
```
% 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;
% e
% f
n = 100; % 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; % initialize weights randomly
w11 = -1 + 2*rand;
w12 = -1 + 2*rand;
```

```

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; % reinitalize 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 because in heavyside
                    function of matlab if x = 0, the output = 1/2
                        % In this way we can get around the
                        problem
                            W2 = W2 + eta1*[1, S(i,:)]; % update weights
                elseif desiredout == 0
                    W2 = W2 - eta1*[1, S(i,:)]; % update weights
                end
            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')

% 1
eta10 = 10;
W3 = W1;
misc = 1; % misclassifications
epochn10 = 0; % epoch number
while misc ~= 0 % while loop until convergence
    epochn10 = epochn10 + 1;
    misc = 0; % reinitalize 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 because in heavyside
                    function of matlab if x = 0, the output = 1/2
                        % In this way we can get around the
                        problem
                            W3 = W3 + eta10*[1, S(i,:)]; % update weights
                elseif desiredout == 0
                    W3 = W3 - eta10*[1, S(i,:)]; % update weights
                end
            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; % reinitalize 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 because in heavyside
                    function of matlab if x = 0, the output = 1/2
                        % In this way we can get around the
                        problem
                            W4 = W4 + eta01*[1, S(i,:)]; % update weights
                elseif desiredout == 0
                    W4 = W4 - eta01*[1, S(i,:)]; % update weights
                end
            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')

