Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας Τεχνολογιας Πληροφορικης και Υπολογιστων

Διάγνωση Σχιζοφρένειας με χρήση Μηχανικής Μάθησης

Διπλωματικη Εργασια

Ελενη Παπαδοπουλου

Επιβλέπων: Ανδρέας-Γεώργιος Σταφυλοπάτης
Καθηγητής Ε.Μ.Π

Αθήνα, Ιούνιος 2022

National Technical University of Athens

School of Electrical and Computer Engineering
Division of Information Technology and Computers

Detection of Schizophrenia using Machine Learning

Diploma Thesis

Eleni Papadopoulou

Supervisor: Andreas-Georgios Stafylopatis
Professor, National Technical University of Athens

Athens, June 2022

Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας Τεχνολογιας Πληροφορικης και Υπολογιστων

Διάγνωση Σχιζοφρένειας με χρήση Μηχανικής Μάθησης

Διπλωματικη Εργασια

Ελενη Παπαδοπουλου

Επιβλέπων: Ανδρεάς-Γεώργιος Σταφυλοπάτης
Καθηγητής Ε.Μ.Π

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 8η Ιουνίου 2022

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
.....	.....	.....
Ανδρέας-Γεώργιος Σταφυλοπάτης Καθηγήτης Ε.Μ.Π.	Γεώργιος Στάμου Καθηγητής Ε.Μ.Π	Στέφανος Κόλιας Καθηγητής Ε.Μ.Π

Νευρωνικα Δικτυα και Ευφυη Υπολογιστικα Συστηματα
Αθήνα, Ιούνιος 2022

Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας Τεχνολογιας Πληροφορικης και Υπολογιστων

(Υπογραφή)

.....
Ελένη Παπαδοπούλου
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Στόχος της παρούσας διπλωματικής εργασίας είναι η διάγνωση της σχιζοφρένειας από δεδομένα που λαμβάνονται από την απεικόνιση του εγκεφάλου με τη χρήση της μεθόδου fMRI χρησιμοποιώντας βασικούς ταξινομητές και μοντέλα βαθιάς μάθησης. Τα δεδομένα που χρησιμοποιήθηκαν ελήφθησαν από τη βάση δεδομένων του Centre for Biomedical Research Excellence (COBRE), η οποία έχει συλλέξει δεδομένα απεικόνισης νευροεγκεφάλου από εργαστήρια σε όλο τον κόσμο. Στη συνέχεια, το σύνολο δεδομένων περιέχει δεδομένα από 72 ασθενείς με σχιζοφρένεια και 75 υγιή δείγματα (ηλικίες που κυμαίνονται από 18 έως 65 σε κάθε ομάδα). Οι βασικοί ταξινομητές που χρησιμοποιούνται σε αυτή τη διπλωματική εργασία είναι ο ταξινομητής kNN, ο ταξινομητής SVM και η λογιστική παλινδρόμηση. Επιπλέον, εξετάζεται μια προσέγγιση βαθιάς μάθησης χρησιμοποιώντας συνελικτικά νευρωνικά δίκτυα, LSTM και συνελικτικό δίκτυο γραφημάτων. Πειραματιστήκαμε με διάφορες παραλλαγές της αρχιτεκτονικής του CNN και με διαφορετικές παραμέτρους των ταξινομητών, και καταφέραμε να επιτύχουμε ακρίβεια πρόβλεψης έως και 72,39%.

Λέξεις Κλειδιά:

νευρωνικά δίκτυα, μηχανική μάθηση, ταξινομητές, σχιζοφρένεια, fMRI, μάθηση γνωρισμάτων, βαθιά μάθηση, μοντέλα δικτύων, πίνακας αλληλοσυσχέτισης, περιοχές ενδιαφέροντος

Abstract

The aim of this diploma thesis is to diagnose schizophrenia from data obtained from brain imaging using the fMRI method using basic classifiers and deep learning models. The data used were obtained from the Center for Biomedical Research Excellence (COBRE) database, which has collected neurobrain imaging data from laboratories around the world. Then dataset contains data from 72 Schizophrenia patients and 75 health controls (ages ranging from 18 to 65 in each group). The basic classifiers used in this diploma thesis are the kNN classifier, SVM classifier and logistic regression. In addition, a deep learning approached is examined using Convolutional Neural Networks, LSTM and Graph Convolutional Network. We experimented with different variants of the CNN's architecture and with different parameters of the classifier, and we managed to achieve prediction accuracy of up 72.39%.

Keywords:

neural networks, machine learning, classifiers, schizophrenia, fMRI, feature engineering, deep learning, graph networks, correlation matrix, regions of interest, ROIs

Ευχαριστίες

Η συγγραφή της παρούσας Διπλωματικής Εργασίας σηματοδοτεί την ολοκλήρωση των προπτυχιακών μου σπουδών. Πριν κλείσει το μεγάλο αυτό κεφάλαιο της ζωής μου, θα ήθελα να ευχαριστήσω τα άτομα που στάθηκαν δίπλα μου και συνέβαλαν στην μέχρι τώρα πορεία μου.

Αρχικά, θα ήθελα να ευχαριστήσω τον Καθηγητή μου, κ. Ανδρέα Στραφυλοπάτη και τον κ. Γεώργιο Σιόλα για την πολύτιμη βοήθειά του στην εκπόνηση τους παρόντος θέματος. Υπήρξαν καταπληκτικοί μέντορες, οι οποίοι με τις ιδέες τους και την προθυμότητά τους, με ενέπνευσαν και με καθοδήγησαν όλους αυτούς τους μήνες. Επιπλέον του είμαι βαθιά ευγνώμων για την πολύτιμη στήριξή του στα μελλοντικά μου σχέδια. Ήταν τιμή μου να εργαστώ μαζί του στο Εργαστήριο Συστημάτων Τεχνητής Νοημοσύνης και Μάθησης.

Επίσης, θα ήθελα να εκφράσω την ευγνωμοσύνη μου στους φίλους μου, με τους οποίους έχω μοιραστεί πολλές από τις πιο ξεχωριστές στιγμές της ζωής μου. Ήταν αυτοί που μου χάρισαν και θα μου χαρίζουν στιγμές ανεμελιάς, άφθονου γέλιου και αγάπης.

Είναι αλήθεια ότι δε θα αισθανόμουν τόσο ολοκληρωμένος άνθρωπος και δε θα είχα κάνει πραγματικότητα τα όνειρά μου χωρίς την αγάπη και τη φροντίδα της οικογένειάς μου. Με στηρίζει σε κάθε μου βήμα, με συμβουλεύει και μου προσφέρει ανεκτίμητα ψυχικά αγαθά. Θα είμαι πάντα κοντά τους. Θα ήθελα να αφιερώσω σε αυτούς την παρούσα εργασία.

Contents

Περίληψη **7**

Abstract **8**

Ευχαριστίες **9**

Εκτεταμένη Περίληψη **12**

1 Introduction **38**
1.1 Schizophrenia and diagnosis 38
1.2 Objectives 38
1.3 Thesis Outline 39

2 Theoretical background **40**
2.1 Machine Learning and Neural Networks 40
2.2 Machine Learning Methods 40
2.3 Feature engineering 42
2.4 Machine learning approach 43
2.4.1 Classification algorithms 43
2.4.2 Logistic regression 44
2.4.3 Support vector machine 47
2.4.4 k-Nearest Neighbor 51
2.4.5 Loss Function 52
2.5 Deep learning approach 53
2.5.1 Artificial Neural Networks structure 54
2.5.2 Activation functions 56
2.5.3 Back propagation 59
2.6 Convolutional neural networks 59
2.6.1 Convolutional layer 60
2.6.2 Pooling layer 60
2.6.3 Fully-connected layer 61
2.6.4 LSTM layer 62
2.7 Graph convolutional networks 64
2.7.1 Graph Convolutional layer 64

3 Schizophrenia and fMRI method **66**
3.1 Symptoms of Schizophrenia 66
3.2 Diagnosis of schizophrenia 67
3.3 The fMRI imaging method 67

4 Dataset **69**
4.1 COBRE database 69
4.2 Data preprocessing 70

5 Experiments **72**
5.1 Experimental procedure 72
5.1.1 Accuracy and loss metrics 72
5.1.2 Batch size 74
5.1.3 CNN model complexity 75
5.1.4 Loss function 76
5.1.5 Model comparison 77
5.2 Future work 82
5.2.1 Graph structure 82
5.2.2 Graph edges 83
5.2.3 GCN-architecture 84

Κατάλογος Σχημάτων **85**

List of Figures **87**

References **91**

Εκτεταμένη Περίληψη

Εισαγωγή

Η σχιζοφρένεια (SZ) είναι μια σοβαρή ψυχιατρική διαταραχή που επηρεάζει τα συναισθήματα, την κοινωνική συμπεριφορά και την αντίληψη της πραγματικότητας ενός ατόμου. Αν και τα βιολογικά αίτια δεν έχουν ακόμη εξακριβωθεί, γενετικοί και περιβαλλοντικοί παράγοντες, όπως το προγεννητικό στρες, οι τραυματικές εμπειρίες ή η εκτεταμένη χρήση ναρκωτικών, μπορεί να είναι κρίσιμα συστατικά για την ανάπτυξη αυτής της διαταραχής. Ζωικά μοντέλα δείχνουν ότι οι αναπτυξιακές βλάβες του ιππόκαμπου προκαλούν αποσύνδεση του προμετωπιαίου φλοιού. Η σχιζοφρένεια επηρεάζει περίπου 24 εκατομμύρια ανθρώπους ή 1 στους 300 ανθρώπους (0,32%) παγκοσμίως. Αυτό το ποσοστό είναι 1 στα 222 άτομα (0,45%) μεταξύ των ενηλίκων. Δεν είναι τόσο συχνή όσο πολλές άλλες ψυχικές διαταραχές. Είναι δυνατόν τα άτομα με σχιζοφρένεια να ζήσουν μια φυσιολογική ζωή, αλλά μόνο με καλή θεραπεία. Δεν υπάρχει σίγουρος τρόπος για την πρόληψη της σχιζοφρένειας, αλλά η τήρηση του σχεδίου θεραπείας μπορεί να βοηθήσει στην πρόληψη υποτροπών ή επιδείνωσης των συμπτωμάτων. Επιπλέον, οι ερευνητές ελπίζουν ότι το να μάθουν περισσότερα για τους παράγοντες κινδύνου για τη σχιζοφρένεια μπορεί να οδηγήσει σε έγκαιρη διάγνωση και θεραπεία. Μέχρι στιγμής, η διάγνωση της σχιζοφρένειας είναι δυνατή μέσω ψυχιατρικής παρατήρησης με βάση κριτήρια συμπεριφοράς. Αν και, το γεγονός ότι υπάρχουν ενδείξεις ότι η σχιζοφρένεια σχετίζεται στενά με τον τρόπο που συνδέονται οι περιοχές του εγκεφάλου μεταξύ τους, οδήγησε την προσοχή του επιστήμονα σε άλλες μεθόδους διάγνωσης, όπως τις μεθόδους fMRI. Οι τεχνικές μηχανικής μάθησης μπορούν να συμβάλουν σε αυτή την προσπάθεια καθώς είναι ένα χρήσιμο εργαλείο που μπορεί να αποκτήσει πληροφορίες από τα δεδομένα fMRI.

Στόχος

Ο στόχος αυτής της διπλωματικής εργασίας είναι να δημιουργήσει ένα μοντέλο για να μπορεί να προβλέψει την πιθανότητα ένας πιθανός ασθενής να πάσχει από σχιζοφρένεια με βάση δεδομένα fMRI (Λειτουργική απεικόνιση μαγνητικού συντονισμού). Είναι μια εναλλακτική προσέγγιση για τη διάγνωση αυτών των ασθενειών σε φυσιολογικό επίπεδο, χωρίς τη χρήση ψυχιατρικών ή φυσιολογικών πληροφοριών για τους πιθανούς

ασθενείς. Με τα χρόνια η τεχνητή νοημοσύνη έχει χρησιμοποιηθεί για την ανίχνευση ψυχικών ασθενειών με διαφορετικά μέσα. Οι πιο κοινές τεχνικές για την ανίχνευση της σχιζοφρένειας με χρήση τεχνητής νοημοσύνης περιλαμβάνουν σαρώσεις PET, EEG, τεχνικές που περιλαμβάνουν ταξινόμηση γονιδίων και πρωτεϊνών [1] και απεικόνιση μαγνητικού συντονισμού (MRI). Η μαγνητική τομογραφία είναι μια ιατρική τεχνική απεικόνισης που χρησιμοποιείται στην ακτινολογία για την απεικόνιση της ανατομίας και των φυσιολογικών διεργασιών του σώματος. Για τους σκοπούς της διπλωματικής εργασίας χρησιμοποιήθηκε ένα σύνολο δεδομένων ανοιχτού κώδικα που ελήφθη από το OpenNeuro [2]. Προκειμένου να δημιουργηθεί το σύνολο δεδομένων, ελήφθησαν σαρωμένες εικόνες του εγκεφάλου τόσο από ασθενείς με διάγνωση σχιζοφρένειας όσο και από υγιείς εξεταζόμενους. Το θέμα αυτής της διπλωματικής εργασίας είναι ένα ενδιαφέρον αλλά και απαιτητικό θέμα καθώς διερευνά πολλαπλά επιστημονικά πεδία: ιατρικές μελέτες για ψυχικές διαταραχές, δεδομένα απεικόνισης εγκεφάλου και ανάλυση των δεδομένων και φυσικά νευρωνικά δίκτυα και αλγόριθμους μηχανικής μάθησης. Η παρούσα διπλωματική εργασία στοχεύει να χρησιμοποιήσει τις προηγούμενες μεθόδους ταξινόμησης και να προτείνει μια νέα, χρησιμοποιώντας όχι μόνο τα δεδομένα fMRI από τον εγκέφαλο αλλά και πρόσθετες πληροφορίες όπως δημογραφικά χαρακτηριστικά.

Θεωρητικό υπόβαθρο

Η Μηχανική Μάθηση είναι η τεχνολογία ανάπτυξης αλγορίθμων υπολογιστών που μπορούν να μιμηθούν την ανθρώπινη νοημοσύνη. Αυτοί οι αλγόριθμοι έχουν κατασκευαστεί για να μπορούν να βελτιώνονται αυτόματα μέσω της εμπειρίας και με τη χρήση δεδομένων, γνωστών ως δεδομένα εκπαίδευσης. Οι αλγόριθμοι μηχανικής μάθησης μπορούν να ταξινομηθούν σε ξεχωριστές κατηγορίες ανάλογα με τη φύση των δεδομένων, τη διαδικασία εκμάθησης και τον τύπο μοντέλου[3]. Η Μηχανική Μάθηση θεωρείται ως μέρος της Τεχνητής Νοημοσύνης και χρησιμοποιείται για τη λήψη προβλέψεων ή αποφάσεων με βάση τη μαθησιακή εμπειρία που απέκτησε το μοντέλο μέσω της εκπαίδευσης. Μέχρι σήμερα η τεχνολογία ΜΛ έχει εφαρμοστεί σε διαφορετικά πεδία όπως η αναγνώριση προτύπων [4], , υπολογιστική όραση [5], μηχανική διαστημικών σκαφών [6], χρηματοδότηση [7], ψυχαγωγία [8],[9], οικολογία [10],υπολογιστική βιολογία [11], [12] και βιοϊατρικές και ιατρικές εφαρμογές [13],[14].Υπάρχουν τρεις τύποι μηχανικής μάθησης: η επιβλεπόμενη μάθηση, η μάθηση χωρίς επίβλεψη και η ενισχυτική μάθηση. Αν και σε αυτή τη διπλωματική εργασία, η επιβλεπόμενη μάθηση χρησιμοποιείται για την πολλαπλή ταξινόμηση.

Επιβλεπόμενη μάθηση

Το καθοριστικό χαρακτηριστικό της επιβλεπόμενης μάθησης είναι η διαθεσιμότητα σχολιασμένων δεδομένων. Για να είμαστε πιο ακριβείς, η επιβλεπόμενη μάθηση συνεπάγεται την εκμάθηση μιας αντιστοίχισης μεταξύ ενός συνόλου μεταβλητών εισόδου και μιας μεταβλητής εξόδου, που ονομάζεται ετικέτα και στη συνέχεια αυτή η αντιστοίχιση εφαρμόζεται για την πρόβλεψη των τιμών για τα αόρατα δεδομένα [15]. Έχοντας επισημανθεί τα δεδομένα, επομένως γνωρίζοντας τη σωστή έξοδο για κάθε είσοδο,

το μοντέλο θα εκπαιδεύεται με την πάροδο του χρόνου, μετρώντας την ακρίβεια μέσω μιας συνάρτησης απώλειας που προσαρμόζεται μέχρι να ελαχιστοποιηθεί επαρκώς το σφάλμα.

Υπάρχουν δύο τύποι τεχνικών εποπτευόμενης μάθησης στη μηχανική μάθηση: παλινδρόμηση και ταξινόμηση.

- Η **παλινδρόμηση** χρησιμοποιείται για την πρόβλεψη μιας συνεχούς μεταβλητής με βάση τη σχέση μεταξύ των μεταβλητών εισόδου και της μεταβλητής εξόδου που μάθαμε κατά τη διάρκεια της εκπαίδευσης. Για παράδειγμα, η παλινδρόμηση μπορεί να είναι χρήσιμη για την πρόβλεψη της τιμής του σπιτιού, με την τιμή του σπιτιού ως έξοδο και οι εισροές θα μπορούσαν να είναι μεταβλητές όπως η τοποθεσία, το μέγεθος του σπιτιού κ.λπ.

- Η **ταξινόμηση** χρησιμοποιείται όταν η μεταβλητή εξόδου είναι κατηγορική. Έτσι, είναι χρήσιμο για την ομαδοποίηση της εξόδου μέσα σε μια κλάση. Εάν ο αλγόριθμος προσπαθήσει να χαρακτηρίσει την είσοδο σε δύο διακριτές κλάσεις, ονομάζεται δυαδική ταξινόμηση. Η επιλογή μεταξύ περισσότερων από δύο τάξεων αναφέρεται ως ταξινόμηση πολλαπλών τάξεων, κάτι που συμβαίνει στην παρούσα διπλωματική εργασία.

Μη-επιβλεπόμενη μάθηση

Αντίθετα με την προαναφερθείσα μέθοδο, υπάρχουν περιπτώσεις στις οποίες δεν είναι δυνατό να ληφθούν δεδομένα με ετικέτα ή είναι πολύ επίπονη η δημιουργία τους. Για την επίλυση αυτού του τύπου περιπτώσεων, χρησιμοποιούνται τεχνικές μάθησης χωρίς επίβλεψη για την εύρεση κρυφών μοτίβων από το δεδομένο σύνολο δεδομένων. Αυτός ο τύπος μάθησης μπορεί να συγκριθεί με τη διαδικασία που λαμβάνει χώρα στον ανθρώπινο εγκέφαλο κατά την εκμάθηση νέων πραγμάτων. Καθώς δεν υπάρχουν αντίστοιχα δεδομένα εξόδου για τα δεδομένα εισόδου, η μάθηση χωρίς επίβλεψη δεν μπορεί να εφαρμοστεί άμεσα σε προβλήματα παλινδρόμησης ή ταξινόμησης. Ο στόχος της μάθησης χωρίς επίβλεψη είναι να βρει την υποκείμενη δομή του συνόλου δεδομένων, να ομαδοποιήσει αυτά τα δεδομένα σύμφωνα με ομοιότητες και να αναπαραστήσει αυτό το σύνολο δεδομένων σε συμπιεσμένη μορφή. Ο αλγόριθμος χωρίς επίβλεψη μπορεί να κατηγοριοποιηθεί περαιτέρω σε προβλήματα ομαδοποίησης και συσχέτισης[16]. Για τους σκοπούς της διπλωματικής εργασίας, διερευνήθηκε πληθώρα ταξινομητών αλλά και μοντέλων βαθυάς μάθησης.

- Η **ομαδοποίηση** είναι μια μέθοδος που επιχειρεί να ομαδοποιήσει τα αντικείμενα με βάση την ομοιότητα μεταξύ τους, έτσι ώστε τα αντικείμενα με τις περισσότερες ομοιότητες να παραμένουν σε μια ομάδα και να έχουν λιγότερες ή καθόλου ομοιότητες με αντικείμενα μιας άλλης ομάδας.

- Το **Association** χρησιμοποιείται για τον εντοπισμό των σχέσεων μεταξύ μεταβλητών σε μια μεγάλη βάση δεδομένων. Χρησιμοποιείται συνήθως για στρατηγικές μάρκετινγκ, όπως οι άνθρωποι που αγοράζουν Χ στοιχείο είναι πιο πιθανό να αγοράσουν το προϊόν Υ.

Ενισχυτική μάθηση

Η ενισχυτική μάθηση είναι ένα υποπεδίο της μηχανικής μάθησης που ασχολείται με το πρόβλημα της εκπαίδευσης ενός πράκτορα για τη μεγιστοποίηση ενός σήματος ανταμοιβής ενώ ενεργεί σε ένα περιβάλλον. Αυτή η μέθοδος βασίζεται στην επιβράβευση επιθυμητών συμπεριφορών ή/και στην τιμωρία των ανεπιθύμητων, επομένως ένας ενισχυτικός εκπαιδευτικός παράγοντας είναι σε θέση να μάθει μέσω δοκιμής και λάθους. Ο κύριος στόχος της ενισχυτικής μάθησης είναι να ορίσει την καλύτερη ακολουθία αποφάσεων που πρέπει να ακολουθήσει ο πράκτορας για να λύσει ένα πρόβλημα μεγιστοποιώντας παράλληλα μια μακροπρόθεσμη ανταμοιβή. Αυτός είναι ο λόγος που εφαρμόζεται κυρίως για σχεδιασμό κίνησης, δυναμική διαδρομή, βελτιστοποίηση ελεγκτή, πολιτικές εκμάθησης βάσει σεναρίων για αυτοκινητόδρομους κ.λπ. Χαρακτηριστικό παράδειγμα της επάρκειας της μεθόδου είναι η χρήση της για στάθμευση που μπορεί να επιτευχθεί με την εκμάθηση πολιτικών αυτόματης στάθμευσης.

Feature engineering

Η μηχανική χαρακτηριστικών είναι η διαδικασία χειρισμού των ακατέργαστων δεδομένων σε ουσιαστικές πληροφορίες, επομένως χαρακτηριστικά που μπορούν να χρησιμοποιηθούν τόσο στην εποπτευόμενη όσο και στην μάθηση χωρίς επίβλεψη. Αρχικά, ένα ¨χαρακτηριστικό' είναι οποιαδήποτε μετρήσιμη είσοδος που μπορεί να χρησιμοποιηθεί σε ένα μοντέλο πρόβλεψης. Το Feature Engineering ενσωματώνει διάφορες τεχνικές μηχανικής δεδομένων, όπως η επιλογή σχετικών χαρακτηριστικών, ο χειρισμός των δεδομένων που λείπουν, η κωδικοποίηση των δεδομένων και η κανονικοποίησή τους. Είναι ένα από τα πιο σημαντικά καθήκοντα και παίζει καθοριστικό ρόλο στον καθορισμό του αποτελέσματος του μοντέλου.

Υπάρχει πληθώρα πλεονεκτημάτων από τη μηχανική χαρακτηριστικών, για παράδειγμα υπάρχει μεγαλύτερη ευελιξία των χαρακτηριστικών και ως αποτέλεσμα γίνεται ευκολότερο για τους αλγόριθμους να ανιχνεύουν μοτίβα στα επεξεργασμένα δεδομένα παρά στα ανεπεξέργαστα δεδομένα. Το πιο σημαντικό πλεονέκτημα είναι ότι μια αποτελεσματική μηχανική χαρακτηριστικών συνεπάγεται υψηλότερη απόδοση του μοντέλου, επομένως καλύτερη ακρίβεια και καλύτερα αποτελέσματα.

Σε αυτή τη διπλωματική εργασία, έχουν χρησιμοποιηθεί μερικές τεχνικές μηχανικής χαρακτηριστικών καθώς το σύνολο δεδομένων είναι προεπεξεργασμένο και τα δεδομένα δεν είναι εντελώς ακατέργαστα.

Προεπεξεργασία δεδομένων

Στο σύνολο δεδομένων COBRE [2] υπάρχει ένα αρχείο που ονομάζεται αρχείο συγχύσεων. Δημιουργούνται σύγχυση κατά τη σάρωση εγκεφάλου και μπορούν να αλλάξουν την ακρίβεια αναπαράστασης της σάρωσης. Ο ευρύς σκοπός του FMRI σε κατάσταση ηρεμίας είναι να χρησιμοποιήσει την κοινή διακύμανση των σημάτων που εξαρτώνται από το επίπεδο οξυγόνωσης του αίματος FMRI (BOLD) σε διαφορετικές περιοχές του εγκεφάλου ως δείκτη της σύγχρονης νευρικής δραστηριότητας. Ωστόσο, στο FMRI σε κατάσταση ηρεμίας, η λειτουργική συνδεσιμότητα προσδιορίζεται με τη μέτρηση της χρονικής ομοιότητας της χρονικής σειράς BOLD σε voxel χρησιμοποιώντας κάποια μέτρηση, συνήθως τον συντελεστή συσχέτισης[17]. Για παράδειγμα,

στο αρχικό χαρτί Biswal [18], υπολογίστηκε ο συντελεστής συσχέτισης μεταξύ της χρονολογικής σειράς BOLD ενός voxel στον κινητικό φλοιό και κάθε άλλου voxel στον εγκέφαλο. Ο συντελεστής συσχέτισης αντικατοπτρίζει πόσο παρόμοιες είναι οι μετρήσεις δύο ή περισσότερων μεταβλητών σε ένα σύνολο δεδομένων. Τα voxel των οποίων ο συντελεστής συσχέτισης πέρασε ένα στατιστικό όριο θεωρήθηκαν λειτουργικά συνδεδεμένα, αποκαλύπτοντας έτσι κοινές αυθόρμητες διακυμάνσεις μεταξύ του αριστερού και του δεξιού κινητικού φλοιού. Δεδομένου ότι οι δύο χρονοσειρές μετρώνται ταυτόχρονα, οποιαδήποτε διεργασία που δεν σχετίζεται με τη νευρωνική δραστηριότητα που επηρεάζει τη μία ή και τις δύο χρονοσειρές θα επηρεάσει το μέτρο της λειτουργικής συνδεσιμότητας, δίνοντας έτσι ένα ψεύτικο αποτέλεσμα. Αυτές οι συγχύσεις FMRI σε κατάσταση ηρεμίας μπορούν όχι μόνο να αυξήσουν τη φαινομενική λειτουργική συνδεσιμότητα εισάγοντας ψευδείς ομοιότητες μεταξύ των χρονοσειρών αλλά και να μειώσουν τη μέτρηση συνδεσιμότητας εάν εισαχθούν διαφορικές συγχύσεις μεταξύ περιοχών.

Η σύγχυση του fMRI μπορεί να προκύψει από πολλές διεργασίες στο περιβάλλον μαγνητικής τομογραφίας. Εκτός από τις δυσλειτουργίες υλικού του σαρωτή (π.χ Το αρχείο confounds περιέχει όλες τις πληροφορίες σχετικά με τα confounds και χρησιμοποιείται στον υπολογισμό του πίνακα συσχέτισης προκειμένου να ληφθούν υπόψη οι κακές επιπτώσεις των confounds. Υπάρχουν ορισμένοι εξεταζόμενοι από τους οποίους λείπουν τα αρχεία confounds, επομένως τα χαρακτηριστικά αυτών των υποκειμένων αφαιρούνται από το διάνυσμα χαρακτηριστικών. Επιπλέον, μέσα στο αρχείο confounds, υπήρχαν τιμές NaN που αντικαταστάθηκαν με μηδενικά.

