Problem 4:

The perceptron learning algorithm works like this: In each iteration t, pick a random x(t),y(t) and compute the signal s(t) = wT(t)x(t). If y(t).s(t)≤0,update w by

w(t+ 1)←w(t) +y(t).x(t);

One may argue that this algorithm does not take the 'closeness' between s(t) and y(t) into consideration. Let's look at another perceptron learning algorithm: In each iteration, pick a random (x(t),y(t)) and compute s(t). If y(t).s(t)≤1, update w by

w(t+ 1)←w(t) +η.(y(t)−s(t)).x(t);

Where η is a constant. That is, if s(t) agrees with y(t) well (their product is > 1), the algorithm does nothing. On the other hand, if s(t) is further from y(t), the algorithm changes w(t) more. In this problem, you are asked to implement this algorithm and study its performance.

(a) Genrate a training data set of size 100 similar to that used in Exercise1.4. Generate a test data set of size 10,000 from the same process. To get g, run the algorithm above with η= 100 on the training data set,until a maximum of 1,000 updates has been reached. Plot the training data set, the target function f, and the final hypothesis g on the same figure. Report the error on the test set.
(b) Use the data set in (a) and redo everything with η= 1.
(c) Use the data set in (a) and redo everything with η= 0.01.
(d) Use the data set in (a) and redo everything with η= 0.0001.
(e) Compare the results that you get from (a) to (d).

The algorithm above is a variant of the so-called Adaline (Adaptive LinearNeuron) algorithm
for perceptron learning.

In the following Figures the red points are the test data that were miss classified and the green were properly classified.

(a)

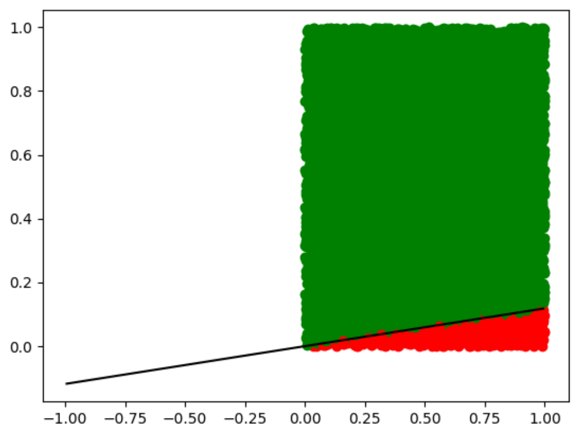

Figure 1: η = 100

(b)

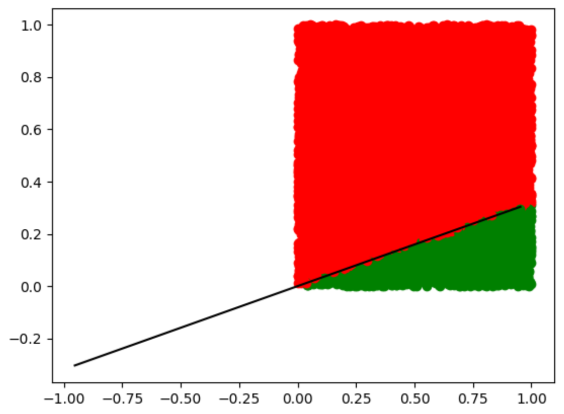

Figure 2 : η = 1

(c)

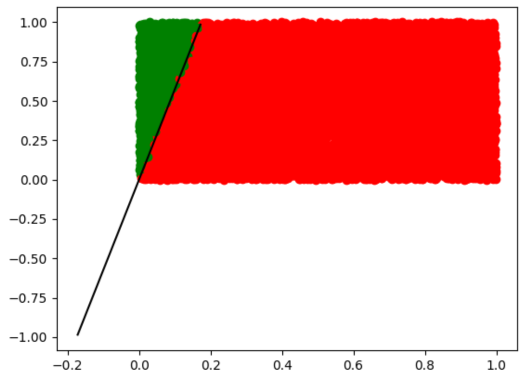

Figure 3 : η = 0.01

(d)

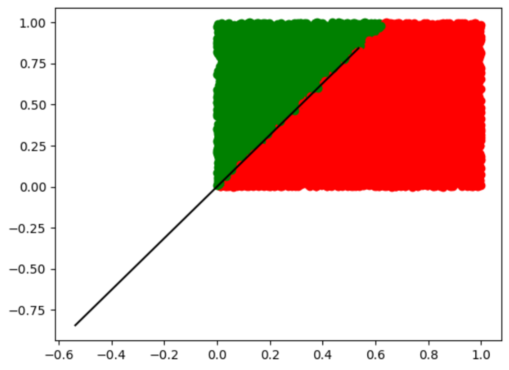

Figure 2 : η = 0.0001

(e) It seems as though as **η** decreased, the number of points that were properly classified decreased. The lower **η** was, the slower the weight was adjusted. This must mean that the lower **η** values couldn't correct the hypothesis fast enough within the 1000 iterations to properly classify all points.