

Πανεπιστήμιο Δυτικής Μακεδονίας
Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών
Υπολογιστών

Αλγόριθμοι αρχικοποίησης βαρών
νευρωνικών δικτύων

Νικόλαος-Παναγιώτης Μακρής (ΑΜ: 1502)

Επιβλέπων Καθηγητής: Νικόλαος Πλόσκας

Εργαστήριο Ευφρών Συστημάτων & Βελτιστοποίησης

21 Ιουνίου 2023

Περίληψη

Την τελευταία δεκαετία, η τεχνητή νοημοσύνη και γενικότερα οι τεχνικές της μηχανικής μάθησης διατηρούν μία ανοδική πορεία και χρησιμοποιούνται κατά κόρον, καθώς αξιοποιούνται με πολλούς τρόπους και δίνουν λύσεις σε αρκετά προβλήματα. Μία πτυχή της μηχανικής μάθησης είναι τα Νευρωνικά Δίκτυα, ένα πολύπλοκο μοντέλο εμπνευσμένο από τους νευρώνες του εγκεφάλου, που επεκτείνεται σε ένα μεγάλο φάσμα προβλημάτων κατηγοριοποίησης και βελτιστοποίησης.

Η εργασία αυτή στοχεύει στο να αναδείξει τα οφέλη αλλά και τα περιθώρια βελτίωσης, που παρέχει ή δέχεται αντίστοιχα ένα Συνελικτικό Νευρωνικό Δίκτυο κατά την εκπαίδευση του, το οποίο είναι ένα εκ των διαφόρων Νευρωνικών Δικτύων που υπάρχουν. Τα Συνελικτικά Δίκτυα έχουν ποικίλες εφαρμογές στην ανάλυση, κατηγοριοποίηση και επεξεργασία εικόνας, και γι' αυτό αποτελούν ένα ιδιαίτερα ενδιαφέρον και επίκαιρο αντικείμενο μελέτης. Η λειτουργία τους, όσον αφορά την κατηγοριοποίηση εικόνων, επικεντρώνεται στη συνέλιξη, μία διαδικασία η οποία επιδιώκει να εξάγει μοναδικά χαρακτηριστικά και μοτίβα από εικόνες που αντικατοπτρίζουν το ίδιο αντικείμενο, και στη συνέχεια να τις κατηγοριοποιήσουν στο εκάστοτε αντικείμενο.

Στην παρούσα εργασία, θα αξιοποιηθούν διάφορα μοντέλα Συνελικτικών Δικτύων ανάλογα με το σύνολο δεδομένων που στοχεύουν να κατηγοριοποιήσουν. Τα μοντέλα είναι τρία: LeNet, ModNet1 και ModNet2. Όσον αφορά τα σύνολα δεδομένων, αρχικά θα εξεταστεί το MNIST και το Fashion MNIST, δύο σχετικά απλά σύνολα δεδομένων, που εμπεριέχουν χειρόγραφους αριθμούς και ρούχα αντίστοιχα, αλλά αμφότερες αποτελούν ασπρόμαυρες εικόνες. Μετέπειτα αξιοποιούνται τα Cifar10 και Cifar100, έγχρωμες εικόνες που διαχωρίζονται σε 10 και 100 κλάσεις αντίστοιχα, όπου εδώ οι διαδικασίες εκπαίδευσης μεγαλώνουν.

Σε όλα τα μοντέλα που χρησιμοποιούνται για την κατηγοριοποίηση, εφαρμόζονται διάφοροι αλγόριθμοι αρχικοποίησης βαρών, όπου βάρη είναι οι πυρήνες συ-

νέλιξης, δηλαδή τα εργαλεία που επιτυγχάνουν τη διαδικασία εύρεσης χαρακτηριστικών σε εικόνες. Οι αλγόριθμοι αυτοί αποτελούν το κύριο αντικείμενο μελέτης, και σκοπός είναι να αναδειχθεί ποιοί φαντάζουν κατάλληλοι για το εκάστοτε ζήτημα κατηγοριοποίησης. Για τον σκοπό αυτό, διεξάγονται πειράματα και εξάγονται μετρικές που αξιολογούν την απόδοση του εκάστοτε αλγορίθμου.

Τα συμπεράσματα της έρευνας ποικίλουν. Τα εκπαιδευμένα δίκτυα, κυρίως πάνω σε σύνολα δεδομένων με ασπρόμαυρες εικόνες, εμφάνισαν καλά αποτελέσματα, κάτι το οποίο όμως είναι αναμενόμενο. Όσον αφορά τα υπόλοιπα μοντέλα εκπαίδευσης πάνω σε έγχρωμες εικόνες, η εκπαίδευση ήταν μία πιο απαιτητική διαδικασία που έδωσε ικανοποιητικά αποτελέσματα, αλλά τα μοντέλα σίγουρα μπορούν να βελτιωθούν παραπάνω. Οι τεχνικές αρχικοποίησης νευρωνικών δικτύων είναι πολλές και αξιοποιήθηκαν επαρκώς, όμως σίγουρα υπάρχει περιθώριο βελτίωσης. Γενικότερα, η αξιοποίηση Συνελικτικών Δικτύων εξετάζεται συνεχώς και αποτελεί αντικείμενο μελλοντικών επεκτάσεων για διάφορους ερευνητές. Και αυτό γιατί υπάρχουν πολλοί τρόποι παραμετροποίησης ενός δικτύου, για την προσέγγιση του εκάστοτε ζητήματος, κάτι ενθαρρυντικό για τις μελλοντικές μελέτες και την εξαγωγή ανταγωνιστικών αποτελεσμάτων.

Λέξεις κλειδιά: Μηχανική Μάθηση, Νευρωνικά Δίκτυα, Συνελικτικά Νευρωνικά Δίκτυα, Μάθηση με Επίβλεψη, Αρχικοποίηση Βαρών

Abstract

In the last decade, artificial intelligence and machine learning techniques in general, have been on the rise and are constantly being used, as they are being utilized in many different ways to solve problems. One aspect of machine learning is Neural Networks, a complex model inspired by the brain's neurons that extends to a wide range of classification and optimization problems.

This work aims to highlight the benefits but also the room for improvement, provided or received respectively by a Convolutional Neural Network during its training, which is one of the various Neural Networks that exist. Convolutional Networks have a variety of applications in image analysis, categorization and processing, which is why they are a particularly interesting subject of study. Their function, as far as image classification is concerned, is Convolution, a process that seeks to extract unique features and patterns from images that reflect the same object, and then categorize them into each object.

In this work, different Convolutional Networks models will be utilized depending on the dataset they aim to classify. The models are three: LeNet, ModNet1 and ModNet2. Concerning the datasets, first we will consider MNIST and Fashion MNIST, two relatively simple datasets, containing handwritten numbers and clothes respectively, but both are black and white images. Cifar10 and Cifar100 are then used, coloured images separated into 10 and 100 classes respectively, where training processes grow.

In all the models used for classification, various weight initialization algorithms are applied, where weights are the convolution kernels, i.e. the tools that achieve the process of finding features in images. These algorithms are the main object of study, and the purpose is to show which ones seem suitable for the classification issue at hand. For this purpose, experiments are conducted and metrics are extracted that evaluate the performance of each algorithm.

Research conclusions vary. The trained networks, especially on black and white

datasets, performed well, which is to be expected since the training model was the most optimal. As for the rest of the models trained on coloured images, training was a more demanding process that delivered satisfactory results, but the models can certainly be further optimized. Neural network initialization techniques are many and have been sufficiently exploited, but there is certainly room for improvement. In general, the exploitation of Convolutional Networks is constantly being considered and is the subject of future extensions for various researchers. And this is because there are many ways to parameterize a network, to approach each issue, something encouraging for future studies and the extraction of competitive results.

Keywords: Machine Learning, Neural Networks, Convolutional Neural Networks, Supervised Learning, Weight Initialization

Δήλωση Πνευματικών Δικαιωμάτων

Δήλωση Πνευματικών Δικαιωμάτων Δηλώνω ρητά ότι, σύμφωνα με το άρθρο 8 του Ν. 1599/1986 και τα άρθρα 2,4,6 παρ. 3 του Ν. 1256/1982, η παρούσα Διπλωματική Εργασία με τίτλο "Αλγόριθμοι αρχικοποίησης βαρών νευρωνικών δικτύων" καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας και αναφέρονται ρητώς μέσα στο κείμενο που συνοδεύουν, και η οποία έχει εκπονηθεί στο Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών του Πανεπιστημίου Δυτικής Μακεδονίας, υπό την επίβλεψη του μέλους του Τμήματος κ. Νικόλαου Πλόσκα αποτελεί αποκλειστικά προϊόν προσωπικής εργασίας και δεν προσβάλλει κάθε μορφής πνευματικά δικαιώματα τρίτων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή / και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα. Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και μόνο.

Copyright (C) Νικόλαος-Παναγιώτης Μακρής & Νικόλαος Πλόσκας, 2023, Κοζάνη

Περιεχόμενα

1	Εισαγωγή	13
1.1	Ορισμός του προβλήματος	13
1.2	Κίνητρα και στόχοι υλοποίησης	14
1.3	Διάρθρωση κειμένου	14
2	Θεωρητικό Υπόβαθρο	16
2.1	Μηχανική Μάθηση	16
2.1.1	Μάθηση με επίβλεψη	16
2.1.2	Κατηγοριοποίηση	17
2.1.3	Παλινδρόμηση	20
2.1.4	Μάθηση χωρίς επίβλεψη	22
2.1.5	Προβλήματα στην εκπαίδευση	25
2.2	Τεχνητά Νευρωνικά Δίκτυα	26
2.2.1	Νευρώνας Perceptron	26
2.2.2	Πολυστρωματικό Perceptron	27
2.2.3	Νευρωνικά Δίκτυα Προσωτροφοδότησης	27
2.2.4	Συνάρτηση ενεργοποίησης	28
2.2.5	Συνάρτηση απώλειας	28
2.2.6	Συνελικτικά Νευρωνικά Δίκτυα	30
2.2.7	Νευρωνικά Δίκτυα Ανάδρασης	30
2.3	Νευρωνικά Δίκτυα και Βαθιά Μάθηση	31
2.3.1	Βαθιά Μάθηση ως εξέλιξη της Μηχανικής Μάθησης	31
2.3.2	Βαθιά Μάθηση και Συνελικτικά Δίκτυα	32
2.3.3	Συνελικτικά Δίκτυα LeNet και AlexNet	33
2.3.4	Βαθιά Μάθηση και Δίκτυα Ανάδρασης	35
2.3.5	Autoencoder-Generative Adversarial-Self Organizing Map	36

2.4	Προεπεξεργασία δεδομένων και αρχικοποίηση	39
2.4.1	Προεπεξεργασία δεδομένων	39
2.4.2	Βελτιστοποίηση υπερπαραμέτρων	39
3	Αλγόριθμοι Αρχικοποίησης των Βαρών σε Νευρωνικά Δίκτυα	43
3.1	Περιγραφή της υποδομής του συστήματος	43
3.1.1	Συνέλιξη	44
3.1.2	Συσώρευση	51
3.1.3	Πλήρως συνδεδεμένα στρώματα	52
3.1.4	Μεταγλώττιση μοντέλου	53
3.2	Αξιολόγηση συστήματος-μετρικές	57
3.2.1	Ακρίβεια	57
3.2.2	Σφάλμα	58
3.2.3	Precision	58
3.2.4	Ανάκληση	58
3.2.5	F1-Score	59
4	Υπολογιστική μελέτη	60
4.1	Μοντέλα εκπαίδευσης	60
4.1.1	LeNet	60
4.1.2	ModNet	61
4.2	Προγραμματιστικό περιβάλλον ανάπτυξης	63
4.3	Εκπαίδευση στο LeNet	64
4.3.1	Σύνολο MNIST	64
4.3.2	Σύνολο Fashion MNIST	66
4.3.3	Σύνολο Cifar10	67
4.3.4	Σύνολο Cifar100	68
4.4	Εκπαίδευση στο ModNet1	69
4.4.1	Σύνολο MNIST	69
4.4.2	Σύνολο Fashion MNIST	71
4.4.3	Σύνολο Cifar10	72
4.4.4	Σύνολο Cifar100	74
4.5	Εκπαίδευση στο ModNet2	76

4.5.1	Σύνολο MNIST	76
4.5.2	Σύνολο Fashion MNIST	77
4.5.3	Σύνολο Cifar10	79
4.5.4	Σύνολο Cifar100	80
4.6	Βελτιστοποίηση υπερπαραμέτρων	81
4.6.1	Βελτιστοποίηση υπερπαραμέτρων για το LeNet	83
4.6.2	Βελτιστοποίηση υπερπαραμέτρων για το ModNet1	84
4.6.3	Βελτιστοποίηση υπερπαραμέτρων για το ModNet2	84
4.7	Πορίσματα υπολογιστικής μελέτης	86
5	Συμπεράσματα	89

Κατάλογος σχημάτων

2.1	Παράδειγμα αναπαράστασης ενός δέντρου απόφασης.	18
2.2	Παράδειγμα κατηγοριοποίησης με τη μέθοδο διανυσμάτων υποστήριξης.	19
2.3	Γραμμική Παλινδρόμηση	21
2.4	K-Means Clustering με την απόσταση των σημείων από το κέντρο.	23
2.5	Ιεραρχική ομαδοποίηση «από κάτω προς τα πάνω».	23
2.6	Εφαρμογή Apriori σε ένα δέντρο και ψαλίδισμα ενός Infrequent Superset.	24
2.7	Υποπροσαρμογή και υπερπροσαρμογή μοντέλου σε σύγκριση με ένα βέλτιστο μοντέλο δυαδικής κατηγοριοποίησης.	25
2.8	Τεχνητό Νευρωνικό Δίκτυο Προσωτροφοδότησης.	28
2.9	Απλό Τεχνητό Νευρωνικό Δίκτυο Ανάδρασης με κοινά επισυναπτόμενα βάρη.	31
2.10	Αναπαράσταση της αποδοτικότητας της Βαθιάς Μάθησης σε σύγκριση με τη Μηχανική Μάθηση.	32
2.11	Αναπαράσταση Βαθιάς Μάθησης μέσω CNN [23]	33
2.12	Αρχιτεκτονική του LeNet.	34
2.13	Αρχιτεκτονική του AlexNet.	34
2.14	RNN για αλληλουχία λέξεων.	35
2.15	Λειτουργία ενός Generative Adversarial Network.	37
2.16	Ορθή σειρά και οργάνωση ενεργειών για εκπαίδευση ενός μοντέλου.	40
2.17	Βελτιστοποίηση παραμέτρων για επίτευξη στόχων.	41
3.1	Διαδικασία εξαγωγής χαρακτηριστικών	44
3.2	Κανονική και ομοιόμορφη κατανομή.	45
3.3	Εφαρμογή ενίσχυσης (P=1) για την εικόνα εισόδου στο Σχήμα 3.1	49
3.4	Γραφικές παραστάσεις του γραμμικού ανορθωτή και της υπερβολικής εφαπτομένης.	50

3.5	Παράδειγμα εφαρμογής Max Pooling σε ένα πλαίσιο 4x4.	51
3.6	Παράδειγμα απεικόνισης ενός πίνακα συσσώρευσης να ισοπεδώνεται σε ένα διάνυσμα και μεταγενέστερα σε ένα νευρωνικό δίκτυο προσω-τροφοδότησης.	52
3.7	Εύρεση ελαχίστου με κατάβαση δυναμικού	55
4.1	LeNet-MNIST: Ακρίβεια και Απώλεια, Εκτέλεση 5η.	65
4.2	LeNet-Fashion MNIST: Ακρίβεια και Απώλεια, Εκτέλεση 1η	66
4.3	LeNet-Cifar10: Ακρίβεια και Απώλεια, Εκτέλεση 3η	68
4.4	LeNet-Cifar100: Ακρίβεια και Απώλεια, Εκτέλεση 3η	71
4.5	ModNet1-MNIST: Ακρίβεια και Απώλεια, Εκτέλεση 3η	72
4.6	ModNet1-FMNIST: Ακρίβεια και Απώλεια, Εκτέλεση 4η	73
4.7	ModNet1-Cifar-10: Ακρίβεια και Απώλεια, Εκτέλεση 2η	74
4.8	ModNet1-Cifar100: Ακρίβεια και Απώλεια, Εκτέλεση 3η	75
4.9	ModNet2-MNIST: Ακρίβεια και Απώλεια, Εκτέλεση 4η	77
4.10	ModNet2-FMNIST: Ακρίβεια και Απώλεια, Εκτέλεση 2η	79
4.11	ModNet2-Cifar10: Ακρίβεια και Απώλεια, Εκτέλεση 4η	80
4.12	ModNet2-Cifar100: Ακρίβεια και Απώλεια, Εκτέλεση 5η	82
4.13	Βελτιστοποιημένο LeNet-Cifar100: Ακρίβεια και Απώλεια, Εκτέλεση 5η	83
4.14	Βελτιστοποιημένο ModNet1-Cifar100: Ακρίβεια και Απώλεια, Εκτέ- λεση 5η	84
4.15	Βελτιστοποιημένο ModNet2-Cifar100: Ακρίβεια και Απώλεια, Εκτέ- λεση 4η	85

Κατάλογος πινάκων

4.1	Μετρικές εκπαίδευσης για το συνδυασμό LeNet-MNIST.	64
4.2	Μετρικές δοκιμής για το συνδυασμό LeNet-MNIST.	65
4.3	Μετρικές εκπαίδευσης για το συνδυασμό LeNet-Fashion MNIST.	66
4.4	Μετρικές δοκιμής για το συνδυασμό LeNet-Fashion MNIST.	67
4.5	Μετρικές εκπαίδευσης για το συνδυασμό LeNet-Cifar10.	67
4.6	Μετρικές δοκιμής για το συνδυασμό LeNet-Cifar10.	68
4.7	Μετρικές εκπαίδευσης για το συνδυασμό LeNet-Cifar100.	69
4.8	Μετρικές δοκιμής για το συνδυασμό LeNet-Cifar100.	69
4.9	Μετρικές εκπαίδευσης για το συνδυασμό ModNet1-MNIST.	70
4.10	Μετρικές δοκιμής για το συνδυασμό ModNet1-MNIST.	70
4.11	Μετρικές εκπαίδευσης για το συνδυασμό ModNet1-Fashion MNIST.	71
4.12	Μετρικές δοκιμής για το συνδυασμό ModNet1-Fashion MNIST.	72
4.13	Μετρικές εκπαίδευσης για το συνδυασμό ModNet1-Cifar10.	73
4.14	Μετρικές δοκιμής για το συνδυασμό ModNet1-Cifar10.	74
4.15	Μετρικές εκπαίδευσης για το συνδυασμό ModNet1-Cifar100.	75
4.16	Μετρικές δοκιμής για το συνδυασμό ModNet1-Cifar100.	76
4.17	Μετρικές εκπαίδευσης για το συνδυασμό ModNet2-MNIST.	76
4.18	Μετρικές δοκιμής για το συνδυασμό ModNet2-MNIST.	77
4.19	Μετρικές εκπαίδευσης για το συνδυασμό ModNet2-Fashion MNIST.	78
4.20	Μετρικές δοκιμής για το συνδυασμό ModNet2-Fashion MNIST.	78
4.21	Μετρικές εκπαίδευσης για το συνδυασμό ModNet2-Cifar10.	79
4.22	Μετρικές δοκιμής για το συνδυασμό ModNet2-Cifar10.	80
4.23	Μετρικές εκπαίδευσης για το συνδυασμό ModNet2-Cifar100.	81
4.24	Μετρικές δοκιμής για το συνδυασμό ModNet2-Cifar100.	81
4.25	Μετρικές εκπαίδευσης και δοκιμής για το συνδυασμό LeNet-Cifar100 με τον αρχικοποιητή GloU, μετά από βελτιστοποίηση υπερπαραμέτρων.	83

4.26	Μετρικές εκπαίδευσης και δοκιμής για το συνδυασμό ModNet1-Cifar100 με τον αρχικοποιητή LeU, μετά απο βελτιστοποίηση υπερπαραμέτρων.	85
4.27	Μετρικές εκπαίδευσης και δοκιμής για το συνδυασμό ModNet2-Cifar100 με τον αρχικοποιητή LeN, μετά απο βελτιστοποίηση υπερπαραμέτρων.	86
4.28	Υψηλότερες μετρικές δοκιμής (Accuracy και F1-Score) για το μοντέλο LeNet.	87
4.29	Υψηλότερες μετρικές δοκιμής (Accuracy και F1-Score) για το μοντέλο ModNet1.	87
4.30	Υψηλότερες μετρικές δοκιμής (Accuracy και F1-Score) για το μοντέλο ModNet2.	88

Κεφάλαιο 1

Εισαγωγή

1.1 Ορισμός του προβλήματος

Το πεδίο των Νευρωνικών Δικτύων καλύπτει ένα μεγάλο εύρος προβλημάτων που επιλύονται μέσω μηχανικής μάθησης, καθώς προσαρμόζεται στα δεδομένα και τα ζητήματα του εκάστοτε προβλήματος. Τα τελευταία χρόνια ωστόσο, η Βαθιά Μάθηση έρχεται να ενδυναμώσει τη μηχανική μάθηση και να προσελκύσει ακόμα παραπάνω το ενδιαφέρον των ερευνητών.

Η Βαθιά Μάθηση όταν επιτυγχάνεται με μη εποπτευόμενο τρόπο, δηλαδή όταν το μοντέλο εξάγει από μόνο του συμπεράσματα και όχι με την παρατήρηση, σημειώνει σημαντικά αποτελέσματα. Στον χώρο των Νευρωνικών Δικτύων υπάρχουν διάφορα μοντέλα που εκπαιδεύονται χωρίς επίβλεψη, όπως οι Αυτόματοι Κωδικοποιητές ή τα Παραγωγικά Αντιπαλικά Δίκτυα, μοντέλα που συνεισφέρουν σημαντικά στον τομέα της επεξεργασίας εικόνας.

Τα Συνελικτικά Δίκτυα από την άλλη, αξιοποιούν τη μάθηση με επίβλεψη, από ήδη γνωστά δεδομένα, με σκοπό να κατηγοριοποιήσουν και όχι να επεξεργαστούν εικόνες. Αποτελούν ένα από τα πιο επίκαιρα θέματα έρευνας, καθώς υπάρχουν πολλά σύνολα δεδομένων εικόνας, και συνεπώς πολλά μοντέλα εκπαίδευσης πάνω στο κάθε σύνολο, ή και σε διάφορα σύνολα.

Παρόλαυτα, η Βαθιά Μάθηση στα Συνελικτικά Δίκτυα για να επιφέρει θετικά αποτελέσματα προαπαιτεί ένα βέλτιστο δίκτυο, γεγονός που προβληματίζει τους ερευνητές καθώς ένα δίκτυο επιδέχεται σε πολλούς παραμέτρους του βελτιστοποίησης.

1.2 Κίνητρα και στόχοι υλοποίησης

Η υλοποίηση αυτή λοιπόν, στοχεύει στη βελτιστοποίηση Συνελικτικών Δικτύων για την κατηγοριοποίηση εικόνων, δίνοντας ιδιαίτερη βάση στους αρχικοποιητές βαρών και ποιοί καθίστανται πιο κατάλληλοι ανα πρόβλημα. Σημαντικό κίνητρο αποτέλεσε η έρευνα των Στεργιόπουλου κ.α. [25] πάνω στην αρχικοποίηση βαρών, και στην οποία βασίζεται η φιλοσοφία της συγκεκριμένης υλοποίησης. Οι Αλγόριθμοι Αρχικοποίησης Βαρών στα Συνελικτικά Δίκτυα, εισηγήθηκαν με πρωτοβουλία του διάσημου ερευνητή Yann LeCun [15], ο οποίος δημιούργησε το πρώτο επίσημο Συνελικτικό Νευρωνικό Δίκτυο, ενώ με τη σειρά τους ακολούθησαν και άλλοι γνωστοί ερευνητές με τις παραλλαγές και τις ιδέες τους.

Στόχος είναι επομένως, να δοκιμαστούν οι αλγόριθμοι αυτοί πάνω σε ορισμένα δίκτυα και ορισμένα σύνολα δεδομένων και να αποφανθεί αν ενισχύουν την απόδοση του δικτύου, και αν ναι ποιός ή ποιοί φαντάζουν καταλληλότεροι. Έπειτα από μία σειρά εκτενών πειραμάτων και πολλών εκτελέσεων, θα εξαχθεί πόρισμα σχετικά με το ποιοί αλγόριθμοι αποδίδουν κατά μέσο όρο, ενώ θα αναδειχθούν και γενικότερα συμπεράσματα από την παρατήρηση των μετρικών που υπολογίστηκαν, αλλά και της συμπεριφοράς του εκάστοτε δικτύου.

1.3 Διάρθρωση κειμένου

Μετά από αυτό το κεφάλαιο, ακολουθούν πέντε κεφάλαια τα οποία καλύπτουν όλες τις απαραίτητες πτυχές προς ανάλυση.

Στο Κεφάλαιο 2 δίνεται το υπόβαθρο της μελέτης. Αρχικά αναλύονται οι βασικές τεχνικές και έννοιες που συσχετίζονται με τη Μηχανική Μάθηση, ενώ μετέπειτα γίνεται αναφορά σε διάφορα είδη Νευρωνικών Δικτύων (πως λειτουργούν και πως εκπαιδεύονται) ενώ εκτενής αναφορά γίνεται σε κάποια από αυτά. Τέλος, γίνεται εισαγωγή στην προεπεξεργασία δεδομένων και την αρχικοποίηση στη Μηχανική Μάθηση.

Στο Κεφάλαιο 3 αναλύεται το πειραματικό υπόβαθρο. Τα Συνελικτικά Δίκτυα και οι πτυχές τους σημειώνονται μία προς μία, για την καλύτερη κατανόηση του πειράματος, ενώ αναφέρονται και οι αλγόριθμοι αρχικοποίησης βαρών και ποιές είναι οι διαφορές τους.

Στο Κεφάλαιο 4 σχολιάζονται τα αποτελέσματα των πειραμάτων. Το κάθε υποκεφάλαιο αναπαριστά την εκάστοτε μελέτη που έγινε ανα σύνολο δεδομένων και επιμέρους μοντέλο. Για κάθε συνδυασμό μοντέλου/συνόλου δεδομένων, παρατίθενται πίνακες με αποτελέσματα από τις μετρικές της εκτέλεσης για κάθε αλγόριθμο αρχικοποίησης, ενώ παράλληλα συμπεριλαμβάνονται και γραφήματα που αναδεικνύουν την ακρίβεια και το σφάλμα στην κατηγοριοποίηση, ανα αλγόριθμο.

Στο Κεφάλαιο 5 γίνεται ανασκόπηση των πειραμάτων, και σχολιασμός των αποτελεσμάτων. Σκοπός είναι να εξαχθούν συμπεράσματα σχετικά με το που υστερεί και που υπερτερεί το κάθε μοντέλο, ανάλογα με το πόρισμα της μελέτης.

Κεφάλαιο 2

Θεωρητικό Υπόβαθρο

2.1 Μηχανική Μάθηση

Η τεχνητή νοημοσύνη προσδοκεί στην εκπαίδευση μιας μηχανής, έτσι ώστε να αναγνωρίζει πρότυπα και να κατηγοριοποιεί δεδομένα γρήγορα και αποδοτικά, όπως μπορεί και ο άνθρωπος με μία απλή παρατήρηση. Αυτό επιτυγχάνεται με τις μεθόδους Μάθηση με Επίβλεψη (Supervised Learning) και Μάθηση χωρίς Επίβλεψη (Unsupervised Learning). Στην πρώτη περίπτωση επενεργεί η λεγόμενη εποπτευόμενη μάθηση, στην οποία χρησιμοποιείται ένα μοντέλο ή πιο γενικά ένας αλγόριθμος που διαχειρίζεται κατηγοριοποιημένα δεδομένα και εξάγει προβλέψεις βασιζόμενο στα δεδομένα και τα χαρακτηριστικά αυτών. Αν τα δεδομένα δε διαθέτουν ευδιάκριτα χαρακτηριστικά, τότε το μοντέλο επιτελεί έργο τόσο "δασκάλου" όσο και "μαθητή" καθώς εκπαιδεύεται στο να εξάγει προβλέψεις από μόνο του.

2.1.1 Μάθηση με επίβλεψη

Ένας μαθητής, στην προκειμένη ένα πρόγραμμα υπολογιστή, επιτελεί τη λεγόμενη εποπτευόμενη μάθηση, αξιοποιώντας δύο διαφορετικά σετ δεδομένων, το σετ εκπαίδευσης (training set) και το σετ δοκιμής (testing set). Σκοπός είναι η μηχανή να εκπαιδευτεί πάνω σε ένα μεγάλο σύνολο εκπαίδευσης, το οποίο εμπεριέχει ήδη κατηγοριοποιημένα δεδομένα, και στη συνέχεια να κατηγοριοποιήσει τα δείγματα ενός μικρότερου δοκιμαστικού σετ, στα πρότυπα που έχει εκπαιδευτεί [12].

Έστω, για παράδειγμα, ότι το σύνολο δεδομένων είναι η κατανάλωση ενέργειας σε κιλοβατώρες κατά τη διάρκεια της ημέρας, και σε συνολικό διάστημα ενός ολόκληρου χρόνου. Αν κάποιος θέλει να εξάγει προβλέψεις σχετικά με το ποιά θα είναι

η κατανάλωση ενέργειας στο προσεχές χρονικό διάστημα, πρέπει να διαχωρίσει κατάλληλα το σετ αυτό. Μία καλή ιδέα θα ήταν να συμπεριληφθούν στο σύνολο εκπαίδευσης δεδομένα για τους μήνες του καλοκαιριού, για να εξαχθούν μοτίβα όταν υπάρχει υψηλή κατανάλωση. Στη συνέχεια επιλέγεται ένα σετ δοκιμής, στο οποίο εφαρμόζεται ο εκπαιδευμένος αλγόριθμος και παρατηρείται αν συμβαδίζει με τα πρότυπα πάνω στα οποία εκπαιδεύτηκε (στην προκειμένη αν αναγνωρίζει τότε θα υπάρχει υψηλή κατανάλωση ενέργειας).

Γενικότερα, συνηθίζεται το σύνολο εκπαίδευσης να είναι μεγαλύτερο από το σύνολο δοκιμής κατά τον διαχωρισμό, έτσι ώστε να εξαχθούν συμπεράσματα σχετικά με διάφορα μοτίβα και συμπεριφορές τις οποίες ακολουθούν τα δεδομένα. Είναι σημαντικό όμως να περιοριστούν οι παράμετροι εκπαίδευσης και το εκπαιδευμένο μοντέλο να καθίσταται σχετικά απλό, γιατί αλλιώς θα υπάρχει υπερπροσαρμογή πάνω σε πολύ ειδικά ζητήματα.

2.1.2 Κατηγοριοποίηση

Στην παραδοσιακή μηχανική μάθηση υπάρχουν διάφορες τεχνικές κατηγοριοποίησης, τεχνικές δηλαδή στις οποίες το μοντέλο μάθησης μπορεί και αναθέτει το δείγμα μας σε μία κλάση/κατηγορία, δηλαδή το ταξινομεί. Η κατηγοριοποίηση μπορεί να είναι δύο ειδών: Δυαδική Κατηγοριοποίηση, δηλαδή το δείγμα κατηγοριοποιείται σε μία εκ των δύο κλάσεων, ή Κατηγοριοποίηση Πολλαπλών Τάξεων (Πολυκατηγορική Ταξινόμηση) όπου υπάρχουν πάνω από δύο κλάσεις διαθέσιμες.

Ένα τεστ εγκυμοσύνης για παράδειγμα, ταξινομείται δυαδικά, είτε είναι θετικό είτε αρνητικό, ενώ αντίθετα μία εικόνα μπορεί να απεικονίζει ένα από εκατοντάδες διαφορετικά ζώα. Είναι σημαντικό να αξιοποιούνται μοντέλα που κατηγοριοποιούν δεδομένα σε δύο ή και παραπάνω κλάσεις, ώστε να υπάρχει ποικιλία στη χρήση τους και να εκπαιδεύονται σε διαφορετικά πρότυπα.

Υπάρχουν πολλές τεχνικές κατηγοριοποίησης, οι οποίες ποικίλουν ανάλογα με τις ανάγκες και τον απώτερο σκοπό του προβλήματος, με κάποιες από τις πιο γνωστές τεχνικές, δυαδικής κατηγοριοποίησης ή και πολλαπλών τάξεων, να είναι οι εξής:

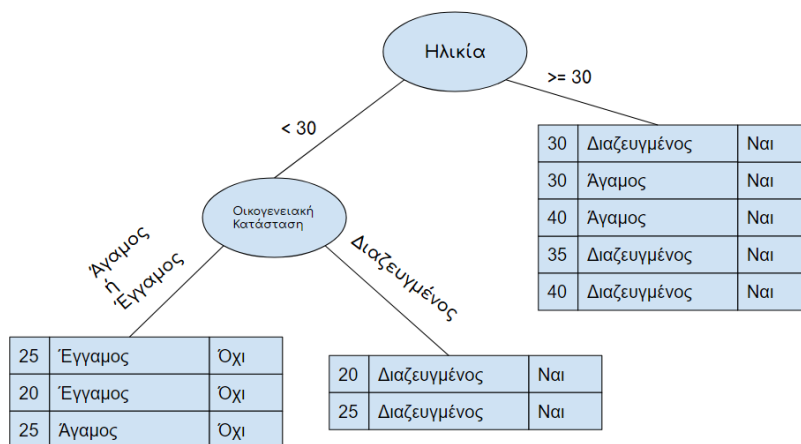
Naive-Bayes

Ο κατηγοριοποιητής Naive-Bayes είναι ένα πρότυπο κατά το οποίο υπολογίζονται οι πιθανότητες των χαρακτηριστικών που συνοδεύουν ένα δείγμα, και συνεπώς υπολογίζεται η συνολική πιθανότητα το συγκεκριμένο δείγμα να ανήκει σε μία κλάση.

