

**Εκπαίδευση Τεχνητών Νευρωνικών Δικτύων ανά
Πρότυπο Εισόδου**

ΑΜΑΛΙΑ ΠΑΠΑΝΙΚΟΛΑΟΥ

Υπεύθυνος

Πλαγιανάκος Βασίλειος

Επίκουρος Καθηγητής

Λαμία, 2008

Ευχαριστίες

Θα ήθελα καταρχήν να ευχαριστήσω τον καθηγητή μου Δρ. Βασίλειο Πλαγιανάκο, Επίκουρο Καθηγητή στο Τμήμα Πληροφορικής με Εφαρμογές στη Βιοϊατρική του Πανεπιστημίου Στερεάς Ελλάδας, που είχε την επίβλεψη αυτής της διπλωματικής και καθόλη τη διάρκεια της εργασίας με τις συμβουλές του και τις παρατηρήσεις του βοήθησε καθοριστικά στην εξέλιξη και τη διαμόρφωση της. Επίσης θα ήθελα να ευχαριστήσω τους γονείς μου για την καθοδήγηση και την ηθική συμπαράσταση που μου προσέφεραν όλα αυτά τα χρόνια.

Περίληψη

Το αντικείμενο αυτής της Διπλωματικής Εργασίας είναι η μελέτη μεθόδων εκπαίδευσης Τεχνητών Νευρωνικών Δικτύων ανά πρότυπο εισόδου. Κατά τη διάρκεια της συγκεκριμένης εργασίας έχουν μελετηθεί οι βασικές αρχιτεκτονικές των Τεχνητών Νευρωνικών Δικτύων και οι διάφοροι μέθοδοι εκπαίδευσης αυτών. Έχει δοθεί έμφαση στη μαθηματική ανάλυση των δικτύων, τις τεχνικές εκπαίδευσης και στις εφαρμογές αυτών σε πραγματικά προβλήματα.

Ακόμη έχει γίνει μελέτη της απόδοσης ενός δικτύου. Μελετούνται οι διάφορες τεχνικές προκειμένου ένα δίκτυο να βελτιστοποιήσει την απόδοσή του με τη χρήση αλγορίθμων οι οποίοι εντοπίζουν το βέλτιστο σύνολο βαρών που επιτρέπει στο δίκτυο να μάθει το σύνολο εκπαίδευσης και έτσι βελτιώνουν την αποδοτικότητα της διαδικασίας εκπαίδευσης.

Σε προγραμματιστικό επίπεδο έχει χρησιμοποιηθεί το λογισμικό Matlab για τη δημιουργία ενός αλγορίθμου εκπαίδευσης. Ο αλγόριθμος αυτός πραγματοποιεί εκπαίδευση ενός Τεχνητού Νευρωνικού Δικτύου με τη μέθοδο της όπισθεν διάδοσης σφάλματος (Backpropagation) και δίνει χρήσιμα συμπεράσματα για το ποσοστό επιτυχίας στην εκπαίδευση συγκεκριμένων προβλημάτων καθώς και το ποσοστό του μέσου τετραγωνικού σφάλματος. Επίσης έχουν εξαχθεί συγκριτικά αποτελέσματα όταν χρησιμοποιούμε εκπαίδευση ανά πρότυπο εισόδου και εκπαίδευση ανά ομάδα προτύπων εισόδου.

Στο Κεφάλαιο 1 παρέχονται κάποιες βασικές έννοιες που αφορούν τη δομή και λειτουργία των βιολογικών νευρώνων και τον τρόπο με τον οποίο η μελέτη αυτή ενέπνευσε τη δημιουργία υπολογιστικών δομών που η οργάνωσή τους είναι βασισμένη στο βιολογικό μοντέλο.

Στο Κεφάλαιο 2 περιγράφεται η δομή των Τεχνητών Νευρωνικών Δικτύων και παρέχονται οι βασικοί ορισμοί. Επίσης αναλύουμε τους τρόπους εκπαίδευσης ενός Νευρωνικού Δικτύου και περιγράφουμε τις ιδιότητες αυτών των Δικτύων. Τέλος, δίνεται μια σύντομη ιστορική αναδρομή και παρουσιάζονται οι διαφορές των Τεχνητών Νευρωνικών Δικτύων από τα παραδοσιακά υπολογιστικά συστήματα και τα Έμπειρα Συστήματα.

Στα επόμενα δύο κεφάλαια παρουσιάζεται η δομή ενός απλού νευρωνικού δικτύου που ονομάζεται Perceptron και ο κανόνας εκπαίδευσης του δικτύου αυτού, καθώς και ο κανόνας εκπαίδευσης του Hebb.

Στο Κεφάλαιο 5 παρουσιάζονται τρεις βασικές μέθοδοι βελτιστοποίησης και δίνονται κάποια βασικά παραδείγματα για την κατανόηση αυτών των μεθόδων.

Στο Κεφάλαιο 6 αναλύουμε τη δομή των πολυστρωματικών δικτύων και περιγράφεται ο αλγόριθμος της όπισθεν διάδοσης σφάλματος. Στο Κεφάλαιο 7 παρουσιάζουμε το δίκτυο Adaline, ενώ στο επόμενο κεφάλαιο παρουσιάζονται κάποιες παραλλαγές του αλγορίθμου

της όπισθεν διάδοση σφάλματος.

Τέλος στο Κεφάλαιο 9 εφαρμόζουμε τις διάφορες μεθόδους εκπαίδευσης σε πραγματικά προβλήματα και συγκρίνουμε την απόδοση του δικτύου όταν χρησιμοποιείται εκπαίδευση ανά ομάδα προτύπων εισόδου και εκπαίδευση ανά πρότυπο εισόδου.

Για την εκπόνηση αυτής της διπλωματικής χρησιμοποιήθηκε το λογισμικό Matlab για την ανάπτυξη του πρακτικού μέρους της εργασίας. Η συγγραφή πραγματοποιήθηκε σε TEX μορφή με τη χρήση του προγράμματος Kile, ενώ η δημιουργία των σχημάτων έγινε με τη βοήθεια του προγράμματος OpenOffice Drawing και του προγράμματος GIMP.

Abstract

The subject of this dissertation is the study of artificial neural network architectures and learning rules. Emphasis is placed in the mathematical analysis of these networks, on methods of training them and on their applications in practical engineering problems.

In addition, we introduce the basic concepts of a class of learning called the performance learning, in which a network is trained to optimize its performance.

We have also developed a training algorithm using Matlab software which trains feed-forward neural networks using the backpropagation algorithm. We give useful information about the performance and the accuracy of the network and give the basic differences between on-line and batch mode learning.

Chapter 1 covers the basic concepts of the biological neuron model and how they inspired the development of artificial neurons.

Chapter 2 describes the basic neural network architectures, the different methods of training a neural network and gives a brief historical background. In the next two chapters we present the perceptron learning rule and the Hebb rule.

In Chapter 5 we present three basic optimization methods and we give some examples for better understanding.

Chapter 6 describes the architecture of multilayer perceptrons and presents a learning rule called backpropagation. Next we present the Adaline network and different variations of the backpropagation algorithm.

Finally, in Chapter 9 we perform different learning methods in practical problems and we present the results.

Περιεχόμενα

Ευχαριστίες	3
Περίληψη	5
Abstract	7
Περιεχόμενα	10
Κατάλογος Σχημάτων	12
Κατάλογος Πινάκων	13
I Θεωρητικό Μέρος	15
1 Η πορεία προς τα Τεχνητά Νευρωνικά Δίκτυα	17
1.1 Εισαγωγή	17
1.2 Στόχος της Εργασίας	19
1.3 Προσομοίωση του βιολογικού μοντέλου	20
1.3.1 Σύγκριση βιολογικών και των Τεχνητών Νευρωνικών δικτύων.	22
2 Τεχνητά Νευρωνικά Δίκτυα	27
2.1 Στοιχεία ενός Τεχνητού Νευρώνα	27
2.2 Συναρτήσεις ενεργοποίησης	28
2.3 Τοπολογία νευρωνικών δικτύων	29
2.4 Εκπαίδευση Τεχνητών Νευρωνικών Δικτύων	30
2.5 Ιδιότητες των Νευρωνικών Δικτύων	33
2.6 Ιστορική Αναδρομή	34
2.7 Διαφορές των Τεχνητών Νευρωνικών Δικτύων από άλλα συστήματα Τεχνητής Νοημοσύνης	36
3 Το δίκτυο Perceptron	41
3.1 Κανόνας εκπαίδευσης Perceptron	43
3.1.1 Παράδειγμα	45
3.2 Τροποποίηση του κανόνα Perceptron	46

4 Ο κανόνας εκπαίδευσης του Hebb	49
5 Μέθοδοι Βελτιστοποίησης	55
5.1 Το διάνυσμα των μερικών παραγώγων και ο Εσσιανός πίνακας	56
5.2 Βέλτιστο σημείο	57
5.3 Απαραίτητες συνθήκες βελτιστοποίησης	58
5.4 Ιδιότητες τετραγωνικών συναρτήσεων	60
5.5 Αλγόριθμοι βελτιστοποίησης	63
5.5.1 Η μέθοδος της πιο απότομης καθόδου	63
5.5.2 Η μέθοδος του Newton	67
5.5.3 Η μέθοδος των συζυγών κλίσεων	69
6 Το δίκτυο ADALINE	71
6.1 Αλγόριθμος LMS	72
7 Πολυστρωματικά Τεχνητά Νευρωνικά Δίκτυα	75
7.1 Το πρόβλημα της αποκλειστικής διάζευξης	76
7.2 Η μέθοδος της όπισθεν διάδοσης σφάλματος	78
7.2.1 Ανάστροφη μετάδοση των σημάτων σφάλματος διαμέσου του δικτύου	81
7.2.2 Ο αλγόριθμος BackPropagation	82
7.3 Τρόποι εκπαίδευσης ενός πολυστρωματικού δικτύου	83
7.3.1 Εκπαίδευση ανά ομάδα προτύπων εισόδου	83
7.3.2 Εκπαίδευση ανά πρότυπο εισόδου	84
7.4 Σχεδίαση νευρωνικών δικτύων	84
8 Παραλλαγές του αλγορίθμου της όπισθεν διάδοσης σφάλματος	89
8.1 Εκπαίδευση με τη μέθοδο των συζυγών κλίσεων	89
8.2 Ο αλγόριθμος Levenberg-Marquardt	93
II Πρακτικό Μέρος	97
9 Πρακτική εφαρμογή των Τεχνητών Νευρωνικών Δικτύων	99
9.1 Παραδείγματα και Αποτελέσματα	99
9.1.1 Προσέγγιση μιας συνεχούς συνάρτησης	99
9.1.2 Το πρόβλημα της περιττής ισοτιμίας	103
9.1.3 Αναγνώριση προτύπων	104
9.1.4 Το πρόβλημα Sonar	104
9.2 Συμπεράσματα	106
III ΠΑΡΑΡΤΗΜΑ	107
Βιβλιογραφία	118

Κατάλογος Σχημάτων

1.1 Ένα χειρόγραφο κεφαλαίο Α.	18
1.2 Σχηματική απεικόνιση νευρωνικού κυττάρου.	21
1.3 Σχηματική απεικόνιση της σύναψης μεταξύ 2 νευρικών κυττάρων.	22
1.4 Μηχανισμός εκπόλωσης μιας νευρικής ίνας.	23
1.5 Μοντέλο βιολογικού νευρώνα.	24
1.6 Μοντέλο τεχνητού νευρώνα.	25
2.1 Τεχνητός Νευρώνας με μία είσοδο.	28
2.2 Αναπαράσταση των βασικών συναρτήσεων	29
2.3 Πλήρως συνδεδεμένο ΤΝΔ πρόσθιας τροφοδότησης 3-4-2.	30
3.1 Δίκτυο Perceptron	41
3.2 Ένα απλό δίκτυο Perceptron.	42
3.3 Διαχωριστική γραμμή για ένα δίκτυο Perceptron δύο εισόδων.	43
3.4 Γραφική αναπαράσταση εισόδων.	45
3.5 Δίκτυο Perceptron του προβλήματος.	45
3.6 Γραμμική διαχωριστικότητα των δεδομένων του προβλήματος.	45
3.7 Αποτέλεσμα μετά από τυχαία αρχικοποίηση των βαρών του δικτύου.	46
3.8 Αναπαράσταση των δεδομένων μετά την εκπαίδευση του δικτύου.	46
4.1 Γραμμικός Συσχετιστής	49
4.2 Δίκτυο Adaline του παραδείγματος 4.1	52
4.3 Αναπαράσταση ψηφίων	53
4.4 Δίκτυο αυτοσυσχέτισης για αναγνώριση προτύπων.	54
4.5 Ανάκτηση των εισόδων για 50% εσφαλμένα πρότυπα.	54
4.6 Ανάκτηση των εισόδων για 67% εσφαλμένα πρότυπα.	54
4.7 Ανάκτηση προτύπων στα οποία έχει εισαχθεί θόρυβος	54
5.1 Συνάρτηση σινημιτόνου και η προσέγγιση της με σειρές Taylor	56
5.2 Παράδειγμα τοπικού και ολικού ελάχιστου	58
5.3 Παράδειγμα τοπικού και ολικού ελάχιστου	58
5.4 Γραμμικό Νευρωνικό Δίκτυο	61
5.5 Τρισδιάστατη αναπαράσταση και διάγραμμα ισοϋψών της συνάρτησης $F(x)$	62

5.6	Η μέθοδος της πιο απότομης καθόδου	64
5.7	Η μέθοδος της πιο απότομης καθόδου για ρυθμό εκπαίδευσης $a = 0.01$. . .	66
5.8	Η μέθοδος της πιο απότομης καθόδου για ρυθμό εκπαίδευσης $a = 0.035$. . .	66
5.9	Η μέθοδος της πιο απότομης καθόδου για $a = 0.039$	67
5.10	Η μέθοδος της πιο απότομης καθόδου για $a = 0.041$	67
5.11	Η μέθοδος του Newton για ένα αρχικό σημείο $\mathbf{x}_0 = [1.5 \ 0]^T$	68
5.12	Η μέθοδος του Newton για ένα αρχικό σημείο $\mathbf{x}_0 = [-1.5 \ 0]^T$	69
5.13	Η μέθοδος του Newton για ένα αρχικό σημείο $\mathbf{x}_0 = [0.75 \ 0.75]^T$	69
6.1	Δίκτυο ADALINE.	71
6.2	Απλό δίκτυο ADALINE με δύο εισόδους.	72
6.3	Διαχωριστική γραμμή για ένα δίκτυο ADALINE δύο εισόδων.	72
7.1	Ένα πλήρως συνδεδεμένο πολυστρωματικό δίκτυο πρόσθιας τροφοδότησης. .	75
7.2	Γραφική αναπαράσταση του προβλήματος XOR	77
7.3	Όρια απόφασης για κάθε νευρώνα του πρώτου επιπέδου στο πρόβλημα XOR .	78
7.4	Πολυστρωματικό νευρωνικό δίκτυο για το πρόβλημα XOR	78
7.5	Τα όρια απόφασης έτσι όπως διαμορφώνονται τελικά για το πρόβλημα XOR . .	78
7.6	Η διαδικασία εκπαίδευσης της όπισθεν διάδοσης σφάλματος.	82
8.1	Περιοχή εντοπισμού του ελάχιστου σημείου.	91
8.2	Μείωση της περιοχής εντοπισμού του ελάχιστου σημείου.	91
8.3	Σύγκλιση με τη μέθοδο των συζυγών κλίσεων.	92
8.4	Τα βήματα του αλγορίθμου των συζυγών κλίσεων.	92
9.1	Γραφική αναπαράσταση της συνάρτησης $y = \sin(x) * \cos(4\pi x)$	99
9.2	Έξοδος του δικτύου για 5 κρυφούς νευρώνες.	100
9.3	Έξοδος του δικτύου για 10 κρυφούς νευρώνες.	101
9.4	Έξοδος του δικτύου για 15 κρυφούς νευρώνες.	101
9.5	Μέσο τετραγωνικό σφάλμα συναρτήσεως του αριθμού των κρυφών νευρώνων. . .	101
9.6	Μέσο τετραγωνικό σφάλμα κατά τη διάρκεια εκπαίδευσης ανά ομάδα προτύπων εισόδου.	102
9.7	Μέσο τετραγωνικό σφάλμα κατά τη διάρκεια εκπαίδευσης ανά πρότυπο εισόδου. .	103
9.8	Μέσο τετραγωνικό σφάλμα κατά τη διάρκεια εκπαίδευσης ανά ομάδα προτύπων εισόδου για το πρόβλημα Sonar.	106
9.9	Μέσο τετραγωνικό σφάλμα κατά τη διάρκεια εκπαίδευσης ανά πρότυπο εισόδου για το πρόβλημα Sonar.	106

Κατάλογος Πινάκων

2.1	Συναρτήσεις ενεργοποίησης	29
2.2	Διαφορές των Τεχνητών Νευρωνικών Δικτύων με τις άλλες υπολογιστικές τεχνικές	38
7.1	Πίνακας αληθείας της συνάρτησης XOR	76
9.1	Αποτελέσματα εκπαίδευσης για τη συνάρτηση y χρησιμοποιώντας διαφορετικό αριθμό κρυφών νευρώνων	100
9.2	Συγκριτικά αποτελέσματα των διαφόρων μεθόδων εκπαίδευσης για μια συνεχή συνάρτηση	102
9.3	Απόδοση του κλασικού αλγορίθμου όπισθεν διάδοσης σφάλματος για διαφορετικές τιμές του ρυθμού εκπαίδευσης	102
9.4	Πίνακας αληθείας του προβλήματος της περιττής ισοτιμίας	103
9.5	Συγκριτικά αποτελέσματα των διαφόρων μεθόδων εκπαίδευσης για το πρόβλημα της περιττής ισοτιμίας	103
9.6	Συγκριτικά αποτελέσματα των διαφόρων μεθόδων εκπαίδευσης για το πρόβλημα της αναγνώρισης αριθμών	104
9.7	Συγκριτικά αποτελέσματα των μεθόδων εκπαίδευσης για το πρόβλημα Sonar .	104
9.8	Συγκριτικά αποτελέσματα των μεθόδων εκπαίδευσης με σύνολα αξιολόγησης και δοκιμής για το πρόβλημα Sonar	105

Μέρος Ι

Θεωρητικό Μέρος

Κεφάλαιο 1

Η πορεία προς τα Τεχνητά Νευρωνικά Δίκτυα

1.1 Εισαγωγή

Η Τεχνητή Νοημοσύνη (ΤΝ) αποτελεί ένα πεδίο της Επιστήμης Υπολογιστών που επιχειρεί να δώσει ανθρώπινες ικανότητες όπως είναι για παράδειγμα η αναγνώριση προτύπων και η ταξινόμησή τους σε κατηγορίες στα υπολογιστικά συστήματα. Μία από τις κύριες τεχνικές μέσω της οποίας μπορούν να αποδοθούν ανθρώπινες ιδιότητες σε έναν υπολογιστή είναι η χρησιμοποίηση των Τεχνητών Νευρωνικών Δικτύων.

Οι άνθρωποι δεν είναι πιο γρήγοροι ούτε πιο ακριβείς από τους υπολογιστές. Οι σημερινοί υπολογιστές φαίνονται πολύ πιο ικανοί στο να εκτελούν πολύπλοκες αριθμητικές και λογικές πράξεις. Σε πολλούς όμως τομείς η ανθρώπινη ικανότητα υπερτερεί κατά πολύ των δυνατοτήτων των μηχανών. Για παράδειγμα οι άνθρωποι αναγνωρίζουν πολύ πιο εύκολα αντικείμενα και κατανοούν τις σχέσεις μεταξύ τους μέσα στο φυσικό περιβάλλον, έστω κι αν αυτά είναι παραμορφωμένα ή δεν είναι ορατά εξ'ολοκλήρου. Η ικανότητα της μάθησης μέσω εμπειρίας είναι ένα από τα κύρια χαρακτηριστικά της ανθρώπινης νοημοσύνης. Επιπλέον η ανθρώπινη μνήμη μπορεί να αποθηκεύσει μεγάλη ποσότητα πληροφοριών και έχει την δυνατότητα να βρίσκει συσχετιζόμενες πληροφορίες μέσα σε αυτή χωρίς μεγάλη προσπάθεια. Αντίθετα οι υπολογιστές έχουν την δυνατότητα να απομνημονεύουν τεράστιες ποσότητες πληροφοριών, αλλά είναι δύσκολο να κάνουν αυτή την γνώση ένα αποτελεσματικό μέρος των δραστηριοτήτων τους [36].

Για παράδειγμα ένας υπολογιστής ο οποίος έχει προγραμματιστεί έτσι ώστε να αναγνωρίζει το χειρόγραφο κεφαλαίο Α θα αποτύχει να αναγνωρίσει το γράμμα αν δοθεί όπως φαίνεται στο Σχήμα 1.1.

Αντίθετα ο άνθρωπος μπορεί εύκολα να αναγνωρίσει το γράμμα χωρίς να έχει δει την εικόνα ξανά στο παρελθόν. Αυτό συμβαίνει επειδή ο ανθρώπινος εγκέφαλος αναγνωρίζει τα χαρακτηριστικά στοιχεία ενός κεφαλαίου Α και η αντίληψή του δεν επηρεάζεται από καλλιγραφίες στην εικόνα. Σε ένα υπολογιστικό σύστημα όμως που είναι προγραμματισμένο να αναγνωρίζει την εικόνα βάση συγκεκριμένων βασικών χαρακτηριστικών τα επιπλέον καλλι-



Σχήμα 1.1: Ένα χειρόγραφο κεφαλαίο Α.

γραφικά στοιχεία θα αποτρέψουν την κατηγοριοποίηση της.

Επιπλέον η αναγνώριση προσώπων ανάμεσα στο πλήθος είναι μια διαδικασία εύκολη για τον ανθρώπινο εγκέφαλο. Ο άνθρωπος δηλαδή έχει την ικανότητα να αναγνωρίζει εύκολα και γρήγορα γνώριμα πρόσωπα και αντικείμενα. Επίσης ένας οικονομολόγος μπορεί εύκολα να προβλέψει τις τάσεις και πού θα στραφεί το ενδιαφέρον. Ο τρόπος με τον οποίο οι άνθρωποι έχουν αναπτύξει την ικανότητα να αναγνωρίζουν εικόνες και να επεξεργάζονται δεδομένα είναι μέσω της δοκιμής και της επανάληψης, η μάθηση δηλαδή γίνεται μέσω της εμπειρίας. Αντίθετα τα προβλήματα αυτά είναι πολύπλοκα για ένα υπολογιστικό σύστημα καθώς δεν είναι δυνατή η εύρεση ενός απλού βήμα προς βήμα αλγορίθμου για την επίλυσή τους. Μια πιθανή εξήγηση είναι ότι ο ανθρώπινος εγκέφαλος και οι υπολογιστές λειτουργούν εντελώς διαφορετικά. Οι άνθρωποι είναι πιο 'έξυπνοι' από τους υπολογιστές επειδή ο εγκέφαλος τους χρησιμοποιεί μια βασική υπολογιστική αρχιτεκτονική η οποία είναι πιο κατάλληλη για την αντιμετώπιση της φυσικής επεξεργασίας πληροφοριών.

Για να αντιμετωπίσει αυτά τα προβλήματα ο ανθρώπινος εγκέφαλος και να επεξεργάζεται τις πληροφορίες χρησιμοποιεί ένα δίκτυο διασυνδεδεμένων στοιχείων που ονομάζονται νευρώνες. Ο εγκέφαλος είναι ένας απίστευτα δυνατός υπολογιστής. Μόνο ο φλοιός του περιέχει εκατομμύρια νευρώνες. Από κάθε νευρώνα ξεκινούν χιλιάδες ίνες οι οποίες συναντούν ίνες από άλλους νευρώνες σε σημεία που αποκαλούνται συνάψεις. Κάθε νευρώνας στέλνει σήματα διέγερσης ή αποδιέγερσης σε άλλους νευρώνες. Η κατάσταση ενεργοποίησης ενός νευρώνα εξαρτάται από τα σήματα διέγερσης ή αποδιέγερσης που λαμβάνει από τους νευρώνες με τους οποίους συνδέεται. Ο συνδυασμός του τεράστιου αυτού αριθμού απλών υπολογιστικών στοιχείων (νευρώνες) δίνει ένα πολύ ισχυρό επακόλουθο.

Η μελέτη των βιολογικών νευρωνικών δικτύων είναι σχετικά πρόσφατη και δύσκολη λόγω της πολυπλοκότητας του εγκεφάλου και οι λειτουργίες του δεν έχουν ακόμη πλήρως ανακα-

λυφθεί. Όλες οι βιολογικές νευρωνικές λειτουργίες, συμπεριλαμβανομένου και της μνήμης, αποθηκεύονται στους νευρώνες και στις συνδέσεις μεταξύ αυτών. Η διαδικασία της μάθησης θεωρείται ως η δημιουργία νέων συνδέσεων μεταξύ των νευρώνων ή την τροποποίηση των ήδη υπάρχοντων συνδέσεων. Υπάρχουν περίπου 10^{11} διασυνδεδεμένοι νευρώνες, ο κάθε νευρώνας είναι αυτόνομος και ανεξάρτητος και η λειτουργία του είναι ασύγχρονη, δηλαδή δεν απαιτείται ο συγχρονισμός άλλων γεγονότων.

Η δυνατότητες αυτές του εγκεφάλου ενέπνευσαν την περαιτέρω μελέτη της δομής των νευρώνων και τη δημιουργία υπολογιστικών δομών που η οργάνωσή τους είναι εμπνευσμένη από το βιολογικό μοντέλο. Η προσομοίωση της λειτουργίας των βιολογικών νευρώνων έγινε με τη δημιουργία των Τεχνητών Νευρωνικών Δικτύων. Τα δίκτυα αυτά των τεχνητών νευρώνων δεν έχουν την πολυπλοκότητα και την υπολογιστική δύναμη του ανθρώπινου εγκεφάλου, μπορούν όμως να εκπαιδευτούν για να εκτελούν χρήσιμες διεργασίες.

Το Τεχνητό Νευρωνικό Δίκτυο σχεδιάστηκε ώστε να αντιγράψει τα χαρακτηριστικά και τις λειτουργίες των βιολογικών νευρώνων του ανθρώπινου εγκεφάλου. Πρόκειται για ένα σύστημα επεξεργασίας πληροφορίας σχεδιασμένο με βάση την λογική συνδεσμολογία των νευρώνων του ανθρώπινου εγκεφάλου. Αποτελείται από ένα μεγάλο αριθμό πολλαπλά διασυνδεδεμένων επεξεργαστών (νευρώνες) που εργάζονται σε πλήρη συμφωνία μεταξύ τους και εκτελούν διάφορους υπολογισμούς πάνω σε δεδομένα εισόδου, τα τμήματα αυτά συνδέονται μεταξύ τους με αμφίδρομα κανάλια επεξεργασίας που μεταφέρουν κωδικοποιημένες πληροφορίες με την μορφή αριθμών.

Τα Τεχνητά Νευρωνικά Δίκτυα διέπονται από κανόνες εκπαίδευσης που προσαρμόζονται με βάση τα πρότυπα των τιμών εισόδου και έχουν την δυνατότητα μάθησης μέσα από τα παραδείγματα (καταχώρηση εμπειρικής γνώσης) και παρέχουν λύσεις στα προβλήματα που απαιτούν την ανθρώπινη παρατήρηση και σκέψη (διαθέσιμη γνώση για χρήση).

1.2 Στόχος της Εργασίας

Τα νευρωνικά δίκτυα αποτελούν μια σχετικά νέα περιοχή στις φυσικές επιστήμες, καθόσον έχουν γίνει γνωστά και έχουν αναπτυχθεί σε διεθνές επίπεδο μόνο κατά τις τελευταίες δεκαετίες. Αποτελούν επομένως ένα θέμα με μεγάλο ενδιαφέρον στις τεχνολογικές επιστήμες. Το κύριο χαρακτηριστικό τους είναι ότι οι πρώτες αρχές και λειτουργίες τους βασίζονται και εμπνέονται από το νευρικό σύστημα των ζώντων οργανισμών, αλλά η μελέτη και η χρήση τους έχει προχωρήσει πολύ πέρα από τους βιολογικούς οργανισμούς. Σήμερα τα νευρωνικά δίκτυα χρησιμοποιούνται για να λύσουν κάθε είδους προβλήματα με ηλεκτρονικό υπολογιστή.

Ο στόχος της παρούσας διπλωματικής εργασίας είναι αφ'ενός η θεωρητική προσέγγιση των μεθόδων εκπαίδευσης των Τεχνητών Νευρωνικών Δικτύων και αφετέρου η προσπάθεια επίλυσης προβλημάτων με τη βοήθεια του αλγορίθμου της όπισθεν διάδοσης σφάλματος. Η μέθοδος της όπισθεν διάδοσης σφάλματος αναπτύσσεται σε βάθος και επιλέχθηκε για την επίλυση των προβλημάτων επειδή το πρότυπο αυτό δεσπόζει σήμερα ανάμεσα σε όλες τις τεχνικές που χρησιμοποιούνται και μπορεί να επιλύσει οποιοδήποτε γραμμικό ή μη

γραμμικό πρόβλημα.

Βασικός σκοπός της εργασίας είναι η κατανόηση των νευρωνικών δικτύων. Αρχικά μελετούνται τα ιδιαίτερα χαρακτηριστικά τους, η διαδικασία εκπαίδευσης που ακολουθούν και ο τρόπος λειτουργίας τους. Μετά τη θεωρητική ενασχόληση με τα Τεχνητά Νευρωνικά Δίκτυα γίνεται εφαρμογή αυτών για την επίλυση συγκεκριμένων προβλημάτων. Αναλύονται τα νευρωνικά δίκτυα που χρησιμοποιούν τον αλγόριθμο της όπισθεν διάδοσης σφάλματος και γίνεται σύγκριση δύο βασικών μεθόδων εκπαίδευσης, της εκπαίδευσης ανά ομάδα προτύπων εισόδου και της εκπαίδευσης ανά πρότυπο εισόδου. Οι δύο μέθοδοι εκπαίδευσης των νευρωνικών δικτύων σχολιάζονται ως προς την απόδοση και την αποτελεσματικότητα τους προκειμένου να αποτελέσουν ένα χρήσιμο εργαλείο για την επιλογή του κατάλληλου τύπου εκπαίδευσης. Σε κάθε πρόβλημα λοιπόν σταθμίζονται τα πλεονεκτήματα και μειονεκτήματα και επιλέγεται μια εκ των δύο μεθόδων εκπαίδευσης.

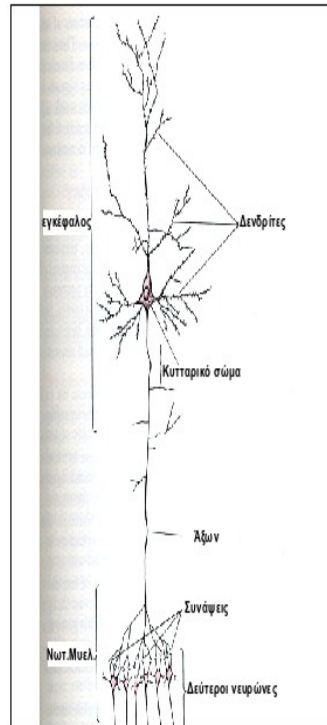
1.3 Προσομοίωση του βιολογικού μοντέλου

Η έμπνευση για τα νευρωνικά δίκτυα, όπως αναφέρθηκε παραπάνω, ξεκινά από την βιολογία. Οι ζώντες οργανισμοί, από τους πιο απλούς μέχρι τον άνθρωπο, έχουν ένα νευρικό σύστημα το οποίο είναι υπεύθυνο για μια πλειάδα από διεργασίες, όπως είναι η επαφή με τον εξωτερικό κόσμο, η μάθηση και η μνήμη. Για την κατανόηση των Τεχνητών Νευρωνικών Δικτύων είναι χρήσιμο αρχικά να μελετήσουμε αυτά τα χαρακτηριστικά της εγκεφαλικής λειτουργίας που ενέπνευσαν την ανάπτυξη αυτών των δικτύων.

Ο ανθρώπινος εγκέφαλος αποτελείται από ένα μεγάλο αριθμό (10^{11}) διασυνδεδεμένων στοιχείων (10^4 συνδέσεις ανά στοιχείο) που αποκαλούνται νευρώνες. Ο νευρώνας αποτελεί τη βασική λειτουργική και ανατομική μονάδα του νευρικού συστήματος. Οι νευρώνες συνεχώς και ασταμάτητα επεξεργάζονται πληροφορίες, παίρνοντας και στέλνοντας ηλεκτρικά σήματα σε άλλους νευρώνες. Κάθε νευρώνας αποτελείται από το κυτταρικό σώμα και από τις προεκτάσεις ή αποφυάδες αυτού (δενδρίτες και άξων) που ποικίλλουν σε μέγεθος και ονομάζονται νευρικές ίνες. Οι δενδρίτες είναι βραχείες διακλαδιζόμενες ίνες οι οποίες φυσιολογικά δέχονται ώσεις από τα τελικά περιφερικά τους άκρα και τις μεταφέρουν στο κυρίως σώμα του κυττάρου. Ο άξων είναι μια μακρά ίνα του νευρικού κυττάρου και μεταφέρει ώσεις από αυτό προς άλλα κύτταρα. Η δομή που έχουν οι συνδεδεμένοι νευρώνες καθορίζεται μερικώς από τη γέννηση. Συνδέσεις όμως δημιουργούνται ή καταστρέφονται και κατά τη διάρκεια της ζωής του ανθρώπου κατά τη διαδικασία της μάθησης [29].

Η μεταφορά των ώσεων από νευρώνα σε νευρώνα πραγματοποιείται στη «σύναψη», το σημείο όπου ο άξωνας ενός κυττάρου έρχεται σε επαφή με το σώμα ενός άλλου κυττάρου ή με τους δενδρίτες του. Τα νευρικά κύτταρα φυσιολογικά διεγείρονται μόνο από μια ομάδα περιφερικών διακλαδώσεων και μεταφέρουν την ώση μόνο προς μια κατεύθυνση αντίθετα από την περιοχή της διέγερσης. Οι «προσαγωγές ίνες» (δενδρίτες) μεταφέρουν την ώση προς το κυτταρικό σώμα, οι «απαγωγές ίνες» (άξων) μεταφέρουν την ώση από το κυτταρικό σώμα.

Στο άκρο κάθε άξωνα υπάρχει μεγάλος αριθμός τελικών κομβίων (πάχυνση του τελικού άκρου) τα οποία έρχονται σχεδόν σε επαφή με την μεμβράνη ενός επόμενου νευρώνα και



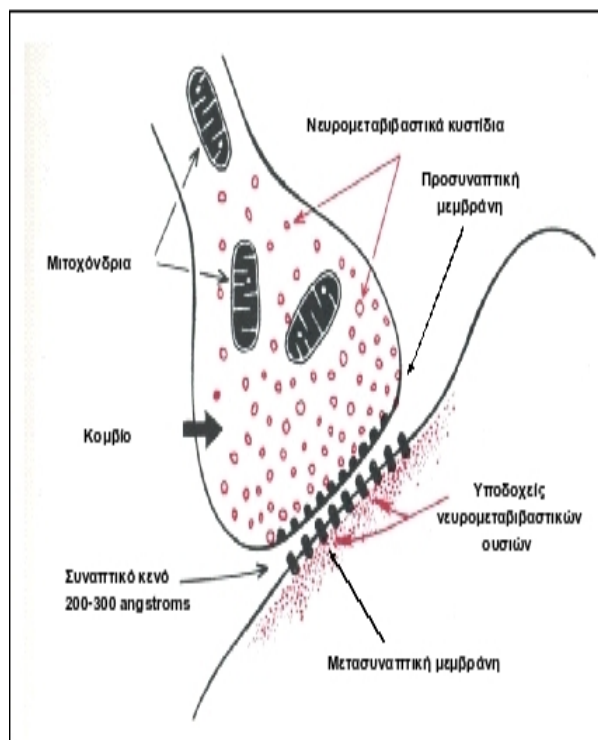
Σχήμα 1.2: Σχηματική απεικόνιση νευρωνικού κυττάρου.

αποτελούν μαζί τις συνάψεις, μέσω των οποίων μεταβιβάζεται η διέγερση από το ένα νευρικό κύτταρο στο άλλο. Ειδικότερα η σύναψη αποτελείται από το τελικό κομβίο, την προσυναπτική μεμβράνη του κομβίου, το συναπτικό διάστημα και την μετασυναπτική μεμβράνη που ανήκει στον επόμενο νευρώνα (Σχήμα 1.3).

Υπάρχουν διεγερτικές και ανασταλτικές συνάψεις. Οι ανασταλτικές συνάψεις είναι το ίδιο σημαντικές με τις διεγερτικές γιατί έχουν την ικανότητα να περιορίζουν ή και να επιλέγουν την συνεχή ροή των νευρικών ώσεων. Έτσι τα σημαντικά σήματα μεταβιβάζονται ενώ τα μη σημαντικά καταστέλλονται. Η διέγερση μεταβιβάζεται με χημικές ουσίες, οι οποίες εκκρίνονται από τα κυστίδια, από τις οποίες η πλέον συχνή είναι η ακετυλοχολίνη (ACh), αλλά και οι κατεχολαμίνες καθώς και η ντοπαμίνη και η σεροτονίνη δρουν ως νευρομεταβιθαστές. Η δραστική ουσία στις ανασταλτικές συνάψεις θεωρείται ότι είναι το γ-αμινο-βουτυρικό οξύ (GABA). Στο συναπτικό διάστημα αυτές οι ουσίες προκαλούν εκπόλωση της μετασυναπτικής μεμβράνης και έτσι γίνεται η μεταβίβαση της διέγερσης. Μετά τη διέγερση αυτές οι νευρομεταβιθαστικές ουσίες πρέπει να εξουδετερωθούν για να μη συνεχίζεται η εκπόλωση, αυτό επιτυγχάνεται αμέσως με ενζυμικά συστήματα.

Το πλάτος της σύναψης, η απόστασή της από τον δενδρίτη και η πυκνότητα του ηλεκτροχημικού υλικού επηρεάζουν την ευκολία με την οποία η ηλεκτρική δραστηριότητα μεταδίδεται από τον άξονα στο δενδρίτη. Το ποσοστό της ηλεκτρικής δραστηριότητας που μεταδίδεται τελικά στο δενδρίτη λέγεται συναπτικό βάρος.

Οι τελικοί κλάδοι ενός νευρικού άξονα ονομάζονται τελικά δενδρύλλια και μπορεί να



Σχήμα 1.3: Σχηματική απεικόνιση της σύναψης μεταξύ 2 νευρικών κυττάρων.

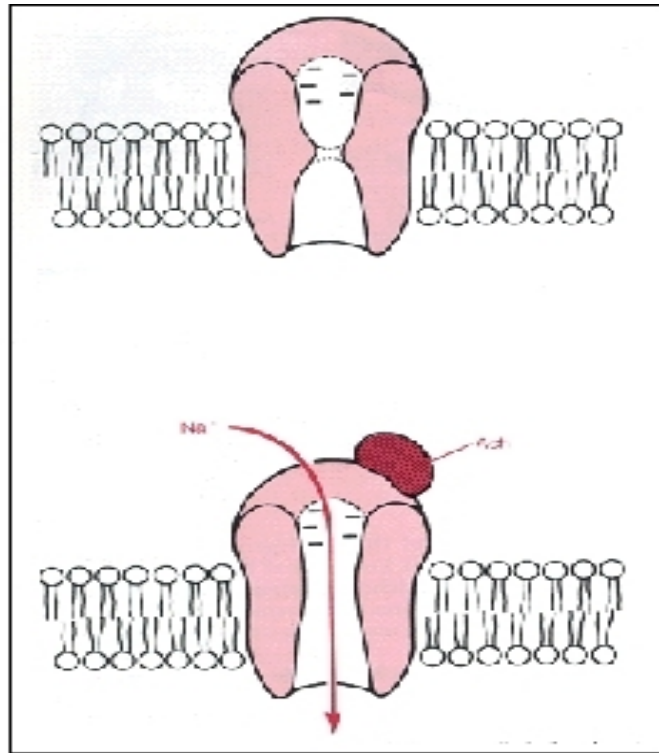
έρχονται σε επαφή με περισσότερους μετασυναπτικούς νευρώνες, έτσι ώστε μια ώση να μεταδίδεται ταυτόχρονα σε πολλά νευρικά κύτταρα. Άλλοτε πάλι η διέγερση ενός νευρώνα προκαλεί ταυτόχρονη αναστολή σε όλα τα γειτονικά νευρικά κύτταρα.

1.3.1 Σύγκριση βιολογικών και των Τεχνητών Νευρωνικών δικτύων.

Αν και οι βιολογικοί νευρώνες είναι πιο αργοί σε σύγκριση με τα ηλεκτρικά κυκλώματα (10^{-3} s σε σύγκριση με 10^{-9} s), ο εγκέφαλος έχει την ικανότητα να εκτελεί διεργασίες πολύ πιο γρήγορα από οποιονδήποτε υπολογιστή. Αυτό οφείλεται εν μέρη από τη μαζική παράλληλη δομή των βιολογικών νευρωνικών δικτύων. Όλοι οι νευρώνες μπορούν να λειτουργούν ταυτόχρονα και τα τεχνητά νευρωνικά δίκτυα αντιγράφουν αυτή την παράλληλη δομή.

Τα Τεχνητά Νευρωνικά Δίκτυα αποτελούνται από στοιχεία (τεχνητοί νευρώνες) τα οποία συμπεριφέρονται κατά τρόπο ανάλογο των πιο στοιχειωδών λειτουργιών των βιολογικών κυττάρων. Οι τεχνητοί νευρώνες είναι οργανωμένοι κατά τέτοιο τρόπο ώστε να προσομοιώνουν την ανατομία του ανθρώπινου εγκεφάλου. Παρά την όχι όμως και τόσο μεγάλη ομοιότητα τους με τον εγκέφαλο επιτυγχάνουν να προσεγγίσουν ένα μεγάλο αριθμό χαρακτηριστικών της δομής αλλά και της λειτουργίας του εγκεφάλου. Για παράδειγμα μαθαίνουν χρησιμοποιώντας εμπειρία την οποία έχουν συσσωρεύσει, έχουν την ικανότητα γενίκευσης από προηγούμενα παραδείγματα σε νέα, μπορούν να επεξεργαστούν μια ομάδα δεδομένων και να ξεχωρίσουν από αυτή τα πιο ουσιώδη χαρακτηριστικά.

Οι πρώτες έρευνες σχετικά με τη λειτουργία και την πρακτική εφαρμογή σε ηλεκτρονικό



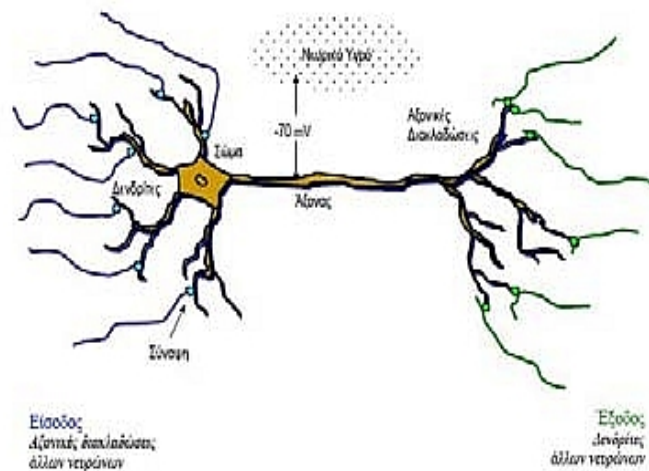
Σχήμα 1.4: Μηχανισμός εκπόλωσης μιας νευρικής ίνας.

υπολογιστή μεθόδων που προσομοιώνουν τη λειτουργία του νευρικού συστήματος και του εγκεφάλου ζωντανών οργανισμών ξεκινά ουσιαστικά με την εργασία των McCulloch και Pitts το 1943 [26], όπου προτάθηκε για πρώτη φορά η δομή του νευρώνα. Η βασική δομή του νευρώνα περιελάμβανε ένα σύνολο παραμέτρων που χρησιμοποιούνταν για την υλοποίηση μιας δυαδικής συσκευής κατωφλίου. Λαμβάνοντας ως είσοδο ένα σύνολο ερεθισμάτων, υπολογίζεται το σταθμισμένο άθροισμα των εισόδων με τις αντίστοιχες παραμέτρους (βάρη) και η τελική έξοδος του νευρώνα είναι η τιμή $+1$ ή -1 , ανάλογα με το όριο κατωφλίου που χρησιμοποιείται (συνήθως την αρχή των αξόνων). Η συνάρτηση αυτή ονομάζεται συνάρτηση ενεργοποίησης ή συνάρτηση μεταφοράς. Έκτοτε πολλές εναλλακτικές μορφές νευρώνων έχουν προταθεί, όπως η χρήση σιγμοειδούς συνάρτησης, υπερβολικών τριγωνομετρικών συναρτήσεων ή συναρτήσεων πιθανοτικών κατανομών στη θέση της απλής συνάρτησης κατωφλίου, όμως η βασική δομή των νευρώνων παραμένει η ίδια.

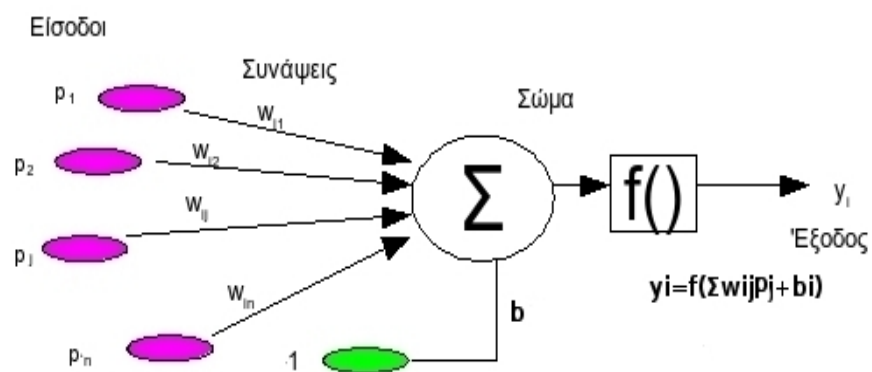
Συνοπτικά, ένας βιολογικός νευρώνας λαμβάνει σήματα από άλλους νευρώνες μέσω των δενδριτών. Το άθροισμα των διεγέρσεων σε όλους τους δενδρίτες αποτελεί τη συνολική διεγερση του νευρώνα. Στο σώμα του νευρώνα πραγματοποιείται η επεξεργασία αυτών των σημάτων, τα οποία συνδυάζει με κάποιον τρόπο και εκτελεί μια μη γραμμική διεργασία για την απόκριση της εξόδου. Το σήμα εξόδου μεταφέρεται μέσω του άξονα και τελικά μέσω των συνάψεων σε άλλους νευρώνες (Σχήμα 1.5).

Αντίστοιχα ένας τεχνητός νευρώνας δέχεται ένα διάνυσμα εισόδων $p(n)$. Κάθε είσοδος πολλαπλασιάζεται με έναν πραγματικό αριθμό που ονομάζεται συντελεστής βάρους. Τα

συναπτικά βάρη w χρησιμοποιούνται για την τροποποίηση των εισόδων του νευρώνα και αντιστοιχούν στη δύναμη μιας σύναψης. Οι τροποποιημένες εισοδοι αθροίζονται και στο αποτέλεσμα που συχνά χαρακτηρίζεται ως είσοδος του δικτύου εφαρμόζεται μια συνάρτηση γνωστή ως συνάρτηση ενεργοποίησης. Το κυτταρικό σώμα του νευρώνα αναπαρίσταται από τον αθροιστή και τη συνάρτηση ενεργοποίησης και η έξοδος του τεχνητού νευρώνα y αναπαριστά το σήμα στον άξονα του νευρώνα (Σχήμα 1.6).



Σχήμα 1.5: Μοντέλο βιολογικού νευρώνα.



Σχήμα 1.6: Μοντέλο τεχνητού νευρώνα.

Κεφάλαιο 2

Τεχνητά Νευρωνικά Δίκτυα

Τα Τεχνητά Νευρωνικά Δίκτυα (Τ.Ν.Δ) είναι μαζικά παράλληλοι κατανεμημένοι επεξεργαστές αποτελούμενοι από απλές μονάδες επεξεργασίας, που έχουν τη δυνατότητα αποθήκευσης «γνώσης» προερχόμενης από την εμπειρία, μέσω της διαδικασίας εκπαίδευσης. Σε ένα δίκτυο μια επεξεργαστική μονάδα είναι ένας τεχνητός νευρώνας με περιορισμένη μνήμη και επεξεργαστική ισχύ. Κάθε τεχνητός νευρώνας δέχεται εισόδους από νευρώνες με τους οποίους συνδέεται και υπολογίζει μια τιμή εξόδου (σαν συνάρτηση των εισόδων του), την οποία την διοχετεύει με την σειρά του σε άλλους νευρώνες με τους οποίους επικοινωνεί. Το σύστημα λειτουργεί παράλληλα και πολλές μονάδες έχουν δυνατότητα να πραγματοποιούν ταυτόχρονα τους υπολογισμούς τους. Υπάρχουν τρεις κατηγορίες μονάδων: μονάδες εισόδου οι οποίες λαμβάνουν τα δεδομένα εισόδου από εξωτερικές πηγές, μονάδες εξόδου οι οποίες στέλνουν τα αποτελέσματα εκτός συστήματος και κρυφές μονάδες. Οι κρυφές μονάδες δεν είναι ορατές στον εξωτερικό κόσμο και οι εισοδοί τους καθώς και οι έξοδοί τους βρίσκονται εντός του Τ.Ν.Δ.

