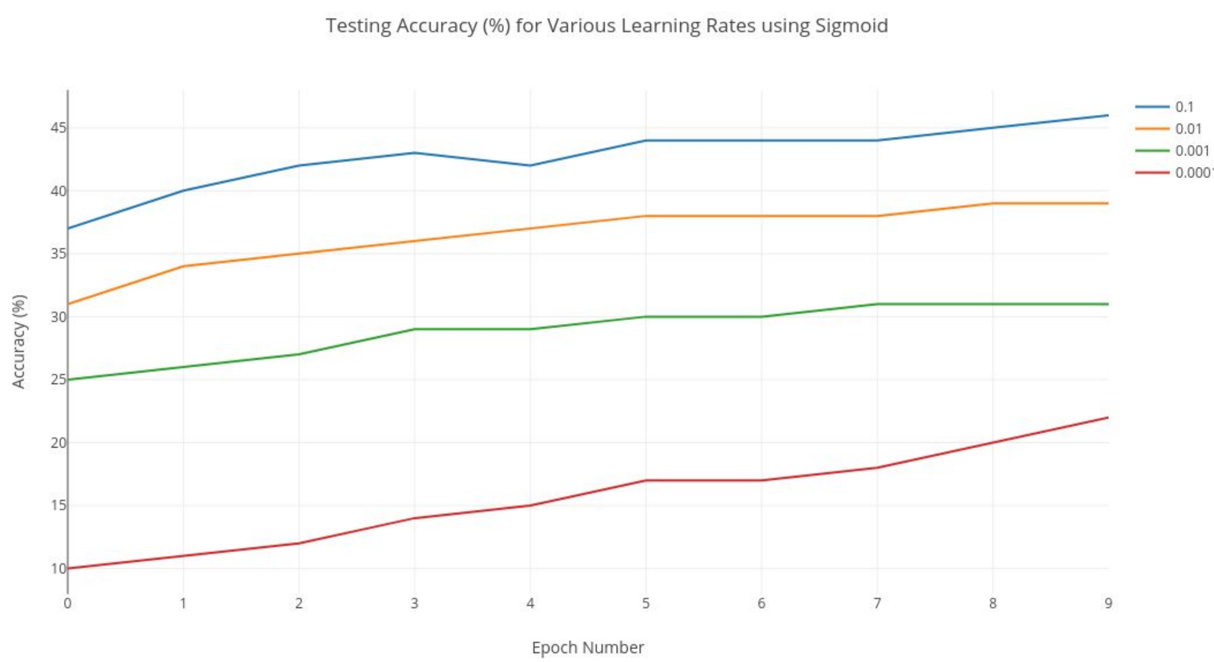
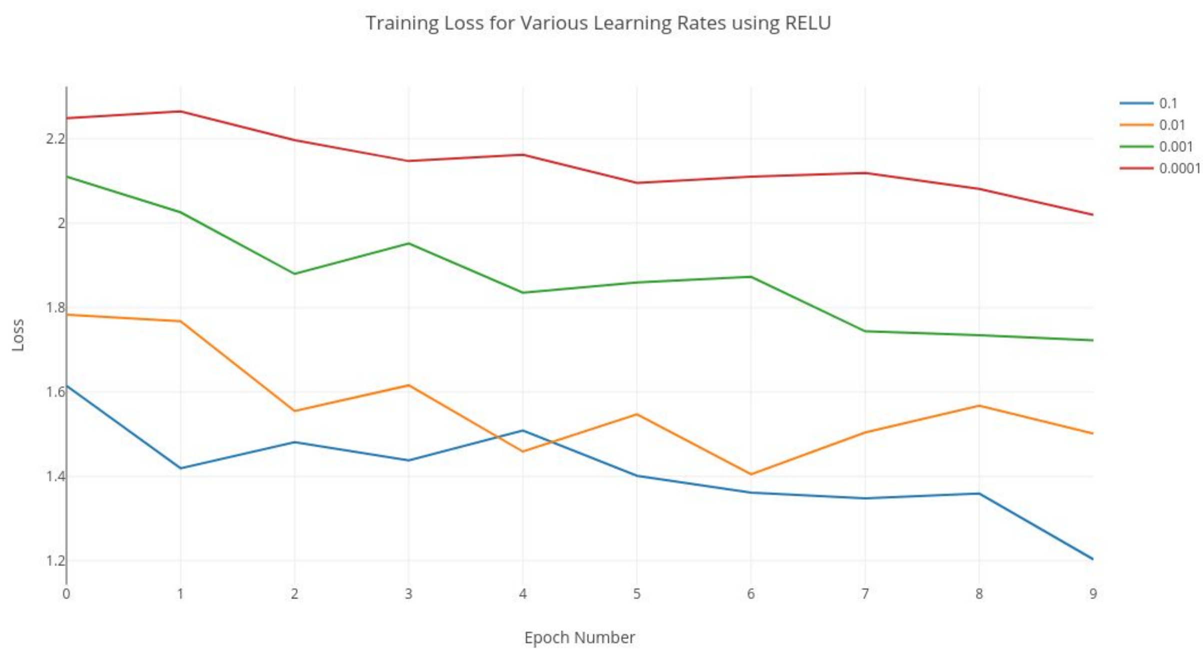
Brian Blythe and Parker Bruni
Machine Learning
Assignment 3
05/15/18

