EEE 443/543 - Project #2

a-c) The code picks the weights uniformly at random on the given intervals:

w_0 is chosen from [-1/4, 1/4]
w_1 is chosen from [-1, 1]
w_2 is chosen from [-1, 1]

```
Optimal Weights w0: 0.022925714967489297, w1: 0.717713221182718, w2: 0.3717851731331583
```

In order to obtain consistent results each time the code runs the random seed has been set to 67.

```
np.random.seed(67)
```

d-f) n = 100 vectors has been picked by the code independently and uniformly at random on $[-1,1]^2$. After that, the set vectors $\boldsymbol{S}$ has been divided into two subsets $\boldsymbol{S_0}$ and $\boldsymbol{S_1}$ satisfying the following:

$\boldsymbol{S_1} \subset$ S: x = [x_1 x_2] $\in \boldsymbol{S}$ such that [1 x_1 x_2][w_0 w_1, w_2]T ≥ 0
$\boldsymbol{S_0} \subset$ S: x = [x_1 x_2] $\in \boldsymbol{S}$ such that [1 x_1 x_2][w_0 w_1, w_2]T < 0

The result is plotted as shown in the *Fig.1* with the line $w_0 + w_1x_1 + w_2x_2 = 0$ and all the points in $\boldsymbol{S_1}$ and $\boldsymbol{S_0}$ are indicated with different symbols.

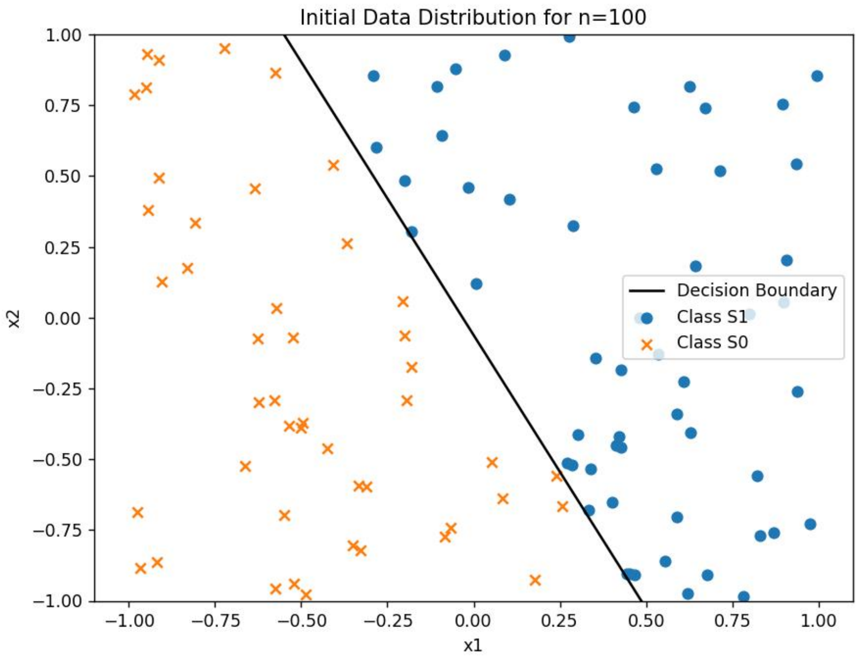

Fig.1 Data distribution along with the decision boundary n = 100

h, i)

```
Optimal Weights w0: 0.022925714967489297, w1: 0.717713221182718, w2: 0.3717851731331583
Initial Weights w0'=-0.3368163579731316, w1'=-0.8800041408787642, w2'=-0.22744443791458213
```

The learning rate has been chosen as $\eta = 1$.

