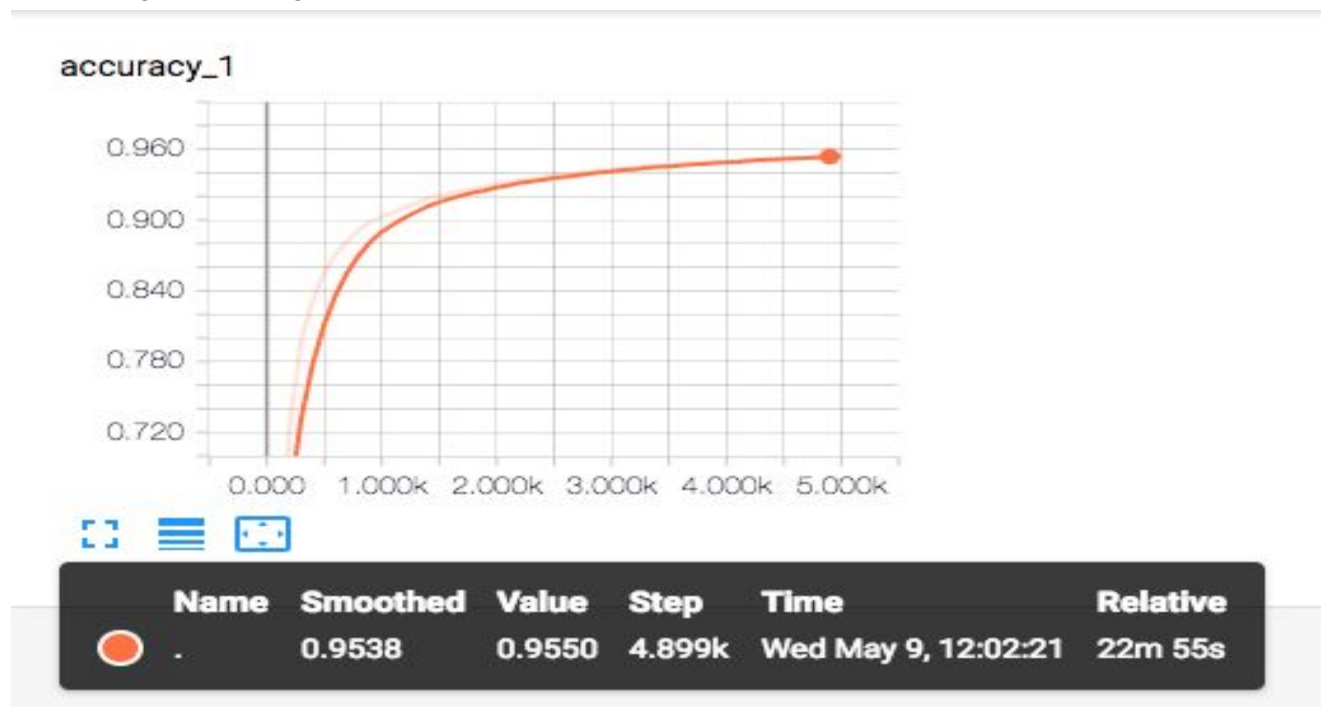


1. Go through the MNIST tutorial [here](#). You may find the tensorboard tutorial [here](#) helpful. Insert appropriate lines of code into the tensorflow example to log the accuracy on tensorboard every 100 batches, for at least 2000 batches. You should screen capture the accuracy graph from tensorboard,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

We tried the tutorial from https://www.tensorflow.org/versions/r1.1/get_started/mnist/pros. We matched the the above example code template to represent the baseline architecture suggested in the homework link. Our code is *original_mnist.py*

By utilizing TensorBoard, we visualized the learning process. We were able to achieve the final accuracy of 95.5% on test data after 5000 batches. Below is our result on TensorBoard for test data during the training process.



