



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΙΩΑΝΝΙΝΩΝ

ΣΧΟΛΗ ΟΙΚΟΝΟΜΙΚΩΝ ΚΑΙ ΔΙΟΙΚΗΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΚΑΤΑΣΚΕΥΑΖΟΜΕΝΑ ΤΕΧΝΗΤΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

Παππάς Δημήτριος

Επιβλέπων: Τσούλος Ιωάννης

Αναπληρωτής Καθηγητής

Άρτα, Μάρτιος, 2024

NEURAL NETWORK CONSTRUCTION

Εγκρίθηκε από τριμελή εξεταστική επιτροπή

Άρτα, 21/3/24

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. Επιβλέπων καθηγητής
Τσούλος Ιωάννης,
2. Μέλος επιτροπής
Τζάλλας Αλέξανδρος,
3. Μέλος επιτροπής
Χαριλόγης Βασίλειος,

© Παπάς, Δημήτριος, 2023.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Δήλωση μη λογοκλοπής

Δηλώνω υπεύθυνα και γνωρίζοντας τις κυρώσεις του Ν. 2121/1993 περί Πνευματικής Ιδιοκτησίας, ότι η παρούσα μεταπτυχιακή εργασία είναι εξ ολοκλήρου αποτέλεσμα δικής μου ερευνητικής εργασίας, δεν αποτελεί προϊόν αντιγραφής ούτε προέρχεται από ανάθεση σε τρίτους. Όλες οι πηγές που χρησιμοποιήθηκαν (κάθε είδους, μορφής και προέλευσης) για τη συγγραφή της περιλαμβάνονται στη βιβλιογραφία.

Παπάς, Δημήτριος

Υπογραφή

X

Παπάς Δημήτριος
Φοιτητής

ΠΕΡΙΛΗΨΗ

Αντικείμενο της παρούσας πτυχιακής εργασίας είναι η παρουσίαση μίας νέας μεθόδου κατασκευής τεχνητού νευρωνικού δικτύου, στο οποίο χρησιμοποιήθηκαν τεχνικές γενετικών αλγορίθμων με σκοπό την κατασκευή, αλλά και την βελτιστοποίηση του νευρωνικού δικτύου. Συγκεκριμένα στην παρούσα πτυχιακή θα σας παρουσιάσουμε το πρόγραμμα Neural Network Constructor ή για συντομία NNC, γραμμένο σε γλώσσα προγραμματισμού ANSI C++ που στοχεύει στην επίλυση διαφορικών εξισώσεων και στην ταξινόμηση δεδομένων, και δοκιμάστηκαν διάφοροι γνωστοί αλγόριθμοι τοπικής ελαχιστοποίησης πάνω σε αυτό, με τελικό στόχο την εξαγωγή της καλύτερης μεθόδου.

Η δομή της πτυχιακής εργασίας θα είναι χωρισμένη σε τέσσερις ενότητες. Ξεκινώντας με τις βασικές έννοιες που χρησιμοποιούνται στον τομέα των νευρωνικών δικτύων, όπως η ταξινόμηση και κατηγοριοποίηση των δεδομένων, η μάθηση και βελτιστοποίηση συναρτήσεων, καθώς και τον βασικό σκοπό της εργασίας. Στην συνέχεια θα γίνει αναφορά στα τεχνητά νευρωνικά δίκτυα, όπως και στα δίκτυα Perceptron, Adaline, Multilayer Perceptron και στις διάφορες μεθόδους βελτιστοποίησης που χρησιμοποιήθηκαν κατά την υλοποίηση της εργασίας και πιο συγκεκριμένα στις μεθόδους ADAM και RPROP, καθώς και σε παραδείγματα από την καθημερινότητα μας, όπου χρησιμοποιούνται τα νευρωνικά δίκτυα για την επίλυση διάφορων προβλημάτων. Συνεχίζοντας με αναφορά στους Γενετικούς αλγορίθμους, στην Γραμματική εξέλιξη, και στις εφαρμογές που έχουν πάνω στον τομέα των νευρωνικών δικτύων και θα ολοκληρώσουμε με την ανάλυση του προτεινόμενου αλγορίθμου NNC, καθώς και την ανάλυση και μελέτη των τελικών πειραματικών αποτελεσμάτων και θα αναφέρουμε τα τελικά συμπεράσματα της έρευνας.

Λέξεις-κλειδιά: Βελτιστοποίηση νευρωνικών δικτύων, Γενετικοί αλγόριθμοι, Διαφορικές εξισώσεις, Ταξινόμηση δεδομένων, Τεχνητό νευρωνικό δίκτυο.

ABSTRACT

The subject of this thesis is the presentation of a new method for constructing an artificial neural network, in which genetic algorithm techniques were used for both construction and optimization of the neural network. Specifically, in this thesis, we will introduce the program Neural Network Constructor, abbreviated as NNC, written in ANSI C++ programming language, which aims to solve differential equations and classify data. Various well-known local optimization algorithms were tested on it, with the ultimate goal of deriving the best method.

The structure of the thesis will be divided into four sections. Starting with the basic concepts used in the field of neural networks, such as data classification and categorization, learning, and function optimization, as well as the main purpose of the work. Next, there will be a reference to artificial neural networks, such as Perceptron, Adaline, Multilayer Perceptron networks, and the various optimization methods used in the implementation of the work, specifically the ADAM and RPROP methods, as well as examples from our daily lives where neural networks are used to solve various problems. Continuing with a reference to Genetic Algorithms, Grammatical Evolution, and the applications they have in the field of neural networks, and we will conclude with the analysis of the proposed NNC algorithm, as well as the analysis and study of the final experimental results, and we will report the final conclusions of the research.

Keywords: Artificial neural network, Classify data, Differential equations, Genetic algorithm, Local optimization algorithms.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ.....	5
ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ.....	7
ΚΑΤΑΛΟΓΟΣ ΔΙΑΓΡΑΜΜΑΤΩΝ/ΕΙΚΟΝΩΝ.....	8
ΠΙΝΑΚΑΣ ΣΥΝΤΟΜΟΓΡΑΦΙΩΝ	9
1 ΚΕΦΑΛΑΙΟ 1.....	11
1.1 Μάθηση με επίβλεψη	11
1.2 Μάθηση συναρτήσεων.....	14
1.3 Κατηγοριοποίηση δεδομένων	14
1.4 Βελτιστοποίηση συναρτήσεων.....	16
1.5 Σκοπός της εργασίας.....	18
2 ΚΕΦΑΛΑΙΟ 2	19
2.1 Τεχνητά νευρωνικά δίκτυα	19
2.2 Τα δίκτυα Perceptron	19
2.3 Τα δίκτυα Adaline.....	21
2.4 Τα δίκτυα MLP	23
2.5 Η μέθοδος Back Propagation	25
2.6 Η μέθοδος RPROP.....	26
2.7 Η μέθοδος ADAM.....	27
2.7 Παραδείγματα εφαρμογής νευρωνικών δικτύων	30
3 ΚΕΦΑΛΑΙΟ 3	33
3.1 Γραμματική εξέλιξη (Grammatical Evolution).....	33
3.2 Ιστορική αναδρομή	35
3.3 Γενετικοί τελεστές	36
3.4 Εφαρμογές γραμματικής εξέλιξης	37
4 ΚΕΦΑΛΑΙΟ 4	38

4.1	Κατασκευαζόμενα τεχνητά νευρωνικά δίκτυα	38
4.2	Παραλλαγές κατασκευαζόμενων τεχνητών νευρωνικών δικτύων.....	41
4.3	Ο προτεινόμενος αλγόριθμος.....	41
4.4	Πειραματικά αποτελέσματά.....	44
4.5	Συμπεράσματα	45
	ΒΙΒΛΙΟΓΡΑΦΙΑ	47

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 2 Τελικά αποτελέσματα δοκιμών	44
---	----

ΚΑΤΑΛΟΓΟΣ ΔΙΑΓΡΑΜΜΑΤΩΝ/ΕΙΚΟΝΩΝ

Εικόνα 0.1	Σφάλμα! Δεν έχει οριστεί σελιδοδείκτης.
Εικόνα 1.1 Διάγραμμα εκπαίδευσης μοντέλου με δεδομένα κατηγοριοποίησης.....	12
Εικόνα 1.2 Διάγραμμα δέντρου δεδομένων	13
Εικόνα 1.3 Διάγραμμα μεθόδου Gradient Descent	17
Εικόνα 2.1 Βιολογικός νευρώνας.....	19
Εικόνα 2.2 Τεχνητό νευρωνικό δίκτυο.....	19
Εικόνα 2.3 Υπολογισμός συνόλου βαρών.....	20
Εικόνα 2.4 Συνάρτηση ενεργοποίησης.....	20
Εικόνα 2.5 Μοντέλο McCulloch-Pitts.....	20
Εικόνα 2.6 Διαφορές Perceptron με ADALINE	22
Εικόνα 2.7 Τύπος πρόβλεψης.....	22
Εικόνα 2.8 Δομή νευρωνικού δικτύου με ένα κρυφό επίπεδο	23
Εικόνα 2.9 Αλγόριθμος RPROP	27
Εικόνα 2.10 Μαθηματικός τύπος Momentum.....	28
Εικόνα 2.11 Αποτέλεσμα με χρήση Momentum	28
Εικόνα 2.12 Αποτέλεσμα με χρήση Gradient Descent.....	28
Εικόνα 2.13 Μαθηματικός τύπος RPROP	29
Εικόνα 2.14 Αλγόριθμος ADAM	29
Εικόνα 2.15 Παράδειγμα εκπαίδευσης για τύπο email	30
Εικόνα 3.1 Διάγραμμα διαδικασίας Γενετικών αλγορίθμων.....	34
Εικόνα 3.2 Βήμα Mutation	36
Εικόνα 3.3 Αποτέλεσμα βήματος Mutation	37
Εικόνα 4.1 Διεπαφή NNC	39
Εικόνα 4.2 Τελικό αποτέλεσμα NNC.....	39
Εικόνα 4.3 Διάγραμμα Γραμματικής εξέλιξης.....	40
Διάγραμμα 1 Διάγραμμα τελικών αποτελεσμάτων NNC.....	45

ΠΙΝΑΚΑΣ ΣΥΝΤΟΜΟΓΡΑΦΙΩΝ

BNF.....	Backus–Naur form
CPU.....	Central processing unit
FNN.....	Feed forward neural networks
GPU.....	Graphics processing unit
NNC.....	Neural Network Constructor
H/Y.....	Ηλεκτρονικός υπολογιστής

ΕΙΣΑΓΩΓΗ

Τα τεχνητά νευρωνικά δίκτυα είναι η προσπάθεια του ανθρώπου να δημιουργήσει έναν ψηφιακό εγκέφαλο ο οποίος θα χρησιμοποιείται από τους Η/Υ με στόχο την επίλυση περίπλοκων προβλημάτων, χρησιμοποιώντας μεγάλο όγκο δεδομένων, τα οποία θα χρειαζόντουσαν πάρα πολύ χρόνο για έναν άνθρωπο να καταλάβει και να εξοικειωθεί ώστε να μπορέσει να βγάλει ένα τελικό αποτέλεσμα το οποίο θα είναι και το σωστό. Στις μέρες μας η μεγάλη και συνεχής ανάπτυξη των επεξεργαστών (*CPU*) στον τομέα των Η/Υ καθώς και των καρτών γραφικών (*GPU*), έχει οδηγήσει στην υλοποίηση και υποστήριξη περίπλοκων αλγορίθμων για κατασκευή και εκπαίδευση νευρωνικών δικτύων οι οποίοι εκτός ότι επωφελούνται από την δύναμη αυτών των συσκευών για να βγάλουν πολύ πιο σύντομα ένα ολοκληρωμένο αποτέλεσμα βοηθάνε και στην ανάπτυξη πολύ πιο περίπλοκων υλοποιήσεων, π.χ. η δημιουργία εικόνας μέσω εισαγωγής κειμένου από τον χρήστη, όπου βασίζεται αρκετά στις δυνατότητες των μοντέρνων καρτών γραφικών για γρήγορα και σωστά αποτελέσματα.

Σε αυτό το κεφάλαιο θα αναλύσουμε τους τρόπους με τους οποίους τα μοντέλα αξιοποιούν και οργανώνουν τις πληροφορίες που τους παρέχουν, καθώς και τα είδη δεδομένων τα οποία χρησιμοποιούνται κατά την εκπαίδευση μαζί με μερικά παραδείγματα. Στην συνέχεια θα μιλήσουμε για τα είδη συναρτήσεων μάθησης και βελτιστοποίησης και τέλος θα αναφερθεί ο γενικός σκοπός της Πτυχιακής εργασίας πάνω στον τομέα των κατασκευαζόμενων νευρωνικών δικτύων, σχετικά με το πως λειτουργεί, αν και θα αναφερθούμε πιο αναλυτικά στο 4ο κεφάλαιο, καθώς και τα προβλήματα που προσπαθεί να λύσει.

1 ΚΕΦΑΛΑΙΟ 1

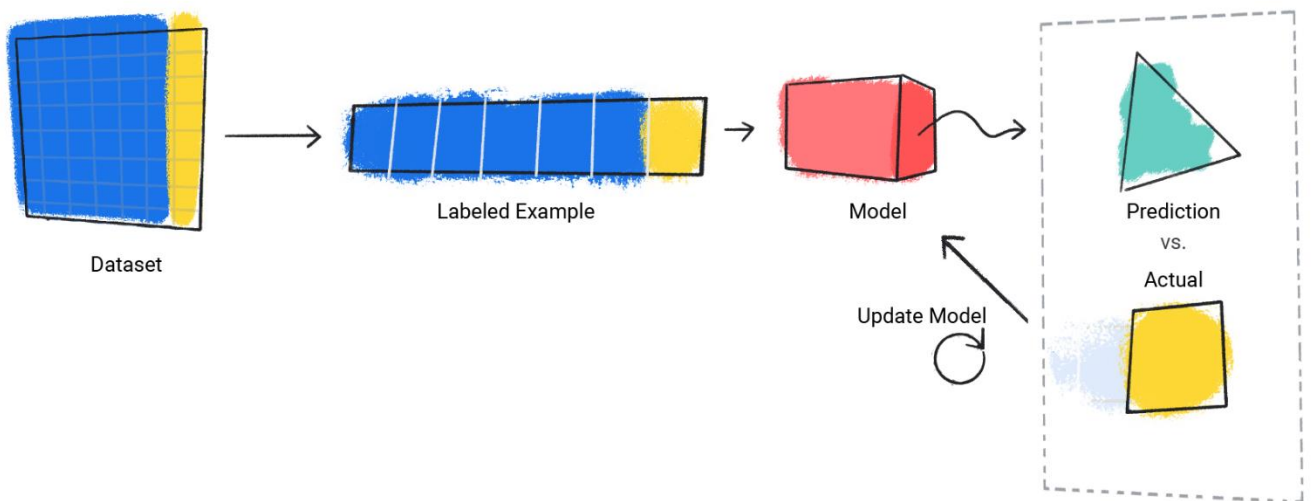
1.1 Μάθηση με επίβλεψη

Στον τομέα την μηχανικής μάθησης (*machine learning*) ένα μοντέλο με σκοπό να αναπτυχθεί και να εξελιχθεί στο τελικό προϊόν, το οποίο θα καταφέρνει να ολοκληρώνει με επιτυχία τα προβλέπει τα αποτελέσματα τα οποία ο χρήστης ζητάει, χρειάζεται αρκετό χρόνο εκπαίδευσης και επιτήρησης. Το κομμάτι του χρόνου είναι κάτι το οποίο μειώνεται χρόνο με τον χρόνο, λόγω της ανάπτυξης πιο ισχυρού hardware, αλλά και αλγορίθμων. Αν και το κομμάτι επιτήρησης δεν θεωρείται πάντα απαραίτητο για την εκπαίδευση ενός μοντέλου, σε αυτό το κεφάλαιο θα αναφερθούμε και θα αναλύσουμε την μέθοδο μάθησης με επίβλεψη γνωστή και ως supervised training, καθώς και τις χρήσεις που έχει σαν μέθοδος στην καθημερινότητα μας αναφέροντας μερικά παραδείγματα, αλλά θα κάνουμε και μία σύντομη περιγραφή για την μάθηση χωρίς επίβλεψη ή αλλιώς unsupervised training ώστε να υπάρχει ένα βασικό μέτρο σύγκρισης μεταξύ των δύο μεθόδων εκπαίδευσης. Ας ξεκινήσουμε βλέποντας πως ακριβώς λειτουργεί η μάθηση με επίβλεψη.

Η μάθηση με επίβλεψη είναι μία μέθοδος η οποία χρησιμοποιείται σε πάρα πολλές εφαρμογές στην καθημερινότητά μας, αρκετές φορές μάλιστα χωρίς ο χρήστης να το γνωρίζει ότι αλληλεπιδρά με ένα μοντέλο το οποίο έχει εκπαιδευτή με αυτόν τον τρόπο. Μερικά παραδείγματα είναι στον τομέα υπηρεσιών email, όπου ένα μοντέλο έχει εκπαιδευτεί ώστε να αναγνωρίζει αν ένα email είναι από τον πραγματικό αποστολέα ή από κάποιον κακόβουλο χρήστη. Στην αναγνώριση περιεχομένου σε μία εικόνα, όπου το μοντέλο μπορεί να αναγνωρίζει τα ανθρώπινα πρόσωπα που βρίσκονται σε αυτή και να υπολογίζει την ηλικία του καθενός, λειτουργεί η οποία είναι διαθέσιμη στα περισσότερα σύγχρονα κινητά τηλέφωνα στις μέρες μας και πολλές ακόμα λειτουργίες. Παρακάτω θα αναλύσουμε τι κάνει την συγκεκριμένη μέθοδο να ξεχωρίζει και τον τρόπο με τον οποίο εκπαιδεύεται ένα μοντέλο μέσω αυτής, αλλά και πως ακριβώς καταφέρνει να ξεχωρίζει με ακρίβεια τα δεδομένα.

