

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Νευρωνικά Δίκτυα Και Ευφυή Υπολογιστικά Συστήματα
Έτος: 2016 - 2017

1^η Εργαστηριακή Άσκηση :

**«Μελέτη των πολυεπίπεδων Perceptrons και εφαρμογή σε
προβλήματα ταξινόμησης εικόνας»**

Συνεργάτες:

Βαρηά Χρυσούλα - ΑΜ: 03112105

Τσιβρά Κατερίνα - ΑΜ: 03108208

Τα scripts, τα οποία χρησιμοποιήθηκαν για την εκτέλεση κάθε βήματος περιέχονται στο φάκελο zip, μαζί με ενδεικτικές μετρήσεις της εκτέλεσής τους.

Ερώτημα 1 - Προεπεξεργασία των δεδομένων:

Για να γίνει σωστή εκπαίδευση του νευρωνικού δικτύου, απαιτείται η προεπεξεργασία των δεδομένων εισόδου. Η διαδικασία αυτή περιγράφεται στο script `preprocess.m` και παρουσιάζεται παρακάτω:

- Αρχικά βρίσκεται το πλήθος των δεδομένων, το οποίο αντιστοιχεί σε κάθε κατηγορία. Αυτό επιτυγχάνεται αθροίζοντας τα στοιχεία κάθε γραμμής του πίνακα `TrainDataTargets`.
- Από τα αντίστοιχα αθροίσματα, υπολογίζεται το ελάχιστο άθροισμα k και με τη χρήση της συνάρτησης `find()` επιλέγονται k στοιχεία από κάθε κατηγορία.
- Συγκεκριμένα με τη χρήση της συνάρτησης `find()` βρίσκονται οι θέσεις των k πρώτων μη μηδενικών στοιχείων και αποθηκεύονται σε έναν πίνακα `indexes`. Αυτή η διαδικασία είναι απαραίτητη για να εξασφαλιστεί ίδιο πλήθος δεδομένων εισόδου από κάθε κατηγορία και να εκπαιδευτεί σωστά το δίκτυο. Αν στα δεδομένα εισόδου υπάρχουν περισσότερα στοιχεία από μια κατηγορία και λιγότερα από κάποια άλλη, το δίκτυο μπορεί να υπερεκπαιδευτεί στην πρώτη περίπτωση και να μην εκπαιδευτεί αρκετά για την δεύτερη κατηγορία.
- Μετά την εύρεση των θέσεων των ζητούμενων δεδομένων, με τη χρήση της συνάρτησης `randperm()` δημιουργείται μια τυχαία αναδιάταξη, η οποία εφαρμόζεται στον πίνακα `indexes` και εμμέσως στους πίνακες `TrainData` και `TrainDataTargets`. Με αυτόν τον τρόπο επιλέγονται k δεδομένα από κάθε κατηγορία, τα οποία όμως εμφανίζονται με τυχαία σειρά στην είσοδο του δικτύου, εξασφαλίζοντας την ομοιόμορφη εκπαίδευση του ως προς κάθε πρότυπο εισόδου.

Έχοντας εξασφαλίσει τα επιθυμητά δεδομένα εισόδου, εφαρμόζονται τρεις μετασχηματισμοί στους πίνακες `TrainData` και `TestData`. Συγκεκριμένα:

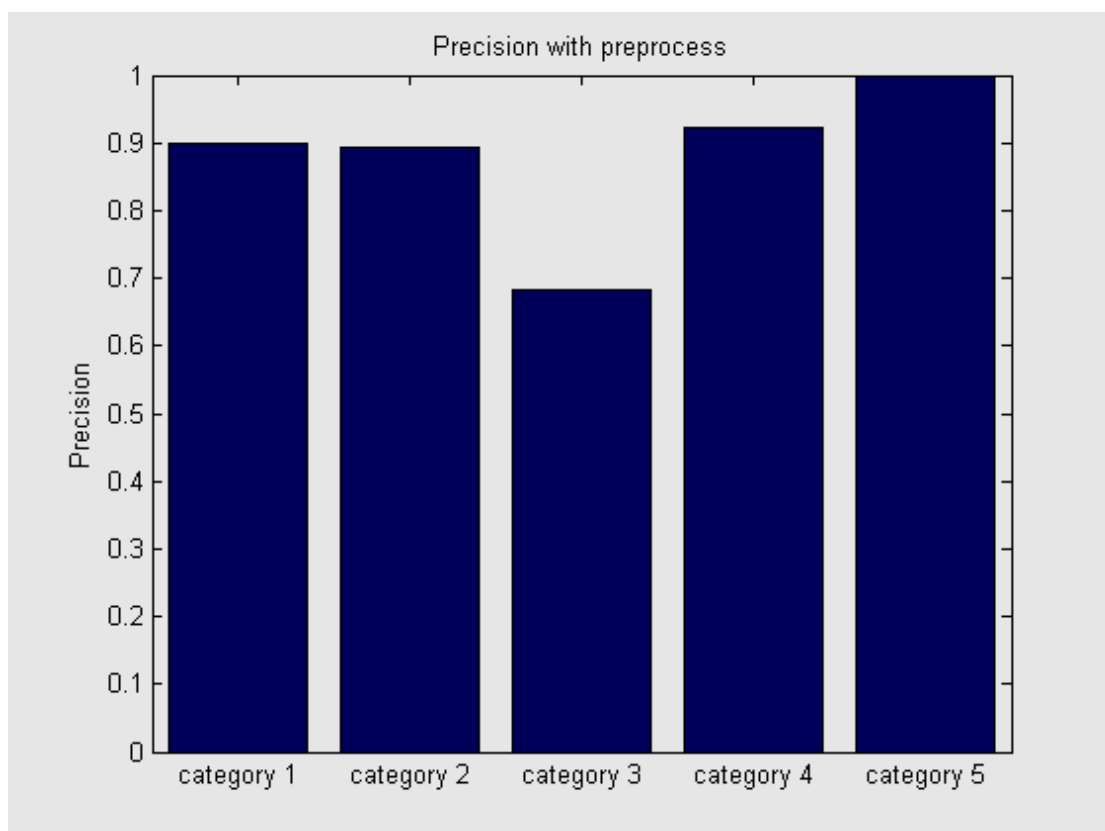
- Πρώτα εφαρμόζεται η συνάρτηση `removeconstantrows()`, η οποία διαγράφει τις γραμμές των στοιχείων που έχουν την ίδια τιμή. Αυτό είναι επιθυμητό διότι δεδομένα με σταθερή τιμή δε συνεισφέρουν στην εκπαίδευση του δικτύου, επομένως δε χάνεται κάποια πληροφορία αν αφαιρεθούν.
- Έπειτα εφαρμόζεται η συνάρτηση `mapstd()`, με την οποία γίνεται κανονικοποίηση των δεδομένων με τέτοιο τρόπο, ώστε η μέση τιμή τους να είναι μηδενική και η τυπική απόκλιση μοναδιαία. Αυτή η διαδικασία είναι απαραίτητη, καθώς δεδομένα διαφορετικής κλίμακας (τάξης μεγέθους) μπορεί να επηρεάζουν την εκπαίδευση του δικτύου σε μεγαλύτερο βαθμό από άλλα.
- Τέλος εφαρμόζεται η συνάρτηση `processpca()`, με την οποία μειώνεται το πλήθος των δεδομένων εισόδου, διαγράφοντας στοιχεία (τα οποία έχουν διασπορά μεγαλύτερη από κάποια τιμή) χωρίς να χάνεται σημαντική πληροφορία.

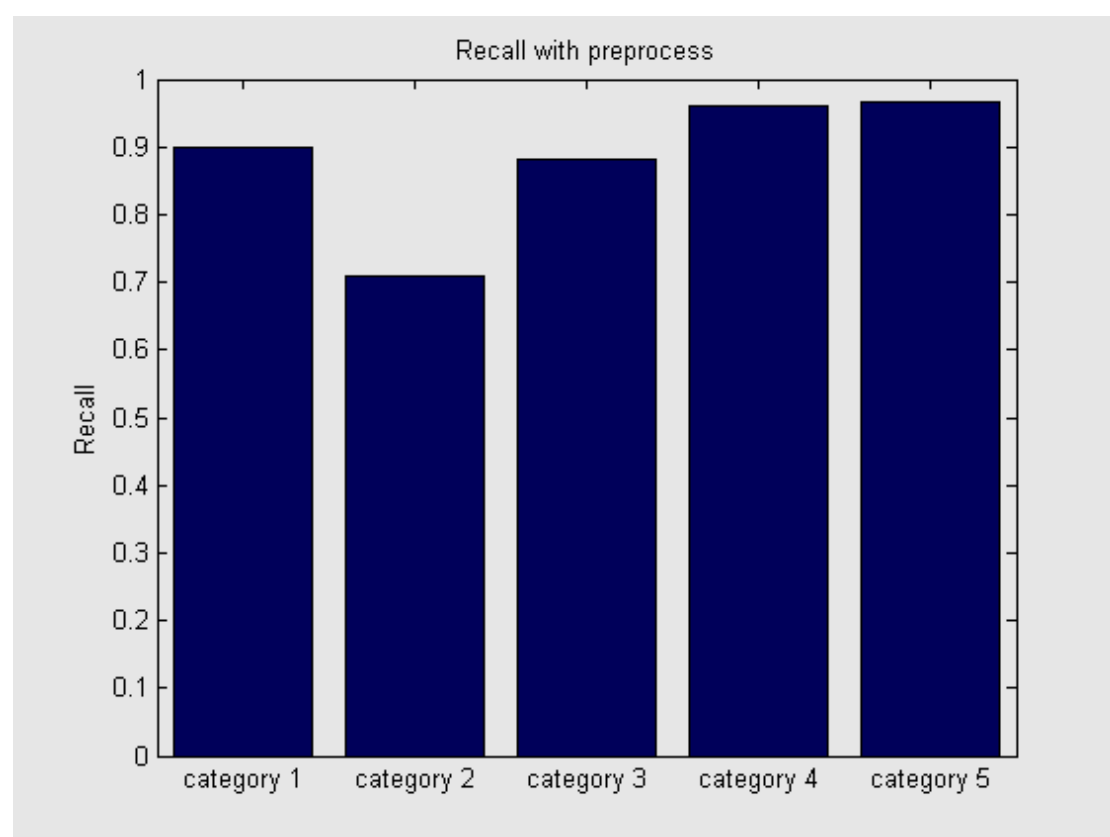
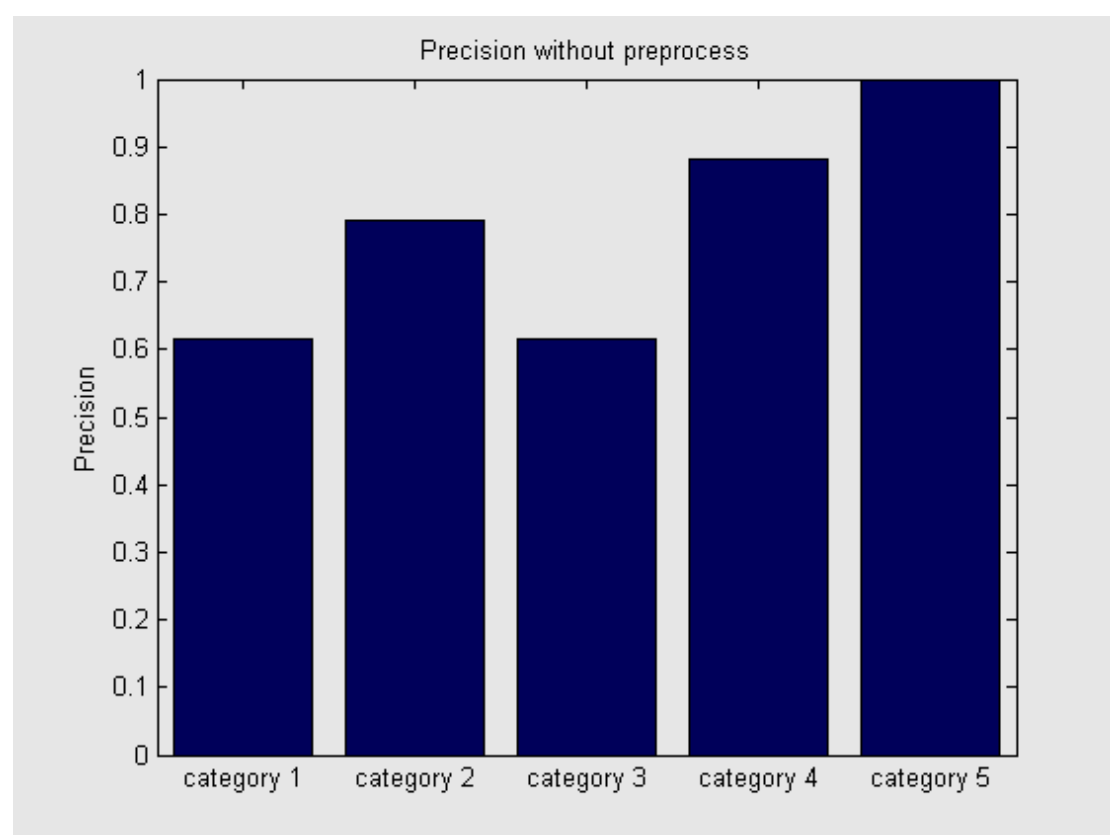
Οι παραπάνω μετασχηματισμοί εφαρμόζονται στον πίνακα TestData με χρήση του ορίσματος 'apply'.

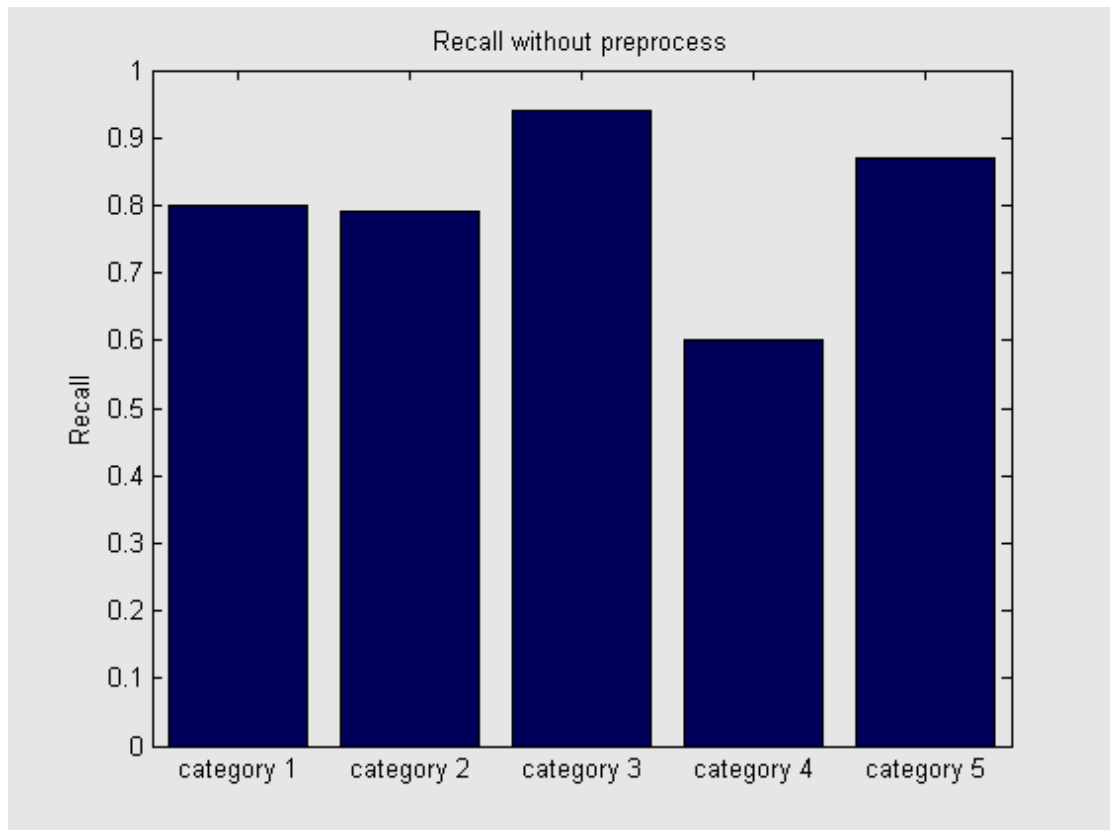
Αν εκπαιδευτεί το δίκτυο χωρίς να γίνει προεπεξεργασία, η απόδοση του είναι χειρότερη, καθώς τα δεδομένα εισόδου εμφανίζονται τυχαία και το πλήθος τους είναι διαφορετικό, με αποτέλεσμα η εκπαίδευση να γίνεται σε διαφορετικό βαθμό για κάθε κατηγορία. Επίσης λόγω πιθανής διαφοράς στην κλίμακα τους, ορισμένα δεδομένα μπορεί να επηρεάζουν το δίκτυο σε μεγαλύτερο βαθμό, ενώ λόγω του μεγάλου όγκου τους, η εκπαίδευση αργεί πολύ περισσότερο.

