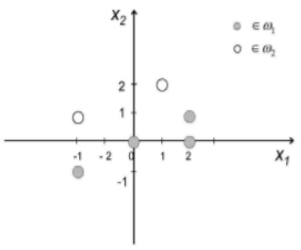
Αναγνώριση προτύπων
Εργασία 7
Σάββας Λιάπης 57403

Άσκηση 7.1 ΓΡΑΜΜΙΚΟ PERCEPTRON

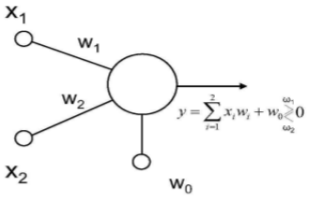
Έστω 6 δείγματα εκπαίδευσης που ανήκουν σε δύο κλάσεις ωι,ω2, όπως φαίνονται στο σχήμα (1):



Σχήμα 1

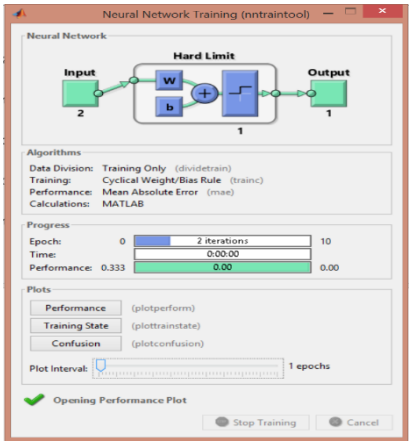
$$S = \left\{ \begin{pmatrix} -2 \\ 1 \end{pmatrix}, \quad \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \quad \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \begin{pmatrix} -2 \\ -1 \end{pmatrix}, \quad \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \quad \begin{pmatrix} 2 \\ 1 \end{pmatrix} \right\}$$

Να ορίσετε ένα νευρώνα (perceptron) που να ορίσει την ταξινόμηση αυτών των δύο κλάσεων. Δηλαδή, ορίστε τις συνάψεις w1,w2 και το κατώφλι w0



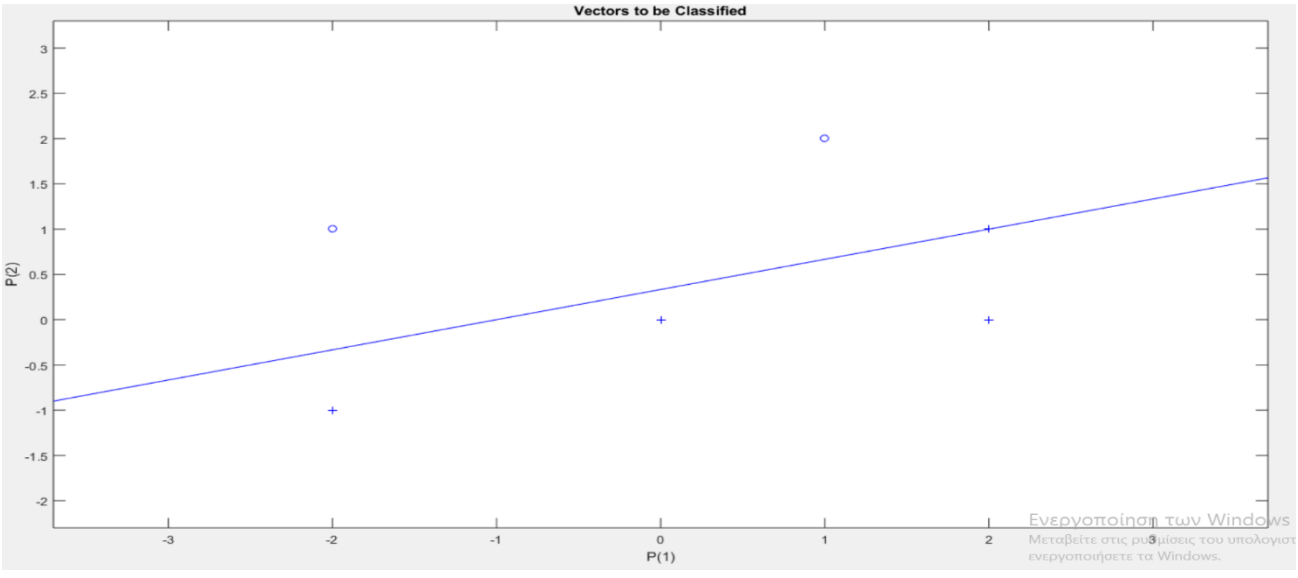
Απάντηση 7.1:

Για την εκτέλεση της άσκησης αρχικοποιούμε τις εισόδους και τις αναμενόμενες εξόδους και έπειτα αρχικοποιούμε το νευρώνα perceptron με την εντολή net=perceptron; . Αφού κάνουμε train παίρνουμε τα βάρη w1 w2 και το κατώφλι b .




```
w =

     1    -3

b =

     1
```

Ο αλγόριθμος έχει χωρίσει σε 2 κλάσεις θεωρώντας ότι όποια έξοδος είναι <0 ανήκει στην κλάση ω1 και ότι είναι ≥ 0 ανήκει στην κλάση ω2 . Δεν είναι μία και μοναδική η λύση που ικανοποιεί αυτή την συνθήκη, αλλά ο αλγόριθμος επιλέγει την πρώτη λύση που ικανοποιεί την συνθήκη.



