



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ**

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ & ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Ψηφιακή επεξεργασία και κατηγοριοποίηση εικόνων μέσω
νευρωνικών δικτύων: εφαρμογή σε βιοϊατρικές εικόνες**

Φοιτητής: Ορέστης Καρέτσας

ΑΜ: 50346505

Επιβλέπουσα Καθηγήτρια

Μαρία Ραγκούση

ΑΘΗΝΑ-ΑΙΓΑΛΕΩ, ΔΕΚΕΜΒΡΙΟΣ 2020



**UNIVERSITY OF WEST ATTICA
FACULTY OF ENGINEERING
DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING**

Diploma Thesis

**Digital processing and neural network classification of images:
application to biomedical images**

Student: Orestis Karetsas

Registration Number: 50346505

Supervisor

Maria Rangoussi, Professor

ATHENS-EGALEO, DECEMBER 2020

Μέλη Εξεταστικής Επιτροπής

Μαρία Ραγκούση, καθηγήτρια	
Στυλιανός Ποτηράκης, καθηγητής	
Νικόλαος-Αλέξανδρος Τάτλας, αναπληρωτής καθηγητής	

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος ΚΑΡΕΤΣΑΣ ΟΡΕΣΤΗΣ του ΠΕΡΙΚΛΗ-ΓΕΩΡΓΙΟΥ, με αριθμό μητρώου 50346505, φοιτητής του Πανεπιστημίου Δυτικής Αττικής της Σχολής Μηχανικών του Τμήματος Ηλεκτρολόγων και Ηλεκτρονικών Μηχανικών, δηλώνω υπεύθυνα ότι:

«Είμαι ο συγγραφέας αυτής της διπλωματικής εργασίας και κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του τίτλου μου».

Ο Δηλών

Ορέστης Καρέτσας

Ο.Καρέτσας

Περίληψη

Στη παρούσα διπλωματική εργασία διερευνάται η ψηφιακή επεξεργασία και η κατηγοριοποίηση βιοϊατρικών εικόνων μέσω τεχνητών νευρωνικών δικτύων (artificial neural networks). Αρχικά γίνεται επισκόπηση (α) των βασικών αρχιτεκτονικών και λειτουργιών των (απλών) τεχνητών νευρωνικών δικτύων αφενός, και (β) των βασικών μεθόδων ψηφιακής επεξεργασίας εικόνων αφετέρου. Στο πλαίσιο της επισκόπησης μεθόδων ψηφιακής επεξεργασίας εικόνων, αναλύονται μέθοδοι βελτίωσης εικόνας μέσω διδιάστατων φίλτρων και μέθοδοι κατάτμησης εικόνας σε περιοχές μέσω ανάπτυξης περιοχών. Επίσης δίνονται απλά παραδείγματα εφαρμογής των ανωτέρω τεχνικών σε γενικές και σε βιοϊατρικές εικόνες. Στη συνέχεια παρουσιάζεται η βαθιά μάθηση (deep learning) και τα νευρωνικά δίκτυα βαθιάς μάθησης (deep neural networks). Στο πειραματικό μέρος της εργασίας, οργανώνονται και εκτελούνται τέσσερα (4) πειράματα δυαδικής κατηγοριοποίησης βιοϊατρικών εικόνων (θώρακα και εγκεφάλου). Για την κατηγοριοποίηση χρησιμοποιούνται αρχικά απλά και στη συνέχεια βαθιά νευρωνικά δίκτυα. Η συγκριτική παρουσίαση των αποτελεσμάτων δείχνει τη βελτίωση που φέρνει η βαθιά μάθηση.

Λέξεις – κλειδιά

Νευρωνικά δίκτυα, βαθιά μάθηση, βαθιά νευρωνικά δίκτυα, φίλτρα, ανάπτυξη περιοχής, βιοϊατρικές εικόνες, βελτιστοποίηση εικόνας, κατάτμηση εικόνας, κατηγοριοποίηση εικόνας

Abstract

Digital processing and classification of biomedical images through artificial neural networks is the aim of this thesis. In the first part of the thesis are reviewed the basics of digital image processing and the basics of classification using simple or ‘shallow’ neural networks. Within the digital image processing section, 2D filters for image enhancement and region growing for image segmentation are studied in more detail and through clarifying examples. Deep learning and deep neural networks are presented and analyzed in the following section. In the experimental part of the thesis, four binary classification experiments are designed and carried out on sets of biomedical images using shallow and deep neural networks. Comparison of the results reveals that deep neural networks offer improved classification performance.

Keywords

Neural networks, deep learning, deep neural networks, filters, region growing, biomedical images, image enhancement, image segmentation, image classification

Περιεχόμενα

Εισαγωγή	13
Κεφάλαιο 1: Απλά Νευρωνικά Δίκτυα	14
1.1 Τεχνητά Νευρωνικά Δίκτυα.....	14
1.2 Κυριότερες κατηγορίες ANN	17
1.3 Ιστορικά στοιχεία για τα ΤΝΔ	22
1.4 Εφαρμογές των ΤΝΔ	24
1.5 Παραδείγματα επίλυσης προβλημάτων με ΤΝΔ.....	25
1.5.1 Ταξινόμηση γραμμικά διαχωρίσιμων δεδομένων με Perceptron	25
1.5.2 Επίλυση του προβλήματος XOR με Perceptron πολλαπλών στρώσεων	27
Κεφάλαιο 2: Βαθιά Μάθηση και Νευρωνικά Δίκτυα Βαθιάς Μάθησης.....	31
2.1 Μηχανική μάθηση - Βαθιά μάθηση	31
2.2 Νευρωνικά Δίκτυα Βαθιάς Μάθησης	33
2.3 Συνελκτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks).....	35
2.3.1 Δημοφιλείς αρχιτεκτονικές CNN	38
2.4 Deep Belief Neural Networks	41
2.4.1 Εφαρμογές των DBNN	43
Κεφάλαιο 3: Βελτίωση Εικόνας με χρήση Φίλτρων	44
3.1 Η βελτίωση εικόνας (image enhancement)	44
3.2 Φίλτρα δύο διαστάσεων	45
3.2.1 Βαθυπερατό Φίλτρο 2 διαστάσεων (2D Low-Pass Filter)	45
3.2.2 Υψιπερατό Φίλτρο 2 διαστάσεων (2D High-Pass Filter)	50
3.2.3 Φίλτρο Ενδιάμεσης Τιμής 2 διαστάσεων (2D Median Filter)	52
Κεφάλαιο 4: Κατάτμηση Εικόνας με ανάπτυξη περιοχών	56
4.1 Κατάτμηση εικόνας	56
4.2. Η μέθοδος ανάπτυξης περιοχής (region growing).....	56
4.2.1 Παράδειγμα κατάτμησης απλής συνθετικής εικόνας με ένα σπόρο.....	57
4.2.2 Παράδειγμα κατάτμησης ασπρόμαυρης εικόνας με περισσότερους σπόρους	57
4.2.3 Παράδειγμα κατάτμησης βιοϊατρικών εικόνων με δέκα (10) σπόρους και βαθυπερατό φίλτρο	59
Κεφάλαιο 5: Κατηγοριοποίηση εικόνων με χρήση νευρωνικών δικτύων	63
5.1 Πρώτο πείραμα	63

5.1.1 Στόχος του 1 ^{ου} πειράματος.....	63
5.1.2 Data Set 1 ^{ου} πειράματος.....	64
5.1.3 Μέθοδος του 1 ^{ου} πειράματος σε βήματα	64
5.1.4 Αποτελέσματα του 1 ^{ου} πειράματος.....	66
5.2 Δεύτερο πείραμα	71
5.2.1 Στόχος του 2 ^{ου} πειράματος.....	71
5.2.2 Data Set 2 ^{ου} πειράματος.....	71
5.2.3 Μέθοδος του 2 ^{ου} πειράματος σε βήματα	72
5.2.4 Αποτελέσματα του 2 ^{ου} πειράματος.....	73
5.3 Τρίτο πείραμα.....	78
5.3.1 Στόχος του 3 ^{ου} πειράματος.....	78
5.3.2 Data Set 3 ^{ου} πειράματος.....	79
5.3.3 Μέθοδος του 3 ^{ου} πειράματος σε βήματα	79
5.3.4 Αποτελέσματα του 3 ^{ου} πειράματος.....	81
5.4 Τέταρτο πείραμα	87
5.4.1 Στόχος του 4 ^{ου} πειράματος.....	87
5.4.2 Data Set 4 ^{ου} πειράματος.....	88
5.4.3 Μέθοδος του 4 ^{ου} πειράματος σε βήματα	88
5.4.4 Αποτελέσματα του 4 ^{ου} πειράματος.....	89
5.4.5 Βελτίωση 4 ^{ου} πειράματος.....	95
Κεφάλαιο 6: Συμπεράσματα – Περαιτέρω έρευνα.....	99
Βιβλιογραφία - Αναφορές - Διαδικτυακές Πηγές	100
Παράρτημα Α.....	103
Παράρτημα Β.....	105
Παράρτημα Γ	111
Παράρτημα Δ.....	113

Κατάλογος εικόνων

Εικόνα 1.1: Απλό παράδειγμα ΤΝΔ τύπου Perceptron.....	17
Εικόνα 1.2: Απλό παράδειγμα ΤΝΔ τύπου Multilayer Perceptron με ένα (1) κρυφό στρώμα.	18
Εικόνα 1.3: Απλό παράδειγμα ANN τύπου Back Propagation.....	20
Εικόνα 1.4: Απλό παράδειγμα ANN τύπου Radial Basis Function.....	21
Εικόνα 1.5: Δημιουργία δύο ομάδων δεδομένων για διαχωρισμό αυτών.....	26
Εικόνα 1.6: Δημιουργία Perceptron για δυαδική ταξινόμηση.....	26
Εικόνα 1.7: Δημιουργία διαχωριστικής γραμμής για επιβεβαίωση σωστής ταξινόμησης. ...	27
Εικόνα 1.8: Πίνακας Αληθείας της λογικής πράξης Exclusive OR (XOR). True = 1, False = -1.	28
Εικόνα 1.9: Δημιουργία δύο κλάσεων με δύο πρότυπα η καθεμία, για δυαδική κατηγοριοποίηση.....	29
Εικόνα 1.10: Δημιουργία Multilayer Perceptron για την ταξινόμηση.....	29
Εικόνα 1.11: Δημιουργία Targets και Network response για έλεγχο του δίκτυο πάνω στα δεδομένα.....	30
Εικόνα 1.12: Δημιουργία διαχωριστικών πεδίων με την βοήθεια των δύο χρωμάτων για την σωστή ταξινόμηση των δύο κλάσεων.....	30
Εικόνα 2.1: Ένα βαθύ νευρωνικό δίκτυο με N κρυφά στρώματα.....	34
Εικόνα 2.2: Ταξινόμηση εικόνων για ένα πουλί.....	36
Εικόνα 2.3: Δομή του CNN με τα τέσσερα στρώματα.....	37
Εικόνα 2.4: Παράδειγμα CNN τύπου LeNet-5 (1998) με επτά στρώματα.....	38
Εικόνα 2.5: Παράδειγμα CNN τύπου AlexNet (2012).....	39
Εικόνα 2.6: Παράδειγμα CNN τύπου GoogleNet (2014).....	40
Εικόνα 2.7: Παράδειγμα CNN τύπου VGGNet (2014).....	41
Εικόνα 2.8: Αρχιτεκτονική DBNN.....	42
Εικόνα 3.1: Εικόνα 2D βαθυπερατού φίλτρου στο πεδίο των χωρικών συχνοτήτων.....	46
Εικόνα 3.1α: Αρχική απλή εικόνα. (Πηγή: https://en.wikipedia.org/wiki/Image_noise).....	47
Εικόνα 3.1β: Εικόνα μετά από εφαρμογή low pass filter με $D = 75$	47
Εικόνα 3.2α: Αρχική εικόνα εγκεφάλου (Πηγή: https://www.lpi.tel.uva.es/node/667).....	47
Εικόνα 3.2β: Εικόνα μετά από εφαρμογή low pass filter με $D = 75$	47
Εικόνα 3.3: Αρχική εικόνα θώρακα.....	48
Εικόνα 3.3α: Εικόνα μετά από εφαρμογή βαθυπερατού φίλτρου με $D=25$	48
Εικόνα 3.3β: Εικόνα μετά από εφαρμογή βαθυπερατού φίλτρου με $D=50$	48
Εικόνα 3.4α: Εικόνα region growing, μετά από εφαρμογή low pass filter με $D=25$	49
Εικόνα 3.4β: Εικόνα region growing, μετά από εφαρμογή low pass filter με $D=50$	49
Εικόνα 3.5α: Εικόνα region growing, μετά από εφαρμογή low pass filter με $D=25$	50
Εικόνα 3.5β: Εικόνα region growing, μετά από εφαρμογή low pass filter με $D=50$	50
Εικόνα 3.5γ: Εικόνα 2D υψιπερατού φίλτρου.....	51
Εικόνα 3.6α: Αρχική εικόνα κάμεραμαν (Πηγή: https://www.researchgate.net/figure/The-cameraman-image-256-256_fig7_228353323).....	51
Εικόνα 3.6β: Εικόνα μετά από εφαρμογή high pass filter με $D=25$	51
Εικόνα 3.7α: Αρχική εικόνα εγκεφάλου (Πηγή: https://www.lpi.tel.uva.es/node/667).....	52
Εικόνα 3.7β: Εικόνα μετά από εφαρμογή high pass filter με $D=5$	52
Εικόνα 3.8α: Αρχική εικόνα καμεραμάν.....	53

Εικόνα 3.8β: Εικόνα με πρόσθετο θόρυβο salt & pepper ισχύος 0.01	54
Εικόνα 3.8γ: Εικόνα μετά από εφαρμογή median filter στην θορυβώδη εικόνα.....	54
Εικόνα 3.9α: Αρχική εικόνα εγκεφάλου	54
Εικόνα 3.9β: Εικόνα με πρόσθετο θόρυβο salt & pepper ισχύος 0.01	55
Εικόνα 3.9γ: Εικόνα μετά από εφαρμογή median filter στη θορυβώδη εικόνα.....	55
Εικόνα 4.1α: Αρχική εικόνα.....	57
Εικόνα 4.1β: Τυχαίο σημείο-σπόρος στην αρχική εικόνα.....	57
Εικόνα 4.1γ: Εικόνα μετά την εφαρμογή region growing	57
Εικόνα 4.2α: Αρχική ασπρόμαυρη εικόνα.	58
Εικόνα 4.2β: Είκοσι (20) τυχαία σημεία-σπόροι πάνω στην αρχική εικόνα.....	58
Εικόνα 4.2γ: Εικόνα μετά την εφαρμογή region growing με κατώφλι 0.1	58
Εικόνα 4.3β: Είκοσι (20) τυχαία σημεία-σπόροι πάνω στην αρχική εικόνα.....	59
Εικόνα 4.3γ: Εικόνα μετά την εφαρμογή region growing με κατώφλι 0.15	59
Εικόνα 4.4α: Αρχική εικόνα θώρακα (Πηγή: https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia).....	60
Εικόνα 4.4β: Δέκα τυχαία σημεία στην αρχική εικόνα	60
Εικόνα 4.4γ: Εικόνα μετά την εφαρμογή region growing με κατώφλι 0.1	60
Εικόνα 4.4δ: Εικόνα μετά την εφαρμογή region growing και βαθυπερατού φίλτρου με D=50	60
Εικόνα 4.4ε: Εικόνα μετά την εφαρμογή region growing με κατώφλι 0.1	61
Εικόνα 4.4στ: Εικόνα μετά την εφαρμογή region growing και βαθυπερατού φίλτρου με D=50	61
Εικόνα 4.5α: Αρχική εικόνα εγκεφάλου (Πηγή: https://stackoverflow.com/questions/31995502/best-method-to-find-edge-of-noise-image)	61
Εικόνα 4.5β: Δέκα τυχαία σημεία στην αρχική εικόνα	61
Εικόνα 4.5γ: Εικόνα μετά την εφαρμογή region growing με κατώφλι 0.1	62
Εικόνα 4.5δ: Εικόνα μετά την εφαρμογή region growing & βαθυπερατού φίλτρου με D=50	62
Εικόνα 4.5ε: Εικόνα μετά την εφαρμογή region growing με κατώφλι 0.1	62
Εικόνα 4.5στ: Εικόνα μετά την εφαρμογή region growing & βαθυπερατού φίλτρου με D=50	62
Εικόνα 5.1α: MRI – Εγκέφαλος (Πηγή: https://www.kaggle.com/navoneel/brain-mri-images-for-brain-tumor-detection)	64
Εικόνα 5.1β: XRAY – Θώρακας (Πηγή: https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia)	64
Εικόνα 5.2: Νευρωνικό δίκτυο με ένα 1 hidden layer τεσσάρων κόμβων.	66
Εικόνα 5.3: Αποτέλεσμα εκπαίδευσης με 100% επιτυχία και στις δύο κατηγορίες.....	67
Εικόνα 5.4 : Αποτέλεσμα ελέγχου με 100% επιτυχία και στις δύο κατηγορίες.....	67
Εικόνα 5.5: Η επίδοση του νευρωνικού δικτύου	68
Εικόνα 5.6: Το ιστόγραμμα σφάλματος του νευρωνικού δικτύου	68
Εικόνα 5.7: Νευρωνικό δίκτυο με ένα hidden layer έξι κόμβων.	69
Εικόνα 5.8: Αποτέλεσμα εκπαίδευσης με 100% επιτυχία και στις δύο κατηγορίες.....	69
Εικόνα 5.9: Αποτέλεσμα ελέγχου με 100% επιτυχία και στις δύο κατηγορίες	69
Εικόνα 5.10: Η επίδοση του νευρωνικού δικτύου	70

Εικόνα 5.11: Το ιστόγραμμα σφάλματος του νευρωνικού δικτύου	70
Εικόνα 5.12α: MRI – Εγκέφαλος (Πηγή: https://www.kaggle.com/navoneel/brain-mri-images-for-brain-tumor-detection).....	71
Εικόνα 5.12β: XRAY - Θώρακας (Πηγή: https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia)	71
Εικόνα 5.13: Νευρωνικό δίκτυο με ένα 1 hidden layer τεσσάρων κόμβων.	74
Εικόνα 5.14: Αποτέλεσμα εκπαίδευσης με 98.75% επιτυχία στην πρώτη κλάση και 96.25% επιτυχία στην δεύτερη κλάση	74
Εικόνα 5.15: Αποτέλεσμα ελέγχου με 95% επιτυχία στην πρώτη κλάση και 65% επιτυχία στην δεύτερη κλάση.....	74
Εικόνα 5.16: Η επίδοση του νευρωνικού δικτύου	75
Εικόνα 5.17: Το ιστόγραμμα σφάλματος του νευρωνικού δικτύου	75
Εικόνα 5.18: Νευρωνικό δίκτυο με ένα 1 hidden layer έξι κόμβων.	76
Εικόνα 5.19: Αποτέλεσμα εκπαίδευσης με 100% επιτυχία στην πρώτη κλάση και 97.5% επιτυχία στην δεύτερη κλάση	76
Εικόνα 5.20: Αποτέλεσμα ελέγχου με 100% επιτυχία στην πρώτη κλάση και 95% επιτυχία στην δεύτερη κλάση.....	77
Εικόνα 5.21: Η επίδοση του νευρωνικού δικτύου	77
Εικόνα 5.22: Το ιστόγραμμα σφάλματος του νευρωνικού δικτύου	78
Εικόνα 5.23α: XRAY - Θώρακας υγιής.....	79
Εικόνα 5.23β: XRAY - Θώρακας με πνευμονία.....	79
Εικόνα 5.24: Νευρωνικό δίκτυο με ένα hidden layer έξι κόμβων.	81
Εικόνα 5.25: Αποτέλεσμα εκπαίδευσης με 100% επιτυχία και στις δύο κατηγορίες	82
Εικόνα 5.26: Αποτέλεσμα ελέγχου με 90% επιτυχία και στις δύο κατηγορίες	82
Εικόνα 5.27: Η επίδοση του νευρωνικού δικτύου	83
Εικόνα 5.28: Το ιστόγραμμα σφάλματος του νευρωνικού δικτύου	83
Εικόνα 5.29: Νευρωνικό δίκτυο με ένα hidden layer δέκα κόμβων.....	84
Εικόνα 5.30: Αποτέλεσμα εκπαίδευσης με 100% επιτυχία και στις δύο κατηγορίες	85
Εικόνα 5.31: Αποτέλεσμα ελέγχου με 90% επιτυχία στην πρώτη κλάση και 100% επιτυχία στην δεύτερη κλάση.....	85
Εικόνα 5.32: Η επίδοση του νευρωνικού δικτύου	86
Εικόνα 5.33: Το ιστόγραμμα σφάλματος του νευρωνικού δικτύου	86
Εικόνα 5.34α: XRAY - Θώρακας με πνευμονία.....	87
Εικόνα 5.34β: XRAY - Θώρακας υγιής	87
Εικόνα 5.35: Νευρωνικό δίκτυο με 1 hidden layer των έξι κόμβων.	90
Εικόνα 5.36: Αποτέλεσμα εκπαίδευσης με 87.5% επιτυχία στην πρώτη κλάση και 96.25% επιτυχία στην δεύτερη κλάση	90
Εικόνα 5.37: Αποτέλεσμα ελέγχου με 55% επιτυχία στην πρώτη κλάση και 90% επιτυχία στην δεύτερη κλάση.....	91
Εικόνα 5.38: Η επίδοση του νευρωνικού δικτύου	92
Εικόνα 5.39: Το ιστόγραμμα σφάλματος του νευρωνικού δικτύου	92
Εικόνα 5.40: Νευρωνικό δίκτυο με 1 hidden layer των δέκα κόμβων.....	93
Εικόνα 5.41: Αποτέλεσμα εκπαίδευσης με 98.75% επιτυχία στην πρώτη κλάση και 53.75% επιτυχία στην δεύτερη κλάση	93

Εικόνα 5.42: Αποτέλεσμα ελέγχου με 100% επιτυχία στην πρώτη κλάση και 10% επιτυχία στην δεύτερη κλάση.....	94
Εικόνα 5.43: Η επίδοση του νευρωνικού δικτύου	94
Εικόνα 5.44: Το ιστόγραμμα σφάλματος του νευρωνικού δικτύου.	95
Εικόνα 5.45: Νευρωνικό δίκτυο με 2 hidden layers των πέντε και δύο κόμβων, αντίστοιχα. 96	
Εικόνα 5.46: Αποτέλεσμα εκπαίδευσης με 100% επιτυχία και στις δύο κατηγορίες	96
Εικόνα 5.47: Αποτέλεσμα ελέγχου με 95% επιτυχία στην πρώτη κλάση και 70% επιτυχία στην δεύτερη κλάση.....	96
Εικόνα 5.48: Η επίδοση του νευρωνικού δικτύου	97
Εικόνα 5.49: Το ιστόγραμμα σφάλματος του νευρωνικού δικτύου	97

Αλφαβητικό Ευρετήριο

ANN - Artificial Neural Networks

MLP - Multilayer Perceptron

SNN - Shallow Neural Network

DNN - Deep Neural Network

RBFNN - Radial Basis Function Neural Network

GRNN - Generalized Regression Neural Network

SOM - Self Organizing Maps

LVQ - Learning Vector Quantization

CNN - Convolutional Neural Networks

DBNN - Deep Belief Neural Networks

ΤΝΔ - Τεχνητά Νευρωνικά Δίκτυα

ΒΝΔ - Βαθιά Νευρωνικά Δίκτυα

Εισαγωγή

Με τη ραγδαία εξέλιξη του κλάδου της τεχνολογίας διαρκώς επιτυγχάνονται νέα και καινοτομικά επιτεύγματα, τα οποία διευκολύνουν την καθημερινότητά μας σε διάφορους τομείς. Ένας από τους σημαντικότερους τομείς, τον οποίο αξιοποιεί και η ιατρική επιστήμη στην προσπάθειά της να βελτιώσει την υγεία του ανθρώπου καθώς και να κάνει ευκολότερη την δουλειά των ιατρών, είναι η αυτοματοποιημένη κατηγοριοποίηση ιατρικών εικόνων.

Στον τομέα των βιοϊατρικών και ιατρικών αναλύσεων, οι εικόνες έχουν τον κυρίαρχο ρόλο στην ανάλυση της κατάστασης μιας ασθένειας και στην διατύπωση μιας ακριβούς διάγνωσης για τους ασθενείς. Παρόλα αυτά, είναι τεράστια απαίτηση η διαχείριση και ανάλυση του τεράστιου όγκου βιοϊατρικών εικόνων που παράγονται καθημερινά. Είναι επίσης εξαιρετικά κουραστικό και πρακτικά / χρονικά ανέφικτο να εκτελείται αυτό το έργο επεξεργασίας και κατηγοριοποίησης αποκλειστικά μέσω ανθρώπινης παρέμβασης. Επομένως, για να γίνει πιο εύκολη η δουλειά των ιατρών, απαιτούνται αποδοτικές μέθοδοι αυτοματοποιημένης ανάλυσης και κατηγοριοποίησης βιοϊατρικών εικόνων. Με το θέμα αυτό ασχολείται η παρούσα διπλωματική εργασία. Για την κατηγοριοποίηση των εικόνων αξιοποιούνται τεχνητά νευρωνικά δίκτυα ρηχής και βαθιάς αρχιτεκτονικής, και συγκρίνονται τα αποτελέσματά τους.

Η παρούσα διπλωματική εργασία είναι οργανωμένη σε πέντε κεφάλαια. Το πρώτο κεφάλαιο περιγράφει τα απλά νευρωνικά δίκτυα. Το δεύτερο κεφάλαιο αναφέρεται στη βαθιά μάθηση και στα νευρωνικά δίκτυα βαθιάς μάθησης. Στο τρίτο κεφάλαιο γίνεται επισκόπηση των βασικών εργαλείων ψηφιακής επεξεργασίας εικόνων, με έμφαση στη βελτίωση εικόνας με χρήση φίλτρων. Στο τέταρτο κεφάλαιο αναλύεται η κατάτμηση εικόνας με τη μέθοδο ανάπτυξης περιοχών. Στο πέμπτο κεφάλαιο, που αποτελεί και το πειραματικό μέρος της εργασίας, οργανώνονται και εκτελούνται τέσσερα πειράματα δυαδικής κατηγοριοποίησης βιοϊατρικών εικόνων με χρήση νευρωνικών δικτύων ρηχής και βαθιάς αρχιτεκτονικής, και συγκρίνονται τα αποτελέσματα. Η εργασία κλείνει με τα Συμπεράσματα και τη Βιβλιογραφία. Τέλος στα Παραρτήματα παρατίθεται ο κώδικας που αναπτύχθηκε για το πειραματικό μέρος.

Κεφάλαιο 1: Απλά Νευρωνικά Δίκτυα

1.1 Τεχνητά Νευρωνικά Δίκτυα

Τα τεχνητά νευρωνικά δίκτυα – ΤΝΔ (Artificial Neural Networks – ANN) είναι προγράμματα λογισμικού που εκτελούνται σε ηλεκτρονικό υπολογιστή και βασίζονται στο πρότυπο του ανθρώπινου εγκεφάλου. Δεν περιέχουν όλες τις λεπτομέρειες της δομής και λειτουργίας του εγκεφάλου, τα οποία εξ άλλου δεν είναι γνωστά ακόμα και σήμερα καθώς οι διεργασίες που επιτελούνται από τα βιολογικά νευρωνικά δίκτυα στους ζώντες οργανισμούς είναι πολύ περίπλοκες, αλλά και τόσο χρήσιμες στην καθημερινή ζωή του ανθρώπου. Μερικές από αυτές είναι εργασίες ρουτίνας, τις οποίες ο ανθρώπινος εγκέφαλος εκτελεί με ελάχιστη ή μηδαμινή προσπάθεια, όπως η αναγνώριση μιας εικόνας. Το ερώτημα που προκύπτει λοιπόν είναι αν μπορούν οι ηλεκτρονικοί υπολογιστές να κάνουν αυτά (ή ορισμένα από αυτά) που κάνει το ανθρώπινο μυαλό.

Πολλά από τα πιο απλά προβλήματα (tasks), όπως η αναγνώριση φωνής ή εικόνας που το μυαλό κάνει πολύ εύκολα, οι υπολογιστές δεν μπορούν να τα κάνουν ούτε με ταχύτητα ούτε με επιτυχία – και βέβαια αυτό δεν οφείλεται στην έλλειψη ταχύτητας του hardware, καθ' ότι οι υπολογιστές σήμερα εξαιρετικά υψηλών ταχυτήτων. Ο λόγος είναι ότι η δομή των υπολογιστών είναι πάρα πολύ διαφορετική από την δομή του εγκεφάλου. Θα υπήρχε όμως η δυνατότητα να φτιαχτεί ένας υπολογιστής με τέτοια εσωτερική δομή που να μοιάζει με την δομή του εγκεφάλου, και έτσι να μπορεί να πετύχει αυτό που θέλουμε. Αυτό έχει οδηγήσει σε πρώτες σκέψεις πώς είναι δυνατόν να δημιουργηθούν πρότυπα (μοντέλα) του νευρωνικού συστήματος του ανθρώπου, τα οποία θα περιέχουν όλα τα χαρακτηριστικά που είναι γνωστά μέχρι σήμερα, και τα οποία θα μπορούσαν από μόνα τους να επιτελέσουν τις εργασίες αυτές, με τον ίδιο τρόπο που γίνονται στα βιολογικά νευρωνικά δίκτυα. Χρησιμοποιούν λοιπόν μόνο την κεντρική ιδέα της λογικής λειτουργίας του εγκεφάλου, ξεκινώντας από μία συλλογή απλών υπολογιστικών μονάδων, που είναι αντίστοιχες προς τους νευρώνες στη φυσιολογία του ανθρώπου, και προσπαθούν να επιτελέσουν τις ανάλογες διεργασίες αξιοποιώντας τη διασύνδεση (network) πολλών από αυτές τις απλές μονάδες.

Η βασική διαφορά των τεχνητών από τα βιολογικά νευρωνικά δίκτυα είναι ότι τα τεχνητά δίκτυα παίρνουν γνώσεις (μαθαίνουν) με την εξάσκηση και την εμπειρία, όπως ακριβώς και οι άνθρωποι, αλλά διαφέρουν στο ότι δεν ακολουθούν ορισμένους προκαθορισμένους κανόνες, που είναι χαρακτηριστικό των υπολογιστών. Υπάρχει σήμερα ένας μεγάλος όγκος έρευνας στην περιοχή αυτή, καθ' όσον και εδώ είναι προφανές πόσο χρήσιμο θα ήταν να μπορεί κάτι το άψυχο να επιτελεί εργασίες που μέχρι σήμερα μόνο ο άνθρωπος μπορούσε να κάνει, είτε αυτό είναι μια μηχανή, είτε ένα πρόγραμμα ηλεκτρονικού υπολογιστή. Τα τελευταία τριάντα χρόνια λοιπόν, τα ΤΝΔ έχουν δει μεγάλη άνθηση και πρόοδο. Τελικά όμως τα βιολογικά και τα τεχνητά δίκτυα διαφέρουν πάρα πολύ ως προς την αρχιτεκτονική και τις ιδιότητές τους.

Ο κυρίαρχος στόχος της λειτουργίας ενός ΤΝΔ είναι να μπορεί να επιτελεί αυτόματα ορισμένες διεργασίες, όπως για παράδειγμα να αναγνωρίζει εικόνες, αφού όμως προηγουμένως εκπαιδευθεί κατάλληλα. Αυτό επιτυγχάνεται σε δύο φάσεις:

- (a) Φάση εκπαίδευσης (training), και
- (b) Φάση ελέγχου (testing).

Φάση Εκπαίδευσης

Κάθε δίκτυο δέχεται ορισμένες εισόδους και δίδει ορισμένες εξόδους (input-output). Η εκπαίδευση γίνεται με το να παρουσιαστεί μια ομάδα από πρότυπα στο δίκτυο, αντιπροσωπευτικά ή παρόμοια με αυτά που θέλει κάποιος να «μάθει» το δίκτυο. Αυτό σημαίνει ότι δίνεται στο δίκτυο ως είσοδος μία σειρά από παραδείγματα (training set), για τα οποία είναι γνωστή η σωστή απάντηση ή έξοδος. Ταυτόχρονα παρέχεται στο δίκτυο και η αντίστοιχη σωστή απάντηση (έξοδος) για κάθε είσοδο, δηλαδή τι πρέπει να δίνει το δίκτυο ως απάντηση στην κάθε είσοδο που του παρουσιάζεται. Ουσιαστικά είναι σαν να δίνουμε στο δίκτυο μια σειρά από ερωτήσεις και ταυτόχρονα και την απάντηση που αντιστοιχεί στην καθεμία. Το δίκτυο με τα δεδομένα αυτά τροποποιεί την εσωτερική του δομή (ρυθμίζει επαναληπτικά τις τιμές των συντελεστών του) ώστε να παράγει ως έξοδο (απάντηση) την επιθυμητή (τη σωστή) απάντηση. Η εκπαίδευση περιλαμβάνει επαναληπτική εισαγωγή των εισόδων και των σωστών εξόδων (training set) πολλές φορές, μέχρι η διαδικασία αυτή να συγκλίνει, δηλαδή να ρυθμιστούν οι τιμές των συντελεστών του ΤΝΔ ώστε το σφάλμα μεταξύ πραγματικής και επιθυμητής εξόδου για κάθε είσοδο να ελαχιστοποιηθεί. Για να

σταματήσει η επαναληπτική διαδικασία της εκπαίδευσης και να θεωρηθεί το ΤΝΔ «εκπαιδευμένο», χρησιμοποιούνται διάφορα κριτήρια που έχουν σχέση με το ανωτέρω σφάλμα (διαφορά πραγματικών και επιθυμητών εξόδων).

Φάση Ελέγχου

Αφού ολοκληρωθεί η φάση της εκπαίδευσης και το ΤΝΔ έχει πλέον συγκλίνει σε συγκεκριμένες τιμές για τους συντελεστές της εσωτερικής του δομής, ακολουθεί η φάση του ελέγχου. Το ΤΝΔ καλείται να λύσει άλλα ανάλογα προβλήματα τα οποία δεν τα έχει δει προηγουμένως, καθώς δεν έχει εκπαιδευθεί στα πρότυπα των προβλημάτων αυτών. Αυτό ισοδυναμεί με το να δεχθεί νέες, άγνωστες εισόδους που δεν υπήρχαν μέσα στο training set, και οι οποίες αποτελούν το testing set. Για τις εισόδους αυτές δεν είναι γνωστή η «σωστή» έξοδος και πρέπει να την παρέχει το ΤΝΔ. Οποσδήποτε όμως, τα προβλήματα αυτά θα πρέπει να είναι της ίδιας φύσης και των ίδιων χαρακτηριστικών όπως αυτά της εκπαίδευσης και όχι ριζικά διαφορετικά, για να έχουμε πιθανότητες επιτυχούς λειτουργίας.

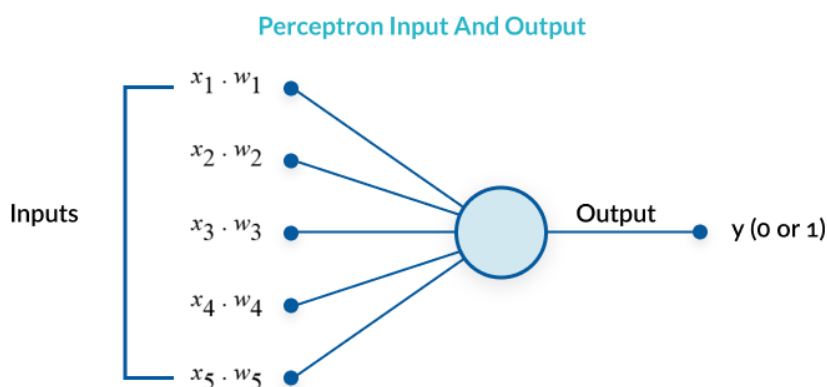
Συνοψίζοντας, τα ΤΝΔ "μαθαίνουν" να εκτελούν εργασίες μετά από εκπαίδευση μέσω παραδειγμάτων, χωρίς να προγραμματίζονται ώστε να παράγουν συγκεκριμένη «λύση» με βάση είτε συγκεκριμένους κανόνες είτε κάποιο μαθηματικό τύπο. Παραδείγματος χάριν, στο θέμα της αναγνώρισης εικόνων, ένα ΤΝΔ μπορεί να μάθει να ξεχωρίζει εικόνες που περιέχουν γάτες, αφού έχει πρώτα εκπαιδευθεί με παραδείγματα εικόνων που έχουν επισημανθεί χειροκίνητα ως "γάτα" ή "όχι γάτα" και χρησιμοποιώντας την εκπαίδευσή του για να ξεχωρίζει αν υπάρχει γάτα ή όχι σε άλλες, άγνωστες εικόνες. Επιλύουν αυτό το πρόβλημα χωρίς προηγούμενη ειδική γνώση για τις γάτες, ή κανόνες για την απόφαση τι είναι γάτα, για παράδειγμα ότι έχουν γούνα, ουρές, μουστάκια, 2 αυτιά, 4 πόδια, κλπ. Αντ' αυτού, παράγουν αυτόματα τα δικά τους αναγνωριστικά χαρακτηριστικά μέσα από τα παραδείγματα του training set που επεξεργάζονται – και δεν μπορούν να μας επιστρέψουν σε κατανοητή μορφή ποια ήταν αυτά τα χαρακτηριστικά.

Τα ΤΝΔ αναπτύχθηκαν κυρίως στο δεύτερο μισό του 20^{ου} αιώνα, και έχουν πετύχει αρκετά εντυπωσιακά αποτελέσματα, αλλά έχει φανεί επίσης ότι έχουν και αρκετούς περιορισμούς. Οι περιορισμοί προκύπτουν συχνά όταν το μέγεθος και η πολυπλοκότητα του συστήματος αυξάνουν. Είναι ιδιαίτερα ικανά σε συνδυαστικά προβλήματα και σε γενικεύσεις. Αντίθετα, δεν είναι ικανά σε προβλήματα λογικής

και σε ακριβείς υπολογισμούς. Υπάρχουν πάρα πολλά μοντέλα δικτύων με διαφορετική φιλοσοφία και τρόπο λειτουργίας, και πολλές και διάφορες εφαρμογές τους σε πλήθος πεδίων της επιστήμης και της τεχνολογίας.

1.2 Κυριότερες κατηγορίες ANN

- **Perceptron:** Το perceptron είναι ένας απλός αλγόριθμος δυαδικής (binary) ταξινόμησης, που προτάθηκε από τον επιστήμονα Frank Rosenblatt του Πανεπιστημίου Cornell των ΗΠΑ. Στόχος του είναι να χωρίσει ένα σύνολο σημάτων εισόδου σε δύο μέρη - "ναι" και "όχι". Σε αντίθεση με πολλούς άλλους αλγορίθμους ταξινόμησης, το perceptron μοντελοποιήθηκε πάνω στη βασική μονάδα επεξεργασίας του ανθρώπινου εγκεφάλου, τον νευρώνα, και έχει μια ασυνήθιστη ικανότητα να μαθαίνει και να λύνει σύνθετα προβλήματα.



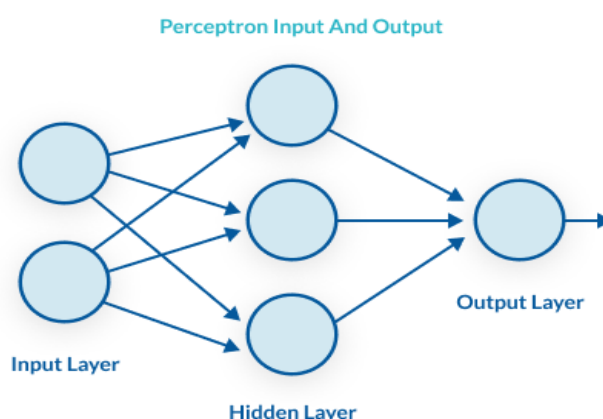
Εικόνα 1.1: Απλό παράδειγμα ΤΝΔ τύπου Perceptron.

(Πηγή: <https://missinglink.ai/guides/neural-network-concepts/perceptrons-and-multi-layer-perceptrons-the-artificial-neuron-at-the-core-of-deep-learning/>)

Ένα perceptron είναι μια πολύ απλή μηχανή εκμάθησης. Μπορεί να δεχθεί μερικές εισόδους x_i , $i = 1, \dots, N$, κάθε μία από τις οποίες πολλαπλασιάζεται με ένα αντίστοιχο συντελεστή ή «βάρος» w_i , $i = 1, 2, \dots, N$ που υποδηλώνει πόσο σημαντική είναι η αντίστοιχη είσοδος, να αθροίσει τα επιμέρους γινόμενα ($x_i \cdot w_i$) και να παράγει μια έξοδο "0" ή "1" που αποτελεί και την απόφαση του perceptron για την κλάση (μία από τις δύο) που ανήκε η συγκεκριμένη είσοδος. Ωστόσο, όταν συνδυάζεται με πολλά άλλα perceptrons, σχηματίζει ένα ΤΝΔ. Ένα ΤΝΔ μπορεί, θεωρητικά, να απαντήσει

σε κάθε ερώτηση, αρκεί να διαθέτει επαρκή δεδομένα εκπαίδευσης και επαρκή υπολογιστική ισχύ.