Είναι μία ιδιαίτερα χρήσιμη τεχνική για ανάλυση κειμένου και κατηγοριοποίηση λέξεων, καθώς εντοπίζει μοτίβα στον γραπτό λόγο και εξάγει πιθανότητες σχετικά με το ποιά είναι η θεματολογία του λόγου αυτού. μία δημοφιλής εφαρμογή του Naive-Bayes είναι το Spam Detection, δηλαδή η κατηγοριοποίηση των αυτοματοποιημένων εισερχομένων e-mail, σχετικά με το αν είναι ανεπιθύμητα ή χρήσιμα, ανάλογα με το περιεχόμενό τους.

Δέντρο απόφασης

Το δέντρο απόφασης (Decision Tree) κατηγοριοποιεί ένα δείγμα σε ένα εκ των λήγοντων "φύλλων" του, αφού πρώτα διαχωρίσει τα δεδομένα του συνόλου δεδομένων βάση κάποιων κριτηρίων. Γενικότερα ακολουθεί ένα μαθηματικό κανόνα και αξιοποιεί την εντροπία, για να επιλέξει τη σωστή ακμή και να δημιουργήσει πρότυπα.



Σχήμα 2.1: Παράδειγμα αναπαράστασης ενός δέντρου απόφασης.

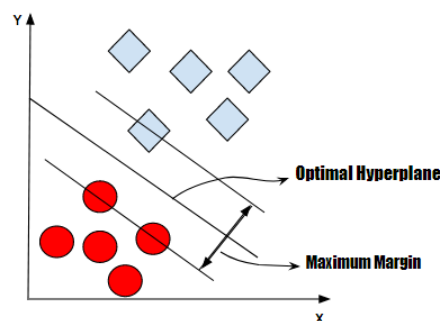
Στο Σχήμα 2.1 αναπαρίσταται ένα απλό μοντέλο δυαδικής κατηγοριοποίησης. Το πρόβλημα είναι αν κάποιος άνδρας θα ήταν αγοραστής ενός συγκεκριμένου προϊόντος, και όπως φαίνεται διαχωρίζει τους υποψήφιους αγοραστές πρώτα με την ηλικία και στη συνέχεια με την οικογενειακή τους κατάσταση.

Τυχαίο δάσος

Έστω ότι έχουμε ένα σύνολο δεδομένων, και τέσσερα τυχαία υποσύνολα του συνόλου αυτού. Η παραπάνω μέθοδος του δέντρου απόφασης (Ενότητα 2.1.2) μπορεί να γενικευτεί και να αξιοποιηθεί πολλαπλές φορές, αποτελώντας το λέγόμενο Τυχαίο Δάσος. Η μηχανή εκπαιδεύεται για κάθε υποσύνολο, δημιουργώντας έτσι τέσσερα ξεχωριστά δέντρα απόφασης τα οποία κατηγοριοποιούν στα φύλλα τους το δείγμα. Έτσι, αν επιλεγθεί ένα δείγμα και εφαρμοστεί στα τέσσερα αυτά δέντρα θα εξαχθεί από το καθένα μία πρόβλεψη για την κλάση στην οποία ανήκει, με την επικρατέστερη πρόβλεψη να υπερισχύει.

Μηχανές Διανυσμάτων Υποστήριξης (SVM)

Μία από τις πιο απλές και σύνθητες επιλογές για κατηγοριοποίηση δειγμάτων, είναι η μέθοδος μηχανών διανυσμάτων υποστήριξης. Η μέθοδος αυτή αξιοποιείται τόσο για δυαδική όσο και για πολυκλασική κατηγοριοποίηση, και βασίζεται στην εύρεση ενός διαχωριστικού επιπέδου (γνωστό και ως υπερεπίπεδο) το οποίο θα διαχωρίζει τα δείγματα.



Σχήμα 2.2: Παράδειγμα κατηγοριοποίησης με τη μέθοδο διανυσμάτων υποστήριξης.

Στο Σχήμα 2.2 αναδεικνύεται η χρήση των SVM για τον διαχωρισμό των μπλε από τα κόκκινα δείγματα και το αντίστροφο. Το βέλτιστο διαχωριστικό υπερεπίπεδο εντοπίζεται έχοντας θέση σαν σημεία αναφοράς τα διανύσματα υποστήριξης (στην προκειμένη το ένα μπλε και τα δύο κόκκινα δείγματα) και ορίζοντας πως η μεταξύ τους απόσταση απέχει το μέγιστο δυνατό περιθώριο, έτσι ώστε η γραμμή διαχωρισμού να αποτελεί και την ιδανική γραμμή που διαχωρίζει και κατηγοριοποιεί τα δεδομένα. Στις δύο διαστάσεις το διαχωριστικό επίπεδο είναι μία γραμμή, ενώ σε τρεις θα αποτελεί ένα διαχωριστικό υπερεπίπεδο δύο διαστάσεων.

Νευρωνικά Δίκτυα

Ιδιαίτερα δημοφιλές μοντέλο εποπτευόμενης μηχανικής μάθησης, διότι είναι ευέλικτο ως προς την υλοποίηση, την εκτέλεση, αλλά και την εκπαίδευση. Όπως θα αναδειχθεί στο Κεφάλαιο 2.2, τα νευρωνικά δίκτυα είναι ένα πολυσύνθετο μοντέλο που απαρτίζεται από πολλά κομμάτια και υπόκειται συγκεκριμένου είδους επεξεργασία. Αυτό επιτρέπει στον άνθρωπο να υπεραναλύσει τα επι μέρους ζητήματα ενός προβλήματος κατηγοριοποίησης, να επιλέξει τα ιδανικά κομμάτια και παραμέτρους για το δίκτυο του και επακόλουθα να το επιλύσει.

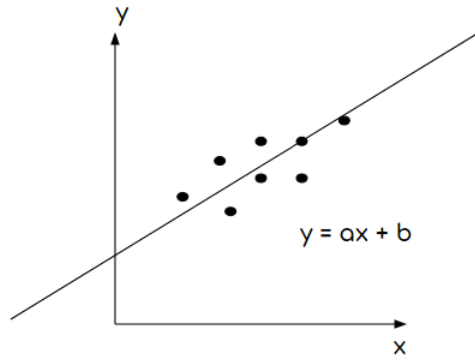
2.1.3 Παλινδρόμηση

Η διαφορά μεταξύ της κατηγοριοποίησης και της παλινδρόμησης (Regression) είναι πως κατά την κατηγοριοποίηση δίνονται στο δείγμα διακριτές τιμές κλάσεων, ενώ αντίθετα κατά την παλινδρόμηση γίνεται συσχέτιση στο συνεχές φάσμα πραγματικών τιμών. Πιο συγκεκριμένα, οι συνηθέστερες κατηγορίες είναι Απλή Γραμμική Παλινδρόμηση (Simple Linear Regression), Πολλαπλή Γραμμική Παλινδρόμηση (Multiple Linear Regression) και Λογιστική Παλινδρόμηση (Logistic Regression) ενώ παράλληλα και τα Νευρωνικά Δίκτυα μπορούν να χρησιμοποιηθούν για ζητήματα παλινδρόμησης.

Απλή Γραμμική Παλινδρόμηση

Οι διακριτές τιμές δεδομένων στο καρτεσιανό επίπεδο, είναι λογικό να παρουσιάζουν ομοιότητες ως προς την κατανομή. Στη μηχανική μάθηση λοιπόν, δημιουργήθηκε η ιδέα ότι θα μπορούσε να χαραχθεί μία γραμμή που θα αποτελεί σημείο αναφοράς, και γύρω από την οποία κατανέμονται τα δείγματα του συνόλου. Αυτό επιτυγχάνει η απλή γραμμική παλινδρόμηση, επιδιώκοντας να χαράξει την πιο κατάλληλη γραμμή για να περιγράψει την ομοιογένεια των δεδομένων πάνω στο επίπεδο. Η ευθεία που εκφράζει την κατανομή των δεδομένων αποτελεί την εξαρτημένη μεταβλητή, και το σύνολο δεδομένων αποτελεί την ανεξάρτητη μεταβλητή που υποκινεί την ευθεία.

Στο Σχήμα 2.3 αποτυπώνεται προσεγγιστικά μία ευθεία y , η οποία εκφράζει την κατανομή των ανεξάρτητων δεδομένων x . Σκοπός της γραμμικής παλινδρόμησης είναι τα δεδομένα να απέχουν όσο το δυνατόν λιγότερο από την ευθεία, κοινώς



Σχήμα 2.3: Γραμμική Παλινδρόμηση

να ελαχιστοποιείται το σφάλμα παλινδρόμησης, ούτως ώστε η ευθεία να εκφράζει βέλτιστα την κατανομή αυτή.

Πολλαπλή Γραμμική Παλινδρόμηση

Η πολλαπλή γραμμική παλινδρόμηση, λειτουργεί ακριβώς όπως η απλή, με τη διαφορά ότι η εξαρτημένη μεταβλητή που τίθεται προς πρόβλεψη δεν εξαρτάται από μία, αλλά από παραπάνω ανεξάρτητες μεταβλητές. Παραπάνω η μεταβλητή x θα μπορούσε να αποτελεί τον παράγοντα της ηλικίας ενός ατόμου, και η πρόβλεψη να είναι το εισόδημα του. Στην περίπτωση που δεν είχαμε σαν κριτήριο μονάχα την ηλικία αλλά και την εκπαίδευση ενός ατόμου, το εισόδημα του ατόμου θα ήταν $y = a_1 * x_1 + a_2 * x_2 + b$ αντί για $y = a * x + b$ η αλλιώς $y = a_1 * x_1 + a_2 * x_2 + \dots + a_n * x_n + b$ ανάλογα με το πόσες ανεξάρτητες μεταβλητές είθισται να οριστούν.

Λογιστική Παλινδρόμηση

Λογιστική παλινδρόμηση χρησιμοποιείται για να παραχθούν καμπύλες αντί για ευθείες, οι οποίες εκφράζουν τις πιθανότητες κατηγοριοποίησης ενός δείγματος σε μία κλάση (συνήθως δυαδική, κλάση 0 ή 1) στο καρτεσιανό επίπεδο. Μέσω της λογιστικής παλινδρόμησης, γίνεται εκτίμηση της πιθανότητας ενός συμβάντος να ανήκει σε μία εκ των κατηγοριών, βασισμένο στις ανεξάρτητες μεταβλητές του συνόλου δεδομένων. Έτσι, η εξαρτόμενη μεταβλητή είναι ένας αριθμός μεταξύ 0 και 1 που εκφράζει πιθανότητα. Το γενικό μαθηματικό μοντέλο που ακολουθεί η κατανομή αυτή είναι παρόμοιο με τη σιγμοειδή συνάρτηση (Εξίσωση 2.2).

Νευρωνικά Δίκτυα

Όπως θα αναλυθεί παρακάτω (Κεφάλαιο 2.2) τα Νευρωνικά Δίκτυα επιτυγχάνουν παρόμοιο έργο με αυτό που προαναφέρθηκε. Δέχονται κάποια είσοδο, η οποία συμπεριλαμβάνει κάποια χαρακτηριστικά (ανεξάρτητες μεταβλητές) και αφού αναπτυχθεί το μοντέλο εξάγουν μία πρόβλεψη (εξαρτημένη μεταβλητή) στο στρώμα εξόδου. Στη συνέχεια εκπαιδεύονται για να ελαχιστοποιήσουν το σφάλμα μεταξύ της προβλεπόμενης και της πραγματικής εξόδου, ακριβώς όπως η γραμμή παλινδρόμησης χαράσσεται βέλτιστα για να ισαπέχει από τα δείγματα.

Εφαρμογές

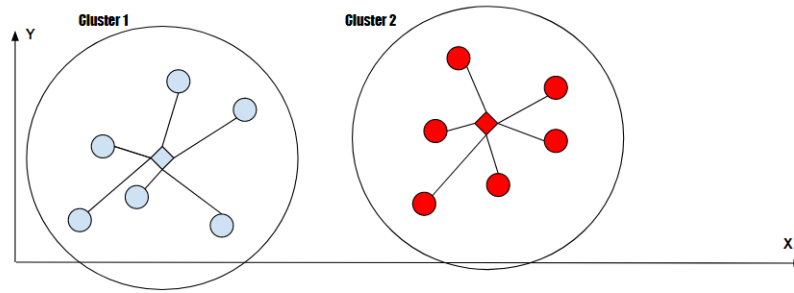
Συνοπτικά λοιπόν, εφόσον η παλινδρόμηση ανήκει στο φάσμα των συνεχών τιμών, θα έχει εφαρμογές σε τομείς όπως το χρηματιστήριο και την πρόβλεψη μετοχών ή οικονομικών δεικτών, ή στην πρόβλεψη του καιρού και την ανάλυση των μετεωρολογικών δεικτών, σε αντίθεση με την κατηγοριοποίηση που, για παράδειγμα, εξάγει συμπεράσματα για κατηγοριοποίηση εικόνων, και γενικότερα για διακριτές προβλέψεις.

2.1.4 Μάθηση χωρίς επίβλεψη

Η μη εποπτευόμενη μάθηση είναι ένα πιά απαιτητικό αλλά συνάμα πιο υποσχόμενο πεδίο, πεδίο στο οποίο η μηχανική μάθηση καθίσταται πιο αυτούσια και αυτοτελής. Στόχος της μη εποπτευόμενης μάθησης είναι να εξάγει μοτίβα, να ανακαλύψει κρυφά πρότυπα και κρυμμένες συσχετίσεις μεταξύ των δεδομένων και να δημιουργήσει ισχυρό σημείο αναφοράς για τα επερχόμενα δεδομένα, μόνο που πλέον βασίζεται σε αυθαίρετα και όχι γνωστά δείγματα [4].

Ομαδοποίηση

Clustering [4] ονομάζεται η διαδικασία ενοποίησης διάφορων οντοτήτων στο ίδιο σύνολο, με κριτήριο τις ομοιότητες που εμφανίζουν μεταξύ τους. Το σύνολο των οντοτήτων που έχουν παρόμοια χαρακτηριστικά αποτελεί μία ξεχωριστή ομάδα. Στη μηχανική μάθηση, ένας γνωστός αλγόριθμος ομαδοποίησης είναι ο K-Means, κατά τον οποίο υπολογίζεται η ευκλείδια απόσταση μεταξύ των διαφόρων σημείων στο επίπεδο, δημιουργώντας έτσι ομάδες σημείων, τα λεγόμενα clusters.



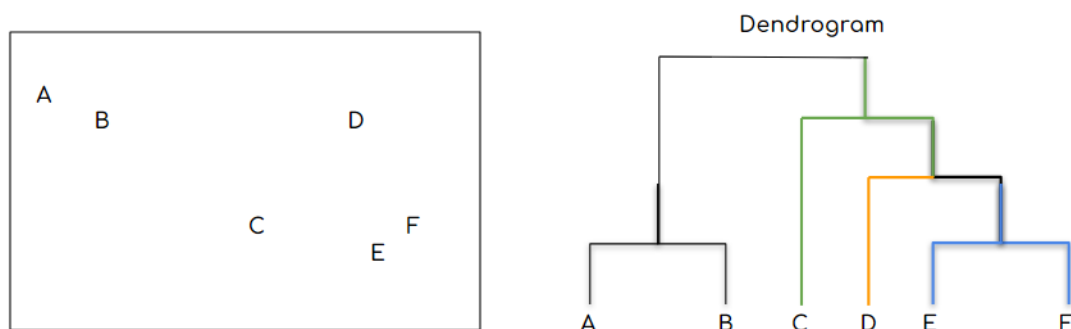
Σχήμα 2.4: K-Means Clustering με την απόσταση των σημείων από το κέντρο.

Ένα ακόμη είδος είναι η ιεραρχική ομαδοποίηση, γνωστή και ως ανάλυση ιεραρχικής ομαδοποίησης (Hierarchical Cluster Analysis). Αποτελεί έναν αλγόριθμο ομαδοποίησης που χωρίζεται σε δύο κατηγορίες, συσσωματική ή διαιρετική ομαδοποίηση.

Η συγκεντρωτική ομαδοποίηση θεωρείται μία «προσέγγιση από κάτω προς τα πάνω». Τα σημεία δεδομένων του απομονώνονται αρχικά ως ξεχωριστές ομαδοποιήσεις και στη συνέχεια συγχωνεύονται μεταξύ τους επαναληπτικά, με βάση την ομοιότητα, μέχρι να επιτευχθεί ένα σύμπλεγμα.

Αντίθετα, η διαιρετική ομαδοποίηση ακολουθεί μία προσέγγιση «από πάνω προς τα κάτω». Σε αυτήν την περίπτωση, ένα μοναδικό σύμπλεγμα δεδομένων χωρίζεται με βάση τις διαφορές μεταξύ των δεδομένων.

Αυτές οι δύο διαδικασίες ομαδοποίησης συνήθως οπτικοποιούνται χρησιμοποιώντας ένα δενδρογράφημα που τεκμηριώνει τη συγχώνευση ή τον διαχωρισμό δεδομένων.



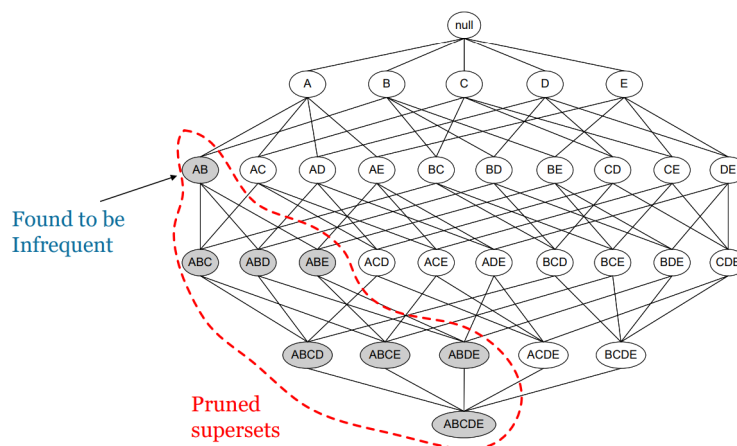
Σχήμα 2.5: Ιεραρχική ομαδοποίηση «από κάτω προς τα πάνω».

Συσχέτιση

Παρόμοια φιλοσοφία έχει και η μέθοδος Κανόνων Συσχέτισης (Association Rule Learning) [11], καθώς πάλι εντοπίζει σχέσεις μεταξύ μη συσχετιζόμενων δεδομένων, μόνο που αναφέρεται σε σχεσιακές βάσεις δεδομένων και γενικά εφαρμόζει άλλες τεχνικές. Η κύρια ανάλυση που συνδέεται με το Association Rule Mining είναι το Market Basket Analysis, ένα μαθηματικό μοντέλο βασισμένο σε κανόνες συσχέτισης και τον αλγόριθμο Apriori, που εξάγει συμπεράσματα σχετικά με την αγοραστική συνήθεια ενός πελάτη σε ένα market.

Η φιλοσοφία είναι πως υπολογίζονται κάποιοι κανόνες, οι $Support = \frac{freq(X,Y)}{N}$ και $Confidence = \frac{freq(X,Y)}{freq(X)}$, οι οποίοι αναπαριστούν τη συχνότητα εμφάνισης κάποιων X σύνολα αντικειμένων ως προς κάποια άλλα Y, όπου X και Y σύνολα αντικειμένων είναι ο συνδυασμός αγοράς N-αντικειμένων σε ένα κατάστημα και N είναι όλα τα διαφορετικά αντικείμενα. Μετέπειτα, επειδή τα σύνολα αντικειμένων είναι πολλά, και σκοπός μας είναι να συσχετισθούν όλοι οι συνδυασμοί των αγορών για να καταλήξουμε στη συνηθέστερη αγορά, εφαρμόζοντας τον αλγόριθμο Apriori, που στην προκειμένη αναδεικνύει ότι ένα συχνό σύνολο αντικειμένων καθιστά όλα τα "παιδιά" του στο δέντρο ανάλυσης ως συχνά και τα επιλέγει, ενώ όλα τα μη συχνά σύνολα αντικειμένων θα υποστούν "ψαλίδισμα" (pruning) στο δέντρο, και συνεπώς θα παραλείπονται.

Έτσι, περιορίζεται η τρομερή πολυπλοκότητα του αλγορίθμου (δοθέντων d στοιχείων θα υπάρχουν 2^d σύνολα αντικειμένων) και καταλήγουμε σε μία αποδεκτή λύση.



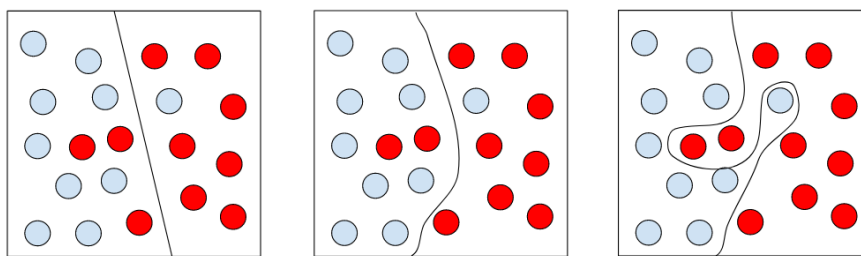
Σχήμα 2.6: Εφαρμογή Apriori σε ένα δέντρο και ψαλίδισμα ενός Infrequent Superset.

2.1.5 Προβλήματα στην εκπαίδευση

Υποπροσαρμογή και υπερπροσαρμογή

Κατά την εκπαίδευση της μηχανής, υπάρχει το ενδεχόμενο της υπερεκπαίδευσης ενός μοντέλου το οποίο μπορεί να μας οδηγήσει και στην υπερπροσαρμογή του πάνω στα δεδομένα. Η υπερπροσαρμογή συμβαίνει όταν το μοντέλο προσαρμόζεται υπερβολικά πάνω στα δεδομένα εκπαίδευσης, αποτυπώνοντας ακρίβεια και θόρυβο και κατά συνέπεια δημιουργείται ένα μη αποδοτικό μοντέλο, καθώς πλέον δε θα μπορεί να προσαρμοστεί σε νέο σετ δεδομένων δοκιμής και θα υπάρχει σφάλμα πρόβλεψης από την υπερεκπαίδευση που του συνέβη.

Στην αντίθετη περίπτωση, πιθανό είναι το φαινόμενο υποπροσαρμογής των δεδομένων εκπαίδευσης. Όταν ένα μοντέλο μηχανικής μάθησης δεν εκπαιδεύεται επαρκώς, τότε τα αποτελέσματα δεν προσεγγίζουν ορθά το ζήτημα, δηλαδή σωστές προβλέψεις με μικρές αποκλίσεις, αλλά παράλληλα δεν αποτελούν και βάση για περαιτέρω ζητήματα, όπως ακριβώς και τα υπερπροσαρμοσμένα μοντέλα.



Σχήμα 2.7: Υποπροσαρμογή και υπερπροσαρμογή μοντέλου σε σύγκριση με ένα βέλτιστο μοντέλο δυαδικής κατηγοριοποίησης.

Στο Σχήμα 2.7 γίνεται αντιληπτό ότι η υποπροσαρμογή (αριστερά) οδηγεί σε λανθάνουσα κατηγοριοποίηση, σε σύγκριση με τη βέλτιστη (μέση) κατηγοριοποίηση που φαντάζει πιο ιδανική επιλογή, ενώ η υπερπροσαρμογή (δεξιά) οδηγεί σε ένα σωστό αλλά πολύ περίπλοκο μοντέλο.

Ποιότητα και ποσότητα δεδομένων εκπαίδευσης

Ένας αλγόριθμος για να καθίσταται αποδοτικός και συνεπής στα αποτελέσματά του, όταν δέχεται καινούρια δεδομένα, σαφώς χρειάζεται να εκπαιδευτεί πάνω σε δεδομένα τα οποία εμφανίζουν ακεραιότητα και δομή. Όπως θα αναλυθεί παρακάτω, στο Κεφάλαιο 2.4, συμβαίνει εκ των προτέρων προετοιμασία στα δεδομένα

εκπαίδευσης, ώστε η εκπαίδευση να αποτελεί πιο εύκολη και άμεση διαδικασία. Τα δείγματα σε ένα σετ δεδομένων, για να έχουν ουσιώδες αντίκτυπο πρέπει να είναι ολόκληρα, να μην επαναλαμβάνονται στο σετ, να μην εμπεριέχουν κενές τιμές στις ιδιότητες τους, να είναι ταξινομημένα και πολλές άλλες τροποποιήσεις, ώστε ένα δείγμα να καθίσταται ποιοτικό.

Παράλληλα, δεδομένη είναι επίσης και η αφθονία των εν λόγω δειγμάτων. Στα περισσότερα ζητήματα μηχανικής μάθησης, ο αλγόριθμος χρειάζεται μεγάλα σύνολα δεδομένων εκπαίδευσης για να μπορέσει να μάθει τις ιδιότητες τους, καθώς όταν τα δείγματα είναι λίγα θεωρείται αναμενόμενο πως το μοντέλο θα υποπροσαρμοστεί στις ανάγκες του εκάστοτε ζητήματος.

2.2 Τεχνητά Νευρωνικά Δίκτυα

Από τις πρώτες κιόλας δεκαετίες μετά την έλευση του πρώτου ηλεκτρονικού υπολογιστή, οι ερευνητές ήταν τόσο εμπνευσμένοι από την υπολογιστική ισχύ του νευρικού μας συστήματος, που θέλησαν να προσομοιώσουν τη λειτουργία του εγκεφάλου στα υπολογιστικά συστήματα, μέσω της μηχανικής μάθησης. Ο πρώτος τεχνητός νευρώνας, ή αλλιώς Perceptron, αποτελεί και την απλούστερη δυνατή μορφή ενός νευρωνικού δικτύου. Πλέον, τα νευρωνικά δίκτυα βρίσκουν πολλά πεδία εφαρμογής, όσα δηλαδή αφορούν την κατηγοριοποίηση, την αναγνώριση προτύπων και γενικά τη βελτιστοποίηση.

2.2.1 Νευρώνας Perceptron

Η λειτουργία του απλού Perceptron, είναι να υπολογίσει έναν γραμμικό συνδυασμό που αφορά τις εισόδους που δέχεται, συναρτήσει των βαρών τους και μίας προδιάθεσης. Η εξίσωση του αποτελέσματος, μας δημιουργεί ένα υπερεπίπεδο, κατάλληλο να διαχωρίσει τα δεδομένα μας σε δύο διαφορετικά πρότυπα στο καρτεσιανό επίπεδο. Σαν συνάρτηση ενεργοποίησης (Ενότητα 2.2.4) χρησιμοποιείται η συνάρτηση προσήμου, ενώ η εξίσωση πρόβλεψης [6] είναι η εξής:

$$y = \text{sign}\left(\sum_{j=1}^d w_j x_j + b\right) \quad (2.1)$$

όπου y είναι η πρόβλεψη, w_j είναι τα επισυναπτόμενα βάρη, x_j είναι η είσοδος και b είναι η προδιάθεση.

2.2.2 Πολυστρωματικό Perceptron

Η φιλοσοφία του απλού νευρώνα έμελε να επεκταθεί, με τη δημιουργία του Multi Layer Perceptron (MLP), δηλαδή ενός δικτύου που απαρτίζεται από διαδοχικά στρώματα, με το καθένα να εμπεριέχει παραπάνω από έναν νευρώνα. Τα MLP επεκτείνουν το φάσμα των επιλύσιμων προβλημάτων, καθώς μπορούν να μας δώσουν ένα συμπέρασμα για μη γραμμικά διαχωρίσιμα ζητήματα. Ένας απλός νευρώνας είναι ικανός να δημιουργήσει ένα υπερεπίπεδο στον χώρο, σε μορφή ευθείας, το οποίο μπορεί να διαχωρίζει τα αντικείμενα του χώρου σε δύο κλάσεις, σαφώς με ένα ποσοστό σφάλματος. Από την άλλη, ένα πολυστρωματικό Perceptron, ανάλογα με τη συνάρτηση ενεργοποίησης (Ενότητα 2.2.4) που επιλέγεται και το πως είναι διαμορφωμένο, μπορεί να δημιουργήσει διαχωριστικές καμπύλες διαφόρων και περίπλοκων σχημάτων, εξαλείφοντας το ποσοστό σφάλματος αν το επιθυμεί ο δημιουργός (κάτι που δεν προτιμάται όπως ειπώθηκε στην Ενότητα 2.1.5).

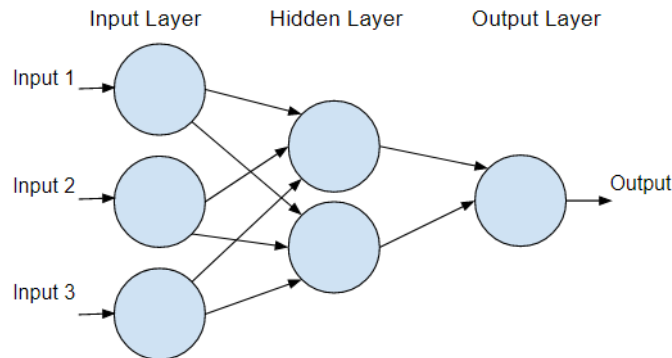
Πλέον, αξιοποιούνται σχεδόν εξ ολοκλήρου μοντέλα πολυστρωματικών δικτύων, καθώς διευρύνουν τις δυνατότητες των ερευνητών και έχουν ποικίλες εφαρμογές, τόσο στην παραδοσιακή μάθηση όσο και στη βαθιά μάθηση.

2.2.3 Νευρωνικά Δίκτυα Προσωτροφοδότησης

Ένα Τεχνητό Νευρωνικό Δίκτυο λαμβάνει την μορφή του Σχήματος 2.8 και επιτελεί το ίδιο έργο με τον απλό νευρώνα, με τη διαφορά ότι οι υπολογιστικές πράξεις που συγχέουν τα εκάστοτε βάρη με τα δεδομένα και την αντίστοιχη προδιάθεση, κάνουν πρόσθια κίνηση επηρεάζοντας το επόμενο επίπεδο [19], γνωστό και ως μοντέλο FNN ή Feed-Forward Neural Network.

Το πρώτο επίπεδο αποτελεί το επίπεδο εισόδου (Input Layer), όπου και εισέρχονται τα δεδομένα, τα οποία και ανακατευθύνονται ανάλογα με την πολυπλοκότητα και τις ανάγκες του προβλήματος στο επόμενο κρυφό επίπεδο (Hidden Layer). Εν τέλει, αφού διαπεράσει όλα τα κρυφά επίπεδα η πληροφορία, όπου ο αριθμός των επιπέδων ποικίλει όπως προειπώθηκε ανάλογα με το μοντέλο, θα καταλήξει στο επίπεδο εξόδου (Output Layer), όπου και θα γνωστοποιείται το πόρισμα το οποίο

εξάγει το δίκτυο μας, βάσει του μαθηματικού μοντέλου που δημιουργήθηκε.



Σχήμα 2.8: Τεχνητό Νευρωνικό Δίκτυο Προσωτροφοδότησης.

2.2.4 Συνάρτηση ενεργοποίησης

Κατά την προσωτροφοδότηση των δεδομένων, καλείται στο κάθε επίπεδο η επιλεγμένη συνάρτηση ενεργοποίησης δημιουργώντας μία πρόβλεψη. Στον απλό νευρώνα επιλέγεται η Συνάρτηση Προσήμου [1] (Sign ή Signum Function) καθώς η πρόβλεψη είναι δυαδικής φύσεως (πρόβλεψη ανάμεσα σε δύο κλάσεις) και γραμμικά διαχωρίσιμη. Στα MLP χρησιμοποιούνται μη γραμμικές συναρτήσεις, με συνηθισμένες επιλογές τη Σιγμοειδή 2.2 και την Υπερβολική Εφαπτομένη 2.3, ενώ υπάρχουν πολλές ακόμη.

$$\phi(u) = \frac{1}{1 + e^{-u}} \quad (2.2)$$

$$\phi(u) = \frac{e^{2u} - 1}{e^{2u} + 1} \quad (2.3)$$

2.2.5 Συνάρτηση απώλειας

Οι νευρώνες εξόδου, εξάγουν ένα πόρισμα το οποίο συνήθως είναι κοντά στο επιθυμητό αλλά όχι ίσο, καθώς έχει μία απόκλιση. Σκοπός της εκπαίδευσης των δικτύων είναι να μετριάσουν και εν τέλει να εξαλείψουν την απόκλιση αυτή.

