# CS 559 – Neural Networks | Homework 2

**Zohair Hashmi**

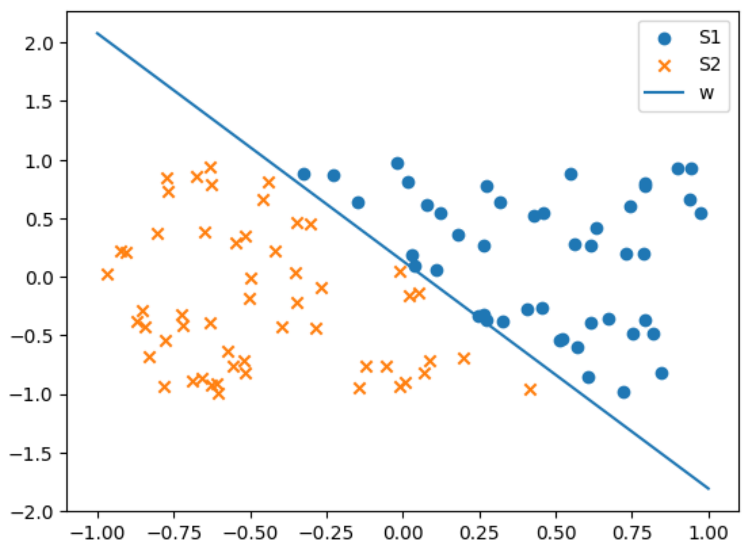
**UIN# 668913771**

**Date: September 19th, 2023**

## Part (a-g)

- $w0_{optimal} = -0.06272994057631875$
- $w1_{optimal} = 0.9014286128198323$
- $w2_{optimal} = 0.4639878836228102$

- Class 1: Blue dots
- Class 2: Orange crosses
- Separation boundary (e.g. $w_0 + w_1x_1 + w_2x_2 = 0$)




## Part (h)

| Epoch | w0 | w1 | w2 | Misclassified Points |
|---|---|---|---|---|
| 0 | 0.176261602 | 0.795427456 | 0.783061459 | 16 |
| 1 | -0.823738398 | 1.097008307 | 0.331150101 | 35 |
| 2 | 0.176261602 | 1.656759398 | 0.615213394 | 5 |
| 3 | -0.823738398 | 1.635264793 | 0.780391387 | 26 |
| 4 | 0.176261602 | 1.901467708 | 0.458450969 | 7 |
| 5 | 1.176261602 | 1.576698051 | 1.344270377 | 30 |
| . | . | . | . | . |

| . | . | . | . | . |
|---|---|---|---|---|
| 125 | -0.823738398 | 9.617479134 | 4.518376404 | 1 |
| 126 | 0.176261602 | 9.657615177 | 4.611796963 | 2 |
| 127 | -0.823738398 | 9.670023984 | 4.566331304 | 1 |
| 128 | 0.176261602 | 9.710160026 | 4.659751863 | 2 |
| 129 | -0.823738398 | 9.722568834 | 4.614286204 | 1 |
| 130 | 0.176261602 | 9.762704876 | 4.707706763 | 2 |
| 131 | -0.823738398 | 9.775113683 | 4.662241104 | 0 |

Final weights for learning rate = 1, n = 100:

- $w0'_{final} = -0.8237383978454516$
- $w1'_{final} = 9.775113683247284$
- $w2'_{final} = 4.662241103967858$

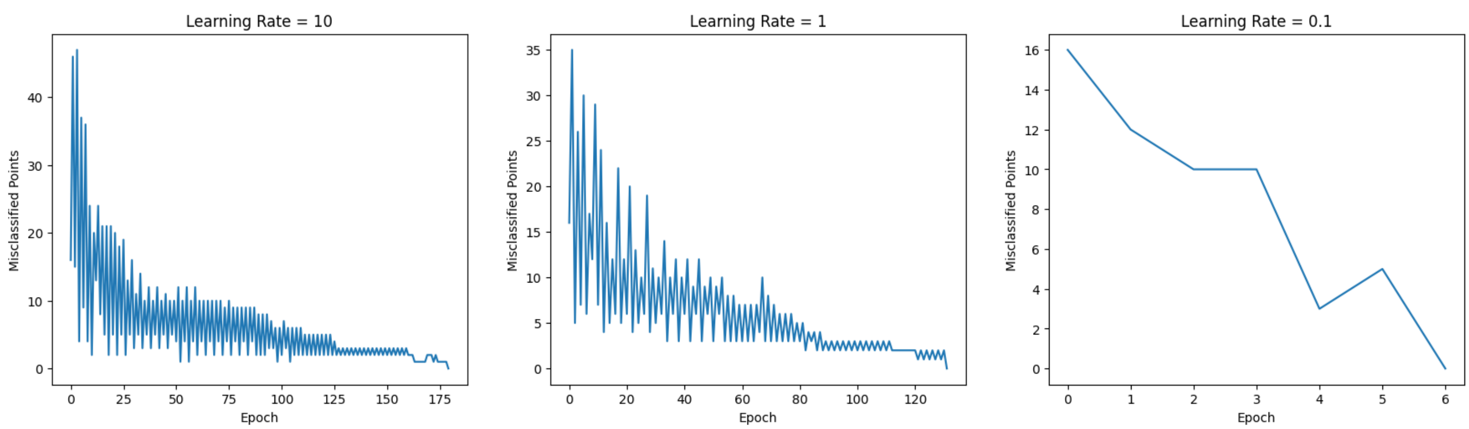
**Q: How does these weights compare to the “optimal” weights $w0, w1, w2$?**

