**HW9 Tensorflow**

**CS498 Applied Machine Learning**

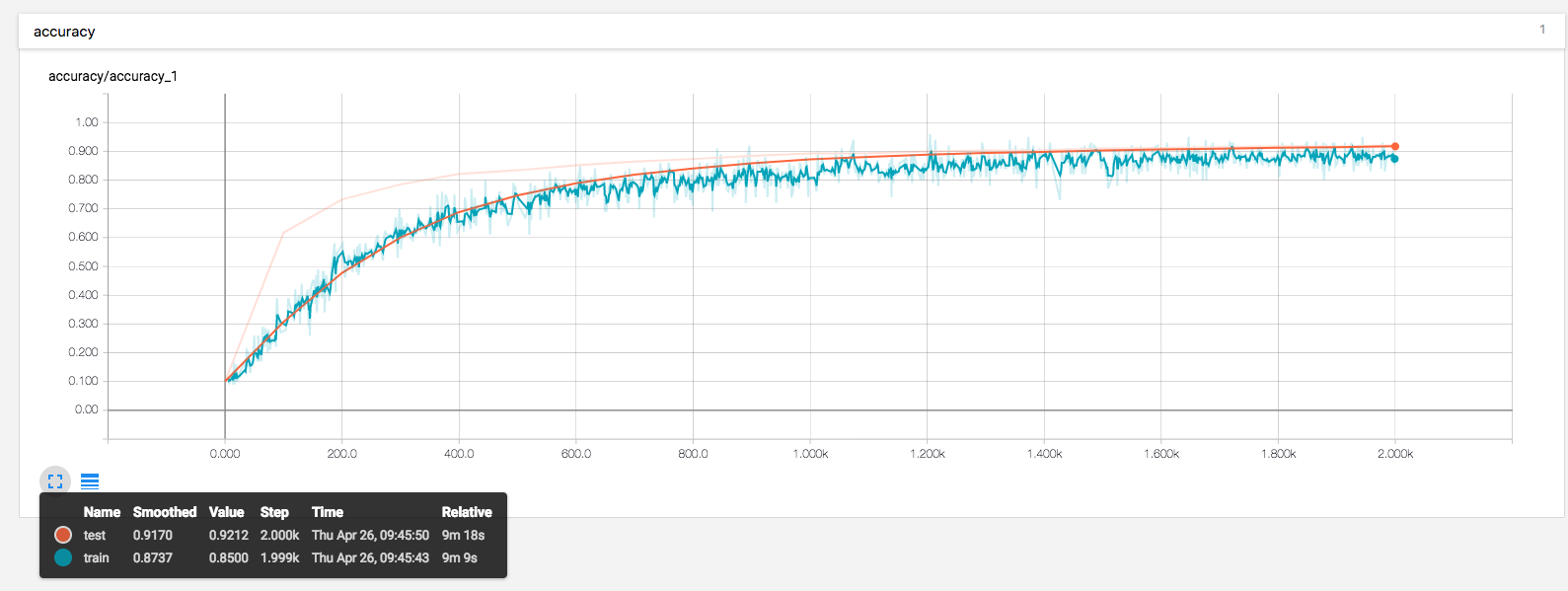
**Group Members**

* Viraj Bhalala
* Jonathan Chang
* Faisal Hoda

**Part 1**

Go through the MNIST tutorial [here.](https://www.tensorflow.org/tutorials/layers) You may find the tensorboard tutorial [here.](https://www.tensorflow.org/programmers_guide/summaries_and_tensorboard)helpful. Insert appropriate lines of code into the tensorflow example to log the accuracy on tensorboard every 100 batches, for at least 2000 batches. You should screen capture the accuracy graph from tensorboard, and submit this.