2. Modify the architecture that is offered in the MNIST tutorial to get the best accuracy you can.

Code is *mnist_modified.py*

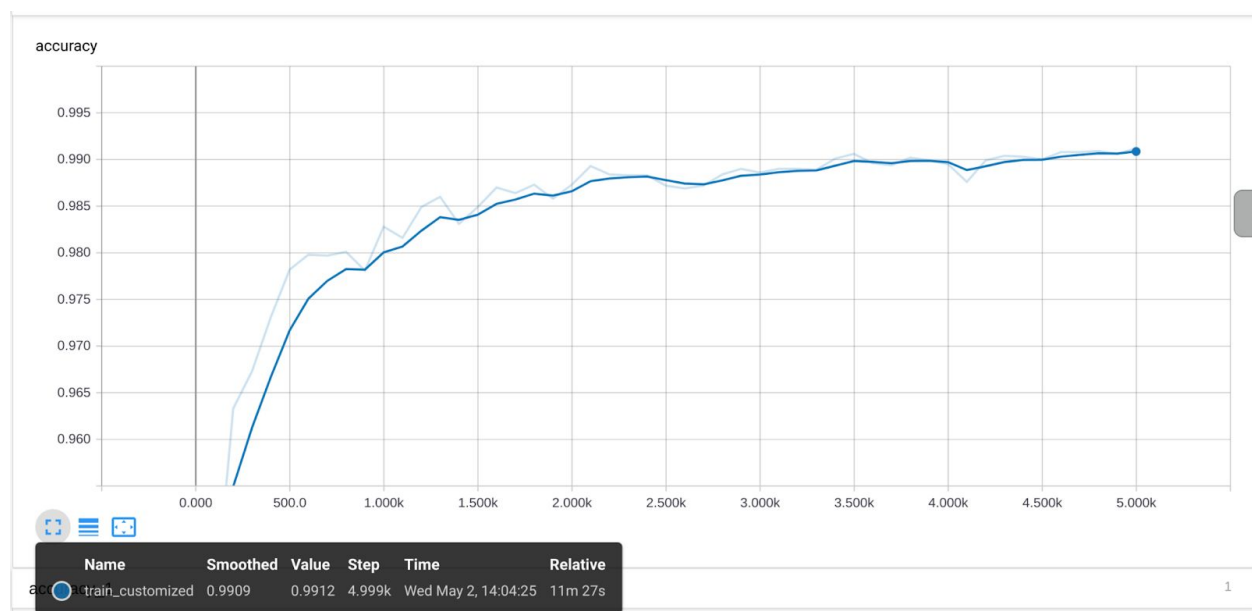
The following modifications were made to the base architecture used in previous section. 3 convolutional layers and 2 fully connected layers were used. Max pooling was removed. The

number of filters at each convolutional layer are 6,12,24 respectively (first,second,third). The width of every filter is 6x6, 5x5, 4x4 (first,second,third). Additionally,the stride of second and third convolutional layers were set to 2. The first fully connected layer has 200 neurons and the second fully connected layer has 10 neurons. Adamoptimizer with exponentially decaying learning rate from 0.03 to 0.0001 was used in the architecture. Biases were added to architecture at every convolutional layer. The keep probability of final layers is set to 1.0 (i.e no dropout)

We were able to achieve beyond 99% after 3000 iterations and the best test accuracy is 99.12% after 5000 iterations. The accuracy of test data during training process is captured below

The architecture was inspired from below tutorial :

<https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/#0>



3. Go through the CIFAR-10 tutorial [here](#), 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.

Codes are : *cifar10*.py*

We added checkpoints in the training script to save model parameters every 100 steps and the evaluation script looks at the checkpoint file every few seconds. We ran evaluation and training script simultaneously. By this way, evaluation script looks at latest checkpoint and uses the

model parameters to calculate test accuracy during training process. We used TA endorsed piazza post @1636 as a reference for logging accuracy which suggested that the way we log results is acceptable.

Below is the screenshot of tensorboard. The accuracy was around 84.2%, after 20K iterations. The accuracy of test data during training process is captured below

