ENGR110 Machine Learning Project Report
greenthom - 300536064

in_median	in_deviation	learning_rate	n_batch	n_hidden	n_epoch	n_samples	vector file name	weight bias file name	error	rate
0	0.5	0.5	1	50	20	5000	v1	wb1	1	0.1016
0	1	0.5	1	50	20	5000	v2	wb2	0.2061	0.8905
0	1	0.5	1	50	20	10000	v3	wb3	0.196648	0.8911
0	1	0.5	1	50	20	15000	v4	wb4	0.178838	0.9062
0	1	0.5	1	50	20	30000	v5	wb5	0.169528	0.9157
0	1	0.5	1.5	50	20	30000	v6	wb6	0.169528	0.9157
0	1	0.5	1.5	50	20	60000	v7	wb7	0.146053	0.9292
0	1	0.5	1	50	20	60000	v8	wb8	0.146053	0.9292
0	1	0.5	1	80	20	60000	v9	wb9	0.242438	0.8545
0	0.8	0.5	1	80	20	60000	v10	wb10	0.168827	0.9178
0	0.5	0.5	1	80	20	60000	v11	wb11	0.1482	0.9252
0	0.3	0.5	1	80	20	60000	v12	wb12	0.165859	0.9142
0	0.3	0.3	1	80	20	60000	v13	wb13	0.096694	0.9495
0	0.1	0.3	1	80	20	60000	v14	wb14	0.080365	0.9576
0	0.1	0.1	1	80	20	60000	v15	wb15	0.058631	0.9689
0	0.1	0.1	1	100	20	60000	v16	wb16	0.052423	0.9717
0	0.1	0.1	1	300	20	60000	v17	wb17	0.03424	0.981
0	0.1	0.1	1	300	100	60000	v18	wb18	0.033442	0.9825

All 18 tests with all variables were used in the process. Best Test: WB18 – 0.9825

Understanding the variables used to gain the specific rate was necessary when starting on testing. On the first couple of tests, I set basic variables such as; batch, hidden, and epoch to a base number so I could determine an ideal deviation, learning rate, and sample variable. This allowed me to gain a rate of 0.9157. Initial testing with base variables set is shown below:

in_median	in_deviation	learning_rate	n_batch	n_hidden	n_epoch	n_samples	vector file name	weight bias file name	error	rate
0	0.5	0.5	1	50	20	5000	v1	wb1	1	0.1016
0	1	0.5	1	50	20	5000	v2	wb2	0.2061	0.8905
0	1	0.5	1	50	20	10000	v3	wb3	0.196648	0.8911
0	1	0.5	1	50	20	15000	v4	wb4	0.178838	0.9062
0	1	0.5	1	50	20	30000	v5	wb5	0.169528	0.9157
0	1	0.5	1.5	50	20	30000	v6	wb6	0.169528	0.9157

Now that I had reached a comfortable rate I then went on to test other variables, either to increase or decrease them from the base number given, to increase the rate. At the start of this process, I set samples to 60000 due to the number of samples in the mnist_train.csv was that amount. Epoch was set either to 20-100 depending on the rate given on testing (increased to 100 when an increase on rate was given testing at 20 epoch). Epoch testing at 100 was however not noted down as this was the process I had to guarantee an increase in rate if variables were to be increased or decreased at this stage, testing was only noted down at 20. The batch was set at a base of 1 but was altered between 1-1.5, however, had no effect due. Hidden was set at a base of 50 but

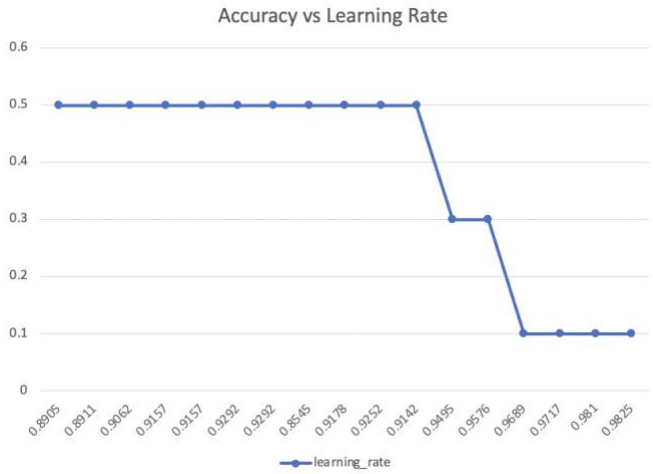
0	1	0.5	1.5	50	20	60000	v7	wb7	0.146053	0.9292
0	1	0.5	1	50	20	60000	v8	wb8	0.146053	0.9292
0	1	0.5	1	80	20	60000	v9	wb9	0.242438	0.8545
0	0.8	0.5	1	80	20	60000	v10	wb10	0.168827	0.9178
0	0.5	0.5	1	80	20	60000	v11	wb11	0.1482	0.9252
0	0.3	0.5	1	80	20	60000	v12	wb12	0.165859	0.9142
0	0.3	0.3	1	80	20	60000	v13	wb13	0.096694	0.9495
0	0.1	0.3	1	80	20	60000	v14	wb14	0.080365	0.9576
0	0.1	0.1	1	80	20	60000	v15	wb15	0.058631	0.9689
0	0.1	0.1	1	100	20	60000	v16	wb16	0.052423	0.9717

raised to 50-100 when the rate was reached for testing, this was further increased to 300 at the final testing stages when we wanted to check if the variables that we were using were able to create complex shapes. Testing with altering the base variables can be shown here:

From vector files 1 to 16 I had reached a rate of 0.9717. From this point, I altered the epoch to 100 and the hidden neuron value to 300 to reach my final rate of 0.9825.

