```
% homework 1 ECE 559 - Neural Networks - Fall 2018
% Montagna Marco

% exercise no. 3

clc
clear all
% a
% b
%w0 = rand/2 -1/4;
% c
%w1 = -1 + 2*rand;
% d
%w2 = -1 + 2*rand;
w0 = -0.2478
w1 = 0.0852
w2 = 0.7227
% e
% f
n = 1000; % number of inputs
S = -1 + 2*rand(n, 2);
% g, h
W = [w0, w1, w2] % weights vector
k = 1; % these are only flag used in the for loop
y = 1;
for i = 1:1:n
    if [1, S(i,:)]*W' >= 0 % creating matrix S1
        S1(k,:) = S(i,:);
        k = k+1;
    elseif [1, S(i,:)]*W' < 0 % creating matrix S0
        S0(y,:) = S(i,:);
        y = y+1;
    end
end
% i
% plot the graph
X1 = linspace(-1, 1);
X2 = -(w0 + w1*X1)/w2;
plot(X1, X2) % plotting boundary line
hold on
plot (S1(:, 1), S1(:, 2), 'ro') % plotting S1 values
hold on
plot (S0(:, 1), S0(:, 2), 'ks') % plotting S0 values
axis([-1 1 -1 1])
title('Figure for Problem 3i')
xlabel('X1')
ylabel('X2')
legend('Boundary','S1','S0')
% j
% i.
eta1 = 1;
% ii.
```

```
% w10 = -1 + 2*rand; % inizialize weights randomly
% w11 = -1 + 2*rand;
% w12 = -1 + 2*rand;
w10 = -0.5374
w11 = -0.8956
w12 = 0.8035
W1 = [w10, w11, w12]
W2 = W1;
% iii., iv., v., vi., k
misc = 1; % misclassifications
epochn1 = 0; % epoch number
while misc ~= 0 % while loop until convergence
    epochn1 = epochn1 + 1;
    misc = 0; % reinizialize misclassifications number
    for i = 1:1:n % for loop for selecting every input
        z = [1, S(i,:)]*W2';
        d = [1, S(i,:)]*W';
        output = heaviside(z); % compute the actual output
        desiredout = heaviside(d); % desired output
        if desiredout ~= output % if the desired output is different
 from the actual output
            misc = misc + 1;
            if desiredout == 1
                W2 = W2 + eta1*[1, S(i,:)]; % update weights
            elseif desiredout == 1/2 % this is becuase in heavyside
 function of matlab if x = 0, the output = 1/2
                                % In this way we can get aroundd the
 problem
                W2 = W2 + eta1*[1, S(i,:)]; % update weights
            elseif desiredout == 0
                W2 = W2 - eta1*[1, S(i,:)]; % update weights
            end
        end
    end
    figure (2)
    bar(epochn1, misc)
    hold on
    title('Epoch/Misc, eta 1')
    xlabel('EpochNumber')
    ylabel('Number of Misclassifications')
end
% vii
disp('epochn1:')
disp(epochn1)
disp('weights1:')
disp(W2)
% this was made only for checking that the boundary line computed with
 PTA
% and eta = 1 divide S1 and S0 correctly
X1 = linspace(-1, 1);
X2 = -(W2(1,1) + W2(1,2)*X1)/W2(1,3);
figure (3)
plot(X1, X2) % plotting boundary line
hold on
```

