# 使用Keras解决线性回归问题

## 1.使用Keras的一般步骤

1) 准备数据

2）搭建网络模型

3）编译模型

4）训练

5）测试

## 2.代码

|  |
| --- |
| *#coding=utf-8* **from** keras.datasets **import** mnist **from** keras.models **import** Sequential **from** keras.layers **import** Dense, Dropout, Activation **from** keras.utils **import** np\_utils **from** keras.optimizers **import** RMSprop **import** matplotlib.pyplot **as** plt  classNumber = 10 batchSize = 64 epochNum = 10  *#准备数据* (X\_train, Y\_train), (X\_test, Y\_test) = mnist.load\_data() X\_train = X\_train.reshape(60000, 784) X\_test = X\_test.reshape(10000, 784) X\_train = X\_train/255 X\_test = X\_test/255 Y\_train = np\_utils.to\_categorical(Y\_train, classNumber) Y\_test = np\_utils.to\_categorical(Y\_test, classNumber)  *#创建模型* model = Sequential() model.add(Dense(512, input\_shape=(784,))) model.add(Activation(**'relu'**)) model.add(Dropout(0.2)) model.add(Dense(10)) model.add(Activation(**'softmax'**))  *#打印模型概况* model.summary()  *#编译* model.compile(loss=**'categorical\_crossentropy'**, optimizer=RMSprop(), metrics=[**'accuracy'**])  *#训练，返回history，记录了损失函数及其它指标在训练过程中的变化* history = model.fit(X\_train, Y\_train, batch\_size=batchSize, nb\_epoch=epochNum, verbose=1,  validation\_data=(X\_test, Y\_test))  *#绘制准确率曲线* plt.plot(history.history[**'acc'**]) plt.plot(history.history[**'val\_acc'**]) plt.title(**'model accuracy'**) plt.ylabel(**'accuracy'**) plt.xlabel(**'epoch'**) plt.legend([**'train'**, **'test'**], loc=**'upper left'**) plt.show()  *#测试* score = model.evaluate(X\_test, Y\_test, verbose=0) print(**'Test score:'**, score[0]) print(**'Test accuracy:'**, score[1]) |



