Marshall Lindsay

CS 4770

Lab1PartB

1)

For convergence testing I used the MSE and not the SSE. After each EPOC the data was shuffled. Figure 1 shows the MSE per EPOC for the initial values of alpha = 0.7, beta = 0.3

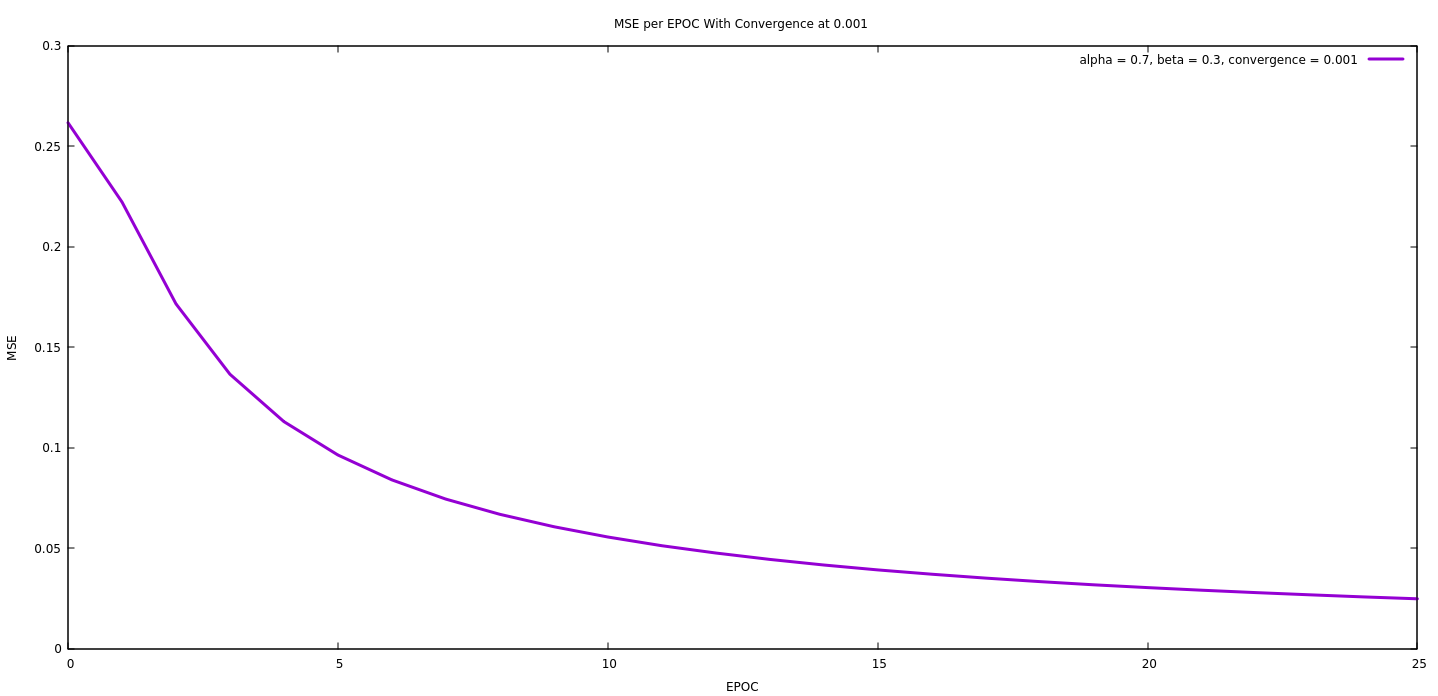
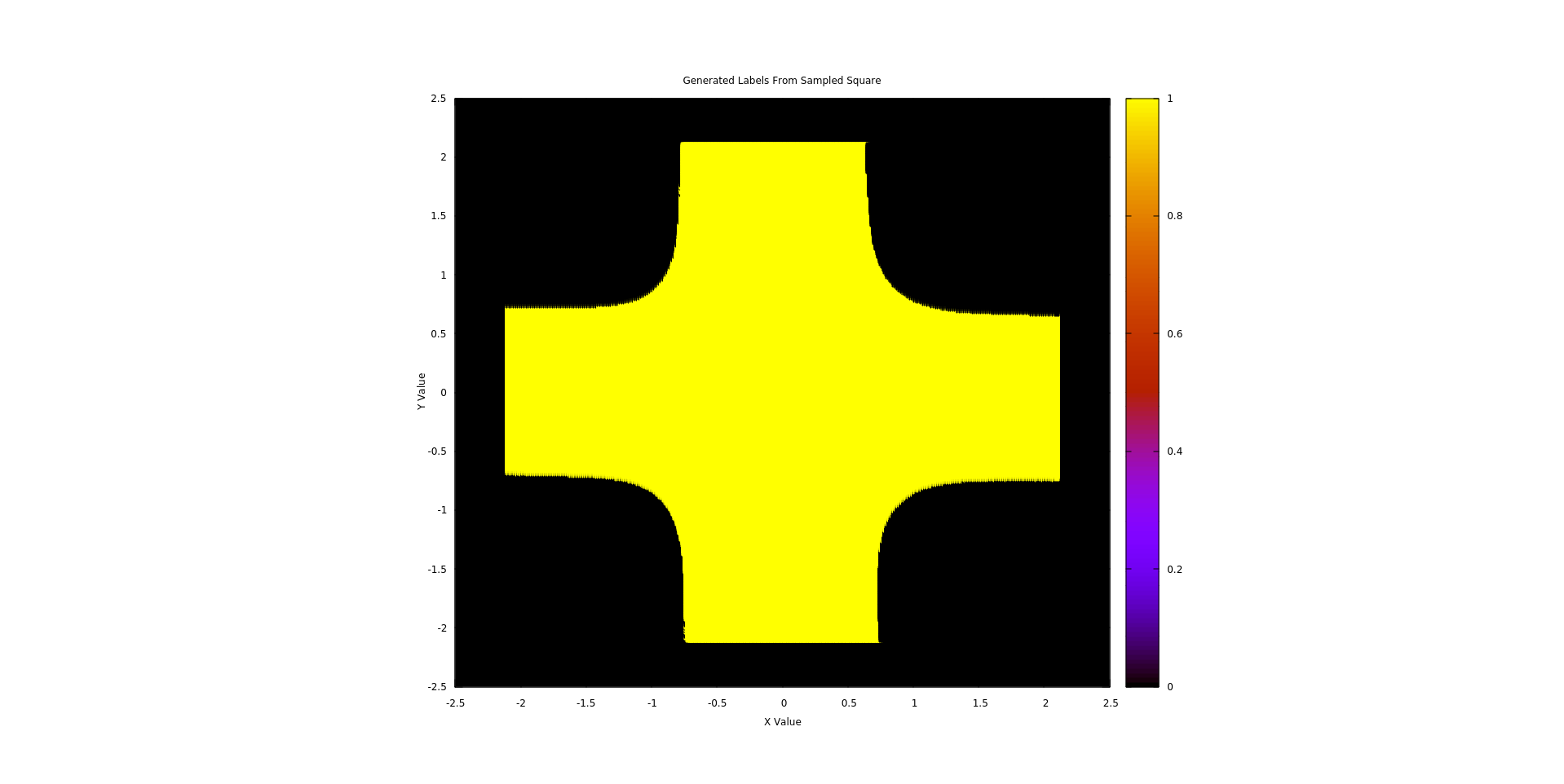


