Ismat Jarin

Midterm Project

. . . . . .

. . . . . .

. . . . . .

. . . . . .

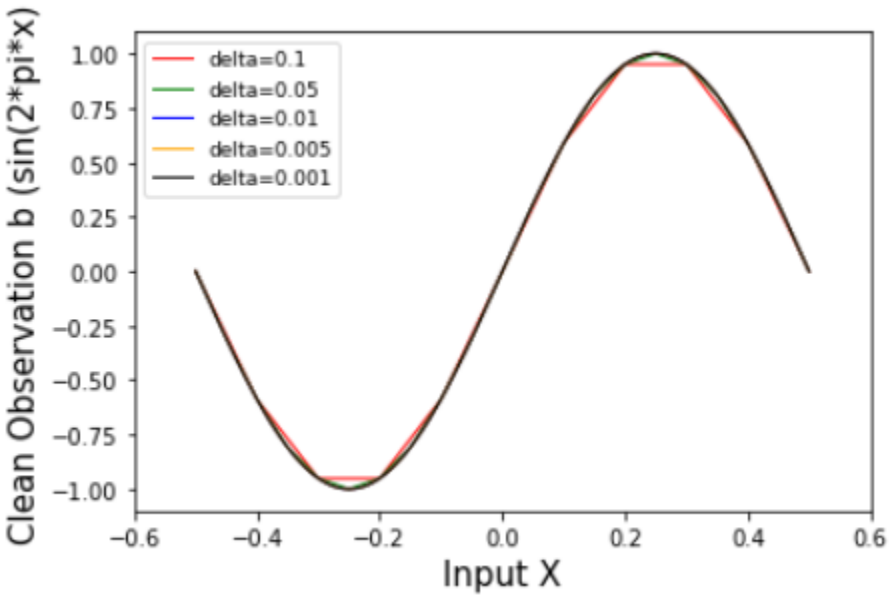
Solution. Question 2

For 2nd Order Model:

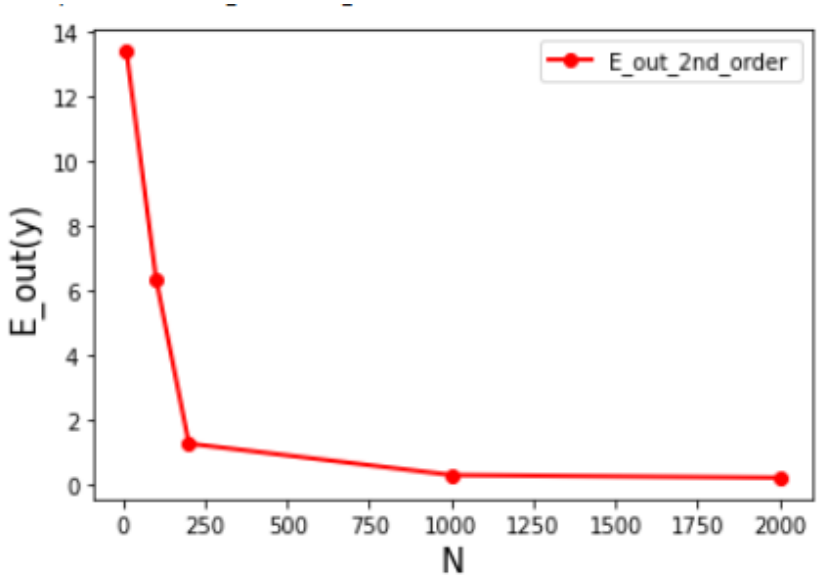
Here, x = [-0.5:delta:0.5]

b = sin(2*pi*x) - 0.1 + 0.2*rand(1,N)

1. Plot input ‘x’ vs ‘sin(2*pi*x) for 2nd order Model :



2. The plot for E_out for 2nd order model for different N parameters:



Experimental Environment:

Number of experiments I have run is=30. I have implemented the code in the way that one can vary number of experiments.

