广西科学院学报 2006,22(1):60～63
Journal of Guangxi Academy of Sciences Vol. 22, No. 1 February 2006

基于 JAVA 的鼠标书写的非常用字符识别*

Recognition of Non-common Characters by Mousewriting Based on JAVA

谢晓兰[1],韩可轶[1],冯嘉礼[2]
XIE Xiao-lan[1], HAN Ke-yi[1], FENG Jia-li[2]

(1. 桂林工学院电子与计算机系,广西桂林 541004;2. 上海海事大学信息工程学院,上海 200135)
(1. Department of Electronics and Computer Science, Guilin University of Technology, Guilin, Guangxi, 541004, China; 2. Infomation Engineering College, Shanghai Maritime University, Shanghai, 200035, China)

摘要:采用 JAVA 语言,通过合理设置网络的变量,改进误差逆传播校正方法(BP 算法),缩短了 BP 算法的学习时间,提高算法的运算速度,实现了鼠标书写的非常用字符识别. 试验程序验证表明,识别程序能快速准确识别用鼠标书写的字符.

关键词:字符识别 非常用字符 鼠标 神经网络 BP 算法 JAVA
中图法分类号:TP183 **文献标识码**:A **文章编号**:1002-7378(2006)01-0060-04

Abstract: Java language and BP algorithm are used to develop a method to identify the writing characters by using a mouse. The BP learning time is getting shorter by improvement of BP algorithm and proper setting of network parameters. In the examination, by using the deducing of theory and the demonstrating of programming, the error and the rate of identification are acceptable for usual inputting.

Key words: character recognition, non-common characters, mouse, neural network, BP algorithm, JAVA language

Microsoft Word 的用户在输入 $\sum$、$\because$、β、μ 等不常用的字符时,需要通过一系列繁琐的菜单选择来查找目标字符,由于字库里符号较多,常常需要多次翻页查找,操作十分不便,降低了输入的效率. 针对这一非常用字符的输入问题,人们通常采用手写输入产品解决这一问题. 由于手写输入产品价格较高且占用有限空间,阻碍了手写输入产品的普及. 基于上述原因,我们提出在计算机屏幕上直接用鼠标书写字符并进行识别的方法,它以神经网络的误差逆传播校正方法(Error Back Propagation Training, BP)作为核心算法,通过缩短 BP 算法学习时间的途径,提高算法的运算速度,实现了鼠标书写字符的识别.

1 BP 算法中网络变量设置

鼠标书写的非常用字符识别主要采用 BP 算法来实现,BP 算法主要利用网络的实际输出值与期望输出值之差,对网络的各层连接权由后向前逐层校正[1～4]. 从理论上讲,BP 算法可以适用于任意多层的网络,为简单起见,本文以三层网络模型来实现非常用字符的识别(图 1).

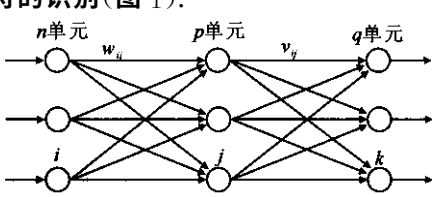

图 1 三层 BP 网络模型

三层网络模型的输入层有 n 个神经单元,输出

收稿日期:2005-05-08
作者简介:谢晓兰(1974-),女,博士研究生,主要从事计算机、制造业信息化、机械制造及其自动化研究。
* 广西教育厅科研项目(桂教科研[2003]22 号)资助。

层有 q 个神经单元,中间层有 p 个神经单元. 由于中间层与输入、输出层没有直接的联系,所以也常把中间层称为隐含层. 为计算方便,首先把网络的变量设置如下:

输入模式向量 $A_k=[a_1^k,a_2^k,\cdots,a_n^k]$;

输出层个单元输入向量 $L_k=[l_1^k,l_2^k,\cdots,l_q^k]$;

