EEL 6935 : BIG DATA ECOSYSTEMS ANEEZ FATIMA

21st Feb, 2017 **PROGRAMMING ASSIGNMENT 1** aneez.fatima@ufl.edu

INTRODUCTION

The aim is to classify MNIST handwritten digits using Tensorflow. Two models have been tried for the classification viz. SimpleModel and DeepModel to analyze that the accuracy is 82.78% and 98.42% respectively.

TENSORFLOW SETUP ON HIPERGATOR

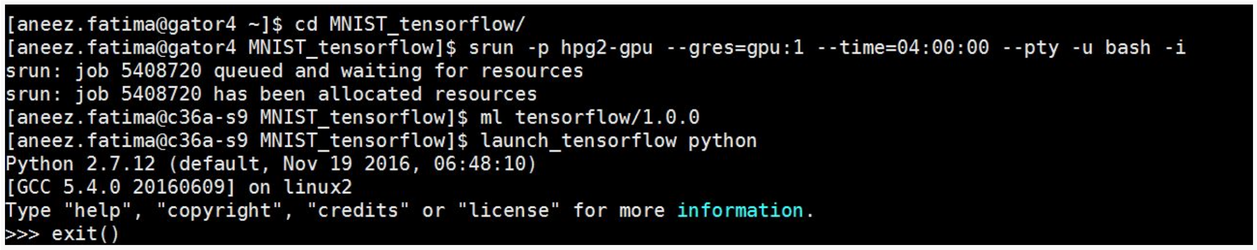
This is implemented on Hipergator 2.0 and the following explains how to access GPUs and setup Tensorflow.

To request access to Tesla K80 GPU for four hours, the command is

```
$ srun -p hpg2-gpu --gres=gpu:1 --time=04:00:00 --pty -u bash -i
```

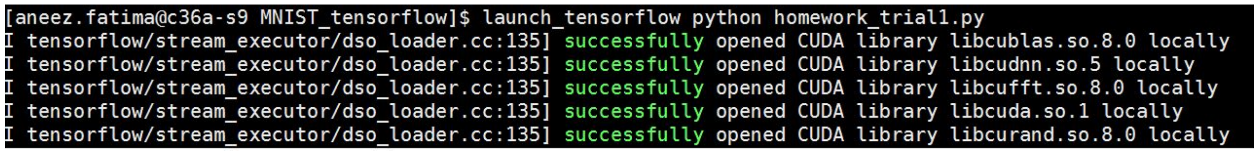
To access Tensorflow, the commands are

```
$ ml tensorflow/1.0.0
$ launch_tensorflow python
```



```
[aneez.fatima@gator4 ~]$ cd MNIST_tensorflow/
[aneez.fatima@gator4 MNIST_tensorflow]$ srun -p hpg2-gpu --gres=gpu:1 --time=04:00:00 --pty -u bash -i
srun: job 5408720 queued and waiting for resources
srun: job 5408720 has been allocated resources
[aneez.fatima@c36a-s9 MNIST_tensorflow]$ ml tensorflow/1.0.0
[aneez.fatima@c36a-s9 MNIST_tensorflow]$ launch_tensorflow python
Python 2.7.12 (default, Nov 19 2016, 06:48:10)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> exit()
```

And when we run the tensorflow code, all libraries get loaded as shown below.



```
[aneez.fatima@c36a-s9 MNIST_tensorflow]$ launch_tensorflow python homework_trial1.py
I tensorflow/stream_executor/dso_loader.cc:135] successfully opened CUDA library libcublas.so.8.0 locally
I tensorflow/stream_executor/dso_loader.cc:135] successfully opened CUDA library libcudnn.so.5 locally
I tensorflow/stream_executor/dso_loader.cc:135] successfully opened CUDA library libcufft.so.8.0 locally
I tensorflow/stream_executor/dso_loader.cc:135] successfully opened CUDA library libcuda.so.1 locally
I tensorflow/stream_executor/dso_loader.cc:135] successfully opened CUDA library libcurand.so.8.0 locally
```

DATA

We know that MNIST is a simple computer vision dataset that consists of images of handwritten digits. For this assignment, two CSV files namely train.csv and test.csv are given.

train.csv contains 32000 rows where each row consists of 1 label column and 784 pixel columns. These 784 columns are the flattened version of each MNIST image of size 28x28 pixels. Similarly, **test.csv** contains 10000 rows with 784 pixel columns for which we need to find out the label.

SIMPLE MODEL

Only one layer of softmax regression is used with 10 classes viz. 0,1,2,...,8,9. For a given image, the probability for its actual class need to be the highest. This way, the model is trained to adjust weights and bias to classify correctly.

