



ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

**Μελέτη του ρόλου της Συνάρτησης Ενεργοποίησης ενός
Νευρομορφικού Συστήματος με σκοπό την υλοποίηση του με
χρήση Φωτονικών Τεχνολογιών**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

ΒΑΣΙΛΙΚΗ ΠΑΠΑΔΗΜΟΥ

Επιβλέπων :

ΧΑΡΗΣ ΜΕΣΑΡΙΤΑΚΗΣ

Σαμος, Σεπτέμβριος 2022

© [2022]

Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων
ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

Υπεύθυνη Δήλωση Πρωτοτυπίας Διπλωματικής Εργασίας

Δηλώνω ότι, κάθε βοήθεια που είχα για την συγγραφή της παρούσας διπλωματικής εργασίας είναι αναγνωρισμένη και αναφέρεται στην εργασία. Τμήματα που έχουν ληφθεί παραφρασμένα από άλλες εργασίες ή αυτολεξεί, έχουν αναφερθεί με αντίστοιχη σημειολογία στο κείμενο και η βιβλιογραφία αναφέρεται λεπτομερώς στο τελευταίο μέρος της διπλωματικής, ενώ Το παρών κείμενο είναι πρωτότυπο και δεν έχει υποβληθεί όλο ή εν μέρει για εξέταση για οποιοδήποτε άλλο πτυχίο ή προσόντα σε αυτό ή οποιοδήποτε άλλο πανεπιστήμιο. Η συγγραφή της εργασίας είναι αποτέλεσμα ατομικής δουλειάς και σε συνεργασία με τον υπεύθυνο καθηγητή. Αυτό το έργο περιλαμβάνει λιγότερες από 65.000 λέξεις συμπεριλαμβανομένων παραρτημάτων, βιβλιογραφίας, υποσημειώσεων, πινάκων και εξισώσεων και είναι λιγότερες από 250.

ΒΑΣΙΛΙΚΗ ΠΑΠΑΔΗΜΟΥ

Σάμος, Σεπτέμβριος 2022

Ευχαριστίες

Η παρούσα διπλωματική εργασία πραγματοποιήθηκε στο Πανεπιστήμιο Αιγαίου στην Πολυτεχνική Σχολή του Τμήματος Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων κατά το έτος 2022.

Η υλοποίηση της διπλωματικής εργασίας είναι μία πολύ ψυχολογικά έντονη περίοδος που μου έμαθε πολλά πράγματα όχι μόνο πάνω στο αντικείμενο μελέτης αλλά και για τον εαυτό μου. Η τριβή μου με το συγκεκριμένο θέμα ήταν ελάχιστη όταν πρωτοξεκίνησα και υπήρξαν κάποια άτομα που έπαιξαν σημαντικό ρόλο και με βοήθησαν να φέρω εις πέρας αυτό το έργο. Χάρη στην βοήθεια αυτών των ατόμων κατέληξα να αγαπήσω και να σκέφτομαι πως θα μπορούσε να εξελιχθεί.

Αρχικά, θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου, Αν. Καθηγητή του Πανεπιστημίου Αιγαίου κ. Χάρη Μεσαριτάκη, για την εμπιστοσύνη να μου αναθέσει το συγκεκριμένο θέμα. Υπήρχε από την αρχή συνεχής υποστήριξη κατανόηση και επικοινωνία κατά την διάρκεια της εκπόνησης.

Θα ήθελα να ευχαριστήσω όλη την επιστημονική κοινότητα του πανεπιστημίου αιγαίου που σε όλη την διάρκεια στων σπουδών μου βοήθησαν να ανακαλύψω έναν κόσμο (αυτόν της πληροφορικής) που όμως περιλαμβάνει γνώσεις από διάφορα πεδία. Αυτό με βοήθησε να βρω τον τομέα που πραγματικά με παθιάζει και θα προσπαθώ συνέχεια να εξελίσσομαι.

Για την συγκεκριμένη διπλωματική θα ήθελα να ευχαριστήσω και όλη την επιστημονική κοινότητα που ασχολείται με την μελέτη πάνω στον συγκεκριμένο τομέα καθώς στην αρχή δεν πίστευα ότι υπάρχει τόσο μεγάλος όγκος εστιασμένης πληροφορίας στο διαδίκτυο.

Επιπλέον, ευχαριστώ τους φίλους και συνεργάτες μου σε όλο το σπουδαστικό μου ταξίδι για τις στιγμές εργασίας εν ώρα πανεπιστημίου αλλά και τις υπέροχες αναμνήσεις που αποκτήσαμε.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου που μου συμπαραστάθηκε όταν χρειάστηκα την στήριξή τους, καθ' όλη την διάρκεια των σπουδών μου, αλλά ιδιαίτερα στην συγκεκριμένη φάση της διπλωματικής εργασίας.

Αυτή η διπλωματική εργασία δεν θα μπορούσε να έχει υλοποιηθεί αν δεν υπήρχε η πρακτική, και ψυχολογική υποστήριξη αυτών των ατόμων.

Ευχαριστώ θερμά.

Περιεχόμενα

ΠΕΡΙΛΗΨΗ.....	9
ΕΙΣΑΓΩΓΗ.....	12
1 ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ (ML) – ΑΛΓΟΡΙΘΜΟΙ.....	14
1.1 Supervised ML και Non-Supervised αλγόριθμοι	14
1.2 Νευρωνικά Δίκτυα (NN)	17
1.2.1 Συναρτήσεις Ενεργοποίησης	18
1.2.2 Single Layer Perceptron (SLP)	20
1.2.3 Multi Layer Perceptron (MLP).....	20
1.2.4 Recurrent Neural Networks (RNN).....	22
1.2.5 Convolutional Neural Networks (CNN)	24
1.2.6 Spiking Neural Networks (SNN)	25
2 RESERVOIR COMPUTING NETWORKS (RCN) – ECHO STATE NETWORKS (ESN)	26
2.1 Ιστορική Αναδρομή	26
2.2 Reservoir Computing	27
2.3 Βιβλιογραφική Αναφορά των RC Εφαρμογών	28
2.4 Echo State Networks (ESN)	30
2.4.1 Μαθηματοποίηση του ESN.....	31
2.4.2 Βασικές παράμετροι ESN	34
2.4.3 Εκπαίδευση και Σφάλμα Εκπαίδευσης.....	35
2.4.4 Τοπολογίες των RC στα ESN.....	35
3 ΠΕΙΡΑΜΑΤΙΚΟ ΜΕΡΟΣ 1 : ΥΛΟΠΟΙΗΣΗ TRAINING ESN ΓΙΑ NARMA10, NARMA5 ΜΕ Tanh....	38
3.1 NARMA10 για Activation Function Tanh	40
3.1.1 Διάγραμμα NRMSE με Φασματική Ακτίνα	40
3.2 NARMA5 για Activation Function Tanh	43
3.2.1 Διάγραμμα NRMSE με Φασματική Ακτίνα	43
3.3 Σύγκριση NARMA10 με NARMA5 και activation function Tanh	45
4 ΠΕΙΡΑΜΑΤΙΚΟ ΜΕΡΟΣ 2: ΥΛΟΠΟΙΗΣΗ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ ΓΙΑ ΔΙΑΦΟΡΕΣ ACTIVATION FUNCTION	46
4.1 Διερεύνηση Προβλήματος NARMA10 για Activation Function Sigmoid	46
4.2 Διερεύνηση προβλήματος NARMA10 για Activation Function Identity	48
4.3 Διερεύνηση Προβλήματος NARMA10 για Activation Function RELU	50
4.4 Διερεύνηση Προβλήματος NARMA10 για Πολυωνυμική Activation Function ax^2.	54

4.5	Γενικά Συμπεράσματα πειραματικής υλοποίησης	60
5	ΦΩΤΟΝΙΚΗ ΤΕΧΝΟΛΟΓΙΑ ΚΑΙ ΦΩΤΟΝΙΚΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ.....	63
5.1	Προβλήματα αρχιτεκτονικής Von Neuman	63
5.2	Φωτονικά εξαρτήματα και εφαρμογές	64
5.3	Νευρομορφική Τεχνολογία	70
5.4	Εξέλιξη των φωτονικών NN	72
5.4.1	Φωτονικός Νευρώνας.....	72
5.4.2	Οπτοηλεκτρονικός Νευρώνας	73
5.4.3	Εξολοκλήρου Οπτικός Νευρώνας	75
5.5	Φωτονικά Νευρωνικά Δίκτυα	76
5.5.1	Συνδέσεις σε ένα NN.....	76
5.5.2	Αρχιτεκτονικές PNN και τρόποι στάθμισης	77
5.6	Φωτονικό Reservoir Computing	79
5.6.1	Διαφορές RC φωτονικού με ηλεκτρονικό.....	79
5.6.2	Τοπολογίες και Υλοποιήσεις.....	79
6	ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΔΡΑΣΕΙΣ	82
7	Παράρτημα.....	84
	Πίνακας Εικόνων	86
	Πίνακας Πινάκων	88
	Βιβλιογραφία	89

ΠΕΡΙΛΗΨΗ

Στόχος της παρούσας διπλωματικής εργασίας είναι να γίνει μια ανασκόπηση των αλγορίθμων νευρωνικών δικτύων και να γίνουν κατανοητές οι κύριες λειτουργίες του Reservoir Computing, πιο συγκεκριμένα η τοπολογία του Echo State Network, στη θεωρία και στην πράξη. Το λογισμικό esnToolbox (MATLAB) χρησιμοποιήθηκε για την υλοποίηση ενός συμβατικού Echo State Network. Το training και το testing του ESN έγινε με NARMA10 και NARMA5. Μερικές παράμετροι που τροποποιήθηκαν για την επίτευξη του στόχου είναι ο αριθμός των κόμβων της δεξαμενής και η φασματική ακτίνα. Χρησιμοποιήθηκαν κοινές activation function, καθώς και μια πολυωνυμική συνάρτηση ax^2 , δοκιμάστηκαν και τελικά συγκρίθηκαν μεταξύ τους ως προς τα αποτελέσματα των γραφημάτων NRMSE. Έγινε αναφορά στις αρχιτεκτονικές Neuromorphic Computing (NC) και των Φωτονικών Νευρωνικών Δικτύων. Τέλος, συζητήθηκαν τα αποτελέσματα με διαφορετικές συναρτήσεις ενεργοποίησης, με βάση το συμβατικό ESN και συγκρίθηκαν με μη γραμμικότητες που χρησιμοποιούνται σε φωτονικές εφαρμογές RC.

Abstract

In this thesis the goal is to give an overview of Neural Network algorithms and understand the main functionalities of Reservoir Computing, more specifically the Echo State Network topology in theory and in practice. The software esnToolbox (MATLAB code) was used to implement a conventional Echo State Network. The training and testing of the ESN was done with NARMA10 and NARMA5. Some parameters were modified, such as the number of reservoir nodes and the spectral radius. Common activation functions, as well as a polynomial ax^2 function were tried and finally compared against each other as of the results of NRMSE graphs. An introduction of Neuromorphic Computing architectures and Photonic Neural Networks was given. Finally, the results with different activation functions, based on the conventional ESN were discussed and compared against non-linearities used in photonic RC implementations.

ΕΙΣΑΓΩΓΗ

Από τους πιο συχνά εμφανιζόμενους αλγορίθμους Μηχανικής Μάθησης είναι οι αλγόριθμοι επιβλεπόμενης εκπαίδευσης (Supervised Learning), οι μη-επιβλεπόμενης εκπαίδευσης (Unsupervised Learning) και τα Νευρωνικά Δίκτυα (NN) τα οποία αναλύονται εκτενώς στην παρούσα διπλωματική εργασία. Πιο συγκεκριμένα, γίνεται αναφορά σε διάφορα είδη NN, όπως είναι τα perceptron (single και multilayer perceptron), τα αναδρομικά NN (RNN), τα Convolutional NN (CNN) και τα Spike NN (SNN). Η διπλωματική εστιάζει σε μια παραλλαγή των RNN, που αντί για πολλά hidden layers στο εσωτερικό τους, δομούνται με ένα reservoir και η τεχνική λέγεται Reservoir Computing (RC). Λόγω των πολλών πλεονεκτημάτων τους, όπως αναφέρεται και στην βιβλιογραφία, τα RC βρίσκουν εφαρμογή σε πολλούς τομείς, όπως υγεία, βιομηχανία κ.α. Οι δύο βασικοί τύποι RC είναι τα Echo State Network (ESN) και τα Liquid State Machine (LSM), και στην διπλωματική γίνεται εμβάθυνση στην πρώτη κατηγορία, η οποία λύνει προβλήματα χρονοσειρών. Ο δημιουργός των ESN ήταν ο Jaeger, ο οποίος έθεσε τις παραμέτρους και τις προϋποθέσεις σύμφωνα με τις οποίες δομείται ένα ESN καθώς και τον τρόπο με τον οποίο εκπαιδεύονται. Μια βασική ιδιότητα που έχουν τα ESN και τα καθιστά εξαιρετικό εργαλείο επίλυσης σύνθετων προβλημάτων είναι η ύπαρξη ενός είδους μνήμης, που όμως για να εξασφαλιστεί πρέπει να ισχύει η ιδιότητα Echo State Property (ESP). Επιπλέον, προϋπόθεση αποτελεί η χρήση κατάλληλης μη-γραμμικής συνάρτησης ενεργοποίησης σε κάθε νευρώνα, προκειμένου η αναμενόμενη έξοδος (predicted) να είναι παρόμοια με την επιθυμητή έξοδο του συστήματος (teacher/target).

Όλες οι λειτουργίες ενός ESN, όπως παρουσιάστηκαν από τον Jaeger, προσομοιώνονται στην παρούσα διπλωματική εργασία με την βοήθεια κώδικα MATLAB και το εργαλείο esnToolbox, για ένα συμβατικό ESN. Προκειμένου να γίνει η μελέτη της συμπεριφοράς του ESN, τροποποιούνται παράμετροι όπως οι κόμβοι του reservoir και η φασματική ακτίνα. Δοκιμάζονται διαφορετικές activation functions (tanh, sigmoid, identity, ReLu, Shift ReLu και πολυωνυμική x^2 με $\alpha=0.1, 0.5, 1, 4, 6$) και εκπαιδεύεται το reservoir με χρήση NARMA10 προκειμένου να παραχθούν κατάλληλα διαγράμματα και να γίνει σύγκρισή τους ως προς την αποδοτικότητα της κάθε μίας. Τα παραχθέντα διαγράμματα αποτυπώνουν την τιμή του σφάλματος NRMSE, ως προς την μεταβολή των κόμβων ή της φασματικής ακτίνας. Αναλόγως αν η καμπύλες για την περίπτωση του train και του test συμπίπτουν, προκύπτει ο βαθμός επιτυχίας της εκπαίδευσης στο σύστημα. Επιπλέον, για την tanh γίνεται σύγκριση των αποτελεσμάτων για NARMA 10 και NARMA5.

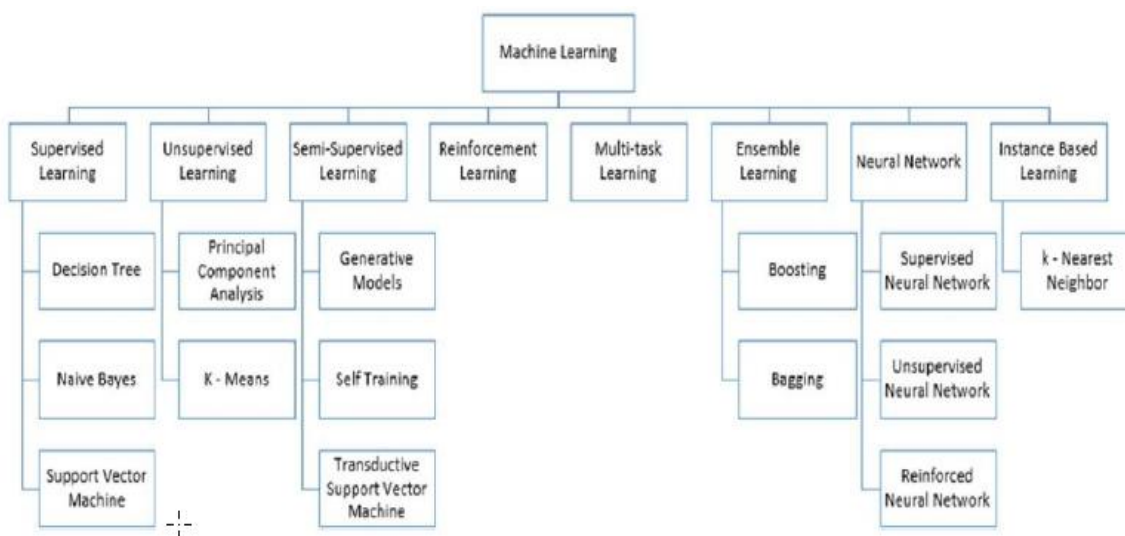
Στην συνέχεια παρουσιάζεται μια σύντομη ιστορική αναδρομή για τις διάφορες αρχιτεκτονικές υπολογιστών (von Neumann και non von Neumann) και σύγκρισή τους. Επίσης έγινε ανάλυση της Νευρομορφικής Τεχνολογίας (NC) και της ανάγκης για δημιουργία hardware δομών που να μπορούν να επεξεργάζονται ταχύτατα δεδομένα. Λύση παρέχεται με χρήση οπτικών εξαρτημάτων για την δημιουργία φωτονικών δικτύων είτε αμιγώς οπτικών, είτε οπτοηλεκτρονικών. Απώτερος στόχος είναι η δημιουργία φωτονικών, εξολοκλήρου οπτικών NN, καθώς δεν απαιτείται κάποια ενδιάμεση μετατροπή της πληροφορίας και συνεπώς παρέχουν ταχύτερη και πιο ενεργοαποδοτική επεξεργασία δεδομένων σε σχέση με τα οπτοηλεκτρονικά. Αντίστοιχα με την ανάλυση των συμβατικών NN, εξετάζεται η δομή ενός οπτικού νευρώνα (photonic perceptron) και η δομή ενός οπτικού και οπτοηλεκτρονικού NN με χρήση κατάλληλων οπτικών hardware. Παρουσιάζονται μερικές αρχιτεκτονικές και υλοποιήσεις για φωτονικό RC που υπάρχουν ως σήμερα. Τέλος, γίνεται αναφορά σε διάφορες συναρτήσεις ενεργοποίησης που

χρησιμοποιούνται κατά την δημιουργία ενός φωτονικού NN με δομή RC και η ανάγκη για χρήση μίας ταχύτερης μη-γραμμικής συνάρτησης εκπαίδευσης όπως η πολυωνυμική ax^2 .

Στην **ενότητα 1**, γίνεται μια ανάλυση των υπαρχόντων αλγορίθμων μηχανικής μάθησης με εκτενέστερη αναφορά στους NN και τις τεχνικές που χρησιμοποιούνται για την εκπαίδευσή τους. Επίσης, αναλύονται τα Echo State Networks (ESN) NN που είναι κατηγορία Reservoir Computing (RC) NN, τίθεται το βασικό υπόβαθρο για την κατανόηση (ενότητα 2) και γίνεται software υλοποίηση ενός ESN (**ενότητα 3 και 4**) με χρήση του εργαλείου esnToolbox. Πιο συγκεκριμένα στην **ενότητα 3**, γίνεται υλοποίηση ESN με NARMA 5 και NARMA 10, προκειμένου να συγκριθούν τα παραγόμενα αποτελέσματα χρονοσειρών. Τα αποτελέσματα της **ενότητας 4**, προκύπτουν τροποποιώντας τον αριθμό των νευρώνων του reservoir, την φασματική ακτίνα του και την συνάρτηση ενεργοποίησης. Συγκρίνονται τα αποτελέσματα που προκύπτουν μετά την εκπαίδευση με κάθε συνάρτηση ενεργοποίησης. Στην **ενότητα 5**, γίνεται βιβλιογραφική αναφορά στη Νευρομορφική Τεχνολογία και τη χρήση φωτονικών NN. Παρουσιάζονται, διάφορες αρχιτεκτονικές και υλοποιήσεις φωτονικού νευρομορφικού RC που υλοποιούνται με οπτικό hardware για να επιλύσουν με αποτελεσματικότητα πολύπλοκα προβλήματα.

1 ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ (ML) – ΑΛΓΟΡΙΘΜΟΙ

Η μηχανική μάθηση (MM) είναι ένας εξελισσόμενος τεχνολογικός τομέας. Έχει διάφορες εφαρμογές σε πολλά επιστημονικά πεδία όπως: την επεξεργασία γλώσσας, υπολογιστική όραση, αναγνώριση φωνής, όπως επίσης και στον τομέα της ρομποτικής για τον χειρισμό των κινήσεων (Basha, 2019). Σύμφωνα με τον Arthur Samuel, η MM δίνει την δυνατότητα σε ένα σύστημα υπολογιστών να «μαθαίνει» να παράγει δεδομένα με αποδοτικό τρόπο χωρίς να προαπαιτεί κάποιο ειδικό πρόγραμμα, (Mahesh, 2020). Ο όγκος δεδομένων που είναι διαθέσιμος σήμερα είναι τεράστιος. Η λύση ενός προβλήματος γίνεται πολύπλοκη και καθίσταται δύσκολη η διαδικασία άντλησης πληροφορίας με μαθηματικά μοντέλα. Έτσι, η MM καθίσταται αναγκαία και χρησιμοποιούνται κατάλληλοι αλγόριθμοι ανάλογα με τον τύπο του προβλήματος προς επίλυση (Osisanwo, 2017), (Mahesh, 2020). Στην *εικόνα 1*, εμφανίζονται οι πιο διαδεδομένοι αλγόριθμοι MM, όπως: Supervised (SML), Unsupervised (Non-SML), Decision Tree, K-Means, Naïve Bayes, Neural Networks (NN) κλπ. Εκτενής ανάλυση των παραπάνω αλγορίθμων γίνεται στα άρθρα (Ayodele, 2010), (Mahesh, 2020), (Basha, 2019), (Singh, 2016).



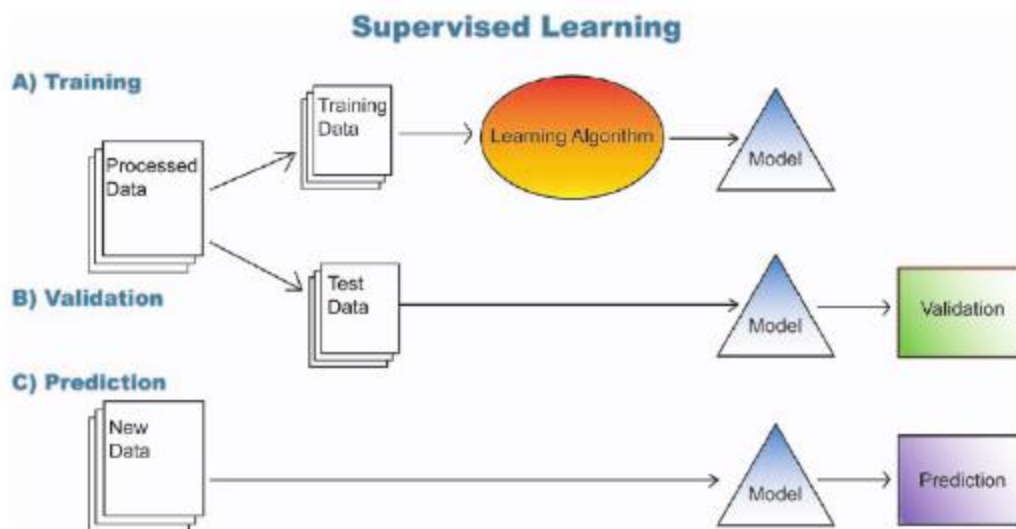
Εικόνα 1. Είδη Αλγορίθμων MM. (Mahesh, 2020)

Για τους σκοπούς της παρούσας διπλωματικής εργασίας θα γίνει μια σύντομη παρουσίαση αλγορίθμων SML και Non-SML.

1.1 Supervised ML και Non-Supervised αλγόριθμοι

Στο SML μοντέλο υπάρχει μία πηγή με λίγα προεπεξεργασμένα δεδομένα τα οποία χωρίζονται σε training data (σε μέγιστο ποσοστό) και test data (σε μικρότερο ποσοστό). Ο αλγόριθμος παράγει μία mapping function η οποία αντιστοιχίζει input σε output δεδομένα (Ayodele, 2010).

Στην *εικόνα 2* παρουσιάζεται η γραφική αναπαράσταση ενός SML. Κατά τη φάση της εκπαίδευσης (*εικόνα 2Α- Training*), τα training data χρησιμοποιούνται για την εκπαίδευση ενός αλγόριθμου μάθησης σε μια προσπάθεια να αναπτυχθεί ένα ακριβές μοντέλο πρόβλεψης. Για την επικύρωση του μοντέλου (*εικόνα 2Β- Validation*), τα δεδομένα δοκιμής (test data) εφαρμόζονται στη συνέχεια στο μοντέλο και αξιολογείται η προγνωστική ακρίβεια. Μόλις επικυρωθούν (*εικόνα 2C - Prediction*), εισάγονται νέα δεδομένα στο μοντέλο σε μια προσπάθεια να γίνουν νέες προβλέψεις (Kanevsky, 2016) .



Εικόνα 2 . Γραφική αναπαράσταση SML. (Kanevsky, 2016)

Παράδειγμα εφαρμογής SML, ο έλεγχος σε συστήματα τραπεζών για απάτες λειτουργεί με SML. Έτσι όταν γίνεται μία αγορά με κάρτα, μειώνεται το υπόλοιπο και δίνεται η δυνατότητα στον χρήστη να ελέγξει αν η κίνησή του ήταν αληθής ή ψευδής (Basha, 2019). Ένα άλλο παράδειγμα είναι οι κριτικές ταινιών σε site όπως IMDB, Amazon, Cnet κλπ προκειμένου ο καταναλωτής να λάβει θετική ή αρνητική απόφαση αγοράς (Zhou, 2005).

Κατά την παραγωγή εξόδων (output) με την τεχνική SML, δύο είναι τα πιο δημοφιλή προβλήματα που μπορούν να προκύψουν (Mahesh, 2020):

- το πρόβλημα του **classification** των εξόδων όταν υπάρχουν διακριτά δεδομένα input
- και αυτό του **regression** όταν υπάρχουν συνεχή δεδομένα input.

Στην πρώτη περίπτωση του classification (εικόνα 4β), αν τα διακριτά δεδομένα που εισάγονται στο σύστημα δεν αντιστοιχούν σε κάποιο προϋπάρχον label, η συνάρτηση ταξινόμησης δεν μπορεί να αντιστοιχίσει την παραγόμενη διακριτή έξοδο με κάποια κατηγορία εισόδου και έτσι προκύπτει το πρόβλημα του classification (Basha, 2019). Ένα παράδειγμα όπου είναι χρήσιμη η κατηγοριοποίηση δεδομένων, είναι η αναγνώριση ψηφίων (Agodele, 2010). Έστω ότι υπάρχει ένα μεγάλο dataset που έχει όλους τους διακριτούς τύπους ψηφίων (0-9) με πολλές διαφορετικές γραφές. Αν εισάγουμε το ψηφίο 8, τότε θα ελεγχθούν όλοι οι τύποι γραφής για το label του ψηφίου 8 και θα προκύψει η αντίστοιχη έξοδος.

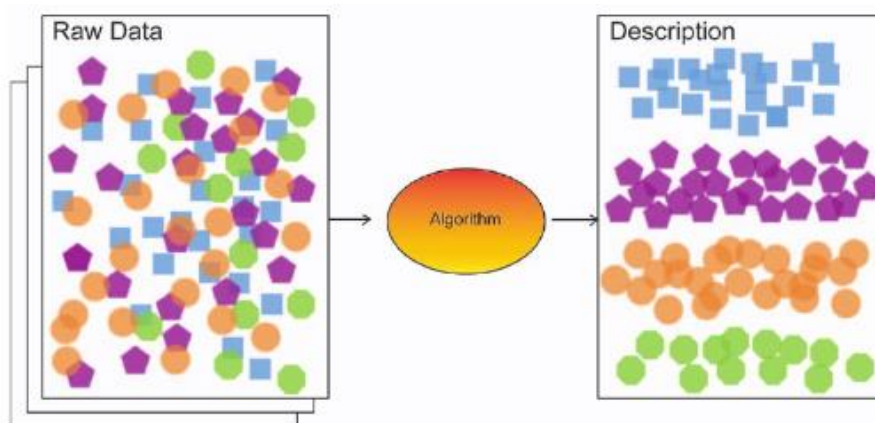
Στην δεύτερη περίπτωση του regression (εικόνα 4α), δεν μπορεί να γίνει η κατηγοριοποίηση, επειδή υπάρχουν συνεχείς τιμές. Έτσι, για κάθε συνεχή τιμή στην είσοδο του train dataset ο αλγόριθμος προσπαθεί να αντιστοιχίσει μέσω κατάλληλων συναρτήσεων, συνεχείς εξόδους χρησιμοποιώντας μεθόδους πρόβλεψης (Basha, 2019).

Η άλλη μεγάλη κατηγορία εκπαίδευσης των αλγορίθμων μηχανικής μάθησης είναι η Non-SML (εικόνα 3). Αυτή η τεχνική αποδίδει καλύτερα όταν υπάρχουν πολλά δεδομένα (Basha, 2019). Ο στόχος είναι να εκπαιδευτεί εξ' αρχής ο υπολογιστής για το πως να εκτελέσει μια διεργασία χωρίς προηγουμένως να έχει λάβει οδηγίες από κάποιον "εκπαιδευτή" (Agodele, 2010) . Με αυτή την μέθοδο τα δεδομένα προς

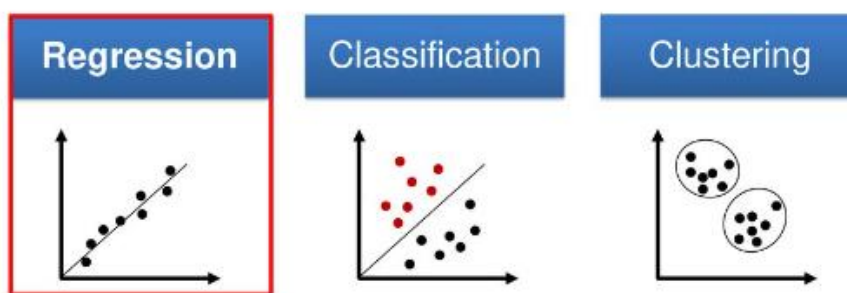
εκπαίδευση δεν μπορούν να ομαδοποιηθούν από μόνα τους απευθείας (Basha, 2019), (Osisanwo, 2017). Σύμφωνα με τον Ghahramani (Ghahramani, 2008), (Ayodele, 2010) οι αλγόριθμοι Non-SML σχεδιάζονται έτσι ώστε να μπορούν μέσα από δείγματα δεδομένων να εξάγουν την δομή τους. Στην Non-SML το σύστημα πρέπει να μάθει να τοποθετεί τα δεδομένα σε κατηγορίες και αυτό μπορεί να επιτευχθεί:

- είτε με την μέθοδο **reward-success**,
- είτε με την δημιουργία **clusters** (εικόνα 4γ).

Η reward success τεχνική μπορεί να χρησιμοποιηθεί όταν έχουμε πολύπλοκα προβλήματα που απαιτούν πολύ χρόνο για να λυθούν. Στην περίπτωση αυτή το σύστημα προσπαθεί να μάθει χρησιμοποιώντας έναν «δείκτη» για τις επιτυχίες και όχι με βάση την ομαδοποίηση. Η περίπτωση του clustering χρησιμοποιείται για να αναγνωριστούν οι ομοιότητες μεταξύ των δεδομένων και να κατηγοριοποιηθούν σε ομάδες (clusters) (Ayodele, 2010).



Εικόνα 3. Unsupervised Machine Learning (Kanevsky, 2016)



Εικόνα 4. Γραφική αναπαράσταση τεχνικών εκμάθησης SML και NSML αλγορίθμων

1.2 Νευρωνικά Δίκτυα (NN)

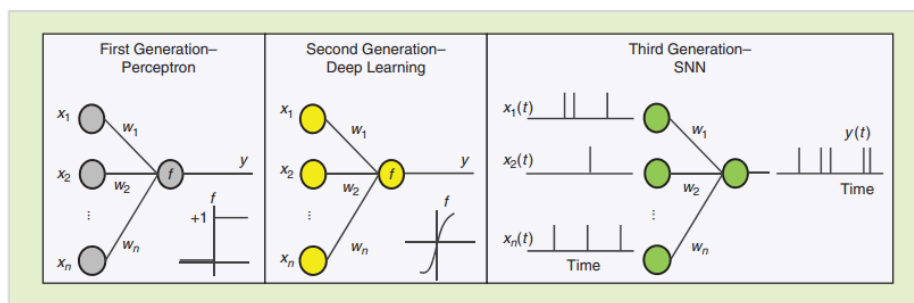
Τα νευρωνικά δίκτυα συναντώνται στον τομέα της επιστήμης της βιολογίας-ιατρικής αλλά και στον τομέα της πληροφορικής σαν αλγόριθμοι Machine Learning. Στον ανθρώπινο εγκέφαλο τα μηνύματα για λήψη αποφάσεων, για απόκτηση γνώσης καθώς και για επίλυση σύνθετων προβλημάτων, μεταφέρονται με την βοήθεια των νευρώνων. Ο εγκέφαλος δέχεται «ηλεκτρικά σήματα» και με την βοήθεια χημικών διεργασιών τα επεξεργάζεται και παράγει μια αντίδραση σαν έξοδο (Brunner D. P., 2018), (McCulloch, 1943), (Ding, 2018, June). Λόγω του ότι στον ανθρώπινο εγκέφαλο υπάρχουν περίπου 86 δισεκατομμύρια νευρώνες, με τον καθένα να πραγματοποιεί 10^4 συνάψεις, τα ερεθίσματα που δέχεται υφίστανται ταχύτατη επεξεργασία (Noriega, 2005). Για παράδειγμα, ο εγκέφαλος αν δεχθεί σαν ερέθισμα μια φωτογραφία ενός λουλουδιού, μπορεί να αναγνωρίσει αυτό το αντικείμενο και όταν το δει στην πραγματικότητα. Αυτό επιτυγχάνεται, επειδή οι νευρώνες έχουν καταφέρει να εκπαιδευτούν κατάλληλα, ώστε ο εγκέφαλος να μπορεί να συμπεράνει ότι και στις δύο περιπτώσεις η εικόνα που δέχτηκε, ήταν ένα λουλούδι. Μια καθυστερημένη αντίδραση σε κάποιο ερέθισμα, θα ήταν καταστροφική για τον άνθρωπο, καθώς θα μπορούσε να του προκαλέσει κάποια κρίση (Barrett, 2019). Οι McCulloch και Pitts (το 1943) ήταν οι πρώτοι που ξεκίνησαν να μελετούν με μαθηματικό τρόπο τα φυσικά NN του εγκεφάλου, χρησιμοποιώντας τεχνητά αλγοριθμικά NN που υφίστανται στον τομέα της πληροφορικής (McCulloch, 1943).

Ο στόχος των NN και στον τομέα της πληροφορικής, και στον εγκέφαλο είναι ίδιος. Ένας (τεχνολογικός) NN αλγόριθμος πρέπει να είναι σε θέση να λαμβάνει έναν τεράστιο όγκο input δεδομένων, να τα επεξεργάζεται ταχύτατα και στη συνέχεια ο αλγόριθμος να παράγει τα σωστά output (Ayodele, 2010), (Fan, 2020). Προς το παρόν, ο αριθμός των τεχνητών νευρώνων, δεν μπορεί να πλησιάσει σε αριθμό τους βιολογικούς νευρώνες, με αποτέλεσμα να μην μπορεί να επιτευχθεί τεχνολογικά η ευφυΐα του εγκεφάλου. Η δυσκολία αυτή συναντάται, επειδή τα υπάρχοντα συστήματα hardware περιέχουν συνδέσεις χαλκού με αποτέλεσμα να περιορίζεται η ταχύτητα μεταφοράς των δεδομένων. Επίσης, ο εγκέφαλος έχει συγκεκριμένο μέγεθος για να χωράει στο κεφάλι του ανθρώπου, συγκρινόμενος με ένα τεχνολογικό NN.

Τα Νευρωνικά Δίκτυα μπορούμε να τα διακρίνουμε σε τρεις γενιές ανάλογα με την σειρά εμφάνισής τους.

- Η πρώτη γενιά περιλαμβάνει τεχνικές όπως τα perceptrons, είτε μονά (Single Layer Perceptron), είτε πολλαπλά (Multi Layer Perceptron).
- Η δεύτερη γενιά αποτελείται από τα Convolutional NNs (CNN) και τα Recurrent NNs (RNN) και
- Τέλος η τρίτη γενιά είναι τα Spike NNs (SNN).

Στην εικόνα 5 εμφανίζονται οι διαφορετικές γενιές NN με τα χαρακτηριστικά τους (Nandakumar, 2018), (Ghosh-Dastidar, 2009) και αναλύονται παρακάτω.



Εικόνα 5 . Γενιές Νευρωνικών Δικτύων (Nandakumar, 2018)

1.2.1 Συναρτήσεις Ενεργοποίησης

Για την κατανόηση της λειτουργίας ενός τεχνολογικού νευρώνα γίνεται εισαγωγή του όρου “συνάρτηση ενεργοποίησης” (activation function). Ο ρόλος της activation function είναι να παράγει έξοδο από ένα σύνολο τιμών εισόδου που τροφοδοτούνται σε έναν τεχνολογικό νευρώνα. Η activation function αποφασίζει εάν ένας νευρώνας πρέπει να ενεργοποιηθεί ή όχι. Αυτό σημαίνει ότι χρησιμοποιώντας απλούστερες μαθηματικές πράξεις θα αποφασίσει εάν η είσοδος του νευρώνα στο δίκτυο είναι σημαντική ή όχι στη διαδικασία πρόβλεψης (Nwankpa, 2018). Ανάλογα με τον τύπο του προβλήματος που επιλύεται κάθε φορά, επιλέγεται και η κατάλληλη activation function.

Οι συναρτήσεις ενεργοποίησης διακρίνονται σε γραμμικές και μη-γραμμικές. Στην συνέχεια περιγράφονται εν συντομία όσες από αυτές χρησιμοποιούνται στην παρούσα διπλωματική εργασία (Sharma, 2017), (Nwankpa, 2018), (Ding, 2018, June).

Γραμμικές Συναρτήσεις Ενεργοποίησης

- **Step Activation Function**

Η συγκεκριμένη συνάρτηση ενεργοποίησης στην έξοδο παράγει δυαδικό αποτέλεσμα (0,1) ανάλογα με την είσοδο που δέχεται. Στην περίπτωση που η έξοδος είναι το 0, τότε η πληροφορία δεν μεταβαίνει στον επόμενο νευρώνα, αντίθετα αν είναι ίση με 1, τότε το αποτέλεσμα αυτού του νευρώνα μεταφέρεται και στους επόμενους.
$$f(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases}$$

- **Linear/ identity Activation Function**

Σε αυτή την περίπτωση ο νευρώνας ότι είσοδο λαμβάνει, την βγάζει και σαν έξοδο. $f(x) = x$

Μη-Γραμμικές Συναρτήσεις Ενεργοποίησης (Ding, 2018, June)

Ο βασικότερος λόγος για τον οποίο χρειάζονται οι μη-γραμμικές activation function είναι διότι παρά την μεγάλη πολυπλοκότητά τους μπορούν βοηθούν τους νευρώνες στην εκπαίδευση και την αναγνώριση περίπλοκων μοτίβων δεδομένων (Sharma, 2017)

- **Sigmoid Activation Function**

Αυτή η συνάρτηση δέχεται οποιαδήποτε πραγματική τιμή ως είσοδο και εξάγει τιμές στην περιοχή από 0 έως 1 (εικόνα 6α). Όσο μεγαλύτερη είναι η είσοδος, τόσο πιο κοντά στο 1 θα είναι η τιμή εξόδου, ενώ όσο μικρότερη είναι η είσοδος, τόσο πιο κοντά στο 0 θα είναι η έξοδος, όπως φαίνεται παρακάτω.
$$f(x) = \frac{1}{1+e^{-x}}, \quad x \in (-\infty, +\infty), \quad f \in (0,1)$$

- **Tanh Activation Function**

Η συνάρτηση Tanh είναι παρόμοια με τη συνάρτηση σιγμοειδούς ενεργοποίησης, και έχει ακόμη και το ίδιο σχήμα S (εικόνα 6β) που διαφέρει όμως στην περιοχή εξόδου από -1 έως 1. Στην Tanh, όσο μεγαλύτερη είναι η είσοδος (πιο θετική), τόσο η τιμή εξόδου θα πλησιάζει στην τιμή 1, ενώ όσο μικρότερη είναι η είσοδος (πιο αρνητική), τόσο η τιμή εξόδου θα πλησιάζει στην τιμή -1.

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

- **Rectified Linear Unit (ReLU) Activation Function**

Η ReLU είναι πιο αποδοτική υπολογιστικά από το tanh και την sigmoid επειδή περιλαμβάνει απλούστερες μαθηματικές πράξεις. Κάθε φορά ενεργοποιούνται μόνο λίγοι νευρώνες,

καθιστώντας το δίκτυο αραιό, αποτελεσματικό και εύκολο στον υπολογισμό. Δίνει έξοδο x αν το x είναι θετικό και 0 σε οποιαδήποτε άλλη περίπτωση (εικόνα 6γ) . $f(x) = \max(0, x)$

Μερικές ακόμα συχνά χρησιμοποιούμενες activation functions είναι οι Leaky Relu, softmax, Exponential Linear Unit.

Σημαντικό ρόλο στην βελτιστοποίηση των NN παίζουν και οι πρώτες παράγωγοι των παραπάνω (μη γραμμικών) activation functions . Με την βοήθειά τους ο αλγόριθμος επιτυγχάνει υψηλότερη απόδοση κατά την διαδικασία πρόβλεψης του αποτελέσματος. Οι πρώτες παράγωγοι των παραπάνω μη-γραμμικών activation function δίνονται παρακάτω.

- **Sigmoid Activation Function derivative** (εικόνα 6α)

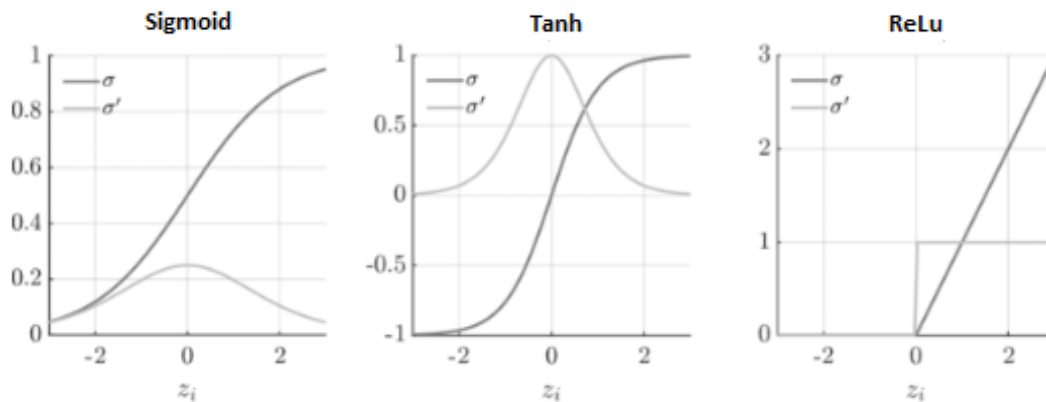
$$\frac{d}{dx} \text{sigmoid}(x) = \text{sigmoid}(x)(1 - \text{sigmoid}(x))$$

- **Tanh Activation Function derivative** (εικόνα 6β)

$$\frac{d}{dx} \tanh(x) = 1 - \tanh^2(x)$$

- **ReLU Activation Function derivative** (εικόνα 6γ)

$$\frac{d}{dx} \text{relu}(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases}$$



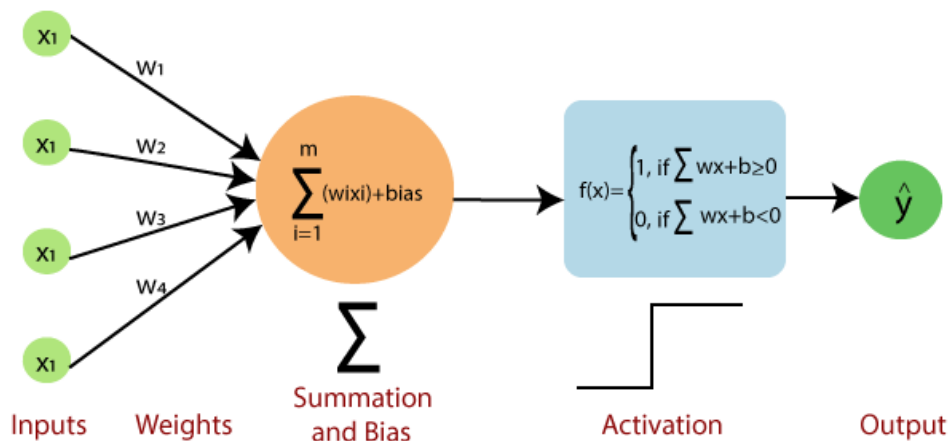
Εικόνα 6. Γραφήματα των πιο συχνά χρησιμοποιούμενων activation functions και των πρώτων παραγώγων τους (Ostwald, 2021)

Είναι μεγάλης σημασίας για την επίλυση ενός προβλήματος να επιλεγθεί η καταλληλότερη activation function που συνήθως γίνεται ανάλογα με τον τύπο του NN (Sharma, 2017). Στην περίπτωση που το NN είναι RNN, γίνεται χρήση Tanh και Sigmoid, ενώ αν το NN είναι CNN γίνεται χρήση RELU activation function. Σύμφωνα με το (Sharma, 2017) ,

- Οι sigmoid συναρτήσεις λειτουργούν καλύτερα στην περίπτωση των classifiers
- Οι συναρτήσεις sigmoid και tanh μερικές φορές αποφεύγονται λόγω του προβλήματος vanishing gradient
- Στα hidden layers δεν πρέπει να γίνεται χρήση των sigmoid ή tanh
- Η ReLU είναι μια γενική συνάρτηση ενεργοποίησης και χρησιμοποιείται στις περισσότερες περιπτώσεις και μόνο στα hidden layers ενός NN.
- Προκειμένου να βρεθεί η βέλτιστη απόδοση του αλγορίθμου γίνεται εξ αρχή χρήση της ReLU και στη συνέχεια, αν η ReLU δεν παρέχει τα βέλτιστα αποτελέσματα, δοκιμάζονται διαφορετικές activation function.

