ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

1η Άσκηση

ΟΝΟΜΑΤΕΠΩΝΥΜΟ : Μαρίνο Τσελάνι
ΑΜ : 20390241

ΕΑΡΙΝΟ ΕΞΑΜΗΝΟ 2024

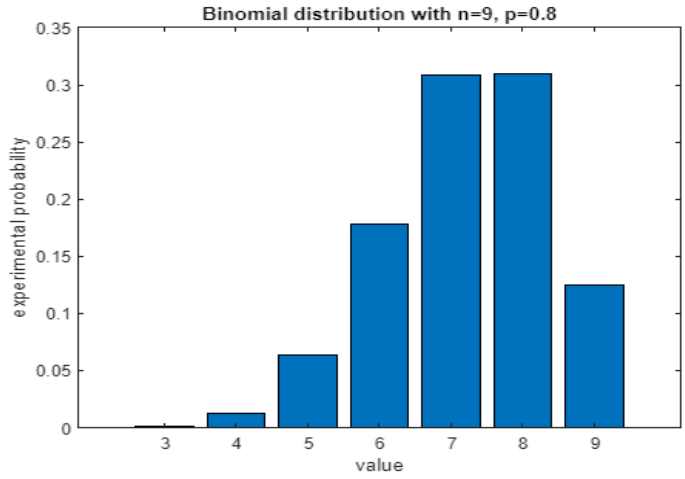
Μέρος Α

A1-A2.

```
N = 1000;
data = binornd(9, 0.8, N,1);
[height,centers] = hist(data,unique(data))
bar(centers,height/sum(height))
xlabel('value');
ylabel('experimental probability')
title('Binomial distribution with n=9, p=0.8')
mean_data=mean(data);
median_data = median(data);
variance_data = var(data);
std_data = std(data);


fprintf('Μέση τιμή: %f\n', mean_data);
fprintf('Ενδιαμέση τιμή: %f\n', median_data);
fprintf('Διασπορά: %f\n', variance_data);
fprintf('Τυπική απόκλιση: %f\n', std_data);
```

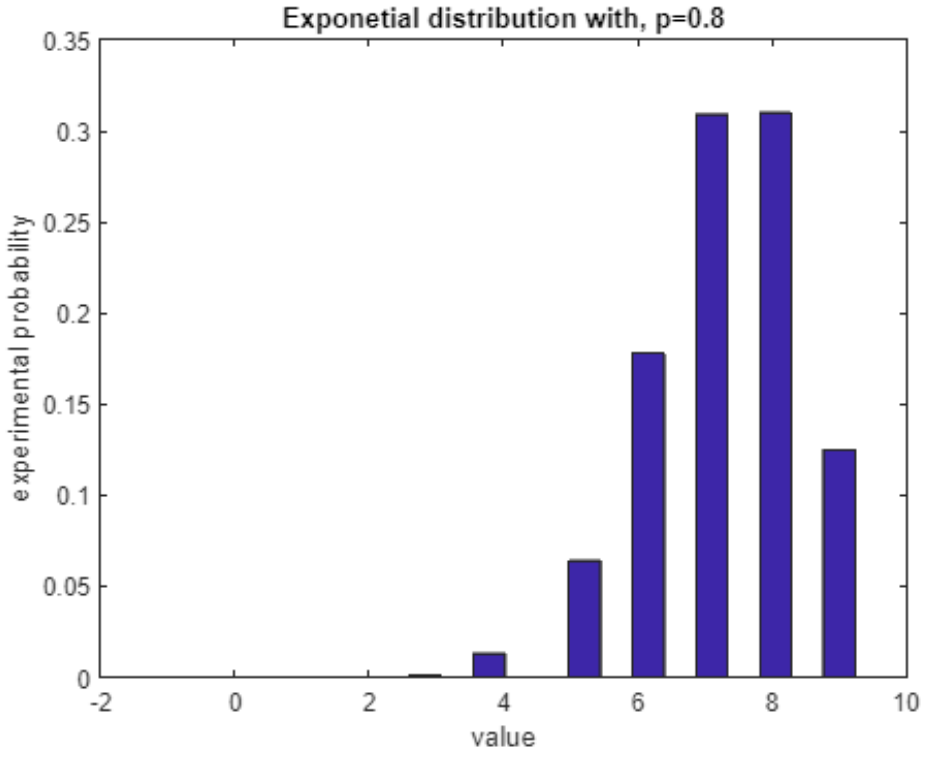
```
height = 1×7
       1    13    64   178   309   310   125

centers = 7×1
       3
       4
       5
       6
       7
       8
       9
```

```
Μέση τιμή: 7.211000
Ενδιαμέση τιμή: 7.000000
Διασπορά: 1.333813
Τυπική απόκλιση: 1.154908
```



