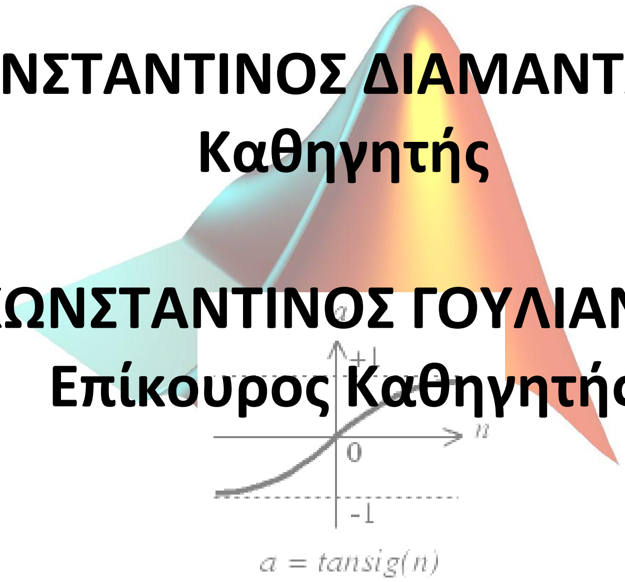
Εργαστηριακές Ασκήσεις Μηχανικής Μάθησης

ΚΩΝΣΤΑΝΤΙΝΟΣ ΔΙΑΜΑΝΤΑΡΑΣ
Καθηγητής

ΚΩΝΣΤΑΝΤΙΝΟΣ ΓΟΥΛΙΑΝΑΣ
Επίκουρος Καθηγητής



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΑΛΕΞΑΝΔΡΕΙΟ Τ.Ε.Ι. ΘΕΣΣΑΛΟΝΙΚΗΣ

Θεσσαλονίκη 2011

Περιεχόμενα

ΚΕΦΑΛΑΙΟ 1

Ο ΤΕΧΝΗΤΟΣ ΝΕΥΡΩΝΑΣ

1.1 Εισαγωγή

Σύμφωνα με το μοντέλο των McCulloch και Pitts η **έξοδος** του νευρώνα δίνεται από τη σχέση

$$v = f(u - \theta)$$

όπου

ϑ	το ***κατώφλι*** (***threshold***)
$u = \sum_{i=1}^{n} w_i x_i = \mathbf{w}^T \mathbf{x}$	η **δικτυακή διέγερση** του νευρώνα

με

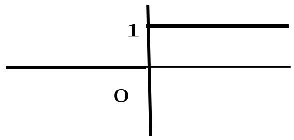
$\mathbf{w} = [w_1, ..., w_n]^T$	το διάνυσμα των **συναπτικών** βαρών
$\mathbf{x} = [x_1, ..., x_n]^T$	το διάνυσμα των **εισόδων**

Η **συνάρτηση ενεργοποίησης** (***neuron activation function***) *f* είναι μια συνάρτηση μιας εισόδου και μιας εξόδου και μπορεί να είναι μια από τις παρακάτω:

Βηματική 0/1

(step function 0/1)

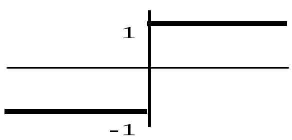
$$f(u-\theta) = \begin{cases} 0, & \alpha\nu \;\; u-\theta \le 0 \\ 1, & \alpha\nu \;\; u-\theta > 0 \end{cases}$$



Βηματική –1/1

(step function –1/1)

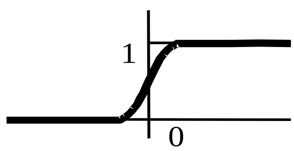
$$f(u-\theta) = \begin{cases} -1, & \alpha\nu \;\; u-\theta \le 0 \\ 1, & \alpha\nu \;\; u-\theta > 0 \end{cases}$$



Σιγμοειδής

(sigmoid)

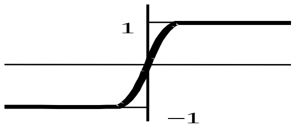
$$f(u-\theta) = \frac{1}{1+e^{-(u-\theta)}}$$



Υπερβολική εφαπτομένη

(hyperbolic tangent):

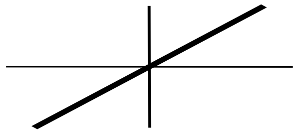
$$f(u-\theta)=\frac{1-e^{-(u-\theta)}}{1+e^{-(u-\theta)}}$$



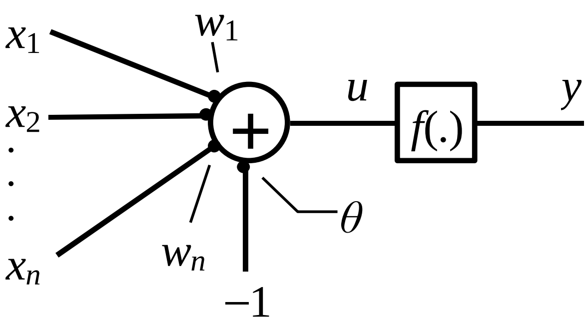
Γραμμική

(linear)

$$f(u-\theta)=u-\theta$$



Σχηματικά το παραπάνω μαθηματικό μοντέλο παριστάνεται από ένα αθροιστή ακολουθούμενο από ένα μη-γραμμικό μετασχηματιστή *f* όπως φαίνεται στο επόμενο σχήμα :



Παρατήρηση

- Το κατώφλι ϑ είναι ένας πραγματικός αριθμός (θετικός ή αρνητικός) όπως επίσης και τα συναπτικά βάρη w_1, ..., w_n. Επομένως, το **κατώφλι ϑ** μπορεί να θεωρηθεί σαν ένα **επί πλέον συναπτικό βάρος** w_{n+1} (το οποίο αποκαλείται **πόλωση**) συνδεδεμένο με μια **σταθερή είσοδο x_{n+1}** η οποία έχει πάντα την τιμή ***1*** ή ***–1***. Έτσι θα μπορούσαμε να γράψουμε

$$u=\sum_{i=1}^{n} w_i x_i-\theta=\sum_{i=1}^{n+1} w_i x_i$$

όπου $w_{n+1}=\vartheta$ και $x_{n+1}=-1$, οπότε θα έχουμε :

$\mathbf{w}=[w_1, w_2, ..., w_{n+1}]^T$ το διάνυσμα των **συναπτικών** βαρών με την πόλωση
$\mathbf{x}=[x_1, x_2, ..., x_n, -1]^T$ το διάνυσμα των **εισόδων** με τη σταθερή είσοδο -1

1.2 Προσομοίωση Τεχνητού Νευρώνα με Πόλωση (Άσκηση 1)

Να γίνει **πρόγραμμα** - ***script*** στο ***Matlab*** που να προσομοιώνει την παραπάνω διαδικασία. Πιο αναλυτικά το ***script ask1.m*** θα κάνει τα παρακάτω :

1. **Διαβάζει** τον **αριθμό** των Εισόδων *n*
2. **Δημιουργεί** με τη συνάρτηση *randn* τυχαίες τιμές στις συνάψεις $\mathbf{w}=[w_1, ..., w_{n+1}]^T$
3. **Δημιουργεί** με τη συνάρτηση *rand* τυχαίες τιμές στις εισόδους $\mathbf{x}=[x_1, ..., x_n]^T$

4. **Αποθηκεύει** το -1 στη n+1 θέση, οπότε το ***x*** γίνεται ***x*** = $[x_1, x_2, ..., x_n, -1]^T$
5. **Υπολογίζει** τη διέγερση $u = \sum_{i=1}^{n+1} w_i x_i = w^T \cdot x$
6. **Εμφανίζει** το παρακάτω ***menu*** επιλογών

Επιλογή Συνάρτησης

1. Βηματική 0/1
2. Βηματική -1/1
3. Σιγμοειδής
4. Υπερβολική Εφαπτομένη
5. Γραμμική

0. Τέλος

Δώσε Επιλογή (0..5) :

7. Ο χρήστης θα **δίνει** μια επιλογή choice
8. Αν η **επιλογή** είναι **< 0 ή > 5**, θα εμφανίζει μήνυμα **λάθους**
9. Αν η **επιλογή** είναι **0**, θα τερματίζει το ***menu*** επιλογών
10. Αν η **επιλογή** είναι **μεταξύ 1 και 5**, θα χρησιμοποιεί την αντίστοιχη συνάρτηση υπολογισμού του $v = f(u)$
11. Θα εμφανίζει την τιμή του $v = f(u)$, αν η επιλογή είναι **μεταξύ 1 και 5**.

Παρατηρήσεις

- Για το Βήμα 1 θα χρησιμοποιηθεί η εντολή **input**.
- Για το Βήμα 5 θα πρέπει να υπολογισθεί το Εσωτερικό Γινόμενο $u = w^T \cdot x$
- Αν τα a, b είναι δύο διανύσματα-στήλες (δηλαδή έχουν δηλωθεί σαν a(n,1) και b(n,1)) τότε το εσωτερικό τους γινόμενο ab θα είναι :

  ab = a**'***b

- Αν και τα δύο είναι διανύσματα-γραμμές, δηλαδή έχουν δηλωθεί σαν a(1,n) και b(1,n), τότε το a πρέπει να παραμείνει ως έχει και το b να αναστραφεί

  ab = a*b**'**

- Μπορεί επίσης να χρησιμοποιηθεί η συνάρτηση dot του MATLAB :

  ab = dot(a,b)

- Για το Βήμα 6 θα πρέπει να χρησιμοποιηθεί η εντολή

  fprintf('1. Step-01\n 2. step-11\n…');

- Για τα Βήματα 8, 9, 10, θα πρέπει να χρησιμοποιηθεί η εντολή switch-case, η οποία θα έχει την παρακάτω μορφή :

```
switch (choice)
case 1
    εντολές- Βηματικής 0/1;
case 2
    εντολές- Βηματικής -1/1;
case 3
    εντολές-Σιγμοειδούς;
case 4
    εντολές-Υπερβολικής Εφαπτομένης;
case 5
    εντολές-Γραμμικής;
case 0
    μήνυμα τέλους;
otherwise
    μήνυμα λάθους;
end % switch
```

- Για το Βήμα 11 θα πρέπει να χρησιμοποιηθεί ο τελεστής & για την εντολή if με διπλή συνθήκη ($1 \le choice \le 5$).