- **Multilayer Perceptron:** Ένα perceptron πολλαπλών στρωμάτων (Multilayer Perceptron, MLP) είναι ένα perceptron που συνδυάζεται με άλλα πρόσθετα perceptrons, στοιβαγμένα σε επάλληλα στρώματα, για την επίλυση πιο σύνθετων προβλημάτων από την απλή δυαδική ταξινόμηση. Το παρακάτω σχήμα δείχνει ένα παράδειγμα MLP με τρία (3) στρώματα. Κάθε perceptron δέχεται εισόδους στο πρώτο αριστερά στρώμα (το στρώμα εισόδου – input layer), τις πολλαπλασιάζει επί τα αντίστοιχα βάρη και στέλνει τα αποτελέσματα (τις εξόδους του) σε όλα τα perceptrons του δεύτερου στρώματος (το κρυμμένο στρώμα – hidden layer). Οι εξοδοί του δεύτερου στρώματος αφού έχουν πολλαπλασιαστεί με τα αντίστοιχα βάρη στέλνονται σε όλους τους κόμβους του τρίτου και τελικού στρώματος δεξιά (στρώμα εξόδου – output layer).



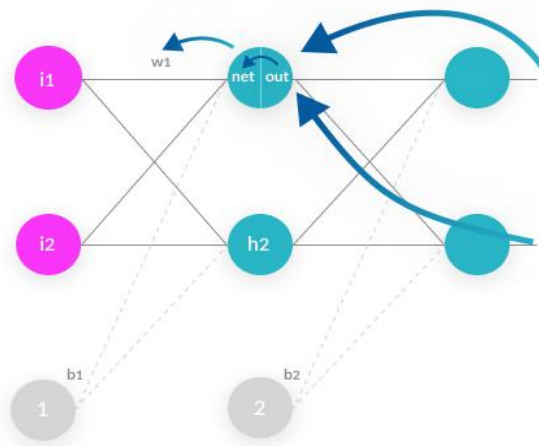
Εικόνα 1.2: Απλό παράδειγμα ΤΝΔ τύπου Multilayer Perceptron με ένα (1) κρυφό στρώμα.

(Πηγή: <https://missinglink.ai/guides/neural-network-concepts/perceptrons-and-multi-layer-perceptrons-the-artificial-neuron-at-the-core-of-deep-learning/>)

Κάθε perceptron στέλνει σήματα εξόδου προς κάθε perceptron στο επόμενο στρώμα. Για κάθε σήμα, το perceptron χρησιμοποιεί διαφορετικά βάρη. Στο διάγραμμα παραπάνω, κάθε γραμμή που πηγαίνει από ένα perceptron ενός στρώματος σε ένα perceptron στο επόμενο στρώμα, αντιπροσωπεύει μια

διαφορετική έξοδο. Κάθε στρώμα μπορεί να έχει μεγάλο αριθμό perceptrons, και επιπλέον μπορεί να υπάρχουν πολλαπλά στρώματα, οπότε το perceptron πολλαπλών στρωμάτων μπορεί να γίνει άμεσα ένα πολύ περίπλοκο σύστημα. Ένα MLP τριών επιπέδων, όπως και στο παραπάνω σχήμα, ονομάζεται «μη-βαθύ» (Non-Deep) ή «ρηχό» νευρωνικό δίκτυο (Shallow Neural Network, SNN). Ένα MLP με τέσσερα (4) ή περισσότερα στρώματα ονομάζεται «βαθύ νευρωνικό δίκτυο (Deep Neural Network, DNN). Μια διαφορά μεταξύ ενός MLP και ενός άλλου νευρωνικού δικτύου είναι ότι στο κλασικό perceptron, η λειτουργία απόφασης αποτελεί μια βαθμίδα και η έξοδος είναι δυαδική. Στα νευρωνικά δίκτυα που εξελίχθηκαν από τα MLPs, μπορούν να χρησιμοποιηθούν και άλλες λειτουργίες ενεργοποίησης οι οποίες έχουν ως αποτέλεσμα εξόδους πραγματικών τιμών, συνήθως μεταξύ 0 και 1 ή μεταξύ -1 και 1. Αυτό επιτρέπει επίλυση προβλημάτων πρόβλεψης με βάση την πιθανότητα (η πιθανότητα αποτελεί τιμή μεταξύ 0 και 1) ή ταξινόμηση αντικειμένων σε πολλαπλές ετικέτες (κατηγορίες).

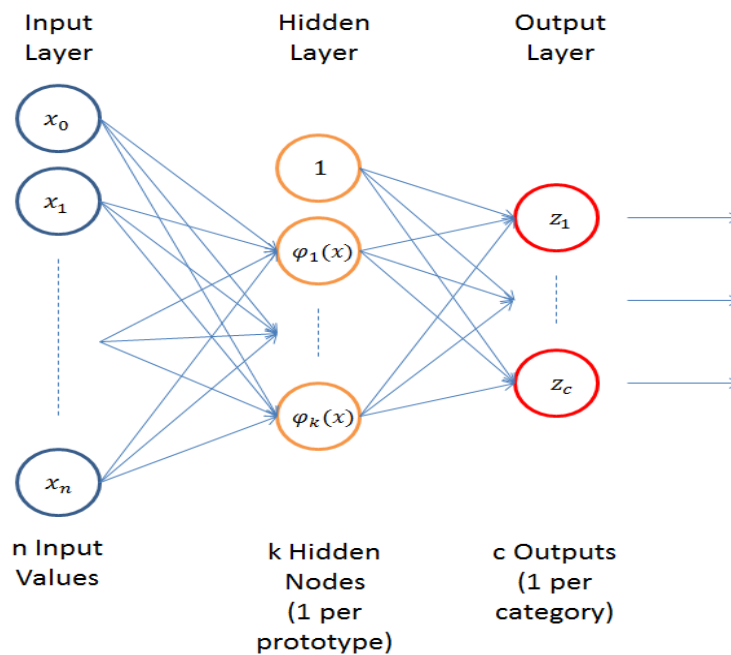
- **Back Propagation NN:** Η οπισθοτροφοδότηση (backpropagation) είναι ένας αλγόριθμος που χρησιμοποιείται αρκετά στην εκπαίδευση των ΤΝΔ. Όταν το ΤΝΔ αρχικοποιηθεί, τα βάρη ρυθμίζονται για κάθε μεμονωμένο στοιχείο του, που ονομάζεται νευρώνας, σε αρχικές τιμές (π.χ. μηδενικές τιμές ή τυχαίες τιμές). Στη συνέχεια οι είσοδοι φορτώνονται, περνούν μέσω του δικτύου των νευρώνων και το δίκτυο παρέχει μια έξοδο για κάθε είσοδο, όπως αυτή προκύπτει από τα αρχικά βάρη. Η πραγματική έξοδος συγκρίνεται με τη σωστή (επιθυμητή) έξοδο, και το σφάλμα (η διαφορά τους) διαδίδεται προς τα πίσω (από την έξοδο πίσω προς την είσοδο) στρώμα-στρώμα, διορθώνοντας κατάλληλα τους συντελεστές των στρωμάτων. Ο αλγόριθμος backpropagation βοηθάει στην προσαρμογή των βαρών των νευρώνων, ώστε η τελική έξοδος πλησιάζει αρκετά την γνωστή (σωστή) έξοδο για κάθε είσοδο εντός του συνόλου εκπαίδευσης (training set).



Εικόνα 1.3: Απλό παράδειγμα ANN τύπου Back Propagation.

(Πηγή: <https://missinglink.ai/guides/neural-network-concepts/perceptrons-and-multi-layer-perceptrons-the-artificial-neuron-at-the-core-of-deep-learning/>)

- **Radial Basis Function NN (RBFNN):** Ένα δίκτυο λειτουργιών ακτινικής βάσης (RBFN) είναι ένας ιδιαίτερος τύπος ΤΝΔ. Γενικά, ο όρος ΤΝΔ συνήθως αναφέρεται στο Multiple Perceptron (MLP). Κάθε νευρώνας σε ένα MLP παίρνει το σταθμισμένο άθροισμα των τιμών εισόδου του, δηλαδή κάθε τιμή εισόδου πολλαπλασιάζεται με έναν συντελεστή (βάρος) και τα αποτελέσματα αθροίζονται όλα μαζί. Ένας απλός νευρώνας MLP είναι ένας απλός γραμμικός ταξινομητής, αλλά σύνθετοι μη γραμμικοί ταξινομητές μπορούν να κατασκευαστούν συνδυάζοντας αυτούς τους νευρώνες σε ένα δίκτυο. Αντίθετα, ένα RBFN εκτελεί την ταξινόμηση μετρώντας την ομοιότητα της κάθε εισόδου με παραδείγματα από το σετ εκπαίδευσης. Κάθε νευρώνας RBFN αποθηκεύει ένα "πρωτότυπο", το οποίο είναι μόνο ένα από τα παραδείγματα από το σετ εκπαίδευσης. Όταν χρειάζεται να ταξινομηθεί μια νέα είσοδος, κάθε νευρώνας υπολογίζει την Ευκλείδεια απόσταση μεταξύ της εισόδου και του πρωτοτύπου της. Σε γενικές γραμμές, σε ένα πρόβλημα δυαδικής κατηγοριοποίησης σε 2 κλάσεις A και B, αν η δεδομένη είσοδος μοιάζει περισσότερο με τα πρωτότυπα κατηγορίας A από τα πρωτότυπα κατηγορίας B, ταξινομείται ως κλάση A.



Εικόνα 1.4: Απλό παράδειγμα ANN τύπου Radial Basis Function.

(Πηγή: <https://mccormickml.com/2013/08/15/radial-basis-function-network-rbfn-tutorial/>)

Εδώ το RBFN θεωρείται ως ένα "δίκτυο 3 επιπέδων" όπου το διάνυσμα εισόδου είναι το πρώτο στρώμα, το δεύτερο "κρυμμένο" στρώμα είναι οι νευρώνες RBF και το τρίτο στρώμα είναι το στρώμα εξόδου που περιέχει γραμμικούς συνδυασμούς νευρώνων.

- **Generalized Regression Neural Network (GRNN):** Το νευρωνικό δίκτυο γενικευμένης παλινδρόμησης (GRNN) είναι μια παραλλαγή των νευρωνικών δικτύων ακτινικής βάσης. Προτάθηκε από τον D.F. Specht το 1991. Μπορεί να χρησιμοποιηθεί για προβλήματα παλινδρόμησης, πρόβλεψης και ταξινόμησης. Επίσης αποτελεί καλή λύση για ηλεκτρονικά δυναμικά συστήματα. Τέλος αντιπροσωπεύει μια βελτιωμένη τεχνική εκπαίδευσης ΤΝΔ με βάση τη μη παραμετρική παλινδρόμηση. Η ιδέα είναι ότι κάθε δείγμα εκπαίδευσης θα αντιπροσωπεύει ένα μέσο για ένα ακτινικό βασικό νευρώνα.
- **Self Organizing Maps (SOM):** Οι αυτό-οργανούμενοι χάρτες (Self-organized Maps - SOM) εισήχθησαν από τον Φινλανδό επιστήμονα Τεουνο Kohonen στη δεκαετία του 1980 και ορισμένες φορές ονομάζονται και Χάρτες Kohonen (Kohonen Map). Ένας αυτοοργανούμενος χάρτης είναι ένας τύπος ΤΝΔ που εκπαιδεύεται χωρίς επίβλεψη, για να παράγει ένα διακριτικό χάρτη

(παράσταση) λίγων διαστάσεων (συνήθως διδιάστατο). Ως εκ τούτου είναι μια μέθοδος για τη μείωση των διαστάσεων. Οι αυτοοργανούμενοι χάρτες διαφέρουν από τα άλλα ΤΝΔ καθώς εφαρμόζουν ανταγωνιστική μάθηση (competitive layers) σε αντίθεση με τη μάθηση μέσω διόρθωσης / ελαχιστοποίησης των σφαλμάτων των MLP, π.χ., αλλά επίσης διαφέρουν και υπό την έννοια ότι χρησιμοποιούν μια λειτουργία γειτονικών κόμβων για τη διατήρηση των τοπολογικών ιδιοτήτων του χώρου εισόδου .

- **Learning Vector Quantization (LVQ):** Εκπαιδευτική ανίχνευση διανύσματος LVQ είναι ένα ανταγωνιστικό (competitive) ΤΝΔ που χρησιμοποιεί εποπτευόμενη μάθηση (supervised learning). Μπορεί να οριστεί ως διαδικασία ταξινόμησης μοτίβων όπου κάθε μονάδα εξόδου αντιπροσωπεύει μια κλάση. Καθώς χρησιμοποιεί εποπτευόμενη μάθηση, στο δίκτυο θα πρέπει να δοθεί ένα σύνολο προτύπων εκπαίδευσης που ανήκει σε γνωστή κλάση, μαζί με μια αρχική κατανομή της τάξης εξόδου. Μετά την ολοκλήρωση της διαδικασίας εκπαίδευσης, το LVQ θα είναι σε θέση να ταξινομήσει ένα άγνωστο διάνυσμα εισόδου, κατατάσσοντάς το στην ίδια κλάση με εκείνη της μονάδας εξόδου.

1.3 Ιστορικά στοιχεία για τα ΤΝΔ

Αν και η μελέτη του ανθρώπινου εγκεφάλου είναι χιλιάδων ετών, το πρώτο βήμα προς τα ΤΝΔ έλαβε χώρα το 1943, όταν ο Warren McCulloch, ένας νευροφυσιολόγος και ο Walter Pitts, ένας νέος μαθηματικός, έγραψαν ένα άρθρο σχετικά με τον τρόπο που οι νευρώνες μπορούν να λειτουργούν. Σχεδίασαν ένα απλό νευρωνικό δίκτυο με ηλεκτρικά κυκλώματα.

Το 1949, ο Donald Hebb ενίσχυσε την έννοια των νευρώνων στο βιβλίο του «Ο Οργανισμός Συμπεριφοράς». Επισημαίνει ότι οι νευρικές οδοί ενισχύονται κάθε φορά που χρησιμοποιούνται.

Στη δεκαετία του 1950, ο Nathaniel Rochester από τα ερευνητικά εργαστήρια της IBM έκανε πρώτος προσπάθεια να προσομοιώσει ένα ΤΝΔ.

Το 1956, το Dartmouth Summer Research Project για την Τεχνητή Νοημοσύνη (Artificial Intelligence, AI) έδωσε ώθηση τόσο στην τεχνητή νοημοσύνη όσο και στα

TND. Έτσι ξεκίνησε η έρευνα στην ΑΙ σε συσχέτιση με τις χαμηλότερου επιπέδου λειτουργίες του νευρικού συστήματος του εγκεφάλου.

Το 1957, ο John von Neumann πρότεινε να γίνει απομίμηση απλών λειτουργιών νευρώνων χρησιμοποιώντας το τηλεγράφημα ή σωλήνες κενού.

Το 1958, ο Frank Rosenblatt, ένας νευροβιολόγος του Πανεπιστημίου Cornell των ΗΠΑ, άρχισε να εργάζεται στο Perceptron. Ήταν ενθουσιασμένος με τη λειτουργία του ματιού μιας μύγας. Ένα μεγάλο μέρος της επεξεργασίας οδηγεί μια μύγα να πετάξει γίνεται στα μάτια της. Το Perceptron, το οποίο προέκυψε από αυτή την έρευνα, χτίστηκε σε υλικό (hardware) και όχι σε λογισμικό (software) και είναι το παλαιότερο TND που χρησιμοποιείται ακόμη και σήμερα. Ένα perceptron μονού στρώματος (single-layer) βρέθηκε χρήσιμο στην δυαδική ταξινόμηση ενός συνόλου αναλογικών εισόδων. Το perceptron υπολογίζει ένα σταθμισμένο άθροισμα των εισόδων, συγκρίνει το άθροισμά τους με ένα «κατώφλι» (threshold) και εξάγει μία από τις δύο πιθανές τιμές ως αποτέλεσμα.

Το 1959, οι Bernard Widrow και Marcian Hoff του Πανεπιστημίου Stanford των ΗΠΑ ανέπτυξαν μοντέλα που ονόμαζαν ADALINE και MADALINE. Αυτά τα μοντέλα ονομάστηκαν έτσι γιατί έκαναν χρήση πολλαπλών ADaptive LINear Elements. Το MADALINE ήταν το πρώτο TND που εφαρμόστηκε σε ένα πραγματικό πρόβλημα. Είναι ένα προσαρμοστικό φίλτρο (adaptive filter) που εξαλείφει την ηχώ στις τηλεφωνικές γραμμές (echo cancellation). Αυτό το TND βρίσκεται ακόμα σε εμπορική χρήση.

Στο βιβλίο τους “Perceptrons” οι Marvin Minsky & Seymour Papert απέδειξαν ότι η εκπαίδευση ενός Perceptron είναι διαδικασία που δεν συγκλίνει πάντα. Αυτό είχε ως αποτέλεσμα να ανακοπεί και τελικά να σταματήσει η πρόοδος στην έρευνα των TND μέχρι το 1981, καθώς σταμάτησε και ένα μεγάλο μέρος της χρηματοδότησης. Αυτή η περίοδος διήρκεσε μέχρι το 1981.

Το 1982, ο John Hopfield παρουσίασε ένα άρθρο στην αμερικανική Εθνική Ακαδημία Επιστημών (National Academy of Sciences), με την προσέγγισή του για τη δημιουργία χρήσιμων διατάξεων TND. Ταυτόχρονα, εκείνη την ίδια χρονιά η κοινή διάσκεψη ΗΠΑ-Ιαπωνίας για τα Συνεργατικά / Ανταγωνιστικά Νευρωνικά Δίκτυα, στην οποία η Ιαπωνία ανακοίνωσε την 5^η γενιά των προσπαθειών της, έκανε τις ΗΠΑ

να ανησυχούν για την αργοπορία τους. Σύντομα η χρηματοδότηση της έρευνας στα ΤΝΔ άρχισε να ρέει και πάλι.

Το 1985 το Αμερικανικό Ινστιτούτο Φυσικής (AIP) ξεκίνησε αυτό που έχει καθιερωθεί πλέον ως η ετήσια συνάντηση «Νευρωνικά Δίκτυα για την Πληροφορική». Μέχρι το 1987, το πρώτο Διεθνές Συνέδριο για τα ΤΝΔ του Ινστιτούτου Ηλεκτρολόγων Ηλεκτρονικών Μηχανικών (IEEE) συγκέντρωσε περισσότερους από 1.800 συμμετέχοντες.

Το 1997 προτάθηκε από τους Schmidhuber & Hochreiter το πλαίσιο του επαναλαμβανόμενου νευρικού δικτύου, Long Short-Term Memory (LSTM).

Το 1998 ο Yann LeCun δημοσίευσε αλγόριθμο εκπαίδευσης ΤΝΔ με εφαρμογή στην αναγνώριση εγγράφων.

1.4 Εφαρμογές των ΤΝΔ

Σήμερα τα ΤΝΔ έχουν φέρει επανάσταση στην επιχειρηματική και καθημερινή ζωή, οδηγώντας μας στο επόμενο επίπεδο της τεχνητής νοημοσύνης (Artificial Intelligence, AI). Με τη μίμηση του τρόπου λειτουργίας των διασυνδεδεμένων εγκεφαλικών κυττάρων, οι μηχανές με δυνατότητα NN (συμπεριλαμβανομένων των smartphones και υπολογιστών που χρησιμοποιούνται καθημερινά) πλέον εκπαιδεύονται για να μάθουν να αναγνωρίζουν πρότυπα και να κάνουν προβλέψεις με ανθρώπινο τρόπο, καθώς και να λύνουν προβλήματα σε κάθε επαγγελματικό τομέα όπως αναφέρεται παρακάτω.

- **Τραπεζικές εργασίες:** Φθορά πιστωτικών καρτών, αξιολόγηση πιστώσεων και αιτήσεων δανείων, αξιολόγηση απάτης και κινδύνου, καθώς και παραβάσεις δανείων
- **Επιχειρηματικές αναλύσεις:** Μοντελοποίηση συμπεριφοράς πελατών, κατακερματισμός πελατών, τάση απάτης, έρευνα αγοράς, μείγμα αγορών, δομή αγοράς και μοντέλα για φθορά, αγορά και ανανέωση.
- **Άμυνα:** Αντιτρομοκρατία, αναγνώριση προσώπου, εξαγωγή χαρακτηριστικών, καταστολή θορύβου, διάκριση αντικειμένων, αισθητήρες, σόναρ, επεξεργασία ραντάρ και σήματος εικόνας, αναγνώριση σημάτων / εικόνων, παρακολούθηση στόχων και διαχείριση όπλων.

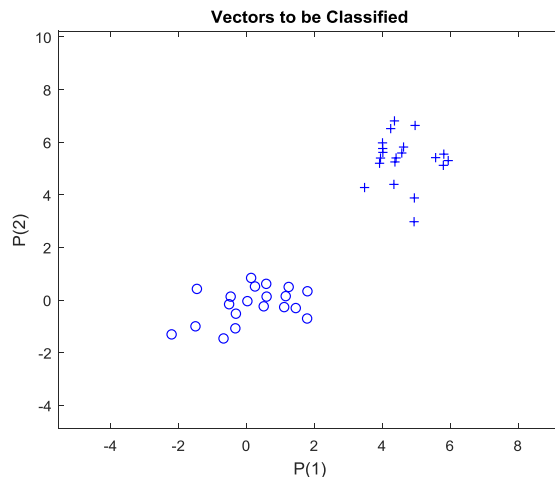
- **Εκπαίδευση:** Προσαρμοστικό λογισμικό μάθησης, δυναμική πρόβλεψη, ανάλυση και πρόβλεψη του εκπαιδευτικού συστήματος, μοντελοποίηση των επιδόσεων των σπουδαστών και διαμόρφωση προφίλ προσωπικότητας.
- **Χρηματοοικονομικά:** Αξιολογήσεις εταιρικών ομολόγων, εταιρική χρηματοοικονομική ανάλυση, ανάλυση χρήσης πιστωτικών γραμμών, πρόβλεψη τιμών νομίσματος, συμβουλές δανείου, έλεγχος υποθηκών, αξιολόγηση ακίνητης περιουσίας και διαπραγμάτευση χαρτοφυλακίου.
- **Ιατρική:** Ανάλυση κυττάρων του καρκίνου, ανάλυση ECG και EEG, παροχή συμβουλών για την εξέταση επειγόντων περιστατικών, μείωση των δαπανών και βελτίωση της ποιότητας για τα νοσοκομειακά συστήματα, βελτιστοποίηση της διαδικασίας μεταμόσχευσης και σχεδιασμός προσθετικών.
- **Χρεόγραφα:** Αυτόματη αξιολόγηση ομολόγων, ανάλυση αγοράς και συμβουλευτικά συστήματα εμπορίας μετοχών.
- **Μεταφορά:** Συστήματα δρομολόγησης, συστήματα διάγνωσης των φρένων φορτηγών και προγραμματισμός οχημάτων.

1.5 Παραδείγματα επίλυσης προβλημάτων με ΤΝΔ

1.5.1 Ταξινόμηση γραμμικά διαχωρίσιμων δεδομένων με Perceptron

Δύο ομάδες δεδομένων, που ανήκουν σε δύο κατηγορίες, ορίζονται σε έναν διδιάστατο χώρο εισόδων. Υποτίθεται ότι οι 2 τάξεις είναι γραμμικά διαχωρίσιμες. Ο στόχος είναι να κατασκευαστεί και να εκπαιδευτεί ένα Perceptron για την ορθή ταξινόμηση άγνωστων δεδομένων εισόδων.

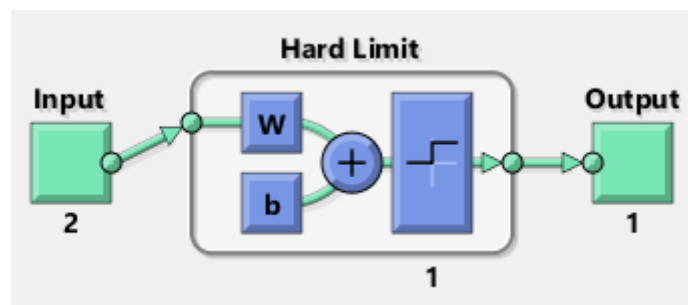
(α) Ορίζονται τα δεδομένα εισόδου και εξόδου για την εκπαίδευση του δικτύου.



Εικόνα 1.5: Δημιουργία δύο ομάδων δεδομένων για διαχωρισμό αυτών.

Το Perceptron συνήθως χρησιμοποιείται για την ταξινόμηση των δεδομένων εισόδου σε δύο τάξεις γραμμικά διαχωρίσιμες. Επομένως, είναι επίσης γνωστό ως γραμμικός δυαδικός ταξινομητής. Το Perceptron με ένα νευρώνα ταξινομεί τα πρότυπα που ανήκουν σε δύο κλάσεις. Οπότε σκοπός του είναι η σωστή ταξινόμηση του συνόλου των εισόδων σε μία από τις δύο κλάσεις.

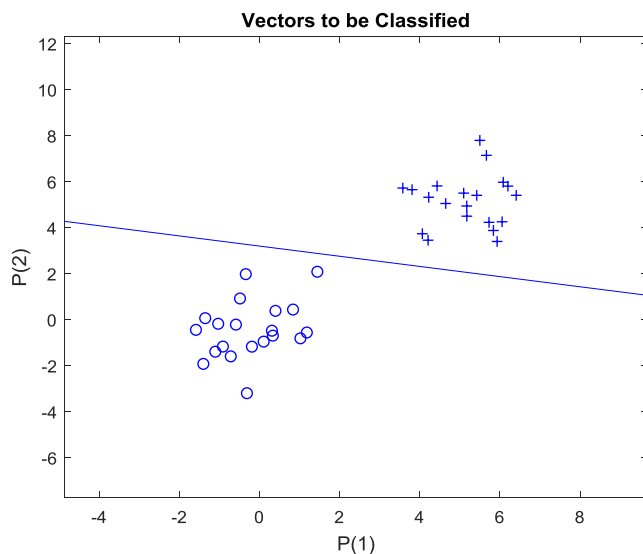
(β) Δημιουργείται και εκπαιδεύεται το Perceptron:



Εικόνα 1.6: Δημιουργία Perceptron για δυαδική ταξινόμηση.

Οι είσοδοι (inputs) πολλαπλασιάζονται με τα βάρη (weights) τους w και στο γινόμενο αθροίζεται και μία σταθερά (offset ή bias). Πολλαπλασιάζει όλες τις τιμές στον αθροιστή και δημιουργεί ένα σταθμισμένο άθροισμα. Έπειτα αυτό το άθροισμα το περνάει από μία συνάρτηση ενεργοποίησης (π.χ. βηματική συνάρτηση στην περίπτωση της παραπάνω εικόνας) και παράγει την έξοδο.

(γ) Σχεδιάζεται το όριο απόφασης (γραμμή διαχωρισμού των 2 τάξεων).



Εικόνα 1.7: Δημιουργία διαχωριστικής γραμμής για επιβεβαίωση σωστής ταξινόμησης.

(δ) Έλεγχος:

Μετά το τέλος της εκπαίδευσης, χρησιμοποιείται η γραμμή διαχωρισμού των 2 τάξεων για τη σωστή ταξινόμηση νέων, άγνωστων δεδομένων εισόδων.

1.5.2 Επίλυση του προβλήματος XOR με Perceptron πολλαπλών στρώσεων

Το Perceptron έχει το βασικό μειονέκτημα ότι δεν μπορεί να λύσει μη γραμμικά διαχωρίσιμα προβλήματα. Στον πραγματικό κόσμο τα περισσότερα προβλήματα είναι μη γραμμικά διαχωρίσιμα και μόνο μία μειονοψηφία προβλημάτων είναι γραμμικά – και αυτά (τα γραμμικά) μπορούν να επιλυθούν και με άλλους ευκολότερους τρόπους εκτός από τα ΤΝΔ και το perceptron.

Ένα απλό, μη γραμμικά διαχωρίσιμο πρόβλημα είναι η συνάρτηση XOR που θα εξεταστεί παρακάτω. Το πρόβλημα αυτό δεν μπορεί να παρασταθεί με ένα δίκτυο ενός μόνο νευρώνα. Το πρόβλημα αυτό οδήγησε στην μελέτη του δικτύου Perceptron πολλών επιπέδων, το οποίο είναι πιο περίπλοκο στην δομή του καθώς περιέχει και περισσότερους νευρώνες.

Στο Perceptron πολλαπλών στρώσεων υπάρχουν περισσότερα από ένα γραμμικά στρώματα (συνδυασμοί νευρώνων). Στο απλό παράδειγμα με το δίκτυο τριών στρωμάτων, το πρώτο στρώμα θα είναι το στρώμα εισόδου και το τελευταίο θα είναι το στρώμα εξόδου, ενώ το μεσαίο στρώμα θα αποκαλείται κρυμμένο στρώμα. Τροφοδοτούνται τα δεδομένα εισόδου στο στρώμα εισόδου και ο χρήστης παίρνει την

τελική έξοδο από το στρώμα εξόδου. Μπορεί να αυξηθεί ο αριθμός των κρυμμένων στρωμάτων όσο χρειάζεται, για να γίνει το μοντέλο («η αρχιτεκτονική του δικτύου») πιο περίπλοκο, ανάλογα με το πρόβλημα που επιλύεται.

Αυτή η αρχιτεκτονική πολλαπλών στρωμάτων είναι πιο πολύπλοκη από αυτή του κλασικού δικτύου Perceptron, αλλά είναι και ικανή να επιτύχει μη γραμμικό διαχωρισμό. Έτσι, με το σωστό σύνολο βαρών, μετά από εκπαίδευση, μπορεί να κάνει σωστή ταξινόμηση με ακρίβεια στις εισόδους XOR.

Το πρόβλημα XOR είναι ένα κλασικό πρόβλημα στην έρευνα των ΤΝΔ. Είναι το πρόβλημα της χρήσης ενός ΤΝΔ για την πρόβλεψη των εξόδων των λογικών πυλών XOR που δέχονται δύο δυαδικές εισόδους. Μια πύλη XOR θα πρέπει να επιστρέψει «TRUE» (π.χ. 1) εάν οι δύο εισοδοί δεν είναι ίσες μεταξύ τους και «FALSE» (π.χ. «0» ή «-1») εάν είναι ίσες μεταξύ τους. Όλες οι δυνατές εισοδοί και οι προβλεπόμενες έξοδοι εμφανίζονται στον Πίνακα Αληθείας της πύλης XOR παρακάτω.

x_1	x_2	$x_1 \text{ XOR } x_2$
-1	-1	-1
-1	1	1
1	-1	1
1	1	-1

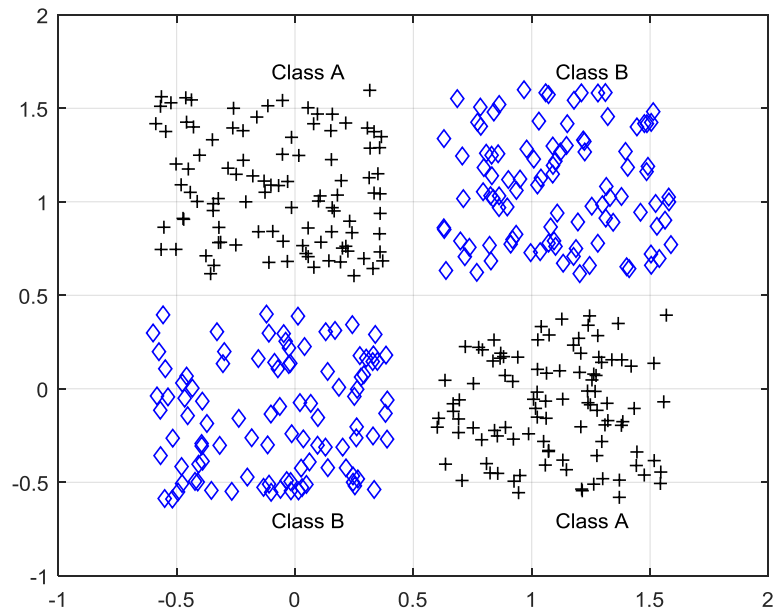
Εικόνα 1.8: Πίνακας Αληθείας της λογικής πράξης Exclusive OR (XOR). True = 1, False = -1.

(Πηγή: http://www.di.unito.it/~cancelli/retineu11_12/FNN.pdf)

Το XOR είναι ένα γνωστό πρόβλημα ταξινόμησης για το οποίο οι αναμενόμενες έξοδοι είναι γνωστές εκ των προτέρων. Επομένως, είναι σκόπιμο να χρησιμοποιηθεί μια προσέγγιση εκπαίδευσης με επίβλεψη (supervised learning) για την εκπαίδευση του ΤΝΔ.

Τέσσερις ομάδες δεδομένων (A, B, C, D) ορίζονται σε έναν 2-διαστάσεων χώρο εισόδου. Τα ζεύγη (A, C) και (B, D) αντιπροσωπεύουν πρόβλημα ταξινόμησης XOR. Ο στόχος είναι να σχεδιαστεί ένα ΤΝΔ για την επίλυση του προβλήματος XOR.

(α) Καθορίζονται τέσσερις ομάδες δεδομένων εισόδου.



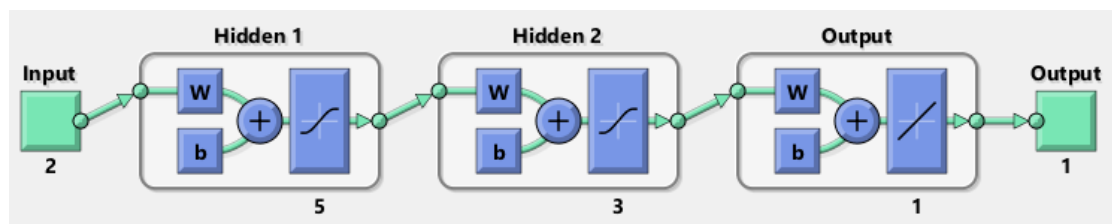
Εικόνα 1.9: Δημιουργία δύο κλάσεων με δύο πρότυπα η καθεμία, για δυαδική κατηγοριοποίηση.

Σε αυτό το γράφημα του XOR υπάρχουν δύο κλάσεις με δύο πρότυπα η καθεμία. Αυτές οι δύο κλάσεις δεν μπορούν να διαχωριστούν με τη χρήση μιας γραμμής οπότε πρέπει να χρησιμοποιηθούν περισσότερες (εδώ, δύο γραμμές, μία οριζόντια και μία κατακόρυφη, που διασταυρώνονται στο μέσον της εικόνας). Το παρακάτω NN πραγματοποιεί αυτό τον μη γραμμικό διαχωρισμό, όπου κάθε κρυμμένος κόμβος περιγράφει μία από τις γραμμές.

(β) Ορίζεται η κωδικοποίηση εξόδου για το πρόβλημα XOR.

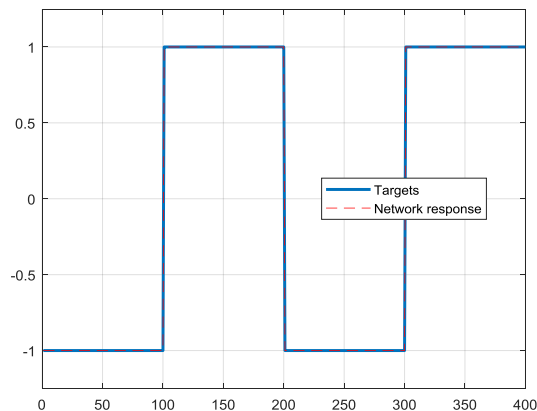
Προετοιμασία εισόδων και εξόδων για εκπαίδευση του δικτύου.

(γ) Δημιουργία και εκπαίδευση ενός Multilayer Perceptron.



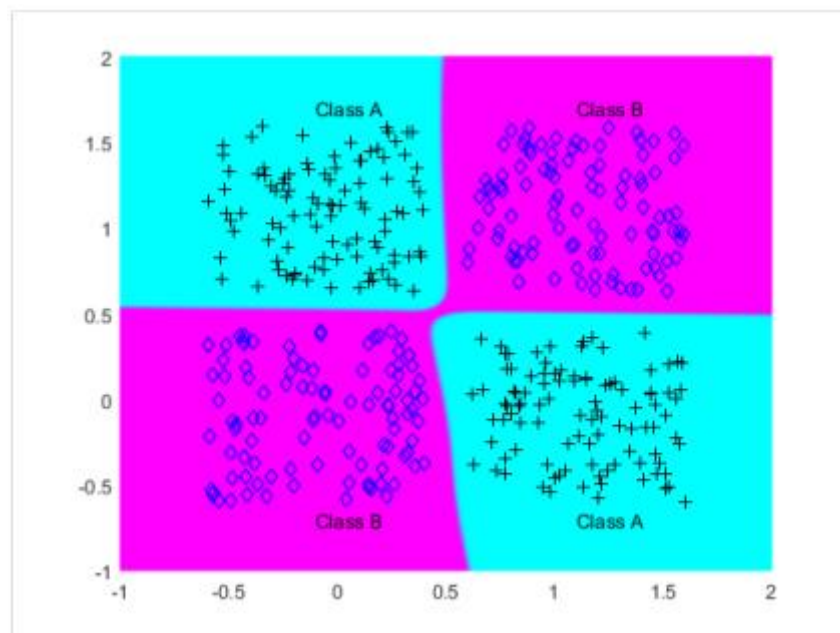
Εικόνα 1.10: Δημιουργία Multilayer Perceptron για την ταξινόμηση.

Σχεδιάζονται μία σειρά από εισόδους και η αντίστοιχη απόκριση (έξοδος) του ΤΝΔ για ναδειχθεί πόσο καλά το ΤΝΔ «μαθαίνει» τα δεδομένα εισόδου.



Εικόνα 1.11: Δημιουργία Targets και Network response για έλεγχο του δίκτυο πάνω στα δεδομένα.

(γ) Χάραξη των διαχωριστικών γραμμών:



Εικόνα 1.12: Δημιουργία διαχωριστικών πεδίων με την βοήθεια των δύο χρωμάτων για την σωστή ταξινόμηση των δύο κλάσεων.

(δ) Έλεγχος:

Σωστή ταξινόμηση άγνωστων δεδομένων εισόδου στις δύο κλάσεις, ανάλογα με τις διαχωριστικές γραμμές που χαράχθηκαν κατά τη φάση εκπαίδευσης.

Κεφάλαιο 2: Βαθιά Μάθηση και Νευρωνικά Δίκτυα Βαθιάς Μάθησης

2.1 Μηχανική μάθηση - Βαθιά μάθηση

Η Μηχανική Μάθηση (Machine Learning, ML) είναι υπο-πεδίο της Επιστήμης των Υπολογιστών που αναπτύχθηκε από την εφαρμογή της αναγνώρισης προτύπων (Pattern Recognition) και της υπολογιστικής θεωρίας μάθησης στην Τεχνητή Νοημοσύνη. Η μηχανική μάθηση αφορά στη μελέτη και την κατασκευή αλγορίθμων που μπορούν να «μαθαίνουν» (εκπαιδεύονται) από δεδομένα και να κάνουν προβλέψεις σχετικά με αυτά. Τέτοιοι αλγόριθμοι λειτουργούν κατασκευάζοντας μοντέλα από πειραματικά δεδομένα, προκειμένου να κάνουν προβλέψεις βασιζόμενες στα δεδομένα ή να εξάγουν αποφάσεις που εκφράζονται ως το αποτέλεσμα. Παρακάτω αναφέρονται μερικοί τρόποι για την επιτυχία αυτών των στόχων.

Τα ΤΝΔ είναι υπολογιστικά μοντέλα εμπνευσμένα από το κεντρικό νευρικό σύστημα των ζώων (βιολογικών) οργανισμών. Έχουν την δυνατότητα μηχανικής μάθησης καθώς αναγνώρισης προτύπων. Αποτελούν συστήματα διασυνδεδεμένων "νευρώνων" που μπορούν να υπολογίζουν τιμές εξόδου από τιμές εισόδου, χρησιμοποιώντας συγκεκριμένα βάρη.