Το συγκεκριμένο είδος εκπαίδευσης ξεκινάει με το μοντέλο να δέχεται έναν αριθμό από δεδομένα τα οποία έχουν είδη την σχετική περιγραφή (*label*) για το τι είναι (π.χ. μία φωτογραφία που περιγράφει τι ζώο απεικονίζει), ή με ποιο απλά λόγια δίνουμε από την αρχή της εκπαίδευσης την σωστή απάντηση στο μοντέλο, γιατί όμως; Ο λόγος που συμβαίνει αυτό, είναι ώστε το μοντέλο να αρχίζει να εκπαιδεύεται πρώτα πάνω σε αυτά τα

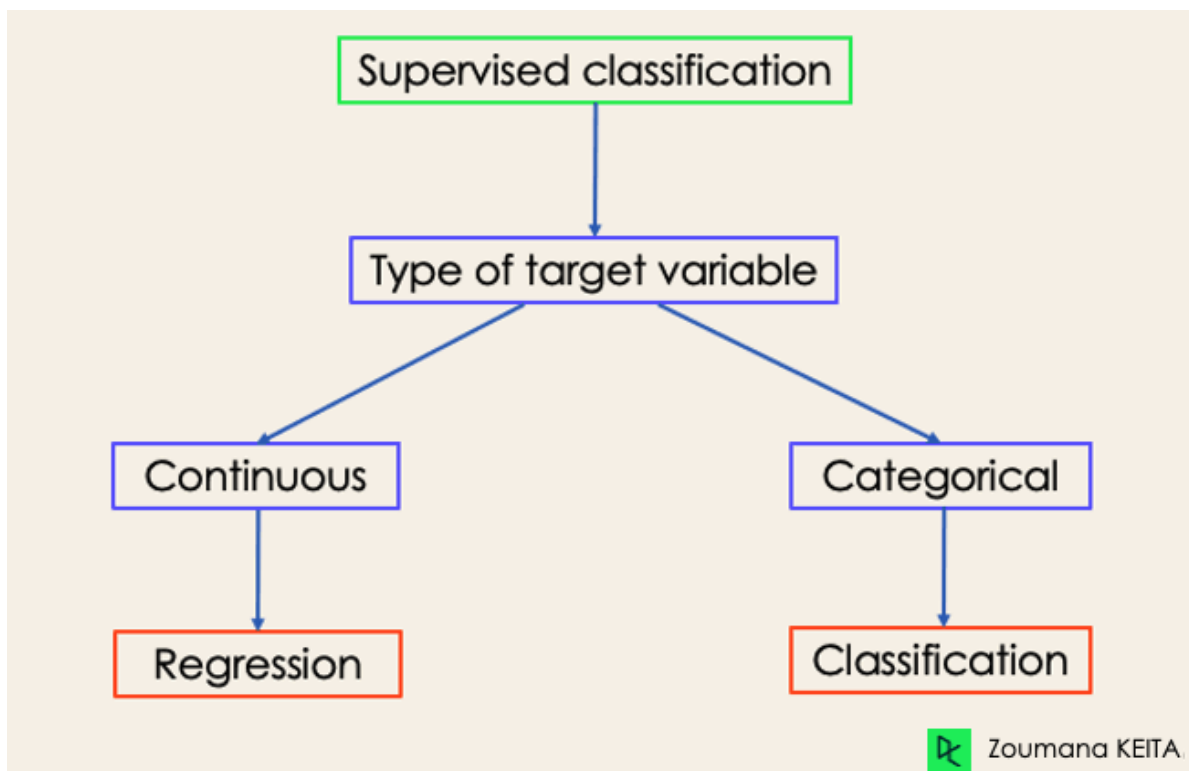
δεδομένα, με σκοπό να καταφέρει να βρει το καλύτερο δυνατό αποτέλεσμα, με βάση την σύγκριση του αρχικού δεδομένου και του τελικού αποτελέσματος που εξάγει το μοντέλο για το πρόβλημα που θέλουμε να λύσουμε και έπειτα εφόσον το μοντέλο είχε επιτυχία στην διαδικασία εκπαίδευσης, χρησιμοποιήσει όλες αυτές τις πληροφορίες πάνω σε δεδομένα άγνωστα για το μοντέλο, τα οποία δεν θα εμπεριέχουν καμία περιγραφή. Όλο το παραπάνω κομμάτι μπορεί να χωριστεί και να απλοποιηθεί με την βοήθεια του παρακάτω διαγράμματος:



Εικόνα 1.1 Διάγραμμα εκπαίδευσης μοντέλου με δεδομένα κατηγοριοποίησης

Όπως μπορούμε να δούμε από την παραπάνω φωτογραφία, ξεκινάμε με ένα σύνολο από δεδομένα (*Dataset*) στα οποία αναγράφουμε μία σχετική περιγραφή (*Labeled Example*) στο κάθε δεδομένο, με σκοπό να βοηθήσουμε το μοντέλο να βρει την σωστή απάντηση κάνοντας μία σύγκριση μεταξύ του τελικού αποτελέσματος που βρήκε (*Prediction*) και του επιθυμητού αποτελέσματος (*Actual*). Αν και εφόσον το αποτέλεσμα δεν ήταν αυτό που ζητάμε, τότε το μοντέλο θα ενημερώσει τις τιμές του (*weights, bias*) μέσω ενός αλγορίθμου βελτιστοποίησης και θα ξανά προσπαθήσει μέχρι να φτάσει στο επιθυμητό αποτέλεσμα. Συνήθως ή παραπάνω διαδικασία χρειάζεται αρκετό χρόνο για να ολοκληρωθεί λόγω του όγκου δεδομένων. Όταν η εκπαίδευση ολοκληρωθεί με επιτυχία, τότε το μοντέλο δοκιμάζεται πάνω σε συνθήκες όπου τα δεδομένα δεν εμπεριέχουν κάποια περιγραφή (*label*), με αποτέλεσμα να βασίζεται καθαρά από τις πληροφορίες και τις συσχετίσεις που έχει συλλέξει στην αρχική του εκπαίδευση, πράγμα που κάνει την ποιότητα των αρχικών δεδομένων που εκπαιδεύτηκε αρκετά σημαντική για το τελικό

αποτέλεσμα. Να σημειωθεί σε αυτό το σημείο ότι ανάλογα το είδος των δεδομένων, το μοντέλο κάνει και τις κατάλληλες προσαρμογές. Συγκεκριμένα αν τα δεδομένα που δέχεται είναι κατηγοριοποιημένα (π.χ. σκύλος ή γάτα), τότε η μέθοδος αυτή ονομάζεται classification και κατατάσσει τα αποτελέσματα σε ανάλογες κατηγορίες (π.χ. spam email ή όχι spam). Στο ενδεχόμενο που τα δεδομένα όμως είναι μεταβαλλόμενα (π.χ. η τιμή μίας μετοχής) όπου υπάρχει μία συνεχόμενη ροή, τότε η μέθοδος ονομάζεται regression και εξάγει μία συνεχόμενα μεταβαλλόμενη τιμή. Παρακάτω μπορούμε να δούμε με την βοήθεια ενός διαγράμματος πως ακριβώς κατηγοριοποιούνται οι μέθοδοι.



Εικόνα 1.2 Διάγραμμα δέντρου δεδομένων

Σε αυτό το σημείο χρειάζεται να αναφερθεί ότι υπάρχει και η μέθοδος χωρίς επίβλεψη ή λεγόμενη unsupervised training, όπου το μοντέλο δεν χρησιμοποιεί δεδομένα που εμπεριέχουν περιγραφή με σκοπό να γνωρίζει από την αρχή της εκπαίδευσης την σωστή πληροφορία του δεδομένου. Αντ' αυτού, σε κάθε εκπαίδευση το μοντέλο προσπαθεί να εντοπίσει μοτίβα και συσχετίσεις που υπάρχουν μεταξύ των δεδομένων που δέχεται, με σκοπό να καταλήξει στο σωστό συμπέρασμα, πράγμα που το κάνει αρκετά πιο περίπλοκο και χρονοβόρο σε αντίθεση με την μάθηση με επίβλεψη, αλλά αρκετά πιο

διαχειρίσιμο για τους προγραμματιστές, εφόσον έχουν να διαχειριστούν λιγότερο όγκο πληροφοριών για το μοντέλο μιας και παρέχουν απευθείας τα δεδομένα στο μοντέλο, χωρίς να ασχοληθούν να εισάγουν επιπλέον πληροφορίες σε αυτά, κάνοντας το κομμάτι οργάνωσης δεδομένων για εκπαίδευση αρκετά πιο απλοποιημένο, συγκριτικά με τον μέθοδο επίβλεψης όπου χρειάζονται επιπλέον βήματα για το πρώτο στάδιο εκπαίδευσης του νευρωνικού (Google for Developers, 2023) (GeeksforGeeks, 2023).

1.2 Μάθηση συναρτήσεων

Ένας αλγόριθμος μηχανικής μάθησης δεν είναι κάτι παραπάνω από μία μαθηματική συνάρτηση f η οποία συγκρίνει τις μεταβλητές εισόδου x (*input*) όπου είναι τα δεδομένα που περνούν στο μοντέλο ως το επιθυμητό αποτέλεσμα, με τις μεταβλητές εξόδου y (*output*) όπου βγάζει ως αποτέλεσμα το μοντέλο. Ο κύριος στόχος αυτής της διαδικασίας είναι να μπορεί να μάθει την σχέση μεταξύ της υπάρχον πληροφορίας που έχουμε και της πρόβλεψης που ζητάμε να βρει, καθώς και των επιπλέον λεπτομερών σχέσεων που υπάρχουν κάτω από κάθε δεδομένο. Ένα απλό παράδειγμα είναι ο υπολογισμός της ποιο σύντομης διαδρομής σε μία εφαρμογή GPS, όπου εκτός από την απόσταση χιλιομέτρων, χρειάζονται και μερικές επιπλέον πληροφορίες όπως είναι η ύπαρξη φαναριών ή διοδίων, καθώς και ο μέσος όρος κίνησης που υπάρχει σε κάθε δρόμο. Αυτές οι επιπλέον λεπτομέρειες βοηθούν στην συσχέτιση και στο γιατί το να προτιμήσει ο χρήστης την Α διαδρομή αντί της Β είναι καλύτερη ανεξάρτητα των χιλιομέτρων από την αρχική του θέση. Βέβαια η συγκεκριμένη διαδικασία δεν είναι αλάνθαστη, δυστυχώς πάντα υπάρχουν λάθη τα οποία δεν μπορούν να εξαλειφθούν πλήρως, παρ' όλο που υπάρχουν διάφοροι αλγόριθμοι που ελαττώνουν ως έναν βαθμό αυτά τα λάθη, κάνοντας το τελικό αποτέλεσμα αρκετά πιο κοντά στο επιθυμητό (Brownlee, 2019).

1.3 Κατηγοριοποίηση δεδομένων

Η κατηγοριοποίηση δεδομένων είναι ένας τρόπος με τον οποίο τα δεδομένα που πρόκειται να χρησιμοποιηθούν για την εκπαίδευση του τεχνητού νευρωνικού δικτύου, παρέχουν κάποια μορφή σχολίων ή αλλιώς ετικέτες (*label*). Αυτά τα σχόλια βοηθάνε στον διαχωρισμό των δεδομένων στις κατάλληλες κατηγορίες που ανήκουν, κάτι το οποίο είναι αρκετά χρήσιμο όταν πρόκειται να χρησιμοποιηθεί ένας τεράστιος όγκος δεδομένων για

την εκπαίδευση ενός μοντέλου, αλλά και στην οργάνωση όλης αυτής της πληροφορίας για μελλοντική χρήση από τους ίδιους τους προγραμματιστές.

Πιο συγκεκριμένα η κατηγοριοποίηση δεδομένων μπορεί να διαχωριστεί σε τουλάχιστον τρία γνωστά είδη. Ξεκινώντας με την πιο βασική κατηγοριοποίηση η οποία βασίζεται στο περιεχόμενο του κάθε δεδομένου. Αναλυτικότερα, ελέγχεται λεπτομερώς τι εμπεριέχει εσωτερικά το κάθε δεδομένο και του ανατίθεται με βάση το περιεχόμενο η ανάλογη κατηγορία. Για παράδειγμα, αν αναφερόμαστε για ένα δεδομένο που εμπεριέχει φωτογραφία από κάποιο ζώο, η συγκεκριμένη κατηγοριοποίηση εστιάζει στις λεπτομέρειες οι οποίες κάνουν το συγκεκριμένο ζώο να ξεχωρίζει από τα υπόλοιπα (π.χ. είναι τετράποδο, έχει μυτερά αυτιά, κλπ.) και το αναθέτει στην κατάλληλη κατηγορία, όπως σκύλος ή γάτα.

Επόμενο είδος είναι η κατηγοριοποίηση με βάση τα συμφραζόμενα που εμπεριέχονται εντός των δεδομένων. Αρκετές φορές το να κατηγοριοποιήσουμε ένα δεδομένο με βάση το περιεχόμενο του δεν είναι πάντα αρκετό, εδώ είναι που χρειάζονται επιπλέον παράγοντες οι οποίοι μπορούν να βοηθήσουν σε αυτό το πρόβλημα. Ένα παράδειγμα είναι η αυτόματη κατηγοριοποίηση των φωτογραφιών σε μία κινητή συσκευή σε φακέλους με βάση την περιοχή που έχει τραβηχτεί η κάθε φωτογραφία. Σε αυτή την περίπτωση δεν είναι εφικτό μόνο από το περιεχόμενο που υπάρχει στην φωτογραφία να γνωρίζουμε την ακριβές τοποθεσία της, ειδικά αν αρκετές λήψεις έχουν γίνει σε αρκετά παρόμοια μέρη, π.χ. έχοντας για υπόβαθρο ένα δάσος ή αν δεν υπάρχει κάποιο αρχαίο μνημείο όπου είναι ένα μοναδικό στοιχείο για μία περιοχή. Παρ' όλα αυτά, μπορούμε να χρησιμοποιήσουμε σαν επιπλέον πληροφορία το πότε έγινε η λήψη της φωτογραφίας χρονολογικά, την τοποθεσία μέσω του GPS της συσκευής που χρησιμοποιήθηκε για την λήψη και αρκετές άλλες πληροφορίες που βρίσκονται αποθηκευμένες σαν μεταδεδομένα (*metadata*) στην φωτογραφία με την βοήθεια της συσκευής που έγινε η λήψη και αποθήκευση.

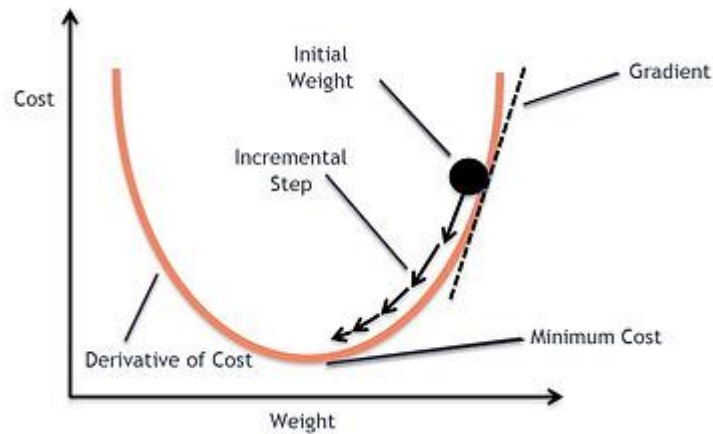
Τέλος είναι η κατηγοριοποίηση που γίνεται χειροκίνητα από κάποιον χρήστη. Σε αυτή την περίπτωση, ένας χρήστης ο οποίος είναι υπεύθυνος με βάση το υπόβαθρο, τις γνώσεις που διαθέτει και την κρίση του, να βάλει κάθε αναθέσει σε κάθε δεδομένο την κατάλληλη κατηγορία. Η συγκεκριμένη μέθοδος είναι αρκετά πιο αργή από τις υπόλοιπες μιας και οι προηγούμενες δύο είναι εφικτό να γίνουν πλήρως αυτοματοποιημένα με την χρήση κάποιου εκπαιδευμένου μοντέλου, το οποίο θα ελαττώσει κατά πολύ τον χρόνο

κατηγοριοποίησης και παράλληλα θα μειώσει το ενδεχόμενο λάθους που είναι εφικτό να γίνει από έναν άνθρωπο αν δεν είναι προσεκτικός ή δεν είναι σίγουρος για την επιλογή που θα κάνει (MonkeyLearn, 2020).

1.4 Βελτιστοποίηση συναρτήσεων

Η βελτιστοποίηση συναρτήσεων αποτελεί ένα από τα πιο σημαντικά κομμάτια για την εκπαίδευση ενός μοντέλου. Κατά την διάρκεια της εκπαίδευσης του μοντέλου, ξεκινάμε με την αρχικοποίηση των μεταβλητών έχοντας τυχαίες τιμές, κάνοντας το ενδεχόμενο να βρεθεί η σωστή λύση αρκετά δύσκολη. Σκοπός της βελτιστοποίησης εφόσον έχει υπολογιστεί η διαφορά μεταξύ της τιμής που βρήκε το μοντέλο και της σωστής τιμής (*συνάρτηση κόστους ή cost function*), είναι να καταφέρει να βρει τις κατάλληλες τιμές για κάθε μεταβλητή στο νευρωνικό, όπως είναι τα βάρη ή ο ρυθμός εκμάθησης, με στόχο να μειωθεί η συνάρτηση κόστους και να εξάγει το σωστό τελικό αποτέλεσμα, αν και όχι πάντα με τέλεια επιτυχία (Doshi, 2019).

