

# 遗传算法构建的神经网络及在机械中的应用

刘道华<sup>1,2</sup>, 原思聪<sup>1</sup>, 汪金友<sup>2</sup>, 赵进昌<sup>1</sup>

(1. 西安建筑科技大学机电工程学院, 陕西 西安 710055;

2. 信阳师范学院计算机科学系, 河南 信阳 464000)

**摘要:** 在分析遗传算法和神经网络优点的基础上, 采用遗传进化的方式自动获得神经网络的结构、权值和阈值. 提出了构建神经网络模型参数的遗传算法分区编码方案, 构建了适应度函数并依据个体适应度的大小动态调整隐层节点及连接权个数, 给出了整体算法过程. 采用该方法构建的神经网络计算两自由度的机械手参数, 并通过实例仿真与常规凭经验构建网络结构及采用 BP 学习算法相比较, 采用遗传算法构建的神经网络具有仿真精度高、占用资源少、计算效率高等优点.

**关键词:** 遗传算法; 神经网络; 机械实例; BP 算法; 自适应参数调整

**中图分类号:** TP18    **文献标识码:** A    **文章编号:** 1001-2400(2008)01-0152-05

## Neural networks based on the genetic algorithm and its application in mechanical engineering

LIU Dao-hua<sup>1,2</sup>, YUAN Si-cong<sup>1</sup>, WANG Jin-you<sup>2</sup>, ZHAO Jin-chang<sup>1</sup>

(1. School of Mech. & Elec. Eng., Xi'an Univ. of Arch. & Tech., Xi'an 710055, China; 2. Dept. of Computer Science, Xinyang Normal Univ., Xinyang 464000, China)

**Abstract:** Authors take advantage of the genetic algorithm (GA) to automatically obtain structures, weights and bias of neural networks (NN). A classified coding scheme is presented to get modeling parameters of an NN. Then a practical fitness function along with a new method that can automatically adjust the number of hidden nodes and connection weights according to the individual fitness values is described in detail. The proposed method is applied to calculate the parameters of a manipulator with a freedom of degree 2. Simulation result is compared with data obtained from practical experience and the back propagation (BP) learning algorithm. Comparison study indicates that the proposed method has many advantages such as higher simulation accuracy, less resource utilization and higher computational efficiency.

**Key Words:** Genetic algorithm; neural network; machinery example; back propagation algorithm; self-adaptive parameter adjustment

神经网络(Neural Network, 简称 NN)具有自适应能力强、鲁棒性好等优点, 然而在神经网络结构创建上没有统一的理论作指导, 人们在使用 NN 时往往都是凭经验构建网络的结构, 并采用层间全连接的方式, 然后采用 BP 学习算法训练网络权值及阈值<sup>[1]</sup>. 这种传统方法构建 NN 的方式, 其缺点最主要体现在隐层数及各层节点数可能因多次试运算才能使网络收敛, 因而效率不高; 其二, 由于采用层间全连接方式, 既扩大了网络的复杂度, 又使计算代价提高; 其三, 采用 BP 反向传播算法, 即依靠能量函数的可导信息并采用梯度误差反向学习, 易陷于局部极小, 同时对于那些提炼不出可导函数的问题, 此种方法便无法实现. 也有很多学者在 BP 算法上加以改进, 但均难以摆脱因采用梯度方式自身易陷于局部极小的困惑. 在网络结构上也有文献总结出创建多少个隐层数及各层需要多少个节点的经验性规则<sup>[2]</sup>, 但因求解问题不同其经验性规则也有很

收稿日期: 2007-06-28

基金项目: 国家自然科学基金资助(50275113); 陕西省自然科学基金资助(2007E218); 信阳师范学院青年基金资助(20070204)

作者简介: 刘道华(1974-), 男, 讲师, 西安建筑科技大学博士研究生, E-mail: ldhzzx@163.com.

大的局限性. 而遗传算法(Genetic Algorithm, 简称 GA)全局搜索性能强, 因此将两者结合起来, 采用 GA 自适应创建网络结构并在机械工程应用中具有现实的意义. 当然也有文献采用遗传算法并用浮点矩阵的编码方式创建网络结构<sup>[3]</sup>, 但该方法因权值及阈值表示易混淆, 且不易动态创建隐层节点.