Ο όρος Fuzzy σημαίνει την ασάφεια. Στην πραγματική ζωή, μπορεί να βρεθεί κάποιος σε μια κατάσταση όπου δεν μπορεί να αποφασίσει αν μία δήλωση ή πρόταση είναι αληθινή ή ψευδής, ισχύει ή δεν ισχύει. Δηλώσεις όπως «Οι μαθητές αυτής της τάξης είναι ψηλοί», «αυτή η εικόνα είναι αρκετά καθαρή», «σήμερα οι τιμές είναι ψηλές», είναι παραδείγματα ασαφών δηλώσεων, δηλαδή δηλώσεων που δεν είναι σαφές αν αληθεύουν ή όχι. Χαρακτηριστικό τους είναι η χρήση ποιοτικών όρων περιγραφής των επιθυμητών χαρακτηριστικών ή ιδιοτήτων, σε αντιδιαστολή με τους ποσοτικούς όρους (π.χ. «Οι μαθητές αυτής της τάξης είναι όλοι πάνω από 1.70μ» που με μέτρηση μπορεί εύκολα να γίνει δεκτή ή να απορριφθεί). Η ασαφής λογική (Fuzzy Logic) προσφέρει πολύτιμη ευελιξία στην προσέγγιση και επίλυση προβλημάτων που εκφράζονται με ποιοτικούς όρους και συνεπώς περιέχουν στοιχεία ασάφειας. Μπορεί επίσης να υπολογίσει την αβεβαιότητα (uncertainty) οποιασδήποτε κατάστασης. Ένας αλγόριθμος ασαφούς λογικής βοηθά στην επίλυση ενός προβλήματος αφού εξετάσει

όλα τα διαθέσιμα δεδομένα. Στη συνέχεια, παίρνει την καλύτερη δυνατή απόφαση για την δεδομένη είσοδο. Η μέθοδος Fuzzy Logic μιμείται τον ανθρώπινο τρόπο λήψης αποφάσεων που εξετάζει όλες τις δυνατότητες μεταξύ των ψηφιακών τιμών True και False.

Οι Support Vector Machine (SVM) είναι μία οικογένεια αλγορίθμων μηχανικής μάθησης με επίβλεψη, που μπορούν να χρησιμοποιηθούν τόσο για προβλήματα ταξινόμησης όσο και για παλινδρομήσεις (regression). Ωστόσο, χρησιμοποιούνται κυρίως σε προβλήματα ταξινόμησης.

Στην επιστήμη των υπολογιστών, ένας γενετικός αλγόριθμος (Genetic Algorithm) εμπνέεται από τη διαδικασία φυσικής επιλογής που ισχύει στη Βιολογία για μηχανισμούς αναπαραγωγής των κυττάρων και τις διαδικασίες όπως η μετάλλαξη, η διασταύρωση και η επιλογή. Οι γενετικοί αλγόριθμοι ανήκουν στην ευρύτερη κατηγορία εξελικτικών αλγορίθμων (Evolutionary Algorithms). Οι γενετικοί αλγόριθμοι χρησιμοποιούνται συνήθως για τη δημιουργία λύσεων υψηλής ποιότητας σε προβλήματα βελτιστοποίησης και αναζήτησης, βασιζόμενοι σε βιο-εμπνευσμένους τελεστές όπως η μετάλλαξη, η διασταύρωση και η επιλογή. Οι γενετικοί αλγόριθμοι κωδικοποιούνται μέσω γενετικού προγραμματισμού (Genetic Programming) που ένα υποσύνολο μηχανικής μάθησης. Χρησιμοποιούνται για την ανεύρεση λύσεων σε προβλήματα που οι άνθρωποι δεν ξέρουν πώς να επιλύσουν άμεσα. Λειτουργώντας χωρίς τις ανθρώπινες προκαταλήψεις ή προτιμήσεις, ο προσαρμοστικός χαρακτήρας των Εξελικτικών Αλγορίθμων μπορεί να δημιουργήσει λύσεις συγκρίσιμες και συχνά καλύτερες από τις ανθρώπινες προσπάθειες.

Τόσο η μηχανική μάθηση (ML) όσο και η βαθιά μάθηση (DL) είναι περιοχές της τεχνητής νοημοσύνης, ωστόσο η βαθιά μάθηση αποτελεί ουσιαστικά εξέλιξη της μηχανικής μάθησης.

Στη μηχανική μάθηση, οι αλγόριθμοι που δημιουργούνται από τους ανθρώπους - προγραμματιστές είναι υπεύθυνοι για την ανάλυση των δεδομένων που λαμβάνουν και την εκπαίδευση από αυτά. Με αυτό τον τρόπο καταλήγουν να λαμβάνουν αποφάσεις βάσει των «κανόνων» που έμαθαν από τα δεδομένα με τα οποία εκπαιδεύθηκαν.

Η βαθιά εκμάθηση, από την άλλη πλευρά, εκπαιδεύει ένα ΤΝΔ που λειτουργεί κατ' αναλογία με τον ανθρώπινο εγκέφαλο, επιτρέποντας στη μηχανή (machine) να

αναλύει δεδομένα με μια συγκεκριμένη διαδικασία, περίπου όπως κάνουν οι άνθρωποι. Οι μηχανές (αλγόριθμοι) βαθιάς μάθησης δεν απαιτούν από έναν άνθρωπο - προγραμματιστή να τους πει τι να κάνουν με τα δεδομένα εισόδου (πώς να τα επεξεργαστούν) ώστε να αρχίσουν να μαθαίνουν από αυτά. Η εκπαίδευση πραγματοποιείται απευθείας από την εξαιρετικά μεγάλη σε όγκο και ποικιλία ποσότητα δεδομένων που συλλέγουν και καταναλώνουν οι ίδιοι - τα δεδομένα δηλαδή είναι το καύσιμο για την λειτουργία των μοντέλων βαθιάς μάθησης.

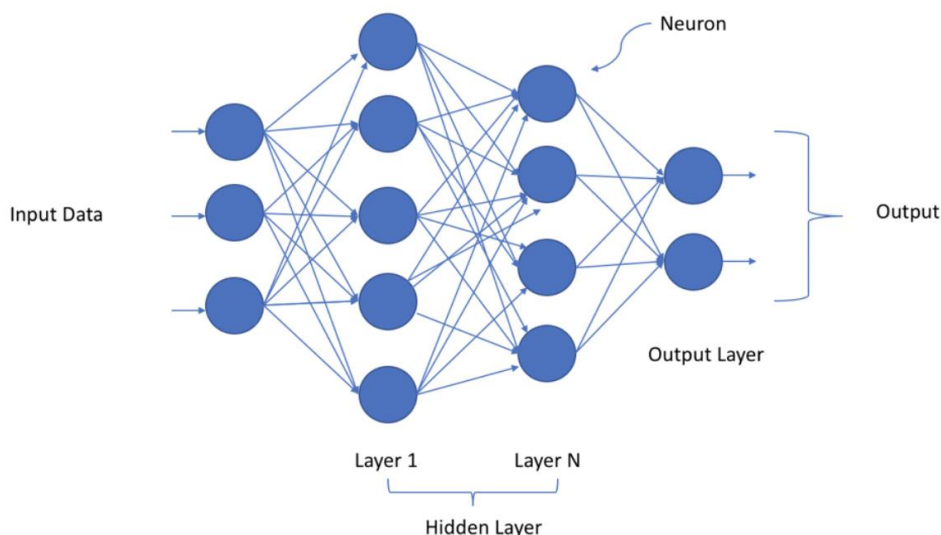
Η βαθιά μάθηση αντιπροσωπεύει την αιχμή της τεχνητής νοημοσύνης (AI). Αντί να διδάσκει τους υπολογιστές πώς επεξεργάζονται δεδομένα και στη συνέχεια πώς να μαθαίνουν από τα επεξεργασμένα δεδομένα, όπως λειτουργεί η μηχανική μάθηση, στη βαθιά μάθηση ο υπολογιστής εκπαιδεύεται να μαθαίνει απευθείας από τα δεδομένα καθώς τα επεξεργάζεται.

Όλα αυτά είναι δυνατά χάρη στα πολλά στρώματα των βαθιών ΤΝΔ, των ΒΝΔ. Ένα ANN που αποτελείται από περισσότερα από τρία (3) επίπεδα, δηλαδή ένα στρώμα εισόδου, ένα στρώμα εξόδου και πολλαπλά κρυμμένα στρώματα, ονομάζεται βαθύ νευρωνικό δίκτυο και αυτό είναι που στηρίζει τη βαθιά μάθηση. Ένα σύστημα βαθιάς μάθησης είναι αυτοδίδακτο και μαθαίνει φιλτράροντας τις πληροφορίες που προωθούνται από την είσοδο μέσω των πολλαπλών κρυφών στρωμάτων του, με τρόπο παρόμοιο με τον άνθρωπο.

Συμπερασματικά, όλοι αυτοί οι όροι και οι μέθοδοι είναι στενά συνδεδεμένα μεταξύ τους, καθώς το ένα εξαρτάται από το άλλο για να λειτουργήσουν χωρίς νευρωνικά δίκτυα δεν θα υπήρχε βαθιά μάθηση.

2.2 Νευρωνικά Δίκτυα Βαθιάς Μάθησης

Μια απλοποιημένη έκδοση του Deep Neural Network (DNN) είναι μια ιεραρχική (στρωματοποιημένη) οργάνωση νευρώνων (παρόμοια με τους νευρώνες στον εγκέφαλο) με συνδέσεις με άλλους νευρώνες. Αυτοί οι νευρώνες μεταδίδουν ένα μήνυμα ή ένα σήμα σε άλλους νευρώνες με βάση την εισερχόμενη εισροή και σχηματίζουν ένα σύνθετο δίκτυο που μαθαίνει με κάποιο μηχανισμό ανάδρασης. Το παρακάτω διάγραμμα αναπαριστά ένα βαθύ νευρωνικό δίκτυο N στρωμάτων (ένα N-Layered Deep Neural Network).



Εικόνα 2.1: Ένα βαθύ νευρωνικό δίκτυο με N κρυφά στρώματα.

(Πηγή: <https://towardsdatascience.com/a-laymans-guide-to-deep-neural-networks-ddcea24847fb>)

Όπως καταγράφεται στο παραπάνω σχήμα, τα δεδομένα εισόδου καταναλώνονται από τους νευρώνες στο πρώτο στρώμα (στρώμα εισόδου) το οποίο στη συνέχεια παρέχει μια έξοδο στους νευρώνες εντός του επόμενου στρώματος (1ο κρυμμένο στρώμα) και ούτω καθεξής έως και την τελική έξοδο. Η έξοδος μπορεί να είναι μια δυαδική πρόβλεψη όπως Ναι ή Όχι, μία πιθανότητα (αριθμός μεταξύ 0 και 1), μία κλάση (ακέραιος αριθμός) ή η τιμή μίας συνάρτησης (δεκαδικός αριθμός).

Κάθε στρώμα μπορεί να έχει έναν ή περισσότερους νευρώνες και καθένας από αυτούς θα υπολογίσει μια μικρή λειτουργία δηλαδή τη λειτουργία ενεργοποίησης. Η λειτουργία ενεργοποίησης μιμείται το σήμα για να περάσει στους επόμενους συνδεδεμένους νευρώνες. Εάν οι εισερχόμενοι νευρώνες έχουν ως αποτέλεσμα μια τιμή μεγαλύτερη από ένα κατώφλι τότε η έξοδος προχωράει αλλιώς αγνοείται.

Η σύνδεση μεταξύ δύο νευρώνων διαδοχικών στρώσεων έχει σχετικό βάρος. Το βάρος καθορίζει την επίδραση της δεδομένης εισόδου στην έξοδο για τον επόμενο νευρώνα και τελικά για τη συνολική τελική έξοδο. Σε ένα ΒΝΔ, τα αρχικά βάρη ορίζονται τυχαία αλλά κατά τη διάρκεια της εκπαίδευσής του, τα βάρη διορθώνουν διαδοχικώς τις τιμές τους, μέχρι να ολοκληρωθεί η εκπαίδευση, δηλαδή να μάθουν να παράγουν τη σωστή έξοδο για κάθε δεδομένη είσοδο μέσα στο σύνολο εισόδων εκπαίδευσης (training set). Αποσυνθέτοντας το δίκτυο, βλέπουμε ότι αποτελείται από λίγα λογικά δομικά στοιχεία όπως ο νευρώνας, το στρώμα, το βάρος, την είσοδο, την έξοδο, μια συνάρτηση ενεργοποίησης (γραμμική ή μη γραμμική) και τέλος έναν

μηχανισμό μάθησης (βελτιστοποιητής) που θα βοηθήσει το BND να ενημερώσει σταδιακά τις τιμές των βαρών ώστε να επιτύχει σωστή πρόβλεψη του αποτελέσματος (σωστή έξοδο για δεδομένη είσοδο).

Ο όρος "ρηγά" νευρωνικά δίκτυα χρησιμοποιείται για να περιγράψει το TNΔ που έχει μόνο ένα κρυμμένο στρώμα σε αντίθεση με τα «βαθιά» νευρωνικά δίκτυα (BND) που έχουν αρκετά κρυμμένα στρώματα, συχνά διαφορετικών τύπων το καθένα. Υπάρχουν δημοσιεύσεις που υπογραμμίζουν ότι ένα BND με τη σωστή αρχιτεκτονική επιτυγχάνει καλύτερα αποτελέσματα από τα ρηγά TNΔ που έχουν την ίδια υπολογιστική ισχύ (π.χ. αριθμό νευρώνων ή συνδέσεων). Η κύρια εξήγηση είναι ότι τα βαθιά μοντέλα είναι σε θέση να εξαγάγουν και να κατασκευάσουν καλύτερα χαρακτηριστικά από ότι τα ρηγά μοντέλα και για να το επιτύχουν χρησιμοποιούν τα (περισσότερα) ενδιάμεσα κρυμμένα στρώματα.

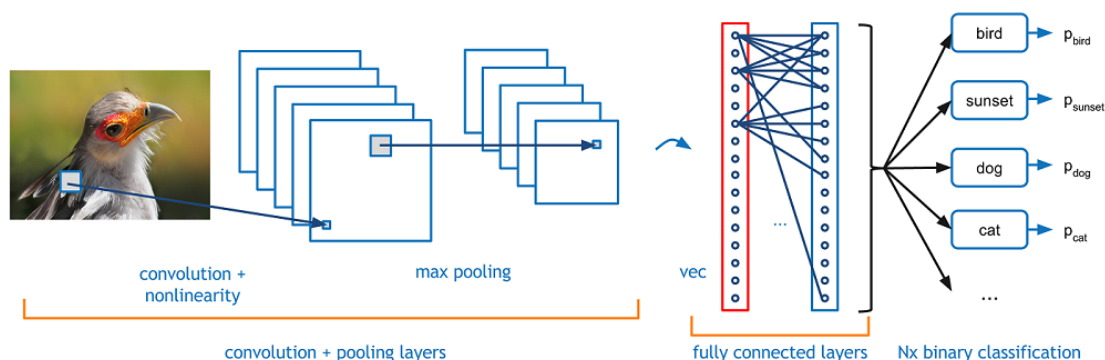
2.3 Συνελικτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks)

Τα Convolutional Neural Networks (CNN) είναι μια κατηγορία βαθιών νευρωνικών δικτύων, και συγχρόνως είναι αλγόριθμοι βαθιάς μάθησης που μπορούν να αναγνωρίσουν και να ταξινομήσουν χαρακτηριστικά σε εικόνες για υπολογιστική όραση. Ένα CNN είναι ένα νευρωνικό δίκτυο πολλαπλών επιπέδων που έχει σχεδιαστεί για να αναλύει τις οπτικές εισόδους και να εκτελεί εργασίες όπως ταξινόμηση εικόνων, κατάτμηση εικόνων και ανίχνευση και εντοπισμός αντικειμένων μέσα στην εικόνα – λειτουργίες με πολλές και χρήσιμες εφαρμογές, όπως π.χ. στην οδήγηση μη επανδρωμένων οχημάτων κ.α.

Τα CNN μπορούν επίσης να χρησιμοποιηθούν για εφαρμογές βαθιάς μάθησης στην υγειονομική περίθαλψη, όπως η ιατρική απεικόνιση. Αυτά μοντελοποιούνται σύμφωνα με τον οπτικό φλοιό στον ανθρώπινο εγκέφαλο και τυπικά χρησιμοποιούνται για εργασίες οπτικής επεξεργασίας. Όταν εισαχθεί σ' αυτά μια εικόνα, μπορούν να αναγνωρίσουν και να διαφοροποιήσουν τα σημαντικά χαρακτηριστικά της εικόνας, διασπώντας τη σε μικρά κομμάτια.

Τα CNN μειώνουν το μέγεθος της εικόνας χωρίς να χάσουν τα βασικά χαρακτηριστικά, ώστε να μπορούν να την επεξεργαστούν ευκολότερα. Τα κρυμμένα στρώματα σε ένα CNN ονομάζονται συνελικτικά στρώματα. Η ικανότητά τους για φιλτράρισμα αυξάνει σε πολυπλοκότητα από στρώμα σε στρώμα. Τα πρώτα στρώματα συνέλιξης προσδιορίζουν απλά μοτίβα ενώ τα επόμενα στρώματα

συνδυάζουν τα απλά μοτίβα σε σύνθετα σύνολα. Χρησιμοποιούνται συνήθως για ανάλυση οπτικών εικόνων. Μια από τις κύριες εφαρμογές τους είναι στην ταξινόμηση εικόνων.



Εικόνα 2.2: Ταξινόμηση εικόνων για ένα πουλί.

(Πηγή: <https://heartbeat.fritz.ai/a-research-guide-to-convolution-neural-networks-a4589b6ca9bd>)

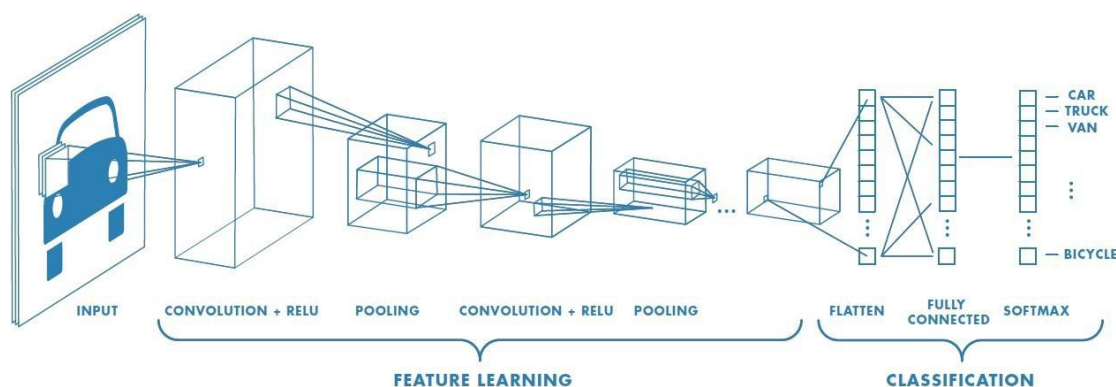
Η ταξινόμηση εικόνας είναι η διαδικασία που λαμβάνει μια εικόνα ως είσοδο και εξάγει την κλάση (κατηγορία) της εικόνας, μεταξύ ενός συνόλου προκαθορισμένων κατηγοριών (π.χ. σε ένα πρόβλημα κατηγοριοποίησης σε 3 κλάσεις, είτε είναι αυτοκίνητο, είτε ποδήλατο, είτε πουλί) ή εξάγει την πιθανότητα η εικόνα να ανήκει σε καθεμία από τις τάξεις αυτές.

Όταν μια εικόνα εισάγεται σε ένα μοντέλο CNN, το μοντέλο τη βλέπει ως μια σειρά από τιμές εικονοστοιχείων (pixels) ανάλογα με την ανάλυση και το μέγεθος της εικόνας. Για παράδειγμα, για μια έγχρωμη εικόνα JPG με διαστάσεις 640 * 640 pixels, ο αντιπροσωπευτικός πίνακας εισόδου του υπολογιστή αυτής της εικόνας θα είναι 640 * 640 * 3, όπου 3 είναι οι τιμές RGB.

Κάθε ένας από τους παραπάνω αριθμούς (pixel) έχει μια τιμή μεταξύ 0 και 255, η οποία περιγράφει την φωτεινή ένταση των εικονοστοιχείων στο συγκεκριμένο σημείο. Με μια τέτοια εισαγωγή, ο υπολογιστής εξάγει έπειτα αριθμούς που περιγράφουν την πιθανότητα μιας εικόνας που ανήκει σε μια δεδομένη κλάση, όπως 0.75 για ένα σκυλί, 0.15 για ένα πουλί, 0.10 για μια γάτα κ.λπ.

Τα CNN έχουν πολλές εφαρμογές στον πραγματικό κόσμο, όπως ανωνυμοποίηση προσώπου σε εικόνα, ανάλυση χειρογράφου, ανάλυση ιατρικής εικόνας, διαφήμιση και πολλά άλλα. Χρησιμοποιούνται επίσης και για άλλες εργασίες στην υπολογιστική όραση, όπως κατάτμηση εικόνας και μεταφορά στυλ.

Η αρχιτεκτονική CNN εμπνέεται από την οργάνωση και τη λειτουργικότητα του οπτικού φλοιού και έχει σχεδιαστεί για να μιμείται το μοτίβο σύνδεσης των νευρώνων μέσα στον ανθρώπινο εγκέφαλο. Οι νευρώνες μέσα σε ένα CNN χωρίζονται σε μια τρισδιάστατη δομή, με κάθε σύνολο νευρώνων να αναλύει μια μικρή περιοχή ή ένα χαρακτηριστικό της εικόνας. Με άλλα λόγια, κάθε ομάδα νευρώνων ειδικεύεται στην αναγνώριση ενός μέρους της εικόνας. Τα CNN χρησιμοποιούν τις προβλέψεις από τα στρώματα για να παράγουν μια τελική έξοδο που παρουσιάζει ένα διάνυσμα βαθμολογίας πιθανότητας για να αναπαραστήσει την πιθανότητα ένα συγκεκριμένο χαρακτηριστικό να ανήκει σε μια συγκεκριμένη κλάση. Τα CNN έχουν τέσσερα στρώματα, όπως φαίνεται παρακάτω.



Εικόνα 2.3: Δομή του CNN με τα τέσσερα στρώματα.

(Πηγή: <https://heartbeat.fritz.ai/a-research-guide-to-convolution-neural-networks-a4589b6ca9bd>)

1. Convolution

Αυτό είναι πάντα το πρώτο στρώμα ενός CNN. Περιλαμβάνει ένα σύνολο ανεξάρτητων φίλτρων που αρχικοποιούνται τυχαία και γίνονται οι παράμετροί μας, τις οποίες το δίκτυο μαθαίνει από τα επόμενα στρώματα. Ανιχνεύουν χαρακτηριστικά χαμηλού επιπέδου όπως άκρες και καμπύλες.

2. ReLU

Αυτό το στρώμα χρησιμοποιείται με το στρώμα συνέλιξης. Με αυτό το στρώμα, εφαρμόζεται μια λειτουργία ενεργοποίησης στους χάρτες χαρακτηριστικών για να αυξάνεται η μη γραμμικότητα στο δίκτυο, καθώς οι περισσότερες εικόνες είναι κατά σημαντικό μέρος μη γραμμικές.

3. Pooling

Το στρώμα συγκέντρωσης μειώνει το χωρικό μέγεθος της αναπαράστασης εισόδου, το οποίο στη συνέχεια μειώνει τον αριθμό παραμέτρων και υπολογισμών στο δίκτυο. Αυτό διευκολύνει την ανίχνευση αντικειμένων σε μια εικόνα, ανεξάρτητα από το πού βρίσκονται. Η συγκέντρωση βοηθά στον έλεγχο της υπερφόρτωσης, έτσι ώστε το δίκτυο να μπορεί να συλλάβει ανεπιθύμητα συγκεκριμένα στοιχεία σε μια εικόνα.

4. Fully Connected

Σε αυτό το στρώμα, προστίθενται τεχνητά νευρωνικά δίκτυα, τα οποία συνδυάζουν τα χαρακτηριστικά εικόνας μας σε περισσότερες ιδιότητες, καθιστώντας ευκολότερη την ακριβέστερη πρόβλεψη της κλάσης της εικόνας μας. Στη συνέχεια, εμφανίζεται το Backpropagation σε αυτό το στρώμα, όπου τα λάθη ελαχιστοποιούνται και τα βάρη ρυθμίζονται για να βελτιστοποιούν την απόδοση του μοντέλου μας.

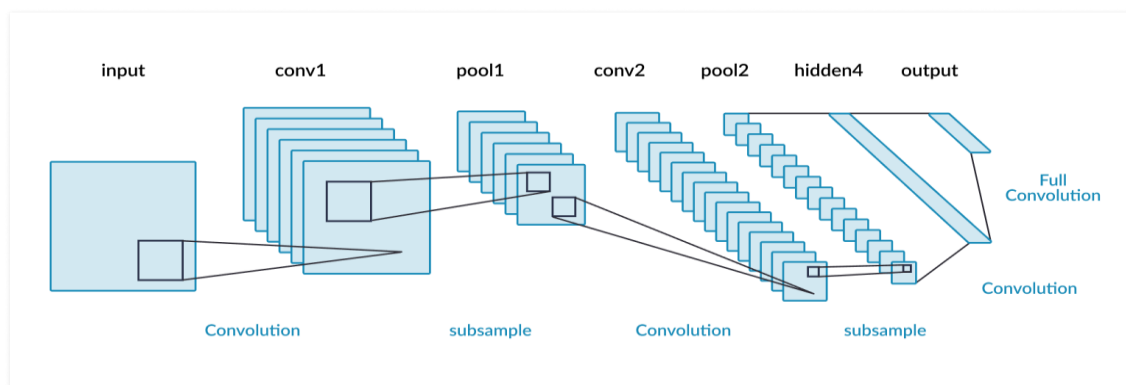
Μια κλασική δομή CNN θα μοιάζει ως εξής:

Input -> Convolution -> ReLU -> Convolution -> ReLU -> Pooling -> ReLU -> Convolution -> ReLU -> Pooling -> Fully Connected

2.3.1 Δημοφιλείς αρχιτεκτονικές CNN

LeNet-5 (1998)

Αυτό το CNN 7 στρωμάτων ταξινομεί χειρόγραφα ψηφία, ψηφιοποιημένα με 32×32 εικονοστοιχεία εισόδου το καθένα, σε κλίμακες του γκρι. Χρησιμοποιήθηκε από πολλές τράπεζες για να αναγνωρίσει τους χειρόγραφους αριθμούς στους ελέγχους.

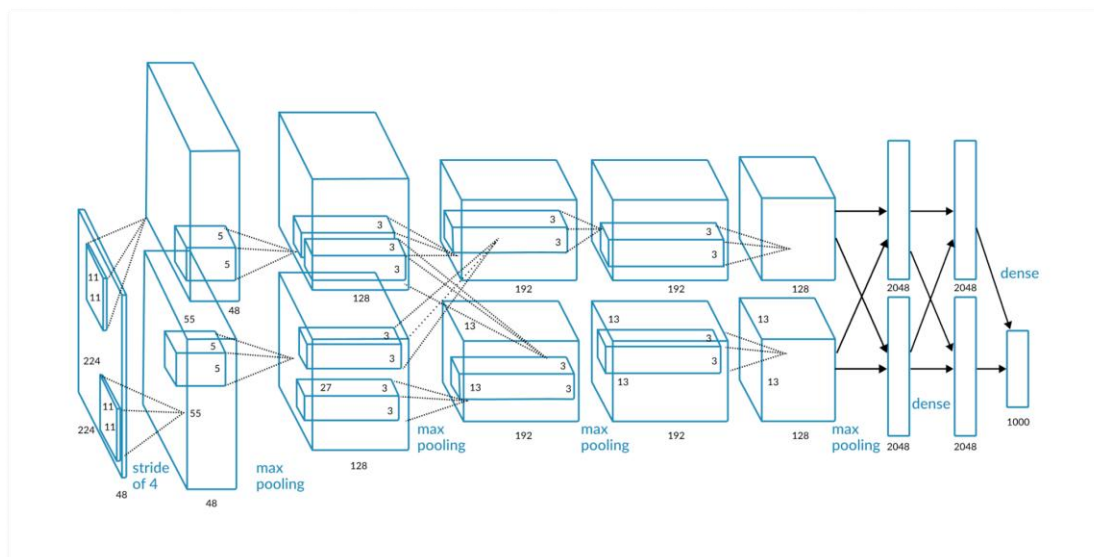


Εικόνα 2.4: Παράδειγμα CNN τύπου LeNet-5 (1998) με επτά στρώματα.

(Πηγή: <https://missinglink.ai/guides/convolutional-neural-networks/convolutional-neural-network-architecture-forging-pathways-future/>)

AlexNet (2012)

Το AlexNet έχει σχεδιαστεί από τον όμιλο SuperVision, με παρόμοια αρχιτεκτονική με το LeNet, αλλά με περισσότερα φίλτρα ανά στρώμα καθώς και στοιβαγμένα στρώματα συνέλιξης. Αποτελείται από πέντε συνελκτικά στρώματα ακολουθούμενα από τρία πλήρως συνδεδεμένα στρώματα. Μία από τις σημαντικότερες διαφορές μεταξύ των αλγορίθμων ανίχνευσης αντικειμένων AlexNet και άλλων αντικειμένων είναι η χρήση της ReLU για το μη γραμμικό μέρος αντί για τη συνάρτηση Sigmoid ή την συνάρτηση tanh που χρησιμοποιούν τα παραδοσιακά νευρωνικά δίκτυα. Το AlexNet αξιοποιεί την ταχύτερη εκπαίδευση της ReLU για να επιταχύνει τον αλγόριθμό του. Οι δημιουργοί του AlexNet χώρισαν το δίκτυό τους σε δύο αγωγούς επειδή χρησιμοποίησαν δύο μονάδες επεξεργασίας γραφικών Nvidia Geforce GTX 580 (GPU) για να εκπαιδεύσουν το CNN τους.



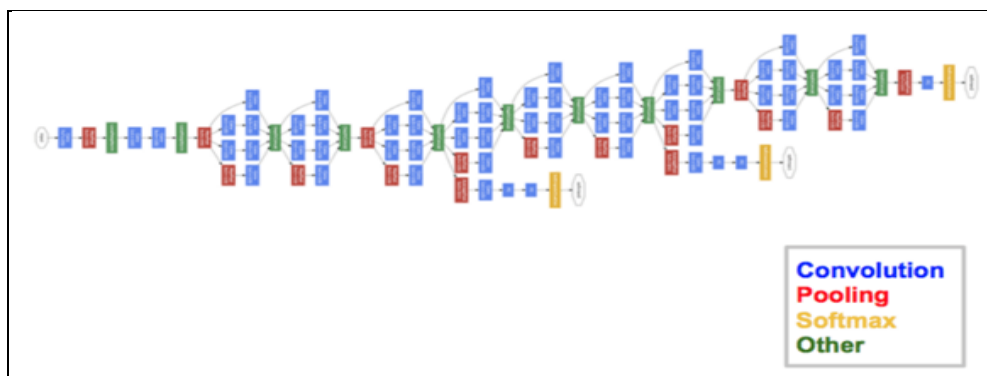
Εικόνα 2.5: Παράδειγμα CNN τύπου AlexNet (2012).

(Πηγή: <https://missinglink.ai/guides/convolutional-neural-networks/convolutional-neural-network-architecture-forging-pathways-future/>)

GoogleNet (2014)

Χτισμένο ως ένα CNN εμπνευσμένο από το LeNet, το δίκτυο GoogleNet, το οποίο ονομάζεται επίσης Inception V1, δημιουργήθηκε από μια ομάδα της Google. Το GoogleNet ήταν ο νικητής του διαγωνισμού ILSVRC 2014 και πέτυχε ένα ποσοστό σφαλμάτων top-5 μικρότερο από 7%, το οποίο είναι κοντά στο επίπεδο της ανθρώπινης απόδοσης. Η αρχιτεκτονική του GoogleNet αποτελείται από ένα βαθύ

CNN 22 βαθμίδων, χρησιμοποιούσε μια μονάδα βασισμένη σε μικρές συσπειρώσεις, που ονομάζεται "αρχική ενότητα", η οποία χρησιμοποίησε κανονικοποίηση παρτίδων, RMSprop και εικόνα για να μειώσει τον αριθμό των παραμέτρων από 60 εκατομμύρια, όπως στο AlexNet, σε μόλις 4 εκατομμύρια.

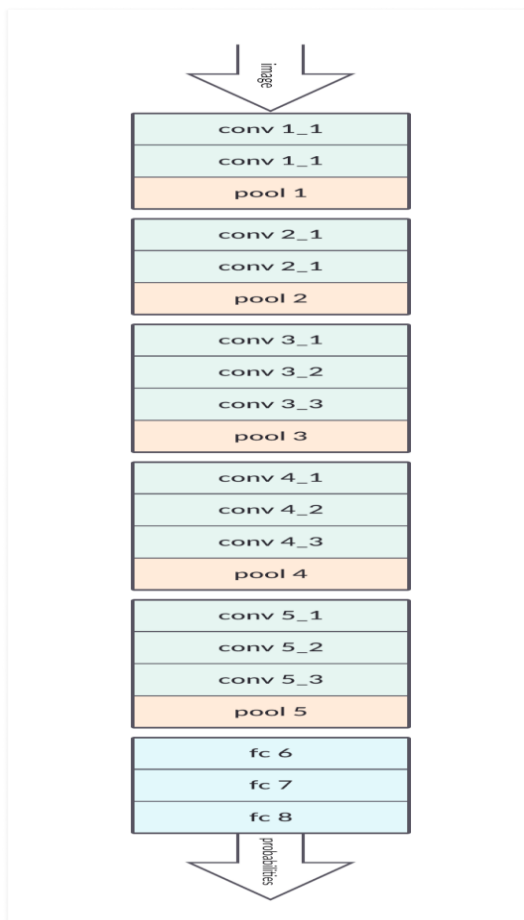


Εικόνα 2.6: Παράδειγμα CNN τύπου GoogleNet (2014).

(Πηγή: <https://missinglink.ai/guides/convolutional-neural-networks/convolutional-neural-network-architecture-forging-pathways-future/>)

VGGNet (2014)

Το VGGNet, που έλαβε μέρος στο διαγωνισμό ILSVRC 2014, αποτελείται από 16 συνθετικά στρώματα. Παρόμοια με το AlexNet, χρησιμοποίησε μόνο 3×3 συνελίξεις αλλά πρόσθεσε περισσότερα φίλτρα. Το VGGNet εκπαιδεύτηκε σε 4 GPU για περισσότερο από δύο εβδομάδες για να επιτύχει την απόδοσή του. Το πρόβλημα με το VGGNet είναι ότι αποτελείται από 138 εκατομμύρια παραμέτρους, 34,5 φορές περισσότερο από το GoogleNet, πράγμα που καθιστά δύσκολη την (επαν-)εκπαίδευσή του.



Εικόνα 2.7: Παράδειγμα CNN τύπου VGGNet (2014).

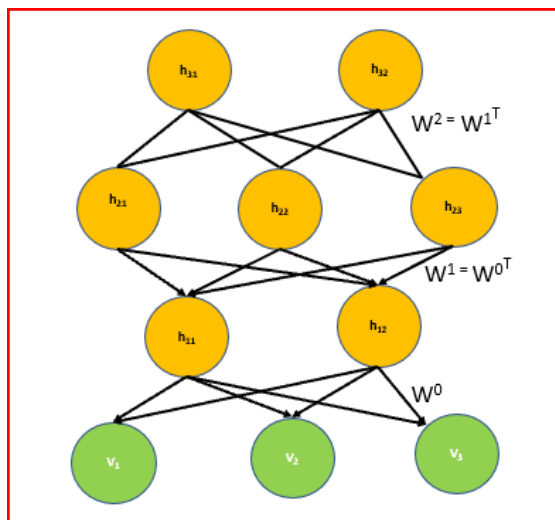
(Πηγή: <https://missinglink.ai/guides/convolutional-neural-networks/convolutional-neural-network-architecture-forging-pathways-future/>)

2.4 Deep Belief Neural Networks

Τα DBNN είναι μια άλλη κατηγορία βαθιών νευρωνικών δικτύων τα οποία έχουν διαμορφωθεί μιμούμενα τον ανθρώπινο εγκέφαλο, πράγμα που τους έδωσε μεγαλύτερη ικανότητα να αναγνωρίζουν πρότυπα και να επεξεργάζονται πολύπλοκες πληροφορίες. Είναι αλγόριθμοι που χρησιμοποιούν τις πιθανότητες και την μάθηση χωρίς επίβλεψη για την παραγωγή εξόδων. Αποτελούνται από δυαδικές λανθάνουσες μεταβλητές και περιέχουν τόσο μη κατευθυνόμενα στρώματα όσο και κατευθυνόμενα στρώματα.

Σε αντίθεση με άλλα μοντέλα, κάθε στρώμα στα DBNN μαθαίνει ολόκληρη την είσοδο. Στα CNN, τα πρώτα επίπεδα φιλτράρουν μόνο εισόδους για βασικά χαρακτηριστικά, όπως οι άκρες, και τα μεταγενέστερα στρώματα ανασυνδυάζουν όλα τα απλά πρότυπα που βρέθηκαν από τα προηγούμενα στρώματα. Τα DBNN, από την

άλλη πλευρά, δουλεύουν γενικά κάθε επίπεδο και ρυθμίζουν κάθε στρώμα με τη σειρά.



Εικόνα 2.8: Αρχιτεκτονική DBNN.

(Πηγή: <https://missinglink.ai/guides/neural-network-concepts/deep-belief-networks-work-applications/>)

Το δίκτυο είναι σαν μια στοίβα από Reduced Boltzmann Machines (RBM), όπου οι κόμβοι σε κάθε στρώμα συνδέονται με όλους τους κόμβους του προηγούμενου και του επόμενου στρώματος. Ωστόσο, σε αντίθεση με τους RBM, οι κόμβοι σε ένα DBNN δεν επικοινωνούν πλευρικά στο στρώμα τους. Ένα δίκτυο συμμετρικών βαρών συνδέει διαφορετικά στρώματα.

Οι συνδέσεις στα επάνω στρώματα είναι μη κατευθυντικές και η συνειρμική μνήμη σχηματίζεται από τις συνδέσεις μεταξύ τους. Οι συνδέσεις στα κατώτερα επίπεδα είναι κατευθυντικές. Οι κόμβοι στο κρυφό στρώμα εκπληρώνουν δύο ρόλους: λειτουργούν ως κρυμμένο στρώμα στους κόμβους που προηγούνται και ως ορατά στρώματα στους κόμβους που ακολουθούν. Αυτοί οι κόμβοι προσδιορίζουν τις συσχετίσεις στα δεδομένα.

Οι αλγόριθμοι εκπαίδευσης τύπου Greedy learning χρησιμοποιούνται για την προ-εκπαίδευση των DBNN. Αυτή είναι μια προσέγγιση επίλυσης προβλημάτων η οποία περιλαμβάνει την πραγματοποίηση της βέλτιστης επιλογής σε κάθε στρώμα στην ακολουθία, βάζοντας τελικά ένα ολικό βέλτιστο. Αυτοί οι αλγόριθμοι ξεκινούν από το κάτω στρώμα και μετακινούνται προς τα επάνω, ρυθμίζοντας με ακρίβεια τα

γενετικά βάρη. Η μάθηση λαμβάνει χώρα σε επίπεδο στρώματος, πράγμα που σημαίνει ότι τα στρώματα των DBN εκπαιδεύονται μία φορά το καθένα.

Επομένως, κάθε στρώμα λαμβάνει επίσης μια διαφορετική έκδοση των δεδομένων και κάθε στρώμα χρησιμοποιεί την έξοδο από το προηγούμενο στρώμα ως είσοδο. Τέλος, χρησιμοποιούνται για την κατάρτιση των DBNN επειδή είναι γρήγοροι και αποδοτικοί. Επιπλέον, συμβάλλουν στη βελτιστοποίηση των βαρών σε κάθε στρώμα.

2.4.1 Εφαρμογές των DBNN

Αναγνώριση εικόνων

Τα DBN μπορούν να χρησιμοποιηθούν στην αναγνώριση εικόνων. Μια εικόνα θα είναι η είσοδος, και η κατηγορία θα είναι η έξοδος. Αυτή η τεχνολογία έχει ευρείες εφαρμογές, που κυμαίνονται από σχετικά απλά καθήκοντα όπως η οργάνωση φωτογραφιών έως κρίσιμες λειτουργίες όπως ιατρικές διαγνώσεις. Για παράδειγμα, οι έξυπνοι μικροσπόροι που μπορούν να πραγματοποιήσουν αναγνώριση εικόνας θα μπορούσαν να χρησιμοποιηθούν για την ταξινόμηση παθογόνων μικροοργανισμών. Αυτό θα αμβλύνει την εξάρτηση από σπάνιους ειδικούς κατά τη διάρκεια σοβαρών επιδημιών, μειώνοντας τον χρόνο απόκρισης.

