

文章编号:1006-7329(2001)04-0104-06

基于结构式二进制编码的 遗传算法及其在BP网络中的应用

余 萌

(中国人民解放军总医院营房处,北京 100853)

摘要:根据BP网络的拓扑特征,本文设计了基于结构式二进制编码的遗传算法。在该算法中,通过先将庞大的解空间进行分解处理,再将分解后的子空间视为个体进行遗传操作,能借助遗传算法的优势在全局范围内搜索到最优解所在的子空间,从而为下一步应用BP算法进行局部搜索明确了起点、缩小了范围,有效解决了BP算法易陷入局部极小、收敛速度慢甚至不收敛等问题。最后,通过字母识别试验证明了该算法的效率。

关键词:遗传算法;神经网络;编码;最优化

中图分类号:TP183

文献标识码:A

基于 Darwin 的进化论和 Mendel 的群体遗传学,密彻根(Michigan)大学的 Holland 等创立了遗传算法(Genetic algorithm)^[1,2]。遗传算法最早用于模拟自然系统的自适应现象,后来则被引向广泛的工程问题。经过几十年的发展,遗传算法正以其解决不同非线性问题的鲁棒性、全局最优性、不依赖于问题模型的特性、本质并行性、内在学习性、高效率等优势成为一种自适应启发式、概率性迭代式的全局搜索算法,并引起了越来越多的研究和应用高潮。

目前,遗传算法的理论研究主要集中在以下几个方面^[3]:1)遗传算法的编码策略研究;2)遗传算法的全局收敛性和搜索效率分析;3)遗传算法的新结构研究;4)遗传算法的基因操作策略及其性能研究;5)遗传算法的参数选择;6)遗传算法与其它算法的综合及比较研究。编码是遗传算法应用中的首要问题,人们进行了大量研究,目前主要有二进制编码^[4]、浮点数编码^[5,6]以及用有序串编码^[7]等方案。二进制编码和浮点数编码各有优势,二进制编码是 Holland 模式(Schema)定理建议采取的方式,它易于用生物遗传理论来解释搜索过程并使得遗传操作很容易实现,但易产生 Hamming 悬崖(Hamming cliffs)使得搜索效率较低,此外在数值优化时还存在精度不高的缺点;浮点数编码是在解的表现型上直接进行遗传操作,有精度高、便于大空间搜索并能利用启发式信息等优点,但存在收敛速度慢,且对非线性强的数值优化问题性能不佳等缺点^[8];有序串编码主要用于组合优化问题。

针对 BP 网络拓扑结构的确定及其权值、阈值的学习问题,本文将引入一种新的编码方式即“结构式二进制编码”。该编码方式通过将染色体的数据结构很自然地表达为一个加权图而不是通常的串,充分利用了 BP 神经网络的拓扑特征,有利于网络结构的调整及遗传过程中知识的积累。另外,通过分解解空间和对分解后的子空间进行相应串长的二进制编码,能有效利用遗传算法全局优化及并行搜索的特点进行大范围寻优。

1 基于结构式二进制编码的遗传算法

1.1 简单遗传算法

• 收稿日期:2001-03-25

• 作者简介:余 萌(1970-),男,湖南岳阳人,中国人民解放军总医院少校,主要从事后勤组织指挥研究。

简单遗传算法可以形式化描述如下:

$$GA = (P(0), N, l, s, g, p, f, t) \tag{1}$$

这里 $P(0) = (a_1(0), a_2(0), \dots, a_N(0)) \in I^N$, 表示初始群体;

$I = B^l$, 表示位串空间, 对二进制编码有 $B = \{0, 1\}$;

N 表示群体中含有个体的个数;

l 表示位串的长度;

s 表示 $I^N \rightarrow I^N$, 表示选择策略;

g 表示遗传算子, 通常包括繁殖算子 $O_r: I \rightarrow I$ 、杂交算子 $O_c: I \times I \rightarrow I \times I$ 和变异算子 $O_m: I \rightarrow I$;

p 表示遗传算子的操作概率, 通常包括繁殖概率 p_r 、杂交概率 p_c 和变异概率 p_m ;

f 表示 $I \rightarrow R^+$ 是适应度函数;

