκήσεις χρησιμοποιώντας το NLTK και το spacy

Άσκηση 1

Χρησιμοποιώντας το NLTK και δίνοντας μια πρόταση όπως " At eight o'clock on Thursday morning Arthur didn't feel very good. " . :

- Εκτυπώστε τα tokens
- Αναγνωρίστε το POS
- Προσδιορίστε τις οντότητες ονόματος

```
import nltk
sentence = """"Στις At eight o'clock on Thursday morning Arthur didn't feel very good.""""
tokens = nltk.word_tokenize(sentence)
tokens
```

Το αποτέλεσμα είναι:

```
['At', 'eight', 'o'clock', 'on', 'Thursday', 'morning', 'Arthur', 'did', 'n't', 'feel', 'very', 'good', '.']
```

Από το να εμφανίσετε POS:

```
tagged = nltk.pos_tag(tokens)
Tagged[0:6]
```

```
[('At', 'IN'), ('eight', 'CD'), ('o'clock', 'JJ'), ('on', 'IN'), ('Thursday', 'NNP'), ('morning', 'NN')]
```

Και για τον προσδιορισμό ονομαστικών οντοτήτων:

```
entities = nltk.chunk.ne_chunk(tagged)
entities
Tree('S', [('At', 'IN'), ('eight', 'CD'), ('o'clock', 'JJ'),
('on', 'IN'), ('Thursday', 'NNP'), ('morning', 'NN'),
Tree('PERSON', [('Arthur', 'NNP'])],
('did', 'VBD'), ('n't', 'RB'), ('feel', 'VB'),
('πολύ', 'RB'), ('καλά', 'JJ'), ('.', '.'))])
```

Άσκηση 2

Χρησιμοποιώντας spaCy και δίνοντας ένα συγκεκριμένο κείμενο:

- Εκτυπώστε φράσεις ουσιαστικών
- Εκτύπωση ρημάτων
- Εύρεση ονομαστικών οντοτήτων, φράσεων και εννοιών

```
import spacy
# Load English tokenizer, tagger, parser, NER and word vectors
nlp = spacy.load("en_core_web_sm")
# Process whole documents
text = ("When Sebastian Thrun started working on self-driving cars at "
"Google in 2007, few people outside of the company took him "
"seriously. "I can tell you very senior CEOs of major American "
"car companies would shake my hand and turn away because I wasn't "
"worth talking to," said Thrun, in an interview with Recode earlier "
"this week.")
• doc = nlp(text)

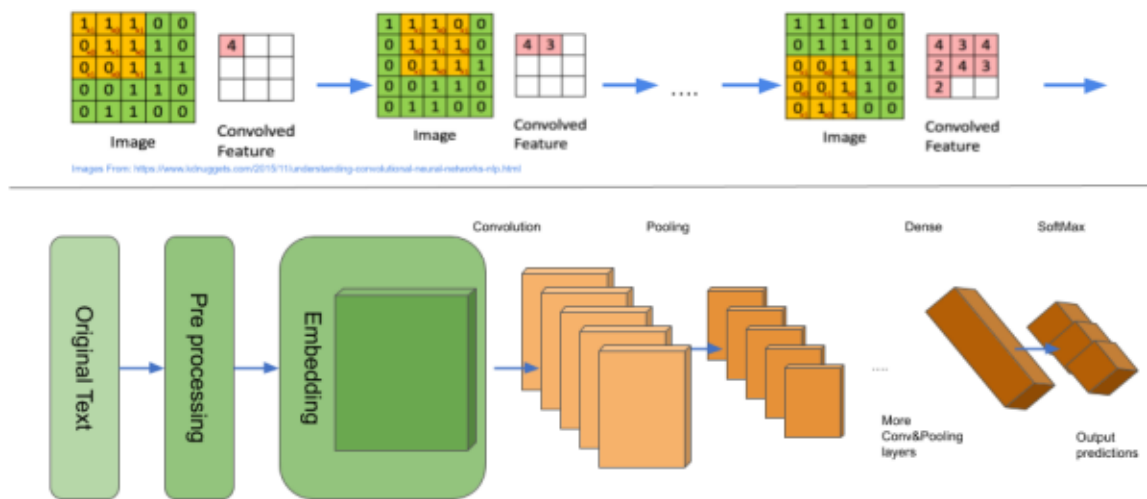
• # Analyze syntax
• print("Noun phrases:", [chunk.text for chunk in doc.noun_chunks])
• print("Verbs:", [token.lemma_ for token in doc if token.pos_ == "VERB"])

• # Find named entities, phrases and concepts
• for entity in doc.ents:
• print(entity.text, entity.label_)
```

Το αποτέλεσμα είναι:

- Noun phrases: ['Sebastian Thrun', 'self-driving cars', 'Google', 'few people', 'the company', 'him', 'I', 'you', 'very senior CEOs', 'major American car companies', 'my hand', 'I', 'Thrun', 'an interview', 'Recode']
- Verbs: ['start', 'work', 'drive', 'take', 'can', 'tell', 'would', 'shake', 'turn', 'talk', 'say']
- Sebastian Thrun PERSON
- Google ORG
- 2007 DATE
- American NORP
- Thrun ORG
- earlier this week DATE

Συνελικτικό νευρωνικό δίκτυο για NLP



Πρώτα απ' όλα θα επαναλάβουμε τι έχουμε μάθει για τα Συνελικτικά Νευρωνικά Δίκτυα, στη συνέχεια θα εξετάσουμε μια πιθανή εφαρμογή για το NLP.

Όταν ακούμε για τα Νευρωνικά Δίκτυα Συνελικτικού Σχεδιασμού (CNN), συνήθως σκεφτόμαστε προβλήματα επιβλεπόμενης μάθησης στα Υπολογιστικά Οπτικά Συστήματα. Τα CNNs ήταν υπεύθυνα για σημαντικές ανακαλύψεις στην ταξινόμηση εικόνων και αποτελούν τον πυρήνα των περισσότερων υπολογιστικών οπτικών συστημάτων σήμερα.

Ας συγκρίνουμε τα πλήρως συνδεδεμένα νευρωνικά δίκτυα και τα CNNs:

- **Ας δούμε πρώτα τις ομοιότητες.** Τόσο τα συνελικτικά νευρωνικά δίκτυα όσο και τα πλήρως συνδεδεμένα νευρωνικά δίκτυα έχουν μαθησιακά βάρη και προκαταλήψεις. Και στα δύο δίκτυα οι νευρώνες λαμβάνουν κάποια είσοδο, εκτελούν ένα βαθμωτό γινόμενο (dot product) και το ακολουθούν με μια μη γραμμική συνάρτηση όπως η ReLU(Rectified Linear Unit)
- **τώρα οι διαφορές.** Μιλώντας για εικόνες, - ας πούμε ότι έχουμε μια εικόνα εισόδου μεγέθους 64x64x3 - ένα πλήρως συνδεδεμένο επίπεδο χρειάζεται 12288 βάρη στο πρώτο κρυφό επίπεδο! Ο αριθμός των βαρών θα είναι ακόμη μεγαλύτερος για εικόνες με

μέγεθος $225 \times 225 \times 3 = 151875$. Τα δίκτυα με μεγάλο αριθμό παραμέτρων αντιμετωπίζουν διάφορα προβλήματα, π.χ. πιο αργή εκπαίδευση χρόνος εκπαίδευσης, πιθανότητες υπερπροσαρμογής κ.λπ. Εφαρμόζοντας μια λειτουργία που ονομάζεται συνέλιξη στην ίδια εικόνα εισόδου, μπορεί να μειωθεί σε $1 \times 1 \times 10$ με μία τάξη λιγότερη στον αριθμό των βαρών και αυξημένη συνολική απόδοση.

Στην κορυφή της διαφάνειας παρουσιάζεται η λειτουργία της συνέλιξης, ο ευκολότερος τρόπος για να κατανοήσουμε μια συνέλιξη είναι να τη σκεφτούμε ως μια συνάρτηση κυλιόμενου παραθύρου (sliding window function) που εφαρμόζεται σε έναν πίνακα:

- Ας υποθέσουμε ότι έχουμε μια εικόνα εισόδου 5×5 δυαδικών εικονοστοιχείων (πράσινο).
- από το να ορίσουμε ένα φίλτρο (ή πυρήνα) δηλ. 3×3 (πορτοκαλί) με κάποια βάρη x_i
- ας φανταστούμε ότι ολισθαίνουμε το φίλτρο πάνω στον πίνακα από την πάνω αριστερή προς την κάτω δεξιά γωνία. Σε κάθε βήμα το φίλτρο επιλέγει ένα υποπίνακα
- το πρώτο στοιχείο των συγκολλημένων χαρακτηριστικών υπολογίζεται χάρη στο βαθμωτό γινόμενο μεταξύ του υποπίνακα και του φίλτρου.
- ολισθαίνοντας το φίλτρο πάνω στον πίνακα, υπολογίζονται όλα τα στοιχεία του πίνακα των συνεπτυγμένων χαρακτηριστικών.

Αν είστε οπαδός των αναλογιών εγκεφάλου/νευρώνα, κάθε εγγραφή στην έξοδο μπορεί επίσης να ερμηνευτεί ως έξοδος ενός νευρώνα που εξετάζει μόνο μια μικρή περιοχή στην είσοδο και μοιράζεται τις παραμέτρους με όλους τους νευρώνες που βρίσκονται χωρικά αριστερά και δεξιά (αφού αυτοί οι αριθμοί προκύπτουν όλοι από την εφαρμογή του ίδιου φίλτρου).

Δεδομένου ότι χρησιμοποιούνται συνήθως πολλαπλά φίλτρα, η έξοδος ενός επιπέδου συνέλιξης είναι συχνά ένας όγκος. Τρεις είναι οι υπερπαραμέτροι που ελέγχουν το μέγεθος του όγκου εξόδου: **το βάθος, το βήμα και η μηδενική συμπλήρωση**:

1. το βάθος αντιστοιχεί στον αριθμό των φίλτρων που θα θέλαμε να χρησιμοποιήσουμε, καθένα από τα οποία μαθαίνει να αναζητά κάτι διαφορετικό στην είσοδο. Είναι σύνηθες να αναφερόμαστε σε ένα σύνολο νευρώνων που εξετάζουν όλοι την ίδια περιοχή της εισόδου ως στήλη βάθους (ορισμένοι προτιμούν επίσης τον όρο ίνα).
2. Δεύτερον, πρέπει να καθορίσουμε το βήμα με το οποίο ολισθαίνουμε το φίλτρο. Όταν το βήμα είναι 1, τότε μετακινούμε τα φίλτρα κατά ένα pixel κάθε φορά. Όταν το βήμα είναι 2 (ή ασυνήθιστα 3 ή περισσότερα, αν και αυτό είναι σπάνιο στην πράξη) τότε τα φίλτρα πηδούν 2 pixel κάθε φορά καθώς τα ολισθαίνουμε. Αυτό θα παράγει μικρότερους όγκους εξόδου χωρικά.
3. Μερικές φορές θα είναι βολικό να γεμίσουμε τον όγκο εισόδου με μηδενικά γύρω από τα όρια. Το μέγεθος αυτού του **μηδενικού συμπληρώματος** είναι μια υπερπαραμέτρος. Το ωραίο χαρακτηριστικό του zero padding είναι ότι θα μας επιτρέψει να ελέγξουμε το χωρικό μέγεθος των όγκων εξόδου.

Μπορούμε να υπολογίσουμε το χωρικό μέγεθος του όγκου εξόδου ως συνάρτηση του μεγέθους του όγκου εισόδου (W_i, W_j), του μεγέθους του δεκτικού πεδίου των νευρώνων του επιπέδου $Conv$ (F_i, F_j), του βήματος με το οποίο εφαρμόζονται (S_i, S_j) και του ποσού της μηδενικής συμπλήρωσης που χρησιμοποιείται (P_i, P_j) στο σύνολο. Μπορείτε να πείσετε τον εαυτό σας ότι ο σωστός τύπος για τον υπολογισμό του πόσοι νευρώνες "χωράνε" δίνεται από $(W_i * F_i + 2 * P_i) / S_i + 1$ (το ίδιο για τον δείκτη j). Για παράδειγμα, για μια είσοδο 7×7 και ένα φίλτρο 3×3 με βήμα 1 και pad 0 θα είχαμε ένα 5×5 έξοδο. Με βήμα 2 θα είχαμε έξοδο 3×3 .

Συνήθως μετά από μια συνέλιξη χρησιμοποιείται ένα επίπεδο συγκέντρωσης (max pooling operation) για να συλλάβει τις "σχετικές" πληροφορίες από τη συνέλιξη. και για να μειωθεί προοδευτικά το χωρικό μέγεθος της αναπαράστασης, ώστε να μειωθεί ο αριθμός των παραμέτρων και των υπολογισμών στο δίκτυο

Πιο πρόσφατα αρχίσαμε επίσης να εφαρμόζουμε τα CNNs σε προβλήματα επεξεργασίας φυσικής γλώσσας και πήραμε μερικά ενδιαφέροντα αποτελέσματα. Οι διαισθήσεις πίσω από τα CNN είναι κάπως ευκολότερο να κατανοηθούν για την περίπτωση χρήσης της Όρασης Υπολογιστών, γι' αυτό και ξεκινάμε από εικόνες

Η προεπεξεργασία κειμένου αποτελεί βασικό βήμα για όλα σχεδόν τα προβλήματα NLP [5] και εξαρτάται κυρίως από το πρόβλημα NLP (ταξινόμηση του κειμένου, ανάλυση συναισθήματος ή πιο σύνθετο καθήκον όπως chatbot ή περίληψη κειμένου) και τη γλώσσα (αγγλικό ή γαλλικό κείμενο χρειάζεται διαφορετική προεπεξεργασία από τα ιαπωνικά ή τα κινεζικά).

Οι στόχοι της προεπεξεργασίας είναι:

- να μειωθούν τα δεδομένα που πρέπει να επεξεργαστεί το μοντέλο
- Να βελτιωθεί η αποδοτικότητα και η αποτελεσματικότητα των πληροφοριών

Οι κύριες τεχνικές προεπεξεργασίας αποτελούνται από (ο κατάλογος δεν είναι εξαντλητικός):

1. Αφαιρέστε όλους τους άσχετους χαρακτήρες, όπως οποιουσδήποτε μη αλφαριθμητικούς χαρακτήρες (όπως το @).
2. Μετατρέψτε όλους τους χαρακτήρες σε πεζούς, ώστε να αντιμετωπίζονται το ίδιο λέξεις όπως "hello", "Hello" και "HELLO".
3. Εξετάστε το ενδεχόμενο συνδυασμού ανορθόγραφων ή εναλλακτικά γραμμένων λέξεων σε μια ενιαία αναπαράσταση (π.χ. "cool"/"kewl"/"coool")
4. Εξετάστε το ενδεχόμενο λημματισμού (αναγωγή λέξεων όπως "am", "are" και "is" σε μια κοινή μορφή όπως "be")
5. Αφαιρέστε τις λέξεις στάσης (όπως "και", "είναι", "αυτό" κ.λπ.), που δεν βελτιώνουν την πληροφορία στο κείμενο, ειδικά για το πρόβλημα της ταξινόμησης.
6. Stemming είναι η διαδικασία συγχώνευσης των παραλλακτικών μορφών μιας λέξης σε μια κοινή αναπαράσταση, το στέλεχος (οι λέξεις: "παρουσίαση", "παρουσιάζεται", "παρουσιάζει" θα μπορούσαν όλες να αναχθούν στην λέξη "παρουσιάζω" που σε αυτή την περίπτωση είναι το στέλεχος).
7. Tokenize το κείμενό σας διαχωρίζοντάς το σε επιμέρους λέξεις

Το ποιο υποσύνολο των βημάτων προεπεξεργασίας εφαρμόζεται σε ένα μεμονωμένο πρόβλημα NLP είναι ένα είδος αλχημείας (δεν θα είναι το μοναδικό) που μόνο η εμπειρία θα σας διδάξει. Η δοκιμή και η αξιολόγηση είναι μια καλή στρατηγική, αλλά θυμηθείτε: τώρα μιλάμε για CNN και προβλήματα ταξινόμησης με επίβλεψη.

Εφόσον εφαρμόζουμε την τεχνική CNN, τώρα πρέπει να αποφασίσουμε το μήκος (τον αριθμό των λέξεων) της τυπικής φράσης. Θα είναι μία από τις υπερπαραμέτρους του μοντέλου. Ας υποθέσουμε ότι επιλέγουμε η ως τον αριθμό των λέξεων που αποτελούν μια πρόταση και ότι επιλέγουμε και την ενσωμάτωση m διαστάσεων για κάθε λέξη. Μετά τη διαδικασία ενσωμάτωσης

θα έχουμε έναν πίνακα $n*m$ στοιχείων που θα είναι η είσοδος του συνελκτικού στρώματος. Είναι σαν να ξεκινάμε από ένα κείμενο, για να συνθέσουμε και την "εικόνα" που θα είναι η είσοδος.

Τι γίνεται αν η πρόταση δεν χωράει ακριβώς η ως αριθμό λέξεων; Αν ο αριθμός των λέξεων είναι μεγαλύτερος από n , πρέπει να κόψουμε (περιορίσουμε) τη φράση. Αν είναι μικρότερος, τότε θα προσθέσουμε μηδενικά διανύσματα μέχρι η διάσταση ενσωμάτωσης να είναι $n*m$.

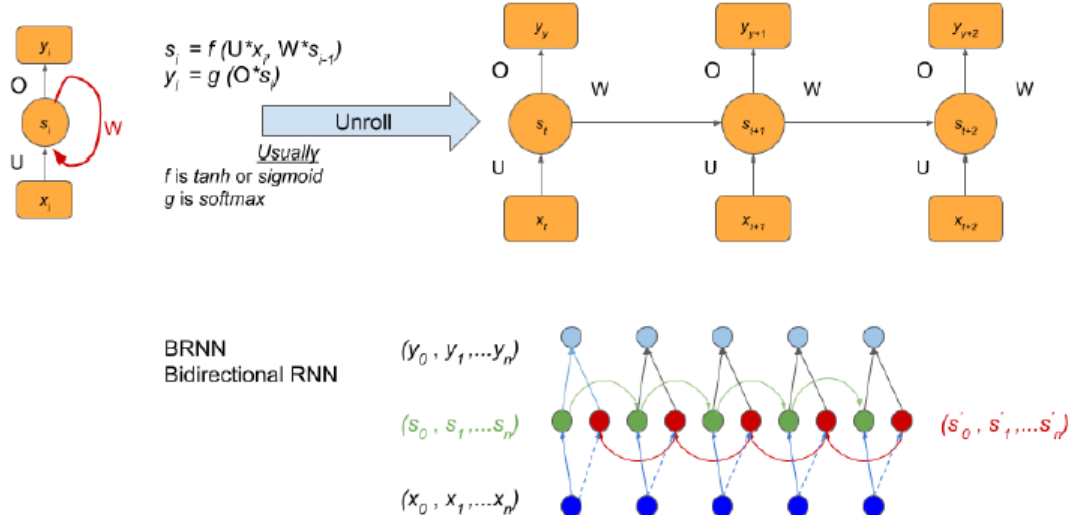
Το αποτέλεσμα της συνέλιξης θα είναι (συνήθως) μια σειρά από στρώματα (ίσα με τον αριθμό των φίλτρων) και κάθε στρώμα είναι μικρότερο από την αρχική εικόνα. Στη συνέχεια, σε κάθε στρώμα συνέλιξης εφαρμόζεται μια πράξη συγκέντρωσης για τη σταδιακή μείωση του χωρικού μεγέθους της αναπαράστασης ώστε να μειωθεί το ποσό των παραμέτρων και των υπολογισμών στο δίκτυο (τυπικά maxPooling με παράθυρο $2*2$ και βήμα ίσο με 2).

Μετά το πρώτο στρώμα συγκέντρωσης, μπορεί να εφαρμοστεί άλλο ένα στρώμα συνελκτικής ανάλυσης για τη σύλληψη περισσότερων πληροφοριών στον λανθάνων χώρο και αυτό συνήθως γίνεται μειώνοντας το μέγεθος του ενιαίου στρώματος και αυξάνοντας τον αριθμό των στρωμάτων. Συνιστάται ιδιαίτερα να ανατρέξετε στο [3] για περισσότερες λεπτομέρειες.

Το μοντέλο που παρουσιάζεται στη διαφάνεια καταλήγει σε ένα πλήρως συνδεδεμένο επίπεδο και ένα επίπεδο softmax που χρησιμοποιείται για προβλήματα ταξινόμησης πολλαπλών κατηγοριών.

Τελευταίο αλλά όχι λιγότερο σημαντικό: το στρώμα ενσωμάτωσης. Μιλήσαμε ήδη γι' αυτό σε προηγούμενο μάθημα, αλλά στα μοντέλα CNN είναι χρήσιμο να επανεκπαιδεύουμε και το στρώμα ενσωμάτωσης; Αλχημεία για άλλη μια φορά...

Recurrent Neural Networks



Τώρα θα παρουσιάσουμε έναν νέο τύπο νευρωνικών δικτύων που ονομάζεται Recurrent Neural Networks και θα εξετάσουμε τις εφαρμογές NLP.

Όταν διαβάζουμε (ή γράφουμε) ένα κείμενο, το κάνουμε μία λέξη μετά την άλλη, προσπαθώντας να κρατήσουμε στο μυαλό μας το νόημα των προηγούμενων λέξεων (τις πληροφορίες) και να χτίσουμε το νόημα της πρότασης. Οι πληροφορίες είναι διαδοχικές, και η ιδέα πίσω από τα RNN είναι να αποτυπώνουν την ακολουθία, και αυτός είναι ο λόγος για τον οποίο έχουν ένα είδος μνήμης.

Προφανώς, όλα τα θέματα προεπεξεργασίας και ενσωμάτωσης που καλύφθηκαν για τα CNN ισχύουν και για τους άλλους τύπους δικτύων όταν αντιμετωπίζουμε προβλήματα NLP.

Στην επάνω αριστερή πλευρά της διαφάνειας, μπορούμε να βρούμε έναν μεμονωμένο νευρώνα των RNNs. Η κύρια διαφορά από τα δίκτυα που έχουμε δει στο παρελθόν είναι ο κόκκινος βρόχος που από την έξοδο του νευρώνα πηγαίνει στην είσοδο, έτσι ώστε η επόμενη εσωτερική κατάσταση (και η νέα έξοδος) να υπολογίζεται λαμβάνοντας υπόψη την πραγματική είσοδο του νευρώνα και την πραγματική κατάσταση. Παρακαλώ κοιτάξτε τον τύπο στα δεξιά.

Πάνω δεξιά βλέπουμε το ξετυλιγμένο δίκτυο με την πάροδο του χρόνου, τα RNN ονομάζονται "επαναλαμβανόμενα" επειδή εκτελούν την ίδια εργασία για κάθε στοιχείο μιας ακολουθίας, με την έξοδο να εξαρτάται από τους προηγούμενους υπολογισμούς. Σε αντίθεση με ένα παραδοσιακό βαθύ νευρωνικό δίκτυο, το οποίο χρησιμοποιεί διαφορετικές παραμέτρους σε κάθε επίπεδο, ένα RNN μοιράζεται τις ίδιες παραμέτρους (U, W, O) σε όλα τα βήματα. Αυτό αντικατοπτρίζει το γεγονός ότι εκτελούμε την ίδια εργασία σε κάθε βήμα, απλώς με διαφορετικές εισόδους. Αυτό μειώνει σημαντικά τον συνολικό αριθμό των παραμέτρων που πρέπει να μάθουμε.

Εμβαθύνοντας λίγο στις λεπτομέρειες, ας δούμε τους τύπους που διέπουν τον εσωτερικό υπολογισμό που συμβαίνει σε έναν μόνο νευρώνα ενός RNN:

- x_t είναι η είσοδος στο χρονικό βήμα t . Για παράδειγμα, x_1 είναι το διάνυσμα ενσωμάτωσης που αντιστοιχεί στη δεύτερη λέξη μιας πρότασης.
- s_t είναι η κρυφή κατάσταση στο χρονικό βήμα t . Είναι η "μνήμη" του δικτύου. s_t υπολογίζεται με βάση την προηγούμενη κρυφή κατάσταση και την είσοδο στο τρέχον βήμα: $s_t = f(U * x_t, W * s_{t-1})$. Η συνάρτηση f είναι συνήθως μια μη γραμμικότητα όπως η $\tanh$ ή η ReLU . s_{-1} , η οποία απαιτείται για τον υπολογισμό της πρώτης κρυφής κατάστασης, συνήθως αρχικοποιείται σε όλα τα μηδενικά.
- y_t είναι η έξοδος στο βήμα t . Για παράδειγμα, αν θέλαμε να προβλέψουμε την επόμενη λέξη σε μια πρόταση θα ήταν ένα διάνυσμα πιθανοτήτων σε όλο το λεξιλόγιό μας: $y_t = \text{softmax}(O * s_t)$

Η ικανότητα διατήρησης πληροφοριών οδηγεί σε διαφορετικές κατηγορίες προβλημάτων που αντιμετωπίσαμε για τα CNN. Για παράδειγμα, τα μοντέλα RNNs μπορούν να χρησιμοποιηθούν για την πρόβλεψη της πιθανότητας της επόμενης λέξης σε μια ακολουθία, γεγονός που μας επιτρέπει να παράγουμε νέο κείμενο με δειγματοληψία από τις πιθανότητες εξόδου, ή για τη δημιουργία ενός αυτόματου διορθωτή προτάσεων.

Θεωρητικά τα RNN μπορούν να αξιοποιήσουν πληροφορίες σε αυθαίρετα μεγάλες ακολουθίες, αλλά στην πράξη περιορίζονται στο να κοιτάζουν πίσω μόνο μερικά βήματα (αυτό ονομάζεται "Το πρόβλημα της εξαφανιζόμενης κλίσης", ανατρέξτε στο [7] για περισσότερες λεπτομέρειες). Αυτό σημαίνει ότι με έναν μόνο νευρώνα, αναγκάζοντας $y_t = x_{t+1}$ μπορούμε να προσπαθήσουμε να προβλέψουμε την επόμενη λέξη σε μια πρόταση κοιτάζοντας την προηγούμενη- θεωρητικά χάρη στην αντίστροφη ροή πληροφοριών, ένας μόνο νευρώνας θα μπορούσε να "θυμάται" τι συνέβη n

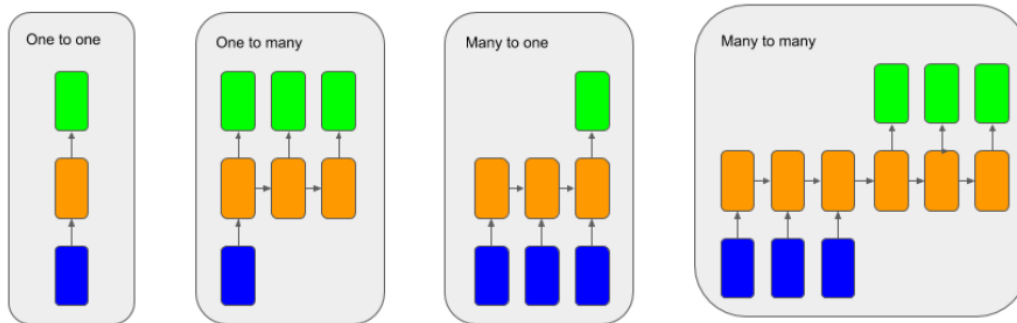
βήματα πριν. Συνήθως η καλύτερη πρακτική είναι να χρησιμοποιούμε περισσότερους νευρώνες για να έχουμε ένα πιο σταθερό δίκτυο: με τη επαναλαμβανόμενους νευρώνες μπορούμε να προσπαθήσουμε να προβλέψουμε την $m+1$ λέξη σε μια ακολουθία ή να χρησιμοποιήσουμε περισσότερα επίπεδα RNN για να δημιουργήσουμε ένα βαθύ νευρωνικό δίκτυο. Σε αυτή την περίπτωση αυξάνουμε την ικανότητα μνήμης (βραχυπρόθεσμη μνήμη), αλλά και πάλι θα αποτύχουμε μακροπρόθεσμα.

Στην πράξη δεν λειτουργεί λόγω της εξαφάνισης της κλίσης, κυρίως επειδή η εκπαίδευση ενός RNN είναι παρόμοια με την εκπαίδευση ενός παραδοσιακού Νευρωνικού Δικτύου. Όπως αναφέρθηκε σε προηγούμενα μαθήματα, χρησιμοποιούμε τον αλγόριθμο backpropagation, αλλά με μια μικρή ανατροπή. Δεδομένου ότι οι παράμετροι είναι κοινές για όλα τα χρονικά βήματα του δικτύου, η κλίση σε κάθε έξοδο εξαρτάται όχι μόνο από τους υπολογισμούς του τρέχοντος χρονικού βήματος, αλλά και από τα προηγούμενα χρονικά βήματα. Με λίγα λόγια, το νευρωνικό δίκτυο κατά τη φάση της εκπαίδευσης είναι επιρρεπές στο να ξεχνάει τι έχει δει στο παρελθόν και επειδή είναι "πιο σημαντικό" αυτό που έχει δει πρόσφατα. Για παράδειγμα, για να υπολογίσουμε την κλίση στο θ πρέπει να κάνουμε backpropagation (οπισθοδιάδοση) 3 βήματα και να αθροίσουμε τις κλίσεις. Αυτό ονομάζεται οπισθοδιάδοση μέσω του χρόνου (BPTT). Τέλος πάντων, υπάρχουν κάποια μηχανήματα για την αντιμετώπιση αυτών των προβλημάτων και ορισμένοι τύποι RNN (όπως τα LSTM) σχεδιάστηκαν ειδικά για να τα παρακάμπτουν.

Σε ορισμένα άλλα προβλήματα της πραγματικής ζωής, όπως οι μεταφράσεις μεταξύ γλωσσών, δεν αρκεί να εξετάσουμε το παρελθόν, αλλά μερικές φορές είναι απαραίτητο να εξετάσουμε και τις επόμενες λέξεις. Σε αυτή την περίπτωση μπορούμε να πάρουμε δύο RNN που εξετάζουν την ίδια πρόταση: το πρώτο από την πρώτη λέξη μέχρι την τελευταία και το δεύτερο προς τα πίσω (από την τελευταία στην πρώτη). Αυτά τα δίκτυα ονομάζονται BRNNs or bidirectional RNNs και ένα απλό σχήμα βρίσκεται στο κάτω μέρος της διαφάνειας. Σημειώστε ότι στη μέση θα μπορούσαμε να έχουμε πολλαπλά στρώματα για να αυξήσουμε την ισχύ του δικτύου, αντίθετα, χρειαζόμαστε όλο και περισσότερα δεδομένα για να εκπαιδευτούμε.

Μια άλλη ενδιαφέρουσα εφαρμογή των CNN και RNN είναι η δημιουργία περιγραφών εικόνων, αν σας ενδιαφέρει, ανατρέξτε στο [9].

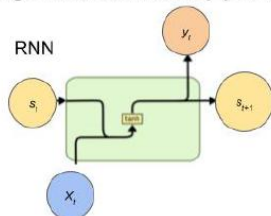
Τοπολογίες επαναλαμβανόμενων νευρωνικών δικτύων (Recurrent Neural Networks Topologies)



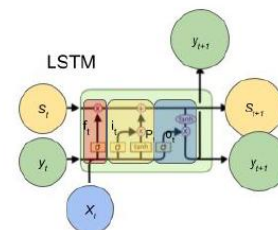
Εδώ μπορείτε να βρείτε μια σύντομη περιγραφή για τις διάφορες τοπολογίες των RNN. Ουσιαστικά οι τοπολογίες που χρησιμοποιούνται εξαρτώνται από το πρόβλημα που αντιμετωπίζουμε και το σύνολο δεδομένων που έχουμε: η πρόβλεψη του επόμενου κόσμου θα μπορούσε να επιλυθεί με "ένα προς ένα" ή "πολλά προς ένα", η μετάφραση είναι "πολλά προς πολλά" κ.λπ.

Σημειώστε ότι κάθε τοπολογία χρειάζεται ένα κατάλληλο σύνολο δεδομένων για να εκπαιδευτεί και επίσης το σχήμα εκπαίδευσης: στο NLP για παράδειγμα μία λέξη κάθε φορά έναντι περισσότερων λέξεων σε ένα πλάνο.

Long Short Term Memory (LSTM)



VS



$$\begin{aligned}
 f_t &= \sigma(W_f [y_t, x_t] + b_f) & \text{Forget gate layer} \\
 i_t &= \sigma(W_i [y_t, x_t] + b_i) & \text{Input gate layer} \\
 P_t &= \tanh(W_p [y_t, x_t] + b_p) & \\
 o_t &= \sigma(W_o [y_t, x_t] + b_o) & \text{Output gate layer} \\
 S_{t+1} &= f_t S_t + i_t P_t & \text{New cell state} \\
 y_{t+1} &= o_t \tanh(S_{t+1}) & \text{New cell output}
 \end{aligned}$$

* RNN & LSTM images are from <http://colah.github.io/posts/2015-08-Understanding-LSTMs/> and adapted to the slide
 * tanh & sigmoid image is from <https://towardsdatascience.com/activation-functions-neural-networks-13bd9f3d91d8>

Όπως είπαμε στην εισαγωγή των RNNs, θέλουμε να ξεπεράσουμε το πρόβλημα της εξαφανιζόμενης κλίσης, γι' αυτό πρέπει να εισάγουμε μια παραλλαγή των αναδρομικών δικτύων που ονομάζεται Μακροχρόνια Βραχυπρόθεσμη Μνήμη. Τα LSTM έχουν επίσης μια δομή που μοιάζει με αλυσίδα, αλλά η επαναλαμβανόμενη μονάδα έχει διαφορετική δομή. Αντί για ένα μόνο στρώμα, υπάρχουν τρία στρώματα που αλληλεπιδρούν με έναν πολύ ιδιαίτερο τρόπο.

Το επίπεδο χρησιμοποιεί δύο κύριες συναρτήσεις

- $\tanh$ (domain $\mathbb{R}$, co-domain $(-1; 1)$)
- sigmoid (domain $\mathbb{R}$, codomain $(0; 1)$)

Η αλληλεπίδραση μεταξύ των στρώματων γίνεται κυρίως με δύο λειτουργίες:

- $\times$: multiplication
- $+$: sum

Τα στρώματα αποτελούνται από πύλες (κόκκινα, κίτρινα και μπλε μπλοκ στην εικόνα του LSTM), ας περιγράψουμε το καθένα:

αποφασίζεται ποια και πόση πληροφορία θα αφαιρεθεί. Η διαδικασία αυτή γίνεται από ένα επίπεδο με σιγμοειδή συνάρτηση το οποίο ονομάζεται "forget gate layer". Η έξοδος του επιπέδου είναι ένας αριθμός μεταξύ 0 και 1, όπου το 1 σημαίνει πως χρησιμοποιείται όλη η πληροφορία της κατάστασης

- **Forget gate:** το πρώτο βήμα είναι να αποφασίσουμε ποια και πόση πληροφορία θα "αφαιρεθεί" από την κατάσταση του κυττάρου. Η διαδικασία αυτή γίνεται από ένα στρώμα με σιγμοειδή συνάρτηση το οποίο ονομάζεται "forget gate layer" Κοιτάζει το h_t και την είσοδο x_t και εξάγει έναν αριθμό μεταξύ 0 και 1 για κάθε αριθμό στην κατάσταση του κυττάρου S_t . Κοιτάζοντας τη συν-περιοχή του σιγμοειδούς, το 1 αντιπροσωπεύει "κρατήστε το εντελώς", ενώ το 0 αντιπροσωπεύει "ξεφορτωθείτε το εντελώς".
- **Πύλη εισόδου(Input gate):** Το επόμενο βήμα είναι να αποφασίσουμε ποιες νέες πληροφορίες θα αποθηκεύσουμε στην κατάσταση του κελιού. Αυτό έχει δύο μέρη. Πρώτον, ένα σιγμοειδές στρώμα που ονομάζεται "στρώμα πύλης εισόδου" αποφασίζει ποιες τιμές θα ενημερώσουμε. Στη συνέχεια, ένα στρώμα $\tanh$ δημιουργεί ένα διάνυσμα νέων υποψήφίων τιμών (P_t) που θα μπορούσαν να προστεθούν στην κατάσταση. Η νέα κατάσταση του κελιού υπολογίζεται ξεκινώντας από αυτές τις τιμές.
- **Πύλη εξόδου(Output gate):** Τώρα πρέπει να αποφασίσουμε τι θα βγάλουμε στην έξοδο. Αυτό θα γίνει με βάση την κατάσταση του κυττάρου μας, αλλά με μια φιλτραρισμένη έκδοση: πρώτα, εκτελούμε ένα σιγμοειδές στρώμα που αποφασίζει ποια μέρη της κατάστασης του κυττάρου θα εξάγουμε, στη συνέχεια περνάμε την κατάσταση του κυττάρου από την $\tanh$ (για να ωθήσουμε τις τιμές να είναι μεταξύ -1 και 1) και την πολλαπλασιάζουμε με την έξοδο της σιγμοειδούς πύλης, έτσι ώστε να εξάγουμε μόνο τα μέρη που αποφασίσαμε.

Ανατρέξτε στους τύπους της διαφάνειας για λεπτομέρειες σχετικά με τα στρώματα πύλης και τον υπολογισμό της νέας κατάστασης και εξόδου. Σημειώστε ότι, όπως και για τα άλλα νευρωνικά δίκτυα, τα W_j και b_j (με $j \in \{f, i, P\}$) είναι βάρη και προκαταλήψεις που πρέπει να μάθουμε κατά τη φάση της εκπαίδευσης.

Έτσι, οι νευρώνες LSTM είναι πιο πολύπλοκοι από τους προηγούμενους, αλλά αν δεν χρειάζεται να τροποποιήσουμε τους εσωτερικούς υπολογισμούς (που είναι εντάξει σε αρκετά όλες τις κοινές εφαρμογές), το Keras προσφέρει όλα όσα χρειαζόμαστε για μια τυπική μακράς βραχυπρόθεσμης μνήμης.

Τελευταίο αλλά όχι λιγότερο σημαντικό: Οι LSTM προσπαθούν να ελαχιστοποιήσουν το πρόβλημα της εξαφάνισης της κλίσης, αλλά πληρώνουν υπολογιστικό κόστος και κόστος κατάληψης μνήμης σε σύγκριση με τα RNN.

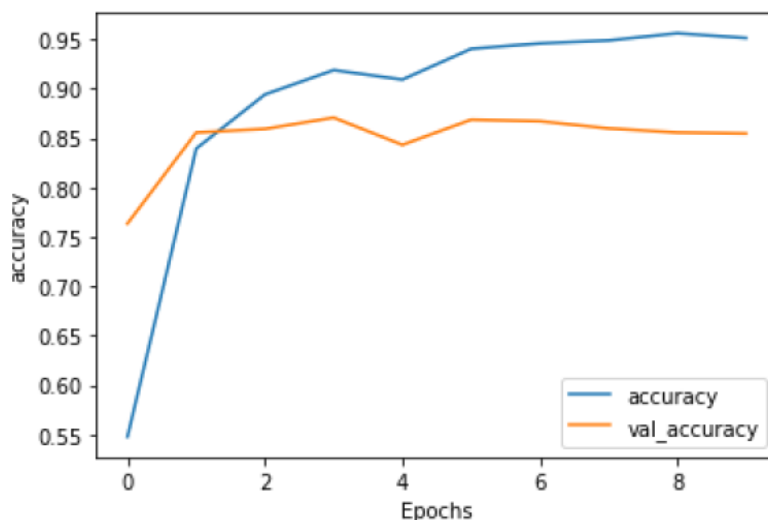
Ας δούμε τώρα μια εφαρμογή των LSTM και ας θεωρήσουμε το IMDB (ένα μεγάλο σύνολο δεδομένων κριτικών ταινιών) όπου όλες οι κριτικές έχουν είτε θετικό είτε αρνητικό συναίσθημα. Ο στόχος είναι να διδάξουμε ένα NN να προβλέπει το συναίσθημα μιας κριτικής).

