**计算机视觉实验**

**实验2 Minist数据集分类**

**姓名：龚思宇**

**班级：计科2002班**

**学号：202001125051**

**实验2 Minist数据集分类**

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:

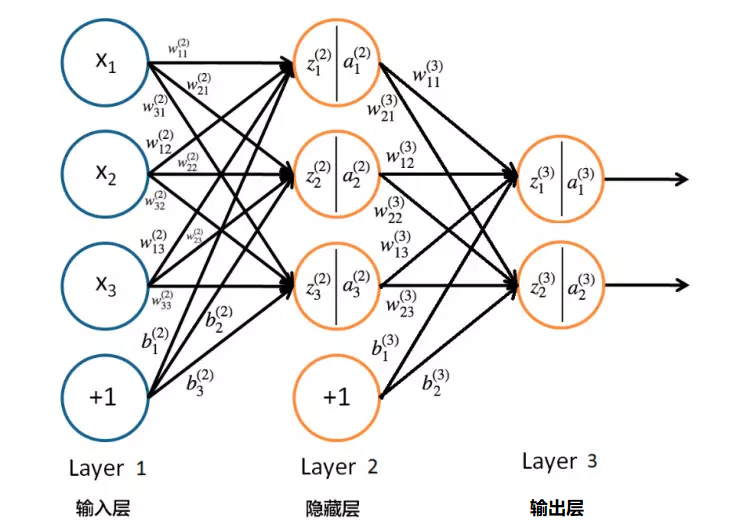


