Neural Networks and Deep Learning

CSCI 5922 – Assignment 6 (Fall 2017) by **Akshit Arora (108631342)**

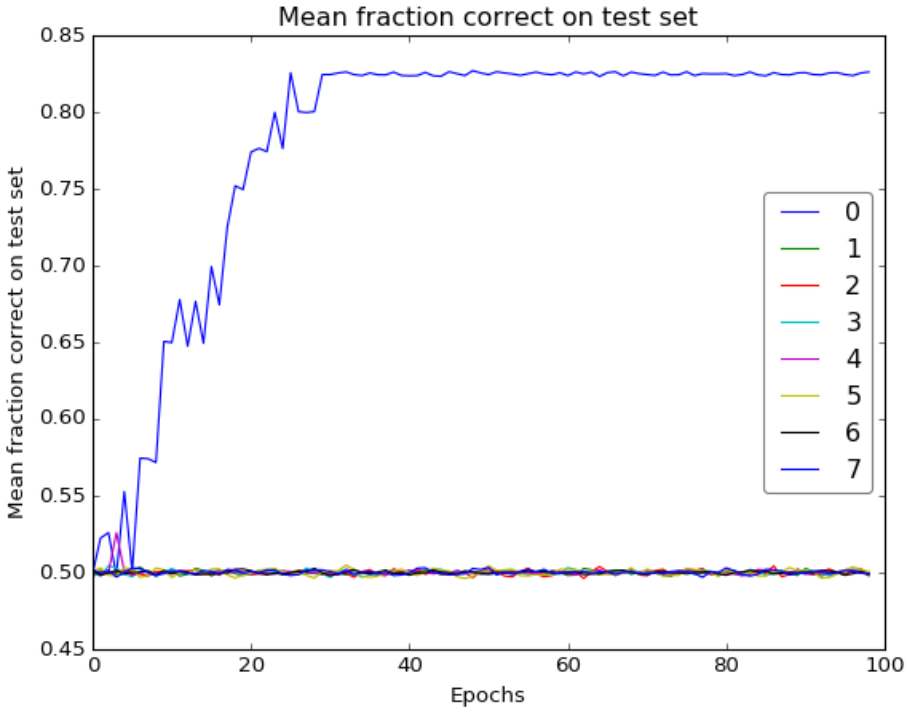
As per guideline on piazza, here is the code I started working with: https://github.com/aymericdamien/TensorFlow-Examples/blob/master/notebooks/3_NeuralNetworks/dynamic_rnn.ipynb. Due to long training time, I used the Summit super-computing cluster (with GPUs) by research computing of CU that took much less time in executing the training process. (https://www.rc.colorado.edu/resources/compute/summit) I wrote a small code to submit the code as a job to the GPU cluster. I saved the numpy arrays as '.npy' extension and then plotted them in my local Jupyter notebook.

Part 1:

Legend for the upcoming plots:

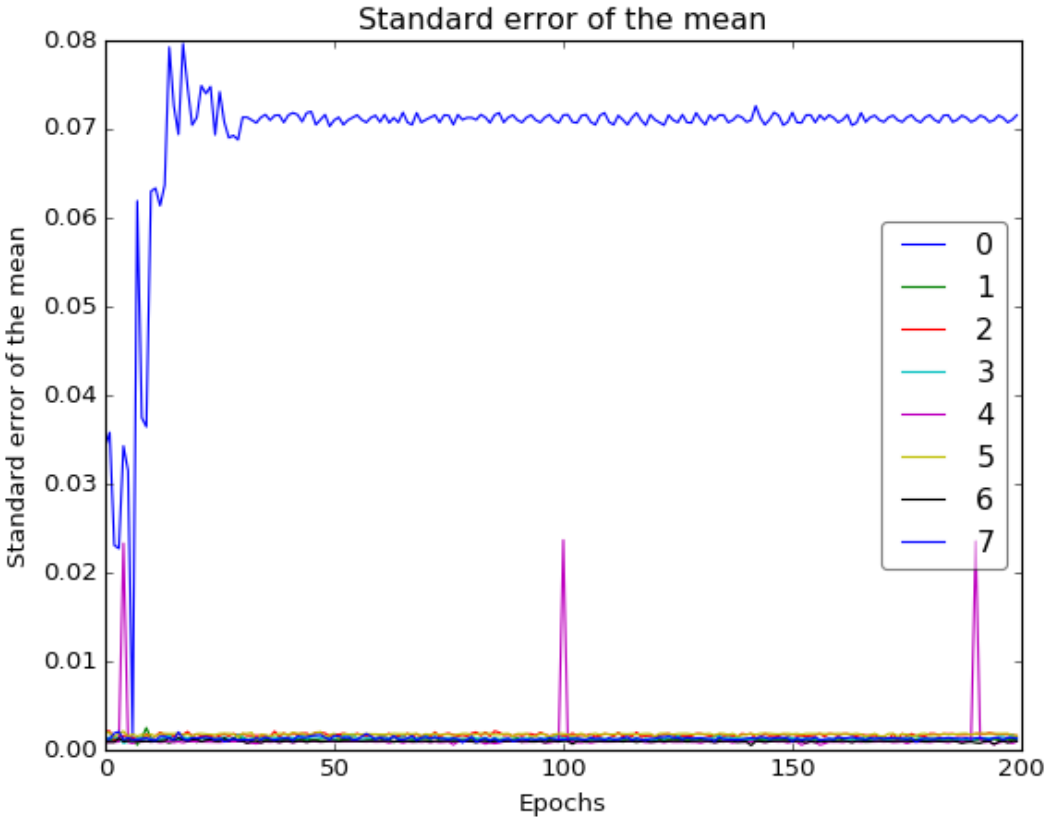
#	Architecture [sequence length, hidden units]
0	2,5
1	10,5
2	25,5
3	50,5
4	2,25
5	10,25
6	25,25
7	50,25

