基于 BP 神经网络的鼠标手势识别实验程序

北京理工大学 01110407 班　林健　学号：20040200

问题陈述

本程序基于 BP 神经网络完成了对用户使用鼠标输入的特定手势的识别，并且可以学习用户创建的新手势。程序中已经存储的手势是 Palm 系列掌上电脑中手写识别专用的 Graffiti 字体的数字 0～9，这种字体的特点是每个字由一个连续笔划构成，适合用一个连贯的鼠标手势表示：

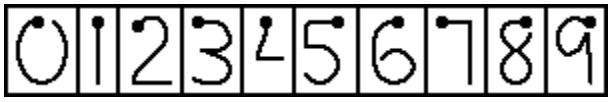

图 1 Graffiti 字体的数字

程序概述

本程序针对鼠标手势的存储、识别和学习功能，需要实现以下要素：

一、鼠标手势的表示

由于每个鼠标手势仅由一个连续笔划构成，因此可将这一笔划分解为一系列连续的向量。不同的手势对应不同的向量组合。为计算方便，所有向量可以归一化为单位长度向量，如图 2 示例：

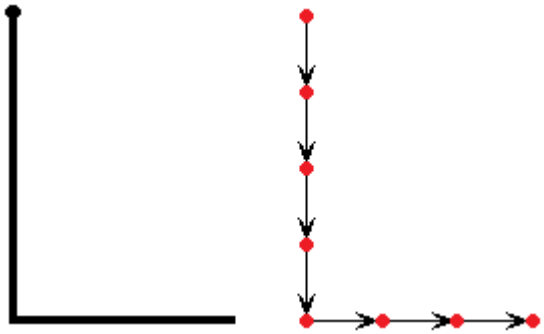

图 2 鼠标手势向量示例

分别以向右、向下为 x 轴、y 轴正方向，则图 2 所示笔画的向量组合为 (0,1)-(0,1)-(0,1)-(0,1)-(1,0)-(1,0)-(1,0)。

二、定义网络输入输出

使用 BP 神经网络，需要定义一组一维的输入向量与输出向量。将一个手势的笔划向量依次连接，x 坐标、y 坐标依次出现，即可构成该手势的输入向量。对于可以识别 N 种手势的神经网络，可以使用长度为 N，第 M 位为 1、其余各位均为 0 的向量作为第 M 种手势的标

准输出向量。

上例对应的输入向量为(0,1,0,1,0,1,0,1,1,0,1,0,1,0)，输出向量形式为(0,…,0,1,0,…,0)。

三、训练神经网络

对于程序中已存储的手势，需要训练神经网络，使得网络对每个输入向量都能够产生误差在允许范围之内的输出向量。BP 神经网络训练使用反向传播过程，即对每个输入向量重复以下步骤：

（1）将它送入网络，计算网络的输出o。

（2）计算输出o与目标输出t之间的误差。

（3）调整输出层权重，为每个隐藏层重复步骤（4）和（5）。

（4）计算隐藏层误差。

（5）调整隐藏层权重。

四、鼠标手势的记录

用户使用鼠标，在程序界面指定区域按下左键画出一个手势，程序定时记录鼠标坐标，构成一个点集（图 3 所示蓝点）。使用一定的算法对点集进行处理，取出比笔画的向量数目多一个的代表点（图 3 所示红圈），相邻两个代表点求差得到的向量经归一化即可得到笔画的向量组合。

取代表点的方法采用了光滑化算法，即对分布不均的原始记录点找到其中跨度最小的两个点，用它们的中点取代它们。重复执行，直到点的数目符合向量数目要求。

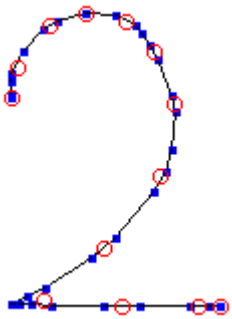

图 3 鼠标手势记录示例

五、鼠标手势的识别

用户的输入不可能与定义的标准输入完全一致。将待测试的手势对应的输入向量输入训练过的神经网络后，网络的输出与定义的第 M 种手势的标准输出向量越接近，说明输入手势越接近第 M 种手势的标准输入。评判“接近”的简单标准是找出输出向量中最大（最接近 1）的元素，认为输入最有可能是该位为 1 的输出对应的手势。

