



Νευρωνικά Δίκτυα και Ευφυή Υπολογιστικά Συστήματα

9ο Εξάμηνο

1η Εργαστηριακή Άσκηση

Μελέτη των Πολυεπίπεδων Perceptrons και Εφαρμογή σε Προβλήματα Ταξινόμησης Εικόνας

Αργυρίου Στέφανος Παναγιώτης
03112006

Κονιδάρης Φίλιππος
03112011

1. Σκοπός:

Σκοπός της πρώτης εργαστηριακής άσκησης αποτέλεσε η δημιουργία και η εκπαίδευση νευρωνικών δικτύων τα οποία διέφεραν σε βασικά χαρακτηριστικά, όπως η αρχιτεκτονική (αριθμός επιπέδων και αριθμός νευρώνων ανά επίπεδο), η συνάρτηση εκπαίδευσης, η συνάρτηση ενεργοποίησης, ο αλγόριθμος μάθησης, η ύπαρξη ή μη ενός validation set, ο αριθμός εποχών εκπαίδευσης και ο ρυθμός μάθησης. Για κάθε μία από τις παραπάνω παραμέτρους πάρθηκαν εκτενείς μετρήσεις των μετρικών που αναφέρονται στη συνέχεια και έγινε η απαραίτητη σύγκριση προκειμένου στο τέλος της υλοποίησης να μπορούμε να καταλήξουμε στις ιδανικές τιμές που συνθέτουν το καλύτερο δίκτυο για τη συγκεκριμένη εφαρμογή.

2. Διαδικασία Αξιολόγησης Δικτύων:

Στη διάρκεια της διεκπεραίωσης του project, όπως αναφέρθηκε και παραπάνω δημιουργήθηκαν δεκάδες νευρωνικά δίκτυα, τα οποία συγκρίναμε μεταξύ τους. Βασική μετρική για τη σύγκριση των δικτύων αποτέλεσε ο δείκτης Accuracy, ο οποίος ορίστηκε ως εξής:

$$Accuracy = \frac{\text{Σωστά ταξινομημένα τμήματα}}{\text{Σύνολο τμημάτων}}$$

Τα δίκτυα τα οποία είχαν μεγαλύτερο Accuracy θεωρούνταν στη γενική περίπτωση καλύτερη από τα άλλα. Ωστόσο, η χρήση της μετρικής Accuracy από μόνης της δεν έδινε όλες τις απαραίτητες πληροφορίες για να καταλήξουμε σε ορθά συμπεράσματα. Προκειμένου ένα δίκτυο να ληφθεί υπόψη στη διαδικασία εξαγωγής του καλύτερου δικτύου θα πρέπει η μετρική fmeasure για το συγκεκριμένο δίκτυο να είναι μεγαλύτερη του 0,8. Η τιμή αυτή επιλέχθηκε με βάση παρατηρήσεις που έγιναν στα δίκτυα. Πιο συγκεκριμένα, παρατηρήσαμε ότι τα δίκτυα που είχαν fmeasure μικρότερο από 0,8 έτειναν να είναι εκπαιδευμένα ανομοιόμορφα όσον αφορά τις 5 κατηγορίες που αφορούσαν τα δεδομένα του προβλήματος. Η μετρική fmeasure προέκυψε ως:

$$fmeasure = \frac{\sum recall + \sum precision}{5}$$

Ταυτόχρονα, οι μετρικές recall και precision υπολογίζονται από τους τύπους:

$$recall = \frac{\text{Σωστά ταξινομημένα τμήματα στην κατηγορία}}{\text{Σύνολο τμημάτων στην κατηγορία}}$$

$$precision = \frac{\text{Σωστά ταξινομημένα τμήματα στην κατηγορία}}{\text{Ταξινομημένα τμήματα στην κατηγορία}}$$

Οι τρεις προαναφερθείσες παράμετροι υπολογίζονται μέσω της συνάρτησης eval_Accuracy_Precision_Recall που δίνεται στην εκφώνηση του project.

3. Προεπεξεργασία των Δεδομένων:

Προτού αρχίσει η εκπαίδευση των νευρωνικών δικτύων που χρησιμοποιούμε, είναι σημαντικό να προβούμε σε κάποια προεπεξεργασία των δεδομένων προκειμένου να μειώσουμε τους απαιτούμενους χρόνους για την εκπαίδευση, αλλά και να εξασφαλίσουμε καλύτερα αποτελέσματα.

- Αρχικά βρίσκουμε ποιο από τα χαρακτηριστικά (5) που μας ενδιαφέρουν εμφανίζεται τις λιγότερες φορές στα δεδομένα εκπαίδευσης και κρατάμε τόσα δεδομένα από το κάθε χαρακτηριστικό. Η δράση αυτή πραγματοποιείται προκειμένου να δημιουργήσουμε ένα ομοιόμορφο σύνολο από δεδομένα εκπαίδευσης χωρίς κάποιο από αυτά να υπερ-εκπροσωπείται, είτε το αντίθετο. Έτσι, αυξάνουμε την ποιότητα του δικτύου πετυχαίνοντας καλύτερες τιμές στο fmeasure, όπως αυτό περιγράφηκε παραπάνω.
- Μέσω της συνάρτησης removeConstantRows, διαγράφουμε όλες τις γραμμές οι οποίες είναι

ίδιες για όλα τα δεδομένα εκπαίδευσης. Με άλλα λόγια, διαγράφουμε όλα αυτά τα στοιχεία τα οποία δε θα προσφέρουν καμία διαφοροποίηση και άρα δε θα είναι αξιοποιήσιμα από το νευρωνικό μας. Η ίδια διαδικασία γίνεται αντίστοιχα και στα δεδομένα ελέγχου.

- Με την `mapstd` κανονικοποιούμε τον πίνακα των δεδομένων εκπαίδευσης ώστε να έχουμε τυπική απόκλιση 1 και μέσο όρο 0.
- Τέλος, μέσω της συνάρτησης `proccessprca` μειώνουμε τις διαστάσεις του πίνακα αποφεύγοντας όμως να χάσουμε σημαντικές πληροφορίες. Σε αυτή την περίπτωση κρατήσαμε τον αριθμό των στηλών του πίνακα `myTrainData` σε 20. Ομοίως κάναμε και για το `myTestData`.