Ο βελτιστοποιητής του δικτύου, είναι αυτός που αναλαμβάνει την εκπαίδευση (αναλύεται παρακάτω στην Ενότητα 3.1.4) και καλείται με τη χρήση κατάλληλων συναρτήσεων να βελτιστοποιήσει το δίκτυο και κατ'επέκταση να ελαχιστοποιήσει το σφάλμα. Έτσι, χρησιμοποιεί τις συναρτήσεις απώλειας για να ενημερώσει τις τιμές

των επισυναπτόμενων βαρών (Εξίσωση 2.1) που συνοδεύουν την είσοδο δεδομένων, σε μία διαδικασία που ονομάζεται οπισθοδιάδοση. Οι συναρτήσεις απώλειας [27] αναλαμβάνουν το πρωταρχικό έργο, το οποίο είναι να υπολογίσουν το σφάλμα αυτό ποσοτικά. Κάποιες από τις πιο γνωστές είναι οι εξής:

Μέσο τετραγωνικό σφάλμα

Όπως αναδεικνύει το όνομα, το μέσο τετραγωνικό σφάλμα (Mean Squared Error ή MSE) υπολογίζει τον μέσο όρο της διαφοράς πρόβλεψης και πραγματικής εξόδου, υψωμένο εις το τετράγωνο.

$$MSE = \frac{\sum_{i=1}^n (y_i - y_p)^2}{n} \quad (2.4)$$

με i να αποτελεί τον αριθμό επανάληψης, n τον αριθμό των δειγμάτων εκπαίδευσης, y_i την πραγματική έξοδο και y_p την πρόβλεψη. Αυτή η συνάρτηση είναι ιδιαίτερα χρήσιμη, καθώς ο τετραγωνικός παράγοντας εξαλείφει τις προβλέψεις που αποκλίνουν σημαντικά από την επιθυμητή τιμή και ευνοεί τις πιο κοντινές τιμές.

Μέσο απόλυτο σφάλμα

Ίδια φιλοσοφία με το μέσο τετραγωνικό σφάλμα ακολουθεί και το μέσο απόλυτο σφάλμα, απλά δε δίνει βάση στο μέγεθος του σφάλματος και αντιμετωπίζει ισάξια τις προβλέψεις.

$$MAE = \frac{\sum_{i=1}^n |y_i - y_p|}{n} \quad (2.5)$$

Η μέθοδος απολύτου σφάλματος αποτελεί μία πιο εύρωστη τεχνική από αυτήν του τετραγωνικού σφάλματος και λόγω του ότι δίνει καλύτερη ερμηνεία χρησιμοποιείται σε ζητήματα παλινδρόμησης (Ενότητα 2.1.3). Αντίθετα, το μέσο τετραγωνικό σφάλμα χρησιμοποιείται τόσο σε ζητήματα παλινδρόμησης όσο και κατηγοριοποίησης (Ενότητα 2.1.2).

Απώλεια άρθρωσης

Η Hinge Loss συνάρτηση, ουσιαστικά υπονοεί ότι η μέγιστη τιμή μεταξύ της σωστής πρόβλεψης και του αθροίσματος των λανθασμένων προβλέψεων, θα είναι η σωστή οπότε και επιλέγεται:

$$H = \sum_{i \neq j} \max(0, y_p - z) \quad (2.6)$$

Οι τιμές $i \neq j$ είναι διαφορετικές καθώς παριστάνουν τις διαφορετικές προβλέψεις y_p . Η τιμή z είναι μία σταθερά που ορίζεται (συνήθως ως μονάδα). Η απώλεια άρθρωσης συναντάται σε προβλήματα κατηγοριοποίησης και ιδιαίτερα στις μηχανές διανυσμάτων υποστήριξης.

2.2.6 Συνελικτικά Νευρωνικά Δίκτυα

Η γενική ιδέα που έχει προκύψει σαν πόρισμα από τα προτίστως αναφερόμενα, είναι πως τα τεχνητά νευρωνικά δίκτυα απαρτίζονται από νευρώνες που αναπτύσσουν ένα μαθηματικό μοντέλο, εξάγουν μία πρόβλεψη και στη συνέχεια αυτοβελτιστοποιούνται. Την ίδια λογική ακολουθούν και τα Συνελικτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks-CNN) με τη διαφορά ότι τα CNN αφορούν κυρίως ζητήματα αναγνώρισης προτύπων μέσα σε εικόνες [19]. Ένα CNN θεωρείται πιο βαθύ δίκτυο, καθώς εντάσσει νέες αρμοδιότητες στα επι μέρους στρώματα, όπως για παράδειγμα το στρώμα συσσώρευσης στο οποίο καταμερίζεται η ανάλυση (σε pixel) του δείγματος μιας εικόνας. Ακόμη, συχνή επιλογή μη γραμμικής συνάρτησης αποτελεί ο γραμμικός ανορθωτής (Rectified Linear Unit-ReLU) [20] καθώς κατά την οπισθοδιάδοση επιτελεί πιο αποδοτικό έργο από τις Sigmoid, και παρόμοιο με την Tanh:

$$\phi(u) = \max(0, u) \quad (2.7)$$

Τα συγκεκριμένα δίκτυα, είναι το σημείο αναφοράς της συγκεκριμένης έρευνας.

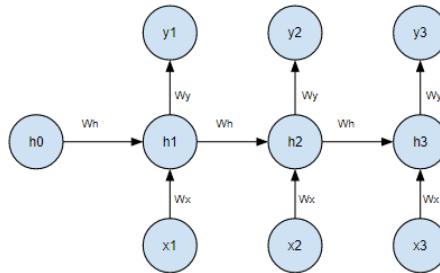
2.2.7 Νευρωνικά Δίκτυα Ανάδρασης

Ένα τελευταίο μοντέλο που θα αναφερθεί είναι τα δίκτυα ανάδρασης (Recurrent Neural Networks-RNN) [21]. Η διαφορά τους με τα συνηθισμένα πρότυπα νευρωνικών δικτύων, είναι πως επιτρέπουν συνδέσεις μεταξύ των κρυφών μονάδων συ-

σχετιζόμενες με μία χρονική καθυστέρηση, λειτουργώντας έτσι ως "μνήμη" για τις συσχετίσεις αυτές:

$$x_t = W_{rec}\sigma(x_{t-1}) + W_{in}u_t + b \quad (2.8)$$

Στην Εξίσωση 2.8 διακρίνεται η τιμή αυτής της μνήμης u_t



Σχήμα 2.9: Απλό Τεχνητό Νευρωνικό Δίκτυο Ανάδρασης με κοινά επισυναπτόμενα βάρη.

2.3 Νευρωνικά Δίκτυα και Βαθιά Μάθηση

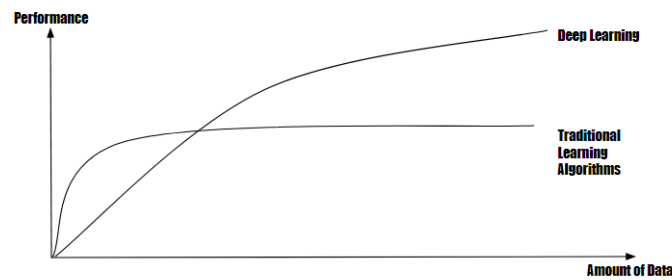
Ανάλογα με την παραμετροποίηση και το βάθος (ποικιλία) των στρωμάτων, ένα απλό νευρωνικό δίκτυο καθίσταται βαθύ νευρωνικό δίκτυο (Deep Neural Network). Οι δύο κυριότερες μορφές δικτύων βαθιάς μάθησης, τα οποία προαναφέρθηκαν, είναι τα Συνελικτικά (Ενότητα 2.2.6) και τα Δίκτυα Ανάδρασης (Ενότητα 2.2.7) με τα CNN να αποτελούν και κύριο μοντέλο της μελέτης αυτής. Πέρα από αυτά λοιπόν, σε αυτό το κεφάλαιο θα γίνει εκτενής αναφορά σε τρία ακόμα δίκτυα βαθιάς μάθησης (Autoencoder, GAN, SOM).

2.3.1 Βαθιά Μάθηση ως εξέλιξη της Μηχανικής Μάθησης

Τα τελευταία χρόνια δικαίως αναδύεται ολοένα και περισσότερο στην επιφάνεια η βαθιά μάθηση [1], καθώς μέσω της αύξησης της περιπλοκότητας του μοντέλου (σε ελεγχόμενα πλαίσια), είναι διαχειρίσιμος μεγαλύτερος όγκος δεδομένων και με μεγαλύτερη ακρίβεια από ότι η παραδοσιακή μηχανική μάθηση.

Στα πλαίσια των νευρωνικών δικτύων, βαθιά μάθηση επικρατεί όταν προσθέτονται παραπάνω στρώματα στο ήδη υπάρχον μοντέλο, αυξάνοντας έτσι τις υπολογιστικές του δυνατότητες. Αυτό βέβαια προαπαιτεί να υπάρχει και η κατάλληλη υπολογιστική ισχύς, ισχύς που στις μέρες μας πληθαίνει.

Τα βαθιά δίκτυα έχουν πολλές επιτυχείς εφαρμογές στον τομέα της τεχνολογίας. Από επεξεργασία εικόνων και ήχου, όπως αναγνώριση μοτίβων, διαστρέυλωση εικόνας, απομάκρυνση θορύβου και πολλά ακόμα, αλλά και σε πιο ειδικές κατηγορίες όπως δημιουργία ανθρώπινης φωνής, αυτόνομη οδήγηση ή αλλαγή εποχής σε ένα βίντεο.



Σχήμα 2.10: Αναπαράσταση της αποδοτικότητας της Βαθιάς Μάθησης σε σύγκριση με τη Μηχανική Μάθηση.

2.3.2 Βαθιά Μάθηση και Συνελικτικά Δίκτυα

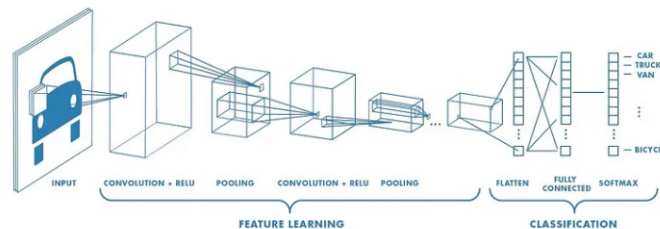
Αρχικά ας δοθεί μία πιο εκτενής προσέγγιση των CNN και της λειτουργίας αυτών. Όπως αναφέρθηκε στην Ενότητα 2.2.6, CNN είναι ένα δίκτυο προσωτροφοδότησης, που ειδικεύεται στον τομέα του pattern recognition (αναγνώριση προτύπων) σε εικόνες, και οι δύο κύριες διαφορές του με ένα απλό δίκτυο πρόσθιας διάδοσης είναι ότι διαθέτει διάφορα είδη στρωμάτων, όπως για παράδειγμα τα στρώματα συνέλιξης και συσσώρευσης και δε χρησιμοποιεί, ως συνήθως, τη Σιγμοειδή, καθώς κατα κόρον προτιμάται ο γραμμικός ανορθωτής (Εξίσωση 2.7) ή η υπερβολική εφαπτομένη (Εξίσωση 2.3).

Τα επί μέρους στρώματα αυτής της σύνθετης αρχιτεκτονικής είναι τα εξής [3]:

- **Στρώμα Συνέλιξης (Convolutional Layer):** Το κύριο στρώμα, από το οποίο το δίκτυο αποκόμισε το όνομα του, και στο οποίο γίνεται εφαρμογή φίλτρων (filter η πυρήνας) με σκοπό την αναπαράσταση όλων των πτυχών μίας εικόνας σε δύο διαστάσεις. Ο πυρήνας είναι ένας πίνακας από bit, που αλλάζει τις τιμές του ανα pixel, και εφαρμόζεται διαδοχικά σε όλη την εικόνα για να χαρτογραφήσει διάφορα σχήματα όπως γωνίες η κύκλους, και να προωθήσει τα λεγόμενα Feature Maps (Χάρτες Χαρακτηριστικών) στο επόμενο επίπεδο.

- **Στρώμα Συσσώρευσης (Pooling Layer):** Το στρώμα Συσσώρευσης, στοχεύει στο να μετριάσει την ανάλυση των pixel. Αν για παράδειγμα δεχτεί μία είσοδο 28X28 pixel, χαμηλώνει την ανάλυση στα 14X14. Αυτό επιτυγχάνεται για να μην διατηρηθεί περίπλοκο πρότυπο εικόνας, και το δίκτυο μας υπερκπαιδευτεί, οδηγώντας σε υπερπροσαρμογή.
- **Πλήρως Συνδεδεμένο Στρώμα (Fully Connected Layer):** Εδώ επενεργεί το τελικό στάδιο, στο οποίο ενώνονται όλοι οι προηγούμενοι κόμβοι και μέσω ενός κατηγοριοποιητή εξάγουν το αποτέλεσμα. Η διαδικασία κατηγοριοποίησης μετουσιώνεται αφού έχει επιτευχθεί το κατάλληλο downsizing, από n-συνδυασμό των προηγούμενων δύο στρωμάτων.
- **Στρώμα Απώλειας (Loss Layer):** Επίσης στρώμα τελικού σταδίου, υπολογίζει το σφάλμα μεταξύ επιθυμητής και πραγματικής εισόδου, όπως έχει προαναφερθεί με τη συνάρτηση απώλειας (Ενότητα 2.2.5).

Συνοπτικά λοιπόν, βαθιά μάθηση επιτυγχάνεται σε ένα CNN από τη στιγμή που θα προστεθούν πολλά Convolutional+Pooling κρυφά στρώματα, καθώς σε ένα δείγμα με πολύ υψηλή ανάλυση, για να εξαχθεί χρήσιμο υλικό από τα επιμέρους του χαρακτηριστικά πρέπει να υπάρχουν πολλά διαδοχικά τέτοια στρώματα.



Σχήμα 2.11: Αναπαράσταση Βαθιάς Μάθησης μέσω CNN [23]

2.3.3 Συνελικτικά Δίκτυα LeNet και AlexNet

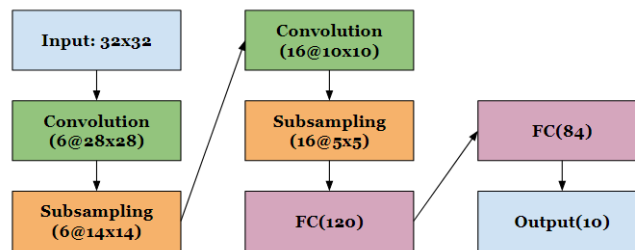
Παρακάτω γίνεται αναφορά συνοπτικά σε κάποια γνωρίσματα, που αφορούν δύο από τα πιο διάσημα και σημαντικά συνελικτικά δίκτυα.

LeNet

Ο Yann LeCun μας παρουσίασε το 1989 το πρώτο μοντέλο συνελικτικού δικτύου [14], το οποίο ήταν ικανό να αναγνωρίζει χειρόγραφους αριθμούς από μία εικόνα.

Ακόμα και σήμερα η αναγνώριση προτύπων σε χειρόγραφους αριθμούς χρησιμοποιείται συχνά, και για την εκπαίδευση τέτοιων συστημάτων υπάρχει το γνωστό σύνολο MNIST που εμπεριέχει 60.000 εικόνες από χειρόγραφους αριθμούς.

Το αρχικό δίκτυο δεχόταν εισόδους εικόνες 32X32 σε γκρι κλίμακα, δηλαδή τις αποχρώσεις που κυμαίνονται μεταξύ λευκού και μαύρου, στη συνέχεια διαδεχόταν δύο διαδικασίες εξαγωγής χαρακτηριστικών και τέλος το κάθε γράμμα κατηγοριοποιούταν από τους ταξινομητές σε μία εκ των δέκα κλάσεων (0 έως 9).

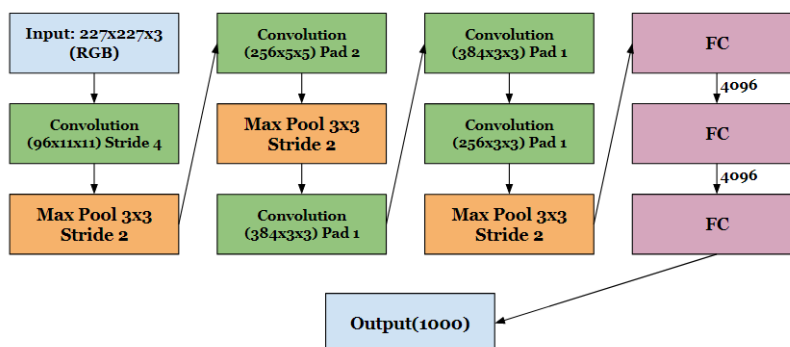


Σχήμα 2.12: Αρχιτεκτονική του LeNet.

AlexNet

Ένα πιο πρόσφατο συνελικτικό δίκτυο, και διακεκριμένο στον διαγωνισμό ImageNet του 2012, είναι το AlexNet [9], του οποίου ιδρυτής είναι ο Alex Krizhevsky.

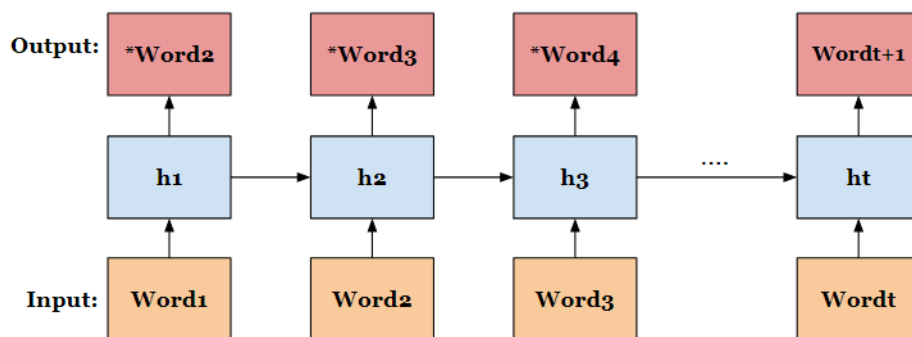
Σκοπός του διαγωνισμού ήταν να ταξινομηθούν 1.2 εκατομμύρια διαφορετικές εικόνες υψηλής ανάλυσης σε 1000 διαφορετικές κλάσεις. Το AlexNet έχει παρόμοια φιλοσοφία και έμπνευση από το LeNet, αλλά είναι πιο βαθύ και χρησιμοποιεί άλλη συνάρτηση ενεργοποίησης (ReLU αντί για Tanh). Ιδιαίτερα αξιοσημείωτο είναι πως για την επιτάχυνση της διαδικασίας εκπαίδευσης, χρησιμοποιήθηκε σύγχρονη κάρτα γραφικών υψηλής απόδοσης.



Σχήμα 2.13: Αρχιτεκτονική του AlexNet.

2.3.4 Βαθιά Μάθηση και Δίκτυα Ανάδρασης

Ένα Recurrent Neural Network (RNN) διαφέρει από τα παραδοσιακά νευρωνικά δίκτυα, καθώς λειτουργεί αναδρομικά σε συνάρτηση με τον χρόνο. Είναι ιδανικό όταν αντικείμενο μελέτης είναι οι αλληλουχιακές σχέσεις που επικρατούν σε κείμενο, ήχο ή ακόμα και εικόνα. Λειτουργεί με τη λεγόμενη βραχυπρόθεσμη μνήμη, δηλαδή όταν δέχεται μία είσοδο, όπως κείμενο όπου ειδικεύεται και περισσότερο, δημιουργεί μνήμη στα κρυφά στρώματα για να διατηρήσει τις πιθανές ακολουθιακές σχέσεις ανάμεσα στις λέξεις, και κατά συνέπεια όταν εκπαιδευτεί να αναγνωρίζει πολλά μοτίβα στον λόγο.



Σχήμα 2.14: RNN για αλληλουχία λέξεων.

Στο Σχήμα 2.14, αναδεικνύεται η πιο απλή μορφή ενός μονοστρωματικού RNN [1] που εκπαιδευτεί στην αναγνώριση κειμένου. Το h_t αναπαριστά την αναδρομική σχέση που διατηρούν οι διαδοχικές λέξεις μεταξύ τους, και ουσιαστικά αποτελεί μία μνήμη που προβλέπει τις πιθανότητες μία λέξη $Word_n$ να διαδέχεται την επόμενη λέξη $Word_{n+1}$. Το φαινόμενο αυτό ονομάζεται Language Model (Γλωσσικό Μοντέλο) και αποτελεί το πιο γνωστό μοντέλο τεχνητής νοημοσύνης που αξιοποιούνται τα λεγόμενα ChatBots.

Πέρα από αναγνώριση ακολουθιών σε κείμενο, τα RNN αξιοποιούνται και σε πολλούς άλλους τομείς όπως μετάφραση κειμένου, αναγνώριση φωνής και χειρόγραφου κειμένου, ανάθεση λεζάντας σε εικόνα (σε συνεργασία με CNN), ενώ μεταξύ άλλων οι εφαρμογές του αγγίζουν ακόμα και την υπολογιστική βιολογία.

2.3.5 Autoencoder-Generative Adversarial-Self Organizing Map

Ειδικότερα στη βαθιά μάθηση σε ένα νευρωνικό δίκτυο, υπάρχουν τρία ακόμα κυρίαρχα μοντέλα δικτύων τα οποία θα αναλυθούν εν συντομία παρακάτω, με σημαντική συνεισφορά στον τομέα της επεξεργασίας εικόνας (image processing) κυρίως στην απομάκρυνση θορύβου, και στη συμπίεση εικόνας.

Αυτόματος Κωδικοποιητής-Autoencoder

Βασισμένος στο κλασσικό δίκτυο προσωτροφοδότησης, ένας αυτόματος κωδικοποιητής επιτελεί ακριβώς το ίδιο έργο, με μία σημαντική διαφορά. Οι κόμβοι εισόδου είναι ίσοι με τους κόμβους εξόδου, και ένα εκ των κρυφών στρωμάτων (γνωστό και ως επίπεδο συμφόρησης) συμπιέζει τα υπάρχοντα δεδομένα πριν τα προωθήσει [26].

Σκοπός του δικτύου αυτού, είναι να κωδικοποιήσει μία εικόνα εισόδου και στη συνέχεια να την αποκωδικοποιήσει (Decoder), δημιουργώντας μία πανομοιότυπη έξοδο. Προφανώς αυτό δε γίνεται για να δημιουργηθεί ένα αντίγραφο της εισόδου, αφού ήδη υπάρχει, αλλά αποσκοπεί στο να εκπαιδευτεί σωστά το δίκτυο ώστε να μπορέσει να διακρίνει μοτίβα από ημιτελείς εικόνες ή εικόνες με θόρυβο, και να βγάζει σωστό αποτέλεσμα.

Το μαθηματικό μοντέλο είναι πανομοιότυπο με αυτό του νευρωνικού δικτύου προσωτροφοδότησης:

$$net_j = \sum_{i=0}^N w_{ij}x_i \quad (2.9)$$

όπου net_j είναι το βεβαρυμένο άθροισμα, w_{ij} είναι το εκάστοτε βάρος και x_i η είσοδος,

$$z_j = f(net_j) \quad (2.10)$$

όπου z_j είναι η έξοδος και f η συνάρτηση ενεργοποίησης,

$$f(s) = \frac{1}{1 + e^{-vs}} \quad (2.11)$$

ενώ η συνάρτηση ενεργοποίησης είναι η σιγμοειδής.

Παραγωγικό Αντιπαλικό Δίκτυο-Generative Adversarial Networks (GAN)

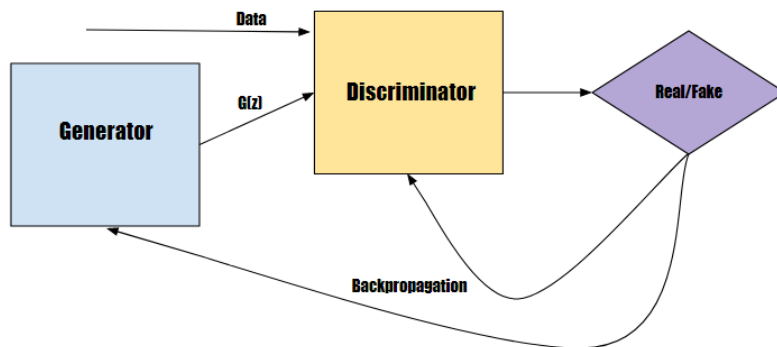
Ένα παραγωγικό αντιπαλικό δίκτυο [22] αναπαρίσταται από δύο διαφορετικά μοντέλα συγχωνευμένα στο ίδιο πολυστρωματικό μοντέλο. Αρχικά υπάρχει μία γεννήτρια η οποία είναι υπεύθυνη να παραγεί ψεύτικα δείγματα εξόδου, από θορυβώδεις εισόδους (εικόνα ή ήχο) ενώ παράλληλα διακρίνεται και το μοντέλο του διαχωριστή, που είναι παραλήπτης και όχι αποστολέας, και είναι υπεύθυνος να διακρίνει αν η είσοδος που δέχεται είναι ένα ψεύτικο δείγμα από τη γεννήτρια, ή πραγματικό δείγμα από το σετ δεδομένων.

Η εκπαίδευση σε αυτό το δίκτυο, όπως ισχυρίζονται και όλοι οι ερευνητές πάνω στα αντιπαλικά δίκτυα, είναι το αποτέλεσμα ενός παιχνιδιού με αντίπαλους τη γεννήτρια και τον διαχωριστή, το οποίο παιχνίδι σταματάει μόλις αναδειχθεί νικητής η γεννήτρια, επειδή ο διαχωριστής δεν μπορεί να αντιληφθεί αν είναι ψεύτικο ή αληθινό δείγμα (Nash equilibrium), όπου και σταματάει η εκπαίδευση.

Ο σκοπός είναι η γεννήτρια να παράγει δείγματα τόσο αληθοφανή, που ο διαχωριστής να μην αντιλαμβάνεται ότι είναι ψεύτικα, σημείο στο οποίο ο διαχωριστής εκπαιδεύεται περαιτέρω, καθώς θεωρεί ότι είναι αληθινά. Αντίθετα, όσο ο διαχωριστής αντιλαμβάνεται ότι τα δείγματα είναι ψεύτικα και παράγονται από τη γεννήτρια τότε κρίνεται νικητής, και η γεννήτρια εκπαιδεύεται για να παράγει πιο αληθοφανή δείγματα.

Η εκπαίδευση αποτυπώνεται στην Εξίσωση 2.12, ένα min,max παιχνίδι μεταξύ των δύο μοντέλων:

$$\min_g \max_d V(D, G) = E_x p_{data(x)}[\log D(x)] + E_z p_z(z)[\log(1 - D(G(z)))] \quad (2.12)$$



Σχήμα 2.15: Λειτουργία ενός Generative Adversarial Network.

Χάρτης Αυτο-Οργάνωσης SOM

Οι Αυτο-Οργανούμενοι χάρτες (Self-Organizing Map) [6] είναι υπεύθυνοι να μετασχηματίσουν ένα πρότυπο εισόδου τυχαίας διάστασης, σε ένα διακριτό χάρτη μίας ή δύο διαστάσεων (συνηθέστερο είναι το 2D).

Αποτελούν, και πάλι, κλασσικά δίκτυα προσωτροφοδότησης, μόνο που κατέχουν ένα ιδιαίτερο στρώμα (φαινόμενο που συναντήθηκε και στους Autoencoders) το γνωστό στρώμα Kohonen, στο οποίο όλοι οι νευρώνες είναι διασυνδεδεμένοι μεταξύ τους και εκεί επενεργεί η εκπαίδευση του μοντέλου.

Ίδια φιλοσοφία "παιχνιδιού" με τα GAN, μόνο που σε αυτή την περίπτωση νικητές διακρίνονται ανάμεσα στους προαναφερθέντες νευρώνες. Σε ένα συγκεκριμένο τοπολογικό δίκτυο, επικρατέστερος νευρώνας κρίνεται αυτός που έχει τη μεγαλύτερη τιμή στη συνάρτηση διάκρισης. Η διαδικασία εύρεσης νικητή ονομάζεται Ανταγωνισμός και νικητής είναι ο νευρώνας που ταιριάζει καλύτερα με το διάλυμα-πρότυπο εισόδου. Συνεπώς, η διάκριση του υπολογίζεται εν ολίγοις από την Ευκλείδεια απόσταση:

$$i(x) = \operatorname{argmin}_j \|x - w_j\| \quad (2.13)$$

όπου x είναι το πρότυπο και w τα συναπτικά βάρη.

Μετάπειτα ακολουθεί η Συνεργασία μεταξύ των νευρώνων, με τον επικρατέστερο να ορίζει τη θέση που καταλαμβάνει χωρικά ο κάθε άλλος νευρώνας στο συγκεκριμένο τοπολογικό δίκτυο. Το δίκτυο αυτό ονομάζεται αλλιώς και τοπολογική γειτονιά και αποτυπώνεται στην Εξίσωση 2.14:

$$h_{j,i(x)} = \exp\left(-\frac{d_{j,i}^2}{2\sigma^2}\right) \quad (2.14)$$

όπου $d_{j,i}$ είναι η πλευρική απόσταση μεταξύ του νικητή νευρώνα i και του διεγερμένου j και σ είναι το μέγεθος της γειτονιάς.

Τέλος, ακολουθεί η διαδικασία της Προσαρμογής των συναπτικών βαρών:

$$\Delta w_j = \eta y_j x - g(y_j) w_j \quad (2.15)$$

όπου η είναι ο ρυθμός εκμάθησης, $g(y_j)$ είναι η γραμμική συνάρτηση του y_j και y_j

είναι ίσο με $h_{j,i}$ δηλαδή την Εξίσωση 2.14.

2.4 Προεπεξεργασία δεδομένων και αρχικοποίηση

Σε αυτήν την ενότητα, αποτυπώνεται η γενική ιδέα της διπλωματικής, και γίνεται εισαγωγή του τι θα ακολουθήσει στην υλοποίηση. Η γενική ιδέα είναι ότι πάντα πρέπει να είσαι προετοιμασμένος. Έτσι, και στα νευρωνικά δίκτυα απαιτείται μία προεργασία, η οποία θα καθιστά το δίκτυο βέλτιστο ως προς τη λειτουργία και την εκπαίδευση. Αυτό πετυχαίνει η προεπεξεργασία δεδομένων (Ενότητα 2.4.1) και η βελτιστοποίηση υπερπαραμέτρων (Ενότητα 2.4.2), όσο αφορά τη δημιουργία ενός σωστά δομημένου και συντονισμένου νευρωνικού δικτύου εκ των προτέρων.

2.4.1 Προεπεξεργασία δεδομένων

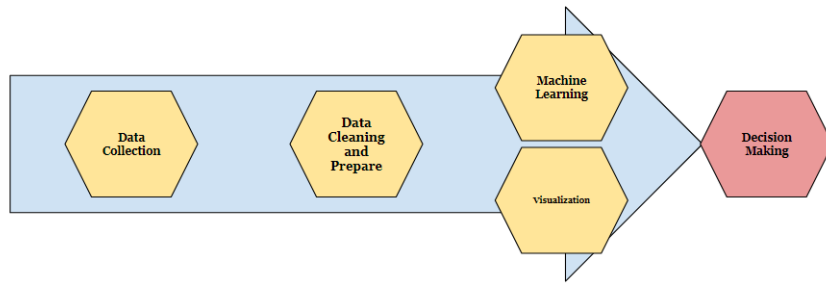
Στη μηχανική μάθηση, ο χειρισμός μεγάλων σετ δεδομένων δεν είναι ιδιαίτερα απλός, και απαιτείται εκτενής επεξεργασία αυτού για να μπορεί να λειτουργήσει οποιοδήποτε μοντέλο. Με την προεπεξεργασία δεδομένων, απομακρύνεται από το σύνολο δεδομένων οποιαδήποτε ατέλεια που θα μας εμποδίσει από το να το χρησιμοποιήσουμε.

Υπάρχουν πολλών ειδών ασάφειες σε ένα μεγάλο σύνολο, όπως η ύπαρξη κενών τιμών οι οποίες πρέπει να αντικατασταθούν με κάποια αποδεκτή τιμή που δε θα επηρεάζει το υπόλοιπο σύνολο, ή για παράδειγμα ο τύπος κάποιου δεδομένου που μπορεί να επηρεάζει την κατηγοριοποίηση και κατ'επέκταση πρέπει να αλλάξει (π.χ. από nominal σε real).

Παρόλα αυτά, στα νευρωνικά δίκτυα δε δίνεται βάση τόσο στην είσοδο, αλλά επίκεντρο προσοχής είναι οι παράμετροι που επηρεάζουν τη λειτουργία του δικτύου και οφείλεται να οριστούν ορθά εκ των προτέρων. Τέτοιες παράμετροι ονομάζονται υπερπαραμέτροι (Hyperparameters) [25] και είναι ζωτικής σημασίας παράμετροι που αρχικοποιούνται πάντα πριν εκτελεστεί το δίκτυο, όπως τα βάρη, η συνάρτηση ενεργοποίησης και ο αριθμός των εποχών.

2.4.2 Βελτιστοποίηση υπερπαραμέτρων

Οι υπερπαραμέτροι που προαναφέρθηκαν για να αρχικοποιηθούν/βελτιστοποιηθούν, υποβάλλονται στο λεγόμενο Hyper-Parameter Tuning [25, 1]. Σύμφωνα με



Σχήμα 2.16: Ορθή σειρά και οργάνωση ενεργειών για εκπαίδευση ενός μοντέλου.

τους ερευνητές, αν δε ληφθεί υπόψη ενδελεχώς η επιλογή υπερπαραμέτρων, το μοντέλο μας είναι ιδιαίτερα ασταθές, κάτι που επηρεάζεται εκθετικά και από το βάθος που έχει το δίκτυο.

