Τυπολόγιο Matlab

1 Οθόνη του Matlab

Η οθόνη του Matlab αποτελείται από τα εξής παράθυρα:

1.1 Command Window:

Είναι το παράθυρο στο οποίο εκτελούμε τις εντολές του Matlab για την δημιουργία πινάκων, το φόρτωμα δεδομένων από αρχείο, τις πράξεις μεταξύ πινάκων, κοκ.

1.2 Command History:

Σε αυτό το παράθυρο το Matlab αποθηκεύει τις εντολές τις οποίες εκτελούμε.

1.3 Current Directory:

Σε αυτό το παράθυρο φαίνεται το current directory του Matlab με τα περιεχόμενά του (τις συναρτήσεις, τα script και τα ".mat" αρχεία που περιέχει). Το matlab μας δίνει τη δυνατότητα να αλλάξουμε το current directory του.

1.4 Workspace:

Σε αυτό το παράθυρο βλέπουμε όλες τις μεταβλητές που είναι αποθηκευμένες στο Matlab.

2 Δημιουργία Πινάκων

2.1 Αναλυτ κά:

$\text{A=[1 2 3 4]} \rightarrow A = \begin{pmatrix} 1 & 2 & 3 & 4 \end{pmatrix}$

$\text{B=[1 2 ; 3 4]} \rightarrow A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$

2.2 Περ γραφ κά:

Χωρίς βήμα

$\text{C=[1:5]} \rightarrow \text{C=} \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \end{pmatrix}$

$\text{C=[1:5; 6:10]} \rightarrow \text{C=} \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \end{pmatrix}$

C=[] δημιουργεί έναν κενό πίνακα

Με βήμα

$\text{D = [0:20:100]} \rightarrow \text{D=} \begin{pmatrix} 0 & 20 & 40 & 60 & 80 & 100 \end{pmatrix}$

$\text{C=[1:5; 6:10]} \rightarrow \text{C=} \begin{pmatrix} 0 & 20 & 40 & 60 & 80 & 100 \\ 100 & 110 & 120 & 130 & 140 & 150 \end{pmatrix}$

2.3 Δημιουργία πίνακα χρησιμοποιώντας μια από τις έτοιμες συναρτήσεις της MATLAB:

E=zeros(2,4) $\rightarrow$ E=$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$ δημιουργεί πίνακα με όλα τα στοιχεία 0,

F=ones(3,3) $\rightarrow$ F=$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$ δημιουργεί πίνακα με όλα τα στοιχεία μονάδα,

G=eye(3) $\rightarrow$ C=$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ δημιουργεί μοναδιαίο πίνακα,

H=rand(3,2) $\rightarrow$ C=$\begin{pmatrix} 0.8147 & 0.9134 \\ 0.9058 & 0.6324 \\ 0.1270 & 0.0975 \end{pmatrix}$ δημιουργεί πίνακα στοιχείων με τυχαίες τιμές μεταξύ 0 και 1.

2.4 Δημιουργία πίνακα με χρήση δικής μας συνάρτησης:

I=myFunction(E,F)

3 Λήψη, Αλλαγή, Διαγραφή στοιχείου ή μέρους ενός πίνακα και Ανακάτεμα πίνακα

3.1 Λήψη

Έστω ο πίνακας $A=\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$ τότε με την εντολή:

- A(2,3) θα πάρουμε την τιμή που βρίσκεται στην δεύτερη γραμμή και δεύτερη στήλη του πίνακα, δηλαδή (6).
- A([1 2], [1 3]) θα πάρουμε τον υποπίνακα που προκύπτει αν κρατήσουμε από τον πίνακα A μόνο τις 2 πρώτες γραμμές και μόνο την $1^{η}$ και $3^{η}$ στήλη, δηλαδή $A=\begin{pmatrix} 1 & 3 \\ 4 & 6 \end{pmatrix}$.
- A([1 2], :) θα πάρουμε τον υποπίνακα που προκύπτει αν κρατήσουμε από τον πίνακα A μόνο τις 2 πρώτες γραμμές και όλες τις στήλες, δηλαδή $A=\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$.
- A(:, [1 3]) θα πάρουμε τον υποπίνακα που προκύπτει αν κρατήσουμε από τον πίνακα A όλες τις γραμμές και μόνο την $1^{η}$ και $3^{η}$ στήλη, δηλαδή $A=\begin{pmatrix} 1 & 3 \\ 4 & 6 \\ 7 & 9 \end{pmatrix}$.

