CS 498 DAF: Final Report

hkumar5 | hagarwa3 | tagarwl2

**Explanation for MNIST:**

The baseline model is just as in the tutorial provided by Tensor Flow. It gave us an accuracy of 99.07%. We then tried a variety of strategies to get better results:

1. We used opencv to deskew and straighten the images as a means of preprocessing. We then ran the standard 2 layer CNN. This gave us our best accuracy of 99.24%.

**(code: deskewed.py)**

1. We also tried 3 layers with no pooling. It didn’t lead to as impressive performance as above.

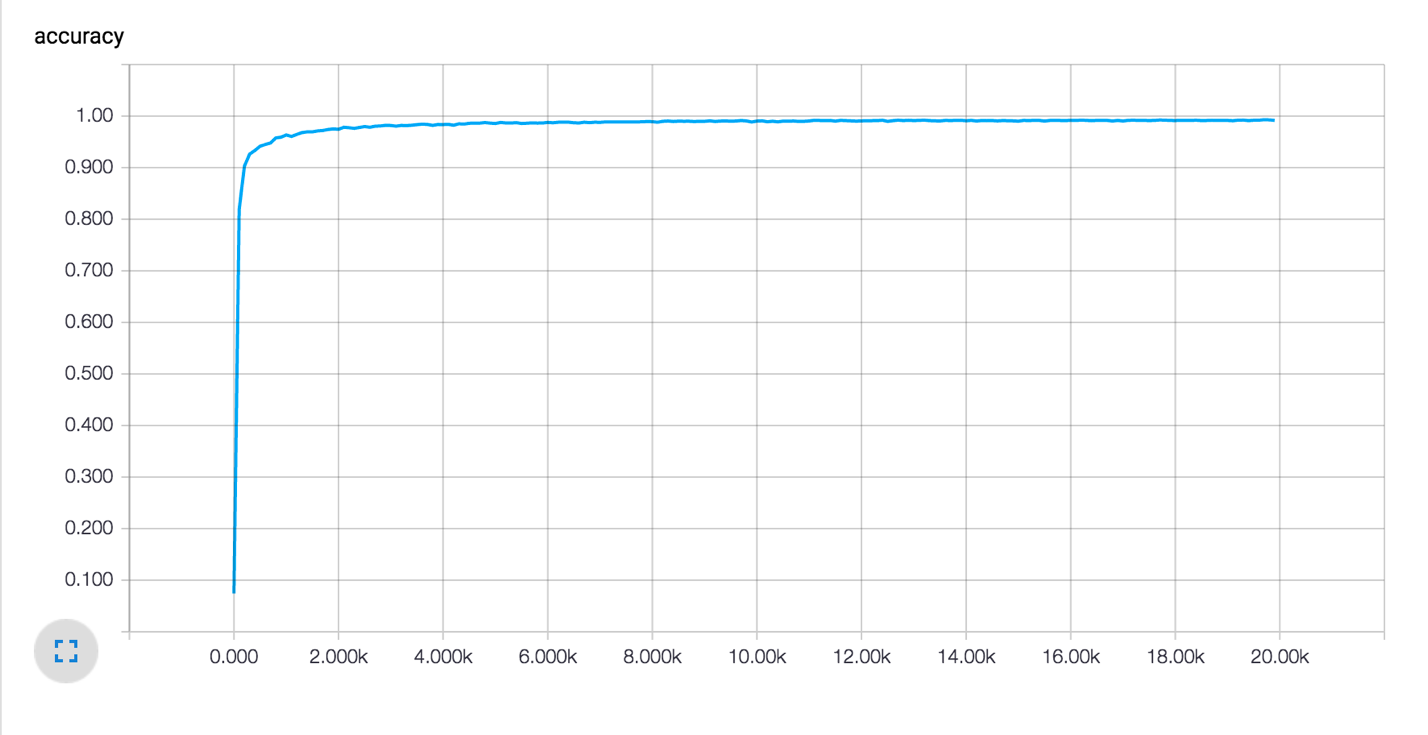
**(code: 3layers\_nopool.py)**

1. We also tried training 4 to 5 CNNs on randomly distorted images and averaging their softmax output. This is based on a paper [here](http://people.idsia.ch/~juergen/icdar2011a.pdf). But our instance ran out of graphic memory trying to do this. So results are uncertain.