Πιο συγκεκριμένα για τα δίκτυα βαθιάς μάθησης, το μαθηματικό μοντέλο είναι ιδιαίτερα περίπλοκο, οπότε είναι απαραίτητο η αρχιτεκτονική του μοντέλου να παρουσιάζει μία ομοιομορφία. Αυτό σημαίνει πως όλοι οι παράμετροι που συντελούν ένα δίκτυο πρέπει να ταιριάζουν, κάτι που επιτυγχάνεται με την βελτιστοποίηση. Αρχικά είναι σκόπιμο να καταμεριστούν τα ζητήματα ενός προβλήματος, και να οριστεί κατάλληλος αριθμός στρωμάτων για κάθε ζήτημα και αριθμός νευρώνων ανά στρώμα, δίνοντας μία συμπαγή αρχιτεκτονική, ένα μοντέλο ιδανικό να δεχθεί τις παραμέτρους του. Στα συνελικτικά δίκτυα για παράδειγμα, τέτοιοι σημαντικοί παράμετροι είναι ο ρυθμός εκμάθησης, δηλαδή η ταχύτητα με την οποία εκπαιδεύεται το δίκτυο σε κάθε εποχή, ή ο αριθμός των φίλτρων, δηλαδή το πόσα χαρακτηριστικά μίας εικόνας επιθυμούμε να εξάγουμε μετά από τη συνέλιξη. Στα γενικότερα πλαίσια όμως των νευρωνικών δικτύων, παρακάτω σχολιάζονται κάποιες από τις πιο πρωταρχικές παραμέτρους ενός μοντέλου.

Βελτιστοποίηση συνάρτησης ενεργοποίησης

Κατά την εκπαίδευση, ένας σημαντικός παράγοντας που συμβάλλει στη μείωση του χρόνου εκπαίδευσης και στην επίτευξη σωστής απόκλισης, είναι η επιλογή σωστής συνάρτησης ενεργοποίησης. Όπως αναφέραμε, δύο εκ των πιο εύχρηστων και συχνών συναρτήσεων είναι η ReLU και η Sigmoid. Άλλες έρευνες όπως αυτών των Στεργιόπουλου κ.α. [25], για τους σκοπούς των συγκεκριμένων νευρωνικών δικτύων, χρησιμοποίησε κάποιες παραλλαγές της ReLU και της Sigmoid, όπως είναι

η Gaussian Error LU (GELU) ή η Swish:

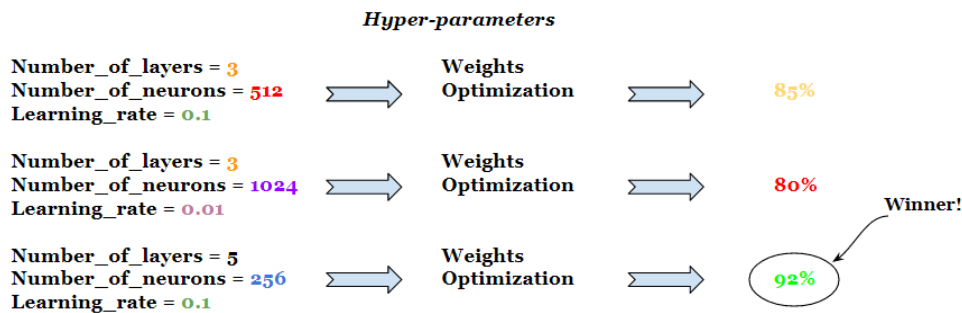
$$gelu(x) = x \cdot \frac{1}{2} [1 + erf(\frac{x}{\sqrt{2}})] \quad (2.16)$$

$$swish(x) = x \cdot sigmoid(x) \quad (2.17)$$

Βελτιστοποίηση εποχών εκπαίδευσης

Η εκπαίδευση της μηχανής είναι μία μεθοδική διαδικασία, που ακολουθεί επαναλήψεις για να μπορέσει να χωριστεί σωστά. Στην πρώτη επανάληψη, το μοντέλο εξάγει μία πρόβλεψη με τις υπάρχουσες παραμέτρους του, έπειτα επενεργεί η διαδικασία της εκπαίδευσης και αλλάζουν οι παράμετροι μέχρι να εξαχθεί και μία δεύτερη πρόβλεψη. Το χρονικό διάστημα που μεσολαβεί μεταξύ της εκπαίδευσης και της πρόβλεψης, πριν την επόμενη επανάληψη εκπαίδευσης, ονομάζεται εποχή εκπαίδευσης.

Ο αριθμός των εποχών εξαρτάται από το πως οι υπόλοιπες παράμετροι συνυπάρχουν, και τι αποτέλεσμα δίνουν κατά την εκπαίδευση [10]. Αρχικά χρησιμοποιούνται διάφοροι αριθμοί, ανα δεκάδες ή εκατοντάδες, και μετά την εκτέλεση, με παρατήρηση των αποτελεσμάτων, ορίζεται ένας μέγιστος αριθμός εποχών που μπορούν να εφαρμοστούν.



Σχήμα 2.17: Βελτιστοποίηση παραμέτρων για επίτευξη στόχων.

Αρχικοποίηση βαρών

Αφού γίνουν οι κατάλληλες επιλογές σε παραμέτρους που επηρεάζουν άμεσα την υποδομή ενός δικτύου, όπως πόσα θα είναι τα ενδιάμεσα στρώματα και τι διασυνδεσιμότητα θα έχουν, ή ποιός θα είναι ο αριθμός δεσμίδων (batch size) με τον οποίο επιλέγονται τα δεδομένα εκπαίδευσης, αμέσως μετά ακολουθεί η επιλογή

βέλτιστου αλγορίθμου αρχικοποίησης βαρών, καθώς όπως έχει γίνει αντιληπτό τα βάρη είναι ο πυλώνας που υποβαστάει κατάλληλα την επιτυχία του μαθηματικού μοντέλου.

Από την αρχή ορίστηκε (Ενότητα 2.2.1) πως σε ένα δίκτυο, τα βάρη αποτελούν μία τιμή η οποία συνοδεύει την είσοδο, ως εσωτερικό γινόμενο των δύο, με σκοπό να παράγει ένα κατάλληλο αποτέλεσμα που αν ενεργοποιηθεί σωστά, θα παραχθεί κάποιο χρήσιμο πόρισμα. Στα Συνελικτικά Δίκτυα επιτελείται η λεγόμενη συνέλιξη (Ενότητα 3.1.1) και οι αλγόριθμοι αρχικοποίησης βαρών (Ενότητα 3.1.1) όπως αναλύονται παρακάτω, επιλέγουν βάρη που αρμόζουν στη διαδικασία αυτή. Η είσοδος σε αυτά τα δίκτυα είναι μία εικόνα (σε pixel) και βάρος είναι ο πυρήνας συνέλιξης, δηλαδή πίνακες τιμών που ακολουθούν κάποια κατανομή και επενεργούν στα pixel της εικόνας.

Οι ερευνητές των Συνελικτικών, και όχι μόνο, Δικτύων έχουν καταλήξει στους εν λόγω αλγόριθμους για να αρχικοποιούν τα βάρη διαφόρων μοντέλων. Γενικά, γίνονται αναφορές στους αλγόριθμους GloN, GloU (N για Normal, U για Uniform), LeU, LeN, HeN και HeU. Πέρα από όσους αναφέρθηκαν, στο κεφάλαιο των αλγορίθμων θα γίνει αναφορά και στους RandomN, RandomU και Orthogonal, καθώς και οι εννιά συγκρίνονται στο πειραματικό πλαίσιο της έρευνας.

Από εδώ και πέρα, επίκεντρο είναι αποκλειστικά το συγκεκριμένο υποτιμήμα του κεφαλαίου, δηλαδή η μελέτη και εφαρμογή αλγορίθμων αρχικοποίησης βαρών σε νευρωνικά δίκτυα.

Κεφάλαιο 3

Αλγόριθμοι Αρχικοποίησης των Βαρών σε Νευρωνικά Δίκτυα

Σκοπός της εργασίας, είναι να αναδείξει πόσο σημαντική είναι η εύρεση κατάλληλου αρχικοποιητή βαρών (weight initializer) από την πληθώρα αλγορίθμων που παρέχει η βιβλιοθήκη Keras. Διαφορετικά μοντέλα νευρωνικών δικτύων και σετ δεδομένων αξιοποιήθηκαν, έτσι ώστε να εξαχθούν οι κατάλληλες μετρικές που θα αξιολογήσουν τις επιδόσεις ενός εκπαιδευμένου μοντέλου.

3.1 Περιγραφή της υποδομής του συστήματος

Επίκεντρο της μελέτης είναι η κατηγοριοποίηση εικόνων με τη χρήση συνελικτικών νευρωνικών δικτύων, όπου κατ'επέκταση θα αποτυπωθεί πιστά και θα αναλυθεί το προαναφερθέν γνωστό συνελικτικό δίκτυο LeNet, ενώ παράλληλα έμπνευση αποτέλεσε το AlexNet για τη δημιουργία των δικτύων βαθιάς μάθησης ModNet1 και ModNet2. Παρακάτω δίνεται περαιτέρω βάση στα επι μέρους κομμάτια που απαρτίζουν ένα συνελικτικό δίκτυο και ποιό μαθηματικό μοντέλο ακολουθούν, τόσο θεωρητικά όσο και πρακτικά.

Στη συνέχεια, αφού τα μοντέλα οριστούν επιτυχώς κατά την παραμετροποίηση, θα δοκιμαστούν διάφοροι αλγόριθμοι αρχικοποίησης βαρών, με σκοπό να εντοπιστεί ο αποδοτικότερος εξ αυτών, για το εκάστοτε δίκτυο. Η αποδοτικότητα θα αποφανθεί από τις πέντε μετρικές που εξάγονται κατά την αξιολόγηση των μοντέλων, όπου και θα αναδειχθεί ο καταλληλότερος αλγόριθμος.

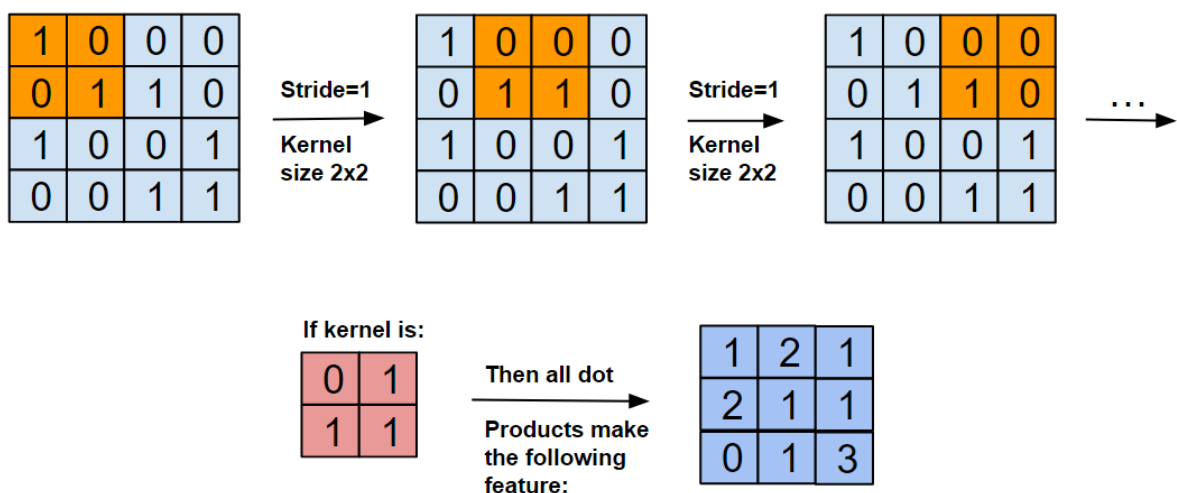
Στο επόμενο κεφάλαιο θα αναδειχθούν τα αποτελέσματα των πειραμάτων.

3.1.1 Συνέλιξη

Η διαδικασία της συνέλιξης [18] αποσκοπεί στη δημιουργία συνελικτικών δομών, δηλαδή στην εξαγωγή και αποδόμηση χαρακτηριστικών, που σαν σύνολο αποτελούν μία εικόνα. Στο στρώμα εισόδου εισάγεται μία εικόνα τριών διαστάσεων, Ύψος x Πλάτος x Βάθος, με τις πρώτες δύο να αφορούν τα pixel της εικόνας και την τρίτη το κανάλι (3 RGB αν είναι πολύχρωμη, 1 αν είναι ασπρόμαυρη). Κατά τη συνέλιξη, δημιουργείται ένας ξέχωρος διδιάστατος χάρτης ενεργοποίησης, ο οποίος σε συνδυασμό με το κανάλι/βάθος αποτελεί το εξαγόμενο χαρακτηριστικό. Παρακάτω, αναλύονται οι επιμέρους παράμετροι που συντελούν ένα συνελικτικό στρώμα.

Πυρήνας

Δοθέντος κάποιας εικόνας εισόδου, ορίζεται ένα φίλτρο ορισμένων διαστάσεων [28], το οποίο διατρέχει όλα τα pixel της εικόνας. Σε κάθε βήμα, εφαρμόζεται βαθμωτό γινόμενο μεταξύ της εικόνας και των βαρών του φίλτρου. Στη μελέτη αυτή, την επιλογή των βαρών αναλαμβάνουν οι αρχικοποιητές αλγόριθμοι (Ενότητα 3.1.1) με σκοπό το νευρωνικό δίκτυο να είναι κατάλληλα προεπεξεργασμένο, και η εκπαίδευση να οδηγήσει σε καλύτερα αποτελέσματα. Σε ένα συνελικτικό στρώμα είναι προτιμότερο να εξάγεις ποικίλα χαρακτηριστικά [2] με διάφορα φίλτρα, ανάλογα τις ανάγκες επεξεργασίας της εικόνας που μπορεί να προκύψουν.



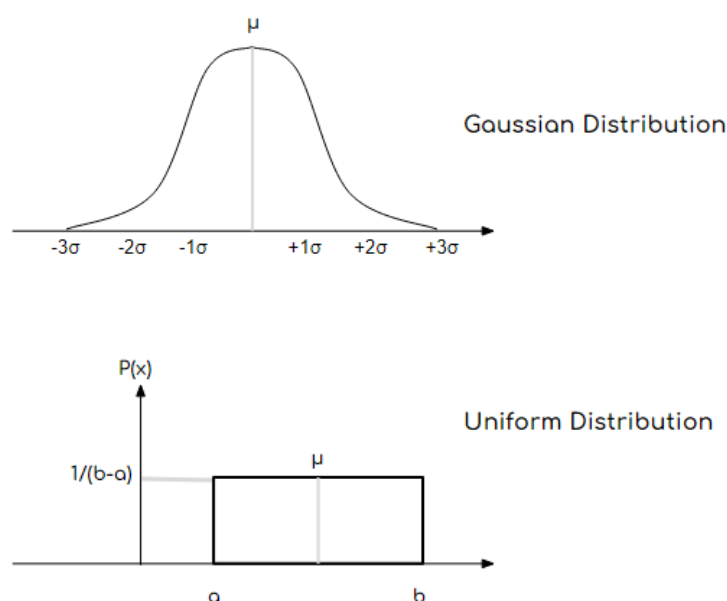
Σχήμα 3.1: Διαδικασία εξαγωγής χαρακτηριστικών

Αλγόριθμοι

Ο Yann LeCun όντας ένας από τους πρωτοπόρους στη βαθιά μάθηση, εισήγαγε ένα πρότυπο που αξιοποιεί την κατανομή πιθανοτήτων για την επιλογή βαρών, και βασίζεται στον αριθμό των μονάδων εισόδου στο διάνυσμα του βάρους. Με τη σειρά τους, διάφοροι ερευνητές με το πέρασ των χρόνων οικειοποιήθηκαν αυτή τη φιλοσοφία και τη μεταποίησαν, ανάλογα με τις ιδέες ή τις ανάγκες τους, και κατέληξαν σε μία σειρά αλγορίθμων αρχικοποίησης βαρών, που πλέον σήμερα υπάρχουν ολάκεροι και είναι διαθέσιμοι στο Keras API.

Πριν από αυτόν, η μόνη μέθοδος προεπεξεργασίας του βάρους ήταν η τυχαία επιλογή (RandomN ή RandomU), που ακολουθούσε κανονική ή ομοιόμορφη κατανομή, κατανομές που αποτελούν υπόβαθρο και για τους υπόλοιπους αλγορίθμους.

Η κανονική (Normal ή Gaussian) κατανομή, αφορά τη συγκέντρωση της πλειοψηφίας των πιθανοτήτων γύρω από το μέσο, όπου αυτή η περιοχή δημιουργείται εντός των κοντινότερων τυπικών αποκλίσεων, και την ισομερή κατάτμηση τους γύρω από αυτό. Αποτελεί την πιο σύνηθη μορφή κατανομής, καθώς πολλά σύνολα δεδομένων ακολουθούν εκ των προτέρων παρόμοια διασπορά. Η ομοιόμορφη (Uniform) κατανομή από την άλλη, δε συγκεντρώνει τις μεγαλύτερες πιθανότητες γύρω από το μέσο των στοιχείων, αλλά υποστηρίζει την ισοπίθανη κατανομή των στοιχείων σε όλα τα διαστήματα, δημιουργώντας έτσι ένα ορθογώνιο συμμετρικό πλαίσιο.



Σχήμα 3.2: Κανονική και ομοιόμορφη κατανομή.

Παρακάτω φαίνονται όλοι οι αλγόριθμοι που χρησιμοποιήθηκαν στο πειραματικό πλαίσιο [17]:

- **Orthogonal:** Χρήση τυχαίου ορθογώνιου πίνακα για αρχικοποίηση [24, 8]. Ορθογώνιος κρίνεται ένας πίνακας, το οποίου οι γραμμές και οι στήλες αποτελούν ορθογώνια μοναδιαία διανύσματα. Αποτελεί μία συχνή επιλογή στη βαθιά μάθηση, τόσο στα συνελικτικά όσο και στα δίκτυα ανάδρασης, καθώς στοχεύει στο να παραχθούν πίνακες βαρών με παρόμοιες ιδιότητες.
- **RandomNormal (RandomN):** Τυχαία κανονική κατανομή όπως ορίστηκε παραπάνω (Ενότητα 3.1.1). Σκοπός είναι ο πίνακας των βαρών να λάβει τυχαίες τιμές, που ακολουθούν Gaussian-κανονική κατάνομη, με το μέσο των τιμών για το πειραματικό πλαίσιο να ορίστηκε στο 0 ($\mu = mean = 0$) και η τυπική απόκλιση, δηλαδή κατά πόσο θετικά η αρνητικά του μέσου είναι επιτρεπτό να παρεκκλίνει η τυχαία τιμή να ορίστηκε στο ± 0.05 ($\sigma = stddev = 0.05$).
- **RandomUniform (RandomU):** Τυχαία επιλογή πίνακα βαρών ακολουθώντας ομοιόμορφη κατανομή, δηλαδή δεν υπάρχει μέση τιμή και τυπική απόκλιση τιμών από το μέσο, αλλά μονάχα τυχαία ισοπίθανη επιλογή τιμών μεταξύ ενός εύρους. Και οι δύο αλγόριθμοι αξιοποιήθηκαν από τα τέλη της δεκαετίας του 1980 ([16, 13]).
- **LecunNormal (LeN):** Ως πρωτοπόρος πριν από αρκετά χρόνια, ο LeCun [15] τόνισε τη χρησιμότητα που επιτελεί η οπισθοδιάδοση στην εκπαίδευση, και τόνισε τη σημαντικότητα που έχει η αρχικοποίηση βαρών. Η επιλογή βαρών θα πρέπει να διακυμαίνεται μεταξύ υψηλών και χαμηλών τιμών, καθώς η κλίση στις ακραίες τιμές μειώνεται σημαντικά. Έτσι, κατέληξε σε μία φόρμουλα περικομμένης κανονικής κατανομής, με μέσο $\mu = 0$ και με την τυπική απόκλιση να εξαρτάται από τον αριθμό των μονάδων εισόδου στο διάνυσμα του βάρους ($fan - in$) δηλαδή στα πλαίσια των εικόνων 1 αν είναι ασπρόμαυρη και 3 αν είναι εγχρωμη. Στο επόμενο στρώμα, οι μονάδες εισόδου είναι ίσες με τον αριθμό των εξαγόμενων χαρακτηριστικών του προηγούμενου στρώματος συνέλιξης, δηλαδή όσα είναι τα φίλτρα που εφαρμόστηκαν. Η τυπική απόκλιση ισούται με:

$$\sigma = \sqrt{\frac{1}{fan-in}}$$

- **LecunUniform (LeU)**: Αντίστοιχα, για ομοιόμορφη κατανομή, όρισε τα όρια επιλογής τυχαίων βαρών ως εξής:

$$\left[-\sqrt{\frac{3}{fan-in}}, +\sqrt{\frac{3}{fan-in}}\right]$$

- **GlorotNormal (GloN)**: Το 2010 οι γνωστοί ερευνητές Glorot και Bengio [5], ανακάλεσαν την πρόκληση που αποτελεί η διαδικασία εκπαίδευσης δικτύων βαθιάς μάθησης. Κατ'επέκταση ειδίκευσαν παραπάνω τον αλγόριθμο του LeCun, ορίζοντας πως η τυπική απόκλιση πρέπει να υπολογίζεται τόσο από τις μονάδες εισόδου, όσο και από τις μονάδες εξόδου ($fan - out$). Η μονάδα εξόδου ορίζεται ως εξής: έστω στρώμα συνέλιξης εισόδου έγχρωμης εικόνας, άρα $fan - in = 3$, αν επιθυμούμε να εφαρμόσουμε 6 φίλτρα τότε θα εξάγουμε 6 χαρακτηριστικά, δηλαδή $fan - out = 6$, το οποίο και αποτελεί $fan - in = 6$ για το επόμενο στρώμα. Το μέσο θα είναι $\mu = 0$ και η τυπική απόκλιση $\sigma = \sqrt{\frac{2}{fan-in+fan-out}}$

- **GlorotUniform (GloU)**: Αντίστοιχα, για ομοιόμορφη κατανομή, όρισαν τα όρια επιλογής τυχαίων βαρών ως εξής:

$$\left[-\sqrt{\frac{6}{fan-in+fan-out}}, +\sqrt{\frac{6}{fan-in+fan-out}}\right]$$

- **HeNormal (HeN)**: Τελευταίος χρονολογικά έρχεται ο Kaiming He με τους συνεργάτες του [7] για να αναδείξουν πως η φόρμουλα αυτή μπορεί να βελτιστοποιηθεί όταν ενεργοποιείται το δίκτυο με τον γραμμικό ανορθωτή ReLU και τις παραλλαγές του. Η ομώνυμη εξίσωση του για κανονική κατανομή, παρομοιάζεται με τον LeCun, καθώς η τυπική απόκλιση αξιοποιεί μονάχα τις μονάδες εισόδου και αυξάνει την κλιμάκωση κατά $\sqrt{2}$:

$$\sigma = \sqrt{\frac{2}{fan-in}}$$

Όπως υποστηρίζει ο He, ο γραμμικός ανορθωτής είναι ιδιαίτερα χρήσιμος και αποτρέπει συνήθως την υπερπροσαρμογή, γιαυτό και επιδίωξε το 2012 να βελτιστοποιήσει τα βάρη πάνω σε αυτόν, όταν έκανε πειράματα στο ImageNet.

- **HeUniform (HeU)**: Αντίστοιχα, για ομοιόμορφη κατανομή, όρισε τα όρια επιλογής τυχαίων βαρών ως εξής:

$$\left[-\sqrt{\frac{6}{fan-in}}, +\sqrt{\frac{6}{fan-in}}\right]$$

Βήματα

Όπως φαίνεται στο Σχήμα 3.1, η μετατόπιση του φίλτρου στον χώρο καθορίζεται από το βήμα (Stride). Στην προκειμένη με βήμα 1, φαίνεται ότι για μέγεθος φίλτρου 2×2 θα δημιουργηθεί ένα map 3×3 , καθώς το φίλτρο διαπερνά την 4×4 εικόνα 9 φορές. Με βήμα 2 αντίστοιχα γίνεται αντιληπτό ότι θα εξαχθεί ένα μικρότερο χαρακτηριστικό (2×2). Για βήμα 1, το εσωτερικό γινόμενο μεταξύ του φίλτρου και της πρώτης τετράδας ισούται με $1 * 0 + 0 * 1 + 0 * 1 + 1 * 1 = 1$, το οποίο και αποτελεί το πρώτο από τις εννιά τιμές του χαρακτηριστικού.

Το Stride, είναι ένα ιδιαίτερα χρήσιμο χαρακτηριστικό αν επιδιώκεται η μείωση της επικάλυψης (Overlap) [20], δηλαδή η αποφυγή επαναληπτικής χρήσης των ίδιων pixel στο εσωτερικό γινόμενο. Για το προαναφερθέν παράδειγμα, η επικάλυψη εξαιρείται αν επιλέγαμε βήμα 2, καθώς θα αξιοποιόταν μία τετράδα pixel τη φορά. Όσο αφορά το συγκεκριμένο πείραμα, λήφθηκαν υπόψη τα εξής:

- Στον ορισμό βήματος για το LeNet δεν ορίστηκε συγκεκριμένο stride στο στρώμα συνέλιξης, καθώς αρχικά ορίστηκε το δίκτυο με βήμα 1, κάτι βολικό μιας και η προκαθορισμένη τιμή "stride = 1" ορίζεται στο TensorFlow. Το στρώμα συσσώρευσης (Ενότητα 3.1.2) που αναλύεται παρακάτω, έχει βήμα 2 και φίλτρο 2×2 , για να εξαλειφθεί η επικάλυψη όπως προαναφέρθηκε.
- Τα μοντέλα ModNet, διαχειρίστηκαν σετ δεδομένων διαφόρων όγκων, οπότε το βήμα ορίστηκε ανάλογα με την εκάστοτε ανάγκη. Για τη συσσώρευση όμως, πάλι λήφθηκαν οι ίδιοι παράμετροι με το LeNet.

Γενικότερα, είναι σημαντικό να ορίζεται σωστό stride όχι μόνο για να αποφεύγεται η επικάλυψη, αλλά και για να εξάγονται ευδιάκριτα γεωμετρικά σχήματα και γωνίες στην αποτύπωση των χαρακτηριστικών. Ο τύπος υπολογισμού της διάστασης της εξόδου [20], βάση των όσων αναφέρθηκαν, είναι:

$$Out = 1 + \frac{N - F}{S} \quad (3.1)$$

όπου N το μέγεθος της εικόνας εισόδου, F το μέγεθος του φίλτρου και S το βήμα.

Ενίσχυση

Ένα μείζον πρόβλημα που προαναφέρθηκε, και μπορεί να επηρεάσει άμεσα την απόδοση του δικτύου, είναι η χαμένη πληροφορία που μπορεί να παραβλέψει ο πυρήνας. Το πιο σύνηθες είναι ο εξαγωγέας χαρακτηριστικών να ανιχνεύει πολλές φορές κάποιο pixel στα ενδότερα της εικόνας, και να προσπεράσει αυτά που βρίσκονται στις γωνίες της εικόνας.

Αυτό το ζήτημα λύνεται με την ενίσχυση των ορίων της εικόνας. Το padding ενισχύει το περίγραμμα της εισόδου με περισσότερα στρώματα από pixel, έτσι ώστε να αποφευχθεί η απώλεια χρήσιμης πληροφορίας αλλά και για να μπορέσουν τα επόμενα επίπεδα συνέλιξης να επεξεργαστούν πιο ποιοτική είσοδο.

Τα είδη ενίσχυσης είναι διάφορα, με επικρατέστερο να είναι το Zero-Padding δηλαδή η ενίσχυση του περιγράμματος της εικόνας με μία σειρά από μηδενικά. Άλλες χρήσιμες τεχνικές είναι το Valid-Padding το οποίο αντιστοιχεί με καθόλου ενίσχυση, και χρήση της εικόνας ως έχει, ή Reflective-Padding αν θέλουμε να αντικατοπτρίσουμε την είσοδο για να διατηρηθεί συμμετρία στην έξοδο. Για τις ανάγκες της έρευνας, χρησιμοποιήθηκε Valid-Padding σε όλα τα μοντέλα.

Η Εξίσωση 3.1 μπορεί να ενημερωθεί, εφόσον πλέον εισάχθηκε η έννοια της ενίσχυσης:

$$Out = 1 + \frac{N + 2P - F}{S} \quad (3.2)$$

όπου N το μέγεθος της εικόνας εισόδου, F το μέγεθος του φίλτρου, S το βήμα και η καινούρια παράμετρος P αντιστοιχεί στο πόσα στρώμα ενίσχυσης υπάρχουν.

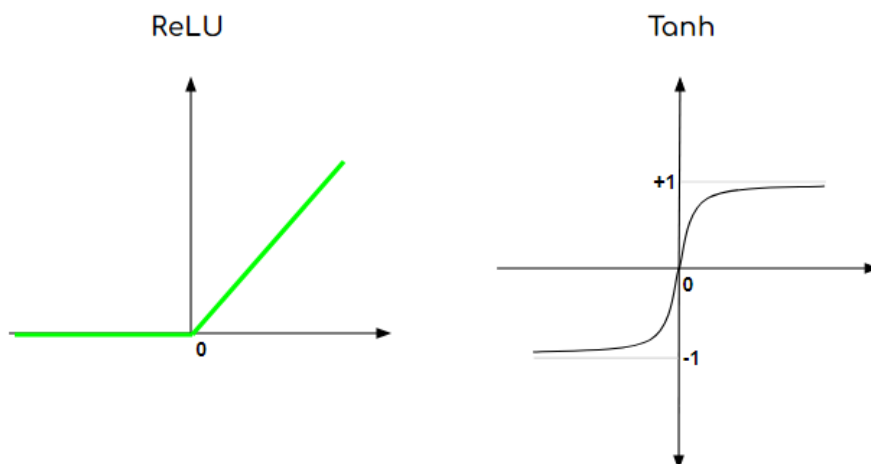
0	0	0	0	0	0
0	1	0	0	0	0
0	0	1	1	0	0
0	1	0	0	1	0
0	0	0	1	1	0
0	0	0	0	0	0

Σχήμα 3.3: Εφαρμογή ενίσχυσης ($P=1$) για την εικόνα εισόδου στο Σχήμα 3.1

Συναρτήσεις Ενεργοποίησης

Είναι σημαντικό να παραληφθούν οι περιττές πληροφορίες κατά την πρόσθια διάδοση, με τη χρήση κατάλληλης συνάρτησης ενεργοποίησης. Όπως προαναφέρθηκε, ειδικά κατά την κατηγοριοποίηση εικόνας, η συνάρτηση ενεργοποίησης επιτελεί σημαντικό έργο στο να χρήσει ένα μοντέλο μη γραμμικό, και η ορθή επιλογή της κρίνεται απαραίτητη.

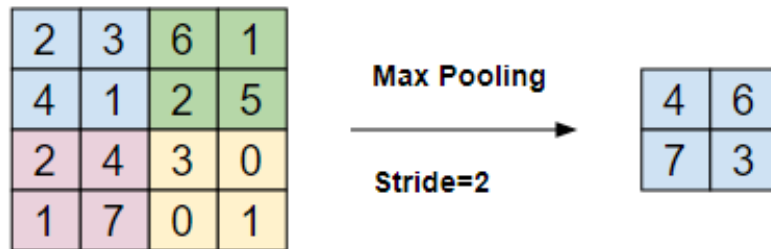
- Στα συνελικτικά μοντέλα, δημοφιλής επιλογή είναι ο γραμμικός ανορθωτής (ReLU) λόγω της απλότητας και της υπολογιστικής αποδοτικότητας που προσφέρει. Κατά την κατάβαση δυναμικού η παράγωγος του είναι 0 ή 1, γεγονός που εξαλείφει το πρόβλημα της εξαφανιζόμενης κλίσης και συνάμα διατηρεί ένα μη γραμμικό μοντέλο που εκπαιδεύεται εύκολα. Στα ModNet χρησιμοποιείται η εν λόγω συνάρτηση (Εξίσωση 2.7) όπως ορίστηκε στην ενότητα συνελικτικά νευρωνικά δίκτυα (Ενότητα 2.2.6).
- Αντίθετα, στο LeNet χρησιμοποιείται η υπερβολική εφαπτομένη (Εξίσωση 2.3). Εδώ, το ζήτημα της εξαφανιζόμενης κλίσης παραμένει σαν ενδεχόμενο κατά την εκπαίδευση, αλλά το μοντέλο και το σετ δεδομένων δεν είναι τόσο περίπλοκα που να το επιτρέπουν. Γενικά, η υπερβολική εφαπτομένη αποτελεί μία εξίσου αξιόπιστη επιλογή, καθώς με τη συμμετρικότητα μπορεί να ανιχνεύσει τόσο αρνητικές όσο και θετικές τιμές.



Σχήμα 3.4: Γραφικές παραστάσεις του γραμμικού ανορθωτή και της υπερβολικής εφαπτομένης.

3.1.2 Συσσώρευση

Μετά τα συνελικτικά στρώματα, ακολουθεί συνήθως ένα στρώμα συσσώρευσης [20] το οποίο είναι υπεύθυνο να μειώσει τη δειγματοληψία, δηλαδή να μετριάσει το μέγεθος του εξαγόμενου χαρακτηριστικού, ώστε να μειώσει την πολυπλοκότητα, διατηρώντας παράλληλα όλα τα χρήσιμα pixel.



Σχήμα 3.5: Παράδειγμα εφαρμογής Max Pooling σε ένα πλαίσιο 4x4.