Εδώ θα δείτε πώς να ορίσετε ένα μοντέλο LSTM με Python και Tensorflow για δυαδική ταξινόμηση:

```
import tensorflow as tf

model = tf.keras.Sequential([
    tf.keras.layers.Embedding(encoder.vocab_size, 64),
    tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(64)),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(1)
])
```

Μετά τη φάση εκπαίδευσης αυτού του ενιαίου στρώματος LSTM πήραμε το ακόλουθο αποτέλεσμα για το σύνολο επικύρωσης επί των epochs:



Και το ακόλουθο αποτέλεσμα στο σύνολο δοκιμών:

Απώλεια δοκιμής: 0,43040600419044495

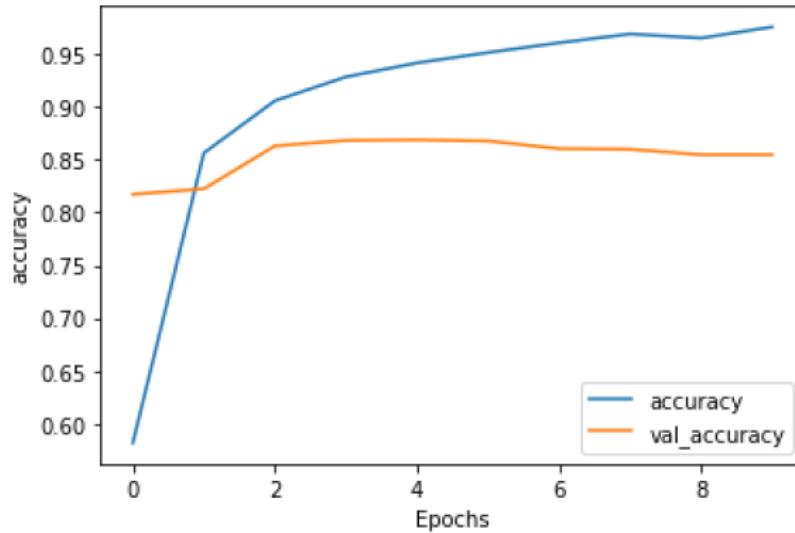
Ακρίβεια δοκιμής: 0,8578799962997437

Προσθήκη ενός ακόμη στρώματος LSTM στο μοντέλο:

```
model = tf.keras.Sequential([
    tf.keras.layers.Embedding(encoder.vocab_size, 64),
    tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(64, return_sequences=True)),
    tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(32)),
    tf.keras.layers.Dense(64, activation='relu'),
```

```
tf.keras.layers.Dropout(0.5),
tf.keras.layers.Dense(1)
])
```

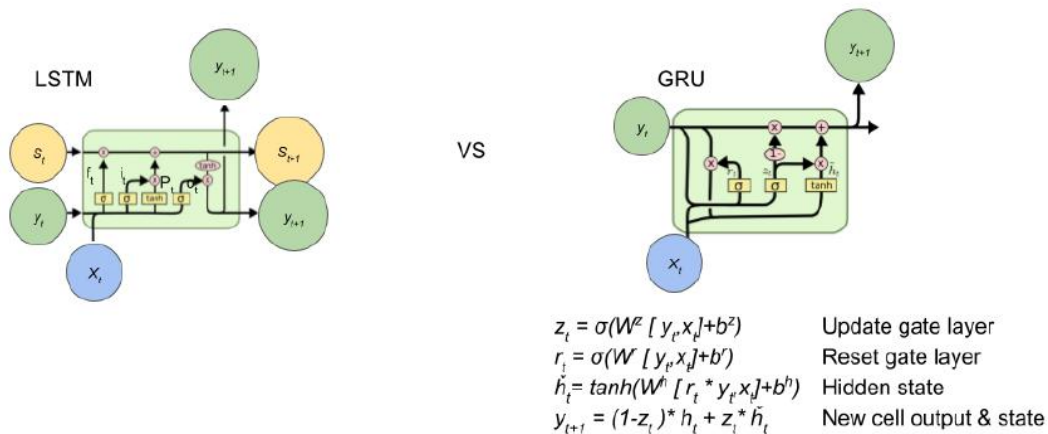
εδώ μπορείτε να βρείτε τα αποτελέσματα για το ίδιο σύνολο δεδομένων:



Απώλεια δοκιμής: 0.4876300096511841

Ακρίβεια δοκιμής: 0,8517199754714966

Για ένα πλήρες παράδειγμα ανατρέξτε σε [αυτόν](#) τον σύνδεσμο, ή δείτε το αρχείο 4.2_03text_classification_rnn.py



* images are from <http://colah.github.io/posts/2015-08-Understanding-LSTMs/> and adapted to the slide

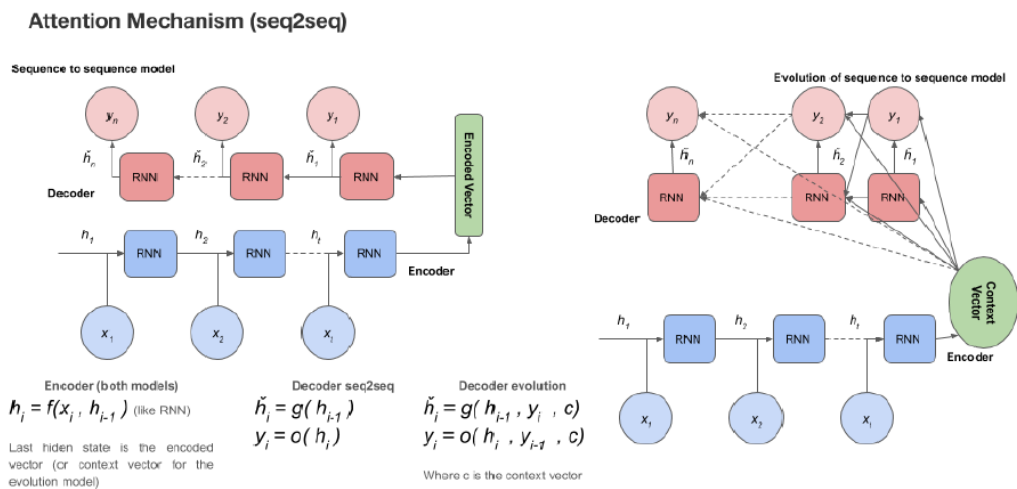
Μια ελαφρώς πιο δραματική παραλλαγή της LSTM είναι η Gated Recurrent Unit, ή GRU, (περισσότερες πληροφορίες [11]) που χρησιμοποιείται για να μειώσει τον τεράστιο όγκο μνήμης και υπολογιστικών πόρων που χρειάζονται οι αρχικές LSTM.

Η διαφάνεια δείχνει τις διαφορετικές τοπολογίες: στην GRU η κατάσταση του κελιού είναι ίση με την έξοδο του κελιού, και συνδυάζει τις πύλες λήθης και εισόδου σε μια ενιαία "πύλη ενημέρωσης". Συγχωνεύει επίσης την κατάσταση του κελιού και την κρυφή κατάσταση, έτσι ώστε η πύλη ενημέρωσης z να επιλέγει αν η κρυφή κατάσταση θα ενημερωθεί με μια νέα κρυφή κατάσταση $\tilde{h}$.

- Η πύλη επαναφοράς (reset gate) r αποφασίζει αν η κρυφή κατάσταση αγνοείται ή όχι. Όταν η πύλη επαναφοράς είναι κοντά στο 0, η κρυφή κατάσταση αναγκάζεται να αγνοήσει την προηγούμενη κρυφή κατάσταση και να επαναφερθεί μόνο με την τρέχουσα είσοδο. Αυτό ουσιαστικά επιτρέπει στην κρυφή κατάσταση να εγκαταλείψει κάθε πληροφορία που κρίνεται άσχετη αργότερα στο μέλλον, επιτρέποντας έτσι μια πιο συμπαγή αναπαράσταση.
- Από την άλλη πλευρά, η πύλη ενημέρωσης ελέγχει πόσες πληροφορίες από την προηγούμενη κρυφή κατάσταση θα μεταφερθούν στην τρέχουσα κρυφή κατάσταση. Αυτό λειτουργεί παρόμοια με το κύτταρο μνήμης στο δίκτυο LSTM και βοηθά το RNN να θυμάται μακροπρόθεσμες πληροφορίες.

Το μοντέλο που προκύπτει είναι απλούστερο από τα τυπικά μοντέλα LSTM και από άποψη υπολογισμού και κατάληψης μνήμης και γίνεται ολοένα και πιο δημοφιλές.

Gated recurrent unit



Στην προηγούμενη διαφάνεια βλέπουμε πώς οι οικογένειες RNNs (LSTM και GRU) προσπαθούν να μάθουν από όλες τις πληροφορίες που ενσωματώνονται σε μια ακολουθία δεδομένων (λέξεις στο NLP). Τώρα θα μιλήσουμε για ένα νέο μοντέλο που προσπαθεί να επιλέξει και να μάθει από τις "σχετικές" πληροφορίες: χρησιμοποιεί την "προσοχή". Αυτός ο μηχανισμός είναι μία από τις πρόσφατες εξελίξεις στις εφαρμογές εργασιών επεξεργασίας φυσικής γλώσσας όπως:

- μηχανική μετάφραση - ένα υποπεδίο της υπολογιστικής γλωσσολογίας που ερευνά τη χρήση λογισμικού για τη μετάφραση κειμένου ή ομιλίας από μια γλώσσα σε μια άλλη.
- image captioning - επίσης γνωστό ως αυτόματος σχολιασμός εικόνας (επίσης γνωστό ως αυτόματη σήμανση εικόνας ή γλωσσική ευρετηρίαση) είναι η διαδικασία με την οποία ένα

σύστημα υπολογιστή αποδίδει αυτόματα μεταδεδομένα με τη μορφή λεζάντας ή λέξεων-κλειδιών σε μια ψηφιακή εικόνα.

- παραγωγή διαλόγου - (ή chatbot) είναι μια εφαρμογή λογισμικού που χρησιμοποιείται για τη διεξαγωγή μιας συνομιλίας μέσω διαδικτυακής συνομιλίας μέσω κειμένου ή μετατροπής κειμένου σε ομιλία, αντί της παροχής άμεσης επαφής με έναν ζωντανό ανθρώπινο πράκτορα. αναγνώριση ομιλίας - είναι ένα διεπιστημονικό υποπεδίο της επιστήμης των υπολογιστών και της υπολογιστικής γλωσσολογίας που αναπτύσσει μεθοδολογίες και τεχνολογίες που επιτρέπουν την αναγνώριση προφορικού λόγου σε κείμενο από υπολογιστές

(*όλοι οι ορισμοί είναι από τη Βικιπαίδεια)

Η προσοχή έχει αναπτυχθεί για να αυξήσει την απόδοση του μοντέλου RNN κωδικοποιητή αποκωδικοποιητή (seq2seq) και, δεδομένου ότι πρόκειται για μια εφαρμογή, ας μιλήσουμε πρώτα εν συντομία για το μοντέλο seq2seq. Εισήχθη για πρώτη φορά το 2014 από την Google [14], ένα μοντέλο sequence-to-sequence έχει ως στόχο να αντιστοιχίσει μια είσοδο σταθερού μήκους με μια έξοδο σταθερού μήκους, όπου το μήκος της εισόδου και της εξόδου μπορεί να διαφέρει. Η διαδικασία αντιστοίχισης, εκτελείται αντίστοιχα από έναν κωδικοποιητή και έναν αποκωδικοποιητή, και στις δύο περιπτώσεις εκτελείται από ένα LSTM (στη διαφάνεια μας βλέπουμε ένα γενικότερο δίκτυο RNN, αλλά έχουμε ήδη συζητήσει τα πλεονεκτήματα και τα μειονεκτήματα των διαφόρων τύπων αναδρομικών νευρωνικών δικτύων).

Εδώ η περιγραφή των κύριων εξαρτημάτων:

- κωδικοποιητής
 - ο μια στοίβα από πολλές αναδρομικές μονάδες (κύτταρα LSTM ή GRU για καλύτερες επιδόσεις), όπου κάθε μία δέχεται ένα μόνο στοιχείο της ακολουθίας εισόδου, συλλέγει πληροφορίες για το στοιχείο αυτό και τις μεταδίδει προς τα εμπρός.
 - ο στο πρόβλημα της αυτόματης μετάφρασης η είσοδος x_i είναι μια ενιαία παγκόσμια ενσωμάτωση και μια πρόταση έχει σταθερό μήκος t (το σταθερό μήκος είναι ένας περιορισμός για αυτό το μοντέλο)
 - ο έχουμε ήδη μιλήσει για τον υπολογισμό της κρυφής κατάστασης για διαφορετικούς τύπους RNNs
- κωδικοποιημένο διάνυσμα (ή πλαίσιο)
 - ο είναι η τελική κρυφή κατάσταση που παράγεται από το τμήμα κωδικοποιητή του μοντέλου.
 - ο αυτό το διάνυσμα έχει ως στόχο να ενθυλακώσει την πληροφορία για όλα τα στοιχεία εισόδου και έχει σταθερή διάσταση
 - ο λειτουργεί ως η αρχική κρυφή κατάσταση του τμήματος αποκωδικοποιητή του μοντέλου.
- αποκωδικοποιητής
 - ο μια στοίβα από πολλές αναδρομικές μονάδες όπου κάθε μία προβλέπει μια έξοδο y_i σε ένα χρονικό βήμα i
 - ο κάθε επαναλαμβανόμενη μονάδα δέχεται μια κρυφή κατάσταση από τις προηγούμενες μονάδες και παράγει μια έξοδο καθώς και τη δική της κρυφή κατάσταση
 - ο στο μεταφραστικό πρόβλημα, η έξοδος υπολογίζεται με softmax σε ολόκληρο το λεξιλόγιο εξόδου

Μια εξέλιξη του μοντέλου seq2seq είναι η τροποποίηση του αποκωδικοποιητή έτσι ώστε ο υπολογισμός της κρυφής κατάστασης και της εξόδου να γίνεται με ανάμειξη των πληροφοριών μεταξύ τους και του διανύσματος πλαισίου. Κοιτάξτε τη διαφάνεια για να δείτε τις λεπτομέρειες της εξίσωσης και το [11] για περισσότερες λεπτομέρειες.

Μια εξέλιξη του μοντέλου seq2seq είναι η τροποποίηση του αποκωδικοποιητή έτσι ώστε ο υπολογισμός της κρυφής κατάστασης και της εξόδου να γίνεται με ανάμειξη των πληροφοριών μεταξύ τους και του διανύσματος πλαισίου. Κοιτάξτε τη διαφάνεια για να δείτε τις λεπτομέρειες της εξίσωσης και το [11] για περισσότερες λεπτομέρειες.

Η ποιοτική ανάλυση δείχνει ότι η εξέλιξη seq2seq αποτυπώνει καλύτερα τις γλωσσικές κανονικότητες στον πίνακα φράσεων, εξηγώντας έμμεσα τις ποσοτικές βελτιώσεις στη συνολική απόδοση. Η περαιτέρω ανάλυση του μοντέλου αποκαλύπτει ότι αυτό το RNN μαθαίνει μια συνεχή χωρική αναπαράσταση μιας φράσης που διατηρεί τόσο τη σημασιολογική όσο και τη συντακτική δομή της φράσης.

Οι έννοιες που αποτελούν τον πυρήνα ενός μοντέλου seq2seq είναι λίγο-πολύ εύκολα κατανοητές, αλλά στην πράξη η κατηγορία των προβλημάτων που αντιμετωπίζουμε τώρα δεν είναι τόσο εύκολο να επιλυθούν.

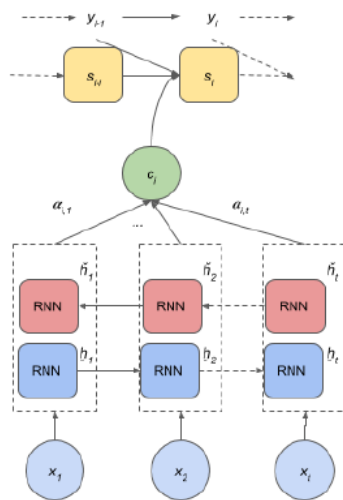
Για να έχετε μια ιδέα της πολυπλοκότητας όσον αφορά τους υπολογιστικούς πόρους που απαιτούνται για την εκπαίδευση ενός τέτοιου βαθύ νευρωνικού δικτύου, παραθέτουμε μια περίληψη από το [14]:

Εφαρμόσαμε τη μέθοδό μας στο έργο WMT'14 English to French MT[...] Χρησιμοποιήσαμε βαθιά LSTMs με 4 επίπεδα, με 1000 κελιά σε κάθε επίπεδο και 1000-διάστατες ενσωματώσεις λέξεων, με λεξιλόγιο εισόδου 160.000 και λεξιλόγιο εξόδου 80.000. [...] Έτσι, το βαθύ LSTM χρησιμοποιεί 8000 πραγματικούς αριθμούς για την αναπαράσταση μιας πρότασης. Το LSTM που προκύπτει έχει 384M παραμέτρους εκ των οποίων οι 64M είναι καθαρά αναδρομικές συνδέσεις (32M για το LSTM "κωδικοποιητή" και 32M για το LSTM "αποκωδικοποιητή").

Η ισχύς αυτού του μοντέλου έγκειται στο γεγονός ότι μπορεί να αντιστοιχίσει ακολουθίες διαφορετικού μήκους μεταξύ τους. Όπως μπορείτε να δείτε από τη διαφάνεια, οι εισοδοί και οι εξοδοί δεν συσχετίζονται και τα μήκη τους μπορεί να διαφέρουν.

Όπως συμβαίνει συχνά, κάθε μοντέλο έχει ένα ελάττωμα, και τα μοντέλα seq2seq δεν αποτελούν εξαίρεση. Η "Προσοχή" προτείνεται ως λύση στον περιορισμό του μοντέλου κωδικοποιητή-αποκωδικοποιητή, το οποίο κωδικοποιεί την ακολουθία εισόδου σε ένα διάνυσμα σταθερού μήκους (το κωδικοποιημένο διάνυσμα) από το οποίο θα αποκωδικοποιήσει την έξοδο σε κάθε χρονικό βήμα. Το ζήτημα αυτό αποτελεί πρόβλημα κατά την αποκωδικοποίηση μεγάλων ακολουθιών, διότι δυσκολεύει το νευρωνικό δίκτυο να ανταπεξέλθει σε μεγάλες προτάσεις.

Attention Mechanism



$$\begin{aligned}
 h &= (\hat{h}; \tilde{h}) && \text{annotation (encoder)} \\
 e_{ij} &= a(s_{i-1}, h_j) && \text{alignment } i, j \\
 \alpha_{ij} &= \varphi(e_{ij}) && \text{weight of annotation } h_j \\
 c_i &= \sum_{j=1}^n \alpha_{ij} h_j && \text{i-th context vector} \\
 s_i &= f(s_{i-1}, y_{i-1}, c_i) && \text{decoder hidden state} \\
 y_i &= o(s_i, y_{i-1}) && \text{output}
 \end{aligned}$$

Attention Model

Encoder: is a bi-directional RNN

Context vector: unlike the previous encoder-decoder approach, here the probability is conditioned on a distinct context vector c_i for each target word y_i .

Decoder: The decoder works as seq2seq enacment model

Η βασική ιδέα στο Attention είναι ότι κάθε φορά που το μοντέλο προσπαθεί να προβλέψει μια λέξη εξόδου, χρησιμοποιεί μόνο τμήματα μιας εισόδου όπου συγκεντρώνονται οι πιο σχετικές πληροφορίες αντί για μια ολόκληρη πρόταση, οπότε προσπαθεί να δώσει μεγαλύτερη σημασία στις λίγες λέξεις εισόδου (σημειώστε ότι στο προηγούμενο μοντέλο υποθέτουμε ότι όλα τα τμήματα της ακολουθίας εισόδου είναι σχετικά με κάθε λέξη στην έξοδο).

Τώρα αλλάζουμε το μοντέλο, αλλά πρώτα απ' όλα πρέπει να εισάγουμε δύο έννοιες για τη μηχανική μετάφραση:

- Ευθυγράμμιση: είναι το πρόβλημα στη μηχανική μετάφραση που προσδιορίζει ποια μέρη της ακολουθίας εισόδου είναι σχετικά με κάθε λέξη στην έξοδο.
- Μετάφραση: είναι η διαδικασία χρήσης των σχετικών πληροφοριών για την επιλογή της κατάλληλης εξόδου.

Η προσοχή προτείνεται ως λύση τόσο για την ευθυγράμμιση όσο και για τη μετάφραση και η ιδέα πίσω από αυτό το μοντέλο είναι ότι η χρήση ενός διανύσματος σταθερού μήκους είναι ένα σημείο συμφόρησης στη βελτίωση της απόδοσης αυτής της βασικής αρχιτεκτονικής κωδικοποιητή-αποκωδικοποιητή, ώστε να επεκταθεί αυτό επιτρέποντας σε ένα μοντέλο να αναζητά αυτόματα τμήματα μιας αρχικής πρότασης που είναι σχετικά με την πρόβλεψη μιας λέξης-στόχου, χωρίς να χρειάζεται να σχηματίζονται αυτά τα τμήματα ως σκληρό τμήμα ρητά.

Σε αυτό το μοντέλο ο αποκωδικοποιητής είναι ένα BIRNN και λαμβάνεται ένα annotation h_i συνδυάζοντας τις δύο κρυφές καταστάσεις (μία προς τα εμπρός και μία προς τα πίσω), έτσι ώστε να περιέχει τις περιλήψεις τόσο των προηγούμενων όσο και των επόμενων λέξεων.

Η ευθυγράμμιση e_{ij} που βαθμολογεί πόσο καλά ταιριάζουν οι είσοδοι γύρω από τη θέση j και η έξοδος στη θέση i , υπολογίζεται

Στη συνέχεια, οι βαθμολογίες ευθυγράμμισης κανονικοποιούνται χρησιμοποιώντας μια συνάρτηση softmax. Η κανονικοποίηση των βαθμολογιών επιτρέπει την αντιμετώπισή τους ως πιθανότητες, υποδεικνύοντας την πιθανότητα κάθε κωδικοποιημένου χρονικού βήματος εισόδου (σχολιασμός) να είναι σχετικό με το τρέχον χρονικό βήμα εξόδου.

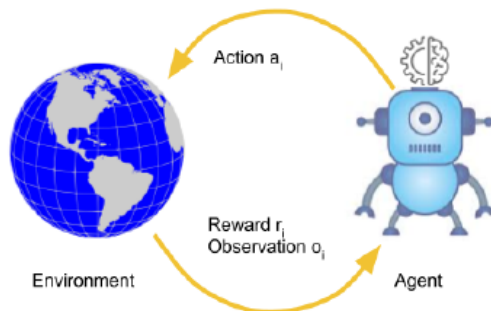
Το διάνυσμα συμφραζομένων c_i εξαρτάται από μια ακολουθία σχολίων ($h_1, \dots, h_t$) στις οποίες ένας κωδικοποιητής αντιστοιχίζει την πρόταση εισόδου. Κάθε σχολιασμός h_i περιέχει πληροφορίες για ολόκληρη την ακολουθία εισόδου με ιδιαίτερη έμφαση στα μέρη που περιβάλλουν την i -οστή λέξη της ακολουθίας εισόδου.

Στη συνέχεια, οι βαθμολογίες ευθυγράμμισης κανονικοποιούνται χρησιμοποιώντας μια συνάρτηση softmax. Η κανονικοποίηση των βαθμολογιών επιτρέπει την αντιμετώπισή τους ως πιθανότητες, υποδεικνύοντας την πιθανότητα κάθε κωδικοποιημένου χρονικού βήματος εισόδου (σχολιασμός) να είναι σχετικό με το τρέχον χρονικό βήμα εξόδου.

Το διάνυσμα συμφραζομένων c_i εξαρτάται από μια ακολουθία σχολίων ($h_1, \dots, h_t$) στις οποίες ένας κωδικοποιητής αντιστοιχίζει την πρόταση εισόδου. Κάθε σχολιασμός h_i περιέχει πληροφορίες για ολόκληρη την ακολουθία εισόδου με ιδιαίτερη έμφαση στα μέρη που περιβάλλουν την i -οστή λέξη της ακολουθίας εισόδου.

Έτσι, ας κάνουμε μια ανασύσταση: το i -οστό διάνυσμα συμφραζομένων c_i είναι ο αναμενόμενος σχολιασμός πάνω σε όλους τους σχολιασμούς με πιθανότητες a_{ij} . Η πιθανότητα a_{ij} , ή η σχετική ενέργεια e_{ij} , αντικατοπτρίζει τη σημασία του σχολιασμού h_j σε σχέση με την προηγούμενη κρυφή κατάσταση s_{i-1} για την απόφαση της επόμενης κατάστασης s_i και τη δημιουργία του y_i . Διαισθητικά, αυτό υλοποιεί έναν μηχανισμό προσοχής στον αποκωδικοποιητή. Ο αποκωδικοποιητής αποφασίζει τα μέρη της αρχικής πρότασης στα οποία θα δώσει προσοχή. Αφήνοντας τον αποκωδικοποιητή να έχει έναν μηχανισμό προσοχής, απαλλάσσουμε τον κωδικοποιητή από το βάρος της κωδικοποίησης όλων των πληροφοριών της αρχικής πρότασης σε ένα διάνυσμα σταθερού μήκους. Με αυτή τη νέα προσέγγιση οι πληροφορίες μπορούν να διασκορπιστούν σε όλη την ακολουθία των σχολίων, οι οποίες μπορούν να ανακτηθούν επιλεκτικά από τον αποκωδικοποιητή αναλόγως.

Reinforcement Learning



An **Agent** interacting with unknown **Environment**

- The Agent act (performing actions) and modify the Environment
- The Agent can look (Observe) at the Environment
- The Agent receive a Reward from each action

Goal: select action to maximize a future cumulative reward.

Experience: is a sequence of actions, observations and rewards.

State: a function over the experience space $s_t = f(a_1, o_1, r_1, \dots, a_t, o_t, r_t)$

Policy: a function that map the state space to action space.

($s_t \rightarrow a_{t+1}$) and could be deterministic or probabilistic.

Value function:

- from state and action (s_t, a_{t+1})
- under the policy π

gives expected future total reward

Η ιδέα της μάθησης μέσω της αλληλεπίδρασης με το περιβάλλον είναι ίσως το πρώτο πράγμα που μας έρχεται στο μυαλό όταν σκεφτόμαστε τη φύση και το πώς οι άνθρωποι και τα ζώα μαθαίνουν πώς να "επιβιώνουν". Όλοι μας λαμβάνουμε μια ανταμοιβή για κάθε πράξη που κάνουμε στη ζωή μας: άλλοτε μια θετική ανταμοιβή και άλλοτε μια αρνητική. Το σύνολο των ανταμοιβών είναι αυτό που ονομάζουμε "μάθημα που πήραμε" για ένα συγκεκριμένο σύνολο ενεργειών (ο κάστοράς μας). Οι μελλοντικές μας ενέργειες εξαρτώνται από αυτά που μάθαμε στο παρελθόν... αλλά αυτό δεν αρκεί, τα πράγματα είναι πιο πολύπλοκα στον πραγματικό λόγο.

Η ενισχυτική μάθηση (RL) είναι ο τρόπος με τον οποίο πραγματοποιείται αυτή η ιδέα στη μηχανική μάθηση και είναι ένα από τα τρία βασικά παραδείγματα, μαζί με την επιβλεπόμενη μάθηση και τη μη επιβλεπόμενη μάθηση.

Στην RL ο πράκτορας ενεργεί στο περιβάλλον, η αλληλεπίδραση αυτή τροποποιεί το περιβάλλον και παράγει μια ανταμοιβή που ο πράκτορας μπορεί να λάβει παράλληλα με την παρατήρηση των αλλαγών που παράγει η δράση. Ο στόχος είναι η μεγιστοποίηση της συνολικής ανταμοιβής (η ανταμοιβή που πήραν οι πράκτορες στο "τέλος του παιχνιδιού").

Αυτό το μοντέλο διαφέρει από την επιβλεπόμενη μάθηση στο ότι δεν χρειάζεται να παρουσιάζονται ζεύγη εισόδου/εξόδου με ετικέτες και ότι δεν χρειάζεται να διορθώνονται ρητά οι υπο-βέλτιστες ενέργειες. Αυτός ο τομέας της μηχανικής μάθησης ασχολείται με τον τρόπο με τον οποίο οι πράκτορες λογισμικού πρέπει να αναλαμβάνουν δράσεις σε ένα περιβάλλον προκειμένου να μεγιστοποιήσουν την ανταμοιβή (ή τη σωρευτική ανταμοιβή).

Μία από τις προκλήσεις που προκύπτουν στην ενισχυτική μάθηση, και όχι σε άλλα είδη μάθησης, είναι η αντιστάθμιση μεταξύ εξερεύνησης και εκμετάλλευσης. Για να επιτύχει μια μεγάλη συνολική ανταμοιβή, ένας πράκτορας ενισχυτικής μάθησης πρέπει να προτιμά ενέργειες που έχει δοκιμάσει στο παρελθόν και έχει διαπιστώσει ότι είναι ενεργές στην παραγωγή ανταμοιβής. Αλλά για να ανακαλύψει τέτοιες ενέργειες, πρέπει να δοκιμάσει ενέργειες που δεν έχει επιλέξει στο παρελθόν. Ο πράκτορας πρέπει να εκμεταλλευτεί αυτό που έχει ήδη βιώσει προκειμένου να λάβει ανταμοιβή, αλλά πρέπει επίσης να εξερευνήσει προκειμένου να κάνει καλύτερες επιλογές ενεργειών στο μέλλον. Το δίλημμα είναι ότι ούτε η εξερεύνηση ούτε η εκμετάλλευση μπορούν να επιδιωχθούν αποκλειστικά χωρίς να αποτύχουν στο έργο. Ο πράκτορας πρέπει να δοκιμάσει μια

ποικιλία ενεργειών και να προτιμήσει προοδευτικά εκείνες που φαίνονται καλύτερες. Σε μια στοχαστική εργασία, κάθε ενέργεια πρέπει να δοκιμαστεί πολλές φορές για να αποκτηθεί μια αξιόπιστη εκτίμηση της αναμενόμενης ανταμοιβής της. Το δίλημμα εξερεύνησης-εκμετάλλευσης έχει μελετηθεί εντατικά από τους μαθηματικούς (βλ. [18] για πολλές λεπτομέρειες) εδώ και πολλές δεκαετίες, ωστόσο παραμένει άλυτο.

Άλλα δύο σημεία που πρέπει να λάβετε υπόψη:

- Η RL αποτελεί επανάσταση στη μηχανική μάθηση, υπό την έννοια ότι οι ερευνητές προσπαθούν για πρώτη φορά να αποδείξουν ότι για να "αφήσουμε μια μηχανή να μάθει όπως οι άνθρωποι" χρειάζεται ένα σώμα για να αλληλεπιδράσει με το περιβάλλον.
- Οι ερευνητές της RL χρειάζονται πράκτορες λογισμικού που "ζουν" σε ένα περιβάλλον λογισμικού για να είναι ταχύτεροι κατά τη διάρκεια της διαδικασίας μάθησης. Η εκπαίδευση από το μηδέν ενός μοντέλου RL υλικού είναι εξαιρετικά χρονοβόρα και στην πραγματικότητα είναι δυνατή μόνο για "εύκολες" εργασίες. Η απόδοση είναι σημαντική επειδή η αξιολόγηση του συστήματος συχνά γίνεται ταυτόχρονα με την εκμάθηση.

Υπάρχουν πολλά διαφορετικά μοντέλα RL και η πολυπλοκότητά τους αυξάνεται ανάλογα με την πολυπλοκότητα του προβλήματος που αντιμετωπίζουμε. Αυτό σημαίνει ότι η δουλειά του πράκτορα να μάθει μια πολιτική, που αντιστοιχίζει καταστάσεις σε ενέργειες, η οποία μεγιστοποιεί κάποιο μακροπρόθεσμο μέτρο ενίσχυσης, θα μπορούσε να είναι πιο δύσκολη για παράδειγμα αν υποθέσουμε ότι το περιβάλλον δεν είναι στάσιμο, οπότε οι πιθανότητες να γίνουν μεταβάσεις καταστάσεων ή να ληφθούν συγκεκριμένα σήματα ενίσχυσης θα μπορούσαν να αλλάξουν με την πάροδο του χρόνου [16].

Ακολουθεί μια σύντομη περίληψη της ταξινόμιας της Ενισχυτικής Μάθησης [24]:

On policy vs Off Policy

Η διάκριση μεταξύ αλγορίθμων on-policy και off-policy αφορά μόνο τη φάση εκπαίδευσης.

Ένας Off Policy αλγόριθμος είναι ένας αλγόριθμος ο οποίος, κατά τη διάρκεια της εκπαίδευσης, χρησιμοποιεί μια πολιτική συμπεριφοράς που διαφέρει από τη βέλτιστη πολιτική που προσπαθεί να εκτιμήσει (τη βέλτιστη πολιτική). Για παράδειγμα, η εκμάθηση Q (βλ. επόμενη διαφάνεια) χρησιμοποιεί συχνά μια πολιτική ϵ -greedy:

- ϵ ποσοστό του χρόνου που επιλέγει μια τυχαία ενέργεια (διερευνητική ενέργεια)
- 1- ϵ ποσοστό του χρόνου που επιλέγει τη βέλτιστη ενέργεια, σύμφωνα με την τρέχουσα καλύτερη εκτίμησή του για τη βέλτιστη πολιτική (χρήση του τελεστή $\max$)

να εκμεταλλευτούν και να εξερευνήσουν το περιβάλλον.

Ένας On policy αλγόριθμος είναι ένας αλγόριθμος που, κατά τη διάρκεια της εκπαίδευσης, επιλέγει ενέργειες χρησιμοποιώντας μια πολιτική που προέρχεται από την τρέχουσα εκτίμηση της βέλτιστης πολιτικής, ενώ οι ενημερώσεις βασίζονται επίσης στην τρέχουσα εκτίμηση της βέλτιστης πολιτικής. Για παράδειγμα, ο SARSA είναι ένας αλγόριθμος on-policy επειδή δεν χρησιμοποιεί τον

τελεστή $\max$ στον κανόνα ενημέρωσης. Η διαφορά μεταξύ του Q-learning (off-policy) και του SARSA (on-policy) είναι αντίστοιχα η χρήση ή μη του τελεστή $\max$ στον κανόνα ενημέρωσής τους.

Βασισμένη στην πολιτική έναντι της αξίας (Policy based vs Value based)

Στις μεθόδους που βασίζονται σε πολιτικές δημιουργούμε ρητά μια αναπαράσταση μιας πολιτικής (απεικόνιση $\pi: s \rightarrow a$) και την κρατάμε στη μνήμη κατά τη διάρκεια της μάθησης. Στις μεθόδους που βασίζονται στην τιμή δεν αποθηκεύουμε καμία ρητή πολιτική, παρά μόνο μια συνάρτηση τιμής. Η πολιτική είναι εδώ έμμεση και μπορεί να προκύψει απευθείας από τη συνάρτηση αξίας (επιλέγουμε την ενέργεια με την καλύτερη τιμή).

Ο ηθοποιός-κριτικός (Actor-critic) είναι ένα μείγμα των δύο.

Βασισμένο σε μοντέλο vs. Ελεύθερο μοντέλο

Το πρόβλημα που αντιμετωπίζουμε συχνά στην RL είναι ότι κάθε φορά που βρίσκεστε στην κατάσταση s και κάνετε μια ενέργεια a , μπορεί να μην γνωρίζετε απαραίτητα την επόμενη κατάσταση s' στην οποία θα καταλήξετε (το περιβάλλον επηρεάζει τον πράκτορα).

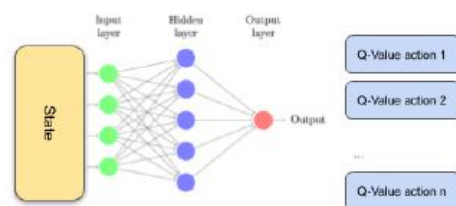
Σε μια προσέγγιση βασισμένη στο μοντέλο, είτε έχετε πρόσβαση στο μοντέλο (περιβάλλον) ώστε να γνωρίζετε την κατανομή πιθανότητας στις καταστάσεις στις οποίες καταλήγετε, είτε προσπαθείτε πρώτα να δημιουργήσετε ένα μοντέλο (συχνά - προσέγγιση) μόνοι σας. Αυτό μπορεί να είναι χρήσιμο επειδή σας επιτρέπει να κάνετε προγραμματισμό (πρόκειται για την πραγματοποίηση μελλοντικών κινήσεων χωρίς στην πραγματικότητα να εκτελείτε ενέργειες).

Στο Model-free δεν σας δίνεται ένα μοντέλο και δεν προσπαθείτε να καταλάβετε ρητά πώς λειτουργεί. Απλώς συλλέγετε κάποια εμπειρία και στη συνέχεια εξάγετε (ελπίζω) τη βέλτιστη πολιτική.

Reinforcement Learning with DNN

	State		
Action	$QV_{1,1}$	$QV_{1,2}$	...
	...	...	...
	...	...	...
	...	...	$QV_{8,8}$

Q-Value is a measure of the overall expected reward assuming the Agent is in state s and performs action a , and then continues playing until the end of the episode following some policy π . All possible values are represented in State/Action table. RL with DNN value based try to learn a State->Q-Value function in order to find the optimal policy. The size of the table is $|S| \times |A|$



Bellman equation

- S set of states s ,
- A set of actions a ,
- Reward function $R(s, a)$,
- Transition function $T(s, a, s') = P(s' | s, a)$,
- Discount factor γ .

$$Q^*(s, a) = E \left[R(s, a) + \gamma \sum_{s'} P(s' | s, a) \max_{a'} Q^*(s', a') \right]$$

the optimal policy π is then $\pi(s) = \operatorname{argmax}_{a \in A} Q^*(s, a)$.

Ας μιλήσουμε για την ενισχυτική μάθηση, τα βαθιά νευρωνικά δίκτυα και το NLP: αυτό σημαίνει πώς τα DNN θα μπορούσαν να εφαρμοστούν στο RL και ειδικότερα σε προβλήματα NLP.

Μια προσέγγιση είναι η βαθιά ενισχυτική μάθηση με βάση την τιμή, δηλαδή η εφαρμογή νευρωνικών δικτύων βαθιάς μάθησης για την εκμάθηση της συνάρτησης Q-value (Deep Q Networks ή DQN).

