

文章编号:1001-9014(2007)01-0065-04

基于遗传算法设计和训练人工神经网络的方法

武妍, 王伟

(同济大学 计算机科学与技术系, 上海 200092)

摘要:提出了一种基于遗传算法的人工神经网络设计和训练方法—NNDT,同时训练网络的拓扑结构和连接权重,提出了一种结构化网络编码方法,有效地解决了网络拓扑结构和网络编码不能一一对应的问题;使用启发式规则约束网络拓扑结构的变异概率以及变异操作的选择趋势,并使用小生境机制进行拓扑变异保护.实验结果证明了NNDT的有效性和高效性.

关键词:人工神经网络;遗传算法;小生境;设计;训练

中图分类号:TP183 **文献标识码:**A

METHOD ON DESIGNING AND TRAINING OF ARTIFICIAL NEURAL NETWORK BASED ON GENETIC ALGORITHM

WU Yan, WAN Wei

(Department of Computer Science and Technology, Tongji University, Shanghai 200092, China)

Abstract: A method for designing and training artificial neural network based on genetic algorithm (NNDT) was presented. NNDT trains both architectures and weights of networks simultaneously. The problem that the one-to-one relationship between the topology and the encoding can not be kept was solved effectively. Heuristic method was used to constraint the probability of topology mutation and the trend of the choice of the kind of mutation. Also, the niching mechanism was used to protect the mutation of network topologies. The experiments results show the efficiency and rapidity of NNDT.

Key words: artificial neural network; genetic algorithm; niche; design; training

引言

用遗传算法来设计和训练人工神经网络已被证实能有效地解决很多问题^[1]. 最早的算法采用的是固定拓扑结构的神经网络^[2], 后来也出现了很多同时进化网络连接权重和拓扑结构的算法^[3-5]. 这类算法中存在着几个比较重要的问题:(1) 怎样对网络编码使得网络拓扑结构和编码能够一一对应,不同的网络间能进行有意义的交叉操作?(2) 网络拓扑结构的变异会导致适应度的降低,怎样保护这些脆弱的拓扑结构?(3) 对于一个特定的任务,如何使得到的网络拓扑结构是最优或近似最优的?

本文提出了一种人工神经网络设计和训练算法(记为NNDT). 对于以上问题,NNDT给出了有效的解决方法,对解决异或问题和平衡杆问题的

实验数据表明,NNDT能快速有效地得到所需的神经网络.

1 NNDT

1.1 基本概念

种群:遗传算法基于达尔文进化论和孟德尔遗传学说,将问题求解表示成基因型. 所有基因型的整体称为种群. NNDT中的种群就是所有的网络.

个体:种群中一个基因型称为个体.

物种:NNDT的种群被划分成一个个物种,物种由拓扑结构相似的个体组成.

物种距离:拓扑结构的相似度由物种距离定义. 两个网络差别越大物种距离越大,反之物种距离越小. 物种距离的计算方法见式(8).

收稿日期:2006-04-05,修回日期:2006-09-07

基金项目:国家自然科学基金资助项目(60475019)

作者简介:武妍(1967-),女,山西晋中人,教授、博士,主要从事神经网络、模式识别研究.

Received date: 2006-04-05, revised date: 2006-09-07

1.2 算法流程

NNDT 采用一种改进的编码方式对网络进行编码,从最小的拓扑结构开始进化.算法流程如下:

(1) 初始化种群.产生 M 个从输入节点到输出节点的全连接网络,连接权重为较小的随机数.此时种群的物种只有一个且只包含第一个网络.

(2) 遍历种群,根据物种距离获得所有网络的物种.如果一个网络不属于已有物种,成立新物种.

(3) 计算适应度.首先计算每个网络的适应度,然后得到物种适应度.如果出现适应度大于目标适应度的网络或者达到最大进化代数,进化结束.

(4) 确定每个物种可以产生的下一代网络数 m 以及其中由交叉产生的网络数 mc .

- a. 交叉操作产生 mc 个后代网络.
- b. 拓扑结构变异产生 $m-mc$ 个子代网络.
- c. 对 m 个子代网络进行权重变异.对每个连接根据权重变异概率决定是否进行变异,如果是,则在原来的权重上加上一个一定范围的随机值.
- d. 按照(2)的方法,确定网络所属物种.

(5) 跳至(3).