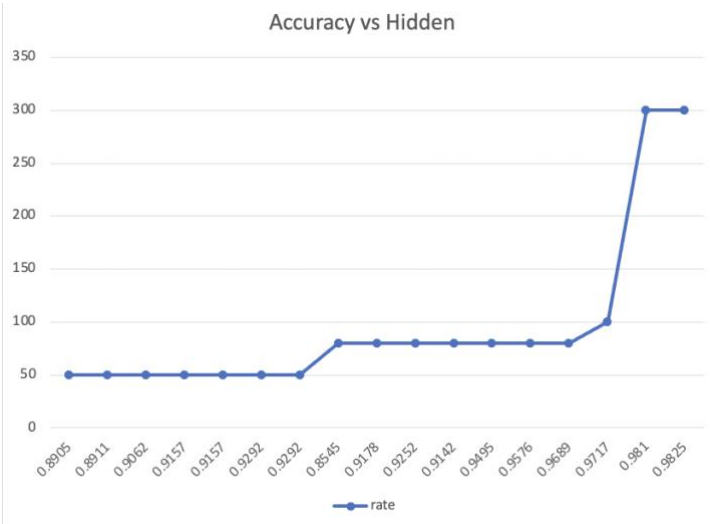
- **What was the final value of the learning rate and how did you arrive at that?**

Deciding on the learning rate was all dependent on the deviation size and what the other variables were set to at the time. When testing at a base variable, the learning rate was set relatively high based on the batch size at the time. At this point in the testing process, I was trying to evaluate proper variables for deviation and sample size to test and find out how the variables worked. This is shown in my early testing. Once we had reached a rate of 0.9157 we started to alter the base variables such as; batch, hidden, and epoch, so we could increase the rate and see if the learning rate and deviation provided would work. Testing for this included increasing samples to 60000 (due to sample size provided), hidden set to 100 (want to execute complex shapes), and batch set to 1 (want an accurate rate). At this point in testing, I realized that what learning rate I set the deviation would need to be similar to due to how I set the other variables. From here we tuned the learning rate, and deviation and hidden again so we could achieve a higher rate.



- **How a number of hidden neurons affected network performance. What is the reason?**

The reason for neuron models in machine learning is to mirror and model the behaviour of neural networks found in living organisms. By involving this we can create a model architecture to process data and how it would process in a living brain. The final number of hidden neurons 300. The number of neurons within the late testing stages affected the rate of 83.25. Overall adding more neurons allows u to create more complex shapes which provide more effectiveness and accuracy on the end rate, however too many may make neurons incorrectly differentiate the dataset. If there aren't enough then we cannot differentiate between data points.

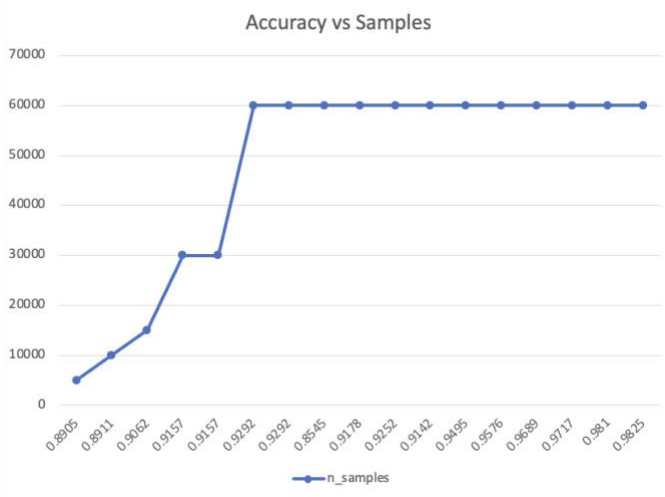


- **Same for initial weights value deviation? Why did you select this particular value?**

Testing deviation throughout the project all relied on what the learning rate was. If the input is the same the output is always going to be the same. If we start at a 0 deviation (neurons start at the same value), this would behave like a single neuron network. By using this theory, in the early stages of testing I used base variables, I was configuring only learning rate, deviation, and samples. Once I reached an appropriate number to alter the other variables from the base number I then had to contrast that with the rate it was coming back as it was altering with other variables. Due to the dataset altering from 10000 to 60000 this meant more samples were being analysed in the algorithm which would affect my first value of deviation. As shown below. When I started testing my base variables at a deviation of 0.5 and 1 this returned the rate 0.9157, I then altered deviation as well as learning rate to achieve a higher rate in comparison to other variables. The final deviation value I got was 0.1 which was identical to my learning rate.

- **Did the increasing number of samples significantly change the optimum values of other parameters? Explain why.**

If the sample size is small, we might not be able to cover the different cases for the neurons to run into. The amount of data provided was 60000 in the mnist_train.csv file so I knew my end variable for samples would have to match that amount. In initial testing, I started with 10000 so I could create stable and accurate variables for deviation and learning rate. Once I had created an accurate deviation and learning rate I then upped the number of samples to 60000 to create more accurate testing.



- **What performed better: online training or mini-batch training? What does it say about our input data?**

Mini-batch is where we use a subset of the dataset provided which would take another step in the learning rate. This means that the mini-batch can have a value that is greater than one that is also less than the initial size of the training set. This would mean that we can calculate the entire dataset without waiting to find the individual learning rate of the sample. This would also reduce the risk of getting stuck at a minimum since the different batches would be considered at each iteration. On-line training is looking at each pattern as it is observed whereas mini-batch is looking over groups of patterns. If we use mini-batch training in our model we would get worse results due to the data points being unorganized. This means the different data is attempting to attract the neuron to diverge which may result in no development of the neuron. Online training is the scenario used.