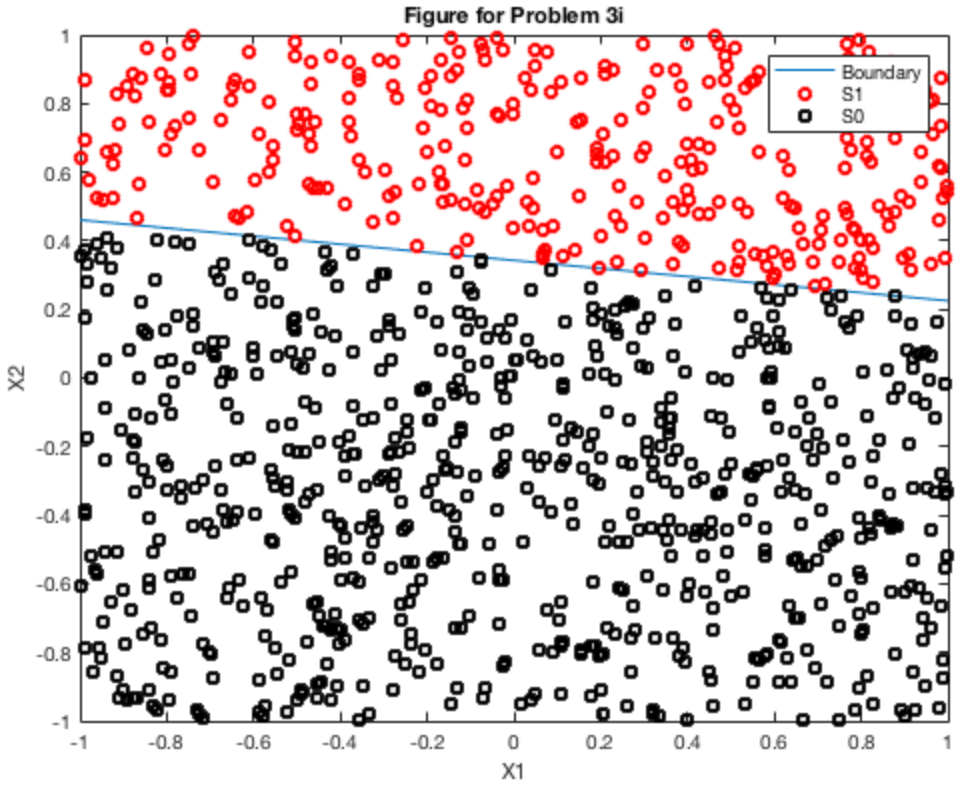
```
plot (S1(:, 1), S1(:, 2), 'ro') % plotting S1 values
hold on
plot (S0(:, 1), S0(:, 2), 'ks') % plotting S0 values
axis([-1 1 -1 1])
title('Figure for Eta 1')
xlabel('X1')
ylabel('X2')
legend('Boundary','S1','S0')

% l
eta10 = 10;
W3 = W1;
misc = 1; % misclassifications
epochn10 = 0; % epoch number
while misc ~= 0 % while loop until convergence
    epochn10 = epochn10 + 1;
    misc = 0; % reinizialize misclassifications number
    for i = 1:1:n % for loop for selecting every input
        z = [1, S(i,:)]*W3';
        d = [1, S(i,:)]*W';
        output = heaviside(z); % compute the actual output
        desiredout = heaviside(d); % desired output
        if desiredout ~= output % if the desired output is different
 from the actual output
            misc = misc + 1;
            if desiredout == 1
                W3 = W3 + eta10*[1, S(i,:)]; % update weights
            elseif desiredout == 1/2 % this is becuase in heavyside
 function of matlab if x = 0, the output = 1/2
                                % In this way we can get aroundd the
 problem
                W3 = W3 + eta10*[1, S(i,:)]; % update weights
            elseif desiredout == 0
                W3 = W3 - eta10*[1, S(i,:)]; % update weights
            end
        end
    end
    figure (4)
    bar(epochn10, misc)
    hold on
    title('Epoch/Misc, eta 10')
    xlabel('EpochNumber')
    ylabel('Number of Misclassifications')
end
disp('epochn10:')
disp(epochn10)
disp('weights10:')
disp(W3)

% this was made only for checking that the boundary line computed with
 PTA
% and eta = 10 divide S1 and S0 correctly
X1 = linspace(-1, 1);
X2 = -(W3(1,1) + W3(1,2)*X1)/W3(1,3);
```