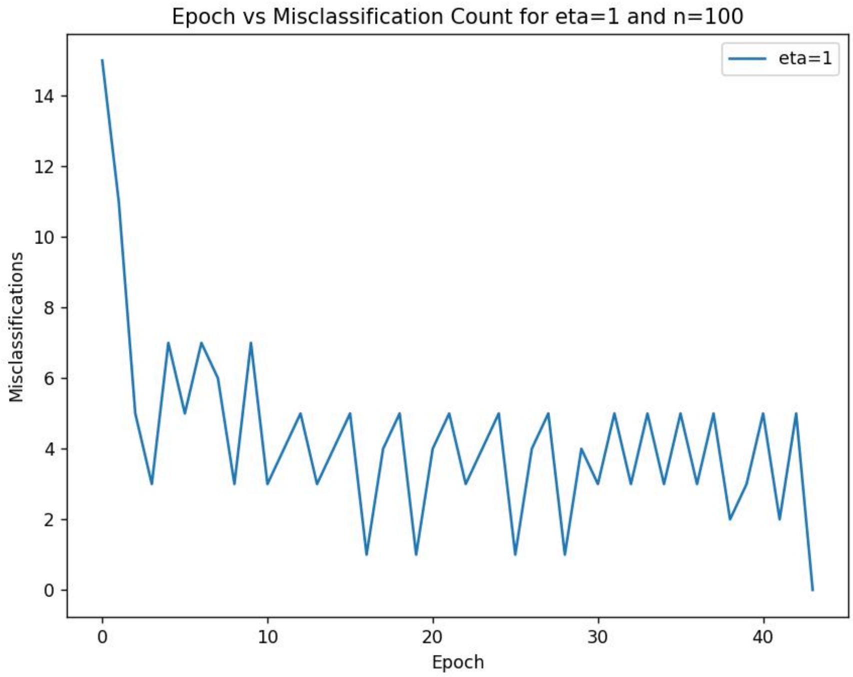

Fig.2 Epoch vs Misclassification Count for eta=1 and n=100

j)

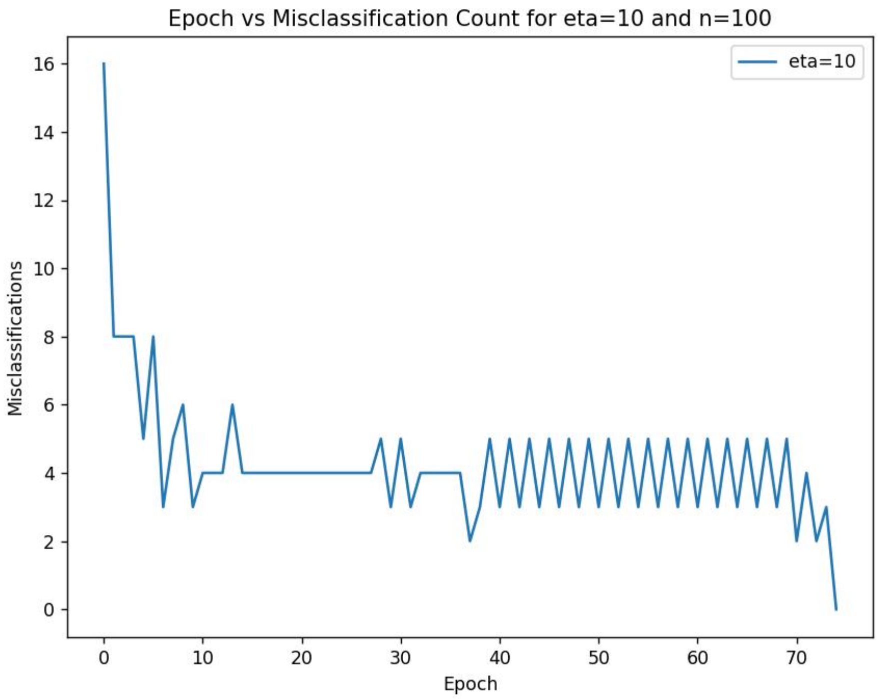

Fig.3 Epoch vs Misclassification Count for eta=10 and n=100

k)

