HOMEWORK 2

I implemented the perceptron training algorithm in the .py file and obtained the following results/observations by keeping the seed constant.

(a)
w0 was uniformly at random from [-¼,¼]
w0 = -0.041489

(b)
w1 was uniformly at random from [-1,1]
w1 = 0.44064899

(c)
w2 was uniformly at random from [-1,1]
w2 = -0.99977125

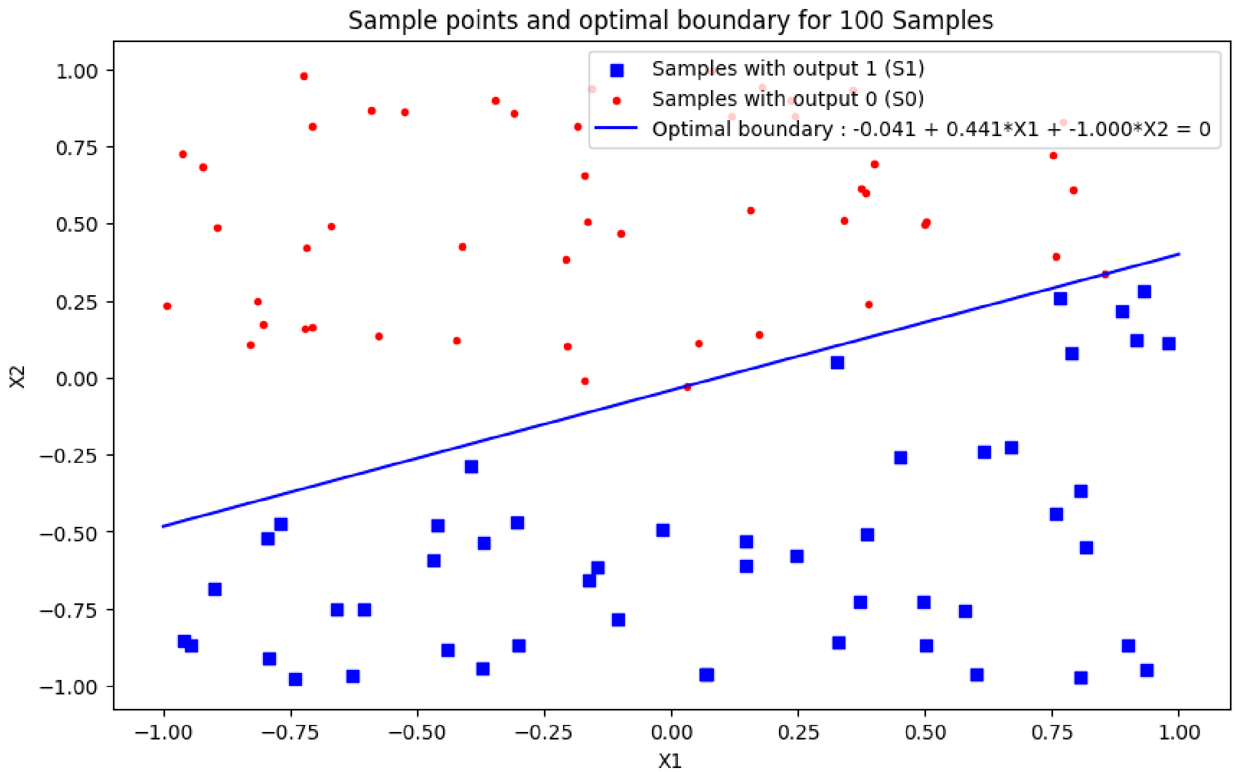
So w = [w0, w1, w2] = [-0.041489, 0.44064899, -0.99977125]

(d)
Initialized S with 100 vectors which are independently and uniformly at random on $[-1, 1]^2$

(e)
By applying $[1\ x1\ x2][w0\ w1,\ w2]^T$ on S, S0 and S1 was obtained. Here S1 is all samples that satisfy $[1\ x1\ x2][w0\ w1\ w2]^T\ >=\ 0$ condition.

(f)
And S0 is all samples that satisfy $[1\ x1\ x2][w0\ w1\ w2]^T\ <\ 0$ condition

(g)
Please find below figure which has plotted all the samples (S1 and S0 separated) and optimal boundary $(w0\ +\ w1\ *\ x1\ +\ w2\ *\ x2\ =\ 0)$. Please note that in the below graph the weights are rounded off to 3 decimal places.



