

ΣΧΟΛΗ : ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ Η/Υ

ΜΑΘΗΜΑ : Νευρωνικά Δίκτυα και Ευφυή Υπολογιστικά Συστήματα

1η Εργαστηριακή Άσκηση:

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



ΚΩΝΣΤΑΝΤΙΝΟΣ ΑΜΕΡΑΝΗΣ : 03112177

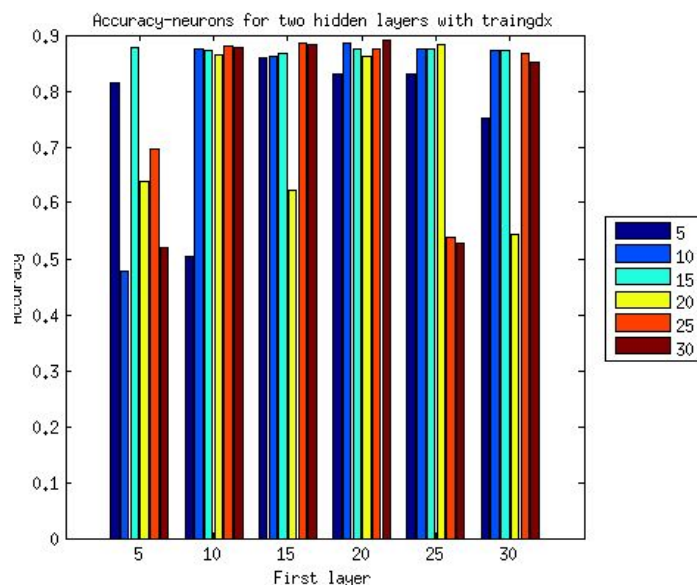
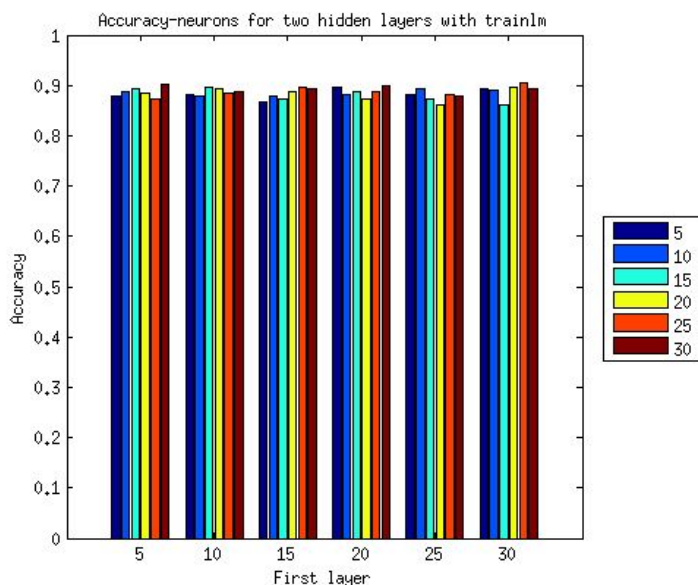
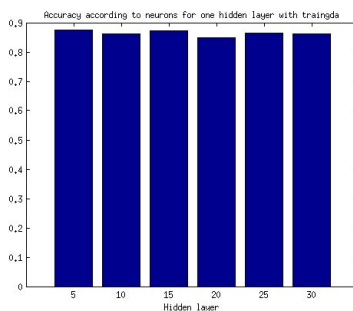
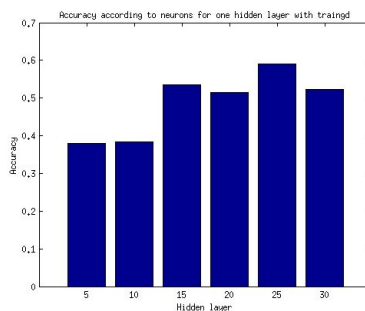
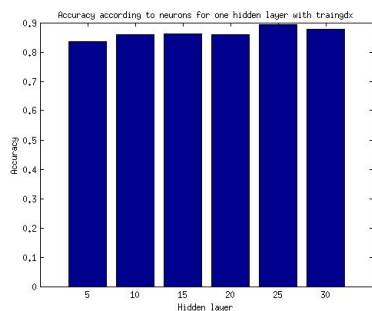
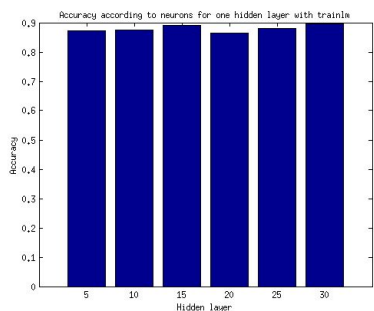
ΒΑΣΙΛΗΣ ΓΚΟΥΜΑΣ : 03113031