中间层各单元输入向量 $S_k=[s_1^k,s_2^k,\cdots,s_p^k]$,输出向量 $B_k=[b_1^k,b_2^k,\cdots,b_p^k]$;

输出层实际输出向量 $C_k=[c_1^k,c_2^k,\cdots,{}_q^k]$,期望输出向量 $Y_k=[y_1^k,y_2^k,\cdots,y_q^k]$;

输入层至中间层的连接权 W_{ij},中间层至输出层的连接权 V_{jk};

中间层个单元阈值 θ_j,输出层个单元阈值 γ_t;

以上各向量的 $i=1,2,\cdots,n;j=1,2,\cdots,p;t=1,2,\cdots,q;k=1,2,\cdots,m$.

由于 S 型函数是连续可微分的,而且更接近于生物神经元的信号输出形式,所以激励函数采用 S 型函数,即

$$f(x)=\frac{1}{1+\mathrm{e}^{-x}}, \tag{1}$$

激励函数的导数为

$$f'(x)=f(x)[1-f(x)], \tag{2}$$

网络的期望输出与实际输出之差设为

$$\delta_j^k=(y_j^k-c_j^k),j=1,2,\cdots,q, \tag{3}$$

采用平方和误差进行计算

$$E_k=\sum_{t=1}^{q}(y_t^k-c_t^k)^2/2=\sum_{t=1}^{q}(\delta_t^k)^2/2, \tag{4}$$

按梯度下降的原则,应使 V_{jt} 的调整量 ΔV_{jt} 与 $\frac{\partial E_k}{\partial V_{jt}}$ 的负值成正比例变化,所以中间层至输出层连接权的调整量应为

$$\Delta V_{jt}=-\alpha\cdot\frac{\partial E_k}{\partial V_{jt}}. \tag{5}$$

由于(5)式中的偏导与已知的变量没有直接的关系,所以将(5)式展开为

$$\Delta V_{jt}=-\alpha\cdot\frac{\partial E_k}{\partial c_t^k}\cdot\frac{\partial c_t^k}{\partial V_{jt}}, \tag{6}$$

利用(4)式可得

$$\frac{\partial E_k}{\partial c_t^k}=-(y_t^k-c_t^k)=-\delta_t^k, \tag{7}$$

因为输出层第 t 个单元的激活值为

$$l_t=\sum_{j=1}^{q}V_{jt}\cdot b_j-\gamma_t,t=1,2,\cdots,q, \tag{8}$$

输出层第 t 个单元的输出值为

$$c_t=f(l_t). \tag{9}$$

由(2)式可得对于输出函数的导数

$$f'(l_t)=c_t(1-c_t), \tag{10}$$

所以 $\frac{\partial c_t^k}{\partial V_{jt}}=\frac{\partial c_t^k}{\partial l_t^k}\cdot\frac{\partial l_t^k}{\partial V_{jt}}=f'(l_t)\cdot b_j^k=c_t^k(1-c_t^k)\cdot b_j^k$. (11)

因此,由(6),(7)和(11)式可得

$$\Delta V_{jt}=-\alpha\cdot\delta_t^k\cdot c_t^k(1-c_t^k)\cdot b_j^k. \tag{12}$$

为进一步简化,设 d_t^k 为 E_k 对输出层输入 l_t^k 的负偏导,则

$$d_t^k=-\frac{\partial E_k}{\partial l_t^k}=-\frac{\partial E_k}{\partial c_t^k}\cdot\frac{\partial c_t^k}{\partial l_t^k}=\delta_t^k\cdot c_t^k(1-c_t^k), \tag{13}$$

所以

$$\Delta V_{jt}=\alpha\cdot d_t^k\cdot b_j^k,0<\alpha<1;t=1,2,\cdots,q;j=1,2,\cdots,p;k=1,2,\cdots,m. \tag{14}$$