(h)

i. Implementing perceptron training algorithm with **η** (eta) = 1 on S

ii. Getting w′ from uniformly random distribution, here w′ obtained was [0.28313242, -0.21998457, -0.02801867]

iii. By using this w′, no of misclassifications obtained were 50

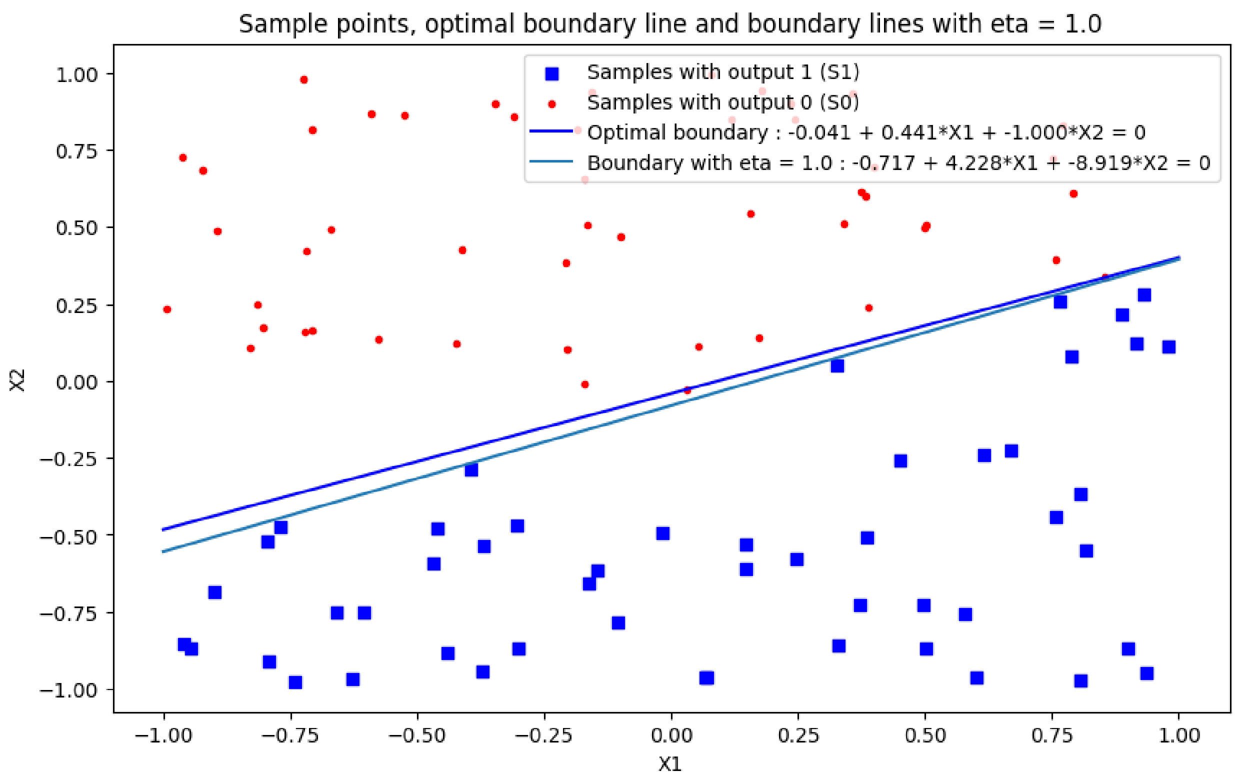
iv. Found the new set of weights w″ by using perceptron training algorithm

v. The number of misclassifications with this new set of weights w″ were 6

vi. Did another epoch of the perceptron training algorithm, found a new set of weights, recorded the number of misclassifications, and so on, and at last it converged. Number of epochs required to converge was 39. For each epoch no of misclassifications were [50, 6, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 4, 4, 4, 3, 3, 3, 3, 2, 5, 5, 4, 4, 4, 4, 4, 3, 3, 3, 3, 3, 3, 3, 2, 0]. Please note that the 0th element of the misclassifications array is obtained by calculating misclassification for w′ which is 0th epoch. So we can see here that at the end of the 39th epoch the misclassification converged to 0.