2.1 Στοιχεία ενός Τεχνητού Νευρώνα

Ένας νευρώνας λαμβάνει συνήθως πολλές ταυτόχρονες εισόδους. Κάθε είσοδος έχει το αντίστοιχο βάρος της. Αυτά τα βάρη εκτελούν τον ίδιο τύπο λειτουργίας όπως οι ποικίλες συναπτικές δυνάμεις των βιολογικών νευρώνων. Και στις δύο περιπτώσεις, κάποια σήματα εισόδου είναι σημαντικότερα από άλλα έτσι ώστε να έχουν μεγαλύτερη επίδραση στο νευρώνα καθώς συνδυάζονται για να παραγάγουν μια απάντηση.

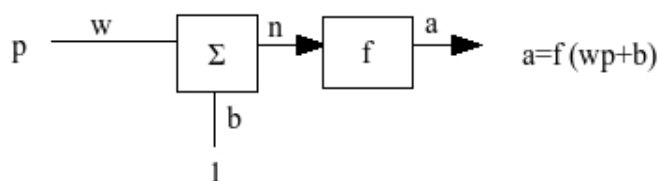
Τα βάρη είναι προσαρμοστικοί συντελεστές μέσα στο δίκτυο που καθορίζουν την ένταση του σήματος εισόδου καθώς εισάγεται στον τεχνητό νευρώνα. Αποτελούν ένα μέτρο της συνδεδετικής δύναμης της εισόδου. Αυτές οι δυνάμεις μπορούν να τροποποιηθούν ανάλογα με τα διάφορα πρότυπα εκπαίδευσης και σύμφωνα με τη συγκεκριμένη τοπολογία του δικτύου ή μέσω των κανόνων εκπαίδευσης.

Το πρώτο βήμα της λειτουργίας ενός τεχνητού νευρώνα είναι να υπολογιστεί το βεβαρημένο άθροισμα όλων των εισόδων. Από μαθηματική άποψη, οι εισοδοί και τα αντίστοιχα βάρη είναι διανύσματα που μπορούν να αντιπροσωπευθούν ως (p_1, p_2, \dots, p_n) και

(w_1, w_2, \dots, w_n) . Το συνολικό σήμα εισόδου είναι το εσωτερικό γινόμενο αυτών των δύο διανυσμάτων.

Στο Σχήμα 2.1 παρουσιάζεται το μοντέλο ενός απλού νευρώνα που δέχεται μια είσοδο. Η είσοδος p πολλαπλασιάζεται με το συντελεστή βάρους w και το αποτέλεσμα στέλνεται στον αθροιστή. Το αποτέλεσμα του αθροιστή περνάει από μια συνάρτηση ενεργοποίησης f η οποία παράγει την έξοδο του νευρώνα a . Η έξοδος του νευρώνα δηλαδή υπολογίζεται από τη συνάρτηση $a = f(wp + b)$. Ο νευρώνας διαθέτει ακόμη μια σύνδεση την πόλωση b . Η πόλωση είναι ουσιαστικά ένα συναπτικό βάρος με τη διαφορά ότι η είσοδος που αντιστοιχεί σε αυτή έχει πάντα την τιμή 1. Η πόλωση είναι μια επιπλέον είσοδος που χρησιμοποιείται προκειμένου να αυξήσει ή να μειώσει το σταθμισμένο άθροισμα ανάλογα με το αν αυτό είναι θετικό ή αρνητικό. Η πόλωση δεν είναι απαραίτητο στοιχείο του νευρώνα και μπορεί να παραλειφθεί.

Η συνάρτηση ενεργοποίησης f επιλέγεται από τον κατασκευαστή του δικτύου και ανάλογα με τις απαιτήσεις του προβλήματος και οι παράμετροι w και b προσαρμόζονται ανάλογα με τη χρήση ενός κανόνα εκπαίδευσης έτσι ώστε η σχέση μεταξύ της εισόδου και της εξόδου του νευρώνα να ικανοποιεί τον επιθυμητό στόχο του προβλήματος.



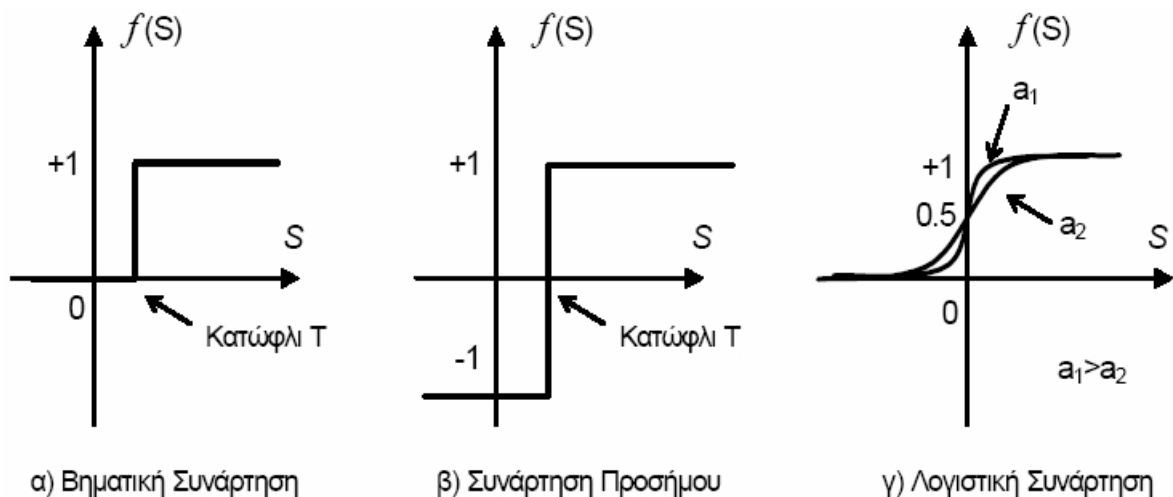
Σχήμα 2.1: Τεχνητός Νευρώνας με μία είσοδο.

Πριν την εφαρμογή της συνάρτησης ενεργοποίησης μπορεί να προστεθεί στο δίκτυο ομοιόμορφα κατανομημένος τυχαίος θόρυβος. Η πηγή και το ποσοστό αυτού του θορύβου καθορίζονται από τον τρόπο εκμάθησης ενός δεδομένου παραδείγματος δικτύων. Αυτός ο θόρυβος αναφέρεται συνήθως ως «θερμοκρασία» των τεχνητών νευρώνων. Το όνομα αυτό προσδιορίζεται από το φυσικό φαινόμενο ότι καθώς οι άνθρωποι γίνονται πάρα πολύ ζεστοί ή κρύοι η δυνατότητά τους να σκεφτούν επηρεάζεται. Ηλεκτρονικά, αυτή η διαδικασία προσομοιώνεται με την προσθήκη του θορύβου. Για να μιμηθούν περισσότερο τα χαρακτηριστικά της φύσης, μερικοί επιστήμονες χρησιμοποιούν γκαουσιανό θόρυβο. Η χρήση της θερμοκρασίας ανήκει στον πρόσφατο ερευνητικό τομέα και δεν χρησιμοποιείται σε πολλές πρακτικές εφαρμογές.

2.2 Συναρτήσεις ενεργοποίησης

Η συνάρτηση ενεργοποίησης ή συνάρτηση μεταφοράς είναι μια γραμμική ή μη γραμμική συνεχής συνάρτηση η οποία επιλέγεται έτσι ώστε να περιορίσει την έξοδο του συστήματος σε ένα επιθυμητό διάστημα. Με τη συνάρτηση ενεργοποίησης η έξοδος του αθροιστή συγκρίνεται με ένα κατώφλι για να υπολογιστεί η έξοδος του νευρώνα. Οι πιο γνωστές συναρτήσεις ενεργοποίησης είναι η βηματική ή δυαδική συνάρτηση, η συνάρτηση προσήμου, η γραμμική συνάρτηση και η σιγμοειδής ή λογιστική συνάρτηση.

Η δυαδική συνάρτηση θέτει την έξοδο του νευρώνα ίση με το 0 αν το όρισμα της συνάρτησης είναι μικρότερο ή ίσο του 0 και ίση με 1 αν το όρισμα είναι μεγαλύτερο του 0. Η έξοδος δηλαδή του νευρώνα περιορίζεται στις δυαδικές τιμές $\{0,1\}$. Η συνάρτηση αυτή χρησιμοποιείται για τη δημιουργία νευρώνων που διαχωρίζουν τις εισόδους σε δύο διαφορετικές κατηγορίες. Παρόμοια η συνάρτηση προσήμου έχει σαν αποτέλεσμα η έξοδος του νευρώνα να παίρνει τις τιμές $\{-1,1\}$ και η λογιστική συνάρτηση περιορίζει το πεδίο τιμών της εξόδου του νευρώνα στο διάστημα $(0,1)$. Η λογιστική συνάρτηση χρησιμοποιείται συνήθως σε πολυστρωματικά νευρωνικά δίκτυα τα οποία εκπαιδεύονται με τη μέθοδο back-propagation.



Σχήμα 2.2: Αναπαράσταση των βασικών συναρτήσεων

Στον πίνακα 2.1 φαίνονται συνοπτικά όλες οι συναρτήσεις ενεργοποίησης που αναφέραμε και ο συμβολισμός τους.

2.3 Τοπολογία νευρωνικών δικτύων

Η μορφή του δικτύου και ο ακριβής τρόπος διασύνδεσης των νευρώνων μεταξύ τους αποτελούν την τοπολογία του νευρωνικού δικτύου. Συνήθως οι μονάδες του δικτύου διατάσσονται σε ξεχωριστές δομές οι οποίες καλούνται «στρώματα» ή «επίπεδα» (layer). Το πρώτο

Πίνακας 2.1: Συναρτήσεις ενεργοποίησης

Όνομα Συνάρτησης	Ορισμός
Βηματική συνάρτηση	$f(s) = 0, \text{ αν } s \leq 0$ $f(s) = 1, \text{ αν } s > 0$
Συνάρτηση προσήμου	$f(s) = -1, \text{ αν } s \leq 0$ $f(s) = 1, \text{ αν } s > 0$
Γραμμική συνάρτηση	$f(s) = x$
Σιγμοειδής συνάρτηση	$f(s) = \frac{1}{1+e^s}$
Υπερβολική εφαπτομένη	$f(s) = \tanh(s) = \frac{1-e^{-s}}{1+e^{-s}}$
Θετικά Γραμμική συνάρτηση	$f(s) = 0, \text{ αν } s < 0$ $f(s) = s, \text{ αν } s \geq 0$
Ανταγωνιστική συνάρτηση	$f(s) = 1$ για τους νευρώνες με μέγιστο s $f(s) = 0$ για όλους του άλλους νευρώνες

από αυτά τα επίπεδα ονομάζεται επίπεδο εισόδου και χρησιμοποιείται για την εισαγωγή δεδομένων. Τα στοιχεία του δηλαδή δεν είναι ουσιαστικά νευρώνες, γιατί δεν εκτελούν κάποιο υπολογισμό, δεν έχουν βάρη εισόδου ούτε συνάρτηση ενεργοποίησης. Οι νευρώνες του κάθε στρώματος συνήθως έχουν την ίδια συνάρτηση ενεργοποίησης. Στα δίκτυα πολλαπλών στρωμάτων τα μεσαία στρώματα ονομάζονται κρυφά (hidden layers) και αποτελούνται από κρυφές μονάδες, ενώ στο τέλος υπάρχει το επίπεδο εξόδου.

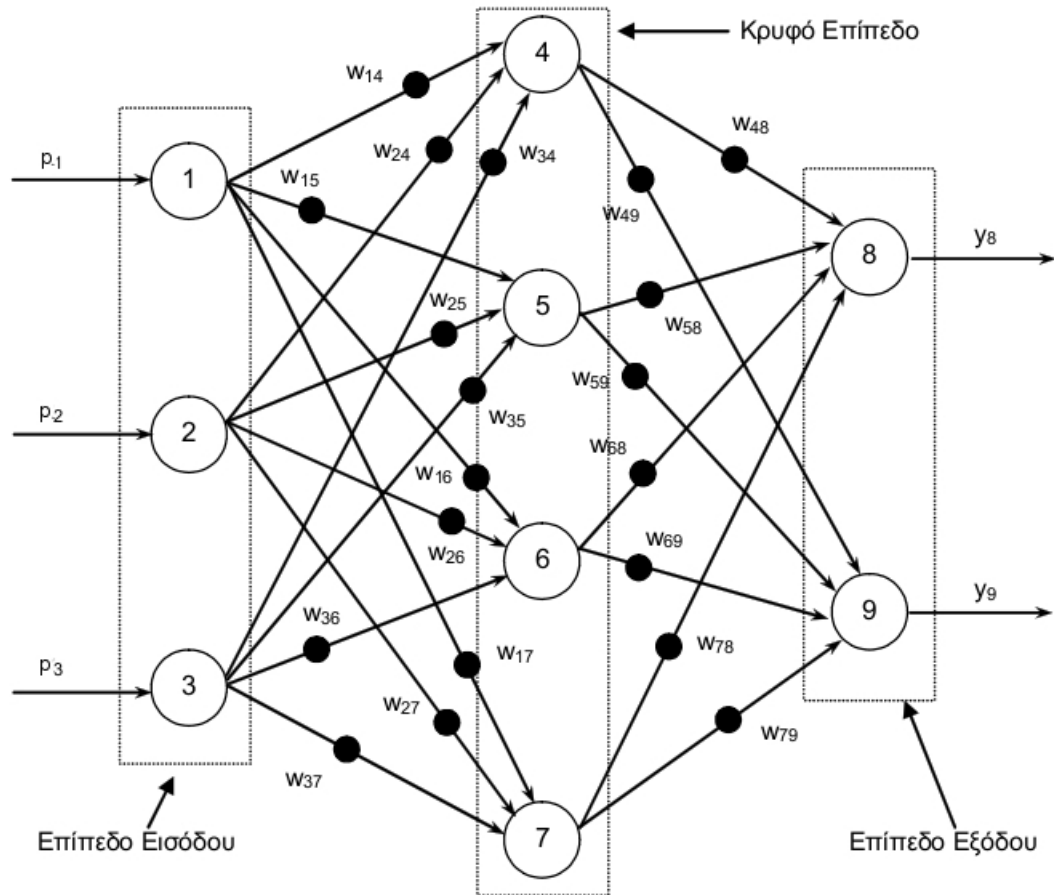
Οι νευρώνες στα Τεχνητά Νευρωνικά Δίκτυα μπορεί να είναι πλήρως ή μερικώς συνδεδεμένοι. Πλήρως συνδεδεμένοι (fully connected) είναι εκείνοι οι οποίοι συνδέονται με όλους τους υπόλοιπους νευρώνες. Σε κάθε άλλη περίπτωση οι νευρώνες είναι μερικώς συνδεδεμένοι (partially connected).

Τα πρωταρχικά μοντέλα αποτελούνται από ένα απλό στρώμα όπου κάθε είσοδος του δικτύου συνδέεται με όλες τις μονάδες του στρώματος αυτού. Οι πληροφορίες ρέουν δια μέσω του στρώματος από τις εισόδους στις εξόδους χωρίς ανάδραση των εξόδων. Τα μοντέλα αυτής της κλάσης ονομάζονται δίκτυα πρόσθιας τροφοδότησης ή feed-forward μοντέλα. Έχουν εισαχθεί επίσης μοντέλα τα οποία χρησιμοποιούν την έννοια των feed-backward συνδέσεων. Τα δίκτυα αυτά βελτιώνουν την διαδικασία εκπαίδευσης και μάθησης του δικτύου τροφοδοτώντας προς τα πίσω αποτελέσματα (σφάλματα) από το επόμενο στρώμα σε ένα προηγούμενο και ονομάζονται δίκτυα ανατροφοδότησης.

Τοπολογίες πολλαπλών επιπέδων προσφέρουν την απαιτούμενη μη γραμμικότητα για την επίλυση πολύπλοκων προβλημάτων αναγνώρισης προτύπων .

2.4 Εκπαίδευση Τεχνητών Νευρωνικών Δικτύων

Ο κανόνας μάθησης ή αλλιώς αλγόριθμος εκπαίδευσης ενός Τ.Ν.Δ. είναι μια διαδικασία για τον καθορισμό ή τη μεταβολή των συντελεστών βαρών και των πολώσεων του δικτύου έτσι ώστε δοθέντος συγκεκριμένου διανύσματος εισόδου να παραχθεί ένα συγκεκριμένο διάνυσμα



Σχήμα 2.3: Πλήρως συνδεδεμένο ΤΝΔ πρόσθιας τροφοδότησης 3-4-2.

εξόδου. Σκοπός του είναι να εκπαιδεύσει το δίκτυο ώστε αυτό να εκτελεί ένα ρόλο. Πρόκειται για την διαδικασία διά μέσου της οποίας το ΤΝΔ αποκρίνεται σε μία διέγερση ώστε αφού τροποποιήσει καταλλήλως τις μεταβλητές που το χαρακτηρίζουν να παράξει το επιθυμητό αποτέλεσμα.

Για να αλλάξουμε την δομή της επεξεργασίας ή της γνώσης σε ένα ΤΝΔ χρειάζεται να τροποποιήσουμε την αρχιτεκτονική του. Οι αλλαγές αυτές μπορούν να γίνουν είτε με την ανάπτυξη καινούργιων συνδέσμων ή την αφαίρεση παλιών συνδέσμων. Η εκπαίδευση αποτελεί επίσης μία διαδικασία συνεχούς ταξινομήσεως των σημάτων εισόδου. Όταν ένα σήμα εμφανίζεται στην είσοδο, τότε το Νευρωνικό Δίκτυο είτε το αναγνωρίζει, είτε δημιουργεί μια νέα κατηγορία στην οποία και το κατατάσει. Κατά την ολοκλήρωση της εκπαίδευσής, το Νευρωνικό Δίκτυο έχει ενσωματώσει «γνώση».

Η εκπαίδευση χωρίζεται σε τρεις κύριες κατηγορίες: στη μάθηση με επίβλεψη, στη μάθηση χωρίς επίβλεψη και στη βαθμολογημένη μάθηση.

1. Η πλειοψηφία των προβλημάτων που λύνονται με Τεχνητά Νευρωνικά Δίκτυα χρησιμοποιούν ως τρόπο εκπαίδευσης την μάθηση με επίβλεψη. Στη μάθηση με επίβλεψη το δίκτυο τροφοδοτείται με ένα σύνολο δεδομένων εισόδου και τις αντίστοιχες εξόδους. Το δίκτυο επεξεργάζεται τις εισόδους και συγκρίνει τα αποτελέσματα εξόδου με τις επιθυμητές εξόδους. Δίνεται δηλαδή ένα σύνολο δεδομένων από παραδείγματα της μορφής $\{p_1, t_1\}, \{p_2, t_2\}, \dots, \{p_n, t_n\}$, όπου p είναι οι εισοδοί στο δίκτυο και t οι επιθυμητές εξοδοί, είναι δηλαδή οι στόχοι του δικτύου. Τα δεδομένα αυτά αποκαλούνται δεδομένα εκπαίδευσης. Καθώς οι εισοδοί εφαρμόζονται στο δίκτυο, οι εξοδοί του δικτύου συγκρίνονται με τους στόχους. Η διαφορά που προκύπτει από τη σύγκριση αποτελεί το σφάλμα του δικτύου. Ο κανόνας εκπαίδευσης χρησιμοποιείται για να προσαρμόσουν τα βάρη προκειμένου να συγκλίνουν οι εξοδοί του δικτύου με τους επιθυμητούς στόχους. Η διαδικασία αυτή επαναλαμβάνεται έως ότου το σφάλμα που προκύπτει να είναι αποδεκτό. Κατά τη διάρκεια της εκπαίδευσης του δικτύου τα ίδια δεδομένα επεξεργάζονται πολλές φορές καθώς προσαρμόζονται τα βάρη.
2. Στη μάθηση χωρίς επίβλεψη τα βάρη και οι πολώσεις καθορίζονται σύμφωνα με τα δεδομένα εισόδου μόνο καθώς δεν υπάρχουν διαθέσιμες εξοδοί στόχοι. Απλώς δίνουμε την πληροφορία στο δίκτυο, αλλά δεν δίνουμε αντίστοιχους στόχους όπως προηγουμένως και έτσι δεν γίνεται κανένας έλεγχος ή σύγκριση για την πορεία του σφάλματος. Το δίκτυο από μόνο του θα πρέπει να αποφασίσει ποια χαρακτηριστικά των δεδομένων εκπαίδευσης θα χρησιμοποιήσει για να κατηγοριοποιήσει τα δεδομένα εισόδου. Αυτά τα δίκτυα δεν χρησιμοποιούν καμία εξωτερική επιρροή για να ρυθμίσουν τα βάρη τους αλλά ελέγχουν εσωτερικά την απόδοσή τους, ψάχνουν τις κανονικότητες ή τις τάσεις στα σήματα εισαγωγής, και κάνουν τις προσαρμογές σύμφωνα με τη λειτουργία του δικτύου. Οι περισσότεροι από αυτούς του αλγόριθμους εκτελούν ένα είδος ταξινόμησης, δηλαδή μαθαίνουν να κατηγοριοποιούν τα πρότυπα εισόδου σε ένα πεπερασμένο αριθμό κλάσεων. Ο τρόπος αυτός δεν συναντάται τόσο συχνά όσο η εκπαίδευση με επίβλεψη και δεν είναι απόλυτα κατανοητός, αλλά είναι πολύ χρήσιμος σε ορισμένες

καταστάσεις που δεν υπάρχουν δεδομένα στο πρόβλημα.

3. Στη βαθμολογημένη μάθηση αντί να παρέχεται η σωστή έξοδος για κάθε είσοδο δίνεται μόνο ένας βαθμός. Ο βαθμός αυτός είναι ένα μέτρο της απόδοσης του δικτύου μετά από μία ακολουθία δεδομένων εισόδου.

Σε όλες τις περιπτώσεις όταν το δίκτυο σταματάει να αλλάζει τις τιμές των βαρών, τότε θεωρούμε ότι η εκπαίδευση έχει επιτευχθεί. Αυτό συμβαίνει επειδή το σφάλμα στην έξοδο γίνεται μηδέν ή είναι πολύ κοντά (τείνει) στο μηδέν. Σε ορισμένες περιπτώσεις όμως η εκπαίδευση του νευρωνικού δικτύου δεν είναι δυνατή. Αυτό πιθανόν συμβαίνει επειδή τα δεδομένα εισόδου δεν περιέχουν ακριβή πληροφορία από την οποία προκύπτουν οι επιθυμητοί στόχοι. Ένα δίκτυο επίσης μπορεί να μη συγκλίνει αν δεν υπάρχουν αρκετά δεδομένα που να επιτρέπουν την πλήρη εκπαίδευση.

Με τη μάθηση με επίβλεψη το δίκτυο πρέπει να εκπαιδευτεί προτού χρησιμοποιηθεί. Τα δεδομένα εκπαίδευσης πρέπει να είναι αρκετά για να περιέχουν όλες τις αναγκαίες πληροφορίες ώστε το δίκτυο να μπορεί να μάθει τα χαρακτηριστικά γνωρίσματα και τις σχέσεις που είναι σημαντικά. Αφού εκπαιδευτεί καλά ένα δίκτυο με τα στοιχεία εκπαίδευσης, κατόπιν είναι σημαντικό να δούμε τι μπορεί να κάνει με τα στοιχεία που δεν έχει δει πριν. Εάν ένα σύστημα δεν δίνει τα λογικά αποτελέσματα για αυτό το σύνολο δοκιμής, η περίοδος εκπαίδευσης δεν τελειώνει. Αυτή η δοκιμή είναι κρίσιμη για να εξασφαλίσει ότι το δίκτυο δεν έχει απομνημονεύσει απλά ένα δεδομένο σύνολο στοιχείων αλλά έχει μάθει τα γενικά πρότυπα που περιλαμβάνονται μέσα σε μια εφαρμογή.

Ο συνηθέστερος τρόπος χρήσης των δεδομένων εκπαίδευσης είναι σε κύκλους εκπαίδευσης. Στη διάρκεια κάθε κύκλου το δίκτυο δέχεται ως είσοδο, ένα ένα, όλα τα διανύσματα εκπαίδευσης, αθροίζει τη μεταβολή στην τιμή των βαρών που προκύπτει από κάθε διάνυσμα και αναπροσαρμόζει τα βάρη στο τέλος κάθε κύκλου, χρησιμοποιώντας τη συσσωρευμένη μεταβολή. Στην περίπτωση αυτή έχουμε εκπαίδευση ανά ομάδα προτύπων εισόδου. Η μέθοδος αυτή είναι γνωστή με το όνομα μάθηση δέσμης (batch training). Εναλλακτικά, η προσαρμογή των βαρών μπορεί να γίνεται μετά τη χρήση ενός από τα διανύσματα εκπαίδευσης οπότε μιλάμε για εκπαίδευση ανά πρότυπο εισόδου ή επαυξητική μάθηση (online training ή incremental training). Οι κύκλοι εκπαίδευσης ονομάζονται εποχές (epochs) αν και ο όρος αυτός είναι περισσότερο συνδεδεμένος με τη μάθηση δέσμης η οποία δίνει γρηγορότερα αποτελέσματα αλλά έχει μεγαλύτερες απαιτήσεις μνήμης.

Η έρευνα στους διαφορετικούς τρόπους εκπαίδευσης των Τεχνητών Νευρωνικών Δικτύων συνεχίζεται καθώς νέες ιδέες παρουσιάζονται συνεχώς. Εδώ παρουσιάζουμε συνοπτικά μερικούς από τους πιο σημαντικούς κανόνες μάθησης οι οποίοι αναλύονται περισσότερο σε επόμενα κεφάλαια της εργασίας.

Ο πρώτος, και αναμφισβήτητα πιο γνωστός κανόνας εκμάθησης εισήχθη από τον Donald Hebb [17]. Η περιγραφή εμφανίστηκε στο βιβλίο του 'Η οργάνωση της συμπεριφοράς' το 1949. Ο βασικός κανόνας του λέει ότι εάν ένας νευρώνας λαμβάνει ένα σήμα εισόδου από έναν άλλο νευρώνα, και εάν και οι δύο είναι ιδιαίτερα ενεργοί (από μαθηματική άποψη έχουν το ίδιο πρόσημο), το βάρος μεταξύ των νευρώνων πρέπει να ενισχυθεί.

Παρόμοιος με τον κανόνα Hebb είναι ο κανόνας του Hopfield [20] με εξαίρεση ότι διευκρινίζει το μέγεθος της ενίσχυσης ή της αποδυνάμωσης. Σύμφωνα με τον κανόνα αυτό εάν η επιθυμητή έξοδος και είσοδος του νευρώνα είναι και οι δύο ενεργές ή και οι δύο ανενεργές, αυξάνεται το συναπτικό βάρος σύμφωνα με το ποσοστό εκμάθησης, διαφορετικά μειώνεται το βάρος σύμφωνα με το ποσοστό εκμάθησης.

Ένας άλλος κανόνας που αποτελεί περαιτέρω παραλλαγή του κανόνα Hebb είναι ο κανόνας Δέλτα. Αυτός ο κανόνας είναι βασισμένος στην ιδέα της συνεχούς τροποποίησης των βαρών του δικτύου έτσι ώστε να μειωθεί η διαφορά (το δέλτα) μεταξύ της επιθυμητής εξόδου και της πραγματικής. Αυτός ο κανόνας αλλάζει τα συναπτικά βάρη με τρόπο ώστε να ελαχιστοποιεί το μέσο τετραγωνικό σφάλμα του δικτύου. Αυτός ο κανόνας είναι επίσης γνωστός ως κανόνας εκμάθησης Widrow-Hoff και ο αλγόριθμος που τον υλοποιεί ονομάζεται και LMS.

Επίσης γνωστός είναι ο κανόνας της πιο απότομης καθόδου ο οποίος είναι παρόμοιος με τον Δέλτα κανόνα δεδομένου ότι η παράγωγος της συνάρτησης μεταφοράς χρησιμοποιείται κι εδώ για να τροποποιήσει το σφάλμα προτού να εφαρμοστεί στα βάρη σύνδεσης. Εδώ όμως μια πρόσθετη σταθερά υπάρχει στον παράγοντα που τροποποιεί το βάρος. Αυτός ο κανόνας χρησιμοποιείται συχνά, ακόμα κι αν συγκλίνει σε ένα σημείο σταθερότητας πολύ αργά. Έχει αποδειχθεί ότι τα διαφορετικά ποσοστά εκμάθησης για τα διαφορετικά στρώματα ενός δικτύου βοηθούν τη διαδικασία εκμάθησης να συγκλίνει γρηγορότερα. Σε αυτές τις δοκιμές, τα ποσοστά εκμάθησης για εκείνα τα στρώματα που βρίσκονταν κοντά στην έξοδο τέθηκαν χαμηλότερα από εκείνα τα στρώματα κοντά στην είσοδο. Αυτό είναι ιδιαίτερα σημαντικό για τις εφαρμογές όπου το δεδομένο εισόδου δεν προκύπτει από ένα ισχυρό πρότυπο.

Τέλος υπάρχει ο κανόνας εκμάθησης Kohonen [21]. Αυτός ο νόμος, αναπτύχθηκε από τον Teuvo Kohonen, και είναι εμπνευσμένος από τη μάθηση στα βιολογικά συστήματα. Σε αυτήν την διαδικασία, τα στοιχεία επεξεργασίας του δικτύου ανταγωνίζονται για να μάθουν, ή να ενημερώσουν τα βάρη τους. Το στοιχείο επεξεργασίας με τη μεγαλύτερη έξοδο είναι ο νικητής και έχει την ικανότητα να παρεμποδίζει τους ανταγωνιστές του καθώς επίσης και να διεγείρει τους γείτονές του. Μόνο ο νικητής παράγει έξοδο και μόνο αυτός και οι γείτονές του επιτρέπεται για να ρυθμίσουν τα βάρη σύνδεσής τους.

2.5 Ιδιότητες των Νευρωνικών Δικτύων

Τα νευρωνικά δίκτυα με την δυνατότητα τους να εξάγουν έννοιες και αποτελέσματα από περίπλοκα ή ανακριβή στοιχεία μπορούν να χρησιμοποιηθούν για να αναγνωρίσουν πρότυπα και να ανιχνεύσουν τάσεις που είναι σύνθετες για να παρατηρηθούν από άλλα υπολογιστικά συστήματα.

Τα Τεχνητά Νευρωνικά Δίκτυα έχουν την ικανότητα να μαθαίνουν μέσω παραδειγμάτων. Αν και δεν είναι τα μόνα συστήματα με αυτήν την δυνατότητα μάθησης εντούτοις διακρίνονται για την ικανότητά τους να οργανώνουν την πληροφορία των δεδομένων εισόδου σε χρήσιμες μορφές. Αυτές οι μορφές αποτελούν στην ουσία ένα μοντέλο που αναπαριστά τη σχέση που ισχύει μεταξύ των δεδομένων εισόδου και εξόδου.

Επίσης μπορούν να θεωρηθούν ως κατανομημένη μνήμη και ως μνήμη συσχέτισης.

Ο χαρακτηρισμός τους ως κατανεμημένη μνήμη πηγάζει από το ότι η κωδικοποίηση που δημιουργούν είναι κατανεμημένη σε όλα τα βάρη της συνδεσμολογίας τους. Για τον ίδιο λόγο τα Τεχνητά Νευρωνικά Δίκτυα χαρακτηρίζονται και ως μνήμες συσχέτισης. Μια μνήμη συσχέτισης δεν αποθηκεύει πληροφορία με τον παραδοσιακό τρόπο αλλά μέσω κατάλληλων συσχετίσεων που δημιουργεί από τα δεδομένα εκπαίδευσης. Η ανάκληση της πληροφορίας γίνεται με βάση το περιεχόμενο και όχι τη διεύθυνση, όπως δηλαδή συμβαίνει και με τον ανθρώπινο εγκέφαλο. Η παραπάνω οργάνωση κάνει ορισμένα είδη νευρωνικών δικτύων να είναι πολύ ανεκτικά σε μικρές αλλαγές στα σήματα εισόδου, δηλαδή είναι σε θέση να παράγουν τη σωστή έξοδο ακόμη κι αν τα δεδομένα εισόδου είναι λίγο διαφορετικά ή ελλιπή, για παράδειγμα λόγω θορύβου.

Οι υπολογισμοί ενός νευρωνικού δικτύου μπορούν να εκτελεστούν παράλληλα, ενώ συγχρόνως παρουσιάζουν ανοχή στην διαστρέβλωση εικόνων-προτύπων. Επίσης έχουν δυνατότητα γενίκευσης.

Τα Τεχνητά Νευρωνικά Δίκτυα όπως και τα βιολογικά έχουν μεγάλη ανοχή σε δομικά σφάλματα. Αυτό σημαίνει ότι η κακή λειτουργία ή η καταστροφή ενός νευρώνα ή κάποιων συνδέσμων δεν είναι ικανή να διαταράξει σημαντικά τη λειτουργία τους καθώς η πληροφορία που εσωκλείουν δεν είναι εντοπισμένη σε συγκεκριμένο σημείο αλλά διάχυτη σε όλο το δίκτυο. Γενικά, το μέγεθος του σφάλματος λόγω δομικών λαθών είναι ανάλογο του ποσοστού των κατεστραμμένων συνδέσεων. Από τη στιγμή που ένα Τεχνητό Νευρωνικό Δίκτυο εκπαιδεύει στο να αναγνωρίζει συνθήκες και καταστάσεις, απαιτείται ένας μόνο κύκλος λειτουργίας τους για να προσδιορίσουν μια συγκεκριμένη κατάσταση.

Το μειονέκτημα των Τεχνητών Νευρωνικών Δικτύων είναι ότι η χρήση τους δεν ενδείκνυται σε περιπτώσεις, στις οποίες υπάρχουν αποδοτικές μαθηματικές ή αλγοριθμικές μέθοδοι επίλυσης, μιας και ο χρόνος που απαιτείται για το σχεδιασμό και την εκπαίδευση ενός ΤΝΔ είναι πιθανά αυξημένος. Επίσης, υπάρχει δυσκολία εξήγησης της αιτίας για την οποία το ΤΝΔ φτάνει σε μία συγκεκριμένη λύση και όχι σε κάποια άλλη.

2.6 Ιστορική Αναδρομή

Η ιστορία των Τεχνητών Νευρωνικών Δικτύων καλύπτεται από την προσπάθεια επιστημόνων που προέρχονται από πολλά διαφορετικά επιστημονικά πεδία.

Στο τέλος του 19ου και αρχή του 20ου αιώνα υπήρξαν διάφορες μελέτες στη φυσική, τη φυσιολογία και τη νευροφυσιολογία από επιστήμονες όπως ο Hermann von Helmholtz, ο Ernst Mach και ο Ivan Pavlov. Οι μελέτες αυτές έδιναν έμφαση κυρίως σε γενικές θεωρίες μάθησης, όρασης και συνθηκών και δεν περιελάμβαναν συγκεκριμένα μαθηματικά μοντέλα για τη λειτουργία των νευρωνικών δικτύων.

Το 1943 οι Warren McCulloch και Walter Pitts [26] παρουσίασαν ένα από τα πρώτα τεχνητά νευρωνικά δίκτυα. Το βασικό χαρακτηριστικό αυτού του μοντέλου είναι ότι ένα βεβαρημένο άθροισμα σημάτων εισόδου συγκρίνεται με ένα κατώφλι για τον καθορισμό της εξόδου του νευρωνικού. Όταν το άθροισμα είναι μεγαλύτερο ή ίσο με το κατώφλι τότε η έξοδος είναι ίση με 1. Αντίθετα, όταν το άθροισμα είναι μικρότερο τότε η έξοδος είναι 0.

Αυτό που προσπάθησαν ήταν να δείξουν ότι τα δίκτυα αυτών των νευρώνων μπορούσαν να υπολογίσουν οποιαδήποτε αριθμητική ή λογική συνάρτηση. Σε αντίθεση με τα βιολογικά δίκτυα, οι παράμετροι αυτών των δικτύων έπρεπε να σχεδιαστούν καθώς καμία μέθοδος εκπαίδευσης δεν ήταν διαθέσιμη. Παρόλα αυτά ο συνδυασμός ανάμεσα στη βιολογία και τους ψηφιακούς υπολογιστές δημιούργησε ένα μεγάλο ενδιαφέρον.

Τους Warren McCulloch και Walter Pitts ακολούθησε ο Donald Hebb [17] ο οποίος το 1949 υποστήριξε ότι η συμπεριφορά του ανθρώπου μπορεί να εξηγηθεί από τη συμπεριφορά των νευρώνων. Η θέση του ήταν ότι όταν ο άξονας ενός νευρώνα A είναι αρκετά κοντά ώστε να διεγείρει το νευρώνα B και συστηματικά συμμετέχει στην ενεργοποίησή του, τότε κάποια μεταβολική αλλαγή συμβαίνει είτε στο ένα απ'τα δύο είτε και στα δύο κύτταρα έτσι ώστε η αποτελεσματικότητα με την οποία ο A διεγείρει τον B αυξάνεται. Η θέση αυτή πρότεινε έναν φυσικό μηχανισμό μάθησης των βιολογικών νευρώνων και ονομάστηκε κανόνας Hebbian.

Στα τέλη του 1950 ο Frank Rosenblatt [31] και μερικοί ακόμη ερευνητές ανέπτυξαν μία κλάση νευρωνικών δικτύων που ονομάζονται Perceptrons. Οι νευρώνες αυτών των δικτύων ήταν παρόμοιοι με αυτούς των McCulloch και Pitts. Η συνεισφορά του Rosenblatt όμως αφορά στην ανάπτυξη ενός κανόνα εκμάθησης για την εκπαίδευση δικτύων Perceptron για τη λύση προβλημάτων αναγνώρισης προτύπων. Απέδειξε ότι ο κανόνας του θα συγκλίνει πάντα στις σωστές τιμές βαρών για το δίκτυο, αν υπάρχουν τέτοιες τιμές που να δίνουν λύση στο πρόβλημα. Η εκμάθηση ήταν απλή και αυτόματη, χρησιμοποιώντας παραδείγματα σωστής συμπεριφοράς τα οποία παρουσιάζονταν στο δίκτυο το οποίο μάθαινε από τα λάθη του. Το δίκτυο Perceptron μπορούσε ακόμη να μάθει όταν γινόταν η αρχικοποίηση με τυχαίες τιμές για τα βάρη και τις πολώσεις. Σήμερα τα δίκτυα Perceptron θεωρούνται ακόμη σημαντικά. Παραμένουν μία γρήγορη και αξιόπιστη μέθοδος για την κλάση των προβλημάτων που μπορούν να λύσουν.

Την ίδια περίπου χρονική περίοδο οι Bernard Widrow και Ted Hoff [39] παρουσίασαν έναν καινούργιο αλγόριθμο εκπαίδευσης και τον χρησιμοποίησαν για να εκπαιδεύσουν γραμμικά Τεχνητά Νευρωνικά Δίκτυα τα οποία είναι παρόμοια σε δομή και ικανότητες με το δίκτυο Perceptron του Rosenblatt και ονομάστηκαν Adaline και Madaline. Το δίκτυο Madaline ήταν το πρώτο Τεχνητό Νευρωνικό Δίκτυο που χρησιμοποιήθηκε σε πραγματικό πρόβλημα. Πρόκειται για ένα προσαρμοστικό φίλτρο το οποίο χρησιμοποιείται για να μειώσει την ηχώ στις τηλεφωνικές γραμμές. Ο αλγόριθμος αυτός χρησιμοποιείται μέχρι σήμερα.

Τα δίκτυα Perceptron χρησιμοποιήθηκαν με επιτυχία σε αρκετές εφαρμογές ταξινόμησης, όμως θεωρητικές και πειραματικές έρευνες κατέληξαν στο συμπέρασμα ότι παρουσιάζουν σημαντικά προβλήματα σε περιπτώσεις όπου οι κλάσεις δεν είναι γραμμικά διαχωρίσιμες. Τους περιορισμούς αυτούς γνώριζαν ο Rosenblatt και ο Widrow και για το λόγο αυτό πρότειναν νέα πιο πολύπλοκα πολυστρωματικά δίκτυα. Πιο συγκεκριμένα για την αντιμετώπιση του συγκεκριμένου ζητήματος προτάθηκε η ιδέα της χρησιμοποίησης αλληπάλληλων νευρωνικών επιπέδων Perceptron. Η βασική ιδέα είναι η διασύνδεση πολλαπλών επιπέδων τύπου Perceptron διαδοχικά, έτσι ώστε η έξοδος του ενός επιπέδου να αποτελεί είσοδο για το επόμενο επίπεδο, δεν κατάφεραν όμως να τροποποιήσουν τους αλγόριθμους εκπαίδευσης έτσι ώστε να εκπαιδεύσουν επιτυχώς τα δίκτυα αυτά. Οι περιορισμοί αυτοί δημοσιεύτηκαν

στο βιβλίο των Marvin Minsky και Seymour Papert [27] και επηρέασαν πολλούς ερευνητές οι οποίοι πίστευαν ότι δεν ήταν δυνατή η περαιτέρω έρευνα για να ξεπεραστούν τα προβλήματα αυτά. Έτσι για περίπου μια δεκαετία δεν υπήρξε καμία σημαντική πρόοδος στην έρευνα των νευρωνικών δικτύων.

Το 1972 ο Teuvo Kohonen [21] και ο James Anderson [1] ανεξάρτητα ο ένας από τον άλλο ανέπτυξαν ένα καινούργιο τύπο νευρωνικών δικτύων τα οποία μπορούσαν να συμπεριφέρονται σαν μνήμες. Την ίδια χρονική περίοδο ο Stephen Grossberg [14] ανέπτυξε μη γραμμικά συνεχούς χρόνου συσχετιζόμενα δίκτυα.

Στη συνέχεια ακολουθεί μια περίοδος αναγέννησης των νευρωνικών δικτύων. Εμφανίζονται νέες ιδέες και σε συνδυασμό με την ανάπτυξη των σύγχρονων υπολογιστών η έρευνα των Τεχνητών Νευρωνικών Δικτύων αναπτύσσεται δραματικά γρήγορα. Η αναγέννηση αυτή στις αρχές του 1980 οφείλεται σε μεγάλο βαθμό στη συμβολή του φυσικού John Hopfield [20]. Ο Hopfield συνδύασε τη χρήση της στατιστικής μηχανικής με τη λειτουργία συγκεκριμένων κλάσεων αναδρομικών δικτύων τα οποία θα μπορούσαν να χρησιμοποιηθούν σαν συσχετιζόμενη μνήμη.

Κλειδί στην ανάπτυξη των νευρωνικών δικτύων υπήρξε η ανακάλυψη του αλγορίθμου της όπισθεν διάδοσης σφάλματος (Backpropagation) για την εκπαίδευση πολυστρωματικών δικτύων Perceptron. Η πρώτη περιγραφή ενός αλγορίθμου για την εκπαίδευση πολυστρωματικών δικτύων έγινε από τον Paul Werbos το 1974. Η θέση του όμως δεν αναγνωρίστηκε από την κοινότητα που ασχολούταν με τα νευρωνικά δίκτυα. Έτσι είχαμε την επάνοδο της ανακάλυψης του αλγορίθμου Backpropagation στα μέσα του 1980 από πολλούς ερευνητές ανεξάρτητα, τους David Rumelhart, Geoffrey Hinton, Ronald Williams, David Parker και Yann Le Cun. Η εργασία που επηρέασε περισσότερο ήταν αυτή των David Rumelhart και James McClelland [32]. Ο αλγόριθμος αυτός ήταν απάντηση στις κριτικές των Minsky και Papert και άνοιξε νέους δρόμους στην έρευνα των Τεχνητών Νευρωνικών Δικτύων.

Από το 1985 και μετά αρχίζουν τα πρώτα συνέδρια που είναι αφιερωμένα αποκλειστικά σε νευρωνικά δίκτυα, από την American Physical Society και από την IEEE (Institute of Electrical and Electronics Engineers). Παρακολουθούνται από περισσότερους από χίλιους συνέδρους. Ταυτόχρονα δημιουργούνται ειδικές επαγγελματικές εταιρίες νευρωνικών δικτύων με χιλιάδες μέλη, όπως η International Neural Network Society με τρεις πόλους: Αμερική (με διευθυντή τον Grossberg), Ευρώπη (Kohonen) και Ιαπωνία (Amari).

Τα τελευταία χρόνια έχουν γίνει πολλές μελέτες και τα Τεχνητά Νευρωνικά Δίκτυα έχουν βρει πολλές εφαρμογές. Αν και είναι δύσκολο να προβλέψουμε το μέλλον της επιτυχίας των νευρωνικών δικτύων, ο μεγάλος αριθμός και η ποικιλία των εφαρμογών αυτής της νέας τεχνολογίας δίνει πολλές υποσχέσεις.

2.7 Διαφορές των Τεχνητών Νευρωνικών Δικτύων από άλλα συστήματα Τεχνητής Νοημοσύνης

Η αρχιτεκτονική των νευρωνικών δικτύων είναι πολύ διαφορετική από αυτήν των παραδοσιακών υπολογιστών που περιέχουν έναν επεξεργαστή. Οι γνωστοί υπολογιστές δουλεύουν

σειριακά, σύμφωνα με τις πρώτες ιδέες του von Neumann¹, και έχουν την ικανότητα να επιτελούν μερικές εκατοντάδες εντολών που είναι πολύ γνωστές, όπως είναι οι αριθμητικές πράξεις. Στην διαδικασία εκτέλεσης των εντολών ακολουθούν πιστά ένα εσωτερικό ρολόι [2].

Από τη φύση τους τα νευρωνικά δίκτυα δεν λειτουργούν σειριακά, αλλά με τρόπο που μοιάζει πιο πολύ σε παράλληλο τρόπο λειτουργίας, διότι μία εργασία μοιράζεται στα διάφορα τμήματα του δικτύου, μοιράζεται σε όλους τους επί μέρους νευρώνες. Έτσι λέμε ότι τα νευρωνικά δίκτυα είναι συστήματα «παράλληλων κατανεμημένων διεργασιών» (parallel distributed processing). Αυτό μας παρέχει μεγάλες ταχύτητες, διότι είναι σαν να έχουμε ταυτόχρονα πολλούς επεξεργαστές στη διάθεσή μας.

Τα νευρωνικά δίκτυα προσφέρουν έναν διαφορετικό τρόπο ανάλυσης των δεδομένων και αναγνώρισης προτύπων από τις παραδοσιακές μεθόδους υπολογισμού. Παρόλα αυτά, δεν αποτελούν λύση για όλα τα υπολογιστικά προβλήματα. Με τις διάφορες γλώσσες και μεθόδους προγραμματισμού μπορούν να επιλυθούν προβλήματα τα οποία είναι καλά ορισμένα. Για παράδειγμα η επεξεργασία λογιστικών δεδομένων, οι υπηρεσίες καταλόγου ή η προστασία εξοπλισμού αποτελούν καλά ορισμένα προβλήματα και δεν απαιτούν τα πρόσθετα χαρακτηριστικά των νευρικών δικτύων.

Όμως η αρχιτεκτονική των νευρωνικών δικτύων διαφέρει από αυτήν των παράλληλων επεξεργασιών, για το λόγο ότι οι απλοί επεξεργαστές των νευρωνικών δικτύων (δηλαδή οι νευρώνες) έχουν μεγάλο αριθμό διασυνδέσεων, ο οποίος συνολικά είναι πολύ μεγαλύτερος από τον αριθμό των νευρώνων. Και αυτό βέβαια γιατί κάθε νευρώνας έχει πολλές συνδέσεις. Αντίθετα, στους παράλληλους υπολογιστές, οι επεξεργαστές είναι συνήθως περισσότεροι από τις διασυνδέσεις μεταξύ τους και ως προς την πολυπλοκότητα τους ακολουθούν την μηχανή von Neumann.

Τα νευρωνικά δίκτυα διαφέρουν από αυτό, διότι οι μονάδες τους είναι πολύ πιο απλές και επιτελούν πολύ απλούστερες λειτουργίες, δηλαδή ξέρουν μόνο να αθροίζουν τα σήματα εισόδου και να τροποποιούν τα βάρη των διασυνδέσεων. Επίσης, οι νευρώνες λειτουργούν ανεξάρτητα ο ένας από τον άλλο και δεν χρειάζονται συγχρονισμό. Αυτό δίνει στα νευρωνικά δίκτυα την ευρωστία και ανοχή σε σφάλματα. Ο Πίνακας 2.2 προσδιορίζει τις βασικές διαφορές μεταξύ αυτών των δύο προσεγγίσεων υπολογισμού.

