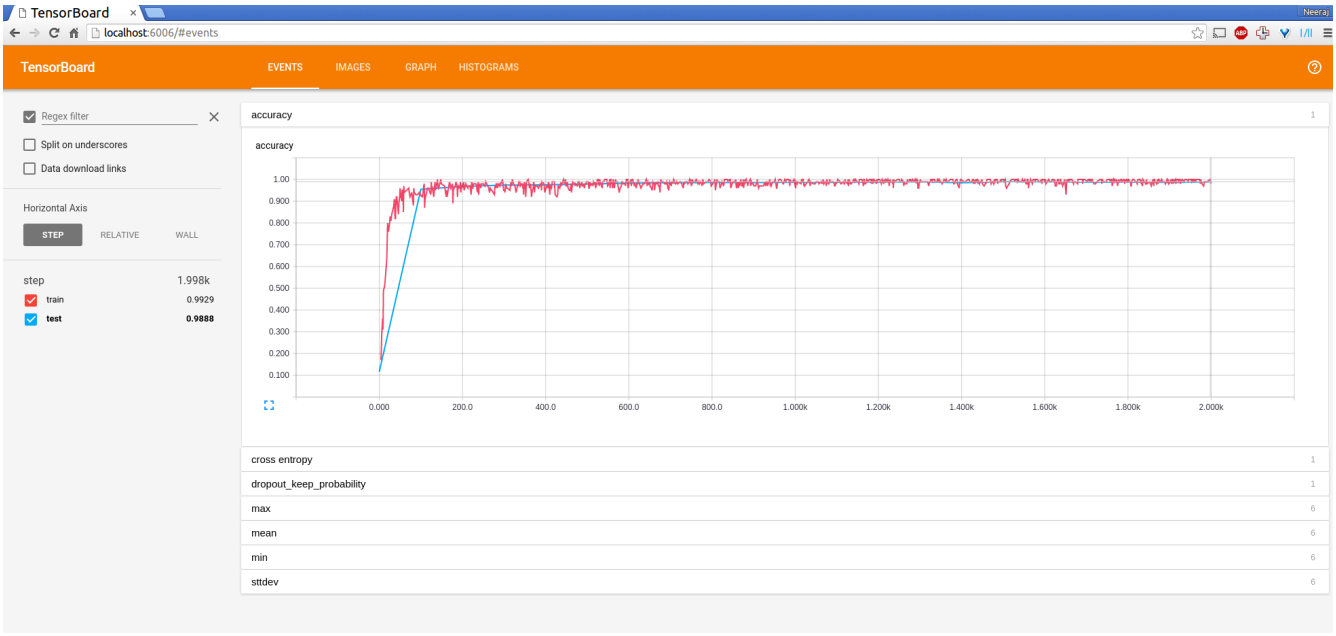
Neeraj Asthana
nasthan2
CS 498: Applied Machine Learning
Homework 8 Report

Plots are more clear in the Screenshots folder included

Part 1:

Accuracy: .9889
Steps: 2000
Plot:



Part 2:

Motivation: I wanted to explore the of adding another convolutional layer to the network while decreasing the size of each convolutional layer. I also wanted to explore what would happen if I increased the window size.

