

# 基于 MATLAB 的遗传神经网络的设计与实现

欧阳玉梅<sup>1,2</sup>, 马志强<sup>1</sup>, 方若森<sup>2</sup>

(1. 东北师范大学计算机学院, 长春 130117; 2. 伊犁师范学院, 奎屯 833200)

**摘要:** 介绍遗传神经网络以及运用 MATLAB 工具箱构造遗传神经网络的方法, 并详细阐述网络实现的关键问题, 包括设计网络架构、选取学习规则、进化训练权重与阈值。给出应用示例, 体现出用 MATLAB 语言建立、训练和仿真网络的编程将会非常容易。通过与 BP 神经网络的比较, 说明遗传神经网络的优良性能。

**关键词:** 神经网络; 遗传算法; MATLAB

## Design and realization of genetic-neural network based on MATLAB

OUYANG Yu-mei<sup>1,2</sup>, MA Zhi-qiang<sup>1</sup>, FANG Ruo-sen<sup>2</sup>

(1. College of Computer, Northeast Normal University, Changchun 130117, China;  
2. Yili Normal College, Kuitun 833200, China)

**Abstract:** This paper introduces genetic-neural network, and a method using the MATLAB tool kit to design genetic-neural network is introduced. Moreover the key issues of network's realization, which include designing network architecture, selecting learning regulation, evolutionary training weight and threshold are illustrated in detail. The application example is given out. It can be easily that creating a network, training a network and simulating a network with MATLAB language. The efficiency of genetic-neural network is showed with comparing the BP neural network with genetic-neural network.

**Key words:** neural networks; genetic algorithms; MATLAB

## 0 引言

人工神经网络由于其强大的并行处理、分布式信息存储、自组织自适应及学习能力, 已广泛应用于模式识别、人工智能、控制工程及信号处理等领域。BP 神经网络是众多神经网络中应用最多的一种。遗传算法 (Genetic Algorithms, 简称 GAs) 是以自然选择和遗传理论为基础, 将生物进化过程中适者生存规则与群体内部染色体随机交换机制相结合的高效全局寻优搜索算法。神经网络主要用来学习, 遗传算法主要用来优化。用遗传算法可以指导神经网络权重与阈值的优化和拓扑结构的选择。由此得到的混合智能系统——遗传神经网络能发挥出各自的优势。MATLAB 是 MathWorks 公司研制的一套高性能的数值计算和可视化软件, 得到了各个领域专家学者的广泛关注。用 MATLAB 语言及其优化工具箱

进行优化设计具有语言简单、函数丰富、用法灵活、编程效率高等特点。本文介绍了运用 MATLAB 工具箱构造遗传神经网络, 并通过打印体数字识别系统实例比较了改进的 BP 神经网络与遗传神经网络的性能。

## 1 BP 神经网络设计的关键

### 1.1 选择神经网络架构

BP 神经网络又称反向传播网络。该神经网络连接权的调整采用的是反向传播学习算法。它是一种前馈网络, 采用最小均方差学习方式。

神经网络的架构和大小取决于问题的复杂程度。商用 ANN 一般有三层或四层, 包含一到两个隐

收稿日期: 2007-10-19

作者简介: 欧阳玉梅 (1964-), 女, 伊犁师范学院副教授, 现就读于东北师范大学计算机学院, 硕士研究生, 主要研究方向为计算智能。

含层。每层有 10 ~ 1000 个神经元。但大多数应用系统仅有三层,因为每增加一层,计算量将呈指数倍增加。具有一个输入层、一个隐含层和一个输出层的两层(按 MATLAB 不计算输入层在内)BP 神经网络的简化模型图如图 1 所示。

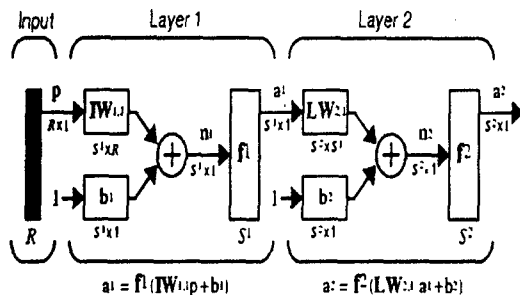


图 1 BP 神经网络结构

由于 MATLAB 是以矩阵为基本运算单元,因此它的每一个输出参数和输入参数均为矩阵。通常把第二层的输出  $a_2$  标记为  $y$ 。输入向量的维数决定了输入层神经元的个数  $R$ 。输出为二值形式时表示分类(输出为连续形式时用于解决预测问题),分类类型数目决定了输出层神经元的个数  $S_2$ (有时用编码表示,如 4 个神经元可以表示 16 个类型)。

