**机器学习实验报告**

**（四）**

****

姓名：崔玉峰

学号：201600301079

班级：2016级4班

1. **实验目的**

通过BP神经网络对Mnist数据集进行分类，并用两种不同的BP模型做性能对比

1. **实验思路**

BP神经网络是使用反向传播算法进行训练的一种神经网络，在本次实验中建立的BP神经网络模型为单隐层激活函数为Sigmoid的神经网络，下面开始本实验模型的相关公式的推导：

1. **一个神经元的输出可以表示为：**



其中w为权值，x为输入，y为输出，()为sigmoid函数

**训练算法：**

1. **在本次实验中假设输入层的神经单元数为d，隐层神经单元数为q，输出层的神经单元数为l，那么对模型进行训练主要为了优化输入层到隐层d\*q个权值；隐层到输出层q\*l个权值。**

BP神经网络基于梯度下降策略：

1. 设当前神经网络的输出为，期望输出为

则此网络的均方误差为：



1. 根据梯度下降策略学习率为：



为隐层神经元输出

 为

因此 ***(1式)***

**其中**



因为sigmoid的导数为：

 ***(2式)***

1. 反向传播计算：

同理可知：输入层到隐层的权值梯度项为

 ***(3式)***

**其中**需要逆向学习算法：



因为为隐层的输出sigmoid的导数为：

 ***(4式)***

1. **通过上述的公式就可以写出训练算法的伪代码**

输入样本训练集Mnist= 学习率

过程：

随机初始网络中所有连接权的值

repeat

For all do:

求出当前神经网络输出。

通过***(2式)***求出输出层神经元梯度项

通过***(4式)***求出输入层神经元梯度项

通过***(1式)***和***(3式)***更新权值

End for

Until 均方误差趋近收敛或超出最大循环数

1. **参数准备：**

因为Mnist数据集是28\*28 =784的像素图，所以输入神经元应该设为784个，输出是0~9的数字所以设置10个输出神经元。对于隐层神经单元数需要以后调优确定。

权值初始化，对于所有的权值不能设置为0否则训练无法开始，对于并且对于每个权值初始值不能过大或过小，应该保证最后每个神经元的权值和在（-1，1），因此每个权值的初始值应该在范围内随机初始化:



1. **具体实现**
2. **实验环境：**
3. 编程语言：Python
4. 软件环境 ：Jupyter Notebook
5. 硬件环境 ： PC
6. **实验准备：**
7. 准备Mnist数据集，共有60000张训练训练集，和10000张测试数据集



1. 准备Numpy库：方便进行矩阵的运算操作。
2. **代码实现：**
3. **数据加载：**

将二进制文件的Mnist数据集读进来，并选择几个样例进行展示：

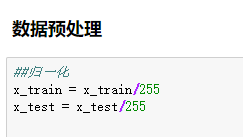




1. **数据预处理：**
2. 因为手写数字的图片x每个像素的取值范围在0~255 ，所以首先要对数据x进行归一化处理。此处有两种方案：

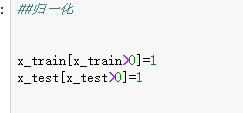
**第一种采用最值归一化：**

对于每个像素的取值除以255，得到每个像素的取值在0~1之间：



**第二种采用二值化：**

因为手写数据集不需要确定颜色，所以将像素>0点设置为1，无像素的点设置为0即可。



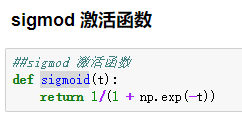
1. 其次要对y进行独热处理，：

比如5 化成 [0,0,0,0,0,1,0,0,0,0]

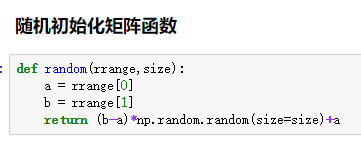
因为每一个每个神经元的输出都是0~1，并且y的取值是0~9，所以要创建10个输出神经元:

1. **相关函数的编写：**

Sigmoid激活函数编写



生成取值范围随机矩阵函数：



根据rrange范围，和size大小，随机初始化一个取值在rrange之间，大小为size的矩阵。用于初始化权值数组。

1. **变量初始化**

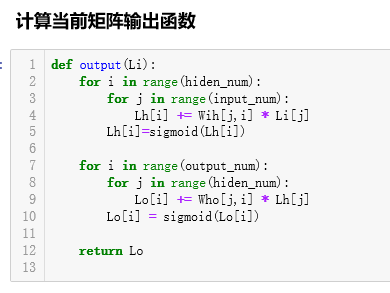
初始化输入神经元个数784，输出神经元个数10，隐身神经元个数以25个为例，在（-1/28，1/28）范围内随机初始化输入层到隐层的权值；在（-1/5，1/5）范围随机初始化隐层到输出层的权值,对于隐层，输出层神经元的阈值初始化。以及记录输入层输出，隐层输出，和输出层输出的变量。



1. **计算当前矩阵输出函数：**

**根据神经网络的输入，计算当前神经网络的输出的函数**

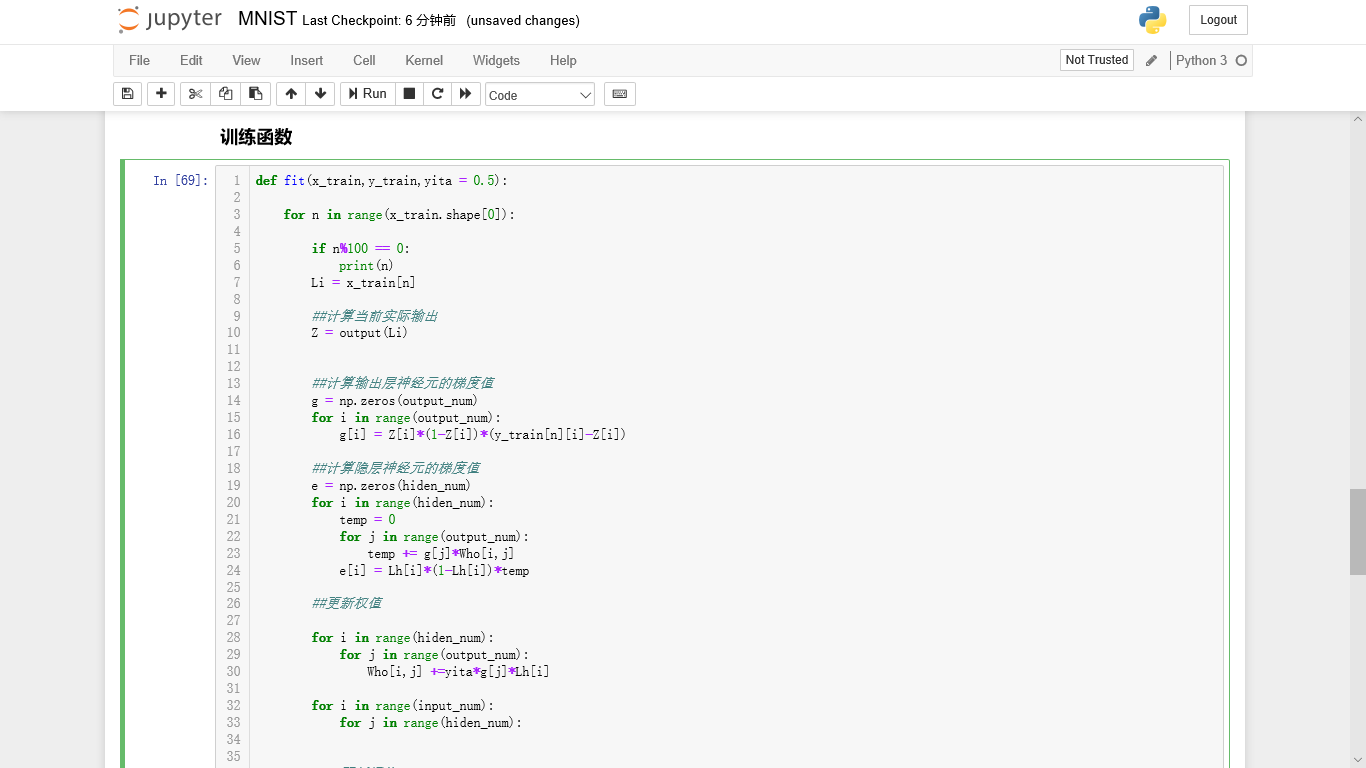




1. **训练函数：**

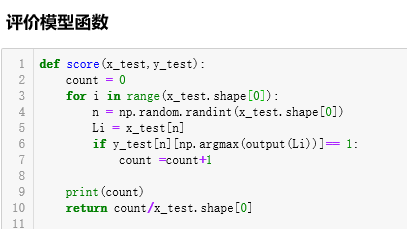
**按照之前推出的公式伪代码实现BP神经网络的训练算法：**

对于每个样本作为输入，计算当前神经网络输出，然后计算输出层神经网络梯度值，计算输入层网络梯度值，根据梯度值和学习率更新响应的权值大小。



1. **评价模型函数：**

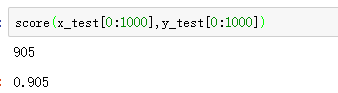
训练出的模型需要输入测试数据集进行评价准确度，根据输入的测试数据集计算神经网络的输出，输出的十个神经元中最大的作为当前神经网络预测的值，与测试数据集的标签进行对比，输出模型在测试数据集上的准确率。

.

1. **训练模型并对模型进行评价：**

先测试一下代码编写是否正确，选择隐层神经元数25，学习率为0.5，不设置阈值，迭代一次全部训练样本（60000）。查看用1000个测试用例测试分类效果：