Άσκηση 7.2 IRIS Data Set με Ταξινομητή Πολυστρωματικά Perceptron

Α. Να ταξινομηθούν τα δεδομένα με το γραμμικό Perceptron. Το νευρωνικό δίκτυο αυτό αποτελείται από τρεις νευρώνες (ένας για κάθε κλάση) με συνάρτηση μεταφοράς την βηματική συνάρτηση. Έτσι, ουσιαστικά, ο κάθε νευρώνας «αποφασίζει», εάν το εισερχόμενο pattern είναι ή δεν είναι στην κλάση που «διαχειρίζεται». Στην συγκεκριμένη υλοποίηση να γίνει αρχικοποίηση των βαρών και των biases με τυχαία συνάρτηση, ώστε να εξασφαλισθεί η ταυτόχρονη σύγκλιση τους στις τελικές τιμές τους.

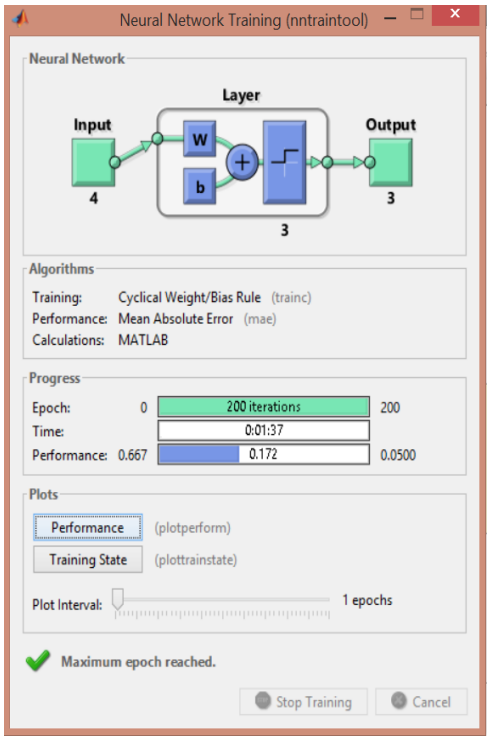
Β. Να ταξινομηθούν τα δεδομένα με το multi-layer Perceptron με ένα hidden layer και τρεις νευρώνες στην έξοδο (ένας για κάθε κλάση). Χρησιμοποιείστε τον backpropagation αλγόριθμο και δοκιμάστε διαφορετικό αριθμό νευρώνων στο hidden layer (2, 5 και 10 νευρώνες).

Γ. (Προαιρετικό) Να ταξινομηθούν τα δεδομένα με το multi-layer Perceptron με δύο hidden layer και τρεις νευρώνες στην έξοδο (ένας για κάθε κλάση). Χρησιμοποιείστε το backpropagation αλγόριθμο και δοκιμάστε διαφορετικό αριθμό νευρώνων στα hidden layer

Συγκρίνετε τα αποτελέσματα για τα Α, Β, Γ.

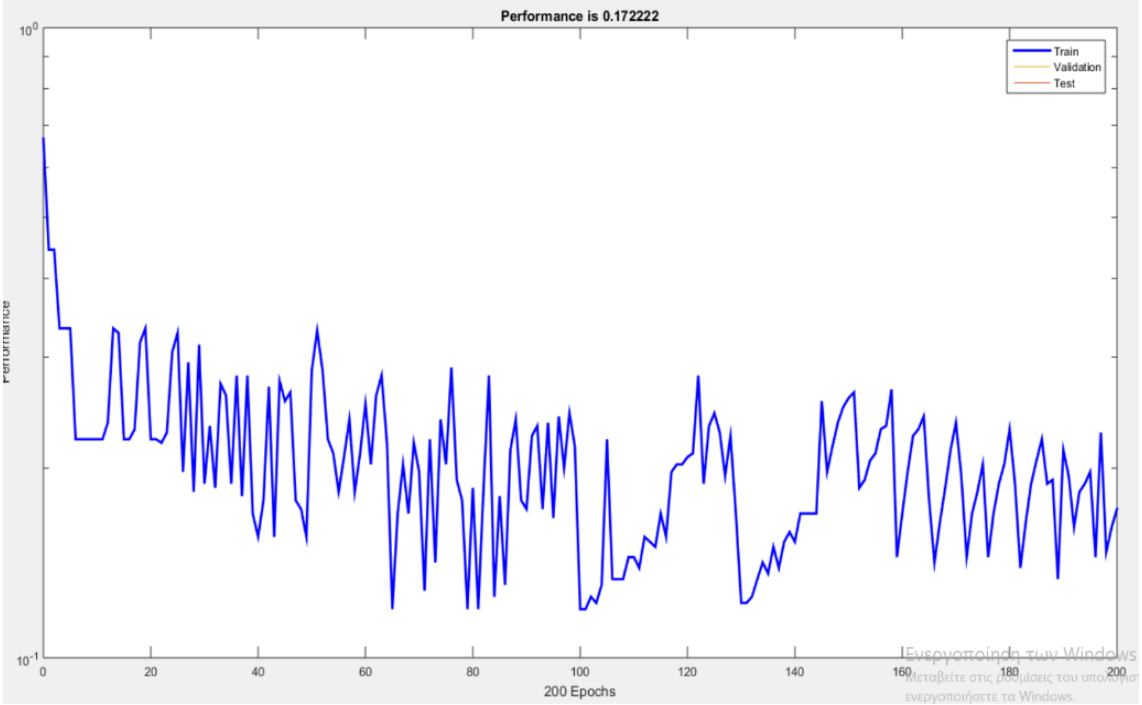
Απάντηση 7.2.Α:

Για να πετύχουμε το ζητούμενο της άσκησης αρχικοποιούμε το iris dataset και ορίζουμε ένα πίνακα target ο οποίος στην ουσία κάνει one hot indication για το σε ποια κλάση ανήκει το κάθε στοιχείο (4 διαστάσεων) του dataset.
Στην συνέχεια αρχικοποιούμε ένα νευρωνικό δίκτυο με κανένα κρυφό νευρώνα και 3 νευρώνες εξόδου με την εντολή net=newp(minmax(dataset),3);
Θέτουμε τα βάρη και τα biases να είναι τυχαία με τις εντολές net.iw{1,1}=rand(3,4); Και net.b{1,1}=rand(3,1);
Μετά από αυτό εκτελούμε το training. Από αυτό παίρνουμε τα εξής αποτελέσματα :




```
mintrainerror =

    0.1194


optimum =

    66    80    82   101   102
```

Το ελάχιστο σφάλμα που επιτυγχάνεται κατά την προπόνηση για 200 εποχές είναι 11%. Παρακάτω φαίνεται και το διάγραμμα του performance.



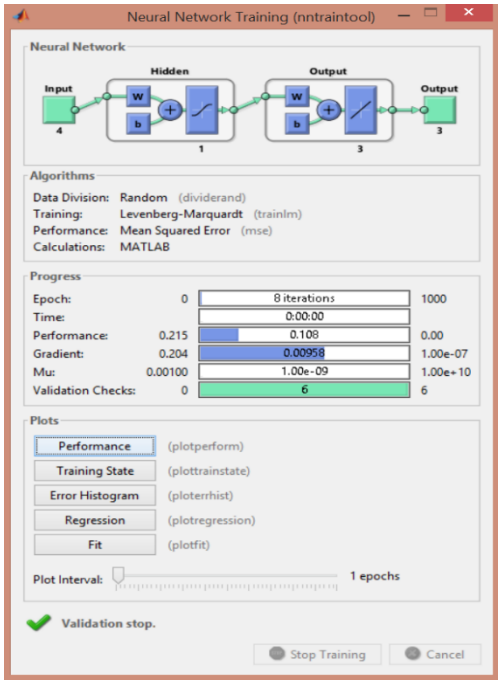
Επίσης δοκιμάζουμε να κάνουμε ένα testing το οποίο δεν μας δίνει πολύ καλά αποτελέσματα καθώς βλέπουμε ότι το error είναι στο 50% καθώς εκτός από τα στοιχεία που δεν κατατάσσονται σε λάθος κλάση κάποια στοιχεία κατατάσσονται και σε 2 κλάσεις

```
testerror =

    0.5333
```

Απάντηση 7.2.Β:

Σε αυτό το ερώτημα αρχικοποιούμε όπως και πριν το dataset και τον πίνακα με τα target classes και έπειτα αρχικοποιούμε το νευρωνικό δίκτυο με την συνάρτηση fitnet μέσω της οποία μπορούμε να ορίσουμε πόσα κρυφά layers θα έχουμε και πόσους νευρώνες σε κάθε layer. Για αυτό το ερώτημα θα πειραματιστούμε με ένα κρυφό επίπεδο αλλάζοντας τον αριθμό των νευρώνων σε αυτό. Επίσης σαν backproagation αλγόριθμος θα χρησιμοποιηθεί ο Levenberg-Marquardt backpropagation algorithm. Για κάθε δοκιμή βλέπουμε το καλύτερο αποτέλεσμα καθώς και την εποχή που πραγματοποιήθηκε αυτό.