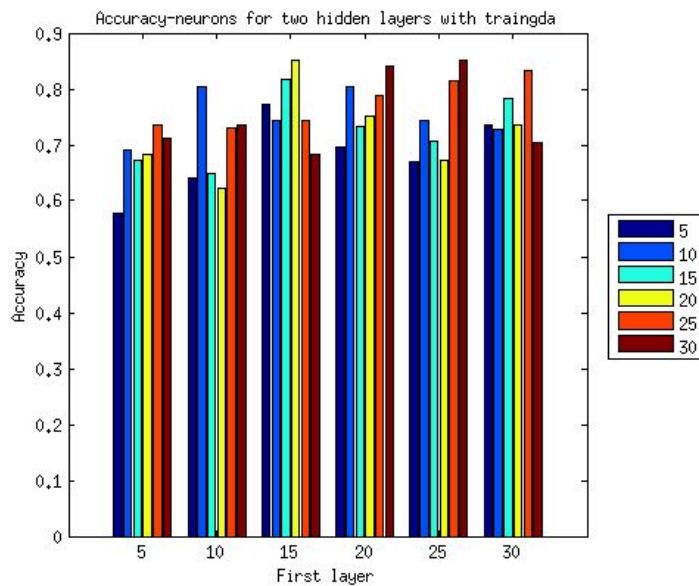
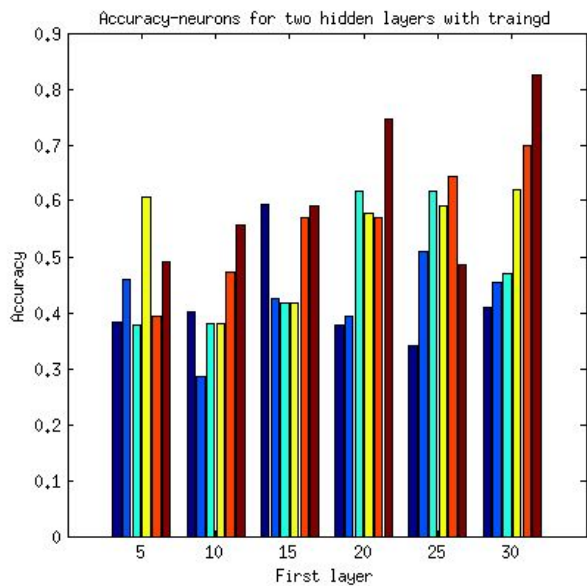
Ερώτημα 1

Πριν την εκπαίδευση του νευρωνικού δικτύου κάνουμε προεπεξεργασία και καθαρισμό των δεδομένων που μας δόθηκαν. Αρχικά κρατάμε ισάριθμο πλήθος δειγμάτων από κάθε κατηγορία που εμφανίζεται στα *Traindata*. Μετά την εφαρμογή της *removeconstantrows* για να ξεφορτωθούμε τα περιττά δεδομένα που δεν συμβάλλουν στην εκπαίδευση εκτελούμε τους μετασχηματισμούς μας στα *train* και *test*. Πρώτα κάνουμε *normalization* των *features* ώστε να ακολουθούν κανονική κατανομή με μηδενική μέση τιμή και μοναδιαία διασπορά. Τέλος εκτελούμε τον ορθογώνιο μετασχηματισμό PCA που εξαλείφει συσχετιζόμενα *features*, μειώνοντας τον αριθμό των αρχικών *features* από 159 σε 19 (με παράμετρο 0.01 στην *processprca*).

Η εκπαίδευση του νευρωνικού μπορεί να γίνει χωρίς προεπεξεργασία, με τίμημα όμως τον **περισσότερο χρόνο** που θα κρατήσει η εκπαίδευση, λόγω των πολύ περισσότερων *features* και της διαφορετικής κλίμακας που βρίσκονται. Επιπλέον θα έχουμε **χειρότερες επιδόσεις** καθώς τα δεδομένα μας θα είναι πιο *skewed*.

