

# 基于遗忘进化规划的 Hopfield 网学习算法\*

孟祥武<sup>1,2</sup> 程虎<sup>2</sup>

<sup>1</sup>(北京邮电大学计算机工程系 北京 100876)

<sup>2</sup>(中国科学院软件研究所 北京 100080)

**摘要** 本文提出了一个基于遗忘进化规划的 Hopfield 网学习算法,通过遗忘部分个体,算法能避免局部最小,给定不动点、极限环或迭代序列,通过解不等式,算法能同时获得 Hopfield 网的拓扑结构和权值.该算法克服了进化 Hopfield 网学习的局限性,它还能找到多个优化解.实验也证明了该算法的有效性.

**关键词** 进化规划,进化算法,Hopfield 网,遗忘,学习算法.

**中图法分类号** TP18

人类的智能是在漫长的进化过程中发展起来的,生物体通过进化逐步提高了对动态环境的适应能力,许多困难问题,人们都难以找出好的算法来解决,但自然界生物体通过自身的进化就能使问题得到完美的解决.

人们开始逐步认识到进化的特征,当遇到困难时,希望能从大自然中找到启发答案,这就逐步形成了进化算法.

近年来出现的这种随机搜索方法——进化算法,主要包括3种典型算法<sup>[1]</sup>,遗传算法 GA(genetic algorithms)、进化规划 EP(evolutionary programming)和进化策略 ES(evolution strategies),这3种典型算法有相同之处,也有很大的不同之处,其中 GA 的理论较为成熟,应用得也最为广泛,但已有研究表明,对于许多优化问题,例如 TSP 问题、模式识别和神经网络学习,EP 优于 GA.<sup>[2~4]</sup>EP 也被成功地应用到了许多领域,如模式识别、自动控制、下棋和神经网络学习方面.

目前 GA 和连接机制的理论都表明:GA 不适合进化神经网络学习,EP 是进化算法的一种,它较为适合进化神经网络学习.<sup>[5]</sup>

1993 年,McDonnell 和 Waagen 用 EP 同时进化前向神经网络的连接和权值.<sup>[3]</sup>Fogel 用 EP 进化一个隐结点可变的、三层全连接前向感知器.<sup>[6]</sup>1994 年,Angeline 等人用 EP 同时获得反馈神经网络的结构和权值.<sup>[4]</sup>1995 年,Yao 等人用 EP 优化小波神经网络.<sup>[7]</sup>

EP 也有缺点,主要就是局部最小问题,本文提出引入遗忘机制,主要就是克服这个缺点,使 EP 的搜索寻优过程能从局部最小中跳出来,继续寻找全局最优解,并提出了一个基于遗忘进化规划的 Hopfield 网学习算法.

## 1 Hopfield 网学习

Hopfield 网有  $n$  个神经元,权矩阵  $W$  为  $n \times n$  阶实数矩阵,每个神经元的状态为 1 或 -1,称  $n$  维矢量训练模式对被存入 Hopfield 网中,若满足

$$Y = \text{sgn}(WX - \theta), X \in \{-1, 1\}^n, Y \in \{-1, 1\}^n, W \in R^{n \times n}, \theta \in R^n,$$

$$\text{sgn} = \begin{cases} 1 & \text{当 } x \geq 0; \\ -1 & \text{当 } x < 0 \end{cases}$$

在学习过程中,学习算法的目的是确定一组权值  $W_{ij}$ ,使得  $m$  个待记忆的二值样本  $[X(u), Y(u)]$  存入 Hopfield 网中,  $u=1, 2, \dots, m$ ;  $W_{ij}$  表示神经元  $i$  与神经元  $j$  之间的连接权,  $i, j=1, 2, \dots, n$ , 每个样本为一对  $n$  维矢量.

$$[X(u), Y(u)] = [X_1(u), X_2(u), \dots, X_n(u), Y_1(u), Y_2(u), \dots, Y_n(u)]; X_i(u) \in \{-1, 1\}, Y_i(u) \in \{-1, 1\}.$$

这里,令阈值  $\theta=0$ ,则  $Y(u) = \text{sgn}[WX(u)]$ ,即

$$Y_i(u) = \text{sgn} \left[ \sum_{j=1}^n W_{ij} X_j(u) \right]$$

\* 作者孟祥武,1966年生,博士,讲师,主要研究领域为进化算法,神经网络和人工智能.程虎,1938年生,研究员,博士生导师,主要研究领域为语言编译,软件工程,神经网络和人工智能.

