2η ΣΕΙΡΑ ΕΡΩΤΗΣΕΩΝ

ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ ΣΤΟΝ ΑΥΤΟΜΑΤΟ ΕΛΕΓΧΟ

Βουδριάς Δημήτριος

Ερώτηση 1

Ο κανόνας δέλτα (delta rule) γνωστός και ως αλγόριθμος των ελαχίστων μέσων τετραγώνων (least mean square - LMS) είναι ένας εποπτευόμενος κανόνας μάθησης. Η βασική ιδέα του κανόνα Δέλτα είναι να αλλάζει, για κάθε παρουσίαση του προτύπου εισόδου/επιθυμητής εξόδου από το σετ εκπαίδευσης, τα βάρη του δικτύου προς την κατεύθυνση που μειώνει το τετραγωνικό σφάλμα που ορίζεται ως:

$$E^{pat} = \sum_{i=1}^{q} E_i^{pat} = \sum_{i=1}^{q} \frac{1}{2}[D_I - Y_I]^2$$

Η ανανέωση των βαρών δίνεται από τον τύπο:

$$\Delta W_{IJ} = -\eta \frac{\partial E^{pat}}{\partial W_{ij}} = -\eta \frac{\partial E^{pat}}{\partial Y_I}\frac{\partial Y_I}{\partial W_{IJ}} = \eta[D_i - Y_i]\frac{\partial Y_I}{\partial W_{IJ}}$$

Από τον τελευταίο όρο προκύπτει ότι η συνάρτηση ενεργοποίησης πρέπει νά είναι συνεχής για να είναι διαφορίσιμη. Για να ξεπεραστεί ο συγκεκριμένος περιορισμός χρησιμοποιείται κατά την διάρκεια της εκπαίδευσης, μια γραμμική συνάρτηση ενεργοποίησης ***Y=W·X+bias.*** Αυτή η τροποποίηση κάνει την μάθηση πιο γρήγορη επειδή αλλάζει τα βάρη ακόμα και όταν η ταξινόμηση της εξόδου είναι σχεδόν σωστή.

Ερώτηση 2

Δοθέντος ενός δείγματος εκπαίδευσης το οποίο περιέχει ένα ζεύγος εισόδου - εξόδου $\{[x^{(k)}, d^{(k)}]\}$, k=1, 2, ..., p, ο αλγόριθμος παρέχει μια διαδικασία με την οποία αλλάζουν τα βάρη σε ένα δίκτυο back - propagation έτσι ώστε να ταξινομηθούν σωστά τα πρότυπα εισόδου. Η βάση για τον αλγόριθμο αυτό αλλαγής των βαρών είναι απλά η μέθοδος κατάβασης βαθμίδας (gradient – descent), όπως χρησιμοποιήθηκε για απλά percerptons με διαφορίσιμες μονάδες.

Ο κανόνας ανανέωσης των βαρών προκύπτει από την παρακάτω μελέτη:
Αν πάρουμε ένα ζεύγος εισόδου - εξόδου (x^k, d^k) θεωρούμε ότι, *ο αλγόριθμος back - propagation, παρουσιάζει δυο φάσεις μετακίνησης των δεδομένων*.

(i) Το πρότυπο εισόδου x^k διαδίδεται από το επίπεδο εισόδου, στο επίπεδο εξόδου και με αποτέλεσμα την προς τα εμπρός διάδοσης των δεδομένων και την δημιουργία μιας πραγματικής εξόδου y^k.
(ii) Στη συνέχεια, τα σήματα σφάλματος που δημιουργούνται από τη διαφορά ανάμεσα στα d^k και y^k, διαδίδονται προς τα πίσω από το επίπεδο εξόδου στα προηγούμενα επίπεδα, έτσι ώστε να μεταβάλλουν τα βάρη τους.

Εστω ενα Multilayer network με ένα κρυφό επίπεδο:

Σύμφωνα με τη μέθοδο gradient - descent, τα βάρη των συνδέσεων μεταξύ του κρυμμένου επίπεδου και του επιπέδου εξόδου μεταβάλλονται σύμφωνα με την σχέση:

$$\Delta W_{iq} = -\eta \frac{\partial E}{\partial W_{iq}}$$

Με την χρήση του κανόνα της αλυσίδας προκύπτει για το επίπεδο εξόδου:

$$\Delta W_{iq} = -\eta \frac{\partial E}{\partial y_i} \frac{\partial y_i}{\partial net_i} \frac{\partial net_i}{\partial w_{iq}} = \eta \delta_{oi} z_q$$

όπου δ_{oi} είναι ένα σήμα σφάλματος (ο διπλός δείκτης που έχει, δείχνει την μονάδα i στη συστοιχία εξόδου).

