智能信息处理

大作业实验报告

姓名：朱文豪

学号：5130309717

一 结构和使用方法说明

Character Recognition 文件夹是原始版本，采用自制的训练集，读 32*64 的 BMP 图片进行学习，训练集样本少，每个数字只有四个样本。但是自己画图比较方便，训练很快，方便查看代码结果等。
Character Recognition Rewrite with MNIST DATABASE 文件夹是我在网上找了一个 MNIST 手写数字数据库重新改写了一下，训练样本数量非常大，输入是 28*28 的 BMP 图片，这个主要用来测试，分析 BP 网络的参数对于训练效果的影响，而且由于数据库中是各种各样不同的手写方式，字符识别训练好的网络泛化能力比较好。我选取了每个字符 800 个不同的训练样本，然后每个字符提供了另外 200 个不同的测试输入，具体可以在程序宏定义中自行修改。两个结构基本一致，下面主要介绍这个文件夹。
数据库连接 http://yann.lecun.com/exdb/mnist/

程序使用 VISUAL STUDIO 2012 编写，编译。引用的都是自带的库。
除了工程文件夹外，DATA 文件夹里是训练数据，其中，Train 文件夹一共有 8000 张 28*28 的图片，每个数字 800 个样本，用来训练。Test Default 里有 2000 张图片，用来测试训练好的网络。Test USER 是用来存放自定义的图片（自己手写的），在运行程序的时候会提示输入自定义图片个数，要求输入自定义识别图片的个数 N 时请严格对应。将所需要识别的图片名字命名为 0 1 2 ..... N-1。

Q&A:
1. 如何观察网络的训练过程，训练后的识别效果?
打开程序工程，在宏定义中可以调节默认参数
#define SAMPLE_NUM 100 每个字符的样本数量，不超过 800
#define TEST_NUM 10 每个字符的默认测试数量，不超过 200
#define LENGTH 28
#define INPUT_DIMENSION 28*28
#define OUTPUT_DIMENSION 10
#define HIDDEN_LAYER_NEURON_NUM 10 隐层节点个数
#define TRAINNING_SPEED 0.005 学习速率
#define MAX_EPOCH 300 最大训练代数（退出条件）
#define MIN_ERROR 0.0005 最小训练误差（退出条件）
(大数据时，需要读取大量图片并处理，请耐心等待)

随后执行，可以选择是否读取已经训练好的网络参数，选择否就可以观察到训练，如果需要训练的具体信息，可以到 Neuron Network 类头文件中反注释其中的输出语句可以观察具体的权值变化。训练完毕后，本次训练的参数会自动保存到根目录下的 Trained_Parameters.txt 中。之后程序自动执行默认测试，默认测试是与训练样本完全不同的图片输入，可以观察是否正确识别，以及计算正确识别率。最后，程序会询问是否自定义测试，输入个数，会对自定义图片进行识别。

2.如何自己提供图片？
如果需要自定义图片，可以从训练文件或者测试文件中拷贝到 Test USER 文件夹，使用

windows 自带画板或其他工具修改，或者用手写板写字等等，请使用黑底白字（原始版本则是白底黑字），并且因为这次对图像预处理的时候没有加上图像的缩放和剪裁，所以请尽量按照已有的样本的画，不要太夸张。
#注：对于白底黑字我的初始做法是，每个识别的时候把它当作白底黑底个算一次，挑出概率更高的一个，可以解决问题，但是考虑到可能彩底彩字等情况，实际是需要图像预处理考虑这部分，为了节约时间这里暂时略去~假期里再尝试。

3.已经训练好的网络参数在哪里？怎么使用？
在根目录下，My_Trained_Parameters.txt，把文件名改为 Trained_Parameters.txt，在程序中读取即可

二 部分原理说明

BP 网络：
这部分上课的时候都讲过了，根据误差反向调整神经元权重，大体实现是将神经元，神经元层，神经网络各自封装成类，提供各类方法等。具体细节，代码中基本也都写了相关注释，这边不再赘述。

图像处理：
利用 CIMAGE 类处理，简单二值化，RGB 相加除 3，判断是否超过一半，超过算白色，不超过算黑色。但是实际处理的时候，因为训练集样本大部分黑底白字，而且字符相对覆盖的区域较小，也就是说黑色算 1 的话会导致很多图片【矩阵】有大量部分都是 1，尽管可能权重很小。我实际测试发现把字符当 1 算，不管什么颜色，没字符的按 0 算，效果更好。

三 实验结果和分析

以下均每个字符默认只测试 10 个数据，观察看，随着测试数据多，Hit 率会上升。

1） 最大训练代数的影响（最小误差是类似的）
学习速率 0.005
每个数字 100 个样本，经过 30 代训练 hit 率 0.28
每个数字 100 个样本，经过 100 代训练 hit 率 0.61；
每个数字 100 个样本，经过 200 代训练 hit 率 0.64
每个数字 100 个样本，经过 300 代训练 Hit 率 0.76
每个数字 100 个样本，经过 500 代训练 hit 率 0.72；
每个数字 100 个样本，经过 1000 代训练 hit 率 0.73；

总体看，代数越多，识别率也越高。但是到一定程度后，识别率又会略微降低。。主要原因可能是最后来上课的女老师所说的，训练过多导致过度拟合了，导致网络的泛化能力较差，对于一些字符的新的写法识别能力较差。因此，训练代数，或者说训练误差不一定是越小越

好。

2） 学习速率的影响

每次用 300 代训练
100 个样本 训练速度 0.05 Hit 率 0.74
100 个样本 训练速度 0.005 Hit 率 0.76
……
识别率基本差异不大，没有找到明显规律，主要影响是在于，当训练后期一个高的学习速率会导致在极值附近不断震荡。如果控制最小误差，最大训练代数调高，就可以发现，一个小的学习速率总体训练时间比较长，大速率则时间比较短，但是相对的，明显最小误差小的话大速率会一直没法收敛，最终因为到达训练代数退出循环，影响训练的效果。
但是速率太小又不行，因为使用的是梯度下降算法，一个小的学习速率可能会收敛到局部的最小值，而非全局最小值。
为了解决这个问题，可以将学习速率随训练代数减小，从而达到前期大震荡找到全局最小区域，后期小震荡趋于收敛的目的。不过我测试了一下，在 100 个样本的情况下没有很明显的差异，可能与样本有关，

3)训练样本的影响

每个数字 50 个样本，经过 100 代训练 hit 率 0.40；
每个数字 100 个样本，经过 100 代训练 hit 率 0.61；
每个数字 300 个样本，经过 100 代训练 hit 率 0.72；

随着训练样本的提高，识别率明显有很大提升（需要一个好的训练样本数据库），除了训练时间明显增加外（线性增长？），基本没有缺陷。

4）隐层神经元个数的影响

通过将个数分别选择 5 10 30
发现隐层节点越多，训练时间越长。
隐层节点个数的增长与训练代数的增长效果相似。提高隐层节点个数不一定会提高识别率，反而有可能造成过度拟合。

5）**其他**

已经训练好的网络参数是基于 0.005 学习速率，200 代训练，800 个训练样本，隐层节点 10 个，0.0005 最小误差下训练得到。 2000 组测试数据，识别率为 0.85

PS:如果碰到不能执行等问题或者我在报告中没说清楚的话请联系我吧~谢谢!!
TEL 13524957036
EMAIL 575877982@qq.com