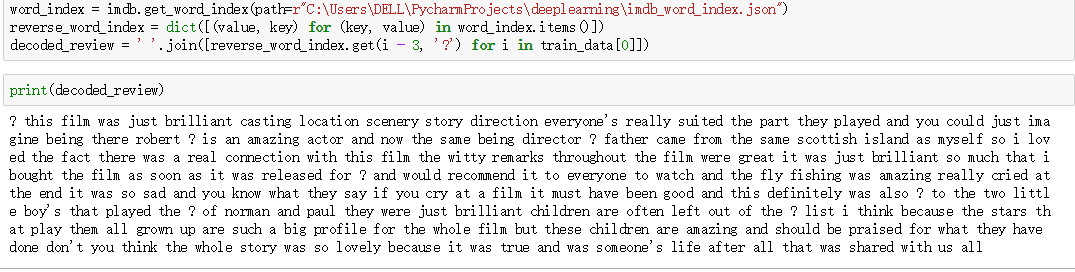
练习2：lmdb数据集练习 评论的二分类

将影评解码输出（0、1、2对应的是特殊符号，统一解成‘？’）



from keras.datasets import imdb

(train\_data,train\_labels),(test\_data,test\_labels)=imdb.load\_data(path=r"C:\Users\DELL\PycharmProjects\deeplearning\imdb.npz",num\_words=10000)

import numpy as np

def vectorize\_sequences(sequences, dimension=10000):

results = np.zeros((len(sequences), dimension))

for i, sequence in enumerate(sequences):

results[i, sequence] = 1.

return results

x\_train = vectorize\_sequences(train\_data)

x\_test = vectorize\_sequences(test\_data)

y\_train=np.asarray(train\_labels).astype('float32')

y\_test=np.asarray(test\_labels).astype('float32')

from keras import models

from keras import layers

model = models.Sequential()

model.add(layers.Dense(16,activation='relu',input\_shape=(10000,)))

model.add(layers.Dense(16,activation='relu'))

model.add(layers.Dense(1,activation='sigmoid'))

model.compile(optimizer='rmsprop',loss='binary\_crossentropy',metrics=['accuracy'])

#留出验证集，原始训练数据留出10000个样本作为验证集

x\_val=x\_train[:10000]

partial\_x\_train=x\_train[10000:]

y\_val=y\_train[:10000]

partial\_y\_train=y\_train[10000:]

history=model.fit(partial\_x\_train,

partial\_y\_train,

epochs=20,

batch\_size=512,

validation\_data=(x\_val,y\_val))

#绘制训练损失和验证损失

import matplotlib.pyplot as plt

history\_dict=history.history

loss\_values=history\_dict['loss']

val\_loss\_values=history\_dict['val\_loss']

epochs=range(1,len(loss\_values)+1)

plt.plot(epochs, loss\_values, 'bo', label='Training loss')

plt.plot(epochs, val\_loss\_values, 'b', label='Validation loss')

plt.title('Training and validation loss')

plt.xlabel('Epochs')

plt.ylabel('Loss')

plt.legend()

plt.show()

#绘制训练精度和验证精度

plt.clf()

acc = history\_dict['acc']

val\_acc = history\_dict['val\_acc']

plt.plot(epochs, acc, 'bo', label='Training acc')

plt.plot(epochs, val\_acc, 'b', label='Validation acc')

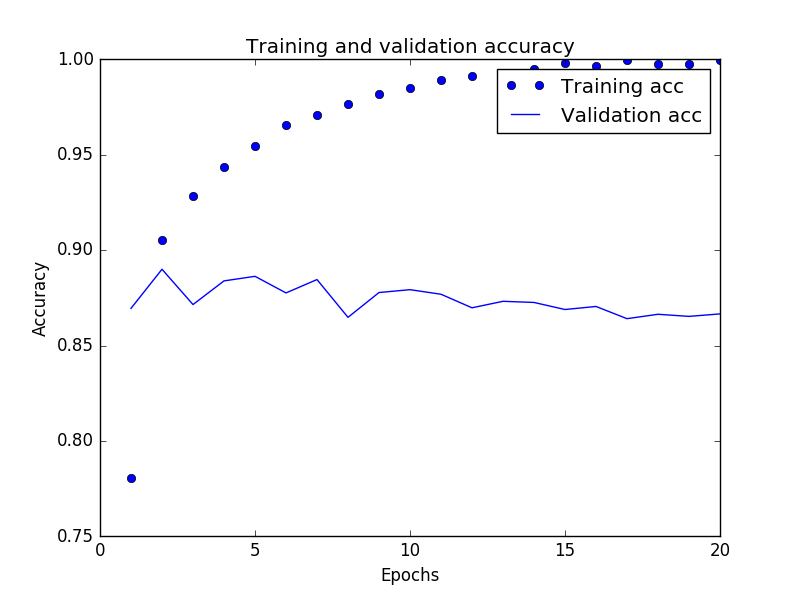
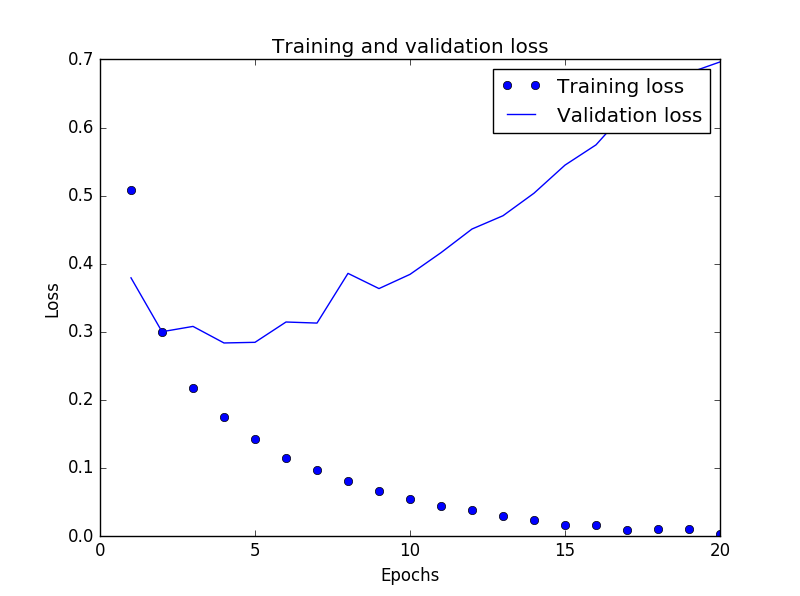
plt.title('Training and validation accuracy')

plt.xlabel('Epochs')

plt.ylabel('Accuracy')

plt.legend()

plt.show()



#精度输出结果

可见训练损失越来越少，精度越来越高

但是在测试集上的表现却不是如此，发生了过拟合，应该减少训练次数。将epochs=4 重新训练。