**实验2 Minist数据集分类**

202013407043 张艺汶

1. **实验目的：**
2. 使用python构建一个简单的BP神经网络；
3. 在Minist数据集上进行训练；
4. 使用训练的网络进行前向推理和测试。
5. **实验内容：**

1. 定义神经网络的输入层、隐藏层、学习率、训练迭代等参数；

2. 训练、测试构建的BP神经网络；

3. 从文件夹中读取一张图片并进行判断分类。

1. **实验指导：**

计算机神经网络则是人工智能中最为基础的也是较为重要的部分，它使用深度学习的方式模拟了人的神经元的工作，是一种全新的计算方法。本文的目标就是通过学习神经网络的相关知识，了解并掌握BP神经网络的实现原理和构造方法，建立一个简单的BP神经网络，并用MNIST数据集训练该网络，使训练后的网络能够成功的分类出MNIST测试数据集上的数字，并能识别从文件中读入的图片上的数字。

**1. BP网络设计**

反向传播算法,即Back Propagation是建立在梯度下降算法基础上，适用多层神经网络的参数训练方法。由于隐藏层节点的预测误差无法直接计算,因此,反向传播算法直接利用输出层节点的预测误差反向估计上一层隐藏节点的预测误差,即从后往前逐层从输出层把误差反向传播到输入层,从而实现对链接权重调整,这也是反向传播算法名称的由来。

一个典型的3层BP神经网络模型如下图所示:

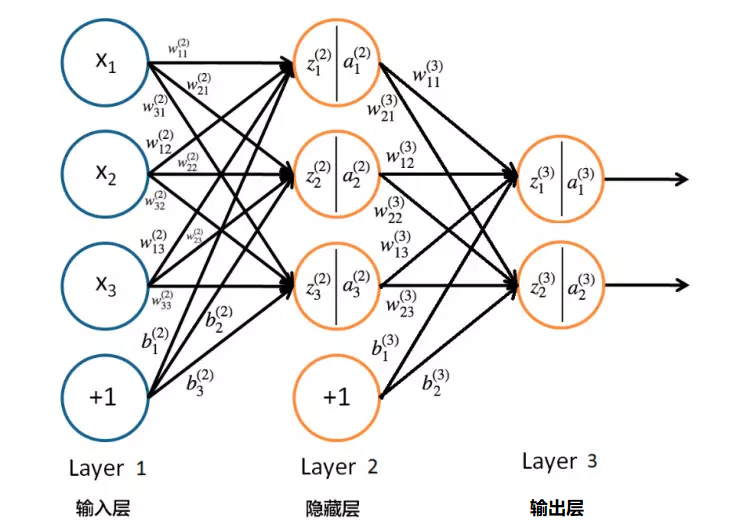


图1 三层神经网络示意图

Mnist数据集的测试图片像素是28X28的，所以输入节点的个数就是28X28=784；识别出的数字有0-9十个数字，所以输出的节点的个数设置为10个；因为输入层的节点较多，所以隐藏层的节点个数设置为100；

考虑到梯度下降算法能够较好的消除产生的误差，所以激活函数设置为sigmoid函数；学习率设置为0.2，太高或太低都会导致不同的问题（梯度爆炸、梯度消失）；训练轮数设置为5