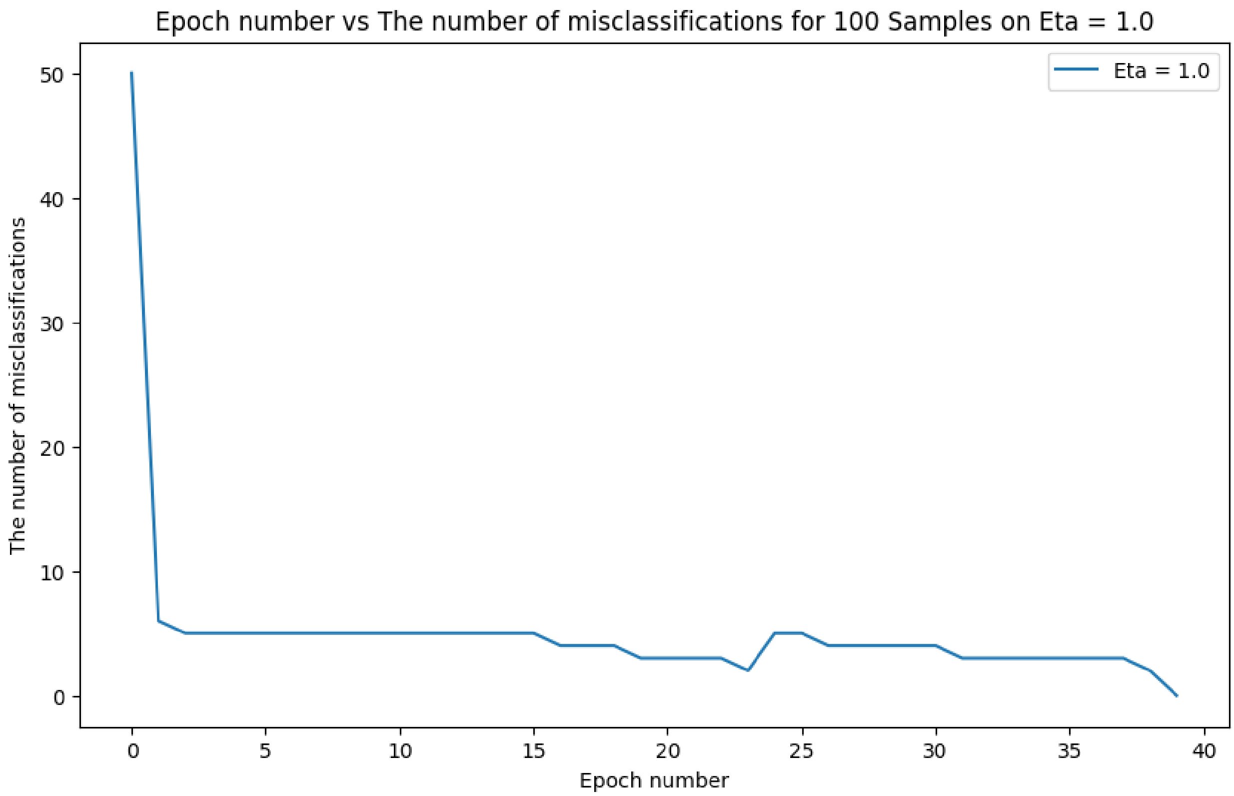
vii. The final (predicted) weights obtained were [-0.71686758, 4.22766547, -8.91902774].

When we compare these predicted weights with optimal weights then we can see that - Both optimal and predicted w0 values were negative (-ve), optimal and predicted w1 values were positive (+ve) and optimal and predicted w2 values were negative (-ve). In the below graph we can see that both predicted and optimal weights can be used to boundary S0 and S1 points. Please note that in the below graph the weights are rounded off to 3 decimal places.



(i)

Please find below a graph that shows the epoch number vs the number of misclassifications for 100 samples on **η** (eta) = 1. We can see that misclassifications converged (was 0) at epoch number 39.



(j)

i. Implementing perceptron training algorithm with **η** (eta) = 10 on S, by using same w0, w1, w2, S, w′0, w′1, w′2

