# Keras Mnist手写数字识别练习

本实验是在anaconda+TensorFlow+Keras环境下实现的，具体配置步骤详见<https://blog.csdn.net/m0_37788308/article/details/80272803>。

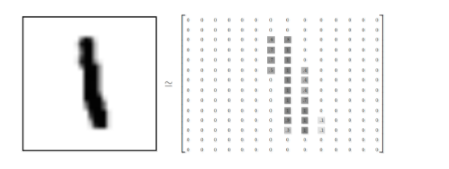
本次实验内容主要包括三部分内容：一是了解Keras Mnist手写数字识别数据集，二是了解Keras多层感知器识别手写数字的过程，三是了解Keras卷积神经网络识别手写数字的过程。

## 一、Keras Mnist手写数字识别数据集

该部分实践操作代码请参照“mnist第6章.ipynb”

### Mnist数据集简介

MNIST（Mixed National Institute of Standards and Technology database）是一个计算机视觉数据集，它包含70000张手写数字的灰度图片，其中每一张图片包含 28 X 28 个像素点。可以用一个数字数组来表示这张图片：



每一张图片都有对应的标签，也就是图片对应的数字，例如上面这张图片的标签就是 1。数据集被分成两部分：60000 行的训练数据集（mnist.train）和10000行的测试数据集（mnist.test）。其中：60000 行的训练集分拆为 55000 行的训练集和 5000 行的验证集。