1.3 Προσομοίωση Τεχνητού Νευρώνα με Χρήση Συναρτήσεων – functions (Άσκηση 1b)

- Να τροποποιηθεί η Άσκηση 1, ώστε να χρησιμοποιεί τις παρακάτω συναρτήσεις - functions :

Αρχείο	*Συνάρτηση MATLAB*	*Μαθηματική Συνάρτηση*
step01.m	step01	Βηματική 0/1
step11.m	step11	Βηματική −1/1
sigmoid.m	sigmoid	Σιγμοειδής

Παρατηρήσεις

- Η συνάρτηση tanh δίνεται έτοιμη στο ίδιο το MATLAB ενώ η γραμμική συνάρτηση περιττεύει (δε χρειάζεται να υλοποιηθεί).
- Η συνάρτηση μπορεί να δεχτεί σαν είσοδο αριθμό ή διάνυσμα.
- Το όνομα της συνάρτησης θα πρέπει να είναι το ίδιο με το όνομα του αρχείου. Π.χ. το αρχείο **step01.m** θα περιέχει τα παρακάτω :

```
function y = step01(u)
if ( u > 0 )
  y = 1;
else
  y = 0;
end;
```

ΚΕΦΑΛΑΙΟ 2

PERCEPTRON

2.1 Εισαγωγή

- Το **perceptron** αποτελείται από ένα νευρώνα τύπου McCulloch και Pitts με n **εισόδους** και μία **έξοδο**.
- Χρησιμοποιεί σαν Συνάρτηση Ενεργοποίησης τη Βηματική 0/1.
- Μπορεί να **διαχωρίζει** πρότυπα 2 κλάσεων οι οποίες είναι **γραμμικά διαχωρίσιμες**.
- Εκτός από τα **πρότυπα** χρειάζονται και **στόχοι**, 0 για την πρώτη κλάση, 1 για τη δεύτερη.
- Είναι ένα δίκτυο που εκπαιδεύεται με **επίβλεψη**.
- Στην εκπαίδευση εισάγονται τα πρότυπα με τη σειρά.
- Η εισαγωγή όλων των προτύπων με τη σειρά αποκαλείται **εποχή**.
- Η έξοδος συγκρίνεται με τον αντίστοιχο στόχο και διορθώνονται οι συνάψεις.
- Οι συνάψεις τροποποιούνται σύμφωνα με τον **Κανόνα Δέλτα** (Delta Rule) :

$$w = w + \beta \cdot (d - v) \cdot x$$

όπου :

$w = [w_1, w_2, \ldots, w_{n+1}]^T$ = το διάνυσμα των **συναπτικών** βαρών
$x = [x_1, x_2, \ldots, x_n, -1]^T$ = το πρότυπο που εισάγεται κάθε φορά
d = ο στόχος, η κλάση στην οποία ανήκει το πρότυπο με τιμές 0, 1.
v = η έξοδος του νευρώνα με τιμές 0, 1.

- Η εκπαίδευση τελειώνει, όταν δεν διορθώνονται πλέον οι συνάψεις.

2.2 Διαχωρισμός Γραμμικά Διαχωρίσιμων Κλάσεων με Perceptron (Άσκηση 2)

Να γίνει **πρόγραμμα** - ***script*** στο ***Matlab*** που να διαχωρίζει στο επίπεδο τα πρότυπα 2 **Γραμμικά Διαχωρίσιμων** Κλάσεων με ένα **Perceptron** 2 εισόδων (3 με την πόλωση). Πιο αναλυτικά το ***script ask2.m*** θα κάνει τα παρακάτω :

1. **Διαβάζει** τον **αριθμό** των Προτύπων *n (άρτιος αριθμός)*
2. **Διαβάζει** το Συντελεστή Εκπαίδευσης (Learning Rate) *beta*
3. **Διαβάζει** το Μέγιστο Αριθμό Επαναλήψεων max_num_of_epochs
4. **Δημιουργεί** με τη συνάρτηση *randn* τυχαίες τιμές στις συνάψεις $w = [w_1, w_2, w_3]$.
5. **Δημιουργεί** με τη συνάρτηση *rand* τυχαίες τιμές για τα πρότυπα **pats**. Το κάθε πρότυπο αποτελείται από 2 τιμές x, y : Για τα πρότυπα της $1^{ης}$ κλάσης 1,2,…,n/2, θα πρέπει να ισχύει 0.0 ≤ x, y ≤ 0.4, ενώ για τα πρότυπα της $2^{ης}$ κλάσης n/2+1, n/2+2,…,n, θα πρέπει να ισχύει 0.5 ≤ x, y ≤ 0.9, έτσι ώστε να είναι γραμμικά διαχωρίσιμα**.**
6. **Δίνει** τις τιμές 0, 1 για τους στόχους $d = [d_1, d_2, \ldots, d_n]^T$
7. **Δίνει** αρχική τιμή στις εποχές 0.
8. **Για όσο** (Γίνονται αλλαγές στις συνάψεις) **και** (εποχή < max_num_of_epochs)

    Α. **Για κάθε πρότυπο i = 1:n**

    i. **Υπολογίζει** τη διέγερση u(i) και την έξοδο v(i)

    ii. **Αν** (v(i) ≠ d(i))

    Διορθώνει τις συνάψεις με τον Κανόνα του Δέλτα

    Τέλος Αν

    Τέλος Για

    Β. **Για κάθε πρότυπο i = 1:n**

    Υπολογίζει τη διέγερση u(i) και την έξοδο v(i) με τις τελικές συνάψεις

    Τέλος Για

    Γ. **Εμφανίζει** ένα γράφημα, το οποίο περιλαμβάνει τα παρακάτω 3 υπο-γραφήματα :

    i. Το γράφημα των προτύπων των 2 κλάσεων
    ii. Το γράφημα των προτύπων των 2 κλάσεων, ανάλογα με την κλάση στην οποία κατατάσσονται
    iii. Τις τιμές των εξόδων για το κάθε πρότυπο

9. **Δημιουργεί** δύο τυχαία πρότυπα, ένα απ' την κάθε κλάση

10. **Για κάθε πρότυπο i = 1:2**

    i. **Υπολογίζει** την έξοδο v(i)

    ii. **Εμφανίζει** το πρότυπο και την κλάση στην οποία κατατάσσεται

Παρατηρήσεις

- Για τα Βήματα 1, 2 και 3 θα χρησιμοποιηθεί η εντολή input
- Για το Βήμα 5 θα πρέπει να χρησιμοποιηθεί η συνάρτηση rand και να προβληθούν οι τιμές που δημιουργούνται στο διάστημα [0, 1] στο διάστημα [0.0, 0.4] και [0.5, 0.9]. Αν $x = rand(n/2,2) \in [0,1]$ και $y \in [0.0,0.4]$, αν $y = ax + b$, τότε το 0 απεικονίζεται στο 0 και το 1 στο 0.4. Παίρνουμε τις εξισώσεις $0 = a \cdot 0 + b$ και $0.4 = a \cdot 1 + b$ οπότε λύνοντας το σύστημα θα έχουμε $b = 0$ και $a = 0.4$, οπότε το y γίνεται $y = 0.4 * rand(n/2,2) + 0.0$ για τα πρότυπα της κλάσης 0.
- Για το Βήμα 6 θα πρέπει να χρησιμοποιηθούν οι συναρτήσεις zeros και ones.
- Για το Βήμα 8 μπορεί να χρησιμοποιηθεί μια μεταβλητή flag που θα ξεκινάει με την τιμή 0 και θα γίνεται 1, όταν γίνονται αλλαγές στις συνάψεις, ή μια μεταβλητή oldw που θα κρατάει τις προηγούμενες τιμές των συνάψεων και στο while θα συγκρίνεται το w με το oldw.
- Για το Βήμα 8Γ θα πρέπει να δηλώσουμε ένα νέο γράφημα με την εντολή figure(1) και να χρησιμοποιήσουμε την εντολή subplot.

  Με την εντολή subplot δηλώνουμε σε πόσες γραμμές και πόσες στήλες θα εμφανιστούν τα γραφήματα και το γράφημα στο οποίο αναφερόμαστε. Π.χ. με την εντολή :

  subplot(1, 2, 1);

  Δηλώνουμε ότι θα προβάλλουμε τα γραφήματα σε 1 γραμμή και 2 στήλες και ότι αναφερόμαστε στο γράφημα 1, ενώ με την εντολή :

  subplot(1, 2, 2);

  Δηλώνουμε ότι θα προβάλλουμε τα γραφήματα σε 1 γραμμή και 2 στήλες και ότι αναφερόμαστε στο γράφημα 2.

  Για το υπογράφημα 8Γi θα χρησιμοποιηθεί η συνάρτηση plot με παραμέτρους τις 2 πρώτες στήλες του πίνακα pats.

  Για το υπογράφημα 8Γii θα χρησιμοποιηθεί η συνάρτηση find που θα δώσει τους δείκτες των στοιχείων του πίνακα v που είναι 0 και 1 και να γίνει το γράφημα σύμφωνα μ' αυτά. Π.χ. η εντολή

  classa = find(v == 0);

  θα επιστρέψει στον πίνακα classa τους δείκτες των στοιχείων του πίνακα v που είναι 0, οπότε στην εντολή plot θα παραστήσουμε με ένα σύμβολο όλα αυτά τα στοιχεία που ταξινομούνται στην κλάση 0.

