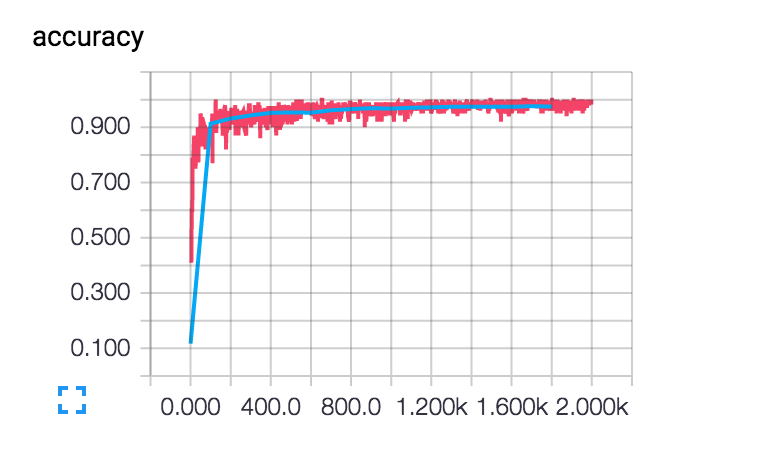
**CS 498 HW 8 (Final Homework)**

**Tlu16, kakilai2**

**Part 1**

Accuracy plot from Tensorboard:

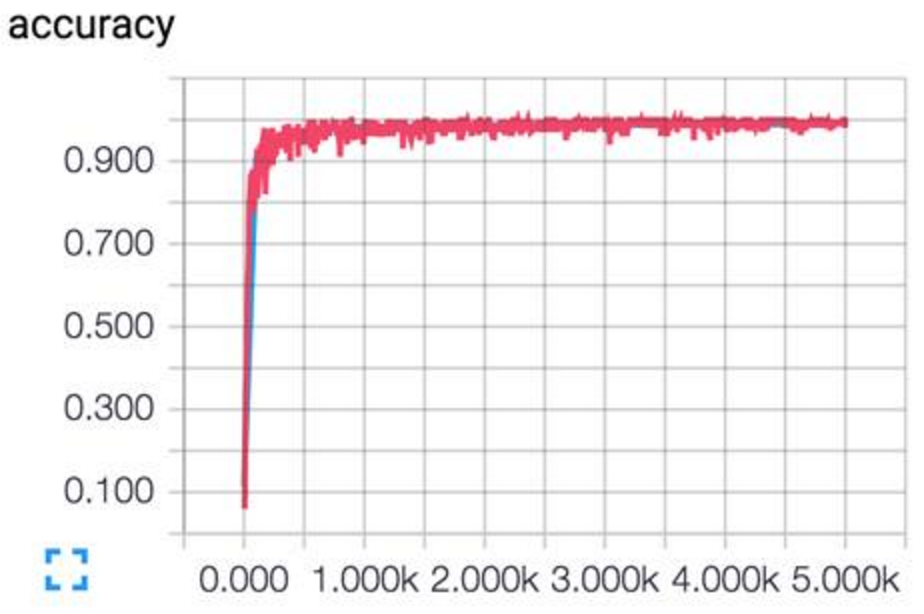


Number of Batches: 2000

Testing Accuracy: 0.9753

**Part 2**

Accuracy plot from Tensorboard:



Number of Batches: 5000

Testing Accuracy: 0.9851 (after running 2000 batches)

