机器学习 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号： | 姓名： | | 班级： |
| 实验题目： | | | |
| 实验学时：4 | | 实验日期： | |
| 实验目的：  使用minist数据集，设计BP神经网络实现手写识别的算法 | | | |
| 硬件环境：  Intel Core（TM） i5-4210U  RAM 8G  64位操作系统，基于x64处理器 | | | |
| 软件环境：  Win 10 64bit  MATLAB | | | |
| 实验步骤与内容：  课本上的内容云里雾里，因此本实验关于BP神经网络的所有内容包括原理与权值计算方式部分均来源于互联网、课堂以及自己的推导计算   1. 根据博客《神经网络浅讲：从神经元到深度学习》可知，在MP模型里，函数g是sgn函数，也就是取符号函数。这个函数当输入大于0时，输出1，否则输出0。而在BP神经网络中则使用了SIGMOD准则函数，定义域为实数，值域（0,1）   20161129162102837.gif  同时，其导数也方便计算，这也有利于我们之后计算更新后的w值  20161129162402837.gif   1. 基本的BP神经网络包括输入层在内共有三层，理论证明，两层神经网络可以无限逼近任意连续函数，这是因为两层神经网络中，隐藏层对原始的数据进行了一个空间变换（维度的变换），使其可以被线性分类。在实验中，我使用三个1\*n的矩阵存储每次循环的输入层、中间层以及输出层，用两个n\*m的矩阵记录权值w，同时使用两个1\*n的矩阵记录两层中b的取值。隐含层、输出层的值由输入层得到，基本原理为   sigmod (W \* a) = z;  sigmod (W \* a + b) = z;   1. 之后则是根据最后的计算结果与实际结果进行比对，然后使用反向传播算法更新w与b的值，结合互联网相关资料进行推导得到并理解公式，经过比对与博客给出的公式一致   fromweb.png  在实际编程中，若按照此方式使用循环逐个计算，则时间非常慢（亲身体会），因此在代码中使用了矩阵乘法以及矩阵的点乘，去除了循环，大大提高了效率。   1. 对于minist数据集，需要进行规范化处理，对改格式的读取方式在官方网站上有所提及，在我的代码中则直接使用了网上的读取方式。   %allTrainImages = loadMNISTImages('train-images.idx3-ubyte');  %allTrainImages = allTrainImages';  %trainLabels = loadMNISTLabels('train-labels.idx1-ubyte');   1. 编程中由于需要测试不同节点对准确率的影响，定义了一个middle变量来控制隐含层节点数，使用高斯分布产生随机的w与b   studyRate1 = 0.3;  studyRate2 = 0.4;  middle = 11;  w1 = normrnd(0,0.5,28\*28,middle);  w2 = normrnd(0,0.5,middle,10);  更新权值以及隐含层、输出层的值的计算使用了矩阵乘法，提高计算速度  outMiddle = input\*w1+b1;  Etotal = sum((y-outFinal).\*(y-outFinal),2)/2;  %反向BP更新权值  w2 = w2 - studyRate2\*(outMiddle'\*s\_outFinal);   1. 由于中间层节点数超过100时，计算所耗费的资源非常大，因此根据群里同学推荐的博客，将节点值设在10左右，学习率设在0.3左右进行尝试，得出部分计算结果。同时，由于对部分数据进行训练时耗费了过长时间，猜测是一些相对异常的例子，在训练循环的过程中设置了跳出的条件   if(whileLope > 1000)  break;  end  以下是部分结果：  rate1:0.4;rate2:0.4;nodes:10  right:80.68%  rate1:0.3;rate2:0.3;nodes:10  right:77.16%  rate1:0.3;rate2:0.3;nodes:9  right:85.14%  rate1:0.3;rate2:0.2;nodes:9  right:84.99%  rate1:0.3;rate2:0.4;nodes:11  right:84.2%  rate1:0.3;rate2:0.4;nodes:20  right:90.26%  rate1:0.3;rate2:0.4;nodes:30  right:90.9%  由于w与b的取值完全随机化，因此结果可能会有一定的浮动，但并不影响其整体的正确率走向。两个学习率的取值与正确率没有显然的联系，当节点数明显增加时，正确率会有提高，但过高时不仅速度变慢，而且正确率没有与之相对应的明显提升。取值为100时曾得到过结果，正确率约为75%，反而大幅度下降。 | | | |
| 结论分析与体会：  这次实验相较之前的一次实验难度又有所提升，但由于这次提前做好了准备，因此在完成时间上并没有十分紧张。本次实验使用了BP神经网络，由于课本给出的起点过高，因此只能选择在网上进行学习以及公式的推导。除此之外还遇到了一些意想不到的bug，比如索引值为1~10而实际上是0~9，当然这对正确率似乎并没有影响。这次的实验，让我对神经网络有了一些浅显的理解，而对深度学习更是门外汉，还需要继续研究和学习，加深对这些技术的理解。 | | | |

**参考链接**

P250 sigmoid函数

神经网络浅讲：从神经元到深度学习

https://www.cnblogs.com/subconscious/p/5058741.html

EasyPR字符识别网络（纯粹是出于兴趣）

<https://www.cnblogs.com/freedomker/p/6767185.html>

反向传播算法（BP）

https://blog.csdn.net/a819825294/article/details/53393837

https://blog.csdn.net/UESTC\_C2\_403/article/details/74908410

https://www.cnblogs.com/charlotte77/p/5629865.html

读取minist数据集

https://blog.csdn.net/tracer9/article/details/51253604

https://blog.csdn.net/u010165147/article/details/50417294