图1 三层神经网络示意图

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

import PIL

import numpy as np

import pandas as pd

import imageio

from PIL import Image, ImageTk # 导入图像处理函数库

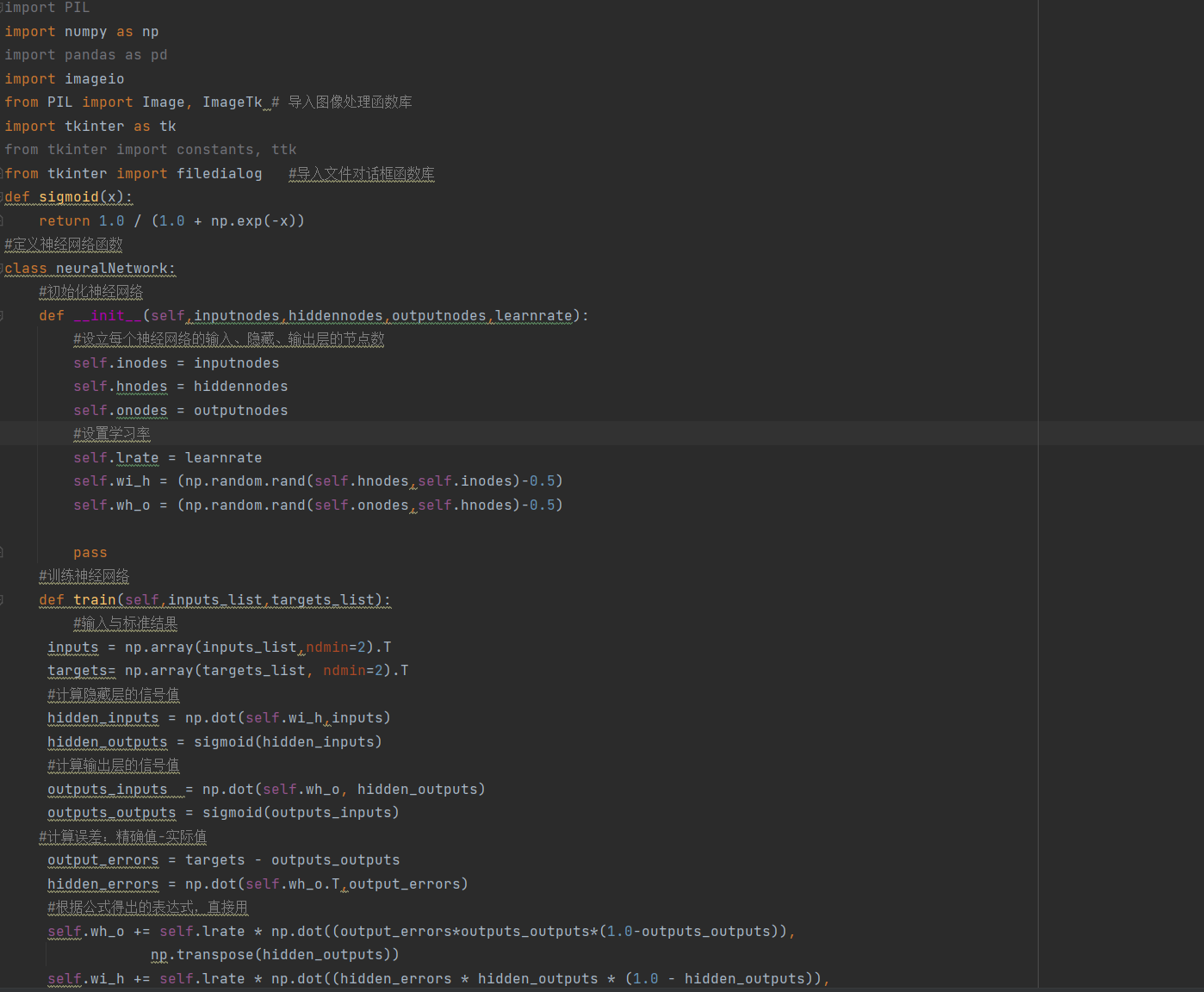
