

# 利用遗传模拟退火算法优化神经网络结构

李振涛, 王淑玲, 张国立

LI Zhen-tao, WANG Shu-ling, ZHANG Guo-li

华北电力大学 数理系, 河北 保定 071003

Department of Mathematics and Physics, North China Electric Power University, Baoding, Hebei 071003, China

E-mail: lizt168@sohu.com

LI Zhen-tao, WANG Shu-ling, ZHANG Guo-li. Application genetic and simulated annealing algorithm for optimization of neural network structure. *Computer Engineering and Applications*, 2007, 43(36): 74-76.

**Abstract:** A conventional neural network often optimizes the weights through invariable network structure which has limited the extensive use of the neural network. The crossover operator based on direction and mutation are introduced. This paper puts forward genetic and simulated annealing algorithm to train the neural networks, combining the merits of genetic algorithm and that of simulated annealing algorithm, which makes weights and structure of artificial neural networks be optimized together. The result shows that the neural network optimized by the presented algorithm has the advantages of quicker convergence rate and higher precision compared with genetic algorithm and simulated annealing algorithm, and that the processing ability of networks is raised.

**Key words:** genetic algorithm; simulated annealing; neural networks; optimization

**摘要:** 常用的神经网络是通过固定的网络结构得到最优权值, 使网络的实用性受到影响。引入了一种基于方向的交叉算子和变异算子, 同时把模拟退火算法引入了遗传算法, 结合遗传算法和模拟退火算法的优点, 提出了一种优化神经网络结构的遗传——模拟退火混合算法, 实现了网络结构和权值的同时优化。仿真实验表明, 与遗传算法和模拟退火算法相比, 该算法优化的神经网络收敛速度较快、预测精度较高, 提高了网络的处理能力。

**关键词:** 遗传算法; 模拟退火算法; 神经网络; 优化

**文章编号:** 1002-8331(2007)36-0074-03 **文献标识码:** A **中图分类号:** TP183

## 1 引言

神经网络是由多个简单的处理单元彼此按某种方式相互连接而形成的计算系统, 该系统通过对连续或断续式的输入作状态响应而进行信息处理。虽然每个神经元的结构和功能十分简单, 但由大量神经元构成的网络系统的行为却丰富多彩和十分复杂。神经网络在国民经济和国防科技现代化建设中具有广阔的应用领域和应用前景。主要应用领域有: 语音识别、图像识别、计算机视觉、智能机器人、故障机器人、故障检测、实施语言翻译、企业管理、市场分析、决策优化、物资调运自适应控制、专家系统、智能接口、神经生理学、心理学和认知科学研究等等。随着神经网络理论研究的深入以及网络计算能力的不断拓展, 神经网络的应用领域将会不断拓展, 应用水平将会不断提高, 最终达到神经网络系统可用来帮人做事的目的, 这也是神经网络研究的最终目标。

神经网络具有很强的自适应、自组织和自学习的能力, 具有大规模并行运算的能力。常用的神经网络系统的行为主要由两方面因素决定: 其一为拓扑结构; 其二是节点间连接权值。一般来讲, 在应用神经网络过程中只是得到最优权值, 结构在整个过程中是固定的, 而且在确定结构时, 经常根据人为经验或经验公式先拟定一个结构, 在此基础上不断地训练网络、调整

结构, 最后得到确定的网络结构, 进而应用这个固定的网络结构得到最优的权值。因此, 构造一个准确定量构效关系人工神经网络模型需要耗费大量的时间, 有时利用该方法构造的模型结构还可能不是最佳的<sup>[1]</sup>。近年来, 用遗传算法(Genetic Algorithms, GA)解决神经网络优化设计问题受到广泛重视。GA在寻优时搜索方向不停的发生变化, 具有较强的全局搜索能力。但是, 在应用遗传算法寻优的过程中, 仍存在着困难<sup>[2]</sup>: 遗传算法对好的个体易造成破坏, 易出现早熟。因此, 许多学者采取了遗传算法和其它算法相结合<sup>[2,3]</sup>的措施, 如先用遗传算法进行全局搜索, 确定最优解所在的区域后, 再用BP算法(拟牛顿LM算法)进行局部搜索, 但仍然存在耗时较多、精度较低的不足。

