Part B Report

Name: James Huang

Assignment 6: Perceptron Classification and Training

CSE 415 Introduction to Artificial Intelligence, Spring 2021, University of Washington

Please answer each question using text in Blue, so your answers stand out from the questions.
Note: If not otherwise specified, use the default parameters present in the code to answer the questions.

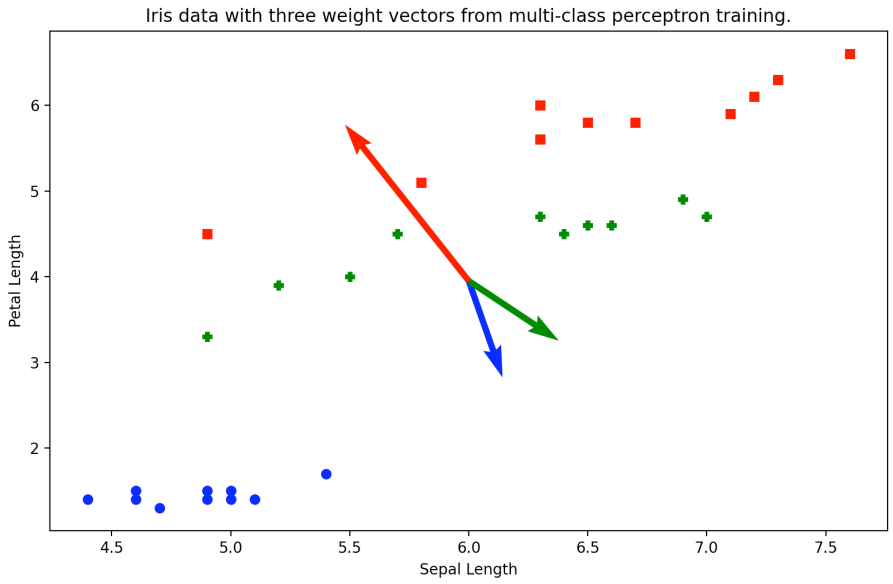
B1. How many epochs were required to train your perceptron on the 3-class Iris data having 4 features (the given training file, with 30 examples)? How many of the test data examples (out of 120) were mis-classified? Determine the percentage error rate and write that here.

85 epochs were required to train the perceptron

14 errors (3 linestyle errors) out of 120 were mis-classified

14/120 = 11.6% error

B2. Capture the plot that is produced by the program showing the training data and the weight vectors when projected onto the 2-D subspace spanned by sepal length and petal length (which is the starter-code default in run_3_class_4_feature_iris_data.py. Paste it here, reduced to fit in the remaining space on this page.



B3. In the file run_3_class_4_feature_iris_data.py, now modify the code so you can see the data projected onto the subspace spanned by features 2 and 3 (petal length and petal width). Describe the how the data seems to be distributed in this view. Describe how the weight vectors seem to be pointing. Finally, describe the relationship between the weight vectors and the distribution of the data.

blue data points are around the bottom left corner, green data points are around the center and the red data points are mostly located at the top right corner. Both green and blue weight vectors point at lower left direction and the red vector points to top right direction. Weight vector is like the classifier of the general direction where the data should located around.

B4. In the file run_3_class_4_feature_iris_data.py, instead of using all zero weight, let

W = [[1, 1, 1, 1, 1], [-1, -1, -1, -1, -1], [0, 0, 0, 0, 0]]. Now, for learning rates starting from 1e-3 to 1e+3 (all powers of 10), investigate how many epochs it takes for the model to converge. (You may similarly investigate the model for other initializations of W if you wish). Also, find the number of errors on the test set for each binary perceptron. What kinds of trends do you observe?