Αν δεν είχε πραγματοποιηθεί η παραπάνω προεπεξεργασία των δεδομένων εισόδου, θα υπήρχε τυχαίότητα στη σειρά εμφάνισής τους, οδηγώντας σε διαφορετικό αριθμό προτύπων ανά κατηγορία. Το γεγονός αυτό έχει ως αποτέλεσμα την άνιση εκπαίδευση του δικτύου και άρα μεγαλύτερη αδυναμία αυτού να κατηγοριοποιήσει ορθά τα δεδομένα ελέγχου (δηλαδή χαμηλότερο `accuracy`).

4. Περιγραφή της διαδικασίας

Την προεπεξεργασία των δεδομένων ακολούθησε το κύριο μέρος του πρότζεκτ, η παρατήρηση δηλαδή της επίδρασης των διάφορων παραμέτρων των αρχιτεκτονικών στην εκπαίδευση και τελική απόδοση τους.

4.1. Γενικά Σχόλια:

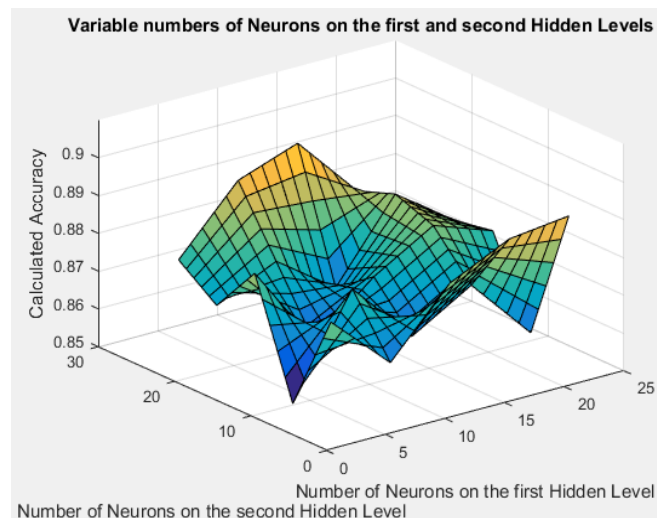
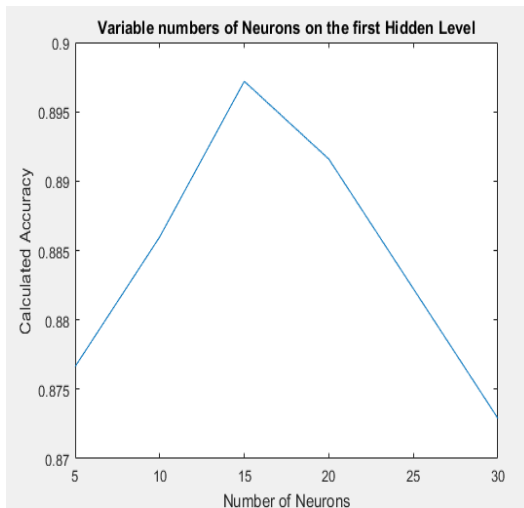
Ακολουθήθηκε μοντέλο συναρτησιακού προγραμματισμού, με σκοπό την ευκολία στον προγραμματισμό του εκάστοτε δικτύου, αλλά και την κατανόηση του υπάρχοντος κώδικα μεταξύ των δύο συνεργατών. Λόγω της τυχαίας αρχικοποίησης των βαρών της συνάρτησης `newff`, κάθε πείραμα εκτελέστηκε πέντε φορές και σα μετρική της απόδοσης χρησιμοποιήθηκε ο εκάστοτε μέσος όρος, με σκοπό την όσο το δυνατόν μεγαλύτερη ελαχιστοποίηση του σφάλματος. Τα πρώτα 5 ζητούμενα εκτελούνται μέσα από το αρχείο `initialScript.m`, ενώ τα ζητούμενα 6 και 7 εκτελούνται από το αρχείο `step6supervisor.m`. Ο διαχωρισμός αυτός προέκυψε από το γεγονός ότι στο τέλος του βήματος 5 γνωρίζουμε την ιδανική αρχιτεκτονική, δηλαδή τον καλύτερο αριθμό επιπέδων (1 ή 2) και τον καλύτερο αριθμό νευρώνων ανά επίπεδο, επομένως κρίναμε άσκοπο να συνεχίσουμε να υλοποιούμε κώδικα παραμετρικά με αντίστοιχες συνθήκες επιλογής (`if`).

4.2. Μελέτη της επίδρασης της Αρχιτεκτονικής του δικτύου:

Η πρώτη παράμετρος που δοκιμάστηκε ήταν ο αριθμός των νευρώνων του δικτύου. Μέσω του script `step3.m` δημιουργήθηκαν αρχιτεκτονικές ενός κρυμμένου επιπέδου, οι οποίες διαθέτουν από 5 έως 30 νευρώνες, ο αριθμός των οποίων αυξάνεται ανά 5 κάθε φορά. Για την εύρεση των `accuracy`, `precision` και `recall` χρησιμοποιείται κάθε φορά η συνάρτηση `eval_Accuracy_Precision_Recall`. Τα αποτελέσματα αποθηκεύονται στους πίνακες `accuracy`, `precision` και `recall`. Βλέπουμε ότι για μικρό αριθμό νευρώνων το δίκτυο έχει χαμηλή απόδοση σε σχέση με αυτά που ακολουθούν. Βέλτιστη απόδοση επιτυγχάνεται για μεγάλο αριθμό νευρώνων, 25 ή 30. Το εκάστοτε `fmeasure` υπολογίζεται εδώ από τη συνάρτηση `calc_fmeasure1`, η οποία δέχεται ως εισόδους τους πίνακες `precision` και `recall` και επιστρέφει έναν πίνακα μίας στήλης.

Στη συνέχεια, αυξάνουμε τον αριθμό των κρυμμένων επιπέδων σε δύο, με το κάθετο να διαθέτει από 5 έως 30 νευρώνες, ο αριθμός των οποίων αυξάνεται ακριβώς όπως και πριν, ενώ οι άλλες παράμετροι παρέμειναν στις προεπιλεγμένες τιμές τους. Τα δίκτυα εκπαιδεύονται και δοκιμάζονται μέσω του script `step3b.m`, ενώ αυτή τη φορά λόγω των δύο υπό δοκιμή παραμέτρων, τα αποτελέσματα αποθηκεύονται σε δισδιάστατους πίνακες. Για την εύρεση των `fmeasure` χρησιμοποιήθηκε η `calc_fmeasure`, η οποία αποτελεί επέκταση της `calc_fmeasure1` για πίνακες 2 διαστάσεων, δέχεται τις ίδιες εισόδους και επιστρέφει έναν συμπυκνωμένο πίνακα με 5 φορές λιγότερες στήλες.