因此，网络设计的参数为：

输入层节点数:784；

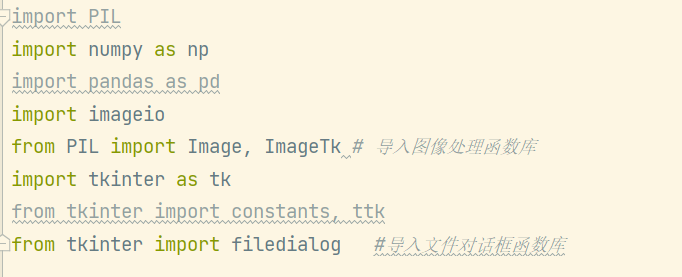
隐藏层节点数：200；

输出层节点数：10

学习率：0.1；

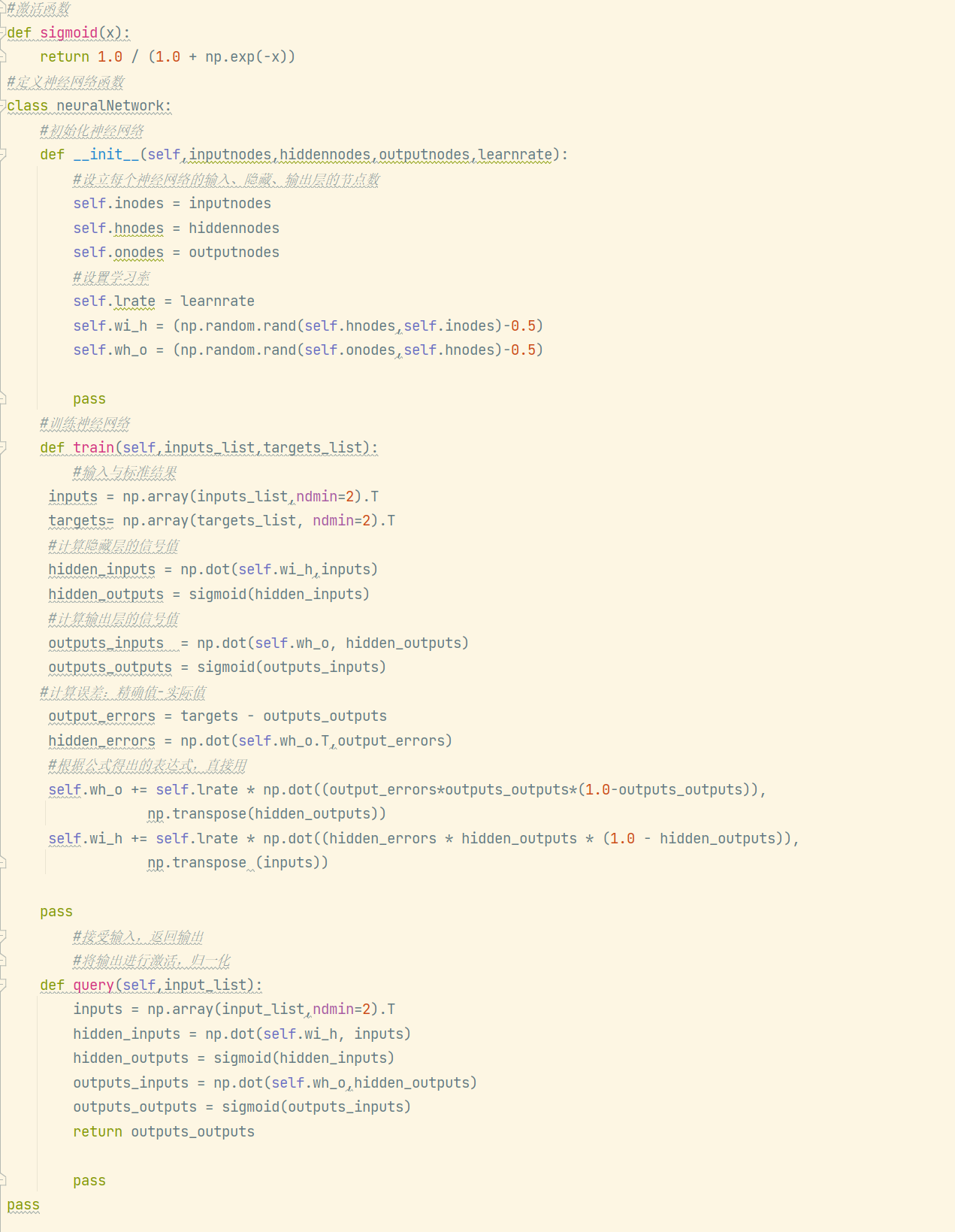
训练世代：5；

激活函数：sigmoid函数



**代码实现：**

#———————————神经网络构建，三层结构————————————————#



这部分对BP神经网络类进行了参数的定义和对训练、激活函数进行了定义