Εικόνα 1

```
data1 = exprnd(0.5, N,1);
bin_edges = linspace(0, max(data), 20);
[counts, bin_centers] = hist(data, bin_edges);


bar(bin_centers, counts / sum(counts), 'hist');


xlabel('value');
ylabel('experimental probability')
title('Exponetial distribution with, p=0.8')
mean_data1 =mean(data1);
median_data1 = median(data1);
variance_data1 = var(data1);
std_data1 = std(data1);


fprintf('Μέση τιμή: %f\n', mean_data1);
fprintf('Ενδιαμέση τιμή: %f\n', median_data1);
fprintf('Διασπορά: %f\n', variance_data1);
fprintf('Τυπική απόκλιση: %f\n', std_data1);
```




```
Μέση τιμή: 0.504471
Ενδιαμέση τιμή: 0.351669
Διασπορά: 0.265673
Τυπική απόκλιση: 0.515435
```

Εικόνα 2

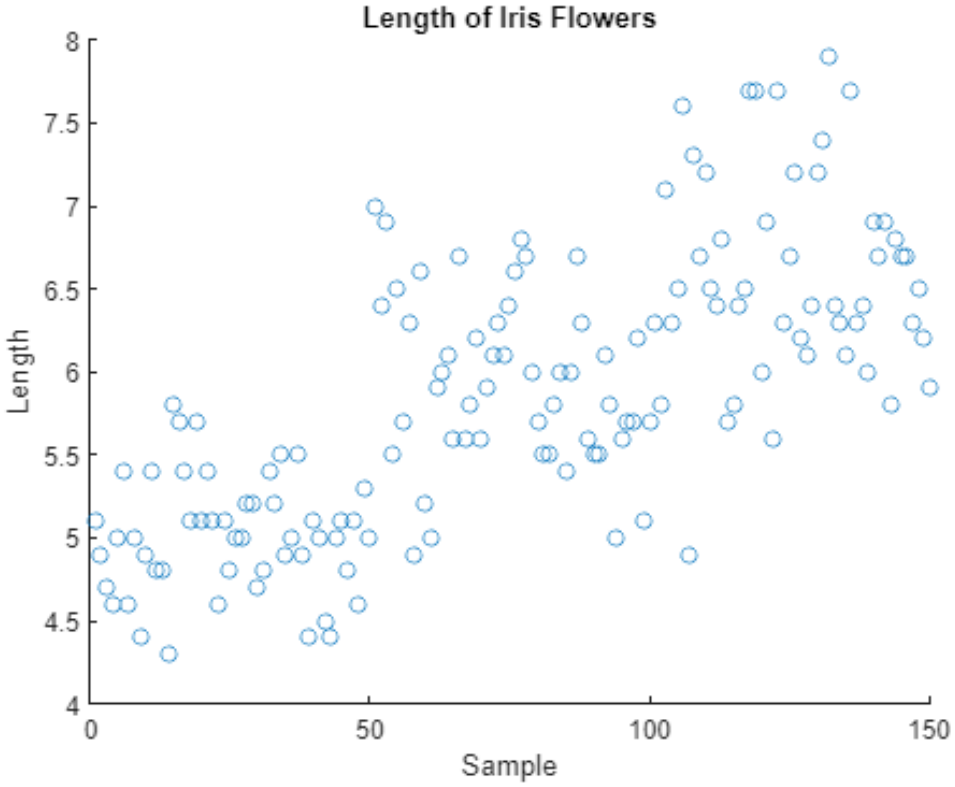
A3-A4.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

                        %BATMOI
load examgrades;
figure;
scatter(1:length(grades), grades);
xlabel('Student');
ylabel('Grade');
title('Exam Grades');
mean_grades = mean(grades(:,1));
median_grades = median(grades(:,1));
variance_grades = var(grades(:,1));
std_grades = std(grades(:,1));
fprintf('Στατιστικά Βαθμών:\n');
fprintf('Μέση τιμή: %.2f\n', mean_grades);
fprintf('Ενδιαμέση τιμή: %.2f\n', median_grades);
fprintf('Διασπορά: %.2f\n', variance_grades);
fprintf('Τυπική απόκλιση: %.2f\n', std_grades);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```


```
Στατιστικά Βαθμών:
Μέση τιμή: 75.01
Ενδιαμέση τιμή: 75.00
Διασπορά: 76.04
Τυπική απόκλιση: 8.72
```