Επίσης οι πληροφορίες που αποθηκεύονται σε ένα νευρωνικό δίκτυο μοιράζονται σε ένα μεγάλο αριθμό μονάδων, δηλαδή σε πολλούς νευρώνες. Αντίθετα, όταν αποθηκεύουμε στοιχεία στην μνήμη του υπολογιστή, κάθε πληροφορία σε δυαδική μορφή τοποθετείται σε μια συγκεκριμένη τοποθεσία.

¹Η von Neumann αρχιτεκτονική υπολογιστών προτάθηκε από τον John von Neumann ο οποίος ήταν ο πρώτος που κατέγραψε τις απαιτήσεις στην δομή ενός ηλεκτρονικού υπολογιστή. Πρόκειται για την βασική αρχιτεκτονική δομή των υπολογιστών που χρησιμοποιούμε σήμερα και η οποία συνίσταται από 4 βασικά μέρη. Την μονάδα αριθμητικής λογικής, την μονάδα ελέγχου, την μνήμη και τις μονάδες εισόδου-εξόδου. Οι ικανότητες ενός συστήματος που ακολουθεί αυτή την αρχιτεκτονική περιορίζονται στην αποθήκευση των δεδομένων και των αποτελεσμάτων που προκύπτουν μετά από υπολογισμούς και την αποθήκευση οδηγιών και εντολών για την εκτέλεση των υπολογισμών.

Πίνακας 2.2: Διαφορές των Τεχνητών Νευρωνικών Δικτύων με τις άλλες υπολογιστικές τεχνικές

Χαρακτηριστικά	Υπολογιστικά Συστήματα (συμπεριλαμβάνονται τα Έμπειρα Συστήματα)	Τεχνητά Νευρωνικά Δίκτυα
Τρόπος επεξεργασίας των δεδομένων	Συνεχής	Παράλληλος
Λειτουργίες	Λογικά μέσω Κανόνων, Υπολογισμών, Εννοιών	Μέσω εικόνων
Τρόπος λειτουργίας	Ασύγχρονος	Σύγχρονος
Μέθοδοι Μηχανικής Μάθησης	Με κανόνες	Με παραδείγματα αλλάζοντας τα βάρη των συνδέσεων τους
Αποθήκευση πληροφορίας	Σε συγκεκριμένες διευθύνσεις της μνήμης	Στα βάρη των συνδέσεων
Εφαρμογές	Λογιστική, Επεξεργασία δεδομένων, Μαθηματικά, Ψηφιακές επικοινωνίες, κτλ.	Επεξεργασία με αισθητήρες, Αναγνώριση προτύπων, Προσέγγιση συναρτήσεων, κτλ.

Μια άλλη νέα ιδιότητα στα νευρωνικά δίκτυα είναι αυτή της ανοχής σφάλματος. Αυτό σημαίνει ότι αν ένα μικρό τμήμα του δικτύου χαλάσει, το υπόλοιπο δίκτυο συνεχίζει να λειτουργεί, έστω και με ένα μικρό σφάλμα. Διαφορετικά σημαίνει ότι, αν τα δεδομένα ενός προβλήματος σε ένα μικρό μέρος τους είναι εσφαλμένα, το δίκτυο δίδει τη σωστή απάντηση και πάλι όμως με ένα μικρό σφάλμα κάτι που δε συμβαίνει με τα άλλα υπολογιστικά συστήματα.

Τα έμπειρα συστήματα αποτελούν επέκταση των παραδοσιακών υπολογιστών και συχνά αποκαλούνται ως η πέμπτη γενιά των υπολογιστικών συστημάτων. Στην πρώτη γενιά ανήκουν οι υπολογιστές που λειτουργούσαν χρησιμοποιώντας διακόπτες και καλώδια. Η δεύτερη γενιά εμφανίστηκε λόγω της ανάπτυξης της κρυσταλλολυχνίας (τρανζίστορ). Η τρίτη γενιά περιελάμβανε τη χρήση των ολοκληρωμένων κυκλωμάτων και των υψηλότερου επιπέδου γλωσσών προγραμματισμού όπως COBOL, FORTRAN, και C. Τέλος, τα εργαλεία που βοηθούν το χρήστη στην ανάπτυξη εφαρμογών, γνωστά ως «γεννήτριες κώδικα» αποτελούν την τέταρτη γενιά υπολογιστικών συστημάτων. Η πέμπτη γενιά περιλαμβάνει την τεχνητή νοημοσύνη.

Τυπικά, ένα Έμπειρο Σύστημα αποτελείται από δύο μέρη, μια μηχανή συμπεράσματος και μια βάση γνώσης. Η μηχανή συμπεράσματος είναι γενική και χειρίζεται τη διεπαφή με τον χρήστη, τα εξωτερικά αρχεία, την πρόσβαση του προγράμματος, και το σχεδιασμό. Η βάση γνώσης περιέχει όλες τις πληροφορίες που είναι απαραίτητες για τη λύση ενός συγκεκριμένου προβλήματος. Αυτή η βάση γνώσης επιτρέπει σε έναν εμπειρογνώμονα να

καθορίσει τους κανόνες με τους οποίους θα εκτελεστεί μια διαδικασία. Αυτός ο εμπειρογνώμονας δεν είναι απαραίτητο να καταλαβαίνει προγραμματισμό. Εκείνο που πρέπει είναι απλά να καθορίσει τι θέλει ο υπολογιστής να κάνει και πώς ο μηχανισμός του έμπειρου συστήματος λειτουργεί.

Καθώς η πολυπλοκότητα του συστήματος αυξάνεται, το σύστημα απαιτεί περισσότερους υπολογιστικούς πόρους οπότε γίνεται πάρα πολύ αργό. Γι' αυτό το λόγο η εφαρμογή έμπειρων συστημάτων είναι εφικτή μόνο όταν περιορίζονται λεπτομερώς.

Τα τεχνητά νευρωνικά δίκτυα προσφέρουν μια διαφορετική προσέγγιση στην επίλυση των προβλημάτων. Προσπαθούν να παρέχουν ένα εργαλείο το οποίο μπορεί να προγραμματίζεται αυτόματα και να μαθαίνει μόνο του από τα δεδομένα. Τα νευρωνικά δίκτυα είναι δομημένα για να παρέχουν την ικανότητα να λύνουν προβλήματα χωρίς τα οφέλη ενός εμπειρογνώμονα και χωρίς την ανάγκη προγραμματισμού.

Τα έμπειρα συστήματα έχουν σημαντικές επιτυχίες σε πολλές εφαρμογές. Εντούτοις, η τεχνητή νοημοσύνη έχει αντιμετωπίσει προβλήματα σε πεδία όπως η όραση, η συνεχής αναγνώριση φωνής και η σύνθεση και η εκμάθηση μηχανών. Διάφορα πολύ περίπλοκα προβλήματα δεν μπορούν να λυθούν με τα παραδοσιακά υπολογιστικά περιβάλλοντα. Παραδείγματος χάριν, η ομιλία είναι κάτι που όλοι οι άνθρωποι μπορούν εύκολα να αναλύσουν και να καταλάβουν. Ένα πρόσωπο μπορεί να καταλάβει μια νότια προφορά ή λέξεις ενός μωρού. Χωρίς τη μαζικά παράλληλη δύναμη επεξεργασίας ενός νευρωνικού δικτύου, αυτή η διαδικασία είναι ουσιαστικά αδύνατη για έναν υπολογιστή. Η αναγνώριση εικόνας είναι ένας άλλος στόχος που ένας άνθρωπος μπορεί εύκολα να κάνει αλλά που δυσκολεύεται ακόμα και ο μεγαλύτερος υπολογιστής. Ένα πρόσωπο μπορεί να αναγνωρίσει ένα αεροπλάνο καθώς γυρίζει, πετά από πάνω, και εξαφανίζεται σε ένα σημείο. Ένας παραδοσιακός υπολογιστής από την άλλη πλευρά μπορεί να προσπαθήσει να συγκρίνει τις μεταβαλλόμενες αυτές εικόνες με διάφορα πολύ διαφορετικά αποθηκευμένα σχέδια κι έτσι να καταλήξει σε λάθος συμπέρασμα.

Όμως, παρά τα πλεονεκτήματα των νευρωνικών δικτύων σε σύγκριση με τα έμπειρα συστήματα και τον παραδοσιακό τρόπο προγραμματισμού, τα νευρωνικά δίκτυα δεν είναι πλήρεις λύσεις. Ακόμα και όταν έχει αναπτυχθεί ένα δίκτυο, δεν υπάρχει κανένας τρόπος να εξασφαλιστεί ότι το δίκτυο αυτό είναι το βέλτιστο. Θα αναφέρουμε μερικές μεθοδολογίες για το βέλτιστο σχεδιασμό νευρωνικών δικτύων σε επόμενη παράγραφο.

Κεφάλαιο 3

Το δίκτυο Perceptron

Το μοντέλο Perceptron είναι από τα πρώτα μοντέλα νευρωνικών δικτύων που αναπτύχθηκαν την δεκαετία του πενήντα και έδωσαν στην περιοχή αυτή μεγάλη ώθηση χάριν στις επιτυχίες που είχε από την αρχή. Πολλά δίκτυα που αναπτύχθηκαν αργότερα, κατά πολύ πιο περίπλοκα, ξεκίνησαν από τη βάση του Perceptron. Προτάθηκε το 1958 από τον Rosenblatt, ο οποίος ήταν ψυχολόγος [31]. Βέβαια καθώς οι γνώσεις μας για το νευρικό σύστημα του ανθρώπου προόδευαν, οι πρώτες αυτές προσπάθειες φαίνονται τώρα πια ότι ήταν πολύ απλοϊκές. Οι Minsky και Papert έδειξαν το 1969 ότι το πρώτο αυτό πρότυπο έχει πολλούς περιορισμούς. Σήμερα, υπάρχουν πολλές παραλλαγές νευρωνικών δικτύων που βασίζονται στο Perceptron και έχουν διαφορετικές δομές, άλλες απλές και άλλες πιο περίπλοκες. Η πιο απλή μορφή είναι το λεγόμενο στοιχειώδες Perceptron (elementary perceptron) και αποτελείται από ένα μόνο νευρώνα. Είναι το πιο απλό, αυτοδύναμο σύστημα που υπάρχει και επιτελεί μία ορισμένη διεργασία. Ανεβαίνοντας ως προς την πολυπλοκότητα, έχουμε νευρωνικά δίκτυα τα οποία έχουν πολλούς νευρώνες οι οποίοι είναι οργανωμένοι σε δύο επίπεδα, ένα επίπεδο στο οποίο εισέρχονται τα σήματα (επίπεδο εισόδου) και ένα επίπεδο όπου βγαίνει το αποτέλεσμα του νευρωνικού δικτύου (επίπεδο εξόδου) [15].

Ένα μονοστρωματικό δίκτυο Perceptron αποτελείται από έναν ή περισσότερους νευρώνες εισόδου ο καθένας από τους οποίους συνδέεται με έναν παράγοντα βάρους w_i για όλες τις εισόδους i . Το γενικό δίκτυο Perceptron φαίνεται στο Σχήμα 3.1.

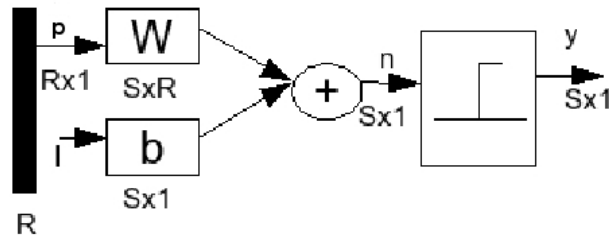
Η έξοδος του δικτύου δίνεται από τη σχέση $\mathbf{y} = f(\mathbf{W}\mathbf{p} + \mathbf{b})$. Το δίκτυο αποτελείται από τα εξής στοιχεία:

$$\text{τον πίνακα βαρών του δικτύου } \mathbf{W} = \begin{pmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,R} \\ w_{2,1} & w_{2,2} & \dots & w_{2,R} \\ \vdots & \vdots & \vdots & \vdots \\ w_{s,1} & w_{s,2} & \dots & w_{s,R} \end{pmatrix},$$

$$\text{το διάνυσμα των εισόδων } \mathbf{p} = \begin{pmatrix} p_1 & p_2 & \dots & p_R \end{pmatrix}$$

$$\text{και τη συνάρτηση μεταφοράς } f(s) = \begin{cases} 1 & \text{αν } s > 0 \\ 0 & \text{αλλιώς} \end{cases}.$$

Αν το εσωτερικό γινόμενο μιας γραμμής του πίνακα βαρών με διάνυσμα εισόδου είναι μεγαλύτερο ή ίσο με το $-b$ τότε η έξοδος του δικτύου θα είναι ίση με 1 αλλιώς η έξοδος θα



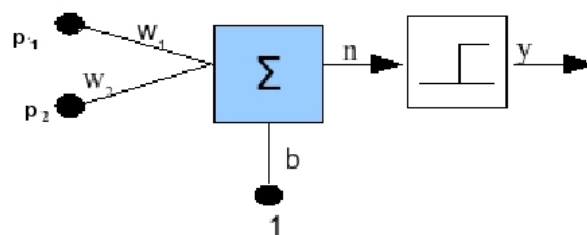
Σχήμα 3.1: Δίκτυο Perceptron

είναι 0. Με τον τρόπο αυτό κάθε νευρώνας του δικτύου χωρίζει τα δεδομένα εισόδου σε δύο κατηγορίες.

Στην πιο απλή περίπτωση το δίκτυο αποτελείται από δύο μόνο εισόδους και μια μοναδική έξοδο, επομένως από ένα μόνο νευρώνα. Η είσοδος του νευρώνα είναι το βεβαρημένο άθροισμα των εισόδων συν την πόλωση. Η έξοδος του δικτύου διαμορφώνεται από την ενεργοποίηση του νευρώνα εξόδου και η εξίσωση δίνεται από τη σχέση

$$y = f\left(\sum w_i p_i + b\right).$$

Το δίκτυο αυτό φαίνεται στο Σχήμα 3.2.



Σχήμα 3.2: Ένα απλό δίκτυο Perceptron.

Η έξοδος του δικτύου θα είναι ίση με 1 ή 0 ανάλογα με την είσοδο. Το δίκτυο μπορεί να χρησιμοποιηθεί για κατηγοριοποίηση των εισόδων σε δύο κατηγορίες. Ο διαχωρισμός μεταξύ των δύο κλάσεων στην περίπτωση αυτή είναι μια ευθεία γραμμή ή οποία δίνεται από τις τιμές της εισόδου για τις οποίες η είσοδος του δικτύου n είναι ίση με 0. Δηλαδή,

$$n = w_1 p_1 + w_2 p_2 + b = 0.$$

Το δίκτυο αυτό χρησιμοποιείται για τη λύση γραμμικά διαχωρίσιμων προβλημάτων.

Αν δώσουμε τυχαία τιμές στα βάρη και την πόλωση του δικτύου, έστω $w_1 = 1$, $w_2 = 2$ και $b = -1$, τότε η διαχωριστική γραμμή θα είναι

$$n = p_1 + p_2 - 1 = 0.$$

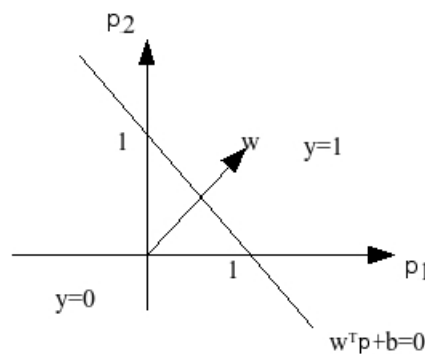
Η εξίσωση αυτή καθορίζει μια ευθεία γραμμή. Στη μία πλευρά της γραμμής η έξοδος του δικτύου θα είναι ίση με 1, ενώ στην άλλη πλευρά θα είναι ίση με 0. Για να σχεδιάσουμε την ευθεία αρκεί να βρούμε δύο σημεία αυτής. Τα σημεία αυτά μπορεί είναι η τομή της ευθείας με τους άξονες x και y , δηλαδή τα σημεία που βρίσκουμε αν θέσουμε τις εισόδους p_1 και p_2 ίσες με 0. Οπότε θα έχουμε για

$$p_1 = 0, p_2 = \frac{-b}{w_2} = -\left(\frac{-1}{1}\right) = 1$$

και για

$$p_2 = 0, p_1 = \frac{-b}{w_1} = -\left(\frac{-1}{1}\right) = 1.$$

Η ευθεία που προκύπτει φαίνεται στο Σχήμα 3.3.



Σχήμα 3.3: Διαχωριστική γραμμή για ένα δίκτυο Perceptron δύο εισόδων.

Για να βρούμε σε ποια πλευρά της ευθείας η έξοδος του δικτύου είναι ίση με 1 αρκεί να δοκιμάσουμε ένα τυχαίο σημείο. Για $p = [2 \ 0]$ η έξοδος του δικτύου θα είναι

$$y = f(wp + b) = f([1 \ 1] [2 \ 0]^T - 1) = 1.$$

Επομένως για όλες τις εισόδους που αντιστοιχούν στα σημεία δεξιά της ευθείας η έξοδος του δικτύου θα είναι ίση με 1.

Παρατηρούμε ότι το διάνυσμα των βαρών είναι πάντα κάθετο στη διαχωριστική γραμμή. Για όλα τα σημεία της διαχωριστικής γραμμής το εσωτερικό γινόμενο του διανύσματος των εισόδων με το διάνυσμα των βαρών είναι το ίδιο. Αυτό προϋποθέτει ότι αυτά τα διανύσματα εισόδων θα έχουν την ίδια προβολή στο διάνυσμα των βαρών κι έτσι θα πρέπει να βρίσκονται σε μια γραμμή ορθογώνια στο διάνυσμα βαρών. Επομένως το διάνυσμα των βαρών θα δείχνει πάντα προς την περιοχή όπου η έξοδος του νευρώνα είναι ίση με 1.

Για δίκτυα Perceptron με πολλαπλούς νευρώνες θα υπάρχει μια διαχωριστική γραμμή για κάθε νευρώνα. Ένα δίκτυο με ένα νευρώνα κατηγοριοποιεί τις εισόδους σε δύο κατηγορίες καθώς η έξοδος που παράγει το δίκτυο είναι είτε 0 ή 1. Ένα Perceptron με πολλούς νευρώνες μπορεί να ταξινομήσει τα διανύσματα εισόδου σε πολλές κατηγορίες. Κάθε κατηγορία αναπαρίσταται από ένα διαφορετικό διάνυσμα εξόδου. Αφού κάθε στοιχείο του διανύσματος εξόδου μπορεί να είναι μόνο 0 ή 1, θα υπάρχουν συνολικά το πολύ 2^S πιθανές κατηγορίες, όπου S είναι ο αριθμός των νευρώνων του δικτύου.

3.1 Κανόνες εκπαίδευσης Perceptron

Η εκπαίδευση του δικτύου συνίσταται στο να μπορέσουμε να βρούμε τις κατάλληλες τιμές των w και b και τότε το δίκτυο θα μπορεί να αναγνωρίζει τα πρότυπα στα οποία έχει εκπαιδευθεί.

Ο κανόνας εκπαίδευσης Perceptron αποτελεί ένα παράδειγμα μάθησης με επίβλεψη. Στην αρχή της μάθησης το σύστημα δεν έχει καμία προηγούμενη γνώση. Τα βάρη w_i πρέπει να έχουν τυχαίες τιμές, για παράδειγμα έχουν τιμές οι οποίες δίνονται από μία κατανομή ψευδοτυχαίων αριθμών, και είναι όλα στο διάστημα $[0, 1]$. Το δίκτυο τροφοδοτείται με ένα σύνολο προτύπων εκπαίδευσης $\{ p_1, t_1 \}, \{ p_2, t_2 \}, \dots, \{ p_n, t_n \}$, όπου p_n είναι η είσοδος που θα εφαρμοστεί στο δίκτυο και t_n είναι η επιθυμητή έξοδος ή στόχος που αντιστοιχεί στην είσοδο αυτή. Καθώς κάθε είσοδος τροφοδοτείται στο δίκτυο, η έξοδος που προκύπτει συγκρίνεται με τον επιθυμητό στόχο. Ο κανόνας εκπαίδευσης τότε προσαρμόζει τα βάρη και τις πολώσεις του δικτύου έτσι ώστε η έξοδος του δικτύου να συγκλίνει με το στόχο. Όταν παρουσιάζουμε τα πρότυπα στο νευρωνικό δίκτυο, τότε το σύστημα μαθαίνει την πληροφορία κάθε προτύπου με το να μεταβάλλει τα βάρη του προς την σωστή κατεύθυνση.

Η διαδικασία αυτή είναι ανάλογη με την εξάσκηση που υφίσταται ένα βιολογικό σύστημα όταν μαθαίνει μια διεργασία. Η μεταβολή των βαρών συνεχίζεται μέχρις ότου το σύστημα μάθει το σήμα που του δόθηκε. Όταν συμβεί αυτό τότε σταματάει η μεταβολή των w και οι τελικές τιμές τους αποθηκεύονται και χρησιμοποιούνται περαιτέρω. Στο σημείο αυτό λέμε ότι το δίκτυο έχει εκπαιδευθεί και έχει μάθει τα πρότυπα που του διδάξαμε.

Ο αλγόριθμος Perceptron αποτελείται από τα ακόλουθα βήματα :

1. Αρχικοποίησε όλα τα βάρη και τις πολώσεις του δικτύου με τυχαίες τιμές.

2. Εισήγαγε τα πρότυπα με τη σειρά. Μόλις τελειώσουν ξανάρχισε από την αρχή.

3. Για κάθε είσοδο i επανέλαβε τα παρακάτω βήματα:

- Αν $t_i = 1$ και $y_i = 0$ τότε $w'_i = w_i + p_i$
- Αν $t_i = 0$ και $y_i = 1$ τότε $w'_i = w_i - p_i$
- Αν $t_i = y_i$ τότε $w'_i = w_i$

4. Ο αλγόριθμος τερματίζεται όταν δεν γίνεται πλέον καμία διόρθωση σε κανένα πρότυπο. Αυτό σημαίνει ότι όλοι οι στόχοι είναι ίσοι με όλες τις εξόδους.

Όταν παρουσιάσουμε στο δίκτυο όλα τα πρότυπα παρατηρούμε ότι το δίκτυο αρχίζει να εκπαιδεύεται και λέμε ότι το δίκτυο μαθαίνει σταδιακά τα πρότυπα τα οποία του παρουσιάζονται. Χρησιμοποιούμε τον όρο «μαθαίνει» ως συνώνυμο του «εκπαιδεύεται», εννοώντας ότι το δίκτυο αποκτά την ικανότητα να λύνει κάποιο πρόβλημα. Η εκπαίδευση αυτή δεν γίνεται σε ένα βήμα, αλλά ακολουθεί μία διαδικασία πολλών κύκλων, μια διαδικασία η οποία επαναλαμβάνεται πολλές φορές και κατά την οποία το δίκτυο καλυτερεύει συνεχώς τις τιμές των βαρών του. Μια κυκλική επανάληψη όλων των προτύπων ονομάζεται εποχή. Μετά από μερικούς κύκλους, που μπορεί και να είναι πολλές χιλιάδες, το δίκτυο έχει ήδη βρει τις κατάλληλες τιμές των w και έτσι έχει αναπτύξει τις ικανότητες του. Ακολούθως όμως η πρόοδος αυτή σταματάει και λέμε ότι το δίκτυο έχει πλέον συγκλίνει. Αυτό σημαίνει ότι οι τιμές των w δεν αλλάζουν πλέον και το δίκτυο έχει εκπαιδευθεί σωστά.

3.1.1 Παράδειγμα

Έστω ότι έχουμε το ακόλουθο σύνολο δεδομένων εισόδου και τους επιθυμητούς στόχους για κάθε είσοδο:

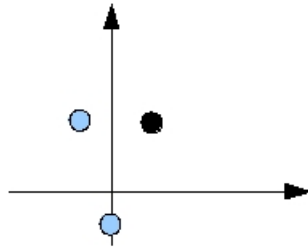
$$\{p_1 = [1 \ 2]^T, t_1 = 1\}, \{p_2 = [-1 \ 2]^T, t_2 = 0\}, \{p_3 = [0 \ -1]^T, t_3 = 0\}.$$

Στο Σχήμα 3.4 φαίνεται η γραφική αναπαράσταση των δεδομένων εκπαίδευσης. Τα σημεία που έχουν στόχο t ίσο με 0 αναπαρίστανται με μπλε κύκλο, ενώ αυτά που έχουν στόχο 1 αναπαρίστανται με μαύρο κύκλο.

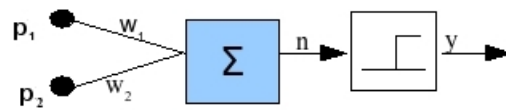
Το δίκτυο αυτού του προβλήματος θα έχει δύο εισόδους και μία έξοδο ενώ για απλότητα θεωρούμε ότι δεν έχει πόλωση. Το δίκτυο αυτό φαίνεται στο Σχήμα 3.5

Αφού έχουμε αφαιρέσει την πόλωση η διαχωριστική γραμμή των δεδομένων εισόδου θα είναι μια ευθεία η οποία περνάει από την αρχή των αξόνων. Η γραμμή αυτή θα πρέπει να διαχωρίζει τα διανύσματα εισόδου p_2, p_3 από το διάνυσμα p_1 . Από το Σχήμα 3.6 αποδεικνύεται ότι υπάρχει άπειρος αριθμός γραμμών που να ικανοποιούν αυτές τις προϋποθέσεις.

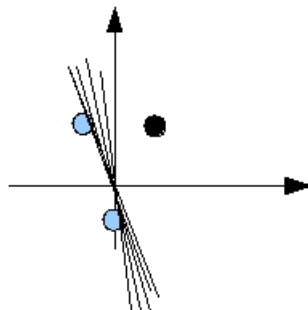
Τα διανύσματα βαρών που αντιστοιχούν στις διαθέσιμες διαχωριστικές ευθείες είναι ορθογώνια προς αυτές. Αυτό που θέλουμε είναι ένας κανόνες εκπαίδευσης που θα βρίσκει ένα διάνυσμα βαρών το οποίο έχει μια από αυτές τις κατευθύνσεις. Το μήκος του διανύσματος αυτού είναι αδιάφορο, σημαντικό είναι μόνο ο προσανατολισμός.



Σχήμα 3.4: Γραφική αναπαράσταση εισόδων.



Σχήμα 3.5: Δίκτυο Perceptron του προβλήματος.



Σχήμα 3.6: Γραμμική διαχωριστικότητα των δεδομένων του προβλήματος.

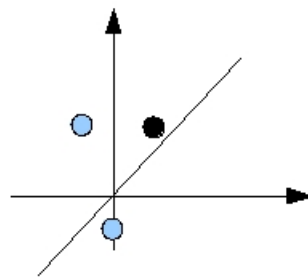
Ξεκινάμε αρχικοποιώντας το διάνυσμα των βαρών με τυχαίες τιμές. Έστω

$$w = [1 \quad -0.8].$$

Αν τώρα τροφοδοτήσουμε το δίκτυο με το διάνυσμα p_1 θα έχουμε

$$y = f(wp_1) = f([1 \quad -0.8] [1 \quad 2]^T) = f(-0.6) = 0.$$

Παρατηρούμε ότι η έξοδος του δικτύου δεν είναι η επιθυμητή. Αυτό φαίνεται και από το Σχήμα 3.7.



Σχήμα 3.7: Αποτέλεσμα μετά από τυχαία αρχικοποίηση των βαρών του δικτύου.

Για την επίλυση του προβλήματος θα πρέπει να προσαρμόσουμε τα βάρη σύμφωνα με τον κανόνα εκπαίδευσης Perceptron που είδαμε:

$$w' = w + p_1 = [1 \quad -0.8]^T + [1 \quad 2]^T = [2 \quad 1.2]^T.$$

Για το επόμενο πρότυπο εισόδου p_2 έχουμε ως έξοδο

$$y = f(wp_2) = f([2 \quad 1.2] [-1 \quad 2]^T) = f(0.4) = 1.$$

Ο στόχος t_2 που αντιστοιχεί στο πρότυπο p_2 είναι 0 ενώ το δίκτυο μας δίνει έξοδο 1. Επομένως

$$w' = w - p_2 = [2 \quad 1.2]^T - [-1 \quad 2]^T = [3 \quad -0.8]^T.$$

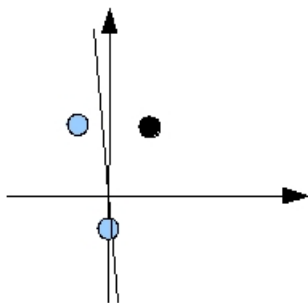
Ομοίως για το διάνυσμα p_3 έχουμε έξοδο

$$y = f(wp_3) = f([3 \quad -0.8] [0 \quad -1]^T) = f(0.8) = 1,$$

οπότε

$$w' = w - p_3 = [3 \quad -0.8]^T - [0 \quad -1]^T = [3 \quad 0.2]^T.$$

Όπως βλέπουμε στο Σχήμα 3.8 το δίκτυο έχει τελικά μάθει να διαχωρίζει τα τρία πρότυπα. Αν τροφοδοτήσουμε οποιοδήποτε από αυτά στο δίκτυο θα έχουμε σαν έξοδο τη σωστή κλάση για την κάθε είσοδο.



Σχήμα 3.8: Αναπαράσταση των δεδομένων μετά την εκπαίδευση του δικτύου.

3.2 Τροποποίηση του κανόνα Perceptron

Ο κανόνας εκπαίδευσης που αναφέραμε προηγουμένως μπορεί να τροποποιηθεί ώστε να έχουμε μία μόνο έκφραση. Αρχικά θα εισάγουμε μια καινούργια μεταβλητή που ονομάζεται σφάλμα του δικτύου. Το σφάλμα υπολογίζεται από τη σχέση $e = t - y$. Οπότε οι κανόνες τροποποιούνται ως εξής:

1. Αρχικοποίησε όλα τα βάρη και τις πολώσεις του δικτύου με τυχαίες τιμές.
2. Εισήγαγε τα πρότυπα με τη σειρά. Μόλις τελειώσουν ξανάρχισε από την αρχή.
3. Για κάθε είσοδο i επανέλαβε τα παρακάτω βήματα:
 - Αν $e = 1$ τότε $w'_i = w_i + p_i$
 - Αν $e = -1$ τότε $w'_i = w_i - p_i$
 - Αν $e = 0$ τότε $w'_i = w_i$
4. Ο αλγόριθμος τερματίζεται όταν δεν γίνεται πλέον καμία διόρθωση σε κανένα πρότυπο. Αυτό σημαίνει ότι όλοι οι στόχοι είναι ίσοι με όλες τις εξόδους.

Παρατηρώντας ότι το πρόσημο του p είναι ίδιο με το πρόσημο του e για κάθε κανόνα, μπορούμε να ενοποιήσουμε τους τρεις αυτούς κανόνες σε έναν. Οπότε έχουμε:

$$w' = w + ep = w + (t - y)p.$$

Η πόλωση αν υπάρχει πρέπει να τροποποιηθεί κι αυτή. Λαμβάνοντας υπόψη ότι η πόλωση είναι απλά ένα βάρος του οποίου η είσοδος είναι πάντα 1 έχουμε ότι

$$b' = b + e.$$

Κεφάλαιο 4

Ο κανόνας εκπαίδευσης του Hebb

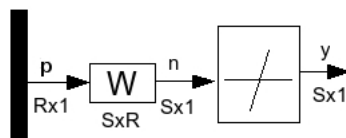
Το 1949 ο Donald O. Hebb στο βιβλίο του ‘ Η Οργάνωση της Συμπεριφοράς ’ [17] παρουσίασε ένα μοντέλο εκπαίδευσης των βιολογικών νευρώνων. Σύμφωνα με τον Hebb όταν ο άξονας ενός νευρώνα A είναι αρκετά κοντά ώστε να διεγείρει το νευρώνα B και συστηματικά συμμετέχει στην ενεργοποίησή του, τότε κάποια μεταβολική αλλαγή συμβαίνει είτε στο ένα απ’ τα δύο είτε και στα δύο κύτταρα έτσι ώστε η αποτελεσματικότητα με την οποία ο A διεγείρει τον B αυξάνεται. Αν και ο Hebb δεν κατάφερε να αποδείξει επιστημονικά τη θεωρία του, αργότερα αποδείχτηκε ότι κάποιοι νευρώνες έχουν πράγματι αυτή τη συμπεριφορά και οι θεωρίες του επηρέασαν την έρευνα της Νευροεπιστήμης.

Ο κανόνας εκπαίδευσης του Hebb μπορεί να χρησιμοποιηθεί σε συνδυασμό με διάφορες αρχιτεκτονικές νευρωνικών δικτύων. Στην ενότητα αυτή θα χρησιμοποιήσουμε μια απλή αρχιτεκτονική δικτύου για την ευκολότερη κατανόηση του κανόνα εκπαίδευσης, το γραμμικό συσχετιστή. Το δίκτυο αυτό παρουσίασαν οι James Anderson [1] και Teuvo Kohonen [21] και φαίνεται στο Σχήμα 4.1. Η έξοδος του δικτύου y καθορίζεται από το διάνυσμα εισόδου \mathbf{p} σύμφωνα με τη σχέση :

$$\mathbf{y} = \mathbf{W}\mathbf{p} \quad (4.1)$$

ή αναλυτικά :

$$y_i = \sum_{j=1}^2 w_{ij}p_j. \quad (4.2)$$



Σχήμα 4.1: Γραμμικός Συσχετιστής

Ο γραμμικός συσχετιστής αποτελεί παράδειγμα ενός τύπου νευρωνικών δικτύων που ονομάζονται μνήμες συσχέτισης. Ο ρόλος μιας μνήμης συσχέτισης είναι να μάθει ένα σύνολο από N ζεύγη διανυσμάτων εισόδου/εξόδου

$$\{p_1, t_1\}, \{p_2, t_2\}, \dots, \{p_N, t_N\}.$$

Δηλαδή αν το δίκτυο δεχτεί μια είσοδο $p = p_n$ τότε θα πρέπει να παράγει μια έξοδο $y = t_n$ για $n = 1, 2, \dots, N$. Αν η είσοδος αυτή αλλάξει κατά ένα πολύ μικρό ποσοστό για παράδειγμα $p = p_n + s$ τότε η έξοδος θα πρέπει επίσης να αλλάξει το ίδιο, δηλαδή $y = t_n + s$.

Σύμφωνα με τον κανόνα του Hebb αν μια θετική είσοδος p_j παράγει μια θετική έξοδο y_i τότε το βάρος w_{ij} θα πρέπει να αυξηθεί. Αυτό μπορεί να αναπαρασταθεί μαθηματικά ως

$$w_{ij}^{new} = w_{ij}^{old} + a f_i(y_{in}) g_j(p_{jn}), \quad (4.3)$$

όπου p_{jn} είναι το j στοιχείο της n εισόδου του διανύσματος p_n , y_{in} είναι το i στοιχείο της εξόδου του δικτύου όταν η n είσοδος παρουσιάζεται στο δίκτυο και a είναι μια θετική μεταβλητή που ονομάζεται ρυθμός εκπαίδευσης. Μπορούμε να απλοποιήσουμε την εξίσωση 4.3 ως

$$w_{ij}^{new} = w_{ij}^{old} + a y_{in} p_{jn}. \quad (4.4)$$

Η εξίσωση 4.4 επεκτείνει τον κανόνα εκπαίδευσης του Hebb ως εξής:

1. Αν δύο νευρώνες στα άκρα μιας σύναψης (σύνδεσης) ενεργοποιούνται ταυτόχρονα, η είσοδος δηλαδή και η έξοδος του δικτύου έχουν και οι δυο θετική τιμή ή και οι δυο αρνητική τιμή τότε η ισχύς της σύναψης αυτής αυξάνεται.
2. Αντίθετα αν δύο νευρώνες στα άκρα μιας σύναψης (σύνδεσης) ενεργοποιούνται ξεχωριστά, δηλαδή η είσοδος του δικτύου και η έξοδος έχουν διαφορετικά πρόσημα τότε η ισχύς της σύναψης αυτής μειώνεται.

Ο κανόνας του Hebb όπως ορίζεται από την εξίσωση 4.4 αποτελεί έναν κανόνα εκπαίδευσης χωρίς επίβλεψη καθώς δεν απαιτεί καμία πληροφορία για το στόχο εξόδου. Η εκπαίδευση με επίβλεψη σύμφωνα με τον κανόνα του Hebb δίνεται από την εξίσωση

$$w_{ij}^{new} = w_{ij}^{old} + t_{in} p_{jn}. \quad (4.5)$$

Η εξίσωση 4.5 μπορεί να γραφεί σε μορφή πινάκων ως

$$\mathbf{W}^{new} = \mathbf{W}^{old} + \mathbf{t}_n \mathbf{p}_n^T. \quad (4.6)$$

Αν υποθέσουμε ότι οι τιμές του πίνακα των βαρών αρχικοποιούνται έτσι ώστε να έχουν μηδενική τιμή και στη συνέχεια κάθε ζευγάρι εισόδου/εξόδου από τα N ζευγάρια εφαρμόζονται μία φορά στην εξίσωση 4.6, τότε μπορούμε να γράψουμε

$$\mathbf{W} = \mathbf{t}_1 \mathbf{p}_1^T + \mathbf{t}_2 \mathbf{p}_2^T + \dots + \mathbf{t}_N \mathbf{p}_N^T = \sum_{n=1}^N \mathbf{t}_n \mathbf{p}_n^T. \quad (4.7)$$

Η εξίσωση 4.7 μπορεί να αναπαρασταθεί σε μορφή πίνακα ως

$$\mathbf{W} = [t_1 \quad t_2 \quad \dots \quad t_N] \begin{bmatrix} p_1^T \\ p_2^T \\ \vdots \\ p_N^T \end{bmatrix} = TP^T, \quad (4.8)$$

όπου $T = [t_1 \quad t_2 \quad \dots \quad t_N]$ και $P = [p_1 \quad p_2 \quad \dots \quad p_N]$.

Αν τα διανύσματα εισόδου p_n είναι ορθοκανονικά, τότε αν p_k είναι μια είσοδος του δικτύου, η έξοδος υπολογίζεται από την εξίσωση

$$\mathbf{y} = \mathbf{W}\mathbf{p}_k = \left(\sum_{n=1}^N t_n p_n^T \right) p_k = \sum_{n=1}^N t_n (p_n^T p_k). \quad (4.9)$$

Αφού τα διανύσματα p_n είναι ορθοκανονικά ισχύει ότι

$$(p_n^T p_k) = \begin{cases} 1 & \text{για } n = k \\ 0 & \text{για } n \neq k \end{cases} \quad (4.10)$$

Οπότε η εξίσωση 4.9 μπορεί να γραφεί

$$\mathbf{y} = \mathbf{W}\mathbf{p}_k = \mathbf{t}_k. \quad (4.11)$$

Δηλαδή η έξοδος του δικτύου ισούται με το στόχο. Αυτό μας δείχνει ότι αν τα διανύσματα εισόδου είναι ορθοκανονικά τότε ο κανόνας του Hebb παράγει τη σωστή έξοδο για κάθε είσοδο. Αν όμως τα διανύσματα δεν είναι ορθογώνια τότε ο κανόνας του Hebb παράγει κάποια λάθη. Η έξοδος του δικτύου δίνεται τότε από την εξίσωση

$$\mathbf{y} = \mathbf{W}\mathbf{p}_k = t_k + \sum_{n \neq k} t_n (p_n^T p_k) \quad (4.12)$$

Υπάρχουν μέθοδοι οι οποίες μπορούν να χρησιμοποιηθούν ώστε να μειώνονται αυτά τα λάθη. Μία από αυτές τις μεθόδους είναι ο ψευδοανάστροφος κανόνας.

Είπαμε ότι ο στόχος του γραμμικού συσχετιστή είναι η παραγωγή μιας εξόδου t_n για μία είσοδο p_n . Αυτό εκφράζεται από τη σχέση

$$\mathbf{W}\mathbf{p}_n = t_n$$

για $q = 1, 2, \dots, N$. Ο ψευδοανάστροφος κανόνας επιλέγει έναν πίνακα βαρών W ο οποίος ελαχιστοποιεί τον ακόλουθο δείκτη απόδοσης

$$F(W) = \sum_{n=1}^N \|t_n - W p_n\|^2. \quad (4.13)$$

Αν τα διανύσματα εισόδου είναι ορθοκανονικά τότε χρησιμοποιούμε τον κανόνα του Hebb για να βρούμε τον πίνακα των βαρών W , ενώ ο δείκτης απόδοσης $F(W)$ είναι 0. Αν όμως

τα διανύσματα δεν είναι ορθοκανονικά τότε επιλέγουμε τον πίνακα W που ελαχιστοποιεί την εξίσωση 4.13 σύμφωνα με τον κανόνα

$$W = TP^+, \quad (4.14)$$

όπου P^+ είναι ένας πίνακας ο οποίος ονομάζεται ψευδοανάστροφος Moore-Penrose και πρέπει να ικανοποιεί τις παρακάτω ιδιότητες

- $PP^+P = P$,
- $P^+PP^+ = P^+$,
- $P^+P = (P^+P)^T$,
- $PP^+ = (PP^+)^T$

Όταν ο αριθμός, R , των γραμμών του πίνακα P είναι μεγαλύτερος από τον αριθμό των στηλών, Q , του P , και οι στήλες του πίνακα P είναι ανεξάρτητες, τότε ο ψευδοανάστροφος πίνακας υπολογίζεται από τη σχέση

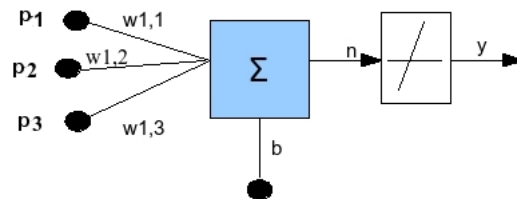
$$P^+ = (P^T P)^{-1} P^T. \quad (4.15)$$

Παράδειγμα 4.1

Έστω ότι έχουμε τα παρακάτω δεδομένα εισόδου και επιθυμητών στόχων

$$\left\{ \mathbf{p}_1 = \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix}, t_1 = [-1] \right\} \left\{ \mathbf{p}_2 = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}, t_2 = [-1] \right\}.$$

Το δίκτυο του προβλήματος έχει τρεις εισόδους και μία έξοδο και φαίνεται στο Σχήμα 4.2. Αν κανονικοποιήσουμε τις εισόδους τότε έχουμε



Σχήμα 4.2: Δίκτυο Adaline του παραδείγματος 4.1

$$\left\{ \mathbf{p}_1 = \begin{bmatrix} 0.5774 \\ -0.5774 \\ -0.5774 \end{bmatrix}, t_1 = [-1] \right\} \left\{ \mathbf{p}_2 = \begin{bmatrix} 0.5774 \\ 0.5774 \\ -0.5774 \end{bmatrix}, t_2 = [-1] \right\}.$$

Τότε αν χρησιμοποιήσουμε τον κανόνα του Hebb για να βρούμε τον πίνακα των βαρών έχουμε

$$\mathbf{W} = \mathbf{TP}^T = \begin{bmatrix} -1 & 1 \end{bmatrix} \begin{bmatrix} 0.5774 & -0.5774 & -0.5774 \\ 0.5774 & 0.5774 & -0.5774 \end{bmatrix} = \begin{bmatrix} 0 & 1.1548 & 0 \end{bmatrix}.$$

Αφού έχουμε βρει τον πίνακα των βαρών του συστήματος αν τροφοδοτήσουμε το δίκτυο με τις πρότυπες εισόδους τότε θα πάρουμε

$$\mathbf{Wp}_1 = \begin{bmatrix} 0 & 1.1548 & 0 \end{bmatrix} \begin{bmatrix} 0.5774 \\ -0.5774 \\ -0.5774 \end{bmatrix} = [-0.6668],$$

$$\mathbf{Wp}_2 = \begin{bmatrix} 0 & 1.1548 & 0 \end{bmatrix} \begin{bmatrix} 0.5774 \\ 0.5774 \\ -0.5774 \end{bmatrix} = [0.6668].$$

Παρατηρούμε ότι οι έξοδοι είναι κοντά αλλά δεν είναι ίσες με τις επιθυμητές εξόδους.

Αν όμως χρησιμοποιήσουμε τον ψευδοανάστροφο κανόνα τότε δε χρειάζεται να κανονικοποιήσουμε τις εισόδους. Στην περίπτωση αυτή ο πίνακας των βαρών υπολογίζεται από την εξίσωση 4.14:

$$\mathbf{W} = \mathbf{TP}^+ = \begin{bmatrix} -1 & 1 \end{bmatrix} \left(\begin{bmatrix} 1 & 1 \\ -1 & 1 \\ -1 & -1 \end{bmatrix} \right)^+,$$

όπου ο ψευδοανάστροφος πίνακας υπολογίζεται από τη σχέση 4.15:

$$\mathbf{P}^+ = (\mathbf{P}^T \mathbf{P})^{-1} \mathbf{P}^T = \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix}^{-1} \begin{bmatrix} 1 & -1 & -1 \\ 1 & 1 & -1 \end{bmatrix} = \begin{bmatrix} 0.25 & -0.5 & -0.25 \\ 0.25 & 0.5 & -0.25 \end{bmatrix}.$$

Οπότε ο πίνακας των βαρών είναι

$$\mathbf{W} = \mathbf{TP}^+ = \begin{bmatrix} -1 & 1 \end{bmatrix} \begin{bmatrix} 0.25 & -0.5 & -0.25 \\ 0.25 & 0.5 & -0.25 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}.$$

Αν τώρα τροφοδοτήσουμε το δίκτυο με τις πρότυπες εισόδους τότε θα πάρουμε

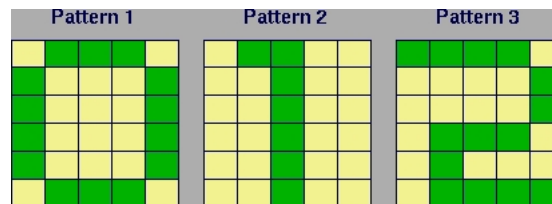
$$\mathbf{Wp}_1 = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} = [-1],$$

$$\mathbf{Wp}_2 = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} = [1].$$

Βλέπουμε ότι οι έξοδοι του δικτύου ταιριάζουν ακριβώς με τις επιθυμητές εξόδους.

Εφαρμογή

Έστω ότι θέλουμε να εφαρμόσουμε τον κανόνα του Hebb σε ένα απλό πραγματικό πρόβλημα αναγνώρισης προτύπων. Για αυτό το παράδειγμα χρησιμοποιούμε μια μνήμη αυτο-συσχετίσης, όπου η επιθυμητή έξοδος ισούται με την είσοδο έτσι ώστε να αποθηκεύσουμε ένα σύνολο προτύπων και στη συνέχεια να ανακαλέσουμε αυτά τα πρότυπα. Τα πρότυπα τα οποία θέλουμε να αποθηκεύσουμε παρουσιάζονται στην εικόνα 4.3 και αναπαριστούν τα ψηφία 0, 1 και 2 αντίστοιχα.



Σχήμα 4.3: Αναπαράσταση ψηφίων

Για να εισάγουμε αυτά τα πρότυπα σαν είσοδο στο δίκτυο θα πρέπει να μετατρέψουμε τα ψηφία αυτά σε πίνακες. Κάθε κίτρινο τετράγωνο θα αναπαρίσταται ως -1 ενώ κάθε πράσινο τετράγωνο ως 1 . Τότε για να δημιουργήσουμε κάθε διάνυσμα εισόδου σκανάρουμε κάθε τετράγωνο παίρνοντας μία στήλη κάθε φορά. Για παράδειγμα το πρώτο διάνυσμα εισόδου θα είναι

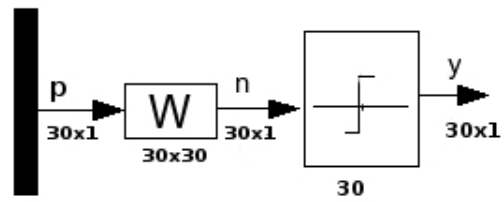
$$\mathbf{p}_1 = [-1 \ 1 \ 1 \ 1 \ 1 \ -1 \ 1 \ -1 \ -1 \ -1 \ -1 \ 1 \ 1 \ -1 \ \dots \ 1 \ -1]^T.$$

Το διάνυσμα \mathbf{p}_1 αντιστοιχεί στο ψηφίο 0, το \mathbf{p}_2 στο ψηφίο 1 και το \mathbf{p}_3 στο ψηφίο 2. Οπότε σύμφωνα με τον κανόνα του Hebb ο πίνακας των βαρών υπολογίζεται από τη σχέση

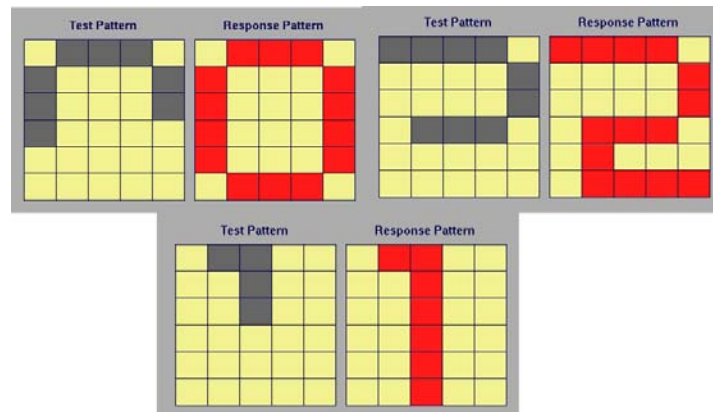
$$\mathbf{W} = \mathbf{p}_1\mathbf{p}_1^T + \mathbf{p}_2\mathbf{p}_2^T + \mathbf{p}_3\mathbf{p}_3^T.$$

Επειδή υπάρχουν μόνο δύο επιτρεπόμενες τιμές για τα στοιχεία των διανυσμάτων εισόδου μπορούμε να αντικαταστήσουμε τη γραμμική συνάρτηση ενεργοποίησης στο γραμμικό συσχετιστή με τη βηματική συνάρτηση. Το δίκτυο του προβλήματος φαίνεται στο Σχήμα 4.4.