Εξαγωγές περιοχών ενδιαφέροντος

Οι προσεγγίσεις που βασίζονται στο voxel, οι οποίες χρησιμοποιούνται ευρέως στην ανίχνευση ψυχικών διαταραχών και χρησιμοποιούν voxel ως χαρακτηριστικά, έχουν ένα σημαντικό πρόβλημα, ειδικά τη διάσταση. Για να είμαστε πιο ακριβείς, στην ανάλυση voxel-wise ο αριθμός των χαρακτηριστικών είναι πολύ μεγάλος σε σύγκριση με τον αριθμό των διαθέσιμων προς τον αριθμό των διαθέσιμων δειγμάτων εκπαίδευσης [19]. Για να ξεπεραστεί αυτό το πρόβλημα, η ομαδοποίηση voxel εκτελείται χρησιμοποιώντας περιοχές ενδιαφέροντος (ROI). Οι ROI προσδιορίζονται χρησιμοποιώντας άτλαντες, όπως ο Άτλας Χάρβαρντ-Οξφόρδης, ο οποίος είναι ένας πιθανολογικός άτλαντας που καλύπτει 48 φλοιώδεις και 21 υποφλοιώδεις δομικές περιοχές.

Τα δεδομένα fMRI που λαμβάνονται από τον σαρωτή αντιπροσωπεύουν ολόκληρη την περιοχή του εγκεφάλου που σαρώθηκε, άρα ολόκληρο τον εγκέφαλο. Επομένως, πραγματοποιώντας μια ανάλυση ROI χρησιμοποιώντας τον Άτλαντα Χάρβαρντ-Οξφόρδης χρησιμοποιούμε τις σημαντικές περιοχές του εγκεφάλου που περιέχουν τις χρήσιμες πληροφορίες που θα βοηθήσουν το μοντέλο να διαφοροποιήσει τα άτομα με βάση τις ψυχικές τους διαταραχές, αφαιρώντας από το σύνολο χαρακτηριστικών εκείνα τα μέρη του εγκεφάλου που μην προσφέρετε καμία χρήσιμη πληροφορία.

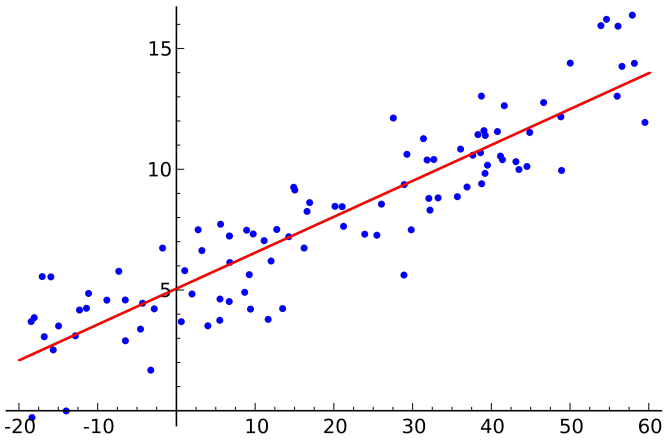
Ταξινομητές και μοντέλα

Για τους σκοπούς της διπλωματικής εργασίας, διερευνήθηκε πληθώρα αλγορίθμων. Θα συζητήσουμε διεξοδικά τους αλγόριθμους ταξινόμησης που χρησιμοποιούνται σε

αυτή τη διπλωματική εργασία. Παρακάτω, θα επεξηγηθεί μια σύντομη εισαγωγή και επεξήγηση αυτών των αλγορίθμων. Τα αποτελέσματα και η σύγκριση της απόδοσης των αλγορίθμων θα αναλυθούν παρακάτω.

Λογιστική παλινδρόμηση

Η γραμμική παλινδρόμηση χρησιμοποιείται για την εκτίμηση πραγματικών τιμών με βάση συνεχείς μεταβλητές. Για παράδειγμα, η πρόβλεψη της τιμής ενός σπιτιού, των συνολικών πωλήσεων κ.λπ. Ο στόχος είναι να βρεθεί η γραμμή που ταιριάζει καλύτερα, γνωστή ως γραμμή παλινδρόμησης που αναπαρίσταται στην εξίσωση $y = \beta_0 + \beta_1 * x + \epsilon$, όπου ϵ είναι το σφάλμα, επομένως η διαφορά μεταξύ της παρατηρούμενης τιμής ψ και της ευθείας γραμμής ($\beta_0 + \beta_1 * x$) [20]. Υπάρχουν δύο τύποι γραμμικής παλινδρόμησης: η απλή γραμμική παλινδρόμηση και η πολλαπλή γραμμική παλινδρόμηση. Η πρώτη χαρακτηρίζεται από μία ανεξάρτητη μεταβλητή, ενώ η δεύτερη από πολλαπλάσια (πάνω από 1). Στο παρακάτω Σχήμα 1 παρουσιάζεται ένα παράδειγμα απλής γραμμικής παλινδρόμησης με μία ανεξάρτητη μεταβλητή.



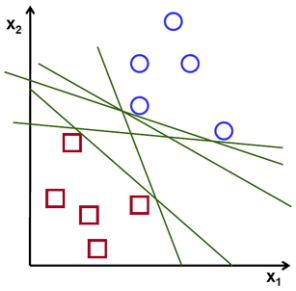
Σχήμα 1: *Απλή λογιστική παλινδρόμηση*

Η εκτίμηση μέγιστης πιθανότητας (MLE) είναι ένας αλγόριθμος που χρησιμοποιείται από τον αλγόριθμο λογιστικής παλινδρόμησης προκειμένου οι συντελεστές (τιμές βήτα) του αλγορίθμου να εκτιμηθούν από τα δεδομένα εκπαίδευσης. Ο αλγόριθμος (MLE) αναζητά τιμές συντελεστών που ελαχιστοποιούν το σφάλμα στις πιθανότητες που προβλέπονται από το μοντέλο αυτές στα δεδομένα. Αυτό υλοποιείται κυρίως χρησιμοποιώντας αποδοτικούς αλγόριθμους αριθμητικής βελτιστοποίησης. Τέτοιοι αλγόριθμοι μπορούν να επιλεγούν ως παράμετροι στον ταξινομητή λογιστικής παλινδρόμησης χρησιμοποιώντας τη βιβλιοθήκη sklearn, επομένως είναι ένας επιπλέον συντονισμός υπερ-παραμέτρου στη διαδικασία ταξινόμησης. Υπάρχουν αρκετοί βελτιστοποιητές που μπορούν να χρησιμοποιηθούν στον ταξινομητή, αλλά δεν είναι όλοι κατάλληλοι για ένα πρόβλημα πολλαπλών κλάσεων όπως το τρέχον. Ως αποτέλεσμα, αναφέρονται μόνο αυτά που χρησιμοποιούνται για τα πειράματα.

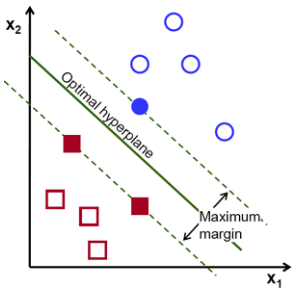
Support Vector Machine

Υποστήριξη διανυσματικού μηχανήματος (SVM) [21] είναι ένας αλγόριθμος υψηλής χρήσης που χρησιμοποιείται τόσο για προβλήματα παλινδρόμησης όσο και για προβλήματα ταξινόμησης. Ο στόχος του αλγόριθμου της μηχανής διανυσμάτων υποστήριξης είναι να βρει ένα υπερεπίπεδο σε ένα χώρο N-διάστασης, όπου N είναι ο αριθμός των χαρακτηριστικών που μπορούν να ταξινομήσουν τα σημεία δεδομένων. Χρησιμοποιεί ένα απλό μαθηματικό μοντέλο $y = w * x\prime + \gamma$ και το χειρίζεται για να επιτρέψει τη γραμμική διαίρεση τομέα [22]. Το SVM μπορεί να χωριστεί σε γραμμικά και μη γραμμικά μοντέλα [23]. Το μηχάνημα διανυσμάτων γραμμικής υποστήριξης μπορεί να διαιρέσει τα δεδομένα με μια γραμμική γραμμή ή υπερεπίπεδο για να διαχωρίσει τις κλάσεις στον αρχικό τομέα. Από την άλλη πλευρά, το μη γραμμικό μηχάνημα διανυσμάτων υποστήριξης υποδεικνύει ότι ο τομέας δεδομένων δεν μπορεί να διαιρεθεί γραμμικά και μπορεί να μετατραπεί σε ένα χώρο που ονομάζεται χώρος χαρακτηριστικών όπου τα δεδομένα μπορούν να διαιρεθούν γραμμικά.

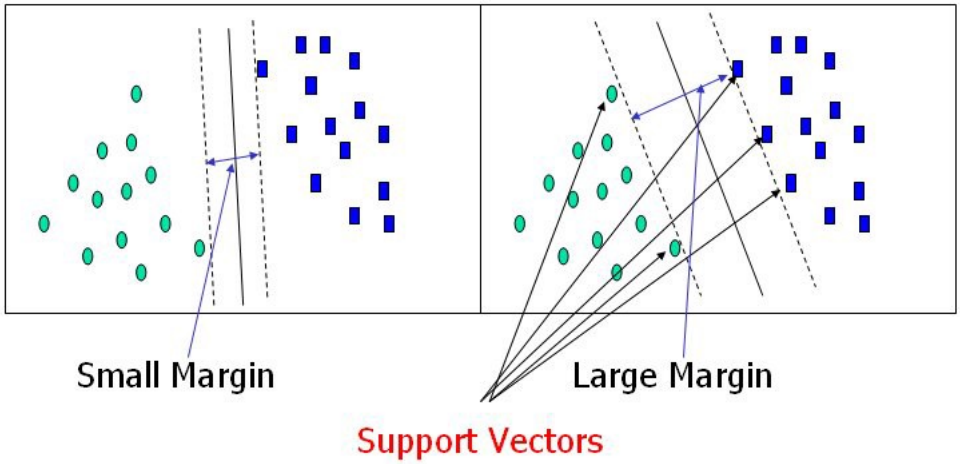
Όπως φαίνεται στο Σχήμα 2, υπάρχουν πολλά πιθανά υπερεπίπεδα που μπορούν να επιλεγούν, προκειμένου να διαχωριστούν οι δύο κατηγορίες σημείων δεδομένων. Ο σκοπός είναι να βρεθεί το επίπεδο που έχει το μέγιστο περιθώριο, άρα τη μέγιστη, απόσταση μεταξύ σημείων δεδομένων και από τις δύο κατηγορίες, όπως φαίνεται στο Σχήμα 3. Τα σημεία δεδομένων με την ελάχιστη απόσταση από το υπερεπίπεδο ονομάζονται διανύσματα υποστήριξης και επηρεάζουν τη θέση και τον προσανατολισμό του υπερεπίπεδου. Εάν διαγράψουμε τα διανύσματα στήριξης, η θέση του υπερεπίπεδου θα αλλάξει. Χρησιμοποιώντας αυτά τα διανύσματα υποστήριξης, μεγιστοποιούμε το περιθώριο του ταξινομητή, όπως απεικονίζεται στο Σχήμα 4.



Σχήμα 2: *Πιθανά υπερεπίπεδα*



Σχήμα 3: *Βέλτιστο υπερεπίπεδο*



Σχήμα 4: *Διανύσματα υποστήριξης*

k-Nearest Neighbor

Ο k-nearest neighbor είναι ένας μη παραμετρικός εποπτευόμενος αλγόριθμος που χρησιμοποιείται είτε σε προβλήματα παλινδρόμησης είτε σε προβλήματα ταξινόμησης, ο οποίος χρησιμοποιεί την εγγύτητα για να κάνει προβλέψεις σχετικά με την ταξινόμηση ενός σημείου δεδομένων. Χρησιμοποιείται περισσότερο για προβλήματα ταξινόμησης, δουλεύοντας με την υπόθεση ότι παρόμοια σημεία μπορούν να βρεθούν το ένα κοντά στο άλλο. Θα συζητήσουμε την ταξινόμηση k-NN και όχι την παλινδρόμηση καθώς εξυπηρετεί το εύρος του τρέχοντος προβλήματος. Η είσοδος αποτελείται από τα k πιο κοντινά παραδείγματα εκπαίδευσης σε ένα σύνολο δεδομένων και η έξοδος είναι μια ιδιότητα μέλους κλάσης. Η διαδικασία κατά την οποία ταξινομείται ένα σημείο ονομάζεται πλειοψηφία των γειτόνων του. Κάθε αντικείμενο εκχωρείται στην κλάση που είναι πιο κοινή μεταξύ των k πλησιέστερων γειτόνων του, όπου κ είναι ένας θετικός ακέραιος, συνήθως ένας μικρός ακέραιος. Επομένως, αν k=1, τότε το αντικείμενο απλώς εκχωρείται στην κλάση αυτού του απλού πλησιέστερου γείτονα.

Για να ρυθμιστεί ποια σημεία δεδομένων είναι πιο κοντά σε ένα δεδομένο σημείο δεδομένων, πρέπει να υπολογιστεί η απόσταση μεταξύ αυτών των σημείων δεδομένων. Υπάρχουν πολλές μετρήσεις απόστασης από τις οποίες μπορούμε να επιλέξουμε, με την Ευκλείδεια απόσταση να είναι η πιο κοινή για συνεχείς μεταβλητές και η απόσταση Hamming για τις διακριτές μεταβλητές.

Μια άλλη κρίσιμη παράμετρος για τον αλγόριθμο k-NN που πρέπει να συντονιστεί είναι η τιμή k. Η τιμή k στον αλγόριθμο k-NN καθορίζει πόσοι γείτονες θα ελεγχθούν πριν την ταξινόμηση του σημείου ερωτήματος. Η επιλογή του k μπορεί να καθορίσει εάν ο αλγόριθμος θα υπερπροσαρμόζεται ή όχι. Οι χαμηλότερες τιμές του k μπορεί να έχουν υψηλή διακύμανση, αλλά χαμηλή προκατάληψη, ενώ μεγαλύτερες τιμές του k μπορεί να έχουν υψηλή διακύμανση και χαμηλότερη διακύμανση.

Προκατάλειψη

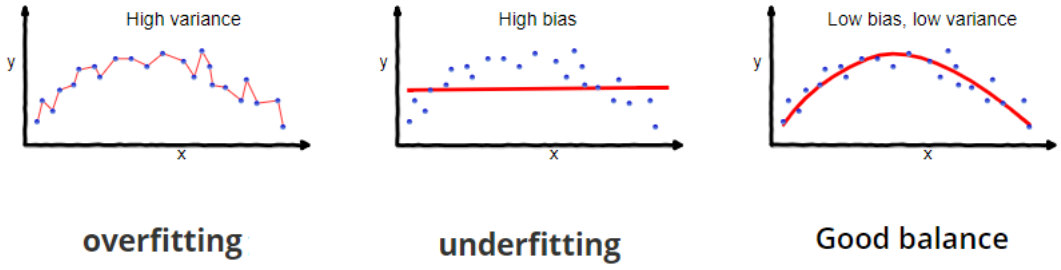
Προκατάληψη είναι η διαφορά μεταξύ της μέσης πρόβλεψης του μοντέλου και της

σωστής τιμής που το μοντέλο προσπαθεί να προβλέψει. Όταν ένα μοντέλο έχει υψηλή προκατάληψη σημαίνει ότι δίνει λίγη προσοχή στα δεδομένα εκπαίδευσης, επομένως υπεραπλουστεύει το μοντέλο. Ως αποτέλεσμα οδηγεί σε υψηλό σφάλμα στα δεδομένα εκπαίδευσης και δοκιμών.

Διακύμανση

Η διακύμανση αναφέρεται στις αλλαγές στο μοντέλο κατά τη χρήση διαφορετικών τμημάτων του σετ εκπαίδευσης. Όταν η διακύμανση είναι υψηλή, σημαίνει ότι το μοντέλο δίνει μεγάλη προσοχή στα δεδομένα εκπαίδευσης. Ως αποτέλεσμα, τέτοια μοντέλα έχουν πολύ καλή απόδοση στα δεδομένα εκπαίδευσης, αλλά έχουν υψηλά σφάλματα στα δεδομένα δοκιμής.

Ως συμπέρασμα, για να δημιουργήσουμε ένα καλό μοντέλο, πρέπει να βρούμε μια καλή αντιστάθμιση μεταξύ της πτοκατάλειψης και της διακύμανσης, ώστε να ελαχιστοποιηθεί το συνολικό σφάλμα και να αποφευχθεί η υποπροσαρμογή ή η υπερπροσαρμογή του Σχήμα 5.



Σχήμα 5: *Προκατάλειψη-Διακύμανση*

Βαθιά μάθηση

Η βαθιά εκμάθηση επεκτείνει την κλασική μηχανική μάθηση προσθέτοντας περισσότερο «βάθος», που σημαίνει μεγαλύτερη πολυπλοκότητα στο μοντέλο. Μπορεί να χαρακτηριστεί ως μια εξελιγμένη και μαθηματικά πολύπλοκη εξέλιξη της μηχανικής μάθησης. Επιπλέον, τα δεδομένα μετασχηματίζονται χρησιμοποιώντας διάφορες συναρτήσεις που επιτρέπουν την αναπαράσταση δεδομένων με ιεραρχικό τρόπο, μέσω πολλών επιπέδων αφαίρεσης [24],[25]. Ένα από τα κύρια πλεονεκτήματα της βαθιάς μάθησης είναι η ικανότητα επίλυσης πολύπλοκων προβλημάτων ιδιαίτερα καλά και γρήγορα, και ο λόγος για αυτό είναι ότι επιτρέπει μαζική παραλληλοποίηση [26]. Οι αλγόριθμοι βαθιάς μάθησης αρχίζουν να γίνονται δημοφιλείς για την ταξινόμηση των ψυχικών διαταραχών και τα προβλήματα ταξινόμησης πολλών τάξεων, όπως αυτό που συζητάμε σε αυτήν τη διπλωματική εργασία, καθώς η βαθιά μάθηση μπορεί να αυξήσει την ακρίβεια ταξινόμησης με την προϋπόθεση ότι υπάρχουν επαρκή μεγάλα σύνολα δεδομένων που περιγράφουν το πρόβλημα.

Η βαθιά εκμάθηση αποτελείται από μια πληθώρα διαφορετικών στοιχείων, για παράδειγμα συνελίξεις, επίπεδα συγκέντρωσης, πλήρως συνδεδεμένα επίπεδα, πύλες, κελιά μνήμης, συναρτήσεις ενεργοποίησης, σχήματα κωδικοποίησης/αποκωδικοποίησης κ.λπ., ανάλογα με την αρχιτεκτονική δικτύου που έχει χρησιμοποιηθεί. Ένα κρίσιμο χαρακτηριστικό της βαθιάς μάθησης είναι ότι είναι πολύ ευέλικτο και προσαρμόσιμο για μια μεγάλη ποικιλία εξαιρετικά πολύπλοκων προκλήσεων, από την άποψη της ανάλυσης δεδομένων, λόγω της εξαιρετικά ιεραρχικής δομής και της τεράστιας ικανότητας μάθησης των μοντέλων βαθιάς μάθησης [26].

Συνάρτηση απώλειας

Η συνάρτηση απώλειας είναι η συνάρτηση που υπολογίζει τη διαφορά μεταξύ της αναμενόμενης τιμής εξόδου και της τρέχουσας τιμής εξόδου του αλγορίθμου. Συγκεκριμένα, είναι μια μέθοδος για την αξιολόγηση του πόσο καλά ο αλγόριθμος μοντελοποιεί το σύνολο δεδομένων. Εάν οι προβλέψεις μας είναι πολύ αποκλίνουσες από τα πραγματικά αποτελέσματα, η έξοδος της συνάρτησης απώλειας θα είναι μεγάλος αριθμός. Εάν οι προβλέψεις είναι αρκετά καλές, η συνάρτηση απώλειας θα παράγει μικρότερο αριθμό.

Υπάρχουν διάφορες συναρτήσεις απώλειας που χρησιμοποιούνται για διαφορετικά είδη αλγορίθμων μηχανικής μάθησης. Δεν υπάρχει μία λειτουργία απώλειας που να ταιριάζει σε όλα τα προβλήματα στη μηχανική μάθηση. Γενικά, οι δύο κύριες κατηγορίες είναι οι απώλειες παλινδρόμησης και οι απώλειες ταξινόμησης. Θα αναλύσουμε περαιτέρω τις απώλειες ταξινόμησης που χρησιμοποιούνται στα πειράματα.

Hinge Loss/Multi class SVM Loss

Η συνάρτηση Hingle loss χρησιμοποιείται κυρίως για μηχανές διανυσμάτων υποστήριξης (SVM), καθώς χρησιμοποιείται για ταξινόμηση μέγιστου περιθωρίου. Συγκεκριμένα, η βαθμολογία της σωστής κατηγορίας θα πρέπει να είναι μεγαλύτερη από το άθροισμα των βαθμολογιών όλων των εσφαλμένων κατηγοριών κατά κάποιο περιθώριο ασφαλείας. Έστω τ η πραγματική έξοδος, όπου $t = \pm 1$, και ψ είναι η βαθμολογία του ταξινομητή. Η απώλεια άρθρωσης της πρόβλεψης y ορίζεται ως:

$$l(y) = max(0, 1 - t * y) \tag{1}$$

Όταν τα t και y έχουν το ίδιο πρόσημο, που σημαίνει ότι το y προβλέπει τη σωστή κλάση και το $|y| \leq 1$, η απώλεια άρθρωσης l(y) είναι 0. Αν t και y έχουν το ίδιο πρόσημο αλλά $|y| < 1$, που σημαίνει ότι η πρόβλεψη είναι σωστή αλλά όχι με αρκετό περιθώριο και αν τα t και y έχουν αντίθετα πρόσημα τότε το l(y) αυξάνεται γραμμικά με το y.

Cross Entropy Loss

Η συνάρτηση απώλειας cross entropy χρησιμοποιείται για τη μέτρηση της απόδοσης ενός προβλήματος ταξινόμησης με τιμές πιθανότητας ως έξοδο, που σημαίνει τιμές εξόδου στην περιοχή από 0 έως 1. Η απώλεια αυξάνεται καθώς η προβλεπόμενη πιθανότητα αποκλίνει από την πραγματική τιμή. Η διασταυρούμενη εντροπία βασίζεται στην ιδέα της εντροπίας από τη θεωρία πληροφοριών και υπολογίζει τον αριθμό των bit που απαιτούνται για την αναπαράσταση ή τη μετάδοση ενός μέσου γεγονότος από

μια διανομή σε σύγκριση με μια άλλη κατανομή. Μαθηματικά, είναι η προτιμώμενη συνάρτηση απώλειας στο πλαίσιο συμπερασμάτων της μέγιστης πιθανότητας. Είναι η συνάρτηση απώλειας που πρέπει να αξιολογηθεί πρώτα και να αλλάξει μόνο εάν έχετε καλό λόγο.

Η Cross Entropy θα υπολογίσει μια βαθμολογία που συνοψίζει τη μέση διαφορά μεταξύ της πραγματικής και της προβλεπόμενης κατανομής πιθανότητας για την πρόβλεψη της κλάσης 1. Η βαθμολογία ελαχιστοποιείται και μια τέλεια τιμή διασταυρούμενης εντροπίας είναι 0.

Συνάρτηση ενεργοποίησης

Οι λειτουργίες ενεργοποίησης είναι ένα κρίσιμο μέρος της αποτελεσματικότητας του νευρωνικού δικτύου. Οι προβλέψεις του μοντέλου επηρεάζονται σε μεγάλο βαθμό από την επιλογή της συνάρτησης ενεργοποίησης, καθώς αυτή η επιλογή ελέγχει πόσο καλά είναι εκπαιδευμένο το μοντέλο δικτύου δεδομένων του συνόλου δεδομένων. Όπως αναφέρθηκε προηγουμένως, μια συνάρτηση ενεργοποίησης καθορίζει την έξοδο του νευρώνα που θα μεταδοθεί στο επόμενο στρώμα. Μια συνάρτηση ενεργοποίησης μπορεί απλά να είναι δυαδική που ενεργοποιεί και απενεργοποιεί τον νευρώνα ανάλογα με την είσοδο. Μπορεί επίσης να κάνει έναν μετασχηματισμό του σήματος εισόδου σε ένα σήμα εξόδου στην περιοχή από -1 έως 1. Οι συναρτήσεις ενεργοποίησης μπορούν βασικά να χωριστούν σε δύο τύπους, τη γραμμική και τη μη γραμμική συνάρτηση ενεργοποίησης. Η πιο χρησιμοποιούμενη συνάρτηση ενεργοποίησης είναι μη γραμμικές συναρτήσεις επειδή διευκολύνει το μοντέλο να γενικεύει ή να προσαρμόζεται με ποικιλία δεδομένων και να διαφοροποιεί την έξοδο.

Σιγμοειδής συνάρτηση

Η σιγμοειδής συνάρτηση ονομάζεται επίσης λογιστική συνάρτηση και είναι μια μαθηματική συνάρτηση με το χαρακτηριστικό σχήμα μιας καμπύλης "S". Η σιγμοειδής συνάρτηση παίρνει οποιονδήποτε πραγματικό αριθμό ως είσοδο και παράγει τιμές εξόδου στο εύρος 0 έως 1. Επομένως, χρησιμοποιείται περισσότερο σε μοντέλα όπου η πιθανότητα πρέπει να προβλεφθεί ως τιμή. Η συνάρτηση Sigmoid εμφανίζεται στο Σχήμα 6 και ορίζεται για όλες τις πραγματικές τιμές εισόδου από τον τύπο:

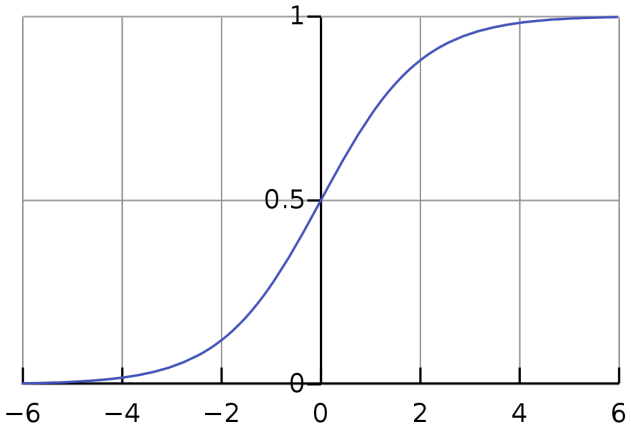
$$S = \frac{e^x}{e^x + 1} \tag{2}$$

Συνάρτηση tanh

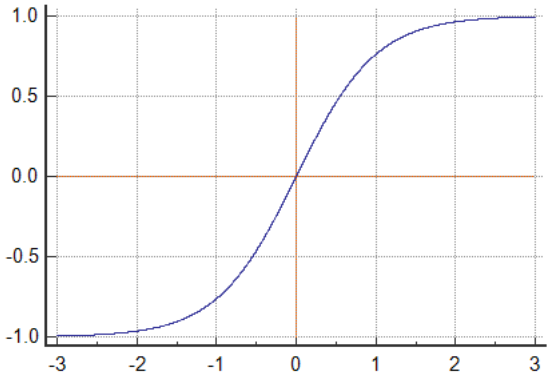
Η συνάρτηση tanh είναι πολύ παρόμοια με τη σιγμοειδή συνάρτηση, η οποία έχει επίσης σχήμα καμπύλης "S", αλλά παράγει τιμές εξόδου στην περιοχή -1 έως 1. Η συνάρτηση Tanh ορίζεται από τον τύπο:

$$tanh = \frac{e^2x - 1}{e^2x + 1} \tag{3}$$

Το πλεονέκτημα είναι ότι οι αρνητικές είσοδοι θα αντιστοιχιστούν ως έντονα αρνητικές και οι μηδενικές είσοδοι θα αντιστοιχιστούν κοντά στο μηδέν στο γράφημα tanh. Η συνάρτηση tanh εμφανίζεται στο Σχήμα 7.