import tkinter as tk

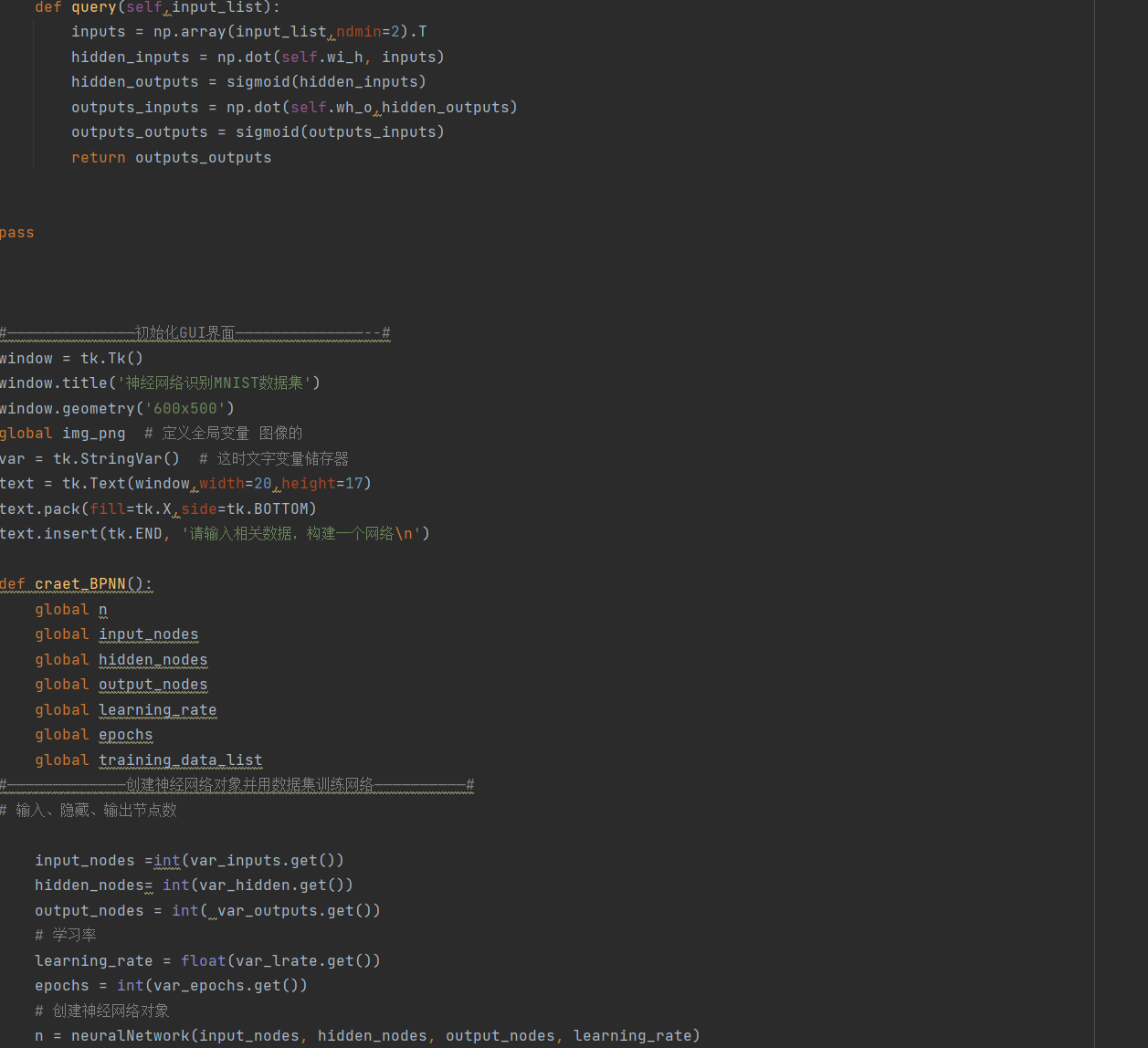
from tkinter import constants, ttk

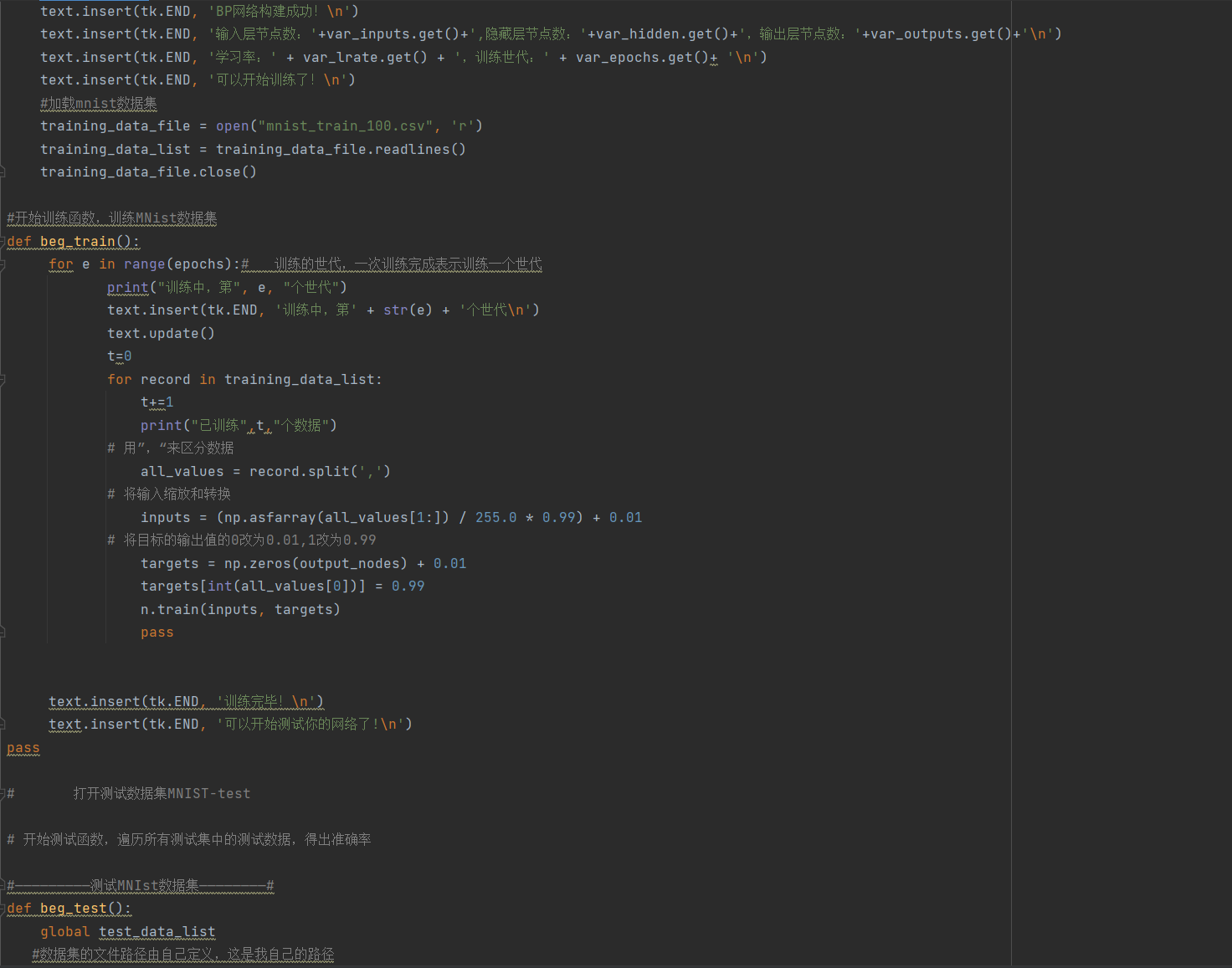
from tkinter import filedialog #导入文件对话框函数库