3.2 Αλλαγή

Έστω ο πίνακας $A=\begin{pmatrix}1 & 2 & 3\\ 4 & 5 & 6\\ 7 & 8 & 9\end{pmatrix}$ τότε με την εντολή:

- A(2,3)=40 στην $2^{η}$ σειρά και $3^{η}$ στήλη έχει την τιμή 40. $A=\begin{pmatrix}1 & 2 & 3\\ 4 & 5 & 40\\ 7 & 8 & 9\end{pmatrix}$
- A([1 2], [1 3])=[40 41; 42 43] ο υποπίνακας που προκύπτει αν κρατήσουμε από τον πίνακα Α μόνο τις 2 πρώτες γραμμές και μόνο την $1^{η}$ και $3^{η}$ στήλη θα έχει τις τιμές 40, 41, 42 ,43 δηλαδή $A=\begin{pmatrix}40 & 2 & 41\\ 42 & 5 & 43\\ 7 & 8 & 9\end{pmatrix}$.
- Αντίστοιχα μπορούμε να αλλάξουμε τους υποπίνακες A([1 2], :) , A(:, [1 3])

3.3 Δ αγραφή

Έστω ο πίνακας $A=\begin{pmatrix}1 & 2 & 3\\ 4 & 5 & 6\\ 7 & 8 & 9\end{pmatrix}$ τότε με την εντολή:

- A([2,3],:)=[] Διαγράφεται η $2^{η}$ και η $3^{η}$ σειρά του πίνακα Α. Δηλαδή έχουμε $A=\begin{pmatrix}1 & 2 & 3\end{pmatrix}$
- A(:,2)=[] Διαγράφεται η $2^{η}$ στήλη του πίνακα Α. Δηλαδή έχουμε $A=\begin{pmatrix}1 & 3\\ 4 & 6\\ 7 & 9\end{pmatrix}$

3.4 Ανακάτεμα Π νακα

Έστω ο πίνακας $A=\begin{pmatrix}1 & 2 & 3\\ 4 & 5 & 6\\ 7 & 8 & 9\end{pmatrix}$ τότε με την εντολή:

- A=A(:, randperm(3)) παίρνουμε (τυχαία) τον πίνακα $A=\begin{pmatrix}2 & 1 & 3\\ 5 & 4 & 6\\ 8 & 7 & 9\end{pmatrix}$

4 Αποθήκευση Φόρτωμα δεδομένων από αρχείο:

save myData.mat: αποθηκεύει όλες τις μεταβλητές που υπάρχουν στο workspace στο current directory

load myData.mat: Φορτώνει της μεταβλητές που έχουν αποθηκευτεί στο αρχείο myData.mat στο current directory.

5 Πράξεις μεταξύ πινάκων

Έστω οι πίνακες $A=\begin{pmatrix}1 & 2\\ 3 & 4\end{pmatrix}$ και $B=\begin{pmatrix}1 & 2\\ 3 & 4\end{pmatrix}$, τότε:

πρόσθεση: $A+B\rightarrow\begin{pmatrix}2 & 4\\ 6 & 8\end{pmatrix}$

αφαίρεση: $A-B\rightarrow\begin{pmatrix}0 & 0\\ 0 & 0\end{pmatrix}$

πολλαπλασιασμός: $A*B\rightarrow\begin{pmatrix}7 & 10\\ 15 & 22\end{pmatrix}$

πολλαπλασιασμός ανά στοιχείο: $A.*B\rightarrow\begin{pmatrix}1 & 4\\ 9 & 16\end{pmatrix}$

διαίρεση ανά στοιχείο: $A./B\rightarrow\begin{pmatrix}1 & 1\\ 1 & 1\end{pmatrix}$

ύψωμα σε δύναμη ανά στοιχείο: $A.\hat{}\,B\rightarrow\begin{pmatrix}1 & 4\\ 27 & 256\end{pmatrix}$

ανάστροφος πίνακας: $A'\rightarrow\begin{pmatrix}1 & 3\\ 2 & 4\end{pmatrix}$

6 Χρήσιμες Πράξεις μεταξύ Πινάκων

Έστω οι πίνακες $A=\begin{pmatrix}1 & 2\\ 3 & 4\end{pmatrix}$ και $B=\begin{pmatrix}0 & 6\\ 3 & 8\end{pmatrix}$, τότε:

