mnist01.py

```
# Learning with old style neural network


# MNIST_data is a collection of 2D gray level images.
# Each image is a picture of  a digit from 0..9
# Each image is of size 28 x 28 pixels


from tensorflow.examples.tutorials.mnist import input_data
mnist = input_data.read_data_sets('MNIST_data', one_hot=True)

import tensorflow as tf
sess = tf.InteractiveSession()

# xi is an image of size n. yi is the N labels of the image
# X is mxn. Row xi of X is an image
# Y is mxN. Row yi of Y is the labels of xi
X = tf.placeholder(tf.float32, shape=[None, 784])
Y = tf.placeholder(tf.float32, shape=[None, 10])

# a method for initializing weights. Initialize to small random values

def weight_variable(shape):
  initial = tf.truncated_normal(shape, stddev=0.1)
  return tf.Variable(initial)

# a method for initializing bias. Initialize to 0.1

def bias_variable(shape):
  initial = tf.constant(0.1, shape=shape)
  return tf.Variable(initial)

# Densely Connected Hidden Layer of 1024 nodes

W_fc1 = weight_variable([784, 1024])
b_fc1 = bias_variable([1024])
v_fc1 = tf.nn.sigmoid(tf.matmul(X, W_fc1) + b_fc1) # v_fc1 ?x1024

# Readout Layer

W_fc2 = weight_variable([1024, 10])
b_fc2 = bias_variable([10])
v_fc2 = tf.nn.sigmoid(tf.matmul(v_fc1, W_fc2) + b_fc2) # v_fc2 ?x10

predicted_Y = v_fc2;

sess.run(tf.global_variables_initializer())

mse = tf.losses.mean_squared_error(Y, predicted_Y)

train_step = tf.train.GradientDescentOptimizer(0.5).minimize(mse)

```

```
for i in range(1500):
    batch = mnist.train.next_batch(100)
    if i % 100 == 0:
        correct_prediction = tf.equal(tf.argmax(predicted_Y,1), tf.argmax(Y,1))
        accuracy = tf.reduce_mean(tf.cast(correct_prediction, tf.float32))
        print(i, accuracy.eval(feed_dict={X: mnist.test.images, Y: mnist.test.labels})
    train_step.run(feed_dict={X: batch[0], Y: batch[1]})

correct_prediction = tf.equal(tf.argmax(predicted_Y,1), tf.argmax(Y,1))
accuracy = tf.reduce_mean(tf.cast(correct_prediction, tf.float32))

print(accuracy.eval(feed_dict={X: mnist.test.images, Y: mnist.test.labels}))
```