仿真实验表明,隐含层神经元的数量能够影响求解问题的精确度和速度。隐含层神经元太少不能识别复杂的模式,但太多又极大地增加了计算负担。另外,为了防止过拟合现象,要选择使网络产生最好的推广结果的最少的隐含层的神经元数量。可以由经验来决定。例如,在实验时可使用 2, 5, 10 和 20 个比较其结果。

隐含层节点的激活函数根据问题特点可以选  $\tanh$  或  $\log\text{sig}$ , 输出层节点的激活函数可以在  $\tanh$ 、 $\log\text{sig}$  和  $\text{purelin}$  中挑选。

## 1.2 选取学习规则

BP 神经网络的优点是只要有足够的隐含层和隐结点,就可以逼近任意的非线性映射关系;学习算法属于全局逼近的方法,因而具有较好的泛化能力;有较好的容错性。缺点为收敛速度慢;容易陷入局部极值,难以确定隐含层数和隐含层节点数。针对这些缺点,人们又提出了 BP 算法的一些改进算法,如加入动量项、学习率自适应、L-M 算法等。

## 1.3 用 MATLAB 神经网络工具箱构造 BP 网络

使用函数  $\text{newff}()$  生成神经网络。函数  $\text{newff}()$  格式为  $\text{net} = \text{newff}(\text{PR}, [\text{S1 S2} \dots \text{SN1}], \{\text{TF1 TF2} \dots \text{TFN1}\}, \text{BTF}, \text{BLF}, \text{PF})$ , 它需要的参数有输入向量的最小值最大值的取值范围、各层神经元个数、各层神

经元传递函数和训练用函数名称等。 $\text{newff}()$  自动完成权值和阈值的初始化。也可用  $\text{init}()$  或  $\text{initf}()$  对网络初始化。

在 MATLAB 神经网络工具箱中 BP 算法及其改进算法相应的训练函数有: 批处理梯度下降法 ( $\text{traingd}$ )、附加动量法 ( $\text{traingdm}$ )、自适应学习速率 ( $\text{traingda}$ ,  $\text{traingdx}$ ), 以及 Levenberg - Marquardt 方法 ( $\text{trainlm}$ ) 等。每一种都有各自的特点。可以先构建网络并设定训练算法, 然后再应用通用的训练函数  $\text{train}()$ 。也可以直接应用相应算法的训练函数进行训练。函数  $\text{train}()$  的格式为  $[\text{net}, \text{tr}, \text{Y}, \text{E}, \text{Pf}, \text{Af}] = \text{train}(\text{net}, \text{P}, \text{T}, \text{Pi}, \text{Ai}, \text{VV}, \text{TV})$ 。

用网络仿真函数  $\text{sim}()$  或  $\text{simuff}()$  可以测试神经网络的性能。 $\text{sim}()$  格式为  $[\text{Y}, \text{Pf}, \text{Af}, \text{E}, \text{perf}] = \text{sim}(\text{net}, \text{P}, \text{Pi}, \text{Ai}, \text{T})$ 。

## 2 遗传神经网络设计的关键

### 2.1 用遗传算法优化网络权重与阈值

遗传算法是一种基于自然选择和基因遗传学原理的全局优化搜索方法。它将“优胜劣汰, 适者生存”的生物进化原理引入待优化参数形成的编码串群体中, 按照一定的适应度函数及自然选择、交叉、变异等一系列遗传操作对各个个体进行筛选, 从而使适应度高的个体被保留下来, 组成新的群体, 新群体中各个个体适应度不断提高, 直至满足一定的极限条件。此时, 群体中最高适应度的个体即为待优化参数的最优解。遗传算法能够在复杂空间进行全局优化搜索, 并且具有较强的鲁棒性。

尽管 BP 算法有了许多改进, 可以用来解决各种问题, 但仍然存在不能保证得到最优解决方案、最佳结构难以选择的严重缺点。可以用遗传算法指导权重与阈值优化、拓扑结构选择以及学习规则进化等。本文只介绍遗传算法优化网络权重与阈值。