由Kolmogorov定理可知在有合理的结构和恰当的权值条件下, 三层前向网络能以任意精度逼近有界非线性函数。对于三层神经网络来说, 输入层、输出层节点的个数由目标函数结构决定, 故而在优化网络结构时, 也优化它的隐层节点的个数。本文将对应用最为广泛的三层前向网络进行优化设计, 把基于方向的遗传交叉算子和变异算子引入了遗传算法, 并且与模拟退火算法(Simulated Annealing, SA)结合, 提出了自适应遗传-模拟退火混合算法, 使在得到神经网络最优权值的同时, 也自动得到最优网络结构。仿真研究表明, 此算法在以负荷作为训

**基金项目:** 国家自然科学基金(the National Natural Science Foundation of China under Grant No.50077007); 留学归国人员基金(No.93409501)。

**作者简介:** 李振涛(1975-), 男, 硕士研究生, 主要研究方向: 智能优化理论及其应用; 王淑玲(1972-), 女, 硕士研究生, 主要研究方向: 信息处理技术; 张国立(1959-), 男, 教授, 主要研究方向: 智能计算理论、方法及其在电力市场中的应用。

练样本的神经网络中具有良好的实用性。不仅能确定出最佳隐层节点个数,同时能生成落在全局最优区域的权值。

## 2 遗传算法

遗传算法是近年来迅速发展起来的一种全新的随机搜索与优化算法,其基本思想是基于 Darwin 的进化论和 Mendel 的遗传学说。该算法由密执安大学教授 Holland 及其学生于 1975 年创建<sup>[4]</sup>。随后经过 20 多年的发展,取得了丰硕的应用成果和理论研究成果,特别是近年来世界范围内形成的进化计算热潮,计算智能已作为人工智能研究的一个重要方向,以及后来的人工生命的兴起,使 GA 受到广泛的关注。遗传算法主要通过选择复制和遗传算子的作用使种群不断进化,最终收敛到优化解。遗传算法是一种并行搜索过程,它具有较强的全局寻优能力。

### 2.1 编码方式<sup>[5]</sup>

为了计算简便,本文省略神经网络阈值。个体的长度由最大隐层节点个数决定,即  $l=(r+c) \cdot h_{\max}+h_{\max}$  其中: $r$  为输入节点数、 $h_{\max}$  为最大隐层节点数、 $c$  为输出节点数。将同一个隐层节点所连接的权值排在一起,当删除或增加隐层节点时,方便操作。将隐层节点个数及网络权值进行混合实数编码,另外对隐层节点增加了一个 0-1 组成的数串用作控制码。控制码主要是控制隐层节点的个数,0-1 数串中的一个数控制一个隐层节点。具体编码方法<sup>[5]</sup>如图 1 所示。

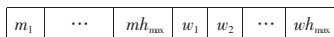


图 1 编码方式

图 1 中, $m_1$  是表示第一个隐层节点有效性的 0-1 控制码, $mh_{\max}$  是表示第  $h_{\max}$  个隐层节点有效性的 0-1 控制码, $w_1$  是与第一个隐层节点相关的权值, $w_2$  是与第二个隐层节点相关的权值, $wh_{\max}$  是与第  $h_{\max}$  个隐层节点相关的权值。

初始群体中不同隐层节点个数的个体按照一定的比例随机产生。当 0-1 数串中的数为 0 时,其所对应的隐含层节点对输出层没有作用(即相当于不存在此神经元),当数串中的数为 1 时,就相当于此隐含层节点对输出层有作用。

采用上述方法编码,每个个体译码后的网络结构都满足前面所述的研究前提条件,因此不存在所对应的网络结构无效的个体。

### 2.2 适应度函数的确定及选择运算

本文采用如式(1)所示适应度函数

$$f = \frac{1}{E} \cdot [1 + a \cdot (1 - \frac{N_y}{N_{y\max}})] \quad (1)$$

式中: $f$  为神经网络的适应度, $E$  为神经网络的误差平方和, $a$  为隐含层神经元数量的影响系数, $N_{y\max}$  为隐含层神经元的最大数量, $N_y$  为对输出层节点有作用的隐含层节点的数量。对于选择操作采用轮盘赌方法。