*In part 1, we first followed CNN MNIST tutorial and Tensorboard tutorial that was suggested by our professor. We found out that professor’s suggested CNN MNIST tutorial was using newer tensorflow API which was not using TF sessions to log train and test set accuracy in tensor board. In addition, it was creating neural net layers in more simpler way by using tf.layers. We followed tensorboard tutorial code to learn how to log train and test accuracy and rewrote MNIST CNN tutorial code with same neural net architecture and parameters so that we can easily use tensorflow session with it to log our train and test accuracy. Below you can see the plot for 2000 steps.*

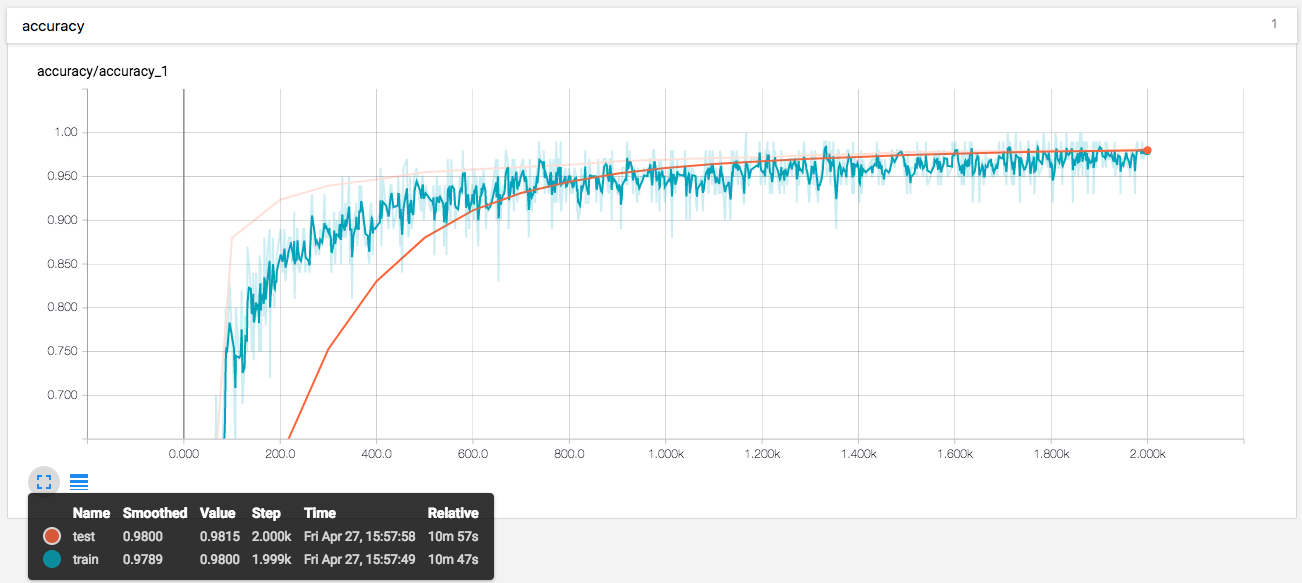


**Part 2**

Modify the architecture that is offered in the MNIST tutorial to get the best accuracy you can. I made three convolutional layers of smaller depth (i.e. the 32 went to 8), dropped the max pooling, and used three layers. Submit a screen capture of tensorboard graphs of accuracy. We will make it possible for people to compare graphs anonymously. This is to allow people to show off how well their model is doing and see how others are doing; it's not required, and won't be graded, but it's been a source of fun and excitement in the past.

* Here is how to submit graphs for comparision. Go to this [Google form](https://goo.gl/forms/EIusPh0H7aj3DRPR2) and supply what it asks for.
* On that page, you'll see a link to results; press that, and you'll get a collection of tensorboard graphs that have been submitted

*For Part 2, we tried many tuning techniques of adding and removing layers, changing input and output channel sizes of different layers, and chaining optimizer. The one approach that we found to be best working was changing optimizer from SGD to AdamOptimzer with 0.0001 learning rate while leaving all layers to be the same.*