Όμοια για αλλαγή βαρών μεταξύ του επιπέδου εισόδου και του κρυμμένου επιπέδου, μεταξύ της j μονάδας εισόδου και της q μονάδας κρυμμένου επιπέδου:

$$\Delta v_{qj} = -\eta \frac{\partial E}{\partial v_{qj}} = -\eta \frac{\partial E}{\partial y_i} \frac{\partial y_i}{\partial z_q} \frac{\partial z_q}{\partial net_q} \frac{\partial net_q}{\partial v_{qj}} = \eta \delta_{hq} x_j$$

όπου δ_{hq} είναι το σήμα τοπικής βαθμίδας σφάλματος της μονάδας q στο κρυμμένο επίπεδο που ορίζεται ως:

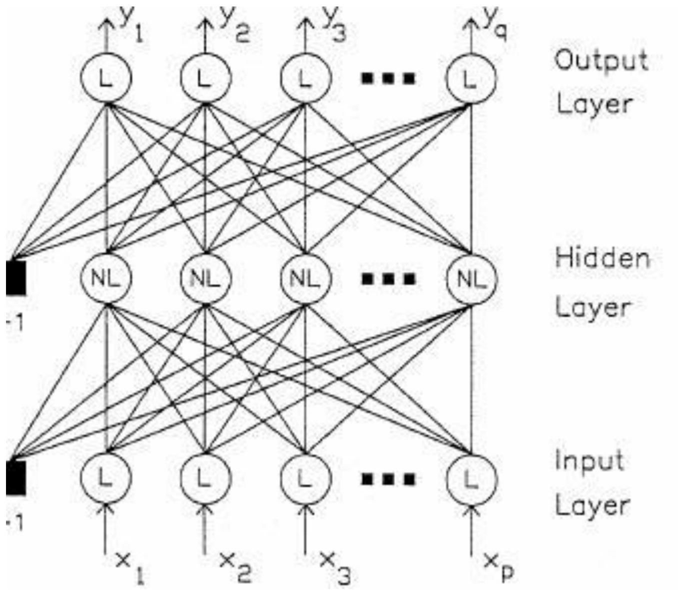
$$\delta_{hq} = -\eta \frac{\partial E}{\partial net_q} = -\eta \frac{\partial E}{\partial z_q} \frac{\partial z_q}{\partial net_q} = a(net_q) \sum_{i=1}^{n} \delta_{oi} w_{iq}$$

Το σήμα σφάλματος μιας μονάδας στο επίπεδο αυτό είναι διαφορετικό από το σήμα σφάλματος μιας μονάδας στο εξωτερικό επίπεδο, όπως φαίνεται και στις εξισώσεις. Παρατηρούμε από την τελευταία εξίσωση ότι το σήμα σφάλματος δ_{hq} μιας κρυμμένης μονάδας q μπορεί να οριστεί με τη βοήθεια των σημάτων σφαλμάτων δ_{oi} των μονάδων y_i , τις οποίες τροφοδοτεί. Οι συντελεστές είναι απλά τα βάρη που χρησιμοποιούνται για τη διάδοση προς τα μπροστά, αλλά εδώ είναι τα διαδιδόμενα σήματα σφάλματος (δ_{oi}) που διαδίδονται προς τα πίσω αντί των διαδιδόμενων προς τα μπροστά σημάτων. Αυτό παρουσιάζει ένα σημαντικό χαρακτηριστικό του αλγόριθμου back-propagation το γεγονός δηλαδή ότι **ο κανόνας αλλαγής των βαρών είναι τοπικός**. Αυτό σημαίνει ότι για να υπολογιστεί η αλλαγή των βαρών για μια δοθείσα σύνδεση, χρειαζόμαστε μόνο τις ποσότητες οι οποίες είναι διαθέσιμες στα δυο άκρα της σύνδεσης αυτής.

