



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΜΕ ΕΦΑΡΜΟΓΕΣ ΣΤΗ ΒΙΟΙΑΤΡΙΚΗ**

Σύστημα διεπαφής ανθρώπινου εγκεφάλου-υπολογιστή

Ειρήνη Ρίτη

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ
Υπεύθυνος
Πλαγιανάκος Βασίλης
Αναπληρωτής Καθηγητής**

Λαμία, 2014



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΜΕ ΕΦΑΡΜΟΓΕΣ ΣΤΗ ΒΙΟΙΑΤΡΙΚΗ**

Σύστημα διεπαφής ανθρώπινου εγκεφάλου-υπολογιστή

Ειρήνη Ρίτη

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ
Υπεύθυνος
Πλαγιανάκος Βασίλης
Αναπληρωτής Καθηγητής**

Λαμία, 2014

Σύστημα διεπαφής ανθρώπινου εγκεφάλου-υπολογιστή

Ειρήνη Ρίτη

Τριμελής Επιτροπή:

Πλαγιανάκος Βασίλης , Αναπληρωτής καθηγητής

Αναγνωστόπουλος Ιωάννης, Επίκουρος καθηγητής

Σανδαλίδης Χάρης, Επίκουρος καθηγητής

Διπλωματική εργασία της

Ρίτη Ειρήνης

eriti[a]dib.uth.gr

eirini.riti[a]gmail.com

Περίληψη

Σκοπός της παρούσας πτυχιακής εργασίας είναι η δημιουργία αλγορίθμου προκειμένου να είναι δυνατή η πρόβλεψη και η ανίχνευση της κίνησης ενός συνόλου ηλεκτροεγκεφαλογραφημάτων (EEG). Για την κατηγοριοποίηση των σημάτων EEG χρησιμοποιήθηκαν ο αλγόριθμος k κοντινότερων γειτόνων και το τεχνητό νευρωνικό δίκτυο. Οι δύο αυτές μέθοδοι είναι από τις πιο γνωστές μεθόδους κατηγοριοποίησης. Ο αριθμός των κοντινότερων γειτόνων και ο αριθμός των κρυμμένων επιπέδων στο τεχνητό νευρωνικό δίκτυο, χρησιμοποιούνται για την επίτευξη της κατηγοριοποίησης με την μεγαλύτερη δυνατή ακρίβεια και κάθε φορά παρέχονται από τον χρήστη.

Το ηλεκτροεγκεφαλογράφημα είναι ένας πολύ αποτελεσματικός τρόπος αναγνώρισης κινήσεων, καθώς κάθε κίνηση είναι αποτέλεσμα των νευρώνων του εγκεφάλου. Ένα βασικό πρόβλημα, ωστόσο, είναι ο μεγάλος όγκος δεδομένων. Για τον λόγο αυτό στα πλαίσια της εργασίας αυτής, χρησιμοποιήθηκαν ο μετασχηματισμός Fourier, η μέση τιμή και η διάδοση των σημάτων EEG για κάθε κίνηση, ώστε να διαμορφωθούν χαρακτηριστικά, να μειωθεί ο όγκος δεδομένων και κατ' επέκταση η μείωση του χρόνου εκτέλεσης. Γίνεται, λοιπόν, μία σύγκριση μεταξύ των δύο μεθόδων, με σκοπό να εντοπιστεί η μέθοδος με την μεγαλύτερη ακρίβεια ενώ παράλληλα γίνεται σύγκριση κάθε μεθόδου με και χωρίς διαμόρφωση χαρακτηριστικών για να αποδειχθεί η αποτελεσματικότητα της χρήσης της διαμόρφωσης χαρακτηριστικών.

Στη συνέχεια, δημιουργήθηκε ένα γραφικό περιβάλλον, το οποίο δέχεται τα ηλεκτροεγκεφαλογραφήματα που είναι ήδη γνωστά, μαζί με την κατηγοριοποίηση του, καθώς και το σύνολο των σημάτων προς κατηγοριοποίηση. Το γραφικό περιβάλλον παρέχει την δυνατότητα επιλογής μεθόδου κατηγοριοποίησης, την είσοδο ορισμένων χαρακτηριστικών στοιχείων, διαφορετικών για κάθε μέθοδο, την εφαρμογή αυτής της επιλογής, την πρόβλεψη, το ποσοστό επί της εκατό (%) της ακρίβειας και ένα γράφημα με τα δεδομένα που χρησιμοποιήθηκαν για τον υπολογισμό της ακρίβειας, ώστε να φαίνονται γραφικά οι λάθος προβλέψεις.

Λέξεις κλειδιά: Τεχνητό νευρωνικό δίκτυο, αλγόριθμος k κοντινότερων γειτόνων, κατηγοριοποίηση σημάτων EEG, εξαγωγή χαρακτηριστικών, γραφικό περιβάλλον.

Περιεχόμενο: κείμενο, κώδικας υλοποίησης αλγορίθμων στο Matlab.

Abstract

The purpose of the current dissertation is the synthesis of an algorithm in order to predict and monitor the movements of a set of EEG signal. For the classification of the EEG signal, were used the k Nearest Neighbor algorithm and the Artificial Neural Network. These two systems are the most popular methods of classification. The number of k nearest neighbor and the number of the hidden layers in the artificial neural network, which are provided by the creator, are used to complete the categorization with the exact precision.

The electroencephalograph is one of the most effective ways to identify movements, whilst every movement that occurs is the result of the brain's neural function. However, the major problem of that action is the large volume of data. For that reason, during this work, it was used the Fourier remodeling, the mean and the dissemination of the EEG signals for every movement, so that features could be formed, the volume of data can be reduced and eventually the diminishing of the running time can be possible. So a comparison between these two methods is made, in order to identify the method with the highest accuracy. Simultaneously, another comparison is made for each method with and without feature formation, in order to demonstrate the effectiveness of the use of the feature formation.

Then, a graphical environment was developed, which accepts first of all the EEG signals that are already known, secondly, it gives the user the opportunity to choose which of the two methods want to deploy, to add some features for every method and final the volume of data to classify them. Subsequently, the GUI implements the chosen method and presents the prediction, the number percentage (%) of the accuracy and a graph which contains the data that were used for the calculation of the accuracy, in order to be present the wrong predictions using graphs.

Keywords: Artificial neural network & nearest neighbor algorithm, classification of EEG signals, feature extraction, graphical user interface.

Content: text, implementation code in Matlab

Ευχαριστίες

Θα ήθελα να επισημάνω την απέραντη ευγνωμοσύνη μου στους γονείς μου, που καθ' όλη την διάρκεια των σπουδών μου ήταν δίπλα μου και με στήριζαν σε κάθε μου αγώνα, καθώς επίσης και στον αδερφό μου που μου στάθηκε!

Ιδιαίτερες ευχαριστίες εκφράζω στον καθηγητή μου κ. Πλαγιανάκο Βασίλη, για την βοήθειά του στην διεκπεραίωση της πτυχιακής εργασίας μου και για την καλή συνεργασία που είχαμε!

Τελευταίο αλλά όχι λιγότερο σημαντικό, θα ήθελα να ευχαριστήσω ένα πολύ ξεχωριστό άνθρωπο, που βρέθηκε στη ζωή μου για να με βοηθήσει να εξελιχθώ ως άτομο και που μου στάθηκε περισσότερο από τον καθένα!

Περιεχόμενα

Κεφάλαιο 1 - Εισαγωγή	1
Κεφάλαιο 2	3
Κατηγοριοποίηση	3
2.1 Βασικές Έννοιες	4
2.2 Απόδοση της κατηγοριοποίησης	5
Sensitivity και Specificity	6
2.3 Αλγόριθμοι κατηγοριοποίησης.....	7
2.4 Ηλεκτροεγκεφαλογράφημα (EEG).....	8
Κεφάλαιο 3	9
Κ Κοντινότεροι Γείτονες	9
3.1 KNN κατηγοριοποιητής	10
3.2 Επιλογή του k.....	10
3.3 Μέτρα απόστασης.....	11
3.4 Ιδιότητες του k-NN κατηγοριοποιητή	13
3.5 Παλινδρόμηση.....	13
Κεφάλαιο 4	14
Τεχνητό Νευρωνικό Δίκτυο	14
4.1 Τεχνητό Νευρωνικό Δίκτυο.....	15
4.2 Μοντέλο Τεχνητού Νευρώνα.....	15
4.2.1 Τυπικές περιπτώσεις για τη συνάρτηση ενεργοποίησης.....	16
4.3 Τεχνητά Νευρωνικά Δίκτυα	18
4.4 Μάθηση και Ανάκληση.....	19
4.5 Μέθοδοι Εκπαίδευσης του Τεχνητού Νευρωνικού Δικτύου	20
4.5.1 Levenberg-Marquardt Backpropagation (LM).....	20
4.5.2 Resilient Backpropagation (RP)	20
4.5.3 Gradient Descent Backpropagation (GD).....	21
4.5.4 Σύγκριση των μεθόδων εκπαίδευσης.....	21
4.6 Ιδιότητες των Νευρώνων Δικτύου	22
4.7 Νευρωνικά Δίκτυα Πρόσθιας Τροφοδότησης	22
4.8 Νευρωνικά Δίκτυα βασισμένα σε ανταγωνισμό	22
Κεφάλαιο 5	24
Πειραματική Μελέτη	24

5.1 Συλλογή δεδομένων	25
5.2 Επεξεργασία δεδομένων	27
5.3 Εφαρμογή Νευρωνικού Δικτύου και Αλγορίθμου k Κοντινότερων Γειτόνων (k-NN)	28
5.4 Υπολογισμός της ακρίβειας κάθε αλγορίθμου	35
5.5 Γραφικό περιβάλλον Χρήστη(GUI)	39
Κεφάλαιο 6	43
Συμπεράσματα.....	43
Κεφάλαιο 7	47
Πηγαίος κώδικας.....	47
7.1 Κώδικας για το Τεχνητό Νευρωνικό Δίκτυο.....	48
7.2 Κώδικας για τον αλγόριθμο k κοντινότερων γειτόνων.....	52
7.3 Κώδικας για το Γραφικό Περιβάλλον	55
Βιβλιογραφία.....	67

Ευρετήριο σχημάτων

Σχήμα 1: Το πρόβλημα της κατηγοριοποίησης.....	4
Σχήμα 2: Κατηγοριοποίηση κλη αλγορίθμου	11
Σχήμα 3: Μοντέλο Τεχνητού Νευρώνα.....	15
Σχήμα 4: Βηματική Συνάρτηση	16
Σχήμα 5: Συνάρτηση Πρόσημου	17
Σχήμα 6: Λογιστική συνάρτηση	17
Σχήμα 7: Τεχνητό Νευρωνικό Δίκτυο με πλήρη διασύνδεση μεταξύ των διαδοχικών επιπέδων.	18
Σχήμα 8: Αρχιτεκτονική ανταγωνιστικού ΤΝΔ.....	23
Σχήμα 9: Συνολική εργασία.....	25
Σχήμα 10: Αριστερά: Θέσεις των ηλεκτροδίων. Δεξιά: Σχήμα χρόνου	26
Σχήμα 11. Τεχνητό Νευρωνικό Δίκτυο.....	29
Σχήμα 12: Best Validation Performance.....	31
Σχήμα 13: Training State.....	32
Σχήμα 14: Error Histogram.....	33
Σχήμα 15: Regression.....	34
Σχήμα 16: Η ακρίβεια των δύο αλγορίθμων για τα δύο σύνολα δεδομένων.	35
Σχήμα 17: Η ακρίβεια των δύο αλγορίθμων για τα δύο σύνολα δεδομένων σε γράφημα.....	36
Σχήμα 18: Ακρίβεια των δύο μεθόδων για δεδομένα με εξαγωγή χαρακτηριστικών και για το σύνολο των δεδομένων.	36
Σχήμα 19: Ακρίβεια των δύο μεθόδων για δεδομένα με εξαγωγή χαρακτηριστικών και για το σύνολο των δεδομένων σε γράφημα.....	37
Σχήμα 20: Confusion matrix.	38
Σχήμα 21: Γραφικό περιβάλλον.....	40
Σχήμα 22: Γραφικό περιβάλλον με εφαρμογή κατηγοριοποίησης τεχνητού νευρωνικού δικτύου.....	41
Σχήμα 23: Γραφικό περιβάλλον με εφαρμογή κατηγοριοποίησης αλγορίθμου k-NN.....	41
Σχήμα 24: Γραφικό περιβάλλον με εφαρμογή κατηγοριοποίησης με k κοντινότερους γείτονες.	42

Κεφάλαιο 1 - Εισαγωγή

Η εξόρυξη δεδομένων χρησιμοποιείται όλο και περισσότερο στην σημερινή εποχή. Με τον όρο εξόρυξη δεδομένων εννοούμε την συλλογή μεγάλης ποσότητας δεδομένων και την μετατροπή τους σε βαρυσήμαντες πληροφορίες, προκειμένου να βγάλουμε κάποια συμπεράσματα ή να προβλέψουμε μελλοντικές τιμές νέων δεδομένων. Για παράδειγμα, η Google χρησιμοποιεί την εξόρυξη δεδομένων ώστε να συλλέξει στοιχεία για κάθε χρήστη, σύμφωνα με τις αναζητήσεις του, με σκοπό την πρόβλεψη μελλοντικής αναζήτησης του συγκεκριμένου χρήστη και κατά συνέπεια την προβολή σε αυτόν κατάλληλου τύπου διαφημίσεις. Γενικά, η εξόρυξη δεδομένων και η κατηγοριοποίηση αυτών συναντά απήχηση σε πολλά πεδία της πληροφορικής. Έτσι, έχουν δημιουργηθεί διάφοροι τρόποι κατάταξης δεδομένων.

Οι κυριότεροι μέθοδοι κατάταξης δεδομένων είναι ο αλγόριθμος κ κοντινότερων γειτόνων, το τεχνητό νευρωνικό δίκτυο και τα δένδρα αποφάσεων. Η πρώτη μέθοδος, ο αλγόριθμος κ κοντινότερων γειτόνων, είναι μία από τις δημοφιλέστερες μεθόδους κατηγοριοποίησης δεδομένων. Ο αλγόριθμος αυτός είναι μια μη παραμετρική μέθοδος κατάταξης δεδομένων, η οποία αφού πρώτα εκπαιδευτεί με ήδη γνωστά στον χρήστη δεδομένα, κατατάσσει τα υπό εξέταση δεδομένα στην κατηγορία στην οποία ανήκουν οι περισσότεροι από τους κ κοντινότερους γείτονες. Η ακρίβεια του αλγορίθμου εξαρτάται από το πόσοι κοντινότεροι γείτονες θα ληφθούν υπόψη για την κατηγοριοποίηση. Έτσι, είναι απαραίτητο να βρούμε τον αριθμό κ που δίνει την καλύτερη κατηγοριοποίηση. Ωστόσο, αν ο αριθμός αυτός είναι πολύ μεγάλος, απαιτείται πολύς χρόνος για την κατηγοριοποίηση. Άρα, καταλήγουμε στο συμπέρασμα ότι μας συμφέρει να αναζητήσουμε έναν μικρότερο αριθμό κοντινότερων γειτόνων, χάνοντας λίγο από την ακρίβεια της κατηγοριοποίησης, κερδίζοντας όμως σε χρόνο εκτέλεσης.

Η δεύτερη μέθοδος κατηγοριοποίησης είναι το τεχνητό νευρωνικό δίκτυο. Πρόκειται για ένα κύκλωμα διασυνδεδεμένων νευρώνων, το οποίο δέχεται ένα σύνολο αριθμητικών εισόδων από διάφορες πηγές (είτε από άλλους νευρώνες, είτε από το περιβάλλον), επεξεργάζεται τις εισόδους αυτές και παράγει μία έξοδο. Η έξοδος αυτή είτε κατευθύνεται στο περιβάλλον, είτε τροφοδοτείται ως είσοδο σε άλλους νευρώνες του δικτύου. Υπάρχουν τρεις τύποι νευρώνων: οι νευρώνες εισόδου, οι οποίοι εισάγουν τα δεδομένα στο δίκτυο, χωρίς να επιτελούν κανέναν υπολογισμό, οι νευρώνες εξόδου, που διοχετεύουν στο περιβάλλον τις τελικές αριθμητικές εξόδους του δικτύου και οι υπολογιστικοί ή κρυφοί νευρώνες που πολλαπλασιάζουν κάθε είσοδο με το αντίστοιχο συναπτικό βάρος και υπολογίζουν το ολικό άθροισμα.

Η τρίτη μέθοδος είναι τα δένδρα αποφάσεων. Τα δένδρα αυτά χρησιμοποιούνται για την κατηγοριοποίηση και πρόβλεψη δεδομένων. Ένα δέντρο απόφασης κατασκευάζεται σύμφωνα με ένα σύνολο εκπαίδευσης προ-κατηγοριοποιημένων δεδομένων. Κάθε εσωτερικός κόμβος προσδιορίζει τον έλεγχο των γνωρισμάτων και κάθε κλαδί που συνδέει τους εσωτερικούς με τους απόγονους αντιστοιχεί σε μία πιθανή τιμή για το γνώρισμα.

Στην παρούσα πτυχιακή εργασία γίνεται μία προσπάθεια μελέτης, υλοποίησης και εκπαίδευσης μεθόδων Τεχνητής Νοημοσύνης με σκοπό τον διαχωρισμό και την πρόβλεψη κλάσης μελλοντικών ηλεκτροεγκεφαλογραφημάτων (EEG). Τα EEG σήματα που χρησιμοποιήθηκαν προέρχονται από τον διαγωνισμό Berlin BCI_Compensation II (BCI = Brain Computer Interface). Πρόκειται για έναν διαγωνισμό με σκοπό την επεξεργασία σημάτων και την εφαρμογή μεθόδων κατάταξης ηλεκτροεγκεφαλογραφημάτων. Στην εργασία αυτή, εφαρμόστηκαν ο αλγόριθμος κ κοντινότερου γείτονα (knn) και το τεχνητό νευρωνικό δίκτυο (neural network). Η εφαρμογή τους πραγματοποιήθηκε με την χρήση του Matlab. Έπειτα, έγινε μία σύγκριση των δύο μεθόδων, προκειμένου να εντοπιστεί η μέθοδος με την μεγαλύτερη ακρίβεια. Τέλος, υλοποιήθηκε ένα Γραφικό Περιβάλλον Χρήστη (GUI = Graphical User Interface), το οποίο δίνει την δυνατότητα χρήσης του knn αλγορίθμου ή του τεχνητού νευρωνικού δικτύου για την κατάταξη νέων ηλεκτροεγκεφαλογραφημάτων και παρουσιάζει την κατηγοριοποίηση αυτών.

Κεφάλαιο 2

Κατηγοριοποίηση

2.1 Βασικές Έννοιες

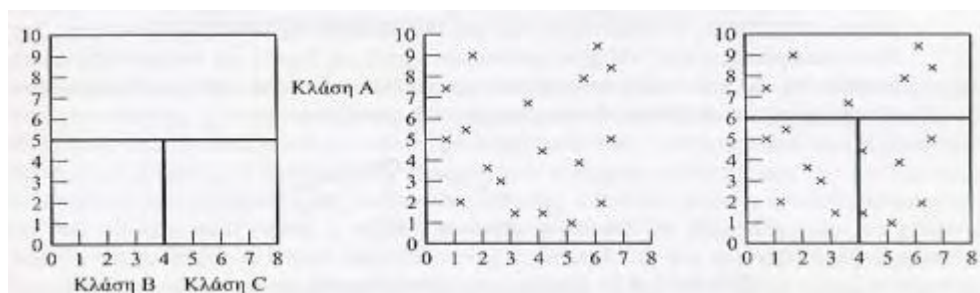
Η κατηγοριοποίηση είναι η πιο γνωστή τεχνική εξόρυξης δεδομένων, κατά την οποία ένα στοιχείο αντιστοιχείται σε ένα προκαθορισμένο σύνολο κατηγοριών. Χρησιμοποιείται καθημερινά στην ζωή και την εργασία μας. Ο όρος αυτός συναντάται στην βιβλιογραφία και ως **ταξινόμηση**. Στόχος της κατηγοριοποίησης είναι η ανάπτυξη ενός μοντέλου, με σκοπό την μελλοντική κατηγοριοποίηση νέων δεδομένων.

Για την εκτέλεση της κατηγοριοποίησης, απαραίτητη προϋπόθεση είναι η γνώση των δεδομένων. Για την εκπαίδευση χρειάζονται τα **δεδομένα εκπαίδευσης (training data)**, δηλαδή ένα σύνολο δεδομένων εισόδου (x_{train}) και η κατηγοριοποίηση αυτών (y_{train}). Τα δεδομένα αυτά χρησιμοποιούνται για την δημιουργία ενός μοντέλου. Η κατηγοριοποίηση δέχεται ως είσοδο τα δεδομένα εκπαίδευσης και παράγει ως έξοδο το μοντέλο. Η διαδικασία αυτή ονομάζεται διαδικασία μάθησης. Εκπαιδεύουμε δηλαδή το μοντέλο με τα δεδομένα που έχουμε και είναι ήδη κατηγοριοποιημένα, με σκοπό να μπορεί να κατηγοριοποιήσει κάθε νέο δεδομένο που θα του εισάγουμε. Υπάρχουν δύο είδη μάθησης:

- **Εποπτευόμενη μάθηση (supervised learning):** Σ' αυτό το είδος κατηγοριοποίησης το σύνολο των δεδομένων εκπαίδευσης περιλαμβάνει ένα χαρακτηριστικό το οποίο προσδιορίζει σε ποια κλάση ανήκει η κάθε πλειάδα. Με άλλα λόγια, είναι γνωστές οι κατηγορίες (y_{train}) των δεδομένων εκπαίδευσης.
- **Μη εποπτευόμενη μάθηση (unsupervised learning):** Στο είδος αυτό δεν είναι γνωστές οι κλάσεις στις οποίες ανήκουν τα δεδομένα εκπαίδευσης. Δίνονται μόνο οι παρατηρήσεις / μετρήσεις, με στόχο να βρεθούν οι κλάσεις μέσα στα δεδομένα.

Το επόμενο βήμα είναι η εισαγωγή μη ταξινομημένων δεδομένων (x_{test}) στο μοντέλο, με σκοπό την εύρεση της κλάσης στην οποία ανήκουν (μελλοντικές περιπτώσεις).

Στην εργασία αυτή δεν εξετάζεται η μη εποπτευόμενη μάθηση.



α. ορισμός κατηγοριών

β. ΒΔ προς κατηγοριοποίηση

γ. κατηγοριοποίηση ΒΔ

Σχήμα 1: Το πρόβλημα της κατηγοριοποίησης

Στην παραπάνω εικόνα παρουσιάζεται η διαδικασία της κατηγοριοποίησης. Στο σχήμα (α) δημιουργούνται οι προκαθορισμένες κατηγορίες / κλάσεις, στο σχήμα (β) παρουσιάζονται δείγματα δεδομένων εισόδου και στο σχήμα (γ) η κατηγοριοποίηση των δεδομένων με βάση τις ορισμένες κατηγορίες.