1.2.2 Single Layer Perceptron (SLP)

Ένας τεχνητός νευρώνας είναι βασισμένος σε μαθηματικές μοντελοποιήσεις. Το single layer perceptron (SLP) που εισήχθη σαν έννοια από τον Rosenbaltt (1957) είναι από τις πρώτες και πιο απλές δομές που χρησιμοποιήθηκαν για την αναπαράσταση ενός νευρώνα (Singh, 2016). Τα προβλήματα που καλούνταν να λύσει ήταν κυρίως η αναγνώριση γραμμικών δεδομένων (Brunner D. P., 2018). Ένα SLP αποτελείται (εικόνα 7) από τα input δεδομένα x_m τα οποία πολλαπλασιάζονται με αντίστοιχα βάρη w_m . Στην συνέχεια, στο άθροισμα των $\sum x_m * w_m$ προστίθεται μία bias τιμή και μετά από κατάλληλη μαθηματική επεξεργασία, παράγεται μια έξοδος. Η μαθηματική επεξεργασία των παραπάνω ονομάζεται step activation function (Mahesh, 2020), (Ghosh-Dastidar, 2009). Όπως φαίνεται στην (εικόνα 7) το αποτέλεσμα είναι δυαδικό (0,1) ανάλογα με την τελική τιμή της γραμμικής συνάρτησης $\sum x_j * w_j + b$ (Nandakumar, 2018). Όταν η τιμή της γραμμικής παράστασης είναι ≤ 0 τότε το $output = 0$ (ο νευρώνας δεν παράγει έξοδο), ενώ όταν η παράσταση αποκτήσει τιμή > 0 , το $output = 1$, δηλαδή ο νευρώνας ενεργοποιείται και μεταφέρει πληροφορία στην έξοδο (Ghosh-Dastidar, 2009).

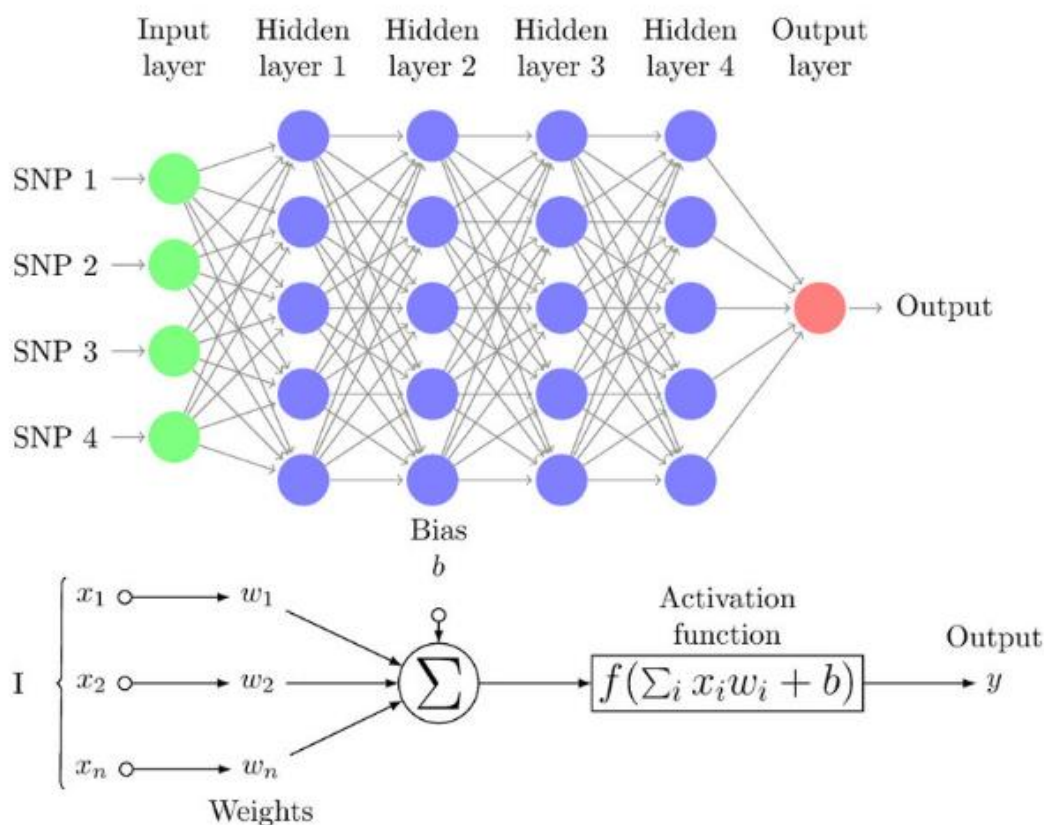


Εικόνα 7. Single layer perceptron με step activation function (Mahesh, 2020)

1.2.3 Multi Layer Perceptron (MLP)

Μία μετεξέλιξη της ιδέας του SLP ήταν τα multilayer perceptron (MLP) που στην βιβλιογραφία (Lukoševičius M. &, 2009), (Gardner, 1998) αναφέρονται και σαν feed-forward NN (FFNN). Η αρχιτεκτονική τους βασίζεται στην ένωση πολλών perceptron μεταξύ τους, με σκοπό την δημιουργία ενός δικτύου νευρώνων (Brunner D. P., 2018). Όπως αναφέρει και ο Gardner, η εκπαίδευση των MLP

γίνεται με supervised τρόπο (Gardner, 1998). Η παρακάτω διαδικασία περιγράφει τη ροή της πληροφορίας σε ένα FFNN. Το δίκτυο αποτελείται από το input layer που δέχεται τα δεδομένα, τα εσωτερικά hidden layers νευρώνων (ο αριθμός των οποίων καθορίζεται εξ αρχής) και το output layer. Όπως αναφέρθηκε στην περίπτωση των SLP, τα δεδομένα εισόδου κάθε νευρώνα του input layer πολλαπλασιάζονται με αντίστοιχα βάρη. Στη συνέχεια, όπως φαίνεται στην εικόνα 8, οι νευρώνες του input layer συνδέονται με τους νευρώνες του πρώτου hidden layer (Ayodele, 2010) όπου γίνεται κατάλληλη επεξεργασία. Με την βοήθεια activation function (f), (εικόνα 8) ενεργοποιείται ο κάθε νευρώνας και ωθεί την τιμή της παράστασης f σαν είσοδο στους νευρώνες του επόμενου hidden layer με τους οποίους συνδέεται. Οι έξοδοι των νευρώνων στο τελευταίο hidden layer, ωθούνται με παρόμοια επεξεργασία στον νευρώνα του output layer (Klibisz, 2016), δημιουργώντας έτσι μία πρόβλεψη του τελικού output με ένα περιθώριο σφάλματος.



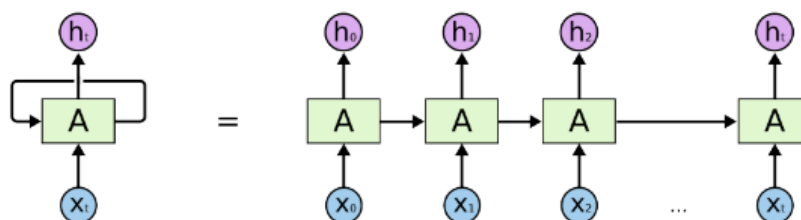
Εικόνα 8. Γράφημα ενός Multi-layer Perceptron με δύο hidden layers και απεικόνιση της μαθηματικής διαδικασίας που ακολουθείται για την παραγωγή εξόδων (Pérez-Enciso, 2019)

Για να ελαχιστοποιηθεί το περιθώριο σφάλματος που προκύπτει, το σύστημα “περνάει” από επαναλαμβανόμενη εκπαίδευση που ονομάζεται Back Propagation (BP) (Ayodele, 2010), (Basodi, 2020), (Gardner, 1998), αλλάζοντας τα βάρη κάθε φορά με την βοήθεια της μεθόδου gradient descent (Guo J. , 2013). Κατά τη διάρκεια κάθε επανάληψης, κάθε ένα από τα βάρη του νευρωνικού δικτύου ενημερώνεται ανάλογα με τη μερική παράγωγο της συνάρτησης σφάλματος σε σχέση με το τρέχον βάρος (Basodi, 2020), (Guo J. , 2013) . Τα MLP, σε αντίθεση με τα SLP που δέχονται μόνο γραμμικά μοτίβα στην είσοδό τους, μπορούν να δεχτούν και μη γραμμικά (Jaeger H. , 2002).

1.2.4 Recurrent Neural Networks (RNN)

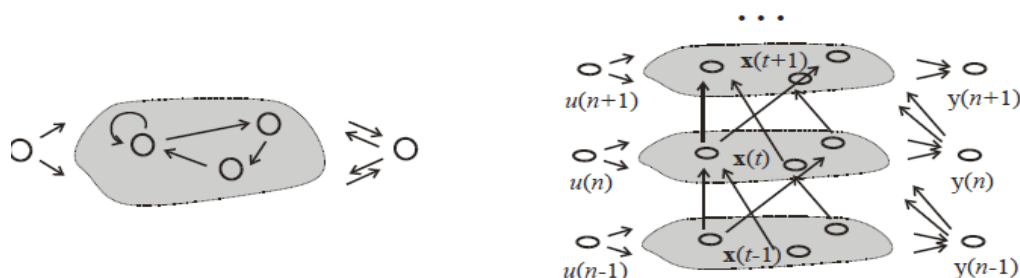
Μια από τις πιο δημοφιλείς κατηγορίες νευρωνικών δικτύων που ανήκουν στην δεύτερη γενιά, είναι και αυτή των Recurrent Neural Networks (RNN).

Όπως αναφέρεται στο (Ayodele, 2010) , τα RNN είναι αρχιτεκτονικά όμοια με τα MLP (input, hidden και output layers καθώς και ίδιες activation functions) με την διαφορά ότι, όπως φαίνεται και στην *εικόνα 9*, οι συνδέσεις των κόμβων των hidden layers (A) μπορούν να δημιουργούν αναδρομικές συνδέσεις. Με αυτό τον τρόπο η πληροφορία μιας ακολουθιακής χρονοσειράς αξιοποιείται χωρίς να χάνεται (Murugan, 2018). Όπως φαίνεται και στο (Tanaka, 2019), εκτός από τις αναδρομικές σχέσεις, μία άλλη διαφορά των MLP feed-forward με τα RNN είναι ότι τα RNN έχουν την δυνατότητα να επεξεργαστούν δυναμικά δεδομένα, που παρουσιάζουν δηλαδή αλλαγές στον χρόνο (temporal data). Συνεπώς, τα RNN συνιστώνται για δυναμική επεξεργασία δεδομένων, όπως αναγνώριση ακολουθιών γραπτού κειμένου και λόγου. Αυτό είναι εφικτό, επειδή οι αναδρομές στα hidden layers δημιουργούν ένα είδος μνήμης προηγούμενων καταστάσεων στο NN (Abiodun, 2018). Στην παρούσα διπλωματική τα NN που θα μας απασχολήσουν είναι τύπου RNN.



Εικόνα 9. Επεξεργασία πληροφορίας σε ένα NN που διαθέτει αναδρομές (S. K. Saksena, 2019)

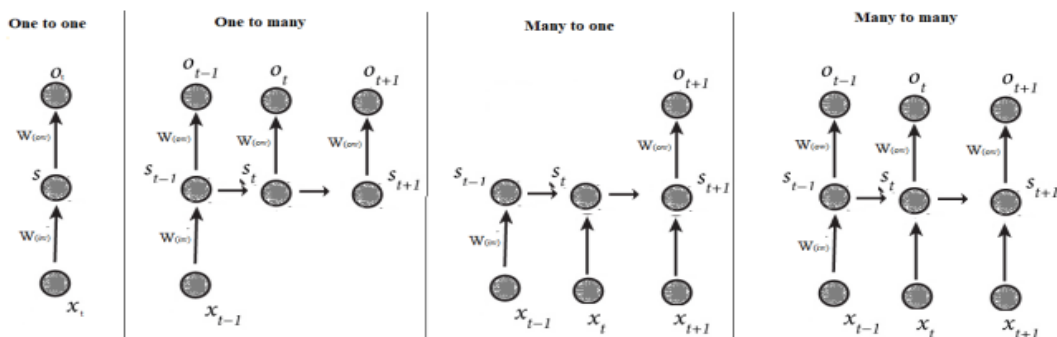
Σύμφωνα με τον Jaeger (Jaeger H. , 2002) ο BP αλγόριθμος δεν μπορεί να λειτουργήσει σωστά στα RNN, διότι προϋποθέτει ότι στις συνδέσεις μεταξύ των νευρώνων δεν υπάρχει η έννοια της αναδρομής. Λύση σε αυτό το πρόβλημα έδωσε ο Werbos (Werbos, 1990) εισάγοντας τον αλγόριθμο Back Propagation Through Time (BPTT) ο οποίος “ξεδιπλώνει” τις αναδρομές (*εικόνα 9*) του RNN στην πορεία του χρόνου. Δημιουργείται μια στοίβα από πανομοιότυπα αντίγραφα του RNN (*εικόνα 10*) και ανακατευθύνει τις συνδέσεις εντός του δικτύου με σκοπό να πετύχει συνδέσεις ανάμεσα στα διαδοχικά αντίγραφα. Έτσι δημιουργείται FFNN δίκτυο και συνεχίζεται η εκπαίδευση με τον αλγόριθμο BP (Jaeger H. , 2002).



Εικόνα 10. Διαδικασία “ξεδιπλώματος” των αναδρομών σε ένα RNN κατά jaeger (Jaeger H. , 2002).

Παράδειγμα η λέξη EAN. Το γράμμα E μπαίνει σαν είσοδος την χρονική στιγμή $t-1$ και θγαίνει σαν έξοδος. Στη συνέχεια την χρονική στιγμή t στο σύστημα υπάρχει το E και μπαίνει και το A. Σαν έξοδος θγαίνει EA την χρονική στιγμή t . Τελικώς, στο σύστημα την χρονική στιγμή $t+1$ μπαίνει το EA και το γράμμα N, γίνεται η ανάλογη επεξεργασία και σαν output παράγεται η χρονοσειρά EAN.

Ανάλογα την εφαρμογή, ένα RNN μπορεί να “ξεδιπλώσει” τις αναδρομές με τις αρχιτεκτονικές της *εικόνας 11*. Κάθε μία από τις παρακάτω αρχιτεκτονικές εξυπηρετεί την επίλυση πολύπλοκων προβλημάτων με state-of-the-art απόδοση (Murugan, 2018).



Εικόνα 11. Παρουσίαση διάφορων τύπων με τους οποίους μπορεί να “ξεδιπλωθούν” οι αναδρομές σε ένα RNN (Murugan, 2018)

Οι παραπάνω τύποι RNN έχουν διάφορες εφαρμογές όπως για κατηγοριοποίηση εικόνας (one to one), για παραγωγή μουσικής και για Image Captioning (one to many), για αξιολογήσεις και κριτικές σε διαδικτυακές σελίδες (many to one) όπως στην *εικόνα 12* και για μετάφραση κειμένου (many to many) πχ google translate (Ketkar, 2021).

Review (X)	Rating (Y)
"This movie is fantastic! I really like it because it is so good!"	★★★★☆
"Not to my taste, will skip and watch another movie"	★★☆☆☆
"This movie really sucks! Can I get my money back please?"	★☆☆☆☆

Εικόνα 12. κριτική σε διαδικτυακή σελίδα

Στον παρακάτω πίνακα (*εικόνα 13*) παρουσιάζονται τα πλεονεκτήματα καθώς και τα μειονεκτήματα της χρήσης ενός RNN (Sharkawy, 2020).

Type of NN	Advantages	Disadvantages
RNN	- The RNN is the dynamic neural network [1, 30] and is computationally powerful and can be used in many temporal processing models and applications [51]. - RNNs possess the so-called universal approximation property [52], that is, they are capable of approximating arbitrary nonlinear dynamical systems (under loose regularity conditions) with arbitrary precision, by realizing complex mappings from input sequences to output sequences [53]. - The RNN size is significantly compact compared with feedforward networks for the same approximation accuracy of dynamic systems [51]. - Recurrent networks can be used as associative memories to build attractors from input-output association [51]. - The recurrent neural network is widely used in human-robot interaction (HRI) tasks as presented in [54-57].	- Although, the recurrent network is a simple and powerful model, in practice, it is hard to train properly [58]. - Among the main reasons why this model is so unwieldy are the vanishing gradient and exploding gradient problems described in [59]. - Due to the nonlinear nature of unit activation output characteristics and the weight adjustment strategies, the network stability is often difficult to ascertain [30]. - It cannot process very long sequences if using tanh or relu as an activation function.

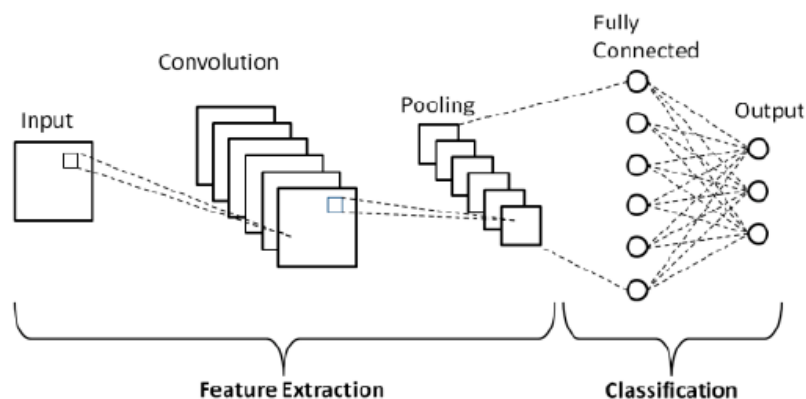
Εικόνα 13. Τα κύρια πλεονεκτήματα και μειονεκτήματα του RNN (Sharkawy, 2020)

Συνοπτικά αναφέρεται (Xu., 2020, March), ότι τα RNN παρουσιάζουν δυσκολία στην εκπαίδευση, δεν είναι σε θέση να χειριστούν μεγάλες ακολουθίες κάνοντας χρήση tanh ή ReLu και επίσης είναι δυνατό να εμφανιστεί το exploding ή το vanishing gradient πρόβλημα που αφορά την εκθετική ή την ελάχιστη μεταβολή των βαρών του μοντέλου. Αντιθέτως, στα θετικά συγκαταλέγεται, ότι οι τιμές των βαρών μεταφέρονται μεταξύ των διάφορων επιπέδων του RNN και το μέγεθος του μοντέλου παραμένει σταθερό ανεξάρτητα από τον αριθμό των input δεδομένων.

1.2.5 Convolutional Neural Networks (CNN)

Στην ίδια γενιά με τα RNNs ανήκουν και τα CNN τα οποία βρίσκουν εφαρμογή σε προβλήματα ταξινόμησης εικόνων (Image Classification), ανίχνευσης αντικειμένων (Object Detection), τμηματοποίησης εικόνας (Image Segmentation) και αναγνώρισης προσώπου (Face Recognition) (Li, 2021), (Schaetti, 2016, August).

Στους αλγόριθμους CNN τα διάφορα επίπεδα ταξινομούνται σε δύο κατηγορίες όπως απεικονίζονται στο σχήμα της εικόνας 14. Η πρώτη κατηγορία αποτελείται από το input layer, το convolution layer και το pooling layer, που περιέχουν φίλτρα που σκοπό έχουν την αναγνώριση του μοτίβου (pattern recognition). Τα pooling layers σκοπό έχουν την μείωση του μεγέθους της εικόνας μεταξύ των hidden layers (Sakib, 2019). Η δεύτερη κατηγορία χρησιμοποιεί την πληροφορία των προηγούμενων layers και ομαδοποιεί με την κατάλληλη κλάση τα μοτίβα που λαμβάνει. Η διαδικασία του classification αποτελείται από ένα Fully Connected και ένα output επίπεδο.

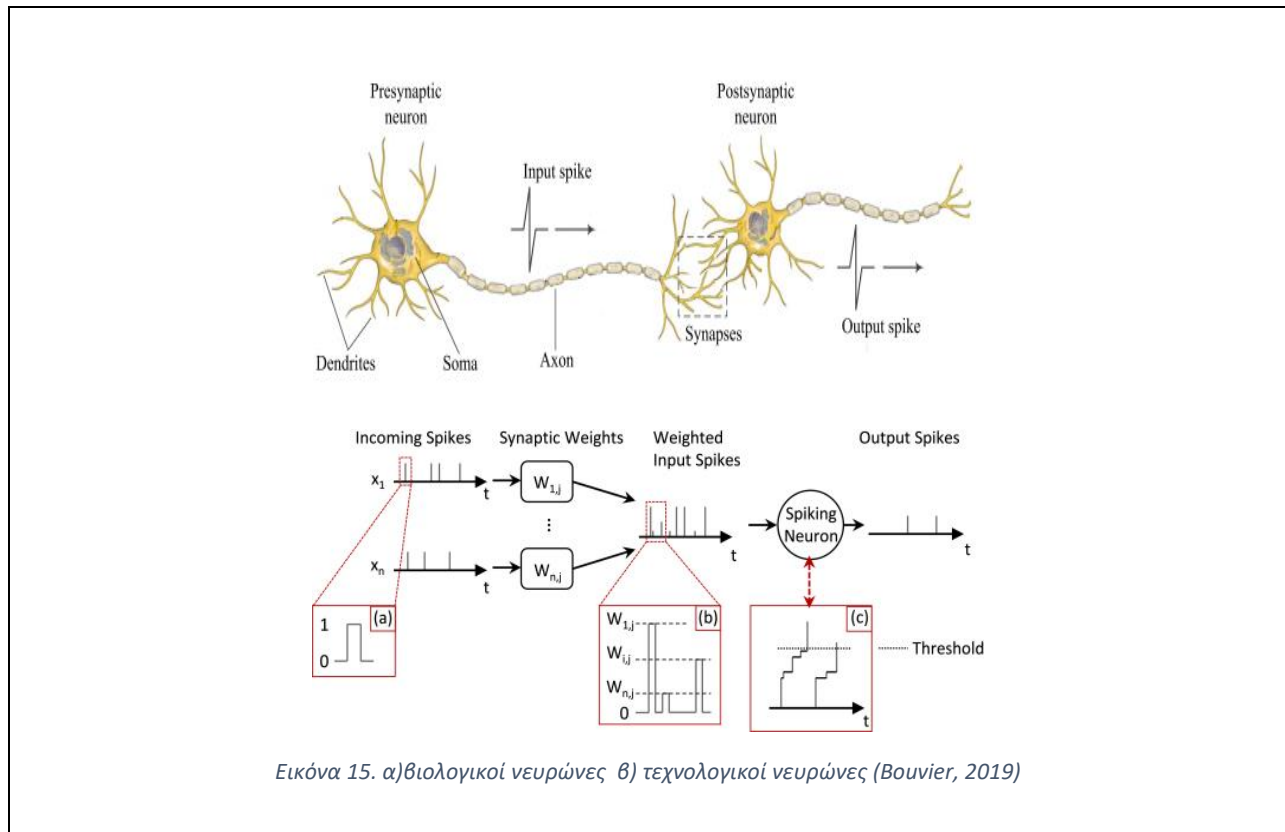


Εικόνα 14. Δομή ενός Convolutional NN (Phung, 2019)

Η βασική διαφορά των CNN με τα RNN βρίσκεται στο ότι τα CNN είναι μία κατηγορία των Deep NN, που εφαρμόζονται πιο συχνά για την ανάλυση οπτικών εικόνων (spatial data). Ένα RNN είναι μια κατηγορία Artificial NN που οι συνδέσεις μεταξύ κόμβων σχηματίζουν ένα κατευθυνόμενο γράφημα κατά μήκος μιας χρονικής ακολουθίας.

1.2.6 Spiking Neural Networks (SNN)

Στην τρίτη γενιά NN ανήκουν τα Spiking Neural Networks (SNN). Όπως αναφέρθηκε παραπάνω, οι βιολογικοί νευρώνες του εγκεφάλου επικοινωνούν μεταξύ τους παράγοντας και διαδίδοντας ηλεκτρικούς παλμούς (spikes), με την βοήθεια χημικών διεργασιών, μεταφέροντας έτσι αποδοτικά και με μεγάλη ταχύτητα την πληροφορία. Η εξέλιξη στον τομέα των artificial neural networks (ANN) οδήγησε στην προσπάθεια για προσομοίωση της διαδικασίας που πραγματοποιείται στον εγκέφαλο (Noriega, 2005). Για την επίτευξη αυτού του σκοπού δημιουργήθηκαν τα spiking neural networks (SNN) (Ponulak, 2011). Η μεταδιδόμενη πληροφορία συνήθως κωδικοποιείται με βάση την συχνότητα των spike (rate encoding) και/ή την χρονική στιγμή που συμβαίνουν spikes (pulse encoding) (Ghosh-Dastidar, 2009).



Οι νευρώνες από spikes συμπεριφέρονται παρόμοια με τους νευρώνες πρώτης γενιάς, με την διαφορά ότι οι spike νευρώνες έχουν μια έμφυτη δυναμική φύση που χαρακτηρίζεται από μια εσωτερική κατάσταση που αλλάζει με τον χρόνο (Ghosh, 2009). Κάθε επόμενος (postsynaptic) νευρώνας στο άκρο μιας σύναψης εκπυρσοκροτεί ένα ηλεκτρικό παλμό την χρονική στιγμή που η εσωτερική κατάστασή του υπερβαίνει ένα συγκεκριμένο κατώφλι (threshold). Παρόμοια με τους βιολογικούς νευρώνες (εικόνα 15α), οι τεχνολογικοί νευρώνες δεν περιέχουν πληροφορία (input-output) στα spikes τους, αλλά όλη η πληροφορία κωδικοποιείται την χρονική στιγμή που γίνονται τα spikes (Ghosh-Dastidar, 2009). Όπως φαίνεται και στην εικόνα 15β τα spikes πολλαπλασιάζονται με τα βάρη, αθροίζονται και δίνουν έναν τετραγωνικό παλμό πληροφορίας. Κατά την είσοδο αυτού του παλμού στον spiking neuron, κωδικοποιείται η πληροφορία και όταν ξεπεράσουν κάποιο threshold ο νευρώνας ενεργοποιείται και βγάζει μία spiking έξοδο.

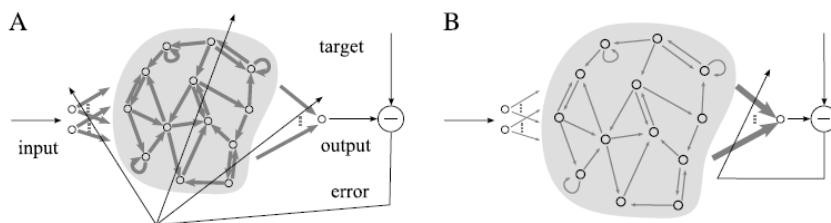
2 RESERVOIR COMPUTING NETWORKS (RCN) – ECHO STATE NETWORKS (ESN)

Όλοι οι προαναφερθέντες αλγόριθμοι και τεχνικές, χρησιμοποιούνται για να υπάρχει καλύτερη απόδοση στα συστήματα MM. Σε προβλήματα όπως η αναγνώριση φωνής ή η πρόβλεψη χρονοσειρών στη χρηματοπιστωτική αγορά, τα input data είναι χρονικές ακολουθίες τιμών. Σε αυτή την περίπτωση χρησιμοποιούνται τα RNN (Schaetti, 2016, August) τα οποία συνήθως είναι πολύ δυνατά εργαλεία για την επίλυση χρονικά πολύπλοκων προβλημάτων. Σε ένα τυπικό RNN, ο κάθε κόμβος μέσα στα hidden layers συνδέεται με κάθε άλλο κόμβο (Schaetti, 2016, August). Ένα από τα μειονεκτήματα των RNN είναι ότι απαιτούν μεγάλη υπολογιστική ισχύ κατά την διάρκεια της εκπαίδευσης τους, με αποτέλεσμα να είναι αργή η επίλυση του προβλήματος (slow convergence) (Schrauwen, 2007), (Lukoševičius M. &, 2009), (Lukoševičius M. J., 2012). Ο κύριος λόγος είναι, ότι υπάρχει δυσκολία στην εκπαίδευση ενός RNN με μεθόδους βασισμένες σε τεχνικές gradient-descent, οι οποίες στοχεύουν στην συνεχή μείωση του σφάλματος εκπαίδευσης. Έτσι, η χρήση των RNN σε μη-γραμμικά μοντέλα παρέμεινε περιορισμένη για μεγάλο χρονικό διάστημα (Lukoševičius M. &, 2009).

2.1 Ιστορική Αναδρομή

Η προαναφερθείσα δυσκολία ξεπεράστηκε με ένα νέο τρόπο προσέγγισης του σχεδιασμού και της εκπαίδευσης ενός RNN. Το 2001 ο Jaeger, εργαζόμενος στον τομέα MM, πρότεινε την προσέγγιση των Echo State Networks (ESN) (Jaeger H, 2001), (Klibisz, 2016). Το 2002 ο Maass, εργαζόμενος στον τομέα του Computational Neuroscience, πρότεινε την προσέγγιση των Liquid State Machines (LSM) (Maass, 2002). Το 2005 ο Schiller και ο Steil πρότειναν την προσέγγιση Backpropagation Decorrelation (BPDC) που βασίζεται στα ESN και LSM (Klibisz, 2016). Επιπλέον, οι Schiller και Steil (2005) έδειξαν ότι στις παραδοσιακές μεθόδους εκπαίδευσης για RNN, όπου όλα τα βάρη αναπροσαρμόζονται, οι κυρίαρχες αλλαγές συμβαίνουν στα βάρη εξόδου (Schiller, 2005). Αυτό πρακτικά σημαίνει, ότι υπάρχει ανάγκη για εκπαίδευση μόνο στο εξωτερικό readout-layer και όχι σε όλο το RNN (Konkoli, 2018). Αργότερα (2007) ο Verstraeten πρότεινε την ενοποίηση των ESN και LSM κάτω από τον όρο “reservoir computing” (RC) (Schrauwen, 2007), (Van der Sande, 2017).

Αναπτύχθηκε έτσι η μέθοδος του “reservoir”, δηλαδή η ύπαρξη ενός δικτύου νευρώνων στο μοναδικό hidden layer ενός RNN, όπου υπάρχουν τυχαίες και πιο αραιές συνδέσεις μεταξύ των κόμβων του (Schaetti, 2016, August). Το reservoir διεγείρεται από το μη-γραμμικό σήμα εισόδου και διατηρεί στο εσωτερικό του ένα μη-γραμμικό μετασχηματισμό των παρελθοντικών τιμών εισόδου του. Το επιθυμητό σήμα εξόδου (output) παράγεται με την τεχνική του linear regression, σαν γραμμικός συνδυασμός των σημάτων που εξέρχονται από τους νευρώνες του reservoir (Lukoševičius M. &, 2009). Παρακάτω στην εικόνα 16, φαίνεται ο τρόπος εκπαίδευσης ενός RNN σε αντιδιαστολή με ένα RC. Στα RNN η εκπαίδευση γίνεται με αναπροσαρμογή των βαρών σε όλο το NN ενώ στο RCN γίνεται μόνο στο output layer.



Εικόνα 16. Διαφορετικοί τρόποι εκπαίδευσης α) ενός RNN β) ενός RCN (Lukoševičius M. &, 2009)

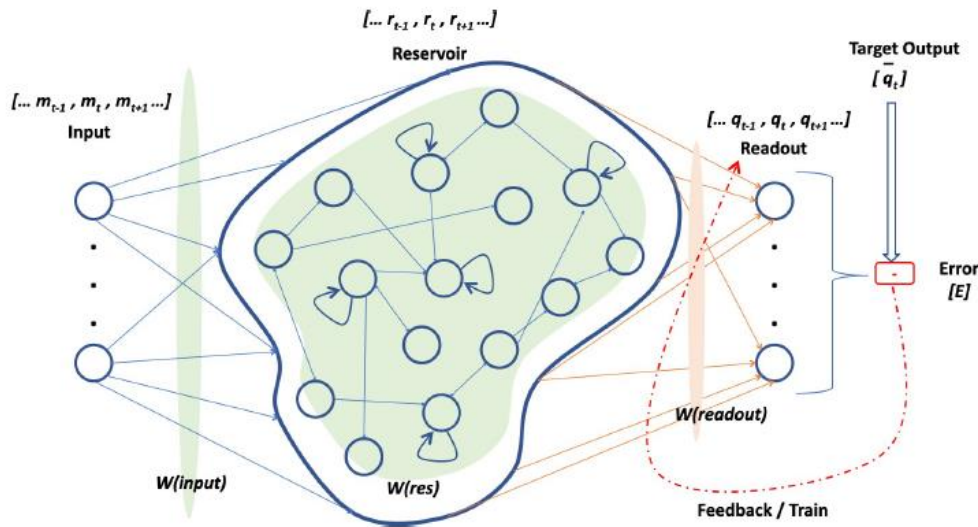
2.2 Reservoir Computing

Η τεχνική του Reservoir Computing (RC) έχει δει μεγάλη ανάπτυξη (Jaeger H. M., 2007) και αποτελεί πλέον μια από τις πιο βασικές μεθόδους μοντελοποίησης ενός RNN (Jaeger H. , 2007). Σύμφωνα με το (Lukoševičius M. &, 2009), (Lukoševičius M. J., 2012), η ανάπτυξη της οφείλεται στους παρακάτω λόγους.

- **Ακρίβεια μοντελοποίησης** : Η τεχνική RC έχει ξεπεράσει τις επιδόσεις προηγούμενων μεθόδων σε μη γραμμικά συστήματα αναγνώρισης, πρόβλεψης και ταξινόμησης. Για παράδειγμα, στην Ιαπωνική γλώσσα, ο συγκριτικός δείκτης φωνήεντος (vowel benchmark) έχει βελτιωθεί (πχ. με χρήση RC το ποσοστό σφάλματος δοκιμής είναι μηδενικό, ενώ προηγουμένως ήταν 1,8%).
- **Ικανότητα μοντελοποίησης** : Το RC είναι υπολογιστικά παγκόσμιο για συστήματα πραγματικού χρόνου (συνεχούς χρόνου και αξίας).
- **Βιολογική αληθοφάνεια** : Είναι πολλές οι συσχετίσεις που έχουν γίνει μεταξύ του τρόπου λειτουργίας του RC και του εγκεφάλου των θηλαστικών. Με την βοήθεια των RC εξηγήθηκαν κάποια ερωτήματα σχετικά με το πως λειτουργεί ο ανθρώπινος εγκέφαλος ο οποίος μπορεί να εκτελέσει ακριβείς υπολογισμούς (πχ accurate timing), όταν τα δεδομένα εισόδου του προέρχονται από ένα φυσικό υπόστρωμα (substrate) με θόρυβο.
- **Επεκτασιμότητα και οικονομία** : Το RC απαντά στην ερώτηση “πως μπορούν να επεκταθούν τα παλιότερα μοντέλα με νέα στοιχεία χωρίς να καταστραφούν αναπαραστάσεις (output) προηγούμενης εκπαίδευσης”. Τα νέα στοιχεία αντιπροσωπεύονται από νέα output units, που προστίθενται στα προηγούμενα αποτελέσματα ενός συγκεκριμένου reservoir.

Επιπλέον, στο άρθρο (Tanaka, 2019) ο Tanaka αναφέρει ότι η **απλότητα της μεθόδου εκπαίδευσης** ενός RC το έκανε ελκυστικό και για developers χωρίς κάποια εξειδίκευση πάνω στον τομέα των νευρωνικών δικτύων.

Το RC όπως και κάθε άλλο RNN είναι κατάλληλο εκ φύσεως μόνο για την επεξεργασία χρονικών δεδομένων όπου οι τιμές του σήματος εισόδου σε κάθε χρονική στιγμή είναι αλληλεξαρτούμενες. Γι' αυτό το λόγο, το RC δεν είναι κατάλληλο για μοντελοποίηση στατικών δεδομένων. Σύμφωνα με τα (Martinuzzi, 2020), (Tanaka, 2019), (Klibisz, 2016), (Roli, 2014) ένα RC προκειμένου να παράξει ένα αποτέλεσμα, αντιστοιχίζει μη-γραμμικά σήματα εισόδου (χρονοσειρές) σε ένα πολυδιάστατο μη-γραμμικό υπολογιστικό χώρο (reservoir) με αναδρομικούς νευρώνες (εικόνα 17). Οι συνδέσεις τους αρχικοποιούνται τυχαία και παραμένουν σκόπιμα σταθερές μη επιδεχόμενες εκπαίδευση. Επιπλέον το RC αποτελείται από ένα readout layer, το οποίο αναλύει διάφορα μοτίβα που προκύπτουν από τα δεδομένα της παραπάνω αντιστοίχισης και εκπαιδεύεται με την μέθοδο linear regression και classification προκειμένου να παράξει το επιθυμητό output. Κατά την δημιουργία ενός reservoir πρέπει να επιλεγθεί η κατάλληλη activation function προκειμένου να μπορούν οι νευρώνες να μετασχηματίσουν την πληροφορία που λαμβάνουν (Roli, 2014), (Ferreira, 2008, September).



Εικόνα 17. Δομή ενός reservoir (Budhiraja, 2021)

Ένα reservoir λειτουργεί σαν μη-γραμμική πολυδιάστατη επέκταση του σήματος εισόδου και ταυτόχρονα σαν μνήμη της πληροφορίας εισόδου, παρέχοντας χρονικό πλαίσιο. Ωστόσο, προκειμένου ένα πολύπλοκο σύστημα να χαρακτηριστεί “καλό reservoir” πρέπει να τηρεί κάποιες προϋποθέσεις και για την υλοποίησή του θα πρέπει να γίνει πρώτα κατανοητό ποια λειτουργία θα επιτελέσει (Lukoševičius M., 2012).

Στο άρθρο (Seoane, 2019) αναφέρεται ότι οι Jaeger και Maass, με την εισαγωγή της τεχνικής του RC, ανέπτυξαν θεωρήματα προκειμένου να αποδείξουν ποιες είναι εκείνες οι προϋποθέσεις που πρέπει να πληροί ένα reservoir (πχ. echo-state, fading memory, separation property κ.α) ώστε να υλοποιείται ορθά. Ενώ τα θεωρήματα απαιτούσαν τις παραπάνω προϋποθέσεις, στην πράξη δημιουργήθηκαν συστήματα reservoir που λειτουργούν αρκετά καλά χωρίς την αυστηρή τήρησή τους. Ένα καλό reservoir είναι σε θέση να διαχωρίσει τις εισόδους. Διαφορετικές εξωτερικές εισοδοί θα πρέπει να οδηγούν το σύστημα σε διαφορετικές περιοχές του χώρου διαμόρφωσης (Konkoli, 2018). Αυτή η συνθήκη ονομάζεται separability. Επιπλέον, μια σημαντική ιδιότητα για τα reservoir των ESN είναι η Echo State Property (Yildiz, 2012) που δηλώνει ότι οι εισοδοί του reservoir αφήνουν echo στο σύστημα για πάντα ή μέχρι να “σβήσουν” σταδιακά (Martinuzzi, 2020). Η echo state ιδιότητα θα αναλυθεί εκτενώς σε επόμενη ενότητα.

2.3 Βιβλιογραφική Αναφορά των RC Εφαρμογών

Πολλά είναι τα άρθρα στην βιβλιογραφία που αναφέρονται σε επιτυχημένες εφαρμογές των RC. Αρχικά, όταν τα RC εισήλθαν σαν έννοια το 2007, υπήρχαν ήδη εφαρμογές για την επίλυση προβλημάτων με χρονική κατηγοριοποίηση (temporal classification), regression ή prediction, καθώς και πιο τεχνικές (real-world) εφαρμογές, όπως στον τομέα της ρομποτικής (Schrauwen, 2007), (Verstraeten D. S., 2007). Οι Lukosevicius και Jaeger (2009) στο (Lukoševičius M. &, 2009) αναφέρουν ότι οι εφαρμογές των μεθόδων RC ήταν πλέον γνωστές. Προκειμένου να βελτιωθεί η αποτελεσματικότητα μίας μεθόδου, αναγκαία προϋπόθεση είναι η βασική κατανόηση βασικών αρχών της MM (εμπειρία, πειραματισμός και ενδοσκόπηση στη φύση του προβλήματος) (Lukoševičius M. J., 2012). Η συνεχής μελέτη και η έρευνα που έγινε πάνω στα RC μέχρι το 2019, όπως φαίνεται στο άρθρο (Tanaka, 2019), οδήγησε στην ανάπτυξη

τροποποιήσεων, οι οποίες κατάφεραν να εξελίσσουν και να βελτιώσουν τα παραγόμενα αποτελέσματα των RC.

Στο άρθρο (Schrauwen, 2007), οι *Schrauwen et al.* το 2007, κατατάσσουν τις εφαρμογές των RC σε αφηρημένες, όπως η αυτόνομη παραγωγή ημιτόνων και η κατηγοριοποίηση μοτίβων (temporal pattern classification), καθώς και σε πρακτικές (real world engineering applications), όπως στον τομέα της ρομποτικής (πχ. έλεγχος ρομποτικών κινήσεων, παρακολούθηση κίνησης αντικειμένου), της ψηφιακής επεξεργασίας σήματος (DSP) και των χαοτικών χρονοσειρών .

Στο άρθρο (Lukoševičius M. J., 2012) των *Lukosevicius et al.* το 2012, αναφέρονται μερικές από τις εφαρμογές ενός RC συστήματος, όπως η χρονική κατηγοριοποίηση μοτίβων, η χρονική παραγωγή μοτίβων (temporal pattern generation), η πρόβλεψη χρονοσειρών (time series prediction), η δρομολόγηση (routing), η απομνημόνευση (memorizing) των input ακολουθιών και ο έλεγχος μη γραμμικών συστημάτων (controlling nonlinear systems).

Το άρθρο (Tanaka, 2019) οι *Tanaka et al.* το 2019, αναφέρεται σε μεθόδους, φυσικά RC και σε σύγχρονες εφαρμογές, που έχουν επιτυχή εφαρμογή στην επίλυση πολλών πρακτικών προβλημάτων και περιέχουν πραγματικά δεδομένα σαν είσοδο.

Table 1
Examples of subjects in RC applications.

Category	Examples
Biomedical	EEG, fMRI, ECG, EMG, heart rates, biomarkers, BMI, eye movement, mammogram, lung images.
Visual	Images, videos.
Audio	Speech, sounds, music, bird calls.
Machinery	Vehicles, robots, sensors, motors, compressors, controllers, actuators.
Engineering	Power plants, power lines, renewable energy, engines, fuel cells, batteries, gas flows, diesel oil, coal mines, hydraulic excavators, steam generators, roller mills, footbridges, air conditioners.
Communication	Radio waves, telephone calls, Internet traffic.
Environmental	Wind power and speed, ozone concentration, PM2.5, wastewater, rainfall, seismicity.
Security	Cryptography.
Financial	Stock price, stock index, exchange rate.
Social	Language, grammar, syntax, smart phone.

Table 2
Applications and related benchmark tasks of RC.

Applications	Benchmark tasks
Pattern classification	Spoken digit recognition (Verstraeten, Schrauwen, Stroobandt and Van Campenhout, 2005) Waveform classification (Paquot et al., 2012) Human action recognition (Soh & Demiris, 2012) Handwritten digit image recognition (Jalalvand, Van Wallendael, & Van de Walle, 2015)
Time series forecasting	Chaotic time series prediction (Jaeger, 2001) NARMA time series prediction (Jaeger, 2003)
Pattern generation	Sine-wave generation (Jaeger, 2002b) Limit cycle generation (Hauser, Ijspeert, Fuchslin, & Maass, 2012)
Adaptive filtering and control	Channel equalization (Jaeger & Haas, 2004)
System approximation	Temporal XOR task (Bertschinger & Natschläger, 2004) Temporal parity task (Bertschinger & Natschläger, 2004)
Short-term memory	Memory capacity (Jaeger, 2002a)

Πίνακας 1 α) Κατηγορίες εφαρμογών RC και β) εφαρμογές RC

Στον πίνακα 1α παρουσιάζονται τομείς (όπως Βιοιατρικός, Μηχανολογικός, Τεχνικός, Περιβαλλοντικός, Οικονομικός κ.α) στους οποίους γίνεται επίλυση προβλημάτων με RC . Στον πίνακα 1β αναφέρονται MM εφαρμογές RC όπως pattern classification, πρόβλεψη χρονοσειρών, δημιουργία patterns κ.α. Ανάλογα με το πρόβλημα προς επίλυση, τα RC ορίζουν την πληροφορία που πρέπει να υπάρχει στα input και στα output.

Για παράδειγμα, στα **pattern classification** προβλήματα, τα input είναι χρονοσειρές και τα output είναι διακριτές τιμές (ταμπέλες) που αντιπροσωπεύουν την κλάση του μοτίβου. Πιο συγκεκριμένα, σε ένα πρόβλημα προφορικής αναγνώρισης ψηφίου (spoken digit recognition), το input είναι το ηχητικό σήμα που αντιστοιχεί σε μία από τις δέκα διαφορετικές εκφωνήσεις των ψηφίων από το μηδέν έως το εννέα και το output είναι η κατηγοριοποίηση σε διακριτή τιμή ενός από τα δέκα ψηφία. Στόχος αυτής της διεργασίας είναι η αναγνώριση του ψηφίου από τον προφορικό λόγο και η σωστή απεικόνισή του σαν νούμερο. Προκειμένου να αφαιρεθούν περιττές πληροφορίες και ο θόρυβος, το αρχικό ηχητικό σήμα περνάει μία μετατροπή πριν εισαχθεί στο reservoir.

Το RC είναι κατάλληλο για **temporal pattern recognition**, όμως μπορεί να εφαρμοστεί στην αναγνώριση εικόνας μετατρέποντας μια εικόνα σε μια ακολουθία τιμών pixel.