Ερώτηση 3

MLP (multilayer perceptrons)

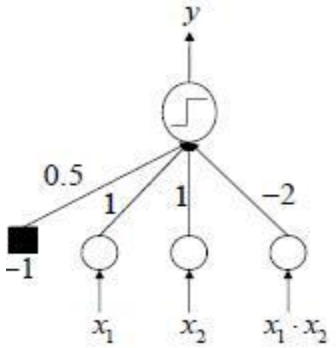
Το σχήμα δείχνει την ελάχιστη συνδεσμολογία για ένα perceptron πολλών επιπέδων. Στην συνδεσμολογία αυτή απαιτείται τουλάχιστον ένα επίπεδο με κρυμμένες μονάδες με μη γραμμικές συναρτήσεις ενεργοποίησης. Η χρήση των κρυμμένων μονάδων κάνει εφικτή την επανακωδικοποίηση των προτύπων εισόδου, δημιουργώντας έτσι μια διαφορετική αντιπροσώπευση. Κάθε κρυμμένο επίπεδο ξανακωδικοποιεί την είσοδο του.



Ανάλογα με τον αριθμό των κρυμμένων μονάδων, η νέα αντιπροσώπευση μπορεί να ανταποκριθεί σε διανύσματα τα οποία είναι στη συνέχεια γραμμικά διαχωρίσιμα. Εάν υπάρχουν πολύ λίγες μονάδες στο κρυμμένο επίπεδο για να δημιουργήσουν την απαραίτητη επανακωδικοποίηση, τότε χρειάζεται ίσως ένα πρόσθετο επίπεδο με κρυμμένες μονάδες.

HONN (High order Neural Networks)

Στα High order Neural Networks για να επιλυθεί το πρόβλημα της γραμμικής διαχωρισιμότητας ο διανυσματικός χώρος εισόδου περιέχει και γινόμενα των μεταβλητών εισόδου. Ενα τυπικό δίκτυο φαίνεται στο παρακάτω σχήμα:

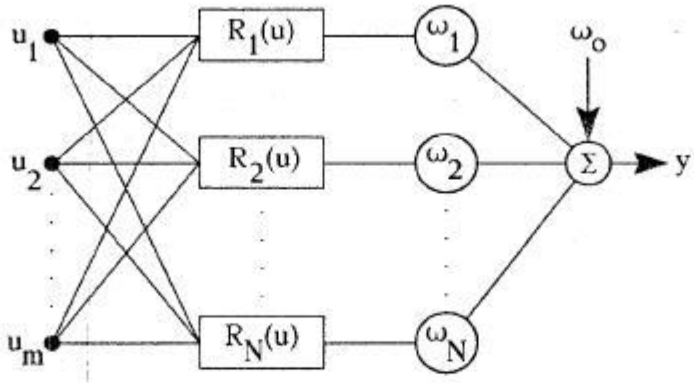


Γενικά, οι μονάδες υψηλότερης τάξης εφαρμόζουν τη συνάρτηση:

$$y_i = F(bias + \sum_j w_{ij}^{(1)} x_j + \sum_j \sum_k w_{ijk}^{(2)} x_j x_k + \sum_j \sum_k \sum_l w_{ijk}^{(3)} x_j x_k x_l + ...$$

RBF (Radial Basis Function Networks)

Ενα τυπικό δίκτυο παρουσιάζεται στο παρακάτω σχήμα :



Η ανάπτυξή τους στηρίχτηκε στο θεώρημα του Cover για την γραμμική διαχωρισιμότητα σύμφωνα με το οποίο:

« Σε ένα σύνθετο πρόβλημα ταξινόμησης προτύπων σε τάξεις *(clusters)* είναι πιθανότερο να επιτευχθεί γραμμική διαχωρισιμότητα όταν αυτό μετασχηματιστεί (με μη γραμμικό τρόπο) σε ένα χώρο υψηλής διάστασης.»

Για κάθε πρότυπο **εισόδου** ορίζεται ένα διάνυσμα, που αποτελείται από ένα σύνολο πραγματικών συναρτήσεων $R(u)=[R_1(u), R_2(u) \ldots R_N(u)]$. Η συνάρτηση εξόδου δίνεται ώς:

$$y=\sum_{i=1}^{N} w_i R_i(u)+w_o$$

Ερώτηση 4

Εστω η σιγμοειδής συνάρτηση $\varphi(u)=\dfrac{1}{1+e^{-au}}$ που ορίζεται στο διάστημα [0,1].

