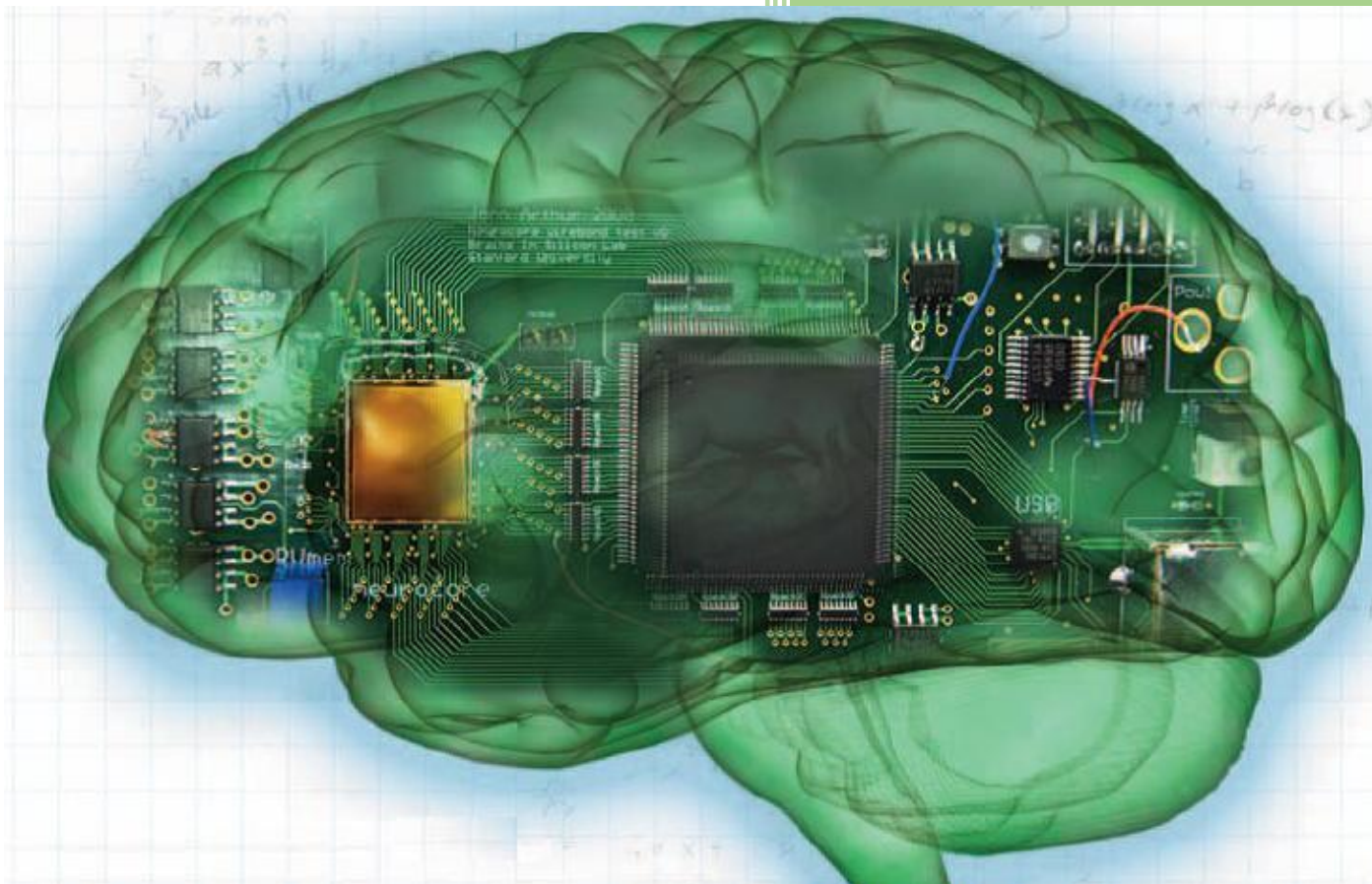


Πολυτεχνείο Κρήτης

Σχολή Μηχανικών Παραγωγής και Διοίκησης

Σεπτέμβριος 2013

Επεξεργασία μηχανικών ιδιοτήτων χαρακτηριστικού κελύφους περιοδικότητας με τη χρήση νευρωνικών δικτύων.



ΓΙΑΚΟΥΜΗΣ ΓΕΩΡΓΙΟΣ

A.M : 2007010029

Επιβλέπων : Γεώργιος Σταυρουλάκης

ΕΥΧΑΡΙΣΤΙΕΣ

Καταρχήν, οφείλω να ευχαριστήσω τον καθηγητή μου κ. Γεώργιο Ε. Σταυρουλάκη για την ανάθεση του θέματος καθώς και για τις πολύτιμες συμβουλές του. Θα ήθελα επίσης να ευχαριστήσω κ. Δρ. Γεώργιο Δροσόπουλο και τον υποψήφιο Διδάκτορα κ. Γεώργιο Ταϊρίδη για την επίβλεψη της διπλωματικής εργασίας καθώς και τον συμφοιτητή μου κ. Κωνσταντίνο Γιάννη που διέθεσε μέρος από τα αποτελέσματα της διπλωματικής του διατριβής.

Περιεχόμενα

I. Κατάλογος Εικόνων	4
II. Κατάλογος Πινάκων	5
Πρόλογος.....	6
Εισαγωγή.....	7
ΚΕΦΑΛΑΙΟ 1	8
Ιστορική αναδρομή.....	8
ΚΕΦΑΛΑΙΟ 2	9
I. Νευρωνικά Δίκτυα	9
II. Μοντέλα νευρώνων	10
III. Συναρτήσεις μεταφοράς.....	12
IV. Αρχιτεκτονικές δικτύων	14
V. Μάθηση των νευρωνικών δικτύων	16
ΚΕΦΑΛΑΙΟ 3	18
I. Μέθοδος επίλυσης του προβλήματος.....	18
II. Χρήση COMSOL για την εύρεση τάσεων, για αυξανόμενο επίπεδο τροπής.....	18
III. Χρήση COMSOL για την εύρεση επαπτομενικού μητρώου ακαμψίας, ανά επίπεδο τροπής.....	20
IV. Δεδομένα εισόδου (input).....	25
V. Δεδομένα εξόδου (output)	25
VI. Δεδομένα ελέγχου	25
VII. Εκπαίδευση δικτύου	26
1 ^{ος} Όγκος Δεδομένων (Χρησιμοποιώντας τους πίνακες strain [50 x 3] & stress [50 x 3])	26
2 ^{ος} Όγκος Δεδομένων (Χρησιμοποιώντας τους πίνακες strain[50x3] & consistent [150x3])	40
ΚΕΦΑΛΑΙΟ 4	53
Συμπεράσματα.....	53
ΚΕΦΑΛΑΙΟ 5	54
Βιβλιογραφία	54

I. Κατάλογος Εικόνων

- I. Εικόνα 2.1 : Προσαρμογή – Εκπαίδευση ΝΔ.
- II. Εικόνα 2.2 : Μοντέλο νευρώνα με βαθμωτή είσοδο, χωρίς bias.
- III. Εικόνα 2.3 : Μοντέλο νευρώνα με βαθμωτή είσοδο και bias.
- IV. Εικόνα 2.4 : Μοντέλο νευρώνα με είσοδο διάνυσμα.
- V. Εικόνα 2.5 : Βηματική συνάρτηση μεταφοράς.
- VI. Εικόνα 2.6 : Γραμμική συνάρτηση μεταφοράς.
- VII. Εικόνα 2.7 : Εφαπτομενική – σιγμοειδής συνάρτηση μεταφοράς.
- VIII. Εικόνα 2.8 : Λογαριθμική – σιγμοειδής συνάρτηση μεταφοράς.
- IX. Εικόνα 2.9 : Επίπεδο νευρώνων.
- X. Εικόνα 2.10 : Μοντέλο ενός επιπέδου πολυεπίπεδου ΝΔ.
- XI. Εικόνα 2.11 : Πολυεπίπεδο ΝΔ.
- XII. Εικόνα 3.12 : Απεικόνιση της αρχιτεκτονικής του δικτύου 3-20-3
- XIII. Εικόνα 3.13 : Απόκλιση μεταξύ πραγματικής και εκτιμώμενης τιμής εξόδου με τυχαία διαίρεση δεδομένων και δεδομένα εκπαίδευσης τις καταστάσεις αρχική, γωνία 30 και γωνία 90 μοίρες.
- XIV. Εικόνα 3.14 : Απόκλιση μεταξύ πραγματικής και εκτιμώμενης τιμής εξόδου με μη τυχαία διαίρεση των δεδομένων.
- XV. Εικόνα 3.15 : Απόκλιση μεταξύ πραγματικής και εκτιμώμενης τιμής εξόδου για τα συνολικά δεδομένα με τυχαία διαίρεση των δεδομένων.
- XVI. Εικόνα 3.16 : Απόκλιση μεταξύ πραγματικής και εκτιμώμενης τιμής εξόδου για τα δεδομένα που είναι για εκπαίδευση.
- XVII. Εικόνα 3.17 : Αρχιτεκτονική του δικτύου με δύο κρυφά επίπεδα 3-20-20-3
- XVIII. Εικόνα 3.18 : Απόκλιση μεταξύ πραγματικής και εκτιμώμενης τιμής εξόδου με τυχαία διαίρεση δεδομένων με δύο κρυφά επίπεδα.
- XIX. Εικόνα 3.19 : Απόκλιση μεταξύ πραγματικής και εκτιμώμενης τιμής εξόδου με τυχαία διαίρεση δεδομένων με δύο κρυφά επίπεδα.
- XX. Εικόνα 3.20 : Αρχιτεκτονική του δικτύου με δύο κρυφά επίπεδα 3-20-20-9
- XXI. Εικόνα 3.21 : Απόκλιση μεταξύ πραγματικής και εκτιμώμενης τιμής εξόδου για το σύνολο των δεδομένων με τυχαία διαίρεση δεδομένων και δύο κρυφά επίπεδα.
- XXII. Εικόνα 3.22 : Αρχιτεκτονική του δικτύου με δύο κρυφά επίπεδα 3-20-20-3
- XXIII. Εικόνα 3.23 : Αρχιτεκτονική του δικτύου με δύο κρυφά επίπεδα 3-20-20-1
- XXIV. Εικόνα 3.24: Απόκλιση μεταξύ πραγματικής και εκτιμώμενης τιμής εξόδου για το σύνολο των δεδομένων με τυχαία διαίρεση δεδομένων και δύο κρυφά επίπεδα.

II. Κατάλογος Πινάκων

- I. Πίνακας 1 : ΑΡΧΕΙΟ COMSOL 1 - ΚΕΝΤΡΙΚΗ ΕΠΙΛΥΣΗ ΑΝΑΦΟΡΑΣ
- II. Πίνακας 2 : Αυξανόμενα επίπεδα τροπής στο ΑΡΧΕΙΟ COMSOL 1
- III. Πίνακας 3 : Αποτελέσματα για το ΑΡΧΕΙΟ COMSOL 1 – Τάσεις
- IV. Πίνακας 4 : ΑΡΧΕΙΟ COMSOL 2
- V. Πίνακας 5 : Αυξανόμενα επίπεδα τροπής στο ΑΡΧΕΙΟ COMSOL 2
- VI. Πίνακας 6 : Αποτελέσματα για το ΑΡΧΕΙΟ COMSOL 2 – Τάσεις
- VII. Πίνακας 7 : Increments Τροπών για $\delta\epsilon_{xx1}$, $\delta\epsilon_{yy1}$, $\delta\epsilon_{xy1}$
- VIII. Πίνακας 8 : ΑΡΧΕΙΟ COMSOL 3
- IX. Πίνακας 9 : Αυξανόμενα επίπεδα τροπής στο ΑΡΧΕΙΟ COMSOL 3
- X. Πίνακας 10 : Αποτελέσματα για το ΑΡΧΕΙΟ COMSOL 3 – Τάσεις
- XI. Πίνακας 11 : Increments Τροπών για $\delta\epsilon_{xx2}$, $\delta\epsilon_{yy2}$, $\delta\epsilon_{xy2}$
- XII. Πίνακας 12 : ΑΡΧΕΙΟ COMSOL 4
- XIII. Πίνακας 13 : Αυξανόμενα επίπεδα τροπής στο ΑΡΧΕΙΟ COMSOL 4
- XIV. Πίνακας 14 : Αποτελέσματα για το ΑΡΧΕΙΟ COMSOL 4 – Τάσεις
- XV. Πίνακας 15: Increments Τροπών για $\delta\epsilon_{xx3}$, $\delta\epsilon_{yy3}$, $\delta\epsilon_{xy3}$
- XVI. Πίνακας 16 : Συνοπτικός πίνακας αποτελεσμάτων των 3 πολιτικών που ακολουθήθηκαν
- XVII. Πίνακας 17 : Συνοπτικός πίνακας αποτελεσμάτων των πολιτικών που ακολουθήθηκαν έχοντας δύο κρυφά επίπεδα.

Πρόλογος

Η μελέτη μιας μεγάλης κατασκευής είναι μια πολύ δύσκολη και απαιτητική διαδικασία αλλά ταυτόχρονα απαραίτητη για την σωστή χρήση, λειτουργία και μελλοντική εξέλιξη της. Κατασκευές όπως μια οικοδομή ή ένα επιβατικό αεροσκάφος είναι μεγάλες σε όγκο, συνεπώς η μελέτη τους σε όλη την έκταση τους είναι δύσκολα πραγματοποιήσιμη. Για το λόγο αυτό την διαιρούμε και παίρνουμε έναν μικρό όγκο ελέγχου - ένα μικρό δοκίμιο, εφόσον παρουσιάζει περιοδικότητα. Ειδικότερα, εξετάζουμε ένα δισδιάστατο τετράγωνο αναφοράς, γραμμικό ελαστικό υλικό, το οποίο θα δέχεται αυξανόμενα επίπεδα τροπής σε 3 σημεία (ϵ_{xx} , ϵ_{yy} , ϵ_{xy}) και λαμβάνουμε τις αντίστοιχες τάσεις (σ_{xx} , σ_{yy} , σ_{xy}). Τα δεδομένα που θα χειριστούμε παράγονται από μη γραμμικές επιλύσεις πεπερασμένων στοιχείων, έχοντας σαν δοκίμιο ένα δισδιάστατο τετράγωνο με μικροδομή, του οποίου η συμπεριφορά είναι μη γραμμική, π.χ. ελαστοπλαστικότητα.

Η παρούσα εργασία βασίζεται στην μελέτη των Σταυρουλάκη Γεώργιο και Δροσόπουλο από την οποία προέρχονται τα δεδομένα των προσομοιώσεων που χρησιμοποιήθηκαν εδώ για την πρόβλεψη των τάσεων παραμόρφωσης. Για την πρόβλεψη των τάσεων παραμόρφωσης της πλάκας χρησιμοποιήθηκε η μέθοδος των νευρωνικών δικτύων. Το παρόν δίκτυο δημιουργήθηκε σε περιβάλλον Matlab το οποίο εκπαιδεύτηκε χρησιμοποιώντας τα αποτελέσματα των προσομοιώσεων και έχει ως έξοδο τις τάσεις παραμόρφωσης. Στη συνέχεια το εκπαιδευμένο δίκτυο θα χρησιμοποιηθεί αντί του μοντέλου πεπερασμένων στοιχείων για την εκτίμηση του αποτελέσματος (τάσεις) για κάθε δυνατή φόρτιση (παραμόρφωση). Επιπλέον σε κάθε θέση μπορούμε να χρησιμοποιήσουμε το νευρωνικό δίκτυο σε συνδυασμό με πεπερασμένες διαφορές για την εκτίμηση της πρώτης παραγώγου της απόκρισης (ευαισθησίας λύσεως, εφαπτομενική δυσκαμψία).

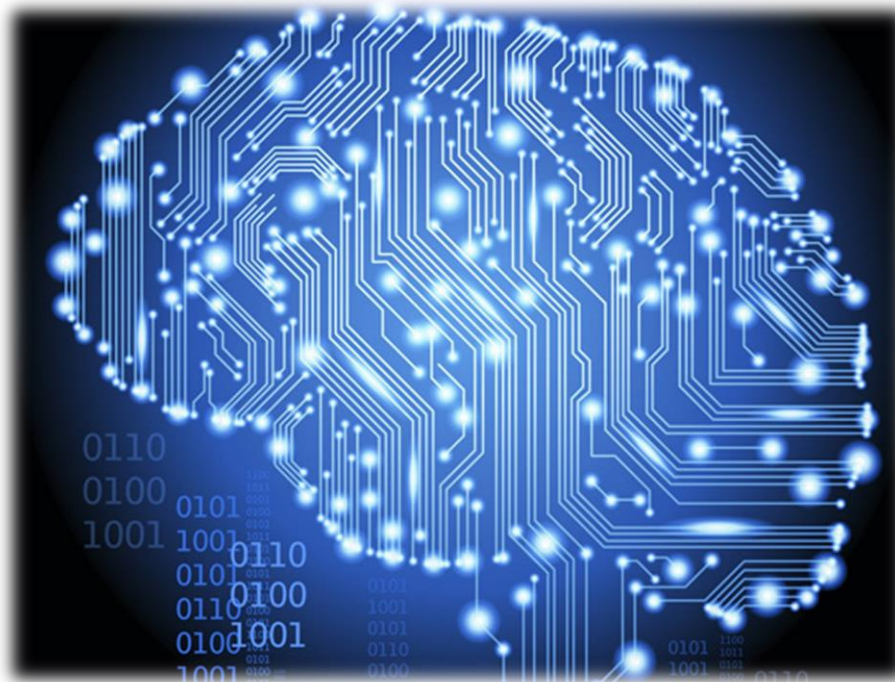
Η διάρθρωση της παρούσης διπλωματικής εργασίας έχει την εξής δομή. Αρχικά, γίνεται αναφορά στην έννοια και στην χρήση του Τεχνητού Νευρωνικού Δικτυού (ΤΝΔ, Artificial Neural Network (ANN)). Στο Κεφάλαιο 1 παρουσιάζεται εν συντομία η ιστορική αναδρομή των ΝΔ. Στο Κεφάλαιο 2 παρατίθεται η αναγκαία για την κατανόηση των νευρωνικών δικτύων θεωρία. Στο Κεφάλαιο 3 αναλύεται συστηματικά η διαδικασία που ακολουθήθηκε για την εκπαίδευση και βελτιστοποίηση των αποτελεσμάτων. Στο Κεφάλαιο 4 εκτίθενται τα συμπεράσματα και τέλος στο Κεφάλαιο 5 η εργασία ολοκληρώνεται με την βιβλιογραφία που χρησιμοποιήθηκε.

Εισαγωγή

Το Τεχνητό Νευρωνικό Δίκτυο (ΤΝΔ, Artificial Neural Network (ANN)), το οποίο συχνά καλείται Προσομοιωμένο Νευρωνικό Δίκτυο ή απλά Νευρωνικό δίκτυο (ΝΔ), είναι ένα μαθηματικό ή υπολογιστικό μοντέλο βασισμένο στα βιολογικά νευρωνικά δίκτυα. Αποτελείται από διασυνδεδεμένες ομάδες τεχνητών νευρώνων και επεξεργάζεται τις πληροφορίες χρησιμοποιώντας την συνδεδετική προσέγγιση (connectionist approach) στους υπολογισμούς. Στις περισσότερες περιπτώσεις ένα Νευρωνικό Δίκτυο αποτελεί ένα προσαρμόσιμο/δυναμικό σύστημα, το οποίο αλλάζει τη δομή του είτε λόγω εξωτερικών είτε λόγω εσωτερικών πληροφοριών που προκύπτουν κατά τη φάση της εκπαίδευσης.

Πρακτικά, το Νευρωνικό Δίκτυο είναι ένα χρήσιμο εργαλείο όχι μόνο για την επεξεργασία πληροφοριών αλλά και για πολλές άλλες εφαρμογές. Λόγω των μοναδικών τους ιδιοτήτων, τα νευρωνικά δίκτυα μπορούν να χρησιμοποιηθούν για την επίλυση σύνθετων προβλημάτων τα οποία είτε δεν μπορούν να επιλυθούν με αναλυτικές μεθόδους είτε το φυσικό και μαθηματικό τους μοντέλο δεν είναι γνωστό. Τα ΝΔ έχουν χρησιμοποιηθεί επίσης για την μοντελοποίηση των μη γραμμικών και σύνθετων σχέσεων μεταξύ των κατασκευαστικών παραμέτρων και των δυναμικών χαρακτηριστικών του υλικού. Γίνεται λοιπόν εύκολα αντιληπτό ότι τα ΝΔ είναι πολύ χρήσιμα για την επίλυση αντίστροφων προβλημάτων που σχετίζονται με τον μη καταστροφικό έλεγχο υλικών και κατασκευαστικών συστημάτων.

Ως αποτέλεσμα, τα ΝΔ αποτελούν ένα τρόπο μοντελοποίησης μη γραμμικών στατιστικών δεδομένων, ενώ αποτελούν ένα σημαντικό εργαλείο για τη μοντελοποίηση περίπλοκων σχέσεων μεταξύ των δεδομένων εισόδου – εξόδου.