Ένα παράδειγμα επιτυχούς εφαρμογής pattern recognition (**image recognition and video processing**) αναφέρεται στο άρθρο (Jalalvand, 2015, June), όπου περιγράφεται εφαρμογή RC αναγνώρισης χειρόγραφων ψηφίων από την βάση δεδομένων MNIST, όπως επίσης και αναγνώριση (real time detection) της κατάστασης μιας εξωτερικής πόρτας οικίας χρησιμοποιώντας μια βάση δεδομένων από κινούμενες εικόνες που βιντεοσκοπήθηκαν από μια χαμηλής ανάλυσης κάμερα επιθεώρησης χώρου (surveillance camera). Όσον αφορά τις εφαρμογές στον ρομποτικό τομέα, αρχικά χρησιμοποιήθηκε η RC τεχνική με βάση την μέθοδο του LSM (Schrauwen, 2007), ενώ στην πορεία αποδείχθηκε ότι και η μέθοδος ESN (ή παραλλαγές της) μπορεί να εφαρμοστεί με επιτυχία (Lukoševičius M. J., 2012), (Polydoros, 2015). Στον τομέα του **Forecasting** τα RC σημειώνουν μεγάλη επιτυχία λόγω της χρήσης χρονικών δεδομένων εισόδου. Η εργασία (Ferreira, 2008, September) των *Ferreira et al*, αναφέρεται στην πρόβλεψη της ταχύτητας του αέρα, η οποία παρουσιάζει συχνές μεταβολές και αποτελεί βασική πληροφορία για τον σχεδιασμό της λειτουργίας αιολικών πάρκων παραγωγής ηλεκτρισμού. Τα αποτελέσματα έδειξαν ότι, το RC είναι πιο αποδοτικό σε σχέση με την προγενέστερη μέθοδο πρόβλεψης ταχύτητας του αέρα (MLP ή persistence method), που χρησιμοποιούσαν οι μετεωρολόγοι για την πρόβλεψη του καιρού. Στην εργασία των *Budhiraja et al* (Budhiraja, 2021), γίνεται εφαρμογή ενός RC σε θέματα Financial Forecasting. Το RC ακολουθεί τις αρχές του ESN σε συνδυασμό με τα δυνατά σημεία μίας χρονοκαθυστερημένης ανάδρασης (time-delayed feedback), με σκοπό την μοντελοποίηση ενός πολύπλοκου οικονομικού συστήματος.

Στο άρθρο (Sun, 2020), οι *Sun et al.* το 2020, αναφέρουν ότι οι προσεγγίσεις με ESN είναι αποδοτικές σε όλους τους προαναφερθέντες τομείς, και δίνουν πληθώρα εφαρμογών. Επιπλέον, γίνεται αναφορά εφαρμογών στον τομέα Ambient-assisted-living (AAL) που σκοπό έχει την αναγνώριση της ανθρώπινης καθημερινότητας. Για παράδειγμα, παρακολουθώντας την κανονικότητα της δραστηριότητας των ανθρώπων, ενισχύουν τον βαθμό προσωποποίησης των smart home υπηρεσιών (IoT). Η υιοθέτηση της μεθοδολογίας του ESN με σκοπό την εύρεση σχετικών μοτίβων στην συμπεριφορά του ανθρώπου, είναι συνεχώς αυξανόμενη.

Αυτές είναι μόνο μερικές από τις εφαρμογές καθώς η δυνατότητα που έχει το RC να χειρίζεται ακολουθίες δεδομένων με ευκολία δίνει προοπτική για νέες εφαρμογές (Lukoševičius M. &, 2009). Επίσης, σύγχρονες μελέτες έχουν οδηγήσει στην ανάπτυξη εφαρμογών φυσικών RC, στηριζόμενα σε νευρομορφική τεχνολογία (Neuromorphic Computing (NC)), που στοχεύουν στην ανάπτυξη ενεργειακά αποδοτικού hardware μηχανικής εκμάθησης. Για τις τεχνικές NC θα υπάρξει εκτενέστερη αναφορά σε επόμενο κεφάλαιο.

2.4 Echo State Networks (ESN)

Ένα Reservoir Computing (RC) σύστημα αναπτύσσεται με βάση κάποιες αρχιτεκτονικές μεθόδους. Η μέθοδος Echo State Network (ESN) αποτελεί μία από τις δύο πρωτοπόρες τεχνικές για την δημιουργία ενός RC και σύμφωνα με το (Rodan, 2010) είναι η πιο απλή μορφή RC. Όπως αναφέρθηκε παραπάνω, σε ένα RNN το οποίο πληροί κάποιες ιδιότητες, αν εκπαιδευτεί μόνο το output layer (readout) και τα input με τα hidden επίπεδα παραμένουν ανεκπαίδευτα, τότε τα δύο τελευταία αποτελούν ένα δυναμικό reservoir. Λόγω των αναδρομικών συνδέσεων μεταξύ των κόμβων του reservoir δημιουργούνται καταστάσεις που αντηχούν (echo), δηλαδή θυμούνται παρελθοντικές καταστάσεις των input. Η αρχική και πιο δημοφιλής μέθοδος εκπαίδευσης κατά παρτίδες (offline batch), όπως αναφέρθηκε και

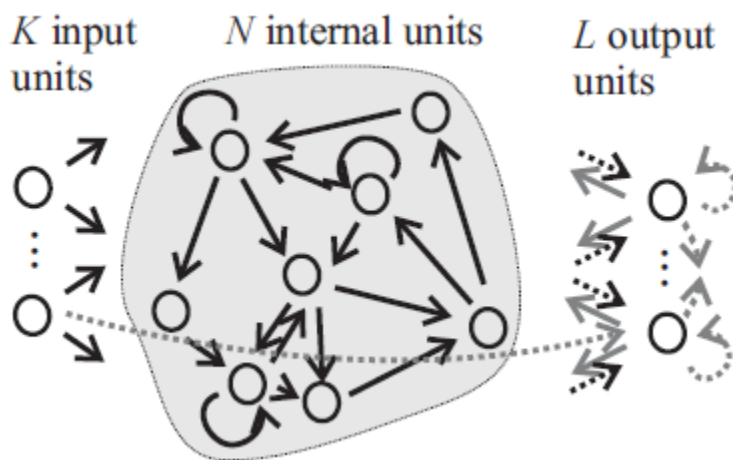
παραπάνω, για τον υπολογισμό των βαρών εξόδου είναι η linear regression (Lukoševičius M. &, 2009). Για ρυθμίσεις online εκπαίδευσης, συνιστάται ο υπολογιστικά “φθηνός” αλγόριθμος των ελαχίστων μέσων τετραγώνων (least mean squares).

2.4.1 Μαθηματικοποίηση του ESN

Σαν ESN ορίζεται η “αρχιτεκτονική μέθοδος” που χρησιμοποιεί ένα RC για τις συνδέσεις των νευρώνων μεταξύ τους. Πρώτος ο Jaeger, το 2001, (Jaeger H, 2001) ανέλυσε μαθηματικά το μοντέλο του ESN (Jaeger H. , 2002, March), βελτιώνοντας το στην συνέχεια, το 2010, στο άρθρο (Jaeger H, 2001) σχετικά με την ορολογία που προϋπήρχε. Παρακάτω περιγράφεται μαθηματικά η δημιουργία ενός ESN όπως ορίστηκε από τον Jaeger το 2010 (generic model).

Ένα reservoir είναι ένα RNN με K input units, N reservoir units και L output units. Ας υποθέσουμε ότι έχουμε ένα input σήμα $u(n)$, όπου $n = 1, \dots, T$ είναι διακριτή τιμή χρόνου και κάθε χρονική στιγμή παίρνει τιμές από ένα σύνολο εκπαίδευσης (training dataset) (Martinuzzi, 2020). Με βάση το (Lukoševičius M. , 2012), για να περάσει η πληροφορία $u(n) = (u_1(n), \dots, u_K(n))$ μέσα στο reservoir, απαιτούνται συνδέσεις με τους νευρώνες του reservoir. Τα βάρη των συνδέσεων εισάγονται στον πίνακα W^{in} . Κατά παρόμοιο τρόπο μέσα στο reservoir διακινείται πληροφορία $x(n) = (x_1(n) \dots x_N(n))$, ανάμεσα σε ένα μεγάλο αριθμό αραιά συνδεδεμένων νευρώνων. Τα βάρη των συνδέσεων ανάμεσα στους κόμβους του reservoir εισάγονται στον πίνακα W . Επίσης, στο output layer μεταφέρεται πληροφορία $y(n) = (y_1(n), \dots, y_L(n))$ από το reservoir προς την έξοδο και τα βάρη των συνδέσεων εισάγονται σε έναν πίνακα W^{out} . Επιπλέον, τα βάρη των εισόδων και του reservoir είναι σταθερά μετά την αρχικοποίηση, ενώ τα βάρη του output layer εκπαιδεύονται με την βοήθεια της τεχνικής γραμμικής παλινδρόμησης (linear regression).

Στην *εικόνα 18*, απεικονίζεται η βασική αρχιτεκτονική του reservoir ESN που θα χρησιμοποιήσουμε στην συνέχεια όπου τα διακεκομμένα βέλη είναι μη- απαιτούμενες συνδέσεις.



Εικόνα 18. Βασική Αρχιτεκτονική Reservoir συστημάτων (Jaeger H, 2001)]

Όσον αφορά τις συνδέσεις των κόμβων σύμφωνα με την *εικόνα 18* έχουμε:

- έναν πίνακα $N \times K$ για τις συνδέσεις input-internal unit, που περιλαμβάνει τα βάρη $W^{in} = (w_{ij}^{in}) \in \mathbb{R}^{N \times K}$
- έναν πίνακα $N \times N$ για τις συνδέσεις των κόμβων στο internal unit μεταξύ τους, που περιλαμβάνει τα βάρη $W = (w_{ij}) \in \mathbb{R}^{N \times N}$
- έναν πίνακα $L \times (K+N+L)$ για τις συνδέσεις των κόμβων προς το output unit, που περιλαμβάνει τα βάρη $W^{out} = (w_{ij}^{out}) \in \mathbb{R}^{L \times (K+N+L)}$ και
- έναν πίνακα $N \times L$ για τις συνδέσεις των κόμβων που από το output unit επιστρέφουν στο internal unit, που περιλαμβάνει τα βάρη $W^{back} = (w_{ij}^{back}) \in \mathbb{R}^{N \times L}$

Η παραπάνω ορολογία απεικονίζεται συγκεντρωμένη στους παρακάτω πίνακες.

Επίπεδα	Καταστάσεις στη χρονική στιγμή n
K input units	$u(n) = (u_1(n), \dots, u_K(n)),$
N reservoir units	$x(n) = (x_1(n) \dots x_N(n))$
L output units	$y(n) = (y_1(n), \dots, y_L(n)).$

Πίνακας 2. Ορολογία και καταστάσεις ενός reservoir

Συνδέσεις ανάμεσα	Πίνακες βαρών
Input (K units) και Internal layer (N units): ($N \times K$)	$W^{in} \in \mathbb{R}^{N \times K}$
Internal (N units) και Internal layer (N units): ($N \times N$)	$W \in \mathbb{R}^{N \times N}$
Output (L units) και όλα τα units : ($L \times (K+N+L)$)	$W^{out} \in \mathbb{R}^{L \times (K+N+L)}$
Internal (N units) και Output layer (L units): ($N \times L$)	$W^{back} \in \mathbb{R}^{N \times L}$

Πίνακας 3. Συνδεσμολογία reservoir και πίνακες βαρών

Είναι προφανές, από τον (πίνακα 3) και την (εικόνα 18), ότι επιτρέπονται συνδέσεις απευθείας από το input προς το output ($L \times (K+N+L)$). Κατά τον Jaeger, δεν απαιτείται όλοι οι κόμβοι στο reservoir να έχουν αναδρομικές σχέσεις στις συνδέσεις των βαρών W , όπως φαίνεται και από την (εικόνα 18). Σημαντική προϋπόθεση αποτελεί ότι οι τιμές των input, weights και activations να ανήκουν στο σύνολο των πραγματικών αριθμών.

Η ενημέρωση των τιμών που προκύπτουν από την εφαρμογή του activation function στα internal units προκύπτει από:

$$x(n+1) = f(W^{in} u(n+1) + Wx(n) + W^{back} y(n)), \quad (1)$$

όπου $f = (f_1, \dots, f_N)$ είναι οι activation functions (συνήθως sigmoid) των εξόδων που προκύπτουν από τις συνδέσεις των κόμβων στο εσωτερικό του reservoir με τους άλλους κόμβους.

Στην βιβλιογραφία αναλόγως το πρόβλημα που λύνεται κάθε φορά, οι τύποι του Jaeger για ένα ESN, διαφοροποιούνται όσον αφορά το χρονισμό τους (Schrauwen, 2007).

Στην εξίσωση (1) το W^{back} μηδενίζεται όταν δεν υπάρχουν αναδραστικές σχέσεις στην έξοδο. Ο υπολογισμός της τιμής εξόδου προκύπτει από την εξίσωση :

$$y(n+1) = f^{out}(W^{out}(u(n+1), x(n+1), y(n))) \quad (2)$$

όπου $f^{out} = (f_1^{out}, \dots, f_L^{out})$ είναι οι συναρτήσεις των κόμβων του output unit και $(u(n+1), x(n+1), y(n))$ είναι η συνένωση των input, Internal και προηγούμενων output activation διανυσμάτων.

Στην εξίσωση 2, σύμφωνα με την βιβλιογραφία (Rodan, 2010), το $y(n)$ αναφέρεται και ως $\hat{y}(n)$ και είναι το trained output. Η επιθυμητή τιμή εξόδου συναντιέται στην βιβλιογραφία είτε με την μορφή $y^{target}(n)$, είτε με την μορφή $y^{teacher}(n)$ (Lukoševičius M. , 2012).

Οι παραπάνω εξισώσεις, όπως ήδη αναφέρθηκε, χρησιμοποιούνται για την περίπτωση που οι τιμές του $u(n)$ είναι διακριτές. Σε περίπτωση όμως που το σήμα εισόδου είναι συνεχές (πχ ημιτονοειδής συνάρτηση) τότε οι (1) και (2) πρέπει να τροποποιηθούν κατάλληλα, για να προσαρμοστούν στην δυναμική του σήματος εισόδου. Προκειμένου να οριστεί μαθηματικά η παραπάνω ιδέα, δημιουργήθηκε η τροποποιημένη εξίσωση Leaky Integrator ESN όπως εμφανίζεται στο (Schaetti, 2016, August). Η μέθοδος leaky integration της κατάστασης των νευρώνων θεωρείται μια τυπική πρακτική στα ESN (Rodan, 2010).

Το leaking rate των νευρώνων είναι $\alpha \in (0,1]$ και η εξίσωση των καταστάσεων του reservoir διαμορφώνεται ως εξής (εξ 3) όταν εισαχθούν οι τιμές συνεχούς χρόνου t:

$$x_{t+1} = (1 - \alpha)x_t + f(W^{in}u_{t+1} + Wx_t + W^{back}y_t) \quad (3)$$

όπου f : sigmoid activation function $\tanh(\cdot)$.

Στα άρθρα (Lukoševičius M. , 2012), (Lukoševičius M. J., 2012), (Jaeger H, 2001), (Schaetti, 2016, August) αναφέρεται ότι η μέθοδος που χρησιμοποιείται για την δημιουργία ενός RC ESN ακολουθεί τα παρακάτω βασικά βήματα:

1. Παραγωγή ενός μεγάλου και τυχαία συνδεδεμένου reservoir (W_{in}, W, α),
2. Εκτέλεση του reservoir με την χρήση σήματος εισόδου $u(n)$ και συλλογή των αντίστοιχων καταστάσεων (states) ενεργοποίησης του reservoir $x(n)$,
3. Υπολογισμός των βαρών της γραμμικής εξόδου (W_{out}) από το reservoir με χρήση της τεχνικής της γραμμικής παλινδρόμησης (linear regression), για να επιτευχθεί ελαχιστοποίηση του σφάλματος εκπαίδευσης της εξόδου του ESN ($y_{target}(n)$),
4. Χρήση του εκπαιδευμένου πλέον δικτύου σε νέα δεδομένα εισόδου $u(n)$, προκειμένου να υπολογιστεί το $y(n)$ χρησιμοποιώντας τα εκπαιδευμένα πλέον βάρη εξόδου (W_{out}).

2.4.2 Βασικές παράμετροι ESN

Στα άρθρα (Rodan, 2010), (Lukoševičius M. , 2012) αναφέρονται οι βασικές παράμετροι (global parameters) που απαιτείται να ορισθούν για τον σχεδιασμό ενός ESN, οι οποίες αναλύονται παρακάτω.

- 1 το **μέγεθος του reservoir**, N reservoir units. Αναλόγως το πρόβλημα που πρέπει να επιλυθεί, επιλέγεται το κατάλληλο μέγεθος. Όσο πιο μεγάλος είναι ο χώρος του reservoir, τόσο ποιο αποδοτικό γίνεται το σύστημα και πιο εύκολο είναι να βρεθεί ένας γραμμικός συνδυασμός των σημάτων του, ώστε να προσεγγιστεί η επιθυμητή έξοδος. Πιο συγκεκριμένα, συστήνεται η τιμή του N να είναι αρκετά μεγάλη, τάξης μεγέθους ως 10^4 . Το μέγεθος του reservoir μπορεί να είναι τουλάχιστον ίσο με τον αριθμό των ανεξάρτητων τιμών που πρέπει το reservoir να θυμάται από το input, έτσι ώστε να υπάρξει επιτυχής επίλυση του προβλήματος. Ο μέγιστος αριθμός αποθηκευμένων τιμών (input) ονομάζεται memory capacity του ESN.

Για τις παρακάτω παραμέτρους συχνά απαιτούνται πολλαπλές δοκιμές για την εύρεση βέλτιστων τιμών.

- 2 ο αριθμός των **αραιών συνδέσεων (sparsity)** στο reservoir. Συστήνεται κάθε κόμβος να συνδέεται με έναν μικρό αριθμό άλλων κόμβων (πχ 5-12) σταθερά, ασχέτως με το μέγεθος του reservoir. Γενικά το sparsity σε ένα reservoir δεν επηρεάζει πολύ την απόδοση με αποτέλεσμα να μην δίνεται μεγάλη προτεραιότητα στην βελτιστοποίησή του, αλλά είναι χρήσιμο για την βελτίωση της ταχύτητας των υπολογισμών.
- 3 **κατανομή των μη-μηδενικών στοιχείων**. Ο πίνακας που περιέχει τα βάρη των συνδέσεων του reservoir δημιουργείται συνήθως αραιός, με μη μηδενικά στοιχεία που έχουν, είτε συμμετρική ομοιόμορφη διακριτή κατανομή δύο τιμών (bi-valued), είτε κανονική κατανομή με κέντρο γύρω από το μηδέν. Και οι δύο κατανομές δίνουν ουσιαστικά την ίδια απόδοση, που εξαρτάται από τις άλλες παραμέτρους. Η διακριτή κατανομή bi-valued τείνει να δίνει έναν ελαφρώς λιγότερο πλούσιο χώρο σήματος (υπάρχει μη μηδενική πιθανότητα πανομοιότυπων νευρώνων), αλλά μπορεί να διευκολύνει την ανάλυση του τι συμβαίνει στο reservoir. Ο πίνακας βαρών των συνδέσεων εισόδου-reservoir δημιουργείται συνήθως σύμφωνα με τον ίδιο τύπο κατανομής του πίνακα που περιέχει τα βάρη των συνδέσεων του reservoir, αλλά συνήθως πιο πυκνό.
- 4 η τιμή της **φασματικής ακτίνας (spectral radius) $\rho(W)$** του πίνακα των βεβαρυμμένων συνδέσεων του reservoir W , που είναι το μέγιστο ιδιοδιάνυσμα (eigenvalue) του πίνακα. Μετά από την δημιουργία ενός τυχαία και αραιά συνδεδεμένου πίνακα του reservoir, η φασματική ακτίνα $\rho(W)$ υπολογίζεται και το W διαιρείται με αυτήν. Αυτό κάνει δυνατή την δημιουργία ενός πίνακα με μια μονάδα φασματικής ακτίνας (unit spectral radius). Προκειμένου να μπορεί ένα πρόβλημα να προσεγγιστεί με ESN, το reservoir θα πρέπει να ικανοποιεί την echo state ιδιότητα (ESP) η οποία θα αναλυθεί παρακάτω. Η φασματική ακτίνα καθορίζει πόσο γρήγορα εξαφανίζεται η επίδραση μίας εισόδου σε ένα reservoir με τον χρόνο και πόσο σταθερές είναι οι ενεργοποιήσεις του reservoir (Verstraeten D. D., 2010, July). Συστήνεται $\rho(W) < 1$, ώστε να εξασφαλιστεί η ύπαρξη του ESP (στις περισσότερες περιπτώσεις), δηλαδή ότι υπάρχει echo της πληροφορίας που φέρουν τα input στο reservoir. Επίσης, το $\rho(W)$ θα πρέπει να ορίζεται μεγαλύτερο για προβλήματα ESN που απαιτούν για την εκτέλεσή τους μεγαλύτερη μνήμη των καταστάσεων εισόδου και μικρότερο, όταν οι τρέχουσες έξοδοι $y(n)$ εξαρτώνται περισσότερο από το πρόσφατο ιστορικό των καταστάσεων εισόδου (Lukoševičius M. &, 2009).
- 5 η τιμή **input scaling**. Η κλιμάκωση του πίνακα βαρών εισόδου W_{in} είναι μια άλλη βασική παράμετρος για βελτιστοποίηση σε ένα ESN. Για ομοιόμορφα κατανεμημένα W_{in} συνήθως ορίζουμε την τιμή input scaling "a" ως το εύρος του διαστήματος $[-a; a]$ από τις οποίες

λαμβάνονται δείγματα τιμών W_{in} . Για κανονικά κατανοημένα βάρη εισόδου μπορεί κανείς να λάβει την τυπική απόκλιση ως μέτρο κλίμακας. Στο άρθρο συστήνεται ότι πρέπει να γίνεται scaling σε όλο το W_{in} με ομοιόμορφη κατανομή, ώστε να υπάρχουν λίγες global παράμετροι στο ESN. Σύμφωνα με το άρθρο, το input scaling ρυθμίζει

- το πόσο οι καταστάσεις του reservoir είναι μη-γραμμικές, που αυξάνονται με το $\rho(W)$
 - τη σχετική επίδραση της τρέχουσας τιμής input στο reservoir $x(n)$, σε αντιδιαστολή με τις προηγούμενες τιμές του input.
- 6 την τιμή του **Leaking rate (α)** των κόμβων του reservoir. Σαν α θεωρείται η ταχύτητα με την οποία γίνεται η δυναμική ενημέρωση των καταστάσεων του reservoir σε διακριτή μορφή χρόνου. Το α παίρνει τιμές από το πεδίο $(0,1]$.

2.4.3 Εκπαίδευση και Σφάλμα Εκπαίδευσης

Ο σκοπός της εκπαίδευσης ενός μοντέλου ESN, όπως αναφέρεται στο (Lukoševičius M. , 2012), είναι η έξοδος $y(n)$, να προσεγγίσει όσο πιο πολύ γίνεται το $y^{target}(n)$ ελαχιστοποιώντας έτσι το σφάλμα μέτρησης $E(y, y^{target})$. Κατά την διαδικασία της εκπαίδευσης και αξιολόγησης το σφάλμα E υπολογίζεται για όλο το output layer με την κατάλληλη συνάρτηση σφάλματος, είτε την Root-Mean-Square Error (RMSE) (εξ.4), είτε με την Normalized Mean-Square-Error (NMSE) (εξ.5) (Liao, 2017). Το output layer ενός ESN μπορεί να εκπαιδευτεί είτε offline batch είτε online (Rodan, 2010).

$$RMSE = \sqrt{\frac{1}{T} \sum_{n=1}^T (y_i(n) - y_i^{target}(n))^2} = \sqrt{MSE} \quad (4)$$

$$NRMSE = \sqrt{\frac{\langle \|y(t) - y^{target}(n)\|^2 \rangle}{\langle \|y^{target}(n) - y^{target}(n)\|^2 \rangle}} = \sqrt{NMSE} \quad (5)$$

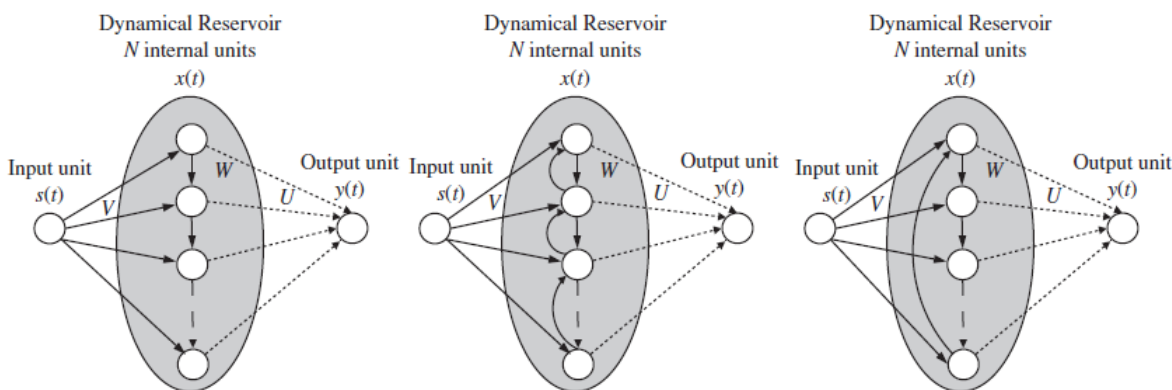
όπου για την εξίσωση 5 το $\langle \cdot \rangle$ δηλώνει τον Ευκλείδειο κανόνα και $\langle \cdot \rangle$ δηλώνει τον εμπειρικό μέσο.

2.4.4 Τοπολογίες των RC στα ESN

Σύμφωνα με το (Klipsis, 2016), κατά την προσέγγιση ενός συγκεκριμένου έργου (task) με RC, είναι κύριας σημασίας να καθοριστεί η διαμόρφωση των μονάδων επεξεργασίας των reservoir, η τοπολογία τους και οι συνδέσεις τους.

Οι Lukosevicius και Jaeger, στο άρθρο (Lukoševičius M. &, 2009), το 2009 αναφέρουν ότι έχουν γίνει προσπάθειες από διάφορους ερευνητές για την εύρεση τοπολογιών ESN διαφορετικών από το κλασικό μοντέλο που να παρουσιάζουν συγκρίσιμα αν όχι καλύτερα αποτελέσματα. Σχολιάζουν ότι, οι προσπάθειες αυτές δεν είναι μάταιες, δεδομένου ότι παρατηρείται σημαντική ποικιλία στην απόδοση του κλασικού ESN που είναι πιο εμφανές σε μικρά reservoir.

Στο άρθρο (Rodan, 2010) οι *Rodan et al.* το 2011 προτείνουν νέες τοπολογίες προκειμένου να απλοποιηθεί το κλασικό reservoir. Χρησιμοποιούν τόσο γραμμικά reservoir, με Identity activation function που βγάζει σαν έξοδο οτιδήποτε της δοθεί σαν input $f(x)=x$, όσο και μη γραμμικά reservoir, που αποτελούνται από νευρώνες με την ευρέως χρησιμοποιούμενη activation function $\tanh$. Τα γραμμικά reservoir προσομοιώνονται γρήγορα, αλλά συχνά οδηγούν σε κατώτερη απόδοση σε σύγκριση με τα μη γραμμικά (Verstraeten D. S., 2007), (Rodan, 2010) .



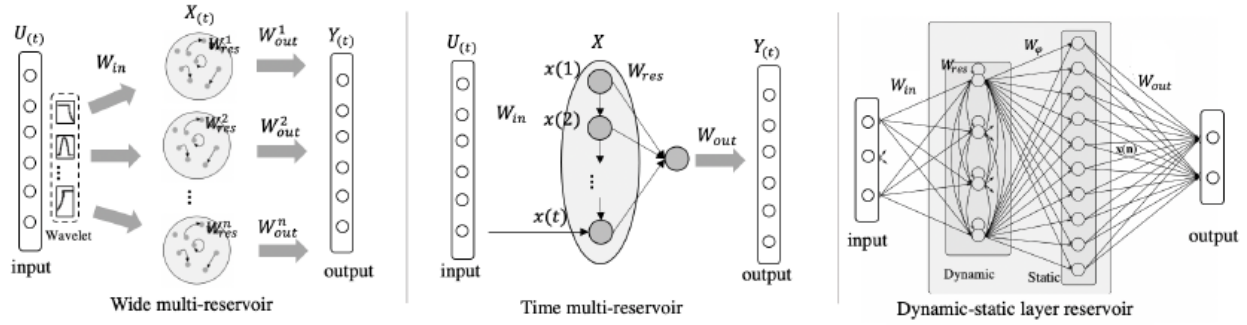
Εικόνα 19. Τοπολογίες ESN (Mahesh, 2020)

Εκτός από τον κλασικό πρότυπο reservoir ESN που εισήχθη στην εικόνα 18, εξετάζονται και τα ακόλουθα τρία πρότυπα reservoir (κατηγορίες μοντέλων) με σταθερές τοπολογίες (εικόνα 19). Στην εικόνα 19α εμφανίζεται η τοπολογία Delay line reservoir (DLR), η οποία αποτελείται από νευρώνες οργανωμένους σε μια γραμμή. Στην εικόνα 19β εμφανίζεται η τοπολογία DLRB με συνδέσεις ανάδρασης, η οποία έχει την ίδια δομή με το DLR και επίσης κάθε reservoir νευρώνας συνδέεται με τον προηγούμενο νευρώνα. Στην εικόνα 19γ εμφανίζεται η τοπολογία simple cycle reservoir (SCR), στην οποία οι μονάδες είναι οργανωμένες σε κύκλο. Οι προτεινόμενες τοπολογίες μελετήθηκαν θεωρητικά και αξιολογήθηκαν με διάφορες χρονοσειρές (πχ NARMA, Sunspot, IPIX Radar, Laser datasets κ.α). Μερικά από τα συμπεράσματα στα οποία οδηγήθηκαν (Mahesh, 2020), (Rodan, 2010), είναι ότι, η SCR τοπολογία παρουσιάζει αρκετά ικανοποιητική απόδοση συγκριτικά με το ESN και το memory capacity του SCR μπορεί να φτιαχτεί ώστε να είναι κοντά στην αποδεδειγμένα βέλτιστη τιμή .

Οι παραπάνω προταθείσες δομές έχουν αναπτυχθεί για την περίπτωση που υπάρχει μόνο ένα reservoir. Στην πράξη όμως μπορεί να υπάρχουν περισσότερα από ένα ESN reservoir για να επιτευχθεί μεγαλύτερη ακρίβεια. Νέα μοντέλα RC έχουν επινοηθεί για την βελτίωση της απόδοσης των πρωταρχικών RC μοντέλων, χρησιμοποιώντας αρχιτεκτονικές, όπως τα multiple reservoirs (εικόνα 20α) από τους (Akiyama, 2019, July), (Gallicchio C. M., 2017), (Malik, 2017) και τα evolving reservoirs (Growing ESN) από τους (Qiao, 2017), τα οποία αποτελούνται από sub-reservoirs με ανεξάρτητους πίνακες βαρών. Οι παραπάνω αρχιτεκτονικές συνδυάζουν το RC με άλλες μεθόδους εξαγωγής χαρακτηριστικών (feature extraction), ενσωματώνοντας νέους αλγόριθμους μάθησης και χρησιμοποιώντας μια ποικιλία στοιχείων reservoir. Σαν μέθοδοι εξαγωγής χαρακτηριστικών αναφέρονται τα μη εκπαιδευσιμα CNN από τους (Tong, 2018) και η εξαναγκασμένη εκπαίδευση (reinforcement learning) από τους (Chang, 2018); (Murakami, 2015, August), ενώ σαν αλγόριθμοι μάθησης αναφέρονται το FORCE learning (Sussillo, 2009) και (Pyle, 2018).

Στο άρθρο (Sun, 2020), οι *Sun et al.* το 2020, αναφέρονται στον σχεδιασμό shared reservoir modular (SRM ESN) των *Chen et al.* (Chen, 2010, July), για να διαιρεθεί πρώτα ο χώρος των νευρωνικών καταστάσεων

σε μικρότερες ενότητες με ανεξάρτητο βάρος εξόδου W_i^{out} η κάθε μία και στη συνέχεια να τοποθετηθούν τα δεδομένα που ανήκουν σε κάθε ενότητα στο ίδιο reservoir. Αυτό σημαίνει ότι το ESN έχει n reservoir και όχι ένα μόνο. Αναφέρονται επίσης και οι τοπολογίες Time Multi Reservoir των *Xia et al.* (Xia, 2010) το 2011 και ϕ -ESN των *Galliechio et al.* (Galliechio C. &, 2011) το 2011 όπως φαίνονται στην παρακάτω εικόνα (εικόνα 20) στο άρθρο (Sun, 2020) .



Εικόνα 20. Τοπολογίες που περιλαμβάνουν πολλαπλά reservoir (Sun, 2020)

Όπως φαίνεται από τα παραπάνω υπάρχει συνεχής εξέλιξη στα ESN και επιτυχής αξιοποίησή τους σε διάφορους τομείς. Παρόλα αυτά, σύμφωνα με τα άρθρα (Rodan, 2010), (Xu., 2020, March), (Millea, 2014) εξακολουθούν να υπάρχουν κάποια προβλήματα που εμποδίζουν την αποδοχή του σαν ευρέως διαδεδομένο εργαλείο, τα οποία αναφέρονται στην συνέχεια. Υπάρχουν ιδιότητες του reservoir οι οποίες δεν είναι απόλυτα κατανοητές. Ο προσδιορισμός των συνδέσεων μεταξύ reservoir και κόμβων εισόδου, απαιτεί πολλές δοκιμές και σε πολλές περιπτώσεις τύχη. Δεν έχουν επινοηθεί στρατηγικές για την επιλογή διαφορετικών reservoir για διαφορετικές εφαρμογές. Η επιβολή περιορισμού στην τιμή της φασματικής ακτίνας του reservoir είναι ένα αδύναμο εργαλείο για να ρυθμιστούν σωστά οι παράμετροι του reservoir. Η τυχαία δομή της συνδεσιμότητας μεταξύ των νευρώνων και του πίνακα βαρών του reservoir είναι απίθανο να είναι η βέλτιστη και δεν παρέχεται σαφής εικόνα για τη δυναμική οργάνωση του reservoir. Τα ESN συχνά είναι ασταθή και αυτό καθιστά δύσκολη την διαδικασία εύρεσης ενός ESN για ένα συγκεκριμένο dataset.

3 ΠΕΙΡΑΜΑΤΙΚΟ ΜΕΡΟΣ 1 : ΥΛΟΠΟΙΗΣΗ TRAINING ESN ΓΙΑ NARMA10, NARMA5 ΜΕ Tanh

Προκειμένου να τρέξω τις παρακάτω προσομοιώσεις χρησιμοποιήθηκε το πρόγραμμα “ESNToolbox_Original.m” του Jaeger, Version 1.0, Revision 2, Aug 17 2007. Αυτό το λογισμικό προορίζεται για ερευνητική χρήση από έμπειρους χρήστες του Matlab. Οι μέθοδοι που ακολουθούνται για την διαδικασία της εκπαίδευσης του ESN περιγράφονται με λεπτομέρεια στα άρθρα (Jaeger H. , 2002), (Jaeger H. , 2002). Το εκτελέσιμο αρχείο (*demoScript.m*) δείχνει πως μπορούν να φορτωθούν τα δεδομένα από ένα dataset, πως να δημιουργηθεί ένα δίκτυο, να εκπαιδευτεί ένα ESN, να ελεγχθεί με νέα δεδομένα και να παραχθούν διαγράμματα για τις τιμές εξόδου του δικτύου.

Για την παραγωγή των δεδομένων στην συγκεκριμένη περίπτωση χρησιμοποιήθηκε η function **[inputSequence,outputSequence] = generate_NARMA_sequence(sequenceLength, memoryLength)**, η οποία παράγει μια input και output χρονοσειρά δεδομένων. Η παραγωγή της input χρονοσειράς $u(t)$ του ESN δέχεται δεδομένα από μια uniform κατανομή [0,0.5]. Η output target χρονοσειρά παράγεται με το μοντέλο NARMA (Nonlinear Auto Regressive Moving Average) όπως φαίνεται στην *εξίσωση 6* (Sakemi, 2020), (Antonik, 2017):

$$y^{target}(t) = 0.3y^{target}(t-1) + 0.05y(t-1) \sum_{i=1}^m y^{target}(t-i) + 1.5u(t-m+1)u(t) + 0.1 \quad (6)$$

όπου m δηλώνει τον αριθμό της NARMA.

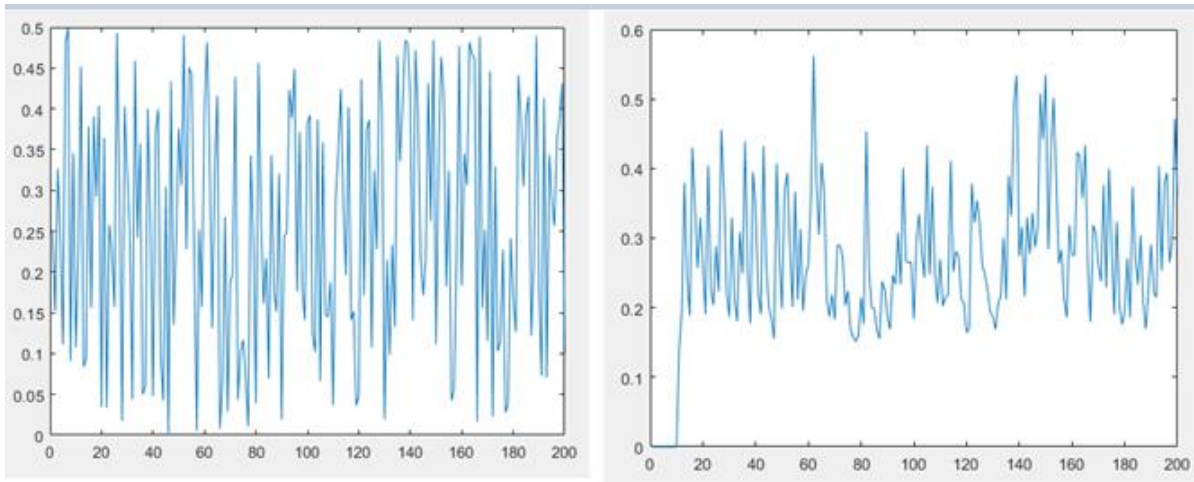
Στην εργασία χρησιμοποιείται το μοντέλο της NARMA 5 και NARMA10 όπου $m=5$ και $m=10$ αντίστοιχα. Ο βαθμός της NARMA ορίζει και τον βαθμό μη γραμμικότητας του συστήματος (Suzuki, 2022) καθώς και την μνήμη προηγούμενων καταστάσεων. Ένα σύστημα που εκτελείται με NARMA10 μπορεί να συγκρατήσει περισσότερες προηγούμενες καταστάσεις σε σχέση με ένα αντίστοιχο σύστημα που εκτελείται με NARMA5 και άρα το πρώτο σύστημα μπορεί να εκπαιδευτεί με μεγαλύτερη ακρίβεια καθώς κρατάει μεγαλύτερη “ιστορία” καταστάσεων.

Ο στόχος είναι η πρόβλεψη των τιμών εξόδου $y(t)$ από παρούσες τιμές εισόδου $u(t)$ και παρελθοντικές τιμές των $u(t)$, $y(t)$.

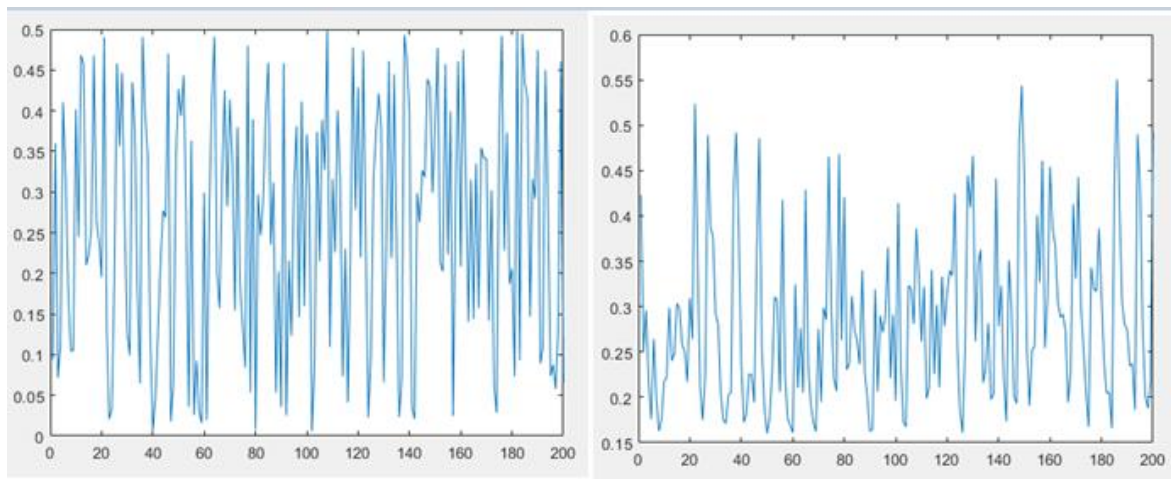
Το πρόγραμμα δημιουργεί μια χρονοσειρά input με `sequenceLength = 10000` τιμές για κάθε ένα μοντέλο NARMA. Στην συνέχεια η ακολουθία αυτή χωρίζεται σε ποσοστό 50% (`train_fraction=50%`) και έτσι τα `train dataset` όπως και τα `test dataset` αποκτούν από 5000 τιμές. Η **εκπαίδευση** (train) γίνεται με τις **πρώτες 5000 τιμές input** από τις 10000. Οι **επόμενες 5000 τιμές** του input χρησιμοποιούνται κατά την **test φάση** όπου ελέγχεται αν η εκπαίδευση (teacher και predicted του output) δίνει Normalized Root Mean Square Error (NRMSE) ιδανικά πολύ κοντά στο μηδέν.

Από τις output (*teacher-target*) 10000 τιμές που δημιουργήθηκαν με βάση την εξίσωση NARMA, οι πρώτες 5000 χρησιμοποιούνται σαν teacher τιμές στο train και οι δεύτερες 5000 τιμές σαν teacher τιμές για το test. Με βάση το πρόγραμμα οι πρώτες 100 τιμές στο output train και test δεν λαμβάνονται υπόψιν.

Τα παρακάτω διαγράμματα (*εικόνα 21, 22*) έχουν τις πρώτες 200 τιμές από την χρονοσειρά train και test input / output .



Εικόνα 21 α) τα πρώτα 200 στοιχεία του input [1-5000] για εκπαίδευση του ESN, β) Οι πρώτες 200 τιμές του teacher output [1-5000] για εκπαίδευση του ESN



Εικόνα 22 α) τα πρώτα 200 στοιχεία του input [5001-5200] για testing του ESN, β) Οι πρώτες 200 τιμές του teacher output [5001-5200] για testing του ESN

Ακολούθησε η παραγωγή του δικτύου ESN με χρήση της function ***generate_esn(nInputUnits, nInternalUnits, nOutputUnits, arguments)*** θέτοντας διάφορες τιμές για τα internal nodes και ορίζοντας Input nodes = 2, output nodes = 1. Επίσης ορίστηκαν και μερικές παράμετροι του ESN όπως το inputScaling = [0.1, 0.1], connectivity = 0.025, το outputActivationFunction = 'identity' και το methodWeightCompute = 'pseudoinverse'.

Κατά την διάρκεια του training και του testing προκειμένου να αποφύγουμε τις επιπτώσεις των αρχικών συνθηκών στο reservoir έγινε αφαίρεση των πρώτων 100 τιμών (nForgetPoints = 100). Αν η τιμή της παραμέτρου οριστεί μικρή, τότε η έξοδος που προκύπτει κάνει overfitting, δηλαδή περιέχει και τιμές που δεν είναι αξιόπιστες για να εκπαιδευτεί το σύστημα. (Mayer, 2015). Τέλος, έγινε η παραγωγή διαγραμμάτων για input και output ακολουθίες τιμών (train teacher με train predicted και test teacher με test predicted).

Στο στάδιο του testing έγινε ο υπολογισμός του σφάλματος με την μέθοδο NRMSE σύμφωνα με την εξίσωση (4) που αναφέρθηκε παραπάνω. Για κάθε 10 δοκιμές train και test υπολογίζεται το μέσο NRMSE και η τυπική απόκλιση. Οι παραπάνω υπολογισμοί έγιναν ξεχωριστά για 6 τιμές internal units

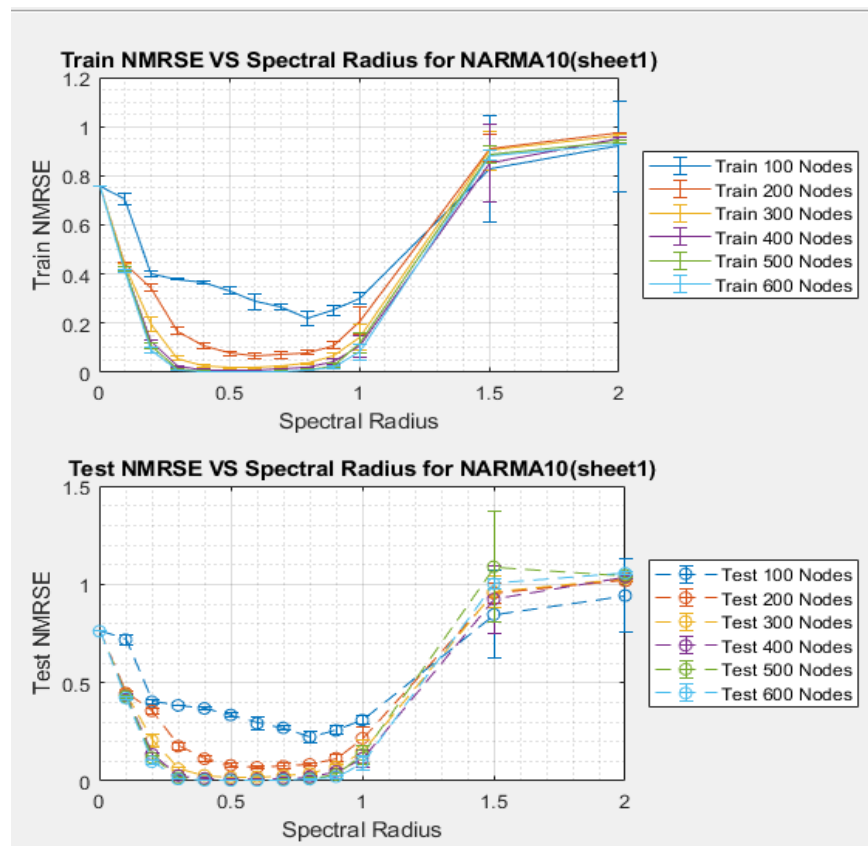
(100,200,300,400,500,600) και 13 τιμές $\rho(W)$ (0.001, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.5, 2.0) με NARMA10 και NARMA5 για όλους τους δυνατούς συνδυασμούς μεταξύ τους. Τα αποτελέσματα αποθηκεύτηκαν σε αρχεία excel. Στην συνέχεια δημιουργήθηκαν διαγράμματα με την βοήθεια του matlab που σχολιάζονται στην πορεία. Όπως αναφέρθηκε παραπάνω, έγινε διερεύνηση και για τιμές $\rho(W) > 1$ δεδομένου ότι ο Lukucevicius στο άρθρο (Lukoševičius M. , 2012), ενώ προτείνει ότι για να υπάρχει η ESP ιδιότητα πρέπει να τίθεται η $\rho(W) < 1$, δεν αποκλείει την περίπτωση να υπάρχει βέλτιστη $\rho(W) > 1$. Ο αριθμός της $\rho(W)$ θα πρέπει να τίθεται μεγάλος σε προβλήματα που το output $y(t)$ εξαρτάται από την ιστορία των input $u(t)$ σε βάθος χρόνου και αντιστρόφως.