Όπως φαίνεται από τα αποτελέσματα, κανοποιητική ακρίβεια ξεκινά να επιτυγχάνεται από 15 νευρώνες και άνω. Μέγιστη ακρίβεια επιτυγχάνεται σε αρχιτεκτονική που συνδυάζει μεγάλο αριθμό νευρώνων στο πρώτο επίπεδο και μικρό αριθμό στο δεύτερο: Εάν το πρώτο επίπεδο διαθέτει μικρό αριθμό νευρώνων (5 ή 10) δεν προσφέρει κάποια ουσιαστική βοήθεια στην αποσύνθεση και τμηματοποίηση των εικόνων και λειτουργεί περισσότερο ως buffer, οπότε αυτόματα μειώνεται και η επίδοση του δεύτερου επιπέδου, όσο πολύπλοκο και να είναι. Η σχετικά μικρή διαφορά στην απόδοση εντοπίζεται στη σωστή προεπεξεργασία που έχει προηγηθεί. Παράλληλα, μεγάλος αριθμός νευρώνων και στα δύο επίπεδα, αν και φαινομενικά δεν επηρεάζει αρνητικά την επίδοση, βαρύνει το σύστημα με πολυπλοκότητα που δεν είναι απαραίτητη.

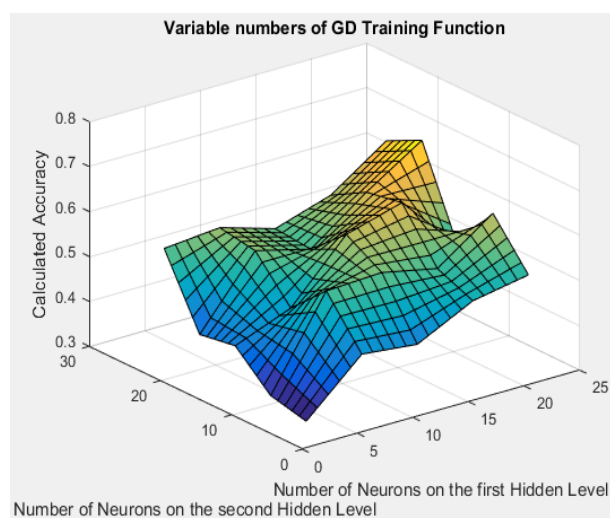
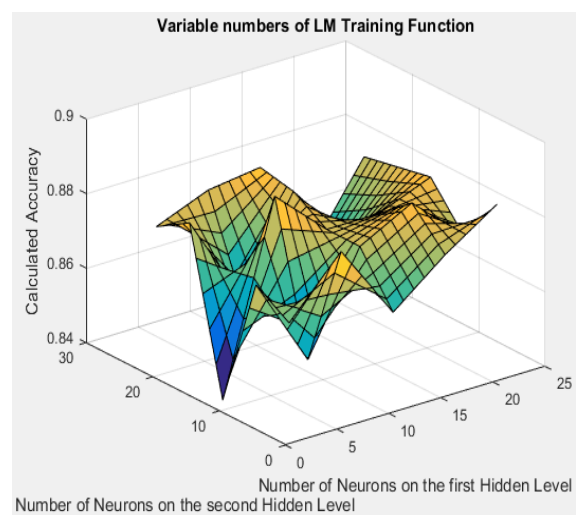
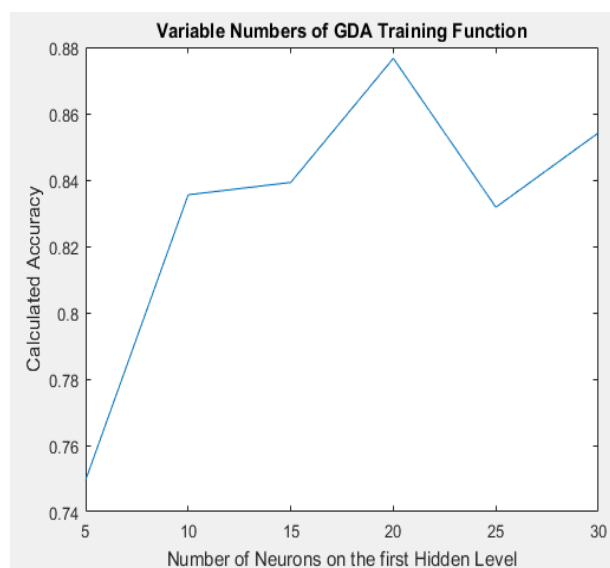
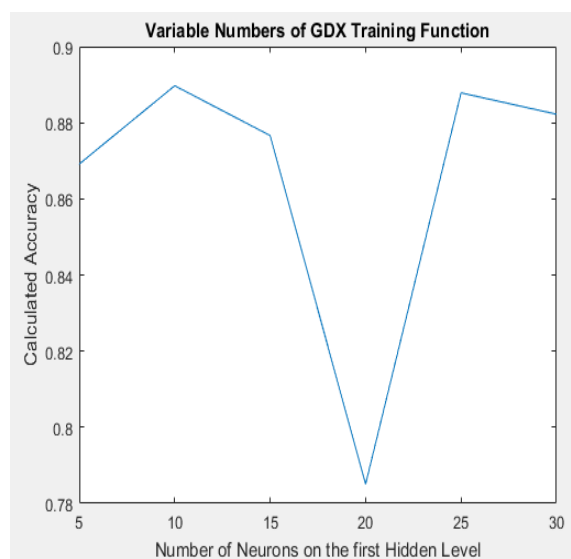
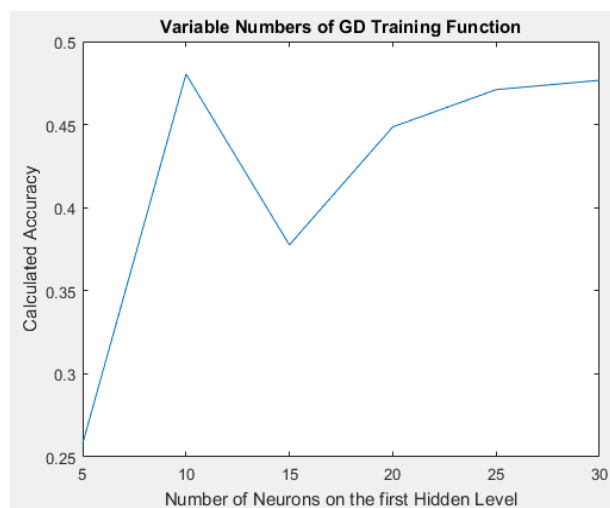
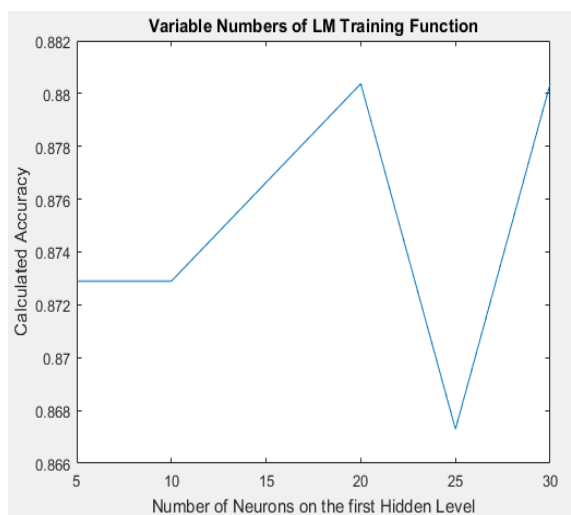