Σχήμα 6: *Σιγμοειδής συνάρτηση*



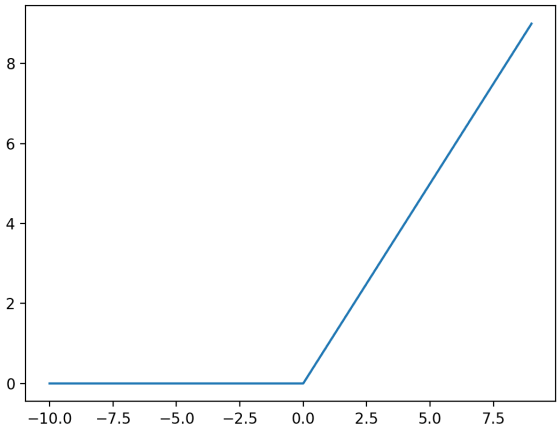
Σχήμα 7: *Συνάρτηση tanh*

Συνάρτηση ReLU

Η συνάρτηση ReLU είναι η πιο χρησιμοποιούμενη συνάρτηση ενεργοποίησης για κρυφά επίπεδα. Ο λόγος για αυτό, εκτός από την απλότητα της υλοποίησης, είναι ότι είναι αποτελεσματικό στην υπέρβαση των περιορισμών άλλων παλαιότερα δημοφιλών λειτουργιών ενεργοποίησης, όπως το Sigmoid και το Tanh. Είναι ένας απλός υπολογισμός που επιστρέφει την τιμή της εισόδου, ή 0, εάν η τιμή εισόδου είναι μικρότερη από 0. Έτσι, η συνάρτηση εξ ορισμού υπολογίζεται ως:

$$f(x) = \max(x, 0) \tag{4}$$

Λόγω της αναπαράστασής της σε γράφημα, όπως βλέπουμε στο Σχήμα 8, η συνάρτηση ReLU ονομάζεται επίσης συνάρτηση ράμπας.



Σχήμα 8: *Συνάρτηση ReLU*

Συνάρτηση Softmax

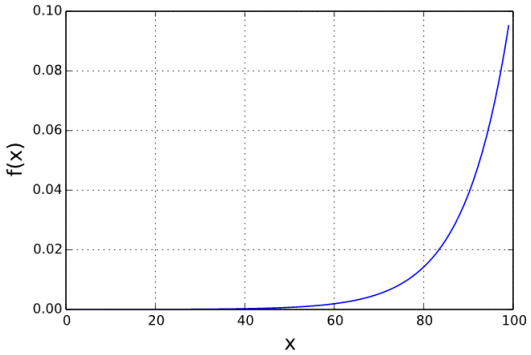
Η συνάρτηση Softmax είναι μια μαθηματική συνάρτηση που μετατρέπει ένα διάνυσμα αριθμών σε ένα διάνυσμα πιθανοτήτων. Η τυπική συνάρτηση Softmax χρησιμοποιείται συχνά στο τελικό επίπεδο ενός ταξινομητή που βασίζεται σε νευρωνικά δίκτυα και πιο συχνά σε προβλήματα ταξινόμησης πολλαπλών κλάσεων. Η έξοδος για κάθε τιμή i_{th} του διανύσματος εισόδου υπολογίζεται από αυτήν τη συνάρτηση:

$$f(x_i) = \frac{e^{x_i}}{\sum_{n=1}^{\infty} e^{x_n}} \tag{5}$$

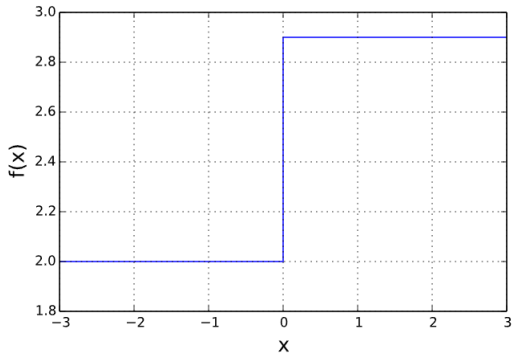
Η συνάρτηση απεικονίζεται στο Σχήμα 9.

Συνάρτηση Binary step

Η συνάρτηση δυαδικού βήματος βασικά είναι ένας ταξινομητής που βασίζεται σε κατώφλι. Εάν η τιμή εισόδου είναι πάνω από ένα όριο, η έξοδος ορίζεται σε 1, επομένως ο νευρώνας ενεργοποιείται, διαφορετικά η έξοδος ορίζεται στο 0 και ο νευρώνας απενεργοποιείται. Η συνάρτηση απεικονίζεται στο Σχήμα 10



Σχήμα 9: *Συνάρτηση Softmax*



Σχήμα 10: *Συνάρτηση Binary step*

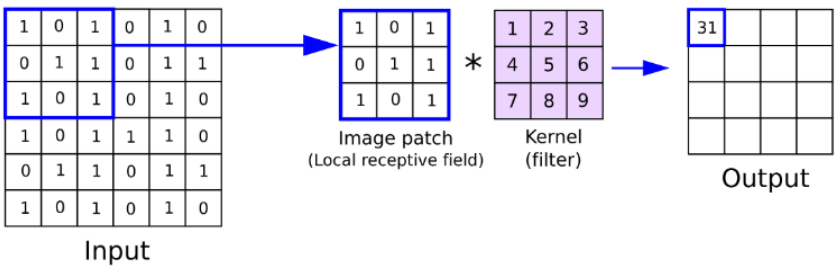
Συνελικτικά Νευρωνικά Δίκτυα

Το συνελικτικό νευρωνικό δίκτυο (CNN) είναι ένας τύπος τεχνητού νευρωνικού δικτύου που χρησιμοποιείται στην αναγνώριση και επεξεργασία εικόνας που έχει σχεδιαστεί ειδικά για την επεξεργασία δεδομένων πιξελ. Η αρχιτεκτονική ενός συνελικτικού δικτύου είναι ανάλογη με αυτή του προτύπου συνδεσιμότητας των νευρώνων στον ανθρώπινο εγκέφαλο. Το CNN έχει τους «νευρώνες» του διατεταγμένους περισσότερο σαν αυτούς του μετωπιαίου λοβού, της περιοχής που είναι υπεύθυνη για την επεξεργασία οπτικών ερεθισμάτων σε ανθρώπους και άλλα ζώα. Ένα συνελικτικό δίκτυο είναι σε θέση να καταγράψει με επιτυχία τις χωρικές και χρονικές εξαρτήσεις σε μια εικόνα εισόδου, εφαρμόζοντας σχετικά φίλτρα. Ένα τυπικό CNN αποτελείται από ένα επίπεδο εισόδου, ένα επίπεδο εξόδου και ένα κρυφό επίπεδο που περιλαμβάνει πολλαπλά συνελικτικά επίπεδα, επίπεδα συγκέντρωσης, πλήρως συνδεδεμένα επίπεδα και επίπεδα κανονικοποίησης.

Συνελικτικό επίπεδο

Στην επίδειξη στο Σχήμα 11 μπορούμε να δούμε μια εικόνα εισόδου που είναι μπερδεμένη σε patches. Κάθε ενημερωμένη έκδοση κώδικα μεταφέρει τη λειτουργία συνέλιξης με τη μήτρα φίλτρου και η μήτρα εξόδου θα είναι η συνέλιξη αυτών των δύο πινάκων. Το φίλτρο (πυρήνας) μετακινείται προς τα δεξιά με μια συγκεκριμένη τιμή διασκελισμού μέχρι να αναλύσει ολόκληρο το πλάτος. Καθώς φτάνει στο τέλος του πλάτους, πηδά κάτω στην αρχή (αριστερά) της εικόνας με την ίδια τιμή διασκελισμού και επαναλαμβάνει τη διαδικασία μέχρι να διασχιστεί ολόκληρη η εικόνα. Για παράδειγμα, στο παράδειγμά μας η τιμή Στριδε είναι 1, ο πυρήνας θα μετατοπίζεται 16 φορές και κάθε φορά θα εκτελεί μια λειτουργία πολλαπλασιασμού μήτρας μεταξύ του K (πυρήνας) και του τμήματος Π της εικόνας πάνω από το οποίο αιωρείται ο πυρήνας.

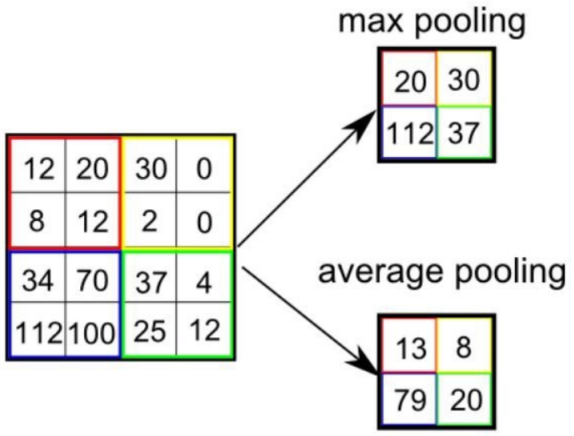
Στα πλαίσια της διπλωματικής, πειραματιστήκαμε σε διαφορετικά βάθη συνελικτικών νευρωνικών δικτύων (έως και 4 συνελικτικά επίπεδα). Ως είσοδος, τα CNN που δημιουργήθηκαν έλαβαν τους πίνακες συσχέτισης που υπολογίζονται από το σύνολο δεδομένων εισόδου. Ο πίνακας συσχέτισης είναι ένας N*N πίνακας που απεικονίζει τις συσχετίσεις όλων των πιθανών ζευγών τιμών σε έναν πίνακα. Στην περίπτωσή μας, αυτές οι τιμές αντιπροσωπεύουν τις περιοχές ενδιαφέροντος (ROI) του ανθρώπινου εγκεφάλου και τη συσχέτιση μεταξύ τους.



Σχήμα 11: *Συνέλιξη του πίνακα εισόδου με χρήση φίλτρου*

Pooling επίπεδο

Παρόμοια με το Convolutional Layer, το Pooling layer είναι υπεύθυνο για τη μείωση του χωρικού μεγέθους του συνελικτικού χαρακτηριστικού. Αυτό γίνεται για να μειωθεί η υπολογιστική ισχύς που απαιτείται για την επεξεργασία των δεδομένων μέσω μείωσης διαστάσεων. Επιπλέον, είναι χρήσιμο για την εξαγωγή κυρίαρχων χαρακτηριστικών που είναι αμετάβλητα περιστροφικά και θέσης, διατηρώντας έτσι τη διαδικασία αποτελεσματικής εκπαίδευσης του μοντέλου. Υπάρχουν δύο τύποι συγκέντρωσης: Μέγιστη συγκέντρωση και Μέση συγκέντρωση. Το Max Pooling επιστρέφει τη μέγιστη τιμή από το τμήμα της εικόνας που καλύπτεται από τον πυρήνα. Από την άλλη πλευρά, το Average Pooling επιστρέφει τον μέσο όρο όλων των τιμών από το τμήμα της εικόνας που καλύπτεται από τον πυρήνα Σχήμα 12. Στο πλαίσιο της διπλωματικής εργασίας χρησιμοποιήσαμε το Max Pooling επειδή αποδίδει πολύ καλύτερα από το Αεραγε Ποολινγ και επιπλέον, λειτουργεί και ως κατασταλτικό θορύβου, άρα εκτελεί αποθορυβοποίηση μαζί με μείωση διαστάσεων.



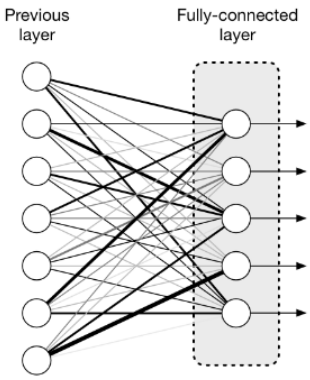
Σχήμα 12: *Παράδειγμα Max και Average Pooling*

Πλήρες συνελικτικό επίπεδο

Η προσθήκη ενός πλήρως συνδεδεμένου επιπέδου Σχήμα 13 στο νευρωνικό δίκτυο είναι ένας εύκολος τρόπος για το μοντέλο να μάθει μη γραμμικούς συνδυασμούς των περίπλοκων χαρακτηριστικών. Πριν τροφοδοτήσετε τα δεδομένα στο πλήρως συνδεδεμένο στρώμα, πρέπει να ισοπεδωθούν σε διάνυσμα στήλης. Η ισοπεδωμένη έξοδος τροφοδοτείται στη συνέχεια σε ένα νευρωνικό δίκτυο τροφοδοσίας προς τα εμπρός και εφαρμόζεται αντίστροφη διάδοση σε κάθε επανάληψη της εκπαίδευσης. Μετά από μια σειρά εποχών, το μοντέλο μπορεί να διακρίνει μεταξύ κυρίαρχων και χαμηλού επιπέδου χαρακτηριστικά σε εικόνες προκειμένου να τα ταξινομήσει χρησιμοποιώντας την τεχνική ταξινόμησης Σοφτμαξ.

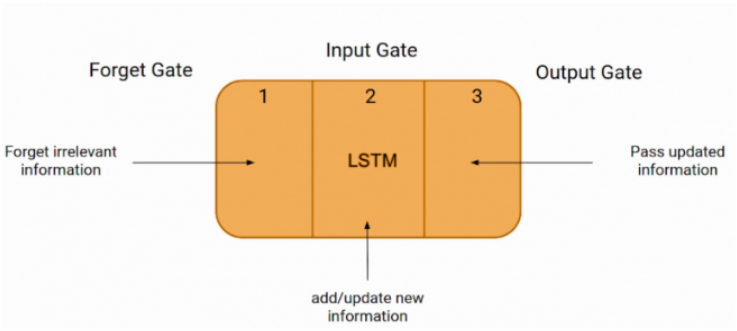
Αρχιτεκτονική LSTM

Το LSTM αποτελείται από τρία μέρη, όπως φαίνεται στην παρακάτω εικόνα και κάθε εξάρτημα εκτελεί μια μεμονωμένη λειτουργία. Αυτά τα τρία μέρη μιας κυψέλης



Σχήμα 13: *Πλήρες επίπεδο*

LSTM είναι γνωστά ως πύλες. Το πρώτο μέρος ονομάζεται πύλη Forget, το δεύτερο μέρος είναι γνωστό ως πύλη εισόδου και το τελευταίο είναι η πύλη εξόδου Σχήμα 14. Ακριβώς όπως ένα απλό RNN, ένα LSTM έχει επίσης μια κρυφή κατάσταση όπου το H(t-1) αντιπροσωπεύει την κρυφή κατάσταση της προηγούμενης χρονικής σήμανσης και το H(t) είναι η κρυφή κατάσταση της τρέχουσας χρονικής σφραγίδας. Εκτός από αυτό το LSTM έχει επίσης μια κατάσταση κελιού που αντιπροσωπεύεται από C(t-1) και C(t) για την προηγούμενη και την τρέχουσα χρονική σήμανση αντίστοιχα.



Σχήμα 14: *Αρχιτεκτονική*

Συνελικτικά δίκτυα γράφων

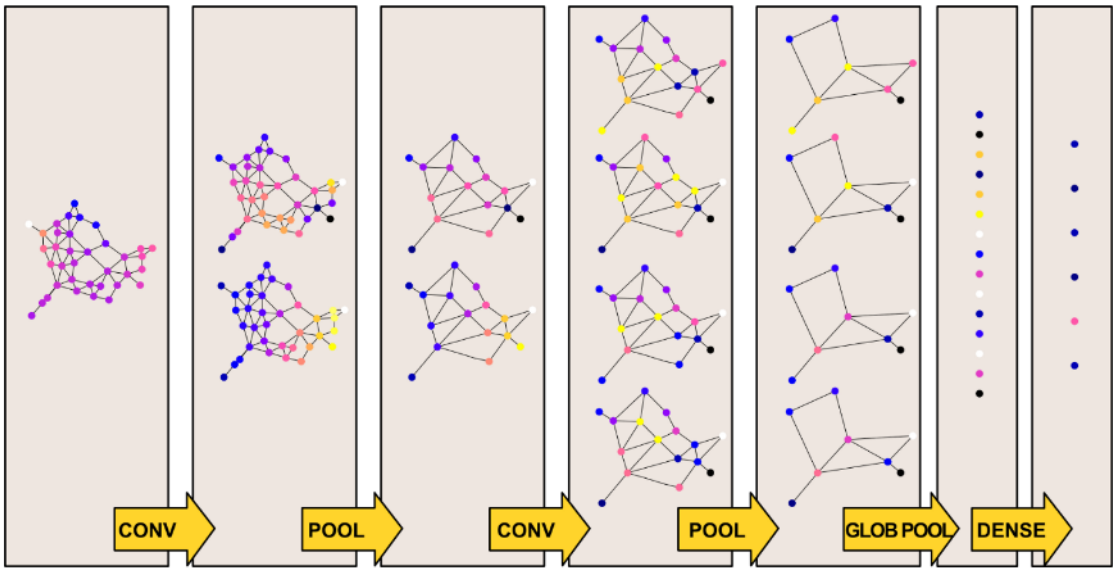
Σε αντίθεση με τα CNN, που χρησιμοποιούν δεδομένα στον ευκλείδειο χώρο, τα Graph Convolutional Networks μπορούν επίσης να λειτουργήσουν με δεδομένα που μπορούν να δομηθούν μόνο σε έναν μη ευκλείδειο χώρο και μπορούν να αναπαρασταθούν μόνο με γραφήματα. Μερικά παραδείγματα μη ευκλείδειων δομών είναι γενετικά δεδομένα, δεδομένα κοινωνικών δικτύων ή δεδομένα βιολογικών δικτύων. Η κύρια δυσκολία για τη μηχανική μάθηση σε γραφήματα είναι να βρει έναν τρόπο ενσωμάτωσης στο μοντέλο, τις πληροφορίες του γραφήματος.

Ένα γράφημα G αναπαρίσταται ως G = (V,E), όπου V είναι το σύνολο των κορυφών και E το σύνολο των ακμών του γραφήματος. $|V| = n$ και $|E| = m$ είναι ο αριθμός

των κορυφών και ο αριθμός των ακμών αντίστοιχα. Κάθε $v_i \in V$ αντιπροσωπεύει μια κορυφή του γραφήματος και κάθε $e_ij = (v_i, v_j) \in E$ αντιπροσωπεύει μια άκρη από την κορυφή v_i στην κορυφή v_j. Ο πίνακας γειτνίασης είναι ένας NxN πίνακας A με $A_ij = 1$ εάν $e_ij \in E$ και $A_ij = 0$ εάν $e_ij \notin E$. Το γράφημα μπορεί επίσης να περιέχει χαρακτηριστικά κορυφής Ξ, όπου το $X \in \mathbb{R}^{n*d}$ είναι ο πίνακας χαρακτηριστικών των κορυφών. Στην περίπτωσή μας αυτός ο πίνακας χαρακτηριστικών είναι ο πίνακας συσχέτισης.

Συνελικτικό επίπεδο γράφου

Σε ευκλείδειους τομείς, η συνέλιξη ορίζεται λαμβάνοντας το γινόμενο μεταφρασμένων συναρτήσεων. Αλλά όπως αναφέραμε, τα GCN χρησιμοποιούν μη ευκλείδειες δομές δεδομένων. Η συνέλιξη σε γραφήματα ορίζεται μέσω του μετασχηματισμού γραφήματος Fourier. Ο μετασχηματισμός Fourier γραφήματος, με τη σειρά του, ορίζεται ως η προβολή στις ιδιοτιμές του Λαπλασιανού. Αυτοί είναι οι «τρόποι δόνησης» του γραφήματος. Όσο για τα παραδοσιακά CNN, ένα GCN αποτελείται από πολλά συνελικτικά και ομαδικά στρώματα για εξαγωγή χαρακτηριστικών, ακολουθούμενα από τα τελικά πλήρως συνδεδεμένα στρώματα.

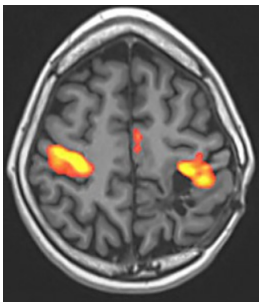


Σχήμα 15: *Συνελικτικό επίπεδο γράφου*

Μέθοδος fMRI και σύνολο δεδομένων

Η λειτουργική μαγνητική τομογραφία (fMRI) ανιχνεύει αλλαγές στη ροή του αίματος στον εγκέφαλο. Δεν είναι δομικό και η διαφορά από τις τεχνικές μαγνητικής τομογραφίας είναι ότι έχει και χρονική παράμετρο. Όταν μια περιοχή του εγκεφάλου ενεργοποιείται, καταναλώνει περισσότερο οξυγόνο, επομένως η ροή του αίματος σε αυτήν την περιοχή αυξάνεται. Το fmri απεικονίζει αυτές τις αλλαγές αλλάζοντας το χρώμα των voxel στην περιοχή που ενεργοποιείται, όπως μπορούμε να δούμε στο Σχήμα 16

Σύνολο δεδομένων COBRE



Σχήμα 16: *Περιοχές ενεργοποίησης στο fMRI*

Το Center for Biomedical Research Excellence (COBRE) συνεισφέρει ακατέργαστα λειτουργικά δεδομένα MR από 72 ασθενείς με Σχιζοφρένεια και 75 υγιείς μάρτυρες (ηλικίες από 18 έως 65 σε κάθε ομάδα). Όλα τα υποκείμενα εξετάστηκαν και αποκλείστηκαν εάν είχαν. ιστορικό νευρολογικής διαταραχής, ιστορικό νοητικής καθυστέρησης, ιστορικό σοβαρού τραύματος στο κεφάλι με απώλεια συνείδησης άνω των 5 λεπτών, ιστορικό κατάχρησης ουσιών ή εξάρτησης τους τελευταίους 12 μήνες. Οι διαγνωστικές πληροφορίες συλλέχθηκαν χρησιμοποιώντας τη δομημένη κλινική συνέντευξη που χρησιμοποιήθηκε για τις διαταραχές DSM (SCID).

Χρησιμοποιήσαμε μια προεπεξεργασμένη έκδοση του συνόλου δεδομένων [27]. Κάθε σύνολο δεδομένων fMRI διορθώθηκε για τη διαφορά μεταξύ των τομών στο χρόνο απόκτησης και οι παράμετροι μιας κίνησης άκαμπτου σώματος εκτιμήθηκαν για κάθε χρονικό πλαίσιο. Η κίνηση του άκαμπτου σώματος υπολογίστηκε τόσο εντός όσο και μεταξύ των διαδρομών, χρησιμοποιώντας ως στόχο τον διάμεσο όγκο της πρώτης διαδρομής. Αν και τα δεδομένα είναι προεπεξεργασμένα, απαιτούνται ακόμη πολλά βήματα για να είναι τα δεδομένα σε κατάλληλη μορφή για εισαγωγή στα μοντέλα μας. Πρώτα απ ́όλα, όπως αναφέρθηκε προηγουμένως, η κίνηση του άκαμπτου σώματος εκτιμήθηκε κατά τη σάρωση του ατόμου. Οι κινήσεις του θέματος μπορεί να επηρεάσουν την ευκρίνεια του σήματος λόγω της δημιουργίας θορύβου. Ο θόρυβος είναι τυχαία, ανεπιθύμητη παραλλαγή ή διακύμανση που παρεμβαίνει στο σήμα, καθιστώντας δύσκολη την εξαγωγή σημαντικών πληροφοριών εκτός εάν αφαιρεθούν. Εκτός από τις κινήσεις του θέματος, υπάρχουν και άλλοι παράγοντες που δημιουργούν θόρυβο στο σήμα, όπως σφάλματα που σχετίζονται με το πείραμα, για παράδειγμα ένα σφάλμα υλικού στη σάρωση. Όλοι αυτοί οι ανεπιθύμητοι παράγοντες που συμβάλλουν στη δημιουργία θορύβου, εκτιμώνται και κατά την προεπεξεργασία των δεδομένων και αποθηκεύονται σε εξωτερικά αρχεία για μεταγενέστερη αφαίρεση. Ονομάζονται σύγχυση και είναι αρχεία .tsv μαζί με τα αρχεία fMRI στο σύνολο δεδομένων.

Κατά συνέπεια, τα αρχεία NifTi που περιέχουν τις σαρώσεις φMPI για κάθε θέμα πρέπει να μετατραπούν σε πίνακες συσχέτισης που θα τροφοδοτηθούν στα μοντέλα. Ένας πίνακας συσχέτισης είναι ένας πίνακας που εμφανίζει τους συντελεστές συσχέτισης για διαφορετικές μεταβλητές. Ο πίνακας απεικονίζει τη συσχέτιση μεταξύ όλων των πιθανών ζευγών τιμών σε έναν πίνακα. Είναι ένα ισχυρό εργαλείο για τη σύνοψη ενός μεγάλου συνόλου δεδομένων και τον εντοπισμό και την οπτικοποίηση μοτίβων στα δεδομένα. Στη συνέχεια, παρουσιάζονται τα βήματα προεπεξεργασίας για τη μετατροπή

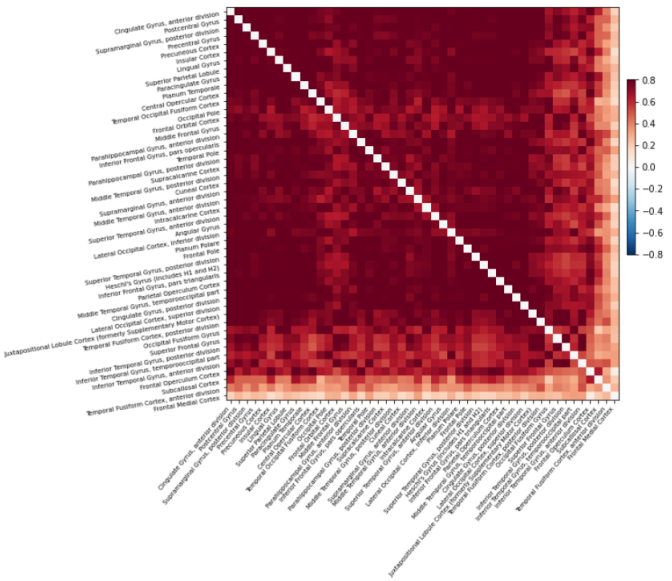
των αρχείων NifTi σε πίνακες συσχέτισης.

1. **Mstl πιθανολογικός άτλαντας:** Περιέχει ένα προκαθορισμένο σύνολο συντεταγμένων, έτσι ώστε να απομονώνονται συγκεκριμένες περιοχές του εγκεφάλου που ονομάζονται Περιοχές Ενδιαφέροντος (ROI). Οι περιοχές ενδιαφέροντος είναι μια ομάδα γειτονικών οξελς, τα οποία ενεργοποιούνται με παρόμοιο τρόπο (έχουν υψηλή συσχέτιση), επομένως μπορούν να εξεταστούν ως ολόκληρη περιοχή.

2. **Masker:** Το Μασκερ χρησιμοποιείται για την εφαρμογή του άτλαντα που ανακτήθηκε στο βήμα 1 στα δεδομένα εικόνας.

3. **Εξαγωγή χρονοσειρών:** Αυτό είναι ένα σημαντικό βήμα. Σε αυτό το βήμα, τα αρχεία confound χρησιμοποιούνται για την εξαγωγή συγχύσεων από το σήμα. Δημιουργείται ένα λειτουργικό σύνδεσμο, το οποίο είναι ένα σύνολο συνδέσεων που αντιπροσωπεύουν τις συσχετίσεις μεταξύ των ROI.