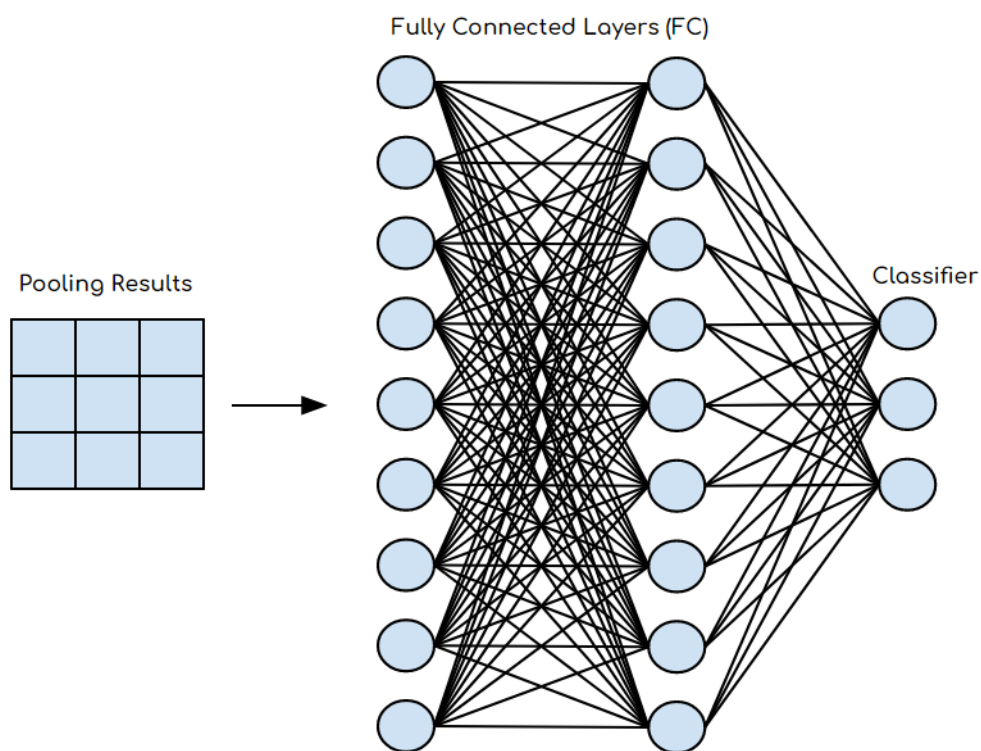
Στην πληθώρα των περιπτώσεων, όπως και στο πείραμα μας, χρησιμοποιούνται δύο κατηγορίες Pooling. Η μία φαίνεται στο Σχήμα 3.5 και αποτελεί τη συσσώρευση μεγίστου, κατά την οποία επιλέγεται η μεγαλύτερη τιμή των pixel σε μία υποομάδα. Στην προκειμένη, αν εφαρμοστεί Max Pooling 2x2 με βήμα 2, από την πάνω αριστερά τετράδα επιλέγεται η μεγαλύτερη εκ των τιμών 2, 3, 4 και 1, η οποία είναι το 4, με τη διαδικασία να επαναλαμβάνεται άλλες τρεις φορές.

Μία άλλη τεχνική είναι το Average Pooling, στο οποίο επιλέγεται η μέση τιμή μεταξύ όλων των τιμών που απαρτίζουν μία υποομάδα. Αντίστοιχα, η μέση συσσώρευση για το παραπάνω παράδειγμα και την πρώτη τετράδα θα ήταν $\frac{2+3+4+1}{2} = 5$.

Η διαδικασία της συσσώρευσης είναι ζωτικής σημασίας, καθώς βοηθάει να διατηρηθούν στο πρότυπο στιγμιότυπα με σημαντικά χαρακτηριστικά, ενώ παράλληλα δε συγκρατεί την περιττή πληροφορία και μειώνει το μέγεθος των δειγμάτων. Όλα τα προαναφερθέντα, συντελούν σε ένα μοντέλο που εκπαιδεύεται εύκολα και ταυτόχρονα δεν υπερπροσαρμόζεται σε κάποιο πρόβλημα, αν βεβαίως χρησιμοποιηθεί η κατάλληλη τεχνική συσσώρευσης, ανάλογα με το μοντέλο.

3.1.3 Πλήρως συνδεδεμένα στρώματα

Εφόσον προηγηθεί και εφαρμοστεί σωστά η συνέλιξη και η συσσώρευση, ως αποτέλεσμα αναμένουμε έναν τρισδιάστατο πίνακα. Το μόνο που απομένει, είναι να ισοπεδωθεί το αποτέλεσμα του pooling (με το στρώμα Flatten) έτσι ώστε να αποτελέσει νευρώνες, οι οποίοι είναι πλήρως συνδεδεμένοι μεταξύ τους και επιτελούν πλέον το έργο ενός απλού δικτύου προσωτροφοδότησης. Το στρώμα ισοπέδωσης, αναλαμβάνει να μετατρέψει το n -διαστάσεων πίνακα, σε ένα $n \times 1$ διάνυσμα, όπου n θα αποτελεί τον αριθμό των νευρώνων στο στρώμα.



Σχήμα 3.6: Παράδειγμα απεικόνισης ενός πίνακα συσσώρευσης να ισοπεδώνεται σε ένα διάνυσμα και μεταγενέστερα σε ένα νευρωνικό δίκτυο προσωτροφοδότησης.

Όπως φαίνεται στο Σχήμα 3.6, αν ένας πίνακας pooling 3×3 μετουσιωθεί σε ένα διάνυσμα 9×1 , θα αποτελεί νευρώνες στο στρώμα εισόδου ενός παραδοσιακού νευρωνικού δικτύου. Η πληροφορία προσωτροφοδότησης θα ενεργοποιείται με τις ίδιες συναρτήσεις ενεργοποίησης που αναφέρθηκαν στην Ενότητα 3.1.1, και στο στρώμα εξόδου, για να γίνει κατηγοριοποίηση της εισόδου, χρησιμοποιείται η συνάρτηση Softmax (Κανονικοποιημένη Εκθετική).

Κανονικοποιημένη εκθετική συνάρτηση εξόδου

Όταν η έξοδος ενός δικτύου, αποτελείται από πολλούς πραγματικούς αριθμούς που δεν συμβαδίζουν με μία κανονική κατανομή, είναι δυσκολότερο για το μηχάνημα να κατηγοριοποιήσει ένα στοιχείο.

Το πρόβλημα αυτό λύνει η Softmax function, επιτυγχάνοντας κανονικοποιημένη πιθανοτική κατανομή των τιμών. Η συνάρτηση είναι η εξής:

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad (3.3)$$

Ως είσοδο δέχεται το διάνυσμα \vec{z} το οποίο εμπεριέχει όλες τις πραγματικές τιμές. Έπειτα, αρχικά υπολογίζει τον εκθετικό όρο e^{z_i} για το καθένα από τα στοιχεία, με σκοπό να διασφαλίσει ότι η τιμή τους θα είναι από 0 έως 1 και στη συνέχεια με τον όρο $\sum_{j=1}^K e^{z_j}$ δημιουργεί την κατανομή, αφού θέτει πως το άθροισμα όλων των υπολογισμένων τιμών θα ισούται με 1.

Ο εκθετικός παράγοντας βοηθάει ιδιαίτερα στους υπολογισμούς, καθώς μία μικρή τιμή θα λάβει ιδιαίτερα χαμηλές τιμές πιθανοτήτων (όπως για παράδειγμα 0,0004) ενώ αντίθετα μεγάλες τιμές προσεγγίζουν τη μονάδα (για παράδειγμα 0,9879), γεγονός που μεγιστοποιεί τις πιθανότητες το μοντέλο μας να εξάγει σωστή πρόβλεψη.

3.1.4 Μεταγλώττιση μοντέλου

Για την ολοκλήρωση της υποδομής του μοντέλου, το μόνο που απομένει είναι να οριστούν οι παράγοντες που συμβάλλουν στην εκπαίδευση. Ένας είναι η συνάρτηση απώλειας, και η επιλογή της καταλληλότερης για να βρεθεί το σφάλμα μεταξύ επιθυμητής και πραγματικής εξόδου, δεύτερη είναι ο βελτιστοποιητής που θα χρησιμοποιήσει αυτή τη συνάρτηση για να εκπαιδεύσει το δίκτυο και τρίτη είναι ο ρυθμός εκμάθησης του δικτύου, η οποία ορίζεται μέσα στον βελτιστοποιητή.

Τόσο στα ModNet όσο και στο LeNet χρησιμοποιήθηκε η Categorical Cross-Entropy συνάρτηση απώλειας, για το LeNet χρησιμοποιήθηκε ο βελτιστοποιητής Adam και στα ModNet χρησιμοποιήθηκε Stochastic Gradient Descent.

Συνάρτηση απώλειας (CCE)

Όλα τα σετ δεδομένων που χρησιμοποιήθηκαν στο πειραματικό πλαίσιο, αφορούν ζητήματα κατηγοριοποίησης πολλαπλών κλάσεων, οπότε η επιλογή της Κατηγορηματικής Διασταυρούμενης Εντροπίας είναι μία ιδιαίτερα σύννηθης.

Η πιθανοτική κατανομή που δημιουργήθηκε μετά την ενεργοποίηση της εξόδου, πλέον συγκρίνεται με το επιθυμητό διάνυσμα εξόδου, με την εξής Εξίσωση:

$$L_{cross-entropy}(\vec{y}, y) = - \sum_i y_i * \log(\vec{y}_i) \quad (3.4)$$

Το σφάλμα μεταξύ της προβλεπόμενης εξόδου \vec{y} και της πραγματικής εξόδου y θα είναι ίσο με το αρνητικό άθροισμα όλων των γινομένων των συνδυασμών ανα δύο.

Αν, για παράδειγμα, έχουμε ένα πρόβλημα στο οποίο ζητείται να βρεθεί σε ποιά κατηγορία ταξινομείται ένα ζώο, μεταξύ των επιλογών Τίγρης, Γατα, Σκύλος και Λιοντάρι, έστω ότι θέτουμε ως είσοδο μία εικόνα ενός σκύλου. Τότε η πραγματική έξοδος του δικτύου θα έπρεπε να είναι της μορφής $[0, 0, 1, 0]$, αλλά η προβλεπόμενη έξοδος στην πρώτη εποχή είναι $[0.147, 0.540, 0.133, 0.180]$, που σημαίνει ότι το δίκτυο προβλέπει ότι απεικονίζεται μία γάτα.

Με διασταυρούμενη εντροπία, η απώλεια στην πρώτη επανάληψη θα υπολογιστεί ως εξής:

$$L = -0 * \log(0.147) - 0 * \log(0.540) - 1 * \log(0.133) - 0 * \log(0.180) \approx 2.9$$

Το σφάλμα 2.9 σαφώς είναι πολύ μεγάλο, αλλά κατά την οπισθοδιάδοση και την ενημέρωση των υπερπαραμέτρων θα μετριαστεί, έως ότου φτάσει σε μία τιμή που να παρεκκλίνει όσο το δυνατόν πιο κοντά στο 0. Αν για παράδειγμα, στη δέκατη εποχή έχει προβλεφθεί μία έξοδος της μορφής $[0.027, 0.063, 0.850, 0.060]$, η οποία με το ανθρώπινο μάτι διακρίνεται ήδη ως σωστή πρόβλεψη, τότε το σφάλμα θα υπολογιστεί ως εξής:

$$L = -0 * \log(0.027) - 0 * \log(0.063) - 1 * \log(0.850) - 0 * \log(0.060) \approx 0.07$$

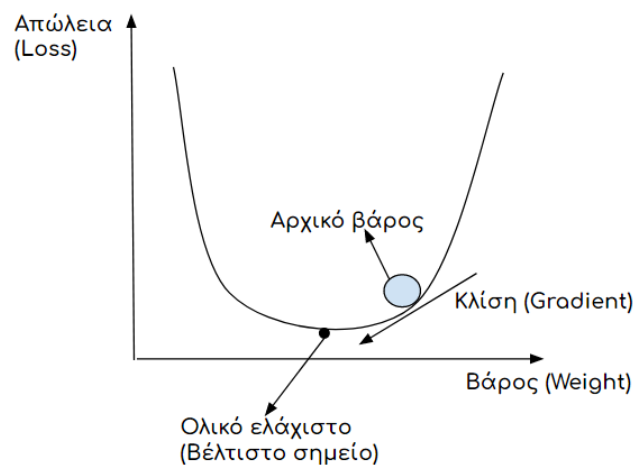
η οποία είναι πολύ προτιμότερη τιμή, και ενώ μπορεί να μετριαστεί περισσότερο, σίγουρα δε θα φτάσει ποτέ την τιμή 0, αλλά ο σκοπός επιτεύχθη και το δείγμα θα κατηγοριοποιηθεί ορθά ως Σκύλος.

Βελτιστοποιητές

- Stochastic Gradient Decent (SGD): Η κατάβαση δυναμικού (Gradient Descent) αποτελεί μία τεχνική αναζήτησης ελαχίστων σε μία συνάρτηση, υπολογίζοντας την κλίση αυτής. Η εύρεση ευθείας/κλίσης μπορεί να επιτευχθεί με τη μέθοδο Batch Gradient Descent (Κατάβαση Δυναμικού Δεσμίδας), που σημαίνει ότι όλη η δεσμίδα των δεδομένων εκπαίδευσης χρησιμοποιήθηκε για να παραχθεί το πρότυπο.

Αν βέβαια ο όγκος δεδομένων είναι μεγάλος, τότε καταφεύγουμε σε εναλλακτικές τεχνικές όπως η Στοχαστική Κατάβαση Δυναμικού (Stochastic Gradient Descent).

Η στοχαστική κατάβαση δυναμικού, είναι μία υποκατηγορία της κατάβασης δυναμικού που αφορά την τυχαιότητα επιλογής δεδομένων από το σύνολο του, δηλαδή επιλέγεται τυχαία μία μικρή δεσμίδα από όλο το σύνολο, και ορίζεται σαν δεδομένα εκπαίδευσης.



Σχήμα 3.7: Εύρεση ελαχίστου με κατάβαση δυναμικού

Γενικότερα, η κατάβαση δυναμικού στοχεύει να ενημερώσει τις υπερπαραμέτρους του δικτύου, ανάλογα με την παράγωγο της συνάρτησης σφάλματος ως προς την εκάστοτε υπερπαραμέτρο. Στο καρτεσιανό επίπεδο, σκοπός είναι η εύρεση ελαχίστου της συνάρτησης, ιδανικά ολικού για να είναι βελτιστοποιημένο το μοντέλο όπως φαίνεται στο Σχήμα 3.7.

$$\theta' = \theta - l_{rate} * \nabla L(\theta) \quad (3.5)$$

Στην Εξίσωση 3.5 περιγράφεται το πρότυπο ενημέρωσης. Ο παράγοντας $\nabla L(\theta)$ αντιπροσωπεύει την παράγωγο της συνάρτησης απώλειας (στην περίπτωση της εργασίας είναι η κατηγορηματική διασταυρούμενη εντροπία) ως προς την παράμετρο προς ενημέρωση (στην προκειμένη το βάρος). Έτσι, η ενημερωμένη παράμετρος θ' θα είναι ίση με την προηγούμενη θ αν αφαιρέσουμε τον παραγοντικό παράγοντα πολλαπλασιασμένο με το ρυθμό εκμάθησης l_{rate} , γεγονός που αναδεικνύει ότι η επιλογή κατάλληλου ρυθμού εκμάθησης επηρεάζει άμεσα το μαθηματικό μοντέλο της εκπαίδευσης.

- Adam: Ο βελτιστοποιητής αυτός προκύπτει αν συνδυαστούν δύο γνωστοί αλγόριθμοι βελτιστοποίησης, με τον έναν να είναι τον SGD και τον άλλον τον RMSProp (Root Mean Square Propagation- Μέση Τετραγωνική Διάδοση Ρίζας). Ο Adam (Adaptive Moment Estimation) κάνει εκτιμήσεις σχετικά με την κατεύθυνση και το μέγεθος των κλίσεων στη συνάρτηση, και έτσι προσαρμόζει τον ρυθμό εκμάθησης ανάλογα με την κλίση.

Η πρώτη εκτίμηση αφορά την ορμή (momentum) της κλίσης:

$$\vec{m} = \frac{m'}{1 - \beta_1} \quad (3.6)$$

όπου

$$m' = \beta_1 * m + (1 - \beta_1) * \nabla L(\theta) \quad (3.7)$$

Εδώ το διάνυσμα μέσου \vec{m} εξαρτάται από τον εκθετικό ρυθμό αποσύνθεσης β_1 . Κατά τα άλλα η Εξίσωση 3.7 έχει αρχικά $m = 0$ και αξιοποιεί όπως πριν την παράγωγο της συνάρτησης απώλειας, ως προς το βάρος θ . Ακόμη έχουμε τη δεύτερη εκτίμηση που αφορά τη διακύμανση v :

$$\vec{v} = \frac{v'}{1 - \beta_2} \quad (3.8)$$

όπου

$$v' = \beta_2 * v + (1 - \beta_2) * [\nabla L(\theta)]^2 \quad (3.9)$$

Η διακύμανση συγκαταλέγει το άθροισμα των τετραγώνων των κλίσεων, και επηρεάζεται από τον εκθετικό ρυθμό αποσύνθεσης β^2 . Οι σχέσεις 3.6 και 3.8 συγκεντρωτικά δημιουργούν την:

$$\theta' = \theta - \vec{m} * \left(\frac{l_{rate}}{\sqrt{\vec{v}} + \epsilon} \right) \quad (3.10)$$

με ϵ να αποτελεί έναν μικρό αριθμό που αποτρέπει τη διαίρεση με το μηδέν όταν η διακύμανση είναι μηδενική. Ο ρυθμός εκμάθησης δηλαδή, προσαρμόζεται για κάθε παράμετρο με βάση τα μεγέθη των κλίσεων. Εάν οι κλίσεις είναι μεγάλες, ο ρυθμός εκμάθησης μειώνεται αποτελεσματικά και εάν οι κλίσεις είναι μικρές τότε αυξάνεται.

3.2 Αξιολόγηση συστήματος-μετρικές

Παρακάτω αναλύονται κάποιες από τις πιο γνωστές μετρικές που χρησιμοποιήθηκαν και εκτιμούν την απόδοση του κάθε μοντέλου. Οι μετρικές εκτός του ότι δίνουν μία τελική προσέγγιση σχετικά με το πόσο καλά λειτουργεί το δίκτυο, αποτελούν και βάση για περαιτέρω βελτιστοποίηση του μοντέλου.

3.2.1 Ακρίβεια

Οι προβλέψεις που εξάγει ένα μοντέλο μας ενδιαφέρει να είναι σωστές, γιαυτό πρωταρχικός παράγοντας που κρίνει ένα δίκτυο είναι το Accuracy. Υπολογίζεται αν από το σύνολο όλων των προβλέψεων που εξάχθηκαν, επιλέξουμε μονάχα τις σωστές προβλέψεις, σε μία αναλογία των δύο. Η μετρική αυτή περιγράφει τη συνολική ορθότητα του μοντέλου, καθώς υπάρχει και η ακρίβεια-precision (Ενότητα 3.2.3) που περιγράφεται παρακάτω.

$$Accuracy = \frac{CorrectPredictions}{TotalPredictions} \quad (3.11)$$

Γενικότερα, συνηθίζεται σε ένα απλό μοντέλο το οποίο επεξεργάζεται λίγα δεδομένα να παρατηρούνται υψηλά ποσοστά ακρίβειας (που να τείνουν στη μονάδα) χωρίς ιδιαίτερη εκπαίδευση. Όσο μεγαλώνει η περιπλοκότητα του δικτύου και η ποσότητα των κλάσεων, είναι σημαντικά δύσκολο να βελτιστοποιηθεί το δίκτυο για

ένα ορισμένο σύνολο δεδομένων, οπότε η ακρίβεια θα κυμαίνεται σε πίο χαμηλές τιμές, εκτός αν γνωρίζουμε ήδη τις βέλτιστες παραμέτρους του μοντέλου.

3.2.2 Σφάλμα

Σφάλμα ή απώλεια είναι το νούμερο που μετράει τη διαφορά μεταξύ της επιθυμητής πρόβλεψης και της πρόβλεψης που εξήγαγε το μοντέλο, γεγονός που επιτυγχάνεται με τη συνάρτηση απώλειας (Ενότητα 2.2.5). Υπάρχουν πολλών ειδών συναρτήσεις που επιλέγονται ανάλογα το πρόβλημα, στην προκειμένη όπως προαναφέρθηκε χρησιμοποιείται η Κατηγορηματική Διασταυρούμενη Εντροπία (Ενότητα 3.1.4).

$$Loss = |DesiredOutput - ActualOutput| \quad (3.12)$$

3.2.3 Precision

Αν θέλουμε να προσεγγίσουμε ποιοτικά τις σωστές προβλέψεις ενός μοντέλου, το Precision είναι η πιο κατάλληλη μετρική. Στην περίπτωση που το ζήτημα ήταν δυαδικής κατηγοριοποίησης, έχουμε Negative και Positive class, αλλά στην προκειμένη Positive σημαίνει σωστή κλάση και Negative λάθος κλάση. Αντίστοιχα True Positives είναι άθροισμα όλων των δειγμάτων που ορθώς κατηγοριοποιήθηκαν στη σωστή κλάση, και False Positive είναι το άθροισμα όλων των δειγμάτων που κακώς κατηγοριοποιήθηκαν στη σωστή κλάση.

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives} \quad (3.13)$$

Το υψηλό Precision μας παροτρύνει να θεωρήσουμε ότι το μοντέλο συνήθως δεν κάνει λανθασμένες προβλέψεις. Αντίθετα, άμα υπάρχει μεγάλος αριθμός False Positives, όπως για παράδειγμα θετικό τεστ εγκυμοσύνης ενώ η γυναίκα δεν κυοφορεί, τότε το μοντέλο χρειάζεται βελτιστοποίηση.

3.2.4 Ανάκληση

Στην περίπτωση που χρειάζεται να ελαχιστοποιηθούν τα False Negative δείγματα, δηλαδή αυτά που κακώς κατηγοριοποιήθηκαν σε άλλη κλάση πέρα από τη σωστή, τότε πρέπει το Recall να είναι υψηλό.

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives} \quad (3.14)$$

Αντίστοιχο παράδειγμα στον τομέα της ιατρικής, θα ήταν αν ένας καρκινοπαθής ασθενής υποβληθεί σε τεστ το οποίο αναδεικνύει πως δεν είναι καρκινοπαθής.

3.2.5 F1-Score

Υπάρχει βέβαια και ένα ενδεχόμενο, τόσο το Recall όσο και το Precision να έχουν χαμηλές τιμές, ανάλογα με τα δεδομένα και τα πιθανά False Negatives η Positives. Αν μία εκ των δύο αυτών τιμών είναι χαμηλή, το F1-Score και η αρμονική του μέση θα το εντοπίσει.

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (3.15)$$

Κεφάλαιο 4

Υπολογιστική μελέτη

Στο πειραματικό πλαίσιο της εργασίας διεξάχθηκαν πειράματα, πάνω σε διαφορετικά συνελικτικά δίκτυα, τα οποία εκπαιδεύτηκαν σε ορισμένα σύνολα. Αφού επιτευχθεί η εκπαίδευση, αναδεικνύονται τα αποτελέσματα που πέτυχε ο κάθε αλγόριθμος αρχικοποίησης με μία σειρά μετρικών (Ενότητα 3.2). Τέλος, για να αποτυπωθούν πιο ρεαλιστικά και βάσιμα αποτελέσματα, από το κάθε πείραμα εξάχθηκαν αποτελέσματα από τον μέσο όρο πέντε εκτελέσεων, καθώς η εκπαίδευση επενεργεί σε διαφορετικές δεσμίδες δεδομένων και συνεπώς τα μοντέλα είναι μη αιτιοκρατικά. Τα μοντέλα και τα χαρακτηριστικά τους αναλύονται παρακάτω.

4.1 Μοντέλα εκπαίδευσης

4.1.1 LeNet

Όπως προαναφέρθηκε, το γνωστό συνελικτικό νευρωνικό δίκτυο LeNet ήταν η πρώτη επιλογή εκπαίδευσης, για να αποφανθεί το πόσο αποδοτικό μοντέλο αποτελεί σε διαφορετικά σύνολα δεδομένων. Τα επι μέρους του στρώματα, είναι παρόμοια με το πρωταρχικό δίκτυο και καταμερίζονται ως εξής, κατά σειρά:

- **Στρώμα Εισόδου-1ο Στρώμα Συνέλιξης:** Γενικός κανόνας, τον οποίο ακολουθούν και τα υπόλοιπα δίκτυα, είναι πως το στρώμα το οποίο διαδέχεται την εικόνα εισόδου σε pixel, αποτελεί και το πρώτο στη σειρά συνελικτικό στρώμα. Σε αυτό, εφαρμόζονται 6 φίλτρα με μέγεθος 5x5 το καθένα, εξάγοντας έτσι 6 χαρακτηριστικά.
- **1ο Στρώμα Συσσώρευσης:** Στη συνέχεια εφαρμόζεται μέση συσσώρευση των

χαρακτηριστικών, μεγέθους 2×2 , και έτσι τα χαρακτηριστικά μετριάζονται κατά το ήμισυ.

- **2ο Στρώμα Συνέλιξης:** Ακολουθεί και πάλι συνέλιξη των χαρακτηριστικών, αυτή τη φορά με 16 φίλτρα μεγέθους 5×5 .
- **2ο Στρώμα Συσσώρευσης:** Μέση συσσώρευση 2×2 των χαρακτηριστικών.
- **1ο εκ των 3 Πλήρως Συνδεδεμένων Στρωμάτων:** Αφού το αποτέλεσμα της συσσώρευσης ισοπεδωθεί σε ένα διάνυσμα 120×1 , αποτελεί και τους 120 νευρώνες του 1ου πλήρως συνδεδεμένου στρώματος.
- **2ο εκ των 3 Πλήρως Συνδεδεμένων Στρωμάτων:** Τα αποτελέσματα επιτυγχάνουν πρόσθια τροφοδότηση σε στρώμα 84 νευρώνων.
- **3ο εκ των 3 Πλήρως Συνδεδεμένων Στρωμάτων:** Τέλος, ακολουθεί προσω-τροφοδότηση σε 10 νευρώνες, ο καθένας εκ των οποίων αποτελεί μία κλάση του ταξινομητή.

4.1.2 ModNet

Στη συνέχεια, έχοντας αναλογιστεί τη λειτουργία και τις μεθόδους που αξιοποιούν πιο σύνθετα γνωστά συνελικτικά δίκτυα όπως το AlexNet, δημιουργήθηκαν τα λεγόμενα Modified Net (**ModNet**), όπου εδώ οι συνελικτικές διαδικασίες είναι πιο σύνθετες και τα δίκτυα πιο βαθιά, για να αποτυπώσουν καλύτερα τα πρότυπα που υπάρχουν σε έγχρωμες εικόνες.

Modnet1

- **Στρώμα Εισόδου-1ο Στρώμα Συνέλιξης:** Στο πρώτο συνελικτικό στρώμα εφαρμόζονται 32 φίλτρα με μέγεθος 3×3 το καθένα, εξάγοντας 32 χαρακτηριστικά.
- **1ο Στρώμα Συσσώρευσης:** Στη συνέχεια εφαρμόζεται μέγιστη συσσώρευση των χαρακτηριστικών, μεγέθους 2×2 .
- **2ο Στρώμα Συνέλιξης:** Ακολουθεί και πάλι συνέλιξη των χαρακτηριστικών, αυτή τη φορά με 64 φίλτρα μεγέθους 3×3 .

- **2ο Στρώμα Συσσώρευσης:** Μέγιστη συσσώρευση 2x2 των χαρακτηριστικών.
- **3ο Στρώμα Συνέλιξης:** Έπειτα ακολουθεί η 3η και τελευταία συνέλιξη, εξαγωγής 128 χαρακτηριστικών.
- **3ο Στρώμα Συσσώρευσης:** Τελική μέγιστη συσσώρευση 2x2 των χαρακτηριστικών.
- **1ο εκ των 3 Πλήρως Συνδεδεμένων Στρωμάτων:** Αφού το αποτέλεσμα της συσσώρευσης ισοπεδωθεί σε ένα διάνυσμα 256x1, αποτελεί και τους 256 νευρώνες του 1ου πλήρως συνδεδεμένου στρώματος.
- **2ο εκ των 3 Πλήρως Συνδεδεμένων Στρωμάτων:** Τα αποτελέσματα επιτυγχάνουν πρόσθια τροφοδότηση σε στρώμα 128 νευρώνων.
- **3ο εκ των 3 Πλήρως Συνδεδεμένων Στρωμάτων:** Τελική κατηγοριοποίηση των δειγμάτων σε μία εκ των 10 κλάσεων του ταξινομητή.

Το ModNet2 αρχικά δημιουργήθηκε για να εκπαιδευτεί με το Cifar100, οπότε για να επιτευχθεί σωστή κατηγοριοποίηση των δεδομένων χρησιμοποιήθηκαν παραπάνω φίλτρα από ότι στο ModNet1, όπου οι κλάσεις είναι 10 αντί για 100 και η κατηγοριοποίηση είναι πιο απλή διαδικασία:

Modnet2

- **Στρώμα Εισόδου-1ο Στρώμα Συνέλιξης:** Εφαρμόζονται 64 φίλτρα με μέγεθος 3x3.
- **2ο Στρώμα Συνέλιξης:** Πιστή επανάληψη του προηγούμενου στρώματος συνέλιξης, και πάλι με 64 φίλτρα.
- **1ο Στρώμα Συσσώρευσης:** Στη συνέχεια εφαρμόζεται μέγιστη συσσώρευση των χαρακτηριστικών, μεγέθους 2x2.
- **3ο Στρώμα Συνέλιξης:** Ακολουθεί και πάλι συνέλιξη των χαρακτηριστικών, αυτή τη φορά με 128 φίλτρα μεγέθους 3x3.
- **4ο Στρώμα Συνέλιξης:** Και πάλι, επανάληψη συνέλιξης με 128 εξαγόμενα χαρακτηριστικά.

-
- **2ο Στρώμα Συσσώρευσης:** Τελική μέγιστη συσσώρευση 2x2 των χαρακτηριστικών.
 - **1ο εκ των 2 Πλήρως Συνδεδεμένων Στρωμάτων:** 512 νευρώνες του 1ου πλήρως συνδεδεμένου στρώματος.
 - **2ο εκ των 2 Πλήρως Συνδεδεμένων Στρωμάτων:** Τα αποτελέσματα επιτυγχάνουν πρόσθια τροφοδότηση σε στρώμα 100 νευρώνων ταξινομητή.

Τα παραπάνω μοντέλα δοκιμάστηκαν με όλα τα σύνολα δεδομένων της μελέτης, οπότε στην περίπτωση αυτή το μόνο που άλλαζε στο μοντέλο ήταν οι νευρώνες στο τελικό στρώμα του ταξινομητή.

4.2 Προγραμματιστικό περιβάλλον ανάπτυξης

Μέσω της Python και της δομής TensorFlow, αξιοποιήθηκε το Keras API, μία εκτενής βιβλιοθήκη που εμπεριέχει πολλά εργαλεία για τη δημιουργία και εκπαίδευση νευρωνικών μοντέλων. Όλα τα σύνολα δεδομένων είναι ενσωματωμένα και προεπεξεργάζονται εύκολα από το Keras, σε οποιοδήποτε μοντέλο ορίσει ο χρήστης. Το κάθε μοντέλο αρχικοποιήθηκε σαν συνάρτηση σε κάθε πείραμα, και σαν μόνο όρισμα δεχόταν τον αλγόριθμο αρχικοποίησης βάρους, έτσι ώστε να εξεταστεί διαδοχικά και στους 9 αλγόριθμους.

Μετά την εκτέλεση του κάθε πειράματος εξάγονταν μετρικές (Ενότητα 3.2) για να αξιολογηθεί το μοντέλο. Οι μετρικές αυτές, για να υπολογιστούν σωστά, να αποθηκευτούν και εν τέλει να αποτυπωθούν, αξιοποιήθηκαν επίσης οι δομές NumPy και Scikit-Learn. Το κάθε πείραμα διεξάχθηκε 5 φορές, και οι μετρικές αναπαριστούν τον μέσο όρο από αυτές τις 5 επαναλήψεις, καθώς η εκπαίδευση των μοντέλων γινόταν σε τυχαίες δεσμίδες δεδομένων κάθε φορά, δηλαδή το πείραμα είναι μη ντετερμινιστικό.

Τέλος, με το πέρας της εκπαίδευσης και της αξιολόγησης των μοντέλων, αποτυπώθηκαν και γραφήματα που συγκρίνουν την ακρίβεια και το σφάλμα του κάθε μοντέλου, για τον κάθε αλγόριθμο αρχικοποίησης. Το Matplotlib, σε συνδυασμό με το ιστορικό που συλλέγει το Keras για ακρίβεια και σφάλμα κατά την εκπαίδευση, βοήθησαν στην αναπαράσταση της απόδοσης των δικτύων.

4.3 Εκπαίδευση στο LeNet

4.3.1 Σύνολο MNIST

Για αρχή μελετήθηκε ένας γνώριμος συνδυασμός μοντέλου-συνόλου δεδομένων, όπου είναι το γνωστό δίκτυο LeNet του ερευνητή Yann Lecun και το πρώτο σετ δεδομένων στο οποίο εκπαιδεύτηκε, και το οποίο είναι το MNIST.