Εικόνα 3

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%FISHER_IRIS
load fisheriris;
figure;
scatter(1:length(meas(:,1)), meas(:,1));
xlabel('Sample');
ylabel('Length');
title('Length of Iris Flowers');
mean_length_iris = mean(meas(:,1));
median_length_iris = median(meas(:,1));
variance_length_iris = var(meas(:,1));
std_length_iris = std(meas(:,1));
fprintf('Στατιστικά Fisher Iris:\n');
fprintf('Μέση τιμή: %.2f\n', mean_length_iris);
fprintf('Ενδιαμέση τιμή: %.2f\n', median_length_iris);
fprintf('Διασπορά: %.2f\n', variance_length_iris);
fprintf('Τυπική απόκλιση: %.2f\n', std_length_iris);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```




```
Στατιστικά Fischer Iris:
Μέση τιμή: 5.84
Ενδιαμέση τιμή: 5.80
Διασπορά: 0.69
Τυπική απόκλιση: 0.83
```

Εικόνα 4

Μέρος Β

Τα **νευρωνικά δίκτυα** είναι μια κατηγορία μοντέλων μηχανικής μάθησης που αποτελούνται από διασυνδεδεμένους κόμβους, που ονομάζονται νευρώνες, οργανωμένους σε επίπεδα. Κάθε νευρώνας λαμβάνει είσοδο, την επεξεργάζεται μέσω μιας συνάρτησης ενεργοποίησης και παράγει μια έξοδο. Μέσω μιας διαδικασίας που ονομάζεται εκπαίδευση, τα νευρωνικά δίκτυα μαθαίνουν να αναγνωρίζουν μοτίβα και σχέσεις στα δεδομένα προσαρμόζοντας τις συνδέσεις μεταξύ των νευρώνων.

Υπάρχουν διάφορα νευρωνικά δικτύα εκ των οποίων είναι οι **perceptron** και **adaline** θα υλοποιηθούν στο Β μέρος της εργασιας.

Β.1 Perceptron Νευρωνικό Δίκτυο

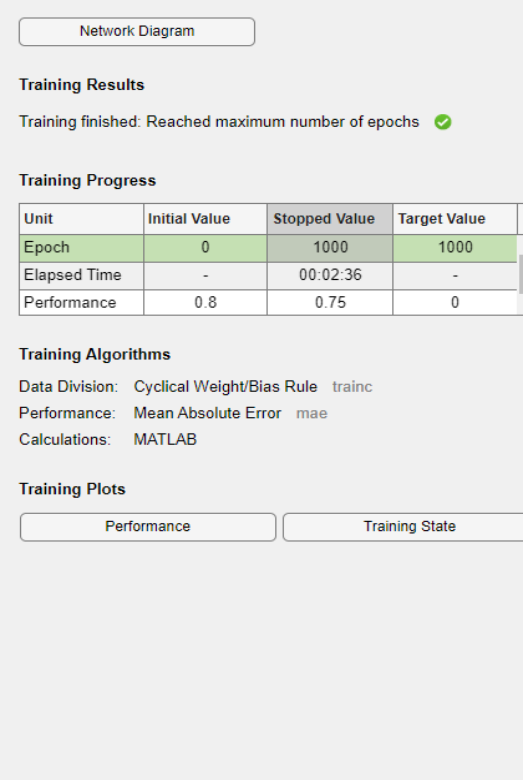
Το **perceptron** είναι ένα βασικό μοντέλο νευρωνικού δικτύου για δυαδική ταξινόμηση. Λαμβάνει χαρακτηριστικά εισόδου, τα πολλαπλασιάζει με βάρη, τα αθροίζει, εφαρμόζει μια συνάρτηση ενεργοποίησης και παράγει μια έξοδο. Είναι μια θεμελιώδης έννοια στη θεωρία των νευρωνικών δικτύων, χρήσιμη για απλές εργασίες ταξινόμησης.