### 2.3 交叉率和变异率的确定

本文采用自适应的交叉率和变异率,其公式如下:

$$pc = \begin{cases} k_1 \cdot (f_{\max} - f^i) / (f_{\max} - f_{avg}) & f^i \geq f_{avg} \\ k_2 & f^i < f_{avg} \end{cases}$$

$$pm = \begin{cases} k_3 \cdot (f_{\max} - f_i) / (f_{\max} - f_{avg}) & f_i \geq f_{avg} \\ k_4 & f_i < f_{avg} \end{cases}$$

式中  $k_1, k_2, k_3, k_4$  是取值范围为  $[0, 1]$  的常数, $f^i$  是要交叉的两个个体适应度中较大的一个, $f_i$  是要变异的个体的适应度, $f_{\max}$

是种群中最大的适应度, $f_{avg}$  是种群的平均适应度。

## 2.4 交叉算子

对控制基因和实数编码的基因分别采用不同的交叉算子。对于控制基因,采用两点交叉;对于用实数编码的代表各个节点权值的基因,采用下面所述交叉方法:

假设要参与交叉操作的两个个体分别为  $x_i, x_j$ , 两个个体上相对应位的权值分别为  $x_i, x_j$ , 并且  $x_i$  的适应度高于  $x_j$  的适应度,定义式(2)、(3)所示的  $\Delta_i^1, \Delta_j^1$  两个中间变量:

$$\Delta_i^1 = \begin{cases} \min\{x_i + \frac{1+pc}{2}(x_i - x_j), x_{\max}\} & x_i \geq x_j \\ \max\{x_i + \frac{1+pc}{2}(x_i - x_j), x_{\min}\} & x_i < x_j \end{cases} \quad (2)$$

$$\Delta_j^1 = \begin{cases} \max\{x_j + \frac{1-pc}{2}(x_j - x_i), x_{\min}\} & x_i \geq x_j \\ \min\{x_j + \frac{1-pc}{2}(x_j - x_i), x_{\max}\} & x_i < x_j \end{cases} \quad (3)$$

式中  $x_{\max}, x_{\min}$  分别为  $x_i(x_j)$  的取值上限和下限,这样,交叉后所产生的两个新个体  $y_i, y_j$  可以由式(4)、(5)确定:

$$y_i = \frac{1+pc}{2} \cdot \Delta_i^1 + \frac{1-pc}{2} \cdot \Delta_j^1 \quad (4)$$

$$y_j = \alpha \cdot x_i + (1-\alpha) \cdot x_j \quad (5)$$

式中  $\alpha$  为一个在  $[0, 1]$  之间的随机数。从式(2)~(5)可以看出,这种交叉算子可以保证产生的两个个体一个在两个父代所在的区域之间搜索,另一个向适应度高的父代个体的方向搜索新的领域。交叉后,根据有作用的隐层节点数确定相应长度的权值域,使得有作用的网络权值的数量与网络结构相符。

## 2.5 变异算子

随机取出需要变异的一个个体,再从控制码中随机选出一个数,这个数的变异采用基本变异算子。若这个数变异后的控制基因是 1, 则从控制基因 1 对应的权值串中选一随机位,设这一随机位上的权值为  $x_c, x_c$  的变异为:在区间  $[x_1, x_2]$  上随机取一数  $x_c^1$  代替  $x_c$ 。区间  $[x_1, x_2]$  如式(6)、(7)确定。

$$x_1 = x^{\min} + \frac{x^{\min} \cdot p_m \cdot f_i}{f_{\max}} \quad (6)$$

$$x_2 = x^{\max} + \frac{x^{\max} \cdot p_m \cdot f_i}{f_{\max}} \quad (7)$$

从式(6)、(7)可以看出,适应度大的个体变异区间较小,适应度小的个体的变异区间较大。这样,既能减少变异操作对好的个体的破坏,又能够保证遗传禁忌算法的搜索能力。变异后,根据有作用的隐层节点数确定相应长度的权值域,使得有作用的网络权值的数量与网络结构相符。

## 3 模拟退火算法

模拟退火算法的思想最早由 Metropolis 等于 1953 年提出<sup>[6]</sup>。SA 又称为模拟冷却法、统计冷却法、Monte—Carlo 退火法、随机松弛法和概率爬山法等。1983 年 Kirkpatrick<sup>[7]</sup>等开展了一些富有成效的工作,成功地将该思想引入组合优化理论,解决了许多诸如 VLSI 等大规模优化设计问题。SA 算法是基于 Monte Carlo 迭代求解策略的一种随机寻优方法,其出发点是基于物理中固体物质的退火过程与一般组合优化问题之间的相似性。SA 算法在某一初温下,伴随温度参数的不断下降,结合概率突跳特性在解空间中随机寻找目标函数的全局最优解,即在局部最优解能概率性的跳出并最终趋于全局最优<sup>[8]</sup>。SA 算法

