Assignment No: 1
Name : I.Krutika
Roll No : 18MCMT20

Neural Network Assignment

A) Is this dataset linearly separable? Take a smaller subset (2 data points from each class) and hand-draw the feasible region of solution (where weight vector W * resides) for this smaller subset. [Hint: Subtract the class-mean from these vectors to plot the homogeneous solution].

Given Data set:
c 1 = [(1; 6); (7; 2); (8; 9); (9; 9); (4; 8); (8; 5)]
c 2 = [(2; 1); (3; 3); (2; 4); (7; 1); (1; 3); (5; 2)]

Yes the given data set is linearly seperable as a line can be drawn inorder to seperate them.

Let us consider two points from each of the classes
c1 : (8,9); (8,5) => Target output is [0,0]
c2 : (3,3); (1,3) => Target output is [1,1]

Now let us take the mean of all the x and y co-ordinates and subtract them from the above points to get homogeneous solution.

Taking Mean:
meanX=(8+8+3+1)/4 = 20/4 =5
meanY=(9+5+3+3)/4 = 20/4 =5

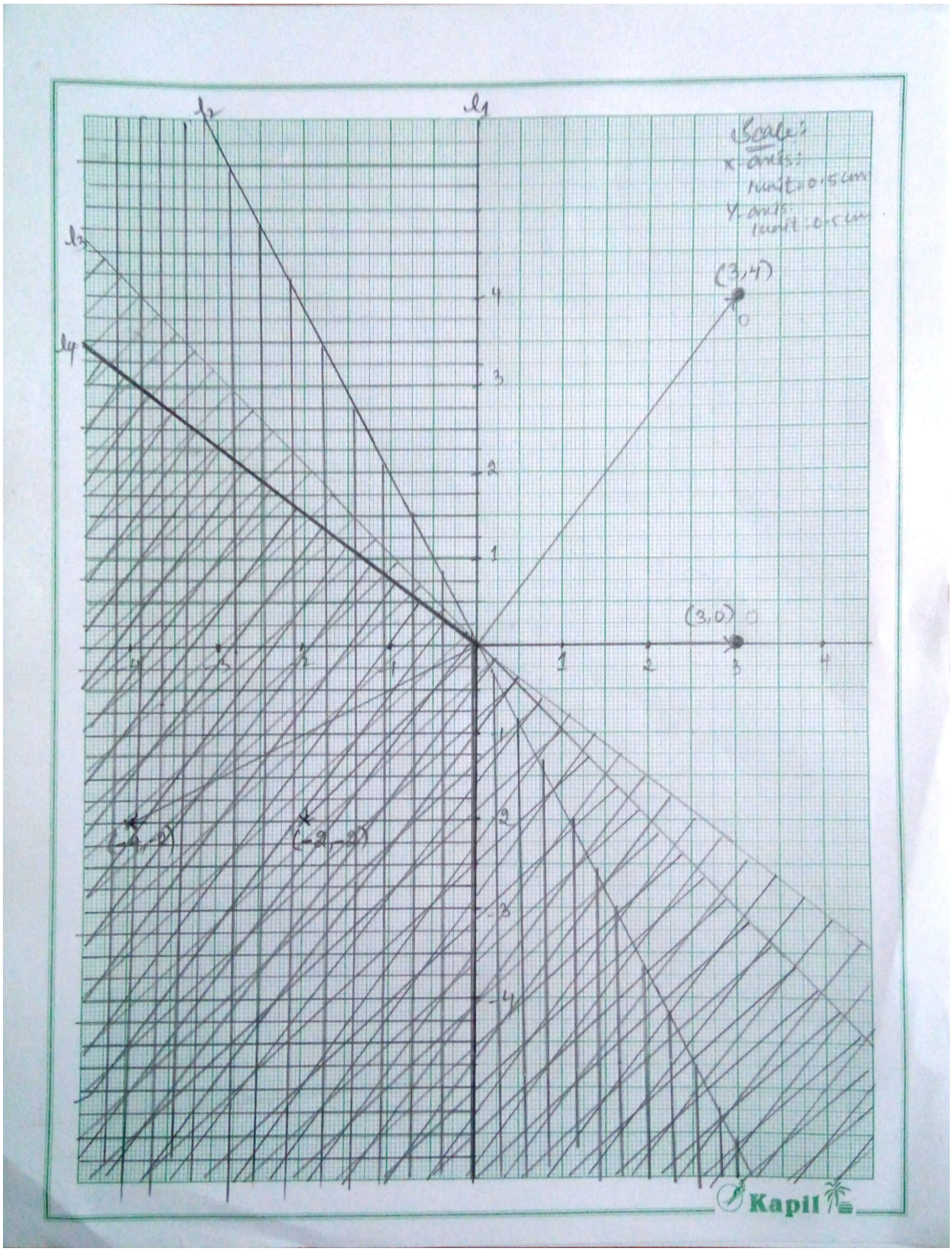
Subtraction mean from the class points

c1: (8-5,9-5); (8-5,5-5); => (3,4);(3,0);
c1: (3-5,3-5); (1-5,3-5); => (-2,-2);(-4,-2);

Now we get 4 lines that are perpendicular to these 4 points. Below are the equations of the four lines:
l1 : x <= 0 (Y- axis)
l2 : y >= -2x
l3 : y >=-x
l4 : y<=-(3/4)x

Below is the graph that gives the feasible region. It is observed that the feasible region is bounded by lines l1 and l4.

Assignment No: 1
Name : I.Krutika
Roll No : 18MCMT20

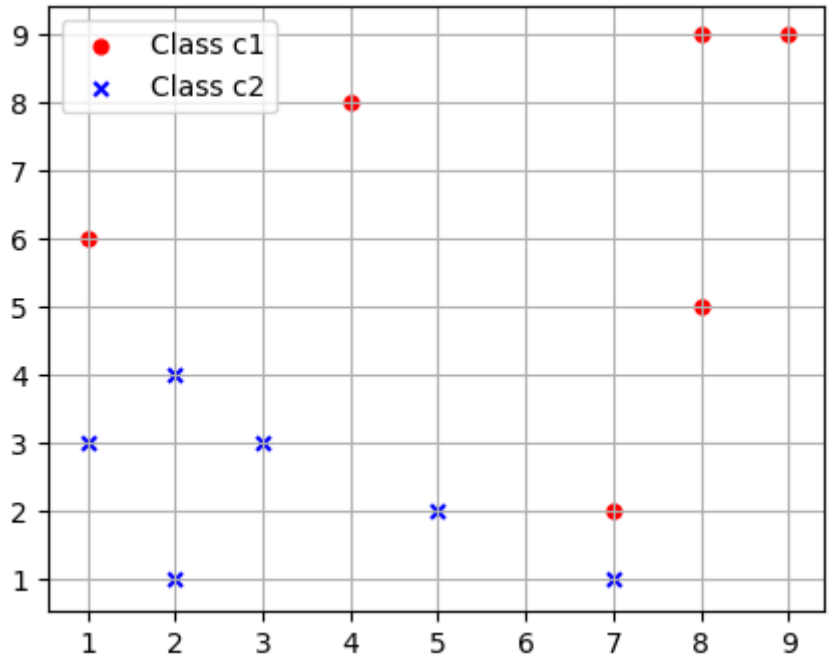


Assignment No: 1
Name : I.Krutika
Roll No : 18MCMT20

Representation of points using Matplotlib:

For this, let us draw a graph that clearly distinguishes between the two classes c1 and c2.
Note :

- c1 is represenetd using BLUE(Circle)
- c2 is represented by RED(Cross)



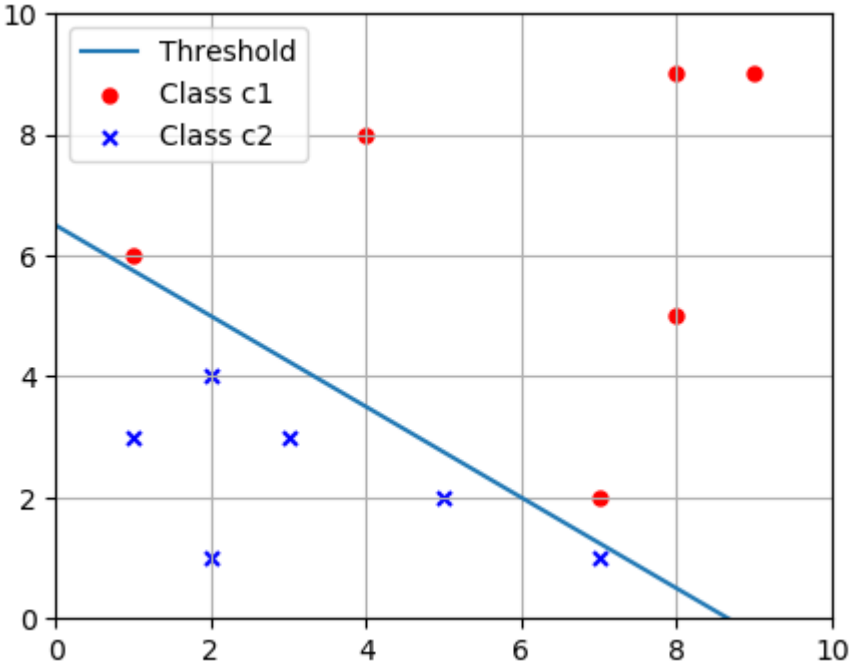
1) *Perceptron with threshold activation function:*