Αναγνώριση βίντεο

Η αναγνώριση βίντεο χρησιμοποιεί επίσης στα DBN. Η αναγνώριση βίντεο λειτουργεί όμοια με την όραση, καθώς βρίσκει νόημα στα δεδομένα βίντεο. Για παράδειγμα, μπορεί να εντοπίσει ένα αντικείμενο ή μια χειρονομία ενός ατόμου. Μπορεί να χρησιμοποιηθεί σε πολλούς διαφορετικούς τομείς όπως αυτοματοποίηση του σπιτιού, ασφάλεια και υγειονομική περίθαλψη.

Δεδομένα δέσμευσης κινήσεων

Τα δεδομένα καταγραφής κίνησης περιλαμβάνουν την παρακολούθηση της κίνησης αντικειμένων ή ανθρώπων και επίσης χρησιμοποιεί στα DBN. Η σύλληψη της κίνησης είναι δύσκολη επειδή μια μηχανή μπορεί να χάσει γρήγορα το ίχνος, για παράδειγμα, ενός ατόμου ή ενός άλλου προσώπου που μοιάζει παρόμοιο, μπαίνει στο πλαίσιο ή αν κάτι παρεμποδίζει προσωρινά την άποψή του. Συνεπώς, η σύλληψη κίνησης βασίζεται όχι μόνο σε αυτό που μοιάζει με ένα αντικείμενο ή πρόσωπο, αλλά και στην ταχύτητα και την απόσταση. Η δέσμευση κίνησης χρησιμοποιείται ευρέως στην ανάπτυξη βιντεοπαιχνιδιών και στην κινηματογραφική παραγωγή.

Κεφάλαιο 3: Βελτίωση Εικόνας με χρήση Φίλτρων

3.1 Η βελτίωση εικόνας (image enhancement)

Η βελτίωση εικόνας είναι η διαδικασία βελτίωσης της ποιότητας και του περιεχομένου πληροφοριών των αρχικών δεδομένων πριν από την επεξεργασία. Η τεχνική αυτή χρησιμοποιείται για να βελτιώσει μια δεδομένη εικόνα, έτσι ώστε τα επιθυμητά χαρακτηριστικά της εικόνας να γίνουν ευκολότερα αντιληπτά για το ανθρώπινο οπτικό σύστημα ή πιο πιθανό να εντοπιστούν από αυτοματοποιημένα συστήματα ανάλυσης εικόνας. Η βελτίωση της εικόνας επιτρέπει στον παρατηρητή να βλέπει λεπτομέρειες σε εικόνες που μπορεί να μην είναι άμεσα παρατηρήσιμες στην αρχική εικόνα.

Οι μέθοδοι βελτίωσης εικόνας ταξινομούνται συχνά σε λειτουργίες σημείου και χωρικές λειτουργίες. Οι λειτουργίες σημείου είναι μη γραμμικές λειτουργίες και περιλαμβάνουν τέντωμα της αντίθεσης, αποκοπή θορύβου, τροποποίηση ιστογράμματος και ψευδοχρωματισμό. Οι χωρικές λειτουργίες που χρησιμοποιούνται στην επεξεργασία εικόνας σήμερα, από την άλλη πλευρά, είναι συνήθως γραμμικές λειτουργίες. Ο λόγος για αυτό είναι ότι οι χωρικές γραμμικές λειτουργίες είναι απλές και εφαρμόζονται εύκολα. Αν και τα εργαλεία γραμμικής βελτίωσης εικόνας είναι συχνά αρκετά σε πολλές εφαρμογές, σημαντικά πλεονεκτήματα στη βελτίωση της εικόνας μπορούν να επιτευχθούν εάν εφαρμόζονται μη γραμμικές τεχνικές. Οι μη γραμμικές μέθοδοι διατηρούν αποτελεσματικά τα άκρα και τις λεπτομέρειες των εικόνων ενώ οι μέθοδοι που χρησιμοποιούν γραμμικούς τελεστές τείνουν να θολώνουν και να παραμορφώνονται. Επιπλέον, τα μη γραμμικά εργαλεία βελτίωσης εικόνας είναι λιγότερο ευαίσθητα στον θόρυβο. Ο θόρυβος υπάρχει πάντα λόγω της φυσικής τυχαιότητας των συστημάτων λήψης εικόνων. Για παράδειγμα, οι συνθήκες χαμηλής έκθεσης και χαμηλού φωτισμού σε αναλογικές συνθήκες φωτογραφίας οδηγούν σε εικόνες με θόρυβο κόκκου φιλμ, οι οποίες, μαζί με το ίδιο το σήμα εικόνας, καταγράφονται κατά τη διαδικασία ψηφιοποίησης.

Μία ακόμη μέθοδο είναι αυτή στο τομέα της συχνότητας όπου η εικόνα μεταφέρεται αρχικά στο πεδίο της συχνότητας, δηλαδή εκτελείτε πρώτα ο μετασχηματισμός Fourier. Όλες οι λειτουργίες βελτίωσης εκτελούνται στον μετασχηματισμό Fourier της εικόνας και στη συνέχεια ο μετασχηματισμός Inverse Fourier Transform

πραγματοποιείται για τη λήψη της προκύπτουσας εικόνας. Αυτές οι λειτουργίες βελτίωσης εκτελούνται προκειμένου να τροποποιηθεί η φωτεινότητα της εικόνας, η αντίθεση ή η κατανομή των επιπέδων του γκρι. Κατά συνέπεια, η τιμή pixel (ένταση) της εικόνας εξόδου θα τροποποιηθεί σύμφωνα με τη συνάρτηση μετασχηματισμού που εφαρμόζεται στις τιμές εισόδου. Η βελτίωση της εικόνας εφαρμόζεται σε κάθε πεδίο όπου οι εικόνες πρέπει να κατανοηθούν και να αναλυθούν όπως με την ανάλυση ιατρικών εικόνων ανάλυση εικόνων από δορυφόρους κ.λπ.

3.2 Φίλτρα δύο διαστάσεων

Παρακάτω περιγράφονται κάποια φίλτρα που συντελούν στην βελτίωση των εικόνων ανάλογα με το κάθε τύπο θορύβου που παρουσιάζει η κάθε μία.

3.2.1 Βαθυπερατό Φίλτρο 2 διαστάσεων (2D Low-Pass Filter)

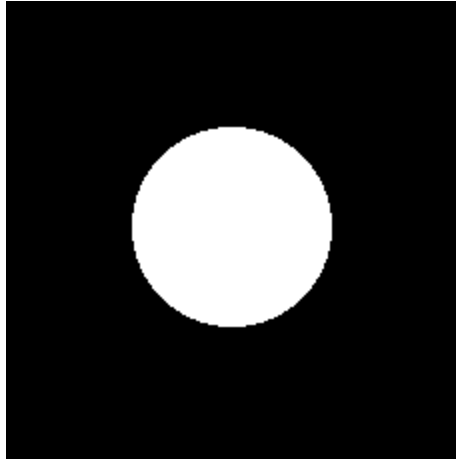
Το βαθυπερατό φίλτρο 2 διαστάσεων αποτελεί γραμμική επεξεργασία της εικόνας και εφαρμόζεται στο πεδίο της χωρικής συχνότητας όπου χρησιμοποιείται για την εξομάλυνση της εικόνας. Μειώνει τις υψηλές οπτικές συχνότητες (λεπτομέρειες) και διατηρεί τις χαμηλές συχνότητας (αδρά χαρακτηριστικά), ανάλογα με την οπτική συχνότητα αποκοπής του φίλτρου. Αυτό επιτυγχάνεται χρησιμοποιώντας φίλτρο με κυκλική συμμετρία γύρω από το σημείο (0,0) στο πεδίο των συχνοτήτων, και με διάμετρο D. Το φίλτρο έχει την τιμή 1 εντός του κύκλου διαμέτρου D (ζώνη διέλευσης – χαμηλές χωρικές συχνότητες) και τιμή 0 εκτός του κύκλου (ζώνη αποκοπής – υψηλές χωρικές συχνότητες).

$$H_{LP}(u, v) = \begin{cases} 1, & r(u, v) < F_c \\ 0, & otherwise \end{cases}$$
$$r(u, v) = \sqrt{m^2 + n^2}$$

όπου $u = 0, 1, \dots, M-1$, $v = 0, 1, \dots, N-1$ και $m = (u - M/2)/M$, $n = (v - N/2)/N$.

Η συχνότητα αποκοπής (F_c) μπορεί να καθοριστεί από την ακτίνα ενός κύκλου γύρω από τη μηδενική συχνότητα, ή ισοδύναμα τη διάμετρο D, όπως περιγράφεται στη συνέχεια.

Στη συνέχεια υλοποιούνται δύο παραδείγματα σε Matlab όπως παρουσιάζεται στις εικόνες. Συγκεκριμένα χρησιμοποιείται μια απλή εικόνα με πρόσθετο θόρυβο (Εικόνα 3.1) και στην συνέχεια μια βιοιατρική εικόνα με πρόσθετο θόρυβο (Εικόνα 3.2). Εμφανίζεται συγκριτικά η αρχική εικόνα καθώς και τα αποτελέσματα από την εφαρμογή του φίλτρου σε καθεμιά από τις εικόνες.



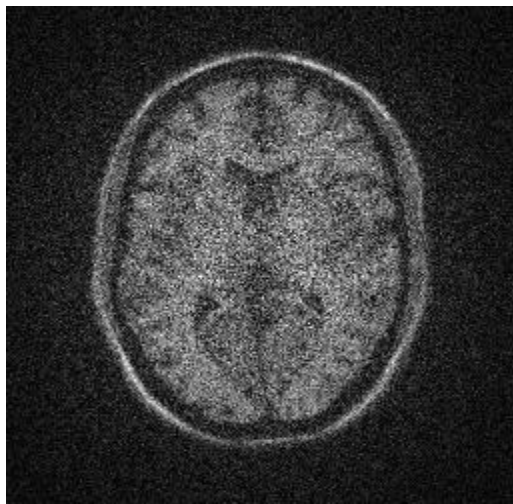
Εικόνα 3.1: Εικόνα 2D βαθυπερατού φίλτρου στο πεδίο των χωρικών συχνοτήτων.



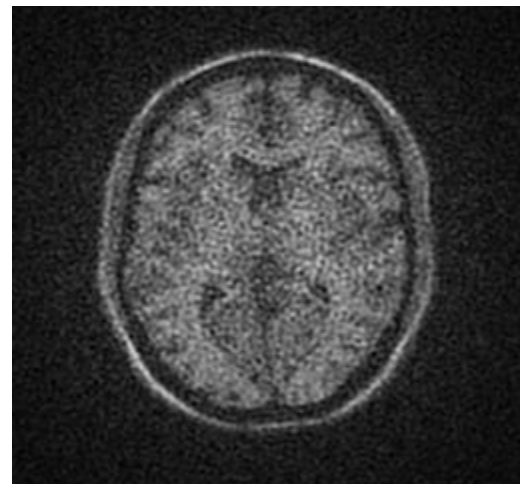
Εικόνα 3.1α: Αρχική απλή εικόνα. (Πηγή: https://en.wikipedia.org/wiki/Image_noise)



Εικόνα 3.1β: Εικόνα μετά από εφαρμογή low pass filter με $D = 75$



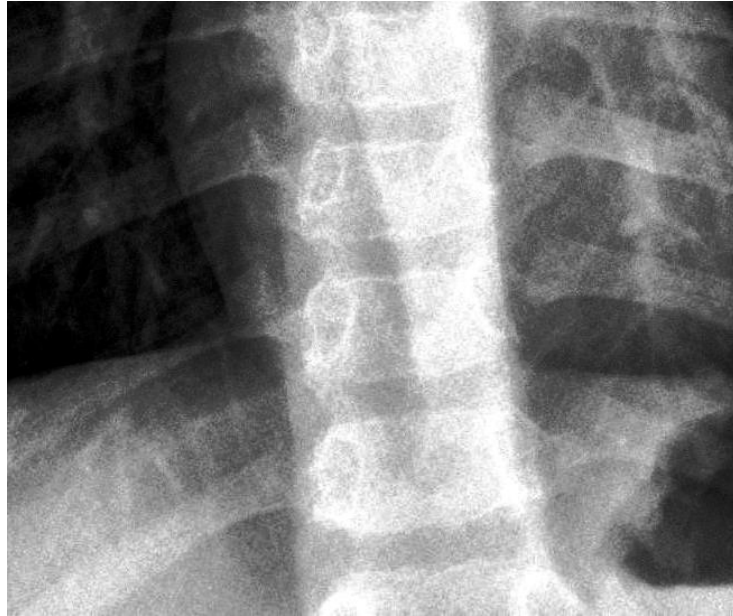
Εικόνα 3.2α: Αρχική εικόνα εγκεφάλου (Πηγή: <https://www.lpi.tel.uva.es/node/667>)



Εικόνα 3.2β: Εικόνα μετά από εφαρμογή low pass filter με $D = 75$.

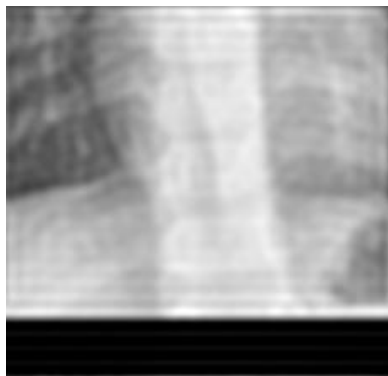
Όπως παρατηρείται στις Εικόνες 3.1β και 3.2β, με την εφαρμογή του βαθυπερατού φίλτρου ο θόρυβος έχει εξαλειφθεί παρότι οι εικόνες έχουν θολώσει λίγο (blurring).

Στο επόμενο, δεύτερο παράδειγμα διαφοροποιείται η τιμή της διαμέτρου του κύκλου (D), δηλαδή η συχνότητα αποκοπής του φίλτρου που απαιτείται ώστε να προκύψει το επιθυμητό αποτέλεσμα. Στο παράδειγμα αυτό επιλέγονται 2 περιοχές της εικόνας και στη συνέχεια συγκρίνονται με διαφορετικό D.



Εικόνα 3.3: Αρχική εικόνα θώρακα

(Πηγή: <https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia>)

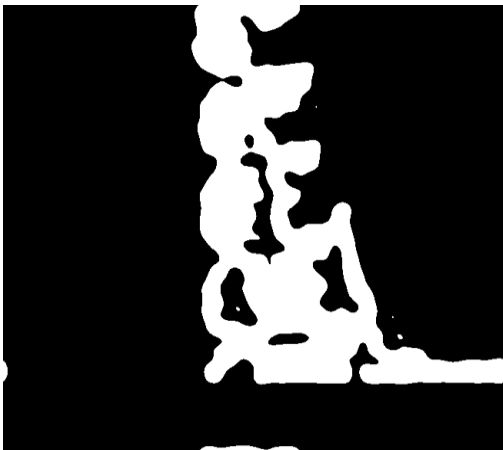


Εικόνα 3.3α: Εικόνα μετά από εφαρμογή βαθυπερατού φίλτρου με D=25

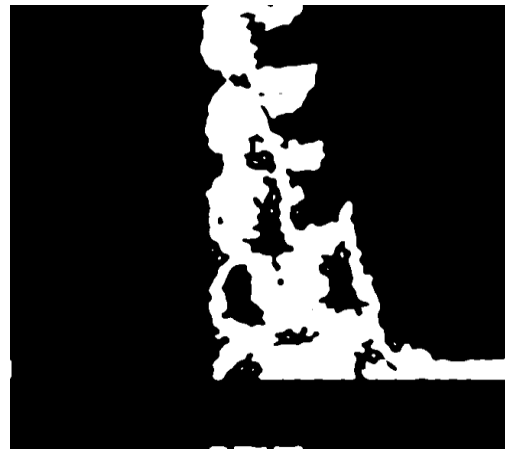


Εικόνα 3.3β: Εικόνα μετά από εφαρμογή βαθυπερατού φίλτρου με D=50

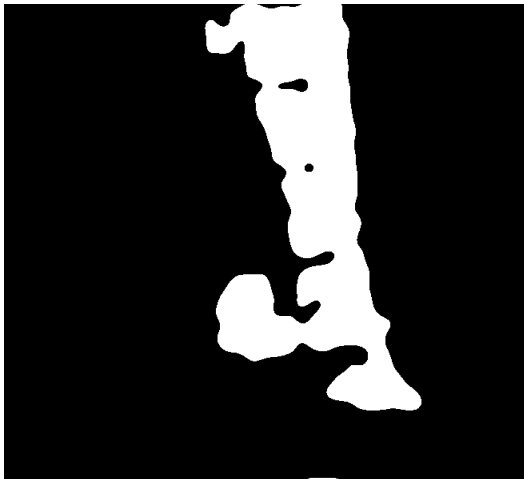
Στο επόμενο παράδειγμα φαίνονται εικόνες ακτινογραφιών θώρακα, μετά την εφαρμογή της μεθόδου region growing. Καθεμία εμφανίζεται πριν και μετά την εφαρμογή βαθυπερατού φίλτρου. Όπως παρατηρεί κανείς στις Εικόνες 3.4β και 3.5β που έχουν περάσει από φίλτρο με $D=50$, εμφανίζονται περισσότερες λεπτομέρειες σε σύγκριση με τις Εικόνες 3.4α και 3.5α όπου έχει χρησιμοποιηθεί «στενότερο» φίλτρο με $D=25$. Οπότε συμπερασματικά όσο αυξάνεται η διάμετρος D της κυκλικής περιοχής διέλευσης του φίλτρου, τόσο περισσότερες λεπτομέρειες διατηρούνται στην έξοδο (στη φιλτραρισμένη εικόνα).



Εικόνα 3.4α: Εικόνα region growing, μετά από εφαρμογή low pass filter με $D=25$.



Εικόνα 3.4β: Εικόνα region growing, μετά από εφαρμογή low pass filter με $D=50$.



Εικόνα 3.5α: Εικόνα region growing, μετά από εφαρμογή low pass filter με $D=25$.



Εικόνα 3.5β: Εικόνα region growing, μετά από εφαρμογή low pass filter με $D=50$.

3.2.2 Υψιπερατό Φίλτρο 2 διαστάσεων (2D High-Pass Filter)

Το υψιπερατό φίλτρο 2 διαστάσεων αποτελεί γραμμική επεξεργασία της εικόνας και εφαρμόζεται στο πεδίο της χωρικής συχνότητας όπου χρησιμοποιείται για τον τονισμό των ακμών και των λεπτομερειών της εικόνας. Μειώνει το περιεχόμενο χαμηλής χωρικής συχνότητας και διατηρεί αυτό της υψηλής χωρικής συχνότητας (λεπτομέρειες, ακμές). Αυτό επιτυγχάνεται χρησιμοποιώντας φίλτρο με κυκλική συμμετρία γύρω από το σημείο $(0,0)$ στο πεδίο των συχνοτήτων, και με διάμετρο D . Το φίλτρο έχει την τιμή 1 εκτός του κύκλου διαμέτρου D (ζώνη διέλευσης – υψηλές συχνότητες) και τιμή 0 εντός του κύκλου (ζώνη αποκοπής – χαμηλές συχνότητες).

$$H_{HP}(u, v) = \begin{cases} 1, & r(u, v) > F_c \\ 0, & otherwise \end{cases}$$

$$H_{HP}(u, v) = 1 - H_{LP}(u, v)$$



Εικόνα 3.5γ: Εικόνα 2D υψιπερατού φίλτρου

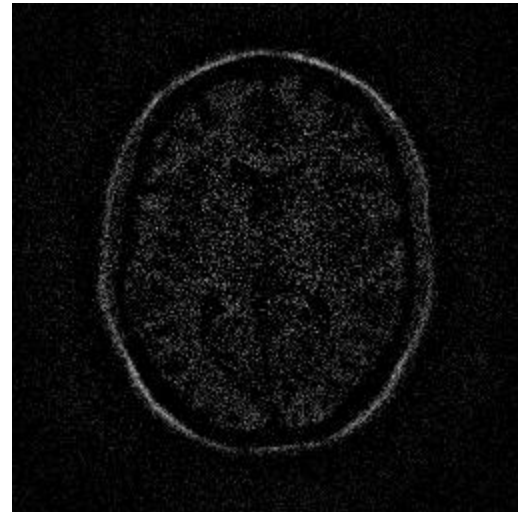
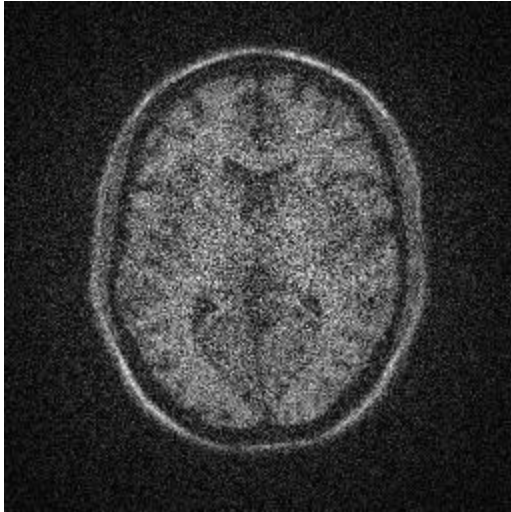
Στη συνέχεια υλοποιούνται δύο παραδείγματα εφαρμογής υψιπερατού φίλτρου σε Matlab. Συγκεκριμένα χρησιμοποιείται μια απλή εικόνα (Εικόνα 3.6) και μια βιοϊατρική εικόνα (Εικόνα 3.7). Σε κάθε περίπτωση εμφανίζεται η αρχική εικόνα καθώς και τα αποτελέσματα μετά την εφαρμογή του υψιπερατού φίλτρου με το αναγραφόμενο D.



Εικόνα 3.6α: Αρχική εικόνα κάμεραμαν
(Πηγή:
https://www.researchgate.net/figure/The-cameraman-image-256-256_fig7_228353323)



Εικόνα 3.6β: Εικόνα μετά από εφαρμογή high pass filter με $D=25$



Εικόνα 3.7α: Αρχική εικόνα εγκεφάλου
(Πηγή:
<https://www.lpi.tel.uva.es/node/667>)

Εικόνα 3.7β: Εικόνα μετά από εφαρμογή
high pass filter με $D=5$

Όπως παρατηρείται και στις δύο εικόνες, με την εφαρμογή του υπερπαστού φίλτρου έχουν τονισθεί οι ακμές παρότι χάνεται σε κάποιο βαθμό το περιεχόμενο των εικόνων.

3.2.3 Φίλτρο Ενδιάμεσης Τιμής 2 διαστάσεων (2D Median Filter)

Το φίλτρο ενδιάμεσης τιμής (median filter) αποτελεί μη γραμμική επεξεργασία και χρησιμοποιείται συχνά στην επεξεργασία εικόνας για τη μείωση του θορύβου, ειδικά του θορύβου τύπου «αλατοπίπερου» (salt & pepper noise). Ο μέσος όρος της γειτονίας των pixels μπορεί να καταστέλλει τον απομονωμένο θόρυβο εκτός εμβέλειας, αλλά η παρενέργεια είναι ότι θολώνει επίσης ξαφνικές αλλαγές όπως χαρακτηριστικά γραμμής, αιχμηρές άκρες και άλλες λεπτομέρειες εικόνας που αντιστοιχούν σε υψηλές χωρικές συχνότητες. Το ενδιάμεσο φίλτρο είναι μια αποτελεσματική μέθοδος που μπορεί, σε κάποιο βαθμό, να διακρίνει απομονωμένο θόρυβο εκτός εμβέλειας από φυσιολογικά χαρακτηριστικά εικόνας, όπως άκρα και γραμμές. Συγκεκριμένα, το ενδιάμεσο φίλτρο αντικαθιστά ένα pixel, αντί του μέσου όρου, όλων των pixels σε μια γειτονιά w

$$y[m, n] = \text{median} \{x[i, j], (i, j) \in w\}$$

όπου w αντιπροσωπεύει μια γειτονιά που ορίζεται από το χρήστη, με επίκεντρο την τοποθεσία $[m, n]$ στην εικόνα.

Παρακάτω υλοποιούνται δύο παραδείγματα σε Matlab με εφαρμογή φίλτρου ενδιάμεσης τιμής. Συγκεκριμένα χρησιμοποιείται μια απλή εικόνα (Εικόνα 3.8) και μια βιοιατρική εικόνα (Εικόνα 3.9). Εμφανίζονται οι αρχικές εικόνες, οι αρχικές εικόνες με πρόσθετο συνθετικό θόρυβο τύπου salt & pepper, καθώς και τα αποτελέσματα μετά την εφαρμογή του φίλτρου στην κάθε θορυβώδη εικόνα.



Εικόνα 3.8α: Αρχική εικόνα καμεραμάν

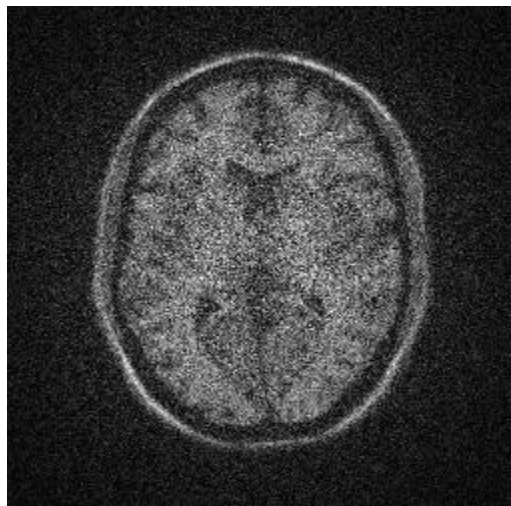
(Πηγή: <https://www.researchgate.net/figure/The-cameraman-image-256-256-fig7-228353323>)



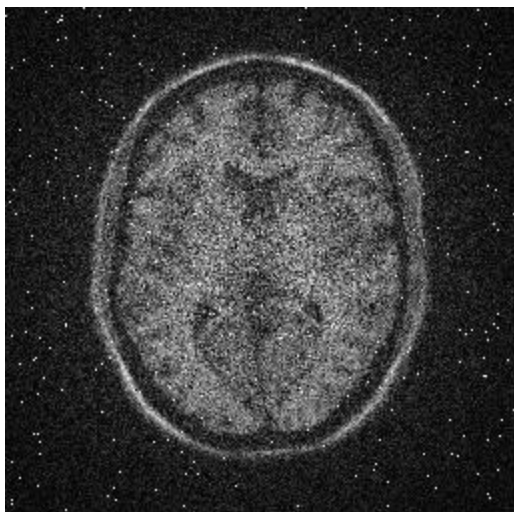
Εικόνα 3.8β: Εικόνα με πρόσθετο θόρυβο salt & pepper ισχύος 0.01



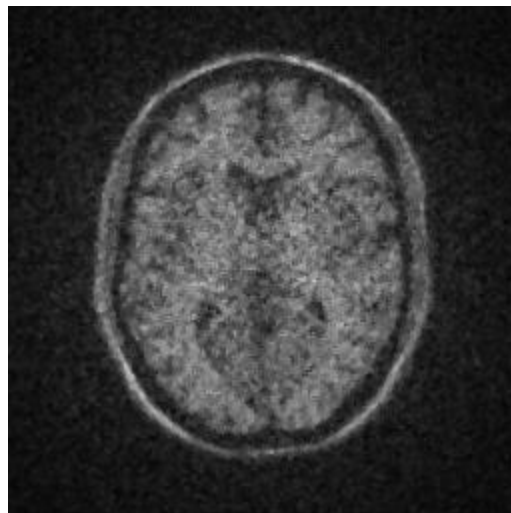
Εικόνα 3.8γ: Εικόνα μετά από εφαρμογή median filter στην θορυβώδη εικόνα.



Εικόνα 3.9α: Αρχική εικόνα εγκεφάλου
(Πηγή: <https://www.lpi.tel.uva.es/node/667>)



Εικόνα 3.9β: Εικόνα με πρόσθετο θόρυβο salt & pepper ισχύος 0.01



Εικόνα 3.9γ: Εικόνα μετά από εφαρμογή median filter στη θορυβώδη εικόνα.

Όπως παρατηρείται στις Εικόνες 3.8 και 3.9, με την εφαρμογή του φίλτρου ενδιάμεσης τιμής, ο θόρυβος έχει φύγει χωρίς να έχει θολώσει πολύ τις εικόνες.

Κεφάλαιο 4: Κατάτμηση Εικόνας με ανάπτυξη περιοχών

4.1 Κατάτμηση εικόνας

Η κατάτμηση (segmentation) εικόνας είναι η διαδικασία διαίρεσης μιας εικόνας σε διαφορετικές περιοχές με βάση τα χαρακτηριστικά των pixels για τον προσδιορισμό αντικειμένων ή ορίων μέσα στην εικόνα, και με τελικό στόχο την απλοποίηση μιας εικόνας και την αποτελεσματικότερη ανάλυσή της. Η κατάτμηση ενδιαφέρει πολλές εφαρμογές, από τη βιομηχανία κινηματογραφίας έως τον τομέα της ιατρικής. Ορισμένες ιατρικές εφαρμογές κατάτμησης περιλαμβάνουν την ταυτοποίηση του τραυματισμένου μυός, τη μέτρηση των οστών και των ιστών και την ανίχνευση ύποπτων δομών για να βοηθήσουν τους ακτινολόγους (Computer Aided Diagnosis ή CAD).

4.2. Η μέθοδος ανάπτυξης περιοχής (region growing)

Η μέθοδος Ανάπτυξης Περιοχής είναι γνωστή ως ένας απλός και γρήγορος αλγόριθμος για κατάτμηση μιας εικόνας. Είναι μία απλή μέθοδος και κατατάσσεται ως μέθοδος κατάτμησης βάση τα pixels, καθώς περιλαμβάνει την επιλογή των αρχικών σημείων (σπόρων, seeds). Αυτή η προσέγγιση κατάτμησης ξεκινά από το αρχικό σημείο-σπόρο, και διαδοχικά εξετάζει τα γειτονικά pixels των αρχικών σημείων-σπόρων και καθορίζει εάν πρέπει να προστεθούν στην περιοχή. Αυτό συμβαίνει αν έχουν ίδιο ή παραπλήσιο επίπεδο του γκρι με το σημείο-σπόρο. Αν ναι, η περιοχή γύρω από το σπόρο μεγαλώνει. Η διαδικασία επαναλαμβάνεται, με τον ίδιο τρόπο όπως οι γενικοί αλγόριθμοι ομαδοποίησης δεδομένων.

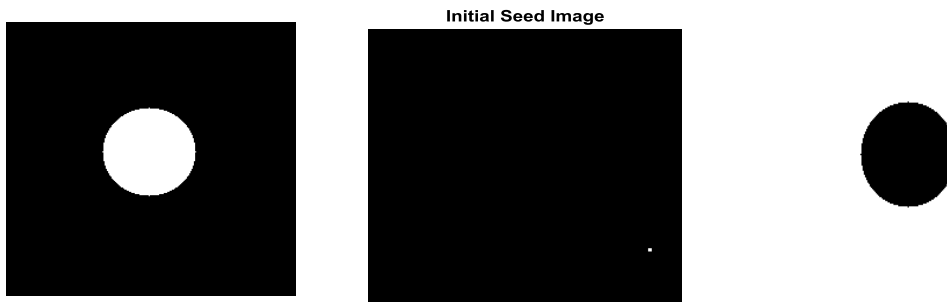
(8 neighbors, predicate: $|z - z_{seed}| < 0.1(\max_z - \min_z)$)

$$P(R_i) = True: \text{ if } |z - z_{seed}| < T$$

Πολλές δημοσιεύσεις ερευνών σχετικά με την κατάτμηση ιατρικής εικόνας έχουν αναφέρει τη χρήση αυτού του αλγορίθμου σε μια ποικιλία εφαρμογών, για παράδειγμα, για την ανίχνευση καρδιακών παθήσεων και καρκίνου του μαστού και για τον καθορισμό όγκων.

4.2.1 Παράδειγμα κατάτμησης απλής συνθετικής εικόνας με ένα σπόρο

Παρακάτω περιγράφεται ένα απλό παράδειγμα κατάτμησης με την χρήση του αλγορίθμου region growing σε μία απλή εικόνα. Αρχικά δημιουργείται μία απλή συνθετική εικόνα με έναν κύκλο και εν συνεχεία τοποθετείται ένα τυχαίο σημείο (σπόρος). Ο σκοπός είναι με αρχή τον σπόρο αυτό να διαχωριστούν οι δυο περιοχές (εντός - εκτός κύκλου).



Εικόνα 4.1α: Αρχική εικόνα

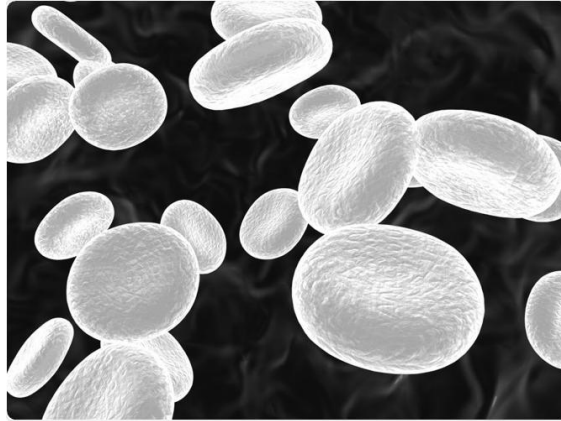
Εικόνα 4.1β: Τυχαίο σημείο-σπόρος στην αρχική εικόνα

Εικόνα 4.1γ: Εικόνα μετά την εφαρμογή region growing

Όπως παρατηρείται στην Εικόνα 4.1, ο σπόρος έπεσε έξω από τον λευκό κύκλο (σε μαύρο pixel), οπότε θα επεκταθεί γύρω-γύρω στις γειτονικές περιοχές (pixels) που είναι ίδια δηλαδή μαύρα, όπως φαίνεται και στο αποτέλεσμα (Εικόνα 4.1γ).

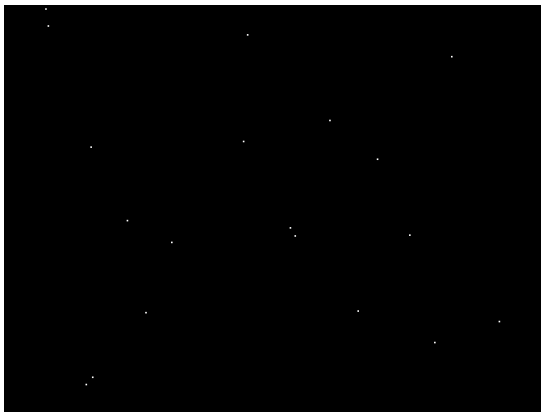
4.2.2 Παράδειγμα κατάτμησης ασπρόμαυρης εικόνας με περισσότερους σπόρους

Στο συγκεκριμένο παράδειγμα περιγράφεται η κατάτμηση εικόνας με τη μέθοδο της ανάπτυξης περιοχής σε μια ασπρόμαυρη εικόνα με αιμοσφαίρια. Τοποθετούνται είκοσι (20) σπόροι σε τυχαίες θέσεις στην εικόνα, με στόχο τον διαχωρισμό της κάθε περιοχής που βρίσκεται ο κάθε σπόρος. Εν συνεχεία ο διαχωρισμός θα πραγματοποιηθεί με βάση κοντινά επίπεδα του γκρι, με χρήση κατωφλίου (Threshold).

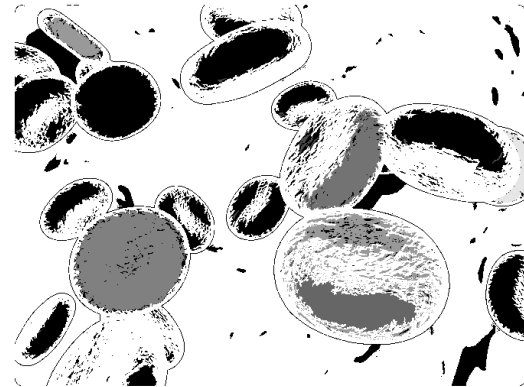


Εικόνα 4.2α: Αρχική ασπρόμαυρη εικόνα.

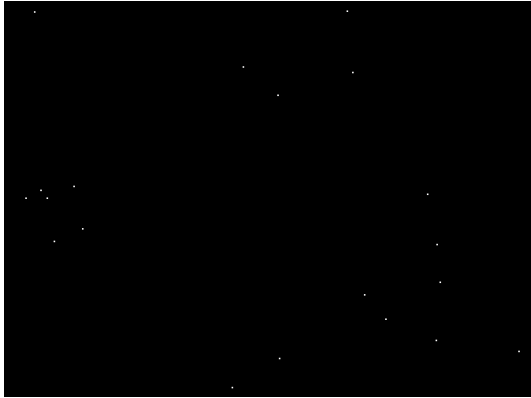
(Πηγή: <https://www.news-medical.net/life-sciences/Using-Flow-Cytometry-to-Analyze-Peripheral-Blood-Cells.aspx>)



Εικόνα 4.2β: Είκοσι (20) τυχαία σημεία-σπόροι πάνω στην αρχική εικόνα.



Εικόνα 4.2γ: Εικόνα μετά την εφαρμογή region growing με κατώφλι 0.1



Εικόνα 4.3β: Είκοσι (20) τυχαία σημεία-σπόροι πάνω στην αρχική εικόνα



Εικόνα 4.3γ: Εικόνα μετά την εφαρμογή region growing με κατώφλι 0.15

Όπως παρατηρείται συγκρίνοντας τις Εικόνες 4.2γ και 4.3γ, ο διαχωρισμός των περιοχών έχει πραγματοποιηθεί, ωστόσο όσο ανεβαίνει η τιμή του κατώφλιου τόσο χάνονται κάποιες λεπτομέρειες της εικόνας καθώς συγχωνεύονται όλο και περισσότερες περιοχές.

4.2.3 Παράδειγμα κατάτμησης βιοϊατρικών εικόνων με δέκα (10) σπόρους και βαθυπερατό φίλτρο

Σε αυτό το παράδειγμα χρησιμοποιούνται δύο βιοϊατρικές εικόνες μία ακτινογραφία θώρακα και μία μαγνητική τομογραφία εγκεφάλου, και εφαρμόζεται ο αλγόριθμος region growing. Η ιδιαιτερότητα στο συγκεκριμένο παράδειγμα είναι ότι θα εμφανίζεται ξεχωριστά η περιοχή που έχει πέσει ο σπόρος στην εικόνα και στη συνέχεια εφαρμόζεται και ένα βαθυπερατό φίλτρο αποκλειστικά σε κάθε μία περιοχή ξεχωριστά που έπεσαν οι δέκα σπόροι. Μερικά από τα σημεία παρουσιάζονται παρακάτω πρώτα για την εικόνα θώρακα και στην συνέχεια για την εικόνα εγκεφάλου.



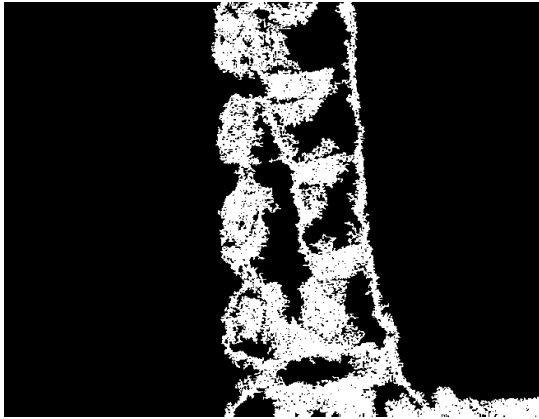
Εικόνα 4.4α: Αρχική εικόνα θώρακα (Πηγή: <https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia>)

Εικόνα 4.4β: Δέκα τυχαία σημεία στην αρχική εικόνα

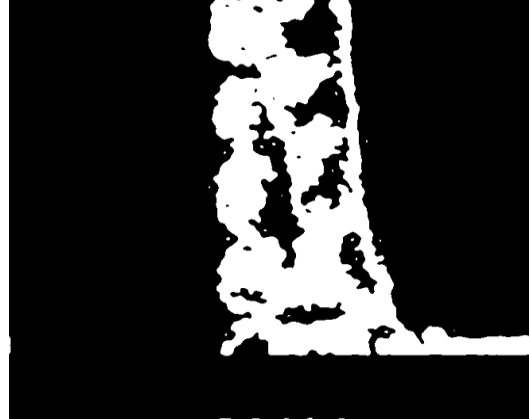


Εικόνα 4.4γ: Εικόνα μετά την εφαρμογή region growing με κατώφλι 0.1

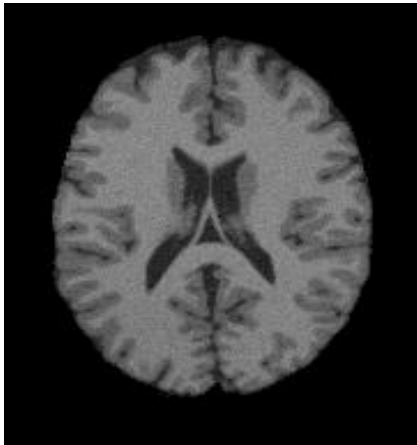
Εικόνα 4.4δ: Εικόνα μετά την εφαρμογή region growing και βαθυπερατού φίλτρου με $D=50$



Εικόνα 4.4ε: Εικόνα μετά την εφαρμογή region growing με κατώφλι 0.1



Εικόνα 4.4στ: Εικόνα μετά την εφαρμογή region growing και βαθυπερατού φίλτρου με $D=50$



Εικόνα 4.5α: Αρχική εικόνα εγκεφάλου (Πηγή: <https://stackoverflow.com/questions/31995502/best-method-to-find-edge-of-noise-image>)



Εικόνα 4.5β: Δέκα τυχαία σημεία στην αρχική εικόνα



Εικόνα 4.5γ: Εικόνα μετά την εφαρμογή region growing με κατώφλι 0.1



Εικόνα 4.5δ: Εικόνα μετά την εφαρμογή region growing & βαθυπερατού φίλτρου με $D=50$



Εικόνα 4.5ε: Εικόνα μετά την εφαρμογή region growing με κατώφλι 0.1



Εικόνα 4.5στ: Εικόνα μετά την εφαρμογή region growing & βαθυπερατού φίλτρου με $D=50$

Όπως παρατηρεί κανείς στις εικόνες 4.4δ, 4.4στ και 4.5δ, 4.5στ, με την διέλευση από το βαθυπερατό φίλτρο χάνονται ορισμένες λεπτομέρειες, σε σύγκριση με τις εικόνες 4.4γ, 4.4ε και 4.5γ, 4.5ε, αντίστοιχα.

