二进制互连神经网络的遗传演化

吕阳鹏[1] 李大宇[1*] 靳其兵[1] 谭天伟[2]
(北京化工大学 1. 信息科学与技术学院; 2. 生命科学与技术学院, 北京 100029)

摘 要: 当前对于二进制神经网络的研究主要集中在前馈模型上,然而前馈模型不具备反馈连接,无法形成记忆结构,从而无法处理时序逻辑问题。提出了一种基于遗传算法的二进时序神经网络演化算法,并且通过双边沿触发计数器实验和迷宫路径覆盖实验验证了演化的二进制互连神经网络具备时序逻辑处理能力。

关键词: 二进制神经网络; 时序逻辑; 遗传算法; 迷宫路径覆盖
中图分类号: TP183

引 言

二进制神经网络(binary neural networks, BNN)是一种经典的人工神经网络模型,其特点是构成网络的神经元都仅有 0 和 1 两种输出。BNN 在超平面分类和逻辑设计领域都具有广泛的应用[1-3]。已有研究证明,具有单隐层的前馈 BNN 能够完备地表达出任意的布尔逻辑[4]。然而,前馈模型不具备反馈连接,当前输出仅跟当前输入相关,从而无法实现时序逻辑功能。

目前基于遗传算法的神经网络结构演化主要有以下两种基本方法:一种是在基本拓扑结构确定的基础上,通过遗传算法优化网络的结构参数[5],收敛速度快,但灵活性较差;另一种是在交叉变异规则中引入结构演化机制,使算法在运算过程中能够动态改变个体的网络结构[6],适应性较强,但搜索空间较大,收敛速度慢。

针对以上问题,本文提出了一种改进算法:基于遗传算法的二进神经网络生成(genetic based binary network generator, GBNG)。在演化过程中,GBNG 可以生成反馈连接,从而使实现时序逻辑功能成为可能。通过限定搜索空间以及改进运算模型,GBNG 可以实现高效的网络拓扑结构演化。

收稿日期: 2009-12-12
基金项目: 国家"863"计划(2008AA04Z131);国家"973"计划(2007CB714300);北京市优秀人才资助项目(2009D013000000003)
第一作者: 男,1985 年生,硕士生
* 通讯联系人
E-mail: lidz@mail.buct.edu.cn

1 限定权值和阈值的取值范围

根据 BNN 的特性,可以限定其权值和阈值的取值范围,从而有效减少搜索空间,提高算法收敛速度,使遗传编码方式也更为直观高效。

1.1 限定权值范围

"与"、"或"和"非"是逻辑中的基本操作,已知权值为 1 的 BNN 可以实现"与"和"或"的功能[3],权值为 1 和 -1 的 BNN 可以实现"非"功能[3]。而通过基本逻辑的组合就可以实现任意的布尔逻辑。因此可以限定 GBNG 中 BNN 的权值取值范围为 $\{-1,1\}$。

1.2 限定阈值范围

设神经元 k 有 N 个输入连接,其中权值为 +1 的连接个数为 M 个,输入加权和为 S_k,第 i 个连接权值为 W_i,第 i 个连接神经元输出为 X_i,则有

$$S_k = \sum_{i=1}^{N} W_i X_i = \sum_{i=1}^{M} X_i - \sum_{j=1}^{N-M} X_j \tag{1}$$

而 $X_i, X_j \in \{0,1\}$,所以有

$$\mathrm{Max}(S_k) = \sum_{i=1}^{M} 1 - \sum_{j=1}^{N-M} 0 = M \tag{2}$$

$$\mathrm{Min}(S_k) = \sum_{i=1}^{M} 0 - \sum_{j=1}^{N-M} 1 = N - M \tag{3}$$

设神经元 k 的输出为 X_k,阈值为 B_k,则有

$$X_k = \begin{cases} 1, S_k + B_k \geqslant 0 \\ 0, S_k + B_k < 0 \end{cases} \tag{4}$$

根据式(4)可知,若使 X_k 恒为 1 或者 0,必须满足如下条件

X_k 恒为 1,当 $B_k \geqslant -\mathrm{Min}(S_k)$ (5)

X_k 恒为 0，当 $B_k < -\mathrm{Max}(S_k)$　　(6)

而在 GBNG 中，如果新添加一个神经元到现有网络中，期望这个神经元的输出能够反应出输入的变化，而不是无论输入如何变化，输出恒为 1 或者 0。所以，根据式(2)、(3)、(5)、(6)可得，阈值 B_k 的合理变化范围是：

$$\{B_k \mid -M \leqslant B_k < N - M, B_k \in I\} \quad (7)$$

其中 N 为神经元的输入连接数，M 为权值为 +1 的连接数。例如一个二进制神经元有 5 个输入连接，连接的节点分别是 ABCDE，其中 ABC 的连接权值为正，DE 的连接权值为负，则这个神经元的阈值在 [-3,1]内取值时与相应的布尔函数对应关系如表 1 所示。