Σε όλα τα πειράματα, όπως και σε αυτό, στα μοντέλα εφαρμόζονται οι αλγόριθμοι στο πεδίο συνέλιξης ως αρχικοποιητές πυρήνα συνέλιξης. Το συγκεκριμένο πείραμα εκτελέστηκε για να αποφανθεί αν κάποιος από τους αλγόριθμους αρχικοποίησης βαρών επηρεάζει αρνητικά τα αποτελέσματα των μετρικών, μιας και το μοντέλο αποτελεί ήδη βέλτιστη επιλογή για το σύνολο δεδομένων MNIST. Επίσης, είναι σημαντικό να σημειωθεί ότι η εκπαίδευση του μοντέλου σε κάθε αλγόριθμο διαφέρει ανα επανάληψη εκτέλεσης, καθώς τίθεται η πρόωρη παύση εκπαίδευσης εκ των προτέρων, σε περίπτωση που το μοντέλο υποεκπαιδεύεται αντί να εκπαιδεύεται σωστά.

Στον Πίνακα 4.1 δίνονται οι μετρικές στο σετ εκπαίδευσης, ενώ στον Πίνακα 4.2 δίνονται οι μετρικές όταν το μοντέλο δοκιμάζεται στο σύνολο δοκιμής:

Πίνακας 4.1: Μετρικές εκπαίδευσης για το συνδυασμό LeNet-MNIST.

LeNet-MNIST	Accuracy	Loss	Precision	Recall	F1-Score
Orthogonal	0.9959	0.0128	0.9959	0.9959	0.9959
RandomN	0.9961	0.0125	0.9961	0.9961	0.9961
RandomU	0.9954	0.0145	0.9954	0.9954	0.9954
GloN	0.9966	0.0107	0.9966	0.9967	0.9966
GloU	0.9960	0.0122	0.9960	0.9960	0.9960
HeN	0.9959	0.0134	0.9959	0.9959	0.9959
HeU	0.9966	0.0112	0.9966	0.9966	0.9966
LeN	0.9963	0.0116	0.9963	0.9963	0.9963
LeU	0.9968	0.0107	0.9968	0.9968	0.9968

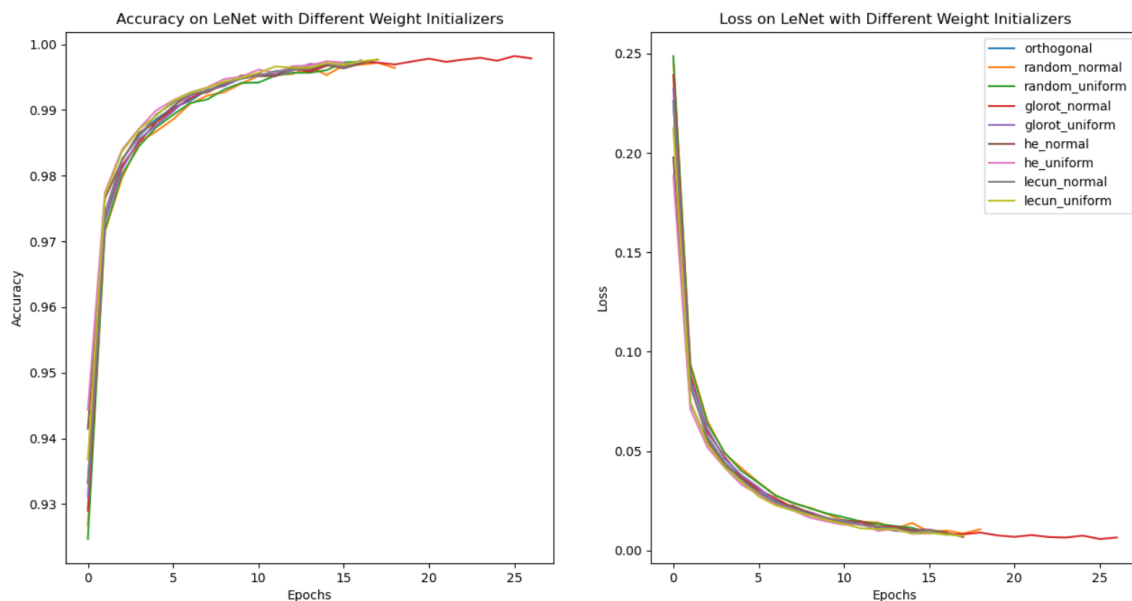
Όπως γίνεται εύκολα αντιληπτό, τα αποτελέσματα ήταν πολύ καλά και οι αλγόριθμοι σημείωσαν ιδανικούς αριθμούς στην εκπαίδευση, ενώ το ίδιο καλά αποτυπώθηκαν και στο τεστ δοκιμής. Η μέθοδος LeCun Uniform μπόρεσε να σημειώσει τις υψηλότερες μετρικές με το λιγότερο σφάλμα κατά την εκπαίδευση, ενώ ο HeU εμφάνισε τα βέλτιστα νούμερα όταν το μοντέλο δοκιμάστηκε στο σετ δοκιμής.

Στο Σχήμα 4.1 αποτυπώνεται πως διακυμαινόταν η ακρίβεια κατηγοριοποίησης

και το σφάλμα καθόλη τη διάρκεια της εκτέλεσης, για μία από τις πέντε εκτελέσεις. Σε κάθε πείραμα θα αναδεικνύεται ένα τέτοιο διάγραμμα, για να σχολιαστούν οι εποχές εκπαίδευσης ανα αλγόριθμο αλλά και αν υπήρχε κάποια σημαντική ανομοιογένεια στη συμπεριφορά κάποιου αλγορίθμου.

Πίνακας 4.2: Μετρικές δοκιμής για το συνδυασμό LeNet-MNIST.

LeNet-MNIST	Accuracy	Loss	Precision	Recall	F1-Score
Orthogonal	0.9888	0.0352	0.9887	0.9887	0.9887
RandomN	0.9869	0.0434	0.9869	0.9868	0.9868
RandomU	0.9862	0.0450	0.9863	0.9861	0.9861
GloN	0.9878	0.0436	0.9878	0.9878	0.9878
GloU	0.9873	0.0420	0.9873	0.9871	0.9872
HeN	0.9887	0.0375	0.9887	0.9886	0.9887
HeU	0.9891	0.0350	0.9891	0.9889	0.9890
LeN	0.9881	0.0393	0.9880	0.9879	0.9879
LeU	0.9881	0.0373	0.9881	0.9880	0.9880



Σχήμα 4.1: LeNet-MNIST: Ακρίβεια και Απώλεια, Εκτέλεση 5η.

Στο στιγμιότυπο της 5ης εκτέλεσης του αλγορίθμου, φαίνεται η εκτενής εκπαίδευση που δεχόταν το μοντέλο με τον GloN αλγόριθμο, καθώς έφτασε τις 27 εποχές ενώ οι υπόλοιποι σημείωσαν 15 με 18. Κάτι παρόμοιο παρατηρήθηκε και στις υπόλοιπες εκτελέσεις, με τους GloN και GloU να παρατείνουν τη διαδικασία εκπαίδευσης, ενώ ο GloN όπως φαίνεται και στον Πίνακα 4.1 είναι εξαιρετικά κοντά στις μετρικές, και με το ίδιο ελάχιστο σφάλμα, με τον αποδοτικότερο LeU.

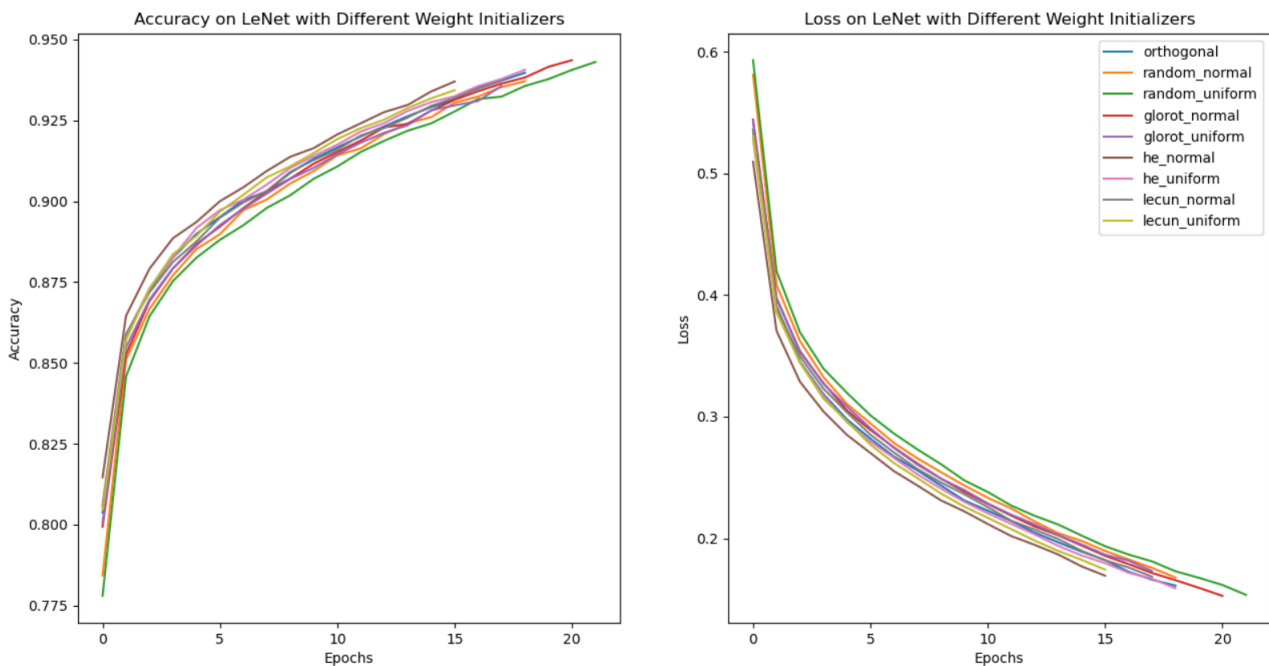
4.3.2 Σύνολο Fashion MNIST

Το ίδιο μοντέλο δοκιμάζεται και σε άλλο ένα σύνολο που εμπεριέχει ασπρόμαυρες εικόνες από ρούχα, το Fashion-MNIST. Στον Πίνακα 4.3 δίνονται οι μετρικές στο σύνολο εκπαίδευσης, όπου τόσο ο GloU και ο HeU ταξινόμησαν σωστά το 92.77% των δειγμάτων. Πέρα από την ακρίβεια όμως, ο HeU σημείωσε καλύτερα νούμερα σε όλες τις υπόλοιπες μετρικές.

Πίνακας 4.3: Μετρικές εκπαίδευσης για το συνδυασμό LeNet-Fashion MNIST.

LeNet-Fashion MNIST	Accuracy	Loss	Precision	Recall	F1-Score
Orthogonal	0.9243	0.2054	0.9252	0.9243	0.9239
RandomN	0.9210	0.2131	0.9216	0.9210	0.9207
RandomU	0.9270	0.1982	0.9276	0.9270	0.9268
GloN	0.9260	0.2010	0.9259	0.9260	0.9256
GloU	0.9277	0.1973	0.9279	0.9277	0.9274
HeN	0.9246	0.2037	0.9250	0.9246	0.9246
HeU	0.9277	0.1956	0.9282	0.9277	0.9276
LeN	0.9249	0.2046	0.9253	0.9249	0.9248
LeU	0.9208	0.2136	0.9213	0.9208	0.9208

Η 1η εκτέλεση (Σχήμα 4.2) της εκπαίδευσης αναδεικνύει επίσης τον λόγο που ο RandomU σημείωσε τις τρίτες καλύτερες επιδόσεις στην εκπαίδευση μετά τους δύο προαναφερθέντες, καθώς συνήθως διαρκούσε παραπάνω η εκπαίδευση σε αυτόν.



Σχήμα 4.2: LeNet-Fashion MNIST: Ακρίβεια και Απώλεια, Εκτέλεση 1η

Στον Πίνακα 4.4 δίνονται οι μετρικές στο σύνολο δοκιμής, όπου ο HeU συνεχίζει να αποδίδει καλύτερα από όλους τους άλλους αλγορίθμους, σε όλες τις επιμέρους μετρικές.

Πίνακας 4.4: Μετρικές δοκιμής για το συνδυασμό LeNet-Fashion MNIST.

LeNet-Fashion MNIST	Accuracy	Loss	Precision	Recall	F1-Score
Orthogonal	0.8866	0.3140	0.8877	0.8866	0.8862
RandomN	0.8835	0.3240	0.8838	0.8835	0.8831
RandomU	0.8869	0.3190	0.8875	0.8869	0.8867
GloN	0.8866	0.3199	0.8861	0.8866	0.8860
GloU	0.8870	0.3192	0.8871	0.8870	0.8867
HeN	0.8882	0.3098	0.8887	0.8882	0.8881
HeU	0.8899	0.3068	0.8908	0.8899	0.8899
LeN	0.8878	0.3133	0.8883	0.8878	0.8878
LeU	0.8868	0.3119	0.8875	0.8868	0.8868

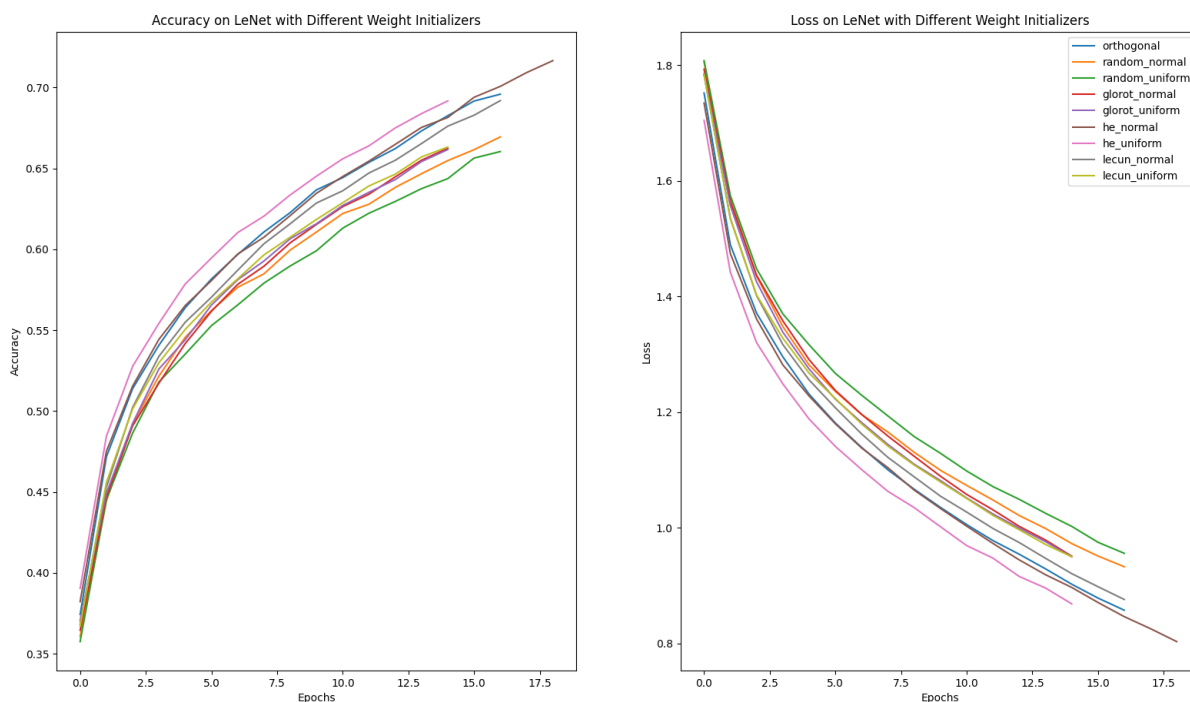
4.3.3 Σύνολο Cifar10

Η εκπαίδευση πλέον καθίσταται μία πιο απαιτητική διαδικασία για το LeNet, καθώς θα συνδυαστεί με το Cifar10 σύνολο, έγχρωμων εικόνων που διαχωρίζονται σε 10 κλάσεις. Ο HeN σημείωσε την μεγαλύτερη ακρίβεια, σχεδόν στο 65%.

Πίνακας 4.5: Μετρικές εκπαίδευσης για το συνδυασμό LeNet-Cifar10.

LeNet-Cifar10	Accuracy	Loss	Precision	Recall	F1-Score
Orthogonal	0.6320	1.0428	0.6347	0.6320	0.6298
RandomN	0.6218	1.0712	0.6208	0.6218	0.6188
RandomU	0.6245	1.0633	0.6230	0.6245	0.6214
GloN	0.6224	1.0708	0.6218	0.6224	0.6182
GloU	0.6323	1.0407	0.6326	0.6323	0.6287
HeN	0.6494	0.9962	0.6504	0.6494	0.6463
HeU	0.6415	1.0173	0.6410	0.6415	0.6381
LeN	0.6413	1.0182	0.6406	0.6413	0.6383
LeU	0.6362	1.0371	0.6369	0.6362	0.6338

Σύνηθες ήταν οι αλγόριθμοι Glorot και He να παρέτειναν την εκπαίδευση, η οποία κυμαινόταν σε χαμηλές τιμές (16-18 εποχές) λόγω της χαμηλής πολυπλοκότητας του δικτύου. Τέλος να σημειωθεί, ότι τα καλύτερα αποτελέσματα στο σετ δοκιμής (Πίνακας 4.6) και πάλι είχε ο HeN.



Σχήμα 4.3: LeNet-Cifar10: Ακρίβεια και Απώλεια, Εκτέλεση 3η

Πίνακας 4.6: Μετρικές δοκιμής για το συνδυασμό LeNet-Cifar10.

LeNet-Cifar10	Accuracy	Loss	Precision	Recall	F1-Score
Orthogonal	0.5502	1.2835	0.5525	0.5502	0.5480
RandomN	0.5394	1.3150	0.5370	0.5394	0.5358
RandomU	0.5367	1.3263	0.5350	0.5367	0.5335
GloN	0.5395	1.3133	0.5379	0.5395	0.5350
GloU	0.5448	1.3016	0.5437	0.5448	0.5404
HeN	0.5516	1.2819	0.5509	0.5516	0.5473
HeU	0.5501	1.2721	0.5478	0.5501	0.5457
LeN	0.5482	1.2922	0.5468	0.5482	0.5449
LeU	0.5499	1.2839	0.5496	0.5499	0.5471

4.3.4 Σύνολο Cifar100

Τα παρακάτω αποτελέσματα για το επόμενο σύνολο, Cifar100, αναδεικνύουν πως τα δύο συνελικτικά στρώματα του LeNet αδυνατούν πλέον να ανταπεξέλθουν στην ορθή κατηγοριοποίηση, και χρειάζεται παραπάνω στρώματα και παραμέτρους για να φτάσει αποτελέσματα όπως τα προηγούμενα. Στην εκπαίδευση (Πίνακας 4.7) για πρώτη φορά ο Orthogonal μπόρεσε να προσπεράσει τους υπόλοιπους αλγορίθμους στους μέσους όρους πέντε επαναλήψεων, με εκτεταμένη εκπαίδευση (Σχήμα

4.4) ενώ στη δοκιμή (Πίνακας 4.8) υπερτερεί ο GloU.

Πίνακας 4.7: Μετρικές εκπαίδευσης για το συνδυασμό LeNet-Cifar100.

LeNet-Cifar100	Accuracy	Loss	Precision	Recall	F1-Score
Orthogonal	0.3610	2.5549	0.3657	0.3610	0.3516
RandomN	0.3406	2.6568	0.3396	0.3406	0.3272
RandomU	0.3325	2.6989	0.3330	0.3325	0.3207
GloN	0.3398	2.6614	0.3418	0.3398	0.3285
GloU	0.3503	2.6180	0.3533	0.3503	0.3391
HeN	0.3460	2.6321	0.3509	0.3460	0.3348
HeU	0.3512	2.6096	0.3553	0.3512	0.3408
LeN	0.3436	2.6410	0.3452	0.3436	0.3318
LeU	0.3478	2.6239	0.3529	0.3478	0.3379

Πίνακας 4.8: Μετρικές δοκιμής για το συνδυασμό LeNet-Cifar100.

LeNet-Cifar100	Accuracy	Loss	Precision	Recall	F1-Score
Orthogonal	0.2550	3.1615	0.2521	0.2550	0.2452
RandomN	0.2488	3.1846	0.2411	0.2488	0.2353
RandomU	0.2509	3.1721	0.2441	0.2509	0.2384
GloN	0.2527	3.1637	0.2491	0.2527	0.2413
GloU	0.2556	3.1636	0.2543	0.2556	0.2453
HeN	0.2547	3.1592	0.2517	0.2547	0.2428
HeU	0.2549	3.1541	0.2530	0.2549	0.2449
LeN	0.2535	3.1500	0.2469	0.2535	0.2415
LeU	0.2521	3.1645	0.2492	0.2521	0.2421

4.4 Εκπαίδευση στο ModNet1

4.4.1 Σύνολο MNIST

Το μοντέλο ModNet1 αποτελεί βαθύτερο δίκτυο από το LeNet, ιδιαίτερα ικανό να ταξινομήσει δεδομένα των συνόλων MNIST και Fashion MNIST, όπου στο MNIST με τη χρήση του LeU μπόρεσαν να ταξινομηθούν σωστά το 99.51% των δεδομένων εκπαίδευσης και 98.87% των δεδομένων δοκιμής.

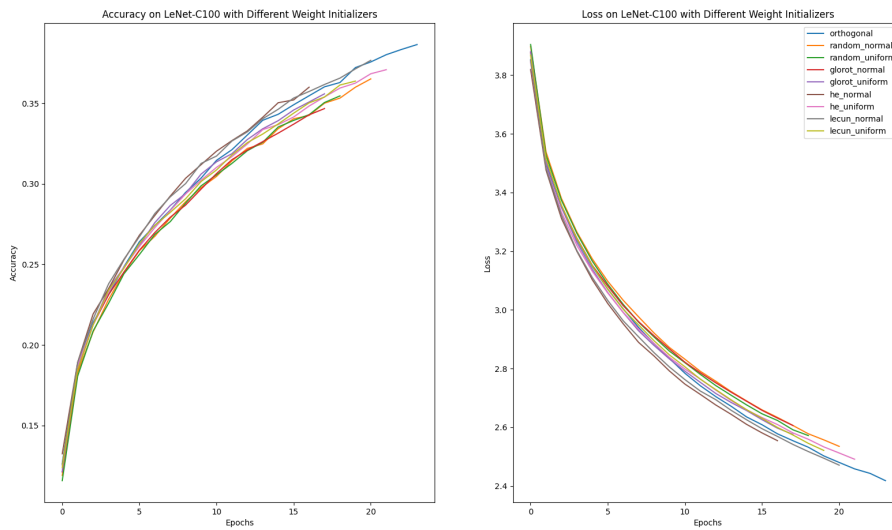
Οι Random και Glorot αλγόριθμοι (Σχήμα 4.5) παρέτειναν παραπάνω την εκπαίδευση.

Πίνακας 4.9: Μετρικές εκπαίδευσης για το συνδυασμό ModNet1-MNIST.

ModNet1-MNIST	Accuracy	Loss	Precision	Recall	F1-Score
Orthogonal	0.9948	0.0176	0.9948	0.9948	0.9948
RandomN	0.9932	0.0228	0.9932	0.9932	0.9932
RandomU	0.9925	0.0249	0.9924	0.9924	0.9924
GloN	0.9947	0.0180	0.9947	0.9947	0.9947
GloU	0.9944	0.0184	0.9944	0.9944	0.9944
HeN	0.9941	0.0198	0.9941	0.9941	0.9941
HeU	0.9950	0.0169	0.9950	0.9950	0.9950
LeN	0.9951	0.0163	0.9951	0.9951	0.9951
LeU	0.9955	0.0157	0.9955	0.9955	0.9955

Πίνακας 4.10: Μετρικές δοκιμής για το συνδυασμό ModNet1-MNIST.

ModNet1-MNIST	Accuracy	Loss	Precision	Recall	F1-Score
Orthogonal	0.9879	0.0468	0.9879	0.9878	0.9878
RandomN	0.9863	0.0490	0.9863	0.9862	0.9862
RandomU	0.9857	0.0498	0.9856	0.9855	0.9856
GloN	0.9873	0.0486	0.9873	0.9872	0.9872
GloU	0.9881	0.0458	0.9880	0.9880	0.9880
HeN	0.9870	0.0485	0.9870	0.9869	0.9870
HeU	0.9881	0.0472	0.9881	0.9880	0.9880
LeN	0.9880	0.0457	0.9880	0.9880	0.9880
LeU	0.9887	0.0445	0.9887	0.9886	0.9886



Σχήμα 4.4: LeNet-Cifar100: Ακρίβεια και Απώλεια, Εκτέλεση 3η

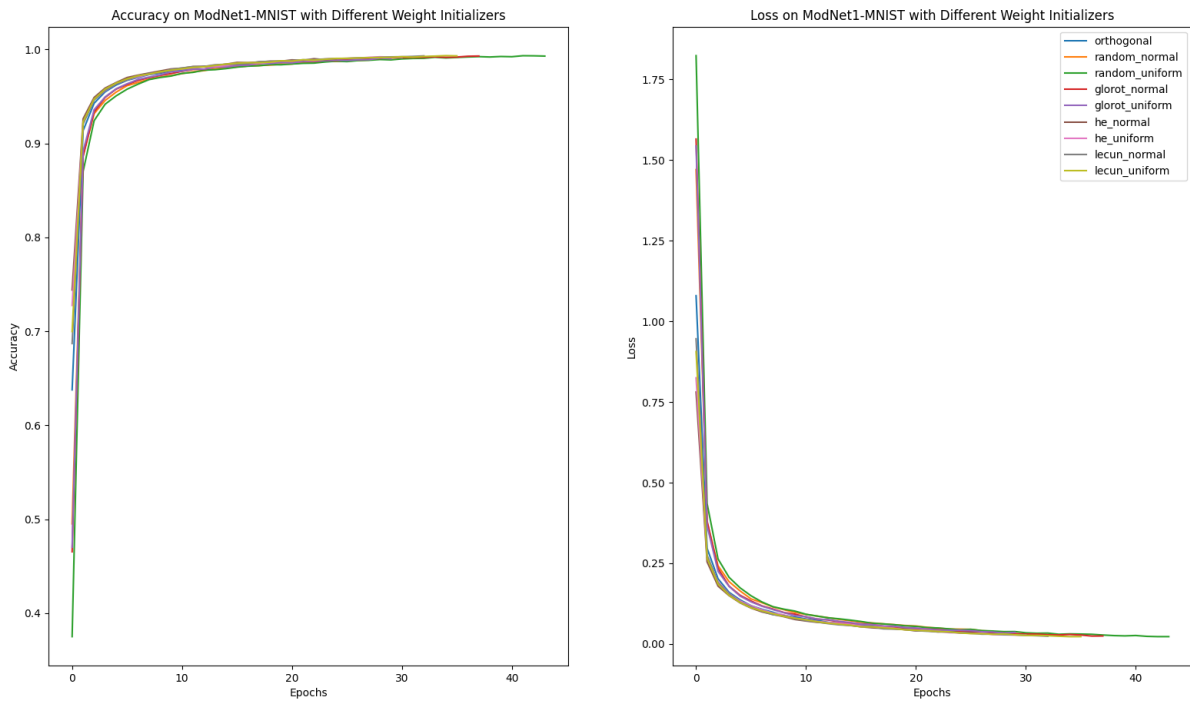
4.4.2 Σύνολο Fashion MNIST

Η υπεροχή του LeU αλγόριθμου φαίνεται και στο πείραμα πάνω στο Fashion MNIST, καθώς το ModNet1 μπόρεσε να σημειώσει τις καλύτερες μετρικές με τον LeU αρχικοποιητή τόσο στην εκπαίδευση όσο και στη δοκιμή.

Πίνακας 4.11: Μετρικές εκπαίδευσης για το συνδυασμό ModNet1-Fashion MNIST.

ModNet1-FMNIST	Accuracy	Loss	Precision	Recall	F1-Score
Orthogonal	0.9250	0.2031	0.9249	0.9250	0.9248
RandomN	0.9240	0.2069	0.9241	0.9240	0.9239
RandomU	0.9243	0.2048	0.9243	0.9243	0.9240
GloN	0.9313	0.1877	0.9316	0.9313	0.9313
GloU	0.9243	0.2028	0.9252	0.9243	0.9244
HeN	0.9274	0.1982	0.9275	0.9274	0.9271
HeU	0.9260	0.2005	0.9261	0.9260	0.9258
LeN	0.9282	0.1950	0.9287	0.9282	0.9280
LeU	0.9348	0.1790	0.9349	0.9348	0.9345

Στην εκπαίδευση, φαίνεται στο Σχήμα 4.6, ο RandomU προσέγγιζε την εκπαίδευση των 70 εποχών, με τους Glorot να ακολουθούσαν και στις υπόλοιπες εκτελέσεις, σημειώνοντας αμφότεροι σημαντικά νούμερα, λίγο πιο κάτω από τον LeU.



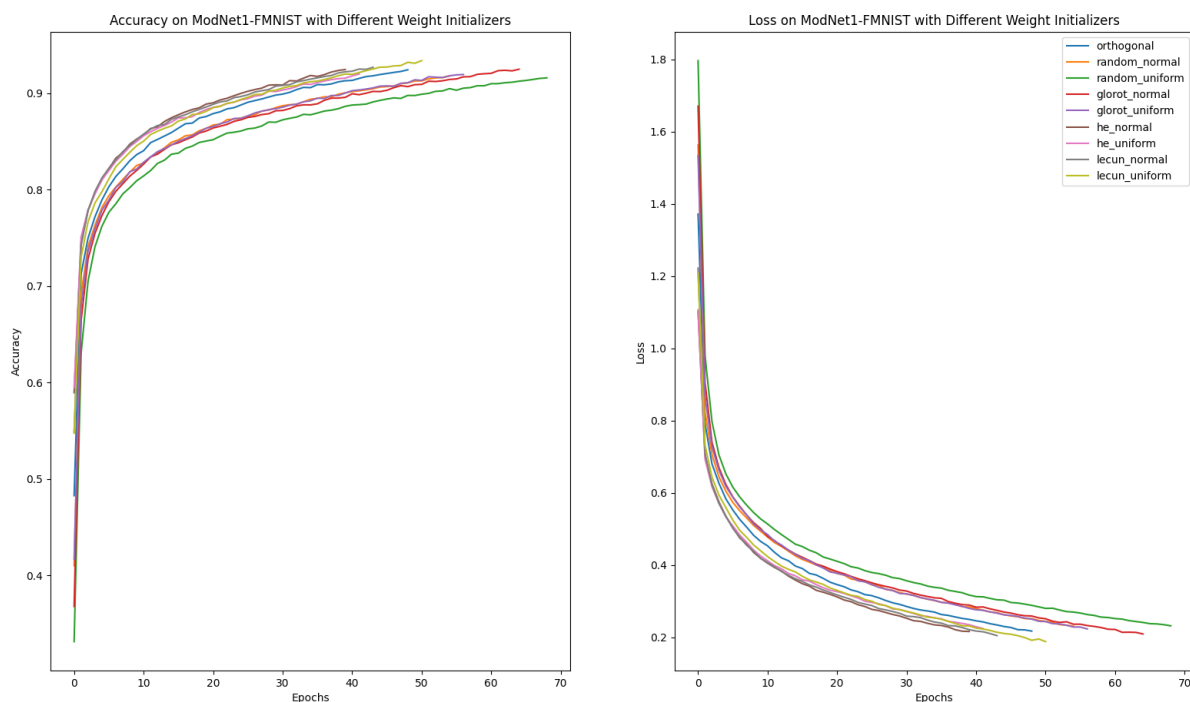
Σχήμα 4.5: ModNet1-MNIST: Ακρίβεια και Απώλεια, Εκτέλεση 3η

Πίνακας 4.12: Μετρικές δοκιμής για το συνδυασμό ModNet1-Fashion MNIST.

ModNet1-FMNIST	Accuracy	Loss	Precision	Recall	F1-Score
Orthogonal	0.8921	0.3079	0.8920	0.8921	0.8919
RandomN	0.8901	0.3114	0.8902	0.8901	0.8899
RandomU	0.8908	0.3147	0.8908	0.8908	0.8905
GloN	0.8942	0.3065	0.8947	0.8942	0.8941
GloU	0.8921	0.3090	0.8932	0.8921	0.8923
HeN	0.8892	0.3145	0.8894	0.8892	0.8889
HeU	0.8894	0.3160	0.8896	0.8894	0.8893
LeN	0.8936	0.3076	0.8941	0.8936	0.8935
LeU	0.8946	0.3058	0.8945	0.8946	0.8943

4.4.3 Σύνολο Cifar10

Το ModNet1 δημιουργήθηκε στα πλαίσια της ευρύτερης ανάλυσης της βαθιάς μάθησης, και παρακάτω εκπαιδεύεται πάνω στα σύνολα Cifar. Οι προκλήσεις στη βαθιά μάθηση είναι μεγαλύτερες, οπότε οι μετρικές δε θα προσεγγίζουν βέλτιστα νούμερα, αλλά προσεγγίζουν αξιολογικά επίπεδα, ανάλογα και το μήκος της εκπαίδευσης. Πρώτα αξιοποιείται το Cifar10 όπου στις υποψήφιες δέκα κλάσεις το έργο εκπαίδευσης κρίνεται πιο απλό.