Εκφράζουμε τη σχέση $e^{-au}=\dfrac{1-\varphi(u)}{\varphi(u)}$

Για την παράγωγο ισχύει: $\varphi(u)^{'}=\dfrac{\partial\varphi(u)}{\partial u}=-\dfrac{(1+e^{-au})^{'}}{(1+e^{-au})^2}=\dfrac{ae^{-au}}{e^{-2au}+2e^{-au}+1}$

Αντικαθιστώντας έχουμε:

$$\varphi(u)^{'}=\frac{a\dfrac{1-\varphi(u)}{\varphi(u)}}{\dfrac{(1-\varphi(u))^2}{\varphi(u)^2}+2\dfrac{1-\varphi(u)}{\varphi(u)}+1}=\frac{a(1-\varphi(u))}{\dfrac{(1-\varphi(u))^2}{\varphi(u)}+2(1-\varphi(u))+\varphi(u)}\Rightarrow$$

$$\varphi(u)^{'}=\frac{a(1-\varphi(u))\cdot\varphi(u)}{(1-\varphi(u))^2+2(1-\varphi(u))\cdot\varphi(u)+\varphi(u)^2}=\frac{a(1-\varphi(u))\cdot\varphi(u)}{1-2\varphi(u)+\varphi(u)^2+2\varphi(u)-2\varphi(u)^2+\varphi(u)^2}$$

$$\varphi(u)^{'}=a\varphi(u)\cdot(1-\varphi(u))$$

Ως αρχή θεωρούμε το 0.

$$\varphi(u)=\frac{1}{1+e^{-a0}}=\frac{1}{1+1}=\frac{1}{2}$$

$$\varphi(u)^{'}=a\varphi(u)\cdot(1-\varphi(u))=a\cdot\frac{1}{2}(1-\frac{1}{2})=\frac{a}{4}$$

Ερώτηση 5

Εστω η σιγμοειδής συνάρτηση $\varphi(u)=\dfrac{1-e^{-au}}{1+e^{-au}}=\tan(\dfrac{au}{2})$ που ορίζεται στο [-1,1].

Η λύση ως προς e^{-au} δίνει: $e^{-au}=\dfrac{1-\varphi(u)}{1+\varphi(u)}$

Για την παράγωγο ισχύει: $\varphi(u)^{'}=\dfrac{\partial\varphi(u)}{\partial u}=\dfrac{(1-e^{-au})^{'}(1+e^{-au})-(1-e^{-au})(1+e^{-au})^{'}}{(1+e^{-au})^{2}}\Rightarrow$

$$\varphi(u)^{'}=\frac{\partial\varphi(u)}{\partial u}=\frac{ae^{-au}(1+e^{-au})+(1-e^{-au})ae^{-au}}{(1+e^{-au})^{2}}=\frac{2ae^{-au}}{(1+e^{-au})^{2}}\Rightarrow$$

Αντικαθιστώντας έχουμε:

$$\varphi(u)^{'}=\frac{\partial\varphi(u)}{\partial u}=\frac{2a\dfrac{1-\varphi(u)}{1+\varphi(u)}}{(1+\dfrac{1-\varphi(u)}{1+\varphi(u)})^{2}}=\frac{2a\dfrac{1-\varphi(u)}{1+\varphi(u)}}{(\dfrac{2}{1+\varphi(u)})^{2}}=\frac{a}{2}(1+\varphi(u))^{2}$$

Ως αρχή θεωρούμε το 0

$$\varphi(u)=\frac{1-e^{-a0}}{1+e^{-a0}}=\frac{1-1}{1+1}=0$$

Τότε $\varphi(u)'=\dfrac{a}{2}(1+0)^{2}=\dfrac{a}{2}$

Αν η παράμετρος α γίνει πολύ μεγάλη θα χαθεί η ευαισθησία ως προς το όρισμα u και η συνάρτηση φ(u) θα συγκλίνει στο 0.

Ερώτηση 6

$$f(x_{1,}x_2) = x_1^{\ 2} + x_2^{\ 2} + 3x_1x_2 + \sin(x_1 + x_2) \text{ στο διάστημα } -2 \le x_1, x_2 \le 2$$

(α) Η ελάχιστη δομή δικτύου που θα μπορούσε να χρησιμοποιηθεί θα ήταν η εξής:

- δύο εισόδοι στο πρώτο επίπεδο για τα x_1,x_2
- ένα κρυφό επίπεδο τεσσάρων νευρώνων για κάθε έναν από τους όρους της συνάρτησης
- μονάδα εξόδου ενός νευρώνα

(β) Ακολουθεί ο ψευδοαλγόριθμος που περιγράφει τα βήματα που πρέπει να ακολουθήσουμε (παραγωγή δειγμάτων, εκπαίδευση δικτύου, έλεγχος εγκυρότητας αποτελεσμάτων, βελτίωση δομής κλπ) για την εκπαίδευση και δοκιμή του δικτύου.