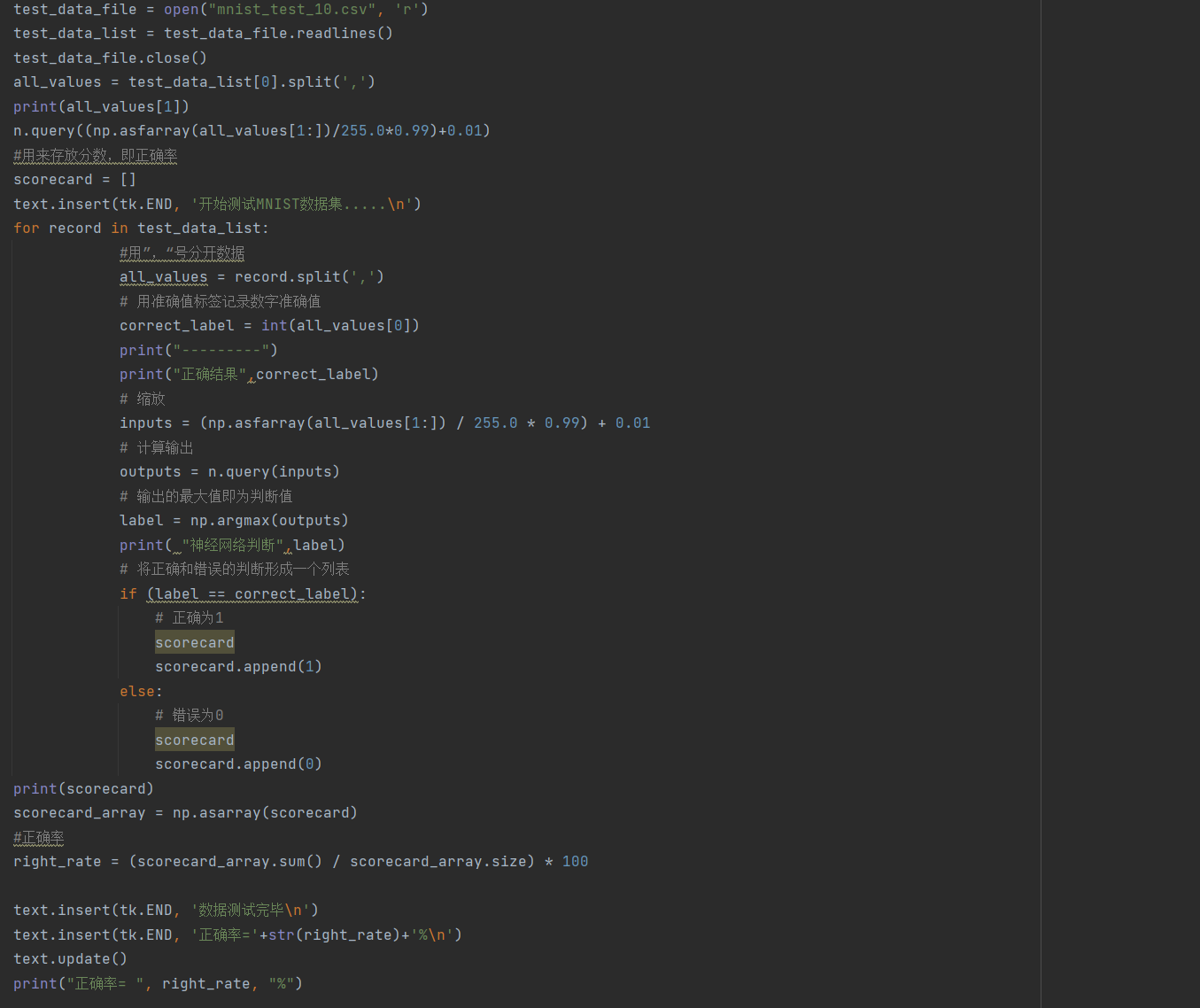
**代码实现：**