Στη συνεχεια ακολουθεί ένας απλος matlab κώδικας όπου υλοποιείται ο **perceptron**, δεχτεται ένα πίνακα με δεδομένα και στην συνέχεια προσομοιώνει την έξοδο και υπολογίζει το ποσοστό σφάλματος.

Το αρχείο ονομάζεται perceptronNN.m :

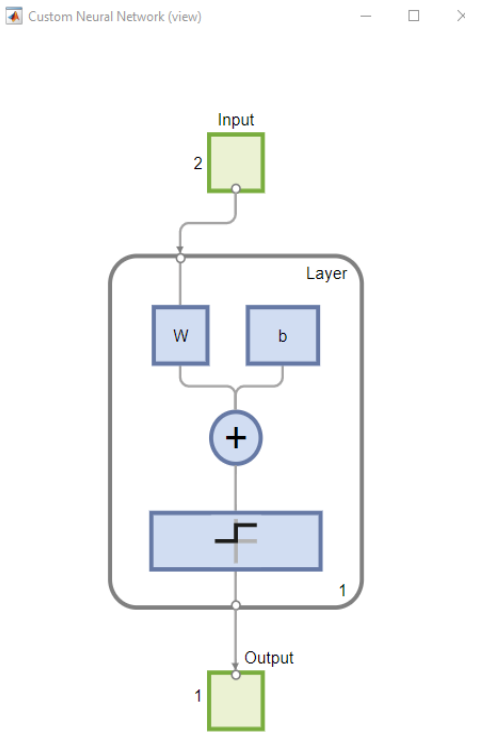
```
inputs = [
   0.67044 -0.437;
   -0.35508 -0.53923;
   0.10452 0.42226;
   0.95826 0.24915;
   0.098617 0.18122;
   -0.33915 0.32088;
   0.23894 -0.90489;
   -0.27873 -0.30243;
   0.51302 -0.09732;
   -0.1722 -0.51819;
   -0.01531 0.43009;
   0.38949 0.71236;
   0.94547 -0.43698;
   -0.34449 0.4621;
   0.67561 -0.72447;
   0.47814 0.67345;
```

```
    0.90835 -0.7228;
    -0.93615 0.17642;
    -0.28626 -0.26769;
    0.32531 0.61352;
];
targets = [1 -1 1 1 1 1 -1 -1 1 -1 1 1 1 1 -1 1 -1 -1 -1 1]';

num_inputs = size(inputs, 2);
perceptron = newp(repmat([-1 1], num_inputs, 1), 1, 'hardlim');

[perceptron, tr] = train(perceptron, inputs', targets');

predicted_targets = sim(perceptron, inputs');
misclassified = find(predicted_targets ~= targets);
cost = length(misclassified);

error_rate = cost / length(targets);

disp(['Cost (Misclassifications): ', num2str(cost)]);
disp(['Error Rate: ', num2str(error_rate * 100), '%']);

figure;
gscatter(inputs(:, 1), inputs(:, 2), targets, 'rb', 'o+');
hold on;


x_domain = linspace(min(inputs(:, 1)), max(inputs(:, 1)), 100);
y_domain = -(perceptron.iw{1}(1) * x_domain + perceptron.b{1}) / perceptron.iw{1}(2);
plot(x_domain, y_domain, 'k', 'LineWidth', 2);
xlabel('Feature 1');
ylabel('Feature 2');
title('Perceptron Decision Boundary');
legend('Class 1', 'Class 2', 'Decision Boundary');
grid on;
hold off;
```