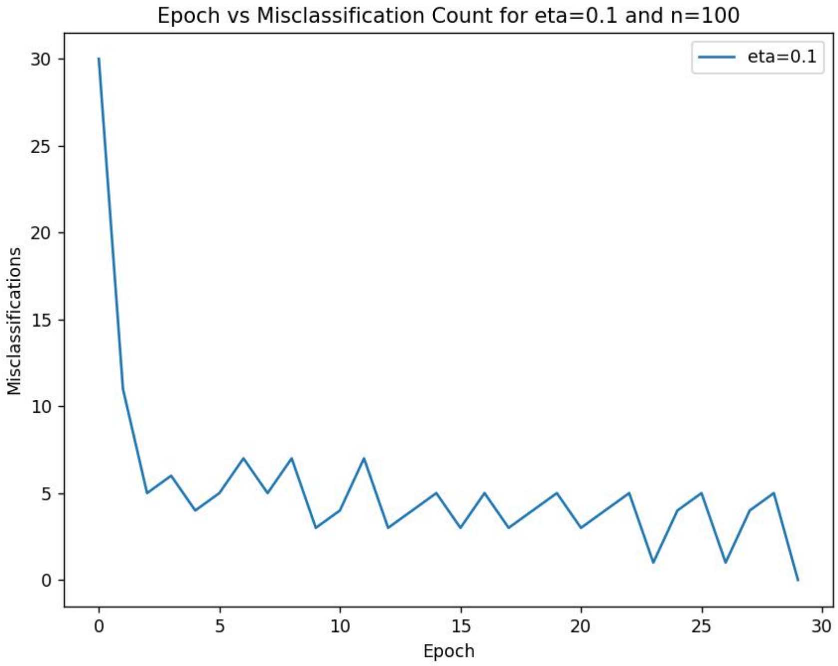

Fig.4 Epoch vs Misclassification Count for eta=0.1 and n=100

l)

Even though, the final weights are different from the optimal weights for each learning rate, the algorithm has converged to 0 misclassifications after certain number of epochs. When the learning rate was high (eta = 10) the number of epochs were larger which means when the learning rate was larger the learning algorithm was more unstable, resulting oscillations. Smaller learning rate (eta = 0.1) resulted in smaller number epochs.

```
n = 100
Final Weights eta=1 w0=0.6631836420268684, w1=9.905631068204144, w2=5.480769095115705
Final Weights eta=10 w0=9.663183642026869, w1=138.17141563034716, w2=78.13087003231081
Final Weights eta=0.1 w0=0.06318364202686844, w1=0.8520231359607596, w2=0.484023881876069
```

m) Different starting weights have resulted in totally different results. This is tested with changing the random seed and observing the results. The resulting number of epochs and final weights were depends on the initial weight assignments. In other words, with different optimal weight values the decision boundary was totally different, therefore the results were totally different. Different ***S*** also resulted in different final results, because it effects the number of misclassifications and therefore number of epochs. Also different ***S*** may result in a slightly different decision boundaries therefore different weights.

n) n = 1000 samples.

```
Optimal Weights w0: 0.022925714967489297, w1: 0.717713221182718, w2: 0.3717851731331583
Initial Weights w0'=-0.3368163579731316, w1'=-0.8800041408787642, w2'=-0.22744443791458213
```