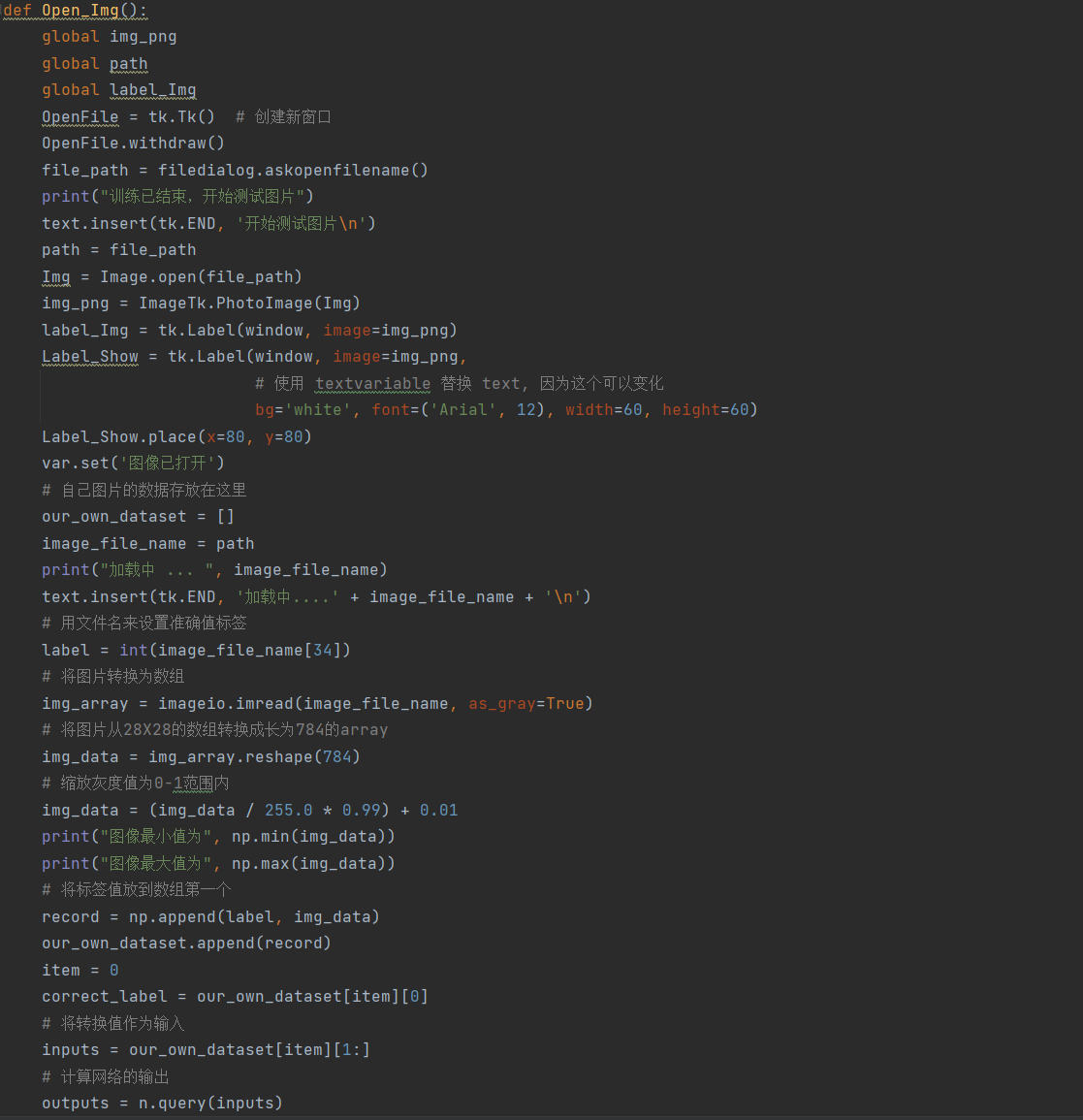
#———————————神经网络构建，三层结构————————————————#

