

CSCI 5525 Machine Learning HW3

Problem2

The test accuracy of logistic regression is around 91%. Fig. 1, 2 show the training process and graph.

Print Messages

```
.....  
Average loss epoch 15: 0.3735628225308754  
Average loss epoch 16: 0.3699406019318632  
Average loss epoch 17: 0.36766438384156125  
Average loss epoch 18: 0.3615034160427836  
Average loss epoch 19: 0.3608970091893123  
Average loss epoch 20: 0.3553532210541216  
Average loss epoch 21: 0.3534583798794202  
Average loss epoch 22: 0.3510596220726733  
Average loss epoch 23: 0.34962106265110293  
Average loss epoch 24: 0.34648945177351675  
Optimization Finished!  
Total time: 15.049790620803833 seconds  
Accuracy on test data: 0.9121
```

Tensorboard Visualization

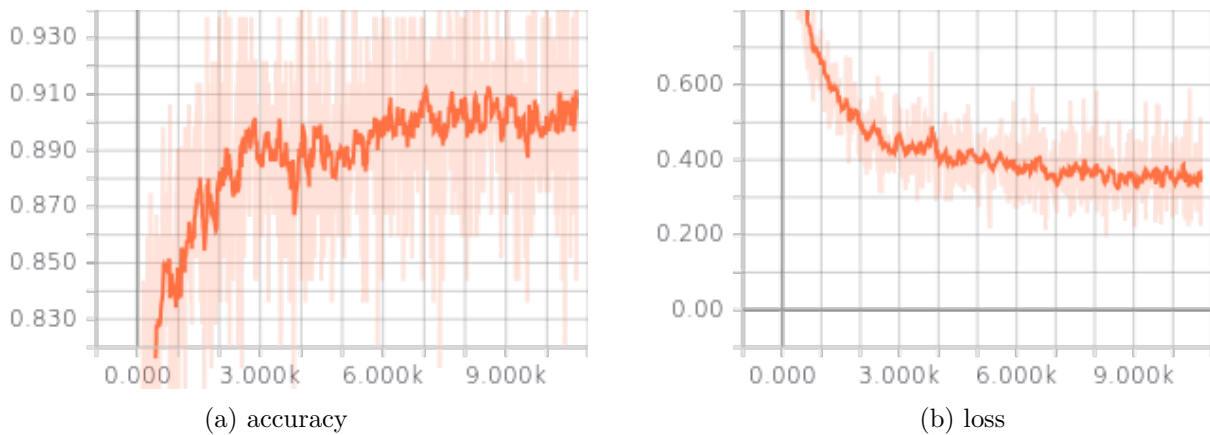