ii. w′ was [0.28313242, -0.21998457, -0.02801867] , same as in **η** (eta) = 1

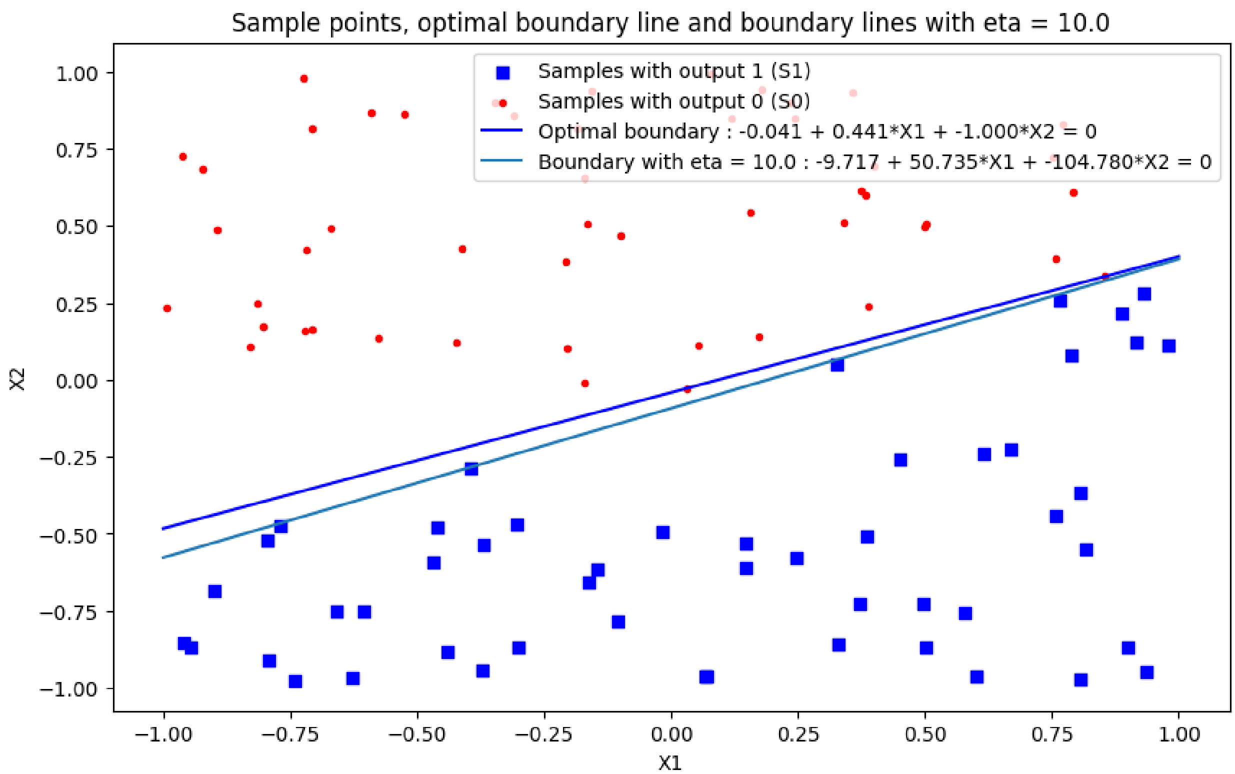
iii. By using this w′, no of misclassifications obtained were 50

iv. Found the new set of weights w″ by using perceptron training algorithm

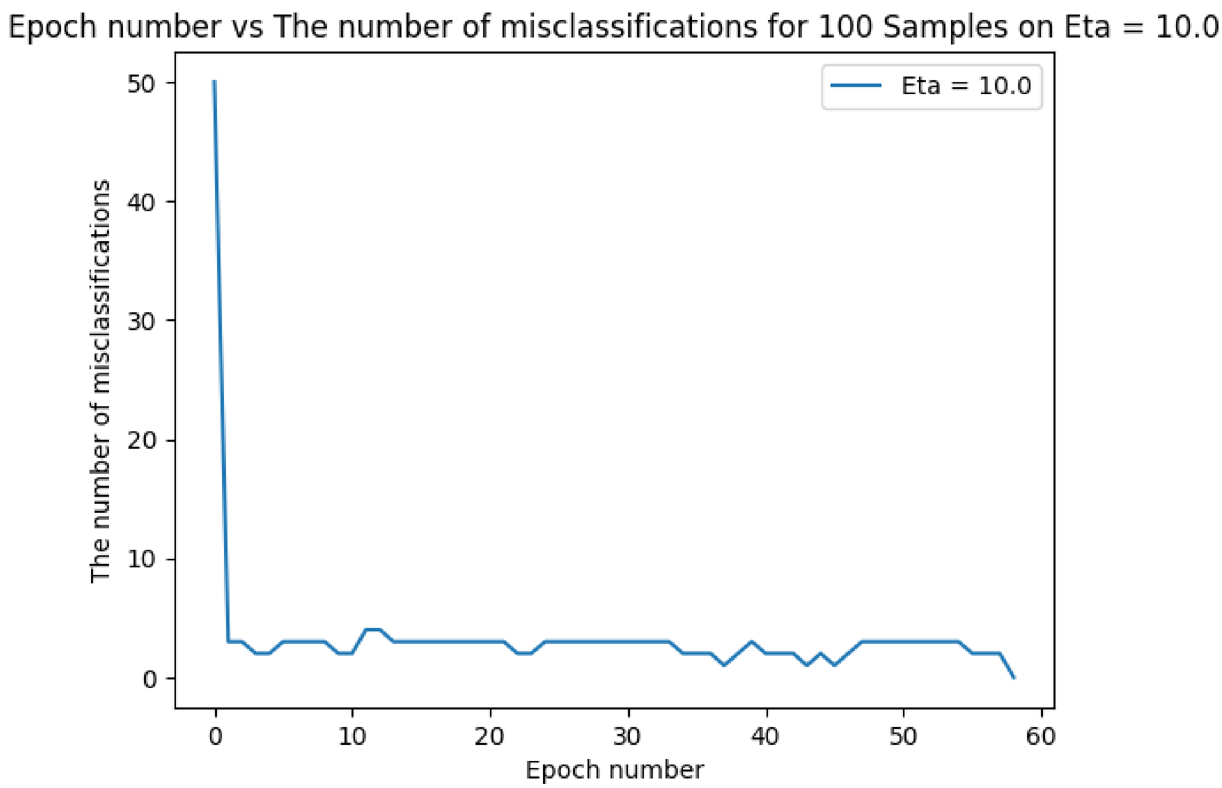
v. The number of misclassifications with this new set of weights w″ were 3