### 实验部分

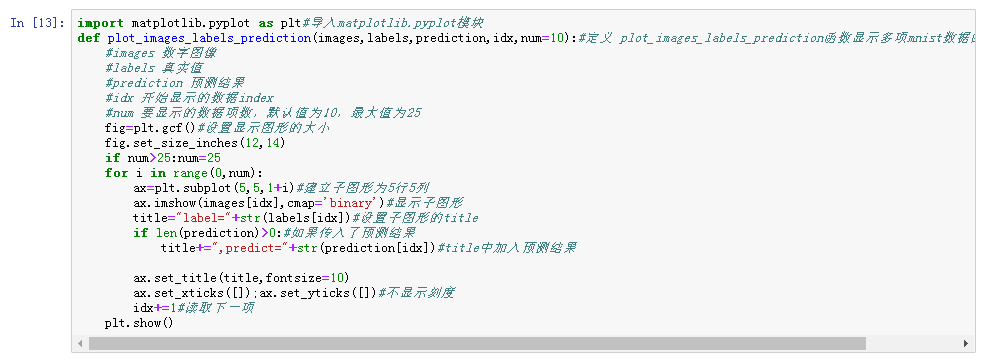
1. 下载并查看mnist数据

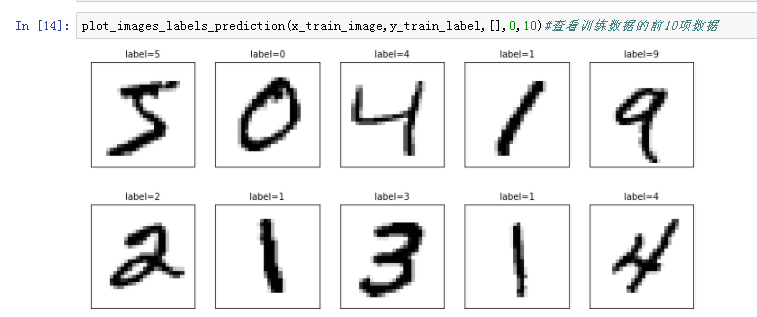


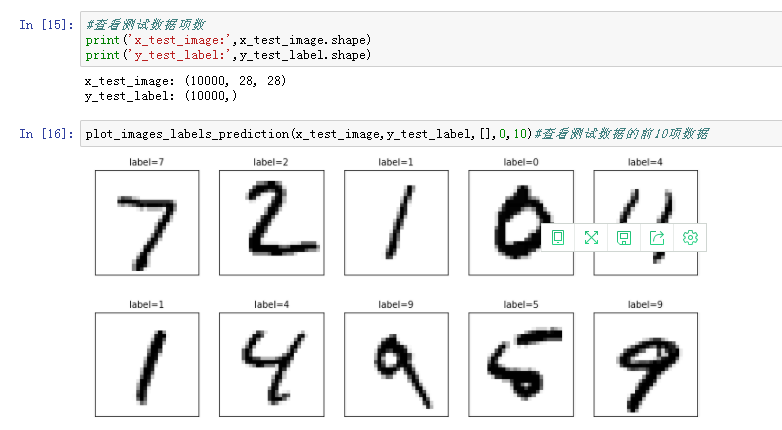
1. 查看训练数据



1. 查看多项训练数据images与label

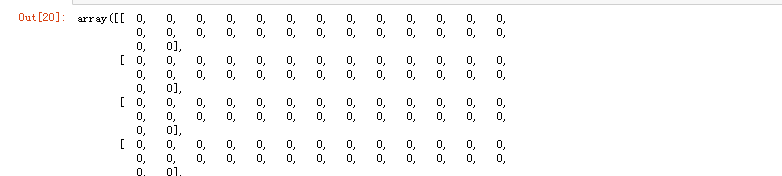