同理,由输入层至中间层连接权的调整仍按此方法进行. 中间层个单元的激活值为

$$S_j=\sum_{i=1}^{n}W_{ij}\cdot\alpha_i-\theta_j,j=1,2,\cdots,p. \tag{15}$$

中间层输出为

$$b_j=f(s_j). \tag{16}$$

由(13),(8),(16)和(15)式有

$$\frac{\partial E_k}{\partial W_{ij}}=(\sum_{t=1}^{q}\frac{\partial E_k}{\partial l^k}\cdot\frac{\partial l^k}{\partial b_j^k})\cdot\frac{\partial b_j^k}{\partial s_j^k}\frac{\partial s_i^k}{\partial W_{ij}}=[\sum_{t=1}^{q}(-d_t^k)\cdot V_{jt}]\cdot f'(s_j^k)\cdot a_i^k=-[\sum_{t=1}^{q}d_t^k\cdot V_{jt}]b_j^k\cdot(1-b_j^k)\cdot a_i^k,$$

$$i=1,2,\cdots,n;j=1,2,\cdots,p. \tag{17}$$

设 e_j^k 为中间层个单元的校正误差,则

$$e_j^k=-\frac{\partial E_k}{\partial s_j^k}=-(\sum_{t=1}^{q}\frac{\partial E_k}{\partial l_t^k}\cdot\frac{\partial l_t^k}{\partial b_j^k})\cdot\frac{\partial b_j^k}{\partial s_j^k}=[\sum_{t=1}^{q}d_t^k\cdot V_{jt}]\cdot b_j^t(1-b_j^t), \tag{18}$$

(17)式可表示为

$$\frac{\partial E_k}{\partial W_{ij}}=-e_j^k\cdot a_i^k, \tag{19}$$

与 ΔV_{jt} 相类似,连接权 W_{ij} 的调节量为

$$\Delta W_{ij}=-\beta\frac{\partial E_k}{\partial W_{ij}}=\beta\cdot e_j^k\cdot a_i^k,0<\beta<1;i=1,2,\cdots,n;j=1,2,\cdots,p. \tag{20}$$

同样也可求出阈值的调整量,取输出层的输入 $b_k^k=1$ 带入 ΔV_{jt},则输出层阈值的调整量为

$$\Delta\gamma_t=\alpha\cdot d_t^k,t=1,2,\cdots,q.$$

同理,中间层阈值的调整量为

$$\Delta\theta_j=\beta\cdot e_j^k,j=1,2,\cdots,p.$$

2 BP 学习算法的改进

为了加快 BP 网络的学习速度和增加网络的识别精度,本文对 BP 算法作了如下改进:

(a)调整量与学习系数成正比. 通常学习系数在0.1～0.8,为使整个学习过程加快且不引起震荡,我们采用变学习率方法,即在学习初期取较大的学习系数,随着学习过程的进行逐渐减少学习系数的值.

(b)利用前次的校正结果来影响本次校正量,加快校正过程,即惯性校正法:

$\Delta W(N) = \Delta W(N) + \rho\Delta W(N-1)$,其中$\rho$为惯性系数,$0 < \rho < 1$.

当前一次的校正过量时,惯性项与本次误差校正项符号相反,使本次实际校正量减少,起到减小震荡的作用. 而当前次校正不够时,惯性项与本次误差校正项符号相同,本次实际校正量增加,起到加速校正的作用(图2).

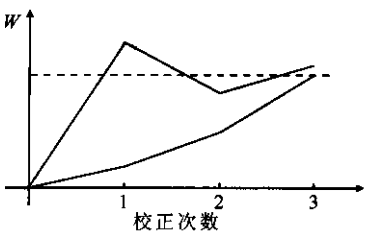

图2 惯性校正法的作用

(c)限制S函数的输出. 由于连接权的校正量为:

$$\Delta V_{jt} = \alpha \cdot d_t^k \cdot b_j^k,$$

$$\Delta W_{ij} = \beta \cdot [\sum_{t=1}^{q} d_t^k \cdot V_{jt}] \cdot b_j \cdot (1 - b_j) \cdot a_i^k,$$

当b_j为0或1时,ΔV_{jt}或ΔW_{ij}为0,不起校正作用. 为了保证每次学习都能进行有效地校正,从而加快收敛过程,可以限制S函数的输出,即当$b_j < 0.01$时,取$b_j = 0.01$;当$b_j > 0.99$时,取$b_j = 0.99$.