ΚΕΦΑΛΑΙΟ 1

Ιστορική αναδρομή

Το ανθρώπινο μυαλό αποτελεί αντικείμενο έρευνας εδώ και χιλιάδες χρόνια. Με την ανάπτυξη όμως των σύγχρονων ηλεκτρονικών ο άνθρωπος προσπάθησε να μιμηθεί τον ανθρώπινο εγκέφαλο και τις νοητικές του διεργασίες. Το πρώτο βήμα για την ανάπτυξη των Νευρωνικών Δικτύων έγινε το 1943 από τον νεύρο-φυσιολόγο Warren McCulloch και τον μαθηματικό Walter Pitts. Ο McCulloch αφιέρωσε περισσότερα από 20 χρόνια έρευνας στην μελέτη του ανθρώπινου εγκεφάλου και νευρικού συστήματος. Όταν λοιπόν συνεργάστηκε με το νεαρό Pitts έκαναν μια δημοσίευση στην οποία αρχικά παρουσίασαν ένα πιθανό σενάριο για τη λειτουργία των νευρώνων και εν συνεχεία ένα πρωταρχικό Νευρωνικό Δίκτυο (ΝΔ), το οποίο είχαν κατασκευάσει χρησιμοποιώντας απλά ηλεκτρικά κυκλώματα. Το επόμενο μεγάλο επίτευγμα στην τεχνολογία των νευρωνικών δικτύων δεν άργησε. Μόλις, το 1949 εκδόθηκε το βιβλίο “Η Οργάνωση της Συμπεριφοράς” (“The Organization of Behavior”), του Donald Hebb. Το βιβλίο υποστήριξε και ενίσχυσε την θεωρία των McCulloch – Pitt σχετικά με τους νευρώνες και την λειτουργία αυτών. Αλλά το κύριο θέμα που έθιξε ήταν ότι οι νευρωνικοί «δρόμοι» αποκτούν δύναμη κάθε φορά που χρησιμοποιούνται. Αυτό αληθεύει και αποτελεί το κλειδί στην εκπαίδευση του δικτύου. Κατά τη διάρκεια της δεκαετίας του 1950 η έρευνα των νευρωνικών δικτύων ουσιαστικά περνά στην αφάνεια αφού εκείνη την περίοδο οι Η/Υ αποτέλεσαν το κύριο ενδιαφέρον. Παρόλα αυτά, το 1954 ο Martin Minsky ολοκληρώνει την διδακτορική του διατριβή με τίτλο «Θεωρία Νεύρο-Αναλογικής Ενίσχυσης Συστημάτων και οι Εφαρμογές τους στο Πρόβλημα του Εγκεφαλικού Μοντέλου» («Theory of Neural- Analog Reinforcement System and its Application to the Brain-Model Problem»). Από τον ίδιο ακολούθησε και μια δημοσίευση με τίτλο «Βήματα προς την Τεχνητή Νοημοσύνη» («Steps Towards Artificial Intelligence»). Στην συγκεκριμένη δημοσίευση γίνεται πρώτη φορά αναλυτική αναφορά στην τεχνητή νοημοσύνη και στα νευρωνικά δίκτυα όπως αυτά είναι γνωστά σήμερα.

Ως «επίσημη» χρονολογία έναρξης της έρευνας των νευρωνικών δικτύων όμως, θεωρείται το 1959 όταν το Dartmouth Summer Research Project on Artificial Intelligence ξεκίνησε επίσημα την έρευνα στο πεδίο της τεχνητής νοημοσύνης. Αρκετά χρόνια αργότερα, ο John von Neumann προσπάθησε να μιμηθεί τις υπεραπλουστευμένες λειτουργίες των νευρώνων χρησιμοποιώντας αναμεταδόσεις του τηλεγράφου ή σωλήνες κενού. Αυτή η προσπάθεια οδήγησε στην εφεύρεση της μηχανής του von Neumann. Δεκαπέντε χρόνια μετά την πρωτοποριακή δημοσίευση των McCulloch και Pitts, παρουσιάζεται μια προσέγγιση της περιοχής των νευρωνικών δικτύων. Συγκεκριμένα το 1958, ο Frank Rosenblatt, ένας νεύρο-βιολόγος στο Cornell University, ξεκίνησε να ερευνά στο Perceptron. Το perceptron είναι ουσιαστικά το πρώτο τεχνητό νευρωνικό δίκτυο. Κατασκευάστηκε χρησιμοποιώντας τον κάπως πρωτόγονο εξοπλισμό εκείνης της εποχής. Το perceptron βασίζεται στην έρευνα που έγινε πάνω στο μάτι της μύγας. Η διεργασία η οποία «λέει» στη μύγα να πετάξει όταν υπάρχει κίνδυνος εκτελείται στο μάτι. Το μειονέκτημα του perceptron είναι ότι έχει περιορισμένες δυνατότητες.

ΚΕΦΑΛΑΙΟ 2

I. Νευρωνικά Δίκτυα

Τι είναι το Νευρωνικό Δίκτυο ;

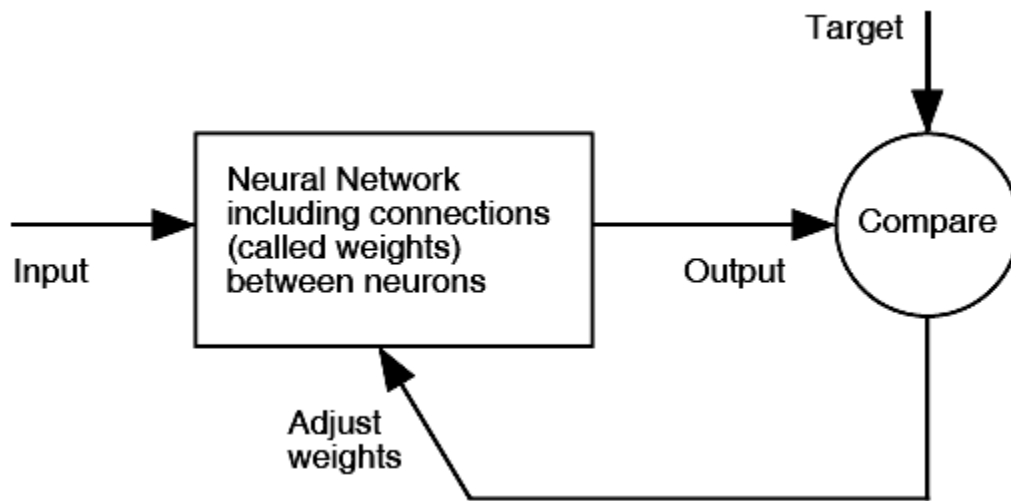
Ορισμός : «Παρόλο που δεν υπάρχει κάποιος καθολικά αποδεκτός ορισμός όσον αφορά τι είναι ένα ΝΔ, οι περισσότεροι θα συμφωνούσαν ότι περιλαμβάνει ένα δίκτυο από απλά επεξεργαστικά στοιχεία, τους νευρώνες (*neurons*), οι οποίοι έχουν την δυνατότητα να επιδείξουν αρκετά σύνθετη καθολική συμπεριφορά. Αυτή καθορίζεται αποκλειστικά από τις συνδέσεις μεταξύ των νευρώνων και από τις παραμέτρους που τους χαρακτηρίζουν.»

Το Τεχνητό Νευρωνικό Δίκτυο (ΤΝΔ, Artificial Neural Network (ANN)), είναι ένα δίκτυο από απλούς υπολογιστικούς κόμβους (νευρώνες), διασυνδεδεμένους μεταξύ τους. Είναι εμπνευσμένο από το Κεντρικό Νευρικό Σύστημα, το οποίο προσπαθεί να προσομοιώσει. Οι νευρώνες είναι τα δομικά στοιχεία του δικτύου. Κάθε τέτοιος κόμβος δέχεται ένα σύνολο αριθμητικών εισόδων από διαφορετικές πηγές (είτε από άλλους νευρώνες, είτε από το περιβάλλον), επιτελεί έναν υπολογισμό με βάση αυτές τις εισόδους και παράγει μία έξοδο. Η εν λόγω έξοδος είτε κατευθύνεται στο περιβάλλον, είτε τροφοδοτείται ως είσοδος σε άλλους νευρώνες του δικτύου. Υπάρχουν τρεις τύποι νευρώνων: οι νευρώνες εισόδου, οι νευρώνες εξόδου και οι υπολογιστικοί νευρώνες ή κρυμμένοι νευρώνες. Οι νευρώνες εισόδου, μεσολαβούν ανάμεσα στις περιβαλλοντικές εισόδους του δικτύου και στους υπολογιστικούς νευρώνες. Οι νευρώνες εξόδου διοχετεύουν στο περιβάλλον τις τελικές αριθμητικές εξόδους του δικτύου. Οι υπολογιστικοί νευρώνες πολλαπλασιάζουν κάθε είσοδό τους με το αντίστοιχο συναπτικό βάρος και υπολογίζουν το ολικό άθροισμα των γινομένων. Το άθροισμα αυτό τροφοδοτείται ως όρισμα στη συνάρτηση ενεργοποίησης, την οποία υλοποιεί εσωτερικά κάθε κόμβος. Η τιμή που λαμβάνει η συνάρτηση για το εν λόγω όρισμα είναι και η έξοδος του νευρώνα για τις τρέχουσες εισόδους και βάρη. Εάν x_{ki} είναι η i -οστή είσοδος του k νευρώνα, w_{ki} : το i -οστό συναπτικό βάρος του k νευρώνα και $\phi(\cdot)$ η συνάρτηση ενεργοποίησης του νευρωνικού δικτύου, τότε η έξοδος y_k του k νευρώνα δίνεται από την εξίσωση:

$$y_k = \phi \left(\sum_{i=0}^N x_{ki} w_{ki} \right)$$

Στον k -οστό νευρώνα υπάρχει ένα συναπτικό βάρος w_{k0} με ιδιαίτερη σημασία, το οποίο καλείται **πόλωση** ή **κατώφλι** (bias, threshold). Η τιμή της εισόδου του είναι πάντα η μονάδα, $x_{k0} = 1$. Εάν το συνολικό άθροισμα από τις υπόλοιπες εισόδους του νευρώνα είναι μεγαλύτερο από την τιμή αυτή, τότε ο νευρώνας ενεργοποιείται. Εάν είναι μικρότερο, τότε ο νευρώνας παραμένει ανενεργός. Η ιδέα προέκυψε από τα βιολογικά νευρικά κύτταρα. Το κύριο χαρακτηριστικό των νευρωνικών δικτύων είναι η εγγενής ικανότητα μάθησης. Ως μάθηση μπορεί να οριστεί η σταδιακή βελτίωση της ικανότητας του δικτύου να επιλύει κάποιο πρόβλημα (π.χ. η σταδιακή προσέγγιση μίας συνάρτησης). Η μάθηση επιτυγχάνεται μέσω της εκπαίδευσης, μίας επαναληπτικής διαδικασίας σταδιακής προσαρμογής των παραμέτρων του δικτύου (συνήθως των βαρών και της πόλωσης του) σε τιμές κατάλληλες ώστε να

επιλύεται με επαρκή επιτυχία το προς εξέταση πρόβλημα. Παρακάτω παρουσιάζεται μια τέτοια κατάσταση σχηματικά (Εικόνα 2.1). Συγκεκριμένα, το δίκτυο προσαρμόζεται, βάση της σύγκρισης που γίνεται μεταξύ των εξόδων του ΝΔ και των αντίστοιχων επιθυμητών στόχων.

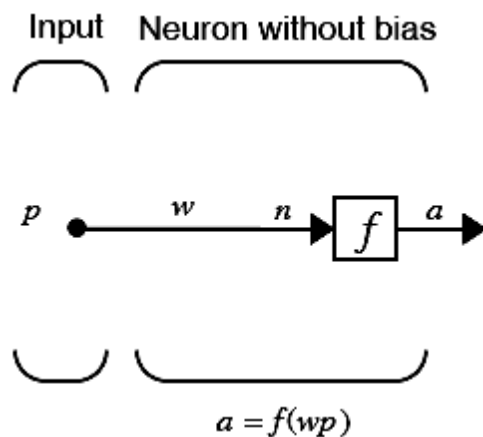


Εικόνα 2.1: Προσαρμογή – Εκπαίδευση ΝΔ.

Αυτή η διαδικασία συνεχίζεται έως ότου αυτές οι ποσότητες ταυτιστούν, δηλαδή μηδενιστεί το σφάλμα (πρακτικά η διαδικασία σταματά όταν το σφάλμα λάβει μια αποδεκτή τιμή). Το ζητούμενο είναι το λειτουργικό δίκτυο να χαρακτηρίζεται από μία ικανότητα **γενίκευσης**: αυτό σημαίνει πως δίνει ορθές εξόδους για εισόδους καινοφανείς και διαφορετικές από αυτές με τις οποίες εκπαιδεύτηκε. Στο σημείο αυτό πρέπει να σημειωθεί ότι για την εκπαίδευση του ΝΔ απαιτούνται πολλά τέτοια ζευγάρια εισόδων/εξόδων.

II. Μοντέλα νευρώνων

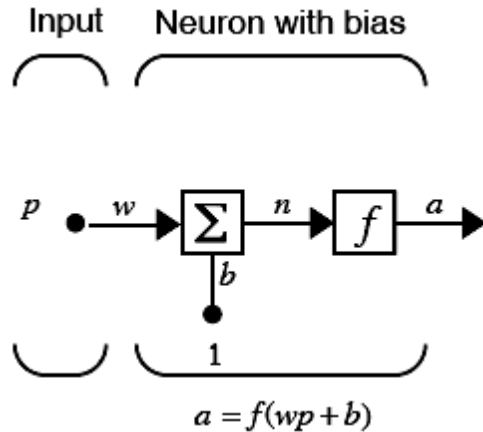
Απλός νευρώνας (χωρίς bias) : Παρακάτω παρουσιάζεται σχηματικά ένας νευρώνας με βαθμωτή είσοδο και χωρίς bias.



Εικόνα 2.2 : Μοντέλο νευρώνα με βαθμωτή είσοδο, χωρίς bias.

Η βαθμωτή είσοδος p μεταδίδεται μέσω μιας σύνδεσης, η οποία πολλαπλασιάζει το μέγεθος της με το συντελεστή βαρύτητας w ώστε να σχηματιστεί το wp , το οποίο με τη σειρά του είναι και αυτό βαθμωτό μέγεθος. Η ποσότητα wp αποτελεί το όρισμα της συνάρτησης μεταφοράς f , η οποία τελικά παράγει τη βαθμωτή έξοδο a .

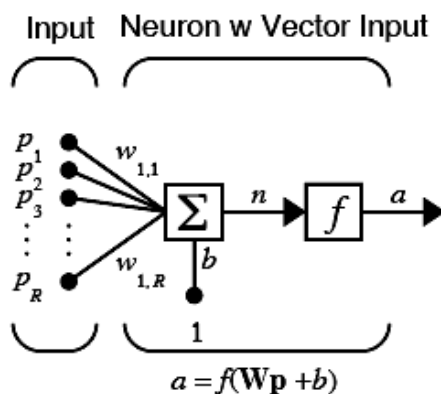
Απλός νευρώνας (με bias) : Ένας νευρώνας με βαθμωτή είσοδο και bias έχει την παρακάτω μορφή.



Εικόνα 2.3 : Μοντέλο νευρώνα με βαθμωτή είσοδο και bias.

Σε αυτήν την περίπτωση το όρισμα της συνάρτησης μεταφοράς f θα είναι το καθαρό μέγεθος n , όπου $n=wp+b$. Το bias μπορεί να θεωρηθεί ένας συντελεστής βαρύτητας, με τη διαφορά ότι έχει σταθερή είσοδο τη μονάδα. Στο σημείο αυτό πρέπει να επισημανθεί το γεγονός ότι τα μεγέθη w και b αποτελούν προσαρμόσιμες παραμέτρους του νευρώνα. Η κύρια ιδέα των νευρωνικών δικτύων είναι ότι τέτοιες παράμετροι μπορούν να μεταβληθούν ώστε τελικά το ΝΔ να έχει την επιθυμητή συμπεριφορά. Η εύρεση αυτών των τιμών, που θα δώσουν τα επιθυμητά αποτελέσματα είναι ο σκοπός της εκπαίδευσης του δικτύου. Όπως αναφέρθηκε και προηγουμένως, το bias « b » είναι μια προσαρμόσιμη παράμετρος του νευρώνα. Δεν αποτελεί κάποια είσοδο. Παρόλα αυτά η σταθερά 1 η οποία κατευθύνει το bias είναι είσοδος και πρέπει να αντιμετωπίζεται ως είσοδος όταν αναφερόμαστε στη γραμμική εξάρτηση των διανυσμάτων εισόδου.

Νευρώνας με είσοδο διάνυσμα : Ένας νευρώνας με είσοδο ένα διάνυσμα p με R στοιχεία έχει την παρακάτω μορφή.



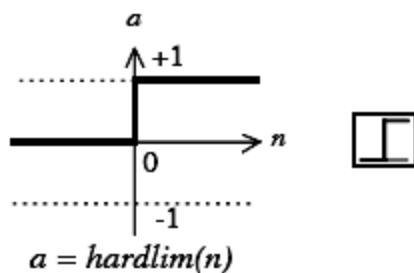
Εικόνα 2.4 : Μοντέλο νευρώνα με είσοδο διάνυσμα.

Σε αυτήν την περίπτωση τα μεγέθη p_1, p_2, \dots, p_R πολλαπλασιάζονται με τους συντελεστές βαρύτητας $w_{1,1}, w_{1,2}, \dots, w_{1,R}$ και το σταθμισμένο αποτέλεσμα οδηγείται στην αθροιστική διασταύρωση. Το άθροισμα είναι απλά Wp , το δεκαδικό μέρος μιας γραμμής του πίνακα W και του διανύσματος p . Ο νευρώνας έχει ένα bias b , το οποίο αθροίζεται με τις σταθμισμένες εισόδους για να σχηματίσει την καθαρή είσοδο n . Το άθροισμα n αποτελεί το όρισμα της συνάρτησης μεταφοράς f .

III. Συναρτήσεις μεταφοράς

Παρακάτω αναλύονται οι τέσσερις πιο συχνά χρησιμοποιούμενες συναρτήσεις μεταφοράς. Υπάρχουν βέβαια αρκετές ακόμα, οι οποίες χρησιμοποιούνται σε ιδιάζουσες περιπτώσεις.

Βηματική συνάρτηση μεταφοράς (Hard-limit)

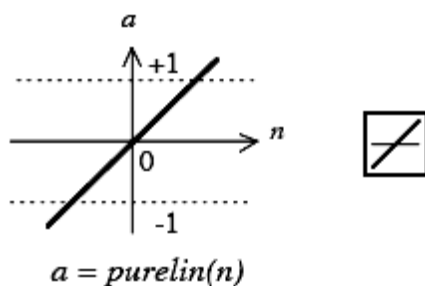


Hard-Limit Transfer Function

Εικόνα 2.5 : Βηματική συνάρτηση μεταφοράς.

Η συνάρτηση θέτει ως έξοδο του νευρώνα είτε το 0, εάν το καθαρό (net) όρισμα εισόδου είναι μικρότερο από 0, είτε το 1, σε οποιαδήποτε άλλη περίπτωση. Συχνά, χρησιμοποιείται στα perceptrons ώστε να δημιουργηθούν νευρώνες κατάλληλοι για αποφάσεις κατηγοριοποίησης.

Γραμμική συνάρτηση μεταφοράς (Linear Transfer Function)

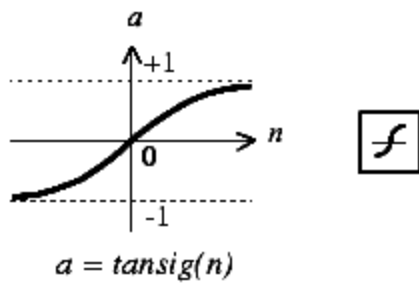


Linear Transfer Function

Εικόνα 2.6: Γραμμική συνάρτηση μεταφοράς.

Κυρίως, χρησιμοποιείται σε νευρώνες που προορίζονται για γραμμική προσέγγιση στα γραμμικά φίλτρα.

Εφαπτομενική - σιγμοειδής συνάρτηση μεταφοράς (Tan-sigmoid Transfer Function)

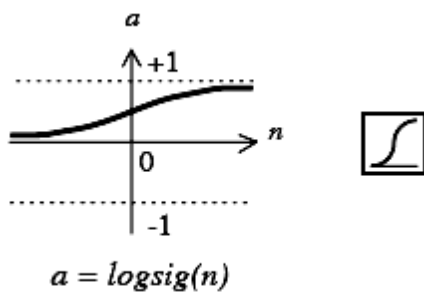


Tan-Sigmoid Transfer Function

Εικόνα 2.7 : Εφαπτομενική – σιγμοειδής συνάρτηση μεταφοράς.

Η είσοδος της μπορεί να είναι οποιαδήποτε τιμή στο διάστημα $(-\infty, +\infty)$. Το πεδίο τιμών της όμως, περιορίζεται στο διάστημα $(-1, 1)$. Αποτελεί την κύρια επιλογή σε δίκτυα τύπου backpropagation λόγω της διαφορισμότητας της.

Λογαριθμική – σιγμοειδής συνάρτηση μεταφοράς (Log-sigmoid Transfer Function)



Log-Sigmoid Transfer Function

Εικόνα 2.8 : Λογαριθμική – σιγμοειδής συνάρτηση μεταφοράς.

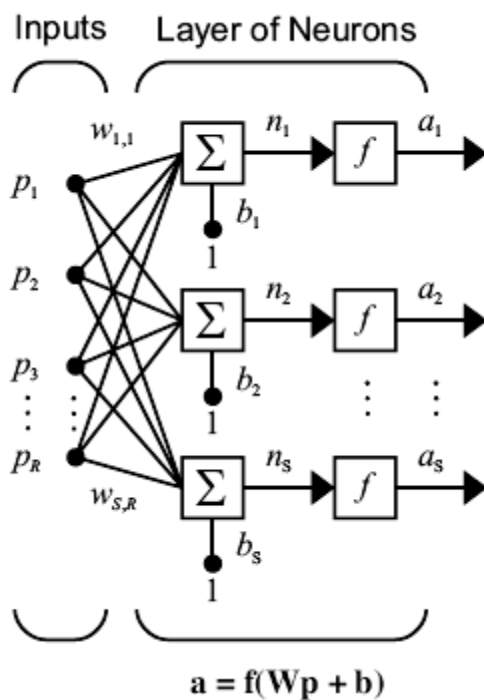
Αυτή η συνάρτηση μεταφοράς παίρνει μια είσοδο, η οποία μπορεί να έχει τιμές από $-\infty$ έως $+\infty$ και περιορίζει την είσοδο στο πεδίο τιμών $(0, 1)$. Είναι κατάλληλη για δίκτυα τύπου backpropagation εφόσον είναι διαφορίσιμη.

IV. Αρχιτεκτονικές δικτύων

Η τοπολογική δομή είναι το κύριο χαρακτηριστικό των ΝΔ και αναφέρεται στην αρχιτεκτονική στην οποία διευθετούνται και διασυνδέονται πολλαπλοί νευρώνες. Οι δύο βασικές ιδιότητες που καθορίζουν την αρχιτεκτονική ενός ΝΔ είναι το *πλήθος των επιπέδων* (layers) και οι *συνδέσεις* μεταξύ των νευρώνων. Το τρίτο χαρακτηριστικό, το οποίο σχετίζεται φυσιολογικά με τον τρόπο κατά τον οποίο είναι δομημένοι οι νευρώνες, είναι ο *αλγόριθμος μάθησης* που χρησιμοποιείται για την εκπαίδευση του δικτύου.