Κεφάλαιο 5: Κατηγοριοποίηση εικόνων με χρήση νευρωνικών δικτύων

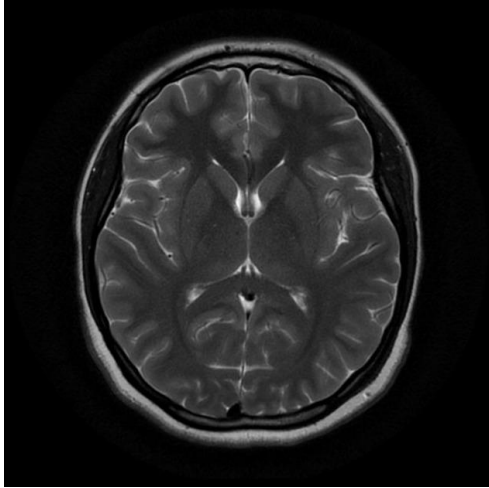
Ένας πολύ μεγάλος αριθμός βιοϊατρικών εικόνων δημιουργούνται σε νοσοκομεία, εργαστήρια απεικόνισης και βιοϊατρικά ιδρύματα σε καθημερινή βάση για να βοηθήσουν τους γιατρούς στη διάγνωση της υγείας των ασθενών. Οι τύποι βιοϊατρικών εικόνων περιλαμβάνουν εκείνες που προέρχονται από υπερήχους (US), μαγνητική τομογραφία (MRI), αξονική τομογραφία (CT) και ακτίνες X. Στον τομέα των βιοϊατρικών και ιατρικών αναλύσεων, οι εικόνες εξακολουθούν να διαδραματίζουν βασικό ρόλο στην ανάλυση της κατάστασης μιας ασθένειας και στη συνέχεια στην παροχή μιας ακριβούς διάγνωσης για τους ασθενείς. Ωστόσο, είναι μεγάλη πρόκληση η διαχείριση και η κατηγοριοποίηση μεγάλων και σημαντικών συλλογών βιοϊατρικών εικόνων που παράγονται καθημερινά. Είναι επίσης εξαιρετικά κουραστικό και ανέφικτο να εκτελείτε αυτό το έργο ανάλυσης αποκλειστικά μέσω ανθρώπινης παρέμβασης. Επομένως, για την καλύτερη διευκόλυνση στους γιατρούς, απαιτείται πιο ισχυρή τεχνολογική ανάλυση βιοϊατρικών εικόνων, ώστε να εντοπίζουν και να διακρίνουν τις ακριβείς καταστάσεις της νόσου κάθε ασθενούς. Παρακάτω περιγράφονται ορισμένα πειράματα με την χρήση νευρωνικού δικτύου για την κατηγοριοποίηση και την ταξινόμηση βιοϊατρικών εικόνων.

5.1 Πρώτο πείραμα

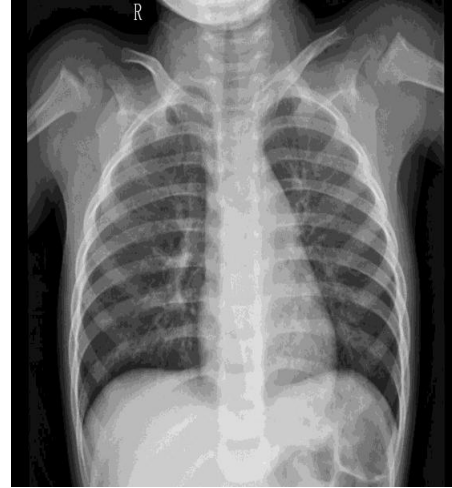
5.1.1 Στόχος του 1^{ου} πειράματος

Στο συγκεκριμένο πείραμα στόχος είναι η κατηγοριοποίηση βιοϊατρικών εικόνων σε δύο κλάσεις (binary classification). Οι εικόνες προέρχονται είτε από μαγνητική τομογραφία εγκεφάλου, είτε από ακτινογραφία θώρακα (kaggle.com). Στην Εικόνα 5.1 φαίνεται ένα αντιπροσωπευτικό παράδειγμα από κάθε κλάση (5.1α, 5.1β).

Για την κατηγοριοποίηση γίνεται χρήση ενός «ρηχού» ΤΝΔ τύπου MLP εμπροσθοτροφοδότησης (feedforward), με ένα (1) hidden layer. Το πείραμα πραγματοποιείται δύο φορές μεταβάλλοντας το πλήθος των κόμβων του hidden layer, (α) $q_1 = 4$ κόμβοι και (β) $q_2 = 6$ κόμβοι.



Εικόνα 5.1α: MRI – Εγκέφαλος (Πηγή: <https://www.kaggle.com/navoneel/brain-mri-images-for-brain-tumor-detection>)



Εικόνα 5.1β: XRAY – Θώρακας (Πηγή: <https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia>)

5.1.2 Data Set 1^ο πειράματος

Το data set αποτελείται από $Q1 = 100$ εικόνες από τον εγκέφαλο και $Q2 = 100$ εικόνες από το θώρακα. Οι εικόνες είναι όλες τύπου JPEG. Οι διαστάσεις των εικόνων κυμαίνονται από 168 έως 2713, το μέγεθος των αρχείων είναι από 4 KB έως 196 KB για τον εγκέφαλο και από 46 KB έως 641 KB για τον θώρακα.

Από τις $Q = Q1+Q2 = 100+100 = 200$ εικόνες, το 80% ήτοι $80+80 = 160$ εικόνες επιλέγονται τυχαία να αποτελέσουν το σύνολο εκπαίδευσης (training set) και το υπόλοιπο 20% ήτοι οι υπόλοιπες $20+20 = 40$ εικόνες αποτελούν το σύνολο ελέγχου (testing set).

5.1.3 Μέθοδος του 1^ο πειράματος σε βήματα

Το πείραμα εκτελείται στο Matlab™. Ο κώδικας δίνεται στο Παράρτημα Α.

ΒΗΜΑ 1^ο Εισάγεται στο Matlab το 1^ο σετ εικόνων ($Q1 = 100$ εικόνες εγκεφάλου) και υπολογίζεται η μεγαλύτερη διάσταση όλων, έστω $N1$.

ΒΗΜΑ 2^ο Εφαρμόζεται σε κάθε εικόνα δισδιάστατος μετασχηματισμός Fourier, αφού όλες οι εικόνες γεμίσουν με μηδενικά (zero-padding) έως την τετραγωνική διάσταση $N1 \times N1$.

ΒΗΜΑ 3^ο Γίνεται υποδειγματοληψία κατά παράγοντα $d1=50$ στο πεδίο της συχνότητας και επιστροφή στο πεδίο του σήματος (του χώρου) με

αντίστροφο δισδιάστατο μετασχηματισμό Fourier. Κάθε επεξεργασμένη εικόνα μετατρέπεται σε διάνυσμα και τοποθετείται ως μία νέα στήλη στον πίνακα με τα δεδομένα εκπαίδευσης P_TRAIN (για τις πρώτες 80 εικόνες) ή στον πίνακα με τα δεδομένα ελέγχου P_TEST (για τις τελευταίες 20 εικόνες).

ΒΗΜΑ 4^ο Εισάγεται στο Matlab το 2^ο σετ εικόνων (Q2 = 100 εικόνες θώρακα), και υπολογίζεται η μεγαλύτερη διάσταση όλων, έστω N2.

ΒΗΜΑ 5^ο Εφαρμόζεται σε κάθε εικόνα διδιάστατος μετασχηματισμός Fourier, αφού όλες οι εικόνες επαυξηθούν με μηδενικά (zero-padding) έως την τετραγωνική διάσταση N2 x N2.

ΒΗΜΑ 6^ο Γίνεται υποδειγματοληψία στο πεδίο της συχνότητας, κατά παράγοντα $d2=d1*(N2/N1) = 132$ σημεία, και επιστροφή στο πεδίο του σήματος (του χώρου) με αντίστροφο δισδιάστατο μετασχηματισμό Fourier. Κάθε επεξεργασμένη εικόνα μετατρέπεται σε διάνυσμα και τοποθετείται ως μία νέα στήλη στον πίνακα με τα δεδομένα εκπαίδευσης P_TRAIN (για τις πρώτες 80 εικόνες) ή στον πίνακα με τα δεδομένα ελέγχου P_TEST (για τις τελευταίες 20 εικόνες).

- Το P_TRAIN τελικά έχει 160 στήλες, μία για κάθε εικόνα στο σύνολο εκπαίδευσης,
- Το P_TEST τελικά έχει 40 στήλες, μία για κάθε εικόνα στο σύνολο ελέγχου.

ΒΗΜΑ 7^ο Δημιουργείται το σύνολο των σωστών κλάσεων (εξόδων) για το σύνολο εκπαίδευσης T_TRAIN, διάστασης 1x160, με τιμές 80 «1» και 80 «2». Επίσης δημιουργείται το σύνολο των σωστών κλάσεων (εξόδων) για το σύνολο ελέγχου T_TEST, διάστασης 1x40, με τιμές 20 «1» και 20 «2».

ΒΗΜΑ 8^ο Αρχικοποιείται το νευρωνικό δίκτυο (feedforwardnet) με 1 hidden layer των q1=4 [q2=6] κόμβων.

ΒΗΜΑ 9^ο Εκπαιδεύεται το ΤΝΔ με χρήση της συνάρτησης train και του αλγορίθμου Levenberg-Marquardt για την ελαχιστοποίηση του σφάλματος στην έξοδο του ΤΝΔ. Χρησιμοποιείται όριο 1000 επαναλήψεων, αν και σε

όλες τις περιπτώσεις η εκπαίδευση του ΤΝΔ συγκλίνει (ολοκληρώνεται) πολύ νωρίτερα.

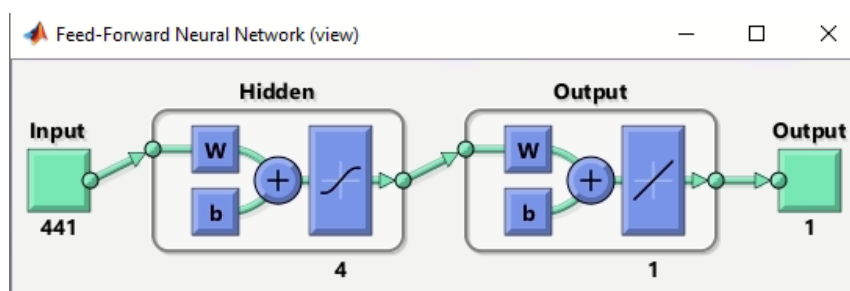
ΒΗΜΑ 10^ο Ελέγχεται η λειτουργία του ΤΝΔ χρησιμοποιώντας ως σύνολο ελέγχου το ίδιο το σύνολο εκπαίδευσης αποτελούμενο από 160 εικόνες. Μετρούνται οι σωστές και οι λάθος αποφάσεις κατηγοριοποίησης και ανάγονται σε ποσοστά %.

ΒΗΜΑ 11^ο Ελέγχεται η λειτουργία του ΤΝΔ χρησιμοποιώντας το σύνολο ελέγχου με τις 20+20 = 40 εικόνες που δεν χρησιμοποιήθηκαν στην εκπαίδευση του ΤΝΔ (νέες, άγνωστες εικόνες). Μετρούνται οι σωστές και οι λάθος αποφάσεις κατηγοριοποίησης και ανάγονται σε ποσοστά %.

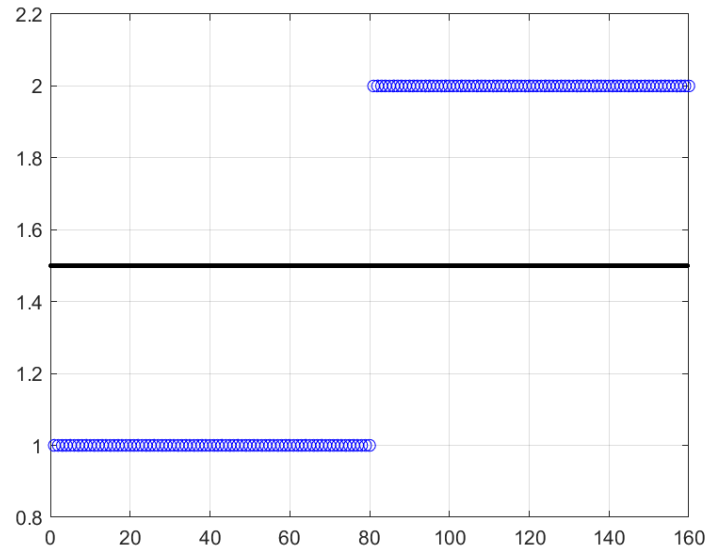
5.1.4 Αποτελέσματα του 1^{ου} πειράματος

Αποτελέσματα από το ΤΝΔ με 1 στρώμα και $q1 = 4$ κόμβους

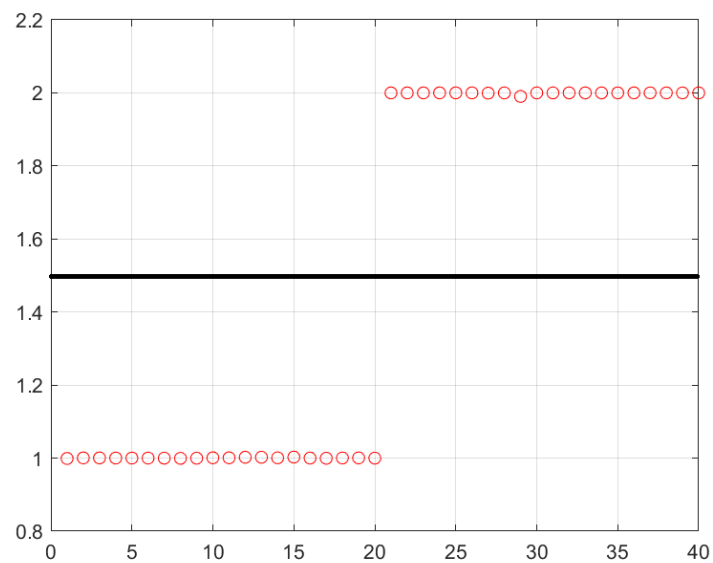
Παρακάτω παρουσιάζονται τα αποτελέσματα από το νευρωνικό δίκτυο που προέκυψαν με πρώτο τη δομή του νευρωνικού δικτύου με ένα στρώμα τεσσάρων κόμβων. Στην συνέχεια ακολουθούν τα αποτελέσματα της εκπαίδευσης και του ελέγχου αντίστοιχα, όπου έχει τοποθετηθεί μία διαχωριστική γραμμή στη μέση για την ευκρίνεια των αποτελεσμάτων. Τέλος εμφανίζεται η καμπύλη επίδοσης του νευρωνικού δικτύου, καθώς και το ιστόγραμμα σφάλματός του.



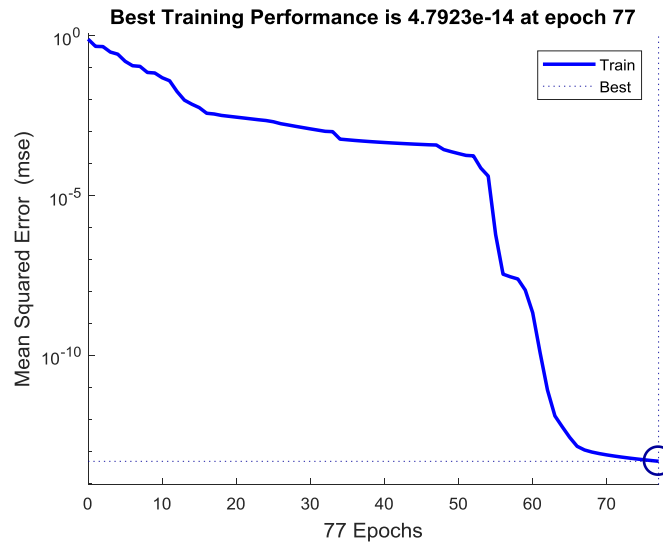
Εικόνα 5.2: Νευρωνικό δίκτυο με ένα 1 hidden layer τεσσάρων κόμβων.



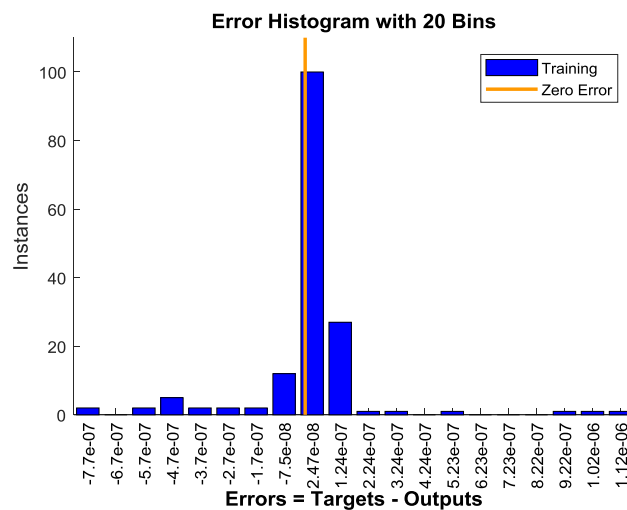
Εικόνα 5.3: Αποτέλεσμα εκπαίδευσης με 100% επιτυχία και στις δύο κατηγορίες



Εικόνα 5.4 : Αποτέλεσμα ελέγχου με 100% επιτυχία και στις δύο κατηγορίες



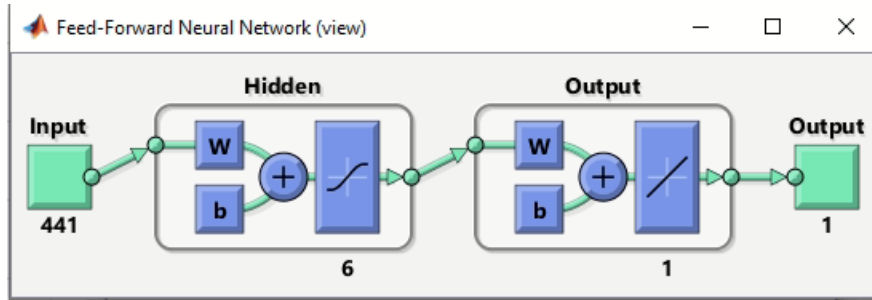
Εικόνα 5.5: Η επίδοση του νευρωνικού δικτύου



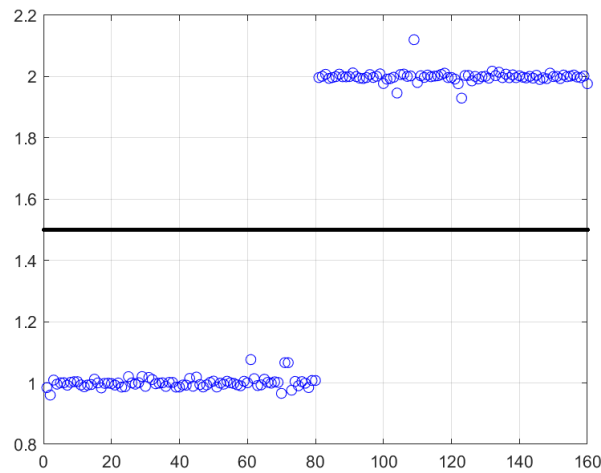
Εικόνα 5.6: Το ιστόγραμμα σφάλματος του νευρωνικού δικτύου

Αποτελέσματα από το ΤΝΔ με 1 στρώμα και q2=6 κόμβους

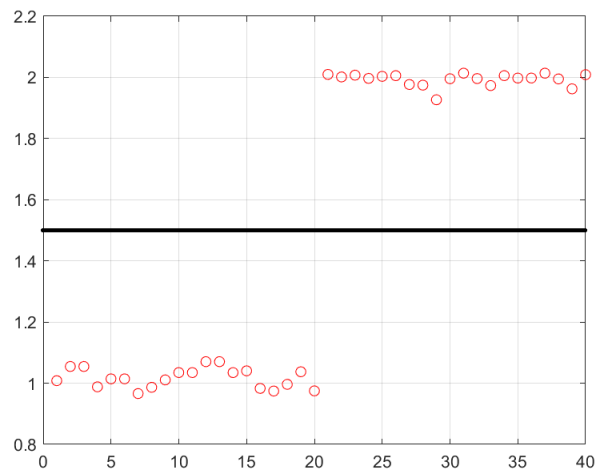
Παρακάτω παρουσιάζονται τα αποτελέσματα από το νευρωνικό δίκτυο που προέκυψαν με πρώτο τη δομή του νευρωνικού δικτύου με ένα στρώμα έξι κόμβων. Στην συνέχεια ακολουθούν τα αποτελέσματα της εκπαίδευσης και του ελέγχου αντίστοιχα, όπου έχει τοποθετηθεί μία διαχωριστική γραμμή στη μέση για την ευκρίνεια των αποτελεσμάτων. Τέλος εμφανίζεται η καμπύλη επίδοσης του νευρωνικού δικτύου, καθώς και το ιστόγραμμα σφάλματός του.



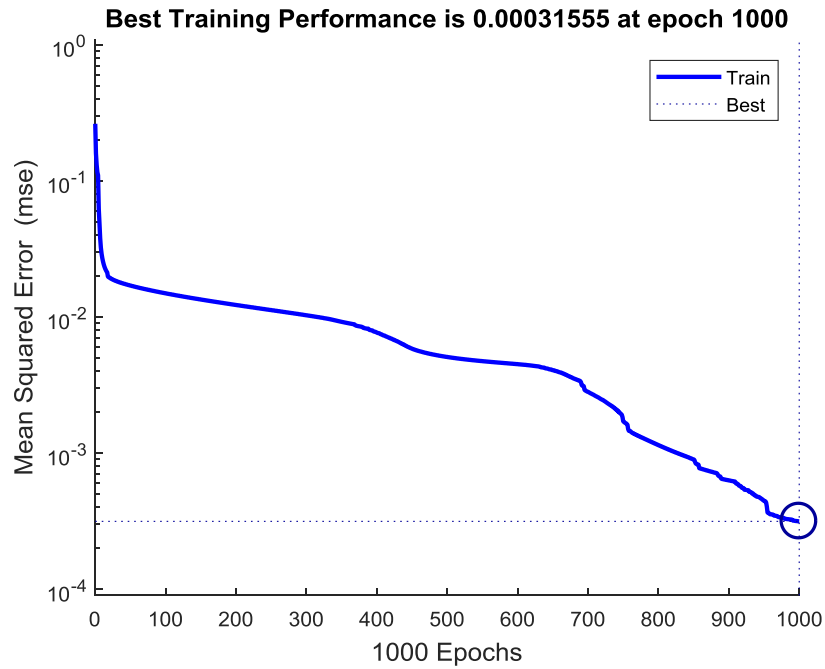
Εικόνα 5.7: Νευρωνικό δίκτυο με ένα hidden layer έξι κόμβων.



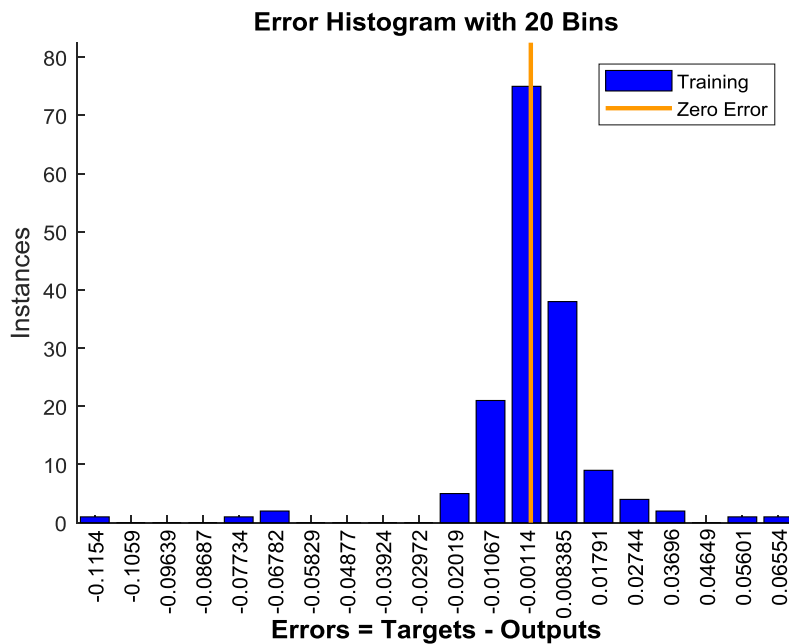
Εικόνα 5.8: Αποτέλεσμα εκπαίδευσης με 100% επιτυχία και στις δύο κατηγορίες



Εικόνα 5.9: Αποτέλεσμα ελέγχου με 100% επιτυχία και στις δύο κατηγορίες



Εικόνα 5.10: Η επίδοση του νευρωνικού δικτύου



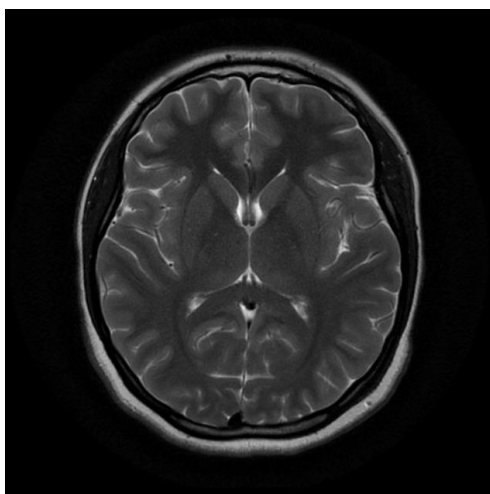
Εικόνα 5.11: Το ιστόγραμμα σφάλματος του νευρωνικού δικτύου

Τα αποτελέσματα και στις δύο περιπτώσεις είτε με τέσσερις κόμβους είτε με έξι κόμβους είναι άριστα, με 100% επιτυχία και στις δύο κατηγορίες, καθώς δεν περνάει καμία εικόνα την διαχωριστική γραμμή που είναι στην μέση μεταξύ 1 και 2, στο 1.5.

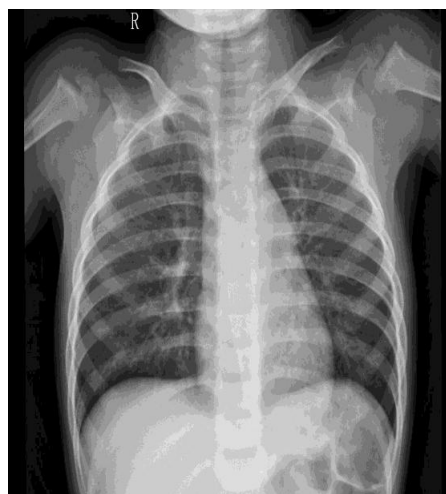
5.2 Δεύτερο πείραμα

5.2.1 Στόχος του 2^{ου} πειράματος

Στο πείραμα αυτό στόχος είναι η κατηγοριοποίηση βιοιατρικών εικόνων σε δύο κλάσεις (binary classification). Οι εικόνες προέρχονται είτε από μαγνητική τομογραφία εγκεφάλου είτε από ακτινογραφία θώρακα. Στην Εικόνα 5.12 φαίνεται ένα αντιπροσωπευτικό παράδειγμα από κάθε κλάση (Εικόνες 5.1α, 5.1β). Για την κατηγοριοποίηση γίνεται χρήση ενός «ρηχού» ΤΝΔ τύπου MLP εμπροσθοτροφοδότησης (feedforward), με ένα (1) hidden layer όπως στο πρώτο πείραμα με την διαφορά ότι στην προεπεξεργασία των εικόνων προστίθενται και δύο επιπλέον στοιχεία που εξετάστηκαν στις δύο προηγούμενες ενότητες, η χρήση βαθυπερατού φίλτρου και η μέθοδος κατάτμησης με ανάπτυξη περιοχής (region growing). Το πείραμα πραγματοποιείται δύο φορές μεταβάλλοντας το πλήθος των κόμβων του hidden layer, (α) $q1 = 4$ κόμβοι και (β) $q2 = 6$ κόμβοι.



Εικόνα 5.12α: MRI – Εγκέφαλος (Πηγή: <https://www.kaggle.com/navoneel/brain-mri-images-for-brain-tumor-detection>)



Εικόνα 5.12β: XRAY - Θώρακας (Πηγή: <https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia>)

5.2.2 Data Set 2^{ου} πειράματος

Το Data set που χρησιμοποιείται είναι το ίδιο με αυτό του πρώτου πειράματος.

5.2.3 Μέθοδος του 2^{ου} πειράματος σε βήματα

Το πείραμα εκτελείται στο Matlab™. Ο κώδικας δίνεται στο Παράρτημα Β.

ΒΗΜΑ 1^ο Εισάγεται στο Matlab το 1^ο σετ εικόνων (Q1 = 100 εικόνες εγκεφάλου) και υπολογίζεται η μεγαλύτερη διάσταση όλων, έστω N1.

ΒΗΜΑ 2^ο Εφαρμόζεται σε κάθε εικόνα δισδιάστατος μετασχηματισμός Fourier, αφού όλες οι εικόνες γεμίσουν με μηδενικά (zero-padding) έως την τετραγωνική διάσταση N1 x N1.

ΒΗΜΑ 3^ο Δημιουργείται η κυκλική συμμετρία του κύκλου που θα χρησιμοποιηθεί στο βαθυπερατό φίλτρο στη συνέχεια, με διάμετρο D1=50.

ΒΗΜΑ 4^ο Γίνεται μετατροπή από έγχρωμες εικόνες, σε εικόνες επιπέδου του γκρι όσες απαιτούνται, τοποθετείται ένα σημείο κεντρικά κάθε εικόνας και πραγματοποιείται η διαδικασία ανάπτυξης περιοχής (region growing) μαζί με ένα δισδιάστατο βαθυπερατό φίλτρο αμέσως μετά. Ακολουθεί η υποδειγματοληψία κατά παράγοντα d1=60 στο πεδίο της συχνότητας και επιστροφή στο πεδίο του σήματος (του χώρου) με αντίστροφο δισδιάστατο μετασχηματισμό Fourier. Κάθε επεξεργασμένη εικόνα μετατρέπεται σε διάνυσμα και τοποθετείται ως μία νέα στήλη στον πίνακα με τα δεδομένα εκπαίδευσης P_TRAIN (για τις πρώτες 80 εικόνες) ή στον πίνακα με τα δεδομένα ελέγχου P_TEST (για τις τελευταίες 20 εικόνες).

ΒΗΜΑ 5^ο Εισάγεται στο Matlab το 2^ο σετ εικόνων (Q2 = 100 εικόνες θώρακα), και υπολογίζεται η μεγαλύτερη διάσταση όλων, έστω N2.

ΒΗΜΑ 6^ο Εφαρμόζεται σε κάθε εικόνα διδιάστατος μετασχηματισμός Fourier, αφού όλες οι εικόνες επαυξηθούν με μηδενικά (zero-padding) έως την τετραγωνική διάσταση N2 x N2.

ΒΗΜΑ 7^ο Γίνεται υποδειγματοληψία κατά παράγοντα $d2=d1*(N2/N1) = 158$ στο πεδίο της συχνότητας και επιστροφή στο πεδίο του σήματος (του χώρου) με αντίστροφο δισδιάστατο μετασχηματισμό Fourier. Κάθε επεξεργασμένη εικόνα μετατρέπεται σε διάνυσμα και τοποθετείται ως μία νέα στήλη στον πίνακα με τα δεδομένα εκπαίδευσης P_TRAIN (για τις

πρώτες 80 εικόνες) ή στον πίνακα με τα δεδομένα ελέγχου P_TEST (για τις τελευταίες 20 εικόνες).

- Το P_TRAIN τελικά έχει 160 στήλες, μία για κάθε εικόνα στο σύνολο εκπαίδευσης,
- Το P_TEST τελικά έχει 40 στήλες, μία για κάθε εικόνα στο σύνολο ελέγχου.

ΒΗΜΑ 8^ο Δημιουργείται το σύνολο των σωστών κλάσεων (εξόδων) για το σύνολο εκπαίδευσης T_TRAIN, διάστασης 1x160, με τιμές 80 «1» και 80 «2». Επίσης δημιουργείται το σύνολο των σωστών κλάσεων (εξόδων) για το σύνολο ελέγχου T_TEST, διάστασης 1x40, με τιμές 20 «1» και 20 «2».

ΒΗΜΑ 9^ο Αρχικοποιείται το νευρωνικό δίκτυο (feedforwardnet) με 1 hidden layer των $q_1=4$ [$q_2=6$] κόμβων.

ΒΗΜΑ 10^ο Εκπαιδύεται το ΤΝΔ με χρήση της συνάρτησης train και του αλγορίθμου Levenberg-Marquardt για την ελαχιστοποίηση του σφάλματος στην έξοδο του ΤΝΔ. Χρησιμοποιείται όριο 1000 επαναλήψεων, αν και σε όλες τις περιπτώσεις η εκπαίδευση του ΤΝΔ συγκλίνει (ολοκληρώνεται) πολύ νωρίτερα.

ΒΗΜΑ 11^ο Ελέγχεται η λειτουργία του ΤΝΔ χρησιμοποιώντας ως σύνολο ελέγχου το ίδιο το σύνολο εκπαίδευσης αποτελούμενο από 160 εικόνες. Μετρούνται οι σωστές και οι λάθος αποφάσεις κατηγοριοποίησης και ανάγονται σε ποσοστά %.

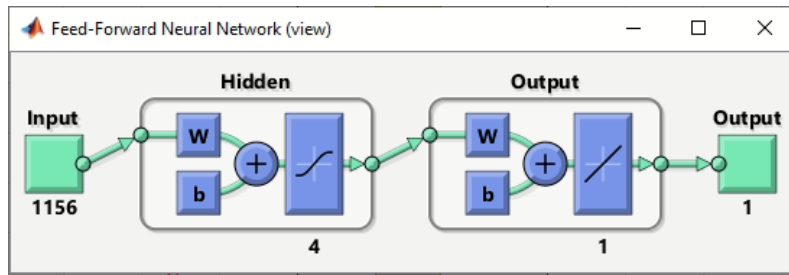
ΒΗΜΑ 12^ο Ελέγχεται η λειτουργία του ΤΝΔ χρησιμοποιώντας το σύνολο ελέγχου με τις 20+20 = 40 εικόνες που δεν χρησιμοποιήθηκαν στην εκπαίδευση του ΤΝΔ (νέες, άγνωστες εικόνες). Μετρούνται οι σωστές και οι λάθος αποφάσεις κατηγοριοποίησης και ανάγονται σε ποσοστά %.

5.2.4 Αποτελέσματα του 2^ο πειράματος

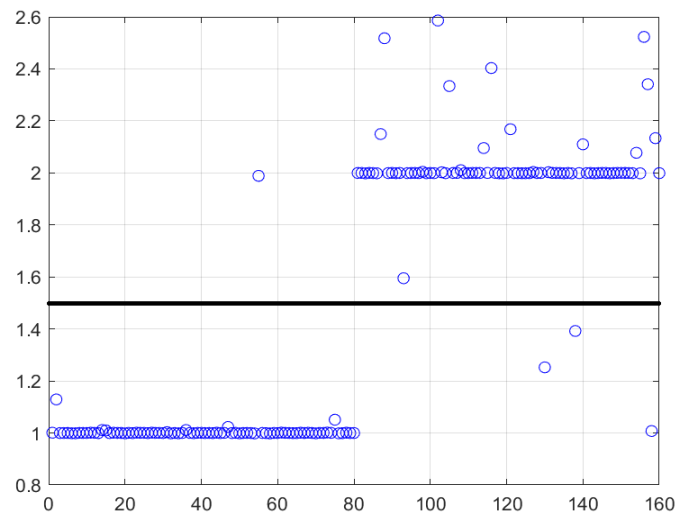
Αποτελέσματα από το ΤΝΔ με 1 στρώμα και $q_1 = 4$ κόμβους

Παρακάτω παρουσιάζονται τα αποτελέσματα κατηγοριοποίησης από το νευρωνικό δίκτυο που φαίνεται στην Εικόνα 5.13, με $q_1 = 4$ κόμβους στο πρώτο στρώμα. Στην συνέχεια ακολουθούν τα αποτελέσματα της εκπαίδευσης και του ελέγχου αντίστοιχα, όπου έχει τοποθετηθεί μία διαχωριστική γραμμή στη μέση για την ευκρίνεια των

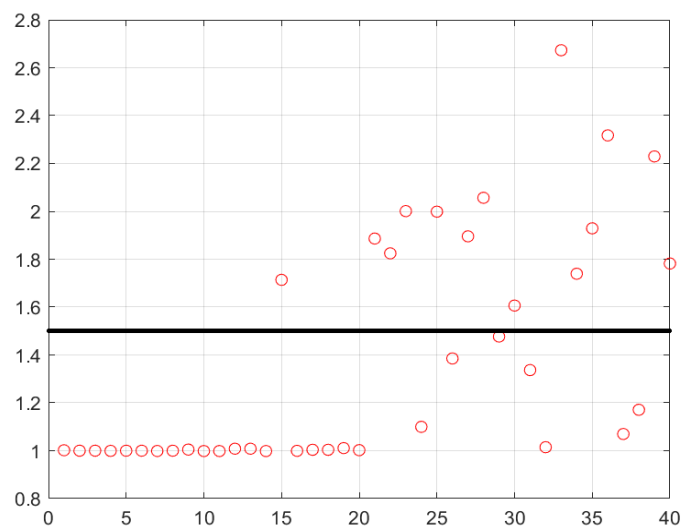
αποτελεσμάτων. Τέλος εμφανίζεται η καμπύλη επίδοσης του νευρωνικού δικτύου, καθώς και το ιστόγραμμα σφάλματός του.



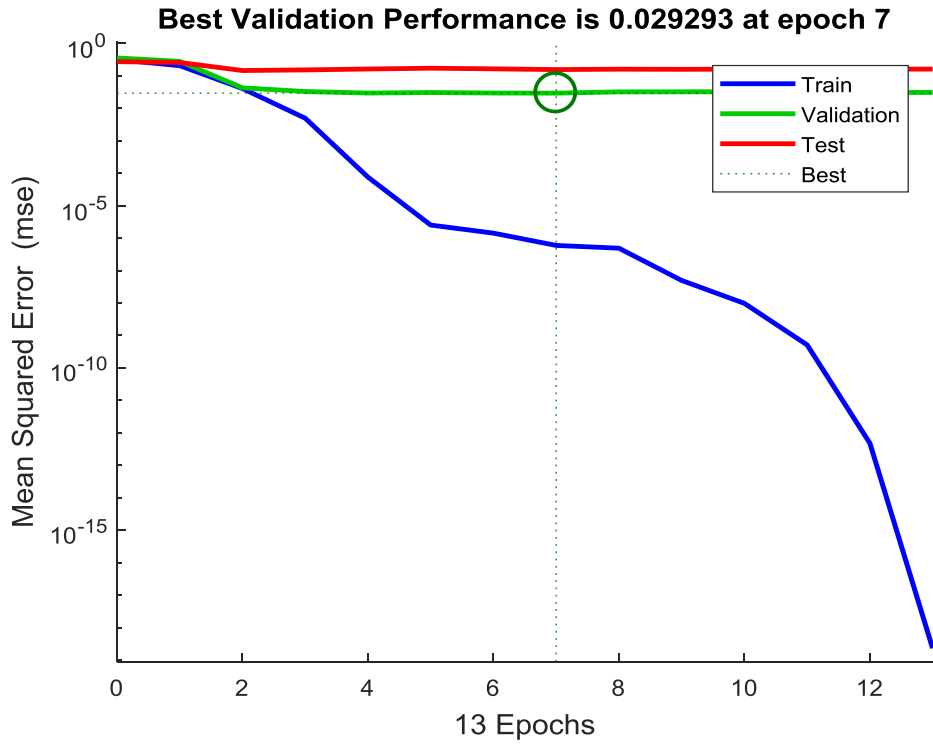
Εικόνα 5.13: Νευρωνικό δίκτυο με ένα 1 hidden layer τεσσάρων κόμβων.