3.1 NARMA10 για Activation Function Tanh

Στα παρακάτω διαγράμματα ελέγχεται η περίπτωση που το reservoir εκπαιδεύεται με την βοήθεια της tanh activation function για ένα συγκεκριμένο I/O sequence. Η παραγωγή του I/O sequence έγινε με την χρήση του μοντέλου της NARMA10.

3.1.1 Διάγραμμα NRMSE με Φασματική Ακτίνα

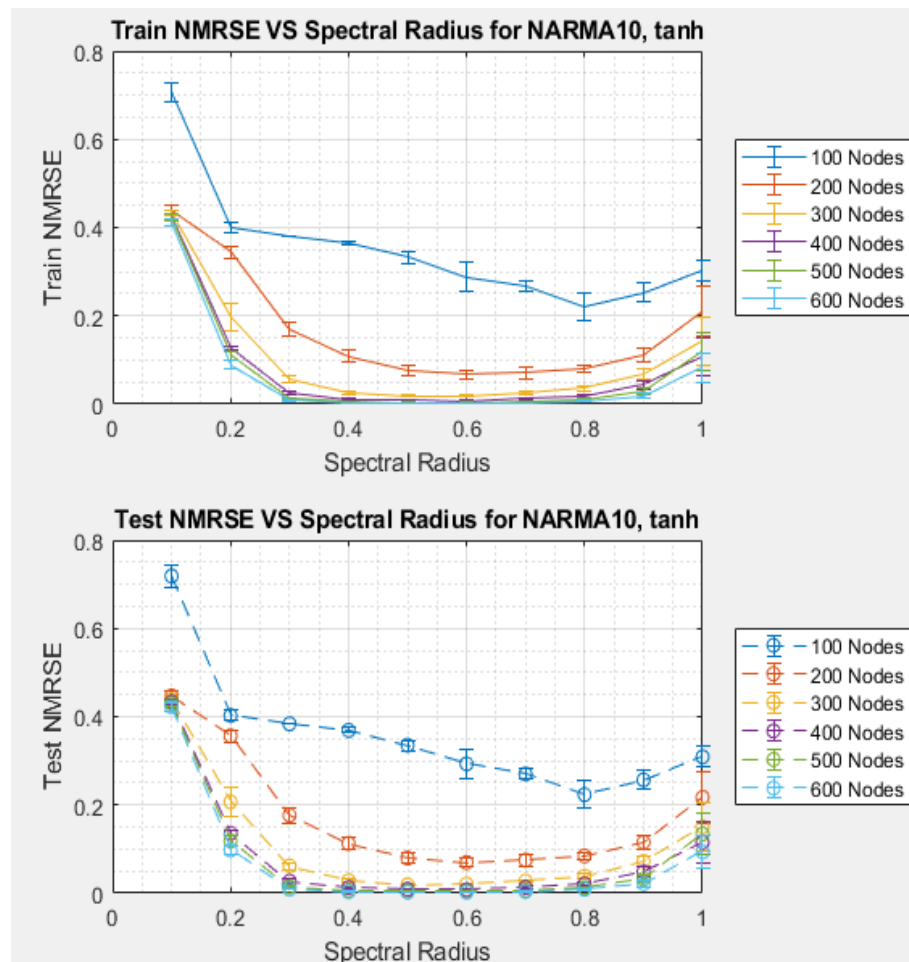
Υλοποιήθηκε η συνάρτηση **plotNARMA10_Err_SpectRad.m** με βάση την οποία δημιουργήθηκαν τα παρακάτω διαγράμματα στα οποία εμφανίζεται το σφάλμα train και test NRMSE για τις προαναφερθείσες τιμές $\rho(W)$. Η κάθε καμπύλη δείχνει την μεταβολή του σφάλματος σε σχέση με τα $\rho(W)$ για τις προαναφερθείσες τιμές κόμβων. Για την επαλήθευση και κατανόηση των διαγραμμάτων έγινε αντιπαραβολή με τα αποτελέσματα του αντίστοιχου αρχείου excel (*activationTanh.xlsx (sheet1)*).



Εικόνα 23. Train and Test NRMSE NARMA10 και για τιμές $\rho(W) > 1$ και activation function tanh

Σύμφωνα με την (εικόνα 23), για κάθε αριθμό κόμβων υπάρχει ένα $\rho(W)$ με ένα minimum NRMSE. Διαπιστώθηκε, ότι για $\rho(W) = (0.3 - 0.8)$ και 400-600 κόμβους (internal units) παρατηρείται πολύ χαμηλό σφάλμα ($NRMSE < 0.1$). Η βέλτιστη λύση υπάρχει μέσα σε αυτό το εύρος τιμών και πιο συγκεκριμένα για 600 κόμβους reservoir και $\rho(W) = 0.5$ το $testNRMSE = 0.0022$, όπως φαίνεται και στον συγκεντρωτικό πίνακα 6. Για τιμές $\rho(W) = \{0.001, 0.1, 1.5, 2.0\}$ και για όλες τις περιπτώσεις κόμβων που εξετάζονται, οι τιμές του σφάλματος NRMSE κρίνονται μεγάλες ($NRMSE > 0.4$).

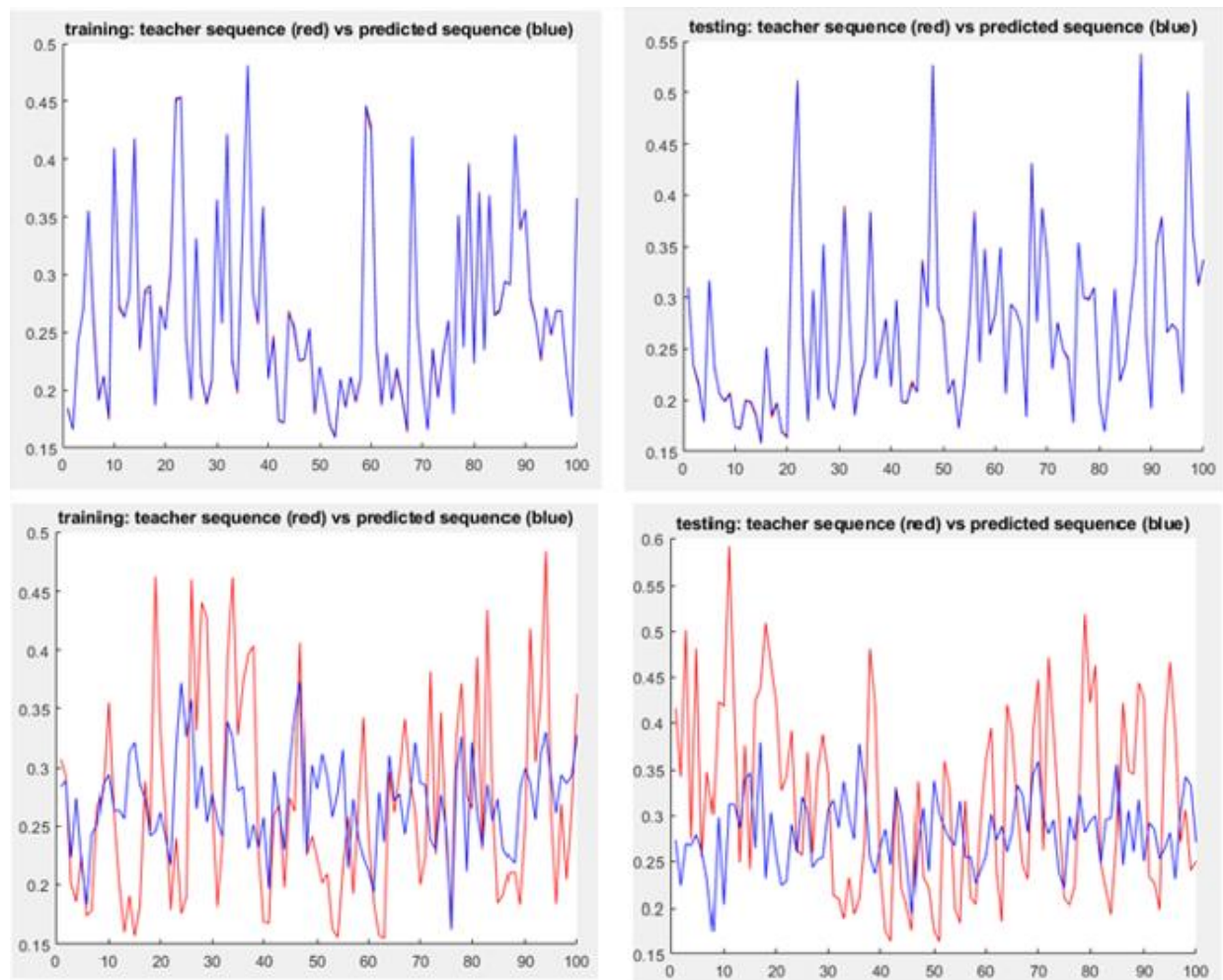
Για την ευκρινέστερη απεικόνιση των τιμών NRMSE στην περιοχή $\rho(W) = [0,1]$ ανασχεδιάστηκαν τα διαγράμματα της (εικόνας 23) και παρουσιάζονται στην (εικόνα 24) (η ασυνέχεια στο διάγραμμα της εικόνας 24, οφείλεται στο ότι το NRMSE γίνεται πολύ μεγάλο σε περιοχές με $\rho(W) < 0.1$).



Εικόνα 24. Train and Test NRMSE NARMA10 για τιμές που παρατηρείται σφάλμα NRMSE γύρω από τις τιμές της μονάδας

Σαν λογικό συμπέρασμα, με βάση το matlab, προκύπτει ότι όταν έχουμε λίγους κόμβους (πχ 100) το σφάλμα είναι μεγάλο για όλα τα $\rho(W)$, ενώ στην περίπτωση που ο αριθμός των κόμβων είναι μεγάλος (πχ 600), το σφάλμα μικραίνει και ο υπολογιστικός χρόνος μεγαλώνει. Ανάλογα με το πρόβλημα προς επίλυση και την activation function που χρησιμοποιείται κάθε φορά, αυτό το trade off διαφοροποιείται. Έτσι, στην συγκεκριμένη περίπτωση για tanh οι 600 κόμβοι δεν εξυπηρετούν λόγω του αυξημένου υπολογιστικού χρόνου, αλλά με μία διαφορετική activation function, πιθανόν να παράγεται η έξοδος ταχύτατα και με μικρό σφάλμα. Η περίπτωση διαφορετικών activation function θα εξεταστεί στην συνέχεια.

Με βάση τα παραπάνω συμπεράσματα και τα κάτωθι διαγράμματα (εικόνας 25 α,β) προκύπτει ότι με $\rho(W) = 0.5$ και αριθμό κόμβων 300, για την δεδομένη χρονοσειρά τα target και predicted output (train και test) σχεδόν ταυτίζονται. Αντίθετα, στα αποτελέσματα της (εικόνας 25γ,δ) είναι εμφανές ότι δεν γίνεται ταύτιση των καμπυλών. Υπάρχει και στις δύο περιπτώσεις μεγάλο σφάλμα NRMSE και τα target και predicted output έχουν απόκλιση μεταξύ τους. Πιο συγκεκριμένα, τα διαγράμματα της εικόνας 25γ,δ που σχηματίστηκαν με $\rho(W) = 1.5$ και αριθμό κόμβων 500, παρουσιάζουν NRMSE train = 0.8975 και NRMSE test = 1.019 με εμφανή απόκλιση μεταξύ των τιμών target (teacher) και predicted output. Ελέγχθηκε επίσης η περίπτωση διαφορετικού ποσοστού εκπαίδευσης (train_fraction = 90%) της ακολουθίας input, με αριθμό κόμβων 500 και $\rho(W) = 1.5$ που έδωσε παρόμοια αποτελέσματα.



Εικόνα 25. (α,β) Διαγράμματα για Reservoir Nodes =300 και Spectral Radius=0.5 ταυτίζονται. (γ,δ) Διαγράμματα για Reservoir Nodes =500 και Spectral Radius=1.5 παρουσιάζουν απόκλιση

Ένας εναλλακτικός τρόπος παρουσίασης των παραπάνω αποτελεσμάτων που υλοποιήθηκε με σχετική συνάρτηση `plotNARMA10_Err_ResNodes.m`, είναι στον άξονα των x να υπάρχουν οι κόμβοι του reservoir και η κάθε καμπύλη να αντιστοιχεί σε διαφορετικό $\rho(W)$. Αυξάνοντας τον αριθμό των κόμβων, για κάθε μία καμπύλη $\rho(W)$ το λάθος μειώνεται.

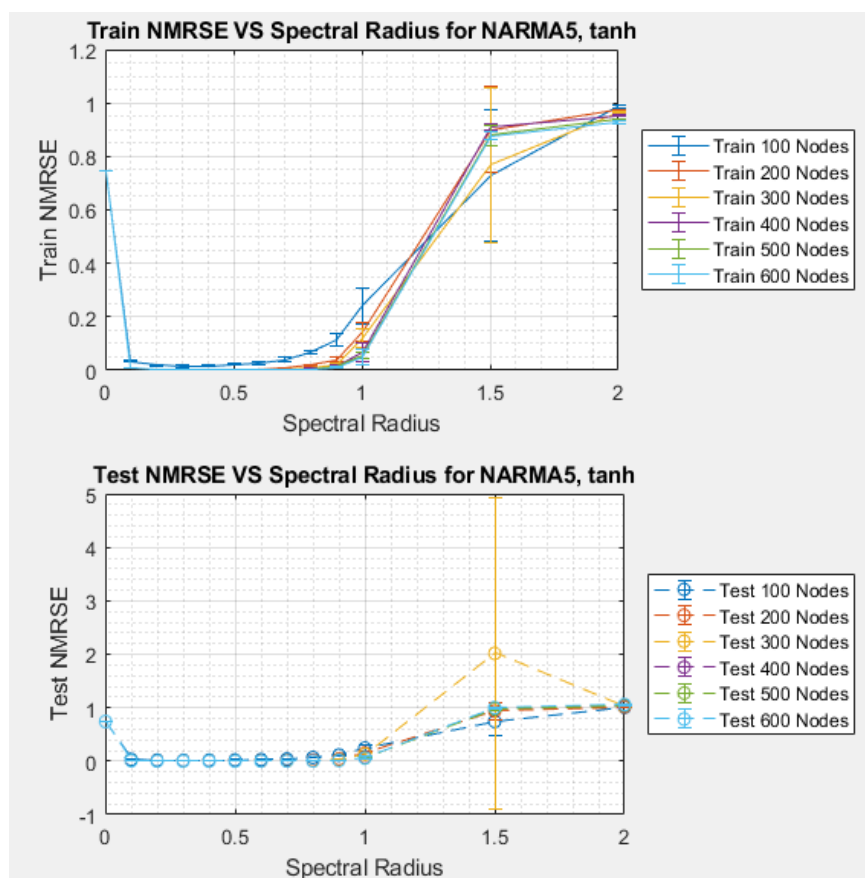
3.2 NARMA5 για Activation Function Tanh

Στα παρακάτω διαγράμματα ελέγχεται η περίπτωση που το reservoir εκπαιδεύεται με την βοήθεια της tanh activation function για ένα συγκεκριμένο I/O sequence. Η παραγωγή του I/O sequence έγινε με την χρήση του μοντέλου της NARMA5.

3.2.1 Διάγραμμα NRMSE με Φασματική Ακτίνα

Όπως στην περίπτωση του NARMA10 έτσι και στην NARMA5 τα διαγράμματα που δημιουργήθηκαν με την βοήθεια της συνάρτησης **plotNARMA5_Err_SpectRad.m**, απεικονίζουν το σφάλμα που προκύπτει κατά την φάση του train και test, για διαφορετικά spectral radius (εικόνα 26,27). Όπως και στις προηγούμενες περιπτώσεις, οι τιμές του διαγράμματος επαληθεύτηκαν με τα αποτελέσματα του αρχείου excel (*activationTanh.xlsx (sheet2)*). Η κάθε καμπύλη δείχνει την μεταβολή του σφάλματος σε σχέση με τα $\rho(W)$ για τις προαναφερθείσες τιμές κόμβων.

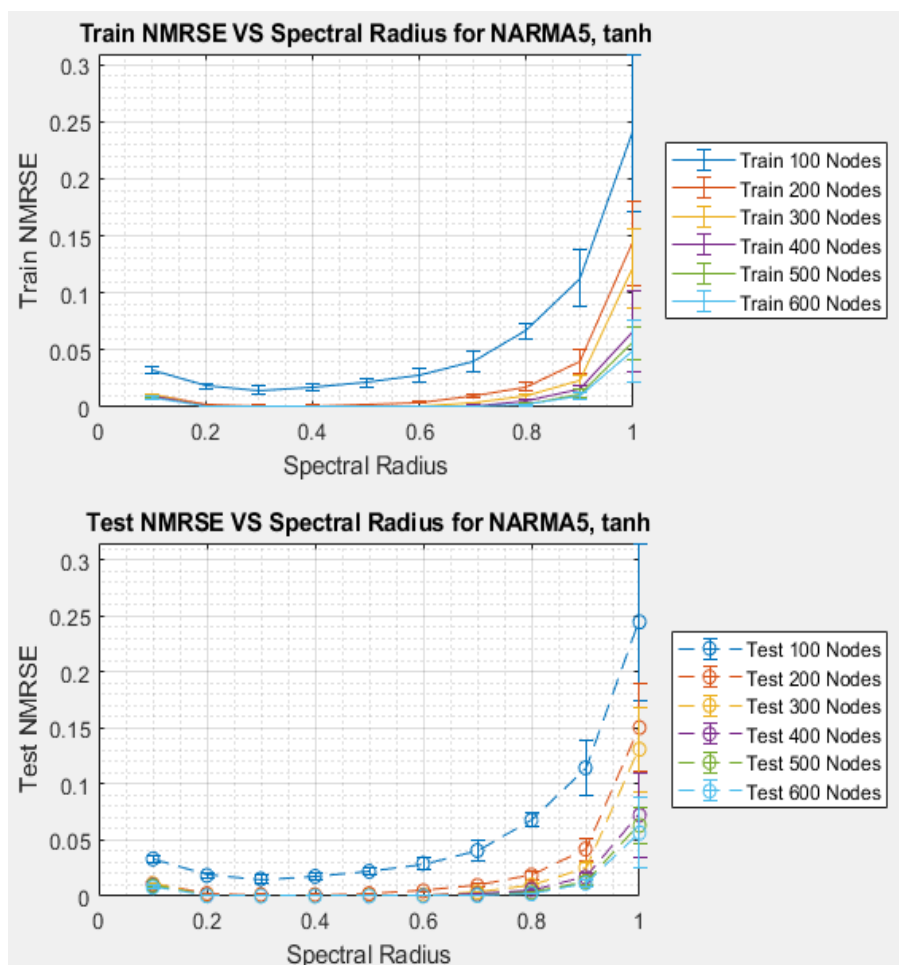
Τα συμπεράσματα που προκύπτουν είναι τα ίδια όσον αφορά τον υπολογιστικό χρόνο και τους κόμβους (πολλοί κόμβοι $\rightarrow$ μεγάλος χρόνος). Το σφάλμα NRMSE γίνεται πολύ μεγάλο για όλους τους κόμβους σε περιοχές με $\rho(W) \leq 0.001$ και $\rho(W) > 1.1$, ενώ για τις υπόλοιπες τιμές $\rho(W)$ η τιμή του σφάλματος NRMSE κυμαίνεται μεταξύ των τιμών (0 , 0.2).



Εικόνα 26. Train and Test NRMSE NARMA5 και για τιμές $\rho(W) > 1$

Για την ευκρινέστερη απεικόνιση των τιμών NRMSE στην περιοχή $\rho(W) = (0.1, 1)$ ανασχεδιάστηκαν τα διαγράμματα της (εικόνας 26) και παρουσιάζονται στην (εικόνα 27).

Διαπιστώθηκε, ότι για $\rho(W) = (0.1 - 0.8)$ παρατηρείται πολύ χαμηλό σφάλμα ($NRMSE < 0.01$) και μέσα σε αυτό το εύρος τιμών υπάρχει και η βέλτιστη λύση για NARMA5 και tanh. Από άποψη υπολογιστικού χρόνου η καλύτερη λύση βρίσκεται για 400 κόμβους και $\rho(W)=0.3$ όπου το $testNRMSE=0.0001$, αλλά υπάρχουν και άλλοι συνδυασμοί κόμβων - $\rho(W)$ που δίνουν το ίδιο $testNRMSE$ (πχ. η περίπτωση με 500 ή 600 κόμβους και $\rho(W) = \{0.3, 0.4, 0.5\}$).



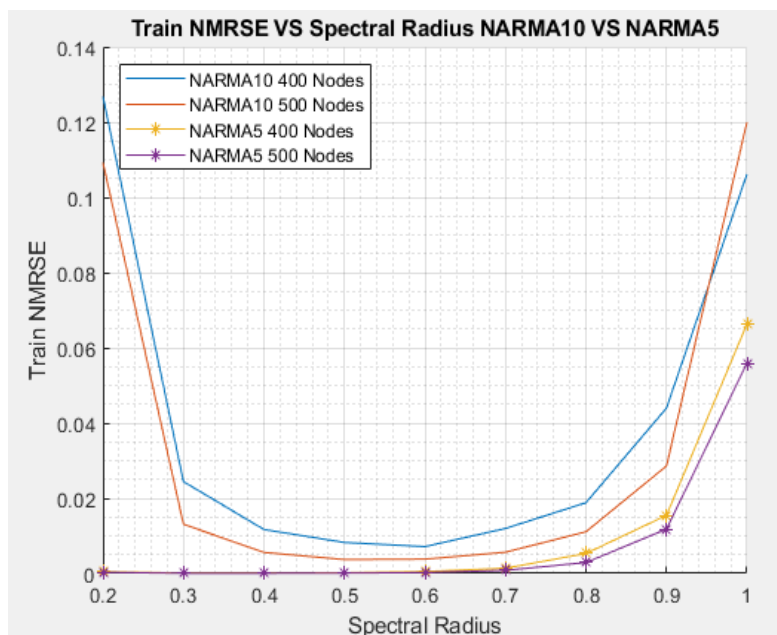
Εικόνα 27. Train and Test NRMSE NARMA5 για τιμές που παρατηρείται μικρό σφάλμα NRMSE ($NRMSE < 0.3$)

Ένας εναλλακτικός τρόπος παρουσίασης των παραπάνω αποτελεσμάτων που υλοποιήθηκε με σχετική συνάρτηση `plotNARMA5_Err_ResNodes.m`, είναι στον άξονα των x να υπάρχουν οι κόμβοι του reservoir και η κάθε καμπύλη να αντιστοιχεί σε διαφορετικά $\rho(W)$. Αυξάνοντας τον αριθμό των κόμβων, για κάθε μία καμπύλη $\rho(W)$ το σφάλμα μειώνεται.

3.3 Σύγκριση NARMA10 με NARMA5 και activation function Tanh

Στο άρθρο (Goudarzi, 2014) γίνεται σύγκριση του μοντέλου NARMA 10 χρονοσειράς και NARMA20 χρονοσειράς. Αναφέρεται πως το μοντέλο NARMA20 απαιτεί διπλάσιο υπολογιστικό χρόνο και μνήμη σε σχέση με το NARMA10. Διαπιστώθηκε ότι ισχύει το ίδιο και για την περίπτωση που εξετάστηκε το μοντέλο της NARMA5 με τη NARMA10 (Manneschi, 2021).

Από τα παραπάνω διαγράμματα γίνεται εμφανές ότι για NARMA10 το σφάλμα train και test NRMSE είναι μεγαλύτερο από ότι για NARMA5. Έτσι, παρατηρείται ότι στην περίπτωση του NARMA10 (εικόνα 24), οι τιμές σφάλματος NRMSE κυμαίνονται στο διάστημα $[0,0.5]$, ενώ στην περίπτωση NARMA5 (εικόνα 27) οι τιμές είναι μικρότερες ($NRMSE = [0,0.03]$). Στην εικόνα 28 αποτυπώνονται τα αποτελέσματα για NARMA10 και NARMA5 για την σύγκρισή τους (για την συγκεκριμένη απεικόνιση υλοποιήθηκε η συνάρτηση : **plot10_5compERR_SpecRad.m**). Οι καμπύλες των κόμβων για NARMA10 είναι πιο αραιά αποτυπωμένες με μεγαλύτερες τιμές σφάλματος NRMSE σε σχέση με αυτές για NARMA5. Ωστόσο, όπως ήδη αναφέρθηκε, το NARMA10 έχει καλύτερη μνήμη καταστάσεων από το NARMA5.



Εικόνα 28. NARMA10 για 400 και 500 κόμβους reservoir και NARMA5 για 400 και 500 κόμβους reservoir.

Μια εναλλακτική ερμηνεία για τα παραπάνω αποτελέσματα δίνεται με την υλοποίηση σχετικής συνάρτησης **plot10_5compERR_ResNodes.m**, που στον x άξονα υπάρχει ο αριθμός των κόμβων και η κάθε καμπύλη δείχνει πως επηρεάζεται το NRMSE για συγκεκριμένο $\rho(W)$. Η διαδικασία αυτή έγινε για λόγους διερεύνησης της συμπεριφοράς διάφορων παραμέτρων που επηρεάζουν ένα reservoir.

4 ΠΕΙΡΑΜΑΤΙΚΟ ΜΕΡΟΣ 2: ΥΛΟΠΟΙΗΣΗ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ ΓΙΑ ΔΙΑΦΟΡΕΣ ACTIVATION FUNCTION

Στο κεφάλαιο 3 εξετάστηκε η περίπτωση χρήσης της **tanh** reservoir activation function. Σαν input data στο σύστημα χρησιμοποιήθηκαν για όλες τις εκτελέσεις του προγράμματος τα ίδια δεδομένα χρονοσειράς (input Sequence) που παρήχθησαν με το μοντέλο της NARMA. Τα αποτελέσματα αποθηκεύτηκαν σε ένα excel αρχείο (*activationTanh.xlsx*) και σχολιάστηκαν με την βοήθεια διαγραμμάτων.

Στο παρόν κεφάλαιο, προκειμένου να γίνει σύγκριση της απόδοσης (NRMSE) του συστήματος κατά την φάση της εκπαίδευσης για διαφορετικές reservoir activation function, υλοποιήθηκαν διάφορες συναρτήσεις (tanh, sigmoid, identity, RELU, πολυωνυμική $\alpha * x^2$) και εκτελέστηκε το πρόγραμμα με διάφορες τιμές $\rho(W)$ και αριθμό reservoirNodes. Τα αποτελέσματα που παρήχθησαν αποθηκεύτηκαν σε αντίστοιχα excel αρχεία και έγινε απεικόνισή τους σε διαγράμματα με σκοπό την παραγωγή συμπερασμάτων. Έγινε χρήση των Input δεδομένων που προέκυψαν από το μοντέλο NARMA10, όπως αναφέρεται παραπάνω.

Για τις συναρτήσεις **tanh**, **sigmoid** και **identity** εξετάστηκαν τιμές $\rho(W)=[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]$ και αριθμό κόμβων reservoirNodes = [100, 200, 300, 400, 500, 600].

- Tanh (αρχείο excel *activationTanh.xlsx* και συνάρτηση την έτοιμη συνάρτηση της matlab για tanh)
- Sigmoid (αρχείο excel *activationSigmoid.xlsx* και συνάρτηση *sigmoid01.m*)
- Identity (αρχείο excel *activationIdentity.xlsx* και συνάρτηση *identity.m*)

Για τη συνάρτηση **RELU** και shifted RELU εξετάστηκαν τιμές $\rho(W)=[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]$ και αριθμό κόμβων reservoirNodes = [100, 200, 300, 400, 500, 600, 800, 1200].

- RELU (αρχείο excel *activationRELU.xlsx* και συνάρτηση *relu.m*)
- Shifted RELU (αρχείο excel *activationShiftRelu.xlsx* και συνάρτηση *shift_relu.m*)

Για τη συνάρτηση **$\alpha * x^2$** εξετάστηκαν τιμές $\rho(W)=[0.1, 0.3, 0.5, 0.7, 0.9]$ και αριθμό κόμβων reservoirNodes = [100, 200, 300, 400, 500, 600, 800, 1200, 1600, 2000].

- Πολυωνυμική συνάρτηση $\alpha * x^2$ με $\alpha=[0.1, 0.5, 1, 4, 6]$ (αρχείο excel *activation_X2.xlsx* και συνάρτηση *polynomial_x2.m*).

4.1 Διερεύνηση Προβλήματος NARMA10 για Activation Function Sigmoid

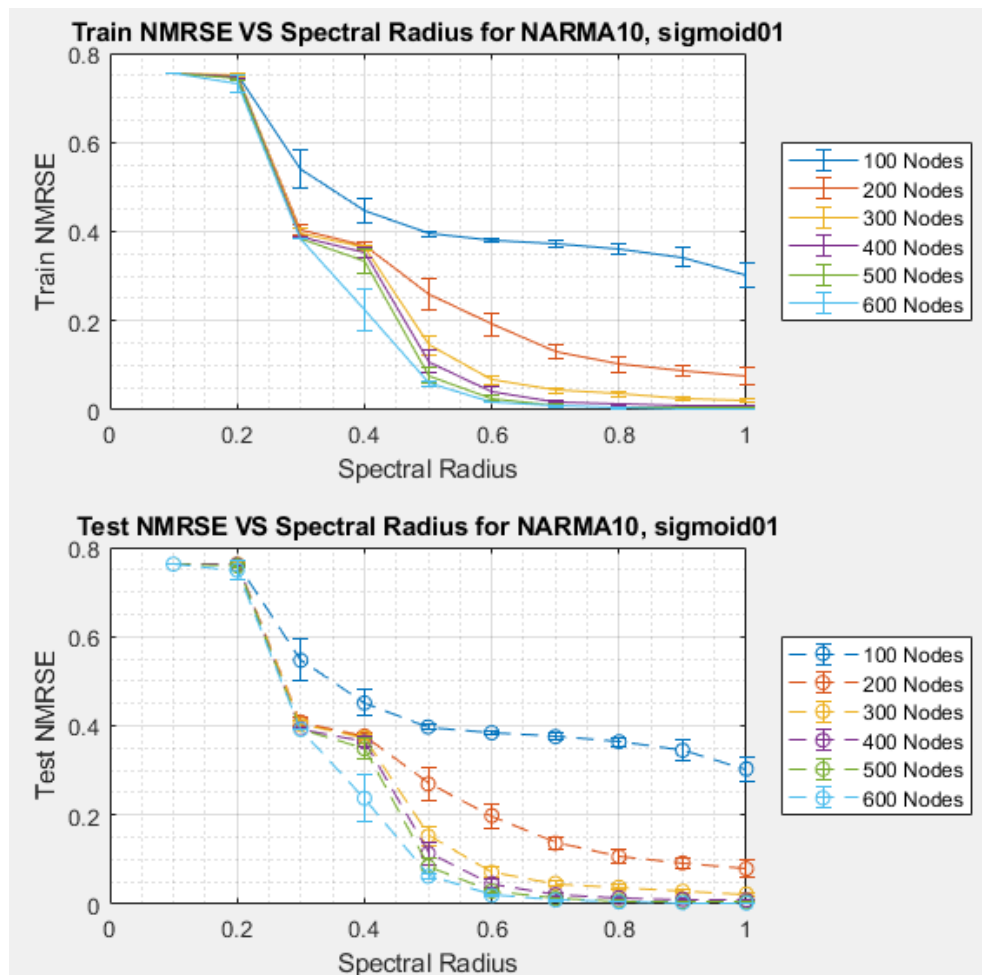
Στην *εικόνα 29* παρουσιάζονται τα αποτελέσματα της εκτέλεσης του προγράμματος με **sigmoid** activation function σε διάγραμμα train NRMSE- spectral radius και test NRMSE- spectral radius για διάφορες τιμές reservoir nodes. Η κάθε καμπύλη δείχνει την μεταβολή του σφάλματος σε σχέση με τα $\rho(W)$ για τις προαναφερθείσες τιμές κόμβων.

Το συμπέρασμα που προκύπτει είναι το ίδιο με αυτό της περίπτωσης για NARMA10 με activation function το tanh. Έτσι, ο υπολογιστικός χρόνος αυξάνεται όταν ο αριθμός των κόμβων μεγαλώνει, και το σφάλμα NRMSE μειώνεται. Επίσης, για σταθερό αριθμό κόμβων, όταν το $\rho(W)$ είναι μικρό, ($\rho(W) = 0.1$) το

NRMSE είναι αρκετά κοντά στην μονάδα και όσο το $\rho(W)$ μεγαλώνει το NRMSE μικραίνει. Για παράδειγμα, όπως καταγράφηκε στο αρχείο activationSigmoid.xlsx,

- για 400 κόμβους και $\rho(W)=0.1$, το Train NRMSE= 0.7555 και το Test NRMSE = 0.7626 ενώ,
- για 400 κόμβους και $\rho(W)= 0.8$, το Train NRMSE = 0.0130 και το Test NRMSE = 0.0146.

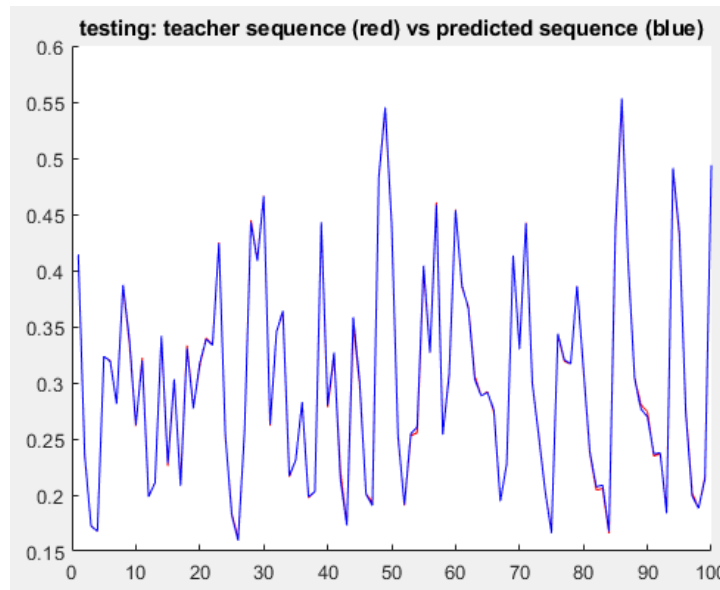
Ιδανικά είναι επιθυμητό να έχουμε μικρό αριθμό κόμβων, βέλτιστη τιμή NRMSE πλησίον του 0 (πχ. $\text{NRMSE} < 0.1$), ώστε η συγκεκριμένη activation function να παράγει μία έξοδο (predicted output) η οποία να προσεγγίζει σε μεγάλο βαθμό την επιθυμητή έξοδο (teacher output) που είχε παραχθεί βάσει του μοντέλου NARMA10. Επιπλέον, το $\rho(W) \leq 1$ συσχετίζεται με την ιδιότητα ESP και προκειμένου να επιτυγχάνεται η καλύτερη απόδοση του συστήματος, ορίζεται ανάλογα με το πρόβλημα προς επίλυση (Lukoševičius M. , 2012). Παρατηρείται ότι για όλες τις περιπτώσεις τιμών κόμβων, το σφάλμα NRMSE μειώνεται όσο αυξάνει το $\rho(W)$. Σύμφωνα με το διάγραμμα (εικόνα 29), καλύτερη απόδοση κατά την εκπαίδευση υπάρχει για $\rho(W) = [0.6, 1.0]$ και αριθμό κόμβων > 300 . Αυτό επιβεβαιώνεται αρχικά από το διάγραμμα του test NRMSE όπου για τις παραπάνω περιοχές τιμών το σφάλμα NRMSE εξακολουθεί να είναι πλησίον του μηδενός. Η τιμή της βέλτιστης λύσης φαίνεται στον πίνακα 6 .



Εικόνα 29 Διάγραμμα απεικόνισης σφάλματος NRMSE με activation function sigmoid για $\rho(W) = [0.1, 1]$

Ένας εναλλακτικός τρόπος παρουσίασης των παραπάνω αποτελεσμάτων που υλοποιήθηκε με σχετική συνάρτηση `plotNARMA10_Err_ResNodes.m`, είναι στον άξονα των x να υπάρχουν οι κόμβοι του reservoir και η κάθε καμπύλη να αντιστοιχεί σε διαφορετικά $\rho(W)$.

Ωστόσο, από το διάγραμμα της *εικόνας 29*, προκύπτει ότι για την sigmoid activation function, μια καλή λύση αποτελεί η περίπτωση των 400 κόμβων και $\rho(W)=0.7$ διότι, όπως φαίνεται και στην *εικόνα 30*, κατά την φάση του testing, οι καμπύλες των teacher και predicted ακολουθιών σχεδόν συμπίπτουν. Αυτό σημαίνει ότι το σφάλμα NRMSE είναι το ίδιο για τις συγκεκριμένες τιμές των παραμέτρων και άρα το σύστημα εκπαιδεύτηκε σωστά.



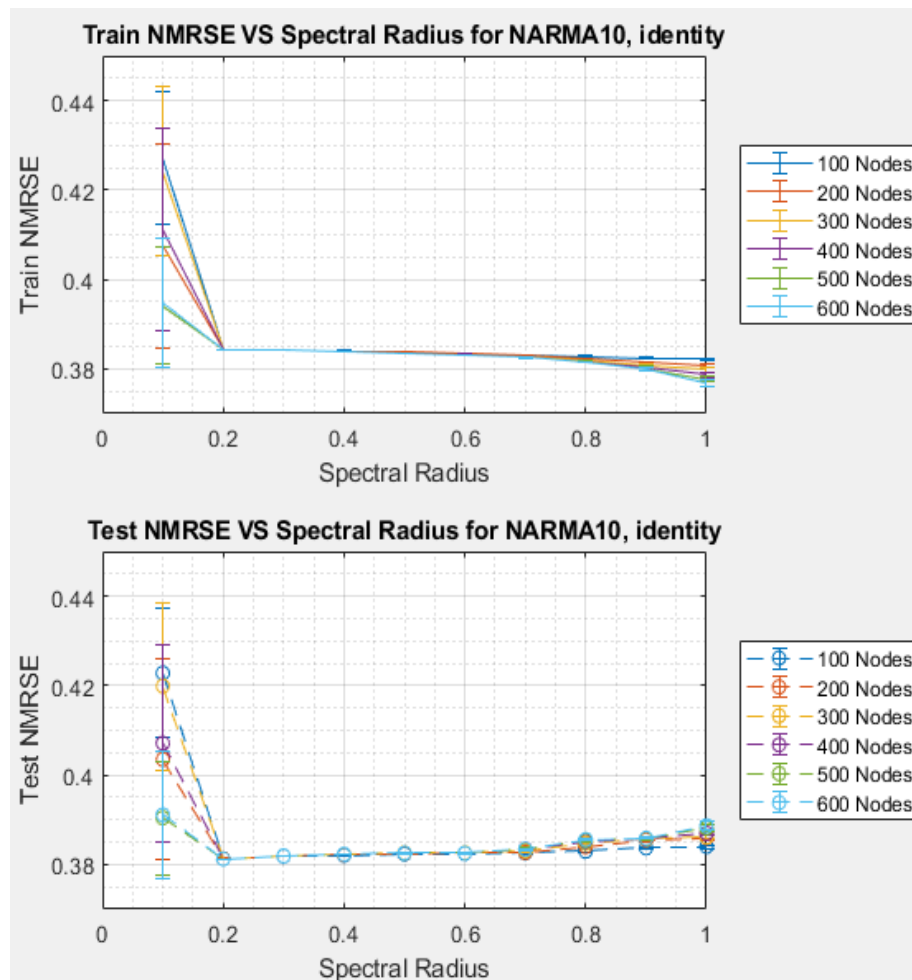
Εικόνα 30. Διάγραμμα για Reservoir Nodes =400 και Spectral Radius=0.7 και activation function sigmoid ταυτίζονται

4.2 Διερεύνηση προβλήματος NARMA10 για Activation Function Identity

Στην *εικόνα 31* παρουσιάζονται τα αποτελέσματα της εκτέλεσης του προγράμματος με **identity** activation function σε διάγραμμα train NRMSE- spectral radius και test NRMSE- spectral radius για διάφορες τιμές reservoir nodes και χρονοσειρά input δεδομένων που παράχθηκε με NARMA10. Η κάθε καμπύλη δείχνει την μεταβολή του σφάλματος σε σχέση με τα $\rho(W)$ για τις προαναφερθείσες τιμές κόμβων.

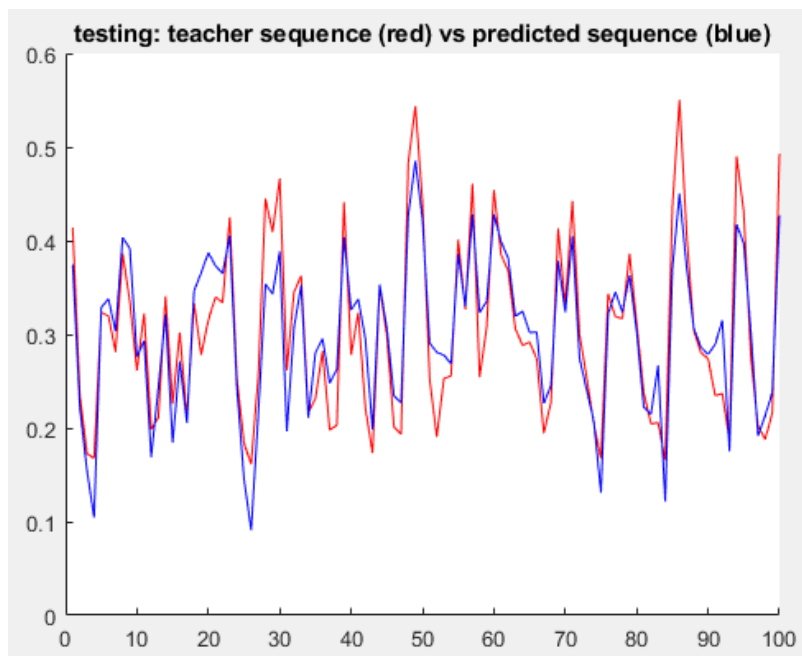
Παρατηρείται ότι, για μεγάλες τιμές $\rho(W)$, κοντά στην μονάδα και αυξανόμενο αριθμό κόμβων το σφάλμα αρχίζει να μειώνεται. Για τις τιμές reservoir nodes που εξετάστηκαν, όπως παρατηρείται από τα train και test NRMSE διαγράμματα, το σύστημα δεν είναι αποδοτικό και η βέλτιστη λύση του έχει μεγάλο σφάλμα $\text{testNRMSE} > 0.3$. Δειγματοληπτικά έγινε εκτέλεση του προγράμματος για 800 και για 2000 κόμβους και τα NRMSE που προέκυψαν στην 1^η περίπτωση (800 κόμβοι) είναι $\text{TrainNRMSE} = 0.3790$ και $\text{TestNRMSE} = 0.3868$, στην 2^η περίπτωση (2000 κόμβοι) είναι $\text{TrainNRMSE} = 0.3850$ και $\text{TestNRMSE} = 0.3912$.

Η identity είναι μια γραμμική activation function που ότι λαμβάνει σαν είσοδο το περνάει και στην έξοδο της (result = input). Αυτός είναι ο λόγος που στα διαγράμματα της εικόνας 31, δεν παρατηρείται σημαντική αλλαγή στις τιμές του NRMSE. Με την βοήθεια του excel αρχείου βρέθηκε ότι η βέλτιστη τιμή βρίσκεται για (100-600) reservoir κόμβους και $\rho(W) = 0.2$ και αναγράφεται στον πίνακα 6.



Εικόνα 31. Διάγραμμα απεικόνισης σφάλματος NRMSE με activation function Identity για $\rho(W) = [0.1, 1]$ σε προσαρμοσμένη κλίμακα για ευκρίνεια

Στην *εικόνα 32*, γίνεται εμφανές ότι κατά την φάση του testing, οι καμπύλες των teacher και predicted ακολουθιών δεν συμπίπτουν, άρα το σύστημα δεν εκπαιδεύτηκε σωστά με identity activation function.



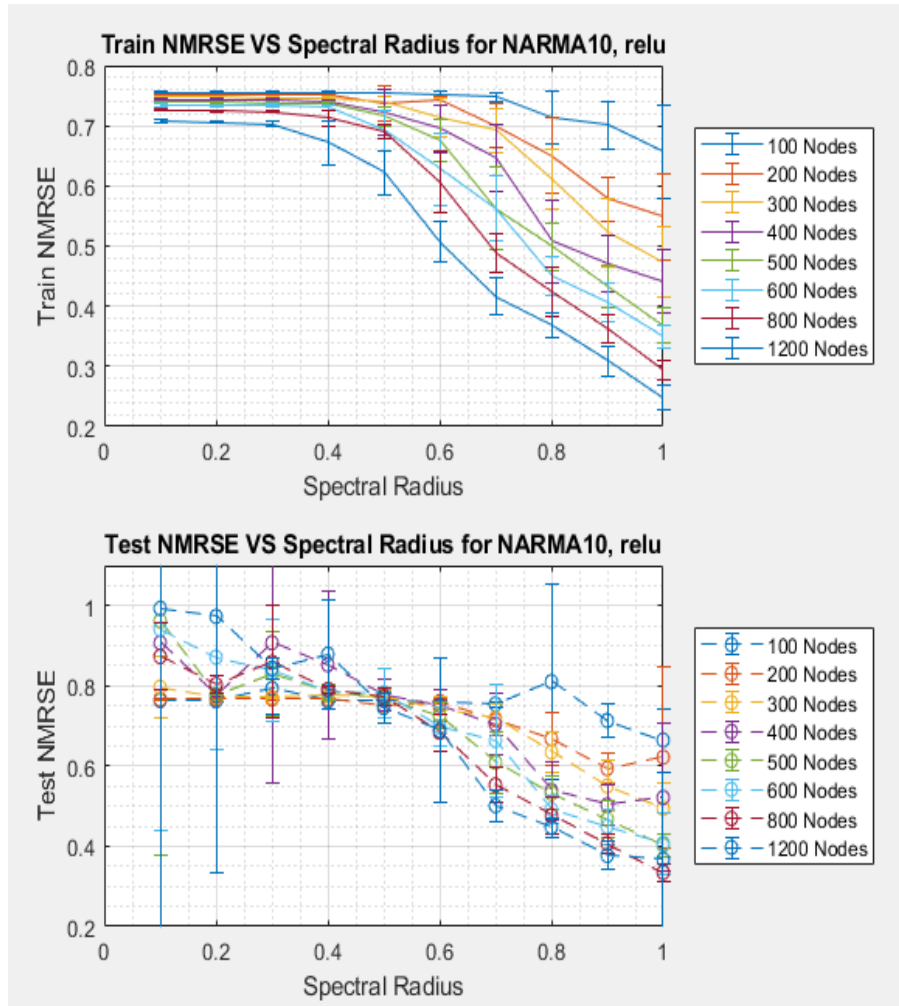
Εικόνα 32 Διάγραμμα για Reservoir Nodes =600 και Spectral Radius=1 και activation function identity δεν ταυτίζονται

4.3 Διερεύνηση Προβλήματος NARMA10 για Activation Function RELU

Στη συνέχεια, εξετάστηκε η περίπτωση χρήσης activation function RELU, για το προαναφερθέν input sequence. Στην *εικόνα 33* παρουσιάζονται διαγράμματα train και test NRMSE - spectral radius για διάφορες τιμές κόμβων στο reservoir. Η κάθε καμπύλη δείχνει την μεταβολή του σφάλματος σε σχέση με τα $\rho(W)$ για τις προαναφερθείσες τιμές κόμβων.