#### 2.1.1 编码染色体

要使用遗传算法, 首先要将问题域表示成染色体。网络初始权重是在一个很小的范围(例如  $[-1, 1]$ )内随机选择的。采用实数编码, 可将各权重和阈值级联在一起, 转换成遗传空间中的染色体。讨论一个两层神经网络, 结构如图 1 所示。其中  $p$  和  $y$  分别表示网络的输入和输出,  $R$  和  $S_2$  分别表示输入和输出的维数,  $S_1$  为隐节点数,  $\text{LW1}, 1$  (简记为  $W_1$ ) 为输入层到隐含层的连接权重矩阵,  $\text{LW2}, 1$  (简记为  $W_2$ ) 为隐含层到输出层的连接权重矩阵。编码的前部分为  $W_1$ , 随后是  $W_2$ , 之后是阈值  $b_1$ , 阈值  $b_2$ 。显然, 染色体长度为:  $S = R * S_1 + S_1 * S_2 + S_1 +$

$S_2$ , 即一个染色体由  $S$  个基因构成, 如表 1 所示。

表 1 染色体的权重与阈值编码

$w_{1,1}$	$w_{1,2}$	...	$w_{s1,R}$	$w_{2,1}$	$w_{2,2}$	...	$w_{s2,s1}$	$b_1$	...	$b_{s1}$	$b_2$	...	$b_{s2}$
-----------	-----------	-----	------------	-----------	-----------	-----	-------------	-------	-----	----------	-------	-----	----------

## 2.1.2 定义适应度函数

定义适应度函数来评估染色体性能。该函数必须评估给定神经网络的性能。可以使用由误差平方和的倒数这个相当简单的函数。这样, 适应度越大, 网络的实际输出和期望输出的误差越小。

## 2.1.3 构造有效的遗传操作算子

遗传操作算子有选择、交叉和变异三类。

**选择(复制):**根据各个个体的适应度, 按照一定的规则或方法, 从第  $t$  代群体中选择出一些优良的个体遗传到下一代群体中。可以挑选轮盘赌选择、标准几何选择、锦标赛选择等算法。

**交叉:**将群体内的各个个体随机搭配成对, 对每一对个体, 以某个概率(称为交叉概率)交换它们之间的部分染色体; 可以挑选算术交叉、启发式交叉、简单交叉等算法。

**变异:**对群体中的每一个个体, 以某一概率(称为变异概率)改变某一个或某一些基因座上的基因值为其他基因值。可以挑选以下的算法: 边界变异、多点非均匀变异、非均匀变异、均匀变异等算法。

## 2.2 用 MATLAB 工具箱构造遗传神经网络

构造遗传神经网络可以用 MATLAB 神经网络工具箱与遗传算法工具箱。关于在 MATLAB 中可以使用的遗传算法工具箱主要有三种: (1) GAOT (Genetic Algorithm Optimization Toolbox) 工具箱: 这是美国北卡罗莱纳那州立大学推出的遗传算法优化工具箱。它不是 MATLAB 软件自带的, 可自行配置使用。(2) GATBX (Genetic Algorithm Toolbox) 工具箱: 这是英国设菲尔德 (Sheffield) 大学开发的遗传算法工具箱, 也不是 MATLAB 软件自带的, 可自行配置使用。(3) GADST (Genetic Algorithm and Direct Search Toolbox) 工具箱: 这是 MathWorks 公司最新发布的遗传算法与直接搜索工具箱。MATLAB 7.x 包含了这个工具箱, MATLAB 7.0 以前的版本没有这个工具箱。本文以 GAOT 工具箱为基础介绍。

GAOT 工具箱包括了许多实用的函数。函数功能及参数说明可在工具箱相应  $m$  文件中找到。其核心函数有: 初始种群生成函数, 格式 `Function[pop] = Initializega(num, bounds, evalFN, evalOps, options)`; 遗传算法函数, 格式为: `function[x, endPop, bPop, traceInfo] = ga(bounds, evalFN, evalOps, startPop, opts, termFN, termOps, selectFN, selectOps, xOverFNs,`

`xOverOps, mutFNs, mutOps)`。

在 MATLAB 环境下, 设定相应的参数, 执行 `ga` 就可以完成优化。

构造遗传神经网络主要步骤如下: (1) BP 网络初始化。(2) 确定 BP 网络权值与阈值的范围和编码长度。(3) 初始化种群。(4) 设定相应的参数, 执行遗传算法; 其中包括对群体中的个体的适应度评价, 选择、交叉、变异遗传操作, 进化生成新一代的群体; 反复进行直至进化到设定的代数; 并由此得到进化解(最佳染色体个体)。(5) 将进化解编码分解为 BP 网络所对应的权值、阈值; (6) 将初步得到的权值、阈值赋给 BP 网络后直接仿真; 也可以进一步设置 BP 训练参数, 再进行 BP 训练(此为混合 GA-BP 算法训练), 然后仿真。(7) 为使网络对输入有一定的容错能力, 还可利用不含和含有噪声的输入数据反复训练网络, 再次仿真。