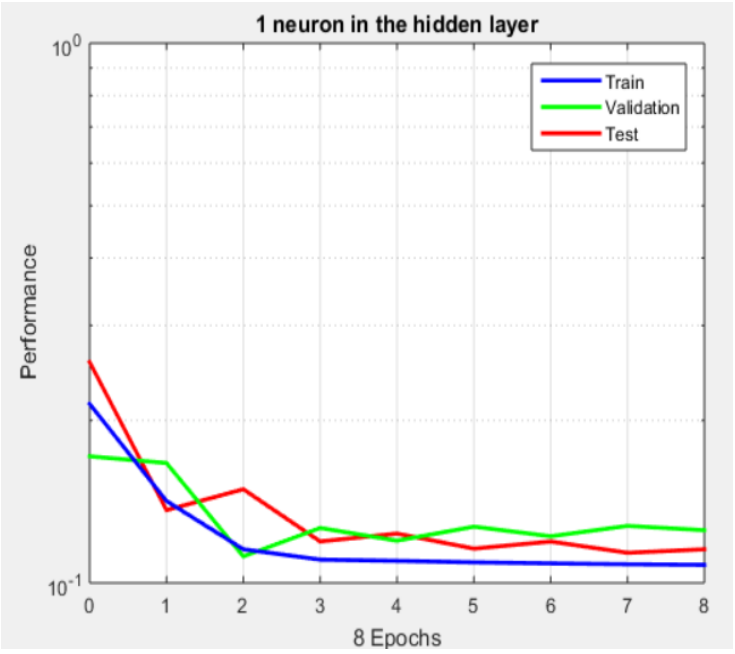
<u>1 νευρώνας :</u>



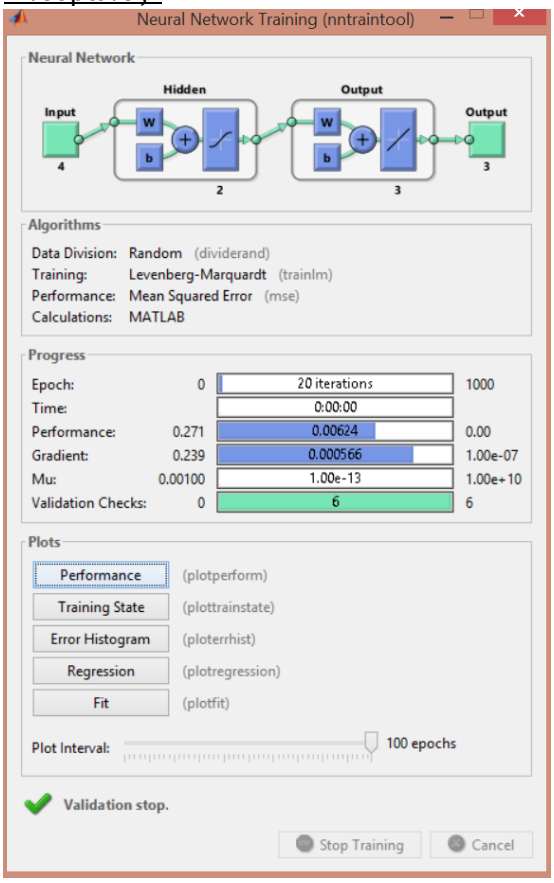
Δεξιά βλέπουμε την αναπαράσταση του νευρωνικού δικτύου. Βλέπουμε ότι έχουμε ένα hidden layer με 1 νευρώνα και ένα output layer με 3 νευρώνες. Υλοποιώντας το νευρωνικό δίκτυο παίρνουμε τα εξής αποτελέσματα

```
best_train_error1 =

    0.1154


best_validation_error1 =

    0.1119


best_test_error1 =

    0.1491


bestepoch1 =

     2
```

Βλέπουμε ότι η καλύτερη εποχή είναι η δεύτερη γιατί το validation error παίρνει την ελάχιστη τιμή του σε αυτή την εποχή. Από εκεί και μετά παρόλο που βλέπουμε το training error να βελτιώνεται το validation error μεγαλώνει και αυτό σημαίνει ότι χάνουμε όλο και περισσότερο το generalization (έχουμε over-fitting που δεν είναι επιθυμητό) . Ωστόσο ο αλγόριθμος δεν σταματάει να τρέχει. Θα σταματήσει όταν για 6 ελέγχους του validation αυτό δεν θα δώσει καλύτερο αποτέλεσμα.

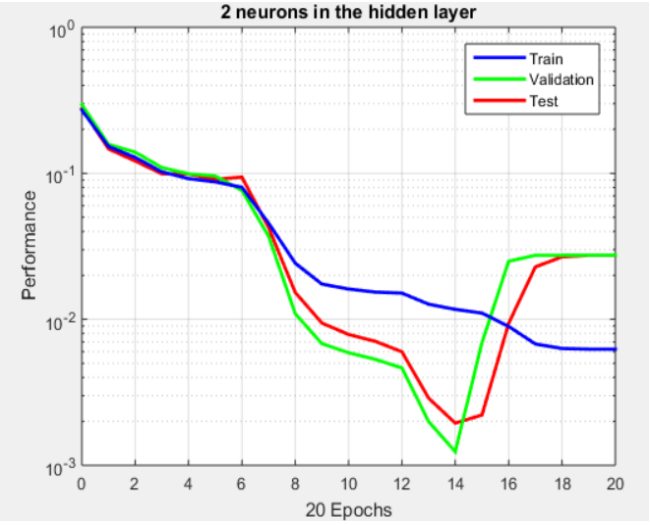


2 νευρώνες :

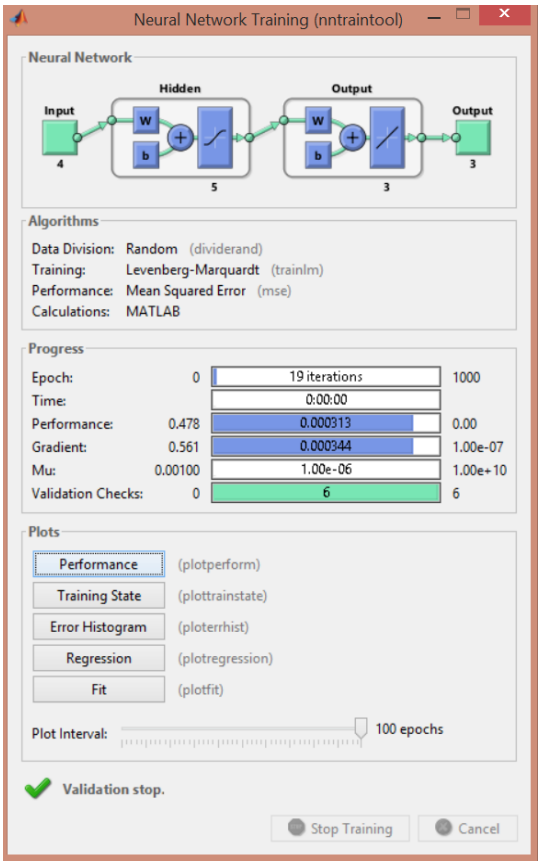



```
train_error2 =
    0.0117

validation_error2 =
    0.0012

test_error2 =
    0.0019

bestepoch2 =
    14
```