6.1 Πράξεις με συναρτήσεις

ημίτονο: $\sin(A)\rightarrow\begin{pmatrix}0.8415 & 0.9093\\ 0.1411 & -0.7568\end{pmatrix}$

συνημίτονο: $\cos(A)\rightarrow\begin{pmatrix}0.5403 & -0.4161\\ -0.9900 & -0.6536\end{pmatrix}$

6.2 Λογικές πράξεις

μεγαλύτερο : $A>B\rightarrow\begin{pmatrix}1 & 0\\ 0 & 0\end{pmatrix}$

μεγαλύτερο ή ίσο : $A>=B\rightarrow\begin{pmatrix}1 & 0\\ 1 & 0\end{pmatrix}$

ίσο : $A==B\rightarrow\begin{pmatrix}0 & 0\\ 1 & 0\end{pmatrix}$

μικρότερο: $A<B\rightarrow\begin{pmatrix}0 & 1\\ 0 & 1\end{pmatrix}$

6.3 Πράξε ς Μέγ στου Ελάχ στου:

μέγιστο στοιχείο ανά στήλη: $\max(A,[],1) \rightarrow \begin{pmatrix} 3 & 4 \end{pmatrix}$ είτε $\max(A) \rightarrow \begin{pmatrix} 3 & 4 \end{pmatrix}$

μέγιστο στοιχείο ανά γραμμή: $\max(A,[],2) \rightarrow \begin{pmatrix} 2 \\ 4 \end{pmatrix}$

μέγιστο στοιχείο πίνακα: $\max(\max(A)) \rightarrow 4$ είτε $\max(A(:)) \rightarrow 4$

ομοίως με min

6.4 Πράξε ς Άθρο σης:

άθροισμα στοιχείων ανά γραμμή: $\text{sum}(A,1) \rightarrow \begin{pmatrix} 4 & 6 \end{pmatrix}$είτε $\text{sum}(A) \rightarrow \begin{pmatrix} 4 & 6 \end{pmatrix}$

άθροισμα στοιχείων ανά στήλη: $\text{sum}(A,2) \rightarrow \begin{pmatrix} 3 \\ 7 \end{pmatrix}$

άθροισμα στοιχείων πίνακα: $\text{sum}(\text{sum}(A)) \rightarrow 10$ είτε $\text{sum}(A(:)) \rightarrow 10$

7 Αλληλουχία Πινάκων

Έστω οι πίνακες $A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$ και $B = \begin{pmatrix} 0 & 6 \\ 3 & 8 \end{pmatrix}$, τότε:

$\text{C= [A B]} \rightarrow \text{C=} \begin{pmatrix} 0 & 6 & 0 & 6 \\ 3 & 8 & 3 & 8 \end{pmatrix}$

$\text{C= [A ; B]} \rightarrow \text{C=} \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 0 & 6 \\ 3 & 8 \end{pmatrix}$

$\text{C=repmat(A,2,3)} \rightarrow \text{C=} \begin{pmatrix} 1 & 2 & 1 & 2 & 1 & 2 \\ 3 & 4 & 3 & 4 & 3 & 4 \\ 1 & 2 & 1 & 2 & 1 & 2 \\ 3 & 4 & 3 & 4 & 3 & 4 \end{pmatrix}$

8 Συναρτήσεις και script του Matlab

Κάθε script ή συνάρτηση του Matlab αποθηκεύεται σε ένα αρχείο με κατάληξη ".m". Για να εκτελεστεί αυτή η συνάρτηση πρέπει είτε να βρίσκεται στο Current Directory (παράγραφος 1.3) είτε σε κάποιο από τα paths που είναι προσβάσιμα από το Matlab (File→ Set Path…).

8.1 Δημ ουργ α M-File

Με δύο τρόπους

- Από το μενού File→New→M-File
- Μέσω του Command window με την εντολή
  - edit "όνομα συνάρτησης/script"

Τα περιεχόμενα του M-File όταν το σώσουμε θα αποθηκευτούν στο "Current Directory".