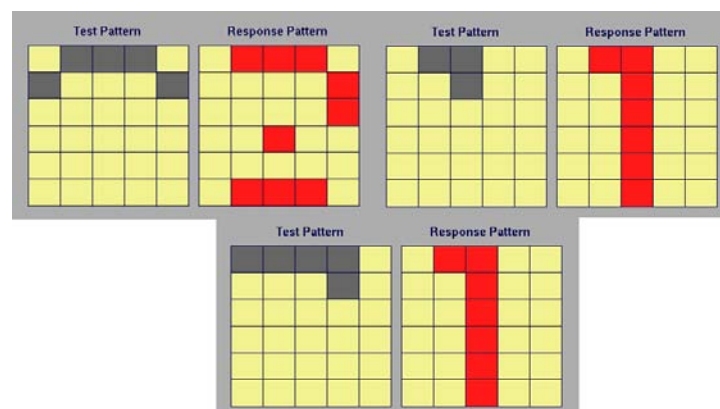
Στη συνέχεια αν παρουσιάσουμε στο δίκτυο εσφαλμένα πρότυπα εισόδου, αν για παράδειγμα το κάτω μισό των προτύπων απουσιάζει τότε όπως παρατηρούμε στο Σχήμα 4.5 το δίκτυο παράγει τη σωστή έξοδο για κάθε είσοδο. Αν αφαιρέσουμε τα δύο τρίτα των προτύπων εισόδου όπως φαίνεται στο Σχήμα 4.6 τότε στην περίπτωση αυτή μόνο το ψηφίο 1 ανακτάται σωστά από το δίκτυο, ενώ τα άλλα δύο πρότυπα παράγουν αποτελέσματα που δεν αντιστοιχούν σε κανένα από τα πρότυπα εκπαίδευσης. Τέλος παρατηρούμε την έξοδο του δικτύου για δεδομένα εισόδου τα οποία περιέχουν θόρυβο. Για να δημιουργήσουμε αυτά τα δεδομένα αλλάζουμε τυχαία 7 στοιχεία από κάθε πρότυπο. Τα αποτελέσματα φαίνονται στο Σχήμα 4.7. Παρατηρούμε ότι το δίκτυο ανακτά σωστά όλα τα πρότυπα.



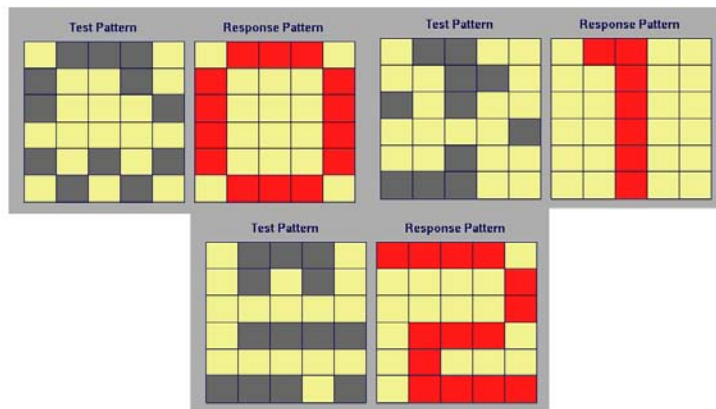
Σχήμα 4.4: Δίκτυο αυτοσυσχέτισης για αναγνώριση προτύπων.



Σχήμα 4.5: Ανάκτηση των εισόδων για 50% εσφαλμένα πρότυπα.



Σχήμα 4.6: Ανάκτηση των εισόδων για 67% εσφαλμένα πρότυπα.



Σχήμα 4.7: Ανάκτηση προτύπων στα οποία έχει εισαχθεί θόρυβος

Κεφάλαιο 5

Μέθοδοι Βελτιστοποίησης

Οι βασικές αρχές της βελτιστοποίησης ανακαλύφθηκαν κατά τη διάρκεια του 17ου αιώνα από επιστήμονες και μαθηματικούς όπως ο Kepler, ο Fermat, ο Newton και ο Leibniz. Ο χώρος της βελτιστοποίησης αναφέρεται στην αναζήτηση βέλτιστων παραμέτρων ενός συστήματος. Στα μαθηματικά ένα πρόβλημα βελτιστοποίησης διατυπώνεται σαν πρόβλημα ελαχιστοποίησης ή μεγιστοποίησης μιας συνάρτησης μιας μεταβλητής ή πολλών μεταβλητών. Η γενική μορφή ενός προβλήματος βελτιστοποίησης ορίζεται μαθηματικά ως η εύρεση ενός διανύσματος

$$\mathbf{x} = [x_1 \quad x_2 \quad \dots \quad x_n]$$

το οποίο ελαχιστοποιεί τη συνάρτηση

$$F(\mathbf{x}).$$

Δεδομένου ότι η μεγιστοποίηση μια συνάρτησης $F(x)$ ισοδυναμεί με ελαχιστοποίηση της συνάρτησης $-F(x)$, ο παραπάνω ορισμός καλύπτει όλα τα δυνατά προβλήματα βελτιστοποίησης. Οι μέθοδοι βελτιστοποίησης αναφέρονται και ως Τεχνικές Μαθηματικού Προγραμματισμού και αποτελούν αντικείμενο μελέτης ξεχωριστού κλάδου των Μαθηματικών. Η ύπαρξη πολλών διαφορετικών κατηγοριών προβλημάτων έχει οδηγήσει στην ανάπτυξη μιας πλειάδας μεθόδων βελτιστοποίησης.

Οι αλγόριθμοι βελτιστοποίησης γενικά είναι αλγόριθμοι οι οποίοι εντοπίζουν τα βέλτιστα σημεία. Μπορούν να χρησιμοποιηθούν στα Τεχνητά Νευρωνικά Δίκτυα και αποτελούν μια διαφορετική κατηγορία κανόνων εκπαίδευσης. Οι κανόνες αυτοί ξεχωρίζουν από το γεγονός ότι κατά τη διάρκεια της εκπαίδευσης οι παράμετροι του δικτύου, δηλαδή τα βάρη και οι πολώσεις, προσαρμόζονται έτσι ώστε να βελτιστοποιείται η απόδοση του δικτύου.

Η διαδικασία βελτιστοποίησης πραγματοποιείται σε δύο βήματα. Αρχικά καθορίζεται η απόδοση του δικτύου χρησιμοποιώντας ένα μέτρο το οποίο αποκαλούμε δείκτη απόδοσης. Ο δείκτης απόδοσης είναι μικρός όταν το δίκτυο εκπαιδεύει σωστά και μεγάλος όταν το δίκτυο δεν αποδίδει με το βέλτιστο τρόπο. Στη συνέχεια γίνεται προσαρμογή των παραμέτρων του δικτύου, ψάχνοντας στο χώρο των βαρών, έτσι ώστε να μειωθεί ο δείκτης απόδοσης.

5.1 Το διάνυσμα των μερικών παραγώγων και ο Εσσιανός πίνακας

Θεωρούμε ότι ο δείκτης απόδοσης που θέλουμε να ελαχιστοποιήσουμε αναπαρίσταται από μια συνάρτηση $F(x)$ όπου x είναι η μεταβλητή που θέλουμε να προσαρμόσουμε. Υποθέτουμε ότι ο δείκτης απόδοσης είναι μια αναλυτική συνάρτηση τέτοια ώστε να υπάρχουν όλες οι παράγωγοί της. Τότε μπορεί να αναπαρασταθεί από τις σειρές Taylor για κάποιο αρχικό σημείο. Η αναπαράσταση μίας συνάρτησης με τη μορφή σειράς Taylor γράφεται ως ανάπτυγμα των παραγώγων της για κάποιο αρχικό σημείο x^* , το οποίο έχει την ακόλουθη μορφή:

$$\begin{aligned} F(x) &= F(x^*) + \frac{d}{dx}F(x)|_{x=x^*}(x - x^*) + \\ &+ \frac{1}{2} \frac{d^2}{dx^2}F(x)|_{x=x^*}(x - x^*)^2 + \dots \\ &+ \frac{1}{n!} \frac{d^n}{dx^n}F(x)|_{x=x^*}(x - x^*) + \dots \end{aligned} \quad (5.1)$$

Αν για παράδειγμα η συνάρτηση η οποία θέλουμε να προσεγγίσουμε είναι η

$$F(x) = \cos(x), \quad (5.2)$$

τότε αυτή μπορεί να αναλυθεί χρησιμοποιώντας τις σειρές Taylor για $x^* = 0$ ως

$$\begin{aligned} F(x) &= \cos(x) \\ &= \cos(0) - \sin(0)(x - 0) - \frac{1}{2}\cos(0)(x - 0)^2 + \frac{1}{6}\sin(0)(x - 0)^3 + \dots \\ &= 1 - \frac{1}{2}x^2 + \frac{1}{24}x^4 + \dots \end{aligned} \quad (5.3)$$

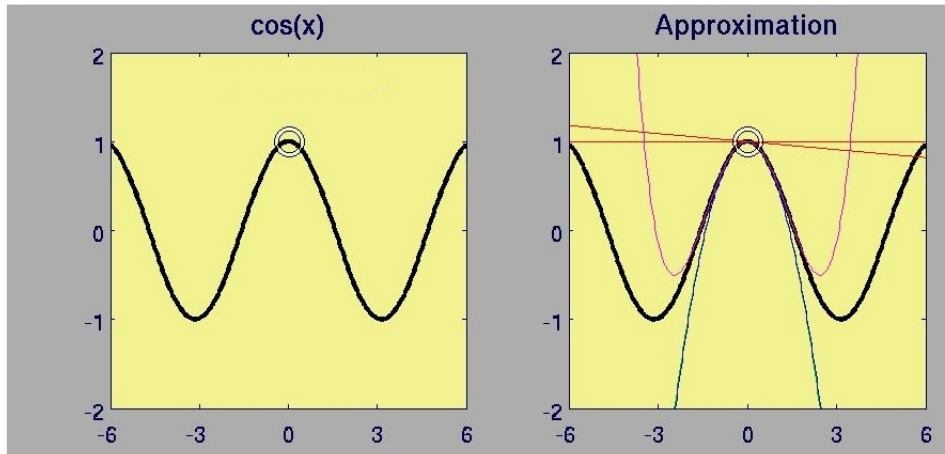
Η συνάρτηση $F(x)$ και η ανάλυσή της με σειρά Taylor για τις τέσσερις πρώτες συνιστώσες φαίνεται στο Σχήμα 5.1.

Χρησιμοποιούμε τις σειρές Taylor έτσι ώστε να υπολογίσουμε την απόδοση του δικτύου μειώνοντας την παραπάνω έκφραση σε έναν πεπερασμένο αριθμό όρων. Ο δείκτης απόδοσης του νευρωνικού δικτύου όμως θα είναι μια συνάρτηση όλων των παραμέτρων του δικτύου. Για το λόγο αυτό η εξίσωση 5.5 επεκτείνεται για πολλές μεταβλητές:

$$F(\mathbf{x}) = F(x_1, x_2, \dots, x_n) \quad (5.4)$$

Επομένως για κάποιο σημείο x^* έχουμε

$$\begin{aligned} F(\mathbf{x}) &= F(\mathbf{x}^*) + \frac{d}{dx_1}F(\mathbf{x})|_{\mathbf{x}=\mathbf{x}^*}(x_1 - x_1^*) + \frac{d}{dx_2}F(\mathbf{x})|_{\mathbf{x}=\mathbf{x}^*}(x_2 - x_2^*) + \dots \\ &+ \frac{d}{dx_n}F(\mathbf{x})|_{\mathbf{x}=\mathbf{x}^*}(x_n - x_n^*) + \frac{1}{2} \frac{d^2}{dx_1^2}F(\mathbf{x})|_{\mathbf{x}=\mathbf{x}^*}(x_1 - x_1^*)^2 + \\ &+ \frac{1}{2} \frac{d^2}{2dx_1dx_2}F(\mathbf{x})|_{\mathbf{x}=\mathbf{x}^*}(x_1 - x_1^*)(x_2 - x_2^*) + \dots \end{aligned} \quad (5.5)$$



Σχήμα 5.1: Συνάρτηση συνημιτόνου και η προσέγγιση της με σειρές Taylor

Η οποία σε μορφή πίνακα γράφεται

$$F(\mathbf{x}) = F(\mathbf{x}^*) + \nabla F(\mathbf{x})^T|_{\mathbf{x}=\mathbf{x}^*}(\mathbf{x} - \mathbf{x}^*) + \frac{1}{2}(\mathbf{x} - \mathbf{x}^*)^T \nabla^2 F(\mathbf{x})|_{\mathbf{x}=\mathbf{x}^*}(\mathbf{x} - \mathbf{x}^*) + \dots \quad (5.6)$$

όπου $\nabla F(\mathbf{x})$ είναι το διάνυσμα των μερικών παραγώγων και ορίζεται ως

$$\nabla F(\mathbf{x}) = \left[\frac{d}{dx_1} F(\mathbf{x}) \quad \frac{d}{dx_2} F(\mathbf{x}) \quad \dots \quad \frac{d}{dx_n} F(\mathbf{x}) \right]^T \quad (5.7)$$

και $\nabla^2 F(\mathbf{x})$ είναι ο Εσσιανός πίνακας

$$\nabla^2 F(\mathbf{x}) = \begin{bmatrix} \frac{d^2}{dx_1^2} F(\mathbf{x}) & \frac{d^2}{dx_1 dx_2} F(\mathbf{x}) & \dots & \frac{d^2}{dx_1 dx_n} F(\mathbf{x}) \\ \frac{d^2}{dx_2 dx_1} F(\mathbf{x}) & \frac{d^2}{dx_2^2} F(\mathbf{x}) & \dots & \frac{d^2}{dx_2 dx_n} F(\mathbf{x}) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{d^2}{dx_n dx_1} F(\mathbf{x}) & \frac{d^2}{dx_n dx_2} F(\mathbf{x}) & \dots & \frac{d^2}{dx_n^2} F(\mathbf{x}) \end{bmatrix}. \quad (5.8)$$

Ενώ στην ελαχιστοποίηση συναρτήσεων μίας μεταβλητής μπορούν να χρησιμοποιηθούν αναλυτικές και αλγεβρικές μέθοδοι για τον ακριβή ορισμό ελαχιστών, στη μελέτη συναρτήσεων πολλών μεταβλητών χρησιμοποιούνται κυρίως αριθμητικές μέθοδοι για έναν προσεγγιστικό ορισμό ελαχιστων σημείων.

5.2 Βέλτιστο σημείο

Σε ένα πρόβλημα βελτιστοποίησης με τον όρο βέλτιστο σημείο εννοούμε ένα ελάχιστο σημείο της συνάρτησης που θέλουμε να ελαχιστοποιήσουμε. Σε ένα νευρωνικό δίκτυο θεωρούμε ότι ένα βέλτιστο σημείο είναι ένα ελάχιστο του δείκτη απόδοσης. Διακρίνουμε τρεις κατηγορίες ελαχιστων σημείων:

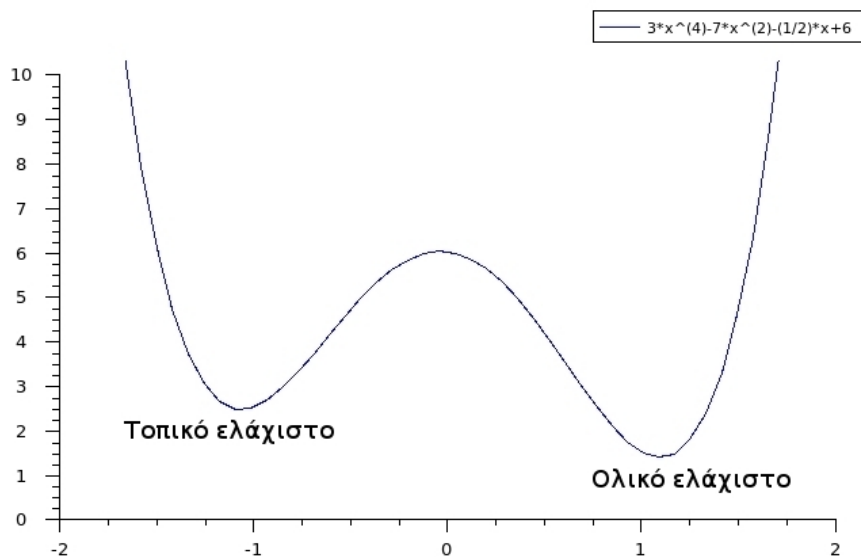
- **Ισχυρό ελάχιστο:** Το σημείο x^* είναι ένα ισχυρό ελάχιστο του $F(x)$ αν υπάρχει μια μεταβλητή $d > 0$ τέτοια ώστε $F(x^*) < F(x^* + Dx)$ για όλα τα Dx ώστε $d > \|Dx\| > 0$. Με άλλα λόγια αν μετακινηθούμε σε μια μικρή περιοχή γύρω από ένα ισχυρό ελάχιστο προς οποιαδήποτε κατεύθυνση τότε η συνάρτηση θα αυξηθεί.

- **Ολικό ελάχιστο:** Το σημείο x^* είναι ένα μοναδικό ολικό ελάχιστο του $F(x)$ αν $F(x^*) < F(x^* + Dx)$ για όλα τα $Dx \neq 0$. Για ένα απλό ισχυρό ελάχιστο, x^* , η συνάρτηση μπορεί να είναι μικρότερη από την $F(x^*)$ σε μερικά σημεία έξω από μια μικρή περιοχή του x^* . Για το λόγο αυτό μερικές φορές λέγεται τοπικό ελάχιστο. Για ένα ολικό ελάχιστο η συνάρτηση θα είναι μεγαλύτερη από το ελάχιστο σημείο για κάθε σημείο στον χώρο των παραμέτρων.
- **Ασθενές ελάχιστο:** Το σημείο x^* είναι ένα ασθενές ελάχιστο του $F(x)$ αν δεν είναι ισχυρό ελάχιστο και υπάρχει μια μεταβλητή $d > 0$ έτσι ώστε $F(x^*) \leq F(x^* + Dx)$ για όλα τα Dx ώστε $d > \|Dx\| > 0$. Προς οποιαδήποτε κατεύθυνση κι αν μετακινηθούμε από ένα ασθενές ελάχιστο η συνάρτηση δεν μπορεί να μειωθεί, αν και μπορεί να υπάρχουν κάποιες κατευθύνσεις όπου αν μετακινηθούμε η συνάρτηση παραμένει σταθερή.

Για παράδειγμα η συνάρτηση

$$F(x) = 3x^4 - 7x^2 - \frac{1}{2}x + 6$$

έχει δύο ισχυρά ελάχιστα σημεία τα οποία είναι κατά προσέγγιση για $x = -1.1$ και $x = 1.1$. Όπως παρατηρούμε στο Σχήμα 5.2 για μια τοπική γειτονική περιοχή αυτών των δύο σημείων η συνάρτηση αυξάνεται. Το σημείο για $x = 1.1$ είναι ολικό ελάχιστο καθώς δεν υπάρχει κανένα άλλο σημείο στο οποίο η συνάρτηση έχει μικρότερη τιμή και για $x = -1.1$ η συνάρτηση έχει ένα τοπικό ελάχιστο ενώ δεν υπάρχει κανένα ασθενές ελάχιστο.



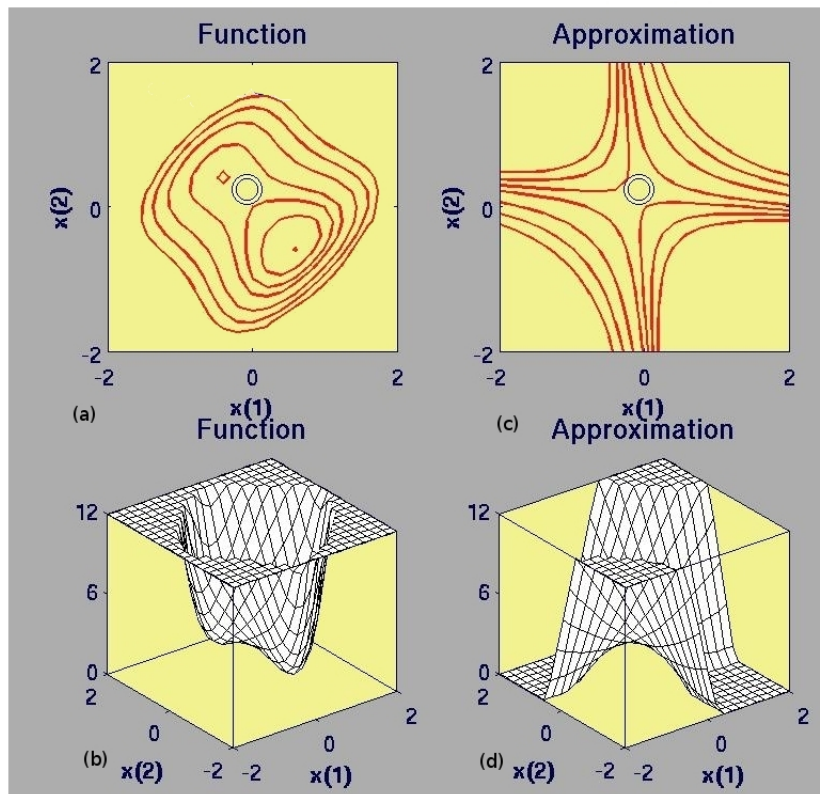
Σχήμα 5.2: Παράδειγμα τοπικού και ολικού ελάχιστου

Επίσης στο Σχήμα 5.3 στο διάγραμμα (b) φαίνεται η τρισδιάστατη αναπαράσταση της συνάρτησης

$$F(x) = (x_2 - x_1)^4 + 8x_1x_2 - x_1 + x_2 + 3$$

και στο διάγραμμα (a) το διάγραμμα ισοϋψών το οποίο είναι μια σειρά από καμπύλες κατά τις οποίες η τιμή της συνάρτησης παραμένει σταθερή. Παρατηρούμε ότι η συνάρτηση έχει ένα τοπικό ελάχιστο στο σημείο $(-0.42, 0.42)$ και ένα ολικό στο $(0.55, -0.55)$.

Υπάρχει ένα ακόμη ενδιαφέρον χαρακτηριστικό αυτής της συνάρτησης στο σημείο $(-0.13, 0.13)$. Το σημείο αυτό ονομάζεται σελοειδές (saddle point) λόγω του σχήματος που έχει η επιφάνεια στη γειτονική περιοχή και φαίνεται στα διαγράμματα (c) και (d). Το σημείο αυτό χαρακτηρίζεται από το γεγονός ότι κατά μήκος της γραμμής $x_1 = -x_2$ είναι τοπικό μέγιστο, ενώ κατά μήκος μιας γραμμής ορθογώνιας σε αυτή είναι τοπικό ελάχιστο.



Σχήμα 5.3: Παράδειγμα τοπικού και ολικού ελάχιστου

5.3 Απαραίτητες συνθήκες βελτιστοποίησης

Ως συνθήκες βελτιστοποίησης ορίζουμε τις συνθήκες που ικανοποιούνται από ένα βέλτιστο σημείο. Αυτές διακρίνονται σε 1ης τάξης και 2ης τάξης.

Η συνθήκη 1ης τάξης ορίζει ότι σε ένα ελάχιστο σημείο το διάνυσμα μερικών παραγώγων είναι ίσο με 0

$$\nabla F(x)|_{x=x^*} = 0. \quad (5.9)$$

Όλα τα σημεία τα οποία ικανοποιούν την εξίσωση 5.9 ονομάζονται στάσιμα σημεία (stationary points). Η συνθήκη αυτή είναι απαραίτητη αλλά δεν επαρκεί έτσι ώστε να πούμε ότι το x^* είναι τοπικό ελάχιστο.

Από την άλλη για να ορίσουμε τη συνθήκη 2ης τάξης είναι απαραίτητο να δώσουμε τον ορισμό για τον θετικά ορισμένο και ημιορισμένο πίνακα.

Ένας πίνακας \mathbf{A} είναι θετικά ορισμένος αν

$$\mathbf{z}^T \mathbf{A} \mathbf{z} > 0$$

για οποιοδήποτε διάνυσμα $\mathbf{z} \neq 0$ και θετικά ημιορισμένος αν

$$\mathbf{z}^T \mathbf{A} \mathbf{z} \geq 0.$$

Συγκεκριμένα αν όλες οι ιδιοτιμές του πίνακα \mathbf{A} είναι θετικές τότε ο πίνακας είναι θετικά ορισμένος, ενώ αν όλες οι ιδιοτιμές του \mathbf{A} είναι μη αρνητικές τότε ο πίνακας είναι θετικά ημιορισμένος.

Αν ο Εσσιανός πίνακας είναι θετικά ορισμένος έχουμε μια επαρκή συνθήκη που ορίζει την ύπαρξη ενός ισχυρού ελάχιστου. Η συνθήκη αυτή δεν είναι απαραίτητη καθώς ένα ελάχιστο μπορεί επίσης να είναι ισχυρό αν ο όρος δεύτερης τάξης των σειρών Taylor είναι μηδενικός αλλά ο όρος τρίτης τάξης είναι θετικός. Για το λόγο αυτό η απαραίτητη συνθήκη που ορίζει ένα ισχυρό ελάχιστο προϋποθέτει ο Εσσιανός πίνακας να είναι ένας πίνακας θετικά ημιορισμένος.

Συνοψίζοντας, οι απαραίτητες συνθήκες για να είναι το x^* ισχυρό ή ασθενές ελάχιστο είναι

$$\nabla F(x)|_{x=x^*} = 0$$

και

$$\nabla^2 F(x)|_{x=x^*} \text{ θετικά ημιορισμένος.}$$

Οι επαρκείς συνθήκες για να είναι το x^* ισχυρό ελάχιστο της $F(x)$ είναι

$$\nabla F(x)|_{x=x^*} = 0$$

και

$$\nabla^2 F(x)|_{x=x^*} \text{ θετικά ορισμένος.}$$

Για παράδειγμα θεωρούμε την παρακάτω συνάρτηση δύο μεταβλητών

$$F(\mathbf{x}) = x_1^4 + x_2^2. \quad (5.10)$$

Αρχικά εντοπίζουμε αν υπάρχουν στάσιμα σημεία, οπότε υπολογίζουμε την πρώτη παράγωγο της συνάρτησης, δηλαδή το διάνυσμα μερικών παραγώγων και το θέτουμε ίσο με 0

$$\nabla F(\mathbf{x}) = \begin{bmatrix} 4x_1^3 \\ 2x_2 \end{bmatrix} = 0.$$

Παρατηρούμε ότι το μόνο στάσιμο σημείο είναι το $x^* = 0$. Στη συνέχεια εξετάζουμε τη συνθήκη 2ης τάξης, υπολογίζουμε δηλαδή τον Εσσιανό πίνακα και τον θέτουμε ίσο 0

$$\nabla^2 F(\mathbf{x})|_{\mathbf{x}=0} = \begin{bmatrix} 12x_1^2 & 0 \\ 0 & 2 \end{bmatrix} |_{\mathbf{x}=0} = \begin{bmatrix} 0 & 0 \\ 0 & 2 \end{bmatrix}.$$

Ο πίνακας είναι θετικά ημιορισμένος, απαραίτητη συνθήκη για να είναι το $x^* = 0$ ισχυρό ελάχιστο. Οι δύο αυτές συνθήκες που εξετάσαμε δεν επαρκούν για να πούμε ότι το $x^* = 0$ είναι ένα ελάχιστο σημείο, δεν το έχουμε απορρίψει όμως ως πιθανότητα.

5.4 Ιδιότητες τετραγωνικών συναρτήσεων

Οι τετραγωνικές συναρτήσεις αποδεικνύονται ιδιαίτερα χρήσιμες καθώς εμφανίζονται σε πληθώρα εφαρμογών και αποτελούν έναν καθολικό τύπο του δείκτη απόδοσης. Επίσης πολλές συναρτήσεις μπορούν θεωρηθούν ως τετραγωνικές συναρτήσεις σε μία μικρή γειτονική περιοχή, ειδικότερα κοντά σε σημεία που είναι τοπικά ελάχιστα.

Η γενική μορφή μιας τετραγωνικής συνάρτησης ορίζεται από την εξίσωση

$$F(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{d}^T \mathbf{x} + c, \quad (5.11)$$

όπου \mathbf{A} είναι ένας συμμετρικός πίνακας.

Το διάνυσμα των μερικών παραγώγων αυτής της εξίσωσης είναι

$$\nabla F(\mathbf{x}) = \mathbf{A} \mathbf{x} + \mathbf{d}, \quad (5.12)$$

και ο Εσσιανός πίνακας ορίζεται ως

$$\nabla^2 F(\mathbf{x}) = \mathbf{A}. \quad (5.13)$$

Η συνάρτηση αυτή μπορεί να αναπαρασταθεί με σειρά Taylor με τις τρεις πρώτες συνιστώσες καθώς όλες οι παράγωγοι της συνάρτησης για τάξη μεγαλύτερη του 2 είναι 0.

Οι τετραγωνικές συναρτήσεις έχουν κάποια σημαντικά χαρακτηριστικά τα οποία συνοψίζονται παρακάτω:

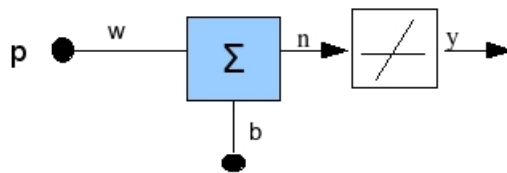
- Αν όλες οι ιδιοτιμές του Εσσιανού πίνακα είναι θετικές, τότε η συνάρτηση θα έχει ένα μοναδικό ισχυρό ελάχιστο.
- Αν όλες οι ιδιοτιμές είναι αρνητικές, τότε η συνάρτηση θα έχει ένα μοναδικό ισχυρό μέγιστο.
- Αν κάποιες ιδιοτιμές είναι θετικές ενώ άλλες είναι αρνητικές, τότε η συνάρτηση θα έχει ένα μοναδικό σελοειδές σημείο.
- Αν οι ιδιοτιμές είναι όλες μη αρνητικές, αλλά κάποιες από αυτές είναι μηδενικές, τότε η συνάρτηση είτε έχει ένα ασθενές ελάχιστο ή δεν έχει κανένα στάσιμο σημείο.
- Αν όλες οι ιδιοτιμές είναι μη θετικές, αλλά κάποιες από αυτές είναι μηδενικές, τότε η συνάρτηση είτε έχει ένα ασθενές μέγιστο ή δεν έχει κανένα στάσιμο σημείο.

Παράδειγμα 6.1 Έστω ότι έχουμε ένα γραμμικό νευρωνικό δίκτυο μιας εισόδου όπως φαίνεται στο Σχήμα 5.4. Υποθέτουμε ότι οι εισοδοί του δικτύου και οι επιθυμητές έξοδοι είναι

$$\{(p_1 = 2), (t_1 = 0.5)\}, \{(p_2 = -1), (t_2 = 0)\}$$

και έστω ότι ο δείκτης απόδοσης του δικτύου περιγράφεται από την εξίσωση

$$F(\mathbf{x}) = (t_1 - y_1(\mathbf{x}))^2 + (t_2 - y_2(\mathbf{x}))^2. \quad (5.14)$$



Σχήμα 5.4: Γραμμικό Νευρωνικό Δίκτυο

Οι παράμετροι αυτού δικτύου είναι το βάρος w και η πόλωση b . Επομένως ο πίνακας των παραμέτρων είναι

$$\mathbf{x} = \begin{bmatrix} w \\ b \end{bmatrix}.$$

Αυτό που θέλουμε είναι να σχεδιάσουμε τη συνάρτηση που ορίζει το δείκτη απόδοσης $F(\mathbf{x})$. Αρχικά θα δείξουμε ότι ο δείκτης απόδοσης είναι μια τετραγωνική συνάρτηση και στη συνέχεια θα υπολογίσουμε τις ιδιοτιμές και τα ιδιοδιανύσματα του Εσσιανού πίνακα για να σχεδιάσουμε το διάγραμμα ισοϋψών της συνάρτησης.

Η συνάρτηση $F(\mathbf{x})$ μπορεί να γραφεί ως

$$F(\mathbf{x}) = e_1^2 + e_2^2,$$

όπου $e_1 = t_1 - (wp_1 + b)$ και $e_2 = t_2 - (wp_2 + b)$. Η συνάρτηση αυτή μπορεί να γραφεί σε μορφή πίνακα ως εξής

$$F(\mathbf{x}) = \mathbf{e}^T \mathbf{e},$$

όπου

$$\mathbf{e} = \mathbf{t} - \begin{bmatrix} p_1 & 1 \\ p_2 & 1 \end{bmatrix} \mathbf{x} = \mathbf{t} - \mathbf{G}\mathbf{x}.$$

Επομένως ο δείκτης απόδοσης έχει τώρα την ακόλουθη μορφή

$$F(\mathbf{x}) = [\mathbf{t} - \mathbf{G}\mathbf{x}]^T [\mathbf{t} - \mathbf{G}\mathbf{x}] = \mathbf{t}^T \mathbf{t} - 2\mathbf{t}^T \mathbf{G}\mathbf{x} + \mathbf{x}^T \mathbf{G}^T \mathbf{G}\mathbf{x}.$$

Παρατηρούμε ότι ο δείκτης απόδοσης αυτού του δικτύου είναι μια τετραγωνική συνάρτηση της μορφής

$$F(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{A}\mathbf{x} + \mathbf{d}^T \mathbf{x} + c,$$

με

$$c = \mathbf{t}^T \mathbf{t}, d = -2\mathbf{G}^T \mathbf{t}, \mathbf{A} = 2\mathbf{G}^T \mathbf{G}.$$

Ξέρουμε ότι το διάνυσμα των μερικών παραγώγων μιας τετραγωνικής συνάρτησης δίνεται από τη σχέση $\nabla F(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{d}$ άρα

$$\nabla F(\mathbf{x}) = 2\mathbf{G}^T\mathbf{G}\mathbf{x} - 2\mathbf{G}^T\mathbf{t}.$$

Για να βρούμε τα στάσιμα σημεία θέτουμε το διάνυσμα των μερικών παραγώγων ίσο με το 0, δηλαδή

$$\mathbf{A}\mathbf{x} + \mathbf{d} = 0.$$

Άρα το στάσιμο σημείο θα είναι ίσο με

$$\mathbf{x}^* = -\mathbf{A}^{-1}\mathbf{d} = [\mathbf{G}^T\mathbf{G}]^{-1}\mathbf{G}^T\mathbf{t}.$$

Για

$$\mathbf{G} = \begin{bmatrix} p_1 & 1 \\ p_2 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ -1 & 1 \end{bmatrix} \text{ και } \mathbf{t} = \begin{bmatrix} 0.5 \\ 0 \end{bmatrix}$$

έχουμε

$$\mathbf{x}^* = [\mathbf{G}^T\mathbf{G}]^{-1}\mathbf{G}^T\mathbf{t} = \begin{bmatrix} 5 & 1 \\ 1 & 2 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} = \begin{bmatrix} 0.167 \\ 0.167 \end{bmatrix}.$$

Επομένως οι βέλτιστες παράμετροι του δικτύου είναι $w = 0.167$ και $b = 0.167$.

Έπειτα υπολογίζουμε τον Εσσιανό πίνακα ο οποίος δίνεται από την εξίσωση

$$\nabla^2 F(\mathbf{x}) = \mathbf{A} = 2\mathbf{G}^T\mathbf{G} = \begin{bmatrix} 10 & 2 \\ 2 & 4 \end{bmatrix}.$$

Οι ιδιοτιμές και τα ιδιοδιανύσματα του Εσσιανού πίνακα είναι

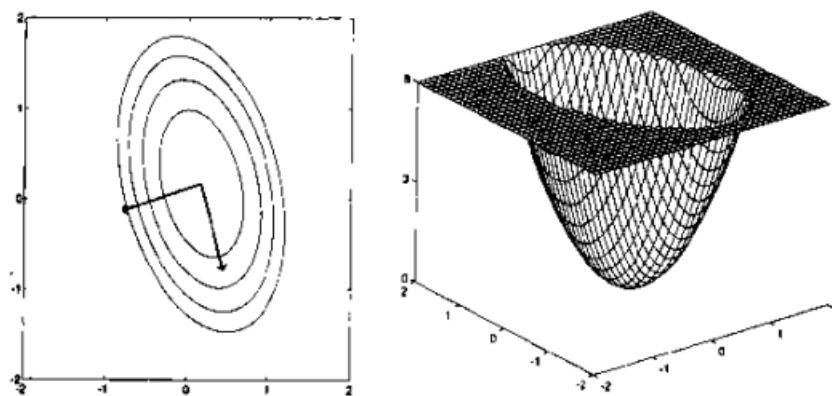
$$\left\{ (\lambda_1 = 10.6), \left(z_1 = \begin{bmatrix} 1 \\ 0.3 \end{bmatrix} \right) \right\}, \left\{ (\lambda_2 = 3.4), \left(z_2 = \begin{bmatrix} 0.3 \\ -1 \end{bmatrix} \right) \right\}.$$

Παρατηρούμε ότι όλες οι ιδιοτιμές του Εσσιανού πίνακα είναι θετικές άρα το x^* είναι ένα ισχυρό ελάχιστο. Επίσης επειδή η ιδιοτιμή λ_1 είναι μεγαλύτερη από την ιδιοτιμή λ_2 , γνωρίζουμε ότι οι καμπύλες στο διάγραμμα ισοϋψών θα έχουν ελλειπτική μορφή και ότι ο ένας από τους δύο άξονες των ελλείψεων θα έχει την κατεύθυνση του ιδιοδιανύσματος z_2 . Επίσης οι στο κέντρο του διαγράμματος θα βρίσκεται το σημείο \mathbf{x}^* . Η τρισδιάστατη αναπαράσταση αυτής της συνάρτησης και το διάγραμμα ισοϋψών φαίνεται στο Σχήμα 5.5.

5.5 Αλγόριθμοι βελτιστοποίησης

Στην ενότητα αυτή θα εξετάσουμε αλγόριθμους οι οποίοι βελτιστοποιούν το δείκτη απόδοσης $F(\mathbf{x})$. Οι αλγόριθμοι βελτιστοποίησης που θα αναλύσουμε είναι όλοι επαναληπτικοί, δηλαδή ξεκινάμε από μια τυχαία αρχική τιμή x_0 και στη συνέχεια προσαρμόζουμε την τιμή αυτή ανάλογα με την εξίσωση

$$\mathbf{x}_{k+j} = \mathbf{x}_k + a_k \mathbf{p}_k \quad (5.15)$$



Σχήμα 5.5: Τρισδιάστατη αναπαράσταση και διάγραμμα ισοϋψών της συνάρτησης $F(x)$

ή

$$\Delta \mathbf{x}_k = (\mathbf{x}_{k+1} - \mathbf{x}_k) = a_k \mathbf{p}_k, \quad (5.16)$$

όπου \mathbf{p}_k είναι ένα διάνυσμα το οποίο αναπαριστά μια κατεύθυνση και ονομάζεται διάνυσμα κλίσης και a_k είναι μια θετική μεταβλητή που ονομάζεται ρυθμός εκπαίδευσης και καθορίζει το μέγεθος του βήματος σε κάθε επανάληψη.

5.5.1 Η μέθοδος της πιο απότομης καθόδου

Ένας από τους πιο γνωστούς αλγορίθμους βελτιστοποίησης είναι η μέθοδος της πιο απότομης καθόδου. Η μέθοδος αυτή έχει σαν στόχο τη συνεχή αναζήτηση καλύτερων λύσεων. Χρησιμοποιώντας την εξίσωση 5.15 για να βρούμε το ελάχιστο σημείο θέλουμε η συνάρτηση που ορίζει το δείκτη απόδοσης να μειώνεται σε κάθε επανάληψη. Δηλαδή, θέλουμε να βρούμε το διάνυσμα κλίσης \mathbf{p}_k έτσι ώστε για ένα μικρό ρυθμό εκπαίδευσης να ικανοποιείται η εξίσωση

$$F(\mathbf{x}_{k+1}) < F(\mathbf{x}_k). \quad (5.17)$$

Η πρώτη συνιστώσα της σειράς Taylor για την αρχική μας υπόθεση x_k δίνεται από τη σχέση

$$F(\mathbf{x}_{k+1}) = F(\mathbf{x}_k + \Delta \mathbf{x}_k) = F(\mathbf{x}_k) + \mathbf{g}_k^T \Delta \mathbf{x}_k,$$

όπου \mathbf{g}_k είναι το διάνυσμα των μερικών παραγώγων για την αρχική υπόθεση \mathbf{x}_k

$$\mathbf{g}_k = \nabla F(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_k}.$$

Για να ισχύει η εξίσωση 5.17 πρέπει

$$\mathbf{g}_k^T \Delta \mathbf{x}_k = a_k \mathbf{g}_k^T \mathbf{p}_k < 0,$$

όπου η τιμή του ρυθμού εκπαίδευσης a_k επιλέγεται να είναι μικρή αλλά μεγαλύτερη από το 0 έτσι ώστε

$$\mathbf{g}_k^T \mathbf{p}_k < 0. \quad (5.18)$$

Οποιοδήποτε διάνυσμα \mathbf{p}_k ικανοποιεί την εξίσωση 5.18 ονομάζεται κλίση καθόδου. Αν επιλέξουμε ένα κοντινό σημείο προς αυτή την κατεύθυνση τότε η συνάρτηση θα πρέπει να ελαττωθεί. Για να έχουμε την πιο απότομη κάθοδο, δηλαδή για να βρούμε την κατεύθυνση προς την οποία η συνάρτηση ελαττώνεται πιο γρήγορα πρέπει η σχέση

$$\mathbf{g}_k^T \mathbf{p}_k \quad (5.19)$$

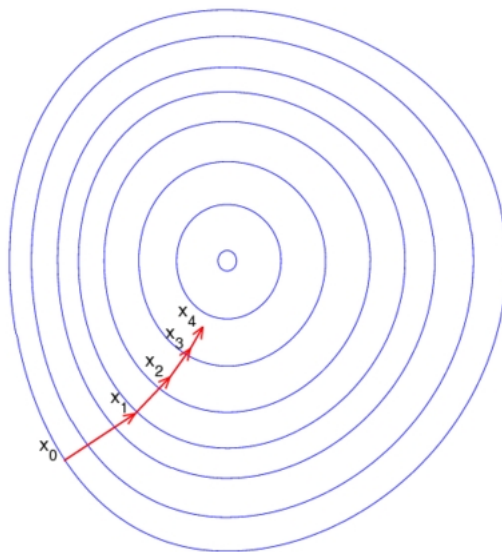
να έχει τη μεγαλύτερη αρνητική τιμή. Η σχέση 5.19 είναι ένα εσωτερικό γινόμενο μεταξύ του διανύσματος μερικών παραγώγων και του διανύσματος κλίσης και έχει τη μεγαλύτερη αρνητική τιμή όταν το διάνυσμα κλίσης ισούται με την αρνητική τιμή του διανύσματος μερικών παραγώγων, δηλαδή

$$\mathbf{p}_k = -\mathbf{g}_k.$$

Επομένως η μέθοδος της πιο απότομης καθόδου υπολογίζεται από την εξίσωση

$$\mathbf{x}_{k+1} = \mathbf{x}_k - a_k \mathbf{g}_k. \quad (5.20)$$

Υπάρχουν δύο μέθοδοι για τον καθορισμό του ρυθμού εκπαίδευσης a_k τις οποίες θα αναλύσουμε στη συνέχεια. Σύμφωνα με τη πρώτη μέθοδο επιλέγουμε το ρυθμό εκπαίδευσης έτσι ώστε να ελαχιστοποιείται ο δείκτης απόδοσης $F(\mathbf{x})$, ενώ ένας δεύτερος τρόπος είναι η επιλογή μιας προκαθορισμένης τιμής για το ρυθμό εκπαίδευσης.



Σχήμα 5.6: Η μέθοδος της πιο απότομης καθόδου

Στο Σχήμα 5.6 φαίνεται το διάγραμμα ισοϋψών μιας συνάρτησης και η μέθοδος της πιο απότομης καθόδου όπως υπολογίζεται από ένα σημείο \mathbf{x}_0 . Το κόκκινο τόξο μας δείχνει την κατεύθυνση του αντίθετου του διανύσματος των μερικών παραγώγων σε κάθε σημείο. Το διάνυσμα αυτό σε κάθε σημείο είναι κάθετο στις καμπύλες που ορίζουν τη συνάρτηση στο διάγραμμα ισοϋψών. Παρατηρούμε ότι η μέθοδος της πιο απότομης καθόδου μας οδηγεί σε ένα ελάχιστο, δηλαδή στο σημείο στο οποίο τη τιμή της συνάρτησης είναι η ελάχιστη.