Ερώτημα 2





Για κάθε αρχιτεκτονική έχουμε εκτελέσει 20 επαναλήψεις, ώστε να έχουμε στατιστικά έγκυρα αποτελέσματα. Για την `trainlm` παρατηρούμε ότι γενικά έχει υψηλό ποσοστό επιτυχίας, που κυμαίνεται στο 85%-90%. Όλες οι ιδιαίτερα καλές αρχιτεκτονικές είναι με την συνάρτηση `trainlm` και για 30 νευρώνες σε έναν επίπεδο (89.18%). Για τα δίκτυα δύο επιπέδων, παρατηρούμε πολύ μικρή βελτίωση (όταν υπάρχει) και θεωρούμε ότι είναι πολύ μικρή για να δικαιολογήσει το υπολογιστικό κόστος που απαιτείται για να εκπαιδεύσουμε ένα τέτοιο νευρωνικό. Οπότε διαλέγουμε την αρχιτεκτονική με ένα κρυμμένο επίπεδο και 30 νευρώνες.

Ερώτημα 3

Διαλέγοντας ως καλύτερη αρχιτεκτονική δικτύου αυτή με ένα hidden layer 30 νευρώνων, δοκιμάζουμε διαφορετικές συναρτήσεις ενεργοποίησης για το στρώμα εξόδου του νευρωνικού μας δικτύου. Η προκαθορισμένη συνάρτηση συνάρτηση ενεργοποίησης (`tansig`) μπορεί να αλλάξει, άμα θέσουμε την παρακάτω παράμετρο:

`net.layers{length(neurons)}.transferFcn = 'tansig'`

Δοκιμάσαμε τις `tansig`, `hardlim`, `logsig`, `purelin`, με την `logsig` και την `tansig` να είναι περίπου στο ίδιο έδαφος με ποσοστά 0.8978 και 0.8909 αντίστοιχα με στατιστικά αμελητέα διαφορά, ενώ οι υπόλοιπες δεν έχουν τόσο καλά αποτελέσματα.

Ερώτημα 4

Για την αρχιτεκτονική που έχουμε επιλέξει, δεν παρατηρούμε κάποια διαφορά. Τα αποτελέσματα έχουν παρόμοια ποιότητα και τελειώνουν σε παρόμοιους χρόνους και iterations.

Ερώτημα 5

Η μέθοδος Early Stopping χρησιμοποιεί το validation set ώστε να αποφύγουμε το overfitting. Πιο συγκεκριμένα, τα δεδομένα εκπαίδευσης διαχωρίζονται σε train και validation set. Το ποσοστό της κάθε κατηγορίας καθορίζεται από το `net.divideParam.trainRatio/valRatio`. Εμείς χρησιμοποιήσαμε τον προτεινόμενο διαχωρισμό 80/20. Η εκπαίδευση του νευρωνικού ξεκινάει κανονικά. Μετά από κάθε εποχή, ελέγχεται το λάθος στο validation set. Αν κάποια στιγμή το λάθος στο validation είναι μεγαλύτερο από το αντίστοιχο στο train, η εκπαίδευση σταματάει ώστε να αποφευχθεί το overfitting στα traindata.

Ερώτημα 6

Όταν δεν χρησιμοποιούμε validation set, παρατηρούμε ότι η εκπαίδευση συνεχίζει για πολύ περισσότερο (μερικές φορές εξαντλεί τα 1000 iterations) και έχει χειρότερα αποτελέσματα. Όταν έχουμε μικρό αριθμό εποχών, το νευρωνικό δεν γενικεύει και έχει κακά αποτελέσματα. Όταν έχουμε μεγάλο αριθμό εποχών, η εκπαίδευση του νευρωνικού διαρκεί πάρα πολύ.

Ερώτημα 7

Όταν έχουμε μικρό ρυθμό μάθησης, το νευρωνικό αργεί να συγκλίνει ή συγκλίνει σε κακό μοντέλο. Όταν έχουμε μεγάλο ρυθμό μάθησης, το νευρωνικό υπάρχει περίπτωση να μην συγκλίνει, οπότε και να μην έχουμε σωστά αποτελέσματα.

Ερώτημα 8

Για 30 νευρώνες και συνάρτηση εκπαίδευσης την `trainlm` τρέξαμε 100 εκπαιδεύσεις και αποτιμήσεις και τελικά πήραμε το παρακάτω confusion matrix.