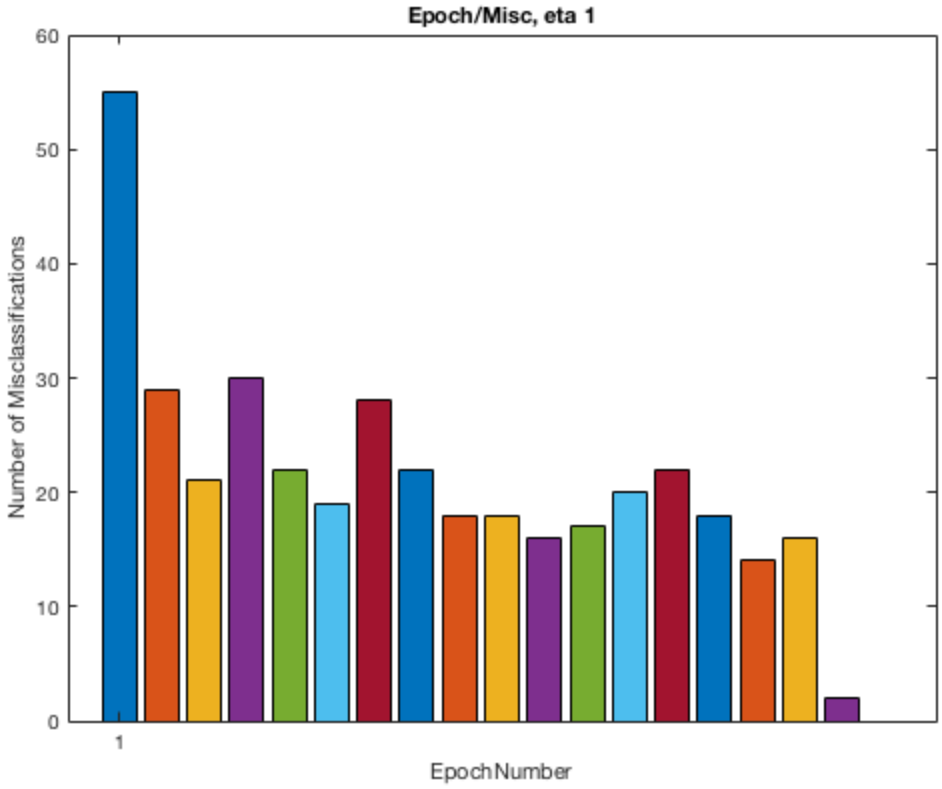
```
figure (5)
plot(X1, X2) % plotting boundary line
hold on
plot (S1(:, 1), S1(:, 2), 'ro') % plotting S1 values
hold on
plot (S0(:, 1), S0(:, 2), 'ks') % plotting S0 values
axis([-1 1 -1 1])
title('Figure for Eta 10')
xlabel('X1')
ylabel('X2')
legend('Boundary','S1','S0')

% m
eta01 = 0.1;
W4 = W1;
misc = 1; % misclassifications
epochn01 = 0; % epoch number
while misc ~= 0 % while loop until convergence
    epochn01 = epochn01 + 1;
    misc = 0; % reinizialize misclassifications number
    for i = 1:1:n % for loop for selecting every input
        z = [1, S(i,:)]*W4';
        d = [1, S(i,:)]*W';
        output = heaviside(z); % compute the actual output
        desiredout = heaviside(d); % desired output
        if desiredout ~= output % if the desired output is different
 from the actual output
            misc = misc + 1;
            if desiredout == 1
                W4 = W4 + eta01*[1, S(i,:)]; % update weights
            elseif desiredout == 1/2 % this is becuase in heavyside
 function of matlab if x = 0, the output = 1/2
                              % In this way we can get aroundd the
 problem
                W4 = W4 + eta01*[1, S(i,:)]; % update weights
            elseif desiredout == 0
                W4 = W4 - eta01*[1, S(i,:)]; % update weights
            end
        end
    end
    figure (6)
    bar(epochn01, misc)
    hold on
    title('Epoch/Misc, eta 01')
    xlabel('EpochNumber')
    ylabel('Number of Misclassifications')
end
disp('epochn01:')
disp(epochn01)
disp('weights01:')
disp(W4)

% this was made only for checking that the boundary line computed with
 PTA
```

```
% and eta = 10 divide S1 and S0 correctly
X1 = linspace(-1, 1);
X2 = -(W4(1,1) + W4(1,2)*X1)/W4(1,3);
figure (7)
plot(X1, X2) % plotting boundary line
hold on
plot (S1(:, 1), S1(:, 2), 'ro') % plotting S1 values
hold on
plot (S0(:, 1), S0(:, 2), 'ks') % plotting S0 values
axis([-1 1 -1 1])
title('Figure for Eta 0.1')
xlabel('X1')
ylabel('X2')
legend('Boundary','S1','S0')
```

```
w0 =

   -0.2478


w1 =

    0.0852


w2 =

    0.7227


W =

   -0.2478    0.0852    0.7227


w10 =

   -0.5374


w11 =

   -0.8956


w12 =

    0.8035


W1 =

   -0.5374   -0.8956    0.8035
```

```
epochn1:
    19

weights1:
   -5.5374    1.8813   16.0423

epochn10:
    33

weights10:
  -60.5374   18.7612  177.5352

epochn01:
    26

weights01:
   -0.6374    0.2044    1.8791
```




