

# 基于 BP 网络的字母识别

谢丽娟, 丁正生, 陈俏  
(西安科技大学 理学院, 陕西 西安 710054)

**摘要:** 介绍了用 BP 神经网络方法对英文字母进行识别, 并在识别过程中考虑了噪声干扰及非线性因素的存在, 使网络具有一定的容错能力, 并用 MATLAB 完成了对字母识别的模拟。

**关键词:** BP 神经网络; 模式识别, MATLAB

中图分类号: N3

文献标识码: A

## The alphabet identification based on BP network

XIE Li Juan, DING Zheng Sheng, CHEN Qiao  
(College of Science, Xi'an University of Science and Technology, Xi'an 710054, China)

**Abstract:** The passage introduces method of using artificial neural network method to recognize alphabet, and considers the existence of noise disturbance and nonlinear factor, so the network has certain fault-tolerance capability. Finally use MATLAB to simulate alphabet recognition.

**Key words:** back propagation network; mode recognition; MATLAB

智能控制作为一门新兴的交叉学科, 在许多方面都优于传统控制, 而智能控制中的人工神经网络由于模仿人类的神经网络, 具有感知识别、学习、联想、记忆、推理等智能, 更是有着广阔的发展前景。其中最核心的是反向传播网络(Back Propagation Network), 简称 BP 网络<sup>[1]</sup>。本文介绍了运用 matlab 工具箱确定隐层神经元的个数和构造 BP 神经网络, 并用两类不同的数据对该神经网络进行训练, 然后运用训练后的网络对字符进行识别。

### 1 BP 网络

#### 1.1 BP 网络的简介

20 世纪 80 年代中期, 学者 Rumelhart、McClelland 和他们的同事提出了多层前馈网络 MFNN(Multilayer Feedforward Neural Networks)的反向传播学习算法, 简称 BP 网络(Back Propagation Network)学习算法。BP 网络是对非线性可微分函数进行权值训练的多层前向网络。在人工神经网络的实际应用中, 80%~90% 的模型都采用 BP 网络或其变化形式。

BP 网络主要作用于以下几个方面:

(1) 函数逼近: 用输入矢量和相应的输出矢量训练一个网络来逼近一个函数;

(2) 模式识别: 用一个特定的输出矢量将它与输入

矢量联系起来;

(3) 分类: 把输入矢量以所定义的合适的方式进行分类;

(4) 数据压缩: 减少输出矢量的维数以便于数据传输或存储。

#### 1.2 BP 网络模型

BP 网络是一种单向传播的多层前向网络<sup>[2]</sup>, 每一层节点的输出只影响下一层节点的输出, 其网络结构如图 1 所示, 其中  $X$  和  $U$  分别为网络输入、输出向量, 每个节点表示一个神经元。网络是由输入层、隐层和输出层节点构成, 隐层节点可为一层或多层, 同层节点没有

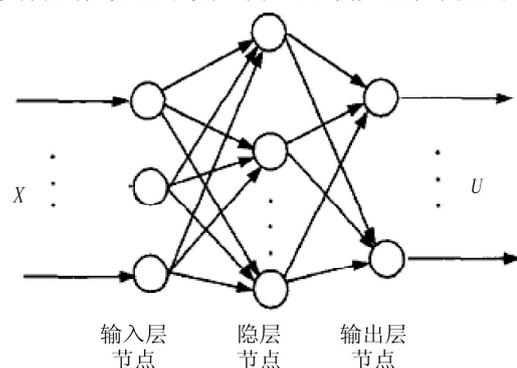


图 1 BP 神经网络

## 应用奇葩 Example of Application

任何耦合, 前层节点到后层节点通过权连接。输入信号从输入层节点依次传过各隐层节点到达输出层节点。

### 2 字符识别问题的描述及网络识别前的预处理

字符识别是模式识别领域的一项传统课题, 这是因为字符识别不是一个孤立的问题, 而是模式识别领域中大多数课题都会遇到的基本问题, 并且在不同的课题中, 由于具体的条件不同, 解决的方法也不尽相同, 因而字符识别的研究仍具有理论和实践意义。这里讨论的是用 BP 神经网络对 26 个英文字母的识别。

在对字母进行识别之前, 首先必须将字母进行预处理<sup>[3]</sup>, 即将待识别的 26 个字母中的每一个字母都通过的方格形式进行数字化处理, 其有数据的位置设为 1, 其他位置设为 0。如图 2 给出了字母 A、B 和 C 的数字化过程, 然后用一个  $1 \times 35$  的向量表示。例如图 2 中字母 A 的数字化处理结果所得对应的向量为:

LetterA=[00100010100101010001111111000110001]

0	0	1	0	0
0	1	0	1	0
0	1	0	1	0
1	0	0	0	1
1	1	1	1	1
1	0	0	0	1
1	0	0	0	1
1	0	0	0	1
1	1	1	1	0

A

1	1	1	1	0
1	0	0	0	1
1	0	0	0	1
1	1	1	1	0
1	0	0	0	1
1	0	0	0	1
1	0	0	0	1
1	1	1	1	0

B

0	1	1	1	0
1	0	0	0	1
1	0	0	0	0
1	0	0	0	0
1	0	0	0	0
1	0	0	0	1
1	0	0	0	1
0	1	1	1	0

C

图 2 数字化字母

由此可得每个字母由 35 个元素组成一个向量。由 26 个标准字母组成的输入向量被定义为一个输入向量矩阵 *alphabet*, 即神经网络的样本输入为一个  $35 \times 26$  的矩阵。其中  $alphabet=[letterA, letterB, letterC, \dots, letterZ]$ 。网络样本输出需要一个对 26 个输入字母进行区分输出向量, 对于任意一个输入字母, 网络输出在字母对应的顺序位置上的值为 1, 其余为 0, 即网络输出矩阵为对角线上为 1 的  $26 \times 26$  的单位阵, 定义为  $target=eye(26)$ 。

本文共有两类这样的数据作为输入: 一类是理想的标准输入信号; 另一类是在标准输入信号中加上用 MATLAB 工具箱里的噪声信号, 即 *randn* 函数。

### 3 网络设计及其试验分析

为了对字母进行识别, 所设计的网络具有 35 个输入节点和 26 个输出节点, 对于隐含层节点的个数的选取在后面有详细的介绍。目标误差为 0.000 1, 从输入层到隐层的激活函数采用了 S 型正切函数 *tansig*, 从隐层到输出层的激活函数采用了 S 型对数函数 *logsig*, 这是因为函数的输出位于区间 [0,1] 中, 正好满足网络输出的要求。

#### 3.1 隐层节点个数的确定

根据 BP 网络的设计目标, 一般的预测问题都可以通过单隐层的 BP 网络实现。难点是隐层节点个数的选择, 隐层节点数对网络的学习和计算特性具有非常重要的影响, 是该网络结构成败的关键。若隐层节点数过少, 则网络难以处理复杂的问题; 但若隐层节点数过

多, 则将使网络学习时间急剧增加, 而且还可能导致网络学习过度, 抗干扰能力下降。

目前为止, 还没有完善的理论来指导隐层节点数的选择, 仅能根据 Kolmogorov 定理, 和单隐层的设计经验公式<sup>[4]</sup>, 并考虑本例的实际情况, 确定隐层节点个数应该介于 8~17 之间。

本文设计了一个隐层节点数目可变的 BP 网络, 通过误差对比, 确定最佳的隐层节点个数, 具体程序如下:

```
[alphabet,targets]=prprrob;
p=alphabet;
t=targets;
s=8:17;
res=zeros(1,10);
res2=zeros(1,10);
for i=1:10
fprintf('s(i)=%.0f\n',s(i));
net=newff(minmax(p),[s(i),26],{'tansig','logsig'},'traingdx');
net.trainParam.epochs=1000;
net.trainParam.goal=0.0001;
[net,tr]=train(net,p,t);
y=sim(net,p);
error=(y(1,:)-t(1,:)).^2;
error2=(y(2,:)-t(2,:)).^2;
res(i)=norm(error);
res2(i)=norm(error2);
pause
i=i+1;
end
```

通过网络的输出显示以及网络训练速度和精度因素, 选取隐层节点的最佳个数为 14。

#### 3.2 生成网络

使用函数 *newff* 创建一个两层网络, 具体函数为:

```
[alphabet,targets]=prprrob;
[R1,Q1]=size(alphabet)
[R2,Q2]=size(targets)
S1=14;
S2=R2;
net=newff(minmax(p),[S1 S2],{'tansig','logsig'},'trainlm')
net.LW{2,1}=net.LW{2,1}*0.01;
net.b{2}=net.b{2}*0.01;
```

#### 3.3 网络训练

为了使产生的网络对输入向量有一定的容错能力, 最好的办法是使用理想的信号和带有噪声的信号对网络进行训练。使用不同信号的训练都是通过 BP 网络来实现的。网络学习的速率和冲量参数设置为自适应改变, 并使用函数 *trainlm* 进行快速训练。

