

```
cmd コマンドプロンプト - vagrant ssh
(tensorflow) vagrant@vagrant-ubuntu-trusty-64:~$ python
Python 3.5.2 |Continuum Analytics, Inc.| (default, Jul 2 2016, 17:53:06)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-1)] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>> from tensorflow.examples.tutorials.mnist import input_data
mnist = input_data.read_data_sets("MNIST_data/", one_hot=True)
>>> mnist = input_data.read_data_sets("MNIST_data/", one_hot=True)
Extracting MNIST_data/train-images-idx3-ubyte.gz
Extracting MNIST_data/train-labels-idx1-ubyte.gz
Extracting MNIST_data/t10k-images-idx3-ubyte.gz
Extracting MNIST_data/t10k-labels-idx1-ubyte.gz
>>> import tensorflow as tf
>>> x = tf.placeholder(tf.float32, [None, 784])
>>> W = tf.Variable(tf.zeros([784, 10]))
>>> b = tf.Variable(tf.zeros([10]))
>>> y = tf.nn.softmax(tf.matmul(x, W) + b)
>>> y_ = tf.placeholder(tf.float32, [None, 10])
>>> cross_entropy = tf.reduce_mean(-tf.reduce_sum(y_ * tf.log(y), reduction_indices=[1]))
>>> train_step = tf.train.GradientDescentOptimizer(0.5).minimize(cross_entropy)
>>> init = tf.initialize_all_variables()
>>> sess = tf.Session()
>>> sess.run(init)
>>> for i in range(1000):
...     batch_xs, batch_ys = mnist.train.next_batch(100)
...     sess.run(train_step, feed_dict={x: batch_xs, y_: batch_ys})
...
>>> correct_prediction = tf.equal(tf.argmax(y,1), tf.argmax(y_,1))
>>> accuracy = tf.reduce_mean(tf.cast(correct_prediction, tf.float32))
>>> print(sess.run(accuracy, feed_dict={x: mnist.test.images, y_: mnist.test.labels}))
0.9179
```