**MATLAB课程期末大作业**



选 题： BP网络识别手写数字

姓 名：许 伟

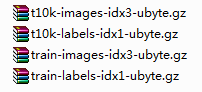
学 号：201820504012

班 级：测试流体控制（7）班

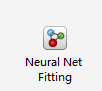
教 师：杨 磊

1. 介绍

本次作业想运用BP神经网络实现手写数字识别，训练的图片集使用的是MINST数字字符, 可以看到前四个是测试数据，后四个是训练数据。



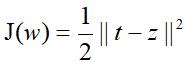
通过MATLAB的Neurl Net Fitting工具箱来进行训练



1. 理论分析

1. 分析BP神经网络的基本原理：通过输入向量与权向量加权求和，得到一个下层神经元的输入，再加上偏置，经过激活函数计算，作下层神经元的输出。

2. 该文采用经典的三层神经网络为例，反向传播的过程如下：

先求出,

其中J(w)为训练误差，t为输出端的期望输出值，z为实际输出值。

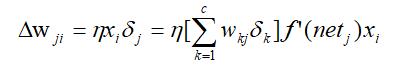
令w表示网络里所有的权值，根据J(w)调整w的大小：

https://img-blog.csdn.net/20180531195609723，

继续的推导需要基于链式法则，具体过程可参考<模式分类>P237 6.3.1.最终得到隐含层到输出曾的权值更新规则：

https://img-blog.csdn.net/20180531195743885

输入层到隐含层权值的权值更新规则是：



具体符号含义都包含在<模式分类>中。

3. 具体训练过程如下：

①首先产生随机的权值w1,w2，生成偏置，初始化所用变量，设置神经网络层数及单元.

②进行前向传递算法，获得实际输出值z，选取激活函数为https://img-blog.csdn.net/20180531195942892

③进行反向传播算法（BP），对权值w1,w2偏置值进行调整。

④迭代‘②③’过程以提高训练的准确率。

4. 测试并收集结果.

将测试集运用前向传递算法算出预测值，与实际期望预测值进行对比.

由于神经网络训练过程中随机生成权值，每次训练的结果具有随机性，因此重复文收集结果.

5. 修改神经网络神经元数量和层数，对结果进行对比分析.

修改隐藏层神经元为20个，其他保持不变，测得平均准确率为：91.70%

1. MATLAB读取以及训练代码部分

通过BP神经网路对MNIST手写数字字符库进行训练和识别，理论上当三层神经网络迭代十次，隐含层包含10个神经元就可以达到89.07%的准确率，当隐含层包含20个神经元时，可以达到91.70%的识别准确率(对于10000条测试用例)。

以下为读取数据库文件的代码1：

clear all

close all

train\_x\_file=char('train-images.idx3-ubyte');%得到vector形式

test\_x\_file=char('t10k-images.idx3-ubyte');%得到vector形式

train\_y\_file=char('train-labels.idx1-ubyte');%得到vector形式

test\_y\_file=char('t10k-labels.idx1-ubyte');%得到vector形式

train\_x=decodefile(train\_x\_file,'image');

test\_x=decodefile(test\_x\_file,'image');

train\_y=decodefile(train\_y\_file,'label');

test\_y=decodefile(test\_y\_file,'label');

save('mnist\_uint8.mat','train\_x','train\_y','test\_x','test\_y');%以上代码已经实现功能要求。

% 如果想检验转化是否正确，可执行以下代码。

train\_x\_matrix=reshape(train\_x,28,28,60000);%reshape后的图像是放倒的

train\_x\_matrix=permute(train\_x\_matrix,[2 1 3]);%对每张图像进行行列的转置处理

test\_x\_matrix=reshape(test\_x,28,28,10000);%reshape后的图像是放倒的

test\_x\_matrix=permute(test\_x\_matrix,[2 1 3]);%对每张图像进行行列的转置处理

figure;

for m=1:5

subplot(2,5,m),imshow(train\_x\_matrix(:,:,m));

title(num2str(train\_y(m)));

end

for m=1:5

subplot(2,5,m+5),imshow(test\_x\_matrix(:,:,m));

title(num2str(test\_y(m)));

end

代码二：

function output=decodefile(filename,type)

% TRAINING SET LABEL FILE (train-labels-idx1-ubyte):

%

% [offset] [type] [value] [description]

% 0000 32 bit integer 0x00000801(2049) magic number (MSB first)

% 0004 32 bit integer 60000 number of items

% 0008 unsigned byte ?? label

% 0009 unsigned byte ?? label

% ........