#/#	1	2	3	4	5
1	40.04	0.00	0.00	2.05	0.00
2	0.00	21.58	2.09	0.44	0.00
3	0.00	7.36	27.70	0.04	2.57
4	1.96	0.77	0.05	23.47	0.00
5	0.00	0.29	0.16	0.00	28.43

Παρατηρούμε ότι πολλά παραδείγματα της κλάσης 3 ταξινομούνται ως παραδείγματα της κλάσης 2 και αντίστροφα. Αντίστοιχα, παρατηρούμε το ίδιο φαινόμενο πιο αδύναμο μεταξύ των κλάσεων 1 και 4. Αυτό οφείλεται στο γεγονός ότι ένα νευρωνικό με μόνο ένα επίπεδο δεν μπορεί να μοντελοποιήσει όλες τις συναρτήσεις. Επίσης, μπορεί να μην έχουμε δώσει τα σωστά features, όμως αυτό είναι πέρα από τους στόχους της άσκησης. Τέλος, μπορεί πράγματι να μην υπάρχει τέλειος διαχωρισμός μεταξύ των 2 κλάσεων. Αυτό μπορούμε να το αντιμετωπίσουμε έχοντας πρώτα έναν classifier μεταξύ των κλάσεων 14, 23, 5 και στην συνέχεια για τις συγχωνευμένες κλάσεις 14 και 23 επιπλέον classifiers για να διαχωρίσει μεταξύ των δύο.

Μέθοδος της αποσύνθεσης βαρών

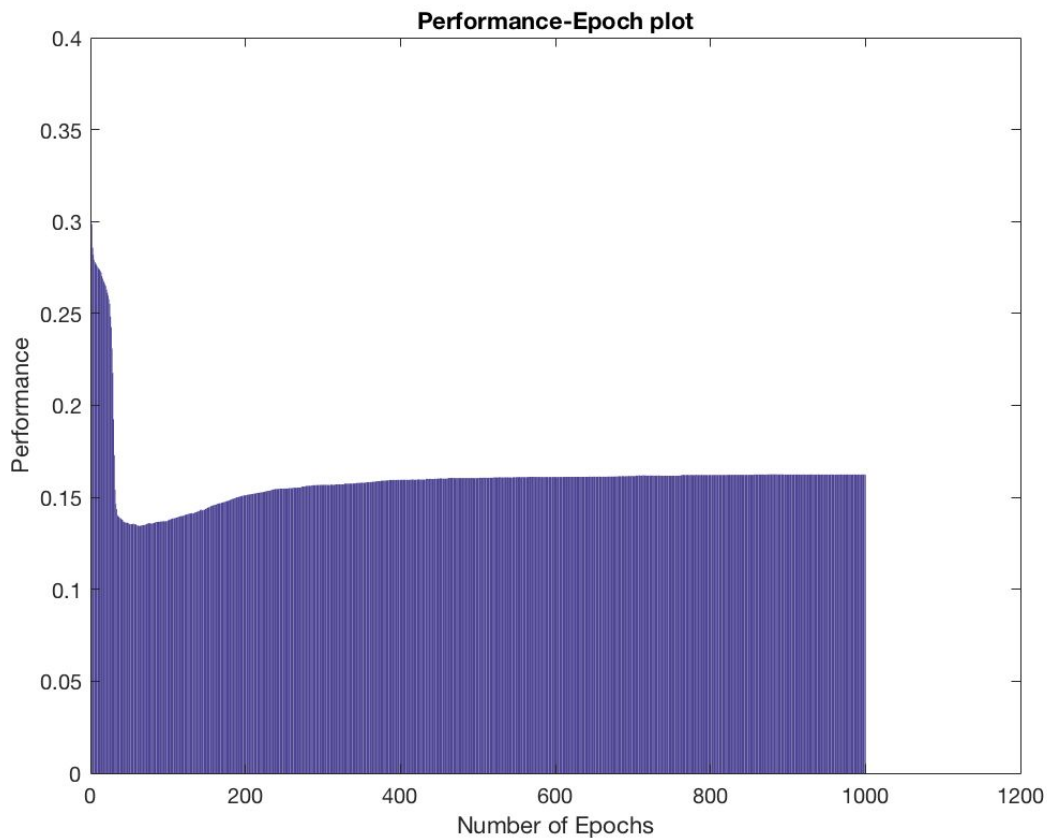
Η μέθοδος της αποσύνθεσης βαρών περιγράφεται από την παρακάτω σχέση:

$$w_{ij}(k) - w_{ij}(k-1) = -\mu \partial E / \partial w_{ij}(k-1) - \lambda w_{ij}(k-1)$$