  Άν θέλουμε να σχεδιάσουμε και μια ευθεία που διαχωρίζει τα πρότυπα των 2 κλάσεων, χρησιμοποιούμε την εξίσωση ευθείας που είναι

  $$w_1 \cdot x + w_2 \cdot y - w_3 = 0$$

  οπότε λύνοντας ως προς y θα έχουμε

  y = -($w_1 \cdot x$ - w_3)/w_2

Αν λοιπόν χρησιμοποιήσουμε τον πίνακα x = 0.1:0.1:0.9, αφού οι τιμές των προτύπων ανήκουν σ' αυτή την κλίμακα, μπορούμε στο ίδιο γράφημα, αφού σχεδιάσουμε τα πρότυπα, να σχεδιάσουμε και την ευθεία με τις παρακάτω εντολές :

```
plot(pats(classa,1),... );
hold on;
x = 0.1:0.1:0.9;
y = -(w1 . x - w3)/w2;
plot(x, y);
```

.

Η εντολή hold on χρησιμοποιείται για να μη κλείσει το υπο-γράφημα, πριν σχεδιαστεί η ευθεία που θα διαχωρίζει τις 2 κλάσεις.

- Αν θέλουμε να ελέγξουμε τη συμπεριφορά του perceptron σε πρότυπα μη γραμμικά διαχωρίσιμων κλάσεων θα πρέπει να τροποποιήσουμε το βήμα 5, έτσι ώστε να εμφανίζει ένα ***menu*** επιλογών

  Επιλογή Προβλήματος

  1. Γραμμικά διαχωρίσιμες κλάσεις
  2. Μη γραμμικά διαχωρίσιμες κλάσεις
  3. Τέλος

  Δώσε Επιλογή (1..3) :

  Ο χρήστης θα **δίνει** μια επιλογή, η οποία, αν είναι 2 θα πρέπει να χρησιμοποιηθεί η συνάρτηση rand και να προβληθούν οι τιμές που δημιουργούνται στο διάστημα [0, 1] στο διάστημα [0.0, 0.6] και [0.5, 0.9].

ΚΕΦΑΛΑΙΟ 3

ADALINE

3.1 Εισαγωγή

- Το **Adaline**, όπως και το **Perceptron**, αποτελείται από ένα νευρώνα τύπου McCulloch και Pitts με n **εισόδους** και μία **έξοδο**.
- Χρησιμοποιεί σαν Συνάρτηση Ενεργοποίησης τη Γραμμική (Linear function) v = f(u) = u.
- Μπορεί να **διαχωρίζει** πρότυπα 2 κλάσεων οι οποίες είναι **γραμμικά ή μη γραμμικά διαχωρίσιμες**.
- Εκτός από τα **πρότυπα** χρειάζονται και **στόχοι** (-1 για την πρώτη κλάση, 1 για τη δεύτερη).
- Όπως και το Perceptron, το Adaline είναι ένα δίκτυο που εκπαιδεύεται με **επίβλεψη**, ενώ στην εκπαίδευση εισάγονται τα πρότυπα με τη σειρά για κάθε **εποχή**.
- Οι συνάψεις τροποποιούνται σύμφωνα με τον **Κανόνα Δέλτα** (Delta Rule) :

$$w = w + \beta \cdot (d - v) \cdot x$$

όπου :

$w = [w_1, w_2, ..., w_{n+1}]^T$ = το διάνυσμα των **συναπτικών** βαρών
$x = [x_1, x_2, ..., x_n, -1]^T$ = το πρότυπο που εισάγεται κάθε φορά
d = ο στόχος, η κλάση στην οποία ανήκει το πρότυπο με τιμές -1, 1.
v = η έξοδος του νευρώνα.

- Η εκπαίδευση τελειώνει, όταν το Μέσο Τετραγωνικό Σφάλμα των Προτύπων πάρει μια επιθυμητή τιμή. Το Μέσο Τετραγωνικό Σφάλμα ορίζεται σαν :

$$mse = \frac{1}{\#of\ patterns} \sum_{i=1}^{\#of\ patterns} (d(i) - v(i))^2$$

3.2 Διαχωρισμός Γραμμικά/Μη Γραμμικά Διαχωρίσιμων Κλάσεων με Adaline (Άσκηση 3)

Να γίνει **πρόγραμμα** - ***script*** στο ***Matlab*** που να διαχωρίζει στο επίπεδο τα πρότυπα 2 Γραμμικά/Μη Γραμμικά Διαχωρίσιμων Κλάσεων με ένα **Adaline** 2 εισόδων (3 με την πόλωση). Πιο αναλυτικά το ***script ask3.m*** θα κάνει τα παρακάτω :

1. **Διαβάζει** τον **αριθμό** των Προτύπων *n (άρτιος αριθμός)*
2. **Διαβάζει** το Συντελεστή Εκπαίδευσης (Learning Rate) *beta*
3. **Διαβάζει** το Μέγιστο Αριθμό Επαναλήψεων max_num_of_epochs
4. **Διαβάζει** το Ελάχιστο Μέσο Τετραγωνικό Σφάλμα min_mean_squared_error
5. **Δημιουργεί** με τη συνάρτηση *randn* τυχαίες τιμές στις συνάψεις $w = [w_1, w_2, w_3]$.
6. **Εμφανίζει** το παρακάτω ***menu*** επιλογών

   Επιλογή Προβλήματος

   1. Γραμμικά διαχωρίσιμες κλάσεις
   2. Μη γραμμικά διαχωρίσιμες κλάσεις 1
   3. Μη γραμμικά διαχωρίσιμες κλάσεις 2
   4. Τέλος

   Δώσε Επιλογή (1..3) :

7. Ο χρήστης θα **δίνει** μια επιλογή choice, ανάλογα με την οποία θα δημιουργεί και τα ανίστοιχα πρότυπα **pats**. Αν η επιλογή είναι 1, για τα πρότυπα της 1ης κλάσης 1,2,…,n/2, θα πρέπει να ισχύει 0.0 ≤ x, y ≤ 0.4, ενώ για τα πρότυπα της 2ης κλάσης n/2+1, n/2+2,…,n, θα πρέπει να ισχύει 0.5 ≤ x, y ≤ 0.9, έτσι ώστε να είναι γραμμικά διαχωρίσιμα**.** Αν η επιλογή είναι 2, για τα πρότυπα της 1ης κλάσης 1,2,…,n/2, θα πρέπει να ισχύει 0.0 ≤ x, y ≤ 0.6, ενώ για τα πρότυπα της 2ης κλάσης n/2+1, n/2+2,…,n, θα πρέπει να ισχύει 0.5 ≤ x, y ≤ 0.9, έτσι ώστε να είναι μη γραμμικά διαχωρίσιμα**.** Αν η επιλογή είναι 3, για τα πρότυπα της 1ης κλάσης 1:n/2, θα πρέπει να ισχύει 0.0 ≤ x, y ≤ 0.4, ενώ για τα πρότυπα της 2ης κλάσης n/2+1:3n/4, θα πρέπει να ισχύει 0.0 ≤ x ≤ 0.9 και 0.4 ≤ y ≤ 0.9, ενώ για τα πρότυπα της 2ης κλάσης 3n/4+1:n, θα πρέπει να ισχύει 0.5 ≤ x ≤ 0.9 και 0.0 ≤ y ≤ 0.5 , έτσι ώστε να είναι μη γραμμικά διαχωρίσιμα και να κυκλώνουν τα πρότυπα της 1ης κλάσης**.**
8. **Δίνει** τις τιμές -1, 1 για τους στόχους $d = [d_1, d_2, \ldots, d_n]^T$
9. **Δίνει** αρχική τιμή στις εποχές 0.
10. **Για όσο** (Σφάλμα > min_mean_squared_error) **και** (εποχή < max_num_of_epochs)

    A. **Για κάθε πρότυπο i = 1:n**

    i. **Υπολογίζει** τη διέγερση u(i) και την έξοδο v(i) = u(i)

    ii. **Υπολογίζει** το delta(i) = d(i) - v(i)

    iii. **Διορθώνει** τις συνάψεις σύμφωνα με τον Κανόνα του Δέλτα

    Τέλος Για

B. **Για κάθε πρότυπο i = 1:n**

i. **Υπολογίζει** τη διέγερση u(i) και την έξοδο v(i) = u(i)

ii. **Υπολογίζει** το delta(i) = d(i) - v(i)

iii. **Προσθέτει** στο sfalma το delta(i)^2

Τέλος Για

Γ. **Αυξάνει** την εποχή

Δ. **Ενημερώνει** το mse

Ε. **Εμφανίζει** ένα γράφημα, το οποίο περιλαμβάνει τα παρακάτω 4 υπο-γραφήματα :

i. Το γράφημα των προτύπων των 2 κλάσεων
ii. Το γράφημα των προτύπων των 2 κλάσεων, ανάλογα με την κλάση στην οποία κατατάσσονται
iii. Τις τιμές των εξόδων για το κάθε πρότυπο
iv. Το γράφημα του Μέσου Τετραγωνικού Σφάλματος στην κάθε εποχή

11. **Για την Ανάκληση, Δημιουργεί** δύο τυχαία πρότυπα, ένα απ' την κάθε κλάση

12. **Για κάθε πρότυπο i = 1:2**
    a. **Υπολογίζει** την έξοδο v(i)
    b. **Εμφανίζει** το πρότυπο και την κλάση στην οποία κατατάσσεται.