Q-Value: είναι ένα μέτρο της συνολικής αναμενόμενης ανταμοιβής υποθέτοντας ότι ο Πράκτορας βρίσκεται στην κατάσταση s και εκτελεί την ενέργεια a , και στη συνέχεια συνεχίζει να παίζει μέχρι το τέλος του επεισοδίου ακολουθώντας κάποια πολιτική π . Το όνομά του είναι συντομογραφία της λέξης "Quality" και ορίζεται μαθηματικά ως εξής:

$$Q_{\pi}(s,a) = E \sum_{n=0}^{\infty} \gamma^n r_n$$

όπου N είναι ο αριθμός των καταστάσεων από την κατάσταση s έως την τελική κατάσταση, γ είναι ο παράγοντας προεξόφλησης και r_0 είναι η άμεση ανταμοιβή που λαμβάνεται μετά την εκτέλεση της ενέργειας a στην κατάσταση s . Q-learning σημαίνει "εκμάθηση της συνάρτησης Q-Value".

Όπως φαίνεται στη διαφάνεια, η συνάρτηση Q-value θα μπορούσε να βρεθεί (να μάθει) μέσω ενός βαθύ νευρωνικού δικτύου, για να το εξηγήσουμε αυτό πρέπει να εισάγουμε την εξίσωση Bellman. Ας πάρουμε:

- ένα σύνολο καταστάσεων S ,
- ένα σύνολο ενεργειών A ,
- μια συνάρτηση ανταμοιβής $R(s, a)$,
- μια συνάρτηση μετάβασης $T(s, a, s') = P(s' | s, a)$,
- έναν συντελεστή προεξόφλησης γ .

Σε κάθε κατάσταση $s \in S$, ο πράκτορας αναλαμβάνει μια ενέργεια $a \in A$. Με την ανάληψη αυτής της ενέργειας, ο πράκτορας λαμβάνει μια ανταμοιβή $R(s, a)$ και φτάνει σε μια νέα κατάσταση s' , η οποία καθορίζεται από την κατανομή πιθανότητας $P(s' | s, a)$. Μια πολιτική π καθορίζει για κάθε κατάσταση ποια ενέργεια θα αναλάβει ο πράκτορας. Ο στόχος του πράκτορα είναι να βρει την πολιτική π που αντιστοιχίζει καταστάσεις σε ενέργειες και μεγιστοποιεί την αναμενόμενη προεξοφλημένη συνολική ανταμοιβή κατά τη διάρκεια της ζωής του πράκτορα. Η τιμή $Q_{\pi}(s, a)$ ενός δεδομένου ζεύγους κατάστασης-δράσης (s, a) είναι μια εκτίμηση της αναμενόμενης μελλοντικής ανταμοιβής που μπορεί να ληφθεί από την (s, a) όταν ακολουθείται η πολιτική π . Η βέλτιστη συνάρτηση αξίας $Q^*(s, a)$ παρέχει μέγιστες τιμές σε όλες τις καταστάσεις και προσδιορίζεται με την επίλυση της εξίσωσης Bellman:

$$Q^*(s, a) = E [R(s, a) + \gamma \sum_{s'} P(s' | s, a) \max_{a'} Q^*(s', a')]$$

Η βέλτιστη πολιτική π είναι $\pi(s) = \arg \max_{a \in A} Q^*(s, a)$.

Η DQN προσεγγίζει (υπό ορισμένες υποθέσεις) τη συνάρτηση αξίας $Q(s, a)$ με ένα βαθύ νευρωνικό δίκτυο που εξάγει ένα σύνολο τιμών δράσης $Q(s, -; \theta)$ για μια δεδομένη κατάσταση εισόδου s , όπου θ είναι οι παράμετροι του δικτύου [19].

Οι προϋποθέσεις για την εύρεση μιας βέλτιστης λύσης με την DQL είναι:

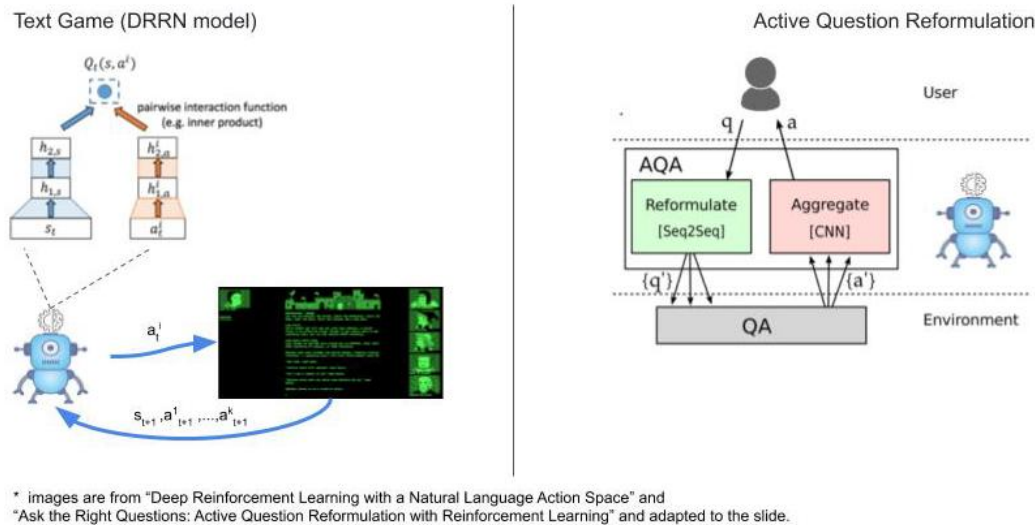
- Δεν υπάρχει συσχέτιση μεταξύ των παραδειγμάτων
- Σταθεροί στόχοι (είναι εκείνοι των οποίων η τιμή ανταμοιβής είναι στατική, δεν αλλάζει και από την άλλη πλευρά τα μη-σταθερά προβλήματα είναι εκείνα των οποίων η τιμή ανταμοιβής αλλάζει με το χρόνο) Μιλώντας για εφαρμογές, τα DQN εφαρμόζονται ευρέως σε παιχνίδια, όπου ο στόχος είναι να μάθουν και στη συνέχεια να προσπαθήσουν να παίξουν με έναν άνθρωπο παίκτη ή άλλο λογισμικό.

Το Deep-Q-Network έχει αποδειχθεί ότι είναι πολύ αποτελεσματικό σε προβλήματα όπου ο πράκτορας έχει ένα πολύ μικρό χώρο ενεργειών, αλλά μεγάλο χώρο καταστάσεων. Για παράδειγμα, στο παιχνίδι του σκακιού, ο χώρος καταστάσεων είναι ολόκληρη η σκακιέρα 20X20, αλλά ο χώρος δράσης είναι μικρός, καθώς ένα πιόνι μπορεί να κινηθεί μόνο προς 1-2 κατευθύνσεις.

Στην Python υπάρχει μια διάσημη βιβλιοθήκη που ονομάζεται GYM και είναι χρήσιμη για την εκμάθηση ενός πειράματος RL μοντέλων.

Ανατρέξτε σε αυτό το σεμινάριο για μια πρακτική εισαγωγή.

Reinforcement Learning with DNN for NLP



Η ενισχυτική μάθηση εφαρμόζεται στα ακόλουθα προβλήματα NLP:

- **Παιχνίδια κειμένου**
οι παίκτες διαβάζουν περιγραφές της τρέχουσας κατάστασης του κόσμου και ανταποκρίνονται με εντολές φυσικής γλώσσας για την ανάληψη ενεργειών. Δεδομένου ότι η υποκείμενη κατάσταση δεν είναι άμεσα παρατηρήσιμη, ο παίκτης πρέπει να κατανοήσει το κείμενο για να ενεργήσει.
- **Περίληψη άρθρων**
είναι η διαδικασία περίληψης ενός μακροσκελούς εγγράφου σε λίγες προτάσεις που αποτυπώνουν την ουσία ολόκληρου του εγγράφου.
- **Απάντηση ερωτήσεων**
ο στόχος είναι να ανακτηθεί η καλύτερη απάντηση σε μια πρωτοβουλία του χρήστη, και η κύρια απόφαση πολιτικής διαλόγου είναι αν θα δοθεί αυτή η καλύτερη απάντηση ή κάποιο άλλο είδος κίνησης (π.χ. αίτημα για διόρθωση, διευκρίνιση ή αλλαγή θέματος), όταν η καλύτερη απάντηση δεν φαίνεται να είναι αρκετά καλή.
- **Αναδιατύπωση ενεργού ερωτήματος**
ένας πράκτορας που βρίσκεται μεταξύ του χρήστη και ενός συστήματος QA μαύρου κουτιού και μαθαίνει να επαναδιατυπώνει τις ερωτήσεις ώστε να αποσπά τις καλύτερες δυνατές απαντήσεις.
- **Παραγωγή διαλόγου**
- ...και άλλα

Δεδομένου του πλήθους των διαφορετικών προβλημάτων και εφαρμογών και της τεράστιας βιβλιογραφίας, θα επικεντρωθούμε μόνο σε δύο από αυτά: Κειμενικά παιχνίδια και ενεργητική αναδιατύπωση ερωτήσεων.

Παιχνίδια κειμένου (περίπτωση χρήσης 1) [20]

Σε ένα παιχνίδι βασισμένο σε κείμενο, ο παίκτης (ή το σύστημα, στην προκειμένη περίπτωση) λαμβάνει μια συμβολοσειρά κειμένου που περιγράφει την τρέχουσα κατάσταση του παιχνιδιού και διάφορες συμβολοσειρές κειμένου που περιγράφουν πιθανές ενέργειες που θα μπορούσε να κάνει κάποιος. Μετά την επιλογή μιας από τις ενέργειες, η κατάσταση του περιβάλλοντος ενημερώνεται και αποκαλύπτεται σε μια νέα κειμενική περιγραφή. Μια ανταμοιβή δίνεται είτε σε κάθε μετάβαση είτε στο τέλος. Ο στόχος είναι να κατανοήσουμε, σε κάθε βήμα, το κείμενο της κατάστασης και όλα τα κείμενα ενεργειών για να επιλέξουμε την πιο κατάλληλη ενέργεια, πλοηγούμενοι στην ακολουθία των κειμένων έτσι ώστε να επιτύχουμε την υψηλότερη μακροπρόθεσμη ανταμοιβή.

Σε αυτή την περίπτωση η έννοια της συνάφειας βασίζεται στην κοινή επίδραση κατάστασης/δράσης στην ανταμοιβή: μια συμβολοσειρά κειμένου δράσης λέγεται ότι είναι "πιο συναφής" (για μια συμβολοσειρά κειμένου κατάστασης) από τα άλλα κείμενα δράσης, εάν η ανάληψη αυτής της δράσης θα οδηγούσε σε υψηλότερη μακροπρόθεσμη ανταμοιβή.

Σημειώστε ότι όχι μόνο ο χώρος καταστάσεων S είναι διακριτός και τεράστιος, αλλά και ο χώρος ενεργειών A που περιγράφεται από συμβολοσειρές κειμένου φυσικής γλώσσας είναι διακριτός και δυνητικά απεριόριστος λόγω της εκθετικής πολυπλοκότητας της γλώσσας σε σχέση με το μήκος της πρότασης.

Τώρα παρουσιάζουμε το βαθύ δίκτυο ενίσχυσης της συνάφειας (DRRN) για να προσπαθήσουμε να λύσουμε αυτό το πρόβλημα (δείτε [20] για όλες τις λεπτομέρειες). Το DRRN χρησιμοποιεί ξεχωριστά βαθιά νευρωνικά δίκτυα για να αντιστοιχίσει τις συμβολοσειρές κειμένου κατάστασης και δράσης σε ξεχωριστά διανύσματα ενσωμάτωσης, από τα οποία η "συνάφεια" μετράται αριθμητικά με μια γενική συνάρτηση αλληλεπίδρασης, όπως το εσωτερικό τους γινόμενο. Η έξοδος αυτής της συνάρτησης αλληλεπίδρασης ορίζει την τιμή της συνάρτησης Q για το τρέχον ζεύγος κατάστασης-δράσης, η οποία χαρακτηρίζει τη βέλτιστη μακροπρόθεσμη ανταμοιβή για τη σύζευξη αυτών των δύο κειμενικών συμβολοσειρών. Η προσέγγιση της συνάρτησης Q μαθαίνεται από άκρο σε άκρο με τη μέθοδο Q -learning.

Για την εκμάθηση του DRRN, εφαρμόζεται η στρατηγική "επανάληψης εμπειριών" [21]. Αυτός ο αλγόριθμος μάθησης χρησιμοποιεί μια σταθερή πολιτική εξερεύνησης για την αλληλεπίδραση με το περιβάλλον ώστε να ληφθεί ένα δείγμα τροχιάς. Στη συνέχεια, δειγματοληπτεί τυχαία μια πλειάδα μετάβασης (s_k, a_k, r_k, s_{k+1}), υπολογίζει το σφάλμα χρονικής διαφοράς για το δείγμα k και το χρησιμοποιεί για την ενημέρωση των παραμέτρων του μοντέλου σύμφωνα με την κάθοδο κλίσης.

Για να αποτρέψουμε τους αλγορίθμους από το να "θυμούνται" τη σειρά κατάστασης-δράσης και να κάνουν επιλογές με βάση τη διατύπωση των δράσεων, κάθε φορά που ο αλγόριθμος/παίκτης διαβάζει κείμενο από τον προσομοιωτή, ανακατεύουμε τυχαία τον κατάλογο των δράσεων. Αυτό θα ενθαρρύνει τους αλγορίθμους να λαμβάνουν αποφάσεις με βάση την κατανόηση των κειμένων που περιγράφουν τις καταστάσεις και τις ενέργειες.

Η επιτυχία του DRRN στο χειρισμό ενός χώρου ενεργειών φυσικής γλώσσας A έγκειται στο γεγονός ότι το κείμενο κατάστασης και τα κείμενα ενεργειών απεικονίζονται σε ξεχωριστούς χώρους

ενσωμάτωσης πεπερασμένων διαστάσεων. Η από άκρη σε άκρη διαδικασία μάθησης καθιστά τα διανύσματα ενσωμάτωσης στους δύο χώρους περισσότερο ευθυγραμμισμένα για τα "καλά" (ή σχετικά) κείμενα δράσης σε σύγκριση με τις "κακές" (ή άσχετες) επιλογές, με αποτέλεσμα την υψηλότερη έξοδο της συνάρτησης αλληλεπίδρασης (τιμή συνάρτησης Q).

Ενεργή αναδιατύπωση ερωτήσεων (Περίπτωση χρήσης 2)

Αυτό το μοντέλο είναι εμπνευσμένο από την ικανότητα των ανθρώπων να θέτουν τις σωστές ερωτήσεις: ένας πράκτορας που μαθαίνει να εκτελεί αυτή τη διαδικασία για τον χρήστη. Βρίσκεται μεταξύ του χρήστη και ενός backend συστήματος QA που αναφέρεται ως "περιβάλλον". Ο πράκτορας ονομάζεται AQA, καθώς υλοποιεί μια ενεργή στρατηγική απάντησης ερωτήσεων και ο στόχος του είναι να μεγιστοποιήσει την πιθανότητα να λάβει τη σωστή απάντηση στέλνοντας μια αναδιατυπωμένη ερώτηση στο περιβάλλον. Ο πράκτορας προσπαθεί να βρει την καλύτερη απάντηση θέτοντας πολλές ερωτήσεις και συγκεντρώνοντας τα επιστρεφόμενα στοιχεία. Τα εσωτερικά στοιχεία του περιβάλλοντος δεν είναι διαθέσιμα στον πράκτορα, οπότε πρέπει να μάθει να διερευνά βέλτιστα ένα μαύρο κουτί χρησιμοποιώντας μόνο συμβολοσειρές ερωτήσεων.

Οι βασικές συνιστώσες του μοντέλου που περιγράφεται [23] είναι:

- Ένα προ-εκπαιδευμένο μοντέλο ακολουθίας σε ακολουθία που δέχεται ως είσοδο μια ερώτηση και επιστρέφει τις αναδιατυπώσεις της. Αυτή η εργασία είναι παρόμοια με τη μηχανική μετάφραση, τη μετάφραση από τα αγγλικά στα αγγλικά, και πράγματι το αρχικό μοντέλο μπορεί να χρησιμοποιηθεί για γενική παράφραση, αλλά πρέπει να προσαρμοστεί για να υποστηρίξει την εκπαίδευση με ενισχυτική μάθηση χρησιμοποιώντας μεθόδους πολιτικής κλίσης.
- Ένα μοντέλο επιλογής απαντήσεων. Ο επιλογέας απαντήσεων χρησιμοποιεί ένα συνελκτικό νευρωνικό δίκτυο και αποδίδει μια βαθμολογία σε κάθε τριπλέτα αρχικής ερώτησης, αναδιατύπωσης και απάντησης. Ο επιλογέας χρησιμοποιεί προ-εκπαιδευμένες ενσωματώσεις λέξεων (π.χ. GloVe [26]).
- Ένα σύστημα απάντησης ερωτήσεων (το περιβάλλον). Για τον σκοπό αυτό, χρησιμοποιούμε το BiDAF [27], ένα δημοφιλές σύστημα απάντησης ερωτήσεων.

Ένα επεισόδιο ξεκινά με μια αρχική ερώτηση q . Ο πράκτορας παράγει στη συνέχεια ένα σύνολο αναδιατυπώσεων $\{q_i\}_{i=1}^N$ έως N . Αυτές αποστέλλονται στο περιβάλλον το οποίο επιστρέφει απαντήσεις $\{a_i\}_{i=1}^N$ έως N . Το μοντέλο επιλογής επιλέγει στη συνέχεια την καλύτερη από αυτές τις υποψήφιες απαντήσεις.

Η μία πλευρά της συνομιλίας, ο πράκτορας AQA, προσπαθεί να προσαρμόσει τη γλώσσα του για να βελτιώσει την απάντηση της άλλης πλευράς, του περιβάλλοντος QA. Πραγματοποιήθηκε ποιοτική ανάλυση της γλώσσας που παράγεται από τον πράκτορα AQA. Το αποτέλεσμα είναι ότι οι αναδιατυπώσεις των ερωτήσεων του πράκτορα αποκλίνουν σημαντικά από τις παραφράσεις της φυσικής γλώσσας. Είναι αξιοσημείωτο, ωστόσο, ότι ο πράκτορας είναι σε θέση να μάθει μη τετριμμένες και διαφανείς πολιτικές. Συγκεκριμένα, ο πράκτορας είναι σε θέση να ανακαλύψει κλασικές λειτουργίες ερωτημάτων ανάκτησης πληροφοριών, όπως η επαναστάθμιση όρων, που μοιάζει με την tf-idf (συντομογραφία για term frequency-inverse document frequency), και η μορφολογική απλοποίηση/αποσύνδεση (morphological simplification/stemming). Ένας πιθανός

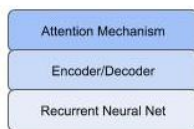
λόγος είναι ότι οι τρέχουσες εργασίες κατανόησης μηχανής περιλαμβάνουν την κατάταξη σύντομων αποσπασμάτων κειμένου, δίνοντας έτσι κίνητρα για τη συνάφεια, περισσότερο από τη βαθιά γλωσσική κατανόηση.

How techniques like RNN, Attention Mechanism, Reinforcement Learning relate to each other

Reinforcement Learning



Supervised Deep Learning



* The right plot is from osservatori.it "CASI DI APPLICAZIONE DI REINFORCEMENT LEARNING"

Όπως αναφέρθηκε στην προηγούμενη διαφάνεια, η ενισχυτική μάθηση είναι ένα εντελώς νέο παράδειγμα της μηχανικής μάθησης και είναι σχετικά νέα στη μηχανική μάθηση.

Το επίκεντρο αυτής της σύντομης διαφάνειας είναι να συγκρίνουμε τους κύριους αλγορίθμους NLP που περιγράφηκαν παραπάνω και να προσπαθήσουμε να κατανοήσουμε τις μεταξύ τους σχέσεις. Όπως εξηγήθηκε, η χρήση ενός αλγορίθμου παρά έναν άλλο εξαρτάται από πολλές μεταβλητές αλλά η κυριότερη είναι το πρόβλημα που επιλύουμε. Τώρα ξεχάστε το πρόβλημα μόνο για λίγα λεπτά και ας δούμε με το μάτι του πουλιού τι σπέρνουμε για τον NLP αλγόριθμο.

Πρώτα απ' όλα, παρουσιάσαμε τα Αναδρομικά Νευρωνικά Δίκτυα, στη συνέχεια παρουσιάσαμε τον Κωδικοποιητή/Αποκωδικοποιητή που χρησιμοποιούν, αποτελούν μια εξειδίκευση, μια ειδική περίπτωση των RNN. Στη συνέχεια, πάνω στον Κωδικοποιητή/Αποκωδικοποιητή εισάγουμε τον Μηχανισμό Προσοχής που αποτελεί μια περαιτέρω εξειδίκευση. Όλες αυτές οι τεχνικές και οι αλγόριθμοι ταξινομούνται ως εποπτευόμενα βαθιά νευρωνικά δίκτυα.

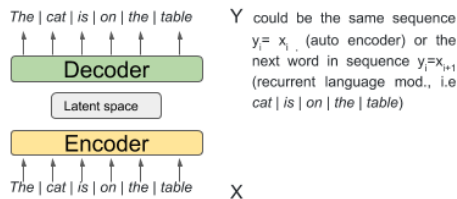
Στο δεξί διάγραμμα παρουσιάζεται η Προσδοκία μιας τεχνολογίας με την πάροδο του χρόνου, στις διάφορες φάσεις που θα παράγει η τεχνολογία με την πάροδο του χρόνου. Λοιπόν, τα Βαθιά Νευρωνικά Δίκτυα έχουν περάσει την "Κορυφή του Ενδιαφέροντος" και φαίνεται ότι κατεβαίνουν, περιμένοντας να περάσουν το "Ελάχιστο Απογοήτευσης" και στη συνέχεια τη "Φάση Παραγωγικότητας".

Η Ενισχυτική Μάθηση βρίσκεται στην αρχή του "Σταδίου Καινοτομίας" που υπόσχεται μια εξαιρετικά μεγάλη επιτυχία! Αυτό είναι το κύριος λόγος που παρουσιάζουμε αυτή την κατηγορία αλγορίθμων, και γι' αυτό θα είναι ενδιαφέρον να δούμε τι θα συμβεί στο μέλλον, και να συγκρίνουμε τις επιδόσεις με τα νευρωνικά βαθιά νευρωνικά δίκτυα, γνωρίζοντας ότι οι μικτές λύσεις που διατηρούνται απαιτούν την Ενισχυτική Μάθηση και τα βαθιά νευρωνικά δίκτυα αποτελούν τη φυσική εξέλιξη, ένα πιθανό βήμα προς τα εμπρός. Αυτό θα παράγει εξαιρετικά

πολύπλοκα αλγόριθμους που είναι εκτός του πεδίου εφαρμογής, οπότε τώρα ας επιστρέψουμε στο έδαφος και ας εξετάσουμε μια άλλη ενδιαφέρουσα κατηγορία αλγορίθμων, τα αυτούς που δεν χρησιμοποιούν ταξινομημένα δεδομένα για να μάθουν: τους μη επιβλεπόμενους.

Unsupervised Learning for NLP

Auto encoder / recurrent language model



Pretrain the model. Then use the encoder (the weights) to perform other tasks, i.e supervised test classification

Pre-training and Training NLP costs

Two distinct Eras of Compute Usage in Training AI Systems (Language)



* right images are from "Improving Language Understanding with Unsupervised Learning" and adapted to the slide

Η ικανότητα αποτελεσματικής μάθησης από ακατέργαστο κείμενο είναι ζωτικής σημασίας για την ανακούφιση της εξάρτησης από την επιβλεπόμενη μάθηση στην επεξεργασία φυσικής γλώσσας. Σχεδόν όλες οι μέθοδοι βαθιάς μάθησης απαιτούν τεράστιες ποσότητες δεδομένων με χειροκίνητη επισήμανση, γεγονός που περιορίζει την εφαρμογή τους σε πολλούς τομείς που υποφέρουν από έλλειψη σχολιασμένων πόρων. Σε αυτές τις περιπτώσεις, τα μοντέλα που μπορούν να αξιοποιήσουν γλωσσικές πληροφορίες από μη επισημειωμένα δεδομένα παρέχουν μια πολύτιμη εναλλακτική λύση στη συλλογή περισσότερων σχολίων, η οποία μπορεί να είναι χρονοβόρα και δαπανηρή. Επιπλέον, είναι δύσκολο να υπάρχει καλή ποιότητα σχολιασμένων δεδομένων και να μην υπάρχουν προκαταλήψεις. Περαιτέρω, ακόμη και σε περιπτώσεις όπου είναι διαθέσιμη σημαντική επίβλεψη, η εκμάθηση καλών αναπαραστάσεων με μη επιβλεπόμενο τρόπο μπορεί να προσφέρει σημαντική αύξηση της απόδοσης.

Ένα παράδειγμα μάθησης χωρίς επίβλεψη είναι ο αυτοκωδικοποιητής και είναι ο πιο διάσημος στη βαθιά μάθηση. Ένας αυτόματος κωδικοποιητής χρησιμοποιεί ένα μοντέλο κωδικοποιητή-αποκωδικοποιητή (το οποίο έχουμε ήδη δει στις διαφάνειες NLP) που προσπαθεί να προβλέψει την ίδια πρόταση εισόδου. Εντάξει, φαίνεται να μην είναι πολύ διασκεδαστικό και χρήσιμο, αλλά αυτό που κάνει αυτό το νευρωνικό δίκτυο είναι να μάθει μια συμπίεσμένη αναπαράσταση εισόδου.

Όπως μπορείτε να δείτε από τη διαφάνεια, αυτό το μοντέλο χρησιμοποιεί μέθοδο μάθησης χωρίς επίβλεψη, αν και τεχνικά, εκπαιδεύεται με μεθόδους μάθησης με επίβλεψη, που αναφέρονται ως αυτοεπιτήρηση. Αυτό που ονομάζεται λανθάνων χώρος, σε αυτή την περίπτωση, είναι το

εσωτερικό (κρυφό) στρώμα που περιγράφει έναν κώδικα που χρησιμοποιείται για την αναπαράσταση της εισόδου.

Ο σχεδιασμός του μοντέλου αυτόματου κωδικοποιητή υλοποιεί σκόπιμα την πρόκληση του περιορισμού της αρχιτεκτονικής σε ένα σημείο συμφόρησης στο μέσο του μοντέλου, από το οποίο πραγματοποιείται η ανακατασκευή των δεδομένων εισόδου.

Όπως και στις λειτουργίες κωδικοποιητή-αποκωδικοποιητή, υπάρχει ισορροπία μεταξύ της απώλειας πληροφοριών λόγω της συμφόρησης και της συμπίεσης: της μείωσης της διαστατικότητας του χώρου που είναι σε θέση να αναπαραστήσει και να γενικεύσει ολόκληρο το σύνολο των πληροφοριών. Σε άλλους κόσμους, δεδομένου ενός προβλήματος p πρέπει να βρούμε έναν μετασχηματισμό θ και έναν ακέραιο αριθμό m ώστε $\theta:n \rightarrow m$, $\theta(xn)=m$ όπου $n < m$ και $|p(xn)|p(m)$ ($|p$ είναι η πληροφορία που σχετίζεται με το πρόβλημα p). Διαισθητικά, η διατήρηση της πληροφορίας είναι πιο δύσκολη μειώνοντας το m .

Υπάρχουν πολλοί τύποι αυτοκωδικοποιητών και η χρήση τους ποικίλλει, αλλά ίσως οι πιο συνηθισμένοι είναι οι εξής

- ως μαθησιακό (ή αυτόματο) μοντέλο εξαγωγής χαρακτηριστικών
- στη φάση προ-εκπαίδευσης για τη βελτίωση της φάσης εκμάθησης για εργασίες ταξινόμησης ή παλινδρόμησης

Είναι το τελευταίο στο οποίο θέλουμε να εστιάσουμε σύντομα την προσοχή. Δεδομένου ότι η μάθηση χωρίς επίβλεψη αφαιρεί το εμπόδιο της ρητής ανθρώπινης επισήμανσης, κλιμακώνεται επίσης καλά με τις τρέχουσες τάσεις αύξησης των υπολογιστών και της διαθεσιμότητας των ακατέργαστων δεδομένων. Η δεξιά πλευρά της διαφάνειας δείχνει το μέγεθος της αύξησης των επιδόσεων με τη χρήση της μάθησης χωρίς επίβλεψη στην εκπαίδευση διαφόρων μοντέλων NLP: η προ-εκπαίδευση των μοντέλων σε ένα μεγάλο σώμα κειμένων βελτιώνει σημαντικά τις επιδόσεις τους σε απαιτητικές εργασίες επεξεργασίας φυσικής γλώσσας.

Για παράδειγμα, οι επιδόσεις σε εργασίες όπως η επιλογή της σωστής απάντησης σε μια ερώτηση πολλαπλής επιλογής αυξάνονται σταθερά όσο βελτιώνεται το υποκείμενο γλωσσικό μοντέλο. Παρόλο που η απόλυτη απόδοση αυτών των μεθόδων εξακολουθεί να είναι συχνά αρκετά χαμηλή σε σύγκριση με την επιβλεπόμενη τεχνολογία αιχμής, είναι ενθαρρυντικό ότι αυτή η συμπεριφορά είναι ισχυρή σε ένα ευρύ σύνολο εργασιών. Τα τυχαία αρχικοποιημένα δίκτυα που δεν περιέχουν καμία πληροφορία σχετικά με την εργασία και τον κόσμο δεν αποδίδουν καλύτερα από τα τυχαία με τη χρήση αυτών των ευρετικών μεθόδων. Αυτό παρέχει κάποια εικόνα για το γιατί η γενετική προ-εκπαίδευση μπορεί να βελτιώσει την απόδοση σε μεταγενέστερες εργασίες.

Στον κόσμο της Τεχνητής Νοημοσύνης, υπάρχει αυξανόμενο ενδιαφέρον για την κατανόηση της σχέσης μεταξύ των υπολογιστικών δαπανών για την εκπαίδευση των μοντέλων και της προκύπτουσας εξόδου. Η τεράστια ποσότητα υπολογιστικής ικανότητας που χρησιμοποιείται σήμερα στην ΤΝ παρήγαγε ένα νέο παράδειγμα: ο σχετικός αριθμός δεν είναι η ταχύτητα μιας μεμονωμένης GPU, ούτε η χωρητικότητα του μεγαλύτερου κέντρου δεδομένων, αλλά η ποσότητα

υπολογισμού που χρησιμοποιείται για την εκπαίδευση ενός μεμονωμένου μοντέλου - αυτός είναι ο αριθμός που είναι πιο πιθανό να συσχετιστεί με το πόσο ισχυρά είναι τα καλύτερα μοντέλα μας.

Εξετάζοντας τα δεδομένα στο σύνολό τους, βλέπουμε ξεκάθαρα δύο διακριτές εποχές εκπαίδευσης συστημάτων τεχνητής νοημοσύνης όσον αφορά τη χρήση υπολογιστών: (α) μια πρώτη εποχή, από το 1959 έως το 2012, η οποία ορίζεται από αποτελέσματα που παρακολουθούν κατά προσέγγιση τον νόμο του Moore, και (α) β) η σύγχρονη εποχή, από το 2012 έως σήμερα, των αποτελεσμάτων που χρησιμοποιούν υπολογιστική ισχύ που ξεπερνά σημαντικά τις μακροοικονομικές τάσεις. Σήμερα χρησιμοποιούμε για να μετρήσουμε την υπολογιστική ισχύ σε ημέρες petaflop (pfs-days).

Περιπτώσεις χρήσης

ΠΑΙΧΝΙΔΙΑ ΚΕΙΜΕΝΟΥ

Σε ένα παιχνίδι που βασίζεται σε κείμενο, δίνεται στον παίκτη (ή στο σύστημα, στην προκειμένη περίπτωση) μια συμβολοσειρά κειμένου που περιγράφει την τρέχουσα κατάσταση του παιχνιδιού και διάφορες συμβολοσειρές κειμένου που περιγράφουν πιθανές ενέργειες που θα μπορούσε να κάνει κάποιος. Μετά την επιλογή μιας από τις ενέργειες, η κατάσταση του περιβάλλοντος ενημερώνεται και αποκαλύπτεται σε μια νέα κειμενική περιγραφή. Μια ανταμοιβή δίνεται είτε σε κάθε μετάβαση είτε στο τέλος. Ο στόχος είναι να κατανοήσουμε, σε κάθε βήμα, το κείμενο της κατάστασης και όλα τα κείμενα ενεργειών για να επιλέξουμε την πιο κατάλληλη ενέργεια, πλοηγούμενοι στην ακολουθία των κειμένων έτσι ώστε να επιτύχουμε την υψηλότερη μακροπρόθεσμη ανταμοιβή.

Σε αυτή την περίπτωση, η έννοια της συνάφειας βασίζεται στην κοινή επίδραση κατάστασης/δράσης στην ανταμοιβή: μια ακολουθία κειμένου δράσης λέγεται ότι είναι "πιο σχετική" (για μια ακολουθία κειμένου κατάστασης) από τα άλλα κείμενα δράσης, εάν η ανάληψη αυτής της δράσης θα οδηγούσε σε υψηλότερη μακροπρόθεσμη ανταμοιβή.

Σημειώστε ότι όχι μόνο ο χώρος καταστάσεων S είναι διακριτός και τεράστιος, αλλά και ο χώρος ενεργειών A που περιγράφεται από συμβολοσειρές κειμένων φυσικής γλώσσας είναι διακριτός και δυνητικά απεριόριστος λόγω της εκθετικής πολυπλοκότητας της γλώσσας σε σχέση με το μήκος της πρότασης.

Τώρα παρουσιάζουμε το βαθύ δίκτυο ενίσχυσης συνάφειας (Deep Reinforcement Relevance Network - DRRN) για να προσπαθήσουμε να λύσουμε αυτό το πρόβλημα (βλέπε Deep Reinforcement Learning with a Natural Language Action Space, Ji He , Jianshu Chen , Xiaodong He , Jianfeng Gao , Lihong Li Li Deng and Mari Ostendorf, <https://arxiv.org/pdf/1511.04636v5.pdf>, για όλες τις λεπτομέρειες). Το DRRN χρησιμοποιεί ξεχωριστά βαθιά νευρωνικά δίκτυα για να αντιστοιχίσει τις συμβολοσειρές κειμένου κατάστασης και δράσης σε ξεχωριστά διανύσματα ενσωμάτωσης, από τα οποία η "συνάφεια" μετράται αριθμητικά με μια γενική συνάρτηση

αλληλεπίδρασης, όπως το εσωτερικό τους γινόμενο. Η έξοδος αυτής της συνάρτησης αλληλεπίδρασης ορίζει την τιμή της συνάρτησης Q για το τρέχον ζεύγος κατάστασης-δράσης, η οποία χαρακτηρίζει τη βέλτιστη μακροπρόθεσμη ανταμοιβή για τη σύζευξη αυτών των δύο κειμενικών συμβολοσειρών. Η προσέγγιση της συνάρτησης Q μαθαίνεται από άκρο σε άκρο με τη μέθοδο Q -learning.

Για την εκμάθηση του DRRN, εφαρμόζεται η στρατηγική "επανάληψη εμπειριών". Αυτός ο αλγόριθμος μάθησης χρησιμοποιεί μια σταθερή πολιτική εξερεύνησης για την αλληλεπίδραση με το περιβάλλον ώστε να ληφθεί ένα δείγμα τροχιάς. Στη συνέχεια, δειγματοληπτεί τυχαία μια πλειάδα μετάβασης ($sk, ak, rk, sk+1$), υπολογίζει το σφάλμα χρονικής διαφοράς για το δείγμα k και το χρησιμοποιεί για την ενημέρωση των παραμέτρων του μοντέλου σύμφωνα με την κάθοδο κλίσης.

Για να αποτρέψουμε τους αλγορίθμους από το να "θυμούνται" τη σειρά κατάστασης-δράσης και να κάνουν επιλογές με βάση τη διατύπωση των δράσεων, κάθε φορά που ο αλγόριθμος/παίκτης διαβάζει κείμενο από τον προσομοιωτή, ανακατεύουμε τυχαία τον κατάλογο των δράσεων. Αυτό θα ενθαρρύνει τους αλγορίθμους να λαμβάνουν αποφάσεις με βάση την κατανόηση των κειμένων που περιγράφουν τις καταστάσεις και τις δράσεις.

Η επιτυχία του DRRN στο χειρισμό ενός χώρου ενεργειών φυσικής γλώσσας A έγκειται στο γεγονός ότι τα κείμενα καταστάσεων και ενεργειών απεικονίζονται σε ξεχωριστούς χώρους ενσωμάτωσης πεπερασμένων διαστάσεων. Η από άκρη σε άκρη διαδικασία μάθησης καθιστά τα διανύσματα ενσωμάτωσης στους δύο χώρους περισσότερο ευθυγραμμισμένα για τα "καλά" (ή σχετικά) κείμενα δράσης σε σύγκριση με τις "κακές" (ή άσχετες) επιλογές, με αποτέλεσμα την υψηλότερη έξοδο της συνάρτησης αλληλεπίδρασης (τιμή συνάρτησης Q).

ΕΝΕΡΓΗ ΑΝΑΔΙΑΤΥΠΩΣΗ ΕΡΩΤΗΣΕΩΝ

Αυτό το μοντέλο είναι εμπνευσμένο από την ικανότητα των ανθρώπων να θέτουν τις σωστές ερωτήσεις: ένας πράκτορας που μαθαίνει να εκτελεί αυτή τη διαδικασία για τον χρήστη. Βρίσκεται μεταξύ του χρήστη και ενός backend συστήματος QA που αναφέρεται ως "περιβάλλον". Ο πράκτορας ονομάζεται AQA, καθώς υλοποιεί μια ενεργή στρατηγική απάντησης ερωτήσεων και ο στόχος είναι να μεγιστοποιήσει την πιθανότητα να λάβει τη σωστή απάντηση στέλνοντας μια αναδιατυπωμένη ερώτηση στο περιβάλλον. Ο πράκτορας προσπαθεί να βρει την καλύτερη απάντηση θέτοντας πολλές ερωτήσεις και συγκεντρώνοντας τα επιστρεφόμενα στοιχεία. Τα εσωτερικά στοιχεία του περιβάλλοντος δεν είναι διαθέσιμα στον πράκτορα, οπότε πρέπει να μάθει να διερευνά βέλτιστα ένα μαύρο κουτί χρησιμοποιώντας μόνο συμβολοσειρές ερωτήσεων.

Τα βασικά στοιχεία του μοντέλου που περιγράφονται είναι:

- Ένα προ-εκπαιδευμένο μοντέλο ακολουθίας σε ακολουθία που δέχεται ως είσοδο μια ερώτηση και επιστρέφει τις αναδιατυπώσεις της. Αυτή η εργασία είναι παρόμοια με τη

μηχανική μετάφραση, τη μετάφραση από τα αγγλικά στα αγγλικά, και πράγματι το αρχικό μοντέλο μπορεί να χρησιμοποιηθεί για γενική παράφραση, αλλά πρέπει να προσαρμοστεί για να υποστηρίξει την εκπαίδευση με ενισχυτική μάθηση χρησιμοποιώντας μεθόδους πολιτικής κλίσης.

