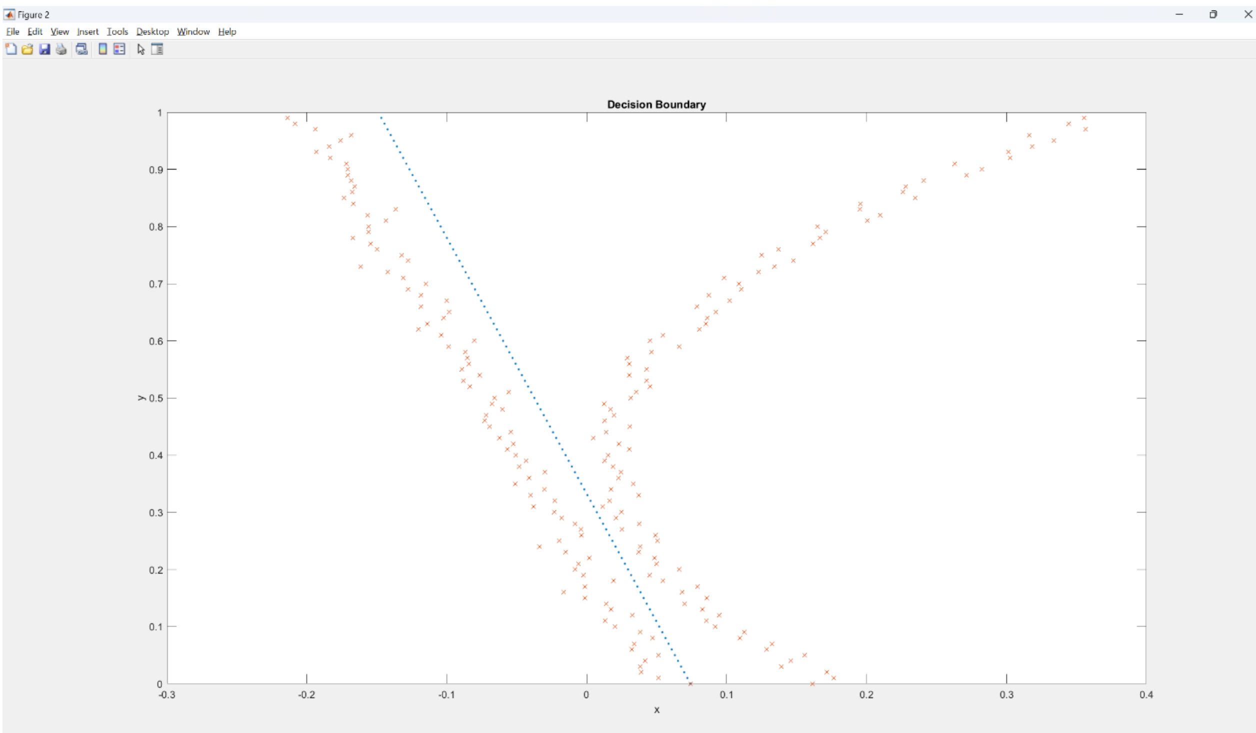
HW2

Q1

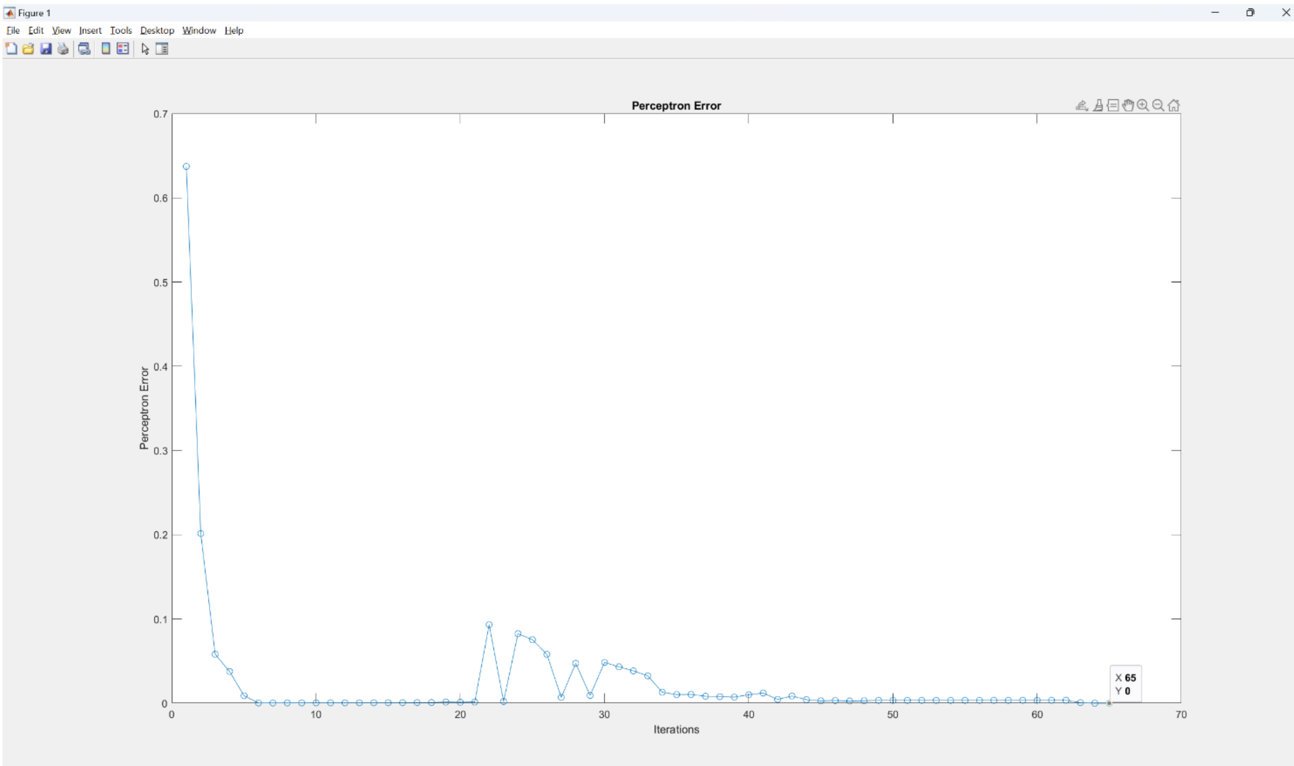
This project implemented linear perceptron using stochastic gradient descent. The following are results generated from a successful run (65 iterations).

When converged, both binary classification error and perceptron error reaches zero (the minimal).