Παρατηρήσεις

- Για το Βήμα 10 μπορεί να χρησιμοποιηθεί ένα flag, το οποίο θα ξεκινάει με την τιμή 0 και το οποίο θα γίνεται 1, όταν το $mse = \frac{1}{n}\sum_{i=1}^{n}(d(i) - v(i))^2$ γίνει ≤ min_mean_squared_error.
- Για το Βήμα 10.Δ αποθηκεύουμε το sfalma/n στο mse(epoch).
- Για το Βήμα 10.Ε.ii θα πρέπει να χρησιμοποιηθεί η συνάρτηση **find** που θα δώσει τους δείκτες των στοιχείων του πίνακα v που είναι **κοντά** στο -1 και το 1 και να γίνει το γράφημα σύμφωνα μ' αυτά. Π.χ. η εντολή

  classb = find(y > 0);

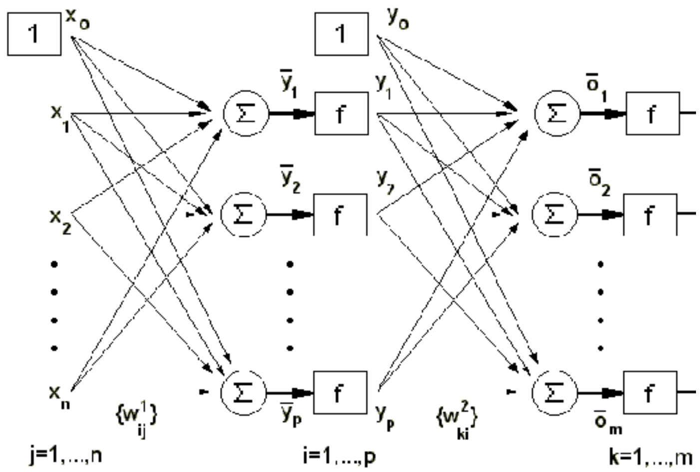
  θα επιστρέψει στον πίνακα classb τους δείκτες των στοιχείων του πίνακα v που είναι > 0, οπότε στην εντολή plot θα παραστήσουμε με ένα σύμβολο όλα αυτά τα στοιχεία που ταξινομούνται στην κλάση 1.
- Για το Βήμα 10.Ε.iv μπορεί να χρησιμοποιηθεί η εντολή **plot(mse).**
- Παρόμοια συνθήκη θα χρησιμοποιηθεί και στο Βήμα 12.b (Αν v > 0, το πρότυπο κατατάσσεται στην κλάση 1).

ΚΕΦΑΛΑΙΟ 4

BACK-PROPAGATION

4.1 Εισαγωγή

- Το Δίκτυο **Back-Propagation** αποτελείται από το στρώμα **εισόδου** με n εισόδους $x_1, x_2, ..., x_n$, συν την εξωτερική διέγερση $x_0 = 1$, ένα τουλάχιστον **κρυφό** στρώμα με p νευρώνες, συν την εξωτερική διέγερση $y_0 = 1$ και ένα στρώμα **εξόδου** με k νευρώνες, όπως φαίνεται στο επόμενο σχήμα :



1

- Χρησιμοποιεί **συνεχείς** Συναρτήσεις Ενεργοποίησης, όπως τη Σιγμοειδή, τη Γραμμική ή την Υπερβολική Εφαπτομένη.
- Μπορεί να **διαχωρίζει** πρότυπα 2 ή περισσότερων κλάσεων οι οποίες είναι **γραμμικά ή μη γραμμικά διαχωρίσιμες**.
- Εκτός από τα **πρότυπα** χρειάζονται και **στόχοι** (στην περίπτωση 2 κλάσεων 0 ή -1 για την πρώτη κλάση, 1 για τη δεύτερη, ανάλογα με τη συνάρτηση ενεργοποίησης στο στρώμα εξόδου).

- Είναι ένα δίκτυο που εκπαιδεύεται με **επίβλεψη**.
- Στην **εκπαίδευση** :

a. Εισάγονται τα πρότυπα με τη σειρά (για κάθε **εποχή**), πρώτα στο κρυφό στρώμα, απ' το οποίο η **έξοδος** χρησιμοποιείται σαν **είσοδος** στο στρώμα εξόδου, απ' το οποίο βγαίνει και η **τελική** έξοδος. Πιο αναλυτικά :

- Υπολογίζεται η **διέγερση** και η **έξοδος** κάθε νευρώνα $i,\ i = 1,2,...,p$ του κρυφού στρώματος :

$$\bar{y}_i = \sum_{j=0}^{n} x_j w_{ij}^1,\ \ y_i = f(\bar{y}_i),\ \ i = 1,2,...,p$$

- Υπολογίζεται η **διέγερση** και η **έξοδος** κάθε νευρώνα $k, k = 1,2,...,m$ του στρώματος εξόδου :

$$\bar{o}_k = \sum_{i=0}^{p} y_i w_{ki}^2,\ \ o_k = f(\bar{o}_k),\ \ k = 1,2,...,m$$

b. Υπολογίζονται τα **Δέλτα** του στρώματος εξόδου και μετά τα **Δέλτα** του κρυφού στρώματος σύμφωνα με τους τύπους :

$\delta_k = (d_k - o_k) \cdot f'(\bar{o}_k),\ \ k = 1,2,...,m$, τα **Δέλτα** του στρώματος εξόδου

$\bar{\delta}_i = \left(\sum_{k=1}^{m} \delta_k w_{ki}^2 \right) \cdot f'(\bar{y}_i),\ \ i = 1,2,...,p$, τα **Δέλτα** του κρυφού στρώματος

c. Τροποποιούνται οι συνάψεις του στρώματος εξόδου και μετά του κρυφού στρώματος σύμφωνα με τους τύπους:

$$w_{ki}^2 = w_{ki}^2 + beta \cdot \delta_k \cdot y_i = w_{ki}^2 + beta \cdot (d_k - o_k) \cdot f'(\bar{o}_k) \cdot y_i,\ k = 1,2,...,m,\ i = 1,2,...,p$$

$$w_{ij}^1 = w_{ij}^1 + beta \cdot \bar{\delta}_i \cdot x_j = w_{ij}^1 + beta \cdot \left(\sum_{k=1}^{m} \delta_k w_{ki}^2 \right) \cdot f'(\bar{y}_i) \cdot x_j,\ i = 1,2,...,p,\ j = 1,2,...,n$$

- Η εκπαίδευση τελειώνει, όταν το Μέσο Τετραγωνικό Σφάλμα των Προτύπων πάρει μια επιθυμητή τιμή. Το Μέσο Τετραγωνικό Σφάλμα ορίζεται σαν :

$$mse = \frac{1}{\#of\ patterns} \sum_{pat=1}^{\#of\ patterns} \sum_{i=1}^{m} (d_i^{(pat)} - o_i^{(pat)})^2$$

- Στην **ανάκληση** :

Εισάγονται τα πρότυπα με τη σειρά πρώτα στο κρυφό στρώμα, απ' το οποίο η **έξοδος** χρησιμοποιείται σαν **είσοδος** στο στρώμα εξόδου, απ' το οποίο βγαίνει και η **τελική** έξοδος, οπότε **εμφανίζει** το πρότυπο και την κλάση στην οποία κατατάσσεται.

4.2 Διαχωρισμός Γραμμικά/Μη Γραμμικά Διαχωρίσιμων Κλάσεων με Back-Propagation (Άσκηση 4)

Να γίνει **πρόγραμμα** - ***script*** στο ***Matlab*** που να διαχωρίζει στο επίπεδο τα πρότυπα 2 **Γραμμικά/Μη Γραμμικά Διαχωρίσιμων** Κλάσεων με ένα Δίκτυο **Back-Propagation** 2 εισόδων (3 με την πόλωση) ενός κρυφού στρώματος με 2 νευρώνες και ενός στρώματος εξόδου με 1 νευρώνα. Πιο αναλυτικά το ***script ask4.m*** θα κάνει τα παρακάτω :

1. **Διαβάζει** τον **αριθμό** των Προτύπων *n (άρτιος αριθμός)*
2. **Εμφανίζει** το παρακάτω ***menu*** επιλογών

   Επιλογή Προβλήματος

   1. Γραμμικά διαχωρίσιμες κλάσεις
   2. Μη γραμμικά διαχωρίσιμες κλάσεις 1
   3. Μη γραμμικά διαχωρίσιμες κλάσεις 2
   4. Μη γραμμικά διαχωρίσιμες κλάσεις 3 - xor
   5. Τέλος

   Δώσε Επιλογή (1..5) :

