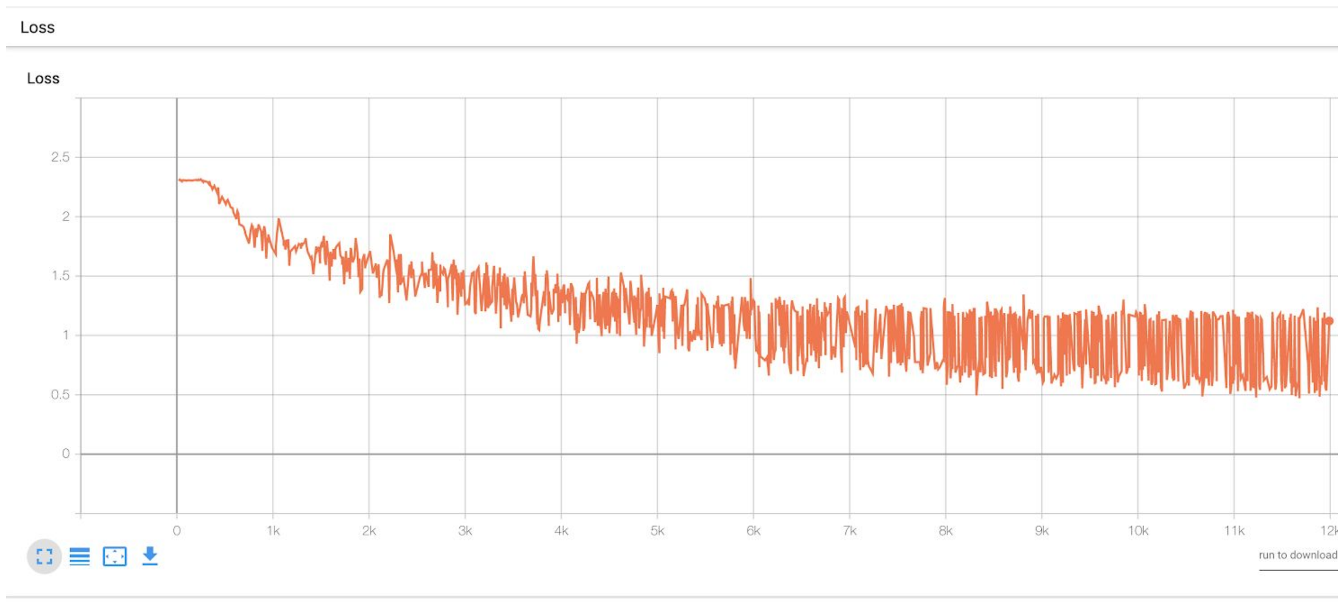
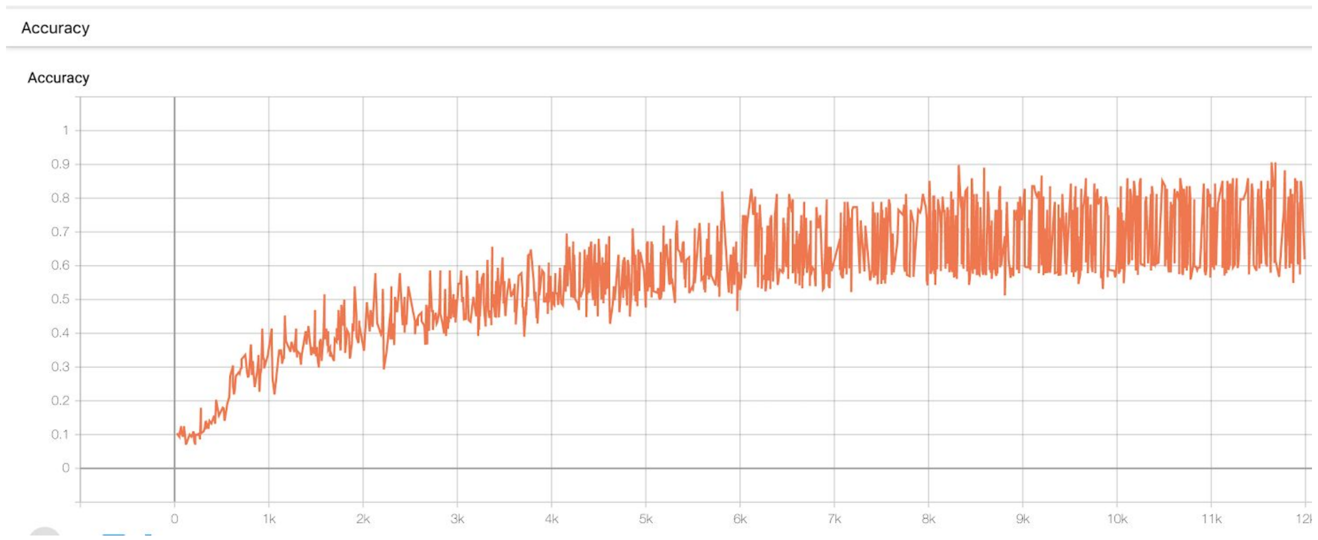
COMP 576 Assignment 2