Figure 1: MSE per EPOC for alpha = 0.7, beta = 0.3 and a convergence factor of 0.001

We then tested this trained network against the square [-2.1,2.1] X [-2.1,2.1] with a granularity of 0.01 and plotted the predicted labels for each point.

Figure 2: Predicted labels of the sampled square.

We found that removing the third value of the input data had little effect on the classification of the data set. The MSE per EPOC without the third data point is shown in Figure 3 and the boundary is shown in Figure 4. This independence is due to the ability to separate the data in 2-space and not needing to go to the third.

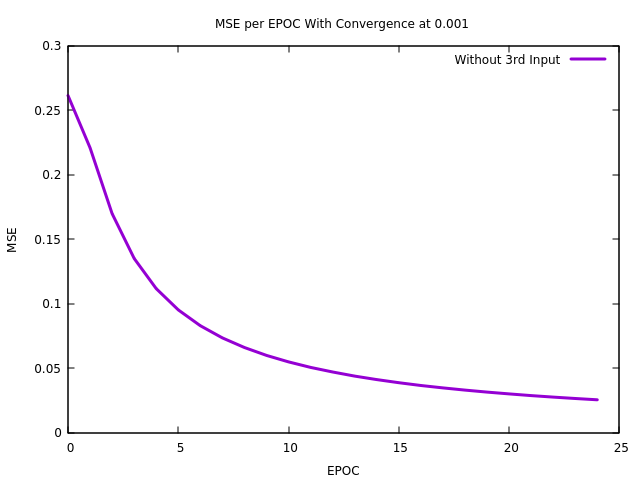


Figure 3 : MSE per EPOC without using the third point.

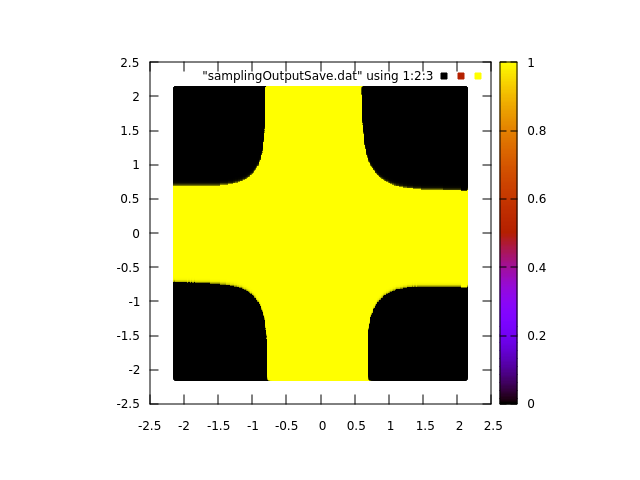
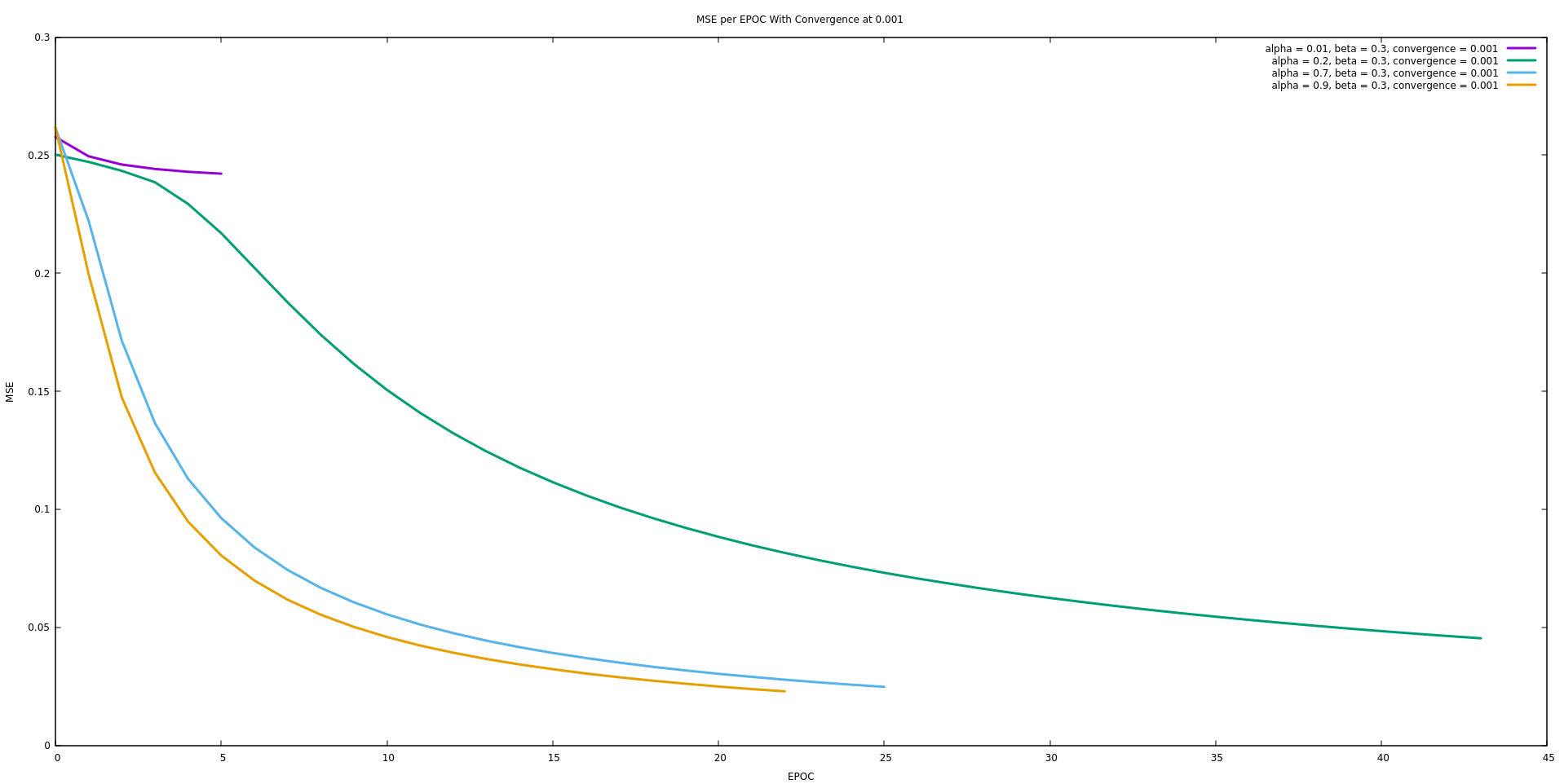