1.3 网络编码

比较普遍的网络编码方式是使用整数为网络节点编号,使用实数表示网络权重.这种编码方式存在着网络拓扑结构和编码不能一一对应的问题^[5],即相同结构的网络可能有不同的编码或者不同结构的网络有相同的编码.对这个问题很多算法采取的做法是完全摒弃交叉操作^[5].文献[3]给出了一个较为有效的解决方案,为每个连接分配唯一的标记.

NNDT 使用了一种更为合理的编码方式.网络的编码包括节点和连接两部分,每个节点有“节点基因编号”,每个连接有“连接基因编号”.节点基因编号是进化过程中分配给新节点的唯一编号;连接基因编号是新连接的唯一编号.NNDT 采用的增加节点的方法是分裂连接,所以用“分裂连接基因编号”来标识一个节点基因编号的来源,所有的节点基因编号组成“节点基因库”;用“始节点基因编号”和“末节点基因编号”来标识一个连接基因编号,所有的连接基因编号组成“连接基因库”.这种基于节点和连接基因库的网络编码方式可以精确地分辨出两个拓扑结构的真正异同,有效地解决了网络拓扑结构和网络编码不能一一对应的问题.

1.4 拓扑进化

1.4.1 交叉

两个父网络交叉产生两个子网络.每个父网络

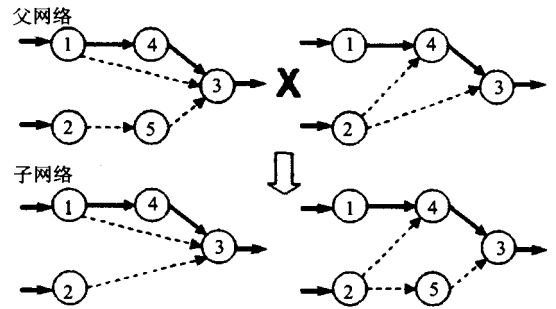


图1 交叉操作
Fig.1 Crossover operation

保持他们相同的节点和连接,每个不同的连接随机放置于一个子网络中,并且与该连接相关的节点和连接都放置于该子网络中.图1中两个父网络的实线部分是他们相同的部分,虚线节点和连接是不同的部分.两个子网络的实线部分是保持的部分,虚线部分是重新分配后的不同部分.

1.4.2 拓扑结构变异

增加连接时,选择好始末节点,如果在连接基因库中不存在相同始末节点的连接基因编号,则分配一个连接基因编号并加入连接基因库.增加节点如图2所示,一个连接分裂为一个节点两个连接.图2中左边网络虚线连接为要分裂的连接,右边网络虚线部分为分裂出的连接和节点.如果在节点基因库中没有相同分裂连接基因编号的节点基因编号,则分配节点基因编号,并加入节点基因库.由于不牵涉到基因编号分配,删除节点和连接比较简单.删除节点时相关的连接都将被删除.

NNDT 采取了一定的启发式规则约束网络拓扑结构的变异概率和变异趋势.算法假设:

- (1) 随着进化的进行,拓扑结构的变异概率整体趋势应缓慢减小;
- (2) 网络适应度越高,变异概率越小;反之越大;
- (3) 网络拓扑结构趋近于“饱和”时,算法应倾

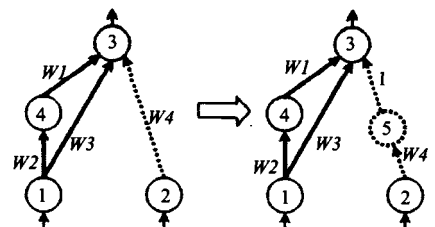


图2 增加节点拓扑结构变异操作
Fig.2 The topology mutation operation of adding node

向于删除网络连接或增加隐藏节点,反之则倾向于增加网络连接或删除隐藏节点。

拓扑结构变异概率计算公式如式(1)所示。

$$P_m = P_{ih} \left(1 - \frac{f}{f_m}\right) a^{gen} \frac{stop}{stop_m}, \quad (1)$$