2.2 Απόδοση της κατηγοριοποίησης

Για την κατηγοριοποίηση δεδομένων υπάρχουν πολλοί αλγόριθμοι. Αυτό που μας απασχολεί όμως είναι ποιος αλγόριθμος είναι ο καλύτερος και πιο αποτελεσματικός. Το μέτρο σύγκρισης και συνεπώς το μέτρο που χαρακτηρίζει έναν αλγόριθμο καλό ή όχι, είναι η εκτίμηση της **ακρίβειας (accuracy)**. Η εκτίμηση της ακρίβειας μας δείχνει πόσο καλά ταξινομεί ο αλγόριθμος τα δεδομένα με τα οποία δεν έχει εκπαιδευτεί. Άλλα μέτρα σύγκρισης των αλγορίθμων είναι η ταχύτητα εκτέλεσής του, η σωστή πρόβλεψη με ελλιπή δεδομένα ή δεδομένα με θόρυβο (robustness), η αποτελεσματική κατασκευή του μοντέλου για μεγάλη ποσότητα δεδομένων (scalability).

Το πιο βασικό μέτρο σύγκρισης όμως είναι η απόδοση, δηλαδή η ακρίβεια στην πρόβλεψη της κλάσης. Παρ' όλα αυτά, θα πρέπει να λαμβάνονται υπόψη και τα υπόλοιπα μέτρα, καθώς δεν έχει νόημα ένας αλγόριθμος με πολύ υψηλή ακρίβεια αλλά με μεγάλο χρόνο εκτέλεσης. Ίσως να ήταν καλύτερη επιλογή ένας αλγόριθμος με λίγο χαμηλότερη ακρίβεια αλλά με μικρότερο χρόνο εκτέλεσης.

Για τον υπολογισμό της ακρίβειας, μπορούμε να χρησιμοποιήσουμε ένα σύνολο δεδομένων για εκπαίδευση και το ίδιο ή μέρος του συνόλου αυτού να το χρησιμοποιήσουμε για δοκιμή. Αυτός ο τρόπος εκτίμησης της ακρίβειας θα ήταν πολύ ακριβής αφού χρησιμοποιεί τα ίδια δεδομένα για εκπαίδευση και δοκιμή. Ένας άλλος τρόπος μέτρησης της ακρίβειας είναι η μέθοδος κατακράτησης. Η μέθοδος αυτή χωρίζει το σύνολο των δεδομένων σε δύο τμήματα, με τυχαίο τρόπο. Το πρώτο τμήμα χρησιμοποιείται ως δεδομένα εισόδου για εκπαίδευση και το δεύτερο τμήμα για δοκιμή και εκτίμηση της ακρίβειας.

Ένας τρίτος τρόπος υπολογισμού της ακρίβειας είναι ο k-fold cross validation. Σύμφωνα με αυτόν, το σύνολο των δεδομένων χωρίζεται σε k υποσύνολα. Το ένα υποσύνολο χρησιμοποιείται για δοκιμή και τα υπόλοιπα k-1 για εκπαίδευση. Αυτό επαναλαμβάνεται k φορές, ώστε να γίνει δοκιμή και για τα k υποσύνολα. Για το λόγο αυτό, η προσέγγιση αυτή απαιτεί περισσότερο χρόνο από ότι οι δύο προαναφερθέντες τρόποι υπολογισμού της ακρίβειας. Η τελική επιλογή του k γίνεται για την τιμή η οποία επιτυγχάνει την μεγαλύτερη ακρίβεια ταξινόμησης.'

Sensitivity και Specificity

Πολύ σημαντικό ρόλο στην μέτρηση της αποδοτικότητας μιας κατηγοριοποίησης παίζουν η Ευαισθησία και η Ειδικότητα (Sensitivity and Specificity) των αποτελεσμάτων. Πρόκειται για μία στατιστική μέτρηση της αποτελεσματικότητας μιας δυαδικής κατηγοριοποίησης. Πιο αναλυτικά:

- Sensitivity, ή αλλιώς ποσοστό αληθώς θετικών αποτελεσμάτων – True Positive Rate. Υπολογίζει το ποσοστό των θετικών αποτελεσμάτων, τα οποία είναι σωστά κατηγοριοποιημένα ως θετικά. Για παράδειγμα, το ποσοστό των μη υγιών ανθρώπων, οι οποίοι όντως είναι άρρωστοι.
- Specificity, ή αλλιώς ποσοστό αληθώς αρνητικών αποτελεσμάτων – True Negative Rate. Υπολογίζει το ποσοστό των αρνητικών αποτελεσμάτων τα οποία είναι σωστά κατηγοριοποιημένα ως αρνητικά. Για παράδειγμα, το ποσοστό των υγιών ανθρώπων, οι οποίοι όντως δεν έχουν την ασθένεια.

Συμπληρωματικά αυτών των δύο μετρήσεων είναι το ποσοστό των ψευδώς θετικών (False Positive Rate) και το ποσοστό των ψευδώς αρνητικών (False Negative Rate) αποτελεσμάτων, αντίστοιχα. Η τέλεια κατηγοριοποίηση θα έπρεπε να έχει ποσοστό ευαισθησίας 100%, δηλαδή όλοι οι μη υγιής άνθρωποι να συγκαταλέγονται στην ομάδα των αρρώστων, και ποσοστό ειδικότητας 100%, δηλαδή κανέναν υγιής άνθρωπο να μην συγκαταλέγεται στην ομάδα των αρρώστων. Ωστόσο, όλες οι μέθοδοι κατηγοριοποίησης έχουν ένα ελάχιστο ποσοστό σφάλματος, το οποίο ονομάζεται ποσοστό σφάλματος Bayes.

Έστω ότι πραγματοποιείται μία έρευνα η οποία συνδέει τους συμμετέχοντες με μία ασθένεια. Κάθε άτομο που συμμετέχει, είτε έχει την ασθένεια αυτή είτε όχι. Τα αποτελέσματα, λοιπόν, μπορεί να είναι είτε θετικά (δηλαδή το άτομο έχει την ασθένεια), είτε αρνητικά (δηλαδή το άτομο δεν νοσεί). Ωστόσο, τα αποτελέσματα της έρευνας για κάθε άτομο μπορεί να συμπίπτουν ή να μην συμπίπτουν με την πραγματικότητα. Έτσι έχουμε τους εξής ορισμούς:

- ✓ True Positive = σωστά διαγνώστηκε ως θετικός στην ασθένεια
- ✓ False Positive = εσφαλμένα διαγνώστηκε ως θετικός στην ασθένεια
- ✓ True Negative = σωστά διαγνώστηκε ως αρνητικός στην ασθένεια
- ✓ False Negative = εσφαλμένα διαγνώστηκε ως αρνητικός στην ασθένεια

Λαμβάνοντας τους παραπάνω ορισμούς υπόψη, μπορούμε να ορίσουμε τους εξής τύπους:

- Sensitivity = true positive / (true positive + false negative)
- Specificity = true negative / (true negative + false positive)

2.3 Αλγόριθμοι κατηγοριοποίησης

Στην τεχνητή νοημοσύνη υπάρχουν δύο κατηγορίες αλγορίθμων μάθησης. Η *lazy learning* (τεμπέληδες) και η *eager learning* (πρόθυμοι). Η *lazy learning* είναι μια μέθοδος μάθησης η οποία δέχεται τα δεδομένα εκπαίδευσης, αλλά ξεκινά την επεξεργασία τους μόνο όταν ένα ερώτημα τεθεί στο σύστημα. Αντίθετα, η *eager learning* είναι μία μέθοδος μάθησης στην οποία το σύστημα προσπαθεί να κατηγοριοποιήσει τα δεδομένα εκπαίδευσης πριν λάβει κάποιο ερώτημα το σύστημα.

Οι αλγόριθμοι κατηγοριοποίησης χωρίζονται σε 4 είδη. Αυτά είναι:

- **Αλγόριθμοι κατηγοριοποίησης βασισμένοι σε μέτρα απόστασης:** η βασική ιδέα των αλγορίθμων αυτών είναι ότι το κάθε αντικείμενο του συνόλου δεδομένων βρίσκεται πιο κοντά στα αντικείμενα της ίδιας κατηγορίας παρά σε αντικείμενα που ανήκουν σε άλλες κατηγορίες. Έτσι, μπορούν να χρησιμοποιηθούν μέτρα ομοιότητας (ή απόστασης), ώστε να οριστεί η ομοιότητα για τα αντικείμενα της βάσης δεδομένων. Μία πολύ γνωστή τεχνική που βασίζεται σε αυτό το είδος κατηγοριοποίησης είναι η τεχνική *k* κοντινότερων γειτόνων (*knn* – *k* nearest neighbors). Οι αλγόριθμοι αυτοί ανήκουν στην κατηγορία μάθησης *lazy*. Τα παρακάτω είδη αλγορίθμων ανήκουν στην μάθηση *eager*.
- **Αλγόριθμοι κατηγοριοποίησης βασισμένοι στα Νευρωνικά Δίκτυα:** το νευρωνικό δίκτυο είναι εμπνευσμένο από το κεντρικό νευρικό σύστημα, το οποίο προσπαθεί να προσομοιώσει. Αποτελείται από ένα σύνολο νευρώνων, οι οποίοι είναι συνδεδεμένοι μεταξύ τους, δέχονται ένα σύνολο εισόδων και έπειτα από μία επεξεργασία παράγουν ένα σύστημα έτοιμο για εκπαίδευση και στη συνέχεια για εφαρμογή. Στην πραγματικότητα, τα νευρωνικά δίκτυα είναι ένα συστήματα επεξεργασίας πληροφορίας που αποτελούνται από έναν γράφο και διάφορους αλγορίθμους που προσπελούν αυτόν τον γράφο.
- **Αλγόριθμοι κατηγοριοποίησης βασισμένοι στα δένδρα απόφασης:** τα Δένδρα Απόφασης (*Decision Trees*) χρησιμοποιούνται ευρέως για την κατηγοριοποίηση και την πρόβλεψη δεδομένων. Κατασκευάζεται ένα δένδρο απόφασης, βασισμένο σε ένα σύνολο δεδομένων εκπαίδευσης. Έπειτα, εφαρμόζεται για κάθε πλειάδα της βάσης δεδομένων και προβλέπει μία κατηγοριοποίηση για κάθε μία από αυτές. Συνολικά, η διαδικασία κατηγοριοποίησης χωρίζεται σε δύο φάσεις: (i) την κατασκευή του δένδρου και (ii) την εφαρμογή του στη βάση δεδομένων.

- **Αλγόριθμοι στατικής κατηγοριοποίησης:** δυο γνωστές μέθοδοι στατικής κατηγοριοποίησης είναι η Bayesian και η Παλινδρόμηση (regression). Η Bayesian κατηγοριοποίηση βασίζεται στην στατική θεωρία κατηγοριοποίησης του Bayes και προβλέπει την πιθανότητα μια νέα πλειάδα να ανήκει σε μια από τις προκαθορισμένες κατηγορίες. Η απόδοση αυτού του είδους κατηγοριοποίησης είναι αρκετά υψηλή. Τα προβλήματα παλινδρόμησης ασχολούνται με την εκτίμηση μιας τιμής εξόδου με βάση τις τιμές εισόδου.

2.4 Ηλεκτροεγκεφαλογράφημα (EEG)

Το ηλεκτροεγκεφαλογράφημα είναι η καταγραφή της ηλεκτρικής δραστηριότητας του εγκεφάλου, όπως καταγράφεται από ηλεκτρόδια που τοποθετούνται στο κρανίο. Στην ουσία, πρόκειται για ηλεκτρικά ρεύματα τα οποία παράγονται από κύτταρα του εγκεφάλου. Για την καταγραφή αυτών την ηλεκτρικών σημάτων χρησιμοποιούμε τα ηλεκτρόδια. Έτσι, όσα ηλεκτρόδια τοποθετούνται στο κρανίο, τόσα είναι και τα κανάλια που χρησιμοποιούμε.

Ο εγκέφαλος είναι το κέντρο ελέγχου του ανθρώπινου σώματος. Στέλνει και λαμβάνει εκατομμύρια σήματα κάθε δευτερόλεπτο, μέρα και νύχτα, με την μορφή ορμονών, νευρικών παλμών και χημικών μηνυμάτων. Αυτή η ανταλλαγή πληροφοριών μας επιτρέπει να κινούμαστε, να τρώμε και να σκεφτόμαστε. Εμπόδια, όπως ο όγκος, μπορούν να διακόψουν την φυσιολογική δραστηριότητα του εγκεφάλου, οδηγώντας σε έλλειμμα του κανονικού συλλογισμού, του κινητικού ελέγχου ή της συνείδησης. Η καταγραφή των ηλεκτροεγκεφαλογραφημάτων, επομένως, χρησιμοποιείται και για διαγνωστικούς σκοπούς. Μια πολύ συχνή εφαρμογή είναι η διάγνωση επιληψίας, η οποία προκαλεί μεγάλες ανωμαλίες στο ηλεκτροεγκεφαλογράφημα.

Μία άλλη χρήση των EEG σημάτων είναι η διεπαφή εγκεφάλου-υπολογιστή (brain computer interface), μέσω της οποίας δίνεται η δυνατότητα άμεσης επικοινωνίας του εγκεφάλου και κατ' επέκταση του ανθρώπου, με τον υπολογιστή. Αυτό βρίσκει εφαρμογή σε παιχνίδια αλλά και σε άτομα με κινητικά προβλήματα ή προβλήματα ομιλίας τα οποία μπορούν να κάνουν πράγματα ή να μιλούν μέσω του χειρισμού του υπολογιστή χρησιμοποιώντας την σκέψη τους.

Κεφάλαιο 3

Κ Κοντινότεροι Γείτονες

3.1 KNN κατηγοριοποιητής

Στην αναγνώριση προτύπων, ο αλγόριθμος k κοντινότερων γειτόνων (knn) είναι μία μη παραμετρική μέθοδος που χρησιμοποιείται για κατηγοριοποίηση και για παλινδρόμηση. Και στις δύο περιπτώσεις, η είσοδος αποτελείται από τα k πλησιέστερα παραδείγματα στο χώρο των χαρακτηριστικών. Η έξοδος εξαρτάται από τον σκοπό χρήσης του knn , (κατηγοριοποίηση ή παλινδρόμηση):

- Στην **κατηγοριοποίηση k κοντινότερων γειτόνων (knn classification)**, η έξοδος είναι μέλος μίας κλάσης. Ένα αντικείμενο κατηγοριοποιείται με βάση την πλεονάζουσα κλάση των k γειτόνων του και ανατίθεται στην κλάση, στην οποία οι περισσότεροι από τους k γείτονες του αντικείμενου, ανήκουν. Το k είναι ένας θετικός ακέραιος, σχετικά μικρός αριθμός.
- Στην **παλινδρόμηση k κοντινότερων γειτόνων (knn regression)**, η έξοδος είναι μία τιμή που προσδιορίζει μία ιδιότητα στο αντικείμενο. Η τιμή αυτή είναι η μέση τιμή των τιμών των k πλησιέστερων γειτόνων.

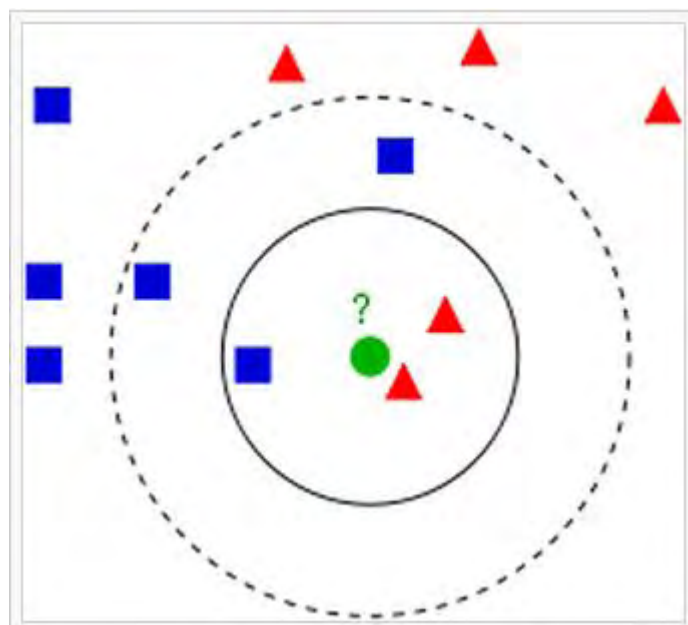
Ο knn αλγόριθμος είναι *lazy* κατηγοριοποιητής ή αλλιώς *instance-based* αλγόριθμος. Προϋποθέτει μόνο την αποθήκευση των γνωστών δεδομένων. Θεωρείται ένας από τους πιο απλούς αλγορίθμους καθώς τα κατηγοριοποιημένα δεδομένα τυγχάνουν επεξεργασίας όταν ένα νέο αντικείμενο δοθεί για κατηγοριοποίηση. Όταν ένα νέο αντικείμενο πρόκειται να κατηγοριοποιηθεί, υπολογίζεται η ομοιότητά του με κάθε ένα από τα αποθηκευμένα δεδομένα. Το αντικείμενο προς κατηγοριοποίηση συγκρίνεται με κάθε ένα από τα γνωστά δεδομένα με μία συνάρτηση απόστασης. Για την κατηγοριοποίηση του, αναζητούνται τα k κοντινότερα αντικείμενα από το σύνολο των κατηγοριοποιημένων δεδομένων που είναι πιο κοντά στο αντικείμενο προς ταξινόμηση. Τέλος, σε αυτό ανατίθεται μια ετικέτα κατηγορίας, από αυτές που ανήκουν τα κοντινότερα αντικείμενα. Η κατηγορία αυτή είναι η πλειοψηφούσα κατηγορία από τους k κοντινότερους γείτονες που έχουν ανακτηθεί.

3.2 Επιλογή του k

Η καλύτερη επιλογή του k εξαρτάται από τα δεδομένα. Μεγάλες τιμές του k μειώνουν την επίδραση του θορύβου στην κατηγοριοποίηση, αλλά κάνουν περισσότερο ευδιάκριτα τα όρια μεταξύ των κλάσεων. Ένα καλό k μπορεί να επιλεγεί με ποικίλες ευριστικές τεχνικές. Στην περίπτωση προβλημάτων με δύο κατηγορίες, είναι προτιμότερο να επιλεγεί περιττός αριθμός για την τιμή του k έτσι ώστε να αποφεύγονται οι ισοψηφίες. Σε περιπτώσεις με περισσότερες κλάσεις, οι πιθανές ισοψηφίες επιλύονται είτε τυχαία είτε αναθέτοντας το νέο αντικείμενο

στην κατηγορία που ανήκει ο εγγύτερος γείτονας. Στον knn αλγόριθμο, η βασική ιδέα είναι πως το προς αναζήτηση αντικείμενο ανήκει στην κατηγορία η οποία είναι η επικρατούσα στους k εγγύτερους γείτονες του.

Η επιλογή του k παίζει σημαντικό ρόλο για τον κατηγοριοποιητή knn, καθώς για διαφορετική τιμή του k τα αντικείμενα ταξινομούνται διαφορετικά. Ένα παράδειγμα φαίνεται στο σχήμα 2. Τα δεδομένα εκπαίδευσης είναι χωρισμένα σε δύο κατηγορίες, τα τρίγωνα και τα τετράγωνα. Ο πράσινος κύκλος είναι το αντικείμεμο προς ταξινόμηση. Στο σχήμα φαίνεται πως για $k=3$, το οποίο υποδεικνύει στο σχήμα ο εσωτερικός από τους δύο ομόκεντρους κύκλους, η πρόβλεψη του αλγορίθμου είναι η κατηγορία τρίγωνο. Ενώ για $k=5$, το οποίο υποδεικνύει ο εξωτερικός κύκλος με τις διακεκομμένες γραμμές, η πρόβλεψη είναι η κατηγορία τετράγωνο.



Σχήμα 2: Κατηγοριοποίηση knn αλγορίθμου

Η επιλογή του k παίζει πολύ σημαντικό ρόλο και στην αποδοτικότητα του αλγορίθμου. Αν η τιμή του k είναι μικρή, το αποτέλεσμα επηρεάζεται από τον θόρυβο των δεδομένων. Ενώ αν η τιμή του k είναι μεγάλη, τότε το αποτέλεσμα περιέχει πολλά αντικείμενα από άλλες κατηγορίες.

3.3 Μέτρα απόστασης

Οι αλγόριθμοι κατηγοριοποίησης που βασίζονται σε μέτρα απόστασης χρησιμοποιούν διάφορες μεθόδους υπολογισμού της απόστασης. Μερικά είδη τέτοιων μεθόδων που χρησιμοποιούνται σαν μέτρα ομοιότητας ανάμεσα σε πλειάδες μιας βάσης δεδομένων είναι:

α. Ευκλείδεια απόσταση: είναι η τεχνική που χρησιμοποιείται από τους περισσότερους κατηγοριοποιητές και ορίζεται ως εξής:

$$d_{ij} = \sqrt{\sum_{k=1}^p (x_{ik} - x_{jk})^2}$$

β. Σταθμισμένη Ευκλείδεια απόσταση:

$$d_{ij} = \sqrt{\sum_{k=1}^p w_k^2 (x_{ik} - x_{jk})^2}$$

Όπου w_k είναι κάποια βάρη που χρησιμοποιούνται για να εξισορροπήσουν την σημαντικότητα των χαρακτηριστικών (διαστάσεων)

γ. Απόσταση Manhattan: μία εναλλακτική της ευκλείδειας απόστασης είναι η Manhattan, όπου η διαφορά της είναι πως δεν παίρνουμε την ρίζα του τετραγώνου αλλά την απόλυτη τιμή και την αθροίζουμε.

$$d_{ij} = \sum_{k=1}^p |x_{ik} - x_{jk}|$$

δ. Απόσταση Minkowski: είναι ένα μέτρο απόστασης που λαμβάνει υπόψη του μεγαλύτερες δυνάμεις από το τετράγωνο.

$$d_{ij} = \left(\sum_{k=1}^p |x_{ik} - x_{jk}|^\lambda \right)^{1/\lambda}$$

όπου λ είναι ένας ακέραιος. Αν $\lambda=1$, τότε έχουμε την απόσταση Manhattan. Αν $\lambda=2$, τότε έχουμε την Ευκλείδεια απόσταση. Ο ρόλος του λ , όταν αυξάνεται, είναι να μεγεθύνει την απόσταση ανάμεσα στα πιο ανόμοια στοιχεία σε σχέση με τα πιο όμοια.

Ωστόσο, υπάρχουν και άλλες, λιγότερο γνωστές, αποστάσεις που έχουν εμφανιστεί στην βιβλιογραφία. Συγκεκριμένα, έχουν προταθεί ως μέτρα ομοιότητας η

απόσταση Camberra, ο συντελεστής Czekanowski, η απόσταση Chebychen ή Maximum.