本文通讯联系人:孟祥武,北京 100876,北京邮电大学计算机工程系 135 信箱

本文 1996-12-10 收到原稿,1997-04-11 收到修改稿

$$Y_2(u) = \text{sgn}[\sum_{j=1}^n W_{2j} X_j(u)]$$

$$\dots$$

$$Y_n(u) = \text{sgn}[\sum_{j=1}^n W_{nj} X_j(u)]$$

若  $Y_i(u) = 1$ , 则  $\sum_{j=1}^n W_{ij} X_j(u) \geq 0$ ; 若  $Y_i(u) = -1$ , 则  $\sum_{j=1}^n W_{ij} X_j(u) < 0$ , 故 Hopfield 网的学习问题可转化为求解  $n$  组线性不等式组

$$X_1(u)W_{i1} + X_2(u)W_{i2} + \dots + X_n(u)W_{in} \geq 0, (\text{或} < 0, \text{根据 } Y_i(u) \text{ 的值})$$

其阈值  $\theta$  均为 0. 其中每组中含有  $m$  个不等式.

对于任意第  $i$  组不等式,  $i = 1, 2, \dots, n$ ; 因有  $m$  个样本, 故要同时满足下列  $m$  个不等式.

$$\begin{cases} X_1(1)W_{i1} + X_2(1)W_{i2} + \dots + X_n(1)W_{in} \geq 0, (\text{或} < 0, \text{根据 } Y_i(1) \text{ 的值}) \\ X_1(2)W_{i1} + X_2(2)W_{i2} + \dots + X_n(2)W_{in} \geq 0, (\text{或} < 0, \text{根据 } Y_i(2) \text{ 的值}) \\ \dots \\ X_1(m)W_{i1} + X_2(m)W_{i2} + \dots + X_n(m)W_{in} \geq 0, (\text{或} < 0, \text{根据 } Y_i(m) \text{ 的值}) \end{cases}$$

其阈值  $\theta$  均为 0. 求解出上述不等式组, 就确定了一组权值  $W_{ij}, i, j = 1, 2, \dots, n$ .

给定训练集, 就可得到权矩阵应满足的不等式组, 这时网络学习就变成了了解关于网络权值的线性不等式组, 即 Hopfield 网的学习变成了解线性不等式组.

当输入为  $n$  维矢量时, 该 Hopfield 网中就有  $n$  个结点, 与每个结点相连接点之间的权值对应一组线性不等式组的解. 解该线性不等式组, 就可获得该结点的权值. 一个结点一个结点地求解, 最后就获得整个网络的所有权值. 就这样, 通过解不等式组, 确定了 Hopfield 网的拓扑结构和权值.

## 2 基于遗忘进化规划的 Hopfield 网学习算法

应用遗忘进化规划进行 Hopfield 网的进化学习, 给定训练集, 通过求解不等式组, 可以获得权值. 下面是相应的算法.

- (1) 初始化群体  $P_i, i = 1, 2, \dots, k$ .  $P_i$  的每一分量均按等可能概率在某一区间上选择.
- (2) 按照适应度函数评价每个个体, 即计算适应值  $\theta_i$  的值. 适应度函数为不等式组中已满足不等式的个数.
- (3) 对每个父辈个体  $P_i, i = 1, 2, \dots, k$ ; 经高斯变异产生一子代个体.

$$P_{i+k,j} = P_{i,j} + \Phi_{i,j} N(0, 1), \quad j = 1, 2, \dots, n$$

$$\Phi_{i,j} = \lambda_j (\theta_i / \Theta)$$

这里  $P_{i,j}$  表示第  $i$  个个体的第  $j$  个分量;  $N(0, 1)$  表示期望值为 0, 标准偏差为 1 的正态分布一维随机变量;  $\Theta$  为所有个体的适应值之和;  $\lambda_j$  为进化率,  $0 \leq \lambda_j \leq 1$ .

