

```

1  import tensorflow as tf
2
3  from tensorflow.examples.tutorials.mnist import input_data
4  mnist = input_data.read_data_sets("MNIST_data/", one_hot=True)
5
6  sess = tf.InteractiveSession
7
8
9  def weight_variable(shape):
10     initial = tf.truncated_normal(shape, stddev=0.1)
11     return tf.Variable(initial)
12
13
14  def bias_variable(shape):
15     initial = tf.constant(0.1, shape=shape)
16     return tf.Variable(initial)
17
18
19  def conv2d(x, W):
20     return tf.nn.conv2d(x, W, strides=[1,1,1,1], padding='SAME')
21
22
23  def max_pool_2x2(x):
24     return tf.nn.max_pool(x, ksize=[1,2,2,1], strides=[1,2,2,1],
25                             padding='SAME')
26
27  x = tf.placeholder(tf.float32, [None, 784])
28  y_ = tf.placeholder(tf.float32, [None, 10])
29  x_image = tf.reshape(x, [-1,28,28,1])
30
31  W_conv1 = weight_variable([5,5,1,32])
32  b_conv1 = bias_variable([32])
33  h_conv1 = tf.nn.relu(conv2d(x_image, W_conv1)+b_conv1)
34  h_pool1 = max_pool_2x2(h_conv1)
35
36  W_conv2 = weight_variable([5,5,32,64])
37  b_conv2 = bias_variable([64])
38  h_conv2 = tf.nn.relu(conv2d(h_pool1,W_conv2)+b_conv2)
39  h_pool2 = max_pool_2x2(h_conv2)
40
41  W_fc1 = weight_variable([7*7*64,1024])
42  b_fc1 = bias_variable([1024])
43  h_pool2_flat = tf.reshape(h_pool2, [-1, 7*7*64])
44  h_fc1 = tf.nn.relu(tf.matmul(h_pool2_flat, W_fc1)+b_fc1)
45
46  keep_prob = tf.placeholder(tf.float32)
47  h_fc1_drop = tf.nn.dropout(h_fc1, keep_prob)
48
49  W_fc2 = weight_variable([1024,10])
50  b_fc2 = bias_variable([10])
51  y_conv = tf.nn.softmax(tf.matmul(h_fc1_drop, W_fc2)+b_fc2)
52
53  cross_entropy = tf.reduce_mean(-tf.reduce_sum(y_ * tf.log(y_conv),
54                                                  reduction_indices=[1]))
55  train_step = tf.train.AdamOptimizer(1e-4).minimize(cross_entropy)
56
57  correct_prediction = tf.equal(tf.argmax(y_conv,1), tf.argmax(y_, 1))
58  accuracy = (tf.cast(correct_prediction,tf.float32))
59
60  tf.global_variables_initializer()
61

```

```
62 for i in range(20000):
63     batch = mnist.train.next_batch(50)
64     if i%100 == 0:
65         train_accuracy = accuracy.eval(feed_dict = {x: batch[0], y_: batch[1],
66                                                     keep_prob: 1.0})
67         print("step %d, training accuracy %g"%(i, train_accuracy))
68     train_step.run(feed_dict={x: batch[0], y_:batch[1], keep_prob: 0.5})
69
70 print("test accuracy %g"%accuracy.eval(feed_dict={
71     x: mnist.test.images, y_: mnist.test.labels, keep_prob: 1.0}))
```