Homework 01

Brian Henderson

09/25/17

1 Part 1

Run the code below for the current CNN architecture and record the time it took to train (estimate) and its final accuracy; include this in a report.

```
for i in range(20000):
    batch = mnist.train.next_batch(50)
    if i%100 == 0:
        train_accuracy = accuracy.eval(feed_dict={
            x:batch[0], y_: batch[1], keep_prob: 1.0})
        print("step %d, training accuracy %g"%(i, train_accuracy))
    train_step.run(feed_dict={x: batch[0], y_: batch[1], keep_prob: 0.5})
```

1.1 Training Time

39 minutes 38 seconds

1.2 Final Training Accuracy

1.0

2 Part 2

Part 2: Do the same but change the architecture slightly (e.g., number of feature maps in each convolutional layer), and record the time it took to train (estimate) and its final accuracy; include this in a report.

2.1 Code Modifications

Modified the number of feature maps in each convolutional layer by reducing original number by 50%.

```
W_conv1 = tf.Variable(tf.truncated_normal([5, 5, 1, 16], stddev=0.1))
b_conv1 = tf.Variable(tf.constant(0.1, shape=[16])) #16 biases for 16 outputs
```

```
W_conv2 = tf.Variable(tf.truncated_normal([5, 5, 16, 32], stddev=0.1))
b_conv2 = tf.Variable(tf.constant(0.1, shape=[32])) #32 biases for 32 outputs
```

```
layer2_matrix = tf.reshape(conv2, [-1, 7*7*32])

W_fc1 = tf.Variable(tf.truncated_normal([7 * 7 * 32, 512], stddev=0.1))
b_fc1 = tf.Variable(tf.constant(0.1, shape=[512])) #512 biases for 512 outputs
```

```
W_fc2 = tf.Variable(tf.truncated_normal([512, 10], stddev=0.1)) #512 neurons
```

2.2 Training Time

18 minutes 30 seconds.

2.3 Final Training Accuracy

1.0