CSC 535/635 – HW 4

Perceptron Implementation

**Instructions:** Students are required to work alone on this assignment.

**Part 1 [75 points]:** For this assignment, you will **implement** the Perceptron algorithm as given on Module 10’s PowerPoint slides on Blackboard. Assume that there are two classes labeled 1 and -1. Use the training data provided in the file training\_data.txt. Use the testing data provided in the file testing\_data.txt. The third attribute in the datasets is the class label. Use a threshold activation function, which will work fine for linearly separable datasets.

For each test object, output the object followed by the actual class label followed by the predicted class label. This will be followed by the output of the overall accuracy rate, followed by the learned parameters (weights and bias.) Sample program output is as follows:

[ 5.32 0.37] Actual label: 1 Predicted label: 1  
[-4.06 -6.25] Actual label: 1 Predicted label: 1  
[-1.17 -6.83] Actual label: 1 Predicted label: 1  
[ 7.6 -4.52] Actual label: 1 Predicted label: 1  
[-1.72 -4.08] Actual label: 1 Predicted label: 1  
[ 2.58 1.6 ] Actual label: 1 Predicted label: 1  
[ 2. -4.68] Actual label: 1 Predicted label: 1  
[-4.31 -4.93] Actual label: 1 Predicted label: 1  
[-3.45 -7.12] Actual label: 1 Predicted label: 1  
[ 9.2 -6.23] Actual label: 1 Predicted label: 1  
[-5.56 7.41] Actual label: -1 Predicted label: -1  
[-5.87 8.37] Actual label: -1 Predicted label: -1  
[-0.23 2.23] Actual label: -1 Predicted label: -1  
[-8.39 4.77] Actual label: -1 Predicted label: -1  
[-6.69 9.28] Actual label: -1 Predicted label: -1  
[-9.51 -5.91] Actual label: -1 Predicted label: -1  
[ 4. 5.59] Actual label: -1 Predicted label: -1  
[-9.54 1.55] Actual label: -1 Predicted label: -1  
[-9.97 0.31] Actual label: -1 Predicted label: -1  
[ 2.8 9.71] Actual label: -1 Predicted label: -1

Accuracy rate: 100.00%

Learned weights are: [ 5.23962701 -5.18962127]  
Learned bias: 0.405526752143

**Part 2 [25 points]:** Use matplotlib to draw a scatter plot of the given **training** dataset. Show the objects in each class using a different color and/or marker. This dataset is linearly separable by a line that is not hard to figure out by looking at the plot and the model’s learned parameters. Provide a separate plot of the **testing** dataset and the **learned** **hyperplane**. An example of how to draw the hyperplane is shown in the SVM lecture slides.

# **What to turn in?**

Name your implementation file hw4.py or hw4.ipynb as appropriate. If you did part 2 in a separate file, name that file hw4\_plots.ipynb. Upload your files to Blackboard. If you need to upload more than one file, zip the files together and upload your zipped file to Blackboard. Please make sure that your code is well organized and properly documented and commented. Seed the random number generator so that we get the same results when we test your code.

At the top of your hw4.py or hw4.ipynb file, in a docstring, write down the following information:

* Your name
* The stopping criteria you used for your implementation of the perceptron algorithm
* The accuracy you got on the testing data
* Any comments about this homework assignment that you may want to share