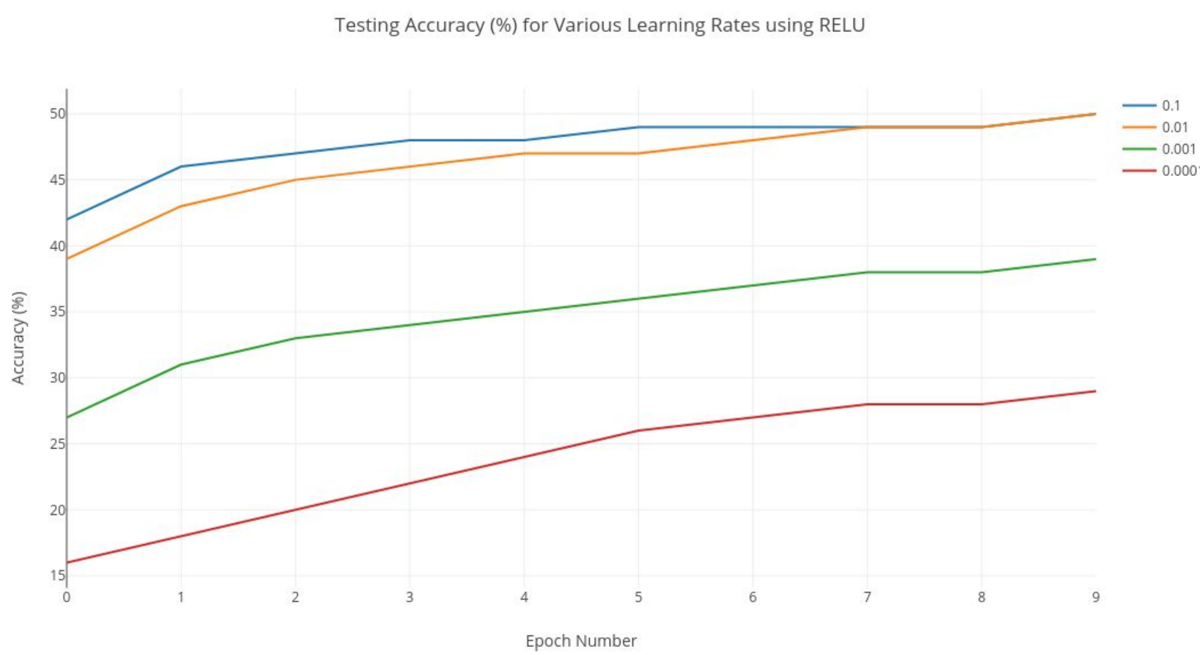
1. Activation function: Sigmoid, Hidden Layers: 1, Output Layers: 1, Hidden nodes: 100, Dropout: 0.2, Momentum: 0.5, Weight Decay: 0



Summary: The smaller the learning rate, the more training loss and less testing accuracy were computed for a fixed number of batches. We stopped training once a fixed number of batches were run, (ten in this case). In this way, we could compare the effect of the learning rate independent of other parameters. Our final system we chose was the one with the learning rate of 0.1. It had testing accuracy of 47%.