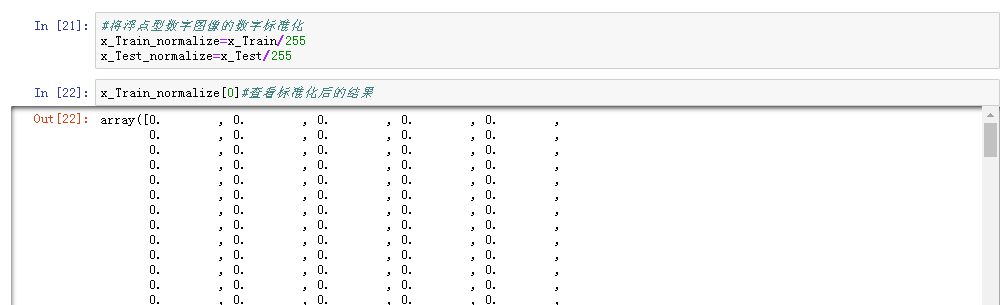




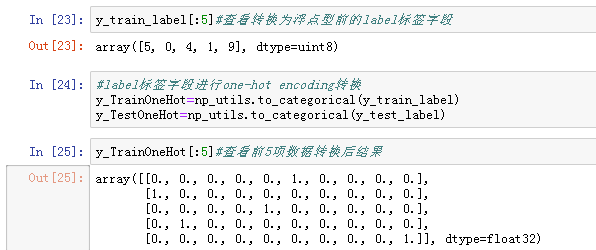
1. features数据预处理







1. label数据预处理



## 二、Keras多层感知器识别手写数字

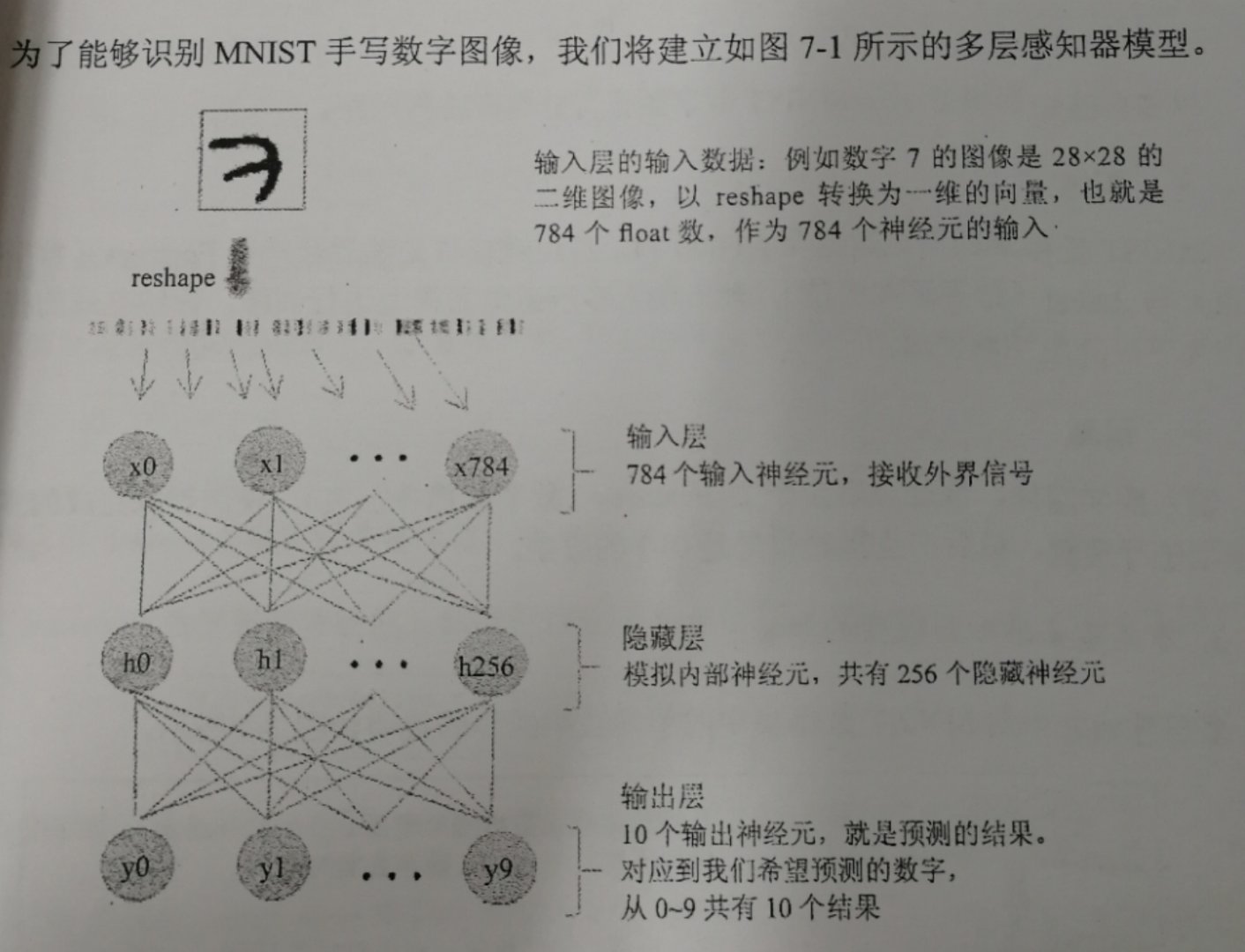
该部分实践操作代码请参照“mnist第7章$7.1-7.8.ipynb、mnist第7章$7.9.ipynb、mnist第7章$7.10.ipynb”