3. Ο χρήστης θα **δίνει** μια επιλογή choice, ανάλογα με την οποία θα δημιουργεί και τα ανίστοιχα πρότυπα **pats**. Αν η επιλογή είναι 1, για τα πρότυπα της 1ης κλάσης 1,2,…,n/2, θα πρέπει να ισχύει 0.0 ≤ x, y ≤ 0.4, ενώ για τα πρότυπα της 2ης κλάσης n/2+1, n/2+2,…,n, θα πρέπει να ισχύει 0.5 ≤ x, y ≤ 0.9, έτσι ώστε να είναι γραμμικά διαχωρίσιμα. Αν η επιλογή είναι 2, για τα πρότυπα της 1ης κλάσης 1,2,…,n/2, θα πρέπει να ισχύει 0.0 ≤ x, y ≤ 0.6, ενώ για τα πρότυπα της 2ης κλάσης n/2+1, n/2+2,…,n, θα πρέπει να ισχύει 0.5 ≤ x, y ≤ 0.9, έτσι ώστε να είναι μη γραμμικά διαχωρίσιμα. Αν η επιλογή είναι 3, για τα πρότυπα της 1ης κλάσης 1:n/2, θα πρέπει να ισχύει 0.0 ≤ x, y ≤ 0.4, ενώ για τα πρότυπα της 2ης κλάσης n/2+1:3n/4, θα πρέπει να ισχύει 0.0 ≤ x ≤ 0.9 και 0.4 ≤ y ≤ 0.9, ενώ για τα πρότυπα της 2ης κλάσης 3n/4+1:n, θα πρέπει να ισχύει 0.5 ≤ x ≤ 0.9 και 0.0 ≤ y ≤ 0.5 , έτσι ώστε να είναι μη γραμμικά διαχωρίσιμα και να κυκλώνουν τα πρότυπα της 1ης κλάσης. Αν η επιλογή είναι 4, για τα πρότυπα της 1ης κλάσης 1:n/4, θα πρέπει να ισχύει 0.0 ≤ x, y ≤ 0.4 και για τα πρότυπα της 1ης κλάσης n/4+1:n/2, θα πρέπει να ισχύει 0.5 ≤ x, y ≤ 0.9, ενώ για τα πρότυπα της 2ης κλάσης n/2+1:3n/4, θα πρέπει να ισχύει 0.5 ≤ x ≤ 0.9 και 0.0 ≤ y ≤ 0.4 και για τα πρότυπα της 2ης κλάσης 3n/4+1:n, θα πρέπει να ισχύει 0.0 ≤ x ≤ 0.4 και 0.5 ≤ y ≤ 0.9 , έτσι ώστε να είναι μη γραμμικά διαχωρίσιμα και να προσομοιώνουν την πύλη xor.
4. **Εμφανίζει** το παρακάτω ***menu*** επιλογών για την Επιλογή Συνάρτησης Ενεργοποίησης για το Κρυφό Στρώμα

   1. Σιγμοειδής
   2. Υπερβολική Εφαπτομένη
   3. Γραμμική
   4. Τέλος

   Δώσε Επιλογή (1..4) :

5. Ο χρήστης θα **δίνει** μια επιλογή fun1.

6. **Εμφανίζει** το ίδιο ***menu*** επιλογών για την Επιλογή Συνάρτησης Ενεργοποίησης για το Στρώμα Εξόδου και ο χρήστης θα **δίνει** μια επιλογή fun2.
7. **Δίνει** τις τιμές 0, 1 ή -1, 1 για τους στόχους $d = [d_1, d_2, ..., d_n]^T$, ανάλογα με την επιλογή fun2.
8. **Διαβάζει** το Συντελεστή Εκπαίδευσης (Learning Rate) *beta*
9. **Διαβάζει** το Μέγιστο Αριθμό Επαναλήψεων max_num_of_epochs
10. **Διαβάζει** το Ελάχιστο Μέσο Τετραγωνικό Σφάλμα min_mean_squared_error
11. **Δημιουργεί** με τη συνάρτηση *randn* τυχαίες τιμές στις συνάψεις $w1_{ij}$, $i = 1:3$, $j = 1:2$ και $w2_i$, $i = 1:3$ (Το δίκτυο θα έχει **2 εισόδους** συν την **πόλωση**, **2 νευρώνες** στο **κρυφό** στρώμα συν την **πόλωση** και **1 νευρώνα** στο στρώμα **εξόδου**).
12. **Δίνει** αρχική τιμή στις εποχές 0.
13. **Για όσο** (Σφάλμα > min_mean_squared_error) **και** (εποχή < max_num_of_epochs)

    Α. **Για κάθε πρότυπο i = 1:n**

    - **Υπολογίζει** τη **διέγερση** u1(k) και την **έξοδο** v1(k) κάθε νευρώνα $k, k = 1,2$ του κρυφού στρώματος ανάλογα με τη Συνάρτηση Ενεργοποίησης fun1 που επιλέχθηκε για το κρυφό στρώμα :

    $$u1_k = pats(i,:) \cdot w1(:,k), \;\; v1_k = fun1(u1_k), \;\; k = 1,2$$

    - Αποθηκεύει την τιμή 1 στο v1(3)
    - **Υπολογίζει** τη **διέγερση** u2 και την **έξοδο** v2 του νευρώνα εξόδου ανάλογα με τη Συνάρτηση Ενεργοποίησης fun2 που επιλέχθηκε για το στρώμα εξόδου :

    $$u2 = v1 \cdot w2', \;\; v2 = fun2(u2)$$

    - **Υπολογίζει** τα **Δέλτα** του στρώματος εξόδου και μετά τα **Δέλτα** του κρυφού στρώματος σύμφωνα με τους τύπους :

    $\delta = (d_i - v2) \cdot fun2'(u2)$, το **Δέλτα** του στρώματος εξόδου

    $\delta 1_k = \delta \cdot w2_k \cdot fun1'(u1_k)$, $k = 1,2$, τα **Δέλτα** του κρυφού στρώματος.

    - **Τροποποιεί** τις συνάψεις του στρώματος εξόδου και **μετά** του κρυφού στρώματος σύμφωνα με τους τύπους:

    $$w2_j = w2_j + beta \cdot \delta \cdot v1_j, \;\; j = 1..3$$

    $$w1_{jk} = w1_{jk} + beta \cdot \delta 1_k \cdot pats_{ij}, k = 1,2, \;\; j = 1..3$$

    Τέλος Για (κάθε πρότυπο i = 1:n)

B. **Για κάθε πρότυπο i = 1:n**

- **Υπολογίζει** τη **διέγερση** u1(k) και την **έξοδο** v1(k) κάθε νευρώνα $k, k = 1,2$ του κρυφού στρώματος ανάλογα με τη Συνάρτηση Ενεργοποίησης fun1 που επιλέχθηκε για το κρυφό στρώμα :

$$u1_k = pats(i,:) \cdot w1(:,k), \;\; v1_k = fun1(u1_k), \;\; k = 1,2$$

- Αποθηκεύει την τιμή 1 στο v1(3)

- **Υπολογίζει** τη **διέγερση** $u2_i$ και την **έξοδο** $v2_i$ του νευρώνα εξόδου ανάλογα με τη Συνάρτηση Ενεργοποίησης fun2 που επιλέχθηκε για το στρώμα εξόδου :

$$u2_i = v1 \cdot w2', \;\; v2_i = fun2(u2_i)$$

- Υπολογίζει τα **σφάλμα** του προτύπου και το αθροίζει στο συνολικό σφάλμα σύμφωνα με τον τύπο :

$$sfalma = sfalma + (d_i - v2_i)\hat{}2$$

Τέλος Για (κάθε πρότυπο i = 1:n)

Γ. Διαιρεί το συνολικό **σφάλμα** δια του αριθμού των προτύπων και το αποθηκεύει στη θέση epoch του πίνακα mse.

Δ. **Εμφανίζει** ένα γράφημα, το οποίο περιλαμβάνει τα παρακάτω 4 υπο-γραφήματα :

i. Το γράφημα των προτύπων των 2 κλάσεων
ii. Το γράφημα των προτύπων των 2 κλάσεων, ανάλογα με την κλάση στην οποία κατατάσσονται
iii. Τις τιμές των εξόδων για το κάθε πρότυπο
iv. το γράφημα του Μέσου Τετραγωνικού Σφάλματος στην κάθε εποχή

14. **Δημιουργεί** δύο τυχαία πρότυπα, ένα απ’ την κάθε κλάση, ανάλογα με την επιλογή του προβλήματος

15. **Για κάθε πρότυπο i = 1:2**

i. **Υπολογίζει** την έξοδο v2(i)

ii. **Εμφανίζει** το πρότυπο και την κλάση στην οποία κατατάσσεται

Παρατηρήσεις

- Αν θέλουμε να σχεδιάσουμε και δύο ευθείες που να διαχωρίζουν τα πρότυπα των 2 κλάσεων, χρησιμοποιούμε δύο εξισώσεις ευθείας (μια εξίσωση για κάθε νευρώνα του κρυφού στρώματος) που είναι

  $w1_{11} \cdot x_1 + w1_{21} \cdot y_1 + w1_{31} = 0$
  $w1_{12} \cdot x_2 + w1_{22} \cdot y_2 + w1_{32} = 0$

  οπότε λύνοντας ως προς y_1, y_2 θα έχουμε

  $y_1 = -(w1_{11} \cdot x_1 + w1_{31})/ w1_{21}$
  $y_2 = -(w1_{12} \cdot x_2 + w1_{32})/ w1_{22}$

- Στο Βήμα 13.Γ.iv το **γράφημα** του **Μέσου Τετραγωνικού Σφάλματος** στην κάθε εποχή μπορεί να εμφανισθεί με την εντολή :

  plot(mse);

- Στην Ανάκληση :
  - Δημιουργούμε πρότυπα και των 2 κλάσεων ανάλογα με την επιλογή του προβλήματος
  - Κάνουμε ότι και στο Βήμα 13.Β, **εκτός απ' τον Υπολογισμό του Σφάλματος.**
  - Για να το κατατάξουμε σε μία απ' τις 2 κλάσεις ελέγχουμε αν η έξοδος είναι > 0 ή > 0.5, ανάλογα με την επιλογή συνάρτησης ενεργοποίησης για το στρώμα εξόδου**.**

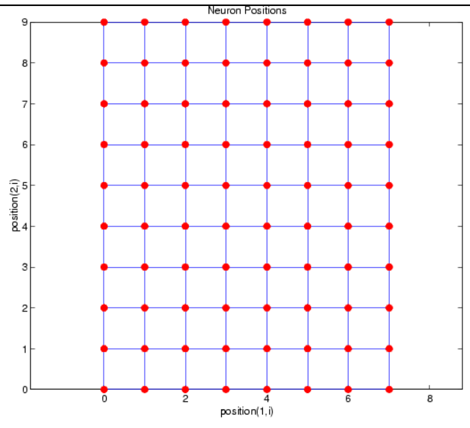
ΚΕΦΑΛΑΙΟ 5

ΑΥΤΟ-ΟΡΓΑΝΟΥΜΕΝΟΙ ΧΑΡΤΕΣ

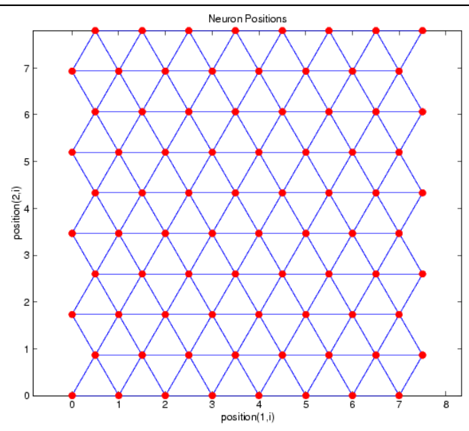
SELF-ORGANIZING MAPS (SOM)

5.1 Εισαγωγή

- Το Δίκτυο **SOM** είναι ένα δίκτυο **χωρίς επίβλεψη**.
- Στηρίζεται στην **τοπογραφική οργάνωση** του εγκεφάλου.
- Συνήθως αποτελούνται από ένα **δισδιάστατο** πλέγμα από νευρώνες.
- Υπάρχουν δύο τρόποι να τοποθετηθούν οι νευρώνες στον χώρο, η **τετραγωνική** και η **εξαγωνική**.