vi. Did another epoch of the perceptron training algorithm, found a new set of weights, recorded the number of misclassifications, and so on, and at last it converged. Number of epochs required to converge was 58. For each epoch no of misclassifications were [50, 3, 3, 2, 2, 3, 3, 3, 3, 2, 2, 4, 4, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 2, 2, 1, 2, 3, 2, 2, 2, 1, 2, 1, 2, 3, 3, 3, 3, 3, 3, 3, 3, 2, 2, 2, 0]. Please note that the 0th element of the misclassifications array is obtained by calculating misclassification for w′ which is 0th epoch. So we can see here that at the end of the 58th epoch the misclassification converged to 0.

vii. The final (predicted) weights obtained were [-9.71686758, 50.73480449, -104.77963].
When we compare these predicted weights with optimal weights then we can see that - Both optimal and predicted w0 values were negative (-ve), optimal and predicted w1 values were positive (+ve) and optimal and predicted w2 values were negative (-ve). In the below graph we can see that both predicted and optimal weights can be used to boundary S0 and S1 points. Please note that in the below graph the weights are rounded off to 3 decimal places.



viii. Please find below a graph that shows the epoch number vs the number of misclassifications for 100 samples on **η** (eta) = 10. We can see that misclassifications converged (was 0) at epoch number 58.



(k)

i. Implementing perceptron training algorithm with **η** (eta) = 0.1 on S, by using same w0, w1, w2, S, w′0, w′1, w′2

ii. w′ was [0.28313242, -0.21998457, -0.02801867], same as in **η** (eta) = 1

iii. By using this w′, no of misclassifications obtained were 50

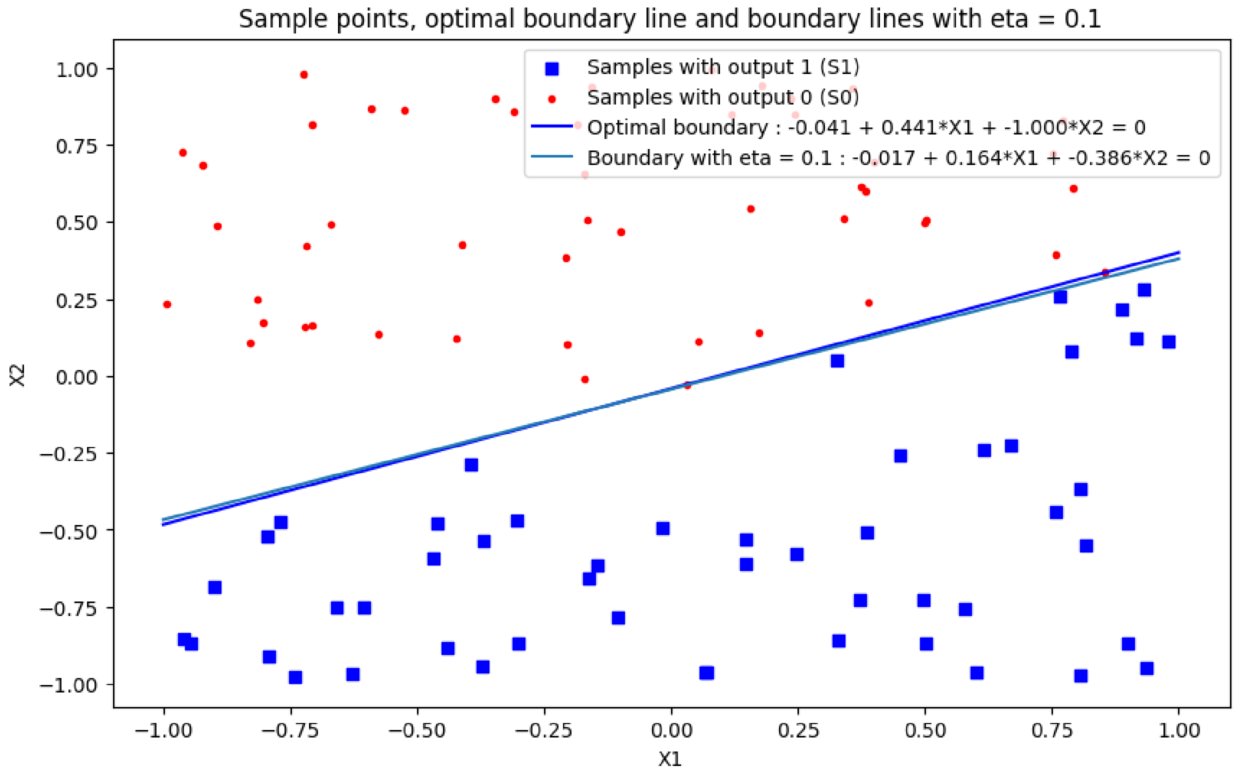
iv. Found the new set of weights w″ by using perceptron training algorithm

