



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΙΓΑΙΟΥ

UNIVERSITY OF THE
AEGEAN

Πρόγραμμα Μεταπτυχιακών Σπουδών
Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών
Συστημάτων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

*«« Μελέτη Νευρωνικών Δίκτυων Ώσεων για αναγνώριση εικόνων
»»*

ΟΝΟΜΑΤΕΠΩΝΥΜΟ: ΚΟΛΛΙΑΣ ΕΥΑΓΓΕΛΟΣ

A.M : ICSDM422002

Διαδίκτυο των Πραγμάτων: Τεχνητή Νοημοσύνη και Σύγχρονα Δίκτυα

ΜΑΙΟΣ 2025



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΙΓΑΙΟΥ

UNIVERSITY OF THE
AEGEAN

Πρόγραμμα Μεταπτυχιακών Σπουδών
Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών
Συστημάτων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

*«« Μελέτη Νευρωνικών Δίκτυων Ώσεων για αναγνώριση εικόνων
»»*

ΟΝΟΜΑΤΕΠΩΝΥΜΟ: ΚΟΛΛΙΑΣ ΕΥΑΓΓΕΛΟΣ

A.M : ICSDM422002

Ο Επιβλέπων

Καθηγητής: ΜΕΣΑΡΙΤΑΚΗΣ ΧΑΡΗΣ

ΜΑΙΟΣ 2025

Copyright © <Κόλλιας Ευάγγελος>, <2025>
Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Η παρούσα διπλωματική εργασία εκπονήθηκε στο πλαίσιο των απαιτήσεων του
Μεταπτυχιακού Προγράμματος *Διαδίκτυο των Πραγμάτων: Τεχνητή Νοημοσύνη και
Σύγχρονα Δίκτυα* του Πανεπιστημίου Αιγαίου.

Η έγκρισή της παρούσας μελέτης δεν υποδηλώνει απαραίτητως και την αποδοχή
των απόψεων του συγγραφέα εκ μέρους του Πανεπιστημίου.

Βεβαιώνω ότι η παρούσα Μεταπτυχιακή Διατριβή είναι αποτέλεσμα δικής μου δουλειάς και δεν αποτελεί προϊόν αντιγραφής. Στις δημοσιευμένες ή μη δημοσιευμένες πηγές που αναφέρω έχω χρησιμοποιήσει εισαγωγικά και πλάγια γραφή, όπου απαιτείται, και έχω παραθέσει όλες τις πηγές τους στο κείμενο και στο τμήμα της βιβλιογραφίας.

Υπογραφή:

Κόλλιας Ευάγγελος

Ευχαριστίες

Θα ήθελα να εκφράσω τις εγκάρδιες ευχαριστίες μου στον καθηγητή μου, Χάρη Μεσαριτάκη, για την πολύτιμη καθοδήγηση, την αδιάκοπη υποστήριξη και την ενθάρρυνσή του καθ' όλη τη διάρκεια της εργασίας αυτής. Η πολύτιμη βοήθειά του και η αμέριστη αφοσίωσή του συνέβαλαν καθοριστικά στην ολοκλήρωση αυτού του έργου. Επίσης θα ήθελα να εκφράσω την ευγνωμοσύνη μου στον φίλο μου Βασίλη για την συμπαράσταση του καθ' όλη την διάρκεια του μεταπτυχιακού προγράμματος. Ιδιαίτερες ευχαριστίες απευθύνω στην μητέρα μου Βαρβάρα και τον πατέρα μου Δημήτρη για την ατελείωτη υποστήριξη, την αγάπη και την κατανόησή τους. Η υπομονή και η ενθάρρυνσή τους ήταν αδιαμφισβήτητα καθοριστικοί παράγοντες για την ολοκλήρωση αυτής της έρευνας.

Αυτή η δουλειά είναι αφιερωμένη σε όλους όσους συνέβαλαν με οποιονδήποτε τρόπο στην ολοκλήρωσή της. Η βοήθειά σας ήταν ανεκτίμητη και η συμβολή σας καθοριστική. Σας ευχαριστώ από καρδιάς.

Περίληψη

Η διπλωματική αυτή εργασία εξετάζει τη χρήση των Νευρωνικών Δικτύων Όσεων (Spiking Neural Networks – SNNs) για την ταξινόμηση εικόνων σε νευρομορφικά δεδομένα. Συγκεκριμένα, γίνεται μια εκτενής ανάλυση της απόδοσης των SNNs στο σύνολο δεδομένων N-MNIST μέσω 30 πειραματικών σεναρίων. Οι μέθοδοι που χρησιμοποιούνται περιλαμβάνουν την αλλαγή των surrogate gradient, διάφορες αρχιτεκτονικές των συνελκτικών νευρωνικών δικτύων, τις συναρτήσεις απώλειας και το μέγεθος των πακέτων δεδομένων που χρησιμοποιούνται.

Από τα αποτελέσματα προκύπτει ότι η επιλογή των surrogate συναρτήσεων, καθώς και η αρχιτεκτονική, έχουν σημαντική επίδραση στη σταθερότητα της εκπαίδευσης, στην ακρίβεια της ταξινόμησης και στην ενεργειακή απόδοση των δικτύων. Επιπλέον, προτείνονται οι βέλτιστοι συνδυασμοί των παραμέτρων που μπορούν να χρησιμοποιηθούν στο μέλλον, ιδίως σε εφαρμογές χαμηλής κατανάλωσης ενέργειας.

Η εργασία αυτή αποτελεί έναν ισχυρό προάγγελο της δυναμικής των SNNs ως μια υποσχόμενη προσέγγιση στη σύγχρονη εποχή της τεχνητής νοημοσύνης, με έμφαση στην αποδοτικότητά τους και στην εμπνευσμένη από τη φυσική λειτουργία τους.

Λέξεις κλειδιά: *Spiking Neural Networks (SNNs), Surrogate Gradients, N-MNIST, Ταξινόμηση Εικόνων, Αραιότητα, Ενεργειακή Απόδοση, snnTorch, Νευρομορφικά Συστήματα, Deep Learning*

Abstract

This thesis examines the use of Spiking Neural Networks (SNNs) for image classification on neuromorphic data. Specifically, an extensive analysis of the performance of SNNs on the N-MNIST dataset is performed through 30 experimental scenarios. The methods used include changing surrogate gradients, various convolutional neural network architectures, loss functions and the size of the data packets used.

The results show that the choice of surrogate functions, as well as the architecture, have a significant impact on the stability of training, the accuracy of classification and the energy efficiency of the networks. In addition, the optimal combinations of parameters that can be used in the future, especially in low-power applications, are proposed.

This work is a strong harbinger of the potential of SNNs as a promising approach in the modern era of artificial intelligence, with an emphasis on their efficiency and nature-inspired functionality.

Keywords: Spiking Neural Networks (SNNs), Surrogate Gradients, N-MNIST, Image Classification, Sparsity, Energy Efficiency, snnTorch, Neuromorphic Systems, Deep Learning

Πίνακας περιεχομένων

Εισαγωγή	10
ΜΕΡΟΣ ΠΡΩΤΟ	14
ΘΕΩΡΗΤΙΚΟ ΠΛΑΙΣΙΟ	14
ΚΕΦΑΛΑΙΟ 1 ^ο	15
Θεωρητικό Υπόβαθρο	15
1.1 Τεχνητά Νευρωνικά Δίκτυα.....	15
1.2 Νευρωνικά Δίκτυα Όσεων.....	16
1.3 Συσχέτιση SNNs με Επαναληπτικά Δίκτυα (RNNs).....	19
1.4 Ο κανόνας STDP.....	20
1.5 Surrogate Gradients και Κωδικοποίηση Ακίδων	22
ΚΕΦΑΛΑΙΟ 2 ^ο	27
Σύγχρονες Εφαρμογές SNNs για Ανάλυση Εικόνας (State of the Art)	27
2.1 Εισαγωγή	27
2.2.1 Sa-SNN: Spiking Attention Network για Αναγνώριση Εικόνων	28
2.2.2 Υβριδικά CNN-SNN Μοντέλα και Εκπαίδευση με Surrogate Gradients	29
2.2.3 Νευρομορφική Υπολογιστική σε Κυτταρομετρία Ροής: Σύγχρονες Προσεγγίσεις.....	30
2.2.4 Μέθοδοι Κωδικοποίησης Πληροφορίας σε SNNs	33
2.2.5 SNNs σε Υπολογιστικά Περιορισμένα Περιβάλλοντα: Εφαρμογές σε Edge Συσκευές και Νευρομορφικό Υλικό	34
<i>Νευρομορφικά Υπολογιστικά Συστήματα</i>	35
<i>Εφαρμογές σε Edge Περιβάλλοντα</i>	36
ΜΕΡΟΣ ΔΕΥΤΕΡΟ	38
ΠΕΙΡΑΜΑΤΙΚΟ ΠΛΑΙΣΙΟ	38
ΚΕΦΑΛΑΙΟ 3 ^ο	39
Μεθοδολογία	39
3.1 Περιγραφή Πειραμάτων.....	39
3.2 Περιγραφή Συνόλου Δεδομένων και Προεπεξεργασία	48
3.3 Αρχιτεκτονική Δικτύου και Ρυθμίσεις Εκπαίδευσης.....	53
ΚΕΦΑΛΑΙΟ 4 ^ο	58
Πειραματικό Μέρος	58
4.1 Πίνακες Πειραμάτων.....	58
4.2 Κατηγορία προσεγγιστικής συνάρτησης ενεργοποίησης	59
4.2.1 ATan	59
4.2.2 Fast Sigmoid.....	60
4.2.3 SpikeRateEscape	60

4.3 Κατηγορία Αρχιτεκτονικής	61
4.3.1 Conv(2→12→32)	61
4.3.2 Conv(2→24→48)	61
4.4 Κατηγορία Συναρτήσεων Απώλειας	62
4.4.1 SF.ce_count_loss	62
4.4.2 SF.ce_rate_loss	62
4.4.3 SF.mse_count_loss	63
4.4.4 SF.mse_membrane_loss	64
4.5 Υπερπαράμετρος Ρυθμού εκμάθησης	64
ΚΕΦΑΛΑΙΟ 5°	66
Αξιολόγηση και Συζήτηση Αποτελεσμάτων	66
5.1 Πειραματική μεθοδολογία και παραλλαγές	66
5.2 Επίδραση του Surrogate Gradient	67
5.3 Επίδραση της Συνάρτησης Απώλειας	68
5.4 Επίδραση της Αρχιτεκτονικής	68
5.5 Επίδραση Ρυθμού Μάθησης και Μέγεθος Παρτίδας	69
5.6 Sparsity και Ενεργειακή Απόδοση	69
5.7 Συγκριτική Ανάλυση Καμπυλών Εκπαίδευσης	70
5.8 Συνολική Τελική Αξιολόγηση	71
ΚΕΦΑΛΑΙΟ 6°	72
Συμπεράσματα και Μελλοντική Εργασία	72
Μελλοντική Εργασία	73
Βιβλιογραφία	75

Εισαγωγή

Τα Τεχνητά Νευρωνικά Δίκτυα (ΤΝΔ) έχουν επιτύχει σημαντικές επιδόσεις σε πολλά προβλήματα που αφορούν την όραση, την αναγνώριση προτύπων και την τεχνητή νοημοσύνη γενικότερα. Παρ' όλα αυτά, η υψηλή αποδοτικότητα των ΤΝΔ συνοδεύεται από την έντονη χρήση υπολογιστικής ισχύος και ενέργειας. Αυτό καθιστά πολύ δύσκολη τη χρήση τους σε συστήματα χαμηλής κατανάλωσης ισχύος ή σε κατανομημένες εφαρμογές του Διαδικτύου των Πραγμάτων (Internet of Things - IoT), όπου η εξοικονόμηση ενέργειας είναι πάρα πολύ σημαντική. (Tavanaei et al., 2019).

Η εξέλιξη των τεχνητών νευρωνικών δικτύων χωρίζεται σε τρεις βασικές φάσεις σύμφωνα με τον Maass (1997), με κύρια αρχή την αναπαράσταση των βιολογικών νευρώνων και την εκτέλεση υπολογισμών.

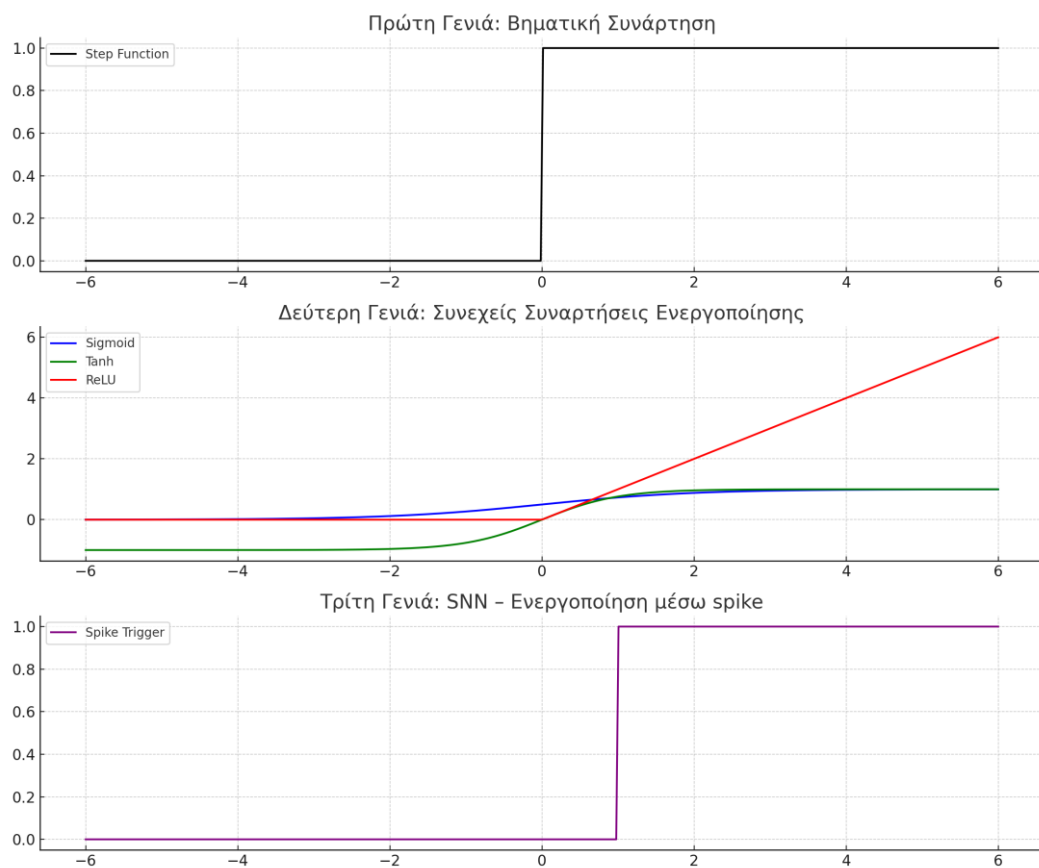
Η Πρώτη Γενιά (Perceptron-Based): Τα αρχικά μοντέλα περιελάμβαναν τον δυαδικό νευρώνα τύπου McCulloch-Pitts και τα δίκτυα Perceptron. Οι συναρτήσεις ενεργοποίησης των νευρώνων ήταν βηματικές και έβγαζαν δυαδικά αποτελέσματα (0 ή 1). Η υπολογιστική τους ισχύς ήταν περιορισμένη, αφού μπορούσαν να αντιμετωπίσουν μόνο γραμμικά διαχωρίσιμα προβλήματα (Maass, W. (1997)).

Η δεύτερη Γενιά (Δίκτυα με συνεχείς τιμές): Εισήχθησαν οι συνεχείς, μη γραμμικές συναρτήσεις ενεργοποίησης, όπως οι sigmoid, tanh και στη συνέχεια η ReLU, επιτρέποντας την οπισθοδιάδοση (backpropagation) και την ανάπτυξη βαθιών νευρωνικών δικτύων (Maass, W. (1997)).

Τα Νευρωνικά Δίκτυα Τρίτης Γενιάς (Spiking Neural Networks): Τα SNNs λειτουργούν με τη βοήθεια της χρονικής κωδικοποίησης και της επικοινωνίας με "spikes" - δηλαδή με τη βοήθεια γεγονότων που μοιάζουν με τις βιολογικές νευρικές ώσεις. Οι νευρώνες ενεργοποιούνται μόνο όταν η δυναμική είσοδος ξεπερνά ένα ορισμένο κατώφλι, κάτι που

προσδίδει ρεαλισμό και εξοικονόμηση ενέργειας (Maass, 1997; Roy et al., 2019; Eshraghian et al., 2021).

Η μετάβαση από μία γενιά στην άλλη συνδυάστηκε με την αύξηση της υπολογιστικής ισχύος και τη δυνατότητα επίλυσης πιο πολύπλοκων προβλημάτων. Το Σχήμα 1 αναλύει τις χαρακτηριστικές συναρτήσεις ενεργοποίησης κάθε γενιάς, οι οποίες επιδεικνύουν τη θεωρητική πρόοδο από τη δυαδική λογική στη συνεχή ενεργοποίηση και στη δυναμική δημιουργία των spikes.



Εικόνα 1: Συναρτήσεις ενεργοποίησης για κάθε γενιά νευρωνικών δικτύων (σχεδιάστηκε με βάση τα πρότυπα από Maass, 1997 και Eshraghian et al., 2021).

Όπως προαναφέρθηκε στα συμβατικά δίκτυα (ANNs), οι τεχνητοί νευρώνες επικοινωνούν με τη χρήση συνεχών πραγματικών τιμών σε διακριτά χρονικά βήματα, ενώ στα SNN οι

νευρώνες εκπέμπουν ακίδες δυαδικού τύπου (0/1) – τους λεγόμενους παλμούς – σε πραγματικό (συνεχή) χρόνο . Αυτή η διαφορά στον τρόπο επικοινωνίας καθιστά τα SNN ενεργειακά αποδοτικότερα, καθώς η πληροφορία κωδικοποιείται με ασύγχρονους και σπάνιους παλμούς αντί για συνεχείς ροές αριθμών . Η φύση των ακίδων που είναι αραιή και δυαδική μειώνει την έκταση των υπολογιστών (Eshraghian et al., 2021) και μπορεί να αξιοποιηθεί σε εξειδικευμένο νευρομορφικό υλικό για την αποτελεσματική μείωση της κατανάλωσης ισχύος.

Παρά τις πολλές αρετές τους, τα Νευρωνικά Δίκτυα Όσεων δεν έχουν επιτύχει έως τώρα το ίδιο επίπεδο ακρίβειας με τα κλασικά δίκτυα σε δύσκολα προβλήματα, εξαιτίας των προκλήσεων που υπάρχουν στο να τα εκπαιδεύσουμε. Το βασικό πρόβλημα είναι ότι οι συναρτήσεις ενεργοποίησης ώσης (βηματική συνάρτηση) είναι μη διαφοροποιήσιμες, δηλαδή δεν παραγωγίζονται με αποτέλεσμα να παίρνουν τιμές μηδέν (0) ή άπειρο, με αποτέλεσμα να μην μπορεί να χρησιμοποιηθεί απευθείας ο αλγόριθμος της οπισθοδιάδοσης (backpropagation), όπως γίνεται στα ANN. Για να ξεπεραστεί αυτό το εμπόδιο, έχουν προταθεί τεχνικές όπως μερικές προσεγγιστικές παράγωγοι (αγγλ. Surrogate gradients) (Neftci et al., 2019; Eshraghian et al., 2023) που βασίζονται στην εκτίμηση των παραγώγων με την εφαρμογή μεθόδων βελτιστοποίησης, χρησιμοποιώντας την κλίση, στα SNN. Τα τελευταία χρόνια, κάνοντας χρήση τέτοιων τεχνικών και υιοθετώντας βαθιές δομές μάθησης, οι επιδόσεις των SNN έχουν σημειώσει σημαντική βελτίωση, με αποτέλεσμα να μπορούν να ανταγωνιστούν ευθέως ορισμένα σύνολα δεδομένων των ANN.

Στην παρούσα διπλωματική εργασία διερευνώνται τα Νευρωνικά Δίκτυα όσεων(SNNs) και η εφαρμογή τους στην ταξινόμηση εικόνων. Αναλύεται η απόδοσή τους στο σύνολο δεδομένων N-MNIST, το οποίο προέκυψε από το διάσημο MNIST ως ακολουθίες γεγονότων (spike events). Η υλοποίηση των SNN μοντέλων γίνεται με τη βοήθεια της βιβλιοθήκης snnTorch (snnTorch Documentation, n.d.) σε περιβάλλον Python/Google Colab. Έγιναν συνολικά 30 πειράματα με διάφορες ρυθμίσεις του δικτύου και των αλγορίθμων εκπαίδευσης. Αυτές περιλάμβαναν τη χρήση διαφορετικών surrogate gradient, συναρτήσεων απώλειας, αρχιτεκτονικών δικτύων και μεγεθών batch. Μέσω αυτών των πειραμάτων εξετάζονται οι συνδυασμοί παραμέτρων που οδηγούν στα καλύτερα αποτελέσματα ως προς την ταξινόμηση, καθώς και η επίδραση των επιλογών στην απόδοση του δικτύου (π.χ. αραιότητα των spikes). Παρουσιάζονται τα βασικά ευρήματα, ενώ δίνονται και προτάσεις για

βέλτιστες ρυθμίσεις (π.χ. ποια surrogate συνάρτηση να επιλεγεί και ποιος αλγόριθμος κόστους).

Δομή της Εργασίας

Η παρούσα διατριβή αποτελείται από δύο βασικά μέρη: το Θεωρητικό Πλαίσιο και το Πειραματικό Πλαίσιο.

Στο Πρώτο Μέρος – *Θεωρητικό Πλαίσιο*, η **Ενότητα 1** παρουσιάζει επιγραμματικά τη θεωρητική βάση των Όσων Νευρωνικών Δικτύων Αιχμών (SNNs), τις λειτουργίες ενεργοποίησης, τις αντικαταστάσεις των κλίσεων (surrogate gradients) και τους κωδικοποιητές ακίδων. Το **Κεφάλαιο 2** μιλά για τις πιο σύγχρονες τεχνικές που επικρατούν αυτή την στιγμή στον κόσμο σχετικά με τα SNN, καθώς και για αποτελέσματα τους στην ανάλυση εικόνας.