### Keras多层感知器简介

多层感知器（MulTI Layer Perceptron，即 MLP）包括至少一个隐藏层（除了一个输入层和一个输出层以外）。单层感知器只能学习线性函数，而多层感知器也可以学习非线性函数。

多层感知器模型原理详见<http://www.elecfans.com/rengongzhineng/579844_2.html>。

多层感知器模型如下图所示：



### 实验部分

1）常规过程：

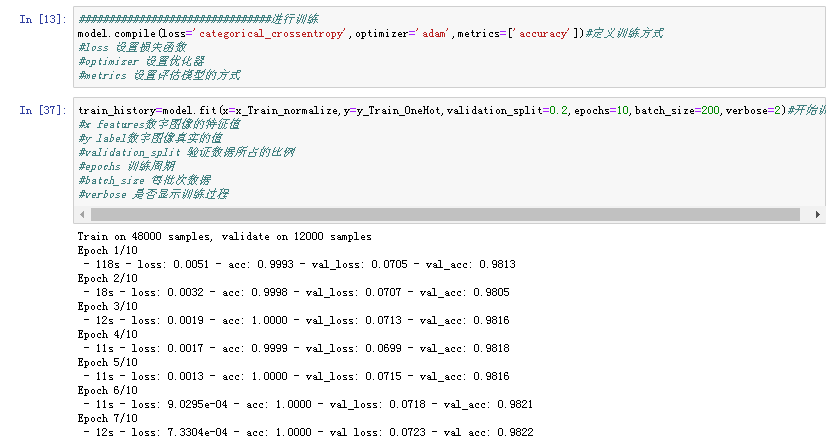
①进行数据预处理

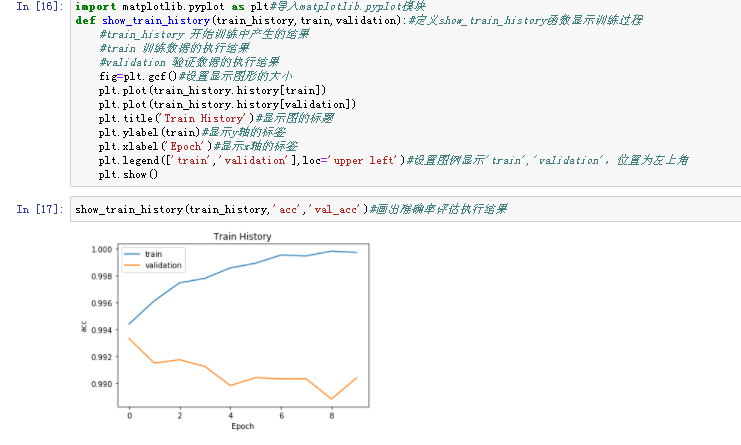


②建立模型



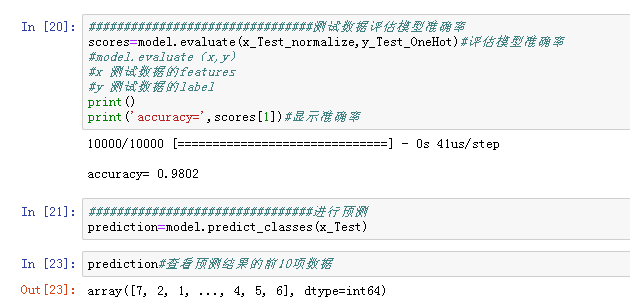
③进行训练

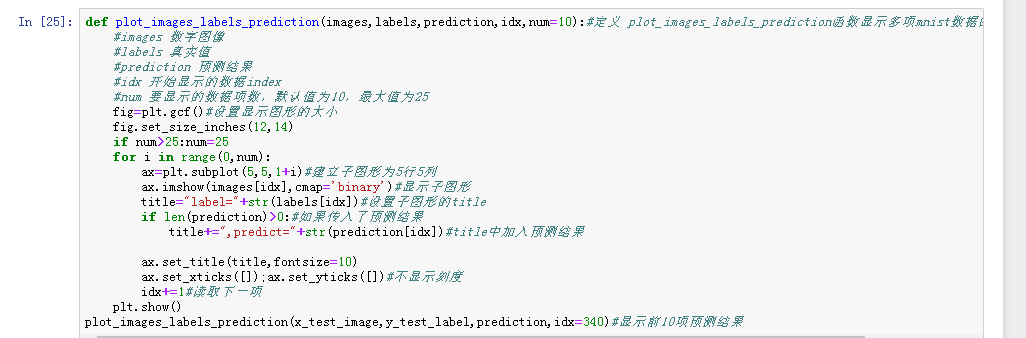


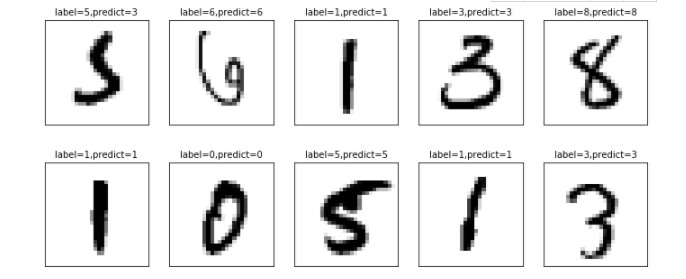




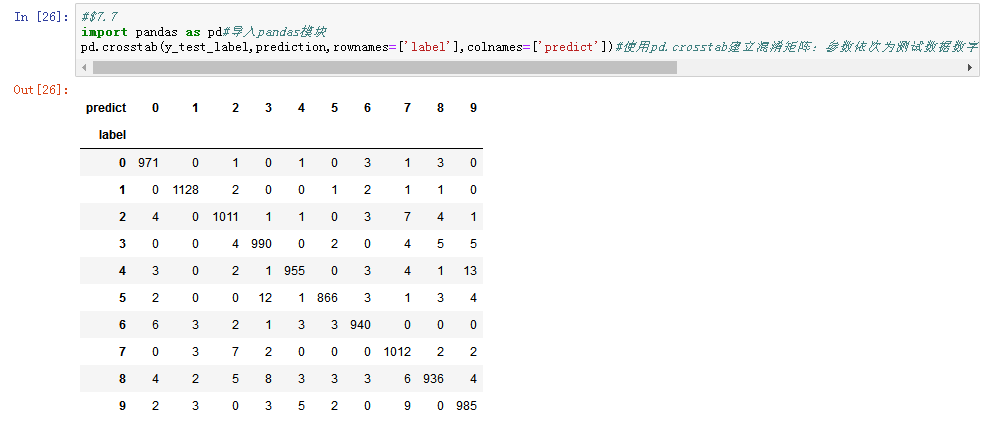