- (4) 按照适应度函数评价子代个体, 即计算适应值  $\theta_{i+k}$  的值.
- (5) 对每个个体  $P_i, i = 1, 2, \dots, 2k$ , 从群体中随机选择  $C$  个个体, 按适应值,  $P_i$  与选中的每个个体比较, 得出比  $P_i$  差的个体数目, 此数目称为  $P_i$  的得分.
- (6) 选择  $k$  个得分最高的个体成为下一代.
- (7) 判断停止条件是否已满足. 若已满足, 停止.
- (8) 判断连续  $G$  代适应值变化率. 若变化率大 (即大于  $\epsilon, \epsilon \geq 0$ ), 继续迭代, 转(3).
- (9) 若适应值变化率小 (即小于或等于  $\epsilon$ ), 遗忘群体中的  $f$  个个体, 即初始化  $f$  个个体, 也就是所选个体的每一分量均按等可能概率在某一区间上选择, 即将在该区间上产生的一随机数赋予每一分量, 转(2).

适应度函数定义为不等式组中已满足不等式的个数, 假如  $m$  个不等式需要满足, 当适应度函数值为  $m$  时, 算法停止, 此时已获得所要权值.

本文的选择方法属于联赛选择法 (Tournament Selection), 其时间复杂度为  $O(n)$ , 轮盘赌 (Roulette Wheel) 的时间复杂度为  $O(n^2)$ . 联赛选择法特别容易并行实现, 其它方法难以并行实现, 因为它们需要一些全局信息.

轮盘赌在进化早期, 最好一类个体数目的增长率可能很高, 后期的增长率可能很低, 从而降低搜索效率, 相对而言, 易陷入局部最小. 当选择大的联赛规模 (即竞争数) 时, 联赛选择法能达到高的增长率.<sup>[8]</sup>

群体规模等参数的选择一般是根据经验或实验确定, 如一般群体大小为 10~160 之间. 较大的群体规模可减少算法陷入局部最小的机会, 但较大的群体规模意味着计算复杂度高.

遗忘策略是:有选择性地遗忘,选择适应值低的个体,然后将它的每一分量均赋予一个在某一区间上产生的随机数.若还连续出现局部最小,遗忘整个群体,即把群体中的每一分量,都重新初始化,相当于重新进化.

这与心理学研究结果是一致的.实验研究表明,遗忘是具有选择性的.材料的不同内容,不同部分,遗忘的发展是不同的.<sup>[9]</sup>

通过该算法获得的连接权矩阵是非对称的,即为一非对称网络,这种采用权值的非对称连接,有利于表达信息成分间的不等同关系,具有一定的优越性.<sup>[10]</sup>

若连接矩阵是对称的,要精确地保证  $W_{ij}=W_{ji}$ ,硬件实现几乎是不可能的;实际上,物理实现的神经网络是不对称的.<sup>[11]</sup>

### 3 计算机试验结果

在对 Hopfield 网进行进化学习时,曾遇到局部最小问题.例如利用 Hopfield 网进行数字字符识别,字符大小为  $10 \times 10$  点阵.用二值矢量表示每个字符,  $b_i$  表示  $10 \times 10$  点阵中第  $i$  位的值.

$$b_i = \begin{cases} 1 & \text{若第 } i \text{ 位为黑点,} \\ -1 & \text{若第 } i \text{ 位为空白,} \end{cases} \quad (0 \leq i \leq 99).$$

每个字符用 100 维的二值矢量表示.下面的实验是在 486/33 微机上的,用 C 语言编程实现的.

(1) 10 个数字字符识别.此时 Hopfield 网中有 100 个神经元,对全连接网络,共有  $100^2$  个权值.为计算每个结点的权值,要解一个具有 100 个变元、每组 10 个不等式的不等式组,该过程要重复 100 次.群体中每个个体为一个 100 维的矢量,矢量中的每一分量均为实数.图 1 是重复过程中进化最长的一次实验数据.

