Part B Report

Name: Tatsuhiko Araki (student number: 2276178)

Assignment 6: Perceptron Classification and Training

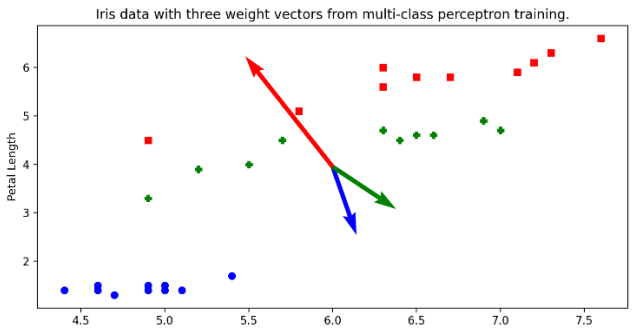
CSE 415 Introduction to Artificial Intelligence, Autumn 2022, University of Washington

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.

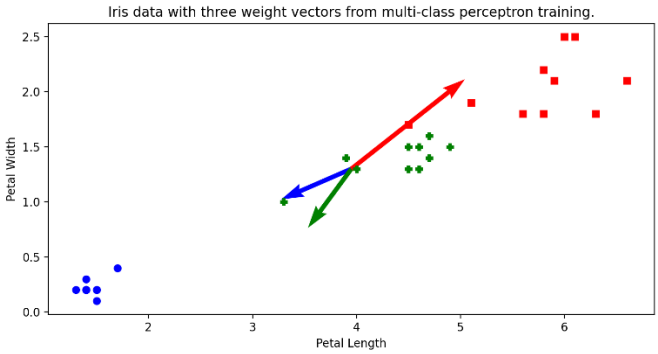
B1. How many epochs were required to train your perceptron on the 3-class Iris data having 4 features (the given training file, with 30 examples)? How many of the test data examples (out of 120) were mis-classified? Determine the percentage error rate and write that here.

85 epochs were required to train my perceptron on the 3-class Iris data having 4 features. 14 test data examples outcc20 were misclassified. In short, error rate is about 11.6 [%].

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 data are plotted along one straight line. The data are distributed above, in the middle, and below the line. The weight vector points to the direction along which each data is distributed. The relationship between the weight vector and data distribution is that the direction indicated by each weight vector is the direction in which each data is distributed. However, the green vector didn't point to the direction where green points are distributed.

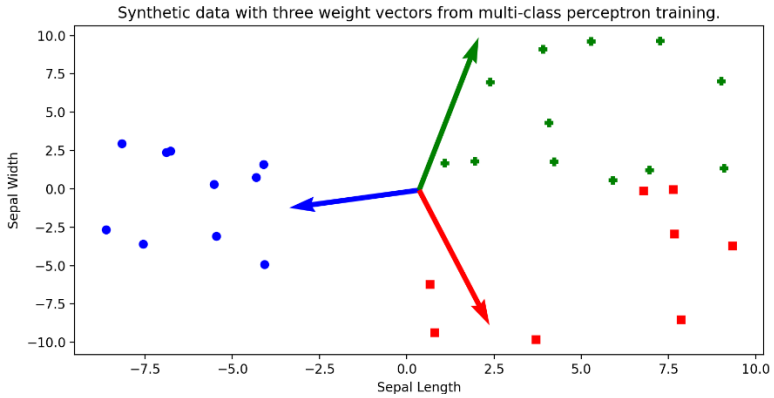
B4. In the file run_3_class_4_feature_iris_data.py, instead of using all zero weight, let

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 to converge	number of 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

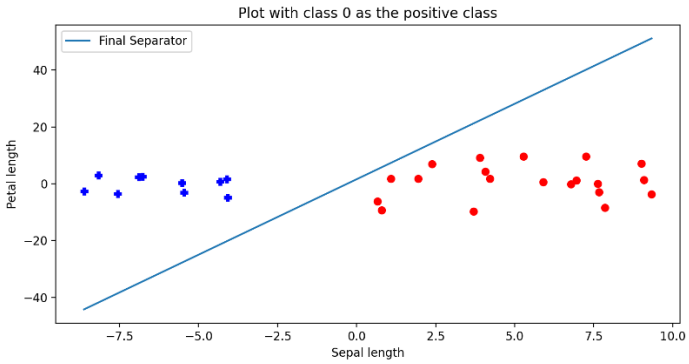
The number of epochs to convergence decreased as the learning rate approached 1. It can be seen that the number of errors decreases as the decimal point of the learning rate becomes smaller. Also, when the learning rate is greater than 1, the number of errors decreases as the learning rate decreases.

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).