2. Activation function: Relu
   Same other parameters

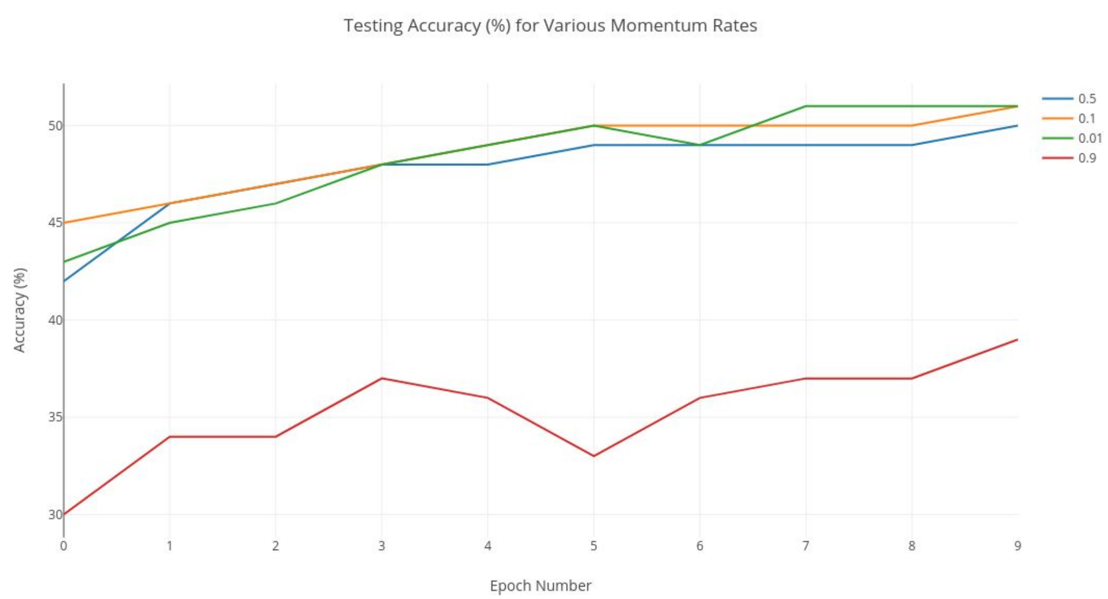




Summary: We saw similar results with respect the the learning rate as with the sigmoid function. However, in general the relu function performed better than the sigmoid function for both accuracy and training loss at all learning rates. For a learning rate of 0.1 the testing accuracy was 51%.