Η activation function RELU είναι μια μη γραμμική συνάρτηση, η οποία για τιμές $data < 0$ παράγει σαν έξοδο μηδενικές τιμές, ενώ για $data \geq 0$ οι τιμές υπολογίζονται όμοια με την identity activation function (γραμμικά). Γι' αυτό τον λόγο στο train διάγραμμα της *εικόνας 33* αρχικά φαίνεται ότι το σύστημα δεν μπορεί να εκπαιδευτεί, ενώ στην πορεία η εκπαίδευση γίνεται σωστά.

Διαπιστώθηκε, όπως αναφέρεται και στην βιβλιογραφία, ο υπολογιστικός χρόνος της εκτέλεσης του προγράμματος για RELU είναι μικρότερος συγκριτικά με την tanh και την sigmoid activation function, για τον ίδιο αριθμό reservoir nodes και $\rho(W)$. Αυτό οφείλεται, αφενός στον αυξημένο υπολογιστικό χρόνο για τον παράγοντα $\exp$ στις tanh και sigmoid, αφετέρου στην ύπαρξη αρκετών μηδενικών τιμών (αυξημένο sparsity) μεταξύ των συνδέσεων των κόμβων λόγω της RELU.



Εικόνα 33. Διάγραμμα απεικόνισης σφάλματος NMRSE με activation function RELU για $\rho(W) = [0.1, 1]$ και όλους τους κόμβους

Από το διάγραμμα train NMRSE vs Spectral Radius (εικόνα 33), παρατηρείται ότι αρχικά το σύστημα, στην περιοχή τιμών $\rho(W)=[0.1, 0.4]$, δεν έχει μεταβολές στο σφάλμα NMRSE, το οποίο είναι μεγάλο. Στην συνέχεια, για $\rho(W) > 0.4$ το σφάλμα NMRSE μειώνεται απότομα όσο ο αριθμός των $\rho(W)$ μεγαλώνει για κάθε αριθμό κόμβων. Στην συγκεκριμένη περίπτωση (εικόνα 33) για reservoir nodes =100 και $\rho(W)=1$, το train NMRSE= 0.6572, ενώ για reservoir nodes = 1200 και $\rho(W) = 1$, το train NMRSE= 0.2480. Από το διάγραμμα γίνεται εμφανές ότι όσο οι κόμβοι αυξάνονται το NMRSE μειώνεται για κάθε $\rho(W)$.

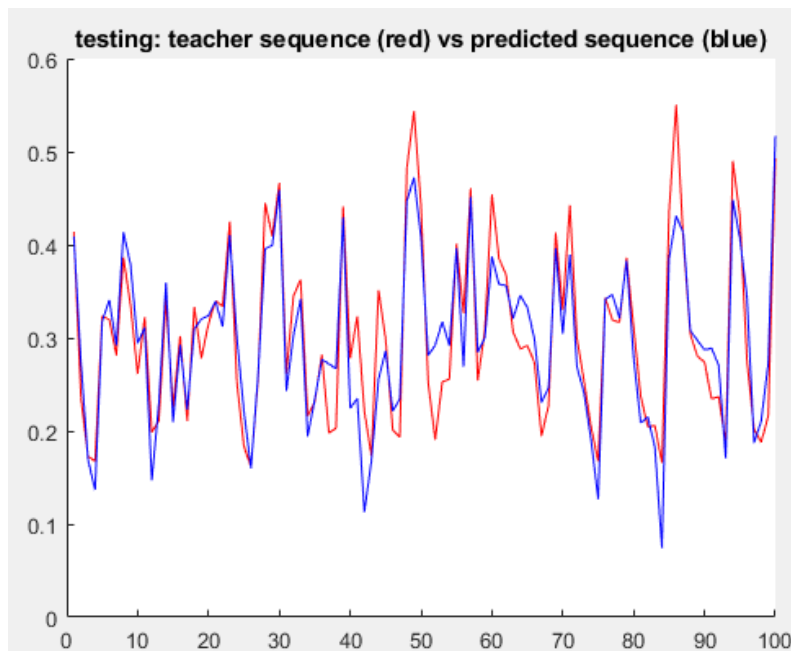
Δειγματοληπτικά έγινε εκτέλεση του προγράμματος και για 1600, για 2000 και για 3000 κόμβους με $\rho(W)= 0.8$ και 1.0 και τα NMRSE που προέκυψαν :

- 1600 κόμβοι, $\rho(W)=0.8$ είναι TrainNMRSE = 0.3421 , TestNMRSE = 0.4403, και $\rho(W)=1$ είναι TrainNMRSE = 0.1982 , TestNMRSE = 0.2657
- 2000 κόμβοι, $\rho(W)=0.8$ είναι TrainNMRSE = 0.3111 , TestNMRSE = 0.4363, και $\rho(W)=1$ είναι TrainNMRSE = 0.1747 , TestNMRSE = 0.2503

- 3000 κόμβοι, $\rho(W)=0.8$ είναι $\text{TrainNRMSE} = 0.2444$, $\text{TestNRMSE} = 0.4259$, και $\rho(W)=1$ είναι $\text{TrainNRMSE} = 0.1213$, $\text{TestNRMSE} = 0.2285$. Παρότι η τιμή είναι αποδεκτή και ίσως βέλτιστη για activation function RELU, ο υπολογιστικός χρόνος είναι αυξημένος.

Στην *εικόνα 33*, φαίνεται ότι, αρχικά το σύστημα (testNRMSE vs Spectral radius) για $\rho(W) < 0.4$ και για όλες τις περιπτώσεις τιμών κόμβων δεν ακολουθεί την ίδια πορεία με αυτή του trainNRMSE vs spectral Radius . Για $\rho(W) > 0.5$ το σύστημα αρχίζει να «μαθαίνει» (να εκπαιδεύεται).

Για την ReLu activation function από το διάγραμμα της *εικόνας 33*, προκύπτει ότι αν και οι τιμές για το σφάλμα NRMSE είναι αποδεκτές, θεωρούνται μεγάλες, έτσι σαν βέλτιστη θεωρήθηκε η περίπτωση των 3000 κόμβων με $\rho(W)=0.8$. Το διάγραμμα που προκύπτει για την αξιολόγηση της φάσης του testing, θα έχει καμπύλες teacher και predicted ακολουθιών που δεν θα συμπίπτουν (*εικόνα 34*) λόγω της διαφοράς μεταξύ των τιμών των trainNRMSE και testNRMSE .



Εικόνα 34. Διάγραμμα για Reservoir Nodes =3000 και Spectral Radius=0.8 και activation function ReLu δεν ταυτίζονται

Όπως φαίνεται από τα παραπάνω, προκειμένου να επιτευχθεί σχετικά καλό σφάλμα testNRMSE πρέπει ο αριθμός των κόμβων να είναι αυξημένος (πχ. 3000 κόμβοι και $\rho(W)=1$ και $\text{TestNRMSE}=0.2285$), αν και επιθυμητό είναι να επιτευχθεί χαμηλό σφάλμα με όσο το δυνατόν λιγότερους κόμβους. Δοκιμάστηκε η περίπτωση όπου η καμπύλη της ReLu μετατοπίζεται στον άξονα των x με σκοπό την πιθανή επίτευξη χαμηλότερων τιμών NRMSE. Για την εύρεση της τιμής μετατόπισης, υπολογίσθηκε η ελάχιστη και μέγιστη τιμή του x για κάθε κόμβο και μετά από δοκιμές επιλέχθηκε η τιμή 0.1 ($< \max(x)$) η οποία έδινε τα καλύτερα αποτελέσματα.

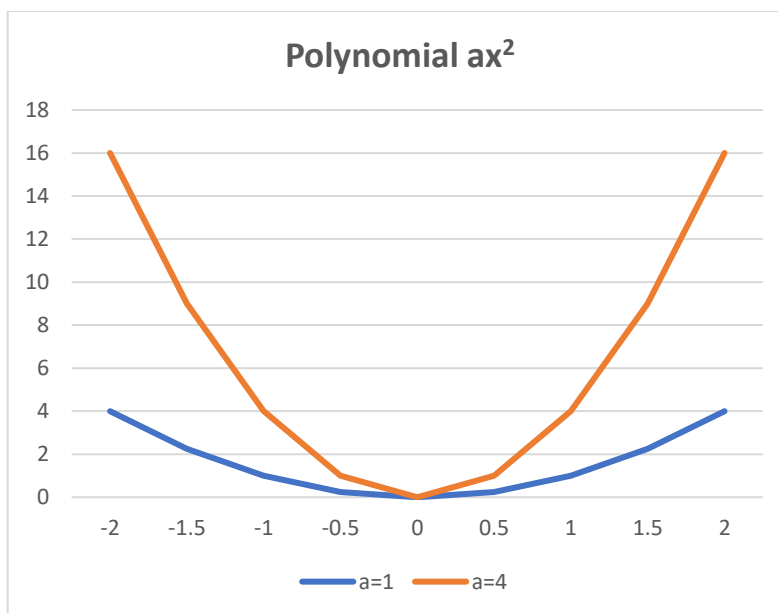
Στον παρακάτω πίνακα 4 φαίνεται η σύγκριση των αποτελεσμάτων που επιτεύχθηκαν με shift ReLu και ReLu . Έτσι, με shift ReLu προκύπτει τιμή testNRMSE=0.2175 για 1600 nodes $\rho(W)=1$, ενώ με κανονική ReLu , απαιτούνται 3000 κόμβοι για παρόμοιο σφάλμα.

ResNodes	SpectralRad	AverageTrain	AverageTest	Activation
800	0.6	0.3507	0.6138	shift_relu
800	0.8	0.3097	0.3629	shift_relu
800	1	0.2347	0.2726	shift_relu
1200	0.6	0.3303	0.4876	shift_relu
1200	0.8	0.2621	0.3521	shift_relu
1200	1	0.1868	0.2441	shift_relu
1600	0.8	0.2189	0.3391	shift_relu
1600	1	0.1510	0.2175	shift_relu
1600	0.8	0.3421	0.4403	relu
1600	1	0.1982	0.2657	relu
2000	0.8	0.3111	0.4363	relu
2000	1	0.1747	0.2503	relu
3000	0.8	0.2444	0.4259	relu
3000	1	0.1213	0.2285	relu

Πίνακας 4. Συγκριτικός πίνακας για Relu και SHift ReLu αποτελέσματα.

4.4 Διερεύνηση Προβλήματος NARMA10 για Πολυωνυμική Activation Function ax^2 .

Η πολυωνυμική συνάρτηση ενεργοποίησης ανάλογα με την τιμή του a παίρνει την μορφή της εικόνας 35. Όσο μεγαλύτερο είναι το a τόσο η κλίση της καμπύλης αυξάνεται.



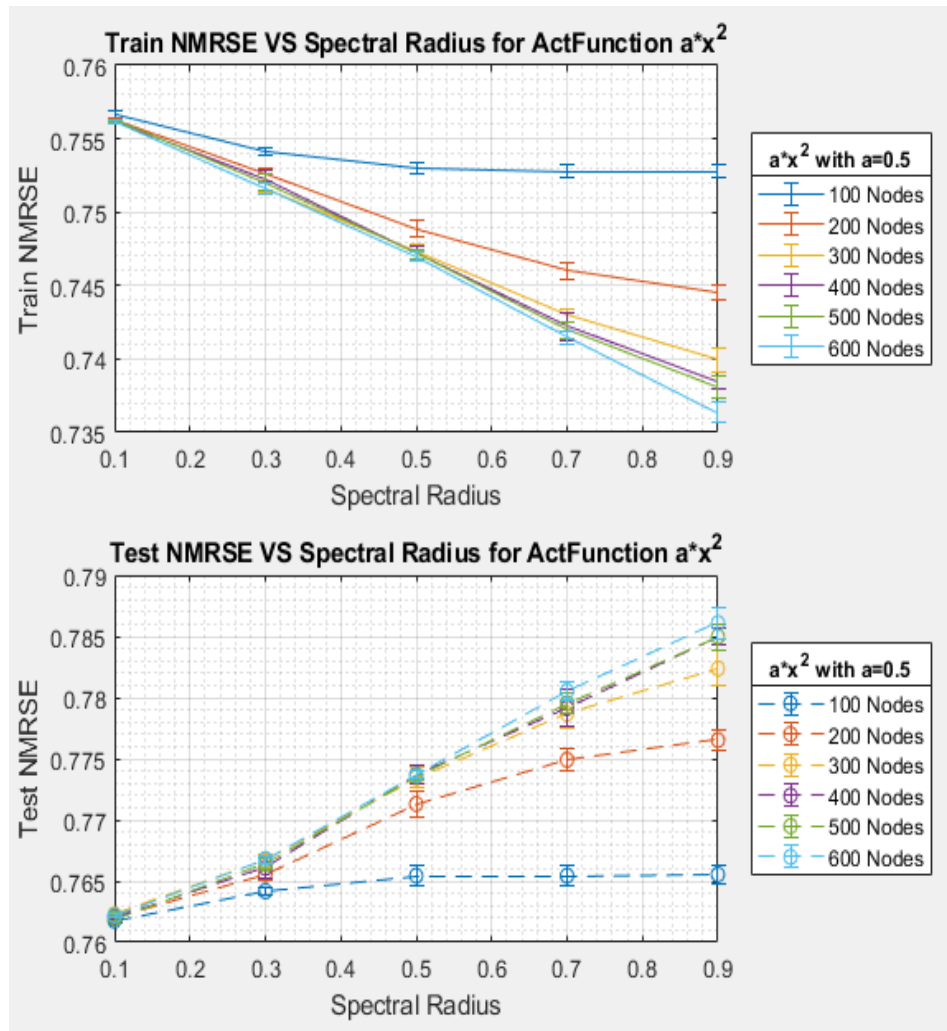
Εικόνα 35. Πολυωνυμική activation function για $a=1$ και $a=4$

Για τη συνάρτηση $a \cdot x^2$ εξετάστηκαν τιμές $\rho(W)=[0.1, 0.3, 0.5, 0.7, 0.9]$ και αριθμό κόμβων $\text{reservoirNodes} = [100, 200, 300, 400, 500, 600, 800, 1200, 1600, 2000]$. Υλοποιήθηκε η συνάρτηση **polynomial_x2.m** προκειμένου να υπολογιστούν οι τιμές του αρχείου excel *activation_X2.xlsx*. Παρακάτω παρουσιάζονται τα σχετικά διαγράμματα για $a \cdot x^2$ με $a=[0.5, 1, 4, 6]$ και γίνεται εξαγωγή συμπερασμάτων. Η κάθε καμπύλη στα παρακάτω διαγράμματα δείχνει την μεταβολή του σφάλματος σε σχέση με τα $\rho(W)$ για τις προαναφερθείσες τιμές κόμβων.

➤ ΥΠΟΛΟΓΙΣΜΟΙ ΓΙΑ $a=0.1$ ΚΑΙ $a=0.5$

Για $a=0.1$ παρατηρήθηκε έγινε υλοποίηση και δεν παρατηρήθηκε κάποια μεγάλη διαφορά σε σχέση με το $a=0.5$.

Στο διάγραμμα της εικόνας 36, όπου το $a=0.5$, παρατηρείται μεγάλη τιμή σφάλματος ($\text{NRMSE}>0.7$), που δεν μειώνεται σημαντικά, όσο αυξάνεται ο αριθμός των κόμβων. Επιπλέον, το διάγραμμα του testNRMSE δεν έχει την ίδια “συμπεριφορά” με το trainNRMSE . Γι’ αυτό τον λόγο δεν συνεχίστηκε περεταίρω η ανάλυση.

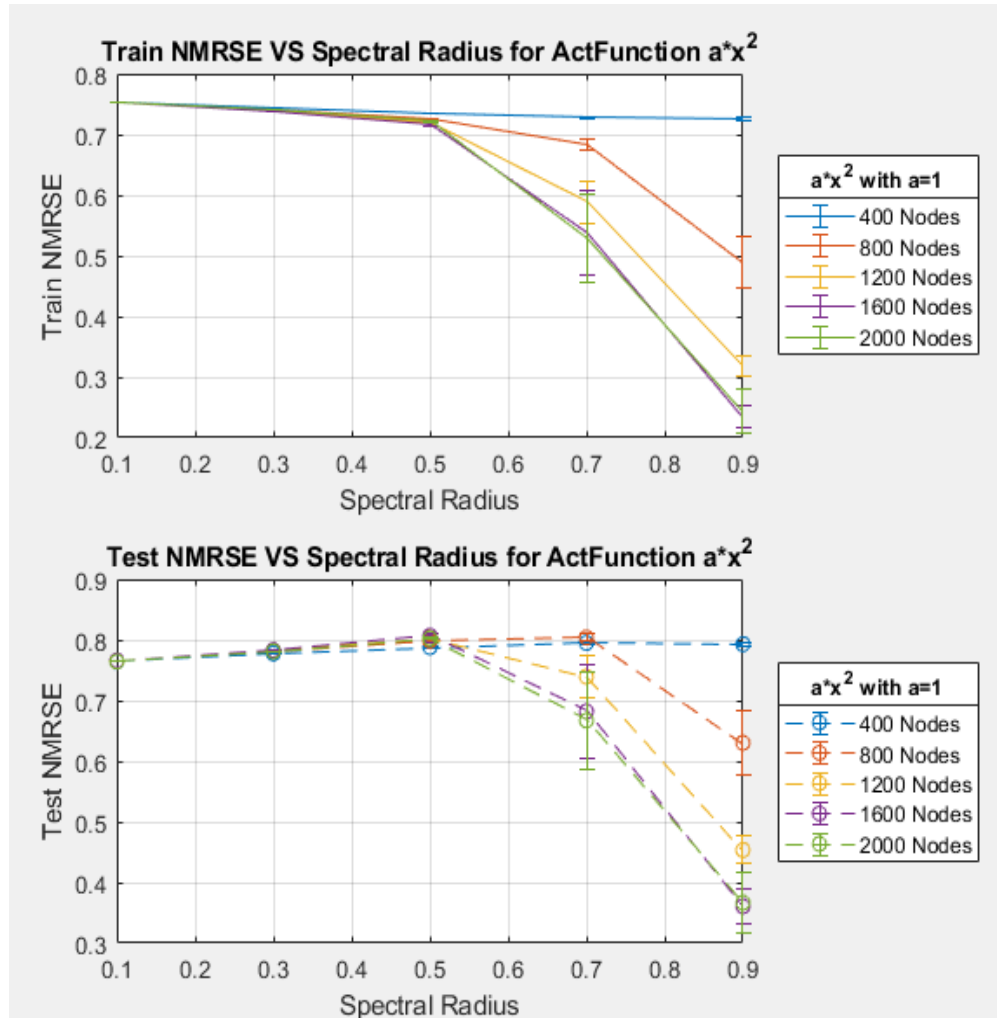


Εικόνα 36. Train και Test NRSME με activation function polynomial $a \cdot x^2$ με $a=0.5$

Από τα παραπάνω διαγράμματα προκύπτει ότι για μικρές τιμές a ($a < 1$) το σύστημα είναι αδύνατον να εκπαιδευτεί. Έτσι, στην περίπτωση που βέλτιστη θεωρηθεί η τιμή για 600 reservoir κόμβους και $\rho(W)=0.9$, αν σχεδιαστεί το διάγραμμα των καμπυλών για τις teacher και predicted ακολουθίες της φάσης train και test οι καμπύλες για οποιαδήποτε τιμή θα αποκλίνουν.

➤ ΥΠΟΛΟΓΙΣΜΟΙ ΓΙΑ $\alpha=1$

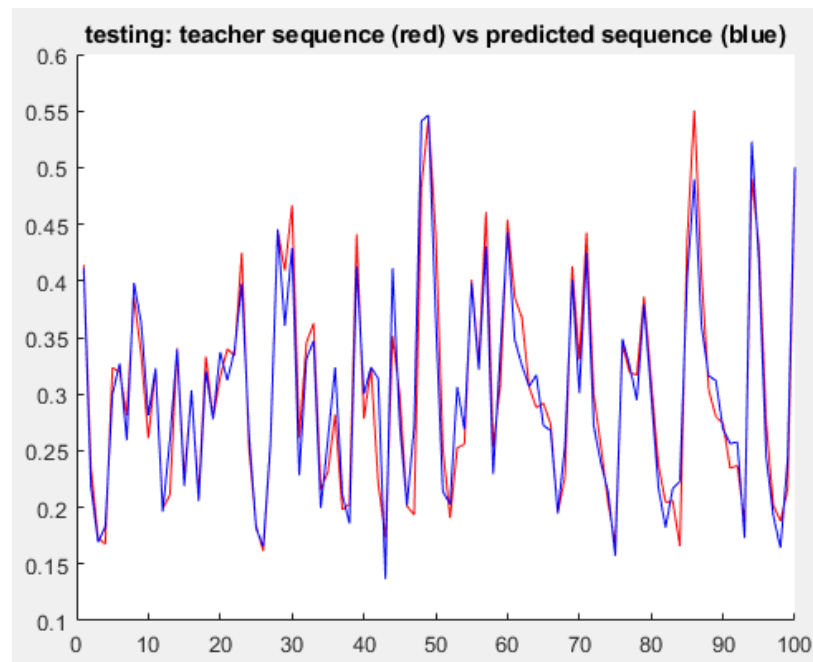
Στην συνέχεια εξετάσθηκε η περίπτωση τιμών $\alpha \geq 1$. Στο διάγραμμα της εικόνας 37 με $\alpha=1$ η πτωτική πορεία τιμών για το σφάλμα NRMSE εμφανίζεται για τιμές $\rho(W)>0.5$ για όλες τις τιμές κόμβων > 400 που εξετάσθηκαν. Συμπερασματικά, η πτωτική τάση είναι έντονη και το NRMSE μειώνεται όσο αυξάνεται ο αριθμός των κόμβων. Σε ένα φωτονικό NN που μπορεί να επεξεργαστεί ταχύτατα τα δεδομένα, η ύπαρξη μεγάλου αριθμού κόμβων δεν αποτελεί πρόβλημα.



Εικόνα 37. Train και Test NRSME με activation function polynomial $a \cdot x^2$ με $a=1$

Το σύστημα στην περίπτωση της πολυωνυμικής activation function με $\alpha=1$, παράγει αποτελέσματα πιο γρήγορα από ότι παρήγαγαν οι activation functions tanh, sigmoid και relu. Επίσης κάνει καλύτερη εκπαίδευση από στις δύο προηγούμενες περιπτώσεις $\alpha=0.1$ και $\alpha=0.5$ (εικόνα 37). Αν θεωρηθεί βέλτιστη η περίπτωση για 2500 reservoir κόμβους και $\rho(W)=0.9$ τότε το $\text{trainNRMSE} = 0.357988$ και το $\text{testNRMSE} = 0.482662$ είναι το βέλτιστο.

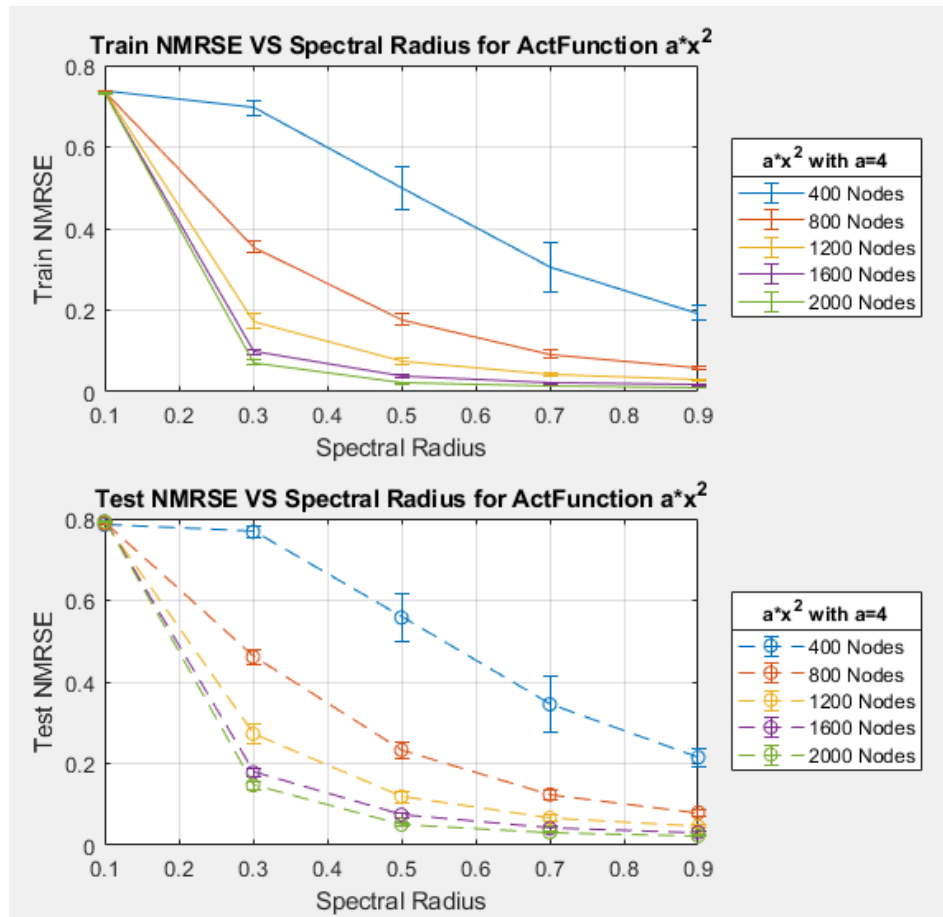
Παρακάτω (εικόνα 38) παρουσιάζεται το διάγραμμα των καμπυλών teacher και predicted ακολουθιών κατά την φάση του testing. Οι καμπύλες των ακολουθιών τείνουν να ταυτιστούν και άρα το σφάλμα να γίνει ακόμα μικρότερο. Πιθανών, αν αυξηθούν και άλλο οι κόμβοι του reservoir, να μικρύνει ακόμα περισσότερο ο αριθμός σφάλματος. Στο τρέχον σύστημα η αύξηση των κόμβων του reservoir, όπως ήδη αναφέρθηκε, θα οδηγούσε σε αύξηση του υπολογιστικού χρόνου.



Εικόνα 38 Διάγραμμα για Reservoir Nodes =2500 και Spectral Radius=0.9 και activation function polynomial με $\alpha=1$ οι καμπύλες τείνουν να ταυτιστούν

➤ ΥΠΟΛΟΓΙΣΜΟΙ ΓΙΑ $\alpha=4$

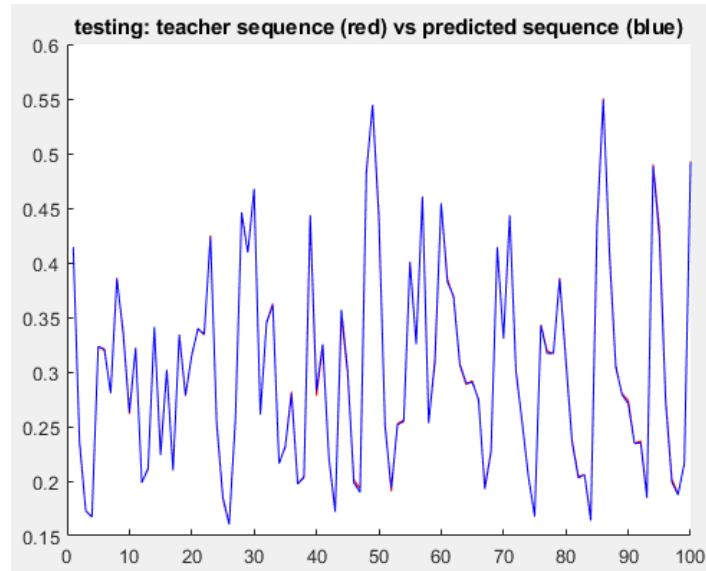
Στην συνέχεια (εικόνα 39) δημιουργήθηκαν διαγράμματα train και test για $\alpha=4$. Παρατηρείται πως το σύστημα πετυχαίνει πολύ καλές επιδόσεις (train NRMSE) ακόμα και με μικρή τιμή αριθμού κόμβων, όπως όταν το $\rho(W) > 0.3$ οι τιμές του NRMSE < 0.1 . Παρόμοια αποτελέσματα εμφανίζονται και για το testNRMSE. Καθώς η τιμή του α αυξάνεται, μεγαλώνει η κλίση των καμπυλών και εμφανίζονται μικρές τιμές σφάλματος NRMSE για μικρότερες τιμές $\rho(W)$.



Εικόνα 39. Train και Test NRMSE με activation function polynomial $a \cdot x^2$ με $a=4$

Σύμφωνα με το παραπάνω διάγραμμα εικόνα 39 και σύμφωνα με τα αποτελέσματα του σχετικού αρχείου excel, η βέλτιστη τιμή παρουσιάζεται στην περίπτωση που ο αριθμός των κόμβων = 2000 και το $\rho(W)=0.9$ όπου το trainNRMSE=0.0101 και το testNRMSE=0.0229.

Λόγω της εικόνας 39, μπορεί να γίνει η θεώρηση ότι βέλτιστη λύση προκύπτει στην περίπτωση reservoir με αριθμό κόμβων = 2000 και $\rho(W)=0.9$. Προκειμένου να ελεγχθεί αν το σύστημα εκπαιδεύτηκε σωστά, κατασκευάζεται το διάγραμμα της εικόνας 40. Οι καμπύλες των ακολουθιών teacher και predicted συμπίπτουν, άρα η εκπαίδευση του συστήματος έχει εκτελεστεί με επιτυχία.



Εικόνα 40 Διάγραμμα για Reservoir Nodes =2000 και Spectral Radius=0.9 και activation function polynomial με $\alpha=4$, οι καμπύλες ταυτίζονται

➤ ΥΠΟΛΟΓΙΣΜΟΙ ΓΙΑ $\alpha > 5$

Με το πρόγραμμα esnToolbox εξετάστηκε η περίπτωση για τιμές $\alpha > 5$. Παρατηρήθηκε ότι όσο περισσότερο μεγαλώνει η τιμή του α , τόσο πιο δύσκολο είναι το πρόγραμμα να εκτελεστεί για μεγάλο αριθμό κόμβων και $\rho(W) > 0.5$. Έτσι, στην περίπτωση του $\alpha=6$ παρότι παρουσιάστηκε δυσκολία στην εκτέλεση του προγράμματος, απομονώθηκαν τιμές $NRSME < 0.1$, όπως φαίνεται στον παρακάτω πίνακα 5.

Δειγματοληπτικά έγινε η εκτέλεση του προγράμματος με $\alpha=[8, 10, 12]$, για 400 κόμβους reservoir και $\rho(W) = [0.4, 0.5]$ προκειμένου να ελεγχθεί η απόδοση της πολυωνυμικής συνάρτησης συγκριτικά με την tanh και την ReLu για τις ίδιες συνθήκες. Οι τιμές για tanh και ReLu προέκυψαν από προηγούμενες εκτελέσεις του προγράμματος.

ResNodes	SpectralRad	AverageTrain	AverageTest	Activation	Alpha
800	0.3	0.2149	0.2865	polynomial_x2	6
800	0.5	0.0897	0.1197	polynomial_x2	6
1200	0.3	0.0939	0.1466	polynomial_x2	6
1200	0.5	0.0345	0.0550	polynomial_x2	6
1600	0.3	0.0464	0.0882	polynomial_x2	6
1600	0.5	0.0188	0.0352	polynomial_x2	6
400	0.5	0.1614	0.1852	polynomial_x2	8
400	0.5	0.1430	0.1600	polynomial_x2	10
400	0.4	0.1508	0.1780	polynomial_x2	12
400	0.4	0.0117	0.0130	tanh	
400	0.5	0.0082	0.0091	tanh	
400	0.4	0.7386	0.8532	ReLu	
400	0.5	0.7224	0.7791	ReLu	

Πίνακας 5. Δειγματοληπτικά αποτελέσματα trainNRMSE και testNRMSE για πολυωνυμική με $\alpha > 5$, και tanh, ReLu

Στον παραπάνω πίνακα (πίνακας 5) φαίνεται ότι με βάση τα τρεξίματα που έγιναν για $\alpha > 5$, καλύτερη απόδοση το σύστημα δίνει με την πολυωνυμική με $\alpha=6$ όταν ο αριθμός των κόμβων ισούται με 1600 και το $\rho(W)=0.5$, όπου το TestNRMSE=0.0352.

4.5 Γενικά Συμπεράσματα πειραματικής υλοποίησης

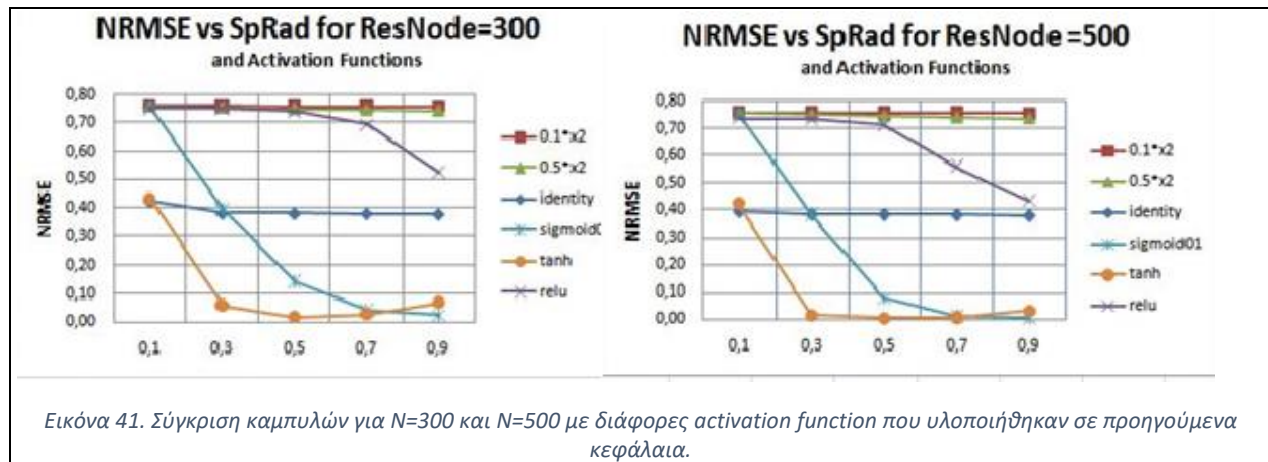
Στις ενότητες 3 και 4 έγινε πρακτική υλοποίηση επίλυσης του προβλήματος NARMA10 για διάφορες μη-γραμμικές activation functions που χρησιμοποιούνται στην βιβλιογραφία όπως οι tanh, sigmoid, ReLu και παρουσιάζουν πολύ καλά αποτελέσματα. Οι tanh και sigmoid έχουν τον παράγοντα $\exp$ και αυτό καθιστά την επίλυση προβλημάτων πιο αργή. Για τον σκοπό αυτό μια καλή λύση θεωρήθηκε η ReLu, πρακτικά όμως δεν αποτελεί καλή λύση καθώς περιέχει και γραμμικές τιμές και γι' αυτό η σωστή εκπαίδευση στο σύστημα αργεί να ξεκινήσει. Προκειμένου να επιτευχθεί άμεση καλύτερη εκπαίδευση στο σύστημα (σε σύγκριση με την ReLu) μελετήθηκε η πολυωνυμική ax^2 με $a \in R$, η οποία παρουσιάζει μικρότερη υπολογιστική πολυπλοκότητα από τις tanh και sigmoid. Παρακάτω ακολουθεί σύγκριση συμπεριφοράς των activation functions.

Στον πίνακα 6, συγκεντρώνονται οι περιοχές τιμών στις οποίες η κάθε μία από τις παραπάνω reservoir activation functions που εμφανίζονται στα διαγράμματα, παρουσιάζουν βέλτιστη περιοχή, δηλαδή περιοχή με το μικρότερο NRMSE. Για τις identity και RELU δεν ορίστηκε εύρος τιμών με βέλτιστη λύση, καθώς η καλύτερη τιμή NRMSE στις συγκεκριμένες περιπτώσεις είναι μεγάλη (NRMSE >0.7) συγκριτικά με τις τιμές NRMSE που προκύπτουν με τις υπόλοιπες activation functions.

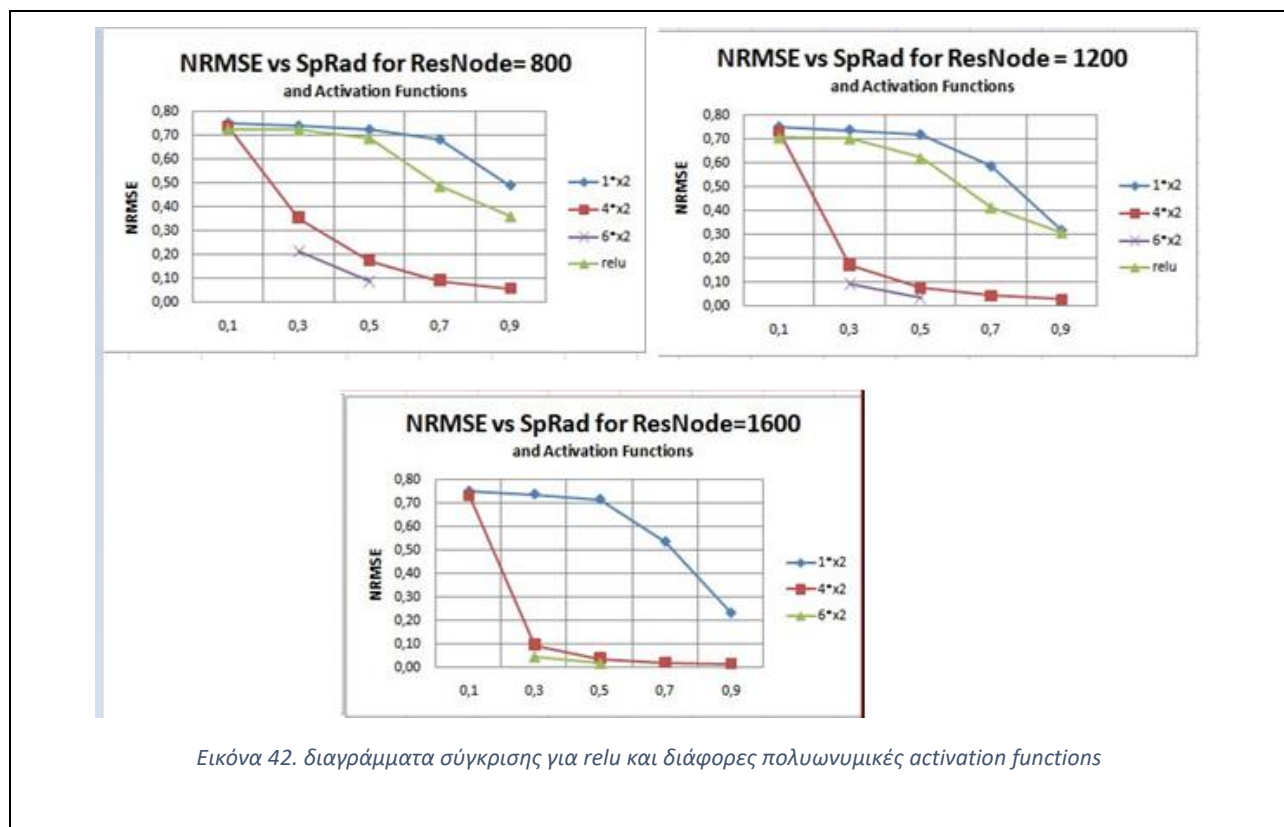
Activation function	Best NRMSE για Reservoir Nodes	Best NRMSE για $\rho(W)$	Best Test NRMSE solution		
			testNRMSE	Nodes	$\rho(W)$
Tanh	[400-600]	[0.6-1.0]	0.0022	600	0.5
Sigmoid	[400-600]	[0.6-1.0]	0.0026	600	1.0
Identity	$\forall ResNode, NRMSE > 0.3$	$\forall \rho(W), NRMSE > 0.3$	0.3813	100	0.2
ReLu	[1600-3000]	[1.0]	0.2285	3000	1.0
ReLu Shifted	[800-1600]	[1.0]	0.2175	1600	1.0
Polynomial ax^2 με $a = 0.1$	$\forall ResNode, NRMSE > 0.7$	$\forall \rho(W), NRMSE > 0.7$	0.7599	200	0.1
Polynomial ax^2 με $a = 0.5$	[600], NRMSE>0.7	[0.9], NRMSE>0.7	0.7618	100	0.1
Polynomial ax^2 με $a = 1$	[1200-2000]	[0.9]	0.3607	1600	0.9
Polynomial ax^2 με $a = 4$	[1200-2000]	[0.5-0.9]	0.0229	2000	0.9

Πίνακας 6. Βέλτιστες περιοχές τιμών κόμβων και $\rho(W)$ για όλες τις activation function που δοκιμάστηκαν με NARMA10

Παρακάτω, στην εικόνα 41, παρουσιάζονται για αριθμό κόμβων $N=300$ και $N=500$ οι καμπύλες για activation function polynomial ax^2 με $a=0.1$, $a=0.5$, identity, sigmoid, tanh και relu. Συγκριτικά μεταξύ τους αυτή που φέρνει καλύτερα αποτελέσματα είναι η tanh και στις δύο περιπτώσεις.



Στην εικόνα 42 παρατηρείται η συμπεριφορά των relu και polynomial με $a=1$, $a=4$ και $a=6$ για αριθμό κόμβων $N=800$ και $N=1600$. Για $a=6$ όπως προαναφέρθηκε δεν υπολογίστηκαν όλες οι τιμές $\rho(W)$, μπορεί όμως να προκύψει το συμπέρασμα πως από τα παρακάτω διαγράμματα καλύτερες επιδόσεις όσον αφορά το σφάλμα NRMSE εμφανίζονται στην περίπτωση της πολυωνμικής με $a=6$.



Στη συνέχεια παρουσιάζονται τα συμπεράσματα που προκύπτουν από την σύγκριση της συμπεριφοράς των καμπυλών για διάφορες activation functions σε σύγκριση με την πολυωνυμική. Για την εξαγωγή συμπερασμάτων λαμβάνονται υπόψιν οι τιμές του παραπάνω πίνακα 6 και τα παραπάνω διαγράμματα (εικόνα 41, 42).

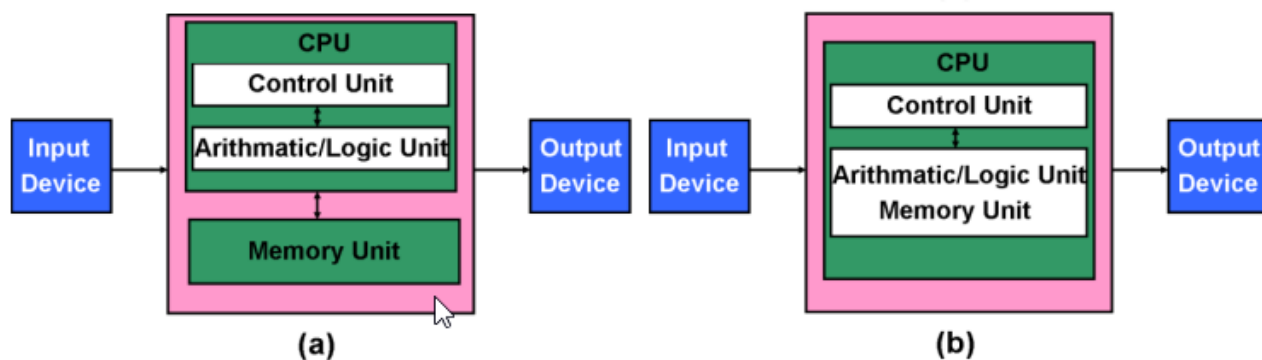
- Όσο το α αυξάνεται (μεγαλώνει η κλίση της καμπύλης του αx^2), για τον ίδιο αριθμό κόμβων πρέπει το $\rho(W)$ να ελαττώνεται έτσι ώστε να μπορέσει το πρόγραμμα να εκτελεστεί χωρίς πρόβλημα.
- Για το ίδιο $\rho(W)$ όσο το α αυξάνεται, η τιμή του σφάλματος NRMSE ελαττώνεται.
- Παρατηρώντας όλες τις υλοποιήσεις η $\tanh$ μαζί με την sigmoid παρέχουν για τις ίδιες συνθήκες, καλύτερα NRMSE σε σχέση με την αx^2 .
- Η ταχύτητα εκτέλεσης την $\tanh$ και της sigmoid είναι υπολογιστικά αργότερη σε σύγκριση με την αx^2 , λόγω της μεγαλύτερης υπολογιστικής πολυπλοκότητας που παρουσιάζει η ύπαρξη του παράγοντα $\exp$.

