Neural Networks LAB-1

Report Submitted by Muhammad Farhan Ahmed

Task -1 : Simple 2D perceptron for binary problems, Mannual Design

Pseudo code:

1. Read given dataset.
2. Plot dateset with corresponding classes(targets).
   a. Patterns with classes =1 as "+".
   b. Patterns with classes as = -1 "*".
3. Manually find a separating hyperplane (line) as data in 2D.
   a. Manually assign weight vactor(w) as w0=1,w2=1,w3=-1.
   b. Plot equation of line as
      Y=-(w0/w1)*x-(w2/w1).
4. Test each perceptron with with weights effected by ramdom pertubations
   a. Max= 10% of norm of weight vactor.
   b. Perturbation values b/w 0 and ma with stepsize of max/10.
   c. Compute no.of pertubations(length of perturbation values ,num_pert).
   d. Initialize errors as zeros.
5. For each perturbation perform fairly large no. of random trials e.g 100.
   a. Main loop for i=1 to num_pert.
   b. Initlize weight rand_vactor as zeros(1,3).
   c. Make it unit norm.
   d. Compute new weight vactor (W) as W= w+(urand_vac*pert_values(i))
   e. Plot new line(hyperplane) in green color
   f. Check error.
   g. If there is error update error .
   h. End of for loop
   i. End of main loop.
   j. Compute average errors corresponding to num_pertibations.
6. Plot a graph of average errors.

Task -2 and Task 3 : **Simple perceptron for Iris data, manual design and perceptron array.**

Pseudo code:

1. Read dataset 'iris.txt'.
2. Multiply first two and last two input vectors to make sepal area and petal area.
3. N= Number of training examples.
4. Add column of ones to take care about bias.
5. Separate class labels(victor)
6. Initialize random weights (Vector).
7. Initial value of eta(Learning rate as 0.1)
8. Plot data with corresponding classes for visualization.
9. Call the function my perceptron (X,Y,eta)

10. Plot the separating hyper plane with weight victor obtained by the prceptron for visualization purposes.

[Weight vactor (w)]= Myperceptron(dataset+bias(X),class labels(Y), initial learning rate (eta))

1. This function implements the Roseblatt's perceptron.
2. Input = X -> Training Data (with columns as feature vectors)
3. Y -> Correponding Class Label of the training data
4. eta -> Learning Rate
5. N = Number of training examples
6. M = Number of features + 1 (to take care of bias term)
7. w = Create a initial weight vector randomly.
8. Set a stopping criteria flag = -1; so that Process terminates when flag = 1.
9. Niter = 0; initialize No. of iterations to find the weights.
10. maxIter = 100 , Maximum number of iteration before search for linearly separable hyper plane continues.
11. Main while loop ,Run until correct weights are found or counter exceeds max iterations.
12. numErrors = 0 , keep track of number of errors in each iteration.
13. for i = 1:N ,For all training examples.
14. Net input on neuron membrane.
15. Compute Output using sign activation function .
16. Compute error .
17. Count number of errors made by hyper plane (corresponding to current set of weights)
18. Update Rule w = w + eta*(y-a)*x
19. Compute correction dw.
20. Apply update
21. End of for loop.
22. Increment iterations counter.
23. Check stopping criterion
24. End of main while loop.
    Delta (w) or correction
25. Finalization
    a. Print unsuccessful attempt if max iterations reached.
    b. Print successful attempt if weight vactor is found within the iteration window.