Luis Clague Fall 2019

Contents

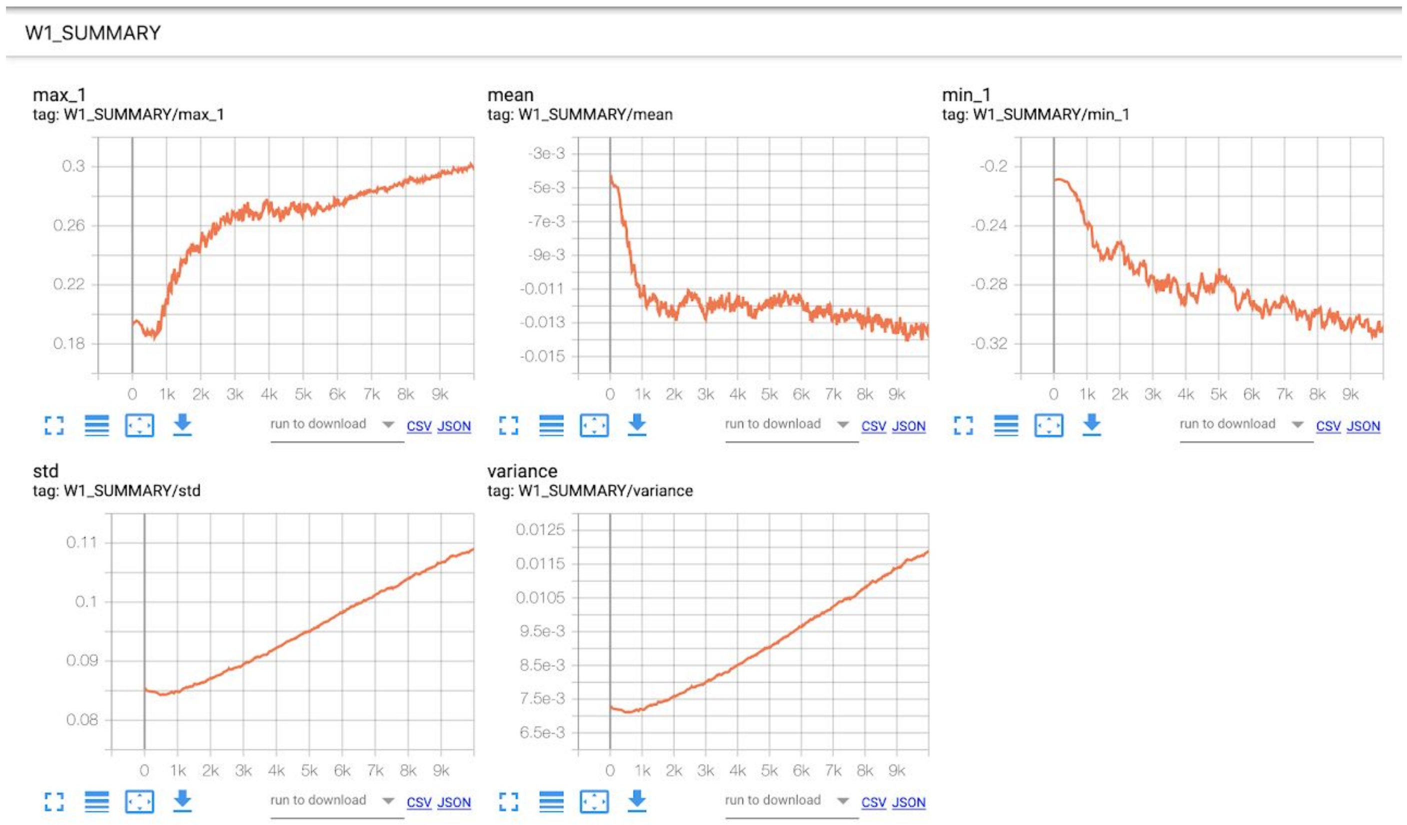
Visualizing a CNN with CIFAR10



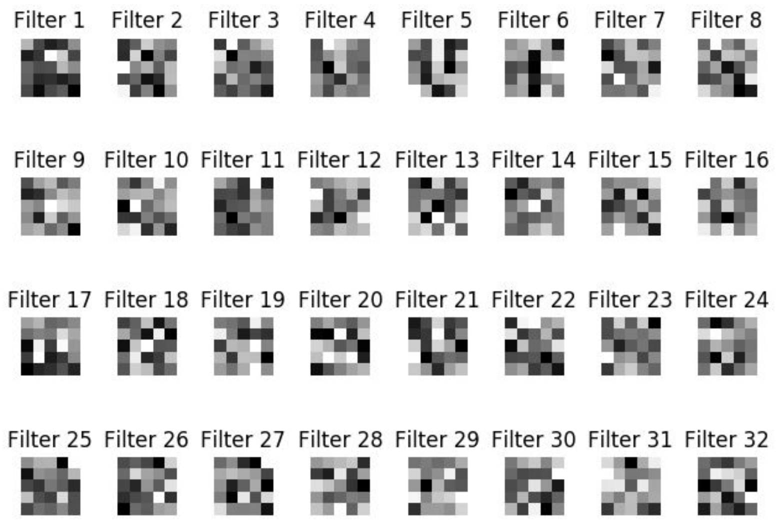


Some Scalar stats:

First Layer Statistics



Visualizing the layer's weights:



Convolutional layer 1

Visualizing and Understanding Convolutional Networks

Summary of “Visualizing and Understanding Convolutional Networks”

The paper’s main focus is to explain the concept of reverse-engineering a previously successful ImageNet conv net in order to gain insight on the underlying characteristics that make each layer successful. The novel technique involves a “Deconvolutional Network”, which the paper explains is made up of unpooling, rectification, and transposed filtering stages that allowed the authors to reverse engineer deep ImageNet models. The beauty behind this approach is that, after training a novel ImageNet model using this technique, the model was extremely generalizable to other datasets.

Build and Train an RNN on MNIST

Number of Hidden Layers

- 128 - Model doesn’t achieve a high enough level of testing accuracy
- 256 - This proved to be the sweet spot between achieving ~.96 testing accuracy and a relatively smooth curve (such that the model doesn’t overfit)
- 1024 - This upper bound resulted in a very jagged set of training curves indicating that the model tended to overfit for the provided training data set.