1e-3 : 261 epochs 11 errs

1e-2: 76 epochs 12 errs

1e-1: 57 epochs 17 errs

1e0: 50 epochs 10 errs

1e+1: 73 epochs 5 errs

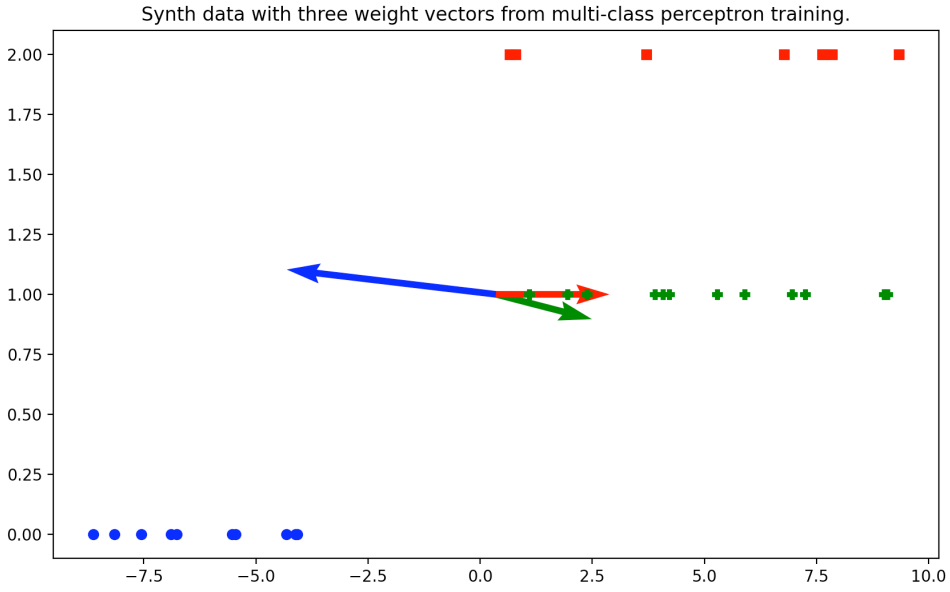
1e+2: 85 epochs 19 errs

1e+3: 85 epochs 14 errs

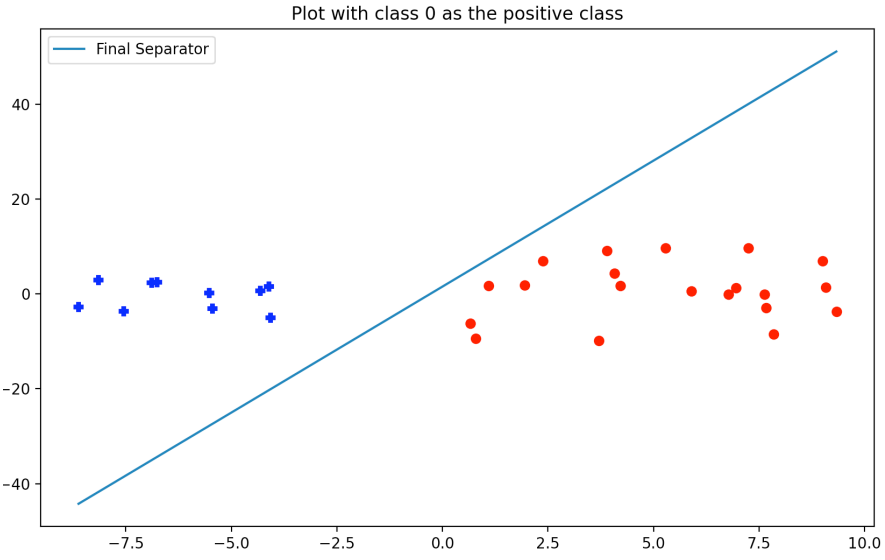
Number of epochs needed is decreasing as the learning rate is approaching to 1,

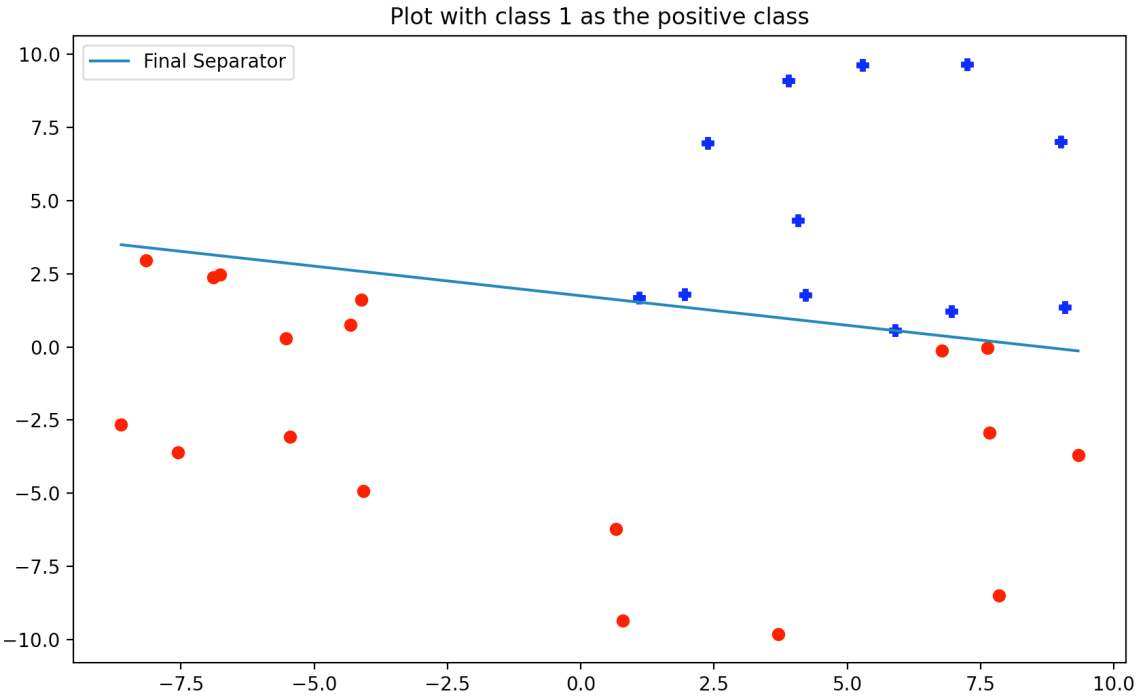
Number of epochs increase as it diverges from 1.