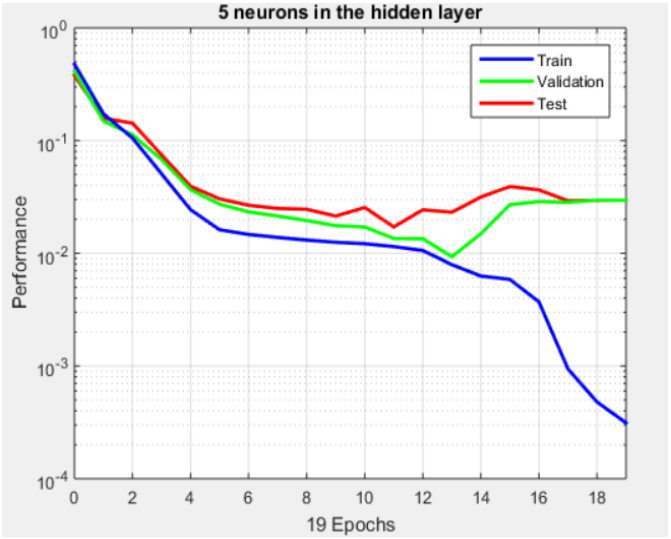
Την εποχή 14 πετυχαίνουμε το καλύτερο validation error. Επίσης βλέπουμε ότι με 2 νευρώνες πετυχαίνουμε καλύτερα σφάλματα από ότι με έναν στο κρυφό επίπεδο.



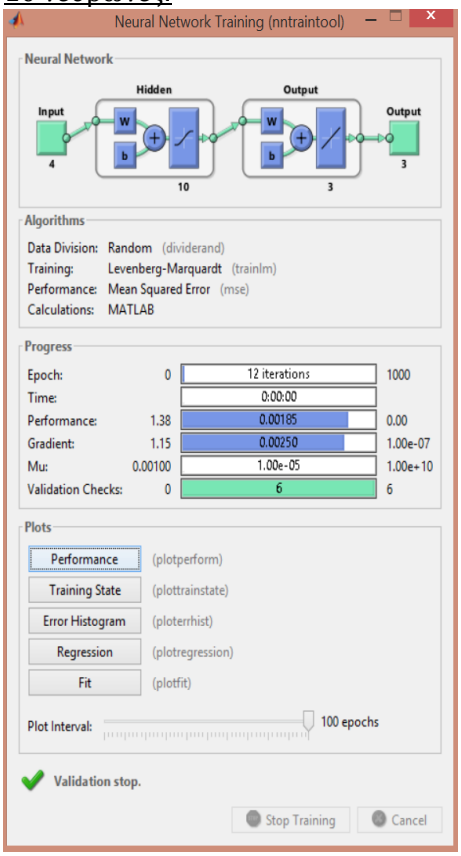
5νευρώνες:



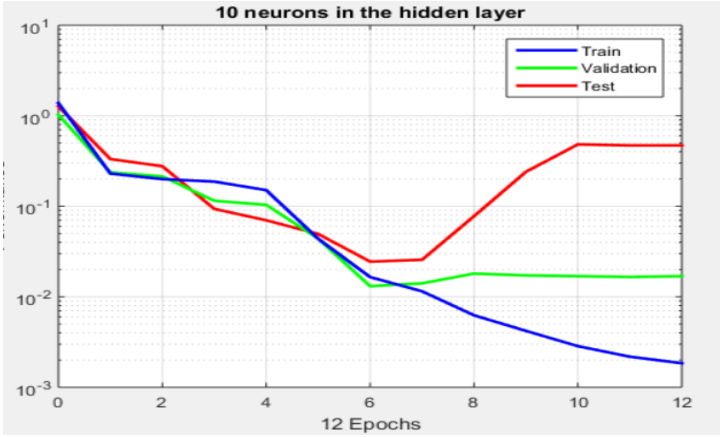

```
train_error5 =
    0.0079

validation_error5 =
    0.0093

test_error5 =
    0.0232

bestepoch5 =
    13
```



10 νευρώνες:



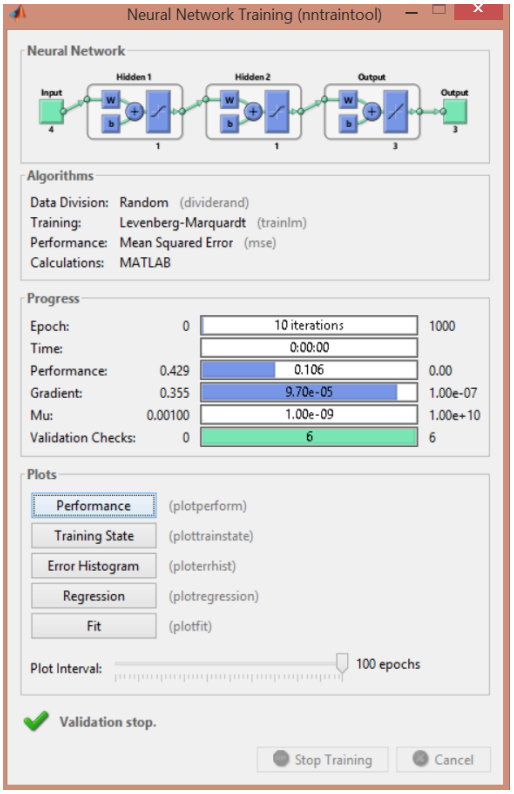

```
train_error10 =

    0.0165


validation_error10 =

    0.0131


test_error10 =

    0.0245


bestepoch10 =

     6
```