Mean fraction correct on test set for RNN (100 epochs):

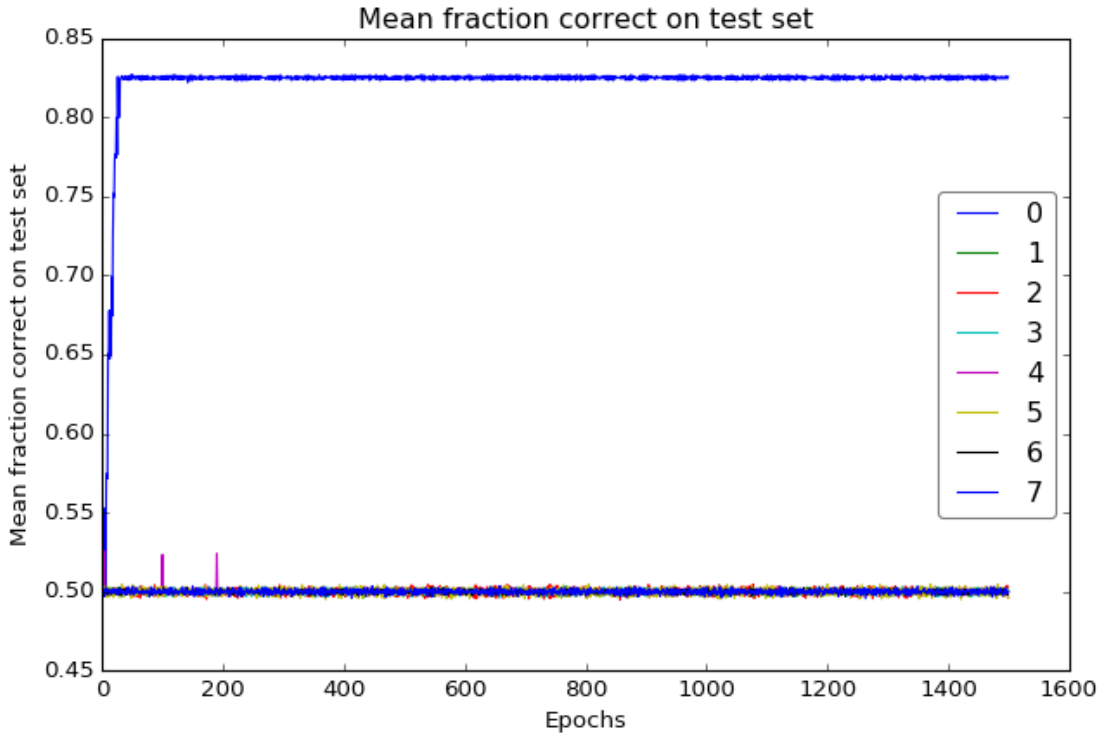


Only model 0 (with sequence length = 2 and hidden units = 5) gives good result. Rest every model gives baseline accuracy of 50%