Σταθερός ρυθμός εκπαίδευσης

Υποθέτουμε ότι ο δείκτης απόδοσης είναι μια τετραγωνική συνάρτηση, έχει δηλαδή τη μορφή

$$F(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{d}^T \mathbf{x} + c.$$

Γνωρίζουμε από την εξίσωση 5.12 ότι η το διάνυσμα των μερικών παραγώγων της συνάρτησης δίνεται από την εξίσωση

$$\nabla F(\mathbf{x}) = \mathbf{A} \mathbf{x} + \mathbf{d}.$$

Αν κάνουμε αντικατάσταση του \mathbf{g}_k στη σχέση 5.20 σύμφωνα με αυτή την εξίσωση τότε έχουμε

$$\mathbf{x}_{k+1} = \mathbf{x}_k - a \mathbf{g}_k = \mathbf{x}_k - a(\mathbf{A} \mathbf{x}_k + \mathbf{d})$$

ή

$$\mathbf{x}_{k+1} = [\mathbf{I} - a\mathbf{A}] \mathbf{x}_k - a\mathbf{d}. \quad (5.21)$$

Η εξίσωση 5.21 ορίζει ένα γραμμικό δυναμικό σύστημα το οποίο θα είναι σταθερό αν οι ιδιοτιμές του πίνακα $[\mathbf{I} - a\mathbf{A}]$ είναι μικρότερες κατά απόλυτη τιμή από τη μονάδα. Αν $\{\lambda_1, \lambda_2, \dots, \lambda_n\}$ και $\{z_1, z_2, \dots, z_n\}$ είναι οι ιδιοτιμές και τα ιδιοδιανύσματα αντίστοιχα του Ερσιανού πίνακα, τότε

$$[\mathbf{I} - a\mathbf{A}]z_i = z_i - a\mathbf{A}z_i = z_i - a\lambda_i z_i = (1 - a\lambda_i)z_i.$$

Αυτό συνεπάγεται ότι τα ιδιοδιανύσματα του πίνακα $[\mathbf{I} - a\mathbf{A}]$ θα είναι τα ίδια με τα ιδιοδιανύσματα του πίνακα \mathbf{A} και οι ιδιοτιμές του $[\mathbf{I} - a\mathbf{A}]$ θα είναι $(1 - a\lambda_i)$. Επομένως για να είναι σταθερό το σύστημα 5.21 πρέπει να ισχύει η σχέση

$$|(1 - a\lambda_i)| < 1.$$

Αν τώρα υποθέσουμε ότι η τετραγωνική συνάρτηση έχει ένα ισχυρό ελάχιστο, τότε σύμφωνα με τις ιδιότητες των τετραγωνικών συναρτήσεων που είδαμε οι ιδιοτιμές θα πρέπει να είναι θετικοί αριθμοί, δηλαδή

$$a < \frac{2}{\lambda_i}.$$

Εφόσον αυτό πρέπει να αληθεύει για όλες τις ιδιοτιμές του Ερσιανού πίνακα έχουμε ότι

$$a < \frac{2}{\lambda_{max}}. \quad (5.22)$$

Ρυθμός εκπαίδευσης που ελαχιστοποιεί το δείκτη απόδοσης

Μία διαφορετική μέθοδος για την προσέγγιση του ρυθμού εκπαίδευσης είναι η κατάλληλη επιλογή μιας τιμής τέτοια ώστε να ελαχιστοποιείται ο δείκτης απόδοσης σε κάθε επανάληψη. Δηλαδή η επιλογή ενός ρυθμού εκπαίδευσης a_k που ελαχιστοποιεί τη συνάρτηση

$$F(x_k + a_k \mathbf{p}_k). \quad (5.23)$$

Για τετραγωνικές συναρτήσεις, η παράγωγος της συνάρτησης 5.23 ως προς a_k αποδεικνύεται ότι είναι

$$\frac{d}{da_k} F(x_k + a_k \mathbf{p}_k) = \nabla F(x)^T|_{x=x_k} \mathbf{p}_k + a_k \mathbf{p}_k^T \nabla^2 F(x)|_{x=x_k} \mathbf{p}_k.$$

Αν θέσουμε τη σχέση αυτή ίση με 0 και λύσουμε ως προς a_k έχουμε

$$a_k = -\frac{\nabla F(x)^T|_{x=x_k} \mathbf{p}_k}{\mathbf{p}_k^T \nabla^2 F(x)|_{x=x_k} \mathbf{p}_k} = -\frac{\mathbf{g}_k^T \mathbf{p}_k}{\mathbf{p}_k^T \mathbf{A}_k \mathbf{p}_k},$$

όπου A_k είναι ο Εσσιανός πίνακας ο οποίος υπολογίζεται για $x = x_k$

$$A_k = \nabla^2 F(x)|_{x=x_k}.$$

Παράδειγμα 6.2 Έστω ότι θέλουμε να εφαρμόσουμε τον αλγόριθμο της πιο απότομης καθόδου στη συνάρτηση

$$F(\mathbf{x}) = x_1^2 + 25x_2^2.$$

Αρχικά ορίζουμε ένα τυχαίο αρχικό σημείο έστω το

$$\mathbf{x}_0 = \begin{bmatrix} 2.5 \\ 0.5 \end{bmatrix}.$$

Το πρώτο βήμα είναι να βρούμε το διάνυσμα των μερικών παραγώγων:

$$\nabla F(\mathbf{x}) = \begin{bmatrix} \frac{d}{dx_1} F(\mathbf{x}) \\ \frac{d}{dx_2} F(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} 2x_1 \\ 50x_2 \end{bmatrix}.$$

Αν υπολογίσουμε το διάνυσμα των μερικών παραγώγων για το αρχικό σημείο x_0 έχουμε

$$\mathbf{g}_0 = \nabla F(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_0} = \begin{bmatrix} 5 \\ 25 \end{bmatrix}.$$

Αν υποθέσουμε ότι χρησιμοποιούμε ένα ρυθμό εκπαίδευσης $a = 0.01$, τότε η πρώτη επανάληψη του αλγορίθμου της πιο απότομης καθόδου θα είναι

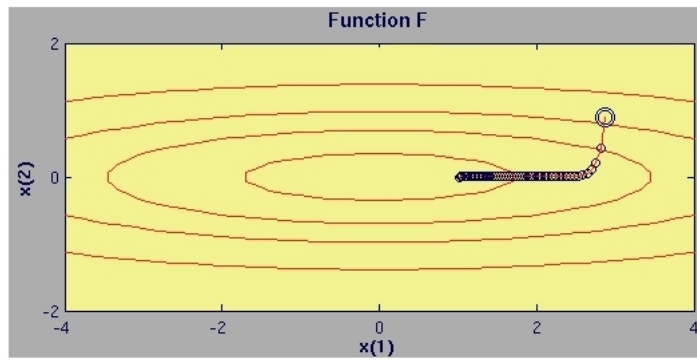
$$\mathbf{x}_1 = \mathbf{x}_0 - a\mathbf{g}_0 = \begin{bmatrix} 2.5 \\ 0.5 \end{bmatrix} - 0.01 \begin{bmatrix} 5 \\ 25 \end{bmatrix} = \begin{bmatrix} 2.45 \\ 0.25 \end{bmatrix}.$$

Με τον ίδιο τρόπο στη δεύτερη επανάληψη θα έχουμε

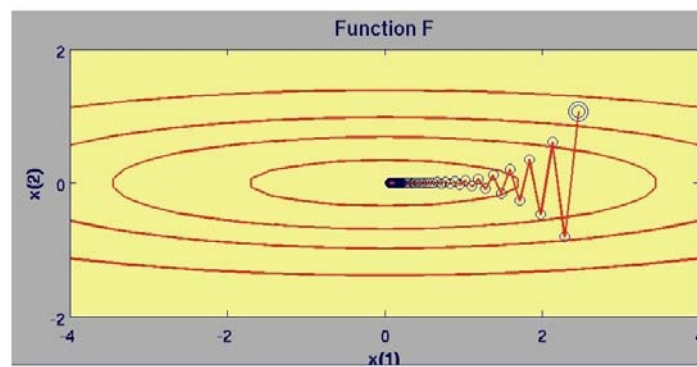
$$\mathbf{x}_2 = \mathbf{x}_1 - a\mathbf{g}_1 = \begin{bmatrix} 2.45 \\ 0.25 \end{bmatrix} - 0.01 \begin{bmatrix} 4.9 \\ 12.5 \end{bmatrix} = \begin{bmatrix} 2.401 \\ 0.125 \end{bmatrix}.$$

Αν συνεχίσουμε τις επαναλήψεις τότε θα έχουμε τη πορεία που φαίνεται στο Σχήμα 5.7

Αν τώρα αλλάξουμε το ρυθμό εκπαίδευσης, έστω $a = 0.035$, παρατηρούμε όπως φαίνεται στο Σχήμα 5.8 ότι αλγόριθμος γίνεται πιο ασταθής. Μεγαλώνοντας το ρυθμός εκπαίδευσης θα περιμέναμε αφού κάνουμε μεγαλύτερα βήματα, ο αλγόριθμος να συγκλίνει και γρηγορότερα. Όπως βλέπουμε όμως και από το παράδειγμα αυτό δε συμβαίνει. Αντίθετα, ο αλγόριθμος γίνεται πιο ασταθής, γι'αυτό είναι απαραίτητο να εκτιμούμε το μέγιστη αποδεκτή τιμή για το ρυθμό εκπαίδευσης.



Σχήμα 5.7: Η μέθοδος της πιο απότομης καθόδου για ρυθμό εκπαίδευσης $a = 0.01$



Σχήμα 5.8: Η μέθοδος της πιο απότομης καθόδου για ρυθμό εκπαίδευσης $a = 0.035$

Επειδή η συνάρτηση $F(\mathbf{x})$ είναι τετραγωνική συνάρτηση μπορούμε να βρούμε το μέγιστο αποδεκτό ρυθμό εκπαίδευσης σύμφωνα με την εξίσωση 5.22. Αρχικά υπολογίζουμε τον Εσσιανό πίνακα αυτής της συνάρτησης

$$\mathbf{A} = \begin{bmatrix} 2 & 0 \\ 0 & 50 \end{bmatrix}.$$

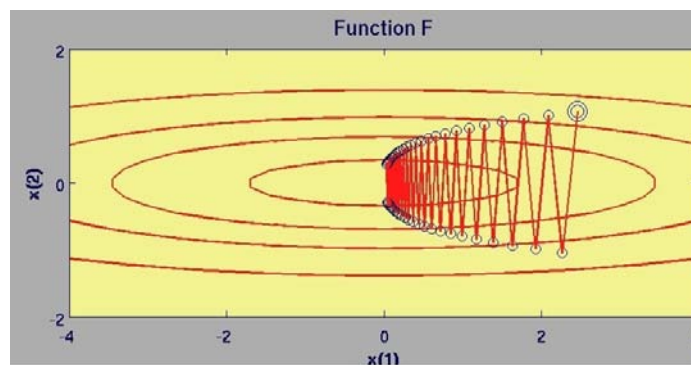
Οι ιδιοτιμές και ιδιοδιανύσματα του πίνακα \mathbf{A} είναι

$$\left\{ (\lambda_1 = 2), \left(z_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right) \right\}, \left\{ (\lambda_2 = 50), \left(z_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right) \right\}.$$

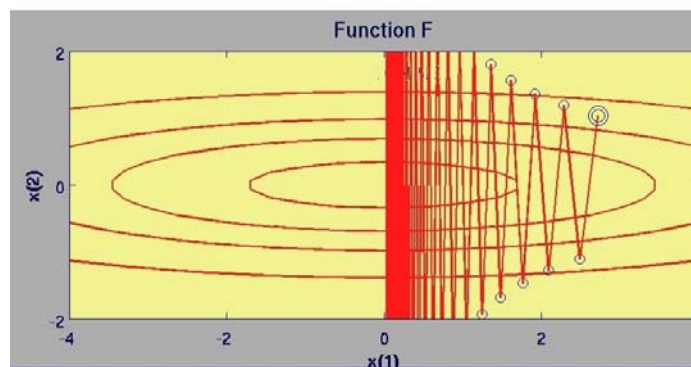
Έπομένως η μέγιστη τιμή του ρυθμού εκπαίδευσης είναι

$$a < \frac{2}{\lambda_{max}} = \frac{2}{50} = 0.04.$$

Στο Σχήμα 5.9 βλέπουμε την πορεία του αλγορίθμου για ρυθμό εκπαίδευσης $a = 0.039$, με τιμή λίγο μικρότερη από τη μέγιστη σταθερή τιμή. Στην περίπτωση αυτή ο αλγόριθμος βρήσκει το ελάχιστο σημείο. Αντίθετα στο Σχήμα 5.10 έχουμε ρυθμό εκπαίδευσης $a = 0.041$, δηλαδή λίγο μεγαλύτερο από τη μέγιστη τιμή, όπου βλέπουμε ότι ο αλγόριθμος ξεπερνάει το ελάχιστο σημείο και δε συγκλίνει.



Σχήμα 5.9: Η μέθοδος της πιο απότομης καθόδου για $a = 0.039$



Σχήμα 5.10: Η μέθοδος της πιο απότομης καθόδου για $a = 0.041$

Επίσης ο αλγόριθμος τείνει να συγκλίνει πιο γρήγορα στην κατεύθυνση του ιδιοδιανύσματος που αντιστοιχεί στη μεγαλύτερη ιδιοτιμή, ενώ αντίθετα συγκλίνει πιο αργά στην κατεύθυνση του ιδιοδιανύσματος που αντιστοιχεί στη μικρότερη ιδιοτιμή. Επομένως όταν υπάρχει μεγάλη διαφορά ανάμεσα στις απόλυτες τιμές της μεγαλύτερης και της μικρότερης ιδιοτιμής τότε ο αλγόριθμος θα συγκλίνει σχετικά αργά.

5.5.2 Η μέθοδος του Newton

Η μέθοδος του Newton σε αντίθεση με τη μέθοδο της πιο απότομης καθόδου, βασίζεται στις δύο πρώτες συνιστώσες της σειράς Taylor

$$F(\mathbf{x}_{k+1}) = F(\mathbf{x}_k + \Delta \mathbf{x}_k) = F(\mathbf{x}_k) + \mathbf{g}_k^T \Delta \mathbf{x}_k + \frac{1}{2} \Delta \mathbf{x}_k^T \mathbf{A}_k \Delta \mathbf{x}_k.$$

Η βασική ιδέα αυτής της μεθόδου είναι να εντοπιστεί το στάσιμο σημείο αυτής της τετραγωνικής συνάρτησης. Αν χρησιμοποιήσουμε την εξίσωση 5.12 για να βρούμε το διάνυσμα των μερικών παραγώγων αυτής της συνάρτησης ως προς $\Delta \mathbf{x}_k$ και θέσουμε την τιμή του ίση με 0 βρίσκουμε

$$\mathbf{g}_k + \mathbf{A}_k \Delta \mathbf{x}_k = 0.$$

Αν λύσουμε ως προς $\Delta \mathbf{x}_k$ τότε έχουμε

$$\Delta \mathbf{x}_k = -\mathbf{A}_k^{-1} \mathbf{g}_k.$$

Συνεπώς η μέθοδος του Newton ορίζεται από την εξίσωση

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{A}_k^{-1} \mathbf{g}_k.$$

Η μέθοδος αυτή βρίσκει πάντα το ελάχιστο μιας τετραγωνικής συνάρτησης σε ένα μόνο βήμα. Αυτό συμβαίνει επειδή η μέθοδος του Newton είναι σχεδιασμένη έτσι ώστε να προσεγγίζει μια οποιαδήποτε συνάρτηση σαν τετραγωνική συνάρτηση και στη συνέχεια να εντοπίζει το στάσιμο σημείο αυτής της προσέγγισης. Αν η αρχική συνάρτηση είναι τετραγωνική με ένα ισχυρό ελάχιστο, τότε θα ελαχιστοποιείται σε ένα βήμα. Αν η συνάρτηση $F(x)$ δεν είναι τετραγωνική, τότε η μέθοδος του Newton δε θα συγκλίνει γενικά σε ένα βήμα, ενώ μπορεί και να μη συγκλίνει ποτέ καθώς αυτό εξαρτάται από τη συνάρτηση και την αρχική τιμή που επιλέξαμε. Επίσης η μέθοδος αυτή δεν μπορεί να διαχωρίσει ένα τοπικό ελάχιστο από ένα ολικό γιατί προσεγγίζει κάθε συνάρτηση σαν τετραγωνική συνάρτηση η οποία μπορεί να έχει ένα μόνο ελάχιστο.

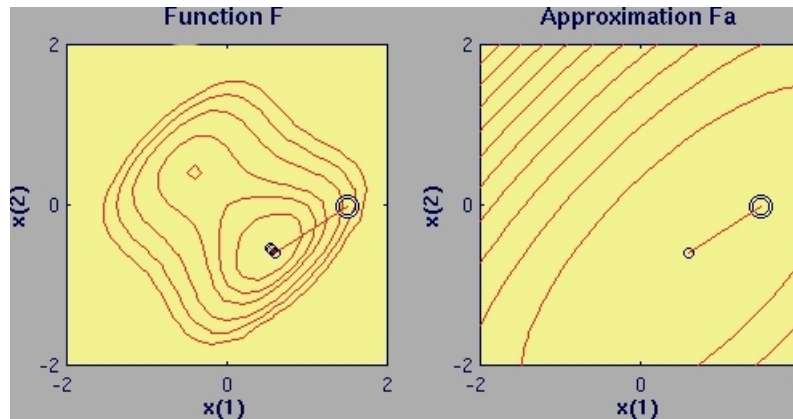
Έστω η συνάρτηση

$$F(\mathbf{x}) = (x_2 - x_1)^4 + 8x_1x_2 - x_1 + x_2 + 3.$$

Γνωρίζουμε ότι η συνάρτηση αυτή έχει τρία στάσιμα σημεία, τα

$$\mathbf{x}^1 = \begin{bmatrix} -0.42 \\ 0.42 \end{bmatrix}, \mathbf{x}^2 = \begin{bmatrix} -0.13 \\ 0.13 \end{bmatrix}, \mathbf{x}^3 = \begin{bmatrix} -0.55 \\ 0.55 \end{bmatrix}.$$

Το \mathbf{x}^1 είναι ισχυρό τοπικό ελάχιστο, το \mathbf{x}^2 είναι ένα σελοειδές σημείο ενώ το \mathbf{x}^3 είναι ισχυρό ολικό ελάχιστο. Αν εφαρμόσουμε το μέθοδο του Newton σε αυτό το πρόβλημα ξεκινώντας από ένα αρχικό σημείο $\mathbf{x}_0 = \begin{bmatrix} 1.5 & 0 \end{bmatrix}^T$ τότε η πορεία του αλγορίθμου φαίνεται στο Σχήμα 5.11. Το διάγραμμα στα αριστερά είναι το διάγραμμα ισοϋψών της συνάρτησης, ενώ στο διάγραμμα στα δεξιά είναι το διάγραμμα ισοϋψών της προσεγγιστικής τετραγωνικής συνάρτησης στο αρχικό σημείο.



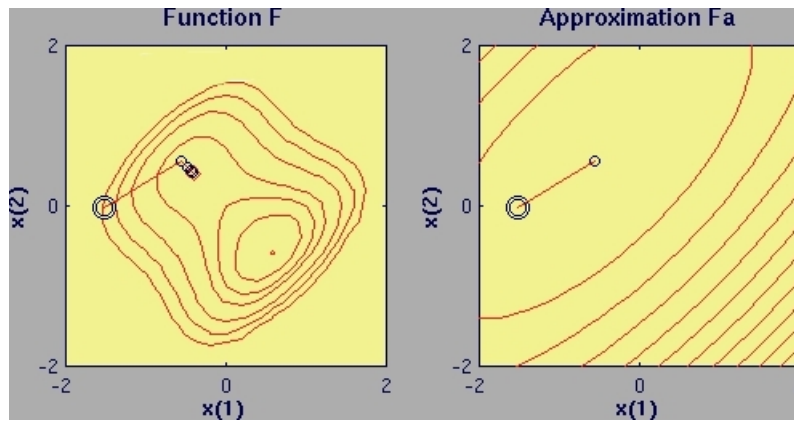
Σχήμα 5.11: Η μέθοδος του Newton για ένα αρχικό σημείο $\mathbf{x}_0 = [1.5 \ 0]^T$

Προφανώς η συνάρτηση δεν ελαχιστοποιείται σε ένα βήμα καθώς δεν είναι τετραγωνική συνάρτηση, όμως από την πρώτη κιάλας επανάληψη η μέθοδος προσεγγίζει το ολικό ελάχιστο. Η μέθοδος του Newton συγκλίνει γρήγορα για πολλές εφαρμογές επειδή οι αναλυτικές συναρτήσεις μπορούν να προσεγγιστούν ακριβώς από τετραγωνικές συναρτήσεις σε μια μικρή γειτονική περιοχή ενός ισχυρού ελάχιστου. Μπορούμε να παρατηρήσουμε ότι το διάγραμμα ισοϋψών της προσεγγιστικής τετραγωνικής συνάρτησης είναι παρόμοιο με το διάγραμμα ισοϋψών της αρχικής συνάρτησης κοντά στο αρχικό σημείο που επιλέγουμε.

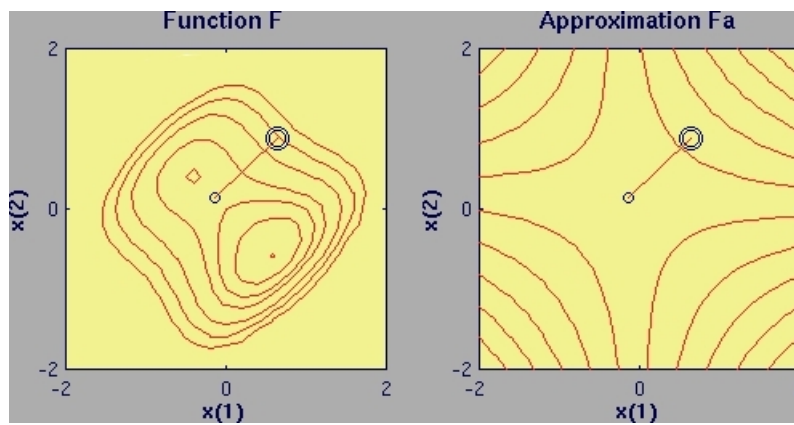
Αντίθετα, αν επιλέξουμε το σημείο $\mathbf{x}_0 = \begin{bmatrix} -1.5 & 0 \end{bmatrix}^T$ σαν αρχικό σημείο, τότε τα βήματα του αλγορίθμου φαίνονται στο Σχήμα 5.12. Στην περίπτωση αυτή ο αλγόριθμος συγκλίνει προς το τοπικό ελάχιστο. Η μέθοδος του Newton όπως και η μέθοδος της πιο απότομης καθόδου βασίζεται στα τοπικά χαρακτηριστικά της επιφάνειας και δεν μπορεί να γνωρίζει τα συνολικά χαρακτηριστικά μιας συνάρτησης. Ομοίως αν επιλέξουμε ως αρχικό σημείο το $\mathbf{x}_0 = \begin{bmatrix} 0.75 & 0.75 \end{bmatrix}^T$, ο αλγόριθμος συγκλίνει στο σελοειδές σημείο της συνάρτησης όπως φαίνεται και στο Σχήμα 5.13. Επομένως, παρατηρούμε ότι η μέθοδος του Newton μπορεί να εντοπίζει τα στάσιμα σημεία μιας συνάρτησης ανάλογα με την αρχική υπόθεση, αλλά δεν μπορεί να κάνει διαχωρισμό ανάμεσα στα ελάχιστα, μέγιστα και σελοειδή σημεία.

Σύγκριση της μεθόδου της πιο απότομης καθόδου και της μεθόδου του Newton

Στο σημείο αυτό μπορούμε να διακρίνουμε κάποιες σημαντικές διαφορές μεταξύ της μεθόδου του Newton και της πιο απότομης καθόδου. Η μέθοδος του Newton συνήθως συγκλίνει πολύ πιο γρήγορα από τη μέθοδο της πιο απότομης καθόδου και για τετραγωνικές συναρτήσεις εντοπίζει το στάσιμο σημείο σε μία μόνο επανάληψη. Από την άλλη πλευρά



Σχήμα 5.12: Η μέθοδος του Newton για ένα αρχικό σημείο $x_0 = [-1.5 \ 0]^T$



Σχήμα 5.13: Η μέθοδος του Newton για ένα αρχικό σημείο $x_0 = [0.75 \ 0.75]^T$

όμως οι ιδιότητες σύγκλισης της μεθόδου του Newton είναι αρκετά πολύπλοκες. Εκτός από το πρόβλημα της σύγκλισης σε σελοειδή σημεία, γεγονός που δε συμβαίνει με τον αλγόριθμο της πιο απότομης καθόδου, επιπρόσθετα ο αλγόριθμος του Newton μπορεί να ταλαντεύεται ή να αποκλίνει. Αντίθετα, με τη μέθοδο της πιο απότομης καθόδου ο αλγόριθμος εγγυάται ότι θα συγκλίνει σε ένα στάσιμο σημείο αν ο ρυθμός εκπαίδευσης δεν είναι πολύ μεγάλος ή αν εφαρμόσουμε γραμμική ελαχιστοποίηση σε κάθε βήμα.

Ένα ακόμη μειονέκτημα της μεθόδου του Newton είναι ότι απαιτεί τον υπολογισμό και την αποθήκευση του Εσσιανού πίνακα και του αντίστροφού του. Αντίθετα η μέθοδος της πιο απότομης καθόδου είναι πολύ απλή και απαιτεί μόνο τον υπολογισμό του διανύσματος των μερικών παραγώγων. Υπάρχουν διάφορες παραλλαγές της μεθόδου του Newton με τις οποίες μπορούν να ξεπεραστούν αυτά τα προβλήματα. Μία από αυτές είναι ο συνδυασμός των δύο μεθόδων την οποία θα δούμε σε επόμενη ενότητα.

5.5.3 Η μέθοδος των συζυγών κλίσεων

Η μέθοδος του Newton έχει την ιδιότητα να ελαχιστοποιεί μια τετραγωνική συνάρτηση σε έναν πεπερασμένο αριθμό επαναλήψεων, απαιτεί όμως τον υπολογισμό και την αποθήκευση του Εσσιανού πίνακα κάτι που μπορεί να μην είναι πρακτικό αν ο αριθμός των αγνώστων είναι πολύ μεγάλος. Ειδικά σε πρακτικές εφαρμογές των Τεχνητών Νευρωνικών Δικτύων μπορεί να απαιτούνται χιλιάδες βάρη. Σε αυτήν την περίπτωση θέλουμε μια μέθοδο για την οποία να απαιτείται ο υπολογισμός μόνο των πρώτων παραγώγων αλλά ταυτόχρονα να έχει την ιδιότητα να ελαχιστοποιεί μια τετραγωνική συνάρτηση σε έναν πεπερασμένο αριθμό επαναλήψεων. Μια τέτοια μέθοδος είναι η μέθοδος των συζυγών κλίσεων (Conjugate Gradient).

Η μέθοδος των συζυγών κλίσεων συνοψίζεται στα παρακάτω βήματα:

1. Επιλέγουμε την κατεύθυνση προς την οποία θα αναζητήσουμε το ελάχιστο σημείο. Η κατεύθυνση αυτή ορίζεται ως το αντίθετο του διανύσματος των μερικών παραγώγων.

$$\mathbf{p}_0 = -\mathbf{g}_0 \quad (5.24)$$

2. Κάνουμε ένα βήμα σύμφωνα με την εξίσωση

$$\Delta \mathbf{x}_k = a_k \mathbf{p}_k, \quad (5.25)$$

όπου a_k είναι ο ρυθμός εκπαίδευσης ο οποίος επιλέγεται έτσι ώστε να ελαχιστοποιεί τη συνάρτηση κατά μήκος της κατεύθυνσης αναζήτησης

$$\mathbf{x}_{k+1} = \mathbf{x}_k + a_k \mathbf{p}_k. \quad (5.26)$$

Για τετραγωνικές συναρτήσεις ο ρυθμός εκπαίδευσης υπολογίζεται σύμφωνα με την εξίσωση

$$a_k = -\frac{\nabla F(\mathbf{x})^T|_{\mathbf{x}=\mathbf{x}_k} \mathbf{p}_k}{\mathbf{p}_k^T \nabla^2 F(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_k} \mathbf{p}_k} = -\frac{\mathbf{g}_k^T \mathbf{p}_k}{\mathbf{p}_k^T \mathbf{A}_k \mathbf{p}_k}. \quad (5.27)$$

3. Στη συνέχεια επιλέγουμε την επόμενη κατεύθυνση της αναζήτησης με την εξίσωση

$$\mathbf{p}_k = -\mathbf{g}_k + \beta_k \mathbf{p}_{k-1}. \quad (5.28)$$

Υπολογίζουμε τη μεταβλητή β_k με κάποια μέθοδο, όπως οι τρεις που ακολουθούν:

- Μέθοδος Hestenes και Steifel

$$\beta_k = \frac{\Delta \mathbf{g}_{k-1}^T \mathbf{g}_k}{\Delta \mathbf{g}_{k-1}^T \mathbf{p}_{k-1}} \quad (5.29)$$

- Μέθοδος Fletcher και Reeves

$$\beta_k = \frac{\mathbf{g}_k^T \mathbf{g}_k}{\mathbf{g}_{k-1}^T \mathbf{g}_{k-1}} \quad (5.30)$$

- Μέθοδος Polak και Ribiere

$$\beta_k = \frac{\Delta \mathbf{g}_{k-1}^T \mathbf{g}_k}{\mathbf{g}_{k-1}^T \mathbf{g}_{k-1}} \quad (5.31)$$

4. Αν ο αλγόριθμος δεν έχει συγκλίνει τότε επιστρέφουμε στο βήμα 2.

Κεφάλαιο 6

Το δίκτυο ADALINE

Ο Bernard Widrow άρχισε την έρευνα πάνω στα νευρωνικά δίκτυα στα τέλη του 1950, την ίδια χρονική περίοδο που ο Frank Rosenblatt ανέπτυξε τον κανόνα εκπαίδευσης Perceptron. Το 1960 ο Widrow και ένας τελειόφοιτος μαθητής του ο Marcian Hoff παρουσίασαν το δίκτυο ADALINE (ADaptive Linear Neuron) και έναν κανόνα εκπαίδευσης τον οποίο ονόμασαν αλγόριθμο LMS (Least Mean Square) [39]. Το δίκτυο αυτό έχει πολλές ομοιότητες με το Perceptron με τη διαφορά ότι η συνάρτηση μεταφοράς του δικτύου είναι η γραμμική. Το δίκτυο ADALINE έχει τους ίδιους περιορισμούς με το Perceptron, δεν μπορεί δηλαδή να λύσει μη γραμμικά διαχωρίσιμα προβλήματα.

Παρόλα αυτά ο αλγόριθμος LMS είναι πιο ισχυρός από τον κανόνα Perceptron. Ενώ το Perceptron εγγυάται ότι θα συγκλίνει σε μία λύση που κατηγοριοποιεί σωστά τα δεδομένα εκπαίδευσης, το δίκτυο που προκύπτει μπορεί να είναι ευαίσθητο στο θόρυβο καθώς τα πρότυπα συχνά είναι πολύ κοντά στη γραμμή που διαχωρίζει τα πρότυπα. Ο αλγόριθμος LMS ελαχιστοποιεί το μέσο τετραγωνικό σφάλμα και με αυτό τον τρόπο προσπαθεί να μετακινήσει τη διαχωριστική γραμμή όσο το δυνατόν πιο μακριά από τα πρότυπα εκπαίδευσης.

Ο αλγόριθμος LMS έχει βρει πολλές εφαρμογές ειδικότερα στο πεδίο της ψηφιακής επεξεργασίας σημάτων. Για παράδειγμα οι περισσότερες τηλεφωνικές γραμμές χρησιμοποιούν δίκτυα ADALINE για τη μείωση της ηχούς.

Το δίκτυο ADALINE φαίνεται στο Σχήμα 6.1. Η έξοδος του δικτύου δίνεται από τη σχέση

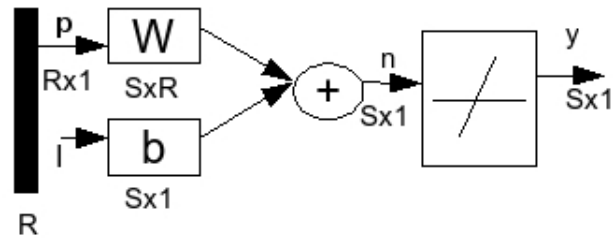
$$y = f(Wp + b) = Wp + b,$$

όπου f είναι η γραμμική συνάρτηση η οποία δίνεται από τον τύπο $f(s) = s$.

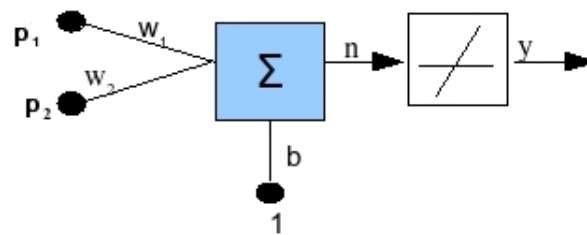
Όπως και πριν για το Perceptron για να μελετήσουμε το δίκτυο ADALINE θεωρούμε την απλή περίπτωση ενός δικτύου με 2 εισόδους όπως φαίνεται στο Σχήμα 6.2. Η έξοδος του δικτύου δίνεται από τη σχέση:

$$y = w_1p_1 + w_2p_2 + b.$$

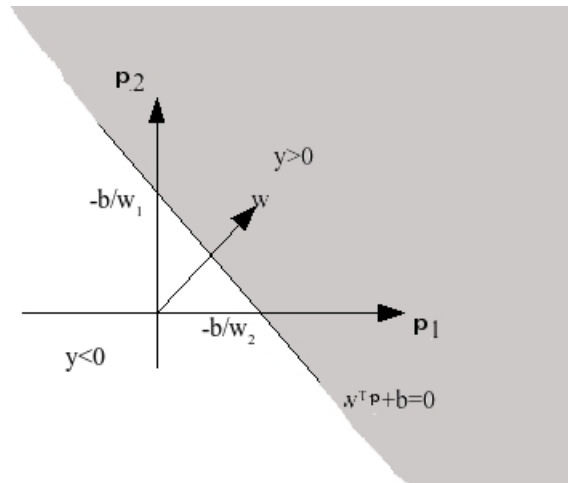
Αν θέσουμε την είσοδο του δικτύου $n = 0$ τότε η σχέση $w_1p_1 + w_2p_2 + b = 0$ ορίζει μια διαχωριστική γραμμή όπως φαίνεται στο Σχήμα 6.3.



Σχήμα 6.1: Δίκτυο ADALINE.



Σχήμα 6.2: Απλό δίκτυο ADALINE με δύο εισόδους.



Σχήμα 6.3: Διαχωριστική γραμμή για ένα δίκτυο ADALINE δύο εισόδων.

Στη σκιασμένη περιοχή η έξοδος του δικτύου είναι μεγαλύτερη από το 0, ενώ για την υπόλοιπη περιοχή η έξοδος είναι μικρότερη του 0. Όμοια με το Perceptron το δίκτυο ADALINE χρησιμοποιείται για να ταξινομήσει αντικείμενα σε δύο διαφορετικές κατηγορίες με την προϋπόθεση ότι τα αντικείμενα αυτά είναι γραμμικά διαχωρίσιμα.

6.1 Αλγόριθμος LMS

Ο αλγόριθμος LMS αποτελεί κι αυτός παράδειγμα εκπαίδευσης με επίβλεψη. Το δίκτυο τροφοδοτείται με ένα σύνολο προτύπων εκπαίδευσης $\{p_1, t_1\}, \{p_2, t_2\}, \dots, \{p_n, t_n\}$, όπου p_n είναι η είσοδος που θα εφαρμοστεί στο δίκτυο και t_n είναι η επιθυμητή έξοδος ή στόχος που αντιστοιχεί στην είσοδο αυτή. Κάθε φορά που ένα διάνυσμα εισόδου από τα δεδομένα εκπαίδευσης τροφοδοτείται στο δίκτυο, η έξοδος του δικτύου συγκρίνεται με τον επιθυμητό στόχο. Ο αλγόριθμος LMS χρησιμοποιείται για να προσαρμόσει τα βάρη και τις πόλωσης του δικτύου έτσι ώστε να ελαχιστοποιεί το μέσο τετραγωνικό σφάλμα, όπου το σφάλμα υπολογίζεται από τη διαφορά της εξόδου του δικτύου από τον επιθυμητό στόχο.

Το μέσο τετραγωνικό σφάλμα υπολογίζεται από τη σχέση

$$F(x) = E[e^2] = E[(t - y)^2] = E[t - \sum (w_i p_i + b)] \quad (6.1)$$

Για να απλοποιήσουμε τη συνάρτηση θεωρούμε ότι η πόλωση συμπεριλαμβάνεται στο διάνυσμα των βαρών δηλαδή

$$\mathbf{r} = \begin{pmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,R} & b_1 \\ w_{2,1} & w_{2,2} & \dots & w_{2,R} & b_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ w_{s,1} & w_{s,2} & \dots & w_{s,R} & b_s \end{pmatrix}.$$

Ομοίως συμπεριλαμβάνουμε την είσοδο που αντιστοιχεί στην πόλωση και είναι ίση με 1, στο

διάνυσμα των εισόδων οπότε έχουμε

$$\mathbf{z}^T = \begin{bmatrix} p_1 & p_2 & \dots & p_R & 1 \end{bmatrix}.$$

Τελικά είναι:

$$\begin{aligned} F(x) &= E[(t - r^T z)^2] \\ &= E[t^2 - 2tr^T z + r^T z z^T p] \\ &= E[t^2] - 2r^T E[tz] + r^T E[zz^T]r. \end{aligned} \quad (6.2)$$

Η εξίσωση 6.2 μπορεί να γραφεί στη μορφή

$$F(x) = c - 2r^T h + r^T R p, \quad (6.3)$$

όπου $c = E[t^2]$, $h = E[tz]$ και $R = E[zz^T]r$.

Ο πίνακας R ονομάζεται πίνακας συσχέτισης εισόδου. Τα διαγώνια στοιχεία αυτού του πίνακα είναι ίσα με τις μέσες τετραγωνικές τιμές των στοιχείων των διανυσμάτων εισόδου.

Αν θεωρήσουμε ότι η συνάρτηση 6.3 είναι μια τετραγωνική συνάρτηση, τότε γράφεται με τη μορφή

$$F(x) = c + d^T x + \frac{1}{2} x^T A x. x^* = R^{-1} h. \quad (6.4)$$

Επομένως ο δείκτης απόδοσης για το μέσο τετραγωνικό σφάλμα για το δίκτυο ADALINE είναι μια τετραγωνική συνάρτηση με $d = -2h$ και $A = 2R$. Στην περίπτωση αυτή ο Εσσιανός πίνακας είναι ίσος με το διπλάσιο του πίνακα συσχέτισης. Γνωρίζουμε ότι όλοι οι πίνακες συσχέτισης είναι θετικά ορισμένοι ή θετικά ημιορισμένοι. Αν ο πίνακας συσχέτισης έχει μόνο θετικές ιδιοτιμές, ο δείκτης απόδοσης θα έχει ένα μοναδικό ολικό ελάχιστο, ενώ αν έχει και μηδενικές ιδιοτιμές, τότε ο δείκτης απόδοσης είτε θα έχει ένα ασθενές ελάχιστο ή δεν θα έχει κανένα ελάχιστο (ανάλογα με τον πίνακα $d = -2h$).

Αν υπολογίσουμε το διάνυσμα των μερικών παραγώγων για το δείκτη απόδοσης έχουμε

$$\nabla F(x) = \nabla(c + d^T x + \frac{1}{2} x^T A x) = d + A x = -2h + 2R x. \quad (6.5)$$

Για να βρούμε το στάσιμο σημείο του δείκτη απόδοσης θέτουμε το διάνυσμα των μερικών παραγώγων ίσο με 0.

$$-2h + 2R x = 0. \quad (6.6)$$

Έτσι αποδεικνύεται ότι αν ο πίνακας R είναι θετικά ορισμένος θα υπάρχει ένα μοναδικό σημείο το οποίο θα είναι ισχυρό ελάχιστο και υπολογίζεται από τη σχέση

$$x^* = R^{-1} h. \quad (6.7)$$

Ο αλγόριθμος LMS είναι ένας αλγόριθμος ο οποίος υπολογίζει το ελάχιστο σημείο. Αν μπορούμε να υπολογίσουμε τους πίνακες h και R , τότε μπορούμε να βρούμε απευθείας το ελάχιστο σημείο από την εξίσωση 6.7. Αν όμως δε θέλουμε να υπολογίσουμε τον αντίστροφο του πίνακα R , τότε η εύρεση του ελαχίστου μπορεί να γίνει με τη μέθοδο της πιο απότομης καθόδου.

Γνωρίζουμε από την εξίσωση 5.20 ότι ο αλγόριθμος της πιο απότομης καθόδου με σταθερό ρυθμό εκπαίδευσης είναι

$$\mathbf{x}_{k+1} = \mathbf{x}_k - a \nabla F(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_k}. \quad (6.8)$$

Το διάνυσμα των μερικών παραγώγων $\nabla F(\mathbf{x})$ δίνεται από την εξίσωση

$$\nabla F(\mathbf{x}) = \nabla e^2(k) = -2e(k)\mathbf{z}(k), \quad (6.9)$$

όπου $e(k)$ είναι το σφάλμα της εξόδου στην επανάληψη k , δηλαδή $e(k) = t(k) - a(k)$. Άρα είναι:

$$w(k+1) = w(k) + 2ae(k)\mathbf{p}(k), \quad (6.10)$$

και

$$b(k+1) = b(k) + 2ae(k). \quad (6.11)$$

Οι εξισώσεις 6.10 και 6.11 αποτελούν τον αλγόριθμο του ελάχιστου μέσου τετραγωνικού σφάλματος (LMS) ο οποίος αποκαλείται επίσης κανόνας δέλτα ή αλγόριθμος εκπαίδευσης των Widrow-Hoff.

Αν έχουμε i νευρώνες εξόδου τότε θα είναι:

$$w_i(k+1) = w_i(k) + 2ae_i(k)\mathbf{p}(k), \quad (6.12)$$

και

$$b_i(k+1) = b_i(k) + 2ae_i(k), \quad (6.13)$$

όπου $e_i(k)$ είναι το i στοιχείο του σφάλματος στην επανάληψη k . Επομένως ο αλγόριθμος LMS μπορεί να γραφεί σε μορφή πίνακα ως εξής:

$$\mathbf{W}(k+1) = \mathbf{W}(k) + 2ae(k)\mathbf{p}^T(k), \quad (6.14)$$

και

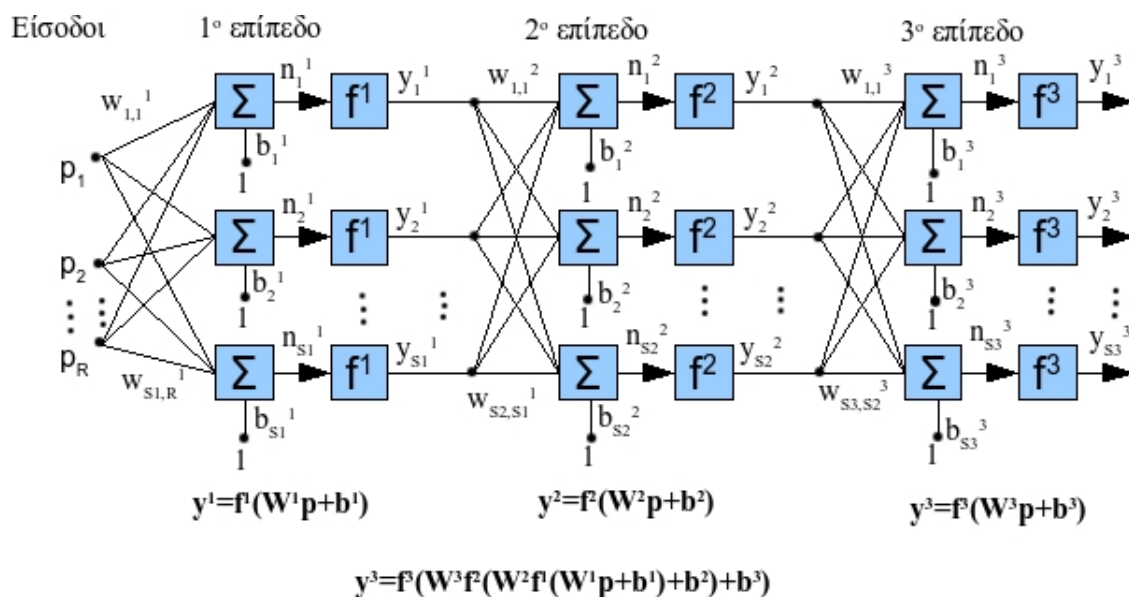
$$\mathbf{b}(k+1) = \mathbf{b}(k) + 2ae(k). \quad (6.15)$$

Κεφάλαιο 7

Πολυστρωματικά Τεχνητά Νευρωνικά Δίκτυα

Τα πολυστρωματικά δίκτυα χαρακτηρίζονται από την ύπαρξη τουλάχιστον ενός κρυφού επιπέδου του οποίου οι κόμβοι υπολογισμών, δηλαδή οι νευρώνες του, ονομάζονται αντίστοιχα κρυφοί νευρώνες. Οι κρυφοί νευρώνες παρεμβάλλονται μεταξύ της εισόδου και της εξόδου του δικτύου. Προσθέτοντας ένα ή περισσότερα κρυφά επίπεδα το δίκτυο είναι ικανό να εκτελεί υψηλότερης τάξης υπολογισμούς και να εξάγει πιο σύνθετα στατιστικά αποτελέσματα.

Ένα τέτοιο δίκτυο πολλαπλών επιπέδων φαίνεται στο Σχήμα 7.1. Ως συντομογραφία ενός πολυστρωματικού νευρωνικού δικτύου συχνά χρησιμοποιείται η εξής: $R-S^1-S^2-\dots-S^m$, όπου p είναι ο αριθμός των εισόδων και m ο αριθμός των επιπέδων του δικτύου με S^1 κόμβους το πρώτο, S^2 κόμβους το δεύτερο και S^m το τελευταίο.



Σχήμα 7.1: Ένα πλήρως συνδεδεμένο πολυστρωματικό δίκτυο πρόσθιας τροφοδότησης.

Πρώτα, υπάρχει ένα επίπεδο εισόδου το οποίο αποτελείται από μία ομάδα νευρώνων οι οποίοι δεν κάνουν ουσιαστικά τίποτα άλλο παρά να δέχονται το σήμα εισόδου. Κατόπιν υπάρχει ένας αριθμός κρυφών επιπέδων, καθένα από τα οποία έχει έναν αριθμό νευρώνων, και τα οποία δέχονται το σήμα από το επίπεδο εισόδου, το επεξεργάζονται και κατόπιν το προωθούν προς την έξοδο. Στο Σχήμα 7.1 υπάρχουν δύο κρυφά επίπεδα, αλλά θα μπορούσε να ήταν οποιοσδήποτε άλλος αριθμός επιπέδων. Τέλος, υπάρχει ένα επίπεδο εξόδου που έχει επίσης έναν αριθμό νευρώνων, οι οποίοι δέχονται σήμα από τα εσωτερικά επίπεδα και το προωθούν προς την έξοδο του δικτύου. Η έξοδος του δικτύου του Σχήματος 7.1 δίνεται από την εξίσωση

$$y^3 = f^3(W^3 f^2(W^2 f^1(W^1 p + b^1) + b^2) + b^3).$$

Προφανώς, η πολυπλοκότητα ενός τέτοιου δικτύου μπορεί να μεταβληθεί ραγδαία μεταβάλλοντας τόσο τον αριθμό των επιπέδων, όσο και τον αριθμό των νευρώνων σε κάθε επίπεδο. Γενικά, δεν υπάρχει κανόνας ως προς τον αριθμό τόσο των κρυφών επιπέδων όσο και ως προς τον αριθμό των νευρώνων που περιλαμβάνει κάθε επίπεδο. Το πλήθος των επιπέδων και των νευρώνων διαφέρει σε κάθε πρόβλημα. Ο αριθμός των νευρώνων στην είσοδο και την έξοδο παρέχεται έμμεσα από τα δεδομένα του προβλήματος. Έχει αποδειχθεί ότι τέτοια νευρωνικά δίκτυα, δοθέντος ενός ικανοποιητικού αριθμού κρυφών νευρώνων και δεδομένων εκπαίδευσης, μπορούν να προσεγγίσουν οποιαδήποτε συνάρτηση σε οποιοδήποτε βαθμό ακρίβειας. Δημιουργούν με αυτόν τον τρόπο μία συσχέτιση μεταξύ εισόδου και εξόδου.

Τα πολυστρωματικά νευρωνικά δίκτυα Perceptron χρησιμοποιούνται σε ποικίλες, αλλά και δύσκολες εφαρμογές και συνήθως εκπαιδεύονται βάσει του αλγορίθμου της όπισθεν διάδοσης σφάλματος (Error Back-Propagation). Η διαδικασία ενός κύκλου εκπαίδευσης περιλαμβάνει δύο στάδια. Αρχικά εισάγονται στην είσοδο δεδομένα από κάποιο διάνυσμα εκπαίδευσης, οπότε οι νευρώνες στο επίπεδο εισόδου παράγουν αποτέλεσμα το οποίο με τη σειρά του αποτελεί είσοδο για το επόμενο, κρυφό επίπεδο. Η διαδικασία αυτή, επαναλαμβάνεται διαδοχικά για τα επόμενα επίπεδα, μέχρι το επίπεδο εξόδου και συνολικά ονομάζεται πρόσθιο πέρασμα. Το δίκτυο ξεκινά υπολογισμούς με τυχαίες τιμές βαρών οι οποίες αναπροσαρμόζονται έτσι ώστε να περιοριστεί το σφάλμα στην έξοδο του δικτύου. Αρχικά υπολογίζονται οι νέες τιμές των βαρών που συνδέουν το επίπεδο εξόδου με το προηγούμενο κρυφό επίπεδο, στη συνέχεια υπολογίζονται τα βάρη που συνδέουν αυτό το κρυφό επίπεδο με το προηγούμενο μέχρι να υπολογιστούν και τα βάρη μεταξύ επιπέδου εισόδου και πρώτου κρυφού επιπέδου. Αυτό το στάδιο αναπροσαρμογής των βαρών ονομάζεται ανάστροφο πέρασμα ή ανάστροφη μετάδοση.

7.1 Το πρόβλημα της αποκλειστικής διάζευξης

Ένα από τα πιο γνωστά προβλήματα που επιλύονται με νευρωνικά δίκτυα είναι αυτό της εκμάθησης της συνάρτησης X-OR (eXclusive-OR), δηλαδή της συνάρτησης της αποκλειστικής διάζευξης. Η συνάρτηση αυτή δέχεται δύο εισόδους και δίνει μία έξοδο. Οι εισοδοί και η έξοδος μπορεί να είναι 0 ή 1 μόνον και ισχύει ο εξής περιορισμός: Εάν και οι δύο εισοδοί

είναι ίδιες, τότε η έξοδος είναι 0, εάν είναι διαφορετικές, τότε η έξοδος είναι 1. Οι όροι αυτοί συνοψίζονται στον Πίνακα 7.1, που ονομάζεται και «πίνακας αληθείας» της συνάρτησης.

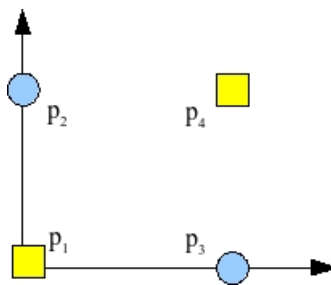
Πίνακας 7.1: Πίνακας αληθείας της συνάρτησης XOR

Είσοδος 1	Είσοδος 2	Έξοδος
0	0	0
0	1	1
1	0	1
1	1	0

Τα δεδομένα εισόδου δηλαδή και οι επιθυμητοί στόχοι για μία πύλη XOR είναι:

$$\left\{ p_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, t_1 = 0 \right\}, \left\{ p_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, t_2 = 1 \right\}, \left\{ p_3 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, t_3 = 1 \right\}, \left\{ p_4 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, t_4 = 0 \right\}.$$

Το πρόβλημα αυτό χρησιμοποιήθηκε από τους Minsky και Papert το 1969 για να αναδείξουν τους περιορισμούς ενός μονοστρωματικού δικτύου perceptron και αναπαρίσταται γραφικά στο Σχήμα 7.2.