式(1)中, p_{ih} 是拓扑变异的概率极限; f 是网络适应度; f_m 为适应度最大值; a 是取值 $0.9 \sim 1$ 的参数; gen 为进化代数; $stop$ 为物种停滞代数(物种适应度增加少于设定值视为停滞), $stop_m$ 为允许物种停滞的最大代数。 P_{ih} 限制了网络变异的最大概率, $(1 - f/f_m)$ 使得网络适应度越高其变异概率就越小, a^{gen} 使得拓扑结构变异概率的整体趋势随着进化进行缓慢地减小, $stop/stop_m$ 使得一个物种的适应度没有得到提高的时间越长, 该物种内的网络变异概率就越大。整个拓扑变异概率公式使得变异概率满足了启发式假设的要求, 在总体趋势随着进化进行逐渐减小的基础上, 网络适应度越高变异概率越小, 并且随着物种适应度没有提高的时间的增长, 其网络的变异概率随之相应增大。

先给出网络连接饱和度和节点饱和度的概念。对于相同数目的输入输出和隐藏节点, 网络连接越多, 则连接饱和度越高; 对于相同数目的输入输出节点, 隐藏节点数越多则节点饱和度就越高。式(2)、(3)分别为连接饱和度和节点饱和度的定义。

$$f_c = \min\left(1, \frac{aN_c}{N_h(N_i + N_o)}\right), \quad (2)$$

$$f_N = 1 - \frac{b(N_i + N_o)}{N_i + N_o + N_h}, \quad (3)$$

式(2,3)中, N_c 为连接数, N_h 为隐藏节点数, N_i 和 N_o 为输入输出节点数, a 、 b 为取值 $0.5 \sim 1$ 的参数。

根据启发式假设以及连接饱和度和节点饱和度的定义, 式(4~7)分别是网络添加节点、添加连接、删除连接、删除节点的概率计算公式。

$$P_{an} = 1 - f_N, \quad (4)$$

$$P_{ac} = f_N(1 - f_c), \quad (5)$$

$$P_{dc} = f_N f_c^2, \quad (6)$$

$$P_{dn} = f_N f_c(1 - f_c). \quad (7)$$

各种变异操作的选择概率体现了一种趋势: 网络拓扑结构越复杂变异就倾向于剪枝, 反之则倾向于增枝; 网络连接饱和度越高变异就倾向于删除连接, 反之就倾向于增加连接; 网络节点饱和度越高变异就倾向于删除节点, 反之则倾向于增加节点。

1.5 小生境进化保护

拓扑结构的变异通常会导致网络适应度的降低。如果没有保护措施, 这些低适应度的网络可能很快就被淘汰出去。算法 NEAT^[3] 中使用了小生境的概念, NEAT 只为连接分配进化号, 没有考虑节点的影响。NNDT 使用物种距离描述网络间的差异。一个网络产生后, 计算该网络与所有物种的代表个体的物种距离, 如果小于设定的极限值, 则加入这个物种。如果该网络不能加入已有的物种, 则成立一个新物种。两个网络间的物种距离的计算公式如下:

$$d = \frac{c_1 C_N}{C} + \frac{c_2}{K} \sum_{i=1}^K abs(W_i - W'_i) + \frac{c_3 N_N}{N}, \quad (8)$$

式(8)中, c_1, c_2, c_3 为系数; C_N 为连接基因编号不同的连接数, C 为连接数目的较大值; W_i, W'_i 分别为两个网络连接基因相同的连接的权重。 N_N 为节点基因编号不同的节点数; N 为节点数的较大值。物种形成后, 可以通过式(9)计算物种适应度, 该适应度用于决定物种可产生下一代的个体数目。

$$F = \max(f_i) 1 \leq i \leq k, \quad (9)$$

式(9)中, f_i 为该物种第 i 个个体的适应度。 k 为该物种的个体总数。计算出物种适应度后, 下一代中该物种能繁殖的个体数为:

$$N_k = Num \frac{F_k}{\sum_{i=1}^k F_i}, \quad (10)$$

式(10)中, Num 为种群大小; F_k 为第 k 个物种的适应度; K 为物种总数。

这样, 个体之间的竞争转化为物种之间的竞争, 物种内的个体获得相同的进化机会, 物种内适应度较小的个体就不会轻易灭绝, 个体数小的物种还能得到较为迅速地扩大。

2 实验及结果分析

2.1 异或问题

很多论文都以异或问题为例证明算法的有效性^[3,4]。异或问题有两个输入和一个输出。设四个输入样本对应的目标输出组成向量 t , 并设网络的四个输出组成的向量为 x , 则可以定义网络的适应度为:

$$f = f_m - (t - x)(t - x)^T. \quad (11)$$

式(11)中 f_m 为最大适应度, 异或问题 f_m 为 4。

多次实验的平均数据以及与其它算法的比较如表 1 所示(其他算法的数据来自文献[4])。

表 1 异或问题结果比较

Table 1 Comparison results for XOR problem

算法	平均计算单元数	平均隐藏神经元数	平均连接数	种群大小
NNDT	3308.62	2.33	7.19	100
NEvA	5904.64	4.62	15.3	13
NEAT	4755	2.35	7.48	150