④以测试数据评估模型准确率

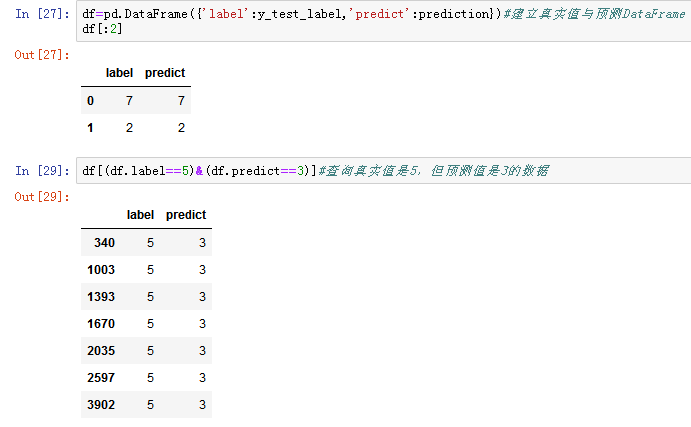


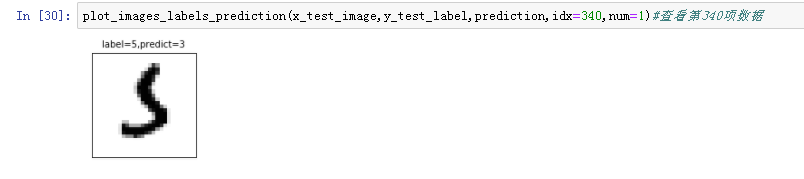




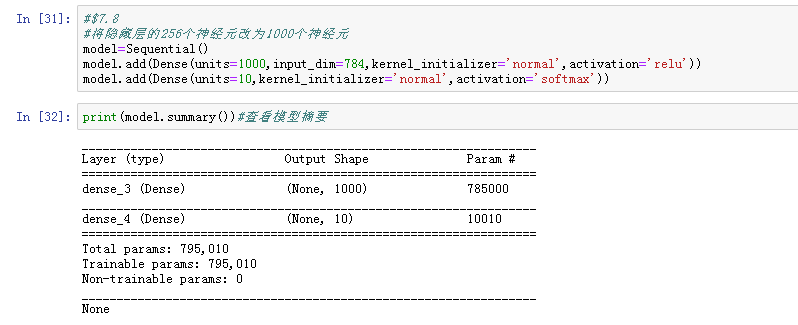
2）显示混淆矩阵：

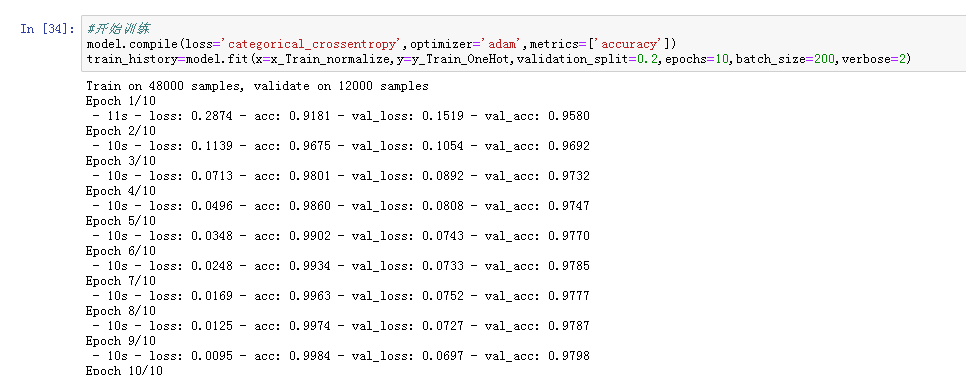


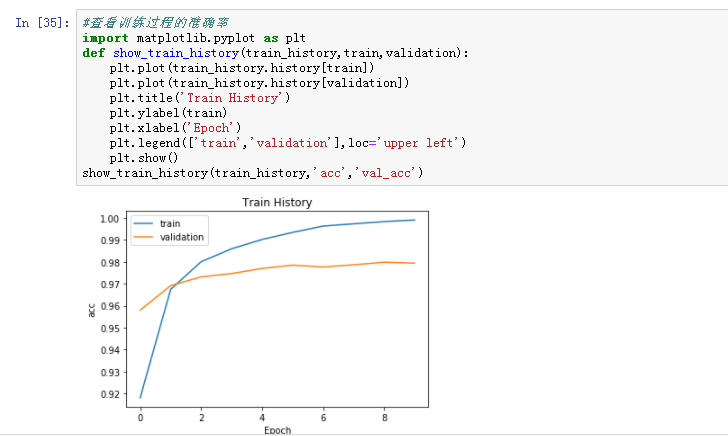


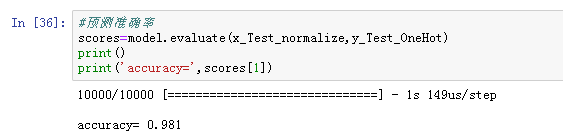


3）隐藏层增加为1000个神经元：







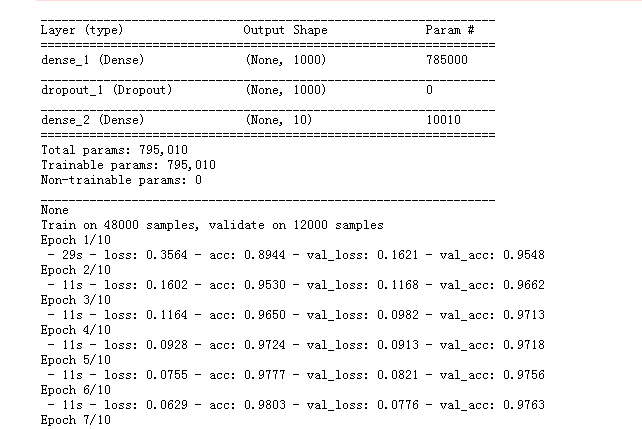


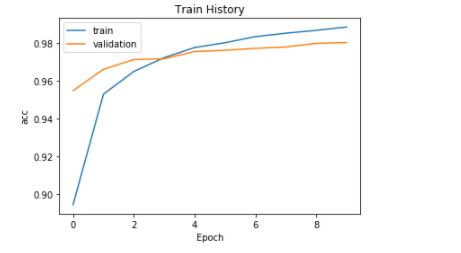
4）加入DropOut功能避免过度拟合：

在建立模型部分加入DropOut功能，其他部分不变。



得到结果如下：