t 表示 $I^N \rightarrow \{0, 1\}$ 是终止准则。

1.2 神经网络模型

神经网络可看成是从输入到输出的高度非线性映射, 即

$$F: R^n \rightarrow R^m, f(X) = Y$$

对于样本集合: 输入 $X^k (\in R^n)$ 和输出 $Y^k (\in R^m)$, 我们可以认为存在映射 g 使

$$g(X^k) = Y^k, k = 1, 2, \dots, P$$

成立。我们的目的便是要求出一映射 f , 使得在某种意义上(通常是最小二乘意义下), f 是 g 的最佳逼近。由于映射 f 是通过网络结构来实现的, 所以求解 f 实际上是确定网络的结构、求解网络的权值和阈值, 属于参数优化问题。

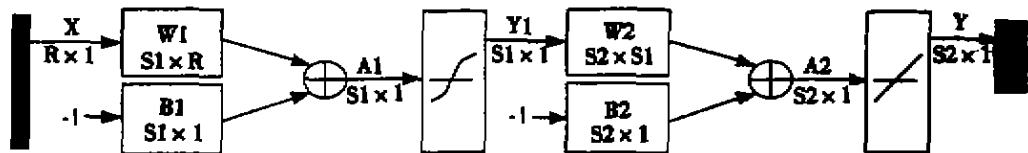
为实现 f 对 g 的最佳逼近, 需要利用输入输出样本集训练网络。以图 1 所示的单隐层 BP 网络为例, 其网络映射完全取决于如下参数

$$f: f(S_1, (W_1, B_1, W_2, B_2), (X^k, Y^k)) \tag{2}$$

这里 S_1 表示网络结构参数(隐层神经元的个数);

(W_1, B_1, W_2, B_2) 为网络的权值矩阵和阈值向量;

$(X^k, Y^k) (k=1, 2, \dots, P)$ 表示训练样本集;



R : 输入维数; S_1 : 第 1 层神经元数; S_2 : 第 2 层神经元素

图 1 具有一个隐层的 BP 网络

对网络的训练过程, 实质上就是对网络结构、权值、阈值进行学习、调整的过程。在网络训练中, 常使用 BP 算法或改进的 BP 算法如动量 BP 算法等。BP 算法是基于梯度下降法的思想, 但由于误差曲面的复杂, 常常具有多个局部极小点, 因此用 BP 算法训练网络时, 会遇到不收敛、训练后网络精度不够或训练速度慢等问题。导致 BP 算法存在上述问题的根本原因是由于神经网络中由权系数和阈值所构成的解空间过于庞大, 造成即使是小型应用问题, 其中的大部分空间也无法探测、搜索到。

实际上, 通过先将庞大的解空间进行分解处理, 再将分解后的子空间视为个体进行遗传操作, 将借助于遗传算法的优势在全局范围内搜索到最优解所在的子空间, 从而为下一步采用 BP 算法进行局部搜索明确起点和缩小范围, 解决 BP 算法现存的问题。

1.3 基于结构式二进制编码的遗传算法

在用遗传算法进行参数优化时, 一般是将各参数编码, 构成子串, 再将子串拼接起来构成“染色

体”串。但是不同的串长和不同的码制,对问题求解的精度和收敛时间会有很大影响。由于我们强调网络的可伸缩性以及演化过程中知识的积累,所以直接采用加权图来表示神经网络,并针对这一表示策略专门设计遗传操作。

1) 编码表示

BP 网络最自然的数据结构表示大概就是一个加权图,这个图的每一条边即是网络的一个连接,其上的权就是一个连接系数;每个顶点即是网络的一个结点,其上的权值即是结点的阈值。因此,我们这里的加权图不仅边上带权,其顶点也带权,从而与通常的加权图概念略有不同。鉴于遗传算法局部寻优的效果并不理想(不如 BP 算法),所以在编码时不宜采用实数制;即使是在采用二进制编码时,其码长也不需太长,以免影响收敛效率。

在二进制编码过程中,首先要确定二进制串的长度 l ,串长 l 依赖于变量的定义域及计算所需的精度。例如,变量 $w \in [-4, 4]$ 、要求精度 0.01 时,需将定义域 $[-4, 4]$ 分成至少 800 个等长小区间,而每个小区间用一个二进制串表示,由于 $[800]_{10} = [1100100000]_2$,故串长等于 10。其解码过程如下:

- 将二进制串 $(b_9 b_8 \dots b_0)$ 按下式转换为一个十进制整数

$$w' = \sum_{i=1}^9 b_i \cdot 2^i$$

- 按下式计算对应变量的值

$$w = -4 + w' \cdot \frac{4 - (-4)}{2^{10} - 1}$$

根据(2)式,可按照输入样本和输出样本的数据分布、权值矩阵和阈值向量的规模大小分别确立串长。

最后,可确定神经网络的结构式二进制编码形式如图 2:

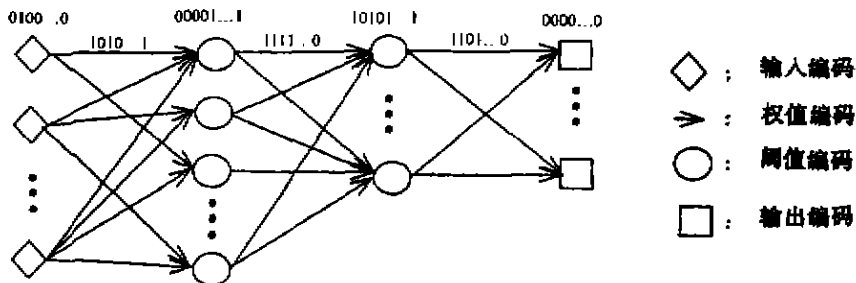


图 2 神经网络的结构式二进制编码

2) 选择策略

适应度函数用来区分群体中个体好坏的标准,是算法演化过程的驱动力,也是自然选择的唯一依据。在设计和训练 BP 网络时,可将网络的实际输出与期望输出之差的平方和,作为问题的原始适应度函数,即

$$E(net) = \frac{1}{2} \sum_{k=1}^P \sum_{i=1}^{S_2} (y_i^k - \hat{y}_i^k)^2$$

式中: P 表示训练集的规模;

S_2 表示输出样本的维数;

y_i^k, \hat{y}_i^k 表示第 k 组网络输出样本和期望输出样本的第 i 个分量。

上面定义的最小二乘误差函数可用来描述对给定数据集当前网络的性能。如果我们期望在具有相同性能的条件下网络的结构尽可能的简单,即使网络尽量具有最少的结点数和最少的连接,则我们可以在函数 $E(net)$ 的右端另外加上一个控制项而成如下形式

$$E(net) = \frac{1}{2} \sum_{k=1}^P \sum_{i=1}^{S_2} (y_i^k - \hat{y}_i^k)^2 + aN_{net}$$

式中: α 是一个控制参数,称为网络复杂性系数;

N_{net} 是网络参数的个数,通常包括网络的结点数与连接数。

为方便运用赌盘式选择策略,对原始适应度函数进行标准化,则

$$f(net, t) = \frac{1}{1 + E(net, t)}$$

上式即为模糊适应度函数。于是,由转盘式选择策略知,网络被选择到下一代作为遗传操作的父体的概率为

$$p(net, t) = \frac{f(net, t)}{\sum_{i=1}^N f(net, t)}$$

这里 N 为群体规模。

3) 遗传操作

由于我们使用的编码策略与简单遗传算法相比有很大不同,从而我们使用的遗传操作也有很大区别。

(1) 繁殖算子:为标准的转盘式繁殖算子。它包括两个步骤,首先根据选择策略及个体的选择概率从群体中选择父体,然后将父体不加改变地拷贝到当前群体。为了得到遗传过程中所发现的最佳网络,我们还使用择优策略(Elitist strategy),即保留找到的最佳个体并复制到当前群体中。

(2) 杂交算子:由于我们使用的编码表示使得在同一种群内的网络可能不同构,这样给杂交算子的设计带来了一定的难度。我们知道,杂交算子的作用即是交换两父体的部分基因以期获得具有更高适应度的后代。所以我们先使用转盘式选择策略在群体中选择两个父体,再随机产生一个具有足够大规模的杂交模板,根据模板决定两父体对应结构上的权值或阈值是否施行杂交;如果是施行杂交的话,再进行单点杂交。

(3) 变异算子:由于我们采用遗传算子非重叠的遗传算法结构,故变异算子与其它操作是独立进行的,变异算子也是首先按转盘式选择策略选择一个父体,然后等概率地进行下列三种操作之一