Ενδεικτικά για αρχιτεκτονική [15 15] (15 νευρώνες σε κάθε κρυφό επίπεδο), τα αποτελέσματα της απόδοσης του δικτύου παρουσιάζονται παρακάτω:

Για εκπαίδευση δικτύου με προεπεξεργασμένα δεδομένα έχουμε **accuracy = 0.8879**, ενώ χωρίς προεπεξεργασία είναι **accuracy = 0.7944**.







Παρατηρούμε ότι η απόδοση του δικτύου είναι καλύτερη για κάθε κατηγορία στην περίπτωση που έχει προηγηθεί προεπεξεργασία δεδομένων, όπως ήταν αναμενόμενο. Επίσης στην περίπτωση εκπαίδευσης χωρίς προεπεξεργασία δεδομένων η απόδοση σε ορισμένες κατηγορίες είναι αρκετά υψηλή, ενώ σε άλλες σχετικά χαμηλή.

Για όλα τα ερωτήματα, τα οποία ακολουθούν επισημαίνεται ότι η εκπαίδευση του δικτύου έγινε με προεπεξεργασμένα δεδομένα.

Ερώτημα 2 – Αρχιτεκτονική δικτύου και συνάρτηση εκπαίδευσης:

Η δημιουργία του νευρωνικού δικτύου γίνεται με χρήση της newff και τα δεδομένα εισόδου χωρίζονται σε 3 sets (training, validation και test set), όπως αναφέρεται στο βήμα 3. Ο διαχωρισμός αυτός εξυπηρετεί τη μέθοδο early stopping, με την οποία αποτρέπεται το φαινόμενο υπερεκπαίδευσης του δικτύου. Το training set χρησιμοποιείται κατά την εκπαίδευση του δικτύου για την ανανέωση των βαρών, το validation set ελέγχει το σφάλμα εξόδου κατά την εκπαίδευση, ενώ το test set δε χρησιμοποιείται κατά την εκπαίδευση, αλλά για την αξιολόγηση διαφορετικών καταστάσεων του δικτύου. Αν παρατηρηθεί ότι, ενώ το σφάλμα εξόδου του training set είναι πολύ μικρό, το αντίστοιχο σφάλμα της εξόδου του validation set μεγαλώνει, τότε υπάρχει ένδειξη ότι το δίκτυο αρχίζει να υπερεκπαιδεύεται και για να αποτραπεί αυτό το φαινόμενο, η εκπαίδευση σταματάει.

Σύμφωνα με την εκφώνηση, στη συγκεκριμένη περίπτωση το 80% των δεδομένων εισόδου αποτελούν το training set και 20% το validation set.

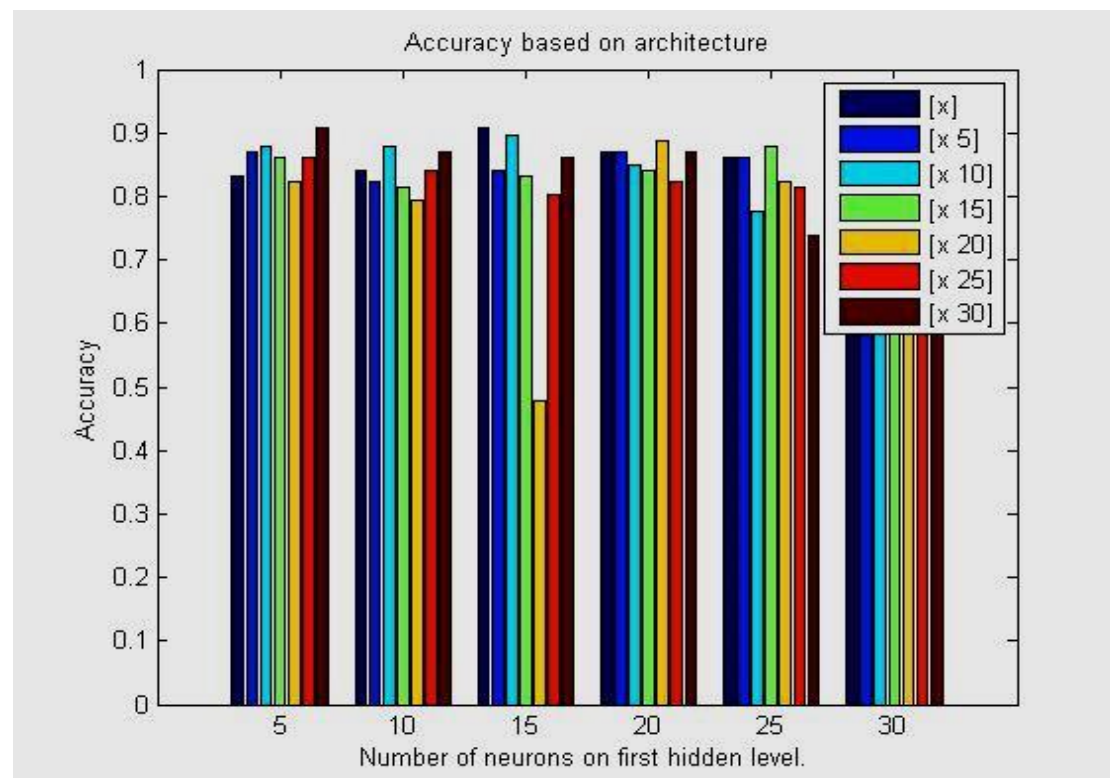
Επίσης πρέπει να σημειωθεί ότι η αξιολόγηση του δικτύου γίνεται με τη χρήση τριών παραμέτρων (accuracy, precision, recall), όπως αναφέρεται στην εκφώνηση της άσκησης. Οι δύο τελευταίες μετρικές αφορούν κάθε κατηγορία εικόνων και θα χρησιμοποιηθούν για τον εντοπισμό πιθανής ανισορροπίας στα αποτελέσματα εξόδου κάθε κατηγορίας ξεχωριστά. Η πρώτη μετρική αφορά τη γενική εικόνα απόδοσης του δικτύου.

Για την μελέτη του δικτύου με βάση την αρχιτεκτονική, χρησιμοποιήθηκαν τα scripts `find_architecture.m` και `find_architecture_random.m`. Στο πρώτο script γίνεται εκπαίδευση του δικτύου για όλες τις πιθανές αρχιτεκτονικές ενός και δύο επιπέδων, ενώ στη δεύτερη περίπτωση ο χρήστης μπορεί να εκπαιδεύσει το δίκτυο του για συγκεκριμένη αρχιτεκτονική. Τα ίδια scripts χρησιμοποιούνται επίσης για το βήμα 5 με τον ορισμό της συνάρτησης εκπαίδευσης μέσω της παραμέτρου `training`.

Ακολουθούν τα διαγράμματα accuracy όλων των πιθανών αρχιτεκτονικών, για κάθε συνάρτηση εκπαίδευσης. Αναφέρεται ότι η default επιλογή συνάρτησης εκπαίδευσης του δικτύου είναι η `trainlm`. Στο υπόμνημα των διαγραμμάτων αναγράφεται το πλήθος των νευρώνων στο δεύτερο κρυφό επίπεδο σε κάθε περίπτωση.

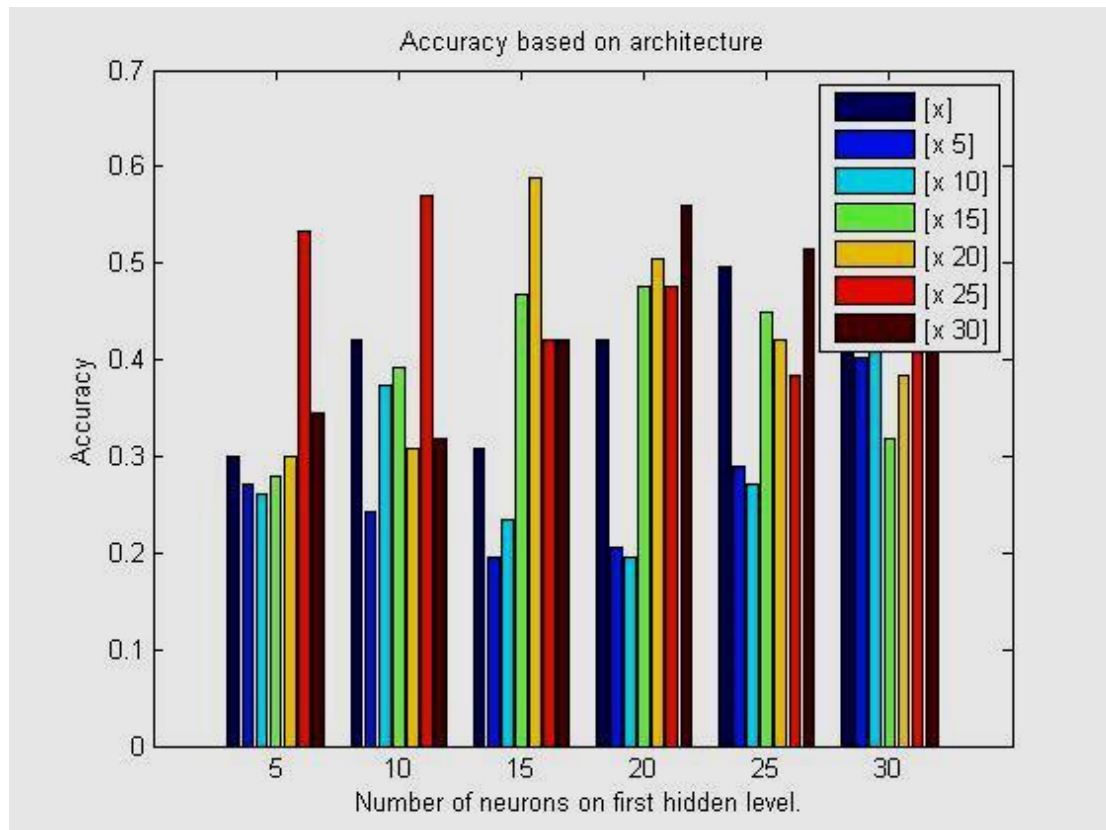
Trainlm:

Χρησιμοποιεί τον αλγόριθμο Levenberg-Marquardt, ο οποίος είναι συνδυασμός των μεθόδων Newton και gradient descent.



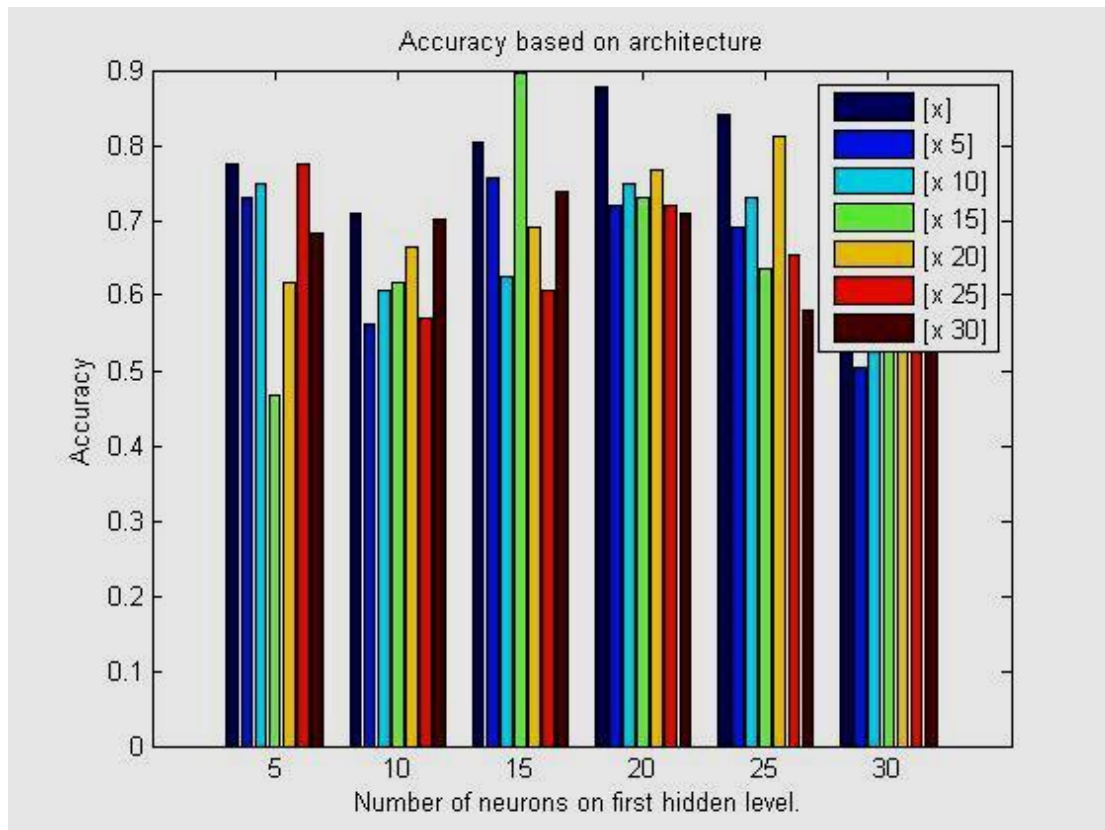
Traingd:

Χρησιμοποιεί τον αλγόριθμο gradient descent για την ανανέωση των βαρών του δικτύου με τέτοιο τρόπο, ώστε να μειώνεται το σφάλμα εξόδου.