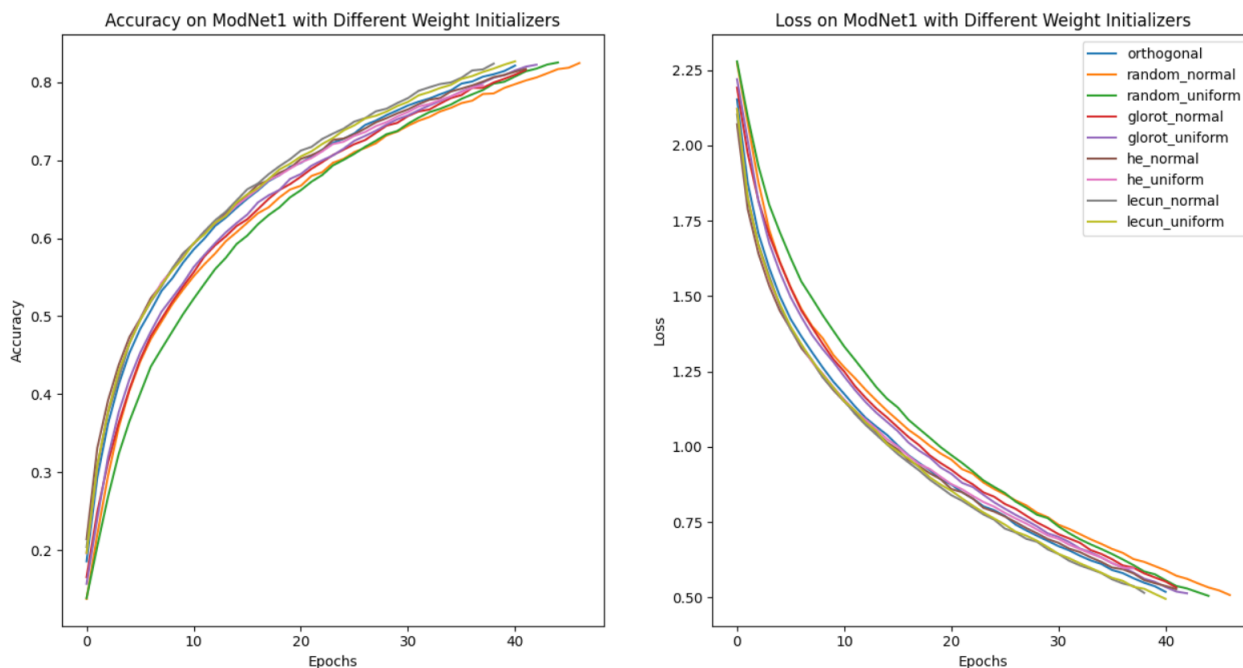
Σχήμα 4.6: ModNet1-FMNIST: Ακρίβεια και Απώλεια, Εκτέλεση 4η

Όπως φαίνεται και στο Σχήμα 4.7, οι RandomN και RandomU είναι αυτοί που σημείωσαν τη βαθύτερη εκπαίδευση, αγγίζοντας και τις 50 εποχές με ακρίβεια ταξι-νόμησης να αγγίζει το 84.34%, αλλά ο LeU προσπέρασε τους Random αλγορίθμους ελαφρώς, έχοντας τις καλύτερες επιδόσεις.

Πίνακας 4.13: Μετρικές εκπαίδευσης για το συνδυασμό ModNet1-Cifar10.

ModNet1-Cifar10	Accuracy	Loss	Precision	Recall	F1-Score
Orthogonal	0.8368	0.4772	0.8388	0.8368	0.8367
RandomN	0.8334	0.4843	0.8345	0.8334	0.8330
RandomU	0.8434	0.4581	0.8468	0.8434	0.8431
GloN	0.8315	0.4878	0.8339	0.8315	0.8311
GloU	0.8351	0.4824	0.8375	0.8351	0.8352
HeN	0.8247	0.5085	0.8272	0.8247	0.8242
HeU	0.8348	0.4802	0.8370	0.8348	0.8342
LeN	0.8307	0.4915	0.8319	0.8307	0.8307
LeU	0.8463	0.4548	0.8483	0.8463	0.8465

Στην αξιολόγηση με το σετ δοκιμής, τα αποτελέσματα παραμένουν σε καλό επίπεδο, με την ακρίβεια να κυμαίνεται μεταξύ 71.81% και 73.03%, γεγονός που σημαίνει ότι δεν παρεκκλίνουν πολύ από την εκπαίδευση και η υπερπροσαρμογή



Σχήμα 4.7: ModNet1-Cifar-10: Ακρίβεια και Απώλεια, Εκτέλεση 2η

τους στο σετ εκπαίδευσης αποτράπηκε. Ο LeN σημείωσε το ελάχιστο σφάλμα κατηγοριοποίησης, με τον LeU να παραμένει ο πιο αποδοτικός σε όλες τις υπόλοιπες μετρικές.

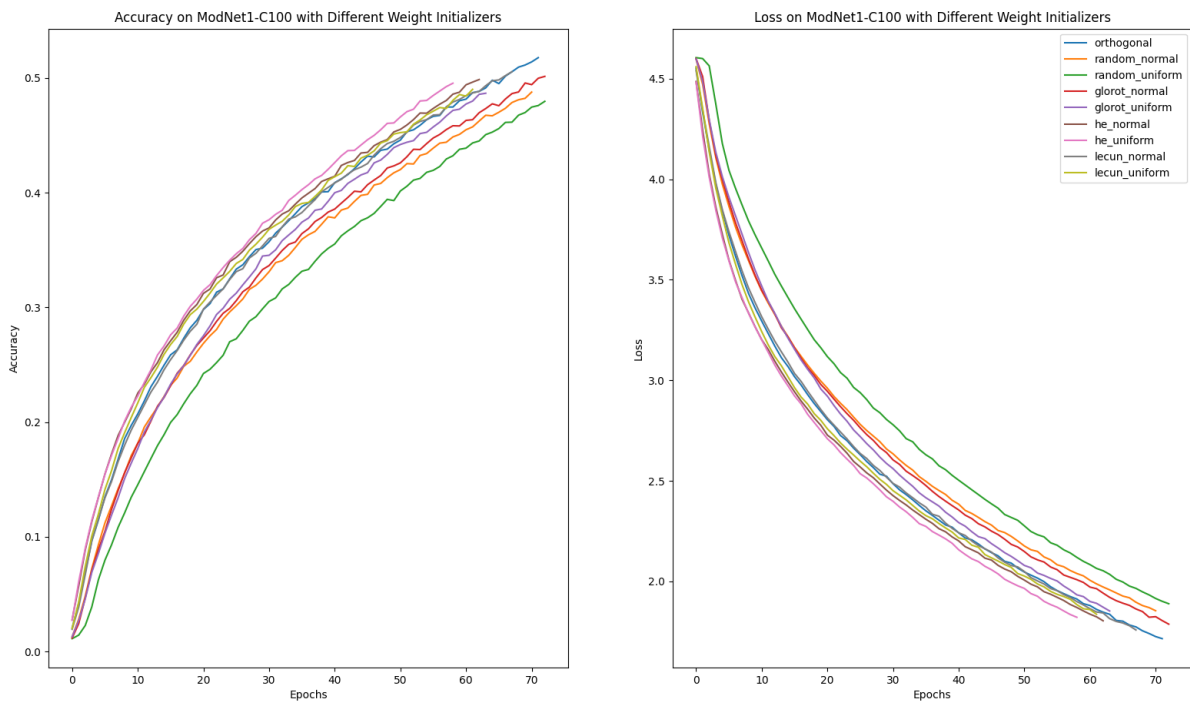
Πίνακας 4.14: Μετρικές δοκιμής για το συνδυασμό ModNet1-Cifar10.

ModNet1-Cifar10	Accuracy	Loss	Precision	Recall	F1-Score
Orthogonal	0.7239	0.8381	0.7262	0.7239	0.7235
RandomN	0.7235	0.8373	0.7242	0.7235	0.7213
RandomU	0.7258	0.8355	0.7292	0.7258	0.7254
GloN	0.7265	0.8244	0.7294	0.7265	0.7257
GloU	0.7219	0.8407	0.7239	0.7219	0.7217
HeN	0.7181	0.8411	0.7203	0.7181	0.7172
HeU	0.7255	0.8466	0.7245	0.7255	0.7215
LeN	0.7266	0.8160	0.7278	0.7266	0.7248
LeU	0.7303	0.8172	0.7327	0.7303	0.7305

4.4.4 Σύνολο Cifar100

Η περιπλοκότητα του ModNet1 περιορίζεται στα τρία στρώματα συνέλιξης, γεγονός που προμηνύει ότι για να κατηγοριοποιήσεις δείγματα σε 100 κλάσεις πλέον, θα απαιτείται σημαντική βελτιστοποίηση του δικτύου. Παρόλαυτα, το ModNet1 αν και φαίνεται πιο λογικό να εκπαιδευτεί με το Cifar10 σύνολο, παρακάτω εκπαιδευ-

εται στο Cifar100.



Σχήμα 4.8: ModNet1-Cifar100: Ακρίβεια και Απώλεια, Εκτέλεση 3η

Το πιο σύνηθες φαινόμενο (Σχήμα 4.8) ήταν οι Random και Glorot αλγόριθμοι να παρατείνουν την εκπαίδευση από ότι οι υπόλοιποι, σημειώνοντας μέχρι και 80 εποχές, ενώ σε κάποιες περιπτώσεις προεξέχει στην εκπαίδευση και ο Orthogonal. Στον Πίνακα εκπαίδευσης 4.15 αναδεικνύεται όμως ότι ο LeN και πάλι διατήρησε βέλτιστα νούμερα με ελάχιστο σφάλμα και υψηλότερη ακρίβεια στο 62.92%.

Πίνακας 4.15: Μετρικές εκπαίδευσης για το συνδυασμό ModNet1-Cifar100.

Mod1-Cifar100	Accuracy	Loss	Precision	Recall	F1-Score
Orthogonal	0.6175	1.4400	0.6224	0.6175	0.6144
RandomN	0.6096	1.4739	0.6145	0.6096	0.6053
RandomU	0.6036	1.4996	0.6069	0.6036	0.5996
GloN	0.6230	1.4164	0.6270	0.6230	0.6189
GloU	0.6145	1.4499	0.6186	0.6145	0.6103
HeN	0.6250	1.4113	0.6312	0.6250	0.6214
HeU	0.6086	1.4739	0.6128	0.6086	0.6045
LeN	0.6292	1.4049	0.6322	0.6292	0.6250
LeU	0.6255	1.4100	0.6284	0.6255	0.6215

Στο σετ δοκιμής (Πίνακας 4.16) είναι πασηφανές πως το δίκτυο δεν ανταπεξήλθε

επαρκώς, με το μέσο σφάλμα να κυμαίνεται στο 2.25, αλλά και πάλι ο LeU σημείωσε τις υψηλότερες δυνατές μετρικές.

Πίνακας 4.16: Μετρικές δοκιμής για το συνδυασμό ModNet1-Cifar100.

Mod1-Cifar100	Accuracy	Loss	Precision	Recall	F1-Score
Orthogonal	0.4237	2.2339	0.4274	0.4237	0.4199
RandomN	0.4158	2.2891	0.4160	0.4158	0.4100
RandomU	0.4127	2.3126	0.4117	0.4127	0.4072
GloN	0.4213	2.2587	0.4215	0.4214	0.4161
GloU	0.4254	2.2466	0.4282	0.4254	0.4214
HeN	0.4234	2.2492	0.4278	0.4234	0.4195
HeU	0.4236	2.2487	0.4237	0.4236	0.4180
LeN	0.4254	2.2426	0.4252	0.4254	0.4199
LeU	0.4282	2.2290	0.4294	0.4282	0.4237

4.5 Εκπαίδευση στο ModNet2

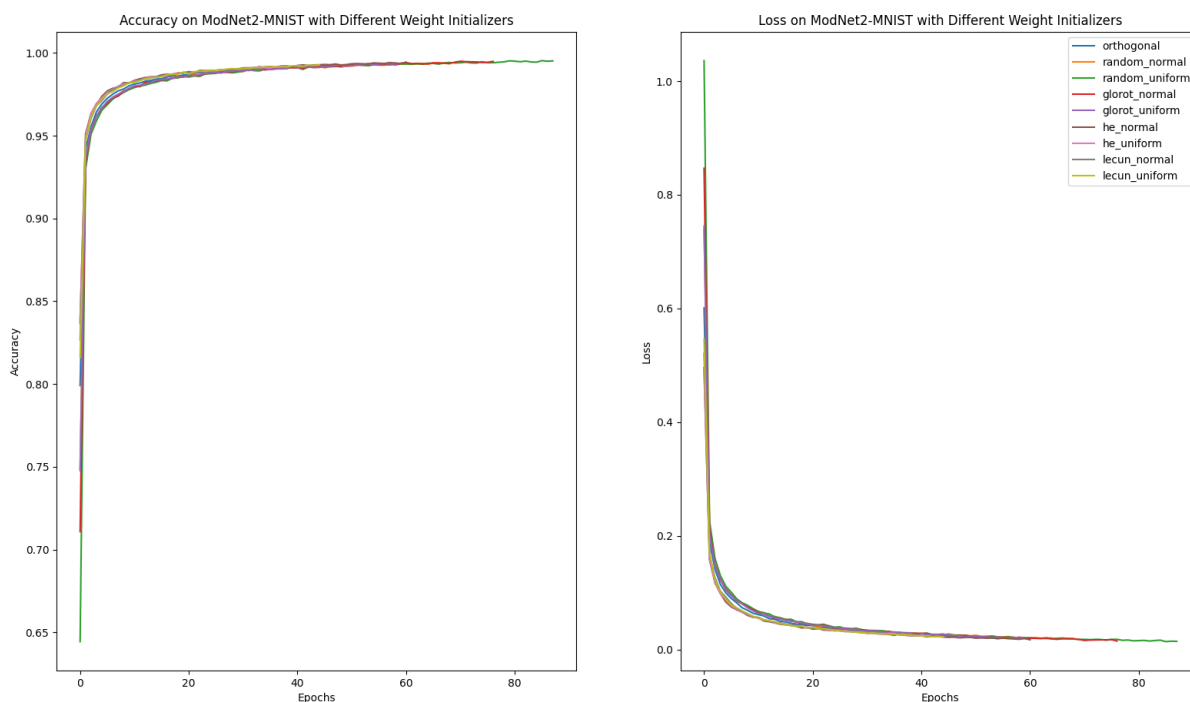
4.5.1 Σύνολο MNIST

Η εκτενής εκπαίδευση που επιτρέπει το ModNet2 με τις πολλές του παραμέτρους, και τα τέσσερα συνελκτικά του στρώματα, προμηγνύει ότι το συγκεκριμένο δίκτυο θα μπορέσει να ανταπεξέλθει πιο εύκολα στα σύνολα δεδομένων. Πρώτα δοκιμάστηκε το MNIST, με τα αποτελέσματα για τον εκάστοτε αλγόριθμο να είναι ιδιαίτερα ικανοποιητικά.

Πίνακας 4.17: Μετρικές εκπαίδευσης για το συνδυασμό ModNet2-MNIST.

ModNet2-MNIST	Accuracy	Loss	Precision	Recall	F1-Score
Orthogonal	0.9987	0.0053	0.9987	0.9987	0.9987
RandomN	0.9986	0.0051	0.9986	0.9986	0.9986
RandomU	0.9988	0.0049	0.9988	0.9988	0.9988
GloN	0.9987	0.0051	0.9987	0.9987	0.9987
GloU	0.9988	0.0048	0.9988	0.9988	0.9988
HeN	0.9981	0.0069	0.9981	0.9980	0.9981
HeU	0.9983	0.0060	0.9983	0.9983	0.9983
LeN	0.9977	0.0077	0.9977	0.9977	0.9977
LeU	0.9984	0.0059	0.9984	0.9984	0.9984

Και πάλι, ο RandomU επέτρεψε το δίκτυο να εκπαιδευτεί παραπάνω, με τον GloU να υπερτερεί στις μετρικές εκπαίδευσης και τον HeN στην δοκιμή.



Σχήμα 4.9: ModNet2-MNIST: Ακρίβεια και Απώλεια, Εκτέλεση 4η

Πίνακας 4.18: Μετρικές δοκιμής για το συνδυασμό ModNet2-MNIST.

ModNet2-MNIST	Accuracy	Loss	Precision	Recall	F1-Score
Orthogonal	0.9955	0.0138	0.9955	0.9955	0.9955
RandomN	0.9955	0.0137	0.9955	0.9954	0.9954
RandomU	0.9943	0.0155	0.9943	0.9942	0.9942
GloN	0.9948	0.0146	0.9948	0.9947	0.9947
GloU	0.9948	0.0149	0.9948	0.9947	0.9947
HeN	0.9958	0.0131	0.9958	0.9957	0.9958
HeU	0.9957	0.0129	0.9957	0.9957	0.9957
LeN	0.9956	0.0136	0.9956	0.9955	0.9955
LeU	0.9958	0.0131	0.9958	0.9957	0.9957

4.5.2 Σύνολο Fashion MNIST

Στο Fashion MNIST επανήλθε στο προσκήνιο ο Orthogonal με τα βέλτιστα νούμερα, που δεν εμφανιζόταν συχνά, ενώ τις 120 εποχές εκπαίδευσης συχνά άγγιζε η εκπαίδευση με τους HeN και HeU.

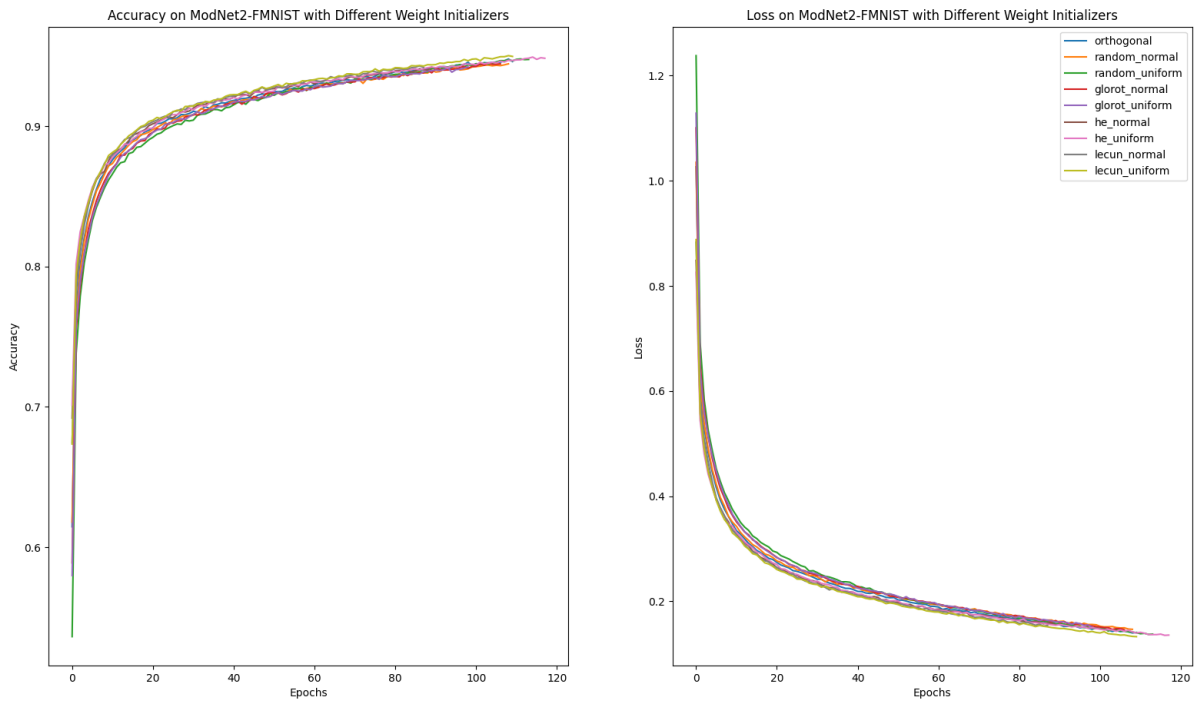
Στο τεστ με το σύνολο δοκιμής, νικητής είναι ο LeN.

Πίνακας 4.19: Μετρικές εκπαίδευσης για το συνδυασμό ModNet2-Fashion MNIST.

ModNet2-Fashion MNIST	Accuracy	Loss	Precision	Recall	F1-Score
Orthogonal	0.9699	0.0921	0.9699	0.9699	0.9699
RandomN	0.9682	0.0960	0.9682	0.9682	0.9681
RandomU	0.9685	0.0956	0.9685	0.9685	0.9684
GloN	0.9696	0.0922	0.9696	0.9696	0.9695
GloU	0.9689	0.0956	0.9688	0.9689	0.9688
HeN	0.9653	0.1001	0.9653	0.9653	0.9652
HeU	0.9660	0.0988	0.9659	0.9659	0.9659
LeN	0.9691	0.0929	0.9690	0.9691	0.9690
LeU	0.9697	0.0911	0.9696	0.9697	0.9696

Πίνακας 4.20: Μετρικές δοκιμής για το συνδυασμό ModNet2-Fashion MNIST.

ModNet2-Fashion MNIST	Accuracy	Loss	Precision	Recall	F1-Score
Orthogonal	0.9358	0.1811	0.9359	0.9358	0.9357
RandomN	0.9356	0.1814	0.9356	0.9356	0.9356
RandomU	0.9341	0.1837	0.9341	0.9341	0.9340
GloN	0.9340	0.1840	0.9338	0.9340	0.9338
GloU	0.9343	0.1827	0.9342	0.9343	0.9342
HeN	0.9362	0.1787	0.9361	0.9362	0.9361
HeU	0.9367	0.1785	0.9366	0.9367	0.9365
LeN	0.9368	0.1759	0.9366	0.9368	0.9366
LeU	0.9366	0.1765	0.9366	0.9366	0.9365



Σχήμα 4.10: ModNet2-FMNIST: Ακρίβεια και Απώλεια, Εκτέλεση 2η

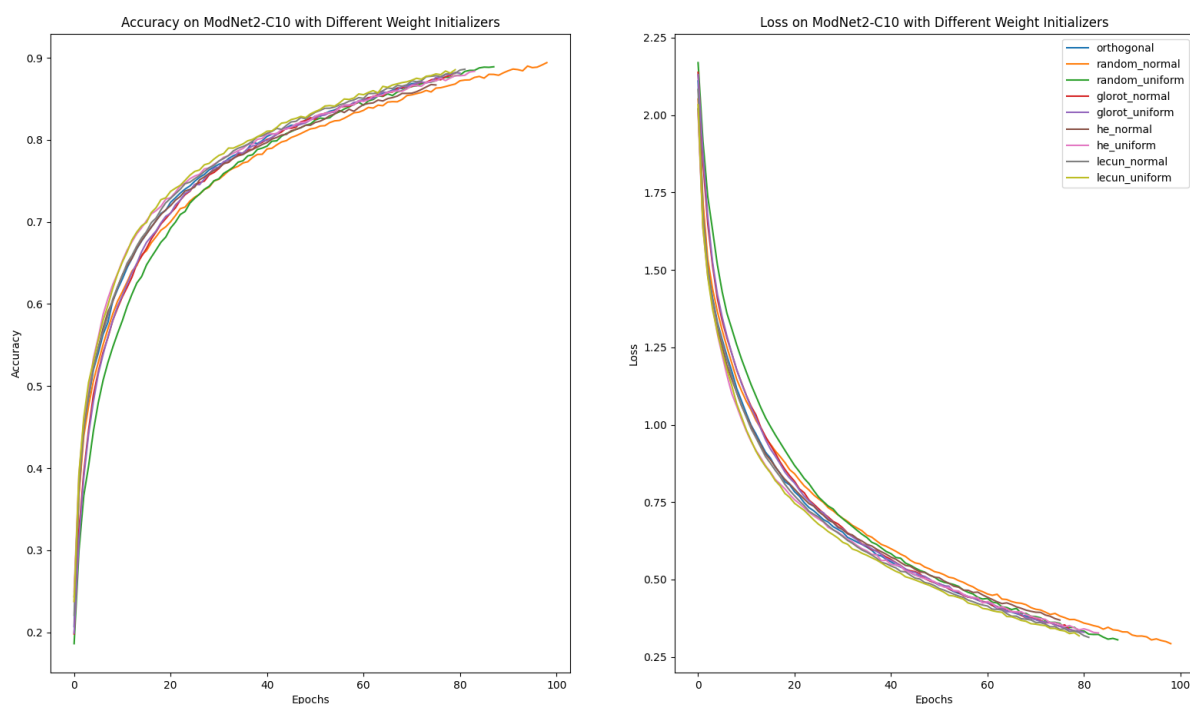
4.5.3 Σύνολο Cifar10

Η εκπαίδευση του ModNet2 στο Cifar10, ανέδειξε για άλλη μία φορά πως οι RandomN και RandomU μπόρεσαν να παρατείνουν την εκπαίδευση, με ένα αντικατοπτριστικό παράδειγμα στο Σχήμα 4.11.

Πίνακας 4.21: Μετρικές εκπαίδευσης για το συνδυασμό ModNet2-Cifar10.

ModNet2-Cifar10	Accuracy	Loss	Precision	Recall	F1-Score
Orthogonal	0.9710	0.1344	0.9712	0.9710	0.9709
RandomN	0.9761	0.1269	0.9763	0.9761	0.9761
RandomU	0.9762	0.1278	0.9762	0.9762	0.9761
GloN	0.9777	0.1198	0.9778	0.9777	0.9777
GloU	0.9693	0.1451	0.9695	0.9693	0.9692
HeN	0.9629	0.1542	0.9632	0.9629	0.9628
HeU	0.9657	0.1510	0.9658	0.9657	0.9656
LeN	0.9694	0.1408	0.9694	0.9694	0.9694
LeU	0.9696	0.1426	0.9698	0.9696	0.9695

Στο τεστ για το σύνολο εκπαίδευσης αποδοτικότερος κρίνεται ο GloN και για τη δοκιμή ο LeN.



Σχήμα 4.11: ModNet2-Cifar10: Ακρίβεια και Απώλεια, Εκτέλεση 4η

Πίνακας 4.22: Μετρικές δοκιμής για το συνδυασμό ModNet2-Cifar10.

ModNet2-Cifar10	Accuracy	Loss	Precision	Recall	F1-Score
Orthogonal	0.8382	0.4803	0.8383	0.8382	0.8373
RandomN	0.8325	0.5001	0.8326	0.8325	0.8316
RandomU	0.8332	0.4942	0.8325	0.8332	0.8322
GloN	0.8364	0.4862	0.8369	0.8364	0.8359
GloU	0.8351	0.4927	0.8349	0.8351	0.8341
HeN	0.8347	0.4878	0.8354	0.8347	0.8338
HeU	0.8363	0.4854	0.8359	0.8362	0.8351
LeN	0.8394	0.4764	0.8391	0.8394	0.8386
LeU	0.8366	0.4807	0.8361	0.8366	0.8354

4.5.4 Σύνολο Cifar100

Τα αποτελέσματα του πειράματος στο ModNet1 για το Cifar100 δεν κυμαίνονταν στις επιθυμητές τιμές, κάτι που σημαίνει πως η μάθηση στο ModNet2 πρέπει να είναι πιο εκτενής. Τα τέσσερα συνελικτικά στρώματα και η χρήση παραπάνω φίλτρων συντελούν μία βάση για να παραχθούν καλύτερες τιμές, όπως φαίνεται παρακάτω.

Στο σετ εκπαίδευσης, φαίνεται πως το δίκτυο εκπαιδεύτηκε σημαντικά, με τα σωστά κατηγοριοποιημένα δείγματα να κυμαίνονται μεταξύ 86.61% και 93.93%

Πίνακας 4.23: Μετρικές εκπαίδευσης για το συνδυασμό ModNet2-Cifar100.

ModNet2-Cifar100	Accuracy	Loss	Precision	Recall	F1-Score
Orthogonal	0.8871	0.5412	0.8906	0.8871	0.8868
RandomN	0.8998	0.5189	0.9031	0.8998	0.8997
RandomU	0.8880	0.5681	0.8916	0.8880	0.8879
GloN	0.9393	0.3849	0.9410	0.9393	0.9493
GloU	0.9165	0.4636	0.9195	0.9165	0.9165
HeN	0.8474	0.6643	0.8534	0.8474	0.8469
HeU	0.8661	0.5991	0.8713	0.8661	0.8660
LeN	0.8856	0.5477	0.8898	0.8856	0.8854
LeU	0.8654	0.6146	0.8700	0.8654	0.8650

ανα αλγόριθμο, με τον GloN να σημειώνει με διαφορά τις καλύτερες τιμές σε σύγκριση με τους υπόλοιπους. Στην εκπαίδευση (Σχήμα 4.12) φαίνεται πως ο GloN υπερτερεί, με τους Random αλγορίθμους να ακολουθούν, κάτι που αποτυπώνεται και στα αποτελέσματα.

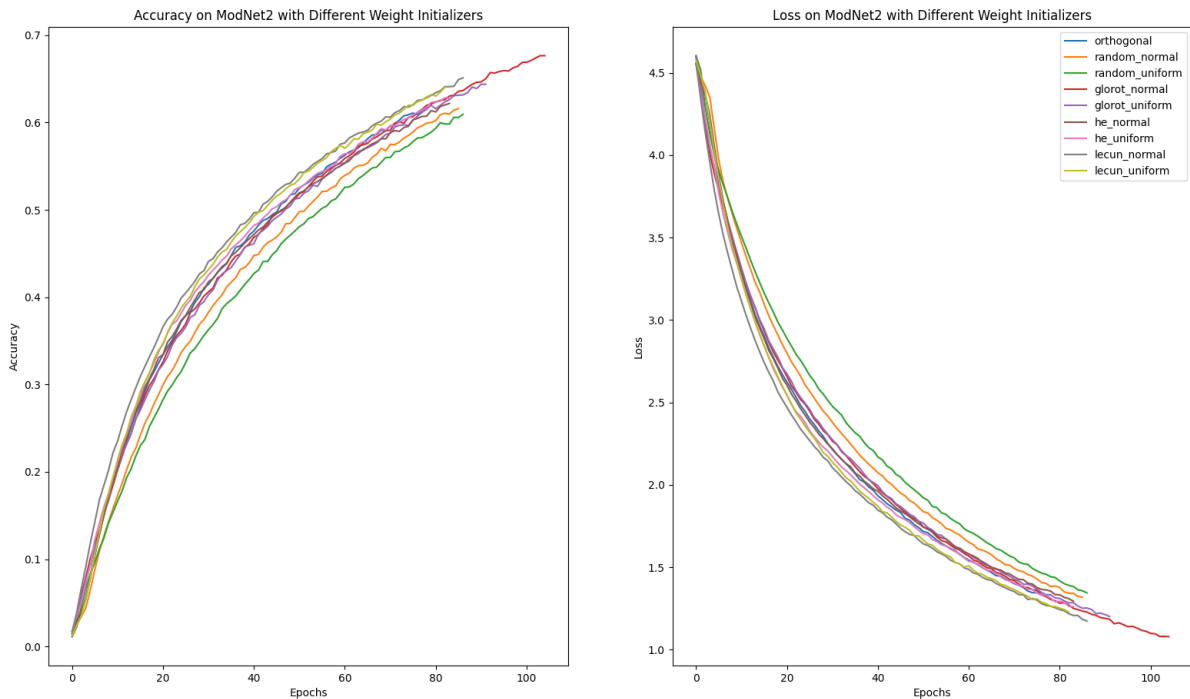
Πίνακας 4.24: Μετρικές δοκιμής για το συνδυασμό ModNet2-Cifar100.

ModNet2-Cifar100	Accuracy	Loss	Precision	Recall	F1-Score
Orthogonal	0.5476	1.6962	0.5503	0.5476	0.5434
RandomN	0.5342	1.7560	0.5385	0.5342	0.5308
RandomU	0.5231	1.8280	0.5243	0.5231	0.5183
GloN	0.5442	1.7327	0.5477	0.5442	0.5409
GloU	0.5397	1.7339	0.5423	0.5397	0.5354
HeN	0.5438	1.6966	0.5470	0.5438	0.5394
HeU	0.5465	1.6898	0.5522	0.5465	0.5428
LeN	0.5544	1.6689	0.5593	0.5544	0.5508
LeU	0.5514	1.6730	0.5548	0.5514	0.5473

Οι μετρικές στο σετ δοκιμής είναι σημαντικά χαμηλότερες από της εκπαίδευσης, που αναδεικνύει υπερπροσαρμογή στο σετ εκπαίδευσης, αλλά οι τιμές τους είναι βελτιωμένες σε σχέση με το συνδυασμό ModNet1-Cifar10, με αποδοτικότερο αλγόριθμο να είναι ο LeN.