3.4 Ιδιότητες του k-NN κατηγοριοποιητή

Οι ιδιότητες που είναι άρρητα συνδεδεμένες με τον k-NN κατηγοριοποιητή και τον καθιστούν έναν από τους δημοφιλέστερους του είδους του, είναι:

- Η υλοποίησή του είναι απλή. Απαιτούνται μόνο μία ακέραια τιμή για την παράμετρο k, ένα γνωστό κατηγοριοποιημένο σύνολο και ένα μέτρο απόστασης.
- Είναι εύκολα κατανοητός από τον άνθρωπο.
- Είναι αποδοτικός σε περιπτώσεις όπου τα κατηγοριοποιημένα δεδομένα είναι πολλά ή περιέχουν θόρυβο.
- Μπορεί να χειριστεί περιπτώσεις πολύ-επίπεδων κλάσεων.
- Έχει πολλές εφαρμογές όπως αναγνώριση προτύπων και κατηγοριοποίηση χρονοσειρών.
- Δεν σχεδιάζει κάποιο μοντέλο, οπότε μπορεί να προσαρμοστεί εύκολα σε αλλαγές στα κατηγοριοποιημένα δεδομένα.

3.5 Παλινδρόμηση

Η παλινδρόμηση είναι μια ευρέως χρησιμοποιημένη στατιστική τεχνική μοντελοποίησης για την συσχέτιση μιας εξαρτώμενης με μιας ή περισσότερων μη εξαρτώμενων μεταβλητών. Πιο συγκεκριμένα, χρησιμοποιεί το σύνολο ανεξάρτητων παρατηρήσεων, προκειμένου να προβλέψει τα αποτελέσματα για εξαρτημένες παρατηρήσεις. Μπορεί να χρησιμοποιηθεί και για πρόβλεψη δεδομένων, όταν έχουμε συνεχή μεταβλητή. Η έξοδος της παλινδρόμησης είναι ένα μοντέλο που χρησιμοποιείται για πρόβλεψη των τιμών νέων δεδομένων. Παραδείγματα εφαρμογής της παλινδρόμησης είναι η πρόβλεψη της ζήτησης για ένα νέο προϊόν ή υπηρεσία, συναρτήσεως των δαπανών διαφήμισης.

Τα μοντέλα παλινδρόμησης περιλαμβάνουν τις ακόλουθες μεταβλητές:

- Οι άγνωστες παράμετροι συσχέτισης: β
- Οι ανεξάρτητες μεταβλητές: X
- Η εξαρτημένη μεταβλητή: Y

Έτσι το μοντέλο παλινδρόμησης ορίζεται ως εξής: $Y = F(X, \beta)$

Κεφάλαιο 4

Τεχνητό Νευρωνικό Δίκτυο

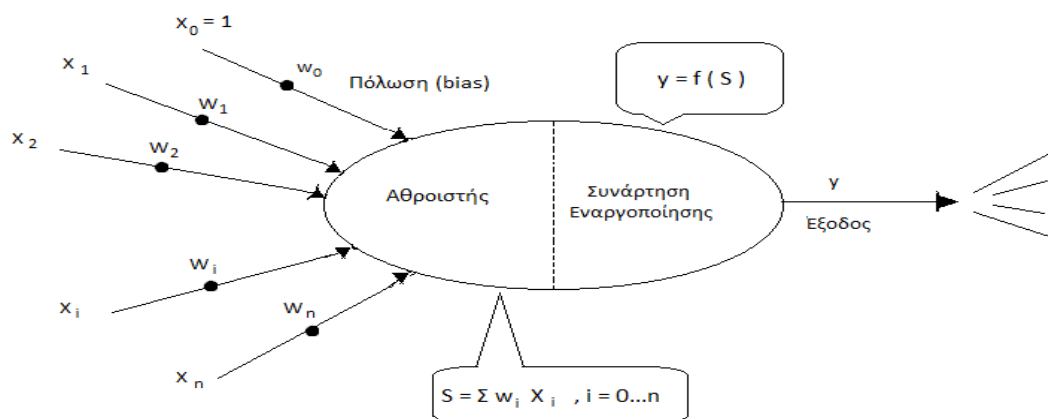
4.1 Τεχνητό Νευρωνικό Δίκτυο

Τα τεχνητά νευρωνικά δίκτυα (ΤΝΔ) βασίζονται στην λειτουργία των βιολογικών νευρωνικών δικτύων, μέσω των οποίων ο άνθρωπος έχει την ικανότητα να θυμάται και να επιλύει προβλήματα. Ο ανθρώπινος εγκέφαλος είναι ένα αρκετά πολύπλοκο υπολογιστικό σύστημα. Αποτελείται από έναν τεράστιο αριθμό νευρώνων οι οποίοι επιτυγχάνουν την εκτέλεση πολύπλοκων υπολογισμών και λειτουργιών, σε ταχύτητες πολύ πιο γρήγορες από τον πιο γρήγορο σημερινό υπολογιστή. Οι νευρώνες αυτοί συνδέονται και επικοινωνούν μεταξύ τους. Κάθε νευρώνας δέχεται κάποια ερεθίσματα (είσοδος) από άλλους νευρώνες, τα οποία επηρεάζουν την κατάσταση του και στέλνει με την σειρά του ερεθίσματα (έξοδος) σε άλλους νευρώνες.

Σύμφωνα με τα παραπάνω, το τεχνητό νευρωνικό δίκτυο ορίζεται ως εξής: “Τεχνητό Νευρωνικό Δίκτυο είναι μια αρχιτεκτονική δομή (δίκτυο) αποτελούμενη από ένα πλήθος διασυνδεδεμένων μονάδων επεξεργασίας (τεχνητοί νευρώνες). Κάθε μονάδα επεξεργασίας χαρακτηρίζεται από εισόδους και εξόδους. Υλοποιεί τοπικά έναν υπολογισμό με βάση τις εισόδους που δέχεται και μεταδίδει το αποτέλεσμα (έξοδος) σε άλλες μονάδες επεξεργασίας με τις οποίες συνδέεται. Οι τιμές των βαρών των συνδέσεων αποτελούν τη γνώση που είναι αποθηκευμένη στο ΤΝΔ και καθορίζουν τη λειτουργικότητά του. Συνήθως ένα ΤΝΔ αναπτύσσει μια συνολική λειτουργικότητα μέσω μιας μορφής εκπαίδευσης (μάθησης)”.

4.2 Μοντέλο Τεχνητού Νευρώνα

Ο τεχνητός νευρώνας (artificial neuron) είναι ένα υπολογιστικό μοντέλο, τα μέρη του οποίου μπορούν να αντιστοιχιστούν με αυτά του βιολογικού νευρώνα. Αποτελεί την βασική μονάδα επεξεργασίας του ΤΝΔ. Στο σχήμα που ακολουθεί απεικονίζεται το μοντέλο ενός τέτοιου νευρώνα.



Σχήμα 3: Μοντέλο Τεχνητού Νευρώνα

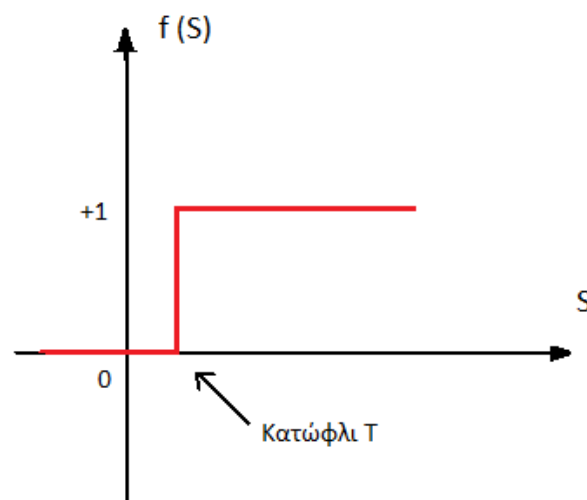
Ένας τεχνητός νευρώνας δέχεται κάποια σήματα εισόδου x_1, x_2, \dots, x_n τα οποία μεταβάλλονται από μια τιμή βάρους w_i (weight). Η τιμή του βάρους μπορεί να είναι αρνητική ή θετική. Το σώμα του τεχνητού νευρώνα χωρίζεται σε δύο μέρη: τον αθροιστή και την συνάρτηση ενεργοποίησης. Ο αθροιστής προσθέτει τα επηρεασμένα από τα βάρη σήματα εισόδου και παράγει την ποσότητα S . Ενώ η συνάρτηση ενεργοποίησης είναι ένα είδος φίλτρου το οποίο διαμορφώνει την τελική τιμή του σήματος εξόδου y , σε συνάρτηση με την ποσότητα S και την τιμή κατωφλίου της συνάρτησης ενεργοποίησης.

Εκτός από τα εισερχόμενα σήματα και τα αντίστοιχα βάρη, ο νευρώνας έχει και κάποιο βάρος w_0 , το οποίο ονομάζεται πόλωση (bias) ή παράγοντας προδιάθεσης του νευρώνα. Η διαφορά αυτού του βάρους από τα υπόλοιπα είναι ότι επιδρά συνεχώς σε μία τιμή εισόδου $x_0 = 1$. Η εισαγωγή της πόλωσης προσδίδει στο νευρώνα επιπλέον υπολογιστικές δυνατότητες μάθησης και προσαρμογής στα δεδομένα εκπαίδευσης.

4.2.1 Τυπικές περιπτώσεις για τη συνάρτηση ενεργοποίησης

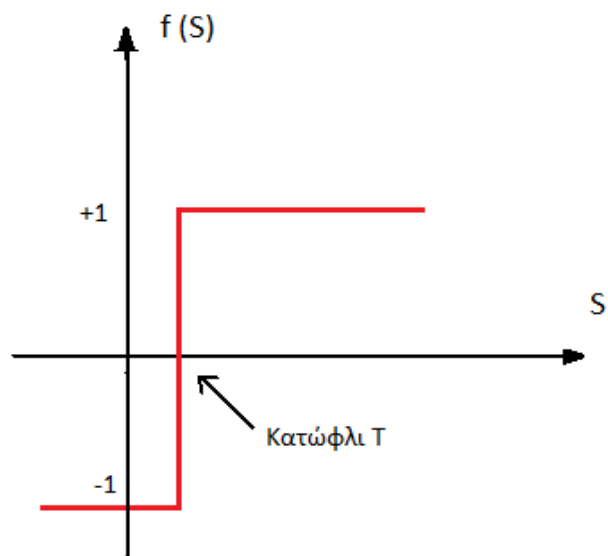
Οι τυπικές περιπτώσεις για την συνάρτηση ενεργοποίησης είναι οι ακόλουθες:

- **Βηματική συνάρτηση:** δίνει στην έξοδο αποτέλεσμα (συνήθως 1) μόνο αν η τιμή που υπολογίζει ο αθροιστής είναι μεγαλύτερη από μία τιμή κατωφλίου T .



Σχήμα 4: Βηματική Συνάρτηση

- **Συνάρτηση πρόσημου:** δίνει στην έξοδο αρνητική (ή θετική) πληροφορία αν η τιμή που υπολογίζει ο αθροιστής είναι μικρότερη (ή μεγαλύτερη) από μία τιμή κατωφλίου T .

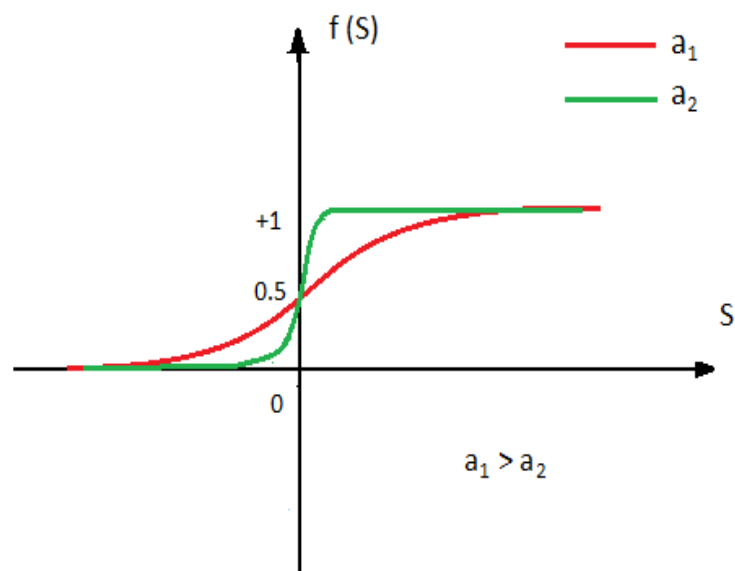


Σχήμα 5: Συνάρτηση Πρόσημου

- **Λογιστική συνάρτηση:** εκφράζεται από την γενική σχέση:

$$f(S) = \frac{1}{1+e^{-aS}}$$

όπου a είναι ένας συντελεστής ρύθμισης της ταχύτητας μετάβασης μεταξύ των δύο ασυμπτωτικών τιμών.

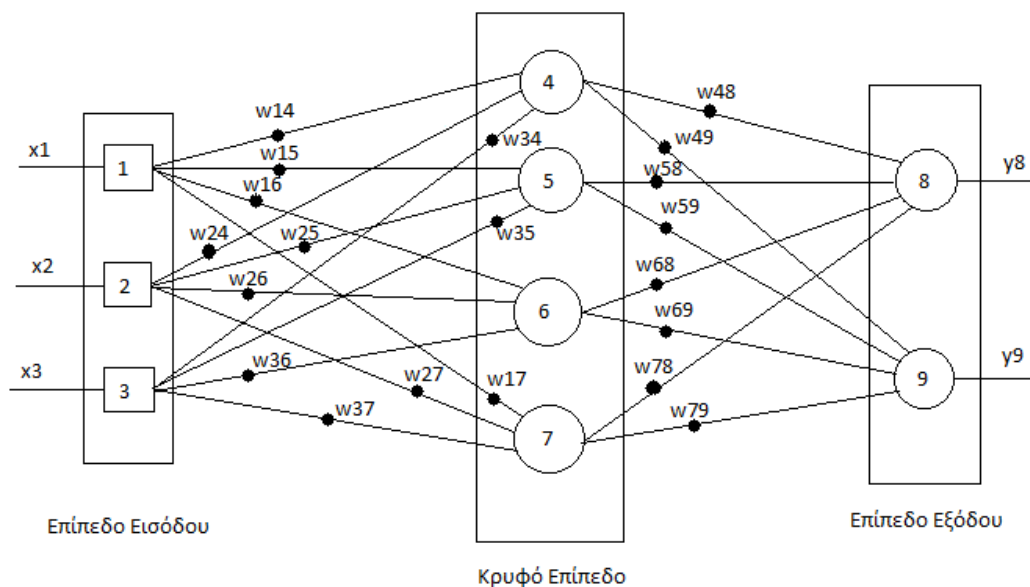


Σχήμα 6: Λογιστική συνάρτηση

4.3 Τεχνητά Νευρωνικά Δίκτυα

Τα τεχνητά νευρωνικά δίκτυα είναι συστήματα που αποτελούνται από ένα πλήθος τεχνητών νευρώνων οργανωμένων σε δομές, δημιουργώντας στρώματα ή αλλιώς επίπεδα (layers). Το πρώτο επίπεδο είναι το επίπεδο εισόδου, στο οποίο γίνεται η εισαγωγή των δεδομένων. Τα στοιχεία του επιπέδου αυτού δεν είναι νευρώνες, καθώς δεν εκτελούν κάποιον υπολογισμό. Στη συνέχεια, ακολουθούν προαιρετικά ένα ή περισσότερα κρυφά επίπεδα, στα οποία γίνονται οι υπολογισμοί για την εκπαίδευση του δικτύου ή την παραγωγή της εξόδου και τέλος υπάρχει το επίπεδο εξόδου, από το οποίο παρουσιάζεται το αποτέλεσμα της κατηγοριοποίησης.

Σε ένα τεχνητό νευρωνικό δίκτυο, οι νευρώνες μπορεί να είναι πλήρως ή μερικώς συνδεδεμένοι. Πλήρως συνδεδεμένοι ονομάζονται αυτοί που συνδέονται με όλους τους υπόλοιπους νευρώνες, ενώ μερικώς συνδεδεμένοι είναι οι νευρώνες σε οποιαδήποτε άλλη περίπτωση.



Σχήμα 7: Τεχνητό Νευρωνικό Δίκτυο με πλήρη διασύνδεση μεταξύ των διαδοχικών επιπέδων.

Στο σχήμα 7 απεικονίζεται ένα τεχνητό νευρωνικό δίκτυο πρόσθιας τροφοδότησης με πλήρη διασύνδεση των νευρώνων. Κατά την λειτουργία του δικτύου ο νευρώνας 4 συμπεριφέρεται ως εξής: Δέχεται και τα τρία σήματα εισόδου από τους νευρώνες του επιπέδου εισόδου, τα οποία πρώτα τροποποιούνται από τα βάρη w_{14} , w_{24} και w_{34} . Στον νευρώνα 4 αθροίζονται τα επεξεργασμένα σήματα μεταξύ τους και

δέχονται την δράση της συνάρτησης ενεργοποίησης. Το αποτέλεσμα στέλνεται στο επίπεδο εξόδου, δηλαδή στους νευρώνες 8 και 9.

4.4 Μάθηση και Ανάκληση

Τα ΤΝΔ χρησιμοποιούν δύο βασικές λειτουργίες, την μάθηση (ή αλλιώς εκπαίδευση) και την ανάκληση. Με την *μάθηση* τροποποιούνται οι τιμές των βαρών του δικτύου, για συγκεκριμένο διάνυσμα εισόδου, έτσι ώστε να παραχθεί συγκεκριμένο διάνυσμα εξόδου, ενώ με την *ανάκληση* υπολογίζεται το διάνυσμα εξόδου για συγκεκριμένο διάνυσμα εισόδου και τιμές βαρών.

Υπάρχουν τρία είδη μάθησης στα ΤΝΔ, τα οποία εξαρτώνται από τον τρόπο που γίνεται η τροποποίηση των βαρών τους κατά την εκπαίδευση του ΤΝΔ.

- *Μάθηση με επίβλεψη (supervised learning)*: στο σύστημα εισάγονται ζευγάρια εισόδου και επιθυμητής εξόδου. Με αυτόν τον τρόπο, παράγεται, με την αρχική κατάσταση βαρών, μία έξοδος η οποία είναι διαφορετική από την επιθυμητή. Έτσι, υπολογίζεται το σφάλμα (error), το οποίο χρησιμοποιείται για την αναπροσαρμογή των βαρών.
- *Βαθμολογημένη μάθηση (graded learning)*: το αν η έξοδος είναι ικανοποιητική ή όχι εξαρτάται από μία αριθμητική κλίμακα. Έπειτα, μέσω αυτού του χαρακτηριστικού προσαρμόζονται και τα βάρη.
- *Μάθηση χωρίς επίβλεψη (unsupervised learning)*: δεν υπάρχει διάνυσμα εξόδου, ενώ το διάνυσμα εισόδου βοηθά στην οργάνωση του δικτύου.

Ένα ΤΝΔ μπορεί να οδηγήσει σε φαινόμενο υποπροσαρμογής ή ατελούς μάθησης (underfitting) ή σε φαινόμενο υπερπροσαρμογής (overfitting). Με τον όρο υποπροσαρμογή εννοούμε ότι το ΤΝΔ αποτυγχάνει να μοντελοποιήσει επιτυχώς τα δεδομένα εκπαίδευσης και οδηγεί σε ατελή μάθηση. Αντίθετα, με τον όρο υπέρπροσαρμογή εννοούμε ότι ένα πολύπλοκο ΤΝΔ μπορεί να μοντελοποιήσει τα δεδομένα εκπαίδευσης, μαζί με τον θόρυβο που πιθανώς φέρουν, τόσο πολύ με αποτέλεσμα να τα απομνημονεύσει. Σε αυτήν την περίπτωση, το δίκτυο δίνει σωστή πρόβλεψη για τα δεδομένα εκπαίδευσης, αλλά για οποιαδήποτε άλλα δεδομένα η πρόβλεψη είναι λανθασμένη.

Για την καταπολέμηση αυτών των καταστάσεων, ο καλύτερος τρόπος είναι να δοθεί στο σύστημα ικανοποιητικός αριθμός δεδομένων εκπαίδευσης. Έτσι, για ΤΝΔ με κρυφά επίπεδα που χρησιμοποιούνται σε προβλήματα κατηγοριοποίησης με δεδομένα εκπαίδευσης που περιέχουν θόρυβο, καλό είναι να υπάρχουν τουλάχιστον 30 φορές περισσότερα δεδομένα εκπαίδευσης από τον αριθμό των βαρών του δικτύου.

Η εκπαίδευση ολοκληρώνεται όταν το κριτήριο ελέγχου της ποιότητας του δικτύου φτάσει σε μια επιθυμητή τιμή. Ως κριτήριο ελέγχου χρησιμοποιείται συνήθως το μέσο σφάλμα ή η μεταβολή του μέσου σφάλματος του συνόλου εκπαίδευσης.

4.5 Μέθοδοι Εκπαίδευσης του Τεχνητού Νευρωνικού Δικτύου

4.5.1 Levenberg-Marquardt Backpropagation (LM)

Ο αλγόριθμος Levenberg – Marquardt είναι μία επαναληπτική τεχνική που εντοπίζει το ελάχιστο μιας συνάρτησης πολλαπλών μεταβλητών. Χρησιμοποιείται για αθροίσματα τετραγώνων μη γραμμικών πραγματικών συναρτήσεων. Ο αλγόριθμος αυτός είναι ένας συνδυασμός της μεθόδου απότομης καθόδου και της μεθόδου Gauss – Newton. Ελέγχει κατά πόσο η λύση διαφέρει από την σωστή. Αν η διαφορά είναι μεγάλη τότε ο αλγόριθμος συμπεριφέρεται όπως η μέθοδος απότομης καθόδου, διαφορετικά, όταν η λύση είναι κοντά στη σωστή, ο αλγόριθμος αλλάζει σε μέθοδο Gauss – Newton.

Ο αλγόριθμος Levenberg – Marquardt σχεδιάστηκε με σκοπό να ελαχιστοποιεί τη συνάρτηση κόστους και να επιτυγχάνει μεγάλες ταχύτητες εκπαίδευσης, χωρίς να χρειάζεται να υπολογιστεί ο πίνακας Hessian. Όταν η συνάρτηση απόδοσης έχει την μορφή αθροίσματος τετραγώνων, τότε ο πίνακας Hessian υπολογίζεται από τον τύπο:

$$H = J^T J$$

και η κλίση υπολογίζεται σύμφωνα με τον παρακάτω τύπο:

$$g = J^T e$$

όπου J είναι ο Ιακωβιανός πίνακας, ο οποίος περιέχει τις πρώτες παραγώγους των λαθών του δικτύου σε σχέση με τα βάρη και τις πολώσεις και e είναι ένα διάνυσμα των λαθών του δικτύου. Ο αλγόριθμος Levenberg – Marquardt χρησιμοποιεί μία λιγότερο πολύπλοκη τεχνική για τον υπολογισμό του πίνακα Hessian:

$$x_{k+1} = x_k - [J^T J + \mu I]^{-1} J^T e$$

όπου μ είναι ένας συντελεστής που αυξάνεται ή μειώνεται ανάλογα με το αν η συνάρτηση απόδοσης έχει αυξηθεί ή μειωθεί μετά από την επανάληψη και I είναι ο διαγώνιος μοναδιαίος πίνακας.