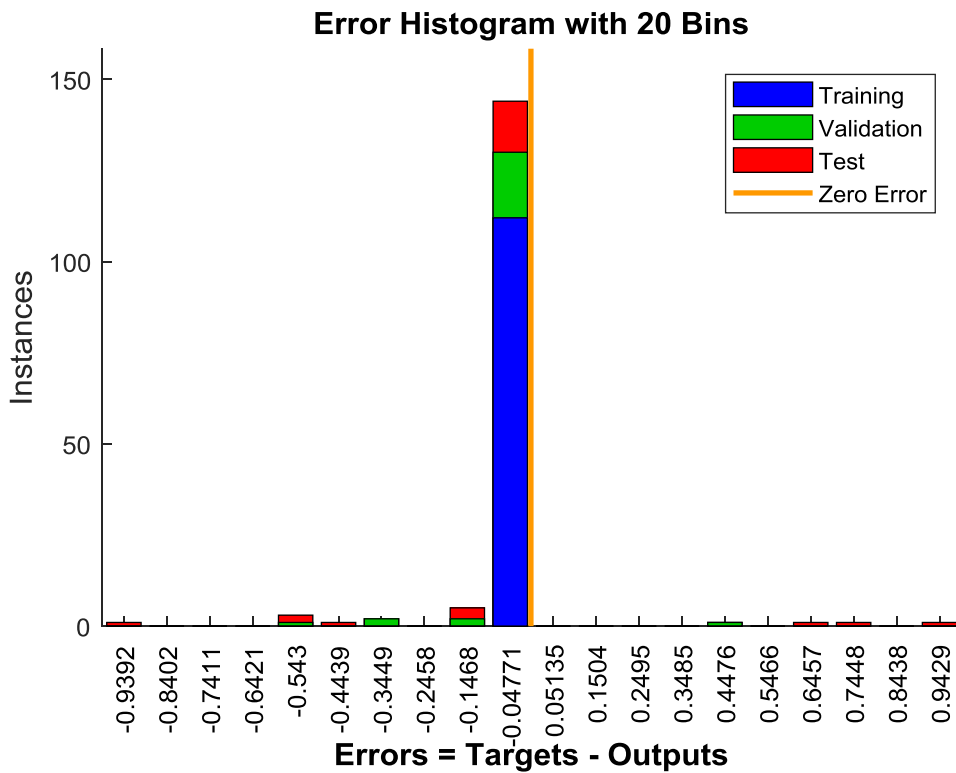
Εικόνα 5.14: Αποτέλεσμα εκπαίδευσης με 98.75% επιτυχία στην πρώτη κλάση και 96.25% επιτυχία στην δεύτερη κλάση



Εικόνα 5.15: Αποτέλεσμα ελέγχου με 95% επιτυχία στην πρώτη κλάση και 65% επιτυχία στην δεύτερη κλάση



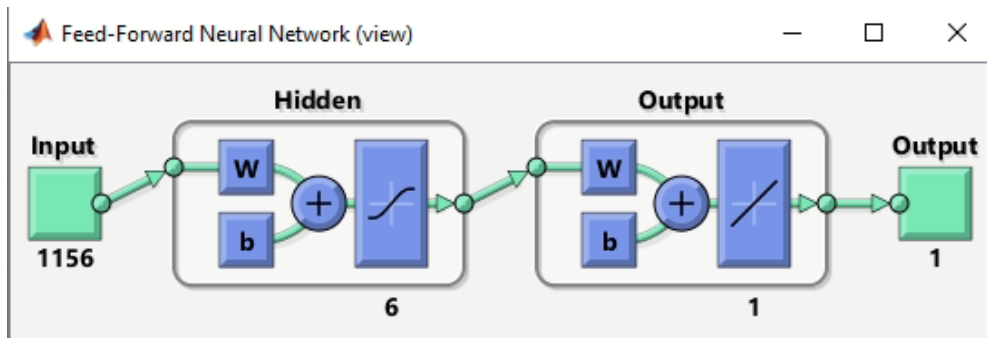
Εικόνα 5.16: Η επίδοση του νευρωνικού δικτύου



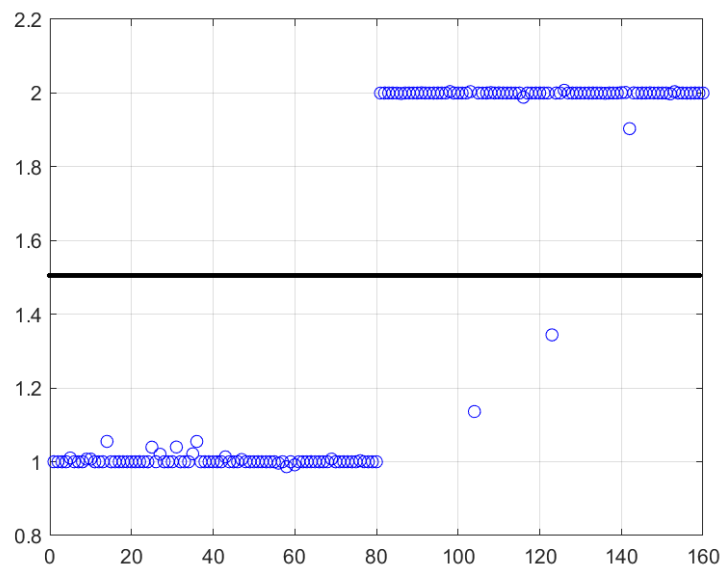
Εικόνα 5.17: Το ιστόγραμμα σφάλματος του νευρωνικού δικτύου

Αποτελέσματα από το ΤΝΔ με 1 στρώμα και $q_2=6$ κόμβους

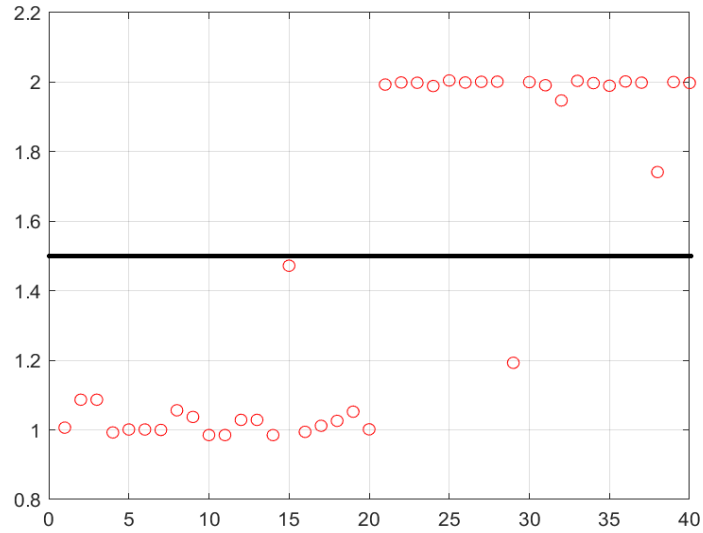
Παρακάτω παρουσιάζονται τα αποτελέσματα κατηγοριοποίησης από το νευρωνικό δίκτυο με ίδια δομή όπως το προηγούμενο, αλλά με $q_2 = 6$ κόμβους στο πρώτο στρώμα (Εικόνα 5.18). Στην συνέχεια ακολουθούν τα αποτελέσματα της εκπαίδευσης και του ελέγχου αντίστοιχα, όπου έχει τοποθετηθεί μία διαχωριστική γραμμή στη μέση για την ευκρίνεια των αποτελεσμάτων. Τέλος εμφανίζεται η καμπύλη επίδοσης του νευρωνικού δικτύου, καθώς και το ιστόγραμμα σφάλματός του.



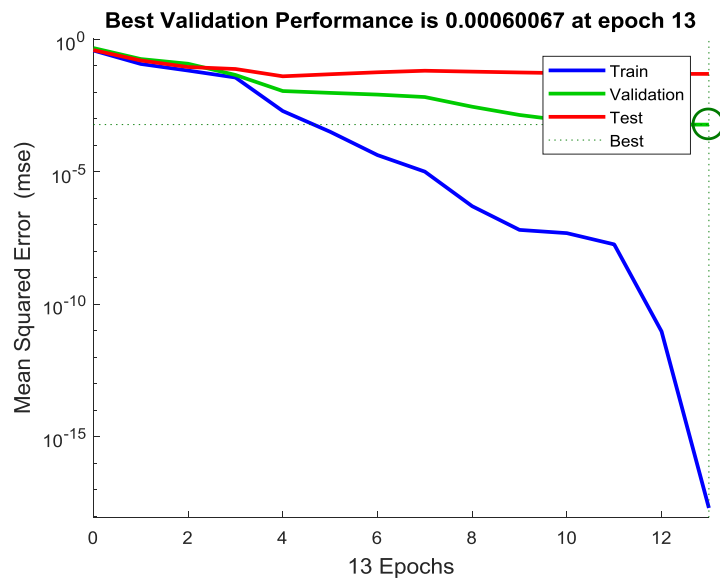
Εικόνα 5.18: Νευρωνικό δίκτυο με ένα 1 hidden layer έξι κόμβων.



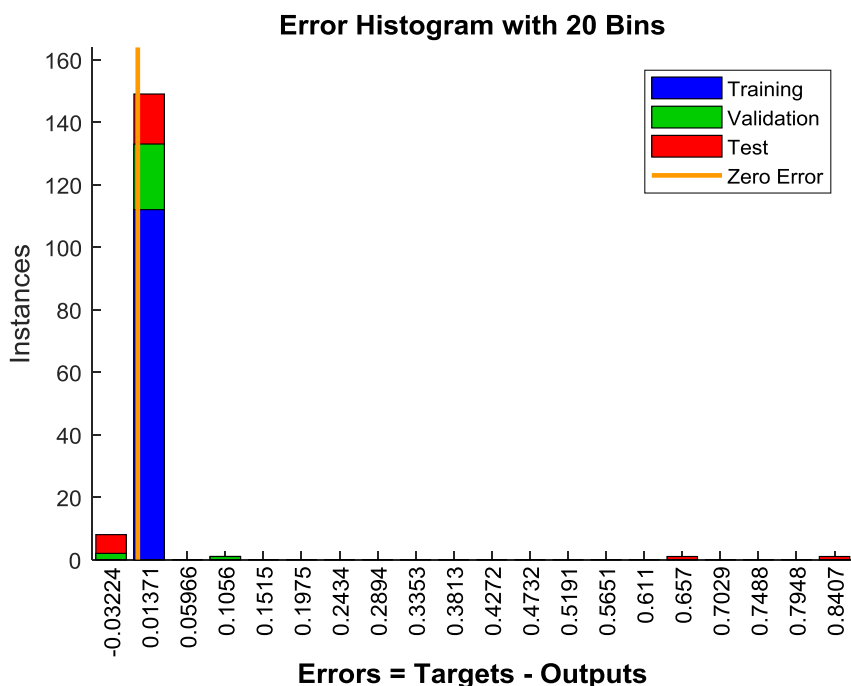
Εικόνα 5.19: Αποτέλεσμα εκπαίδευσης με 100% επιτυχία στην πρώτη κλάση και 97.5% επιτυχία στην δεύτερη κλάση



Εικόνα 5.20: Αποτέλεσμα ελέγχου με 100% επιτυχία στην πρώτη κλάση και 95% επιτυχία στην δεύτερη κλάση



Εικόνα 5.21: Η επίδοση του νευρωνικού δικτύου



Εικόνα 5.22: Το ιστόγραμμα σφάλματος του νευρωνικού δικτύου

Τα αποτελέσματα και στις δύο περιπτώσεις είτε με τέσσερις κόμβους είτε με έξι κόμβους είναι αρκετά ικανοποιητικά και συγκεκριμένα στην δεύτερη περίπτωση με τους έξι κόμβους, καθώς έχει κάνει μόνο ένα λάθος έναντι της πρώτης περίπτωσης όπου στην δεύτερη κατηγορία έχει κάνει επτά λάθη καθώς έχουν περάσει την διαχωριστική γραμμή που είναι στο 1.5.

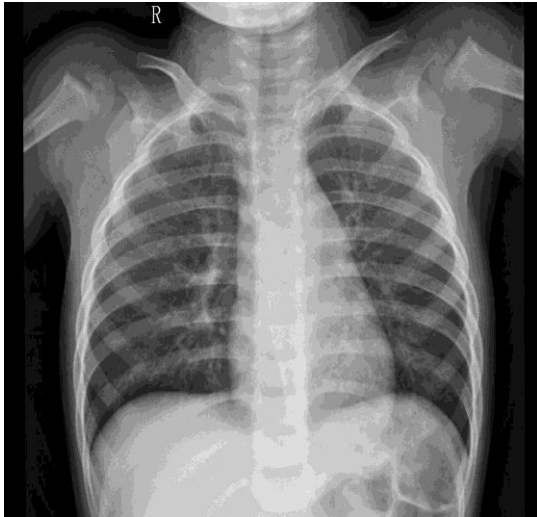
5.3 Τρίτο πείραμα

5.3.1 Στόχος του 3^{ου} πειράματος

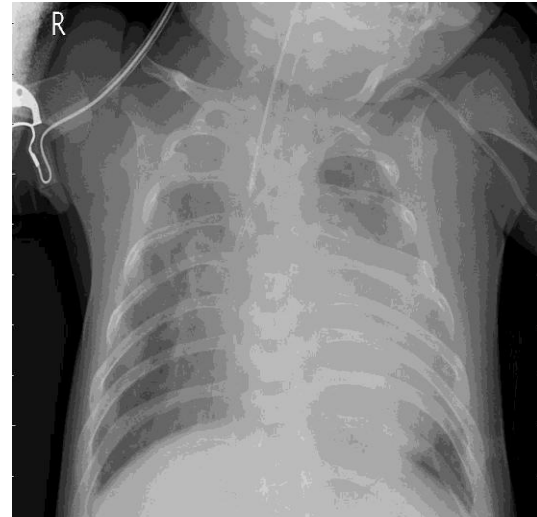
Σε αυτό το πείραμα στόχος είναι η κατηγοριοποίηση βιοϊατρικών εικόνων σε δύο κλάσεις (binary classification). Οι εικόνες προέρχονται είτε από ακτινογραφία θώρακα είτε από ακτινογραφία θώρακα με ασθένεια (πνευμονία). Εικόνες 5.1β) Στην Εικόνα 5.23 φαίνεται ένα αντιπροσωπευτικό παράδειγμα από κάθε κλάση (Εικόνες 5.23α και 5.23β).

Για την κατηγοριοποίηση γίνεται χρήση ενός «ρηχού» ΤΝΔ τύπου MLP εμπροσθοτροφοδότησης (feedforward), με ένα (1) hidden layer. Το πείραμα

πραγματοποιείται δύο φορές μεταβάλλοντας το πλήθος των κόμβων του hidden layer, (α) $q_1 = 6$ κόμβοι και (β) $q_2 = 10$ κόμβοι.



Εικόνα 5.23α: XRAY - Θώρακας υγής



Εικόνα 5.23β: XRAY - Θώρακας με πνευμονία

(Πηγή: <https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia>)

5.3.2 Data Set 3^ο πειράματος

Το data set αποτελείται από $Q_1 = 100$ εικόνες από τον θώρακα και $Q_2 = 100$ εικόνες από την πνευμονία. Οι εικόνες είναι όλες τύπου JPEG. Οι διαστάσεις των εικόνων κυμαίνονται από 344 έως 2713, το μέγεθος των αρχείων είναι από 46 KB έως 641 KB για τον θώρακα και από 20 KB έως 257 KB για την πνευμονία.

Από τις $Q = Q_1 + Q_2 = 100 + 100 = 200$ εικόνες, το 80% ήτοι $80 + 80 = 160$ εικόνες επιλέγονται τυχαία να αποτελέσουν το σύνολο εκπαίδευσης (training set) και το υπόλοιπο 20% ήτοι οι υπόλοιπες $20 + 20 = 40$ εικόνες αποτελούν το σύνολο ελέγχου (testing set).

5.3.3 Μέθοδος του 3^ο πειράματος σε βήματα

Το πείραμα εκτελείται στο Matlab™. Ο κώδικας δίνεται στο Παράρτημα Γ.

ΒΗΜΑ 1^ο Εισάγεται στο Matlab το 1^ο σετ εικόνων ($Q_1 = 100$ εικόνες θώρακα) και υπολογίζεται η μεγαλύτερη διάσταση όλων, έστω N_1 .

ΒΗΜΑ 2^ο Εφαρμόζεται σε κάθε εικόνα δισδιάστατος μετασχηματισμός Fourier, αφού όλες οι εικόνες γεμίσουν με μηδενικά (zero-padding) έως την τετραγωνική διάσταση $N1 \times N1$.

ΒΗΜΑ 3^ο Γίνεται υποδειγματοληψία κατά παράγοντα $d1=80$ στο πεδίο της συχνότητας και επιστροφή στο πεδίο του σήματος (του χώρου) με αντίστροφο δισδιάστατο μετασχηματισμό Fourier. Κάθε επεξεργασμένη εικόνα μετατρέπεται σε διάνυσμα και τοποθετείται ως μία νέα στήλη στον πίνακα με τα δεδομένα εκπαίδευσης P_TRAIN (για τις πρώτες 80 εικόνες) ή στον πίνακα με τα δεδομένα ελέγχου P_TEST (για τις τελευταίες 20 εικόνες).

ΒΗΜΑ 4^ο Εισάγεται στο Matlab το 2^ο σετ εικόνων ($Q2 = 100$ εικόνες πνευμονία), και υπολογίζεται η μεγαλύτερη διάσταση όλων, έστω $N2$.

ΒΗΜΑ 5^ο Εφαρμόζεται σε κάθε εικόνα διδιάστατος μετασχηματισμός Fourier, αφού όλες οι εικόνες επαυξηθούν με μηδενικά (zero-padding) έως την τετραγωνική διάσταση $N2 \times N2$.

ΒΗΜΑ 6^ο Γίνεται υποδειγματοληψία κατά παράγοντα $d2=d1*(N2/N1) = 132$ στο πεδίο της συχνότητας και επιστροφή στο πεδίο του σήματος (του χώρου) με αντίστροφο δισδιάστατο μετασχηματισμό Fourier. Κάθε επεξεργασμένη εικόνα μετατρέπεται σε διάνυσμα και τοποθετείται ως μία νέα στήλη στον πίνακα με τα δεδομένα εκπαίδευσης P_TRAIN (για τις πρώτες 80 εικόνες) ή στον πίνακα με τα δεδομένα ελέγχου P_TEST (για τις τελευταίες 20 εικόνες).

- Το P_TRAIN τελικά έχει 160 στήλες, μία για κάθε εικόνα στο σύνολο εκπαίδευσης,
- Το P_TEST τελικά έχει 40 στήλες, μία για κάθε εικόνα στο σύνολο ελέγχου.

ΒΗΜΑ 7^ο Δημιουργείται το σύνολο των σωστών κλάσεων (εξόδων) για το σύνολο εκπαίδευσης T_TRAIN, διάστασης 1×160 , με τιμές 80 «1» και 80 «2». Επίσης δημιουργείται το σύνολο των σωστών κλάσεων (εξόδων) για το σύνολο ελέγχου T_TEST, διάστασης 1×40 , με τιμές 20 «1» και 20 «2».

ΒΗΜΑ 8^ο Αρχικοποιείται το νευρωνικό δίκτυο (feedforwardnet) με 1 hidden layer των $q_1=6$ [$q_2=10$] κόμβων.

ΒΗΜΑ 9^ο Εκπαιδεύεται το ΤΝΔ με χρήση της συνάρτησης train και του αλγορίθμου Levenberg-Marquardt για την ελαχιστοποίηση του σφάλματος στην έξοδο του ΤΝΔ. Χρησιμοποιείται όριο 1000 επαναλήψεων, αν και σε όλες τις περιπτώσεις η εκπαίδευση του ΤΝΔ συγκλίνει (ολοκληρώνεται) πολύ νωρίτερα.

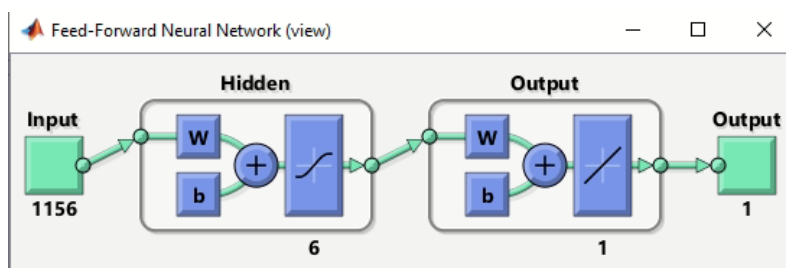
ΒΗΜΑ 10^ο Ελέγχεται η λειτουργία του ΤΝΔ χρησιμοποιώντας ως σύνολο ελέγχου το ίδιο το σύνολο εκπαίδευσης αποτελούμενο από 160 εικόνες. Μετρούνται οι σωστές και οι λάθος αποφάσεις κατηγοριοποίησης και ανάγονται σε ποσοστά %.

ΒΗΜΑ 11^ο Ελέγχεται η λειτουργία του ΤΝΔ χρησιμοποιώντας το σύνολο ελέγχου με τις $20+20 = 40$ εικόνες που δεν χρησιμοποιήθηκαν στην εκπαίδευση του ΤΝΔ (νέες, άγνωστες εικόνες). Μετρούνται οι σωστές και οι λάθος αποφάσεις κατηγοριοποίησης και ανάγονται σε ποσοστά %.

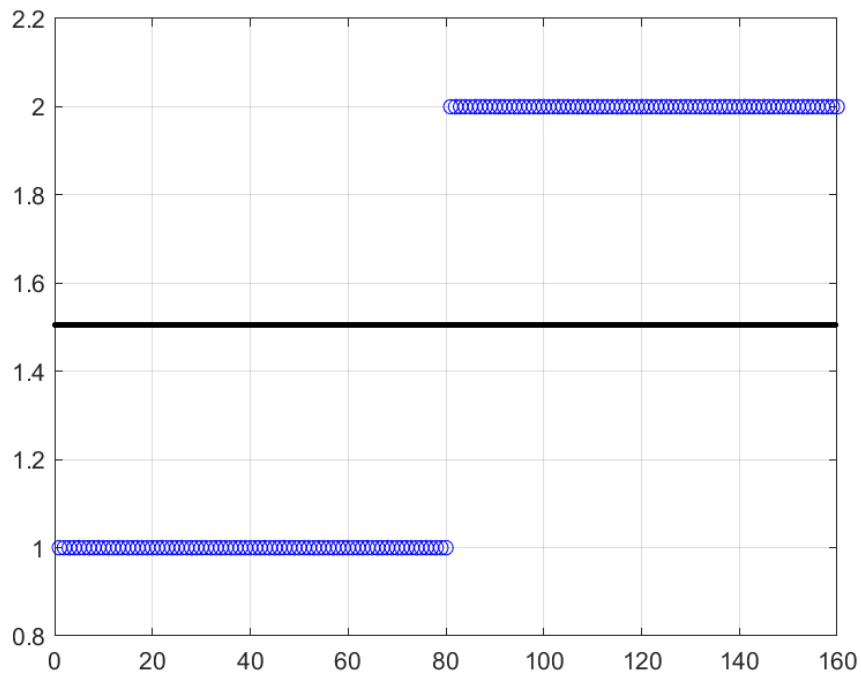
5.3.4 Αποτελέσματα του 3^{ου} πειράματος

Αποτελέσματα από το ΤΝΔ με 1 στρώμα και $q_1 = 6$ κόμβους

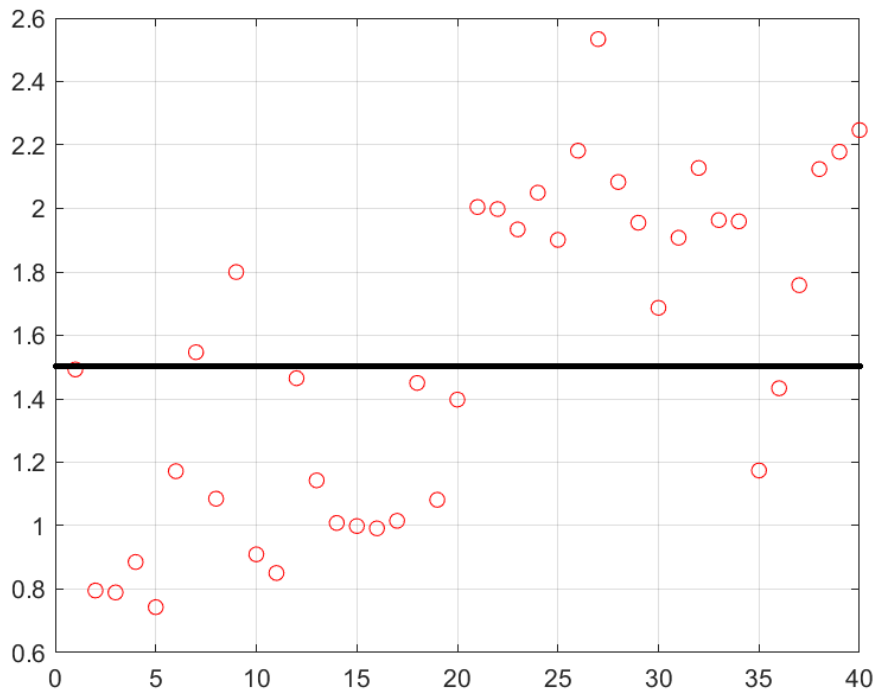
Παρακάτω παρουσιάζονται τα αποτελέσματα από το νευρωνικό δίκτυο που προέκυψαν με πρώτο τη δομή του νευρωνικού δικτύου με ένα στρώμα έξι κόμβων. Στην συνέχεια ακολουθούν τα αποτελέσματα της εκπαίδευσης και του ελέγχου αντίστοιχα, όπου έχει τοποθετηθεί μία διαχωριστική γραμμή στη μέση για την ευκρίνεια των αποτελεσμάτων. Τέλος εμφανίζεται η καμπύλη επίδοσης του νευρωνικού δικτύου, καθώς και το ιστόγραμμα σφάλματός του.



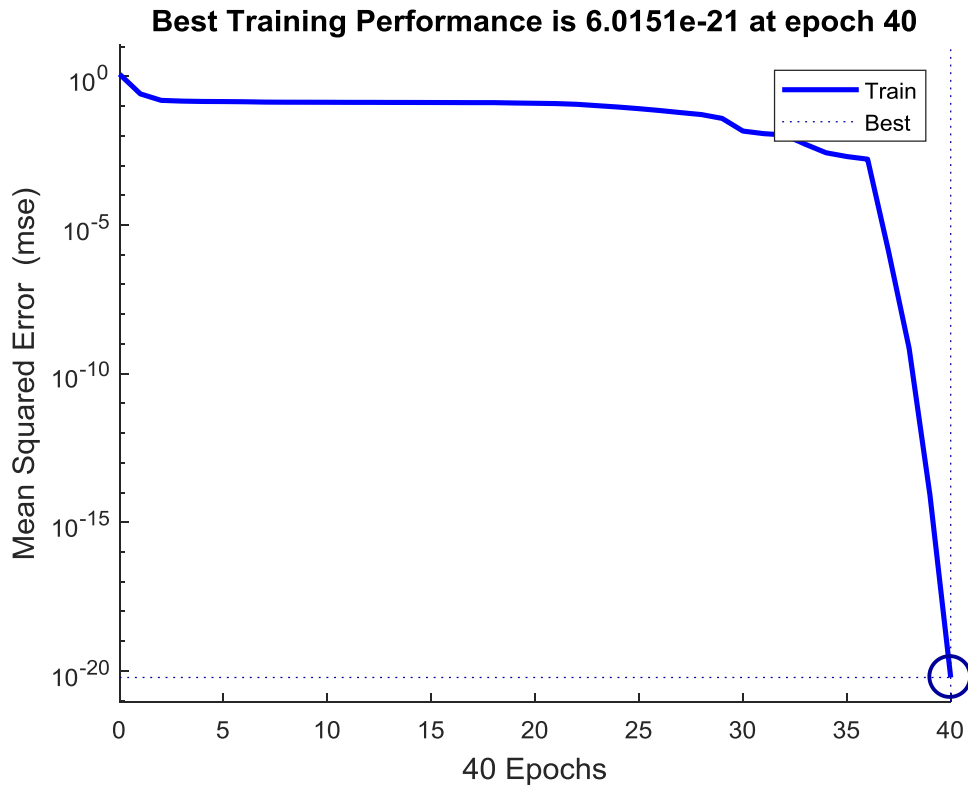
Εικόνα 5.24: Νευρωνικό δίκτυο με ένα hidden layer έξι κόμβων.



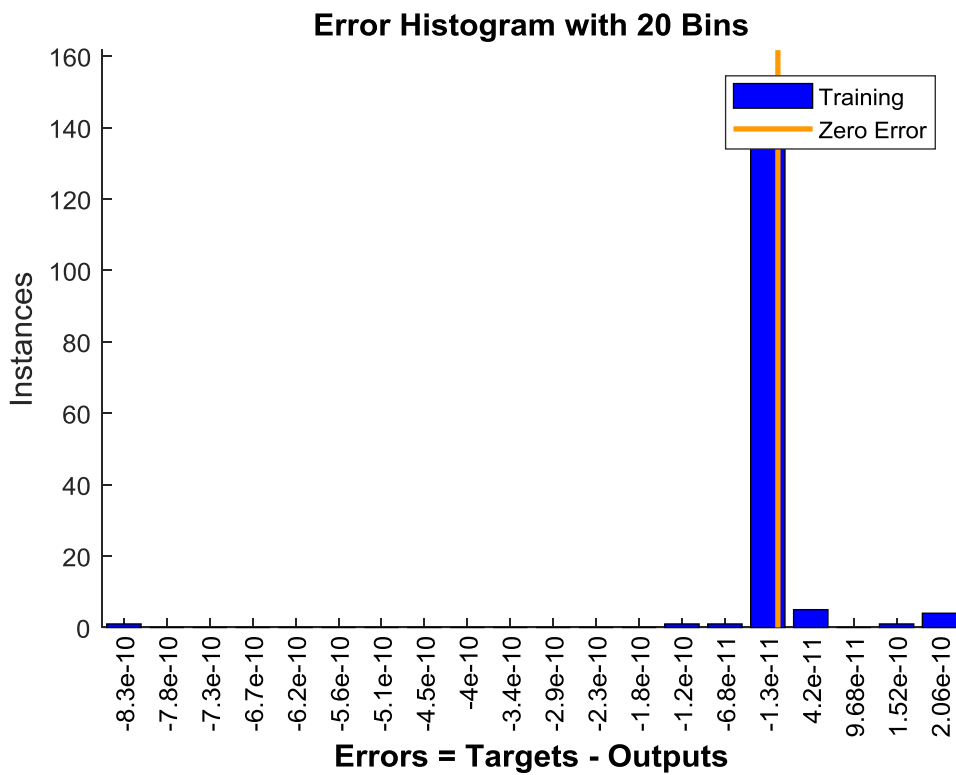
Εικόνα 5.25: Αποτέλεσμα εκπαίδευσης με 100% επιτυχία και στις δύο κατηγορίες



Εικόνα 5.26: Αποτέλεσμα ελέγχου με 90% επιτυχία και στις δύο κατηγορίες



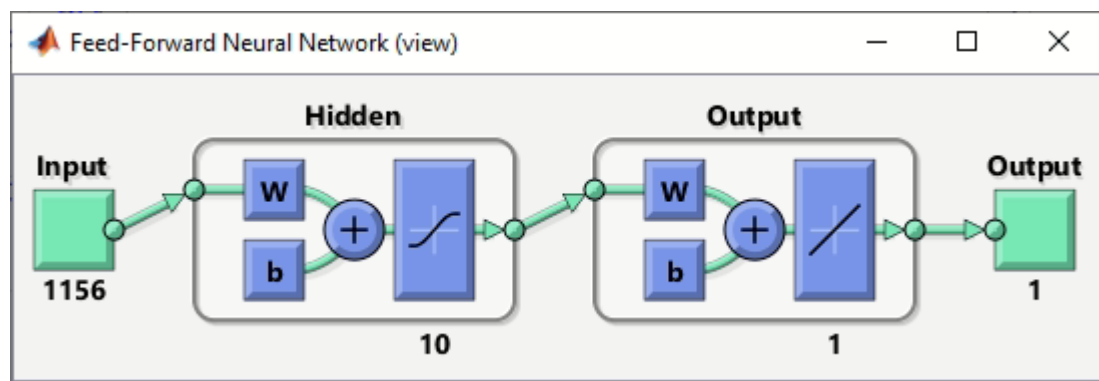
Εικόνα 5.27: Η επίδοση του νευρωνικού δικτύου



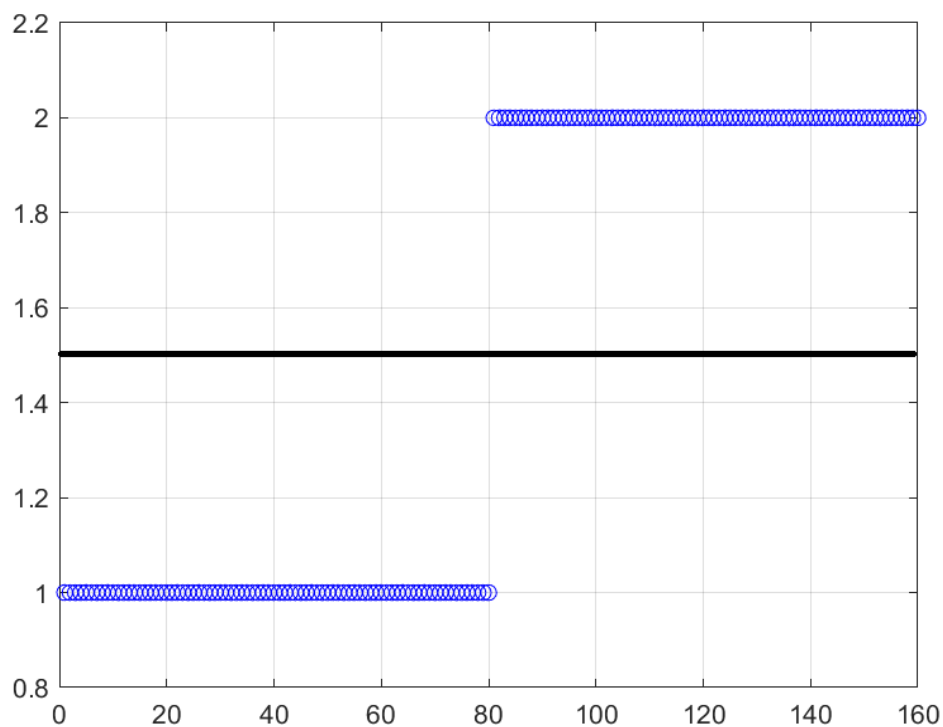
Εικόνα 5.28: Το ιστόγραμμα σφάλματος του νευρωνικού δικτύου

Αποτελέσματα από το ΤΝΔ με 1 στρώμα και $q_1 = 10$ κόμβους

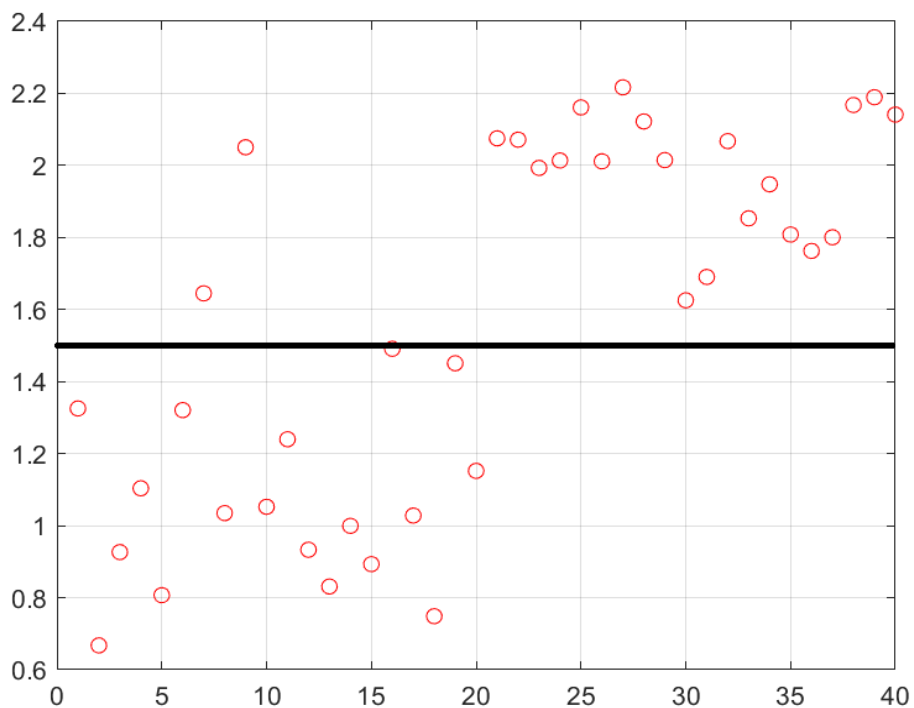
Παρακάτω παρουσιάζονται τα αποτελέσματα από το νευρωνικό δίκτυο που προέκυψαν με πρώτο τη δομή του νευρωνικού δικτύου με ένα στρώμα δέκα κόμβων. Στην συνέχεια ακολουθούν τα αποτελέσματα της εκπαίδευσης και του ελέγχου αντίστοιχα, όπου έχει τοποθετηθεί μία διαχωριστική γραμμή στη μέση για την ευκρίνεια των αποτελεσμάτων. Τέλος εμφανίζεται η καμπύλη επίδοσης του νευρωνικού δικτύου, καθώς και το ιστόγραμμα σφάλματός του.



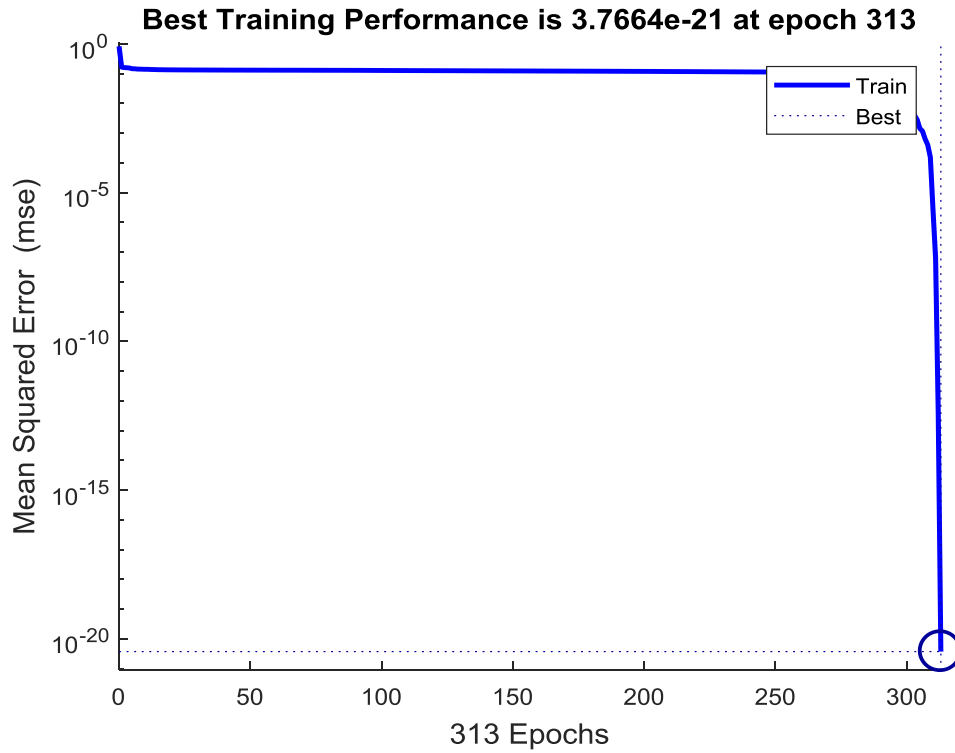
Εικόνα 5.29: Νευρωνικό δίκτυο με ένα hidden layer δέκα κόμβων.