Πιο αναλυτικά, υπάρχουν αρκετοί αλγόριθμοι βελτιστοποίησης όπως θα δούμε και στο 2ο και 3ο κεφάλαιο, οι οποίοι εκτός ότι έχουν διαφορετική προσέγγιση για τον τρόπο που θα βρουν το κατάλληλο αποτέλεσμα, μερικοί μάλιστα αντλούν έμπνευση ακόμα και από την ίδια την φύση και τον τρόπο εξέλιξης, όπως για παράδειγμα οι γενετικοί αλγόριθμοι. Ένας από τους πιο διαδεδομένους αλγορίθμους είναι ο Gradient Descent όπου κάνει συνεχόμενη προσαρμογή των μεταβλητών του μοντέλου, όπως είναι τα βάρη προς την κατεύθυνση της πιο απότομης κλήσης της συνάρτησης, φτάνοντας με αυτό τον τρόπο στο χαμηλότερο σημείο της, δηλαδή στο τοπικό ελάχιστο της. Το πόσο μεγάλο ή μικρό θα είναι το βήμα που θα κάνει προς το τοπικό ελάχιστο εξαρτάται από τον συντελεστή μάθησης (*learning rate*) και θα δούμε στην συνέχεια των επόμενων κεφαλαίων, πως έχοντας μεταβαλλόμενο συντελεστή μάθησης αντί για σταθερό όπως στην περίπτωση του Gradient Descent, βοηθάει στην γρηγορότερη και καλύτερη εξαγωγή των τελικών αποτελεσμάτων (Chauhan, 2024).



Εικόνα 1.3 Διάγραμμα μεθόδου Gradient Descent

Όλοι αυτοί οι αλγόριθμοι χωρίζονται σε κατηγορίες, μία από αυτές είναι η κατηγορία των στοχαστικών αλγορίθμων βελτιστοποίησης, στην οποία και θα εστιάσουμε σε αυτό το υπό-κεφάλαιο, αν και χρειάζεται να αναφερθεί για την καλύτερη ολική εικόνα, ότι υπάρχουν και οι ντετερμινιστικοί αλγόριθμοι βελτιστοποίησης όπου ανήκει ο αλγόριθμος Gradient Descent που αναφέραμε παραπάνω. Ένα από τα κύρια χαρακτηριστικά τους το οποίο αποτελεί και σε αρκετές περιπτώσεις η κύρια αδυναμία τους είναι το ενδεχόμενο να μην καταφέρουν να βρουν το καλύτερο εφικτό αποτέλεσμα. Ο λόγος πίσω από αυτό το πρόβλημα, είναι η τυχαία προσέγγιση που έχουν για την εύρεση του καλύτερου αποτελέσματος, κάνοντας τους ιδανικούς για μεγάλα προβλήματα που χρειάζεται να βελτιστοποιηθούν. Με μία πιο απλή περιγραφή, το συγκεκριμένο είδος αλγορίθμων δεν βασίζεται στο να αναλύσει όλα τα δεδομένα ένα προς ένα, αλλά ένα μέρος αυτού με βάση διάφορες τεχνικές που διαθέτει κάθε αλγόριθμος, ώστε να βγάλει το τελικό αποτέλεσμα. Ένας από τους πιο γνωστούς αλγορίθμους είναι ο Stochastic Gradient Descent, που είναι μία παραλλαγή του αλγορίθμου Gradient Descent και υπολογίζει με βάση ένα μικρό μέρος των δεδομένων, αντί για όλα τα δεδομένα, κάνοντας τον αρκετά γρήγορο σε σενάρια με μεγάλο όγκο πληροφορίας. Οι γενετικοί αλγόριθμοι, στους οποίους θα δούμε αναλυτικά στο 3ο κεφάλαιο και βασίζονται στην ιδέα της φυσικής επιλογής και της γενετικής, είναι άλλο ένα είδος αλγορίθμων βελτιστοποίησης. Τέλος πρέπει να αναφέρουμε άλλον έναν αλγόριθμο ο οποίος όπως οι γενετικοί αλγόριθμοι, αντλεί επιρροές από την φύση. Ο αλγόριθμος αυτός ονομάζεται Particle Swarm Optimization και βασίζεται στην συμπεριφορά που έχουν ζώα στην φύση που μετακινούνται σε σμήνους όπως είναι τα πουλιά ή τα ψάρια. Στον συγκεκριμένο αλγόριθμο ο τρόπος εύρεσης της καλύτερης τιμής γίνεται με την χρήση σωματιδίων (*particles*), τα οποία “απλώνονται” σε όλες τις πιθανές λύσεις και όπως και τα πουλιά στην πραγματική ζωή καταγράφουν και

θυμούνται από ποια σημεία έχουν βρεθεί και ποιο είναι το καλύτερο από αυτά (Chandrakant, 2023).

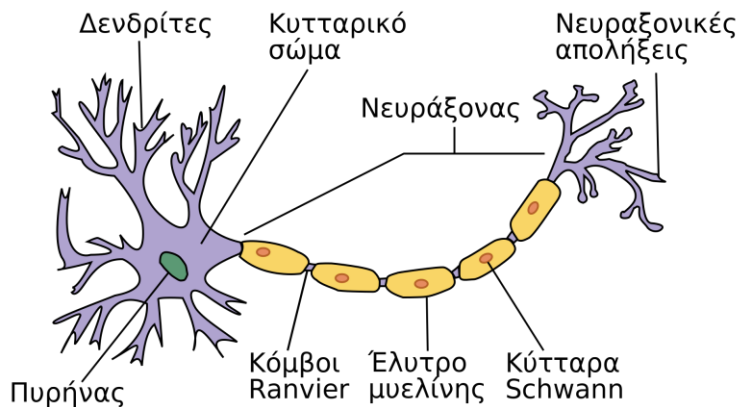
1.5 Σκοπός της εργασίας

Σκοπός της πτυχιακής εργασίας είναι η παρουσίαση ενός νέου προγράμματος με όνομα NNC (*Neural Network Constructor*) που σκοπεύει στην κατασκευή ενός τεχνητού νευρωνικού δικτύου χρησιμοποιώντας μία μέθοδο Γραμματικής εξέλιξης (*Grammatical Evolution*) με την βοήθεια της γραμματικής BNF για την κατασκευή του. Είναι γραμμένο σε γλώσσα προγραμματισμού ANSI C++ και χρησιμοποιεί την δωρεάν βιβλιοθήκη ανοιχτού κώδικα QT, κάνοντας το διαθέσιμο για τα περισσότερα λειτουργικά συστήματα, όπως είναι τα Windows, MacOS και Linux, αρκεί μονάχα ο χρήστης να έχει εγκατεστημένα τα ανάλογα πακέτα για την εκτέλεση του προγράμματος. Η χρήση του συγκεκριμένου προγράμματος στοχεύει στην επίλυση προβλημάτων διαφορετικών εξισώσεων, καθώς και για προβλήματα κατηγοριοποίησης δεδομένων. Στην συνέχεια της εργασία πρόκειται να δούμε πιο αναλυτικά τον ακριβές τρόπο λειτουργίας του προγράμματος, την δομή που έχει, τον τρόπο με τον οποίο ο χρήστης αλληλοεπιδρά με αυτό, καθώς και την μέθοδο που χρησιμοποιεί για την κατασκευή των τεχνητών νευρωνικών δικτύων και θα συγκρίνουμε την προτεινόμενη μέθοδο με διάφορους γνωστούς αλγόριθμους βελτιστοποίησης, όπως είναι ο ADAM και RPROP, όπου δοκιμάστηκαν πάνω σε κοινά παραδείγματα για την εξαγωγή των τελικών συμπερασμάτων.

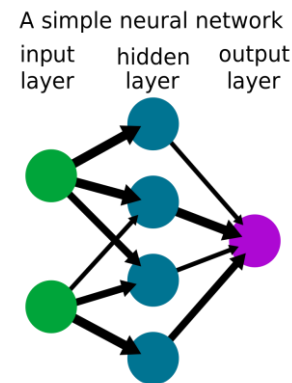
2 ΚΕΦΑΛΑΙΟ 2

2.1 Τεχνητά νευρωνικά δίκτυα

Τα τεχνητά νευρωνικά δίκτυα είναι μία προέκταση της μηχανικής μάθησης και πιο συγκεκριμένα, είναι ένα μοντέλο του οποίου η κύρια πηγή έμπνευσης για την υλοποίηση του είναι ένα από τα πιο περίπλοκα όργανα του ανθρώπινου οργανισμού, ο εγκέφαλος. Πιο συγκεκριμένα, αποτελείται από ένα κύκλωμα διασυνδεδεμένων νευρώνων οι οποίοι σκοπεύουν στην επίλυση υπολογιστικών προβλημάτων. Η δομή τους είναι ένα δίκτυο από κόμβους οι οποίοι ενώνεται μεταξύ τους. Χωρίζονται σε τρία είδη, τους νευρώνες εισόδου (*input*), υπεύθυνοι για την εισαγωγή των δεδομένων στο νευρωνικό δίκτυο. Τους κρυμμένους νευρώνες (*hidden*) οι οποίοι πολλαπλασιάζουν κάθε είσοδο με τα σχετικά βάρη και το αποτέλεσμα που βγάζουν το χρησιμοποιούν στην συνάρτηση ενεργοποίησης, που με βάση την τιμή ο νευρώνας ενεργοποιείται ή όχι. Τέλος τους νευρώνες εξόδου (*output*) που εξάγουν το τελικό αποτέλεσμα που υπολόγισε το νευρωνικό δίκτυο. Σε αυτό το κεφάλαιο θα αναλύσουμε τα είδη πολλαπλών τεχνητών νευρωνικών δικτύων καθώς και μερικούς από τους γνωστούς αλγόριθμους βελτιστοποίησης οι οποίοι και χρησιμοποιήθηκαν στις δοκιμές της έρευνας (Wikipedia, 2024) (AWS, 2024).



Εικόνα 2.1 Βιολογικός νευρώνας



Εικόνα 2.2 Τεχνητό νευρωνικό δίκτυο

2.2 Τα δίκτυα Perceptron

Τα δίκτυα Perceptron αποτελούν τα πρώτα και πιο απλά είδη τεχνητών νευρωνικών δικτύων. Συγκεκριμένα ξεκίνησε πίσω το 1943 από τον νευρο-επιστήμονα Warren McCulloch και τον λογικό επιστήμονα Walter Pitts οι οποίοι πρότειναν στην έρευνα τους με τίτλο “*A Logical calculus of the ideas immanent in nervous activity*”, την

δημιουργία ενός νευρώνα εν ονόματι νευρώνας McCulloch-Pitts, ο οποίος θα είναι όσο πιο πιστός γίνεται στον ανθρώπινο νευρώνα. Ο τρόπος λειτουργίας του θα ήταν να δέχεται πολλαπλές πληροφορίες ως είσοδο σε δυαδική μορφή και θα εξάγει ένα μόνο αποτέλεσμα πάλι σε δυαδική μορφή (Pitts, 1943).

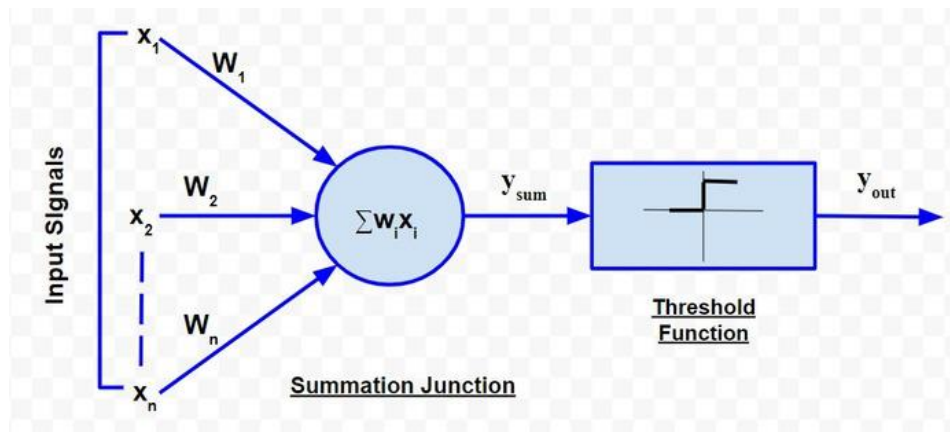
$$y_{sum} = \sum_{i=1}^2 w_i x_i$$

Εικόνα 2.3 Υπολογισμός συνόλου βαρών

$$y_{out} = f(y_{sum}) = \begin{cases} 1, & x \geq 1 \\ 0, & x < 1 \end{cases}$$

Εικόνα 2.4 Συνάρτηση ενεργοποίησης

Στην ουσία το σκεπτικό πίσω από την υλοποίηση των McCulloch και Pitts θυμίζει μία λογική πύλη, όπως είναι η AND ή OR, όπου έχουμε δύο δυαδικές μεταβλητές που χρησιμοποιούνται σαν είσοδοι και μία συνάρτηση η οποία ενεργοποιείται μονάχα με τις κατάλληλες τιμές στα βάρη και τις μεταβλητές εισόδου.



Εικόνα 2.5 Μοντέλο McCulloch-Pitts

Την σκυτάλη σε αυτή την έρευνα πήρε το 1957 ο ψυχολόγος Frank Rosenblatt, ο οποίος βασισμένος από την έρευνα του McCulloch & Pitts, παρουσίασε το πρώτο γνωστό σε αρκετούς πλέον μαθηματικό μοντέλο εν ονόματι Perceptron, το οποίο όμως ξεκίνησε σαν μία συσκευή με στόχο την αναγνώριση περιεχομένου από εικόνες, όπου και παίρνει το

όνομα του από την αγγλική λέξη *perception*, όπου σημαίνει οπτική (Banoula, 2023) (Bento, 2021).

Πιο αναλυτικά είναι ένα τεχνητό νευρωνικό δίκτυο με ένα μόνο επίπεδο και έχει την δυνατότητα να μαθαίνει με βάση τις πληροφορίες που λαμβάνει μία την φορά, με σκοπό να κάνει δυαδικές κατηγοριοποιήσεις. Ο τρόπος λειτουργίας του, χωρίζεται σε 5 κομμάτια. Αρχικά, υπάρχουν τα επίπεδα εισόδου (*input layers*), τα οποία αποτελούνται από έναν ή περισσότερους νευρώνες όπως θα δούμε και στην συνέχεια, οι οποίοι δέχονται τα ερεθίσματα, δηλαδή τις πληροφορίες είτε από τον έξω κόσμο, είτε από κάποιο διαφορετικό νευρωνικό δίκτυο. Στην συνέχεια, υπάρχουν τα βάρη (*weights*), όπου είναι υπεύθυνα για το πόσο έντονο θα είναι το ερέθισμα για κάθε είσοδο που δέχεται το νευρωνικό δίκτυο. Τέλος περνάμε στην συνάρτηση ενεργοποίησης (*activation function*), η οποία είναι υπεύθυνη για το τελικό αποτέλεσμα (*output*) που θα βγάλει το νευρωνικό δίκτυο ή κάθε μεμονωμένος νευρώνας σε πιο περίπλοκα νευρωνικά δίκτυα, η οποία βασίζεται από το άθροισμα όλων των προηγούμενων μεταβλητών, δηλαδή των δεδομένων σε συνδυασμό με τα βάρη και το bias.

Ένα δίκτυο Perceptron μπορεί να χωριστεί σε δύο κατηγορίες. Πιο συγκεκριμένα είτε σε ένα μόνο επίπεδο (*Single layer*) το οποίο είναι και το προτεινόμενο από τον Rosenblatt, όπου η δομή του αποτελείται μονάχα από μια στρώση κόμβων εισόδου οι οποίοι συνδέονται απευθείας με τον κόμβο εξόδου. Όντας αρκετά απλοϊκός τον κάνει και δύσκολο για περίπλοκες χρήσεις. Εδώ είναι που εντάσσονται τα Perceptron πολλαπλών επιπέδων (*Multilayer Perceptron*), τα οποία εισάγουν μεταξύ των κόμβων εισόδου και εξόδου επιπλέον κρυφά στρώματα (*Hidden layers*), τα οποία κάνουν το μοντέλο πιο ικανό να υποστηρίξει πιο απαιτητικά προβλήματα. Τα πολλαπλά στρώματα, είναι ο προτεινόμενη δομή νευρωνικών δικτύων στις μέρες μας και έχουν δημιουργηθεί αρκετοί αλγόριθμοι με σκοπό την βελτιστοποίηση του, όπως θα δούμε και παρακάτω.