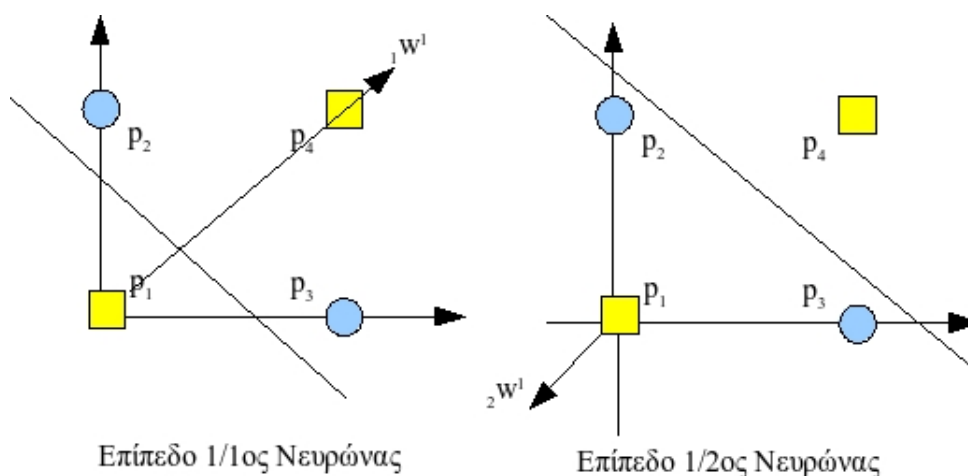
Σχήμα 7.2: Γραφική αναπαράσταση του προβλήματος XOR

Έστω ότι χρησιμοποιούμε ένα απλό νευρωνικό δίκτυο perceptron με δύο εισόδους. Αρχικά αναγνωρίζουμε ότι η χρήση ενός απλού νευρώνα με δυο εισόδους έχει ως αποτέλεσμα μια ευθεία γραμμή για όριο απόφασης στον χώρο εισόδου. Για όλα τα σημεία στη μια πλευρά αυτής της γραμμής ο νευρώνας δίνει έξοδο 1, ενώ για όλα τα σημεία στην άλλη πλευρά της γραμμής δίνει έξοδο 0. Η θέση και ο προσανατολισμός της γραμμής στον χώρο εισόδου καθορίζονται από τα συναπτικά βάρη του νευρώνα με τα οποία είναι συνδεδεμένος με τους κόμβους εισόδου, και το κατώφλι που εφαρμόζεται στο νευρώνα. Με τα πρότυπα εισόδου (0,0) και (1,1) τοποθετημένα σε αντίθετες γωνίες του μοναδιαίου τετραγώνου και παρομοίως για τα άλλα δυο πρότυπα εισόδου (0,1) και (1,0), είναι προφανές ότι δεν μπορούμε να κατασκευάσουμε μια ευθεία γραμμή για ένα όριο απόφασης έτσι ώστε (0,0) και (1,1) να βρίσκονται στην μια περιοχή απόφασης και (0,1) και (1,0) να βρίσκονται στην άλλη περιοχή απόφασης. Με άλλα λόγια ένα στοιχειώδες δίκτυο perceptron δεν μπορεί να επιλύσει το πρόβλημα της αποκλειστικής διάζευξης.

Υπάρχουν πολλές συναρτήσεις, παρόμοιες με την συνάρτηση XOR, οι οποίες δεν μπορούν να παρασταθούν με ένα δίκτυο perceptron. Για όλες αυτές τις συναρτήσεις λέμε ότι

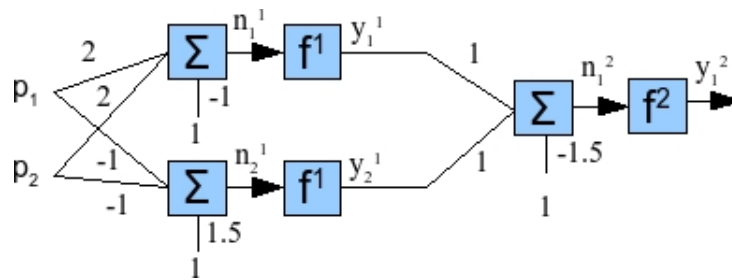
είναι γραμμικά μη διαχωρίσιμες. Είδαμε ότι στην περίπτωση που έχουμε δύο εισόδους, τότε ο διαχωρισμός γίνεται από μία ευθεία γραμμή. Αν το πρόβλημά μας είχε τρεις εισόδους, τότε ο διαχωρισμός θα γινόταν από ένα επίπεδο που θα έτεμνε τον τρισδιάστατο χώρο. Για την περίπτωση τεσσάρων και πάνω εισόδων, πρέπει να δημιουργήσουμε έναν υπερχώρο n διαστάσεων που θα τέμνεται από ένα υπερεπίπεδο, όπου υπερεπίπεδο θεωρούμε ένα γεωμετρικό σχήμα που διαιρεί τον χώρο σε τέσσερις ή παραπάνω διαστάσεις. Δεν υπάρχει κανένας απλός τρόπος να ξέρουμε εκ των προτέρων εάν η συνάρτηση που μας παρουσιάζεται είναι γραμμικά διαχωρίσιμη, ειδικά όταν ο αριθμός των μεταβλητών είναι μεγάλος. Ένας νευρώνας με n εισόδους μπορεί να έχει 2^n διαφορετικούς συνδυασμούς από 0 και 1. Αφού κάθε συνδυασμός μπορεί να δώσει δύο διαφορετικές εξόδους (0 ή 1), υπάρχουν 2^{2^n} διαφορετικές συναρτήσεις n μεταβλητών. Καταλαβαίνουμε, λοιπόν, ότι η πιθανότητα να είναι μία συνάρτηση γραμμικά διαχωρίσιμη είναι πολύ μικρή, όταν μάλιστα υπάρχουν πολλές εισοδοί.

Το πρόβλημα της αποκλειστικής διάζευξης μπορεί να λυθεί αν χρησιμοποιήσουμε ένα νευρωνικό δίκτυο δύο στρωμάτων. Με τη χρήση πολυστρωματικών δικτύων το πρόβλημα μπορεί να λυθεί με πολλούς διαφορετικούς τρόπους. Μία προσέγγιση είναι να χρησιμοποιήσουμε δύο νευρώνες στο πρώτο επίπεδο για τη δημιουργία δύο ορίων απόφασης. Το πρώτο όριο χωρίζει το πρότυπο p_1 από τα υπόλοιπα πρότυπα, ενώ το δεύτερο όριο χωρίζει το πρότυπο p_4 . Το δεύτερο επίπεδο του δικτύου μπορεί να συνδυάσει τα δύο όρια απόφασης χρησιμοποιώντας τη λειτουργία της σύζευξης (AND). Τα όρια απόφασης για κάθε νευρώνα του πρώτου επιπέδου φαίνονται στο Σχήμα 7.3.

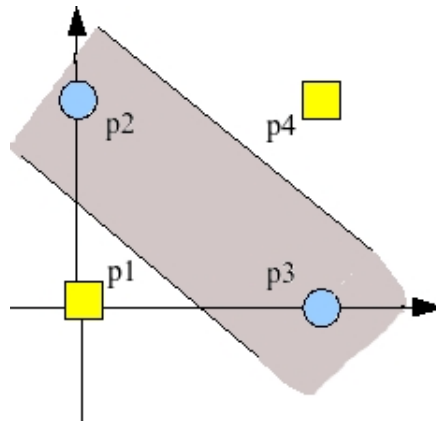


Σχήμα 7.3: Όρια απόφασης για κάθε νευρώνα του πρώτου επιπέδου στο πρόβλημα XOR

Το δίκτυο που χρησιμοποιούμε φαίνεται στο Σχήμα 7.4. Η συνάρτηση ενεργοποίησης του δικτύου αυτού σε κάθε επίπεδο, f^1 και f^2 είναι η βηματική συνάρτηση και τα όρια απόφασης που προκύπτουν από αυτό το δίκτυο παρουσιάζονται στο Σχήμα 7.5. Η σκιασμένη περιοχή δείχνει εκείνες τις εισόδους που παράγουν έξοδο 1 όταν τροφοδοτηθούν στο δίκτυο.



Σχήμα 7.4: Πολυστρωματικό νευρωνικό δίκτυο για το πρόβλημα XOR



Σχήμα 7.5: Τα όρια απόφασης έτσι όπως διαμορφώνονται τελικά για το πρόβλημα XOR

7.2 Η μέθοδος της όπισθεν διάδοσης σφάλματος

Για πολλά χρόνια δεν υπήρχε τρόπος εκπαίδευσης δικτύων με πολλά επίπεδα και οι γνώσεις μας περιορίζονταν σε δίκτυα με ένα ή δύο επίπεδα μόνο. Φυσικά, τα προβλήματα που μπορούσαν να λύσουν τα απλά δίκτυα ήταν λίγα και οι περιορισμοί που υπεισέρχονταν τα καθιστούσαν όχι και τόσο χρήσιμα. Έτσι, έγινε γρήγορα αντιληπτό ότι ήταν απαραίτητο να βρούμε ένα τρόπο να εκπαιδεύσουμε νευρωνικά δίκτυα που μπορούν να αναπαραστήσουν πιο περίπλοκες διεργασίες.

Το κενό αυτό ήρθε να καλύψει η μέθοδος εκπαίδευσης της όπισθεν διάδοσης σφάλματος που αναπτύχθηκε και έγινε ευρύτατα γνωστή. Ως τεχνική η μέθοδος αυτή βασίζεται σε καθαρά μαθηματική θεώρηση με αυστηρά τεκμηριωμένες αποδείξεις. Το νευρωνικό δίκτυο στο οποίο εφαρμόζεται είναι αρκετά πιο περίπλοκο από το μοντέλο του απλού Perceptron. Είναι ένα δίκτυο πολλαπλών επιπέδων και κάθε επίπεδο μπορεί να έχει πολλούς νευρώνες. Οι νευρώνες μέσα στο ίδιο επίπεδο δεν συνδέονται μεταξύ τους, αλλά οι νευρώνες που ανήκουν σε διαφορετικά επίπεδα συνδέονται ως συνήθως με τις γνωστές συνάψεις.

Η κεντρική ιδέα της δομής και λειτουργίας τέτοιων δικτύων είναι σχετικά απλή. Το δίκτυο ξεκινά την διαδικασία μάθησης από τυχαίες τιμές των βαρών του. Εάν δώσει λάθος απάντηση, τότε τα βάρη διορθώνονται έτσι ώστε το σφάλμα να γίνει μικρότερο. Η ίδια διαδικασία επαναλαμβάνεται πολλές φορές έτσι ώστε σταδιακά το σφάλμα ελαττώνεται μέχρις ότου γίνει πολύ μικρό και ανεκτό. Στο σημείο αυτό λέμε ότι το δίκτυο έχει μάθει τα παραδείγματα που

του διδάξαμε με την ακρίβεια που θέλαμε να μάθει.

Η διαδικασία εκπαίδευσης κατά τον αλγόριθμο της όπισθεν διάδοσης σφάλματος λαμβάνει χώρα με την παρουσίαση και εφαρμογή στο νευρωνικό δίκτυο ενός συνόλου παραδειγμάτων εκπαίδευσης. Η παρουσίαση όλων των προτύπων εκπαίδευσης μια φορά το καθένα στο δίκτυο ονομάζεται εποχή (epoch). Κατά τη συνολική διαδικασία εκπαίδευσης εκτελούνται επαναλήψεις των εποχών, ώσπου τα βάρη του δικτύου να σταθεροποιηθούν σε συγκεκριμένες τιμές που θα προκαλούν σύγκλιση της μέσης τιμής των σφαλμάτων, για όλα τα πρότυπα εκπαίδευσης, στην ελάχιστη δυνατή τιμή της.

Όπως αναφέραμε στα πολυστρωματικά δίκτυα η έξοδος ενός επιπέδου γίνεται είσοδος στο επόμενο επίπεδο. Η διαδικασία αυτή περιγράφεται με την εξίσωση

$$\mathbf{y}^{m+1} = \mathbf{f}^{m+1}(\mathbf{W}^{m+1}\mathbf{y}^m + \mathbf{b}^{m+1}) \text{ για } m = 0, 1, \dots, M - 1,$$

όπου M είναι ο αριθμός των στρωμάτων του δικτύου. Οι νευρώνες στο πρώτο επίπεδο δέχονται εξωτερικές εισόδους

$$y^0 = p.$$

Οι έξοδοι των νευρώνων του τελευταίου επιπέδου αποτελούν τις εξόδους του νευρωνικού δικτύου. Αρχικά ο αλγόριθμος τροφοδοτείται με ένα σύνολο προτύπων εκπαίδευσης

$$\{\mathbf{p}_1, \mathbf{t}_1\}, \{\mathbf{p}_2, \mathbf{t}_2\}, \dots, \{\mathbf{p}_N, \mathbf{t}_N\},$$

όπου p_n είναι η είσοδος που τροφοδοτείται στο δίκτυο και t_n ο επιθυμητός στόχος εξόδου για τη συγκεκριμένη είσοδο. Καθώς κάθε είσοδος τροφοδοτείται στο δίκτυο η έξοδος του δικτύου συγκρίνεται με το στόχο. Ο αλγόριθμος της όπισθεν διάδοσης σφάλματος προσαρμόζει τα βάρη του δικτύου έτσι ώστε να ελαχιστοποιείται το μέσο τετραγωνικό σφάλμα

$$F(\mathbf{x}) = E[e^2] = E[(t - y)^2], \quad (7.1)$$

όπου \mathbf{x} είναι το διάνυσμα των βαρών και των πολώσεων του δικτύου. Αν το δίκτυο έχει πολλαπλές εξόδους η εξίσωση 7.1 γίνεται

$$F(\mathbf{x}) = E[\mathbf{e}^T \mathbf{e}] = E[(\mathbf{t} - \mathbf{y})^T (\mathbf{t} - \mathbf{y})]. \quad (7.2)$$

Όπως και με τον αλγόριθμο LMS προσεγγίζουμε το μέσο τετραγωνικό σφάλμα από την εξίσωση

$$F(\mathbf{x}) = (\mathbf{t}(k) - \mathbf{y}(k))^T (\mathbf{t}(k) - \mathbf{y}(k)) = \mathbf{e}^T(k) \mathbf{e}(k). \quad (7.3)$$

Σύμφωνα με την εξίσωση 7.3 χρησιμοποιούμε τον αλγόριθμο της πιο απότομης καθόδου για να υπολογίσουμε τα βάρη και τις πολώσεις του δικτύου ο οποίος δίνεται από τη σχέση:

$$w_{i,j}^m(k+1) = w_{i,j}^m(k) - a \frac{dF}{dw_{i,j}^m}, \quad (7.4)$$

$$b_i^m(k+1) = b_i^m(k) - a \frac{dF}{db_i^m}, \quad (7.5)$$

όπου a είναι ο ρυθμός εκπαίδευσης.

Επειδή το σφάλμα δεν είναι μια άμεση συνάρτηση των βαρών του δικτύου στα κρυφά επίπεδα για τον υπολογισμό των παραγώγων $\frac{dF}{dw_{i,j}^m}$ και $\frac{dF}{db_i^m}$ χρησιμοποιείται ο κανόνας της αλυσίδας. Αν υποθέσουμε ότι έχουμε μια αναλυτική συνάρτηση f με παράμετρο μια μεταβλητή n τότε η παράγωγος της συνάρτησης αυτής ως προς μια άλλη μεταβλητή w σύμφωνα με τον κανόνα της αλυσίδας δίνεται από τη σχέση

$$\frac{df(n(w))}{dw} = \frac{df(n)}{dn} \cdot \frac{dn(w)}{dw}.$$

Αποδεικνύεται ότι

$$\frac{dF}{dw_{i,j}^m} = s_i^m y_j^{m-1},$$

και

$$\frac{dF}{db_i^m} = s_i^m,$$

όπου $s_i^m = \frac{dF}{dn_i^m}$. Η μεταβλητή n_i^m αναπαριστά την είσοδο του δικτύου στο επίπεδο m και είναι μια αναλυτική συνάρτηση των βαρών του δικτύου σε αυτό το επίπεδο

$$n_i^m = \sum_{j=1}^{S^{m-1}} w_{i,j}^m y_j^{m-1} + b_i^m.$$

Επομένως, οι εξισώσεις 7.4 και 7.5 τώρα γίνονται

$$w_{i,j}^m(k+1) = w_{i,j}^m(k) - a s_i^m y_j^{m-1},$$

$$b_i^m(k+1) = b_i^m(k) - a s_i^m,$$

οι οποίες σε μορφή πινάκων γράφονται ως

$$\mathbf{W}^m(k+1) = \mathbf{W}^m(k) - a \mathbf{s}^m (\mathbf{y}^{m-1})^T,$$

$$\mathbf{b}^m(k+1) = \mathbf{b}^m(k) - a \mathbf{s}^m,$$

όπου

$$\mathbf{s}^m = \frac{dF}{d\mathbf{n}^m} = \begin{bmatrix} \frac{dF}{dn_1^m} \\ \frac{dF}{dn_2^m} \\ \vdots \\ \frac{dF}{dn_{S^m}^m} \end{bmatrix}.$$

7.2.1 Ανάστροφη μετάδοση των σημάτων σφάλματος διαμέσου του δικτύου

Προκειμένου να υπολογίσουμε το διάνυσμα s^m στο επίπεδο m χρειάζεται να υπολογίσουμε το s στο επίπεδο $m+1$. Για να το κάνουμε αυτό χρησιμοποιούμε τον παρακάτω

πίνακα

$$\frac{d\mathbf{n}^{m+1}}{d\mathbf{n}^m} = \begin{bmatrix} \frac{dn_1^{m+1}}{dn_1^m} & \frac{dn_1^{m+1}}{dn_2^m} & \cdots & \frac{dn_1^{m+1}}{dn_{S^m}^m} \\ \frac{dn_2^{m+1}}{dn_1^m} & \frac{dn_2^{m+1}}{dn_2^m} & \cdots & \frac{dn_2^{m+1}}{dn_{S^m}^m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{dn_{S^{m+1}}^{m+1}}{dn_1^m} & \frac{dn_{S^{m+1}}^{m+1}}{dn_2^m} & \cdots & \frac{dn_{S^{m+1}}^{m+1}}{dn_{S^m}^m} \end{bmatrix},$$

ο οποίος ονομάζεται και Ιακωβιανός πίνακας. Για ένα στοιχείο (i, j) αυτού του πίνακα έχουμε

$$\begin{aligned} \frac{dn_i^{m+1}}{dn_j^m} &= \frac{d(\sum_{l=1}^{S^m} w_{i,l}^{m+1} y_l^m + b_i^{m+1})}{dn_j^m} = w_{i,j}^{m+1} \frac{dy_j^m}{dn_j^m} \\ &= w_{i,j}^{m+1} \frac{df^m(n_j^m)}{dn_j^m} = w_{i,j}^{m+1} f^m(n_j^m), \end{aligned} \quad (7.6)$$

όπου

$$f^m(n_j^m) = \frac{df^m(n_j^m)}{dn_j^m}.$$

Επομένως ο Ιακωβιανός πίνακας μπορεί να γραφτεί ως

$$\frac{d\mathbf{n}^{m+1}}{d\mathbf{n}^m} = \mathbf{W}^{m+1} \mathbf{F}^m(\mathbf{n}^m), \quad (7.7)$$

όπου

$$\mathbf{F}^m(\mathbf{n}^m) = \begin{bmatrix} f^m(n_1^m) & 0 & \cdots & 0 \\ 0 & f^m(n_2^m) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & f^m(n_{S^m}^m) \end{bmatrix}. \quad (7.8)$$

Στη συνέχεια χρησιμοποιώντας τον κανόνα της αλυσίδας έχουμε

$$\begin{aligned} \mathbf{s}^m &= \frac{dF}{d\mathbf{n}^m} = \left(\frac{d\mathbf{n}^{m+1}}{d\mathbf{n}^m} \right)^T \frac{dF}{d\mathbf{n}^{m+1}} \\ &= \mathbf{F}^m(\mathbf{n}^m) (\mathbf{W}^{m+1})^T \frac{dF}{d\mathbf{n}^{m+1}} = \mathbf{F}^m(\mathbf{n}^m) (\mathbf{W}^{m+1})^T \mathbf{s}^{m+1}. \end{aligned} \quad (7.9)$$

Σύμφωνα λοιπόν με τον αλγόριθμο της όπισθεν διάδοσης σφάλματος έχουμε ένα ανάστροφο πέρασμα από το τελευταίο επίπεδο του δικτύου στο πρώτο, υπολογίζουμε δηλαδή τα σήματα του σφάλματος

$$\mathbf{s}^M \rightarrow \mathbf{s}^{M-1} \rightarrow \cdots \rightarrow \mathbf{s}^2 \rightarrow \mathbf{s}^1.$$

Το s^M αποκτάται στο επίπεδο εξόδου και υπολογίζεται από την εξίσωση

$$s_i^M = \frac{dF}{dn_i^M} = \frac{d(\mathbf{t} - \mathbf{y})^T (\mathbf{t} - \mathbf{y})}{dn_i^M} = \frac{d \sum_{j=1}^{S^M} (t_j - y_j)^2}{dn_i^M} = -2(t_i - y_i) \frac{dy_i}{dn_i^M}. \quad (7.10)$$

Επειδή όμως

$$\frac{dy_i}{dn_i^M} = \frac{dy_i^M}{dn_i^M} = \frac{df^M(n_i^M)}{dn_i^M} = f^M(n_i^M),$$

η εξίσωση 7.10 γράφεται ως

$$s_i^M = -2(t_i - y_i) f^M(n_i^M),$$

ή αλλιώς

$$\mathbf{s}^M = -2\mathbf{F}^M(\mathbf{n}^M)(\mathbf{t} - \mathbf{y}).$$

7.2.2 Ο αλγόριθμος BackPropagation

Ο αλγόριθμος της όπισθεν διάδοσης σφάλματος συνοψίζεται στα παρακάτω βήματα:

- Το πρώτο βήμα είναι να τροφοδοτήσουμε το δίκτυο με τις εισόδους

$$\mathbf{y}^0 = \mathbf{p},$$

$$\mathbf{y}^{m+1} = \mathbf{f}^{m+1}(\mathbf{W}^{m+1}\mathbf{y}^m + \mathbf{b}^{m+1}), \text{ για } m = 0, 1, \dots, M-1$$

και

$$\mathbf{y} = \mathbf{y}^M.$$

- Το δεύτερο βήμα είναι να μεταδώσουμε προς τα πίσω τα σήματα σφάλματος εξόδου του δικτύου

$$\mathbf{s}^M = -2\mathbf{F}^M(\mathbf{n}^M)(\mathbf{t} - \mathbf{y})$$

$$\mathbf{s}^m = \mathbf{F}^m(\mathbf{n}^m)(\mathbf{W}^{m+1})^T \mathbf{s}^{m+1}, \text{ για } m = M-1, \dots, 2, 1.$$

- Τέλος υπολογίζονται τα βάρη και οι πολώσεις του δικτύου σύμφωνα με τις σχέσεις

$$\mathbf{W}^m(k+1) = \mathbf{W}^m(k) - a\mathbf{s}^m(\mathbf{y}^{m-1})^T,$$

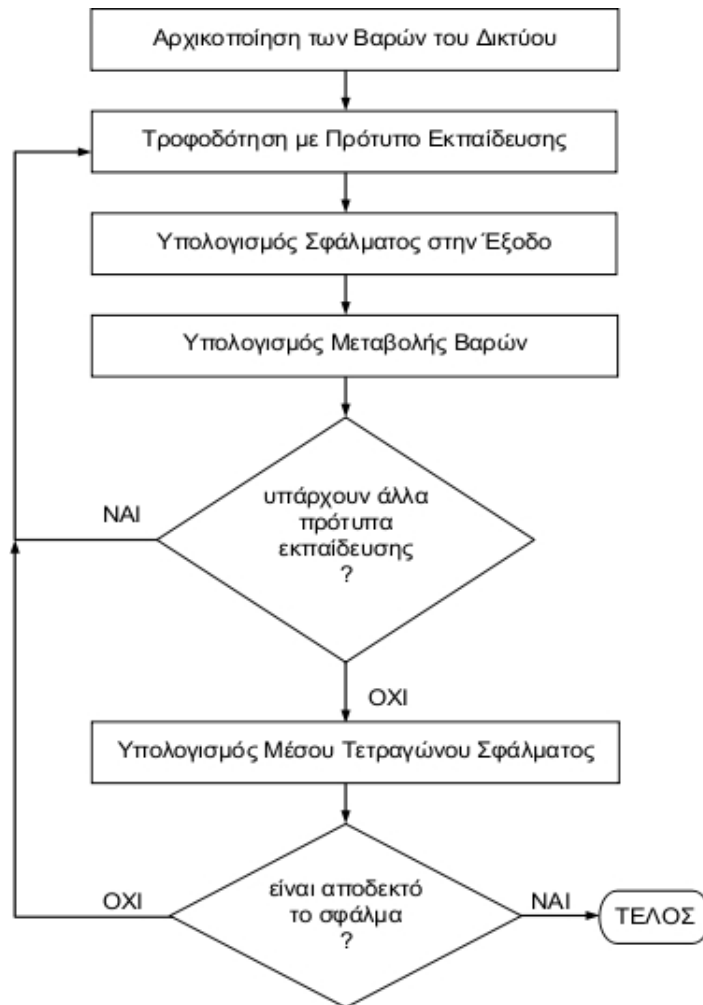
$$\mathbf{b}^m(k+1) = \mathbf{b}^m(k) - a\mathbf{s}^m.$$

Η συνολική διαδικασία εκπαίδευσης μέσω της όπισθεν διάδοσης σφάλματος, δίνεται διαγραμματικά στο Σχήμα 7.6.

7.3 Τρόποι εκπαίδευσης ενός πολυστρωματικού δικτύου

Όπως αναφέρθηκε η διαδικασία εκπαίδευσης με τον αλγόριθμο της όπισθεν διάδοσης σφάλματος λαμβάνει χώρα με την παρουσίαση και εφαρμογή στο νευρωνικό δίκτυο ενός συνόλου παραδειγμάτων εκπαίδευσης. Η παρουσίαση όλων των προτύπων εκπαίδευσης μια φορά το καθένα στο δίκτυο ονομάζεται εποχή. Η συνήθης διαδικασία εκπαίδευσης αναφέρεται στη μεταβολή των βαρών για κάθε διάνυσμα εκπαίδευσης και όταν παρουσιαστούν όλα τα διανύσματα ελέγχεται αν το μέσο τετραγωνικό σφάλμα μειώθηκε σε κάποια προκαθορισμένη, αποδεκτή τιμή.

Εδώ αξίζει να αναφερθεί ότι σε κάθε εποχή υπάρχει ο τυχαίος και ο σειριακός τρόπος παρουσίασης των προτύπων εκπαίδευσης. Ο πρώτος τρόπος έχει το πλεονέκτημα της εξασφάλισης του στοχαστικού χαρακτήρα της μάθησης και το μειονέκτημα των συνεχών και μη



Σχήμα 7.6: Η διαδικασία εκπαίδευσης της όπισθεν διάδοσης σφάλματος.

αμελητέων ταλαντώσεων των σφαλμάτων εκπαίδευσης και αξιολόγησης, ώσπου να φτάσουμε στο τελικό αποτέλεσμα. Από την άλλη ο δεύτερος τρόπος, της εν σειρά παρουσίασης προτύπων, δίνει συνήθως χαμηλότερο σφάλμα εκπαίδευσης και μεγαλύτερη δυνατότητα ελέγχου αυτού του σφάλματος, αλλά εμπεριέχει τον κίνδυνο της λεγόμενης υπερπροσαρμογής του δικτύου της συγκεκριμένης σειράς παρουσίασης των προτύπων, με αποτέλεσμα το τελικό σφάλμα αξιολόγησης να είναι κάποιες φορές υψηλότερο σε σχέση με εκείνο του τυχαίου τρόπου παρουσίασης.

Για ένα δεδομένο σύνολο εκπαίδευσης η διαδικασία εκπαίδευσης μπορεί να εκτελεστεί με δύο τρόπους. Η πρώτη αναφέρεται στην εκπαίδευση ανά ομάδα προτύπων εισόδου ή batch-training ενώ ο δεύτερος αφορά την εκπαίδευση ανά πρότυπο εισόδου ή online-training. Οι τεχνικές εκπαίδευσης ανά πρότυπο εισόδου απαιτούν λιγότερη ικανότητα αποθήκευσης από την εκπαίδευση ανά ομάδα προτύπων εισόδου. Παρόλα αυτά, λόγω της αυθαίρετης επιλογής της σειράς των προτύπων εκπαίδευσης, η εκπαίδευση ανά πρότυπο εισόδου αποτελεί μια στοχαστική διαδικασία, ενώ η εκπαίδευση ανά ομάδα προτύπων είναι μια αιτιοκρατική διαδικασία.

7.3.1 Εκπαίδευση ανά ομάδα προτύπων εισόδου

Στην εκπαίδευση ανά ομάδα προτύπων εισόδου αρχικά τροφοδοτούμε το δίκτυο με όλα τα πρότυπα εκπαίδευσης και στη συνέχεια προσαρμόζουμε τα βάρη. Τα βάρη ενημερώνονται έπειτα από την παρουσίαση στο δίκτυο όλου του συνόλου προτύπων της εποχής. Για μια τέτοια εποχή ορίζουμε τη μέση τιμή των τετραγωνικών σφαλμάτων. Εδώ λοιπόν η αναπροσαρμογή όλων των βαρών γίνεται μια φορά μετά το τέλος κάθε εποχής. Μια ιδανική συμπεριφορά θα πρέπει να οδηγήσει τα βάρη του δικτύου σε ένα ελάχιστο σημείο.

Η εκπαίδευση ανά ομάδα προτύπων εισόδου αντιμετωπίζεται με την ελαχιστοποίηση του μέσου τετραγωνικού σφάλματος για κάθε είσοδο και είναι μια αποκλειστικά αιτιοκρατική μέθοδος που απαιτεί όμως επιπλέον χώρο για την αποθήκευση κάθε βάρους, κάτι που μπορεί να είναι δύσκολο για εφαρμογές υλικού.

7.3.2 Εκπαίδευση ανά πρότυπο εισόδου

Στην περίπτωση της εκπαίδευσης ανά πρότυπο εισόδου τα βάρη αναπροσαρμόζονται έπειτα από την παρουσίαση κάθε προτύπου εκπαίδευσης στο δίκτυο. Για να γίνει αυτό πιο κατανοητό έστω ότι έχουμε μια εποχή που αποτελείται από N πρότυπα εκπαίδευσης

$$\{x_1, t_1\}, \{x_2, t_2\}, \dots, \{x_n, t_n\},$$

όπου x_i είναι ένα διάνυσμα εισόδου για $i = 1, \dots, N$, και t_i η επιθυμητή έξοδος του δικτύου που αντιστοιχεί στην είσοδο αυτή. Το πρώτο πρότυπο εφαρμόζεται στο δίκτυο και εκτελούνται τόσο το ευθύ, όσο και το αναστροφο πέρασμα που περιγράψαμε προηγουμένως, με αποτέλεσμα να μεταβάλλονται τα βάρη. Στη συνέχεια παρουσιάζεται το δεύτερο στη σειρά πρότυπο και εκτελείται η ίδια διαδικασία που οδηγεί σε νέα μεταβολή των βαρών. Η εποχή τελειώνει με την παρουσίαση του N -οστού προτύπου.

Η εκπαίδευση ανά πρότυπο εισόδου μπορεί να γίνει είτε επιλέγοντας τυχαία κάθε φορά ένα πρότυπο από το σύνολο των δεδομένων εκπαίδευσης και τροφοδοτώντας το πρότυπο αυτό στο δίκτυο ή με κυκλική εκπαίδευση η οποία προϋποθέτει την ύπαρξη κύκλων εκπαίδευσης στους οποίους κάθε πρότυπο εμφανίζεται μόνο μια φορά. Η τυχαία επιλογή προτύπων οδηγεί σε ένα ξεχωριστό είδος θορύβου που βοηθά να ξεφύγουμε από τοπικά ελάχιστα της συνάρτησης σφάλματος. Αυτός ο τύπος εκπαίδευσης θεωρείται κατάλληλος για μεγάλα σύνολα προτύπων, όπου η εκπαίδευση ανά ομάδα προτύπων εισόδου θα απαιτούσε περισσότερο χρόνο και μεγαλύτερο αποθηκευτικό χώρο.

Στην πράξη η μέθοδος που προτιμάται είναι η εκπαίδευση ανά πρότυπο, καθώς ενισχύει τον στοχαστικό χαρακτήρα και είναι πιο ικανή στην αποτροπή παγίδευσης του αλγορίθμου σε τοπικά ελάχιστα. Βέβαια η εκπαίδευση ανά εποχή, υπολογίζει με πιο ακριβή τρόπο το διάλυμα κλίσης. Σε κάθε πρόβλημα λοιπόν σταθμίζονται τα πλεονεκτήματα και μειονεκτήματα και επιλέγεται μια εκ των δύο μεθόδων εκπαίδευσης.

7.4 Σχεδίαση νευρωνικών δικτύων

Η πλήρης σχεδίαση ενός νευρωνικού δικτύου συνίσταται στον καθορισμό της τοπολογίας και στην διαδικασία εκπαίδευσης. Αν και ο αλγόριθμος εκπαίδευσης ασχολείται με όλες τις λεπτομέρειες της διαδικασίας σύγκλισης, δεν σχετίζεται με κανένα τρόπο με την τοπολογία, δηλαδή τη μορφή του δικτύου. Το πρόβλημα του καθορισμού της βέλτιστης τοπολογίας είναι ίσως το δυσκολότερο στον τομέα της σχεδίασης νευρωνικών δικτύων.

Γενικά, το νευρωνικό δίκτυο μπορεί να περιλαμβάνει αυθαίρετο αριθμό επιπέδων με αυθαίρετο πλήθος νευρώνων στο καθένα από αυτά. Η θεωρία αποδεικνύει ότι τρία το πολύ επίπεδα είναι αρκετά για την επίλυση οποιουδήποτε προβλήματος, όμως ο αριθμός των επιπέδων σχετίζεται άμεσα με το μέγεθος του κάθε ενός από αυτά. Για πολυεπίπεδα δίκτυα τύπου perceptron που χρησιμοποιούν αλγόριθμο εκπαίδευσης Back-Propagation, μελέτες σε επίπεδο περισσότερο πειραματικό παρά θεωρητικό, έχουν προτείνει διάφορες μεθοδολογίες για τον καθορισμό του βέλτιστου αριθμού επιπέδων και νευρώνων στο δίκτυο.

Το νευρωνικό δίκτυο μπορεί να θεωρηθεί ως ένας γενικής μορφής μη γραμμικός ταξινομητής ικανός να υλοποιήσει αυθαίρετες απεικονίσεις μεταξύ εισόδου και εξόδου. Όπως κάθε παρόμοιο σύστημα, οι βαθμοί ελευθερίας καθορίζονται από το πλήθος των ελεύθερων παραμέτρων που πρόκειται να καθοριστούν μέσα από τη διαδικασία της εκπαίδευσης. Στην περίπτωση των γραμμικών συστημάτων ο καθορισμός των παραμέτρων γίνεται με μονοσήμαντο τρόπο μέσω κάποιας άμεσης ή έμμεσης μεθόδου ανάδειξης των εξαρτήσεων μεταξύ εισόδου και επιθυμητής εξόδου. Το ίδιο μπορεί θεωρητικά να εφαρμοστεί και στην περίπτωση των μη γραμμικών συστημάτων, με την εφαρμογή περισσότερο πολύπλοκων μεθόδων για τον καθορισμό των παραμέτρων.

Παρόλα αυτά, η περίπτωση των μη γραμμικών συστημάτων είναι στην πραγματικότητα πολύ πιο σύνθετη από ότι αρχικά φαίνεται. Επειδή τα μη γραμμικά συστήματα είναι συχνά δύσκολο να επιλυθούν άμεσα, για τον καθορισμό των παραμέτρων χρησιμοποιούνται σχεδόν πάντοτε έμμεσες μέθοδοι και διάφορα επαναληπτικά σχήματα, όπως η μορφή

του αλγορίθμου Back-Propagation. Επιπλέον, συνήθως εφαρμόζονται σε προβλήματα όπου, είτε λόγω πολυπλοκότητας, είτε λόγω υψηλού βαθμού μη γραμμικότητας, είναι δύσκολο να καθοριστούν σύνολα εκπαίδευσης που να καλύπτουν όλο το εύρος του χώρου των προτύπων. Έτσι ο αλγόριθμος εκπαίδευσης θα πρέπει, από τη μια πλευρά να εκτελεστεί με βάση ελλιπή πληροφορία σχετικά με το χώρο, ενώ παράλληλα απαιτείται η σχεδίαση ενός συστήματος με ικανοποιητική απόδοση για κάθε συνδυασμό εισόδου/εξόδου. Ακόμα, είναι απαραίτητο το δίκτυο να λειτουργεί με υψηλό ποσοστό επιτυχίας και στην περίπτωση που τα δεδομένα εισόδου είναι ελλιπή ή περιέχουν θόρυβο. Τα χαρακτηριστικά αυτά καθορίζουν την ιδιότητα που ονομάζεται ικανότητα γενίκευσης του δικτύου και αποτελεί έναν από τους σημαντικότερους παράγοντες στη σχεδίασή του. Ο υψηλός βαθμός ικανότητας γενίκευσης και η ανθεκτικότητα σε θόρυβο ή ελλιπή δεδομένα, καθώς επίσης η προσαρμοστικότητα και ο υψηλός βαθμός παραλληλίας, είναι μερικά από τα σημαντικότερα στοιχεία επιτυχίας των νευρωνικών δικτύων σε μεγάλο εύρος εφαρμογών.

Για την επίτευξη υψηλής ικανότητας γενίκευσης του νευρωνικού δικτύου, θα πρέπει, εκτός από την πληρότητα του συνόλου εκπαιδευτικών προτύπων, η δομή και τοπολογία να επιτρέπει την κατασκευή μιας επαρκούς περιγραφής του χώρου μέσω των τιμών των βαρών του. Αν και θεωρητικά για τον καθορισμό της τιμής N ελεύθερων παραμέτρων απαιτούνται επίσης N πρότυπα εκπαίδευσης, παρόλα αυτά για την επίτευξη υψηλού βαθμού ικανότητας γενίκευσης απαιτούνται πολύ περισσότερα πρότυπα. Συχνά το απαιτούμενο μέγεθος του συνόλου εκπαίδευσης υπολογίζεται ως ακέραια πολλαπλάσια του συνολικού αριθμού των βαρών του δικτύου, συνήθως μεταξύ 3 και 10. Η ίδια τεχνική μπορεί να εφαρμοστεί και με αντίστροφη λογική, δηλαδή με βάση το μέγεθος του διαθέσιμου συνόλου εκπαίδευσης, να καθορισθεί το μέγιστο πλήθος συνολικά των βαρών του δικτύου, έτσι ώστε να επιτυγχάνεται αρκετά καλή ικανότητα γενίκευσης.

Στην περίπτωση που το σύνολο των εκπαιδευτικών προτύπων δεν επαρκεί για την ικανοποιητική εκπαίδευση του δικτύου, είναι δυνατό να δημιουργηθούν προβλήματα στην συμπεριφορά του τελικού δικτύου. Αν και η εκπαίδευση μπορεί να ολοκληρωθεί με επιτυχία μετά από κάποιο αριθμό επαναλήψεων, η απόδοση του δικτύου για άγνωστα (πρωτοεμφανιζόμενα) πρότυπα μπορεί να είναι απογοητευτική λόγω της μη αντιπροσωπευτικής αναπαράστασης του χώρου από τη δομή και το σύνολο των βαρών. Το ίδιο συμβαίνει επίσης στην περίπτωση όπου η διαδικασία εκπαίδευσης συνεχίζεται πέρα από το όριο όπου έχει επιτευχθεί επαρκής αναπαράσταση του χώρου και από εκεί και πέρα το δίκτυο εκπαιδεύεται με βάση τα λεπτομερειακά χαρακτηριστικά των συγκεκριμένων προτύπων του συνόλου εκπαίδευσης, σε αντίστοιχο κόστος στην ικανότητα γενίκευσης και στην γενική απόδοση του δικτύου. Η κατάσταση αυτή είναι σαφώς μη επιθυμητή και χαρακτηρίζεται ως υπερ-εκπαίδευση (over-training).

Για την αποφυγή φαινομένων υπερ-εκπαίδευσης, καθώς επίσης και την παρακολούθηση της ικανότητας γενίκευσης του νευρωνικού δικτύου, έχουν προταθεί διάφορες τεχνικές που αποτελούν ουσιαστικά παραλλαγές της γενικότερης μεθοδολογίας διαχωρισμού του διαθέσιμου συνόλου προτύπων σε σύνολα εκπαίδευσης και σύνολα αξιολόγησης. Μια αποτελεσματική τεχνική που χρησιμοποιείται ευρύτατα στην εκπαίδευση νευρωνικών δικτύων είναι η μέθοδος

Leave-Half-Out (LHO) ή Cross- Validation. Σύμφωνα με την τεχνική αυτή, το σύνολο των διαθέσιμων προτύπων διαμερίζεται με τυχαίο τρόπο σε δύο ίσα σύνολα. Το πρώτο ονομάζεται σύνολο εκπαίδευσης και χρησιμοποιείται για τη σχεδίαση του δικτύου και το δεύτερο ονομάζεται σύνολο αξιολόγησης και χρησιμοποιείται για τη μέτρηση της απόδοσης του δικτύου σε άγνωστα πρότυπα. Η βασική ιδέα είναι η διαδικασία εκπαίδευσης να ακολουθείται από μια φάση αξιολόγησης με άγνωστα πρότυπα, έτσι ώστε να μετρηθεί η ικανότητα γενίκευσης και η πραγματική απόδοση του δικτύου.

Εναλλακτικά, κάθε κύκλος εκπαίδευσης μπορεί να ακολουθείται από ένα κύκλο αξιολόγησης για την μέτρηση της τρέχουσας απόδοσης του δικτύου. Με τον τρόπο αυτό είναι εύκολος ο έλεγχος και η διακοπή της διαδικασίας εκπαίδευσης όταν έχει επιτευχθεί ο μέγιστος βαθμός γενίκευσης, και παράλληλα εξάλειψη του κινδύνου της υπερπροσαρμογής του δικτύου. Ακόμα καλύτερα αποτελέσματα επιτυγχάνονται όταν υπάρχει φάση αξιολόγησης, τόσο κατά τη διάρκεια, όσο και μετά την ολοκλήρωση της εκπαίδευσης. Συχνά για να περιοριστεί ο απαιτούμενος χρόνος ολοκλήρωσης της εκπαίδευσης, χρησιμοποιείται ένα μικρό σύνολο αξιολόγησης σε κάθε κύκλο εκπαίδευσης, το οποίο ονομάζεται σύνολο επικύρωσης, ενώ η τελική αξιολόγηση γίνεται με άλλο μεγαλύτερο σύνολο που δίνει μια καλή εκτίμηση για το βαθμό αξιοπιστίας του δικτύου. Τεχνικές αυτού του τύπου εφαρμόζονται κατά κόρο σε εφαρμογές όπου τα στατιστικά χαρακτηριστικά του χώρου των προτύπων είναι άγνωστα ή όταν τα διαθέσιμα σύνολα εκπαίδευσης είναι σχετικά περιορισμένα σε μέγεθος και ανεπαρκή για την πλήρη περιγραφή του.

Αξίζει να σημειωθεί πως ο συνδυασμός φάσεων τροποποίησης και αξιολόγησης του νευρωνικού δικτύου κατά τη διαδικασία εκπαίδευσης δίνει τη δυνατότητα συνεχούς παρακολούθησης της απόδοσης του δικτύου, καλύτερου ελέγχου της διαδικασίας και περισσότερο βέλτιστη επιλογή της τελικής διαμόρφωσης. Συχνά η διαδικασία εκπαίδευσης ενσωματώνει αυτόματους μηχανισμούς καταγραφής και διατήρησης της βέλτιστης διαμόρφωσης του δικτύου, καθώς διαμορφώνεται διαμέσου των εκπαιδευτικών κύκλων. Έτσι μετά την ολοκλήρωση της εκπαίδευσης είναι εύκολη η επιλογή της διαμόρφωσης που δίνει την βέλτιστη απόδοση.

Αν και οι όλες οι παραπάνω τεχνικές είναι ιδιαίτερα χρήσιμες για τη βελτίωση της ποιότητας της διαδικασίας εκπαίδευσης, παρόλα αυτά παραμένει το πρόβλημα του καθορισμού της τοπολογίας του δικτύου. Για την αντιμετώπιση του συγκεκριμένου ζητήματος έχουν δημοσιευτεί αναρίθμητα άρθρα και εργασίες με σκοπό την διατύπωση κάποιας μεθοδολογίας αυτόματης επιλογής του πλήθους και του μεγέθους των νευρωνικών επιπέδων στο δίκτυο. Το στοιχείο που αξίζει να σημειωθεί εδώ είναι ότι, αν και ο συνολικός αριθμός βαρών στο δίκτυο σχετίζεται έμμεσα με το πλήθος των προτύπων εκπαίδευσης, η ακριβής διάταξη και διασύνδεση των νευρώνων στο δίκτυο αποτελούν επιπλέον ελεύθερες παραμέτρους για το σύστημα, που δεν σχετίζονται με κανένα τρόπο ούτε με το σύνολο εκπαίδευσης, ούτε με τη δομή και τη λειτουργία του αλγορίθμου εκπαίδευσης. Σε κάθε περίπτωση όμως η τοπολογία και η δομή του δικτύου επηρεάζει δραματικά την πολυπλοκότητα και το βαθμό σύγκλισης της διαδικασίας εκπαίδευσης.

Στις περισσότερες περιπτώσεις η επιλογή της τοπολογίας του δικτύου αποτελεί έναν από

τους δυσκολότερους και ταυτόχρονα πιο κρίσιμους παράγοντες για την επιτυχία του νευρωνικού δικτύου. Για πολυστρωματικά δίκτυα τύπου perceptron που εκπαιδεύονται με τον αλγόριθμο εκπαίδευσης Back-Propagation έχουν διατυπωθεί κάποιοι εμπειρικοί κανόνες που συνδέουν το μέγεθος των ακραίων επιπέδων (εισόδου, εξόδου) με το μέγεθος των κρυφών επιπέδων. Το μέγεθος των ακραίων επιπέδων μπορεί να καθοριστεί εν μέρει από τη φύση του προβλήματος και συγκεκριμένα από τη διάσταση των διανυσμάτων εισόδου και εξόδου. Για δίκτυα με τρία συνολικά νευρωνικά επίπεδα, το μέγεθος του μοναδικού κρυφού επιπέδου συχνά καθορίζεται είτε ως η μέση τιμή των δύο μεγεθών, είτε ως η τετραγωνική ρίζα του συνολικού πλήθους των βαρών στο δίκτυο. Τα επίπεδα σχεδόν πάντα θεωρούνται πλήρως διασυνδεδεμένα, δηλαδή όλες οι εξοδοί του ενός επιπέδου οδηγούνται ως εισοδοί σε όλους τους νευρώνες του επόμενου επιπέδου.

Η απενεργοποίηση κάποιας διασύνδεσης πραγματοποιείται απλά με το μηδενισμό της τιμής του αντίστοιχου βάρους. Σε μια προσπάθεια αυτοματοποίησης της διαδικασίας καθορισμού της τοπολογίας του νευρωνικού δικτύου έχουν προταθεί διάφορες τεχνικές, οι οποίες μπορούν γενικά να κατηγοριοποιηθούν σε μεθόδους αύξησης και μεθόδους μείωσης της τοπολογίας. Στην πρώτη περίπτωση ο αλγόριθμος ξεκινά από μια ελάχιστη τοπολογία και προχωρά αυξάνοντας διαδοχικά την έκταση και την πολυπλοκότητά της. Αντίθετα, στη δεύτερη περίπτωση ο αλγόριθμος ξεκινά από μια μέγιστη τοπολογία και προχωρά μειώνοντας διαδοχικά την έκταση και την πολυπλοκότητά της. Η καταλληλότητα κάθε μιας από αυτές εξαρτάται από το εκάστοτε πρόβλημα και τις απαιτήσεις σε ότι αφορά την απόδοση του τελικού δικτύου.