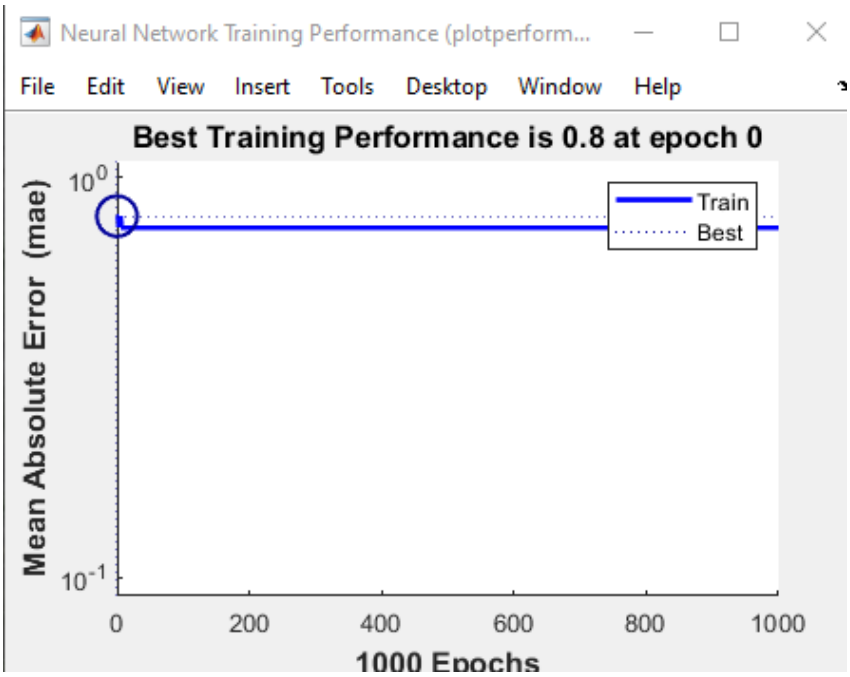
Εικόνα 5

Με την εκτέλεση του παραπάνω κώδικα, ξεκινάει η εκπαίδευση του νευρωνικού δικτύου και στην *Εικόνα 5* μπορούμε να διακρίνουμε ένα παράθυρο με περισσότερες λεπτομέρειες καθ' όλη την διάρκεια της εκπαίδευσης.

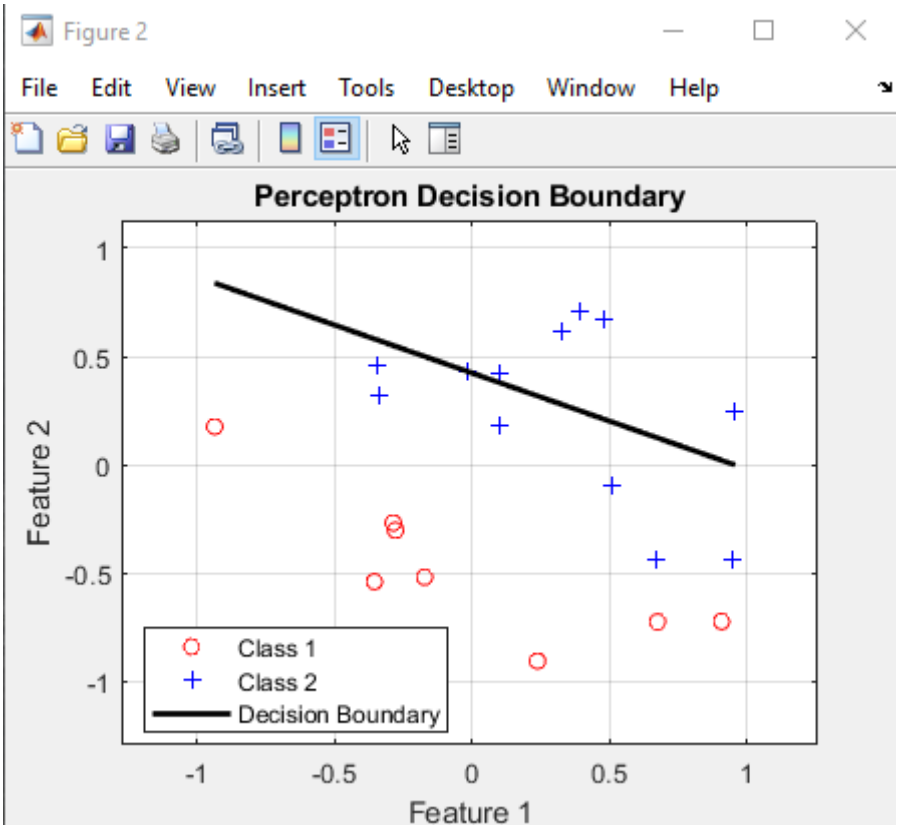


Εικόνα 6

Στην *Εικόνα 6* βλέπουμε το διάγραμμα του δικτύου Perceptron.



Εικόνα 7



Εικόνα 8

```
>> perceptronNN
Cost (Misclassifications): 340
Error Rate: 1700%
>>
```

Εικόνα 9

Με το τέλος της εκπαίδευσης του νευρωνικού δικτύου εμφανίζονται οι υπολογισμοί του κόστους και του ποσοστού σφάλματος και τέλος το plot για την απόφαση του perceptron για τα όρια, τα οποία μπορούμε να δούμε πως δεν είναι τα βέλτιστα.

B.2 Adaline Νευρωνικό Δίκτυο

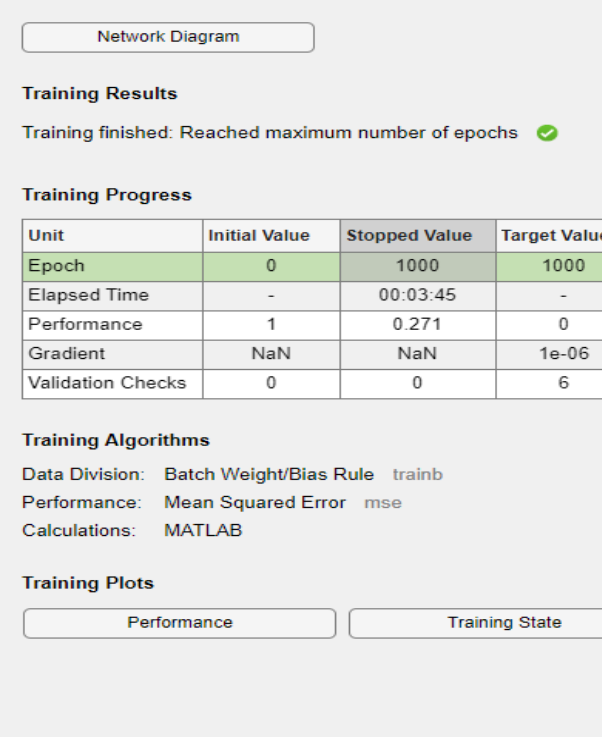
Το **Adaline** είναι ένα νευρωνικό δίκτυο για δυαδική ταξινόμηση. Προσαρμόζει τα βάρη του με βάση τη συνεχή έξοδο χρησιμοποιώντας μια γραμμική συνάρτηση ενεργοποίησης. Χρησιμοποιείται όταν απαιτείται γραμμικός διαχωρισμός των κλάσεων.

Στη συνέχεια ακολουθεί ένας απλός matlab κώδικας όπου υλοποιείται ο **adaline**, δέχεται ένα πίνακα με δεδομένα και στην συνέχεια προσομοιώνει την έξοδο και υπολογίζει το ποσοστό σφάλματος.

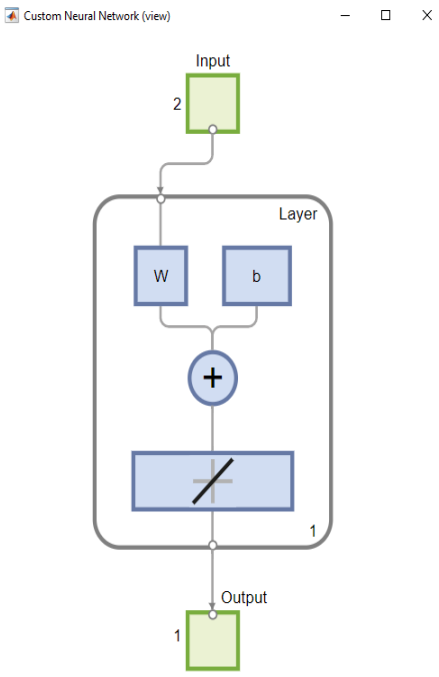
Το αρχείο ονομάζεται adalineNN.m :

```
inputs = [
    0.67044 -0.437;
    -0.35508 -0.53923;
    0.10452 0.42226;
    0.95826 0.24915;
    0.098617 0.18122;
    -0.33915 0.32088;
    0.23894 -0.90489;
    -0.27873 -0.30243;
```

```
    0.51302 -0.09732;
    -0.1722 -0.51819;
    -0.01531 0.43009;
    0.38949 0.71236;
    0.94547 -0.43698;
    -0.34449 0.4621;
    0.67561 -0.72447;
    0.47814 0.67345;
    0.90835 -0.7228;
    -0.93615 0.17642;
    -0.28626 -0.26769;
    0.32531 0.61352;
];
targets = [1 -1 1 1 1 1 -1 -1 1 -1 1 1 1 1 -1 1 -1 -1 -1 1]';

net = newlin(inputs', 1);

[net, tr] = train(net, inputs', targets');

predicted_targets = sim(net, inputs');
mse = mean((predicted_targets - targets).^2);

error_rate = sum(predicted_targets ~= targets) / length(targets);


disp(['Error Rate: ', num2str(error_rate * 100), '%']);
disp(['Mean Squared Error: ', num2str(mse)]);
figure;
gscatter(inputs(:, 1), inputs(:, 2), targets, 'rb', 'o+');
hold on;


x_domain = linspace(min(inputs(:, 1)), max(inputs(:, 1)), 100);
y_domain = -(net.IW{1}(1) * x_domain + net.b{1}) / net.IW{1}(2);
plot(x_domain, y_domain, 'k', 'LineWidth', 2);
xlabel('Feature 1');
ylabel('Feature 2');
title('Adaline Decision Boundary');
legend('Class 1', 'Class 2', 'Decision Boundary');
grid on;
hold off;
```