2.3 Τα δίκτυα Adaline

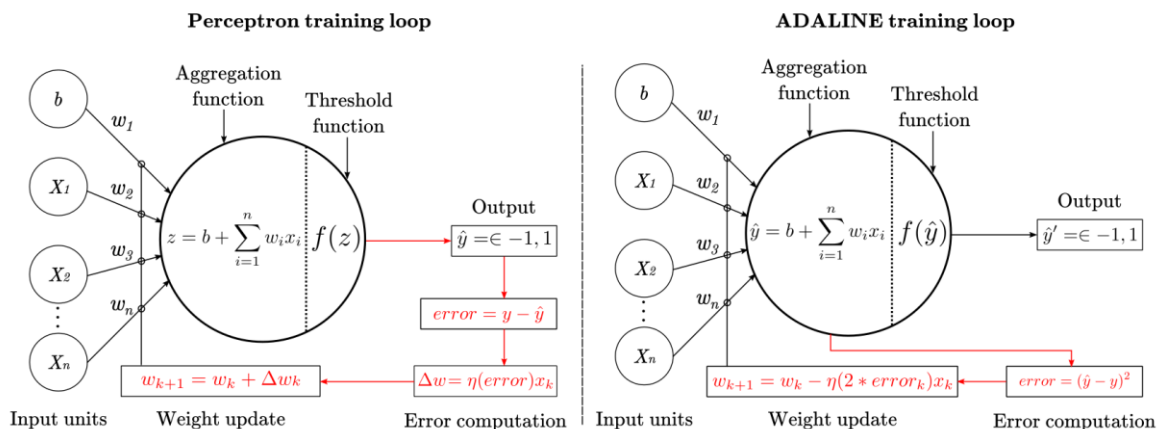
Τα δίκτυα Adaline (*Adaptive Linear Neuron* ή *Adaptive Linear Element*) παρουσιάστηκαν στο πανεπιστήμιο του Stanford από τους Bernard Widrow και τον μαθητή του Ted Hoff, δημιουργό των μικροεπεξεργαστών, το 1959, δύο χρόνια από την εμφάνιση του μοντέλου Perceptron από τον Rosenblatt (Wikipedia, 2024). Η βασική δομή

τους δεν διαφέρει πολύ από την δομή του μοντέλου Perceptron, δηλαδή παραμένει ένας νευρώνας ενός επιπέδου με πολλαπλές εισόδους και μία έξοδος δυαδικού τύπου (0 ή 1). Παρ' όλο που ακόμα και η μαθηματική συνάρτηση τους είναι η ίδια με αυτή του Perceptron, η κύρια διαφορά τους είναι ο τρόπος με τον οποίο μαθαίνουν να εξάγουν το τελικό αποτέλεσμα.

$$\hat{y} = b + \sum_{i=1}^n w_i x_i = w_0 x_0 + w_1 x_1 + \dots + w_n x_n$$

Εικόνα 2.7 Τύπος πρόβλεψης

Πιο αναλυτικά, τα δίκτυα Adaline έχουν την δυνατότητα να μπορούν να προσαρμόζουν τα βάρη του μοντέλου, με σκοπό την μείωση σφαλμάτων που εξάγει το νευρωνικό δίκτυο, συγκρίνοντας το επιθυμητό αποτέλεσμα με αυτό που κατάφερε να προβλέψει το μοντέλο και χρησιμοποιώντας τον γνωστό αλγόριθμο βελτιστοποίησης Gradient Descent, ο οποίος έχει ως βασικό σκοπό να προσαρμόσει τα βάρη του μοντέλου επαναλαμβανόμενα μέχρι να καταλήξει στο επιθυμητό αποτέλεσμα, δηλαδή να μειωθεί το σφάλμα σε λογικά πλαίσια, πριν να φτάσουμε στον μέγιστο αριθμό εποχών (*epoch*) στο μοντέλο, όπου και θα σταματήσει η διαδικασία εκπαίδευσης. Παρακάτω μπορούμε να δούμε την δομή η οποία είναι ουσιαστικά πανομοιότυπη, αλλά και τις κύριες διαφορές στον τρόπο εκπαίδευσης μεταξύ των δύο αλγορίθμων Perceptron και Adaline, οι οποίες τονίζονται στο γράφημα με κόκκινο χρώμα και αφορούν τον τρόπο που υπολογίζεται το



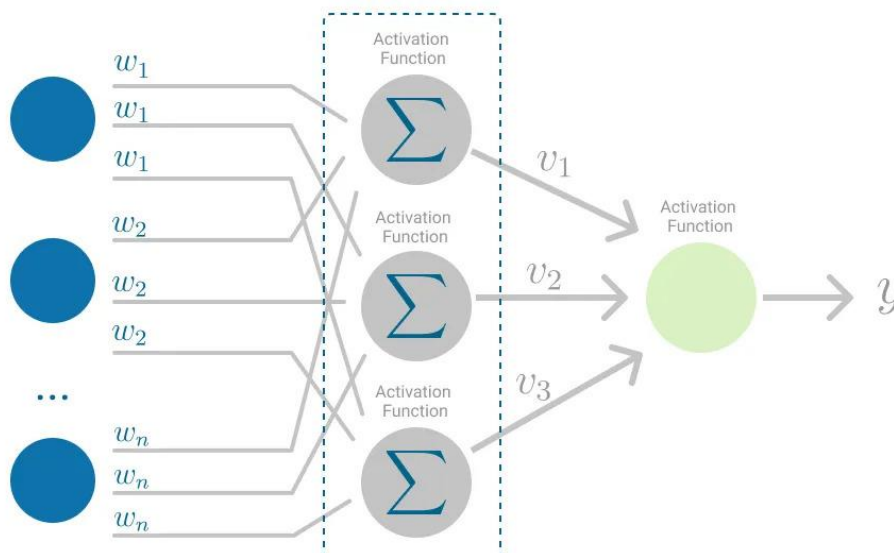
Εικόνα 2.6 Διαφορές Perceptron με ADALINE

σφάλμα καθώς και ο τρόπος που ενημερώνονται τα βάρη (Caceres, 2020).

2.4 Τα δίκτυα MLP

Τα δίκτυα MLP (*Multilayer Perceptron*) χρησιμοποιούνται αρκετά στον τομέα των βαθιών νευρικών δικτύων. Δημιουργήθηκαν με σκοπό να καλύψουν το κενό που υπήρχε από τον αλγόριθμο Perceptron, όπου είναι ικανός μόνο για χρήση με δεδομένα γραμμικής παλινδρόμησης, κάνοντας τον αρκετό περιορισμένο για χρήση σε περίπλοκα προβλήματα που απαιτούν την χρήση μη γραμμικών δεδομένων. Αναλυτικότερα τα δίκτυα MLP είναι ένα νευρωνικό δίκτυο όπου η συσχέτιση μεταξύ των δεδομένων εισόδου και των αποτελεσμάτων είναι μη γραμμική.

Στον πυρήνα τους τα συγκεκριμένα δίκτυα αποτελούνται από τους κόμβους εισόδου και εξόδου, αλλά και από μία νέα προσθήκη επιπέδου, το λεγόμενο κρυφό επίπεδο (*hidden layers*), η οποία εμπεριέχει πολλαπλούς νευρώνες τον έναν πάνω στον άλλο. Ανήκουν στην κατηγορία των δικτύων *feed forward neural networks (FNN)*. Το όνομα τους βασίζεται από την διαδικασία με την οποία διοχετεύεται η πληροφορία μέσα στο νευρωνικό. Ξεκινώντας από τον κόμβο εισόδου όπου περνάει η πληροφορία ως μεταβλητή εισόδου, έπειτα η μεταβλητή περνάει στους κρυφούς κόμβους, όπου και συνδυάζεται με τα βάρη, με σκοπό το τελικό άθροισμα το οποίο χρησιμοποιείται στην συνάρτηση ενεργοποίησης κάθε νευρώνα. Με την σειρά του το αποτέλεσμα περνάει στον επόμενο κόμβο, μέχρι να φτάσει στον τελικό όπου είναι η έξοδος. Όμως τα συγκεκριμένα δίκτυα από μόνα τους δεν μπορούν να παρέχουν ολοκληρωμένα και σωστά αποτελέσματα, διότι δεν μπορούν να γνωρίζουν τα κατάλληλα βάρη όπου θα ελαχιστοποιήσουν το σφάλμα του δικτύου. Γι' αυτό και βασίζονται στον αλγόριθμο *Back Propagation* για την



ενημέρωση των βαρών, τον οποίο θα δούμε αναλυτικά το πως ακριβώς λειτουργεί στο επόμενο υπό κεφάλαιο (Bento, 2021).

2.5 Η μέθοδος Back Propagation

Ένα μοντέλο στα αρχικά στάδια υλοποίησης και δοκιμών έχει σίγουρο ποσοστό λαθών κατά τις εκτιμήσεις που πρόκειται να κάνει. Εδώ είναι που χρησιμοποιείται η μέθοδος Back Propagation. Ο συγκεκριμένος αλγόριθμος είναι υπεύθυνος στην βελτιστοποίηση του νευρωνικού δικτύου, ώστε να καταφέρει να αλλάξει κατάλληλα τα βάρη του μοντέλου, με σκοπό στην επόμενη εποχή εκπαίδευσης (*training epoch*), τα αποτελέσματα του μοντέλου να έχουν μεγαλύτερη πιθανότητα να είναι κοντά στο επιθυμητό αποτέλεσμα μειώνοντας την τιμή της συνάρτησης κόστους (Kostadinov, 2019).

Πιο αναλυτικά, ο αλγόριθμος Back Propagation χρησιμοποιείται σε νευρωνικά δίκτυα τα οποία εκπαιδεύονται μέσω μάθησης με επίβλεψη και στην ουσία θεωρείται ο αλγόριθμος Gradient Descent, ο οποίος είναι υπεύθυνος σε ένα νευρωνικό για τον υπολογισμό του τοπικού ελάχιστου και στην συνέχεια, της καλύτερης χαμηλότερης τιμής σφάλματος που μπορεί να βγει από την έξοδο (*output*) του νευρωνικού, θέτοντας τα κατάλληλα βάρη που θα μπορέσουν να το επιτύχουν. Η έξοδος όπως έχουμε αναφέρει και σε προηγούμενα υπό κεφάλαια, είναι η διαφορά μεταξύ του αποτελέσματος που κατάφερε να βγάλει το νευρωνικό δίκτυο και του επιθυμητού αποτελέσματος που θέλουμε να λάβουμε βασισμένο από το δεδομένο που έχουμε δώσει ως μεταβλητή εισόδου στο μοντέλο. Η διαδικασία αυτή είναι γνωστή ως συνάρτηση κόστους. Η ροή που γίνεται για να επιτευχθεί αυτό το αποτέλεσμα είναι η παρακάτω:

1. Ξεκινάμε με τον υπολογισμό της εξόδου (*output*) του νευρωνικού δικτύου χρησιμοποιώντας τα δεδομένα ως είσοδο (*input*) και περνάμε από κάθε επίπεδο (*layers*), χρησιμοποιώντας τις προκαθορισμένες τιμές για τα βάρη (*weights*), η διαδικασία είναι η FNN όπως έχουμε αναφέρει και σε προηγούμενο υπό κεφάλαιο.
2. Υπολογίζουμε την διαφορά μεταξύ του επιθυμητού αποτελέσματος και του αποτελέσματος που βγήκε από το νευρωνικό δίκτυο, δηλαδή την συνάρτηση κόστους.
3. Ακολουθούμε αντίθετη φορά, ξεκινώντας από το τέλος (*output*) του νευρωνικού δικτύου προς την είσοδο του (*input*), από εκεί παίρνει και το όνομα του ο αλγόριθμος Back Propagation, λόγω της φοράς που ακολουθεί.
4. Κατά την πορεία του περνώντας από κάθε επίπεδο ελέγχει τα λάθη για κάθε κόμβο (*node*) και προσαρμόζει κατάλληλα τα βάρη (*weights*), με σκοπό να αυξήσει την

τιμή της συνάρτησης ενεργοποίησης (*activation function*) για τους σωστούς κόμβους και να μειώσει εξίσου για τους λάθος.

5. Επαναλαμβάνουμε ξανά το πρώτο βήμα, αυτή την φορά έχοντας τις νέες τιμές στα βάρη όπου ο αλγόριθμος έθεσε, μέχρι να βρεθεί η λύση ή να ολοκληρωθεί ο αριθμός των επαναλήψεων.

Σε αυτό το σημείο πρέπει να αναφερθεί το γεγονός ότι υπάρχουν κάποια όρια σχετικά με την αποδοτικότητα του αλγορίθμου, διότι υπάρχουν περιπτώσεις όπου ο αλγόριθμος χρειάζεται να γνωρίζει από την αρχή τις παραγώγους (*derivatives*) της κάθε συνάρτησης ενεργοποίησης, με σκοπό να ολοκληρωθεί η διαδικασία, καθώς εξίσου επηρεάζει το πόσο περίπλοκο είναι το μέγεθος ενός νευρωνικού δικτύου. Ακριβώς λόγω αυτών των περιορισμένων ερευνητές έχουν καταφέρει να βρουν εναλλακτικούς τρόπους βελτιστοποίησης των νευρωνικών δικτύων όπως θα δούμε παρακάτω.

2.6 Η μέθοδος RPROP

Η μέθοδος RPROP (*Resilient Back Propagation*), είναι ένας αλγόριθμος βελτιστοποίησης νευρωνικών δικτύων. Δημοσιεύθηκε σαν ερευνητικό έγγραφο το 1993 από τους Martin Riedmiller και Heinrich Braun με τίτλο “*A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm*” (Braun, 1993) , ο οποίος καταφέρνει να εξαλείψει ένα πρόβλημα που εμφανίζεται στον αλγόριθμο βελτιστοποίησης Back Propagation, το οποίο επηρεάζεται κυρίως από το σταθερό learning rate το οποίο και χρησιμοποιεί κατά την διάρκεια ενημέρωσης των βαρών, κάνοντας την διαδικασία αρκετά πιο αργή.

Συγκεκριμένα, η RPROP καταφέρνει να διορθώσει αυτό το ελάττωμα, κάνοντας το learning rate να είναι μεταβλητό κατά την ενημέρωση για κάθε βάρος μεμονωμένα στο δίκτυο, έχοντας στην ουσία πολλαπλές τιμές αντί για μία σταθερή μεταξύ των μεταβλητών. Η τιμή που λαμβάνει το learning rate για κάθε βάρος, βασίζεται αν το σφάλμα στο συγκεκριμένο βάρος είναι αυξανόμενο ή όχι και στο ενδεχόμενο που το σφάλμα αλλάζει κατεύθυνση, τότε το learning rate επαναφέρεται στην αρχική του τιμή. Εκτός από την διαφορετική προσέγγιση με την αλλαγή της τιμής learning rate, η μέθοδος RPROP εισάγει ακόμα μία νέα μεταβλητή που ονομάζεται step size ή delta. Σκοπός της μεταβλητής είναι να επιταχύνει την διαδικασία ενημέρωσης των βαρών στο ενδεχόμενο

που το σφάλμα παραμείνει σταθερό, προσφέροντας πιο γρήγορα το επιθυμητό αποτέλεσμα. Μπορούμε να το φανταστούμε σαν να γνωρίζουμε με σιγουριά ότι ο δρόμος που ακολουθούμε είναι ο σωστός, με αποτέλεσμα να επιτυγχάνουμε την διαδικασία να φτάσουμε στο τέρμα της διαδρομής.

```

input :  $\theta_0 \in \mathbf{R}^d$  (params),  $f(\theta)$  (objective),
          $\eta_{+/-}$  (etaplus, etaminus),  $\Gamma_{max/min}$  (step sizes)
initialize :  $g_{prev}^0 \leftarrow 0$ ,  $\eta_0 \leftarrow lr$  (learning rate)

```

```

for  $t = 1$  to ... do
   $g_t \leftarrow \nabla_{\theta} f_t(\theta_{t-1})$ 
  for  $i = 0, 1, \dots, d - 1$  do
    if  $g_{prev}^i g_t^i > 0$ 
       $\eta_t^i \leftarrow \min(\eta_{t-1}^i \eta_+, \Gamma_{max})$ 
    else if  $g_{prev}^i g_t^i < 0$ 
       $\eta_t^i \leftarrow \max(\eta_{t-1}^i \eta_-, \Gamma_{min})$ 
       $g_t^i \leftarrow 0$ 
    else
       $\eta_t^i \leftarrow \eta_{t-1}^i$ 
   $\theta_t \leftarrow \theta_{t-1} - \eta_t \text{sign}(g_t)$ 
   $g_{prev} \leftarrow g_t$ 

```

```

return  $\theta_t$ 

```

Εικόνα 2.9 Αλγόριθμος RPROP

2.7 Η μέθοδος ADAM

Η μέθοδος ADAM (*Adaptive Moment Estimation*), είναι ένας αλγόριθμος βελτιστοποίησης, ο οποίος χρησιμοποιείται αρκετά τα τελευταία χρόνια στα Βαθιά νευρωνικά δίκτυα (*Deep neural networks*), μιας και πρόκειται για αλγόριθμο αρκετά πρόσφατο. Δημοσιεύθηκε το 2014 από τους Diederik P. Kingma και Jimmy Lei Ba ως ερευνητικό έγγραφο με τίτλο “ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION” (Ba, 2014).

Η μέθοδος αυτή συνδυάζει δύο άλλες γνωστές μεθόδους βελτιστοποίησης νευρωνικών δικτύων για την υλοποίηση της, όπου είναι παραλλαγές της μεθόδου Stochastic Gradient Descent, και πιο συγκεκριμένα την RMSProp και την Momentum. Για να καταλάβουμε όμως πως ακριβώς επιτυγχάνονται τα τελικά αποτελέσματα από την

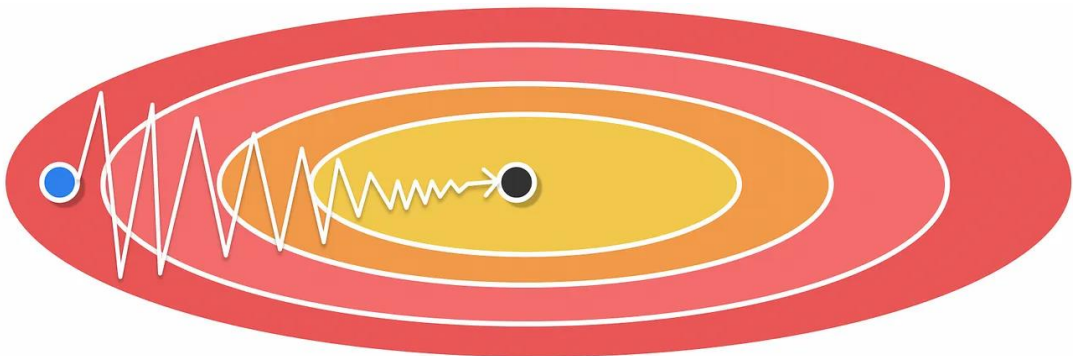
μέθοδο ADAM, πρέπει να καταλάβουμε τις βασικές λειτουργίες των δύο μεθόδων που αξιοποιεί για την υλοποίησή της. Ας ξεκινήσουμε με την μέθοδο Momentum.