## 3 打印体数字识别系统的实现

### 3.1 BP 神经网络架构设计

设计一个两层 BP 神经网络实现打印体数字识别。为了简化问题, 把数字成像系统数字化了的数字用  $9 \times 5$  的位图表示, 如图 2 所示。因此, 输入层神经元的个数  $R$  为 45。输出层选 10 个神经元 ( $S_2 = 10$ ), 每个神经元用来识别一个数字。数字识别系统输入和预期的输出模式如表 2 所示。

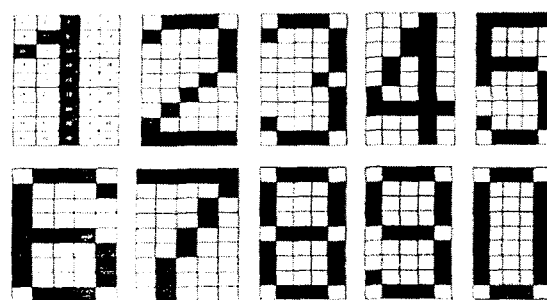


图 2 数字的  $9 \times 5$  位图

表 2 数字识别系统输入和预期的输出模式

数 字	输入模式 $p$ (45 行 10 列矩阵)										预期的输出 模式 $T$ (10 行 10 列矩阵)
	$p_1$	.....	$p_{45}$								$T_1 \dots T_{10}$
1	00100	01100	10100	00100	00100	00100	00100	00100	00100	00100	1000000000
2	01110	10001	00001	00001	00010	00100	01000	10000	11111	0100000000	0100000000
3	01110	10001	00001	00001	00010	00001	00001	10001	01110	0010000000	0010000000
4	00010	00110	00110	01010	01010	10010	11111	00010	00010	0001000000	0001000000
5	11111	10000	10000	11110	10001	00001	00001	10001	01110	0000100000	0000100000
6	01110	10001	10000	10000	11110	10001	10001	10001	01110	0000010000	0000010000
7	11111	00001	00010	00010	00100	00100	01000	01000	01000	0000001000	0000001000
8	01110	10001	10001	10001	01110	10001	10001	10001	01110	0000000100	0000000100
9	01110	10001	10001	10001	01111	00001	00001	10001	01110	0000000010	0000000010
0	01110	10001	10001	10001	10001	10001	10001	10001	01110	0000000001	0000000001

隐含层神经元的个数选 5 ( $S1 = 5$ )。隐含层节点的激活函数选 `logsig`; 输出层节点的激活函数选 `logsig`; 训练函数选 `trainlm`。

### 3.2 遗传算法主要参数设计

染色体编码采用实数编码, 适应度函数为误差平方和的倒数, 种群规模为 50, 遗传代数为 80, 选择函数为 `normGeomSelect`, 交叉函数为 `arithXover`, 变异函数为 `nonUnifMutation`。

### 3.3 MATLAB 程序

主程序: `GABPNETSHUZI.m`。它包括以下子程序: 适应度计算函数 `gabpEvalSHUZI.m`; 将遗传算法的编码解码为 BP 网络所对应的权值、阈值函数 `ga-decodSHUZI.m`。

主程序如下(子程序从略):

```
function net = GABPNETSHUZI(XX, YY)
% GABPNETSHUZI.m
% 使用遗传算法对 BP 网络权值阈值进行优化, 再用 BP 算法训练网络
nntwarn off
load XYY % 加载数据
% 创建网络
net = newff(minmax(XX), [5, 10], {'logsig', 'logsig'}, 'trainlm');
% 下面使用遗传算法对网络进行优化
P = XX; T = YY;
[R, N1] = size(P); [S2, N1] = size(T)
S1 = 5; % 隐含层节点数
S = R * S1 + S1 * S2 + S1 + S2; % 遗传算法编码长度
aa = ones(S, 1) * [-1, 1];
popu = 50; % 种群规模
initPpp = initializega(popu, aa, 'gabpEvalSHUZI'); % 初始化种群
gen = 80; % 遗传代数
% 下面调用 gaot 工具箱, 其中目标函数定义为 gabpEvalSHUZI
[x, endPop, bPop, trace] = ga(aa, 'gabpEvalSHUZI', [], initPpp, [1e-6 1 1], 'maxGenTerm', gen, 'normGeomSelect', [0.09], ['arithXover'], [2], 'nonUnifMutation', [2 gen 3]);
% 绘收敛曲线图
figure(1)
plot(trace(:, 1), trace(:, 3), 'r-');
xlabel('Generation');
ylabel('Sum-Squared Error');
```