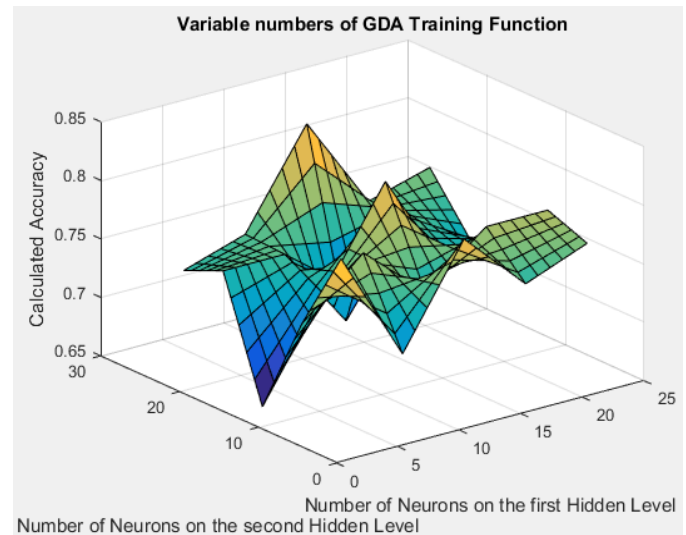
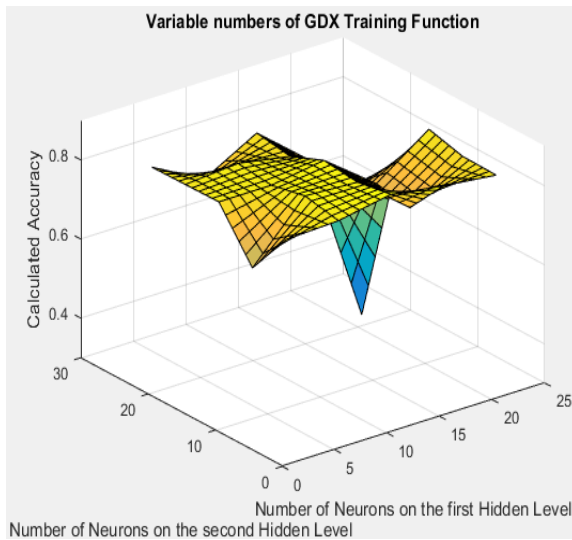
4.3. Μελέτη της επίδοσης των συναρτήσεων εκπαίδευσης:

Μετά εξετάστηκαν ως προς την απόδοσή τους οι συναρτήσεις εκπαίδευσης LM, GD, GDX και GDA. Λόγω του καθοριστικού τρόπου με τον οποίο η συνάρτηση εκπαίδευσης επηρεάζει την απόδοση, και για την όσο το δυνατόν πιο σίγουρη επιλογή του καλύτερου δικτύου, εξετάστηκαν από την αρχή όλες οι αρχιτεκτονικές με όλες τις συναρτήσεις εκπαίδευσης. Οι αρχιτεκτονικές ενός κρυμμένου επιπέδου δημιουργήθηκαν στο script `step4.m`, και αντίστοιχα των δύο στο `step4b.m`. Το βήμα αυτό είναι το πιο απαιτητικό όσον αφορά τόσο το χρόνο, όσο και τους πόρους που απαιτεί για να παράξει το αποτέλεσμα του, τους πίνακες δηλαδή με τους μέσους όρους των accuracy και του fmeasure κάθε αρχιτεκτονικής και μεθόδου εκπαίδευσης.

Η εύρεση της αρχιτεκτονικής και της συνάρτησης εκπαίδευσης που δίνουν μέγιστο accuracy πραγματοποιείται μέσω της συνάρτησης `findBestTraining`, η οποία δέχεται ως ορίσματα τους παραπάνω πίνακες και επιστρέφει στο `initialScript.m` τη μεταβλητή `name`, στην οποία είναι αποθηκευμένο το όνομα της βέλτιστης μεθόδου εκπαίδευσης, και τις μεταβλητές `neuronsFirstLevel` και `neuronsSecondLevel`, ο σκοπός των οποίων είναι προφανής.

Τα αποτελέσματα του συγκεκριμένου βήματος έδειξαν ότι η καλύτερη συνάρτηση εκπαίδευσης είναι η LM, ενώ η καλύτερη αρχιτεκτονική είναι αυτή που έχει 2 επίπεδα νευρώνων με 25 νευρώνες στο πρώτο και 10 νευρώνες στο δεύτερο.



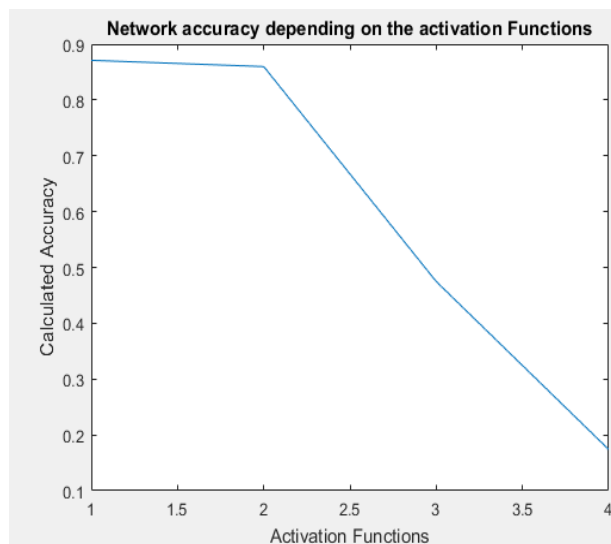


4.4. Μελέτη των υπόλοιπων υπερπαραμέτρων του δικτύου:

Οι διάφορες παράμετροι που εξετάζονται στο βήμα 6 είναι ανεξάρτητες της μεθόδου εκπαίδευσης που θα χρησιμοποιηθεί.