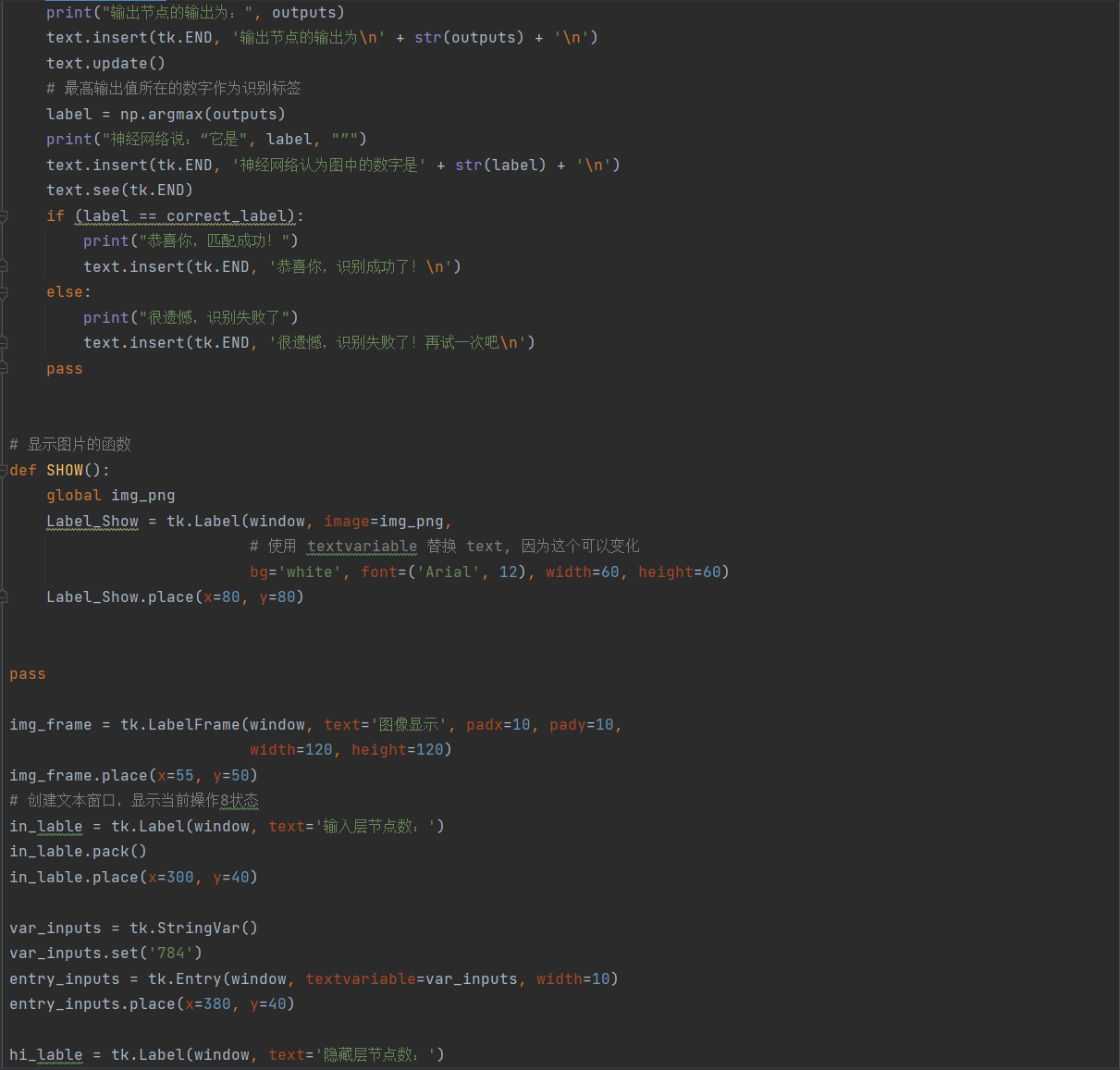
****

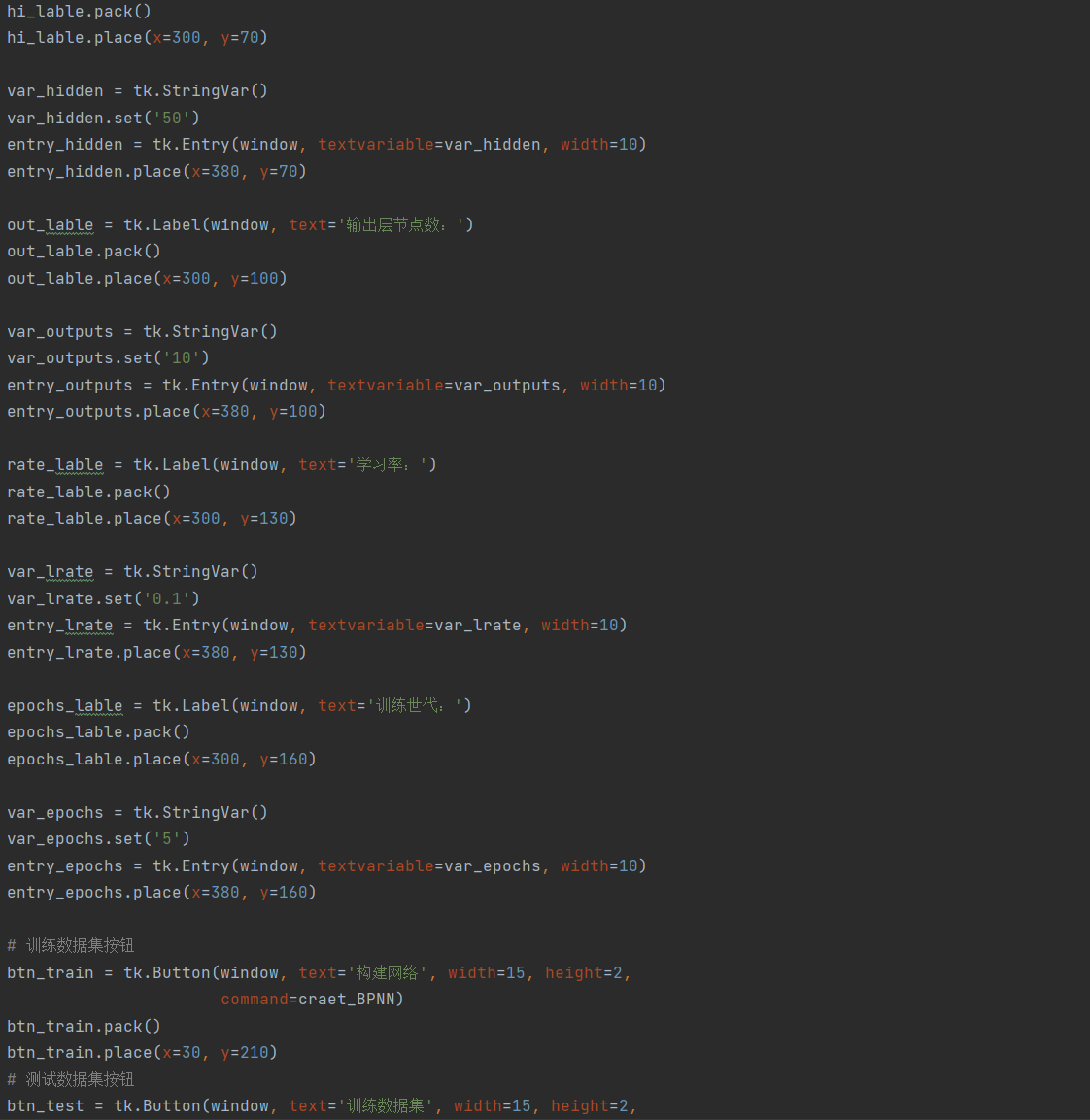
****

