from \_\_future\_\_ import print\_function

import numpy as np

from keras.datasets import mnist

from keras.models import Sequential

from keras.layers.core import Dense , Activation

from keras.optimizers import SGD

from keras.utils import np\_utils

np.random.seed(1671) #重复性设置

#网络和训练

NB EPOCH = 200

BATCH SIZE = 128

VERBOSE = 1

NB CLASSES = 10 #输出个数等于数字个数

OPTIMIZER = SGD () #SGD 优化器，本章稍后介绍

N HIDDEN = 128

VALIDATION SPLIT=0.2 #训练集中用作验证集的数据比例

#数据：混合并划分训练集和测试集数据

#

(X\_train , y\_train) , (X\_test , y\_test) = mnist.load\_data()

#X train 是60 000 行28 x 28 的数据，变形为60000 × 784

RESHAPE D = 784

#

X\_train = X\_train.reshape(60000 , RESHAPED)

X\_test = X\_test reshape(lOOOO , RESHAPED)

X\_train = X\_train astype ( ’ float32 ’)

X\_test = X\_test astype ( ’ float32 ’ )

#归一化

#

X train /= 255

X test / = 255

print (X\_tra in.shape[O] ，’train samples ’ )

print(X\_tes t .shape[O] ，’ test samples ’ )

#将类向量转换为二值类别矩阵

Y\_train = np\_utils.to\_categorical(y train , NB CLASSES )

Y\_test = np\_utils.to categorical(y test, NB\_CLASSES)

#10 个输出

#最后是softmax 激活函数

model= Sequential()

model.add(Dense(NB\_CLASSES , input\_shape=(RESHAPED , )))

model.add(Activation ( ” softmax ’))

model. summary ()

model.comp工le (l oss ＝ ’ categor工cal\_crossentropy ’， optimizer=OPTIMIZER , metrics= [ ’ accuracy ’])

history= model.fit(X train, Y train ,

batch\_size=BATCH\_SIZE, epochs=NB EPOCH,

verbose=VERBOSE, validation\_split=VA LID ATION SPLIT )

score = model.evaluate(X test, Y test , verbose=VERBOSE)

print (” Test score : ”, score[OJ )

print ( ’ Test accuracy :’ , score[l) )