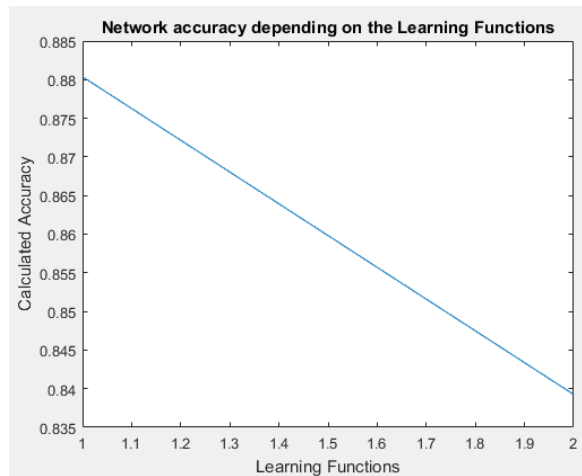
- Μελέτη της επίδοσης των συναρτήσεων ενεργοποίησης:

Στο βήμα αυτό δοκιμάζονται διάφορες συναρτήσεις ενεργοποίησης, μέσω του script `step6a.m`, που δέχεται ως εισόδους τα αποτελέσματα του Βήματος 5. Τα αποτελέσματα επιστρέφονται στον supervisor σε πίνακες 4 γραμμών, οι οποίοι προωθούνται στη συνέχεια στη συνάρτηση `findBestActivation`, η οποία δέχεται τον μονοδιάστατο πίνακα τεσσάρων γραμμών (`purelin`, `tansig`, `logsig`, `hardlim`) των μέσων όρων ακρίβειας των συναρτήσεων ενεργοποίησης και επιστρέφει το όνομα αυτής που οδήγησε το δίκτυο στη βέλτιστη λειτουργία. Το όνομα αυτό αποθηκεύεται στη μεταβλητή `BestActivation` του supervisor και η εκτέλεση προχωρά στο επόμενο βήμα. Καλύτερη συνάρτηση ενεργοποίησης προέκυψε η `purelin`.



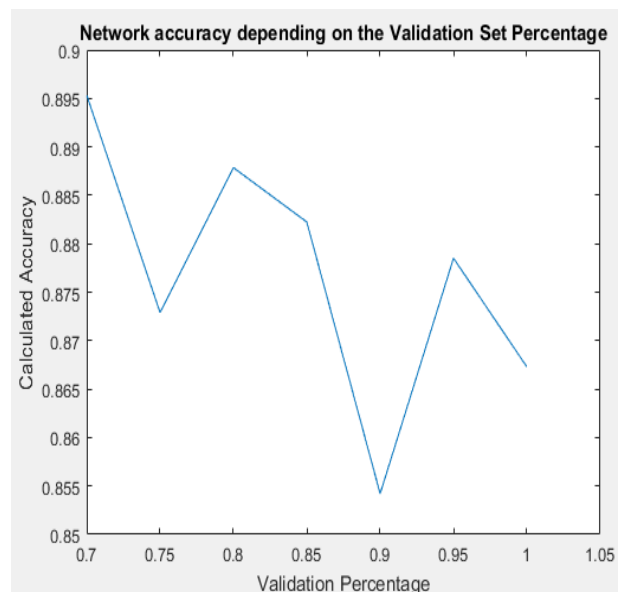
- Μελέτη της επίδοσης των συναρτήσεων μάθησης:

Στο βήμα αυτό εξετάζονται οι επιδόσεις των συναρτήσεων μάθησης `learngd` και `learngdm`. Το script `step6b.m` δέχεται ως είσοδο τα αποτελέσματα των προηγούμενων βημάτων και επιστρέφει έναν πίνακα 2 γραμμών όπου στην πρώτη αποθηκεύονται τα αποτελέσματα της `learngd` και στη δεύτερη της `learngdm`. Ο πίνακας τροφοδοτείται στη συνάρτηση `findBestLearning` μέσω του `supervisor`, η οποία επιστρέφει το όνομα της καλύτερης από τις δύο συναρτήσεων μάθησης, η οποία αποθηκεύεται στη μεταβλητή `BestLearn`. Καλύτερη συνάρτηση μάθησης προέκυψε η `learngd`.



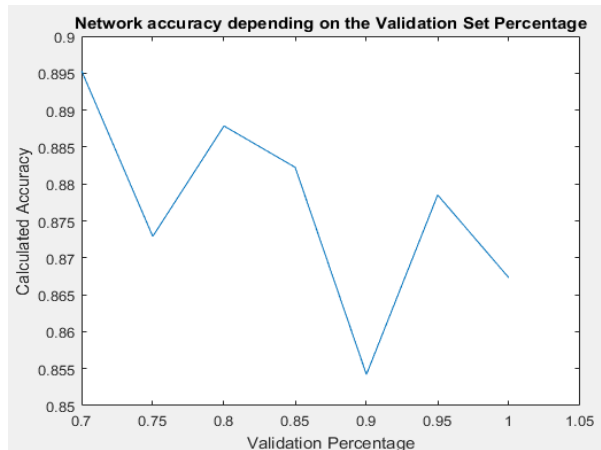
- Επίδραση της ύπαρξης του validation set στο δίκτυο:

Σε αυτή τη φάση της εκτέλεσης δοκιμάζεται ένα πολύ σημαντικό χαρακτηριστικό του δικτύου, η ύπαρξη (οπότε και το ποσοστό) ή όχι validation set. Στο script `step6c.m` δοκιμάζουμε ποσοστά εκπαίδευσης από 65% και άνω, με την ιδέα ότι οτιδήποτε χαμηλότερα θα επηρέαζε την επίδοση του δικτύου, και φτάνουμε μέχρι και το 100%, να μην υπάρχει δηλαδή validation set. Η συνάρτηση `findBestValidation` βρίσκει και επιστρέφει το ποσοστό εκπαίδευσης που οδηγεί σε βέλτιστη απόδοση, αλλά και τον αριθμό των εποχών που σταμάτησε η εκπαίδευση του δικτύου αυτού. Τα αποτελέσματα φαίνονται στο παρακάτω διάγραμμα.