(d)给连接权加上随机数. 由于BP网的收敛有可能收敛于局部极小点,我们给每个连接权加上一个很小的随机数,使收敛过程避开局部极小点.

(e)增加隐含层中的神经元数目. 增加层数可以进一步降低误差,提高精度. 但降低误差的同时也使网络复杂化,从而增加了网络权值的学习时间. 实际上我们可以通过增加隐含层中的神经元数目来提高精度,其学习效果也更容易观察和调整.

学习结束后,网络将一组学习样本的模式分布记忆在各层之间的连接权和各层的阈值中. 当再次给网络提供已经记忆的输入模式时,网络将计算出期望的输出值,并根据输出值判断出这一输入模式属于或接近记忆中的何种模式.

3 基于JAVA的鼠标书写的非常用字符识别

考虑到JAVE语言是纯粹的面向对象语言,我们选用JAVA语言来实现BP算法. JAVA语言使代码容易理解和维护,另外JAVA的跨平台特性也使得程序可以很轻松地移植到不同的操作系统上运行[5].

为了验证本论文的实际效果,并直观地说明BP算法对字符的识别能力,我们依据算法设计了一个手写输入识别的试验演示程序,实现3个字符的识别,其分别是英文大写字母"B"以及数学中的符号"∵"和"$\sum$". 由于样本很少,程序中没有设计样本输入模块,直接设置了组成字符点阵的矩阵值.

3.1 程序代码结构

演示程序总体上分4个部分:程序的主类Application1.class,界面及操作类Frame1.class,鼠标输入绘图类DrawPanel.class和算法实现类Calculate.class.

手写输入部分由DrawPanel.class完成,调用Graphics类的绘图方法,在界面上绘制由32 * 32个黑色小方块组成的正方形输入区域. 算法部分主要在Calculate.class中实现,该类中对算法划分成初始化、产生随机值、隐含层计算、输出层计算、误差计算、误差校正等几个主要方法分别编写后,在Frame1.class调用这些方法.

3.2 功能块主要代码

绘图方法中,矩阵inputArray中保存每个小方块的状态值,分为无笔划的0和有笔划的1;参数length为单元小方块的边长. 调用Graphics的setColor方法对32 * 32的输入单元中每一个小方块根据其状态值设置颜色,再调用fill3Drect方法在界面上绘出输入区域,具体如下:

```
g.setColor(color[(int)inputArray[i][j]]);
g.fill3DRect(dx + j * length,dy + i * length, length, length, true).
```

产生随机值方法是调用数学类Math类的random方法,产生0～1的随机浮点数,逐个对输入层到隐含层连接权W_{ij}、隐含层到输出层连接权V_{jk}、隐含层阈值和输出层阈值赋值:

```
wij[i][j]=(float)(2 * Math.random( )-1);//在-1到+1之间取值
```

隐含层和输出层则根据推导的公式对隐含层个单元的输入输出值和输出层个单元的输入输出值进

行计算,其中激励函数S型函数的$e-x$部分调用数学类Math类的exp( )方法,使用方法如下:

```
outputN[i]=outputN[i]-thresholdout[i];
outputO[i]=1/(1+(float)Math.exp(-outputN[i]));
```

依据算法完成每个功能块和界面的设计,并对整体的程序进行调试,完成最后的试验程序.

3.3 非常用字符的识别

实验演示程序的手写输入区域是一个由32 * 32个小方块组成的点阵输入模块.打开程序后,可以选择菜单"产生参数文件",程序可依据内部算法自动生成各层连接权值和阈值,并将计算得的参数保存在参数文件中,以后再次打开程序,即可以直接从已经产生的文件中读取参数.

