Deep Learning
Assignment 2
10 points

This assignment will build on the software found at
https://www.tensorflow.org/get_started/mnist/pros

You should read through this tutorial, along with the supplementary slides on Convolutional Networks (wk2_CNN.pdf)

The necessary code can be downloaded from here
https://github.com/tensorflow/tensorflow/blob/r1.3/tensorflow/examples/tutorials/mnist/mnist_deep.py

1. [2 marks]

Open the file mnist_deep.py and start editing it. In order to speed up the training, we will reduce the number of training epochs, by changing this line:

```
    for i in range(20000):
```

to this:

```
    for i in range(10000):
```

We also want to generate a confusion matrix; at line 169, change this:

```
    print('test accuracy %g' % accuracy.eval(feed_dict={
        x: mnist.test.images, y_: mnist.test.labels, keep_prob: 1.0}))

if __name__ == '__main__':
```

to this:

```
    print('test accuracy %g' % accuracy.eval(feed_dict={
        x: mnist.test.images, y_: mnist.test.labels, keep_prob: 1.0}))

    print(sess.run(cm, feed_dict={x: mnist.test.images,
                                  y_: mnist.test.labels, keep_prob: 0.5}))

if __name__ == '__main__':
```

At line 151, change this:

```
  accuracy = tf.reduce_mean(correct_prediction)
```

```
  graph_location = tempfile.mkdtemp()
```

to this:

```
  accuracy = tf.reduce_mean(correct_prediction)

  with tf.name_scope('accuracy'):
    true_class=tf.argmax(y_conv, 1)
    predicted_class=tf.argmax(y_, 1)
  cm=tf.confusion_matrix(predicted_class,true_class)

  graph_location = tempfile.mkdtemp()
```

Now run the code for the convolutional network, by typing:

python mnist_deep.py

Copy the last lines of the output (test performance and confusion matrix) into your answers.

How does the accuracy compare with that of the MLP model from Assignment 1?
Which three digits are most likely to be misclassified?

2. [4 marks]

The network used in this code is similar to the LeNet model shown in Slide 5 of the supplementary slides, but with some differences. The original image is 28x28 instead of 32x32, but zero-padding with p=2 ensures that the first convolutional layer is also 28x28. The first pooling layer reduces the size to 14x14, and zero-padding again ensures that the second convolutional layer is also 14x14. You can find out other details about the network by looking through the code (including comments).

For each layer in the network (first and second convolutional layer, fully connected layer and output layer) calculate the number of:

a. weights per neuron?
b. neurons?
c. connections? (including connections from the "zero-padding" regions)
d. independent parameters?

3. [4 marks]

a. In the context of CNN with ReLU, briefly explain (in one or two sentences) why Max Pooling may be a good strategy.
b. List two different methods of Data Augmentation for image classification.
c. Briefly explain how dropout simulates a bagged ensemble of network architectures.