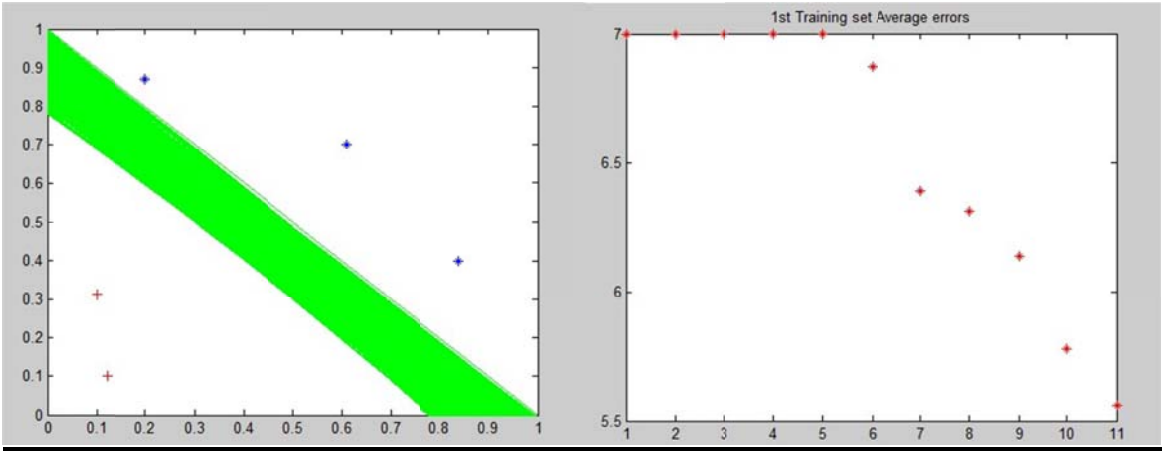
For Task 3 change the dataset as follows.

1. one with class 1 as 1 and 2 and 3 as -1;
2. one with class 2 as 1 and 1 and 3 as -1;
3. one with class 3 as 1 and 1 and 2 as -1.

Plot the results graphically.

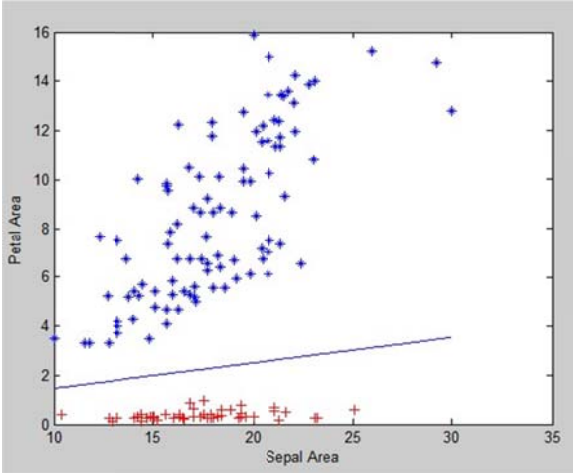
Results:

Task -1:



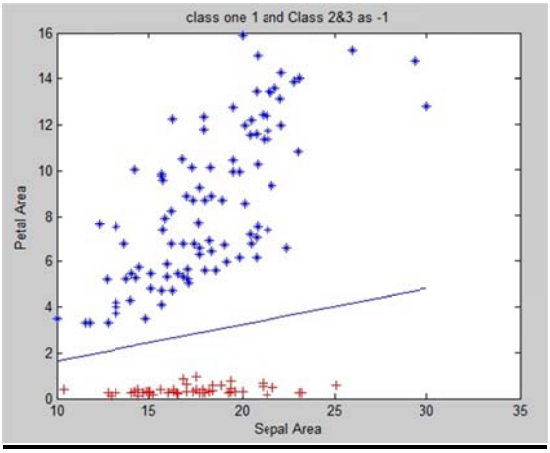
Task-2:

Eta = 0.01;

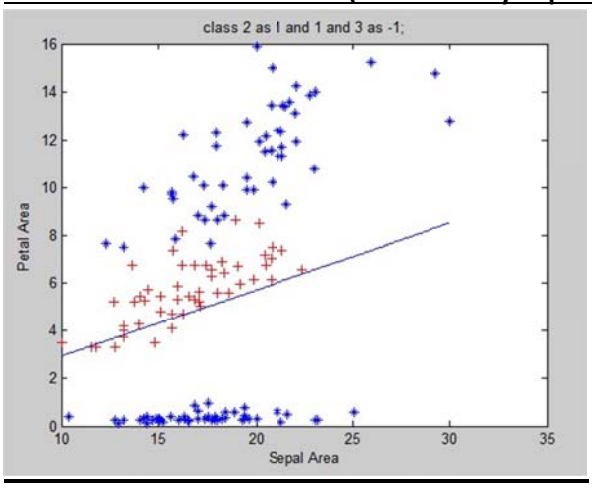


Task-3:

1. **class 1 as 1 and 2 and 3 as -1;**



2. **class 2 as 1 and 1 and 3 as -1(Non-Linearly Seperable)**



3. **class 3 as 1 and 1 and 2 as -1.(Non-Linearly Seperable)**