8.2 Script

Ένα script είναι ένα αρχείο το οποίο περιέχει μία ακολουθία εντολών π.χ.

```
A=[1:3;4:6];
B=sin(A);
```

Το script αυτό θα δημιουργήσει στο workspace τις μεταβλητές A και B με τιμές $A=\begin{pmatrix}1 & 2 & 3\\ 4 & 5 & 6\end{pmatrix}$ και $B=\begin{pmatrix}0.8415 & 0.9093 & 0.1411\\ -0.7568 & -0.9589 & -0.2794\end{pmatrix}$.

Σημείωση: Το ερωτηματικό μετά από κάθε εντολή έχει ως συνέπεια να μην εμφανίζεται το αποτέλεσμα εκτέλεσης της εντολής.

8.3 Συνάρτηση

Μία συνάρτηση είναι ένα αρχείο της μορφής:

```
function B=myfunction(A,C)
B=A.^2+sin(C);
```

Όπου οι μεταβλητές A, C είναι οι είσοδοι της συνάρτησης και B η τιμή της εξόδου της. Αν εκτελέσω π.χ. X=myfunction([1 2],[3 4]) θα πάρω το αποτέλεσμα $X=\begin{pmatrix}1.1411 & 3.2432\end{pmatrix}$.

Μία συνάρτηση μπορεί να επιστρέφει περισσότερες της μίας μεταβλητές. Π.χ.

```
function [B,D]=myfunction(A,C)
B=A.^2+sin(C);
D=A*3;
```

Αν εκτελέσω π.χ. [X,Y]=myfunction([1 2],[3 4]) θα πάρω το αποτέλεσμα $X=\begin{pmatrix}1.1411 & 3.2432\end{pmatrix}$, $Y=\begin{pmatrix}3 & 6\end{pmatrix}$

9 Έλεγχος Ροής

9.1 If

Έχει τη μορφή "**if** expression, statement, **elseif** expression, statement, **elseif … else** statement **end**". Για παράδειγμα το επόμενο script θα δώσει στη μεταβλητή B την τιμή 7.

```
A=3;
if (A==2)
    B=6;
elseif (A==3)
    B=7;
else
    (B==8)
end
```

9.2 For

Έχει τη μορφή "**for** x=initval:endval, statements, **end**". Για παράδειγμα το επόμενο script δημιουργεί την ακολουθία Fibonacci:

```
fibonacci=[0 1]
for x=1:10
    fibonacci(x+2)=fibonacci(x)+fibonacci(x+1);
end
```

9.3 While

Έχει τη μορφή "**while** expression, statements, **end**". Όσο ικανοποιείται η συνθήκη expression τότε εκτελούνται οι αντίστοιχες εντολές στο statements. Π.χ. το ακόλουθο script θα δώσει στο x την τιμή 10.

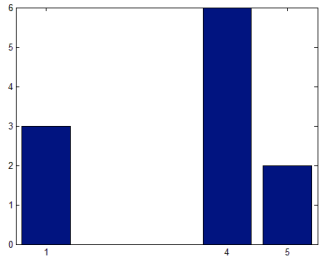
```
x=0;
while(x<10)
    x=x+1;
end
```

10 Χρήσιμες Συναρτήσεις

10.1 Bar

bar([1 4 5],[3 6 2])

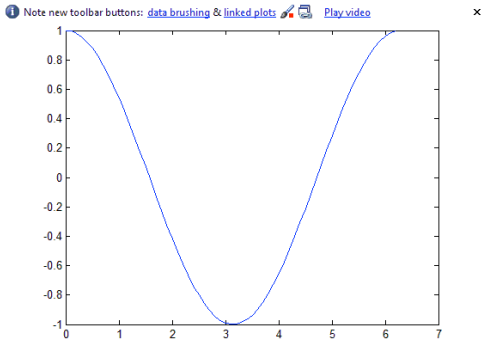
Θα δημιουργήσει ένα διάγραμμα bar που στις τιμές 1, 4, 5 στον άξονα των x θα έχει ύψος 3, 6, 2 αντίστοιχα.



10.2 Plot

A=0:0.1:2*pi, B=cos(A), plot(A,B)

Θα δημιουργήσει ένα διάγραμμα του οποίου οι τιμές στον άξονα του x προσδιορίζονται από τις τιμές του διανύσματος A και οι τιμές στον άξονα του y προσδιορίζονται από τις τιμές του διανύσματος B.