```

```
W =
```

```

-0.2478    0.0852    0.7227

```

```
W1 =
```

```

-0.5374   -0.8956    0.8035

```

```
epochn1:
```

```

71

```

```
weights1:
```

```

-4.5374    1.4175   13.5878

```

```
epochn10:
```

```

99

```

```
weights10:
```

```

-50.5374   15.5613  152.5786

```

```
epochn01:
```

```

31

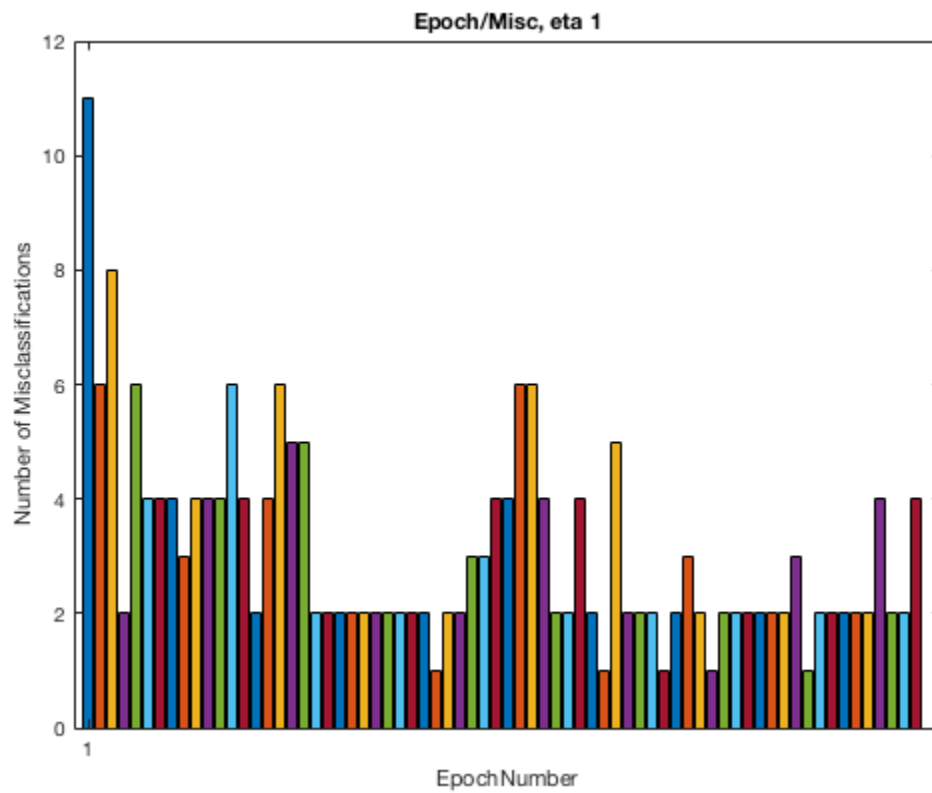
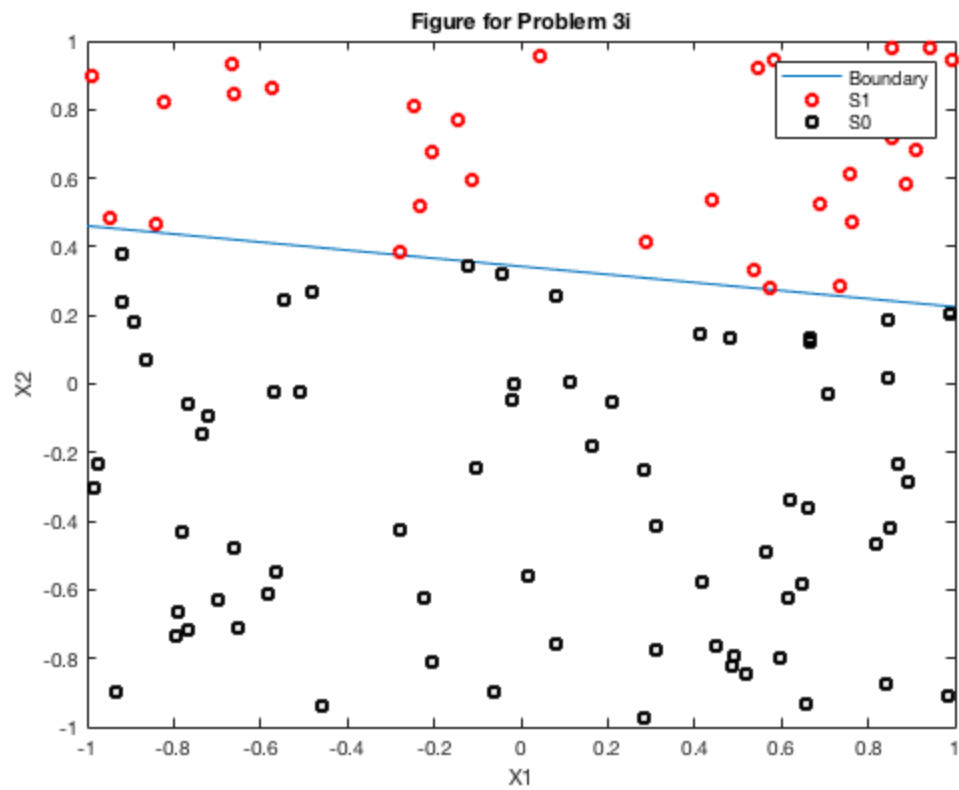
```