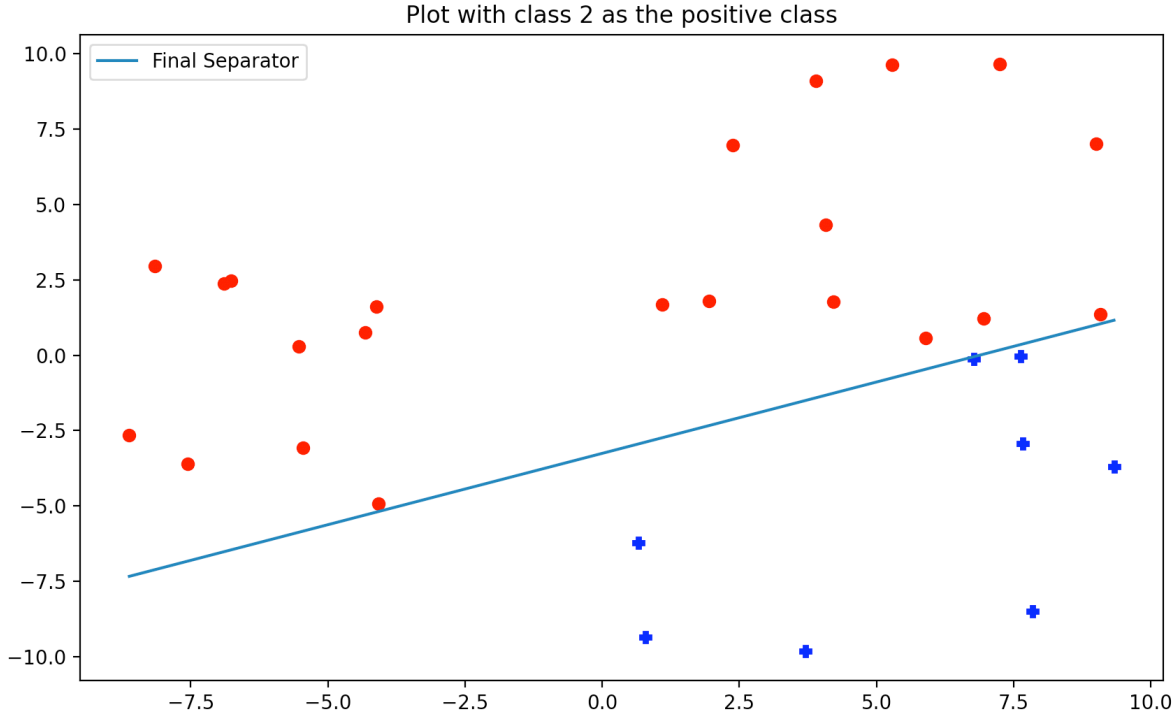
B5. Using the file run_synth_data_ternary.py, capture the plot of the ternary perceptron for the synthetic dataset and paste it here. (Let the maximum number of epochs be 50 and learning rate 0.5).