##### 3.3.1 理想样本训练

首先用理想的输入信号对网络进行训练, 直到平

## 应用奇葩 Example of Application

方和误差足够小。下面进行理想样本训练，训练结束条件为：最大次数为 1 000，误差平方和为 0.000 01。训练代码如下：

```
net.performFcn='sse';
net.trainParam.goal=0.00001;
net.trainParam.show=5;
net.trainParam.epochs=1000;
net.trainParam.mc=0.95;
[net,tr]=train(net,p,t);
```

训练过程误差变化情况可通过 MATLAB 进行观察，训练结果为：

TRAINLM, Epoch 77/1000, SSE 6.58108e-006/1e-005,  
Gradient 8.03024e-005/1e-010

TRAINLM, Performance goal met.

可见，经过 77 次训练后，网络误差达到要求，结果如图 3 所示。

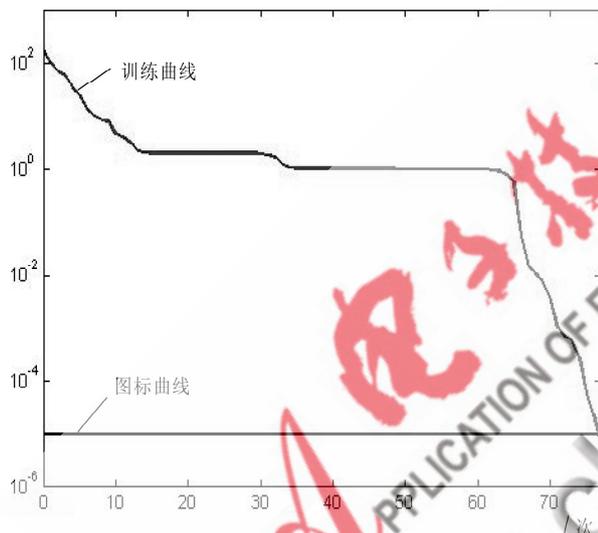


图3 理想样本训练结果图

### 3.3.2 加噪样本训练

为了保证设计的网络对噪声不敏感，有必要用 10 组带有噪声的信号对网络进行训练，设置向字母表加入的噪声信号平均值分别为 0.1 和 0.2。这样就可以保证神经网络学会在辨别带噪声信号的字母表向量时，也能对理想的字母向量有正确的识别。同时在输入带有误差的向量时，要输入两倍重复的无误差信号，其目的是为了保证网络在分辨理想输入向量时的稳定性。

在输入理想样本上加入噪声的信号后，网络的训练过程误差变化情况也可通过 MATLAB 进行观察。选取其中的一组，观察系统输出结果如下：

TRAINLM, Epoch 30/1000, SSE 4.45738e-006/1e-005,  
Gradient 5.97808e-005/1e-010

TRAINLM, Performance goal met.

结果如图 4 所示。

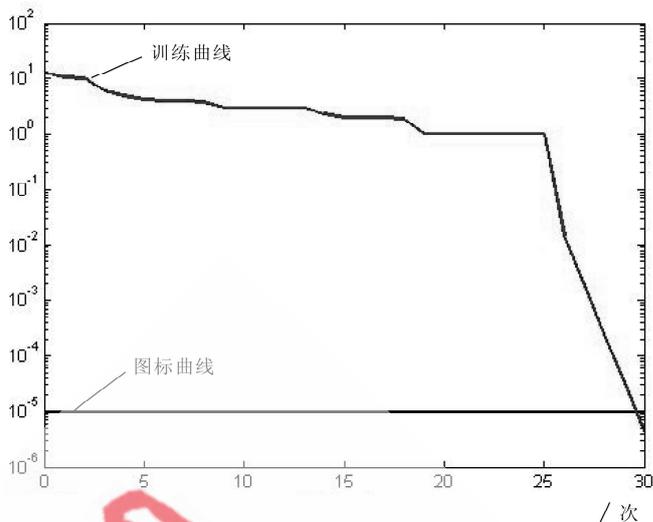


图4 加噪样本训练结果图

### 3.3.3 再次用理想样本训练

在网络进行了上述的训练以后，网络对无误差的信号可能也会采用对带有噪声信号的办法。这样做会付出较大的代价。因此，必须再次使用理想的样本进行训练。这样就可以保证在输入理想数字信号时，网络能够最好地对其做出反应。其训练代码如下：

```
netn.trainParam.goal=0.00001;
netn.trainParam.epochs=1000;
netn.trainParam.show=5;
[netn,tr]=train(netn,p,t);
```

训练结果为：

TRAINLM, Epoch 0/1000, SSE 4.60127e-007/1e-005, Gradient 4.23932e-006/1e-010

TRAINLM, Performance goal met.