Τετραγωνική τοπολογία.



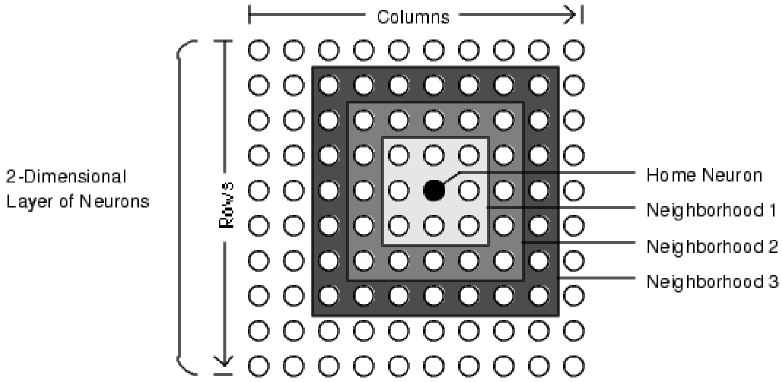
Εξαγωνική τοπολογία.

- Όταν εισάγεται ένα διάνυσμα εισόδου στο δίκτυο οι νευρώνες **ανταγωνίζονται** μεταξύ τους. **Νικητής (winner)** είναι εκείνος που ταιριάζει καλύτερα στο διάνυσμα εισόδου.

Πιο αναλυτικά :

- Υπολογίζεται η **Ευκλείδια** απόστασή του από όλους τους νευρώνες του δικτύου.

- Ο νευρώνας με την μικρότερη απόσταση ονομάζετει **winner**.
- Ο νευρώνας αυτός **μετακινείται** προς την κατεύθυνση του διανύσματος εισόδου.
- Το ίδιο γίνεται και για τους υπόλοιπους νευρώνες της **γειτονιάς** του.
- Το μήκος της μετακίνησης είναι ανάλογο του learning rate.
- Το learning rate στην αρχή είναι μεγάλο και μετά το μειώνουμε.
- Το ίδιο γίνεται και με το μέγεθος της γειτονιάς, όπως φαίνεται στο επόμενο σχήμα :.



- Στην ανάκληση : Όταν το SOM λάβει μια διέγερση ενεργοποιείται εκείνος ο νευρώνας που είναι πιο κοντά στο διάνυσμα εισόδου.

5.2 Διαχωρισμός Γραμμικά/Μη Γραμμικά Διαχωρίσιμων Κλάσεων με Δίκτυο SOM (Άσκηση 5)

Να γίνει **πρόγραμμα** - ***script*** στο ***Matlab*** που να διαχωρίζει στο επίπεδο τα πρότυπα 2 **Γραμμικά ή Μη Γραμμικά Διαχωρίσιμων** Κλάσεων με ένα Δίκτυο **SOM** 2 εισόδων και ενός στρώματος με **τουλάχιστον** 16 νευρώνες. Πιο αναλυτικά το ***script ask5.m*** θα κάνει τα παρακάτω :

1. **Διαβάζει** τον **αριθμό** των Προτύπων ***n*** *(άρτιος αριθμός πολλαπλάσιο του 4)*
2. **Διαβάζει** τον **αριθμό** των νευρώνων Kohonen στην κάθε γραμμή του δισδιάστατου πλέγματος **neurons**
3. **Διαβάζει** το Συντελεστή Εκπαίδευσης (Learning Rate) ***beta***
4. **Διαβάζει** το Μέγιστο Αριθμό Επαναλήψεων **max_num_of_epochs**
5. **Διαβάζει** τον **αριθμό** της γειτονιάς **geit** των νευρώνων Kohonen (γειτονιά 0 = 1 νευρώνας, γειτονιά 1 = 8 νευρώνες, γειτονιά 2 = 16 νευρώνες, γειτονιά 3 = 24 νευρώνες, κ.λ.π.)
6. **Διαβάζει** τον **αριθμό** της **τελικής** γειτονιάς **min_geit** των νευρώνων Kohonen (τελική γειτονιά 0 = 1 νευρώνας, τελική γειτονιά 1 = 8 νευρώνες)

7. **Δημιουργεί** με τη συνάρτηση *randn* τυχαίες τιμές στις συνάψεις w_{ij}, $i = 1..2$, $j = 1,2,...,neurons * neurons$. Το δίκτυο θα έχει **2 εισόδους** και $neurons * neurons$ νευρώνες στο στρώμα kohonen.
8. **Εμφανίζει** το παρακάτω ***menu*** επιλογών

   Επιλογή Προβλήματος

   1. Γραμμικά διαχωρίσιμες κλάσεις
   2. Μη γραμμικά διαχωρίσιμες κλάσεις 1
   3. Μη γραμμικά διαχωρίσιμες κλάσεις 2
   4. Μη γραμμικά διαχωρίσιμες κλάσεις 3 - xor
   5. Τέλος

   Δώσε Επιλογή (1..5) :

Ο χρήστης θα **δίνει** μια επιλογή choice, ανάλογα με την οποία θα δημιουργεί και τα ανίστοιχα πρότυπα **pats**.

9. **Εμφανίζει** ένα γράφημα, το οποίο περιλαμβάνει τα παρακάτω 2 υπο-γραφήματα :
   i. Το γράφημα των προτύπων των 2 κλάσεων
   ii. Το γράφημα των συνάψεων – νευρώνων kohonen

Πρώτη Φάση Εκπαίδευσης για 200 επαναλήψεις

10. **Για 200 επαναλήψεις**

    Για κάθε πρότυπο $i = 1,2,...,n$

    I. Βρίσκει την **απόσταση** του προτύπου i απ' τον κάθε νευρώνα $j = 1,2,...,neurons * neurons$.
    II. Βρίσκει την **ελάχιστη απόσταση** και τη θέση (winner) του νικητή νευρώνα.
    III. Αποθηκεύει τη θέση του νικητή νευρώνα στον πίνακα winners
    IV. Βρίσκει τη γραμμή και στήλη του νικητή στο δυσδιάστατο πλέγμα με τη χρήση της συνάρτησης mod :
        i. $j = mod(winner, neurons)$
        ii. $A\nu\ \ j = 0\ \ \tau\acute{o}\tau\varepsilon\ j = neurons$
        iii. $i = (winner - j) / neurons + 1$
    V. Μετακινεί κοντά στο πρότυπο εισόδου το νικητή νευρώνα και τους γείτονές του διορθώνοντας τις συνάψεις (Στη γειτονιά θα ανήκουν οι νευρώνες με δείκτες γραμμής $i - geit : i + geit$ και στήλης $j - geit : j + geit$).

    VI. **Εμφανίζει** ένα γράφημα, το οποίο περιλαμβάνει τα παρακάτω 4 υπο-γραφήματα :

i. Το γράφημα των προτύπων των 2 κλάσεων
ii. Το γράφημα των συνάψεων – νευρώνων kohonen, το νικητή νευρώνα και τους γείτονές του με διαφορετικό χρώμα
iii. Το τοπολογικό γράφημα των νευρώνων kohonen, το νικητή νευρώνα και τους γείτονές του με διαφορετικό χρώμα
ii. Το γράφημα του νικητή νευρώνα για το κάθε πρότυπο

Τέλος Για

Β. **Ενημερώνει** τον αριθμό νευρώνων της γειτονιάς, ώστε να γίνει ελάχιστη στο τέλος της Α' Φάσης Εκπαίδευσης.

Δεύτερη Φάση Εκπαίδευσης για 500*neurons*neurons επαναλήψεις

11. **Για όσο** (ο συντελεστής εκπαίδευσης δεν έχει μηδενισθεί) και (στη γειτονιά υπάρχουν περισσότεροι από ένας νευρώνες)

    Α. επαναλαμβάνει τις εντολές του Βήματος 10

    Β. **Ενημερώνει** το συντελεστή εκπαίδευσης

12. **Για την Ανάκληση, Δημιουργεί** n τυχαία πρότυπα, ανάλογα με την επιλογή του προβλήματος

13. **Για κάθε πρότυπο i = 1:n**

    i. **Υπολογίζει** την απόσταση του προτύπου από κάθε νευρώνα

    ii. **Εμφανίζει** το πρότυπο και τη θέση του νικητή νευρώνα.

