```
from keras.datasets import mnist
data = mnist.load_data()


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


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)
```

```
(10000,)
(10000, 10)
```

```
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: "sequential"
```

```
Model: "sequential_7"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_3 (Dense)              (None, 32)                25120
_________________________________________________________________
dense_4 (Dense)              (None, 64)                2112
_________________________________________________________________
dense_5 (Dense)              (None, 10)                650
=================================================================
Total params: 27,882
Trainable params: 27,882
Non-trainable params: 0
_________________________________________________________________
```

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

```
Epoch 1/10
600/600 [==============================] - 1s 2ms/step - loss: 0.0561 - accurac
Epoch 2/10
600/600 [==============================] - 1s 2ms/step - loss: 0.0521 - accurac
Epoch 3/10
600/600 [==============================] - 1s 2ms/step - loss: 0.0488 - accurac
Epoch 4/10
600/600 [==============================] - 1s 2ms/step - loss: 0.0459 - accurac
Epoch 5/10
600/600 [==============================] - 1s 2ms/step - loss: 0.0408 - accurac
Epoch 6/10
600/600 [==============================] - 1s 2ms/step - loss: 0.0387 - accurac
Epoch 7/10
600/600 [==============================] - 1s 2ms/step - loss: 0.0356 - accurac
Epoch 8/10
600/600 [==============================] - 1s 2ms/step - loss: 0.0331 - accurac
Epoch 9/10
600/600 [==============================] - 1s 2ms/step - loss: 0.0305 - accurac
Epoch 10/10
600/600 [==============================] - 1s 2ms/step - loss: 0.0300 - accurac
<tensorflow.python.keras.callbacks.History at 0x7f11d20a0750>
```

```
scores = model.evaluate(x_test,y_test)
print(scores)
```

```
313/313 [==============================] - 0s 949us/step - loss: 0.1186 - accur
[0.1186273917555809, 0.97079998254776]
```

Double-click (or enter) to edit