Part B Report

Name: Haolou Sun

Assignment 6: Perceptron Classification and Training

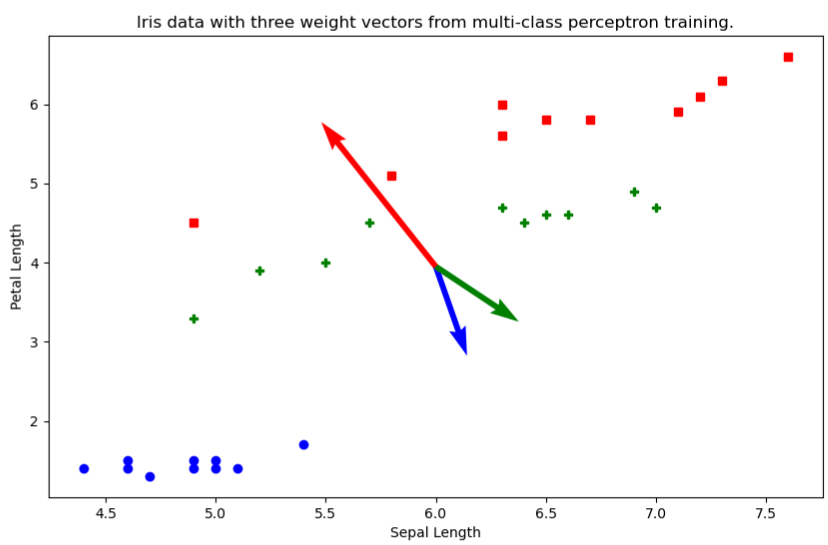
CSE 415 Introduction to Artificial Intelligence, Winter 2023, University of Washington

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.

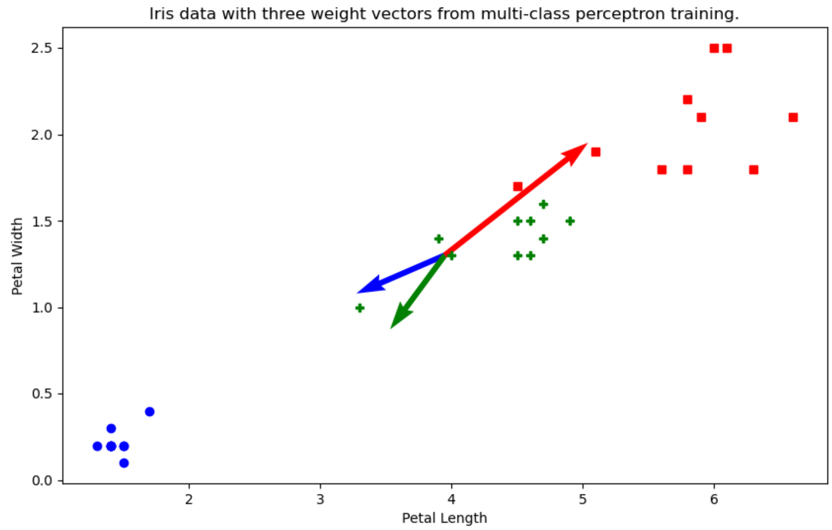
B1. How many epochs were required to train your perceptron on the 3-class Iris data having 4 features (the given training file, with 30 examples)? How many of the test data examples (out of 120) were mis-classified? Determine the percentage error rate and write that here.

85 epochs were required to train. 14 errors on the test data out of 120 items. Thus the error rate is 14/120 = 11.7%

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.



The three points cluster is nearly on a linear line. The blue points are at the bottom left corner, the green points are in the middle, and the red points are at the above right corner. The weight vectors seem to point in the direction where the same color points locate. The relationship between the weight vectors and the distribution of the data is that the data distribution is almost along the weight vector direction line.