笔者提出了一种实数分区编码方案, 将网络的拓扑结构、连接权和阈值级联分区编码, 在满足某一特定性能指标的前提下, 采用遗传算法寻找全局最优解; 给出了 GA 动态创建网络参数的整体算法; 结合具体的机械设计实例, 并同 BP 算法作对比, 从而体现该方法的优越性.

## 1 采用 GA 构建 NN 结构参数的方法

### 1.1 基本原理

GA 构建 NN 结构参数的基本原理是利用遗传进化的优点, 将网络的拓扑结构、连接权及阈值连成分区编码串而构成进化的初始群体, 通过遗传算子操作, 在满足某一特定性能指标的前提下, 寻找全局最优解, 而该最优解的编码串经解码即为特定求解问题的合适神经网络结构参数.

### 1.2 关键技术的解决

在 GA 构建 NN 结构的过程中, 要求解决三个关键问题: 其一是要将网络的结构、权重及阈值形成统一的编码串; 其二是遗传进化只依靠适应度评价函数这一惟一信息来指导进化的过程, 因此适应度函数的定义相当关键; 最后就是采用合适的机制使网络结构具有动态调整功能.

#### 1.2.1 编码方式

传统的 GA 编码方式常采用二进制编码, 但因该编码需多次转化使计算复杂度提高, 也有文献采用浮点数矩阵表示, 但因为构成 NN 的结构及权重都采用实数编码不易区别那些是权值信息, 那些是结构信息, 因此笔者提出了 GA 的实数分区编码思想, 即将网络的结构、权值及阈值统一编码.

描述一个 NN 模型的参数主要包括网络层数、各层神经元数、神经元的互连方式、各连接权值、阈值及传递函数等<sup>[4]</sup>. 而 Kosmagoro 定理说明在有合理的结构和恰当权值的条件下, 三层前馈网络可以逼近任一可微函数到任意精度<sup>[5]</sup>. 同时对于复杂问题能够采用增加层数及节点数的方式同层数固定只增加层内节点数的方式相等价的原理. 为计算方便, 本文中采用固定的 3 层网络结构, 即输入层、隐层、输出层各一个的结构. 基于这些分析, 整个编码由 4 部分组成, 即:

$$\begin{array}{c} \overbrace{a_{1_{11}} \ a_{1_{12}} \ \cdots \ a_{1_{1L_1}} \ a_{1_{21}} \ a_{1_{22}} \ \cdots \ a_{1_{2L_2}} \ \cdots \ a_{1_{L_1 1}} \ a_{1_{L_1 2}} \ \cdots \ a_{1_{L_1 L_2}}}^{U_1} \quad \overbrace{a_{2_1} \ a_{2_2} \ \cdots \ a_{2_{L_2}}}^{U_2} \\ \overbrace{a_{3_{11}} \ a_{3_{12}} \ \cdots \ a_{3_{1L_3}} \ a_{3_{21}} \ a_{3_{22}} \ \cdots \ a_{3_{2L_3}} \ \cdots \ a_{3_{L_2 1}} \ a_{3_{L_2 2}} \ \cdots \ a_{3_{L_2 L_3}}}^{U_3} \quad \overbrace{a_{4_1} \ a_{4_2} \ \cdots \ a_{4_{L_3}}}^{U_4} \end{array}$$

其中  $a_{i_{kj}} \in \mathbf{R}$ , 网络整体编码的位串总长度  $L = L_1 \times L_2 + L_2 + L_2 \times L_3 + L_3$ ,  $L_1$  和  $L_3$  是相对固定的, 其大小由具体问题的输入与输出参数个数决定;  $U_1$  区为输入层节点到隐层的连接权编码,  $a_{1_{ij}}$  表示输入层第  $i$  节点同隐层第  $j$  节点的连接权编码;  $U_2$  区为隐层节点的阈值编码,  $a_{2_i}$  表示隐层第  $i$  节点的阈值编码;  $U_3$  区是隐层节点到输出节点的连接权编码,  $a_{3_{ij}}$  表示隐层第  $i$  节点同输出层第  $j$  节点的连接权编码;  $U_4$  区为输出节点各神经元的阈值编码,  $a_{4_i}$  表示输出层第  $i$  节点的阈值编码. GA 初始的编码以充分大的隐层个数  $L_2$  及层间全相连的方式组成初始群体, 随着进化过程中自适应地调整隐层节点数  $L_2$  及删去部分层间的连接权, 从而实现自适应地动态改变编码的长度.