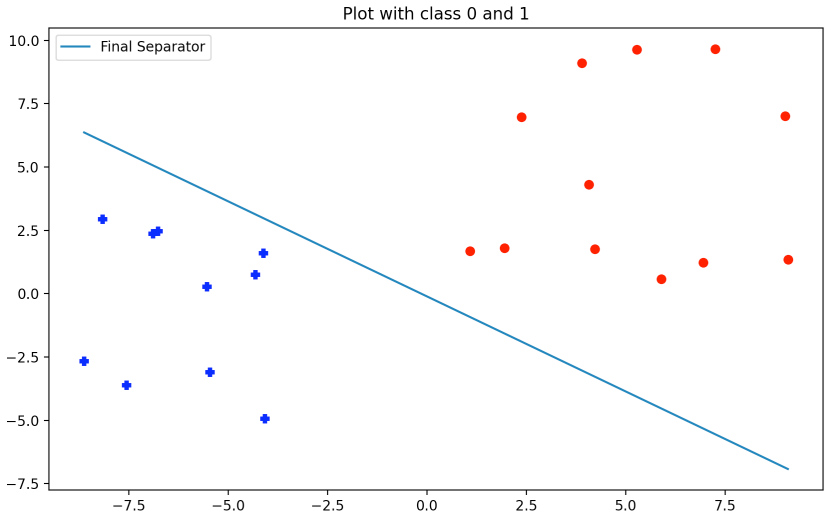
B6. Using the file run_synth_data_1_vs_all.py, capture the plots of all the One-Vs-All classifiers for the synthetic data and paste them here. (Let the maximum number of epochs be 50 and learning rate 0.5).

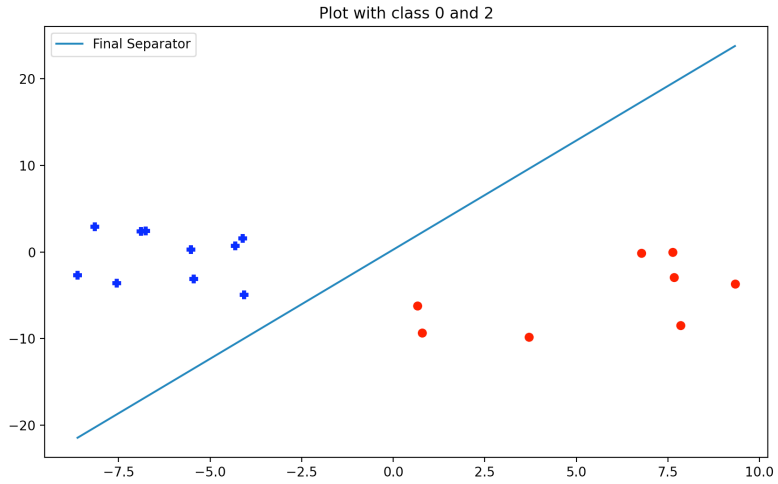


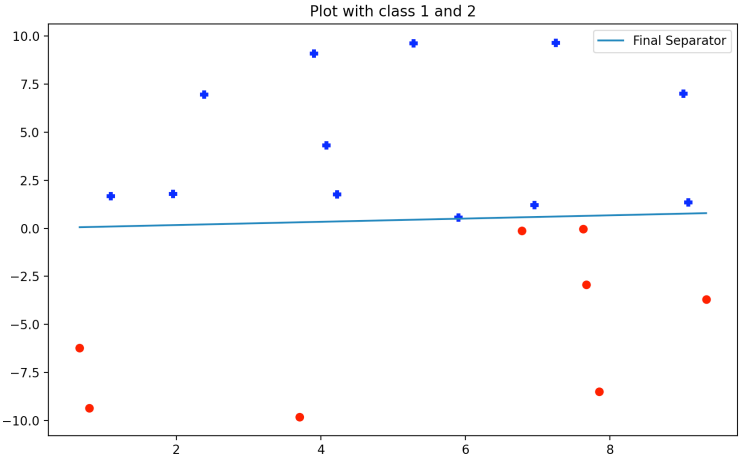

Plot with class 1 as the positive class
Final Separator
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0


Plot with class 2 as the positive class
Final Separator
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

B7. Using the file run_synth_data_1_vs_1.py, capture the plots of all the One-Vs-One classifiers for the synthetic dataset and paste them here. (Let the max number of epochs be 50, and learning rate 0.5)







B8. Using the One-Vs-All classifier, classify the point [6.78, -0.12] as either in class 0, 1, or 2. Briefly explain how you got that class using the individual classifiers. Repeat the same process for One-Vs-One.

One-Vs-All, the point [6,78, -0.12] is classified as -1 in class 0, I can see this because the point is in the red zone.

In One-Vs-One, the point [6,78, -0.12] is classified as -1 in class (1,0), I can see this because the point is in the red zone.