4.5.2 Resilient Backpropagation (RP)

Πρόκειται για έναν αλγόριθμο βελτιστοποίησης πρώτης τάξης, ιδανικό για ευριστική μάθηση που χρησιμοποιείται στα τεχνητά νευρωνικά δίκτυα με ανατροφοδότηση για επιβλεπόμενη μάθηση. Σκοπός του αλγορίθμου Resilient Backpropagation είναι

να φτάσουν τα βάρη και οι πολώσεις στις βέλτιστες τιμές τους πιο γρήγορα. Από το πρόσημο των παραγώγων καθορίζεται η κατεύθυνση της ενημέρωσης του βάρους. Επιρροή στην ενημέρωση των βαρών έχει το σήμα της μερικής παραγώγου και όχι το μέγεθος της παραγώγου. Αν υπάρχει αλλαγή πρόσημου της μερικής παραγώγου στη συνολική συνάρτηση σφάλματος σε σύγκριση με την τελευταία επανάληψη, τότε η τιμή για το εν λόγω βάρος μειώνεται κατά έναν παράγοντα η . Αν δεν υπάρχει αλλαγή πρόσημου τότε το βάρος αυξάνεται κατά έναν παράγοντα η .

4.5.3 Gradient Descent Backpropagation (GD)

Ο αλγόριθμος Gradient Descent ξεκινάει από ένα σημείο και προσπαθεί να εντοπίσει το ολικό ελάχιστο της συνάρτησης σφάλματος των βαρών. Αν και ακούγεται αρκετά απλή διαδικασία, η διακύμανση του σφάλματος είναι αρκετά πιο δύσκολη καθώς ενδέχεται να υπάρχουν πολλαπλά τοπικά ελάχιστα. Ο αλγόριθμος αυτός μεταβάλλει τα βάρη του τεχνητού νευρωνικού δικτύου σύμφωνα με την ακόλουθη σχέση:

$$dw = \mu dE / dw$$

όπου μ είναι ο ρυθμός εκμάθησης και dE είναι η μεταβολή της συνάρτησης σφάλματος.

Η εκμάθηση τερματίζεται όταν ο αριθμός των επαναλήψεων συμπληρωθεί ή όταν το σφάλμα έχει μειωθεί στην επιθυμητή τιμή.

4.5.4 Σύγκριση των μεθόδων εκπαίδευσης

Από έρευνες που έχουν πραγματοποιηθεί, αλλά και από την παρούσα πειραματική μελέτη, διαπιστώθηκε πως ο αλγόριθμος Levenberg-Marquardt Backpropagation (LM) απαιτεί μικρότερο χρόνο εκπαίδευσης και προσφέρει καλύτερη απόδοση μάθησης σε σύγκριση με τους άλλους δύο αλγορίθμους. Επιπλέον, παρατηρήθηκε πως ο αλγόριθμος Resilient Backpropagation (RP) είναι πιο γρήγορος από τον Gradient Descent Backpropagation (GD).

Είναι δύσκολο να γνωρίζουμε ποιος αλγόριθμος θα είναι καλύτερος για κάθε πρόβλημα. Εξαρτάται από πολλούς παράγοντες, συμπεριλαμβανομένου της πολυπλοκότητας του προβλήματος, τον αριθμό των δεδομένων στο σύνολο εκπαίδευσης, καθώς και τον αριθμό των βαρών και των πολώσεων του δικτύου.

4.6 Ιδιότητες των Νευρώνων Δικτύου

Οι ιδιότητες του ΤΝΔ είναι:

- Μάθηση μέσω παραδειγμάτων.
- Μεγάλη ανοχή σε σφάλματα.
- Αναγνώριση προτύπων.
- Μπορούν να θεωρηθούν ως κατανεμημένη μνήμη και ως μνήμη συσχέτισης.

4.7 Νευρωνικά Δίκτυα Πρόσθιας Τροφοδότησης

Τα δίκτυα πρόσθιας τροφοδότησης (feedforward networks) αποτελούν την απλούστερη μορφή ΤΝΔ. Από το όνομα τους καταλαβαίνουμε ότι η ροή πληροφορίας μέσα στο δίκτυο είναι μονής κατεύθυνσης: από την είσοδο προς την έξοδο. Αποτελούνται από ένα επίπεδο εισόδου, ένα επίπεδο εξόδου και προαιρετικά ένα ή περισσότερα ενδιάμεσα ή κρυφά επίπεδα. Χαρακτηριστικό των δικτύων αυτών είναι ότι δεν υπάρχει ανατροφοδότηση της εξόδου ενός νευρώνα προς τους νευρώνες από τους οποίους επηρεάζεται άμεσα ή έμμεσα.

Perceptron

Το perceptron αποτελεί την πιο απλή μορφή δικτύου πρόσθιας τροφοδότησης, χωρίς κρυφά επίπεδα. Είναι μία από τις πρώτες προσεγγίσεις τεχνητού νευρωνικού δικτύου.

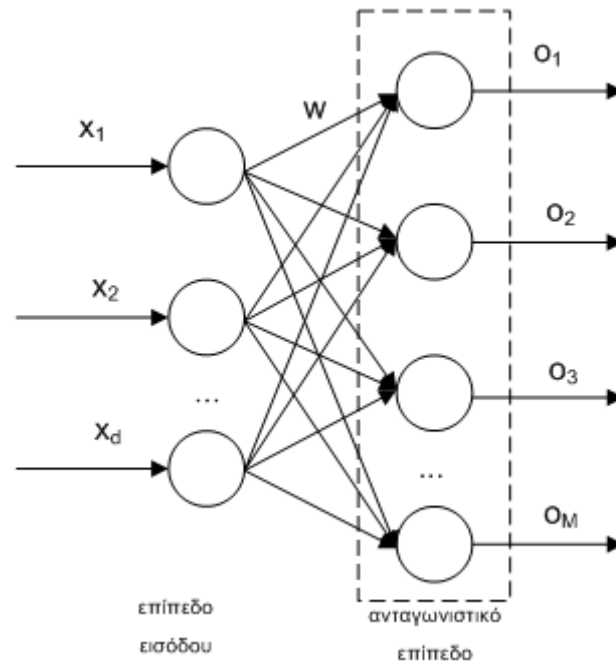
4.8 Νευρωνικά Δίκτυα Βασισμένα σε Ανταγωνισμό

Η έννοια του ανταγωνισμού είναι σημαντική στα ΤΝΔ. Τα ΤΝΔ που είναι βασισμένα στον ανταγωνισμό αποτελούνται από ένα επίπεδο από ανταγωνιστικούς νευρώνες, το οποίο ονομάζεται ανταγωνιστικό επίπεδο. Αυτό που δημιουργεί ανταγωνισμό μεταξύ των νευρώνων είναι ότι κάθε ένας μπορεί να επηρεάσει θετικά, ουδέτερα ή αρνητικά τους υπόλοιπους νευρώνες του δικτύου. Κάθε φορά που το δίκτυο δέχεται μία είσοδο, έχουμε ανταγωνισμό μεταξύ των νευρώνων του ανταγωνιστικού επιπέδου για την ανάδειξη του νευρώνα-νικητή, του οποίου το διάνυσμα βαρών εμφανίζει την μεγαλύτερη ομοιότητα για την συγκεκριμένη είσοδο.

Δίκτυα Kohonen

Τα δίκτυα Kohonen είναι δίκτυα που βασίζονται στον ανταγωνισμό. Αποτελούνται από ένα πλήθος νευρώνων, οι οποίοι είναι τοποθετημένοι σε μία γεωμετρική

τοπολογία (ευθεία, επίπεδο, σφαίρα, κ.α.). Κάθε νευρώνας είναι συνδεδεμένος μέσω βαρών με την είσοδο. Τα δίκτυα Kohonen μπορούν να πραγματοποιήσουν κατηγοριοποίηση, καθώς αντιστοιχούν ένα σήμα της εισόδου με έναν συγκεκριμένο νευρώνα στο επίπεδο εξόδου. Τέλος, αποτελούν μία από τις πιο χαρακτηριστικές περιπτώσεις δικτύου που χρησιμοποιούν μάθηση χωρίς επίβλεψη.

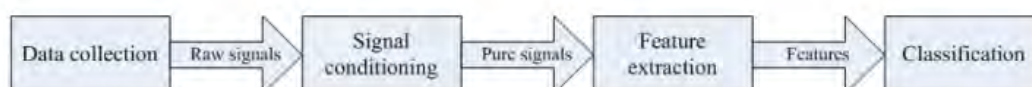


Σχήμα 8: Αρχιτεκτονική ανταγωνιστικού ΤΝΔ

Κεφάλαιο 5

Πειραματική Μελέτη

Η εργασία αυτή επικεντρώνεται στην επεξεργασία των δεδομένων (raw data) και στην κατηγοριοποίηση τους σε κλάσεις με την χρήση τεχνητού νευρωνικού δικτύου ή του αλγορίθμου k κοντινότερων γειτόνων. Αρχικά γίνεται μία επεξεργασία των δεδομένων (signal conditioning), έπειτα πραγματοποιείται εξαγωγή χαρακτηριστικών (feature extraction) και το αποτέλεσμα χρησιμοποιείται για την κατηγοριοποίηση (classification). Η διαδικασία φαίνεται στην παρακάτω εικόνα και αναλύεται παρακάτω.



Σχήμα 9: Συνολική εργασία

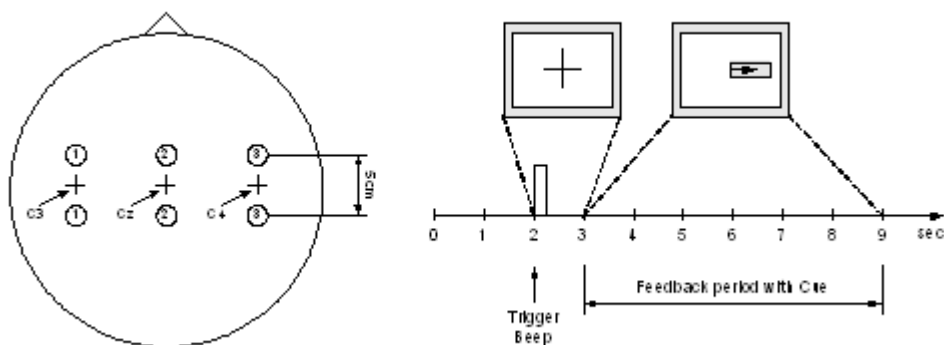
5.1 Συλλογή δεδομένων

Τα δεδομένα που χρησιμοποιήθηκαν για την εργασία αυτή είναι από δύο διαγωνισμούς του BCI-experiment και παραχωρήθηκαν το πρώτο σύνολο από το Department of Medical Informatics, Institute for Biomedical Engineering, University of Technology Graz. (Gert Pfurtscheller), ενώ το δεύτερο από το Department of Neurology, Neurophysics Group του Πανεπιστημίου του Βερολίνου.

Σύμφωνα με το BCI-competition, το πρώτο σύνολο δεδομένων καταγράφηκε από μία φυσιολογική γυναίκα (25 χρονών). Η γυναίκα καθόταν σε μία άνετη καρέκλα με μπράτσα και η διαδικασία ήταν να ελέγξει μία μπάρα μέσω μίας εικόνας για κίνηση του δεξιού ή του αριστερού χεριού, ανάλογα με το σημάδι που εμφανιζόταν σε μία οθόνη. Η σειρά εμφάνισης των στοιχείων για κίνηση του δεξιού ή του αριστερού χεριού ήταν τυχαία.

Το πείραμα αποτελείται από 7 συνεδρίες με 40 δοκιμές η κάθε μία. Όλες οι συνεδρίες πραγματοποιήθηκαν την ίδια ημέρα με λίγα λεπτά διάλειμμα ενδιάμεσα. Δόθηκαν 280 δοκιμές και η κάθε μία διαρκούσε 9 δευτερόλεπτα. Τα πρώτα 2 δευτερόλεπτα ήταν ήσυχια. Στο $t=2\text{sec}$ ακουγόταν ένας θόρυβος που σήμαινε την αρχή της δοκιμασίας και ένας σταυρός '+' εμφανιζόταν στην οθόνη για ένα δευτερόλεπτο. Στο $t=3\text{sec}$ ένα βέλος, δεξιά ή αριστερά, εμφανιζόταν ως σύνθημα. Την ίδια στιγμή, ζητήθηκε στο αντικείμενο να κινήσει μία μπάρα στην κατεύθυνση του συνθήματος. Το υπόλοιπο διάστημα μέχρι και $t=9\text{sec}$, δίνεται στο αντικείμενο προκειμένου να ελέγξει την κίνηση της μπάρας προς τα αριστερά ή προς τα δεξιά, σύμφωνα με την φορά του

βέλους. Χρησιμοποιήθηκαν τρία διπολικά κανάλια ηλεκτροεγκεφαλογράφηματος (EEG), τα οποία μετρήθηκαν πάνω στα C3, Cz και C4. Η δειγματοληψία έγινε στα 128 Hz και υπέστη φιλτράρισμα μεταξύ των τιμών 0.5 και 30 Hz.



Σχήμα 10: Αριστερά: Θέσεις των ηλεκτροδίων. Δεξιά: Σχήμα χρόνου

Τα δεδομένα:

Είναι αποθηκευμένα ως αρχεία του matlab. Η μεταβλητή x_train περιέχει 3 κανάλια EEG, 140 δοκιμές με 9 δευτερόλεπτα η κάθε μία. Η μεταβλητή y_train περιέχει τις κλάσεις '1' και '2' για αριστερή και δεξιά κίνηση αντίστοιχα. Τέλος, η μεταβλητή x_test περιέχει άλλο ένα σύνολο από 140 δοκιμές. Το σύνθημα εμφανιζόταν από $t=3sec$ μέχρι $t=9sec$.

Το δεύτερο σύνολο δεδομένων που χρησιμοποιήθηκε, καταγράφηκε από ένα φυσιολογικό αντικείμενο. Το αντικείμενο και σε αυτό το πείραμα καθόταν σε μία άνετη καρέκλα και είχε τα χέρια στηριγμένα πάνω σε ένα τραπέζι, έτοιμη να πληκτρολογήσει στο πληκτρολόγιο ενός υπολογιστή. Η διαδικασία ήταν να πληκτρολογήσει με τον δείκτη ή το μικρό δάχτυλο τα αντίστοιχα πλήκτρα με σειρά δικής του επιλογής και με τον δικό του χρόνο πληκτρολόγησης. Το πείραμα αποτελείται από τρεις συνεδρίες των 6 λεπτών. Όλες οι συνεδρίες πραγματοποιήθηκαν την ίδια ημέρα με μερικά λεπτά διάλειμμα ενδιάμεσα. Η πληκτρολόγηση πραγματοποιήθηκε με μέσο όρο ταχύτητας 1 πλήκτρο ανά δευτερόλεπτο. Στόχος είναι η πρόβλεψη των επερχόμενων κινήσεων των δακτύλων.

Δεδομένα: Συνολικά καταγράφηκαν 416 μετρήσεις, από τις οποίες οι 316 χρησιμοποιήθηκαν για εκπαίδευση (training data) και οι 100 για δοκιμή (test data). Τα δεδομένα για εκπαίδευση είναι χαρακτηρισμένα με 0 για επερχόμενη κίνηση αριστερού χεριού ή με 1 για επερχόμενη κίνηση δεξιού χεριού. Χρησιμοποιήθηκαν 28 διπολικά κανάλια

ηλεκτροεγκεφαλογραφήματος (EEG), τα οποία μετρήθηκαν πάνω στα F3, F1, Fz, F2, F4, FC5, FC3, FC1, FCz, FC2, FC4, FC6, C5, C3, C1, Cz, C2, C4, C6, CP5, CP3, CP1, CPz, CP2, CP4, CP6, O1, O2. Η δειγματοληψία έγινε στα 1000 Hz και αποτελείται από 500 δείγματα ανά κανάλι.

5.2 Επεξεργασία δεδομένων

Μετά την συλλογή τους, τα δεδομένα χρειάστηκε να υποστούν μία επεξεργασία προκειμένου να δοθούν στις μεθόδους κατηγοριοποίησης με κατάλληλο τρόπο. Πιο συγκεκριμένα, η κάθε διαδικασία κατηγοριοποίησης δέχεται τα στοιχεία προς κατηγοριοποίηση (με βάση τα οποία γίνεται ο διαχωρισμός σε κλάσεις) ως γραμμές ή ως στήλες, ανάλογα. Έτσι, αρχικά χρειάστηκε να γίνει η επεξεργασία των δεδομένων ώστε να είναι έτοιμα να εισαχθούν στους αλγορίθμους κατηγοριοποίησης.

Έπειτα, επειδή ο όγκος των δεδομένων είναι αρκετά μεγάλος, χρειάστηκε να γίνει μία επεξεργασία για την μείωσή του. Κάθε συνεδρία αποτελείται από 140 κινήσεις και η κάθε κίνηση διαρκεί 9 δευτερόλεπτα. Αυτά τα 9 δευτερόλεπτα αντιστοιχούν σε 1152 μετρήσεις. Προκειμένου να μειωθεί ο όγκος αυτός πραγματοποιήθηκε εξαγωγή χαρακτηριστικών (feature extraction). Για το σκοπό αυτό, χρησιμοποιήθηκαν ο μετασχηματισμός Fourier, η μέση τιμή και η διασπορά κάθε κίνησης/δοκιμής.

Ο **Μετασχηματισμός Fourier** είναι ένας μαθηματικός μετασχηματισμός που μετατρέπει μία μαθηματική συνάρτηση του χρόνου $f(t)$, σε μία συνάρτηση συχνότητας, η οποία συμβολίζεται F . Μονάδα μέτρησης του μετασχηματισμού Fourier είναι ο κύκλος/δευτερόλεπτο (Hertz) ή τα ακτίνια ανά δευτερόλεπτο. Ο μετασχηματισμός Fourier ή αλλιώς φάσμα συχνοτήτων της f , είναι και αντίστροφη συνάρτηση. Το f είναι γνωστό ως πεδίο του χρόνου και το F ως πεδίο συχνότητας. Έτσι, μέσω του μετασχηματισμού Fourier μπορεί κανείς να πάει από το πεδίο του χρόνου στο πεδίο συχνότητας ή και το αντίστροφο. Τις περισσότερες φορές η f είναι πραγματική συνάρτηση, ενώ η F είναι μιγαδική. Όπως κάθε μιγαδικός αριθμός, έτσι και η F , περιγράφει το πλάτος και την φάση της αντίστοιχης συνιστώσας συχνότητας.

Η **Μέση τιμή** ή αλλιώς η κεντρική τιμή ενός συνόλου δεδομένων είναι το άθροισμα των παρατηρήσεων, διαιρεμένο με το σύνολο των παρατηρήσεων. Από την άλλη, η **Διακύμανση** μιας τυχαίας μεταβλητής X δηλώνει πόσο συγκεντρωμένες γύρω από την μέση τιμή είναι οι τιμές της τυχαίας μεταβλητής. Δηλώνει, δηλαδή, πόσο μακριά απλώνεται ένα σύνολο αριθμών. Μηδενική διακύμανση σημαίνει πως όλες οι τιμές είναι πανομοιότυπες. Η διακύμανση δεν μπορεί να πάρει αρνητικές τιμές. Μικρή διακύμανση σημαίνει πως τα δεδομένα τείνουν στην μέση τιμή και συνεπώς

τείνουν να μοιάσουν μεταξύ τους. Ενώ μεγάλη διακύμανση σημαίνει ότι τα δεδομένα είναι πολύ απομακρυσμένα από την μέση τιμή και μεταξύ τους. Η διακύμανση είναι μία παράμετρος που περιγράφει είτε την πραγματική κατανομή πιθανότητας ενός συνόλου αριθμών/παρατηρήσεων, είτε την θεωρητική κατανομή πιθανότητας ενός δείγματος αριθμών.

Η είσοδος ή το μέτρο σύγκρισης (inputs) για τις μεθόδους κατηγοριοποίησης (Artificial Neural Network, k-Nearest Neighbors) αποτελείται από τον μετασχηματισμό Fourier, την μέση τιμή και την διασπορά κάθε δοκιμής για το σύνολο των καναλιών που χρησιμοποιούνται (εδώ 3). Επομένως, συνολικά αποτελείται από 9 στήλες για το σύνολο δεδομένων που περιγράφηκε παραπάνω ή 3 επί τον αριθμό των καναλιών για οποιοδήποτε άλλο σύνολο δεδομένων.

5.3 Εφαρμογή Νευρωνικού Δικτύου και Αλγόριθμος k Κοντινότερων Γειτόνων (k-NN)

Το Τεχνητό Νευρωνικό Δίκτυο (ΤΝΔ) δέχεται ως είσοδο τα δεδομένα που είναι ήδη γνωστή η κατηγοριοποίησή τους (inputs) και την τον πίνακα με την κλάση στην οποία ανήκει η κάθε είσοδος (targets). Έτσι, το ΤΝΔ είναι σε θέση να απομνημονεύσει τις τιμές και την κλάση στην οποία αυτές αντιστοιχούν, προκειμένου να προβλέψει σε ποια κλάση ανήκουν μελλοντικές τιμές. Επιπλέον, δέχεται ως είσοδο τα δεδομένα που θέλουμε να κατηγοριοποιήσουμε (test), την μέθοδο την οποία θέλουμε να χρησιμοποιήσουμε για την κατηγοριοποίηση (method), καθώς και τον αριθμό των κρυμμένων επιπέδων, δίνοντας την δυνατότητα στον χρήστη να επιλέξει τις τιμές που θέλει κάθε φορά.

Το τεχνητό νευρωνικό δίκτυο παρουσιάζεται στο παρακάτω γραφικό περιβάλλον.



Σχήμα 11. Τεχνητό Νευρωνικό Δίκτυο

Στο επάνω μέρος του γραφικού περιβάλλοντος απεικονίζεται το νευρωνικό δίκτυο εικονικά, με τις εισόδους, τις εξόδους και τα κρυμμένα επίπεδα. Αμέσως μετά, δίνονται πληροφορίες για τους αλγορίθμους που χρησιμοποιήθηκαν. Πιο συγκεκριμένα, καταγράφεται ο τρόπος που γίνεται ο διαχωρισμός των δεδομένων προς εκπαίδευση. Τα δεδομένα με τα οποία κάνει εκπαίδευση το ΤΝΔ (inputs και targets) χρησιμοποιούνται κατά ένα ποσοστό για εκπαίδευση (train), ένα ποσοστό για επικύρωση (validation) και ένα τρίτο ποσοστό για έλεγχο (test). Μία

συνηθισμένη αναλογία των ποσοστών αυτών είναι 70%, 15% και 15% αντίστοιχα. Στη συνέχεια, αναγράφεται η μέθοδος εκπαίδευσης που χρησιμοποιείται. Στην παρούσα πτυχιακή εργασία διαθέσιμες μέθοδοι είναι οι εξής τρεις: Levenberg-Marquardt Backpropagation (LM), Resilient Backpropagation (RP), Gradient Descent Backpropagation (GD).