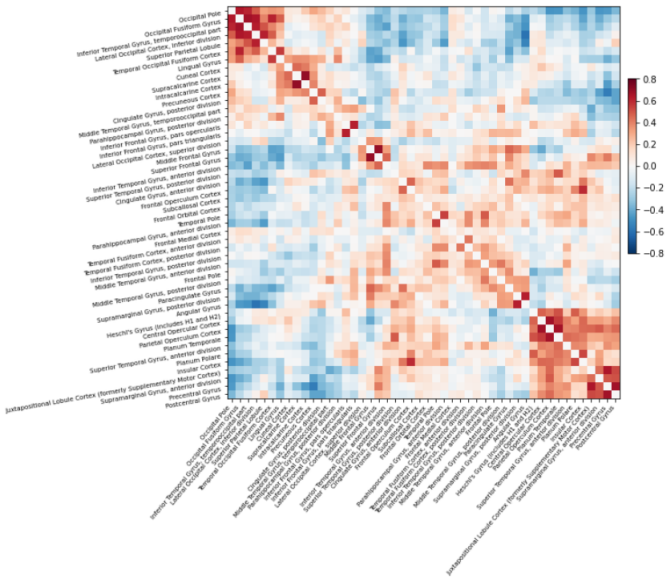
4. **Πίνακες αλληλοσυσχέτισης:** Οι πίνακες συσχέτισης υπολογίζονται χρησιμοποιώντας τις χρονοσειρές από το βήμα 3. Για να τονιστεί η σημασία της εξαγωγής συγχύσεων, συγκρίνονται δύο εικόνες. Ένας που απεικονίζει τον πίνακα συσχέτισης που προέρχεται από την ακατέργαστη χρονοσειρά Σχήμα 17 και ένας που απεικονίζει τον πίνακα συσχέτισης χωρίς τον θόρυβο που προκαλείται από τις συγχύσεις καθώς αφαιρούνται από το σήμα στο βήμα 3Σχήμα 18.



Σχήμα 17: *Πίνακας συσχέτισης από ακατέργαστες χρονοσειρές*

Πειράματα

Πρώτα απ΄όλα, τα πειράματα βασίζονται σε ποικιλία ταξινομητών που χρησιμοποιούνται για την ταξινόμηση και τη βαθιά μάθηση επίσης. Ως εκ τούτου, θα συζητήσουμε

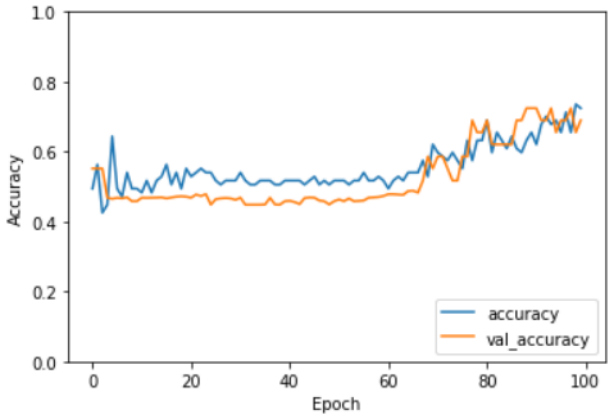


Σχήμα 18: *Πίνακας συσχέτισης μετά την αφαίρεση των συγχύσεων από τις χρονοσειρές*

τη διαφορά μεταξύ του διαφορετικού μοντέλου που χρησιμοποιείται για την ταξινόμηση της σχιζοφρένειας, την ακρίβεια που μπορεί να προσφέρει το καθένα από αυτά και πόσο κατάλληλο είναι κάθε μοντέλο για τα δεδομένα που χρησιμοποιούνται στη διατριβή. Κάθε πείραμα διεξάγεται τρεις φορές και η τελική ακρίβεια που παρουσιάζεται είναι η μέση τιμή της ακρίβειας των τριών πειραμάτων.

Μετρήσεις ακρίβειας και απώλειας

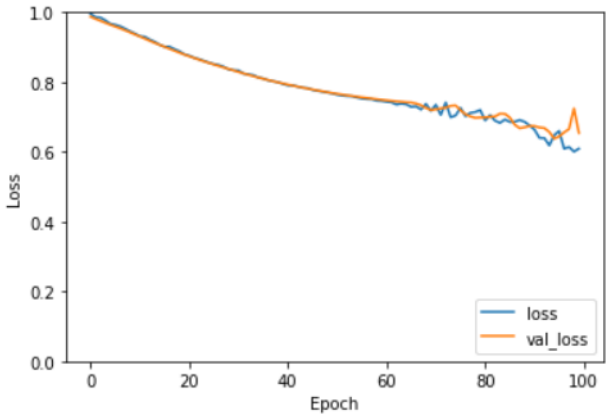
Ακολουθεί μια επίδειξη της συμπεριφοράς ακρίβειας κατά τη διάρκεια των εποχών. Για το σκοπό αυτό εξήχθησαν αποτελέσματα από το μοντέλο του ΝΝ. Η συμπεριφορά της ακρίβειας και της απώλειας είναι παρόμοια για όλα τα μοντέλα που δημιουργήθηκαν στο αντικείμενο της διπλωματικής εργασίας.



Σχήμα 19: *Η ακρίβεια της εκπαίδευσης και της επικύρωσης αλλάζει κατά τη διάρκεια των εποχών (100)*

Στο Σχήμα 20 παρατηρείται κατά τις εποχές ότι η ακρίβεια εκπαίδευσης αυξάνεται καθώς και η ακρίβεια επικύρωσης. Το σετ εκπαίδευσης χρησιμοποιείται για την εκπαίδευση του μοντέλου ενώ το σύνολο επικύρωσης χρησιμοποιείται μόνο για την αξιολόγηση της απόδοσης του μοντέλου. Η ακρίβεια επικύρωσης που υπολογίζεται στο σύνολο δεδομένων δεν χρησιμοποιείται για εκπαίδευση, αλλά χρησιμοποιείται (κατά τη διάρκεια της εκπαιδευτικής διαδικασίας) για την επικύρωση (ή «δοκιμή») της ικανότητας γενίκευσης του μοντέλου. Η αύξηση της ακρίβειας επικύρωσης έδειξε ότι το μοντέλο μαθαίνει μετά το τέλος ορισμένων εποχών ότι εξάγει γνώση από το σύνολο εκπαίδευσης.

Μια άλλη μέτρηση που αξίζει να εξεταστεί είναι η συνάρτηση απωλειών εκπαίδευσης και επικύρωσης. Η συνάρτηση απωλειών εκπαίδευσης είναι μια μέτρηση που χρησιμοποιείται για την αξιολόγηση του τρόπου με τον οποίο ένα μοντέλο βαθιάς μάθησης ταιριάζει στα δεδομένα εκπαίδευσης. Δηλαδή, εκτιμά το σφάλμα του μοντέλου στο σετ εκπαίδευσης. Από την άλλη πλευρά, η απώλεια επικύρωσης είναι μια μέτρηση που χρησιμοποιείται για την αξιολόγηση της απόδοσης ενός μοντέλου βαθιάς μάθησης στο σύνολο επικύρωσης. Η συνάρτηση απώλειας επικύρωσης είναι παρόμοια με την απώλεια εκπαίδευσης και υπολογίζεται από το άθροισμα των σφαλμάτων για κάθε παράδειγμα στο σύνολο επικύρωσης. Ακολουθεί μια απεικόνιση της συμπεριφοράς απώλειας εκπαίδευσης και επικύρωσης κατά τη διάρκεια των εποχών (Σχήμα 5.2).

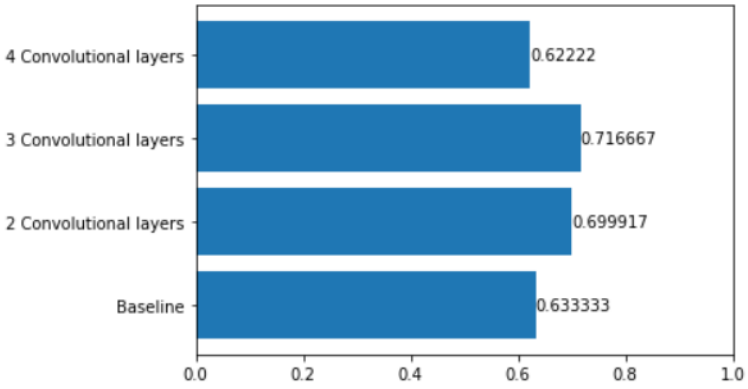


Σχήμα 20: *Η συνάρτηση απώλειας της εκπαίδευσης και της επικύρωσης αλλάζει κατά τη διάρκεια των εποχών (100)*

Πολυπολοκότητα CNN αρχιτεκτονικής

Συνεχίζοντας με το μοντέλο του CNN, ένας άλλος παράγοντας που συμβάλλει στην απόδοση του μοντέλου είναι η πολυπλοκότητά του. Ένα Συνελικτικό Νευρωνικό Δίκτυο αποτελείται συνήθως από τρία βασικά επίπεδα, ένα συνελικτικό επίπεδο, ένα στρώμα συγκέντρωσης και ένα πλήρως συνδεδεμένο επίπεδο. Ως αποτέλεσμα, υπάρχει μια πληθώρα διαφορετικών αρχιτεκτονικών ένα CNN που μπορεί να κατασκευαστεί χρησιμοποιώντας αυτά τα επίπεδα. Ένα εξαιρετικά περίπλοκο μοντέλο μπορεί να οδηγήσει σε υπερπροσαρμογή καθώς απομνημονεύει γρήγορα τα μοτίβα δεδομένων εκπαίδευσης, ενώ ένα πολύ απλό μπορεί να έχει το αντίθετο αποτέλεσμα καθώς μπορεί να μην αναγνωρίζει σημαντικά μοτίβα στα δεδομένα.

Στο Σχήμα 21 μπορούμε να παρατηρήσουμε ότι η βέλτιστη αρχιτεκτονική για τα δεδομένα μας είναι το CNN με 3 συνελικτικά επίπεδα. Είναι καλύτερο το βασικό μοντέλο (που έχει μόνο 2 συνελικτικά επίπεδα) γιατί αυξήσαμε την πολυπλοκότητα και, όπως αναφέρθηκε προηγουμένως, βελτίωσε την ακρίβεια καθώς το μοντέλο ήταν σε θέση να εξάγει περισσότερα χαρακτηριστικά και να αποκτήσει περισσότερες πληροφορίες από τα δεδομένα που τροφοδοτήθηκαν σε αυτό. Αν και υπάρχει ένα όριο στην πολυπλοκότητα του μοντέλου και εξαρτάται σε μεγάλο βαθμό από το σύνολο δεδομένων που χρησιμοποιείται, και κυρίως από το μέγεθός του. Σε αυτήν την περίπτωση, ο αριθμός των δειγμάτων δεδομένων είναι μικρός, επομένως, αντί να εξάγει περισσότερα χαρακτηριστικά, το μοντέλο αρχίζει να υπερπροσαρμόζεται. Αν και η ακρίβεια είναι καλύτερη στο σετ εκπαίδευσης, στο σετ δοκιμών γίνεται φτωχή όπως φαίνεται ξεκάθαρα στο Σχήμα 21.



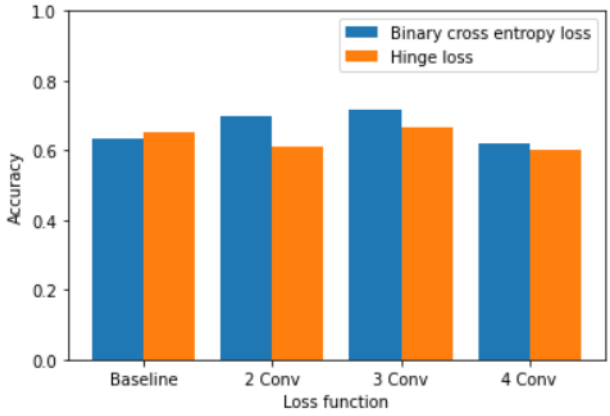
Σχήμα 21: *Ακρίβεια για 3 τύπους αρχιτεκτονικών CNN*

Συνάρτηση απώλειας

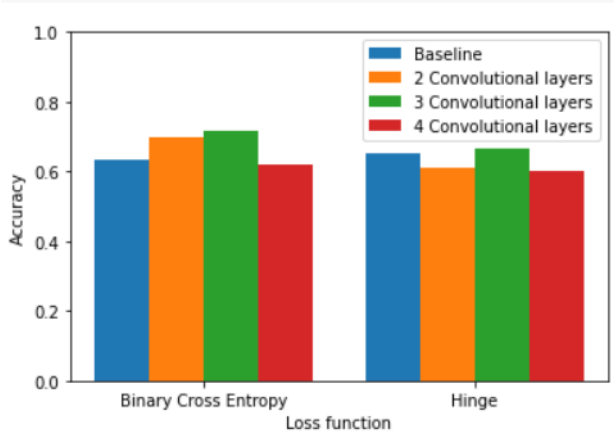
Ένα άλλο πείραμα που πραγματοποιήσαμε σχετικά με τα μοντέλα ΝΝ είναι σχετικά με τη συνάρτηση απώλειας. Η απόδοση των διαφορετικών αρχιτεκτονικών που δημιουργήθηκαν ελέγχεται σε δύο διαφορετικές συναρτήσεις απώλειας. Δεδομένου ότι το πρόβλημά μας είναι ένα πρόβλημα δυαδικής ταξινόμησης, δεν μπορούμε να έχουμε πολλές συναρτήσεις απώλειας. Επιλέξαμε τα 2 πιο βασικά για τα πειράματα προκειμένου να αποφασίσουμε ποιο είναι το καλύτερο για το μοντέλο μας. Όπως απεικονίζεται στο Σχήμα 22, η απώλεια δυαδικής διασταυρούμενης εντροπίας είναι αυτή που δίνει καλύτερη απόδοση στις περισσότερες αρχιτεκτονικές του ΝΝ. Για να είμαστε πιο ακριβείς για τις βαθύτερες αρχιτεκτονικές, καθώς στο βασικό μοντέλο η απώλεια άρθρωσης δίνει ελαφρώς υψηλότερη απόδοση. Στο Σχήμα 23 είναι σαφές ότι ανεξάρτητα από τη συνάρτηση απώλειας, η αρχιτεκτονική των 3 συνελικτικών επιπέδων δίνει τα βέλτιστα αποτελέσματα για τα δεδομένα μας.

Σύγκριση μοντέλων

Σε αυτήν την ενότητα πειραμάτων, θα συγκρίνουμε την ακρίβεια διαφορετικών μοντέλων και ταξινομητών που δημιουργήθηκαν προκειμένου να βρούμε τον καταλληλότερο για την ταξινόμηση της σχιζοφρένειας για το σύνολο δεδομένων μας. Ορισμένοι βασικοί ταξινομητές δοκιμάστηκαν, καθώς και μοντέλα και ταξινομητές που χρησιμο-



Σχήμα 22: *Συνάρτηση απώλειας για κάθε αρχιτεκτονική CNN*

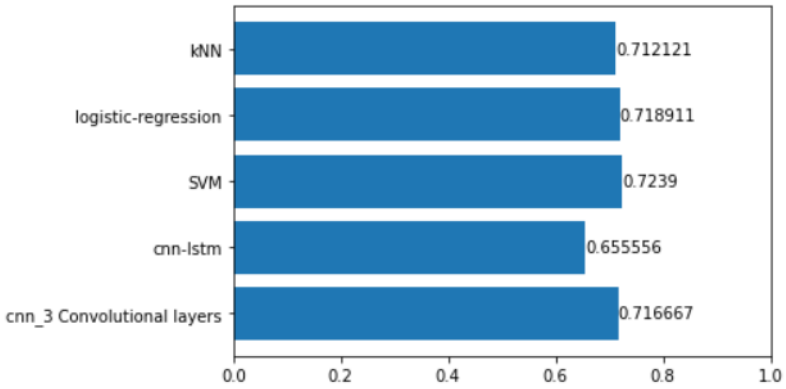


Σχήμα 23: *Ακρίβεια αρχιτεκτονικής CNN για κάθε συνάρτηση απώλειας*

ποιούνται συνήθως στη βιβλιογραφία, όπως SVM και LSTM. Για άλλη μια φορά, η ακρίβεια που απεικονίζεται για κάθε μοντέλο είναι ο μέσος όρος των ακρίβειων που υπολογίστηκαν σε τρία διαδοχικά πειράματα για κάθε μοντέλο.

Όπως μπορούμε να δούμε, δεν υπάρχουν ακραίες διαφορές στην ακρίβεια των μοντέλων μας. Φαίνεται ότι το SVM είναι το καταλληλότερο για τα δεδομένα μας που επιβεβαιώνουν την υψηλή παρουσία αυτής της μεθόδου που χρησιμοποιείται στη βιβλιογραφία για προβλήματα ταξινόμησης φMPI. Το μοντέλο CNN αποδίδει επίσης πολύ καλά καθώς και οι βελτιστοποιημένοι ταξινομητές kNN και λογιστικής παλινδρόμησης. Το υβριδικό μοντέλο CNN-LSTM απέδωσε επίσης ικανοποιητικά καλά, αν και είναι λογικό το ΎNN να έχει καλύτερη απόδοση καθώς το LSTM χρησιμοποιείται συνήθως για την επεξεργασία και την πραγματοποίηση προβλέψεων δεδομένων ακολουθιών δεδομένων (όπως χρονοσειρές), ενώ το CNN έχει σχεδιαστεί για να εκμεταλλεύεται τη ¨χωρική συσχέτιση' δεδομένα, επομένως λειτουργεί καλά σε δεδομένα ιατρικών εικόνων.

Αν και, η ακρίβεια είναι μια καλή μέτρηση, δεν αρκεί. Υπάρχουν και άλλες μετρήσεις που είναι κρίσιμες για την αξιολόγηση του μοντέλου. Ένα από αυτά είναι η μέτρηση ανάκλησης, δηλαδή μια μέτρηση που ποσοτικοποιεί τον αριθμό των σωστών θετικών προβλέψεων που έγιναν από όλες τις θετικές προβλέψεις που θα μπορούσαν να είχαν γίνει. Στην περίπτωσή μας, αξιολογεί πόσα από τα θετικά άτομα (εννοεί τους ασθενείς με σχιζοφρένεια) ταξινομήθηκαν σωστά. Όπως φαίνεται στους αλγόριθμους Σχήμα 25 kNN και cnn-lstm, αν και με υψηλές ακρίβειες, δεν έχουν τόσο υψηλή απόδοση στη μέτρηση ανάκλησης. Τα άλλα τρία μοντέλα από την άλλη τα πάνε καλά με τη μέτρηση ανάκλησης μαζί με την ακρίβεια , με το CNN (με 3 Convolutional layers) να έχει την υψηλότερη ανάκληση.



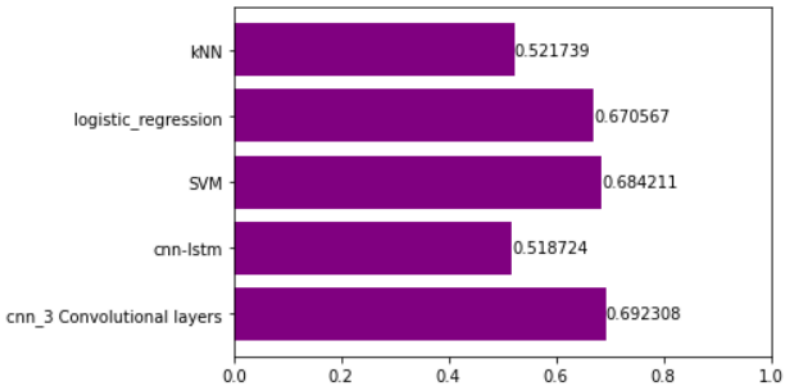
Σχήμα 24: *Ακρίβεια για διαφορετικά μοντέλα*

Μελλοντική εργασία

Στα πλαίσια της διπλωματικής εργασίας δημιουργήθηκε και ένα Συνελικτικό δίκτυο γράφου για να μπορέσουμε να ενσωματώσουμε στην γνώση μας και φαινοτυπικά δεδομένα.

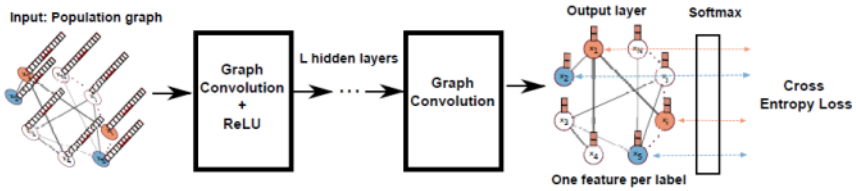
Συνελικτικά δίκτυα γράφων

Η αρχιτεκτονική του μοντέλου μας απεικονίζεται στο Σχήμα 26. Το μοντέλο αποτελείται από ένα πλήρως συνελικτικό ΓΎΝ με Λ κρυφά επίπεδα που ενεργοποιούνται



Σχήμα 25: *Μετρική recall για διαφορετικά μοντέλα*

χρησιμοποιώντας τη λειτουργία Ρεστιφιεδ Λινεαρ Υνιτ (ΡεΛΥ). Το επίπεδο εξόδου ακολουθείται από μια λειτουργία ενεργοποίησης σοφτμαξ. Το γράφημα εκπαιδεύεται χρησιμοποιώντας ολόκληρο το γράφημα του πληθυσμού ως είσοδο. Επιπλέον, χρησιμοποιούμε μια συνάρτηση απώλειας διασταυρούμενης εντροπίας για τη διαδικασία βελτιστοποίησης. Μετά την εκπαίδευση του μοντέλου ΓΝΝ, οι ενεργοποιήσεις σοφτμαξ υπολογίζονται στο δοκιμαστικό σύνολο και στους κόμβους χωρίς ετικέτα αποδίδονται οι ετικέτες μεγιστοποιώντας την έξοδο σοφτμαξ.



Σχήμα 26: *Συνελικτικά δίκτυα γράφων*

Chapter 1

Introduction

1.1 Schizophrenia and diagnosis

Schizophrenia (SZ) is a serious psychiatric disorder that affects a person's feelings, social behavior and perception of reality. Although the biological causes are not yet established, genetic and environmental factors, such as prenatal stress, traumatic experiences or extensive drug use, can be crucial components for the development of this disorder. Animal models show that developmental hippocampal lesions are causing dis-connectivity of the prefrontal cortex. Schizophrenia affects approximately 24 million people or 1 in 300 people (0.32%) worldwide. This rate is 1 in 222 people (0.45%) among adults. It is not as common as many other mental disorders. It is possible for individuals with schizophrenia to live a normal life, but only with good treatment. There's no sure way to prevent schizophrenia, but sticking with the treatment plan can help prevent relapses or worsening of symptoms. In addition, researchers hope that learning more about risk factors for schizophrenia may lead to earlier diagnosis and treatment. So far, diagnosis of schizophrenia is possible through psychiatric observation based on behavioural criteria. Although, the fact that there are indications that schizophrenia is strongly related to the way the brain regions are connected with each other, has led scientist's attention to other methods of diagnosis like fMRI methods. Machine learning techniques can contribute to this effort as it is a useful tool that can gain insight from the fMRI data.

1.2 Objectives

The target of this diploma thesis is to build a model to be able to predict the probability that a potential patient is suffering from schizophrenia based on fMRI (Functional magnetic resonance imaging) data. It is an alternative approach for the diagnosis of this diseas in a physiological level, without using psychiatric or phycological information for the potential patients. Over the years AI has been used to detect mental illnesses via different means. The most common techniques for schizophrenia detection using AI include PET scans, EEG, techniques that involve

gene and protein classification [1] and magnetic resonance imaging (MRI) . MRI is a medical imaging technique used in radiology to depict the anatomy and physiological processes of the body. For the purpose of the diploma thesis an open source dataset has been used that was obtained from OpenNeuro [2]. In order for the dataset to be created, scanned images of the brain were taken from both patients diagnosed with SZ and healthy controls. The subject of this diploma thesis is an interesting yet challenging subject as it delves and explores multiple scientific fields: medical studies about mental disorders, brain imaging data and analysis of the data and of course neural networks and machine learning algorithms. This diploma thesis aims to use the previous methods of classification and and propose a novel one, using not only the fMRI data from the brain but also additional information such as demographic characteristics.

1.3 Thesis Outline

The current diploma thesis is organised as follows. In Chapter §1, a brief introduction to the problem is provided, some fundamental definitions and the goals of this thesis. In Chapter §2 a theoretical background is given regarding the Artificial Intelligence field and the Neural Networks used in the scope of the current diploma thesis. All the models and classifiers that was used are analyzed and explained thoroughly. In Chapter §3 we explain the reason that fMRI are useful for schizophrenia diagnosis and additionally we give a brief explanation of what is the fMRI method how it is used. The dataset used to conduct all the experiments is described in Chapter §4 along with the preprocessing procedure followed to transform the data to correlation matrices. Finally, in Chapter §5 the experiments conducted to evaluate the model's performance are presented.

Chapter 2

Theoretical background

2.1 Machine Learning and Neural Networks

Machine Learning (ML) is the technology of developing computer algorithms that are able to emulate human intelligence. These algorithms are built as to be able to improve automatically through experience and by the use of data, known as training data. Machine learning algorithms can be classified into separate classes according to the nature of the data, the learning process, and the model type[3]. ML is seen as a part of Artificial Intelligence and it is used to make predictions or decisions based on the learning experience the model gained through training. To this day ML technology has been applied to such diverse fields as pattern recognition [4] , computer vision [5], spacecraft engineering [6], finance [7], entertainment [8],[9], ecology [10], computational biology [11],[12], and biomedical and medical applications [13],[14].

2.2 Machine Learning Methods

There are three types of machine learning: supervised learning, unsupervised learning, and reinforcement learning. Although in this diploma thesis, supervised learning is used for the multi-classification, a brief introduction of both methods follows for clearer understanding.

Supervised learning

The defining characteristic of supervised learning is the availability of annotated data. To be more precise, supervised learning entails learning a mapping between a set of input variables and an output variable, called label and after that this mapping is applied to predict the values for the unseen data[15]. Having the data labeled, hence knowing the correct output for each input the model will be trained over time, measuring the accuracy through a loss function adjusting until the error has been sufficiently minimized.

There are two types of supervised learning techniques in machine learning: regression and classification.

- *Regression* is used for prediction of a continuous variable based on the relationship between the input variables and output variable learned during the training. For example, regression can be useful for house price prediction, with the price of the house as output and inputs could be variables such as locality, size of house etc.

- *Classification* is used when the output variable is categorical. Thus, it is useful for grouping the output inside a class. If the algorithm tries to label input into two distinct classes, it is called binary classification. Selecting between more than two classes is referred to as multi-class classification, which is the case in this diploma thesis.

Unsupervised learning

On the contrary of the aforementioned method, there are cases in which labeled data are not possible to be obtained or very strenuous to create. To solve these type of cases, unsupervised learning techniques are used to find hidden patterns from the given dataset. This type of learning can be compared with the procedure taking place in the human brain while learning new things. As there is no corresponding output data for the input data unsupervised learning cannot be directly applied to regression or classification problems. The goal of unsupervised learning is to find the underlying structure of dataset, group that data according to similarities, and represent that dataset in a compressed format. The unsupervised algorithm can be further categorized into clustering and association problems[16].

- *Clustering* is a method that attempts to group the objects based on the similarity between them, such that objects with most similarities remain into a group and have less or no similarities with objects in another group.

- *Association* is used to detect the relationships between variables in a large database. It is commonly used for marketing strategies, such as people who buy X item are more likely to buy item Y.

