*# Lab 10 MNIST and NN*import numpy as np  
import random  
import tensorflow as tf  
  
random.seed(777) *# for reproducibility*learning\_rate = 0.001  
batch\_size = 100  
training\_epochs = 15  
nb\_classes = 10  
  
(x\_train, y\_train), (x\_test2, y\_test) = tf.keras.datasets.mnist.load\_data()  
print(x\_train.shape)  
  
x\_train = x\_train.reshape(x\_train.shape[0], 28 \* 28)  
x\_test = x\_test2.reshape(x\_test2.shape[0], 28 \* 28)  
  
y\_train = tf.keras.utils.to\_categorical(y\_train, nb\_classes)  
y\_test = tf.keras.utils.to\_categorical(y\_test, nb\_classes)  
  
tf.model = tf.keras.Sequential()  
tf.model.add(tf.keras.layers.Dense(input\_dim=784, units=256, activation=**'relu'**))  
tf.model.add(tf.keras.layers.Dense(units=256, activation=**'relu'**))  
tf.model.add(tf.keras.layers.Dense(units=nb\_classes, activation=**'softmax'**))  
tf.model.compile(loss=**'categorical\_crossentropy'**,  
 optimizer=tf.keras.optimizers.Adam(lr=learning\_rate), metrics=[**'accuracy'**])  
tf.model.summary()  
  
tf.model.fit(x\_train, y\_train, batch\_size=batch\_size, epochs=training\_epochs)  
  
*# predict 10 random hand-writing data*y\_predicted = tf.model.predict(x\_test)  
for x in range(0, 10):  
 random\_index = random.randint(0, x\_test.shape[0]-1)  
 print(**"index: "**, random\_index,  
 **"actual y: "**, np.argmax(y\_test[random\_index]),  
 **"predicted y: "**, np.argmax(y\_predicted[random\_index]))  
  
*# evaluate test set*evaluation = tf.model.evaluate(x\_test, y\_test)  
print(**'loss: '**, evaluation[0])  
print(**'accuracy'**, evaluation[1])