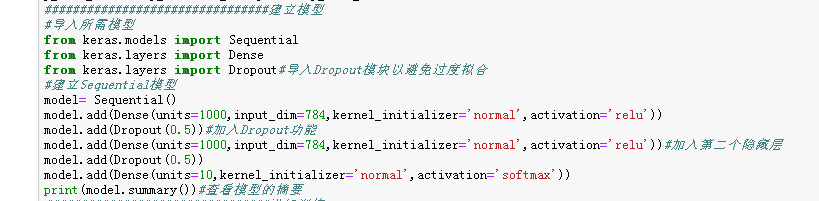




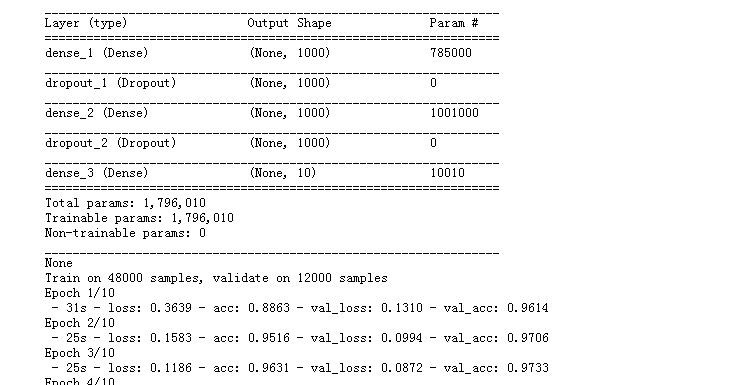


5）建立多层感知器模型中包含两个隐藏层

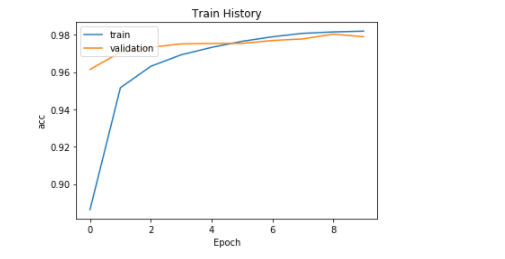
在建立模型部分增加一个隐藏层，其他部分不变。



得到结果如下：







## 三、Keras卷积神经网络识别手写数字

该部分实践操作代码请参照“mnist第8章.ipynb”

### 卷积神经网络简介

多层感知器与卷积神经网络主要的差异是：卷积神经网络增加了卷积层1、池化层1、卷积层2、池化层2的处理提取特征。

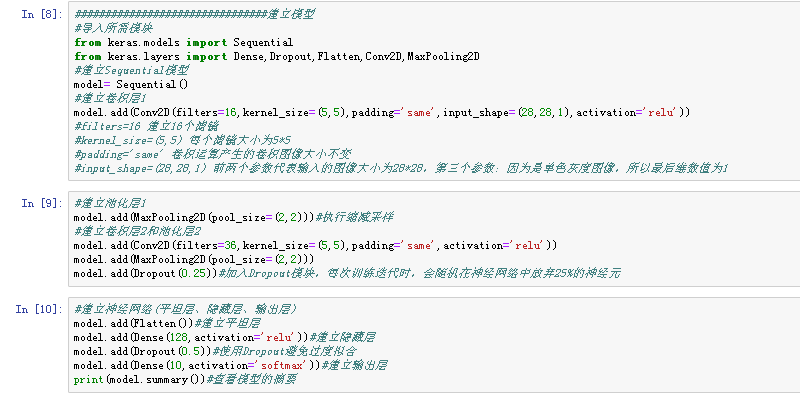
神经网络模型原理详见：<https://xudonghecs.github.io/2018/08/19/%E5%8D%B7%E7%A7%AF%E7%A5%9E%E7%BB%8F%E7%BD%91%E7%BB%9C%E6%A8%A1%E5%9E%8B%E7%AE%80%E8%BF%B0/>。