- Επίδραση του αριθμού εποχών στην επίδοση του δικτύου:

Εξετάζουμε τις επιδόσεις του δικτύου χωρίς validation set, όταν όμως αυτό εκπαιδεύεται σε αριθμό εποχών συγκρίσιμο με αυτόν που εκπαιδεύτηκε υπό την παρουσία validation set. Για το λόγο αυτό λαμβάνει ως είσοδο των αριθμό των βέλτιστων εποχών που υπολογίστηκε προηγούμενα και το 20% του αριθμού αυτού μετά από στρογγυλοποίηση και αύξηση κατά 1, εκτελεί τις προσομοιώσεις και επιστρέφει τα αποτελέσματα. Οποιοδήποτε ποσοστό μικρότερο από 20% θεωρήθηκε υπερβολικά μικρό, αφού σχεδόν κανένα δίκτυο δεν εκπαιδεύονταν για πάνω από 20 εποχές, ενώ ακριβώς λόγω του μικρού αριθμού εποχών, κάθε ποσοστό μεγαλύτερο από 20% θα οδηγούσε σε μεγάλες αποκλίσεις. Ο βέλτιστος αριθμός εποχών βρίσκεται από τη συνάρτηση `findBestEpochsNoValidation` και είναι ίσος με 12 εποχές.



- Επίδραση του ρυθμού μάθησης στην επίδοση της Gradient Descent και της Gradient Descent with Momentum

Στο τελευταίο βήμα στο οποίο καταγράψαμε μετρήσεις, ασχοληθήκαμε με την επίδοση των GD και GDX για διάφορες τιμές του ρυθμού μάθησης. Πιο συγκεκριμένα, ασχοληθήκαμε με μικρές τιμές του ρυθμού μάθησης, από 0.05 έως 0.4. Σε αυτές τις μικρές τιμές φάνηκε ότι καλύτερη συνάρτηση εκπαίδευσης ήταν η απλή Gradient Descent για ρυθμό μάθησης ίσο με 0.3.

4.5. Καταγραφή παραμέτρων του καλύτερου δικτύου για τμηματοποίηση εικόνας:

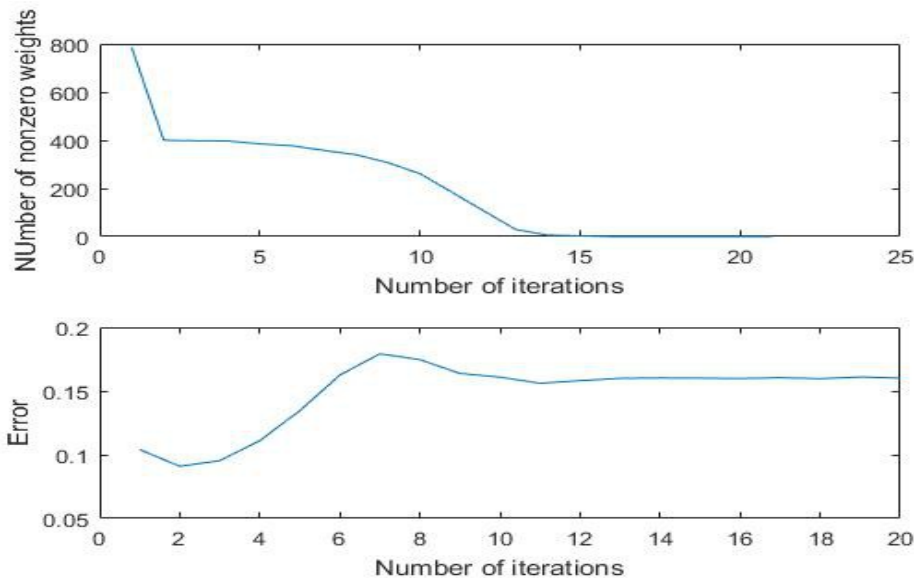
Με βάση όλα τα παραπάνω, το καλύτερο νευρωνικό για τη συγκεκριμένη εφαρμογή προέκυψε το εξής:

- Νευρωνικό 2 κρυμμένων επιπέδων:
 - 25 νευρώνες στο πρώτο επίπεδο
 - 10 νευρώνες στο δεύτερο επίπεδο
- Συνάρτηση Εκπαίδευσης: Levenberg-Marquardt (LM)
- Συνάρτηση Ενεργοποίησης: Purelin
- Συνάρτηση Μάθησης: Gradient Descent (learnGD)

5. Μέθοδος Weight Decay

Η μέθοδος της αποσύνθεσης βαρών (weight decay method) έχει ως βάση την παραδοχή ότι τα βάρη του δικτύου που δεν υπερβαίνουν κάποιο κατώφλι, δρουν ως θόρυβος επηρεάζοντας την απόδοση του δικτύου και συνεπώς θα ήταν προτιμότερο να αγνοηθούν.

Για τις μετρήσεις λάβαμε τις παραμέτρους λ και d ίσες με 0.3 και 0.01 αντίστοιχα, καθώς αυτές οι τιμές προέκυψαν ως οι καλύτερες μετά από πειραματική παρατήρηση.



Παρατηρούμε ότι ο αριθμός των μη-μηδενικών βαρών μηδενίζεται περίπου στην 14η εποχή, πράγμα που σημαίνει ότι τα βάρη που θεωρούνται μη-πρόσφορα έχουν πλέον σταθεροποιηθεί. Κάτι τέτοιο έχει σαν άμεση συνέπεια τη σταδιακή αύξηση και κατόπιν τη σταθεροποίηση του σφάλματος. Για να αποφευχθεί κάτι τέτοιο θα μπορούσαμε να χρησιμοποιήσουμε μικρότερο αριθμό εποχών, κοντά στις 14.

6. Θεωρητικά ζητήματα:

6.1. Χρησιμότητα Προεπεξεργασίας Δεδομένων

Σε θεωρητικά πλαίσια, η εκπαίδευση του δικτύου θα μπορούσε να ολοκληρωθεί ακόμα και στην περίπτωση όπου παραλείπαμε το σύνολο της διαδικασίας προεπεξεργασίας, αλλά τα αποτελέσματα που θα είχαμε δε θα είχαν την ανάλογη ακρίβεια, ενώ παράλληλα θα επιμηκυνόταν ο χρόνος που θα απαιτούνταν για την εκπαίδευση του δικτύου σε κάθε εποχή.