Το Δεύτερο Μέρος – *Πειραματικό Πλαίσιο* περιλαμβάνει τα **Κεφάλαια 3 έως 6**. Στο κεφάλαιο 3 παρουσιάζεται η μεθοδολογία που ακολουθήθηκε, περιλαμβανομένου του συνόλου δεδομένων, της διαδικασίας προ επεξεργασίας, των δομών δικτύων και της διαμόρφωσης της εκπαίδευσης. Το κεφάλαιο 4 παρέχει λεπτομερή ανάλυση των πειραματικών αποτελεσμάτων από 30 διαφορετικές δοκιμές, ενώ το Κεφάλαιο 5 γίνεται αξιολόγηση των αποτελεσμάτων και συζητούνται οι επιδράσεις των διάφορων παραμέτρων στην απόδοση των μοντέλων. Τέλος, στο Κεφάλαιο 6 περιλαμβάνονται τα βασικά συμπεράσματα και παρουσιάζονται κάποιες προτάσεις για μελλοντικές έρευνες.

Ακολουθεί ο Κατάλογος Αναφορών και βιβλιογραφίας με τις επιστημονικές πηγές που χρησιμοποιήθηκαν στη διατριβή.

ΜΕΡΟΣ ΠΡΩΤΟ

ΘΕΩΡΗΤΙΚΟ ΠΛΑΙΣΙΟ

ΚΕΦΑΛΑΙΟ 1^ο

Θεωρητικό Υπόβαθρο

1.1 Τεχνητά Νευρωνικά Δίκτυα

Τα τεχνητά νευρωνικά δίκτυα είναι υπολογιστικά μοντέλα που εμπνέονται από την αρχιτεκτονική των βιολογικών νευρικών δικτύων (Tavanaei et al., 2019). Η κύρια δομική τους μονάδα είναι ο τεχνητός νευρώνας, ο οποίος λαμβάνει πολλαπλές εισόδους (συνήθως πραγματικούς αριθμούς) με διάφορα βάρη, κάνει έναν υπολογισμό και εκπέμπει μια έξοδο

$$y = f(\sum w_i \cdot x_i + b),$$

όπου f είναι μια μη λογαριθμική συνάρτηση ενεργοποίησης και b ένας σταθμιστικός όρος (bias). Σε ένα πολύ-επίπεδο δίκτυο, οι έξοδοι πολλών νευρώνων ενός επιπέδου μετατρέπονται σε inputs για τους νευρώνες του επόμενου. Συχνά χρησιμοποιούμενες συναρτήσεις ενεργοποίησης στα ANNs είναι η sigmoid, η tanh και πιο πρόσφατα η ReLU (Rectified Linear Unit), η οποία εισάγει μη-γραμμικότητα.

Η εκπαίδευση ενός ANN είναι ένας αλγόριθμος και γίνεται μέσω backpropagation (οπίσθιας διάδοσης) μαζί με μεθόδους βελτιστοποίησης όπως ο στοχαστικός καθοδικός βαθμός (SGD) και άλλοι (π.χ. Adam). Σε ένα πρόβλημα ταξινόμησης εικόνων, το δίκτυο παίρνει τις τιμές των pixels της εικόνας ως είσοδο και βγάζει πίσω προβλέψεις για το σε ποια κατηγορία ανήκει. Η συνάρτηση κόστους (π.χ. Σταυρωμένη Εντροπία) συγκρίνει την πρόβλεψη με την πραγματική ετικέτα και υπολογίζει το σφάλμα. Με τη συνέχεια, το backpropagation μεταδίδει το σφάλμα πίσω στο δίκτυο και αλλάζει τα βάρη έτσι ώστε το δίκτυο να μαθαίνει να μειώνει το σφάλμα.

Με αυτόν τον τρόπο, τα ANNs έχουν φτάσει το state-of-the-art σε πάρα πολλές εφαρμογές της όρασης των υπολογιστών, όπως η αναγνώριση χειρόγραφων ψηφίων, η ταξινόμηση αντικειμένων σε εικόνες, η ανάλυση του βίντεο κ.ά. (Tavanaei κ.ά., 2019)

Παρά την επιτυχία τους, όπως αναφέρεται, τα τεχνητά νευρωνικά δίκτυα έχουν μεγάλη κατανάλωση ενέργειας και κατά την ανάπτυξη τους και κατά τη λειτουργία τους. Κάθε νευρώνας κάνει διαρκώς κινητές υπολογιστικές πράξεις, ενώ σε βαθιά δίκτυα με εκατομμύρια παραμέτρους οι απαιτήσεις για μνήμη και υπολογισμούς είναι τεράστιες. Για παράδειγμα, τα νευρωνικά δίκτυα αντιλαμβάνονται κάθε είσοδο (εικόνα) ως ένα στατικό διάνυσμα αριθμών και κάνουν πολλές πράξεις ακόμα και για περιοχές της εικόνας όπου δεν υπάρχει ιδιαίτερα σημαντική πληροφορία. Αυτό ανοίγει τον δρόμο για νέες αρχιτεκτονικές που θα εκμεταλλεύονται την αραιότητα των δεδομένων. Ακριβώς αυτόν τον στόχο έχουν οι Νευρωνικά δίκτυα Όσεων (Eshraghian et al., 2021).

Επιπλέον, τα ANNs είναι βασισμένα σε σταθερές ενεργοποιήσεις που δεν αλλάζουν στο χρόνο, γεγονός που σημαίνει ότι δεν μπορούν να λάβουν υπόψη τους τη χρονική εξέλιξη των δεδομένων, σε αντίθεση με ό,τι συμβαίνει στον ανθρώπινο εγκέφαλο. Όπως επισημαίνουν οι Tavanaei κ.ά. (2019), η έλλειψη φυσικής χρονικής διάστασης στα ANNs καθιστά αδύνατη τη μοντελοποίηση των δυναμικών αλληλεπιδράσεων που εμφανίζονται στο νευρικό σύστημα. Αυτό το κενό γεφυρώνουν τα SNNs, τα οποία προσπαθούν να ξεπεράσουν την απόκλιση μεταξύ υπολογιστικής ισχύος και βιολογικής ακρίβειας.

1.2 Νευρωνικά Δίκτυα Όσεων

Τα Spiking Neural Networks είναι πιο βιολογικά ακριβή μοντέλα, όπου η επικοινωνία μεταξύ νευρώνων γίνεται με χρονικά καθορισμένες ακίδες, αντί για συνεχή σήματα. (Tavanaei et al., 2019; Eshraghian et al., 2021)

Ο κάθε τεχνητός νευρώνας-ώσης ενσωματώνει έναν δυναμικό μηχανισμό που μοιάζει με τη συμπεριφορά της κυτταρικής μεμβράνης των βιολογικών νεύρων. Συγκεκριμένα, ένα απλό μοντέλο είναι ο Ολοκληρωτής-Πυροδότης (Integrate-and-Fire): ο νευρώνας δέχεται γεγονότα/ άξονες από προϋπάρχοντες νευρώνες, τα οποία αυξάνουν το δυναμικό της μεμβράνης του. Όταν το δυναμικό φθάσει πάνω από ένα ορισμένο όριο (κατώφλι), ο νευρώνας εκπέμπει μια ακίδα (spike) στην έξοδό του και ταυτόχρονα επαναφέρεται (reset) σε χαμηλότερο δυναμικό. Ένα πιο ρεαλιστικό σενάριο είναι το Leaky Integrate-and-Fire (LIF) μοντέλο, όπου το δυναμικό παρουσιάζει εκθετική διαρροή με το πέρασμα του χρόνου: αν δεν υπάρχουν άλλες ακίδες, το δυναμικό μειώνεται σταδιακά προς μια τιμή ηρεμίας.

Με μαθηματικούς όρους, η εξέλιξη του δυναμικού $V(t)$ ενός LIF νευρώνα μπορεί να περιγραφεί από μια διαφορική εξίσωση της μορφής:

$$\tau \frac{dV}{dt} = -V + I(t)$$

όπου τ είναι η χρονική σταθερά (χαρακτηρίζει τον ρυθμό διαρροής) και $I(t)$ το συνεισφερόμενο ρεύμα από εισερχόμενες ακίδες εκείνη τη στιγμή. Κάθε εισερχόμενη ακίδα αυξάνει στιγμιαία το δυναμικό κατά ένα ποσό (ανάλογο του βάρους της σύναψης). Όταν $V(t)$ υπερβεί το κατώφλι V_{thr} , τότε ο νευρώνας παράγει μια ακίδα εξόδου και το δυναμικό του επανέρχεται σε μια βασική τιμή (reset). Στο διακριτό ανά βήματα μοντέλο που συχνά χρησιμοποιείται σε υπολογισμούς, το παραπάνω μεταφράζεται σε αναδρομική σχέση:

$$V[t + 1] = \beta V[t] + \sum_i w_i x_i[t] - z[t] V_{thr},$$

όπου $0 < \beta < 1$ συνδέεται με τη σταθερά διαρροής ($\beta = e^{dt/\tau}$ στο συνεχές μοντέλο), $x_i[t] \in \{0,1\}$ είναι οι δυαδικές είσοδοι (ακίδες από i -οστό προσυναπτικό νευρώνα) στο βήμα t , w_i τα αντίστοιχα βάρη, και $z[t]$ η δυαδική έξοδος του νευρώνα (ακίδα: 1 αν $V[t]$ πέρασε το κατώφλι,

αλλιώς 0). Η τελευταία αφαίρεση $-z[t]V_{thr}$ εφαρμόζει την επαναφορά (reset) μετά από μια ακίδα.

Η παραπάνω λειτουργία παρουσιάζει τον τρόπο λειτουργίας ενός νευρώνα όσης, ο οποίος δέχεται, επεξεργάζεται και στέλνει τα ερεθίσματα (spikes) τόσο ως χρονικός ολοκληρωτής (integrator) όσο και ως πυροδότης των ακίδων. Τα SNN, τα οποία αποτελούνται από αυτούς τους νευρώνες, επεξεργάζονται βάσει των spikes που διαδίδονται στο δίκτυο. Άρα, η πληροφορία συνήθως κωδικοποιείται ταυτόχρονα με τον ρυθμό ή/και τον ακριβή χρόνο εμφάνισής της, όπως θα δούμε αμέσως.

Ένα από τα πιο σημαντικά πλεονεκτήματα των SNN είναι η τεράστια δυνατότητά τους να εξοικονομούν ενέργεια, καθώς, κατά μέσο όρο, κάθε νευρώνας εκπέμπει λιγότερες ακίδες (έναντι συνεχών σημάτων). Έτσι, πολλοί υπολογισμοί μπορούν να πραγματοποιηθούν με αποδοτικό τρόπο σε ειδικές αρχιτεκτονικές (γνωστές και ως event-driven). Πράγματι, σε ένα καλά "ρυθμισμένο" SNN, πολλοί νευρώνες παραμένουν "σιωπηλοί" (δεν εκπέμπουν) για μεγάλο χρονικό διάστημα και μόνο λίγοι "πυροδοτούντε" όταν υπάρχει αιχμή.

Σύμφωνα με τον Eshraghian et al., (2023), η sparsity (σπανιότητα ή αραιότητα ενεργοποίησης νευρώνων) των SNNs έχει ως αποτέλεσμα τη μείωση της ενέργειας που καταναλώνουν, γεγονός που τα καθιστά ιδιαίτερα κατάλληλα για χρήση σε τομείς με χαμηλή κατανάλωση ισχύος, όπως το IoT και τα συστήματα υπολογισμού αιχμής (edge computing), όπου οι υπολογιστικές μονάδες βρίσκονται κοντά στην πηγή των δεδομένων. Επιπλέον, όπως επισημαίνουν οι συγγραφείς, η λειτουργία των SNNs ως μοντέλα που ενεργοποιούνται από γεγονότα (event-driven models) συμβάλλει περαιτέρω στη βελτίωση της ενεργειακής τους απόδοσης, καθώς οι υπολογισμοί πραγματοποιούνται μόνο όταν υπάρχουν εισερχόμενα spikes.

Ερευνητές έχουν αποδείξει ότι με μέσο ρυθμό αιχμών 0.3 ανά νευρώνα, μπορούν να πετύχουν εξαιρετικές επιδόσεις σε προβλήματα ταξινόμησης σε ένα βαθύ SNN, γεγονός που σημαίνει ότι αυτοί οι τύποι δικτύων καταναλώνουν πολύ λιγότερη ενέργεια σε σύγκριση με τα παραδοσιακά. (Eshraghian et al., 2023)

Ωστόσο, το μεγάλο ερώτημα είναι πώς θα εκπαιδεύσουμε τα SNN. Οι κλασικές τεχνικές μηχανικής μάθησης που βασίζονται στον υπολογισμό των παραγώγων (όπως το backpropagation) δεν είναι άμεσα εφαρμόσιμες σε αυτό το πλαίσιο, επειδή η συνάρτηση εκπομπής spike έχει την ιδιότητα ότι η παράγωγός της είναι μηδέν παντού εκτός από ένα σημείο ασυνέχειας. Αυτό σημαίνει ότι δεν μπορούμε να υπολογίσουμε αναλυτικά τον ακριβή τρόπο με τον οποίο μια μικρή αλλαγή στα βάρη επηρεάζει την έξοδο – με άλλα λόγια, δεν υπάρχει τοπικός gradient (δηλαδή παραγώγος σε επίπεδο νευρώνα/βάρους) για να εκμεταλλευτούμε στην εκπαίδευση.

Αυτό το πρόβλημα έχει αντιμετωπιστεί με διάφορους τρόπους: αρχικά, χρησιμοποιήθηκαν βιολογικώς-εμπνευσμένοι κανόνες πλαστικότητας, όπως ο κανόνας STDP (Tavanaei et al., 2019), για την αναστροφή των αλλαγών στα βάρη, ή ακόμα και προσεγγίσεις όπου ένα ANN εκπαιδεύεται πρώτα και στη συνέχεια μετατρέπεται σε SNN.

1.3 Συσχέτιση SNNs με Επαναληπτικά Δίκτυα (RNNs)

Μια πρωτότυπη προσέγγιση για να κατανοήσουμε καλύτερα τον τρόπο λειτουργίας των SNN είναι να τα συσχετίσουμε με τα RNN (Recurrent Neural Networks). Σύμφωνα με το Neftci και συνεργάτες (2019), τα SNN έχουν ενσωματωμένη χρονική εξάρτηση, καθώς η κατάσταση ενός νευρώνα (π.χ. το δυναμικό του κυττάρου) εξαρτάται από το ιστορικό των εισόδων του. Το χαρακτηριστικό αυτό τους τα κάνει παρόμοια με τα RNNs που, κάθε χρονική στιγμή, παράγουν έξοδο η οποία εξαρτάται από την τρέχουσα είσοδο και την εσωτερική τους κατάσταση.

Αυτός ο τρόπος σκέψης μας επιτρέπει να εφαρμόσουμε στα SNN, για παράδειγμα, τη μέθοδο οπισθό διάδοσης μέσω του χρόνου (Backpropagation Through Time – BPTT) που χρησιμοποιούνται συνήθως στα RNN. Όπως δηλώνουν οι Eshraghian και συνεργάτες (2023), τα μοντέλα LIF, όταν υλοποιούνται σε διακριτό χρόνο, μπορούν να χρησιμοποιηθούν σε υπολογιστικά γραφήματα και να εκπαιδευτούν όπως τα επαναλαμβανόμενα μοντέλα, ανοίγοντας νέες προοπτικές για ισχυρές βελτιστοποιήσεις και επεκτάσεις των εφαρμογών τους σε ποικίλα προβλήματα, όπως μοντελοποίηση και επεξεργασία δεδομένων με χρονική δομή.

1.4 Ο κανόνας STDP

Ο κανόνας Spike-Timing-Dependent Plasticity (STDP) είναι ένα βιολογικά εμπνευσμένο θεμελιώδες μοντέλο πλαστικότητας, το οποίο διέπει τη μεταβολή της συναπτικής ισχύος βάσει της χρονικής σχέσης μεταξύ των πυροδοτήσεων (spikes) του προσυναπτικού και του μετασυναπτικού νευρώνα. Αν ένα προσυναπτικό spike εμφανιστεί πριν από ένα μετασυναπτικό spike, εντός ενός συγκεκριμένου χρονικού παραθύρου, τότε η σύνδεση ενισχύεται (φαινόμενο μακροπρόθεσμης ενίσχυσης – Long-Term Potentiation, LTP). Αντίθετα, εάν το μετασυναπτικό spike προηγηθεί του προσυναπτικού, η σύνδεση αποδυναμώνεται (φαινόμενο μακροπρόθεσμης καταστολής – Long-Term Depression, LTD).

Η μαθηματική έκφραση του κανόνα STDP δίνεται ως εξής:

$$\Delta w = \begin{cases} Ae^{\frac{-(t_{pre} - t_{post})}{\tau}} & t_{pre} - t_{post} \leq 0, \quad A > 0 \\ Be^{\frac{-(t_{pre} - t_{post})}{\tau}} & t_{pre} - t_{post} > 0, \quad B < 0 \end{cases}$$

Εικόνα 2: Μαθηματική έκφραση STDP, Πηγή[1]

όπου:

- **Δw** : μεταβολή της συναπτικής ισχύος
- **t_{pre}, t_{post}** : χρονικές στιγμές πυροδότησης προσυναπτικού και μετασυναπτικού νευρώνα
- **A, B** : συντελεστές ενίσχυσης και αποδυνάμωσης αντίστοιχα
- **τ** : χρονική σταθερά

Το STDP είναι ένα είδος τοπικής μάθησης που δεν χρειάζεται γενικευμένο σφάλμα, γεγονός που το καθιστά έναν ιδανικό συνεργάτη των neuromorphic υπολογιστικών συστημάτων. Έχει εφαρμοστεί επιτυχώς τόσο σε unsupervised learning μοντέλα όσο και σε βιολογικά μοντέλα του εγκεφάλου. (βλ. Tavanaei et al., 2019, Eshraghian et al., 2023)

Επιπλέον, σύμφωνα με τους Tavanaei και συν. (2019), ο κανόνας STDP δεν αποτελεί απλώς έναν τρόπο για να συσχετίζουμε τους χρονικούς παλμούς, αλλά μπορεί να θεωρηθεί και ως ένα είδος πιθανοκρατικής μάθησης. Ανάλογα, δηλαδή, με τις συνθήκες, ο κανόνας STDP μπορεί να προσεγγίσει αλγόριθμους τύπου expectation-maximization για την εκμάθηση των latent μεταβλητών.

Ο κανόνας STDP μπορεί να ερμηνευτεί ως μια προσέγγιση τύπου expectation-maximization (EM), όπου το δίκτυο εκτελεί εναλλασσόμενες εκτιμήσεις "προσδοκίας" (estimation-expectation) και "μεγιστοποίησης" (maximization) για την εκμάθηση λανθανουσών μεταβλητών (latent variables), δηλαδή μεταβλητών που δεν παρατηρούνται άμεσα, όπως οι υποκείμενες ή βαθιές αιτίες μιας εισόδου. Στο πλαίσιο αυτό, η πλαστικότητα των συνάψεων, η οποία εξαρτάται από τη χρονική σειρά των spikes, λειτουργεί ως ένας μηχανισμός σταδιακής προσαρμογής των βαρών με βάση την πιθανοκρατική συσχέτιση εισόδου-εξόδου.

Επιπλέον, έχει προταθεί ότι οι SNNs εκτελούν Bayesian inference (Μπεϋζιανή συμπερασματολογία), δηλαδή υπολογίζουν πιθανολογικά συμπεράσματα βάσει των εισερχόμενων spikes και των συνάψεων, μοντελοποιώντας την αβεβαιότητα στις εισόδους και τα βάρη. Αυτή η προσέγγιση συνδέεται στενά με τη βιολογική θεώρηση της αντίληψης και της μάθησης, σύμφωνα με την οποία ο εγκέφαλος λειτουργεί ως ένας προβλεπτικός μηχανισμός που ενημερώνει συνεχώς τις γνώσεις του βάσει των νέων δεδομένων.

Ωστόσο, ένα από τα κύρια προβλήματα που αντιμετωπίζουν οι SNN στην εκπαίδευσή τους είναι ότι η συνάρτηση spike δεν είναι διαφορίσιμη (η παράγωγός της είναι σχεδόν παντού μηδέν), κάτι που καθιστά την παραδοσιακή μέθοδο της οπισθοδιάδοσης (backpropagation) αδύνατη να εφαρμοστεί. Για να ξεπεραστεί αυτό το πρόβλημα, χρησιμοποιούνται οι λεγόμενες surrogate gradient functions (προσεγγιστικές υποκατάστατες συναρτήσεις κλίσης), οι οποίες παρέχουν μία ομαλή προσέγγιση της spike συνάρτησης μόνο κατά το βήμα της οπισθοδιάδοσης, ενώ ταυτόχρονα διατηρούν τη

βιολογικά ρεαλιστική συμπεριφορά του δικτύου κατά την εμπρός διάδοση του σήματος.
(Neftci et al., 2019, Eshraghian et al., 2023)

1.5 Surrogate Gradients και Κωδικοποίηση Ακίδων

Μια surrogate gradient συνάρτηση είναι μία προσέγγιση της παραγώγου, η οποία χρησιμοποιείται για να αντικαταστήσει την πραγματική παράγωγο της μη διαφοροποιήσιμης κατά την οπίσθια διάδοση. Αλλιώς, κατά τη διάρκεια του εμπρόσθιου περάσματος του νευρωνικού (forward pass), κάθε spike εξακολουθεί να παρέχει ακριβείς χωρικές πληροφορίες (π.χ. εκπέμπει δυαδικό spike αμέσως μόλις η τάση ξεπεράσει το κατώφλι). Αντίθετα, κατά την οπίσθια διάδοση του αλγορίθμου εκπαίδευσης, αντικαθιστούμε την παράγωγο που είναι μηδέν στην αρχική συνάρτηση, με μία διαφορετική, μη-μηδενική συνάρτηση, η οποία να "μοιάζει" όσο το δυνατόν περισσότερο με την αρχική. Με αυτόν τον τρόπο, αποφεύγεται το πρόβλημα των μηδενικών παραγώγων κατά την οπίσθια διάδοση (backpropagation), το οποίο καθιστά αδύνατη την εκπαίδευση μέσω κλασικών τεχνικών βελτιστοποίησης. Η αντικατάσταση αυτή επιτρέπει την υπολογισμό των παραγώγων και την ενημέρωση των βαρών με τεχνικές όπως ο Adam optimizer (βελτιστοποιητής Adam).