Η μέθοδος Momentum καταφέρνει να εισάγει μία νέα φόρμουλα σε σύγκριση με την μέθοδο Gradient Descent, όπου καταγράφει προηγούμενες τιμές παραγώγων που έχουν γίνει, με σκοπό να χρησιμοποιηθούν κατά την ενημέρωση των βαρών (Efimov, 2023).

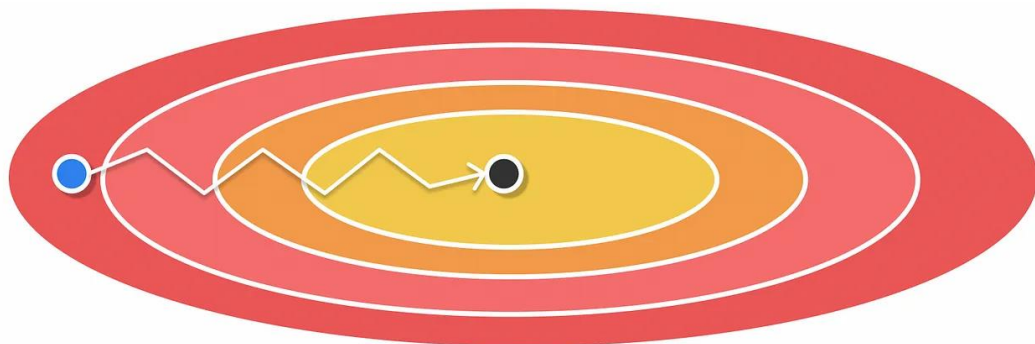
$$v_t = \beta v_{t-1} + (1 - \beta)dw_t$$
$$w_t = w_{t-1} - \alpha v_t$$

Εικόνα 2.10 Μαθηματικός τύπος Momentum

Με αυτό τον τρόπο καταφέρνουμε να κάνουμε λιγότερες άσκοπες αλλαγές στις κλίσεις και να ελαττώσουμε τον θόρυβο όπως μπορούμε να δούμε και στις παρακάτω φωτογραφίες. Έτσι επιτυγχάνονται καλύτερα και γρηγορότερα αποτελέσματα σε αντίθεση με την απλή μέθοδο Gradient Descent.



Εικόνα 2.12 Αποτέλεσμα με χρήση Gradient Descent



Εικόνα 2.11 Αποτέλεσμα με χρήση Momentum

Η μέθοδος RMSProp (*Root Mean Square Propagation*) η οποία θεωρείται μία αναβάθμιση της μεθόδου AdaGrad (*Adaptive Gradient Algorithm*), καταφέρνει να είναι καλύτερη στα τελικά της αποτελέσματα συγκριτικά με τον προκάτοχο και την ίδια στιγμή συνεχίζει να χρησιμοποιεί τις ίδιες φόρμουλες με την AdaGrad, με σκοπό την ενημέρωση των βαρών. Συγκεκριμένα χρησιμοποιεί όπως και η μέθοδος Momentum με την σειρά της, τη παράγωγο από προηγούμενες επαναλήψεις, αλλά αυτή την φορά εστιάζουμε στις τετραγωνικές τιμές (dw^2) και έπειτα το αποτέλεσμα της φόρμουλας χρησιμοποιείται για την ενημέρωση των βαρών, με την κύρια διαφορά να είναι ότι αντί να κάνουμε χρήση απλά του learning rate (α) κατά τον υπολογισμό των βαρών, το διαιρούμε με την ρίζα της φόρμουλας και εισάγουμε την μεταβλητή (ϵ), ώστε να αποτρέψουμε ενδεχόμενο διαιρέσεων με το 0.

$$v_t = \beta v_{t-1} + (1 - \beta) dw_t^2$$

$$w_t = w_{t-1} - \frac{\alpha}{\sqrt{v_t} + \epsilon} dw_t$$

Εικόνα 2.13 Μαθηματικός τύπος RPROP

Algorithm 1: *Adam*, our proposed algorithm for stochastic optimization. See section 2 for details, and for a slightly more efficient (but less clear) order of computation. g_t^2 indicates the elementwise square $g_t \odot g_t$. Good default settings for the tested machine learning problems are $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. All operations on vectors are element-wise. With β_1^t and β_2^t we denote β_1 and β_2 to the power t .

Require: α : Stepsize

Require: $\beta_1, \beta_2 \in [0, 1)$: Exponential decay rates for the moment estimates

Require: $f(\theta)$: Stochastic objective function with parameters θ

Require: θ_0 : Initial parameter vector

$m_0 \leftarrow 0$ (Initialize 1st moment vector)

$v_0 \leftarrow 0$ (Initialize 2nd moment vector)

$t \leftarrow 0$ (Initialize timestep)

while θ_t not converged **do**

$t \leftarrow t + 1$

$g_t \leftarrow \nabla_{\theta} f_t(\theta_{t-1})$ (Get gradients w.r.t. stochastic objective at timestep t)

$m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ (Update biased first moment estimate)

$v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$ (Update biased second raw moment estimate)

$\hat{m}_t \leftarrow m_t / (1 - \beta_1^t)$ (Compute bias-corrected first moment estimate)

$\hat{v}_t \leftarrow v_t / (1 - \beta_2^t)$ (Compute bias-corrected second raw moment estimate)

$\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon)$ (Update parameters)

end while

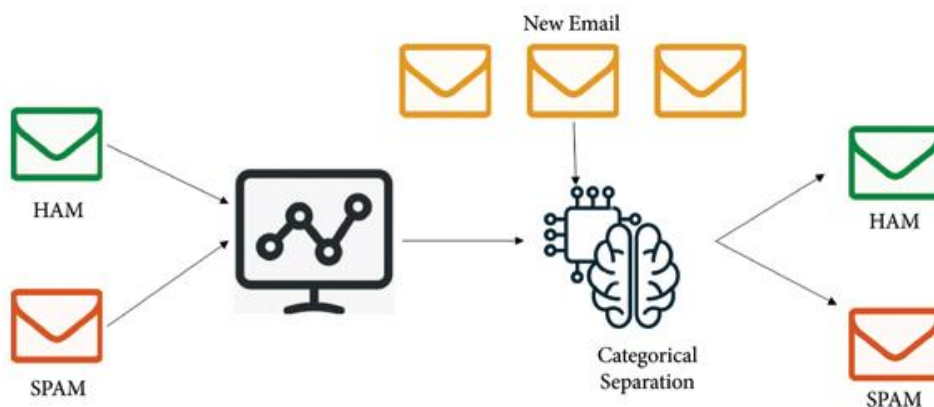
return θ_t (Resulting parameters)

Εικόνα 2.14 Αλγόριθμος ADAM

Έχοντας τις παραπάνω μεθόδους ως βάση της, η μέθοδος ADAM καταφέρνει να είναι μία από τις πιο διαδεδομένες μεθόδους βελτιστοποίησης στον τομέα των βαθιών νευρωνικών δικτύων τα τελευταία χρόνια και δικαίως, εφόσον καταφέρνει να χρησιμοποιεί τα καλύτερα χαρακτηριστικά της Momentum και RMSProp. Αυτό το αποτέλεσμα, γίνεται χάρης τον συνδυασμό των φόρμουλων των προηγούμενων δύο μεθόδων, όπου αξιοποιεί στον υπολογισμό της τις παραγώγους (*Momentum*) καθώς και τα τετράγωνα τους (*RMSProp*), κάνοντας τα τελικά αποτέλεσμα να έχουν μεγαλύτερη ακρίβεια ακόμα και από τις αρχικές επαναλήψεις. Παρακάτω μπορούμε να δούμε τον τρόπο υλοποίησης της μεθόδου όπως δημοσιεύθηκε στο ερευνητικό έγγραφο, καθώς και τις προτεινόμενες τιμές των μεταβλητών που χρησιμοποίησαν για τα καλύτερα αποτελέσματα.

2.7 Παραδείγματα εφαρμογής νευρωνικών δικτύων

Τα νευρωνικά δίκτυα παρ' όλο που είναι μία κατηγορία στον τομέα της πληροφορικής που για πολλούς θεωρείται αρκετά πρόσφατη, κυρίως λόγω της έντονης επιρροής που υπάρχει γύρω από τα μεγάλα λεκτικά μοντέλα (*LLM*), όπου έγιναν διαθέσιμα τους τελευταίους μήνες στο ευρύ κοινό, όπως το γνωστό μοντέλο ChatGPT το οποίο υλοποίησε η εταιρεία OpenAI το 2023 και αναβαθμίζει μέχρι και σήμερα. [<https://openai.com/chatgpt>] Στην πραγματικότητα, είναι ένας τομέας ο οποίος είναι μέρος της καθημερινότητας μας τα εδώ και αρκετά χρόνια, χωρίς ο κόσμος να το γνωρίζει άμεσα.



Εικόνα 2.15 Παράδειγμα εκπαίδευσης για τύπο email

Ξεκινώντας με ένα πολύ απλό παράδειγμα, τα email. Τα email αποτελούν ένα μέσω επικοινωνίας που είναι διαδεδομένο σε όλο τον κόσμο και είναι ο προτιμότερος τρόπος επικοινωνίας μεταξύ ατόμων, πολλών εταιρειών μεταξύ του προσωπικό, αλλά και των πελατών κάθε εταιρείας, καθώς και σε πολλαπλά σχολεία και δημόσιες υποδομές. Παρ' όλα αυτά, πάντα υπάρχει το ρίσκο κάποιος κακόβουλος χρήστης να προσπαθήσει να προωθήσει κάποιο email που εμπεριέχει περιεχόμενο που θα βλάψει τον χρήστη, είτε με την μορφή αρχείων εντός του email, τα λεγόμενα συνημμένα αρχεία (π.χ. έγγραφα pdf, εικόνες, εκτελέσιμα αρχεία, κλπ.), αλλά και με την χρήση συνδέσμων (*URLs*), όπου μπορεί να ζητάνε από τον χρήστη να εισάγει τα προσωπικά του στοιχεία σε κάποιον πεδίο με στόχο να καταφέρει ο κακόβουλος χρήστης να τα υποκλέψει. Οι πάροχοι email κατατάζουν τα email σε δύο κατηγορίες, στην κατηγορία Ham που αφορά τα γνήσια email και τα spam email που αφορούν τα κακόβουλα. Οι διαθέσιμοι τρόποι για να καταφέρνουν οι πάροχοι να εντοπίσουν σε ποια κατηγορία ανήκει ένα email είναι αρκετοί και μέσα σε αυτούς υπάρχει και η χρήση νευρωνικού δικτύου το οποίο μπορεί να εκπαιδευτεί με διάφορες τεχνικές, όπως είναι η εκπαίδευση με επίβλεψη πάνω σε πολλαπλά κατηγοριοποιημένα email, ham ή spam, όπου έχει ως στόχο να μπορεί να εντοπίζει τα κοινά σημεία που έχουν τα spam emails και να μπορεί να προβλέπει πότε ένα email που θα λαμβάνει ο χρήστης είναι ανεπιθύμητο και πότε όχι, κάνοντας την διαδικασία αρκετά πιο αυτοματοποιημένη, χωρίς να χρειάζεται ανθρώπινο δυναμικό με σκοπό να επιβλέπει χειροκίνητα έναν τεράστιο όγκο email, κάτι που στις μέρες μας θα είναι αδύνατο. (Naeem Ahmed, 2022)

Επόμενο παράδειγμα είναι ο τομέας της ιατρικής. Τα τελευταία χρόνια τα τεχνητά νευρωνικά δίκτυα εκπαιδεύονται πάνω στον τομέα της ιατρικής εκμεταλλευόμενα τον τεράστιο όγκο πληροφοριών που υπάρχει διαθέσιμος (φωτογραφίες, αιματολογικές εξετάσεις, διαγνωστικές εξετάσεις, κλπ.), πάνω σε πολλαπλά ιατρικά θέματα ανά τα χρόνια, με σκοπό την ευκολότερη και γρήγορη διάγνωση αυτών των προβλημάτων, με ελάχιστο ποσοστό λάθους, αλλά και με το ενδεχόμενο εντοπισμού αυτών των προβλημάτων σε αρκετά πρόωρο στάδιο, όπως είναι ο καρκίνος του παχέος εντέρου, γυναικολογικές παθήσεις, προδιάθεση ζαχάρου, και πολλά άλλα (Filippo Amato, 2013).

Τέλος ένα νευρωνικό δίκτυο χρησιμοποιείται σε πάρα πολλές εφαρμογές καμερών που έρχονται προ εγκαταστημένες σε κινητά τηλέφωνα. Σκοπός του νευρωνικού είναι διάφορες λειτουργίες, όπως ο εντοπισμός προσώπου σε μία φωτογραφία, ώστε να μπορεί

να εισάγει τα λεγόμενα φίλτρα τα οποία παρέχουν ακόμα και αρκετά κοινωνικά δίκτυα όπως είναι τα γνωστά Instagram της εταιρείας Meta και Snapchat της εταιρείας Snap, όπου είναι υπεύθυνα για την προσθήκη επιπλέον χαρακτηριστικών πάνω στο πρόσωπο του χρήστη με σκοπό την βελτίωση της εικόνας. Άλλη μία χρήσιμη εφαρμογή τους είναι να ενισχύει την ένταση φωτεινότητας, χρωμάτων, της ελάττωσης θορύβου και θολών σημείων στην εικόνα, καθώς ακόμα και να προσπαθήσει να αυξήσει την συνολική ποιότητα της εικόνας (Luca, 2023).

3 ΚΕΦΑΛΑΙΟ 3

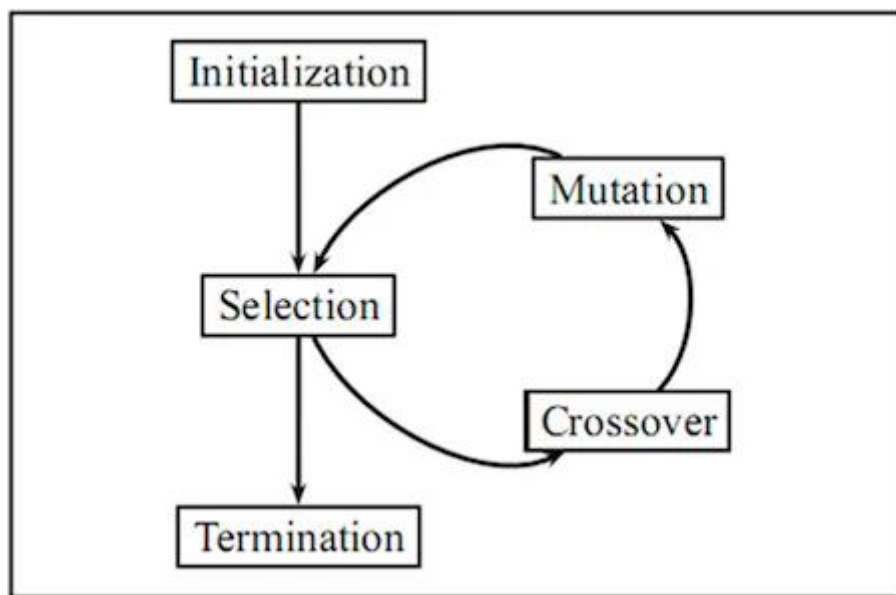
3.1 Γραμματική εξέλιξη (Grammatical Evolution)

Η γραμματική εξέλιξη (*Grammatical Evolution*) είναι ένα νέο σχετικά είδος αλγορίθμου εξέλιξης από τον χώρο του Γενετικού προγραμματισμού (*Genetic Programming*), όπου συνδυάζει τις τεχνικές από τους γενετικούς αλγορίθμους (*Genetic Algorithms*) και της γραμματικής χωρίς συμφραζόμενα (*context-free grammars*) όπως είναι η γνωστή Backus–Naur Form (*BNF*). Το τελικό αποτέλεσμα που προσφέρει, είναι ένα ολοκληρωμένο σύστημα, το οποίο έχει την δυνατότητα να παράγει προγράμματα σε οποιαδήποτε γλώσσα προγραμματισμού θελήσουμε με την βοήθεια της γραμματικής χωρίς συμφραζόμενα και να χρησιμοποιηθεί για την επίλυση προβλημάτων και την βελτιστοποίηση συναρτήσεων (O'Neill, 2017), (Wikipedia, 2024), (Adam, 2017). Για να μπορέσουμε όμως να καταλάβουμε πως ακριβώς λειτουργεί, χρειάζεται να κατανοήσουμε τους Εξελικτικούς αλγορίθμους, και τα παρακλάδια του όπως είναι οι Γενετικοί αλγόριθμοι.

Ξεκινώντας με τους Εξελικτικούς αλγορίθμους (*Evolutionary algorithm*), όπου είναι μία κατηγορία ευρετικών αλγορίθμων όπου βασίζονται στην ιδέα της εξέλιξης στην φύση όπως είναι η αναπαραγωγή, η μετάλλαξη, και στην επιβίωση του ισχυρότερου και διάφορες άλλες λειτουργίες που υπάρχουν και έχουν δυνατότητες επίλυσης περίπλοκων προβλημάτων και βελτιστοποιήσεων, συγκριτικά με τους γνωστούς αλγορίθμους και μεθόδους που χρησιμοποιούνται στα νευρωνικά δίκτυα. Ο τρόπος που το επιτυγχάνουν είναι μέσω της διαφορετικής προσέγγισης που έχουν για την επίλυση προβλημάτων, την οποία θα δούμε σε μορφή βημάτων (Soni, 2018), (Wikipedia, 2024):

- Αρχικοποιούμε έναν πληθυσμό από πιθανές λύσεις, όπου κάθε λύση εμπεριέχει σαν μορφή περιγραφής μία σειρά από δυαδικούς αριθμούς.
- Έπειτα με την χρήση μίας συνάρτησης που ονομάζεται *fitness function*, ελέγχουμε κάθε πιθανή λύση και της αναθέτουμε έναν αριθμό με βάση του πόσο κοντά ήταν στο επιθυμητό αποτέλεσμα. Το πόσο μεγάλος η μικρό θα είναι ο αριθμός βασίζεται καθαρά στο πόσο σωστή ήταν η λύση.