- Ένα μοντέλο επιλογής απαντήσεων. Ο επιλογέας απαντήσεων χρησιμοποιεί ένα συνελκτικό νευρωνικό δίκτυο και αποδίδει μια βαθμολογία σε κάθε τριπλέτα αρχικής ερώτησης, αναδιατύπωσης και απάντησης. Ο επιλογέας χρησιμοποιεί προ-εκπαιδευμένες ενσωματώσεις λέξεων (π.χ. GloVe).
- Ένα σύστημα απάντησης ερωτήσεων (το περιβάλλον). Για τον σκοπό αυτό, χρησιμοποιούμε το BiDAF, ένα δημοφιλές σύστημα απάντησης ερωτήσεων.

Ένα επεισόδιο ξεκινά με μια αρχική ερώτηση q . Ο πράκτορας παράγει στη συνέχεια ένα σύνολο αναδιατυπώσεων $\{q^i\}_{i=1}^N$ έως N . Αυτές αποστέλλονται στο περιβάλλον το οποίο επιστρέφει απαντήσεις $\{a^i\}_{i=1}^N$ έως N . Το μοντέλο επιλογής επιλέγει στη συνέχεια την καλύτερη από αυτές τις υποψήφιες απαντήσεις.

Η μία πλευρά της συνομιλίας, ο πράκτορας AQA, προσπαθεί να προσαρμόσει τη γλώσσα του για να βελτιώσει την απάντηση από την άλλη πλευρά, το περιβάλλον QA. Πραγματοποιήθηκε μια ποιοτική ανάλυση της γλώσσας που παράγεται από τον πράκτορα AQA. Το αποτέλεσμα είναι ότι οι αναδιατυπώσεις των ερωτήσεων του πράκτορα αποκλίνουν σημαντικά από τις παραφράσεις της φυσικής γλώσσας. Είναι αξιοσημείωτο, ωστόσο, ότι ο πράκτορας είναι σε θέση να μάθει μη τετριμμένες και διαφανείς πολιτικές. Συγκεκριμένα, ο πράκτορας είναι σε θέση να ανακαλύψει κλασικές λειτουργίες ερωτημάτων ανάκτησης πληροφοριών, όπως η επαναστάθμιση όρων, που μοιάζει με την tf-idf (συντομογραφία για term frequency-inverse document frequency), και η μορφολογική απλοποίηση/αποσύνδεση (morphological simplification/stemming). Ένας πιθανός λόγος είναι ότι οι τρέχουσες εργασίες κατανόησης μηχανής περιλαμβάνουν την κατάταξη σύντομων αποσπασμάτων κειμένου, δίνοντας έτσι κίνητρα για τη συνάφεια, περισσότερο από τη βαθιά γλωσσική κατανόηση.

Μάθημα 4.3: Νέες προσεγγίσεις, εφαρμογές, ανοικτά προβλήματα

Θέματα προς συζήτηση

- Χαρακτήρας
- Βασισμένο σε Νευρωνικό Μοντέλο Γλώσσας
- Μετασχηματιστές, BERT και ELMo
- Εφαρμογές επεξεργασίας φυσικής γλώσσας
- Ανοικτά προβλήματα

Σημειώσεις διάλεξης

Σε αυτό το μάθημα θα δούμε μερικές από τις προσεγγίσεις που τα τελευταία χρόνια έχουν δώσει εξαιρετικά αποτελέσματα, σε πολλές περιπτώσεις ξεπερνώντας κατά πολύ τα αποτελέσματα που είχαν επιτευχθεί προηγουμένως, ακόμη και αν κάθε μια από τις τεχνικές που είδαμε (και που θα δούμε) μπορεί να είναι η καλύτερη σε μια συγκεκριμένη περίπτωση. Θα είναι σημαντικό να κατανοήσουμε τις ιδιαιτερότητες της καθεμιάς, προκειμένου να καταλάβουμε κατά περίπτωση ποια είναι η καλύτερη να χρησιμοποιηθεί ή πώς να συνδυάσουμε τα πλεονεκτήματα περισσότερων από μία.

Δεν θα είναι πάντα δυνατό να υπεισέλθουμε σε βαθιές λεπτομέρειες, όπως έγινε προηγουμένως, επειδή αυτό θα απαιτούσε πολύ περισσότερο χρόνο από αυτόν που μπορούμε να αφιερώσουμε σε αυτό το μάθημα, αλλά θα προσπαθήσουμε να δείξουμε τον πυρήνα αυτών των τεχνικών και θα δοθούν όλες οι αναφορές για περαιτέρω εμβάθυνση.

Στη συνέχεια, θα αναφερθούμε λίγο περισσότερο στις κύριες εφαρμογές του NLP και θα κλείσουμε με σημαντικούς προβληματισμούς σχετικά με τις επιπτώσεις της ανάπτυξης αυτών των τεχνολογιών.

Όταν η είσοδος ή η έξοδος είναι κείμενο, όπως έχουμε ήδη δει, έχουμε το αρχικό πρόβλημα της μετατροπής αυτού του κειμένου σε αριθμούς. Στο πρώτο μάθημα είδαμε έναν εξαιρετικό τρόπο για να το κάνουμε αυτό: την ενσωμάτωση λέξεων.

Μετά από αυτή την προσέγγιση, ωστόσο, αναπτύχθηκε και μια άλλη εκπληκτικά αποτελεσματική. Η προσέγγιση αυτή συνίσταται στην έναρξη με τους χαρακτήρες.

Αυτή η στρατηγική δεν φαίνεται πολύ ελπιδοφόρα, καθώς οι λέξεις που διαφέρουν μόνο κατά ένα χαρακτήρα μπορεί να έχουν εντελώς διαφορετικές έννοιες, με τα ίδια γράμματα μπορώ να σχηματίσω άπειρες διαφορετικές λέξεις που δεν έχουν καμία σχέση μεταξύ τους. Εν ολίγοις, οι προϋποθέσεις για τη χρήση αυτής της προσέγγισης δεν θα ήταν καλές. Πρέπει όμως να ληφθούν υπόψη και άλλες πτυχές της γλώσσας. Οι περισσότερες από τις λέξεις που χρησιμοποιούμε έχουν μια ρίζα (η οποία μπορεί να βρεθεί με το stemming) στην οποία προστίθενται σωματίδια που τροποποιούν ελαφρώς τη σημασία της συγκεκριμένης λέξης. Για παράδειγμα, παίρνω τη λέξη "limit" και έχει ένα νόημα, προσθέτω το "ed" και είναι επίθετο, το νόημα του οποίου μπορεί να γίνει το ακριβώς αντίθετό του με την πρόθεση "a". Παίρνω τη λέξη "free" προσθέτω "ly" και γίνεται το επίρρημα με τη σημασία που θα μπορούσαμε ήδη να περιμένουμε ξεκινώντας από αυτό το

ουσιαστικό. Και σε αυτή την περίπτωση, εισάγοντας το 'a', αντιστρέφω τη σημασία του. Με τα ρήματα προσθέτοντας το 'ed' μου δίνει τον αόριστο του (τουλάχιστον για τα τακτικά), προσθέτοντας το ing παίρνω το γερουνδιακό. Αυτές οι μορφολογικές διαφοροποιήσεις δεν είναι μεγάλες στα αγγλικά, αλλά μπορούμε ήδη να εκτιμήσουμε το φαινόμενο, το οποίο είναι πολύ πιο εμφανές σε άλλες γλώσσες, όπως τα φινλανδικά, τα ουγγρικά, τα τουρκικά και, με διαφορετικό τρόπο, τα κινέζικα.

Στην πραγματικότητα, η στρατηγική της εκκίνησης από τους χαρακτήρες είναι ιδιαίτερα κατάλληλη για τις μορφολογικά πλούσιες γλώσσες, δηλαδή εκείνες που έχουν πολλές συζυγίες, πτώσεις, κλίσεις που ξεκινούν από μια ρίζα.

Σε αυτές τις γλώσσες υπάρχει μεγαλύτερη αντιστοιχία μεταξύ της τυπικής (ή μορφολογικής) ομοιότητας και της λειτουργικής (ή νοηματικής) ομοιότητας.

Αν και πρέπει να είναι σαφές ότι γενικά η μία ομοιότητα δεν αποτελεί ούτε αναγκαία ούτε ικανή συνθήκη για την άλλη.

Μικρές διαφορές στη μορφή μπορούν να οδηγήσουν σε μεγάλες διαφορές στο νόημα και αντίστροφα μπορεί να υπάρχουν παρόμοιες έννοιες με εντελώς διαφορετικές μορφές. Σε πολλές περιπτώσεις, ωστόσο, η προσθήκη ορισμένων σωματιδίων οδηγεί σε προβλέψιμες αλλαγές στο νόημα, οι οποίες μπορούν επομένως να καταγραφούν από συγκεκριμένους τύπους δικτύων κατάλληλων για το σκοπό αυτό.

NLM ΜΕ ΒΑΣΗ ΤΟΝ ΧΑΡΑΚΤΗΡΑ

Πλεονεκτήματα

- Μικρότερο πρώτο στρώμα του νευρωνικού δικτύου,
- Εξαλείφεται το πρόβλημα των λέξεων που δεν υπάρχουν στο λεξιλόγιο,
- Δεν είναι απαραίτητο να διαιρεθεί σωστά σε λέξεις,
- Η καλύτερη επιλογή για μορφολογικά πολύ πλούσιες γλώσσες.

Μειονεκτήματα

- Πολύ αργά δίκτυα LSTM
- Παραιτούνται από τα προηγούμενα πλεονεκτήματα της ενσωμάτωσης λέξεων

Το μέγεθος του λεξιλογίου: για να προσπαθήσουμε να συλλάβουμε όλες τις λέξεις έχουμε πολύ μεγάλα λεξιλόγια που έχουν να κάνουν με το μέγεθος του πρώτου επιπέδου του δικτύου μας, ξεκινώντας από τους χαρακτήρες ξεπερνάμε αυτό το πρόβλημα, το πρώτο επίπεδο θα είναι πολύ πιο περιορισμένο.

Το πρόβλημα των λέξεων που δεν υπάρχουν στο λεξιλόγιο: με την ενσωμάτωση των λέξεων, αν μια λέξη δεν υπάρχει σε αυτή τη μορφή, παρόλο που το λεξιλόγιο είναι ίσως ένα ή δύο εκατομμύρια, δεν υπάρχει τίποτα να κάνουμε, αυτό το μέρος της πληροφορίας θα χαθεί, δεν υπάρχει τίποτα παρόμοιο που μπορεί να μπει στη θέση της, η λέξη απλά δεν υπάρχει. Ξεκινώντας από το επίπεδο του χαρακτήρα, από την άλλη πλευρά, μπορεί κανείς σχεδόν πάντα να χτίσει ξεκινώντας από τα βασικά στοιχεία, ιδίως σε περιπτώσεις όπου η λέξη δεν υπάρχει στο λεξιλόγιο επειδή είναι μια σπάνια μορφολογική παραλλαγή μιας λέξης που υπάρχει στο λεξιλόγιο. Αλλά ακόμη και στις περιπτώσεις όπου αυτή είναι εντελώς διαφορετική από τις άλλες, θα αναπαρίσταται και πάλι και θα μπορεί να συσχετιστεί με τις υπόλοιπες και έτσι να βοηθήσει στην αποτύπωση του νοήματος της πρότασης.

Δεν είναι απαραίτητο να χωριστεί σωστά σε λέξεις, κάτι που για ορισμένες γλώσσες (π.χ. Τσεχική) μπορεί να μην είναι εύκολο.

Εξαιρετικά αποτελέσματα επιτυγχάνονται, σε σύγκριση με άλλες τεχνικές, ακόμη και στην περίπτωση της μικτής γλώσσας, όπως φαίνεται στη σύγκριση [1] που πραγματοποιήθηκε λαμβάνοντας υπόψη διάφορες προηγούμενες εργασίες και συγκρίνοντάς τις με άλλες τεχνικές.

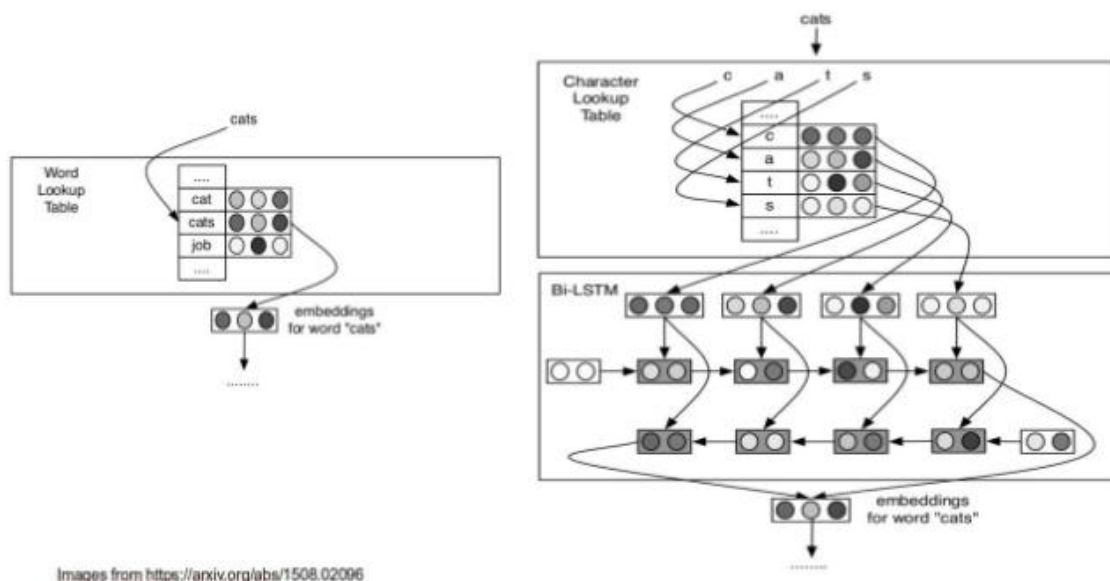
Το μειονέκτημα μπορεί να είναι η βραδύτητα στην εκμάθηση, ειδικά στις πρώτες προσπάθειες που έγιναν απαιτήθηκε πολύς χρόνος.

Ωστόσο, δεν πρέπει να εστιάζουμε υπερβολικά στα πλεονεκτήματα της μιας τεχνικής έναντι της άλλης, περισσότερο από οτιδήποτε άλλο είναι κατάλληλες για διαφορετικούς σκοπούς, αυτή είναι η καλύτερη επιλογή για τη συμπλήρωση λέξεων και για μεταφράσεις που περιλαμβάνουν μια πλούσια μορφολογικά ομιλούσα γλώσσα ή ως γλώσσα προέλευσης ή ως γλώσσα-στόχο. Επιπλέον, μπορεί επίσης να είναι η καλύτερη επιλογή για τις περισσότερες εργασίες, εάν αυτές πρέπει να πραγματοποιηθούν σε μια γλώσσα στην οποία είναι δύσκολο να διακρίνει κανείς τις λέξεις, καθώς αυτές αλλάζουν και συνδυάζονται για να σχηματίζουν πάντα διαφορετικές ανάλογα με την πρόταση.

Πάνω απ' όλα, θα είναι μια ακόμη τεχνική που πρέπει να ληφθεί υπόψη και να επιλεγεί αξιολογώντας όλες τις πτυχές του συγκεκριμένου προβλήματος που πρέπει να αντιμετωπιστεί.

Σχεδόν ταυτόχρονα, ξεκινώντας από το 2015, αναπτύχθηκαν διάφορες ερευνητικές ομάδες που δοκίμασαν αυτό το μονοπάτι χρησιμοποιώντας διαφορετικά δίκτυα.

Ας εξετάσουμε μία από τις πρώτες και πιο αναφερόμενες: τη μελέτη [2].



Για να μοντελοποιήσουν τις πολύπλοκες σχέσεις μεταξύ μορφής και λειτουργίας, οι συγγραφείς βασίζονται σε LSTM που μπορούν να συλλάβουν πολύπλοκες μη γραμμικές και μη τοπικές δυναμικές σε ακολουθίες.

Χρησιμοποιούν αμφίδρομους LSTM (ώστε να λαμβάνονται υπόψη τόσο οι επόμενοι όσο και οι προηγούμενοι χαρακτήρες) για να διαβάσουν τους χαρακτήρες που αποτελούν κάθε λέξη και να τους συνδυάζουν στο τέλος σε μια διανυσματική αναπαράσταση της λέξης.

Αυτό το μοντέλο υποθέτει ότι κάθε χαρακτήρας συνδέεται με ένα διάνυσμα και οι παράμετροι ενός LSTM κωδικοποιούν τόσο για ιδιοσυγκρασιακές λεξικές διαφορές όσο και για μορφολογικά κανονικές λεξικές διαφορές.

Το αποτέλεσμα είναι ότι με αυτούς τους συνδυασμούς είναι δυνατόν να μάθουμε τους κανόνες που διέπουν τις σχέσεις μεταξύ μορφολογικών και νοηματικών διαφορών. Έτσι, στην ενσωμάτωση των λέξεων το καθένα είναι ανεξάρτητο από το άλλο και ακόμη και αν διαπιστώσουμε ότι οι πληθυντικοί μπορούν να τοποθετηθούν στο διανυσματικό χώρο με έναν ορισμένο τρόπο σε σχέση με τους ενικούς, το κάθε ένα διάνυσμα έχει μάθει ανεξάρτητα από το άλλο, ενώ με την ενσωμάτωση των χαρακτήρων το δίκτυο μαθαίνει ότι η προσθήκη του *s* (στα αγγλικά) μετατρέπει τον ενικό σε πληθυντικό. Παρομοίως, οι καταλήξεις που μετασχηματίζουν ένα στέλεχος με συγκεκριμένο τρόπο, όπως ένα επίρρημα, μπορούν να κάνουν μια αόρατη λέξη να αντιμετωπιστεί σωστά, αφού το δίκτυο γνωρίζει τη ρίζα και το αποτέλεσμα αυτής της κατάληξης. Αυτό επιτρέπει να μειωθεί δραστικά ο αριθμός των παραμέτρων, αφού για την ενσωμάτωση των λέξεων μια από τις διαστάσεις δίνεται από το λεξικό, αλλά αν συμπεριλάβουμε στο λεξικό όλες τις κλίσεις των λέξεων, φτάνουμε εύκολα σε πολλά εκατομμύρια λέξεις ακόμη και για μια φτωχή μορφολογικά γλώσσα όπως η αγγλική. Μια λύση είναι η δημιουργία μιας διανυσματικής αναπαράστασης με βάση τα μορφώματα, τις ελάχιστες μονάδες σημασίας, για τις οποίες οι λέξεις είναι το αποτέλεσμα της σύνθεσης αυτών των βασικών στοιχείων. Η δυσκολία σε αυτό το μονοπάτι είναι το γεγονός ότι πρέπει να εξαρτηθείτε από έναν μορφολογικό αναλυτή, είναι περίπλοκο να τον φτιάξετε, χρειάζεστε πρόσθετες γλωσσικές γνώσεις για να το κάνετε, είναι πρακτικά αδύνατο να προβλέψετε με αυτόν τον τρόπο όλες τις δυνατότητες και τις εξαιρέσεις, συμπεριλαμβανομένων των περιπτώσεων στις οποίες η λέξη μορφολογικά φαίνεται να είναι μια παραλλαγή που έχει μια συγκεκριμένη κατάληξη, ενώ δεν είναι έτσι. Μια απλούστερη λύση, ακόμη και υπολογιστικά, είναι να ξεκινήσετε με χαρακτήρες.

Στο σχήμα, που προέρχεται από την εργασία τους, μπορούμε να δούμε πώς λειτουργεί το μοντέλο μετατροπής χαρακτήρων σε λέξεις (C2W), σε σύγκριση με την κλασική ενσωμάτωση λέξεων.

Η είσοδος είναι μια λέξη, που αναπαρίσταται στο δίκτυο από τους μεμονωμένους χαρακτήρες που την αποτελούν, αυτό που θέλετε να πάρετε στο τέλος είναι ένα ενιαίο διάνυσμα που την αναπαριστά.

Οι χαρακτήρες, τα σημεία στίξης, τα ψηφία και κάθε άλλο στοιχείο που θέλετε να συμπεριλάβετε αναπαρίστανται ως διανύσματα ενός σημείου. Το επόμενο επίπεδο είναι το αμφίδρομο LSTM, η έξοδος είναι ένα διάνυσμα με τις ίδιες τιμές με αυτή τη λέξη σε μια ενσωμάτωση.

Έτσι ώστε το δίκτυο, αφού εκπαιδευτεί, να μπορεί να χρησιμοποιηθεί το ίδιο ως ενσωμάτωση και να τοποθετηθεί στην αρχή ενός άλλου δικτύου.

Ας δούμε αναλυτικότερα πώς λειτουργεί η εκπαίδευση.

$x_1, \dots, x_m$ τα διανύσματα των χαρακτήρων
 $h_1, \dots, h_{m+1}$ η ακολουθία καταστάσεων
 σ η λογιστική σισμοειδής συνάρτηση
 $\odot$ γινόμενο Hadamard
 C_t έξτρα μνήμη κυττάρων

$$\begin{aligned} i_t &= \sigma(W_{ix}x_t + W_{ih}h_{t-1} + W_{ic}c_{t-1} + b_i) \\ f_t &= \sigma(W_{fx}x_t + W_{fh}h_{t-1} + W_{fc}c_{t-1} + b_f) \\ c_t &= f_t \odot c_{t-1} + i_t \odot \tanh(W_{cx}x_t + W_{ch}h_{t-1} + b_c) \\ o_t &= \sigma(W_{ox}x_t + W_{oh}h_{t-1} + W_{oc}c_t + b_o) \\ h_t &= o_t \odot \tanh(c_t), \end{aligned}$$

Η πληροφορία διαδίδεται από το c_{t-1} στο c_t και ελέγχεται από τις τρεις πύλες i_t , f_t και o_t . Αυτές οι πύλες καθορίζουν τι θα συμπεριληφθεί από την είσοδο, τι θα ξεχαστεί από την c_{t-1} και τι είναι σχετικό με την τρέχουσα κατάσταση h_t .

Δεδομένου ότι το στρώμα LSTM είναι αμφίδρομο, έχουμε μια ακολουθία καταστάσεων που λαμβάνονται προς τα εμπρός και μία που λαμβάνεται στην αντίθετη κατεύθυνση, επίσης τα βάρη των συνδέσεων είναι διαφορετικά στις δύο κατευθύνσεις, έτσι έχουμε ένα σύνολο βαρών για προς τα εμπρός και ένα για προς τα πίσω. Στο τέλος, οι δύο ακολουθίες καταστάσεων σε συνδυασμό με τις κατάλληλες παραμέτρους καθορίζουν την αναπαράσταση της λέξης.

Τα μοντέλα που βασίζονται σε χαρακτήρες χρησιμοποιούνται στις περισσότερες εφαρμογές NLP, ειδικότερα:

Μεταφράσεις

Αρκετές μελέτες έχουν δείξει την ίδια ή καλύτερη απόδοση με την τρέχουσα κατάσταση, ιδίως όταν εμπλέκονται επίσης μορφολογικά πλούσιες ή μικτές γλώσσες.

Συμπλήρωση λέξεων και διόρθωση φυσικής γλώσσας

Είναι ο πιο πρόσφορος τομέας για την τεχνική αυτή- αρκετές μελέτες αποδεικνύουν την αποτελεσματικότητα των γλωσσικών μοντέλων που λαμβάνονται με αυτόν τον τρόπο στη συμπλήρωση λέξεων και στη διόρθωση των πληκτρολογούμενων.

Αναγνώριση ονομαστικών οντοτήτων

Η τεχνική αυτή έχει επίσης χρησιμοποιηθεί με επιτυχία στον τομέα αυτό, ιδίως στον ιατρικό τομέα.

Ταξινόμηση και ομαδοποίηση

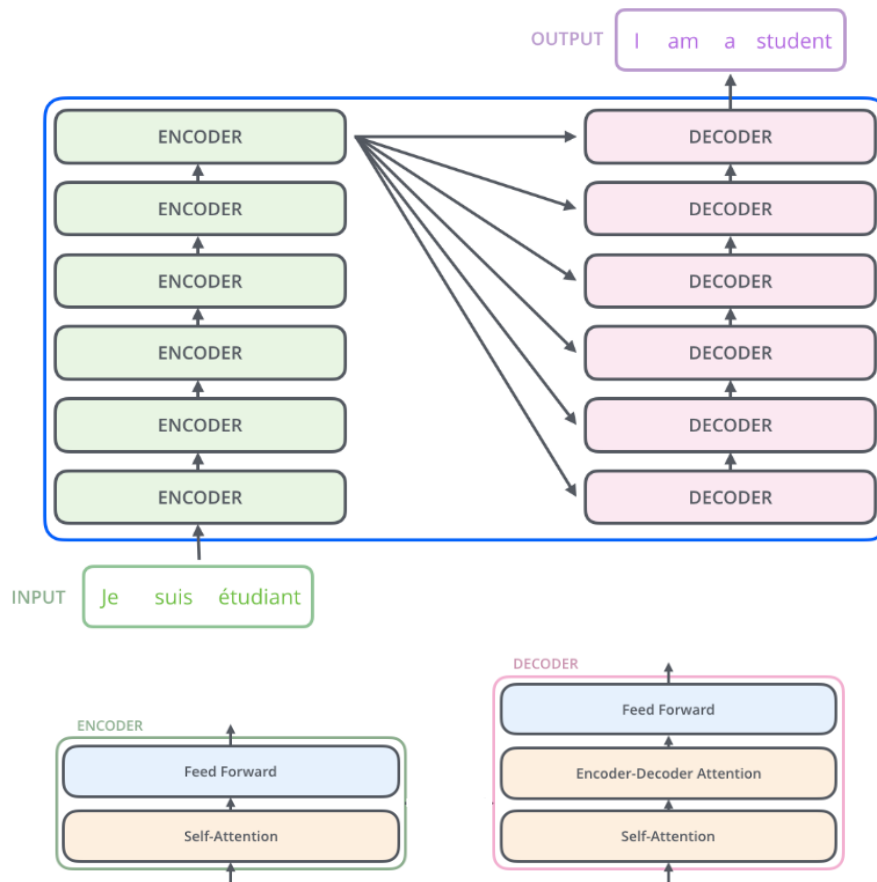
Ιδιαίτερα σε περιοχές με πολλούς μη κανονικούς όρους και μεικτές γλώσσες (π.χ. twitter).

Γεννήτριες κειμένου

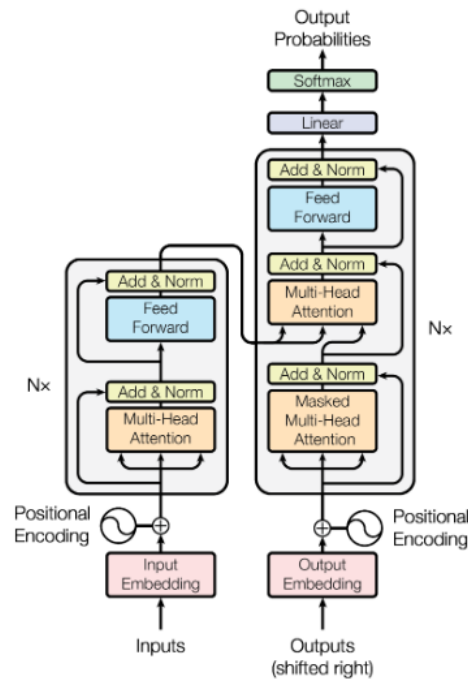
Για διαφορετικούς τύπους κειμένων, επίσης για ομοιοκατάληκτους στίχους

Αναγνώριση ομιλίας

Τα μοντέλα που βασίζονται σε χαρακτήρες είναι ιδιαίτερα καλά στην πρόβλεψη νέων μορφών λέξεων που δεν έχουν εμφανιστεί στα δεδομένα εκπαίδευσης.



[High level model description](#)



[Attention Is All You Need](#)

Σε μια προηγούμενη διαφάνεια (4.2 Μάθηση χωρίς επίβλεψη για NLP) δείξαμε πώς είναι υπολογιστικά βαριά η αντιμετώπιση της σύγχρονης εκπαίδευσης μοντέλων NLP (και βαθιάς μάθησης γενικότερα).

Ας εστιάσουμε τώρα την προσοχή μας στο πρόβλημα της μηχανικής μετάφρασης κειμένου.

Όπως είδαμε στην προηγούμενη ενότητα, η Προσοχή είναι μια έννοια που βοηθά στη βελτίωση της απόδοσης των αναδρομικών βαθιών νευρωνικών δικτύων με μοντέλα κωδικοποιητή-αποκωδικοποιητή όσον αφορά την "ποιότητα της μετάφρασης". Η εκπαίδευση ενός βαθύ νευρωνικού δικτύου είναι ίσως υπολογιστικά βαριά, γι' αυτό και εισήχθη ο Μετασχηματιστής. Το μεγαλύτερο όφελος προκύπτει από τον τρόπο με τον οποίο ο Transformer προσφέρεται για παραλληλισμό. Στην πραγματικότητα, είναι σύσταση της Google Cloud να χρησιμοποιήσετε το The Transformer ως μοντέλο αναφοράς για να χρησιμοποιήσετε την προσφορά Cloud TPU της.

Το The Transformer προτάθηκε στην εργασία "Attention is All You Need" (ανατρέξτε στα [4] και [5] για λεπτομερή περιγραφή του μοντέλου). Στην αριστερή πλευρά της διαφάνειας απεικονίζεται το διάγραμμα υψηλού επιπέδου αυτού του μοντέλου. Πρόκειται ουσιαστικά για έναν κωδικοποιητή-αποκωδικοποιητή που χρησιμοποιεί ως είσοδο την ενσωμάτωση λέξεων και ως έξοδο τις πιθανότητες λέξεων ενός λεξιλογίου.

Κωδικοποιητής: Ο κωδικοποιητής αποτελείται από μια στοίβα $N = 6$ πανομοιότυπων στρωμάτων. Κάθε στρώμα έχει δύο υποστρώματα. Το πρώτο είναι ένας μηχανισμός αυτοπροστασίας πολλαπλών κεφαλών, και το δεύτερο είναι ένα απλό, πλήρως συνδεδεμένο δίκτυο τροφοδότησης προς τα εμπρός με γνώμονα τη θέση.

Αποκωδικοποιητής: Ο αποκωδικοποιητής αποτελείται επίσης από μια στοίβα $N = 6$ πανομοιότυπων στρωμάτων. Εκτός από τα δύο υποστρώματα σε κάθε στρώμα κωδικοποιητή, ο αποκωδικοποιητής εισάγει στη μέση ένα τρίτο υποστρώμα, το οποίο εκτελεί προσοχή πολλαπλών κεφαλών στην έξοδο της στοίβας κωδικοποιητών.

Έτσι, τόσο ο κωδικοποιητής όσο και ο αποκωδικοποιητής έχουν στρωματοποιημένη δομή και κοιτάζοντας μέσα σε κάθε στρώμα βρίσκουμε άλλα στρώματα. Το πεδίο εφαρμογής του στρώματος υψηλού επιπέδου και κάθε εσωτερικού στρώματος είναι να "βρει" μια κατάλληλη αναπαράσταση ενσωμάτωσης κάθε λέξης: η είσοδος ενσωμάτωσης κάθε στρώματος είναι κατάλληλα παραμορφωμένη, τυλιγμένη και στριμμένη από το ίδιο το στρώμα. Για να καταλάβουμε γιατί το χρειαζόμαστε αυτό, ας εμβαθύνουμε τώρα εν συντομία στην "αυτοπροσοχή" (δομικό στοιχείο της περιγραφής υψηλού επιπέδου) και στην "προσοχή πολλαπλών κεφαλών" (δομικό στοιχείο της λεπτομερέστερης περιγραφής του μοντέλου στα δεξιά της διαφάνειας):

Αυτο-προσοχή (Self-attention)

ας πούμε ότι η ακόλουθη πρόταση είναι μια πρόταση εισόδου που θέλουμε να μεταφράσουμε: "Το ζώο δεν διέσχισε το δρόμο γιατί ήταν πολύ κουρασμένο." Σε τι αναφέρεται το "αυτό" σε αυτή την πρόταση; Αναφέρεται στο δρόμο ή στο ζώο; Είναι μια απλή ερώτηση για έναν άνθρωπο, αλλά όχι τόσο απλή για έναν αλγόριθμο. Όταν το μοντέλο επεξεργάζεται τη λέξη "αυτό", η αυτοπροσοχή του επιτρέπει να συνδέσει το "αυτό" με το "ζώο". Καθώς το μοντέλο επεξεργάζεται κάθε λέξη (κάθε θέση στην ακολουθία εισόδου), η αυτοπροσοχή του επιτρέπει να εξετάζει άλλες θέσεις στην ακολουθία εισόδου για ενδείξεις που μπορούν να το βοηθήσουν να οδηγηθεί σε μια καλύτερη κωδικοποίηση για αυτή τη λέξη.

Η αυτο-προσοχή είναι η μέθοδος που χρησιμοποιεί ο Μετασχηματιστής για να ενσωματώσει την "κατανόηση" άλλων σχετικών λέξεων σε εκείνη που επεξεργαζόμαστε αυτή τη στιγμή.

Τεχνικά αυτό γίνεται με μια συνάρτηση προσοχής που μπορεί να περιγραφεί ως αντιστοίχιση ενός ερωτήματος και ενός συνόλου ζευγών κλειδιών-τιμών σε μια έξοδο, όπου το ερώτημα, τα κλειδιά, οι τιμές και η έξοδος είναι όλα διανύσματα. Η έξοδος υπολογίζεται ως σταθμισμένο άθροισμα των τιμών, όπου το βάρος που αποδίδεται σε κάθε τιμή υπολογίζεται από μια συνάρτηση συμβατότητας του ερωτήματος με το αντίστοιχο κλειδί

Πολλαπλές κεφαλές (Multi-head)

Αντί να εκτελεί μια ενιαία συνάρτηση προσοχής με $d_{\text{μοντέλο}}$ -διαστάσεων κλειδιά, τιμές και ερωτήματα, το μοντέλο θεώρησε επωφελές να προβάλλει γραμμικά τα ερωτήματα, τα κλειδιά και τις τιμές h φορές με διαφορετικές, μαθημένες γραμμικές προβολές σε d_k , d_v και d_n διαστάσεις, αντίστοιχα. Σε κάθε μία από αυτές τις προβαλλόμενες εκδοχές των ερωτημάτων, των κλειδιών και των τιμών **εκτελούμε στη συνέχεια τη συνάρτηση προσοχής παράλληλα**, αποδίδοντας d_n -διάστατες τιμές εξόδου. Αυτές συνενώνονται και προβάλλονται και πάλι, καταλήγοντας στις τελικές τιμές.

Ο μετασχηματιστής χρησιμοποιεί την προσοχή πολλαπλών κεφαλών με τρεις διαφορετικούς τρόπους:

- Τα ερωτήματα προέρχονται από το προηγούμενο στρώμα αποκωδικοποιητή και τα κλειδιά μνήμης και οι τιμές προέρχονται από την έξοδο του κωδικοποιητή. Αυτό επιτρέπει σε κάθε θέση του αποκωδικοποιητή να παρακολουθεί όλες τις θέσεις της ακολουθίας εισόδου. Αυτό μιμείται τους τυπικούς μηχανισμούς προσοχής κωδικοποιητή-αποκωδικοποιητή στα μοντέλα ακολουθίας-αλληλουχίας.
- Ο κωδικοποιητής περιέχει στρώματα αυτοπροσοχής. Σε ένα στρώμα αυτοπροσοχής όλα τα κλειδιά, οι τιμές και τα ερωτήματα προέρχονται από την ίδια θέση, στην περίπτωση αυτή, την έξοδο του προηγούμενου στρώματος στον κωδικοποιητή. Κάθε θέση στον κωδικοποιητή μπορεί να παρακολουθεί όλες τις θέσεις στο προηγούμενο στρώμα του κωδικοποιητή.
- Ομοίως, τα στρώματα αυτοπροσοχής στον αποκωδικοποιητή επιτρέπουν σε κάθε θέση στον αποκωδικοποιητή να παρακολουθεί όλες τις θέσεις στον αποκωδικοποιητή μέχρι και τη συγκεκριμένη θέση.

Συνιστάται ιδιαίτερα να ανατρέξετε στις παραπομπές για περισσότερες λεπτομέρειες, καθώς το προηγούμενο είναι μόνο μια πρότυπη επισκόπηση.

Σχετικά με τα αποτελέσματα, στην εργασία μετάφρασης από τα αγγλικά στα γερμανικά WMT 2014, το μοντέλο big transformer ξεπερνά τα καλύτερα μοντέλα που είχαν αναφερθεί προηγουμένως (συμπεριλαμβανομένων των συνόλων) κατά περισσότερο από 2,0 BLEU (το BLEU -bilingual evaluation understudy- είναι ένας αλγόριθμος για την αξιολόγηση της ποιότητας του κειμένου που έχει μεταφραστεί μηχανικά από μια φυσική γλώσσα σε μια άλλη), καθιερώνοντας μια νέα κορυφαία βαθμολογία BLEU 28,4. Η εκπαίδευση διήρκεσε 3,5 ημέρες σε 8 GPU P100. Ακόμα και το βασικό μας μοντέλο ξεπερνά όλα τα μοντέλα και τα σύνολα που είχαν δημοσιευτεί προηγουμένως, με ένα κλάσμα του κόστους εκπαίδευσης οποιουδήποτε από τα ανταγωνιστικά μοντέλα.

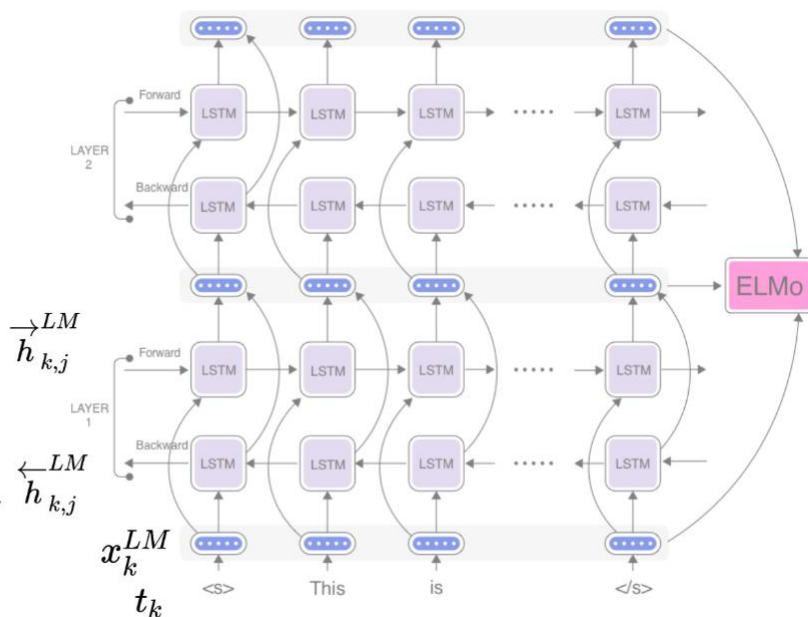
Structure

Each token t_k

L-layer biLM
computes $2L+1$
representations

k is the k -th token

j is the j -th biLM layer



[εικόνα](#)

Source		Nearest Neighbors
GloVe	play	playing, game, games, played, players, plays, player, Play, football, multiplayer
biLM	Chico Ruiz made a spectacular <u>play</u> on Alusik 's grounder {...}	Kieffer , the only junior in the group , was commended for his ability to hit in the clutch , as well as his all-round excellent <u>play</u> .
	Olivia De Havilland signed to do a Broadway <u>play</u> for Garson {...}	{...} they were actors who had been handed fat roles in a successful <u>play</u> , and had talent enough to fill the roles competently , with nice understatement .