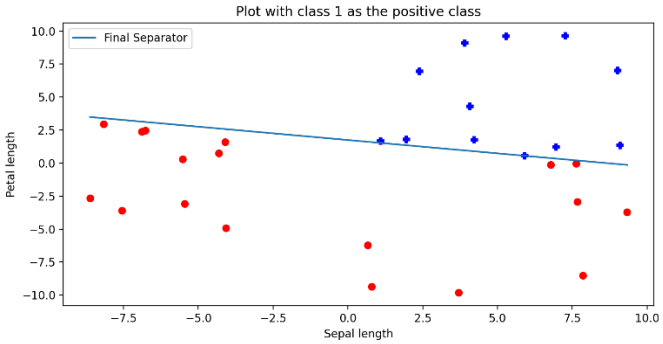


B6. Using the file run_synth_data_1_vs_all.py, capture the plots of all the One-Vs-All classifiers for the synthetic data and paste them here. (Let the maximum number of epochs be 50 and learning rate 0.5).

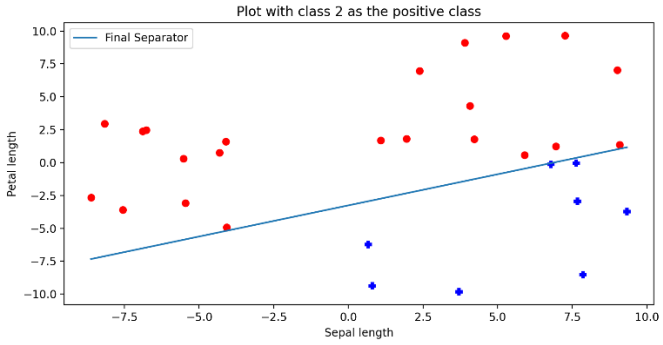
If class 0 is positive and others classes are negative.



If class 1 is positive and others classes are negative.

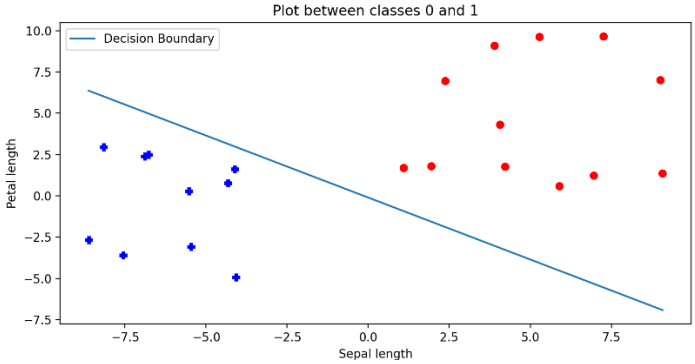


If class 2 is positive and others classes are negative.

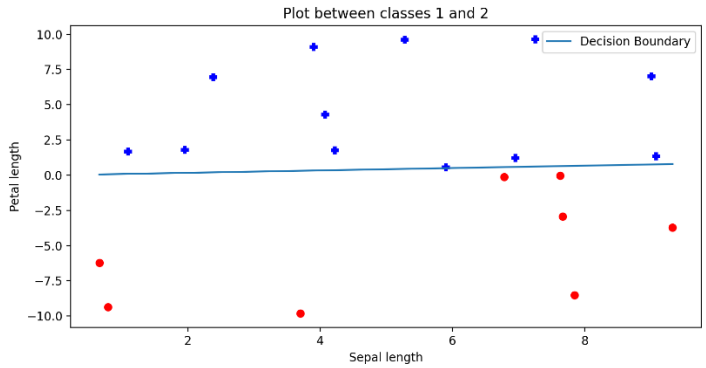


B7. Using the file run_synth_data_1_vs_1.py, capture the plots of all the One-Vs-One classifiers for the synthetic dataset and paste them here. (Let the max number of epochs be 50, and learning rate 0.5)

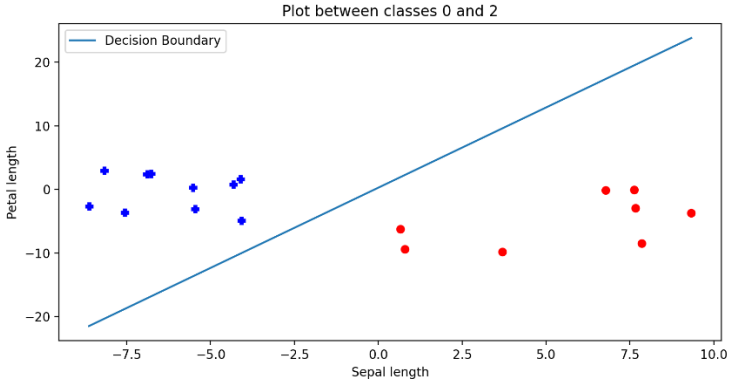
If class 0 and class 1 are plotted.



If class 1 and class 2 are plotted.



If class 0 and class 2 are plotted.



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.

Determine which class this point is assigned to by considering the position of the decision boundary and the position of this point. In One-Vs-All class, if self.POSITIVE=0, the point is classified 1 or 2. if self.POSITIVE=1, the point is classified 0 or 2. if self.POSITIVE=2, the point is classified 2. That is why in One-Vs-All class, the point should be classified 2.

In One-Vs-One class, if the data are plotted between classes 0 and 1, the point is classified to class 1. If the data are plotted between classes 1 and 2, the point is classified to class 2. If the data are plotted between classes 0 and 2, the point is classified to class 2. In One-Vs-One class, class1 have one vote and class 2 have two votes. That is why the point should be classified 2.