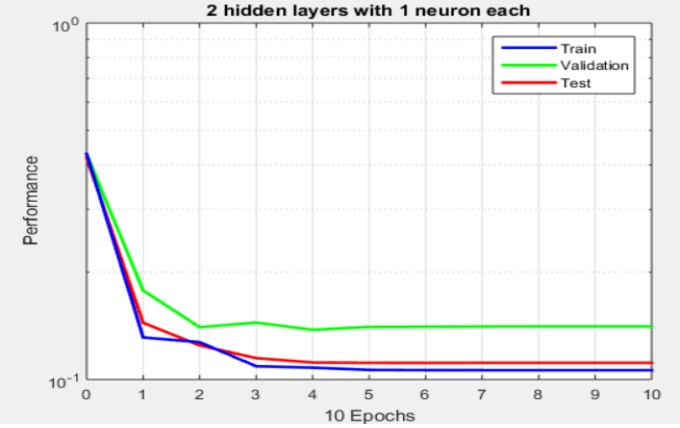
Εξετάζοντας τα αποτελέσματα από όλα τα πειράματα καταλαβαίνουμε ότι αυξάνοντας τους νευρώνες το σφάλμα μειώνεται . Ωστόσο στους 10 νευρώνες πετυχαίνουμε χειρότερα αποτελέσματα από ότι με 5 νευρώνες στο κρυφό επίπεδο (και αυτό συνέβαινε για όλες τις προσπάθειες όχι μόνο για αυτή που απεικονίζεται). Οπότε μπορούμε να συμπεράνουμε ότι υπάρχουν νευρώνες που στην ουσία είναι άχρηστοι και δεν βελτιώνουν την συμπεριφορά του νευρωνικού δικτύου μας.

Απάντηση 7.2.Γ:

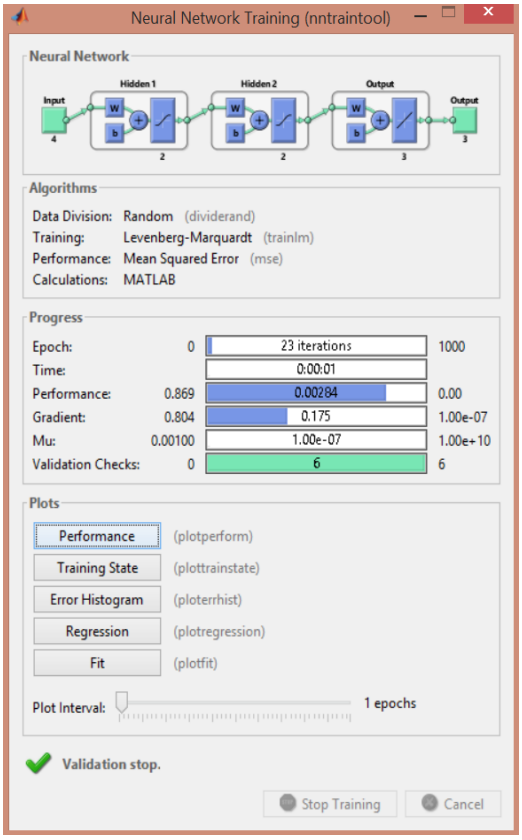
1 νευρώνας σε κάθε κρυφό επίπεδο :



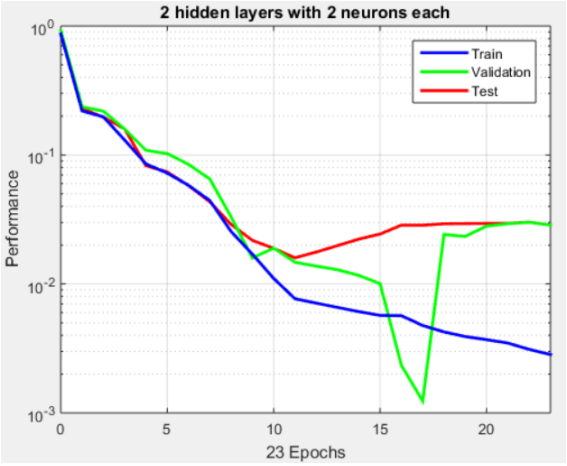

```
train_errorC1 =

    0.1080

validation_errorC1 =

    0.1379

test_errorC1 =

    0.1116

bestepochC1 =

     4
```