Η αύξηση ή η μείωση της τοπολογίας του δικτύου μπορεί να γίνεται σε επίπεδο πλήθους βαρών (διασυνδέσεων), πλήθους νευρώνων ή πλήθους νευρωνικών επιπέδων. Η δυσκολία έγκειται κυρίως στο γεγονός ότι η επιμέρους εκτέλεση της διαδικασίας επιλογής τοπολογίας και της διαδικασίας εκπαίδευσης δημιουργεί πρακτικά μεγάλο κόστος ως προς το συνολικό χρόνο και την πολυπλοκότητα της σχεδίασης του δικτύου. Συχνά ο καθορισμός της τοπολογίας συνδυάζεται με την εκπαίδευση του δικτύου για λόγους οικονομίας, κάτι που οδηγεί σε υποβέλτιστες λύσεις. Σε εφαρμογές όμως όπου δεν μπορεί να γίνει καλή αρχική εκτίμηση της τοπολογίας, οποιαδήποτε αυτόματη μέθοδος είναι ιδιαίτερα χρήσιμη.

Μια από τις πιο γνωστές μεθόδους αναζήτησης μιας ελάχιστης τοπολογίας είναι η επιλεκτική καταστροφή τμημάτων του δικτύου (pruning). Μετά την ολοκλήρωση της εκπαίδευσης, η μέθοδος αναζητά βάρη των οποίων οι τιμές είναι σχετικά κοντά στο μηδέν, κάτι που υποδεικνύει ότι οι συγκεκριμένες συνδέσεις είναι ασήμαντες για την απόδοση του δικτύου. Ανάλογα με την ανοχή σφάλματος στην έξοδο, καταστρέφονται επιλεκτικά με αυτό τον τρόπο διασυνδέσεις νευρώνων, δίνοντας τελικά μια απλούστερη δομή για την τοπολογία του δικτύου. Ακριβώς το αντίθετο μπορεί να εφαρμοστεί με αυξητικές μεθόδους, ξεκινώντας από μικρές τοπολογίες και προσθέτοντας συνεχώς βάρη ή ολόκληρους νευρώνες με σκοπό τη βελτίωση της απόδοσης του δικτύου. Οι πιο σύγχρονες τεχνικές χρησιμοποιούν γενετικούς αλγορίθμους για τη σταδιακή ανάπτυξη και βελτιστοποίηση της τοπολογίας. Αν και συχνά αποδεικνύονται χρήσιμες σε πρακτική χρήση, παρόλα αυτά το βασικό μειονέκτημα των αυτοματοποιημένων μεθόδων είναι το γεγονός ότι δύσκολα αποσυνδέονται από τον ί-

διο τον αλγόριθμο εκπαίδευσης και κατά συνέπεια η εφαρμογή τους σε ευρεία κλίμακα αρχιτεκτονικών νευρωνικών δικτύων παραμένει μέχρι σήμερα προβληματική.

Κεφάλαιο 8

Παραλλαγές του αλγορίθμου της όπισθεν διάδοσης σφάλματος

Όταν ο βασικός αλγόριθμος της όπισθεν διάδοσης σφάλματος εφαρμόζεται σε ένα πρακτικό πρόβλημα, η εκπαίδευση μπορεί να διαρκέσει μέρες ή εβδομάδες υπολογιστικού χρόνου. Για το λόγο αυτό έχουν γίνει έρευνες για νέες μεθόδους που επιταχύνουν τη σύγκλιση του αλγορίθμου και οι οποίες χωρίζονται σε δύο κατηγορίες. Η πρώτη περιλαμβάνει την ανάπτυξη ευρετικών τεχνικών όπως είναι η τεχνική της μεταβολής του ρυθμού εκπαίδευσης. Η δεύτερη κατηγορία επικεντρώνεται σε τεχνικές αριθμητικής βελτιστοποίησης [33], [6]. Στην ενότητα αυτή θα αναφερθούμε σε δύο τεχνικές αριθμητικής βελτιστοποίησης που εφαρμόζονται στην εκπαίδευση πολυστρωματικών δικτύων *perceptron*, τη μέθοδο των συζυγών κλίσεων και τον αλγόριθμο *Levenberg-Marquardt*.

8.1 Εκπαίδευση με τη μέθοδο των συζυγών κλίσεων

Όπως περιγράψαμε σε προηγούμενη ενότητα η μέθοδος της πιο απότομης καθόδου είναι ο πιο απλός αλγόριθμος για την εύρεση του ελαχίστου μιας συνάρτησης όμως συγκλίνει σχετικά αργά. Η μέθοδος του *Newton* είναι πολύ γρηγορότερη, όμως απαιτεί τον υπολογισμό του *Εσσιανού* πίνακα και του αντίστροφού του. Αντίθετα, η μέθοδος των συζυγών κλίσεων δεν απαιτεί τον υπολογισμό των δεύτερων παραγώγων και έχει την ιδιότητα να συγκλίνει στο ελάχιστο σημείο μιας τετραγωνικής συνάρτησης σε πεπερασμένο αριθμό βημάτων [10]. Η μέθοδος αυτή μπορεί να χρησιμοποιηθεί εξίσου για την εκπαίδευση πολυστρωματικών δικτύων χρησιμοποιώντας μια παραλλαγή του αλγορίθμου της όπισθεν διάδοσης σφάλματος.

Όπως είδαμε ο αλγόριθμος των συζυγών κλίσεων συνοψίζεται στα παρακάτω βήματα :

1. Επιλέγουμε την κατεύθυνση προς την οποία θα αναζητήσουμε το ελάχιστο σημείο. Η κατεύθυνση αυτή ορίζεται ως η αρνητική τιμή του διανύσματος των μερικών παραγώγων.

$$p_0 = -g_0$$

2. Κάνουμε ένα βήμα σύμφωνα με την εξίσωση

$$\Delta \mathbf{x}_k = a_k \mathbf{p}_k,$$

όπου a_k είναι ο ρυθμός εκπαίδευσης ο οποίος επιλέγεται έτσι ώστε να ελαχιστοποιεί τη συνάρτηση κατά μήκος της κατεύθυνσης αναζήτησης

$$\mathbf{x}_{k+1} = \mathbf{x}_k + a_k \mathbf{p}_k.$$

3. Στη συνέχεια επιλέγουμε την επόμενη κατεύθυνση της αναζήτησης με την εξίσωση

$$\mathbf{p}_k = -\mathbf{g}_k + \beta_k \mathbf{p}_{k-1}.$$

όπου

$$\beta_k = \frac{\Delta \mathbf{g}_{k-1}^T \mathbf{g}_k}{\Delta \mathbf{g}_{k-1}^T \mathbf{p}_{k-1}}$$

ή

$$\beta_k = \frac{\mathbf{g}_k^T \mathbf{g}_k}{\mathbf{g}_{k-1}^T \mathbf{g}_{k-1}}$$

ή

$$\beta_k = \frac{\Delta \mathbf{g}_{k-1}^T \mathbf{g}_k}{\mathbf{g}_{k-1}^T \mathbf{g}_{k-1}}$$

4. Αν ο αλγόριθμος δεν έχει συγκλίνει τότε επιστρέφουμε στο βήμα 2.

Αυτό ο αλγόριθμος δεν μπορεί να εφαρμοστεί απευθείας στην εκπαίδευση ενός πολυ-στρωματικού νευρωνικού δικτύου επειδή ο δείκτης απόδοσης δεν είναι μια τετραγωνική συνάρτηση. Αυτό έχει σαν αποτέλεσμα να μην μπορούμε την εξίσωση 5.27 για ελαχιστοποίηση του δείκτη απόδοσης όπως περιγράφεται στο βήμα 2. Επιπλέον, η ακριβής θέση του ελάχιστου σημείου δεν μπορεί να βρεθεί συνήθως σε πεπερασμένο αριθμό βημάτων και ο αλγόριθμος χρειάζεται να σταματήσει μετά από κάποιον αριθμό επαναλήψεων.

Αρχικά απαιτείται μια διαδικασία για τον εντοπισμό του ελάχιστου μίας συνάρτησης προς μια συγκεκριμένη κατεύθυνση. Η διαδικασία αυτή αποτελείται από δύο βήματα. Εντοπίζεται μια εσωτερική περιοχή η οποία περιέχει ένα τοπικό ελάχιστο. Στη συνέχεια μειώνουμε το μέγεθος αυτής της αρχικής περιοχής έτσι ώστε να εντοπιστεί το ελάχιστο με την επιθυμητή ακρίβεια.

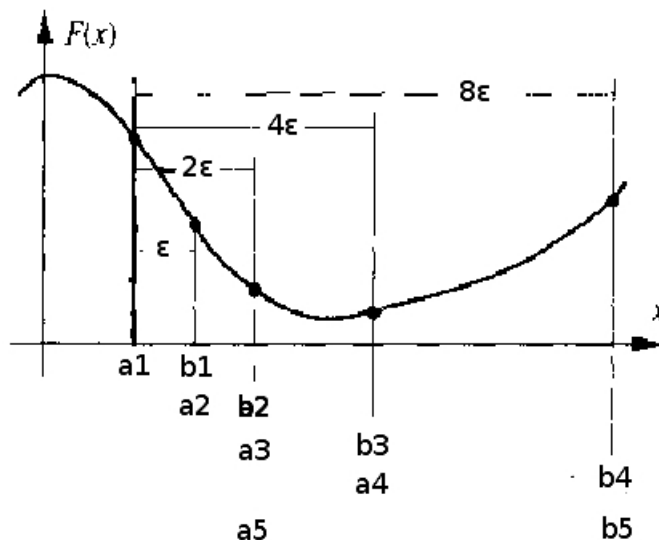
Για τον καθορισμό της εσωτερικής περιοχής ξεκινάμε υπολογίζοντας το δείκτη απόδοσης σε ένα αρχικό σημείο, a_1 . Το σημείο αυτό αντιστοιχεί στις τιμές των βαρών του δικτύου, υπολογίζουμε δηλαδή το

$$F(\mathbf{x}_0).$$

Το επόμενο βήμα είναι να υπολογιστεί η συνάρτηση σε ένα δεύτερο σημείο, b_1 , το οποίο απέχει απόσταση e από το αρχικό σημείο κατά μήκος μιας διεύθυνσης p_0 . Δηλαδή υπολογίζουμε το

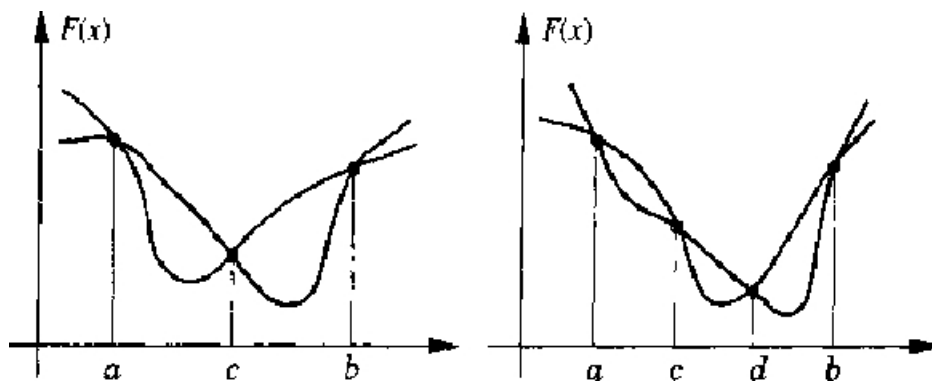
$$F(\mathbf{x}_0 + e\mathbf{p}_0).$$

Η διαδικασία συνεχίζεται υπολογίζοντας το δείκτη απόδοσης σε νέα σημεία b_l , διπλασιάζοντας σταδιακά την απόσταση μεταξύ των σημείων. Η διαδικασία σταματάει όταν η συνάρτηση μεταξύ δύο διαδοχικών σημείων που υπολογίσαμε, αρχίζει να αυξάνεται. Όπως παρατηρούμε στο Σχήμα 8.1 γνωρίζουμε ότι το ελάχιστο σημείο θα βρίσκεται μεταξύ των δύο σημείων a_5 και b_5 . Το ελάχιστο μπορεί να βρίσκεται είτε ανάμεσα στα a_4 και b_4 , ή ανάμεσα στα a_3 και b_3



Σχήμα 8.1: Περιοχή εντοπισμού του ελαχίστου σημείου.

όπως φαίνεται στο Σχήμα 8.2. Για να μειώσουμε αυτήν την περιοχή προκειμένου να βρούμε το ελάχιστο πρέπει να υπολογίσουμε την τιμή της συνάρτησης $F(x)$ για σημεία μεταξύ του διαστήματος $[a_5, b_5]$. Θα χρειαστεί να υπολογίσουμε την τιμή της $F(x)$ σε τουλάχιστον δύο



Σχήμα 8.2: Μείωση της περιοχής εντοπισμού του ελαχίστου σημείου.

εσωτερικά σημεία έτσι ώστε να μειώσουμε το μέγεθος της περιοχής όπου υπάρχει αβεβαιότητα. Ο υπολογισμός της συνάρτησης σε ένα μόνο σημείο δε μας παρέχει καμία πληροφορία σχετικά με την τοποθεσία του ελαχίστου. Αν όμως υπολογίσουμε τη συνάρτηση σε δύο σημεία c και d όπως φαίνεται στο Σχήμα 8.2, μπορούμε να μειώσουμε αυτήν την περιοχή. Αν $F(c) > F(d)$, τότε το ελάχιστο σημείο θα βρίσκεται στην περιοχή ανάμεσα στο διάστημα

$[c, b]$, ενώ αν $F(c) < F(d)$ τότε το ελάχιστο βρίσκεται στην περιοχή $[a, b]$. Στην περίπτωση αυτή υποθέτουμε ότι υπάρχει ένα μόνο ελάχιστο σημείο στην περιοχή που εντοπίσαμε.

Για να καθορίσουμε τα σημεία c και d χρησιμοποιούμε μια μέθοδο η οποία μειώνει τον αριθμό των υπολογισμών που απαιτούνται της συνάρτησης $F(x)$ και ονομάζεται Αναζήτηση Χρυσής Περιοχής (Golden Section Search). Σύμφωνα με τη μέθοδο αυτή σε κάθε επανάληψη απαιτείται ένας νέος υπολογισμός της συνάρτησης. Για παράδειγμα στην περίπτωση του Σχήματος 8.2(β) η συνάρτηση για το σημείο a δεν υπολογίζεται και το σημείο c θεωρείται ως εξωτερικό σημείο αυτής της περιοχής. Ένα καινούργιο σημείο c' υπολογίζεται μεταξύ των αρχικών σημείων c και d . Έτσι έχουμε μείωση αυτής της περιοχής.

Ο αλγόριθμος που κάνει αυτήν την αναζήτηση παρουσιάζεται παρακάτω:

- $\tau = 0.618$, που ονομάζεται χρυσή τομή

- Θέτουμε

$$c_1 = a_1 + (1 - \tau)(b_1 - a_1), F_c = F(c_1)$$

και

$$d_1 = b_1 - (1 - \tau)(b_1 - a_1), F_d = F(d_1)$$

- Για $k = 1, 2, \dots$

- Αν $F_c < F_d$ τότε θέσε

$$a_{k+1} = a_k; b_{k+1} = d_k; d_{k+1} = c_k$$

$$c_{k+1} = a_{k+1} + (1 - \tau)(b_{k+1} - a_{k+1})$$

$$F_d = F_c; F_c = F(c_{k+1})$$

- Αλλιώς θέσε

$$a_{k+1} = c_k; b_{k+1} = b_k; c_{k+1} = d_k$$

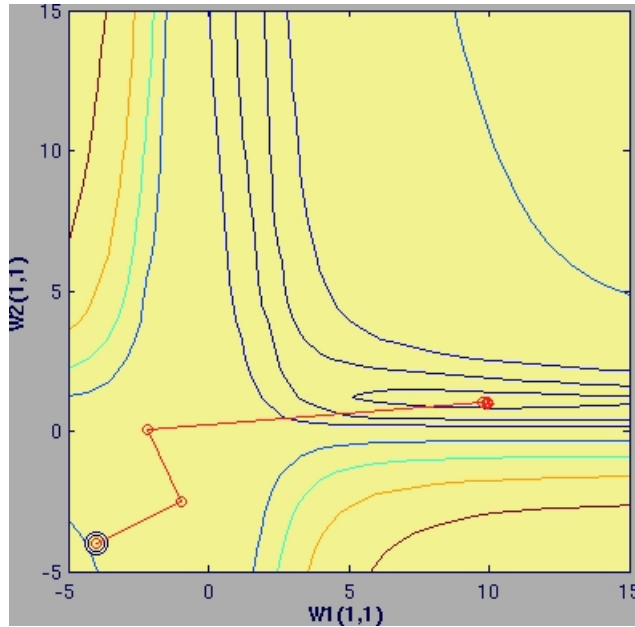
$$d_{k+1} = b_{k+1} + (1 - \tau)(b_{k+1} - a_{k+1})$$

$$F_c = F_d; F_d = F(d_{k+1})$$

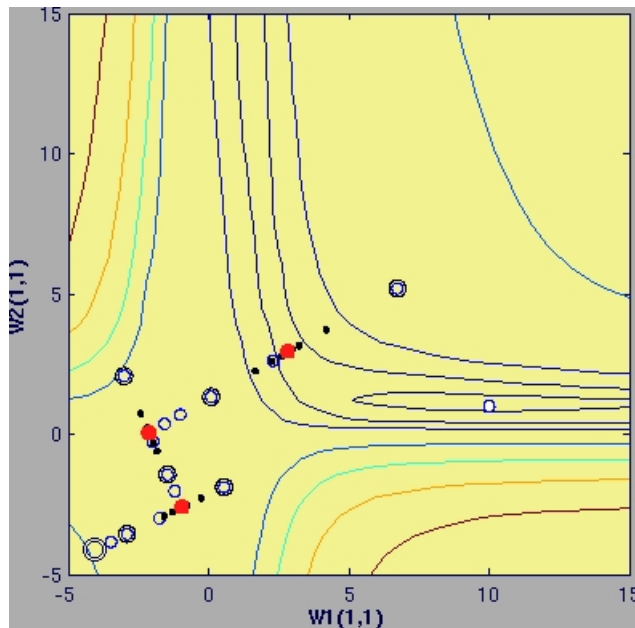
- Ο αλγόριθμος σταματάει όταν $b_{k+1} - a_{k+1} < tol$, όπου tol είναι η ακρίβεια που έχει δοθεί από το χρήστη.

Τέλος υπάρχει άλλη μία μετατροπή που πρέπει να γίνει στον αλγόριθμο των συζυγών κλίσεων πριν εφαρμοστεί στην εκπαίδευση ενός νευρωνικού δικτύου. Για τετραγωνικές συναρτήσεις ο αλγόριθμος συγκλίνει στο ελάχιστο σημείο το πολύ σε n επαναλήψεις, όπου n είναι ο αριθμός των βαρών και των πολώσεων του δικτύου που βελτιστοποιούνται. Ο δείκτης απόδοσης για τα πολυστρωματικά δίκτυα όμως δεν είναι μια τετραγωνική συνάρτηση. Η ανάπτυξη του αλγορίθμου των συζυγών κλίσεων δεν υποδεικνύει ποια κατεύθυνση αναζήτησης θα πρέπει να χρησιμοποιηθεί αφού έχει ολοκληρωθεί ένα κύκλος n επαναλήψεων. Η πιο απλή μέθοδος είναι να ορίσουμε σαν κατεύθυνση αναζήτησης μετά από n επαναλήψεις την κατεύθυνση που ορίζεται από τη μέθοδο της πιο απότομης καθόδου.

Στο Σχήμα 8.3 φαίνεται η σύγκλιση του αλγορίθμου στο βέλτιστο σύνολο των βαρών του δικτύου, ενώ το Σχήμα 8.4 μας δείχνει τα βήματα που ακολούθησε ο αλγόριθμος. Κάθε μπλε κύκλος παριστάνει τα σημεία στα οποία υπολογίζεται η συνάρτηση για την εύρεση της εσωτερικής περιοχής, ενώ οι κόκκινοι κύκλοι μας δείχνουν τα σημεία τα οποία υπολογίστηκαν για κάθε επανάληψη της μεθόδου Αναζήτησης Χρυσής Περιοχής.



Σχήμα 8.3: Σύγκλιση με τη μέθοδο των συζυγών κλίσεων.



Σχήμα 8.4: Τα βήματα του αλγορίθμου των συζυγών κλίσεων.

8.2 Ο αλγόριθμος Levenberg-Marquardt

Ο αλγόριθμος Levenberg-Marquardt αποτελεί μια παραλλαγή της μεθόδου του Newton ο οποίος έχει σχεδιαστεί έτσι ώστε να ελαχιστοποιεί συναρτήσεις οι οποίες αποτελούν άθροισμα τετραγώνων άλλων μη γραμμικών συναρτήσεων.

Αν υποθέσουμε ότι ο δείκτης απόδοσης είναι ένα άθροισμα τετραγώνων τότε η μέθοδος του Newton για τη βελτιστοποίηση του δείκτη απόδοσης $F(\mathbf{x})$ είναι

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{A}_k^{-1} \mathbf{g}_k, \quad (8.1)$$

όπου $\mathbf{A}_k = \nabla^2 F(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_k}$ και $\mathbf{g}_k = \nabla F(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_k}$. Ο δείκτης απόδοση δίνεται από τη σχέση

$$F(\mathbf{x}) = \sum_{i=1}^N \mathbf{v}_i^2(\mathbf{x}) = \mathbf{v}^T(\mathbf{x})\mathbf{v}(\mathbf{x}).$$

Τότε το j στοιχείο του διανύσματος των μερικών παραγώγων θα είναι

$$[\nabla F(\mathbf{x})]_j = \frac{dF(\mathbf{x})}{dx_j} = 2 \sum_{i=1}^N v_i(\mathbf{x}) \frac{dv_i(\mathbf{x})}{dx_j}.$$

Ο πίνακας των μερικών παραγώγων γράφεται ως

$$\nabla F(\mathbf{x}) = 2\mathbf{J}^T(\mathbf{x})\mathbf{v}(\mathbf{x}), \quad (8.2)$$

όπου $\mathbf{J}(\mathbf{x})$ είναι ο Ιακωβιανός πίνακας

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \frac{dv_1(\mathbf{x})}{dx_1} & \frac{dv_1(\mathbf{x})}{dx_2} & \dots & \frac{dv_1(\mathbf{x})}{dx_n} \\ \frac{dv_2(\mathbf{x})}{dx_1} & \frac{dv_2(\mathbf{x})}{dx_2} & \dots & \frac{dv_2(\mathbf{x})}{dx_n} \\ \vdots & \vdots & & \vdots \\ \frac{dv_N(\mathbf{x})}{dx_1} & \frac{dv_N(\mathbf{x})}{dx_2} & \dots & \frac{dv_N(\mathbf{x})}{dx_n} \end{bmatrix}.$$

Το k, j στοιχείο του Εσσιανού πίνακα υπολογίζεται από την εξίσωση

$$[\nabla^2 F(\mathbf{x})]_{k,j} = \frac{d^2 F(\mathbf{x})}{dx_k dx_j} = 2 \sum_{i=1}^N \left\{ \frac{dv_i(\mathbf{x}) dv_i(\mathbf{x})}{dx_k dx_j} + v_i(\mathbf{x}) \frac{d^2 v_i(\mathbf{x})}{dx_k dx_j} \right\},$$

η οποία σε μορφή πινάκων γίνεται

$$\nabla^2 F(\mathbf{x}) = 2\mathbf{J}^T(\mathbf{x})\mathbf{J}(\mathbf{x}) + 2\mathbf{S}(\mathbf{x}),$$

όπου

$$\mathbf{S}(\mathbf{x}) = \sum_{i=1}^N v_i(\mathbf{x}) \nabla^2 v_i(\mathbf{x}).$$

Αν υποθέσουμε ότι το $\mathbf{S}(\mathbf{x})$ είναι μικρό τότε ο Εσσιανός πίνακας προσεγγιστικά υπολογίζεται από την εξίσωση

$$\nabla^2 F(\mathbf{x}) = 2\mathbf{J}^T(\mathbf{x})\mathbf{J}(\mathbf{x}). \quad (8.3)$$

Με αντικατάσταση στην εξίσωση 8.3 των εξισώσεων 8.1 και 8.7 έχουμε μια παραλλαγή της μεθόδου του Newton η οποία ονομάζεται μέθοδος Gauss-Newton [7]

$$\begin{aligned}\mathbf{x}_{k+1} &= \mathbf{x}_k - [2\mathbf{J}^T(\mathbf{x}_k)\mathbf{J}(\mathbf{x}_k)]^{-1}2\mathbf{J}^T(\mathbf{x}_k)\mathbf{v}(\mathbf{x}_k) \\ &= \mathbf{x}_k - [\mathbf{J}^T(\mathbf{x}_k)\mathbf{J}(\mathbf{x}_k)]^{-1}\mathbf{J}^T(\mathbf{x}_k)\mathbf{v}(\mathbf{x}_k)\end{aligned}\quad (8.4)$$

Το πλεονέκτημα αυτής της μεθόδου είναι ότι δεν απαιτεί τον υπολογισμό των δεύτερων παραγώγων, όμως αντιμετωπίζει πρόβλημα αν ο πίνακας $\mathbf{H} = \mathbf{J}^T\mathbf{J}$ είναι μη αντιστρέψιμος. Αυτό αντιμετωπίζεται χρησιμοποιώντας την ακόλουθη μετατροπή για την προσέγγιση του Εσσιανού πίνακα

$$\mathbf{G} = \mathbf{H} + \mu\mathbf{I}.$$

Έτσι ο αλγόριθμος Levenberg-Marquardt γίνεται

$$\mathbf{x}_{k+1} = \mathbf{x}_k - [\mathbf{J}^T(\mathbf{x}_k)\mathbf{J}(\mathbf{x}_k) + \mu_k\mathbf{I}]^{-1}\mathbf{J}^T(\mathbf{x}_k)\mathbf{v}(\mathbf{x}_k), \quad (8.5)$$

ή

$$\Delta\mathbf{x}_k = -[\mathbf{J}^T(\mathbf{x}_k)\mathbf{J}(\mathbf{x}_k) + \mu_k\mathbf{I}]^{-1}\mathbf{J}^T(\mathbf{x}_k)\mathbf{v}(\mathbf{x}_k). \quad (8.6)$$

Ο αλγόριθμος αυτός έχει το χαρακτηριστικό ότι καθώς το μ_k αυξάνεται προσεγγίζει τον αλγόριθμο της πιο απότομης καθόδου για μικρό ρυθμό εκπαίδευσης:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \frac{1}{\mu_k}\mathbf{J}^T(\mathbf{x}_k)\mathbf{v}(\mathbf{x}_k) = \mathbf{x}_k - \frac{1}{2\mu_k}\nabla F(\mathbf{x}),$$

ενώ καθώς το μ_k πλησιάζει το 0 ο αλγόριθμος συμπεριφέρεται όπως η μέθοδος Gauss-Newton. Πιο αναλυτικά, ο αλγόριθμος ξεκινάει για μια μικρή τιμή του μ_k . Αν σε ένα βήμα του αλγορίθμου η τιμή που παίρνουμε για το δείκτη απόδοσης $F(\mathbf{x})$ είναι μικρότερη από αυτή του προηγούμενου βήματος, τότε το μ_k πολλαπλασιάζεται με έναν παράγοντα $\theta > 1$ και το βήμα αυτό επαναλαμβάνεται. Τότε ο δείκτης απόδοσης θα πρέπει να μειωθεί αφού θα έχουμε κάνει ένα βήμα προς την κατεύθυνση που ορίζεται από τη μέθοδο της πιο απότομης καθόδου. Αν σε ένα επόμενο βήμα δεν πάρουμε μικρότερη τιμή του δείκτη απόδοσης τότε το μ_k διαιρείται με το θ για το επόμενο βήμα έτσι ώστε ο αλγόριθμος να προσεγγίζει τη μέθοδο Gauss-Newton, η οποία θα πρέπει να παρέχει γρηγορότερη σύγκλιση. Ο αλγόριθμος αυτός αποτελεί έναν συμβιβασμό ανάμεσα στην ταχύτητα που προσφέρει η μέθοδος του Newton και στη βέβαιη σύγκλιση της μεθόδου της πιο απότομης καθόδου.

Στη συνέχεια θέλουμε να εφαρμόσουμε τον αλγόριθμο Levenberg-Marquardt σε προβλήματα εκπαίδευσης πολυστρωματικών νευρωνικών δικτύων. Από την εξίσωση 7.2 ο δείκτης απόδοσης για την εκπαίδευση πολυστρωματικών δικτύων είναι το μέσο τετραγωνικό σφάλμα. Αν κάθε επιθυμητή έξοδος έχει ίση πιθανότητα, το μέσο τετραγωνικό σφάλμα είναι ανάλογο με το άθροισμα των τετραγωνικών σφαλμάτων των Q επιθυμητών εξόδων του συνόλου των δεδομένων εκπαίδευσης

$$F(\mathbf{x}) = \sum_{q=1}^Q (\mathbf{t}_q - \mathbf{y}_q)^T (\mathbf{t}_q - \mathbf{y}_q) = \sum_{q=1}^Q \mathbf{e}_q^T \mathbf{e}_q = \sum_{q=1}^Q \sum_{j=1}^{S^M} (e_{j,q})^2 = \sum_{i=1}^N (v_i)^2, \quad (8.7)$$

όπου $e_{j,q}$ είναι το j στοιχείο του σφάλματος για το q ζεύγος εισόδου και επιθυμητής εξόδου.

Στη συνέχεια πρέπει να υπολογίσουμε τον Ιακωβιανό πίνακα. Για τον υπολογισμό αυτού του πίνακα χρησιμοποιείται μια παραλλαγή του αλγορίθμου της όπισθεν διάδοσης σφάλματος. Αρχικά πρέπει να υπολογίσουμε τις παραγώγους των σφαλμάτων. Το διάνυσμα των σφαλμάτων δίνεται από τη σχέση

$$\mathbf{v}^T = [v_1 \quad v_1 \quad \dots \quad v_N] = [e_{1,1} \quad e_{2,1} \quad \dots \quad e_{S^M,1} \quad e_{1,2} \quad \dots \quad e_{S^M,Q}],$$

και το διάνυσμα των βαρών είναι

$$\mathbf{x}^T = [x_1 \quad x_1 \quad \dots \quad x_N] = [w_{1,1}^1 \quad w_{1,2}^1 \quad \dots \quad w_{S^1,R}^1 \quad b_1^1 \quad \dots \quad b_{S^1}^1 \quad w_{1,1}^2 \quad \dots \quad b_{S^M}^M],$$

$$N = Q \cdot S^M$$

και

$$n = S^1(R+1) + S^2(S^1+1) + \dots + S^M(S^{M-1}+1).$$

Επομένως ο Ιακωβιανός πίνακας είναι ίσος με

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \frac{de_{1,1}}{dw_{1,1}^1} & \frac{de_{1,1}}{dw_{1,2}^1} & \dots & \frac{de_{1,1}}{dw_{S^1,R}^1} & \frac{de_{1,1}}{db_1^1} & \dots \\ \frac{de_{2,1}}{dw_{1,1}^1} & \frac{de_{2,1}}{dw_{1,2}^1} & \dots & \frac{de_{2,1}}{dw_{S^1,R}^1} & \frac{de_{2,1}}{db_1^1} & \dots \\ \vdots & \vdots & & \vdots & \vdots & \\ \frac{de_{S^M,1}}{dw_{1,1}^1} & \frac{de_{S^M,1}}{dw_{1,2}^1} & \dots & \frac{de_{S^M,1}}{dw_{S^1,R}^1} & \frac{de_{S^M,1}}{db_1^1} & \dots \\ \frac{de_{1,2}}{dw_{1,1}^1} & \frac{de_{1,2}}{dw_{1,2}^1} & \dots & \frac{de_{1,2}}{dw_{S^1,R}^1} & \frac{de_{1,2}}{db_1^1} & \dots \\ \vdots & \vdots & & \vdots & \vdots & \end{bmatrix}.$$

Στη συνέχεια ο αλγόριθμος Levenberg-Marquardt συνοψίζεται στα παρακάτω βήματα :

1. Τροφοδότησε όλες τις εισόδους στο δίκτυο και υπολόγισε τις αντίστοιχες εξόδους σύμφωνα με τις εξισώσεις

$$\mathbf{y}^0 = \mathbf{p},$$

$$\mathbf{y}^{m+1} = \mathbf{f}^{m+1}(\mathbf{W}^{m+1}\mathbf{y}^m + \mathbf{b}^{m+1}), \text{ για } m = 0, 1, \dots, M-1,$$

όπως στον αλγόριθμο της όπισθεν διάδοσης σφάλματος. Υπολόγισε επίσης τα σφάλματα

$$\mathbf{e}_q = \mathbf{t}_q - \mathbf{y}_q^M$$

και το άθροισμα των τετραγώνων των σφαλμάτων για όλες τις εισόδους χρησιμοποιώντας την εξίσωση 8.7.

2. Υπολόγισε τον Ιακωβιανό πίνακα $\mathbf{J}(\mathbf{x})$. Ο αλγόριθμος Levenberg-Marquardt αρχικοποιείται σύμφωνα με την εξίσωση

$$\mathbf{S}_q^M = \mathbf{F}^M(\mathbf{n}_q^M),$$

όπου το $\mathbf{F}^M(\mathbf{n}^m)$ ορίζεται από την εξίσωση 7.8. Κάθε στήλη του πίνακα \mathbf{S}_q^M τροφοδοτείται ανάστροφα στο δίκτυο σύμφωνα με την εξίσωση 7.9 και παράγει μια γραμμή του Ιακωβιανού πίνακα. Οι στήλες μπορούν επίσης να τροφοδοτηθούν μαζί

$$\mathbf{S}_q^M = F^m(\mathbf{n}_q^m)(\mathbf{W}^{m+1})^T \mathbf{S}_q^{m+1}.$$

Οι συνολικοί πίνακες για κάθε επίπεδο δημιουργούνται χωρίζοντας τους πίνακες που υπολογίστηκαν για κάθε είσοδο σύμφωνα με τη σχέση

$$\mathbf{S}^m = [\mathbf{S}_1^m | \mathbf{S}_2^m | \dots | \mathbf{S}_Q^m].$$

Στη συνέχεια υπολογίζονται τα στοιχεία του Ιακωβιανού πίνακα από την εξίσωση

$$[\mathbf{J}]_{h,i} = \frac{dv_h}{dx_i} = \frac{de_{k,q}}{dw_{i,j}^m} = \frac{de_{k,q}}{dn_{i,q}^m} x \frac{dn_{i,q}^m}{dw_{i,j}^m} = s_{i,h}^m x \frac{dn_{i,q}^m}{dw_{i,j}^m} = s_{i,h}^m x y_{j,q}^{m-1},$$

ή αν το x_i είναι πόλωση τότε

$$[\mathbf{J}]_{h,i} = \frac{dv_h}{dx_i} = \frac{de_{k,q}}{db_i^m} = \frac{de_{k,q}}{dn_{i,q}^m} x \frac{dn_{i,q}^m}{db_i^m} = s_{i,h}^m x \frac{dn_{i,q}^m}{db_i^m} = s_{i,h}^m,$$

$$s_{i,h}^m = \frac{dv_h}{dn_{i,q}^m} = \frac{de_{k,q}}{dn_{i,q}^m}, \text{ όπου } \mathbf{h} = (\mathbf{q} - \mathbf{1})\mathbf{S}^M + \mathbf{k}.$$

3. Έπειτα λύσε την εξίσωση 8.6 για να υπολογιστεί το Δx_k .
4. Τέλος υπολόγισε ξανά το άθροισμα των τετραγωνικών σφαλμάτων χρησιμοποιώντας τη σχέση $\mathbf{x}_k + \Delta \mathbf{x}_k$. Αν το καινούργιο άθροισμα είναι μικρότερο από αυτό που υπολογίστηκε στο βήμα 1, τότε διαίρεσε το μ με θ και επέστρεψε στο βήμα 1. Αν το άθροισμα δεν ελαττώνεται τότε πολλαπλασίασε το με μ με θ και επέστρεψε στο βήμα 3.

Ο αλγόριθμος συγκλίνει όταν η νόρμα του διανύσματος των μερικών παραγώγων έτσι όπως υπολογίζεται από την εξίσωση 8.7 είναι μικρότερη από κάποια προκαθορισμένη τιμή, ή όταν το άθροισμα των τετραγώνων μειώνεται έτσι ώστε να έχουμε ένα ανεκτό σφάλμα.

Μέρος ΙΙ

Πρακτικό Μέρος

Κεφάλαιο 9

Πρακτική εφαρμογή των Τεχνητών Νευρωνικών Δικτύων

Στην ενότητα αυτή θα παρουσιάσουμε μερικά βασικά προβλήματα εκπαίδευσης Τεχνητών Νευρωνικών Δικτύων και θα συγκρίνουμε τα αποτελέσματα του αλγορίθμου της όπισθεν διάδοσης σφάλματος σε αυτά τα προβλήματα όταν χρησιμοποιούμε εκπαίδευση ανά ομάδα προτύπων εισόδου και εκπαίδευση ανά πρότυπο εισόδου. Σε όλα τα προβλήματα χρησιμοποιήσαμε Νευρωνικά Δίκτυα με ένα κρυφό επίπεδο. Αρχικά έγιναν δοκιμές προκειμένου να βρούμε το βέλτιστο αριθμό των κρυφών νευρώνων που απαιτούνται για την εκπαίδευση του δικτύου έτσι ώστε να ελαχιστοποιείται το μέσο τετραγωνικό σφάλμα. Διατηρήσαμε μια σταθερή τιμή για το ρυθμό εκπαίδευσης και σταθερό αριθμό εποχών.

9.1 Παραδείγματα και Αποτελέσματα

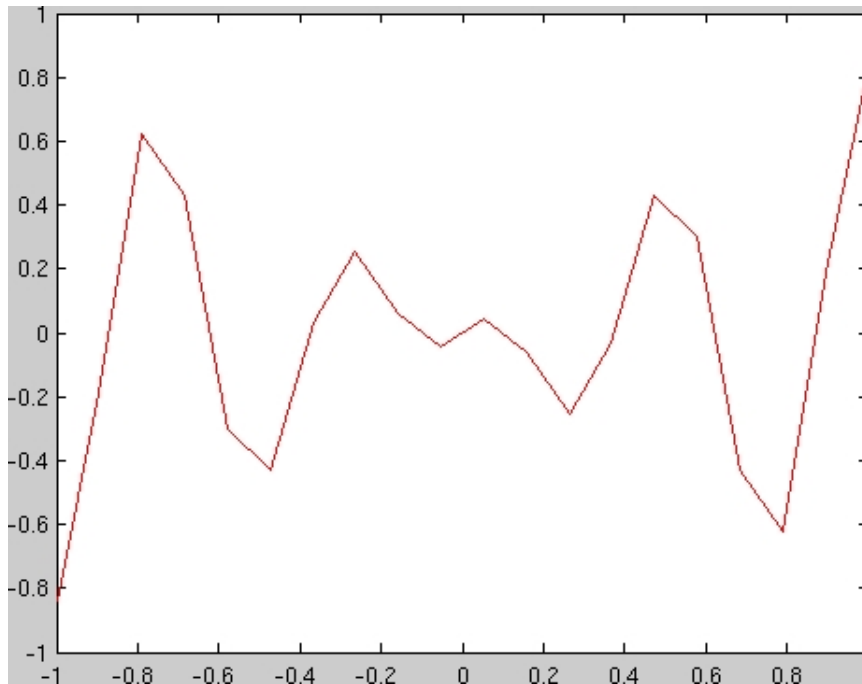
9.1.1 Προσέγγιση μιας συνεχούς συνάρτησης

Αρχικά θα εκπαιδεύσουμε ένα πολυστρωματικό δίκτυο Perceptron έτσι ώστε να προσαρμοστεί στα δεδομένα μιας συνάρτησης η οποία περιγράφεται από την εξίσωση

$$y = \sin(x) * \cos(4\pi x).$$

Η γραφική αναπαράσταση αυτής της συνάρτησης φαίνεται στο Σχήμα 9.1. Για την εκπαίδευση του νευρωνικού δικτύου χρησιμοποιήθηκαν ως δεδομένα εκπαίδευσης 20 ισαπέχουσες τιμές της συνάρτησης στο διάστημα $[-1, 1]$. Αρχικά εκπαιδεύσαμε ένα δίκτυο τριών επιπέδων για διαφορετικό αριθμό κρυφών νευρώνων κάθε φορά χρησιμοποιώντας τον αλγόριθμο Levenberg-Marquardt. Το αποδεκτό σφάλμα του δικτύου επιλέχθηκε να είναι ίσο με 0.01. Τα αποτελέσματα παρουσιάζονται στον πίνακα 9.1. Η απόδοση του δικτύου ορίζεται ως το μέσο τετραγωνικό σφάλμα της εξόδου του δικτύου.

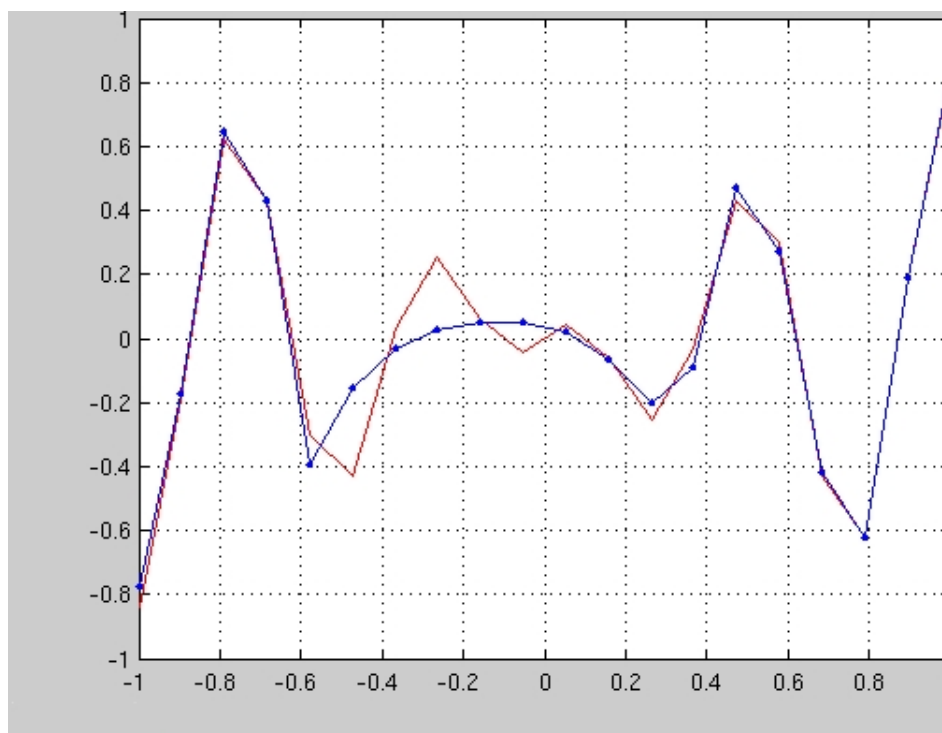
Παρατηρούμε ότι όσο αυξάνεται ο αριθμός των νευρώνων, το μέσο τετραγωνικό σφάλμα μειώνεται. Για 2 και 4 κρυφούς νευρώνες το δίκτυο δε συγκλίνει στον επιθυμητό στόχο. Όταν χρησιμοποιήσαμε 5 κρυφούς νευρώνες, το δίκτυο εκπαιδεύτηκε σύμφωνα με το στόχο



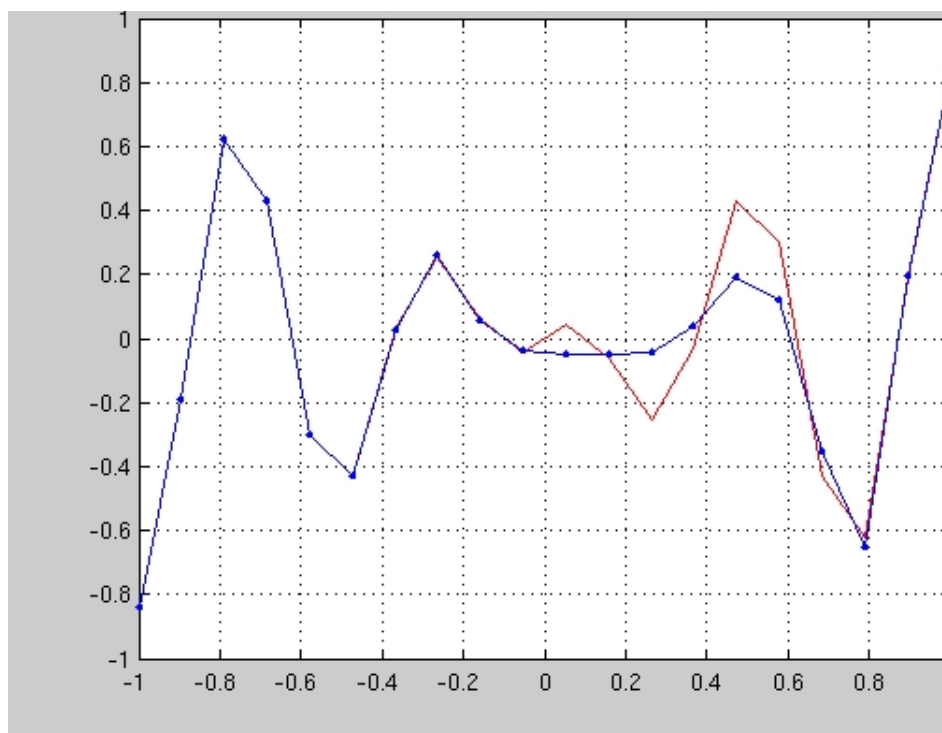
Σχήμα 9.1: Γραφική αναπαράσταση της συνάρτησης $y = \sin(x) * \cos(4\pi x)$.

που του είχαμε δώσει, δηλαδή να έχουμε μικρότερο τετραγωνικό σφάλμα από 0.01. Στο Σχήμα 9.2 με μπλε γραμμή αναπαρίσταται η έξοδος του δικτύου όταν χρησιμοποιήθηκαν 5 νευρώνες στο κρυφό επίπεδο, ενώ με κόκκινη γραμμή αναπαρίσταται η επιθυμητή έξοδος που προσεγγίζει τη συνάρτηση. Παρατηρούμε ότι η προσαρμογή της συνάρτησης δεν είναι ακριβής, δηλαδή για κάποια δεδομένα εισόδου το δίκτυο αυτό δεν παράγει τον επιθυμητό στόχο. Η προσαρμογή είναι πιο καλή όταν χρησιμοποιούμε 10 νευρώνες στο κρυφό επίπεδο όπως φαίνεται στο Σχήμα 9.3, ενώ όταν χρησιμοποιήθηκαν 15 κρυφοί νευρώνες η έξοδος του δικτύου προσαρμόζεται ακριβώς στα δεδομένα της συνάρτησης όπως παρατηρούμε στο Σχήμα 9.4 Στο Σχήμα 9.5 βλέπουμε πώς μεταβάλλεται το μέσο τετραγωνικό σφάλμα ανάλογα με τον αριθμό των νευρώνων στο κρυφό επίπεδο.