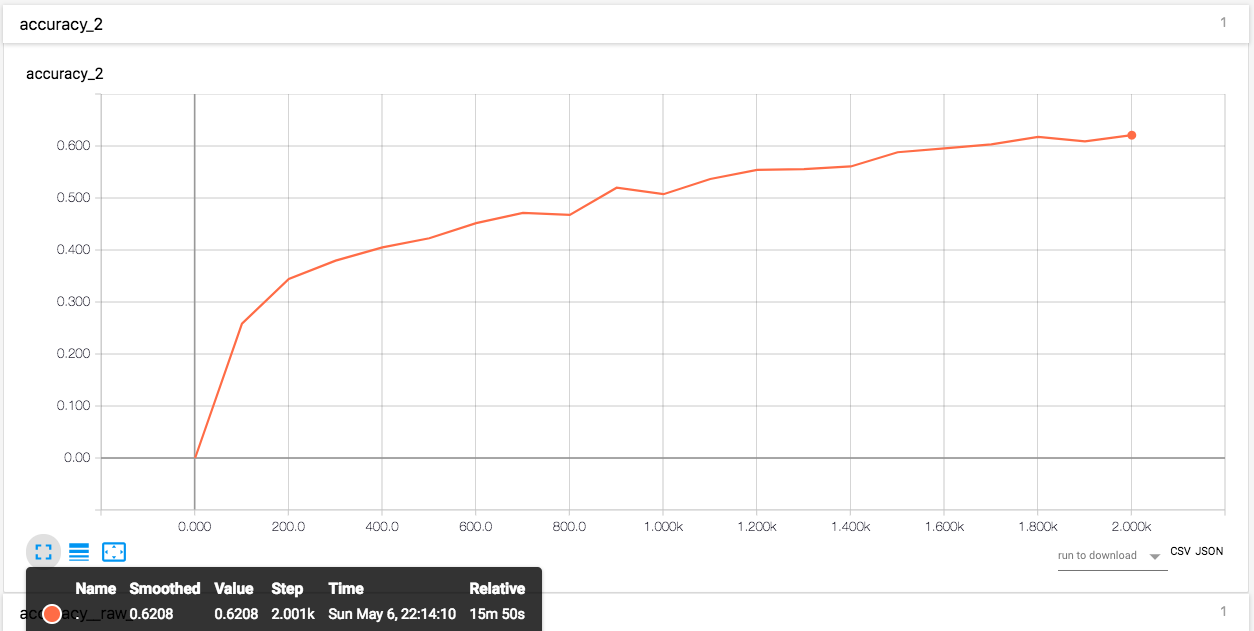


**Part 3**

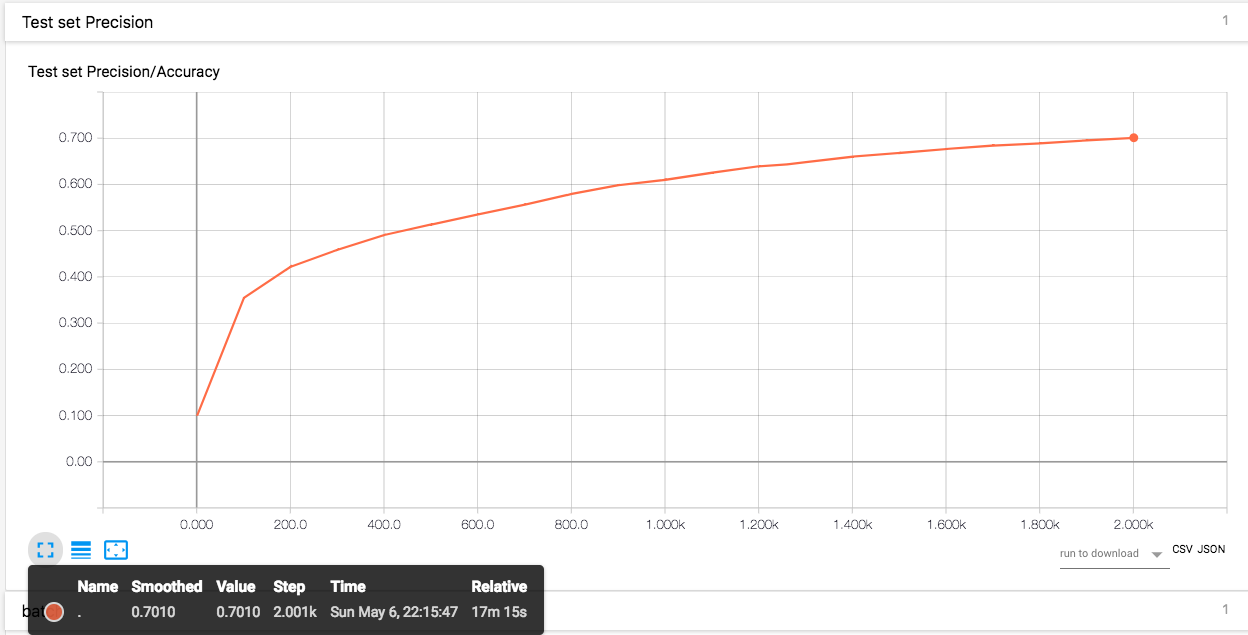
Go through the CIFAR-10 tutorial [here](https://www.tensorflow.org/tutorials/deep_cnn), and ensure you can run the code. Note the warning at the top: "This tutorial is intended for advanced users of TensorFlow and assumes expertise and experience in machine learning." Enjoy the sense that you are one of these. Finally, insert appropriate lines of code into the tensorflow example to log the accuracy on tensorboard every 100 batches, for at least 2000 batches. You should screen capture the accuracy graph from tensorboard and submit this.