Ο Adam (Adaptive Moment Estimation) είναι ένας αλγόριθμος εκπαίδευσης που προσαρμόζει δυναμικά τον ρυθμό μάθησης για κάθε παράμετρο ξεχωριστά. Αυτό το κάνει υπολογίζοντας και συνδυάζοντας το κινητό μέσο (momentum) και τη διακύμανση (variance) των παραγώγων κατά τη διάρκεια της εκπαίδευσης. Λόγω αυτής της ιδιότητας, επιτυγχάνει πιο σταθερή και ταχύτερη σύγκλιση, ακόμη και σε δύσκολα ή θορυβώδη προβλήματα, όπως η εκπαίδευση SNNs

Η χρήση surrogate gradient καθιστά εφικτή την εφαρμογή της backpropagation ακόμη και σε δίκτυα που περιλαμβάνουν μη διαφορίσιμες συναρτήσεις ενεργοποίησης, όπως τα spikes. Έτσι, τα SNN μπορούν να συνδυάσουν την υψηλή αποδοτικότητα στον υπολογισμό και τον ρεαλισμό με την ικανότητα εκπαίδευσης των κλασικών ANNs. (Neftci et al., 2019)

Αυτή η έξυπνη "συνέχεια" μας επιτρέπει να εκπαιδεύσουμε deep SNN άμεσα μέσω supervised learning και έχει αποδειχθεί ιδιαίτερα επιτυχημένη, καθώς θεωρείται ως η βασική αιτία για την πρόσφατη ικανότητα των SNN να φτάνουν στα επίπεδα των ANNs.

Υπάρχουν διάφοροι τρόποι να προσεγγίσουμε τη συνάρτηση παράγωγου που χρησιμοποιούμε. Αρκεί να είναι μία συνεχής συνάρτηση που να προσεγγίζει με απότομο τρόπο το κατώφλι ενεργοποίησης. Οι πιο δημοφιλείς επιλογές (που θα ασχοληθούμε και εμείς στα πειράματά μας) είναι οι ακόλουθες:

- 1) **Σιγμοειδής (Sigmoid):** Χρησιμοποιούμε την παράγωγο της λογιστικής συνάρτησης $\sigma(x) = 1/(1 + e^{-x})$ που δίνει μια ωραία ομαλή καμπύλη γύρω από το 0 (αν θέσουμε το κατώφλι στο 0). Η παράγωγος $\sigma'(x)$ έχει μέγιστη τιμή στο 0 και μετά φθίνει συμμετρικά.
- 2) **Γρήγορη Σιγμοειδής (Fast Sigmoid):** Παραλλαγή της sigmoid που έχει μεγαλύτερη κλίση (slope) κοντά στο μηδέν έτσι ώστε να προσεγγίζει πιο εύκολα την απότομη αλλαγή. Π.χ. η συνάρτηση $f(x) = \frac{1}{1+e^{ax}}$ έχει μεγαλύτερο a δίνοντας πιο "ψυλή" αλλά όχι τόσο "απότομη" καμπύλη.
- 3) **Αρκταντ (ArcTan):** Χρησιμοποιούμε την παράγωγο της ή κάποια κλιμακωμένη έκδοσή της. Η αυξάνει λίγο πιο αργά στα άκρα από τη sigmoid, έτσι ώστε να έχουμε ένα σχετικά ήπιο surrogate gradient.
- 4) **SpikeRateEscape:** Αυτή η συνάρτηση χρησιμοποιεί μία παραλλαγή της εκθετικής εξομάλυνσης που δίνει gradient της μορφής:

$$\frac{1}{\beta + e^{-a(x-\theta)}}$$

Όταν το δυναμικό πλησιάζει το κατώφλι, η εκθετική εξομάλυνση λειτουργεί ομαλά για να εξομαλύνει τη μετάβαση. Η ένωση των β και α μας επιτρέπει να ελέγχουμε το “πόσο απότομη” θα είναι η μετάβαση. Αυτό το μοντέλο είναι εμπνευσμένο από βιολογικά spikes.

Κάθε μία από αυτές τις επιλογές έχει ξεχωριστά χαρακτηριστικά. Συχνά, υπάρχει ένας συμβιβασμός μεταξύ του να προσεγγίζει με ακρίβεια την πραγματική σχέση (π.χ. μια απότομη γραφική που κατεβαίνει κάθετα είναι πιο κοντά στο ιδανικό βήμα) και του εύρους τιμών του dynamic range στο οποίο η παράγωγος είναι μηδενική (effective gradient domain). Έτσι, μια πολύ απότομη (στενή) υποκατάστατη παράγωγος μπορεί να είναι μη μηδενική μόνο σε ένα πολύ στενό εύρος γύρω από το κατώφλι της – αυτό σημαίνει ότι αν οι περισσότεροι νευρώνες δεν είναι πολύ κοντά στο threshold, τότε δεν θα έχουν καθόλου gradient και θα παραμένουν “νεκροί”. Αντίθετα, μια πιο “ήπια” (ή, αν θέλετε, με μικρότερη κλίση) παράγωγος (όπως, για παράδειγμα, η $\arctan$ ή η sigmoid που έχει μικρότερη κλίση) σηματοδοτούν λάθη σε ένα πιο ευρύ εύρος τιμών της, αλλά βοηθάει περισσότερους νευρώνες να “προσαρμοστούν”, με το κόστος όμως της λιγότερο ακριβούς προσέγγισης της πραγματικής λειτουργίας.

Σύμφωνα με τον Eshraghian και συν(2023), η κατάλληλη επιλογή της surrogate function είναι κρίσιμη για την ακρίβεια της ταξινόμησης και τη σταθερότητα της εκπαίδευσης. Οι ερευνητές τονίζουν ότι μια ισορροπία μεταξύ “απότομων” και “ηπίων” παραγώγων επιτρέπει την ενεργοποίηση του μεγαλύτερου αριθμού νευρώνων, διατηρώντας παράλληλα υψηλή την ακρίβεια. Αυτό έχει οδηγήσει σε βελτιώσεις ακόμη και σε δύσκολα σύνολα δεδομένων, όπως το CIFAR-10.

Στην πράξη, οι Fast Sigmoid και οι ATan συναρτήσεις που έχουν προταθεί στη βιβλιογραφία (Neftci et al., 2019; Eshraghian et al., 2023) έχουν θεωρηθεί ιδιαιτέρως αποτελεσματικές, καθώς προσφέρουν σχετικά μεγάλο εύρος gradient χωρίς όμως να εμφανίζεται το πρόβλημα των “τελείως ανενεργών” νευρώνων.

Πέρα από τον ίδιο το σχεδιασμό της αρχιτεκτονικής και τον αλγόριθμο εκπαίδευσης, το σχήμα της κωδικοποίησης των δεδομένων σε ακίδες παίζει ζωτικό ρόλο στα SNN. Εφόσον τα SNN επεξεργάζονται δυαδικά γεγονότα, πρέπει να αναμορφώσουμε τα κανονικά δεδομένα (π.χ. εικόνες που αποτελούνται από πίνακες με *pixel*) σε χρονικές ακολουθίες ακίδων. Υπάρχουν διάφορες τεχνικές κωδικοποίησης:

Κωδικοποίηση Ρυθμού (Rate Coding): Η πληροφορία κωδικοποιείται ανάλογα με τον ρυθμό πυροδότησης (συχνότητα) των ακίδων. Για παράδειγμα, ένα φωτεινό *pixel* μπορεί να αντιπροσωπεύεται από μία είσοδο που πυροδοτεί πολλές ακίδες κατά μήκος ενός συγκεκριμένου χρονικού παραθύρου, ενώ ένα σκοτεινό *pixel* αντιστοιχεί σε σπάνιες ή μηδαμινές ακίδες κατά το ίδιο διάστημα. Άρα, η μέση συχνότητα των ακίδων μεταφέρει την αντίστοιχη τιμή.

Χρονική Κωδικοποίηση (Temporal Coding): Η πληροφορία κωδικοποιείται μέσω της ακριβούς χρονικής στιγμής εμφάνισης των ακίδων, όχι μέσω του αριθμού τους. Ένα συνηθισμένο είδος είναι η κωδικοποίηση καθυστέρησης (*latency coding*). Για παράδειγμα, μπορούμε να θέσουμε ότι ο κάθε νευρώνας αντιστοιχεί σε μία ακίδα και η χρονική καθυστέρηση μέχρι να πυροδοτηθεί σχετίζεται με την τιμή του (μικρή καθυστέρηση για μεγάλη τιμή, μεγαλύτερη καθυστέρηση για μικρή τιμή). Έτσι, η σειρά και το χρονικό διάστημα μεταξύ των ακίδων μεταφέρουν την πληροφορία. Ένα ειδικό είδος είναι το *Time-To-First-Spike (TTFS)*, όπου κάθε νευρώνας εκπέμπει το πολύ μία ακίδα – η χρονική στιγμή της πρώτης (και μοναδικής) ακίδας του αντιστοιχεί στο σήμα. Έχει δείξει ότι με τέτοια σχήματα θεωρητικά τα SNN μπορούν να υπολογίζουν με την ίδια ακρίβεια με ένα ANN με ReLU νευρώνες και, ταυτόχρονα, να καταναλώνουν ελάχιστη ενέργεια (λιγότερη από μία ακίδα ανά νευρώνα).

Κωδικοποίηση Πληθυσμού (Population Coding): Εδώ, η πληροφορία της μιας μεταβλητής κατανέμεται στην κοινή δραστηριότητα πολλών νευρώνων. Μια συγκεκριμένη ακολουθία ακίδων σε ένα σύνολο νευρώνων, για παράδειγμα, μπορεί να αντιστοιχεί σε μια τιμή εισόδου. Αυτός ο τρόπος κωδικοποίησης είναι ακόμη περίπλοκος και συνήθως χρησιμοποιείται μαζί με τους παραπάνω (π.χ. οι νευρώνες να μοιράζονται την αναπαράσταση ενός αριθμού, έτσι ώστε να μειωθεί ο θόρυβος).

Στη πράξη, για πολλά προβλήματα, ιδιαίτερα όταν τα δεδομένα προέρχονται από συσκευές event-based (νευρομορφικές κάμερες κ.λπ.), προτιμάται ο απλούστερος συνήθως τρόπος περιγραφής τους σε μορφή ακίδων: υποθέτουμε ότι σε ένα μεγάλο χρονικό παράθυρο, το νούμερο των ακίδων αντανakλά μια συνεχή τιμή. Και η απόφαση, για παράδειγμα, για το ποια κατηγορία ανήκει ένα δείγμα, συχνά βασίζεται στο ποιος νευρώνας έχει το μεγαλύτερο ρυθμό ακίδων καθ' όλη τη διάρκεια της παρουσίασης του ερεθίσματος. Αυτό ονομάζεται ταξινόμηση με χρήση αναλογίας: κάθε κατηγορία αντιστοιχεί σε ένα νευρώνα εξόδου και το δίκτυο "μαθαίνει" να ενεργοποιεί περισσότερο τους νευρώνες της σωστής κατηγορίας. Διαφορετικά, σε ένα σενάριο χρονικής κωδικοποίησης, η απόφαση θα μπορούσε να βασίζεται στον νευρώνα που "έβγαλε" πρώτος ακίδα (TTFS ταξινόμηση). Στην παρούσα μας έρευνα, η μέθοδος της κωδικοποίησης ρυθμού και η μέθοδος της καταμέτρησης των ακίδων θα χρησιμοποιηθούν, διότι το σύνολο δεδομένων N-MNIST παρέχεται βασικά με τις εισόδους σε μορφή ακολουθίας ακίδων και η έξοδος θα προκύψει από την καταμέτρησή τους. Να σημειωθεί ότι το N-MNIST, αν και περιέχει χρονική πληροφορία, μπορεί σε μεγάλο βαθμό να ταξινομηθεί μόνο με βάση τη χωρική πληροφορία. Αυτό υποδηλώνει ότι κυρίως ο ρυθμός των γεγονότων (αριθμός ακίδων) είναι αρκετός για την ταξινόμηση των ψηφίων και πράγματι πολλά SNN μοντέλα αντιμετωπίζουν το N-MNIST με βάση τις μετρήσεις των ακίδων τους παρά με βάση τη χρονική ακολουθία τους.

Τέλος, να σημειωθεί ότι, παρά τις αντιθέσεις τους, τα SNN μπορούν να κάνουν τα ίδια πράγματα με τα παραδοσιακά δίκτυα, αν γίνουν τα κατάλληλα SNN. Πρόσφατες έρευνες έδειξαν ότι ένα SNN που έχει εκπαιδευτεί βαθιά μπορεί να πετύχει την ίδια ακρίβεια με ένα ANN, χρησιμοποιώντας γνωστές βάσεις δεδομένων εικόνων (π.χ. MNIST, CIFAR10/100) και αποδεικνύοντας έτσι ότι οι θεωρητικοί περιορισμοί μπορούν να ξεπεραστούν. Σε αυτές τις περιπτώσεις, το SNN έχει, επιπλέον, το πλεονέκτημα ότι μπορεί να υλοποιηθεί σε ειδικό υλικό που καταναλώνει λιγότερη ενέργεια, κάτι που το καθιστά ιδανικό για εφαρμογές edge και IoT, όπου η χαμηλή κατανάλωση ενέργειας και η αυτόνομη λειτουργία είναι ζωτικής σημασίας.

ΚΕΦΑΛΑΙΟ 2°

Σύγχρονες Εφαρμογές SNNs για Ανάλυση Εικόνας (State of the Art)

2.1 Εισαγωγή

Η εφαρμογή των Spiking Neural Networks (SNNs) σε προβλήματα ανάλυσης εικόνας είναι ένα αναπτυσσόμενο πεδίο έρευνας που βασίζεται στη βιολογική έμπνευση και την ενεργειακή αποδοτικότητα των SNNs για να ανταγωνιστεί τα παραδοσιακά τεχνητά νευρωνικά δίκτυα (ANNs). Τα Convolutional Neural Networks (CNNs) έχουν κατακτήσει τον χώρο της υπολογιστικής όρασης, αλλά η έντονη χρήση υπολογιστικών πόρων και ενέργειας αποτελεί περιορισμό σε edge(αιχμής) και real-time(πραγματικού χρόνου) εφαρμογές.

Τα SNNs εισάγουν μια θεμελιώδη αλλαγή στον τρόπο με τον οποίο γίνεται η αναπαράσταση και η επεξεργασία της πληροφορίας, χρησιμοποιώντας γεγονότα (events) και spikes αντί για συνεχείς τιμές. Η χρήση surrogate gradients και η υιοθέτηση αρχιτεκτονικών που είναι εμπνευσμένες από τα CNNs έχουν βοηθήσει τα SNNs να πετύχουν εντυπωσιακές επιδόσεις σε διάφορα datasets εικόνας, όπως τα MNIST, CIFAR-10 και N-MNIST.

Σε αυτό το κεφάλαιο παρουσιάζονται πρόσφατες έρευνες που χρησιμοποιούν SNNs για ταξινόμηση εικόνων, αποθορυβοποίηση, segmentation (τμηματοποιήσεις), καθώς και ανάλυση ιατρικών εικόνων. Για κάθε έρευνα αναλύονται η αρχιτεκτονική του SNN, τα datasets που χρησιμοποιήθηκαν, οι επιδόσεις και τα πλεονεκτήματα σε σχέση με τα παραδοσιακά CNNs. Ο στόχος αυτής της ανασκόπησης είναι να αναδείξει την τρέχουσα κατάσταση της έρευνας και να εντοπίσει τις προκλήσεις και τις μελλοντικές κατευθύνσεις.

2.2.1 Sa-SNN: Spiking Attention Network για Αναγνώριση Εικόνων

Μια από τις πλέον σημαντικές εξελίξεις στον τομέα των SNNs είναι η ενσωμάτωση μηχανισμών προσοχής (attention) προκειμένου να βελτιστοποιηθεί η επιλογή των χαρακτηριστικών κατά τη διαδικασία της επεξεργασίας των εισόδων. Ένα εξαιρετικό παράδειγμα αυτής της προσπάθειας αποτελεί το μοντέλο Sa-SNN (Spiking Attention Neural Network), το οποίο εισάγει την έννοια της προσοχής στον τομέα των όσεων (spike) (Dan et al., 2024).

Το βασικό εγχείρημα του Sa-SNN είναι η δημιουργία ενός μηχανισμού που εστιάζει σε συγκεκριμένα κανάλια χαρακτηριστικών, χρησιμοποιώντας ένα Spiking-Efficient Channel Attention (SECA) module. Το module αυτό υπολογίζει τα βάρη προσοχής βασιζόμενο στα χαρακτηριστικά των καναλιών (π.χ. μέσος όρος, μέγιστη τιμή) και ενεργοποιεί εκείνα που μεταφέρουν σημαντικές πληροφορίες. Με αυτό τον τρόπο, πετυχαίνεται σημαντική αραίωση στην ενεργοποίηση των νευρώνων, πράγμα που σημαίνει ότι η ενεργειακή απόδοση του δικτύου βελτιώνεται αισθητά.

Το δίκτυο αξιολογήθηκε με επιτυχία χρησιμοποιώντας διάφορα datasets, όπως τα MNIST, Fashion-MNIST και N-MNIST, επιτυγχάνοντας εξαιρετικά αποτελέσματα:

Dataset	Ακρίβεια Sa-SNN (%)
MNIST	99.61
Fashion-MNIST	94.13
N-MNIST	99.63

Η ακρίβεια αυτή είναι συγκρίσιμη με εκείνη των CNNs, ενώ η ενεργειακή κατανάλωση είναι αισθητά χαμηλότερη, κάτι που καθιστά το Sa-SNN κατάλληλο για εφαρμογές edge computing και IoT.

Αξιοσημείωτο είναι ότι το Sa-SNN διατηρεί την υψηλή απόδοση ακόμη και όταν μειώνεται ο αριθμός των spikes ανά δείγμα, στοιχείο που αποδεικνύει την αποδοτικότητα της αρχιτεκτονικής του.

2.2.2 Υβριδικά CNN-SNN Μοντέλα και Εκπαίδευση με Surrogate Gradients

Ένας από τους βασικούς τομείς έρευνας στα Spiking Neural Networks (SNNs) αφορά τον τρόπο συνδυασμού τους με τα επιτυχημένα μοντέλα των Συνελκτικών Νευρωνικών Δικτύων (Convolutional Neural Networks - CNNs). Τα υβριδικά μοντέλα CNN-SNN έχουν ως στόχο να εκμεταλλευτούν τα πλεονεκτήματα των CNNs όσον αφορά την εξαγωγή χαρακτηριστικών, προσφέροντας ταυτόχρονα την ενεργειακή αποδοτικότητα και το βιολογικό ρεαλισμό των SNNs (Neftci, Mostafa, & Zenke, 2019).

Σύμφωνα με αυτήν την προσέγγιση, τα πρώτα επίπεδα του δικτύου εκτελούν συνελίξεις (convolutions) με τον ίδιο τρόπο που γίνεται σε ένα κανονικό CNN, ενώ τα ακόλουθα επίπεδα μετατρέπουν τις ενεργοποιήσεις σε ακίδες (spikes), χρησιμοποιώντας spiking layers. Το υβριδικό δίκτυο εκπαιδεύεται end-to-end (τέλους-προς-τέλος) με τη χρήση μεθόδων surrogate gradient – όπως η fast sigmoid – οι οποίες επιτρέπουν τη διάδοση των σφαλμάτων ακόμη και μέσω μη διαφορίσιμων spike-generating συναρτήσεων.

Αυτή η προσέγγιση προσφέρει άριστα αποτελέσματα σε εφαρμογές αναγνώρισης εικόνας, εξοικονομώντας ταυτόχρονα ενέργεια. Σύμφωνα με ορισμένα πειράματα, η μείωση της κατανάλωσης ενέργειας μπορεί να φτάσει έως και 50% σε σχέση με τα συμβατικά CNNs, με ελάχιστες απώλειες στην ακρίβεια.

Η χρονική κωδικοποίηση (temporal coding) ενισχύει επιπλέον τη δυνατότητα του δικτύου να ανιχνεύει δυναμικά πρότυπα (temporal patterns), γεγονός που καθιστά τα υβριδικά μοντέλα ιδιαίτερα κατάλληλα για εφαρμογές νευρομορφικής επεξεργασίας σε πραγματικό χρόνο - όπως αναγνώριση κίνησης ή επεξεργασία βίντεο σε χαμηλής κατανάλωσης συστήματα.

Συνολικά, τα υβριδικά μοντέλα CNN-SNN έρχονται να γεφυρώσουν το χάσμα ανάμεσα στην ακρίβεια των deep learning μεθόδων και τη βιολογική αποδοτικότητα των SNNs, και αναμένεται να αποτελέσουν μια πολλά υποσχόμενη προσέγγιση για εφαρμογές σε ενσωματωμένα (embedded) και έξυπνα νευρομορφικά συστήματα (Neftci et al., 2019).

2.2.3 Νευρομορφική Υπολογιστική σε Κυτταρομετρία Ροής: Σύγχρονες Προσεγγίσεις

Η κυτταρομετρία ροής με απεικόνιση (Imaging Flow Cytometry - IFC) αντιπροσωπεύει μια προηγμένη αναλυτική τεχνική, η οποία ενσωματώνει τα πλεονεκτήματα της συμβατικής κυτταρομετρίας ροής και της μικροσκοπίας, παρέχοντας πληροφορίες για τόσο το μέγεθος όσο και το σχήμα των κυττάρων. Ωστόσο, η IFC παρουσιάζει δύο βασικές προκλήσεις: το πρόβλημα της χωρικής ανάλυσης λόγω της θόλωσης κατά τη διάρκεια της μετακίνησης των υψηλών ταχυτήτων σωματιδίων και τον όγκο των δεδομένων, ο οποίος απαιτεί μεγάλους χώρους αποθήκευσης και υπολογιστικούς πόρους για την επεξεργασία τους.

Οι παραδοσιακές λύσεις, όπως οι Time-Delayed Integration (TDI) κάμερες ή η χειροκίνητη εξαγωγή των χαρακτηριστικών, έχουν τα μειονεκτήματά τους, όπως υψηλό κόστος και έλλειψη ευκολίας χρήσης σε εφαρμογές που απαιτούν υψηλό ρυθμό ροής δεδομένων. Τελευταία, προηγμένα μοντέλα βαθιάς μάθησης έχουν εφαρμοστεί με επιτυχία, αλλά απαιτούν μεγάλα νευρικά δίκτυα, ισχυρούς υπολογιστικούς πόρους και μεγάλο χρόνο εκπαίδευσης.

Μια καινοτόμος προσέγγιση για την αντιμετώπιση αυτών των προβλημάτων είναι η χρήση βιολογικά εμπνευσμένων event cameras. Αυτοί οι ασύγχρονοι αισθητήρες καταγράφουν μόνο τις αλλαγές στο επίπεδο φωτεινότητας κάθε pixel και, συνεπώς, παρέχουν υψηλή χρονική ανάλυση (έως 1 μ s), χαμηλή καθυστέρηση, υψηλό δυναμικό εύρος και χαμηλή κατανάλωση ενέργειας. Τα δεδομένα καταγράφονται ως αραιά event streams, με αποτέλεσμα να είναι πολύ μικρότερος ο όγκος των πληροφοριών και ταυτόχρονα να διατηρούνται μόνο οι σχετικές χρονικές αλλαγές.