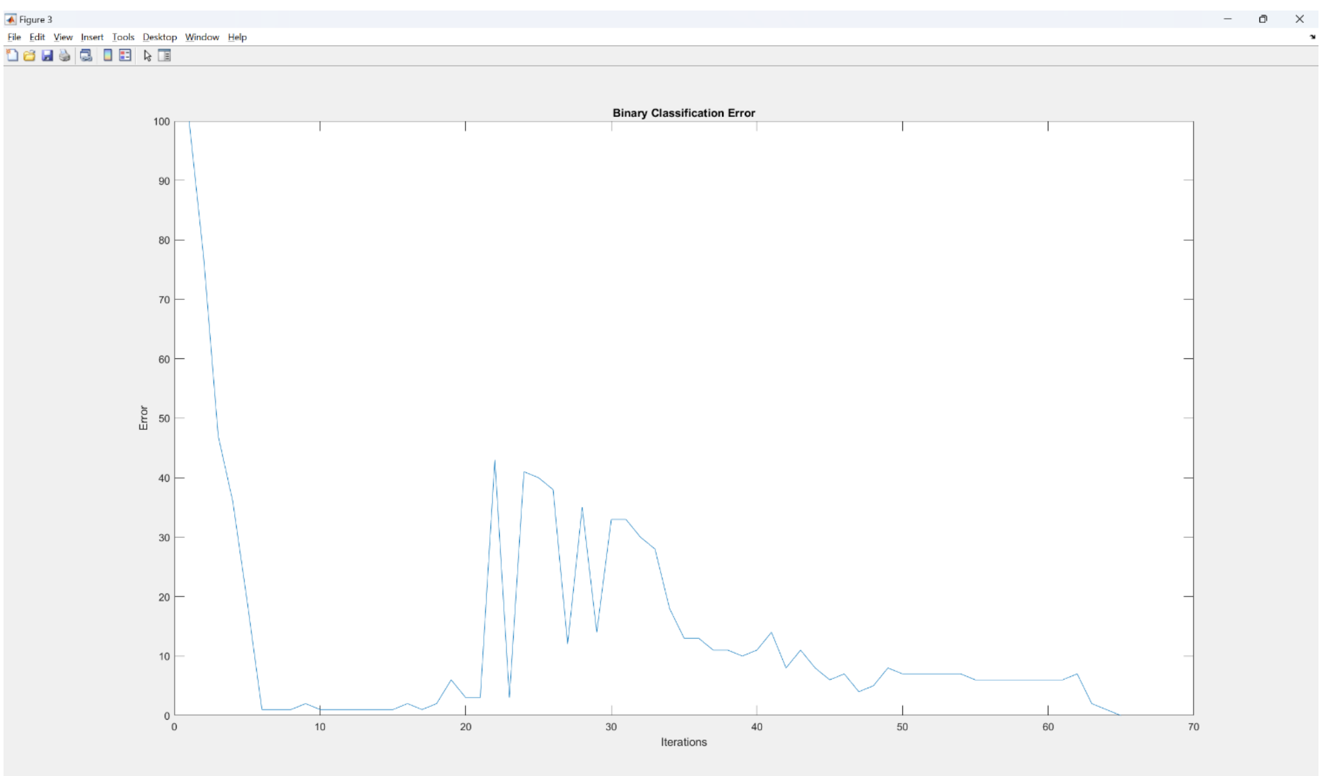
Decision Boundary



Perceptron Error



Binary Classification Error



Code Screenshot

Perceptron.m

```
function [theta, iterations,x, Errors, Risk] =  Perceptron(Data)
    theta = rand(3,1) ; %3 1
    t = 1;
    y = Data(:,3); %200 1
    x = [ones(200,1) Data(:,[1,2])]; %200 3
    Errors = [];
    Risk = [];
    err = 10;
    while (err~=0)
        xx = [zeros(200,3)]; %miss classified x, reset
        yy = [zeros(200,1)]; %miss classified y, reset
        err = 0;
        temp = theta;
        k = 1;
        for i = 1:length(Data)
            if(y(i) * (x(i,:) * theta) <= 0)
                xx(k,:) = x(i,:);
                yy(k,:) = y(i);
                k = k+1;
                err= err + 1;
                %disp(xx);
            else
                err = err + 0;
            end
            Errors(t) = err;
        end

        if size(xx)>0 %calculate risk
            Risk(t) = (-1/length(yy))*sum(yy .* (xx * theta));
        end

        for i = 1:length(Data) %update theta
            if(y(i) * (x(i,:) * theta) <= 0)
                theta = theta + (y(i) .* x(i,:)');
            end
        end
        t = t+1;

    end
        iterations = t-1;
end
```

Main.m

```
load data3.mat
[Theta, Iterations, x, Errors, Risk] = Perceptron(data);

% Perceptron error - iteration plot
figure
plot(1:length(Risk), Risk, '-o');
xlabel("Iterations");
ylabel("Perceptron Error")
title('Perceptron Error');

% Linear decision boundary plot
figure;
A = -(Theta(2)/Theta(3)) * x(:,2) - (Theta(1)/Theta(3));
B = x(:,2);
plot(A,B, '.');
hold on
plot (x(:,3), x(:,2), 'x');
xlabel("x");
ylabel("y");
title('Decision Boundary');

%Binary classification error plot
figure;
plot (1:length(Errors), Errors);
title('Binary Classification Error');
xlabel("Iterations");
ylabel("Error")
```

