Matthew Oakley
Deep Learning w/ Tensor Flow
Dr. Rivas
October 24th, 2017

Test Accuracy	**Learning Rate**				
Neurons	**0.00001**	**0.0001**	**0.001**	**0.01**	**0.1**
16	0.0958	0.3161	0.8941	0.9526	0.7806
32	0.1613	0.6847	0.926	0.9691	0.8507
64	0.2547	0.8352	0.9562	0.9743	0.8728
128	0.4502	0.895	0.9664	0.9792	0.101
256	0.6478	0.9265	0.9568	0.9802	0.1028
512	0.7533	0.937	0.9728	0.9318	0.1135
1024	0.8191	0.9488	0.975	0.4147	0.101
2048	0.8548	0.9336	0.9687	0.0974	0.098
4096	0.8832	0.9369	0.9499	0.0935	0.0958

Fig 1. Chart of test accuracy based on different learning rates and neurons.

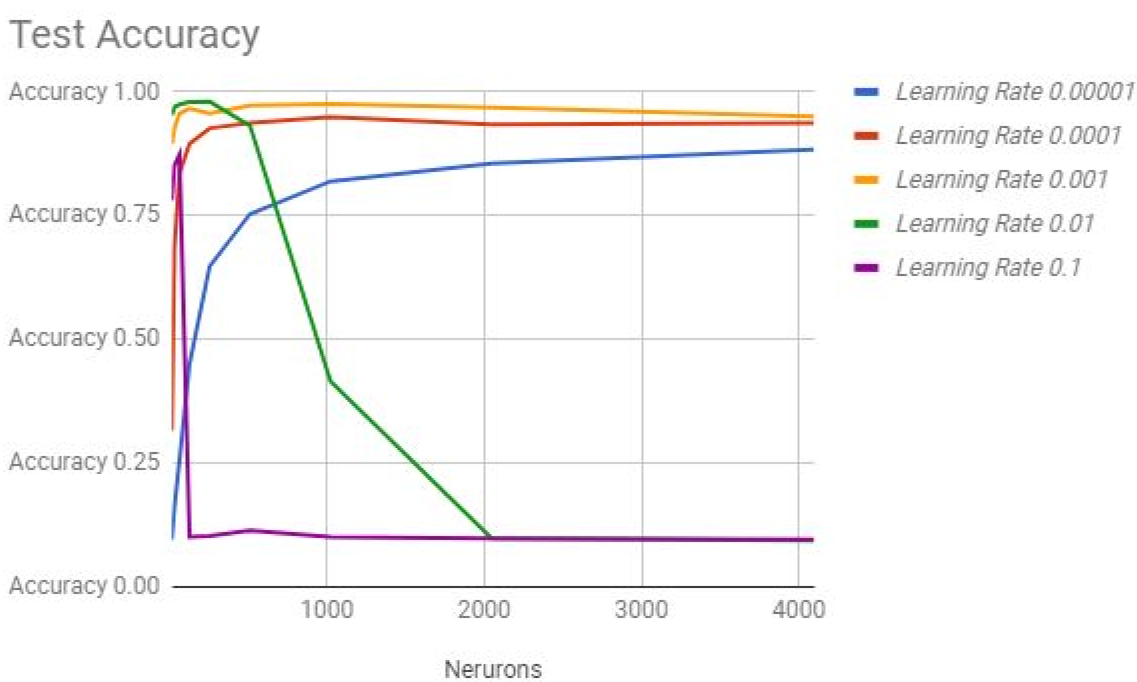

Fig 2. Line graph of test accuracy with different values for learning rate and neurons

The amount of neurons helps increase the accuracy up to a certain point depending on the learning rate. For example with smaller changes in learning rates more neurons help it get a higher accuracy. This is seen in (1) where a very low learning rate improves dramatically when the number of neurons is increased. Also seen in (1) having a high learning rate with an increase in neurons causes problems. These problems are caused because the LSTM is being overtrained or took to many steps and experienced explosive gradient. Figure 1 and 2 both show that it is important to make sure you do not overtrain. Overtraining is when you teach it to learn the training data and it will not work with the test data. Also another problem that overtraining brings is that it will not be a generalized. Generalized programs will work well with never before seen data and if a system is overtrained it will not work well with new data.