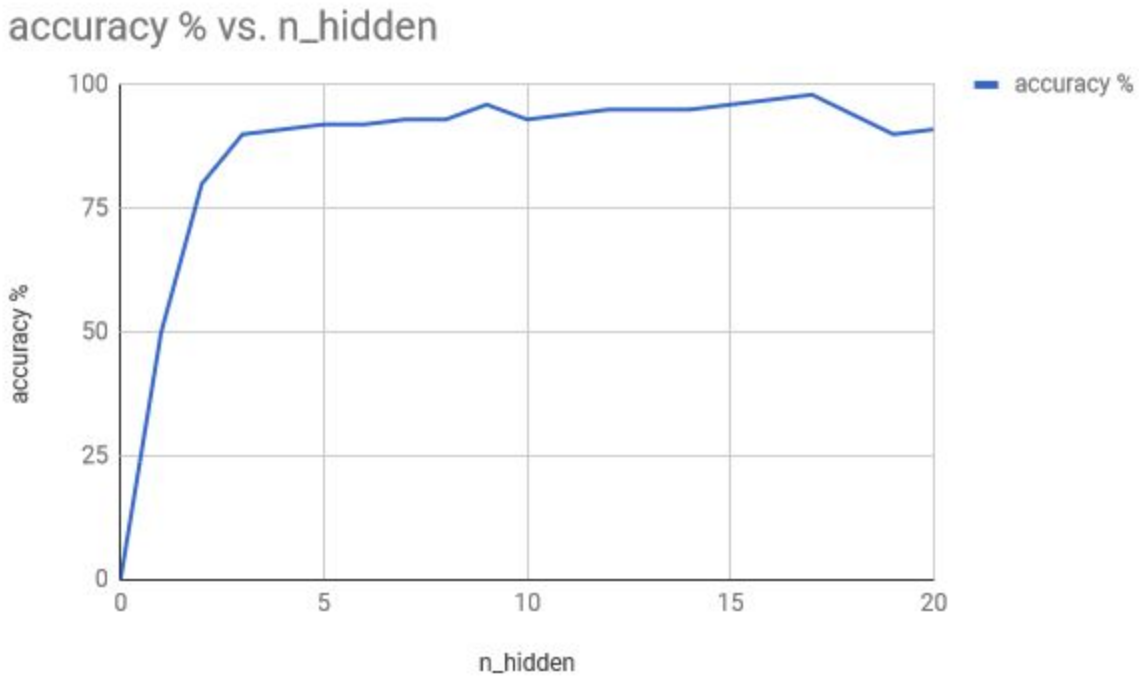
Drew Wentka (50110060), Nate Sopko (50120166), Cole Hastings (5015503)
CSE 474 Assignment 1 Report

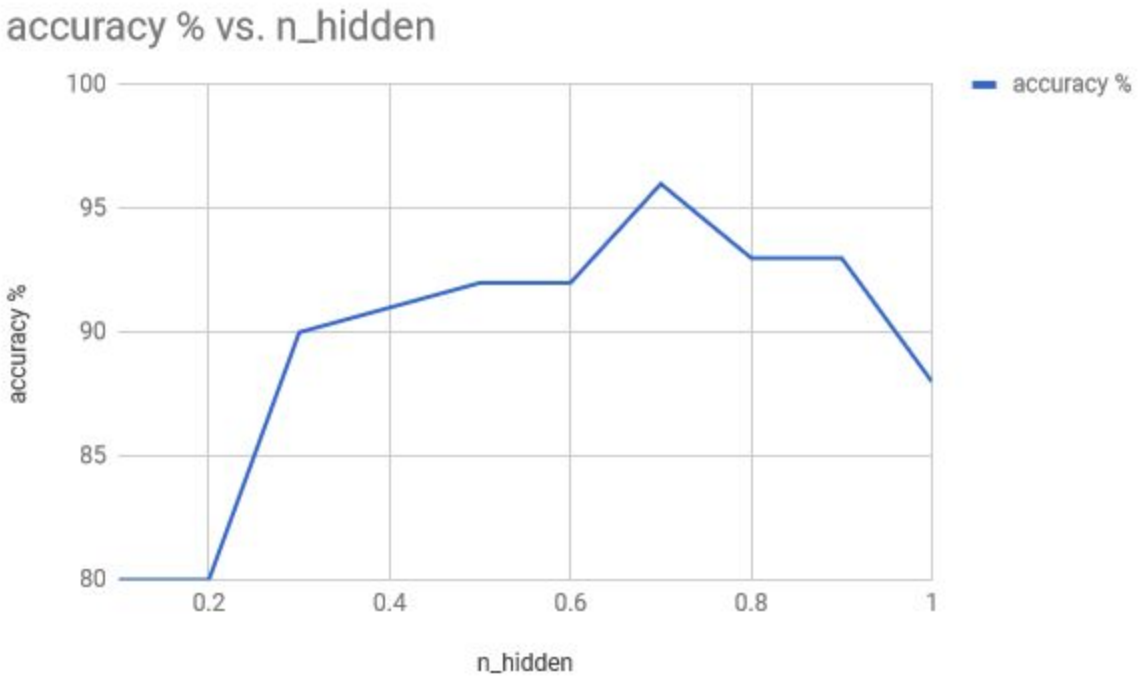
Choosing Optimal N_Hidden

To calculate the best n_hidden value to use we would run experiments with our neural network to simulate trial and error testing. Starting low at 1 we would increment the n_hidden values to add more layers in every test run while comparing accuracy of the network. As soon as we start seeing diminishing returns on layer additions after two runs it would be safe to say we have gone too far and accept the last value that resulted in positive accuracy gain.



Choosing Optimal λ

When choosing optimal lambda, the process we would follow starts with a very small number close to 0 while keeping all other parameters constant. After running this test, we would note that accuracy value to base our adjustment of lambda off of. This would prompt a test with lambda higher around 1, in which we would let our neural network run and calculate its accuracy with this extreme. Finally, this initiates a ‘closing in’ portion of our testing to find the optimal lambda for our neural network.



TensorFlow Findings

When running the deep neural network with three layers, we had an overall test accuracy of 89.77% and a run-time of 34 minutes and 46 seconds.

When running the deep neural network with five layers, we had an overall test accuracy of 88.95% and a run-time of 39 minutes and 49 seconds.

When running the deep neural network with seven layers, we had an overall test accuracy of 86.71% and a run-time of 49 minutes and 6 seconds.

Overall, a small increase in layers doesn't cause much of a notable change in accuracy or runtime. However, when increasing layers further we noted a drop in accuracy and an increase in runtime, displaying diminishing returns on layer increases.

Number of Layers	Accuracy %	Runtime HH:MM:SS
3	89.77	00:34:46
5	88.95	00:39:49
7	86.71	00:49:06