- Εφόσον ολοκληρωθούν οι έλεγχοι, επιλέγουμε τις καλύτερες πιθανές λύσεις βασιζόμενοι στην ιδέα του Δαρβίνου, την επιβίωση του ισχυρότερου (*survival of the fittest*) (Wikipedia, 2024), με σκοπό την αναπαραγωγή.
- Στο στάδιο της αναπαραγωγής χρησιμοποιούμε τους γενετικούς τελεστές που θα δούμε αναλυτικά σε ξεχωριστό υπό κεφάλαιο, ώστε να δημιουργήσουμε νέες λύσεις από τις καλύτερες που έχουμε.
- Όταν ολοκληρωθούν όλα τα προηγούμενα βήματα έχουμε μία γενιά από λύσεις, και προχωράμε στην επόμενη αν και εφόσον τα κριτήρια τερματισμού δεν ισχύουν.



Εικόνα 3.1 Διάγραμμα διαδικασίας Γενετικών αλγορίθμων

Παρόμοια λογική ακολουθούν και οι Γενετικοί αλγόριθμοι όπου έχουν τα ίδια βήματα για την εύρεση της λύση με μοναδική διαφορά το αρχικό βήμα της διαδικασίας, όπου αντί για την αρχικοποίηση ενός γενικού πληθυσμού από λύσεις, βασίζεται στην ιδέα των χρωμοσωμάτων, όπου παρουσιάζονται σαν μία σειρά από ακέραιες τιμές.

Φτάνοντας στην Γραμματική εξέλιξη η χρήση της BNF σε συνδυασμό με την αναπαράσταση των χρωμοσωμάτων σαν μία σειρά από ακεραίους, βοηθάει στην περιγραφή η οποία

Σε αυτό το κεφάλαιο θα δούμε την ιστορία πίσω από τους γενετικούς αλγορίθμους την εξέλιξη τους ανά τα χρόνια και πως έφτασαν στην τωρινή τους μορφή, θα αναλύσουμε την χρήση των γενετικών τελεστών πάνω στους διάφορους Εξελικτικούς αλγορίθμους,

όπου χρησιμοποιούνται για την εύρεση του καλύτερου τελικού αποτελέσματος, και η ανάλυση τους θα γίνει με την χρήση ενός γνωστού παραδείγματος με όνομα πρόβλημα σακιδίου, καθώς και τις διάφορες εφαρμογές που έχουν οι αλγόριθμοι γραμματικής εξέλιξης στις μέρες μας.

3.2 Ιστορική αναδρομή

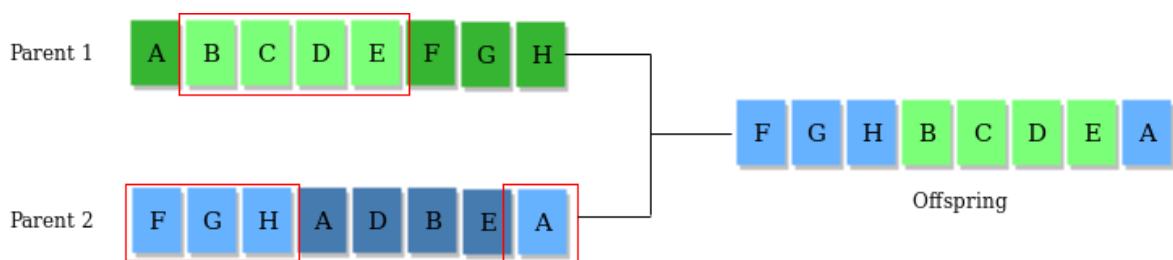
Η ιδέα των γενετικών αλγορίθμων και οι έρευνες πάνω στην εξερεύνηση αυτού του είδους αλγορίθμων ξεκινάει πίσω το 1950, όπου ο Άγγλος μαθηματικός και θεωρητικός βιολόγος Άλαν Τούρινγκ (*Alan Matheson Turing*) πρότεινε στην έρευνα του με τίτλο “Computing Machinery and Intelligence” (Turing, 1950) το ενδεχόμενο μίας μηχανής η οποία θα έχει την δυνατότητα να μαθαίνει σταδιακά με βάση την εμπειρία που θα αποκτάει όσο περνάει ο καιρός, βασισμένος από την ιδέα της εξέλιξης. Η ιδέα του άρχισε να ερευνάτε σε βάθος από τον μαθηματικό Nils Aall Barricelli το 1954 και η έρευνα του έθεσε τα θεμέλια για τις έρευνες του Alex Fraser το 1957, όπου κατάφεραν με τις έρευνες τους να προσομοιώσουν την εξέλιξη μέσω υπολογιστή, και να ξεκινήσουν μία σειρά από μελλοντικές έρευνες όπου θα είναι η σημερινή μορφή των γενετικών αλγορίθμων όπως τους ξέρουμε. Περνώντας στις δεκαετίες του 1960 και 1970 όλες οι παραπάνω έρευνες κατάφεραν να εμπνεύσουν τους Ingo Rechenberg και Hans-Paul Schwefel να υλοποιήσουν την πρώτη μέθοδο βελτιστοποίησης με στόχο της επίλυση περίπλοκων προβλημάτων μηχανικής. Το 1970 δημοσιεύθηκαν αρκετά ερευνητικά έγγραφα καθώς και βιβλία, ένα από αυτά είναι του Fraser Alex και Burnell Donald με τίτλο “*Computer Models in Genetics*”, όπου προτείνουν πολλές από τις μεθόδους που χρησιμοποιούνται και στις μέρες μας. Παράλληλα ο Lawrence J. Fogel πρότεινε την χρήση του Εξελικτικού προγραμματισμού για τεχνητή νοημοσύνη. Φτάνοντας στα μέσα του 1980 όπου όλες οι παραπάνω θεωρητικές ιδέες θα ξεκινήσουν να ελκύουν το ευρύ κοινό και να γίνονται πράξη χάρις την πρώτη διεθνή συνάντηση για γενετικούς αλγορίθμους που έγινε στο Πίτσμπεργκ της Πενσυλβανίας (Wikipedia, 2024).

Αντίστοιχα η ιστορία της γραμματικής εξέλιξης ξεκινάει λίγα χρόνια μετά, το 1998, όπου ο Conor Ryan, JJ Collins και Michael O'Neill από την ομάδα BDS (*Biocomputing and Development Systems*) του Πανεπιστημίου του Λίμερικ, όπου πρότειναν την πρώτη υλοποίηση του αλγορίθμου βασισμένοι στις τεχνικές της γραμματικής χωρίς συμφραζόμενα (Wikipedia, 2024).

3.3 Γενετικοί τελεστές

Το είδος της Γραμματικής Εξέλιξης είναι στην ουσία ένας ακόμα γενετικός αλγόριθμος, κάνοντας έτσι εφικτό την χρήση των γενετικών τελεστών. Οι γενετικοί τελεστές αποτελούνται από τις συναρτήσεις επιλογής (*Selection function*), της διασταύρωση (*Crossover function*) και της μετάλλαξη (*Mutation function*). Για την καλύτερη κατανόηση και επεξήγηση των παραπάνω τελεστών, θα αναφέρουμε ένα γνωστό πρόβλημα ως παράδειγμα, το οποίο χρησιμοποιείται στην δοκιμή διάφορων αλγορίθμων και μελετιέται εδώ και πολλά χρόνια. Το πρόβλημα ονομάζεται, πρόβλημα σακιδίου (*Knapsack problem*), όπου έχουμε έναν δεδομένο αριθμό αντικειμένων και μεγέθους που μπορούμε να διαλέξουμε, αλλά παράλληλα υπάρχει περιορισμένος χώρος εντός του σακιδίου. Στόχος είναι να βρεθεί ο καλύτερος εφικτός συνδυασμός αντικειμένων όπου θα χωρέσει στο σακίδιο (Wikipedia, 2024).

Θα ξεκινήσουμε το παράδειγμα από το σημείο όπου είδη έχουμε με την βοήθεια της fitness function τις καλύτερες λύσεις αντικειμένων για αυτή την γενιά. Από αυτό το σημείο αρχίζουμε να διαλέγουμε από τον πληθυσμό συνήθως τυχαία, ένα ζευγάρι λύσεων από την τωρινή γενιά. Στην συνέχεια έχοντας επιλέξει τις καλύτερες λύσεις, διαλέγουμε ένα τυχαίο μέρος για τις δύο λύσεις και τις διαχωρίζουμε σε δύο κοινά μέρη, από εκεί παίρνουμε το τελικό μέρος από κάθε λύση και το αλλάζουμε μεταξύ των δύο, αυτή η διαδικασία είναι γνωστή ως διασταύρωση ενός σημείου (*single point crossover*) (Wikipedia, 2024), (GeeksforGeeks, 2023), με αποτέλεσμα να μας επιτρέπει να παράγουμε δύο νέες λύσεις για την επόμενη γενιά.



Εικόνα 3.2 Βήμα Mutation

Αυτή η διαδικασία συνεχίζεται μέχρι να μην έχουμε άλλα ζευγάρια λύσεων για κάνουμε διασταύρωση μεταξύ τους. Τέλος πριν μεταφέρουμε τις καλύτερες λύσεις στην επόμενη γενιά, εντάσσουμε στην διαδικασία τον τελεστή της μετάλλαξη, η οποία στοχεύει στην

αλλαγή μίας μονάχα τυχαίας δυαδικής τιμής σε κάθε γονιδίωμα, με σκοπό την ανακάλυψη νέων λύσεων όπου προηγουμένως δεν θα ήταν εφικτό, αλλά παράλληλα περιορίζει το ενδεχόμενο πρόωρης σύγκλισης.



Εικόνα 3.3 Αποτέλεσμα βήματος Mutation

Ένας ακόμα παράγοντας ο οποίος επηρεάζει την διαδικασία με τον ίδιο τρόπο που το επιτυγχάνουν οι γενετικοί τελεστές είναι η χρήση του ελιτισμού, όπου εστιάζει στο να επιλέξει απευθείας τις καλύτερες λύσεις από την γενιά και να τις μεταφέρει στην επόμενη.

3.4 Εφαρμογές γραμματικής εξέλιξης

Οι εφαρμογές της γραμματικής εξέλιξης σε προβλήματα της καθημερινότητας όλο και αυξάνονται με τα χρόνια. Στις μέρες μας υλοποιήσεις με την χρήση της γραμματικής εξέλιξης προσφέρουν μεταξύ άλλων την δυνατότητα διάγνωση του καρκίνου του μαστού, καθώς και στην πρόβλεψη κρουσμάτων του ιού COVID-19 (Ioannis G. Tsoulos, 2022), στον τομέα της ιατρικής (Yumnah Hasan, 2024). Στην καταγραφή των ανθρώπινων δραστηριοτήτων με σκοπό την ελαχιστοποίηση της κατανάλωσης μπαταρίας σε συσκευές καταγραφής υγείας και φυσικής κατάστασης σε μορφή ρολογιού (Sungwon Yang, 2010). Στην δημιουργία νέων τρόπων κατασκευής γεννήτριας παραγωγής ψευδοτυχαίων αριθμών και κρυπτογράφησης δεδομένων (Meghana Kshirsagar, 2022) και πολλά άλλα.

4 ΚΕΦΑΛΑΙΟ 4

4.1 Κατασκευαζόμενα τεχνητά νευρωνικά δίκτυα

Στο τελικό κεφάλαιο αυτής της πτυχιακής εργασίας, θα δούμε και θα αναλύσουμε την προτεινόμενη μέθοδο κατασκευής και εκπαίδευσης ενός τεχνητού νευρωνικού δικτύου χρησιμοποιώντας μία από τις πιο γνωστές τεχνικές, την Γραμματική εξέλιξη και της γραμματικής BNF, καθώς και θα συγκρίνουμε την προτεινόμενη μέθοδο τοπικής ελαχιστοποίησης που παρέχετε με το πρόγραμμα NNC, η οποία είναι η Broyden–Fletcher–Goldfarb–Shanno (BFGS) με μερικές γνωστές εναλλακτικές όπως είναι η μέθοδος ADAM και RPROP τις οποίες και αναλύσαμε σε προηγούμενη ενότητα. Τέλος θα κλείσουμε με τα συμπεράσματα αυτής της έρευνας καθώς και με τις δυνατότητες επέκτασης και αναβάθμισης που θα μπορέσει να δεχτεί το πακέτο NNC στο μέλλον.

Ξεκινώντας αυτό το κεφάλαιο θα μιλήσουμε πρώτα για την διεπαφή την οποία χρησιμοποιεί το πρόγραμμα NNC. Συγκεκριμένα το πρόγραμμα όπως έχουμε αναφέρει και σε προηγούμενο κεφάλαιο είναι γραμμένο σε γλώσσα προγραμματισμού ANSI C++, και χρησιμοποιεί την γνωστή βιβλιοθήκη ανοιχτού κώδικα QT, δίνοντας του την δυνατότητα συμβατότητας με πολλαπλά λειτουργικά συστήματα. Όταν ο χρήστης ξεκινήσει την διαδικασία να χτίσει το πρόγραμμα από τον πηγαίο κώδικα του, θα βρει ένα νέο εκτελέσιμο αρχείο με όνομα NNC, το οποίο μπορεί να τρέξει μέσα από οποιοδήποτε τερματικό. Το αποτέλεσμα όταν ο χρήστης δοκιμάσει την εφαρμογή είναι το παρακάτω.

```

./NNC -h
Print this help screen.
-p Specify problem file.
-c Specify population size.
-l Specify genome length.
-s Specify selection rate.
-m Specify mutation rate.
-r Specify random seed.
-k Specify kind of the network. (neural,ode,pde,sode,kdv)
-o Specify plot file.
-n Specify maximum number of generations.
-g Specify local search generations
-d Specify local search chromosomes
-t Specify test file.

```

Εικόνα 4.1 Διεπαφή NNC

Το πρόγραμμα λειτουργεί με την χρήση επιλογών όπως είναι συνηθισμένο σε τέτοιες μορφής εφαρμογές και παρέχει πολλές επιλογές όπως είναι η -t για να επιλέξουμε ένα συγκεκριμένο αρχείο για test ή το -s για να θέσουμε μία συγκεκριμένη τιμή ως selection rate, και αρκετές ακόμα. Αν και εφόσον ο χρήστης έχει στα χέρια του ένα παράδειγμα τύπου regression, μπορεί να τρέξει την επόμενη εντολή,

```
./NNC -p BK.train -t BK.train -n 20 -o output.txt
```

η οποία θα χρησιμοποιήσει το αρχείο BK.train ως αρχείο train με την χρήση της επιλογής -t, και με τον ίδιο τρόπο θα επιλέξουμε το test αρχείο, τον αριθμό μέγιστων γενιών καθώς και ένα αρχείο το οποίο θα αποθηκεύσει τα τελικά αποτελέσματα που κατάφερε να βγάλει το μοντέλο. Όλη η διαδικασία εκτέλεσης και υπολογισμού καθώς και τα τελικά αποτελέσματα, φαίνονται αναλυτικά στην παρακάτω φωτογραφία.