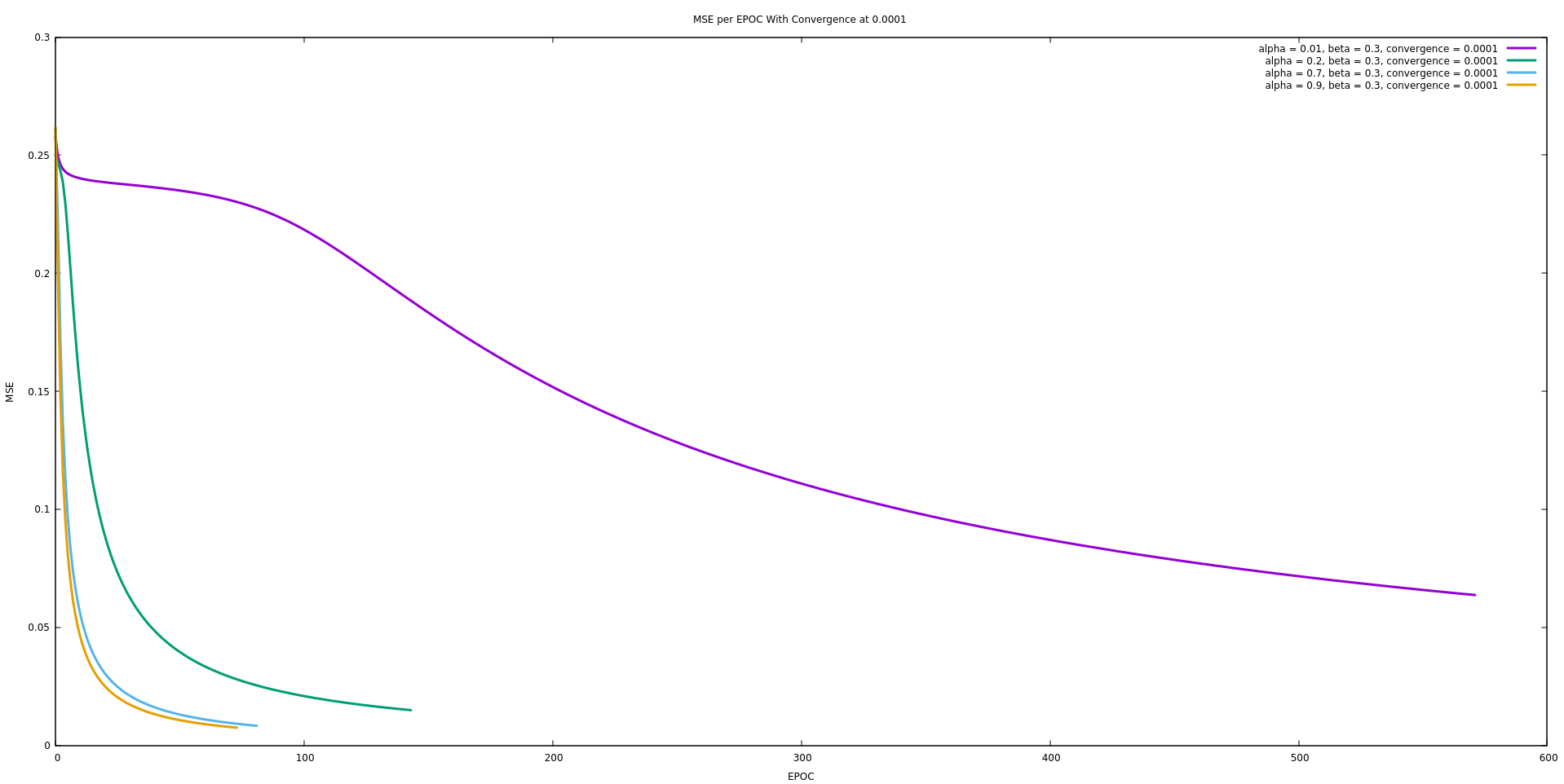
Figure 4: Predicted labels of the sampled square from the network trained without the third input.