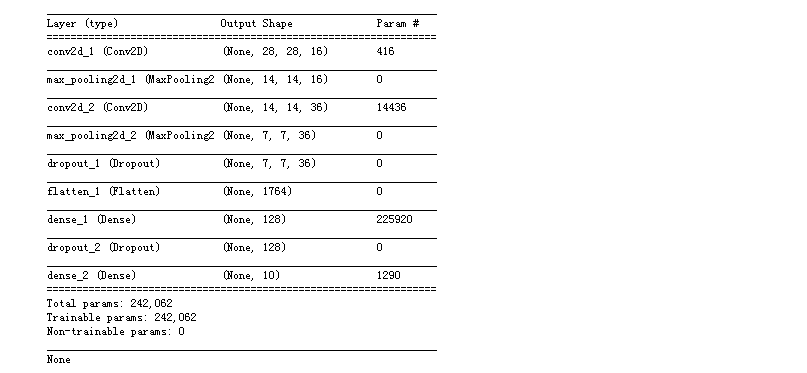
### 实验部分

①进行数据预处理

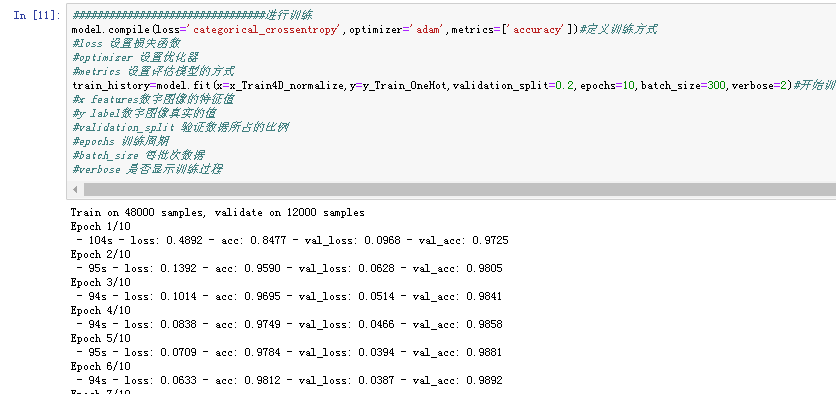


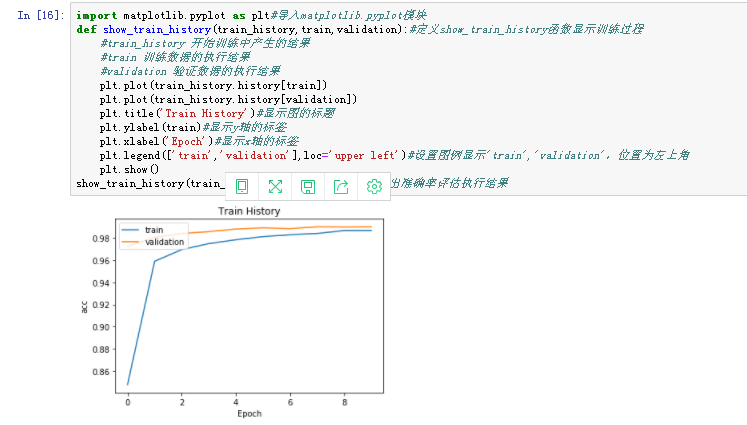
②建立模型

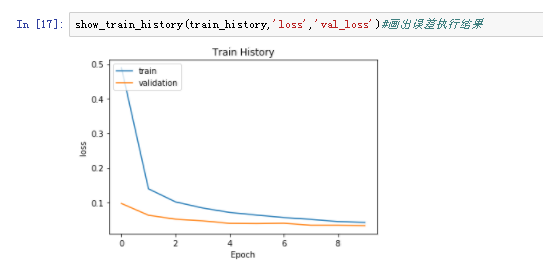




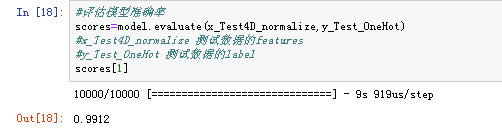
③进行训练



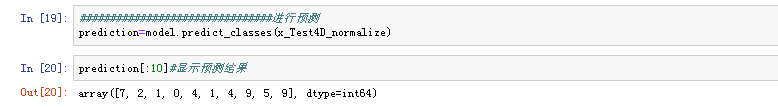


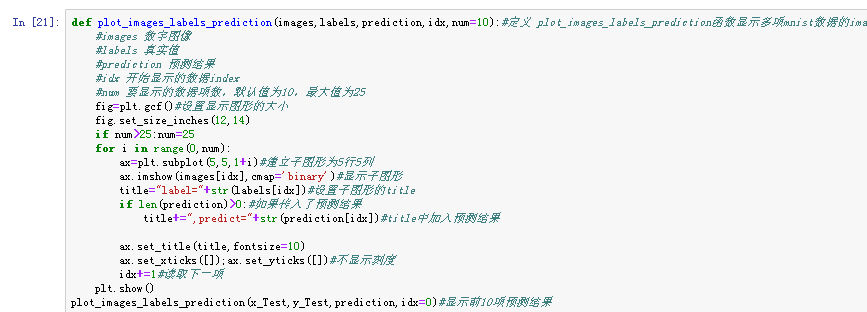


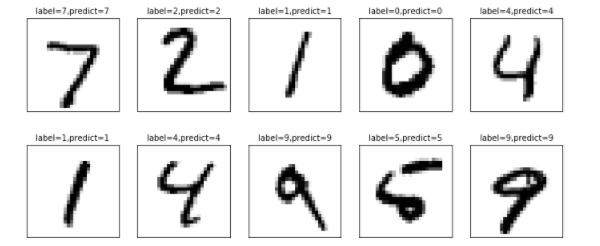
④评估模型准确率



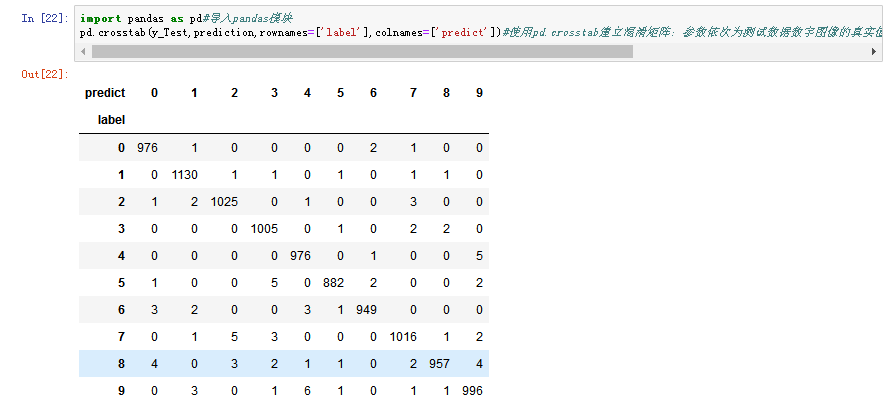
⑤进行预测







⑥显示混淆矩阵



## 四、引入库简单说明

### numpy模块

numpy是python中一个处理数组的强大模块，且该模块也是其他数据分析模块（如pandas和scipy）的核心。常用部分主要包括以下五个方面内容：

1）数组的创建

2）有关数组的属性和函数

3）数组元素的获取--普通索引、切片、布尔索引和花式索引

4）统计函数与线性代数运算

5）随机数的生成

### pandas模块

pandas 是基于numpy 的一种工具，该工具是为了解决数据分析任务而创建的。Pandas 纳入了大量库和一些标准的数据模型，提供了高效地操作大型数据集所需的工具。pandas提供了大量能使我们快速便捷地处理数据的函数和方法。常用部分主要包括以下八个方面内容：

1）数据结构：DataFrame和Series

2）数据索引index

3）利用pandas查询数据

4）利用pandas的DataFrames进行统计分析

5）利用pandas实现SQL操作

6）利用pandas进行缺失值的处理

7）利用pandas实现Excel的数据透视表功能

8）多层索引的使用

### Keras

Keras是一个高层神经网络API，Keras由纯Python编写而成并基Tensorflow、Theano以及CNTK后端，也就是Keras基于什么东西来做运算。Keras主要包括以下几个模块：