The delta values for the experiments are: [.1,.01,.005,.001,.0005]

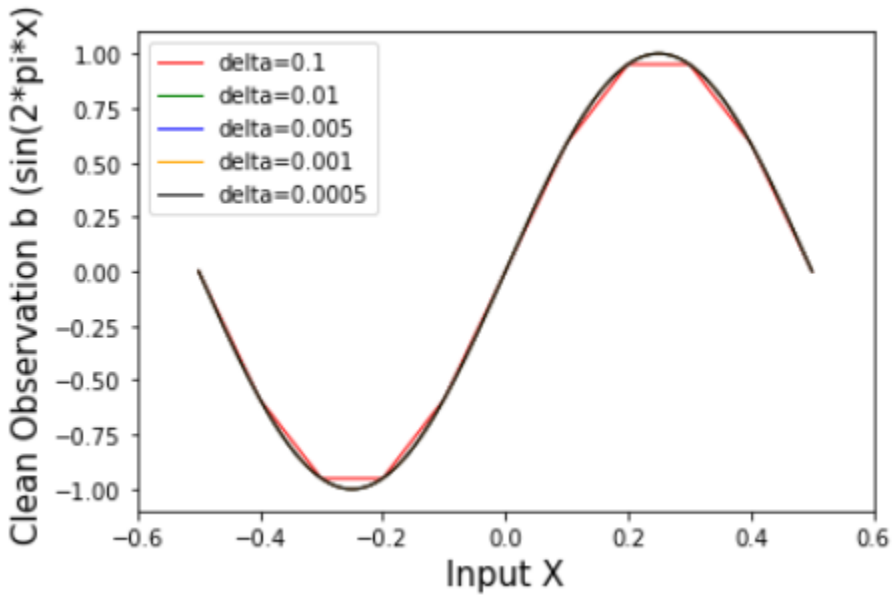
Thus the number of N's are: [11. 101. 201. 1001. 2001.]

Output Evaluation:

From the above graph we can clearly observe that as the input data increase, the E_out is decreasing for second order model which clearly supports our discussion in class that 'We have the intuition that the Model will give better output if the model is trained on a large dataset'.

For 10th Order Model:

1. Plot input 'x' vs sin(2*pi*x) for 10th order Model :

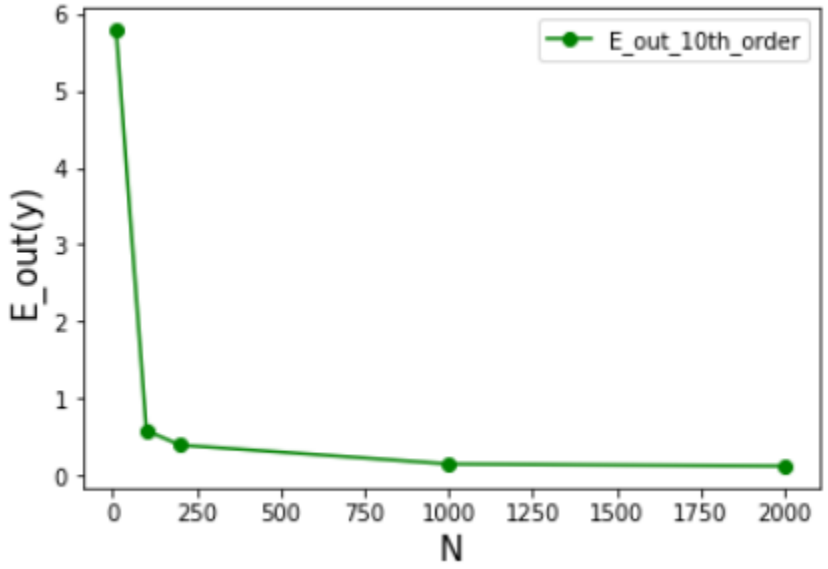


1. The plot for E_out for 10^{th} order model for different N parameters:

Experimental Environment:

The experimental setup is same as the 10^{th} order model

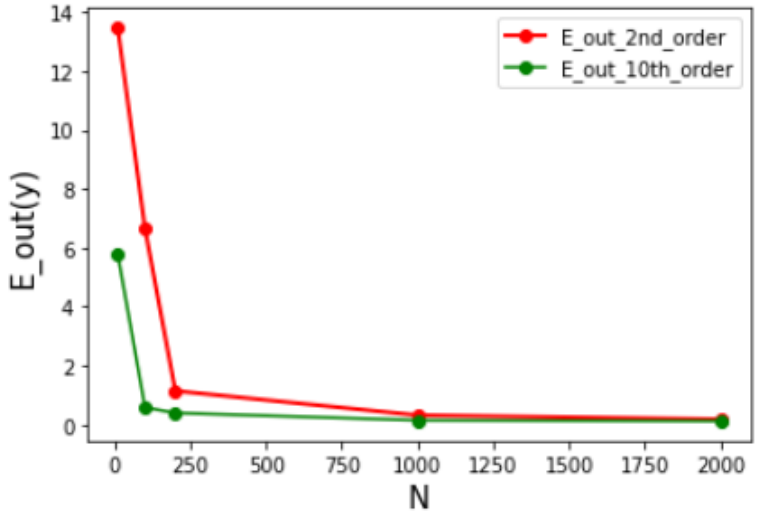
2. The plot for E_out for 2^{nd} order model for different N parameters:



Output Evaluation:

From the above graph we can clearly observe that as the input data increase, the E_out is decreasing for 10^{th} order model which clearly supports our discussion in class that 'We have the intuition that the Model will give better output if the model is trained on a large dataset'.

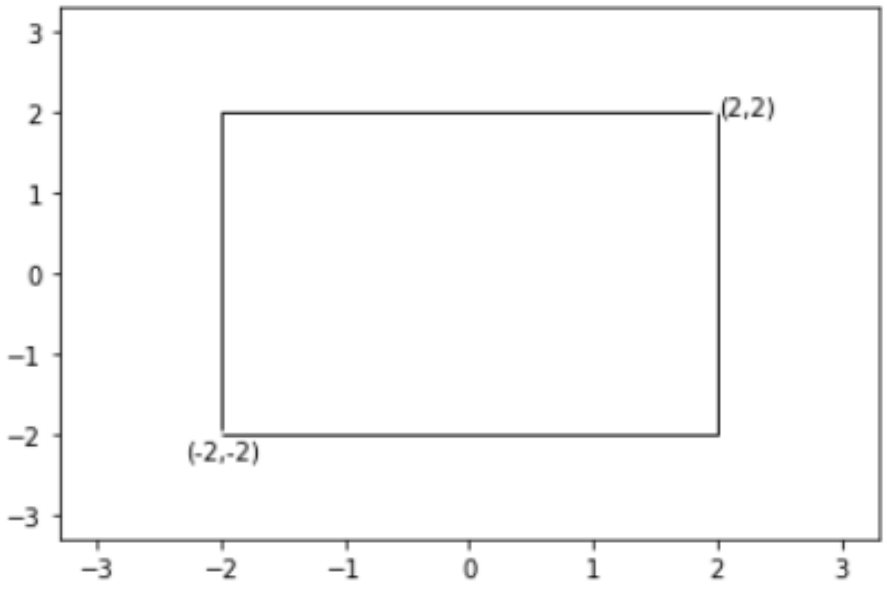
If We plot 2nd order and 10th order Model in one plot:



From the observation, we can clearly estimate that, for this experiment 10th order model should be the right choice performance wise.