4.6 Βελτιστοποίηση υπερπαραμέτρων

Οι μετρικές που εξάχθηκαν από τα παραπάνω πειράματα ποικίλουν, καθώς οι ανάγκες του κάθε συνδυασμού μοντέλο-σύνολο ήταν διαφορετικές και η αρχιτεκτονική του μοντέλου μαζί την επιλογή υπερπαραμέτρων διαφέρουν ανα πείραμα. Ωστόσο, οι μετρικές που σημείωσε το καθένα από τα τρία μοντέλα είχαν χαμη-



Σχήμα 4.12: ModNet2-Cifar100: Ακρίβεια και Απώλεια, Εκτέλεση 5η

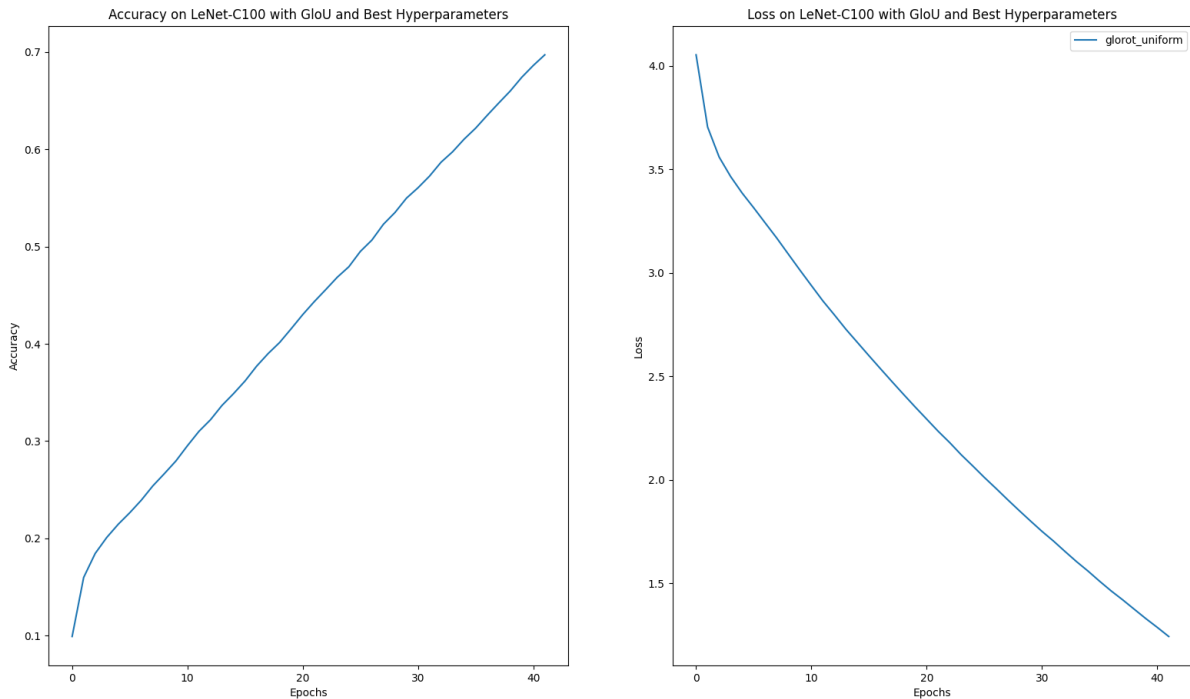
λές τιμές όταν εκπαιδεύτηκαν με το σύνολο Cifar100, κάτι το αναμενόμενο βέβαια καθώς το συγκεκριμένο σύνολο, όπως προαναφέρθηκε, κρίνεται το πιο απαιτητικό όσο αφορά την κατηγοριοποίηση, διότι διαχωρίζεται σε 100 κλάσεις εικόνων (οι περισσότερες σε σύγκριση με τα υπόλοιπα σύνολα) οι οποίες είναι και έγχρωμες.

Στην ενότητα αυτή, επιλέγονται οι αλγόριθμοι αρχικοποίησης με τους οποίους το μοντέλο σημείωσε τις καλύτερες μετρικές για καθένα από τα τρία πειράματα, και το πείραμα επαναλαμβάνεται για αυτούς τους αλγόριθμους αρχικοποίησης, με τη διαφορά ότι πρώτα το κάθε μοντέλο θα βελτιστοποιηθεί πάνω σε αυτόν τον αλγόριθμο. Το Optuna, ένα λογισμικό βελτιστοποίησης μοντέλων μηχανικής μάθησης, εφαρμόστηκε στα μοντέλα με σκοπό να επιλέξει καταλληλότερες υπερπαραμέτρους για το κάθε πείραμα.

Οι υπερπαραμέτροι που επιλέχθηκαν για να βελτιστοποιηθούν είναι ο αριθμός φίλτρων που εξάγεται μετά από κάθε συνελικτικό στρώμα και ο ρυθμός εκμάθησης της εκπαίδευσης, παράμετροι που στα αρχικά πειράματα για την επιλογή μοντέλων φαίνεται πως επηρέαζαν σημαντικά τη λειτουργία του δικτύου.

4.6.1 Βελτιστοποίηση υπερπαραμέτρων για το LeNet

Το μοντέλο LeNet είναι αυτό που εξήγαγε τα λιγότερα φίλτρα στα στρώματα του, οπότε κρίνεται αναγκαίο κατά τη βελτιστοποίηση του να προταθούν αρκετά παραπάνω φίλτρα για καλύτερα αποτελέσματα. Πλέον εξάγει 96 και 128 χαρακτηριστικά στη σειρά, και ο ρυθμός εκμάθησης ορίστηκε στο 0.000111.



Σχήμα 4.13: Βελτιστοποιημένο LeNet-Cifar100:Ακρίβεια και Απώλεια, Εκτέλεση 5η

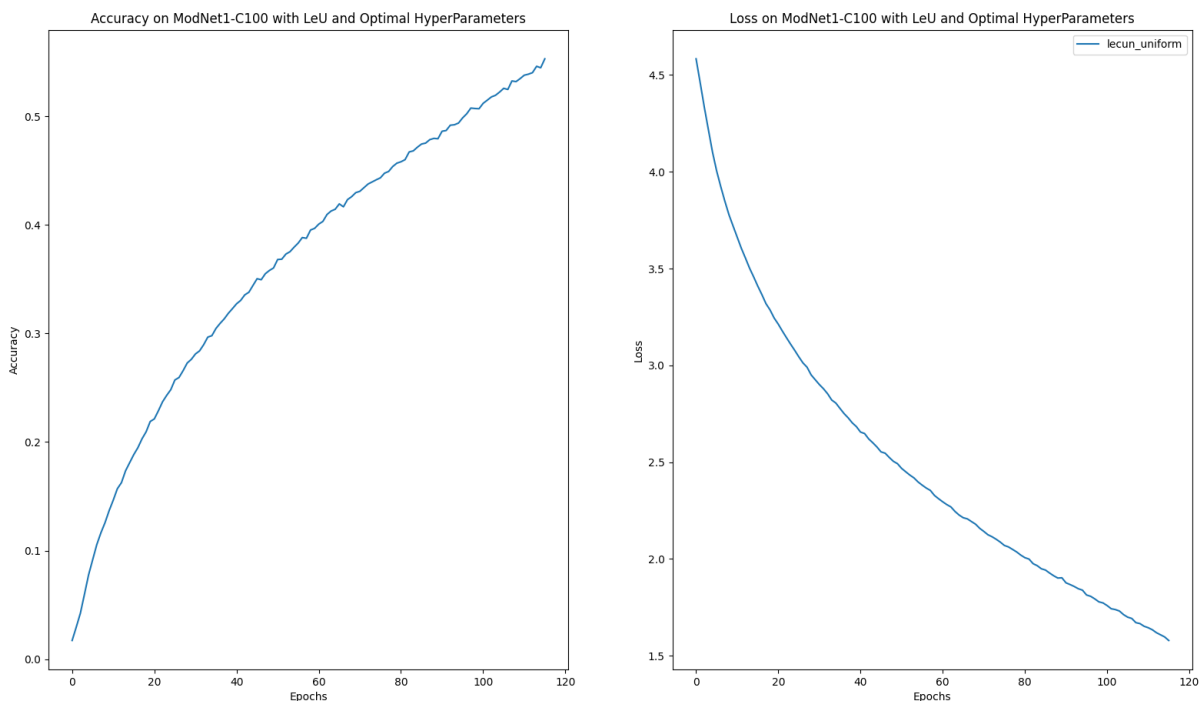
Το δίκτυο εκπαιδεύτηκε σημαντικά παραπάνω με τον GloU, και οι μετρικές είναι εξαιρετικά βελτιωμένες στο σετ εκπαίδευσης, σχεδόν αγγίζοντας το 60%, με τις μετρικές στο σετ δοκιμής να μην ικανοποιούν αλλά να είναι επίσης βελτιωμένες περίπου 10% σε σύγκριση με τις προηγούμενες μετρικές δοκιμής.

Πίνακας 4.25: Μετρικές εκπαίδευσης και δοκιμής για το συνδυασμό LeNet-Cifar100 με τον αρχικοποιητή GloU, μετά από βελτιστοποίηση υπερπαραμέτρων.

LeNet-Cifar100	Accuracy	Loss	Precision	Recall	F1-Score
Train	0.3503	2.6180	0.3533	0.3503	0.33915
Optimized Train	0.5983	1.6258	0.5992	0.5983	0.5933
Test	0.2556	3.1636	0.2543	0.2556	0.2453
Optimized Test	0.3423	2.7238	0.3368	0.3423	0.3354

4.6.2 Βελτιστοποίηση υπερπαραμέτρων για το ModNet1

Ο αρχικοποιητής με τον οποίο το μοντέλο εμφάνισε τις καλύτερες μετρικές ήταν ο LeCun Uniform, όταν το μοντέλο εκπαιδεύτηκε με το Cifar100. Μετά απο 10 δοκιμές, προτάθηκαν τα φίλτρα 64, 128 και 160 για τα τρία συνελικτικά στρώματα αντίστοιχα, ενώ ο ρυθμός εκμάθησης μειώθηκε στο 0.000426. Όπως φαίνεται (Σχήμα 4.14) η εκπαίδευση διήρκησε 110 εποχές κατά την πέμπτη εκτέλεση, πάνω απο 40 εποχές παραπάνω σε σχέση με το μη βελτιστοποιημένο δίκτυο.



Σχήμα 4.14: Βελτιστοποιημένο ModNet1-Cifar100: Ακρίβεια και Απώλεια. Εκτέλεση 5η

Τόσο οι μετρικές εκπαίδευσης όσο και αυτές της δοκιμής ήταν βελτιωμένες, αλλά όχι σε μεγάλο βαθμό, καθώς η ακρίβεια ταξινόμησης βελτιώθηκε μονάχα κατά 0.8% στο σετ δοκιμής.

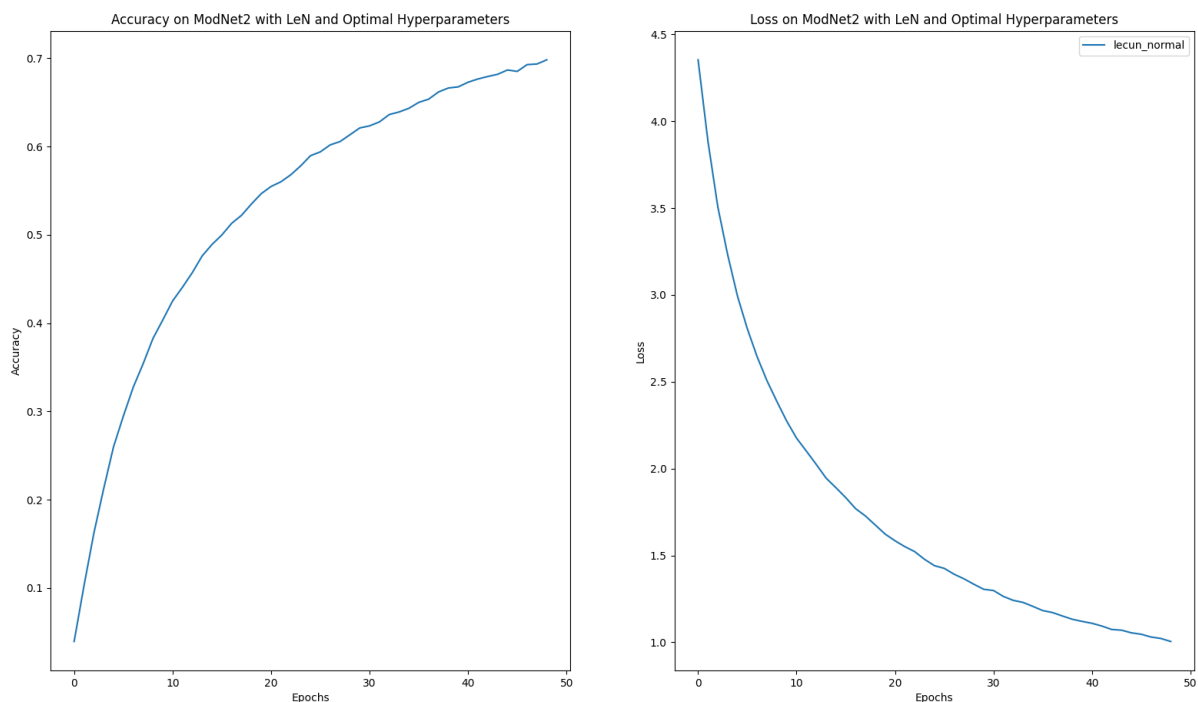
4.6.3 Βελτιστοποίηση υπερπαραμέτρων για το ModNet2

Τέλος ακολουθεί το ModNet2, όπου μετά απο 10 δοκιμές επιλέχθηκε για το δίκτυο διαδοχικά 128, 128, 256 και 192 φίλτρα. Ο LeN ήταν ο αποδοτικότερος για το δίκτυο αλγόριθμος στο αρχικό πείραμα, και αυτή τη φορά ο ρυθμός εκμάθησης

Πίνακας 4.26: Μετρικές εκπαίδευσης και δοκιμής για το συνδυασμό ModNet1-Cifar100 με τον αρχικοποιητή LeU, μετά απο βελτιστοποίηση υπερπαραμέτρων.

ModNet1-Cifar100	Accuracy	Loss	Precision	Recall	F1-Score
Train	0.6255	1.4100	0.6284	0.6255	0.6215
Optimized Train	0.6562	1.3132	0.6589	0.6562	0.6533
Test	0.4282	2.2290	0.4294	0.4282	0.4237
Optimized Test	0.4362	2.1843	0.4385	0.4362	0.4329

ορίστηκε στο 0.0037, σχεδόν τετραπλάσιος δηλαδή απο τον αρχικό.



Σχήμα 4.15: Βελτιστοποιημένο ModNet2-Cifar100: Ακρίβεια και Απώλεια, Εκτέλεση 4η

Απο το Σχήμα 4.15 αποτυπώνεται η επιλογή μεγαλύτερου ρυθμού εκμάθησης, καθώς στην 4η εκτέλεση το δίκτυο πέτυχε πάνω απο 90% ακρίβεια σε 49 εποχές, κάτι που για να πετύχει με το αρχικό πείραμα ήθελε πάνω απο 100. Στο σετ δοκιμής παρόλαυτα (Πίνακας 4.27) η βελτιστοποίηση δεν είχε κάποιο ιδιαίτερο αποτέλεσμα, καθώς οι μετρικές κυμαίνονται σε σχεδόν πανομοιότυπες τιμές.

Πίνακας 4.27: Μετρικές εκπαίδευσης και δοκιμής για το συνδυασμό ModNet2-Cifar100 με τον αρχικοποιητή LeN, μετά απο βελτιστοποίηση υπερπαραμέτρων.

ModNet2-Cifar100	Accuracy	Loss	Precision	Recall	F1-Score
Train	0.8856	0.5477	0.8898	0.8856	0.8854
Optimized Train	0.9100	0.4572	0.9143	0.9100	0.9103
Test	0.5544	1.6689	0.5593	0.5544	0.5508
Optimized Test	0.5541	1.6759	0.5634	0.5541	0.5521

4.7 Πορίσματα υπολογιστικής μελέτης

Τα παραπάνω πειράματα διεξάχθηκαν στα πλαίσια της μελέτης στην επιρροή την οποία έχουν οι αλγόριθμοι αρχικοποίησης βαρών, πάνω σε διάφορα συνελικτικά νευρωνικά μοντέλα. Μετά την επιτυχή ολοκλήρωση αυτών, μπορούν να διατυπωθούν διάφορα συμπεράσματα που αφορούν τη βαθιά μάθηση δικτύων με αρχικοποιητές βαρών, τόσο για την εκπαίδευση τους όσο και για τη δοκιμή τους σε άλλα σύνολα δεδομένων.

Με το πέρας δώδεκα διαφορετικών πειραμάτων, με διαφορετικές ανάγκες και ιδιαιτερότητες, η πλειοψηφία αυτών ανέδειξε πως οι αλγόριθμοι με τους οποίους το δίκτυο εκπαιδεύτηκε εκτενέστερα ήταν οι Random και Glorot. Σε θέμα απόδοσης, θεωρείται περιττό να αναλογιστούμε πόσο πολύ εκπαιδεύεται ένα δίκτυο, διότι αυτό που προέχει είναι οι καλές μετρικές, αλλά στα πλαίσια του πειράματος, και γενικά στα πλαίσια της μηχανικής μάθησης στα νευρωνικά δίκτυα, είναι σημαντικό κριτήριο, καθώς προμηνύει ότι με σωστή βελτιστοποίηση του δικτύου, οι συγκεκριμένοι αλγόριθμοι μπορούν να βοηθήσουν το δίκτυο να επιτύχει τις βέλτιστες μετρικές, σε σύγκριση με τους υπόλοιπους αλγόριθμους.

Όσο αφορά την απόδοση βέβαια, παρακάτω αναλύονται κάποιες μετρικές στα πειράματα πάνω στο σύνολο δοκιμής, που αναδεικνύουν τους επικρατέστερους αλγόριθμους προς επιλογή. Οι μετρικές που θα συζητηθούν είναι το Accuracy, η πιο σημαντική εκ των μετρικών καθώς δείχνει ποιο ποσοστό των δειγμάτων ταξινομήθηκε σωστά, και το F1-Score το οποίο δίνει μία καθολική εικόνα και για τις υπόλοιπες μετρικές (Precision, Recall) και συνοψίζει αν οι προβλέψεις που εξάχθηκαν συμπεριλάμβαναν πολλά False Positives ή False Negatives (Ενότητα 3.2).

Το μοντέλο LeNet (Πίνακας 4.28), πέτυχε τις καλύτερες μετρικές κυρίως με τους αρχικοποιητές He, ενώ το ModNet1 (Πίνακας 4.29) απέδιδε καλύτερα σχεδόν

Πίνακας 4.28: Υψηλότερες μετρικές δοκιμής (Accuracy και F1-Score) για το μοντέλο LeNet.

MNIST	Accuracy	F1-Score
HeU	0.9891	0.9890
Orthogonal	0.9888	0.9887
Fashion MNIST		
HeU	0.8899	0.8899
HeN	0.8882	0.8881
Cifar10		
HeN	0.5516	0.5473
Orthogonal	0.5502	0.5480
Cifar100		
GloU	0.2556	0.2453
Orthogonal	0.2550	0.2452

αποκλειστικά με τους αρχικοποιητές LeCun.

Πίνακας 4.29: Υψηλότερες μετρικές δοκιμής (Accuracy και F1-Score) για το μοντέλο ModNet1.

MNIST	Accuracy	F1-Score
LeU	0.9887	0.9886
LeN	0.9880	0.9880
Fashion MNIST		
LeU	0.8946	0.8943
GloN	0.8942	0.8941
Cifar10		
LeU	0.7303	0.7305
LeN	0.7266	0.7248
Cifar100		
LeU	0.4282	0.4237
LeN	0.4254	0.4199

Τα πειράματα συνοψίζονται στο 3ο μοντέλο, με τις μετρικές (Πίνακας 4.30) να ανακηρύσσουν επικρατέστερους αλγόριθμους τους He και LeCun για άλλη μία φορά, ενώ σε γενικές γραμμές οι πιο αποδοτικοί μέθοδοι είναι οι LeCun. Τόσο οι αλγόριθμοι LeCun όσο και οι He, ορίζουν πως η επιλογή βαρών θα λειτουργεί βάση τον αριθμό των μονάδων εισόδου, δηλαδή των χαρακτηριστικών φίλτρων που επιδέχεται η κάθε συνέλιξη, σε αντίθεση με τις μεθόδους Glorot που συνυπολογίζουν και τον αριθμό των χαρακτηριστικών που θα εξαχθούν στην έξοδο της συνέλιξης, γεγονός που πιθανότατα επηρέασε τις επιδόσεις αυτές.

Τέλος, εκτελέστηκαν 3 ακόμα πειράματα στα πλαίσια της βελτιστοποίησης των

Πίνακας 4.30: Υψηλότερες μετρικές δοκιμής (Accuracy και F1-Score) για το μοντέλο ModNet2.

MNIST	Accuracy	F1-Score
HeN	0.9958	0.9958
LeU	0.9958	0.9957
Fashion MNIST		
LeN	0.9368	0.9366
HeU	0.9367	0.9365
Cifar10		
LeN	0.8394	0.8386
LeU	0.8366	0.8354
Cifar100		
LeN	0.5544	0.5508
LeU	0.5514	0.5473

ήδη υπάρχοντων δικτύων, με τους αλγόριθμους που απέδωσαν καλύτερα. Οι αποδόσεις των δικτύων βελτιώθηκαν και στις 3 περιπτώσεις, όχι σε μεγάλο βαθμό, αλλά σε βαθμό που δείχνει ότι με παραπάνω δοκιμές βελτιστοποίησης, και διαφορετικές επιλογές υπερπαραμέτρων, θα μπορούσε να βελτιωθεί ακόμα περισσότερο το δίκτυο.

Κεφάλαιο 5

Συμπεράσματα

Η συγκεκριμένη διπλωματική αναλύει κάποιες πτυχές της βαθιάς μάθησης σε συνελκτικά νευρωνικά δίκτυα και πιο συγκεκριμένα επιδιώκει να αναδείξει την απόδοση ορισμένων μοντέλων στην ταξινόμηση εικόνων, όταν χρησιμοποιούν γνωστούς αλγόριθμους αρχικοποίησης βαρών. Στα πλαίσια της βελτιστοποίησης των υπερπαραμέτρων ενός νευρωνικού δικτύου, η επιλογή βάρους αποτελεί ένα μείζον ζήτημα, το οποίο οι ερευνητές φροντίζουν να διευθετήσουν εκ των προτέρων.

Πρωταρχικό μέλημα ήταν να μελετηθούν διεξοδικά οι παράμετροι που συντελούν ένα ορθολογικό συνελκτικό δίκτυο, το οποίο επιτελεί κατηγοριοποίηση πολλαπλών κλάσεων σε μία σειρά απο εικόνες. Έτσι, δημιουργήθηκαν δύο μοντέλα βαθιάς μάθησης (ModNet1, ModNet2) και αξιοποιήθηκε άλλο ένα υπάρχων μοντέλο (LeNet), τα οποία μοντέλα υποβλήθηκαν, το καθένα ξεχωριστά, σε μία σειρά πειραμάτων πάνω σε τέσσερα διαφορετικά σύνολα δεδομένων. Μετά από κάθε πείραμα κατηγοριοποίησης, εξάχθηκαν μετρικές ανάλογα με τον αρχικοποιητή που χρησιμοποίησε σε κάθε επανάληψη το δίκτυο, για να αποφανθεί ποιός ή ποιοί αλγόριθμοι καθιστούν το μοντέλο πιο αποδοτικό. Πέρα απο την απόδοση ενός δικτύου, με το πέρας της μελέτης δημιουργήθηκαν και γενικά πορίσματα που αφορούν την εκπαίδευση μοντέλων βαθιάς μάθησης.

Όσο αφορά τα εν λόγω πειράματα, το κάθε μοντέλο εκπαιδεύτηκε πάνω σε τέσσερα διαφορετικά σύνολα δεδομένων (MNIST, Fashion MNIST, Cifar10, Cifar100) ασπρόμαυρων και έγχρωμων εικόνων, επιδιώκοντας πολυκατηγορική ταξινόμηση μεταξύ 10 ή 100 κλάσεων για το κάθε σύνολο. Το μοντέλο εκπαιδεύτηκε πέντε ξεχωριστές φορές στο εκάστοτε σύνολο και εξάχθηκαν μετρικές για τον μέσο όρο των πέντε εκτελέσεων, καθώς στην εκπαίδευση επιλέγεται διαφορετική δεσμίδα δεδο-

μένων κάθε φορά, οπότε πρέπει να αποτυπωθεί μία πιο σφαιρική εικόνα. Ακόμη, αφού ολοκληρώθηκαν τα βασικά πειράματα κατηγοριοποίησης, στο κάθε μοντέλο αναθεωρήθηκαν κάποιοι υπερπαραμέτροι, σύμφωνα με τις προτάσεις του λογισμικού Optuna, και ξαναδοκιμάστηκαν πάνω στο πιο απαιτητικό εκ των τεσσάρων συνόλων Cifar100.

Σχετικά με τα 12 αρχικά πειράματα, δηλαδή τρία μοντέλα εκπαίδευσης πάνω σε τέσσερα ξεχωριστά σύνολα δεδομένων, τα πορίσματα ποικίλουν:

- Στη συντριπτική πλειοψηφία των πειραμάτων, τα μοντέλα σημείωσαν τις καλύτερες μετρικές όταν χρησιμοποιούσαν τους αρχικοποιητές βαρών LeCun, με τους He να είναι οι πιο συχνοί επιλαχώντες. Οι υπόλοιποι αλγόριθμοι κανονικής η ομοιόμορφης κατανομής (Glorot, Random και Orthogonal) ήταν ανταγωνιστικοί στην εκπαίδευση και προσέγγιζαν συχνά την απόδοση των επικρατέστερων, αλλά οι μέθοδοι LeN και LeU σχεδόν πάντα βελτιώναν τις αποδόσεις του εκάστοτε μοντέλου περισσότερο από τους υπολοίπους.
- Στην εκπαίδευση, ιδιαίτερα σύνηθες φαινόμενο ήταν τά μοντέλα που αξιοποιούσαν τους αλγόριθμους RandomN και RandomU να υποβάλλονταν σε παρατεταμένη, κατά αρκετές εποχές, εκπαίδευση. Στον μεγαλύτερο βαθμό, ο ρυθμός εκμάθησης είναι ο παράγοντας που καθορίζει τη διάρκεια της εκπαίδευσης, αλλά σε όλα τα βασικά πειράματα ήταν ίδιος, οπότε είναι αξιωματικό να ξεχωρίσουμε ποιού αλγόριθμοι επιτρέπουν την παραπάνω εκπαίδευση, διότι μπορεί να σημαίνει πως οι Random αλγόριθμοι θα δημιουργούσαν μοντέλα με καλύτερες μετρικές, μετά από σωστή βελτιστοποίηση.
- Ο συνδυασμός μοντέλου-συνόλου δεδομένων μπορεί να αποτελεί καθοριστικός παράγοντας στην εξαγωγή μετρικών απόδοσης, ανεξάρτητα της επιλογής αλγορίθμου αρχικοποίησης βαρών, κάτι που παρατηρήθηκε στο σύνολο Cifar100. Πιο συγκεκριμένα, αποφάνθηκε ότι οι μετρικές και των 3 μοντέλων όταν εκπαιδεύτηκαν με το Cifar100 δεν ήταν πάντα ικανοποιητικές, και η επιλογή οποιουδήποτε εκ των αλγορίθμων δεν αποτέλεσε ανασταλτικό παράγοντα για να αποτραπεί αυτό. Ως εκ τούτου, κρίνεται αναγκαία η βελτιστοποίηση των υπερπαραμέτρων του δικτύου, και γενικότερα η αναθεώρηση άλλων παραμέτρων, πέρα από την αρχικοποίηση βαρών.

Στα πλαίσια της εργασίας και τη χρήση αρχικοποιητών βάρους, το παραπάνω πόρισμα μετουσιώθηκε σε τρία ακόμα πειράματα, τα οποία, όπως προαναφέρθηκε, αφορούν τη βελτιστοποίηση των τριών μοντέλων, όταν αυτά χρησιμοποιούν τον αλγόριθμο που συνέβαλε στις υψηλότερες μετρικές. Τα αποτελέσματα δείχνουν βελτίωση των μετρικών και στις τρεις περιπτώσεις, ενώ σε μία εξ αυτών είχε αξιοποιηθεί ο Glorot Uniform, ο οποίος μετά τους Random, οδηγούσε το μοντέλο σε παραπάνω εκπαίδευση. Στο εν λόγω πείραμα, οι μετρικές στο σύνολο εκπαίδευσης βελτιώθηκαν κατά 25%, γεγονός που μερικώς επιβεβαιώνει την προηγούμενη θέση, τη θέση δηλαδή που εξισώνει αποδοτικά του υπόλοιπους αλγορίθμους με τους LeN και LeU, αν επιτευχθεί σωστή βελτιστοποίηση.

Συνοψίζοντας, η εν λόγω διπλωματική αναδεικνύει πως οι αρχικοποιητές βαρών LeCun είθισται να είναι οι πιο συνεπείς αποδοτικά, όταν χρησιμοποιούνται σε συνελκτικά δίκτυα κατηγοριοποίησης εικόνων μεταξύ πολλαπλών κλάσεων. Ακόμα, τονίζεται πως μελλοντικές επεκτάσεις παρόμοιες με αυτήν, μπορούν να φέρουν στην επιφάνεια και άλλους αλγορίθμους προς επιλογή, όπως οι Random και Glorot, καθώς φαίνεται πως η εκτενής βελτιστοποίηση υπερπαραμέτρων ενός μοντέλου μπορεί να ανακατάξει τις μετρικές απόδοσης, και να κρίνει τους Random ή Glorot ως καταλληλότερους αρχικοποιητές βαρών. Εφόσον τα πειραματικά πλαίσια κυμαίνονταν σε ορισμένες δοκιμές και σε ορισμένο εύρος εποχών εκπαίδευσης, είναι πιθανό η αξιοποίηση αυτών των αλγορίθμων να εξήγαγε μετρικές απόδοσης που να έφταναν ή και να ξεπερνούσαν τις αντίστοιχες που σημειώθηκαν με τους αλγορίθμους αρχικοποίησης LeCun.

Βιβλιογραφία

- [1] Charu C Aggarwal. *Neural networks and deep learning*, volume 10. Springer, 2018.
- [2] Arohan Ajit, Koustav Acharya, and Abhishek Samanta. A review of convolutional neural networks. In *2020 international conference on emerging trends in information technology and engineering (ic-ETITE)*, pages 1–5. IEEE, 2020.
- [3] Neena Aloysius and M Geetha. A review on deep convolutional neural networks. In *2017 International Conference on Communication and Signal Processing (ICCSP)*, pages 0588–0592. IEEE, 2017.
- [4] Happiness Ugochi Dike, Yimin Zhou, Kranthi Kumar Deveerasetty, and Qingtian Wu. Unsupervised learning based on artificial neural network: A review. In *2018 IEEE International Conference on Cyborg and Bionic Systems (CBS)*, pages 322–327. IEEE, 2018.
- [5] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.
- [6] Simon Haykin. *Neural networks and learning machines*, 3/E. Pearson Education India, 2009.
- [7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.
- [8] Wei Hu, Lechao Xiao, and Jeffrey Pennington. Provable benefit of orthogonal initialization in optimizing deep linear networks. *arXiv preprint arXiv:2001.05992*, 2020.
- [9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.
- [10] Puneet Kumar, Shalini Batra, and Balasubramanian Raman. Deep neural network hyperparameter tuning through twofold genetic approach. *Soft Computing*, 25:8747–8771, 2021.
- [11] Trupti A Kumbhare and Santosh V Chobe. An overview of association rule mining algorithms. *International Journal of Computer Science and Information Technologies*, 5(1):927–930, 2014.
- [12] Erik G Learned-Miller. Introduction to supervised learning. *I: Department of Computer Science, University of Massachusetts*, page 3, 2014.

-
- [13] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.
- [14] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. volume 86, pages 2278–2324. Ieee, 1998.
- [15] Yann LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–50. Springer, 2002.
- [16] Yann LeCun, John Denker, and Sara Solla. Optimal brain damage. *Advances in Neural Information Processing Systems*, 2, 1989.
- [17] Huimin Li, Marina Krček, and Guilherme Perin. A comparison of weight initializers in deep learning-based side-channel analysis. In *Applied Cryptography and Network Security Workshops: ACNS 2020 Satellite Workshops, AIBlock, AIHWS, AIoTS, Cloud S&P, SCI, SecMT, and SiMLA, Rome, Italy, October 19–22, 2020, Proceedings 18*, pages 126–143. Springer, 2020.
- [18] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.
- [19] Keiron O’Shea and Ryan Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.
- [20] Keiron O’Shea and Ryan Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.
- [21] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318. Pmlr, 2013.
- [22] Santiago Pascual, Antonio Bonafonte, and Joan Serra. Segan: Speech enhancement generative adversarial network. *arXiv preprint arXiv:1703.09452*, 2017.
- [23] Sumit Saha. A comprehensive guide to convolutional neural networks. In <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>, 2018.
- [24] Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.
- [25] Vaios Stergiopoulos, Michael Vassilakopoulos, Eleni Tousidou, and Antonio Corral. Hyper-parameters tuning of artificial neural networks: An application in the field of recommender systems. In *New Trends in Database and Information Systems: ADBIS 2022 Short Papers, Doctoral Consortium and Workshops: DOING, K-GALS, MADEISD, MegaData, SWODCH, Turin, Italy, September 5–8, 2022, Proceedings*, pages 266–276. Springer, 2022.

-
- [26] Chun Chet Tan and Chikkannan Eswaran. Performance comparison of three types of autoencoder neural networks. In *2008 Second Asia International Conference on Modelling & Simulation (AMS)*, pages 213–218. IEEE, 2008.
- [27] Qi Wang, Yue Ma, Kun Zhao, and Yingjie Tian. A comprehensive survey of loss functions in machine learning. *Annals of Data Science*, pages 1–26, 2020.
- [28] Jianxin Wu. Introduction to convolutional neural networks. *National Key Lab for Novel Software Technology. Nanjing University, China*, 5(23):495, 2017.