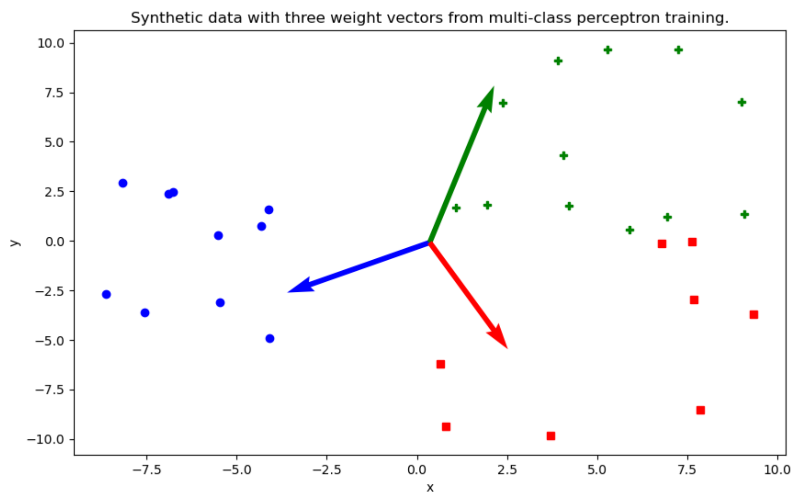
B4. In the file run_3_class_4_feature_iris_data.py, instead of using all zero weight, let

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 ternary perceptron. What kinds of trends do you observe?

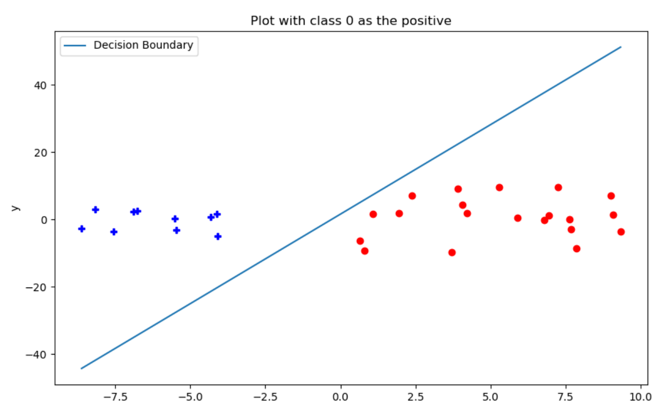
learning rate	epochs	errors
0.001	261	11
0.01	76	12
0.1	57	17
1	50	10
10	73	5
100	85	19
1000	85	14

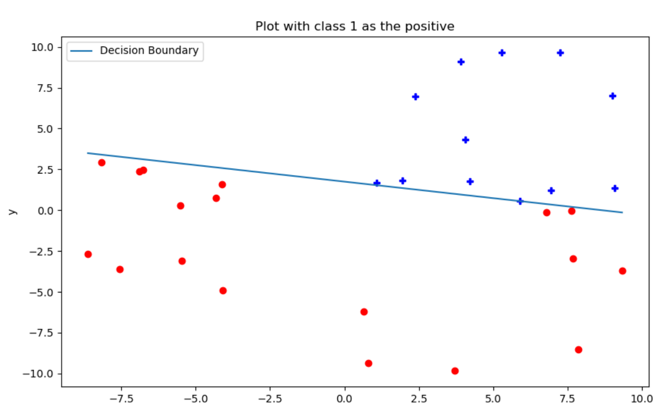
The further the learning rate is away from 1, the more epochs need to converge, while errors don't have a clear trend.

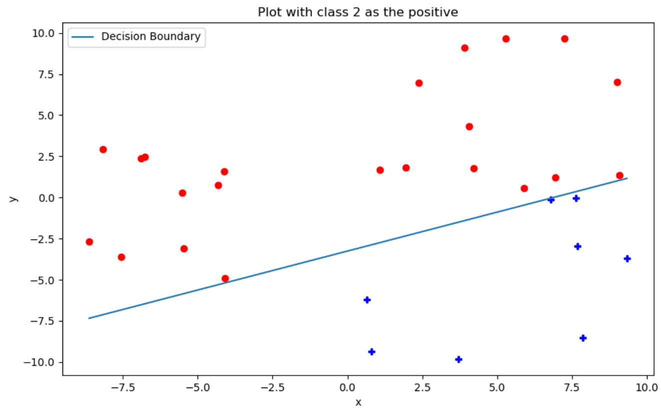
B5. Using the file run_synth_data_ternary.py, capture the plot of the ternary perceptron for the synthetic dataset and paste it here. (Let the maximum number of epochs be 50 and learning rate 0.5, and the weights be all zeros).