六、鼠标手势的学习

学习过程是将用户新输入的一个手势构建输入向量，加入原有的输入向量集；对输出向量的长度加一，设置其对应的输出向量，然后将网络重新训练一遍。

程序架构

本程序使用 Visual Studio 2005（C#语言）开发，基于.Net Framework 2.0 运行。

一、关键数据结构

程序包含的类如下：

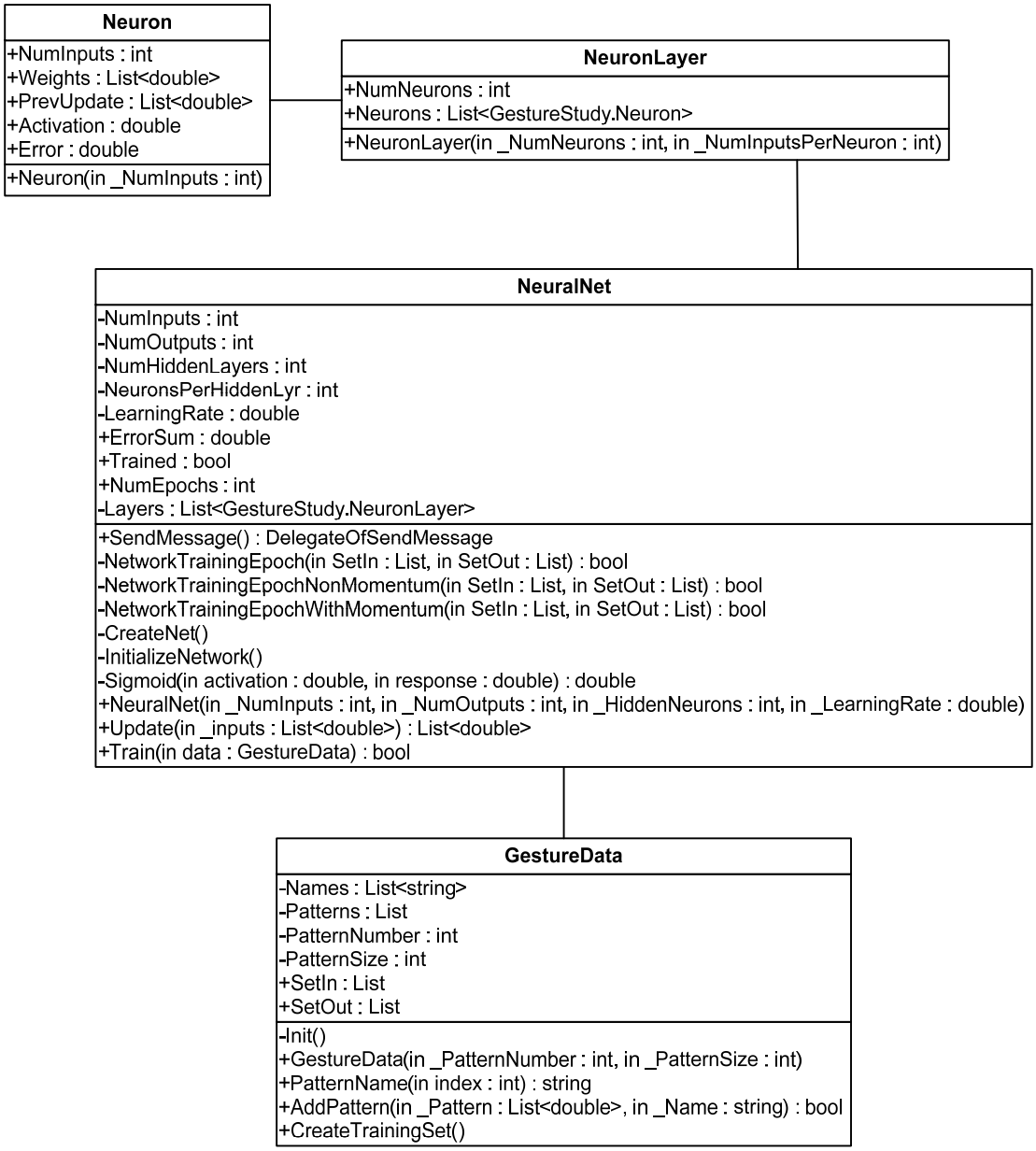

图 4 程序类图

（1）Neuron 类（神经元）：记录一个神经元的输入数目、各个输入的权重、偏差值等数据。

（2）NeuronLayer 类（神经网络层）：记录一个神经网络层的神经元数目和各个神经元对象。

（3）NeuralNet 类（神经网络）：记录一个 BP 神经网络的各个神经网络层以及隐藏层数、学习率、训练与否等信息；提供训练、更新等方法。