Επίπεδο νευρώνων

Όταν δύο ή περισσότεροι νευρώνες συνδυαστούν τότε σχηματίζουν ένα επίπεδο νευρώνων. Ακολουθεί η γραφική αναπαράσταση ενός δικτύου με ένα επίπεδο νευρώνων με R στοιχεία εισόδου και S νευρώνες.

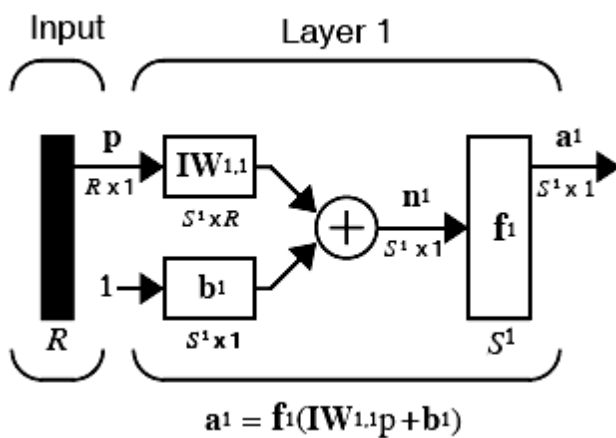


Εικόνα 2.9 : Επίπεδο νευρώνων.

Σε αυτό το δίκτυο κάθε στοιχείο του διανύσματος εισόδου συνδέεται με την είσοδο κάθε νευρώνα μέσω του πίνακα των συντελεστών βαρύτητας W . Ο i -στός νευρώνας έχει έναν αθροιστή ο οποίος συλλέγει τις σταθμισμένες εισόδους και τα bias για να σχηματίσει τη δική του βαθμωτή είσοδο $n(i)$. Εάν συνδυαστούν τα διάφορα $n(i)$, δημιουργούν ένα διάνυσμα εισόδου n με πλήθος στοιχείων S . Τελικά, το νευρωνικό επίπεδο έχει ως έξοδο ένα διάνυσμα a . Να σημειωθεί ότι είναι σύνηθες ο αριθμός των εισόδων σε ένα επίπεδο να είναι διαφορετικός από τον αριθμό των νευρώνων που το αποτελούν. Η γραμμή του πίνακα υποδεικνύει τον νευρώνα στον οποίο θα εφαρμοστεί αυτό το βάρος, ενώ η στήλη υποδηλώνει την είσοδο.

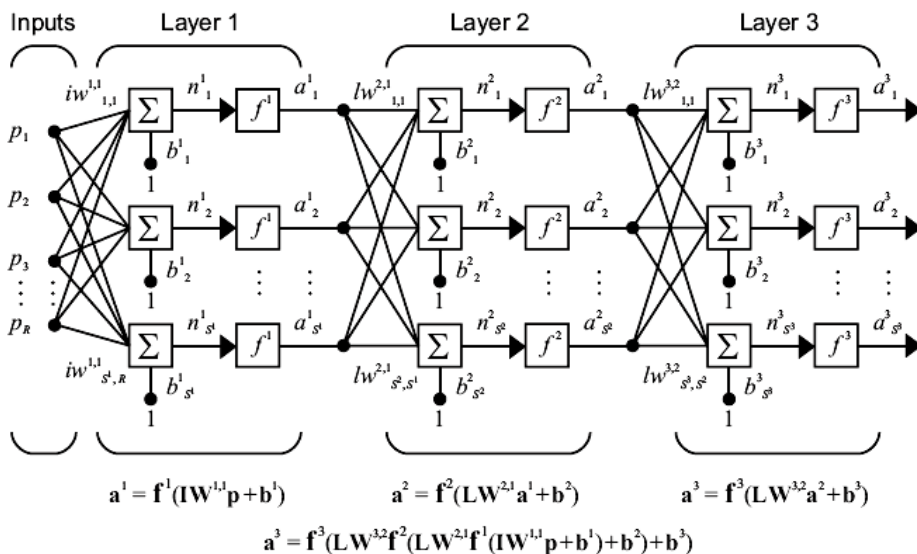
Πολλαπλά επίπεδα νευρώνων (Multiple Layers of Neurons).

Για να περιγραφούν τα δίκτυα πολλαπλών επιπέδων, πρέπει να προηγηθούν ορισμένες διευκρινήσεις. Ειδικότερα, πρέπει να γίνει διαχωρισμός μεταξύ των πινάκων των βαρών που συνδέονται με τις εισόδους και των πινάκων των συντελεστών βαρύτητας, οι οποίοι συνδέονται με τα διάφορα επίπεδα. Θα καλούνται λοιπόν τους πίνακες που συνδέονται με τις εισόδους, *βάρη εισόδων*, ενώ τους πίνακες που συνδέονται με τις εξόδους των νευρωνικών επιπέδων, *βάρη επιπέδων*. Για να γίνεται αντιληπτό σε ποιο επίπεδο αναφέρεται η κάθε παράμετρος γίνεται χρήση ενός δείκτη. Από εδώ και στο εξής θα υιοθετηθεί η παραπάνω σύμβαση για την περαιτέρω περιγραφή των ΝΔ. Το παρακάτω σχήμα αναπαριστά γραφικά ένα επίπεδο ενός πολυεπίπεδου ΝΔ.



Εικόνα 2.10 : Μοντέλο ενός επιπέδου πολυεπίπεδου ΝΔ.

Όπως αναφέρθηκε και παραπάνω ένα ΝΔ μπορεί να αποτελείται από αρκετά επίπεδα. Κάθε επίπεδο έχει ένα πίνακα W, ένα bias b και ένα διάνυσμα εξόδου a. Ο πάνω συντελεστής που χρησιμοποιείται στα διάφορα μεγέθη σκοπό έχει να καθορίσει σε ποιο επίπεδο αναφέρονται οι πίνακες βαρών, τα διανύσματα εξόδου κτλ.(βλ. Εικόνα 2.11).



Εικόνα 2.11 : Πολυεπίπεδο ΝΔ.

Το παραπάνω δίκτυο έχει R1 εισόδους, S1 νευρώνες στο πρώτο επίπεδο, S2 νευρώνες στο δεύτερο επίπεδο κοκ. Είναι σύνηθες τα διαφορετικά επίπεδα να έχουν διαφορετικό αριθμό νευρώνων. Η είσοδος του bias είναι σταθερή ποσότητα για κάθε νευρώνα, ίση με 1. Από το σχήμα φαίνεται επίσης ότι η έξοδος ενός ενδιάμεσου επιπέδου αποτελεί την είσοδο για το ακόλουθο επίπεδο. Επομένως, τα επίπεδο 2 μπορεί να αναλυθεί ως ένα επίπεδο με S1 εισόδους, S2 νευρώνες και έναν S2xS1 πίνακα βαρών W2. Η είσοδος του επιπέδου 2 είναι a1 και η έξοδος του a2. Ύστερα, από τον καθορισμό των παραπάνω παραμέτρων μπορεί το συγκεκριμένο επίπεδο να αντιμετωπιστεί ως ένα ΝΔ ενός νευρωνικού επιπέδου. Αυτή η προσέγγιση μπορεί να γίνει για κάθε επίπεδο του δικτύου. Τα διάφορα επίπεδα ενός πολυεπίπεδου ΝΔ διαδραματίζουν διαφορετικούς ρόλους το καθένα. Το επίπεδο που παράγει την έξοδο του δικτύου καλείται επίπεδο εξόδου (output layer). Όλα τα υπόλοιπα επίπεδα καλούνται κρυφά επίπεδα (hidden layers).

V. Μάθηση των νευρωνικών δικτύων

Η μάθηση είναι μια θεμελιώδης ικανότητα των ΝΔ η οποία τους επιτρέπει να μαθαίνουν από το περιβάλλον τους και να βελτιώνουν τη συμπεριφορά τους με το πέρασμα του χρόνου. Η μάθηση είναι μια γενική έννοια της επιστήμης των συστημάτων και ορίζεται με ποικίλους τρόπους ανάλογα με το εκάστοτε πεδίο εφαρμογής. Στα ΝΔ, η μάθηση αναφέρεται στη διεργασία επίτευξης μιας επιθυμητής συμπεριφοράς μέσω ανανέωσης της τιμής των συναπτικών βαρών. Έτσι, ένα ΝΔ μαθαίνει για το περιβάλλον του μέσω μιας επαναληπτικής διαδικασίας ανανέωσης (αλλαγής) των συναπτικών βαρών και κατωφλίων. *Αλγόριθμος μάθησης ή εκπαίδευσης* είναι κάθε προκαθορισμένο σύνολο καλά ορισμένων κανόνων επίλυσης του προβλήματος μάθησης (ή εκπαίδευσης) του ΝΔ. Γενικά, υπάρχουν πολλοί αλγόριθμοι μάθησης στα ΝΔ, καθένας από τους οποίους έχει συγκεκριμένα πλεονεκτήματα και μειονεκτήματα. Κάθε αλγόριθμος μάθησης προσφέρει έναν άλλο τρόπο προσαρμογής (επιλογής/ανανέωσης) των συναπτικών βαρών.

Ένα άλλο θέμα το οποίο πρέπει να ληφθεί υπ' όψη κατά τη σχεδίαση ενός ΝΔ είναι το περιβάλλον στο οποίο εργάζεται το ΝΔ. Έτσι, διαφορετικά μοντέλα του περιβάλλοντος έχουν ως αποτέλεσμα την επιλογή διαφορετικών αλγορίθμων μάθησης. Όλες οι μέθοδοι μάθησης μπορούν να καταταχθούν σε δύο κατηγορίες: τη **μάθηση με επίβλεψη** (supervised learning) και τη **μάθηση χωρίς επίβλεψη** (unsupervised learning).

Μάθηση με επίβλεψη: Η μάθηση αυτή είναι μια διαδικασία η οποία συνδυάζει έναν εξωτερικό εκπαιδευτή και τη συνολική ή γενικευμένη πληροφορία. Κάποιες από τις μεθόδους οι οποίες συγκαταλέγονται σε αυτή την κατηγορία είναι η μάθηση με διόρθωση σφάλματος, η στοχαστική μάθηση. Παραδείγματα τα οποία αντιπροσωπεύουν την μάθηση με επίβλεψη συμπεριλαμβάνουν αποφάσεις για το πότε θα πρέπει να σταματήσει η διαδικασία εκπαίδευσης, αποφάσεις αναφορικά με τη συχνότητα παρουσίασης στο δίκτυο τα πρότυπα εκπαίδευσης και η παρουσίαση προόδου του δικτύου. Η μάθηση με επίβλεψη χωρίζεται σε δύο ακόμα κατηγορίες: στη **δομική** (structural) και στην **προσωρινή** (temporal) εκμάθηση. Οι αλγόριθμοι οι οποίοι βρίσκονται στην πρώτη κατηγορία, χρησιμοποιούνται για την εύρεση της βέλτιστης σχέσης μεταξύ εισόδων και εξόδων για κάθε ξεχωριστό ζευγάρι προτύπων. Παραδείγματα της δομικής εκμάθησης αποτελούν η αναγνώριση και η κατηγοριοποίηση προτύπων, ενώ παραδείγματα της προσωρινής εκμάθησης η πρόβλεψη και ο έλεγχος.

Μάθηση χωρίς επίβλεψη : Οι αλγόριθμοι της εν λόγω μάθησης αναφέρονται ως αυτό-οργανώμενοι (self-organized) και είναι διαδικασίες οι οποίες δεν απαιτούν να είναι παρών ένας «εξωτερικός» δάσκαλος ή επιβλέπων. Βασίζονται, μάλιστα, μόνο σε τοπική πληροφορία καθ' όλη τη διάρκεια της εκπαίδευσης του Τεχνητού Νευρωνικού Δικτύου. Οι συγκεκριμένοι αλγόριθμοι οργανώνουν τα δεδομένα και ανακαλύπτουν τις σημαντικές συλλογικές ιδιότητες. Για παράδειγμα, αλγόριθμοι εκπαίδευσης χωρίς επίβλεψη είναι ο αλγόριθμος Hebbian, ο διαφορικός αλγόριθμος Hebbian και ο Min-Max αλγόριθμος.

Κατά κύριο λόγο οι περισσότερες διαδικασίες εκπαίδευσης είναι off line. Όταν χρησιμοποιείται όλο το δείγμα προτύπων για την τροποποίηση των τιμών των βαρών, πριν την τελική χρήση του δικτύου ως εφαρμογή, τότε ονομάζεται off line εκπαίδευση. Οι αλγόριθμοι εκπαίδευσης off line έχουν την απαίτηση να βρίσκονται στην εκπαίδευση του δικτύου παρόντα όλα τα πρότυπα. Το γεγονός αυτό αποκλείει την πιθανότητα εισαγωγής νέων πληροφοριών μέσω νέων προτύπων. Βέβαια, υπάρχουν και Τεχνητά Νευρωνικά Δίκτυα τα οποία δεν αποκλείουν την εισαγωγή νέας πληροφορίας, μετά την τελική τους μοντελοποίηση. Αν παρουσιαστεί ανάγκη εισαγωγής νέου προτύπου στο δίκτυο, μπορεί να γίνει απευθείας χωρίς τον κίνδυνο να χαθεί κανένα μέρος της αρχικής πληροφορίας. Το πλεονέκτημα των δικτύων που χρησιμοποιούν off line διαδικασίες εκπαίδευσης επικεντρώνεται κυρίως στη δυνατότητα να δίνουν καλύτερες λύσεις σε δύσκολα προβλήματα.

Τα δεδομένα σε αυτή την εργασία εκπαιδεύτηκαν με τον Αλγόριθμο Ανάστροφης Διάδοσης (back-propagation). Στον Αλγόριθμο Ανάστροφης Διάδοσης τα MLPs εκπαιδεύονται με έναν επιβλεπόμενο τρόπο. Αυτός ο αλγόριθμος βασίζεται στον κανόνα μάθησης διόρθωσης του λάθους. Η διαδικασία της πίσω διάδοσης του λάθους αποτελείται από δυο περάσματα διαμέσου των διαφορετικών επιπέδων του δικτύου ένα προς τα εμπρός πέρασμα και ένα προς τα πίσω πέρασμα. Στο εμπρός πέρασμα ένα διάνυσμα εισόδου εφαρμόζεται στους νευρώνες εισόδου του δικτύου. Η επίδραση του διαδίδεται μέσα στο δίκτυο από επίπεδο σε επίπεδο. Ένα σύνολο από εξόδους παράγεται ως τη πραγματική απόκριση του δικτύου. Κατά τη διάρκεια του εμπρός περάσματος τα βάρη του δικτύου είναι σταθερά. Κατά την πίσω διάδοση, η πραγματική απόκριση του δικτύου αφαιρείται από την επιθυμητή απόκριση για την παραγωγή ενός σήματος λάθους. Το σήμα λάθους διαδίδεται προς τα πίσω στο δίκτυο. Τα βάρη προσαρμόζονται σε συμφωνία με τον κανόνα διόρθωσης λάθους.

Τα πλεονεκτήματα της μεθόδου Backpropagation :

1. Λίγες παράμετροι προς ρύθμιση
2. Αλγόριθμος εύκολος σε υλοποίηση
3. Μπορεί να εφαρμοστεί σε ευρεία περιοχή δεδομένων
4. Πολύ δημοφιλής

Τα μειονεκτήματα της μεθόδου Backpropagation :

1. Η εκμάθηση είναι αργή
2. Τα νέα στοιχεία θα υπερκαλύψουν τα παλαιά εκτός αν συνεχίσουν να παρέχονται
3. Δύσκολο να διατηρηθεί το δίκτυο ενημερωμένο
4. Το δίκτυο είναι ουσιαστικά black box
5. Δεν μπορεί να υπάρξει εγγύηση γενίκευσης ακόμα και με ελάχιστο σφάλμα

ΚΕΦΑΛΑΙΟ 3

I. Μέθοδος επίλυσης του προβλήματος

Στο κεφάλαιο αυτό, γίνεται περιγραφή της μεθόδου επίλυσης του προβλήματος, το οποίο πραγματεύεται η παρούσα διπλωματική εργασία. Η λύση του προβλήματος επιχειρείται μέσω της χρήσης Νευρωνικών Δικτύων Ανάστροφης Διάδοσης (backpropagation neural networks) στη προσπάθεια να βρεθεί μια άμεση σχέση μεταξύ της υπολογισμένης τάσης παραμόρφωσης για δεδομένη τροπή της πλάκας.

Για την λύση του αντίστροφου προβλήματος έχει προηγηθεί η λύση του ευθέως προβλήματος (υπολογισμός τάσεων) για την απόκτηση των απαραίτητων δεδομένων, τα οποία αποτελούν τα δεδομένα εκπαίδευσης του ΝΔ.

Για την εκπαίδευση και τον έλεγχο, χρησιμοποιούνται οι πίνακες input και output. Ο πίνακας input περιέχει τις εισόδους στο νευρωνικό δίκτυο ενώ ο πίνακας output τις εξόδους του δικτύου. Επίσης το πλήθος των νευρώνων του κρυφού επιπέδου μεταβάλλεται, από 2 έως 20 με βήμα 2, ώστε να βρεθεί η αρχιτεκτονική του δικτύου που βγάζει τα βέλτιστα δυνατά αποτελέσματα. Τέλος υπάρχει και ένας τρίτος πίνακας με όνομα target ο οποίος περιέχει την επιθυμητή έξοδο του δικτύου και χρησιμοποιείται κατά την διαδικασία της εκπαίδευσης.

Αρχικά, για να δημιουργηθούν τα δεδομένα εισόδου έγιναν δοκιμές στο πρόγραμμα Comsol για τη δημιουργία της φόρτισης του RVE. Συγκεκριμένα, θα δημιουργηθούν συνολικά 4 μοντέλα με το πρόγραμμα αυτό. Στο κάθε ένα από αυτά, θα εφαρμοσθεί φόρτιση τροπή ακριβώς όπως δίνεται στους πίνακες του excel που παρατίθενται παρακάτω, με τις ονομασίες: ΑΡΧΕΙΟ COMSOL 1 - ΚΕΝΤΡΙΚΗ ΕΠΙΛΥΣΗ ΑΝΑΦΟΡΑΣ, ΑΡΧΕΙΟ COMSOL 2, ΑΡΧΕΙΟ COMSOL 3, ΑΡΧΕΙΟ COMSOL 4. Το ζητούμενο για κάθε τιμή της παραμέτρου (δηλαδή για κάθε επίπεδο του **διανύσματος** της φόρτισης τροπών) είναι η μέση τάση του RVE (**διάνυσμα τάσεων**).

II. Χρήση COMSOL για την εύρεση τάσεων, για αυξανόμενο επίπεδο τροπής

<u>ΑΡΧΕΙΟ COMSOL 1 - ΚΕΝΤΡΙΚΗ ΕΠΙΛΥΣΗ ΑΝΑΦΟΡΑΣ</u>		
	Loading path 1	
	<u>ε: macro τροπή</u>	Αυτό μπαίνει σε υπολογισμούς Comsol, κλασικά, σαν φόρτιση τροπής (φυσικά μπαίνει και σε input νευρωνικού δικτύου)
εxx1	0.01*param	
εyy1	0.02*param	
εxy1	0.03*param	
	param: 0.1:0.1:1	

Πίνακας 1 : ΑΡΧΕΙΟ COMSOL 1 - ΚΕΝΤΡΙΚΗ ΕΠΙΛΥΣΗ ΑΝΑΦΟΡΑΣ

Αυξανόμενα επίπεδα τροπής				
param: Παράμετρος σε Comsol		εχx1	εγγ1	εχγ1
0,1		0,001	0,002	0,003
0,2		0,002	0,004	0,006
0,3		0,003	0,006	0,009
0,4		0,004	0,008	0,012
0,5		0,005	0,01	0,015
0,6		0,006	0,012	0,018
0,7		0,007	0,014	0,021
0,8		0,008	0,016	0,024
0,9		0,009	0,018	0,027
1		0,01	0,02	0,03

Πίνακας 2 : Αυξανόμενα επίπεδα τροπής στο ΑΡΧΕΙΟ COMSOL 1

Αποτέλεσμα Comsol: Τάση				
σxχ1		σγγ1		σxγ1
Λαμβάνονται από επιλύσεις comsol				

Πίνακας 3 : Αποτελέσματα για το ΑΡΧΕΙΟ COMSOL 1 – Τάσεις

III. Χρήση COMSOL για την εύρεση επαπτομενικού μητρώου ακαμψίας, ανά επίπεδο τροπής

Λαμβάνονται καινούρια αρχεία comsol, στα οποία εφαρμόζεται μικρή διαφοροποίηση, στις τιμές ϵ_{xx} , ϵ_{yy} , ϵ_{xy} . Η διαφορά των τροπών μεταξύ των νέων αρχείων και του αρχικού-κεντρικού αρχείου, θα δώσει τα «δε» ανά επίπεδο φόρτισης.