Ακολουθεί πληροφορία για την λειτουργία επίδοσης του δικτύου. Πιο συγκεκριμένα, κατά την διάρκεια της εκπαίδευσης του δικτύου ένας δείκτης υπολογίζει τα σφάλματα που γίνονται, κρατώντας κάθε φορά το μικρότερο σφάλμα. Με αυτόν τον τρόπο αν μέχρι την ολοκλήρωση της εκπαίδευσης το δίκτυο έχει αυξήσει τα σφάλματα που πραγματοποιούνται, τότε επιστρέφει πίσω στο σημείο όπου είχε υπολογίσει τον μικρότερο αριθμό σφαλμάτων και δέχεται αυτήν την λύση ως την βέλτιστη εκπαίδευση. Για την μέτρηση της απόδοσης του δικτύου υπάρχουν οι εξής τρόποι:

- ✓ Μέσο Τετραγωνικό Σφάλμα (Mean Squared Error – mse). Χρησιμοποιείται στην παρούσα πτυχιακή εργασία.
- ✓ Άθροισμα των τετραγωνικών σφαλμάτων (Sum of squared error – sse)
- ✓ Μέσο απόλυτο σφάλμα (Mean absolute error – mae)
- ✓ Ρίζα της μέσης τιμής του τετραγωνικού σφάλματος (Root mean squared error - rmse)

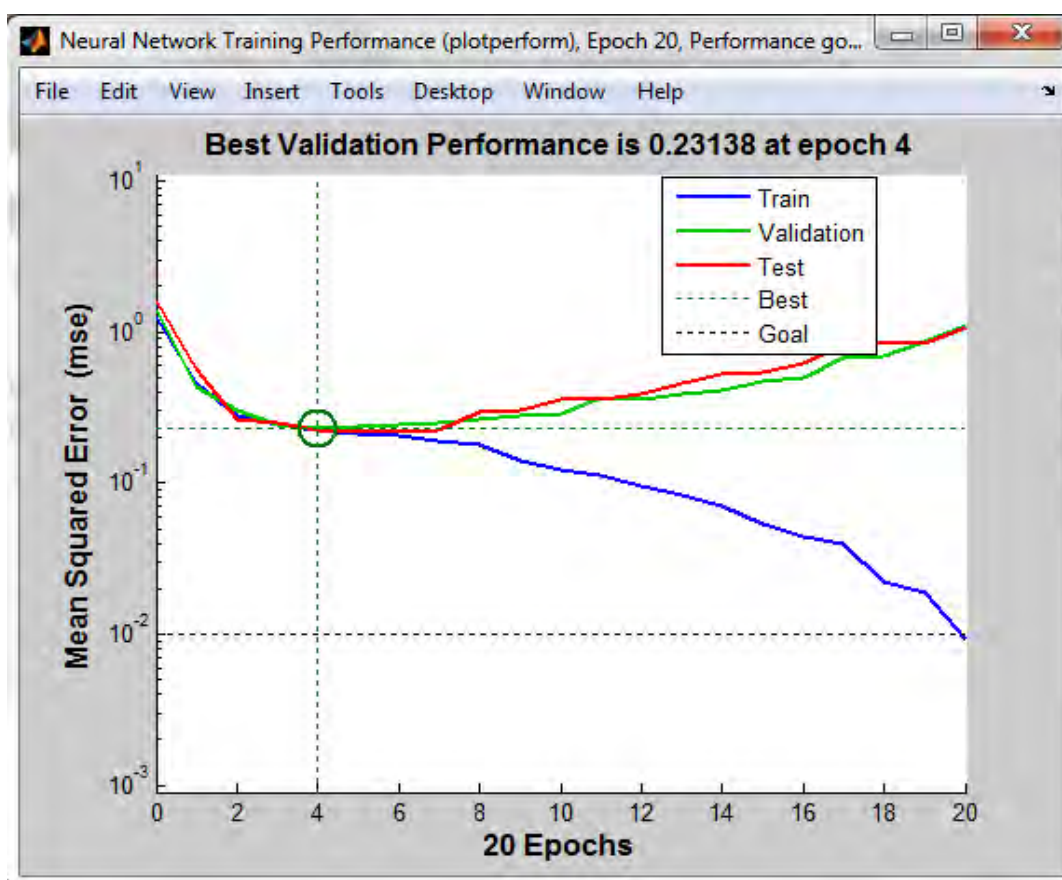
Τελειώνοντας το κουτάκι με τους αλγορίθμους, αναφέρεται το γραφικό περιβάλλον όπου πραγματοποιούνται όλοι αυτοί οι υπολογισμοί, δηλαδή το Matlab.

Στη συνέχεια, ακολουθούν πληροφορίες για την εξέλιξη της εκπαίδευσης του νευρωνικού δικτύου. Πιο αναλυτικά, καταγράφονται ο αριθμός των επαναλήψεων κατά την διάρκεια της εκπαίδευσης (Epoch) και ο χρόνος διεκπεραίωσης της εκπαίδευσης (Time). Να επισημάνω πως ως ένα epoch ορίζεται ένα μόνο πέρασμα μέσα από το σύνολο εκπαίδευσης το οποίο ακολουθείται από τις δοκιμές του συνόλου επαλήθευσης. Μπορούμε να καθορίσουμε, δηλαδή, στις πόσες επαναλήψεις ή σε πόσο χρόνο να σταματήσει η εκπαίδευση. Έπειτα, αναγράφεται η απόδοση (Performance), δηλαδή το κατώτερο σημείο σφαλμάτων στο οποίο όταν το σύστημα φτάσει, τότε σταματάει. Ορίζεται η αντοχή της καθόδου (Gradient). Δηλαδή, τα όρια στα οποία κυμαίνεται η συνάρτηση του σφάλματος. Αν η τιμή του gradient γίνει μικρότερη από το καθορισμένο όριο τότε η εκπαίδευση σταματάει. Με άλλα λόγια, όταν η συνάρτηση του σφάλματος δεν αλλάζει κατά πολύ τότε η εκπαίδευση σταματάει καθώς δεν πρόκειται να αλλάξει πολύ από το σημείο αυτό και πέρα. Επίσης, καταγράφεται ο όρος ορμής (Momentum term - Mu), ο οποίος χρησιμοποιείται για τον έλεγχο της διαδικασίας ενημέρωσης του βάρους. Δηλαδή, ελέγχει πόσο πολύ αλλάζουν τα βάρη σε κάθε επανάληψη και σταματά μόλις το τρέχον βάρος με το επόμενο βρίσκονται σε συμφωνία. Με αυτόν τον τρόπο βοηθάει το δίκτυο να βρει μία λογική λύση σε λιγότερες επαναλήψεις κατά την εκπαίδευση.

Τέλος, απεικονίζεται ο έλεγχος επικύρωσης (Validation Checks) στον οποίο ορίζεται ο αριθμός των επαναλήψεων κατά τις οποίες, αν το σύστημα αποτύχει να μειώσει το σφάλμα επικύρωσης, τότε θα σταματήσει η εκπαίδευση.

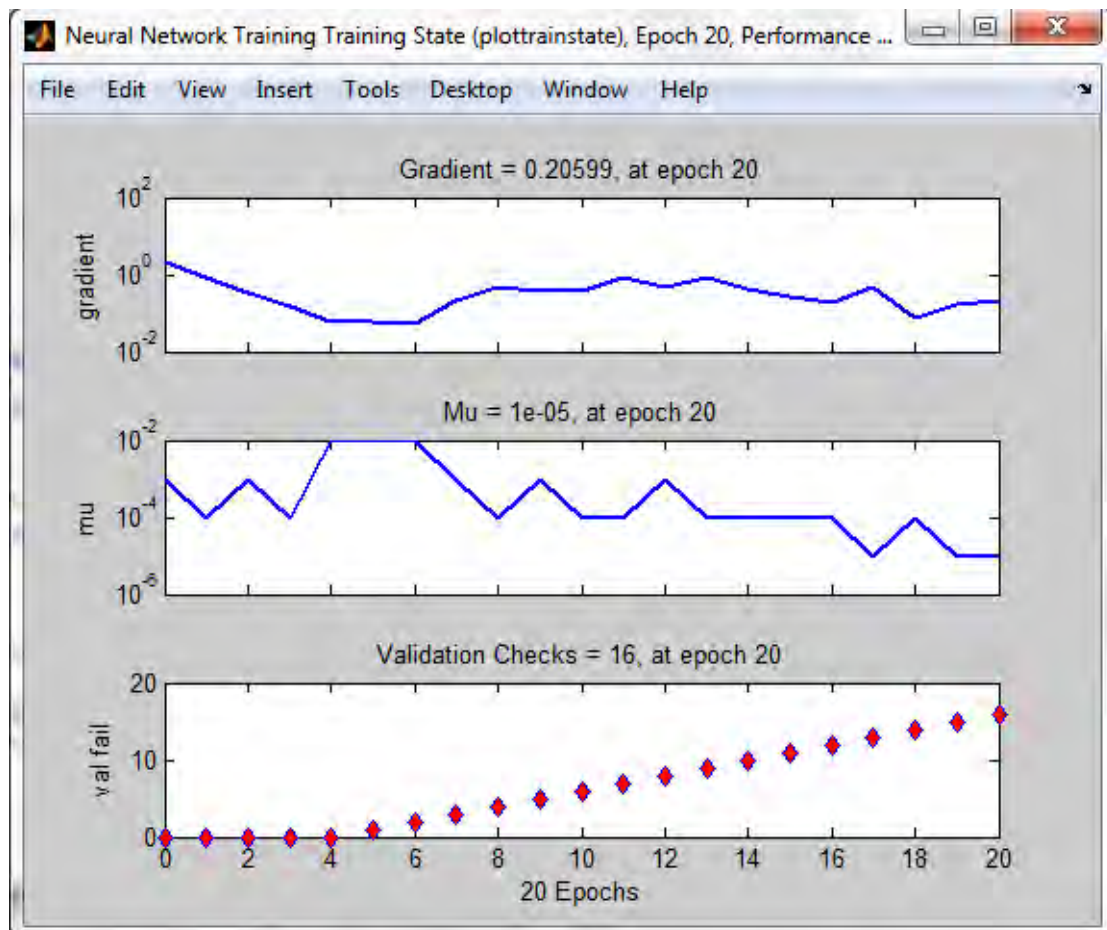
Όλα αυτά έχουμε την δυνατότητα να τους δίνουμε εμείς αρχικές τιμές με σκοπό να πάρουμε το καλύτερο δυνατό αποτέλεσμα στην κατηγοριοποίηση. Έπειτα από συνεχείς αλλαγές, τα νούμερα που έχουν δοθεί στο σύστημα (και φαίνονται στο Σχήμα 10) απέδιδαν την καλύτερη απόδοση.

Επιπρόσθετα, το επόμενο και τελευταίο πλαίσιο στο γραφικό περιβάλλον του δικτύου δίνει την δυνατότητα στον χρήστη να απεικονίσει γραφικά την εκπαίδευση του δικτύου, για όσες επαναλήψεις επιθυμεί. Πρώτα από όλα με το κουμπί Performance απεικονίζεται η καμπύλη εξέλιξης των σφαλμάτων που είναι για εκπαίδευση (train), για επικύρωση (validation) και για δοκιμή (test) με διαφορετικό χρώμα. Υποδεικνύει την επανάληψη κατά την οποία η απόδοση επικύρωσης φτάνει στο ελάχιστο. Η εκπαίδευση παρόλα αυτά συνεχίζεται μέχρι των αριθμό των επαναλήψεων που έχει οριστεί. Οι καμπύλες του test και του validation μοιάζουν πολύ. Αν η καμπύλη του test αυξηθεί απότομα πριν την καμπύλη του validation τότε είναι πιθανό να έχει συμβεί υπερπροσαρμογή.



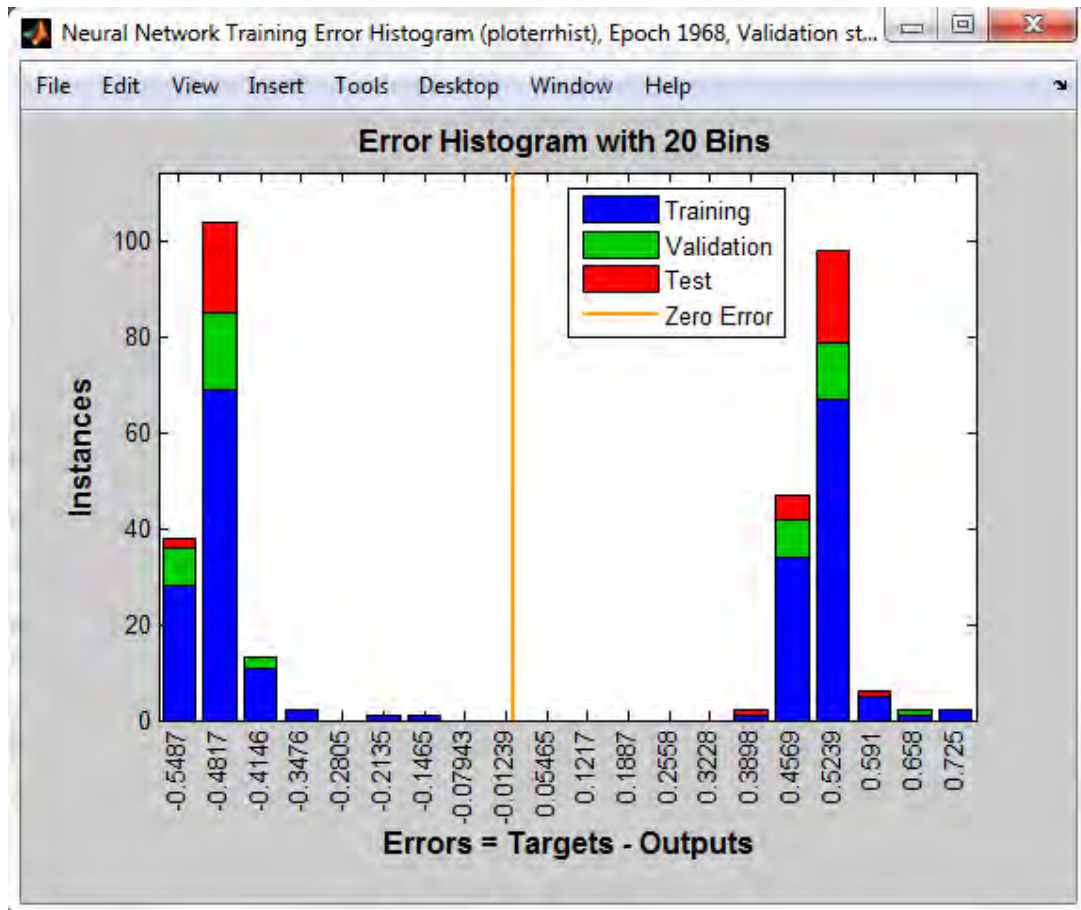
Σχήμα 12: Best Validation Performance

Το επόμενο κουμπί είναι το Training state το οποίο παρουσιάζει σε γράφημα τις τιμές του όρου ορμής (Mu), του ελέγχου επικύρωσης (validation checks) και της αντοχής της καθόδου (Gradient) για το σύνολο των επαναλήψεων που πραγματοποιήθηκαν. Οι όροι αυτοί αναλύονται λίγο πιο πάνω.



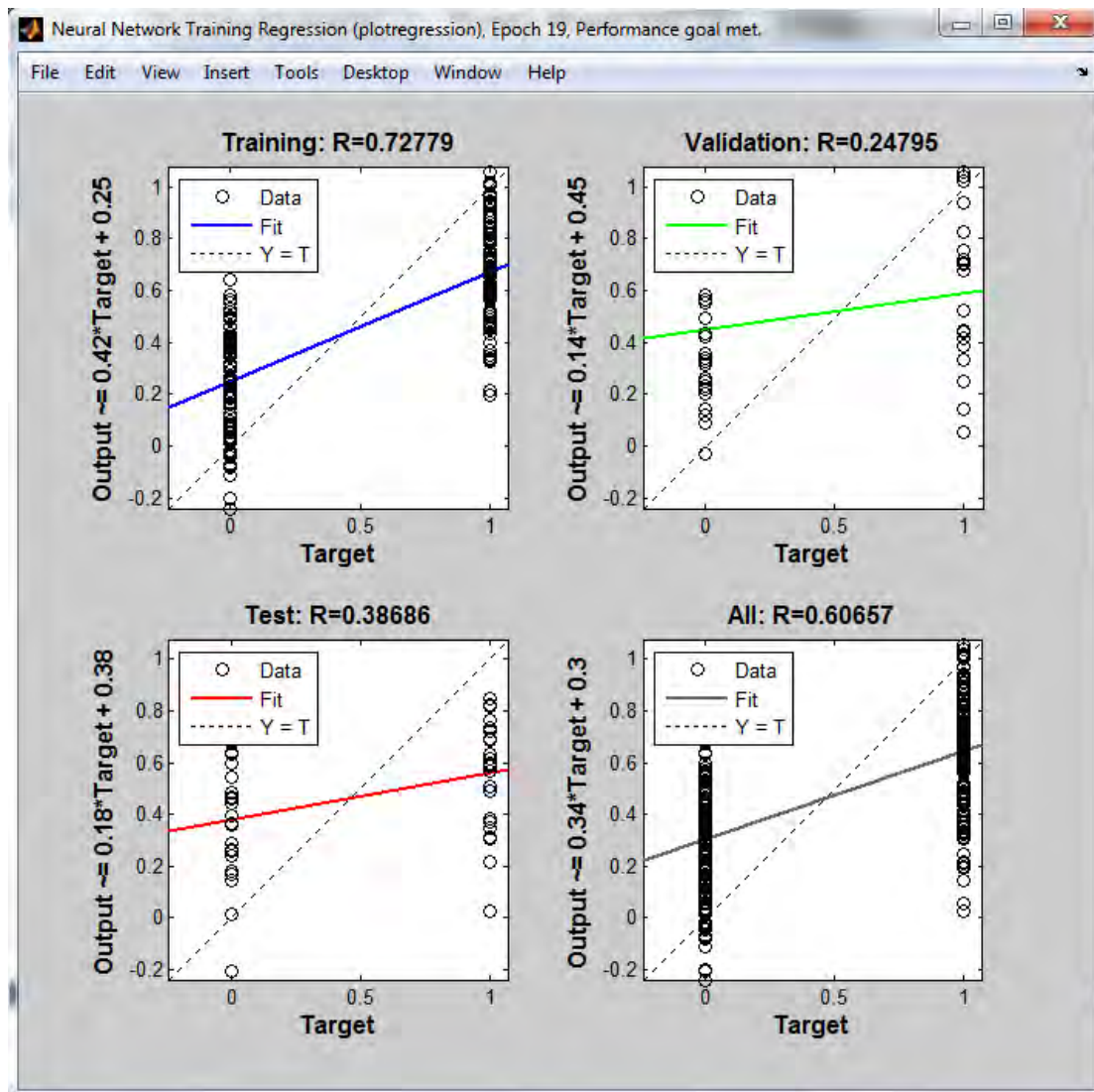
Σχήμα 13: Training State

Στη συνέχεια, το κουμπί που ακολουθεί είναι το Error Histogram, το οποίο απεικονίζει το ιστόγραμμα των σφαλμάτων που έγιναν κατά την εκπαίδευση του νευρωνικού δικτύου και χρησιμοποιεί διαφορετικό χρώμα για τα δεδομένα εκπαίδευσης, τα δεδομένα επικύρωσης και τα δεδομένα δοκιμής.



Σχήμα 14: Error Histogram

Το τελευταίο κουμπί είναι το Regression, το οποίο αποτελεί μία ανάλυση της απόκρισης του δικτύου. Πιο συγκεκριμένα, εκτελεί μία γραμμική παλινδρόμηση μεταξύ των εξόδων του δικτύου και των αρχικών δεδομένων απαντήσεων (targets) με σκοπό να απεικονίσει την απόδοση του δικτύου για τα δεδομένα εκπαίδευσης, τα δεδομένα επικύρωσης και τα δεδομένα δοκιμής ξεχωριστά, καθώς και για το σύνολο των δεδομένων σε μία γραφική παράσταση. Η διακεκομμένη γραμμή σε κάθε γραφική παράσταση αντιπροσωπεύει τον τέλειο στόχο. Η συνεχής γραμμή αντιπροσωπεύει την καλύτερη προσαρμογή γραμμικής παλινδρόμησης που πραγματοποιήθηκε μεταξύ των αποτελεσμάτων και των στόχων. Η τιμή R είναι μία ένδειξη της σχέσης μεταξύ των εξόδων και των στόχων. Εάν το $R = 1$, τότε υπάρχει μία ακριβής γραμμική σχέση μεταξύ των αποτελεσμάτων και των στόχων. Εάν το R είναι κοντά στο μηδέν, τότε δεν υπάρχει γραμμική σχέση μεταξύ των αποτελεσμάτων και των στόχων.



Σχήμα 15: Regression

Τέλος, το γραφικό περιβάλλον του νευρωνικού δικτύου αποδίδει τον λόγο για τον οποίο σταμάτησε η εκπαίδευση. Ο λόγος κάθε φορά είναι ένας από τα στοιχεία που φαίνονται στο πλαίσιο Progress και τα οποία μπορούμε κάθε φορά να τα αλλάξουμε εμείς, πειράζοντας τον κώδικα. Στο παράδειγμα που φαίνεται στο Σχήμα 10, η εκπαίδευση σταμάτησε επειδή τα σφάλματα ξεπέρασαν το κατώτερο όριο που είχε δοθεί στο σύστημα.

Από την άλλη μεριά, ο αλγόριθμος k- κοντινότερων γειτόνων δέχεται την είσοδο (inputs), την κλάση στην οποία ανήκουν τα στοιχεία της εισόδου (targets) και τον πίνακα τον οποίο θέλουμε να κατηγοριοποιήσουμε (test) και βρίσκει τους k κοντινότερους γείτονες του πίνακα προς κατηγοριοποίηση σε σχέση με την είσοδο.

Με αυτόν τον τρόπο, αναθέτει σε κάθε γραμμή του συνόλου προς κατηγοριοποίηση (test) την κλάση στην οποία ανήκουν οι περισσότεροι από τους k γείτονες της. Ο αλγόριθμος k κοντινότερων γειτόνων, σε αντίθεση με το τεχνητό νευρωνικό δίκτυο, δεν εκπαιδεύεται. Αυτό που κάνει είναι να συγκρίνει τις τιμές που θέλει να κατηγοριοποιήσει με τις τιμές που έχει ως δεδομένες χρησιμοποιώντας μία μέθοδο μέτρησης της απόστασης, έτσι ώστε να αποφασίσει σε ποια κατηγορία θα ταξινομήσει το κάθε δεδομένο προς κατηγοριοποίηση. Στην μέθοδο αυτή δεν χρησιμοποιούμε κάποιες επιμέρους μεθόδους για την εκτέλεσή της. Το μόνο που αλλάζουμε είναι ο αριθμός των k κοντινότερων γειτόνων που θα λαμβάνει υπόψη της η μέθοδος για να εκτελεί την σύγκριση.