（4）GestureData 类（鼠标手势）：记录所有鼠标手势的名称、笔划向量组合、对应输出等信息；提供追加新鼠标手势等方法。

二、程序界面操作说明

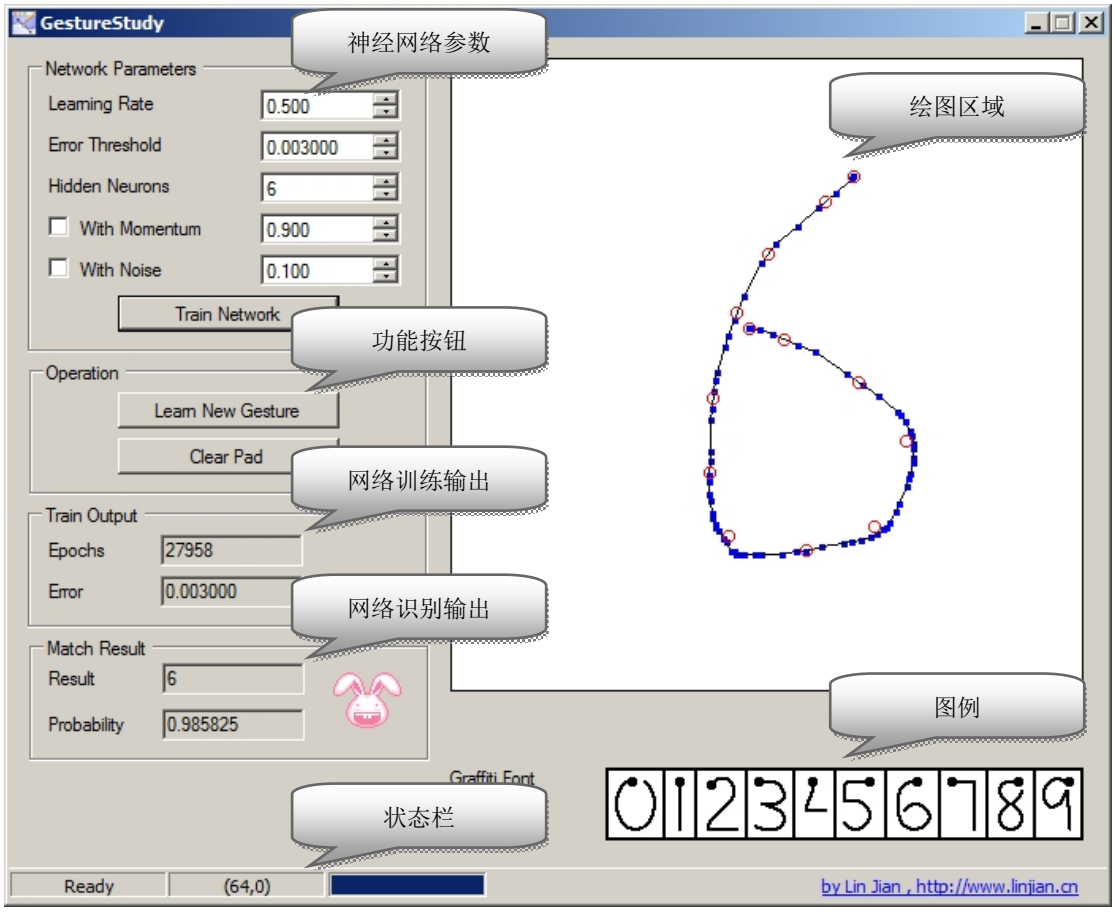

图 5 程序界面

（1）程序运行后，首先需要训练神经网络。用户可以设置网络的一些参数，包括学习率、可允许的偏差门限、隐藏层数等。此外可以选择开启带动量的算法或带噪声的算法，并设置动量分数和噪声最大值。单击“Train Network”按钮开始训练网络，界面左下方显示当前训练的迭代数和偏差。

（2）训练之后，在绘图区域按下鼠标左键画出一个连续的手势，程序会将代表点标注在手势上，并在界面左下方给出通过神经网络运算得到的最佳匹配输出值与对应手势的名称，用卡通图标显示最佳匹配输出值的大小。