| Weights | Optimal | Optimal_Normalized | Final | Final_Normalized | Abs_Difference |
|---|---|---|---|---|---|
| $w_0$ | -0.063 | 1.000 | -0.824 | 1.000 | 0.000 |
| $w_1$ | 0.901 | -14.370 | 9.775 | -11.867 | 2.503 |
| $w_2$ | 0.464 | -7.397 | 4.662 | -5.660 | 1.737 |

In this case, it appears that the final weights are quite close to the "optimal" weights, with slight difference for $w1$ and $w2$, where the absolute differences are greater than zero. This suggests that the training process with a learning rate of 1 resulted in significant adjustments to the weights, shooting off from the optimal values.

**<u>Part (i)(j)(k)</u>**

Following graphs are sketched for each learning rate (10, 1 , 0.1) demonstrating changes in misclassified points after updating weights at each epoch.




**<u>Part (l):</u>**

**Comment on how the changes in $\eta$ effect the number of epochs needed until convergence.**

A higher learning rate of 10 causes the weights of the neural network to be updated by large amounts during each training step while a lower training rate of 0.1 causes the weights to be updated by small amounts during each step. In our case having a higher learning rate caused the training process to become unstable and overshoot the optimal weights, while a lower training rate catered this problem and led to convergence within few epochs.

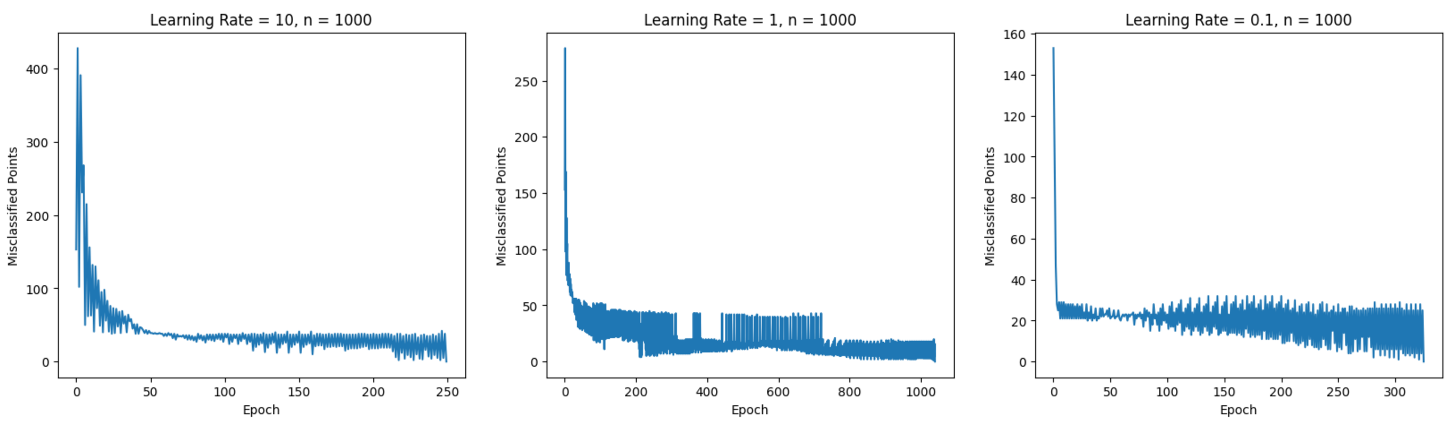
**<u>Part (m):</u>**

**Comment on whether we would get the exact same results (in terms of the effects of η on training performance) if we had started with different $w_0, w_1, w_2, S, w'_0, w'_1, w'_2$.**

Starting with a different weights and different set of points would give us a different result because the behavior of convergence depends a lot on the initial weights and the points being classified. As seen in our example, a lower learning rate converged faster as the weights were already closer to the optimal weights. The same learning rate might not be efficient if the difference between our initial weights and the optimal weights is higher. In that case, a higher learning rate would have been more effective in reaching the optimal weights.

**<u>Part (n):</u>**

**Do the same experiments with $n = 1000$ samples. Comment on the differences compared to $n = 100$.**




It took lesser number of epochs for a higher learning rate of 10 to reach convergence and higher number of epochs for a lower learning rate of 0.1 to reach convergence, as expected. However, the number of epochs for a higher number of data points required to achieve convergence has increased significantly as it required more iterations of weight adjustment to correctly classify all the points, than it takes for lesser number of data points in the dataset.