5.4 Υπολογισμός της ακρίβειας κάθε αλγορίθμου

Στην παρούσα εργασία, για τον υπολογισμό της ακρίβειας χρησιμοποιήθηκε ένα σύνολο δεδομένων για εκπαίδευση και για δοκιμή από το νευρωνικό δίκτυο. Με αυτόν τον τρόπο η εκτίμηση της ακρίβειας είναι αρκετά ακριβής, καθώς χρησιμοποιούνται τα ίδια δεδομένα και για εκπαίδευση και για πρόβλεψη. Επιπλέον, στην κατηγοριοποίηση με τον k -NN αλγόριθμο χρησιμοποιήθηκε το ίδιο σύνολο δεδομένων ως πρότυπο και για κατηγοριοποίηση.

Έπειτα, υπολογίστηκε το σύνολο των σφαλμάτων που έγινε κατά την πρόβλεψη, συγκρινόμενο με τον ήδη γνωστό πίνακα με τις κλάσεις. Με αφαίρεση του συνόλου των σφαλμάτων από το σύνολο των δεδομένων προς κατηγοριοποίηση βρέθηκε η ακρίβεια των αλγορίθμων, η οποία αποδίδεται στους παρακάτω τύπους:

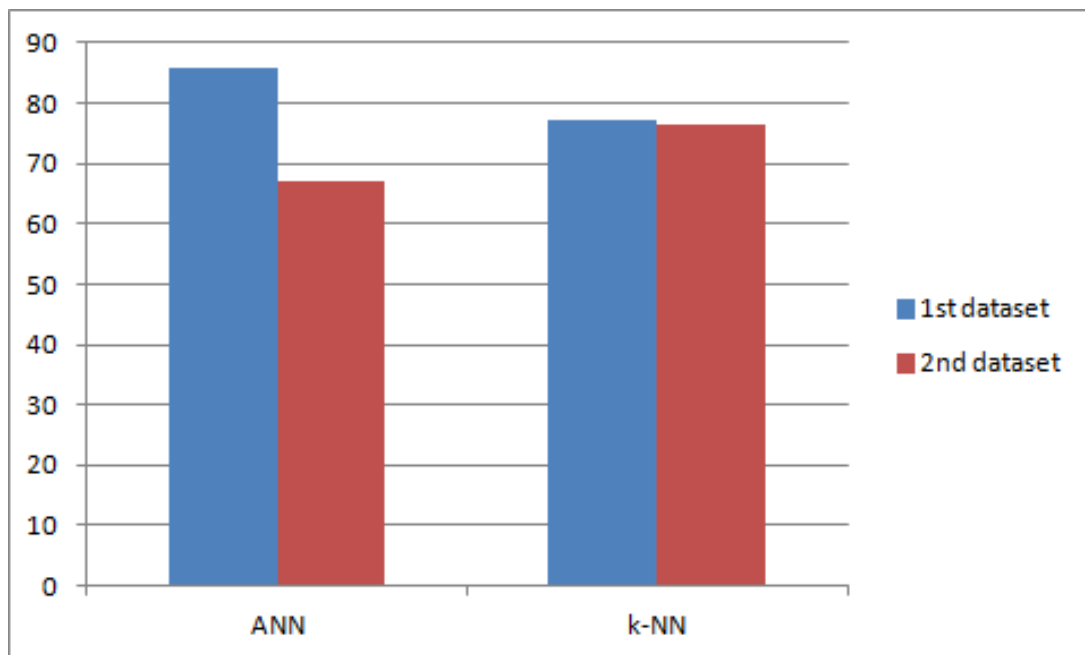
$$errors = abs(targets - class)$$

$$accuracy = [(\#data - errors) * 100] / \#data$$

Στον παρακάτω πίνακα και στο γράφημα που ακολουθεί απεικονίζεται η ακρίβεια για τα δύο σύνολα δεδομένων που αναλύθηκαν παραπάνω.

Accuracy		
	ANN	k-NN
1st dataset	85,71	77,14
2nd dataset	67,08	76,6

Σχήμα 16: Η ακρίβεια των δύο αλγορίθμων για τα δύο σύνολα δεδομένων.



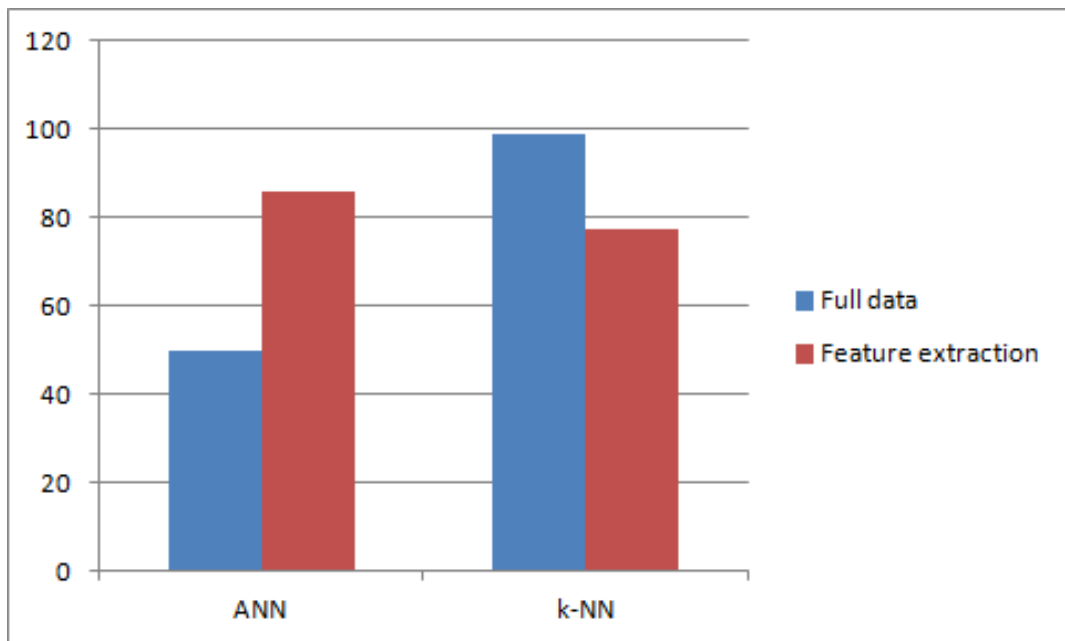
Σχήμα 17: Η ακρίβεια των δύο αλγορίθμων για τα δύο σύνολα δεδομένων σε γράφημα.

Όπως φαίνεται και στο διάγραμμα, η κατηγοριοποίηση δεδομένων με το τεχνητό νευρωνικό δίκτυο είναι κατά λίγο καλύτερη από αυτήν που προσφέρει ο αλγόριθμος k-NN. Η διαφορά μεταξύ τους όμως είναι μικρή. Για τον k-NN αλγόριθμο η ακρίβεια για το πρώτο σύνολο δεδομένων είναι 77.14% και για το δεύτερο 70.56%, ενώ για το τεχνητό νευρωνικό δίκτυο, το πρώτο σύνολο δεδομένων κατηγοριοποιείται με ακρίβεια 85.71% και το δεύτερο με ακρίβεια 67,08%. Συνολικά, επομένως, το τεχνητό νευρωνικό δίκτυο ταξινομεί με μεγαλύτερη ακρίβεια τα δεδομένα. Ωστόσο, το συμπέρασμα αυτό δημιουργήθηκε δίχως να ληφθούν υπόψη άλλοι παράγοντες εκτός της ακρίβειας. Σημαντικό ρόλο διαδραματίζει και ο χρόνος κατηγοριοποίησης, δηλαδή η ταχύτητα εκτέλεσης του αλγορίθμου. Επίσης, η σωστή πρόβλεψη με ελλιπή δεδομένα ή δεδομένα με θόρυβο είναι ένας ακόμα παράγοντας για την εγκυρότητα των μεθόδων κατηγοριοποίησης.

Έπειτα, έγινε μία σύγκριση των δύο μεθόδων κατηγοριοποίησης όταν έχουν ως είσοδο τα δεδομένα που έχουν υποστεί εξαγωγή χαρακτηριστικών την μία φορά και το σύνολο των δεδομένων (χωρίς εξαγωγή χαρακτηριστικών) την δεύτερη φορά. Τα αποτελέσματα φαίνονται στα διαγράμματα παρακάτω.

	ANN	k-NN
Full data	50	99
Feature extraction	85,71	77,14

Σχήμα 18: Ακρίβεια των δύο μεθόδων για δεδομένα με εξαγωγή χαρακτηριστικών και για το σύνολο των δεδομένων.



Σχήμα 19: Ακρίβεια των δύο μεθόδων για δεδομένα με εξαγωγή χαρακτηριστικών και για το σύνολο των δεδομένων σε γράφημα.

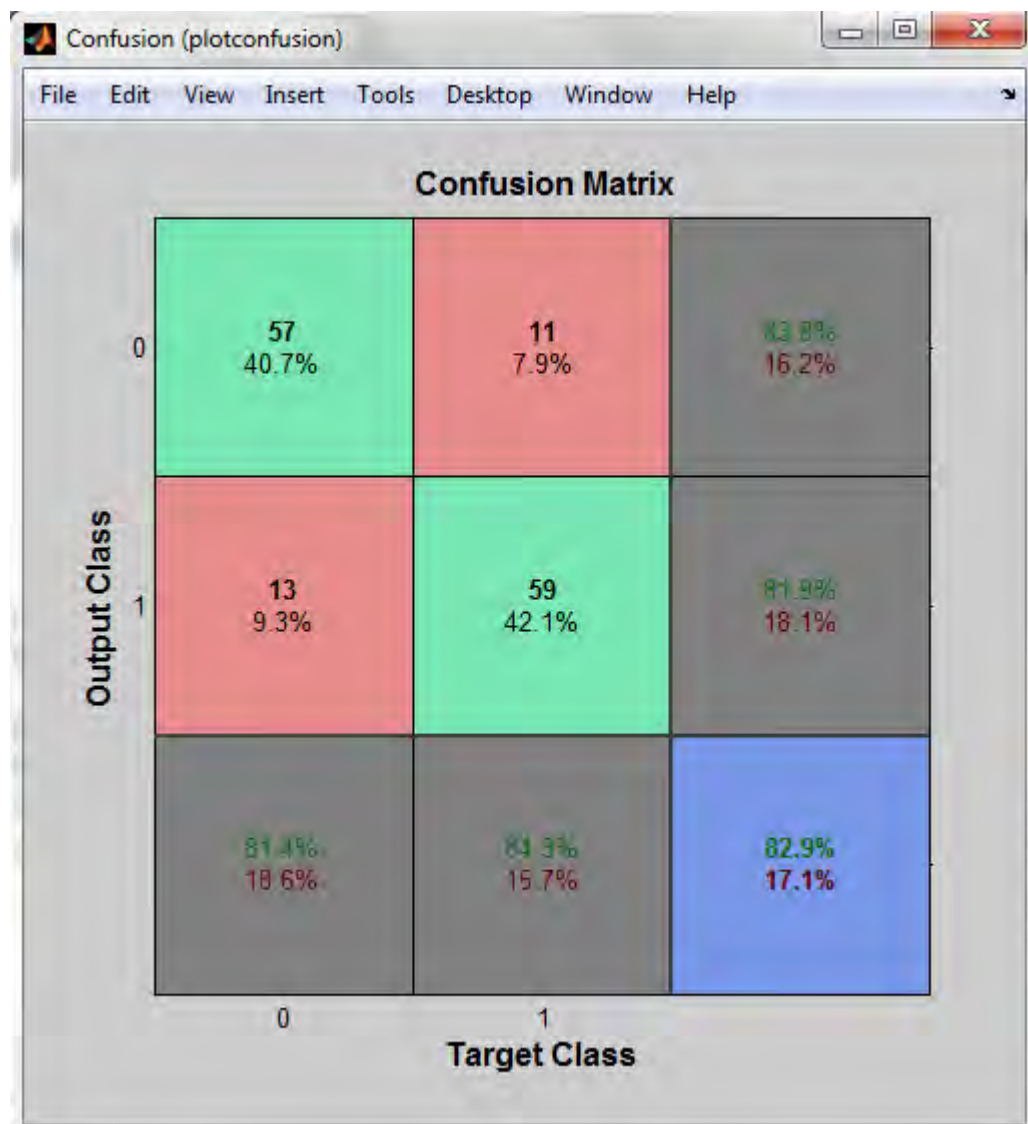
Παρατηρούμε ότι η διαφορά στην αποδοτικότητα των αλγορίθμων είναι αρκετά μεγάλες. Αρχικά, στο τεχνητό νευρωνικό δίκτυο το σύνολο των δεδομένων κατηγοριοποιείται με αποδοτικότητα 50%, το οποίο είναι ένα πολύ χαμηλό ποσοστό. Στην ουσία, είναι σαν να ρίχνουμε κέρμα για να μαντέψουμε την κατηγορία στην οποία ανήκει το κάθε σύνολο ηλεκτροεγκεφαλογραφήματος. Αντίθετα, η κατηγοριοποίηση των δεδομένων που έχουν υποστεί εξαγωγή χαρακτηριστικών (feature extraction) έχει πολύ μεγαλύτερη αποδοτικότητα 85.71%, το οποίο γενικά είναι ένα πολύ καλό ποσοστό.

Απεναντίας, η κατηγοριοποίηση με τον αλγόριθμο k κοντινότερων γειτόνων παρουσιάζει αρκετά καλύτερο ποσοστό αποδοτικότητας όταν χρησιμοποιούμε ως είσοδο ολόκληρο το σύνολο των δεδομένων συγκριτικά με το σύνολο έπειτα από την εφαρμογή της εξαγωγής χαρακτηριστικών. Συγκεκριμένα, το ποσοστό επιτυχίας για το σύνολο των δεδομένων είναι 99% (σχεδόν τέλεια κατηγοριοποίηση), ενώ για τα δεδομένα έπειτα από εξαγωγή χαρακτηριστικών είναι 77.14%. Η διαφορά είναι αρκετά σημαντική. Αυτό είναι αναμενόμενο, όμως, καθώς λαμβάνοντας υπόψη τον τρόπο με τον οποίο λειτουργεί ο αλγόριθμος k κοντινότερων γειτόνων, όσο περισσότερα δεδομένα έχει για να συγκρίνει τα δεδομένα προς κατηγοριοποίηση και μέσω αυτών να αποφασίσει την κατηγορία στην οποία αυτά ανήκουν, τόσο περισσότερες είναι οι πιθανότητες για να κάνει καλύτερη κατηγοριοποίηση.

Ωστόσο, ένας άλλος παράγοντας εξίσου σημαντικός είναι ο χρόνος εκτέλεσης. Στα δεδομένα που έχουν υποστεί εξαγωγή χαρακτηριστικών, ο χρόνος εκτέλεσης είναι

σημαντικά πολύ μικρότερος σε σύγκριση με τον χρόνο εκτέλεσης των μεθόδων για το σύνολο των δεδομένων.

Όπως έχει αναφερθεί και στο πρώτο κεφάλαιο, σημαντικό ρόλο στην κατηγοριοποίηση δεν παίζει μόνο η αποτελεσματικότητα αλλά και το ποσοστό των αληθώς θετικών απαντήσεων (sensitivity) καθώς και το ποσοστό των αληθώς αρνητικών απαντήσεων (specificity). Στην παρακάτω εικόνα απεικονίζονται τα ποσοστά αυτά για το τεχνητό νευρωνικό δίκτυο.



Σχήμα 20: Confusion matrix.

Πιο αναλυτικά, στο πλάι δίνονται οι τιμές που απέδωσε το τεχνητό νευρωνικό δίκτυο και αναγράφονται και οι κλάσεις (εδώ 0 και 1), ενώ στο κάτω μέρος δίνονται οι τιμές που ήταν ήδη γνωστές και με τις οποίες έγινε η σύγκριση. Στο πρώτο κελί

λοιπόν, απεικονίζεται το ποσοστό των τιμών που αληθώς το TND τις κατηγοριοποίησε στην κλάση 0 (True Positive). Στο δεύτερο κελί καταγράφεται το ποσοστό των τιμών που ψευδώς το TND τις κατηγοριοποίησε στην κλάση 0, ενώ στην πραγματικότητα ανήκουν στην κλάση 1 (False Positive). Στο τρίτο κελί απεικονίζεται το ποσοστό των αληθώς θετικών τιμών ως προς το σύνολο των τιμών που κατηγοριοποιήθηκαν ως θετικά (Positive predictive value = True Positive / (True Positive + False Positive)). Στην δεύτερη γραμμή, τα δύο πρώτα κελιά απεικονίζουν το ποσοστό των τιμών που ψευδώς κατηγοριοποιήθηκαν αρνητικά (False Negative) και αληθώς κατηγοριοποιήθηκαν αρνητικά (True Negative), αντίστοιχα. Ενώ, στο αμέσως επόμενο κελί, παρουσιάζεται το ποσοστό των ψευδώς αρνητικών τιμών ως προς το σύνολο των τιμών που κατηγοριοποιήθηκαν ως αρνητικά (Negative predictive value = False Negative / (False Negative + True Negative)). Στην τρίτη και τελευταία γραμμή, τα δύο πρώτα κελιά παρουσιάζουν τα ποσοστά των αληθώς θετικών τιμών (True Positive Rate) και των αληθώς αρνητικών τιμών (True Negative Rate), αντίστοιχα. Τέλος, το τελευταίο κελί παρουσιάζει την ακρίβεια της κατηγοριοποίησης, η οποία στην ουσία είναι το σύνολο των αληθώς κατηγοριοποιημένων κελιών (Accuracy = True Positive Rate + True Negative Rate).

Σε ότι αφορά τον αλγόριθμο k κοντινότερων γειτόνων, ο υπολογισμός των ποσοστών sensitivity και specificity, δεν απεικονίζεται γραφικά, αλλά παρουσιάζεται στο command window του Matlab.

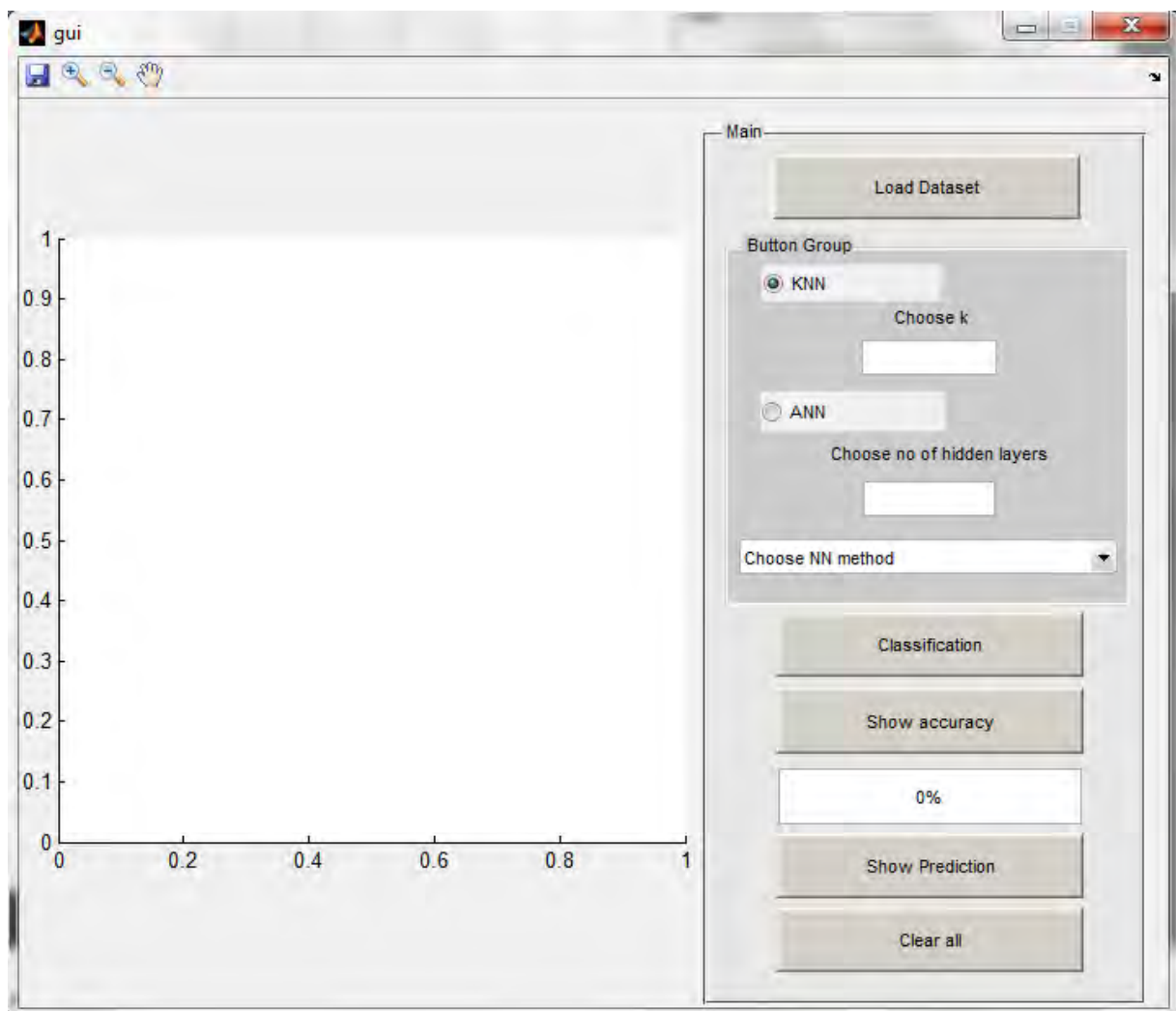
5.5 Γραφικό περιβάλλον Χρήστη(GUI)

Γραφικό περιβάλλον χρήστη (Graphical User Interface – GUI) είναι ένα σύνολο γραφικών στοιχείων, τα οποία παρέχουν την δυνατότητα στον χρήστη να επικοινωνεί με τον υπολογιστή και να φέρει εις πέρας κάποιες λειτουργίες, μέσω εργαλείων και ενδείξεων που του παρέχονται. Το Matlab προσφέρει ένα εργαλείο σχεδιασμού γραφικού περιβάλλοντος και χρήσιμα λειτουργικά στοιχεία για την ανάπτυξη ευπαρουσίαστων εφαρμογών.