①删除隐含层的某些结点和(或)连接以及相应的连接权值,当删除的仅是某些连接时,只需将其对应的连接权二进制串全部置 0 即可;

②在隐含层插入一些结点和(或)连接并随机生成其相应的连接权二进制串;

③在父体中随机地选择一个结点或连接,然后将其权值按自适应变异策略进行变异,其中个体的温度定义为

$$T = 1 - \frac{1}{E(net) + 1}$$

2 模拟试验

在这里我们主要以大写英文字母的识别问题来测试我们的算法。假设英文字母由 7×5 个单色的象元构成(如图 3),每个象元按灰度进行 8 位二进制编码,则输入样本为 35 维的列向量;对应的输出是仅有对应位为 1、其它位为 0 的 24 维列向量。

Kolmogorov 已经证明了三层感知机(即含有两个隐含层的网络)能够解决可以表示成从某个输入空间到某个输出空间的连续映射的任意问题。所以在以下的试验中,我们限定网络最多仅含两个隐含层,并假定网络的输出皆为连续值。

2.1 参数设置

我们的算法有 8 个控制参数,它们是:群体规模 N 、杂交概率 p_c 、变异概率 p_m 、繁殖概率 p_r 、最大代数 G 、隐

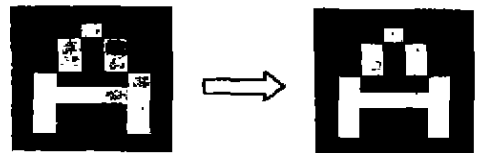


图 3 输入—输出样本对(字母 A)

含层内的最大结点数 m , 网络参数的定义区间 $[a, b]$ 及网络复杂性系数 c 。

在以后的试验中, 我们选取 $N=100, G=2000, p_r=0.1, p_c=0.6, p_m=0.3, [a, b]=[-4, 4]$ (编码长度定为 10)。 c 和 m 将根据试验选取。

2.2 试验结果

对英文大写字母识别问题, 我们进行了一系列试验以测试算法的效率, 结果如图 4。

该图上部分是运用遗传算法对网络进行训练的结果 ($c=0.002, m=20$), 下部分是在遗传算法训练结果之上再用动量 BP 算法进行局部寻优的结果。由上图可知, 网络经遗传算法进行搜索, 已经获得了良好的性能指标 ($E_{net}=0.068$) 和较好的结构参数 (含两个隐含层, 每层均含 10 个神经元); 在此基础上, 再利用动量 BP 算法局部搜索的优势, 可进一步优化权值矩阵和阈值向量, 获得更好的性能指标 ($E_{net}=0.005$)。为方便比较, 下面给出了随机生成初始权值及阈值后单独使用动量 BP 算法训练网络的一次尝试, 由图 4、5 可见, 网络在训练 5085 次后才能收敛到一个并不很优的指标。

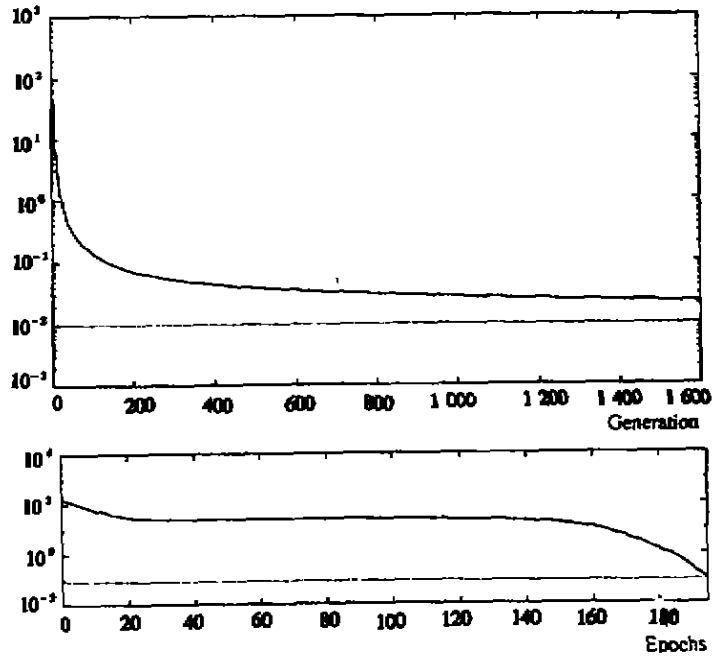


图 4 试验结果

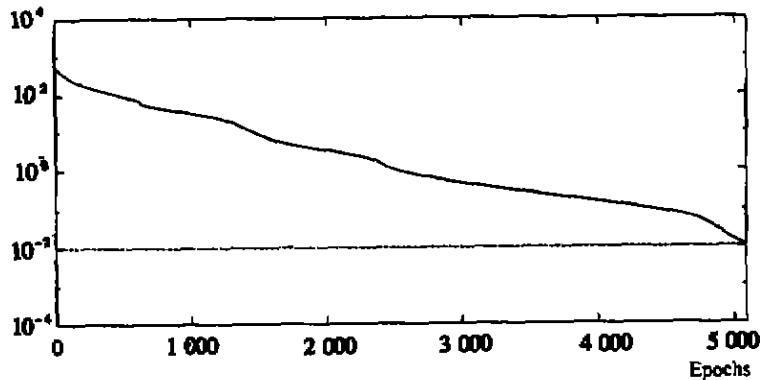


图 5 动量 BP 算法训练结果

3 结论

根据 BP 网络的拓扑特征, 本文设计了基于结构式二进制编码的遗传算法。在该算法中, 通过先将庞大的解空间进行分解处理, 再将分解后的子空间视为个体进行遗传操作, 能借助遗传算法的优势在全局范围内搜索到最优解所在的子空间, 从而为下一步采用 BP 算法进行局部搜索明确起点和缩小范围, 解决 BP 算法现存的问题。

试验表明所设计的算法是非常有效的, 它使得 BP 网络的设计及训练问题得到了较好解决, 特别是解决了使用 BP 算法训练网络时初始权值和阈值的确定问题, 在网络结构合适的情况下能完全避免 BP 算法陷入局部极小并加快误差的收敛速度。

参考文献:

- [1] Holland, J. H. . Concerning Efficient Adaptive Systems[J]. In Yovits, M. C. , Ends. , Self—Organizing Systems, 1962;215—230.
- [2] Bagley, J. D. . The Behavior of Adaptive Systems Which Employ Genetic and Correlation Algorithms[J]. Dissertation Abstracts International. 1967, 28(12).
- [3] 席裕庚,柴天佑,等. 遗传算法综述[J]. 控制理论与应用, 1996, 13(6):697—708.
- [4] Holland, J. H. . Adaptation in Natural And Artificial Systems[M], 1st ed. ,Cambridge, MA: MIT Press, 1992.
- [5] Goldberg, D. E. . Real—Coded Genetic Algorithm, Virtual Alphabets and Blocking [J]. Complex Systems, 1991, (5):139—167.
- [6] Wright, A. H. . Genetic Algorithms for Real Parameter Optimization. In Foundations of Genetic Algorithms, Rawlins[M], G. J. E. , Ed. San Mateo, CA: Morgan Kaufmann, 1991:205—218.
- [7] Fogel, D. B. . Applying Evolutionary Programming to Selected Traveling Salesman Problems[J]. Cybern and Sys, 1993, (24):27—36.
- [8] Michalewicz, Z. ,et. al. . Genetic Algorithms and Optimal Control Problems[A]. Proc. 29th. IEEE Conf Decision and Control[C], 1990:1664—1666.
- [9] 潘正君,康立山,等. 演化计算[M]. 北京:清华大学出版社,1998.

GA Based on Structural Formula Binary—coding and Its Applications in BP Networks

YU Meng

(Department of Barracks, China PLA General Hospital. Beijing 100853, China)

Abstract: According to the topological characteristics of BP networks, a Genetic Algorithm based on the structural formula binary—coding has been designed in this paper. By means of fractionalizing the large—scale solution—space and performing the GA operations to the fractionalized subspaces; the GA's global—convergence and parallelism can be utilized to search the subspace for the optimal solution in the whole solution—space, thus definitude the starting point and narrow the domain for the next BP's local—search. Testing shows that the two—step algorithm (GA—BP) can solve the existed problems in the NN's training such as local minimum, tardy convergence and so on.

Keywords: genetic algorithm (GA); neural network (NN); coding; optimization