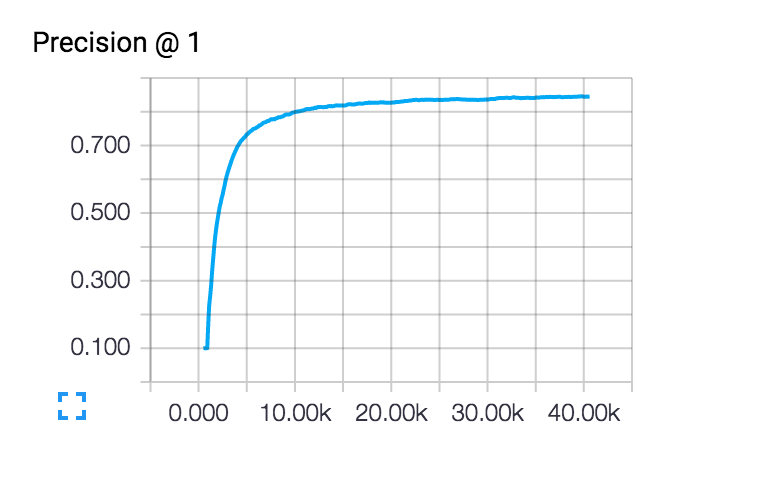
0.9885 (after running 5000 batches)

Modification for part 2:

We have change to 2 convolutional layers, outputting 8 channels for first layer, output 16 channels for second layer. Moreover, we have added dropout layers after each convolution.

**Part 3**

Accuracy plot from Tensorboard:



Number of Batches: 40000

Accuracy (Precision on evaluation set): 0.845

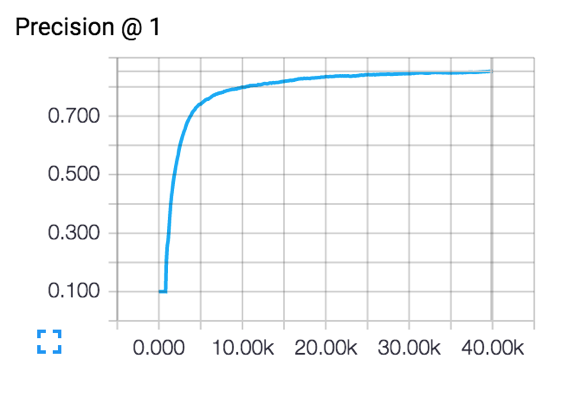
We modified part of the code ('eval\_interval\_secs') in cifar10\_eval.py to log the accuracy for every 60 seconds, which is approximately equivalent to every 100 batches of output.

**Part 4**:

After various runs based on different modifications, we have concluded the following modification generate the best result (most comparable to the result from tutorial) based on the limited running time. The modified architecture is described as below: (refer to attached part4.py)

After each of the 2 convolution steps, we added dropout layer. Besides, we use filter size of 7x7 instead of 5x5 as the filter size for the convolution steps. Similar to part 3, we modified part of the code('eval\_interval\_secs') in cifar10\_eval.py to log the accuracy for every 60 seconds, which is approximately equivalent to every 100 batches of output.

Accuracy plot from Tensorboard:



Number of Batches: 40000

Accuracy (Precision on evaluation set): 0.8533

We have tried different kinds of modifications, for example adding 3 convolution layers, and found that most of the time the performance is worsen. Therefore, our modification focus on 2 layers, summarized as below:

**Summary of modification on architecture and respective performance:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Modification (based on the architecture in tutorial) | Number of batches | Accuracy  (Precision) | Runtime |
| Based model  (Reference for comparison) | 2 layers of convolution,  64 channels,  max pooling  3x3 filters  no dropout | 5000 | 0.733 | 120 mins |
| 15000 | 0.819 | 270 mins |
| 28000 | 0.835 | 450 mins |
| 40000 | 0.845 | 570 mins |
| Model 1 | 2 layers of convolution, average pooling | 28000 | 0.819 | 200 mins |
| Model 2 | 2 layers, 7x7 filters, dropout rate = 0.6 | 5000 | 0.628 | 110 mins |
| Model 3 | 2 layers of convolution, change  remove the second layer of max pooling and add the dropout with rate = 0.6 | 5000 | 0.663 | 85 mins |
| Model 4 | 2 layers of convolution,  dropout rate = 0.7 | 15000 | 0.792 | 500 mins |
| **Model 5**  **(Best)** | **2 layers of convolution, dropout rate = 0.6, filter size 7x7** | **40000** | **0.853** | **700 mins** |

However, due to the limitation of time and machine capability, comparison of the model performance could only be based on limited number of batches run, it is believed that the higher number of batches (over 1000K) would lead to a fairer comparison.