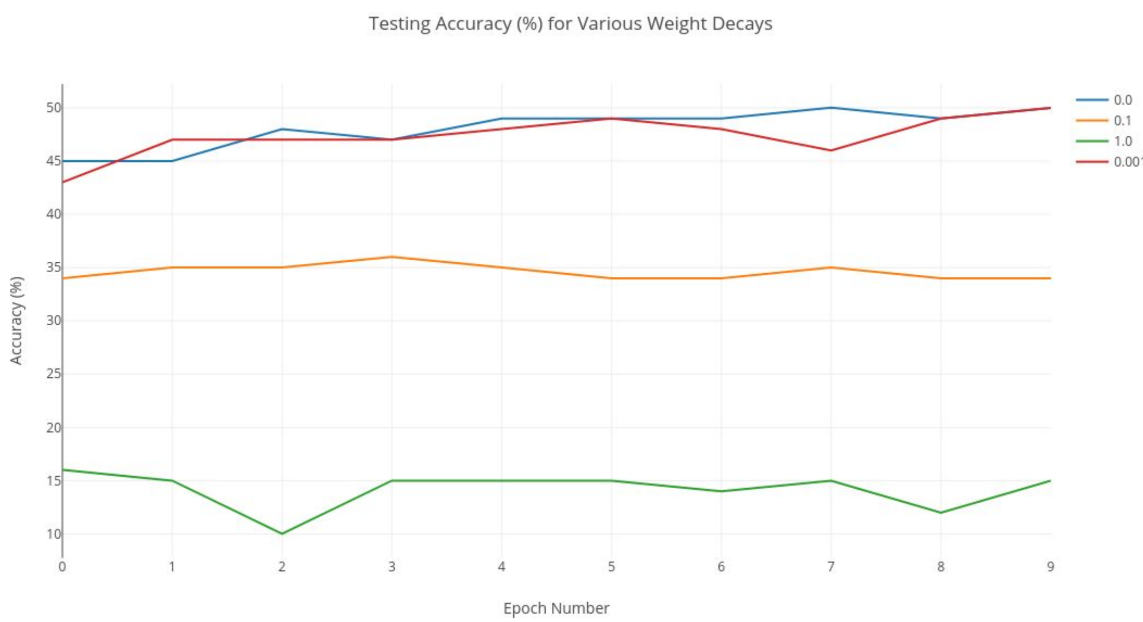
3. Activation function: Relu, Learning rate: 0.1, Hidden Layers: 1, Output Layers: 1, Hidden nodes: 100, Dropout: 0.2, Weight Decay: 0

Changing the momentum



Summary: A momentum value of 0.9 had relatively poor training loss values and testing accuracy. As momentum values got smaller, up to 0.01, the performance for both loss on the testing data and accuracy of the training data increases. It seems that there are diminishing returns for momentum values lower than 0.5, as the performance of the testing accuracy seemed relatively unchanged for any lower values. Ideally, a momentum value of about 0.5 seems sufficient for highest performance. If there is no computation cost for smaller values of momentum, than the smallest value approaching 0 would be ideal. We did not test for computation/time costs for extremely low or negative momentum values, but if we assume that there is computation cost for lower momentum values, then we select a value of 0.5 as the best momentum value. Otherwise, a momentum value as close to 0 as possible is ideal.

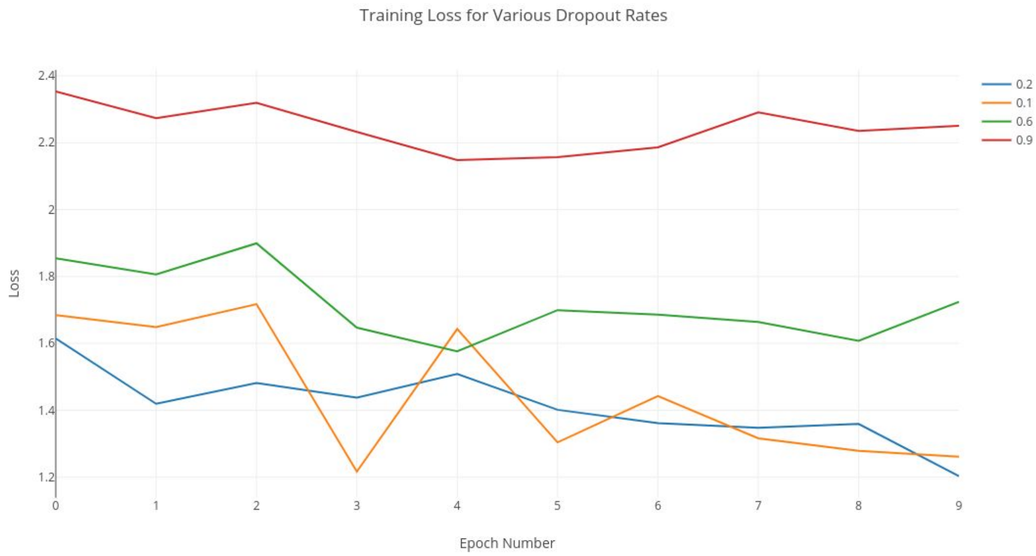
Changing the weight decay
Set momentum back to 0.5