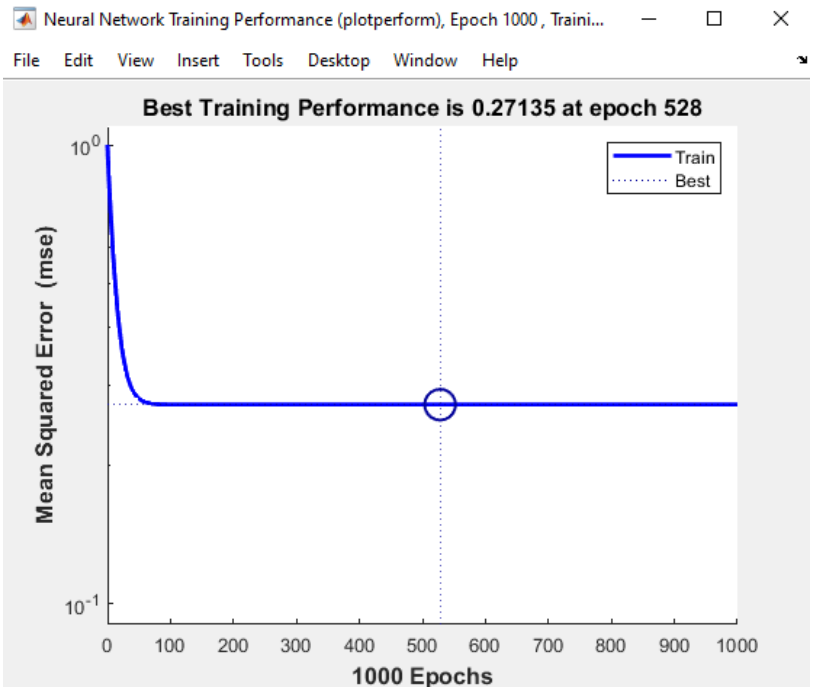
Εικόνα 10



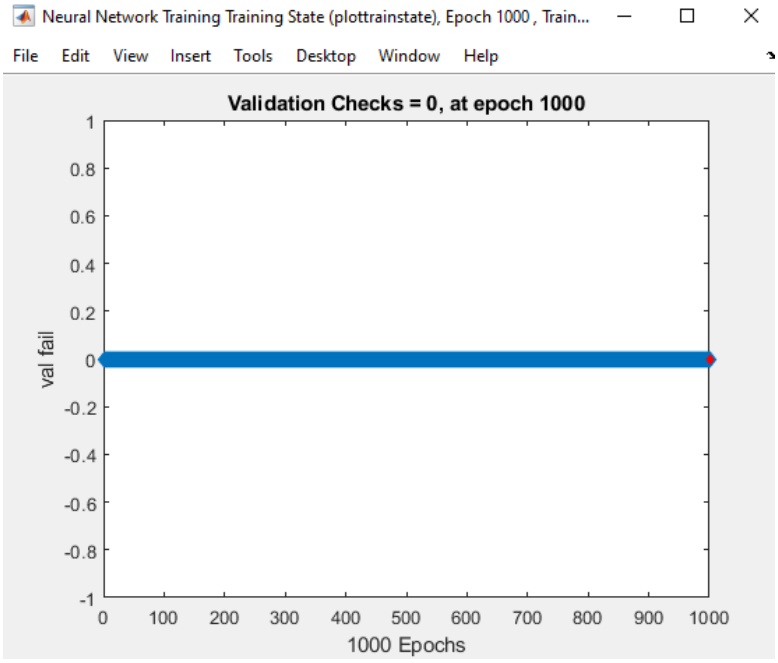
Εικόνα 11

Με την εκτέλεση του παραπάνω κώδικα, ξεκινάει η εκπαίδευση του νευρωνικού δικτύου και στην *Εικόνα 10* μπορούμε να διακρίνουμε ένα παράθυρο με περισσότερες λεπτομέρειες καθ' όλη την διάρκεια της εκπαίδευσης.

