Experiments with a deep neural network using TensorFlow

Background

In this assignment you are asked to experiment with methods of combating overfitting and the ADAM optimization algorithm. In all cases the network should be constructed so that its accuracy on test data is as high as possible. Your results should be four python scripts created by modifying the provided example **gh2.py**. Please notice that in order to run the script you should have the file **our_mnist.py** in the same folder as the script you are running. (You are not allowed to make any changes to **our_mnist.py**). It was created by reducing the number of training data of the **mnist** dataset. With a small number of training data it is expected that overfitting will be significant.

Script 1. Please name your script **script1.py**. In this script you must use the **tf.train.GradientDescentOptimizer**. You may change all the numeric parameters (with the exception of the number of batches). You may also use different **fully connected** layers (e.g., sigmoids), a different number of layers, and nodes, etc. The condition is that whatever used in this script was previously used in a script given in class. The scripts given in class are: **mnist01.py**, **mnist02.py**, **mnist2.py**, and **gh2.py**.

Script 2. Please name your script **script2.py**. The limitations here are exactly the same as in Script1, except that you should use **tf.train.AdamOptimizer** instead of **tf.train.GradientDescentOptimizer**.

Script 3. Please name your script **script3.py**. The limitations here are similar to Script1. You must use the **tf.train.GradientDescentOptimizer**, and all layers are fully connected. But here you are not limited to what was previously described in class. You can use anything from the TensorFlow API.
See https://www.tensorflow.org/api_docs/python.

Script 4. Please name your script **script4.py**. The limitations here are similar to Script2. You must use the **tf.train.AdamOptimizer**, and all layers are fully connected. But here you are not limited to what was previously described in class. You can use anything from the TensorFlow API.
See https://www.tensorflow.org/api_docs/python.

Evaluation

The challenge is to get as high accuracy as possible, with the limitation that each script uses **at most 3000 stochastic batches**. Observe that because of the random initialization different runs of the programs may produce (slightly) different accuracy values. The one that counts for your grade is the one produced by your program when you meet with the TAs.

What you need to submit

1. Source code of the 4 python scripts.

2. Documentation describing your experiments and the test accuracy that your programs achieved.

You **must** be available to demonstrate your program to the TAs. Time slots and additional instructions will be announced later.

Deadline:

TBA.