<u>ΑΡΧΕΙΟ COMSOL 2</u>			
		Loading path 1	
		<u>ϵ: macro τροπή</u>	Αυτό μπαίνει σε υπολογισμούς Comsol, κλασικά, σαν φόρτιση τροπής (φυσικά μπαίνει και σε input νευρωνικού δικτύου)
ϵ_{xx2}		$0.01*param+0.001$	
ϵ_{yy2}		$0.02*param$	
ϵ_{xy2}		$0.03*param$	
		param: 0.1:0.1:1	

Πίνακας 4 : ΑΡΧΕΙΟ COMSOL 2

Αυξανόμενα επίπεδα τροπής				
param: Παράμετρος σε Comsol	ϵ_{xx2}	ϵ_{yy2}	ϵ_{xy2}	
0,1	0,002	0,002	0,003	
0,2	0,003	0,004	0,006	
0,3	0,004	0,006	0,009	
0,4	0,005	0,008	0,012	
0,5	0,006	0,01	0,015	
0,6	0,007	0,012	0,018	
0,7	0,008	0,014	0,021	
0,8	0,009	0,016	0,024	
0,9	0,01	0,018	0,027	
1	0,011	0,02	0,03	

Πίνακας 5 : Αυξανόμενα επίπεδα τροπής στο ΑΡΧΕΙΟ COMSOL 2

Αποτέλεσμα Comsol: Τάση				
σxx2		σyy2		σxy2
Λαμβάνονται από επιλύσεις comsol				

Πίνακας 6 : Αποτελέσματα για το ΑΡΧΕΙΟ COMSOL 2 – Τάσεις

Εύρεση increments τροπών	δεxx1	δεyy1	δεxy1
$\delta_{εxx1} = ε_{xx2} - ε_{xx1}$	0,00100	0,00000	0,00000
$\delta_{εyy1} = ε_{yy2} - ε_{yy1}$	0,00100	0,00000	0,00000
$\delta_{εxy1} = ε_{xy2} - ε_{xy1}$	0,00100	0,00000	0,00000
	0,00100	0,00000	0,00000
	0,00100	0,00000	0,00000
	0,00100	0,00000	0,00000
	0,00100	0,00000	0,00000
	0,00100	0,00000	0,00000
	0,00100	0,00000	0,00000
	0,00100	0,00000	0,00000

Πίνακας 7 : Increments Τροπών για $\delta_{εxx1}$, $\delta_{εyy1}$, $\delta_{εxy1}$

<u>ΑΡΧΕΙΟ COMSOL 3</u>				
		Loading path 1		
		<u>ε: macro τροπή</u>	Αυτό μπαίνει σε υπολογισμούς Comsol, κλασικά, σαν φόρτιση τροπής (φυσικά μπαίνει και σε input νευρωνικού δικτύου)	
εxx3		0.01*param		
εyy3		0.02*param+0.001		
εxy3		0.03*param		
		param: 0.1:0.1:1		

Πίνακας 8 : ΑΡΧΕΙΟ COMSOL 3

Αυξανόμενα επίπεδα τροπής				
param: Παράμετρος σε Comsol		εχχ3	εγγ3	εχυ3
0,1		0,001	0,003	0,003
0,2		0,002	0,005	0,006
0,3		0,003	0,007	0,009
0,4		0,004	0,009	0,012
0,5		0,005	0,011	0,015
0,6		0,006	0,013	0,018
0,7		0,007	0,015	0,021
0,8		0,008	0,017	0,024
0,9		0,009	0,019	0,027
1		0,01	0,021	0,03

Πίνακας 9 : Αυξανόμενα επίπεδα τροπής στο ΑΡΧΕΙΟ COMSOL 3

Αποτέλεσμα Comsol: Τάση				
σχη3		σγγ3		σχυ3
Λαμβάνονται από επιλύσεις comsol				

Πίνακας 10 : Αποτελέσματα για το ΑΡΧΕΙΟ COMSOL 3 – Τάσεις

Εύρεση increments τροπών	δεxx2	δεγγ2	δεχγ2
$\delta_{εxx2} = ε_{xx3} - ε_{xx1}$	0,00000	0,00100	0,00000
$\delta_{εγγ2} = ε_{γγ3} - ε_{γγ1}$	0,00000	0,00100	0,00000
$\delta_{εχγ2} = ε_{χγ3} - ε_{χγ1}$	0,00000	0,00100	0,00000
	0,00000	0,00100	0,00000
	0,00000	0,00100	0,00000
	0,00000	0,00100	0,00000
	0,00000	0,00100	0,00000
	0,00000	0,00100	0,00000
	0,00000	0,00100	0,00000
	0,00000	0,00100	0,00000

Πίνακας 11 : Increments Τροπών για $\delta_{εxx2}$, $\delta_{εγγ2}$, $\delta_{εχγ2}$

ΑΡΧΕΙΟ COMSOL 4				
	Loading path 1			
	<u>ε: macro τροπή</u>		Αυτό μπαίνει σε υπολογισμούς Comsol, κλασικά, σαν φόρτιση τροπής (φυσικά μπαίνει και σε input νευρωνικού δικτύου)	
εxx4	0.01*param			
εγγ4	0.02*param			
εχγ4	0.03*param+0.001			
	param: 0.1:0.1:1			

Πίνακας 12 : ΑΡΧΕΙΟ COMSOL 4

Αυξανόμενα επίπεδα τροπής				
param: Παράμετρος σε Comsol	εxx4	εγγ4	εχγ4	
0,1	0,001	0,002	0,004	
0,2	0,002	0,004	0,007	
0,3	0,003	0,006	0,01	
0,4	0,004	0,008	0,013	
0,5	0,005	0,01	0,016	
0,6	0,006	0,012	0,019	
0,7	0,007	0,014	0,022	
0,8	0,008	0,016	0,025	
0,9	0,009	0,018	0,028	
1	0,01	0,02	0,031	

Πίνακας 13 : Αυξανόμενα επίπεδα τροπής στο ΑΡΧΕΙΟ COMSOL 4

Αποτέλεσμα Comsol: Τάση				
σ _{xx4}		σ _{yy4}		σ _{xy4}
Λαμβάνονται από επιλύσεις comsol				

Πίνακας 14 : Αποτελέσματα για το ΑΡΧΕΙΟ COMSOL 4 – Τάσεις

Εύρεση increments τροπών	δε _{xx3}	δε _{yy3}	δε _{xy3}
δε _{xx3} = ε _{xx4} -ε _{xx1}	0,00000	0,00000	0,00100
δε _{yy3} = ε _{yy4} -ε _{yy1}	0,00000	0,00000	0,00100
δε _{xy3} = ε _{xy4} -ε _{xy1}	0,00000	0,00000	0,00100
	0,00000	0,00000	0,00100
	0,00000	0,00000	0,00100
	0,00000	0,00000	0,00100
	0,00000	0,00000	0,00100
	0,00000	0,00000	0,00100
	0,00000	0,00000	0,00100
	0,00000	0,00000	0,00100

Πίνακας 11 : Increments Τροπών για δε_{xx3}, δε_{yy3}, δε_{xy3}

Επίσης, σε κάθε επίπεδο φόρτισης όπου θα υπάρχει ένα ε, το comsol θα έχει δώσει το αντίστοιχο σ. Τα increments των τροπών, δηλαδή τα δε, προκύπτουν όπως φαίνεται παραπάνω. Εντελώς αντίστοιχα προκύπτουν και τα δσ, δηλαδή σαν διαφορές των τάσεων, ανά επίπεδο φόρτισης.

Παράδειγμα: δσ₁=σ₂-σ₁

δσ₂=σ₃-σ₁

δσ₃=σ₄-σ₁

Τελικά, σε κάθε επίπεδο φόρτισης, και για το συγκεκριμένο loading path, μπορεί να βρεθεί το εφαλτομενικό μητρώο ακαμψίας D (διαστάσεων 3x3):

Ανά επίπεδο φόρτισης: δσ=[δσ₁;δσ₂;δσ₃], διαστάσεις 3x3

δε=[δε₁;δε₂;δε₃], διαστάσεις 3x3

$$\delta\sigma = \mathbf{D} * \delta\epsilon \Rightarrow \mathbf{D} = \delta\sigma * \delta\epsilon^{-1}$$

IV. Δεδομένα εισόδου (input)

Στη διπλωματική αυτή εξετάζονται δύο περιπτώσεις που αφορούν την εκπαίδευση του νευρωνικού μας δικτύου. Για το λόγο αυτό θα χρησιμοποιηθούν δύο διαφορετικά ζεύγη δεδομένων εισόδου – εξόδου. Τα αρχεία από τα αποτελέσματα της προσομοίωσης περιέχουν στη πρώτη περίπτωση ένα ζεύγος από πενήντα δεδομένα εισόδου στους τρεις άξονες x , y , xy , δηλαδή το πίνακα strain (50 x 3) ο οποίος περιλαμβάνει τις τροπές του κελύφους περιοδικότητας και έναν πίνακα εξόδου, τον stress με 50 στοιχεία όπου παρουσιάζονται οι αντίστοιχες τάσεις του πίνακα strain. Ο πίνακας εισόδου strain (50 x 3) στην πρώτη περίπτωση περιέχει τις παραμορφώσεις σε πέντε διαφορετικές καταστάσεις έχοντας δέκα δεδομένα για κάθε κατάσταση. Αυτές οι καταστάσεις είναι η αρχική, σε μηδέν μοίρες, σε 30 μοίρες, σε 60 μοίρες και τέλος σε 90 μοίρες.

Στην δεύτερη περίπτωση ο πίνακας εισόδου διαθέτει επίσης πενήντα στοιχεία στις ίδιες πέντε διαφορετικές καταστάσεις. Ο πίνακας εξόδου ωστόσο είναι διαφορετικός. Αμέσως παρακάτω υπογραμμίζεται και αναλύεται η διαφορά αυτή.

V. Δεδομένα εξόδου (output)

Όπως ήδη αναφέρθηκε, θα μελετηθούν δύο περιπτώσεις για την εκπαίδευση του νευρωνικού δικτύου. Τα δεδομένα εξόδου είναι τα αποτελέσματα του νευρωνικού δικτύου ύστερα από τη διαδικασία της εκπαίδευσης. Τα δεδομένα εξόδου στην πρώτη περίπτωση περιλαμβάνονται σε πέντε ζεύγη των δέκα στοιχείων στον πίνακα stress και αφορούν τις πέντε διαφορετικές καταστάσεις στους τρεις άξονες. Ο πίνακας stress ουσιαστικά περιλαμβάνει τις τάσεις που προέκυψαν από τις παραμορφώσεις. Στην δεύτερη περίπτωση όμως είναι διαφορετικά δομημένοι οι πίνακες εξόδου. Ενώ στην πρώτη περίπτωση, στον πίνακα εξόδου stress (50 x 3), κάθε γραμμή αντιστοιχεί σε κάθε γραμμή του πίνακα εισόδου strain (50 x 3), στην δεύτερη περίπτωση, στον πίνακα εξόδου consistent stiffness (150 x 3) σε κάθε γραμμή του πίνακα strain (50 x 3) αντιστοιχεί ένας πίνακας consistent stiffness (3 x 3). Συνεπώς αφού υπάρχουν πενήντα γραμμές διανυσμάτων τροπών θα υπάρχουν $50 * 3 = 150$ γραμμές με το πίνακα consistent stiffness.

VI. Δεδομένα ελέγχου

Τα δεδομένα ελέγχου του ΝΔ, δημιουργούνται όπως ακριβώς τα δεδομένα εκπαίδευσης και εμπεριέχονται σ' αυτά. Παρόλα αυτά δεν γίνεται εκπαίδευση με αυτά τα δεδομένα διότι με το σετ δεδομένων επιχειρείται η εξέταση της δυνατότητας γενίκευσης του ΝΔ, δηλαδή της συμπεριφοράς του δικτύου, όταν αυτό τροφοδοτείται με άγνωστες τιμές.

VII. Εκπαίδευση δικτύου

1^{ος} Όγκος Δεδομένων (Χρησιμοποιώντας τους πίνακες *strain* [50 x 3] & *stress* [50 x 3])

Στο τμήμα αυτό περιγράφεται η διαδικασία που ακολουθήθηκε για την δημιουργία του νευρωνικού δικτύου. Ειδικότερα δεν δημιουργήθηκε μόνο ένα δίκτυο αλλά έγινε διερεύνηση για την εύρεση της βέλτιστης αρχιτεκτονικής. Όλα τα δίκτυα που δημιουργήθηκαν για την επίλυση του προβλήματος είναι ΝΔ προσοτροφοδότησης (feed-forward) μέχρι δύο κρυφά στρώματα. Επιπλέον, όλα εκπαιδεύτηκαν με τον Αλγόριθμο Ανάστροφης Διάδοσης (back-propagation).

Αρχικά εξετάστηκε το σενάριο όπου υπάρχει ένα κρυφό στρώμα και ο αριθμός των νευρώνων στο κρυφό στρώμα θα μεταβάλλεται από 2 σε 20 με βήμα 2. Το δίκτυο τέθηκε προς εκπαίδευση 4 φορές. Επίσης τα δεδομένα εκπαίδευσης του νευρωνικού δικτύου αποτελούν ένας πίνακας εισόδου *strain_for_train* (30 x 3) και ένας πίνακας εξόδου *stress_for_train* (30 x 3) οι οποίοι περιέχουν δεδομένα από τους αρχικούς πίνακες *strain* (50 x 3) και *stress* (50 x 3). Όσον αφορά τους πίνακες εκπαίδευσης, επιλέχθηκαν με τρόπο αυθαίρετο διαλέγοντας τα δεδομένα που αφορούν την αρχική κατάσταση, την κατάσταση στις 30 μοίρες και την κατάσταση στις 90 μοίρες. Ο τρόπος που έχει δομηθεί ο παρουσιαζόμενος κώδικας είναι τέτοιος ώστε να επιστρέφει και να εμφανίζει στο command window της MatLab το βέλτιστο μέγιστο ποσοστό απόκλισης επί τοις εκατό από την πραγματική τιμή, ενημερώνοντάς παράλληλα τον χρήστη για τον αριθμό των νευρώνων που χρειάστηκαν για να επιτευχθεί αυτό και για τον αριθμό των εκπαιδεύσεων που απαιτήθηκαν. Παρακάτω παρατίθεται το κώματι του κώδικα που μόλις περιγράφηκε.

26

```

y=10;
f=1;
k=1;
while k<=50
    if k>=1 && k<11 | k>20 && k<31 | k>40 && k<46
        for l=f:y
            for o=1:3
                % toy exw dwseis san dedomena eisodou gia ekpaideusi tis
                paramorfwsis stin arxiki stis 30 moires
                %kai stis 60 moires kathws epiasis kaai gia ta dedomena
                eksodou apo tous
                %sinolikous pinakes 50x3
                strain_for_train(l,o)=strain(k,o);
                stress_for_train(l,o)=stress(k,o);
            end
            k=k+1;
        end
        f=y+1;
        y=y+10;
    end
    k=k+1;
end

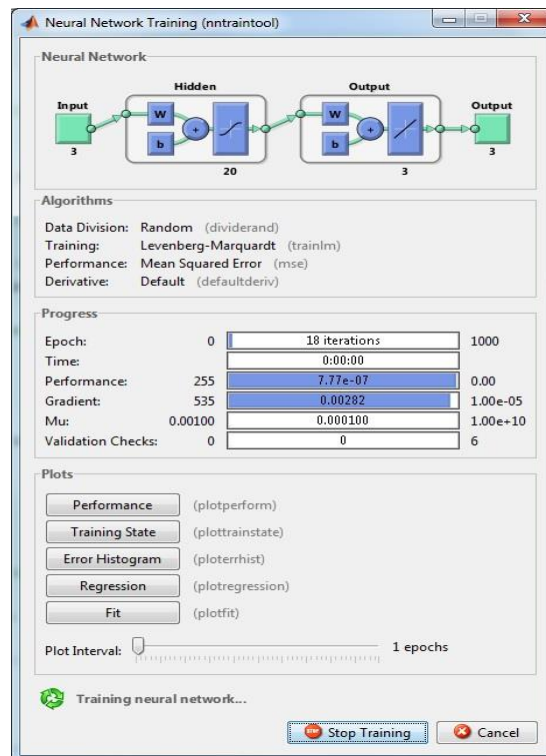
```

```
fores_ekpaideusis=4; %fores που θα εκπαιδεύει το δίκτυο
%ari8mos neurwnwn
elaxistos_arithmos_neurwnwn=2;
vima_auksisis_neurwnwn=2;
megistos_arithmos_neurwnwn=20;
```

Και τέλος αυτό που μου επιστρέφει και μου εμφανίζει στο command window της MatLab το κάνει αυτή η εντολή.

```
fprintf(1, ' pososto la8wn: %.2f number of neurons:%d
trainings %d \n\n', e*100, i, j);
```

Με το πάτημα του κουμπιού της έναρξης και την εκκίνηση της εκπαίδευσης του νευρωνικού δικτύου, αρχίζει μια πολύπλοκη διαδικασία. Αναλυτικότερα, γίνεται χρήση της συνάρτησης `feedforwardnet`, για την δημιουργία του ΝΔ και της συνάρτησης `'trainlm'` (Levenberg-Marquardt backpropagation) για την εκπαίδευση του. Η συνάρτηση ενεργοποίησης που χρησιμοποιήθηκε για τους νευρώνες του κρυφού επιπέδου είναι η σιγμοειδής συνάρτηση **tansig** ενώ όσον αφορά τους νευρώνες στο στρώμα εξόδου χρησιμοποιήθηκε η γραμμική συνάρτηση ενεργοποίησης **purelin**. Η **tansig** υπολογίζει την έξοδο της βάση του τύπου $f(x) = 2/(1+\exp(-2*x))-1$ ενώ η **purelin** είναι ουσιαστικά η $f(x)=x$. Τα δεδομένα διαιρούνται σε σύνολα εκπαίδευσης (training), επαλήθευσης (validation) και ελέγχου (test) με την συνάρτηση **dividerand** με ποσοστό 70%, 15% και 15% αντίστοιχα. Για τη βελτίωση των αποτελεσμάτων η εκπαίδευση για κάθε αριθμό δεδομένων εισόδου και νευρώνων πραγματοποιείται τέσσερις φορές και επιλέγεται το δίκτυο με την καλύτερη έξοδο. Οι παραπάνω συναρτήσεις αποτελούν μέρος του Neural Network Toolbox του λογισμικού MATLAB (R2011a), το οποίο χρησιμοποιήθηκε για την υλοποίηση όλων των ΝΔ.



Εικόνα 3.12 : Απεικόνιση της αρχιτεκτονικής του δικτύου 3-20-3

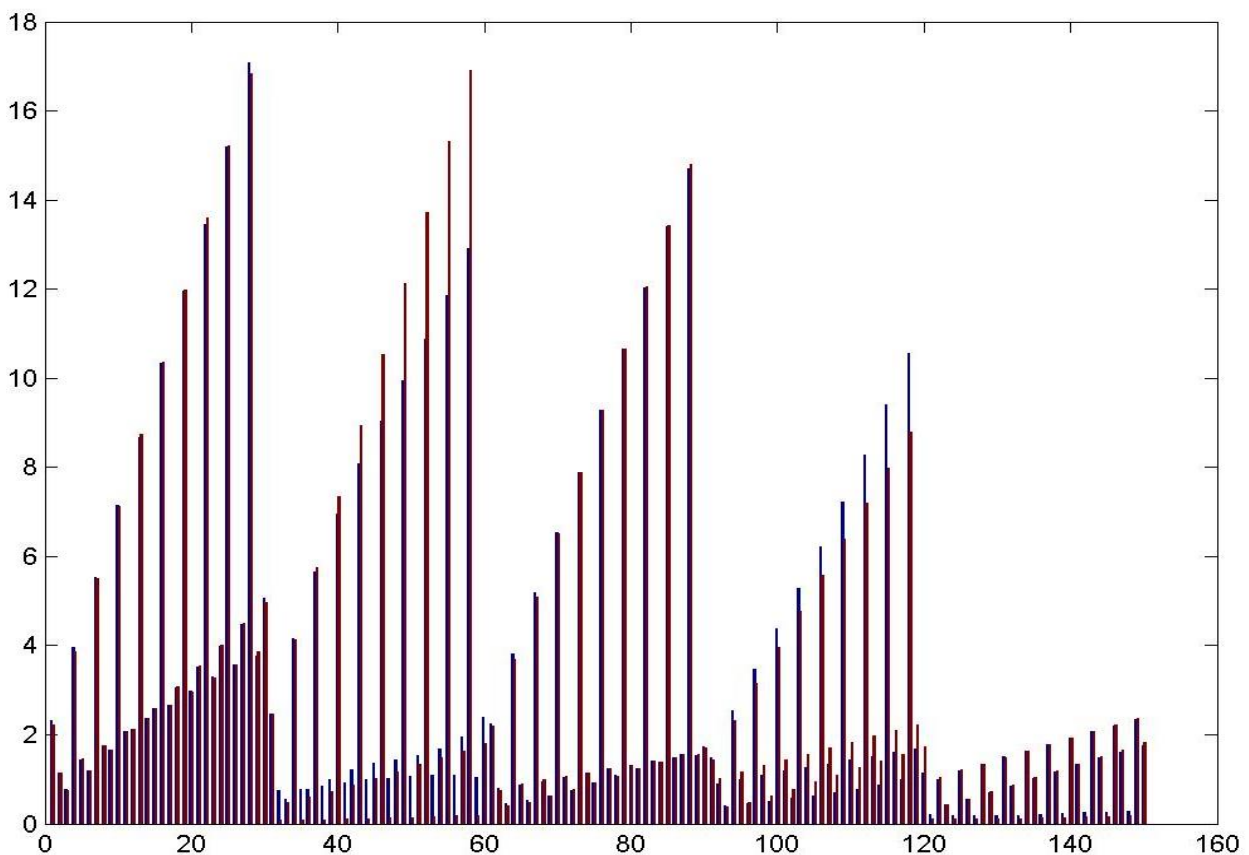
Τα αποτελέσματα που επέστρεψε το πρόγραμμα είναι:

>>Το ποσοστό λάθων είναι: 91% με αριθμό νευρώνων: 4 μετά από 4 εκπαιδεύσεις

>>Το ποσοστό λάθων είναι: 90% με αριθμό νευρώνων: 14 μετά από 4 εκπαιδεύσεις

>>Το ποσοστό λάθων είναι: 88% με αριθμό νευρώνων: 16 μετά από 4 εκπαιδεύσεις

Στον πίνακα αποτυπώνεται γραφικά το συνολικό αποτέλεσμα σφάλματος σε σχέση με την πραγματική τιμή και την τιμή η οποία προκύπτει από την εκπαίδευση του νευρωνικού δικτύου.



Εικόνα 3.13 : Απόκλιση μεταξύ πραγματικής και εκτιμώμενης τιμής εξόδου με τυχαία διαίρεση δεδομένων και με δεδομένα εκπαίδευσης τις καταστάσεις αρχική, γωνία 30 και γωνία 90 μοίρες.