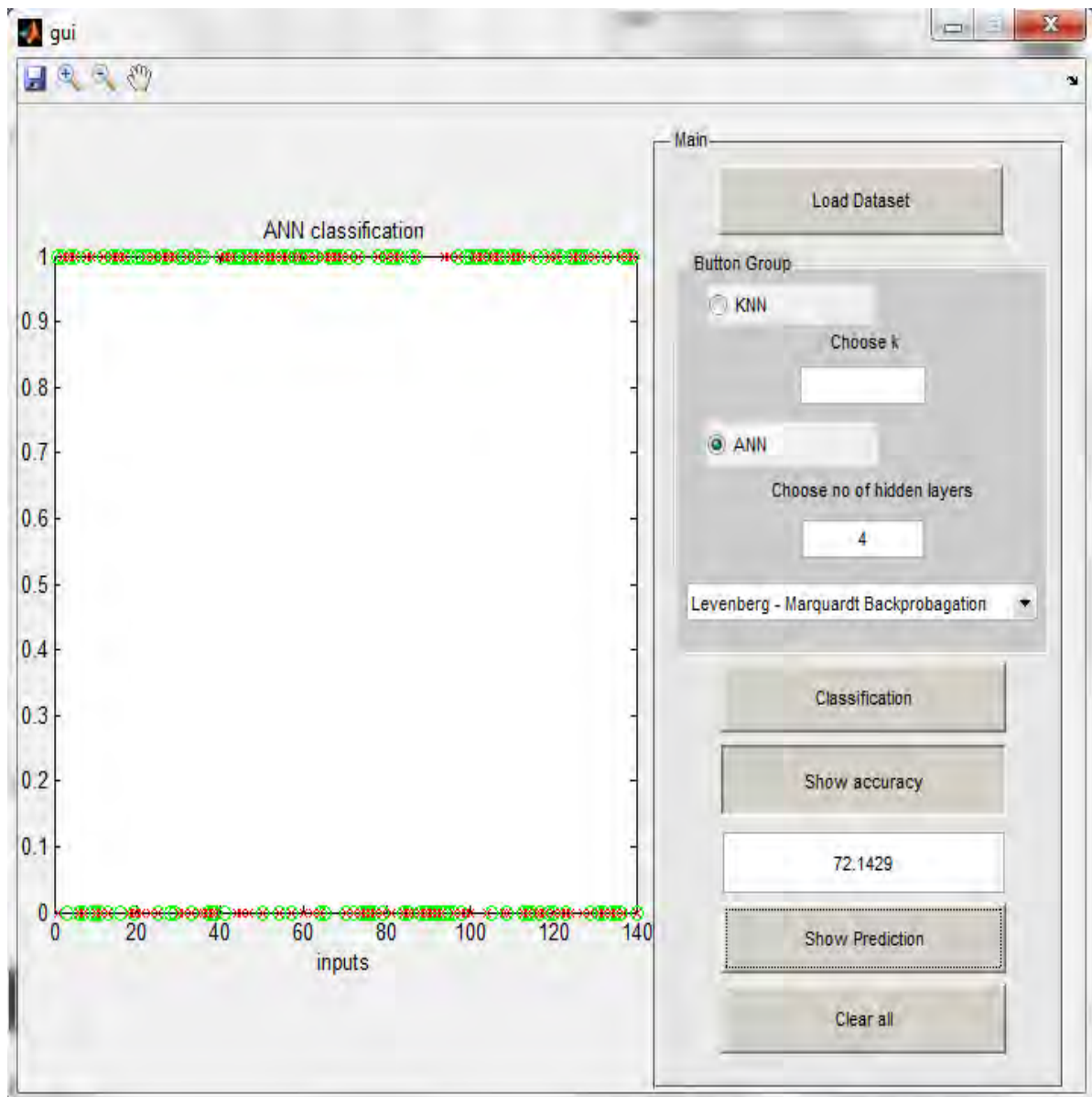
Στην παρούσα πτυχιακή εργασία, δημιουργήθηκε ένα γραφικό περιβάλλον το οποίο δίνει την δυνατότητα στον χρήστη να εισάγει τους πίνακες εκπαίδευσης (training data), κλάσης (group or targets) και δοκιμής (test data), να επιλέξει ποιον τρόπο κατηγοριοποίησης θέλει να χρησιμοποιήσει δίνοντας παράλληλα και κάποια στοιχεία για κάθε είδος κατηγοριοποίησης. Πιο συγκεκριμένα, για την κατηγοριοποίηση με τεχνητό νευρωνικό δίκτυο ο χρήστης καλείται να δώσει τον αριθμό των κρυμμένων επιπέδων που θα χρησιμοποιήσει καθώς και να διαλέξει την μέθοδο με την οποία επιθυμεί να πραγματοποιηθεί η κατηγοριοποίηση. Οι τρεις μέθοδοι που χρησιμοποιούνται στην εργασία αυτή αναφέρονται αναλυτικά παραπάνω (κεφάλαιο 5.3). Για την κατηγοριοποίηση με τον αλγόριθμο k

κοντινότερων γειτόνων ο χρήστης καλείται να πληκτρολογήσει τον αριθμό k των κοντινότερων γειτόνων που επιθυμεί να λάβει υπόψη του ο αλγόριθμος κατά την εκτέλεσή του. Στη συνέχεια δίνει την δυνατότητα εκτέλεσης της κατηγοριοποίησης, εμφανίζοντας παράλληλα και την γραφική παράσταση των δεδομένων που χρησιμοποιήθηκαν για την μέτρηση της ακρίβειας, δίνοντας την δυνατότητα να διακρίνει κανείς ποια στοιχεία της πρόβλεψης συμπίπτουν με τις κλάσεις που είναι ήδη γνωστές στον χρήστη. Επιπρόσθετα, ο χρήστης μπορεί να δει το επί τις εκατό ποσοστό ακρίβειας της κατηγοριοποίησης που εκτέλεσε, καθώς και να αποθηκεύσει στο workspace του Matlab την πρόβλεψη με τις κλάσεις στις οποίες κατηγοριοποίησε τα δεδομένα δοκιμής η μέθοδος κατηγοριοποίησης που χρησιμοποιήθηκε. Τέλος, παρέχεται ένα κουμπί που καθαρίζει το ιστορικό και επιτρέπει να γίνει κατηγοριοποίηση από την αρχή.

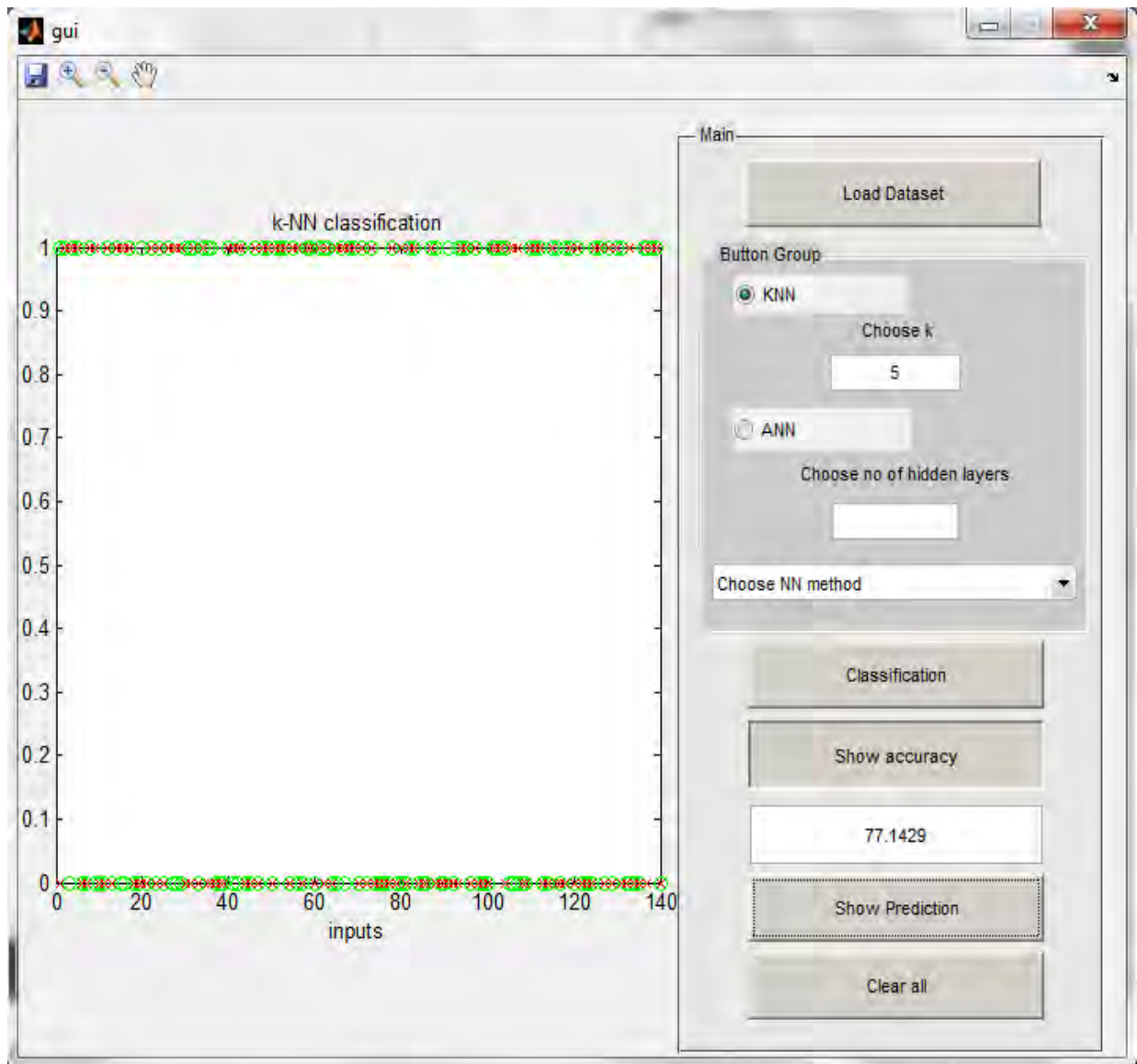
Στις εικόνες που ακολουθούν απεικονίζονται το γραφικό περιβάλλον και οι δυνατότητες που σου δίνει πριν και μετά την εφαρμογή του.



Σχήμα 21: Γραφικό περιβάλλον.



Σχήμα 22: Γραφικό περιβάλλον με εφαρμογή κατηγοριοποίησης τεχνητού νευρωνικού δικτύου.



Σχήμα 24: Γραφικό περιβάλλον με εφαρμογή κατηγοριοποίησης με k κοντινότερους γείτονες.

Κεφάλαιο 6

Συμπεράσματα

Σε αυτήν την πτυχιακή εργασία, χρησιμοποιήθηκαν μέθοδοι κατηγοριοποίησης με σκοπό την πρόβλεψη κίνησης δεξιού ή αριστερού χεριού. Χρησιμοποιήθηκαν κάποια ηλεκτροεγκεφαλογραφήματα, τα οποία μετρήθηκαν από τον διαγωνισμό BCI-experiment. Τα δεδομένα αυτά ήταν καταγραφή ηλεκτροεγκεφαλογραφημάτων κατά την κίνηση του δεξιού ή του αριστερού χεριού ύστερα από καθοδήγηση του αντικειμένου. Σκοπός της εργασίας αυτής ήταν με την χρήση τεχνητού νευρωνικού δικτύου και την χρήση του αλγορίθμου k κοντινότερων γειτόνων να κατηγοριοποιηθούν άγνωστα σήματα.

Προκειμένου να ελαττωθεί ο όγκος των δεδομένων χρειάστηκε να κάνουμε εξαγωγή χαρακτηριστικών (feature extraction). Για το σκοπό αυτό χρησιμοποιήθηκαν ο μετασχηματισμός Fourier, η μέση τιμή και η διασπορά των δεδομένων εισαγωγής και των δεδομένων πρόβλεψης, έτσι ώστε να εξαιρεθεί ο χρόνος από τα δεδομένα αυτά και να διατηρηθεί το σύνολο των κινήσεων για κάθε ηλεκτρόδιο.

Σύμφωνα με την έρευνα που πραγματοποιήθηκε, το τεχνητό νευρωνικό δίκτυο απέδωσε καλύτερα αποτελέσματα σε σύγκριση με τον αλγόριθμο k κοντινότερων γειτόνων. Η διαφορά της ακρίβειάς τους ωστόσο είναι σχετικά μικρή. Για την μέτρηση της ακρίβειας χρησιμοποιήθηκαν τα ίδια δεδομένα για εκπαίδευση ή σύγκριση (ανάλογα την μέθοδο) και για πρόβλεψη. Ο αλγόριθμος k κοντινότερων γειτόνων προέβλεψε για το ποια κίνηση πρόκειται με ακρίβεια 77,14%, ενώ το τεχνητό νευρωνικό δίκτυο με ακρίβεια περίπου 85,71%.

Ωστόσο, με την χρήση των κατηγοριοποιητών αυτών για το σύνολο του όγκου δεδομένων, χωρίς την εξαγωγή χαρακτηριστικών με τον μετασχηματισμό Fourier, την μέση τιμή και την διασπορά, ο αλγόριθμος k κοντινότερων γειτόνων απέδωσε με ποσοστό ακρίβειας σχεδόν 100%. Από την άλλη μεριά, το τεχνητό νευρωνικό δίκτυο με είσοδο το σύνολο των δεδομένων, έριξε την απόδοση της πρόβλεψης κάτω από το 50% (συγκεκριμένα 49.9%).

Για την μέτρηση της αποδοτικότητας στο νευρωνικό δίκτυο χρησιμοποιήθηκαν 10 κρυφά επίπεδα (hidden layers). Η επιλογή των κρυφών επιπέδων έγινε έπειτα από δοκιμές. Η τιμή αυτή έδινε την μεγαλύτερη απόδοση σε σύγκριση με μεγαλύτερες και μικρότερες τιμές. Απεναντίας, ο αριθμός k των κοντινότερων γειτόνων που χρησιμοποιήθηκε είναι 4. Παρατηρήθηκε πως η χρήση μικρότερου k (π.χ. $k=1$) αποδίδει καλύτερα σε σχέση με μεγάλες τιμές.

Οι τιμές αυτές αποδόθηκαν από τους κατηγοριοποιητές με εφαρμογή στα δεδομένα εισόδου, εξαγωγής χαρακτηριστικών. Τα αποτελέσματα που δίνουν οι κατηγοριοποιητές με δεδομένα εισόδου το σύνολο των ηλεκτροεγκεφαλογραφημάτων, χωρίς να έχουν υποστεί εξαγωγή χαρακτηριστικών, είναι πολύ καλύτερα για τον αλγόριθμο k κοντινότερων γειτόνων, αλλά πολύ χαμηλότερα για το τεχνητό νευρωνικό δίκτυο. Για τον αλγόριθμο k -NN είναι

αναμενόμενο να αποδίδει καλύτερα με το σύνολο των δεδομένων γιατί όσο περισσότερα δεδομένα έχει για να συγκρίνει τα δεδομένα προς κατηγοριοποίηση τόσο καλύτερα θα είναι και τα αποτελέσματα. Απεναντίας, το τεχνητό νευρωνικό δίκτυο, όχι μόνο είχε χαμηλότερη αποδοτικότητα, αλλά και ο χρόνος ο οποίος χρειάστηκε για να πραγματοποιηθεί η κατηγοριοποίηση ήταν κατά πολύ μεγαλύτερος σε σύγκριση με την εφαρμογή του για δεδομένα με εξαγωγή χαρακτηριστικών. Κι αυτό ήταν αναμενόμενο καθώς για μεγαλύτερο όγκο δεδομένων θα χρειαζόταν σίγουρα περισσότερο χρόνο για την εκπαίδευση του νευρωνικού δικτύου.

Να επισημάνω πως σημαντικό ρόλο για την μέτρηση της απόδοσης ενός κατηγοριοποιητή δεν παίζει μόνο η ακρίβεια της πρόβλεψης αλλά και ο χρόνος εκτέλεσης και κατηγοριοποίησης, η σωστή πρόβλεψη για μεγάλο όγκο δεδομένων καθώς και για ελλιπή δεδομένα εισόδου.

Σε ότι αφορά τον αλγόριθμο k κοντινότερων γειτόνων, η τιμή του k εξαρτάται από τα δεδομένα. Για μεγάλη τιμή του k τα όρια των κλάσεων είναι περισσότερο ευδιάκριτα και μειώνεται η επίδραση του θορύβου στην κατηγοριοποίηση. Για προβλήματα με δύο κατηγοριοποιητές μία καλή τιμή του k είναι ένας περιττός αριθμός έτσι ώστε να μην τύχει περίπτωση ισοψηφίας. Γενικά, η επιλογή του k παίζει πολύ σημαντικό ρόλο και στην κατηγοριοποίηση, καθώς για διαφορετικό k είναι δυνατόν να κατηγοριοποιηθούν διαφορετικά τα αντικείμενα προς κατηγοριοποίηση, αλλά και στην αποδοτικότητα, καθώς για μικρές τιμές το αποτέλεσμα επηρεάζεται από τον θόρυβο των δεδομένων ενώ για μεγάλες τιμές το αποτέλεσμα περιέχει πολλά αντικείμενα από άλλες κατηγορίες.

Από την άλλη μεριά, για το τεχνητό νευρωνικό δίκτυο μπορούμε να ρυθμίσουμε κάποιες μεταβλητές σχετικές με την εκπαίδευση του δικτύου έτσι ώστε να πάρουμε το βέλτιστο αποτέλεσμα. Οι μεταβλητές αυτές είναι ο αριθμός επαναλήψεων / περασμάτων στο σύνολο εκπαίδευσης, ο χρόνος εκτέλεσης της εκπαίδευσης, το κατώτερο όριο των σφαλμάτων στο οποίο πρέπει να φτάσει το σύστημα για να σταματήσει, η αντοχή της καθόδου (δηλαδή όταν η συνάρτηση του σφάλματος δεν αλλάζει κατά πολύ τότε η εκπαίδευση σταματάει), ο όρος ορμής (αν τα βάρη παραμείνουν ίδια σε σύγκριση με την προηγούμενη επανάληψη τότε η εκπαίδευση σταματά) και τέλος ο έλεγχος επικύρωσης (δηλαδή αν για ένα συγκεκριμένο αριθμό επαναλήψεων το σφάλμα επικύρωσης δεν μειωθεί τότε η εκπαίδευση σταματά). Τον χρόνο εκπαίδευσης τον αφήσαμε ελεύθερο, ωστόσο θέσαμε όρια στις υπόλοιπες μεταβλητές. Αρχικά, ως μέγιστο αριθμό επαναλήψεων βάλουμε 3000, το κατώτερο όριο σφαλμάτων ορίστηκε στο 0.01, έτσι ώστε να έχουμε όσο το δυνατόν λιγότερα σφάλματα, η αντοχή στην κάθοδο τέθηκε να κυμαίνεται από 1.4 έως 10^{-6} . Δηλαδή, όταν οι αλλαγές στα σφάλματα κυμαίνονται σε αυτές τις τιμές τότε η εκπαίδευση σταματά. Έπειτα, ο όρος ορμής ορίστηκε να ξεκινάει από την τιμή 1 και

να αυξάνεται κατά 1.5 όταν είναι μεγαλύτερο από το προηγούμενο βάρος και να μειώνεται κατά 0.8 στην αντίθετη περίπτωση. Τέλος, ο έλεγχος επικύρωσης ορίστηκε στην τιμή 1000. Οι τιμές αυτές δόθηκαν έπειτα από πειραματικές δοκιμές προκειμένου να επιδιωχθεί το βέλτιστο αποτέλεσμα.

Λαμβάνοντας υπόψη όλα τα παραπάνω, διαπιστώθηκε πως ο αλγόριθμος k κοντινότερων γειτόνων για τα δεδομένα με εξαγωγή χαρακτηριστικών μπορεί να επιφέρει αποτελέσματα σε πολύ λιγότερο χρόνο, συγκριτικά με το τεχνητό νευρωνικό δίκτυο, ωστόσο η απόδοσή του είναι χαμηλότερη. Για μεγάλο όγκο δεδομένων, όμως, ο αλγόριθμος k-NN εκτελείται σε μικρότερο χρόνο και η απόδοσή του αγγίζει την πραγματικότητα. Τέλος, το τεχνητό νευρωνικό δίκτυο αποδίδει καλύτερα, και από άποψη αποδοτικότητας και από άποψη χρόνου, με την εφαρμογή στα δεδομένα εξαγωγής χαρακτηριστικών.