#### 1.2.2 GA 适应度的确定

GA 的适应度函数(即 NN 性能评价函数)是算法指导搜索的惟一信息, 它的选取是算法好坏的关键<sup>[6]</sup>. 适应度函数要能有效地指导搜索沿着优化参数组合的方向, 逐渐逼近最佳参数组合, 而不会导致搜索不收敛或陷入局部最优的状况, 同时也要求这个函数易于计算. 本文中将初始群体  $p_0(N, L)$  中的每一码串进行解码, 从而组成  $N$  个前向网络, 其中每个码串的  $U_1$  与  $U_3$  区确定网络权值,  $U_2$  与  $U_4$  区确定网络的阈值, 其结构对应关系由  $U_1 U_2 U_3 U_4$  编码串下标对应关系确定, 以训练样本集作为每一个网络的输入和期望输出, 然后计算期望输出  $y(k)$  与实际输出  $\hat{y}(k)$  的平均误差平方和  $E(i)$ , 并将  $E(i)$  作为 GA 的适应度函数.

$$E(i) = \frac{1}{m} \sum_{k=1}^m \sum_{j=1}^{L_3} [y(k) - \hat{y}(k)]^2, \quad i = 1, 2, \dots, N, \quad (1)$$

式(1)中,  $E(i)$  表示第  $i$  个体的平均误差平方和,  $m$  为网络输入输出样本对个数,  $L_3$  为输出层节点个数.

由于不同结构的 NN 进化的情况不同, 如隐层节点数多的网络前期进化的较快而使种群多样性退化. 另一方面, 期望在相同性能条件下网络的结构尽可能的简单. 因此, 本文中采用式(2)作为改进的适应度函数:

$$F(i) = \frac{1}{E(i)} [1 + \alpha_i [1 - N_i/N_{\max}]] \quad i = 1, 2, \dots, N, \quad (2)$$

式(2)中,  $F(i)$  为第  $i$  个体的适应度,  $\alpha_i$  为隐层神经元数量的影响系数,  $N_i$  为第  $i$  个体所对应的隐层神经元的节点数,  $N_{\max}$  为设定的隐层神经元最大的节点数.

### 1.2.3 动态调整隐层节点及连接权个数

在 GA 进化过程中, 计算每一个体的适应度时, 要将编码转化为对应的 NN 结构, 并按式(2)计算各个体的适应度, 为提高进化个体的适应度值及进化个体的质量, 则事先规定每进化  $t$  代后, 如适应度变化量处在某一很小的阈值内, 则动态增加或减少隐层的节点数. 如增加隐层第  $i$  节点, 即将  $U_1$  编码区中随机添加  $a_{1_{i_1}} a_{1_{2i}} \dots a_{1_{L_1}}$  共  $L_1$  个连接权, 将  $U_2$  区添加  $a_{2_{2i}}$  一个节点阈值, 将  $U_3$  区添加  $a_{3_{i1}} a_{3_{i2}} \dots a_{3_{iL_3}}$  共  $L_3$  个连接权. 如删除第  $i$  隐层节点, 则将  $U_1$  编码区  $a_{1_{pi}}$  下标尾数为  $i$  的连接权全删除, 将  $U_2$  区中  $a_{2_i}$  删除, 将  $U_3$  区中的  $a_{3_{ip}}$  下标首数为  $i$  的所有连接权删除. 除 GA 过程中动态改变隐层节点数外, 还动态判断各连接权值是否大于事先设计的阈值, 如不满足则删除该连接权, 此时各个节点并不调整, 这样就使得层间节点不是全相连, 以至因减少了连接权数而提高了 GA 进化的速度.

当然这样处理会导致编码长度的动态变化, 对于 GA 选择操作并不影响, 因为进化时只选择编码串适应度高的个体, 因此不同串的长度只影响对应 NN 的结构, 而不影响适应度的计算. 对于交叉操作, 当以某种概率选中两个不等串长的个体时, 其交叉操作只在被选中分区内的交叉点处实现交叉运算, 而其余编码区不等长时则对应一次性代换, 这样更有利于使 GA 个体保持群体多样性. 变异操作只对变异点进行而与编码长度无关, 故变异操作也不受变长编码的影响.