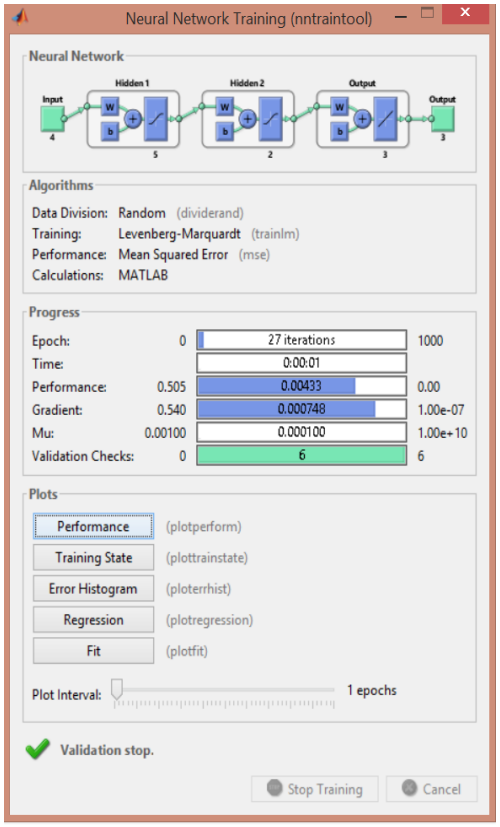
2 νευρώνες σε κάθε κρυφό επίπεδο :



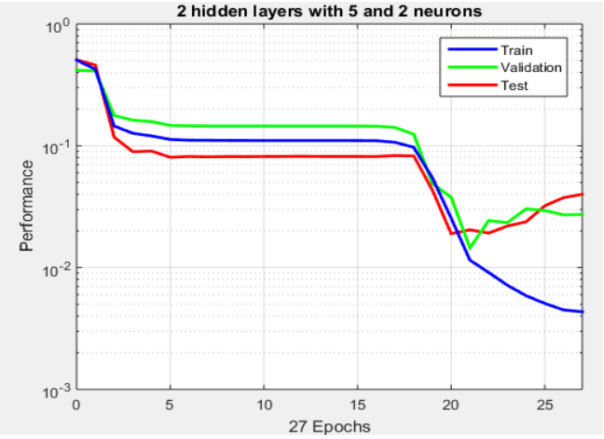

```
train_errorC2 =

    0.0048

validation_errorC2 =

    0.0012

test_errorC2 =

    0.0285

bestepochC2 =

    17
```



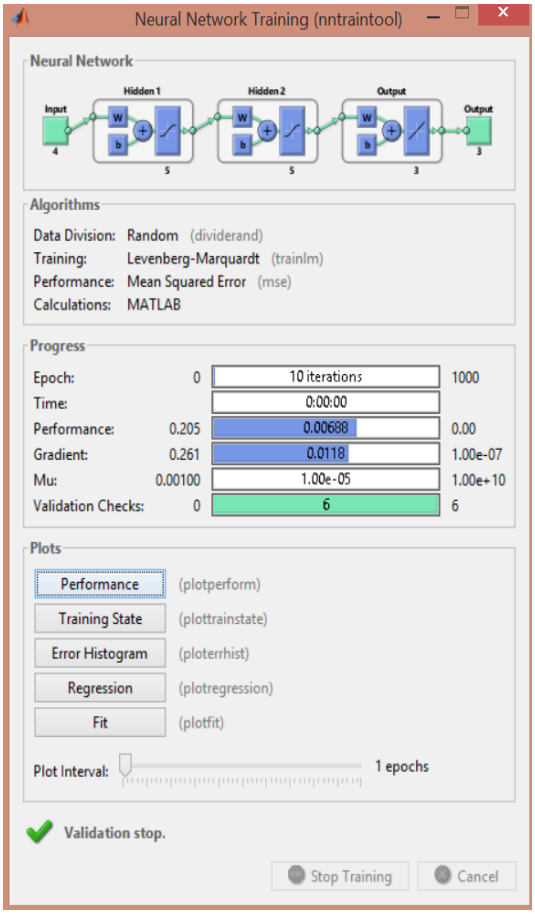
Ένα κρυφό επίπεδο με 5 νευρώνες και ένα με 2 :



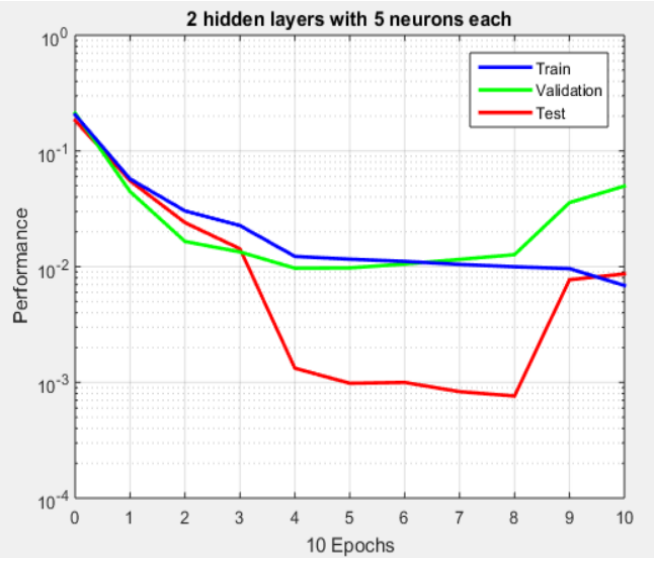

```
train_errorC2 =

    0.0115


validation_errorC2 =

    0.0145


test_errorC2 =

    0.0204


bestepochC2 =

    21
```