*In part 3, we had to write our accuracy function to log train accuracy, and we had to create checkpoint hooker to create checkpoint at every 100 steps in order to use the current model weights with current step in order to predict and calculate accuracy of test set at every 100 steps.*

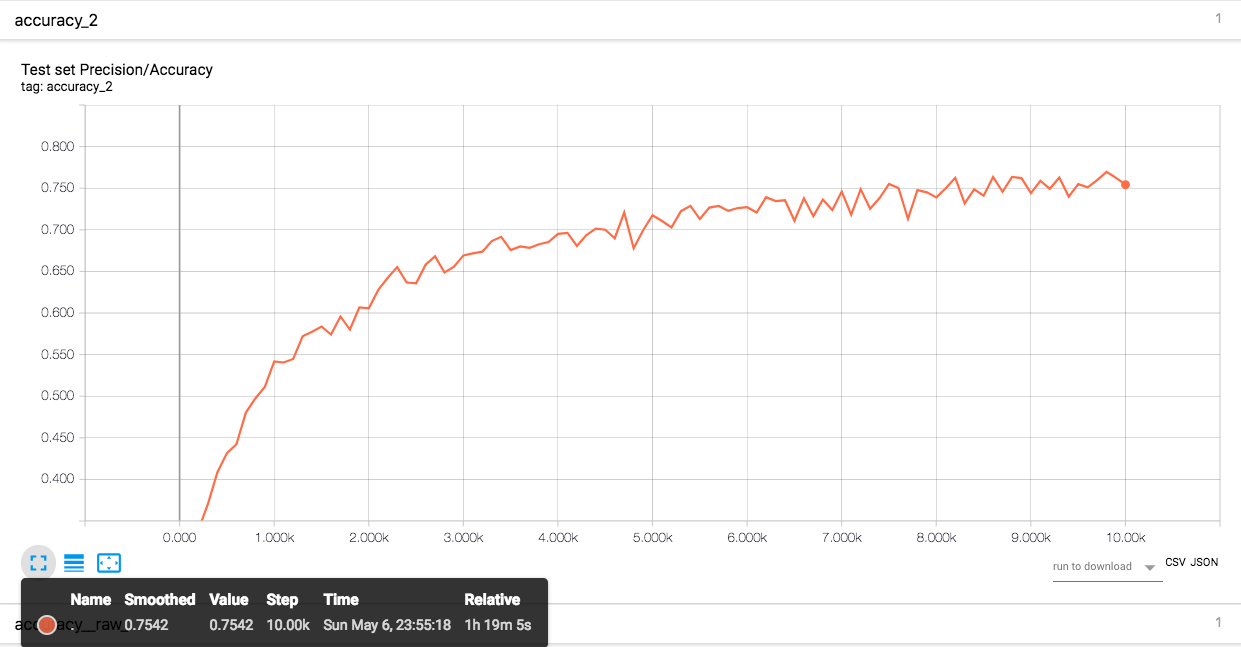
Tensorboard’s train set accuracy plot for 2001 batches