Στην *Εικόνα 11* βλέπουμε το διάγραμμα του δικτύου Adaline.

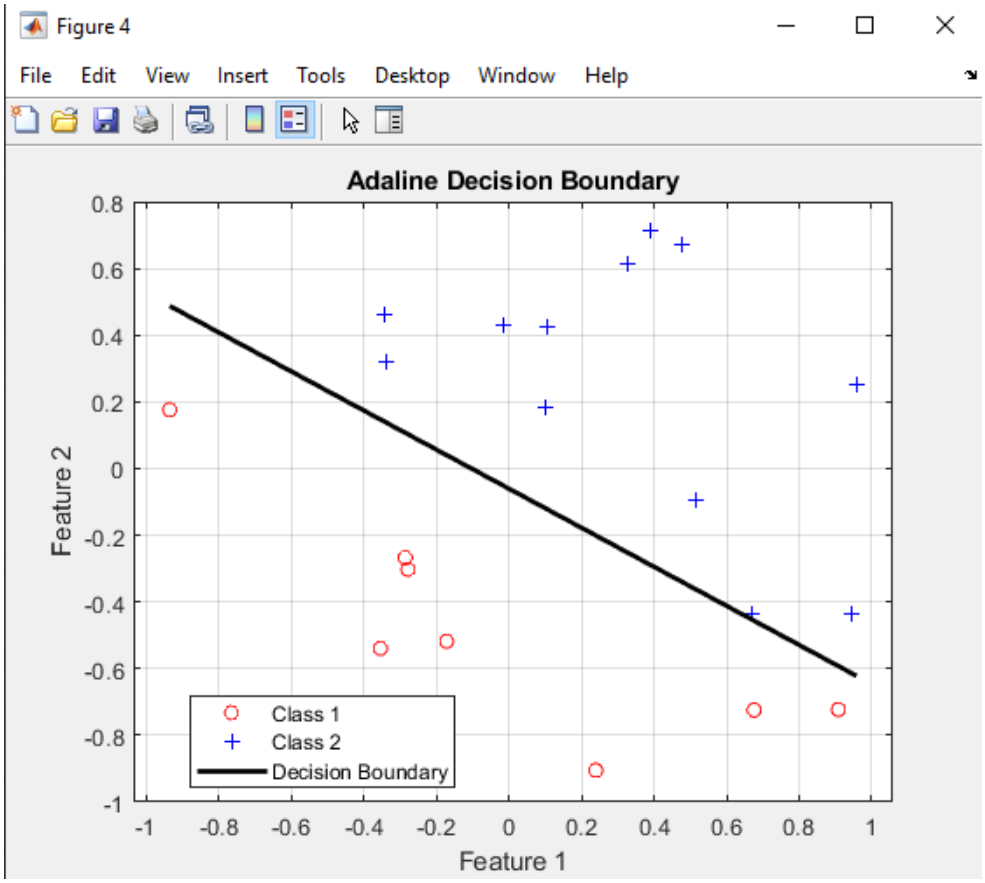


Εικόνα 12



Εικόνα 13

Κατά την διάρκεια της εκπαίδευσης μπορούμε να μελετήσουμε την ζωντανή κατάσταση και απόδοση του Adaline με διαγράμματα σχήματος (Εικόνες 12 &13).



Εικόνα 14

Error Rate: 100%

Mean Squared Error:

0.9906 2.5364 1.3657 2.2583 1.0281 0.96665 2.6061 1.6367 1.0023
2.0805 1.2528 2.7547 0.96529 1.0464 1.335 2.8086 1.1202 1.4199
1.5622 2.2366

Με το τέλος της εκπαίδευσης του νευρωνικού δικτύου εμφανίζονται οι υπολογισμοί του κόστους και του ποσοστού σφάλματος και τέλος το plot για την απόφαση του adaline για τα όρια, τα οποία μπορούμε να δούμε πως είναι ορθά καθώς γίνεται γραμμικός διαχωρισμός μια από τις ειδικότητες του adaline .