2)

The network was then trained using different values for alpha (learning rate). The MSE was plotted per EPOC and a comparison is shown in Figure 5.

Figure 5: Comparison of the MSE per EPOC given different learning rates.

It’s clear from the comparison graph that larger learning rates cause the network to converge faster. However it was noted that for alpha = 0.01, the network was never “fully trained” i.e. the final MSE is much larger for alpha = 0.01 than the other values. We assumed this was because of our definition of convergence being that the delta MSE > 0.001. For small values of alpha, its possible that the delta MSE is very small throughout the training process and the program quits before its properly trained. To test this theory, we changed our definition of convergence to delta MSE > 0.0001 and ran the tests again. The results confirmed our suspicion and the results are shown in Figure 6.

Figure 6: Comparison of different learning rates with a definition of convergence of delta MSE > 0.0001. It’s clear from this graph that with a learning rate of 0.01, the network takes more time to finally converge on a solution.

3)

The network was retrained with variable beta (momentum rate) values and an alpha = 0.01. The results from these tests are shown in Figure 7.

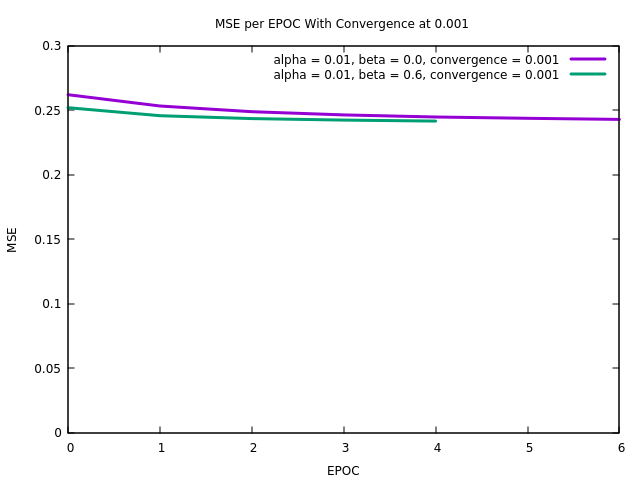


Figure 7: Comparison of MSE per EPOC for variable beta values.

Figure 7 shows terrible convergence on a solution. We assume this is partly due to the value of alpha and it’s associated problems from 2) but also on the beta values. We note that the as beta increases, the network trains to a better solution. We believe that the network would train to a smaller error if the alpha value was increased or if our definition of convergence changed to give the network a better chance.

4)

For part four… somehow I broke my program. It’s in the shop right now so If there is a chance for partial credit on the part please let me know.

What I learned from what I was able to do was how variable these networks can be. We can *think* we have a trained network based on our definition of “trained” when in reality it’s no where near a solution. In order to properly use neural nets, you must pay attention to all of the adjustable variables, and the calculated outputs to see if it is behaving how you assume it should.

Also there seems to be some dumb luck with initialization. If I initialize and guess well, I have a better chance at converging than if I guess bad.