Είσοδος

Παραγωγή διανυσμάτων εκπαίδευσης εισόδου και εξόδου.

Βήμα 1

Χωρισμός διανυσμάτων εισόδου και εξόδου αντίστοιχα σε ποσοστά 70% και 30 %.

Βήμα 2

Δημιουργία δικτύου.

1. Ορισμός πεδίου τιμών προτύπων
2. Ορισμός συναρτήσεων ενεργοποίησης
3. Τοπολογία δικτύου (νευρώνες για κάθε επίπεδο)
4. Ορισμός συνάρτησης εκπαίδευσης

Βήμα 3

Ορισμός παραμέτρων εκπαίδευσης και εκπαίδευση δικτύου

Βήμα 4

Ελεγχος εγκυρότητας: προσέγγιση επιθυμητού σφάλματος

Αν **ναι** : προσομοίωση δικτύου με διανύσματα δοκιμής

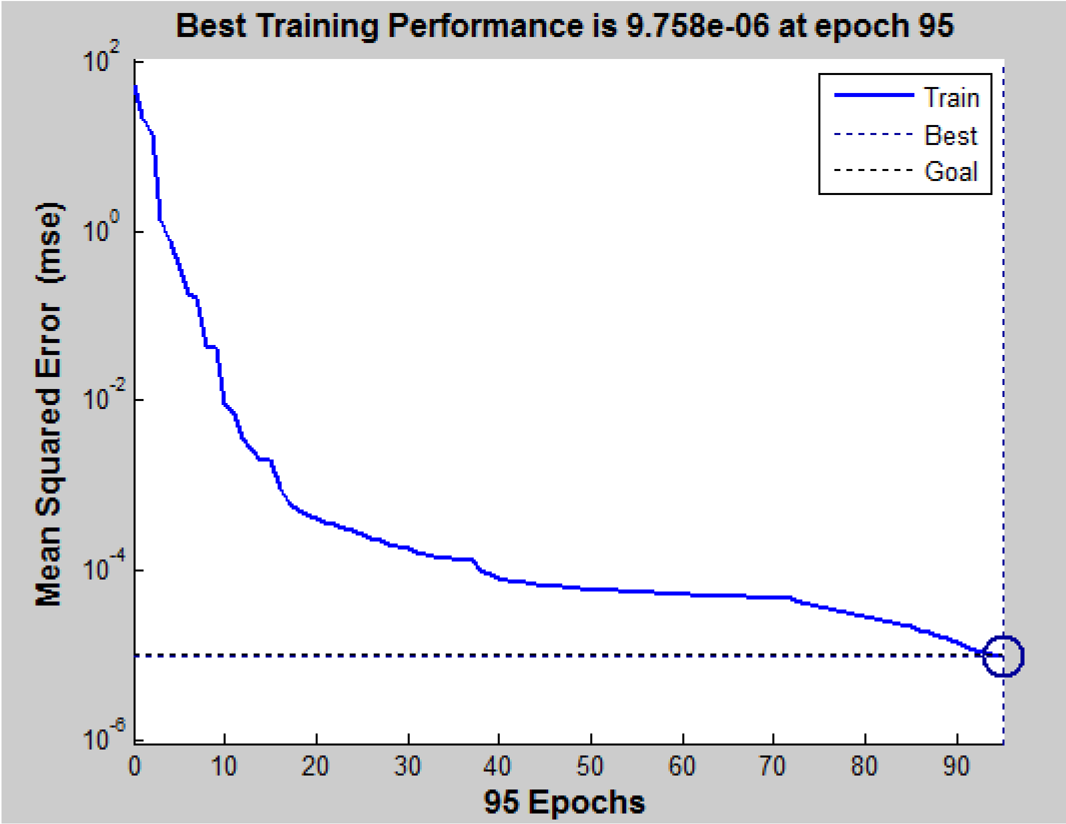
Αν **όχι** :

1. Αύξηση ή μείωση αριθμού νευρώνων ή επιπέδων
2. Αλλαγή παραμέτρων εκπαίδευσης
3. Αλλαγή συναρτήσεων ενεργοποίησης

(γ) Θα πρέπει να ληφθεί μνεία για το εύρος τιμών των δεδομένων εκπαίδευσης διότι θα καθορίσουν την εκπαίδευση του δικτύου.