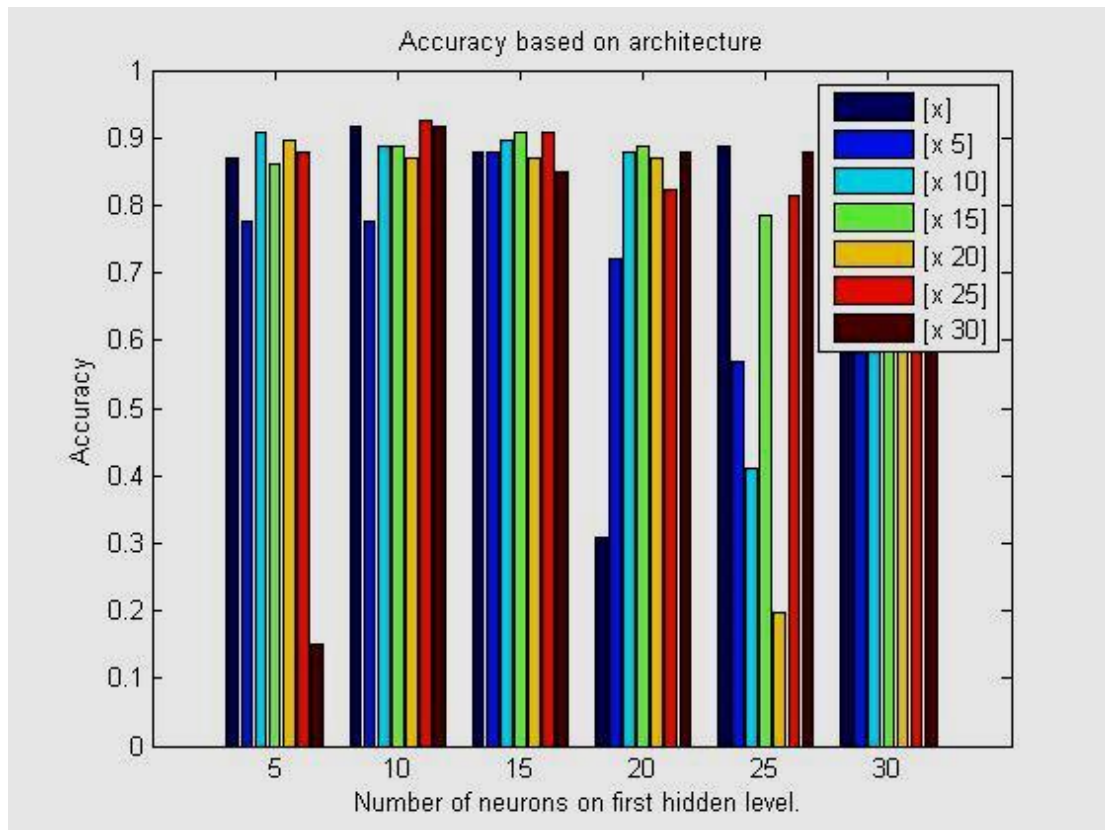
Traingda:

Όπως και στην προηγούμενη περίπτωση, η traingda χρησιμοποιεί τον αλγόριθμο gradient descent, αλλά με την προσαρμογή του learning rate κατά την εκπαίδευση. Αυτό σημαίνει ότι αποφεύγονται πολύ μεγάλες αλλαγές των βαρών, οι οποίες μπορεί να προκαλέσουν ταλάντωση της εξόδου γύρω από τη λύση και πολύ μικρές αλλαγές, οι οποίες μπορεί να καθυστερήσουν τη σύγκλιση του συστήματος σε μεγάλο βαθμό. Γενικά στα αρχικά στάδια της εκπαίδευσης μπορεί να χρησιμοποιείται μεγάλο learning rate για να πλησιάσει η έξοδος του δικτύου πιο γρήγορα στην επιθυμητή λύση και έπειτα το learning rate μπορεί να μειώνεται για να υπάρχει μεγαλύτερη ακρίβεια στη σύγκλιση του συστήματος. Επίσης με την προσαρμογή του learning rate κατά την εκπαίδευση του δικτύου, οι αρχικές τιμές που ορίζει ο χρήστης για το ρυθμό μάθησης θεωρητικά δεν επηρεάζουν πολύ τη παραπάνω διαδικασία.



Traingdx:

Η traingdx χρησιμοποιεί τον αλγόριθμο του gradient descent με προσθήκη ορμής και προσαρμόζόμενο learning rate. Η διαφορά της από την traingda είναι ότι μπορεί να αντιμετωπίσει τοπικά ελάχιστα, διαμορφώνοντας κατάλληλα τα βάρη του δικτύου με τον όρο της ορμής. Με αυτό τον τρόπο η εύρεση ολικού ελαχίστου μπορεί να γίνεται με μεγαλύτερη επιτυχία.



Παρατηρήσεις:

Η καλύτερη απόδοση του δικτύου ως προς το μέγεθος accuracy παρατηρείται για αρχιτεκτονική [15 15] με συνάρτηση εκπαίδευσης την `trainidx` (**ενδεικτικά accuracy = 0.9252**). Ακολουθεί η `trainlm` (**ενδεικτικά accuracy = 0.9065**) και η `trainda` (**ενδεικτικά accuracy = 0.8972**), ενώ τελευταία είναι η `traingd` (**ενδεικτικά accuracy = 0.6542**). Οι διαφορές της απόδοσης οφείλονται στα διαφορετικά χαρακτηριστικά της κάθε συνάρτησης εκπαίδευσης, τα οποία περιγράφηκαν παραπάνω. Επίσης η `traingd` παρουσιάζει τη χαμηλότερη επίδοση, καθώς δε χρησιμοποιεί learning rate και οι 1000 εποχές δεν αρκούν για τη σύγκλιση του δικτύου.

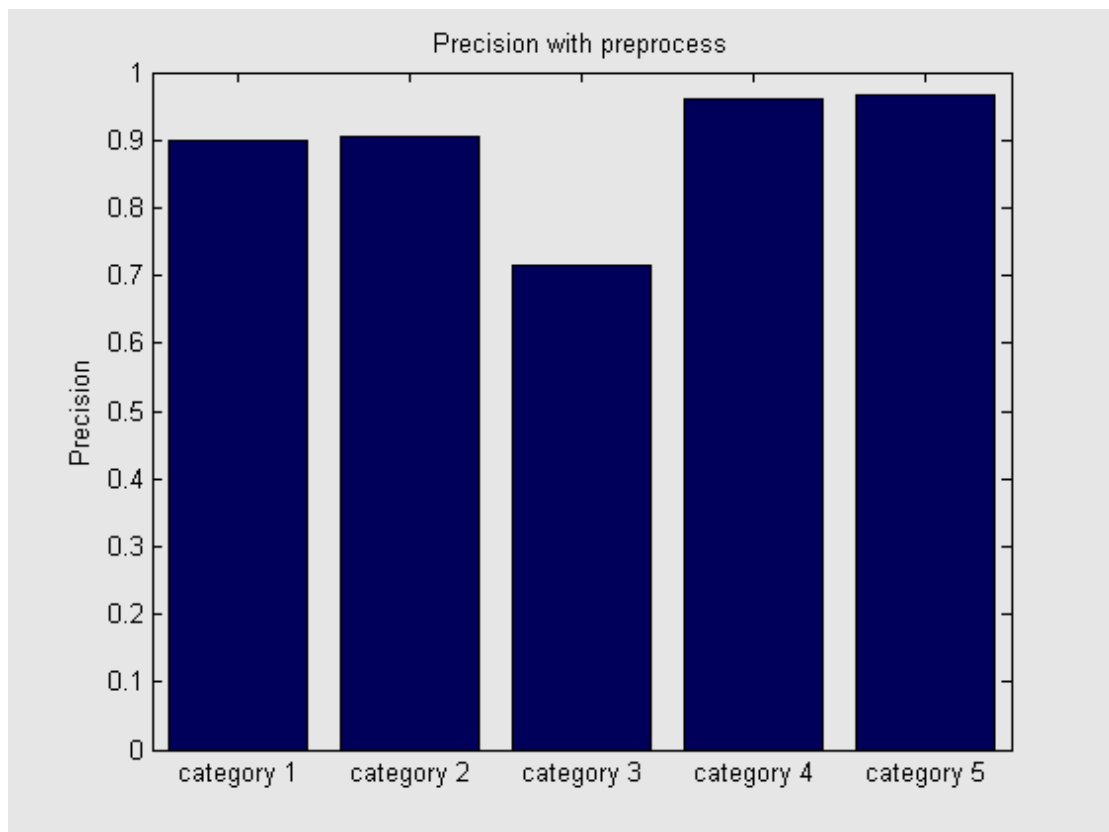
Σχετικά με την αρχιτεκτονική του δικτύου, οι μετρήσεις έγιναν περισσότερες από μία φορές και παρατηρήθηκαν ορισμένες αποκλίσεις στην απόδοση. Αυτό μπορεί να οφείλεται στο γεγονός ότι η αρχικοποίηση των βαρών είναι τυχαία κατά την εκπαίδευση του δικτύου. Γενικά οι αρχιτεκτονικές, για τις οποίες παρατηρήθηκε καλύτερη απόδοση είναι οι [10 15], [10 25] και [15 15], ενώ οι τιμές accuracy, οι οποίες αναγράφονται στο προηγούμενο σχόλιο είναι κοντά στη μέση τιμή των μετρήσεων που έγιναν.

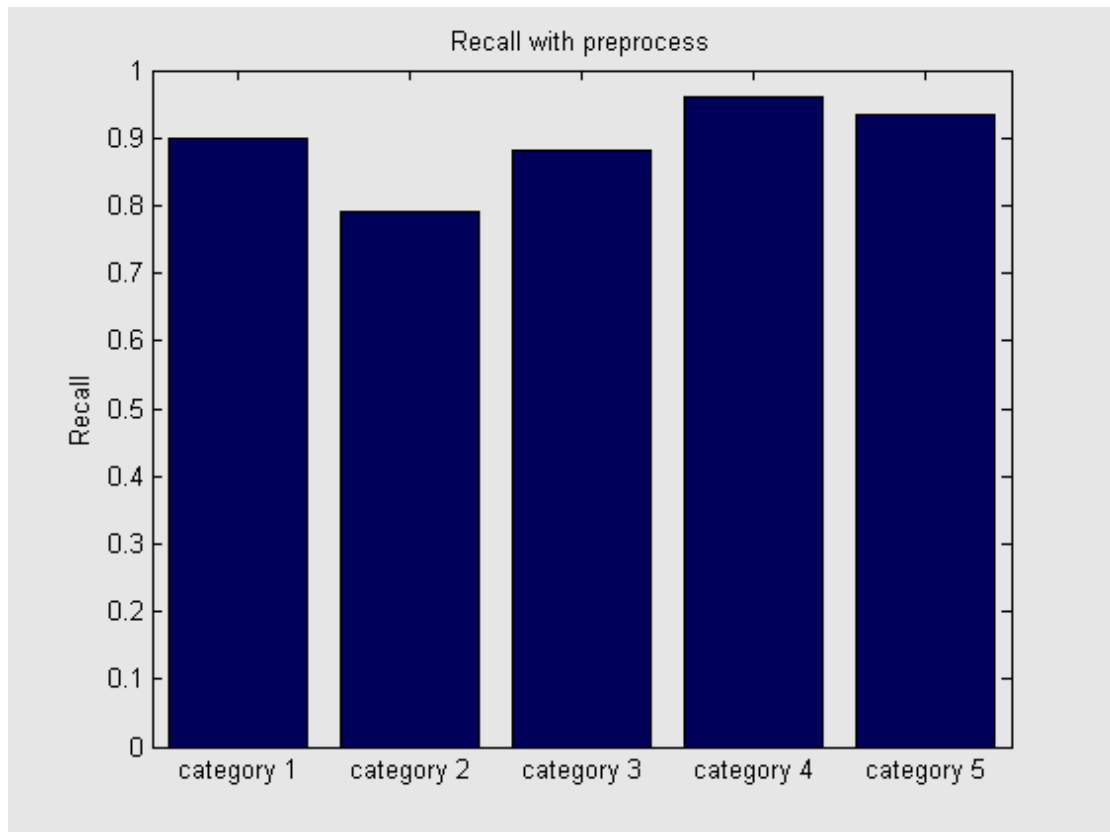
Επισημαίνεται ότι η απόδοση για δύο κρυφά επίπεδα με πλήθος νευρώνων μεγαλύτερο από 35 ήταν χειρότερη από τις υπόλοιπες αρχιτεκτονικές. Με άλλα λόγια χρήση περισσότερων νευρώνων στο δίκτυο δε συνεπάγεται καλύτερη απόδοση.

Σημείωση:

Επειδή τα παραπάνω διαγράμματα παρουσιάστηκαν για όλες τις πιθανές αρχιτεκτονικές του δικτύου, επιλέχθηκε οι εποχές για την μελέτη της `traingd` να είναι 1000. Ωστόσο παρουσιάζονται οι μετρήσεις, οι οποίες έγιναν για την αρχιτεκτονική δύο επιπέδων [15 15] με αριθμό εποχών 100000. Στα παρακάτω διαγράμματα η απόδοση της `traingd` είναι καλύτερη σε σύγκριση με την απόδοση για 1000 εποχές, καθώς η έξοδο του δικτύου πλησιάζει περισσότερο στη λύση.

accuracy = 0.8972





Επισημαίνεται ότι ούτε για 100000 εποχές δε συγκλίνει το δίκτυο με το συγκεκριμένο learn rate (0.2).

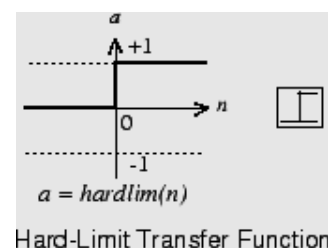
Για τα επόμενα ερωτήματα χρησιμοποιείται αρχιτεκτονική [15 15] με συνάρτηση εκπαίδευσης traingdx.

Ερώτημα 3 – Συνάρτηση ενεργοποίησης:

Οι συναρτήσεις ενεργοποίησης χρησιμοποιούνται για την αντιστοίχιση των δεδομένων εισόδου σε κάποια έξοδο. Μια συνάρτηση ενεργοποίησης μπορεί να είναι γραμμική ή μη γραμμική. Οι γραμμικές συναρτήσεις ενεργοποίησης είναι επιθυμητές όταν το δίκτυο αποτελείται από ένα επίπεδο νευρώνων, καθώς αν υπάρχουν περισσότερα επίπεδα, αυτά δεν επηρεάζουν την έξοδο αν εφαρμοστεί η γραμμική συνάρτηση στο πρώτο επίπεδο του δικτύου. Οι συναρτήσεις, οι οποίες εξετάζονται στο συγκεκριμένο ερώτημα παρουσιάζονται παρακάτω:

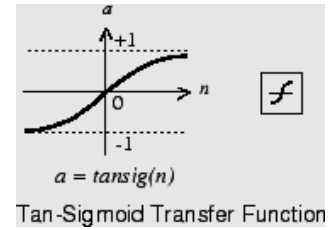
Hardlim:

$$\text{hardlim}(n) = \begin{cases} 1 & \text{αν } n \geq 0 \\ 0 & \text{διαφορετικά} \end{cases}$$