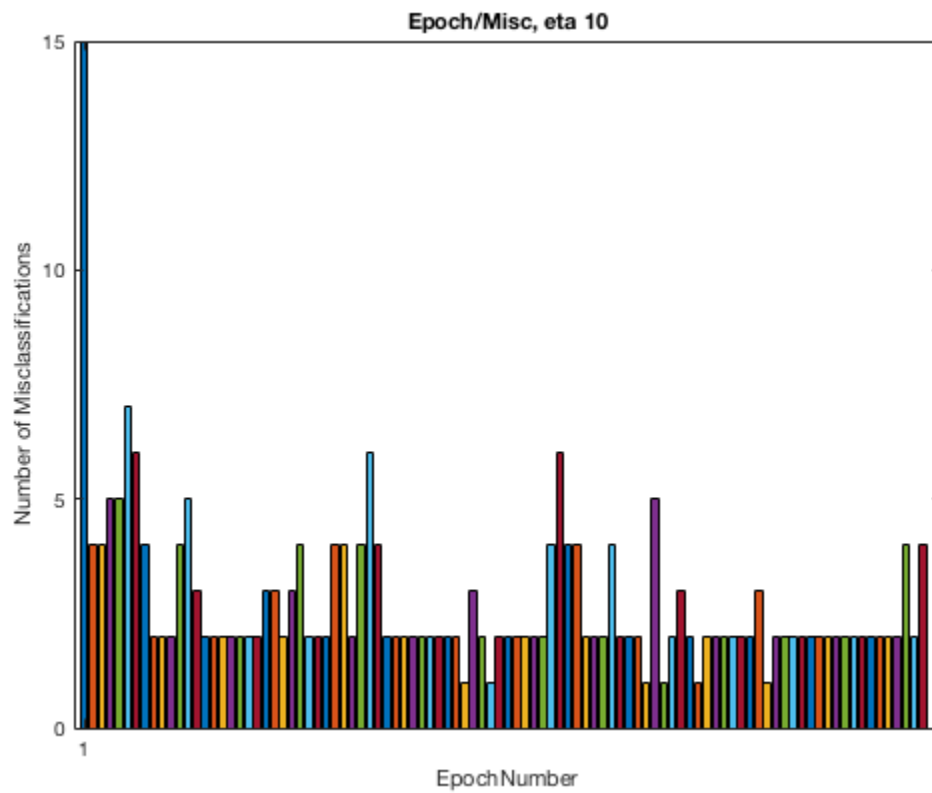
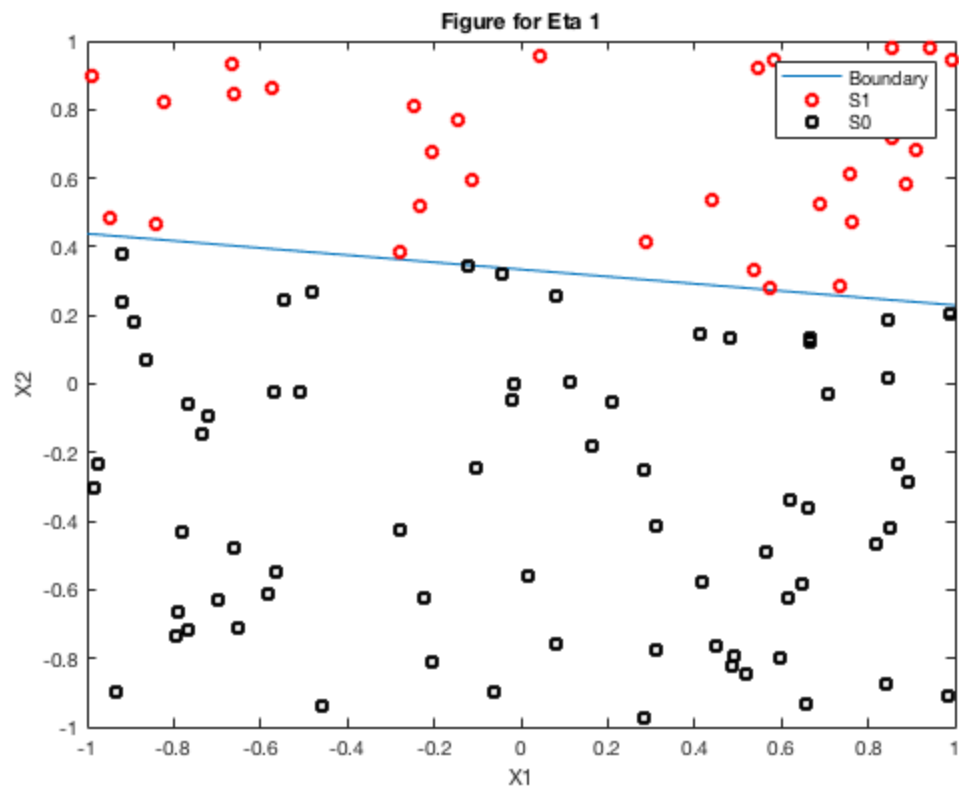
```
weights01:
```