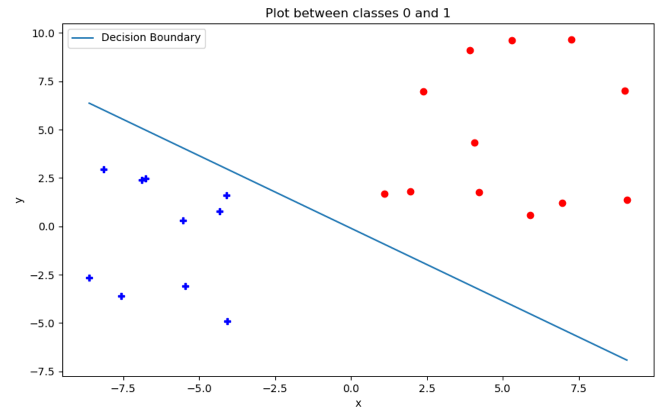
B6. Using the file run_synth_data_1_vs_all.py, capture the plots of all the One-Vs-All classifiers for the synthetic data and paste them here. (Let the maximum number of epochs be 50 and learning rate 0.5).

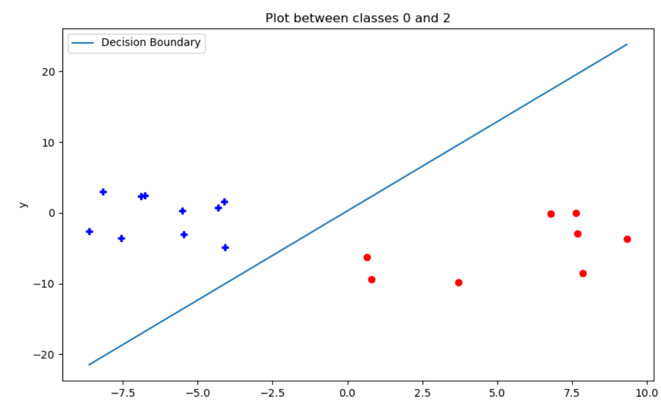


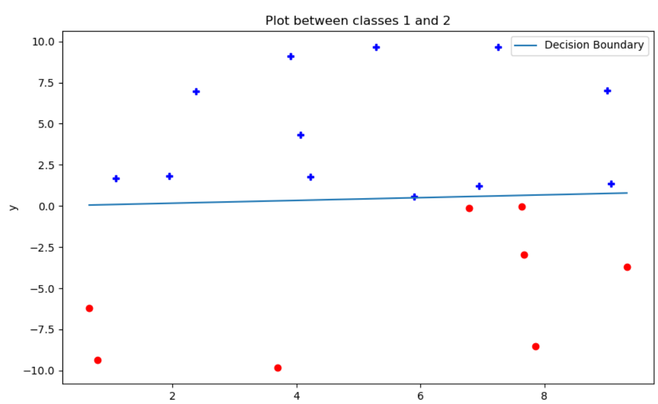




B7. Using the file run_synth_data_1_vs_1.py, capture the plots of all the One-Vs-One classifiers for the synthetic dataset and paste them here. (Let the max number of epochs be 50, and learning rate 0.5)







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.

From the "class 0 as the positive" graph, we can see that the point is classified as either 1 or 2. From the "class 1 as the positive" graph, we can see that the point is classified as either 0 or 2. From the "class 2 as the positive" graph, we can see that the point is classified as 2. Thus, the point is classified as 2 in the One-Vs-All classifier.

From the "between classes 0 and 1" graph, we can see that the point is classified as not 0. From the "between classes 0 and 2" graph, we can see that the point is classified as not 0. From the "between classes 1 and 2" graph, we can see that the point is classified as not 1. Thus, the point is classified as 2 in the One-Vs-One classifier.