## 2 整体算法及过程

(1) 初始化. 包括编码各分区的初始化, 基本遗传操作控制参数的初始化, 初始群体  $p_0(N, L)$ , 编码串长度  $L$ , 在  $(-1, 1)$  范围内随机设置初始权和阈值, 群体规模  $N$ .

(2) 参数集位串的解码操作. 将所有染色体串转化为对应的 NN 模型.

(3) 计算群体的个体适应度. 以所有的训练样本对通过式(2)计算各个体的适应度, 在计算适应度时, 先确定各对应网络层的激励函数, 对于隐层的激励函数采用公式(3)的形式.

$$f(x) = 1/(1 + \exp(-x)) \quad (3)$$

而输出层的激励函数采用公式(4)的形式.

$$f(x) = \pi(1 - \exp(-x))/(1 + \exp(-x)) \quad (4)$$

(4) 判断网络性能是否收敛到指定的精度范围内, 如达到则转步骤(10), 否则转步骤(5).

(5) 选择操作, 将所有的训练样本对通过式(2)计算各个体的适应度以决定该个体是否被选为下一代进化群体.

(6) 交叉操作, 任选一对个体  $(Z_i, Z_{i+1})$ , 在区间  $[0, 1]$  上产生一个随机数  $\lambda$ , 按式(5)在  $Z_i$  与  $Z_{i+1}$  之间进行交叉操作, 从而产生两个子个体  $A$  和  $B$ .

$$A = Z_i + \lambda(Z_{i+1} - Z_i) \quad B = Z_{i+1} + \lambda(Z_i - Z_{i+1}) \quad (5)$$

检验  $A$  和  $B$  的可行性, 保留其中可行的子个体. 若其中有不可行的子个体, 则需新产生随机数  $\lambda$ , 重新进行交叉操作, 直到获得两个可行的子个体或循环到预先给定的次数为止.

(7) 变异操作, 对每一变异父个体用  $Z = (x_1, x_2, \dots, x_m)$  表示, 并进行如下的变异操作, 在  $m$  维实空间  $R^m$  中随机产生一个变异方向  $H$ , 然后用

$$C = Z + GH \quad (6)$$

取代  $Z$ . 注意, 必须保证  $C$  是可行的. 因此在用  $C$  代替  $Z$  之前要对其进行可行性检查, 若  $C$  不可行, 则取  $G$  为  $(0, G)$  之间的随机数重新计算, 直到  $C$  可行为止.

- (8) 动态调整隐层节点数及连接权数, 具体方法参见 1.2.3 节.
- (9) 形成新一代种群, 转步骤(2).
- (10) 解码操作, 输出获得的 NN 结构、权值及阈值.

### 3 两自由度机械手实例分析

为了验证遗传算法创建 NN 的可行性及有效性, 以两自由度机械手逆运动学求解为例<sup>[7]</sup>, 应用 Matlab 软件进行仿真分析.

图 1 为两杆平面机械手, 机械手的前向运动学方程为

$$p_x = l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2) \quad (7)$$

$$p_y = l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2) \quad (8)$$

$p_x$  和  $p_y$  为机械手终端位置,  $\theta_1$  和  $\theta_2$  为机械手连杆转角, 为简化计算, 取杆长  $l_1$  和  $l_2$  为单位长 1. 图 2 给出了两杆机械手所要达到的期望运动轨迹, 欲求相应的连杆转角.