Κεφάλαιο 7

Πηγαίος κώδικας

7.1 Κώδικας για το Τεχνητό Νευρωνικό Δίκτυο

```
ANN.m  k_NN.m  gui_pt2.m  +
1  function [class,accuracy] = ANN (x_train,y_train,x_test,hidden_layer,method)
2  %ann_fourier_mean_variance
3  %load dataset_BCIcomp1
4  % Inputs process
5  [l,m,k]=size(x_train);
6  [l1,m1,k1]=size(x_test);
7  % Fourier
8  for i=1:k
9      q=fftshift(abs(fft(x_train(:, :, i))));
10     string1=['fouriervalue.move',num2str(i),'=max(q);'];
11     evalc(string1);
12 end
13 a=struct2cell(fouriervalue);
14 a=cat(1,a{:});
15 a=a';
16 % Fourier test
17 for i=1:k1
18     q=fftshift(abs(fft(x_test(:, :, i))));
19     string1=['fouriervaluete.move',num2str(i),'=max(q);'];
20     evalc(string1);
21 end
22 a1=struct2cell(fouriervaluete);
23 a1=cat(1,a1{:});
24 a1=a1';
25 % Mean
26 for i=1:k
27     q=median(x_train(:, :, i));
28     string1=['meanvalue.move',num2str(i),'=q;'];
29     evalc(string1);
```

Ln 1 Col 1

```
ANN.m x k_NN.m gui_pt2.m +
30 - end
31 - b=struct2cell(meanvalue);
32 - b=cat(1,b{:});
33 - b=b';
34 - % Mean test
35 - for i=1:k1
36 -     q=median(x_test(:, :, i));
37 -     string1=['meanvalue.move', num2str(i), '=q;'];
38 -     evalc(string1);
39 - end
40 - b1=struct2cell(meanvaluete);
41 - b1=cat(1,b1{:});
42 - b1=b1';
43 - % Variance
44 - for i=1:k
45 -     q=var(x_train(:, :, i));
46 -     string1=['varvalue.move', num2str(i), '=q;'];
47 -     evalc(string1);
48 - end
49 - c=struct2cell(varvalue);
50 - c=cat(1,c{:});
51 - c=c';
52 - % Variance test
53 - for i=1:k1
54 -     q=var(x_test(:, :, i));
55 -     string1=['varvaluete.move', num2str(i), '=q;'];
56 -     evalc(string1);
57 - end
58 - c1=struct2cell(varvaluete);
```

Ln 1 Col 1

```
ANN.m* x k_NN.m x gui_pt2.m x +
59 - c1=cat(1,c1{:});
60 - c1=c1';
61 - inputs=[a; b; c];
62 - test=[a1; b1; c1];
63 % Tatgets and test
64 - targets=y_train';
65
66 % Create NN
67 % Create a Pattern Recognition Network
68 - hiddenLayerSize = str2num(hidden_layer);
69 - whos hiddenLayerSize
70 - net = feedforwardnet(hiddenLayerSize);
71 % Set up Division of Data for Training, Validation, Testing
72 - net.divideParam.trainRatio = 70/100;
73 - net.divideParam.valRatio = 15/100;
74 - net.divideParam.testRatio = 15/100;
75 % Use the RProp training algorithm
76 - disp('ANN method=')
77 - disp(method)
78 - net.trainFcn = method;
79 % net.trainParam.time = inf;
80 - net.trainParam.goal = 0.01;
81 - net.trainParam.epochs = 2000;
82 - net.trainParam.min_grad = 0.0000000001;
83 - net.trainParam.max_fail = 1000;
84 - net.trainParam.mu = 1;
85 - net.trainParam.mu_dec = 0.8;
86 - net.trainParam.mu_inc = 1.5;
87 % Train the Network (tr=training record)
```

ANN Ln 64 Col 18

```
ANN.m* x k_NN.m x gui_pt2.m x +
86 - net.trainParam.mu_inc = 1.5;
87 % Train the Network (tr=training record)
88 %net.trainParam.showWindow = false;
89 - [net,tr] = train(net,inputs,targets);
90 % Test the Network
91 - outputs = net(inputs);
92
93 % View the Network
94 % Use the Network for prediction
95 - outputs1 = net(test);
96 - class=round(outputs1);
97 % Calculation of accuracy
98 - outputs1 = net(inputs);
99 - class1=round(outputs1);
100 - error=sum(abs(targets-class1));
101 - accuracy=((k-error)*100)/k;
102 % Plot the initial and the new results of training data
103 - plot(class1,'go')
104 - hold on
105 - plot(targets,'rx')
106 - title('ANN classification')
107 - xlabel('inputs')
108 - ylabel('class')
109
110 % Sensitivity and specificity
111 %[c,cm,ind,per] = confusion(targets,outputs1);
112 - figure;
113 - plotconfusion(targets,outputs1)
114 - end
```

ANN Ln 108 Col 16

7.2 Κώδικας για τον αλγόριθμο k κοντινότερων γειτόνων

```
ANN.m x k_NN.m* x gui_pt2.m x +
1 function [class,accuracy] = k_NN (x_train,y_train,x_test,k)
2 %k-NN_fourier_mean_variance
3 %load dataset_BCIcomp1
4 % Inputs process
5 [l,m,n]=size(x_train);
6 [l1,m1,n1]=size(x_test);
7 % Fourier
8 for i=1:n
9     q=fftshift(abs(fft(x_train(:, :, i))));
10    string1=['fouriervalue.move',num2str(i), '=max(q);'];
11    evalc(string1);
12 end
13 a=struct2cell(fouriervalue);
14 a=cat(1,a{:});
15 % Fourier test
16 for i=1:n1
17     q=fftshift(abs(fft(x_test(:, :, i))));
18    string1=['fouriervaluete.move',num2str(i), '=max(q);'];
19    evalc(string1);
20 end
21 a1=struct2cell(fouriervaluete);
22 a1=cat(1,a1{:});
23 % Mean
24 for i=1:n
25     q=median(x_train(:, :, i));
26    string1=['meanvalue.move',num2str(i), '=q;'];
27    evalc(string1);
28 end
29 b=struct2cell(meanvalue);
```

k_NN Ln 76 Col 16

```
ANN.m  k_NN.m*  gui_pt2.m  +
30 - b=cat(1,b{:});
31 - % Mean test
32 - for i=1:n1
33 -     q=median(x_test(:,:,i));
34 -     string1=['meanvalue.move',num2str(i),'=q;'];
35 -     evalc(string1);
36 - end
37 - b1=struct2cell(meanvalue);
38 - b1=cat(1,b1{:});
39 - % Variance
40 - for i=1:n
41 -     q=var(x_train(:,:,i));
42 -     string1=['varvalue.move',num2str(i),'=q;'];
43 -     evalc(string1);
44 - end
45 - c=struct2cell(varvalue);
46 - c=cat(1,c{:});
47 - % Variance test
48 - for i=1:n1
49 -     q=var(x_test(:,:,i));
50 -     string1=['varvalue.move',num2str(i),'=q;'];
51 -     evalc(string1);
52 - end
53 - c1=struct2cell(varvalue);
54 - c1=cat(1,c1{:});
55 - inputs=[a b c];
56 - test=[a1 b1 c1];
57 - % Tatgets and test
58 - targets=v train:
```

k_NN Ln 76 Col 16

```
ANN.m x k_NN.m* x gui_pt2.m x +
56 - test=[a1 b1 c1];
57   % Tatgets and test
58 - targets=y_train;
59
60   % Predict the class of test data
61 - k = str2num(k);
62 - whos k
63 - class=knnclassify(test,inputs,targets,k,'euclidean');
64
65   % Calculation the accuracy
66 - class1=knnclassify(inputs,inputs,targets,k,'euclidean');
67 - error=sum(abs(targets-class1));
68 - accuracy=((n-error)*100)/n;
69
70   % Plot the initial and the new results of training data
71 - plot(class1,'go')
72 - hold on
73 - plot(targets,'rx')
74 - title('k-NN classification')
75 - xlabel('inputs')
76 - ylabel('class')
77
78   % Sensitivity and specificity
79 - cp = classperf (targets,class1);
80 - disp('Sensitivity = ')
81 - disp(cp.sensitivity*100)
82 - disp('Specificity = ')
83 - disp(cp.specificity*100)
84 - end
```

k_NN Ln 76 Col 16

7.3 Κώδικας για το Γραφικό Περιβάλλον

```
ANN.m x k_NN.m x gui_pt2.m x +
1 function varargout = gui_pt2(varargin)
2 % GUI_PT2 MATLAB code for gui_pt2.fig
3 % GUI_PT2, by itself, creates a new GUI_PT2 or raises the existing
4 % singleton*.
5 %
6 % H = GUI_PT2 returns the handle to a new GUI_PT2 or the handle to
7 % the existing singleton*.
8 %
9 % GUI_PT2('CALLBACK',hObject,eventData,handles,...) calls the local
10 % function named CALLBACK in GUI_PT2.M with the given input arguments.
11 %
12 % GUI_PT2('Property','Value',...) creates a new GUI_PT2 or raises the
13 % existing singleton*. Starting from the left, property value pairs are
14 % applied to the GUI before gui_pt2_OpeningFcn gets called. An
15 % unrecognized property name or invalid value makes property application
16 % stop. All inputs are passed to gui_pt2_OpeningFcn via varargin.
17 %
18 % *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
19 % instance to run (singleton)".
20 %
21 % See also: GUIDE, GUIDATA, GUIHANDLES
22
23 % Edit the above text to modify the response to help gui_pt2
24
25 % Last Modified by GUIDE v2.5 24-Jun-2014 00:01:52
26
27 % Begin initialization code - DO NOT EDIT
28 gui_Singleton = 1;
```

gui_pt2 Ln 10 Col 76

```
ANN.m x k_NN.m x gui_pt2.m x +
29 - gui_State = struct('gui_Name',      mfilename, ...
30 -                 'gui_Singleton',  gui_Singleton, ...
31 -                 'gui_OpeningFcn', @gui_pt2_OpeningFcn, ...
32 -                 'gui_OutputFcn',  @gui_pt2_OutputFcn, ...
33 -                 'gui_LayoutFcn',  [] , ...
34 -                 'gui_Callback',   []);
35 - if nargin && ischar(varargin{1})
36 -     gui_State.gui_Callback = str2func(varargin{1});
37 - end
38
39 - if narginout
40 -     [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
41 - else
42 -     gui_mainfcn(gui_State, varargin{:});
43 - end
44 - % End initialization code - DO NOT EDIT
45
46
47 - % --- Executes just before gui_pt2 is made visible.
48 - function gui_pt2_OpeningFcn(hObject, eventdata, handles, varargin)
49 - % This function has no output args, see OutputFcn.
50 - % hObject    handle to figure
51 - % eventdata  reserved - to be defined in a future version of MATLAB
52 - % handles    structure with handles and user data (see GUIDATA)
53 - % varargin   command line arguments to gui_pt2 (see VARARGIN)
54
55 - % Choose default command line output for gui_pt2
56 - handles.output = hObject;
```

gui_pt2 / clear_all_Callback Ln 300 Col 5

```
ANN.m x k_NN.m x gui_pt2.m x +
57
58 % Update handles structure
59 - guidata(hObject, handles);
60
61 % UIWAIT makes gui_pt2 wait for user response (see UIRESUME)
62 % uiwait(handles.figure1);
63
64
65 % --- Outputs from this function are returned to the command line.
66 [function varargout = gui_pt2_OutputFcn(hObject, eventdata, handles)
67 [% varargout cell array for returning output args (see VARARGOUT);
68 % hObject handle to figure
69 % eventdata reserved - to be defined in a future version of MATLAB
70 % handles structure with handles and user data (see GUIDATA)
71
72 % Get default command line output from handles structure
73 - varargout{1} = handles.output;
74
75
76 % --- Executes on button press in load_button.
77 [function load_button_Callback(hObject, eventdata, handles)
78 [% hObject handle to load_button (see GCBO)
79 % eventdata reserved - to be defined in a future version of MATLAB
80 % handles structure with handles and user data (see GUIDATA)
81 - [filename, pathname] = uigetfile( ...
82 {'*.m;*.fig;*.mat;*.slx;*.mdl',...
83 'MATLAB Files (*.m,*.fig,*.mat,*.slx,*.mdl)';
84 '*.m', 'Code files (*.m)'; ...
```

gui_pt2 / clear_all_Callback Ln 300 Col 5


```
ANN.m x k_NN.m x gui_pt2.m x +
85     '*.fig','Figures (*.fig)'; ...
86     '*.mat','MAT-files (*.mat)'; ...
87     '*.mdl;*.slx','Models (*.slx, *.mdl)'; ...
88     '*.*', 'All Files (*.*)'}, ...
89     'Pick a file');
90     %[filename, pathname] = uigetfile('*.m', 'Select a MATLAB code file');
91     data=load([pathname,filename]);
92     if isequal(filename,0)
93         disp('User selected Cancel')
94     else
95         disp(['User selected ', fullfile(pathname, filename)])
96     end
97     disp(data)
98     x='KNN';
99     whos
100     % Save variables to your figure's workspace.
101     setappdata(handles.figure1, 'x', x)
102     setappdata(handles.figure1, 'data', data)
103     guidata(hObject, handles);
104
105     % --- Executes during object creation, after setting all properties.
106     function button_group_CreateFcn(hObject, eventdata, handles)
107     % hObject    handle to button_group (see GCBO)
108     % eventdata reserved - to be defined in a future version of MATLAB
109     % handles    empty - handles not created until after all CreateFcns called
110     %uicontrol('Style','Radio', 'Parent',button_group, 'HandleVisibility','off', 'Pos
111     %uicontrol('Style','Radio', 'Parent',button_group, 'HandleVisibility','off', 'Pos
112
gui_pt2 / clear_all_Callback Ln 300 Col 5
```

```
ANN.m x k_NN.m x gui_pt2.m x +
113
114 % --- Executes when selected object is changed in button_group.
115 function button_group_SelectionChangeFcn(hObject, eventdata, handles)
116 % hObject    handle to the selected object in button_group
117 % eventdata  structure with the following fields (see UIBUTTONGROUP)
118 %   EventName: string 'SelectionChanged' (read only)
119 %   OldValue: handle of the previously selected object or empty if none was select
120 %   NewValue: handle of the currently selected object
121 % handles    structure with handles and user data (see GUIDATA)
122 x = get(eventdata.NewValue, 'Tag');
123 setappdata(handles.figure1, 'x', x)
124
125
126 % --- Executes during object creation, after setting all properties.
127 function KNN_CreateFcn(hObject, eventdata, handles)
128 % hObject    handle to KNN (see GCBO)
129 % eventdata  reserved - to be defined in a future version of MATLAB
130 % handles    empty - handles not created until after all CreateFcns called
131
132 function k_Callback(hObject, eventdata, handles)
133 % hObject    handle to k (see GCBO)
134 % eventdata  reserved - to be defined in a future version of MATLAB
135 % handles    structure with handles and user data (see GUIDATA)
136
137 % Hints: get(hObject, 'String') returns contents of k as text
138 %       str2double(get(hObject, 'String')) returns contents of k as a double
139 k = get (hObject, 'String');
140 setappdata(handles.button_group, 'k', k)

gui_pt2 / clear_all_Callback Ln 300 Col 5
```



```
ANN.m x k_NN.m x gui_pt2.m x +
141
142
143 % --- Executes during object creation, after setting all properties.
144 function k_CreateFcn(hObject, eventdata, handles)
145 % hObject    handle to k (see GCBO)
146 % eventdata  reserved - to be defined in a future version of MATLAB
147 % handles    empty - handles not created until after all CreateFcns called
148
149 % Hint: edit controls usually have a white background on Windows.
150 %         See ISPC and COMPUTER.
151 - if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgrou
152 -     set(hObject,'BackgroundColor','white');
153 - end
154
155 function hidden_layers_Callback(hObject, eventdata, handles)
156 % hObject    handle to hidden_layers (see GCBO)
157 % eventdata  reserved - to be defined in a future version of MATLAB
158 % handles    structure with handles and user data (see GUIDATA)
159
160 % Hints: get(hObject,'String') returns contents of hidden_layers as text
161 %         str2double(get(hObject,'String')) returns contents of hidden_layers as a
162 - hidden_layer = get (hObject,'String');
163 - setappdata(handles.button_group,'hidden_layer',hidden_layer)
164
165
166 % --- Executes during object creation, after setting all properties.
167 function hidden_layers_CreateFcn(hObject, eventdata, handles)
168 % hObject    handle to hidden_layers (see GCBO)
```

gui_pt2 / clear_all_Callback Ln 300 Col 5

```
ANN.m x k_NN.m x gui_pt2.m x +
169 % eventdata reserved - to be defined in a future version of MATLAB
170 % handles empty - handles not created until after all CreateFcns called
171
172 % Hint: edit controls usually have a white background on Windows.
173 % See ISPC and COMPUTER.
174 - if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgrou
175 -     set(hObject,'BackgroundColor','white');
176 - end
177
178 % --- Executes on selection change in choose_method.
179 function choose_method_Callback(hObject, eventdata, handles)
180 % hObject handle to choose_method (see GCBO)
181 % eventdata reserved - to be defined in a future version of MATLAB
182 % handles structure with handles and user data (see GUIDATA)
183
184 % Hints: contents = cellstr(get(hObject,'String')) returns choose_method contents
185 % contents{get(hObject,'Value')} returns selected item from choose_method
186
187
188 % --- Executes during object creation, after setting all properties.
189 function choose_method_CreateFcn(hObject, eventdata, handles)
190 % hObject handle to choose_method (see GCBO)
191 % eventdata reserved - to be defined in a future version of MATLAB
192 % handles empty - handles not created until after all CreateFcns called
193
194 % Hint: popupmenu controls usually have a white background on Windows.
195 % See ISPC and COMPUTER.
196 - if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgrou
```

gui_pt2 / clear_all_Callback Ln 300 Col 5

```
ANN.m x k_NN.m x gui_pt2.m x +
197 - set(hObject,'BackgroundColor','white');
198 - end
199
200
201 % --- Executes on button press in classification.
202 function classification_Callback(hObject, eventdata, handles)
203 % hObject handle to classification (see GCBO)
204 % eventdata reserved - to be defined in a future version of MATLAB
205 % handles structure with handles and user data (see GUIDATA)
206
207 % Do this to retrieve variables from your figure's workspace.
208 - data = getappdata(handles.figure1, 'data');
209 - disp(data)
210 - z = get(handles.choose_method,'Value');
211 - disp('z=')
212 - disp(z)
213 - if (z == 2)
214 -     method = 'trainlm';
215 - else if (z == 3)
216 -     method = 'trainrp';
217 -     else
218 -         method = 'traingd';
219 -     end
220 - end
221 - x=getappdata(handles.figure1,'x');
222 - disp('x=')
223 - disp(x)
224 - switch x

gui_pt2 / clear_all_Callback Ln 300 Col 5
```

```
ANN.m x k_NN.m x gui_pt2.m x +
225 -     case 'KNN'
226 -         k=getappdata(handles.button_group, 'k');
227 -         disp('k=')
228 -         disp(k)
229 -         [class,accuracy] = k_NN (data.x_train,data.y_train,data.x_test,k);
230 -         whos
231 -     case 'ANN'
232 -         hidden_layer=getappdata(handles.button_group, 'hidden_layer');
233 -         disp('hidden_layer=')
234 -         disp(hidden_layer)
235 -         [class,accuracy] = ANN (data.x_train,data.y_train,data.x_test,hidden_layer
236 -         whos
237 -     end
238 -     % Do this to save variables to your figure's workspace.
239 -     setappdata(handles.figure1, 'class', class)
240 -     setappdata(handles.figure1, 'accuracy', accuracy)
241 -     guidata(hObject, handles);
242
243
244 -     % --- Executes on button press in accuracy_button.
245 - function accuracy_button_Callback(hObject, eventdata, handles)
246 - % hObject    handle to accuracy_button (see GCBO)
247 - % eventdata  reserved - to be defined in a future version of MATLAB
248 - % handles    structure with handles and user data (see GUIDATA)
249 - accuracy = getappdata(handles.figure1 , 'accuracy');
250 - set(handles.accuracy_text, 'String', accuracy)
251
252 - function accuracy_text_Callback(hObject, eventdata, handles)
```

gui_pt2 / clear_all_Callback Ln 300 Col 5

```
ANN.m x k_NN.m x gui_pt2.m x +
253 - % hObject    handle to accuracy_text (see GCBO)
254   % eventdata reserved - to be defined in a future version of MATLAB
255   % handles    structure with handles and user data (see GUIDATA)
256
257   % Hints: get(hObject,'String') returns contents of accuracy_text as text
258   %        str2double(get(hObject,'String')) returns contents of accuracy_text as a
259
260
261   % --- Executes during object creation, after setting all properties.
262   function accuracy_text_CreateFcn(hObject, eventdata, handles)
263   % hObject    handle to accuracy_text (see GCBO)
264   % eventdata reserved - to be defined in a future version of MATLAB
265   % handles    empty - handles not created until after all CreateFcns called
266
267   % Hint: edit controls usually have a white background on Windows.
268   %        See ISPC and COMPUTER.
269   if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgrou
270       set(hObject,'BackgroundColor','white');
271   end
272
273
274   % --- Executes on button press in prediction_button.
275   function prediction_button_Callback(hObject, eventdata, handles)
276   % hObject    handle to prediction_button (see GCBO)
277   % eventdata reserved - to be defined in a future version of MATLAB
278   % handles    structure with handles and user data (see GUIDATA)
279   class = getappdata(handles.figure1 , 'class');
280   assignin('base','prediction_class',class);

gui_pt2 / clear_all_Callback Ln 300 Col 5
```



```
ANN.m x k_NN.m x gui_pt2.m* x +
267 % Hint: edit controls usually have a white background on Windows.
268 % See ISPC and COMPUTER.
269 - if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgrou
270 - set(hObject,'BackgroundColor','white');
271 - end
272
273
274 % --- Executes on button press in prediction_button.
275 [function prediction_button_Callback(hObject, eventdata, handles)
276 [% hObject handle to prediction_button (see GCBO)
277 [% eventdata reserved - to be defined in a future version of MATLAB
278 [% handles structure with handles and user data (see GUIDATA)
279 - class = getappdata(handles.figure1 , 'class');
280 - assignin('base','prediction_class',class);
281 %disp(class)
282
283
284 % --- Executes on button press in clear_all.
285 [function clear_all_Callback(hObject, eventdata, handles)
286 [% hObject handle to clear_all (see GCBO)
287 [% eventdata reserved - to be defined in a future version of MATLAB
288 [% handles structure with handles and user data (see GUIDATA)
289 - cla(handles.axes,'reset');
290 %set(handles.button_group,'Tag','0'); %resets the focus of the radio button
291 - set(handles.k,'String','');
292 - set(handles.hidden_layers,'String','');
293 - set(handles.accuracy_text,'String','0%');
294 - set(handles.choose_method,'Value',1)
```

gui_pt2 / clear_all_Callback Ln 294 Col 37

Βιβλιογραφία

- [1] Neural Networks Backpropagation General Gradient Descent.
- [2] ABDUL-BARY RAOUF SULEIMAN, T.A.-H.F., FEATURES EXTRACTION TECHNIQUES OF EEG SIGNAL FOR BCI APPLICATIONS.
- [3] Abdulhamit Subasi, E.E., Classification of EEG signals using neural network and logistic regression. Elsevier, 2004.
- [4] Ales Prochazka, M.M., Oldrich Vysata, Robert Hava and Carmen Paz Suarez Araujo, Multi-Channel EEG Signal Segmentation and Feature Extraction.
- [5] ASHISH PANAT, A.P., GAYATRI DESHMUKH, FEATURE EXTRACTION OF EEG SIGNALS IN DIFFERENT EMOTIONAL STATES.
- [6] Author(s), A. EEG-based Individual Identification using KNN Classification Approach.
- [7] C.Vigneshwari, V.V., S.Vaira Vignesh, G.Sumithra, Analysis of Finger Movements Using EEG Signal. International Journal of Emerging Technology and Advanced Engineering, 2013. 3(1).
- [8] II, B.C. 2003.
- [9] Mahfuzah Mustafa, M.N.T., Zunairah Hj. Murat and Norizam Sulaiman, Comparison between KNN and ANN Classification in Brain Balancing Application via Spectrogram Image. Journal of Computer Science & Computational Mathematics, 2012. 2(4).
- [10] Martin Riedmiller, I.f.L., Komplexitat and Deduktionssysteme, Rprop - Description and Implementation Details. 1994.
- [11] MathWorks, T., MATLAB Creating Graphical User Interfaces.
- [12] Nandish.M, S.M., Hemanth Kumar P, Faizan Ahmed, Feature Extraction and Classification of EEG Signal Using Neural Network Based Techniques. International Journal of Engineering and Innovative Technology (IJEIT), 2012. 2(4).
- [13] S.Sapna, D.A.T.a.M.P.K., BACKPROPAGATION LEARNING ALGORITHM BASED ON LEVENBERG MARQUARDT ALGORITHM.
- [14] Toolbox, N.N. Resilient Backpropagation (trainrp). Available from: <http://www-rohan.sdsu.edu/doc/matlab/toolbox/nnet/backpr57.html>.
- [15] Tummala Pradeep, P.S., P.S.Avadhani and Y.V.S. Murthy, Comparison of variable learning rate and Levenberg-Marquardt back-propagation training

algorithms for detecting attacks in Intrusion Detection Systems. International Journal on Computer Science and Engineering (IJCSE), 2011. 3.

[16] U. RAJENDRA ACHARYA, V.S.S., SUBHAGATA CHATTOPADHYAY, WENWEIYU and ANG PENG CHUAN ALVIN, APPLICATION OF RECURRENCE QUANTIFICATION ANALYSIS FOR THE AUTOMATED IDENTIFICATION OF EPILEPTIC EEG SIGNALS. World Scientific Publishing Company, 2011. 21.

[17] Uncuoğlu, Ö.K.E., Comparison of three back-propagation training algorithms for two case studies Indian Journal of Engineering & Materials Sciences 2005. 12.

[18] Βλαχάβας, Κ., Βασιλειάδης, Κόκκορας, Σακελλαρίου, Τεχνητή Νοημοσύνη, ed. Ε.Π. Μακεδονίας. 2006.

[19] Özgür Kişi & Erdal Uncuoğlu, Comparison of three back-propagation training algorithms for two case studies, Indian Journal of Engineering & Materials Sciences, 2005. 12.