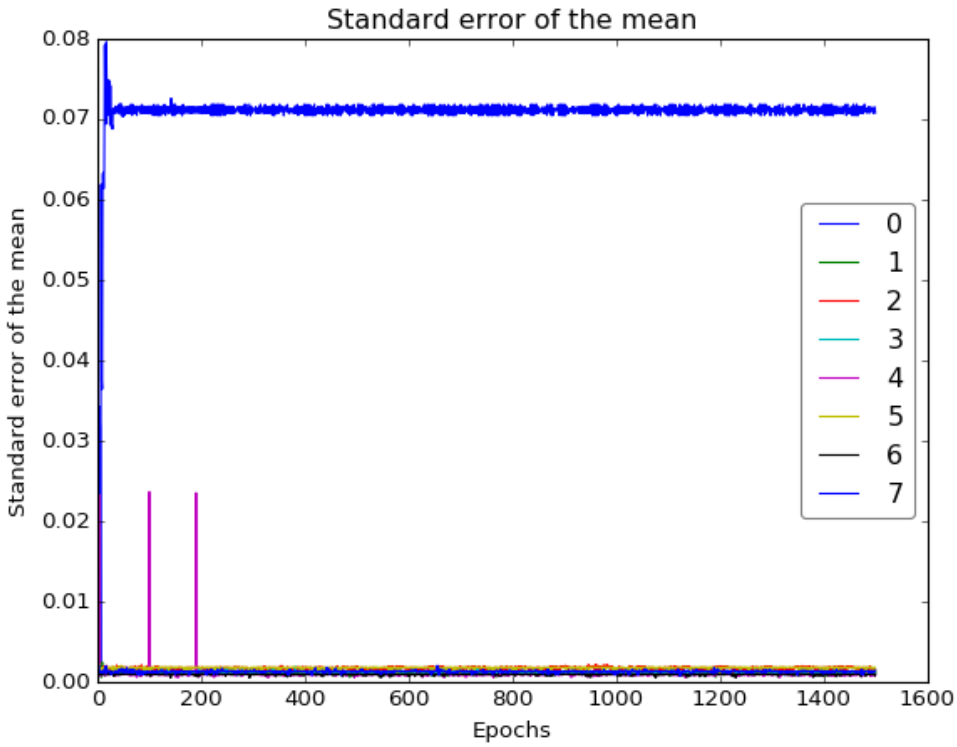
Here's what the standard error of the mean looks like:



Here are the plots for 1500 epochs:

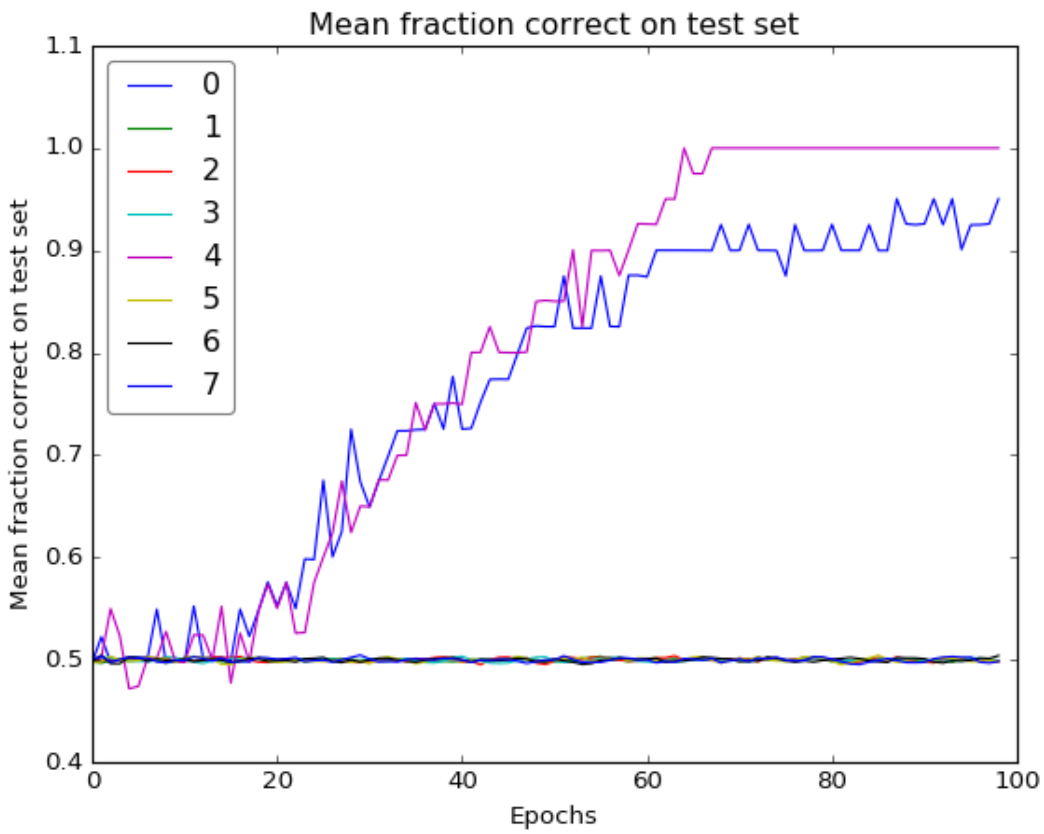


For 1500 epochs, the standard deviation looks like:

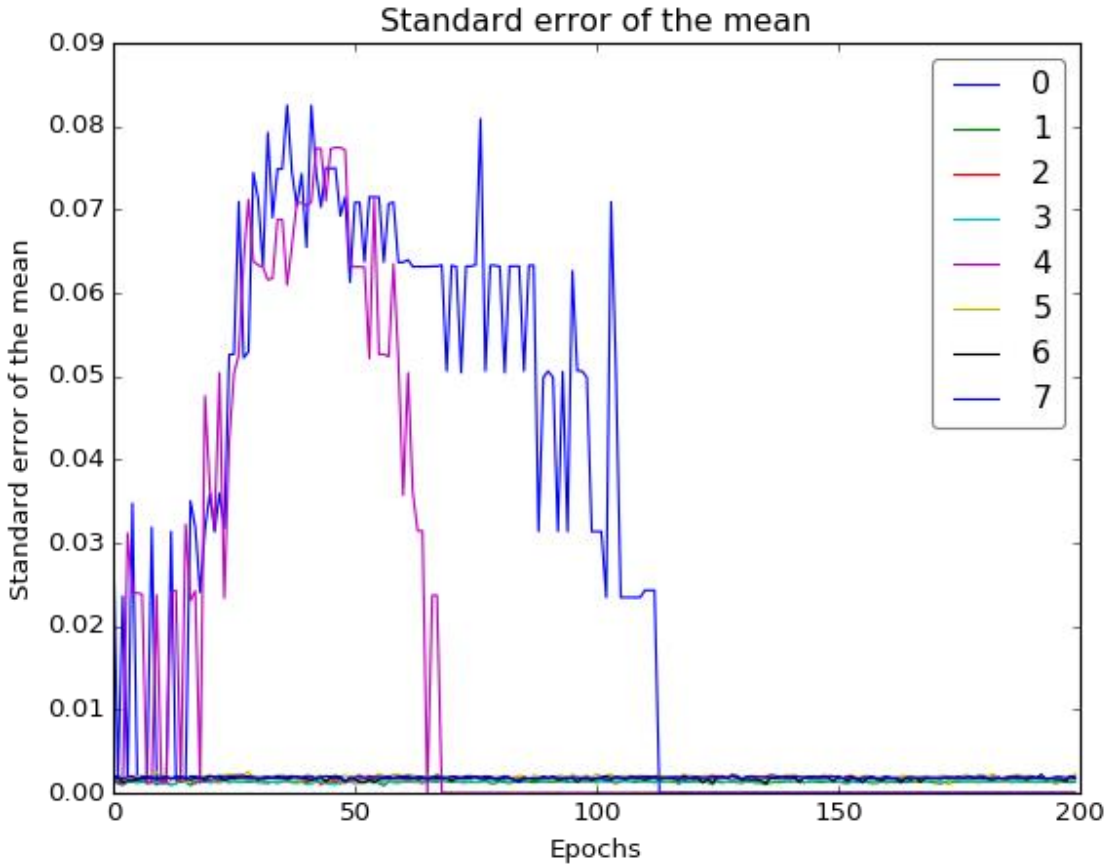


Part 2:

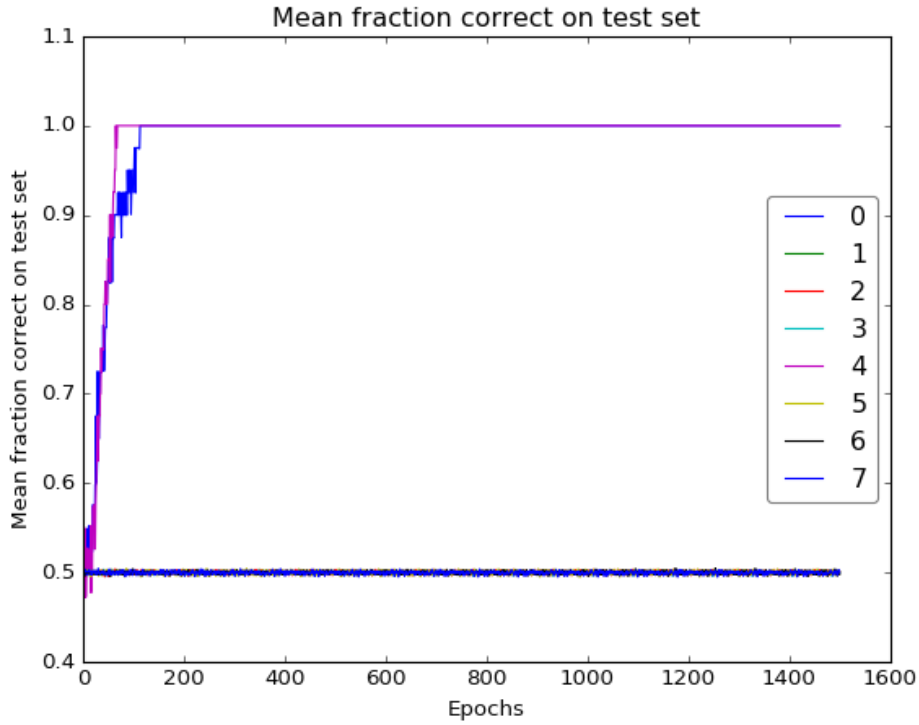
For LSTM, we have not only model 0 (with sequence length = 2 and hidden units = 5) stands out but also, model 4 (with sequence length = 2 and hidden units = 25).

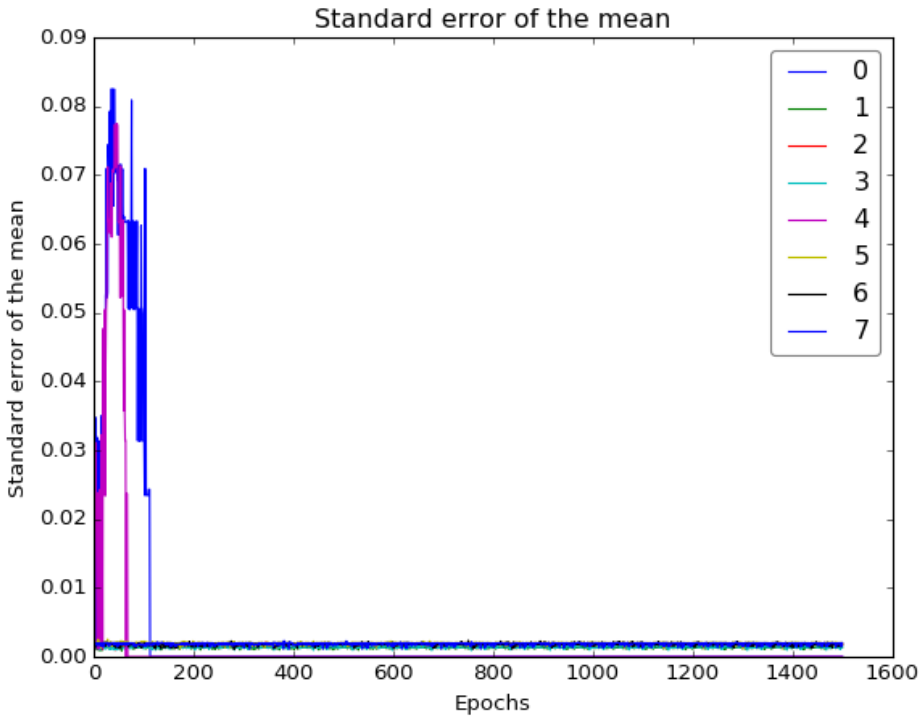


Here's what the standard error of the mean looks like:



Here are the plots for 1500 epochs for LSTM:





Please note that in the code submitted, Tensorflow version 0.12 has been used instead of the latest 1.4. It is because CUDA, CUDNN have been setup with python 3.5.1 and I didn't have the rights to upgrade any of those modules on, Research Computing's supercomputer, Summit.