figure(2)

```
plot(trace(:, 1), trace(:, 3), 'b-');
```

```
xlabel('Generation');
```

```
ylabel('Fitness');
```

% 将遗传算法的结果分解为 BP 网络所对应的权值与阈值

```
[W1, B1, W2, B2, P, T, A1, A2, SE, val]
```

```
= gadecondSHUZI(x);
```

```
XX = P; YY = T;
```

```
% 设置训练参数
```

```
disp_fqre = 100; max_epoch = 3000;
```

```
err_goal = 0.002; lr = 0.01;
```

```
TP = [disp_fqre max_epoch err_goal lr];
```

```
% 训练网络
```

```
[W1, B1, W2, B2, te, tr] = trainlm(W1, B1, 'logsig', W2, B2, 'logsig', XX, YY, TP);
```

% 为使网络对输入有一定的容错能力, 再利用不含和含有噪声的输入数据训练网络

```
net.trainParam.epochs = 500;
```

```
net.trainParam.goal = 0.002;
```

```
net1 = net;
```

```
YY1 = [YY YY YY YY];
```

```
for i = 1:10
```

```
    XX1 = [XX XX (XX-randn(R, N1) * 0.1)
```

```
    (XX-randn(R, N1) * 0.2)];
```

```
    [net1, tr, Y, E, Pf, Af] = train(net1, XX1,
```

```
    YY1);
```

```
end
```

```
y1 = sim(net1, XX);
```

% 为了保证网络总能够正确对理想输入信号进行识别, 再次用理想信号进行训练 net1.trainParam.epochs = 2000

```
net1.trainParam.goal = 0.001
```

```
[net1, tr, Y, E, Pf, Af] = train(net1, XX, YY;
```

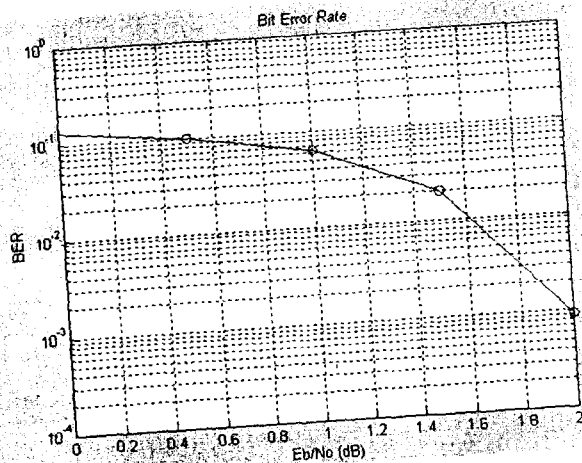
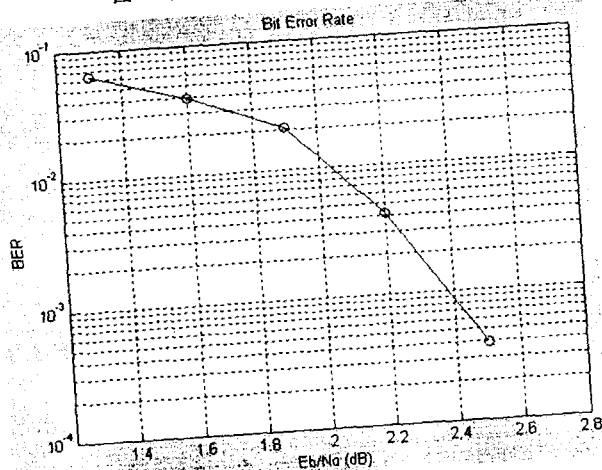
```
% 定义网络的输入为 1, 3, 5, 7; 仿真并输出结果
```

```
XX_test = XX(:, 2:2:9);
```

```
y2 = sim(net1, XX_test)
```