Με το να κρατάμε ίσο αριθμό από TestData σε κάθε κατηγορία επιτυγχάνουμε πιο ομοιόμορφα δεδομένα και κατ' επέκταση καλύτερες τιμές του δείκτη fmeasure. Παράλληλα, αφαιρούμε τις τιμές που είναι όλες ίδιες για κάθε χαρακτηριστικό, καθώς δεν προσδίδουν επιπλέον πληροφορίες στο σύστημα. Αμέσως μετά, χρησιμοποιούμε τη συνάρτηση processpca, η οποία μας επιτρέπει σε ένα σύνολο δεδομένων να μειώσουμε τις διαστάσεις των δεδομένων χωρίς να χάσουμε σημαντική πληροφορία. Χρησιμοποιεί έναν ορθογώνιο μετασχηματισμό για να μετατρέψει ένα σύνολο συσχετιζόμενων τιμών σε γραμμικά ασυσχέτιστες.

6.2. Σύγκριση διαφορετικών αρχιτεκτονικών και συναρτήσεων μάθησης

Ιδανικότερος συνδυασμός προέκυψε να είναι το νευρωνικό που προκύπτει από την αρχιτεκτονική 2 επιπέδων με 25 νευρώνες στο πρώτο επίπεδο και 10 στο δεύτερο με χρήση συνάρτησης μάθησης να είναι η LM. Η διαφορά στην απόδοση των LM, GDA και GDX ήταν πολύ μικρή, ενώ η απόδοση της GD βρισκόταν πολύ χαμηλότερα από τις άλλες. Αυτό είναι εύλογο αν αναλογιστεί κανείς ότι η GD μέθοδος (gradient descent) έχει σταθερό βήμα μάθησης, το οποίο ορίζεται αρκετά μικρό, με αποτέλεσμα οι 1000 εποχές οι οποίες τέθηκαν ως παράμετρος στο χρόνο εκπαίδευσης να είναι πάρα πολύ λίγες για την ορθή εκπαίδευση του δικτύου. Οι GDA και GDX είναι τροποποιημένες μέθοδοι της GD, οι οποίες έχουν μεταβαλλόμενο βήμα μάθησης (adaptive learning rate), αλλά αφενός είναι αργές

μέθοδοι (απαιτούν μεγάλο αριθμό εποχών) και αφετέρου δεν είναι βέβαιο ότι μπορούν να καταλήξουν πάντα στο ολικό ελάχιστο. Αντίθετα, η μέθοδος LM (Levenberg-Marquadt) αποτελεί συνδυασμός των παραπάνω μεθόδων μαζί με τη μέθοδο Newton, γεγονός που προσδίδει και ταχύτητα και καλύτερα αποτελέσματα στην εκπαίδευση του δικτύου μας.

Από την άλλη πλευρά, αν και διαισθητικά είναι δύσκολη η αιτιολόγηση της επιλεγόμενης αρχιτεκτονικής, αξίζει να σημειωθεί ότι η ύπαρξη των 25 νευρώνων στο πρώτο επίπεδο δηλώνει την αποσύνθεση της πληροφορίας εισόδου σε πολυάριθμα τμήματα, τα οποία εν συνεχεία συγκεντρώνονται στους 10 νευρώνες του δεύτερου επιπέδου προτού το σύστημα δώσει το προτεινόμενο αποτέλεσμα.

6.3. Σύγκριση συναρτήσεων ενεργοποίησης

Καλύτερο αποτέλεσμα έδινε η purelin. Εκτός από αυτήν, δοκιμάστηκαν οι tansig, logsig και hardlim. Από αυτές, η tansig έδινε παρόμοια, αλλά λίγο χειρότερα αποτελέσματα, ενώ η επίδοση των logsig και hardlim ήταν απογοητευτική. Αυτό συνέβη διότι οι δύο συναρτήσεις δε θα έπρεπε καν να χρησιμοποιούνται σε συνδυασμό με τις παραμέτρους του δικτύου που είχαμε διαλέξει. Συγκεκριμένα, η hardlim δίνει ως έξοδο 1 για $u \geq 0$, και 0 για $u < 0$, είναι δηλαδή μη παραγωγίσιμη, πράγμα που την καθιστά μη συμβατή με τη συνάρτηση trainlm, η οποία χρησιμοποιεί μία παραλλαγή της μεθόδου απότομης καθόδου οπότε αυτόματα απαιτεί παραγωγισιμότητα. Το αποτέλεσμα είναι να παράγονται δεδομένα NaN (Not a Number).

Όσον αφορά τη logsig, αυτή δεν δίνει σωστά αποτελέσματα λόγω του default τρόπου με τον οποίο το νευρωνικό δίκτυο κανονικοποιεί τα δεδομένα εξόδου. Συγκεκριμένα, τα μετασχηματίζει χρησιμοποιώντας τη συνάρτηση marminmax, η οποία τα κανονικοποιεί σε διάστημα $[-1,1]$, ενώ το πεδίο τιμών της logsig είναι το $[0,1]$, με αποτέλεσμα τα μισά δεδομένα ουσιαστικά να αγνοούνται. Η logsig θα μπορούσε να χρησιμοποιηθεί εάν τα δεδομένα εξόδου κανονικοποιούνταν με κάποιον συμβατό τρόπο, πχ με marminmax, αύξηση κατά 1 και μετά διαίρεση με 2.

6.4. Gradient Descent και Gradient Descent με προσθήκη όρου ορμής

Βασική παρατήρηση που μπορεί να γίνει στα δεδομένα σε σχέση με την Gradient Descent και την παραλλαγή της είναι ότι με την προσθήκη του όρου ορμής έχουμε πολύ πιο ακριβή και πολύ και γρήγορη σύγκλιση του νευρωνικού στην επιθυμητή κατάσταση. Κάτι τέτοιο είναι εύλογο, αν αναλογιστεί κανείς ότι ο απλός αλγόριθμος του Gradient Descent εμφανίζει δυσκολία στη διάσχιση “χαραδρών”, δηλαδή των περιοχών όπου η επιφάνεια καμπυλώνεται πολύ πιο απότομα στη μία διάσταση σε σχέση με τις υπόλοιπες διαστάσεις του προβλήματος. Τέτοιες περιοχές είναι συχνό φαινόμενο γύρω από τοπικά μέγιστα. Σε αυτές της περιπτώσεις, η απλή GD περιτριγυρίζει το γεωμετρικό τόπο των δεδομένων κάνοντας κάθε φορά μικρά και διστακτικά βήματα προς τη λύση. Αντίθετα, με την προσθήκη του όρου ορμής, η διαδικασία της εκπαίδευσης συμπεριφέρεται όπως ένα σώμα που συλλέγει ορμή και αυξάνει την κινητική του κατάσταση καθώς προχωράει προς το στόχο του. Έτσι και η τροποποιημένη GD καθώς προχωράει η εκπαίδευση “κινείται” με ολοένα και μεγαλύτερα βήματα προς την επιθυμητή κατάσταση του δικτύου.