Παρατηρήσεις

- Στο Βήμα 10.VI.iii εμφανίζει πρώτα το πλέγμα με neurons*neurons τετράγωνα και μετά το νικητή και τους γείτονες
- Στο Βήμα 11.Β θα πρέπει να χρησιμοποιηθούν οι συναρτήσεις mod και round
- Επειδή η εκπαίδευση απαιτεί πολλές επαναλήψεις, μπορεί να τροποποιηθεί το πρόγραμμα ώστε να εμφανίζεται ένα menu επιλογών :

1. Νέο Δίκτυο
2. Αποθηκευμένο Δίκτυο

- Αν η επιλογή είναι 1, θα αποθηκεύει σε κάποιο αρχείο τις παραμέτρους των Βημάτων 1-6, τις συνάψεις, τα πρότυπα και τους νικητές νευρώνες.
- Αν η επιλογή είναι 2, θα διαβάζει απ' το αρχείο τις παραμέτρους των Βημάτων 1-6, τις συνάψεις, τα πρότυπα και τους νικητές νευρώνες και θα συνεχίζει με εκπαίδευση – για κάποιον αριθμό επαναλήψεων που θα επιλέγει ο χρήστης - και ανάκληση.
- Για το σκοπό αυτό μπορούν να χρησιμοποιηθούν οι εντολές fopen('Όνομα Αρχείου', 'r'), fopen('Όνομα Αρχείου', 'w'), fclose, fprintf και fscanf.

ΚΕΦΑΛΑΙΟ 6

K-MEANS (ΑΛΓΟΡΙΘΜΟΣ K-ΜΕΣΩΝ)

6.1 Εισαγωγή

- Ο αλγόριθμος k-means ομαδοποιεί κάποια πρότυπα, ανάλογα με τη θέση τους στο επίπεδο ή στο χώρο, σε **ομάδες** που η καθεμιά αντιπροσωπεύεται από κάποιο **κέντρο**, ένα διάνυσμα ίδιας διάστασης με τα πρότυπα.
- Το κέντρο είναι ο **μέσος όρος** των προτύπων που ανήκουν σ΄ αυτή την ομάδα.
- Κάθε πρότυπο ανήκει στην ομάδα της οποίας το κέντρο βρίσκεται πιο κοντά στο πρότυπο, δηλαδή απέχει τη **μικρότερη απόσταση** απ' το κέντρο της.

6.2 Ο Αλγόριθμος K-Means (Άσκηση 6)

Να γίνει **πρόγραμμα** - ***script*** στο ***Matlab*** που να ομαδοποιεί τα πρότυπα 2 **Γραμμικά ή Μη Γραμμικά Διαχωρίσιμων** Κλάσεων σε όσα κέντρα επιλέξει ο χρήστης. Πιο αναλυτικά το ***script ask6.m*** θα κάνει τα παρακάτω :

1. **Διαβάζει** τον **αριθμό** των Προτύπων **n** *(άρτιος αριθμός πολλαπλάσιο του 4)*
2. **Διαβάζει** τον **αριθμό** των Κέντρων **k**
3. **Εμφανίζει** το παρακάτω ***menu*** επιλογών

    Επιλογή Προβλήματος

    1. Γραμμικά διαχωρίσιμες κλάσεις
    2. Μη γραμμικά διαχωρίσιμες κλάσεις 1
    3. Μη γραμμικά διαχωρίσιμες κλάσεις 2
    4. Μη γραμμικά διαχωρίσιμες κλάσεις 3 - xor
    5. Τέλος

    Δώσε Επιλογή (1..5) :

Ο χρήστης θα **δίνει** μια επιλογή choice, ανάλογα με την οποία θα δημιουργεί και τα ανίστοιχα πρότυπα **pats**.

4. **Εμφανίζει** το γράφημα των προτύπων των 2 κλάσεων.
5. Δημιουργεί k τυχαία κέντρα $c(j,1:2),\ j=1,2,...,k$ με τιμές στο [0, 0.9]
6. **Για όσο** τα κέντρα αλλάζουν (*c_old ~= c*)

Α. Για κάθε πρότυπο $i=1,2,...,n$

i. Βρίσκει την **απόσταση** του προτύπου i απ' το κάθε κέντρο $j,\ j=1,2,...,k$:

$$apostash(j) = norm(pats(i,:)-c(j,:))\verb|^|2$$

ii. Βρίσκει την **ελάχιστη απόσταση** του προτύπου i από όλα τα κέντρα $j,\ j=1,2,...,k$ χρησιμοποιώντας τη συνάρτηση min, η οποία επιστρέφει την **ελάχιστη** απόσταση και το **δείκτη** του κέντρου με την ελάχιστη απόσταση, ο οποίος αποθηκεύεται στον πίνακα deiktes με την εντολή :

$$[elaxisto\ \ deiktes(i)] = min(apostash);$$

Τέλος για

Β. Για κάθε πρότυπο $i=1,2,...,n$

i. Προσθέτει το πρότυπο i στον αντίστοιχο αθροιστή-κέντρο :

$$c(deiktes(i),1:2) = c(deiktes(i),1:2) + pats(i,1:2);$$

ii. Αυξάνει τον αντίστοιχο μετρητή-κέντρο $count(deiktes(i))$

Γ. Για κάθε κέντρο $j=1,2,...,k$

Αν ο αντίστοιχος μετρητής είναι ≠ 0,

Βρίσκει το νέο κέντρο = αθροιστής/μετρητής

Δ. Εμφανίζει ένα γράφημα, το οποίο περιλαμβάνει τα παρακάτω 2 υπο-γραφήματα :

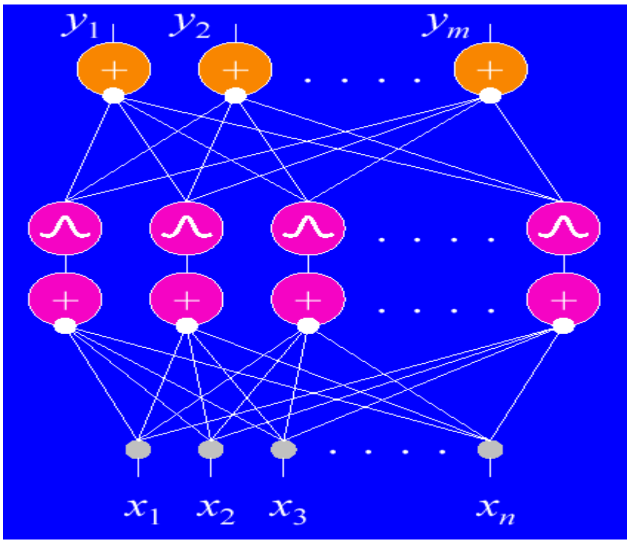
i. Το γράφημα των προτύπων των 2 κλάσεων
ii. Το γράφημα των προτύπων των 2 κλάσεων, και των αντίστοιχων κέντρων με διαφορετικό χρώμα

ΚΕΦΑΛΑΙΟ 7

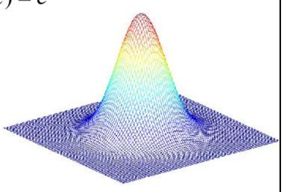
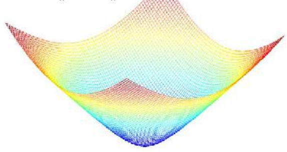
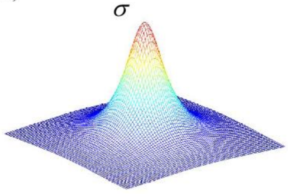
RBF-ΔΙΚΤΥΑ ΒΑΣΗΣ ΑΚΤΙΝΙΚΟΥ ΤΥΠΟΥ

7.1 Εισαγωγή

- Το Δίκτυο **RBF (Radial Basis Functions)** αποτελείται από το στρώμα **εισόδου** με n εισόδους $x_1, x_2, \ldots, x_n$, ένα **ενδιάμεσο** στρώμα με p νευρώνες και ένα στρώμα **εξόδου** με m νευρώνες, όπως φαίνεται στο επόμενο σχήμα :



- Μπορεί να **διαχωρίζει** πρότυπα 2 ή περισσότερων κλάσεων οι οποίες είναι **γραμμικά ή μη γραμμικά διαχωρίσιμες**.
- Εκτός από τα **πρότυπα** χρειάζονται και **στόχοι** (στην περίπτωση 2 κλάσεων -1 για την πρώτη κλάση, 1 για τη δεύτερη).
- Χρησιμοποιεί σαν Συναρτήσεις Ενεργοποίησης, τις συναρτήσεις ακτινικής βάσης (Radial Based Functions). Η τιμή των συναρτήσεων αυτών είναι συνάρτηση της απόστασης του διανύσματος εισόδου από ένα προκαθορισμένο κέντρο. Στα επόμενα σχήματα φαίνονται οι γραφικές παραστάσεις τριών συναρτήσεων RBF :

$f(x)=e^{-\frac{\|\mathbf{x}-\mathbf{c}\|^2}{\sigma^2}}$ 	$f(x)=(\|\mathbf{x}-\mathbf{c}\|^2+\sigma^2)^{1/2}$ 	$f(x)=\frac{(\|\mathbf{x}-\mathbf{c}\|^2+\sigma^2)^{-1}}{\sigma}$ 

- Έστω π.χ. ότι μια συνάρτηση έχει κέντρο c. Τότε, αν x είναι η είσοδος (το όρισμα δηλαδή) της συνάρτησης, η έξοδος της συνάρτησης RBF είναι τόσο μεγαλύτερη όσο πιο κοντά είναι το x στο c. Δηλαδή, όσο η διαφορά x-c μικραίνει τόσο πιο μεγάλη έξοδος παράγεται. Η συνάρτηση RBF έχει δηλαδή την ενδογενή δυνατότητα να «αναγνωρίζει» εκείνα τα ορίσματα που είναι κοντά στο κέντρο της.
- Το πλάτος της συνάρτησης (spread), δηλαδή το πόσο γρήγορα ή αργά θα «πέφτει» δεξιά και αριστερά από το κέντρο μπορεί να καθοριστεί ελεύθερα. Συνήθως όμως φροντίζουμε τα πλάτη να είναι ίσα μεταξύ τους: $\sigma=\frac{max\_d}{\sqrt{2n}}$, όπου d το πλάτος του διαστήματος προτύπων και n το πλήθος τους.
- Το **ενδιάμεσο** στρώμα χρησιμοποιείται για την **ομαδοποίηση** των προτύπων, ανάλογα με τη θέση τους στο επίπεδο ή στο χώρο, με την **εύρεση** των **κέντρων** των **ομάδων**, του μέσου όρου δηλαδή των διανυσμάτων της κάθε ομάδας. Το κάθε πρότυπο-διάνυσμα εισόδου κατατάσσεται στην ομάδα εκείνη που απέχει τη **μικρότερη απόσταση** απ' το κέντρο της.
- Το τμήμα του δικτύου που αποτελείται απ' το ενδιάμεσο στρώμα και το στρώμα εξόδου εκπαιδεύεται με **επίβλεψη**.
- Για τους νευρώνες του στρώματος εξόδου χρησιμοποιείται η γραμμική συνάρτηση ενεργοποίησης.