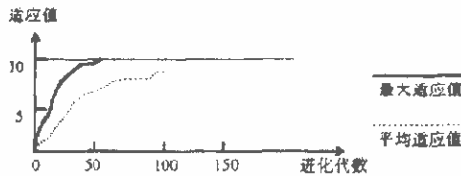


图 1 识别数字字符的进化过程

实验参数为:群体中个体数目  $k=10$ ,竞争数  $C=4$ ,进化率  $\lambda=1$ ,间隔代数  $G=10$ ,遗忘个体数目  $f=3$ , $\epsilon=0$ .算法的停止条件为:最大适应值为 10,即群体中出现适应值为 10 的个体,或进化代数超过 500 代.

(2) 用极限环来表示概念.每个数字有不同的两种形式,即这两种形式的同一个数字字符构成一个长度为 2 的极限环.将不同字体串联成一极限环,系统可同时识别多种字体.该网络中有 100 个神经元,对全连接网络,共有  $100^2$  个权值.为计算每个结点的权值,要解一个具有 100 个变元、每组 20 个不等式的不等式组,该过程要重复 100 次,即计算 100 个结点的权值.在计算过程中,会出现局部最小问题,适应度函数取已满足不等式的个数,此时在迭代进化过程中,群体中个体的最大适应值变为 17,18 或 19 时,就不再改变了.只有最大适应值为 20 时,即全部不等式已满足,解才是该结点的权值,采用遗忘进化规划,可以从局部最小中跳出来,继续迭代,重新寻找全局最优,找到一组解(这里最大适应值为 20)或多组解.

实验参数为:群体中个体数目  $k=10$ ,竞争数  $C=4$ ,进化率  $\lambda=0.1$ ,间隔代数  $G=20$ ,遗忘个体数目  $f=3$ , $\epsilon=0$ .算法的停止条件为:最大适应值为 20,即群体中出现适应值为 20 的个体,或进化代数超过 500 代.图 2、3 是出现局部最小情况时的实验数据.

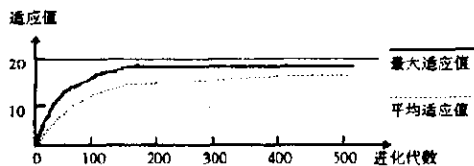


图 2 基于进化规划的 Hopfield 网学习算法

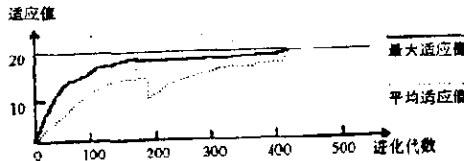


图3 基于遗忘进化规划的 Hopfield 网学习算法

#### 4 讨论和结论

本文所提算法主要有以下几个优点:

(1) 传统 Hopfield 网学习算法是基于外积规则的<sup>[12]</sup>,许多权矩阵不能用外积规则构造得到.也就是说,有时尽管存在权矩阵,用传统 Hopfield 网学习算法是无法得到的.给定不动点、极限环或迭代序列,构造 Hopfield 网也是非常困难的.<sup>[13]</sup>本文所提算法,给定不动点、极限环或迭代序列,只要存在权矩阵,就可得到;即可构造出相应的 Hopfield 网.克服了上述缺点.

(2) 引入遗忘机制,当陷入局部最小时,搜索寻优过程可从局部最小中跳出来,继续寻找全局最优解.解决了神经网络进化学习过程中的局部最小问题.遗忘进化规划还可以应用到其它搜索寻优过程中.

(3) 能够得到多组解,这给用户提供了多种选择.对于一类特定问题,有时某些约束条件未体现或无法体现在适应度函数中,此时可再考虑这些约束条件,从多组解中选择.

(4) 用极限环表示概念,这很适合于字符识别.一个字符有不同的字体,一个字的不同字体可用极限环表示.如汉字有宋体、楷体等.