Figure 1: Logistic Regression Tensorboard Scalars

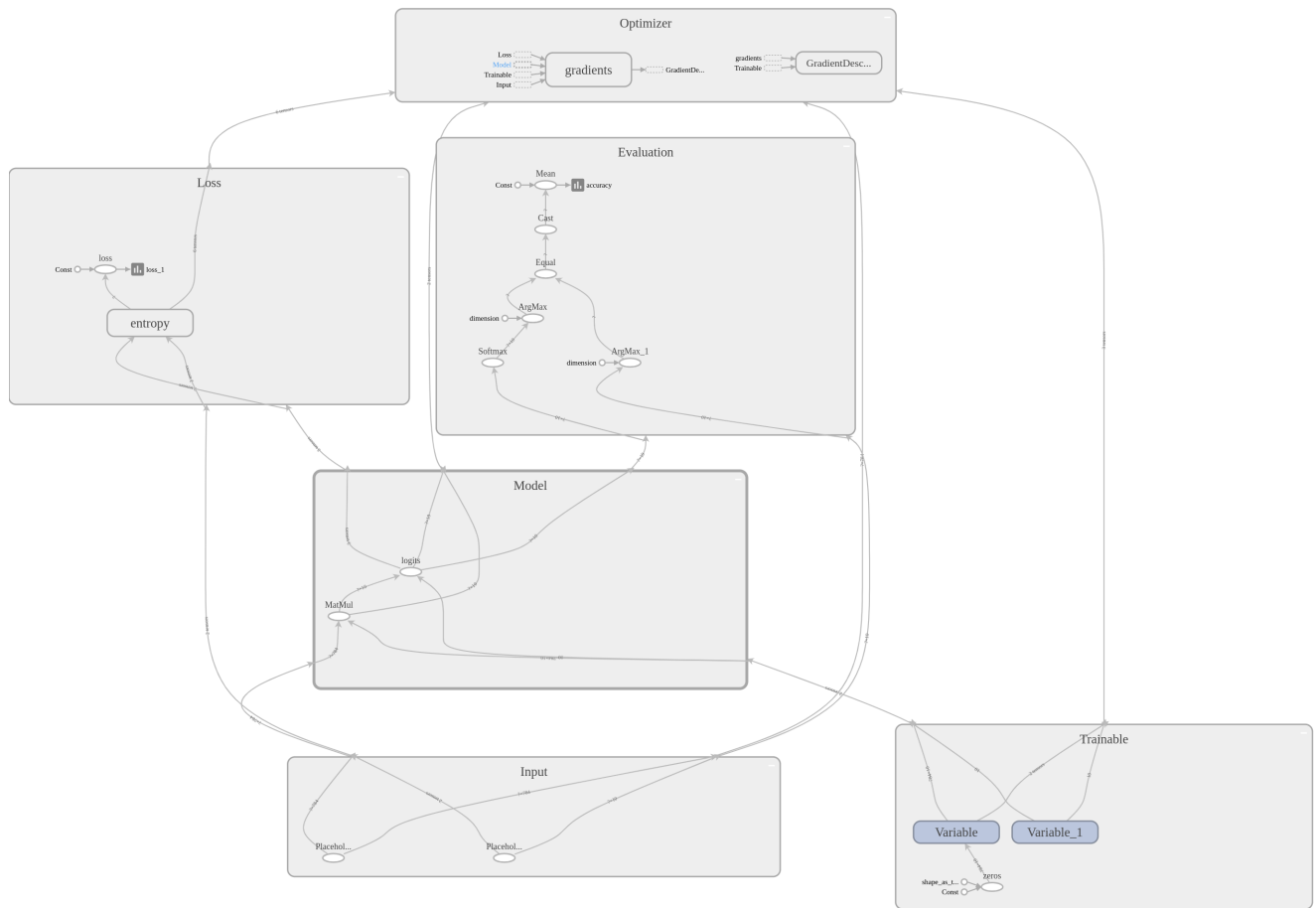


Figure 2: Logistic Regression Tensorboard Graphs

Problem3

The test accuracy of CNNs v1 is around 97%. Fig. 3, 4 show the training process and graph.

Print Messages

```

.....
Average loss at step 2050: 118.81773414611817
Average loss at step 2060: 149.64141769409179
Average loss at step 2070: 122.3837387084961
Average loss at step 2080: 98.15159149169922
Average loss at step 2090: 105.03060417175293
Average loss at step 2100: 111.31688957214355
Average loss at step 2110: 134.82549896240235
Average loss at step 2120: 137.8991569519043
Average loss at step 2130: 135.01569061279298
Average loss at step 2140: 179.05587310791014
Optimization Finished!
Total time: 221.5052146911621 seconds