v. The number of misclassifications with this new set of weights w″ were 4

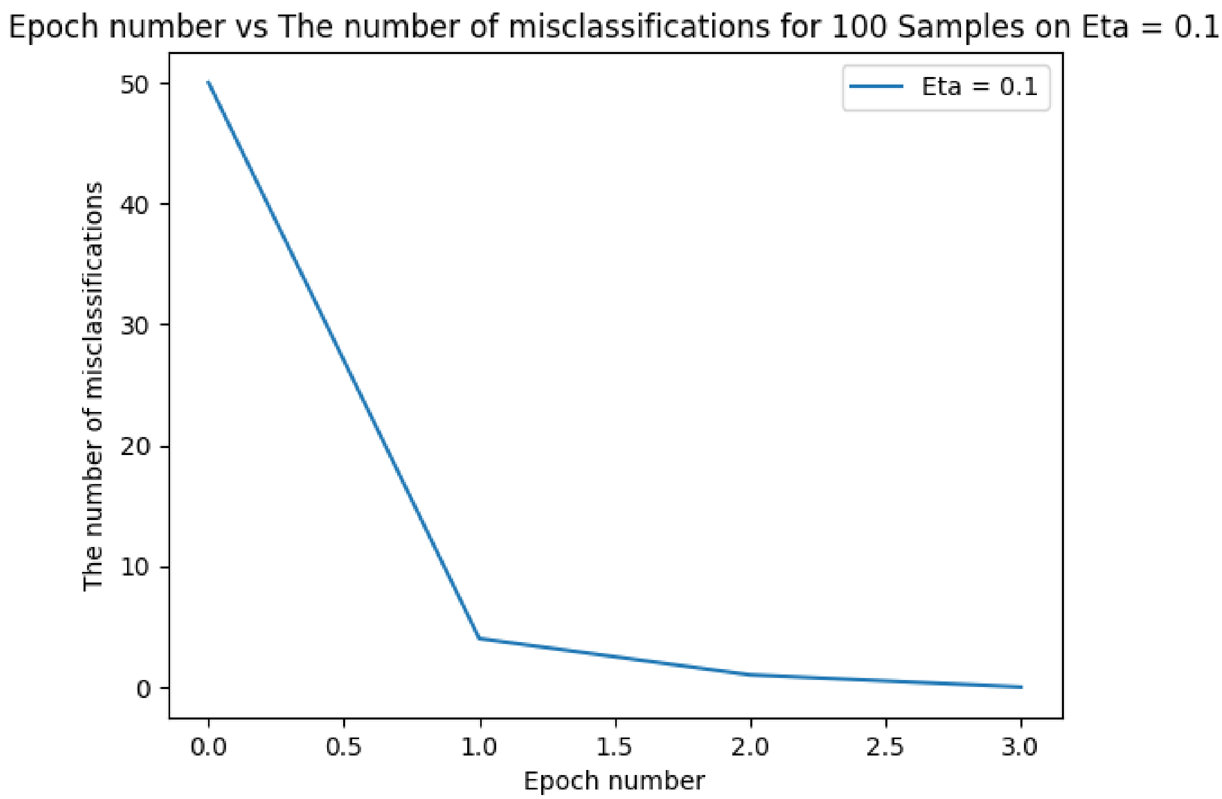
vi. Did another epoch of the perceptron training algorithm, found a new set of weights, recorded the number of misclassifications, and so on, and at last it converged. Number of epochs required to converge was 3. For each epoch no of misclassifications were [50, 4, 1, 0]. Please note that the 0th element of the misclassifications array is obtained by calculating misclassification for w′ which is 0th epoch. So we can see here that at the end of the 3rd epoch the misclassification converged to 0.

vii. The final (predicted) weights obtained were [-0.01686758 0.1635715 -0.38646323].

When we compare these predicted weights with optimal weights then we can see that - Both optimal and predicted w0 values were negative (-ve), optimal and predicted w1 values were positive (+ve) and optimal and predicted w2 values were negative (-ve). In the below graph we can see that both predicted and optimal weights can be used to boundary S0 and S1 points. Please note that in the below graph the weights are rounded off to 3 decimal places.



viii. Please find below a graph that shows the epoch number vs the number of misclassifications for 100 samples on **η** (eta) = 0.1. We can see that misclassifications converged (was 0) at epoch number 3.