Παρότι έχουν γίνει εφαρμογές των αισθητήρων συμβάντων στο IFC, τα περισσότερα μοντέλα μηχανικής μάθησης συνεχίζουν να μετατρέπουν τα events σε εικόνες (frames), χωρίς να αξιοποιούν πλήρως τη χρονική διάσταση των δεδομένων. Ακόμη, πολλές υλοποιήσεις βασίζονται σε συμβατικά νευρωνικά δίκτυα, τα οποία χρειάζονται πολλούς υπολογιστικούς πόρους και περίπλοκη προεπεξεργασία.

Η έρευνα των Moustakas et al. (2024) αντιμετωπίζει αυτά τα προβλήματα, εξετάζοντας διάφορες ελαφριές αρχιτεκτονικές Spiking Neural Networks (SNNs) σε δεδομένα IFC που καταγράφηκαν με κάμερα συμβάντων. Η δοκιμή επικεντρώνεται στην αναγνώριση σωματιδίων PMMA (με διάμετρο 12, 16 και 20 μικρόμετρα), τα οποία κινούνται με ταχύτητα 0,7 m/s μέσα από ένα μικρορευστοποιητικό κανάλι.

Οι SNN αρχιτεκτονικές που εξετάζονται είναι:

- Spiking Multilayer Perceptron (SMLP) με νευρώνες τύπου Leaky Integrate-and-Fire (LIF), με ή χωρίς μηχανισμούς προσαρμογής,
- Spiking Long Short-Term Memory (SLSTM), δηλαδή η spiking έκδοση του LSTM,
- Spiking Gated Recurrent Unit (SGRU), αντίστοιχα η spiking έκδοση του GRU.

Τα κύρια ευρήματα της μελέτης είναι τα εξής:

- Τα cyclic δίκτυα (SLSTM και SGRU) είχαν τη μεγαλύτερη ακρίβεια, με το SLSTM να έχει την καλύτερη επίδοση, 98,43%, όταν οι γραμμές και οι στήλες ταυτόχρονα χρησιμοποιούνταν ως είσοδοι (Rows & Columns). Για τα σωματίδια 12 μm , το SLSTM έφτασε την ακρίβεια του 100%.
- Οι προσαρμοστικοί (adaptive) μηχανισμοί στα feedforward δίκτυα SMLP βελτίωσαν την ακρίβεια έως και 4,3%, με το Adaptive SMLP να έχει 96,875% ακρίβεια.
- Όσον αφορά την πολυπλοκότητα, τα SMLP έχουν σημαντικά λιγότερες παραμέτρους (π.χ. 305 για το LIF SMLP έναντι 50.556 για το SLSTM) και απαιτούν λιγότερους υπολογισμούς (60.600 πολλαπλασιασμοί έναντι 5.035.600), με μια μικρή απώλεια στην ακρίβεια.
- Λόγω του ότι τα δεδομένα από τους αισθητήρες συμβάντων είναι δυαδικά, μπορούμε να κάνουμε αθροίσματα αντί για πολλαπλασιασμούς, με αποτέλεσμα να μειώνονται ακόμα περισσότερο οι απαιτήσεις σε υλικό και ενέργεια.
- Η είσοδος των δεδομένων χρησιμοποιώντας ταυτόχρονα σειρές και στήλες έχει αποδειχθεί η πιο αποδοτική στρατηγική, καθώς εκμεταλλεύεται στο έπακρο τη χρονική πληροφορία.

Τέλος, η έρευνα των Moustakas et al. αποδεικνύει ότι τα ελαφριά SNNs είναι εξαιρετικά αποτελεσματικά στην αναγνώριση βιοϊατρικών δεδομένων σε πραγματικό χρόνο, κάτι που ανοίγει νέους ορίζοντες για την έρευνα στον τομέα της οπτικής ροής συμβάντων με τη χρήση εξειδικευμένων νευρομορφικών επεξεργαστών σε εφαρμογές όπως η ροή κυττάρων.

2.2.4 Μέθοδοι Κωδικοποίησης Πληροφορίας σε SNNs

Οι μέθοδοι κωδικοποίησης των spikes είναι βασικός παράγοντας που επηρεάζει την απόδοση των SNNs, καθώς καθορίζουν τον τρόπο με τον οποίο οι εισερχόμενες πληροφορίες μετατρέπονται σε spikes και διαδίδονται μέσα στο νευρικό δίκτυο. Η επιλογή του κατάλληλου σχήματος κωδικοποίησης έχει κρίσιμη επίδραση τόσο στην ακρίβεια της ταξινόμησης όσο και στην υπολογιστική και ενεργειακή απόδοση των SNNs.

Στη μελέτη του Rueckauer και Liu (2018) παρουσιάζονται διάφορα σχήματα χρονικής κωδικοποίησης (temporal encoding), όπως:

Κωδικοποίηση βάσει του ρυθμού εμφάνισης των spikes (Rate coding): Η πληροφορία κωδικοποιείται στη συχνότητα εμφάνισης των spikes.

- Χρονική κωδικοποίηση (Temporal coding): Η πληροφορία κωδικοποιείται με βάση το χρονικό διάστημα μέχρι την εμφάνιση του spike.
- Φασική κωδικοποίηση (Phase coding): Η χρονική στιγμή του spike συσχετίζεται με τη φάση ενός κυκλικού εξωτερικού σήματος.

Στο ίδιο άρθρο, οι συγγραφείς αξιολόγησαν τις παραπάνω τεχνικές χρονικής κωδικοποίησης χρησιμοποιώντας κλασικά datasets όπως τα MNIST και CIFAR-10 και παρατήρησαν ότι η latency coding (πρώιμη εμφάνιση των spikes) προσέφερε μεγαλύτερη ακρίβεια σε σενάρια εξοικονόμησης ενέργειας, ενώ η χρήση του rate coding παρείχε μεγαλύτερη σταθερότητα κατά τη διάρκεια της εκπαίδευσης.

Επίσης, οι Tavanaei et al., (2019) αναφέρουν ότι η προεπεξεργασία των δεδομένων και η χρήση τεχνικών data augmentation (όπως η τυχαία περιστροφή ή μετάθεση εικόνων)

μπορούν σημαντικά να βελτιώσουν την απόδοση των SNNs. Σύμφωνα με τη μελέτη τους, μοντέλα SNN που εφαρμόστηκαν στα datasets MNIST και CIFAR-10 και χρησιμοποίησαν τη μέθοδο κωδικοποίησης βάσει του ρυθμού εμφάνισαν ακρίβεια αυξημένη έως και 2-5% όταν συνδυάστηκαν με κατάλληλες τεχνικές προεπεξεργασίας .

Ταυτόχρονα, δίνεται έμφαση στις υβριδικές στρατηγικές κωδικοποίησης (hybrid encoding schemes), οι οποίες συνδυάζουν πληροφορίες χρόνου και συχνότητας και φαίνεται πως έχουν ιδιαίτερο ενδιαφέρον, καθώς προσφέρουν βελτιωμένη χωρική κάλυψη και είναι πιο προσαρμόσιμες σε διάφορους τύπους αισθητήρη, όπως οι event based cameras.

Συνοψίζοντας, η επιλογή του κατάλληλου σχήματος κωδικοποίησης δεν είναι απλά μία τεχνική επιλογή, αλλά καθορίζει τον τρόπο με τον οποίο τα SNNs λαμβάνουν υπόψη τους τις χρονικές πληροφορίες. Η επιστημονική κοινότητα αναζητά ολοένα και περισσότερο ευέλικτες μεθόδους κωδικοποίησης, οι οποίες να προσαρμόζονται σε κάθε dataset και χρήση, ενώ παράλληλα, να εξασφαλίζουν τη μέγιστη δυνατή απόδοση των νευρομορφικών συστημάτων.

2.2.5 SNNs σε Υπολογιστικά Περιορισμένα Περιβάλλοντα: Εφαρμογές σε Edge Συσκευές και Νευρομορφικό Υλικό

Η ενεργειακή απόδοση και η φυσική αλληλεπίδραση των Spiking Neural Networks (SNNs – Δίκτυα νευρώνων με αιχμές) τα κάνουν ιδανικά για χρήση σε συστήματα με περιορισμένους υπολογιστικούς πόρους, όπως τα ενσωματωμένα (embedded) συστήματα και το edge computing.

Ο όρος edge computing αναφέρεται στην επεξεργασία των δεδομένων σε κοντινή απόσταση από την πηγή τους, όπως για παράδειγμα σε έναν αισθητήρα ή μια φορητή συσκευή, αντί να στέλνονται σε ένα απομακρυσμένο κέντρο δεδομένων (cloud). Έτσι

μειώνονται οι απαιτήσεις για εύρος ζώνης και η καθυστέρηση, ενώ επιτρέπονται γρηγορότερες και πιο αποδοτικές αποφάσεις σε πραγματικό χρόνο.

Αντίθετα με τα συμβατικά τεχνητά νευρωνικά δίκτυα (ANNs), τα SNNs ενεργοποιούνται μόνο όταν λάβουν σημαντικά spikes και όχι σε κάθε χρονικό βήμα. Επομένως, επιτυγχάνεται σημαντική εξοικονόμηση ενέργειας και μειώνονται οι απαιτήσεις για την επεξεργασία και τη μεταφορά δεδομένων – κάτι ιδιαίτερα σημαντικό για μικρές συσκευές με περιορισμένη μπαταρία.

Νευρομορφικά Υπολογιστικά Συστήματα

Ο ευρύς χώρος εφαρμογών των SNNs οφείλεται κυρίως στην ανάπτυξη νευρομορφικού υλικού (neuromorphic hardware) – δηλαδή, εξειδικευμένων μικροεπεξεργαστών που έχουν σχεδιαστεί με βάση τον ανθρώπινο εγκέφαλο. Τα συστήματα αυτά επιτρέπουν την ασύγχρονη επεξεργασία και χρησιμοποιούν τους spikes σαν μονάδες πληροφορίας.

Παραδείγματα τέτοιων συστημάτων περιλαμβάνουν τα εξής:

- Intel Loihi: Ένα νευρομορφικό chip που υποστηρίζει την online μάθηση και την αξιοποίηση των spikes. Σύμφωνα με την Intel, εξοικονομεί έως και 100 φορές περισσότερη ενέργεια σε συγκεκριμένες εφαρμογές (Davies et al., 2021).
- SpiNNaker: Ένα project του Πανεπιστημίου του Μάντσεστερ το οποίο αποσκοπεί στην προσομοίωση μεγάλης κλίμακας νευρικών δικτύων, με δυνατότητα ταυτόχρονης επεξεργασίας εκατομμυρίων spikes ανά δευτερόλεπτο (Furber κ.ά., 2014).

- BrainScaleS: Μία νευρομορφική πλατφόρμα του Πανεπιστημίου της Χαϊδελβέργης, η οποία συνδυάζει αναλογικά και ψηφιακά κυκλώματα για την προσομοίωση δυναμικών SNNs σε πραγματικό χρόνο (real-time).

Εφαρμογές σε Edge Περιβάλλοντα

Τα SNNs έχουν εφαρμοστεί με επιτυχία σε πολλές πραγματικές καταστάσεις, όπως για παράδειγμα:

- Αναγνώριση κινήσεων σε ρομποτικά συστήματα, με χρήση event cameras,
- Ανίχνευση συμβάντων σε wearable συσκευές, όπως πτώσεις ή αρρυθμίες της καρδιάς,
- Αυτόματη κίνηση, με αισθητήρες που μετρούν τις δυναμικές αλλαγές στο περιβάλλον (π.χ. φως) ως spikes.

Υπό την έννοια αυτή, ο όρος "on-device intelligence" (νοημοσύνη στη συσκευή) χαρακτηρίζει τη δυνατότητα ενός συστήματος να επεξεργάζεται τα δεδομένα και να λαμβάνει αποφάσεις τοπικά, χωρίς να χρειάζεται υποδομές cloud ή εξωτερικό data center. Τα SNNs είναι ιδιαίτερα κατάλληλα για τέτοιου είδους εφαρμογές, λόγω της χαμηλής κατανάλωσης ενέργειας και της ικανότητάς τους να επεξεργάζονται γεγονότα με αραιή και εκρηκτική ενεργοποίηση (sparse activation).

Ακόμη και σε περιβάλλοντα όπως το Lava framework της Intel (2022), οι SNNs είναι εξαιρετικά ευέλικτοι και μπορούν να υποστηρίξουν πολλά παραδείγματα μάθησης, όπως η μάθηση με surrogate gradients ή οι κανόνες STDP.

Συμπερασματικά η χρήση SNNs σε υπολογιστικά περιορισμένα περιβάλλοντα βοηθάει στην προσπάθεια να δημιουργήσουμε συστήματα που να μοιάζουν όσο το δυνατόν περισσότερο με τον ανθρώπινο εγκέφαλο, χωρίς να αγνοείται η πρακτική προσέγγιση της μηχανικής μάθησης. Το γεγονός ότι χρειάζονται λίγη ενέργεια, μπορούν να λειτουργούν ασύγχρονα και να χρησιμοποιούν εξειδικευμένο υλικό ανοίγει νέους ορίζοντες σε εφαρμογές όπως η έξυπνη κίνηση, τα φορετά ιατρικά συστήματα, οι αισθητήρες περιβάλλοντος και το διαδίκτυο των Πραγμάτων (IoT).

ΜΕΡΟΣ ΔΕΥΤΕΡΟ

ΠΕΙΡΑΜΑΤΙΚΟ ΠΛΑΙΣΙΟ

ΚΕΦΑΛΑΙΟ 3^ο

Μεθοδολογία

3.1 Περιγραφή Πειραμάτων

Η παρούσα εργασία εστιάζεται στη μελέτη της απόδοσης των Νευρωνικών Δικτύων Όσεων (SNNs) σε προβλήματα ταξινόμησης εικόνων, χρησιμοποιώντας το σετ δεδομένων N-MNIST. Πραγματοποιήθηκαν συνολικά 30 πειράματα, στα οποία εξετάστηκαν εκτενώς διάφορες συναρτήσεις προσεγγιστικής παραγώγου (surrogate gradient functions), παράμετροι ρύθμισης και συναρτήσεις απώλειας.

Τα βασικά στοιχεία που άλλαξαν στα πειράματα καταγράφονται παρακάτω:

1) Προσεγγιστικές Παράγωγοι (Surrogate Gradients): Δοκιμάστηκαν τέσσερις διαφορετικές προσεγγίσεις:

- i. Fast Sigmoid
- ii. ATan (ArcTan)
- iii. Sigmoid
- iv. SpikeRateEscape

Η επιλογή αυτών των συναρτήσεων βασίστηκε στη θεωρητική τους και πρακτική τους σημασία, όπως περιγράφεται στο Κεφάλαιο 2. Οι παράμετροι slope/alpha/beta επιλέχθηκαν δοκιμαστικά έτσι, ώστε να εξασφαλίζεται αντιπροσωπευτική συμπεριφορά σε κάθε περίπτωση.

2) Συναρτήσεις Απώλειας (Loss Functions): Για την εκπαίδευση των μοντέλων χρησιμοποιήθηκαν τρεις διαφορετικές συναρτήσεις κόστους:

i. Συνάρτηση Cross Entropy με spike counts (SF.ce_count_loss)

Η συνάρτηση Cross Entropy με spike counts προσαρμογές είναι του κλασικού τύπου της για SNNs (Spiking Neural Networks), στα οποία η έξοδος κάθε νευρώνα αντιστοιχεί στον αριθμό των ακίδων(spikes) που παράγονται σε μια συγκεκριμένη χρονική περίοδο. Αυτή η συνάρτηση χρησιμοποιείται για προβλήματα ταξινόμησης και βασίζεται στο άθροισμα των ακίδων ως μέτρο ενεργοποίησης των νευρώνων εξόδου.

Εστω:

- C : το πλήθος κατηγοριών (νευρώνες εξόδου)
- $s_{j(t)}$: spike του εξόδου νευρώνα j στη χρονική στιγμή t
- c_j : συνολικά spikes του νευρώνα j
- y_j : η σωστή κατηγορία (one-hot: δυαδικό διάνυσμα (vector) μήκους ίσου με τον αριθμό των κατηγοριών, όπου μόνο μία θέση είναι 1 και οι υπόλοιπες είναι 0)

1. Υπολογισμός των spike counts :

$$c_j = \sum_{t=1}^T s_j(t)$$

2. Εφαρμογή Softmax:

$$p_j = \frac{\exp(c_j)}{\sum_{k=1}^C \exp(c_k)}$$

3. Cross Entropy Loss:

$$L = - \sum_{j=1}^C y_j \log(p_j)$$

ή, για κατηγορική μεταβλητή(class index) y :

$$L = \log \left(\sum_{k=1}^C \exp(c_k) \right) - c_y$$

Η συγκεκριμένη συνάρτηση τιμωρεί τα λάθη κατά τις περιπτώσεις όπου η σωστή κατηγορία δεν έχει τα περισσότερα spikes. Η υπολογίζεται μόνο μία φορά στο τέλος της προσομοίωσης και γι' αυτό είναι πολύ γρήγορη από άποψη υπολογιστικής επίδοσης. Επιπλέον, την υποστηρίζουν σχεδόν όλες οι βασικές βιβλιοθήκες για SNNs, όπως η `spnnTorch`.

ii. Τετραγωνικό Μέσο Σφάλμα με spike counts (SF.mse_count_loss)

Η συνάρτηση `SF.mse_count_loss` του Τετραγωνικού Μέσου Σφάλματος με spike counts είναι μία παραλλαγή της κλασικής Mean Squared Error (MSE), που έχει προσαρμοστεί για να χρησιμοποιείται σε Νευρωνικά Δίκτυα Όσεων (Spiking Neural Networks – SNNs). Αντί να χρησιμοποιείται σε συνεχείς εξόδους όπως σε ένα κλασικό νευρωνικό δίκτυο, εφαρμόζεται στον αριθμό των spikes (πυροδοτήσεων) που παράγει ο κάθε νευρώνας εξόδου κατά τη διάρκεια της προσομοίωσης.

Αυτή η συνάρτηση χρησιμοποιείται τόσο σε προβλήματα παλινδρόμησης (regression) όσο και σε προβλήματα ταξινόμησης (classification), όταν οι στόχοι δίνονται σε μορφή πλήθους επιθυμητών spikes.

Εστω:

- C : το πλήθος κατηγοριών (νευρώνες εξόδου)
- c_j : συνολικά spikes (πλήθος) του νευρώνα j
- t_j : ο επιθυμητός αριθμός spikes για την κατηγορία j (δηλαδή ο στόχος – target spike count).

Η συνάρτηση υπολογίζεται ως εξής:

$$\text{MSE}_{\text{count}_{\text{loss}}} = \frac{1}{C} \sum_{j=1}^C (c_j - t_j)^2$$

1. Υπολογισμός των spike counts :

$$c_j = \sum_{t=1}^T s_j(t)$$

2. Υπολογισμός τετραγωνικής διαφοράς:

$$(c_j - t_j)^2$$

3. Μέσος Όρος στο σύνολο των κατηγοριών:

$$\text{MSE}_{\text{count}_{\text{loss}}} = \frac{1}{C} \sum_{j=1}^C (c_j - t_j)^2$$

Στο παρακάτω παράδειγμα, το δίκτυο εξόδου επέστρεψε spike counts [2, 8, 1] για τρεις κατηγορίες, ενώ ο στόχος (one-hot target) ήταν [0, 10, 0], δηλαδή η σωστή κατηγορία είναι η δεύτερη με επιθυμητό πλήθος spikes ίσο με 10. Η συνάρτηση απώλειας MSE υπολογίζεται ως:

$$\text{MSE}_{\text{count}_{\text{loss}}} = \frac{1}{3} [(2 - 0)^2 + (8 - 10)^2 + (1 - 0)^2] = \frac{1}{3} (4 + 4 + 1) = 3$$

Η τιμή που προκύπτει (3) δεν αποτελεί προβλεπόμενη κατηγορία, αλλά αντιπροσωπεύει το μέσο τετραγωνικό σφάλμα της πρόβλεψης σε σχέση με τον επιθυμητό στόχο. Όσο μικρότερη είναι αυτή η τιμή, τόσο πιο κοντά βρίσκεται το δίκτυο στο να επιτύχει την επιθυμητή έξοδο.

Η συνάρτηση τιμωρεί τις περιπτώσεις όπου το πλήθος των spikes αποκλίνει σημαντικά από την τιμή που είναι επιθυμητό. Αυτό εφαρμόζεται ιδιαίτερα σε περιβάλλοντα ταξινόμησης, όπου ο "σωστός" νευρώνας πρέπει να έχει αισθητά περισσότερα spikes από τους υπόλοιπους. Με αυτόν τον τρόπο, γίνεται εκπαίδευση χρησιμοποιώντας surrogate παραγώγους και το `nnTorch` έχει όλη την απαραίτητη υλοποίηση για αυτό.

iii. Τετραγωνικό Μέσο Σφάλμα μεμβράνης (SF.mse_membrane_loss)

Η συνάρτηση Mean Squared Error Membrane Loss (SF.mse_membrane_loss) αποτελεί μία παραλλαγή της κλασικής συνάρτησης Τετραγωνικού Μέσου Σφάλματος (Mean Squared Error - MSE), που έχει σχεδιαστεί ειδικά για να χρησιμοποιείται σε Νευρωνικά Δίκτυα Ώσεων (Spiking Neural Networks – SNNs). Αντί να κάνει την εκτίμησή της βασιζόμενη στον αριθμό των spikes, η συνάρτηση αυτή συγκρίνει τη δυναμική της μεμβράνης (membrane potential) ανάμεσα στην έξοδο του SNN και την επιθυμητή τιμή-στόχο σε κάθε χρονικό βήμα.

Επεξήγηση Όρων και μαθηματική περιγραφή

- Μεμβρανική Τάση (Membrane Potential, $V_j(t)$) : Η ηλεκτρική δυναμική του νευρώνα εξόδου j σε χρονικό βήμα t , που επηρεάζει το αν ο νευρώνας θα εκπέμψει spike.
- Στόχος Μεμβράνης (Target Membrane Value, $T_j(t)$): Η επιθυμητή τιμή της μεμβρανικής τάσης για τον νευρώνα j στο χρονικό βήμα t .
- Χρονικά βήματα (T): Ο αριθμός χρονικών βημάτων μιας προσομοίωσης.

- Κατηγορίες/Εξοδοι (C): Ο αριθμός των εξόδων του δικτύου.