6.5. Αναφορά στη μέθοδο Early Stopping

Η μέθοδος Early Stopping είναι μία μορφή κανονικοποίησης, η οποία χρησιμοποιείται για την αποφυγή του προβλήματος του overfitting όταν χρησιμοποιούνται επαναληπτικές μέθοδοι για την εκπαίδευση ενός νευρωνικού δικτύου. Επειδή και στη συγκεκριμένη περίπτωση χρησιμοποιούμε επαναληπτικές μεθόδους για την εκπαίδευση του δικτύου, η χρήση της κρίνεται απαραίτητη. Πιο συγκεκριμένα, επιλέγουμε το χωρισμό των δεδομένων του training set σε 2 κατηγορίες, training και validation, σε ποσοστά 80% και 20% αντίστοιχα. Με τη διαδικασία αυτή, το 80% των δεδομένων χρησιμοποιείται για την εκπαίδευση του perceptron (δηλαδή τον υπολογισμό των βαρών του), ενώ το υπόλοιπο 20% χρησιμοποιείται για validation στο τέλος της κάθε εποχής. Έτσι, αν σε μία εποχή

προκύψει μεγαλύτερο classification error στο validation set σε σχέση με την προηγούμενη, η εκπαίδευση σταματά ακόμα και αν δεν έχει συμπληρωθεί ο αριθμός των εποχών που έχουμε ορίσει. Πράγματι, στα νευρωνικά δίκτυα όπου δεν έχουμε Early Stopping, το accuracy όταν το δίκτυο ξεκινάει να εργάζεται πάνω στο σύνολο των TestData είναι εμφανώς μικρότερο από όταν χρησιμοποιείται το Early Stopping.

6.6. Σημασία αριθμού εποχών

Όταν ο αριθμός των εποχών είναι μικρός, ορισμένες training methods δεν μπορούν να βγάλουν ορθά αποτελέσματα (GD), ενώ σε άλλες (LM) μειώνεται η αποτελεσματικότητα. Πιο συγκεκριμένα, η GD, η οποία διαθέτει σταθερό ρυθμό μάθησης δεν προλαβαίνει να εκπαιδεύσει σωστά το σύστημα, ενώ ακόμα και σε μεθόδους με προσαρμοζόμενο ρυθμό μάθησης υπάρχει κίνδυνος να συμβεί το ίδιο. Από την άλλη πλευρά, όταν ο αριθμός των εποχών είναι υπερβολικά μεγάλος, υπάρχει κίνδυνος overfitting και αυξάνεται γραμμικά ο χρόνος εκπαίδευσης. Η αύξηση του χρόνου είναι αυτόδηλη, ενώ το πρόβλημα του overfitting μπορεί να γίνει ιδιαίτερα έντονο αν δεν έχει χρησιμοποιηθεί η τεχνική Early Stopping. Φυσικά, αν αυτή χρησιμοποιείται, η αύξηση του αριθμού των εποχών ελάχιστα επηρεάζεται το τελικό accuracy του δικτύου.

6.7. Επίδραση του ρυθμού μάθησης στην απόδοση των νευρωνικών

Όταν ο ρυθμός μάθησης είναι πολύ μικρός, ο χρόνος που απαιτείται για την εκπαίδευση του δικτύου αυξάνεται, γεγονός το οποίο αποτυπώνεται στον απαραίτητο αριθμό εποχών που πρέπει να προηγηθεί προκειμένου να φτάσουμε στα επιθυμητά αποτελέσματα. Αντίθετα, όταν ο ρυθμός μάθησης είναι πολύ μεγάλος, μειώνεται η ακρίβεια του δικτύου, με αποτέλεσμα στο τέλος της εκπαίδευσης να υπάρχει ο κίνδυνος δημιουργία ενός δικτύου που δίνει λάθος αποτελέσματα.

Η αδυναμία των μικρών ρυθμών μάθησης να παράγουν ικανοποιητικά αποτελέσματα σε μικρό αριθμό εποχών φαίνεται από τη σύγκριση της απλής Gradient Descent (GD) με την GDX. Η απλή GD μέθοδος χρησιμοποιεί σταθερό ρυθμό μάθησης, ο οποίος για τα δεδομένα της άσκησης είναι μικρός. Αυτό έχει ως αποτέλεσμα για μικρό αριθμό εποχών, η GDX, που χρησιμοποιεί προσαρμοζόμενο ρυθμό μάθησης να έχει πολύ καλύτερα αποτελέσματα από την GD.

6.8. Βήματα για την περαιτέρω βελτίωση της απόδοσης

Με παρατήρηση των αποτελεσμάτων του ταξινομητή είναι σαφές ότι οι μετρικές φαίνεται να δείχνουν ότι το δίκτυο έχει εκπαιδευτεί καλύτερα να αναγνωρίζει συγκεκριμένες κατηγορίες από τις 5 που περιέχονται στα δεδομένα. Αν και η προεπεξεργασία που έχουμε κάνει διασφαλίζει μεγαλύτερη ομοιομορφία και ορθότητα στα αποτελέσματα, η τυχαία επιλογή των βαρών στην αρχή της εκπαίδευσης, όπως και η πειραματική επιλογή των υπερπαραμέτρων του δικτύου, αλλοιώνουν την έξοδο του συστήματος. Βασικό βήμα, επομένως, στην προσπάθεια βελτίωσης της απόδοσης του νευρωνικού στο οποίο καταλήξαμε θα ήταν η μη τυχαία επιλογή των βαρών, αλλά η στοχευμένη τοποθέτησή τους στο χώρο των διαστάσεων που ορίζει το πρόβλημα. Παράλληλα, μπορεί να γίνει χρήση ευριστικών και όχι πειραματικών μεθόδων για την εύρεση κατάλληλων υπερπαραμέτρων που ορίζουν το σύστημα.