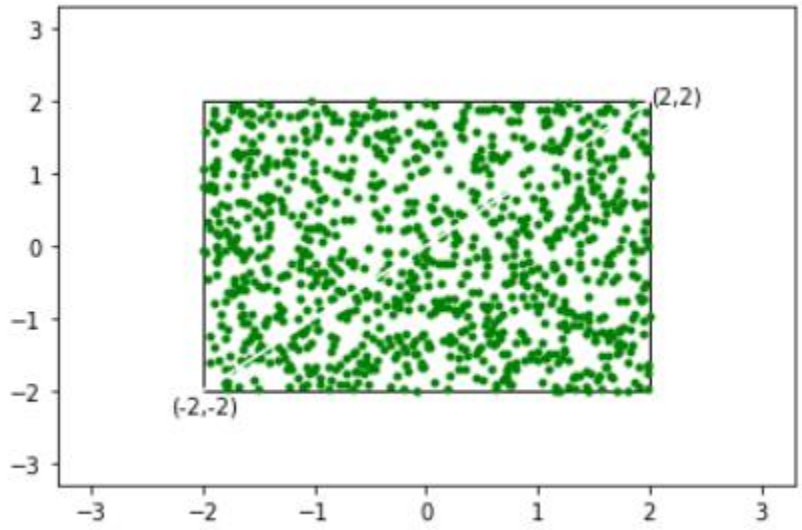
Solution. Question 2

1. Plot an rectangular box while consider (x1,y1)=(-2,-2) and (x2,y2)=(2,2) :



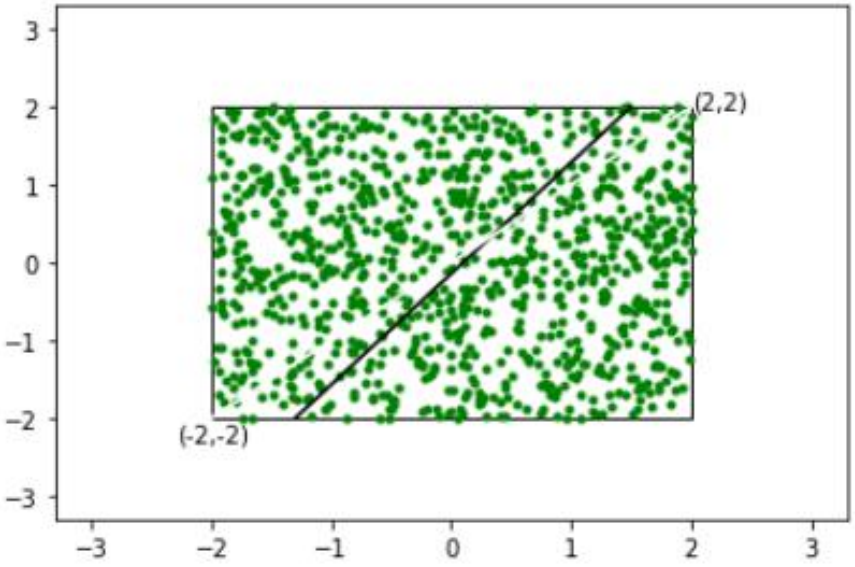
Now Draw a random line over the box:

For this, I have generated 1000 random data within the box. The scatter plot for the data is below:



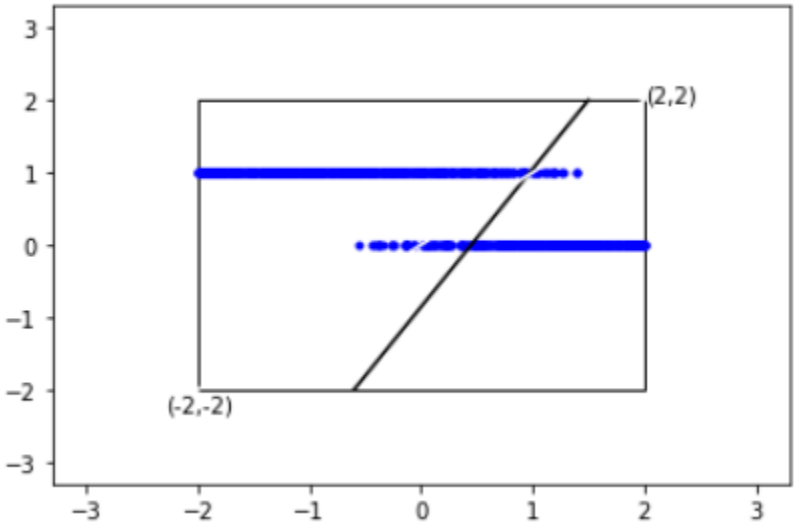
Now I have to take two random points and draw a line to divide the data. I have choose these two points in a way that there is difference in x value, otherwise the output will be confused with respect to feature and the model will be poor. But I put a choice in code that one can choose different random values for x and y.

Plot a random line through the box:



Split them with 1 and zeros according to line (Here the line will be the threshold):

So, the points above the line will be labeled as ‘1’ and point below the line will be labeled as ‘0’.



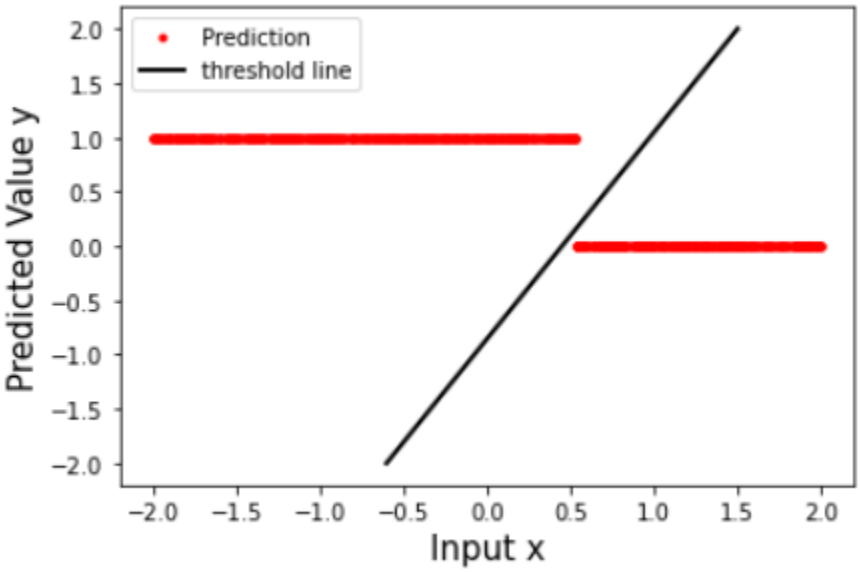
Experimental Setup:

We have used some random weight in [-1,1] scale and try to separate the data using perceptron learning algorithm.