5 ΦΩΤΟΝΙΚΗ ΤΕΧΝΟΛΟΓΙΑ ΚΑΙ ΦΩΤΟΝΙΚΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

5.1 Προβλήματα αρχιτεκτονικής Von Neuman

Από το 1940 που κατασκευάστηκαν οι πρώτοι ηλεκτρονικοί υπολογιστές μέχρι και σήμερα, η αρχιτεκτονική von Neumann αποτελεί την βάση της δομής τους. Όπως φαίνεται στο σχήμα της (εικόνα 43α), ένα von Neumann μοντέλο αποτελείται από μια κεντρική μονάδα επεξεργασίας (CPU) η οποία περιέχει μια μονάδα ελέγχου που είναι υπεύθυνη για τις αριθμητικές και λογικές πράξεις του συστήματος (ALU) και την μνήμη. Η CPU επικοινωνεί με την μονάδα μνήμης και τις I/O συσκευές και εκτελεί ροές εντολών (streams of instructions) που δέχεται (Eigenmann, 1998). Οι υπολογιστές von Neumann δουλεύουν αποδοτικά, όταν το πρόβλημα που καλούνται να επεξεργαστούν είναι αλγοριθμικό. Στην περίπτωση που κληθούν να λύσουν ένα πιο περίπλοκο ή αφηρημένο πρόβλημα, όπως η αναγνώριση λόγου ή η αναγνώριση προσώπου, αντιμετωπίζουν προβλήματα πχ μεγάλη ενεργειακή κατανάλωση (AlphaGo Vs Lee Sedol (El Srouji, 2022)) , καθώς και καθυστερήσεις κατά την μεταφορά δεδομένων μεταξύ της CPU και της μνήμης (Goi, 2020), (Zou, 2021), (Schuman C. D., 2017), (Van der Sande, 2017). Για να λυθούν τα προβλήματα του von Neumann μοντέλου, στα τέλη της δεκαετίας του '60 προτάθηκε από τους Kautz και Stone η τεχνολογία processing in memory (PIM), που είχε σαν σκοπό να μειώσει την απόσταση που διανύουν τα δεδομένα μεταξύ της μνήμης και της CPU (εικόνα 43β) (Zou, 2021), (Ou, 2020).

Το 1965 ο **Gordon Moore** πρόβλεψε ότι κάθε 2 χρόνια η πυκνότητα των transistors σε ένα μικροεπεξεργαστή (integrated circuit), θα αυξάνεται εκθετικά (Prucnal, 2017), (Eigenmann, 1998). Αυτή η πρόβλεψη έχει επαληθευτεί, με αποτέλεσμα οι ανάγκες για επεξεργασία του εκθετικά αυξανόμενου όγκου δεδομένων, να μην μπορούν να καλυφθούν από το κλασικό μοντέλο von Neumann. Τον νόμο του Moore έρχεται να στηρίξει το 1974 (R. H. Dennard, 1974) η μελέτη του Dennard, γνωστή στην βιβλιογραφία και ως **Dennard** κλιμάκωση, η οποία δηλώνει ότι όσο τα transistor μικραίνουν σε μέγεθος, η πυκνότητα ισχύος τους παραμένει σταθερή. Αυτό σημαίνει, ότι τα μικρότερα transistor χρειάζονται λιγότερο ρεύμα και τάση για να λειτουργήσουν (Prucnal, 2017), (Esmailzadeh, 2011, June), (Streetman, 2006). Όπως συμπέραναν οι *Sarbazi-Azad et al.* στο άρθρο τους (Sarbazi-Azad, 2022), περίπου το 2004 έπαψε να ισχύει η κλιμάκωση Dennard και γι' αυτό τον λόγο δημιουργήθηκαν οι multicore processors, οι οποίοι συντέλεσαν στην μείωση της κατανάλωσης ισχύος. Παρόλα αυτά, δεν αποτέλεσαν λύση στο πρόβλημα της ενεργειακής απόδοσης που απαιτείται για να λειτουργήσουν ταυτόχρονα όλοι οι πυρήνες των παρόντων και μελλοντικών τεχνολογιών. Υποστηρίζεται πως ο νόμος του Moore θα πάψει να ισχύει περίπου ως το 2025 (Schuller, 2015).

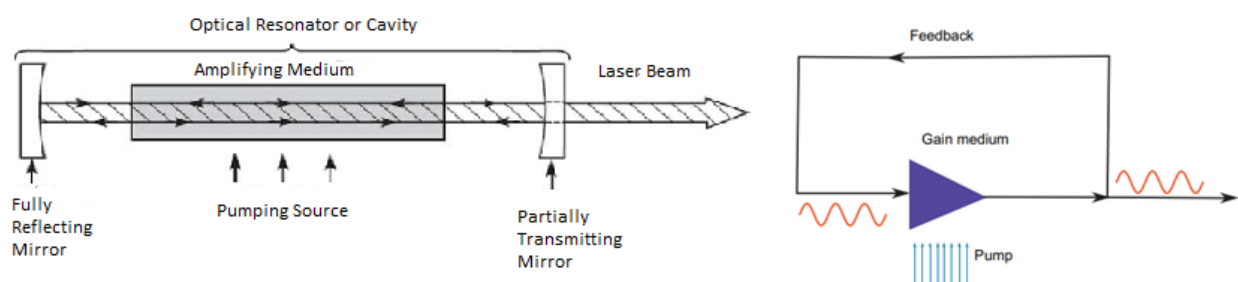


Εικόνα 43 α) Von Neumann μοντέλο β) non-Von Neumann μοντέλο (Ou, 2020)

5.2 Φωτονικά εξαρτήματα και εφαρμογές

Προηγουμένως αναφέρθηκαν οι λόγοι για τους οποίους καθώς εξελίσσονταν η τεχνολογία των υπολογιστών, οι ηλεκτρονικές δομές δεν κάλυπταν ορισμένες προϋποθέσεις. Έτσι, αναπτύχθηκε η φωτονική τεχνολογία η οποία με την βοήθεια ενεργών και παθητικών στοιχείων μπορεί να μεταδώσει σήματα πληροφορίας. Μερικά από τα πλεονεκτήματά της είναι η ενεργοαποδοτικότητα που δίνει στα παθητικά στοιχεία, η μικρή παραγωγή θερμότητας, η μικρή παρεμβολή (crosstalk) και η παράλληλη επεξεργασία (Goι, 2020). Σαν παθητικά στοιχεία ορίζονται αυτά που επηρεάζουν με την οπτική ακτινοβολία που περνάει από μέσα τους, αλλά δεν παράγουν φως (κυματοδηγοί, οπτικές ίνες, ring resonators και διαμορφωτές (modulators)) (De Lima T. F., 2017), (Shastri B. J., 2017). Φωτονική είναι η επιστήμη που περιλαμβάνει τη δημιουργία ενός φωτονίου (φως), καθώς και τον χειρισμό του, μέσω μετάδοσης, εκπομπής, επεξεργασίας σήματος, διαμόρφωσης, μεταγωγής (switching), ενίσχυσης και ανίχνευσης. Οι Amsalu et al αναφέρουν, ότι ο τομέας της φωτονικής βρίσκεται σε αρχικό στάδιο ανάπτυξης και ο όρος «φωτονική» επινοήθηκε κατ' αναλογία με την ηλεκτρονική (Amsalu, 2020). Η **οπτοηλεκτρονική** αναφέρεται σε κυκλώματα ή συσκευές που αποτελούνται τόσο από ηλεκτρικά όσο και από οπτικά εξαρτήματα (LASER ημιαγωγού, φωτοανιχνευτές, διαμορφωτές, φωτονικά ολοκληρωμένα κυκλώματα) που αλληλεπιδρούν (Shastri B. J., 2017). Οπτικά εξαρτήματα όπως οι ανιχνευτές φωτός (πχ. ηλιακής ακτινοβολίας ή φωτοδιόδων), ή τα εξαρτήματα που αναφέρονται παρακάτω χρησιμοποιούν φωτονική τεχνολογία (Amiri, 2018).

Σαν πηγή φωτός για την υλοποίηση οπτικών επικοινωνιών και υπολογισμών χρησιμοποιούνται τα LASER (είτε on-chip είτε off-chip) (Sunny, 2021). Η πρώτη φορά που έγινε λόγος για **LASER's** (Light Amplification by Stimulated Emission of Radiation) ήταν το 1959, από τον Gordon Gould, ενώ το 1962 προτάθηκαν τα LASER ημιαγωγών από τους Basov και Javan (Pankove). Ένα LASER εκμεταλλεύεται την κυματική μορφή των φωτονίων για να κάνει εξαναγκασμένη εκπομπή τους (εικόνα 44β). Τα φωτόνια που παράγονται είναι όλα του ίδιου μήκους κύματος και εστιάζονται σε μια δέσμη φωτός (εικόνα 44α).

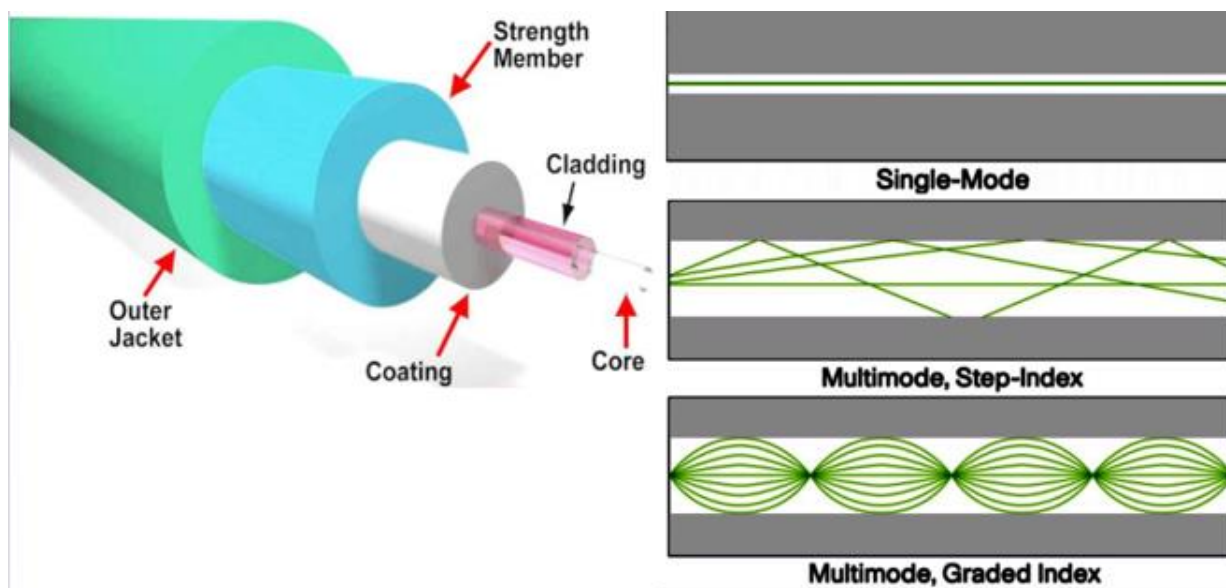


Εικόνα 44 α) Routing in Terrestrial Free Space Optical Ad-Hoc networks β) (Pathak, 2019)

Σύμφωνα με την βιβλιογραφία, (Sunny, 2021), (Shastri B. J., 2017), ένα LASER μπορεί είτε να βρίσκεται off-chip είτε on-chip. Ο τρόπος με τον οποίο ένα LASER διαμορφώνει την πληροφορία σε ένα NN μπορεί να είναι είτε απευθείας το LASER να διαμορφώνει τα δεδομένα σε ένα οπτικό σήμα, είτε η έξοδος λείζερ μπορεί να διαμορφωθεί από έναν modulator που είναι υπεύθυνος για τη διαμόρφωση των δεδομένων στο οπτικό σήμα.

Η οπτική επικοινωνία γίνεται εφικτή λόγω της μετάδοσης φωτός από μία δίοδο Laser ή LED μέσα σε **οπτικές ίνες** (waveguide) (Idachaba, 2014, July), (Agrawal G. P., 2012), (Sunny, 2021). Οι οπτικές ίνες, καθώς από μόνες τους είναι ευαίσθητες και εύθραυστες, περικλείονται από διάφορα στοιχεία με πιο σημαντικά τον μανδύα και τον πυρήνα, όπως φαίνεται και στην (εικόνα 45α). Ανάλογα με την χρήση τους, η οπτική ίνα στον πυρήνα, μπορεί να δέχεται φως με τους παρακάτω (εικόνα 45β) διαφορετικούς τρόπους (Amiri, 2018), (Fermann, 1998) :

- Με ένα μόνο τρόπο για μετάδοση (Single Mode Fibers - SMF). Οι SMF είναι πολύ μικρές σε μέγεθος και απαιτούν τέλεια ευθυγράμμιση της δομής με το laser και αυτό προσθέτει μία δυσκολία κατά την σύνδεσή τους με άλλα φωτονικά στοιχεία,.
- Με πολλούς τρόπους (Multi Mode Fibers - MMF) και κλιμακωτού δείκτη διάθλασης. Οι MMF διαθέτουν μεγαλύτερο πυρήνα ώστε να μπορούν να δέχονται το φως από την δίοδο Laser ή LED με περισσότερες γωνίες. Επίσης, ο δείκτης διάθλασης του πυρήνα και του μανδύα αλλάζει βηματικά, επειδή το υλικό γυαλιού στον πυρήνα και στον μανδύα είναι διαφορετικά.
- Με πολύτροπες βαθμωτού δείκτη διάθλασης. Σε αυτό το είδος, το κέντρο του πυρήνα έχει διαφορετικό δείκτη διάθλασης από τον μανδύα και γι' αυτό παρατηρείται αυτή η βαθμωτή μεταβολή των τρόπων στην οπτική ίνα.

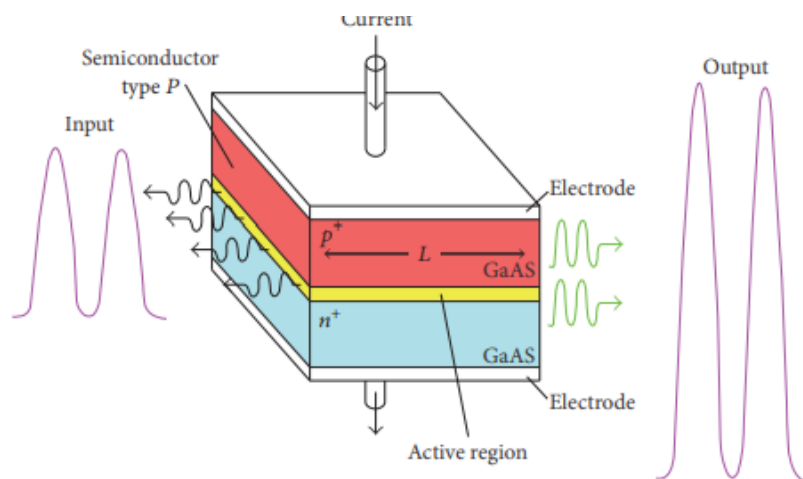


Εικόνα 45 α) οπτική ίνα και περιβλήματά της β) είδη διάδοσης μέσα στην οπτική ίνα (Amiri, 2018).

Οι **φωτοдиодοι** (photodiodes), σύμφωνα με τον (George J. K., 2019), είναι εξαρτήματα που δέχονται οπτικά σήματα, κάνουν την άθροιση τους και στην συνέχεια παράγουν μία τάση (ηλεκτρονικό σήμα). Οι φωτοдиодοι χρειάζονται τους φωτοανιχνευτές (photodetectors), ώστε να γίνει η ανίχνευση των φωτονίων και σύμφωνα με το βιβλίο του (Agrawal G. P., 2010, october), χωρίζονται σε δύο κατηγορίες, φωτοαγωγίμους και φωτοβολταϊκούς. Στο βιβλίο των (Rongqing Hui, 2009), γίνεται μαθηματική ανάλυση της λειτουργίας των φωτοдиодων, αναφέροντας ότι στο συγκεκριμένο εξάρτημα, το

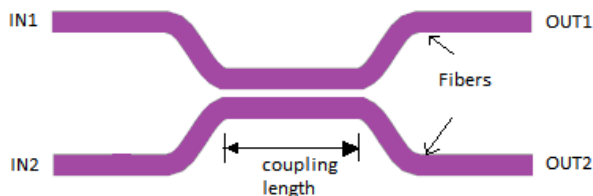
παραγόμενο ρεύμα είναι ανάλογο του τετραγώνου του οπτικού πεδίου $I(t) = \Re|E(t)|^2$, όπου $\Re$ είναι η απόδοση (responsivity) της φωτοδιόδου με μονάδα μέτρησης A/W .

Ο οπτικός **ενισχυτής**, (Ivaniga, 2017), είναι μια συσκευή που σε αντίθεση με τους επαναλήπτες (repeaters), επιτρέπει την απευθείας ενίσχυση ενός οπτικού σήματος, χωρίς να απαιτείται η μετατροπή του σε ηλεκτρικό σήμα. Επί του παρόντος, υπάρχουν ενισχυτές οπτικών ημιαγωγών SOA (semiconductor optical amplifier), ενισχυτές ινών EDFA (erbium doped fiber amplifier) και ενισχυτές στη βάση του φαινομένου Raman. Οι SOA ενισχυτές όπως φαίνεται και από την (εικόνα 46), ενισχύουν το σήμα εισόδου σε όλη την ενεργή περιοχή, λόγω της εκπομπής που διεγείρεται από τον ημιαγωγό. Οι SOA δομούνται και λειτουργούν παρόμοια με τα laser's, με την διαφορά ότι οι ενισχυτές SOA περιέχουν antireflex επίστρωση κατά της δημιουργίας συντονισμού και της συσσώρευσης σήματος μέσα στο μέσο. Οι επιδόσεις των SOA είναι χειρότερες από των EDFA. Οι SOA, (O'Mahony, 1989, October) έχουν περιορισμό στους ρυθμούς με τους οποίους μπορούν να επεξεργαστούν, έχουν πολύ μικρότερο συντελεστή ενίσχυσης, αλλά είναι πιο μικροί σε μέγεθος (ολοκληρωμένα), άρα παρέχουν ευχρηστία. Στις κλασικές επικοινωνίες προτιμώνται οι EDFA καθώς έχουν πολλά θετικά με κυριότερο ότι το μήκος κύματος τους είναι πιο ευρύ από των SOA.



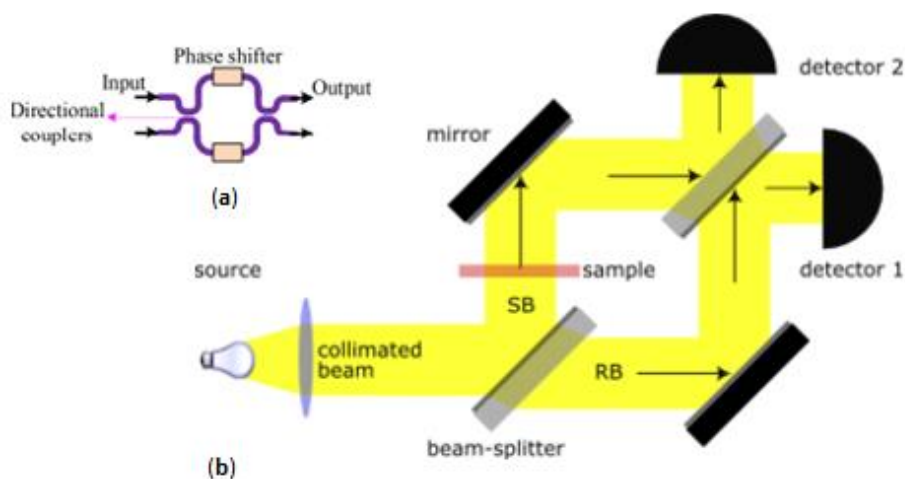
Εικόνα 46. Σχηματική απεικόνιση ενισχυτή SOA

Οι φωτονικοί συζεύκτες (**coupler**), όπως φαίνεται στο σχήμα της (εικόνας 47), χρησιμοποιούνται για τη σύζευξη ενός οπτικού σήματος από μια οπτική ίνα (π.χ. συνδεδεμένη με μια πηγή λέιζερ εκτός τσιπ) σε έναν κυματοδηγό εντός τσιπ, λόγω της σημαντικής αναντιστοιχίας μεταξύ της διατομής του οπτικών ινών (δεκάδες μικρά) και αυτή των φωτονικών κυματοδηγών πυριτίου (εκατοντάδες νανόμετρα). Η συσκευή (couplers) είναι ικανές να συνδυάζουν δύο ή περισσότερες εισόδους σε μια ενιαία έξοδο και επίσης να διαιρούν μια μεμονωμένη είσοδο σε δύο ή περισσότερες εξόδους (Sunny, 2021), (Pathak, 2019).



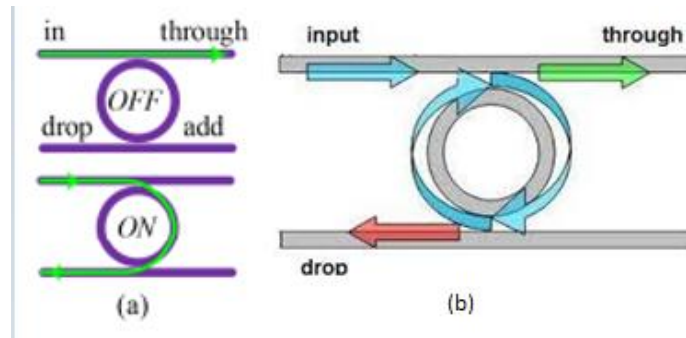
Εικόνα 47. Σχηματική απεικόνιση ενός Coupler οπτικού σήματος

Το **ιντερφερόμετρο** είναι μια οπτική δομή που χρησιμοποιεί το φαινόμενο της παρεμβολής. Είναι γνωστό ότι σε ένα σήμα, την στιγμή που διαδίδεται, μπορούν να προκύπτουν διάφορες παρεμβολές με αποτέλεσμα να υπάρχει απώλεια πληροφορίας. Υπάρχουν πολλά διαφορετικά είδη ιντερφερόμετρων με πιο γνωστό το Mach-Zehnder (MZI) που πήρε την ονομασία από τους δημιουργούς του Ludwig Mach και Ludwig Zehnder. Ένας MZI βρόγχος μπορεί να λειτουργήσει ως modulator, switch, insulator, (de)-multiplexer κ.λπ. (Zetie, 2000) και χρησιμοποιεί το φαινόμενο της παρεμβολής προκειμένου να υλοποιήσει πολλαπλασιασμό μαθηματικών πινάκων για αμιγώς παθητικές μονάδες φωτονικών NN (Guo X. X., 2021). Όπως φαίνεται και από την (εικόνα 48 α,β), ένα MZI είναι μια βασική συσκευή παρεμβολής κυματοδηγού. Αποτελείται από δύο couplers (ή κλάδους Y), που ο ένας λειτουργεί σαν διαχωριστής (splitter) και ο άλλος σαν συνδυαστής (combiner). Με την σειρά τους αυτοί συνδέονται με δύο κυματοδηγούς διαφορετικού μήκους (Yuan P. Li, 1997). Ο splitter χωρίζει το φως εισόδου σε δύο συμμετρικούς/ασύμμετρους κυματοδηγούς και οι εξοδοί αυτών των δύο κυματοδηγών συνδυάζονται από έναν combiner. Η κατανομή της οπτικής ισχύος στις δύο εξόδους εξαρτάται από την διαφορά στα μήκη του οπτικού “βραχίονα” και από το μήκος κύματος (ή την οπτική συχνότητα) (Zetie, 2000). Εάν χρησιμοποιούνται δύο συμμετρικοί κυματοδηγοί, το MZI είναι γνωστό ως ισορροπημένο MZI. (Pathak, 2019).



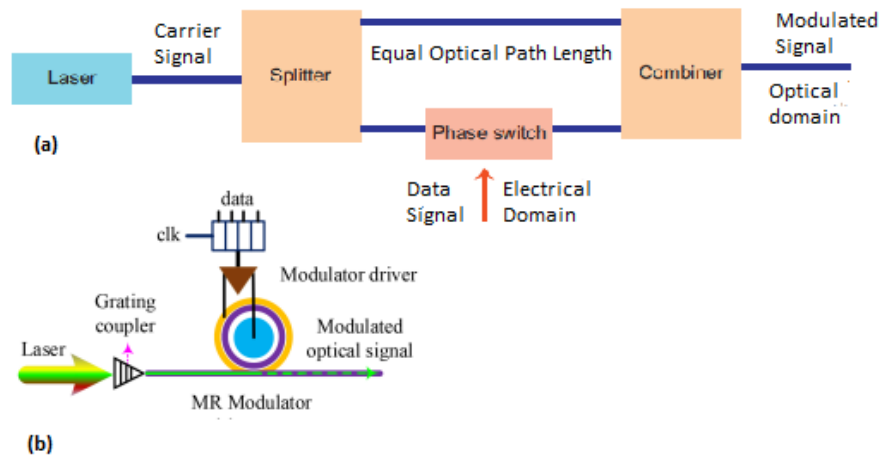
Εικόνα 48. Σχηματική απεικόνιση MZI α) (Sunny, 2021) β) (Pathak, 2019)

Ο συντονιστής δακτυλίου (**ring resonator**) είναι βασικό στοιχείο στα ολοκληρωμένα φωτονικά κυκλώματα καθώς βρίσκει πολλές εφαρμογές σαν, οπτικό φίλτρο, φίλτρο add/drop (Bazian, 2021), οπτικός αισθητήρας φωτός (Pathak, 2019) κλπ. Ένας ring resonator (εικόνα 49β) αποτελείται από έναν βρόχο κυματοδηγού που ονομάζεται δακτύλιος και έναν ή δύο κυματοδηγούς διαύλου για πρόσβαση στον δακτύλιο (Pathak, 2019). Όπως φαίνεται στο σχήμα της (εικόνας 49α), ένας μικροδακτύλιος (MRR) μπορεί να άγει ένα φέρον σήμα στις πύλες THROUGH και DROP, με on ή off συντονισμό αντίστοιχα. Με αυτό τον τρόπο ρυθμίζεται αναλόγως και το βάρος της σύνδεσης σαν θετικό ή αρνητικό (Guo X. X., 2021) (Tait A. N., 2016). Βασικές παράμετροι για τον ορισμό ενός MR είναι η ακτίνα του δακτυλίου, μήκος κύματος συντονισμού, διατομή του MR, και πόλωση του φωτός (Zhang, 2020), (El Srouji, 2022).



Εικόνα 49. α) Σχηματική απεικόνιση MRR δομής σαν ADD/ DROP filter β) Σχηματική απεικόνιση MRR (Sunny, 2021)

Ο οπτικός διαμορφωτής (**optical modulator**) είναι μια συσκευή που χρησιμοποιείται για τη διαμόρφωση μιας δέσμης φωτός που διαδίδεται είτε σε ελεύθερο χώρο, είτε σε έναν οπτικό κυματοδηγό. Αυτή η συσκευή μπορεί να αλλάξει διαφορετικές παραμέτρους της δέσμης και επομένως ο οπτικός διαμορφωτής μπορεί να κατηγοριοποιηθεί σαν διαμορφωτής πλάτους, φάσης, πόλωσης, ηλεκτροδιάθλασης και ηλεκτροαπορρόφησης (Reed, 2010). Επίσης, σύμφωνα με τον (Pathak, 2019), ένας οπτικός διαμορφωτής μπορεί να οριστεί ως μια οπτική συσκευή για τη μετατροπή των δεδομένων/πληροφοριών από ηλεκτρικό τομέα σε οπτικό τομέα με χειρισμό των οπτικών ιδιοτήτων. Ο MZI, εικόνα (εικόνα 50α) και ο ring resonator, (εικόνα 50β), χρησιμοποιούνται συχνά για την διαμόρφωση φωτός. Ο MZI Modulator (MZM) δεν είναι ευαίσθητος στη θερμοκρασία και παρέχει μεγάλο εύρος ζώνης διαμόρφωσης. Ο ring resonator μπορεί να παρέχει χαμηλότερη κατανάλωση ενέργειας και πολύ υψηλά εύρη ζώνης, όμως είναι πιο ευαίσθητος στην θερμοκρασία σε σχέση με τους MZM και λόγω αυτού επηρεάζεται η ευθυγράμμισή τους με τα μήκη κύματος του laser (Margalit, 2021).



Εικόνα 50 α) Mach- Zehnder διαμορφωτής (Pathak, 2019) β) Microring διαμορφωτής (Sunny, 2021)

Στο άρθρο (Sunny, 2021) γίνεται αναφορά και άλλων οπτοηλεκτρονικών εξαρτημάτων μαζί με την χρήση τους σε νευρομορφικά φωτονικά συστήματα.

Το κύριο πλεονέκτημα των φωτονικών συστημάτων σε σχέση με τα ψηφιακά ηλεκτρονικά είναι, ότι τα διακριτά σήματα μπορούν να συνδυαστούν αξιοποιώντας αποδοτικούς ηλεκτροοπτικούς (EO) modulators, phase shifters, συνδυαστές (combiners) και απλοποιώντας βασικές λειτουργίες, όπως πρόσθεση, πολλαπλασιασμό διανυσματικών πινάκων ή συνελίξεις (convolutions). Επιπλέον, τα φωτονικά είναι σε θέση να λειτουργήσουν με υψηλή παραλληλία, εκμεταλλευόμενα την τεχνική wavelength-division multiplexing (WDM), όπου διαφορετικά μήκη κύματος φωτός μπορούν να συνδυαστούν, να μεταδοθούν στο ίδιο κανάλι και να διαχωριστούν ξανά. Η χρήση πολυπλεξίας στις οπτικές επικοινωνίες, αντισταθμίζει το σχετικά μεγάλο μέγεθος των φωτονικών εξαρτημάτων, συγκριτικά με τα ηλεκτρονικά transistor (De Lima T. F., 2020). Τέλος, τα φωτονικά συστήματα μπορούν να διαμορφώνουν την κυματοδηγούμενη πληροφορία αλλάζοντας τη φάση, το πλάτος ή την πόλωση του σήματος (Amin, 2019).

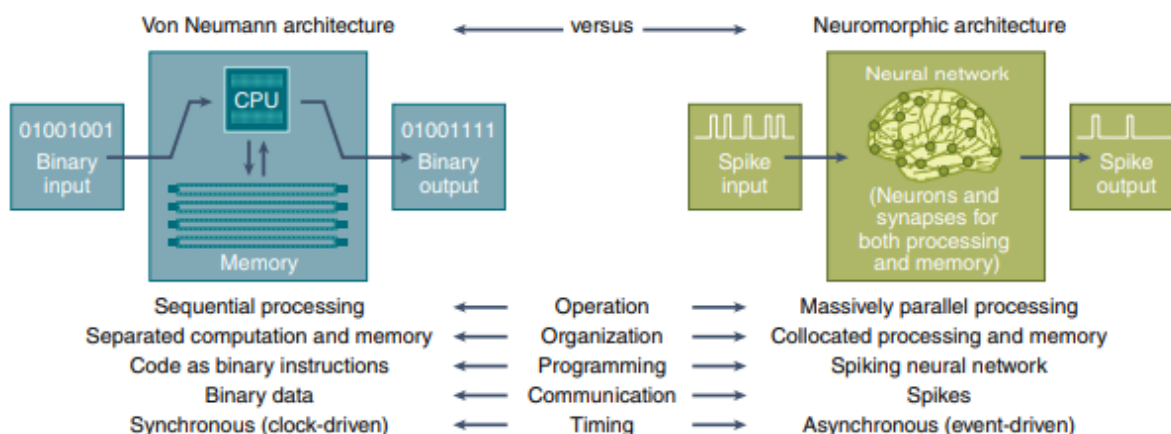
Το φως έχει καθιερωθεί ως το μέσο επικοινωνίας των τηλεπικοινωνιών και των datacenters, όμως δεν είναι ακόμα ευρεία η χρήση του στην επεξεργασία πληροφοριών και στους υπολογιστές (Shastri B. J., 2021). Τα υπάρχοντα data centers που συντονίζουν την παγκόσμια ροή δεδομένων από συστήματα οπτικών επικοινωνιών απαιτούν τεράστιες ποσότητες ηλεκτρικής ενέργειας. Για το σκοπό αυτό, οι φωτονικές ολοκληρωμένες τοπολογίες που διαθέτουν μόνο παθητικά φωτονικά στοιχεία, όπως οπτικοί διαχωριστές (splitters), interferometers και συντονιστές microring, μαζί με πλήρως οπτικό σχεδιασμό, γίνονται πιο ενεργειακά αποδοτικές (Argyris, 2022), (De Lima T. F., 2017).

Ο τομέας της φωτονικής ενισχύει την ποιότητα ζωής, δημιουργεί θέσεις εργασίας και ωθεί την οικονομική ανάπτυξη, καθώς έχει ευρύ φάσμα επιστημονικών και τεχνολογικών εφαρμογών. Οι τομείς αυτοί είναι: η ιατρική θεραπεία και διάγνωση, οι βιομηχανικές κατασκευές με χρήση laser, οι χημικοί και βιολογικοί αισθητήρες, η εξερεύνηση του διαστήματος και η αεροπορική βιομηχανία, οι στρατιωτικές επιχειρήσεις και η πυρηνική ενέργεια με σύντηξη. Επίσης βρίσκει εφαρμογή σε τηλεπικοινωνίες οπτικών ινών, στο νευρομορφικό φωτονικό computing, στην ναυτοφωτονική τεχνολογία, στην εκτύπωση πλακετών, στην τεχνολογία απεικόνισης και σε οπτικούς υπολογιστές. (Amiri, 2018).

5.3 Νευρομορφική Τεχνολογία

Για την επίλυση του προβλήματος μειωμένης αποδοτικότητας που παρουσιάζει η αρχιτεκτονική von Neumann, είναι σε εξέλιξη η νευρομορφική τεχνολογία, που ονομάστηκε έτσι από τον Carver Meads στα τέλη της δεκαετίας του '80 (Schuller, 2015), (Schuman C. D., 2022). Σκοπός της τεχνολογίας αυτής, είναι η ανάπτυξη του **Neuromorphic Computing (NC)** προκειμένου να μιμηθεί την λειτουργία του εγκεφάλου με την βοήθεια hardware δομών εμπνευσμένων ή από τα βιολογικά NN, ή τα artificial NN ή τις non-von-Neumann αρχιτεκτονικές (Schuman C. D., 2017), (Islam, 2019). Βασικός στόχος του neuromorphic engineering ή neuromorphic computing είναι η χρήση των αλγορίθμων μηχανικής μάθησης και της τεχνητής νοημοσύνης για υλοποίηση hardware (Shastri B. J., 2021). Το πρώτο βήμα στην δημιουργία NC είναι η κατανόηση των λειτουργιών και της δομής ενός νευρωνικού συστήματος (Goi, 2020).

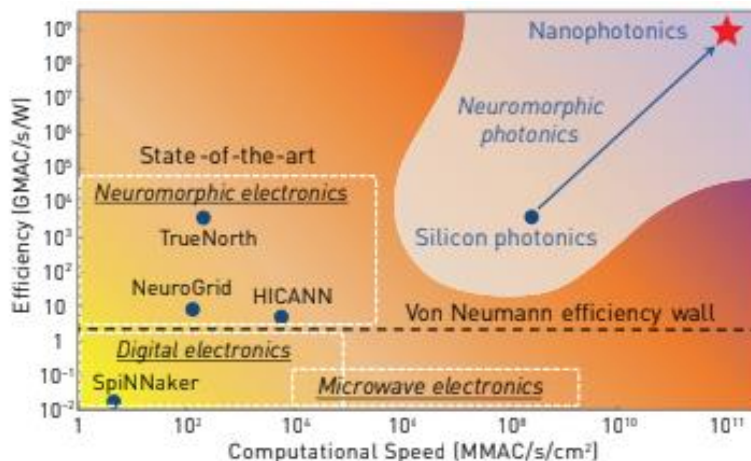
Στην (εικόνα 51) εμφανίζεται μία σύγκριση των von-Neumann με τα νευρομορφικά μοντέλα. Μια διαφορά των νευρομορφικών computer από τα von-Neumann είναι, ότι τα προγράμματα των πρώτων ορίζονται από την δομή του NN και τις παραμέτρους, ενώ στα δεύτερα, βασίζονται στις "οδηγίες" που δίνει ο αλγόριθμος (Sunny, 2021). Επιπλέον, τα von-Neumann κωδικοποιούν την πληροφορία σε δυαδικό σύστημα ενώ τα νευρομορφικά δέχονται spikes σαν εισόδους και προκειμένου να κωδικοποιήσουν την αριθμητική πληροφορία που δέχθηκαν, χρησιμοποιούν στοιχεία των spikes, όπως την χρονική στιγμή που συνέβη, το πλάτος και το σχήμα τους (Schuman C. D., 2022).



Εικόνα 51. Σύγκριση Von Neumann με Νευρομορφική αρχιτεκτονική επεξεργασίας (Schuman C. D., 2022).

Προκειμένου να βελτιωθούν οι περιορισμοί που δημιουργούνται από τη χρήση ηλεκτρονικών στοιχείων, ερευνήθηκε η αντικατάστασή τους με φωτονικά. Οι *De Marinis et al.* στο άρθρο (De Marinis, 2019) αναφέρουν ότι το πυρίτιο (silicon), παρουσιάζει εξαιρετικές ιδιότητες και αποτελεί άριστο υλικό για την δημιουργία chips με ενσωματωμένη φωτονική τεχνολογία. Η ανάπτυξη του **silicon photonics** βασισμένη στην ύπαρξη CMOS τεχνολογίας επέφερε την ενσωμάτωση οπτικών στοιχείων με μειωμένο κόστος. Επίσης βοήθησε στην χρήση φωτονικών συνδέσεων (photonic links) για την μετάδοση πληροφορίας με ταχύτητα φωτός, αντικαθιστώντας τις μεταλλικές συνδέσεις (Pasricha, 2020), (Sunny, 2021), (De Marinis, 2019). Στο άρθρο (Shastri B. J., 2021), γίνεται αναφορά στα silicon photonics ως ένας από τους παράγοντες που βοήθησαν στην ανάπτυξη των φωτονικών NN, επειδή (Huang, 2022) παρέχουν μια εξαιρετική πλατφόρμα για την μαζική παραγωγή οπτικών συστημάτων σε χαμηλό κόστος. Οι silicon photonics πλατφόρμες (chip) μπορούν να "φιλοξενήσουν" παθητικά (passive) εξαρτήματα υψηλής ποιότητας σε συνδυασμό με υψηλής ταχύτητας ενεργά (active) οπτοηλεκτρονικά εξαρτήματα (Shastri B. J., 2021), (De Lima T. F., 2017).

Στην παρακάτω (εικόνα 52) εμφανίζονται μερικές από τις hardware υλοποιήσεις με νευρομορφική τεχνολογία, όπως το chip **TrueNorth** της IBM που δημιουργήθηκε το 2014 (Rose, 2021), (Goi, 2020). Το TrueNorth χρησιμοποιεί silicon τεχνολογία και είναι μία πλατφόρμα χαμηλής ισχύος, που εκτελεί NN αλγορίθμους για αναγνώριση λόγου και εικόνas (Akopyan, 2015) .



Εικόνα 52 . εξέλιξη αρχιτεκτονικών για υπολογιστικά συστήματα (Prucnal, 2017)

Τα νευρομορφικά συστήματα, (εικόνα 52), είναι πιο αποδοτικά ως προς την ταχύτητα και την κατανάλωση ενέργειας σε σχέση με τα ηλεκτρονικά (digital electronics) (De Marinis, 2019). Απώτερος στόχος, σύμφωνα με την (εικόνα 52), είναι να υπάρξουν συστήματα που να λειτουργούν με νανοφωτονική τεχνολογία, όπου αναμένονται αποδόσεις ισχύος της τάξης του 10^9 GMAC/s/W και υπολογιστική ταχύτητα 10^{11} MMAC/s/cm².

Το **Photonic Neuromorphic Computing (PNC)**, σχεδόν πάντα χρειάζεται ηλεκτρονικά στοιχεία για κατανομή ισχύος, για έλεγχο και σηματοδότηση (signaling) (El Srouji, 2022). Στα πρώτα βήματα του NC, χρησιμοποιήθηκαν φωτονικές υλοποιήσεις οι οποίες είναι εγγενώς παράλληλες (inherently parallel) ως προς το μήκος κύματος, τον χώρο και τον χρόνο (σε αντίθεση με την αρχιτεκτονική von Neumann που είναι σειριακή) όπως συμβαίνει στον εγκέφαλο, ενώ σημειώθηκε δυσκολία στην υλοποίηση αποθηκευτικού χώρου σε οπτικά συστήματα. Σε επόμενη φάση του NC, αναπτύχθηκαν φωτονικές υλοποιήσεις και πλατφόρμες, επειδή τα φωτονικά συστήματα μπορούν να λειτουργήσουν ταχύτατα σε προβλήματα με σχετικά μέτρια πολυπλοκότητα και δυνατότητα προγραμματισμού (Schuman C. D., 2017), (El Srouji, 2022). Σε πολλές υλοποιήσεις NC, οι διεργασίες της επεξεργασίας πληροφορίας και της αποθήκευσης της σε μνήμη συνυπάρχουν στους νευρώνες και στις συνάψεις τους, σε αντίθεση με το κλασικό μοντέλο von-Neuman (Schuman C. D., 2022) η. Για αποθήκευση της πληροφορίας σε μνήμη, σύμφωνα με τα άρθρα (Goi, 2020), (El Srouji, 2022) γίνεται χρήση memristors που προσομοιάζουν τις βιολογικές συνάψεις του εγκεφάλου. Τα συναπτικά δίκτυα memristors αποτελούνται από φωτονικούς συζεύκτες και φωτονικούς ή οπτοηλεκτρονικούς νευρώνες που παρέχουν μη γραμμικότητα.

5.4 Εξέλιξη των φωτονικών NN

Στη δεκαετία του 1980, (Psaltis, 1988), (Wagner, 1993), (El Srouji, 2022), οι μελέτες οπτικών νευρωνικών δικτύων έγιναν ένας πολύ ενεργός τομέας μελέτης για την επίτευξη μαζικά παράλληλων υπολογισμών με την ταχύτητα του φωτός. Όπως αναφέρεται στην προαναφερθείσα βιβλιογραφία και στο άρθρο (El Srouji, 2022), (Huang, 2022), ο ίδιος ο πρωτοπόρος, ο **Psaltis**, δήλωσε τη δεκαετία του 1990 ότι εγκαταλείπει τον οπτικό NC για δύο λόγους: (1) έλλειψη πρακτικών συσκευών που μπορούν να ενσωματωθούν και (2) ανεπαρκής γνώση περίπλοκων νευρωνικών δικτύων. Τρεις δεκαετίες αργότερα (El Srouji, 2022), οι αλγόριθμοι μηχανικής μάθησης που χρησιμοποιούν Deep NN (DNN) έχουν προχωρήσει. Ο ρυθμός αύξησης των ολοκληρωμένων στα silicon photonics είναι τώρα δύο φορές ταχύτερος από των ηλεκτρονικών και είναι εμφανής η επιβράδυνση της τάσης για συνεχείς αυξήσεις στην πυκνότητα των τρανζίστορ (νόμος του Moore). Το λογικό συμπέρασμα μετά από τις τρεις μεγάλες αλλαγές παραπέμπει στο Photonic Neuromorphic Computing (PNC) ως τη βασική λύση για μελλοντικούς υπολογιστές. Ο βασικός στόχος του PNC, (De Lima T. F., 2019), είναι η φυσική (hardware-physical) μίμηση ενός βιολογικού NN σε πραγματικό χρόνο όσο το δυνατόν πιο αποτελεσματικά και με μεγάλη ταχύτητα επεξεργασίας. Επίσης, παρέχει έναν τρόπο επιτάχυνσης των αλγορίθμων εκπαίδευσης σε NN, βοηθώντας στη διαδικασία της εκπαίδευσης, η οποία απαιτεί πολύ πιο προηγμένη υλοποίηση και υπολογιστική ισχύ.

5.4.1 Φωτονικός Νευρώνας

Τα PNC αποτελούνται από **φωτονικούς νευρώνες**, οι οποίοι κατ' αντιστοιχία με την απλή δομή του κλασικού Single Layer Perceptron (SLP), αποτελούν την βασική δομή ενός φωτονικού δικτύου προσομοιάζοντας έναν βιολογικό νευρώνα. Ένας silicon φωτονικός νευρώνας μπορεί να κατηγοριοποιηθεί, ανάλογα αν η υλοποίησή του γίνεται με εξ ολοκλήρου οπτικό ή οπτοηλεκτρονικό τόπο και αν ο νευρώνας είναι coherent ή non-coherent (Sunny, 2021), (Guo X. X., 2021), (El Srouji, 2022), (Prucnal, 2017), (De Lima T. F., 2017). Ένας φωτονικός νευρώνας λαμβάνει πολλαπλά οπτικά σήματα εισόδου (fan-in), τα επεξεργάζεται και τελικά προωθεί το οπτικό σήμα εξόδου σε άλλους νευρώνες (fan-out) (Amin, 2019). Κατά τη διάρκεια της επεξεργασίας, οι λειτουργίες που εκτελούνται είναι η εισαγωγή βαρών στα σήματα εισόδου (μαθηματικά εκφράζεται ως γινόμενο διανυσμάτων), το άθροισμα γινομένων και η εισαγωγή μη-γραμμικότητας με μία συνάρτηση ενεργοποίησης. Η προαναφερθείσα επεξεργασία σημάτων στον φωτονικό νευρώνα μπορεί να υλοποιηθεί, είτε εξ' ολοκλήρου οπτικά (O/O) είτε οπτοηλεκτρονικά (O/E/O). Επίσης, γίνεται ανανέωση των βαρών κατά την διάρκεια της εκπαίδευσης του NN (Sunny, 2021), (Jha, 2020), (De Lima T. F., 2020), (Shastri B. J., 2017). Στο άρθρο (Sunny, 2021), παρουσιάζεται συνοπτικός πίνακας με τις παραπάνω νευρωνικές λειτουργίες και τον τρόπο υλοποίησής τους με φωτονικές συσκευές πυριτίου, καθώς επίσης δίνεται πίνακας, που αποτυπώνει πως αυτές επηρεάζονται από το είδος υλοποίησης (ηλεκτρονικό ή φωτονικό) σε ένα νευρωνικό δίκτυο.

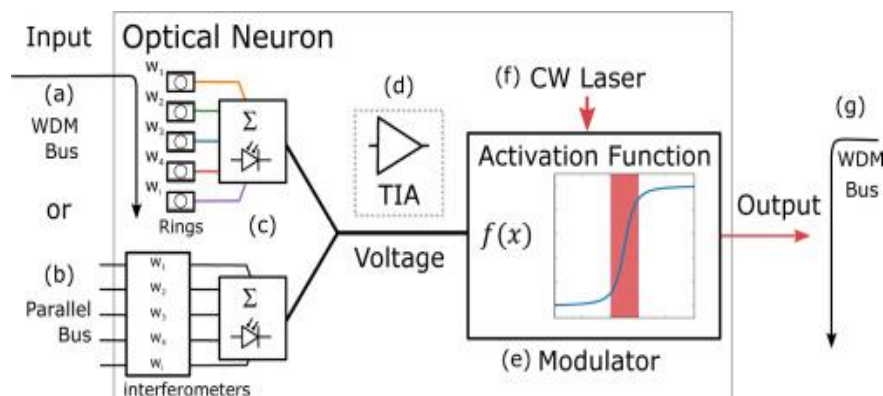
Στον *πίνακα 7*, φαίνονται τα τρία βασικά στοιχεία της δομής ενός νευροσυναπτικού συστήματος (νευρώνας, σύναψη, άξονας και δενδρίτης), οι λειτουργίες τους και βιβλιογραφικές αναφορές σχετικές με εξαρτήματα που υλοποιούν τις ίδιες λειτουργίες σε φωτονικό νευρώνα (Guo X. X., 2021).