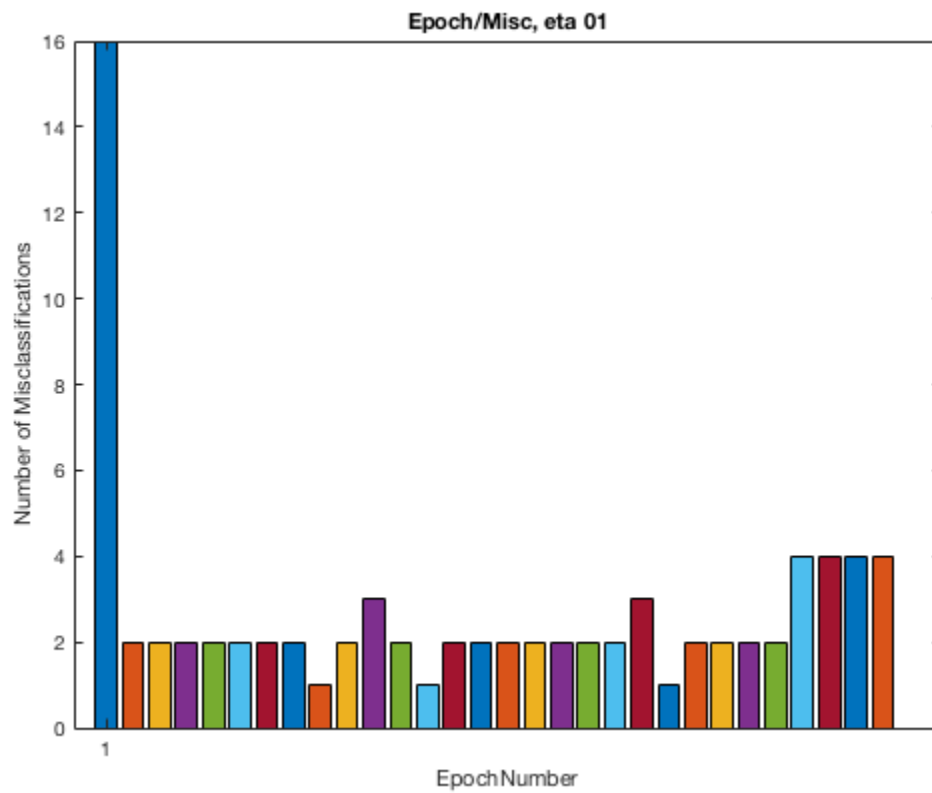
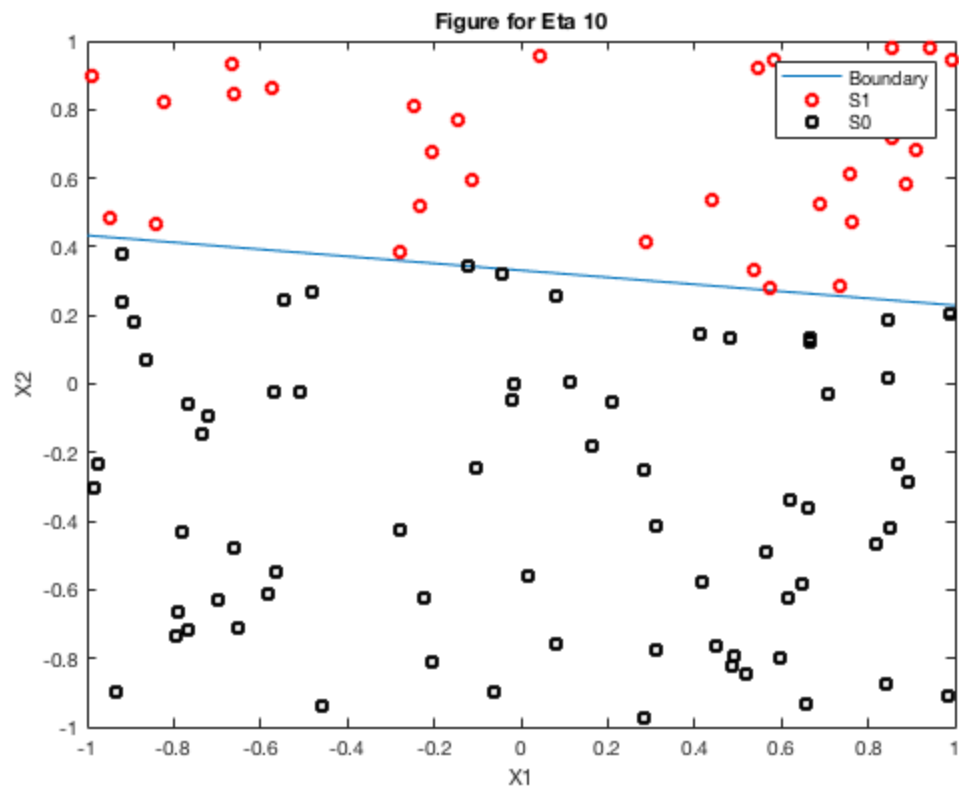
```