Table 4: Nearest neighbors to “play” using GloVe and the context embeddings from a biLM.

Πίνακας

Όπως είπαμε σε όλες σχεδόν τις διαφάνειες NLP, μετά την προεπεξεργασία του κειμένου εισόδου (αν είναι απαραίτητο), το πρώτο βήμα είναι η μετατροπή της πρότασης σε διανύσματα: συνήθως μια λέξη αντιστοιχεί σε ένα διάνυσμα n -διαστάσεων, όπου το n είναι μια σταθερή υπερπαράμετρος. Αυτό το βήμα ονομάζεται ενσωμάτωση.

Η εκμάθηση αναπαράστασεων υψηλής ποιότητας για κάθε λέξη ενός λεξιλογίου μπορεί να αποτελέσει πρόκληση. Θα πρέπει ιδανικά να μοντελοποιούν και τα δύο:

1. πολύπλοκα χαρακτηριστικά της χρήσης των λέξεων (π.χ. σύνταξη και σημασιολογία),
2. τον τρόπο με τον οποίο αυτές οι χρήσεις ποικίλλουν σε διάφορα γλωσσικά περιβάλλοντα (δηλ. να μοντελοποιούν την πολυσημία).

Σε αντίθεση με τις παραδοσιακές ενσωματώσεις λέξεων, όπως οι word2vec και GloVe, το διάνυσμα ELMo που αποδίδεται σε ένα σύμβολο ή μια λέξη είναι στην πραγματικότητα μια συνάρτηση ολόκληρης της πρότασης που περιέχει αυτή τη λέξη. Ως εκ τούτου, η ίδια λέξη μπορεί να έχει διαφορετικά διανύσματα λέξης σε διαφορετικά συμφραζόμενα. Πρόκειται για έναν νέο τύπο βαθιάς αναπαράστασης λέξεων με βάση τα συμφραζόμενα που αντιμετωπίζει άμεσα και τις δύο προκλήσεις και μπορεί εύκολα να ενσωματωθεί σε υπάρχοντα μοντέλα.

Τα διανύσματα λέξεων ELMo υπολογίζονται πάνω σε ένα αμφίδρομο γλωσσικό μοντέλο δύο επιπέδων (biLM). Αυτό το μοντέλο biLM έχει δύο στρώματα στοιβαγμένα μεταξύ τους. Κάθε στρώμα έχει 2 πέρασματα - προς τα εμπρός και προς τα πίσω. Ακολουθεί μια περιγραφή των όσων απεικονίζονται στη διαφάνεια:

- η αρχιτεκτονική χρησιμοποιεί ένα νευρωνικό δίκτυο συνελκτικού επιπέδου (CNN) για την αναπαράσταση των λέξεων μιας συμβολοσειράς κειμένου σε ακατέργαστα διανύσματα λέξεων
- αυτά τα ακατέργαστα διανύσματα λέξεων λειτουργούν ως είσοδοι στο πρώτο επίπεδο του biLM
- το εμπρόσθιο πέρασμα περιέχει πληροφορίες σχετικά με μια συγκεκριμένη λέξη και το πλαίσιο (άλλες λέξεις) πριν από αυτή τη λέξη
- το αντίστροφο πέρασμα περιέχει πληροφορίες σχετικά με τη λέξη και το πλαίσιο μετά από αυτή
- αυτό το ζεύγος πληροφοριών, από το εμπρόσθιο και το οπίσθιο πέρασμα, σχηματίζει τα ενδιάμεσα διανύσματα λέξεων
- αυτά τα ενδιάμεσα διανύσματα λέξεων τροφοδοτούνται στο επόμενο επίπεδο του biLM
- η τελική αναπαράσταση (ELMo) είναι το σταθμισμένο άθροισμα των ακατέργαστων διανυσμάτων λέξεων και των 2 ενδιάμεσων διανυσμάτων λέξεων

Το ELMo εισήχθη το 2018 και σηματοδότησε μια πραγματική επανάσταση στο NLP: σε κάθε μία από τις έξι εξεταζόμενες εργασίες αναφοράς, η απλή προσθήκη του ELMo καθιερώνει ένα νέο κορυφαίο αποτέλεσμα, με σχετικές μειώσεις σφαλμάτων που κυμαίνονται από 6 έως 20% σε σχέση με τα ισχυρά μοντέλα βάσης.

Ερώτηση – Απάντηση

Μετά την προσθήκη του ELMo στο βασικό μοντέλο, το σύνολο δοκιμών F1 βελτιώθηκε κατά 4,7% από 81,1% σε 85,8%, δηλαδή κατά 24,9% μείωση του σχετικού σφάλματος σε σχέση με το βασικό μοντέλο, και βελτίωση του συνολικού state-of-the-art του μεμονωμένου μοντέλου κατά 1,4%. Ένα σύνολο 11 μελών ωθεί το F1 στο 87,4, το συνολικό state-of-the-art κατά τη στιγμή του πειράματος.

Συμπερασματολογία κειμένου

Η συνεπαγωγή κειμένου είναι το έργο του προσδιορισμού του κατά πόσον μια "υπόθεση" είναι αληθής, δεδομένης μιας "προϋπόθεσης". Συνολικά, η προσθήκη του ELMo στο μοντέλο ESIM βελτιώνει την ακρίβεια κατά μέσο όρο 0,7% σε πέντε τυχαίες κατατάξεις. Ένα πενταμελές σύνολο ωθεί τη συνολική ακρίβεια στο 89,3%, ξεπερνώντας το προηγούμενο καλύτερο σύνολο 88,9%

Σημασιολογική επισήμανση ρόλων

Ένα σύστημα σημασιολογικής επισήμανσης ρόλων (SRL) μοντελοποιεί τη δομή κατηγορήματος-αρχείου μιας πρότασης και συχνά περιγράφεται ως απάντηση στο ερώτημα "Ποιος έκανε τι σε ποιον". Όταν προστέθηκε το ELMo, το σύνολο δοκιμών F1 του μεμονωμένου μοντέλου αυξήθηκε κατά 3,2% από 81,4% σε 84,6%, βελτιώνοντας ακόμη και το προηγούμενο καλύτερο αποτέλεσμα του συνόλου κατά 1,2%.

Ανάλυση συμπερασμάτων

Η ανάλυση συμπερασμάτων είναι το έργο της ομαδοποίησης των αναφορών σε κείμενο που αναφέρονται στις ίδιες υποκείμενες οντότητες του πραγματικού κόσμου. Η βασική γραμμή χρησιμοποιεί ένα μηχανισμό biLSTM και προσοχής για να υπολογίσει πρώτα τις αναπαραστάσεις span και στη συνέχεια εφαρμόζει ένα μοντέλο κατάταξης αναφορών softmax για την εύρεση αλυσίδων πυρηνοαναφορών. Η προσθήκη του ELMo βελτίωσε τον μέσο όρο F1 κατά 3,2% από 67,2 σε 70,4, καθιερώνοντας μια νέα κατάσταση της τεχνολογίας, βελτιώνοντας και πάλι το προηγούμενο καλύτερο αποτέλεσμα του συνόλου κατά 1,6% F1.

Εξαγωγή ονομαστικών οντοτήτων

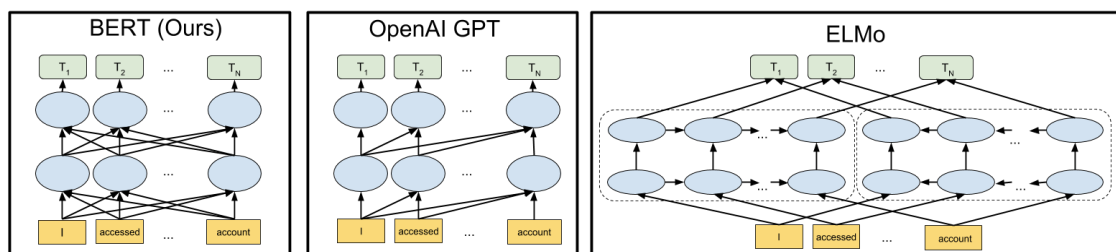
Η εργασία CoNLL 2003 NER αποτελείται από ειδησεογραφικά δελτία από το σώμα του Reuters RCV1 με τέσσερις διαφορετικούς τύπους οντοτήτων (PER, LOC, ORG, MISC). Το

βασικό μοντέλο χρησιμοποιεί προ-εκπαιδευμένες ενσωματώσεις λέξεων, μια αναπαράσταση CNN με βάση τους χαρακτήρες, δύο στρώματα biLSTM και μια απώλεια τυχαίου πεδίου υπό συνθήκη (CRF), παρόμοια με το Collobert. Το ενισχυμένο biLSTM-CRF του ELMo επιτυγχάνει 92,22% F1 κατά μέσο όρο σε πέντε εκτελέσεις. Η βασική διαφορά μεταξύ αυτού του συστήματος και του προηγούμενου state of the art είναι ότι επιτρέπει στο μοντέλο εργασίας να μαθαίνει έναν σταθμισμένο μέσο όρο όλων των στρωμάτων biLM, ενώ το προηγούμενο χρησιμοποιεί μόνο το κορυφαίο στρώμα biLM.

Ανάλυση συναισθήματος

Η λεπτή εργασία ταξινόμησης συναισθήματος στην τράπεζα συναισθημάτων Stanford περιλαμβάνει την επιλογή μιας από τις πέντε ετικέτες (από πολύ αρνητική έως πολύ θετική) για την περιγραφή μιας πρότασης από μια κριτική ταινίας. Οι προτάσεις περιέχουν ποικίλα γλωσσικά φαινόμενα, όπως ιδιωτισμούς και σύνθετες συντακτικές κατασκευές, όπως αρνήσεις, που είναι δύσκολο να μάθουν τα μοντέλα. Το βασικό μας μοντέλο είναι το δίκτυο ταξινόμησης bi-attentive (BCN) από το McCann, το οποίο είχε επίσης το προηγούμενο κορυφαίο αποτέλεσμα όταν συμπληρώθηκε με ενσωματώσεις CoVe. Η αντικατάσταση του CoVe με ELMo στο μοντέλο BCN οδηγεί σε βελτίωση της απόλυτης ακρίβειας κατά 1,0% σε σχέση με την τρέχουσα κατάσταση της τεχνολογίας.

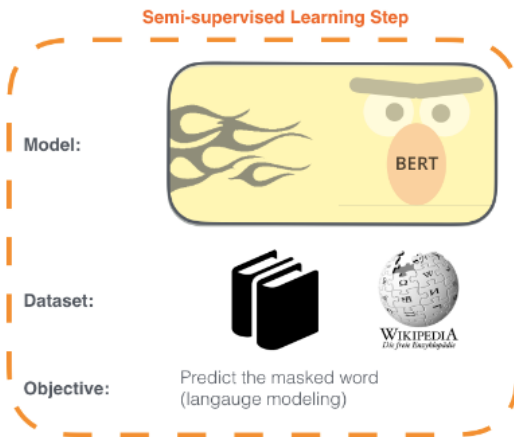
Δεδομένου ότι η προσθήκη ELMo βελτιώνει την απόδοση της εργασίας σε σχέση με τα διανύσματα λέξεων, οι αναπαραστάσεις πλαισίου biLM θα πρέπει να κωδικοποιούν πληροφορίες γενικά χρήσιμες για εργασίες NLP που δεν αποτυπώνονται στα διανύσματα λέξεων. Διαισθητικά, η biLM θα πρέπει να αποσαφηνίζει το νόημα των λέξεων χρησιμοποιώντας το περιεχόμενό τους. Σκεφτείτε τη λέξη "παιχνίδι", μια λέξη με μεγάλη πολυσημία. Ο πίνακας στη δεξιά πλευρά παραθέτει τους πλησιέστερους γείτονες της λέξης "play" χρησιμοποιώντας διανύσματα GloVe. Είναι διάσπαρτοι σε διάφορα μέρη του λόγου (π.χ. "έπαιξε", "παίζει" ως ρήματα και "παίκτης", "παίζω" ως ουσιαστικά), αλλά εστιάζουν στις αθλητικές έννοιες της λέξης "play". Αντίθετα, οι δύο κάτω σειρές παρουσιάζουν προτάσεις πλησιέστερων γειτόνων από το σύνολο δεδομένων SemCor ([visit:http://web.eecs.umich.edu/~mihalcea/downloads.html#semcor](http://web.eecs.umich.edu/~mihalcea/downloads.html#semcor)) χρησιμοποιώντας την αναπαράσταση πλαισίου biLM για το "play" στην αρχική πρόταση. Σε αυτές τις περιπτώσεις, το biLM είναι σε θέση να αποσαφηνίσει τόσο το μέρος του λόγου όσο και τη σημασία της λέξης στην αρχική πρόταση.



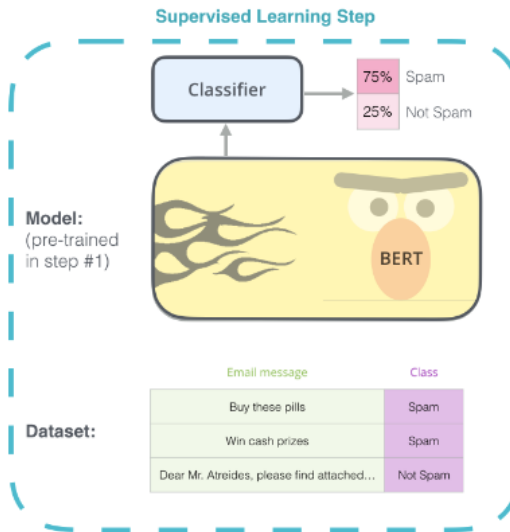
[Το BERT είναι βαθιά αμφίδρομο, το OpenAI GPT είναι μονόδρομο και το ELMo είναι επιφανειακά αμφίδρομο.](#)

1 - **Semi-supervised** training on large amounts of text (books, wikipedia..etc).

The model is trained on a certain task that enables it to grasp patterns in language. By the end of the training process, BERT has language-processing abilities capable of empowering many models we later need to build and train in a supervised way.



2 - **Supervised** training on a specific task with a labeled dataset.



Τα δύο στάδια του τρόπου ανάπτυξης του BERT. Μπορείτε να κατεβάσετε το μοντέλο που έχει προ-εκπαιδευτεί στο βήμα 1 (που έχει εκπαιδευτεί σε δεδομένα χωρίς σχόλια) και να ανησυχείτε μόνο για τη λεπτομερή ρύθμισή του στο βήμα 2. [Πηγή για το εικονίδιο του βιβλίου].

Μιλώντας για το NLP και τα μοντέλα κωδικοποιητή-αποκωδικοποιητή, έχουμε ήδη συζητήσει ότι μία από τις μεγαλύτερες προκλήσεις είναι η έλλειψη δεδομένων εκπαίδευσης [8]. Επειδή η NLP είναι ένας διαφοροποιημένος τομέας με πολλές διαφορετικές εργασίες, τα περισσότερα σύνολα δεδομένων για συγκεκριμένες εργασίες περιέχουν μόνο μερικές χιλιάδες ή μερικές εκατοντάδες χιλιάδες παραδείγματα εκπαίδευσης με ανθρώπινη επισήμανση. Ωστόσο, τα σύγχρονα μοντέλα NLP που βασίζονται στη βαθιά μάθηση βλέπουν οφέλη από πολύ μεγαλύτερες ποσότητες δεδομένων, βελτιώνονται όταν εκπαιδεύονται σε εκατομμύρια ή δισεκατομμύρια σχολιασμένων παραδειγμάτων εκπαίδευσης. Για να βοηθήσουν να καλυφθεί αυτό το κενό σε δεδομένα, οι ερευνητές έχουν αναπτύξει μια ποικιλία τεχνικών για την εκπαίδευση μοντέλων γλωσσικής αναπαράστασης γενικού σκοπού χρησιμοποιώντας τον τεράστιο όγκο μη σχολιασμένου κειμένου στον ιστό (γνωστό ως προ-εκπαίδευση). Όπως απεικονίζεται στη δεξιά πλευρά της διαφάνειας, το προ-εκπαιδευμένο μοντέλο μπορεί στη συνέχεια να ρυθμιστεί λεπτομερώς σε εργασίες NLP με μικρά δεδομένα, όπως η απάντηση ερωτήσεων και η ανάλυση συναισθήματος, με αποτέλεσμα σημαντική βελτίωση της ακρίβειας σε σύγκριση με την εκπαίδευση σε αυτά τα σύνολα δεδομένων από το μηδέν.

Το BERT βασίζεται στην πρόσφατη εργασία στην προ-εκπαίδευση αναπαραστάσεων με βάση το πλαίσιο - συμπεριλαμβανομένων των Semi-supervised Sequence Learning, Generative Pre-Training, ELMo και ULMFit. Ωστόσο, σε αντίθεση με αυτά τα προηγούμενα μοντέλα, το BERT είναι η πρώτη βαθιά αμφίδρομη, μη επιβλεπόμενη γλωσσική αναπαράσταση, που προ-εκπαιδεύεται χρησιμοποιώντας μόνο ένα σώμα απλού κειμένου (στην προκειμένη περίπτωση, τη Wikipedia).

Οι προ-εκπαιδευμένες αναπαραστάσεις μπορούν να είναι είτε χωρίς συμφραζόμενα είτε με συμφραζόμενα, και οι αναπαραστάσεις με συμφραζόμενα μπορούν επιπλέον να είναι μονόδρομες

ή αμφίδρομες. Τα μοντέλα χωρίς συμφραζόμενα, όπως το word2vec ή το GloVe, δημιουργούν μια ενιαία αναπαράσταση ενσωμάτωσης λέξης για κάθε λέξη του λεξιλογίου. Για παράδειγμα, η λέξη "τράπεζα" θα έχει την ίδια αναπαράσταση χωρίς συμφραζόμενα στις λέξεις "τραπεζικός λογαριασμός" και "όχθη του ποταμού". Τα συγκεκριμένα μοντέλα αντ' αυτού δημιουργούν μια αναπαράσταση κάθε λέξης που βασίζεται στις άλλες λέξεις της πρότασης. Για παράδειγμα, στην πρόταση "Είχα πρόσβαση στον τραπεζικό λογαριασμό", ένα μονόδρομο πλαίσιακό μοντέλο θα αναπαριστούσε την "τράπεζα" με βάση το "είχα πρόσβαση στον" αλλά όχι τον "λογαριασμό". Ωστόσο, το BERT αναπαριστά την "τράπεζα" χρησιμοποιώντας τόσο το προηγούμενο όσο και το επόμενο πλαίσιο - "είχα πρόσβαση στον ... λογαριασμό" - ξεκινώντας από το κάτω μέρος ενός βαθύ νευρωνικού δικτύου, καθιστώντας το βαθιά αμφίδρομο.

Στην αριστερή πλευρά της διαφάνειας, μπορείτε να βρείτε μια οπτικοποίηση της αρχιτεκτονικής του νευρωνικού δικτύου της BERT σε σύγκριση με προηγούμενες σύγχρονες μεθόδους προεκπαίδευσης με βάση το πλαίσιο, όπως φαίνεται παρακάτω. Τα βέλη υποδεικνύουν τη ροή πληροφοριών από το ένα επίπεδο στο επόμενο. Τα πράσινα κουτιά στην κορυφή υποδεικνύουν την τελική αναπαράσταση κάθε λέξης εισόδου με συμφραζόμενα.

Ένα άλλο σημαντικό σημείο για την BERT είναι ότι είναι δυνατή η εκπαίδευση από Cloud TPUs και αυτό μας έδωσε την ελευθερία να πειραματιστούμε γρήγορα, να αποσφαλματώσουμε και να βελτιώσουμε τα μοντέλα μας, κάτι που ήταν κρίσιμο για να μπορέσουμε να προχωρήσουμε πέρα από τις υπάρχουσες τεχνικές προεκπαίδευσης. Η αρχιτεκτονική του μοντέλου Transformer, που αναπτύχθηκε από ερευνητές της Google το 2017, μας έδωσε επίσης τα θεμέλια που χρειαζόμασταν για να κάνουμε το BERT επιτυχημένο. Ο Transformer υλοποιείται στην έκδοση ανοιχτού κώδικα, καθώς και στη βιβλιοθήκη tensor2tensor. Για να αξιολογήσουμε την απόδοση, μπορούμε να συγκρίνουμε το BERT με άλλα συστήματα NLP τελευταίας τεχνολογίας. Είναι σημαντικό ότι το BERT πέτυχε όλα τα αποτελέσματά του χωρίς σχεδόν καμία αλλαγή στην αρχιτεκτονική του νευρωνικού δικτύου που να αφορά συγκεκριμένη εργασία. Στο SQuAD v1.1, το BERT επιτυγχάνει 93,2% της βαθμολογίας F1 (ένα μέτρο ακρίβειας), ξεπερνώντας την προηγούμενη βαθμολογία 91,6% της τελευταίας τεχνολογίας και τη βαθμολογία 91,2% σε ανθρώπινο επίπεδο.

Η BERT βελτιώνει επίσης την τελευταία λέξη της τεχνολογίας κατά 7,6% απόλυτα στο πολύ δύσκολο σημείο αναφοράς GLUE, ένα σύνολο 9 διαφορετικών εργασιών κατανόησης φυσικής γλώσσας (NLU). Ο όγκος των δεδομένων εκπαίδευσης με ανθρώπινη επισήμανση σε αυτές τις εργασίες κυμαίνεται από 2.500 έως 400.000 παραδείγματα και η BERT βελτιώνει σημαντικά την ακρίβεια της τελευταίας τεχνολογίας σε όλες τις εργασίες.

Εφαρμογές επεξεργασίας φυσικής γλώσσας



[Εικ1](#) [Εικ2](#)

Κατά τη διάρκεια αυτού του μαθήματος NLP είδαμε πολλές τεχνικές NLP που σχετίζονται με τις διάφορες εφαρμογές, περιγράψαμε διαφορετικά μοντέλα, μιλήσαμε για τα πλεονεκτήματα και τα μειονεκτήματα και, ως συμπέρασμα μπορούμε να πούμε ότι δεν υπάρχει μόνο ένα μοντέλο για κάθε εφαρμογή (όπως άλλες εφαρμογές μηχανικής μάθησης).

Όπως πάντα: "εξαρτάται από το πρόβλημα" ...και από τα δεδομένα, την υπολογιστική ισχύ κ.λπ. Η συνδυαστική μάθηση είναι στις περισσότερες περιπτώσεις η μεθοδολογία με τις καλύτερες επιδόσεις για την αντιμετώπιση προβλημάτων NLP.

Η αλληλεπίδραση υπολογιστών-γλώσσας κατηγοριοποιείται ανάλογα με την εργασία που πρέπει να επιτελεστεί: η περίληψη ενός μακροσκελούς εγγράφου, η μετάφραση μεταξύ δύο ανθρώπινων γλωσσών ή η ανίχνευση ανεπιθύμητων μηνυμάτων ηλεκτρονικού ταχυδρομείου είναι παραδείγματα εργασιών που σήμερα μπορούν να επιτελεστούν αξιοπρεπώς από μια μηχανή.

Τώρα θέλουμε να εξετάσουμε το NLP από διαφορετική οπτική γωνία: τις επιχειρηματικές εφαρμογές :

Εξυπηρέτηση πελατών (Customer service)

Οι αλληλεπιδράσεις μεταξύ πελατών και εταιρειών περιέχουν πολλά χρήσιμα ψίχουλα που υποδεικνύουν τους λόγους δυσαρέσκειας των πελατών, ενώ η ίδια η αλληλεπίδραση μπορεί να αποτελέσει αιτία δυσαρέσκειας. Πολλές εταιρείες απομαγνητοφωνούν και αναλύουν πλέον τις καταγραφές κλήσεων πελατών. Επίσης, αναπτύσσουν chatbots και αυτοματοποιημένους διαδικτυακούς βοηθούς για να παρέχουν άμεση ανταπόκριση σε απλές ανάγκες και να μειώνουν το φόρτο για τους αντιπροσώπους εξυπηρέτησης πελατών. Έτσι, οι σχετικές εργασίες NLP περιλαμβάνουν:

- Ανάλυση κειμένου από την "αναγνώριση ομιλίας", η οποία μετατρέπει τον προφορικό λόγο σε κείμενο (Siri, Google Now, μεταφραστής του Skype)
- Απάντηση ερωτήσεων, η οποία περιλαμβάνει ακριβώς αυτό - απάντηση σε ερωτήσεις που θέτουν οι άνθρωποι σε φυσική γλώσσα, προκειμένου να αυτοματοποιηθεί όσο το δυνατόν περισσότερο η διαδικασία

Παρακολούθηση φήμης (Reputation Monitoring)

Καθώς οι καταναλωτές έχουν αρχίσει να εκφράζουν τα παράπονά τους στο Twitter και το Facebook, η παρακολούθηση και η διαχείριση της φήμης έχει καταστεί ύψιστη προτεραιότητα για τις επιχειρήσεις. Οι εταιρείες μπορούν πλέον να σαρώνουν ολόκληρο τον ιστό για αναφορές της μάρκας και των προϊόντων τους και να αναγνωρίζουν περιπτώσεις στις οποίες πρέπει να αναλάβουν δράση. Οι σχετικές εργασίες NLP για αυτή την εφαρμογή περιλαμβάνουν:

- Ανάλυση συναισθήματος, η οποία προσδιορίζει τη στάση, τη συναισθηματική κατάσταση, την κρίση ή την πρόθεση του συγγραφέα.
- Ανάλυση πυρήνων αναφοράς, η οποία συνδέει τις αντωνυμίες με τα σωστά αντικείμενα.

Νοημοσύνη της αγοράς (Market Intelligence)

Οι αγορές επηρεάζονται από την ανταλλαγή πληροφοριών - μεταξύ εταιρειών/μετόχων, κυβερνήσεων/πολιτών ή απλώς ατόμων. Η γνώση της κατάστασης ενός κλάδου είναι απαραίτητη για την ανάπτυξη μιας αποτελεσματικής στρατηγικής, αλλά τα κανάλια διανομής περιεχομένου σήμερα παράγουν τόσες πολλές πληροφορίες που γίνεται δύσκολο να συμβαδίσει κανείς. Αυτό ισχύει ιδιαίτερα στις χρηματοπιστωτικές αγορές, γι' αυτό και τα hedge funds χρησιμοποιούν συστηματικά NLP για να βελτιώσουν τα μοντέλα τους. Οι σχετικές εργασίες NLP για αυτή την εφαρμογή περιλαμβάνουν:

- Εξαγωγή συμβάντων, η οποία αναγνωρίζει τι συμβαίνει σε μια οντότητα. Οι συγχωνεύσεις και εξαγορές, οι αλλαγές στην απασχόληση, το κλείσιμο συμφωνιών και οτιδήποτε άλλο συμβαίνει σε οργανισμούς ή ανθρώπους μπορεί να εξαχθεί αυτόματα.
- Ταξινόμηση προτάσεων, ή τοποθέτηση μιας πρότασης σε ένα προκαθορισμένο σύνολο κάδων.

Κανονιστική συμμόρφωση

Ένα παράδειγμα συμμόρφωσης είναι η φαρμακοεπαγρύπνηση, οι μελέτες που γίνονται μετά την κυκλοφορία ενός φαρμάκου στην αγορά για τη συλλογή πληροφοριών σχετικά με τις παρενέργειές του. Πολλές πληροφορίες σχετικά με τις ανεπιθύμητες ενέργειες φαρμάκων (ADEs) βρίσκονται σε αυτό που ονομάζεται μη δομημένες κλινικές αφηγήσεις, ή στις αναφορές των ασθενών για την υγεία τους. Οι φαρμακευτικές εταιρείες εξάγουν τα ADEs από: ηλεκτρονικά αρχεία υγείας (EHR), μέσα κοινωνικής δικτύωσης και φόρουμ για ασθενείς που διαμαρτύρονται για ένα πρόβλημα, και τάσεις και μοτίβα αναζήτησης στο διαδίκτυο για ασθενείς που αναζητούν μια λύση. Οι σχετικές εργασίες NLP για αυτή την εφαρμογή περιλαμβάνουν:

- Αναγνώριση ονομαστικών οντοτήτων (NER), η οποία εξάγει τα ονόματα φαρμάκων, ασθενειών, ασθενών και φαρμακευτικών εταιρειών χρησιμοποιώντας μεθόδους βασισμένες σε κανόνες ή στατιστικές μεθόδους.
- Ανίχνευση συσχετίσεων, που χρησιμοποιείται για τον εντοπισμό του πλαισίου στο οποίο αναφέρεται η ADE.

Υγειονομική περίθαλψη

Ο κλάδος της υγειονομικής περίθαλψης αναπτύσσεται διαρκώς και προσελκύει πολλούς πόρους, οπότε είναι φυσικό οι κύριοι παίκτες να αναζητούν την ΤΝ ως πιθανή ώθηση για την επιχείρησή τους. Στη συντήρηση πολύπλοκων (και δαπανηρών) διαγνωστικών συστημάτων, η ΤΝ χρησιμοποιείται για την αποφυγή βλαβών και την όσο το δυνατόν μεγαλύτερη επέκταση του χρόνου λειτουργίας του μηχανήματος. [10] Μια καινοτόμος εφαρμογή της τεχνολογίας ΤΝ στις πρακτικές υγείας είναι η διάγνωση και η θεραπεία για την παροχή αποτελεσματικών και αποδοτικών υπηρεσιών, στις οποίες μπορεί να έχει πρόσβαση η πλειονότητα των ανθρώπων, ξεπερνώντας έτσι τυχόν περιφερειακά μειονεκτήματα (πρόσβαση σε αγροτικές περιοχές μέσω υπηρεσιών ηλεκτρονικής υγείας).

Ωστόσο, ο τεράστιος όγκος δεδομένων που συλλέγεται με τη μορφή τέτοιων διαγνώσεων που γράφουν οι άνθρωποι, πληροφορίες αισθητήρων, δεδομένα παρακολούθησης, δεδομένα λειτουργιών και διαχείρισης της υγειονομικής περίθαλψης είναι αδόμετος, κακής ποιότητας και, λαμβάνοντας υπόψη την τρέχουσα μορφή, είναι ουσιαστικά άχρηστος. Η ΝLP, σε αυτό το πλαίσιο, μπορεί να θεωρηθεί ως μία από τις αποτελεσματικές προσεγγίσεις για την αντιμετώπιση αυτών των προβλημάτων με τη χρήση της για την ανάλυση πληροφοριών και την εξαγωγή κρίσιμων σειρών δεδομένων, προσφέροντας έτσι την ευκαιρία αξιοποίησης των αδόμετων δεδομένων. Έχει ήδη χρησιμοποιηθεί σε διάφορες επιχειρήσεις υγειονομικής περίθαλψης σε όλο τον κόσμο.

Οι κύριοι παράγοντες που οδηγούν στις εφαρμογές ΝLP στην υγειονομική περίθαλψη είναι οι εξής:

- η αυξανόμενη ανάγκη για το χειρισμό της αύξησης των κλινικών δεδομένων,
- η υποστήριξη της φροντίδας με βάση την αξία και της διαχείρισης της υγείας του πληθυσμού,
- βελτίωση της κλινικής τεκμηρίωσης και αποτελεσματικότερη κωδικοποίηση με τη βοήθεια υπολογιστή,
- βελτίωση των αλληλεπιδράσεων μεταξύ ασθενών και παρόχων υγειονομικής περίθαλψης,
- ενδυνάμωση των ασθενών με υγειονομική παιδεία,
- αυξανόμενη ανάγκη για υψηλότερη ποιότητα υγειονομικής περίθαλψης.

Μέχρι στιγμής, οι εφαρμογές της ΝLP στην υγειονομική περίθαλψη έχουν επιτύχει αξιοσημείωτα αποτελέσματα, με αποτέλεσμα η ανάπτυξή της στις τεχνολογίες υγειονομικής περίθαλψης να αυξάνεται τα τελευταία χρόνια.

Όπως μπορείτε να φανταστείτε, ο κατάλογος των πιθανών εφαρμογών του ΝLP είναι μεγαλύτερος (καταπολέμηση του εγκλήματος, βοηθός στο σχολείο και στο μάθημα...), αλλά η διαφάνεια θέλει να οδηγήσει την προσοχή στην "ηθική της εφαρμογής του ΝLP στις επιχειρήσεις". Σύμφωνα με τον ορισμό [40] "Επιχειρηματική Ηθική" είναι η μελέτη των κατάλληλων επιχειρηματικών πολιτικών και πρακτικών σχετικά με δυνητικά αμφιλεγόμενα θέματα, όπως η εταιρική διακυβέρνηση, οι εσωτερικές συναλλαγές, η δωροδοκία, οι διακρίσεις, η εταιρική κοινωνική ευθύνη και οι ευθύνες εμπιστοσύνης.

Η δεοντολογία είναι κάτι που κάθε επιχειρηματική εφαρμογή πρέπει να λαμβάνει υπόψη της, ιδίως στην ΤΝ όπου η χρήση της βαθιάς μάθησης παράγει συστήματα "μαύρου κουτιού" εξαιρετικά δύσκολο να διερευνηθούν και να κατανοηθούν με κάθε λεπτομέρεια: στην πραγματικότητα κανείς δεν μπορεί να πει γιατί ένα βαθύ νευρωνικό δίκτυο παράγει μια έξοδο από μια συγκεκριμένη είσοδο. Η πιο πιθανή απάντηση είναι "επειδή το νευρωνικό δίκτυο έχει εκπαιδευτεί σε συγκεκριμένα δεδομένα και έχει μάθει...", που κρύβει πολλή πολυπλοκότητα αλλά δεν αρκεί: εξαρτάται επίσης από το μοντέλο, από τον σχεδιασμό της λύσης, τον τύπο εκπαίδευσης, τις εκατοντάδες υπερπαραμέτρους κ.λπ. [40]

Μην υποτιμάτε τους κινδύνους αποτυχίας στη δεοντολογία κατά την παροχή επιχειρηματικών λύσεων NLP, επειδή η γλώσσα είναι ένα υποκατάστατο της ανθρώπινης συμπεριφοράς και ένα ισχυρό σήμα ατομικών χαρακτηριστικών. Οι άνθρωποι χρησιμοποιούν αυτό το σήμα συνειδητά, για να παρουσιάσουν τον εαυτό τους με συγκεκριμένο τρόπο, αλλά μπορούν επίσης να αναγνωριστούν ως μέλη συγκεκριμένων ομάδων από τη χρήση υποσυνείδητων χαρακτηριστικών. Οι κυριότεροι κίνδυνοι που συνδέονται με τη NLP στις επιχειρηματικές εφαρμογές είναι:

Αποκλεισμός

Ως αποτέλεσμα της σύνδεσης μεταξύ της γλώσσας και του ανθρώπου, κάθε σύνολο δεδομένων φέρει μια δημογραφική προκατάληψη, δηλαδή λανθάνουσες πληροφορίες σχετικά με τα δημογραφικά στοιχεία σε αυτό. Η υπερβολική προσαρμογή σε αυτούς τους παράγοντες μπορεί να έχει σοβαρές επιπτώσεις στη δυνατότητα εφαρμογής των ευρημάτων. Πιθανές συνέπειες είναι ο αποκλεισμός ή η δημογραφική παραποίηση και αποτελεί παρενέργεια των δεδομένων που χρησιμοποιούνται για την εκπαίδευση του μοντέλου.

Υπεργενίκευση

είναι μια παρενέργεια της μοντελοποίησης. Ως παράδειγμα, εξετάζουμε την αυτόματη εξαγωγή συμπερασμάτων για τα χαρακτηριστικά των χρηστών, μια κοινή και ενδιαφέρουσα εργασία NLP, η λύση της οποίας υπόσχεται επίσης πολλές χρήσιμες εφαρμογές, όπως μηχανές συστάσεων και ανίχνευση απάτης ή εξαπάτησης. Το κόστος των ψευδώς θετικών αποτελεσμάτων φαίνεται χαμηλό: μπορεί να προβληματιστούμε ή να διασκεδάσουμε όταν λαμβάνουμε ένα μήνυμα ηλεκτρονικού ταχυδρομείου που μας απευθύνεται με λάθος φύλο, αλλά τι θα γινόταν αν το σύστημα χρησιμοποιούνταν για την πρόβλεψη του σεξουαλικού προσανατολισμού ή των θρησκευτικών απόψεων; Το θέμα αυτό εισάγει νέες προκαταλήψεις.

Έκθεση

Ένα τέτοιο θέμα μπορεί να οδηγήσει σε ένα ψυχολογικό φαινόμενο που ονομάζεται ευριστική της διαθεσιμότητας (availability heuristic) (Tversky και Kahneman, 1973): αν οι άνθρωποι μπορούν να ανακαλέσουν ένα συγκεκριμένο γεγονός ή έχουν γνώσεις για συγκεκριμένα πράγματα, συμπεραίνουν ότι πρέπει να είναι πιο σημαντικό. Η ευρετική αυτή κατάσταση φορτίζεται ηθικά όταν χαρακτηριστικά όπως η βία ή τα αρνητικά συναισθήματα συνδέονται πιο έντονα με ορισμένες ομάδες ή εθνότητες. Η υπερβολική έκθεση δημιουργεί έτσι προκαταλήψεις που μπορούν να οδηγήσουν σε διακρίσεις. Από την άλλη πλευρά, η υποέκθεση μπορεί να επηρεάσει αρνητικά την αξιολόγηση. Η NLP

τείνει να εστιάζει σε ινδοευρωπαϊκές πηγές δεδομένων/κειμένων, παρά σε μικρές γλώσσες από άλλες γλωσσικές ομάδες, για παράδειγμα στην Ασία ή την Αφρική. Αυτή η εστίαση δημιουργεί μια ανισορροπία στις διαθέσιμες ποσότητες επισημασμένων δεδομένων. Τα περισσότερα από τα υπάρχοντα επισημειωμένα δεδομένα καλύπτουν μόνο ένα μικρό σύνολο γλωσσών.

Διτλή η χρήση

Όπως κάθε τεχνολογία, θα μπορούσε να χρησιμοποιηθεί με "καλό οδηγό" ή όχι: δηλαδή, οι τεχνικές NLP μπορούν να χρησιμοποιηθούν για την ανίχνευση πλαστών κριτικών, αλλά και για την αρχική παραγωγή τους.

Κατά τη διάρκεια αυτού του μαθήματος μιλήσαμε για εφαρμογές του NLP και ηθικά θέματα, αλλά τι γίνεται με το μέλλον; Το μέλλον αυτού του τομέα αφορά τη μετάβαση από το NLP στην κατανόηση φυσικής γλώσσας (NLU), όπου δημιουργείται μια βαθύτερη σύνδεση μεταξύ των εννοιών που επικοινωνούνται και των γεγονότων για τον κόσμο. Στην τεχνητή νοημοσύνη, αυτό θεωρείται ένα AI-hard πρόβλημα: για να επιλυθεί, οι υπολογιστές πρέπει να είναι εξίσου ευφυείς με τους ανθρώπους.

Περιπτώσεις χρήσης

ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 1: ΕΞΥΠΗΡΕΤΗΣΗ ΠΕΛΑΤΩΝ

Οι αλληλεπιδράσεις μεταξύ πελατών και εταιρειών περιέχουν πολλά χρήσιμα στοιχεία που υποδεικνύουν τους λόγους δυσαρέσκειας των πελατών, ενώ η ίδια η αλληλεπίδραση μπορεί να αποτελέσει αιτία δυσαρέσκειας.

