# Neural Networks Practice 2- Two Neurons Perceptron

## Elizarraras Llanos Angel Gustavo

Designs and implement an artificial neural network able to separate four different classes.

Proposed classes:

## Generating the variables for the neuron

P11 = [-3,1];  
P12 = [-1 2];  
P21 = [-2 2];  
P22 = [4 -3];  
P31 = [-2 -4];  
P32 = [-1 -3];  
P41 = [3 0];  
P42 = [4 2];  
Targets = [1; 1; 0; 0; 1; 0;0; 1];%Targets  
p = [P11; P12; P21; P22; P31; P32; P41; P42];%Points  
P = p';  
Number\_classes = 4;  
N=2;%Number of neurons  
W = rand(2,N);%random weigth  
b = rand(N,1);%random bias  
Original\_random\_bias = b  
W\_Random = W  
Epochs = 10;%I did not ask to the user because if i do, i can't publish with matlab.  
e=zeros(2,1);  
x=zeros(2,1);

Original\_random\_bias =  
  
 0.9595  
 0.6557  
  
  
W\_Random =  
  
 0.1419 0.9157  
 0.4218 0.7922

## Training

for i = 1:Epochs  
 for j = 1:Number\_classes  
  
 a(1) = dot(W(1,:),P(:,(j+4))) + b(1);  
 a(2) = dot(W(2,:),P(:,j)) + b(2);  
 A = hardlim(a(1) + a(2));%Getting only one A  
 e(1) = Targets(j) - A;  
 e(2) = Targets(j+4) - A;  
 x1 = e(1)\*P(:,j);  
 x2 = e(2)\*P(:,j+4);  
 W(1,:) = W(1,:) + x1';  
 W(2,:) = W(2,:) + x2';  
 b(1) = b(1) + e(1);  
 b(2) = b(2) + e(2);  
 end  
 end

## Checking training

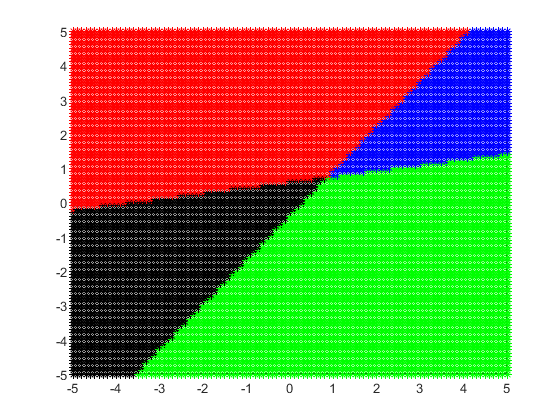
b\_after\_training = b  
W\_after\_training = W  
X = -5:0.1:5;  
figure(1)  
  
 for i = 1:length(X)  
 for j = 1:length(X)  
  
 allP = [X(i);X(j)];  
  
 C = hardlim(W\*allP + b);  
  
 if( C(1) == 0 && C(2) == 0 )  
 plot(X(i) , X(j), '\*b')  
  
 elseif ( C(1) == 0 && C(2) == 1 )  
 plot(X(i) , X(j), '\*g')  
  
 elseif ( C(1) == 1 && C(2) == 1 )  
 plot(X(i) , X(j), '\*k')  
  
 elseif ( C(1) == 1 && C(2) == 0 )  
 plot(X(i) , X(j), '\*r')  
 end  
 hold on  
 end  
 end

b\_after\_training =

16.9595  
 16.6557

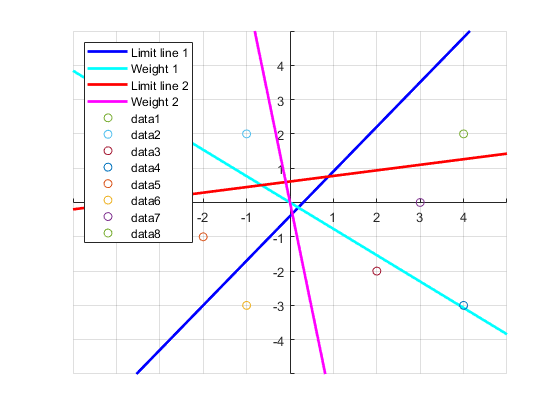
W\_after\_training =  
  
 -55.8581 42.915

4.4218 -27.2



## Getting the limit line

xpoint1 = -b(1) / W(1);  
ypoint1 = -b(1) / W(3);  
slope1 = -ypoint1/xpoint1;  
Y1 = slope1\*X + ypoint1;  
  
mw1 = -1 / slope1;  
Yw1 = mw1\*X;  
  
xpoint2 = -b(2) / W(2);  
ypoint2 = -b(2) / W(4);  
slope2 = -ypoint2/xpoint2;  
Y2 = slope2\*X + ypoint2;  
  
mw2 = -1 / slope2;  
Yw2 = mw2\*X;  
  
%Plotting the limit line and the weight  
figure(2)  
%Axis  
ax = gca;  
ax.XAxisLocation = 'origin';  
ax.YAxisLocation = 'origin';  
hold on  
grid on  
axis([-5 5 -5 5]);  
plot(X,Y1,'b',X,Yw1, 'c','Linewidth',2)  
hold on  
plot(X,Y2,'r',X,Yw2,'m', 'Linewidth',2)  
legend({'Limit line 1','Weight 1','Limit line 2','Weight 2'},'Location','northwest')  
hold on  
scatter(-3,1)  
hold on  
scatter(-1,2)  
hold on  
scatter(2, -2)  
hold on  
scatter(4,-3)  
hold on  
scatter(-2,-1)  
hold on  
scatter(-1, -3)  
hold on  
scatter(3,0)  
hold on  
scatter(4,2)  
hold on



[*Published with MATLAB® R2019b*](https://www.mathworks.com/products/matlab)