```
from keras.datasets import mnist

data=mnist.load_data()

(x_train,y_train),(x_test,y_test)=data

x_train[0].shape

x_train=x_train.reshape((x_train.shape[0],28*28)).astype('float32')

x_test=x_test.reshape((x_test.shape[0],28*28)).astype('float32')

x_train=x_train/255

x_test=x_test/255

from keras.utils import np_utils

print(y_test.shape)

y_train=np_utils.to_categorical(y_train)

y_test=np_utils.to_categorical(y_test)

num_classes=y_test.shape[1]

print(y_test.shape)

from keras.models import Sequential

from keras.layers import Dense

model=Sequential()

model.add(Dense(32,input_dim=28*28,activation='relu'))

model.add(Dense(64,activation='relu'))

model.add(Dense(10,activation='softmax'))

model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])

model.summary()

model.fit(x_train,y_train,epochs=10,batch_size=100)

scores=model.evaluate(x_test,y_test)

print(scores)
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datase
11493376/11490434 [==============================] - 0s 0us/step
(10000,)
(10000, 10)
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
```

```
=================================================================
dense (Dense)                (None, 32)                25120
_________________________________________________________________
dense_1 (Dense)              (None, 64)                2112
_________________________________________________________________
dense_2 (Dense)              (None, 10)                650
=================================================================
Total params: 27,882
Trainable params: 27,882
Non-trainable params: 0
_________________________________________________________________
Epoch 1/10
600/600 [==============================] - 2s 2ms/step - loss: 0.7534 - accurac
Epoch 2/10
600/600 [==============================] - 1s 2ms/step - loss: 0.2173 - accurac
Epoch 3/10
600/600 [==============================] - 1s 2ms/step - loss: 0.1563 - accurac
Epoch 4/10
600/600 [==============================] - 1s 2ms/step - loss: 0.1262 - accurac
Epoch 5/10
600/600 [==============================] - 1s 2ms/step - loss: 0.1129 - accurac
Epoch 6/10
600/600 [==============================] - 1s 2ms/step - loss: 0.0979 - accurac
Epoch 7/10
600/600 [==============================] - 1s 2ms/step - loss: 0.0913 - accurac
Epoch 8/10
600/600 [==============================] - 1s 2ms/step - loss: 0.0776 - accurac
Epoch 9/10
600/600 [==============================] - 1s 2ms/step - loss: 0.0738 - accurac
Epoch 10/10
600/600 [==============================] - 1s 2ms/step - loss: 0.0685 - accurac
313/313 [==============================] - 0s 1ms/step - loss: 0.1120 - accurac
[0.11204968392848969, 0.968500018119812]
```