可以看出进行一次迭代后，正确率达到了90.5%，这个正确率还是非常令人满意的。下面会进行完整参数优化，以训练出最优模型

1. **参数调优：**

编写完代码就可以进行模型的调优过程了，这个过程是漫长且无聊的，因为训练样本集的样本数量有60000个，即使隐层只设置25个神经元，经过训练算法进行一次迭代(每次迭代训练所有的样本数据集)的时间也需要20分钟。如果设置更多的隐层神经元则训练时间是指数级的增长。所以在训练时会减少每次迭代的训练样本集的数量（2000个），尽可能减少训练时间。

根据本次代码的实际情况和本人能力，选择需要进行调优的参数有：

1. 隐层神经元数量
2. 是否设置阈值
3. 迭代次数

|  |  |  |
| --- | --- | --- |
| 隐层神经元数  是否设置阈值 | 否 | 是 |
| 25 | 0.784\0.83\0.848 |  |
| 49 | 0.802\0.839 |  |
| 100 |  |  |
| 400 |  |  |

1. **实验总结：**

实验的代码，会以ipynb文件的形式上传可以随时进行运行和查看。

实验完成了上机实验的全部题目，并且测试通过，通过实验无参数估计的两个重要方法，Parzen窗方法和K-近邻算法，并且通过实验加强了课上所学知识的理解，并应用于实践，对于一些机器学习算法题目的实现重要的是对于公式的理解和掌握，通过推导出的公式进行代码编写实现的过程就相对简单很多了。

本次实验中问题二设计的分类器较为简单，不过分类结果还算良好。问题三中对于N值取得越大概率密度曲线越陡峭；N值越小，概率密度曲线越平滑。

1. **实验代码：**
2. **参考文献**

*[美]RichardO.Duda PeterE.Hart DavidG.Stork 著 模式分类 第二版*