10.3 Find

Έστω ο πίνακας $A=\begin{pmatrix}1 & 0 & 0\\ 0 & 0 & 5\\ 0 & 0 & 0\end{pmatrix}$

Η συνάρτηση find επιστρέφει δύο ορίσματα τα οποία περιέχουν τις σειρές και τις στήλες που έχουν τιμή διάφορη του 0. Για παράδειγμα αν τρέξουμε [X,Y]=find(A) παίρνουμε $X=\begin{pmatrix}1\\2\end{pmatrix}$ και $Y=\begin{pmatrix}1\\3\end{pmatrix}$ ενώ αν τρέξουμε [X,Y]=find(A>2) θα πάρουμε $X=(2)$ και $Y=(3)$.

Σε περίπτωση που θέλουμε να περιορίσουμε τον αριθμό των ορισμάτων που θα επιστραφούν μπορούμε να το κάνουμε χρησιμοποιώντας ένα δεύτερο όρισμα το οποίο θέτει ένα άνω όριο στο μέγεθος των X, Y που θα επιστραφούν. Π.χ. [X,Y]=find(A,1) θα επιστρέψει $X=(1)$ και $Y=(1)$.

10.4 Removeconstantrows

Η συνάρτηση αυτή χρησιμοποιείται για να αφαιρέσουμε από έναν πίνακα τις γραμμές που έχουν σταθερή τιμή.

Έστω οι πίνακες $A=\begin{pmatrix}1&8&4&8\\2&5&1&8\\4&4&4&4\\2&1&0&1\end{pmatrix}$ και $B=\begin{pmatrix}4&9&1&1\\3&2&9&9\\8&3&6&6\\4&1&5&4\end{pmatrix}$:

- χρησιμοποιώντας την εντολή [A2,ps] = removeconstantrows(A) έχουμε $A2=\begin{pmatrix}1&8&4&8\\2&5&1&8\\2&1&0&1\end{pmatrix}$ ενώ η μεταβλητή ps περιέχει πληροφορίες που σχετίζονται με το ποιες σειρές διαγράφηκαν και ποια η σταθερή τιμή τους.
- Αυτή την πληροφορία μπορούμε να την χρησιμοποιήσουμε για να διαγράψουμε τις αντίστοιχες σειρές και από έναν άλλο πίνακα: Π.χ. με την εντολή B2=removeconstantrows('apply',B,ps) παίρνουμε $B2=\begin{pmatrix}4&9&1&1\\3&2&9&9\\4&1&5&4\end{pmatrix}$

10.5 Mapstd

Η συνάρτηση αυτή χρησιμοποιείται για να μετασχηματίσει έναν πίνακα σε έναν νέο πίνακα που τα στοιχεία κάθε γραμμής έχουν μέση τιμή 0 και διασπορά 1.

Έστω ότι έχω τον πίνακα $A=\begin{pmatrix}23&25&25\\19&28&17\end{pmatrix}$, $B=\begin{pmatrix}23&4\\13&28\end{pmatrix}$

- Εφαρμόζοντας τον μετασχηματισμό mapstd: [A2,ps]=mapstd(A) , παίρνω έναν νέο πίνακα A2 του οποίου τα στοιχεία ανά γραμμή έχουν μέση τιμή 0 και διασπορά 1. $A2=\begin{pmatrix}-1.1547&0.5774&0.5774\\-0.3982&1.1378&-0.7395\end{pmatrix}$. Η μεταβλητή ps μας παρέχει πληροφορίες για τον μετασχηματισμό.
- Ο αντίστροφος μετασχηματισμός μπορεί να εφαρμοστεί στο A2 ως εξής: A_new=mapstd('reverse',A2,ps) και ο πίνακας A_new θα είναι ίδιος με τον A.
- Επίσης μπορούμε να χρησιμοποιήσουμε την μεταβλητή ps για να μετασχηματίσουμε τον πίνακα B ο οποίος έχει τις ίδιες σειρές με τον A ως εξής B2=mapstd('apply',B,ps).

10.6 Processpca

Η συνάρτηση αυτή μας επιτρέπει σε ένα σύνολο δεδομένων να μειώσουμε τις διαστάσεις των δεδομένων χωρίς να χάσουμε σημαντική πληροφορία.