Πολλές εταιρείες απομαγνητοφωνούν και αναλύουν πλέον τις καταγραφές κλήσεων πελατών. Επίσης, αναπτύσσουν chatbots και αυτοματοποιημένους διαδικτυακούς βοηθούς για να παρέχουν άμεση ανταπόκριση σε απλές ανάγκες και να μειώνουν το φόρτο για τους αντιπροσώπους εξυπηρέτησης πελατών. Έτσι, οι σχετικές εργασίες NLP περιλαμβάνουν:

Ανάλυση κειμένου από την "αναγνώριση ομιλίας", η οποία μετατρέπει τον προφορικό λόγο σε κείμενο (Siri, Google Now, μεταφραστής του Skype)

Απάντηση ερωτήσεων, η οποία περιλαμβάνει ακριβώς αυτό - απάντηση σε ερωτήσεις που θέτουν οι άνθρωποι σε φυσική γλώσσα, προκειμένου να αυτοματοποιηθεί όσο το δυνατόν περισσότερο η διαδικασία.

ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 2: ΠΑΡΑΚΟΛΟΥΘΗΣΗ ΤΗΣ ΦΗΜΗΣ

Καθώς οι καταναλωτές έχουν αρχίσει να εκφράζουν τα παράπονά τους στο Twitter και το Facebook, η παρακολούθηση και η διαχείριση της φήμης έχουν γίνει ύψιστη προτεραιότητα για τις

επιχειρήσεις. Οι εταιρείες μπορούν πλέον να σαρώνουν ολόκληρο τον ιστό για αναφορές της μάρκας και των προϊόντων τους και να αναγνωρίζουν περιπτώσεις στις οποίες πρέπει να αναλάβουν δράση.

Οι σχετικές εργασίες NLP για αυτή την εφαρμογή περιλαμβάνουν:

Ανάλυση συναισθήματος, η οποία προσδιορίζει τη στάση, τη συναισθηματική κατάσταση, την κρίση ή την πρόθεση του συγγραφέα.

Ανάλυση πυρήνων αναφοράς, η οποία συνδέει τις αντωνυμίες με τα σωστά αντικείμενα.

ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 3: ΠΛΗΡΟΦΟΡΗΣΗ ΑΓΟΡΑΣ

Οι αγορές επηρεάζονται από την ανταλλαγή πληροφοριών - μεταξύ εταιρειών/μετόχων, κυβερνήσεων/πολιτών ή απλώς ατόμων. Η γνώση της κατάστασης ενός κλάδου είναι απαραίτητη για την ανάπτυξη μιας αποτελεσματικής στρατηγικής, αλλά τα κανάλια διανομής περιεχομένου σήμερα παράγουν τόσες πολλές πληροφορίες που είναι δύσκολο να συμβαδίσει κανείς.

Αυτό ισχύει ιδιαίτερα στις χρηματοπιστωτικές αγορές, γι' αυτό και τα hedge funds χρησιμοποιούν συστηματικά NLP για να βελτιώσουν τα μοντέλα τους. Οι σχετικές εργασίες NLP για αυτή την εφαρμογή περιλαμβάνουν:

Εξαγωγή γεγονότων, η οποία αναγνωρίζει τι συμβαίνει σε μια οντότητα. Συγχωνεύσεις και εξαγορές, αλλαγές στην απασχόληση, κλείσιμο συμφωνιών και οτιδήποτε άλλο συμβαίνει σε οργανισμούς ή ανθρώπους μπορεί να εξαχθεί αυτόματα.

Ταξινόμηση προτάσεων, ή τοποθέτηση μιας πρότασης σε ένα προκαθορισμένο σύνολο κάδων.

ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 4: ΚΑΝΟΝΙΣΤΙΚΗ ΣΥΜΜΟΡΦΩΣΗ

Ένα παράδειγμα συμμόρφωσης είναι η φαρμακοεπαγρύπνηση, οι μελέτες που διεξάγονται μετά την κυκλοφορία ενός φαρμάκου στην αγορά για τη συλλογή πληροφοριών σχετικά με τις παρενέργειές του. Πολλές πληροφορίες σχετικά με τις ανεπιθύμητες ενέργειες φαρμάκων (ADEs) βρίσκονται σε αυτό που ονομάζεται μη δομημένες κλινικές αφηγήσεις, ή στις αναφορές των ασθενών για την υγεία τους.

Οι φαρμακευτικές εταιρείες εξαγάγουν τα ADEs από τα ηλεκτρονικά αρχεία υγείας (EHR), τα μέσα κοινωνικής δικτύωσης και τα φόρουμ για ασθενείς που διαμαρτύρονται για ένα πρόβλημα, καθώς και τις τάσεις και τα μοτίβα αναζήτησης στο διαδίκτυο για ασθενείς που αναζητούν μια λύση. Οι σχετικές εργασίες NLP για αυτή την εφαρμογή περιλαμβάνουν:

Αναγνώριση ονομαστικών οντοτήτων (NER), η οποία εξάγει τα ονόματα φαρμάκων, ασθενειών, ασθενών και φαρμακευτικών εταιρειών χρησιμοποιώντας μεθόδους βασισμένες σε κανόνες ή στατιστικές μεθόδους.

Ανίχνευση συσχετίσεων που χρησιμοποιείται για τον εντοπισμό του πλαισίου στο οποίο αναφέρεται η ΑΔΕ.

ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 5: ΥΓΕΙΟΝΟΜΙΚΗ ΠΕΡΙΘΑΛΨΗ

Ο κλάδος της υγειονομικής περίθαλψης αναπτύσσεται συνεχώς και προσελκύει πολλούς πόρους, οπότε είναι φυσικό ότι οι βασικοί παίκτες αναζητούν την τεχνητή νοημοσύνη ως πιθανή ώθηση για την επιχείρησή τους. Στη συντήρηση πολύπλοκων (και ακριβών) διαγνωστικών συστημάτων, η τεχνητή νοημοσύνη χρησιμοποιείται για την αποφυγή βλαβών και την όσο το δυνατόν μεγαλύτερη επέκταση του χρόνου λειτουργίας του μηχανήματος. Μια καινοτόμος εφαρμογή της τεχνολογίας ΤΝ στις πρακτικές υγειονομικής περίθαλψης αφορά τη διάγνωση και τη θεραπεία για την παροχή αποτελεσματικών και αποδοτικών υπηρεσιών, στις οποίες μπορεί να έχει πρόσβαση η πλειονότητα των ανθρώπων, ξεπερνώντας έτσι τυχόν περιφερειακά μειονεκτήματα (πρόσβαση σε αγροτικές περιοχές μέσω υπηρεσιών ηλεκτρονικής υγείας).

Ωστόσο, ο τεράστιος όγκος δεδομένων που συλλέγονται με τη μορφή τέτοιων διαγνώσεων που γράφουν οι άνθρωποι, πληροφορίες αισθητήρων, δεδομένα παρακολούθησης, λειτουργίες υγειονομικής περίθαλψης και δεδομένα διαχείρισης είναι αδόμητα, κακής ποιότητας και, λαμβάνοντας υπόψη την τρέχουσα μορφή, είναι ουσιαστικά άχρηστα. Η NLP, σε αυτό το πλαίσιο, μπορεί να θεωρηθεί ως μία από τις αποτελεσματικές προσεγγίσεις για την αντιμετώπιση αυτών των προβλημάτων με τη χρήση της για την ανάλυση πληροφοριών και την εξαγωγή κρίσιμων σειρών δεδομένων, προσφέροντας έτσι την ευκαιρία αξιοποίησης των αδόμητων δεδομένων. Έχει ήδη χρησιμοποιηθεί σε διάφορες επιχειρήσεις υγειονομικής περίθαλψης σε όλο τον κόσμο.

Οι κύριοι παράγοντες που οδηγούν στις εφαρμογές NLP στην υγειονομική περίθαλψη είναι οι εξής:

- η αυξανόμενη ανάγκη για το χειρισμό της αύξησης των κλινικών δεδομένων,
- η υποστήριξη της φροντίδας με βάση την αξία και η διαχείριση της υγείας του πληθυσμού,
- η βελτίωση της κλινικής τεκμηρίωσης και η αποτελεσματικότερη κωδικοποίηση με τη βοήθεια υπολογιστή,
- βελτίωση των αλληλεπιδράσεων μεταξύ ασθενών και παρόχων υγειονομικής περίθαλψης,
- ενδυνάμωση των ασθενών με υγειονομική παιδεία,
- αυξανόμενη ανάγκη για υψηλότερη ποιότητα υγειονομικής περίθαλψης.

Μέχρι στιγμής, οι εφαρμογές της NLP στην υγειονομική περίθαλψη έχουν επιτύχει αξιοσημείωτα αποτελέσματα, με αποτέλεσμα η ανάπτυξή της στις τεχνολογίες υγειονομικής περίθαλψης να αυξάνεται τα τελευταία χρόνια.

Μάθημα 4.4: Προβλήματα μεγάλων δεδομένων, τεχνικές, Hadoop

Θέματα προς συζήτηση

- Εισαγωγή στα μεγάλα δεδομένα και την ανάλυση μεγάλων δεδομένων
- Παράδειγμα μεγάλων δεδομένων
- Αλυσίδα αξίας μεγάλων (μαζικών) δεδομένων
- Επισκόπηση Map reduce
- Πλαίσιο Hadoop
- Από το Hadoop στο Spark

Σημειώσεις διάλεξης

Μεγάλες εταιρείες όπως η Google, η Amazon ή το Netflix, επεξεργάζονται μεγάλα δεδομένα που παρέχουμε για να καλύψουν τις προσωπικές μας ανάγκες, να μας πουλήσουν κάποια προϊόντα ή να μας δείξουν ποιες τηλεοπτικές σειρές μπορεί να μας αρέσουν με βάση τα ενδιαφέροντά μας.

Η αποθήκευση και η διάδοση των πληροφοριών έχει φθάσει σε πρωτοφανές επίπεδο. Παράγεται μια τόσο μεγάλη ποσότητα κάθε λεπτό, κάτι που ήταν αδιανόητο μόλις πριν από λίγα χρόνια.

Εκτιμάται ότι 2,7 Zetabyte κυκλοφορούν σήμερα στο ψηφιακό σύμπαν (1 zetabyte ισούται με ένα τρισεκατομμύριο bytes!) Και προβλέπεται να γίνουν 180 έως το 2025

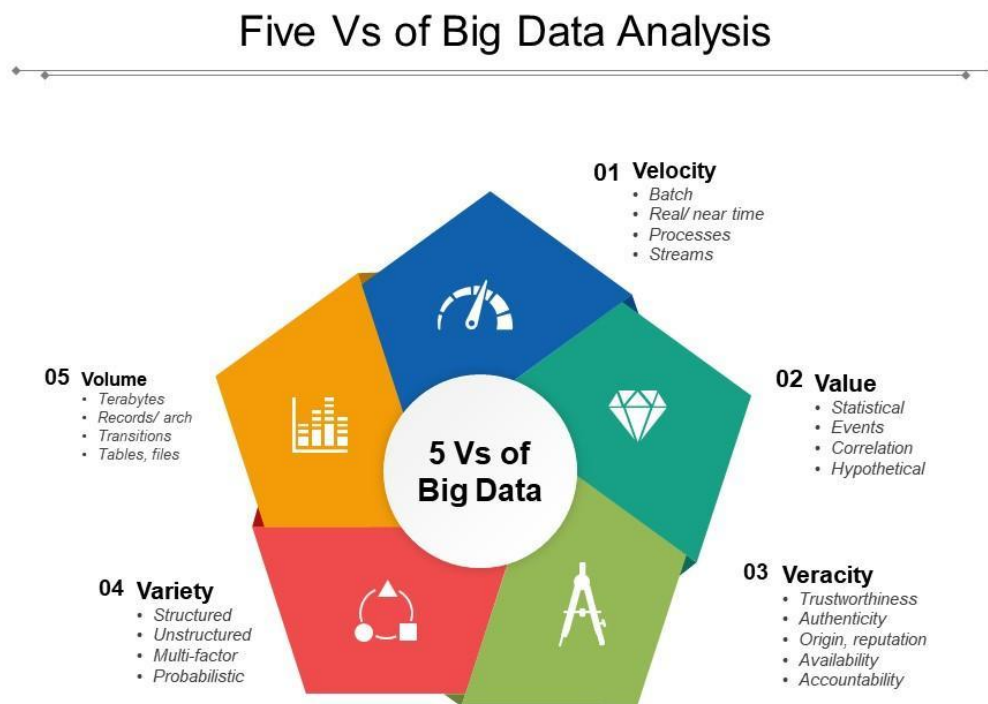
Ο όρος **Big Data** αναφέρεται σε σύνολα δεδομένων που είναι τόσο μεγάλα σε όγκο και τόσο πολύπλοκα που οι παραδοσιακές αρχιτεκτονικές λογισμικού και πληροφορικής αδυνατούν να τα συλλάβουν, να τα διαχειριστούν και να τα επεξεργαστούν σε εύλογο χρόνο. Εάν μια παραδοσιακή βάση δεδομένων μπορεί να διαχειριστεί πίνακες που να αποτελούνται ίσως από εκατομμύρια γραμμές, αλλά σε δεκάδες ή μερικές εκατοντάδες στήλες, τα μεγάλα δεδομένα απαιτούν εργαλεία ικανά να διαχειριστούν τα ίδια αριθμό εγγράφων, αλλά με χιλιάδες στήλες.

Επιπλέον, συχνά τα δεδομένα δεν είναι καν διαθέσιμα σε δομημένη μορφή, εύκολα ταξινομήσιμα σε γραμμές και στήλες με ακρίβεια, αλλά υπάρχουν με τη μορφή εγγράφων, μεταδεδομένων, γεωγραφικών θέσεων, τιμών που ανιχνεύονται από αισθητήρες IoT και πολυάριθμων άλλες μορφές, από την ημιδομημένη έως την πλήρως αποδομημένη.

Από το λιανικό εμπόριο έως την υγειονομική περίθαλψη και ακόμη και τις επιχειρήσεις, τα οφέλη από τη χρήση των μεγάλων δεδομένων έχουν ήδη γίνει αντιληπτά.

Η ποσότητα και η πολυπλοκότητα που κάνουν ένα σύνολο δεδομένων να μπορεί να ονομαστεί "Big Data" είναι ένα θέμα που συζητείται.

Το 2001, ο Doug Laney (σήμερα αντιπρόεδρος και διακεκριμένος αναλυτής της Gartner Chief Data Officer Research and Advisory Team) περιέγραψε σε μια έκθεση το μοντέλο που περιγράφει εν συντομία τα μεγάλα δεδομένα με 3V: όγκος, ταχύτητα και ποικιλία. Ένα απλό μοντέλο για τον ορισμό των νέων δεδομένων που δημιουργούνται από την αύξηση των πηγών πληροφόρησης και γενικότερα από την εξέλιξη των τεχνολογιών. Σήμερα το παράδειγμα Laney έχει εμπλουτιστεί με τις μεταβλητές της Αληθινότητας και της Μεταβλητότητας και για το λόγο αυτό μιλάμε για 5V:



Ταχύτητα (Velocity): Η ταχύτητα αναφέρεται στην ταχύτητα με την οποία παράγονται, συλλέγονται και αναλύονται τα δεδομένα. Τα δεδομένα ρέουν συνεχώς μέσω πολλαπλών καναλιών, όπως συστήματα υπολογιστών, δίκτυα, μέσα κοινωνικής δικτύωσης, κινητά τηλέφωνα κ.λπ. Στο σημερινό επιχειρηματικό περιβάλλον με βάση τα δεδομένα, ο ρυθμός με τον οποίο αυξάνονται τα δεδομένα μπορεί να περιγραφεί καλύτερα ως "καταιγιστικός" και "άνευ προηγούμενου". Τώρα, τα δεδομένα αυτά θα πρέπει επίσης να συλλέγονται όσο το δυνατόν σε πραγματικό χρόνο, καθιστώντας τα σωστά δεδομένα διαθέσιμα τη σωστή στιγμή. Η ταχύτητα πρόσβασης στα δεδομένα έχει άμεσο αντίκτυπο στη λήψη έγκαιρων και ακριβών επιχειρηματικών αποφάσεων. Αρκετές τεχνολογίες μεγάλων δεδομένων μας επιτρέπουν σήμερα να συλλέγουμε και να αναλύουμε δεδομένα καθώς παράγονται σε πραγματικό χρόνο

Όγκος (Volume): Ο όγκος των μεγάλων δεδομένων ορίζει την "ποσότητα" των δεδομένων που παράγονται. Η αξία των δεδομένων εξαρτάται επίσης από το μέγεθος των δεδομένων. Σήμερα τα δεδομένα παράγονται από διάφορες πηγές σε διαφορετικές μορφές - δομημένες και μη δομημένες. Ορισμένες από αυτές τις μορφές περιλαμβάνουν έγγραφα word και excel, PDF και εκθέσεις μαζί με περιεχόμενο πολυμέσων όπως εικόνες και βίντεο. Λόγω της έκρηξης των

δεδομένων που προκλήθηκε από τα ψηφιακά και κοινωνικά μέσα, τα δεδομένα παράγονται με ταχείς ρυθμούς σε τόσο μεγάλα μεγέθη, που αποτελεί πρόκληση για τις επιχειρήσεις να τα αποθηκεύουν και να τα επεξεργάζονται χρησιμοποιώντας συμβατικές μεθόδους επιχειρηματικής ευφυΐας και ανάλυσης. Οι επιχειρήσεις πρέπει να εφαρμόσουν σύγχρονα εργαλεία επιχειρηματικής ευφυΐας για την αποτελεσματική καταγραφή, αποθήκευση και επεξεργασία αυτού του πρωτοφανούς όγκου δεδομένων σε πραγματικό χρόνο.

Ποικιλία (Variety): Ενώ ο όγκος και η ταχύτητα των δεδομένων είναι σημαντικοί παράγοντες που προσθέτουν αξία σε μια επιχείρηση, τα μεγάλα δεδομένα συνεπάγονται επίσης την επεξεργασία ποικίλων τύπων δεδομένων που συλλέγονται από ποικίλες πηγές δεδομένων. Οι πηγές δεδομένων μπορεί να περιλαμβάνουν εξωτερικές πηγές καθώς και εσωτερικές επιχειρηματικές μονάδες. Γενικά, τα μεγάλα δεδομένα ταξινομούνται σε δομημένα, ημιδομημένα και αδόμητα δεδομένα. Ενώ τα δομημένα δεδομένα είναι εκείνα των οποίων η μορφή, το μήκος και ο όγκος είναι σαφώς καθορισμένα, τα ημιδομημένα δεδομένα είναι εκείνα που μπορεί να συμμορφώνονται εν μέρει με μια συγκεκριμένη μορφή δεδομένων. Από την άλλη πλευρά, τα μη δομημένα δεδομένα είναι μη οργανωμένα δεδομένα και δεν συμμορφώνονται με τις παραδοσιακές μορφές δεδομένων. Τα δεδομένα που παράγονται μέσω ψηφιακών και κοινωνικών μέσων (εικόνες, βίντεο, tweets κ.λπ.) μπορούν να ταξινομηθούν ως μη δομημένα δεδομένα.

Ο τεράστιος όγκος δεδομένων που συνήθως συλλέγουν και παράγουν οι οργανισμοί μπορεί να φαίνεται χαοτικός και αδόμητος. Στην πραγματικότητα, σχεδόν 80 τοις εκατό των δεδομένων που παράγονται παγκοσμίως, συμπεριλαμβανομένων των φωτογραφιών, των βίντεο, των δεδομένων κινητής τηλεφωνίας και του περιεχομένου των μέσων κοινωνικής δικτύωσης, είναι αδόμητης φύσης.

Αλήθεια/Εγκυρότητα: Η Αληθινότητα των μεγάλων δεδομένων ή η εγκυρότητα, όπως είναι πιο γνωστή, είναι η διασφάλιση της ποιότητας ή της αξιοπιστίας των δεδομένων που συλλέγονται. Δεδομένου ότι τα μεγάλα δεδομένα είναι τεράστια και περιλαμβάνουν τόσες πολλές πηγές δεδομένων, υπάρχει η πιθανότητα να μην είναι όλα τα συλλεχθέντα δεδομένα καλής ποιότητας ή ακριβή από τη φύση τους. Ως εκ τούτου, κατά την επεξεργασία μεγάλων συνόλων δεδομένων, είναι σημαντικό να διασφαλίζεται ο έλεγχος της εγκυρότητας των δεδομένων πριν προχωρήσει η επεξεργασία τους.

Ανάλυση μεγάλων δεδομένων

Η **ανάλυση μεγάλων δεδομένων**, η οποία ονομάζεται επίσης "big data analytics", συνίσταται στη χρήση ιδιαίτερα εξελιγμένων τεχνικών ανάλυσης σε μεγάλες ποσότητες δεδομένων. όγκους δεδομένων (δομημένων, ημιδομημένων και αδόμητων), με σκοπό την περιγραφή γεγονότων ή καταστάσεων, τον εντοπισμό προτύπων, συσχετίσεων ή τάσεων και έτσι να μετατρέψει τα δεδομένα σε χρήσιμες και λειτουργικές πληροφορίες για τη βελτιστοποίηση της λήψης αποφάσεων.

Μπορούν να διακριθούν τέσσερις κύριες κατηγορίες ανάλυσης δεδομένων, ανάλογα με τον σκοπό της ανάλυσης:

Περιγραφική ανάλυση: είναι απαραίτητη για τη δημιουργία εκθέσεων και πραγματοποιείται με τη διενέργεια συνοπτικής και ακριβούς περιγραφικής ανάλυσης των γεγονότων που σας επιτρέπει να έχετε μια σφαιρική εικόνα ενός πλαισίου ή μιας κατάστασης.

Προβλεπτική ανάλυση: είναι ένας από τους πιο συχνά χρησιμοποιούμενους τύπους ανάλυσης δεδομένων και επιτρέπει τον εντοπισμό τάσεων, συσχετίσεων ή σχέσεων αιτίου-αποτελέσματος μεταξύ δεδομένων. Πρόκειται για έναν τύπο ανάλυσης που επιτρέπει τη σκιαγράφηση μελλοντικών σεναρίων ανάπτυξης σε διάφορα πλαίσια ή καταστάσεις.

Συντακτική ανάλυση: χάρη σε καινοτομίες όπως η μηχανική μάθηση και, ως εκ τούτου, χάρη στην υποστήριξη της τεχνητής νοημοσύνης, είναι πλέον δυνατή όχι μόνο η πραγματοποίηση προβλέψεων σχετικά με συγκεκριμένα αποτελέσματα, αλλά και σχετικά με τις βέλτιστες ενέργειες που πρέπει να ληφθούν.

Διαγνωστική ανάλυση: πρόκειται για το είδος της ανάλυσης που χρησιμοποιείται για τον συγκεκριμένο εντοπισμό της αιτίας για κάτι ή για ένα δεδομένο γεγονός, επιτρέποντάς σας να βρείτε τις αιτίες που οδήγησαν στην τρέχουσα κατάσταση. Για τον σκοπό αυτό, οι εταιρείες χρησιμοποιούν συχνά τεχνικές όπως η διάτρηση και η εξόρυξη δεδομένων για να προσδιορίσουν τις αιτίες των τάσεων ή των γεγονότων και να είναι σε θέση να προσδιορίσουν, να επαναλάβουν και να βελτιστοποιήσουν τις ενέργειες που οδήγησαν σε θετικά αποτελέσματα.

(Η προγνωστική ανάλυση βοηθά στην εύρεση πιθανών αποτελεσμάτων, ενώ η συντακτική ανάλυση εξετάζει αυτά τα αποτελέσματα και βρίσκει περισσότερες επιλογές.)

Παρόλο που τα μεγάλα δεδομένα και η επιχειρηματική ευφυΐα (BI) είναι δύο τεχνολογίες που χρησιμοποιούνται για την ανάλυση δεδομένων ώστε να βοηθήσουν τις εταιρείες στη διαδικασία λήψης αποφάσεων, υπάρχουν διαφορές μεταξύ τους. Διαφέρουν τόσο στον τρόπο λειτουργίας τους όσο και στον τύπο των δεδομένων που αναλύουν. Η Επιχειρηματική Ευφυΐα βασίζεται στην αρχή του συνδυασμού όλων των συνόλων επιχειρηματικών δεδομένων σε έναν κεντρικό διακομιστή, τα δεδομένα αυτά θα αναλυθούν σε λειτουργία εκτός σύνδεσης, αφού αποθηκευτούν οι πληροφορίες σε μια πλατφόρμα ή περιβάλλον που ονομάζεται Αποθήκη Δεδομένων (Data Warehouse). Τα σύνολα δεδομένων δομούνται σε μια σχεσιακή βάση δεδομένων με πρόσθετα ευρετήρια και μορφές πρόσβασης στους πίνακες της αποθήκης δεδομένων. Ενώ στην περιβάλλον Big Data, τα δεδομένα αποθηκεύονται σε ένα κατανεμημένο σύστημα αρχείων (π.χ. HDFS), αντί να αποθηκεύονται σε έναν κεντρικό διακομιστή. Δεδομένα θα κατανεμηθούν στους κόμβους εργασίας για εύκολη επεξεργασία. Το κατανεμημένο σύστημα αρχείων είναι πολύ πιο ασφαλές και ευέλικτο

Αλυσίδα αξίας των μεγάλων δεδομένων

Η αλυσίδα αξίας των μεγάλων δεδομένων μπορεί να χρησιμοποιηθεί για τη μοντελοποίηση των δραστηριοτήτων υψηλού επιπέδου που συνθέτουν ένα σύστημα πληροφοριών. Η αλυσίδα αξίας των μεγάλων δεδομένων προσδιορίζει τις ακόλουθες βασικές δραστηριότητες υψηλού επιπέδου:

Η απόκτηση δεδομένων (Data acquisition) είναι μία από τις σημαντικότερες προκλήσεις των μεγάλων δεδομένων όσον αφορά τις απαιτήσεις υποδομής και είναι η διαδικασία της συλλογής, φιλτραρίσματος και καθαρισμού των δεδομένων προτού αυτά τοποθετηθούν σε μια αποθήκη δεδομένων ή σε οποιαδήποτε άλλη λύση αποθήκευσης στην οποία μπορεί να πραγματοποιηθεί ανάλυση δεδομένων. Η υποδομή που απαιτείται για την υποστήριξη της απόκτησης μεγάλων δεδομένων πρέπει να παρέχει χαμηλή, προβλέψιμη καθυστέρηση τόσο στη σύλληψη των δεδομένων όσο και στην εκτέλεση ερωτημάτων- να μπορεί να χειρίζεται πολύ υψηλούς όγκους συναλλαγών, συχνά σε κατανεμημένο περιβάλλον. και να υποστηρίζει ευέλικτες και δυναμικές δομές δεδομένων.

Η Ανάλυση Δεδομένων ασχολείται με το να καταστήσει τα ακατέργαστα δεδομένα που αποκτά ένας άνθρωπος ικανά να χρησιμοποιηθούν για τη λήψη αποφάσεων, καθώς και για τη χρήση σε συγκεκριμένους τομείς. Στόχος της ανάλυσης δεδομένων είναι η ανάδειξη των σχετικών δεδομένων, η σύνθεση και η εξαγωγή χρήσιμων κρυφών πληροφοριών με υψηλή δυναμική από επιχειρηματική άποψη μέσω της εξερεύνησης, του μετασχηματισμού και της μοντελοποίησης δεδομένων.

Η επιμέλεια δεδομένων είναι η ενεργός διαχείριση των δεδομένων κατά τη διάρκεια του κύκλου ζωής τους, ώστε να διασφαλίζεται ότι πληρούν τις απαραίτητες απαιτήσεις ποιότητας δεδομένων για την αποτελεσματική χρήση τους. Οι κύριες δραστηριότητες επιμέλειας δεδομένων είναι: δημιουργία περιεχομένου, επιλογή, ταξινόμηση, μετασχηματισμός, επικύρωση και διατήρηση.

Η αποθήκευση δεδομένων είναι η διατήρηση και η διαχείριση δεδομένων με έναν κλιμακούμενο τρόπο που ικανοποιεί τις ανάγκες των εφαρμογών που απαιτούν γρήγορη πρόσβαση στα δεδομένα. Τα συστήματα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS) δεν εγγυώνται την έλλειψη ευελιξία συναλλαγών στη βάση δεδομένων όσον αφορά τις αλλαγές του σχήματος και την απόδοση και την ανοχή σε σφάλματα όταν αυξάνεται ο όγκος και η πολυπλοκότητα των δεδομένων, καθιστώντας τα ακατάλληλα για σενάρια μεγάλων δεδομένων, για το λόγο αυτό οι τεχνολογίες NoSQL έχουν σχεδιαστεί με την επεκτασιμότητα στόχο και παρουσιάζουν ένα ευρύ φάσμα λύσεων που βασίζονται σε εναλλακτικά μοντέλα δεδομένων.

Η χρήση δεδομένων σημαίνει την εφαρμογή των μεγάλων δεδομένων, καλύπτει τους επιχειρηματικούς στόχους που χρειάζονται πρόσβαση στα δεδομένα, τις αναλύσεις τους και τις ενσωμάτωση στη λήψη επιχειρηματικών αποφάσεων, περιλαμβάνει τις χρησιμοποιούμενες τεχνολογικές δέσμες και τα επιμέρους πεδία της υποστήριξης αποφάσεων, προγνωστική ανάλυση, προσομοίωση, εξερεύνηση, οπτικοποίηση και πιο τεχνικές πτυχές της επεξεργασίας ροής δεδομένων.

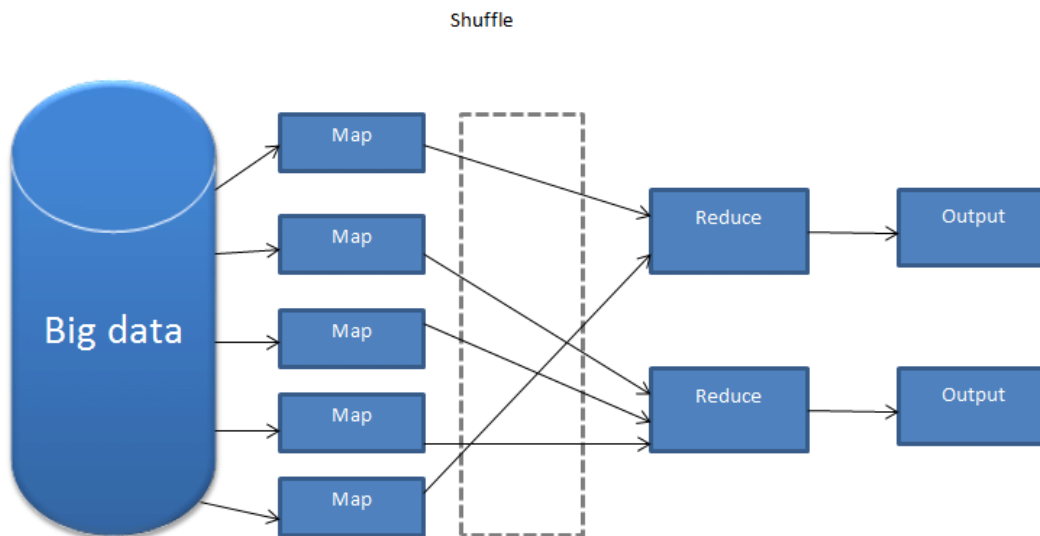
Πώς να διαχειριστεί ένας τόσο μεγάλος όγκος δεδομένων, διαφορετικών μορφών, με παραδοσιακά συστήματα, όπως τα σχεσιακά ΣΔΒΔ που λειτουργούν σε ένα μόνο μηχάνημα; Η λύση εισήχθη από την Google (2004) με το παράδειγμα Map Reduce.

Το MapReduce σχεδιάστηκε για να παίρνει μεγάλα δεδομένα και να χρησιμοποιεί παράλληλους κατανεμημένους υπολογισμούς για να μετατρέπει τα μεγάλα δεδομένα σε δεδομένα μικρού ή κανονικού μεγέθους. δεδομένα.

Η παράλληλη καταναεμημένη επεξεργασία αναφέρεται σε ένα ισχυρό πλαίσιο όπου μαζικός όγκος δεδομένων επεξεργάζεται πολύ γρήγορα κατανέμοντας τις εργασίες επεξεργασίας σε συστάδες διακομιστών κοινής ωφέλειας. Όσον αφορά το MapReduce, οι πλειάδες αναφέρονται σε ζεύγη κλειδιών-τιμών με τα οποία τα δεδομένα ομαδοποιούνται, ταξινομούνται και υποβάλλονται σε επεξεργασία.

Οι εργασίες MapReduce λειτουργούν μέσω ακολουθιών διαδικασιών χαρτογράφησης και μείωσης σε ένα καταναεμημένο σύνολο διακομιστών. Στην εργασία map, αναθέτετε τα δεδομένα σας σε ζεύγη κλειδιών-τιμών, τα μετασχηματίζετε και τα φιλτράρετε. Στη συνέχεια αναθέτετε τα δεδομένα σε κόμβους για επεξεργασία.

Στην εργασία reduce, συγκεντρώνετε αυτά τα δεδομένα σε σύνολα δεδομένων μικρότερου μεγέθους. Τα δεδομένα από το βήμα reduce μετατρέπονται σε μια τυπική μορφή κλειδιού-τιμής - όπου το κλειδί λειτουργεί ως αναγνωριστικό εγγραφής και η τιμή είναι η τιμή που προσδιορίζεται από το το κλειδί. Οι υπολογιστικοί κόμβοι των συστάδων επεξεργάζονται τις εργασίες map και reduce που ορίζει ο χρήστης.



Οι φάσεις για τον υπολογισμό μιας εργασίας MapReduce είναι κατά βάση 4:

1. **Διαχωρισμός των δεδομένων εισόδου:** τα δεδομένα εισόδου χωρίζονται σε η διαμερίσματα, ορισμένου μεγέθους, κάθε διαμέρισμα ανατίθεται σε διαφορετικό χαρτογράφο
2. **Φάση χαρτογράφησης:** παράλληλα, κάθε χαρτογράφος δέχεται ως είσοδο ένα ζεύγος κλειδιού/τιμής, που αντιπροσωπεύεται από μια λογική εγγραφή εισόδου. Στην έξοδο, επιστρέφει μηδέν ή περισσότερα ζεύγη κλειδιών/τιμών για το συγκεκριμένο ζεύγος κλειδιών/τιμών στην είσοδο.
3. **Φάση ανακατανομής και ταξινόμησης:** για κάθε ζεύγος κλειδιού/τιμής που λαμβάνεται από τη φάση map, καθορίζεται reducer που θα πρέπει να το λάβει (κατάτμηση) , επιπλέον τα κλειδιά που εισάγονται σε κάθε μειωτή θα ταξινομηθούν.

4. **Φάση μείωσης:** στα ζεύγη που προέκυψαν στην προηγούμενη φάση, εφαρμόζεται η συνάρτηση μείωσης στις τιμές που έχουν το ίδιο κλειδί, τα αποτελέσματα ομαδοποιούνται, επιστρέφονται στην έξοδο και ενδεχομένως αποθηκεύονται, με βάση τις απαιτήσεις της εργασίας.

Hadoop

Το **Hadoop** είναι ένα πλαίσιο ανοικτού κώδικα για τη συγγραφή και εκτέλεση εφαρμογών που κατανέμονται σε ένα σύμπλεγμα μηχανών, όπου επεξεργάζονται μεγάλες ποσότητες δεδομένων. Η συστάδα Hadoop είναι ένα σύνολο μηχανών βασικών προϊόντων. Οι μηχανές συνδέονται μεταξύ τους μέσω δικτύου και βρίσκονται στην ίδια τοποθεσία. Διάφοροι πελάτες συνδέονται εξ αποστάσεως στη συστάδα Hadoop και υποβάλλουν την εργασία που πρόκειται να εκτελεστεί σε αυτήν. Το Hadoop παρουσιάζει μια κατανεμημένη αρχιτεκτονική master / slave που αποτελείται από τα εξής κύρια στοιχεία:

- Hadoop Common: κοινό επίπεδο λογισμικού που παρέχει λειτουργίες υποστήριξης σε άλλες ενότητες του Hadoop.
- Hadoop Distributed File System (HDFS): κατανεμημένο σύστημα αρχείων που ασχολείται με την αποθήκευση δεδομένων.
- Hadoop Map Reduce μοντέλο προγραμματισμού παρτίδας που επιτρέπει την επεξεργασία μεγάλου όγκου δεδομένων. Στο Hadoop 2, Το MapReduce υλοποιείται ως εφαρμογή YARN.
- Yet Another Resource Negotiator (YARN): είναι ο διαχειριστής συστάδων του Hadoop:

Yet Another Resource Negotiator (YARN): είναι ο διαχειριστής συστάδων Hadoop:

- ενεργεί ως γενικός και κατανεμημένος χρονοπρογραμματιστής εφαρμογών και πόρων.
- εισήχθη στο Hadoop 2 για να ξεπεράσει τα όρια του MapReduce στο Hadoop 1.
- ανταποκρίνεται στα αιτήματα των πελατών δημιουργώντας ένα δοχείο ή μια αφηρημένη έννοια που αντιπροσωπεύει τη συλλογή φυσικών πόρων που διατίθενται για μια εφαρμογή
- παρακολουθεί την εκτέλεση του περιέκτη, τερματίζοντάς την εάν είναι απαραίτητο

Το HDFS master είναι υπεύθυνο για την κατάτμηση των δεδομένων που είναι αποθηκευμένα στους slave κόμβους και παρακολουθεί πού βρίσκονται τα δεδομένα. Ο YARN master εκτελεί τον πραγματικό προγραμματισμό εργασιών για κάθε εφαρμογή YARN. Ο κύριος MapReduce ασχολείται με το πού πρέπει να εκτελεστεί η υπολογιστική εργασία στους κόμβους slave.

Οι πελάτες υποβάλλουν εργασίες στη συστάδα Hadoop: Ο JobTracker προσδιορίζει τις εργασίες που πρέπει να εκτελεστούν και κατανέμει τις εργασίες στο TaskTracker, οι οποίες εκτελούνται στους DataNodes. Ο JobTracker θα ελέγχει την εκτέλεση των TaskTrackers, παρακολουθεί τους πόρους που καταναλώνονται και τους διαθέσιμους και επιλύει τυχόν καταστάσεις αποτυχίας εργασιών. Κάθε TaskTracker διαχειρίζεται τις εργασίες που του έχει αναθέσει ο JobTracker. Κάθε TaskTracker μπορεί να δημιουργήσει πολλαπλές JVM για την παράλληλη διαχείριση πολλαπλών εργασιών χαρτογράφησης και μείωσης. Κάθε TaskTracker πρέπει να αναφέρει συνεχώς την κατάσταση της εργασίας στον JobTracker. Εάν ο JobTracker δεν λαμβάνει τίποτα για ορισμένο

χρονικό διάστημα, τότε θα θεωρήσει ότι ο TaskTracker έχει καταρρεύσει και θα αναθέσει εκ νέου τις εργασίες που αντιστοιχούν σε άλλους κόμβους της συστάδας.

