COMP6245 : Lab 2 Report

Thanakorn Panyapiang(31446612)

1 Implementing the Perceptron algorithm

Given the data which has the distribution like Figure 1a, using the Perceptron algorithm to train a linear classifier gives a decision boundary as shown in Figure 1b. It can be noticed that the accuracy of the classifier increase over a number of iteration which can be observed from Figure 1c.
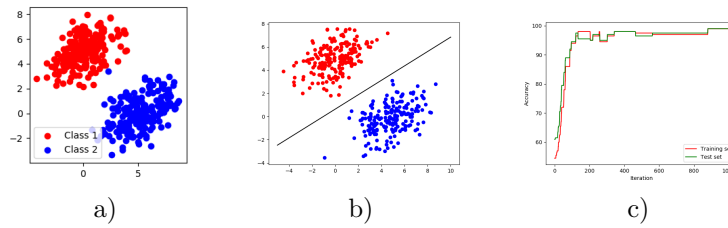

Figure 1

However, the algorithm fails to train a decent linear classifier for the sample which has the distribution like Figure 2a as can be seen from the decision boundary and the accuray in Figure 2b and 2c respectively.
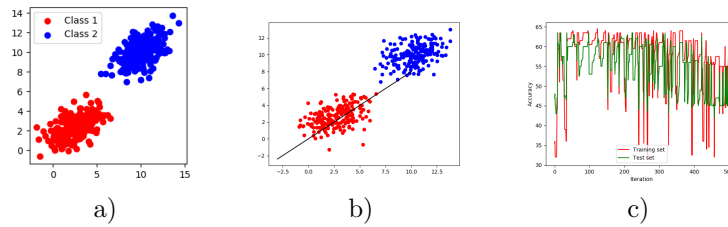

Figure 2

The cause of this issue is the algorithm tries to find the decision boundary that pass the origin, but unlike the first dataset there is no line which passes the origin that can split two classes in this dataset.

To fix this issue, the algorithm need to be modified so that the decision boundary doesn't have to pass the origin. This can be done by adding a bias feature to the dataset which will allow the algorithm to shift the decision boundary. The result after adding a bias feature is shown in Figure 3.
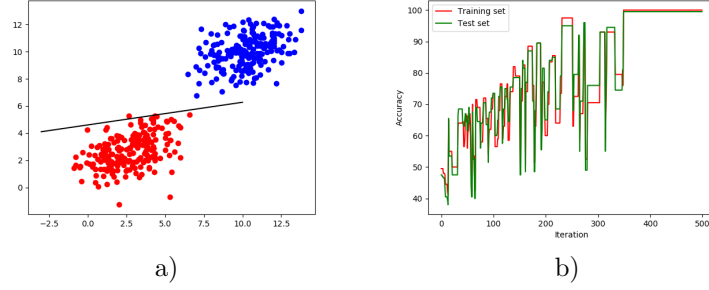
a)   b)

Figure 3

2   Testing with a benchmark dataset

The benchmark dataset I use to test the algorithm is the Iris dataset. From the scatter plot in Figure 4a and 4b, it can be noticed that the data can be seperated using only two attributes.
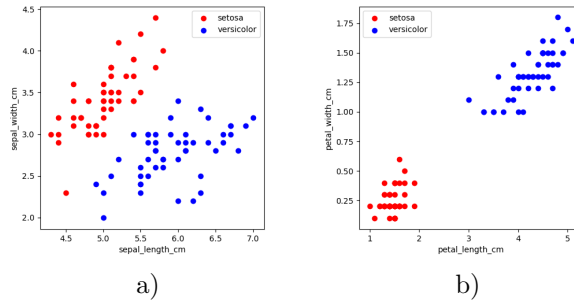

a)   b)

Figure 4

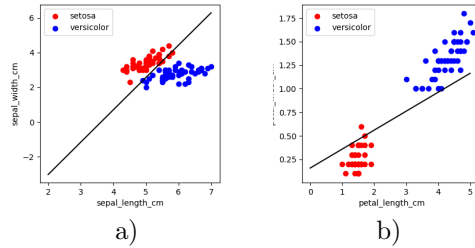Training a linear classifier for Iris dataset using the Perceptron algorithm gives the result as illustrated in Figure 5.

a)   b)

Figure 5