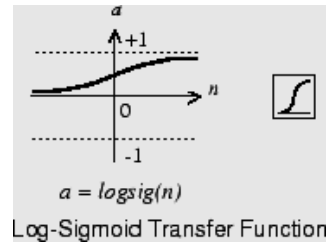
Tansig:

$$tansig(n) = \frac{2}{(1 + e^{-2n})} - 1$$



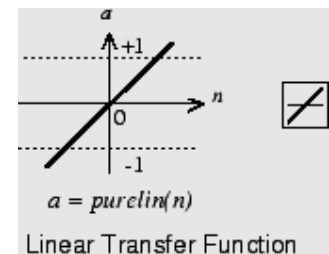
Logsig:

$$logsig(n) = \frac{1}{1 + e^{-n}}$$

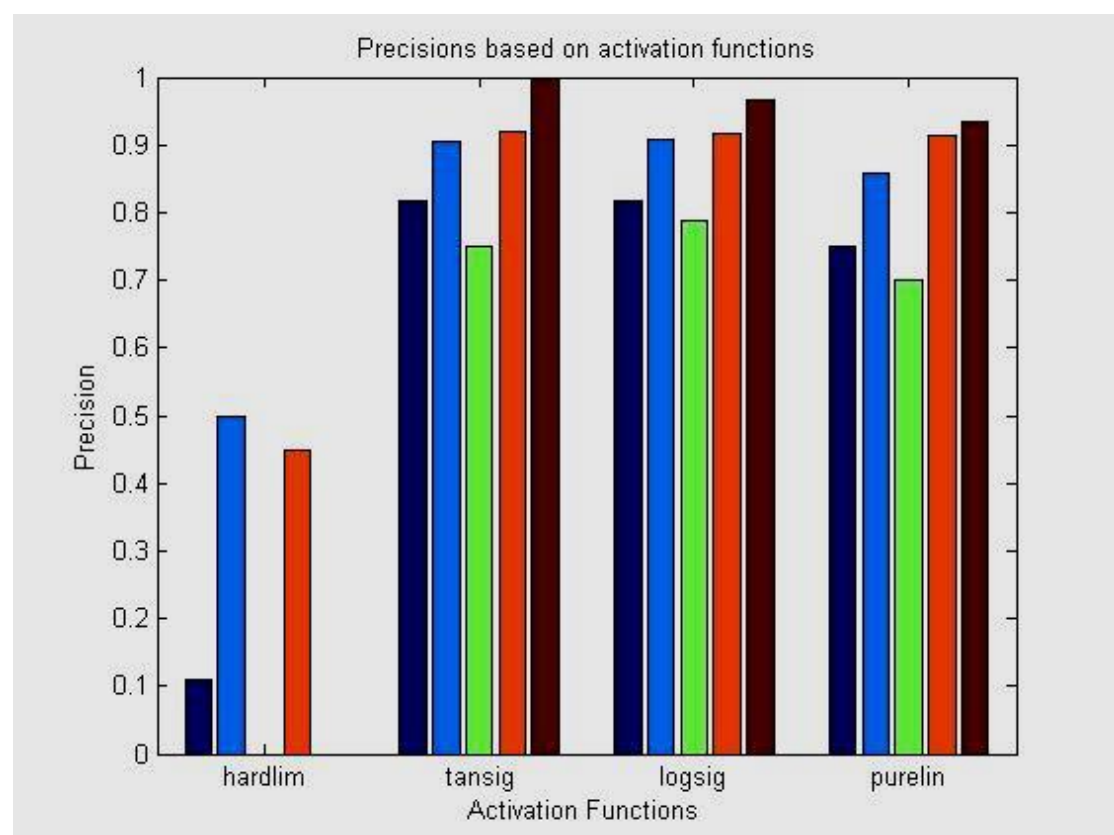
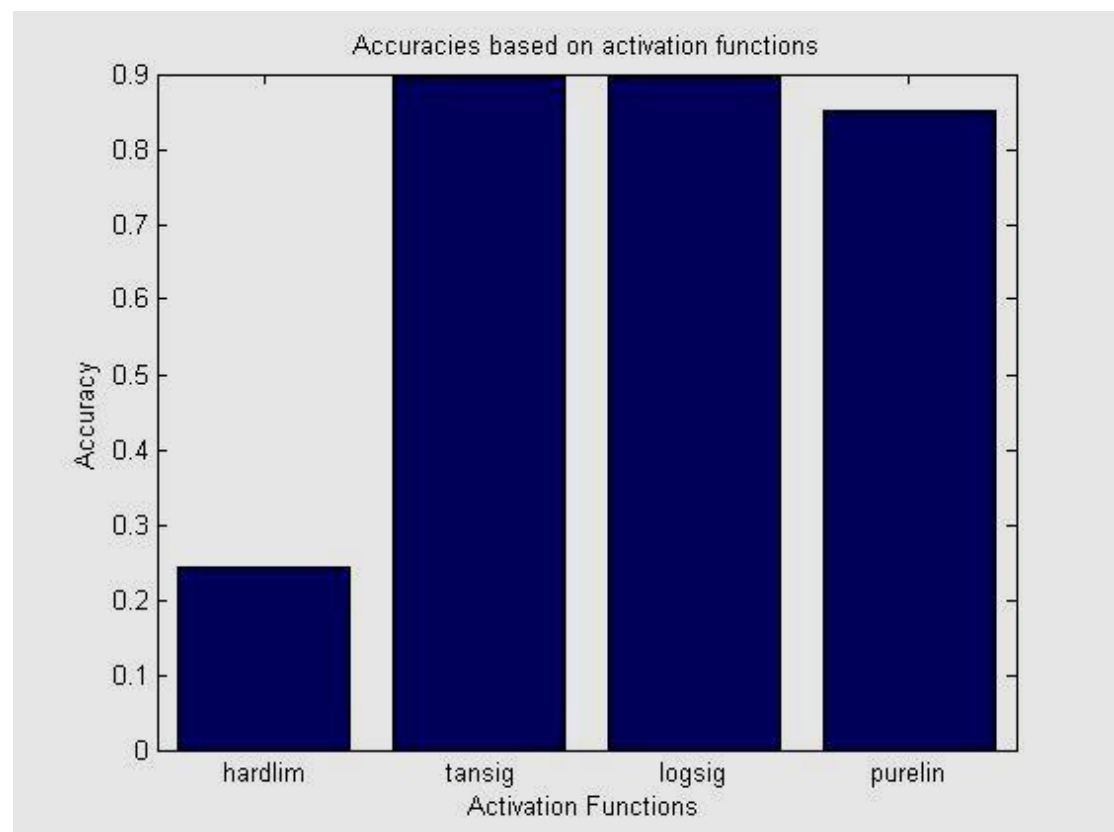


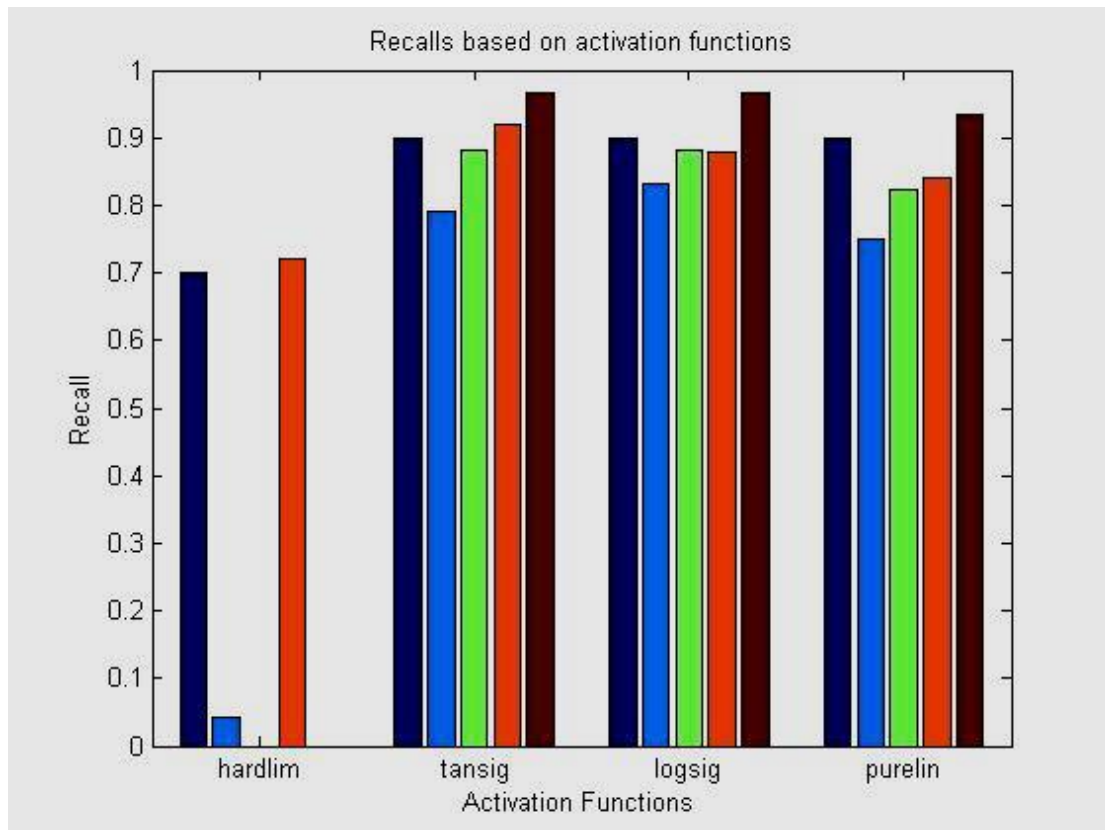
Purelin:

$$purelin(n) = n$$



Οι μετρήσεις της απόδοσης του δικτύου σύμφωνα με τις παραμέτρους accuracy, precision και recall παραθέτονται παρακάτω. Τα αντίστοιχα scripts, τα οποία χρησιμοποιήθηκαν είναι τα find_activation.m και find_all_activation.m.





Παρατηρήσεις:

Υψηλότερες επιδόσεις παρατηρούνται για τις συναρτήσεις tansig, logsig, purelin, ενώ την χαμηλότερη απόδοση παρουσιάζει η hardlim. Οι tansig και purelin έχουν υψηλές επιδόσεις, καθώς το διάστημα ενεργοποίησης τους είναι το $[-1, 1]$, το οποίο συμπίπτει με το διάστημα στο οποίο κανονικοποιεί η newff τα δεδομένα. Η hardlim και logsig έχουν διάστημα ενεργοποίησης το $[0, 1]$ με αποτέλεσμα ένας νευρώνας να θεωρείται ανενεργός στην τιμή 0 αντί για -1. Γενικά για την αναγνώριση εικόνων συνήθως χρησιμοποιούνται οι tansig, logsig για να προσδίδουν μη γραμμικότητα στο σύστημα, ενώ μπορεί να χρησιμοποιηθεί επίσης η συνάρτηση purelin, αλλά μόνο στο τελευταίο επίπεδο νευρώνων (επίπεδο εξόδου).

Η εκτέλεση των μετρήσεων έγινε αρκετές φορές και σε ορισμένες περιπτώσεις η logsig παρουσίαζε την υψηλότερη απόδοση, ενώ σε άλλες τη χαμηλότερη. Οι tansig και purelin είχαν σταθερά υψηλές αποδόσεις και η hardlim σχετικά χαμηλή. Οι διαφορές, οι οποίες παρατηρήθηκαν μπορεί να οφείλονται στην τυχαία αρχικοποίηση των βαρών του δικτύου.

Σημειώνεται ότι αν η κανονικοποίηση της newff μπορεί να γίνει στο διάστημα $[0, 1]$ αντί για $[-1, 1]$, ίσως η απόδοση των hardlim και logsig να είναι καλύτερη.

Ερώτημα 4 – Σύγκριση αλγορίθμων μάθησης:

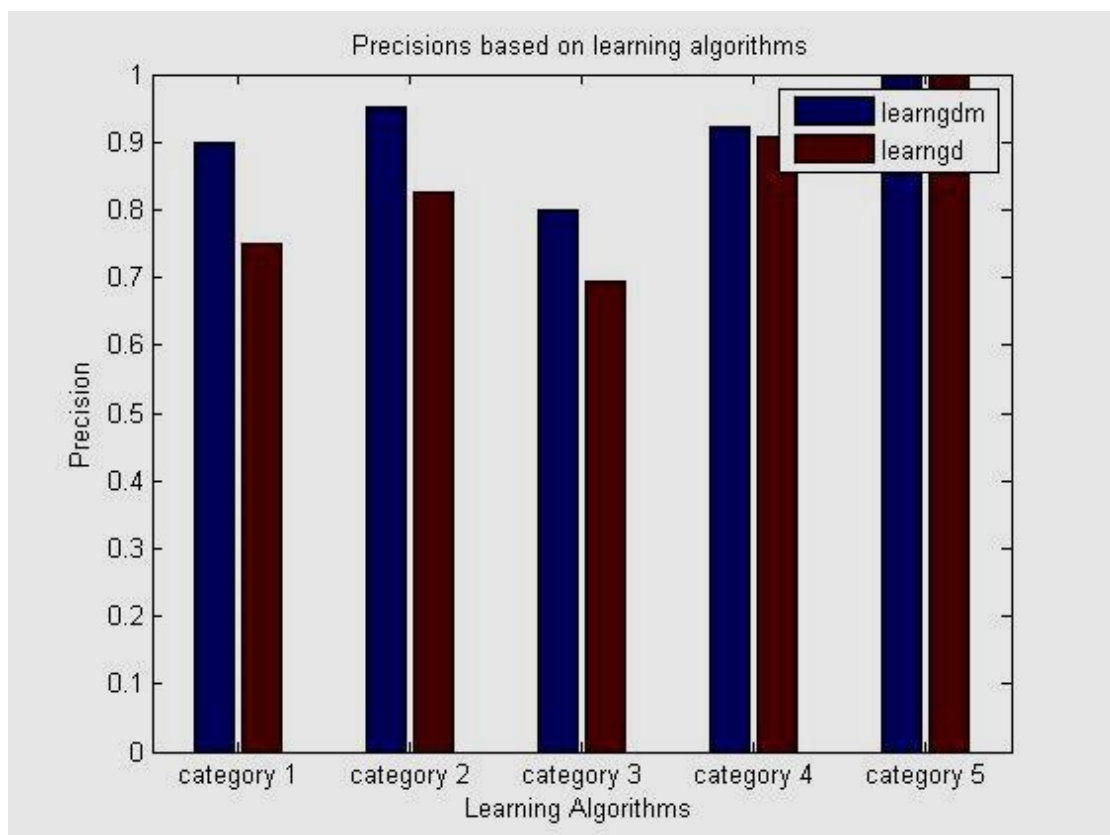
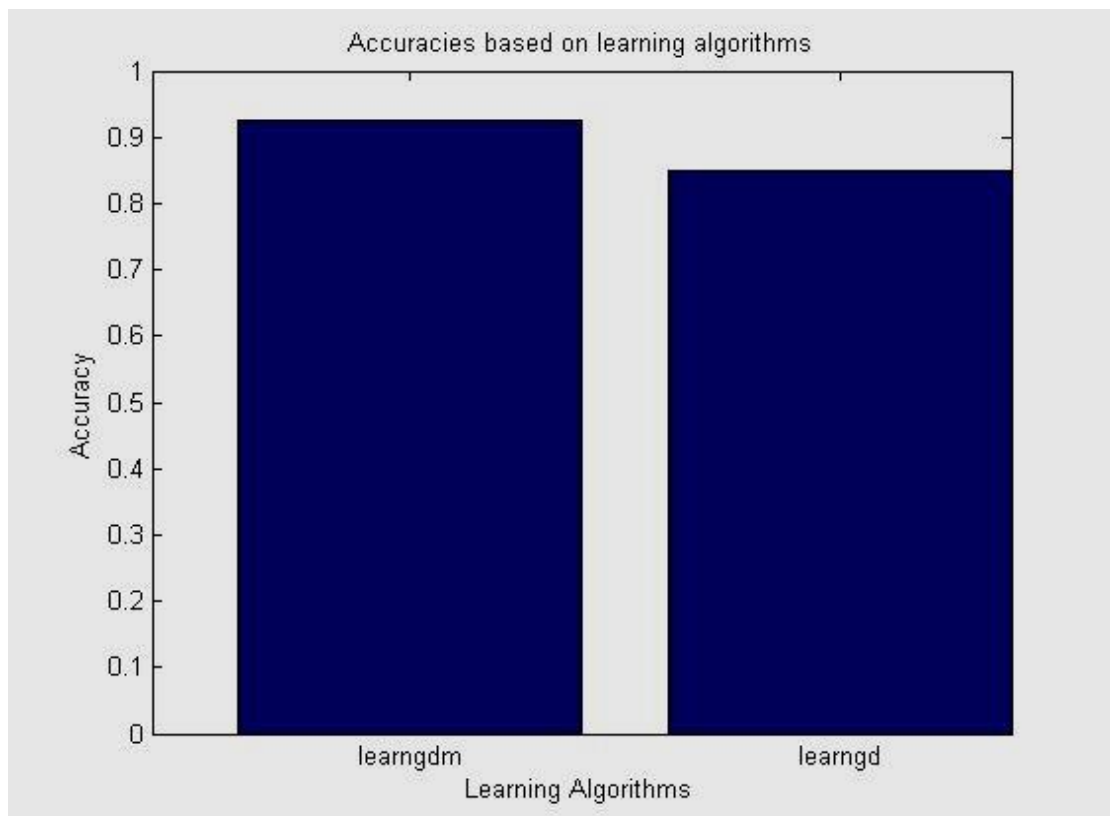
Οι αλγόριθμοι μάθησης είναι υπεύθυνοι για τον τρόπο με τον οποίο ανανεώνονται τα βάρη του δικτύου σε μια επανάληψη. Υπάρχουν τρία είδη μάθησης:

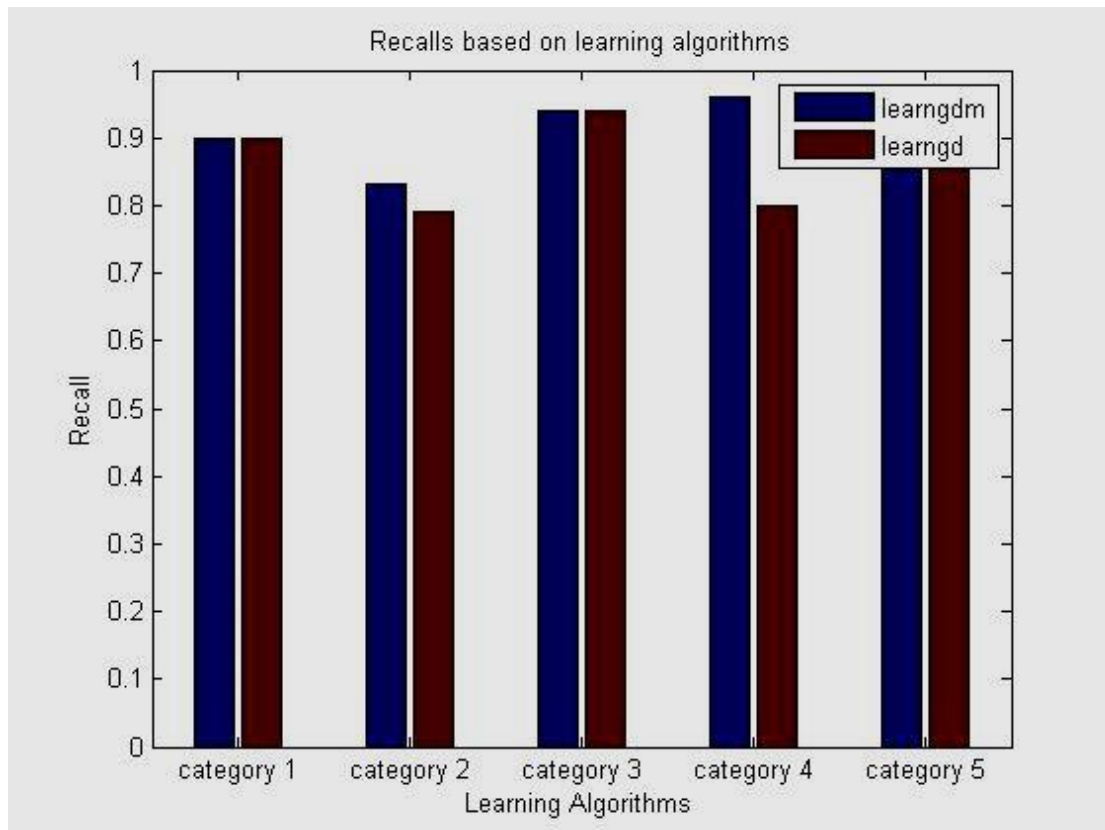
- **Supervised learning:** Το επιθυμητό αποτέλεσμα εξόδου διατίθεται στο δίκτυο μαζί με τα δεδομένα εισόδου για την εκπαίδευση του δικτύου. Κατά την εκπαίδευση, το δίκτυο ελέγχει το σφάλμα εξόδου και είναι σε θέση να διαμορφώσει κατάλληλα τα βάρη για τις επόμενες επαναλήψεις προκειμένου να πλησιάσει την επιθυμητή έξοδο.
- **Unsupervised learning:** Σε αυτή την περίπτωση δε διατίθεται η επιθυμητή έξοδος στο δίκτυο, οπότε το ίδιο το δίκτυο καλείται να αναπτύξει κάποιο πρότυπο για τη διαμόρφωση των βαρών και τη λειτουργία του.
- **Reinforcement learning:** Αυτή η κατηγορία εμφανίζει ομοιότητες με το supervised learning με τη διαφορά ότι δεν παρέχεται στο δίκτυο η επιθυμητή έξοδος, αλλά μετά από κάθε επανάληψη εκπαίδευσης το δίκτυο «επιβραβεύεται» με βάση το βαθμό, στον οποίο η έξοδος του πλησιάζει την επιθυμητή.

Στην άσκηση ζητείται η μελέτη των επιδόσεων δύο αλγορίθμων μάθησης (learngd, learngdm), οι οποίοι ανήκουν στην πρώτη κατηγορία.

Οι `learngd` , `learngdm` βασίζονται στον αλγόριθμο απότομης καθόδου (εύρεσης ολικού ελαχίστου σφάλματος) με τη διαφορά ότι ο `learngdm` στηρίζεται στην προσθήκη όρου ορμής. Στη δεύτερη περίπτωση τα βάρη διαμορφώνονται με βάση κάποιο όρο ορμής, ο οποίος επιτρέπει την αντιμετώπιση/ υπερπήδηση των τοπικών ελαχίστων.

Για την εκτέλεση των μετρήσεων χρησιμοποιείται το script `find_learning.m` και η απόδοση του δικτύου παρουσιάζεται παρακάτω:



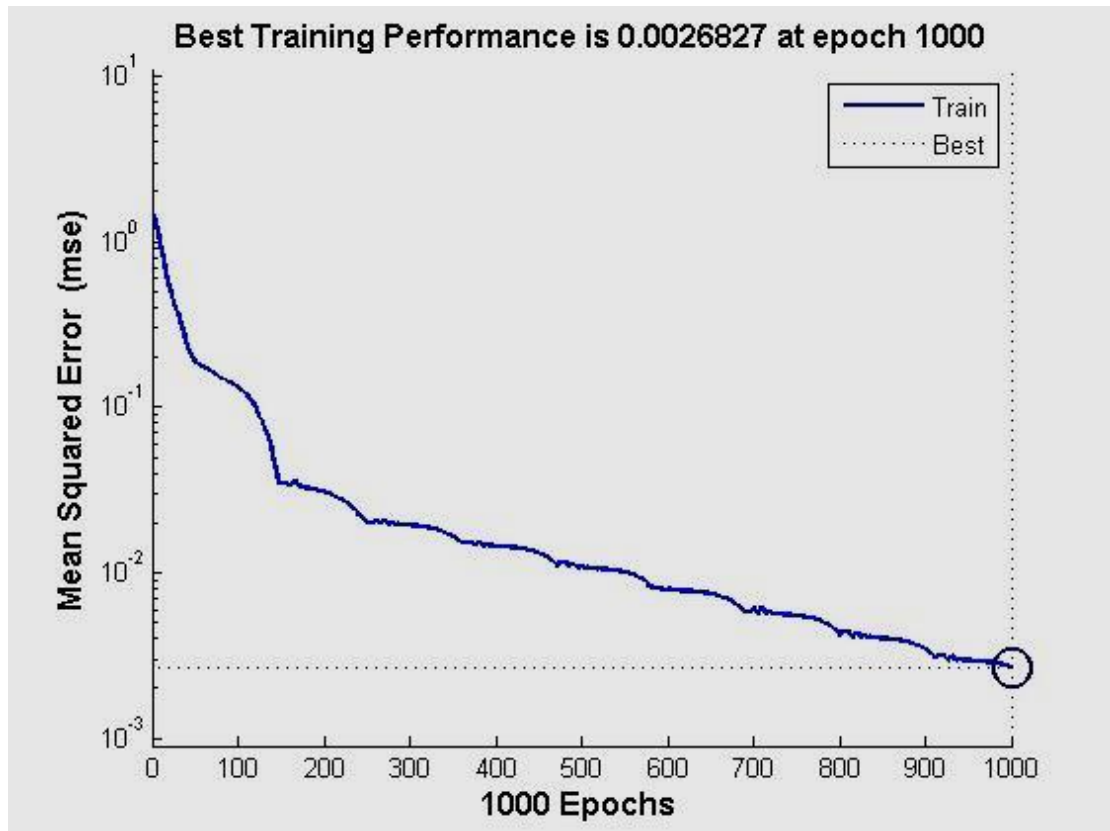
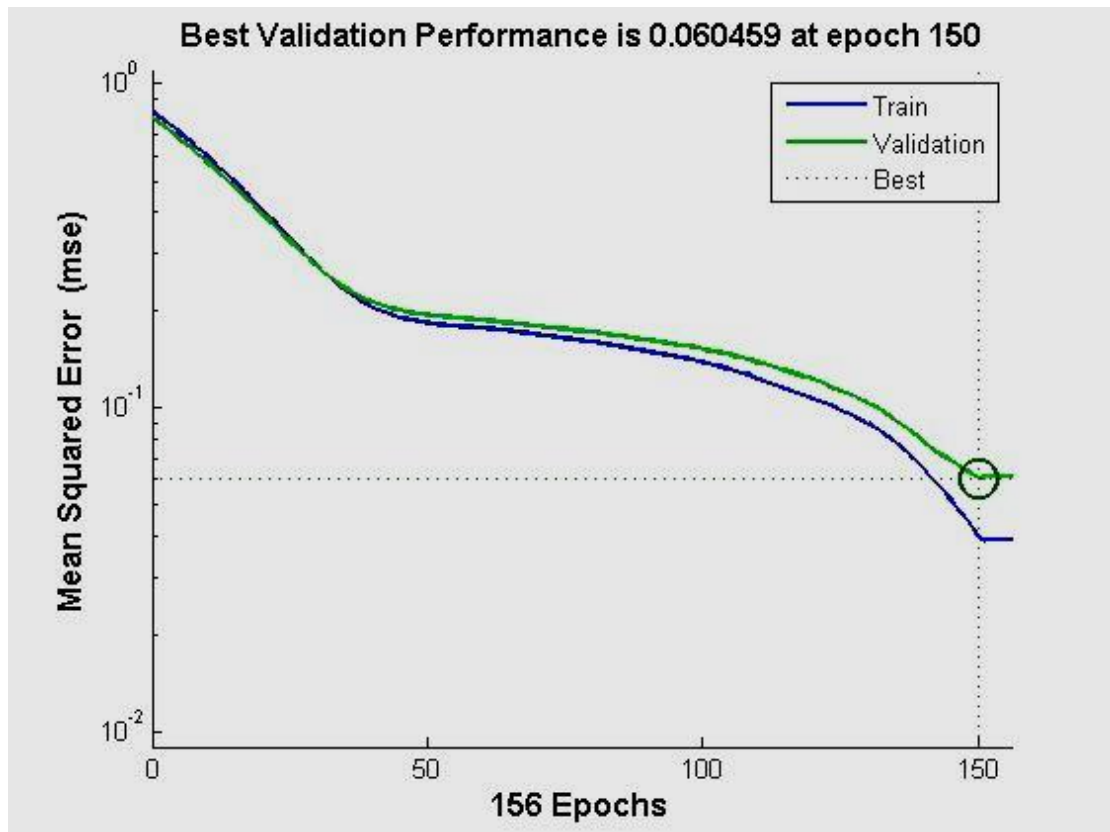


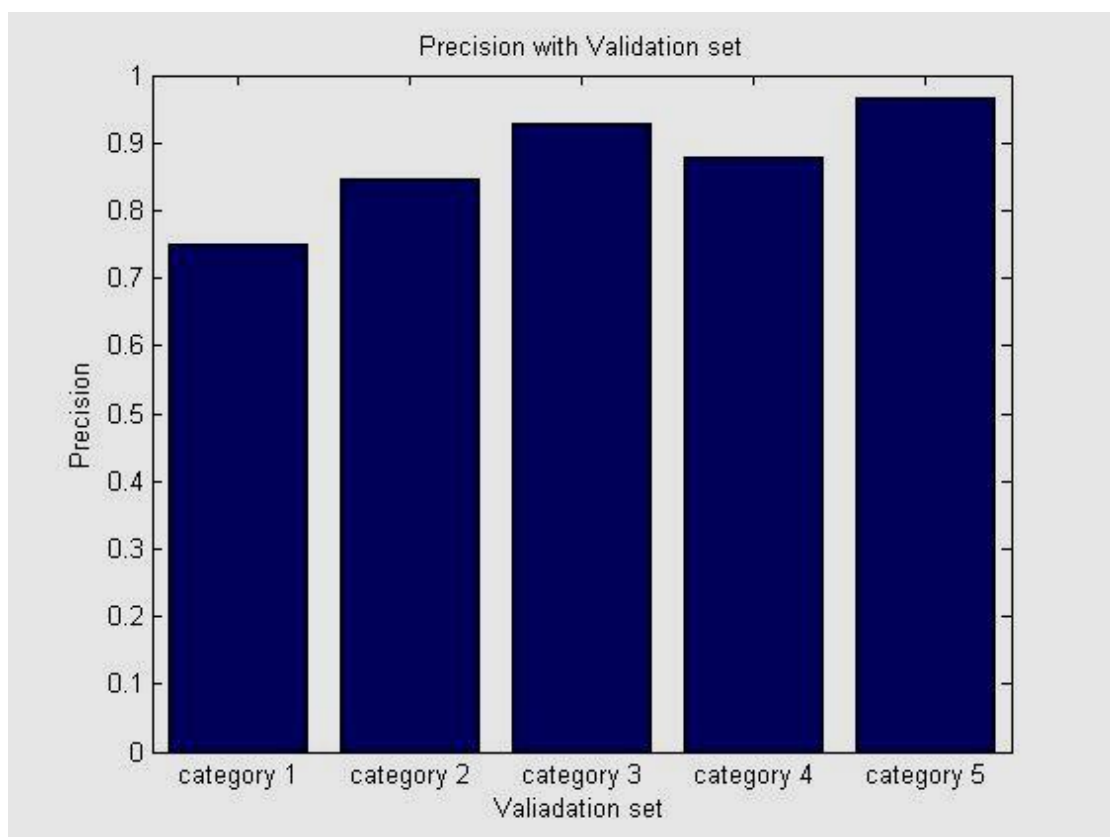
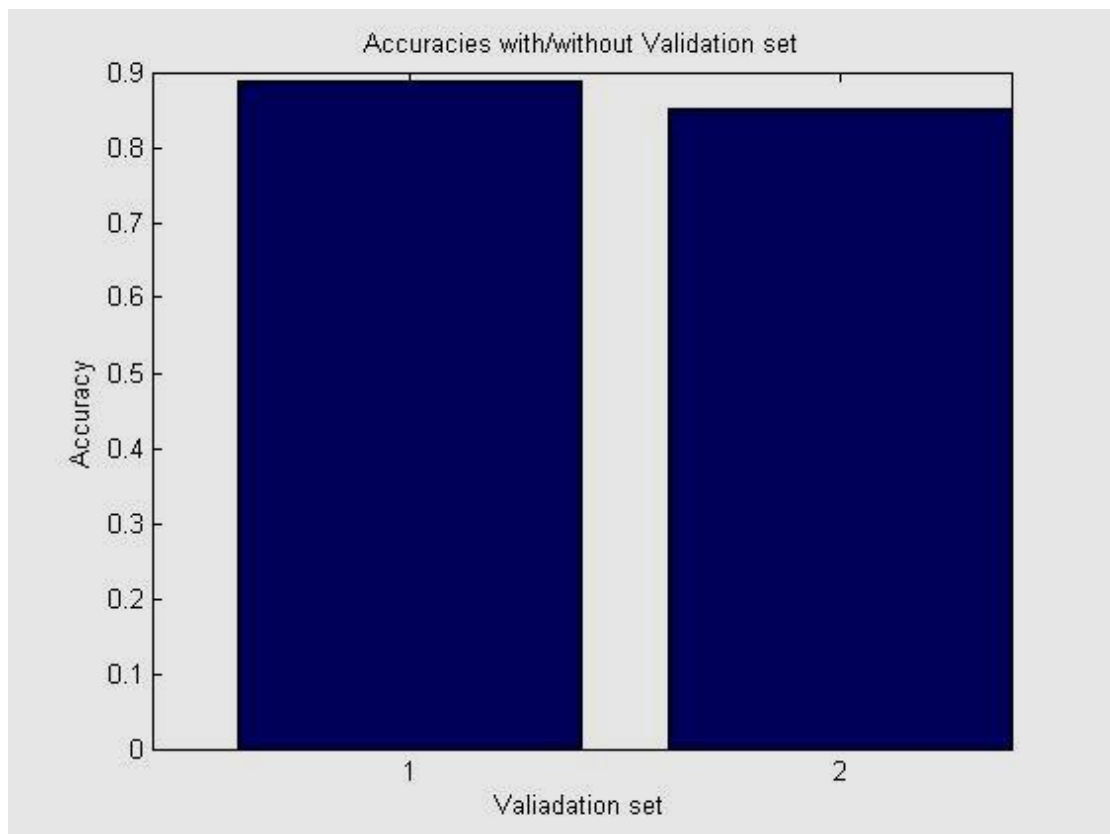
Ο αλγόριθμος μάθησης απότομης καθόδου με προσθήκη του όρου ορμής παρουσιάζει καλύτερη απόδοση, το οποίο είναι αναμενόμενο, καθώς υπάρχει η δυνατότητα αντιμετώπισης τοπικών ελαχίστων.