NEAT 和 NEvA 分别是文献[3]和[4]提出的算法。

一个进化过程的计算单元数即为进化过程中处理的网络个数,计算方法为种群大小乘以进化代数。比如一次 NNDT 试验中如果进化 30 代结束,那么计算单元数为 100×30 等于 3000。可见,平均计算单元与算法速度成正比,平均计算单元数越多,则算法越慢,反之越快。可以看出,解决异或问题 NNDT 效果是最为理想的,其得到的网络的平均连接数和平均隐藏节点数都是最小的,其次是 NEAT,但 NNDT 的速度要比 NEAT 快 30% (平均计算单元数少 30%)。

2.2 平衡杆问题

平衡杆问题是一个很常见的非线性控制问题。其控制目标为,使用水平方向的推力,一方面保持由马达驱动的小车的车载平衡杆(一根或两根,本文实验采用两根平衡杆)呈垂直平衡的状态,另一方面要使小车的位置在一定长度的轨道内。其中小车与平衡杆的动作空间维度相同且为一维空间。本文实验模型的参数同文献[3]。

两根平衡杆问题的实验数据如表 2 所示(其他算法的数据来自文献[4])。

可以看出,对于解决两根平衡杆问题 NNDT 也是最优的。NNDT 和 NEvA 结果拓扑结构复杂性差不多,NNDT 的隐藏神经元略多于 NEvA,但连接数少,而且同时 NNDT 的速度要快于 NEvA。

3 结语

本文提出了一种基于遗传算法的人工神经网络

表 2 两根平衡杆问题结果比较

Table 2 Comparison results for 2-Pole problem

算法	平均计算单元数	平均隐藏神经元数	平均连接数	种群大小
NNDT	2032	0.81	7.2	150
NEvA	2177	0.74	7.7	40
NEAT	3578	N/A	N/A	150

设计和训练算法 NNDT。定义了一种改进的网络编码方式,这种基于连接和结点基因库的编码方式解决了网络拓扑结构和编码不能一一对应的问题。NNDT 从最小的网络拓扑结构出发,使用启发式假设约束网络拓扑变异的概率和拓扑结构的增减趋势,使得进化收敛于最优或近似最优的网络拓扑结构,并使用小生境原理对网络拓扑结构变异进行保护鼓励有效的拓扑结构进化操作。最后通过实验数据有力地证明了 NNDT 算法的有效性和高效性。

REFERENCES

- [1] WU Yan, WANG Shou-Jue. Method for improving classification performance of neural network based on fuzzy input and network inversion [J]. *J. Infrared Millim. Waves* (武研,王守觉.基于模糊化输入和反转提高神经网络分类性能的方法,红外与毫米波学报),2005,24(1):15—18.
- [2] XIAO Ben-Xian, ANG Wei-Bing, WANG Qun-Jing. On fast neural network training algorithm with hybrid genetic algorithm [J]. *Journal of Hefei University of Technology* (肖本贤,昂卫兵,王群京.用混合遗传算法实现神经网络快速训练,合肥工业大学学报).2001,24(5):901—906.
- [3] Stanley K O, Miikkulainen R. Evolving neural networks through augmenting topologies [J], *Evolutionary Computation*, 2002,10(2):99—127.
- [4] Tsoy Y R, Spitsyn V G. Using genetic algorithm with adaptive mutation mechanism for neural networks design and training [C], *Proceedings of 9th Korea-Russia International Symposium on Science and Technology KORUS-2005*, Novosibirsk,2005:709—714.
- [5] Angeline P J, Saunders G M, Pollack J B. An evolutionary algorithm that constructs recurrent neural networks [J], *IEEE Transactions on Neural Networks*,1994,5(1):54—65.