满足要求。

### 3.4 对网络进行仿真和测试

为了测试系统的可靠性，本文用了加入不同级别的噪声的字母样本作为输入，来观察用理想样本和加噪样本训练出来的网络的性能，并绘制出误识率曲线，如图 5 所示。

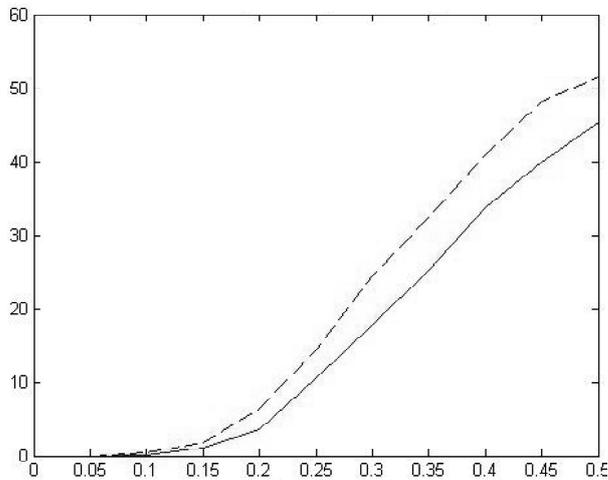


图5 误识率曲线表

（下转第74页）

图5其中虚线代表用无噪声训练网络的出错率，实线代表用有噪声训练网络的出错率。从图5可以看出，在均值为0~0.05之间的噪声环境下，两个网络都能够准确地进行识别。当所加的噪声均值超过0.05时，待识别字符在噪声作用下不再接近于理想字符，无噪声训练网络的出错率急剧上升，此时有噪声训练网络的性能较优。

### 3.5 测试实例

本文用一个含噪声的字母F作为网络输入，并绘出含噪声的字母F，其输出语句为：

```
noisyF=alphabet(:,6)+randn(35,1)*0.2; plotchar(noisyF);
```

其结果如图6所示。

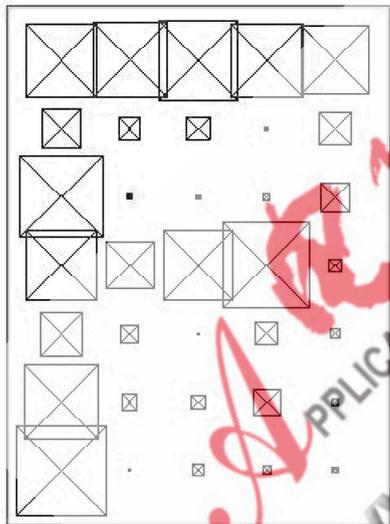


图6 含噪声的字母F显示图

然后再用训练后的网络进行识别，其识别语句为：

```
A2=sim(net,noisyF);
A2=compet(A2);
answer=find(compet(A2)==1)。识别结果如图7所示。
```

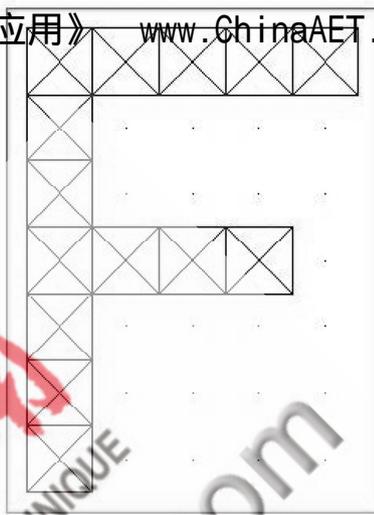


图7 经过训练后的网络后显示的F图

本文利用BP网络对有噪声的字母进行识别和仿真，结果表明此网络具有联想记忆和抗干扰功能，对字母具有一定的辨识能力，是一种对字母识别的有效方法。

### 参考文献

- [1] 侯媛彬, 杜京义, 汪梅. 神经网络[M]. 西安: 西安电子科技大学出版社, 2007.
- [2] 丛爽. 面向MATLAB工具箱的神经网络理论与应用[M]. 北京: 中国科学技术大学出版社, 1998.
- [3] 丛爽, 陆婷婷. 用于英文字母识别的三种人工神经网络的设计[J]. 仪器仪表学报, 2006, 27(6): 2242-2244.
- [4] 孟娜, 周以齐. 基于Matlab的时序数据两种建模和预测方法比较. 山东农业大学学报, 2006, 37(3): 471-476.

(收稿日期: 2009-03-20)