Εστω όλα τα παραπάνω η συνάρτηση υπολογίζεται ως εξής:

$$\text{MSE}_{\text{membrane}} = \frac{1}{C \cdot T} \sum_{j=1}^C \sum_{t=1}^T \left(V_j(t) - T_j(t) \right)^2$$

Ωπου :

- $V_j(t)$: η τιμή της μεμβράνης (δυναμικό)
- $T_j(t)$: τιμή στόχου (σταθερού ή χρονικά μεταβαλλόμενου)
- $\left(V_j(t) - T_j(t) \right)^2$: η τετραγωνική διαφορά (υπολογισμός σφάλματος)
- $\frac{1}{C \cdot T} \sum_{j=1}^C \sum_{t=1}^T$: Μέσος όρος σε όλες τις εξόδους και timesteps

Η συνάρτηση αυτή καταγράφει το σφάλμα σε κάθε χρονική στιγμή και για κάθε έξοδο, επιτρέποντας έτσι στον αλγόριθμο εκπαίδευσης να ρυθμίζει τη συμπεριφορά των νευρώνων όχι μόνο με βάση το αν έχουν εκπέμψει σήμα, αλλά και με βάση τη δυναμική της μεμβράνης.

Είναι ιδιαίτερα χρήσιμη σε περιβάλλοντα:

- Παλινδρόμησης (regression)
- Latency coding (όπου η πληροφορία κωδικοποιείται μέσω του χρόνου ενεργοποίησης του spike)
- Εκπαίδευση SNNs με προσεγγιστικές παραγώγους (surrogate gradients)

iv. Συνάρτηση Απώλειας: Cross Entropy βάσει Ρυθμού Πυροδότησης (SF.ce_rate_loss)

Η συνάρτηση SF.ce_rate_loss αποτελεί μια παραλλαγή της συνήθους συνάρτησης cross-entropy, σχεδιασμένη ειδικά για την εκπαίδευση Spiking Neural Networks (SNNs) με χρονική δυναμική. Αντί να εφαρμόζει απλά το σφάλμα cross-entropy στο συνολικό spike count, η συγκεκριμένη συνάρτηση υπολογίζει το cross-entropy ξεχωριστά για κάθε χρονικό βήμα

(timestep), λαμβάνοντας υπόψη τα spikes που εκπέμπονται από τους νευρώνες εξόδου σε κάθε χρονική στιγμή.

Αναλυτικά, για κάθε χρονικό βήμα t , υπολογίζεται η softmax πιθανότητα για τα spikes και έπειτα υπολογίζεται το cross-entropy loss για την πραγματική κλάση εξόδου. Η τελική απώλεια υπολογίζεται ως άθροισμα ή ως μέσος όρος των απωλειών των διαφόρων χρονικών βημάτων, επιτρέποντας την ανάλυση της ακρίβειας ταξινόμησης με χρονική διάσταση.

Έστω :

- $S_j(t)$: το spike του εξόδου νευρώνα j στο χρονικό βήμα t (τιμή 0 ή 1).
- C : αριθμός κατηγοριών (νευρώνες εξόδου).
- T : αριθμός χρονικών βημάτων (timesteps).
- y : στόχος (target class) είτε σε one-hot μορφή είτε ως δείκτης κατηγορίας.

1. Υπολογισμός πιθανότητας (Softmax) για κάθε t :

$$p_j(t) = \frac{\exp(s_j(t))}{\sum_{k=1}^C \exp(s_k(t))}$$

2. Cross Entropy ανά χρονικό βήμα (one hot target):

$$L_t = - \sum_{j=1}^C y_j \log(p_j(t))$$

3. Cross-Entropy loss ανά timestep (Κατηγορική ετικέτα)

$$L_t = - \log(p_y(t))$$

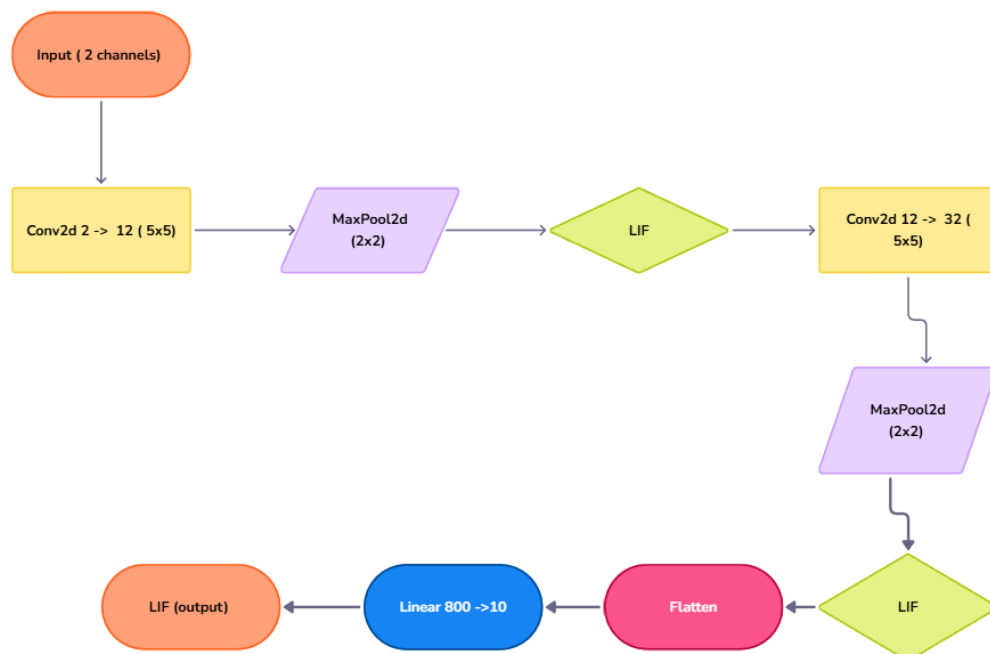
4. Συνολική απώλεια με μέσο όρο ή άθροισμα

$$L = \frac{1}{T} \sum_{t=1}^T L_t \text{ ή } L = \sum_{t=1}^T L_t$$

Αυτή η συνάρτηση ενισχύει το σήμα εξόδου του σωστού νευρώνα σε κάθε χρονικό βήμα, αντί να βασίζεται μόνο στο συνολικό πλήθος spikes. Έτσι, εκμεταλλεύεται πλήρως τη χρονική πληροφορία της νευρικής δραστηριότητας, κάτι που είναι ιδιαίτερα χρήσιμο για για rate-coded SNNs. Έχει παρατηρηθεί ότι αυτή η μέθοδος μπορεί να οδηγήσει σε πιο σταθερή εκπαίδευση και να επιτρέπει την καλύτερη ανάλυση της χρονικής δομής των δεδομένων, ιδίως σε προβλήματα ταξινόμησης που εξελίσσονται με τον χρόνο.

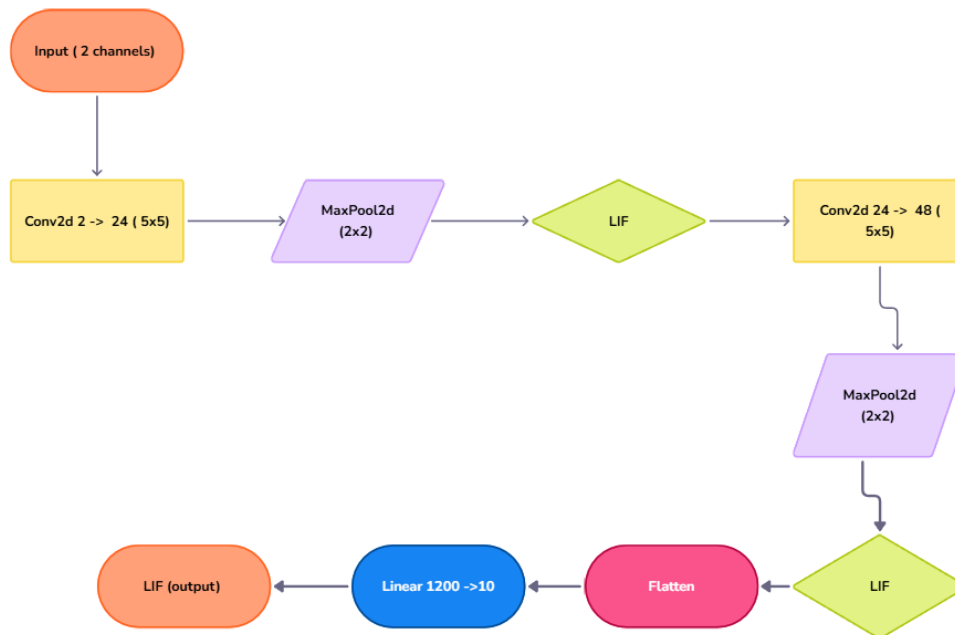
3) Σχεδίαση Δικτύου: Δοκιμάστηκαν δύο διαφορετικές εκδοχές των συνελκτικών δικτύων (Conv A και Conv B), με τις εξής δομές:

- i. Conv(2→12→32): Αποτελείται από δύο στρώματα συνέλιξης, (Conv2d με 12 και 32 φίλτρα αντίστοιχα), τα οποία ακολουθούνται από MaxPooling και ένα επίπεδο Linear(800→10).



Εικόνα 3: Η αρχιτεκτονική Conv A περιλαμβάνει δύο συνελκτικά επίπεδα με 12 και 32 φίλτρα αντίστοιχα, καθένα εκ των οποίων ακολουθείται από max-pooling και LIF (Leaky Integrate-and-Fire) νευρώνα. Τα δεδομένα εν συνεχεία ισοπεδώνονται (flatten) και διέρχονται από ένα πλήρως συνδεδεμένο επίπεδο (Linear) με 800 εισόδους και 10 εξόδους. Οι νευρώνες εξόδου είναι επίσης τύπου LIF.

- ii. Conv(2→24→48): Έχει παρόμοια δομή, αλλά μεγαλύτερο αριθμό φίλτρων (24 και 48 αντίστοιχα) και fully connected επίπεδο Linear(1200→10).



Εικόνα 4: Η αρχιτεκτονική Conv B είναι παρόμοια, αλλά με μεγαλύτερη χωρητικότητα: χρησιμοποιούνται 24 και 48 φίλτρα στα συνεκτικά επίπεδα και το fully connected επίπεδο περιλαμβάνει 1200 εισόδους. Αυτή η διαμόρφωση έχει ως στόχο να αυξήσει την εκφραστική ικανότητα του δικτύου, με απώτερο σκοπό την βελτίωση της ακρίβειας ταξινόμησης.

Οι νευρώνες σε όλα τα στρώματα είναι τύπου *Leaky Integrate-and-Fire* (LIF) με δυνατότητα ρύθμισης της παραμέτρου διαρροής β.

4) Μέγεθος Παρτίδας (Batch Size):

Το μέγεθος παρτίδας (Batch Size) αναφέρεται στον αριθμό των δειγμάτων που χρησιμοποιεί το δίκτυο σε κάθε ενημέρωση βαρών κατά τη διάρκεια της εκπαίδευσης. Είναι

μία σημαντική υπερπαράμετρος, μιας και επηρεάζει τόσο τη σταθερότητα της διαδικασίας εκμάθησης όσο και την απόδοση των υπολογιστικών πόρων

Στην παρούσα εργασία επιλέχθηκαν οι τιμές 64 και 128 ανάλογα με το συγκεκριμένο πείραμα, έτσι ώστε να εξασφαλιστεί η βέλτιστη σταθερότητα και αποδοτικότητα της εκπαίδευσης. Τα μικρότερα μεγέθη του batch size κάνουν πιο συχνές τις ενημερώσεις των βαρών και προκαλούν αυξημένη στοχαστικότητα, ενώ τα μεγαλύτερα παρέχουν πιο σταθερές εκτιμήσεις του σφάλματος, με χαμηλότερο υπολογιστικό κόστος ανά δείγμα.

Για κάθε πείραμα χρησιμοποιούνταν διαφορετικές τιμές των παραπάνω χαρακτηριστικών, με στόχο να εξεταστεί η απόδοση (ακρίβεια ταξινόμησης, σύγκλιση, διαφορετικά επίπεδα spikes) σε διάφορα σενάρια. Η υλοποίηση έγινε με τη βοήθεια του περιβάλλοντος Google Colab και των βιβλιοθηκών snntorch, tonic και PyTorch. Τα αποτελέσματα καταγράφηκαν σε πίνακα για ευκολότερη ανάλυση.

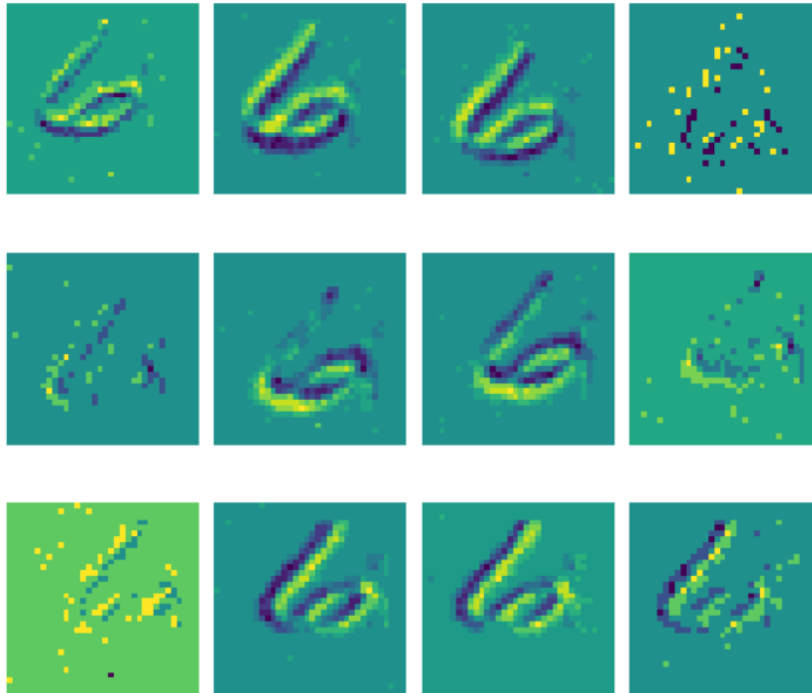
3.2 Περιγραφή Συνόλου Δεδομένων και Προεπεξεργασία

Το N-MNIST (Neuromorphic MNIST) είναι ένα neuromorphic dataset που αποτελεί επέκταση του κλασικού MNIST και περιλαμβάνει εικόνες χειρόγραφων αριθμών σε μορφή 28x28 pixel. Σκοπός του N-MNIST είναι να παρέχει σύνολα δεδομένων για την εκπαίδευση και αξιολόγηση Spiking Neural Networks (SNNs), παρέχοντας χρονική πληροφορία σε μορφή γεγονότων (spikes) αντί για στατικές εικόνες (Orchard et al., 2015).



Εικόνα 5: Η εικόνα αυτή περιέχει πολλαπλά δείγματα από το κλασικό MNIST dataset. Κάθε εικόνα δείχνει έναν χειρόγραφο αριθμό ως μία στατική εικόνα 28x28 pixels. Δίπλα από κάθε ψηφίδα υπάρχει η ετικέτα (label) της, από 0 έως 9. Είναι στατικά δεδομένα χωρίς καμία χρονική πληροφορία, κατάλληλα για την εκπαίδευση Κλασικών Τεχνητών Νευρωνικών Δικτύων (ANNs, CNNs) (Πηγή : m-alcu.github.io/).

Διαφορετικά από το MNIST, όπου κάθε δείγμα (sample) αντιστοιχεί σε μια στατική εικόνα, το N-MNIST καταγράφει μία ακολουθία γεγονότων (spike events) που παράγονται από έναν neuromorphic αισθητήρα τύπου DVS (Dynamic Vision Sensor) καθώς κινείται μπροστά από την εικόνα MNIST σε ένα monitor. Με άλλα λόγια, κάθε δείγμα του N-MNIST αναπαριστά μια χρονική ακολουθία κωδικοποιημένων spikes που δείχνει το πώς το αισθητήριο αντιλαμβάνεται τον αριθμό μέσα στο χρόνο.



Εικόνα 6: Η παρούσα εικόνα παρουσιάζει πολλές διαδοχικές χρονικές στιγμές του ίδιου ψηφίου από το νευρομορφικό dataset N-MNIST. Πρόκειται για frame-based αναπαράσταση (π.χ. 12 στιγμιότυπα) της ίδιας χρονικής ροής πυροδότησης, όπως αυτή παρήχθη από τον αισθητήρα DVS κατά την προβολή του ψηφίου. Κάθε υπό-εικόνα δείχνει τη φωτεινότητα/ενεργοποίηση τη συγκεκριμένη στιγμή, όπου η έντασή της υποδηλώνει το πόσο έντονα έγινε η αλλαγή (Πηγή: github.com/halykoss/SNN-NMNIST-Object-Recognition).

Η ιδέα για το N-MNIST προήλθε από τη χρήση ενός αισθητήρα τύπου ATIS (Asynchronous Time-based Image Sensor – Ασύγχρονος Χρονικά Κωδικοποιημένος Αισθητήρας Εικόνας), που πραγματοποιεί ελεγχόμενες κινήσεις μικρής διάρκειας (γνωστές και ως saccades) μπροστά από οθόνη πάνω στην οποία προβάλλονται οι εικόνες του MNIST. Αυτές οι κινήσεις προκαλούν αλλαγές στη φωτεινότητα των pixels, οι οποίες ο αισθητήρας καταγράφει ως spikes. Το αποτέλεσμα είναι ένα σύνολο από γεγονότα που περιέχουν χωρικές και χρονικές πληροφορίες, γεγονός που αναπαριστά το πώς ένα ψηφίο αντιλαμβάνεται η οπτική είσοδος κατά τη διάρκεια του χρόνου.

Η δομή του συνόλου δεδομένων για το N-MNIST περιλαμβάνει:

- I. **Αριθμός δειγμάτων:** 60.000 για εκπαίδευση, 10.000 για δοκιμή.

II. **Διαστάσεις:** 28x28 pixels (χωρική ανάλυση).

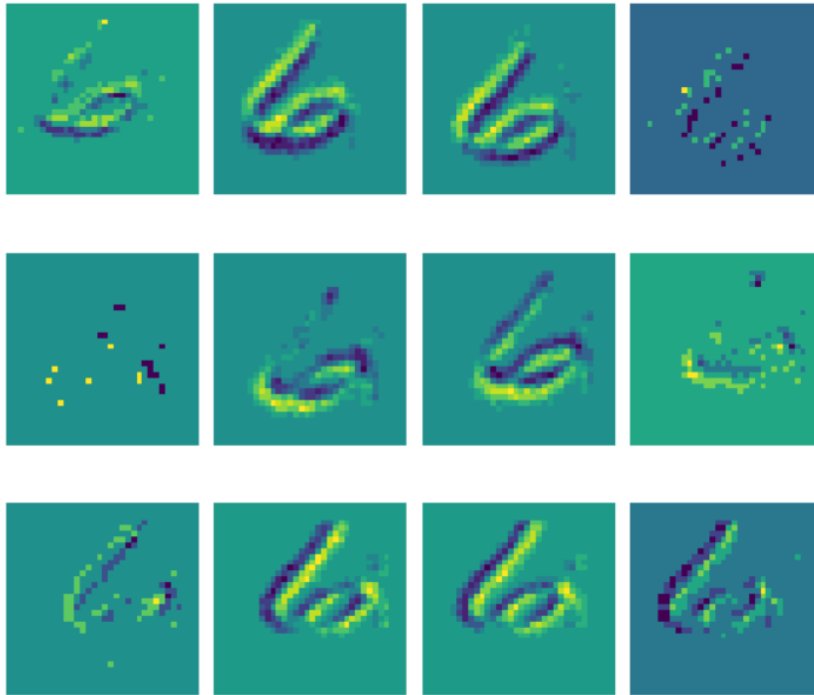
III. **Δεδομένα:** Για κάθε δείγμα, υπάρχει μια λίστα από γεγονότα (events), το καθένα με μορφή (x, y, t, p) , όπου x, y είναι οι συντεταγμένες, t ο χρόνος και p η πολικότητα (on/off).

IV. **Χρονικός χαρακτήρας:** Η πληροφορία περιλαμβάνει τόσο χωρική όσο και χρονική διάσταση, προσομοιώνοντας τον τρόπο που ένα βιολογικό μάτι επεξεργάζεται την εικόνα μέσω μικροκινήσεων.

Κάθε δείγμα αναπαρίσταται ως μια (x, y, t, p) ακολουθία, όπου το t είναι ο χρόνος σε microseconds. Τα spikes είναι κατανεμημένα στον χρόνο και αυτό επιτρέπει στα SNNs να μαθαίνουν χρονικά πρότυπα, που είναι ένα από τα βασικά τους πλεονεκτήματα σε σχέση με τα κλασικά ANNs. Επιπλέον, οι πληροφορίες για τον φωτισμό καταγράφονται μόνο όταν υπάρχουν αλλαγές, γεγονός που σημαίνει ότι ο όγκος των δεδομένων που παράγονται είναι πολύ μικρότερος σε σχέση με τις στατικές εικόνες.

Η προεπεξεργασία των δεδομένων του N-MNIST απαιτεί τα εξής στάδια, ώστε να είναι κατάλληλα για είσοδο σε ένα Spiking Neural Network:

Αρχικά, πρέπει να γίνει η διαδικασία του καθαρισμού του θορύβου με τη χρήση φίλτρων, όπως το Denoise της βιβλιοθήκης tonic, έτσι ώστε να αφαιρεθούν τα spikes που προκαλούνται από αδρανείς ή τυχαίες μεταβολές. Στη συνέχεια, η χρονική ροή των δεδομένων πρέπει να μετατραπεί σε μια ακολουθία καρέ (frames) διάρκειας 1ms, με τη βοήθεια της τεχνικής Frame Encoding, ώστε να μπορούν να χρησιμοποιηθούν και τα frame-based SNNs. Τέλος, τα spikes χωρίζονται σε δύο διαφορετικά κανάλια, ένα για τα θετικά (on) και ένα για τα αρνητικά (off) spikes, με αποτέλεσμα να προκύπτει ένα σήμα εισόδου 2 καναλιών, ισοδύναμο με τα χρωματικά κανάλια RGB.



Εικόνα 7: Η εικόνα παρουσιάζει ένα ψηφιακό σήμα που έχει κωδικοποιηθεί την ίδια χρονική στιγμή με την προηγούμενη φωτογραφία (εικόνα 6) , μετά από την εφαρμογή τεχνικών αποθρομβοποίησης. Είναι προφανές ότι τα *strikes* είναι πιο "καθαρά", χωρίς πολλά σποραδικά σημεία, κάτι που βελτιώνει την ακρίβεια και την σταθερότητα των *SNNs*. Ο θόρυβος που προέρχεται από τον αισθητήρα ή από εξωτερικές παρεμβολές έχει εξαλειφθεί (Πηγή : github.com/halykoss/SNN-NMNIST-Object-Recognition).

Για να διασφαλιστεί η συνοχή των δεδομένων, γίνεται επίσης η zero padding (μηδενική συμπλήρωση) έτσι ώστε όλα τα δείγματα σε κάθε παρτίδα (batch) να έχουν το ίδιο χρονικό μήκος. Οι ετικέτες (labels) των ψηφίων αντιστοιχούν σε one-hot διανύσματα (π.χ. το ψηφίο 2 αντιστοιχεί στο διάνυσμα $[0, 0, 1, 0, \dots, 0]$), ώστε να είναι δυνατή η χρήση των κατάλληλων συναρτήσεων κόστους κατά την εκπαίδευση του μοντέλου.