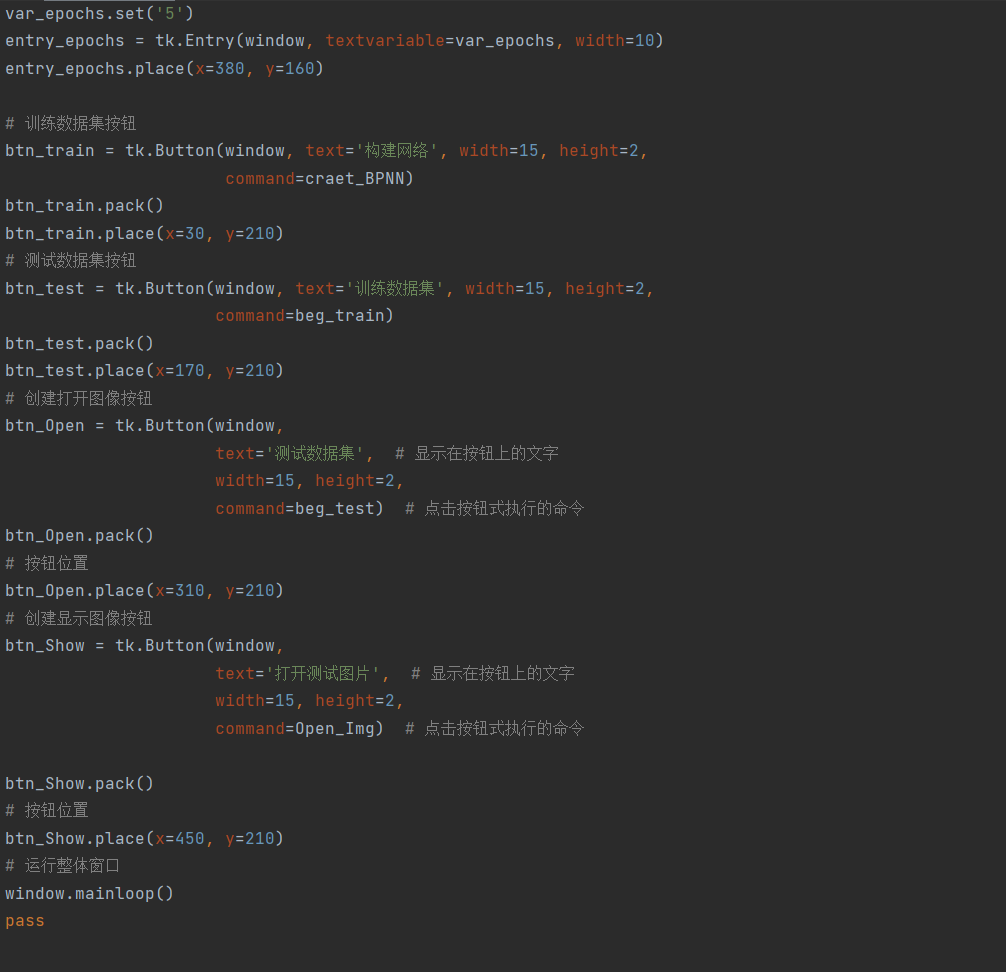
****

****

****

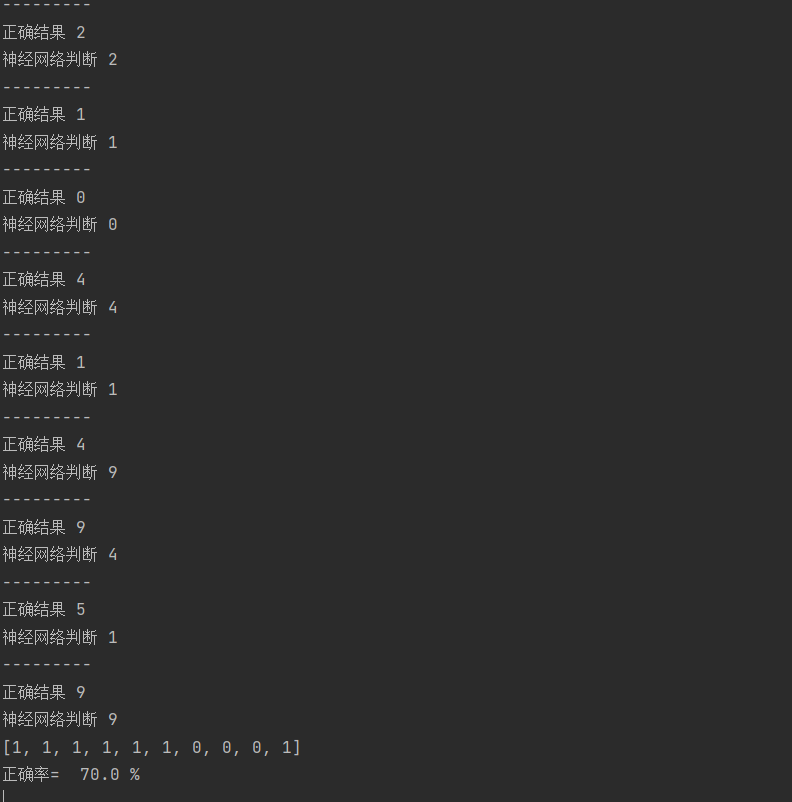
****