Components	Functions	Photonic devices
Neuron	Summation and thresholding (nonlinear node)	Lasers, ^[47-51] optical amplifiers, ^[45] SAs, ^[102] optical bistable devices, ^[103] electric-optic modulators, ^[43] photodetectors ^[100]
Synapse	Weighting and memory (coding scheme)	MZIs, ^[42] ring resonators, ^[78] PCMs ^[89]
Axon and dendrite	Interconnection (network)	Waveguides

Πίνακας 7 Περίληψη των βασικών συστατικών των PNN (Guo X. X., 2021)

5.4.2 Οπτοηλεκτρονικός Νευρώνας

Στον Ο/Ε/Ο σχεδιασμό υπάρχει η ανάγκη για μετατροπή του κύματος από οπτικό σε ηλεκτρικό και στην συνέχεια από ηλεκτρικό ξανά σε οπτικό. Σε αυτή την κατηγορία νευρώνων το βασικό μειονέκτημα είναι, ότι η μετατροπή από οπτικό σε ηλεκτρικό και αντίστροφα προσθέτει απώλεια ισχύος κατά την μετάδοση της πληροφορίας (Sunny, 2021). Το οπτικό σήμα (σταθμισμένα μήκη κύματος) εισάγεται σε μια φωτοδίοδο (balanced) και εξέρχεται ρεύμα (ηλεκτρονικό σήμα) που αντιπροσωπεύει την συνολική οπτική ισχύ του. Το σήμα αυτό είναι σε θέση να διαμορφώσει μια συσκευή laser (Tait A. N., 2014). Στην συνέχεια, με την βοήθεια Ε/Ο μετατροπέα, ο οποίος προσθέτει μη- γραμμικότητα στο σύστημα, το σήμα εξέρχεται σαν οπτικό (El Srouji, 2022). Το ρόλο του Ε/Ο μετατροπέα μπορεί να τον παίξει ένα laser που τοποθετείται είτε εντός του chip, οπότε υπάρχει πιθανό πρόβλημα αύξησης θερμοκρασίας, είτε εκτός του chip (George J. K., 2019).

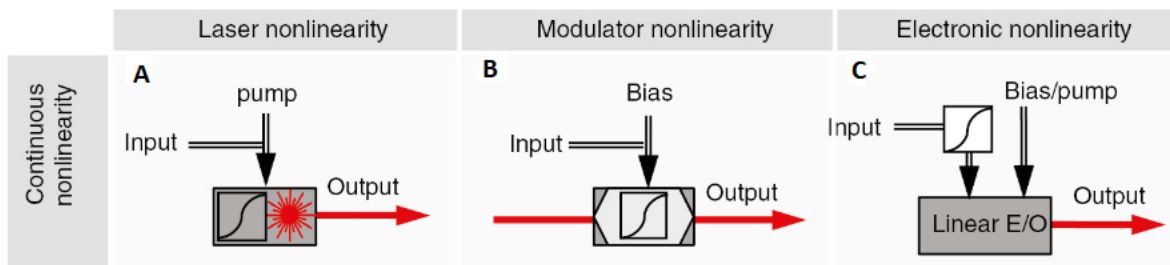


Εικόνα 53 Νευρώνας με MRR και MZI και με LASER modulator (George J. K., 2019)

Πιο συγκεκριμένα, στο σχήμα (εικόνα 53) εμφανίζεται ένας Ο/Ε/Ο νευρώνας ο οποίος δέχεται σαν είσοδο διάφορα μήκη κύματος από έναν δίαυλο WDM και θέτει βάρη με MRR (53α). Στη συνέχεια, ο νευρώνας **αθροίζει** τα σταθμισμένα οπτικά σήματα με μια φωτοδίοδο (μπορεί να είναι και τύπου Balanced) που μετατρέπει το σήμα σε τάση (53c) και προαιρετικά το ενισχύει (53d). Αυτός ο τρόπος

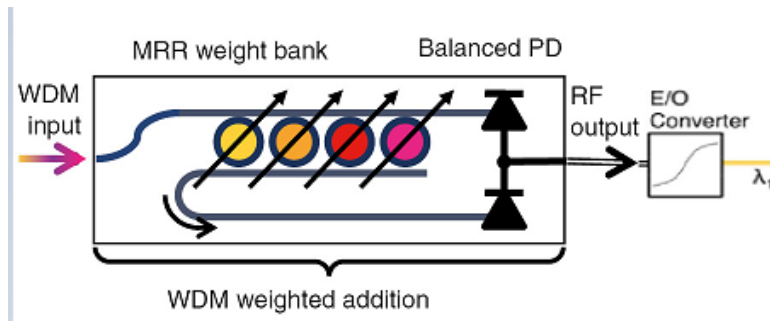
υλοποίησης αποτελεί το non-coherent τμήμα του νευρώνα (Shastri B. J., 2017). Στην ίδια εικόνα υπάρχει και είσοδος σημάτων από έναν παράλληλο δίαυλο που θέτει βάρη με ένα δίκτυο MZI (53b) τα οποία και πάλι **αθροίζονται** με μια φωτοδίοδο και αυτός ο τρόπος υλοποίησης του νευρώνα χαρακτηρίζεται ως coherent (Guo X. X., 2021). Το τμήμα του νευρώνα (MRR-φωτοδίοδος, MZI- φωτοδίοδος) που εκτελεί τον **πολλαπλασιασμό** και την πρόσθεση (MAC), αποτελεί το **γραμμικό** μέρος (εικόνα 53c) (Liu, 2021).

Στην συνέχεια το ηλεκτρικό σήμα εισέρχεται στο **μη-γραμμικό** τμήμα του νευρώνα όπου υπάρχει ένας ηλεκτροοπτικός διαμορφωτής (53e) που διαμορφώνει ένα Continuous Wave (CW) λέιζερ (53f), το οποίο παράγει μια μη γραμμική συνάρτηση μεταφοράς πριν το σήμα βγει σαν έξοδος (53g) ενός φωτονικού νευρωνικού δικτύου (NN) (Liu, 2021). Οι φυσικές υλοποιήσεις αυτών των μη- γραμμικοτήτων μπορούν να προκύψουν από συσκευές είτε καθαρά ηλεκτρονικές, είτε ηλεκτροοπτικούς διαμορφωτές, είτε Laser. Η *εικόνα 54α* περιγράφει ένα λέιζερ με συνεχή μη γραμμικότητα, η *54β* ένα διαμορφωτή με συνεχή μη γραμμικότητα και η *54c* μια αμιγώς ηλεκτρονική μη γραμμικότητα με οπτική έξοδο.



Εικόνα 54. Κατηγοριοποίηση O/E/O PNN με συνεχείς μη-γραμμικότητες και πιθανές υλοποιήσεις. Προσαρμοσμένο σχήμα από (De Lima T. F., 2017) (Shastri B. J., 2017)

Στην βιβλιογραφία αναφέρονται (Huang, 2022), (De Lima T. F., 2019) προσπάθειες για χρήση εναλλακτικών τύπων modulators, όπως στο άρθρο (Tait A. N., 2019), όπου οι *Tait et al* χρησιμοποιούν MRR με ενσωματωμένο modulator αντί για laser. Στην *εικόνα 55* (De Lima T. F., 2017), (Sunny, 2021) τα διερχόμενα μήκη κύματος μέσα από έναν κυματοδηγό εισέρχονται σε ένα δίκτυο MRR για να λάβουν αντίστοιχα βάρη από ένα MRR weight bank. Τα σταθμισμένα σήματα εισόδου, προστίθενται σε ένα balanced photodetector και παράγεται ένα ηλεκτρικό σήμα, το οποίο με την σειρά του μετατρέπεται σε οπτικό περνώντας μέσα από έναν διαμορφωτή.



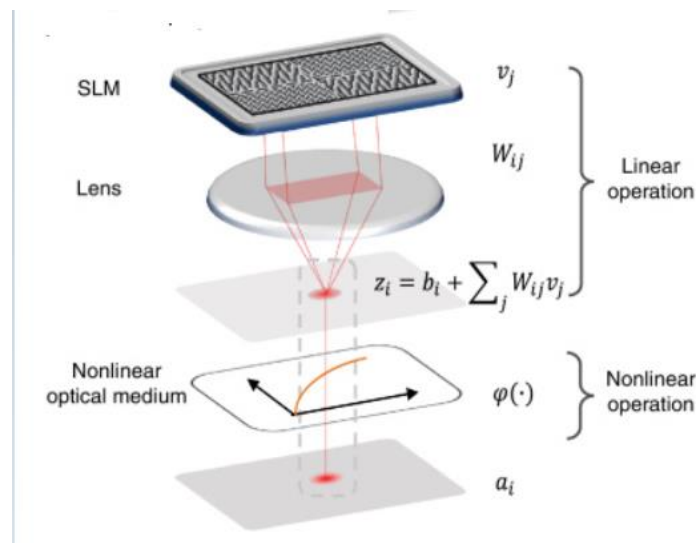
Εικόνα 55. νευρώνας με MRR weight bank και με ενσωματωμένο modulator (Sunny, 2021) (De Lima T. F., 2017)

5.4.3 Εξολοκλήρου Οπτικός Νευρώνας

Στα Ο/Ο όλες οι λειτουργίες εκτελούνται οπτικά και όλα τα εξαρτήματα του νευρώνα υποστηρίζουν την μεταφορά οπτικών σημάτων. Αυτό συνεπάγεται μεγαλύτερες ταχύτητες, καθώς δεν υπάρχει ενδιάμεση μετατροπή (όπως στα Ο/Ε/Ο) και η απώλεια πληροφορίας είναι μικρότερη. Οι πλήρως οπτικές προσεγγίσεις δεν υποστηρίζουν WDM, επομένως δεν μπορούν να παρέχουν υψηλή διασυνδεσιμότητα για την υλοποίηση κλιμακούμενων φωτονικών νευρωνικών δικτύων. Στην περίπτωση που ο νευρώνας είναι εξολοκλήρου οπτικός, η λειτουργία του **αθροίσματος** μπορεί να επιτευχθεί με DFB lasers, VCSELs, MRs και laser ημιαγωγού βασισμένα σε integrated saturable absorber (Sunny, 2021). Ωστόσο, η εφαρμογή **μη-γραμμικότητας** στους Ο/Ο νευρώνες είναι δύσκολη, επειδή απαιτείται σχετικά μεγάλη οπτική ισχύς (De Marinis, 2019). Στα (El Srouji, 2022), (De Marinis, 2019) αναφέρεται η χρήση SOA και vertical cavity surface emitting laser (VCSEL) και MRR για την εισαγωγή μη γραμμικότητας στο σύστημα.

Ένας από τους κύριους περιορισμούς είναι ότι η διαθέσιμη οπτική μη γραμμικότητα είναι σχετικά ασθενής. Κατά συνέπεια, απαιτούνται μεγάλα μήκη αλληλεπίδρασης κυματοδηγών και υψηλή οπτική ισχύς για την επίτευξη ισχυρών μη γραμμικοτήτων. Ένας άλλος βασικός περιορισμός είναι ότι η οπτική μη γραμμικότητα τείνει να διορθώνεται μετά την κατασκευή της συσκευής, αποτρέποντας τον επαναπρογραμματισμό των ANN με ευελιξία για διαφορετικές εργασίες εκμάθησης (Guo X. X., 2021).

Στο (Zuo, 2019) και στην *εικόνα 56*, παρουσιάζεται μια γραφική απεικόνιση ενός οπτικού νευρώνα (AONN) χωρίς κυματοδηγούς και ενσωματωμένα κυκλώματα. Η γραμμική λειτουργία $z_i = b_i + \sum_j W_{ij}v_j$ προγραμματίζεται συνδυάζοντας διαμορφωτές φωτός χώρου (SLM) και φακούς Fourier. Στο στάδιο αυτό η ισχύς του προσπίπτοντος φωτός σε διαφορετικές περιοχές του SLM αντιπροσωπεύει διαφορετικούς κόμβους εισόδου. Η μη γραμμική λειτουργία πραγματοποιείται με laser-cooled άτομα που έχουν ηλεκτρομαγνητική επαγόμενη διαφάνεια. Επειδή όλα τα σφάλματα από διαφορετικούς οπτικούς νευρώνες είναι ανεξάρτητα, είναι δυνατό να κλιμακωθεί το μέγεθος ενός τέτοιου AON (Demertzis, 2022), (Liu, 2021).



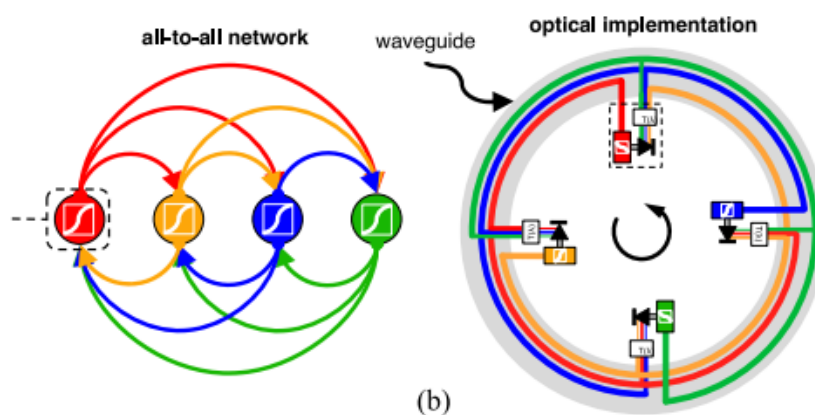
Εικόνα 56. Υλοποίηση οπτικού νευρώνα με γραμμική λειτουργία (SLM και φακούς) και μη γραμμική λειτουργία (συνάρτηση ενεργοποίησης $\varphi(\cdot)$) (Demertzis, 2022).

5.5 Φωτονικά Νευρωνικά Δίκτυα

Οι φωτονικοί νευρώνες συνδέονται με κυματοδηγούς και σχηματίζουν ένα **φωτονικό NN** (De Lima T. F., 2019). Ένα φωτονικό NN υψηλής απόδοσης, αναμένεται να διαθέτει σε αυξημένο βαθμό αξιοπιστία, scalability και cascadability. Η αξιοπιστία βελτιώνεται όταν το Signal to Noise Ratio (SNR) στην έξοδο του νευρώνα μειώνεται ή αυξάνοντας την ισχύ στο επιθυμητό οπτικό σήμα ώστε να φέρει μόνο την χρήσιμη πληροφορία χωρίς τον θόρυβο (Sunny, 2021). Ένας scalable νευρώνας υποστηρίζει επαρκή αριθμό εισόδων ώστε να είναι σε θέση να ενεργοποιήσει δίκτυα μεγάλης κλίμακας (Sunny, 2021), (Tait A. N., 2014). Το cascadability είναι η ιδιότητα ενός νευρώνα να άγει άλλους νευρώνες βασιζόμενος στην ισχύ του οπτικού σήματος (Shastri B. J., 2017).

5.5.1 Συνδέσεις σε ένα NN

Οι συνδέσεις μεταξύ των νευρώνων γίνονται με την τεχνική του WDM, όπου όλες οι είσοδοι σε κάθε νευρωνικό επίπεδο (layer) βρίσκονται σε έναν μόνο κυματοδηγό χωρισμένο κατά μήκος κύματος (De Lima T. F., 2020). Οι κυματοδηγοί σε ένα φωτονικό NN, (Huang, 2022), μπορούν να πετύχουν μεγάλη ταχύτητα μετάδοσης δεδομένων, με χαμηλή εξασθένηση σήματος. Επίσης, είναι εφικτή η πολυπλεξία του μήκους κύματος των φωτονίων, καθώς δεν επηρεάζονται από τις αλλαγές στην συχνότητα του σήματος και δεν αλληλεπιδρούν με άλλα φωτόνια. Οι νευρωνικές συνδέσεις είναι παθητικές (De Lima T. F., 2019) και χωρίς switches (δεν χρειάζονται ισχύ για να μεταφέρουν την πληροφορία μεταξύ των νευρώνων), χωρίς clocks (η πληροφορία “ρέει” από το ένα επίπεδο στο επόμενο με ταχύτητα φωτός) και υπάρχει επεκτασιμότητα - scalability (τα δεδομένα που μετακινούνται έχουν σταθερό κόστος λόγω των χαμηλών απωλειών που υπάρχουν στον κυματοδηγό). Στο άρθρο (De Lima T. F., 2020), γίνεται μια σύγκριση συνδέσεων σε ένα κλασικό NN σύστημα (εικόνα 62α) και σε ένα φωτονικό NN (εικόνα 57β). Η κυριότερη διαφορά του είναι ότι στην περίπτωση των οπτικών NN όλα τα σήματα διαφορετικών μηκών κύματος δρομολογούνται μέσα σε έναν κυματοδηγό ταυτόχρονα. Το σχήμα της εικόνας 57β αποτελεί τροποποίηση της αρχικής ιδέας των Tait et al. (Tait A. N., 2014).



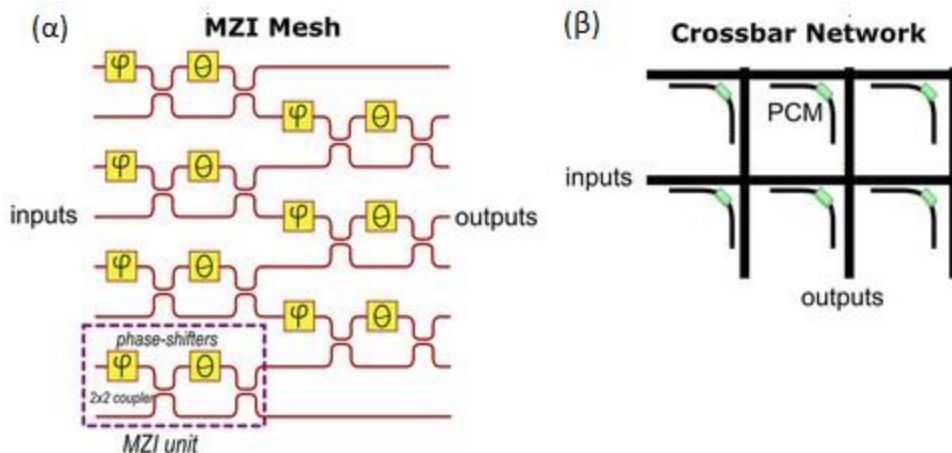
Εικόνα 57 (α)Αρχιτεκτονική NN που όλοι οι κόμβοι συνδέονται με κάθε άλλο κόμβο (β)Αρχιτεκτονική Φωτονικών NN με πολλά μήκη κύματος σε έναν κυματοδηγό (De Lima T. F., 2020). (Tait A. N., 2014)

5.5.2 Αρχιτεκτονικές PNN και τρόποι στάθμισης

Οι υλοποιήσεις φωτονικών NN διαθέτουν **αρχιτεκτονική** που διέπεται από τα βασικά οπτικά εξαρτήματα καθώς και από τις θεμελιώδεις αρχές, όπως optical resonance, optical interference, οπτική περίθλαση, στις οποίες στηρίζονται τα εξαρτήματα. Έτσι, υπάρχει η αρχιτεκτονική που εκμεταλλεύεται το οπτικό resonance και κάνει χρήση των MR, το οπτικό interference χρησιμοποιώντας τα MZI, χρησιμοποιεί περιθλαστικά στοιχεία (on-chip) για νευρωνικά δίκτυα χαμηλής πυκνότητας νευρώνων (Demertzis, 2022) ή χρησιμοποιεί SOA (Sunny, 2021). Στη συνέχεια θα γίνει περιγραφή του τρόπου που οι φωτονικοί νευρώνες συνδέονται μεταξύ τους για να προκύψει ένα φωτονικό NN και τον τρόπο που γίνεται η στάθμιση με τα παραπάνω οπτικά εξαρτήματα.

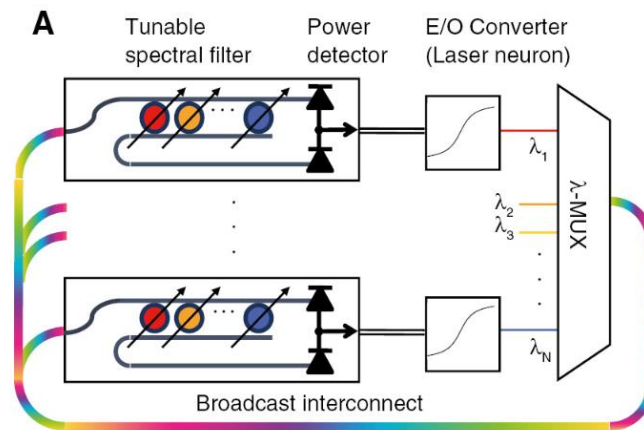
Στην *εικόνα 58* αποτυπώνονται δύο από τους τρόπους με τους οποίους πραγματοποιείται **στάθμιση** (πολλαπλασιασμός βαρών με τα μήκη κύματος) στον φωτονικό νευρώνα, χρησιμοποιώντας αντίστοιχα οπτικά εξαρτήματα, MR και MZI (El Srouji, 2022). Η πράξη του πολλαπλασιασμού, μπορεί να γίνει επίσης με περιθλαστικά στοιχεία και με **Crossbar Networks** (*εικόνα 58β*) (Demertzis, 2022), (El Srouji, 2022). Σε ένα crossbar network τα εισερχόμενα σήματα ευθυγραμμίζονται κατά την μια κατεύθυνση ενώ τα εξερχόμενα κατά την άλλη (κάθετα μεταξύ τους). Χρησιμοποιούν αναδιαμορφώσιμα υλικά, όπως τα PCM ή τα οπτικά υλικά μεμβράνης, που επιτρέπουν στο εισερχόμενο φως να συζευχθεί στους κυματοδηγούς εξόδου σύμφωνα με τη συναπτική ισχύ (El Srouji, 2022).

Τα **MZI** λόγω της ικανότητάς τους να αναπαριστούν αποτελεσματικά πίνακες για λειτουργίες νευρωνικών δικτύων, έχουν χρησιμοποιηθεί ευρέως σε φωτονικά NN. Συνήθως, αυτά τα συστήματα βασίζονται σε χειρισμό της φάσης και του πλάτους ηλεκτρικού πεδίου ενός μόνο οπτικού μήκους κύματος (Sunny, 2021). Με αυτό τον τρόπο πραγματοποιείται ο φωτονικός πολλαπλασιασμός πινάκων όπου τα βάρη ρυθμίζονται με έλεγχο της φάσης και του πλάτους των οπτικών σημάτων, τοποθετώντας attenuators και phase shifters στους βραχίονες MZI (El Srouji, 2022). Στην *εικόνα 58α* φαίνεται ένα δίκτυο MZI (MZI mesh) που αποτελείται από πολλές οπτικές γραμμικές μονάδες (Optical Linear Units - OLU) και έτσι σχηματίζεται ένας NxN πίνακας, όπου υλοποιείται πολλαπλασιασμός πινάκων. Τα MZI σε σύγκριση με τα MR απαιτούν περισσότερο χώρο για την υλοποίηση μη γραμμικών συνδέσεων (Sunny, 2021).



Εικόνα 58 α) MZI αρχιτεκτονική στάθμισης NN β) Crossbar Network αρχιτεκτονική στάθμισης NN (El Srouji, 2022)

Ο φωτονικός νευρώνας της εικόνας 55 μπορεί να συνδεθεί με άλλους παρόμοιους νευρώνες και να δημιουργηθεί ένα φωτονικό NN on chip με την βοήθεια του πρωτοκόλλου Broadcast & Weight (B&W) (Tait A. N., 2016). Στην εικόνα 59 μια σειρά από λέιζερ εξάγει διαφορετικά μήκη κύματος (που αντιπροσωπεύονται από συμπαγές χρώμα). Αυτά τα κανάλια πολυπλέκονται σε μήκος κύματος (WDM) σε έναν μόνο κυματοδηγό (πολύχρωμα). Το tunable spectral filter (ή αλλιώς weight bank), (De Lima T. F., 2017) αποτελείται από MRs και βρίσκεται στην είσοδο κάθε νευρώνα όπου πραγματοποιεί ανεξάρτητες συναρτήσεις στάθμισης. Τα weight banks περιλαμβάνουν **tunable rings**, τα οποία μπορούν να συντονίζονται με την ενσωμάτωση phase shifter, ώστε να “τραβάνε” ενέργεια από το μήκος κύματος που ρυθμίζουν, με απώτερο στόχο τα φιλτραρισμένα wavelengths να αντικατοπτρίζουν το αποτέλεσμα του πολλαπλασιασμού $w \times x$ (El Srouji, 2022). Η αποπολυπλεξία δεν πραγματοποιείται στο δίκτυο. Αντίθετα, ανιχνεύεται η συνολική οπτική ισχύς κάθε φασματικά σταθμισμένου σήματος, αποδίδοντας το άθροισμα των καναλιών εισόδου σαν ηλεκτρονικό σήμα. Το ηλεκτρονικό σήμα μετατρέπεται σε οπτικό σήμα μετά από μη γραμμικό μετασχηματισμό με την βοήθεια Laser.



Εικόνα 59. MRR αρχιτεκτονική δόμηση ενός φωτονικού νευρωνικού δικτύου (De Lima T. F., 2019), (Tait A. N., 2016), (De Lima T. F., 2017), (Tait A. N., 2014)

Αντίστοιχα με τα κλασικά NN, έτσι και στα οπτικά υπάρχουν διάφορες υλοποιήσεις σύνδεσης κόμβων φωτονικών NN όπως FFNN, CNN, SNN και RC. Σε επόμενη ενότητα θα γίνει εκτενής αναφορά στα φωτονικά RC. Στα άρθρα (De Marinis, 2019), (Demertzis, 2022), οι έως τώρα υλοποιήσεις φωτονικών NN, κατηγοριοποιούνται ανάλογα αν διαθέτουν ή δεν διαθέτουν μνήμη (stateless και stateful) καθώς επίσης ανάλογα με τον σχεδιασμό τους και τον τρόπο εκπαίδευσής τους (trainable ή inference only). Οι *De Marinis et al*, στο άρθρο (De Marinis, 2019), κατηγοριοποιούν υλοποιήσεις που έχουν γίνει σε οπτικά συστήματα και παρουσιάζουν σε πίνακες τον τρόπο υλοποίησης του φωτονικού hardware, τα αναφερθέντα αποτελέσματα και εφαρμογές καθώς και τον τρόπο εκπαίδευσης για κάθε μια από αυτές.

5.6 Φωτονικό Reservoir Computing

Το reservoir computing, βασισμένο στα βιολογικά συστήματα, είναι μια πολύ αποδοτική προσέγγιση επεξεργασίας χρονικά εξαρτώμενων δεδομένων. Η βασική δομή ενός RC, που αναλύθηκε σε προηγούμενο κεφάλαιο, αποτελείται από ένα μη-γραμμικό αναδρομικό δυναμικό σύστημα συνδεδεμένο με ένα επίπεδο εισόδου και ένα επίπεδο εξόδου στο οποίο γίνεται η εκπαίδευση (Paquiot, 2012). Το ίδιο το reservoir δεν χρειάζεται εκπαίδευση και έτσι καθίσταται κατάλληλο για εφαρμογή του σε φωτονικά συστήματα, αφού απαιτεί ελάχιστη χρήση hardware εξαρτημάτων (Van der Sande, 2017). Το γεγονός αυτό παρουσιάζει πλεονεκτήματα, όσον αφορά την ταχύτητα και την απόδοση ισχύος, λόγω του μεγάλου bandwidth και των γρήγορων μη γραμμικών επιδράσεων μιας φωτονικής υλοποίησης (Vandoorne K. D., 2008).

5.6.1 Διαφορές RC φωτονικού με ηλεκτρονικό

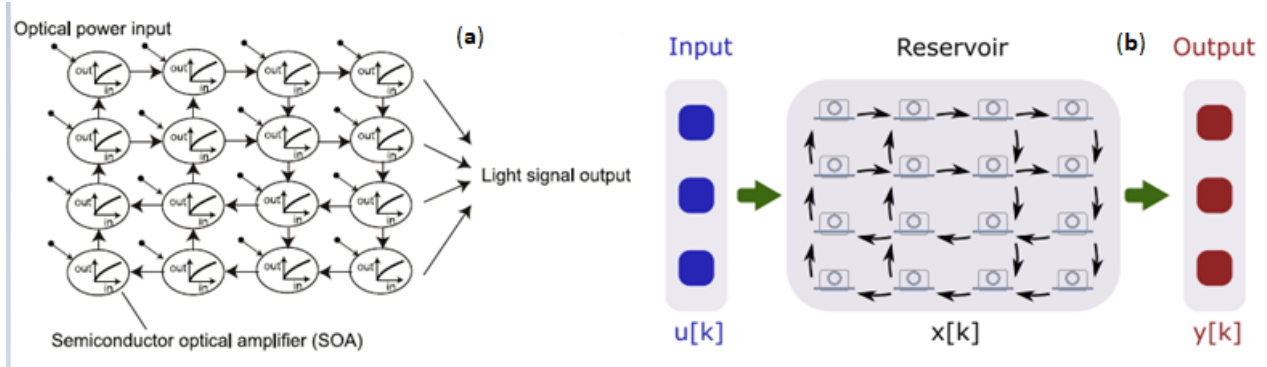
Στην βιβλιογραφία αναγράφονται οι διαφορές μεταξύ του ηλεκτρονικού και του φωτονικού RC. Τα φωτονικά είναι πιο αποτελεσματικά στην επίλυση πολύπλοκων προβλημάτων (όπως η αναγνώριση φωνής, η πρόβλεψη χαοτικών χρονοσειρών, η κατηγοριοποίηση σημάτων και η μοντελοποίηση δυναμικών συστημάτων) σε σχέση με τα κλασικά RC που βασίζονται σε software. Ένα ακόμα πλεονέκτημα παρουσιάζεται από τον Vandoorne *et al* στο άρθρο (Vandoorne K. M., 2014) το 2014, και υποστηρίζει ότι η ταυτόχρονη εκμετάλλευση τόσο της φάσης όσο και του πλάτους του φωτός προσδίδει αξιοσημείωτη βελτίωση σε σχέση με τα δίκτυα RC που βασίζονται σε software. Αυτό συμβαίνει γιατί ένα reservoir που λειτουργεί σε μιγαδικούς αριθμούς στην ουσία διπλασιάζει τους εσωτερικούς βαθμούς ελευθερίας στο σύστημα, οδηγώντας σε ένα μέγεθος reservoir που είναι περίπου διπλάσιο από την ίδια συσκευή που λειτουργεί με μη-συνεκτικό φως (Shastri B. J., 2017) .

5.6.2 Τοπολογίες και Υλοποιήσεις

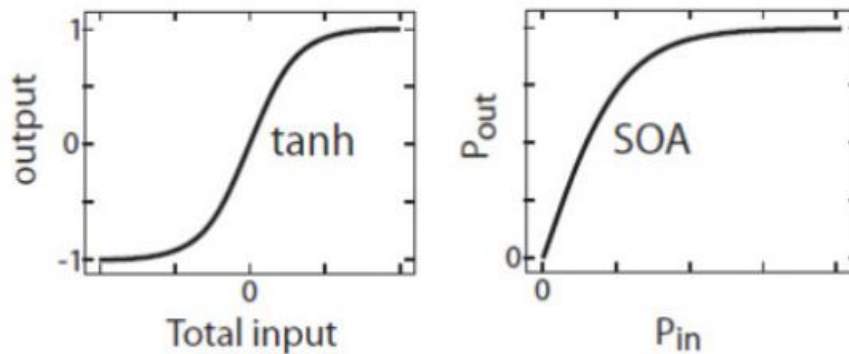
Έχουν παρουσιαστεί διάφορες τοπολογίες για φωτονικά reservoir, με χρήση οπτικών ινών (DLF) με έναν μοναδικό μη-γραμμικό δυναμικό κόμβο (Appeltant, 2011), καθώς και ενσωματωμένες λύσεις με MRR, ένα δίκτυο με SOAs και παθητικά φωτονικά chip πυριτίου (Shastri B. J., 2017), (De Lima T. F., 2020).

Οι Vandoorne *et al*, το 2008, στο (Vandoorne K. D., 2008), για πρώτη φορά προτείνουν το **Photonic Reservoir Computing (PRC)** ως ένα νέο “εργαλείο” για την αντιμετώπιση προβλημάτων classification και regression, το οποίο βασίζεται σε μια φωτονική υλοποίηση της προταθείσας υπολογιστικής ιδέας του reservoir (Jaeger H. &, 2004). Πιο συγκεκριμένα, πρότειναν μια spatial φωτονική υλοποίηση με διασυνδεδεμένα SOA την οποία και υλοποίησαν το 2014 (Vandoorne K. M., 2014). Στο άρθρο (Tanaka, 2019), αναφέρεται η παραπάνω τοπολογία swirl με SOA, όπου κάθε ένας SOA κόμβος συνδέεται με άλλους 4. Όπως φαίνεται στην *εικόνα 60α* η συσκευή χρησιμοποιεί μονότροπους κυματοδηγούς (Single Mode Fiber (SMF) που σχηματίζουν έναν 4x4 πίνακα. Το σήμα πληροφορίας μπαίνει σαν είσοδος σε έναν μόνο κόμβο, ενώ το training του reservoir και η επεξεργασία της εξόδου υλοποιούνται offline. Στο άρθρο (Vandoorne K. D., 2008) , οι Vandoorne *et al*, εξηγούν ότι ο λόγος που προτιμήθηκε η χρήση των SOA για την υλοποίηση ενός reservoir, είναι διότι έχουν πιο πλούσια εσωτερική δυναμική συμπεριφορά σε αντίθεση με τους νευρώνες RC που λειτουργούν με tanh. Σε δεδομένα που εμφανίζονται συχνά, αυτή η δυναμικότητα είναι χρήσιμη, λόγω της αλληλεπίδρασης των φωτονίων με τα σήματα που φέρουν την πληροφορία. Επιπλέον, οι γραφικές παραστάσεις των tanh και SOA είναι παρόμοιες (*εικόνα 61*).

Μια μεταγενέστερη υλοποίηση οπτοηλεκτρονικού reservoir έγινε από τους *Mesaritakis et al*, στο άρθρο (Mesaritakis, 2013). Περιλαμβάνει την χρήση MRR κόμβων (4x4 πίνακας) για την παραγωγή της μη γραμμικότητας στο reservoir. Τα MRR μπορούν να παρέχουν μη γραμμικές αποκρίσεις εξαρτώμενες από την ισχύ, αν ληφθεί υπόψιν η επίδραση της απορρόφησης δύο φωτονίων (TPA) και η μη γραμμική διακύμανση του δείκτη διάθλασης. Στην *εικόνα 60b*, απεικονίζεται τροποποίηση αυτής της αρχικής υλοποίησης στο άρθρο (Denis-Le Coarer, 2018).



Εικόνα 60. Swirl Τοπολογία 4x4 PNN (α) με χρήση SOA (Tanaka, 2019) (b) με χρήση MRR Adapted by (Denis-Le Coarer, 2018)



Εικόνα 61. Καμπύλες tanh και SOA (Vandoorne K. D., 2008)

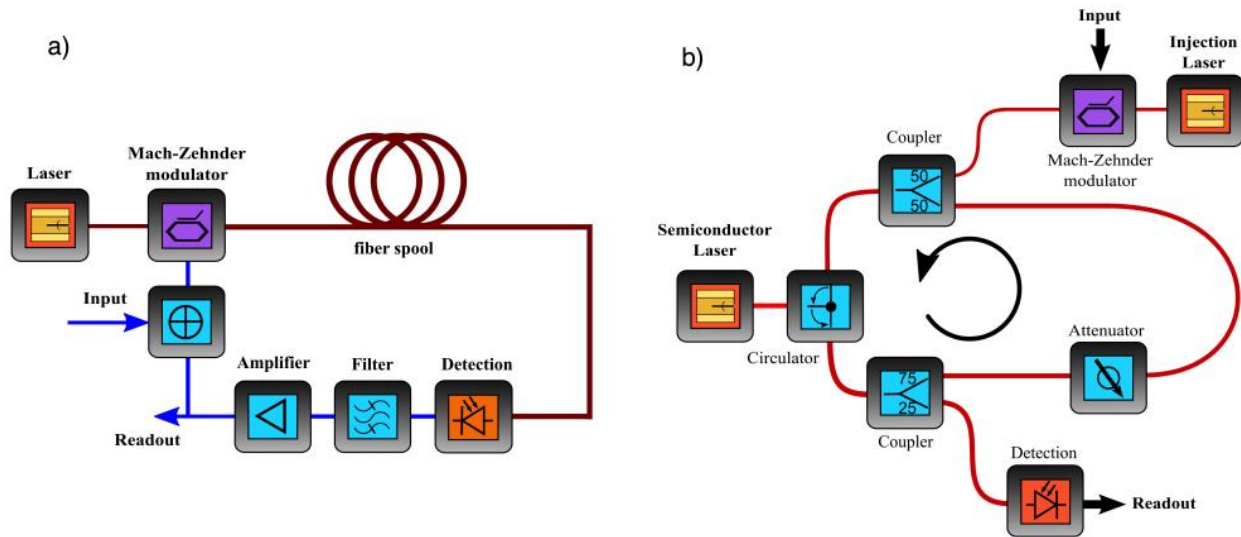
Οι έρευνες που γίνονται πάνω στον τομέα της ανάπτυξης ενός οπτικού/οπτοηλεκτρονικού RC συστήματος είναι συνεχείς και έτσι έχουν προταθεί πολλές υλοποιήσεις ως τώρα. Αν και γίνονται προσπάθειες ώστε και τα τρία επίπεδα RC να υλοποιηθούν με αναλογικό hardware, μέχρι στιγμής η έρευνα επικεντρώνεται στο επίπεδο του reservoir (Van der Sande, 2017).

RESERVOIR LAYER

Στην *εικόνα 62a* και *62b* παρουσιάζεται μια οπτοηλεκτρονική και μια εξολοκλήρου οπτική time delay υλοποίηση αντίστοιχα του Reservoir επιπέδου όπως αναλύεται στα άρθρα (Van der Sande, 2017). Σε ένα RC, οπτικό ή οπτοηλεκτρονικό, τα input και output επίπεδα προσομοιώνονται offline σε έναν υπολογιστή. Η οπτοηλεκτρονική υλοποίηση του reservoir layer στο σχήμα *62a* (Brunner D. L., 2022) περιλαμβάνει το οπτικό μέρος που αποτελείται από οπτικές ίνες, μια πηγή Laser, έναν MZI διαμορφωτή (σαν κόμβος του reservoir) και μια DLF ίνα που προσθέτει την καθυστέρηση. Το ηλεκτρονικό μέρος

διαθέτει μια φωτοδίοδο (για οπτοηλεκτρονική μετατροπή), ένα ηλεκτρονικό φίλτρο και έναν ενισχυτή. Στην συνέχεια το σήμα βγαίνει από το readout και στέλνεται στο output επίπεδο όπου γίνεται η εκπαίδευση.

Η εξολοκλήρου οπτική υλοποίηση που παρουσιάζεται στο σχήμα 62β (Brunner D. L., 2022), αποτελείται από ένα laser ημιαγωγού (που παίζει τον ρόλο του reservoir κόμβου), ένα laser για να εισάγεται η οπτική πληροφορία, ένα MZI διαμορφωτή, έναν οπτικό εξασθενιτή (optical attenuator), έναν circulator (που επιτρέπει στο φως να ταξιδεύει σε μία μόνο κατεύθυνση), συζεύκτες και έναν φωτοανιχνευτή για ανίχνευση του σήματος.



Εικόνα 62. (α) Οπτοηλεκτρονικό reservoir (β) Εξολοκλήρου οπτικό reservoir. Οι κόκκινες γραμμές αντιπροσωπεύουν το οπτικό κομμάτι και οι μπλε το ηλεκτρονικό (Brunner D. L., 2022)

Όπως και σε μία κλασική ηλεκτρονική reservoir υλοποίηση, έτσι και στην περίπτωση του φωτονικού μόνο τα βάρη του output layer είναι αυτά που εκπαιδεύονται. Υπάρχουν πολλές υλοποιήσεις για την δημιουργία ενός output επιπέδου. Στην βιβλιογραφία αναφέρεται (Brunner D. L., 2022), (Smerieri, 2012), ότι η εκπαίδευση γίνεται offline σε έναν υπολογιστή με τεχνικές όπως linear regression, καταγράφοντας τις τιμές των βαρών W_{out} του readout layer. Στο άρθρο των Brunner et al. παρουσιάζεται μία δομή για το output layer η οποία περιλαμβάνει: έναν διπλό MZI διαμορφωτή διπλής εξόδου, ο οποίος δέχεται τα δεδομένα του readout επιπέδου, έναν φωτοανιχνευτή και ένα RLC φίλτρο. Το φίλτρο φέρει το άθροισμα των σταθμισμένων αναλογικών τιμών του reservoir και βγάζει την έξοδο του συστήματος y_{out} .

6 ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΔΡΑΣΕΙΣ

Στην παρούσα διπλωματική εργασία, αρχικά έγινε μια θεωρητική προσέγγιση ενός νευρωνικού δικτύου (NN). Στη συνέχεια μελετήθηκε με την βοήθεια software, η συμπεριφορά των reservoir computing NN που χρησιμοποιούνται για την πρόβλεψη χρονοσειρών. Υλοποιήθηκαν διαφορετικές συναρτήσεις ενεργοποίησης και χρησιμοποιήθηκαν προκειμένου να λυθεί με αποδοτικό τρόπο το πρόβλημα της χρονοσειράς NARMA10. Επιπλέον έγινε σύγκριση της NARMA10 με την NARMA5 με χρήση της συνάρτησης ενεργοποίησης tanh. Όπως φάνηκε, από τα διαγράμματα στα κεφάλαια 3 και 4, αλλά και από τον συγκεντρωτικό πίνακα 6, το σύστημα κατά την εκπαίδευσή του με tanh και sigmoid παρουσιάζει καλύτερες τιμές ($\text{testNRMSE} < 0.1$), καθώς το σφάλμα test NRMSE στις βέλτιστες τιμές κόμβων και $\rho(W)$ είναι ελάχιστο και προσεγγίζει το μηδέν. Η ReLu συνάρτηση ενεργοποίησης κατά την εκτέλεσή της για τιμές κόμβων και $\rho(W)$ δεν έδωσε καλά αποτελέσματα, αφού το σφάλμα που προέκυψε ήταν πολύ μεγαλύτερο από 0.1. Έγιναν δειγματοληπτικές εκτελέσεις του προγράμματος για αριθμό κόμβων μεγαλύτερο από [1600-3000] και προέκυψε ότι η τιμή του σφάλματος βελτιώνεται με best $\text{TestNRMSE} = 0.2285$ για 3000 κόμβους και $\rho(W) = 1$. Συγκριτικά με τις tanh και sigmoid, η ReLu υπολογιστικά είναι πιο γρήγορη, αλλά δίνει καλές τιμές σφάλματος με μεγαλύτερο αριθμό κόμβων. Λόγω αυτού, έγινε η υπόθεση ότι με μια μετατόπιση της καμπύλης ReLu (*Shifted ReLu*) στον άξονα x, θα προέκυπταν βελτιωμένες τιμές σφάλματος NRMSE. Πειραματικά αυτό επιβεβαιώθηκε και φάνηκε ότι με μικρότερο αριθμό κόμβων (1600), επιτυγχάνεται παρόμοια επίδοση συγκριτικά με την ReLu και 3000 κόμβους.

Επίσης εξετάστηκε και η περίπτωση που σαν activation function χρησιμοποιείται μία πολυωνυμική συνάρτηση ($a * x^2$). Παρατηρήθηκε ότι αν προστεθεί μία ενίσχυση $\alpha > 1$ στο σύστημα, τα αποτελέσματα για το test NRMSE είναι πολύ καλά. Πιο συγκεκριμένα, όπως φάνηκε και στον πίνακα 5, παρατηρείται πως για $\alpha = 6$, αριθμό κόμβων = 1600 και $\rho(W) = 0.5$, το $\text{testNRMSE} = 0.0352$. Με ένα συμβατικό software RC, όπως το esnToolbox, είναι δύσκολη η επιτυχής εκτέλεση με μεγαλύτερο αριθμό κόμβων καθώς και με μεγαλύτερο αριθμό $\rho(W)$.

Στην συνέχεια, πραγματοποιήθηκε μια θεωρητική προσέγγιση των νευρομορφικών δικτύων με έμφαση στα φωτονικά NN και φωτονικά reservoir computing. Παρουσιάστηκαν τα εξαρτήματα που χρησιμοποιούνται για την υλοποίηση ενός φωτονικού NN και οι τρόποι με τους οποίους μπορεί να εκτελεσθεί το γραμμικό και το μη-γραμμικό μέρος υπολογισμών. Αφού αναλύθηκαν τα φωτονικά NN παρουσιάστηκαν διάφορες τεχνικές υλοποίησης ενός φωτονικού RC. Σε ένα φωτονικό RC κατά την φάση του training τα δεδομένα του readout επιπέδου λαμβάνονται από μία φωτοδίοδο. Όπως αναφέρθηκε στην ενότητα 5.2, μια φωτοδίοδος λαμβάνει το οπτικό σήμα από το νευρομορφικό δίκτυο και το μετατρέπει σε ψηφιακό, ώστε να περαστεί στον υπολογιστή και να γίνει εκπαίδευση. Ο τρόπος με τον οποίο η φωτοδίοδος πετυχαίνει αυτή την μη-γραμμική μετατροπή είναι υψώνοντας στο τετράγωνο το ηλεκτρικό πεδίο του σήματος $I(t) = \Re|E(t)|^2$, όπου $\Re$ είναι η απόδοση (responsivity) της φωτοδίοδου με μονάδα μέτρησης A/W (Rongqing Hui, 2009). Αυτός ο τρόπος υπολογισμού που γίνεται στην φωτοδίοδο, θυμίζει την μορφή της πολυωνυμικής συνάρτησης ενεργοποίησης $a * x^2$, η οποία με μία ενίσχυση της τάξης $\alpha = [4, 6]$ πετυχαίνει βέλτιστο σφάλμα NRMSE κατά την εκπαίδευση και είναι πολύ εύκολη στην υλοποίησή της. Έτσι, χρησιμοποιώντας μία απλή φωτοδίοδο με αυτά τα χαρακτηριστικά θα μπορούσε να προκύψει μια πολύ καλή μη-γραμμικότητα με πολύ καλά αποτελέσματα.