Apache Spark

Το Apache Spark είναι ένα υπολογιστικό πλαίσιο ανοικτού κώδικα γενικού σκοπού που διανέμεται σε συστάδες, παρέχει μια διεπαφή που εγγυάται παραλληλισμό δεδομένων και ανοχή σε σφάλματα κατά τη διάρκεια των υπολογισμών. Αναπτύχθηκε από το UC Berkeley το 2009 και στη συνέχεια δωρήθηκε στο Ίδρυμα Λογισμικού Apache το 2013. Επιπλέον, σχεδιάστηκε για να επιταχύνει τους υπολογισμούς, βασίζεται σε Hadoop MapReduce αλλά επεκτείνει το μοντέλο MapReduce επιτρέποντας περισσότερους τύπους υπολογισμών, οι οποίοι περιλαμβάνουν διαδραστικά ερωτήματα, επεξεργασία ροής και μηχανική μάθηση.

Σημειώσεις εγκατάστασης

Για να εγκαταστήσετε και να ρυθμίσετε το Hadoop ανατρέξτε σε αυτόν τον σύνδεσμο <https://hadoop.apache.org>

Για να εγκαταστήσετε και να ρυθμίσετε το Spark ανατρέξτε σε αυτόν τον σύνδεσμο <https://spark.apache.org>

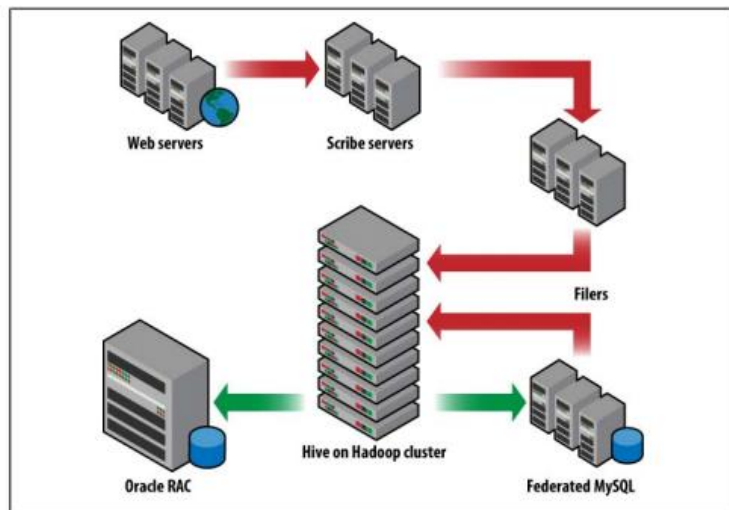
Περιπτώσεις χρήσης

ΑΡΧΙΤΕΚΤΟΝΙΚΕΣ ΔΕΔΟΜΕΝΩΝ BACKEND ΚΑΙ Ο ΡΟΛΟΣ ΤΟΥ HADOOP

Το Hadoop μπορεί επίσης να χρησιμοποιηθεί για την αποθήκευση και την αρχειοθέτηση μαζικών συνόλων δεδομένων. Σε αυτή την περίπτωση χρήσης, θα εξερευνήσουμε τις αρχιτεκτονικές δεδομένων backend και τον ρόλο του Hadoop.

Αρχικά, η αποθήκευση δεδομένων στο Facebook γινόταν εξ ολοκλήρου σε μια περίπτωση Oracle. Στη συνέχεια, το Facebook υιοθέτησε το Hadoop τόσο επειδή οι προγραμματιστές είναι ελεύθεροι να γράφουν προγράμματα για τη μείωση των χαρτών στη γλώσσα που προτιμούν όσο και επειδή το Hadoop χρησιμοποιεί την SQL ως οικογενειακό παράδειγμα για τη λειτουργία σε μεγάλα σύνολα δεδομένων. Αυτό διευκόλυνε ακόμη περισσότερο τους χρήστες να επεξεργάζονται δεδομένα στο σύμπλεγμα Hadoop, καθώς ήταν σε θέση να εκφράζουν κοινούς υπολογισμούς με τη μορφή της SQL, μιας γλώσσας με την οποία οι περισσότεροι μηχανικοί και αναλυτές είναι εξοικειωμένοι. Επιπλέον, το Facebook συνέβαλε στην ανάπτυξη του λογισμικού αποθήκης δεδομένων The Apache Hive™ που διευκολύνει την ανάγνωση, την εγγραφή και τη διαχείριση μεγάλων συνόλων δεδομένων που βρίσκονται σε κατανεμημένη αποθήκευση με τη χρήση SQL, παρέχοντας κλασικά χαρακτηριστικά αποθήκης δεδομένων, όπως η κατάτμηση, η δειγματοληψία και η ευρετηρίαση.

Δεδομένα που παράγονται στο Facebook



Facebook solution for processing data is shown in Figure.

ους τουλάχιστον μία φορά την

ουν κάθε μήνα

μήνα

ίου (σύνδεσμοι στο διαδίκτυο, ες κ.λπ.) που μοιράζονται κάθε

ημερινά
ρα

Το Facebook χρησιμοποιεί το Scribe, μια υπηρεσία συλλογής αρχείων καταγραφής ανοικτού κώδικα, για να καταθέτει σε μια χούφτα διακομιστές NFS εκατοντάδες σύνολα δεδομένων καταγραφής που παράγονται από διακομιστές ιστού, καθώς και από εσωτερικές υπηρεσίες όπως το Search backend. Το Scribe, μια υπηρεσία συλλογής αρχείων καταγραφής ανοικτού κώδικα που αναπτύχθηκε στο Facebook και καταθέτει εκατοντάδες σύνολα δεδομένων καταγραφής με ημερήσιο όγκο δεκάδων terabytes σε μια χούφτα διακομιστές NFS. Ένα μεγάλο μέρος αυτών των δεδομένων καταγραφής αντιγράφεται σε μια κεντρική περίπτωση HDFS με επίσης δεδομένα διαστάσεων που συλλέγονται από εσωτερικές βάσεις δεδομένων MySQL. Όλα τα δεδομένα που συλλέγονται στο HDFS συμπεριλαμβανομένων των δεδομένων καταγραφής από το Scribe και των δεδομένων διαστάσεων από τη βαθμίδα MySQL διατίθενται ως πίνακες με λογικές κατατμήσεις. Μια γλώσσα ερωτημάτων τύπου SQL που παρέχεται από το Hive χρησιμοποιείται σε συνδυασμό με το MapReduce για τη δημιουργία/δημοσίευση μιας ποικιλίας περιλήψεων και αναφορών, καθώς και για την εκτέλεση ιστορικής ανάλυσης σε αυτούς τους πίνακες. Ενώ οι διεπαφές με βάση το πρόγραμμα περιήγησης που είναι χτισμένες πάνω στο Hive επιτρέπουν στους χρήστες να συνθέτουν και να εκκινούν ερωτήματα Hive (τα οποία με τη σειρά τους εκκινούν εργασίες MapReduce) χρησιμοποιώντας μόνο μερικά κλικ του ποντικιού. Τα παραδοσιακά RDBMS όπως οι βάσεις δεδομένων Oracle και MySQL χρησιμοποιούνται για τη δημοσίευση αυτών των περιλήψεων.

Τα δεδομένα από τη βαθμίδα NFS που αποθηκεύει τα δεδομένα Scribe αναπαράγονται συνεχώς στη συστάδα HDFS μέσω εργασιών αντιγραφής. Οι συσκευές NFS είναι προσαρτημένες στην βαθμίδα Hadoop και οι διεργασίες αντιγραφής εκτελούνται ως εργασίες μόνο χαρτών στη συστάδα Hadoop. Αυτό καθιστά εύκολη την κλιμάκωση των διεργασιών αντιγραφής και τις καθιστά ανθεκτικές σε σφάλματα.

Facebook χρησιμοποιώντας μία μόνο περίπτωση HDFS και η συντρυπτική πλειοψηφία της επεξεργασίας γίνεται σε μία μόνο συστάδα MapReduce (εκτελώντας ένα μόνο jobtracker), με αυτόν τον τρόπο προκειμένου να ελαχιστοποιηθούν τα διοικητικά έξοδα από τη λειτουργία μίας μόνο συστάδας. Επιπλέον, τα δεδομένα δεν χρειάζεται να αναπαράγονται και όλα τα δεδομένα είναι διαθέσιμα σε ένα μόνο μέρος.

Αναφορές

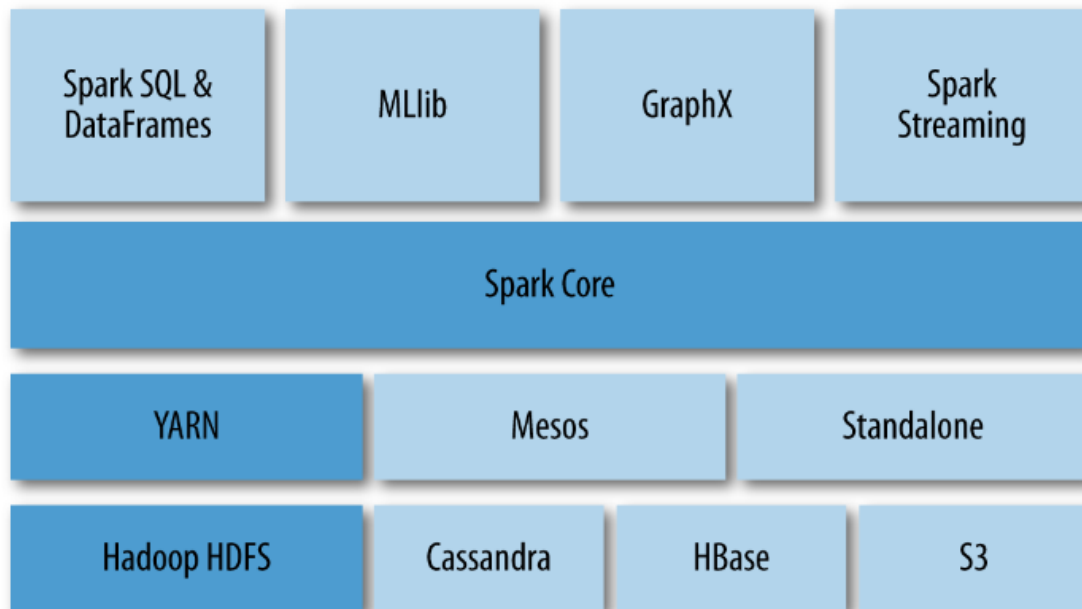
- Marin, I., Shukla, A., & Sarang, V. K. (2019). Big Data Analysis with Python: Combine Spark and Python to unlock the powers of parallel computing and machine learning. Packt Publishing Ltd.
- Erl, T., Khattak, W., & Buhler, P. (2016). Βασικές αρχές μεγάλων δεδομένων: έννοιες, οδηγοί & τεχνικές. Prentice Hall Press.
- <https://hadoop.apache.org>
- Rajaraman, V. (2016). Αναλυτική των μεγάλων δεδομένων. 21(8), 695-716.

Μάθημα 4.5: Μεγάλα δεδομένα Hadoop και Spark

Σημειώσεις διάλεξης

Το Spark βασίζεται στο MapReduce του Hadoop, αλλά επεκτείνει το μοντέλο MapReduce επιτρέποντας περισσότερους τύπους υπολογισμών, οι οποίοι περιλαμβάνουν διαδραστικά ερωτήματα, επεξεργασία ροής και μηχανική μάθηση.

Το Spark δεν είναι μια τροποποιημένη έκδοση του Hadoop, το τελευταίο είναι μόνο ένας τρόπος υλοποίησης του Spark. Σε αυτή την περίπτωση το Spark επωφελείται από τον διαχειριστή συστάδων του Hadoop (YARN) και το επίπεδο αποθήκευσης (HDFS). Ωστόσο, το Spark μπορεί επίσης να διαχωριστεί πλήρως από το Hadoop, ενσωματώνοντας το Cluster Manager και εναλλακτικές πλατφόρμες αποθήκευσης δεδομένων



Τα κύρια συστατικά του Spark είναι:

Το Spark Core περιέχει στοιχεία: για τον χρονοπρογραμματισμό εργασιών, για στοιχεία που αφορούν τη διαχείριση της μνήμης, για την ανάκτηση από αποτυχίες. Επιπλέον, περιέχει τις κύριες δομές αφαίρεσης δεδομένων (RDD).

Το πακέτο Spark SQL είναι αφιερωμένο στη διασύνδεση με δομημένα και ημιδομημένα δεδομένα. Επιτρέπει ερωτήματα σε δεδομένα σε SQL γλώσσα, διαβάζοντας από διάφορες πηγές δεδομένων (πίνακες Hive, Parquet, JSON ...), και επιτρέπει το συνδυασμό ερωτημάτων σε SQL γλώσσα με τον χειρισμό δεδομένων που είναι χαρακτηριστικός των RDD (σε διάφορες γλώσσες, π.χ. Java, Python, Scala).

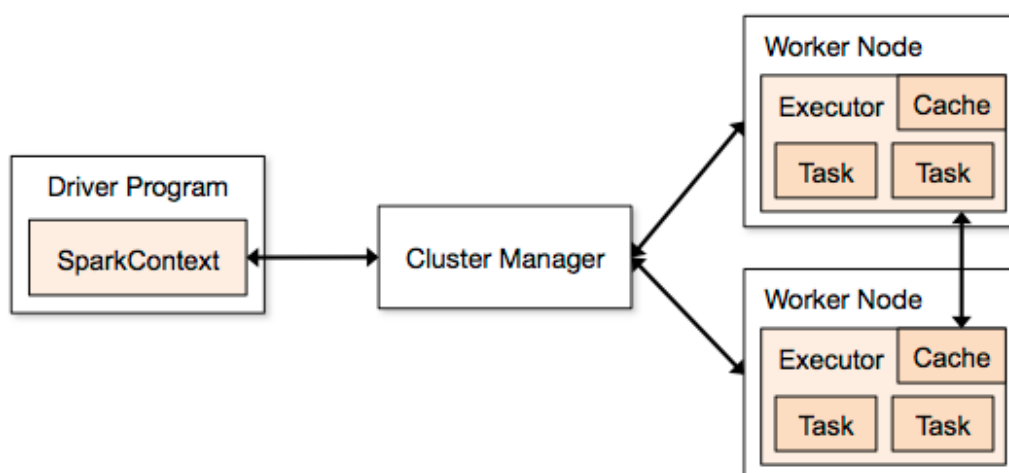
Στοιχείο ροής Spark Streaming για την επεξεργασία συνεχών ροών δεδομένων σε πραγματικό χρόνο Βιβλιοθήκη GraphX για τη διαχείριση γραφημάτων (π.χ. κοινωνικά γραφήματα με βάση τις

φιλίες, σχετικά με τα μηνύματα μεταξύ χρηστών κ.λπ.). Παρέχει υψηλής παραλληλοποίησης υπολογισμούς στον γράφο

MLlib: Βιβλιοθήκη που περιέχει τους πιο συνηθισμένους αλγορίθμους μηχανικής μάθησης (ML) (ταξινόμηση, παλινδρόμηση, ομαδοποίηση ...). Όλες οι μέθοδοι είναι σχεδιασμένες ώστε να μπορούν να διανέμουν τους υπολογισμούς σε συστάδες μηχανών.

Τα ανθεκτικά καταναμημένα σύνολα δεδομένων (RDD) είναι μια θεμελιώδης δομή δεδομένων του Spark. Είναι μια αμετάβλητη καταναμημένη συλλογή από αντικείμενα. Κάθε σύνολο δεδομένων στο RDD χωρίζεται σε λογικά διαμερίσματα, τα οποία μπορούν να εκτελούνται σε διαφορετικούς κόμβους της συστάδας. RDDs μπορούν να περιέχουν οποιονδήποτε τύπο αντικειμένων Python, Java ή Scala, συμπεριλαμβανομένων των κλάσεων που ορίζονται από τον χρήστη. Τυπικά, ένα RDD είναι μια διαμερισματοποιημένη συλλογή εγγραφών μόνο για ανάγνωση. Τα RDDs μπορούν να δημιουργηθούν μέσω ντετερμινιστικών λειτουργιών είτε σε δεδομένα σε σταθερό αποθηκευτικό χώρο είτε σε άλλα RDDs. Η RDD είναι μια ανεκτική σε σφάλματα συλλογή στοιχείων που μπορεί να λειτουργήσει παράλληλα. Υπάρχουν δύο τρόποι για τη δημιουργία RDDs - η παραλληλοποίηση μιας υφιστάμενης συλλογής στο πρόγραμμα οδήγησης ή η αναφορά σε ένα σύνολο δεδομένων σε ένα εξωτερικό σύστημα αποθήκευσης, όπως ένα κοινόχρηστο σύστημα αρχείων, το HDFS, η HBase ή οποιαδήποτε πηγή δεδομένων που προσφέρει ένα Hadoop Input Format. Το Spark κάνει χρήση του έννοια του RDD για την επίτευξη ταχύτερων και αποδοτικότερων λειτουργιών MapReduce.

Το Spark μπορεί να χρησιμοποιηθεί είτε ως διαδραστικό κέλυφος με Scala, Python και R, είτε ως πλατφόρμα υποβολής εργασιών με την εντολή `sparksubmit`. Η μέθοδος `submit` χρησιμοποιείται για την αποστολή εργασιών σε μια συστάδα Spark κωδικοποιημένων σε ένα σενάριο. Η διεπαφή του κελύφους Spark για Python ονομάζεται PySpark. Μπορείτε να έχετε πρόσβαση απευθείας από το τερματικό, όπου η έκδοση Python που είναι η προεπιλεγμένη θα Επιπλέον, μπορεί να γίνει προσπέλαση μέσω του κελύφους IPython ή ακόμη και μέσα σε ένα σημειωματάριο Jupyter.



Μια εφαρμογή Spark εκτελείται ως ένα σύνολο ανεξάρτητων διεργασιών στη συστάδα, οι οποίες συντονίζονται από το αντικείμενο `SparkContext` που περιέχεται στο κύριο πρόγραμμα που ονομάζεται πρόγραμμα οδήγησης.

Το SparkContext πρέπει να συνδεθεί με τον Cluster Manager, ο οποίος έχει ως καθήκον την κατανομή των πόρων. Μόλις συνδεθεί, το SparkContext εκκινεί τους Executors, διεργασίες που είναι υπεύθυνες για την εκτέλεση των λειτουργιών, μέσα στους κόμβους Worker Nodes

Κάθε διεργασία Executor, μία για κάθε πυρήνα επεξεργαστή, είναι στην πραγματικότητα μια εικονική μηχανή Java, στην οποία ο κώδικας του προγράμματος (που περιέχεται σε ένα αρχείο JAR) και οι εργασίες που πρέπει να εκτελέσει αποστέλλονται.

Κατάσταση εκτέλεσης

Ο τρόπος εκτέλεσης της spark καθορίζει πού θα εκτελεστεί το πρόγραμμα οδήγησης, βασικά με τρεις τρόπους:

Τοπική λειτουργία (Local mode): Το Spark τρέχει στον υπολογιστή σας τοπικά, δηλαδή δεν διανέμεται.

Λειτουργία συστάδας (Cluster mode): Το Spark τρέχει σε κατανεμημένη κατάσταση στο cluster. Στο cluster, υπάρχει ένας master και η αριθμός από workers. προγραμματίζει και κατανέμει τους πόρους στο μηχανήμα υποδοχής που αποτελεί τη συστάδα. Το πρόγραμμα οδήγησης εκτελείται σε έναν από τους κόμβους Worker του συμπλέγματος. Τρέχει ως αποκλειστική, αυτόνομη διεργασία μέσα στον Worker.

Λειτουργία πελάτη (Client mode): Το Spark τρέχει σε κατανεμημένη λειτουργία στο σύμπλεγμα. Το πρόγραμμα οδήγησης δεν εκτελείται εντός της συστάδας αλλά στον πελάτη, στο μηχανήμα που εκτελεί την εφαρμογή Spark

Επιπλέον, το Spark υποστηρίζει επί του παρόντος διάφορους διαχειριστές συστάδων:

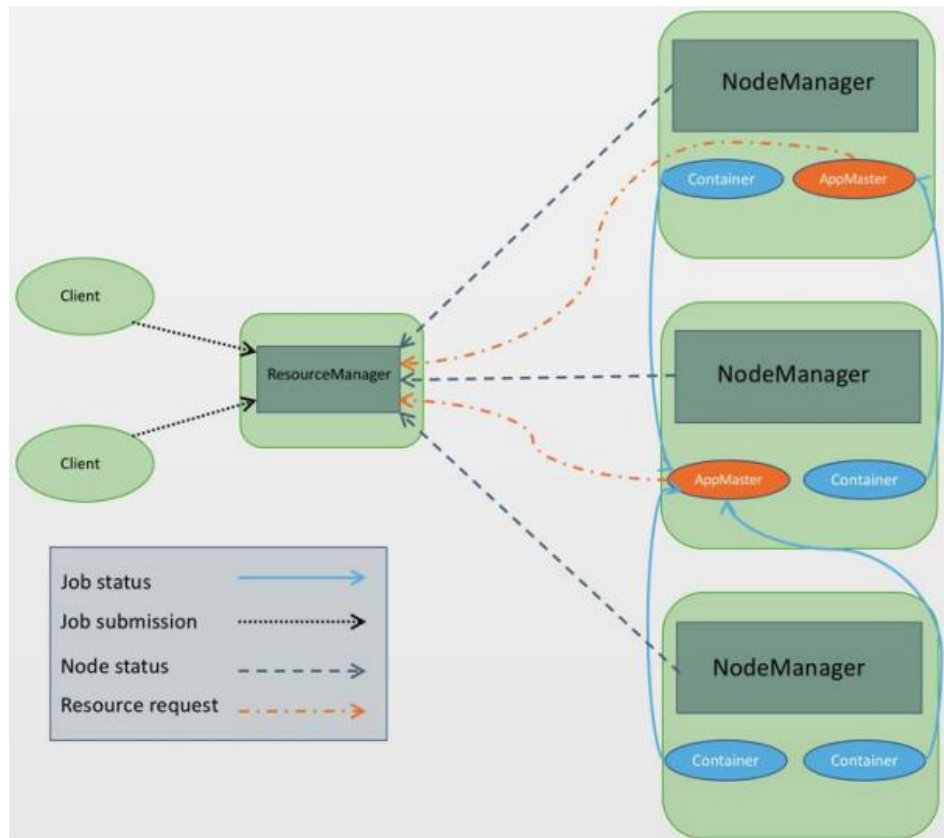
Standalone - ένας απλός διαχειριστής συστάδων που περιλαμβάνεται στο Spark και διευκολύνει τη δημιουργία μιας συστάδας.

Apache Mesos - ένας γενικός διαχειριστής συστάδων που μπορεί επίσης να εκτελεί εφαρμογές Hadoop MapReduce και υπηρεσιών.

Kubernetes - ένα σύστημα ανοικτού κώδικα για την αυτοματοποίηση της ανάπτυξης, της κλιμάκωσης και της διαχείρισης εφαρμογών σε κοντέινερ.

To Hadoop YARN - ο διαχειριστής πόρων στο Hadoop 2, αποτελείται από τρία κύρια στοιχεία:

- **ResourceManager:** κατανέμει τους πόρους του cluster χρησιμοποιώντας έναν Scheduler και έναν ApplicationManager.
- **ApplicationMaster:** διαχειρίζεται τον κύκλο ζωής μιας εργασίας κατευθύνοντας τον NodeManager να δημιουργήσει ή να καταστρέψει ένα container. για μια εργασία. Υπάρχει μόνο ένας ApplicationMaster για μια εργασία.
- **NodeManager:** Διαχειρίζεται τις εργασίες ή τη ροή εργασιών σε έναν συγκεκριμένο κόμβο δημιουργώντας και καταστρέφοντας εμπορευματοκιβώτια σε έναν κόμβο συστάδας.



Διεπαφή του κελύφους Spark

Ένα άλλο χαρακτηριστικό του Spark είναι ότι υποστηρίζει πολλές γλώσσες παρέχοντας API γραμμένα σε Scala, Java, Python ή R.

Η διεπαφή του κελύφους Spark για Python ονομάζεται PySpark. Πρόκειται για ένα API που αναπτύχθηκε σε python για τον προγραμματισμό και τη συγγραφή spark εφαρμογών spark σε στυλ Python, προσφέρει ένα κέλυφος Spark που συνδέει το API της python με τον πυρήνα spark και αρχικοποιεί το Spark Πλαίσιο.

Spark SQL

Το Spark SQL είναι μια ενότητα Spark για την επεξεργασία δομημένων δεδομένων. Σε αντίθεση με το βασικό API του Spark RDD, οι διεπαφές που παρέχονται από το Spark SQL παρέχουν στο Spark περισσότερες πληροφορίες σχετικά με τη δομή τόσο των δεδομένων όσο και των υπολογισμών που εκτελούνται. Εσωτερικά, το Spark SQL χρησιμοποιεί αυτές τις πρόσθετες πληροφορίες για να εκτελεί πρόσθετες βελτιστοποιήσεις. SQL

Μια χρήση του Spark SQL είναι η εκτέλεση ερωτημάτων SQL. Το Spark SQL μπορεί επίσης να χρησιμοποιηθεί για την ανάγνωση δεδομένων από μια υπάρχουσα εγκατάσταση Hive. Για περισσότερες πληροφορίες σχετικά με τον τρόπο διαμόρφωσης αυτής της λειτουργίας, ανατρέξτε

στην ενότητα Hive Tables. Όταν εκτελείτε SQL από ένα άλλο γλώσσα προγραμματισμού, τα αποτελέσματα θα επιστρέφονται ως σύνολο δεδομένων/πλαίσιο δεδομένων. Μπορείτε επίσης να αλληλεπιδράσετε με τη διεπαφή SQL χρησιμοποιώντας τη γραμμή εντολών ή μέσω JDBC/ODBC

Η Spark SQL διαθέτει τις ακόλουθες τέσσερις βιβλιοθήκες που χρησιμοποιούνται για την αλληλεπίδραση με τη σχεσιακή και τη διαδικαστική επεξεργασία:

- **Data Source API** (Application Programming Interface): είναι ένα καθολικό API για τη φόρτωση και την αποθήκευση δομημένων δεδομένων.
- **DataFrame API**: Είναι ισοδύναμο με έναν σχεσιακό πίνακα στην SQL που χρησιμοποιείται για την αποθήκευση δεδομένων σε πίνακες.
- **Διερμηνευτής και βελτιστοποιητής SQL**: Ο βελτιστοποιητής Catalyst Optimizer υλοποιείται στην Spark SQL, η οποία υποστηρίζει τεχνικές βελτιστοποίησης βάσει κανόνων και κόστους. Στη βελτιστοποίηση βάσει κανόνων, έχουμε ορίσει ένα σύνολο κανόνων που θα καθορίσουν τον τρόπο εκτέλεσης του ερωτήματος. Θα ξαναγράψει το υπάρχον ερώτημα με καλύτερο τρόπο για να βελτιώσει την απόδοση.
- **SQL Service**: είναι το σημείο εισόδου για την εργασία μαζί με δομημένα δεδομένα στο Spark. Επιτρέπει τη δημιουργία αντικειμένων DataFrame καθώς και την εκτέλεση ερωτημάτων SQL.

Από την έκδοση 2.0 του Spark, το συνιστώμενο API για τον χειρισμό δεδομένων είναι το API DataFrame. Το API DataFrame είναι κατασκευασμένο πάνω στο RDD API, αν και η πρόσβαση στο RDD API μπορεί να εξακολουθήσει να είναι δυνατή. Το API DataFrame (παρόμοιο με το Pandas DataFrame) αποτελεί μέρος της ενότητας SQL, η οποία λειτουργεί με δομημένα δεδομένα. Αυτή η διεπαφή για τα δεδομένα βοηθά στη δημιουργία επιπλέον βελτιστοποιήσεων, με την ίδια μηχανή εκτέλεσης που χρησιμοποιείται, ανεξάρτητα από το API ή τη γλώσσα που χρησιμοποιείται για την έκφραση αυτών των υπολογισμών. Το διεπαφή για τα Spark DataFrames είναι παρόμοια με τη διεπαφή pandas, τα Spark DataFrames είναι αμετάβλητα, οπότε μετά τη δημιουργία τους, δεν μπορούν να τροποποιηθούν. Επιπλέον, η Spark διαθέτει δύο είδη πράξεων:

- **Μετασχηματισμοί**: Δημιουργία ενός νέου RDD από ένα προηγούμενο RDD, μέσω λειτουργιών όπως αντιστοίχιση, φιλτράρισμα και άλλες.
- **Ενέργειες**: Επιστροφή στο πρόγραμμα οδήγησης ενός αποτελέσματος που υπολογίζεται από ένα RDD ή εγγραφή του στο δίσκο.

Μόνο όταν καλείται μια ενέργεια λαμβάνονται δεδομένα και εκτελούνται όλοι οι μετασχηματισμοί που βρίσκονται στην ουρά (Lazy Evaluation). Που σημαίνει ότι μια εργασία δεν εκτελείται μέχρι να εκτελεστεί μια ενέργεια.

Δεδομένου ότι το Spark Dataframe, όπως είπαμε προηγουμένως, είναι εμπνευσμένο από τη διεπαφή Panda, οι κύριες λειτουργίες χειρισμού δεδομένων μπορούν να εφαρμοστούν, συμπεριλαμβανομένων των συναθροίσεων, των στατιστικών, των υπολογισμών και της οπτικοποίησης σε συναθροισμένα δεδομένα

Το Spark επιτρέπει την ανάγνωση διαφόρων μορφών αρχείων (CSV, JSON, ORC, Parquet, Text) που βρίσκονται σε διαφορετικά συστήματα αποθήκευσης (JDBC, ODBC, Hive, S3, HDFS).

Parquet, μια μορφή αρχείου ανοικτού κώδικα για το Hadoop. Το Parquet αποθηκεύει εμφωλευμένες δομές δεδομένων σε επίπεδη μορφή στήλης. Σε σύγκριση με το με μια

παραδοσιακή προσέγγιση όπου τα δεδομένα αποθηκεύονται σε προσέγγιση προσανατολισμένη σε σειρές, το parquet είναι πιο αποδοτικό από άποψη αποθήκευσης και απόδοσης

Το JSON (JavaScript Object Notation) είναι μια ελαφριά μορφή ανταλλαγής δεδομένων είναι μια μορφή κειμένου που είναι εντελώς ανεξάρτητη από τη γλώσσα, αλλά χρησιμοποιεί συμβολισμούς που είναι οικείοι στους προγραμματιστές της οικογένειας γλωσσών C, όπως η C, η C++, η C#, η Java, η JavaScript, η Perl, η Python και πολλές άλλες. Το JSON βασίζεται σε δύο δομές: Μια συλλογή από ζεύγη ονομάτων/τιμών. Σε διάφορες γλώσσες, αυτό υλοποιείται ως αντικείμενο, εγγραφή, δομή, λεξικό, πίνακας κατακερματισμού, λίστα με κλειδιά ή συσχετιστικός πίνακας- μια διατεταγμένη λίστα τιμών. Στις περισσότερες γλώσσες, αυτό υλοποιείται ως πίνακας, διάνυσμα, λίστα ή ακολουθία.

Το Amazon S3 ή Amazon Simple Storage Service είναι μια υπηρεσία που προσφέρεται από την Amazon Web Services (AWS) και παρέχει αποθήκευση αντικειμένων μέσω μιας διεπαφής web service.

Το Apache Hive είναι ένα έργο λογισμικού αποθήκης δεδομένων που έχει κατασκευαστεί πάνω στο Apache Hadoop για την παροχή ερωτημάτων και ανάλυσης³ δεδομένων. Το Hive παρέχει μια διεπαφή που μοιάζει με SQL για την αναζήτηση δεδομένων που είναι αποθηκευμένα σε διάφορες βάσεις δεδομένων και συστήματα αρχείων που ενσωματώνονται με το Hadoop.

Η μορφή αρχείου Optimized Row Columnar (ORC) παρέχει έναν εξαιρετικά αποδοτικό τρόπο αποθήκευσης δεδομένων Hive

Η Java Database Connectivity (JDBC) είναι μια διεπαφή προγραμματισμού εφαρμογών (API) για τη γλώσσα προγραμματισμού Java, η οποία καθορίζει τον τρόπο με τον οποίο ένας πελάτης μπορεί να έχει πρόσβαση σε μια βάση δεδομένων. Είναι μια τεχνολογία πρόσβασης σε δεδομένα βασισμένη στη Java που χρησιμοποιείται για τη συνδεσιμότητα βάσεων δεδομένων Java

Το ODBC (Open DataBase Connectivity) είναι ένας τυποποιημένος οδηγός σύνδεσης με μια βάση δεδομένων, ο οποίος αναπτύχθηκε από την SQL Group το 1992.

Το σημείο εισόδου για τη χρήση της Spark SQL (άρα του DataFrame API) είναι ένα αντικείμενο που ονομάζεται

SparkSession

```
from pyspark.sql import SparkSession
spark = SparkSession.builder \
    .master("local[1]") \
    .appName("Learning_Spark") \
    .getOrCreate()
```


- `.builder` - παρέχει πρόσβαση στο API Builder το οποίο χρησιμοποιείται για τη διαμόρφωση της συνεδρίας .
- `.master()` - καθορίζει πού θα εκτελεστεί το πρόγραμμα- το `"local[*]"` το θέτει να εκτελεστεί τοπικά σε όλους τους πυρήνες, αλλά μπορείτε να χρησιμοποιήσετε το `"local[1]"` για να να εκτελεστεί σε έναν πυρήνα για παράδειγμα.
- `.appName()` - προαιρετική μέθοδος για την ονομασία της εφαρμογής Spark
- `.getOrCreate()` - παίρνει μια υπάρχουσα `SparkSession` ή δημιουργεί μια νέα αν δεν υπάρχει. Πριν από το Spark 2.0.0 το `sparkContext` χρησιμοποιούνταν ως κανάλι για την πρόσβαση σε όλες τις λειτουργίες του spark. Το πρόγραμμα του οδηγού spark χρησιμοποιεί το `spark context` για να συνδεθεί με τη συστάδα μέσω ενός διαχειριστή πόρων.

Όλες οι λειτουργίες που είναι διαθέσιμες με το `sparkContext` είναι επίσης διαθέσιμες στο `sparkSession`.

Για να χρησιμοποιήσετε τα API της SQL, του HIVE και του Streaming, δεν χρειάζεται να δημιουργήσετε ξεχωριστά πλαίσια, καθώς το `sparkSession` περιλαμβάνει όλα τα APIs.

Μόλις η `SparkSession` ενσωματωθεί, μπορούμε να ρυθμίσουμε τις ιδιότητες ρυθμίσεων εκτέλεσης της Spark

```
spark.conf.set("spark.sql.shuffle.partitions", 6)
spark.conf.set("spark.executor.memory", "2g")
```

Υπάρχουν πολλές διαθέσιμες λειτουργίες σε ένα πλαίσιο δεδομένων, μερικές από αυτές είναι:

Η μέθοδος `createDataFrame: DataFrame`: Δημιουργεί ένα `DataFrame` από ένα RDD, μια λίστα ή ένα `pandas.DataFrame`.

Μέθοδος `Dataframe.printSchema` : Εμφανίζει τις λεπτομέρειες για κάθε στήλη του `DataFrame`.

Μέθοδος `Dataframe.select("name_column")`: επιτρέπει την επιλογή μίας ή περισσότερων στηλών εισάγοντας το όνομα της στήλης ως όρισμα.

Μέθοδος `Dataframe.filter(condition)`: φιλτράρισμα των δεδομένων του πλαισίου δεδομένων με βάση διάφορες συνθήκες

Λειτουργία `Group by` και `Order by`. Η `GroupBy` ουσιαστικά επιστρέφει ομαδοποιημένο σύνολο δεδομένων στο οποίο εκτελούμε αθροίσματα όπως το `count`.

Η `Order by` επιτρέπει την ταξινόμηση των γραμμών σε ένα `DataFrame`, χρησιμοποιώντας μία ή περισσότερες συνθήκες, σε αύξουσα ή φθίνουσα σειρά.

GraphX

Το GraphX είναι ένα νέο συστατικό στο Spark για γράφους και παράλληλους υπολογισμούς με γράφους. Σε υψηλό επίπεδο, το GraphX επεκτείνει το Spark RDD εισάγοντας μια νέα αφαίρεση γράφου: έναν κατευθυνόμενο πολυγράφο με ιδιότητες που συνδέονται με κάθε κορυφή και ακμή. Για την υποστήριξη του υπολογισμού γράφων, το GraphX εκθέτει ένα σύνολο θεμελιωδών τελεστών (π.χ. subgraph, joinVertices, και aggregateMessages), καθώς και μια βελτιστοποιημένη παραλλαγή του API της Pregel. Επιπλέον, το GraphX περιλαμβάνει μια αυξανόμενη συλλογή αλγορίθμων και κατασκευαστών γράφων για την απλούστευση των εργασιών ανάλυσης γράφων.

Το GraphFrames είναι ένα πακέτο για το Apache Spark που παρέχει γραφήματα βασισμένα σε DataFrame. Παρέχει API υψηλού επιπέδου στη Scala, Java και Python. Στόχος του είναι να παρέχει τόσο τη λειτουργικότητα του GraphX όσο και την εκτεταμένη λειτουργικότητα που εκμεταλλεύεται το πλεονέκτημα του Spark DataFrames. Αυτή η εκτεταμένη λειτουργικότητα περιλαμβάνει εύρεση μοτίβων, σειριοποίηση βασισμένη σε DataFrame και εξαιρετικά εκφραστικό γράφο ερωτήματα.

Το GraphX είναι για τα RDD όπως είναι τα GraphFrames για τα DataFrames.

Τα GraphFrames αναπαριστούν γράφους: κορυφές (π.χ. χρήστες) και ακμές (π.χ. σχέσεις μεταξύ χρηστών). Εάν είστε εξοικειωμένοι με το GraphX, τότε το GraphFrames θα είναι εύκολο να το μάθετε. Η βασική διαφορά είναι ότι τα GraphFrames βασίζονται στο Spark DataFrames, και όχι σε RDDs.

Τα GraphFrames παρέχουν επίσης ισχυρά εργαλεία για την εκτέλεση ερωτημάτων και τυποποιημένων αλγορίθμων γράφων. Με τα GraphFrames, μπορείτε να εύκολα να αναζητήσετε μοτίβα σε γράφους, να βρείτε σημαντικές κορυφές και πολλά άλλα. Ανατρέξτε στον Οδηγό χρήσης για έναν πλήρη κατάλογο ερωτημάτων και αλγορίθμων.

Έτσι ξεκινάμε το GraphX στο PySpark

```
pyspark -packages graphframes:graphframes:0.5.0-spark2.1-s_2.11
```

Περιπτώσεις χρήσης

ΕΤΑΙΡΕΙΑ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Μια εταιρεία τηλεπικοινωνιών έχει μια κατεστραμμένη βάση δεδομένων που πρέπει να καθαριστεί. Ένα μέρος αυτού του συνόλου δεδομένων είναι το ακόλουθο. Πριν από την εισαγωγή των δεδομένων στο πληροφοριακό σύστημα της εταιρείας είναι απαραίτητο να καθαριστούν για να αποφευχθούν λανθασμένες πληροφορίες ή πληροφορίες που λείπουν.