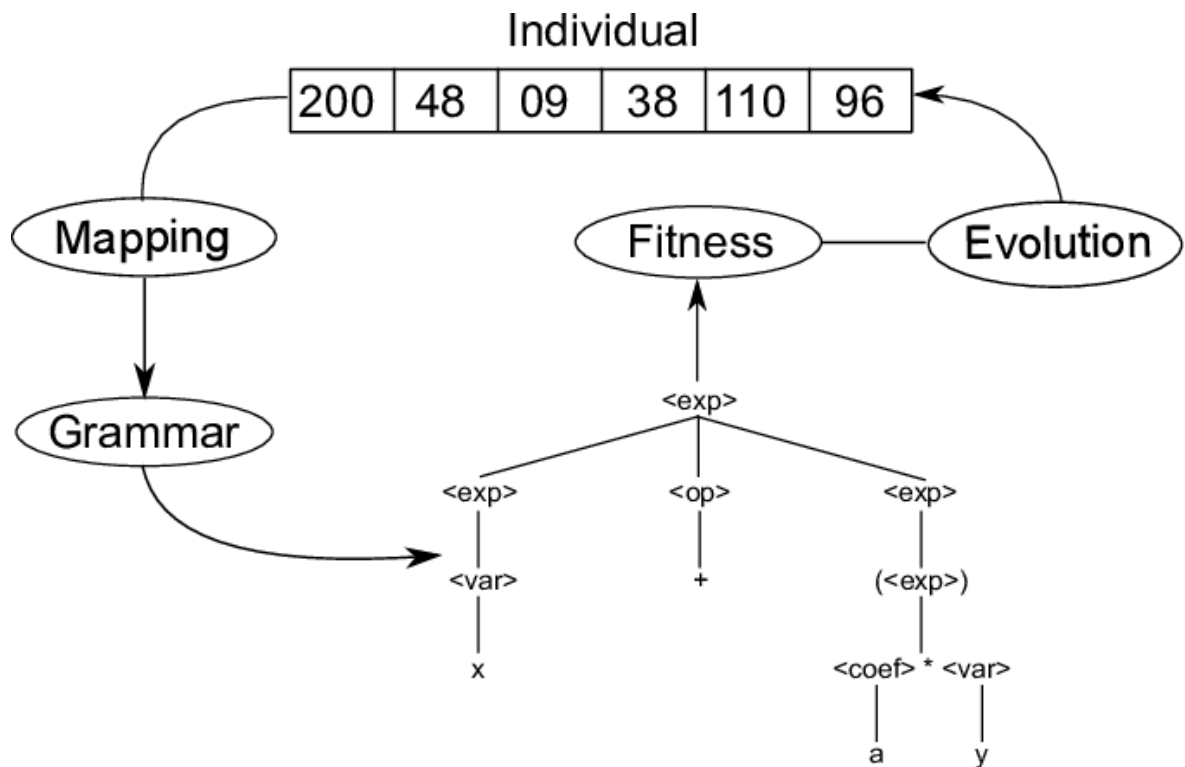
```

Terminal - box@virtualbox:~/Downloads/Optimizers/Tolmin-Optimizer/NNC-untouched
0:376.723989 10:376.723989 20:376.723989 30:376.723989 40:376.723989 50:376.723989 60:376.723989 70:376.723989 80:376.723989 90:376.723989 100:376.723989 110:376.723989 120:376.723989 130:376.723989 140:376.723989 150:376.723989 160:376.7239
89 170:376.723989 180:376.723989 190:376.723989 200:376.723989 210:376.723989 220:376.723989 230:376.723989 240:376.723989 250:376.723989 260:376.723989 270:376.723989 280:376.723989 290:376.723989 300:376.723989 310:376.723989 320:376.7239
9 330:376.723989 340:376.723989 350:376.723989 360:376.723989 370:376.723989 380:376.723989 390:376.723989 400:376.723989 410:376.723989 420:376.723989 430:376.723989 440:376.723989 450:376.723989 460:376.723989 470:376.723989 480:376.723989
490:376.723989
495 -376.723989 (0.2870)+sig((0.0214,571)+x1+(-75.82)+x15+(-676.5))+(0.99)+sig((0.71)+x5+(-7.0))+(-68.1)+sig((-0.624)+x11+(-1.0))
0:376.723989 10:376.723989 20:376.723989 30:376.723989 40:376.723989 50:376.723989 60:376.723989 70:376.723989 80:376.723989 90:376.723989 100:376.723989 110:376.723989 120:376.723989 130:376.723989 140:376.723989 150:376.723989 160:376.7239
89 170:376.723989 180:376.723989 190:376.723989 200:376.723989 210:376.723989 220:376.723989 230:376.723989 240:376.723989 250:376.723989 260:376.723989 270:376.723989 280:376.723989 290:376.723989 300:376.723989 310:376.723989 320:376.7239
9 330:376.723989 340:376.723989 350:376.723989 360:376.723989 370:376.723989 380:376.723989 390:376.723989 400:376.723989 410:376.723989 420:376.723989 430:376.723989 440:376.723989 450:376.723989 460:376.723989 470:376.723989 480:376.723989
490:376.723989
495 -376.723989 (0.2870)+sig((0.0214,571)+x1+(-75.82)+x15+(-676.5))+(0.99)+sig((0.71)+x5+(-7.0))+(-68.1)+sig((-0.624)+x11+(-1.0))
0:376.723989 10:376.723989 20:376.723989 30:376.723989 40:376.723989 50:376.723989 60:376.723989 70:376.723989 80:376.723989 90:376.723989 100:376.723989 110:376.723989 120:376.723989 130:376.723989 140:376.723989 150:376.723989 160:376.7239
89 170:376.723989 180:376.723989 190:376.723989 200:376.723989 210:376.723989 220:376.723989 230:376.723989 240:376.723989 250:376.723989 260:376.723989 270:376.723989 280:376.723989 290:376.723989 300:376.723989 310:376.723989 320:376.7239
9 330:376.723989 340:376.723989 350:376.723989 360:376.723989 370:376.723989 380:376.723989 390:376.723989 400:376.723989 410:376.723989 420:376.723989 430:376.723989 440:376.723989 450:376.723989 460:376.723989 470:376.723989 480:376.723989
490:376.723989
495 -376.723989 (0.2870)+sig((0.0214,571)+x1+(-75.82)+x15+(-676.5))+(0.99)+sig((0.71)+x5+(-7.0))+(-68.1)+sig((-0.624)+x11+(-1.0))
0:376.723989 10:376.723989 20:376.723989 30:376.723989 40:376.723989 50:376.723989 60:376.723989 70:376.723989 80:376.723989 90:376.723989 100:376.723989 110:376.723989 120:376.723989 130:376.723989 140:376.723989 150:376.723989 160:376.7239
89 170:376.723989 180:376.723989 190:376.723989 200:376.723989 210:376.723989 220:376.723989 230:376.723989 240:376.723989 250:376.723989 260:376.723989 270:376.723989 280:376.723989 290:376.723989 300:376.723989 310:376.723989 320:376.7239
9 330:376.723989 340:376.723989 350:376.723989 360:376.723989 370:376.723989 380:376.723989 390:376.723989 400:376.723989 410:376.723989 420:376.723989 430:376.723989 440:376.723989 450:376.723989 460:376.723989 470:376.723989 480:376.723989
490:376.723989
495 -376.723989 (0.2870)+sig((0.0214,571)+x1+(-75.82)+x15+(-676.5))+(0.99)+sig((0.71)+x5+(-7.0))+(-68.1)+sig((-0.624)+x11+(-1.0))
0:376.723989 10:376.723989 20:376.723989 30:376.723989 40:376.723989 50:376.723989 60:376.723989 70:376.723989 80:376.723989 90:376.723989 100:376.723989 110:376.723989 120:376.723989 130:376.723989 140:376.723989 150:376.723989 160:376.7239
89 170:376.723989 180:376.723989 190:376.723989 200:376.723989 210:376.723989 220:376.723989 230:376.723989 240:376.723989 250:376.723989 260:376.723989 270:376.723989 280:376.723989 290:376.723989 300:376.723989 310:376.723989 320:376.7239
9 330:376.723989 340:376.723989 350:376.723989 360:376.723989 370:376.723989 380:376.723989 390:376.723989 400:376.723989 410:376.723989 420:376.723989 430:376.723989 440:376.723989 450:376.723989 460:376.723989 470:376.723989 480:376.723989
490:376.723989
495 -376.723989 (0.2870)+sig((0.0214,571)+x1+(-75.82)+x15+(-676.5))+(0.99)+sig((0.71)+x5+(-7.0))+(-68.1)+sig((-0.624)+x11+(-1.0))
TRAIN ERROR =0.02, 0.02486509
TEST ERROR =22.2632948000
CLASS ERROR=91.158
SOLUTION: y(x)=(0.2870)+sig((0.0214,571)+x1+(-75.82)+x15+(-676.5))+(0.99)+sig((0.71)+x5+(-7.0))+(-68.1)+sig((-0.624)+x11+(-1.0))

```

Εικόνα 4.2 Τελικό αποτέλεσμα NNC

Συνεχίζοντας θα μιλήσουμε για τον τρόπο με τον οποίο η προτεινόμενη μέθοδος λειτουργεί. Αρχικά η μέθοδος βασίζεται πάνω στην ιδέα της Γραμματικής εξέλιξης, όπου για να κατασκευάσει το τεχνητό νευρωνικό δίκτυο το οποίο θα μπορεί να δεχτεί τα δεδομένα για τους υπολογισμούς, χρησιμοποιεί μία γλώσσα τύπου C, όπως η γνωστή γλώσσα προγραμματισμού, για την διαδικασία και βασίζεται στην γραμματική BNF για την περιγραφή της δομής του νευρωνικού δικτύου. Έτσι κάθε χρωμόσωμα περιγράφεται σαν μία σειρά από ακέραιους αριθμούς οι οποίοι συσχετίζονται με τον ανάλογο κανόνα παραγωγής (*production rule*), ώστε να δημιουργήσουν την δομή της γραμματικής βήμα βήμα, η οποία θα κατασκευάσει το νευρωνικό δίκτυο.



Εικόνα 4.3 Διάγραμμα Γραμματικής εξέλιξης

4.2 Παραλλαγές κατασκευαζόμενων τεχνητών νευρωνικών δικτύων

Υπάρχουν αρκετές παραλλαγές κατασκευαζόμενων τεχνητών νευρωνικών δικτύων. Τα πρώτα και πιο γνωστά είναι τα feedforward νευρωνικά δίκτυα. Ο τρόπος λειτουργίας τους όπου και βασίζεται το όνομα τους, είναι ότι εισάγουν τα δεδομένα από μία σταθερή κατεύθυνση, στην περίπτωση ενός τυπικού μοντέλο ξεκινάει από την είσοδο προς τους κρυφούς κόμβους μέχρι να φτάσουν στον τελικό κόμβο της εξόδου και να εξάγουν το τελικό αποτέλεσμα. Ένα παράδειγμα feedforward μοντέλου αποτελεί το μοντέλο Perceptron που είναι και από τα πιο απλά στο είδος του (Monsters, 2017). Επόμενο είδος κατασκευαζόμενου τεχνητού νευρωνικού δικτύου είναι το Recursive neural network, το οποίο αποτελεί είδος βαθιού νευρωνικού δικτύου όπου χρησιμοποιεί την λογική να συνδυάζει αναδρομικά τις πληροφορίες κοντινών δεδομένων ώστε να εξάγει το τελικό αποτέλεσμα χρησιμοποιώντας κοινή τιμή βάρους σε όλες τις εισόδους (Simplilearn, 2023). Άλλο ένα είδος νευρωνικού δικτύου το οποίο είναι κοντά στο είδος του Recursive neural network, είναι το Recurrent neural network. Ο λόγος όπου είναι σχετικά κοντά, βρίσκεται πίσω στην δυνατότητα διαδοχικής επεξεργασία των δεδομένων κάνοντας αυτά τα είδη αρκετά πιο αποδοτικά συγκριτικά με τα feedforward νευρωνικά. Ο τρόπος λειτουργίας του Recurrent νευρωνικού είναι ότι μπορεί να κάνει καταγραφεί σε μία μνήμη τις πληροφορίες που δέχεται και να μπορεί να χρησιμοποιεί αυτή την μνήμη σαν παράγοντα που θα επηρεάζει τις μελλοντικές τιμές που θα λαμβάνει, δίνοντας του την δυνατότητα να καταγράφει μοτίβα από αυτές τις τιμές (Simplilearn, 2023). Τέλος υπάρχουν τα Transformers νευρωνικά δίκτυα τα οποία είναι και τα πιο πρόσφατα στην λίστα. Παρουσιάστηκαν το 2017 στο ερευνητικό έγγραφο με τίτλο “*Attention Is All You Need*” (Ashish Vaswani, 2017) και έκανε αναφορά σε μία μέθοδο όπου θα επεξεργάζεται τα διαδοχικά δεδομένα έχοντας την δυνατότητα να παρακολουθεί ταυτόχρονα όλα τα στοιχεία. Θα κωδικοποιεί τις ακολουθίες εισόδου, θέτοντας σε κάθε στοιχείο από ένα σχετικό βάρος βασισμένο στην σχέση του στοιχείου με άλλα. Με αυτό τον τρόπο καταφέρνουν να παράγουν πολύ καλύτερα τελικά αποτελέσματα, καθώς και να υποστηρίζουν πιο περίπλοκες σχέσεις (Ankit, 2022).

4.3 Ο προτεινόμενος αλγόριθμος

Ο προτεινόμενος αλγόριθμος μπορεί να χωριστεί σε 5 βήματα τα οποία αποτελούνται από την αρχικοποίηση των μεταβλητών όπως είναι ο μέγιστος αριθμός των γενιών και ο αριθμός των χρωμοσωμάτων εντός ενός πληθυσμού. Η διαδικασία που αφορά το γενετικό κομμάτι του αλγορίθμου. Το σημείο όπου εντάσσουμε τεχνική τοπικής

ελαχιστοποίησης βασισμένοι από τα αποτελέσματα του προηγούμενου βήματος. Και τέλος το σημείο όπου κατασκευάζουμε ένα νευρωνικό δίκτυο με βάση τα αποτελέσματα και εξάγουμε την πρόβλεψη με βάση το πρόβλημα που εισάγουμε στο NNC. Όλη η παραπάνω διαδικασία μπορεί να περιγραφεί πιο αναλυτικά με την βοήθεια ψευδοκώδικα όπως αναγράφεται και στο ερευνητικό έγγραφο όπως θα δούμε παρακάτω.

1. Βήμα αρχικοποίησης μεταβλητών:

- 1.1.Θέτουμε μία μεταβλητή N_G , όπου αφορά τον μέγιστο αριθμό γενιών και μία μεταβλητή N_C για τον αριθμό χρωμοσωμάτων για τον πληθυσμό.
- 1.2.Θέτουμε μία μεταβλητή P_S , ως ποσοστό επιλογής (*selection rate*) και αντίστοιχα P_m για ποσοστό μετάλλαξης (*mutation rate*).
- 1.3.Θέτουμε την μεταβλητή L_I , ως τον αριθμό γενιών που θα πρέπει να γίνουν πριν εισάγουμε την μέθοδο τοπικής ελαχιστοποίησης.
- 1.4.Θέτουμε την μεταβλητή L_C , ως τον αριθμό χρωμοσωμάτων που θα χρησιμοποιηθούν στην διαδικασία της τοπικής ελαχιστοποίησης.
- 1.5.Αρχικοποιούμε τα χρωμοσώματα σαν τυχαία διανύσματα ακεραίων.
- 1.6.Αρχικοποιούμε μία μεταβλητή $iter = 0$, την οποία χρησιμοποιούμε για να καταγράφουμε τον αριθμό των επαναλήψεων.

2. Γενετικό βήμα:

2.1.Για κάθε $i = 1, \dots, Ng$ κάνε

- a) Δημιούργησε για κάθε χρωμόσωμα ένα νευρωνικό δίκτυο C_i με χρήση γραμματικής εξέλιξης.
- b) Υπολόγισε την “φυσική κατάσταση” f_i για κάθε χρωμόσωμα, με βάση τον τύπο του προβλήματος (διαφορική εξίσωση, regression ή classification) και κάθε χρωμόσωμα μετατρέπεται με την διαδικασία της γραμματικής εξέλιξης σε ένα τεχνητό νευρωνικό δίκτυο με ένα κρυφό επίπεδο (*hidden layer*) και μία έξοδο (*output layer*).

- c) Εισάγουμε στην διαδικασία τους γενετικούς τελεστές της διασταύρωσης και μετάλλαξης .

2.2.Ολοκλήρωση επανάληψης

3. Βήμα τοπικής ελαχιστοποίησης:

3.1.Αν $iters \bmod Li = 0$ τότε

- a) Επιλέγουμε ένα τυχαίο Lc χρωμόσωμα από τον γενετικό πληθυσμό και δημιουργούμε ένα νέο σύνολο Ls βασισμένοι από αυτά τα χρωμοσώματα.

b) Για κάθε Xi σε Ls

- Επιλέγουμε ένα τυχαίο Y από τον πληθυσμό.
- Δημιουργούμε έναν απόγονο Z του Xi και Y , χρησιμοποιώντας της διαδικασία one point crossover.
- Αν $f(z) < fi$ τότε $Xi = Z, fi = f(z)$

4. Θέτουμε $iter += 1$. Αν $iter > iterman$ τότε τερμάτισε την διαδικασία, αλλιώς πηγαινε σε 2. Γενετικό βήμα.

5. Βήμα αξιολόγησης:

5.1.Δημιουργούμε ένα τεχνητό νευρωνικό δίκτυο βασισμένοι στα καλύτερα χρωμοσώματα.

5.2.Αξιολογούμε το νέο τεχνητό νευρωνικό δίκτυο με την βοήθεια κάποιου συνόλου δεδομένων είτε classification ή regression και εξάγουμε τα αποτελέσματα.

6. Τέλος αλγορίθμου.

4.4 Πειραματικά αποτελέσματά

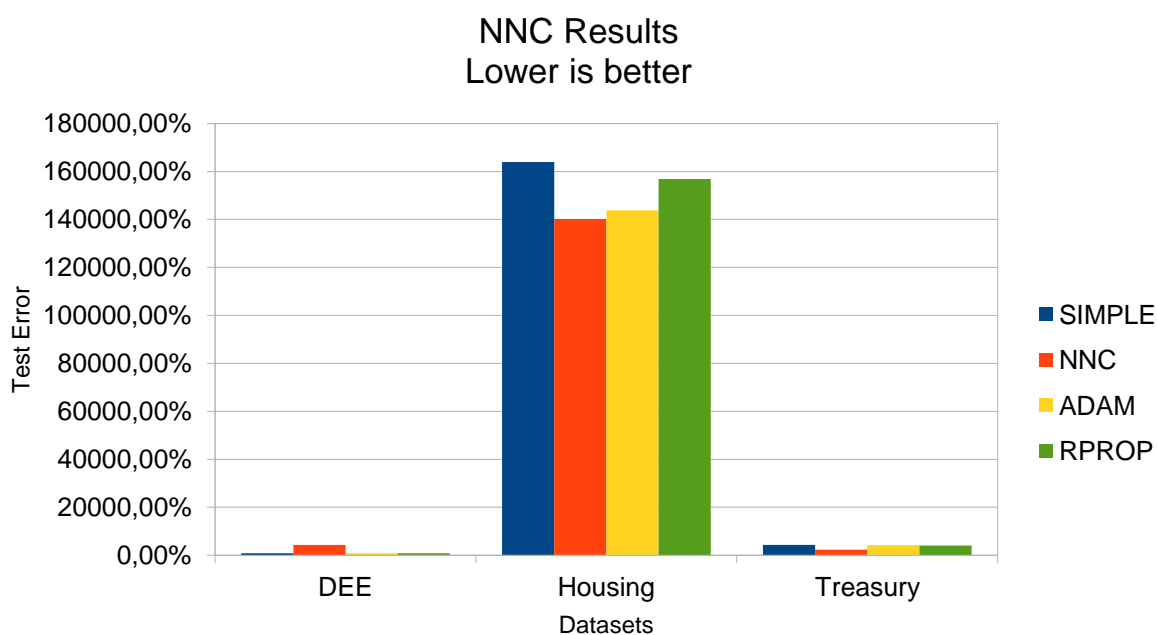
Η προτεινόμενη μέθοδος δοκιμάστηκε πάνω σε τρία διαφορετικά πακέτα από δεδομένα τύπου regression, τα οποία υπάρχουν διαθέσιμα δωρεάν στο διαδίκτυο. Πιο αναλυτικά τα πακέτα που χρησιμοποιήθηκαν είναι τα εξής:

- DEE: Σύνολο δεδομένων που αφορά την πρόβλεψη της τιμής του ηλεκτρικού ρεύματος.
- Housing: Σύνολο δεδομένων που αφορά το ποσοστό καθαρού αέρα σε ένα κτίριο που πρόκειται να αγοραστεί.
- Treasury: Σύνολο δεδομένων που αφορά τα οικονομικά δεδομένα από τις ΗΠΑ.