Από την συγκεκριμένη μελέτη προκύπτει ότι από όλες τις activation function, η πιο αποδοτική υπολογιστικά είναι η τετραγωνική πολυωνυμική, η οποία εκτιμάται ότι θα μπορούσε εναλλακτικά να χρησιμοποιηθεί σε ένα φωτονικό σύστημα reservoir. Λαμβάνοντας υπόψιν ότι οι ερευνητές έχουν χρησιμοποιήσει τις tanh και sigmoid για υλοποιήσεις μη-γραμμικότητας στα φωτονικά RC συστήματα, προτείνεται σε συνέχεια της παρούσας διπλωματικής εργασίας, η μελέτη, ο σχεδιασμός και η υλοποίηση ενός φωτονικού RC που θα χρησιμοποιεί την κατάλληλη φωτοδίοδο για την παραγωγή της μη-γραμμικότητας.

7 Παράρτημα

Όπως ήδη αναφέρθηκε, ο κώδικας (esnToolbox.m) που χρησιμοποιήθηκε υλοποιήθηκε από τον Jaeger. Για τις ανάγκες της διπλωματικής εργασίας, υλοποιήθηκαν επιπλέον κομμάτια κώδικα προκειμένου να δημιουργηθούν οι activation function ReLu, Shift ReLu και πολυωνυμική.

```
function r = relu(x)
%% RELU activation function
% If the function receives any negative input, it returns 0;
% however, if the function receives any positive value x, it returns that
% value. As a result, the output has a range of 0 to infinite.

r0= zeros(length(x),1);

for n = 1:length(x)
    if x(n)>r0(n)
        r(n)=x(n);

    elseif x(n)<=r0(n)
        r=max(0,x);
    end
end
end
```

```
function r = shift_relu(x)
%% RELU activation function shifted
% like normal ReLu but shifted at x axes by 0.1 which is less than max(x)

% minl= min(x);
% maxl =max(x);
% %disp(sprintf('min ...%d, max...%d', minl, maxl));

x = x + 0.1;
r= zeros(length(x),1);
%disp(sprintf('mean(x) ...%d', mean(x)));
for n = 1:length(x)
    if x(n)>0
        r(n)=x(n);
    else
        r(n)=0;
    end
end
end
```

```
function p = polynomial_x2(x)
%% Polynomial Activation Function
%The formula is  $a \cdot x^2$ . The number of a is given through demoscrypt.m each
%time

global alpha

p = alpha*(x.^2);

end
```

Πίνακας Εικόνων

ΕΙΚΟΝΑ 1. ΕΙΔΗ ΑΛΓΟΡΙΘΜΩΝ MM. (MAHESH, 2020)	14
ΕΙΚΟΝΑ 2. ΓΡΑΦΙΚΗ ΑΝΑΠΑΡΑΣΤΑΣΗ SML. (KANEVSKY, 2016)	15
ΕΙΚΟΝΑ 3. UNSUPERVISED MACHINE LEARNING (KANEVSKY, 2016)	16
ΕΙΚΟΝΑ 4. ΓΡΑΦΙΚΗ ΑΝΑΠΑΡΑΣΤΑΣΗ ΤΕΧΝΙΚΩΝ ΕΚΜΑΘΗΣΗΣ SML ΚΑΙ NSML ΑΛΓΟΡΙΘΜΩΝ	16
ΕΙΚΟΝΑ 5. ΓΕΝΙΕΣ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ (NANDAKUMAR, 2018)	17
ΕΙΚΟΝΑ 6. ΓΡΑΦΗΜΑΤΑ ΤΩΝ ΠΙΟ ΣΥΧΝΑ ΧΡΗΣΙΜΟΠΟΙΟΥΜΕΝΩΝ ACTIVATION FUNCTIONS ΚΑΙ ΤΩΝ ΠΡΩΤΩΝ ΠΑΡΑΓΩΓΩΝ ΤΟΥΣ (OSTWALD, 2021)	19
ΕΙΚΟΝΑ 7. SINGLE LAYER PERCEPTRON ΜΕ STEP ACTIVATION FUNCTION (MAHESH, 2020)	20
ΕΙΚΟΝΑ 8. ΓΡΑΦΗΜΑ ΕΝΟΣ MULTI- LAYER PERCEPTRON ΜΕ ΔΥΟ HIDDEN LAYERS ΚΑΙ ΑΠΕΙΚΟΝΙΣΗ ΤΗΣ ΜΑΘΗΜΑΤΙΚΗΣ ΔΙΑΔΙΚΑΣΙΑΣ ΠΟΥ ΑΚΟΛΟΥΘΕΙΤΑΙ ΓΙΑ ΤΗΝ ΠΑΡΑΓΩΓΗ ΕΞΟΔΩΝ (PEREZ-ENCISO, 2019)	21
ΕΙΚΟΝΑ 9. ΕΠΕΞΕΡΓΑΣΙΑ ΠΛΗΡΟΦΟΡΙΑΣ ΣΕ ΕΝΑ NN ΠΟΥ ΔΙΑΘΕΤΕΙ ΑΝΑΔΡΟΜΕΣ (S. K. SAKSENA, 2019)	22
ΕΙΚΟΝΑ 10. ΔΙΑΔΙΚΑΣΙΑ “ΞΕΔΙΠΛΩΜΑΤΟΣ” ΤΩΝ ΑΝΑΔΡΟΜΩΝ ΣΕ ΕΝΑ RNN ΚΑΤΑ JAEGER (JAEGER H. , 2002)	22
ΕΙΚΟΝΑ 11. ΠΑΡΟΥΣΙΑΣΗ ΔΙΑΦΟΡΩΝ ΤΥΠΩΝ ΜΕ ΤΟΥΣ ΟΠΟΙΟΥΣ ΜΠΟΡΕΙ ΝΑ “ΞΕΔΙΠΛΩΘΟΥΝ” ΟΙ ΑΝΑΔΡΟΜΕΣ ΣΕ ΕΝΑ RNN (MURUGAN, 2018)	23
ΕΙΚΟΝΑ 12. ΚΡΙΤΙΚΗ ΣΕ ΔΙΑΔΙΚΤΥΑΚΗ ΣΕΛΙΔΑ	23
ΕΙΚΟΝΑ 13. ΤΑ ΚΥΡΙΑ ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΚΑΙ ΜΕΙΟΝΕΚΤΗΜΑΤΑ ΤΟΥ RNN (SHARKAWY, 2020)	23
ΕΙΚΟΝΑ 14. ΔΟΜΗ ΕΝΟΣ CONVOLUTIONAL NN (PHUNG, 2019)	24
ΕΙΚΟΝΑ 15. Α)ΒΙΟΛΟΓΙΚΟΙ ΝΕΥΡΩΝΕΣ Β) ΤΕΧΝΟΛΟΓΙΚΟΙ ΝΕΥΡΩΝΕΣ (BOUVIER, 2019)	25
ΕΙΚΟΝΑ 16. ΔΙΑΦΟΡΕΤΙΚΟΙ ΤΡΟΠΟΙ ΕΚΠΑΙΔΕΥΣΗΣ Α) ΕΝΟΣ RNN Β) ΕΝΟΣ RCN (LUKOSEVICIUS M. &, 2009)	26
ΕΙΚΟΝΑ 17. ΔΟΜΗ ΕΝΟΣ RESERVOIR (BUDHIRAJA, 2021)	28
ΕΙΚΟΝΑ 18. ΒΑΣΙΚΗ ΑΡΧΙΤΕΚΤΟΝΙΚΗ RESERVOIR ΣΥΣΤΗΜΑΤΩΝ (JAEGER H, 2001)]	31
ΕΙΚΟΝΑ 19. ΤΟΠΟΛΟΓΙΕΣ ESN (MAHESH, 2020)	36
ΕΙΚΟΝΑ 20. ΤΟΠΟΛΟΓΙΕΣ ΠΟΥ ΠΕΡΙΛΑΜΒΑΝΟΥΝ ΠΟΛΛΑΠΛΑ RESERVOIR	37
ΕΙΚΟΝΑ 21 Α) ΤΑ ΠΡΩΤΑ 200 ΣΤΟΙΧΕΙΑ ΤΟΥ INPUT [1-5000] ΓΙΑ ΕΚΠΑΙΔΕΥΣΗ ΤΟΥ ESN, Β) ΟΙ ΠΡΩΤΕΣ 200 ΤΙΜΕΣ ΤΟΥ TEACHER OUTPUT [1-5000] ΓΙΑ ΕΚΠΑΙΔΕΥΣΗ ΤΟΥ ESN	39
ΕΙΚΟΝΑ 22 Α) ΤΑ ΠΡΩΤΑ 200 ΣΤΟΙΧΕΙΑ ΤΟΥ INPUT [5001-5200] ΓΙΑ TESTING ΤΟΥ ESN, Β) ΟΙ ΠΡΩΤΕΣ 200 ΤΙΜΕΣ ΤΟΥ TEACHER OUTPUT [5001-5200] ΓΙΑ TESTING ΤΟΥ ESN	39
ΕΙΚΟΝΑ 23. TRAIN AND TEST NRMSE NARMA10 ΚΑΙ ΓΙΑ ΤΙΜΕΣ $P(W) > 1$ ΚΑΙ ACTIVATION FUNCTION TANH	40
ΕΙΚΟΝΑ 24. TRAIN AND TEST NRMSE NARMA10 ΓΙΑ ΤΙΜΕΣ ΠΟΥ ΠΑΡΑΤΗΡΕΙΤΑΙ ΣΦΑΛΜΑ NRMSE ΓΥΡΩ ΑΠΟ ΤΙΣ ΤΙΜΕΣ ΤΗΣ ΜΟΝΑΔΑΣ	41
ΕΙΚΟΝΑ 25. (Α,Β) ΔΙΑΓΡΑΜΜΑΤΑ ΓΙΑ RESERVOIR NODES =300 ΚΑΙ SPECTRAL RADIUS=0.5 ΤΑΥΤΙΖΟΝΤΑΙ. (Γ,Δ) ΔΙΑΓΡΑΜΜΑΤΑ ΓΙΑ RESERVOIR NODES =500 ΚΑΙ SPECTRAL RADIUS=1.5 ΠΑΡΟΥΣΙΑΖΟΥΝ ΑΠΟΚΛΙΣΗ	42
ΕΙΚΟΝΑ 26. TRAIN AND TEST NRMSE NARMA5 ΚΑΙ ΓΙΑ ΤΙΜΕΣ $P(W) > 1$	43
ΕΙΚΟΝΑ 27. TRAIN AND TEST NRMSE NARMA5 ΓΙΑ ΤΙΜΕΣ ΠΟΥ ΠΑΡΑΤΗΡΕΙΤΑΙ ΜΙΚΡΟ ΣΦΑΛΜΑ NRMSE (NRMSE <0.3)	44
ΕΙΚΟΝΑ 28. NARMA10 ΓΙΑ 400 ΚΑΙ 500 ΚΟΜΒΟΥΣ RESERVOIR ΚΑΙ NARMA5 ΓΙΑ 400 ΚΑΙ 500 ΚΟΜΒΟΥΣ RESERVOIR.	45
ΕΙΚΟΝΑ 29 ΔΙΑΓΡΑΜΜΑ ΑΠΕΙΚΟΝΙΣΗΣ ΣΦΑΛΜΑΤΟΣ NRMSE ΜΕ ACTIVATION FUNCTION SIGMOID ΓΙΑ $P(W) = [0.1, 1]$	47
ΕΙΚΟΝΑ 30. ΔΙΑΓΡΑΜΜΑ ΓΙΑ RESERVOIR NODES =400 ΚΑΙ SPECTRAL RADIUS=0.7 ΚΑΙ ACTIVATION FUNCTION SIGMOID ΤΑΥΤΙΖΟΝΤΑΙ	48
ΕΙΚΟΝΑ 31. ΔΙΑΓΡΑΜΜΑ ΑΠΕΙΚΟΝΙΣΗΣ ΣΦΑΛΜΑΤΟΣ NRMSE ΜΕ ACTIVATION FUNCTION IDENTITY ΓΙΑ $P(W) = [0.1, 1]$ ΣΕ ΠΡΟΣΑΡΜΟΣΜΕΝΗ ΚΛΙΜΑΚΑ ΓΙΑ ΕΥΚΡΙΝΕΙΑ	49
ΕΙΚΟΝΑ 32 ΔΙΑΓΡΑΜΜΑ ΓΙΑ RESERVOIR NODES =600 ΚΑΙ SPECTRAL RADIUS=1 ΚΑΙ ACTIVATION FUNCTION IDENTITY ΔΕΝ ΤΑΥΤΙΖΟΝΤΑΙ ..	50
ΕΙΚΟΝΑ 33. ΔΙΑΓΡΑΜΜΑ ΑΠΕΙΚΟΝΙΣΗΣ ΣΦΑΛΜΑΤΟΣ NRMSE ΜΕ ACTIVATION FUNCTION RELU ΓΙΑ $P(W) = [0.1, 1]$ ΚΑΙ ΟΛΟΥΣ ΤΟΥΣ ΚΟΜΒΟΥΣ	51
ΕΙΚΟΝΑ 34. ΔΙΑΓΡΑΜΜΑ ΓΙΑ RESERVOIR NODES =3000 ΚΑΙ SPECTRAL RADIUS=0.8 ΚΑΙ ACTIVATION FUNCTION RELU ΔΕΝ ΤΑΥΤΙΖΟΝΤΑΙ ..	52
ΕΙΚΟΝΑ 35. ΠΟΛΥΩΝΙΜΙΚΗ ACTIVATION FUNCTION ΓΙΑ $A=1$ ΚΑΙ $1=4$	54

ΕΙΚΟΝΑ 36. TRAIN ΚΑΙ TEST NRSME ΜΕ ACTIVATION FUNCTION POLYNOMIAL $A \cdot x^2$ ΜΕ $A=0.5$	55
ΕΙΚΟΝΑ 37. TRAIN ΚΑΙ TEST NRSME ΜΕ ACTIVATION FUNCTION POLYNOMIAL $A \cdot x^2$ ΜΕ $A=1$	56
ΕΙΚΟΝΑ 38 ΔΙΑΓΡΑΜΜΑ ΓΙΑ RESERVOIR NODES =2500 ΚΑΙ SPECTRAL RADIUS=0.9 ΚΑΙ ACTIVATION FUNCTION POLYNOMIAL ΜΕ $A=1$ ΟΙ ΚΑΜΠΥΛΕΣ ΤΕΙΝΟΥΝ ΝΑ ΤΑΥΤΙΣΤΟΥΝ	57
ΕΙΚΟΝΑ 39. TRAIN ΚΑΙ TEST NRSME ΜΕ ACTIVATION FUNCTION POLYNOMIAL $A \cdot x^2$ ΜΕ $A=4$	58
ΕΙΚΟΝΑ 40 ΔΙΑΓΡΑΜΜΑ ΓΙΑ RESERVOIR NODES =2000 ΚΑΙ SPECTRAL RADIUS=0.9 ΚΑΙ ACTIVATION FUNCTION POLYNOMIAL ΜΕ $A=4$, ΟΙ ΚΑΜΠΥΛΕΣ ΤΑΥΤΙΖΟΝΤΑΙ	59
ΕΙΚΟΝΑ 41. ΣΥΓΚΡΙΣΗ ΚΑΜΠΥΛΩΝ ΓΙΑ $N=300$ ΚΑΙ $N=500$ ΜΕ ΔΙΑΦΟΡΕΣ ACTIVATION FUNCTION ΠΟΥ ΥΛΟΠΟΙΗΘΗΚΑΝ ΣΕ ΠΡΟΗΓΟΥΜΕΝΑ ΚΕΦΑΛΑΙΑ.	61
ΕΙΚΟΝΑ 42. ΔΙΑΓΡΑΜΜΑΤΑ ΣΥΓΚΡΙΣΗΣ ΓΙΑ RELU ΚΑΙ ΔΙΑΦΟΡΕΣ ΠΟΛΥΩΝΥΜΙΚΕΣ ACTIVATION FUNCTIONS	61
ΕΙΚΟΝΑ 43 Α) VON NEUMANN ΜΟΝΤΕΛΟ Β) NON-VON NEUMANN ΜΟΝΤΕΛΟ (ΟΥ, 2020)	63
ΕΙΚΟΝΑ 44 Α) ROUTING IN TERRESTRIAL FREE SPACE OPTICAL AD-HOC NETWORKS Β) (ΠΑΘΑΚ, 2019)	64
ΕΙΚΟΝΑ 45 Α) ΟΠΤΙΚΗ ΙΝΑ ΚΑΙ ΠΕΡΙΒΛΗΜΑΤΑ ΤΗΣ Β) ΕΙΔΗ ΔΙΑΔΟΣΗΣ ΜΕΣΑ ΣΤΗΝ ΟΠΤΙΚΗ ΙΝΑ (ΑΜΙΡΙ, 2018).	65
ΕΙΚΟΝΑ 46. ΣΧΗΜΑΤΙΚΗ ΑΠΕΙΚΟΝΙΣΗ ΕΝΙΣΧΥΤΗ SOA	66
ΕΙΚΟΝΑ 47. ΣΧΗΜΑΤΙΚΗ ΑΠΕΙΚΟΝΙΣΗ ΕΝΟΣ COUPLER ΟΠΤΙΚΟΥ ΣΗΜΑΤΟΣ	66
ΕΙΚΟΝΑ 48. ΣΧΗΜΑΤΙΚΗ ΑΠΕΙΚΟΝΙΣΗ MZI Α) (SUNNY, 2021) Β) (ΠΑΘΑΚ, 2019)	67
ΕΙΚΟΝΑ 49. Α) ΣΧΗΜΑΤΙΚΗ ΑΠΕΙΚΟΝΙΣΗ MRR ΔΟΜΗΣ ΣΑΝ ADD/ DROP FILTER Β) ΣΧΗΜΑΤΙΚΗ ΑΠΕΙΚΟΝΙΣΗ MRR (SUNNY, 2021)	68
ΕΙΚΟΝΑ 50 Α) MACH-ZEHNDER ΔΙΑΜΟΡΦΩΤΗΣ (ΠΑΘΑΚ, 2019) Β) MICRORING ΔΙΑΜΟΡΦΩΤΗΣ (SUNNY, 2021)	68
ΕΙΚΟΝΑ 51. ΣΥΓΚΡΙΣΗ VON NEUMANN ΜΕ ΝΕΥΡΟΜΟΡΦΙΚΗ ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΕΠΕΞΕΡΓΑΣΙΑΣ (SCHUMAN C. D., 2022).	70
ΕΙΚΟΝΑ 52 . ΕΞΕΛΙΞΗ ΑΡΧΙΤΕΚΤΟΝΙΚΩΝ ΓΙΑ ΥΠΟΛΟΓΙΣΤΙΚΑ ΣΥΣΤΗΜΑΤΑ (PRUCNAL, 2017)	71
ΕΙΚΟΝΑ 53 ΝΕΥΡΩΝΑΣ ΜΕ MRR ΚΑΙ MZI ΚΑΙ ΜΕ LASER MODULATOR (GEORGE J. K., 2019)	73
ΕΙΚΟΝΑ 54. ΚΑΤΗΓΟΡΙΟΠΟΙΗΣΗ Ο/Ε/Ο PNN ΜΕ ΣΥΝΕΧΕΙΣ ΜΗ-ΓΡΑΜΜΙΚΟΤΗΤΕΣ ΚΑΙ ΠΙΘΑΝΕΣ ΥΛΟΠΟΙΗΣΕΙΣ. ΠΡΟΣΑΡΜΟΣΜΕΝΟ ΣΧΗΜΑ ΑΠΟ (DE LIMA T. F., 2017) (SHASTRI B. J., 2017)	74
ΕΙΚΟΝΑ 55. ΝΕΥΡΩΝΑΣ ΜΕ MRR WEIGHT BANK ΚΑΙ ΜΕ ΕΝΣΩΜΑΤΩΜΕΝΟ MODULATOR (SUNNY, 2021) (DE LIMA T. F., 2017)	74
ΕΙΚΟΝΑ 56. ΥΛΟΠΟΙΗΣΗ ΟΠΤΙΚΟΥ ΝΕΥΡΩΝΑ ΜΕ ΓΡΑΜΜΙΚΗ ΛΕΙΤΟΥΡΓΙΑ (SLM ΚΑΙ ΦΑΚΟΥΣ) ΚΑΙ ΜΗ ΓΡΑΜΜΙΚΗ ΛΕΙΤΟΥΡΓΙΑ (ΣΥΝΑΡΤΗΣΗ ΕΝΕΡΓΟΠΟΙΗΣΗΣ $\Phi(\cdot)$) (DEMERTZIS, 2022)	75
ΕΙΚΟΝΑ 57 (Α) ΑΡΧΙΤΕΚΤΟΝΙΚΗ NN ΠΟΥ ΟΛΟΙ ΟΙ ΚΟΜΒΟΙ ΣΥΝΔΕΟΝΤΑΙ ΜΕ ΚΑΘΕ ΑΛΛΟ ΚΟΜΒΟ (Β) ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΦΩΤΟΝΙΚΩΝ NN ΜΕ ΠΟΛΛΑ ΜΗΚΗ ΚΥΜΑΤΟΣ ΣΕ ΕΝΑΝ ΚΥΜΑΤΟΔΗΓΟ (DE LIMA T. F., 2020). (ΤΑΙΤ Α. Ν., 2014)	76
ΕΙΚΟΝΑ 58 Α) MZI ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΤΑΘΜΙΣΗΣ NN Β) CROSSBAR NETWORK ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΤΑΘΜΙΣΗΣ NN (EL SROUJI, 2022)	77
ΕΙΚΟΝΑ 59. MRR ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΔΟΜΗΣ ΕΝΟΣ ΦΩΤΟΝΙΚΟΥ ΝΕΥΡΩΝΙΚΟΥ ΔΙΚΤΥΟΥ (DE LIMA T. F., 2019), (ΤΑΙΤ Α. Ν., 2016), (DE LIMA T. F., 2017), (ΤΑΙΤ Α. Ν., 2014)	78
ΕΙΚΟΝΑ 60. SWIRL ΤΟΠΟΛΟΓΙΑ 4x4 PNN (Α) ΜΕ ΧΡΗΣΗ SOA (ΤΑΝΑΚΑ, 2019) (Β) ΜΕ ΧΡΗΣΗ MRR ADAPTED BY (DENIS-LE COARER, 2018)	80
ΕΙΚΟΝΑ 61. ΚΑΜΠΥΛΕΣ ΤΑΝΗ ΚΑΙ SOA (VANDOORNE K. D., 2008)	80
ΕΙΚΟΝΑ 62. (Α) ΟΠΤΟΗΛΕΚΤΡΟΝΙΚΟ RESERVOIR (Β) ΕΞΟΛΟΚΛΗΡΟΥ ΟΠΤΙΚΟ RESERVOIR. ΟΙ ΚΟΚΚΙΝΕΣ ΓΡΑΜΜΕΣ ΑΝΤΙΠΡΟΣΩΠΕΥΟΥΝ ΤΟ ΟΠΤΙΚΟ ΚΟΜΜΑΤΙ ΚΑΙ ΟΙ ΜΠΛΕ ΤΟ ΗΛΕΚΤΡΟΝΙΚΟ (BRUNNER D. L., 2022)	81

Πίνακας Πινάκων

ΠΙΝΑΚΑΣ 1 Α) ΚΑΤΗΓΟΡΙΕΣ ΕΦΑΡΜΟΓΩΝ RC ΚΑΙ Β) ΕΦΑΡΜΟΓΕΣ RC	29
ΠΙΝΑΚΑΣ 2. ΟΡΟΛΟΓΙΑ ΚΑΙ ΚΑΤΑΣΤΑΣΕΙΣ ΕΝΟΣ RESERVOIR	32
ΠΙΝΑΚΑΣ 3. ΣΥΝΔΕΣΜΟΛΟΓΙΑ RESERVOIR ΚΑΙ ΠΙΝΑΚΕΣ ΒΑΡΩΝ	32
ΠΙΝΑΚΑΣ 4. ΣΥΓΚΡΙΤΙΚΟΣ ΠΙΝΑΚΑΣ ΓΙΑ RELU ΚΑΙ SHIFT RELU ΑΠΟΤΕΛΕΣΜΑΤΑ	53
ΠΙΝΑΚΑΣ 5. ΔΕΙΓΜΑΤΟΛΗΠΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ TRAINNRMSE ΚΑΙ TESTNRMSE ΓΙΑ ΠΟΛΥΩΝΥΜΙΚΗ ΜΕ $A>5$, ΚΑΙ TANH, RELU.....	59
ΠΙΝΑΚΑΣ 6. ΒΕΛΤΙΣΤΕΣ ΠΕΡΙΟΧΕΣ ΤΙΜΩΝ ΚΟΜΒΩΝ ΚΑΙ $P(W)$ ΓΙΑ ΟΛΕΣ ΤΙΣ ACTIVATION FUNCTION ΠΟΥ ΔΟΚΙΜΑΣΤΗΚΑΝ ΜΕ NARMA10 ..	60
ΠΙΝΑΚΑΣ 7 ΠΕΡΙΛΗΨΗ ΤΩΝ ΒΑΣΙΚΩΝ ΣΥΣΤΑΤΙΚΩΝ ΤΩΝ PNN (Guo X. X., 2021)	73

Βιβλιογραφία

- Abiodun, O. I. (2018). State-of-the-art in artificial neural network applications: A survey. *Heliyon*, 4(11), e00938.
- Agrawal, G. P. (2010, october). Chapter 4: Optical Transmitters. In G. P. Agrawal, *Fiber-Optic Communication Systems* (pp. 79-127). John Wiley & Sons, Inc.
- Agrawal, G. P. (2012). *Fiber-optic communication systems*. John Wiley & Sons.
- Akiyama, T. &. (2019, July). Analysis on characteristics of multi-step learning echo state networks for nonlinear time series prediction. *2019 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 1-8.
- Akopyan, F. S.-I. (2015). Truenorth: Design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip. *IEEE transactions on computer-aided design of integrated circuits and systems*, 34(10), 1537-1557.
- Amin, R. G. (2019). ITO-based electro-absorption modulator for photonic neural activation function. *APL Materials*, 7(8), 081112.
- Amiri, I. S. (2018). Introduction to photonics: Principles and the most recent applications of microstructures. *Micromachines*, 9(9), 452.
- Amsalu, K. &. (2020). A review on photonics and its applications. . *Materials Today: Proceedings*, 33, , 3372-3377.
- Antonik, P. H. (2017). Online training for high-performance analogue readout layers in photonic reservoir computers. *Cognitive Computation*, 9(3), 297-306.
- Appeltant, L. S. (2011). Information processing using a single dynamical node as complex system. *Nature communications*, 2(1), 1-6.
- Argyris, A. (2022). Photonic neuromorphic technologies in optical communications. *Nanophotonics*, 11(5), 897-916.
- Ayodele, T. O. (2010). Types of machine learning algorithms. In *New advances in machine learning*, 3 (pp. 19-48).
- Barrett, D. G. (2019). Analyzing biological and artificial neural networks: challenges with opportunities for synergy? . *Current opinion in neurobiology*, 55-64.
- Basha, S. M. (2019). Survey on evaluating the performance of machine learning algorithms: past contributions and future roadmap. In *Deep Learning and Parallel Computing Environment for Bioengineering Systems*. Academic Press, (pp. 153-164).
- Basodi, S. J. (2020). Gradient amplification: An efficient way to train deep neural networks. . *Big Data Mining and Analytics*, 3(3), 196-207.
- Bazian, M. (2021). Photonic crystal add-drop filter: a review on principles and applications. *Photonic Network Communications*, 41(1), 57-77.

- Bouvier, M. V. (2019). Spiking neural networks hardware implementations and challenges: A survey. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 15(2), 1-35.
- Brunner, D. L. (2022). Nonlinear photonic dynamical systems for unconventional computing. Nonlinear Theory and Its Applications. *IEICE*, 13(1), 26-35.
- Brunner, D. P. (2018). Tutorial: Photonic neural networks in delay systems. *Journal of Applied Physics*, 15.
- Budhiraja, R. K. (2021). A reservoir computing approach for forecasting and regenerating both dynamical and time-delay controlled financial system behavior. *Plos one*, 16(2).
- Chang, H.-H. S. (2018). Distributive dynamic spectrum access through deep reinforcement learning: a reservoir computing based approach. *IEEE Internet of Things Journal*.
- Chen, W. B. (2010, July). Shared reservoir modular echo state networks for chaotic time series prediction. *2010 International Conference on Machine Learning and Cybernetics (Vol. 5)*. IEEE., 2439-2443.
- De Lima, T. F. (2017). Progress in neuromorphic photonics. *Nanophotonics*, 6(3), , 577-599.
- De Lima, T. F. (2019). Machine learning with neuromorphic photonics. *Journal of Lightwave Technology*, 37(5), 1515-1534.
- De Lima, T. F. (2020). Primer on silicon neuromorphic photonic processors: architecture and compiler. *Nanophotonics*, 9(13), , 4055-4073.
- De Marinis, L. C. (2019). Photonic neural networks: A survey. *IEEE Access*, 7, 175827-175841.
- Demertzis, K. P. (2022). A Comprehensive Survey on Nanophotonic Neural Networks: Architectures, Training Methods, Optimization, and Activations Functions. *Sensors*, 22(3), 720.
- Denis-Le Coarer, F. S. (2018). All-optical reservoir computing on a photonic chip using silicon-based ring resonators. *IEEE Journal of Selected Topics in Quantum Electronic* 24(6), 1-8.
- Ding, B. Q. (2018, June). Activation functions and their characteristics in deep neural networks. *2018 Chinese control and decision conference (CCDC)*. IEEE., 1836-1841.
- Eigenmann, R. &. (1998). Von neumann computers. *Wiley Encyclopedia of Electrical and Electronics Engineering*, 23, 387-400.
- El Srouji, L. K. (2022). Photonic and optoelectronic neuromorphic computing. *APL Photonics*, 7(5).
- Esmailzadeh, H. B. (2011, June). Dark silicon and the end of multicore scaling. . *38th Annual international symposium on computer architecture (ISCA)*, (pp. 365-376). IEEE.
- Fan, J. F. (2020). From brain science to artificial intelligence. *Engineering*, 248-252.
- Fermann, M. E. (1998). Single-mode excitation of multimode fibers with ultrashort pulses. *Optics Letters*, 23(1), 52-54.

- Ferreira, A. A. (2008, September). Using reservoir computing for forecasting time series: Brazilian case study. *Eighth International Conference on Hybrid Intelligent Systems, IEEE*, 602-607.
- Gallicchio, C. &. (2011). Architectural and markovian factors of echo state networks. *Neural Networks*, 24(5), 440-456.
- Gallicchio, C. M. (2017). Deep reservoir computing: a critical experimental analysis. *Neurocomputing*, 268, 87-99.
- Gardner, M. W. (1998). Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric environment*, 32(14-15), 2627-2636.
- George J. K., M. A. (2019). Neuromorphic photonics with electro-absorption modulators. *Optics express*, 27(4), 5181-5191.
- Ghahramani, Z. (2008). Unsupervised learning algorithms are designed to extract structure from data. 178. *IOS Press*, pp. 1-8.
- Ghosh, S. &. (2009). Spiking neural networks. *International journal of neural systems*, 19(04), 295-308.
- Ghosh-Dastidar, S. &. (2009). Third generation neural networks: Spiking neural networks. *Advances in Computational Intelligence, Springer, Berlin, Heidelberg*, 167-178.
- Goi, E. Z. (2020). Perspective on photonic memristive neuromorphic computing. *Photonix*, 1(1), 1-26.
- Goudarzi, A. B. (2014). A comparative study of reservoir computing for temporal signal processing. *arXiv preprint arXiv:1401.2224*.
- Guo, J. (2013). Backpropagation through time. *Unpubl. ms., Harbin Institute of Technology*, 40, 1-6.
- Guo, X. X. (2021). Integrated neuromorphic photonics: synapses, neurons, and neural networks. . *Advanced Photonics Research*, 2(6), 29.
- Huang, C. S.-Q. (2022). Prospects and applications of photonic neural networks. . *Advances in Physics: X*, 7(1), 1981155.
- Idachaba, F. I. (2014, July). Future trends in fiber optics communication . *Proceedings of the World Congress on Engineering (Vol. 1) London, UK: WCE.*, pp. 2-4.
- Islam, R. L. (2019). Device and materials requirements for neuromorphic computing. *Journal of Physics D: Applied Physics*, 52(11), 113001.
- Ivaniga, T. &. (2017). Comparison of the Optical Amplifiers EDFA and SOA Based on the BER and Q-Factor in C-Band. *Advances in Optical Technologies*.
- Jaeger H. (2001). The “echo state” approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, 148(34), 13.
- Jaeger, H. &. (2004). Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *science* 304(5667), 78-80.

- Jaeger, H. . (2002, March). Short term memory in echo state networks. *GMD-German National Research Institute for Computer Science*, gmd-report 152. Retrieved from [http://www. faculty. jacobs-university. de/hjaeger/pubs/STMEchoStatesTechRep. pdf](http://www.faculty.jacobs-university.de/hjaeger/pubs/STMEchoStatesTechRep.pdf).
- Jaeger, H. (2002). Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the "echo state network" approach.
- Jaeger, H. (2007). Echo state network. *Scholarpedia*, 2(9):2330.
- Jaeger, H. M. (2007). Special issue on echo state networks and liquid state machines.
- Jalalvand, A. V. (2015, June). Real-time reservoir computing network-based systems for detection tasks on visual contents. . *7th International Conference on Computational Intelligence, Communication Systems and Networks IEEE.*, 146-151.
- Jha, A. H. (2020). Reconfigurable all-optical nonlinear activation functions for neuromorphic photonics. *Optics letters*, 45(17), 4819-4822.
- Kanevsky, J. &. (2016). Big Data and Machine Learning in Plastic Surgery: A New Frontier in Surgical Innovation. *Plastic and reconstructive surgery.*, pp. 890-897.
- Ketkar, N. M. (2021). Recurrent Neural Networks . In N. M. Ketkar, *Deep Learning with Python* (pp. pp. 243–285).
- Klibisz, A. &. (2016). A Primer on Reservoir Computing.
- Konkoli, Z. N. (2018). Reservoir computing with computational matter. *Computational Matter , Springer, Cham*, 269-293.
- Li, Z. L. (2021). A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems*.
- Liao, Y. (2017). Reservoir computing trend on software and hardware implementation. *Global Journal of Research In Engineering*.
- Liu, J. W. (2021). Research progress in optical neural networks: theory, applications and developments. *Photonix*, 2(1), 1-39.
- Lukoševičius, M. &. (2009). Reservoir computing approaches to recurrent neural network training. *Computer Science Review.*, 127-149.
- Lukoševičius, M. (2012). A practical guide to applying echo state networks. *Neural networks: Tricks of the trade, Springer, Berlin, Heidelberg*, 659-686.
- Lukoševičius, M. J. (2012). Reservoir computing trends. *KI-Künstliche Intelligenz*, 26(4), 365-371.
- Maass, W. N. (2002). Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural computation*, 14(11), 2531-2560.
- Mahesh, B. (2020). Machine learning algorithms-a review. *International Journal of Science and Research (IJSR).[Internet]*, 9, pp. 381-386.

- Malik, Z. K. (2017). Multilayered echo state machine: a novel architecture and algorithm. . *IEEE Transactions on cybernetics*, 47,, 946-959.
- Manneschi, L. E. (2021). Exploiting multiple timescales in hierarchical echo state networks. *Frontiers in Applied Mathematics and Statistics*, 6, ,76.
- Margalit, N. X. (2021). Perspective on the future of silicon photonics and electronics. *Applied Physics Letters*, 118(22) , 220501.
- Martinuzzi, F. (2020). A brief introduction to Reservoir Computing. 2.
- Mayer, N. M. (2015). Input-anticipating critical reservoirs show power law forgetting of unexpected input events. *Neural computation*, 27(5), 1102-1119.
- McCulloch, W. S. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 115-133.
- Mesaritakis, C. P. (2013). Micro ring resonators as building blocks for an all-optical high-speed reservoir-computing bit-pattern-recognition system. *JOSA B*, 30(11), 3048-3055, 8.
- Millea, A. (2014). Explorations in echo state networks . *Doctoral dissertation, Faculty of Science and Engineering*.
- Murakami, M. K. (2015, August). Seeing [u] aids vocal learning: Babbling and imitation of vowels using a 3D vocal tract model, reinforcement learning, and reservoir computing. *2015 joint IEEE international conference on development and learning and epigenetic robotics (ICDL-EpiRob)*, IEEE., 208-213.
- Murugan, P. (2018). Learning the sequential temporal information with recurrent neural networks. . *arXiv preprint arXiv:1807.02857*.
- Nandakumar, S. R. (2018). Building brain-inspired computing systems: Examining the role of nanoscale devices. *IEEE Nanotechnology Magazine*, 19-35.
- Noriega, L. (2005). Multilayer perceptron tutorial. School of Computing. *Staffordshire University*.
- Nwankpa, C. I. (2018). Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint arXiv:1811.03378*. 20.
- O'Mahony, M. J. (1989, October). Optical Amplification Techniques Using Semiconductors And Fibres. *Fibre Optics' 89 (Vol. 1120)*. SPIE, pp. 43-44.
- Osisanwo, F. Y. (2017). Supervised machine learning algorithms: classification and comparison. *International Journal of Computer Trends and Technology (IJCTT)*, 48(3), 128-13.
- Ostwald, D. &. (2021). An induction proof of the backpropagation algorithm in matrix notation. *arXiv preprint arXiv:2107.09384*.
- Ou, Q. F. (2020). In-memory logic operations and neuromorphic computing in non-volatile random access memory. *Materials*, 13(16), 3532.
- Pankove, J. I. (n.d.). Semiconductor Diode Lasers: Early History. *OSA Century of Optics*, 107.

- Paquot, Y. D. (2012). Optoelectronic reservoir computing. *Scientific reports*, 2(1), 1-6.
- Pasricha, S. &. (2020). A survey of silicon photonics for energy-efficient manycore computing. *IEEE Design & Test*, 37(4), 60-81.
- Pathak, S. (2019). Photonics Integrated Circuits. In B. K. Kaushik, *Nanoelectronics* (pp. 219-270). Elsevier Inc.
- Pérez-Enciso, M. &. (2019). A guide on deep learning for complex trait genomic prediction. *Genes*, 10(7), 553.
- Phung, V. H. (2019). A high-accuracy model average ensemble of convolutional neural networks for classification of cloud image patches on small datasets. *Applied Sciences*, 9(21).
- Polydoros, A. N. (2015). Advantages and limitations of reservoir computing on model learning for robot control. *IROS Workshop on Machine Learning in Planning and Control of Robot Motion*.
- Ponulak, F. &. (2011). Introduction to spiking neural networks: Information processing, learning and applications. *Acta neurobiologiae experimentalis*, 71(4), 409-433.
- Prucnal, P. R. (2017). *Neuromorphic photonics*. CRC Press.
- Psaltis, D. B. (1988). Adaptive optical networks using photorefractive crystals. *Applied Optics*, 27(9), 1752-1759.
- Pyle, R. &. (2018). A model of reward-modulated motor learning with parallelcortical and basal ganglia pathways. *arXiv preprint arXiv:1803.03304*.
- Qiao, J. L. (2017). Growing echo-state network with multiple subreservoirs. *IEEE Transactions on Neural Networks and Learning Systems*, 28,, 391–404.
- R. H. Dennard, F. H.-N. (1974, Oct). Design of ion-implanted MOSFET's with very small physical dimensions," *IEEE Journal of Solid-State Circuits*, vol. 9, no. 5, pp. 256-268.
- Reed, G. T. (2010). Silicon optical modulators. *Nature photonics*, 4(8), 518-526.
- Rodan, A. &. (2010). Minimum Complexity Echo Sstate Networks. *IEEE transactions on neural networks*, 22(1), 131-144.
- Roli, A. &. (2014). Introduction to Reservoir Computing Methods. 52.
- Rongqing Hui, M. O. (2009). Chapter 3 - Characterization of Optical Devices. In //doi.org/10.1016/B978-0-12-373865-3.00003-3., *Fiber Optic Measurement Techniques* (pp. 259-363). Academic Press.
- Rose, G. S. (2021). A system design perspective on neuromorphic computer processors. *Neuromorphic Computing and Engineering*, , 1(2), 022001.
- S. K. Saksena, N. B. (2019). Towards Behavioural Cloning for Autonomous Driving. *2019 Third IEEE International Conference on Robotic Computing (IRC)*, 560-567, doi: 10.1109/IRC.2019.00115.
- Sakemi, Y. M. (2020). Model-size reduction for reservoir computing by concatenating internal states through time. *Scientific reports*, 10(1), 1-13.

- Sakib, S. A. (2019). An overview of convolutional neural network: its architecture and applications.
- Sarbazi-Azad, P. L.-K. (2022). Chapter One - Introduction to data prefetching,. In H. S.-A. Pejman Lotfi-Kamran, *Advances in computers* (pp. 1-17). <https://doi.org/10.1016/bs.adcom.2021.11.003>.
- Schaetti, N. S. (2016, August). Echo state networks-based reservoir computing for mnist handwritten digits recognition. *IEEE Intl conference on CSE and IEEE Intl conference on EUC and 15th Intl symposium on DCABES*, 484-491.
- Schiller, U. D. (2005). Analyzing the weight dynamics of recurrent learning algorithms. *Neurocomputing*, 63, 5-23.
- Schrauwen, B. V. (2007). An overview of reservoir computing: theory, applications and implementations. *Proceedings of the 15th european symposium on artificial neural networks*, 471-482.
- Schuller, I. K. (2015). Neuromorphic computing—from materials research to systems architecture roundtable. *USDOE Office of Science (SC)(United States)*.
- Schuman, C. D. (2017). A survey of neuromorphic computing and neural networks in hardware. *arXiv preprint arXiv:1705.06963*.
- Schuman, C. D. (2022). Opportunities for neuromorphic computing algorithms and applications. *Nature Computational Science*, 2(1), 10-19.
- Seoane, L. F. (2019). Evolutionary aspects of reservoir computing. *Philosophical Transactions of the Royal Society B*, 374(1774).
- Sharkawy, A. N. (2020). Principle of neural network and its main types. *Journal of Advances in Applied & Computational Mathematics*, 7, 8-19.
- Sharma, S. S. (2017). Activation functions in neural networks. *towards data science*, 6(12), 310-316.
- Shastri, B. J. (2017). Principles of neuromorphic photonics. *arXiv preprint arXiv:1801.00016*.
- Shastri, B. J. (2021). Photonics for artificial intelligence and neuromorphic computing. *Nature Photonics*, 15(2), 102-114.
- Singh, A. T. (2016, March). A review of supervised machine learning algorithms. In *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)* . Ieee, pp.1310-1315.
- Smerieri, A. D. (2012). Analog readout for optical reservoir computers. *Advances in neural information processing systems*, 25.
- Streetman, B. G. (2006). *Solid state electronic devices (Vol. 10)*. . Upper Saddle River: Pearson/Prentice Hall.
- Sun, C. S. (2020). A review of designs and applications of echo state networks. *arXiv preprint arXiv:2012.02974*.
- Sunny, F. P. (2021). A survey on silicon photonics for deep learning. *ACM Journal of Emerging Technologies in Computing System*, 17(4), 1-57.

- Sussillo, D. &. (2009). Generating coherent patterns of activity from chaotic neural networks . *Neuron*, 63(4), 544-557.
- Suzuki, Y. G. (2022). Natural quantum reservoir computing for temporal information processing. *Scientific reports*, 12(1), 1-15.
- Tait, A. N. (2014). Broadcast and weight: an integrated network for scalable photonic spike processing. *Journal of Lightwave Technology*, 32(21), 3427-3439.
- Tait, A. N. (2016). Microring weight banks. *IEEE Journal of Selected Topics in Quantum Electronics*, 22(6), 312-325.
- Tait, A. N. (2019). Silicon photonic modulator neuron. *Physical Review Applied*, 11(6), 064043.
- Tanaka, G. Y. (2019). Recent advances in physical reservoir computing: A review . *Neural Networks*, 115, 100-123.
- Tong, Z. &. (2018). Reservoir computing with untrained convolutional neural networks for image recognition. *Proc. 24th international conference on pattern recognition (ICPR). IEEE.*, 1289–1294.
- Van der Sande, G. B. (2017). Advances in photonic reservoir computing. *Nanophotonics*, 6(3) , 561-576.
- Vandoorne, K. D. (2008). Toward optical signal processing using photonic reservoir computing. *Optics express*, 16(15), 11182-11192.
- Vandoorne, K. M. (2014). Experimental demonstration of reservoir computing on a silicon photonics chip. *Nature communications*, 5(1), 1-6.
- Verstraeten, D. D. (2010, July). Memory versus non-linearity in reservoirs. *The 2010 international joint conference on neural networks (IJCNN), IEEE*, 1-8.
- Verstraeten, D. S. (2007). An experimental unification of reservoir computing methods. *Neural networks*, 20(3), 391-403.
- Wagner, K. &. (1993). Optical neural networks: an introduction by the feature editors. *Applied Optics*,, 32(8), 1261-1263.
- Werbos, P. J. (1990). Back Propagation Through Time what it does and how to do it. *Proc. IEEE*, 10.
- Xia, Y. J. (2010). An augmented echo state network for nonlinear adaptive filtering of complex noncircular signals. *IEEE Transactions on Neural Networks*, 22(1), 74-83.
- Xu., Y. (2020, March). Machine Learning With Echo State Networks. 65.
- Yildiz, I. B. (2012). Re-visiting the echo state property . *Neural networks*, 35, 1-9.
- Yuan P. Li, C. H. (1997). Silicon Optical Bench Waveguide Technology. In *Optical Fiber Telecommunications (Third Edition), Volume B*.
- Zetie, K. P. (2000). How does a Mach-Zehnder interferometer work? *Physics Education*, 35(1), 46.
- Zhang, D. M. (2020). Tuning the performance of polymeric microring resonator with femtosecond laser. *Optics Communications*, 465, 125571.

- Zhou, P. C. (2005). Movie Review Mining: a Comparison between Supervised and Unsupervised Classification Approaches. *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, pp. 112.
- Zou, X. X. (2021). Breaking the von Neumann bottleneck: architecture-level processing-in-memory technology. *Science China Information Sciences*, 64(6), 1-10.
- Zuo, Y. L. (2019). All-optical neural network with nonlinear activation functions. *Optica*, 6(9), 1132-1137.