Τα εξαγόμενα από αυτό το κώδικα αποτελέσματα δεν ήταν τα επιθυμητά και γι' αυτό το λόγο τροποποιήθηκε. Η αλλαγή γίνεται στο κομμάτι του κώδικα που αφορά την συνάρτηση `neurwniko_test_3` που καλεί ο main code και είναι η συνάρτηση μέσα στην οποία εκπαιδεύεται ουσιαστικά το νευρωνικό μας δίκτυο. Ο προηγούμενος τρόπος που χρησιμοποιήθηκε από τη MATLAB ήταν τυχαίος για την διαίρεση των δεδομένων εκπαίδευσης κάνοντας χρήση της συνάρτησης `'dividerand'` ενώ τώρα χρησιμοποιείται μια διαφορετική προσέγγιση. Η προσέγγιση αυτή για την διαίρεση των δεδομένων εκπαίδευσης πραγματοποιείται κάνοντας χρήση της συνάρτησης `'divideind'` και ο διαχωρισμός των δεδομένων θα γίνει ως εξής :

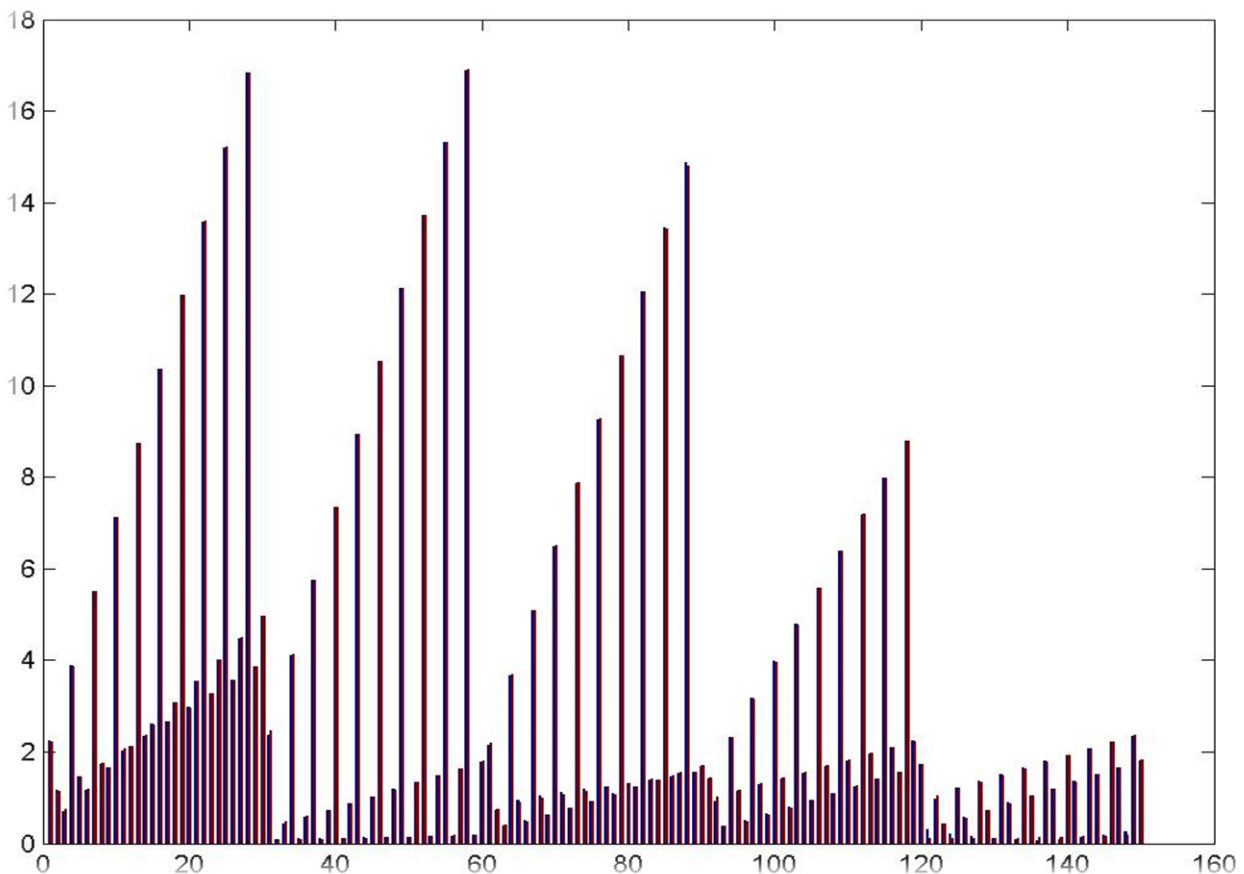
Όπου γ το μέγεθος του πίνακα :

```
net.divideParam.trainInd = 1:3: $\gamma$ ;      %για την εκπαίδευση  
net.divideParam.valInd   = 2:3: $\gamma$ ;      %για validation  
net.divideParam.testInd  = 3:3: $\gamma$ ;      %για test
```

Η αλλαγή αυτή εξήγαγε πολύ καλύτερα αποτελέσματα όσον αφορά το ποσοστό απόκλισης από την πραγματική τιμή μετά την εκπαίδευση του ΝΔ.

```
>>to pososto la8wn einai : 64976.22 % me aritmo neurwnwn :  
2 epeita apo 4 ekpaideuseis  
>>to pososto la8wn einai : 83.87 % me aritmo neurwnwn : 4  
epeita apo 4 ekpaideuseis  
>>to pososto la8wn einai : 80.71 % me aritmo neurwnwn : 8  
epeita apo 4 ekpaideuseis
```

δηλαδή το ποσοστό σφάλματος είναι **80.71 %** και το ιστόγραμμα στο οποίο απεικονίζεται η διαφορά της πραγματικής τιμής και της τιμής εξόδου από το νευρωνικό δίκτυο.



Εικόνα 3.14 : Απόκλιση μεταξύ πραγματικής και εκτιμώμενης τιμής εξόδου με μη τυχαία διαίρεση δεδομένων.

Και πάλι όμως το αποτέλεσμα αυτό δεν ήταν επαρκώς ικανοποιητικό ώστε να σταματήσει εδώ η διερεύνηση και κατ' επέκταση η εκπαίδευση του νευρωνικού δικτύου. Διερευνώντας μια άλλη αρχιτεκτονική που ενδεχομένως να επιφέρει καλύτερα αποτελέσματα, κατέστη σαφές ότι με την τυχαία επιλογή δεδομένων από το πίνακα εισόδου strain και των αντίστοιχων δεδομένα από το πίνακα εξόδου stress και εν συνεχεία την εκ νέου αλλαγή του τρόπου διαίρεσης των δεδομένων σε τυχαίο, τα εξαγόμενα αποτελέσματα είναι καλύτερα από τα προηγούμενα. Ειδικότερα, λήφθηκαν πέντε δεδομένα από κάθε κατάσταση δηλαδή πέντε από την αρχική, πέντε από την κατάσταση στις μηδέν μοίρες, πέντε στις 30 μοίρες, πέντε στις 60 μοίρες και πέντε στις 90 μοίρες δημιουργώντας έτσι τους πίνακες εκπαίδευσης, για την είσοδο strain_for_train (25 x 3) και για την έξοδο stress_for_train (25 x 3). Ακόμη δημιουργήθηκαν συμπληρωματικοί πίνακες των πινάκων αυτών που προορίζονται για εκπαίδευση, ώστε να εξάγουν και αυτοί ένα ποσοστό σφάλματος για την ομάδα δεδομένων εκπαίδευσης – ελέγχου του ΝΔ. Οι συμπληρωματικοί πίνακες ουσιαστικά είναι αυτοί που δημιουργήθηκαν για να συμβάλλουν στην επαλήθευση και τον έλεγχο του νευρωνικού δικτύου. Και αυτοί αντίστοιχα είναι strain_for_test (25 x 3) και stress_for_test (25 x 3). Οι πίνακες που αφορούν την εκπαίδευση δημιουργούνται με τις παρακάτω εντολές

```
k=1;
for i=1:2:50
    for j=1:3
        % toy exw dwseis san dedomena eisodou gia ekpaideusi tis
        paramorfwseis stin arxiki stis 30 moires
        %kai stis 60 moires kathws episis kaai gia ta dedomena
        eksodou apo tous
        %sinolikous pinakes 50x3
                strain_for_train(k,j)=strain(i,j);
                stress_for_train(k,j)=stress(i,j);
    end
    k=k+1;
end
```

και αντίστοιχα οι πίνακες οι συμπληρωματικοί θα δημιουργηθούν με τις παρακάτω ίδιες εντολές με την μόνη διαφορά να εντοπίζεται στην πρώτη επανάληψη

```
k=1;
for i=2:2:50
    for j=1:3
        % toy exw dwseis san dedomena eisodou gia test tis
        paramorfwseis stin arxiki stis 30 moires
        %kai stis 60 moires kathws episis kaai gia ta dedomena
        eksodou apo tous
        %sinolikous pinakes 50x3
                strain_for_test(k,j)=strain(i,j);
                stress_for_test(k,j)=stress(i,j);
    end
    k=k+1;
End
```

Στην συνάρτηση, όπου εκπαιδεύεται το νευρωνικό μας δίκτυο, υπολογίζεται το μέγιστο σφάλμα δηλαδή το ποσοστό απόκλισης από την πραγματική τιμή αυτό πραγματοποιείται με τις ακόλουθες εντολές :

```
%είναι το ποσοστό αποκλίσης από την επιθυμητή τιμή  
error_percentage = (gsubtract(net(A),B)./net(A));  
%i megaliteri apoklisi pou emfanizete sta dedomena eksodou  
errors=max(max(abs(error_percentage)));
```

Οι παραπάνω εντολές θα επιστρέψουν το μέγιστο σφάλμα όσο αφορά το **σύνολο των δεδομένων**, δηλαδή για τους πίνακες strain (50 x 3) και stress (50 x 3). Στη προκειμένη όμως έρευνα έχουν δημιουργηθεί άλλοι τέσσερις πίνακες από τους οποίους θα εξαχθεί το μέγιστο σφάλμα, με τη μόνη αλλαγή μέσα σε αυτές τις εντολές των μεταβλητών.

```
%είναι το ποσοστό αποκλίσης από την επιθυμητή τιμή  
error_percentage = (gsubtract(net(strain_for_test),  
stress_for_test)./net(strain_for_test));  
%i megaliteri apoklisi pou emfanizete sta dedomena eksodou  
errors=max(max(abs(error_percentage)));
```

Οι παραπάνω εντολές θα επιστρέψουν το μέγιστο ποσοστό απόκλισης αναφορικά με τα δεδομένα που προορίζονται για τον **έλεγχο** του δικτύου.

```
%είναι το ποσοστό αποκλίσης από την επιθυμητή τιμή  
error_percentage = (gsubtract(net(strain_for_train),  
stress_for_train)./net(strain_for_train));  
%i megaliteri apoklisi pou emfanizete sta dedomena eksodou  
errors=max(max(abs(error_percentage)));
```

Οι παραπάνω εντολές θα επιστρέψουν το μέγιστο ποσοστό απόκλισης όσον αφορά τα δεδομένα που προορίζονται για **εκπαίδευση** του δικτύου.

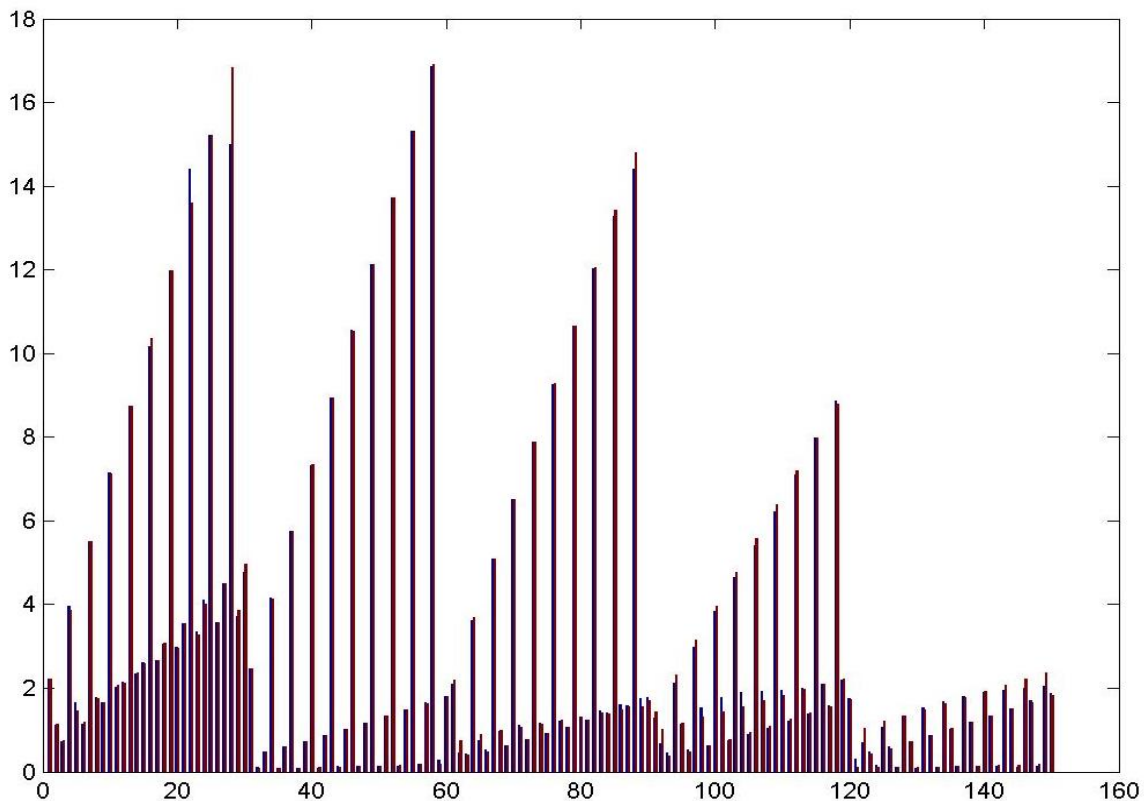
Όπως είναι αναμενόμενο, δύο από τα τρία σφάλματα που θα εξαχθούν θα πρέπει να ταυτίζονται. Έτσι τρέχοντας το κώδικα, λήφθηκαν τα παρακάτω αποτελέσματα. Αρχικά για τα **συνολικά** δεδομένα, τα αποτελέσματα από αυτή τη δοκιμή είναι εμφανώς καλύτερα από ότι πριν :

```
>>To pososto la8wn einai: 952.72 % me arithmo neurwnwn sto  
kryfo epipedo: 2 meta apo 4 ekpaideuseis  
>>To pososto la8wn einai: 180.19 % me arithmo neurwnwn sto  
kryfo epipedo: 4 meta apo 4 ekpaideuseis  
>>To pososto la8wn einai: 79.83 % me arithmo neurwnwn sto  
kryfo epipedo: 6 meta apo 4 ekpaideuseis  
>>To pososto la8wn einai: 68.13 % me arithmo neurwnwn sto  
kryfo epipedo: 10 meta apo 4 ekpaideuseis
```

Εν συνεχεία παρατηρείται ότι το σφάλμα για τα δεδομένα ελέγχου ταυτίζεται με αυτό των συνολικών δεδομένων 68.13 % και τέλος ότι το σφάλμα για τα δεδομένα εκπαίδευσης είναι 57.38%. Συνεπώς διακρίνεται ότι το ποσοστό είναι **68.13 %** σε σχέση με πριν που ήταν 80.71 %, άρα η στρατηγική που ακολουθήθηκε ήταν αποτελεσματική.

Το παρακάτω ιστόγραμμα που μας απεικονίζει την απόκλιση που υπάρχει μεταξύ της πραγματικής και εκτιμώμενης τιμής εξόδου στο σύνολο των δεδομένων. Το γράφημα προκύπτει χρησιμοποιώντας μια συνάρτηση την οποία καλούμε μέσα στο κώδικα μας.

```
%dhmiourgia grafhmatos
han=figure;
set(han, 'Visible', 'off');
all=bar_data(net(A),B);
cd(str);
%apo8hkeush grafhmatos
print(han, '-
djpeg', 'Histogramm_test4_1layer.jpeg')
close(han);
%dhmiourgia kai apo8hkeush grafhmatos
ha=figure;
set(ha, 'Visible', 'off');
all=bar_data(net(A),B);
print(ha, '-
djpeg', 'Histogramm_test4_1layer.jpeg')
close(ha);
```



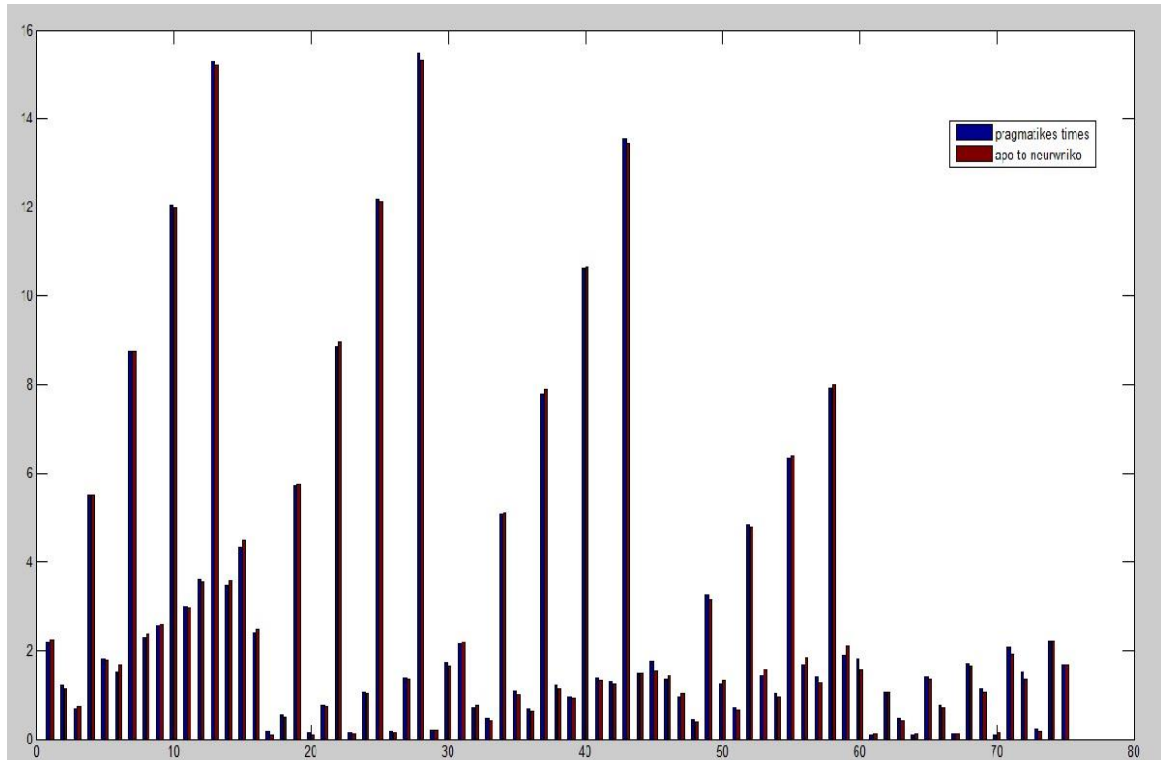
Εικόνα 3.15 : Απόκλιση μεταξύ πραγματικής και εκτιμώμενης τιμής εξόδου για τα συνολικά δεδομένα με τυχαία διαίρεση των δεδομένων.

Τροποποιώντας λίγο τις παραπάνω γραμμές κώδικα και πιο συγκεκριμένα βάζοντας αντί για `all=bar_data (net (A) , B) ;`

το

`all=bar_data (net (strain_for_train') , stress_for_train') ;`

και επιστρέφει το ιστόγραμμα που αφορά τα δεδομένα εκπαίδευσης.



Εικόνα 3.16 : Απόκλιση μεταξύ πραγματικής και εκτιμώμενης τιμής εξόδου για τα δεδομένα που είναι για εκπαίδευση.

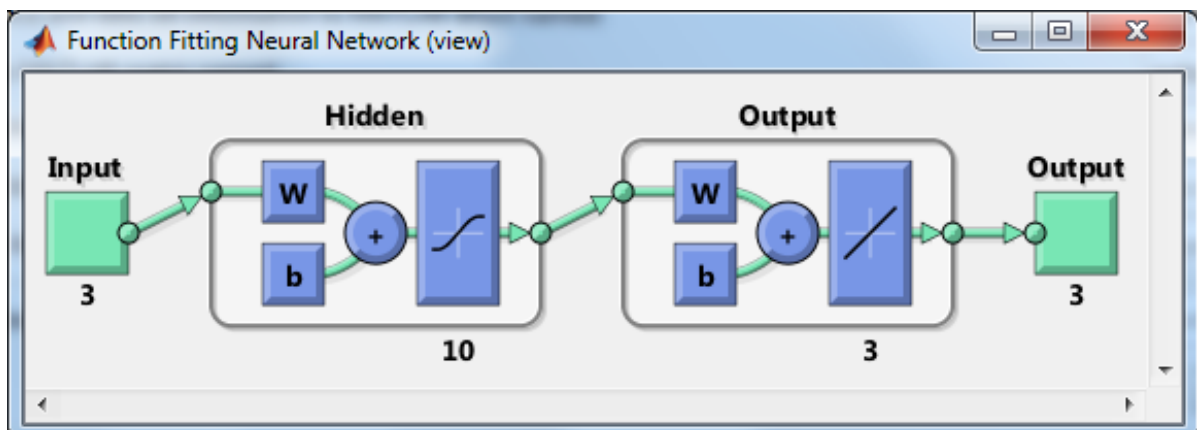
Αν αλλάξουν και πάλι οι ίδιες γραμμές, το

`all=bar_data (net (A) , B) ;`

με

`all=bar_data (net (strain_for_test') , stress_for_test') ;` μας

επιστρέφει το ιστόγραμμα που αφορά τα δεδομένα εκπαίδευσης το οποίο έχει ελάχιστες διαφορές με το προηγούμενο.



Ο παρακάτω συνοπτικός πίνακας περιέχει τα ποσοστά απόκλισης μεταξύ της πραγματικής τιμής και την εκτιμώμενης τιμής για τα συνολικά δεδομένα, των τριών πολιτικών που ακολουθήθηκαν για την εκπαίδευση του δικτύου.

