CS498-AML Homework 9
Part 1: Convolutional Neural Network on MNIST Dataset
Eloise Rosen & Mark Berman

Part 1 A:

The Python based tutorial that was cited in the homework instructions was used as the baseline for our submission for Part 1. The Python tutorial was manually converted to the R based Tensorflow API and then integrated with TensorBoard "summary" operations to log histograms of output tensors and scalar metrics to capture "loss" (e.g., "cross entropy") and test set accuracy. Loss and test set accuracy were logged to TensorBoard every 100 iterations of the training loop. The training loop was set for a run of 10,000 iterations. The "R" code file for Part 1 is available for inspection and is labeled "hw9_mnist_part1_gd.Rmd".

The Convolutional Neural Network architecture for P 1 A Is as follows.

- Loss Minimization function: cross_entropy <- tf$reduce_mean(-tf$reduce_sum(y_ * tf$log(y_conv), reduction_indices=1L))
- Optimization function: GradientDescentOptimizer
- Learning Rate: 0.001
- Mini-batch size: 100
- Keep probability: 0.4
- Max. # of steps: 10,000
- # Training Elements: 50,000
- # Test Elements: 10,000
- Two Convolutional layers; with each of these layers followed by a pooling layer. Followed by two fully connected layers and final" Softmax" function call that produces a matrix of 50,000 predicted image labels.
    - Convolutional Layer 1:
        - Dim of each input training element ($a^{[0]}$): h =28, w=28, channels =1
        - kernel size: h=5, w=5, channels=1
        - # kernels: 32
        - stride: 1
        - padding: 2 (e.g., "SAME")
        - bias: 32
        - activation function: RELU
        - Dim of each output training element ($a^{[1]}$): h=28, w=28, channels=32

- Pooling Layer 1:
  - Dim of each training input element ($a^{[1]}$): h=28, w=28, channels=32
  - Kernel size: h=2, w=2, channels=1
  - # kernels: 32
  - stride: 2
  - Dim of each training output element ($a^{[2]}$): h=14, w=14, channels=32
- Convolutional Layer 2:
  - Dim of each training input element ($a^{[2]}$): h =14, w=14, channels =32
  - kernel size: h=5, w=5, channels=32
  - # kernels: 64
  - stride: 1
  - padding: 2 (e.g., "SAME")
  - bias: 64
  - activation function: RELU
  - Dim of each training output element ($a^{[3]}$): h=14, w=14, channels=64
- Pooling Layer 2:
  - Dim of each training input element ($a^{[3]}$): h =14, w=14, channels =64
  - Kernel size: h=2, w=2, channels=1
  - # kernels: 64
  - stride: 2
  - Dim of each training output element ($a^{[4]}$): h=7, w=7, channels=64
- Fully Connected Layer 1:
  - Dim of each input training element (after flattening $a^{[4]}$): h=1, w=3136
  - Dim of each output training element ($a^{[5]}$): h=1, w=1024
- Fully Connected Layer 2:
  - Dim of each input training element ($a^{[5]}$): h=1, w=1024
  - Dim of each output training element ($a^{[6]}$): h=1, w=10
- Softmax Function
  - Dim of each predicted element ($a^{[7]}$): h=1, w=10
  - Dim of prediction matrix: h=50,000, w=10

Findings:

Test set accuracy is measured every 100 iterations the Convolutional Neural Network (CNN) performs training over the training data set. Test set accuracy is 91.51 percent at 2,000th iteration of the CNN over the training data set. This is shown in the following screen capture from R-studio.

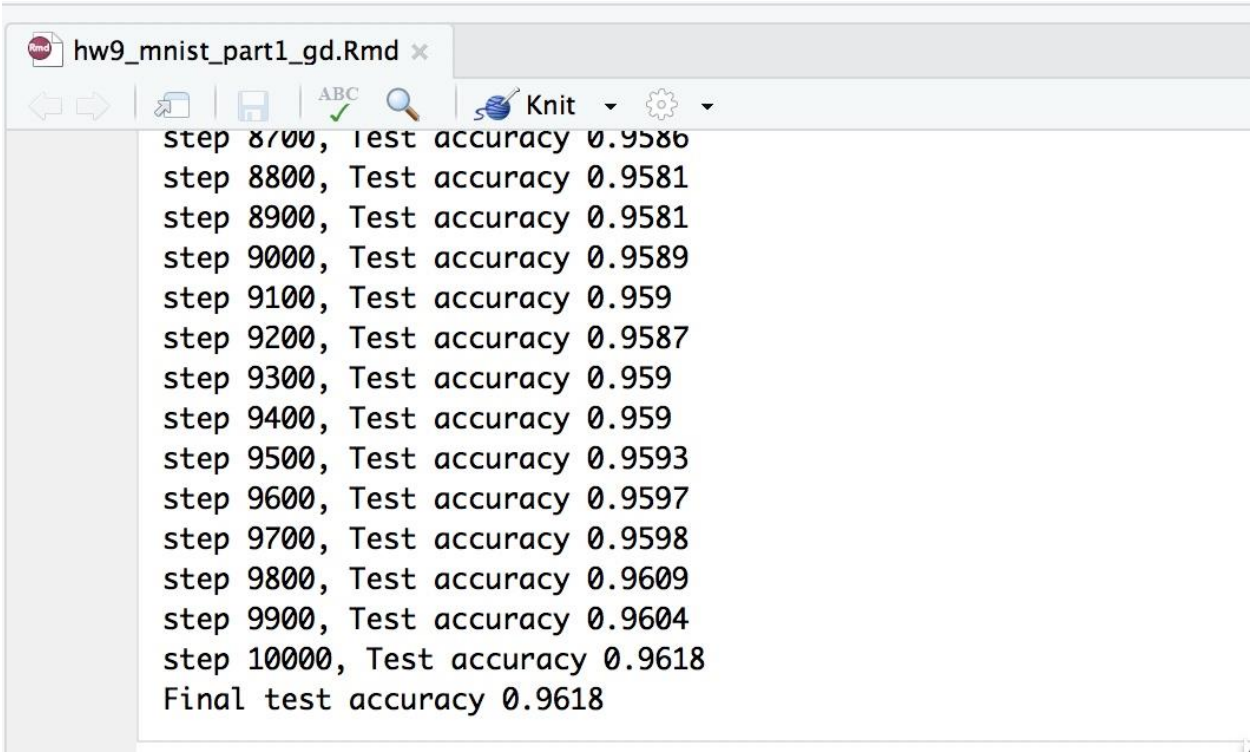
```
step 100, Test accuracy 0.5336
step 200, Test accuracy 0.6597
step 300, Test accuracy 0.7267
step 400, Test accuracy 0.7923
step 500, Test accuracy 0.8117
step 600, Test accuracy 0.8351
step 700, Test accuracy 0.853
step 800, Test accuracy 0.8639
step 900, Test accuracy 0.8728
step 1000, Test accuracy 0.8794
step 1100, Test accuracy 0.8868
step 1200, Test accuracy 0.8904
step 1300, Test accuracy 0.8963
step 1400, Test accuracy 0.8991
step 1500, Test accuracy 0.9011
step 1600, Test accuracy 0.9047
step 1700, Test accuracy 0.9084
step 1800, Test accuracy 0.9114
step 1900, Test accuracy 0.9136
step 2000, Test accuracy 0.9151
step 2100, Test accuracy 0.9185
step 2200, Test accuracy 0.9212
step 2300, Test accuracy 0.9242
step 2400, Test accuracy 0.9225
step 2500, Test accuracy 0.9282
step 2600, Test accuracy 0.9285
step 2700, Test accuracy 0.9297
step 2800, Test accuracy 0.9314
step 2900, Test accuracy 0.9328
step 3000, Test accuracy 0.9333
step 3100, Test accuracy 0.9335
step 3200, Test accuracy 0.9353
step 3300, Test accuracy 0.9362
```

7:1 (Top Level)

Console Terminal

~/cs498_AML/assignment9/MNIST_part1_gd/

Test set accuracy climbs to 96.18 percent accuracy after the 10,000th iteration over the training set by the CNN. This is shown below in the following screen capture from R-Studio.



The trajectory of the increase in test set accuracy is shown below in the following screen capture from TensorBoard. The slope of the test set accuracy curve (blue line) begins to flatten out at 3,000 iterations over the training set by the CNN. A larger version of this TensorBoard plot is available for inspection and is labeled "tensor_board_part1.jpeg".

TensorBoard SCALARS GRAPHS DISTRIBUTIONS HISTOGRAMS

Show data download links

Ignore outliers in chart scaling

Tooltip sorting method: default

Smoothing

0.6

Horizontal Axis

STEP RELATIVE WALL

Runs

Write a regex to filter runs

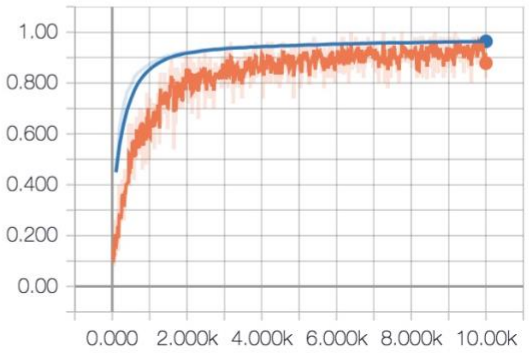
minst_logs/train

minst_logs/test

Filter tags (regular expressions supported)

accuracy

accuracy/accuracy



cross_entropy

cross_entropy/cross_entropy