具有质量高、初始鲁棒性强、通用易实现的优点。但是为寻到最优解,算法通常要求较高的初温、较慢的降温、较低的终止温度以及各温度下足够多的抽样,因而SA算法往往优化过程较长,这是SA算法的最大缺点。

#### 模拟退火算法基本原理<sup>[9]</sup>

(1)给出初始温度  $T_0$  与退火次数  $Num$  以及初始点  $x_p$ , 计算该点的适应度值  $f(p)$ 。

(2)产生随机扰动向量  $\Delta x$ , 得到新点  $x_p' = x_p + \Delta x$ , 计算新点的适应度  $f(p')$ , 以及适应度值之差  $\Delta f = f(x_p') - f(x_p)$ 。

(3)如果  $\Delta f \geq 0$ , 则接受新点, 作为下一次模拟的初始点。

(4)如果  $\Delta f < 0$ , 则计算新点的接收概率:  $P(\Delta f) = \exp(\Delta f \cdot T)$ , 产生  $[0, 1]$  区间上的均匀随机数  $rand$ , 如果  $P(\Delta f) \geq rand$ , 则接受新点作为下一次模拟的初始点, 否则放弃新点, 仍取原来的点作为下一次模拟的初始点。

(5)  $T_{Num+1} = \lambda(T_{Num})$ ,  $Num = Num + 1$ , 若满足停止条件, 则停止计算, 否则, 返回(2)。

## 4 训练步骤

**步骤 1** 设定三层前向神经网络的输入节点  $r$ 、隐含层节点  $h$ 、输出节点  $c$ 、初始温度  $T_0$ 、退火次数  $Num$ 、退火个体数  $N$ ;

**步骤 2** 在  $[0, 1]$  范围内随机产生  $pp$  条染色体, 得到初始种群, 并设初始进化代数  $L=0$ ;

**步骤 3** 由设定的选择概率按选择机制对个体选择;

**步骤 4** 由自适应交叉概率按交叉算子进行交叉操作;

**步骤 5** 由自适应变异概率按变异算子进行变异操作;

**步骤 6** 把由步骤 3、4、5 得到的个体放到一起作为中间种群;

**步骤 7** 计算中间种群的适应度, 将适应度大的个体分别进行模拟退火  $Num$  次;

**步骤 8** 将模拟退火次数后得到的  $N$  个最好的个体替代中间种群中取出的  $N$  个个体, 组成新的种群;

**步骤 9** 如果进化代数小于预先设定的最大遗传代数, 令  $L=L+1$ , 转步骤 3, 否则, 终止网络的训练, 选出最优个体, 转步骤 10;

**步骤 10** 将最优个体按顺序拆分, 作为 BP 网络的最优权值;

**步骤 11** 输入与训练样本集模式相一致的预测样本, 得到预测值, 并计算出相对误差。

## 5 仿真实验及结果

本文在河北地区 1999 年 7 月 1 日至 2003 年 12 月 31 日的历史负荷中筛选了多组样本数据, 训练神经网络。在训练神经网络时所用到的主要参数如表 1 所示。

表 1 主要参数

	本文算法	GA 算法	SA 算法
输入节点数	6	6	6
隐层节点数	4-9	4-9	4-9
输出节点数	1	1	1
种群规模	30	30	1
初始权值范围	$[-1, 1]$	$[-1, 1]$	$[-1, 1]$
初始温度	10 000		10 000
退火次数	20		1 000
运行代数	10	10	1 000