(δ) Η δομή δικτύου επιλέχθηκε μετά από σειρά δοκιμών. Απο τις προσομοιώσεις φάνηκε ότι ένας ή δύο νευρώνες παραπάνω στο κρυφό επίπεδο δίνουν καλύτερα αποτελέσματα. Ετσι επιλέχθηκαν 6 νευρώνες στο κρυφό επίπεδο και εφαρμόστηκαν οι συνάρτησεις εκπαίδευσης 'trainlm' και 'traingd' όπου αποδείχθηκε ότι η πρώτη είχε καλύτερη προσέγγιση επιθυμητού σφάλματος. Για τις συναρτήσεις ενεργοποίησης στο κρυφό επίπεδο δοκιμάστηκε η σιγμοειδής tansig [-1 1] ενώ στην έξοδο η purelin (γραμμική). Με την παραπάνω μέθοδο επιτύχαμε σφάλμα εκπαίδευσης 10^{-5} όπως φαίνεται και από το παρακάτω σχήμα. Η ανάλυση πραγματοποιήθηκε στο πρόγραμμα Matlab και παρουσιάζεται παρακάτω:

```
% Δημιουργία διανυσμάτων εισόδου πολλαπλασιάζοντας με 2 για να καλύψει
% εύρος στο διάστημα [-2 2].
x1=2*rand(1,1000);
x2=2*rand(1,1000);
p=[x1;x2];

% Προσέγγιση της συνάρτησης εξόδου όπως ορίστηκε.
y=x1.^2+x2.^2+3.*x1.*x2+sin(x1+x2);
t=y;

% Διαχωρισμός δεδομένων εκπαίδευσης και επαλήθευσης σε ποσοστά 70% και 30%
p1=p(:,1:700);
p2=p(:,700:1000);    %  ποσοστό 30% του αρχικού δείγματος.
t1=t(:,1:700);
t2=t(:,700:1000);

% Δημιουργία του προσοτροφοδοτούμενου δικτύου
% Ορισμός των παραμέτρων εκπαίδευσης

net=newff(minmax(p),[6,1],{'tansig','purelin'},'trainlm');

net.trainParam.epochs=1000;
net.trainParam.goal=1e-5;
net.trainParam.lr=0.05;

% Εκπαίδευση και προσομοίωση του δικτύου.

[net,tr]=train(net,p1,t1);
a=sim(net,p2);
```



Ερώτηση 7

(a) Είναι γνωστό ότι για να εκπαιδεύσουμε τον γραμμικό αντιστοιχηστή **Y =W X** χρησιμοποιώντας τον κανόνα του Hebb, τα διανύσματα εισόδου πρέπει να είναι ορθογώνια το ένα σε σχέση με το άλλο. Οι πίνακες που δίνονται δεν μπορούν να ικανοποιήσουν αυτή τη συνθήκη.

(b) Το δίκτυο θα δέχεται για είσοδο τους πίνακες 10x10 ως ένα διάνυσμα εισόδου 100 στοιχείων. Στην έξοδο θα γίνεται η αναγνώριση με διάνυσμα εξόδου 10 στοιχείων για κάθε ένα από τα γράμματα. Αυτά θα οριστούν στο αρχείο Matlab ως εξής:

% Διανύσματα εισόδου

alphabet=[Aa;Bb;Dd;Ee;Gg;Hh;Ii;Jj;Kk;Mm];

```
% Διανύσματα εξόδου
A=[1 0 0 0 0 0 0 0 0 0];
B=[0 1 0 0 0 0 0 0 0 0];
D=[0 0 1 0 0 0 0 0 0 0];
E=[0 0 0 1 0 0 0 0 0 0];
G=[0 0 0 0 1 0 0 0 0 0];
H=[0 0 0 0 0 1 0 0 0 0];
I=[0 0 0 0 0 0 1 0 0 0];
J=[0 0 0 0 0 0 0 1 0 0];
K=[0 0 0 0 0 0 0 0 1 0];
M=[0 0 0 0 0 0 0 0 0 1];

targets=[A;B;D;E;G;H;I;J;K;M];
```