Το τελικό input στο μοντέλο αποτελείται από έναν τανυστή (πολυδιάστατο πίνακα) που έχει πέντε διαστάσεις και είναι της μορφής:

$$[T, B, C, H, W] = [\text{Χρονικά βήματα}, \text{Μέγεθος Παρτίδας (Batch Size)}, 2, 34, 34]$$

Όπου

- T : ο αριθμός χρονικών βημάτων (πχ 300 frames)
- B : μέγεθος παρτίδας (batch size)
- $C = 2$: αριθμός καναλιών (on/off spikes)
- H, W : ύψος και πλάτος της εικόνας (με padding 34x34)

Σύγκριση MNIST και N-MNIST

Χαρακτηριστικό	MNIST (Στατικό)	N-MNIST (Νευρομορφικό)
Τύπος Δεδομένων	Εικόνες (28x28, 8-bit)	Ακολουθίες γεγονότων (spikes)
Πληροφορία	Χωρική	Χωρική και Χρονική
Εφαρμογή	CNNs και ANNs	SNNs και νευρομορφικό υλικό

3.3 Αρχιτεκτονική Δικτύου και Ρυθμίσεις Εκπαίδευσης

Τα SNN (Spiking Neural Networks) ονομάζονται έτσι επειδή βασίζονται στη βιολογική αρχή των ακίδων (spikes) για τη μεταφορά πληροφορίας. Το δίκτυο που χρησιμοποιήθηκε αποτελείται από συνελκτικά (convolutional) και πλήρως διασυνδεδεμένα (fully connected) επίπεδα. Εδώ παρουσιάζεται το πιο πολύπλοκο σχήμα που χρησιμοποιήθηκε στις δοκιμές, με τα συνελκτικά επίπεδα $\text{Conv}(2 \rightarrow 24 \rightarrow 48)$ να διαδέχονται το ένα το άλλο και τέλος να έχουμε δέκα νευρώνες στην έξοδο (1 ανά κλάση του dataset):

```

spike_grad = surrogate.fast_sigmoid(slope=25)

beta = 0.7

net = nn.Sequential(
    nn.Conv2d(2, 24, 5),
    nn.MaxPool2d(2),
    snn.Leaky(beta=beta, spike_grad=spike_grad, init_hidden=True),
    nn.Conv2d(24, 48, 5),
    nn.MaxPool2d(2),
    snn.Leaky(beta=beta, spike_grad=spike_grad, init_hidden=True),
    nn.Flatten(),
    nn.Linear(48*5*5, 10),
    snn.Leaky(beta=beta, spike_grad=spike_grad, init_hidden=True, output=True)
)

```

Οι δοκιμές πραγματοποιήθηκαν με χρήση των νευρώνων `snn.Leaky` που υλοποιούν το μοντέλο Leaky Integrate-and-Fire (LIF). Πρόκειται για ένα από τα πιο διαδεδομένα μοντέλα spiking νευρώνων που προσομοιώνουν τη συμπεριφορά ενός βιολογικού νευρώνα βασιζόμενα στις εξής αρχές:

- i. Ο νευρώνας έχει δυναμικό μεμβράνης (το συσσωρεύει) που υπολογίζεται από τα εισερχόμενα σήματα (ενσωμάτωση).
- ii. Υπάρχει εκθετική διαρροή του δυναμικού με την πάροδο του χρόνου.
- iii. Όταν το δυναμικό ξεπεράσει ένα ορισμένο όριο ή κατώφλι (threshold), ο νευρώνας εκπέμπει μία ακίδα (spike) και το δυναμικό επιστρέφει στην αρχική του τιμή.

Η λειτουργία αυτή περιγράφεται από την ακόλουθη διαφορική εξίσωση:

$$\tau_m \frac{dV(t)}{dt} = -[V(t) - V_{\text{rest}}] + RI(t)$$

Όπου :

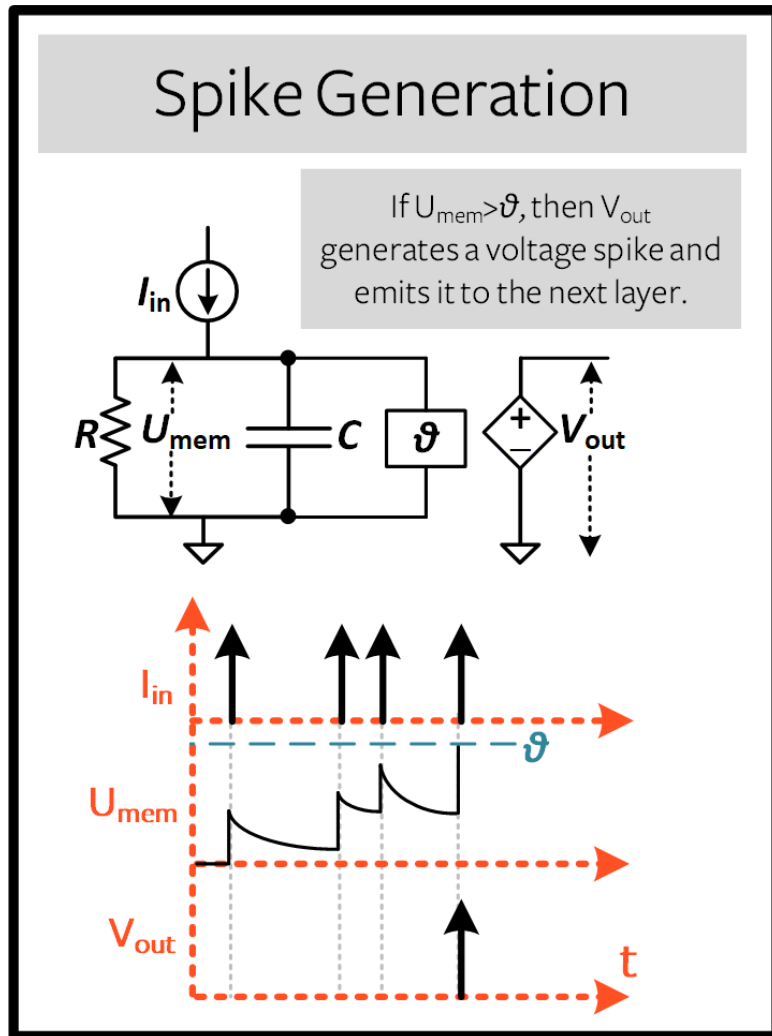
- $V(t)$: Δυναμικό μεμβράνης τη χρονική στιγμή t
- V_{rest} : Δυναμικό ηρεμίας
- R : Η Αντίσταση της μεμβράνης
- $I(t)$: Το εισερχόμενο Ρεύμα
- $\tau_m = RC$: Η χρονική σταθερά της μεμβράνης

Σε αριθμητική (διακριτή) μορφή για προσομοίωση σε υπολογιστή, η εξίσωση γράφεται ως:

$$V[t + 1] = \beta V[t] + (1 - \beta)(RI[t + 1] + V_{\text{rest}})$$

Όπου:

$\beta = e^{-\Delta t / \tau_m}$ με Δt το χρονικό βήμα.



Εικόνα 8: Αναπαράσταση ενός LIF νευρώνα. **Πάνω:** Το ηλεκτρικό ισοδύναμο κύκλωμα αποτελείται από αντίσταση R και πυκνωτή C , τα οποία προσομοιώνουν τη διαρροή και την αποθήκευση του δυναμικού μεμβράνης U_{mem} . Όταν το δυναμικό υπερβεί το κατώφλι ϑ , εκπέμπεται spike μέσω της εξόδου V_{out} . **Κάτω:** Το διάγραμμα παρουσιάζει την αύξηση του δυναμικού εξαιτίας των εισερχόμενων ρευμάτων I_{in} , τη διαρροή μέσα από το χρόνο και την πυροδότηση spike όταν το κατώφλι ξεπερνιέται (Πηγή: [4],[5]).

Για να μπορεί να χρησιμοποιηθεί η μέθοδος της οπισθόδρομης διάδοσης σφάλματος για την εκπαίδευση spiking νευρώνων, πρέπει να χρησιμοποιούμε surrogate συναρτήσεις ενεργοποίησης. Σε αυτήν την περίπτωση, χρησιμοποιήθηκε η παράγωγος της surrogate συνάρτησης, που ονομάζεται `spike_grad`, για να καθορίσουμε πώς θα πρέπει να μεταδίδεται το σφάλμα όταν η έξοδος είναι δυαδική (spike ή όχι).

Το δίκτυο εκπαιδεύτηκε για 2 έως 10 epochs, ανάλογα το σενάριο και τις παραμέτρους που χρησιμοποιήθηκαν. Ο βελτιστοποιητής Adam χρησιμοποιήθηκε για την εκπαίδευση, με

σταθερό ρυθμό μάθησης που ποικίλλε μεταξύ 0,001 και 0,005 ανάλογα με τις surrogate και loss function.

Στην περίπτωση του υπολογισμού του σφάλματος χρησιμοποιήθηκαν τρεις βασικές συναρτήσεις απώλειας. Η πρώτη, `SF.ce_count_loss`, εφαρμόζει cross-entropy στους αριθμούς των spikes για κάθε έξοδο. Η δεύτερη, `SF.mse_count_loss`, υπολογίζει το mean square error μεταξύ των spikes και των στόχων (80% για τους σωστούς νευρώνες και 20% για τους λάθος), χρησιμοποιώντας μια παραλλαγή του standard deviation. Η τρίτη, `SF.mse_membrane_loss`, ελέγχει το MSE της δυναμικής της μεμβράνης αντί των spikes, προσφέροντας διαφορετική οπτική στη βελτιστοποίηση.

Η αξιολόγηση στο test set έγινε χρησιμοποιώντας τη μέθοδο του rate-coded output, όπου τελικής απόφασης είναι η κλάση του νευρώνα εξόδου που είχε τις περισσότερες ακίδες. Συγχρόνως, υπολογίστηκαν διάφορες στατιστικές μετρήσεις που βοηθούν στην ερμηνεία της συμπεριφοράς του δικτύου. Αυτές περιλαμβάνουν τη μέση ακρίβεια εκπαίδευσης και επικύρωσης, το πλήθος των αιχμών ανά δείγμα, την αραιότητα (sparsity) ως το ποσοστό νευρώνων που παρέμειναν αδρανείς, τη σταθερότητα της εκπαίδευσης με βάση τη διακύμανση της ακρίβειας στις τελευταίες επαναλήψεις, καθώς και την εκτίμηση της σύγκλισης, όταν παρατηρείται σταθεροποίηση της απόδοσης εντός συγκεκριμένου αριθμού επαναλήψεων.

Αυτές οι τεχνικές έχουν παρουσιαστεί εκτενώς στη βιβλιογραφία των εργαλείων SNN και έχουν εκτελεστεί με οδηγό τα documentations του SnnTorch (2025).

ΚΕΦΑΛΑΙΟ 4^ο

Πειραματικό Μέρος

4.1 Πίνακες Πειραμάτων

Στο παρόν κεφάλαιο, παρουσιάζονται λεπτομερώς τα αποτελέσματα των πειραμάτων που διενεργήθηκαν στα πλαίσια της παρούσας έρευνας με στόχο να αξιολογηθεί η απόδοση των Spiking Neural Networks (SNNs) υπό διάφορες παραμέτρους. Οι πίνακες που ακολουθούν κατηγοριοποιούνται ανά θέμα με βάση μία κύρια μεταβλητή κάθε φορά, όπως η surrogate function που χρησιμοποιήθηκε, η συνάρτηση απώλειας (loss function), η αρχιτεκτονική του δικτύου και ο ρυθμός μάθησης (learning rate).

Scenario	Surrogate	Parameters	Learning Rate	Architecture	Loss Function	Batch Size	Train Loss	Train Accuracy	Test Accuracy	Avg S/s	Sparsity %
1	FastSigmoid	slope=25	0,001	Conv(2→12→32)	SF.mse_count_loss	128	4,5466	78,12	86,66	802	0
2	FastSigmoid	slope=50	0,001	Conv(2→12→32)	SF.mse_count_loss	128	3,4051	90,62	91,79	789,8	0
3	ATan	alpha=2	0,001	Conv(2→12→32)	SF.mse_count_loss	128	2,8032	94,79	91,46	792,75	0
4	ATan	alpha=5	0,001	Conv(2→12→32)	SF.mse_count_loss	128	2,9791	93,75	91,76	788,1	0
5	Sigmoid	slope=25	0,001	Conv(2→12→32)	SF.mse_count_loss	128	12,6143	8,33	10,32	700	10
6	SpikeRateEscape	beta=1, slope=25	0,001	Conv(2→12→32)	SF.mse_count_loss	128	2,6233	94,79	94,39	805,56	0
7	FastSigmoid	slope=75	0,001	Conv(2→12→32)	SF.mse_count_loss	128	4,2297	85,42	86,02	797,05	0
8	ATan	alpha=10	0,001	Conv(2→12→32)	SF.mse_count_loss	128	3,4776	89,58	88,5	777,9	0
9	SpikeRateEscape	beta=1, slope=25	0,002	Conv(2→12→32)	SF.mse_count_loss	128	1,6128	98,96	95,95	803,28	0
10	SpikeRateEscape	beta=1, slope=25	0,005	Conv(2→12→32)	SF.mse_count_loss	128	1,6997	96,88	96,03	831,08	0
11	FastSigmoid	slope=50	0,002	Conv(2→12→32)	SF.mse_count_loss	128	2,8094	94,79	94,35	801,85	0
12	FastSigmoid	slope=50	0,005	Conv(2→12→32)	SF.mse_count_loss	128	1,8805	96,88	95,83	784,71	0
13	ATan	alpha=2	0,002	Conv(2→12→32)	SF.mse_count_loss	128	2,2953	95,83	94,97	803,64	0
14	ATan	alpha=2	0,005	Conv(2→12→32)	SF.mse_count_loss	128	1,7798	93,75	96,06	811,92	0
15	SpikeRateEscape	beta=0.5, slope=25	0,005	Conv(2→24→48)	SF.mse_count_loss	128	1,1508	98,96	97,71	801,16	0
16	SpikeRateEscape	beta=0.7, slope=25	0,005	Conv(2→24→48)	SF.mse_count_loss	128	0,8139	96,88	97,98	810,17	0
17	ATan	alpha=2	0,005	Conv(2→24→48)	SF.mse_count_loss	128	1,0269	97,92	97,77	809,86	0
18	FastSigmoid	slope=50	0,005	Conv(2→24→48)	SF.mse_count_loss	128	1,1306	97,92	97,57	791,06	0
19	SpikeRateEscape	beta=0.7, slope=25	0,005	Conv(2→24→48)	SF.mse_count_loss	64	0,9231	96,88	98,3	789,19	0
20	ATan	alpha=2	0,005	Conv(2→24→48)	SF.mse_count_loss	64	0,6851	100	98,24	774,38	0
21	FastSigmoid	slope=50	0,005	Conv(2→24→48)	SF.mse_count_loss	64	0,581	100	97,98	787,62	0
22	SpikeRateEscape	beta=0.7, slope=25	0,005	Conv(2→24→48)	SF.ce_rate_loss()	64	1,637	96,88	95,58	484,75	19,38
23	ATan	alpha=2	0,005	Conv(2→24→48)	SF.ce_rate_loss()	64	1,6251	96,88	96,23	429,31	15,62
24	FastSigmoid	slope=50	0,005	Conv(2→24→48)	SF.ce_rate_loss()	64	1,7075	100	93,38	474,66	9,38
25	SpikeRateEscape	beta=0.7, slope=25	0,005	Conv(2→24→48)	SF.ce_count_loss()	64	0,0112	100	96,24	56,97	28,12
26	ATan	alpha=2	0,005	Conv(2→24→48)	SF.ce_count_loss()	64	0,2837	93,75	97,43	74,53	20
27	FastSigmoid	slope=50	0,005	Conv(2→24→48)	SF.ce_count_loss()	64	0,0911	96,88	95,59	22,03	71,25
28	SpikeRateEscap	beta=1, slope=25	0,005	Conv(2→24→48)	SF.mse_membrane_	64	0,0544	96,88	97,8	246,94	23,44
29	ATan	alpha=2	0,005	Conv(2→24→48)	SF.mse_memb_loss	64	0,0328	100	96,26	284,81	42,81
30	FastSigmoid	slope=50	0,005	Conv(2→24→48)	SF.mse_membrane_	64	0,0588	75	77,89	176,22	60

Πίνακας 1: Πίνακας των 30 πειραμάτων παραλλαγών σεναρίου

Η κατηγοριοποίηση αυτή επιτρέπει την προσεκτική ανάλυση του πώς επηρεάζει κάθε παράμετρος ξεχωριστά την τελική ακρίβεια του μοντέλου, τη συμπεριφορά των spikes και, γενικότερα, τη σταθερότητα και την αποδοτικότητα της διαδικασίας εκπαίδευσης. Κάθε πίνακας περιλαμβάνει συγκριτικά τα αποτελέσματα στο εκπαιδευτικό σύνολο δεδομένων και στο δοκιμαστικό σετ, καθώς επίσης και δευτερεύουσες μετρικές σε μερικούς εξ αυτών, όπως ο μέσος αριθμός spikes ανά δείγμα (Avg Spikes/Sample) και η αραιότητα (Sparsity).

Οι πίνακες συνοδεύονται από αναλυτικά σχόλια, προκειμένου να διευκολυνθεί η κατανόηση του περιεχομένου. Τα κύρια συμπεράσματα αναλύονται στο επόμενο κεφάλαιο.

4.2 Κατηγορία προσεγγιστικής συνάρτησης ενεργοποίησης

4.2.1 ATan

Ο παρακάτω πίνακας απεικονίζει τα αποτελέσματα των πειραμάτων με χρήση της της surrogate συνάρτησης ATan, ενώ οι υπόλοιποι υπερπαράμετροι (της ρυθμός μάθησης, μέγεθος πακέτου και συνάρτηση απώλειας) είναι διαφορετικοί. Ο στόχος είναι να αξιολογηθεί η απόδοση της εν λόγω surrogate υπό διάφορες συνθήκες, έτσι ώστε να διαπιστωθεί η σταθερότητά της και η γενική της αποτελεσματικότητα στην εκπαίδευση SNNs.

Scenario	Surrogate	Parameters	Learning Rate	Architecture	Loss Function	Batch Size	Train Loss	Train Accuracy	Test Accuracy	Avg S/s	Sparsity %
3	ATan	alpha=2	0,001	Conv(2→12→32)	SF.mse_count_loss	128	2,8032	94,79	91,46	792,75	0
4	ATan	alpha=5	0,001	Conv(2→12→32)	SF.mse_count_loss	128	2,9791	93,75	91,76	788,1	0
8	ATan	alpha=10	0,001	Conv(2→12→32)	SF.mse_count_loss	128	3,4776	89,58	88,5	777,9	0
13	ATan	alpha=2	0,002	Conv(2→12→32)	SF.mse_count_loss	128	2,2953	95,83	94,97	803,64	0
14	ATan	alpha=2	0,005	Conv(2→12→32)	SF.mse_count_loss	128	1,7798	93,75	96,06	811,92	0
17	ATan	alpha=2	0,005	Conv(2→24→48)	SF.mse_count_loss	128	1,0269	97,92	97,77	809,86	0
20	Atan	alpha=2	0,005	Conv(2→24→48)	SF.mse_count_loss	64	0,6851	100	98,24	774,38	0
23	Atan	alpha=2	0,005	Conv(2→24→48)	SF.ce_rate_loss()	64	1,6251	96,88	96,23	429,31	15,62
26	Atan	alpha=2	0,005	Conv(2→24→48)	SF.ce_count_loss()	64	0,2837	93,75	97,43	74,53	20
29	Atan	alpha=2	0,005	Conv(2→24→48)	SF.mse_memb_loss	64	0,0328	100	96,26	284,81	42,81

Πίνακας 2: Πειραματικά αποτελέσματα με χρήση της συνάρτησης ATan ως surrogate gradient.

4.2.2 Fast Sigmoid

Ο παρακάτω πίνακας απεικονίζει τα αποτελέσματα των πειραμάτων με χρήση της της surrogate συνάρτησης Fast Sigmoid. Επικρατεί η ίδια λογική.

Scenario	Surrogate	Parameters	Learning Rate	Architecture	Loss Function	Batch Size	Train Loss	Train Accuracy	Test Accuracy	Avg S/s	Sparsity %
1	FastSigmoid	slope=25	0,001	Conv(2→12→32)	SF.mse_count_loss	128	4,5466	78,12	86,66	802	0
2	FastSigmoid	slope=50	0,001	Conv(2→12→32)	SF.mse_count_loss	128	3,4051	90,62	91,79	789,8	0
7	FastSigmoid	slope=75	0,001	Conv(2→12→32)	SF.mse_count_loss	128	4,2297	85,42	86,02	797,05	0
11	FastSigmoid	slope=50	0,002	Conv(2→12→32)	SF.mse_count_loss	128	2,8094	94,79	94,35	801,85	0
12	FastSigmoid	slope=50	0,005	Conv(2→12→32)	SF.mse_count_loss	128	1,8805	96,88	95,83	784,71	0
18	FastSigmoid	slope=50	0,005	Conv(2→24→48)	SF.mse_count_loss	128	1,1306	97,92	97,57	791,06	0
21	FastSigmoid	slope=50	0,005	Conv(2→24→48)	SF.mse_count_loss	64	0,581	100	97,98	787,62	0
24	FastSigmoid	slope=50	0,005	Conv(2→24→48)	SF.ce_rate_loss()	64	1,7075	100	93,38	474,66	9,38
27	FastSigmoid	slope=50	0,005	Conv(2→24→48)	SF.ce_count_loss()	64	0,0911	96,88	95,59	22,03	71,25
30	FastSigmoid	slope=50	0,005	Conv(2→24→48)	SF.mse_membrane_loss	64	0,0588	75	77,89	176,22	60

Πίνακας 3: Πειραματικά αποτελέσματα με χρήση της Fast Sigmoid.

4.2.3 SpikeRateEscape

Ο παρακάτω πίνακας περιλαμβάνει πειραματικά δεδομένα για τη χρήση της SpikeRateEscape ως surrogate συνάρτηση. Η λειτουργία της βασίζεται στη δυναμική διαφυγής της μεμβράνης και εξετάζεται η απόδοσή της υπό διαφορετικές τιμές υπερπαραμέτρων.