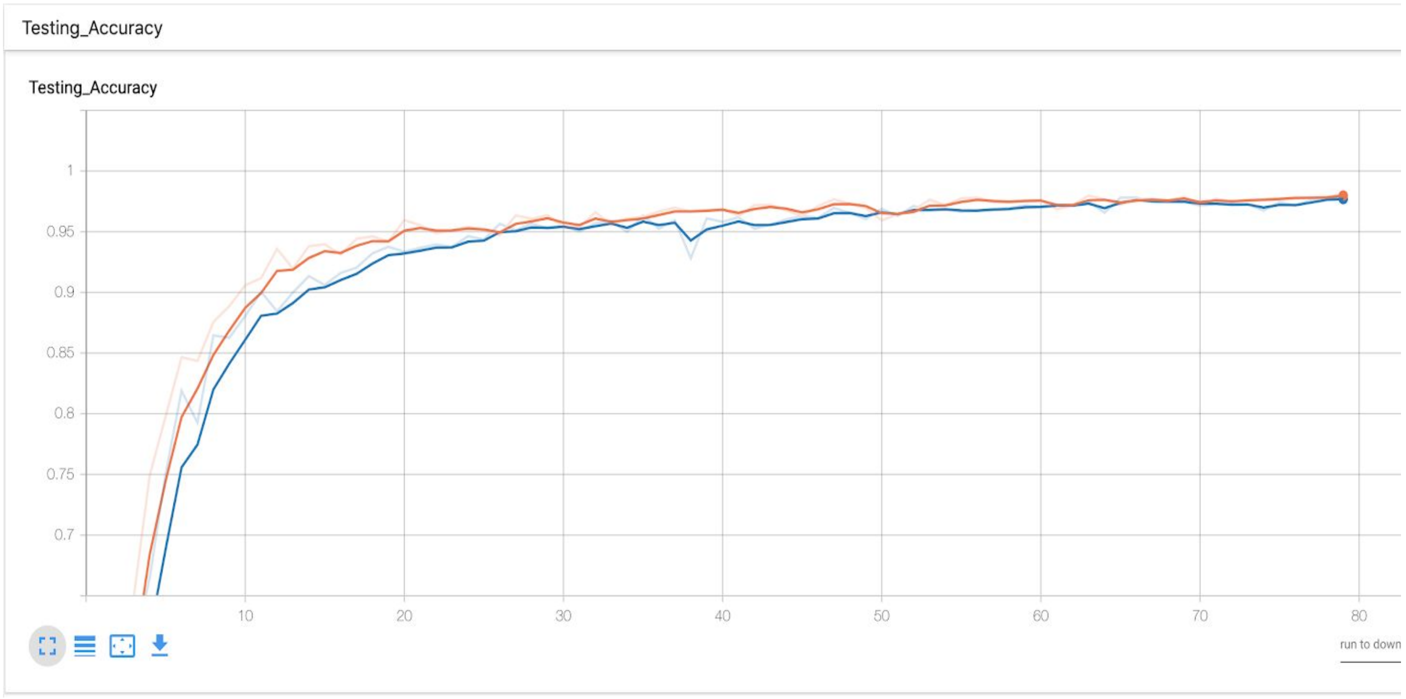
GRU vs LSTM

Overall both GRU and LSTM learned relatively quickly, with very high testing and training accuracies.

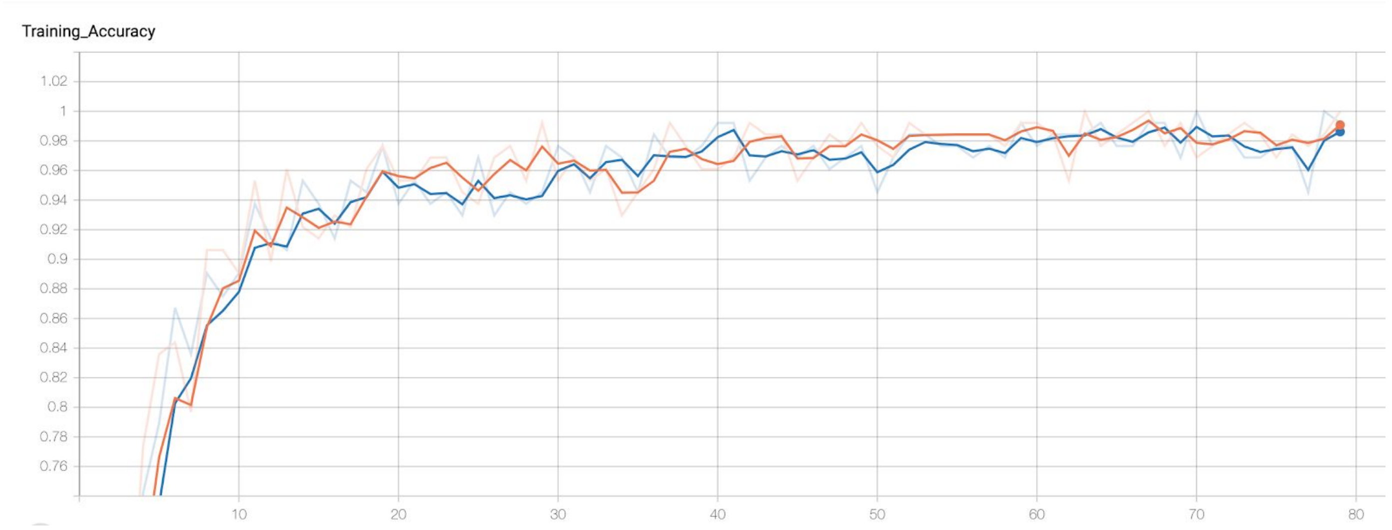
Some prominent features:

- LSTM is probably overfitting since it is a lot more jagged than GRU (both were smoothed at a .5 factor)
- The GRU (Orange line) was always above its LSTM counterpart in testing accuracy and of course below it in loss accuracy, indicating a very very slightly better performance, but not significantly

256 hidden layer GRU (Orange) vs 256 hidden layer LSTM (Blue)
Testing Accuracy across 80 epochs

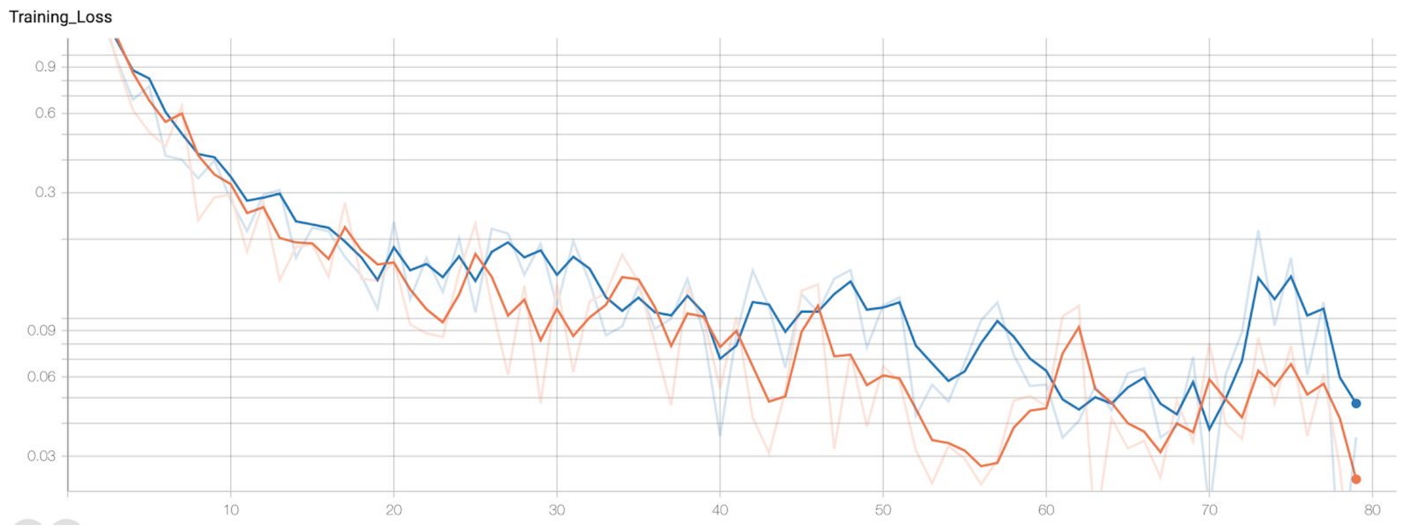


256 hidden layer GRU (Orange) vs 256 hidden layer LSTM (Blue)
Training Accuracy across 80 epochs



Training Accuracy across 80 epochs

256 hidden layer GRU (Orange) vs 256 hidden layer LSTM (Blue)
Training Loss across 80 epochs



RNN vs CNN

The RNN proved to be a lot more accurate than my previous CNN, but this is only after 10 times more training iterations (10,000 vs 100,000 respectively). Overall, it seems that the RNN was severely overfitting for high testing accuracy in this case, but both nets take a very large number of iterations to train before they're actually useful.