Reinforcement learning

Reinforcement learning is a sub-field of machine learning that deals with the problem of training an agent to maximize a reward signal while acting in an environment. This method is based on rewarding desired behaviors and/or punishing undesired ones, hence a reinforcement learning agent is able to learn through trial and error. The main goal of reinforcement learning is to define the best sequence of decisions the agent has to follow to solve a problem while maximizing a long-term reward. This is why it is primarily applied for motion planning, dynamic pathing, controller optimization, scenario-based learning policies for highways etc. A characteristic example of the adequacy of the method is its use for parking that can be achieved by learning automatic parking policies.

2.3 Feature engineering

Feature engineering is the process of manipulating the raw data into meaningful information, thus features that can be used in both supervised learning and unsupervised learning. To begin with, a "feature" is any measurable input that can be used in a predictive model. Feature Engineering encapsulates various data engineering techniques such as selecting relevant features, handling missing data, encoding the data, and normalizing it. It is one of the most important tasks and plays a crucial role in determining the outcome of the model.

There is an abundance of benefits of feature engineering, for instance there is greater flexibility of the features and as a result it becomes easier for algorithms to detect patterns in the processed data rather that the raw data. The most important benefit is that an effective feature engineering implies higher efficiency of the model, thus better accuracy and better results.

In this diploma thesis, a few feature engineering techniques have been used as the dataset is pre-processed and the data is not completely raw.

Preprocessed dataset

In the COBRE dataset [2] there is a file called the confounds file. Confounds are created during the brain scan and can alter the scan's representation accuracy. The broad purpose of resting-state FMRI is to use the common variance of the FMRI blood oxygenation level dependent (BOLD) signals in different regions of the brain as an indicator of synchronous neural activity. However, in resting-state FMRI, functional connectivity is determined by measuring the temporal similarity of the BOLD time series in voxels using some metric, commonly the correlation coefficient[17]. For example, in the original Biswal paper [18], the correlation coefficient between the BOLD time series of a voxel in the motor cortex and every other voxel in the brain was calculated. Correlation coefficient reflects how similar the measurements of two or more variables are across a dataset. Voxels whose correlation coefficient passed a statistical threshold were considered to be functionally connected, thus revealing common spontaneous fluctuations between left and right motor cortices. Since the two time series are measured simultaneously, any non-neural activity-related process that affects one or both time series will affect the measure of functional connectivity, thus yielding a spurious result. These resting-state FMRI confounds can not only increase the apparent functional connectivity by introducing bogus similarities between the time series' but also reduce the connectivity metric if differential confounds between regions are introduced.

FMRI confounds can arise by many processes in the MRI environment. Apart from hardware disabilities of the scanner (e.g. spiking), FMRI confounds can arise from flaws of the participant, such as movement or cardiac and respiratory noised that cannot be avoided during the experiment. The confounds file contains all the information regarding the confounds and it is used in the correlation matrix calculation in order for the confounds bad effects to be taken into consideration. There are some subjects that are missing the confounds files, therefore those subject's features

are removed from the feature vector.In addition, inside the confounds file, there were NaN values that was replaced with zeros.

ROIs extraction

Voxel-wise based approaches, which are widely used in mental disorders detection and use voxels as features, have a major problem specifically the dimensionality. To be more precise, in voxel-wise analysis the number of features is very large compared to the number of available to the number of available training samples [19]. To overcome this problem, voxel grouping is performed using regions of interest (ROIs). ROIs are determined using atlases, such as Harvard-Oxford Atlas, which is a probabilistic atlas covering 48 cortical and 21 sub-cortical structural areas.

The fMRI data obtained from the scanner represents the whole brain region that was scanned, thus the whole brain. Therefore, by performing a ROIs analysis using the Harvard-Oxford Atlas we use the meaningful regions of the brain containing the useful information that will help the model differentiate the subjects based on their mental disorders,removing from the feature set those parts of the brain that do not offer any useful information.

2.4 Machine learning approach

Machine learning is the general term to describe the procedure followed by a computer to learn from data. A machine learning algorithm is a computational process that uses the input data to perform a task without being explicitly programmed to do so, instead they recognize patterns in data when it arrives and make predictions.As mentioned earlier machine learning algorithms can be divided into supervised, unsupervised and reinforcement learning. The broad categories of classification and regression algorithms were mentioned before as well. Although in this diploma thesis we are dealing with a classification problem, some regression algorithms are also worth mentioning.

2.4.1 Classification algorithms

For the purpose of the diploma thesis, an abundance of algorithms were explored. We will thoroughly discuss the classification algorithms used in this diploma thesis. In this chapter, a brief introduction and explanation of these algorithms will be illustrated. The results and comparison of the algorithms performance will be analyzed in chapter 5.

Linear regression

Linear regression is used to estimate real values based on continuous variables. For instance, the prediction of a house price, total sales etc. The goal is to find the best fit line, known as the regression line which is represented be the equation $y = \beta_0 + \beta_1 * x + \epsilon$, where ϵ is the error thus the difference between the observed value y and the straight line ($\beta_0 + \beta_1 * x$) [20]. There are two types of linear

regression: Simple Linear Regression and Multiple Linear Regression. The former one is characterized by one independent variable, whereas the latter one by multiple (more than 1). In the following Figure 2.1 an example of simple linear regression with one independent variable is illustrated.

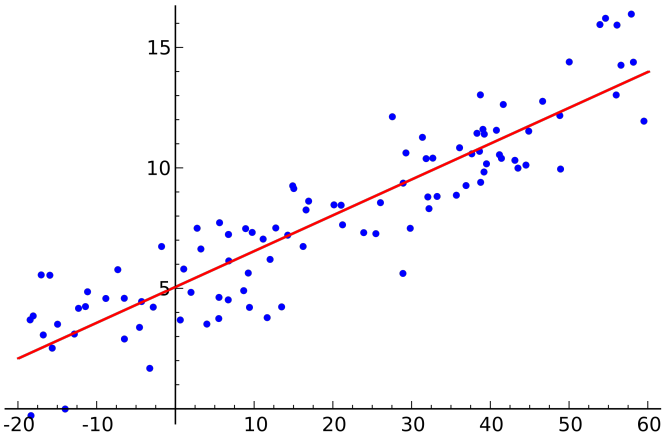


Figure 2.1: *Simple linear regression*

For the scope of the diploma thesis classification algorithms will be analyzed from now on, as the detection of Schizophrenia is a classification problem. There are two approaches that can be followed in order to classify our subjects into control subjects and mentally ill subjects. The first approach, that will be discussed in this section is the use of classification algorithms, whilst the second approach will be thoroughly discussed in the next section as it regards deep learning approach.

2.4.2 Logistic regression

Despite its name the logistic regression [28] algorithm is used in classification problems and not in regression problems as the linear regression. Instead of fitting a straight line or hyperplane, like in the linear regression model, the logistic regression model used the logistic function to squeeze the output of a linear equation between 0 and 1. The goal is to model the probability of a random variable Y being 0 or 1 (for binary classification) given experimental data. The logistic function is defined as Equation 6 and it can be considered as a generalized linear model function parameterized by θ. The logistic function is a sigmoid function and it is illustrated in the Figure 2.2

$$h_\theta(X) = \frac{1}{1 + e^{-\theta^T X}} = Pr(Y = 1|X; \theta) \tag{6}$$

The equation used by logistic regression for representation is very similar to the linear regression's. Therefore, the input values (x) are combined linearly using weights or coefficient values to predict an output variable. The difference with the linear regression model is that the output value is modeled as a binary output (0 or

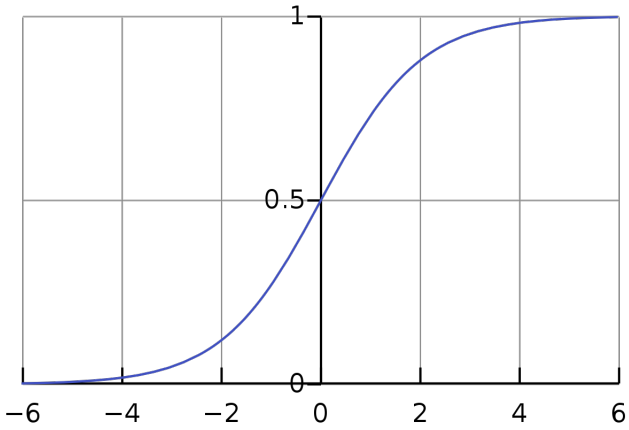


Figure 2.2: *Logistic function*

1) rather than a numeric value. Thus, the logistic regression equation is defined as follows,

$$y = \frac{e^{\beta_0+\beta_1*x}}{1+e^{\beta_0+\beta_1*x}} \tag{7}$$

,where y is the predicted output, b0 is the bias or intercept term and b1 is the coefficient for the single input value (x). In other words, logistic regression is a linear model but the output is transorformed using the logistic function.

Maximum-likelihood estimation (MLE) is an algorithm used by the logistic regression algorithm in order for the coefficients (beta values) of the algorithm to be estimated from the training data. The MLE algorithm searches for coefficient values that minimize the error in the probabilities predicted by the model those in the data. This is mainly implemented using efficient numerical optimization algorithms. Such algorithms can be selected as a parameter in the logistic regression classifier using the sklearn library, therefore is an extra hyper-parameter tuning in the classification procedure. There are several optimizers that can be used in the classifier but not all of them are suitable for a multiclass problem like the current one. As a result, only the ones used for the experiments are referred.

Before we dive in to the optimizer functions there are some terms that need to be clarified first.

- The Hessian of a function f(x,y) is a matrix of second order partial derivatives formed from all pairs of variables in the domain of f. Suppose we have a function f of n variables $f : R^n \longrightarrow R$. The Hessian of the function f is given in the following figure Figure 2.3

- A quadratic function is one of the form $f(x) = a * x^2 + b * x + c$, where a,b and c are numbers with a not equal to zero.

- Linear approximation: Given a function, $f(x)$, we can find its tangent at $x = a$. The equation of the tangent line $\mathcal{L}(x)$ is: $\mathcal{L}(x) = f(a) + f'(a) * (x - a)$. A

$$\mathbf{H}_f = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \, \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \, \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \, \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \, \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \, \partial x_1} & \frac{\partial^2 f}{\partial x_n \, \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix},$$

Figure 2.3: *Hessian matrix*

shown in the Figure 2.4 near $x = a$ the tangent line and the function have nearly the same graph. On occasion, we will use the tangent line, $\mathcal{L}(x)$, as an approximation to the function, $f(x)$, near $x = a$. In these cases, we call the tangent line the "Linear Approximation" to the function at $x = a$.

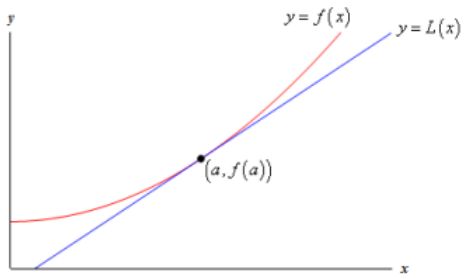


Figure 2.4: *A function and its tangent*

- Quadratic approximation:

  Same like linear approximation, yet this time we are dealing with a curve where we cannot find the point near to 0 by using only the tangent line. The reason we need a point near zero because the slope of the tangent line in the minimum cost point (global optima) is zero. In this case we use the parabola, as shown in Figure 2.5. In order to fit a good parabola, both parabola and quadratic function should have same value, same first derivative, AND the same second derivative.

- Penalization is used to avoid overfitting by penalizing the algorithm for fitting a model that fits the training data tightly. L1 regularization penalizes the sum of absolute values of the weights, whereas L2 regularization penalizes the sum of squares of the weights.

Newton's Method

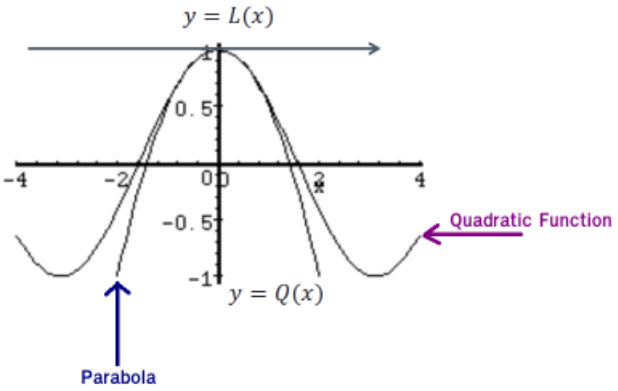


Figure 2.5: *Parabola*

Newton's method ("newton-cg" in sklearn) is using the quadratic approximation in order to mimic the loss function, thus the quadratic function that is to be minimized in order to minimize the error. It is like a twisted Gradient Descent with the Hessian. The disadvantages of the Newton's method are that it is computationally expensive due to the computation of the Hessian and that it attracts saddle points, which are stable points where the function has a local maximum in one direction, but a local minimum in another direction. Although, Newton's method is expensive at each iteration, it has very fast convergence rates.

Limited-memory Broyden–Fletcher–Goldfarb–Shanno Algorithm

Limited-memory Broyden–Fletcher–Goldfarb–Shanno Algorithm ("lbfgs" in sklearn) is analogue to the Newton's method, with the difference that the Hessian matrix is approximated using an estimation to the inverse Hessian matrix. With the term Limited-memory it is indicated that it stores only a few vectors that represent the approximation implicitly. The most important disadvantage of this solver is that it may not coverage to anything.

Stochastic Average Gradient

Stochastic average gradient is a method for optimizing the sum of a finite number of smooth convex functions. The SAG method's iteration cost is independent of the number of terms in the sum [29]. It is faster than other solvers for large datasets, when both the number of samples and the number of features are large. A drawback is that it only supports L2 penalization.

SAGA

The SAGA solver is a variant of SAG that also supports the non-smooth penalty L1 option (i.e. L1 Regularization). This is therefore the solver of choice for sparse multinomial logistic regression and it's also suitable for very large dataset.

2.4.3 Support vector machine

Support vector machine (SVM) [21] is a highly used algorithm that is used for both regression and classification problems. The objective of the support vector machine

algorithm is to find a hyperplane in a N-dimensional space, where N is the number of features that can classify the data points. It uses a simple mathematical model $y = w * x\prime + \gamma$, and manipulates it to allow linear domain division[22]. SVM can be divided into linear and non-linear models [23]. Linear support vector machine can divide the data with a linear line or hyperplane to separate the classes in the original domain. On the other side, non-linear support vector machine indicates that the data domain cannot be divided linearly and can be transformed to a space called the feature space where the data can be divided linearly.

As shown in the Figure 2.6 there are many possible hyperplanes that can be chosen, in order to separate the two classes of data points. The purpose is to find the plane that has the maximum margin, thus the maximum, distance between data points from both classes, as illustrated in the Figure 2.7. The data points with the minimum distance to the hyperplane are called Support Vectors and influence the position and the orientation of the hyperplane. If we delete the support vectors the position of the hyperplane will change. Using these support vectors we maximize the margin of the classifier, as depicted in Figure 2.8.

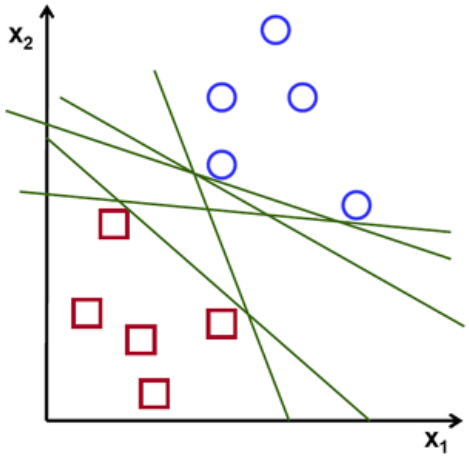


Figure 2.6: *Support vector machine possible hyperplanes*

The simplest type of SVM as explained before is used for binary classification problems dividing the data point into two categories 0 or 1. In order to perform multiclass classification, which is the scope of this diploma thesis, we use the same principle as in binary classification. To be more precise, the multiclass problem is broken down into multiple binary classification problems, using a heuristic function. There are two types of heuristic functions:

- One-vs-Rest:

  This method simple splits the multiclass data into binary classification data so the binary classification algorithms can be applied to the binary classification data. In this technique the **N**-class instances are divided into **N** binary classifier models.

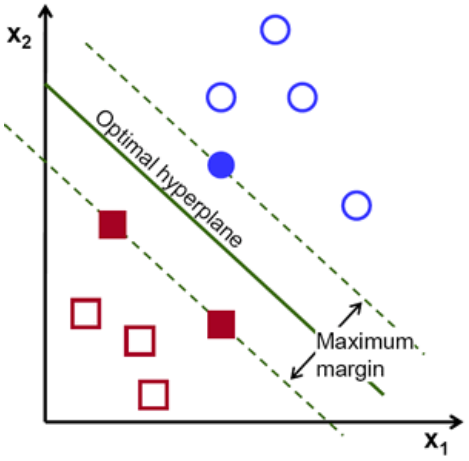


Figure 2.7: *Support vector machine optimal plane*

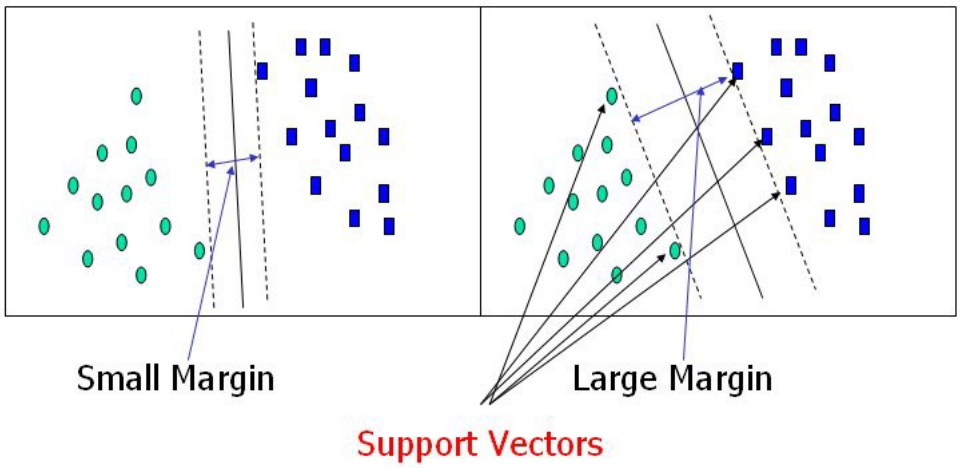


Figure 2.8: *Support vectors*

- One-vs-One:

  This method is similar to the One-vs-Rest method as it is based on splitting the multiclass data into binary classification data, but the splitting behaviour is different. In this technique the **N**-class instances are divided into $\frac{\mathbf{N}*(\mathbf{N}-\mathbf{1})}{\mathbf{2}}$ binary classifier models.

SVM is also called kernelized SVM as it uses the kernel function method to take data as input and transform it so that a non-linear decision surface is able to transform to a linear equation in a higher number of dimension spaces. The transformation is possible if we use the score calculated by calculating the distance score of two datapoints. This score is higher for closer datapoints and vice versa. We will discuss the most popular kernel functions that are used in the experiments and are available in scikit-learn.

Linear Function

Linear function is used when data is linearly separable,that is, it can be separated using as single line. It is referred for the sake of completeness as it cannot be used for multiclass classification problems. The kernel function is defined as:

$$k(x, y) = x * y \tag{8}$$

Polynomial Function

Polynomial function represents the similarity of the training samples in a feature space over polynomials of the original variables used in the kernel, allowing learning of non-linear models. For degree-d polynomials, the polynomial kernel is defined as:

$$k(x, y) = (x^T y * c)^d. \tag{9}$$

where x and y are vector in the input space, and c ≥ 0 is a free parameter trading off the influence of higher-order versus lower-order terms in the polynomial.

Gaussian Kernel Radial Basis Function (RBF)

The value of the RBF kernel decreases with distance and ranges between zero and one it can be portrayed as a similarity measure, thus real-valued function that quantifies the similarity between two objects. The RBF kernel on two samples x and y, represented as feature vectors in some input space, is defined as:

$$k(x, y) = e^{-\gamma \|x-y\|^2} \tag{10}$$

Sigmoid Function

Is similar to the sigmoid function in logistic regression. The sigmoid kernel comes from the neural network field, where the bipolar sigmoid function is often used as an activation function for artificial neurons, as we will see in the next section section 2.5. An SVM model using a sigmoid kernel function is equivalent to a two-layer, perceptron neural network. The definition of the sigmoid kernel function is:

$$k(x, y) = tanh(\gamma x^T y + r) \tag{11}$$

2.4.4 k-Nearest Neighbor

The k-nearest neighbor is a non-parametric supervised algorithms used in either regression or classification problems, which uses proximity to make predictions about the classification of a data point. It is most used for classification problems, working with the assumption that similar point can be found near one another. We will discuss k-NN classification and not regression as it serves the scope of the current problem. The input consists of the k closest training examples in a data set and the output is a class membership. The procedure in which a point is classified is called majority vote of its neighbors. Each object is assigned in the class that is more common among its k nearest neighbors, where k is a positive integer, typically a small integer. Therefore, if k=1, then the object is simply assigned to the class of that single nearest neighbor.

In order to regulate which data points are closest to a given data point, the distance between these data point must be calculated. There are several distance metrics that we can choose from with the Euclidean distance being the most common for continuous variables and Hamming distance for discrete variables.

Euclidean distance

Euclidean distance is limited in real-valued vectors and it measures a straight line between the query point (data point that needs to be labeled) and the other points being measured. The euclidean distance is calculated using the following formula:

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (y_i - x_i)^2} \tag{12}$$

Hamming distance

This technique is most commonly used with Boolean or string vectors and it measures the minimum number of errors that could transformed one string into the other. The above can be represented by the following formula: —

$$D_H = \sum_{i=1}^{k} |x_i - y_i| \tag{13}$$

Another crucial parameter for the k-NN algorithm that needs to be tuned is the k value. The k value in the k-NN algorithm defines how many neighbors will be checked before the query point is classified. The choice of k can define whether the

algorithm will overfit or underfit. Lower values of k can have high variance, but low bias, whilst larger values of k may have high bias and lower variance. Let's dive in a little bit more in theory.

Bias

Bias is the difference between the average prediction of the model and the correct value that the model is trying to predict. When a model has high bias it means that it pays little attention to the training data, thus it oversimplifies the model. As a result it leads to high error on training and test data.

Variance

Variance refers to the changes in the model when using different portions of the training set. When variance is high it means that the model pays a lot of attention to training data. As a result, such models perform very well on training data but has high error in test data.

As a conclusion, to build a good model we need to find a good trade-off between bias and variance such as to minimize the total error and avoid underfitting or overfitting Figure 2.9.

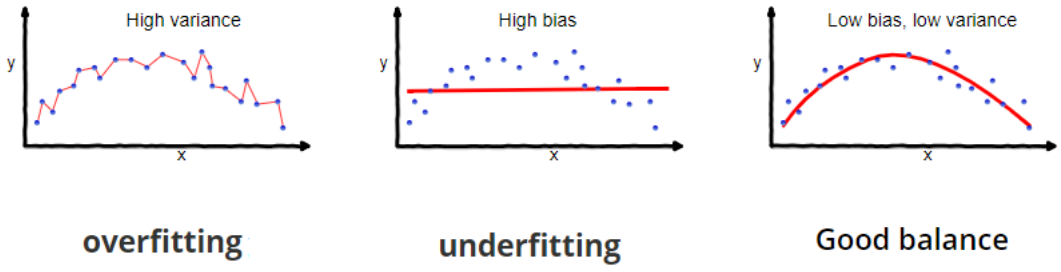


Figure 2.9: *Bias-variance trade-off*

2.4.5 Loss Function

In simple terms the loss function is the function that computes the difference between the expected output value and the current output value of the algorithm. In particular it is a method to evaluate how good the algorithm models the dataset. If our predictions are very deviant from the actual results the loss function output will be a large number.If the predictions are pretty good, the loss function will output a lower number.

There are various loss function used for different kinds of machine learning algorithms. There is not one loss function that fits all problems in machine learning. Broadly, the two major categories are regression losses and classification losses. We will further analyze the classification losses that are used in the experiments.

Hinge Loss/Multi class SVM Loss

Hinge loss is most notably used for support vector machines (SVMs) as it is used for maximum-margin classification. In particular, the score of the correct category should be greater that the sum of scores of all incorrect categories by some safety margin. Let t be the actual output, where $t = \pm 1$, and y is the classifier score. The hinge loss of the prediction y is defined as:

$$l(y) = max(0, 1 - t * y) \tag{14}$$

When t and y have the same sign, which means that y predicts the right class and $|y| \leq 1$, the hinge loss l(y) is 0. If t and y have the same sign but $|y| < 1$, which means that the prediction is correct but not by enough margin and if t and y have opposite signs then l(y) increases linearly with y.

Cross Entropy Loss

Cross entropy loss is used to measure the performance of a classification problem with probability values as the output, meaning output values in the range 0 to 1. The loss increases as the predicted probability diverges from the actual value. Cross-entropy builds upon the idea of entropy from information theory and calculates the number of bits required to represent or transmit an average event from one distribution compared to another distribution. Mathematically, it is the preferred loss function under the inference framework of maximum likelihood. It is the loss function to be evaluated first and only changed if you have a good reason.

Cross-entropy will calculate a score that summarizes the average difference between the actual and predicted probability distributions for predicting class 1. The score is minimized and a perfect cross-entropy value is 0.

2.5 Deep learning approach

Deep learning extends classical machine learning by adding more "depth", meaning more complexity into the model. It can be characterized as a sophisticated and mathematically complex evolution of machine learning. In addition, data is transformed using various functions that allow data representation in a hierarchical way, through several levels of abstraction [24],[25]. One of the main advantages of deep learning is the ability to solve complex problems particularly well and fast, and the reason for this is that it allows massive parallelization [26]. Deep learning algorithms begin to become popular for mental disorder classification and multi-class classification problems like the one we are discussing in this diploma thesis, as deep learning can increase classification accuracy providing that there are sufficient large datasets available describing the problem.

Deep learning consists of an abundance of different components, for instance convolutions, pooling layers, fully connected layers, gates, memory cells, activation functions, encode/decode schemes etc., depending on the network architecture that is been used. A crucial attribute of deep learning is that it is greatly flexible and

adaptable for a wide variety of hugely complex challenges, from a data analysis perspective, due to the highly hierarchical structure and vast learning capacity of deep learning models [26].

2.5.1 Artificial Neural Networks structure

Artificial neural networks (ANNs) are computational systems that mimic the biological neural networks that constitute animal brains. The basic component of a neural network is a unit or node called artificial neuron, which model the neurons in a biological brain. A neuron is an electrically excitable cell that communicates with other cells via specialized connections called synapses. As we see in the Figure 2.10 a typical neuron consist of a cell body called soma, dendrites and a single axon. The soma is a compact structure and the axon and dendrites are threads extruding from the soma. Dendrites fork in a great extend typically a few hundred micrometers from the soma. The axon leaves the soma in the axon hillock which looks like a swelling and extends for as far as 1 meter in humans. At the axon's tip there are branches called the axon terminals, where the neuron can transmit a signal across the synapse to another cell.