Scenario	Surrogate	Parameters	Learning Rate	Architecture	Loss Function	Batch Size	Train Loss	Train Accuracy	Test Accuracy	Avg S/s	Sparsity %
6	SRE	beta=1, slope=25	0,001	Conv(2→12→32)	SF.mse_count_loss	128	2,6233	94,79	94,39	805,56	0
9	SRE	beta=1, slope=25	0,002	Conv(2→12→32)	SF.mse_count_loss	128	1,6128	98,96	95,95	803,28	0
10	SRE	beta=1, slope=25	0,005	Conv(2→12→32)	SF.mse_count_loss	128	1,6997	96,88	96,03	831,08	0
15	SRE	beta=0.5, slope=25	0,005	Conv(2→24→48)	SF.mse_count_loss	128	1,1508	98,96	97,71	801,16	0
16	SRE	beta=0.7, slope=25	0,005	Conv(2→24→48)	SF.mse_count_loss	128	0,8139	96,88	97,98	810,17	0
19	SRE	beta=0.7, slope=25	0,005	Conv(2→24→48)	SF.mse_count_loss	64	0,9231	96,88	98,3	789,19	0
22	SRE	beta=0.7, slope=25	0,005	Conv(2→24→48)	SF.ce_rate_loss()	64	1,637	96,88	95,58	484,75	19,38
25	SRE	beta=0.7, slope=25	0,005	Conv(2→24→48)	SF.ce_count_loss()	64	0,0112	100	96,24	56,97	28,12
28	SRE	beta=1, slope=25	0,005	Conv(2→24→48)	SF.mse_membrane_loss	64	0,0544	96,88	97,8	246,94	23,44

Πίνακας 4: Αποτελέσματα με χρήση της SpikeRateEscape

4.3 Κατηγορία Αρχιτεκτονικής

4.3.1 Conv(2→12→32)

Ο πίνακας αυτός συνοψίζει τα αποτελέσματα των πειραμάτων που βασίζονται στη συνελικτική αρχιτεκτονική Conv2d(2→12→32), , με στόχο την αξιολόγηση της επίδρασης του βάθους και της πολυπλοκότητας του μοντέλου στην ακρίβεια και την αποδοτικότητα των SNNs.

Scenario	Architecture	Surrogate	Parameters	Learning Rate	Loss Function	Batch Size	Train Loss	Train Accuracy	Test Accuracy	Avg S/s	Sparsity %
1	Conv(2→12→32)	FastSigmoid	slope=25	0,001	SF.mse_count_loss	128	4,5466	78,12	86,66	802	0
2	Conv(2→12→32)	FastSigmoid	slope=50	0,001	SF.mse_count_loss	128	3,4051	90,62	91,79	789,8	0
3	Conv(2→12→32)	ATan	alpha=2	0,001	SF.mse_count_loss	128	2,8032	94,79	91,46	792,75	0
4	Conv(2→12→32)	ATan	alpha=5	0,001	SF.mse_count_loss	128	2,9791	93,75	91,76	788,1	0
5	Conv(2→12→32)	Sigmoid	slope=25	0,001	SF.mse_count_loss	128	12,6143	8,33	10,32	700	10
6	Conv(2→12→32)	SpikeRateEscape	beta=1, slope=25	0,001	SF.mse_count_loss	128	2,6233	94,79	94,39	805,56	0
7	Conv(2→12→32)	FastSigmoid	slope=75	0,001	SF.mse_count_loss	128	4,2297	85,42	86,02	797,05	0
8	Conv(2→12→32)	ATan	alpha=10	0,001	SF.mse_count_loss	128	3,4776	89,58	88,5	777,9	0
9	Conv(2→12→32)	SpikeRateEscape	beta=1, slope=25	0,002	SF.mse_count_loss	128	1,6128	98,96	95,95	803,28	0
10	Conv(2→12→32)	SpikeRateEscape	beta=1, slope=25	0,005	SF.mse_count_loss	128	1,6997	96,88	96,03	831,08	0
11	Conv(2→12→32)	FastSigmoid	slope=50	0,002	SF.mse_count_loss	128	2,8094	94,79	94,35	801,85	0
12	Conv(2→12→32)	FastSigmoid	slope=50	0,005	SF.mse_count_loss	128	1,8805	96,88	95,83	784,71	0
13	Conv(2→12→32)	ATan	alpha=2	0,002	SF.mse_count_loss	128	2,2953	95,83	94,97	803,64	0
14	Conv(2→12→32)	ATan	alpha=2	0,005	SF.mse_count_loss	128	1,7798	93,75	96,06	811,92	0

Πίνακας 5: Απόδοση SNNs με αρχιτεκτονική Conv(2→12→32).

4.3.2 Conv(2→24→48)

Στην συνέχεια στόν επόμενο πίνακα παρουσιάζονται τα αποτελέσματα για την αρχιτεκτονική Conv2d(2→24→48), η οποία είναι βαθύτερη και πιο σύνθετη σε σχέση με την προηγούμενη.

Scenario	Architecture	Surrogate	Parameters	Learning Rate	Loss Function	Batch Size	Train Loss	Train Accuracy	Test Accuracy	Avg S/s	Sparsity %
15	Conv(2→24→48)	SpikeRateEscape	beta=0.5, slope=25	0,005	SF.mse_count_loss	128	1,1508	98,96	97,71	801,16	0
16	Conv(2→24→48)	SpikeRateEscape	beta=0.7, slope=25	0,005	SF.mse_count_loss	128	0,8139	96,88	97,98	810,17	0
17	Conv(2→24→48)	ATan	alpha=2	0,005	SF.mse_count_loss	128	1,0269	97,92	97,77	809,86	0
18	Conv(2→24→48)	FastSigmoid	slope=50	0,005	SF.mse_count_loss	128	1,1306	97,92	97,57	791,06	0
19	Conv(2→24→48)	SpikeRateEscape	beta=0.7, slope=25	0,005	SF.mse_count_loss	64	0,9231	96,88	98,3	789,19	0
20	Conv(2→24→48)	Atan	alpha=2	0,005	SF.mse_count_loss	64	0,6851	100	98,24	774,38	0
21	Conv(2→24→48)	FastSigmoid	slope=50	0,005	SF.mse_count_loss	64	0,581	100	97,98	787,62	0
22	Conv(2→24→48)	SpikeRateEscape	beta=0.7, slope=25	0,005	SF.ce_rate_loss()	64	1,637	96,88	95,58	484,75	19,38
23	Conv(2→24→48)	Atan	alpha=2	0,005	SF.ce_rate_loss()	64	1,6251	96,88	96,23	429,31	15,62
24	Conv(2→24→48)	FastSigmoid	slope=50	0,005	SF.ce_rate_loss()	64	1,7075	100	93,38	474,66	9,38
25	Conv(2→24→48)	SpikeRateEscape	beta=0.7, slope=25	0,005	SF.ce_count_loss()	64	0,0112	100	96,24	56,97	28,12
26	Conv(2→24→48)	Atan	alpha=2	0,005	SF.ce_count_loss()	64	0,2837	93,75	97,43	74,53	20
27	Conv(2→24→48)	FastSigmoid	slope=50	0,005	SF.ce_count_loss()	64	0,0911	96,88	95,59	22,03	71,25
28	Conv(2→24→48)	SpikeRateEscap	beta=1, slope=25	0,005	SF.mse_membrane_loss	64	0,0544	96,88	97,8	246,94	23,44
29	Conv(2→24→48)	Atan	alpha=2	0,005	SF.mse_memb_loss	64	0,0328	100	96,26	284,81	42,81
30	Conv(2→24→48)	FastSigmoid	slope=50	0,005	SF.mse_membrane_loss	64	0,0588	75	77,89	176,22	60

Πίνακας 6: Επιδόσεις των SNNs με συνελκτική αρχιτεκτονική Conv(2→24→48).

4.4 Κατηγορία Συναρτήσεων Απώλειας

4.4.1 SF.ce_count_loss

Ο πίνακας παρουσιάζει τα αποτελέσματα που καταγράφηκαν από πειράματα που εκτελέστηκαν χρησιμοποιώντας τη συνάρτηση SF.ce_count_loss. Πρόκειται για έναν τρόπο υπολογισμού της cross-entropy, όπου η έξοδος κάθε νευρώνα αξιολογείται βάσει του αριθμού των spikes. Σκοπός αυτής της συνάρτησης είναι να προάγει την πιο αξιόπιστη ταξινόμηση των εξόδων με βάση τα spike counts, ενώ χρησιμοποιείται και για να εξεταστεί η ακρίβεια σε συστήματα χαμηλής πυκνότητας spikes.

Scenario	Loss Function	Surrogate	Parameters	Learning Rate	Architecture	Batch Size	Train Loss	Train Accuracy	Test Accuracy	Avg S/s	Sparsity %
25	SF.ce_count_loss()	SpikeRateEscape	beta=0.7, slope=25	0,005	Conv(2→24→48)	64	0,0112	100	96,24	56,97	28,12
26	SF.ce_count_loss()	Atan	alpha=2	0,005	Conv(2→24→48)	64	0,2837	93,75	97,43	74,53	20
27	SF.ce_count_loss()	FastSigmoid	slope=50	0,005	Conv(2→24→48)	64	0,0911	96,88	95,59	22,03	71,25

Πίνακας 7: Απόδοση των SNN δικτύων με χρήση της συνάρτησης απώλειας SF.ce_count_loss

4.4.2 SF.ce_rate_loss

Αυτός ο πίνακας περιέχει πειραματικά αποτελέσματα που προέκυψαν από τη χρήση της συνάρτησης SF.ce_rate_loss, η οποία υπολογίζει cross-entropy βάσει του συνολικού αριθμού spikes (rate-coded outputs) για κάθε έξοδο. Σκοπός είναι να διερευνηθεί η

επίδραση που έχει η χρήση αυτής της συνάρτησης, η οποία προσεγγίζει καλύτερα την πιθανότητα πυροδότησης, στην ακρίβεια και τη αραιότητα των SNNs.

Scenario	Loss Function	Surrogate	Parameters	Learning Rate	Architecture	Batch Size	Train Loss	Train Accuracy	Test Accuracy	Avg S/s	Sparsity %
22	SF.ce_rate_loss()	SpikeRateEscape	beta=0.7, slope=25	0,005	Conv(2→24→48)	64	1,637	96,88	95,58	484,75	19,38
23	SF.ce_rate_loss()	Atan	alpha=2	0,005	Conv(2→24→48)	64	1,6251	96,88	96,23	429,31	15,62
24	SF.ce_rate_loss()	FastSigmoid	slope=50	0,005	Conv(2→24→48)	64	1,7075	100	93,38	474,66	9,38

Πίνακας 8: Επιδόσεις SNNs με χρήση της συνάρτησης απώλειας SF.ce_rate_loss.

4.4.3 SF.mse_count_loss

Ο επόμενος πίνακας συγκεντρώνει αποτελέσματα από πειράματα στα οποία εφαρμόστηκε η συνάρτηση απώλειας SF.mse_count_loss. Η συνάρτηση αυτή βασίζεται στον μέσο τετραγωνικό σφάλμα (Mean Squared Error) στον αριθμό των spikes που παράγονται από κάθε νευρώνα εξόδου. Η χρήση της προϋποθέτει στόχους spike rate (π.χ. 80% για σωστή κλάση, 20% για τις υπόλοιπες), και η απόδοσή της αξιολογείται σε διαφορετικά σενάρια surrogate συναρτήσεων, learning rate και batch size.

Scenario	Loss Function	Surrogate	Parameters	Learning Rate	Architecture	Batch Size	Train Loss	Train Accuracy	Test Accuracy	Avg S/s	Sparsity %
1	SF.mse_count_loss	FastSigmoid	slope=25	0,001	Conv(2→12→32)	128	4,5466	78,12	86,66	802	0
2	SF.mse_count_loss	FastSigmoid	slope=50	0,001	Conv(2→12→32)	128	3,4051	90,62	91,79	789,8	0
3	SF.mse_count_loss	ATan	alpha=2	0,001	Conv(2→12→32)	128	2,8032	94,79	91,46	792,75	0
4	SF.mse_count_loss	ATan	alpha=5	0,001	Conv(2→12→32)	128	2,9791	93,75	91,76	788,1	0
5	SF.mse_count_loss	Sigmoid	slope=25	0,001	Conv(2→12→32)	128	12,6143	8,33	10,32	700	10
6	SF.mse_count_loss	SpikeRateEscape	beta=1, slope=25	0,001	Conv(2→12→32)	128	2,6233	94,79	94,39	805,56	0
7	SF.mse_count_loss	FastSigmoid	slope=75	0,001	Conv(2→12→32)	128	4,2297	85,42	86,02	797,05	0
8	SF.mse_count_loss	ATan	alpha=10	0,001	Conv(2→12→32)	128	3,4776	89,58	88,5	777,9	0
9	SF.mse_count_loss	SpikeRateEscape	beta=1, slope=25	0,002	Conv(2→12→32)	128	1,6128	98,96	95,95	803,28	0
10	SF.mse_count_loss	SpikeRateEscape	beta=1, slope=25	0,005	Conv(2→12→32)	128	1,6997	96,88	96,03	831,08	0
11	SF.mse_count_loss	FastSigmoid	slope=50	0,002	Conv(2→12→32)	128	2,8094	94,79	94,35	801,85	0
12	SF.mse_count_loss	FastSigmoid	slope=50	0,005	Conv(2→12→32)	128	1,8805	96,88	95,83	784,71	0
13	SF.mse_count_loss	ATan	alpha=2	0,002	Conv(2→12→32)	128	2,2953	95,83	94,97	803,64	0
14	SF.mse_count_loss	ATan	alpha=2	0,005	Conv(2→12→32)	128	1,7798	93,75	96,06	811,92	0
15	SF.mse_count_loss	SpikeRateEscape	beta=0.5, slope=25	0,005	Conv(2→24→48)	128	1,1508	98,96	97,71	801,16	0
16	SF.mse_count_loss	SpikeRateEscape	beta=0.7, slope=25	0,005	Conv(2→24→48)	128	0,8139	96,88	97,98	810,17	0
17	SF.mse_count_loss	ATan	alpha=2	0,005	Conv(2→24→48)	128	1,0269	97,92	97,77	809,86	0
18	SF.mse_count_loss	FastSigmoid	slope=50	0,005	Conv(2→24→48)	128	1,1306	97,92	97,57	791,06	0
19	SF.mse_count_loss	SpikeRateEscape	beta=0.7, slope=25	0,005	Conv(2→24→48)	64	0,9231	96,88	98,3	789,19	0
20	SF.mse_count_loss	Atan	alpha=2	0,005	Conv(2→24→48)	64	0,6851	100	98,24	774,38	0
21	SF.mse_count_loss	FastSigmoid	slope=50	0,005	Conv(2→24→48)	64	0,581	100	97,98	787,62	0

Πίνακας 9: Αποτελέσματα με χρήση της συνάρτησης απώλειας SF.mse_count_loss.

4.4.4 SF.mse_membrane_loss

Εδώ φαίνονται πειράματα που χρησιμοποιούν το SF.mse_membrane_loss. Αντί να βασίζεται στα spikes για να υπολογίσει το σφάλμα, αυτή η συνάρτηση εκτιμά τη διαφορά του δυναμικού της μεμβράνης των νευρώνων. Αυτό επιτρέπει την προσομοίωση της βελτιστοποίησης σε πολύ πρώιμα στάδια της νευρωνικής δικτύωσης, προσφέροντας έναν διαφορετικό τρόπο εκπαίδευσης των SNNs, με στόχο τη μείωση αραιότητας και την ενίσχυση της σταθερότητας.

Scenario	Loss Function	Surrogate	Parameters	Learning Rate	Architecture	Batch Size	Train Loss	Train Accuracy	Test Accuracy	Avg S/s	Sparsity %
28	SF.mse_membrane_loss	SpikeRateEscap	beta=1, slope=25	0,005	Conv(2→24→48)	64	0,0544	96,88	97,8	246,94	23,44
30	SF.mse_membrane_loss	FastSigmoid	slope=50	0,005	Conv(2→24→48)	64	0,0588	75	77,89	176,22	60
29	SF.mse_memb_loss	Atan	alpha=2	0,005	Conv(2→24→48)	64	0,0328	100	96,26	284,81	42,81

Πίνακας 10 : Αποτελέσματα πειραμάτων με συνάρτηση SF.mse_membrane_loss.

4.5 Υπερπαραμέτρος Ρυθμού εκμάθησης

Οι παρακάτω πίνακες εξετάζουν την επίδραση της υπερπαραμέτρου learning rate (ρυθμού μάθησης) στην απόδοση των SNNs.

Scenario	Learning Rate	Surrogate	Params	Loss Function	Architecture	Train Acc	Test Acc
1	0,001	FastSigmoid	slope=25	SF.mse_count_loss	Conv(2→12→32)	78,12	86,66
2	0,001	FastSigmoid	slope=50	SF.mse_count_loss	Conv(2→12→32)	90,62	91,79
3	0,001	ATan	alpha=2	SF.mse_count_loss	Conv(2→12→32)	94,79	91,46
4	0,001	ATan	alpha=5	SF.mse_count_loss	Conv(2→12→32)	93,75	91,76
5	0,001	Sigmoid	slope=25	SF.mse_count_loss	Conv(2→12→32)	8,33	10,32
6	0,001	SpikeRateEscape	beta=1, slope=25	SF.mse_count_loss	Conv(2→12→32)	94,79	94,39
7	0,001	FastSigmoid	slope=75	SF.mse_count_loss	Conv(2→12→32)	85,42	86,02
8	0,001	ATan	alpha=10	SF.mse_count_loss	Conv(2→12→32)	89,58	88,5

Πίνακας 11 : Πειραματικές παραλλαγές με learning rate 0.001.

Scenario	Learning Rate	Surrogate	Params	Loss Function	Architecture	Train Acc	Test Acc
9	0,002	SpikeRateEscape	beta=1, slope=25	SF.mse_count_loss	Conv(2→12→32)	98,96	95,95
11	0,002	FastSigmoid	slope=50	SF.mse_count_loss	Conv(2→12→32)	94,79	94,35
13	0,002	ATan	alpha=2	SF.mse_count_loss	Conv(2→12→32)	95,83	94,97

Πίνακας 12: Πειραματικές παραλλαγές με learning rate 0.002.

Scenario	Learning Rate	Surrogate	Params	Loss Function	Architecture	Train Acc	Test Acc
10	0,005	SpikeRateEscape	beta=1, slope=25	SF.mse_count_loss	Conv(2→12→32)	96,88	96,03
12	0,005	FastSigmoid	slope=50	SF.mse_count_loss	Conv(2→12→32)	96,88	95,83
14	0,005	ATan	alpha=2	SF.mse_count_loss	Conv(2→12→32)	93,75	96,06
15	0,005	SpikeRateEscape	beta=0.5, slope=25	SF.mse_count_loss	Conv(2→24→48)	98,96	97,71
16	0,005	SpikeRateEscape	beta=0.7, slope=25	SF.mse_count_loss	Conv(2→24→48)	96,88	97,98
17	0,005	ATan	alpha=2	SF.mse_count_loss	Conv(2→24→48)	97,92	97,77
18	0,005	FastSigmoid	slope=50	SF.mse_count_loss	Conv(2→24→48)	97,92	97,57
19	0,005	SpikeRateEscape	beta=0.7, slope=25	SF.mse_count_loss	Conv(2→24→48)	96,88	98,3
20	0,005	ATan	alpha=2	SF.mse_count_loss	Conv(2→24→48)	100	98,24
21	0,005	FastSigmoid	slope=50	SF.mse_count_loss	Conv(2→24→48)	100	97,98

Πίνακας 13: Πειραματικές παραλλαγές με learning rate 0.005.

ΚΕΦΑΛΑΙΟ 5°

Αξιολόγηση και Συζήτηση Αποτελεσμάτων

5.1 Πειραματική μεθοδολογία και παραλλαγές

Για την συστηματική αξιολόγηση των Spiking Neural Networks (SNNs), έγινε μια σειρά πειραμάτων, όπου αλλάζουμε μια κρίσιμη υπερπαραμέτρο κάθε φορά και κρατάμε τις υπόλοιπες σταθερές, έτσι ώστε να μπορούμε να δούμε την επίδραση κάθε παράγοντα στην απόδοση του μοντέλου.

Συγκεκριμένα, ακολουθήθηκε η παρακάτω μεθοδολογία:

Αλλαγή Surrogate Gradient: Δοκιμάσαμε πολλές διαφορετικές συναρτήσεις ως προσεγγιστικές παραγώγους (π.χ. Fast Sigmoid, ATan, SpikeRateEscape, Sigmoid) και κρατήσαμε σταθερές τις υπόλοιπες παραμέτρους: αρχιτεκτονική Conv(2→24→48), ρυθμός μάθησης 0.005, μέγεθος batch 64, και συνάρτηση απώλειας ce_count_loss.

Αλλαγή Loss Function: Χρησιμοποιήσαμε τέσσερις διαφορετικές συναρτήσεις απώλειας (ce_count_loss, mse_count_loss, mse_membrane_loss, ce_rate_loss) κρατώντας σταθερές τις υπόλοιπες παραμέτρους (surrogate: SpikeRateEscape, αρχιτεκτονική: Conv(2→24→48), ρυθμό μάθησης: 0.005, μέγεθος batch: 64).

Αλλαγή Αρχιτεκτονικής: Συγκρίναμε δύο διαφορετικές αρχιτεκτονικές, Conv(2→12→32) και Conv(2→24→48), κάτω από ίδιες συνθήκες surrogate (ATan), loss (mse_count_loss), ρυθμό μάθησης (0.005), και μέγεθος batch (64).

Αλλαγή Learning Rate: Δοκιμάσαμε τις τιμές 0.001, 0.002 και 0.005, κρατώντας σταθερές τις υπόλοιπες παραμέτρους (surrogate: SpikeRateEscape, loss: ce_count_loss, αρχιτεκτονική: Conv(2→24→48), μέγεθος batch: 64).

Η διαδικασία αυτή οργανώθηκε σε 30 σενάρια (πειράματα), όπου κάθε σενάριο αντιστοιχούσε σε έναν συνδυασμό υπερπαραμέτρων. Τα αποτελέσματα συγκεντρώθηκαν σε συγκριτικούς πίνακες (βλ. Κεφάλαιο 4), με σκοπό να βγουν ποιοτικά και ποσοτικά συμπεράσματα.

Το παρακάτω κείμενο αφορά τις αλλαγές που έγιναν εκ προθέσεως και τα αποτελέσματα που προέκυψαν από αυτές.

5.2 Επίδραση του Surrogate Gradient

Ο τρόπος με τον οποίο η συνάρτηση surrogate gradient είναι διαμορφωμένη φαίνεται να έχει καθοριστική σημασία για τη δυνατότητα ενός νευρωνικού δικτύου να μάθει με επιτυχία.