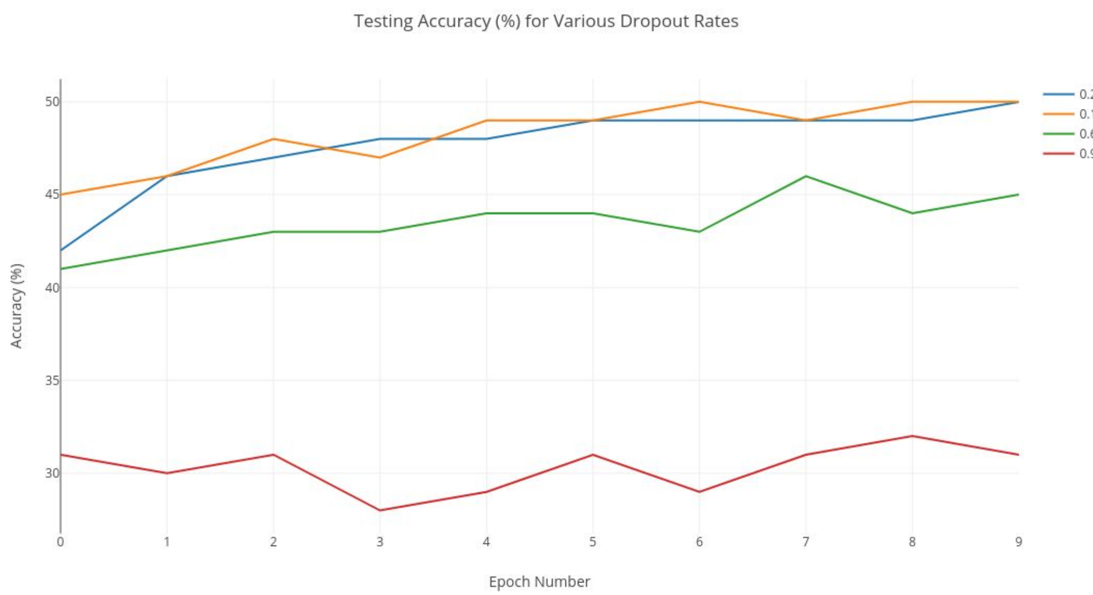


A larger weight decay reduces the testing accuracy and increases the training losses. For large values of the weight decay, (we tested up to values of 1), the testing accuracy dropped down to almost as low as random guessing, effectively making the neural network useless. A weight decay of 0 works best.