Το δίκτυο χρειάζεται 100 inputs και 10 νευρώνες στο επίπεδο εξόδου. Για το επίπεδο εξόδου επιλέγεται η log-sigmoid συνάρτηση επειδή το εύρος της [0 1] είναι ιδανικό για την αναγνώριση δυαδικών τιμών. Για το κρυφό επίπεδο η επιλογή νευρώνων είναι εμπειρική διαδικασία και εμείς θα επιλέξουμε 10 νευρώνες . Η εκπαίδευση θα γίνει αρχικά για ιδανικά διανύσματα εισόδου χωρίς θόρυβο.

<u>Αρχικοποίηση</u>

Δημιουργία του δικτύου.

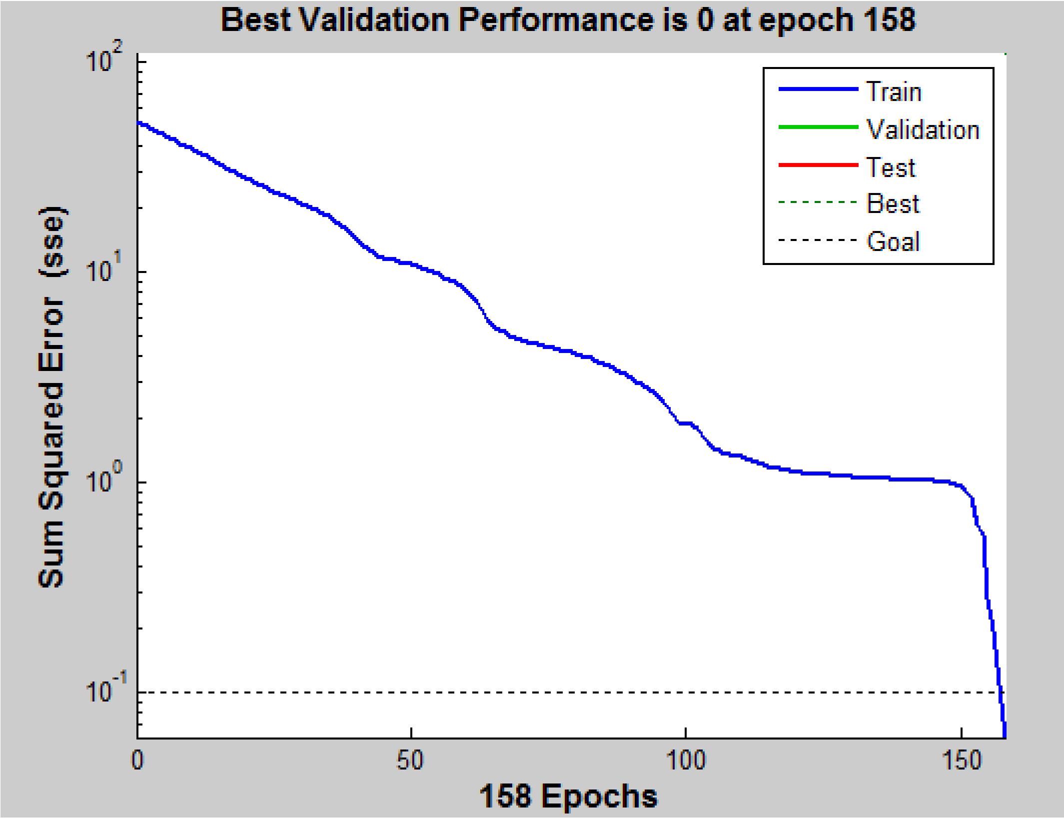
```
S1=10; % Κρυφό επίπεδο
[R,Q]=size(alphabet);
[S2,Q] = size(targets);
P=alphabet;
 net=newff(minmax(P),[S1 S2],{'logsig' 'logsig' },'traingdx');
```

<u>Εκπαίδευση του δικτύου χωρίς θόρυβο</u>

Το δίκτυο εκπαιδεύεται μέχρι να επιτευχθεί αθροιστικό τετραγωνικό σφάλμα της τάξης του 0.1.

```
T = targets;
net.performFcn = ’sse’;
net.trainParam.goal = 0.1;
net.trainParam.show = 20;
net.trainParam.epochs = 5000;
net.trainParam.mc = 0.95;
[net,tr] = train(net,P,T);
```