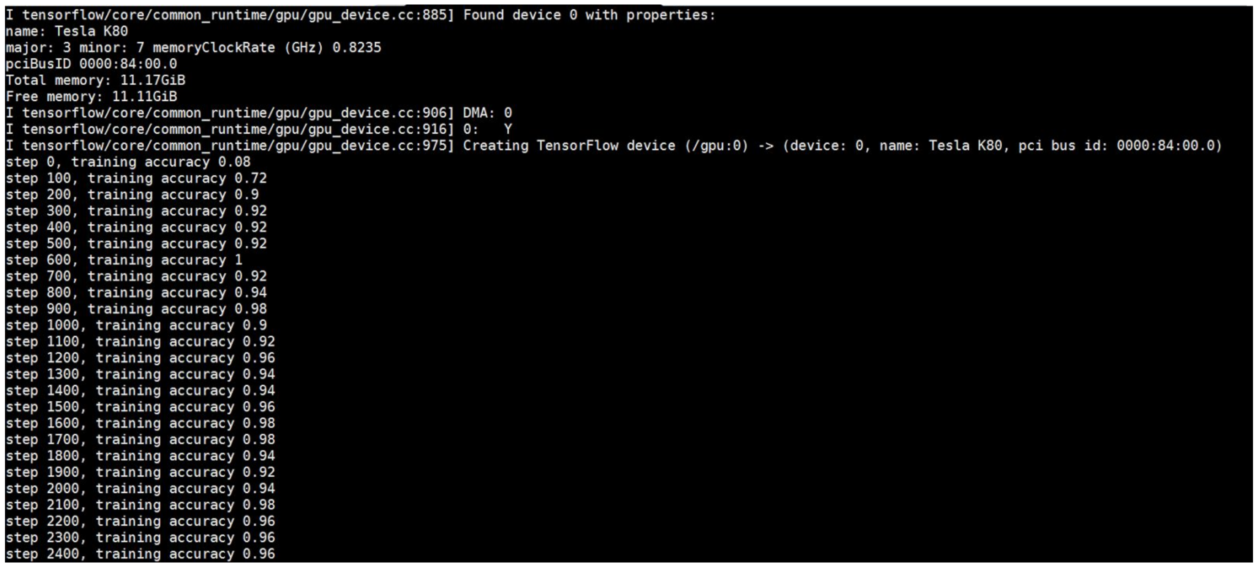
The training is done in 1000 steps by selecting 100 random images from the train dataset each time and determining the loss function using cross-entropy.

This model acheived an accuracy of 82.78% and code can be found here. https://github.com/aneezfatima/MNIST-Tensorflow/blob/master/simplemodel.py

DEEP MODEL

This is similar to SimpleModel but Multilayer Convolutional Network is used. It consists of two convolutional layers and one densely connected layer. The first layer computes 32 features for 5x5 patch and is followed by max pooling that will reduce the image size to 14x14. The second layer computes 64 features for each 5x5 patch. The densely connected layer has 1024 neurons that process on entire image. Now, followed by pooling, these are multiplied with weights, the biases are applied and then ReLUs are applied.

The training is done in 20000 steps with a batch of 50 randomly selected images from train set. The accuracy is reported for every 100th iteration as shown in the screenshot below.



```
I tensorflow/core/common_runtime/gpu/gpu_device.cc:885] Found device 0 with properties:
name: Tesla K80
major: 3 minor: 7 memoryClockRate (GHz) 0.8235
pciBusID 0000:84:00.0
Total memory: 11.17GiB
Free memory: 11.11GiB
I tensorflow/core/common_runtime/gpu/gpu_device.cc:906] DMA: 0
I tensorflow/core/common_runtime/gpu/gpu_device.cc:916] 0:   Y
I tensorflow/core/common_runtime/gpu/gpu_device.cc:975] Creating TensorFlow device (/gpu:0) -> (device: 0, name: Tesla K80, pci bus id: 0000:84:00.0)
step 0, training accuracy 0.08
step 100, training accuracy 0.72
step 200, training accuracy 0.9
step 300, training accuracy 0.92
step 400, training accuracy 0.92
step 500, training accuracy 0.92
step 600, training accuracy 1
step 700, training accuracy 0.92
step 800, training accuracy 0.94
step 900, training accuracy 0.98
step 1000, training accuracy 0.9
step 1100, training accuracy 0.92
step 1200, training accuracy 0.96
step 1300, training accuracy 0.94
step 1400, training accuracy 0.94
step 1500, training accuracy 0.96
step 1600, training accuracy 0.98
step 1700, training accuracy 0.98
step 1800, training accuracy 0.94
step 1900, training accuracy 0.92
step 2000, training accuracy 0.94
step 2100, training accuracy 0.98
step 2200, training accuracy 0.96
step 2300, training accuracy 0.96
step 2400, training accuracy 0.96
```

This model acheived an accuray of 98.42% and code can be found here. https://github.com/aneezfatima/MNIST-Tensorflow/blob/master/deepmodel.py

REFERENCES:

1. https://wiki.rc.ufl.edu/doc/GPU_Access
2. https://wiki.rc.ufl.edu/doc/TensorFlow
3. https://www.tensorflow.org/get_started/mnist/beginners
4. https://www.tensorflow.org/get_started/mnist/pros