（3）单击“Learn New Gesture”按钮，程序进入学习状态。此时用户可以输入一个新的手势，并在弹出的对话框中为之命名。程序随即重新训练神经网络。

（4）单击“Clear Pad”按钮清空绘图区域。

核心算法

BP 神经网络学习算法属于误差修正型学习，其学习过程由信息正向传播和误差反向传播两个阶段构成。当给定一组输入模式时，BP 网络将依次对这组输入模式中的每个输入模式按如下方式进行学习：把输入模式从输入层传到隐含层单元，经隐含层单元逐层处理后，产生一个输出模式传至输出层，这一过程称为正向传播；如果经正向传播在输出层没有得到所期望的输出模式，则转为误差反向传播过程，即把误差信号沿着原连接路径返回，并通过修改各层神经元的连接权值，使误差信号为最小；重复正向传播和反向传播过程，直至得到所期望的输出模式为止。

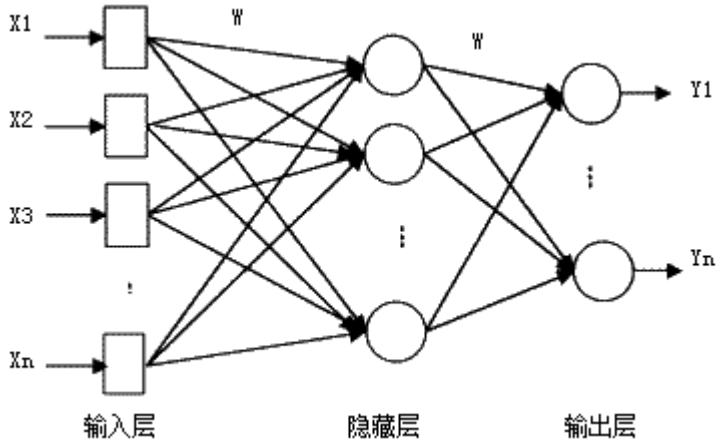

图 6 BP 神经网络结构

一、训练神经网络

1、调整输出层的权重

记第 k 个输出神经元的输出为 o_k，期望的目标输出为 t_k，则第 k 个输出神经元的初始误差值为：

$$E_k = (t_k - o_k) o_k (1 - o_k)$$

为了改变隐藏层第 j 个神经元和输出层第 k 个输出神经元之间的权重，使用下面的公式（L 为学习率）：

$$W_{jk} += L E_k o_j$$

2、调整隐藏层的权重

计算隐藏层第 j 个神经元的权重调整公式如下（n 为输出层神经元的个数）：

$$E_j = o_k (1 - o_k) \sum_{k=1}^{n} E_k W_{jk}$$

从隐藏层第 j 个神经元到输入 i 的权重使用下面的公式计算：

$$W_{ij} += LE_j o_i$$

重复以上各步骤，直到总误差减小到指定的门限以下，这样训练出来的神经网络参数达到了可接受的水平。

二、网络优化策略

1、使用带动量的算法避免陷入局部最优解

反向传播算法根据误差分布来确定当前点的误差梯度，并朝向全局最小的方向移动。但通常的误差分布并不光滑，算法容易收敛到一个局部最优解。为避免这一情况的发生，可以将前一步的权重的更新量加入到当前的权重中，这样可以抵消算法中任何小的波动，从而为寻找全局最优解提供有利条件。加入动量后的权重计算公式为：

$$W_{ij} += LE_j o_i + m\Delta W_{ij}$$

ΔW_{ij} 为前一步的权重的更新量，m 为被加入的分数。

2、使用带噪声的算法避免过拟合

神经网络的问题之一是训练得与原训练集过于耦合，从而失去了归纳推广能力。为避免这一情况，可以使用带噪声的算法。噪声是平均值为 0 的小的随机数，将其加入训练数据，可以阻止网络训练时过分拟合指定的数据点。

实验数据

在不学习新手势的前提下，将神经网络的其它参数固定为默认值，对某一参数设定不同值，以便测试该参数对网络性能的影响。这里设定 2 个简单的评判标准：网络训练速度（迭代次数）与识别率。其中对识别率的测试方法是使用 3 组易混淆的数字（2-7、0-6、5-9）各测试 10 次，求出正确的百分比。

一、网络参数的影响

1、误差门限对训练速度的影响

误差门限	第 1 次	第 2 次	第 3 次	第 4 次	第 5 次	平均值
E=0.0003	209665	224152	226261	163520	227724	210264
E=0.003	18225	18982	18685	21254	17351	18899
E=1	200	174	183	207	155	184

从实验结果可以明显看出，误差门限越大，收敛速度越快。

2、误差门限对识别率的影响

误差门限	测试“2”	测试“7”	测试“0”	测试“6”	测试“5”	测试“9”
E=0.0003	30%	80%	90%	80%	60%	50%
E=0.003	60%	100%	80%	90%	90%	60%
E=1	80%	100%	90%	90%	100%	50%