Inputs Given: Data set, $\eta=1$, number of epochs =5000, initial weights =[0,0,0]

Output obtained:

Updated Weights : [26. -3. -4.], Epochs Taken : 45

Graph:



Assignment No: 1
Name : I.Krutika
Roll No : 18MCMT20

Observartions:

- Converges very quickly when compared to Widrow Hoff and Sigmoid
- Learning rate can be kept higher than the other two algorithms
- Number of epochs taken to converge is 31

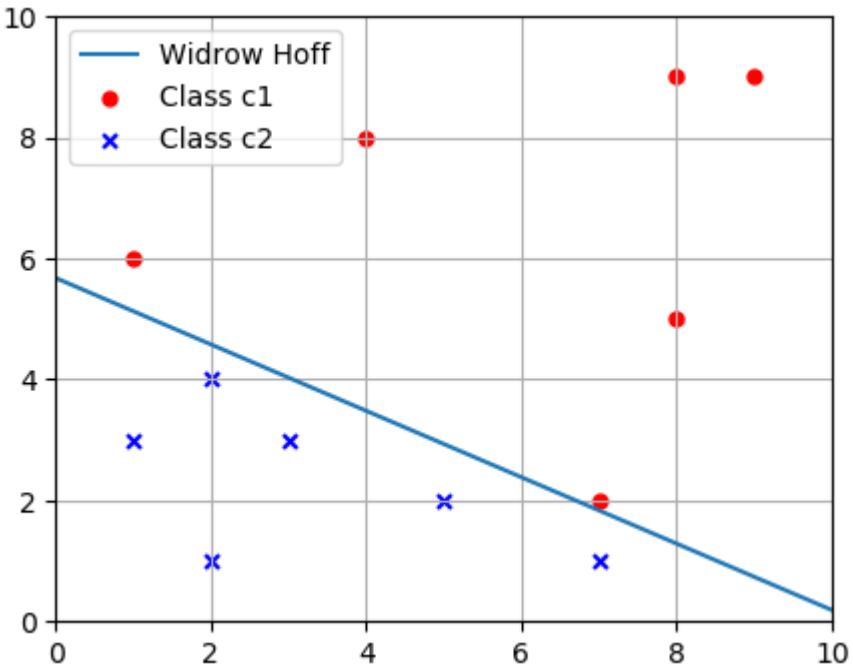
2) *Perceptron with Widrow-Hoff or Least Mean Squared (LMS) Learning Rule:*

Inputs Given: Data set, η=0.05, number of epochs =5000, initial weights =[0,0,0]

Output obtained:

Updated Weights : [3.06455232 -0.29619638 -0.53974422], Epochs Taken : 4999

Graph:



Observartions:

- Slowest of the all the algorithms
- requires the lowest learning rate.
- Increase in the learning rate does not give expected output
- Number of epochs taken to converger is throughout the epochs
- Zero error is not obtained
-