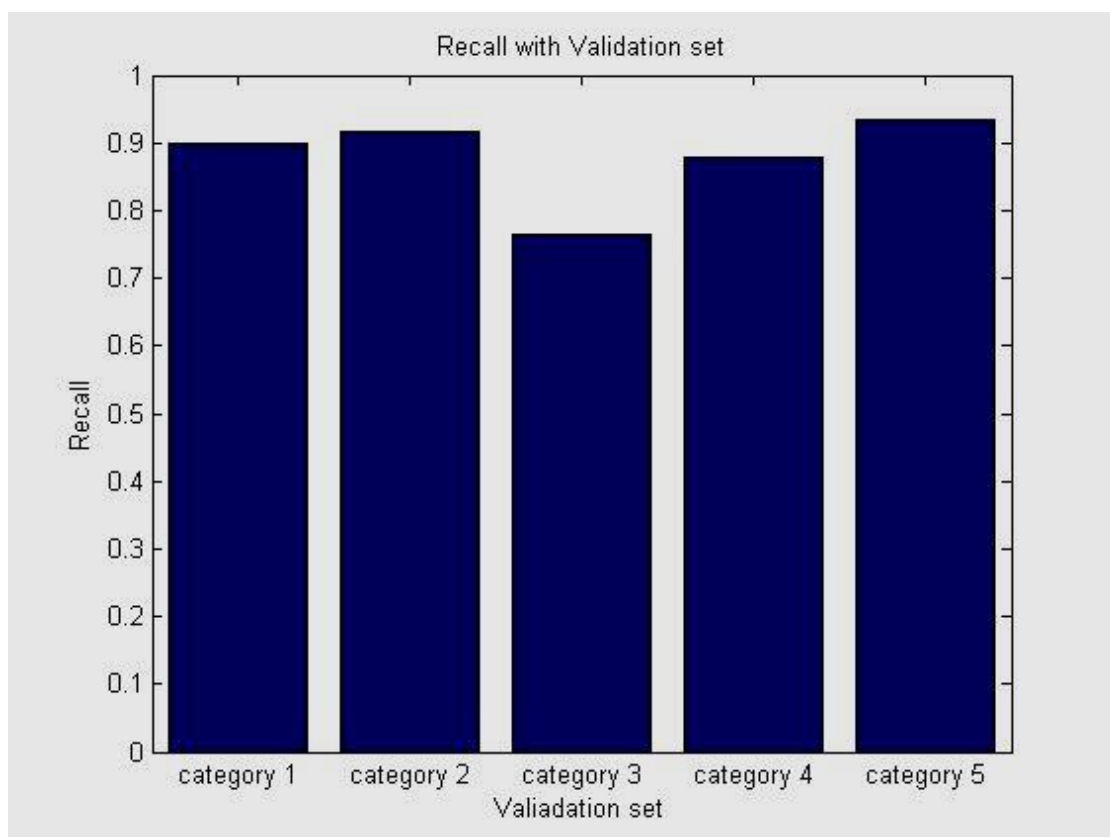
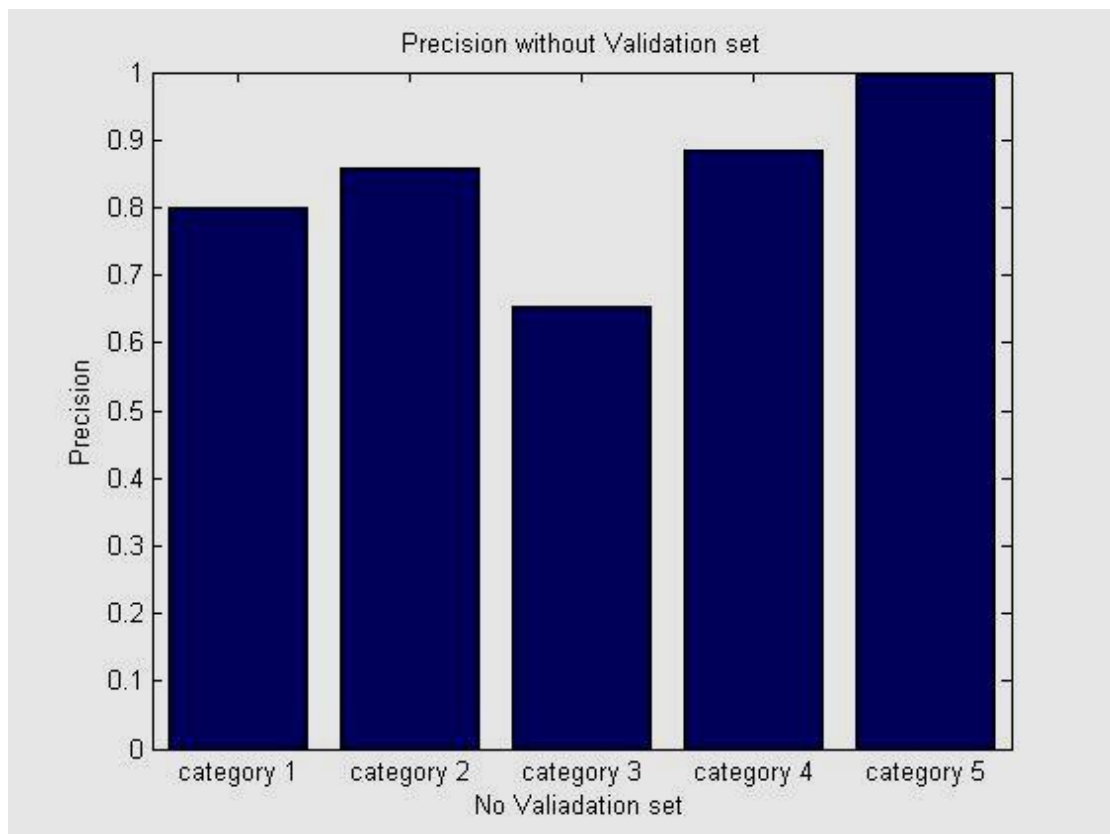
Ερώτημα 5 – Μέθοδος early stopping:

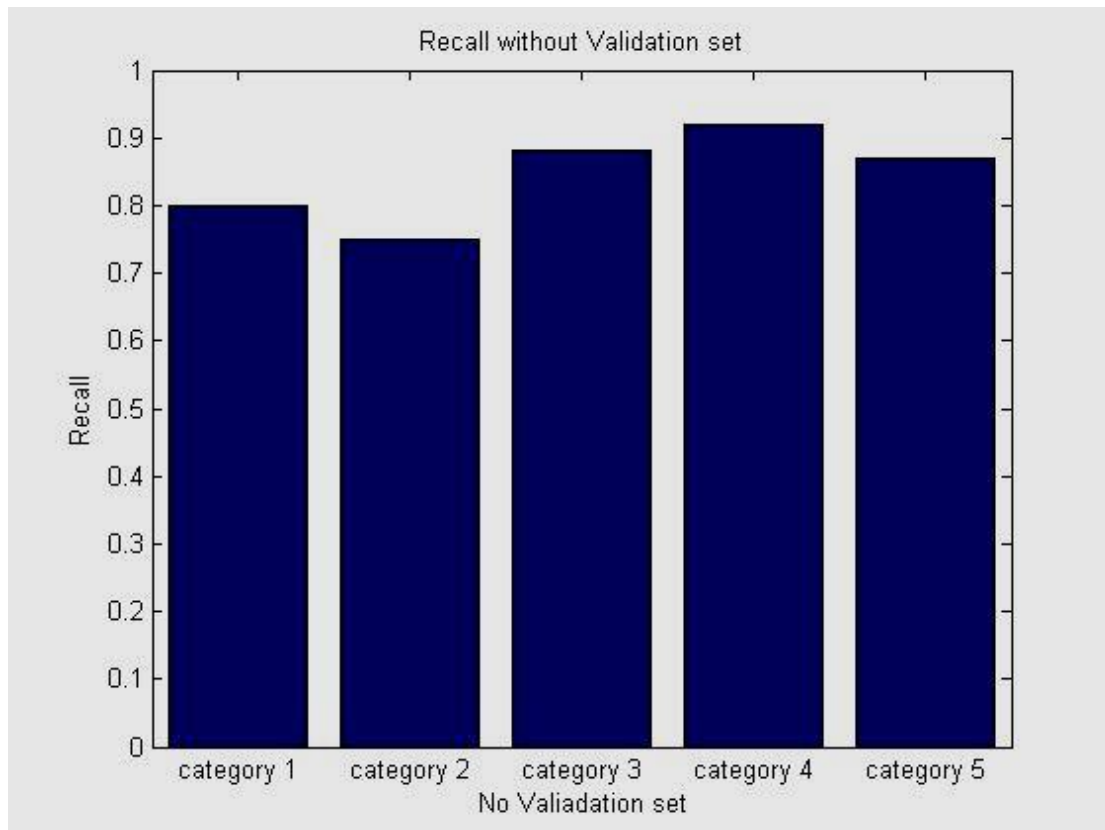
Η μέθοδος early stopping, όπως αναφέρθηκε στο ερώτημα 2, χρησιμοποιείται για την αντιμετώπιση της υπερεκπαίδευσης του δικτύου. Η υπερεκπαίδευση του δικτύου είναι επιθυμητό να αποφεύγεται, γιατί σε αυτή την περίπτωση το δίκτυο εξειδικεύεται στην αντιμετώπιση συγκεκριμένου συνόλου δεδομένων (χαμηλό σφάλμα στα train data), με αποτέλεσμα η απόδοση του να είναι χαμηλή όταν καλείται να ελέγξει διαφορετικό σύνολο δεδομένων εισόδου (υψηλό σφάλμα στα test data).

Για τις ζητούμενες μετρήσεις χρησιμοποιήθηκε στο script compare_val.m, ενώ έγινε σύγκριση της απόδοσης του δικτύου όταν χρησιμοποιείται μέθοδος early stopping και όταν όχι.









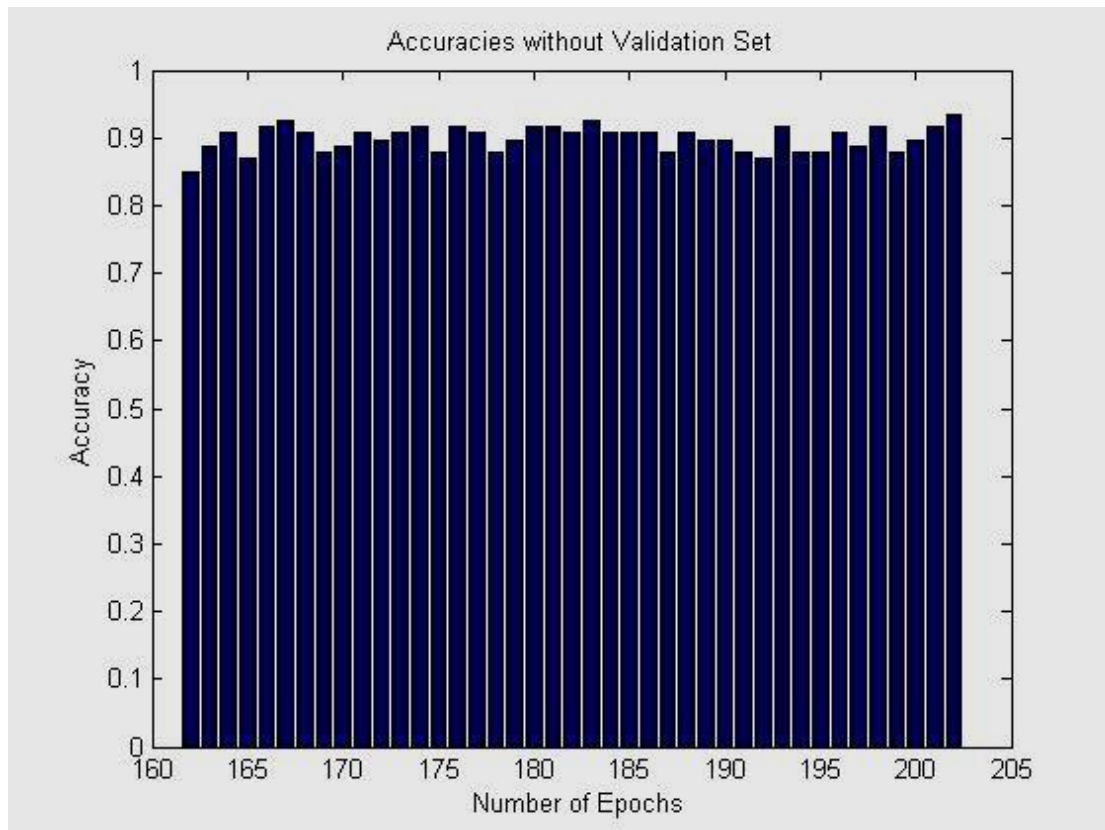
Χωρίς τη χρήση της μεθόδου early stopping, το δίκτυο εκπαιδεύεται για όλες τις εποχές που έχουν οριστεί (1000) , ενώ με τη χρήση της μεθόδου η εκπαίδευση σταματάει πολύ νωρίτερα (159).