Έστω ο πίνακας A=rand(10,30):

- Χρησιμοποιώντας την εντολή [A2 ps]=processpca(A,0.02) παίρνουμε τον πίνακα A2 που προκύπτει από τον μετασχηματισμό του πίνακα A και τη

μεταβλητή ps που περιέχει πληροφορίες για το μετασχηματισμό. Όταν αυξήσουμε την τιμή 0.02 τότε μειώνονται περαιτέρω οι διαστάσεις των δεδομένων. Το αντίστροφο συμβαίνει όταν την μειώσουμε.

- Χρησιμοποιώντας την εντολή A_new=processpca('reverse',A2,ps) παίρνουμε τον αντίστροφο μετασχηματισμό.
- Χρησιμοποιώντας την εντολή B_new=processpca('apply',B,ps) μπορούμε να εφαρμόσουμε τον μετασχηματισμό και σε έναν πίνακα B.

10.7 Mapminmax

Η συνάρτηση αυτή μετασχηματίζει γραμμικά τα δεδομένα ενός πίνακα έτσι ώστε το μέγιστο στοιχείο ανά γραμμή να είναι 1 και το ελάχιστο -1. Έστω ότι έχω τον πίνακα $A=\begin{pmatrix}23 & 25 & 25\\ 19 & 28 & 17\end{pmatrix}$, $B=\begin{pmatrix}23 & 4\\ 13 & 28\end{pmatrix}$

- Εφαρμόζοντας τον μετασχηματισμό mapminmax: [A2,ps]= mapminmax(A), παίρνω έναν νέο πίνακα A2 του οποίου μέγιστο στοιχείο είναι 1 και το ελάχιστο -1. $A2=\begin{pmatrix}-1.0000 & 1.0000 & 1.0000\\ -0.6364 & 1.0000 & -1.0000\end{pmatrix}$. Η μεταβλητή ps μας παρέχει πληροφορίες για τον μετασχηματισμό.
- Ο αντίστροφος μετασχηματισμός μπορεί να εφαρμοστεί στο A2 ως εξής: A_new= mapminmax ('reverse',A2,ps) και ο πίνακας A_new θα είναι ίδιος με τον A.
- Επίσης μπορούμε να χρησιμοποιήσουμε την μεταβλητή ps για να μετασχηματίσουμε τον πίνακα B ο οποίος έχει τις ίδιες σειρές με τον A ως εξής mapstd('apply',B,ps).

11 Συναρτήσεις Νευρωνικού Δικτύου

11.1 Newff

Η συνάρτηση newff δημιουργεί ένα νέο νευρωνικό δίκτυο. Για παράδειγμα χρησιμοποιώντας την εντολή:

net = newff(TrainData,TrainDataTargets,[20 12]);

Δημιουργούμε ένα νευρωνικό δίκτυο το οποίο έχει στην είσοδό του τόσους νευρώνες όσες και οι γραμμές του πίνακα TrainData, στην έξοδό του τόσους νευρώνες όσες και οι γραμμές του πίνακα TrainDataTargets και περιέχει 2 κρυμμένα επίπεδα το πρώτο με 20 και το δεύτερο με 12 νευρώνες. Τα TrainData είναι τα πρότυπα που θέλουμε να μάθει το νευρωνικό μας και τα TrainDataTargets οι επιθυμητές έξοδοι για τα πρότυπα αυτά.

11.1.1 Συναρτήσεις Ενεργοποίησης

Το νευρωνικό δίκτυο που δημιουργήθηκε έχει σαν συνάρτηση ενεργοποίησης σε κάθε κρυμμένο επίπεδο την συνάρτηση 'tansig' και σαν συνάρτηση ενεργοποίησης στο επίπεδο εξόδου την συνάρτηση ' purelin'. Το Matlab μου δίνει τη δυνατότητα να αλλάξω τη συνάρτηση ενεργοποίησης κάθε επιπέδου ως εξής:

net = newff(TrainData,TrainDataTargets,[20 12],{'tansig' 'tansig' 'logsig'});

Εδώ όρισα σαν συνάρτηση ενεργοποίησης την tansig για τα κρυμμένα επίπεδα και την logsig για το επίπεδο εξόδου.