### 3.4 评价与比较

网络训练完成后, 就可以用测试集来测试网络, 检查它的性能。测试集与训练集完全无关, 否则会导致“用训练集进行测试”的方法论上的错误。应该用包含“噪声”的例子——扭曲的输入模式。例如, 从表示 10 个数字的位图的输入向量的正态分布中选择一些小的随机值 (MATLAB 中用 `Randn` 函数产生) 加入测试模式中。测试方案可采用 (下转第 80 页)

图4 分层解码下  $R=1/2$  的 BER 曲线图5 非分层解码下  $R=2/3A$  的 BER 曲线

(上接第76页)神经网络分类器的统计评估:“简单验证”或“m-重交叉验证”。

测试表明,遗传神经网络比纯 BP 神经网络识别误差率小,平均分类精度高。虽然用遗传算法寻找最优权重与阈值需要一些时间,内存消耗也较大;但是对 BP 的训练非常快,而且很快就收敛到了目标值。经多次训练,基本在 50 代以内就找到了最优结构,对于优化后的 BP 神经网络的学习结果,基本在 50 步以内就能收敛到最优解。少数情况下,遗传神经网络的学习需要比较多的代数,甚至在最大迭代步数内不能找到最优解,但这样的情况出现的概率比纯 BP 神经网络要小得多。用遗传算法优化神经网络的算法,在处于极值点的时候能使 BP 神经网络的学习跳出局部极值点,从而继续搜索找到最优解。可以看出遗传算法对神经网络的性能优化有明显效果。

#### 4 结束语

遗传神经网络是进化计算与神经网络结合在一

而采用改进型 SP 分层解码算法的解码延时比不采用分层解码算法的延时降低了 20%,而在  $R=2/3A$ ,  $SNR=2.4$  时误码率已经降低到了  $10^{-3}$  以下。

#### 5 结束语

本文提出了一种基于 WiMax 802.16e 采用修改的 SP 算法的 LDPC 码解码器的结构,该结构支持六种不同码率,其中码字最大长度为 2304bit 的 LDPC 码的解码,并且分别对采用分层解码和不采用分层解码两种情况进行了仿真验证,同时降低了解码延时和误码率。

#### 参考文献:

- [1] Gallager R G. Low density parity check codes[J]. IRE Trans. Inform. Theory, 1962, IT-8: 21-28.
- [2] Mansour M M, Shanbhag N R. High-Throughput LDPC Decoders[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2003, 11(6): 976-996.
- [3] Hocevar D E. A reduced complexity decoder architecture via layered decoding of LDPC codes[C]//Proc. IEEE Workshop on Signal Processing Systems (SiPS'04), Austin, USA, October 2004: 107-112.
- [4] Gunnam K, Choi G, Yeary M. VLSI Architectures for Layered Decoding for Irregular LDPC Codes of WiMax[C]. Communications, ICC'07. IEEE International Conference, 2007.
- [5] Hu X Y, et al. Efficient implementations of the sum-product algorithm for decoding LDPC codes[C]//Proc. IEEE GLOBECOM, San Antonio, TX, 2001, 2: 1036-1036E.
- [6] IEEE Std 802.16e[S]. 2005 AMENDMENT 2 AND CORRIGENDUM 1 TO IEEE Std 802.16, 2004: 626-630.

责任编辑:公丽苹

起的混合智能系统,体现了计算智能的精华部分。而 MATLAB 中的工具箱使遗传神经网络的建立、训练以及仿真都变得非常简单,可以使用户从学科范围内的基础程序编写的繁重负担中解脱出来,更多地致力于研究待解决问题。并且训练过程及效果非常直观,为广大科研工作者提供了便利。

#### 参考文献:

- [1] 李国勇.智能控制及其 MATLAB 实现[M].电子工业出版社, 2007: 122-126.
- [2] [澳]Michael Negnevitsky.人工智能.智能系统指南[M].机械工业出版社, 2007: 196-224.
- [3] 苏高利,邓芳萍.论基于 MATLAB 语言的 BP 神经网络的改进方法[J].科技通报, 2003, 19(2): 130-135.
- [4] Frank H F Leung, Lam H K, Ling S H. Tuning of structure and parameters of neural network using an improved genetic algorithm[J]. IEEE Trans on Neural Networks, 2003, 14(1): 79-88.
- [5] Rovithakis G A, Chalkiadakis I, Zervakis M E. High-order neural network structure selection using genetic algorithms[J]. IEEE Transactions on Systems, Man and Cybernetics, 2004, 34(1): 150-158.

责任编辑:公丽苹