从实验结果可以明显看出，误差门限在 0.003 时具有较好的识别效果，更高的门限对最终输出结果影响不大，但最佳匹配输出值的大小距 1 更远了，输出向量的区分度降低。更小的门限会造成过拟合现象，使识别率有所下降。

3、隐藏层数对训练速度的影响

当隐藏层数为 1、2 时，网络收敛速度极慢，50 万次迭代之后仍距离误差门限有 1～3 个数量级的差距。

当隐藏层数为 3～12 时，网络收敛速度可以接受。在一定范围内层数越多的网络收敛速度越快。

4、隐藏层数对识别率的影响

对于本程序，隐藏层数在 3～12 层之间变化，对识别率没有明显的影响。但据文献[1]使用大量数据测试，发现在其它网络参数为默认值的情况下 6 层隐藏层的性能最佳。

5、学习率对训练速度的影响

学习率	第 1 次	第 2 次	第 3 次	第 4 次	第 5 次	平均值
L=0.1	97913	101167	121343	95163	93873	101892
L=0.5	18225	18982	18685	21254	17351	18899
L=0.9	9276	11469	12780	11552	11457	11307

神经网络学习率影响网络的收敛速度及稳定性。当网络的学习率为 0.1 时，网络震荡小，但收敛速度慢。当网络学习率增大时，网络收敛快，但震荡大。

6、学习率对识别率的影响

学习率	测试“2”	测试“7”	测试“0”	测试“6”	测试“5”	测试“9”
L=0.1	80%	100%	90%	90%	100%	80%
L=0.5	60%	100%	80%	90%	90%	60%
L=0.9	60%	90%	70%	70%	80%	50%

实验基本证实了较低的学习率可以保证网络收敛到全局最优解，而学习率过高时网络有可能收敛到局部最优解，进而影响识别的性能。综合考虑收敛速度和识别率，可以选择网络学习率为 0.5。

二、网络优化策略的影响

1、带动量的算法对训练速度的影响

网络参数使用默认值，测试带动量的算法对训练速度的影响，结果如下：

算法	第 1 次	第 2 次	第 3 次	第 4 次	第 5 次	平均值
不使用动量	18225	18982	18685	21254	17351	18899
$m=0.3$	16833	13201	15081	16995	14281	15278
$m=0.9$	13990	12357	10763	11823	12042	12195

可见适当的动量可以显著加快网络训练收敛速度。

2、带动量的算法对识别率的影响

网络参数使用默认值，测试带动量的算法对识别率的影响，结果如下：

算法	测试“2”	测试“7”	测试“0”	测试“6”	测试“5”	测试“9”
不使用噪声	60%	100%	80%	90%	90%	60%
$m=0.3$	90%	100%	100%	90%	100%	70%
$m=0.9$	90%	100%	90%	100%	100%	60%

可见适当的动量对提高部分数字识别率有一定作用，但不及其加快网络训练收敛速度那样明显的效果。

3、带噪声的算法对训练速度的影响

网络参数使用默认值，测试带噪声的算法对训练速度的影响，结果如下：

算法	第 1 次	第 2 次	第 3 次	第 4 次	第 5 次	平均值
不使用噪声	18225	18982	18685	21254	17351	18899
最大值 0.2	20083	19086	20750	21409	17774	19820
最大值 0.6	26025	28470	33167	41028	29025	31543

可见噪声的加入会降低网络训练收敛速度。

4、带噪声的算法对识别率的影响

网络参数使用默认值，测试带噪声的算法对识别率的影响，结果如下：

算法	测试“2”	测试“7”	测试“0”	测试“6”	测试“5”	测试“9”
不使用噪声	60%	100%	80%	90%	90%	60%
最大值 0.2	100%	100%	100%	80%	90%	70%
最大值 0.6	50%	100%	100%	50%	100%	20%

实验发现，噪声最大值在 0.1～0.2 附近时，程序对鼠标手势的识别率有所上升，特别是对 2 和 7 的区分正确率提高了，但总体改善幅度不大。更大的噪声不但降低了训练速度，对识别率也有负面的影响。例如将 2 误判为 7、将 6 误判为 0、将 9 误判为 1、7 或 5 的机率大为上升。

参考文献

[1] Mat Buckland. 游戏编程中的人工智能，北京：清华大学出版社，2006.
[2] 朱大奇，史慧. 人工神经网络原理及应用，北京：科学出版社，2006.