% xxxx unsigned byte ?? label

% The labels values are 0 to 9.

% TRAINING SET IMAGE FILE (train-images-idx3-ubyte):

%

% [offset] [type] [value] [description]

% 0000 32 bit integer 0x00000803(2051) magic number

% 0004 32 bit integer 60000 number of images

% 0008 32 bit integer 28 number of rows

% 0012 32 bit integer 28 number of columns

% 0016 unsigned byte ?? pixel

% 0017 unsigned byte ?? pixel

% ........

% xxxx unsigned byte ?? pixel

% TEST SET LABEL FILE (t10k-labels-idx1-ubyte):

%

% [offset] [type] [value] [description]

% 0000 32 bit integer 0x00000801(2049) magic number (MSB first)

% 0004 32 bit integer 10000 number of items

% 0008 unsigned byte ?? label

% 0009 unsigned byte ?? label

% ........

% xxxx unsigned byte ?? label

% The labels values are 0 to 9.

%

% TEST SET IMAGE FILE (t10k-images-idx3-ubyte):

%

% [offset] [type] [value] [description]

% 0000 32 bit integer 0x00000803(2051) magic number

% 0004 32 bit integer 10000 number of images

% 0008 32 bit integer 28 number of rows

% 0012 32 bit integer 28 number of columns

% 0016 unsigned byte ?? pixel

% 0017 unsigned byte ?? pixel

% ........

% xxxx unsigned byte ?? pixel

fio=fopen(filename,'r');%原始文件中数据是以2进制存储的。

a = fread(fio,'uint8');%以8进制方式读取源文件。虽然前几项是32bit的，但是图像像素数据是8bit的，所以此处用8bit处理。

if strcmp(type,'image')

% magic\_num=a(1)\*256^3+a(2)\*256^2+a(3)\*256+a(4);

% image\_num=a(5)\*256^3+a(6)\*256^2+a(7)\*256+a(8);

% image\_rows=a(9)\*256^3+a(10)\*256^2+a(11)\*256+a(12);%默认训练和测试图像都已经reshape到一个size

% image\_cols=a(13)\*256^3+a(14)\*256^2+a(15)\*256+a(16);

output=a(17:end);%提取像素数据

else if strcmp(type,'label')

% magic\_num=a(1)\*256^3+a(2)\*256^2+a(3)\*256+a(4);

% image\_num=a(5)\*256^3+a(6)\*256^2+a(7)\*256+a(8);

output=a(9:end);

end

end

1. 结果分析与思考

当分别将以下两张图片加载到工程中去读取，可以得出训练模型的返回值接近4以及2。

C:\Users\Administrator\Desktop\4.bmp C:\Users\Administrator\Desktop\2.png

但是由于某些问题，导致返回值和实际值相差有些大。以识别图片数字为例，图片中黑色像素点和白色像素点之间的空间编排关系构成了我们看到的数字，可以找到一个方程组，方程的输入参数是一张图片，返回一个数字表示识别结果，神经网络刚好就完成了这样的映射功能，所有的关于像素点空间排布信息的解析都隐藏于矩阵W和V之中，所以W和V实际上代表了这样一种映射关系。神经网络对映射关系未知的样本设计一套机理让它自己来寻找，这就是学习和训练的过程，而最后得到的W和V就是学习的结果，自然界的各种各样的映射关系显然是比较复杂的，关靠线性关系是无力描述的，而神经网络中用的是加权相加的运算，如果没有经过激活函数处理，后层的所有输入都可以由多层之前的信号经过线性运算来得到，一个现象模型显然表现力是远远不够的，所以我们在神经网络模型中加入了非线性的因素，就是激活函数。通过这次大作业，我初步认识了深度学习领域的一些知识，对此非常感兴趣，但由于时间仓促，调试结果不太理想，接下来希望能够更加深入地学习相关方面的知识，以及python的学习，希望可以在深度学习图像识别的某个领域有所收获。感谢杨老师一学期的授课，我学到了很多。