Συνοπτικός πίνακας αποτελεσμάτων των 3 πολιτικών που ακολουθήθηκαν

ΑΡΙΘΜΟΣ ΝΕΥΡΩΝΩΝ ΣΤΟ ΔΙΚΤΥΟ ΣΤΟ ΠΡΩΤΟ ΕΠΙΠΕΔΟ	ΝΔ μετά από 4 εκπαιδεύσεις και με δεδομένα τις καταστάσεις : αρχική, 30 μοίρες και 90 μοίρες με διαίρεση των δεδομένων από τη MatLab (70% για εκπαίδευση 15% για επαλήθευση & 15% για τεστ)	ΝΔ μετά από 4 εκπαιδεύσεις και με δεδομένα από όλες τις καταστάσεις αλλά με χρήση μη τυχαίας διαίρεσης των δεδομένων	ΝΔ μετά από 4 εκπαιδεύσεις και με δεδομένα από όλες τις καταστάσεις με χρήση της τυχαίας διαίρεσης των δεδομένων από τη MatLab (70% για εκπαίδευση 15% για επαλήθευση & 15% για τεστ)
ΠΟΣΟΣΤΟ ΑΠΟΚΛΙΣΗΣ % ΑΠΌ ΤΗΝ ΠΡΑΓΜΑΤΙΚΗ ΤΙΜΗ			
2	94	64976.22	952.72
4	91	83.87	180.19
6			79.83
8		80.71	
10	89		
12	88		68.13
14			
16			
18			
20			

Πίνακας 16 : Συνοπτικός πίνακας αποτελεσμάτων των 3 πολιτικών που ακολουθήθηκαν

Έχοντας ένα πολύ καλό αποτέλεσμα μετά την αλλαγή στο τρόπο δημιουργίας των πινάκων εκπαίδευσης, δηλαδή την επιλογή των δεδομένων από τους αρχικούς πίνακες, προστέθηκε ένα ακόμα κρυφό επίπεδο στο δίκτυο με την προσδοκία να επιστρέψει ακόμα καλύτερα αποτελέσματα.

Για την επίτευξη της παραπάνω αλλαγής χρειάστηκε να προστεθεί μια ακόμα επαναληπτική διαδικασία, στο κυρίως πρόγραμμα την :

```
for
m=elaxistos_arithmos_neurwnwn:vima_auksisis_neurwnwn:megis
tos_arithmos_neurwnwn
```

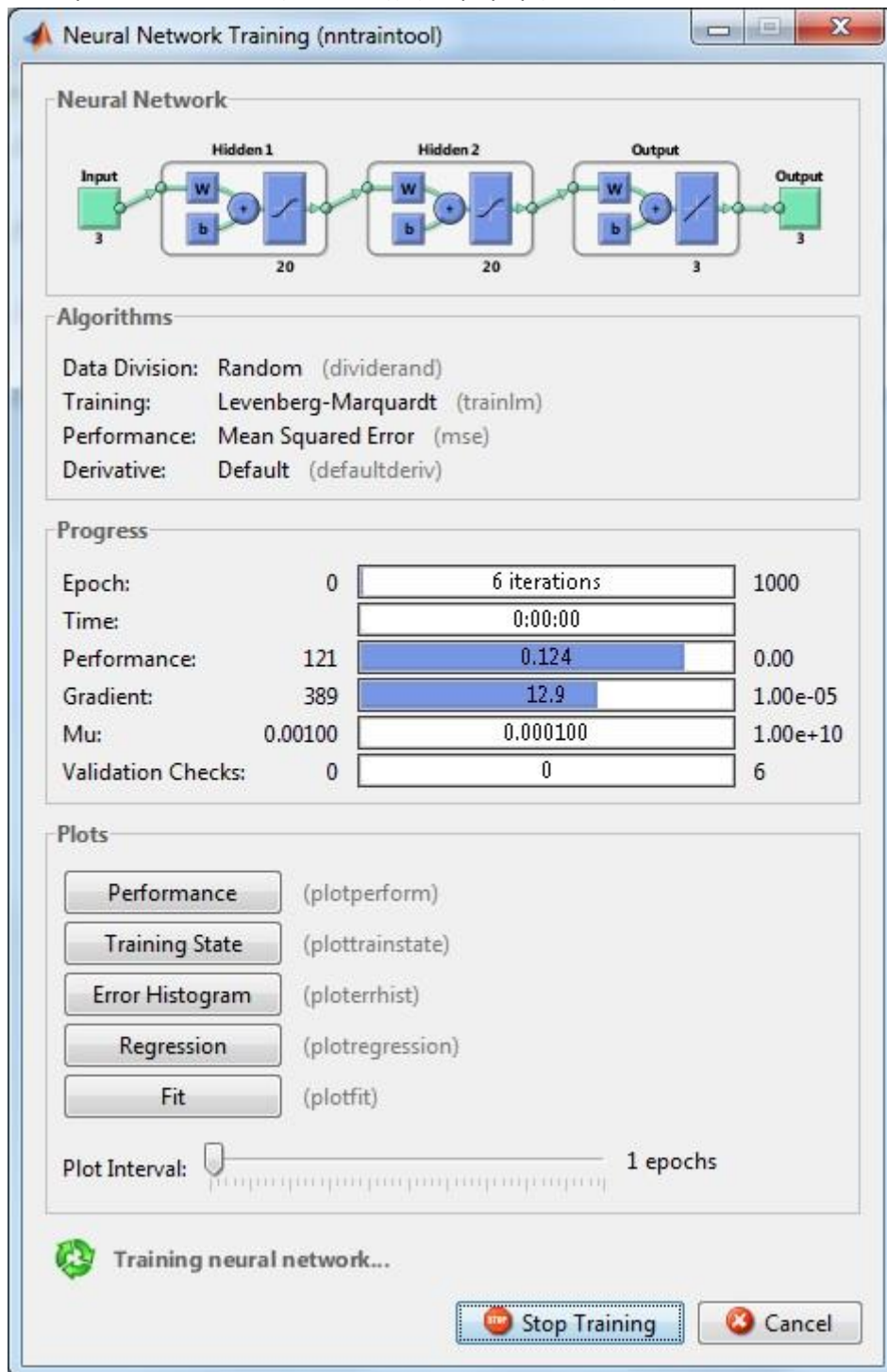
καθώς επίσης να προστεθούν μερικά ακόμα ορίσματα στη συνάρτηση που αφορά το ΝΔ

```
neurwniko_test_4_2layer(strain_for_train',stress_for_train
',m,i);
```

αλλά και μέσα στην συνάρτηση του ΝΔ την οποία καλεί ο κώδικας και η οποία εκπαιδεύει το δίκτυο

```
s=[layers1 layers2];
hiddenLayerSize = s;
net = fitnet(hiddenLayerSize);
```

Παρακάτω εμφανίζεται η εικονική απεικόνιση της αρχιτεκτονικής που επιλέχθηκε με τρεις εισόδους τις παραμορφώσεις στους τρεις άξονες και με δύο κρυφά επίπεδα που αποτελούνται και τα δύο από είκοσι νευρώνες και τέλος τρεις εξόδους οι οποίες είναι οι τάσεις που προκύπτουν από τις αντίστοιχες παραμορφώσεις.



Εικόνα 3.17 : Αρχιτεκτονική του δικτύου με δύο κρυφά επίπεδα 3-20-20-3

Τρέχοντας λοιπόν τον ανανεωμένο κώδικα λήφθηκαν τα παρακάτω αποτελέσματα τα οποία αποδεικνύουν ότι το δεύτερο κρυφό επίπεδο βελτίωσε έστω και λίγο τα αποτελέσματα.

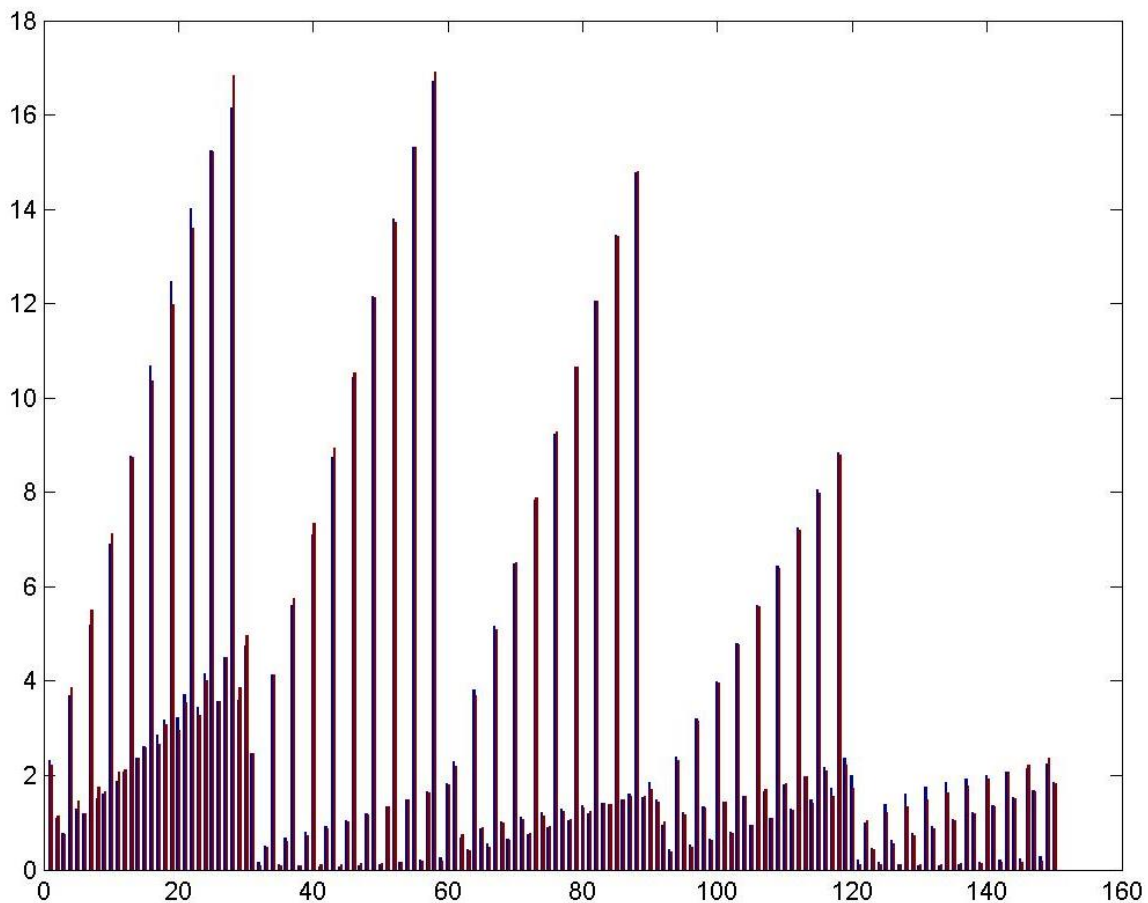
>>Το ποσοστό λαθών είναι: 189 % με αριθμό νευρώνων στο πρώτο κρυφό επίπεδο: 2 και με αριθμό νευρώνων στο δεύτερο κρυφό επίπεδο: 2 μετά από 4 εκπαιδεύσεις

>>Το ποσοστό λαθών είναι: 83.87 % με αριθμό νευρώνων στο πρώτο κρυφό επίπεδο: 2 και με αριθμό νευρώνων στο δεύτερο κρυφό επίπεδο: 4 μετά από 4 εκπαιδεύσεις

>>Το ποσοστό λαθών είναι: 83 % με αριθμό νευρώνων στο πρώτο κρυφό επίπεδο: 4 και με αριθμό νευρώνων στο δεύτερο κρυφό επίπεδο: 4 μετά από 4 εκπαιδεύσεις

>>Το ποσοστό λαθών είναι: 66.20 % με αριθμό νευρώνων στο πρώτο κρυφό επίπεδο: 4 και με αριθμό νευρώνων στο δεύτερο κρυφό επίπεδο: 8 μετά από 4 εκπαιδεύσεις

Διακρίνεται ότι το ποσοστό είναι **66.20 %** σε σχέση με το προηγούμενο που ήταν 68.23 %.



Εικόνα 3.18 : Απόκλιση μεταξύ πραγματικής και εκτιμώμενης τιμής εξόδου με τυχαία διαίρεση δεδομένων με δύο κρυφά επίπεδα.

Παρόλο που το αποτέλεσμα είναι καλύτερο με την προσθήκη του δεύτερου κρυφού επιπέδου, το οποίο επέφερε μια μικρή μείωση του σφάλματος, ότι είναι δυνατόν να επιτευχθούν πολύ καλύτερα αποτελέσματα εάν αυξηθεί κατά μια φορά η εκπαίδευση του ΝΔ και παράλληλα αλλάξει το βήμα αύξησης και ο αριθμός των νευρώνων στα κρυφά επίπεδα, από δύο σε ένα· δηλαδή να αρχίζει το σύστημα με δύο νευρώνες και να αυξάνει κατά έναν σε κάθε επανάληψη μέχρι τους είκοσι.

>>pososto la8wn: 132.52 % me arithmo neurwnwn sto 1 kryfo epipedo: 2 kai me arithmo neurwnwn sto 2 krufo epipedo:2 meta apo 5 ekpaideuseis

>>pososto la8wn: 86.03 % me arithmo neurwnwn sto 1 kryfo epipedo: 2 kai me arithmo neurwnwn sto 2 krufo epipedo:3 meta apo 5 ekpaideuseis

>>pososto la8wn: 81.97 % me arithmo neurwnwn sto 1 kryfo epipedo: 2 kai me arithmo neurwnwn sto 2 krufo epipedo:4 meta apo 5 ekpaideuseis

>>pososto la8wn: 68.80 % me arithmo neurwnwn sto 1 kryfo epipedo: 2 kai me arithmo neurwnwn sto 2 krufo epipedo: 5 meta apo 5 ekpaideuseis

>>pososto la8wn: 57.56 % me arithmo neurwnwn sto 1 kryfo epipedo: 3 kai me arithmo neurwnwn sto 2 krufo epipedo:4 meta apo 5 ekpaideuseis

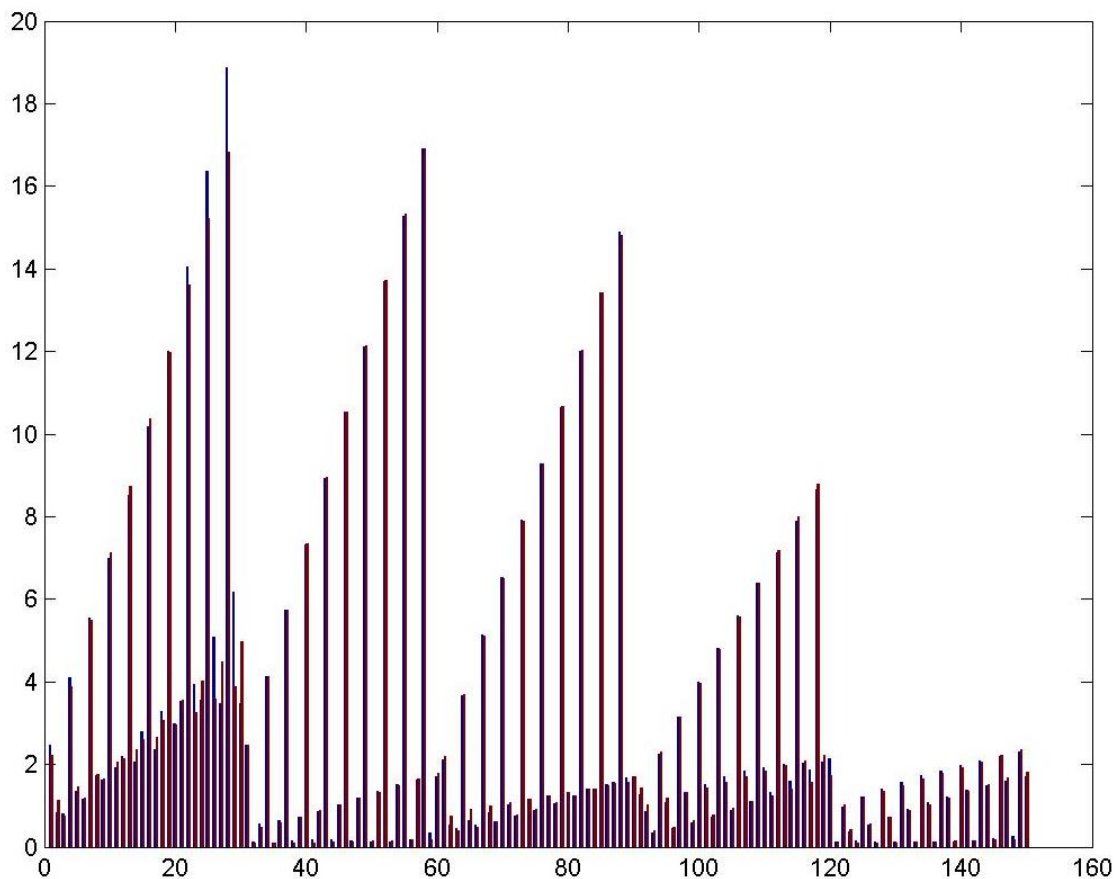
>>pososto la8wn: 55.94 % me arithmo neurwnwn sto 1 kryfo epipedo: 8 kai me arithmo neurwnwn sto 2 krufo epipedo:6 meta apo 5 ekpaideuseis

>>pososto la8wn: 45.37 % me arithmo neurwnwn sto 1 kryfo epipedo: 10 kai me arithmo neurwnwn sto 2 krufo epipedo:8 meta apo 5 ekpaideuseis

Τρέχοντας λοιπόν τον βελτιωμένο κώδικα, λήφθηκαν τα παραπάνω αποτελέσματα τα οποία αποδεικνύουν ότι η αλλαγή αύξησης των φορών εκπαίδευσης του ΝΔ από τέσσερις σε πέντε με παράλληλη αύξηση του βήματος των νευρώνων στο σύστημα επέφερε πολύ καλά αποτελέσματα.

Το ποσοστό είναι **45.37 %** σε σχέση με το προηγούμενο που ήταν 66.20 %.

Το παρακάτω ιστόγραμμα απεικονίζει την απόκλιση μεταξύ της πραγματικής και εκτιμώμενης τιμής εξόδου από το νευρωνικό δίκτυο στο σύνολο των δεδομένων. Το γράφημα προκύπτει χρησιμοποιώντας μια συνάρτηση την οποία καλεί ο κώδικας.



Εικόνα 3.19 : Απόκλιση μεταξύ πραγματικής και εκτιμώμενης τιμής εξόδου με τυχαία διαίρεση δεδομένων με δύο κρυφά επίπεδα.

Στην επόμενη σελίδα, υπάρχει ένας συνοπτικός πίνακας παρόμοιος με τον προηγούμενο, ο οποίος περιέχει τα ποσοστά απόκλισης μεταξύ της πραγματικής τιμής και την εκτιμώμενης τιμής για τα συνολικά δεδομένα, των πολιτικών που ακολουθήθηκαν για την εκπαίδευση του δικτύου.

Συνοπτικός πίνακας αποτελεσμάτων των πολιτικών που ακολουθήθηκαν

ΑΡΙΘΜΟΣ ΝΕΥΡΩΝΩΝ ΣΤΟ ΔΙΚΤΥΟ	ΝΔ μετά από 4 εκπαιδεύσεις, βήμα αύξησης νευρώνων κατά δύο και με δεδομένα από όλες τις καταστάσεις με χρήση της τυχαίας διαίρεσης των δεδομένων από τη MatLab (70% για εκπαίδευση 15% για επαλήθευση & 15% για τεστ) προσθέτοντας ακόμα ένα κρυφό επίπεδο					ΝΔ μετά από 5 εκπαιδεύσεις, βήμα αύξησης νευρώνων κατά ένα και με δεδομένα από όλες τις καταστάσεις με χρήση της τυχαίας διαίρεσης των δεδομένων από τη MatLab (70% για εκπαίδευση 15% για επαλήθευση & 15% για τεστ) προσθέτοντας ακόμα ένα κρυφό επίπεδο				
	ΠΟΣΟΣΤΟ ΑΠΟΚΛΙΣΗΣ % ΑΠΟ ΤΗΝ ΠΡΑΓΜΑΤΙΚΗ ΤΙΜΗ									
1ο επίπεδο/ 2ο επίπεδο	2	3	4	5	6	8	10	12	16	18
2	86	132.52	86.03	81.97	68.80		85			
3			57.56							
4							84.9		84.78	84.20
5										
6								83	70.34	
7										
8					55.94					
9										
10						55.37				

Πίνακας 17 : Συνοπτικός πίνακας αποτελεσμάτων των πολιτικών που ακολουθήθηκαν έχοντας δύο κρυφά επίπεδα.



2^{ος} Όγκος Δεδομένων (Χρησιμοποιώντας τους πίνακες `strain[50x3]` & `consistent[150x3]`)

Στο σημείο αυτό περιγράφεται η διαδικασία που ακολουθήθηκε για την δημιουργία του 2^{ου} νευρωνικού δικτύου. Ειδικότερα, δεν δημιουργήθηκε μόνο ένα δίκτυο αλλά έγινε διερεύνηση για την εύρεση της βέλτιστης αρχιτεκτονικής. Όλα τα δίκτυα που δημιουργήθηκαν όπως και προηγουμένως, για την επίλυση του προβλήματος είναι ΝΔ προστροφοδότησης (feed-forward) με ένα κρυφό στρώμα, τα οποία εκπαιδεύτηκαν με τον Αλγόριθμο Ανάστροφης Διάδοσης (back-propagation).