('Mario',3,170, 56, 24, 'M', None,1),

```
("Massimo",7,167, 64, 45, "M", 265000,2),
(None, None,8 , 52, None, None, None, None,1),
("Luca",22,172, 69, 53, "M", None,1),
("Silia",32,176, 72, 54, "F", 380000,1),
('Lucia',25,158, 55, 25, 'F', None,3),
("Carlo",18,180, 74, 58, "M", 386000,1),
("Francesco",10,172, 69, 43, "M", 215000,2),
("Andrea",29,170, 58, 28, "M", 185000,3),
("Massimiliano",24,187, 62, 45, "M", 215000,2),
("Silia",32,176, 72, 54, "F", 380000,1),
("Lucia",27,128, 58, 32, "F", 210000,2),
('Carlo Maria',103,180, 74,55, 'Me', 376000,1),
('Francesco Cristiano',25,172, 69, 48, 'M', 230000,2),
('Stefano',12,170, 56, 28, 'M', 170000,3),
('Gilberto',15,168, 64, 45, 'M',None,2]],
['Name','ID_Surname','months_work','contracts_signed','age','gender','income','contract_level']])
```

Βήμα 1: Δημιουργία ενός DataFrame

```
df = spark.createDataFrame([('Mario',3,170, 56, 24, 'M', None,1),
                             ('Massimo',7,167, 64, 45, 'M', 265000,2),
                             (None, None,8 , 52, None, None, None,1),
                             ('Luca',22,172, 69, 53, 'M', None,1),
                             ('Silia',32,176, 72, 54, 'F', 380000,1),
                             ('Lucia',25,158, 55, 25, 'F', None,3),
                             ('Carlo',18,180, 74, 58,'M', 386000,1),
                             ('Francesco',10,172, 69, 43, 'M', 215000,2),
                             ('Andrea',29,170, 58, 28, 'M', 185000,3),
                             ('Massimiliano',24,187, 62, 45, 'M', 215000,2),
                             ('Silia',32,176, 72, 54, 'F', 380000,1),
                             ('Lucia',27,128, 58, 32, 'F', 210000,2),
                             ('Carlo Maria',103,180, 74,55, 'Me', 376000,1),
                             ('Francesco Cristiano',25,172, 69, 48, 'M', 230000,2),
                             ('Stefano',12,170, 56, 28,'M', 170000,3),
                             ('Gilberto',15,168, 64, 45, 'M',None,2)],
                             ['Name','ID_Surname','months_work', 'contracts_signed', 'age', 'gender',
                             'income','contract_level'])
```

Βήμα 2: Καταμέτρηση αντιγράφων

```
print('Number of Row: {0}'.format(df.count()))
```

```
print('Number of non-duplicated cases: {0}'.format(df.distinct().count()))
```

Βήμα 3: Αφαίρεση αντιγράφων και έλεγχος ότι η λειτουργία έχει πραγματοποιηθεί

```
df = df.dropDuplicates()
```

```
df.show()
```

```
df.count()
```

Βήμα 4: Υπάρχει ένα ανορθόγραφο σαν Me. Αντικαταστήστε το με M

```
df = df.replace('Me','M')
```

```
df.show()
```

Βήμα: 5 Έλεγχος μηδενικών τιμών

Αυτή η συνάρτηση χρησιμοποιείται για την εκτύπωση χαρακτηριστικών με μηδενικές τιμές και μηδενικό αριθμό

```
def null_value_count(df):
```

```
    null_columns_counts = []
```

```
    numRows = df.count()
```

```
    for k in df.columns:
```

```
        nullRows = df.where(col(k).isNull()).count()
```

```
        if(nullRows > 0):
```

```
            temp = k,nullRows
```

```
null_columns_counts.append(temp)

return(null_columns_counts)

# Κλήση της συνάρτησης και εμφάνιση του αποτελέσματος

null_columns_count_list = null_value_count(df)

spark.createDataFrame(null_columns_count_list, ['Column_With_Null_Value',
'Null_Values_Count']).show()
```

Βήμα 6: Διαγραφή γραμμών με ελλείπουσα τιμή

Η δομή της συνάρτησης drop () έχει ως εξής:

```
DataFrame.na.drop (how = 'any', thresh = None, subset = None)
```

εάν 'any' διαγράφει κάθε γραμμή που έχει επίσης μόνο ένα δεδομένο που λείπει,

εάν 'all' διαγράφει μια γραμμή μόνο αν λείπουν όλα τα δεδομένα.

```
df.na.drop(how='any').show()
```

```
df.na.drop(how='all').show()
```

Μπορούμε να αποσύρουμε μια μόνο στήλη, για παράδειγμα να αποσύρουμε τη γραμμή με ελλείπουσα τιμή στη στήλη "name":

```
df=df.na.drop(subset=['Name'])
```

Βήμα 7: Επανάληψη των τιμών που λείπουν από τη στήλη "εισόδημα" με τον αριθμό μηδέν

```
df=df2.na.fill(0)
```

Βήμα 8: Για να αντικαταστήσουμε αυτούς τους μηδενικούς αριθμούς στη στήλη "εισόδημα", μπορούμε να τους αποδώσουμε το μέσο εισόδημα σε σχέση με το επίπεδο σύμβασης

```
# Ας ελέγξουμε το μέσο εισόδημα ανά 'contract_level'
```

```
df.groupby('contract_level').avg('Income').collect()
```

Let's impute missing values (= 0) in income column with averages compute before

```
df = df.withColumn('income',when((df['contract_level'] == 1) &
(df['income']==0),228400).otherwise(df["Income"]))
```

```
df= df.withColumn('income',when((df['contract_level'] == 2) &
(df['income']==0),189166).otherwise(df["Income"]))
```

```
df = df.withColumn('income',when((df['contract_level'] == 3) &
(df['income']==0),118333).otherwise(df["Income"]))
```

Name	ID_Surname	months_work	contracts_signed	age	gender	income	contract_level
Luca	22	172	69	53	M	228400	1
Massimo	7	167	64	45	M	265000	2
Lucia	27	128	58	32	F	210000	2
Carlo Maria	103	180	74	55	M	376000	1
Francesco Cristiano	25	172	69	48	M	230000	2
Andrea	29	170	58	28	M	185000	3
Stefano	12	170	56	28	M	170000	3
Gilberto	15	168	64	45	M	189166	2
Carlo	18	180	74	58	M	386000	1
Massimiliano	24	187	62	45	M	215000	2
Francesco	10	172	69	43	M	215000	2
Silia	32	176	72	54	F	380000	1
Mario	3	170	56	24	M	228400	1
Lucia	25	158	55	25	F	118333	3

Βήμα 9: Υπάρχει ένα ανορθόγραφο σαν Me. Αντικαταστήστε το με M

```
df = df.replace('Me','M')
```

Βήμα 10: Εμφάνιση αποτελεσμάτων

```
df.show()
```

Αναφορές

- Marin, I., Shukla, A., & Sarang, V. K. (2019). Ανάλυση μεγάλων δεδομένων με Python : Συνδυάστε το Spark και την Python για να ξεκλειδώσετε τις δυνάμεις του παράλληλου υπολογισμού και της μηχανικής μάθησης. Packt Publishing Ltd.
- Chambers, B., & Zaharia, M.(2018). Spark : Ο οριστικός οδηγός : Επεξεργασία μεγάλων δεδομένων με απλό τρόπο " O'Reilly Media,Inc."
- <http://spark.apache.org/docs/latest/cluster-overview.html>
- https://www.tutorialspoint.com/apache_spark/apache_spark_rdd.htm
- <http://spark.apache.org/docs/2.2.0/sql-programming-guide.html>

Μάθημα 4.6: Μεγάλα δεδομένα Ανάλυση, Οπτικοποίηση, Εφαρμογές

Θέματα προς συζήτηση

- Αρχιτεκτονική Spark
- Συστατικά Spark
- Λειτουργία εκτέλεσης Spark και Cluster Manager
- Λειτουργία πλαισίου δεδομένων στο Spark
- Χειρισμός διεργασίας σε Spark DataFrame

Σημειώσεις διάλεξης

Όταν μιλάμε για Μεγάλα δεδομένα (Big Data), αναφερόμαστε σε έναν συνδυασμό δεδομένων, περισσότερο ή λιγότερο δομημένων, που συλλέγονται από μια εταιρεία και από τα οποία μπορούν να εξαχθούν πληροφορίες για χρήση σε έργα μηχανικής μάθησης, προγνωστικής ανάλυσης και άλλων αναπτυγμένων αναλύσεων εφαρμογές.

Για την ανάλυση αυτού του τεράστιου όγκου δεδομένων είναι απαραίτητο να βασιστούμε στο Cloud Computing. Αντί για την αποθήκευση αρχείων σε έναν ιδιόκτητο σκληρό δίσκο ή σε τοπική συσκευή αποθήκευσης, η αρχειοθέτηση μέσω Cloud Computing σας επιτρέπει να τα αποθηκεύσετε σε μια βάση δεδομένων ή σε απομακρυσμένα μέσα: εφόσον μια ηλεκτρονική συσκευή έχει πρόσβαση στο διαδίκτυο, έχει πρόσβαση σε δεδομένα και προγράμματα λογισμικού για την εκτέλεσή τους.

Ταυτόχρονα, οι χρήστες μπορούν να αποθηκεύουν αρχεία και εφαρμογές σε απομακρυσμένους διακομιστές και να έχουν πρόσβαση σε όλα τα δεδομένα μέσω του Διαδικτύου, ανεξαρτήτως της φυσικής θέσης στην οποία βρίσκεται ο χρήστης. Έτσι, ενώ τα Big Data αναφέρονται στον όγκο των δεδομένων, το Cloud Computing σημαίνει τον τρόπο με τον οποίο αποθηκεύονται και επεξεργάζονται αυτά τα δεδομένα.

ΥΠΟΔΟΜΗ ΩΣ ΥΠΗΡΕΣΙΑ (IAAS) INFRASTRUCTURE AS A SERVICE

Εδώ ο πάροχος υπηρεσιών προσφέρει ολόκληρη την υποδομή καθώς και τις σχετικές δραστηριότητες συντήρησης.

ΠΛΑΤΦΟΡΜΑ ΩΣ ΥΠΗΡΕΣΙΑ (PAAS) PLATFORM AS A SERVICE

Σε αυτή την υπηρεσία, ο πάροχος Cloud προσφέρει πόρους όπως αποθήκευση, χρόνο εκτέλεσης, ουρά αναμονής, βάσεις δεδομένων κ.λπ. Ωστόσο, η ευθύνη για τις δραστηριότητες διαμόρφωσης και υλοποίησης εξαρτάται από τον καταναλωτή.

ΛΟΓΙΣΜΙΚΟ ΩΣ ΥΠΗΡΕΣΙΑ (SAAS) SOFTWARE AS A SERVICE

Αυτή η υπηρεσία είναι η πλέον διευκολυνόμενη, διότι παρέχει όλες τις ρυθμίσεις και τις υποδομές που είναι απαραίτητες για τις ανάγκες του χρήστη. Οι σχέσεις μεταξύ των μεγάλων δεδομένων και του υπολογιστικού νέφους μπορούν να ταξινομηθούν ανάλογα με τους τύπους υπηρεσιών:

IAAS ΣΤΟ ΔΗΜΟΣΙΟ ΝΕΦΟΣ (IAAS IN PUBLIC CLOUD)

Το IaaS είναι μια φθηνή λύση και, χρησιμοποιώντας αυτή την υπηρεσία νέφους, οι υπηρεσίες Big Data επιτρέπουν στους ανθρώπους να έχουν πρόσβαση σε απεριόριστο αποθηκευτικό χώρο και υπολογιστική ισχύ. Είναι μια πολύ βολική λύση για τις εταιρείες, επειδή ο πάροχος υπηρεσιών νέφους φέρει όλη τη διαχείριση του υποκείμενου υλικού.

PAAS ΣΤΟ ΙΔΙΩΤΙΚΟ ΝΕΦΟΣ (PAAS IN PRIVATE CLOUD)

Οι πωλητές PaaS ενσωματώνουν τεχνολογίες Big Data στις υπηρεσίες τους, επομένως, εξαλείφουν την ανάγκη διαχείρισης των περιπλοκών της διαχείρισης μεμονωμένων στοιχείων λογισμικού και υλικού, η οποία αποτελεί πραγματική ανησυχία όταν πρόκειται για terabytes δεδομένων.

SAAS ΣΤΟ ΥΒΡΙΔΙΚΟ ΝΕΦΟΣ (SAAS IN HYBRID CLOUD)

Το υβριδικό νέφος είναι μια εξαιρετική λύση για τη διεξαγωγή της ανάλυσης των δεδομένων των μέσων κοινωνικής δικτύωσης, η οποία, σήμερα, αποτελεί βασικό παράμετρος για τις εταιρείες

Υπάρχουν πολλαπλά οφέλη από την ανάλυση μεγάλων δεδομένων στο Cloud.

Βελτιωμένη ανάλυση

Με την πρόοδο της τεχνολογίας Cloud, η ανάλυση μεγάλων δεδομένων έχει γίνει πιο βελτιωμένη προκαλώντας καλύτερα αποτελέσματα. Ως εκ τούτου, οι εταιρείες προτιμούν να εκτελούν ανάλυση μεγάλων δεδομένων στο Cloud. Επιπλέον, το Cloud βοηθά στην ενσωμάτωση δεδομένων από πολυάριθμες πηγές.

Απλοποιημένη υποδομή

Η ανάλυση μεγάλων δεδομένων είναι μια τρομερά επίπονη εργασία για την υποδομή, καθώς τα δεδομένα έρχονται σε μεγάλους όγκους με διαφορετικές ταχύτητες, και τύπους, με τα οποία οι παραδοσιακές υποδομές συνήθως δεν μπορούν να αντεπεξέλθουν. Καθώς το υπολογιστικό νέφος παρέχει ευέλικτη υποδομή, την οποία μπορούμε να κλιμακώσουμε ανάλογα με τις εκάστοτε ανάγκες, είναι εύκολη η διαχείριση των φόρτων εργασίας.

Μείωση του κόστους

Τόσο τα μεγάλα δεδομένα όσο και η τεχνολογία Cloud προσφέρουν αξία στους οργανισμούς μειώνοντας την ιδιοκτησία. Το μοντέλο πληρωμής ανά χρήστη του Cloud μετατρέπει το CAPEX σε OPEX. Από την άλλη πλευρά, το Apache μειώνει το κόστος αδειοδότησης των μεγάλων δεδομένων, το οποίο υποτίθεται ότι είναι κοστίζει εκατομμύρια για την κατασκευή και την αγορά. Το Cloud επιτρέπει στους πελάτες την επεξεργασία μεγάλων δεδομένων χωρίς πόρους μεγάλης κλίμακας μεγάλων δεδομένων. Ως εκ τούτου, τόσο τα μεγάλα δεδομένα όσο και η τεχνολογία Cloud μειώνουν το κόστος για εταιρικούς σκοπούς και προσδίδουν αξία στην επιχείρηση.

Ασφάλεια και προστασία της ιδιωτικής ζωής

Η ασφάλεια των δεδομένων και η προστασία της ιδιωτικής ζωής είναι δύο σημαντικές ανησυχίες όταν πρόκειται για επιχειρηματικά δεδομένα. Επιπλέον, όταν η εφαρμογή σας είναι φιλοξενείται σε μια πλατφόρμα Cloud, λόγω του ανοικτού περιβάλλοντος και του περιορισμένου ελέγχου των χρηστών, η ασφάλεια αποτελεί πρωταρχικό μέλημα. Στο από την άλλη πλευρά, ως εφαρμογή ανοικτού κώδικα, η λύση μεγάλων δεδομένων όπως το Hadoop χρησιμοποιεί πολλές υπηρεσίες και υποδομές τρίτων. Ως εκ τούτου, στις μέρες μας, οι ολοκληρωτές συστημάτων φέρνουν σε λύση Private Cloud που είναι ελαστική και επεκτάσιμη. Επιπλέον, αξιοποιεί επίσης Κλιμακούμενη κατανεμημένη επεξεργασία.

Εκτός αυτού, τα δεδομένα νέφους αποθηκεύονται και υποβάλλονται σε επεξεργασία σε μια κεντρική τοποθεσία, γνωστή ως διακομιστής αποθήκευσης νέφους. Μαζί με ο πάροχος υπηρεσιών και ο πελάτης υπογράφουν μια συμφωνία επιπέδου υπηρεσιών (SLA) για να κερδίσουν την εμπιστοσύνη μεταξύ τους. Εάν απαιτείται η ο πάροχος αξιοποιεί επίσης το απαιτούμενο προηγμένο επίπεδο ελέγχου ασφαλείας. Αυτό επιτρέπει την ασφάλεια των μεγάλων δεδομένων στο υπολογιστικό νέφος. καλύπτοντας τα ακόλουθα ζητήματα: Πώς οι πάροχοι υπηρεσιών νέφους διατηρούν την αποθήκευση και τα δεδομένα.

Εικονικοποίηση (Virtualization)

Η υποδομή παίζει καθοριστικό ρόλο για την υποστήριξη κάθε εφαρμογής. Η τεχνολογία εικονικοποίησης είναι η ιδανική πλατφόρμα για μεγάλα δεδομένα. Οι εικονικοποιημένες εφαρμογές μεγάλων δεδομένων, όπως το Hadoop, παρέχουν πολλαπλά οφέλη τα οποία δεν είναι προσβάσιμα στη φυσική υποδομή, αλλά απλοποιεί τη διαχείριση των μεγάλων δεδομένων. Τα μεγάλα δεδομένα και η υπολογιστική νέφους δείχνουν τη σύγκλιση διαφόρων τεχνολογιών και τάσεων που καθιστά την υποδομή ΤΠ και τις σχετικές εφαρμογές πιο δυναμικές, πιο αναλώσιμες και πιο αρθρωτές και. Ως εκ τούτου, τα μεγάλα δεδομένα και τα έργα υπολογιστικού νέφους βασίζονται σε μεγάλο βαθμό στην εικονικοποίηση

Οι πιο δημοφιλείς πλατφόρμες υπολογιστικού νέφους είναι οι Amazon Web Services (AWS), Microsoft Azure και Google Cloud Platform (GCP).

Αυτοί οι πάροχοι υπολογιστικού νέφους προσφέρουν επί του παρόντος βολικές κατά παραγγελία διαχειριζόμενες συστάδες μεγάλων δεδομένων (PaaS) με μοντέλο pay-as-you-go. Στο PaaS, οι αναλυτικές μηχανές, όπως το Spark και το Hive, έρχονται έτοιμες προς χρήση, με διαμόρφωση και αναβάθμιση γενικού σκοπού διαχείρισης.

Σε αυτό το μάθημα θα εργαστούμε και θα παρουσιάσουμε το Databricks, μια διαχειρίσιμη πλατφόρμα για την εκτέλεση του Apache Spark, που σημαίνει ότι δεν χρειάζεται να μάθετε πολύπλοκες έννοιες διαχείρισης συστάδων ούτε να εκτελείτε κουραστικές εργασίες συντήρησης για να επωφεληθείτε από το Spark. Το Databricks παρέχει επίσης ένα πλήθος χαρακτηριστικών που βοηθούν τους χρήστες του να είναι πιο παραγωγικοί με το Spark. Πρόκειται για μια πλατφόρμα point and click για όσους προτιμούν ένα περιβάλλον εργασίας χρήστη, όπως οι επιστήμονες δεδομένων ή οι αναλυτές δεδομένων. Επιπλέον, το Databricks προσφέρει την απαραίτητη κλίμακα και τις επιδόσεις του νέφους, συμπεριλαμβανομένης της διαλειτουργικότητας με ηγέτες όπως οι AWS και Azure

Το Databricks έχει βασικές έννοιες που αξίζει να κατανοήσετε:

Οι χώροι εργασίας (Workspaces) σας επιτρέπουν να οργανώνετε όλες τις εργασίες που εκτελείτε στα Databricks. Όπως μια δομή φακέλων στον υπολογιστή σας, είναι σας επιτρέπει να αποθηκεύετε σημειωματάρια και βιβλιοθήκες και να τα μοιράζεστε με άλλους χρήστες. Οι χώροι εργασίας δεν συνδέονται με δεδομένα και δεν πρέπει να χρησιμοποιούνται για την αποθήκευση δεδομένων. Είναι απλώς για να αποθηκεύετε τα σημειωματάρια και τις βιβλιοθήκες που χρησιμοποιείτε για να λειτουργείτε και να χειρίζεστε τα δεδομένα σας.

Τα σημειωματάρια (Notebooks) είναι ένα σύνολο οποιουδήποτε αριθμού κελιών που σας επιτρέπουν να εκτελείτε εντολές. Τα κελιά περιέχουν κώδικα σε οποιοδήποτε από τα ακόλουθα γλώσσες: Scala, Python, R, SQL ή Markdown.

Τα σημειωματάρια πρέπει να είναι συνδεδεμένα σε μια **συστάδα (cluster)** για να μπορούν να εκτελούν εντολές, ωστόσο δεν είναι μόνιμα συνδεδεμένα με μια συστάδα. Αυτό επιτρέπει την κοινή χρήση των σημειωματάρων μέσω του διαδικτύου ή τη λήψη τους στον τοπικό σας υπολογιστή.

Οι πίνακες ελέγχου (Dashboards) μπορούν να δημιουργηθούν από τα **σημειωματάρια** ως ένας τρόπος εμφάνισης της εξόδου των κελιών χωρίς τον κώδικα που τα παράγει.

Τα σημειωματάρια μπορούν επίσης να προγραμματιστούν ως **εργασίες (jobs)** με ένα κλικ, είτε για την εκτέλεση ενός αγωγού δεδομένων, είτε για την ενημέρωση ενός μοντέλου μηχανικής μάθησης, ή να ενημερώσετε ένα ταμπλό.

Οι βιβλιοθήκες (Libraries) είναι πακέτα ή ενότητες που παρέχουν πρόσθετη λειτουργικότητα. Αυτές μπορεί να είναι ειδικά γραμμένες βάζα Scala ή Java, αυγά python ή ειδικά γραμμένα πακέτα. Μπορείτε να τα γράψετε και να τα φορτώσετε χειροκίνητα ή να τα εγκαταστήσετε απευθείας μέσω του πακέτου βοηθητικά προγράμματα διαχείρισης όπως το rpy2 ή το maven.

Οι πίνακες (Libraries) είναι δομημένα δεδομένα που εσείς και η ομάδα σας θα χρησιμοποιήσετε για ανάλυση. Οι πίνακες μπορούν να υπάρχουν σε διάφορα μέρη. Οι πίνακες μπορεί να είναι αποθηκευτούν στο Amazon S3 ή στο Azure Blob Storage, μπορούν να αποθηκευτούν στο cluster που χρησιμοποιείτε επί του παρόντος ή μπορούν να είναι αποθηκευτούν στη μνήμη.

Οι συστάδες (Clusters) είναι ομάδες υπολογιστών τις οποίες αντιμετωπίζετε ως έναν ενιαίο υπολογιστή. Στο Databricks, αυτό σημαίνει ότι μπορείτε ουσιαστικά να αντιμετωπίζετε 20 υπολογιστές όπως θα αντιμετωπίζατε έναν υπολογιστή. Οι συστάδες σας επιτρέπουν να εκτελείτε κώδικα από σημειωματάρια ή βιβλιοθήκες σε σύνολο δεδομένων. Τα δεδομένα αυτά μπορεί να είναι ακατέργαστα δεδομένα που βρίσκονται στο S3 ή δομημένα δεδομένα που έχετε μεταφορτώσει ως πίνακα στη συστάδα στην οποία εργάζεστε.

Είναι σημαντικό να σημειωθεί ότι οι συστάδες διαθέτουν ελέγχους πρόσβασης για τον έλεγχο του ποιος έχει πρόσβαση σε κάθε συστάδα.

Οι εργασίες (Jobs) είναι το εργαλείο με το οποίο μπορείτε να προγραμματίσετε την εκτέλεση να συμβεί είτε σε μια ήδη υπάρχουσα συστάδα είτε σε μια δική της συστάδα. Αυτές μπορεί να είναι σημειωματάρια καθώς και jars ή σενάρια rpython. Μπορούν να δημιουργηθούν είτε χειροκίνητα είτε μέσω του REST API.

Οι εφαρμογές (Apps) είναι ενσωματώσεις τρίτων μερών με την πλατφόρμα Databricks. Σε αυτές περιλαμβάνονται εφαρμογές όπως το Tableau.

Για φόρτους εργασίας μηχανικής μάθησης, η Databricks παρέχει το Databricks Runtime for Machine Learning (Databricks Runtime ML), ένα έτοιμο περιβάλλον για μηχανική μάθηση και επιστήμη δεδομένων. Περιέχει πολλές δημοφιλείς βιβλιοθήκες, συμπεριλαμβανομένης της Spark ML, TensorFlow, PyTorch, Keras και XGBoost. Υποστηρίζει επίσης κατανεμημένη εκπαίδευση με τη χρήση του Horovod.

Σε αυτό το μάθημα θα επικεντρωθούμε στο Spark ML στην έκδοση Databricks Community Edition.

Το Databricks Community Edition έχει σχεδιαστεί για προγραμματιστές, επιστήμονες δεδομένων, μηχανικούς δεδομένων και όλους όσους θέλουν να μάθουν το Spark. Οι χρήστες θα έχουν πρόσβαση σε ένα micro-cluster, έναν διαχειριστή cluster και το περιβάλλον σημειωματάριου για την πρωτοτυποποίηση απλών εφαρμογών.

Για να ρυθμίσετε έναν λογαριασμό Databricks, μεταβείτε στη διεύθυνση <https://databricks.com/try-databricks> και επιλέξτε τη δωρεάν Community Edition για να ανοίξετε το λογαριασμό σας. Αυτή η επιλογή διαθέτει ένα μόνο cluster με έως και 6 GB δωρεάν αποθηκευτικό χώρο. Σας επιτρέπει να δημιουργήσετε ένα βασικό Notebook. Θα χρειαστείτε μια έγκυρη διεύθυνση ηλεκτρονικού ταχυδρομείου για την επαλήθευση του λογαριασμού σας.

Το Pipeline API μηχανικής μάθησης εισήχθη στην έκδοση 1.2 του πλαισίου Apache Spark. Παρέχει το API για τους προγραμματιστές να δημιουργούν και να εκτελούν σύνθετες ροές εργασίας ML. Ο στόχος του Pipeline API είναι να επιτρέπει στους χρήστες να συγκεντρώνουν γρήγορα και εύκολα και να διαμορφώνουν πρακτικές κατανεμημένων Pipelines μηχανικής μάθησης τυποποιώντας τα API για διαφορετικές έννοιες μηχανικής μάθησης.

Το Pipeline API είναι διαθέσιμο στο πακέτο `org.apache.spark.ml`.

Το Spark ML βοηθά επίσης στο συνδυασμό πολλαπλών αλγορίθμων μηχανικής μάθησης σε έναν ενιαίο αγωγό. Το API μηχανικής μάθησης του Spark χωρίζεται σε δύο πακέτα που ονομάζονται `spark.mllib` και `spark.ml`. Το πακέτο `spark.mllib` περιέχει το αρχικό API που βασίζεται σε RDDs. Από την άλλη πλευρά, το πακέτο `spark.ml` παρέχει API υψηλότερου επιπέδου που είναι χτισμένο πάνω σε DataFrames για την κατασκευή pipelines ML.

Ο αγωγός δεδομένων (data pipeline) μηχανικής μάθησης αποτελείται από διάφορα στοιχεία για την εκτέλεση της ανάλυσης δεδομένων. Τα βασικά συστατικά στοιχεία ενός data pipeline παρατίθενται κατωτέρω:

Σύνολα δεδομένων(Datasets): DataFrames χρησιμοποιούνται για την αναπαράσταση συνόλων δεδομένων στον ML pipeline. Επιτρέπει την αποθήκευση δομημένων δεδομένων σε ονομαστικές στήλες.

Οι στήλες μπορούν να χρησιμοποιηθούν για την αποθήκευση κειμένου, διανυσμάτων χαρακτηριστικών, πραγματικών ετικετών και προβλέψεων.

Αγωγοί (Pipelines): Οι ροές εργασίας ML μοντελοποιούνται ως Pipelines, οι οποίοι αποτελούνται από μια ακολουθία σταδίων. Κάθε στάδιο μετασχηματίζει δεδομένα εισόδου για την παραγωγή εξόδου για τα επόμενα στάδια. Ένας αγωγός συνδέει πολλαπλούς μετασχηματιστές και εκτιμητές μαζί για να καθορίσει ένα ML workflow. Ένας αγωγός ορίζεται ως μια ακολουθία σταδίων και κάθε στάδιο είναι είτε ένας μετασχηματιστής είτε ένας εκτιμητής. Αυτά τα στάδια εκτελούνται με τη σειρά και το πλαίσιο δεδομένων εισόδου μετασχηματίζεται καθώς διέρχεται από κάθε στάδιο. Για τα στάδια Transformer, καλείται η μέθοδος transform() στο DataFrame. Για τα στάδια Estimator, καλείται η μέθοδος fit() για την παραγωγή ενός Transformer (ο οποίος γίνεται μέρος του PipelineModel, ή προσαρμοσμένου Pipeline), και η μέθοδος transform() αυτού του Transformer καλείται στο DataFrame.

Στάδια αγωγών (Pipelines): Ορίζουμε δύο είδη σταδίων:

Μετασχηματιστής: DataFrame σε ένα άλλο DataFrame (παράδειγμα: ML είναι ένα μετασχηματιστής που μετατρέπει ένα DF με χαρακτηριστικά σε ένα DF με προβλέψεις).

Ένας μετασχηματιστής μετατρέπει ένα DataFrame σε ένα άλλο DataFrame με ένα ή περισσότερα προστιθέμενα χαρακτηριστικά σε αυτό. Για παράδειγμα, στο πακέτο Spark ML, ο OneHotEncoder μετασχηματίζει μια στήλη με δείκτη ετικέτας σε μια στήλη με διανυσματικά χαρακτηριστικά. Κάθε Transformer έχει μια μέθοδο transform() η οποία καλείται για να μετατρέψει ένα DataFrame σε ένα άλλο.

Εκτιμητής: Ο Estimator είναι ένας αλγόριθμος μηχανικής μάθησης που μαθαίνει από τα δεδομένα που παρέχονται. Η είσοδος σε έναν εκτιμητή είναι ένα DataFrame και η έξοδος είναι ένας Transformer. Ένας Estimator χρησιμοποιείται για την εκπαίδευση του μοντέλου. Παράγει έναν μετασχηματιστή. Για το παράδειγμα, ένας εκτιμητής LogisticRegression παράγει έναν μετασχηματιστή LogisticRegressionModel. Ένα άλλο παράδειγμα είναι ο K Means, καθώς ένας εκτιμητής δέχεται ένα πλαίσιο δεδομένων εκπαίδευσης και παράγει ένα μοντέλο K Means, το οποίο είναι ένας μετασχηματιστής.

Παράμετρος (Parameter) Τα στοιχεία μηχανικής μάθησης χρησιμοποιούν ένα κοινό API για τον καθορισμό παραμέτρων. Ένα παράδειγμα παραμέτρου είναι ο μέγιστος αριθμός επαναλήψεων που θα πρέπει να χρησιμοποιεί το μοντέλο

Τα στοιχεία του pipeline δεδομένων για την περίπτωση χρήσης ταξινόμησης κειμένου φαίνονται στο ακόλουθο διάγραμμα.

Το πακέτο Spark ML περιέχει πολλούς αλγορίθμους και βοηθητικά προγράμματα. Οι κύριες ενότητες είναι οι εξής:

- pyspark.ml.feature για εξαγωγή χαρακτηριστικών και σχηματισμό χαρακτηριστικών
- pyspark.ml.classification module για εργασία ταξινόμησης

- pyspark.ml.clustering module για εργασία ομαδοποίησης
- pyspark.ml.linalg module για λειτουργία άλγεβρας
- pyspark.ml.regression module για εργασία παλινδρόμησης
- pyspark.ml.stat module για στατιστική ανάλυση
- pyspark.ml.evaluation module για την αξιολόγηση της απόδοσης

Στις διαφάνειες αυτού του μαθήματος, παρουσιάζουμε τρία παραδείγματα εφαρμογής του Spark ML. Χρησιμοποιούμε το σύνολο δεδομένων ίριδας (<https://gist.github.com/netj/8836201>) που περιέχει τέσσερα χαρακτηριστικά (μήκος και πλάτος των σέπαλων και των πετάλων) από 50 δείγματα τριών ειδών Iris (Iris setosa, Iris virginica και Iris versicolor). Αντιμετωπίζουμε τρεις διαφορετικές εργασίες:

- Εργασία παλινδρόμησης : ο στόχος είναι να εκτιμηθεί το πλάτος των πετάλων χρησιμοποιώντας το μήκος και το πλάτος των σέπαλων και το μήκος των πετάλων ως χαρακτηριστικό γνώρισμα. Για να αντιμετωπίσουμε αυτό το έργο, εφαρμόζουμε μια γραμμική παλινδρόμηση.
- Εργασία ταξινόμησης: στόχος είναι η ταξινόμηση των δειγμάτων ίριδας στη σωστή κατηγορία (είδος). Χρησιμοποιούμε και τα τέσσερα χαρακτηριστικά ως είσοδο και τα είδη ως κλάσεις για να προβλέψουμε. Για να αντιμετωπίσουμε αυτό το έργο, χρησιμοποιούμε ένα δέντρο απόφασης.
- Εργασία ομαδοποίησης: στόχος είναι η ομαδοποίηση των δειγμάτων σε κλάσεις παρόμοιων στοιχείων. Χρησιμοποιούμε και τα τέσσερα χαρακτηριστικά ως είσοδο. Για να αντιμετωπίσουμε αυτό το έργο, χρησιμοποιούμε έναν αλγόριθμο kmeans

Σε όλα τα παραδείγματα που χρησιμοποιούν το PySpark πρέπει να θυμόμαστε ότι θα υπάρχουν μόνο δύο στήλες δεδομένων: η μεταβλητή που πρέπει να προβλεφθεί και η όλες οι άλλες μεταβλητές (χαρακτηριστικά) συγκεντρωμένες και μετασχηματισμένες σε μία μόνο στήλη. Για το λόγο αυτό, στη συνέχεια, εισάγουμε τις ενότητες για τον μετασχηματισμό των δεδομένων. Για παράδειγμα, στην εργασία μας της παλινδρόμησης γράφουμε:

```
# to import modules
from pyspark.ml.linalg import Vectors
from pyspark.ml.feature import VectorAssembler
# to aggregate the feature in one column
assembler = VectorAssembler(inputCols=["Sepal_Length", "Sepal_Width", "Petal_Length"],
                             outputCol="features")
# to insert the new column to dataframe
transform = assembler.transform(dataframe)
```

Το Databricks παρέχει ένα περιβάλλον που διευκολύνει τη δημιουργία, την εκπαίδευση και την ανάπτυξη μοντέλων βαθιάς μάθησης σε κλίμακα. Πολλά deep βιβλιοθήκες μάθησης είναι διαθέσιμες στο Databricks Runtime ML, ένα περιβάλλον εκτέλεσης μηχανικής μάθησης που παρέχει ένα έτοιμο περιβάλλον για τη μηχανική μάθηση και την επιστήμη των δεδομένων.

Το Databricks για εκπαίδευση σε έναν κόμβο υποστηρίζει τις τρεις δημοφιλείς βιβλιοθήκες βαθιάς μάθησης, TensorFlow και PyTorch και Deep Learning Pipelines, η τελευταία είναι μια βιβλιοθήκη

ανοικτού κώδικα που δημιουργήθηκε από την Databricks και παρέχει υψηλού επιπέδου APIs για κλιμακούμενες βαθιές μάθηση σε Python με το Apache Spark.

Επιπλέον, το Databricks υποστηρίζει κατανεμημένη εκπαίδευση χρησιμοποιώντας:

HorovodRunner, ένα εργαλείο που απλοποιεί τη διαδικασία μετάβασης από TensorFlow, Keras και PyTorch για μία μηχανή σε μηχανήματα πολλαπλών GPU και συστάδες πολλαπλών κόμβων.

ή

Το Spark-tensorflow-distributor είναι ένα εγγενές πακέτο ανοικτού κώδικα στο TensorFlow που βοηθά τους χρήστες να κάνουν κατανεμημένη εκπαίδευση με το TensorFlow στις συστάδες Spark τους. Είναι χτισμένο πάνω στο distribute.Strategy, ένα από τα σημαντικότερα χαρακτηριστικά του TensorFlow 2.

Περιπτώσεις χρήσης

TENSORFLOW

Αυτή η περίπτωση χρήσης δείχνει πώς να χρησιμοποιήσετε το TensorFlow στον κόμβο οδήγησης Spark για να προσαρμόσετε ένα νευρωνικό δίκτυο σε δεδομένα αναγνώρισης χειρόγραφων ψηφίων MNIST.

MNIST για αρχάριους ML

https://docs.databricks.com/_static/notebooks/deep-learning/tensorflow-single-node.html

Προαπαιτούμενα:

Συστάδα με δυνατότητα GPU.

Το TensorFlow 1.15 ή 2.x με υποστήριξη GPU εγκατεστημένο χειροκίνητα.

[ΚΑΤΕΒΑΣΤΕ ΤΟΝ ΚΩΔΙΚΑ ΕΛΩ](#)

Βήμα 1: Εισαγωγή TensorFlow

```
import tensorflow as tf
from tensorflow.keras import models, layers, datasets
```

Βήμα 2: Φορτώστε τα δεδομένα (αυτό το βήμα μπορεί να πάρει λίγο χρόνο)

```
(x_train, y_train), (x_test, y_test) = datasets.mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0
```

Βήμα 3: Ορισμός του μοντέλου

```
model = models.Sequential([
    layers.Flatten(input_shape=(28, 28)),
```

```
layers.Dense(10, activation='softmax'))
```

Βήμα 4: Ορισμός απώλειας και βελτιστοποιητή

```
model.compile(optimizer=tf.keras.optimizers.SGD(learning_rate=0.5),  
              loss=tf.keras.losses.sparse_categorical_crossentropy,  
              metrics=['accuracy'])
```

Βήμα 5: Ξεκινήστε το TensorBoard για να μπορείτε να παρακολουθείτε την πρόοδο της εκπαίδευσης

```
log_dir = "/dbfs/ml/tmp/tensorflow_log_dir"  
dbutils.tensorboard.start(log_dir)
```

Βήμα 6: Εκπαίδευση του μοντέλου σε παρτίδες

```
tensorboard_callback = tf.keras.callbacks.TensorBoard(log_dir=log_dir, histogram_freq=1)  
model.fit(x_train, y_train, epochs=5, batch_size=64, callbacks=[tensorboard_callback])
```

Βήμα 7: Δοκιμή του εκπαιδευμένου μοντέλου

```
model.evaluate(x_test, y_test, verbose=2)
```

Αναφορές

- Marin, I., Shukla, A., & Sarang, V. K. (2019). Ανάλυση μεγάλων δεδομένων με Python : Συνδυάστε το Spark και την Python για να ξεκλειδώσετε τις δυνάμεις του παράλληλου υπολογισμού και της μηχανικής μάθησης. Packt Publishing Ltd.
- Chambers, B., & Zaharia, M.(2018). Spark : Ο οριστικός οδηγός : Επεξεργασία μεγάλων δεδομένων με απλό τρόπο " O'Reilly Media,Inc."
- <http://spark.apache.org/docs/latest/cluster-overview.html>
- https://www.tutorialspoint.com/apache_spark/apache_spark_rdd.htm
- <http://spark.apache.org/docs/2.2.0/sql-programming-guide.html>