```

Accuracy on test data: 0.9726

Tensorboard Visualization

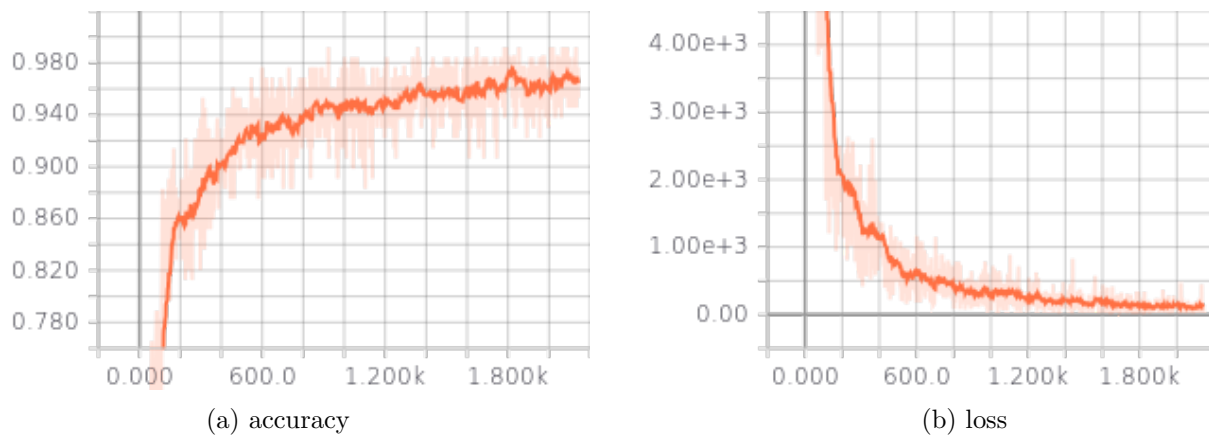


Figure 3: CNNs Tensorboard Scalars

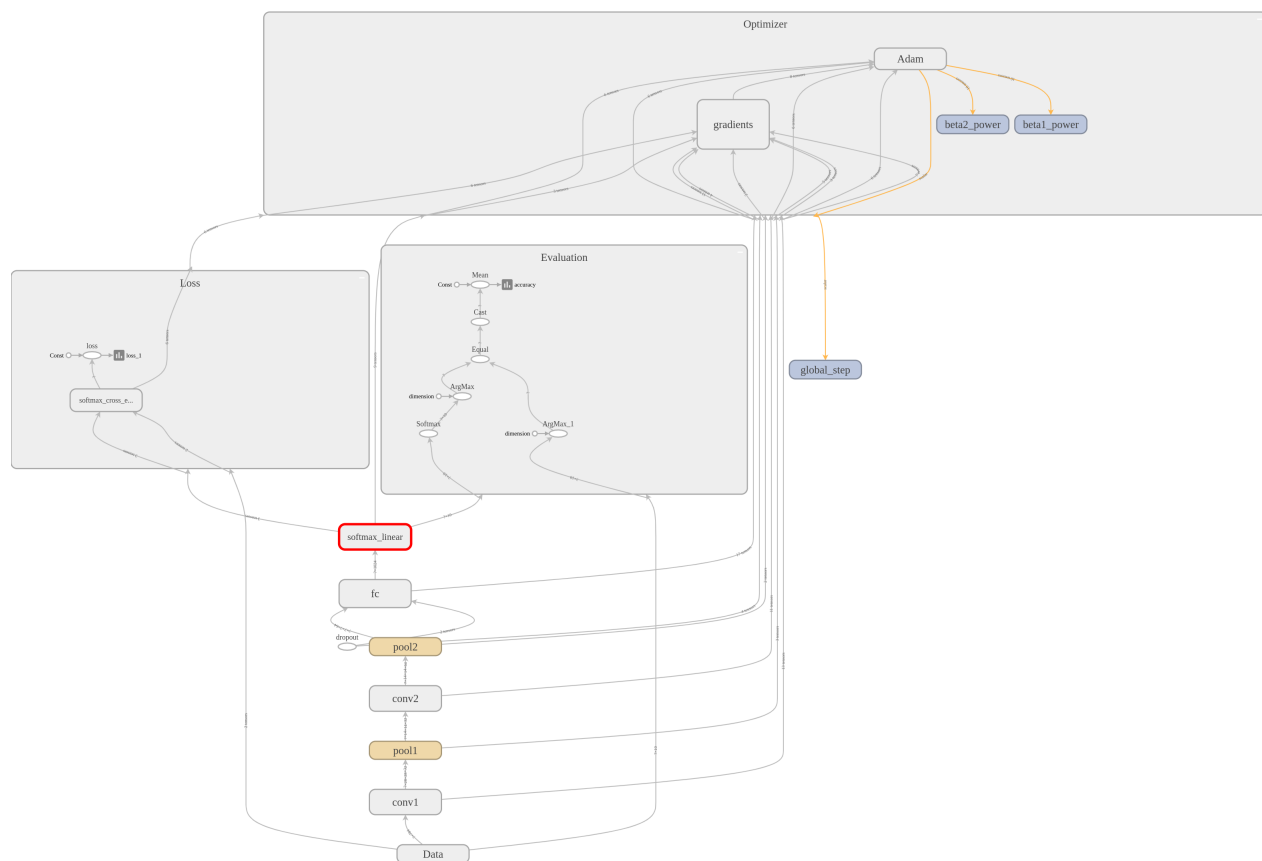


Figure 4: CNNs Tensorboard Graphs

Problem4

We re-implement CNNs and finally achieve over 99% test accuracy. Instead of using `tf.nn` APIs, we use higher-level `tf.layers` and most of the settings and parameters are exactly same with CNNs v1. The dropout is not used in CNNs v2. Fig. 5, 6 show the training process and graph.