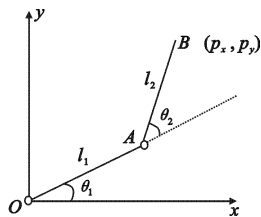


图 1 两杆平面机械手模型

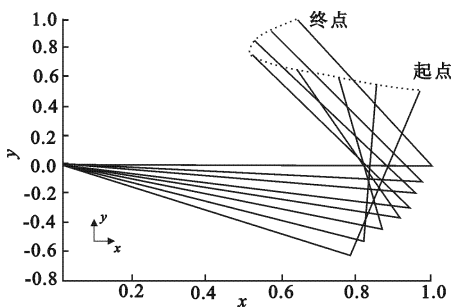


图 2 两杆平面机械手运动轨迹

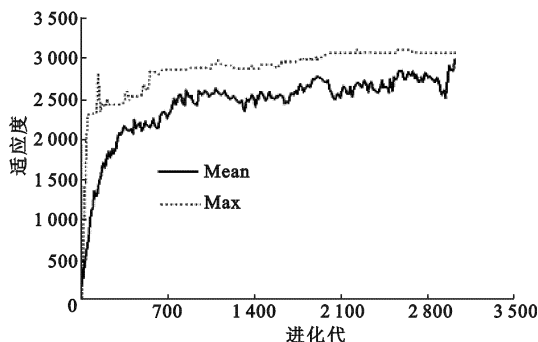


图 3 最大适应度及平均适应度随时间变化曲线

采用 GA 的方法构建 NN 模型参数, 对该问题取输入层为两个节点, 两个节点分别接收机械手的位置  $p_x, p_y$  的值, 输出节点为  $\theta_1$  和  $\theta_2$ , 即输出为  $[-\pi, \pi]$  连续取值的机械手连杆转角, 并取  $N = 50, p_c = 0.92, p_m = 0.03$ , 在 GA 进化中采用保留最优策略, 经过近 1500 代的学习, 得到了较理想的最优编码串个体, 将此最优解转化成对应的 NN, 其隐层仅需 4 个神经元节点, 共有 12 条连接边. 图 3 给出了每代中的最大适应度  $F(i)$  及平均适应度值  $\bar{F}$  随迭代次数  $T$  变化的曲线图. 为了进一步分析采用该方法构建的网络同 BP 作性能对比分析, 仅取  $\theta_1$  的初值及网络学习获得的输出模拟值作对比, 并取  $\theta_1$  初始值为  $-7\pi/32, -6\pi/32, \dots, 0$  共 8 个值进行分析, 其模拟值及各种误差数据见表 1.

为了同传统 BP 算法相对比, 现采用三层前向神经网络建立机械手逆运动学模型. 依经验构建 NN 结构为  $N_{2,5,2}$ , 即输入、隐层及输出节点数分别为 2, 5, 2, 隐层节点激励函数为式(3), 输出层激励函数为式(4). 依文献 [7] 的方法将传统 BP 算法加以改进, 即在输入层及隐层各增加 1 个偏置节点, 其目的是提高 BP 网络的收敛性, 故网络的连接权数为  $30(3 \times 6 + 6 \times 2 = 30)$ , 假定网络权值分布在区间  $[-1, 1]$  内, 其具体结果参见表 1 所示.

为进一步将两种方法的仿真结果与真实值作对比, 采用 Matlab 中的 Simulink 工具箱进行仿真, 对比结果如图 4 所示.

对比分析: 采用 GA 方法自动学习 NN 结构、权值及阈值仅需 8 个神经元, 而且节点间无需全连接, 共需 12 条连接权. 而依据经验且采用 BP 算法需要 11 个神经元 30 条连接边, 且需要全连接才能获得基本稳定的网络结构, 而连接权个数的增加无疑在网络使用时需计算比较的次数增加, 从而占用了更多的计算比较时间. 从对同一个问题的映射精度上对比, 图 4 反映出采用 GA 方法构建的神经网络在节点映射接近程度上明显高于采用 BP 训练的网络, 从表 1 结果对比分析也可看出, 采用 GA 构建的 NN 在映射数据的残差平方和及平均相对

表 1 两种方法的仿真结果误差比较

样本序号 k	原始数据 $\theta_1^{(0)}(k)$	模拟数据 $\hat{\theta}_1^{(0)}(k)$		残差 $\epsilon(k) = \theta_1^{(0)}(k) - \hat{\theta}_1^{(0)}(k)$		残差平方和 $s = \sum_{k=1}^8 \epsilon^2(k)$		平均相对误差 $\Delta = \frac{1}{8} \sum_{k=1}^8 \frac{ \epsilon(k) }{\theta_1^{(0)}(k)}$	
		BP	GA 构建	BP	GA 构建	BP	GA 构建	BP	GA 构建
		方法	NN 方法	方法	NN 方法	方法	NN 方法	方法	NN 方法
1	-0.6872	-0.6792	-0.6901	-0.0080	0.0029	0.0010	0.00013	6.1%	3.7%
2	-0.5890	-0.5948	-0.5915	0.0058	0.0025				
3	-0.4909	-0.4775	-0.4905	-0.0134	-0.0004				
4	-0.3927	-0.4012	-0.3892	0.0085	-0.0035				
5	-0.2945	-0.2807	-0.2949	-0.0138	0.0004				
6	-0.1964	-0.2095	-0.1901	0.0132	-0.0062				
7	-0.0982	-0.0801	-0.0901	-0.0181	-0.0081				
8	-0.0100	-0.0112	-0.0117	0.0012	0.0017				