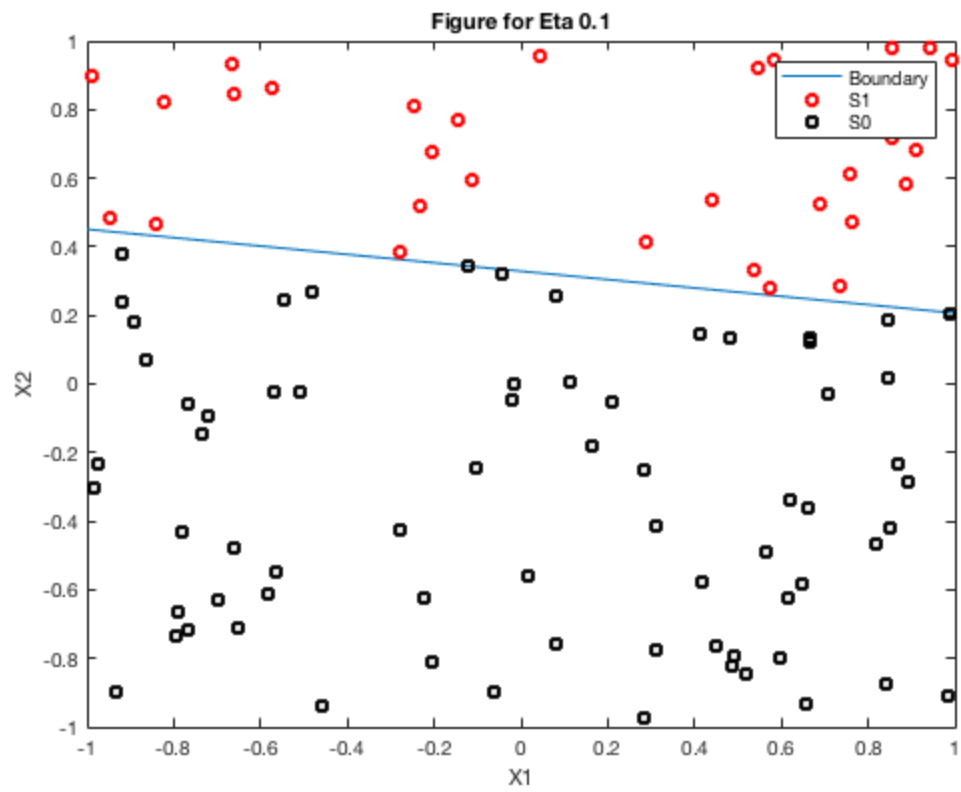
-0.4374    0.1625    1.3302

```









Published with MATLAB® R2017b