Τα παραπάνω δεδομένα δοκιμάστηκαν πάνω στην προτεινόμενη μέθοδο, χρησιμοποιώντας δύο γνωστούς αλγορίθμους ως μέθοδο τοπικής ελαχιστοποίησης, την ADAM και RPROP, την προτεινόμενη μέθοδο NNC χωρίς την χρήση κάποιας μεθόδου τοπικής ελαχιστοποίησης, καθώς και την προκαθορισμένη μέθοδο την οποία χρησιμοποιεί το πρόγραμμα, την BFGS. Σε όλες τις δοκιμές, χρησιμοποιήθηκαν οι παράμετροι -d που αφορά το πλήθος των χρωμοσωμάτων με τις τιμές 5-10-20-40 αντίστοιχα, καθώς και η παράμετρος -r, που αφορά το seed, με τιμές από 1 έως 30, για συνολικά 30 δοκιμές σε κάθε σύνολο δεδομένων. Τα τελικά αποτελέσματα για κάθε μέθοδο βρίσκονται στον παρακάτω πίνακα καθώς και σε μορφή γραφήματος. Κάθετα αναγράφεται η μέση τιμή για κάθε μέθοδο, ενώ οριζόντια βρίσκονται τα είδη των δεδομένων που χρησιμοποιήθηκαν για τις δοκιμές.

Πίνακας 2 Τελικά αποτελέσματα δοκιμών

	SIMPLE	NNC	ADAM	RPROP
DEE	880.28%	4355.01%	837.48%	866.48%
Housing	163946.17%	140181.34%	143834.53%	156937.62%
Treasury	4355.01%	2364.37%	4212.09%	4007.57%



Διάγραμμα 1 Διάγραμμα τελικών αποτελεσμάτων NNC

Όπως μπορούμε να δούμε, η προτεινόμενη μέθοδος σε συνδυασμό με την BFGS ως μέθοδο τοπικής ελαχιστοποίησης καταφέρνει να εξάγει πολύ καλύτερα και γρήγορα τελικά αποτελέσματα συγκριτικά με τις υπόλοιπες μεθόδους βελτιστοποίησης, καθώς και να μειώσει αρκετά τον μέσο όρο, σε σχέση με την εκτέλεση του προγράμματος χωρίς την χρήση οποιασδήποτε μεθόδου τοπικής ελαχιστοποίησης, με εξαίρεση το σύνολο δεδομένων DEE, όπου έχει την υψηλότερη τιμή απ' όλες τις άλλες μεθόδους, και εκεί παίρνει την σκυτάλη της καλύτερης τιμής η μέθοδος ADAM.

4.5 Συμπεράσματα

Φτάνοντας στο τέλος αυτής της πτυχιακής εργασίας και έρευνας και στα τελικά συμπεράσματα πάνω στα κατασκευαζόμενα τεχνητά νευρωνικά δίκτυα, μπορούμε να δούμε ότι η προτεινόμενη μέθοδος κατασκευής και εκπαίδευσης ενός τεχνητού νευρωνικού δικτύου, καταφέρνει να ολοκληρώσει τον τελικό της στόχο πολύ πιο σύντομα και αποτελεσματικά συγκριτικά με άλλες διαθέσιμες μεθόδους βελτιστοποίησης, με βάση τις δοκιμές που είδαμε στο προηγούμενο υπό κεφάλαιο, αλλά είναι και αρκετά εύκολη στον τρόπο όπου ο χρήστης θα εισάγει τα δεδομένα, είτε αυτά είναι προβλήματα

διαφορικών εξισώσεων, είτε προβλήματα κατηγοριοποίησης. Βέβαια το συγκεκριμένο πρόγραμμα έχει αρκετά σημεία εξέλιξης τα οποία θα μπορέσουν να βοηθήσουν μελλοντικά και στον τρόπο όπου η μέθοδος κατασκευάζει την τοπολογία του νευρωνικού, όπως για παράδειγμα με την χρήση μίας ενισχυμένης γραμματικής BNF για υποστήριξη πιο περίπλοκων νευρωνικών δικτύων, αλλά και στον τρόπο αλληλεπίδρασης του με τον χρήστη. Εφόσον δεν υπάρχει διεπαφή και το NNC βασίζεται καθαρά σε χρήση τερματικής διεπαφής, υπάρχει η δυνατότητα δημιουργίας γραφικής διεπαφής (GUI), είτε με χρήση βιβλιοθήκης όπως είναι η δωρεάν βιβλιοθήκη QT, την οποία χρησιμοποιεί εξ' αρχής το πρόγραμμα για την υλοποίηση του, αλλά και με χρήση τεχνολογιών web, το οποίο θα κάνει την αλληλεπίδραση αρκετά πιο εύκολη για τον χρήστη καθώς θα μπορεί να υποστηρίξει και δυνατότητα εμφάνισης γραφημάτων και πολλών άλλων χρήσιμων πληροφοριών όλα σε μία ολοκληρωμένη διεπαφή.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- Adam, P., 2017. *Introduction to PonyGE2 for Grammatical Evolution*. [Ηλεκτρονικό]
Available at: <https://towardsdatascience.com/introduction-to-ponyge2-for-grammatical-evolution-d51c29f2315a>
[Πρόσβαση Φεβρουάριος 2024].
- Ankit, U., 2022. *Transformer Neural Networks: A Step-by-Step Breakdown*. [Ηλεκτρονικό]
Available at: <https://builtin.com/artificial-intelligence/transformer-neural-network>
[Πρόσβαση Φεβρουάριος 2024].
- Ashish Vaswani, N. S. N. P. J. U. L. J. A. N. G. L. K. I. P., 2017. *Attention Is All You Need*.
[Ηλεκτρονικό]
Available at: <https://arxiv.org/pdf/1706.03762.pdf>
[Πρόσβαση Φεβρουάριος 2024].
- AWS, 2024. *What is a Neural Network?*. [Ηλεκτρονικό]
Available at: <https://aws.amazon.com/what-is/neural-network/>
[Πρόσβαση Φεβρουάριος 2024].
- Ba, D. P. K. & J. L., 2014. *Adam: A Method for Stochastic Optimization*. [Ηλεκτρονικό]
Available at: <https://arxiv.org/pdf/1412.6980.pdf>
[Πρόσβαση Φεβρουάριος 2024].
- Banoula, M., 2023. *What is Perceptron: A Beginners Guide for Perceptron*. [Ηλεκτρονικό]
Available at: <https://www.simplilearn.com/tutorials/deep-learning-tutorial/perceptron>
[Πρόσβαση Φεβρουάριος 2024].
- Bento, C., 2021. *Multilayer Perceptron Explained with a Real-Life Example and Python Code: Sentiment Analysis*. [Ηλεκτρονικό]
Available at: <https://towardsdatascience.com/multilayer-perceptron-explained-with-a-real-life-example-and-python-code-sentiment-analysis-cb408ee93141>
[Πρόσβαση Φεβρουάριος 2024].
- Braun, M. R. & H., 1993. *A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm*. [Ηλεκτρονικό]
Available at: <https://paginas.fe.up.pt/~ee02162/dissertacao/RPROP%20paper.pdf>
[Πρόσβαση Φεβρουάριος 2024].

Brownlee, J., 2019. *How Machine Learning Algorithms Work (they learn a mapping of input to output)*. [Ηλεκτρονικό]

Available at: <https://machinelearningmastery.com/how-machine-learning-algorithms-work/>

[Πρόσβαση Φεβρουάριος 2024].

Caceres, P., 2020. *The ADALINE - Theory and Implementation of the First Neural Network Trained With Gradient Descent*. [Ηλεκτρονικό]

Available at: <https://pabloinsente.github.io/the-adaline>

[Πρόσβαση Φεβρουάριος 2024].

Chandrakant, K., 2023. *Overview of Nature-Inspired Metaheuristic Algorithms*. [Ηλεκτρονικό]

Available at: <https://www.baeldung.com/cs/nature-inspired-algorithms>

[Πρόσβαση Φεβρουάριος 2024].

Chauhan, N. S., 2024. *Optimization Algorithms in Neural Networks*. [Ηλεκτρονικό]

Available at: <https://www.kdnuggets.com/2020/12/optimization-algorithms-neural-networks.html>

[Πρόσβαση Φεβρουάριος 2024].

Doshi, S., 2019. *Various Optimization Algorithms For Training Neural Network*. [Ηλεκτρονικό]

Available at: <https://towardsdatascience.com/optimizers-for-training-neural-network-59450d71caf6>Sanket Doshi

[Πρόσβαση Φεβρουάριος 2024].

Efimov, V., 2023. *Understanding Deep Learning Optimizers: Momentum, AdaGrad, RMSProp & Adam*. [Ηλεκτρονικό]

Available at: <https://towardsdatascience.com/understanding-deep-learning-optimizers-momentum-adagrad-rmsprop-adam-e311e377e9c2>

[Πρόσβαση Φεβρουάριος 2024].

Filippo Amato, A. L.-R. E. M. P.-M. J. H. P. V., 2013. *Artificial neural networks in medical diagnosis*. [Ηλεκτρονικό]

Available at:

https://www.researchgate.net/publication/250310836_Artificial_neural_networks_in_medical_diagnosis

[Πρόσβαση Φεβρουάριος 2024].

GeeksforGeeks, 2023. *Crossover in Genetic Algorithm*. [Ηλεκτρονικό]

Available at: https://www.geeksforgeeks.org/crossover-in-genetic-algorithm/?ref=header_search

[Πρόσβαση Φεβρουάριος 2024].

GeeksforGeeks, 2023. *Supervised and Unsupervised learning*. [Ηλεκτρονικό]

Available at: <https://www.geeksforgeeks.org/supervised-unsupervised-learning/>

[Πρόσβαση Φεβρουάριος 2024].

Google for Developers, 2023. *Supervised Learning*. [Ηλεκτρονικό]

Available at: <https://developers.google.com/machine-learning/intro-to-ml/supervised>

[Πρόσβαση Φεβρουάριος 2024].

Ioannis G. Tsoulos, A. T. T. D. G. T., 2022. *Prediction of COVID-19 Cases Using Constructed Features by Grammatical Evolution*. [Ηλεκτρονικό]

Available at: <https://typeset.io/papers/prediction-of-covid-19-cases-using-constructed-features-by-1hl8rf2l>

[Πρόσβαση Φεβρουάριος 2024].

Kostadinov, S., 2019. *Understanding Backpropagation Algorithm*. [Ηλεκτρονικό]

Available at: <https://towardsdatascience.com/understanding-backpropagation-algorithm-7bb3aa2f95fd>

[Πρόσβαση Φεβρουάριος 2024].

Luca, J., 2023. *How AI Camera Works: The Future of Smartphone Photography*. [Ηλεκτρονικό]

Available at: <https://medium.com/@rmndrathna4/how-ai-camera-works-the-future-of-smartphone-photography-50854b6269fd>

[Πρόσβαση Φεβρουάριος 2024].

Meghana Kshirsagar, G. V. H. W., 2022. *Design of a cryptographically secure pseudo random number generator with grammatical evolution*. [Ηλεκτρονικό]

Available at: <https://typeset.io/papers/design-of-a-cryptographically-secure-pseudo-random-number-27ph4hpm>

[Πρόσβαση Φεβρουάριος 2024].

MonkeyLearn, 2020. *What Is Data Classification? Types & AI Tools*. [Ηλεκτρονικό]

Available at: <https://monkeylearn.com/blog/data-classification/>

[Πρόσβαση Φεβρουάριος 2024].

Monsters, D., 2017. *7 types of Artificial Neural Networks for Natural Language Processing*. [Ηλεκτρονικό]

Available at: <https://medium.com/@datamonsters/artificial-neural-networks-for-natural-language-processing-part-1-64ca9ebfa3b2>

[Πρόσβαση Φεβρουάριος 2024].

Naeem Ahmed, R. A. H. A. D. K. B. A. T. S., 2022. *Machine Learning Techniques for Spam Detection in Email and IoT Platforms: Analysis and Research Challenges*. [Ηλεκτρονικό]

Available at: <https://www.hindawi.com/journals/scn/2022/1862888/>

[Πρόσβαση Φεβρουάριος 2024].

O'Neill, M., 2017. *Grammatical-Evolution.org*. [Ηλεκτρονικό]

Available at: <http://www.grammatical-evolution.org/>

[Πρόσβαση Φεβρουάριος 2024].

Pitts, W. M. & W., 1943. *A Logical calculus of the ideas immanent in nervous activity*.

[Ηλεκτρονικό]

Available at: <https://www.cs.cmu.edu/~epxing/Class/10715/reading/McCulloch.and.Pitts.pdf>

[Πρόσβαση Φεβρουάριος 2024].

Simplilearn, 2023. *Intro to Recursive Neural Network in Deep Learning*. [Ηλεκτρονικό]

Available at: <https://www.simplilearn.com/recursive-neural-network-in-deep-learning-article>

[Πρόσβαση 2024].

Simplilearn, 2023. *Power of Recurrent Neural Networks (RNN): Revolutionizing AI*. [Ηλεκτρονικό]

Available at: <https://www.simplilearn.com/tutorials/deep-learning-tutorial/rnn>

[Πρόσβαση Φεβρουάριος 2024].

Soni, D., 2018. *Introduction to Evolutionary Algorithms*. [Ηλεκτρονικό]

Available at: <https://towardsdatascience.com/introduction-to-evolutionary-algorithms-a8594b484ac>

[Πρόσβαση Φεβρουάριος 2024].

Sungwon Yang, M. G., 2010. *Energy-Efficient Accelerometer Data Transfer for Human Body Movement Studies*. [Ηλεκτρονικό]

Available at: <https://typeset.io/papers/energy-efficient-accelerometer-data-transfer-for-human->

body-z4ttz0k2dk

[Πρόσβαση Φεβρουάριος 2024].

Turing, A. M., 1950. *Computing Machinery and Intelligence*. [Ηλεκτρονικό]

Available at: <https://academic.oup.com/mind/article/LIX/236/433/986238?login=false>

[Πρόσβαση Φεβρουάριος 2024].

Wikipedia, 2024. *ADALINE*. [Ηλεκτρονικό]

Available at: <https://en.wikipedia.org/wiki/ADALINE>

[Πρόσβαση Φεβρουάριος 2024].

Wikipedia, 2024. *Crossover (genetic algorithm)*. [Ηλεκτρονικό]

Available at: [https://en.wikipedia.org/wiki/Crossover_\(genetic_algorithm\)](https://en.wikipedia.org/wiki/Crossover_(genetic_algorithm))

[Πρόσβαση Φεβρουάριος 2024].

Wikipedia, 2024. *Evolutionary algorithm*. [Ηλεκτρονικό]

Available at: https://en.wikipedia.org/wiki/Evolutionary_algorithm

[Πρόσβαση Φεβρουάριος 2024].

Wikipedia, 2024. *Genetic algorithm*. [Ηλεκτρονικό]

Available at: https://en.wikipedia.org/wiki/Genetic_algorithm#cite_note-Fraser_1970-43

[Πρόσβαση Φεβρουάριος 2024].

Wikipedia, 2024. *Grammatical evolution*. [Ηλεκτρονικό]

Available at: https://en.wikipedia.org/wiki/Grammatical_evolution

[Πρόσβαση Φεβρουάριος 2024].

Wikipedia, 2024. *Knapsack problem*. [Ηλεκτρονικό]

Available at: https://en.wikipedia.org/wiki/Knapsack_problem

[Πρόσβαση Φεβρουάριος 2024].

Wikipedia, 2024. *Neural network (machine learning)*. [Ηλεκτρονικό]

Available at: [https://en.wikipedia.org/wiki/Neural_network_\(machine_learning\)](https://en.wikipedia.org/wiki/Neural_network_(machine_learning))

[Πρόσβαση Φεβρουάριος 2024].

Wikipedia, 2024. *Survival of the fittest*. [Ηλεκτρονικό]

Available at: https://en.wikipedia.org/wiki/Survival_of_the_fittest

[Πρόσβαση Φεβρουάριος 2024].

Yumnah Hasan, A. d. L. F. A. D. R. F. d. B. P. H. C. R., 2024. *Interpretable Solutions for Breast Cancer Diagnosis with Grammatical Evolution and Data Augmentation*. [Ηλεκτρονικό]

Available at: <https://arxiv.org/abs/2401.14255>

[Πρόσβαση Φεβρουάριος 2024].