Αρχικά εξετάστηκε το σενάριο σύμφωνα με το οποίο υπάρχει ένα κρυφό στρώμα και ο αριθμός των νευρώνων στο κρυφό στρώμα αυτό θα μεταβάλλεται από δύο σε είκοσι με βήμα δύο. Το δίκτυο αυτό θα εκπαιδευτεί τέσσερις φορές. Επίσης τα δεδομένα εκπαίδευσης του νευρωνικού δικτύου αποτελούν ένας πίνακας εισόδου `strain_for_train (30 x 3)` και ένας πίνακας εξόδου `consistent_for_train (30 x 9)` οι οποίοι περιέχουν δεδομένα από τους αρχικούς πίνακες `strain (50 x 3)` και `consistent (50 x 9)`. Όσον αφορά τους πίνακες εκπαίδευσης επιλέχθηκαν με τρόπο αυθαίρετο διαλέγοντας τα δεδομένα που αφορούν την αρχική κατάσταση, την κατάσταση στις 30 μοίρες και την κατάσταση στις 90 μοίρες. Η διαφορά που υπάρχει στην δεύτερη περίπτωση δεδομένων είναι ότι τα δεδομένα εξόδου είναι ένας πίνακας διαστάσεων [150 3]. Ο πίνακας αυτός περιέχει 150 δεδομένα, εκ των οποίων κάθε τριάδα πίνακα, δηλαδή ένας [3 3] πίνακας, αντιστοιχεί σε μια γραμμή του πίνακα εισόδου. Για την καλύτερη εισαγωγή δεδομένων στο νευρωνικό δίκτυο τροποποιήθηκε αυτός ο πίνακας εξόδου και από [150 3] έγινε [50 9]. Αυτό επιτεύχθηκε με τις παρακάτω γραμμές κώδικα:

```
B=consistent;
p=1;
for i=1:3:150
    x=B(i,:);
    y=B(i+1,:);
    z=B(i+2,:);
    B1(p,:)=[x y z];
    p=p+1;
end
```

Ο πίνακας εισόδου για την εκπαίδευση είναι ο `strain_for_train` και της εξόδου ο `consistent_for_train` οι οποίοι δημιουργήθηκαν με τις παρακάτω εντολές και σύμφωνα με την πολιτική που περιγράφηκε παραπάνω:

```
y=10;
f=1;
k=1;
while k<=50
    if k>=1 && k<11 | k>20 && k<31 | k>40 && k<46
        for l=f:y
            for o=1:3
                strain_for_train(l,o)=strain(k,o);
            end
            k=k+1;
        end
    end
end
```

```

        end
        f=y+1;
        y=y+10;
    end
    k=k+1;
end
r=10;
t=1;
g=1;
while g<=50
    if g>=1 && g<11 | g>20 && g<31 | g>40 && g<46
        for l=t:r
            for o=1:9
                consistent_for_train(l,o)=B1(g,o);
            end
            g=g+1;
        end
        t=r+1;
        r=r+10;
    end
    g=g+1;
end
end

```

Ο τρόπος που έχει δομηθεί ο κώδικας είναι τέτοιος ώστε να επιστρέφει και να εμφανίζει στο command window της MatLab το βέλτιστο μέγιστο ποσοστό απόκλισης επί τοις εκατό από την πραγματική τιμή, ενημερώνοντάς παράλληλα τους χρήστες με πόσους νευρώνες επιτεύχθηκε αυτό και μετά από πόσες εκπαιδεύσεις. Εν προκειμένω, θα πραγματοποιηθούν τέσσερις εκπαιδεύσεις του δικτύου, ο αριθμός των νευρών ξεκινά με δύο και καταλήγει σε 30 με βήμα ένα.

```

fores_ekpaideusis=4; %fores pou 8a ekpaideutei to diktuo
%arismos neurwnwn
elaxistos_arithmos_neurwnwn=2;
vima_auksisis_neurwnwn=1;
megistos_arithmos_neurwnwn=30;

```

Τα αποτελέσματα που επέστρεψε το πρόγραμμα είναι μη φυσιολογικά.

```

>>To pososto la8wn einai: 2100 % me arithmo neyrwnwn:2 meta
apo 4 ekpaideuseis
>>To pososto la8wn einai: 698 % me arithmo neyrwnwn:3 meta
apo 4 ekpaideuseis
>>To pososto la8wn einai: 432 % me arithmo neyrwnwn:25 meta
apo 4 ekpaideuseis

```

Και αυτό ήταν αναμενόμενο αφενός μεν γιατί η επιλογή των δεδομένων εισόδου και εξόδου έγινε αντίστοιχα με αυθαίρετη επιλογή μερικών καταστάσεων και αφετέρου ο πίνακας εξόδου των δεδομένων λόγω της δομής του. Έτσι τροποποιήθηκε ο κώδικας και επιλέχθηκε μια διαφορετική αρχιτεκτονική. Αρχικά, θα διορθωθούν όσοι παράγοντες συνέβαλαν στα

παραπάνω μη ικανοποιητικά αποτελέσματα. Ακολουθούν οι πίνακες δεδομένων εισόδου και εξόδου οι οποίοι διαμορφώνονται στην παρακάτω κατάσταση :

```
k=1;
for l=1:2:50
    for o=1:3
        strain_for_train(k,o)=strain(l,o);
    end
    k=k+1;
end
k=1;
for l=1:2:50
    for o=1:9
        consistent_for_train(k,o)=B1(l,o);
    end
    k=k+1;
end
```

αυξάνοντας τις φορές εκπαίδευσης από τέσσερις σε πέντε και αλλάζοντας τον αριθμό των νευρώνων

```
fores_ekpaideusis=5; %fores που θα εκπαιδευτεί το δίκτυο
elaxistos_arithmos_neurwnwn=10;
vima_auksisis_neurwnwn=5;
megistos_arithmos_neurwnwn=20;
```

προσθέτοντας ένα ακόμα κρυφό επίπεδο στο δίκτυο :

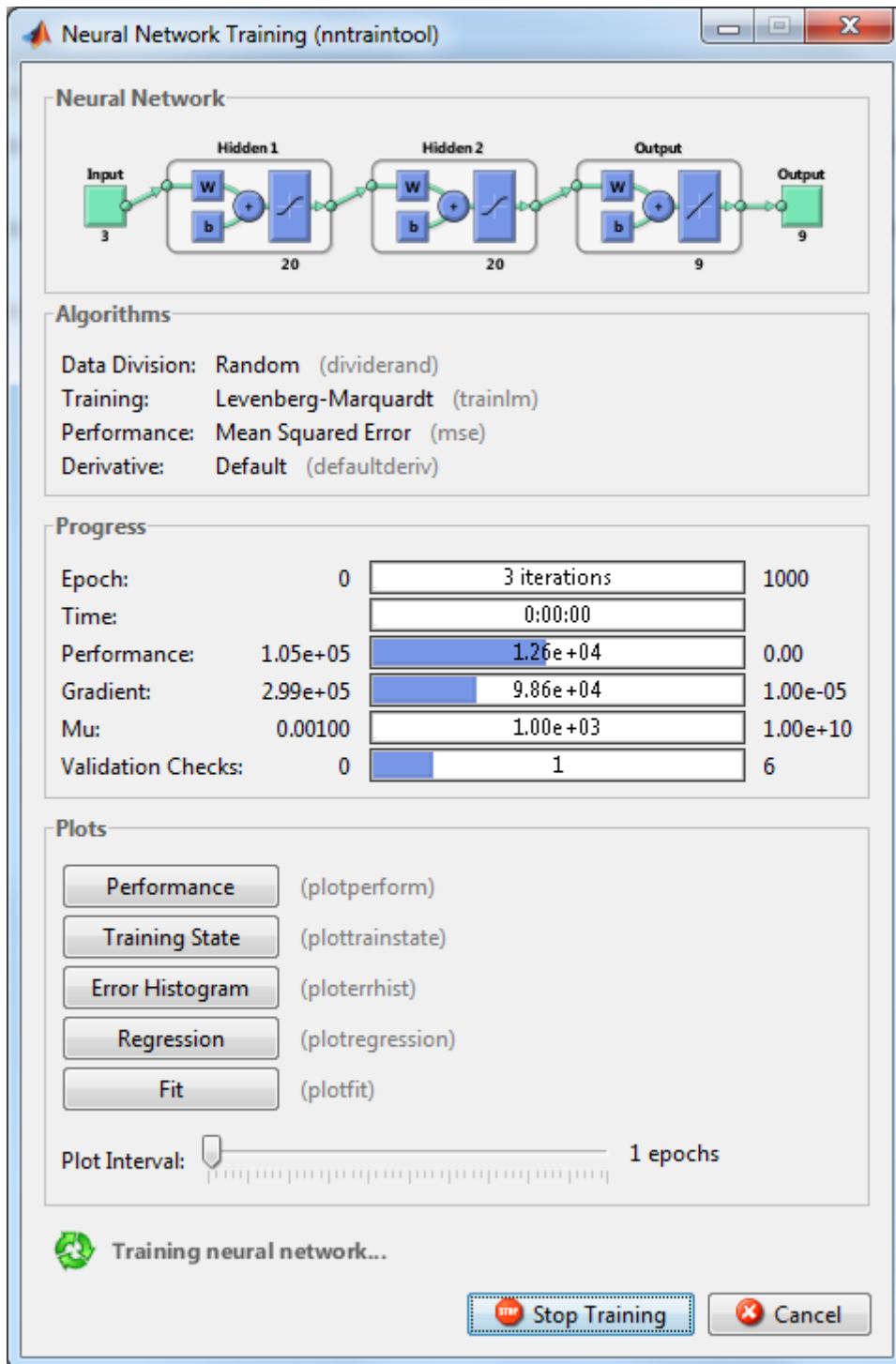
```
for
m=elaxistos_arithmos_neurwnwn:vima_auksisis_neurwnwn:megistos_arithmos_neurwnwn
:
:
end
```

Έτσι η αρχιτεκτονική που ακολουθείται είναι 3-20-20-9 και τα αποτελέσματα που εξήχθησαν είναι τα παρακάτω:

```
>>To pososto la8wn einai: 10069 % me aritmo neurwnwn sto
prwto kryfo epipedo: 4 kai arithmo neurwnwn sto deutero
kryfo epipedo: 10 meta apo 5 ekpaideuseis
>>To pososto la8wn einai: 9531 % me aritmo neurwnwn sto
prwto kryfo epipedo: 8 kai arithmo neurwnwn sto deutero
kryfo epipedo: 15 meta apo 5 ekpaideuseis
>>To pososto la8wn einai: 2867 % me aritmo neurwnwn sto
prwto kryfo epipedo: 10 kai arithmo neurwnwn sto deutero
kryfo epipedo: 10 meta apo 5 ekpaideuseis
>>To pososto la8wn einai: 2669 % me aritmo neurwnwn sto
prwto kryfo epipedo: 15 kai arithmo neurwnwn sto deutero
kryfo epipedo: 10 meta apo 5 ekpaideuseis
```

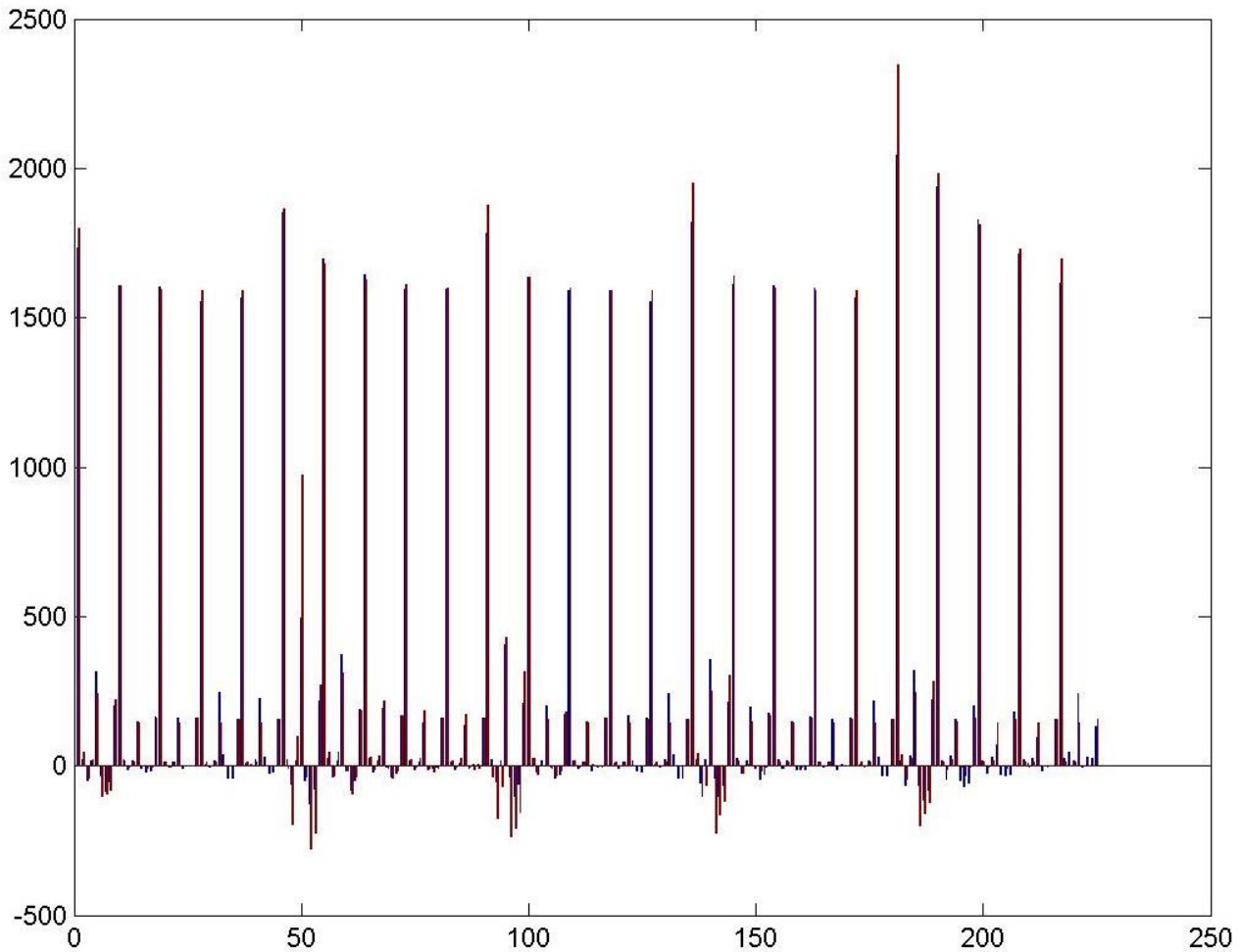
>>Το ποσοστό λάθων είναι: 399 % με αριθμό νευρώνων στο πρώτο κρυφό επίπεδο: 10 και αριθμό νευρώνων στο δεύτερο κρυφό επίπεδο: 20 μετά από 5 εκπαιδεύσεις

Η παρακάτω εικόνα παρουσιάζει την αρχιτεκτονική που έχει επιλεγεί με τρεις εισόδους τις παραμορφώσεις στους τρεις άξονες και με δύο κρυφά επίπεδα τα οποία αποτελούνται και τα δύο από είκοσι νευρώνες και τέλος εννιά εξόδους, που είναι οι τάσεις που προκύπτουν από τις αντίστοιχες παραμορφώσεις.



Εικόνα 3.20: Αρχιτεκτονική του δικτύου με δύο κρυφά επίπεδα 3-20-20-9

Το παρακάτω ιστόγραμμα το οποίο απεικονίζει την απόκλιση που υπάρχει μεταξύ της πραγματικής και εκτιμώμενης τιμής εξόδου από το νευρωνικό δίκτυο στο σύνολο των δεδομένων. Το γράφημα προκύπτει χρησιμοποιώντας μια συνάρτηση την οποία καλούμε μέσα στο κώδικα μας.



Εικόνα 3.21: Απόκλιση μεταξύ πραγματικής και εκτιμώμενης τιμής εξόδου για το σύνολο των δεδομένων με τυχαία διαίρεση δεδομένων και δύο κρυφά επίπεδα.

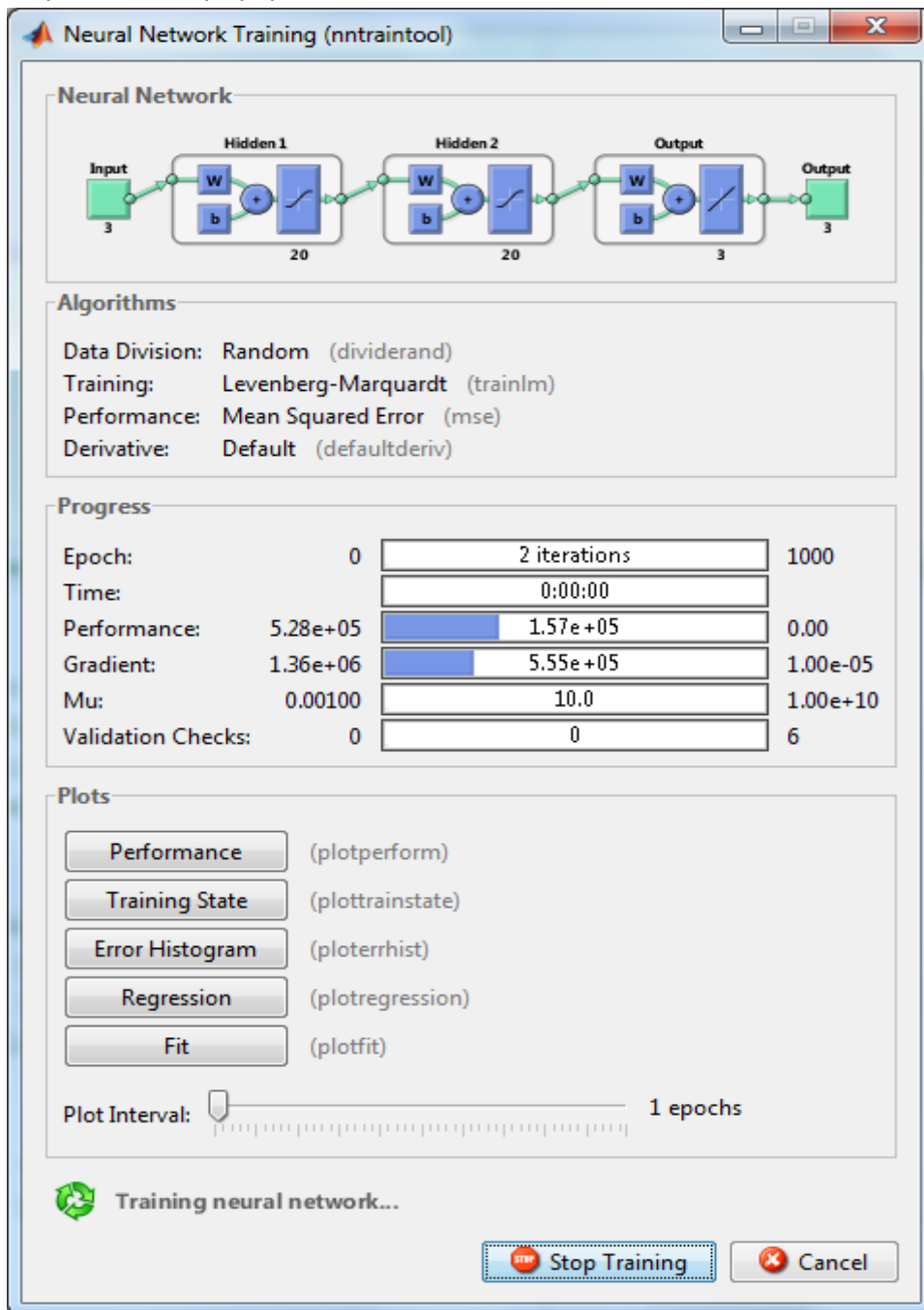
Αν και τα αποτελέσματα είναι καλύτερα με την προσθήκη του δεύτερου κρυφού επιπέδου, το οποίο επέφερε μια μικρή μείωση του σφάλματος, παραμένουν ωστόσο όχι τόσο ρεαλιστικά και λογικά. Προκειμένου να διορθωθεί αυτό και να επιτευχθούν ικανοποιητικότερα αποτελέσματα, κρίνεται αναγκαία η αύξηση κατά μια φορά της εκπαίδευσης του ΝΔ και ταυτόχρονα της αρχιτεκτονικής. Η αλλαγή στην αρχιτεκτονική του δικτύου σημαίνει ότι από το ήδη αποτελούμενο από τρεις εισόδους με δύο κρυφά επίπεδα έχοντας το κάθε ένα από είκοσι νευρώνες και εννέα εξόδους (3-20-20-9) δίκτυο θα μετατραπεί σε δίκτυο με τρεις εισόδους με δύο κρυφά επίπεδα αποτελούμενα το κάθε ένα από είκοσι νευρώνες αλλά έχοντας τρεις εξόδους (3-20-20-3) παίρνοντας για πίνακα εξόδου την διαγώνιο. Με τον όρο διαγώνιο ουσιαστικά σημαίνει την πρώτη στήλη, την πέμπτη και την ένατη στήλη από το πίνακα [50 9]. Ο νέος πίνακας προκύπτει απλά :

```
k=1;
n=1;
for l=1:2:50
```

```

for o=1:4:9
    consistent_for_train(k,n)=B1(1,o);
    n=n+1;
end
k=k+1;
n=1;
end
    
```

ο αριθμός των εκπαιδεύσεων θα παραμείνει ο ίδιος (5) και θα αλλάξει πάλι τον αριθμό των νευρώνων σε κάθε κρυφό επίπεδο. Εκκινώντας με πέντε νευρώνες, αυτοί θα αυξάνονται μέχρι να γίνουν είκοσι με βήμα πέντε.



Εικόνα 3.22: Αρχιτεκτονική του δικτύου με δύο κρυφά επίπεδα 3-20-20-3

Τα αποτελέσματα από την παραπάνω διερεύνηση είναι τα ακόλουθα:

>>Το ποσοστό λαθών είναι: 165.60 % με αριθμό νευρώνων στο πρωτό κρυφο επίπεδο: 10 και με αριθμό νευρώνων στο δευτερο κρυφο επίπεδο: 10 μετά από 5 εκπαιδεύσεις

>>Το ποσοστό λαθών είναι: 128.43 % με αριθμό νευρώνων στο πρωτό κρυφο επίπεδο: 10 και με αριθμό νευρώνων στο δευτερο κρυφο επίπεδο: 15 μετά από 5 εκπαιδεύσεις

>>Το ποσοστό λαθών είναι: 122.45 % με αριθμό νευρώνων στο πρωτό κρυφο επίπεδο: 10 και με αριθμό νευρώνων στο δευτερο κρυφο επίπεδο: 20 μετά από 5 εκπαιδεύσεις

>>Το ποσοστό λαθών είναι: 71.85 % με αριθμό νευρώνων στο πρωτό κρυφο επίπεδο: 15 και με αριθμό νευρώνων στο δευτερο κρυφο επίπεδο: 10 μετά από 5 εκπαιδεύσεις

>>Το ποσοστό λαθών είναι: 26.96 % με αριθμό νευρώνων στο πρωτό κρυφο επίπεδο: 15 και με αριθμό νευρώνων στο δευτερο κρυφο επίπεδο: 15 μετά από 5 εκπαιδεύσεις

Παρόλο που τα αποτελέσματα είναι πολύ καλύτερα με την αλλαγή της αρχιτεκτονικής του δικτύου αλλά και της αλλαγής των πινάκων εκπαίδευσης του δικτύου, τα οποία επέφεραν μια μεγάλη μείωση του σφάλματος, είναι δυνατόν να επιτευχθούν ακόμη καλύτερα αποτελέσματα εάν εκπαιδευτούν τρία διαφορετικά νευρωνικά δίκτυα. Ειδικότερα, το κάθε δίκτυο θα είναι της αρχιτεκτονικής 3-20-20-1 το οποίο πρακτικά σημαίνει ότι θα λαμβάνονται πολύ καλές απαντήσεις όσον αφορά την τάση σε κάθε έναν άξονα χωριστά. Προς υλοποίηση αυτού η μόνη τροποποίηση που θα χρειαστεί ο κώδικας είναι η ακόλουθη :

```
k=1;
n=1;
for l=1:2:50
    for o=1
        consistent_for_train(k,n)=B1(l,o);
        n=n+1;
    end
    k=k+1;
    n=1;
end
```

```
n=1;
for l=1:50
    for o=1
        B2(l,n)=B1(l,o);
        n=n+1;
    end
    n=1;
end
```

Δηλαδή χρησιμοποιείται ένας πίνακας consistent_for_train[25 1] και ένας B2[50 1] όπου ο αριθμός 1 υποδηλώνει ποια στήλη του συνολικού πίνακα εξόδου

επιλέγεται κάθε φορά. Εν προκειμένω λαμβάνεται η πρώτη στήλη, η πέμπτη και η ένατη στήλη αντίστοιχα.