Changing the dropout

Set weight decay back to 0



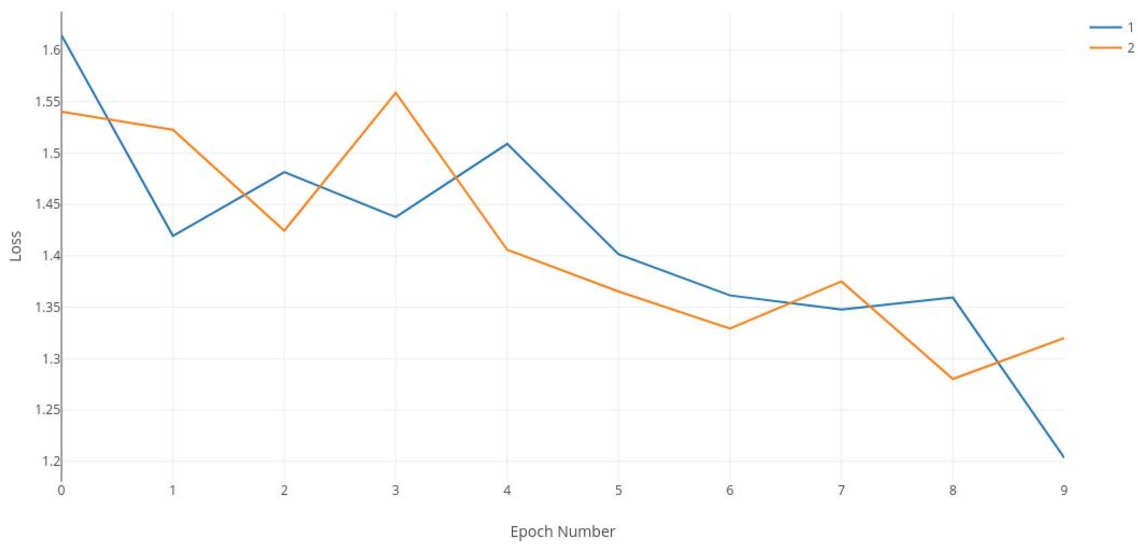


Summary: Lower dropout rates seemed to give better performance than higher rates. Our highest tested dropout rate of 0.9 performed significantly worse for both loss and accuracy than our other dropout rates. It seemed that lower dropout rates performed better overall down to about 0.2. Values lower than 0.2 improved slightly but not significantly.

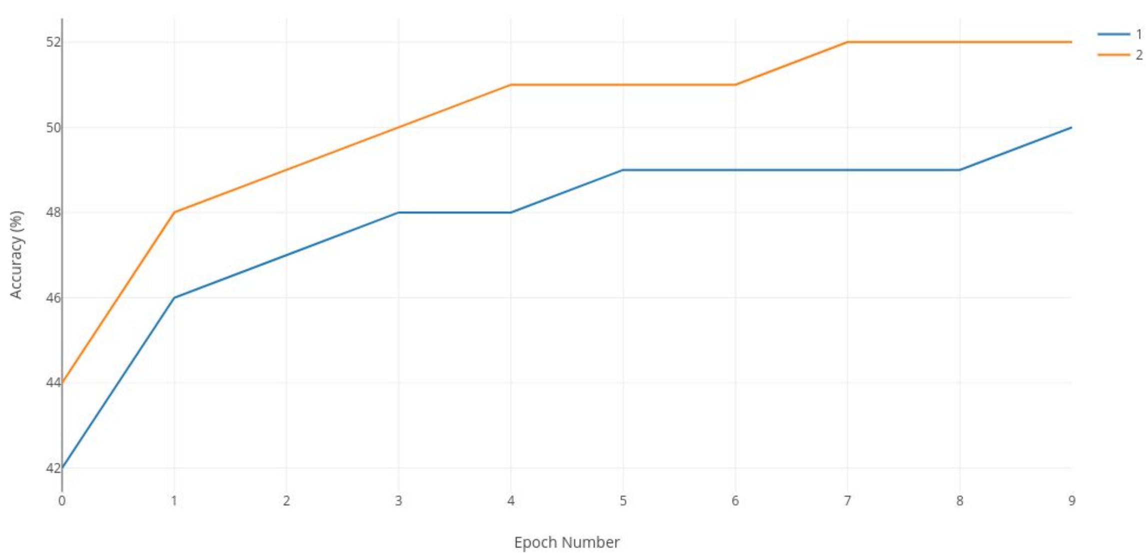
4. Activation function: Relu, Hidden Layers: 2, Output Layers: 1, Hidden nodes: 100 (50 per layer), learning rate: 0.1, Dropout: 0.1, Momentum: 0.5, Weight Decay: 0

   To compare the three layer network to the two layer network, we chose the highest performing parameters from the two layer network and applied them to the three layer network as well.

Training Loss for 2 layer VS 1 Layer



Testing Accuracy (%) for 2 Layer VS 1 Layer



Summary: The two layer graph performed better than the one layer graph in terms of testing accuracy but had comparable training loss values.