用鼠标进行手写输入时,在黑色的输入区拖动鼠标写出所需要字符的笔划.书写完成后,按"识别输入"按钮即可获得识别结果.在算法设定的误差范围内,可以识别出所写的字符.

手写输入的字符即便与预先设定的样本相差达50%,仍可以很好的识别.本演示程序可以通过菜单选择来查看预先设置的样本.图3为样本与相应手写输入字符的比较.试验程序验证表明,3个字符均可以快速准确地被识别出来.

从图3上可以看出,尽管鼠标书写非常用字符时出现不确定的偏移和形变,与设置的样本字符有一定的误差,但仍能实现鼠标书写非常用字符的识别,并有较高的准确率.

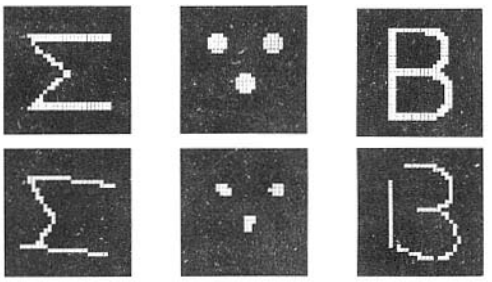

图3 样本与手写输入的比较

4 结束语

本文在改进BP算法的基础上,采用JAVA语言,实现了鼠标书写的非常用字符识别.如果继续完善程序,还可设计出样本输入模块,以便通过学习样本熟悉使用者的书写习惯,使得网络对字符学习更为准确.更进一步,如果使用者根据需要增加学习样本,经过长时间使用和适应,并逐渐生成数量可观的样本库后,用鼠标也可实现文字的录入工作.

参考文献:

[1] 陆汝钤.人工智能[M].北京:科学出版社,2000.

[2] 易继锴,侯媛彬.智能控制技术[M].北京:北京工业大学出版社,1999.

[3] 丛爽.神经网络、模糊系统及其在运动控制中的应用[M].合肥:中国科学技术大学出版社,2001.

[4] 王旭,王宏,王文辉.人工神经网络原理与应用[M].沈阳:东北大学出版社,2000.

[5] HERBERT SCHILDT. JAVA 2:The complete reference[M].北京:电子工业出版社,2003.

(责任编辑:黎贞崇)

(上接第59页)

保险.

3 结束语

由于风险的不确定性,为了保持经营的稳建性及偿付能力,保险公司有各种方法规避风险,进行再保险是一种常用的方法.评估再保险的优劣有许多方法,我们利用VaR原理,得到了保持原保险人在财务稳健基础上的一种考虑再保险的策略.虽然我们只考虑2种再保险的策略,但对其他类型再保险也可类似考虑.

参考文献:

[1] 谢志刚,韩天雄.风险理论与非寿险精算[M].天津:南开大学出版社,2000.

[2] MULLER A. Ordering of risks,a comparison study Via stop loss transform [J]. Insurance: Mathematics & Economics,1996(17):215-222.

[3] DAVID C,DICKSON M. Relative reinsurance retention levels[J]. Astin,1997,27(2):207-227.

[4] DEPRENT O,GERBER H. On convex principles of premium calculation [J]. Insurance: Mathematics & Economics,1985(4):179-189.

[5] HOJGAARD B,TSKSAR M. Optimal proportional reinsurance policies for diffusion models [J]. Scandinavian Actuarial Journal,1997(2):166-180.

[6] LESAW GAJEK,DARIUSZ ZAGRODNY. Optimal reinsurance under general risk measures[J]. Insurance: Mathematics and Economics,2004,34:227-240.

[7] JORION P. Value at risk:benchmark for controlling market risk[M]. Second edition. McGraw-Hill,2000.

(责任编辑:黎贞崇)