The screenshot shows the 'Neural Network Training (nntool)' window. At the top, the 'Neural Network' diagram illustrates a feedforward network with 3 input nodes, two hidden layers of 20 nodes each, and 1 output node. Below the diagram, the 'Algorithms' section lists: Data Division: Random (dividerand), Training: Levenberg-Marquardt (trainlm), Performance: Mean Squared Error (mse), and Derivative: Default (defaultderiv). The 'Progress' section displays a table of training metrics:

Epoch:	0	4 iterations	1000
Time:		0:00:00	
Performance:	3.24e+06	1.04e+05	0.00
Gradient:	8.79e+06	6.12e+05	1.00e-05
Mu:	0.00100	100	1.00e+10
Validation Checks:	0	3	6

The 'Plots' section includes buttons for Performance (plotperform), Training State (plottrainstate), Error Histogram (ploterrhist), Regression (plotregression), and Fit (plotfit). A 'Plot Interval' slider is set to 1 epochs. At the bottom, a status bar shows 'Training neural network...' with a 'Stop Training' button and a 'Cancel' button.

Εικόνα 3.23 : Αρχιτεκτονική του δικτύου με δύο κρυφά επίπεδα 3-20-20-1

Τα αποτελέσματα που θα ληφθούν για την πρώτη στήλη ακολουθώντας αυτή την πολιτική είναι :

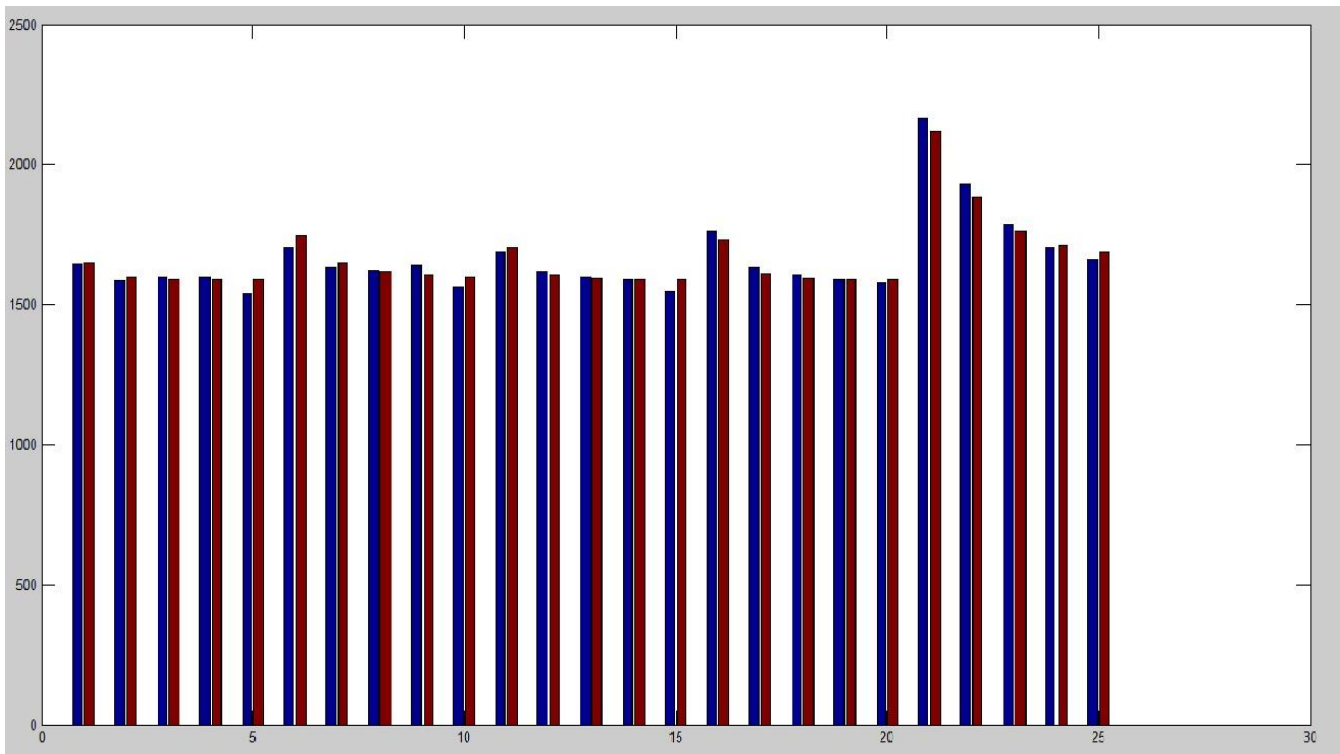
>>Το ποσοστό λαθών είναι: 17.43 % με αριθμό neurwnwn στο prwto krufo epipedo: 5 και με αριθμό neurwnwn στο deutero krufo epipedo: 5 meta apo 5 ekpaideuseis

>>Το ποσοστό λαθών είναι: 12.67 % με αριθμό neurwnwn στο prwto krufo epipedo: 5 και με αριθμό neurwnwn στο deutero krufo epipedo: 10 meta apo 5 ekpaideuseis

>>Το ποσοστό λαθών είναι: 4.44 % με αριθμό neurwnwn στο prwto krufo epipedo: 5 και με αριθμό neurwnwn στο deutero krufo epipedo: 15 meta apo 5 ekpaideuseis

>>Το ποσοστό λαθών είναι: 3.15 % με αριθμό neurwnwn στο prwto krufo epipedo: 15 και με αριθμό neurwnwn στο deutero krufo epipedo: 5 meta apo 5 ekpaideuseis

>>Το ποσοστό λαθών είναι: **2.26 %** με αριθμό neurwnwn στο prwto krufo epipedo: 15 και με αριθμό neurwnwn στο deutero krufo epipedo: 20 meta apo 5 ekpaideuseis



Εικόνα 3.24: Απόκλιση μεταξύ πραγματικής και εκτιμώμενης τιμής εξόδου για το σύνολο των δεδομένων με τυχαία διαίρεση δεδομένων και δύο κρυφά επίπεδα.

Τα αποτελέσματα τα οποία θα ληφθούν για την πέμπτη στήλη ακολουθώντας αυτή την πολιτική είναι:

>>Το ποσοστό λαθών είναι: 99.4 % με αριθμό νευρώνων στο πρωτό κρυφό επίπεδο: 10 και με αριθμό νευρώνων στο δευτερο κρυφό επίπεδο: 10 μετά από 5 εκπαιδεύσεις

>>Το ποσοστό λαθών είναι: **36 %** με αριθμό νευρώνων στο πρωτό κρυφό επίπεδο: 10 και με αριθμό νευρώνων στο δευτερο κρυφό επίπεδο: 15 μετά από 5 εκπαιδεύσεις

Όπως προκύπτει από τα παραπάνω αποτελέσματα αυτή η πολιτική δεν οδηγεί σε πάρα πολύ καλά αποτελέσματα όσον αφορά την τάση στο γ-άξονα. Γι' αυτό θα πρέπει να γίνει μια περαιτέρω διερεύνηση. Η επόμενη αλλαγή είναι το βήμα αύξησης των νευρώνων. Πριν υπήρχαν δύο κρυφά επίπεδα τα οποία ξεκινούσαν από πέντε νευρώνες και έφταναν τους είκοσι με βήμα πέντε. Τώρα οι νευρώνες θα ξεκινούν από τέσσερεις και να φτάνουν πάλι μέχρι είκοσι με βήμα δύο. Τα αποτελέσματα παρατίθενται παρακάτω :

>>Το ποσοστό λαθών είναι: 239.35 % με αριθμό νευρώνων στο πρωτό κρυφό επίπεδο: 4 και με αριθμό νευρώνων στο δευτερο κρυφό επίπεδο: 4 μετά από 5 εκπαιδεύσεις

>>Το ποσοστό λαθών είναι: 110.74 % με αριθμό νευρώνων στο πρωτό κρυφό επίπεδο: 4 και με αριθμό νευρώνων στο δευτερο κρυφό επίπεδο: 6 μετά από 5 εκπαιδεύσεις

>>Το ποσοστό λαθών είναι: 109.36 % με αριθμό νευρώνων στο πρωτό κρυφό επίπεδο: 4 και με αριθμό νευρώνων στο δευτερο κρυφό επίπεδο: 8 μετά από 5 εκπαιδεύσεις

>>Το ποσοστό λαθών είναι: 79.84 % με αριθμό νευρώνων στο πρωτό κρυφό επίπεδο: 4 και με αριθμό νευρώνων στο δευτερο κρυφό επίπεδο: 10 μετά από 5 εκπαιδεύσεις

>>Το ποσοστό λαθών είναι: 54.90 % με αριθμό νευρώνων στο πρωτό κρυφό επίπεδο: 4 και με αριθμό νευρώνων στο δευτερο κρυφό επίπεδο: 12 μετά από 5 εκπαιδεύσεις

>>Το ποσοστό λαθών είναι: 53.71 % με αριθμό νευρώνων στο πρωτό κρυφό επίπεδο: 6 και με αριθμό νευρώνων στο δευτερο κρυφό επίπεδο: 8 μετά από 5 εκπαιδεύσεις

>>Το ποσοστό λαθών είναι: 52.75 % με αριθμό νευρώνων στο πρωτό κρουφο επίπεδο: 6 και με αριθμό νευρώνων στο δευτερο κρουφο επίπεδο: 10 μετά από 5 εκπαιδεύσεις

>>Το ποσοστό λαθών είναι: 47.14 % με αριθμό νευρώνων στο πρωτό κρουφο επίπεδο: 8 και με αριθμό νευρώνων στο δευτερο κρουφο επίπεδο: 10 μετά από 5 εκπαιδεύσεις

>>Το ποσοστό λαθών είναι: 40.51 % με αριθμό νευρώνων στο πρωτό κρουφο επίπεδο: 10 και με αριθμό νευρώνων στο δευτερο κρουφο επίπεδο: 4 μετά από 5 εκπαιδεύσεις

>>Το ποσοστό λαθών είναι: **29.33 %** με αριθμό νευρώνων στο πρωτό κρουφο επίπεδο: 10 και με αριθμό νευρώνων στο δευτερο κρουφο επίπεδο: 12 μετά από 5 εκπαιδεύσεις

Η αλλαγή στο βήμα αύξησης των νευρώνων είχε καλύτερο αποτέλεσμα. Με την πολιτική αυτή λήφθηκε σαν βέλτιστο αποτέλεσμα 29.33% ενώ με την προηγούμενη 36%. Θα δοκιμασθεί μια ακόμη πολιτική προς βελτίωση των αποτελεσμάτων. Θα μειωθεί και πάλι κατά μια μονάδα το βήμα αύξησης των νευρώνων από δύο σε ένα. Τα νέα αποτελέσματα είναι τα παρακάτω:

>>Το ποσοστό λαθών είναι: 169.31 % με αριθμό νευρώνων στο πρωτό κρουφο επίπεδο: 5 και με αριθμό νευρώνων στο δευτερο κρουφο επίπεδο: 5 μετά από 5 εκπαιδεύσεις

>>Το ποσοστό λαθών είναι: 43.58 % με αριθμό νευρώνων στο πρωτό κρουφο επίπεδο: 5 και με αριθμό νευρώνων στο δευτερο κρουφο επίπεδο: 6 μετά από 5 εκπαιδεύσεις

>>Το ποσοστό λαθών είναι: 32.41 % με αριθμό νευρώνων στο πρωτό κρουφο επίπεδο: 5 και με αριθμό νευρώνων στο δευτερο κρουφο επίπεδο: 10 μετά από 5 εκπαιδεύσεις

>>Το ποσοστό λαθών είναι: 30.36 % με αριθμό νευρώνων στο πρωτό κρουφο επίπεδο: 6 και με αριθμό νευρώνων στο δευτερο κρουφο επίπεδο: 5 μετά από 5 εκπαιδεύσεις

>>Το ποσοστό λαθών είναι: **28.52 %** με αριθμό νευρώνων στο πρωτό κρουφο επίπεδο: 7 και με αριθμό νευρώνων στο δευτερο κρουφο επίπεδο: 7 μετά από 5 εκπαιδεύσεις

Το νέο αποτέλεσμα είχε ελάχιστη διαφορά με το προηγούμενο οπότε φαίνεται ανώφελο να υιοθετηθεί η τελευταία πολιτική, γιατί ο χρόνος που απαιτείται είναι περισσότερος για να εξάγει τα αποτελέσματα σε σχέση με την προηγούμενη.

Τα αποτελέσματα τα οποία θα λάβουμε για την ένατη στήλη ακολουθώντας την πρώτη πολιτική, δηλαδή δύο κρυφά επίπεδα με είκοσι νευρώνες το κάθε ένα ξεκινώντας από πέντε και αυξάνοντας με πέντε, είναι :

>>Το ποσοστό λάθων είναι: 24.3% με αριθμό νευρώνων στο πρώτο κρυφό επίπεδο: 10 και με αριθμό νευρώνων στο δεύτερο κρυφό επίπεδο: 10 μετά από 5 εκπαιδεύσεις

>>Το ποσοστό λάθων είναι: 22% με αριθμό νευρώνων στο πρώτο κρυφό επίπεδο: 10 και με αριθμό νευρώνων στο δεύτερο κρυφό επίπεδο: 20 μετά από 5 εκπαιδεύσεις

>>Το ποσοστό λάθων είναι: **19%** με αριθμό νευρώνων στο πρώτο κρυφό επίπεδο: 15 και με αριθμό νευρώνων στο δεύτερο κρυφό επίπεδο: 10 μετά από 5 εκπαιδεύσεις

Όπως συνάγεται από τα παραπάνω αποτελέσματα αυτή η πολιτική δεν εξάγει ιδιαίτερα ικανοποιητικά αποτελέσματα όσον αφορά την τάση στον z-άξονα. Γι' αυτό θα πρέπει να γίνει μια περαιτέρω διερεύνηση όπως και στον γ-άξονα. Η επόμενη αλλαγή είναι το βήμα αύξησης των νευρώνων. Προηγουμένως γινόταν χρήση δύο κρυφών επιπέδων τα οποία ξεκινούσαν από πέντε νευρώνες και έφταναν τους είκοσι με βήμα πέντε, ενώ τώρα θα ξεκινούν από τέσσερεις και θα φτάνουν πάλι μέχρι είκοσι με βήμα δύο. Τα αποτελέσματα από αυτήν την αλλαγή είναι τα εξής :

>>Το ποσοστό λάθων είναι: 25.03 % με αριθμό νευρώνων στο πρώτο κρυφό επίπεδο: 4 και με αριθμό νευρώνων στο δεύτερο κρυφό επίπεδο: 4 μετά από 5 εκπαιδεύσεις

>>Το ποσοστό λάθων είναι: 17.16 % με αριθμό νευρώνων στο πρώτο κρυφό επίπεδο: 4 και με αριθμό νευρώνων στο δεύτερο κρυφό επίπεδο: 8 μετά από 5 εκπαιδεύσεις

>>Το ποσοστό λάθων είναι: 17.08 % με αριθμό νευρώνων στο πρώτο κρυφό επίπεδο: 4 και με αριθμό νευρώνων στο δεύτερο κρυφό επίπεδο: 10 μετά από 5 εκπαιδεύσεις

>>Το ποσοστό λάθων είναι: **13.41 %** με αριθμό νευρώνων στο πρώτο κρυφό επίπεδο: 4 και με αριθμό νευρώνων στο δεύτερο κρυφό επίπεδο: 12 μετά από 5 εκπαιδεύσεις

Η αλλαγή στο βήμα αύξησης των νευρώνων έδωσε ένα πολύ καλύτερο αποτέλεσμα. Με την πολιτική αυτή λήφθηκε σαν βέλτιστο αποτέλεσμα 13.41 % ενώ με την προηγούμενη 19 %. Θα γίνει δοκιμή μιας ακόμη πολιτικής προκειμένου να διερευνηθεί η δυνατότητα περαιτέρω βελτίωσης. Για το λόγο αυτό θα μειωθεί κατά μια μονάδα το βήμα αύξησης των νευρώνων από δύο σε ένα. Τα νέα αποτελέσματα είναι τα παρακάτω:

>>Το ποσοστό λάθων είναι: 18.54 % με αριθμό νευρώνων στο πρώτο κρουφίο επίπεδο: 4 και με αριθμό νευρώνων στο δεύτερο κρουφίο επίπεδο: 4 μετά από 5 εκπαιδευτές

>>Το ποσοστό λάθων είναι: 17.89 % με αριθμό νευρώνων στο πρώτο κρουφίο επίπεδο: 4 και με αριθμό νευρώνων στο δεύτερο κρουφίο επίπεδο: 13 μετά από 5 εκπαιδευτές

>>Το ποσοστό λάθων είναι: 11.33 % με αριθμό νευρώνων στο πρώτο κρουφίο επίπεδο: 4 και με αριθμό νευρώνων στο δεύτερο κρουφίο επίπεδο: 15 μετά από 5 εκπαιδευτές

>>Το ποσοστό λάθων είναι: **10.95 %** με αριθμό νευρώνων στο πρώτο κρουφίο επίπεδο: 16 και με αριθμό νευρώνων στο δεύτερο κρουφίο επίπεδο: 5 μετά από 5 εκπαιδευτές

ΚΕΦΑΛΑΙΟ 4

Συμπεράσματα

Απ' όσα εξετάσθηκαν παραπάνω προκύπτει το συμπέρασμα πως η πρόβλεψη των τάσεων παραμόρφωσης ενός κελύφους περιοδικότητας στους τρεις άξονες αναφοράς έχοντας τις τροπές στους αντίστοιχους άξονες, με τη χρήση νευρωνικών δικτύων είναι εφικτή και τα εξαγόμενα αποτελέσματα αρκετά ικανοποιητικά. Απαιτείται η προσοχή του χρήστη στην επιλογή της κατάλληλης αρχιτεκτονικής του δικτύου, των κατάλληλων παραμέτρων που θα χρησιμοποιηθούν καθώς οι παράγοντες αυτοί επηρεάζουν σημαντικά το αποτέλεσμα. Αναλυτικότερα, είναι ανάγκη να γίνει διερεύνηση τόσο για την επιλογή του κατάλληλου μεγέθους δικτύου, δηλαδή του αριθμού των νευρώνων και τον αριθμό των κρυφών επιπέδων που θα χρειαστούν, όσο και την επιλογή των δεδομένων που θα χρησιμοποιηθούν για την εκπαίδευση του ΝΔ. Η βιβλιογραφία και η θεωρία που υπάρχει για τα νευρωνικά δίκτυα τονίζει ότι δεν υπάρχει ένας γενικός κανόνας που να καθορίζει αυτές τις παραμέτρους. Η λύση που προτείνεται από τους μελετητές αλλά και από την παρούσα εργασία είναι η διερεύνηση, δηλαδή μόνο έπειτα από δοκιμές είναι δυνατόν να εξαχθούν ασφαλή συμπεράσματα.

Από τις δοκιμές που πραγματοποιήθηκαν, γίνεται αντιληπτό πως ένα ισχυρό δίκτυο (ένα τέτοιο δίκτυο απαιτεί την χρήση περισσότερων νευρώνων) δεν είναι πάντοτε καλύτερο, καθώς ενώ είναι αποτελεσματικό όσον αφορά τα δεδομένα εκπαίδευσης, δεν είναι σε θέση να γενικεύσει στον ίδιο βαθμό. Κάτι τέτοιο εξαρτάται άμεσα από τον αριθμό των δεδομένων εκπαίδευσης. Ενώ χρειάζεται επαρκής αριθμός δεδομένων για την σωστή εκπαίδευση του δικτύου, η περαιτέρω αύξηση δεν προσφέρει κάτι περισσότερο στην βελτίωση των αποτελεσμάτων ενώ αυξάνει τον υπολογιστικό φόρτο κατά τη διαδικασία της εκπαίδευσης.

ΚΕΦΑΛΑΙΟ 5

Βιβλιογραφία

1. M. Lefik a,* , D.P. Boso b, B.A. Schrefler b, « Artificial Neural Networks in numerical modelling of composites », Comput. Methods Appl. Mech. Engrg. 198 (2009) 1785–1804.
2. Marek Lefik, « Hybrid, finite element-artificial neural network model for composite materials», Journal of Theoretical and Applied Mechanics 42, 3, pp. 539-563, Warsaw 2004.
3. Wikipedia, θεωρία των νευρωνικών δικτύων.
4. NEURAL NETWORKS: Basics using MATLAB Neural Network Toolbox.
5. Παλαιότερες Διπλωματικές Εργασίες φοιτητών:
 - «Εντοπισμός ατελειών σε επίπεδη πλάκα ομογενούς και σύνθετου υλικού με χρήση νευρωνικών δικτύων και μεθόδου πεπερασμένων στοιχείων», του κυρίου Ευστάθιου Α. Ανδριανάκης.
 - «Μοντέλο πρόβλεψης του συνολικού ηλεκτρονικού περιεχομένου της ιονόσφαιρας με μέθοδο νευρωνικών δικτύων» του κυρίου Χαράλαμπου Α. Δημήτριου.