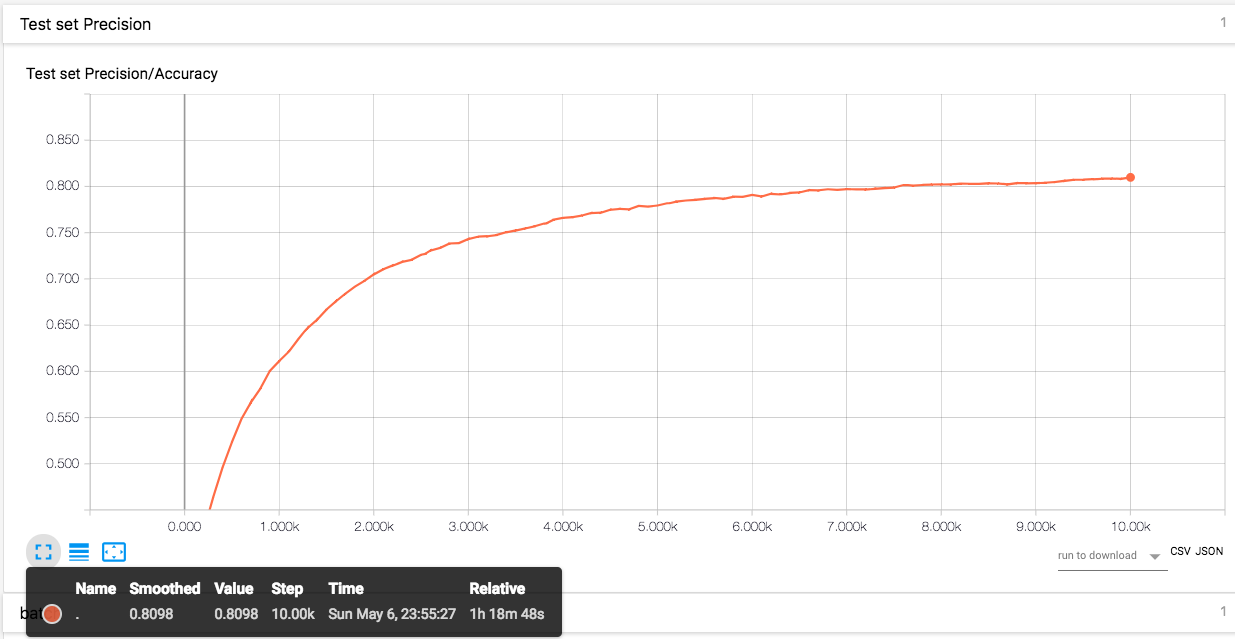
Tensorboard’s test set accuracy plot for 2001 batches



Tensorboard’s train set accuracy plot for 10000 batches



Tensorboard’s test set accuracy plot for 10000 batches



**Part 4**

If you are taking the 4-hour (online) version of this class, this is **required**. For all others, this part is **optional**. Modify the architecture that is offered in the CIFAR-10 tutorial to get the best accuracy you can. Anything better than about 93.5% will be comparable with current research. Be aware that people with bigger computers will likely do better at this exercise (so I won't make grades depend on accuracy). Submit a screen capture of tensorboard graphs of accuracy. We will make it possible for people to compare these graphs anonymously.This is to allow people to show off how well their model is doing, and see how others are doing; it's not required, and won't be graded, but it's been a source of fun and excitement in the past.

* Here is how to submit graphs for comparision. Go to this [Google form](https://goo.gl/forms/ij3qm5hx1gu0E8lE2) and supply what it asks for.
* On that page, you'll see a link to results; press that, and you'll get a collection of tensorboard graphs that have been submitted. There may be two junk PDF's submitted by DAF to test the system; you can ignore those.

*In part 4, we first tried to add extra one and two convolutional layers along with pooling and normalizing with them to the current CNN architecture with SGD. However, this dropped our accuracy down by 3-5%. We tried these layers with different number of input and output channels as well. We then followed other students’ recommendation on slack and piazza about changing learning rate and batch size for optimizer. However, this only gave us tiny improvement. Similarly, to part 2, we then tried using AdamOptimzer with 0.001 learning rate and learning rate decay factor of 0.001 at every 750 epochs and with batch size of 256. For the CNN layers, we also added one extra CNN layer along with norm and max pooling with input channel of 64 and output channel of 128 after layer 1 and before layer 2 layer. This parameter helped us to improve accuracy of our model and also help us to converge accuracy of the model at better value than that of part 3.*