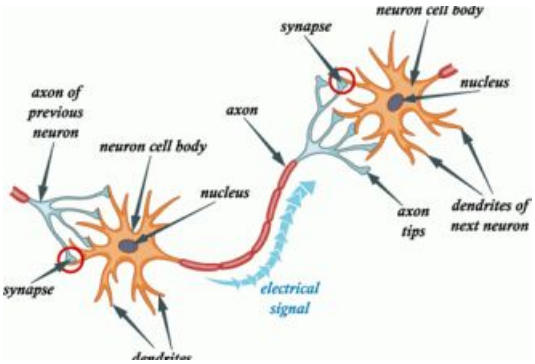


Figure 2.10: *Biological neurons synapse. Image source* `https://pulpbits.net`

In parallel to the biological structure a typical artificial neuron is depicted in Figure 2.11. An artificial neuron is a mathematical function conceived as a model of biological neurons, a neural network. An artificial neuron receives one or more inputs which represents the postsynaptic part of the biological neuron the dendrites and sums them to produce an output (or activation) which represents a neuron's action potential which is transmitted along its axon. Usually each input (or edge) is separately weighted (w_1,w_2,...,w_m) so it can mimic the excitability of the biological neuron. The output is calculated as the sum of the multiplication of each input by the corresponding edge weight. This functions that is used to calculate the output is called transfer function. Consequently, the final value calculated by the transfer function is compared with a threshold through another function called the activation function. If the value is greater than the threshold the artificial neuron is "fired" and pass its information to the next neuron.

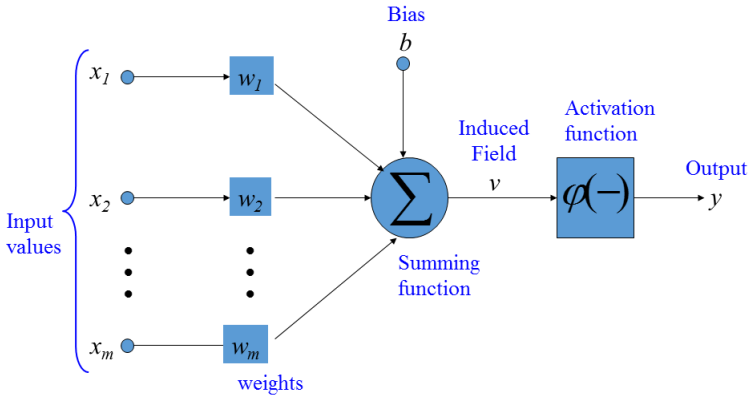


Figure 2.11: *Artificial neuron. Image source:* `https://www.gabormelli.com/RKB/HomePage`

The structure of an Artificial Neural Network (ANN) is shown in Figure 2.12. As it can be observed from the figure the neurons are organized into multiple layers. Each layer can have different number of neurons and the neurons in the same level are not connected to each other but connect only to neurons of the immediately preceding and immediately following layers. Inputs are fed into the network through the input layer. The final output is produced in the output layer. In between there can be zero or more hidden layers. In the case of multiple hidden layers, the network is referred to as Deep Neural Network (DNN).

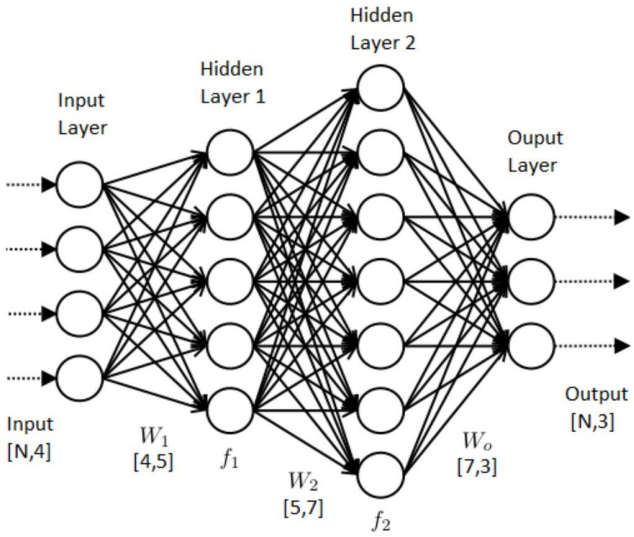


Figure 2.12: *Artificial Neural Network. Image source:* `https://www.datasciencecentral.com`

2.5.2 Activation functions

Activation functions are a crucial part of the effectiveness of the neural network. The predictions of the model are highly affected by the activation function choice, as this choice controls how well the network model is trained given the dataset. As mentioned before, an activation function determines the output of the neuron that will be transmitted in the next layer. An activation function can simple be binary that turns the neuron on and off depending on the input. It can also make a transformation of the input signal to an output signal in the range of -1 to 1. The activation functions can basically be divided into two types, the linear and the non-linear activation functions. The most used activation function are non-linear functions because it makes it easy for the model to generalize or adapt with variety of data and to differentiate between the output.

Sigmoid Function

The Sigmoid function is also called the logistic function and its a mathematical function with the characteristic shape of an "S" curve. The sigmoid function takes any real number as an input and produces output values in the range 0 to 1. Therefore it is most used in models where the probability is to be predicted as the value. Sigmoid function is shown in the Figure 2.13 and it is defined for all real input values by the formula:

$$S = \frac{e^x}{e^x + 1} \tag{15}$$

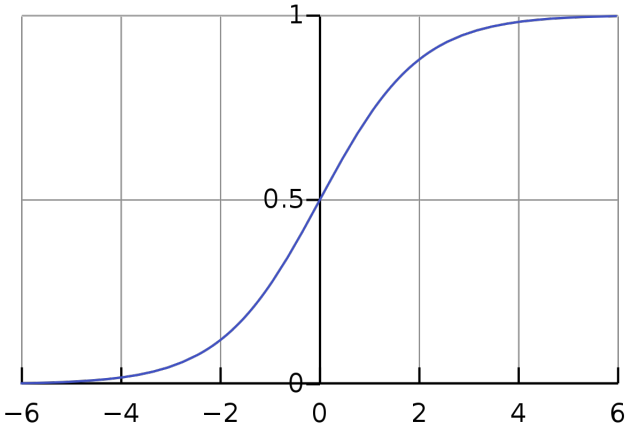


Figure 2.13: *Sigmoid function*

Tanh/Hyperbolic Tangent Function

The tanh function is very similar to the sigmoid function, also shaped as "S" curve, but produces output values in the range -1 to 1. Tanh function is defined by the formula:

$$tanh = \frac{e^2x - 1}{e^2x + 1} \tag{16}$$

The advantage is that negative inputs will be mapped as strongly negative and the zero inputs will be mapped near zero in the tanh graph. Tanh function is shown in the Figure 2.14.

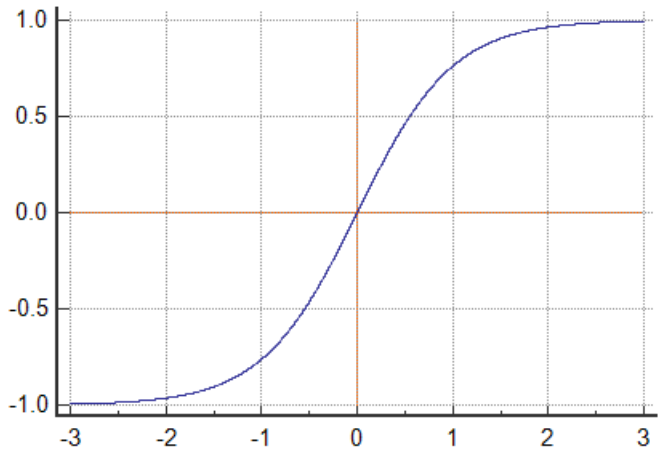


Figure 2.14: *Tanh/Hyperbolic Tangent function*

ReLU (Rectified Linear Unit) Function

The ReLU function is the most used activation function for hidden layers. The reason for this besides the simplicity of implementation is that it is effective at overcoming the limitations of other previously popular activation functions, such as Sigmoid and Tanh. It is a simple calculation that returns the value of the input, or 0, if the input value is less than 0. Thus, the function is by definition calculated as:

$$f(x) = \max(x, 0) \tag{17}$$

Because of its graph representation as we see in Figure 2.15 the ReLU function is also called ramp function.

Softmax Function

Softmax function is a mathematical function that converts a vector of numbers into a vector of probabilities. The standard softmax function is often used in the final layer of a neural network-based classifier and most commonly in multiclass classification problems. The output for each i_{th} value of the input vector is calculated by this function:

$$f(x_i) = \frac{e^{x_i}}{\sum_{n=1}^{\infty} e^{x_n}} \tag{18}$$

The function is depicted in the Figure 2.16.

Binary Step Function

Binary step function basically a threshold based classifier. If the input value is above a threshold, the output is set to 1, hence the neuron is activated, otherwise the output is set to 0 and the neuron is deactivated. The function is illustrated in Figure 2.17

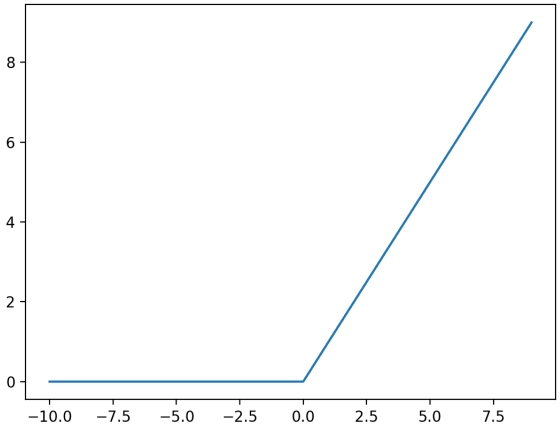


Figure 2.15: *ReLU Function*

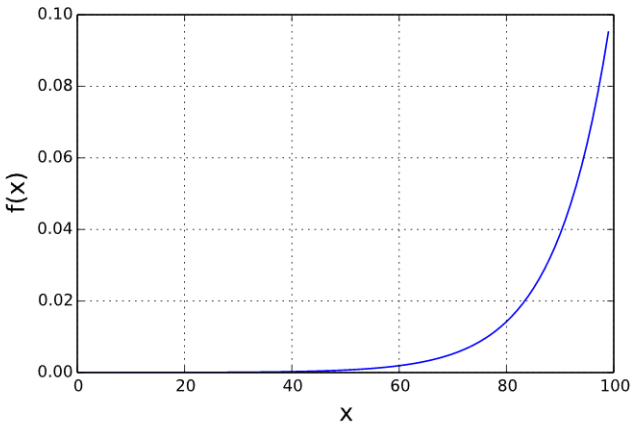


Figure 2.16: *Softmax Function*

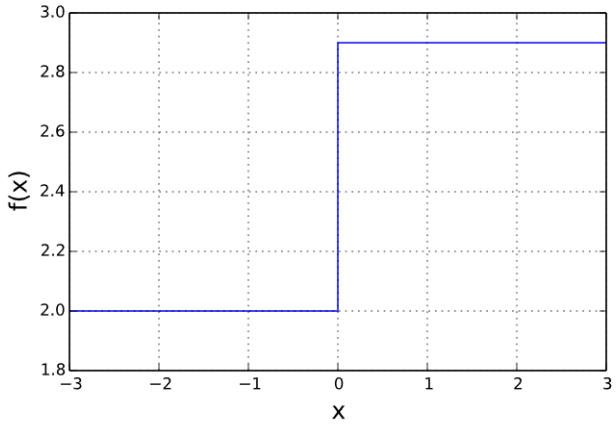


Figure 2.17: *Binary Step Function*

2.5.3 Back propagation

A explained before, each input is multiplied by their weights. Weights is an indication of the input's strength. After assigning the weights a bias variable is added as it is illustrated in the Figure 2.11. Consequently, the activation function is applied to the equation $Z = w_1 * i_1 + ... + w_n * i_n$, thus its a transformation that is applied to the input before sending it to the next layer of neurons. After passing through every hidden layer, we move to the last layer i.e our output layer which gives us the final output. This process is called forwarding Propagation. After the predictions from the output layer are produced, the error is calculated. If the error is not small enough Back Propagation is performed.

Back propagation [30] is a widely used algorithm for updating and finding the optimal values of weights or coefficients that minimize the error. The way in which the weights are updated depends in the optimizers, which are methods to change the attributes of a neural network. The back propagation algorithm computes the gradient of the loss function for a single weight by the chain rule. It efficiently computes one layer at a time, starting from the last layer and iterating backwards. This can be clearely observed in the following figure Figure 2.18

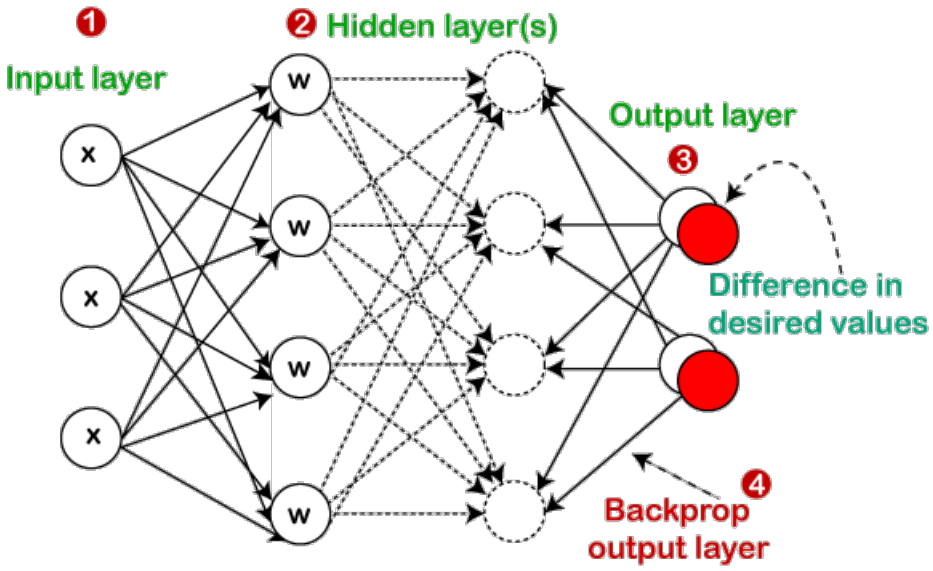


Figure 2.18: *How back propagation works*

Evaluation between s and y happens through a cost function. This can be as simple as MSE (mean squared error) or more complex like cross-entropy.

2.6 Convolutional neural networks

A convolutional neural network (CNN) is a type of artificial neural network used in image recognition and processing that is specifically designed to process pixel data.

The architecture of a convolutional network is analogous to that of the connectivity pattern of neurons in the human brain. CNN have their "neurons" arranged more like those of the frontal lobe, the area responsible for processing visual stimuli in humans and other animals. A convolutional network is capable of successfully capture the spatial and temporal dependencies in an input image by applying relevant filters. A typical CNN consists of an input layer, an output layer and a hidden layer that includes multiple convolutional layers, pooling layers, fully connected layers and normalization layers.

2.6.1 Convolutional layer

In the demonstration at Figure 2.19 we can see an input image that is convoluted in patches. Each patch is carying the convolution operation with the filter matrix and the output matrix will be the convolution of these two matrices. The filter (kernel) moves to the right with a certain Stride Value until it parses the complete width. As it reaches the end of the width it hops down to the beginning (left) of the image with the same Stride Value and repeats the process until the entire image is traversed. For instance, in our example is Stride value is 1, the kernel will shift 16 times and each time it will perform a matrix multiplication operation between K (kernel) and the portion P of the image over which the kernel is hovering.

In this project, we experimented in different depths of convolutional neural networks (up to 4 Convolutional layers) as analyzed in subsection 5.1.3. As input, the CNNs built received the correlation matrices that are calculated from the input dataset chapter 4. Correlation matrix is an N*N matrix that depicts the correlations of all the possible pairs of values in a table. In our case these values represent the regions of interest (ROIs) of the human brain section 3.3 and the correlation between them.

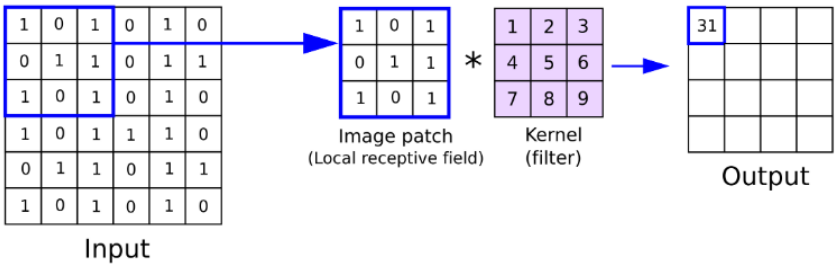


Figure 2.19: *Convolution of the input matrix using a filter*

2.6.2 Pooling layer

Similar to the Convolutional Layer, the Pooling layer is responsible for reducing the spatial size of the convolutioned feature. This is to decrease the computational power required to process the data through dimensionality reduction. Furthermore,

it is useful for extracting dominant features which are rotational and positional invariant, thus maintaining the process of effectively training of the model. There are two types of pooling: Max Pooling and Average Pooling. Max Pooling returns the maximum value from the portion of the image covered by the Kernel. On the other hand, Average Pooling returns the average of all the values from the portion of the image covered by the Kernel Figure 2.20. In the scope of the diploma thesis we used Max Pooling because it performs much better that Average Pooling and in addition, it also performs as a noise suppressant, thus performs denoising along with dimensionality reduction.

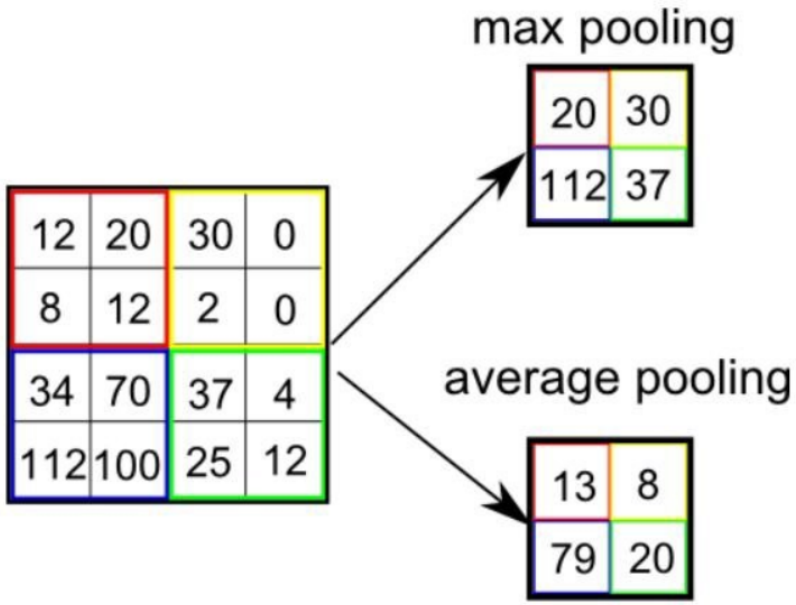


Figure 2.20: *Max and average pooling example*

The Convolutional Layer and the Pooling Layer, together form the i-th layer of a Convolutional Neural Network. Depending on the complexities in the images, the number of such layers may be increased for capturing low-levels details even further, but at the cost of more computational power.

2.6.3 Fully-connected layer

Adding a fully connected layer Figure 2.21 in the neural network is a cheap way for the model to learn non-linear combinations of the convoluted features. Before feeding the data into the fully connected layer, it needs to be flattened into a column vector. The flattened output is then fed to a feed-forward neural network and back-propagation applied to every iteration of training. After a series of epochs, the model is able to distinguish between dominating and low-level features in images in order to classify them using the Softmax Classification technique.

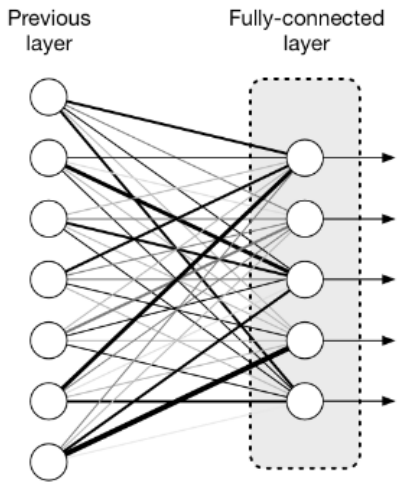


Figure 2.21: *Fully-connected layer*

2.6.4 LSTM layer

Long Short Term Memory Network is an advanced RNN (Recurrent Neural Network), a sequential network, that allows information to persist. It is capable of handling the vanishing gradient problem faced by RNN. A recurrent neural network is also known as RNN is used for persistent memory.

LSTM Architecture

The LSTM consists of three parts, as shown in the image below and each part performs an individual function. These three parts of an LSTM cell are known as gates. The first part is called Forget gate, the second part is known as the Input gate and the last one is the Output gate Figure 2.22. Just like a simple RNN, an LSTM also has a hidden state where H(t-1) represents the hidden state of the previous timestamp and H(t) is the hidden state of the current timestamp. In addition to that LSTM also have a cell state represented by C(t-1) and C(t) for previous and current timestamp respectively.

Forget Gate

In a cell of the LSTM network, the first step is to decide whether we should keep the information from the previous timestamp or forget it. The equation for the forget gate is:

$$f_t = \sigma(x_t * U_f + H_{t-1} * W_f) \tag{19}$$

Where,

- X_t: input to the current timestamp
- U_f: weight associated with the input
- H_{t-1}: The hidden state of the previous timestamp

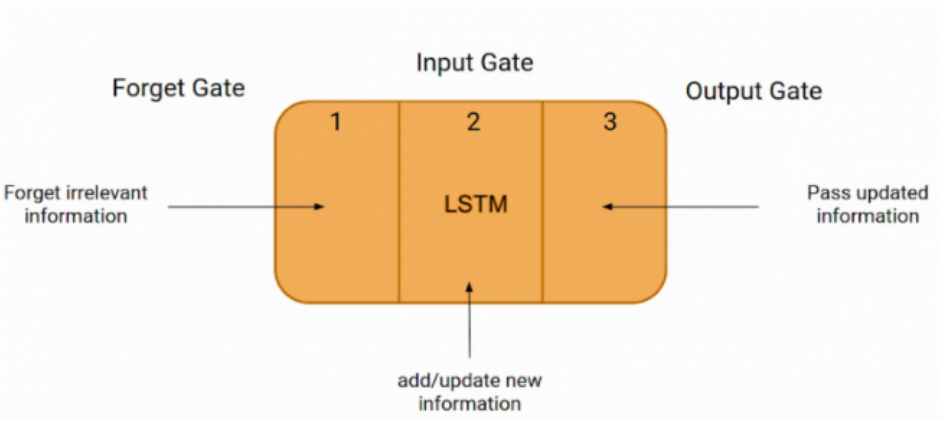


Figure 2.22: *LSTM architecture*

- W_f: It is the weight matrix associated with hidden state

Later, a sigmoid function is applied over it. That will make f_t a number between 0 and 1. This f_t is later multiplied with the cell state of the previous timestamp:

- $C_{t-1} * f_t = 0$ if $f_t = 0$ (forget everything)
- $C_{t-1} * f_t = C_{t-1}$ if $f_t = 1$ (forget nothing)

Input Gate

Input gate is used to quantify the importance of the new information carried by the input. Here is the equation of the input gate:

$$i_t = \sigma(x_t * U_i + H_{t-1} * W_i) \tag{20}$$

Again we have applied sigmoid function over it. As a result, the value of I at timestamp t will be between 0 and 1. Now the new information that needed to be passed to the cell state is a function of a hidden state at the previous timestamp t-1 and input x at timestamp t. The activation function here is tanh. Due to the tanh function, the value of new information will between -1 and 1. If the value is of Nt is negative the information is subtracted from the cell state and if the value is positive the information is added to the cell state at the current timestamp. So the final equation is:

$$C_t = (f_t * C_{t-1} + i_t * N_t) \tag{21}$$

Output Gate

The equation for the output gate is similar to the two previous gates:

$$o_t = \sigma(x_t * U_o + H_{t-1} * W_o) \tag{22}$$

Its value will also lie between 0 and 1 because of this sigmoid function. To calculate the current hidden state we will use O_t and tanh of the updated cell state.

$$H_t = o_t * tanh(C_t) \tag{23}$$

Finally the output of the current timestamp is just hidden state with the Softmax activation applied.

$$Output = SoftMax(H_t) \tag{24}$$

2.7 Graph convolutional networks

In contrary to the CNNs, that they use data in the eucledian space, Graph Convolutional Networks can also operate with data that can only be structured in a non-euclidean space and can only be represented by graphs. Some examples of non-euclidean structures are genetic data, social network data or biological networks data. The main difficulty for the machine learning in graphs is to find a way to embed in the model, the graph information.

A graph G is represented as G = (V,E), where V is the set of vertices and E the set of edges of the graph. $|V|$ = n and $|E|$ = m are the number of vertices and the number of edges respectively. Each $v_i \in V$ represents a vertex of the graph and each $e_ij = (v_i, v_j) \in E$ represents an edge from vertex v_i to vertex v_j. The adjacency matrix is a NxN matrix A with $A_ij = 1$ if $e_ij \in E$ and $A_ij = 0$ if $e_ij \notin E$. Graph can also contain vertex attributes X, where $X \in \mathbb{R}^{n*d}$ is the feature matrix of the vertices. In our case this feature matrix is the correlation matrix.

2.7.1 Graph Convolutional layer

On Euclidean domains, convolution is defined by taking the product of translated functions. But as we mentioned, GCNs do use non-euclidean data structures. Convolution on graphs are defined through the graph Fourier transform. The graph Fourier transform, on turn, is defined as the projection on the eigenvalues of the Laplacian. These are the "vibration modes" of the graph. As for traditional CNNs, a GCN consists of several convolutional and pooling layers for feature extraction, followed by the final fully-connected layers.

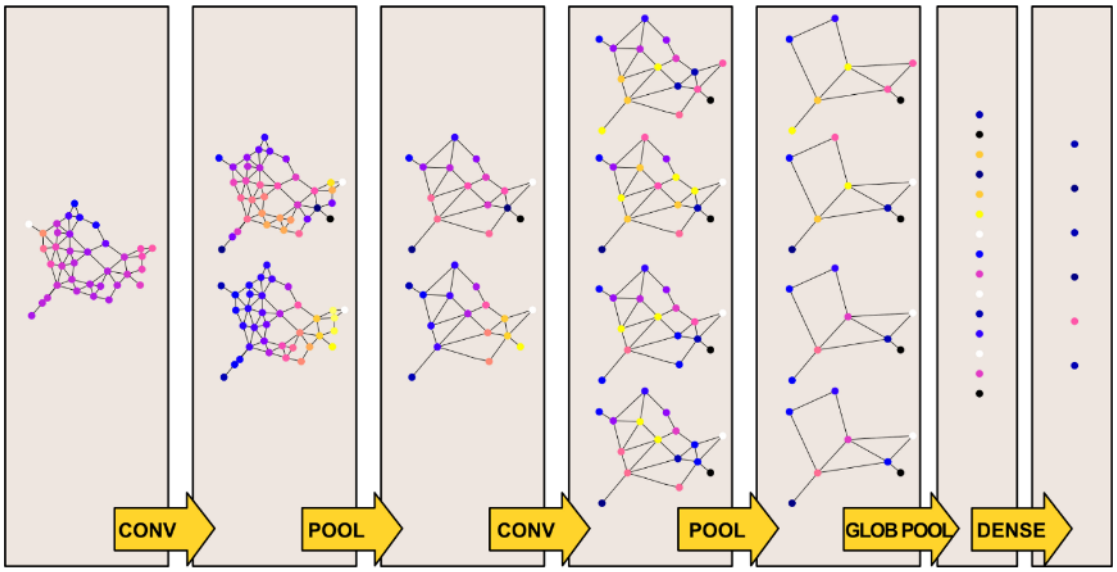


Figure 2.23: *Graph Convolutional layers*

Chapter 3

Schizophrenia and fMRI method

3.1 Symptoms of Schizophrenia

As mentioned before schizophrenia is a severe mental disorder whuch is characterized by a wide range of unusual behaviours. Its symptoms can be divided in two categories, positive and negative symptoms [31]. The former correspond to the presence of abnormal functions,for instance, delusions, and hallucinations. The latter, correspond to decreased functions, meaning the lack of ability to do something, i.e diminished emotional expression. Positive symptoms are more recognizable and generally respond better to medication. Negative symptoms are more subtle and less responsive to pharmacological treatment. Below, are listed some of the most common symptoms of SZ according to DSM-5 [32]:

- *Positive Symptoms*
    1. *Delusions*: belief or impression maintained despite being contradicted by reality or rational argument
    2. *Hallucinations*: an experience involving the apparent perception of something not present.
    3. *Disorganized thinking*: difficulty to keep track of thoughts, drift between unrelated ideas during speech.
    4. *Disorganized or abnormal movements*: difficulties to perform goal-directed tasks, catatonia (lack of movement and communication) or stereotypy (physical movements that are both aimless and repetitive).
- *Negative symptoms*
    1. *Diminished emotional expression*: lack of showing emotion, apathetic and unchanging facial expression.
    2. *Avolition*: lack of motivation, inaction.
    3. *Anhedonia*: diminished capacity to experience pleasant emotions.
    4. *Alogia*: poverty of speech.

5. *Asociality*: reduction in social initiative due to decreased interest in forming close relationships with others

3.2 Diagnosis of schizophrenia

Diagnosis of SZ is a challenging problem due to the heterogeneity of this mental disorder and lack of specific effective bio-markers [33]. In order for a patient to be diagnosed, three types of symptoms need to be evaluated which are the physical, the psychiatric and psychological symptoms. Clinical examination includes various tests such as blood tests as well as medical imaging [34],[35]. If no physical cause for the suspected SZ symptoms are found, the physicians may refer the patient to a psychologist or psychiatrist for psychological evaluation based on diagnostic and statistical manual of mental disorders (DSM-5)[32].

Early identification and treatment of schizophrenia is very important for the general mortality rate and for the life quality of the patient. Although there is not a single test to diagnose schizophrenia. As part of the examination the psychiatrist tries and unravels the changes in their behavior and biological functions (sleeplessness, lack of interest in eating or socializing). Information about the deviations in the patient's behavior is also collected from the family or caregivers. It need at least a month of observation from specialist doctors for a patient to be diagnosed as the symptoms are several and can be caused by other mental disorders, like addiction, depression or bipolar disorder.

3.3 The fMRI imaging method

Magnetic resonance imaging is a medical imaging technique used in radiology to form images depicting anatomy. With various sequences, MRI may provide insight of physiological processes of the body. Overall, MRI based structural neuroimaging modalities are suitable for visualizing white matter (WM), gray matter (GM), and cerebrospinal fluid (CSF) tissues of the brain as well as exploring their abnormalities. Furthermore, past studies have discovered that schizophrenia affects the temporal and anterior lobes of hippocampus regions of the brain. In addition, patients with SZ can display increased volume of CSF and decreased volume of white and gray matter. As a result, besides blood tests (as mentioned above), functional and structural neuroimaging techniques are another important category of methods that are able to diagnose SZ.

Functional magnetic resonance imaging (fMRI) detects changing in blood flow in the brain. It is not structural and the difference from MRI techniques is that it has a time parameter as well. When a brain region is activated it consumes more oxygen , thus the blood flow in this area is increasing. Fmri depicts those changes by changing the color of the voxels in the area that is activates as we can see in Figure 3.1.

As an imaging method fMRI has some advantages:

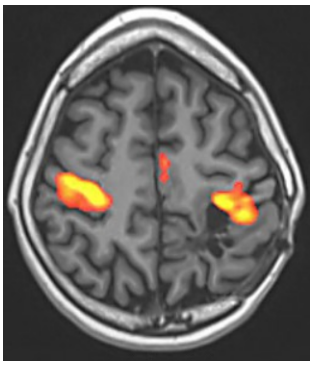


Figure 3.1: *Activation regions in fMRI*

- It does not use radiation (like X-rays,computed tomography (CT) and positron emission tomography (PET) scans, so it is completely safe for the subject.
- If done correctly, fMRI has virtually no risks. It can evaluate brain function safely, noninvasively and effectively.
- fMRI is easy to use, and the images it produces are very high resolution (as detailed as 1 millimeter)

The basic structural unit of a 3D fMRI image is a voxel. A voxel is a unit of graphic information that defines a point in three-dimensional space. A voxel size can change based on the part of the brain that is being examined. Studies examining the whole brain use bigger voxels whilst studies that need a small part of the brain used smaller voxel size. A voxel can contain millions of neurons and billions of neural synapses.

Regions of interest (ROIs) is a groups of neighboring voxels, that are activated in a similar way (have great correlation), so we can examine them like a biggest region of the brain.

Chapter 4

Dataset

In this chapter the COBRE database is referred [27] and its contexts are described analytically. We will explain how the data was obtained and what pre-processing steps were made in order to be used in the right form as inputs in the models.

4.1 COBRE database

The Center for Biomedical Research Excellence (COBRE) is contributing raw functional MR data from 72 patients with Schizophrenia and 75 healthy controls (ages ranging from 18 to 65 in each group). All subjects were screened and excluded if they had; history of neurological disorder, history of mental retardation, history of severe head trauma with more than 5 minutes loss of consciousness, history of substance abuse or dependence within the last 12 months. Diagnostic information was collected using the Structured Clinical Interview used for DSM Disorders (SCID).

A multi-echo MPRAGE (MEMPR) sequence was used with the following parameters: TR/TE/TI = 2530/[1.64, 3.5, 5.36, 7.22, 9.08]/900 ms, flip angle = 7°, FOV = 256x256 mm, Slab thickness = 176 mm, Matrix = 256x256x176, Voxel size =1x1x1 mm, Number of echos = 5, Pixel bandwidth =650 Hz, Total scan time = 6 min. With 5 echoes, the TR, TI and time to encode partitions for the MEMPR are similar to that of a conventional MPRAGE, resulting in similar GM/WM/CSF contrast. Rest data was collected with single-shot full k-space echo-planar imaging (EPI) with ramp sampling correction using the intercomissural line (AC-PC) as a reference (TR: 2 s, TE: 29 ms, matrix size: 64x64, 32 slices, voxel size: 3x3x4 mm3). In addition to that, phenotypic information are available regarding age, sex etc.

We used a preprocessed version of the dataset [27]. Each fMRI dataset was corrected for inter-slice difference in acquisition time and the parameters of a rigid-body motion were estimated for each time frame. Rigid-body motion was estimated within as well as between runs, using the median volume of the first run as a target. The median volume of one selected fMRI run for each subject was co-registered with a T1 individual scan using Minctracc [36], which was itself non-linearly transformed to the Montreal Neurological Institute (MNI) template [37] using the CIVET pipeline.

4.2 Data preprocessing

Although the data is preprocessed, there are still many steps needed for the data to be in an appropriate form for input in our models. First of all as mentioned in the previous section section 4.1 rigid-body motion was estimated during the subject's scan. Subject movements can affect the clarity of the signal due to noise generation. The noise is random, unwanted variation or fluctuation that interferes with the signal, making it difficult to extract meaningful information unless its removed. Apart from the subject movements there are other factors generating noise to the signal, such as experiment related errors, for instance a hardware error in the scan. All these unwanted factors contributing to the noise generation, are estimated and during the data preprocessing and are saved in external files for later removal. They are called confounds, and they are .tsv files along with the fMRI files in the dataset.

Consequently, the NifTi files containing the fMRI scans for each subject have to be transformed into correlation matrices to be fed in the models. A correlation matrix is a table which displays the correlation coefficients for different variables. The matrix depicts the correlation between all the possible pairs of values in a table. It is a powerful tool to summarize a large dataset and to identify and visualize patterns in the given data. Following, the preprocessing steps for transforming the NifTi files to correlation matrices are presented.

1. **Mstl probablistic atlas:** Contains a predetermined set of coordinates, so as to isolate specific regions of the brain called Regions of Interest (ROIs). Regions of interest are a group of adjacent voxels, which are activated in a similar way (have high correlation), so can be examined as as a whole area.

2. **Masker:** Masker is used to apply the atlas fetched at step 1 in the image data.

3. **Extracting time series:** This is an imortant step. In this step, the confound files are used for the confounds extraction from the signal. A functional connectome is created, which is a set of connections that represent the correlations between the ROIs.

4. **Correlation matrices:** Correlation matrices are calculated using the time series from step 3. To accent the importance of the confounds extraction, two images are compared. One illustrating the correlation matrix derived from the raw time series Figure 4.1 and one depicting the correlation matrix without the noise caused by the confounds as they are removed from the signal at step 3 Figure 4.2.

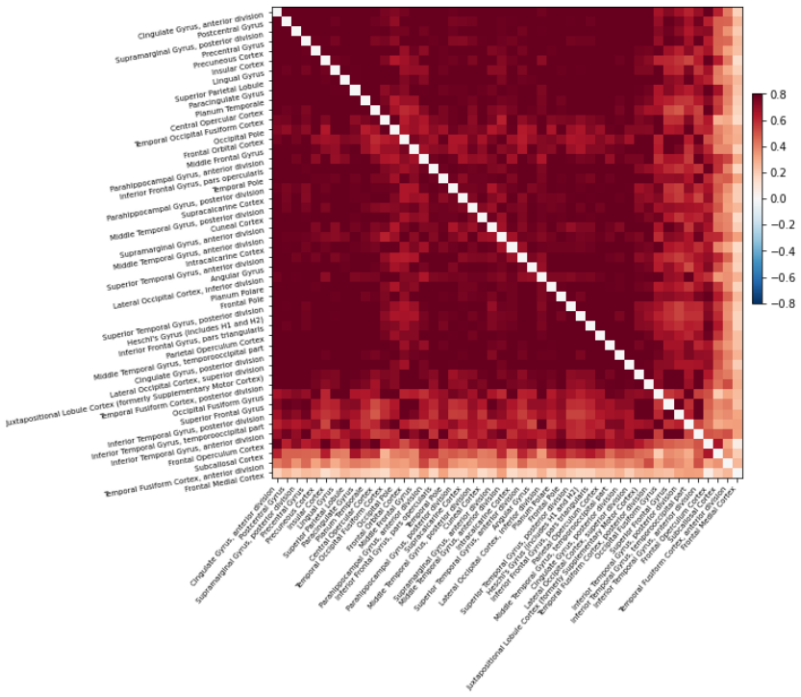


Figure 4.1: *Correlation matrix from raw time series*

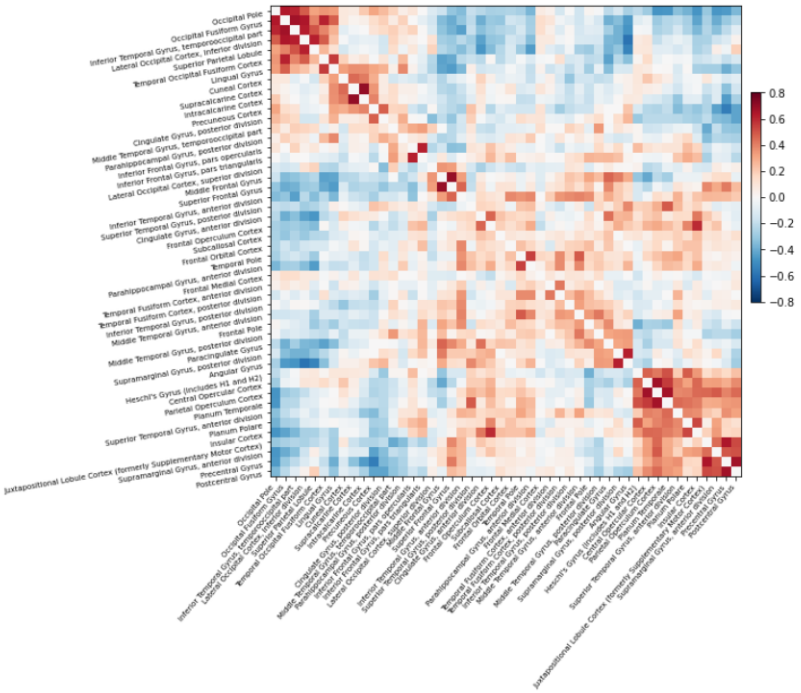


Figure 4.2: *Correlation matrix after removing confounds from time series*

Chapter 5

Experiments

In this chapter, we discuss briefly the experiments and testing conducted in the scope of the diploma thesis. In addition, difficulties and challenges that we faced during the project are mentioned and of course what are the conclusions and what I learned by the results of these experiments.

5.1 Experimental procedure

First of all,the experiments are based variety of classifiers used for the classification and deep learning as well. Hence, we will discuss the difference between the different model used for the schizophrenia classification, the accuracy that each one of them can offer and how appropriate each model is for the data used in the thesis. Every experiment is conducted three times and the final accuracy presented is the mean value of the accuracies of the three experiments.

5.1.1 Accuracy and loss metrics

Before the models comparison, a demonstration of the accuracy behaviour during the epochs follows. For this purpose, results were extracted from the CNN model. The behaviour of accuracy and loss is similar for all the models created in the scope of the diploma thesis.

In the Figure 5.1 it is observed over the epochs that the train accuracy is increasing as well as the validation accuracy. The training set is used to train the model while the validation set is only used to evaluate the model's performance. The validation accuracy calculated on the data set is not used for training, but is used (during the training process) for validating (or "testing") the generalisation ability of the model. The validation accuracy increase indicated that the model is learning after a number of epochs end that it extracts knowledge from the training set.

Another metrics worth examined are the training and validation loss. The training loss is a metric that is used to assess how a deep learning model fits the training data. That is to say, it estimates the error of the model on the training set. On the

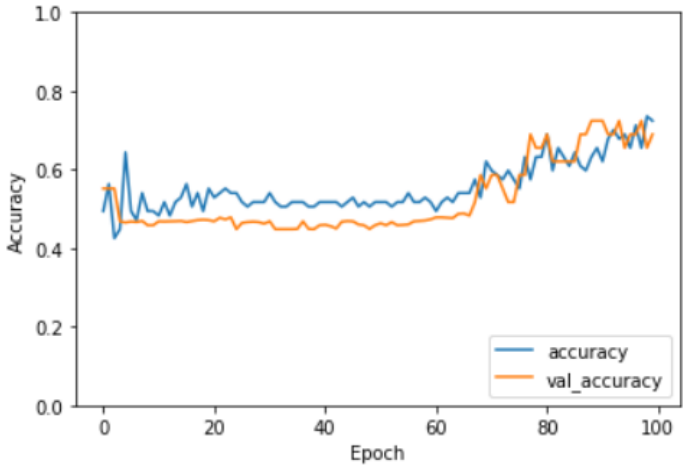


Figure 5.1: *Train and validation accuracy changes through epochs(100)*

other hand, validation loss is a metric used to evaluate the performance of a deep learning model on the validation set. The validation loss is similar to the training loss and is calculated from a sum of the errors for each example in the validation set. Following is an illustration of the training and validation loss behaviour during the epochs (Figure 5.2).

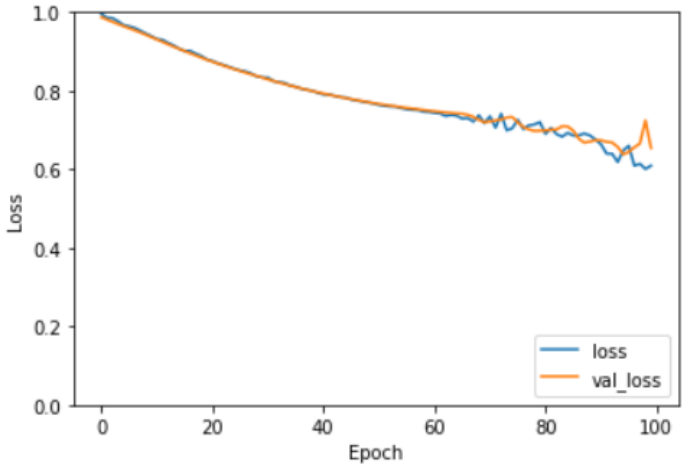


Figure 5.2: *Train and validation loss changes through epochs (100)*

It is clear from the Figure 5.2 that both training and validation loss are decreasing through the epochs indicating that the error is decreasing,as the loss represents the difference between the right answer. After a number of epochs train and validation loss begin to stabilize around a value. This is better depicted in Figure 5.3 which represents the same experiment for 150 epochs.

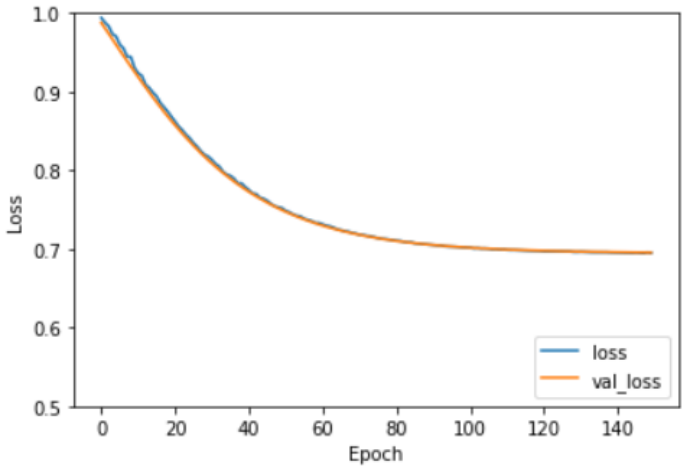


Figure 5.3: *Train and validation loss changes through epochs (150)*

5.1.2 Batch size

Batch size is referred to as the number of training examples utilized in one iteration, in other words the amount of data included in each sub-epoch weight change. There are a lot of options in parametrizing the batch size, as the full batch where the batch size is equal to the number of samples, thus making the iteration and epoch values equivalent. In the experimental procedure for the diploma thesis,both mini-batch learning approach and batch-mode approaches are followed. The mini-batch size values usually tested are 34,64,128,256 and 512. The current dataset has 146 samples and the training set after the train, validation and test split has 92 samples. There is no point in setting the batch size in a value bigger than the number of samples, so the values we will examine are 32,64 and 92.

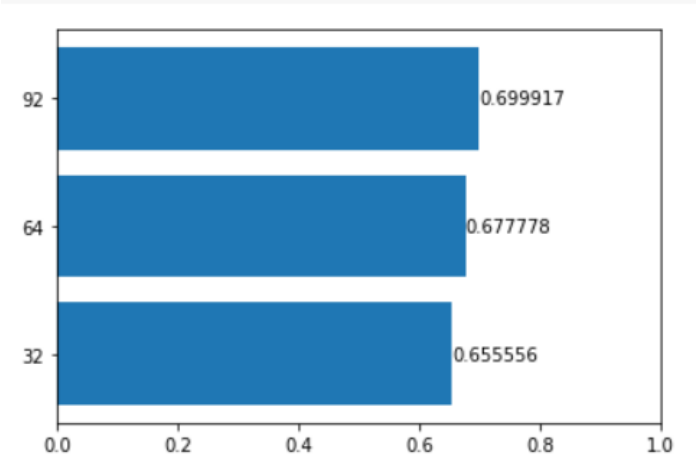


Figure 5.4: *Accuracy based on batch size for baseline model*

As it can be observed by the Figure 5.4 the differences in accuracy are not that broad, with the batch size of 92 appears to be the optimal batch size that fits our

data. In the baseline model examined, a method called early stopping is used in order to prevent overfitting. This means that we can set an arbitrary large number of epochs and the training will stop once the model performance stops improving. With the batch epochs experiment, it is clear that the smallest the batch size is the earlier the model overfits.By conducting each experiment 3 times to calculate the mean of the accuracy, and with the early stopping method being used, we observed that for smallest batch sizes the model stopped improving earlier.That is rational as the smallest the batch size is, the more batches can fit to an epoch, leading to more weight updates. In our case, where the dataset is somewhat small (92 samples in the training set as mentioned earlier), the model performs better in the batch-mode learning approach, which means one batch per epoch as the batch size is equal to the number of samples. This is well demonstrated in Figure 5.4.

5.1.3 CNN model complexity

Continuing with the CNN model, another contributing factor to the performance of the model is the complexity of it. A Convolutional Neural Network typically consist of three basic layers, a convolutional layer, a pooling layer, and a fully connected layer, as analytically explained in chapter 2.. As a result there is an abundance of different architectures of a CNN that can be built using these layers. A highly complex model can lead to overfitting as it quickly memorizes the training dataset patterns, whilst a very simple one can have the opposite result as it may fail to recognize meaningful patterns in the data.

Baseline-CNN model

A baseline model will establish a minimum model performance to which all of our other models can be compared, as well as a model architecture that we can use as the basis of study and improvement. It is a very simple sequence of the basic layers and it is the base upon which a more complex model can be built. The architecture of our baseline model involves stacking convolutional layers with small 3×3 filters followed by a max pooling layer. This block is repeated two times before the fully connected level that gives the output of the model, meaning that the model two convolutional layers . It is the simplest form of a CNN with 2 convolutional layers, so the number of training parameters are not that high (23,426).

CNN-2 Convolutional layers

For the next experiment, we add a convolutional in our baseline model to increase the complexity of the model. In general,by increasing the complexity of the model, the number of training parameters is increasing as well (29,186). To be more precise, adding layers to the model increases the number of weights in the model, and helps us extract more features.

CNN-3 Convolutional layers

For the next experiment, we add an extra convolutional in our previous model to increase the complexity of the model.As we mentioned earlier,by increasing the

complexity of the model, the number of training parameters is increasing as well (96,770)

CNN-4 Convolutional layers

Adding one more convolutional layer in the model will lead us to the deepest CNN for the scope of the experiments. Note that as we add layers, in parallel we increase the size of each layer (32,64,128,128 in the current experiment) and that is because as we move forward in the layers, the patterns get more complex; hence there are larger combinations of patterns to capture and the goal is to capture as many combinations as possible. Again the training parameters increase (241,282) compared to the previous architecture with 2 hidden layers.

Results

In Figure 5.5 we can observe that the optimum architecture for our data is the CNN with 3 convolutional layers. It is better that the baseline model (that only has 2 convolutional layer) because we increased the complexity and as mentioned earlier that improved the accuracy as the model was able to extract more features and gain more insights from the data that was fed to it. Although there is a limit in the model complexity and it is highly dependant of the dataset used, and mainly the size of it. In this case the number of data samples is small, hence instead of extracting more features the model starts to overfit. Although the accuracy is better in the training set, in the test set becomes poor as clearly depicted in Figure 5.5.

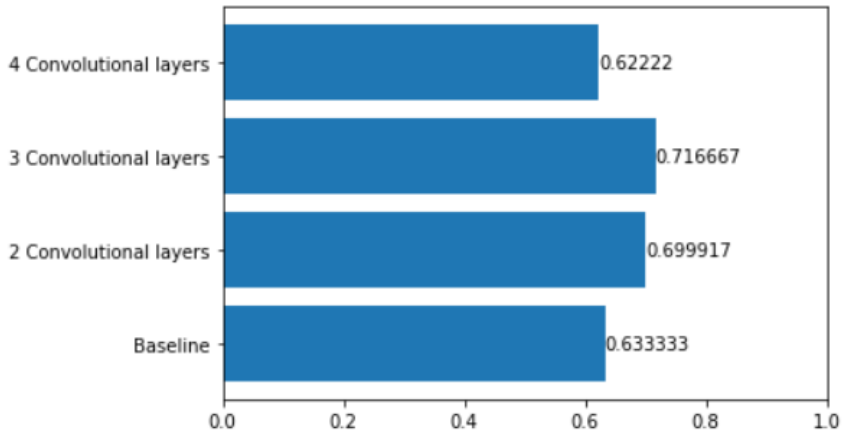


Figure 5.5: *Accuracy for 3 types of CNN architectures*

5.1.4 Loss function

Another experiment we conducted regarding the CNN models is about the loss function. The performance of the different architectures builded is tested throughout twpo different loss functions. Since our problem is a binary classification problem we cannot plenty of loss functions. We chose the 2 most basic for the experiments

in order to decide which is the best for our model. As depicted in Figure 5.6 the Binary Cross Entropy loss is the one that gives better performance in most of the CNN architectures. To be more precise for the deepest architectures as in baseline model the hinge loss give slightly higher performance. In Figure 5.7 it is clear that regardless the loss function, the 3 Convolutional layers architecture gives the optimal results for our data.

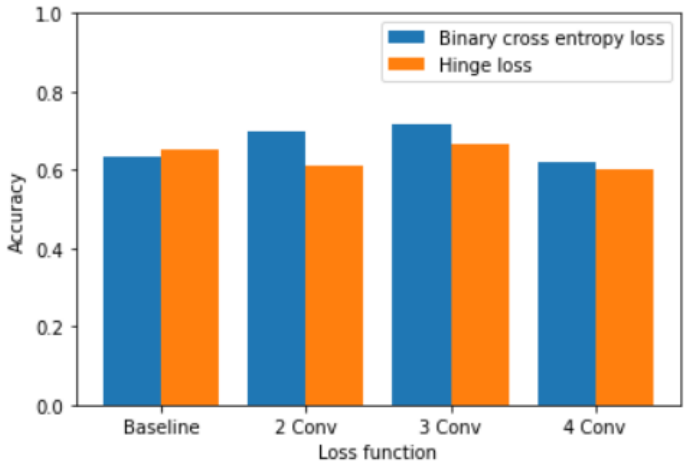


Figure 5.6: *Loss function for each CNN architecture*

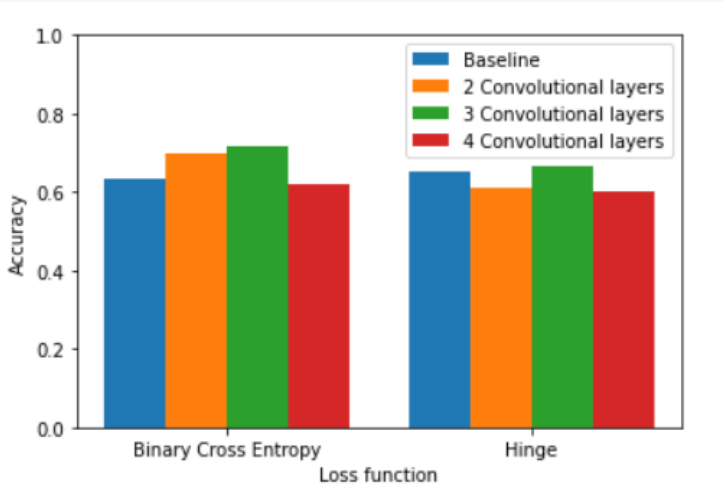


Figure 5.7: *CNN architectures for each loss function*

5.1.5 Model comparison

In this section of experiments, we will compare the accuracy of different models and classifiers that were created in order to find the most suitable one to classify

schizophrenia for our dataset. Some basic classifiers were tested, as well as models and classifiers commonly used in the bibliography such as SVM and LSTM. Once again the accuracy illustrated for each model is the mean of the accuracies calculated in three consecutive experiments for each model.

CNN-LSTM model

Another model we experimented on is the hybrid CNN-LSTM model. This model consists of input layer, convolutional layer, pooling layer, flatten layer, LSTM layer and fully connected layer. This is the most basic form of the CNN-LSTM model and it is the one we used in the scope of the diploma thesis. LSTM is a special kind of recurrent neural network that is capable of learning long term dependencies in data, as explained in chapter 2.

SVM

In the scope of the experiments,all four possible kernels are used, meaning polynomial,rbf,linear and sigmoid. All kernels manage to classify the data satisfyingly with an accuracy variating from 68% to 73%. In Figure 5.12 we use the SVM with rbf kernel accuracy as in the experiments it tends to have slightly higher accuracy than the others.

A Grid Search was performed for all the models in order to determine the best combination. First of all, we conducted a Grid Search algorithm to find out what is the most suitable kernel for our data, and with what parameters. Hence, the parameters of the Grid Search was:

- **C**:It is the penalty parameter, which represents misclassification or error term. The misclassification or error term tells the SVM optimisation how much error is bearable. This is how we can control the trade-off between decision boundary and misclassification term.
- **gamma**: It defines how far influences the calculation of plausible line of separation. when gamma is higher, nearby points will have high influence; low gamma means far away points also be considered to get the decision boundary.
- **kernels**: It indicates the kernel to be used in the SVM model either it is polynomial, rbf, linear or sigmoid kernel. The theory for each one of them can be found in chapter 2

After the grid search it was clear that the RBF kernel is the optimal for the dataset. In Figure 5.8 it is clear that the optimal values for the RBF kernel are, C=100 and gamma=0.01. We can observe that for highest values of C the performance of the model is better, which means that for our data the trade-off between decision boundary and miss-classsification error tends to be in favor of miss-classification error, as the model has better accuracy if it prefers to include more outliers in the decision boundary that having a maximum margin line. In addition, we can observe that the model performs better if not only the close neighbor influence the result but also the most distinct ones.

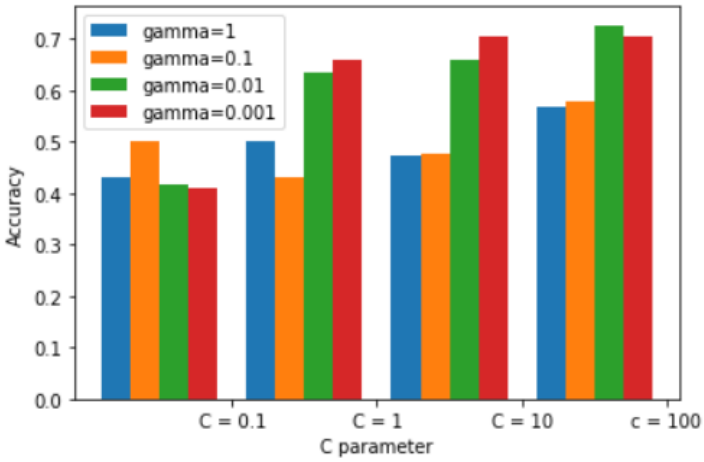


Figure 5.8: *Optimal parameters for SVM with RBF kernel*

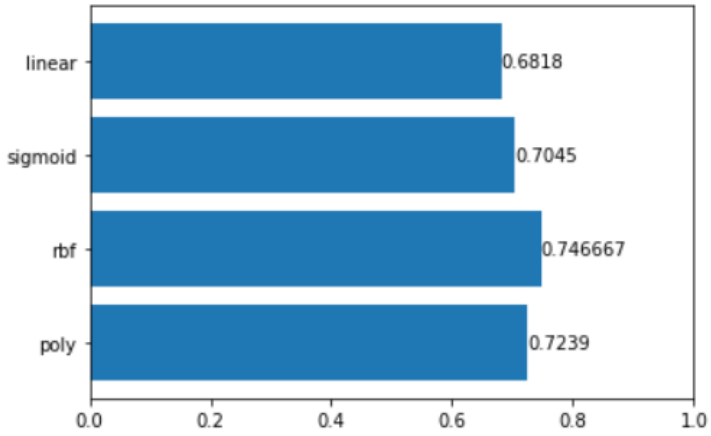


Figure 5.9: *SVM accuracy based on different kernels*

Logistic regression As the problem to be solved is binary, the option for the multiclass parameter is the 'ovr' (one-vs-rest). For the solver parameter, which is the algorithm to be used in the optimization problem, the 'liblinear' solver is used to calculate the accuracy in the experiments as it gave the better performance for our data comparing to the other solvers. For the regularization L2 was used because it is compatible with all the solvers and was easier for the experiments. Liblinear solver is the most suitable for our dataset as we can see in Figure 5.10, because it is designed for small datasets, whilst 'sag' and 'saga' are better for large datasets.

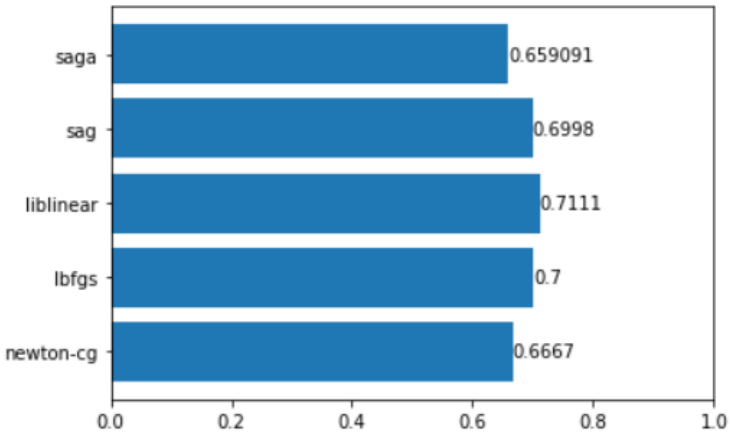


Figure 5.10: *Logistic regression accuracy for different solvers*

kNN

In KNN, finding the value of k is not easy. A small value of k means that noise will have a higher influence on the result and a large value make it computationally expensive. A lot of k values were tested and the result was that the optimum value of k is around 34 which is the square root of the number of training samples (102 - 70% of the total samples of 146) . In Figure 5.11 we can see that indeed the optimal value for k is 33 whilst the others give slightly lower accuracy.

Results

As we can see there are not extreme differences in our models accuracy. It seems that the SVM is the most suitable for our data confirming the high presence of this method used in bibliography for fMRI classification problems. The CNN model also performs very well as well as the optimised kNN and logistic regression classifiers. The hybrid CNN-LSTM model also performed satisfactory well, although it is reasonable that CNN have better performance as LSTM is usually used to process and make predictions given sequences of data (like timeseries), whilst CNN is designed to exploit "spatial correlation" in data,so works well on medical images data.

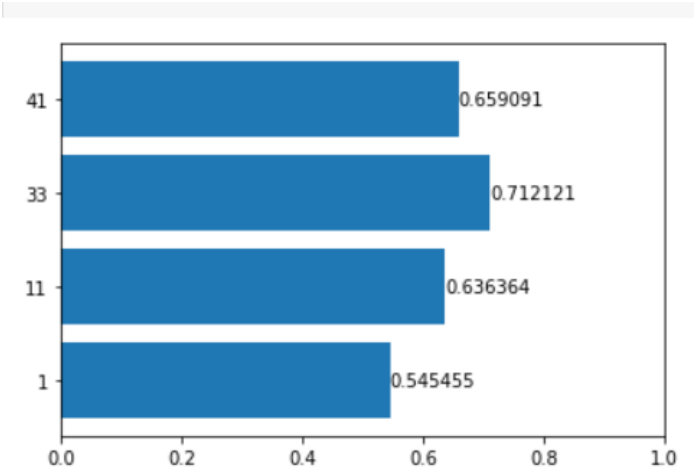


Figure 5.11: *Accuracy for significant values of k*

Although, accuracy is a good metric it is not enough. There are other metrics as well that are crucial for the model evaluation. One of them is recall metric, that is a metric that quantifies the number of correct positive predictions made out of all positive predictions that could have been made. In our case, it evaluates how many of the positive subjects (meaning the patients with schizophrenia) were classified correctly. As illustrated at Figure 5.13 kNN and cnn-lstm algorithms although with high accuracies they do not have that high performance in the recall metric.The other three models on the other hand are doing fine with the recall metric along with accuracy, with CNN (with 3 Convolutional layers) having the highest recall.

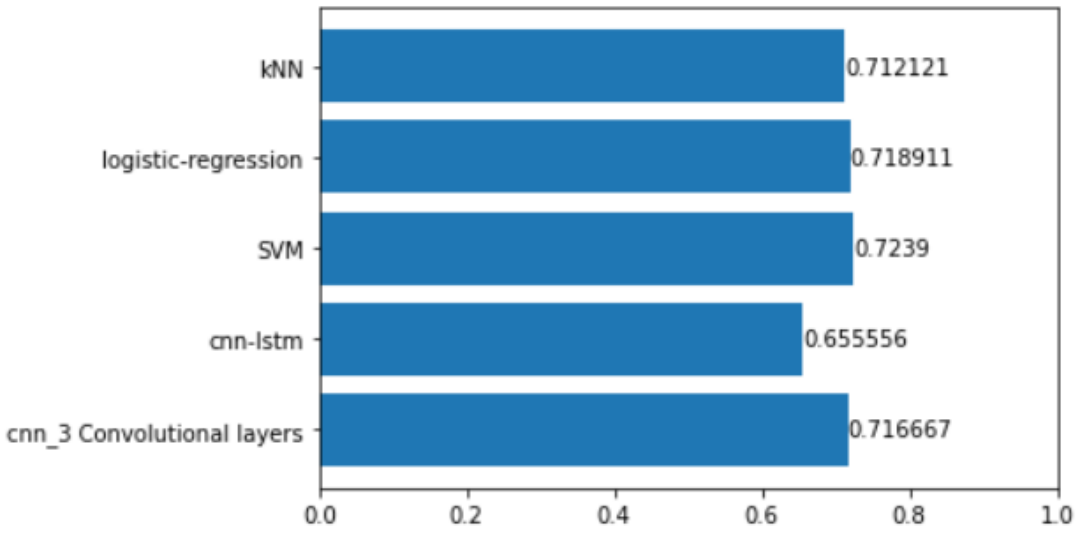


Figure 5.12: *Accuracy for different models*

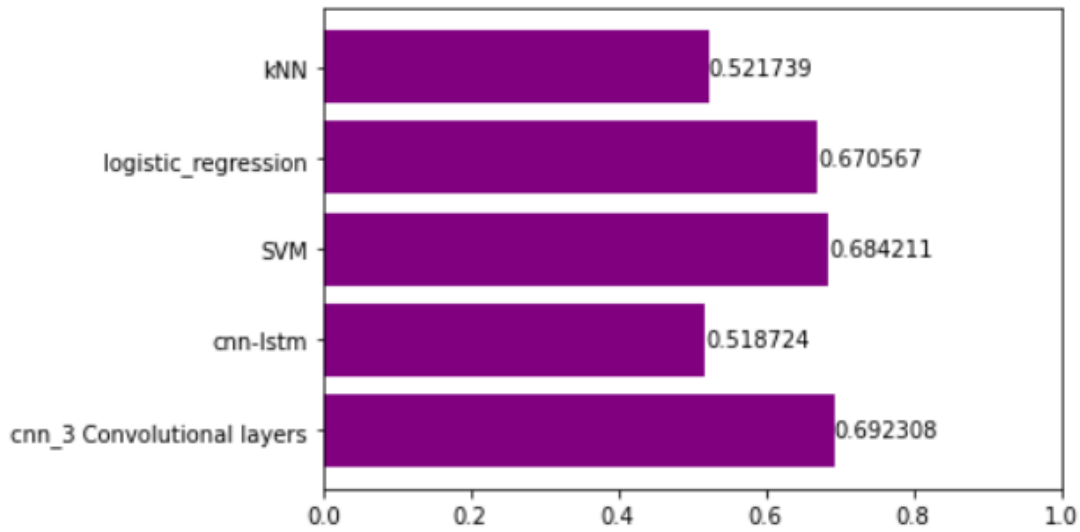


Figure 5.13: *Recall for different models*

5.2 Future work

In the scope of the diploma thesis a Graph Convolutional Network has been built. It is not in satisfactory level regarding the performance, hence there are no experiments for the GCN. However, the process of building the graph and the graph convolutional model are described above.

5.2.1 Graph structure

The model we used is the one proposed from Parisot in [38]. We are modelizing the population of the dataset into a graph, where every subject is represented by a vertex. Each vertex corresponds to a feature vector that has been extracted from the fMRI data. The feature vector for each vertex is the flattened upper triangle of the correlation matrix that corresponds to the subject that represents the vertex. A graph is a great way for embedding in our input data more information about each subject, thus phenotype data like age and gender. This data is represented as weight on the edges, that indicates the similarity between the two vertices that this edge connects.

For the diagnosis of schizophrenia, we are solving the node classification problem. This is a form of semi-supervised learning as only a subset of the vertices are labeled. The goal is to give each vertex a lablel $l \in (0, 1)$, where of course l describes the subject's situation with l=0 meaning that its a healthy control and l=1 that its a patient of schizophrenia. Node classification models aim to predict the labels in the unlabeles nodes based on the labels in the labeled nodes.

5.2.2 Graph edges

A crucial part of the graph structure is the correct definition of the edges and the weights. The goal is for the edges to depict the similarities between the vertices as accurately as possible. Similarly to pixel neighbourhood systems, the graph structure provides a broader field of view, filtering the value of a feature with respect to its neighbours' instead of treating each feature individually.

For this project, the phenotype data used are the age and the gender.The gender was chosen because several studies indicate that the incidence of schizophrenia is higher in men [39]. A significantly increased risk of schizophrenia was associated with paternal age younger than 20 years and with all age groups 40 years or older [40].

The weigh of an edge between vertex v and u is defined as follows:

$$W(v,u) = Sim(x(v),x(u))\sum_{h=0}^{H}\gamma(M_h(u),M_h(v)) \tag{25}$$

where:

- Sim(x(v),x(u)) is a measure of similarity between subjects, increasing the edge weights between the most similar graph nodes
- H number of phenotypic data used for the weight calculation. In our case H = 2
- e.g $M_{age} = 25$
- γ is a measure of distance between phenotypic measures. For categorical information such as subject's sex, we define γ as the Kronecker delta function δ, meaning that the edge weight between subjects is increased if e.g. they have the same sex.

$$\gamma(M_h(u),M_h(v)) = \begin{cases} 1 & ,if|M_h(u) - M_h(v)| < \theta \\ 0 & ,else \end{cases}$$

Finally, we define the similarity measure as follows:

$$Sim(x(v),x(u)) = e^{-\frac{[\rho(x(v),x(u))]^2}{2\sigma^2}} \tag{26}$$

where ρ is the correlation distance and σ determines the width of the kernel. The idea behind this similarity measure is that subjects belonging to the same class (healthy or Schizophrenia) tend to have more similar networks (larger Sim values) than subjects from different classes.

5.2.3 GCN-architecture

Our model architecture is illustrated in Figure 5.14. The model consists of a fully convolutional GCN with L hidden layers activated using the Rectified Linear Unit (ReLU) function. The output layer is followed by a softmax activation function. The graph is trained using the whole population graph as input. In addition we use a cross entropy loss function for the optimisation process. After training the GCN model, the softmax activations are computed on the test set, and the unlabelled nodes are assigned the labels maximising the softmax output.

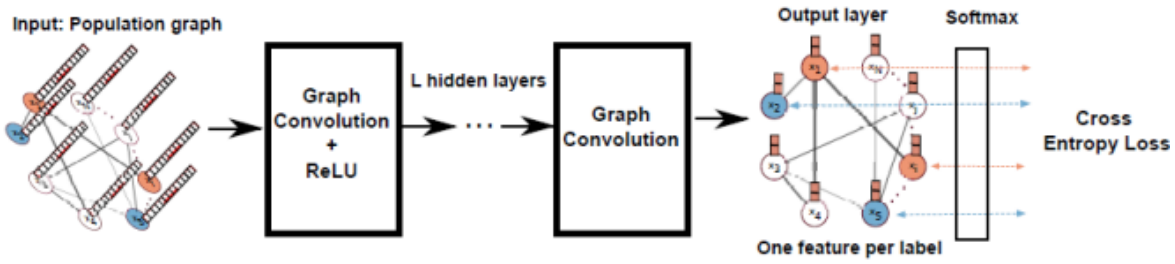


Figure 5.14: *Graph Convolutional Network*

Κατάλογος Σχημάτων

1 Απλή λογιστική παλινδρόμηση . . . 17
2 Πιθανά υπερεπίπεδα . . . 18
3 Βέλτιστο υπερεπίπεδο . . . 18
4 Διανύσματα υποστήριξης . . . 19
5 Προκατάλειψη-Διακύμανση . . . 20
6 Σιγμοειδής συνάρτηση . . . 23
7 Συνάρτηση tanh . . . 23
8 Συνάρτηση ReLU . . . 24
9 Συνάρτηση Softmax . . . 25
10 Συνάρτηση Binary step . . . 25
11 Συνέλιξη του πίνακα εισόδου με χρήση φίλτρου . . . 26
12 Παράδειγμα Max και Average Pooling . . . 27
13 Πλήρες επίπεδο . . . 28
14 Αρχιτεκτονική . . . 28
15 Συνελικτικό επίπεδο γράφου . . . 29
16 Περιοχές ενεργοποίησης στο fMRI . . . 30
17 Πίνακας συσχέτισης από ακατέργαστες χρονοσειρές . . . 31
18 Πίνακας συσχέτισης μετά την αφαίρεση των συγχύσεων από τις χρονοσειρές . . . 32
19 Η ακρίβεια της εκπαίδευσης και της επικύρωσης αλλάζει κατά τη διάρκεια των εποχών (100) . . . 32
20 Η συνάρτηση απώλειας της εκπαίδευσης και της επικύρωσης αλλάζει κατά τη διάρκεια των εποχών (100) . . . 33
21 Ακρίβεια για 3 τύπους αρχιτεκτονικών CNN . . . 34
22 Συνάρτηση απώλειας για κάθε αρχιτεκτονική CNN . . . 35
23 Ακρίβεια αρχιτεκτονικής CNN για κάθς συνάρτηση απώλειας . . . 35
24 Ακρίβεια για διαφορετικά μοντέλα . . . 36
25 Μετρική recall για διαφορετικά μοντέλα . . . 37
26 Συνελικτικά δίκτυα γράφων . . . 37

List of Figures

2.1 Simple linear regression . . . 44
2.2 Logistic function . . . 45
2.3 Hessian matrix . . . 46
2.4 A function and its tangent . . . 46
2.5 Parabola . . . 47
2.6 Support vector machine possible hyperplanes . . . 48
2.7 Support vector machine optimal plane . . . 49
2.8 Support vectors . . . 49
2.9 Bias-variance trade-off . . . 52
2.10 Biological neurons synapse. Image source https://pulpbits.net . . . 54
2.11 Artificial neuron. Image source: https://www.gabormelli.com/RKB/HomePage . . . 55
2.12 Artificial Neural Network. Image source: https://www.datasciencecentral.com . . . 55
2.13 Sigmoid function . . . 56
2.14 Tanh/Hyperbolic Tangent function . . . 57
2.15 ReLU Function . . . 58
2.16 Softmax Function . . . 58
2.17 Binary Step Function . . . 58
2.18 How back propagation works . . . 59
2.19 Convolution of the input matrix using a filter . . . 60
2.20 Max and average pooling example . . . 61
2.21 Fully-connected layer . . . 62
2.22 LSTM architecture . . . 63
2.23 Graph Convolutional layers . . . 65

3.1 Activation regions in fMRI . . . 68

4.1 Correlation matrix from raw time series . . . 71
4.2 Correlation matrix after removing confounds from time series . . . 71

5.1 Train and validation accuracy changes through epochs(100) . . . 73
5.2 Train and validation loss changes through epochs (100) . . . 73
5.3 Train and validation loss changes through epochs (150) . . . 74
5.4 Accuracy based on batch size for baseline model . . . 74

5.5 Accuracy for 3 types of CNN architectures 76
5.6 Loss function for each CNN architecture 77
5.7 CNN architectures for each loss function 77
5.8 Optimal parameters for SVM with RBF kernel 79
5.9 SVM accuracy based on different kernels 79
5.10 Logistic regression accuracy for different solvers 80
5.11 Accuracy for significant values of k 81
5.12 Accuracy for different models 81
5.13 Recall for different models 82
5.14 Graph Convolutional Network 84

References

[1] J. W. Lai, C. K. E. Ang, U. R. Acharya, and K. H. Cheong, «Schizophrenia: A survey of artificial intelligence techniques applied to detection and classification», *International Journal of Environmental Research and Public Health*, vol. 18, no. 11, 2021, ISSN: 1660-4601. [Online]. Available: https://www.mdpi.com/1660-4601/18/11/6099.

[2] K. J. Gorgolewski, J. Durnez, and R. A. Poldrack, «Preprocessed consortium for neuropsychiatric phenomics dataset», *F1000Research*, vol. 6, 2017.

[3] I. El Naqa and M. J. Murphy, «What is machine learning?», in *Machine Learning in Radiation Oncology: Theory and Applications*, I. El Naqa, R. Li, and M. J. Murphy, Eds. Cham: Springer International Publishing, 2015, pp. 3–11, ISBN: 978-3-319-18305-3. DOI: 10.1007/978-3-319-18305-3_1. [Online]. Available: https://doi.org/10.1007/978-3-319-18305-3_1.

[4] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*, 4. Springer, 2006, vol. 4.

[5] B. Apolloni, A. Ghosh, F. Alpaslan, and S. Patnaik, *Machine learning and robot perception*. Springer Science & Business Media, 2005, vol. 7.

[6] S.-I. Ao, B. B. Rieger, and M. Amouzegar, *Machine learning and systems engineering*. Springer Science & Business Media, 2010, vol. 68.

[7] L. Györfi, G. Ottucsák, and H. Walk, *Machine learning for financial engineering, 2012*.

[8] Y. Gong and W. Xu, *Machine learning for multimedia content analysis*. Springer Science & Business Media, 2007, vol. 30.

[9] J. Yu and D. Tao, *Modern machine learning techniques and their applications in cartoon animation research*. John Wiley & Sons, 2013, vol. 4.

[10] A. Fielding, *Machine learning methods for ecological applications*. Springer Science & Business Media, 1999.

[11] S. Mitra, S. Datta, T. Perkins, and G. Michailidis, *Introduction to machine learning and bioinformatics*. CRC Press, 2008.

[12] Z. R. Yang, *Machine learning approaches to bioinformatics*. World scientific, 2010, vol. 4.

[13] T. J. Cleophas, A. H. Zwinderman, and H. I. Cleophas-Allers, *Machine learning in medicine*. Springer, 2013, vol. 9.

[14] J. D. Malley, K. G. Malley, and S. Pajevic, *Statistical learning for biomedical data*. Cambridge University Press, 2011.

[15] P. Cunningham, M. Cord, and S. J. Delany, «Supervised learning», in *Machine Learning Techniques for Multimedia: Case Studies on Organization and Retrieval*, M. Cord and P. Cunningham, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 21–49, ISBN: 978-3-540-75171-7. DOI: 10.1007/978-3-540-75171-7_2. [Online]. Available: https://doi.org/10.1007/978-3-540-75171-7_2.

[16] T. Hastie, R. Tibshirani, and J. Friedman, «Unsupervised learning», in *The elements of statistical learning*, Springer, 2009, pp. 485–585.

[17] K. Murphy, R. M. Birn, and P. A. Bandettini, «Resting-state fmri confounds and cleanup», *Neuroimage*, vol. 80, pp. 349–359, 2013.

[18] B. Biswal, F. Zerrin Yetkin, V. M. Haughton, and J. S. Hyde, «Functional connectivity in the motor cortex of resting human brain using echo-planar mri», *Magnetic resonance in medicine*, vol. 34, no. 4, pp. 537–541, 1995.

[19] H. Dehghan, H. Hassanpour, and A. A Pouyan, «Roi analysis using harvard-oxford atlas in alzheimer's disease diagnosis based on pca», *Iranian (Iranica) Journal of Energy & Environment*, vol. 3, no. 3, 2012.

[20] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to linear regression analysis*. John Wiley & Sons, 2021.

[21] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, «Support vector machines», *IEEE Intelligent Systems and their applications*, vol. 13, no. 4, pp. 18–28, 1998.

[22] S. Suthaharan, «Support vector machine», in *Machine learning models and algorithms for big data classification*, Springer, 2016, pp. 207–235.

[23] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2009, vol. 2.

[24] A. Kamilaris and F. X. Prenafeta-Boldú, «Deep learning in agriculture: A survey», *Computers and Electronics in Agriculture*, vol. 147, pp. 70–90, 2018, ISSN: 0168-1699. DOI: https://doi.org/10.1016/j.compag.2018.02.016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168169917308803.

[25] Y. LeCun, Y. Bengio, *et al.*, «Convolutional networks for images, speech, and time series», *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.

[26] S. J. Pan and Q. Yang, «A survey on transfer learning», *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[27] P. Bellec and P. Bellec, «COBRE preprocessed with NIAK 0.17 - lightweight release», Nov. 2016. DOI: 10.6084/m9.figshare.4197885.v1. [Online]. Available: https://figshare.com/articles/dataset/COBRE_preprocessed_with_NIAK_0_17_-_lightweight_release/4197885.

[28] D. G. Kleinbaum, K. Dietz, M. Gail, M. Klein, and M. Klein, *Logistic regression*. Springer, 2002.

[29] M. Schmidt, N. Le Roux, and F. Bach, «Minimizing finite sums with the stochastic average gradient», *Mathematical Programming*, vol. 162, no. 1, pp. 83–112, 2017.

[30] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, «Learning internal representations by error propagation», California Univ San Diego La Jolla Inst for Cognitive Science, Tech. Rep., 1985.

[31] A. Sims, *Symptoms in the mind: An introduction to descriptive psychopathology*. Bailliere Tindall Publishers, 1988.

[32] D. American Psychiatric Association, A. P. Association, *et al.*, *Diagnostic and statistical manual of mental disorders: DSM-5*. American psychiatric association Washington, DC, 2013, vol. 5.

[33] A. M. Shepherd, K. R. Laurens, S. L. Matheson, V. J. Carr, and M. J. Green, «Systematic meta-review and quality assessment of the structural brain alterations in schizophrenia», *Neuroscience & Biobehavioral Reviews*, vol. 36, no. 4, pp. 1342–1356, 2012, ISSN: 0149-7634. DOI: https://doi.org/10.1016/j.neubiorev.2011.12.015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0149763411002223.

[34] J. Tomasik, E. Schwarz, P. C. Guest, and S. Bahn, «Blood test for schizophrenia», *European archives of psychiatry and clinical neuroscience*, vol. 262, no. 2, pp. 79–83, 2012.

[35] M. E. Shenton, R. Kikinis, F. A. Jolesz, *et al.*, «Abnormalities of the left temporal lobe and thought disorder in schizophrenia: A quantitative magnetic resonance imaging study», *New England Journal of Medicine*, vol. 327, no. 9, pp. 604–612, 1992.

[36] D. L. Collins, A. P. Zijdenbos, V. Kollokian, *et al.*, «Design and construction of a realistic digital brain phantom», *IEEE transactions on medical imaging*, vol. 17, no. 3, pp. 463–468, 1998.

[37] V. Fonov, A. C. Evans, K. Botteron, *et al.*, «Unbiased average age-appropriate atlases for pediatric studies», *Neuroimage*, vol. 54, no. 1, pp. 313–327, 2011.

[38] S. Parisot, S. I. Ktena, E. Ferrante, *et al.*, «Disease prediction using graph convolutional networks: Application to autism spectrum disorder and alzheimer's disease», *Medical image analysis*, vol. 48, pp. 117–130, 2018.

[39] A. Barajas, S. Ochoa, J. E. Obiols, and L. Lalucat-Jo, «Gender differences in individuals at high-risk of psychosis», *Psychiatry*, vol. 22, no. 5, pp. 472–484, 2010.

[40] M. Byrne, E. Agerbo, H. Ewald, W. W. Eaton, and P. B. Mortensen, «Parental Age and Risk of Schizophrenia: A Case-control Study», *Archives of General Psychiatry*, vol. 60, no. 7, pp. 673–678, Jul. 2003, ISSN: 0003-990X. DOI: 10.1001/archpsyc.60.7.673. eprint: https://jamanetwork.com/journals/jamapsychiatry/articlepdf/207596/yoa20596.pdf. [Online]. Available: https://doi.org/10.1001/archpsyc.60.7.673.