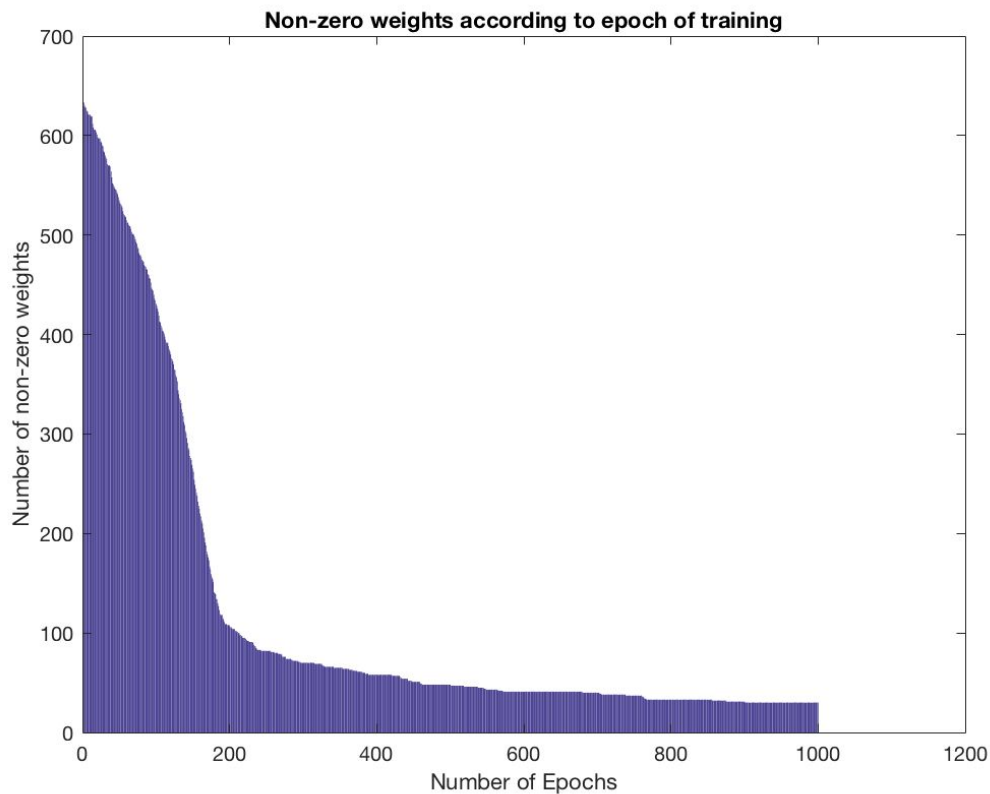
Η επίδραση του επιπλέον όρου μετά την κλίση, είναι η εκθετική μείωση των βαρών, αν σε αυτά δεν εφαρμόζεται καμία αλλαγή. Ουσιαστικά προσθέτουμε χειροκίνητα την κλίση (gradient) του όρου κανονικοποίησης $-\frac{\lambda}{2}w^2$. Έτσι αφαιρούμε τα βάρη των οποίων η νόρμα δεν ξεπερνάει κάποιο κατώφλι που ορίσαμε, θεωρώντας ότι είναι περιττά για το νευρωνικό μας και εισάγουν περιττή πολυπλοκότητα.

Για τα τρεξίματα που κάναμε χρησιμοποιήσαμε την απλή `traingd` και για συνάρτηση μάθησης την `learngd` και **κριτήριο τερματισμού τις 1000 εποχές**. Για τον παραπάνω λόγο δεν χρησιμοποιήσαμε `validation set`. Παρακάτω βλέπουμε τα ζητούμενα διαγράμματα.

Σημείωση: Οι συναρτήσεις `traingd/learngd` δεν μας έδωσαν τα καλύτερα αποτελέσματα αλλά χρησιμοποιήθηκαν καθώς θέλουμε ανανέωση σε *batch mode*.



Στο παραπάνω διάγραμμα (**pruned-perf.png**) φαίνεται η απόδοση του νευρωνικού καθώς εκτελούμε το κλάδεμα των νευρώνων για 1000 επαναλήψεις. Παρατηρούμε ότι καθώς προχωράει το κλάδεμα η απόδοση πέφτει μέχρι κάποιο σημείο που σταθεροποιείται. Πιθανώς κάποιο διαφορετικό στοιχείο τερματισμού, πχ χρήση Validation Set/Early Stopping ή όταν φτάσουμε σε συγκεκριμένο σφάλμα, να έκοβε την εκπαίδευση πριν χειροτερεύσει τόσο πολύ η επίδοση. Στη συνέχεια βλέπουμε τον αριθμό των βαρών που παραμένουν με την πάροδο των εποχών εκπαίδευσης. (**non-zero.png**).



Βλέπουμε λοιπόν ότι ο αριθμός των νευρώνων που κλαδεύονται μεγαλώνει εκθετικά ανά εποχή για να σταθεροποιηθεί προς το τέλος των 1000 εποχών.