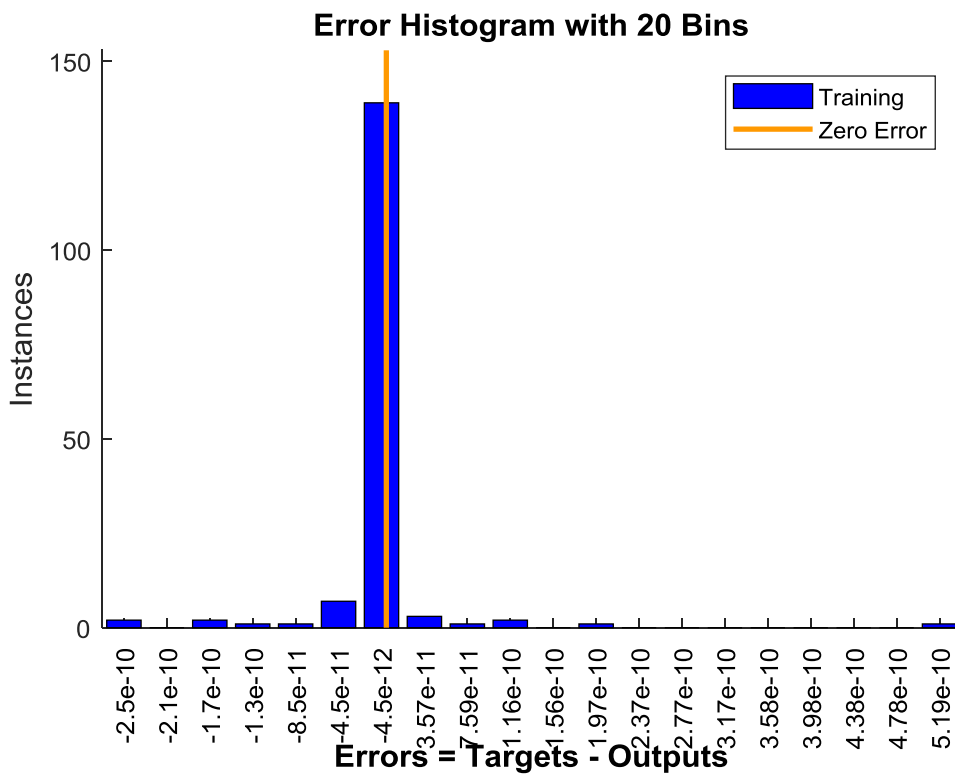
Εικόνα 5.30: Αποτέλεσμα εκπαίδευσης με 100% επιτυχία και στις δύο κατηγορίες



Εικόνα 5.31: Αποτέλεσμα ελέγχου με 90% επιτυχία στην πρώτη κλάση και 100% επιτυχία στην δεύτερη κλάση



Εικόνα 5.32: Η επίδοση του νευρωνικού δικτύου



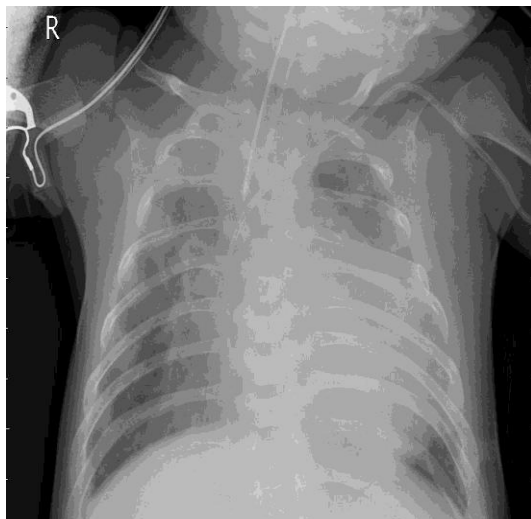
Εικόνα 5.33: Το ιστόγραμμα σφάλματος του νευρωνικού δικτύου

Τα αποτελέσματα και στις δύο περιπτώσεις είτε με έξι κόμβους είτε με δέκα κόμβους είναι αρκετά ικανοποιητικά με 90% επιτυχία και πάνω και στις δύο κατηγορίες.

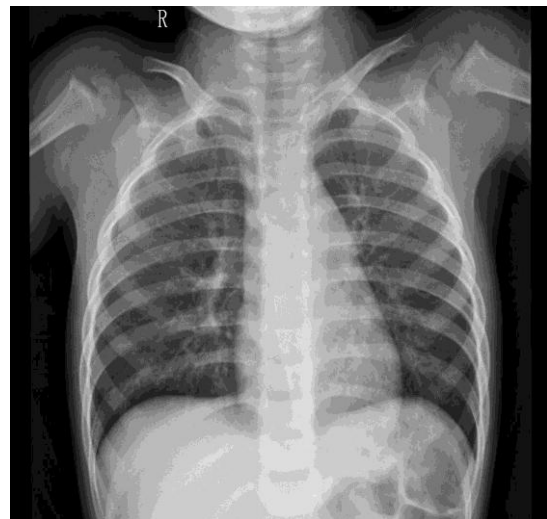
5.4 Τέταρτο πείραμα

5.4.1 Στόχος του 4^{ου} πειράματος

Στο πείραμα αυτό στόχος είναι η κατηγοριοποίηση βιοιατρικών εικόνων σε δύο κλάσεις (binary classification). Οι εικόνες προέρχονται είτε από ακτινογραφία θώρακα με ασθένεια (πνευμονία) είτε από ακτινογραφία θώρακα. Στην Εικόνα 5.34 φαίνεται ένα αντιπροσωπευτικό παράδειγμα από κάθε κλάση (Εικόνες 5.34α και 5.34β). Για την κατηγοριοποίηση γίνεται χρήση ενός «ρηχού» ΤΝΔ τύπου MLP εμπροσθοτροφοδότησης (feedforward), με ένα (1) hidden layer όπως στο προηγούμενο πείραμα με την διαφορά ότι στην προεπεξεργασία των εικόνων προστίθενται και δύο επιπλέον στοιχεία, η διέλευση από βαθυπερατό φίλτρο και η κατάτμηση με τη μέθοδο ανάπτυξης περιοχής (region growing). Το πείραμα πραγματοποιείται δύο φορές μεταβάλλοντας το πλήθος των κόμβων του hidden layer, (α) $q_1 = 6$ κόμβοι και (β) $q_2 = 10$ κόμβοι.



Εικόνα 5.34α: XRAY - Θώρακας με πνευμονία



Εικόνα 5.34β: XRAY - Θώρακας υγιής

(Πηγή: <https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia>)

5.4.2 Data Set 4^ο πειράματος

Το Data set που χρησιμοποιείται είναι το ίδιο με αυτό του τρίτου πειράματος.

5.4.3 Μέθοδος του 4^ο πειράματος σε βήματα

Το πείραμα εκτελείται στο Matlab™. Ο κώδικας δίνεται στο Παράρτημα Δ.

ΒΗΜΑ 1^ο Εισάγεται στο Matlab το 1^ο σετ εικόνων ($Q1 = 100$ εικόνες με πνευμονία) και υπολογίζεται η μεγαλύτερη διάσταση όλων, έστω $N1$.

ΒΗΜΑ 2^ο Εφαρμόζεται σε κάθε εικόνα δισδιάστατος μετασχηματισμός Fourier, αφού όλες οι εικόνες γεμίσουν με μηδενικά (zero-padding) έως την τετραγωνική διάσταση $N1 \times N1$.

ΒΗΜΑ 3^ο Δημιουργείται η κυκλική συμμετρία του κύκλου που θα χρησιμοποιηθεί στο βαθυπερατό φίλτρο στη συνέχεια, με διάμετρο $D1=50$.

ΒΗΜΑ 4^ο Γίνεται μετατροπή από έγχρωμες εικόνες, σε εικόνες επιπέδου του γκρι όσες απαιτούνται, τοποθετείται ένα σημείο κεντρικά κάθε εικόνας και πραγματοποιείται η διαδικασία ανάπτυξης περιοχής (region growing) μαζί με ένα δισδιάστατο βαθυπερατό φίλτρο αμέσως μετά. Ακολουθεί η υποδειγματοληψία κατά παράγοντα $d1=30$ στο πεδίο της συχνότητας και επιστροφή στο πεδίο του σήματος (του χώρου) με αντίστροφο δισδιάστατο μετασχηματισμό Fourier. Κάθε επεξεργασμένη εικόνα μετατρέπεται σε διάνυσμα και τοποθετείται ως μία νέα στήλη στον πίνακα με τα δεδομένα εκπαίδευσης P_TRAIN (για τις πρώτες 80 εικόνες) ή στον πίνακα με τα δεδομένα ελέγχου P_TEST (για τις τελευταίες 20 εικόνες).

ΒΗΜΑ 5^ο Εισάγεται στο Matlab το 2^ο σετ εικόνων ($Q2 = 100$ εικόνες θώρακα), και υπολογίζεται η μεγαλύτερη διάσταση όλων, έστω $N2$.

ΒΗΜΑ 6^ο Εφαρμόζεται σε κάθε εικόνα διδιάστατος μετασχηματισμός Fourier, αφού όλες οι εικόνες επαυξηθούν με μηδενικά (zero-padding) έως την τετραγωνική διάσταση $N2 \times N2$.

ΒΗΜΑ 7^ο Γίνεται υποδειγματοληψία κατά παράγοντα $d2=d1*(N2/N1) = 41$ στο πεδίο της συχνότητας και επιστροφή στο πεδίο του σήματος (του χώρου) με αντίστροφο δισδιάστατο μετασχηματισμό Fourier. Κάθε

επεξεργασμένη εικόνα μετατρέπεται σε διάνυσμα και τοποθετείται ως μία νέα στήλη στον πίνακα με τα δεδομένα εκπαίδευσης P_TRAIN (για τις πρώτες 80 εικόνες) ή στον πίνακα με τα δεδομένα ελέγχου P_TEST (για τις τελευταίες 20 εικόνες).

- Το P_TRAIN τελικά έχει 160 στήλες, μία για κάθε εικόνα στο σύνολο εκπαίδευσης,
- Το P_TEST τελικά έχει 40 στήλες, μία για κάθε εικόνα στο σύνολο ελέγχου.

ΒΗΜΑ 8^ο Δημιουργείται το σύνολο των σωστών κλάσεων (εξόδων) για το σύνολο εκπαίδευσης T_TRAIN, διάστασης 1x160, με τιμές 80 «1» και 80 «2». Επίσης δημιουργείται το σύνολο των σωστών κλάσεων (εξόδων) για το σύνολο ελέγχου T_TEST, διάστασης 1x40, με τιμές 20 «1» και 20 «2».

ΒΗΜΑ 9^ο Αρχικοποιείται το νευρωνικό δίκτυο (feedforwardnet) με 1 hidden layer των $q_1=4$ [$q_2=6$] κόμβων.

ΒΗΜΑ 10^ο Εκπαιδύεται το ΤΝΔ με χρήση της συνάρτησης train και του αλγορίθμου Levenberg-Marquardt για την ελαχιστοποίηση του σφάλματος στην έξοδο του ΤΝΔ. Χρησιμοποιείται όριο 1000 επαναλήψεων, αν και σε όλες τις περιπτώσεις η εκπαίδευση του ΤΝΔ συγκλίνει (ολοκληρώνεται) πολύ νωρίτερα.

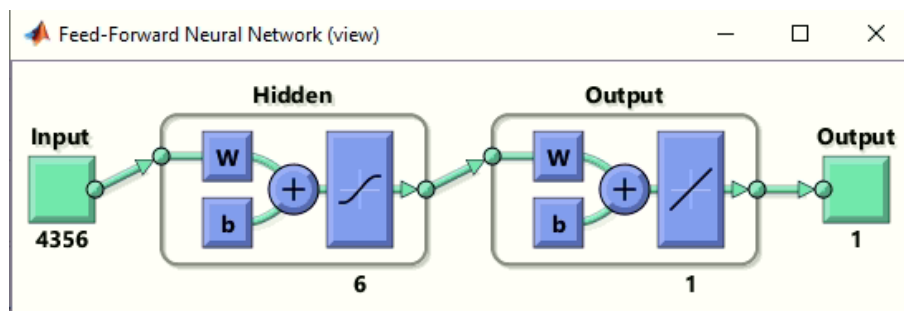
ΒΗΜΑ 11^ο Ελέγχεται η λειτουργία του ΤΝΔ χρησιμοποιώντας ως σύνολο ελέγχου το ίδιο το σύνολο εκπαίδευσης αποτελούμενο από 160 εικόνες. Μετρούνται οι σωστές και οι λάθος αποφάσεις κατηγοριοποίησης και ανάγονται σε ποσοστά %.

ΒΗΜΑ 12^ο Ελέγχεται η λειτουργία του ΤΝΔ χρησιμοποιώντας το σύνολο ελέγχου με τις $20+20 = 40$ εικόνες που δεν χρησιμοποιήθηκαν στην εκπαίδευση του ΤΝΔ (νέες, άγνωστες εικόνες). Μετρούνται οι σωστές και οι λάθος αποφάσεις κατηγοριοποίησης και ανάγονται σε ποσοστά %.

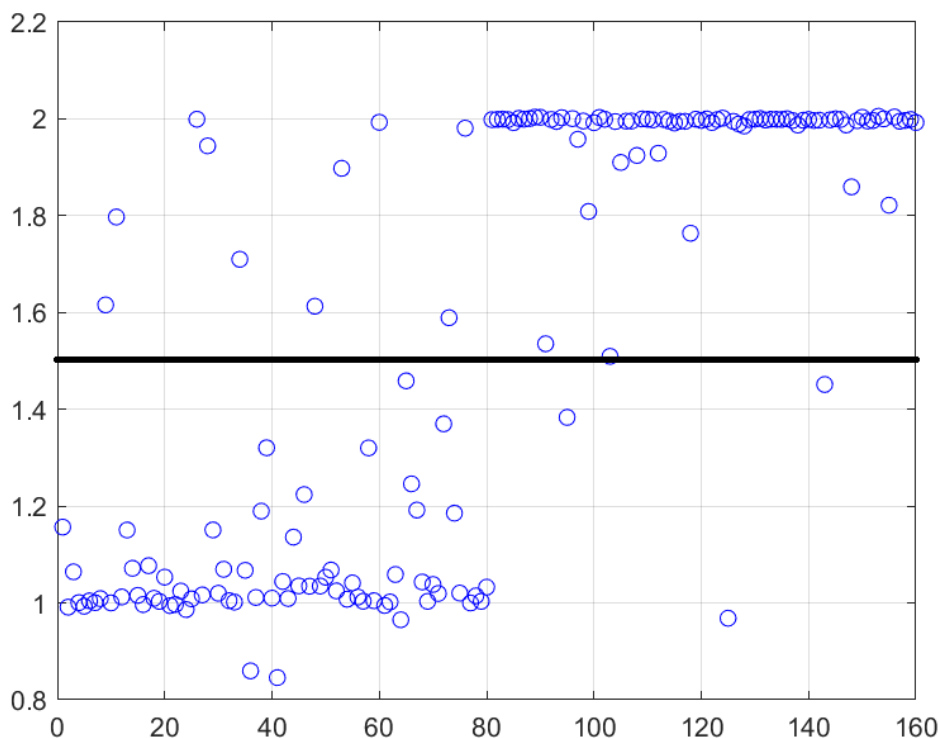
5.4.4 Αποτελέσματα του 4^{ου} πειράματος

Αποτελέσματα από το ΤΝΔ με 1 στρώμα και $q_1 = 6$ κόμβους

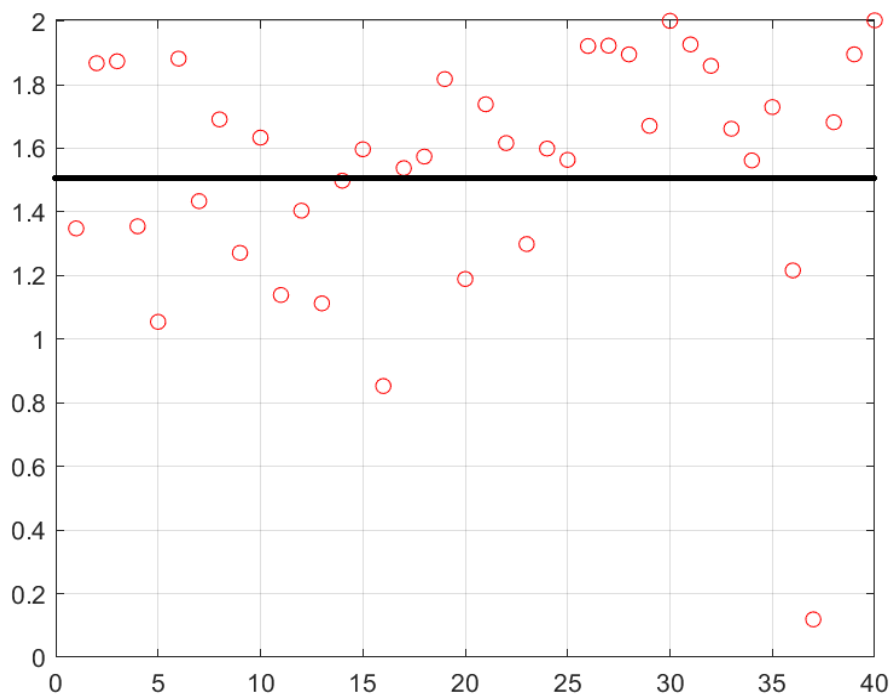
Παρακάτω παρουσιάζονται τα αποτελέσματα κατηγοριοποίησης από το νευρωνικό δίκτυο της Εικόνας 5.35, με $q_1 = 6$ κόμβους στο πρώτο στρώμα. Στην συνέχεια ακολουθούν τα αποτελέσματα της εκπαίδευσης και του ελέγχου αντίστοιχα, όπου έχει τοποθετηθεί μία διαχωριστική γραμμή στη μέση για την ευκρίνεια των αποτελεσμάτων. Τέλος εμφανίζεται η καμπύλη επίδοσης του νευρωνικού δικτύου, καθώς και το ιστόγραμμα σφάλματός του.



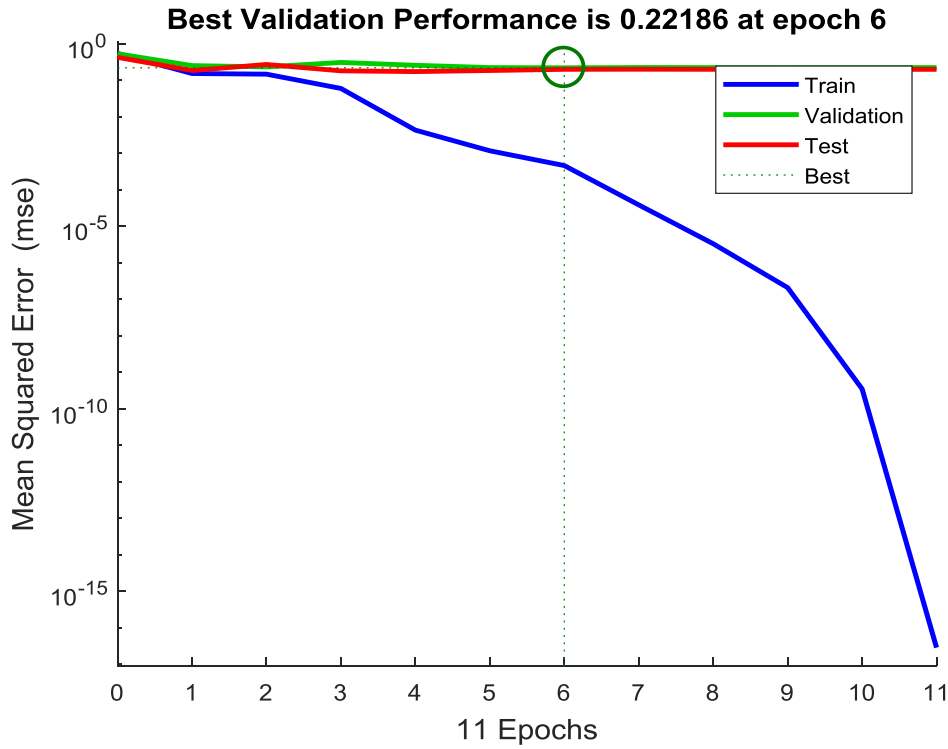
Εικόνα 5.35: Νευρωνικό δίκτυο με 1 hidden layer των έξι κόμβων.



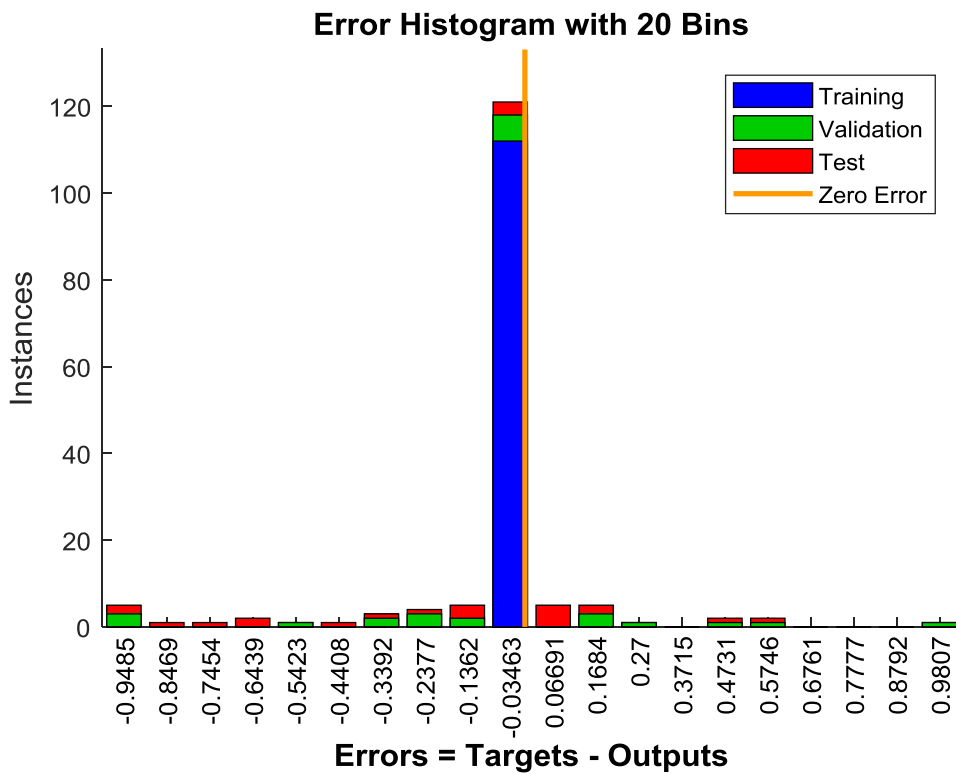
Εικόνα 5.36: Αποτέλεσμα εκπαίδευσης με 87.5% επιτυχία στην πρώτη κλάση και 96.25% επιτυχία στην δεύτερη κλάση



Εικόνα 5.37: Αποτέλεσμα ελέγχου με 55% επιτυχία στην πρώτη κλάση και 90% επιτυχία στην δεύτερη κλάση



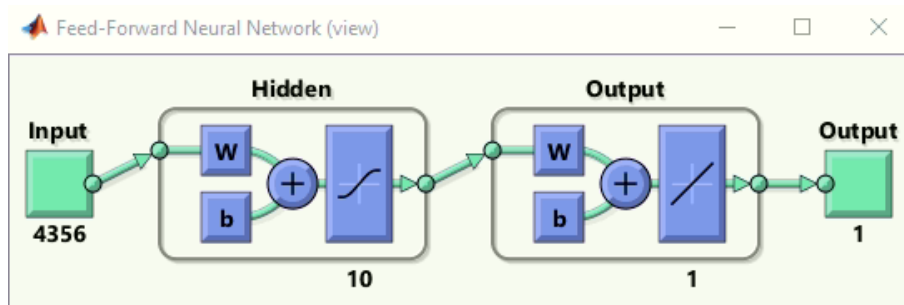
Εικόνα 5.38: Η επίδοση του νευρωνικού δικτύου



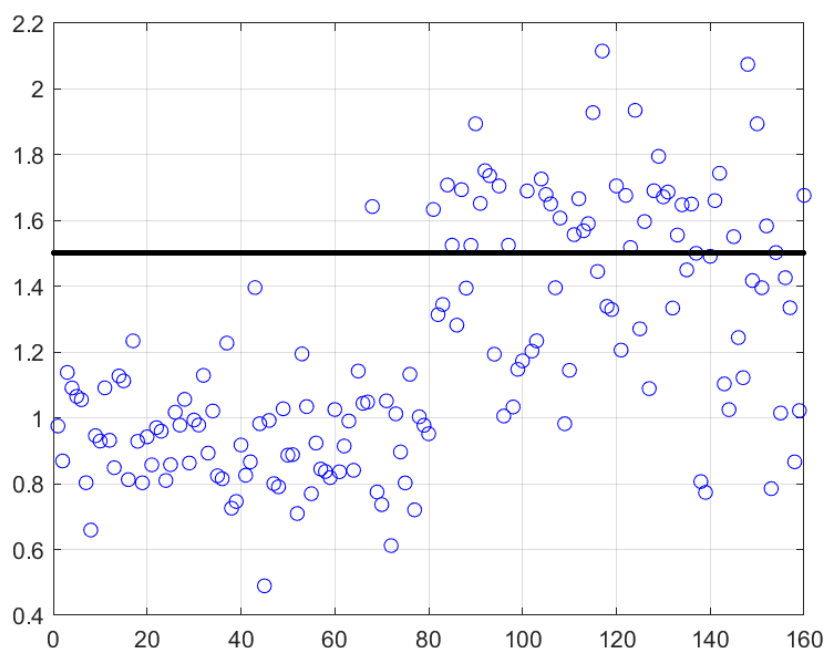
Εικόνα 5.39: Το ιστόγραμμα σφάλματος του νευρωνικού δικτύου

Αποτελέσματα από το ΤΝΔ με 1 στρώμα και $q_1 = 10$ κόμβους

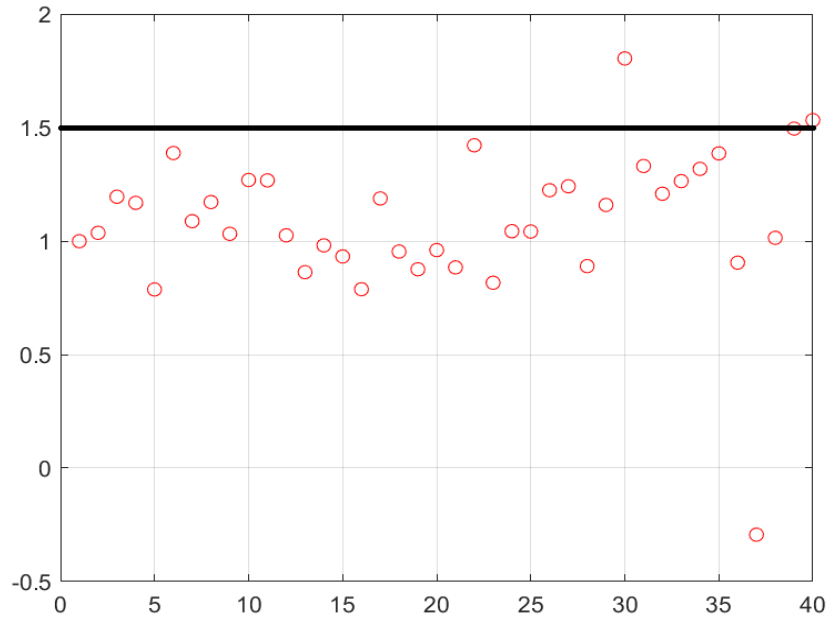
Παρακάτω παρουσιάζονται τα αποτελέσματα κατηγοριοποίησης από το νευρωνικό δίκτυο της Εικόνας 5.40, με ένα στρώμα των $q_2 = 10$ κόμβων. Στην συνέχεια ακολουθούν τα αποτελέσματα της εκπαίδευσης και του ελέγχου αντίστοιχα, όπου έχει τοποθετηθεί μία διαχωριστική γραμμή στη μέση για την ευκρίνεια των αποτελεσμάτων. Τέλος εμφανίζεται η καμπύλη επίδοσης του νευρωνικού δικτύου, καθώς και το ιστόγραμμα σφάλματός του.



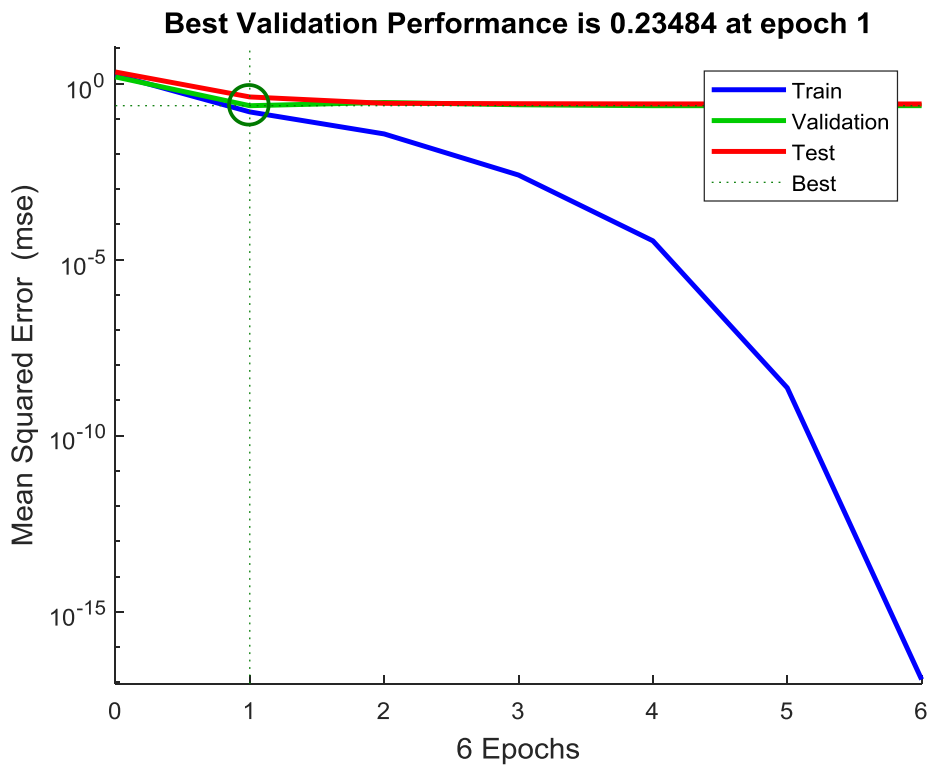
Εικόνα 5.40: Νευρωνικό δίκτυο με 1 hidden layer των δέκα κόμβων.



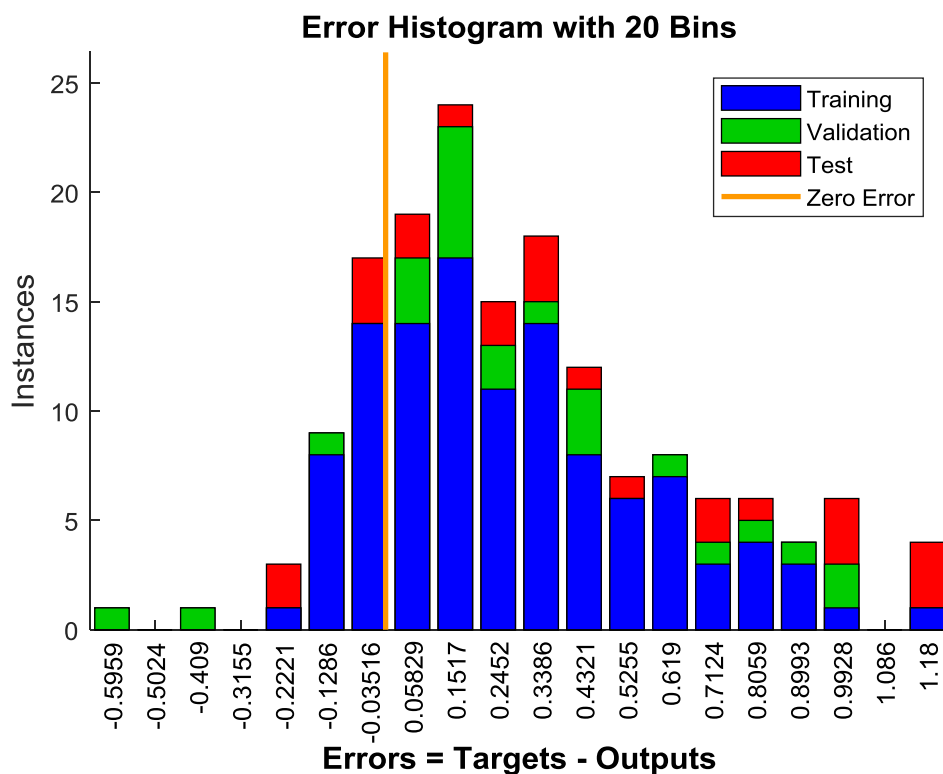
Εικόνα 5.41: Αποτέλεσμα εκπαίδευσης με 98.75% επιτυχία στην πρώτη κλάση και 53.75% επιτυχία στην δεύτερη κλάση



Εικόνα 5.42: Αποτέλεσμα ελέγχου με 100% επιτυχία στην πρώτη κλάση και 10% επιτυχία στην δεύτερη κλάση



Εικόνα 5.43: Η επίδοση του νευρωνικού δικτύου



Εικόνα 5.44: Το ιστόγραμμα σφάλματος του νευρωνικού δικτύου.

Τα αποτελέσματα και στις δύο περιπτώσεις είτε με έξι κόμβους είτε με δέκα κόμβους δεν είναι ικανοποιητικά καθώς υπάρχουν αρκετά λάθη, κυρίως στην δεύτερη κλάση της δεύτερης περίπτωσης με 18 λάθη στα 20.

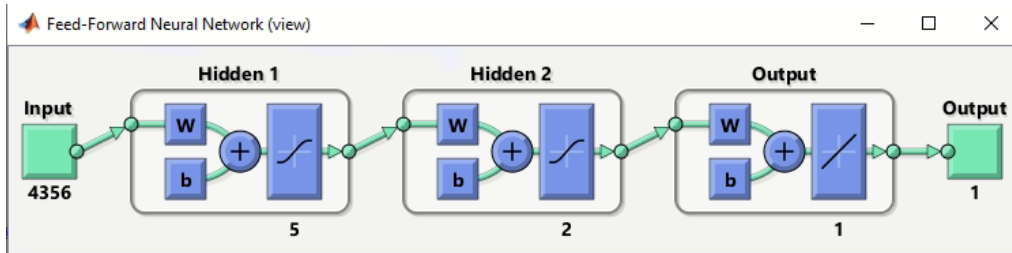
5.4.5 Βελτίωση 4^ο πειράματος

Για βελτίωση αυτού του πειράματος χρησιμοποιείται ένα νευρωνικό δίκτυο με δύο hidden layers, δηλαδή ένα βαθύ νευρωνικό δίκτυο, με το πρώτο layer να έχει 5 κόμβους και το δεύτερο layer να έχει 2 κόμβους, με όλα τα υπόλοιπα στοιχεία του πειράματος (data set, κλάσεις) να παραμένουν ίδια.

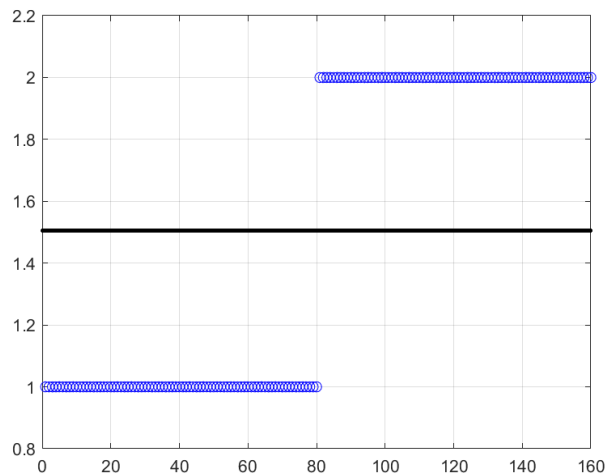
Αποτελέσματα από το TNA με 2 στρώματα $q_1 = 5$ κόμβους και $q_2 = 2$ κόμβους

Παρακάτω παρουσιάζονται τα αποτελέσματα κατηγοριοποίησης από το βαθύ νευρωνικό δίκτυο της Εικόνας 5.45, με δύο στρώματα των 5 και 2 κόμβων, αντίστοιχα. Στην συνέχεια ακολουθούν τα αποτελέσματα της εκπαίδευσης και του

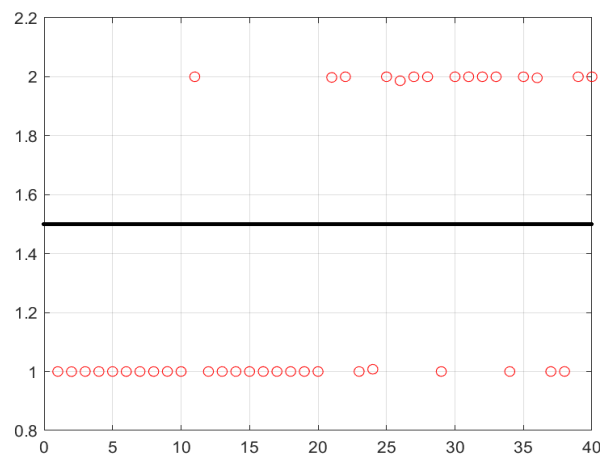
ελέγχου αντίστοιχα, όπου έχει τοποθετηθεί μία διαχωριστική γραμμή στη μέση για την ευκρίνεια των αποτελεσμάτων. Τέλος εμφανίζεται η καμπύλη επίδοσης του νευρωνικού δικτύου, καθώς και το ιστόγραμμα σφάλματός του.



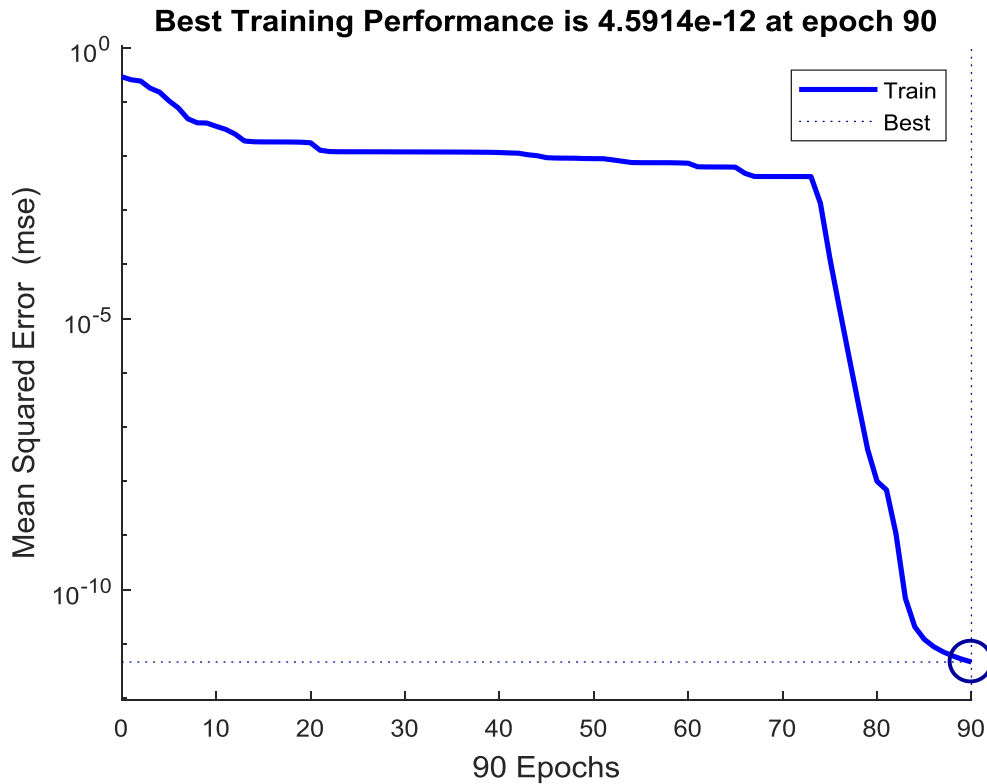
Εικόνα 5.45: Νευρωνικό δίκτυο με 2 hidden layers των πέντε και δύο κόμβων, αντίστοιχα.