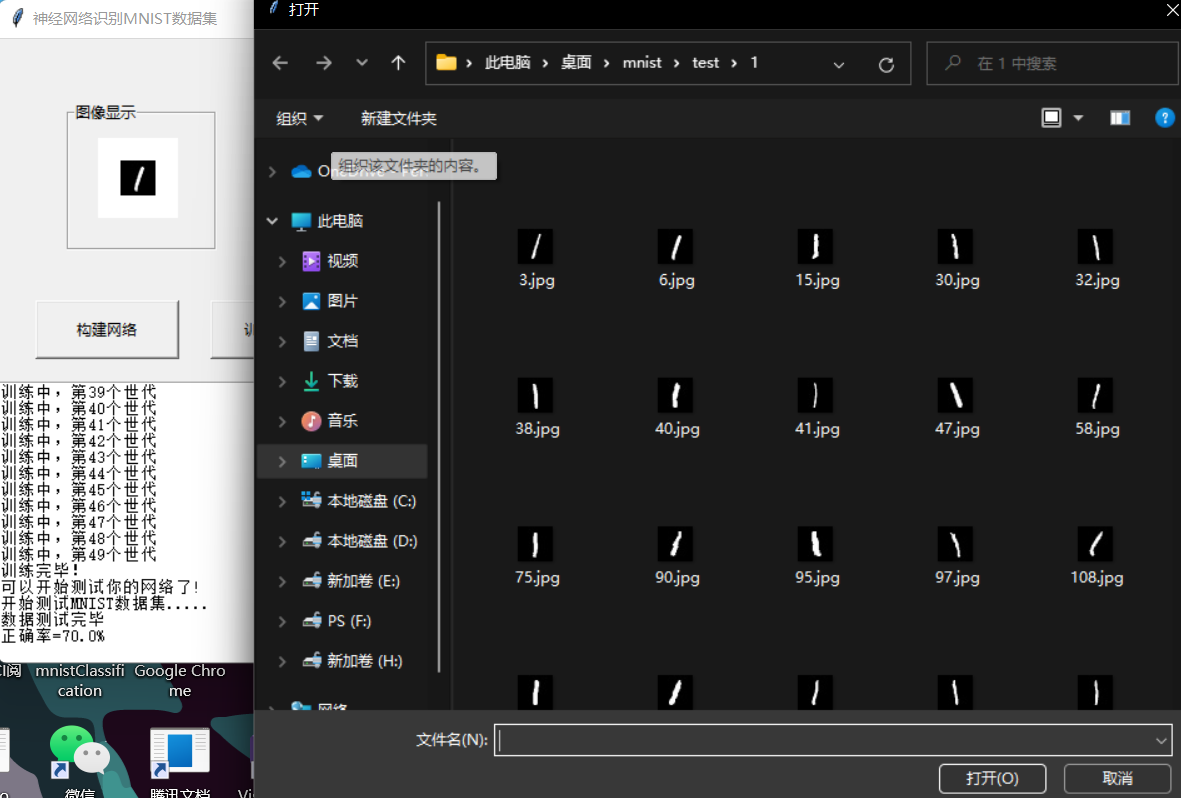
****

****

****

**实现效果**





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。