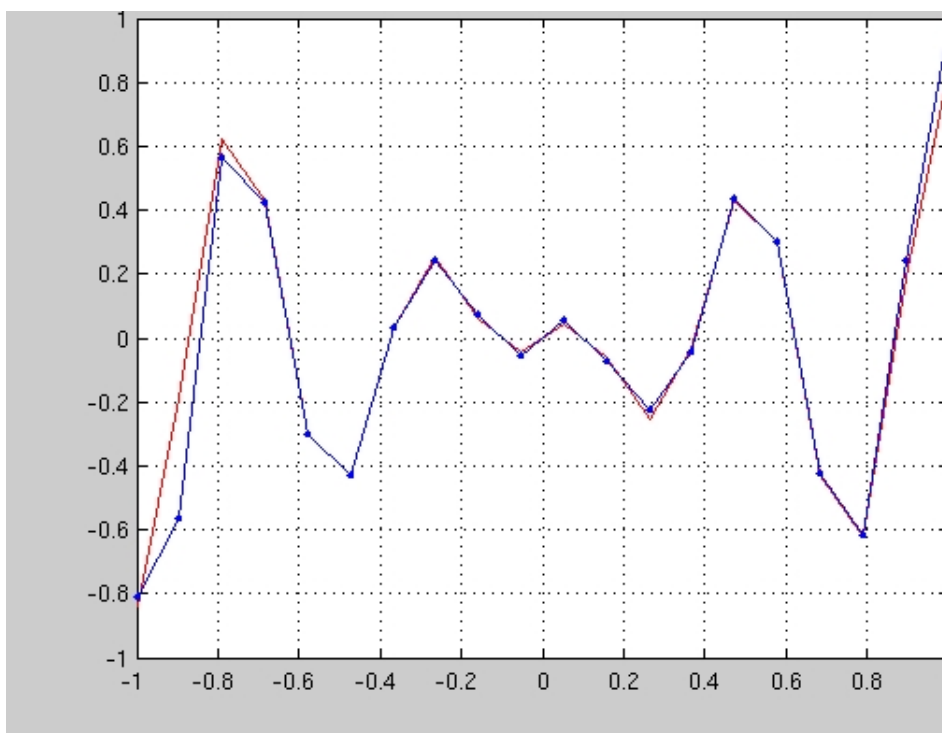
Επομένως χρησιμοποιούμε σαν βέλτιστη τοπολογία ένα Τεχνητό Νευρωνικό Δίκτυο 1-15-1. Στη συνέχεια εκτελέσαμε 100 επαναλήψεις για την εκπαίδευση του δικτύου χρησιμοποιώντας τέσσερις διαφορετικές μεθόδους εκπαίδευσης, τις `traingd`, `trainlm`, `trainr` και `trainc`. Οι δύο πρώτες μέθοδοι `traingd` και `trainlm` πραγματοποιούν εκπαίδευση ανά ομάδα προτύπων εισόδου σύμφωνα με τον βασικό αλγόριθμο της όπισθεν διάδοσης σφάλματος και τον αλγόριθμο Levenberg-Marquardt αντίστοιχα. Η μέθοδος `trainr` πραγματοποιεί εκπαίδευση ανά πρότυπο εισόδου, επιλέγοντας τυχαία κάθε φορά το πρότυπο το οποίο θα τροφοδοτηθεί στο δίκτυο, ενώ η μέθοδος `trainc` πραγματοποιεί εκπαίδευση ανά πρότυπο εισόδου χρησιμοποιώντας όμως κύκλους εκπαίδευσης στους οποίους κάθε πρότυπο παρουσιάζεται μία μόνο φορά. Στην περίπτωση αυτή μειώσαμε το αποδεκτό σφάλμα από 0.01 σε 0.001. Για κάθε μέθοδο μετράμε την επιτυχία εκπαίδευσης του δικτύου, η οποία ορίζεται ως ο αριθμός των επιτυχιών εκπαίδευσης του δικτύου σύμφωνα με τα κριτήρια που του έχουμε



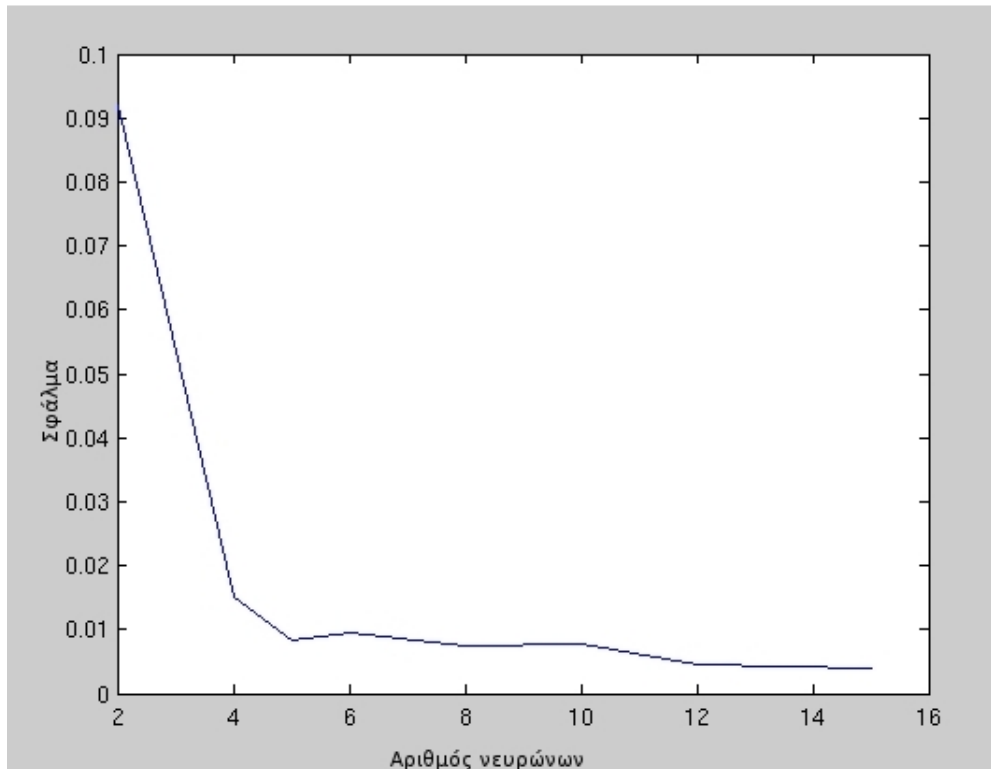
Σχήμα 9.2: Έξοδος του δικτύου για 5 κρυφούς νευρώνες.



Σχήμα 9.3: Έξοδος του δικτύου για 10 κρυφούς νευρώνες.



Σχήμα 9.4: Έξοδος του δικτύου για 15 κρυφούς νευρώνες.



Σχήμα 9.5: Μέσο τετραγωνικό σφάλμα συναρτήσει του αριθμού των κρυφών νευρώνων.

Πίνακας 9.1: Αποτελέσματα εκπαίδευσης για τη συνάρτηση y χρησιμοποιώντας διαφορετικό αριθμό κρυφών νευρώνων

Αριθμός κρυφών νευρώνων	Ρυθμός εκπαίδευσης	Αποδεκτό σφάλμα	Απόδοση
2	0.01	0.01	0.0921
4	0.01	0.01	0.0151
5	0.01	0.01	0.0082
6	0.01	0.01	0.0095
8	0.01	0.01	0.0076
10	0.01	0.01	0.0077
12	0.01	0.01	0.0045
15	0.01	0.01	0.0040

δώσει, μετά από 100 δοκιμές. Στον πίνακα 9.2 παρουσιάζονται τα συγκριτικά αποτελέσματα των μεθόδων. Ως M ορίζεται η μέση τιμή του τετραγωνικού σφάλματος και σ η τυπική απόκλιση για 100 επαναλήψεις.

Πίνακας 9.2: Συγκριτικά αποτελέσματα των διαφόρων μεθόδων εκπαίδευσης για μια συνεχή συνάρτηση

Αλγόριθμος	M	σ	Επιτυχία
traingd	0.0275	0.0094	-
trainlm	0.0279	0.1033	85%
trainr	0.00092149	0.000068194	100%
trainc	0.00099915	0.000000551	100%

Παρατηρούμε ότι όταν πραγματοποιήθηκε εκπαίδευση ανά πρότυπο εισόδου με τη μέθοδο trainr, το δίκτυο είχε 100% επιτυχία, ενώ όταν η εκπαίδευση έγινε ανά ομάδα προτύπων η επιτυχία εκπαίδευσης του δικτύου ήταν μικρότερη, 85% όταν χρησιμοποιήθηκε ο αλγόριθμος Levenberg-Marquardt, ενώ όταν χρησιμοποιήσαμε το βασικό αλγόριθμο της όπισθεν διάδοσης σφάλματος το δίκτυο απέτυχε να εκπαιδευτεί. Αυτό συμβαίνει επειδή η εκπαίδευση ανά τυχαίο πρότυπο εισόδου έχει το πλεονέκτημα να ξεφεύγει από τοπικά ελάχιστα κατά τη διαδικασία εκπαίδευσης. Για την εκπαίδευση με τον κλασικό αλγόριθμο όπισθεν διάδοσης σφάλματος όταν χρησιμοποιήθηκε μεγαλύτερος ρυθμός εκπαίδευσης, 0.12, η εκπαίδευση είχε 100% επιτυχία. Στον πίνακα 9.3 παρουσιάζεται η απόδοση του αλγορίθμου για διαφορετικές τιμές του ρυθμού εκπαίδευσης.

Στα επόμενα παραδείγματα δε θα χρησιμοποιήσουμε τον κλασικό αλγόριθμο της όπισθεν διάδοσης σφάλματος, καθώς αυτός είναι σχετικά αργός και υπερκαλύπτεται από τη χρήση του αλγορίθμου Levenberg-Marquardt.

Στο Σχήμα 9.6 φαίνεται η απόδοση του αλγορίθμου όταν χρησιμοποιείται εκπαίδευση ανά ομάδα προτύπων εισόδου για 1000 εποχές, ενώ στο Σχήμα 9.7 παρουσιάζεται η απόδοση

Πίνακας 9.3: Απόδοση του κλασικού αλγορίθμου όπισθεν διάδοσης σφάλματος για διαφορετικές τιμές του ρυθμού εκπαίδευσης

Ρυθμός εκπαίδευσης	Αποδεκτό σφάλμα	Απόδοση
0.01	0.001	0.0176
0.05	0.001	0.0019
0.06	0.001	0.0018
0.07	0.001	0.0017
0.1	0.001	0.0012
0.12	0.001	0.00099

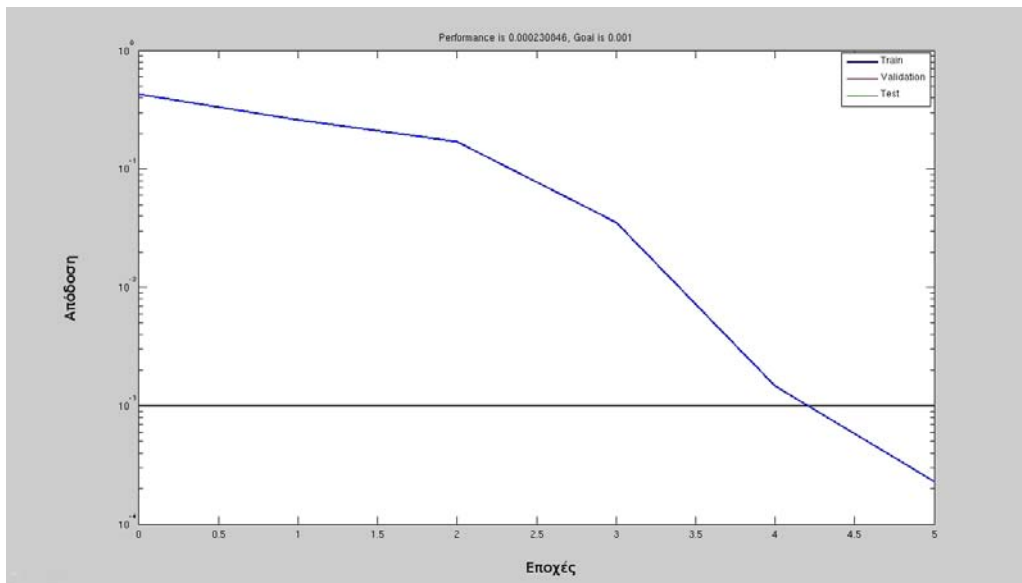
του αλγορίθμου όταν έχουμε εκπαίδευση ανά πρότυπο εισόδου. Παρατηρούμε ότι όταν έχουμε εκπαίδευση ανά ομάδα προτύπων το μέσο τετραγωνικό σφάλμα μειώνεται σε κάθε εποχή μέχρι να συγκλίνει στο επιθυμητό αποτέλεσμα, γι'αυτό αναπαρίσταται συνήθως ως μία καμπύλη η οποία μειώνεται συνεχώς με το χρόνο. Αντίθετα όταν έχουμε εκπαίδευση ανά πρότυπο εισόδου, το μέσο τετραγωνικό σφάλμα διαφέρει σε κάθε εποχή και μπορεί να είναι μικρότερο ή μεγαλύτερο από την προηγούμενη εποχή, όμως συνολικά το σφάλμα μειώνεται με το χρόνο. Αυτό συμβαίνει επειδή το μέσο τετραγωνικό σφάλμα υπολογίζεται για ένα πρότυπο που τροφοδοτείται στο δίκτυο σε κάθε εποχή. Η αναπροσαρμογή των βαρών γίνεται μετά την τροφοδότηση του κάθε προτύπου στο δίκτυο, επομένως με την παρουσίαση του επόμενου προτύπου το μέσο τετραγωνικό σφάλμα της εξόδου μπορεί να είναι μεγαλύτερο ή και μικρότερο από την προηγούμενη τιμή του. Καθώς όμως τα πρότυπα παρουσιάζονται στο δίκτυο και γίνεται η αναπροσαρμογή των βαρών, το συνολικό σφάλμα του δικτύου τείνει να συγκλίνει προς το ελάχιστο.

9.1.2 Το πρόβλημα της περιττής ισοτιμίας

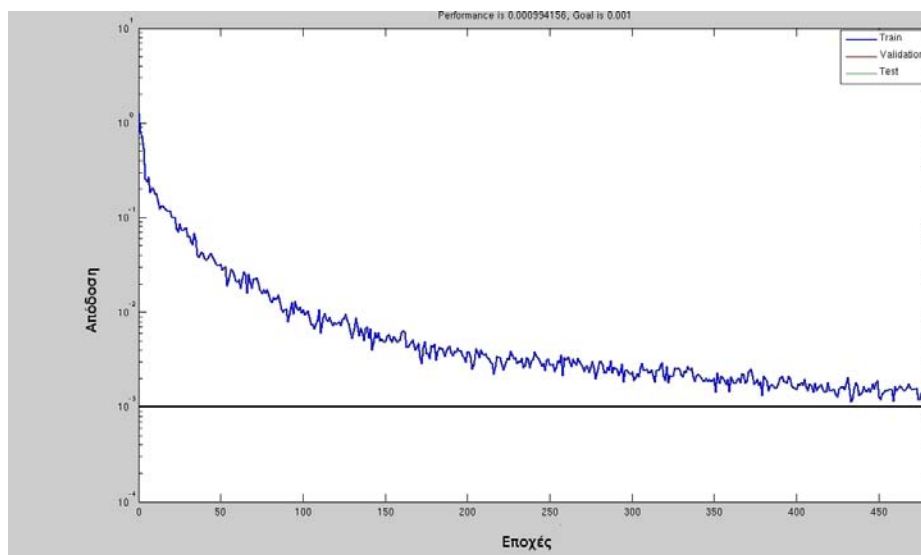
Το πρόβλημα της περιττής ισοτιμίας είναι παρόμοιο με το πρόβλημα της αποκλειστικής διάζευξης, αλλά η διαδικασία εκπαίδευσης ενός νευρωνικού δικτύου το οποίο επιλύει το πρόβλημα αυτό είναι πιο δύσκολη. Η συνάρτηση αυτή δέχεται τρεις εισόδους και δίνει μία έξοδο. Οι εισοδοί και η έξοδος μπορεί να είναι 0 ή 1 μόνον και ισχύει ο εξής περιορισμός: Εάν οι εισοδοί αποτελούνται από άρτιο πλήθος 1, τότε η έξοδος είναι 0, ενώ εάν το πλήθος των 1 είναι περιττό, τότε η έξοδος είναι 1. Οι όροι αυτοί συνοψίζονται στον Πίνακα 9.4.

Στην περίπτωση αυτή χρησιμοποιήσαμε ένα Τεχνητό Νευρωνικό Δίκτυο με 5 κρυφούς νευρώνες. Στη συνέχεια εκτελέσαμε 100 επαναλήψεις για την εκπαίδευση του δικτύου χρησιμοποιώντας τις μεθόδους εκπαίδευσης `trainlm`, `trainr` και `trainc`. Ο ρυθμός εκπαίδευσης για κάθε επανάληψη τίθεται ίσος με 0.01, ενώ ο αλγόριθμος εκπαίδευσης συγκλίνει όταν το μέσο τετραγωνικό σφάλμα γίνει μικρότερο ή ίσο από 0.01. Τα αποτελέσματα αυτών των μεθόδων φαίνονται στον πίνακα 9.5.

Όπως παρατηρούμε οι μέθοδοι `trainlm` και `trainr` έχουν σχεδόν το ίδιο ποσοστό επιτυ-



Σχήμα 9.6: Μέσο τετραγωνικό σφάλμα κατά τη διάρκεια εκπαίδευσης ανά ομάδα προτύπων εισόδου.



Σχήμα 9.7: Μέσο τετραγωνικό σφάλμα κατά τη διάρκεια εκπαίδευσης ανά πρότυπο εισόδου.

Πίνακας 9.4: Πίνακας αληθείας του προβλήματος της περιττής ισοτιμίας

Είσοδος 1	Είσοδος 2	Είσοδος 3	Έξοδος
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Πίνακας 9.5: Συγκριτικά αποτελέσματα των διαφόρων μεθόδων εκπαίδευσης για το πρόβλημα της περιττής ισοτιμίας

Αλγόριθμος	M	σ	Επιτυχία
trainlm	0.0968	0.2146	76%
trainr	0.0219	0.0240	75%
trainc	0.0199	0.0240	80%

χίας με ελαφρύ προβάδισμα της πρώτης. Αντίθετα η μέθοδος trainc παρουσιάζει καλύτερα αποτελέσματα με ποσοστό επιτυχίας εκπαίδευσης 80%.

9.1.3 Αναγνώριση προτύπων

Στη συνέχεια εξετάζουμε ένα πρόβλημα εκπαίδευσης ενός πολυστρωματικού δικτύου το οποίο αναγνωρίζει τους αριθμούς από 0 ως 9. Για την εκπαίδευση χρησιμοποιήθηκε νευρωνικό δίκτυο με 9 κρυφούς νευρώνες. Ο ρυθμός εκπαίδευσης σε κάθε περίπτωση όπως και πριν είναι ίσος με 0.01 και ο αλγόριθμος εκπαίδευσης τερματίζεται όταν το μέσο τετραγωνικό σφάλμα γίνει μικρότερο ή ίσο από 0.01. Στην περίπτωση αυτή όπως φαίνεται στον πίνακα 9.6 ο αλγόριθμος Levenberg-Marquardt έχει ποσοστό επιτυχίας εκπαίδευσης 98% ενώ όταν χρησιμοποιήσαμε εκπαίδευση ανά πρότυπο εισόδου, ο αλγόριθμος δεν κατάφερε να συγκλίνει σε καμία περίπτωση.

Πίνακας 9.6: Συγκριτικά αποτελέσματα των διαφόρων μεθόδων εκπαίδευσης για το πρόβλημα της αναγνώρισης αριθμών

Αλγόριθμος	M	σ	Επιτυχία
trainlm	0.0078	0.0031	98%
trainr	0.0283	0.0069	-
trainc	0.0298	0.0097	-

9.1.4 Το πρόβλημα Sonar

Είναι πρακτικό να εφαρμόσουμε ακόμη ένα πρόβλημα εκπαίδευσης έτσι ώστε να συγκρίνουμε τη συμπεριφορά των διαφόρων αλγορίθμων εκπαίδευσης που εξετάσαμε. Στο πρόβλημα Sonar τα δεδομένα εκπαίδευσης προκύπτουν από τη χρήση αισθητήρων σε ένα ειδικό ρομπότ. Το ρομπότ αυτό προσπαθεί να αποφασίσει αν ένα εμπόδιο είναι βράχος ή νάρκη. Το δίκτυο το οποίο χρησιμοποιήθηκε για την επίλυση αυτού του προβλήματος είναι ένα πολυστρωματικό δίκτυο τριών επιπέδων με 8 κρυφούς νευρώνες. Επειδή το σύνολο των δεδομένων εκπαίδευσης είναι αρκετά μεγάλο για την πιο αποτελεσματική και γρήγορη εκπαίδευση του δικτύου χρησιμοποιήσαμε σύνολα αξιολόγησης και δοκιμής.

Τα αποτελέσματα της εκπαίδευσης όταν χρησιμοποιήσαμε εκπαίδευση ανά ομάδα προτύπων εισόδου και εκπαίδευση ανά πρότυπο εισόδου φαίνονται στον πίνακα 9.7. Στην περίπτωση αυτή ο αλγόριθμος που πραγματοποιεί την εκπαίδευση ανά ομάδα προτύπων έχει μεγαλύτερη επιτυχία από την εκπαίδευση ανά πρότυπο εισόδου. Παρόλα αυτά παρατηρούμε ότι η διακύμανση του σφάλματος στη περίπτωση που έχουμε εκπαίδευση ανά ομάδα προτύπων εισόδου είναι πολύ μεγαλύτερη, ενώ στην εκπαίδευση ανά πρότυπο εισόδου το μέσο τετραγωνικό σφάλμα παίρνει τιμές κοντά στο 0.01.

Πίνακας 9.7: Συγκριτικά αποτελέσματα των μεθόδων εκπαίδευσης για το πρόβλημα Sonar

Αλγόριθμος	M	σ	Επιτυχία
trainlm	0.1076	0.1659	62%
trainr	0.0112	0.0015	13%

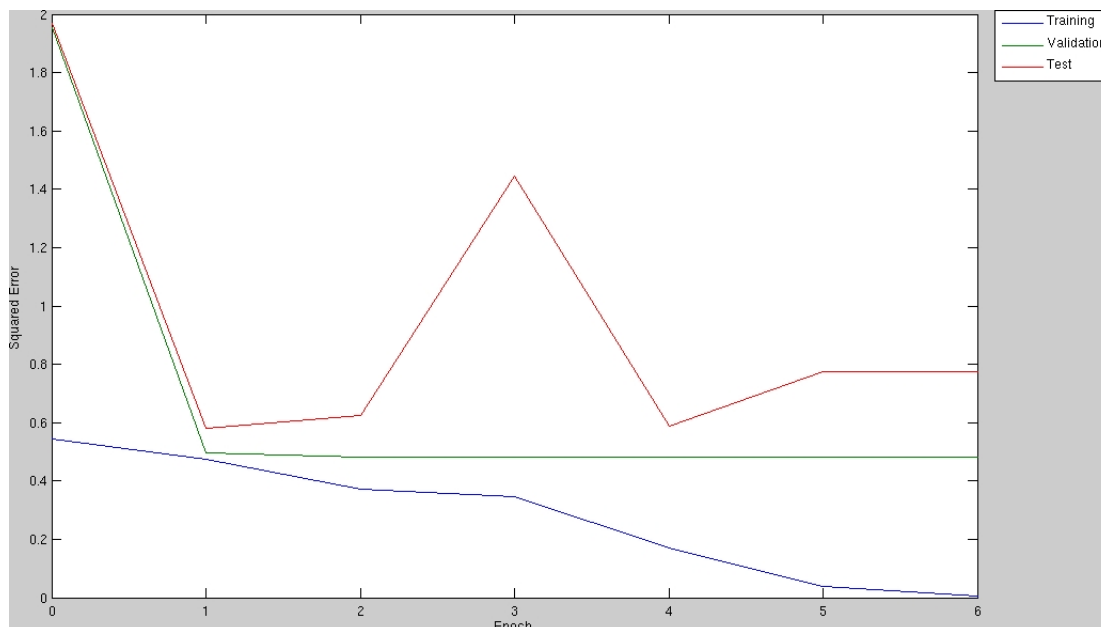
Χρησιμοποιώντας όλο το σύνολο των δεδομένων για την εκπαίδευση δεν μπορούμε να έχουμε άποψη σχετικά με το αν συμβαίνει υπερπροσαρμογή του δικτύου. Για το λόγο αυτό μπορούμε να χωρίσουμε το σύνολο των δεδομένων εκπαίδευσης σε τρία υποσύνολα. Το πρώτο υποσύνολο αποτελείται από τα δεδομένα που θα χρησιμοποιηθούν για την εκπαίδευση του δικτύου. Το δεύτερο υποσύνολο αποτελεί το σύνολο αξιολόγησης του δικτύου και το μέγεθός του είναι το ένα τέταρτο από το αρχικό σύνολο των δεδομένων. Το τρίτο υποσύνολο, το σύνολο δοκιμής, το οποίο αποτελεί επίσης το ένα τέταρτο του αρχικού συνόλου, δε χρησιμοποιείται στην εκπαίδευση αλλά τροφοδοτείται στο δίκτυο μετά από αυτήν για να αξιολογηθεί η αποδοτικότητα του δικτύου. Όταν έχουμε τέτοια σύνολα η εκπαίδευση μπορεί επίσης να τερματιστεί όταν το σφάλμα του συνόλου αξιολόγησης αυξηθεί πολύ.

Τα αποτελέσματα της εκπαίδευσης όταν χρησιμοποιούμε σύνολα αξιολόγησης και δοκιμής συνοψίζονται στον πίνακα 9.8. Εδώ η εκπαίδευση ανά πρότυπο εισόδου έχει σαφώς καλύτερα αποτελέσματα με ποσοστό επιτυχίας 99%. Στο σημείο αυτό μπορούμε να υπολογίσουμε την ακρίβεια με την οποία το δίκτυο αναγνωρίζει τα δεδομένα εισόδου αφού αυτό έχει εκπαιδευτεί. Όταν λοιπόν τροφοδοτούμε το δίκτυο με ολόκληρο το σύνολο των δεδομένων εκπαίδευσης, τότε η ακρίβεια του δικτύου δηλαδή το ποσοστό των προτύπων που έχουν την επιθυμητή έξοδο είναι 5% στην περίπτωση που η εκπαίδευση ήταν ανά ομάδα προτύπων, και 52% όταν έγινε εκπαίδευση ανά πρότυπο εισόδου. Το ποσοστό στην πρώτη περίπτωση

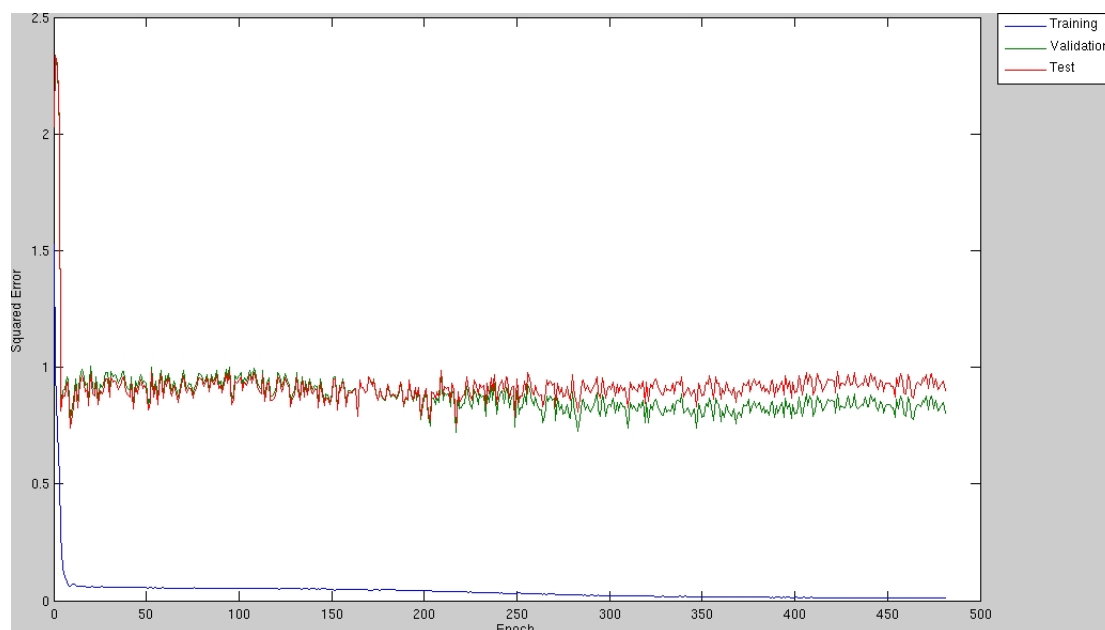
Πίνακας 9.8: Συγκριτικά αποτελέσματα των μεθόδων εκπαίδευσης με σύνολα αξιολόγησης και δοκιμής για το πρόβλημα Sonar

Αλγόριθμος	M	σ	Επιτυχία
trainlm	0.2347	0.2628	34%
trainr	0.0096	0.00047	99%

είναι απογοητευτικό, το δίκτυο δεν έχει καταφέρει να αναγνωρίσει τα πρότυπα. Η συμπεριφορά αυτή μπορεί να εξηγηθεί αν παρατηρήσουμε το Σχήμα 9.8, το οποίο μας δείχνει τη συμπεριφορά του αλγορίθμου κατά την εκπαίδευση ανά ομάδα προτύπων εισόδου. Με μπλε γραμμή απεικονίζεται το μέσο τετραγωνικό σφάλμα του συνόλου εκπαίδευσης, με πράσινη του συνόλου αξιολόγησης και με κόκκινη του συνόλου δοκιμής κατά τη διάρκεια της εκπαίδευσης του δικτύου. Στο Σχήμα αυτό το σφάλμα του συνόλου αξιολόγησης παρουσιάζεται να έχει μεγάλη απόκλιση από το σφάλμα του συνόλου δοκιμής. Αντίθετα το Σχήμα 9.9 μας δείχνει τη συμπεριφορά του αλγορίθμου κατά την εκπαίδευση ανά πρότυπο εισόδου. Το αποτέλεσμα εδώ είναι λογικό, καθώς το σφάλμα του συνόλου αξιολόγησης και του συνόλου δοκιμής έχουν παρόμοια χαρακτηριστικά και δεν παρουσιάζεται κάποια σημαντική υπερπροσαρμογή στο δίκτυο. Όταν χρησιμοποιήθηκε εκπαίδευση ανά πρότυπο εισόδου, το δίκτυο κατάφερε να αναγνωρίσει όλα τα πρότυπα τα οποία χρησιμοποιήθηκαν για την εκπαίδευση του δικτύου, δηλαδή το 50% του αρχικού συνόλου των δεδομένων. Παρόλα αυτά δεν καταφέρνει να αναγνωρίσει άγνωστα πρότυπα που τροφοδοτούνται για πρώτη φορά στο δίκτυο.



Σχήμα 9.8: Μέσο τετραγωνικό σφάλμα κατά τη διάρκεια εκπαίδευσης ανά ομάδα προτύπων εισόδου για το πρόβλημα Sonar.



Σχήμα 9.9: Μέσο τετραγωνικό σφάλμα κατά τη διάρκεια εκπαίδευσης ανά πρότυπο εισόδου για το πρόβλημα Sonar.

9.2 Συμπεράσματα

Τα πολυστρωματικά νευρωνικά δίκτυα είναι ικανά να εκτελέσουν σχεδόν οποιοδήποτε γραμμικό ή μη γραμμικό υπολογισμό και μπορούν να προσομοιώσουν οποιαδήποτε τυχαία συνάρτηση αρκετά καλά.

Ο κλασικός αλγόριθμος της όπισθεν διάδοσης σφάλματος είναι γενικά πολύ αργός γιατί απαιτεί μικρές τιμές του ρυθμού εκπαίδευσης έτσι ώστε να έχουμε σταθερή εκπαίδευση. Ο αλγόριθμος Levenberg-Marquardt μπορεί να χρησιμοποιηθεί μόνο για μικρά ή μεσαίου μεγέθους δίκτυα γιατί έχει μεγάλες απαιτήσεις σε μνήμη. Οι αλγόριθμοι εκπαίδευσης ανά πρότυπο εισόδου μπορούν να εκπαιδεύσουν αποδοτικά μεγάλου μεγέθους τεχνητά νευρωνικά δίκτυα. Επίσης παρέχουν τη δυνατότητα συνεχούς εκπαίδευσης με νέα δεδομένα μετά από την πρώτη εκπαίδευση του δικτύου.

Στα προβλήματα που είδαμε οι τεχνικές εκπαίδευσης ανά πρότυπο εισόδου στις περισσότερες περιπτώσεις ήταν πιο αποτελεσματικές όμως η εκπαίδευση ήταν πιο αργή σε σύγκριση με τις τεχνικές εκπαίδευσης ανά ομάδα προτύπων εισόδου. Επίσης η επιφάνεια σφάλματος ενός μη γραμμικού νευρωνικού δικτύου είναι περισσότερο πολύπλοκη από την επιφάνεια σφάλματος ενός γραμμικού δικτύου καθώς μπορεί να περιέχει πολλά τοπικά ελάχιστα. Οι τεχνικές εκπαίδευσης ανά πρότυπο εισόδου λόγω της στοχαστικότητας που τις διακρίνουν μπορούν να ξεφύγουν από τοπικά ακρότατα.

Σε κάθε περίπτωση, η επιλογή το κατάλληλου αλγορίθμου εκπαίδευσης εξαρτάται αποκλειστικά από το εκάστοτε πρόβλημα και την πολυπλοκότητα του προβλήματος αυτού.

Μέρος III

ΠΑΡΑΡΤΗΜΑ

Για την εφαρμογή του πρακτικού μέρους αυτής της εργασίας χρησιμοποιήθηκε το λογισμικό Matlab. Παρακάτω δίνεται ο κώδικας που υλοποιήθηκε για την εκπαίδευση ενός Τεχνητού Νευρωνικού Δικτύου με τον αλγόριθμο Levenberg-Marquardt:

```
function [apodosi, success, net1, accuracy1]=nnt_lm(P, T, hidden, tp)

if nargin < 3
    error('Not enough arguments.Arguments: P(input),T(target),
    hidden(number of neurons in the hidden layer),
    tp(vector of network parmeters)')

elseif nargin == 3
    tp=[];
    df=50; %Default values for the parameters.
    lr=0.01;
    me=1000;
    eg=0.001;
    iterations=3;

else
    df = tp(1); % Display frequency
    lr = tp(2); % Learning rate
    me = tp(3); % Maximum epochs
    eg = tp(4); % Error goal
    iterations = tp(5); % Number of iterations
end

%Create a feed-forward neural network
net=newff(minmax(P), [hidden,length(T(:,1))], {'tansig','tansig'}, 'trainlm');

net.trainParam.show = df;
net.trainParam.lr = lr;
net.trainParam.epochs = me;
net.trainParam.goal = eg;
%net.performFcn = 'sse';

success=0;
for i=1:iterations

    net=init(net);
    [net1,tr]=train(net,P,T); %Train the network
    [k l] = size(tr.perf);
```

```
apodosi(i)=tr.perf(1);

    if apodosi(i) <= eg
        success = success+1; %Give the number of succesful trainings
    end
end

a=sim(net1,P);
accuracy1=accuracy(a,T); %Give the accuracy of the network
```

Στη συνέχεια παρατίθεται ο κώδικας που χρησιμοποιήθηκε για την εκπαίδευση με τον κλασικό αλγόριθμο της όπισθεν διάδοσης σφάλματος:

```
function [apodosi,success,net1,accuracy1]=nnt_gd(P,T,hidden,tp)

if nargin < 3
    error('Not enough arguments.Arguments: P(input),T(target),
    hidden(number of neurons in the hidden layer),
    tp(vector of network parmeters)')

elseif nargin == 3
    tp=[];
    df=50; %Default values for the parameters.
    lr=0.01;
    me=1000;
    eg=0.001;
    iterations=3;

else
    df = tp(1); % Display frequency
    lr = tp(2); % Learning rate
    me = tp(3); % Maximum epochs
    eg = tp(4); % Error goal
    iterations = tp(5); % Number of iterations
end

%Create a feed-forward neural network
net=newff(minmax(P),[hidden,length(T(:,1))],{'tansig','tansig'},'traingd');

net.trainParam.show = df;
```

```

net.trainParam.lr = lr;
net.trainParam.epochs = me;
net.trainParam.goal = eg;
%net.performFcn = 'sse';

success=0;
for i=1:iterations

    net=init(net);
    [net1,tr]=train(net,P,T);    %Train the network
    [k l] = size(tr.perf);

    apodosi(i)=tr.perf(1);

    if apodosi(i) <= eg
        success = success+1;    %Give the number of succesful trainings
    end
end

a=sim(net1,P);
accuracy1=accuracy(a,T);    %Give the accuracy of the network

```

Ομοίως δίνεται ο κώδικας που χρησιμοποιήθηκε για την εκπαίδευση ανά πρότυπο εισόδου με τη μέθοδο `trainr`:

```

function [apodosi,success,net1,accuracy1]=nnt_on(P,T,hidden,tp)

if nargin < 3
    error('Not enough arguments.Arguments: P(input),T(target),
        hidden(number of neurons in the hidden layer),
        tp(vector of network parmeters)')

elseif nargin == 3
    tp=[];
    df=50;                %Default values for the parameters.
    lr=0.01;
    me=1000;
    eg=0.001;
    iterations=3;

else

```

```
df      = tp(1); % Display frequency
lr      = tp(2); % Learning rate
me      = tp(3); % Maximum epochs
eg      = tp(4); % Error goal
iterations = tp(5); % Number of iterations
end

%Create a feed-forward neural network
net=newff(minmax(P), [hidden, length(T(:,1))], {'tansig', 'tansig'}, 'trainr');

net.trainParam.show = df;
net.trainParam.lr = lr;
net.trainParam.epochs = me;
net.trainParam.goal = eg;
%net.performFcn = 'sse';

success=0;
for i=1:iterations

    net=init(net);
    [net1,tr]=train(net,P,T);    %Train the network
    [k l] = size(tr.perf);

    apodosi(i)=tr.perf(l);

    if apodosi(i) <= eg
        success = success+1;    %Give the number of succesful trainings
    end
end

a=sim(net1,P);
accuracy1=accuracy(a,T);    %Give the accuracy of the network
```

Ο κώδικας που χρησιμοποιήθηκε για την εκπαίδευση ανά πρότυπο εισόδου με τη μέθοδο `trainc` είναι ο εξής:

```
function [apodosi, success, net1, accuracy1]=nnt_on2(P, T, hidden, tp)

if nargin < 3
    error('Not enough arguments.Arguments: P(input), T(target),
        hidden(number of neurons in the hidden layer),
```

```
    tp(vector of network parameters)')

elseif nargin == 3
    tp=[];
    df=50;           %Default values for the parameters.
    lr=0.01;
    me=1000;
    eg=0.001;
    iterations=3;

else
    df    = tp(1); % Display frequency
    lr    = tp(2); % Learning rate
    me    = tp(3); % Maximum epochs
    eg    = tp(4); % Error goal
    iterations = tp(5); % Number of iterations
end

%Create a feed-forward neural network
net=newff(minmax(P), [hidden, length(T(:,1))], {'tansig', 'tansig'}, 'trainc');

net.trainParam.show = df;
net.trainParam.lr = lr;
net.trainParam.epochs = me;
net.trainParam.goal = eg;
%net.performFcn = 'sse';

success=0;
for i=1:iterations

    net=init(net);
    [net1,tr]=train(net,P,T);    %Train the network
    [k l] = size(tr.perf);

    apodosi(i)=tr.perf(1);

    if apodosi(i) <= eg
        success = success+1;    %Give the number of succesful trainings
    end
end

end
```

```
a=sim(net1,P);  
accuracy1=accuracy(a,T); %Give the accuracy of the network
```

Και τέλος δίνεται ο κώδικας που υλοποιήθηκε για εκπαίδευση με τον αλγόριθμο Levenberg-Marquardt και τη μέθοδο εκπαίδευσης ανά πρότυπο εισόδου `trainr` όταν χρησιμοποιήθηκαν σύνολα αξιολόγησης και δοκιμής:

```
function [net1,net2,success1,success2,apodosi1,apodosi2,accuracy1,accuracy2]=  
nnt(P,T,hidden,tp)  
  
if nargin < 3  
    error('Not enough arguments.Arguments: P(input),T(target),  
    hidden(number of neurons in the hidden layer),  
    tp(vector of network parmeters)')  
  
elseif nargin == 3  
    tp=[];  
    df=50; %Default values for the parameters.  
    lr=0.01;  
    me=1000;  
    eg=0.001;  
    iterations=3;  
  
else  
    df = tp(1); % Display frequency  
    lr = tp(2); % Learning rate  
    me = tp(3); % Maximum epochs  
    eg = tp(4); % Error goal  
    iterations = tp(5); % Number of iterations  
end  
  
%Divide into validation set, training set and test set.  
[R,Q]=size(P);  
iitst = 2:4:Q;  
iival = 4:4:Q;  
iitr = [1:4:Q 3:4:Q];  
val.P = P(:,iival); val.T = T(:,iival);  
test.P = P(:,iitst); test.T = T(:,iitst);  
ptr = P(:,iitr); ttr = T(:,iitr);  
  
%Create network
```

```
net=newff(minmax(ptr), [hidden, length(ttr(:,1))],
{'tansig','tansig'},'trainlm');
neton=newff(minmax(ptr), [hidden, length(ttr(:,1))],
{'tansig','tansig'},'trainr');

net.trainParam.show = df;
net.trainParam.lr = lr;
net.trainParam.epochs = me;
net.trainParam.goal = eg;

neton.trainParam.show = df;
neton.trainParam.lr = lr;
neton.trainParam.epochs = me;
neton.trainParam.goal = eg;

success1=0;
success2=0;

for i=1:iterations

    net=init(net);

    [net1,tr]=train(net,ptr,ttr,[],[],val,test);

    [net2,tr2]=train(neton,ptr,ttr,[],[],val,test);
    figure(2)
    plot(tr.epoch,tr.perf,tr3.epoch,tr3.perf)
    legend('Training');
    ylabel('Squared error'); xlabel('Epoch')

    [k 1] = size(tr.perf);

    apodosi(i)=tr.perf(1);

    if apodosi(i) <= eg
        success1 = success1+1;
    end

[k2 12] = size(tr3.perf);
```

```
    apodosi2(i)=tr3.perf(l2);

    if apodosi2(i) <= eg
        success2 = success2+1;
    end

end

figure(3)
plot(tr.epoch,tr.perf,tr.epoch,tr.vperf,tr.epoch,tr.tperf)
legend('Training','Validation','Test',-1);
ylabel('Squared Error'); xlabel('Epoch')

figure(4)
plot(tr3.epoch,tr3.perf,tr3.epoch,tr3.vperf,tr3.epoch,tr3.tperf)
legend('Training','Validation','Test',-1);
ylabel('Squared Error'); xlabel('Epoch')

a1=sim(net1,P);
accuracy1=accuracy(a,T);
a2=sim(net2,P);
accuracy2=accuracy(a2,T);
```

Η ακρίβεια της εκπαίδευσης του δικτύου υπολογίστηκε σύμφωνα με τον παρακάτω κώδικα:

```
function accuracy=accuracy(a,T)
out=[];
correctpatterns=0;
[r t]=size(a);
for j=1:r
for l=1:t
if a(j,l)<0.4
out(j,l)=round(a(j,l));
elseif a(j,l)>0.6
out(j,l)=round(a(j,l));
else
out(j,l)=-1;
end
end
end
```

```
for l=1:t
if out(:,l) == T(:,l)
correctpatterns=correctpatterns+1;
end

end
out;
correctpatterns;
accuracy=(correctpatterns/length(T))*100;
```


Βιβλιογραφία

- [1] J.A.Andersen, “A Simple Neural Network generating an interactive memory”, Mathematical Biosciences, 1972.
- [2] Dave Anderson και George McNeill, “Artificial Neural Networks Technology”, ELIN: A011, Kaman Sciences Corporation, σελίδες:1-30, 1992.
- [3] Σάββας Ι. Ανδρουτσόπουλος, “Βραχυπρόθεσμη Πρόβλεψη Φορτίου με τη χρήση παραλλαγών του αλγορίθμου της Ανάστροφης Διάδοσης Σφάλματος των Τεχνητών Νευρωνικών Δικτύων”, Διπλωματική εργασία, Εθνικό Μετσόβιο Πολυτεχνείο, Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Τομέας Ηλεκτρικής Ισχύος , Αθήνα, 2003.
- [4] Oya Aran, Ethem Alpaydin, “An Incremental Neural Network Construction Algorithm for Training Multilayer Perceptrons”, Department of Computer Engineering, Bogazici University, Ινσταμβουλ, Τυρκεσ.
- [5] Πάνος Αργυράκης, “Νευρωνικά Δίκτυα και Εφαρμογές”, Ελληνικό Ανοικτό Πανεπιστήμιο, Θεματική ενότητα :Τεχνητή Νοημοσύνη, Πάτρα 2001.
- [6] E.Barnard, “Optimization for training neural nets”, IEEE Transaction on Neural Networks, 1992.
- [7] R.Battiri, “First and second order methods for learning:Between steepest descent and Newton’s method”, Neural Computation, 1992.
- [8] Cristopher M. Bishop, “Neural Networks for Pattern Recognition”, Department of Computer Science and Applied Mathematics, Aston University, Birmingham, UK, Clarendon Press, Oxford, 1995.
- [9] Βλαχάβας Ιωάννης, Κεφαλάς Πέτρος, Βασιλειάδης Νικόλαος, Κοκκορας Φώτης, Σακελλαρίου Ηλίας, ““Τεχνητή Νοημοσύνη””, Γ’ Έκδοση, 2006.
- [10] C.Charalambous, “Conjugate gradient algorithm for efficient training of artificial neural networks”, IEEE Proceedings, 1992.
- [11] Howard Demuth και Mark Beale, “Neural Network Toolbox for Use with Matlab”, The Mathworks, www.mathworks.com, version 4.0.4, 2004.

- [12] Κωνσταντίνος Διαμαντάρας, “S’untomo egqeir’idio Matlab”, Τμήμα Πληροφορικής, Αλεξάνδρειο Τ.Ε.Ι. Θεσσαλονίκης, 2007.
- [13] James A. Freeman, David M. Skapura, “Neural Networks, Algorithms, Applications and Programming Techniques”, Loral Space Information Systems and Adjunct Faculty, School of Natural and Applied Sciences, University of Houston at Clear Lake, Addison-Wesley Publishing Company, 1991.
- [14] S.Grossberg, “Some Physiological and Biomedical consequences of psychological postulates”, Proceedings of the National Academy of Sciences, 1968.
- [15] Martin T. Hagan, Howard B. Domuth, Mark Beale, “Neural Network Design”, 2000.
- [16] Jeff Heaton, “Programming Neural Networks in Java” , 2005.
- [17] D.O.Hebb, “The Organization of Behaviour”, New York, Willey, 1949.
- [18] Tom Heskes, Bert Kappen, “On-line learning processes in artificial neural networks”, Department of Medical Physics and Biophysics, University of Nijmegen.
- [19] Tom Heskes, Wim Wiegerinck, “A Theoretical Comparison of bath-mode, on-line, cyclic and almost cyclic learning”, RWCP Novel Functions, SNN Laboratory, Department of Medical Physics and Biophysics, University of Nijmegen.
- [20] J.J.Hopfield, “Neyral Networks and physical systems with emergent collective computational properties”, Proceedings of the National Academy of Sciences, 1982.
- [21] T.Kohonen, “Self-Organization and Associative Memory”, Berlin:Springer-Verlag, 2nd Edition, 1987.
- [22] Heikki N. Koivo, “Neural Networks: Basics using MATLAB,Neural Network Toolbox”, The University of Amsterdam, 1996.
- [23] Ben Krose και Patrick van der Smagt, “An Introduction to Neural Networks”, The University of Amsterdam, 1996.
- [24] Λαγαρός Δ. Νικόλαος, “Εφαρμογές των νευρωνικών δικτύων στην στατική ανάλυση και τον σχεδιασμό των κατασκευών”, Εθνικ’ο Mets’obio Poluteqne’io, Diplwmatik’h ergas’ia, 1994.
- [25] Tomas Natschlager, “Optical Character Recognition”, Institute of Theoretical Computer Science, Inffeldgass.
- [26] W.McCulloch και Walter Pitts, “A logical calculus of the ideas immament in nervous activity”, Bulletin of Mathematical Biophysics, 1943.
- [27] M.Minsky, S.Papert, “Perceptrons”, Cambridge, MIT Press, 1969.

- [28] Θεωδωρής Πιτικάρης, Ιωάννης Τσαγκατάκης, Μιχάλης Νικητάκης, Γιώργος Παπαδουράκης, “Digital libraries: information retrieval with neural network”, 11ο Πα-νελλήνιο Συνέδριο Ακαδημαϊκών Βιβλιοθηκών, Α.Τ.Ε.Ι ΚΡΗΤΗΣ, 2002.
- [29] Ράμμος Γεώργιος, ““Anatom’ia kai Fusiolog’ia, Neurik’o S’usthma””, Σημειώσεις, Τμήμα Πληροφορικής με Εφαρμογές στη Βιοϊατρική, Λαμία, 2006.
- [30] Valluru B. Rao, “C++ Neural Networks and Fuzzy Logic”, M&T Books, 1995.
- [31] F.Rosenblatt, “The Perceptron: A probabilistic model for information storage and organization in the brain”, Psychological Review, 1958.
- [32] D.E.Rumelhart και J.L.McClelland, “Parallel Distributed Processing:Explorations in the Microstructure of Cognition”, Cambridge, MIT Press, 1986.
- [33] D.F.Shanno, “Recent advances in numerical techniques for large-scale optimization”, Neural Networks for Control, Miller, Sutton and Werbos, Cambridge, MIT Press, 1990.
- [34] Σταματιάδη Θεοδώρα, “Εφαρμογή των Νευρωνικών Δικτύων Στις Τηλεπικοινωνίες Αναγνώριση Σήματος”, Διπλωματική εργασία, Εθνικό Μετσόβιο Πολυτεχνείο, Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Τομέας Ηλεκτροεπιστήμης, Αθήνα, 2003.
- [35] Hooman Yousefizadeh και Ali Zilouchian, “Neural Network Architectures”, CRC Press LLC, 2001.
- [36] L.P.J Veelenturf, “Analysis and Applications of Artificial Neural Networks”, Pentice Hall, σελίδες:1-96, 1995.
- [37] Thomas Weise, “Global Optimization Algorithms-Theory and Application”, <http://www.it-weise.de/>, 2008.
- [38] Jason Westmacott, “An Artificial Neural Network for Robust Shape Recognition in Real Time”, Thesis, School of Electical and Information Engineering, University of South Australia, 2000.
- [39] B.Widrow, M.E.Hoff, “Adaptive switching circuits”, 1960.