TensorFlow Project
Experiments with Convolutional Neural Network in TensorFlow

Background

Convolutional neural network is a powerful learning mechanism that was applied successfully to many computer vision tasks. In this project you are asked to run experiments using tensorflow. The experiments are closely related to an example given in the TensorFlow tutorial.
The goal is to train a CNN to recognize images of digits from the MNIST dataset. This dataset consists of gray level images of size 28x28. There is a standard partitioning of the dataset into training and testing. The training has 60,000 examples and the testing has 10,000 examples. Standard CNN models achieves over 99% accuracy on this dataset.

In this project you are asked to solve a similar problem of creating a classifier for the MNIST data. However, the training data in our case has only 6,000 images, and each image is shrunk to size 7x7. Specifically, your program must include the following:

1. Your program must set the random seeds of python and tensorflow to 1 to make sure that your results are reproducible.

2. The first layer in the network must be a 4×4 **maxpooling** layer. This effectively shrinks the images from 28x28 to 7x7.

3. Your program will be tested by training on a fraction of 0.1 of the standard training set. The testing data will be the entire standard testing set.

4. The training and testing in you program should not take more than 6 minutes.

Provided programs and data

1. The original 60,000 training examples are in the folder **big**.

2. A random subset of 6,000 the training examples is in the folder **small**.

3. An example program that satisfies the project requirements is available as **proj-tf1.py**.

4. A program that can extract a random fraction from the training data is available as **fraction_xy.py**.

What you need to do

Design a network to solve this problem. You can use all the functionality of tensorflow, not only the parts that were described in class.

Grading

We will generate a random set of 6,000 training examples by running the program fraction_xy.py with a seed that is kept secret. If, for example, the seed is 7, the program is run as follows:

```
python3 fraction\_xy.py  big/x\_train.csv big/y\_train.csv 0.1 7
```

Your grade will be based on the accuracy of your model trained with the generated examples and tested on the standard testing data.

What you need to submit

1. Source code of the python script.

2. Documentation describing your network, and the results of experiments/accuracy that your program achieves on the provided data.