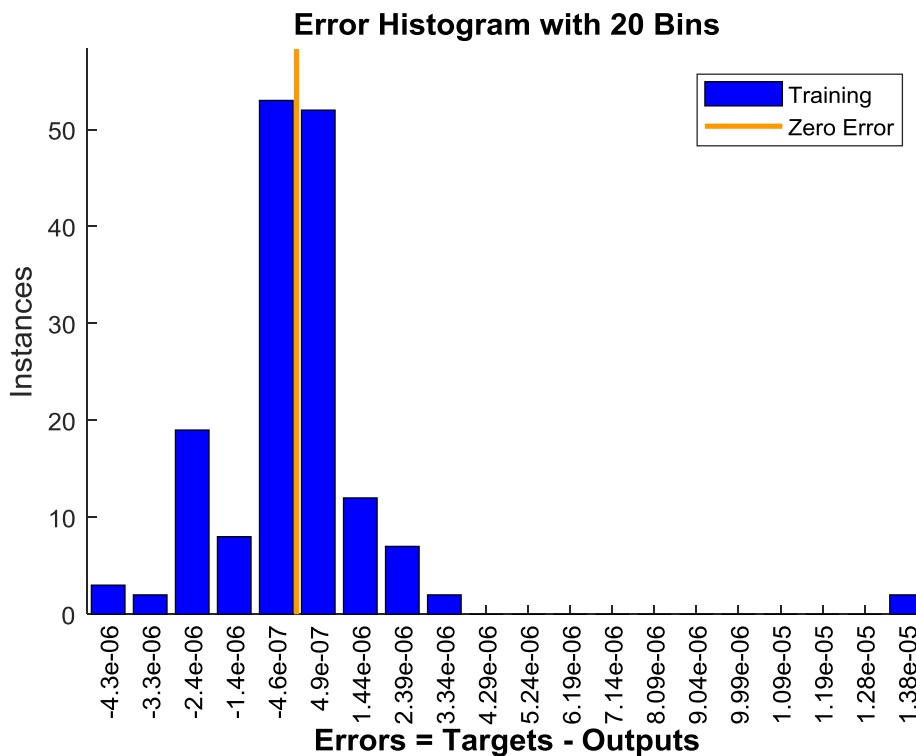
Εικόνα 5.46: Αποτέλεσμα εκπαίδευσης με 100% επιτυχία και στις δύο κατηγορίες



Εικόνα 5.47: Αποτέλεσμα ελέγχου με 95% επιτυχία στην πρώτη κλάση και 70% επιτυχία στην δεύτερη κλάση



Εικόνα 5.48: Η επίδοση του νευρωνικού δικτύου



Εικόνα 5.49: Το ιστόγραμμα σφάλματος του νευρωνικού δικτύου

Το αποτέλεσμα έχει εμφανώς βελτιωθεί σε σχέση με το νευρωνικό δίκτυο 1 στρώματος των δέκα κόμβων που εφαρμόστηκε αρχικά στο 4^ο πείραμα. Ειδικά στη 2^η κλάση, το ποσοστό επιτυχίας από το 10% ανέβηκε στο 70% περιορίζοντας σε μεγάλο βαθμό τις λάθος κατηγοριοποιήσεις των εικόνων της κλάσης 2.

Συνοψίζοντας, για την βελτίωση του 4^{ου} Πειράματος χρησιμοποιήθηκε νευρωνικό δίκτυο με παραπάνω από ένα hidden layers, δηλαδή ένα **deep neural network (βαθύ νευρωνικό δίκτυο)**, με δύο στρώματα πέντε και δύο κόμβων. Αντίθετα, όλα τα προηγούμενα αποτελέσματα που στηρίχθηκαν σε **shallow neural networks (ρηχά νευρωνικά δίκτυα)** με ένα μόνο hidden layer. Παρότι το ΤΝΔ με 1 στρώμα των δέκα κόμβων είναι αντίστοιχο σε πολυπλοκότητα με το ΤΝΔ με 2 στρώματα των πέντε και των δύο κόμβων, αντίστοιχα, καθώς από πλευράς πλήθους συντελεστών είναι ισοδύναμα ($1 \times 10 = 5 \times 2$), τα αποτελέσματα στη δεύτερη περίπτωση που το δίκτυο έχει αποκτήσει 'βάθος' (δηλαδή έχει εισαχθεί και 2^ο hidden layers) είναι σαφώς βελτιωμένα. Συγκρίνοντας τα πειραματικά αυτά αποτελέσματα, διαπιστώνεται ότι η μετάβαση από τα απλά (ρηχά) στα βαθιά νευρωνικά δίκτυα, δηλαδή η αύξηση του βάθους του δικτύου με προσθήκη περισσότερων hidden layers και ενδεχομένως με λιγότερους κόμβους το καθένα, επιφέρει βελτίωση της επίδοσης του δικτύου στο δεδομένο πρόβλημα.

Κεφάλαιο 6: Συμπεράσματα – Περαιτέρω έρευνα

Στη διπλωματική αυτή εργασία πραγματοποιήθηκε μελέτη στο πεδίο της ψηφιακής επεξεργασίας και κατηγοριοποίησης βιοϊατρικών εικόνων με τη χρήση νευρωνικών δικτύων. Στο πεδίο της ψηφιακής επεξεργασίας εικόνων μελετήθηκαν διάστατα φίλτρα χωρικών συχνοτήτων για τη βελτίωση των εικόνων, καθώς και η κατάτμηση εικόνας με ανάπτυξη περιοχών (region growing). Στη συνέχεια οργανώθηκαν και εκτελέστηκαν τέσσερα πειράματα κατηγοριοποίησης ψηφιακά επεξεργασμένων βιοϊατρικών εικόνων, με στόχο τη δυαδική κατηγοριοποίηση. Χρησιμοποιήθηκαν αρχικά απλά ('ρηγά') και στη συνέχεια 'βαθιά' νευρωνικά δίκτυα, με ικανοποιητικά αποτελέσματα. Η ανάλυση και η επεξεργασία των εικόνων, και τα πειράματα κατηγοριοποίησης έγιναν στο περιβάλλον προσομοίωσης Matlab ®.

Ωστόσο απομένει να γίνει αρκετή δουλειά ακόμα για την περαιτέρω βελτίωση των αποτελεσμάτων, καθώς είναι τεράστιο το πλήθος των παραμέτρων ενός νευρωνικού δικτύου βαθιάς μάθησης που μπορούν να ρυθμιστούν (συνδυασμών στρωμάτων και πλήθος κόμβων ανά στρώμα, αλλά και λοιπές παράμετροι εκπαίδευσης του δικτύου όπως αλγόριθμος ελαχιστοποίησης, κριτήριο ελαχιστοποίησης, κανόνες τερματισμού, κ.α.). Όπως υποστηρίζεται και από τις σχετικές δημοσιευμένες έρευνες, αν χρησιμοποιηθούν βαθιά νευρωνικά δίκτυα ακόμα πιο σύνθετης αρχιτεκτονικής, είναι αναμενόμενο οι επιδόσεις να βελτιωθούν ακόμα περισσότερο.

Βιβλιογραφία - Αναφορές - Διαδικτυακές Πηγές

- [1] Luis Rueda Domingo Mery Josef Kittler (Eds.), “Progress in Pattern Recognition, Image Analysis and Applications”, 12th Iberoamerican Congress on Pattern Recognition, CIARP 2007 Vina del Mar-Valparaiso, Chile, November 13-16, 2007 Proceedings.
- [2] Simon Haykin, “Neural Networks and Learning Machines”, 3rd edition.
- [3] Sameera V. Mohd Sagheer, Sudhish N. George, “Biomedical Signal Processing and Control journal homepage: www.elsevier.com/locate/bspc A review” from <https://www.journals.elsevier.com/biomedical-signal-processing-and-control>
- [4] Ziad Alqadi, “Salt and Pepper Noise: Effects and Removal”, July 2018.
- [5] Subrato Bharati, Tanvir Zaman Khan, Prajoy Podder, Nguyen Quoc Hung, “A comparative analysis of image denoising problem: noise models, denoising filters and applications”, 4 June 2020.
- [6] Anil K. Jain, “Fundamentals of Digital Image Processing”, 1997.
- [7] Rafael C. Gonzalez, Paul Wintz, “Digital Image Processing”, 2nd edition.
- [8] Raman Maini and Himanshu Aggarwal, “A Comprehensive Review of Image Enhancement Techniques” from <https://arxiv.org/ftp/arxiv/papers/1003/1003.4053.pdf>.

Διαδικτυακές πηγές (Links)

- [1] kelifos.physics.auth.gr, URL: <http://kelifos.physics.auth.gr/COURSES/neural/K1.pdf>
- [2] en.wikipedia.org, URL: https://en.wikipedia.org/wiki/Artificial_neural_network
- [3] medium.com, URL: <https://medium.com/@jamesdacombe/an-introduction-to-artificial-neural-networks-with-example-ad459bb6941b>
- [4] data-flair.training, URL: <https://data-flair.training/blogs/artificial-neural-networks-for-machine-learning/>
- [5] en.wikipedia.org, URL: https://en.wikipedia.org/wiki/General_regression_neural_network
- [6] medium.com, URL: <https://towardsdatascience.com/self-organizing-maps-ff5853a118d4>
- [7] tutorialspoint.com,
URL: https://www.tutorialspoint.com/artificial_neural_network/artificial_neural_network_learning_vector_quantization.htm

- [8] medium.com, URL: <https://medium.com/analytics-vidhya/brief-history-of-neural-networks-44c2bf72eec>
- [9] smartsheet.com, URL: <https://www.smartsheet.com/neural-network-applications>
- [10] lab.fs.uni-lj.si, URL: http://lab.fs.uni-lj.si/lasin/wp/IMIT_files/neural/NN-examples.pdf
- [11] medium.com URL: https://medium.com/@AI_with_Kain/understanding-of-multilayer-perceptron-mlp-8f179c4a135f
- [12] medium.com URL: <https://medium.com/@jayeshbahire/the-xor-problem-in-neural-networks-50006411840b>
- [13] csc.com.gr, URL: <https://www.csc.com.gr/machine-learning-%CE%BC%CE%B7%CF%87%CE%B1%CE%BD%CE%B9%CE%BA%CE%AE-%CE%BC%CE%AC%CE%B8%CE%B7%CF%83%CE%B7-%CF%84%CE%B9-%CE%B5%CE%AF%CE%BD%CE%B1%CE%B9/>
- [14] datasciencecentral.com, URL: <https://www.datasciencecentral.com/profiles/blogs/artificial-neural-network-ann-in-machine-learning>
- [15] guru99.com, URL: <https://www.guru99.com/what-is-fuzzy-logic.html>
- [16] analyticsvidhya.com, URL: <https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/>
- [17] en.wikipedia.org, URL: https://en.wikipedia.org/wiki/Genetic_algorithm
- [18] geneticprogramming.com, URL: <http://geneticprogramming.com/>
- [19] insider.gr, URL: <https://www.insider.gr/epiheiriseis/tehnologia/92698/10-kathimerines-efarmoges-tis-vathias-mathisis>
- [20] bernardmarr.com, URL: <https://bernardmarr.com/default.asp?contentID=1789>
- [21] neuralnetworksanddeeplearning.com,
URL: <http://neuralnetworksanddeeplearning.com/chap6.html>
- [22] sciencedirect.com, URL: <https://www.sciencedirect.com/topics/earth-and-planetary-sciences/image-enhancement>
- [23] sciencedirect.com, URL: <https://www.sciencedirect.com/topics/computer-science/image-enhancement>
- [24] ai.stanford.edu, URL: <https://ai.stanford.edu/~syyeung/cvweb/tutorial3.html>
- [25] link.springer.com, URL: <https://link.springer.com/article/10.1007/s11517-018-1819-y>

[26] originlab.com URL: <https://www.originlab.com/doc/Origin-Help/2DFFT-Filter-Algorithm>

[27] fourier.eng.hmc.edu URL:
http://fourier.eng.hmc.edu/e161/lectures/smooth_sharpen/node2.html

[28] cse.unr.edu URL: <https://www.cse.unr.edu/~bebis/CS791E/Notes/RegionGrowing.pdf>

Παράρτημα Α

Πείραμα 1^ο: Δυαδική κατηγοριοποίηση εικόνων (α) MRI - Εγκέφαλος (β) XRAY - Θώρακας.

```
clc;
clear all;
close all;
%% Taking an Image
MAX_SIZE_1 = zeros(1,2);
cd ../f1
for i=1:1:100,
    i
    eval(['im = imread(''ima_',int2str(i),''.jpg'',''.jpg'');'])
    %imshow(im)
    [k,l] = size(im);
    if k>MAX_SIZE_1(1,1)
        MAX_SIZE_1(1,1) = k;
    if l>MAX_SIZE_1(1,2)
        MAX_SIZE_1(1,2) = l;
    end
end

end

N1 = max(MAX_SIZE_1(1,1), MAX_SIZE_1(1,2))
TRAIN_DATA=[];
d1=50;
for i=1:1:80,
    i
    eval(['im = imread(''ima_',int2str(i),''.jpg'',''.jpg'');'])
    IM = fft2(im, N1, N1);
    IMs = IM(1:d1:N1, 1:d1:N1);
    ims = ifft2(IMs);
    TRAIN_DATA = [TRAIN_DATA real(ims(:))];
end

end

TEST_DATA=[];
for i=81:1:100,
    i
    eval(['im = imread(''ima_',int2str(i),''.jpg'',''.jpg'');'])
    IM = fft2(im, N1, N1);
    IMs = IM(1:d1:N1, 1:d1:N1);
    ims = ifft2(IMs);
    TEST_DATA = [TEST_DATA real(ims(:))];
end

end

MAX_SIZE_2 = zeros(1,2);
cd ../f2
for i=1:1:100,
    i
    eval(['IM = imread(''ima_',int2str(i),''.jpeg'',''.jpg'');'])
    %imshow(IM)
    [k,l] = size(IM);
    if k>MAX_SIZE_2(1,1)
        MAX_SIZE_2(1,1) = k;
    if l>MAX_SIZE_2(1,2)
        MAX_SIZE_2(1,2) = l;
    end
end

end
```

```
end
N2 = max(MAX_SIZE_2(1,1), MAX_SIZE_2(1,2))
%TRAIN_DATA=[];

d2=floor(d1*N2/N1);
for i=1:1:80,
    i
    eval(['im = imread(''ima_',int2str(i),'.jpeg','jpg');'])
    IM = fft2(im, N2, N2);
    IMs = IM(1:d2:N2, 1:d2:N2);
    IMs = IMs(1:ceil(N1/d1),1:ceil(N1/d1));
    ims = ifft2(IMs);
    TRAIN_DATA = [TRAIN_DATA real(ims(:))];
end

%TEST_DATA=[];

for i=81:1:100,
    i
    eval(['im = imread(''ima_',int2str(i),'.jpeg','jpg');'])
    IM = fft2(im, N2, N2);
    IMs = IM(1:d2:N2, 1:d2:N2);
    IMs = IMs(1:ceil(N1/d1),1:ceil(N1/d1));
    ims = ifft2(IMs);
    TEST_DATA = [TEST_DATA real(ims(:))];
end

T = [ones(1,80) 2*ones(1,80)];
T_test = [ones(1,20) 2*ones(1,20)];

MY_FF = feedforwardnet(4);

MY_FF.divideFcn = '';
[MY_FF, TR] = train(MY_FF, TRAIN_DATA, T);
view(MY_FF)
Y_TRAIN = MY_FF(TRAIN_DATA)
figure(1)
plot(Y_TRAIN, 'ob');
grid;
print -dbitmap ex1_ff4_train.bmp
Y_TEST = MY_FF(TEST_DATA)
figure(2)
plot(Y_TEST, 'or');
grid;
print -dbitmap ex1_ff4_test.bmp
perf = perform(MY_FF, T_test, Y_TEST)
```


Παράρτημα Β

Πείραμα 2^ο: Δυαδική κατηγοριοποίηση εικόνων (α) MRI - Εγκέφαλος (β) XRAY - Θώρακας.

```
clc;
clear all;
close all;
%% Taking an Image
MAX_SIZE_1 = zeros(1,2);
cd '..\f1'
for i=1:1:100,
    eval(['im = imread(''ima_',int2str(i),'.jpg',''jpg');'])
    %imshow(im)
    [k,l] = size(im);
    if k>MAX_SIZE_1(1,1)
        MAX_SIZE_1(1,1) = k;
    if l>MAX_SIZE_1(1,2)
        MAX_SIZE_1(1,2) = l;
    end
end

end

N1 = max(MAX_SIZE_1(1,1), MAX_SIZE_1(1,2))
TRAIN_DATA=[];
d1=60;
LP_2d = zeros(N1,N1);
D1 = 50;
%for q = 1:size(IM_shift,1)
for q = 1:N1
    %for j = 1:size(IM_shift,2)
    for j = 1:N1
        if sqrt((q - N1/2)^2 + (j - N1/2)^2) <= D1
            LP_2d(q,j) = 1;
        end
    end
end

end

for i=1:1:80,
    i
    eval(['imrgb = imread(''ima_',int2str(i),'.jpg',''jpg');'])
    %IM = fft2(im, N1, N1);
    %IMs = IM(1:d1:N1, 1:d1:N1);
    %ims = ifft2(IMs);
    [k,l,m]=size(imrgb);

    if (m>1)
        im = rgb2gray(imrgb);
    else
        im=imrgb;
    end
    [M1, M2] = size(im);
    S=zeros(M1, M2);
    im=im2double(im);
    %im_h = histeq(im);
    J=zeros(M1, M2);
    R = 1;

    % MM = max(M1,M2);
    MM=N1;
```

```
for i=1:R,
    %seeds = [rand(R,1) rand(R,1)];
    seeds = [3/8, 3/5];
end

for i=1:R,

S(round(M1*seeds(i,1)):round(M1*seeds(i,1))+1,round(M2*seeds(i,2)):ro
und(M2*seeds(i,2))+1) = 1;
%J(:, :, i) =
regiongrowing(I,round(590*rand2(i))+1,round(440*rand1(i))+1,0.5);
%figure(3)
%imshow(S)

%J = J + ((1/R)*i)*regiongrowing(I_h, round(N1*seeds(i,1)),
round(N2*seeds(i,2)),0.1);

J =regiongrowing(im, round(M1*seeds(i,1)), round(M2*seeds(i,2)),0.1);
%figure(4)
%imshow(J)

    IM = fft2(J, MM, MM);
    %IMs = IM(1:d:N, 1:d:N);
    IM_shift = fftshift(IM);
    IM_lp = IM_shift.*LP_2d;
    IM_lp = ifftshift(IM_lp);
    ims = ifft2(IM_lp);
    ims=ims(1:d1:N1,1:d1:N1);
    ims_real = real(ims);
    % figure(5)
    % imshow(uint8(ims_real), [0,1])

end

    TRAIN_DATA = [TRAIN_DATA uint8(ims_real(:))];
end

TEST_DATA=[];
for i=81:1:100,
    i
    eval(['imrgb = imread(''ima_',int2str(i),'.jpg',''jpg');'])
    %IM = fft2(im, N1, N1);
    %IMs = IM(1:d1:N1, 1:d1:N1);
    %ims = ifft2(IMs);
    [k,l,m]=size(imrgb);
    if (m>1)
        im = rgb2gray(imrgb);
    else
        im=imrgb;
    end
    [M1, M2] = size(im);
    S=zeros(M1, M2);
    im=im2double(im);
    %im_h = histeq(im);
    J=zeros(M1, M2);
    R = 1;

    % MM = max(M1,M2);
    MM=N1;
end
```

```
for i=1:R,
    %seeds = [rand(R,1) rand(R,1)];
    seeds = [3/8, 3/5];
end

for i=1:R,

S(round(M1*seeds(i,1)):round(M1*seeds(i,1))+1,round(M2*seeds(i,2)):ro
und(M2*seeds(i,2))+1) = 1;
%J(:, :, i) =
regiongrowing(I,round(590*rnd2(i))+1,round(440*rnd1(i))+1,0.5);
%figure(3)
%imshow(S)

%J = J + ((1/R)*i)*regiongrowing(I_h, round(N1*seeds(i,1)),
round(N2*seeds(i,2)),0.1);

J =regiongrowing(im, round(M1*seeds(i,1)), round(M2*seeds(i,2)),0.1);
%figure(4)
%imshow(J)

    IM = fft2(J, MM, MM);
    %IMs = IM(1:d:N, 1:d:N);
    IM_shift = fftshift(IM);
    IM_lp = IM_shift.*LP_2d;
    IM_lp = ifftshift(IM_lp);
    ims = ifft2(IM_lp);
    ims=ims(1:d1:N1,1:d1:N1);
    ims_real = real(ims);
    % figure(5)
    % imshow(uint8(ims_real), [0,1])

end

    TEST_DATA = [TEST_DATA uint8(ims_real(:))];
end

MAX_SIZE_2 = zeros(1,2);
cd ../f2
for i=1:1:100,
    eval(['IM = imread(''ima_',int2str(i),''.jpeg'', ''jpg'');'])
    %imshow(IM)
    [k,l] = size(IM);
    if k>MAX_SIZE_2(1,1)
        MAX_SIZE_2(1,1) = k;
    if l>MAX_SIZE_2(1,2)
        MAX_SIZE_2(1,2) = l;
    end
end

end

N2 = max(MAX_SIZE_2(1,1), MAX_SIZE_2(1,2))
%TRAIN_DATA=[];

d2=floor(d1*N2/N1);
LP_2d = zeros(N2,N2);
D2 = floor(D1*(N2/N1));
    %for q = 1:size(IM_shift,1)
    for q = 1:N2
```

```
%for j = 1:size(IM_shift,2)
for j = 1:N2
    if sqrt((q - N2/2)^2 + (j - N2/2)^2) <= D2
        LP_2d(q,j) = 1;
    end
end

end
end
for i=1:1:80,
    i
    eval(['imrgb = imread(''ima_',int2str(i),'.jpeg',''jpg'');'])
    %IM = fft2(im, N2, N2);
    %IMs = IM(1:d2:N2, 1:d2:N2);
    %IMs = IMs(1:ceil(N1/d1),1:ceil(N1/d1));
    %ims = ifft2(IMs);

[k,l,m]=size(imrgb);

    if (m>1)
        im = rgb2gray(imrgb);
    else
        im=imrgb;
    end
    [M1, M2] = size(im);
    S=zeros(M1, M2);
    im=im2double(im);
    %im_h = histeq(im);
    J=zeros(M1, M2);
    R = 1;

    % MM = max(M1,M2);
    MM=N2;
    for i=1:R,
        %seeds = [rand(R,1) rand(R,1)];
        seeds = [3/8, 3/5];
    end

    for i=1:R,

S(round(M1*seeds(i,1)):round(M1*seeds(i,1))+1,round(M2*seeds(i,2)):ro
und(M2*seeds(i,2))+1) = 1;
%J(:, :, i) =
regiongrowing(I,round(590*rnd2(i))+1,round(440*rnd1(i))+1,0.5);
%figure(3)
%imshow(S)

%J = J + ((1/R)*i)*regiongrowing(I_h, round(N1*seeds(i,1)),
round(N2*seeds(i,2)),0.1);

J =regiongrowing(im, round(M1*seeds(i,1)), round(M2*seeds(i,2)),0.1);
%figure(4)
%imshow(J)

    IM = fft2(J, MM, MM);
    %IMs = IM(1:d:N, 1:d:N);
    IM_shift = fftshift(IM);
    IM_lp = IM_shift.*LP_2d;
    IM_lp = ifftshift(IM_lp);
    ims = ifft2(IM_lp);
```

```
    ims=ims(1:d2:N2,1:d2:N2);
    ims_real = real(ims);
    % figure(5)
    % imshow(uint8(ims_real),[0,1])

end

    TRAIN_DATA = [TRAIN_DATA uint8(ims_real(:))];
end

%TEST_DATA=[];

for i=81:1:100,
    i
    eval(['imrgb = imread(''ima_',int2str(i),'.jpeg',''jpg'');'])
    %IM = fft2(im, N2, N2);
    %IMs = IM(1:d2:N2, 1:d2:N2);
    %IMs = IMs(1:ceil(N1/d1),1:ceil(N1/d1));
    %ims = ifft2(IMs);

    [k,l,m]=size(imrgb);

    if (m>1)
        im = rgb2gray(imrgb);
    else
        im=imrgb;
    end
    [M1, M2] = size(im);
    S=zeros(M1, M2);
    im=im2double(im);
    %im_h = histeq(im);
    J=zeros(M1, M2);
    R = 1;

    % MM = max(M1,M2);
    MM=N2;
for i=1:R,
    %seeds = [rand(R,1) rand(R,1)];
    seeds = [3/8, 3/5];
end

for i=1:R,

S(round(M1*seeds(i,1)):round(M1*seeds(i,1))+1,round(M2*seeds(i,2)):ro
und(M2*seeds(i,2))+1) = 1;
%J(:, :, i) =
regiongrowing(I,round(590*rand2(i))+1,round(440*rand1(i))+1,0.5);
%figure(3)
%imshow(S)

%J = J + ((1/R)*i)*regiongrowing(I_h, round(N1*seeds(i,1)),
round(N2*seeds(i,2)),0.1);

J =regiongrowing(im, round(M1*seeds(i,1)), round(M2*seeds(i,2)),0.1);
%figure(4)
%imshow(J)

    IM = fft2(J, MM, MM);
```

```
%IMs = IM(1:d:N, 1:d:N);
IM_shift = fftshift(IM);
IM_lp = IM_shift.*LP_2d;
IM_lp = ifftshift(IM_lp);
ims = ifft2(IM_lp);
ims=ims(1:d2:N2,1:d2:N2);
ims_real = real(ims);
% figure(5)
% imshow(uint8(ims_real), [0,1])

end

TEST_DATA = [TEST_DATA uint8(ims_real(:))];
end

T = [ones(1,80) 2*ones(1,80)];
T_test = [ones(1,20) 2*ones(1,20)];

MY_FF = feedforwardnet(4);

[MY_FF, TR] = train(MY_FF, TRAIN_DATA, T);
view(MY_FF)
Y_TRAIN = MY_FF(TRAIN_DATA)
perf = perform(MY_FF, T, Y_TRAIN)
figure(1)
plot(Y_TRAIN, 'ob');
grid;
print -dbitmap ex2_ff4_train.bmp
Y_TEST = MY_FF(TEST_DATA)
view(MY_FF)
perf = perform(MY_FF, T_test, Y_TEST)
figure(2)
plot(Y_TEST, 'or');
grid;
print -dbitmap ex2_ff4_test.bmp
```

Παράρτημα Γ

Πείραμα 3^ο: Δυαδική κατηγοριοποίηση εικόνων (α) XRAY - Θώρακας υγής (β) XRAY - Θώρακας με πνευμονία.

```
clc;
clear all;
close all;
%% Taking an Image
MAX_SIZE_1 = zeros(1,2);
cd ../f2
for i=1:1:100,
    i
    eval(['im = imread('ima_',int2str(i),'.jpeg','jpg');'])
    imshow(im)
    [k,l] = size(im);
    if k>MAX_SIZE_1(1,1)
        MAX_SIZE_1(1,1) = k;
    if l>MAX_SIZE_1(1,2)
        MAX_SIZE_1(1,2) = l;
    end
end

end

N1 = max(MAX_SIZE_1(1,1), MAX_SIZE_1(1,2))
TRAIN_DATA=[];
d1=80;
for i=1:1:80,
    i
    eval(['im = imread('ima_',int2str(i),'.jpeg','jpg');'])
    IM = fft2(im, N1, N1);
    IMs = IM(1:d1:N1, 1:d1:N1);
    ims = ifft2(IMs);
    TRAIN_DATA = [TRAIN_DATA real(ims(:))];
end

end

TEST_DATA=[];
for i=81:1:100,
    i
    eval(['im = imread('ima_',int2str(i),'.jpeg','jpg');'])
    IM = fft2(im, N1, N1);
    IMs = IM(1:d1:N1, 1:d1:N1);
    ims = ifft2(IMs);
    TEST_DATA = [TEST_DATA real(ims(:))];
end

end

MAX_SIZE_2 = zeros(1,2);
cd ../f3
for i=1:1:100,
    i
    eval(['IM = imread('ima_',int2str(i),'.jpeg','jpg');'])
    imshow(IM)
    [k,l] = size(IM);
    if k>MAX_SIZE_2(1,1)
        MAX_SIZE_2(1,1) = k;
    if l>MAX_SIZE_2(1,2)
        MAX_SIZE_2(1,2) = l;
    end
end
end
```

```
end
N2 = max(MAX_SIZE_2(1,1), MAX_SIZE_2(1,2))
%TRAIN_DATA=[];

d2=floor(d1*N2/N1);
for i=1:1:80,
    i
    eval(['im = imread(''ima_',int2str(i),'.jpeg','jpg');'])
    IM = fft2(im, N2, N2);
    IMs = IM(1:d2:N2, 1:d2:N2);
    IMs = IMs(1:ceil(N1/d1),1:ceil(N1/d1));
    ims = ifft2(IMs);
    TRAIN_DATA = [TRAIN_DATA real(ims(:))];
end

%TEST_DATA=[];

for i=81:1:100,
    i
    eval(['im = imread(''ima_',int2str(i),'.jpeg','jpg');'])
    IM = fft2(im, N2, N2);
    IMs = IM(1:d2:N2, 1:d2:N2);
    IMs = IMs(1:ceil(N1/d1),1:ceil(N1/d1));
    ims = ifft2(IMs);
    TEST_DATA = [TEST_DATA real(ims(:))];
end

T = [ones(1,80) 2*ones(1,80)];
T_test = [ones(1,20) 2*ones(1,20)];

MY_FF = feedforwardnet(6);

MY_FF.divideFcn = '';
[MY_FF, TR] = train(MY_FF, TRAIN_DATA, T);
view(MY_FF)
Y_TRAIN = MY_FF(TRAIN_DATA)
figure(1)
plot(Y_TRAIN, 'ob');
grid;
print -dbitmap ex3_ff6_train.bmp
Y_TEST = MY_FF(TEST_DATA)
figure(2)
plot(Y_TEST, 'or');
grid;
print -dbitmap ex3_ff6_test.bmp
perf = perform(MY_FF, T_test, Y_TEST)
```


Παράρτημα Δ

Πείραμα 4^ο: Δυαδική κατηγοριοποίηση εικόνων (α) XRAY - Θώρακας με πνευμονία
(β) XRAY - Θώρακας υγιής.

```
clc;
clear all;
close all;
%% Taking an Image
MAX_SIZE_1 = zeros(1,2);
cd ../f3
for i=1:1:100,
    i
    eval(['im = imread(''ima_',int2str(i),''.jpeg'', ''jpg'');'])
    %imshow(im)
    [k,l] = size(im);
    if k>MAX_SIZE_1(1,1)
        MAX_SIZE_1(1,1) = k;
    if l>MAX_SIZE_1(1,2)
        MAX_SIZE_1(1,2) = l;
    end
end

end

N1 = max(MAX_SIZE_1(1,1), MAX_SIZE_1(1,2))
TRAIN_DATA=[];
d1=30;
LP_2d = zeros(N1,N1);
D1 = 50;
%for q = 1:size(IM_shift,1)
for q = 1:N1
    %for j = 1:size(IM_shift,2)
    for j = 1:N1
        if sqrt((q - N1/2)^2 + (j - N1/2)^2) <= D1
            LP_2d(q,j) = 1;
        end
    end
end

end

for i=1:1:80,
    i
    eval(['imrgb = imread(''ima_',int2str(i),''.jpeg'', ''jpg'');'])
    %IM = fft2(im, N1, N1);
    %IMs = IM(1:d1:N1, 1:d1:N1);
    %ims = ifft2(IMs);
    [k,l,m]=size(imrgb);

    if (m>1)
        im = rgb2gray(imrgb);
    else
        im=imrgb;
    end
    [M1, M2] = size(im);
    S=zeros(M1, M2);
    im=im2double(im);
    %im_h = histeq(im);
    J=zeros(M1, M2);
    R = 1;

    % MM = max(M1,M2);
```

```
MM=N1;
for i=1:R,
    %seeds = [rand(R,1) rand(R,1)];
    seeds = [3/8, 3/5];
end

for i=1:R,

S(round(M1*seeds(i,1)):round(M1*seeds(i,1))+1,round(M2*seeds(i,2)):ro
und(M2*seeds(i,2))+1) = 1;
%J(:, :, i) =
regiongrowing(I,round(590*rnd2(i))+1,round(440*rnd1(i))+1,0.5);
%figure(3)
%imshow(S)

%J = J + ((1/R)*i)*regiongrowing(I_h, round(N1*seeds(i,1)),
round(N2*seeds(i,2)),0.1);

J =regiongrowing(im, round(M1*seeds(i,1)), round(M2*seeds(i,2)),0.1);
%figure(4)
%imshow(J)

    IM = fft2(J, MM, MM);
    %IMs = IM(1:d:N, 1:d:N);
    IM_shift = fftshift(IM);
    IM_lp = IM_shift.*LP_2d;
    IM_lp = ifftshift(IM_lp);
    ims = ifft2(IM_lp);
    ims=ims(1:d1:d1*floor(N1/d1),1:d1:d1*floor(N1/d1));
    ims_real = real(ims);
    % figure(5)
    % imshow(uint8(ims_real), [0,1])

end

    TRAIN_DATA = [TRAIN_DATA ims_real(:)];
end

TEST_DATA=[];
for i=81:1:100,
    i
    eval(['imrgb = imread(''ima_',int2str(i),'.jpeg',''jpg');'])
    %IM = fft2(im, N1, N1);
    %IMs = IM(1:d1:N1, 1:d1:N1);
    %ims = ifft2(IMs);
    [k,l,m]=size(imrgb);
    if (m>1)
        im = rgb2gray(imrgb);
    else
        im=imrgb;
    end
    [M1, M2] = size(im);
    S=zeros(M1, M2);
    im=im2double(im);
    %im_h = histeq(im);
    J=zeros(M1, M2);
    R = 1;

    % MM = max(M1,M2);
```

```
MM=N1;
for i=1:R,
    %seeds = [rand(R,1) rand(R,1)];
    seeds = [3/8, 3/5];
end

for i=1:R,

S(round(M1*seeds(i,1)):round(M1*seeds(i,1))+1,round(M2*seeds(i,2)):ro
und(M2*seeds(i,2))+1) = 1;
%J(:, :, i) =
regiongrowing(I,round(590*rnd2(i))+1,round(440*rnd1(i))+1,0.5);
%figure(3)
%imshow(S)

%J = J + ((1/R)*i)*regiongrowing(I_h, round(N1*seeds(i,1)),
round(N2*seeds(i,2)),0.1);

J =regiongrowing(im, round(M1*seeds(i,1)), round(M2*seeds(i,2)),0.1);
%figure(4)
%imshow(J)

    IM = fft2(J, MM, MM);
    %IMs = IM(1:d:N, 1:d:N);
    IM_shift = fftshift(IM);
    IM_lp = IM_shift.*LP_2d;
    IM_lp = ifftshift(IM_lp);
    ims = ifft2(IM_lp);
    ims=ims(1:d1:d1*floor(N1/d1),1:d1:d1*floor(N1/d1));
    ims_real = real(ims);
    % figure(5)
    % imshow(uint8(ims_real), [0,1])

end

    TEST_DATA = [TEST_DATA ims_real(:)];
end

MAX_SIZE_2 = zeros(1,2);
cd '..\f2'
for i=1:1:100,
    i
    eval(['IM = imread(''ima_',int2str(i),'.jpeg',''jpg');'])
    %imshow(IM)
    [k,l] = size(IM);
    if k>MAX_SIZE_2(1,1)
        MAX_SIZE_2(1,1) = k;
    if l>MAX_SIZE_2(1,2)
        MAX_SIZE_2(1,2) = l;
    end
end
end

end
N2 = max(MAX_SIZE_2(1,1), MAX_SIZE_2(1,2))
%TRAIN_DATA=[];

d2=ceil(d1*N2/N1);
LP_2d = zeros(N2,N2);
D2 = floor(D1*(N2/N1));
```

```
%for q = 1:size(IM_shift,1)
for q = 1:N2
    %for j = 1:size(IM_shift,2)
    for j = 1:N2
        if sqrt((q - N2/2)^2 + (j - N2/2)^2) <= D2
            LP_2d(q,j) = 1;
        end
    end
end
end
for i=1:1:80,
    i
    eval(['imrgb = imread(''ima_',int2str(i),'.jpeg',''jpg'');'])
    %IM = fft2(im, N2, N2);
    %IMs = IM(1:d2:N2, 1:d2:N2);
    %IMs = IMs(1:ceil(N1/d1),1:ceil(N1/d1));
    %ims = ifft2(IMs);

    [k,l,m]=size(imrgb);

    if (m>1)
        im = rgb2gray(imrgb);
    else
        im=imrgb;
    end
    [M1, M2] = size(im);
    S=zeros(M1, M2);
    im=im2double(im);
    %im_h = histeq(im);
    J=zeros(M1, M2);
    R = 1;

    % MM = max(M1,M2);
    MM=N2;
    for i=1:R,
        %seeds = [rand(R,1) rand(R,1)];
        seeds = [3/8, 3/5];
    end

    for i=1:R,

        S(round(M1*seeds(i,1)):round(M1*seeds(i,1))+1,round(M2*seeds(i,2)):ro
und(M2*seeds(i,2))+1) = 1;
        %J(:, :, i) =
        regiongrowing(I, round(590*rnd2(i))+1, round(440*rnd1(i))+1, 0.5);
        %figure(3)
        %imshow(S)

        %J = J + ((1/R)*i)*regiongrowing(I_h, round(N1*seeds(i,1)),
round(N2*seeds(i,2)), 0.1);

        J =regiongrowing(im, round(M1*seeds(i,1)), round(M2*seeds(i,2)), 0.1);
        %figure(4)
        %imshow(J)

        IM = fft2(J, MM, MM);
        %IMs = IM(1:d:N, 1:d:N);
        IM_shift = fftshift(IM);
        IM_lp = IM_shift.*LP_2d;
```

```
    IM_lp = ifftshift(IM_lp);
    ims = ifft2(IM_lp);
    ims=ims(1:d2:d2*floor(N2/d2),1:d2:d2*floor(N2/d2));
    %ims=ims(1:floor(N1/d1),1:floor(N1/d1));
    ims_real = real(ims);
    % figure(5)
    % imshow(uint8(ims_real),[0,1])

end

    TRAIN_DATA = [TRAIN_DATA ims_real(:)];
end

%TEST_DATA=[];

for i=81:1:100,
    i
    eval(['imrgb = imread(''ima_',int2str(i),'.jpeg',''jpg'');'])
    %IM = fft2(im, N2, N2);
    %IMs = IM(1:d2:N2, 1:d2:N2);
    %IMs = IMs(1:ceil(N1/d1),1:ceil(N1/d1));
    %ims = ifft2(IMs);

    [k,l,m]=size(imrgb);

    if (m>1)
        im = rgb2gray(imrgb);
    else
        im=imrgb;
    end
    [M1, M2] = size(im);
    S=zeros(M1, M2);
    im=im2double(im);
    %im_h = histeq(im);
    J=zeros(M1, M2);
    R = 1;

    % MM = max(M1,M2);
    MM=N2;
    for i=1:R,
        %seeds = [rand(R,1) rand(R,1)];
        seeds = [3/8, 3/5];
    end

    for i=1:R,

        S(round(M1*seeds(i,1)):round(M1*seeds(i,1))+1,round(M2*seeds(i,2)):ro
und(M2*seeds(i,2))+1) = 1;
        %J(:, :, i) =
        regiongrowing(I, round(590*rnd2(i))+1, round(440*rnd1(i))+1, 0.5);
        %figure(3)
        %imshow(S)

        %J = J + ((1/R)*i)*regiongrowing(I_h, round(N1*seeds(i,1)),
round(N2*seeds(i,2)), 0.1);

        J =regiongrowing(im, round(M1*seeds(i,1)), round(M2*seeds(i,2)), 0.1);
        %figure(4)
    end
end
```

```
%imshow(J)

IM = fft2(J, MM, MM);
%IMs = IM(1:d:N, 1:d:N);
IM_shift = fftshift(IM);
IM_lp = IM_shift.*LP_2d;
IM_lp = ifftshift(IM_lp);
ims = ifft2(IM_lp);
ims=ims(1:d2:d2*floor(N2/d2), 1:d2:d2*floor(N2/d2));
%ims=ims(1:floor(N1/d1), 1:floor(N1/d1));
ims_real = real(ims);
% figure(5)
% imshow(uint8(ims_real), [0,1])

end

TEST_DATA = [TEST_DATA ims_real(:)];
end

T = [ones(1,80) 2*ones(1,80)];
T_test = [ones(1,20) 2*ones(1,20)];

MY_FF = feedforwardnet(6);
MY_FF.divideFcn='';
[MY_FF, TR] = train(MY_FF, TRAIN_DATA, T);
view(MY_FF)
Y_TRAIN = MY_FF(TRAIN_DATA)
perf = perform(MY_FF, T, Y_TRAIN)
figure(1)
plot(Y_TRAIN, 'ob');
grid;
print -dbitmap ex4_ff6_train.bmp

Y_TEST = MY_FF(TEST_DATA)
view(MY_FF)
perf = perform(MY_FF, T_test, Y_TEST)
figure(2)
plot(Y_TEST, 'or');
grid;
print -dbitmap ex4_ff6_test.bmp
```