Η επιλογή της surrogate function έχει απευθείας επίδραση στη σταθερότητα και την ταχύτητα εκπαίδευσης των SNN. Σύμφωνα με το Neftci et al. (2019), η χρήση προσεγγιστικών διαφοροποιήσεων επιτρέπει να εφαρμοστεί το backpropagation σε δίκτυα με δυαδικές, άσημες εξόδους. Επιπλέον, σύμφωνα με τους Eshraghian et al. (2023), οι "μαλακές" συναρτήσεις, όπως η fast sigmoid, οδηγούν σε καλύτερη διάδοση του σήματος και βοηθούν στο να μην "νεκρώνουν" νευρώνες κατά την εκπαίδευση.

Πράγματι από τα αποτελέσματα που προέκυψαν:

Το SpikeRateEscape και το ATan ξεχωρίζουν για τη σταθερότητά τους και την ακρίβειά τους καθώς κατάφεραν να πετύχουν ποσοστά επιτυχίας που φτάνουν έως και 98.3% (Σενάριο 19 με Fast Sigmoid και Σενάριο 20 με ATan). Η fast Sigmoid, ορισμένες φορές, όταν χρησιμοποιείται μαζί με την κατάλληλη αρχιτεκτονική και τον κατάλληλο ρυθμό εκπαίδευσης (π.χ. Σενάριο 21), παρουσιάζει επίσης πολύ υψηλή απόδοση (97.98% test accuracy). Ένα αντίθετο παράδειγμα αποτελεί η Sigmoid με τιμή slope=25 (Σενάριο 5) καθώς εμφάνισε πολύ χαμηλή απόδοση (10,32%), πράγμα που σημαίνει ότι η επιλογή των παραμέτρων slope/beta έχει κρίσιμη σημασία για την απόδοση.

Επομένως, επιβεβαιώνεται η θεωρητική πρόταση ότι πιο «ομαλές» συναρτήσεις έχουν ως αποτέλεσμα την πιο σταθερή εκπαίδευση και τα καλύτερα αποτελέσματα.

5.3 Επίδραση της Συνάρτησης Απώλειας

Όπως προκύπτει, η επιλογή της συνάρτησης κόστους επηρεάζει τόσο την τελική ακρίβεια όσο και τα χαρακτηριστικά συμπεριφοράς του δικτύου:

Η Cross-Entropy με spike counts (SF.ce_count_loss) απέδιδε σταθερά και είχε υψηλή ακρίβεια σε πολλά σενάρια, ιδιαίτερα σε συνδυασμό με Conv(2→24→48). Για παράδειγμα, το Σενάριο 25 (SpikeRateEscape + ce_count_loss) έφτανε το 96,24%, έχοντας πολύ χαμηλή απώλεια κατά την εκπαίδευση (0,0112). Η MSE με spike counts (SF.mse_count_loss) είχε εξίσου καλή επίδοση και η υψηλότερη ακρίβεια παρατηρήθηκε στο Σενάριο 20 (ATan + mse_count_loss) με 98,24%. Η MSE με δυναμικό (SF.mse_membrane_loss) ήταν αποδοτική σε συνδυασμό με κατάλληλες τιμές βαρών, αλλά βραχυκύκλωνε το δίκτυο (π.χ. Σενάριο 30: μόλις 176,22 spikes/δείγμα).

Συνολικά, η απώλεια CE αποδείχθηκε λίγο πιο αποτελεσματική σε προβλήματα καθαρής ταξινόμησης, όπως ήταν αναμενόμενο καιρού έκαστου, ενώ η απώλεια spike της MSE διατήρησε την καλύτερη ισορροπία μεταξύ τάσης ενεργοποίησης και ακρίβειας.

5.4 Επίδραση της Αρχιτεκτονικής

Παρατηρείται ότι η αρχιτεκτονική Conv(2→24→48) έχει καλύτερες επιδόσεις σε σχέση με την πιο απλή Conv(2→12→32):

Τα περισσότερα από τα πειράματα με την αρχιτεκτονική Conv(2→24→48) είχαν ποσοστό επιτυχίας >97% και υπήρχαν ακόμα και περιπτώσεις όπου η ακρίβεια ήταν σταθερά υψηλή κατά τη διάρκεια της εκπαίδευσης του μοντέλου. Αντίθετα, η Conv(2→12→32) είχε μεγαλύτερη διακύμανση και χαμηλότερες μέγιστες επιδόσεις (π.χ. Σενάρια 1–10). Η αύξηση

της χωρητικότητας με περισσότερα φίλτρα και feature maps βελτίωσε τη δυνατότητα του μοντέλου να αναγνωρίζει χρονο-χωρικά μοτίβα.

5.5 Επίδραση Ρυθμού Μάθησης και Μέγεθος Παρτίδας

Τα πειραματικά αποτελέσματα δείχνουν ότι το ποσοστό μάθησης είναι με 0,005 υψηλότερο, δηλαδή, τα SNN δίνουν καλύτερες επιδόσεις, ιδίως όταν δουλεύουν με βελτιστοποιημένες υποκατάστατες συναρτήσεις.

Όταν το batch size είναι 64 φαίνεται πως υπάρχει μεγαλύτερη σταθερότητα σε σχέση με το 128, καθώς, ενδεχομένως, τα βάρη ενημερώνονται πιο συχνά και παρουσιάζουν μικρότερες διακυμάνσεις.

5.6 Sparsity και Ενεργειακή Απόδοση

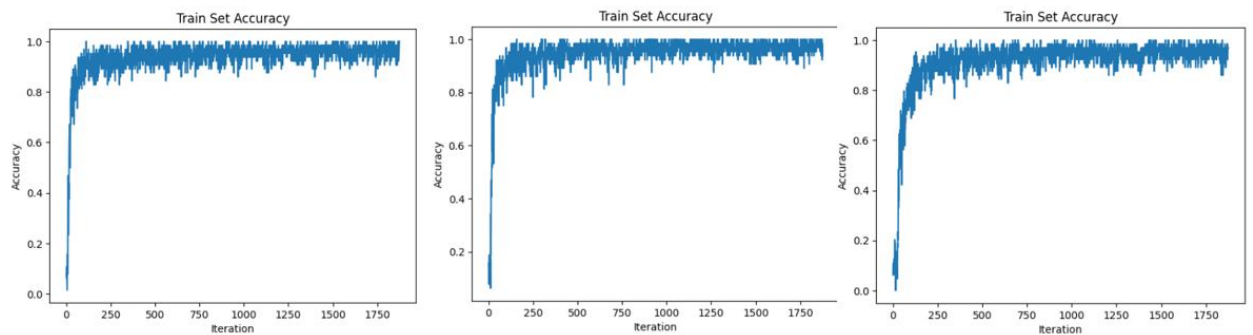
Οι Tavanaei et al.,(2019) επισημαίνουν ότι μετρήσεις όπως η sparsity και ο μέσος ρυθμός πυροδότησης διαδραματίζουν σημαντικό ρόλο στην ενεργειακή απόδοση των SNN. Όταν η sparsity είναι υψηλή, τότε λιγότερες μονάδες ενεργοποιούνται συνολικά στο δίκτυο, με αποτέλεσμα την χαμηλότερη κατανάλωση ενέργειας και την αύξηση της αξιοπιστίας λειτουργίας του.

Η συχνότητα των spikes διαφοροποιείται ανάλογα με τη μέθοδο απώλειας των δεδομένων:

Τα μοντέλα που χρησιμοποιούν `mse_membrane_loss` ή `ce_count_loss` είχαν τα περισσότερα spikes (π.χ. Σενάριο 30: 60%, Σενάριο 27: 71.25%). Αντίθετα, τα μοντέλα που χρησιμοποιούν `mse_count_loss` παρήγαγαν περισσότερα spikes ανά δείγμα (π.χ. ~800 spikes/sample), δείχνοντας ότι καταναλώνουν περισσότερη ενέργεια.

Η βέλτιστη ισορροπία ανάμεσα στην απόδοση και τη συχνότητα των spikes, φαίνεται πως επιτυγχάνεται στα μοντέλα όπως το 26 (ATan + ce_count_loss), με 97.43% ακρίβεια, ~74 spikes/sample και 20% sparsity, τα οποία θεωρούνται ιδανικά για εφαρμογές σε νευρομορφικά συστήματα χαμηλής κατανάλωσης.

5.7 Συγκριτική Ανάλυση Καμπυλών Εκπαίδευσης



Εικόνα 9: Απεικόνιση της ακρίβειας της εκπαίδευσης (Train Set Accuracy), από τα αριστερά προς τα δεξιά με την σειρά SpikeRateEscape-Atan-Fast_Sigmoid, συναρτήσεων των επαναλήψεων, για τρεις παραλλαγές surrogate gradients υπό τις ίδιες πειραματικές συνθήκες (learning rate = 0.005, batch size = 64, αρχιτεκτονική Conv(2→24→48), loss function).

Για να ενισχυθεί η ποιότητα της αξιολόγησης των πειραματικών αποτελεσμάτων, παρουσιάζονται οι καμπύλες ακρίβειας εκπαίδευσης (train set accuracy) για τρεις χαρακτηριστικές παραλλαγές των πειραμάτων, οι οποίες διαφέρουν στη συνάρτηση surrogate gradient: Fast Sigmoid, SpikeRateEscape και ATan.

Οι καμπύλες δείχνουν την πορεία της ακρίβειας εκπαίδευσης όσον αφορά στις επαναλήψεις (iterations). Παρατηρείται ότι όλες οι συναρτήσεις καταλήγουν σε υψηλή ακρίβεια (>95%), ωστόσο παρουσιάζονται διαφορές στην ποιότητα της μάθησης και στη σταθερότητα των τιμών τους.

Συγκεκριμένα, η SpikeRateEscape έχει τη γρηγορότερη άνοδο και συνοδεύεται από σχεδόν σταθερή τιμή κοντά στο 97% ήδη από τα πρώτα 500 βήματα, που υποδηλώνει αποτελεσματική και σταθερή μάθηση. Η ATan επίσης έχει ικανοποιητική απόδοση, με πιο ήπια σύγκλιση, αλλά με μικρότερες διακυμάνσεις στο τέλος της. Αντίθετα, η Fast Sigmoid έχει ελαφρώς πιο ασταθή συμπεριφορά κατά τη διάρκεια της εκπαίδευσης, με έντονη διακύμανση γύρω από το 95%, που υποδηλώνει ότι η εν λόγω surrogate συνάρτηση, παρότι είναι αποτελεσματική, ενδέχεται να μην εξασφαλίζει τη σταθερότητα της μάθησης όπως οι άλλες.

Αυτά τα ευρήματα συμφωνούν με τις γενικές παρατηρήσεις του πειραματικού πίνακα, επιβεβαιώνοντας την υπόθεση ότι ο τύπος του surrogate gradient έχει σημαντική επίδραση στη συμπεριφορά της σύγκλισης και στη σταθερότητα της εκπαίδευσης στο SNN.

5.8 Συνολική Τελική Αξιολόγηση

Τα αποτελέσματα είναι ικανοποιητικά και δείχνουν ότι ένα SNN μπορεί να έχει τέτοιες ή και καλύτερες επιδόσεις από συμβατικά ANNs στο N-MNIST, αν επιλεγούν σωστά surrogate function, loss function και αρχιτεκτονική. Να σημειωθεί ότι σε αρκετές περιπτώσεις η επίδοση στο Test ήταν άνω του 98%, πράγμα που αποδεικνύει ότι τα SNN μπορούν να λειτουργούν αποτελεσματικά στο supervised learning μέσω surrogate gradients. (Neftci et al., 2019; Eshraghian et al., 2023).

Το γεγονός ότι τα SNN έχουν υψηλή αποδοτικότητα και χρησιμοποιούνται σε περιβάλλοντα με αυστηρούς περιορισμούς κατανάλωσης (όπως σε ενσωματωμένα IoT συστήματα ή σε νευρομορφικά τσιπ) οφείλεται στην αραιότητά τους και τη χαμηλή μέση ενεργοποίηση. Επιπλέον, η πληθώρα των διαθέσιμων συναρτήσεων κόστους και των μηχανισμών εκπαίδευσης είναι εγγύηση ότι το μοντέλο μπορεί να προσαρμοστεί με ευελιξία σε διάφορους στόχους.

ΚΕΦΑΛΑΙΟ 6°

Συμπεράσματα και Μελλοντική Εργασία

Η προκείμενη μελέτη εξέτασε εκτενώς τη διαδικασία εκπαίδευσης και την απόδοση των Νευρωνικών Δικτύων Όσεων (Spiking Neural Networks - SNNs) επί ενός προβλήματος ταξινόμησης εικόνων, χρησιμοποιώντας το neuromorphic dataset N-MNIST. Αναλύσαμε τον τρόπο με τον οποίο επηρεάζονται η απόδοση και η σταθερότητα των SNNs από διάφορες παραμέτρους όπως οι surrogate συναρτήσεις κλίσης, οι αρχιτεκτονικές τους καθώς και οι συναρτήσεις απώλειας, μέσα από τη διεξαγωγή 30 πειραμάτων με διάφορες εκδοχές.

Τα πειραματικά αποτελέσματα κατέδειξαν ότι η επιλογή των surrogate συναρτήσεων επηρεάζει σημαντικά τόσο την απόδοση όσο και τη σταθερότητα της διαδικασίας εκπαίδευσης του δικτύου. Οι ομαλές συναρτήσεις, όπως η SpikeRateEscape και η ArcTan, παρουσίασαν κατά μέσο όρο υψηλότερη ακρίβεια ταξινόμησης και καλύτερη σύγκλιση, σε αντίθεση με τις πιο απότομες προσεγγίσεις. Συγκεκριμένα, η SpikeRateEscape με $\text{slope}=25$ και η ArcTan με $\alpha=2$, σε συνδυασμό με κατάλληλες παραμέτρους, φτάσανε ακρίβεια τάξης του 98% στο test set, διατηρώντας ταυτόχρονα το SNN αραιό.

Επιπλέον, επιβεβαιώθηκε ότι η επιλογή της συνάρτησης απώλειας παίζει καθοριστικό ρόλο. Τα spike counts σε συνδυασμό με τη Cross-Entropy συνήθως οδηγούσαν σε ελαφρώς καλύτερη ακρίβεια και σε πιο ομαλή διάκριση κατηγοριών σε σχέση με την MSE-based loss, παραδειγματίζοντας την ήδη υπάρχουσα γνώση από τη βιβλιογραφία ότι η CE αποβλέπει απευθείας στην αύξηση της πιθανότητας να ταξινομηθεί σωστά ένα δείγμα (Tavanaei et al., 2019; Neftci et al., 2019). Ακόμα, τα πειράματα έδειξαν ότι οι μέθοδοι που προωθούν το sparsity και τον αραιό προγραμματισμό μπορεί να οδηγήσουν σε σημαντική μείωση του μέσου αριθμού των spikes, εντός όμως υψηλών επιπέδων ταξινόμησης.

Η αξιολόγηση μετρήσεων όπως το sparsity (δηλαδή η «αραιή» ενεργοποίηση των νευρώνων), ο μέσος αριθμός spikes ανά δείγμα, η σταθερότητα της εκπαίδευσης και η ταχύτητα σύγκλισης Δύναται να οδηγήσει στην καλύτερη κατανόηση της αποδοτικότητας κάθε πειράματος. Σημαντικό είναι ότι ακόμα και στις περιπτώσεις όπου επιτυγχάνονται ιδιαίτερες υψηλές ακρίβειες, ο μέσος αριθμός των spikes ανά δείγμα μπορεί να είναι μικρότερος των 800, κάτι που καθιστά τις παραπάνω προσεγγίσεις ιδιαίτερες κατάλληλες για την υλοποίηση σε νευρομορφικό εξοπλισμό χαμηλής κατανάλωσης ενέργειας.

Μελλοντική Εργασία

Αν και τα αποτελέσματα που προκύπτουν από την παρούσα μελέτη είναι ενθαρρυντικά, υπάρχει περιθώριο για περαιτέρω έρευνα.

Συγκεκριμένα:

Νέες Τεχνικές Κωδικοποίησης: Στη μελέτη αυτή χρησιμοποιήθηκαν δεδομένα που είχαν ήδη κωδικοποιηθεί σε μορφή spiking (DVS από το N-MNIST). Θα ήταν ενδιαφέρον να δοκιμαστούν κι άλλες τεχνικές κωδικοποίησης (π.χ. latency coding, TTFS) και να συγκριθούν με τις προσεγγίσεις που ακολουθήθηκαν.

Loss Functions που βασίζονται στον χρόνο: Μια άλλη πιθανή προσέγγιση θα ήταν η χρήση συναρτήσεων απώλειας που λαμβάνουν υπόψη τους τη χρονική στιγμή του πρώτου spike (time-to-first-spike), αντί για το total number των spikes, και επομένως επιτρέπουν στο δίκτυο να ανταποκρίνεται πιο γρήγορα.

Μεγαλύτερα και πιο Πολύπλοκα Δεδομένα: Μια άλλη δυνατότητα θα ήταν η επέκταση αυτής της έρευνας σε πιο προηγμένα σύνολα δεδομένων, όπως τα DVS-CIFAR10 ή το N-Caltech101, προκειμένου να επιβεβαιωθούν τα αποτελέσματα.

Καλύτερες τεχνικές regularization για την αραιότητα: Μια άλλη ιδέα θα ήταν η χρήση συγκεκριμένων όρων κόστους που προωθούν την αραιότητα (π.χ. L1 regularization στα spike counts) και επομένως να μπορεί να επιτευχθεί ακόμα καλύτερη απόδοση χωρίς να υποφέρει η απόδοση στο σύνολο.

Εφαρμογή στο Neuromorphic Hardware: Τέλος, αν ένα από τα μοντέλα που παρουσιάστηκαν σε αυτή τη μελέτη μεταφερόταν σε neuromorphic υλικό, θα ήταν δυνατή η επιβεβαίωση των ενεργειακών πλεονεκτημάτων του, ειδικά σε εφαρμογές edge-computing.

Συνολικά, τα ευρήματα της παρούσας διατριβής δείχνουν ότι οι Spiking Neural Networks είναι μια τεχνολογία μεγάλης προοπτικής, η οποία συνδυάζει υψηλή απόδοση και αποδοτική ενέργεια, και μπορεί να βρει εφαρμογή σε πολλούς τομείς της τεχνητής νοημοσύνης στο μέλλον.

Βιβλιογραφία

1. Tavanaei, Amirhossein, et al. "Deep Learning in Spiking Neural Networks." *Neural Networks*, vol. 111, Mar. 2019, pp. 47–63, <https://doi.org/10.1016/j.neunet.2018.12.002>. Accessed 20 Jan. 2025.
2. Eshraghian, Jason K., et al. "Training Spiking Neural Networks Using Lessons from Deep Learning." *Proceedings of the IEEE*, vol. 111, no. 9, Sept. 2023, pp. 1016–1054, <https://doi.org/10.1109/jproc.2023.3308088>. Accessed 20 Jan. 2025.
3. Nefci, Emre, et al. *LEARNING ALGORITHMS and SIGNAL PROCESSING for BRAIN-INSPIRED COMPUTING IEEE SIGNAL PROCESSING MAGAZINE* |. 2019, <https://doi.org/10.1109/MSP.2019.2931595>. Accessed 20 Jan. 2025.
4. [Jason K. Eshraghian, Max Ward, Emre Nefci, Xinxin Wang, Gregor Lenz, Girish Dwivedi, Mohammed Bennamoun, Doo Seok Jeong, and Wei D. Lu "Training Spiking Neural Networks Using Lessons From Deep Learning". *Proceedings of the IEEE*, 111\(9\) September 2023.](#)
5. *SnnTorch Documentations* - <https://snntorch.readthedocs.io/en/latest/>
6. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press. <https://www.deeplearningbook.org/>
7. Haykin, S. (2010). *Νευρωνικά δίκτυα και μηχανική μάθηση (3η έκδ.)*. Εκδόσεις Παπασωτηρίου.
8. Maass, W. (1997). *Networks of spiking neurons: The third generation of neural network models*. *Neural Networks*, 10(9), 1659–1671. <https://www.sciencedirect.com/science/article/abs/pii/S0893608097000117?via%3Dihub>

9. Eshraghian, J. K., Ward, M., Neftci, E. O., & Orchard, G. (2023). Training spiking neural networks using lessons from deep learning. *Proceedings of the IEEE*, 111(9), 1069–1102. <https://arxiv.org/abs/2109.12894>
10. Moustakas, G., Tsilikas, I., Bogris, A., & Mesaritakis, C. (2024). Neuromorphic imaging flow cytometry combined with adaptive recurrent spiking neural networks. *Journal of Biophotonics*, 27(1). <https://arxiv.org/abs/2505.11433>
11. Orchard, G., Cohen, G., Jayawant, A., & Thakor, N. V. (2015). Converting static image datasets to spiking neuromorphic datasets using saccades. *Frontiers in Neuroscience*, 9, 437. <https://doi.org/10.3389/fnins.2015.00437>
12. Iyer, L. R., Chua, Y., & Li, H. (2021). Is neuromorphic MNIST neuromorphic? Analyzing the discriminative power of neuromorphic datasets in the time domain. *Frontiers in Neuroscience*, 15, 608567. <https://doi.org/10.3389/fnins.2021.608567>
13. Shrestha, S. B., Zhu, L., & Sun, P. (2021). SpikeMax: Spike-based loss methods for classification. *arXiv preprint, arXiv:2103.01458*. <https://arxiv.org/abs/2205.09845>
14. Wu, J., Chua, Y., Zhang, M., Yang, Q., Li, G., & Li, H. (2021). Deep spiking neural network with spike count based learning rule. *Frontiers in Neuroscience*, 15, 638564. <https://arxiv.org/abs/1902.05705>
15. Fang, W., Yu, Z., Chen, Y., Masquelier, T., Huang, T., & Tian, Y. (2021). Incorporating learnable membrane time constant to enhance learning of spiking neural networks. *arXiv preprint, arXiv:2007.05785*. <https://doi.org/10.48550/arXiv.2007.05785>
16. Chen, H., Yu, L., Zhan, S., Yao, P., & Shao, J. (2024). Time-independent spiking neuron via membrane potential estimation for efficient spiking neural networks. *arXiv preprint, arXiv:2409.04978v2*. <https://doi.org/10.48550/arXiv.2409.04978>
17. Ma, C., Chen, X., Li, Y., Yang, Q., Wu, Y., Li, G., Pan, G., Tang, H., Tan, K. C., & Wu, J. (2022). Spiking neural networks for temporal processing: Status quo and future prospects. *IEEE Transactions on Neural Networks and Learning Systems*. <https://arxiv.org/abs/2502.09449>

18. *Wikipedia. (n.d.). History of artificial neural networks. Retrieved from*
https://en.wikipedia.org/wiki/History_of_artificial_neural_networks

19. *Emin Orhan. (n.d.). The Leaky Integrate-and-Fire Neuron Model. Retrieved from*
<https://eminorhan.github.io/>