11.1.2 Συνάρτηση Εκπαίδευσης

Μπορώ να ορίσω την συνάρτηση εκπαίδευσης που θα χρησιμοποιεί το νευρωνικό μου δίκτυο ως εξής:

net = newff(TrainData,TrainDataTargets,[20 12],{'tansig' 'tansig' 'logsig'},'traingda');

Είτε στην περίπτωση που θέλω να κρατήσω τις default συναρτήσεις ενεργοποίησης ως εξής:

net = newff(TrainData,TrainDataTargets,[20 12],{},'traingda');

Διάφοροι αλγόριθμοι εκπαίδευσης είναι οι Batch Gradient Descent (traingd), Variable Learning Rate (traingda, traingdx), Levenberg-Marquardt (trainlm), Resilient Backpropagation (trainrp), Powell-Beale Restarts (traincgb).

11.1.3 Συνάρτηση Μάθησης

Η συνάρτηση μάθησης καθορίζει τον τρόπο που αλλάζουν τα βάρη και τα κατώφλια του νευρωνικού μας δικτύου. Χρησιμοποιείστε τις συναρτήσεις μάθησης με τη μέθοδο απότομης καθόδου (Gradient descent, 'learngd') και με τη μέθοδο απότομης καθόδου με προσθήκη του όρου ορμής (Gradient descent with momentum, 'learngdm').

net = newff(TrainData,TrainDataTargets,[20 12],{'tansig' 'tansig' 'logsig'},'traingda', 'learngd');

net = newff(TrainData,TrainDataTargets,[20 12],{},'', 'learngd');

11.1.4 Ρυθμός Μάθησης

Μπορούμε να αλλάξουμε τον ρυθμό μάθησης ενός MLP χρησιμοποιώντας την παράμετρο trainParam.lr. Π.χ.

net.trainParam.lr=0.02

11.1.5 Κριτήρια τερματισμού Εκπαίδευσης

Μπορούμε να αλλάξουμε τον μέγιστο αριθμό εποχών που θα εκπαιδευτεί το νευρωνικό μας δίκτυο με την παράμετρο net.trainParam.epochs. Π.χ. γράφοντας net.trainParam.epochs=200 καθορίζω πως μετά από 200 κύκλους εκπαίδευσης θα σταματήσει να εκπαιδεύεται το νευρωνικό μου.

Επίσης με την παράμετρο net.trainParam.goal σταματάει η εκπαίδευση του νευρωνικού δικτύου όταν η τιμή του μέσου τετραγωνικού σφάλματος πέσει κάτω από την τιμή goal. Π.χ. με net.trainParam.goal=0.1 αν έχω μέσω τετραγωνικό σφάλμα μικρότερο από 0.1 τερματίζεται η εκπαίδευση.

11.1.6 Τερματισμός με Early Stopping

Η μέθοδος Early Stopping χωρίζει τα δεδομένα μας σε ένα σύνολο δεδομένων εκπαίδευσης, ένα σύνολο δεδομένων επαλήθευσης και ένα σύνολο δεδομένων test. Στην περίπτωση που τα αποτελέσματα για τα δεδομένα επαλήθευσης δεν είναι ικανοποιητικά τότε τερματίζεται η εκπαίδευση του νευρωνικού δικτύου. Η αναλογία των δεδομένων καθορίζεται από τις παραμέτρους net.divideParam.trainRatio, net.divideParam.valRatio, net.divideParam.testRatio. Για παράδειγμα με :

net.divideParam.trainRatio=1.0

net.divideParam.valRatio=0

net.divideParam.testRatio=0

δεν έχω καθόλου δεδομένα επαλήθευσης και για testing οπότε στην περίπτωση αυτή η μέθοδος early stopping δεν εφαρμόζεται.

Η παράμετρος net.trainParam.max_fail καθορίζει τον μέγιστο αριθμό διαδοχικών εποχών που μπορώ να έχω χειρότερα αποτελέσματα για τα δεδομένα επαλήθευσης σε σχέση με προηγούμενες εποχές.

11.2 Train

Η συνάρτηση train εκπαιδεύει ένα νευρωνικό δίκτυο βάσει ενός συνόλου εισόδων και επιθυμητών αποκρίσεων ως εξής:

net=train(net,x,t)

Όπου x είναι οι είσοδοι και t οι επιθυμητές αποκρίσεις.

11.3 Sim

Η συνάρτηση έχει ορίσματα ένα νευρωνικό δίκτυο και ένα σύνολο εισόδων και η έξοδός της είναι η απόκριση του νευρωνικού δικτύου για το συγκεκριμένο σύνολο εισόδων.

```
y = sim(net,x);
```