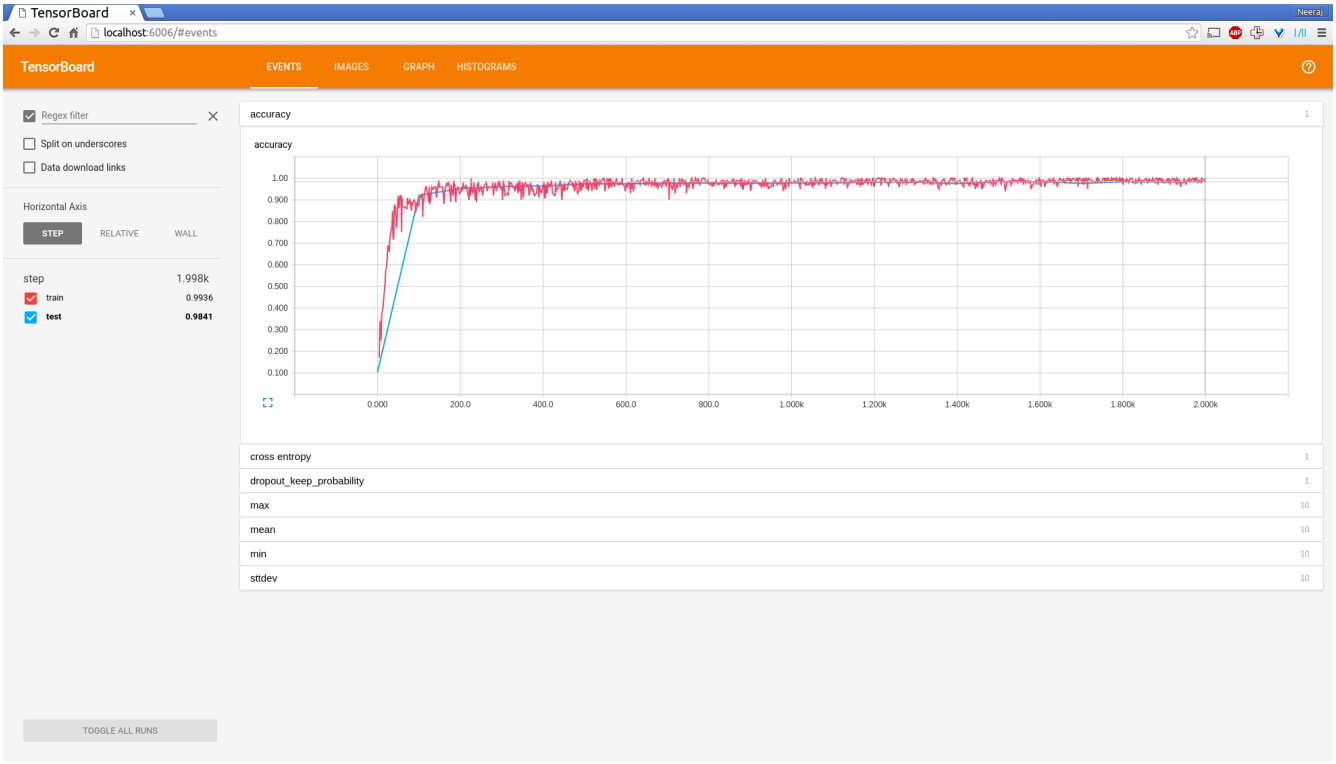
Description: I first use a window size of 10x10 instead of the provided 5x5. I also changed the convolutional layer to go from a 1x8 layer to a 8x16 layer to a third and final 16x32 layer (each with a 10x10 window). Afterwards I kept max pooling, the dropout layer, and the dense layer from the previous neural net.

Accuracy: .9841

Steps: 2000

Performance: This neural network had decent performance, however it did not reach an accuracy as high as the neural net before. However, I believe if I ran this network for more steps, it would perform much better.

Plot:

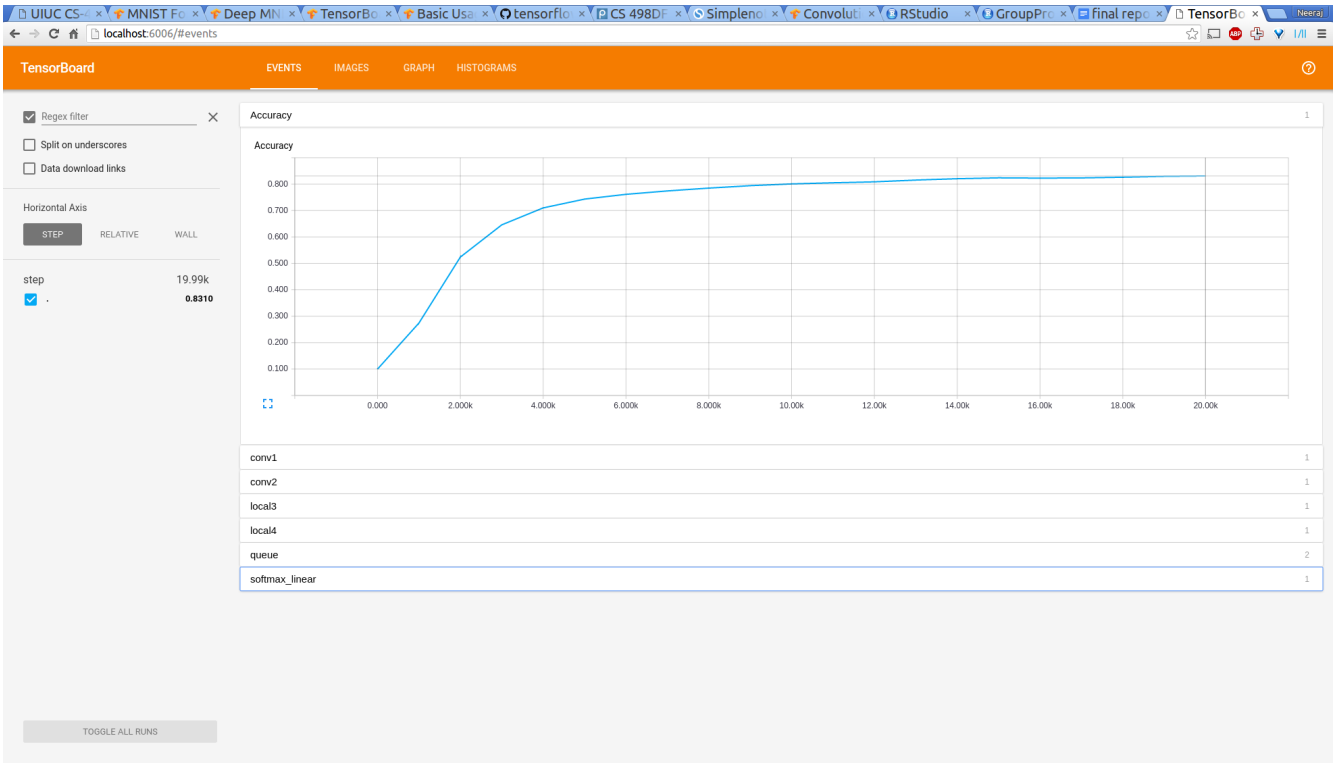


Part 3:

Accuracy: .8310

Steps: 20000

Plot:



Part 4:

Motivation: In general, I wanted to explore the effects of adding additional layers to the network while reducing the size of other layers for local layers (not convolutional layers). I also wanted to explore what would happen if parameters such as the learning rate and epoch length are changed.

Description: I kept the initial part of the neural network the same. However, I changed the local layers to go from a 192x96 layer to a 96x48 layer to a third and final 16x32 layer (each having a 5x5 window). I also changed the Epoch Length from 350 to 300 (increasing the number of time I change the learning rate). I also increased the learning rate from .1 to .2 while also decreasing the learning rate decay from .1 to .075.

Accuracy: .09968

Steps: 10000

Performance: All of these changes had significant dips in the performance and I have many regrets by changing the neural network so much. After around 3000 steps, the loss of the model plateaus at around 2.30 and never improves from the value. The accuracy also does not improve at all for inexplicable reasons. There is no increase in the test accuracy as there was in the previous example and accuracy actually seems to decrease over time. I had higher hopes for this neural network, however, it took lots of time to train (longer than the previous example) and did not perform well at all. If I had time I would have tried a different model, however this was the best performance I was able to receive with such a bad network. Perhaps if I trained this network further it would have better performance but that is unknown.

Plot: