

# 用于前馈神经网络的遗传设计<sup>\*</sup>

陆建峰<sup>\*\*</sup> 上 上 杨静宇

(南京理工大学信息学院, 南京 210094)

**摘要** 在人工神经网络应用中,由于存在网络规模和拓扑结构难以预先确定,网络学习速度慢,且易于收敛到局部最优等等问题,有关文献提出了采用基于遗传算法(GAs)思想进行设计和学习的方法,该方法能够同时确定网络的结构及有关参数。该文在此基础上,对此方法进行了改进,改进之处在于,采用浮点数矩阵来表示编码,同时对于遗传算法的进化过程也进行了一定的改进,使该方法能够接受一定的约束条件。针对前馈型神经网络,该方法在满足一定约束条件的情况下,能同时有效地寻找到合适的网络结构和相应的参数(神经网络的权值和阈值),新方法较原方法在精度和速度上都有较大的提高。

**关键词** 神经网络,网络拓扑学,算法;遗传算法

**分类号** TP 14

人工神经网络在很多方面得到了广泛的应用,但是在神经网络的使用中,却不得不面对如何预先确定网络拓扑结构的问题。由于神经网络潜在和固有的并行性、非线性及不确定性,很难用数学模型对其进行精确的分析。

Saha 提出了一种可以同时寻找神经网络结构和参数的方法<sup>[1]</sup>,该方法力求使神经网络中经验型的设计工作算法化,且这种方法与具体应用问题无关,具有广泛的推广价值。在这种方法中,使用了遗传算法来搜索合适的网络结构以及相应的权值和阈值,此时神经网络学习的含义已经扩展为权(阈)值学习(即传统意义上的神经网络训练)和拓扑结构的学习(即传统意义上的神经网络设计)。但是,该文中采用二进制串的方法来表示有关的参数,这样会影响参数的精度,从而直接影响神经网络的精度和速度,本文中采用了浮点数矩阵的形式来表示参数,可以很好地克服上述缺陷。此外,本文的算法还可以接受一定的约束条件,使该方法具有更广的适用性。关于遗传算法方面的问题可以参见文献[2]。

## 1 用于神经网络设计的遗传算法

人工神经网络的设计和学习实质上就是一个带约束条件的多目标优化问题。网络的误

\* 收稿日期:1998-09-03

\* 南京理工大学科研发展基金资助项目

\*\* 陆建峰 男 30岁 讲师

差和复杂度可以用来构造适应度函数,而一组以编码形式出现的网络结构和相关参数,则被认为是神经网络学习问题的一个可能解,在遗传算法中则可视作一条染色体(chromosome)。从算法时间复杂度考虑,遗传算法中绝大多数处理都集中在适应度函数的计算上。与 BP 算法相比,遗传算法只进行简单的前馈计算,而没有进行牵涉到需要大量求导运算的误差反向传播操作,算法复杂度大为降低。

## 2 神经网络的编码表示

神经网络的编码方法对于有效利用遗传算法进行网络设计和学习有着非常重要的作用。本文采用了浮点数矩阵编码格式表示网络的结构和参数<sup>[3]</sup>。该编码方式的优点在于可以精确地描述网络,包括全连接和稀疏型神经网络;这种方法的缺点是,当处理比较复杂的问题时,编码矩阵的规模会很快增大(与神经元个数的平方成正比)。

参数及变量说明:

$NN$ 为网络允许的最大神经元数	[约束条件 1]
$NC$ 为每个神经元的最大输入边个数	[约束条件 2]
$ERROR$ 为网络的期望误差	[优化标准 1]
$HOPE.LINK$ 为网络内部的期望连接边数	[优化标准 2]
$INPUT.NUM$ 为网络的输入神经元个数	[由应用问题确定]
$OUTPUT.NUM$ 为网络的输出神经元个数	[由应用问题确定]

$weight[NN][NN]$  为网络连接矩阵,含义如下:

$weight[i][j] = X \quad (i > j)$	表示一条由结点 $j$ 传向结点 $i$ 的信息通路,权值为 $X$ ;
$weight[i][j] = Y \quad (i = j)$	表示网络中的结点 $i$ 的阈值为 $Y$ ;
$weight[i][j] = 0 \quad (i < j)$	表示信息的单向前馈传递性;
$weight[i][j] = 0 \quad (i > j)$	表示结点 $j$ 和结点 $i$ 之间无直接信息通路。

其中,  $i, j \in [0, NN - 1], X \geq 0$ 。

对于该编码方式,此处给出几点说明:(1) 信号从输入传向输出,即前馈型网络,且不存在自反馈;(2) 一个矩阵代表一个网络所有的特征(拓扑结构,参数);(3) 最大结点数、最大输入边数分别设定为  $NN$  和  $NC$ ;(4) 所有结点依次被分配一个从 0 到  $NN - 1$  的序列号;(5) 网络中前  $INPUT.NUM$  个神经元固定作为输入结点,最后  $OUTPUT.NUM$  个神经元固定作为输出结点;(6) 除输入结点,与序列号小的结点均不相连的结点(无输入结点)将被删除,除输出结点,与序列号大的结点均不相连的结点(无输出结点)将被删除;(7) 输入边数大于  $NC$  的结点将被随机删除掉多余连接;(8) 连接边的删除:置编码矩阵中的相应位为零;结点的删除:置编码矩阵中所有与该结点相关的位为零,例:(a) 要删除从结点  $j$  到结点  $i$  的有向边,则令  $weight[i][j] = 0, i > j$ ;(b) 要删除结点  $i$ ,则令  $weight[i][j] = 0, j \in [0, i], weight[j][i] = 0, j \in [i, NN - 1]$ ;(9) 网络中神经元之间没有明显的层次关系,除输入结点之间以外,神经元的连接不受限制,结点的序列号仅用来区分有向边的首尾。

### 3 算法描述

首先,定义神经元的输出为

$$\text{neural}[i] = F(\text{weight}[i][j] \cdot \text{neural}[j] - \text{weight}[i][i]) \quad j \in [0, i-1] \quad (1)$$

式中,  $\text{neural}[i]$  表示序列号为  $i$  的神经元的输出,  $i \in [INPUT\_NUM, NN - 1]$ 。

由于本文规定作为输入的神经元间不能存在连接,故对于  $i \in [0, INPUT\_NUM - 1]$  神经元的输出等于输入。

在笔者的实验中,神经元的激活函数  $f(x)$  定义为,  $f(x) = (\exp(x) - \exp(-x)) / (\exp(x) + \exp(-x))$ , 与常用的 Sigmoid 相比, 有更明显的非线性特征。

网络的误差函数(能量函数)定义为  $E = 0.5 \times \sum_{k=0}^{m-1} \sum_{i=0}^{q-1} (Y(k, i) - C(k, i))^2$ , 式中,  $m$  = 训练集中的样本数;  $q$  = 输出单元的个数;  $Y(k, i)$  = 对应第  $k$  个学习样本, 第  $i$  个输出神经元的实际输出;  $C(k, i)$  = 对应第  $k$  个学习样本, 第  $i$  个输出神经元的期望输出。个体的适应值定义为

$$\text{value}[i] = \begin{cases} T(E(i)) & \text{if } E(i) > ERROR \\ T(ERROR) * D(LINK(i) - HOPE\_LINK) & \text{if } E(i) \leq ERROR \end{cases} \quad (2)$$

式中,  $\text{value}[i]$  表示序列号为  $i$  的个体(染色体)的适应值;  $T(x)$  为基于网络误差的估价函数, 本文中设为  $T(x) = \sqrt{1/x}$ ;  $D(x)$  为基于网络复杂度的估价函数, 本文中设为  $D(x) = 1 + \exp(-x/\mu)$ , 和  $\mu$  为大于 0 的常量;  $LINK(i)$  为序列号为  $i$  的个体中包含的连接数;  $E(i)$  为序列号为  $i$  的个体对应的网络的均方误差。

在实现过程中没有使用对优化目标加权值的方法,而是采用下述分层次的优化方案。

(1) 网络误差大于  $ERROR$  时,染色体的适应度以基于误差的估价函数  $T(x)$  来衡量,染色体的进步表现为所对应的网络误差的降低;

(2) 网络的误差小于或等于  $ERROR$  时,染色体的适应度等于  $T(x) \cdot D(x)$ , 此时染色体的进步表现为网络内部连接数的减少,此时  $T(x)$  中的参数  $x = ERROR$ 。

算法步骤:(1) 初始化染色体矩阵;(2) 评估每个染色体的性能;(3) 用赌盘算法进行选择操作;(4) 用随机数方法产生杂交点;(5) 在杂交过程中保留最佳个体;(6) 用随机数产生变异位置,该位置为矩阵中的某一元素;(7) 间隔一定的代数进行灾变;(8) 是否有个体误差达到期望误差或演化次数已达指定的次数,若是,则结束,否则转(2)。

### 4 实验结果及结论

本文以 XOR 问题为例进行了训练,对于 XOR 问题共有 4 组样本。与一般的 XOR 问题不同,这里以 -1 和 +1 分别代表二值逻辑中的 0 和 1,这是与所选择的激励函数的特性相适应的 ( $f(x)$  的值域为  $[-1, +1]$ )。

遗传算法的参数设置如下。

种群规模:100;杂交率:0.8;变异率:0.006;变代间隔:20代;搜索空间:  $[-20, +20]$ 。

使用本文提出的算法,在不同的约束条件下,找到了具有不同结构的网络,即由遗传算

法优化产生的解决异或问题的神经网络,整个网络仅包含有 4 个神经元和 5 条连接边。

同样,一个解决相同规模的 XOR 问题的典型 BP 网(输入结点:2 个,中间层结点:3 个,输出层结点:1 个),却需要 6 个神经元和 9 条连接边。

算法约束条件:  $NN = 6, NC = 3, HOPE\_LINK = 5, ERROR = 1.0 \times 10^{-6}$ 。

在此约束条件下,在 PC486/80 运行环境下得到如下的结果。

算法运行时间: 5 s; 演化代数: 57 代; 实际误差:  $1.207\ 923 \times 10^{-13}$ 。

如果采用传统的 BP 算法,该 XOR 问题至少需要训练数千次,时间花费至少为数十秒。

从上述研究可以看出,虽然由遗传算法寻找到的网络结构并不一定最优,但却满足约束条件,从工程技术角度来讲是可行的。该方法的最大特点在于避开了必须预先确定网络结构的困难,而以编码的方式将网络的结构和相关参数有机地融合在一起,利用遗传算法特有的并行处理能力和全局优化能力完成网络的学习,为神经网络的设计和学习开辟了新的思路。目前的研究还只局限于前馈型网络和类似于异或问题这样比较小规模的问题,对于更加复杂的神经网络应用问题,还需要做进一步的努力,以提高算法的效率和适应能力。

## 参 考 文 献

- 1 Saha S, Chrisensen J P. Genetic design of sparse feedforward neural network. *Information Science*, 1994, 79: 191 ~ 200
- 2 刘勇,康立山,陈毓屏. 非数值并行算法——遗传算法. 北京:科学出版社,1995
- 3 金希东,李治. 遗传-灾变算法及其在神经网络和控制系统中的应用. 见:靳蕃. 神经网络理论与应用研究. 成都:西南交通大学出版社,1996. 255 ~ 260

## A Genetic Design for Feedforward Neural Network

Lu Jianfeng Shang Shang Yang Jingyu

(School of Information, NUST, Nanjing 210094)

**ABSTRACT** During application of neural network, there exist some problems, including difficult determination of the size and structure of neural network in advance, the learning speed of neural network is slow, and it's easy to converge to local optimum. In the view of these problems, some references proposed to use Genetic Algorithms (GAs) to design and train neural network, the top structure and related parameters (weights and thresholds) can be obtained simultaneously with this method. On this basis, this presentation made some improvement on proposed method. Main improvement includes that float-point matrix is adopted to encode, evolution of GAs itself is modified and the proposed method can satisfy some constrain conditions. For feedforward neural network, this method can find suitable network structure and corresponding parameter (weights and thresholds) simultaneously under certain constrain condition. New method has great improvement over the old one in both accuracy and speed.

**KEY WORDS** neural network, network topology, algorithm; genetic algorithm