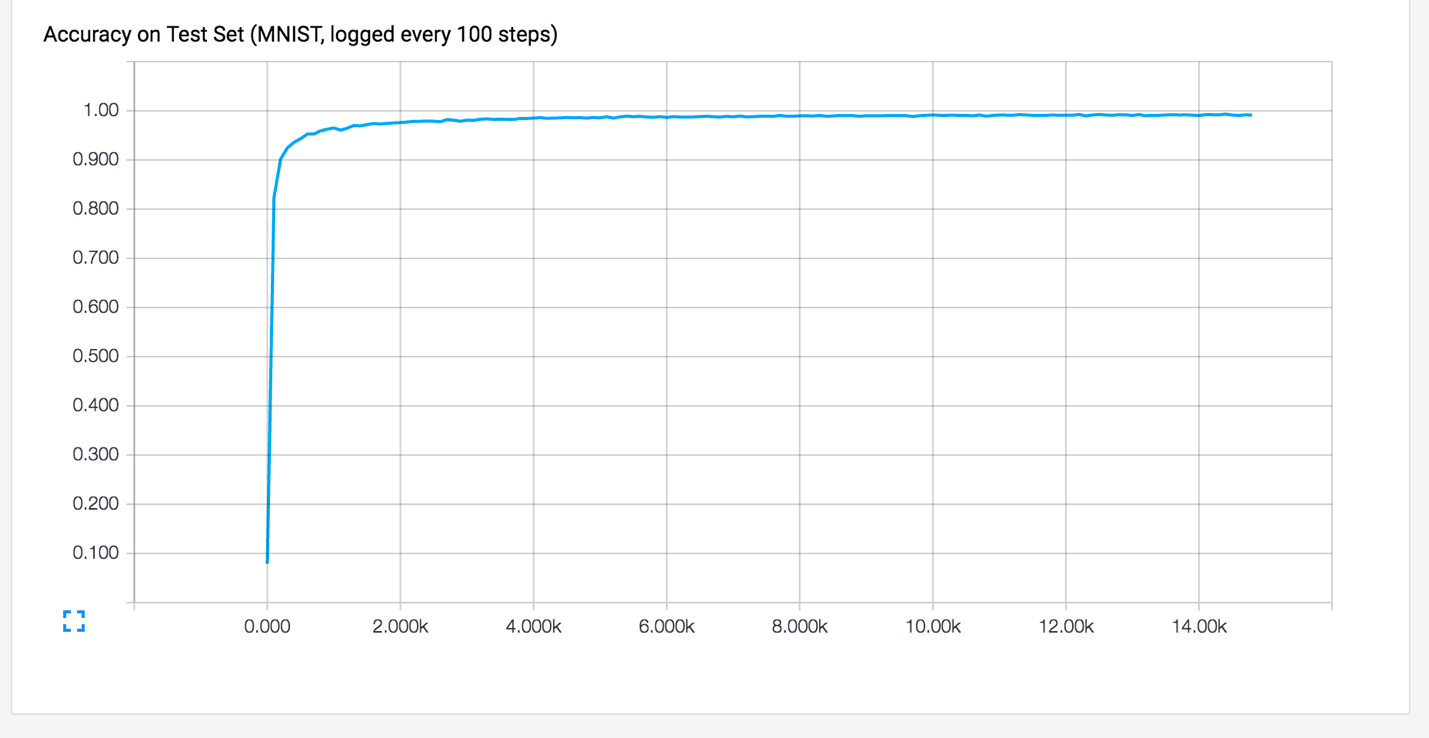
**(code: committee.py)**

Here are the test accuracy graphs for both cases. The images are also included as separate files.