Print Messages

```
Average loss at step 100: 0.3393741825222969
Average loss at step 200: 0.0819725345633924
Average loss at step 300: 0.0538727815169841
Average loss at step 400: 0.043934245305135844
Average loss at step 500: 0.03696376217063516
Average loss at step 600: 0.02999047989025712
Average loss at step 700: 0.028441412085667254
Average loss at step 800: 0.02279190320521593
Average loss at step 900: 0.021667199654038995
Average loss at step 1000: 0.01573014471854549
Optimization Finished!
Total time: 212.64736008644104 seconds
Accuracy on test data: 0.9921
```

Tensorboard Visualization

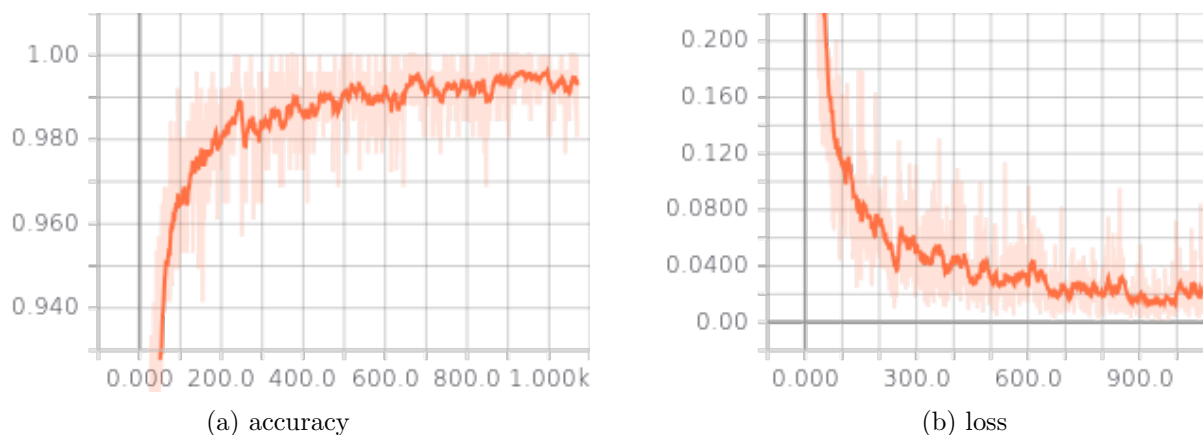


Figure 5: CNNs v2 Tensorboard Scalars

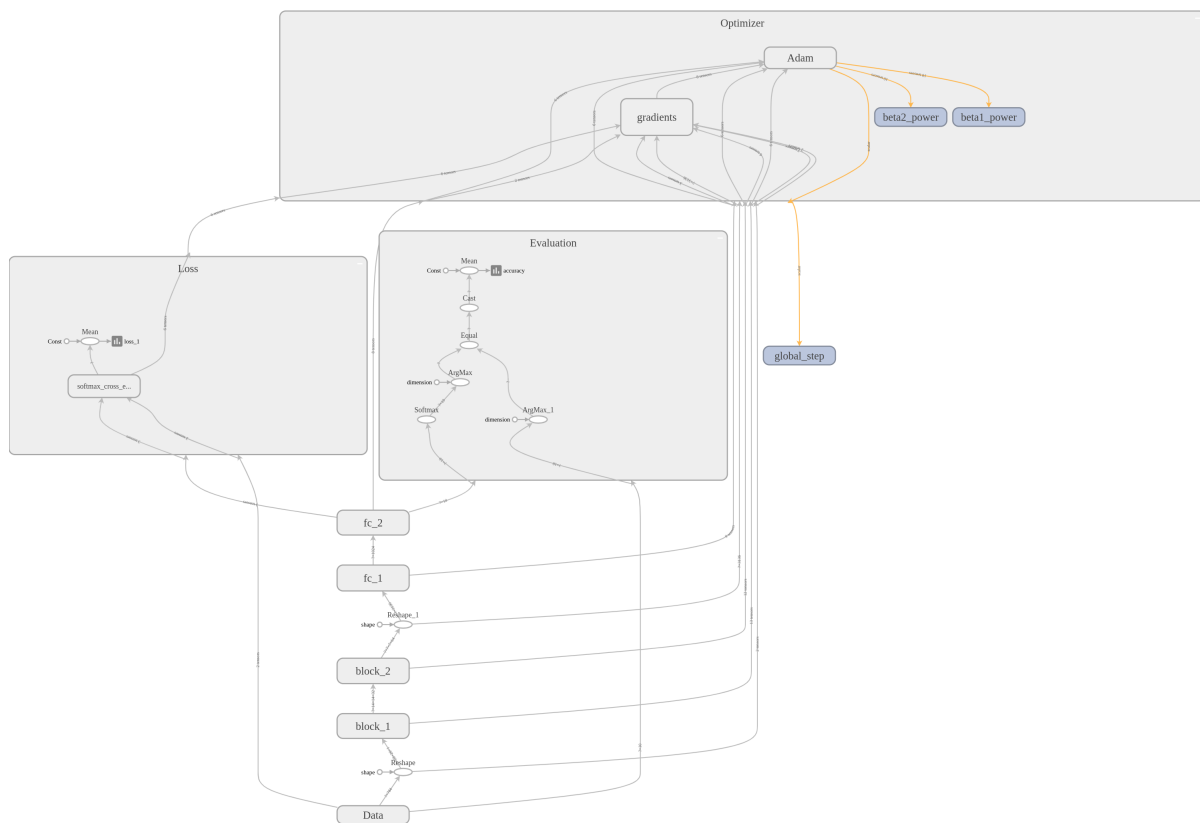


Figure 6: CNNs v2 Tensorboard Graphs

Problem5

To classify images using a recurrent neural network, we consider every image row as a sequence of pixels, as shown in Fig 7. The test accuracy of LSTM is around 98%. Fig. 8, 9 show the training process and graph.

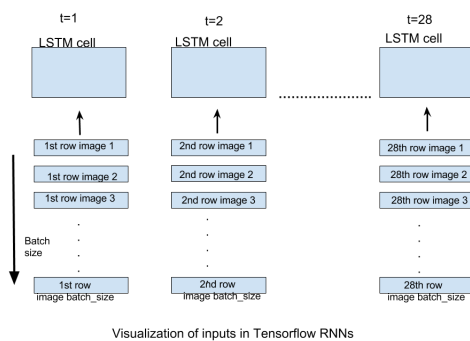


Figure 7: Visualization of inputs in LSTM

Print Messages

```
.....
Step 1200, minibatch loss= 0.0356, training accuracy= 0.984
Step 1400, minibatch loss= 0.0690, training accuracy= 0.969
Step 1600, minibatch loss= 0.0852, training accuracy= 0.969
Step 1800, minibatch loss= 0.0678, training accuracy= 0.984
Step 2000, minibatch loss= 0.0353, training accuracy= 0.992
Step 2200, minibatch loss= 0.0145, training accuracy= 0.992
Step 2400, minibatch loss= 0.0317, training accuracy= 0.984
Step 2600, minibatch loss= 0.0352, training accuracy= 0.992
Step 2800, minibatch loss= 0.0618, training accuracy= 0.977
Step 3000, minibatch loss= 0.0299, training accuracy= 0.992
Optimization Finished!
Total time: 106.87019944190979 seconds
Testing Accuracy: 0.9809
```

Tensorboard Visualization

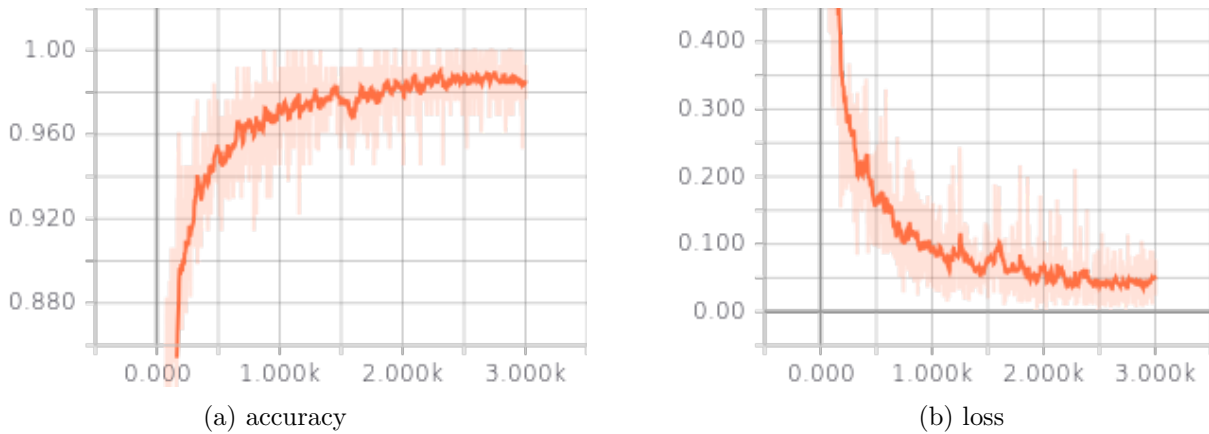


Figure 8: LSTM Tensorboard Scalars

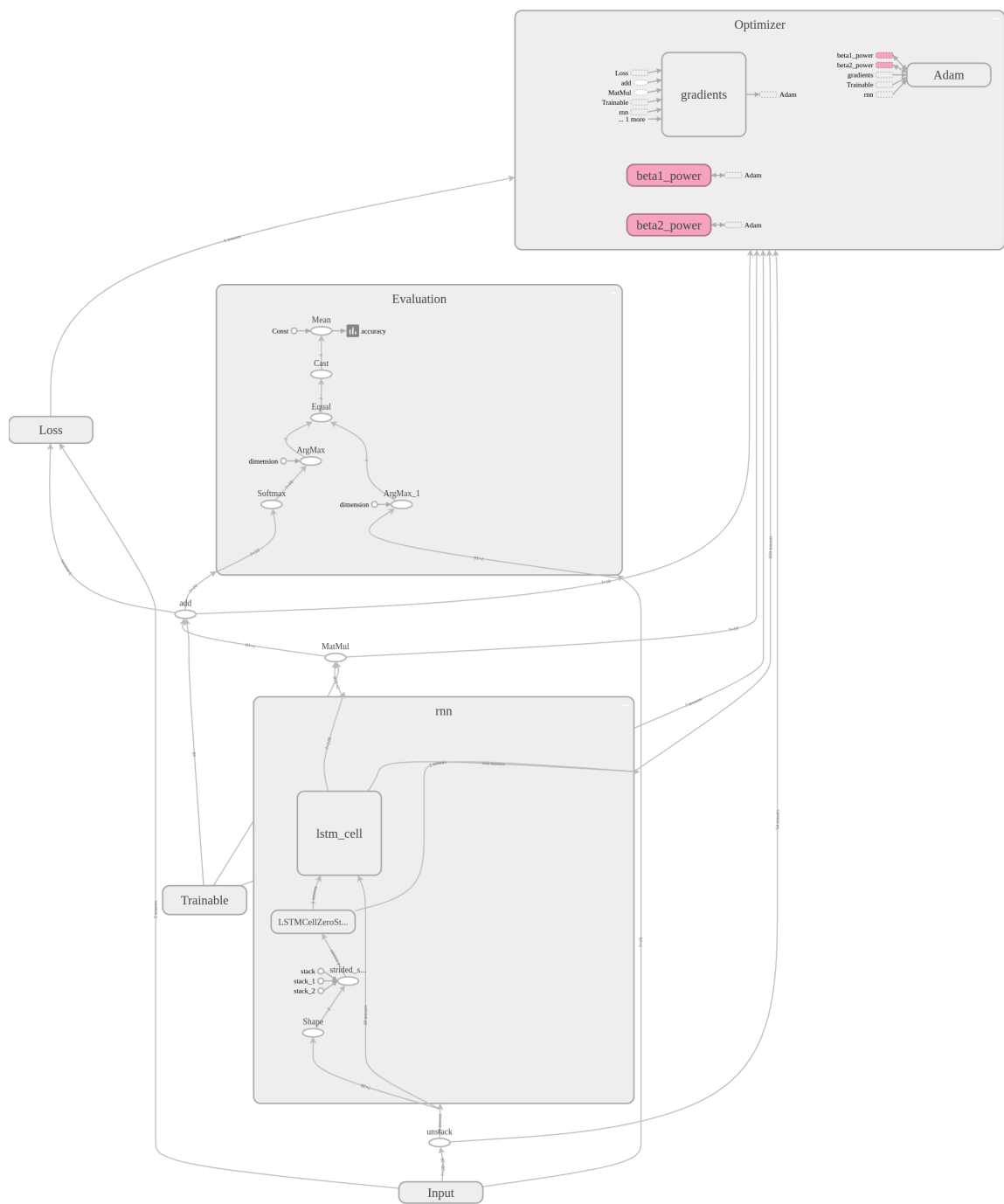


Figure 9: LSTM Tensorboard Graphs