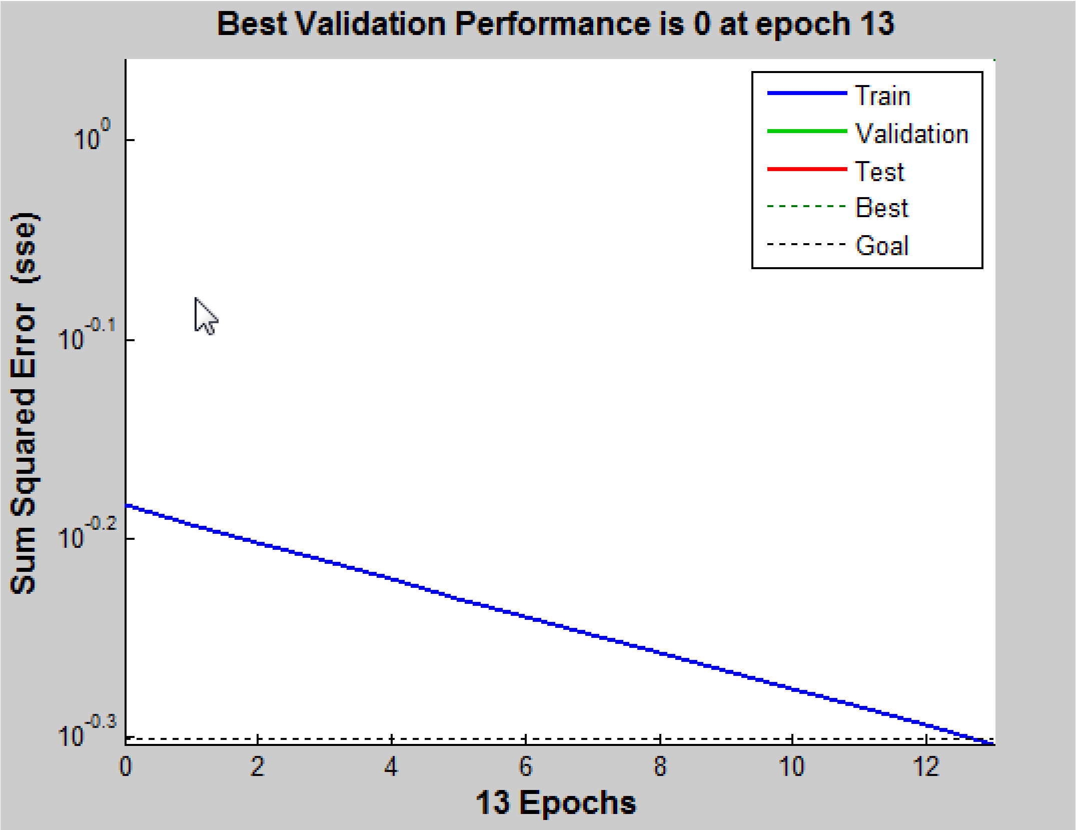
Παρακάτω φαίνεται το διάγραμμα προσέγγισης σφάλματος.



Εκπαίδευση του δικτύου με θόρυβο

Στο δίκτυο με θόρυβο εκπαιδεύουμε με δύο όμοια ιδανικά διανύσματα εισόδου και δύο διανύσματα με εισαγωγή θορύβου 0.1 και 0.2 αντίστοιχα. Αυτό ωθεί το δίκτυο να ανταποκρίνεται σε περιπτώσεις που τα γράμματα δεν είναι πολύ καθαρά. Σεαυτή την περίπτωση μειώνουμε τον αριθμό epochs και το επιθυμητό σφάλμα διότι εισάγωνται περισσότερα διανύσματα .

```
netn=net;
net.trainParam.goal=0.5;
net.trainParam.epochs=200;
T=[targets targets targets targets];
for pass = 1:10
P=[alphabet, alphabet,(alphabet+randn(R,Q)*0.1),(alphabet+randn(R,Q)*0.2)]
   [net,tr]=train(net,P,T);
end
```



Best Validation Performance is 0 at epoch 13
Train
Validation
Test
Best
Goal
Sum Squared Error (sse)
10^{0}
$10^{-0.1}$
$10^{-0.2}$
$10^{-0.3}$
0
2
4
6
8
10
12
13 Epochs