Experiments with Convolutional Neural Network in TensorFlow

Class Project

Background

Convolutional neural network is a powerful learning mechanism that was applied successfully to many computer vision applications. In this project we will run experiments using tensorflow. The experiments are closely related to the online example available at:

https://www.tensorflow.org/get_started/mnist/pros.

The above online example shows how to apply convolutional neural net to learn how to classify data from the MNIST dataset. The code in the example takes about 30 minutes to train, and achieves over 99% accuracy.

For this project, you are asked to solve the exact same problem, creating a classifier for the MNIST data. However, we put two constraints on the network being created that are intended to **reduce** the network accuracy:

1. The first layer in the network must be a 2×2 **maxpool** layer.

2. Training must be in batches of 100 examples, and only 1000 batches are allowed.

You should produce programs and report results for experiments that have 1,2,3,4,5 convolutional layers. The other layers in the network are entirely up to you. You will be graded on the performance of your network on the standard testing data for this dataset.

Example program

An example program that satisfies these constraints is available as **proj2.py**.

Limitations

1. The first layer is max pool of 2x2.

2. Train for 1000 batches.

3. Each training batch should have 100 images.

4. Keep test accuracy code as it is in proj2.py

Things you may want to change

1. No limitation on max pool after first one(where and how many).

2. No limitation on fully connected layer.

3. No limitation on using ReLu or other functions.

4. No limitation on size of the convolution masks.

5. No limitation on number of convolution masks used in each layer.

6. No limitation on using/not using dropout.

7. No limitation on how to initialize weights.

8. No limitation on Optimizer nor their learning rate.

What you need to submit

1. Source code of 5 programs. Please call them: mnist1.py, mnist2.py, mnist3.py, mnist4.py, mnist5.py.

The program minst$i$.py is an implementation with i convolutional layers.

2. Documentation that explains what you have done.