Q2.1

$$E = -(t_i \log(x_i) + (1-t_i)\log(1-x_i))$$

$$x_i = \frac{1}{1+e^{-s_i}}, \quad s_i = y_j w_{ji}$$

$$\frac{dE}{dw} = \frac{dE}{dx}\cdot\frac{dx}{ds}\cdot\frac{ds}{dw}$$

$$\frac{dE}{dx} = -t_i \log(x_i)' - (1-t_i)\log(1-x_i)' = -\frac{t_i}{x_i} + \frac{1-t_i}{1-x_i} = \frac{x_i - t_i}{x_i(1-x_i)}$$

$$\frac{dx}{ds} = \frac{d}{ds}\left[(1+e^{-s})^{-1}\right] = (-1)(1+e^{-s})^{-2}\cdot(-e^{-s}) = x_i(1-x_i)$$

$$\frac{ds}{dw} = y_j' w_{ji} + y_j w_{ji}' = y_j$$

$$\frac{dE}{dw} \Rightarrow (x_i - t_i)y_j$$

Q2.2

$$\frac{dE}{ds} = \frac{dE}{dx}\cdot\frac{dx}{ds}$$

① $i \neq c$ $\quad \dfrac{dx_i}{ds_c} = \dfrac{d}{ds_c}\left(\dfrac{e^{s_i}}{e^{s_c} + \sum_{c\neq i} e^{s_c}}\right) = -x_i \cdot x_c$

② $i = c$ $\quad \dfrac{dx_i}{ds_c} = x_i(1-x_i)$

$$\Rightarrow \frac{dE}{ds} = -\frac{d}{ds}\left[y_i \log(x_i) + \sum_{c\neq i} y_c \log(x_c)\right]$$

$$= -\frac{y_i}{x_i}\cdot\frac{dx_i}{ds_i} - \sum_{c\neq i}\frac{y_c}{x_c}\cdot\frac{dx_c}{ds_i} = x_i - y_i$$

Q3

$$H = -\Sigma_{k=1}^{N} P_k \ln(P_k)$$

$$J(p) = \Sigma P_k \ln(P_k) - \lambda_0 (\Sigma P_k - 1)$$

$$\frac{\partial J}{\partial P_k} = 1 + \ln(P_k) - \lambda_0 = 0$$

$$\Rightarrow \ln(P_k) = 1 - \lambda_0$$

$$P_k = e^{1-\lambda_0}$$

Therefore: $\Sigma P_k = 1 = \Sigma e^{1-\lambda_0}$

$$\Rightarrow \Sigma e^{1-\lambda_0} = 1$$

$$e^{1-\lambda_0} (N-1) = 1$$

$$e^{1-\lambda_0} = \frac{1}{N-1}$$

$$-\lambda_0 + 1 = \ln \frac{1}{N-1}$$

$$\lambda_0 = 1 - \ln \frac{1}{N-1}$$

$$\Rightarrow P_k = e^{1-\lambda_0}$$

$$P_k = e^{1-(1-\ln \frac{1}{N-1})}$$

$$P_k = e^{\ln \frac{1}{N-1}}$$

$$P_k = \frac{1}{N-1}$$

Q4

Solution:

For three points, Let A, B, C be (1,0), (0,1), (-1,0) in a Cartesian coordinate system. Label two of these points positive, an axis-aligned square with these two points as its corners can shatter and classify the three points correctly. Therefore, the VC-dimension is at least three. For four points, Let A be the highest point on y-axis, B be the lowest point on y-axis, C be the leftmost point on x-axis, D be the rightmost point on x-axis. Assume dAB is greater than dCD, without losing generality. When A and B are positive, C and D cannot be labeled negative. Therefore, the VC-dimension of axis-aligned square is 3.