Epoch/Misc, eta 1
Number of Misclassifications
EpochNumber
0
10
20
30
40
50
60
1

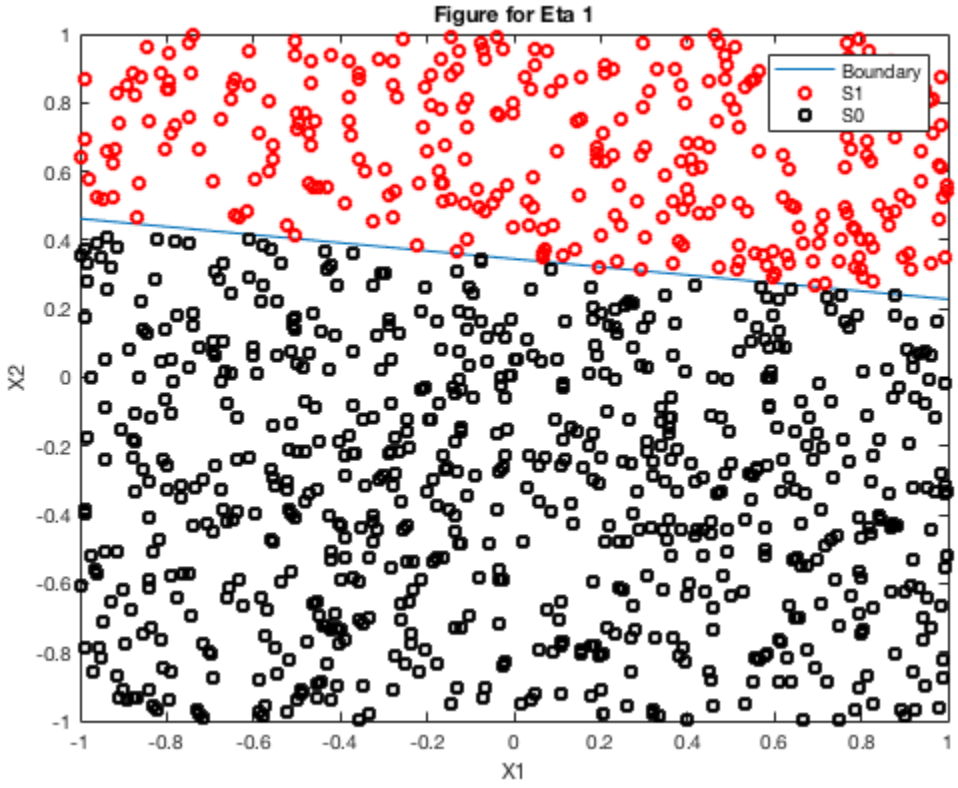
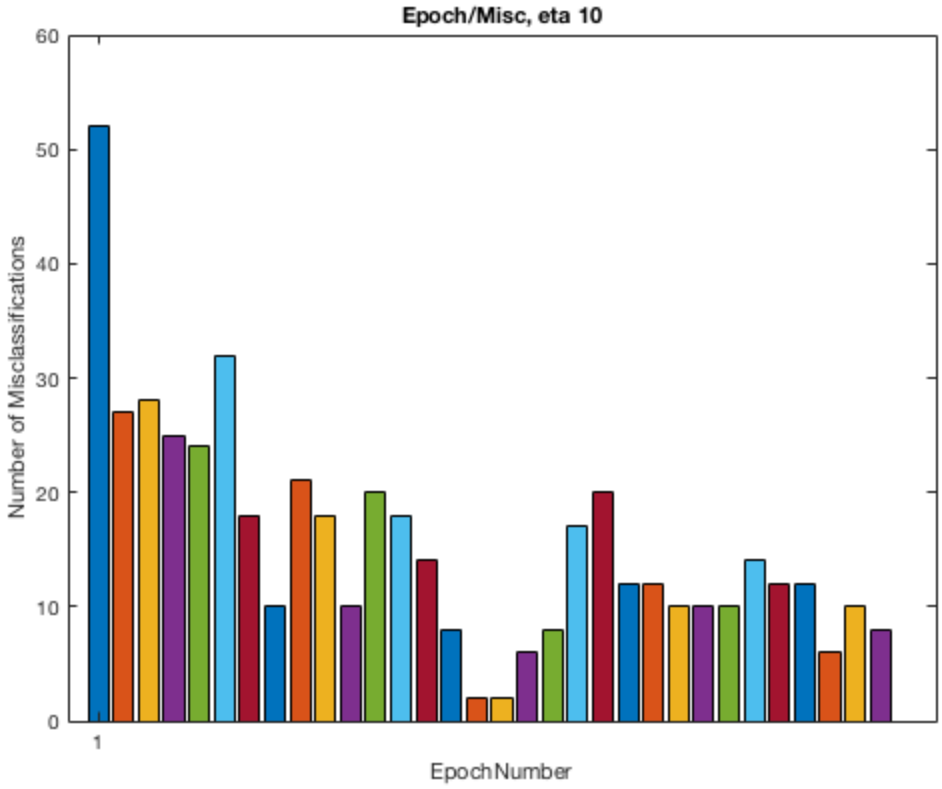