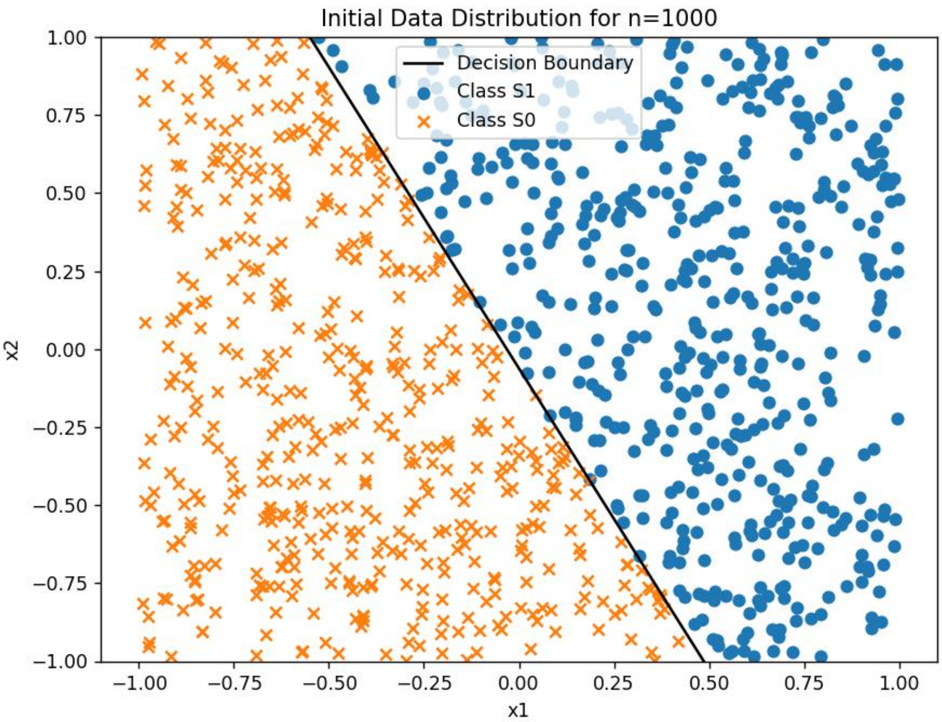

Fig.5 Data distribution along with the decision boundary n = 1000

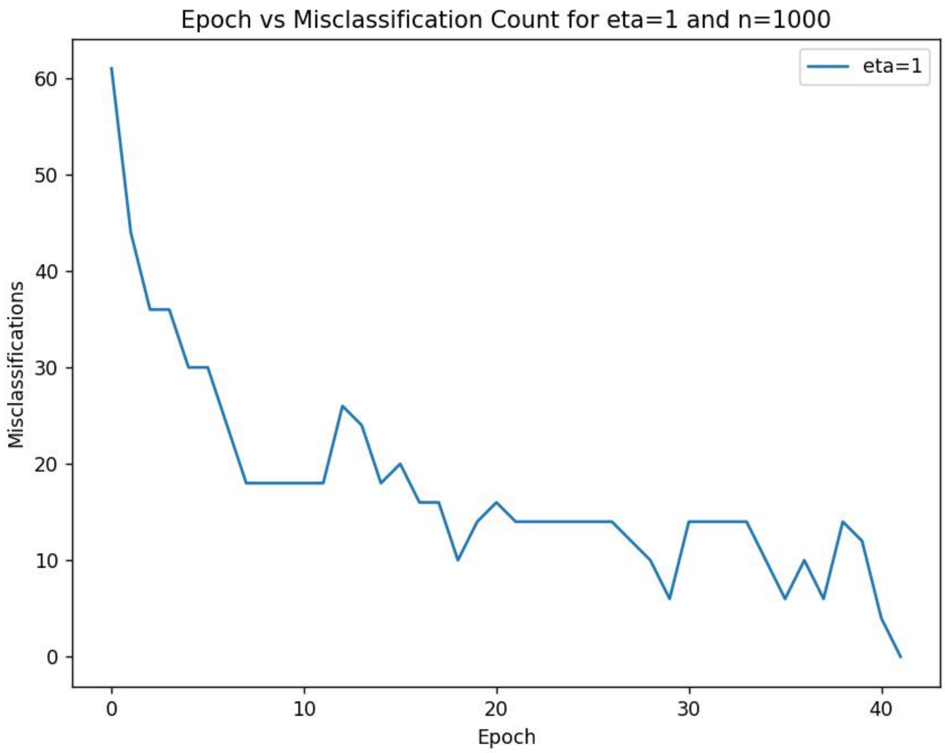

Fig.6 Epoch vs Misclassification Count for eta=1 and n=1000

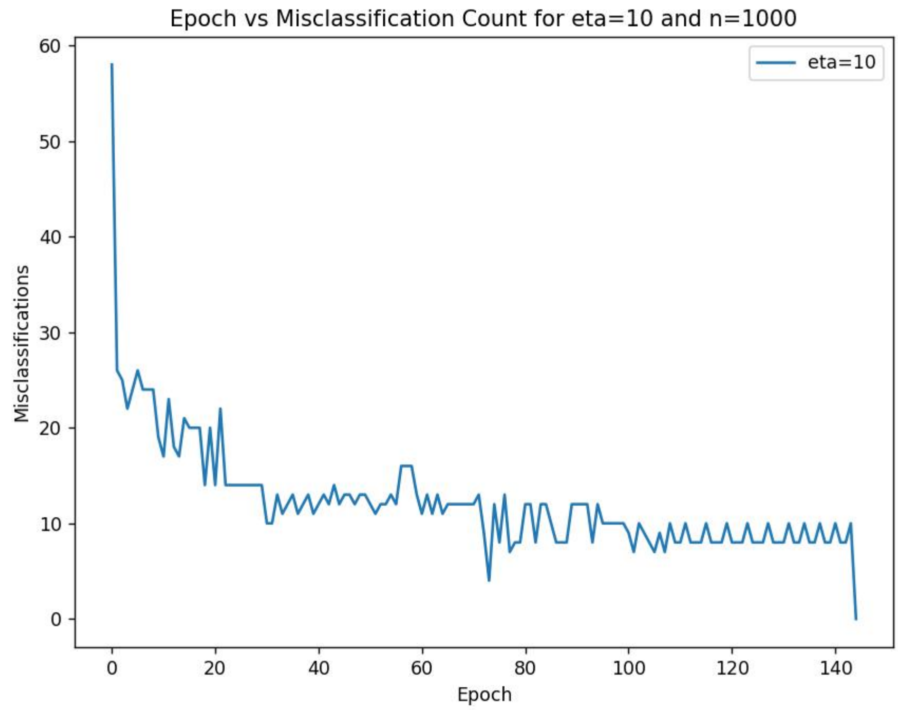

Fig.7 Epoch vs Misclassification Count for eta=10 and n=1000

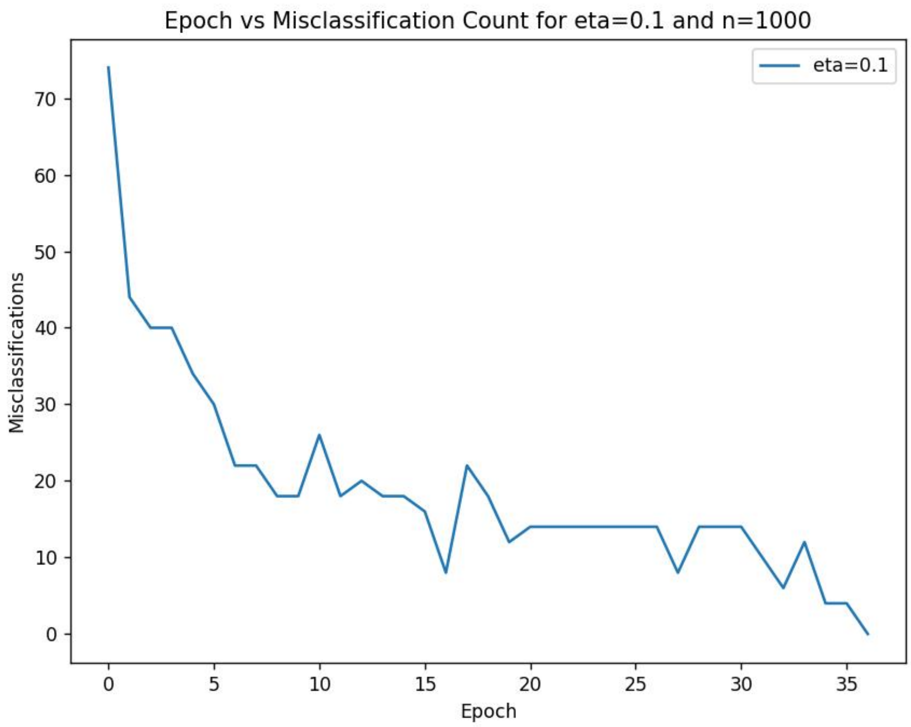

Fig.8 Epoch vs Misclassification Count for eta=0.1 and n=1000

```
n = 1000
Final Weights eta=1 w0=0.6631836420268684, w1=21.19683517430739, w2=10.839504436282704
Final Weights eta=10 w0=9.663183642026869, w1=323.73659421352136, w2=164.98167202999704
Final Weights eta=0.1 w0=0.06318364202686844, w1=2.045207155660015, w2=1.0477144339736144
```

Increasing the sample size from n = 100 to n = 1000 generally increased the magnitude of the final weights for higher learning rates. Other than that the results were similar, such that larger eta resulted in larger number of epochs and the algorithm converged to 0 misclassifications. When number of samples were larger (n = 1000) the learning algorithm is prone to overfitting and outliers.