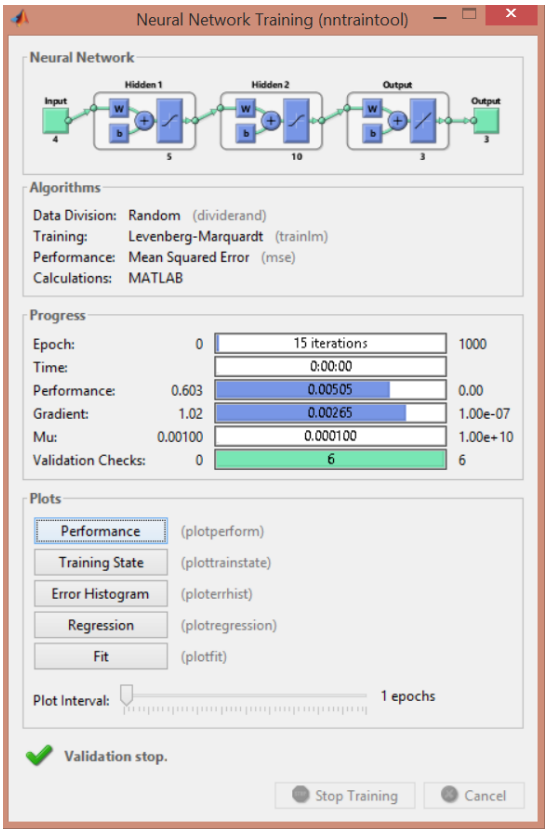
5 νευρώνες σε κάθε κρυφό επίπεδο :



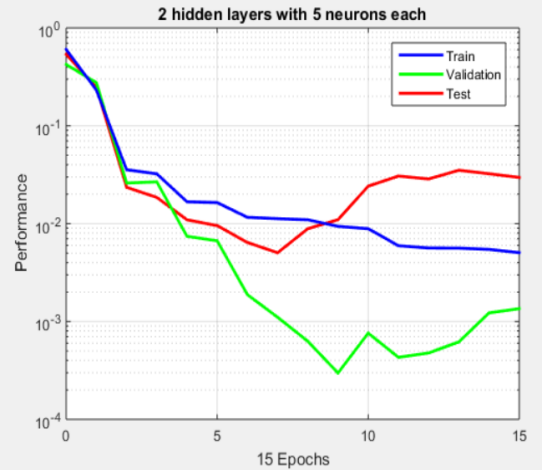

```
train_errorC2 =

    0.0123


validation_errorC2 =

    0.0097


test_errorC2 =

    0.0013


bestepochC2 =

     4
```



Ένα κρυφό επίπεδο με 5 νευρώνες και ένα με 10 :




```
train_errorC2 =

    0.0094

validation_errorC2 =

   2.9759e-04

test_errorC2 =

    0.0110

bestepochC2 =

     9
```



Βλέπουμε ότι γενικά παίρνουμε αρκετά καλά αποτελέσματα όσο αυξάνονται τα κρυφά επίπεδα και οι νευρώνες σε κάθε κρυφό επίπεδο, με καλύτερο αποτέλεσμα αυτό που πήραμε για 2 κρυφά επίπεδα με 5 νευρώνες το ένα και 10 στο άλλο.

Αν συγκρίνουμε συνολικά τα αποτελέσματα μπορούμε να καταλήξουμε ότι μία δομή νευρωνικού δικτύου χωρίς κανένα κρυφό επίπεδο δεν μας δίνει ικανοποιητικά αποτελέσματα, το training error κυμαίνεται 10-20% και το test error γύρω στο 50-55% ποσοστά που καθιστούν την ταξινόμηση απαγορευτική . Αν υπάρχει ένα κρυφό δίκτυο τα αποτελέσματα βελτιώνονται αρκετά καθώς αυξάνουμε τους νευρώνες (για 2 και 5 νευρώνες παίρνουμε σφάλματα γύρω στο 1%) αλλά μέχρι ένα σημείο διότι μετά υπάρχουν νευρώνες που δεν προσφέρουν τίποτα στο δίκτυο και το performance είναι της ίδιας τάξης. Με άλλα λόγια με 2 νευρώνες στο κρυφό επίπεδο πήραμε παρόμοια (και μάλιστα καλύτερα) αποτελέσματα με το δίκτυο που είχε 10 νευρώνες στο κρυφό δίκτυο, οπότε αυξήσαμε την πολυπλοκότητα της δομής χωρίς να έχουμε κανένα κέρδος στην απόδοση. Έπειτα εξετάζοντας τα αποτελέσματα που μας δίνουν 2 κρυφοί νευρώνες αυτά είναι αρκετά ικανοποιητικά αλλά είναι της ίδια τάξης με αυτά τα αποτελέσματα που παίρνουμε με 1 κρυφό νευρώνα. Οπότε πάλι με αυξημένη πολυπλοκότητα της δομής του νευρωνικού δικτύου δεν πετυχαίνουμε κάποιο αποτέλεσμα που να δηλώνει ότι αξίζει αυτή η δομή από την απλούστερη δομή του ερωτήματος Β. Συμπεραίνουμε ότι πρέπει να προσαρμόζουμε την πολυπλοκότητα του νευρωνικού μας δικτύου στις απαιτήσεις της κάθε ταξινόμησης . Δηλαδή για μία όχι τόσο απαιτητική ταξινόμηση δεν χρειάζεται να χρησιμοποιούμε πολύπλοκες δομές και να σπαταλάμε υπολογιστική ισχύ.