(I)
As per the above information, please find below table to see no. of epochs required to converge for each **η** (eta)

η (eta)	No. of epochs required to converge
0.1	3
1	39
10	58

By increasing the **η** (eta) from 0.1 to 10, the no. of epochs required to converge is also increasing. I tried by **η** (eta) as 0.01, the no. of epochs required to converge was 7. So, in this case **η** (eta) = 0.1 is an optimal **η** (eta) which requires minimum epochs to converge. So by increasing or decreasing **η** (eta) from 0.1 we will get the more no. of epochs required to converge than at **η** (eta) = 0.1.

For every random set S and weight w, we can get an optimal **η** (eta) which requires minimum epochs to converge. So this **η** (eta) is global minimum of the graph **η** (eta) vs No. of epochs required to converge

(m)

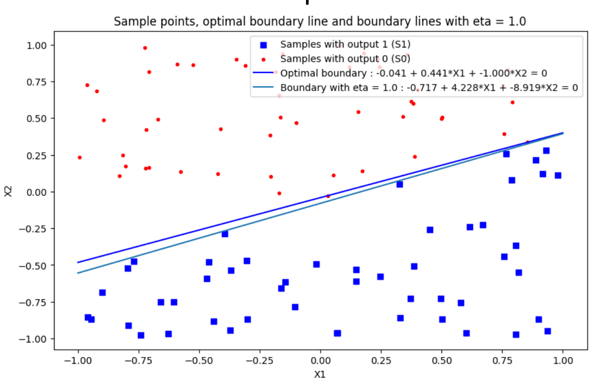
If we start with different w0, w1, w2, S, w′ 0 , w′ 1 , w′ 2, we won't get an optimal **η** (eta) at the same location. But there will exist one **η** (eta) which will require minimum epochs to converge. So this **η** (eta) is called global minimum of the graph **η** (eta) vs No. of epochs required to converge. Whereas the location of the global minimum would change by changing any value in w0, w1, w2, S, w′ 0 , w′ 1 , w′ 2.

(n)

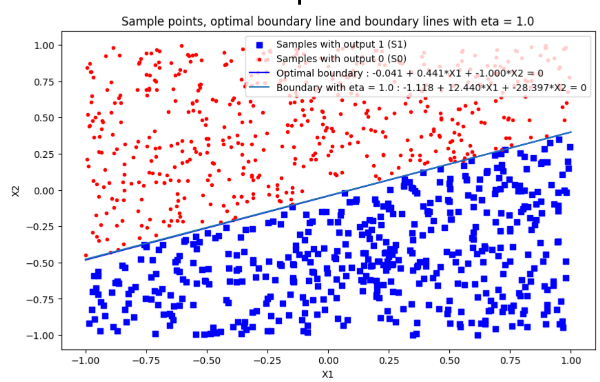
By taking more no. of samples, the boundary created by predicted weights is getting closer to the boundary created by the actual weights. For all **η** (eta) we will see that - for 1000 samples predicted weights boundary is getting closer to the boundary created by the actual weights compared to 100 samples.