误差明显低于 BP 方法,这也说明其对问题映射求解的精度显著高于 BP 方式.当然采用 GA 自学习构建 NN 方式也存在不足之处,即构建的网络只能是前馈网络不能构建反馈网络,否则该分区编码无法实现.

### 4 结 论

(1)从构建网络及学习过程来说,本文中方法既能构建网络结构,又能起到网络学习的作用,因此它比传统 BP 采用误差反传在学习速度上显著提高,且极易摆脱局部极小的困扰;(2)采用 GA 自学习构建 NN 结构的方法能大大减少 NN 的连接权及隐层节点数,从而使构建的 NN 在工作时运算效率明显提高,而且网络结构自适应能力强;(3)通过 GA 自学习构建的 NN 结构在求解相同问题时明显比传统 BP 学习的网络在求解问题的精度上显著提高.

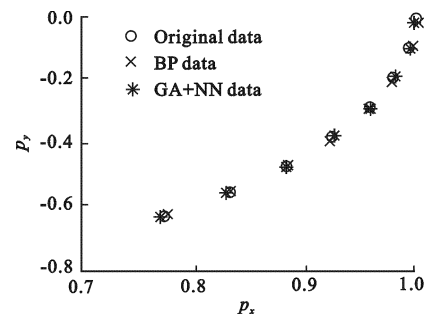


图 4 仿真结果对比图

### 参考文献:

[1] Castillo P A, Merelo J J, Arenas M G, et al. Comparing Evolutionary Hybrid Systems for Design and Optimization of Multilayer Perceptron Structure Along Training Parameters [J]. Information Sciences, 2007, 177(14,15): 2884-2905.

[2] TAO Jili, WANG Ning. Splicing System Based Genetic Algorithms for Developing RBF Networks Models [J]. Chinese Journal of Chemical Engineering, 2007,15(2):240-246.

[3] Rafael M, Abdellah E F. Multilayer Neural Networks: an Experimental Evaluation of On-line Training Methods [J]. Computers & Operations Research, 2004, 31(9): 1491-1513.

[4] 胡方明, 简琴, 张秀君. 基于 BP 神经网络的车型分类器[J]. 西安电子科技大学学报, 2005, 32(3): 439-442. Hu Fangming, Jian Qin, Zhang Xiujun. The Classifier of Car Types Using BP Neural Networks[J]. Journal of Xidian University, 2005, 32(3): 439-442.

[5] 苏怀智, 吴中如, 温志萍, 等. 遗传算法在大坝安全监控神经网络预报模型建立中的应用[J]. 水利学报, 2001, 38(8): 44-48. Su Huaizhi, Wu Zhongru, Wen Zhiping, et al. The Application of Genetic Algorithm in Establishment of Neural Network Forecast Model for Dam Safety Monitoring[J]. Journal of Hydraulic Engineering, 2001, 38(8): 44-48.

[6] 杨淑媛, 刘芳, 焦李成. 一种基于量子染色体的遗传算法[J]. 西安电子科技大学学报, 2004, 31(1): 76-81. Yang Shuyuan, Liu Fang, Jiao Licheng. A Novel Genetic Algorithm Based on the Quantum Chromosome[J]. Journal of Xidian University, 2004, 31(1): 76-81.

[7] 杨国军, 崔平远, 李琳琳. 遗传算法在神经网络控制中的应用与实现[J]. 系统仿真学报, 2001,13(5): 567-570. Yang Guojun, Cui Pingyuan, Li Linlin. Applying and Realizing of enetic Algorithm in Neural Networks Control[J]. Journal of System Simulation, 2001,13(5): 567-570.