Figure for Eta 1
Boundary
S1
S0
X1
X2
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1


Epoch/Misc, eta 10
Number of Misclassifications
EpochNumber
0
10
20
30
40
50
60
1

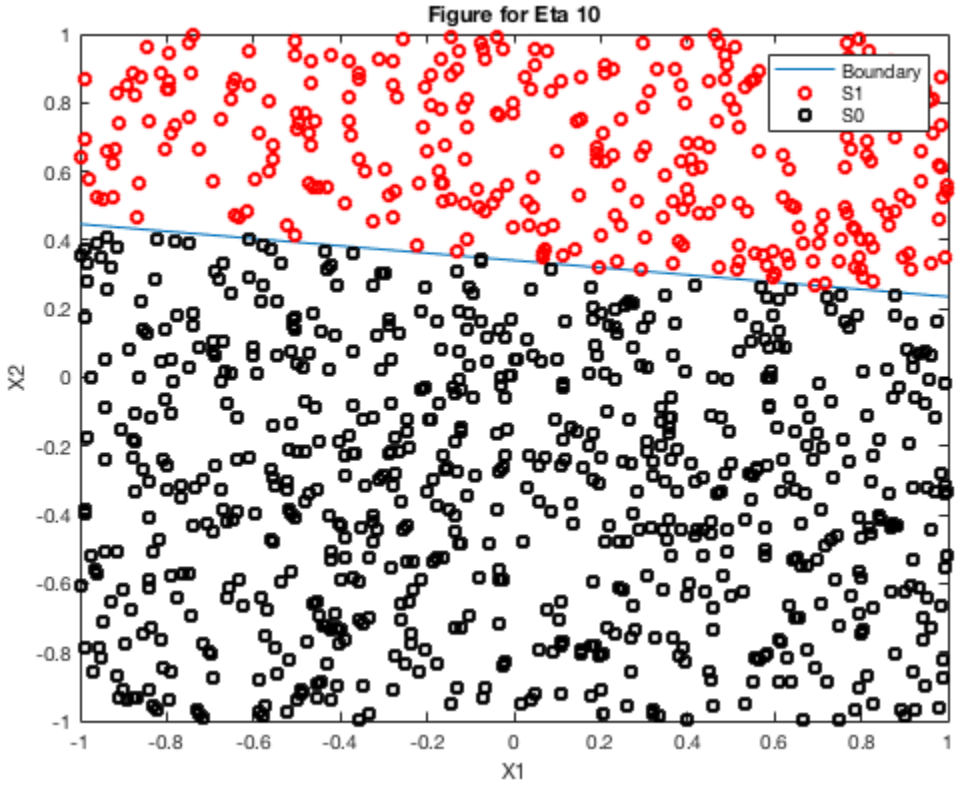
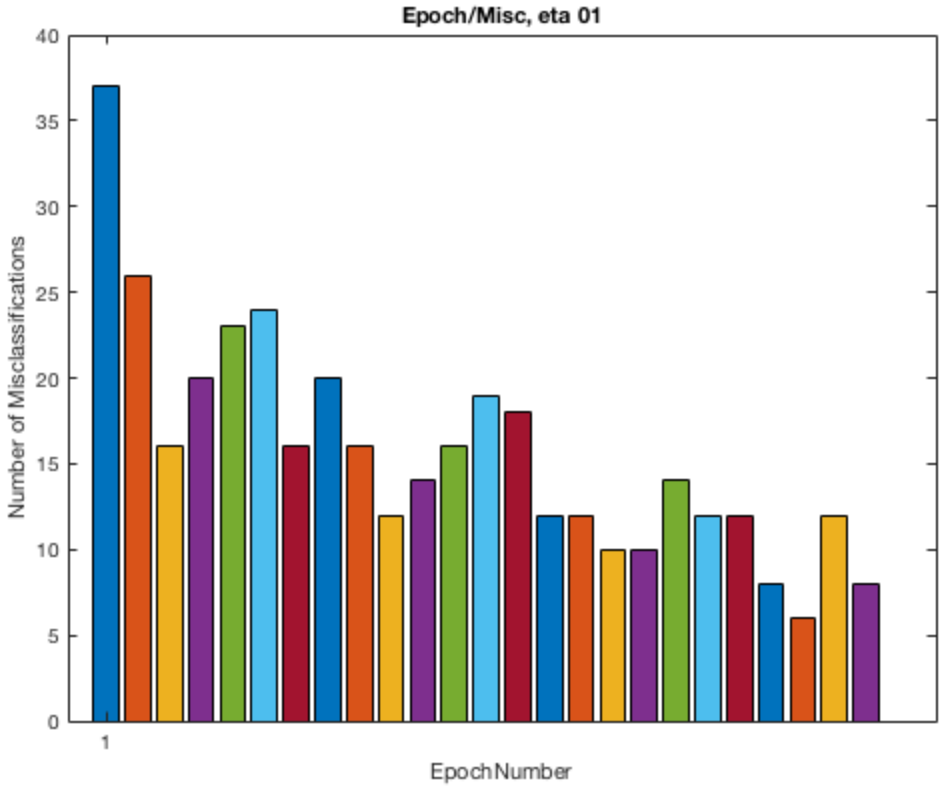