Η απόδοση του δικτύου είναι καλύτερη στην περίπτωση της χρήσης validation set, το οποίο είναι αναμενόμενο.

Ερώτημα 6 – Μελέτη αριθμού εποχών:

Ο αριθμός των εποχών που χρησιμοποιείται για την εκπαίδευση του δικτύου σχετίζεται άμεσα με την εμφάνιση του φαινομένου overfitting (αν δεν υπάρχει validation set). Συγκεκριμένα αν ο αριθμός των εποχών είναι πολύ μικρός το δίκτυο ίσως να μην εκπαιδεύεται επαρκώς, ενώ αν είναι πολύ μεγάλος μπορεί να υπερεκπαιδεύεται.

Όταν ο αριθμός των εποχών είναι κοντά σε αυτόν, στον οποίο σταμάτησε το δίκτυο την εκπαίδευση όταν υπήρχε validation set, τότε η αντίστοιχη απόδοση είναι παρόμοια. Αυτό φαίνεται από το παρακάτω διάγραμμα του accuracy για διαφορετικό αριθμό εποχών.

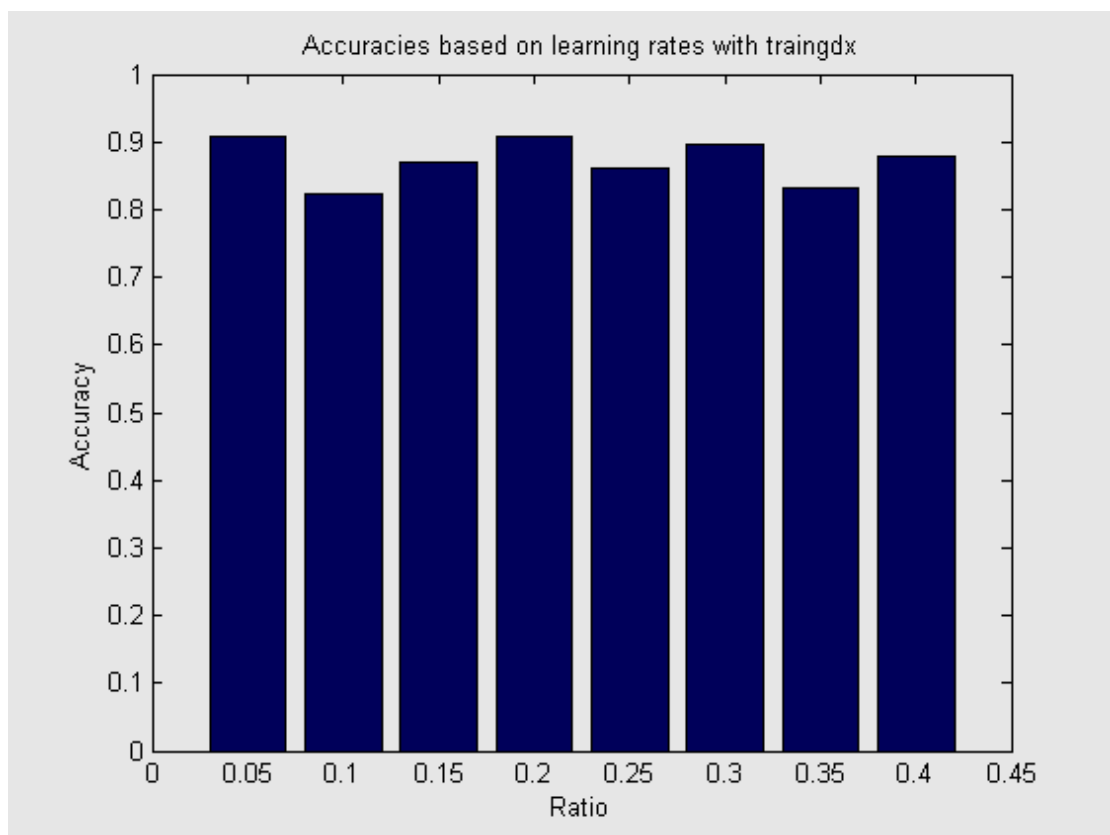
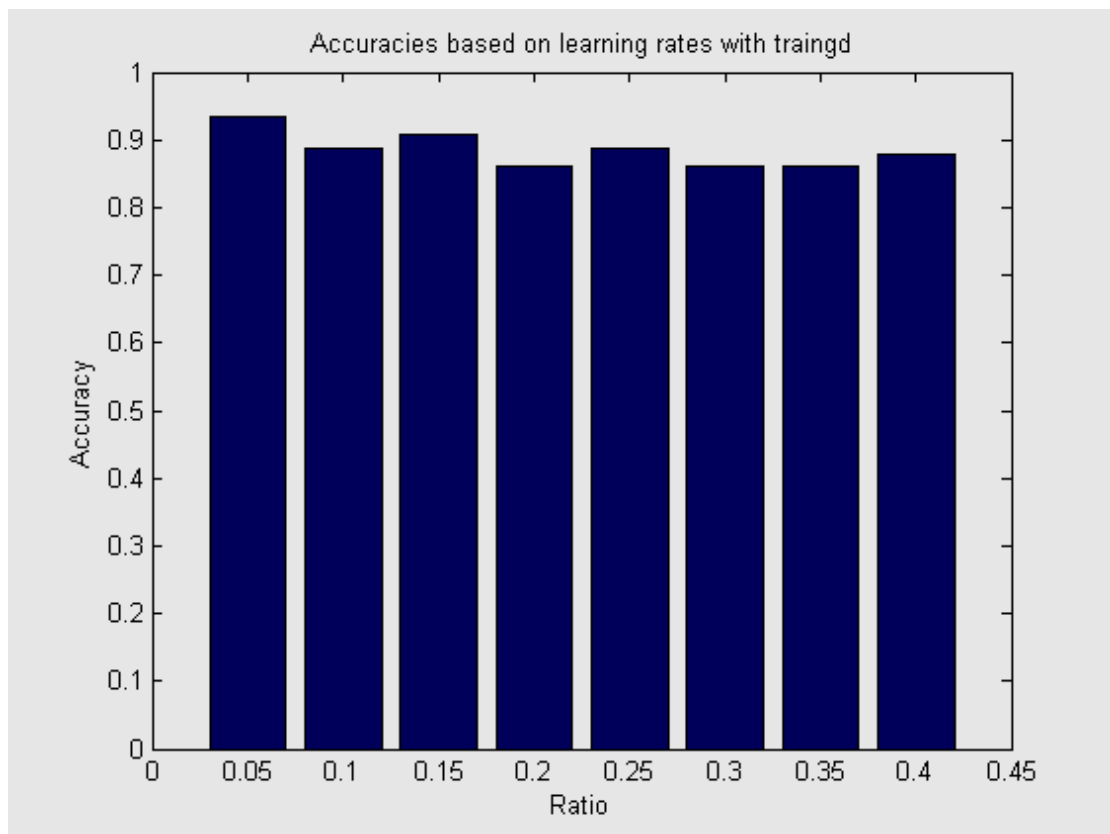


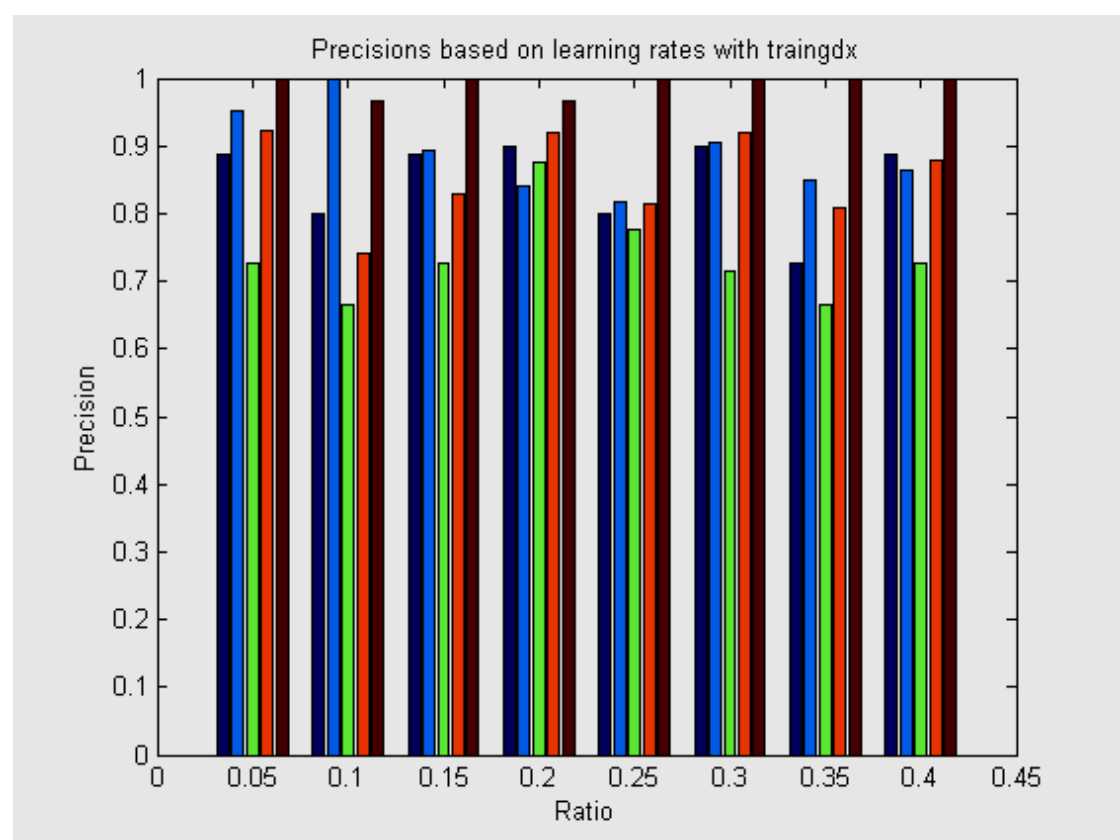
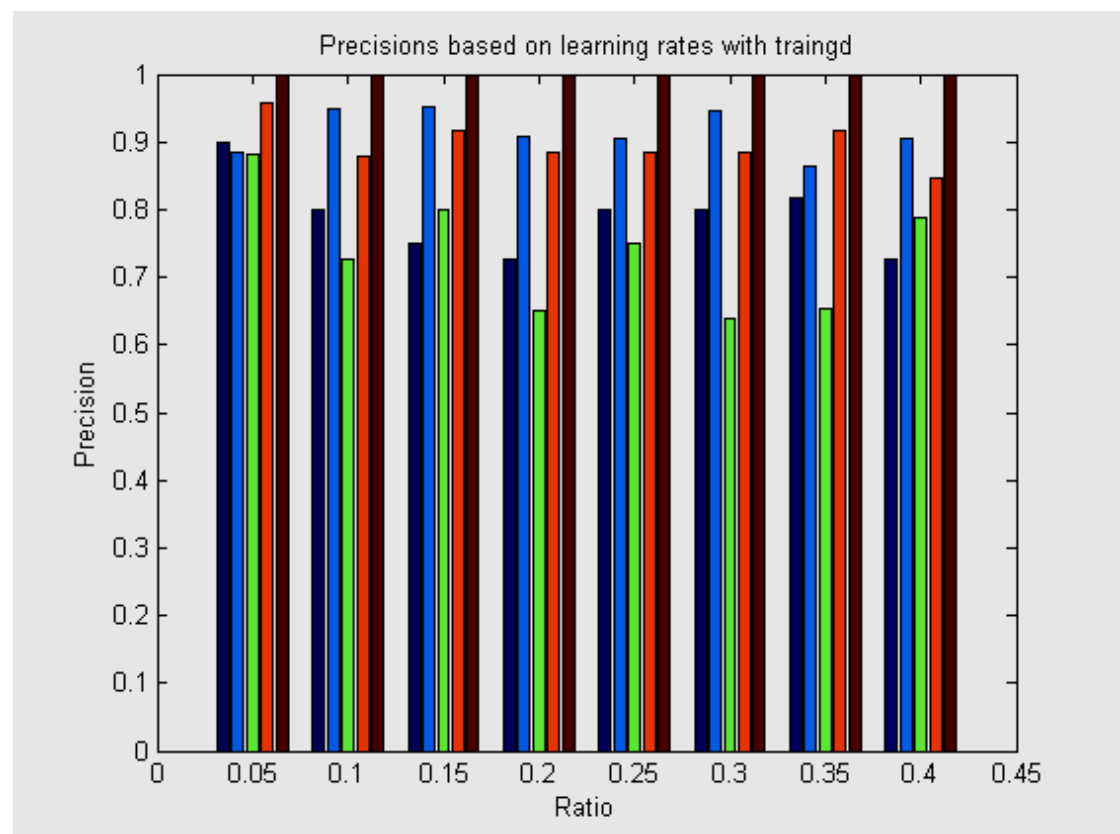
Ερώτημα 7 – Μελέτη ρυθμού μάθησης:

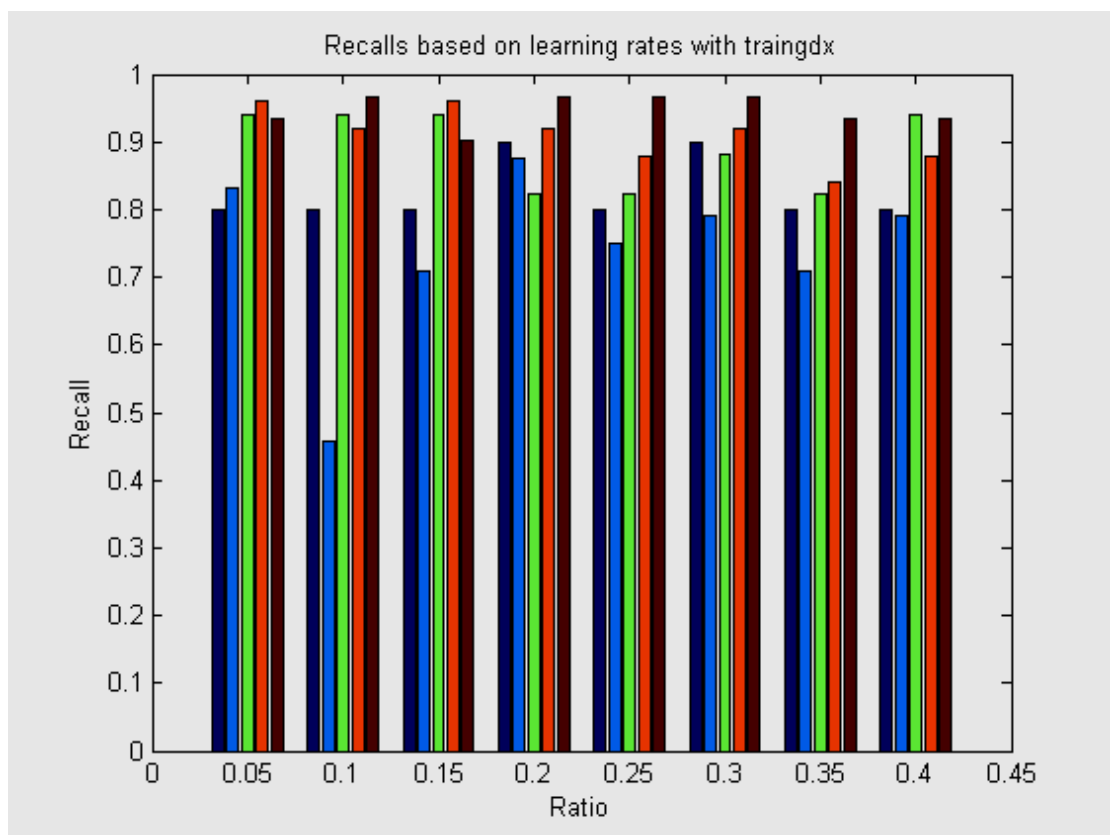
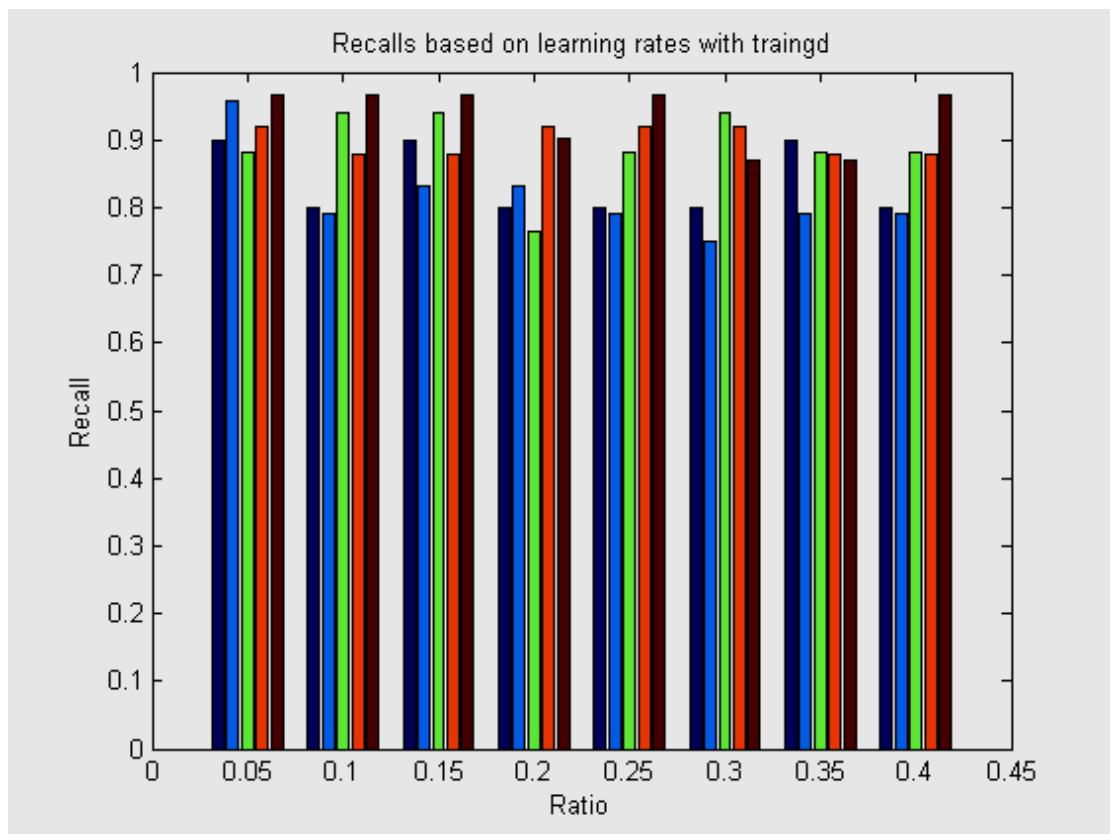
Ο ρυθμός μάθησης (learning rate) επηρεάζει άμεσα τον τρόπο με τον οποίο το δίκτυο προσεγγίζει την επιθυμητή έξοδο / λύση. Αν είναι πολύ μεγάλος, οι αλλαγές των βαρών σε μια επανάληψη είναι έντονες, γεγονός το οποίο μπορεί να οδηγήσει σε ταλάντωση του δικτύου γύρω από τη λύση (δηλαδή οι αλλαγές των βαρών προκαλούν μεγάλη αλλαγή στην έξοδο του δικτύου). Αν είναι πολύ μικρός, υπάρχει μεγαλύτερη ακρίβεια στην προσέγγιση της λύσης, καθώς οι τιμές των βαρών αλλάζουν ελάχιστα, αλλά μπορεί να υπάρχει μεγάλη καθυστέρηση στη σύγκλιση.

Γενικά μπορεί να είναι επιθυμητό αρχικά ο ρυθμός μάθησης να είναι μεγάλος, προκειμένου το δίκτυο να πλησιάσει πιο γρήγορα την επιθυμητή λύση και έπειτα ο ρυθμός να μειωθεί για να υπάρξει μεγαλύτερη ακρίβεια στη σύγκλιση του δικτύου.

Οι ζητούμενες μετρήσεις για τις συναρτήσεις εκπαίδευσης `traingd`, `traingdx` έγιναν με χρήση του script `find_learning_rate.m` και παρουσιάζονται παρακάτω:

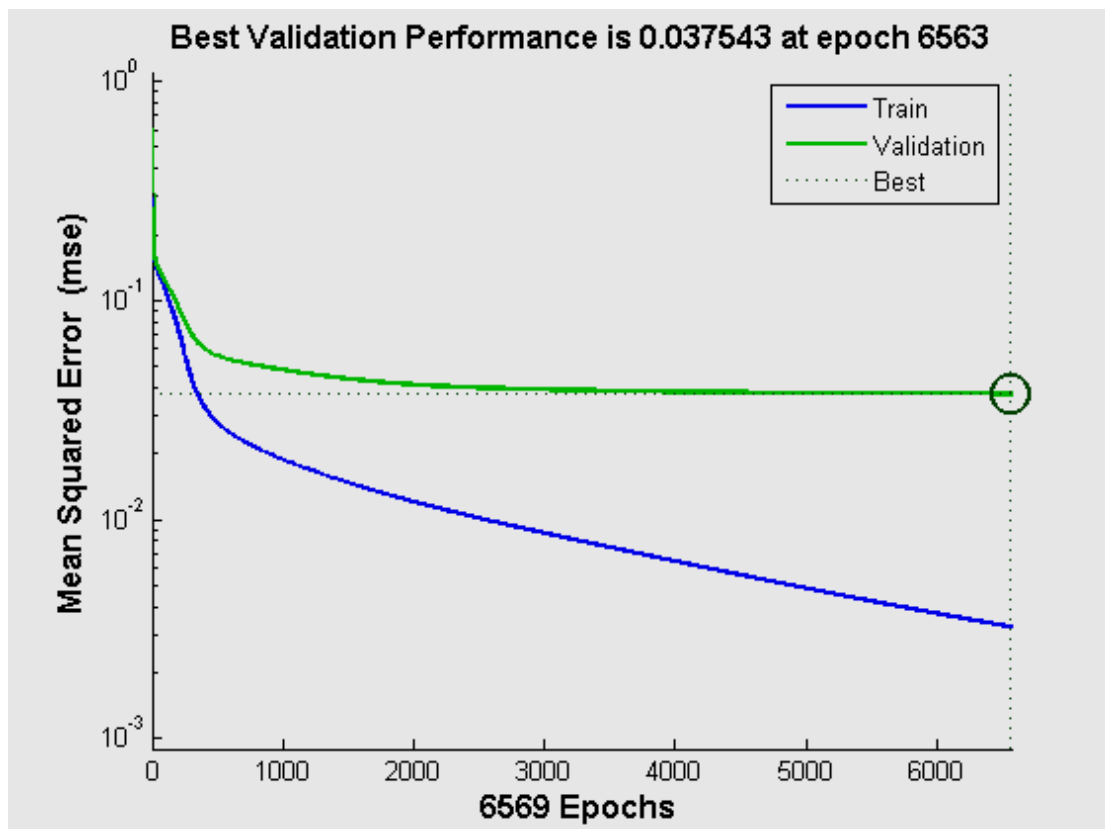






Σύμφωνα με τα παραπάνω διαγράμματα η απόδοση των δύο συναρτήσεων εκπαίδευσης είναι περίπου ίδια . Η `trainidx` δεν επηρεάζεται από το δεδομένο `learn rate`, καθώς προσαρμόζει τον ρυθμό μάθησης κατά την εκπαίδευση. Επισημαίνεται ότι οι μετρήσεις για την `trainidx` έγιναν για 150000 εποχές, ενώ για την `trainidx` έγιναν για 1000 εποχές.

Επίσης παρατηρείται ότι όσο μεγαλώνει το `learn rate`, το δίκτυο στην περίπτωση της `trainidx` συγκλίνει πιο γρήγορα. Συγκεκριμένα για `learn rate` 0.05 οι 150000 εποχές δεν είναι αρκετές για τη σύγκλιση του δικτύου, ενώ για `learn rate` 0.4 το δίκτυο συγκλίνει περίπου στις 7000 εποχές, όπως φαίνεται παρακάτω:



Ερώτημα 8 – Μελέτη απόδοσης κάθε κατηγορίας:

Με βάση τις παραπάνω μετρήσεις καλύτερη απόδοση παρουσιάζει το δίκτυο για αρχιτεκτονική [15 15], με συνάρτηση εκπαίδευσης `trainidx` και συνάρτηση ενεργοποίησης `tansig` (ή `purelin`). Επίσης χρησιμοποιείται μέθοδος `early stopping` και και αλγόριθμος μάθησης `learnidx`.

Στα προηγούμενα ερωτήματα, η απόδοση του δικτύου μελετήθηκε με βάση την παράμετρο `accuracy`, η οποία παρουσιάζει μια συνολική εικόνα ως προς την εκπαίδευση του δικτύου. Ωστόσο σε ορισμένα από τα παραπάνω διαγράμματα `precision` και `recall` παρατηρείται μια σχετικά μικρή ανισορροπία στην απόδοση ορισμένων κατηγοριών (κυρίως στις κατηγορίες 1, 3 και 4). Αυτό μπορεί να

οφείλεται στο γεγονός ότι ορισμένες κατηγορίες παρουσιάζουν περισσότερες ομοιότητες μεταξύ τους, επομένως οι εικόνες τους είναι πιο δύσκολο να ταξινομηθούν σωστά και μπορεί το δίκτυο να μην έχει εκπαιδευτεί επαρκώς για τον συγκεκριμένο διαχωρισμό.

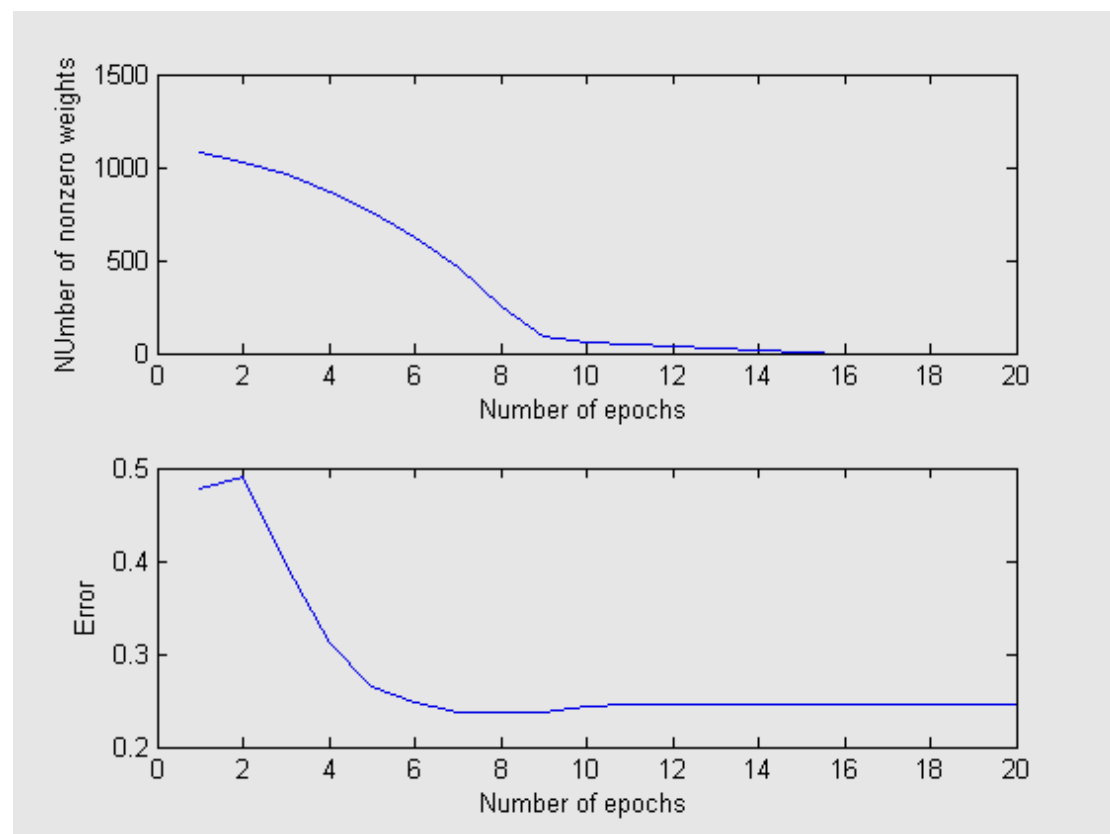
Για να αντιμετωπιστεί το παραπάνω πρόβλημα μια λύση είναι να εισάγεται μεγαλύτερο πλήθος δεδομένων από τις κατηγορίες που παρουσιάζουν μεταξύ τους ομοιότητες. Με αυτό τον τρόπο το δίκτυο θα εκπαιδεύεται περισσότερο στην αναγνώριση των εικόνων από τις παραπάνω κατηγορίες και θα μειώνονται τα σφάλματα εξόδου.

Μια άλλη λύση είναι να διαμορφώνονται τα δεδομένα εισόδου κατά την προεπεξεργασία, προκειμένου να διατηρούνται περισσότερα χαρακτηριστικά για τις κατηγορίες που παρουσιάζουν μεγαλύτερη ομοιότητα μεταξύ τους. Αυτό ωστόσο δεσμεύει το χρήστη με την εισαγωγή δεδομένων συγκεκριμένης μορφής κατά τη διαδικασία της προεπεξεργασίας (μπορεί τα δεδομένα να μην έχουν όλα ίδια μορφή).

Μέθοδος αποσύνθεσης βαρών:

Στη συγκεκριμένη ενότητα γίνεται επέκταση του αλγορίθμου backpropagation για να χρησιμοποιείται η μέθοδος pruning. Η υλοποίηση των βημάτων, τα οποία περιγράφονται στην εκφώνηση γίνεται στο script `weight_decay2.m`.

Τα ζητούμενα διαγράμματα είναι τα εξής:



Παρατηρήσεις:

- Το πλήθος των νευρώνων με μη μηδενικά βάρη παρουσιάζει μεγάλη μείωση, η οποία οφείλεται στη δράση της μεθόδου κλαδέματος.
- Το σφάλμα εξόδου μειώνεται εκθετικά με το πλήθος των εποχών, αλλά μετά από ένα σημείο φαίνεται να σταθεροποιείται. Αυτό μπορεί να οφείλεται στο γεγονός ότι με την πάροδο των επαναλήψεων όλο και περισσότεροι νευρώνες γίνονται ανενεργοί με αποτέλεσμα το μέγεθος του δικτύου να περιορίζεται και επομένως το αντίστοιχο σφάλμα να ελαττώνεται. Όταν το πλήθος των νευρώνων γίνει αρκετά μικρό, το σφάλμα εξόδου σταθεροποιείται σε κάποια τιμή.