1）layers：层

2）models：模块

3）objectives：目标函数

4）optimizers：优化器

5）utils：本模块提供了一系列有用工具

可参考Keras的中文文档以及<https://blog.csdn.net/qq_35082030/article/details/77170284>；

<https://blog.csdn.net/zjm750617105/article/details/51302299>；

<https://blog.csdn.net/lk7688535/article/details/52862377>。

如Keras.model主要包括两种如下：

1）序贯模型

序贯模型是多个网络层的线性堆叠，也就是“一条路走到黑”。这也是非常常用的和傻瓜式的方法。

2）函数式（Functional）模型

Keras函数式模型接口是用户定义多输出模型、非循环有向模型或具有共享层的模型等复杂模型的途径。一句话，只要你的模型不是类似VGG一样一条路走到黑的模型，或者你的模型需要多于一个的输出，那么你总应该选择函数式模型。函数式模型是最广泛的一类模型，序贯模型（Sequential）只是它的一种特殊情况。

### matplotlib.pyplot模块

Python优秀的数据可视化第三方库。Matplotlib库由各种可视化类构成，内部结构复杂，受Matlab启发；matplotlib.pyplot是绘制各类可视化图形的命令子库，相当于快捷方式。

pyplot的plot函数：  plt.plot(x,y, format\_string, \*\*kwargs)

参数说明：x  : X轴数据，列表或数组，可选；y  : Y轴数据，列表或数组； format\_string: 控制曲线的格式字符串，可选；\*\*kwargs : 第二组或更多(x,y,format\_string)。

## 五、总结

本次实验虽然操作起来比较简单，但里面包含的原理以及软件安装对于刚入门的小白来说都有一定的难度。在整个实验过程中最大的问题就是在anaconda建立的虚拟环境中安装好TensorFlow后无法成功导入（总出现错误），也没有找到很好的解决方法，只能换个电脑安装。最需要注意的问题是不要因为马虎敲错代码，当运行时出现错误首先要检查自己的代码是不是敲错了。本次实验最大的收获就是通过动手实践能够了解一些深度学习的建模原理。