(5) 适用于变化的环境,即不动点、极限环或迭代序列是常变化的.当增加几个样本时,算法一般不需要重新计算,继续进化就能发现新解.当几个样本被删除了,算法一般也不需计算,进化过程中的中间解就变成了新解,也就是说,有时可从进化过程的中间解中选择新解.

(6) 算法具有良好的并行性,进化过程中的群体是一个可行解的集合.适合于并行计算.

本文提出了基于遗忘进化规划的 Hopfield 网学习算法,能同时获得 Hopfield 网的拓扑结构和权值.引入遗忘机制,可以解决进化规划的局部最小问题,该算法简单、通用,易于实现,实验也证明了该算法的有效性.

#### 参考文献

- Schwefel H P. On the evolution of evolutionary computation. In: Zurada J M, Mark II R J, Robinson C J eds. Computational Intelligence, Imitating Life. New York: IEEE Press, 1994. 116~124
- Fogel L J. Evolutionary programming in perspective: the top-down view. In: Zurada J M, Mark II R J, Robinson C J eds., Computational Intelligence, Imitating Life. New York: IEEE Press, 1994. 135~146
- Fogel D B, Stayton L C. On the effectiveness of crossover in simulated evolutionary optimization. *BioSystems*, 1994, 32: 171~182
- Angeline P J, Saunders G M, Pollack J B. An evolutionary algorithm that constructs recurrent neural networks. *IEEE Transactions on Neural Networks*, 1994, 5(1): 54~65
- McDonnell J R, Waagen D. Evolving neural network connectivity. *IEEE International Conference on Neural Networks*, Vol 2. San Francisco, California, 1993. 863~868
- Fogel D B. Using Evolutionary programming to create neural networks that are capable of playing Tic-Tac-Toe. In: Proceedings of IEEE International Conference on Neural Networks, Vol 2. San Francisco, California, 1993. 875~880
- Yao Susu *et al.* Optimization of wavelet neural network using evolutionary programming. In: Proceedings of the International Conference on Neural Information Processing. Beijing, China, 1995. 313~316
- Goldberg D E, Deb K. A comparative analysis of selection schemes used in genetic algorithms. In: Rawlins G J E ed. Foundations of Genetic Algorithms. Morgan Kaufmann Publishers, Inc., 1991. 69~93
- Li Zheng, Zhang Lü-xiang. General psychology. Hefei: Press of University of Science and Technology of China, 1995
- Zhang Cheng-fu *et al.* The analysis of some associative memory neural network models. *Pattern Recognition and Artificial Intelligence*, 1990, 3(1): 14~21
- Xu Z B *et al.* Asymmetric hopfield-type networks: theory and applications. *Neural Networks*, 1996, 9(3): 483~501
- Hopfield J J. Neural networks and physical systems with emergent collective computational abilities. In: Proceedings of the National Academy of Science, Vol 79. USA, 1982. 2554~2558
- Buhmann J, Schulten K. Noise-driven temporal association in neural networks. *Europhysics Letters*, 1987, 4: 1205~1209

## A Learning Algorithm of Hopfield Neural Network Based on Evolutionary Programming with Forgetting

MENG Xiang-wu<sup>1,2</sup> CHENG Hu<sup>2</sup>

<sup>1</sup>(Department of Computer Engineering Beijing University of Posts and Telecommunications Beijing 100876)

<sup>2</sup>(Institute of Software The Chinese Academy of Sciences Beijing 100080)

**Abstract** This paper presents a learning algorithm of Hopfield neural network based on evolutionary programming with forgetting. The algorithm can avoid local minima by forgetting some individuals. Under constraints of fixed points, limit cycles or iteration sequences, the algorithm simultaneously acquires both the topology and weights for Hopfield neural network by solving inequalities. It copes with the limitations of evolving Hopfield learning algorithm. It can also find several optimal solutions. The experimental results also demonstrate the effectiveness of the algorithm.

**Key words** Evolutionary programming, evolutionary algorithm, Hopfield neural network, forgetting, learning algorithm.

**Class number** TP18