Figure for Eta 10
Boundary
S1
S0
X1
X2
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1


Epoch/Misc, eta 01
Number of Misclassifications
EpochNumber
0
5
10
15
20
25
30
35
40
1

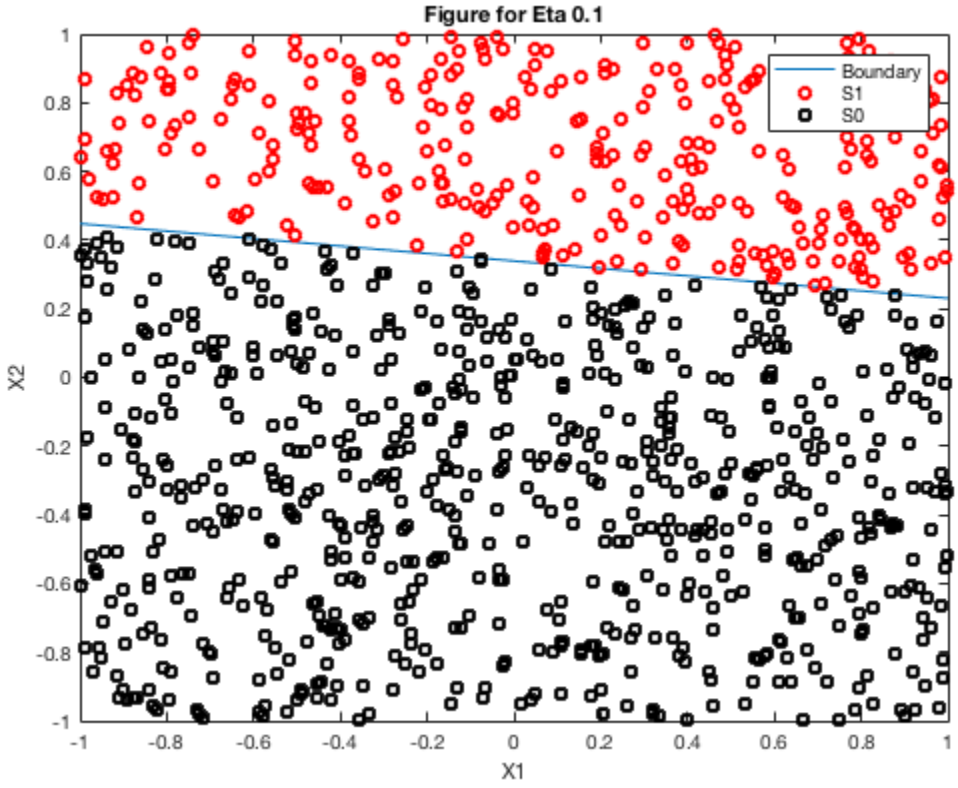


Figure for Eta 0.1
Boundary
S1
S0
X1
X2
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

Published with MATLAB® R2017b