3) *Perceptron with Sigmoid Activation Function:*

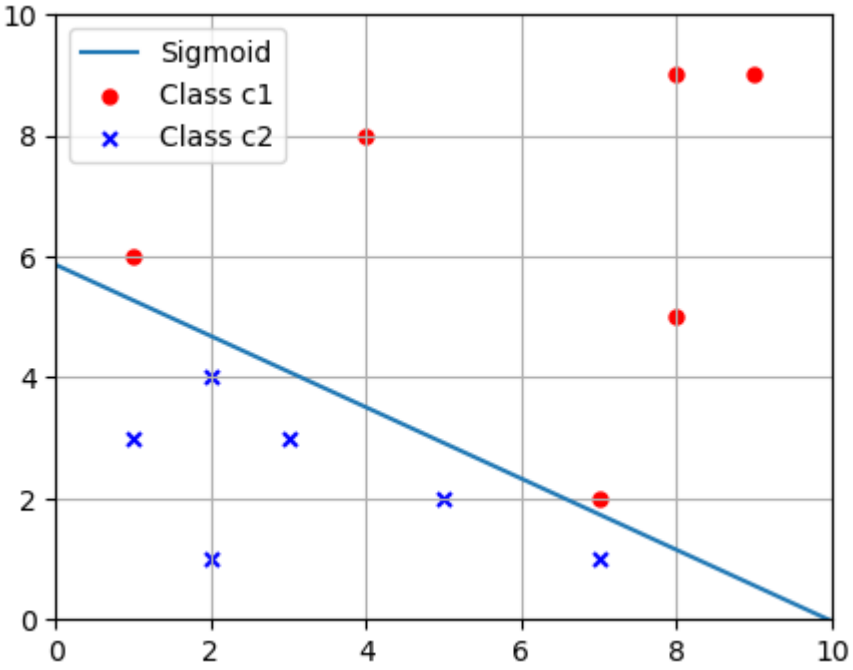
Inputs Given: Data set, η=0.001, number of epochs =5000, initial weights =[0,0,0]

Output obtained:

Updated Weights : [16.78804273 -1.68632536 -2.86309005],Epochs Taken : 1616

Assignment No: 1
Name : I.Krutika
Roll No : 18MCMT20

Graph:

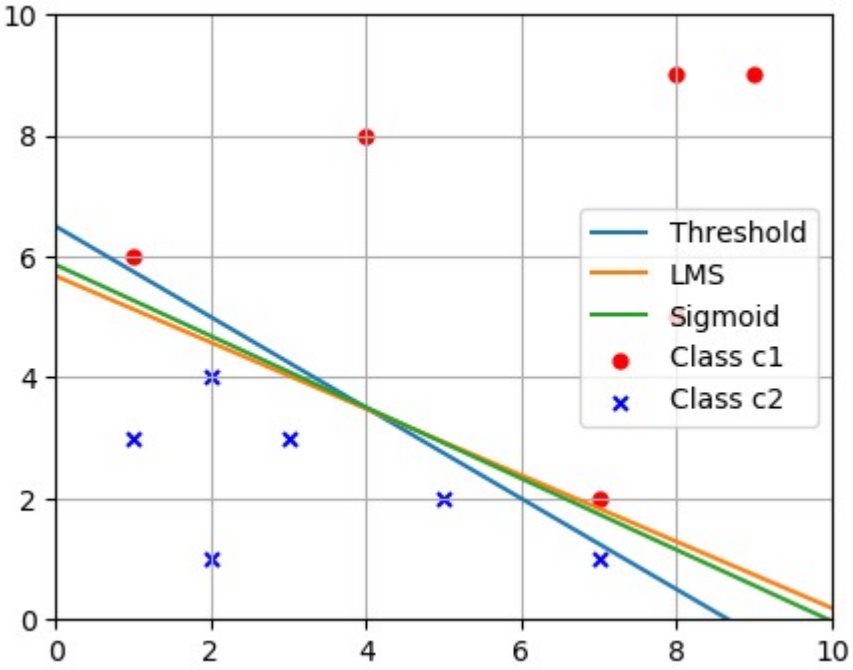


Observartions:

- Takes in moderate learning rate
- Number of epochs taken to converge is 1616
- Slower than basic perceptron learning rule but faster than LMS

B. In each case, plot the data points in a graph (e.g. Circle: class-c 1 and Cross: class-c 2) and also show the weight vector W * learned from all of the above algorithms in the same graph (labeling clearly to distinguish different solutions).

Graph:



Assignment No: 1
Name : I.Krutika
Roll No : 18MCMT20

Observation:

All the three algorithms have learned to classify the points correctly. It can be seen through graph that the decision boundary is exactly in between the two classes

C. Create a test set comprising three more data samples (x p) for each class and test your implementation by computing for each of the test samples the output (class label) predicted by the respective algorithm. Create a comparison table listing test set accuracies of each of the above algorithms.

Test Data

- class c1: [8,8],[10,8],[6,10] => Target output 0
- class c2: [2,2],[0,0],[4,0] => Target output 1

Output obtained:

System	Accuracy
BTU	100.0%
LMS	100.0%
Sigmoid	100.0%

Observation:

All the test data is classified correctly. 100% accuracy is achieved in all three algorithms.

D. Run each of the above algorithms for various initial values of the weight vector, and comment on the dependence of convergence time (run-time) on initialization.