- Στην **εκπαίδευση** :
  i. Για το κάθε πρότυπο υπολογίζεται η Συνάρτηση της απόστασης του προτύπου από το κάθε κέντρο, η οποία αποτελεί και την είσοδο στο ενδιάμεσο στρώμα.
  ii. Γίνεται η εκπαίδευση όπως και στο δίκτυο Adaline.

- Στην **ανάκληση** :
  i. Εισάγεται το πρότυπο για ταξινόμηση
  ii. Υπολογίζεται η Συνάρτηση της απόστασης του προτύπου από το κάθε κέντρο.
  iii. Γίνεται η ανάκληση όπως και στο δίκτυο Adaline.

7.2 Διαχωρισμός Γραμμικά/Μη Γραμμικά Διαχωρίσιμων Κλάσεων με Δίκτυο RBF (Άσκηση 7)

Να γίνει **πρόγραμμα** - ***script*** στο ***Matlab*** που να διαχωρίζει στο επίπεδο τα πρότυπα 2 **Γραμμικά ή Μη Γραμμικά Διαχωρίσιμων** Κλάσεων με ένα Δίκτυο **RBF** 2 εισόδων (η πόλωση θα χρησιμοποιηθεί στο ενδιάμεσο στρώμα) ενός ενδιάμεσου στρώματος με όσους νευρώνες επιλέξει ο χρήστης και ενός στρώματος εξόδου με 1 νευρώνα. Πιο αναλυτικά το ***script ask7.m*** θα κάνει τα παρακάτω :

1. **Διαβάζει** τον **αριθμό** των Προτύπων **n** *(άρτιος αριθμός πολλαπλάσιο του 4)*
2. **Διαβάζει** τον **αριθμό** των Κέντρων **k**
3. **Εμφανίζει** το παρακάτω ***menu*** επιλογών

   Επιλογή Προβλήματος

   1. Γραμμικά διαχωρίσιμες κλάσεις
   2. Μη γραμμικά διαχωρίσιμες κλάσεις 1
   3. Μη γραμμικά διαχωρίσιμες κλάσεις 2
   4. Μη γραμμικά διαχωρίσιμες κλάσεις 3 - xor
   5. Τέλος

   Δώσε Επιλογή (1..5) :

Ο χρήστης θα **δίνει** μια επιλογή choice, ανάλογα με την οποία θα δημιουργεί και τα ανίστοιχα πρότυπα **pats**.

4. **Δημιουργεί** με τη συνάρτηση *randn* τυχαίες τιμές στις συνάψεις w_j, $j = 1,2,...,k+1$. (Το δίκτυο θα έχει **k εισόδους** συν την **πόλωση** και **1 νευρώνα** στο στρώμα **εξόδου**).
5. **Δίνει** τις τιμές -1, 1 για τους στόχους **d** = $[d_1, ..., d_n]^T$
6. **Εμφανίζει** το γράφημα των προτύπων των 2 κλάσεων.
7. Δημιουργεί k τυχαία κέντρα $c(j,1:2),\ j = 1,2,...,k$ με τιμές στο [0, 0.9]
8. **Για όσο** τα κέντρα αλλάζουν (*c_old ~= c)*

   Α. Για κάθε πρότυπο $i = 1,2,...,n$

   i. Βρίσκει την **απόσταση** του προτύπου i απ' το κάθε κέντρο $j,\ j = 1,2,...,k$:

      $$apostash(j) = norm(pats(i,:) - c(j,:))\text{^}2$$

   ii. Βρίσκει την **ελάχιστη απόσταση** του προτύπου i από όλα τα κέντρα $j,\ j = 1,2,...,k$ χρησιμοποιώντας τη συνάρτηση min, η οποία επιστρέφει την **ελάχιστη** απόσταση και το **δείκτη** του κέντρου με την ελάχιστη απόσταση, ο οποίος αποθηκεύεται στον πίνακα deiktes με την εντολή :

      [elaxisto deiktes(i)] = min(apostash);

   Τέλος για

Β. Για κάθε πρότυπο $i = 1,2,...,n$

i. Προσθέτει το πρότυπο i στον αντίστοιχο αθροιστή-κέντρο :

$$c(deiktes(i),1:2) = c(deiktes(i),1:2) + pats(i,1:2);$$

ii. Αυξάνει τον αντίστοιχο μετρητή-κέντρο $count(deiktes(i))$

Γ. Για κάθε κέντρο $j = 1,2,...,k$

Αν ο αντίστοιχος μετρητής είναι ≠ 0,

Βρίσκει το νέο κέντρο = αθροιστής/μετρητής

Δ. Εμφανίζει ένα γράφημα, το οποίο περιλαμβάνει τα παρακάτω 2 υπο-γραφήματα :

i. Το γράφημα των προτύπων των 2 κλάσεων
ii. Το γράφημα των προτύπων των 2 κλάσεων, και των αντίστοιχων κέντρων με διαφορετικό χρώμα

9. **Για κάθε** κέντρο *c(j):*

i. **Υπολογίζει** τις αποστάσεις μεταξύ τους
ii. **Βρίσκει** τη Μέγιστη Απόσταση max_d με τη συνάρτηση max
iii. **Υπολογίζει** το $\sigma = \dfrac{max\_d}{\sqrt{2n}}$

10. **Εμφανίζει** το παρακάτω ***menu*** επιλογών

Επιλογή Συνάρτησης RBF

1. e^((x-c)^2/c^2)
2. sqrt((x-c)^2+c^2)
3. 1/(σ*((x-c)^2+c^2))
4. Τέλος

Δώσε Επιλογή (1..4) :

Ο χρήστης θα **δίνει** μια επιλογή choice, ανάλογα με την οποία θα δημιουργεί και τα ανίστοιχα πρότυπα **pats**.

11. **Για κάθε** πρότυπο $i = 1,2,...,n$

Για κάθε κέντρο *c(j):*

Βρίσκει την Απόσταση x(i,j) του προτύπου απ' το κάθε κέντρο με τη χρήση της αντίστοιχης συνάρτησης RBF

12. **Δίνει** αρχική τιμή στις εποχές 0.

13. **Διαβάζει** το Συντελεστή Εκπαίδευσης (Learning Rate) *beta*

14. **Διαβάζει** το Μέγιστο Αριθμό Επαναλήψεων max_num_of_epochs

15. **Διαβάζει** το Ελάχιστο Μέσο Τετραγωνικό Σφάλμα min_mean_squared_error

16. **Για όσο** (Σφάλμα > min_mean_squared_error) **και** (εποχή < max_num_of_epochs)

    Α. **Για κάθε πρότυπο i = 1:n**

    i. **Υπολογίζει** την έξοδο v(i) = u(i)= x(i,:)* w

    ii. **Υπολογίζει** το delta(i) = d(i) - v(i)

    iii. **Διορθώνει** τις συνάψεις σύμφωνα με τον Κανόνα του Δέλτα

    Τέλος Για

    Β. **Για κάθε πρότυπο i = 1:n**

    i. **Υπολογίζει** την έξοδο v(i) = u(i)= x(i,:)* w

    ii. **Υπολογίζει** το delta(i) = d(i) - v(i)

    iii. **Προσθέτει** στο sfalma το delta(i)^2

    Τέλος Για

    Γ. **Αυξάνει** την εποχή

    Δ. **Ενημερώνει** το mse

    Ε. **Εμφανίζει** ένα γράφημα, το οποίο περιλαμβάνει τα παρακάτω 4 υπο-γραφήματα :

    i. Το γράφημα των προτύπων των 2 κλάσεων
    ii. Το γράφημα των προτύπων των 2 κλάσεων, ανάλογα με την κλάση στην οποία κατατάσσονται
    iii. Τις τιμές των εξόδων για το κάθε πρότυπο
    iv. Το γράφημα του Μέσου Τετραγωνικού Σφάλματος στην κάθε εποχή

17. **Για την Ανάκληση, δημιουργεί** δύο τυχαία πρότυπα, ένα απ' την κάθε κλάση ανάλογα με την επιλογή του προβλήματος

18. **Για κάθε πρότυπο i = 1:2**

    i. Εισάγεται το πρότυπο για ταξινόμηση

    ii. Υπολογίζεται η Συνάρτηση της απόστασης του προτύπου από το κάθε κέντρο.

    iii. Γίνεται η ανάκληση όπως και στο δίκτυο Adaline.

    iv. **Εμφανίζει** το πρότυπο και την κλάση στην οποία κατατάσσεται.