训练神经网络时所用的激活函数是 Sigmoid 函数。在交叉和变异的过程中, 可能出现隐层节点为 1、2、3 的个体, 但由于

它们的适应度较小, 不会影响最优结构及最优权值。

下面列出了本文算法独立训练神经网络 7 次对应的最优神经网络结构, 以及权值(如表 2 所示)和由三种算法得到的负荷预测结果(如表 3 所示)。

表 2 7 次得到的最优神经网络结构及权值

序号	1	2	3	4	5	6	7
隐层节点	4	4	4	5	7	5	6
	-0.652 1	-0.279 7	-0.113 6	0.036 3	-0.623 1	0.335 3	0.997 8
	0.761 1	0.478 7	0.574 7	0.286 1	-0.247 9	-0.398 4	0.278 4
	0.110 2	-0.969 8	-0.998 4	0.643 2	0.915 4	0.360 5	0.060 2
	0.923 5	0.778 2	-0.928 3	-0.530 5	-0.556 6	0.605 9	0.326 7
	0.463 5	-0.473 2	0.431 9	-0.519 8	-0.476 1	0.499 0	-0.800 1
	0.091 4	-0.619 8	-0.141 2	0.790 9	-0.682 7	-0.893 4	-0.778 6
	-0.279 0	-0.084 0	0.250 3	-0.135 1	0.244 3	0.667 1	-0.151 5
	0.939 6	-0.604 7	-0.827 3	-0.528 8	-0.286 8	-0.299 3	-0.016 9
	-0.178 0	-0.365 4	0.672 4	0.177 6	-0.709 2	-0.539 8	-0.820 7
	0.564 6	0.341 7	0.176 4	0.067 9	0.375 8	-0.214 9	0.905 1
	-0.926 0	0.493 9	0.633 6	-0.240 5	0.059 5	0.224 9	-0.017 8
	0.481 5	0.076 2	-0.836 9	0.196 5	0.760 3	0.012 0	0.689 0
	0.149 2	-0.274 6	-0.523 6	-0.092 4	-0.875 9	-0.298 0	0.057 7
	0.897 1	0.023 0	0.091 8	-0.231 8	0.547 1	0.630 4	0.034 8
	0.218 0	0.028 0	-0.898 9	0.928 9	0.514 3	-0.099 3	0.471 0
	-0.275 6	0.772 6	-0.281 5	0.650 8	0.583 8	-0.540 9	0.349 2
	-0.284 6	-0.926 4	0.780 2	-0.711 9	0.353 8	-0.022 3	-0.731 1
	-0.208 3	0.465 7	-0.258 3	-0.144 8	0.098 4	-0.686 1	0.404 0
	-0.343 1	-0.553 7	0.536 8	-0.763 3	0.138 2	-0.197 2	0.392 9
	0.539 2	0.812 4	0.671 7	-0.562 3	-0.260 3	0.256 1	-0.286 1
	-0.881 1	0.216 3	0.988 3	-0.623 8	-0.671 3	0.391 3	0.751 1
相应权值	-0.989 7	0.022 3	0.886 9	0.952 9	-0.829 4	0.548 4	0.293 0
	0.356 3	-0.359 8	0.712 9	-0.963 3	-0.537 6	-0.415 0	0.859 0
	-0.493 5	-0.346 3	-0.836 0	0.833 7	0.337 2	-0.852 3	0.846 9
	-0.181 9	-0.918 5	0.247 0	-0.743 6	0.234 3	-0.828 5	-0.354 0
	0.513 6	0.046 2	0.543 0	0.503 6	-0.575 3	-0.432 5	-0.527 9
	0.764 7	0.340 4	0.417 8	0.171 7	-0.448 4	0.907 1	-0.187 2
	-0.195 4	-0.310 6	-0.755 5	0.782 3	0.971 4	-0.196 2	-0.627 0
				-0.852 3	-0.537 8	0.916 2	0.179 1
				-0.226 0	-0.778 7	0.841 1	0.241 6
				-0.417 6	0.361 2	0.297 7	0.203 5
				0.207 8	0.851 8	-0.721 8	-0.732 1
				0.473 0	0.955 4	-0.851 1	0.485 1
				-0.834 6	-0.698 1	-0.974 0	-0.160 1
				0.432 7	0.916 5	-0.937 2	0.910 2
						-0.047 4	0.431 6
						0.633 4	-0.991 2
						-0.441 5	-0.707 9
						0.574 3	-0.847 8
						-0.033 5	0.513 0
						-0.380 9	-0.336 5
						-0.369 1	0.011 4
						-0.678 6	
						-0.704 6	
						0.533 9	
						-0.029 3	
						0.333 1	
						-0.143 7	
						-0.924 5	

## 6 结论

本文提出的遗传模拟退火算法是一种同时优化神经网络(下转 156 页)