Here we are using below 5 weights against each of the implemented algorithm.

- w1 = [.5,.5,.5]
- w2 = [-10,-10,-10]
- w3 = [-25,-25,-25]
- w4 = [50,50,50]

Observations:

1) Threshold and Sigmoid:

-ve weights => slower convergence ,

+ve weights are increased => faster convergence

2) LMS :

This is running through out the epochs. For larger weights converging faster.

- It is observed that among the weights of same amgnitute but different sign the +ve weights converge faster than -ve weights

Output obtained:

Weights	System	Stoch	Epochs
[0 0 0]	BTU	0.033	45
[0.5 0.5 0.5]	BTU	0.019	36
[-10 -10 -10]	BTU	0.023	50
[-25 -25 -25]	BTU	0.020	45
[25 25 25]	BTU	0.009	19
[0 0 0]	LMS	2.147	4999
[0.5 0.5 0.5]	LMS	2.090	4999
[-10 -10 -10]	LMS	2.096	4999
[-25 -25 -25]	LMS	2.124	4999
[25 25 25]	LMS	0.177	440
[0 0 0]	Sigmoid	0.750	1616
[0.5 0.5 0.5]	Sigmoid	0.742	1609
[-10 -10 -10]	Sigmoid	0.776	1687
[-25 -25 -25]	Sigmoid	0.808	1768
[25 25 25]	Sigmoid	0.015	28

F. Modify the dataset such that it becomes linearly non-separable (clearly list the changes in the report). You know that algorithm (1) will fail to converge for this dataset. Now run algorithms (2) and (3) with suitable stopping criteria. Plot the data points in a graph (e.g.Circle: class- c 1 and Cross: class- c 2) and also show the weight vector W * learned from these two algorithms in the same graph (labeling clearly to distinguish different solutions) andcomment on the nature of the solution found in each case.

Half of the vectors in c1 are moved to c2 and vice versa. Datasets formed are:

1) class c1: [2, 1],[7, 2],[2, 4],[9, 9],[1, 3],[8, 5] => Target Output 0

2) calss c2: [1, 6],[3, 3],[8, 9],[7, 1],[4, 8],[5, 2] => Target Output 1

Stopping Criteria:

- *LMS:* The mean squared error should be <=0.45
- *Sigmoid:* The absolute mean error should be <=0.05

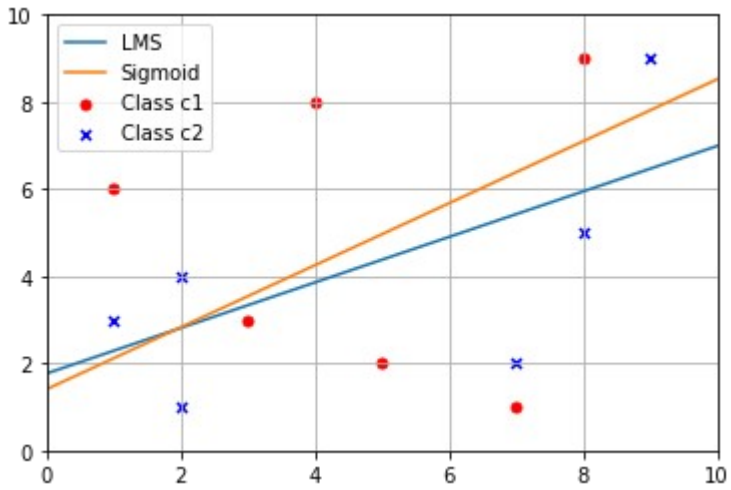
Input given: Data set, η=0.001 for Sigmoid,η=0.001 for LMS, number of epochs =5000, initial weights =[0,0,0]

Output obtained:

LMS : Epochs Taken : 4861 : Updated Wieghts : [0.16696455 0.04930653 -0.09443461]
Sigmoid : Epochs Taken : 1603 : Updated Wieghts : [0.16189511 0.08183828 -0.11505885]

Assignment No: 1
Name : I.Krutika
Roll No : 18MCMT20

Graph:



Observations

- It is observed that the decision boundary is between the midpoints dividing the ponts equally such that there is 50% accuracy.
- The slope of decision boundary is +ve in Non Linear case but in Linear seperable data the decision boundary slope was -ve
- The learning rates are kept sufficently small so that the learning takes place at a slower rate when compared to that of Linear Seperable data.
    - LMS: $\eta=0.0001$
    - Sigmoid: $\eta=0.001$