IT585 Advanced Machine Learning
Lab 3
Playing with the Perceptron Learning Algorithm

Instructions:

1. You have to code the solution in Google colab
2. You can use inbuilt libraries from python but not for the PLA algorithm. That needs to be coded by you
3. Your plots, code, any insights, observations written as text should be submitted as one ipynb file to google classroom
4. Deadline : February 11,2024 11:59 PM IST
5. Name of your file should be : yourrollno_lab3.ipynb

In this lab you will learn a popular learning algorithm which forms one of the basic parts of neural networks: the perceptron learning algorithm

For binary classification the perceptron learning algorithm will find a w using a simple iterative method .

At iteration t, where t = 0 , 1, 2, . . . There is a current value of the weight vector, call it w(t) .

The algorithm picks an example from (x1 , Y1) · · · (xN , YN) that is currently misclassified, call it (x(t),y(t)) , and uses it to update w(t) .

Since the example is misclassified, we have y(t) not equal to the sign(w^T(t) x(t)) . The update rule is

w(t + 1) = w(t) + y(t)x(t) .

Here for prediction the algorithm outputs sign(w^Tx) as the class of a point x. Labels are +1 and -1.

Let us create our own target function f and data set $\mathcal{D}$ and see how the perceptron learning algorithm works. Take $d = 2$ so you can visualize the problem, and choose a random line in the plane as your target function, where one side of the line maps to $+1$ and the other maps to -1. Choose the inputs $\mathbf{x}_n$ of the data set as random points in the plane, and evaluate the target function on each $\mathbf{x}_n$ to get the corresponding output y_n.

Now, generate a data set of size 20. Try the perceptron learning algorithm on your data set and see how long it takes to converge and how well the final hypothesis g matches your target f.

Now generate 1000 random points instead of 20 and repeat the experiment. Again apply the perceptron algorithm on the dataset and report the number of iterations it takes to give final results i.e. converge. Code the perceptron algorithm from scratch and also compare it with any python library implementation in terms of time, no. of iterations etc. Plot appropriate graphs to highlight

Next take the iris data set and divide it into 80-20 train test split. Since it has 3 classes use one vs all technique to generate two hyperplanes using the PL algorithm. Use the hyperplanes to classify the test data. Report your insights and results.