The learning rate=.5, number of epoch=500

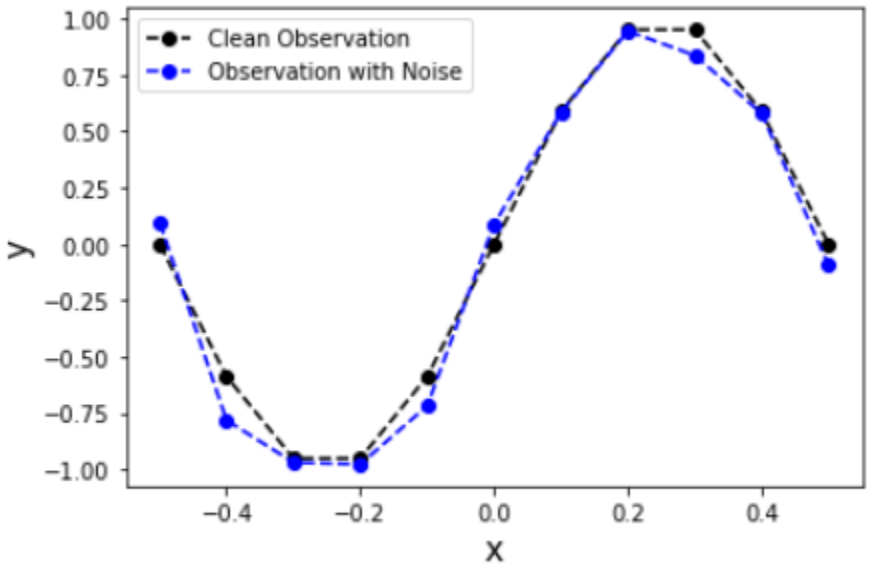
The error value is= 168

The plotted Predicted value:



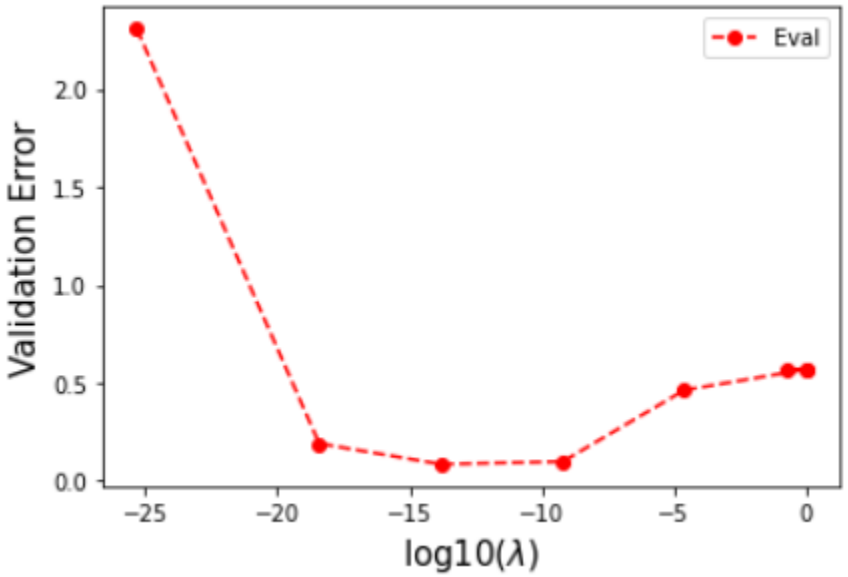
Solution. Question1

Plot for clean observation and noisy observation is given below:



The validation error with respect to different regularization parameter is given below:

(We have used leave one out cross validation). The lambda is presented as log10 scale to ease the visualization.



From the code we have found that the best regularization parameters where the validation error is minimum is 10e-6 for my experiment. This parameter can be varies based on random input data.