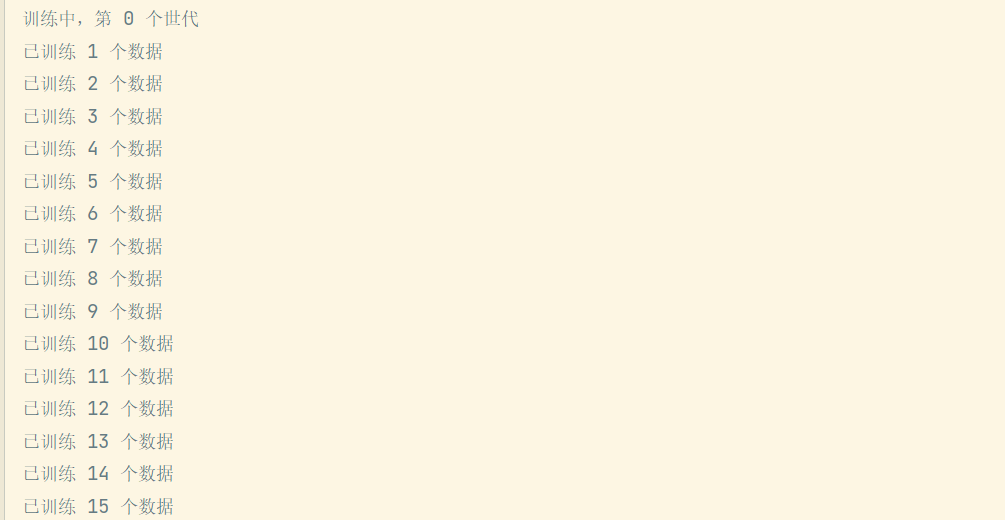
#—————————————创建神经网络对象并用数据集训练网络——————————#

以上代码的作用是使用函数创建神经网络对象，并加载数据集



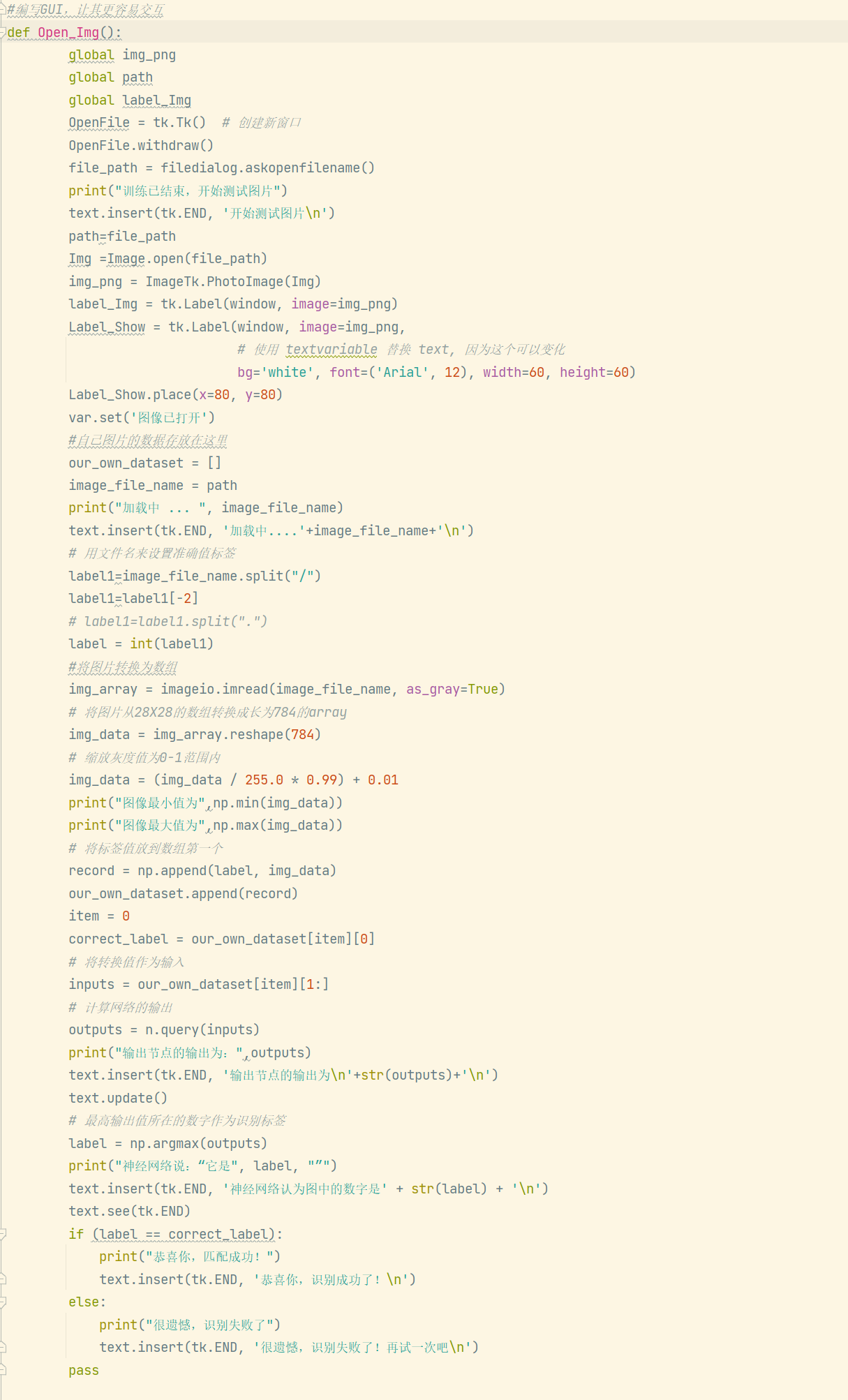
以上代码的作用则是对数据集测试功能的实现，它可以遍历测试集中的所有测试图片，并得出最终正确率。到这一步，其实我们已经实现了识别手写数字的功能了，我们的目标已经完成了。接下来就是完善它，让它更加实用了。