Please find the below comparison for **η** (eta) = 1

No. of Sample = 100

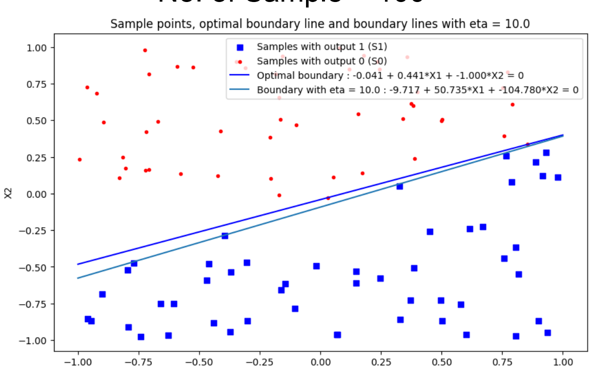


No. of Samples = 1000

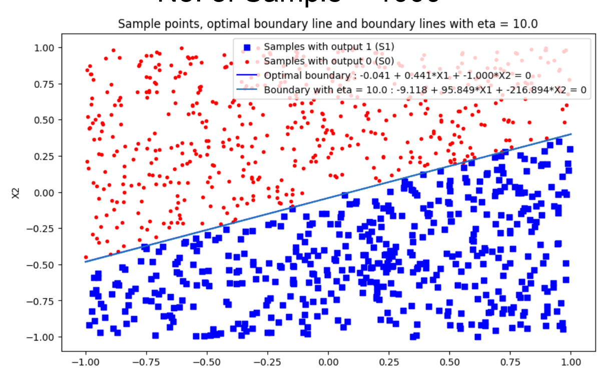


Please find the below comparison for **η** (eta) = 10

No. of Sample = 100

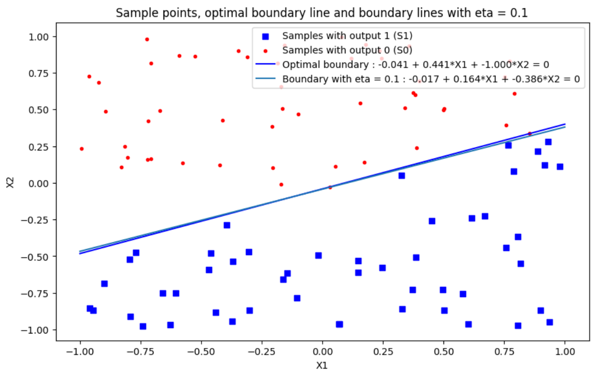


No. of Sample = 1000

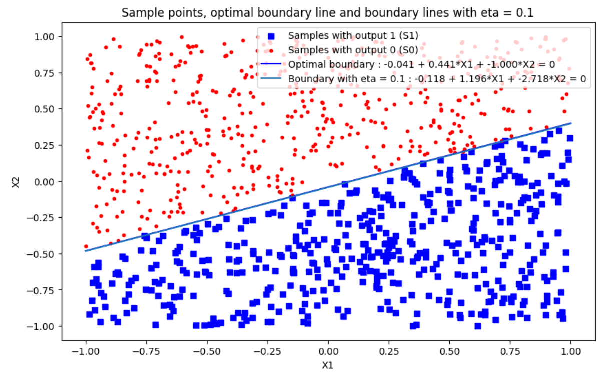


Please find the below comparison for **η** (eta) = 0.1

No. of Sample = 100

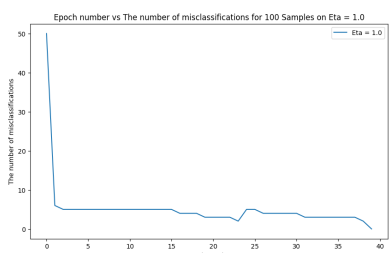


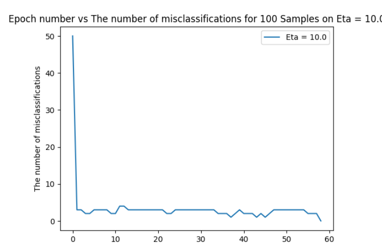
No. of Sample = 1000

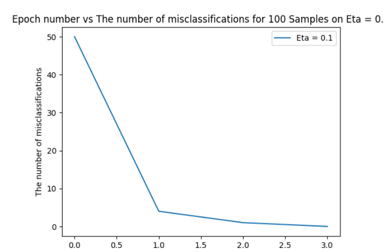


Taking **η** (eta) = 0.1,1,10
For no. of samples = 100 the minimum epochs required to converge is obtained at **η** (eta) = 0.1. Whereas for no. of samples = 1000 the minimum epochs required to converge is obtained at **η** (eta) = 10. So, even by changing the number of samples, the **η** (eta) which requires minimum epochs to converge also changes. Please see the graph below for reference.

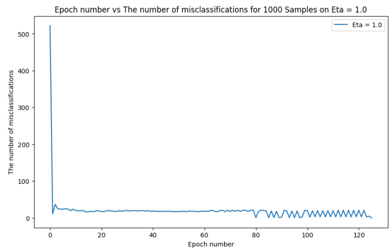
Epoch number vs The number of misclassifications for 100 Samples : minimum epoch is for **η** (eta) = 0.1

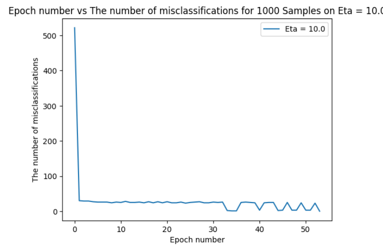


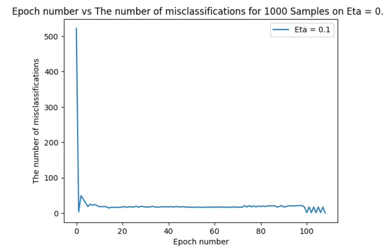




Epoch number vs The number of misclassifications for 1000 Samples : minimum epoch is for **η** (eta) = 10







And from the above graph, we can see that generally the no of epochs required to converge will increase if the no. of samples are increased.