**Baseline Test Accuracy for MNIST:**

****

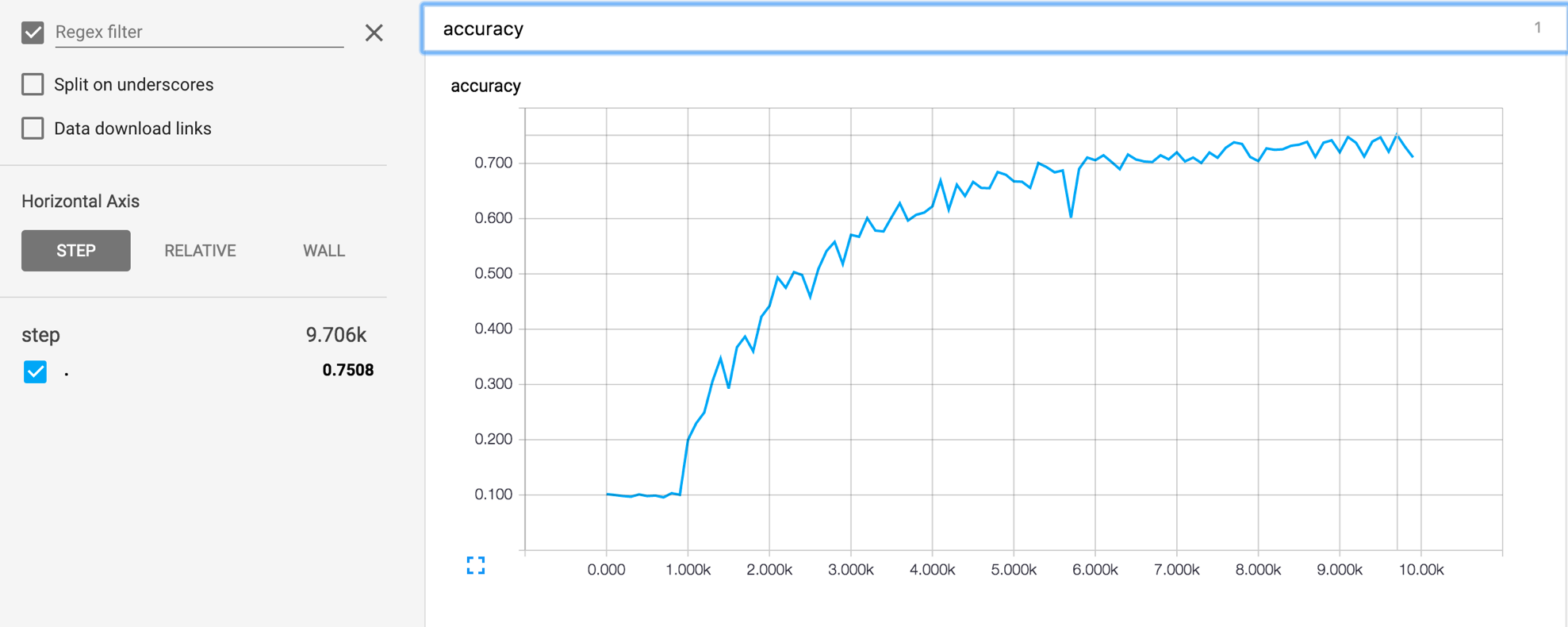
**Improved Test Accuracy for MNIST:**

****

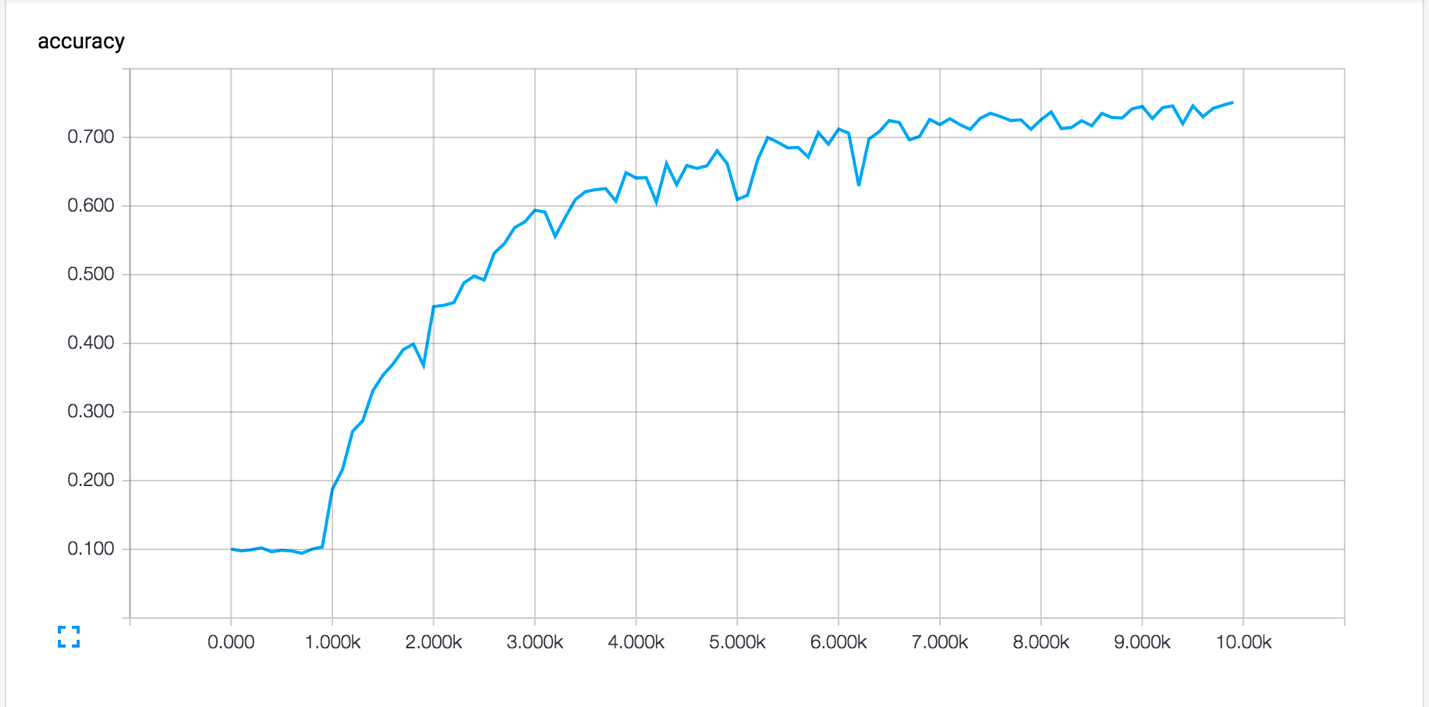
**Explanation for CIFAR:**

The baseline model is just as in the tutorial provided by Tensor Flow. It gave us an accuracy of 80.3%. We then were inspired by the architecture explained [here](http://cs231n.github.io/convolutional-networks/). Specifically, an arrangement of CONV -> RELU -> CONV -> RELU -> POOL. We removed normalizations to increase speed but it might have cost us accuracy. We believe the modified model would have produced better accuracy given a longer training time.

**Baseline Test Accuracy for CIFAR:**

****

**Test Accuracy for CIFAR with Our Modifications:**

****