mnist-eager-linear.py

```python
# Learning with a linear model

import numpy as np
import tensorflow as tf
tf.enable_eager_execution()

# supress warnings
old_v = tf.logging.get_verbosity()
tf.logging.set_verbosity(tf.logging.ERROR)

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

W = tf.contrib.eager.Variable(tf.zeros([784,10]))
b = tf.contrib.eager.Variable(tf.zeros([10]))

def prediction(X):
    predicted_Y = tf.matmul(X,W) + b
    return predicted_Y

def loss(X,Y):
    logits = prediction(X)
    loss = tf.losses.softmax_cross_entropy(onehot_labels=Y,logits=logits)
    return loss

def binary_accuracy(X, Y):
    logits = prediction(X)
    predict = tf.argmax(logits, 1).numpy()
    target = np.argmax(Y, 1)
    binary_accuracy = np.sum(predict == target)/len(target)
    return(binary_accuracy)

for i in range(1000):
    X, Y = mnist.train.next_batch(100)
    tf.train.GradientDescentOptimizer(0.5).minimize(lambda: loss(X,Y))

# evaluate the result
X, Y = mnist.test.images, mnist.test.labels
test_accuracy = binary_accuracy(X, Y)
print("test accuracy %g" % (test_accuracy))
```