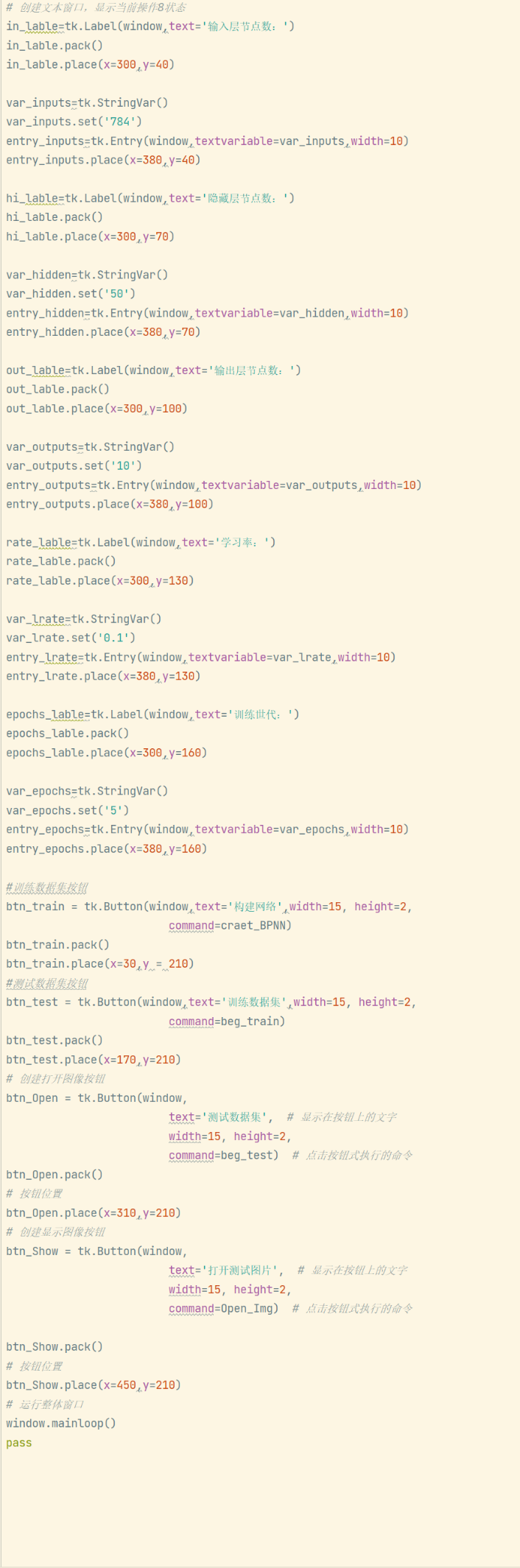
**以上步骤替换代码图，及输出的训练epcho截图。**



由于数量太多，仅截取部分数据。

#打开图片的函数，并尝试识别自己的图片



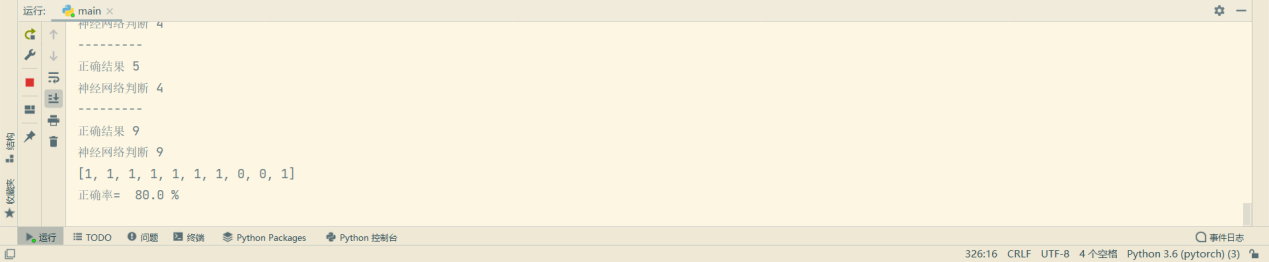


以上代码实现了GUI的编写，让用户自行从文件中读取图片并识别，并由网络给出识别结果。因为GUI比较繁琐，所以代码看起来偏长。不过和最终效果比起来，这点付出是值得的。我们的功能也已经实现完毕了，接下来就看一下具体效果了。

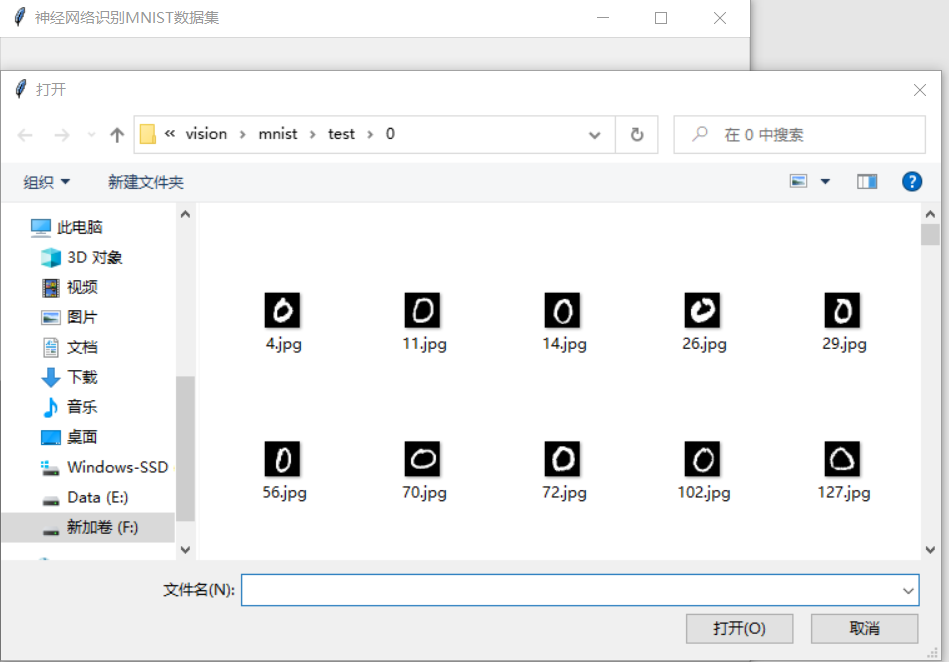
**实现效果**



这是程序运行时的界面，可以对参数进行自定义的输入



在控制台界面也能看到每张图片的识别结果，1是识别正确，0是识别错误。可以看到识别的正确率还是挺高的，由97%。



这里是用户自行选择图片进行识别，可以自己写然后进行识别，但前提是图像尺寸必须是28X28，如果尺寸打了就必须对其进行池化到28X28的大小，否则就会导致输入参数量巨大（几万乃至几十万个输入参数），那就不是BP神经网络可以解决的问题了，就必须要用到卷积深度神经网络进行特征提取再来分类了。



可以看到，我们建立的神经网络已经成果的识别了我们手写的数字了。

**GUI能在测试时读入图像、显示图像，并显示识别结果即可。**

1. **实验小结：**

本次实验中自定义的函数为：

1. train，用于训练神经网络；
2. query，用于接受输入，返回输出，并将输出进行激活，归一化；
3. craet\_BPNN，用于创建神经网络对象并用数据集训练网络；
4. beg\_train，用于训练MNist数据集；
5. beg\_test，用于测试MNist数据集；
6. Open\_Img，用于打开图片的函数，并尝试识别自己的图片并且编写GUI界面，便于交互；
7. SHOW，用于显示图片；

通过本次实验，我学会了使用python构建一个简单的BP神经网络，在Minist数据集上进行训练，并使用训练好的网络向前推理和预测，收获很大。