表 1　阈值与布尔函数的对应关系

Table 1　The relationship between neuron bias and Boolean functions

阈值	布尔函数
-3	$ABC\,\overline{DE}$
-2	$AB\,\overline{DE} + AC\,\overline{DE} + BC\,\overline{DE} + \cdots + ABC\,\overline{D}$
-1	$A\,\overline{DE} + B\,\overline{DE} + C\,\overline{DE} + \cdots + ABC$
0	$A\,\overline{E} + B\,\overline{E} + C\,\overline{E} + \cdots + BC$
1	$A + B + C + \overline{D} + \overline{E}$

2　基于遗传算法的二进神经网络生成

2.1　GBNG 基本流程

1）初始化基因池，创建初始个体，每个个体仅具有输入节点和输出节点，节点间不存在连接。

2）对现有基因池中的个体按照适应度由大到小排序。如果存在个体适应度达到或超过最大适应度，则算法成功退出。

3）如果当前演化代数达到或者超过最大演化代数限制，则算法失败退出。

4）创建新的基因池，用于存放下一代个体。

5）按复制率复制部分适应度高的个体到新的基因池中；按变异率使部分个体发生变异，复制到新的基因池中；按交叉率对部分个体进行交叉，生成的子代复制到新的基因池中。

6）用新的基因池替换原有的基因池，跳转到 2）。

2.2　基因编码

在 32 位微机系统中，一个无符号整数占用 4 个字节即 32 Bit 的空间，由于算法的实现程序将在 32 位微机系统上执行和测试，因此基因编码方式也做出了相应的调整优化，以达到 32 位系统性能最佳化。规定单个网络最多可容纳 32 个节点，则具有 m 个输入节点，n 个输出节点，k 个隐层节点的网络基因表示法如图 1 所示：

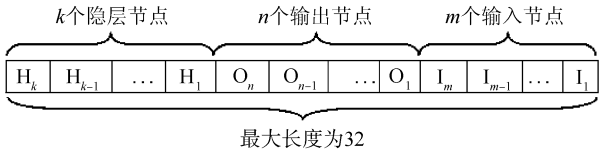


图 1　网络的编码形式

Fig. 1　The encoding of BNN

其中 $H_1 \cdots H_k$ 为隐层节点基因，O_1 到 O_n 为输出层节点基因，I_1 到 I_m 为输入层节点基因。由输入层到输出层再到隐层，节点由低位到高位顺序存储。这里面实际上隐含了每个节点的唯一标识符(ID)信息：由低位到高位，节点相对应的 ID 为 0，1，2，3，…，直到($m + n + k - 1$)。

对于每个神经元基因，占用 32 × 3 Bit 空间进行存储。

第 1 个 32 Bit 存储神经元接受的权值分布信息，由低到高的每个比特位代表具有相对应 ID 的接入神经元连接权值，0 代表权值为 -1，1 代表权值为 +1。

第 2 个 32Bit 存储神经元接受的连接分布信息，由低到高的每个比特位代表具有相应 ID 的神经元与该神经元的连接情况，0 为无连接，1 为存在相应 ID 的神经元到该神经元的连接。

第 3 个 32 Bit 存储神经元节点自身的阈值，最高位为符号位，1 表示阈值为负，0 表示阈值为正，其他各位用来表示阈值的绝对值大小。

2.3　变异算子

GBNG 包括 4 种变异操作：权值变异、阈值变异、连接变异和节点变异。

权值变异、阈值变异和连接变异方式基本相同，都是随机选择网络中的一个节点，用在限定范围内随机生成的新 32 Bit 覆盖原有的区域。

节点变异将会产生新的节点基因，加入到网络基因高位的空位中。这样在 GBNG 中随着演化代数的增加，网络的规模逐渐趋于复杂化，由简单网络结构到复杂结构拓展。

2.4　交叉算子

交叉操作会产生两个子代。对于父代基因中具

有相同 ID 的结点,按 50% 的概率分划到两个子代中。如果父代基因长度不同,则必然一个父代的节点 ID 种类是另一个的子集,即某一父代含有多余的结点基因,将这部分基因直接复制到子代 1 的相应位置中,保持交叉后的个体与父代具有相同长度。

2.5 隐性基因

在交叉变异过程中,某些神经元节点可能产生冗余的权值位或者连接位,这并不会影响到 GBNG 的运行效率。在 32 位系统中,CPU 处理运算的最小单位就是 32 Bit,所以在 32 Bit 中具有冗余位不会增加运算负担。而这些冗余位暂时是不起作用的,所以称之为隐性基因。随着演化的深入,新结点的加入,有可能使隐性基因变为显性。

3 计数器及迷宫路径覆盖实验

3.1 GBNG 公共参数设置

下面的双边沿触发计算器和迷宫路径覆盖实验中,GBNG 都采用表 2、表 3 所示的参数设置。

表 2 GBNG 公共参数设置

Table 2 GBNG public parameter settings

最大代数	基因个数	复制率	交叉率	总变异率
10000	200	5%	30%	65%

表 3 GBNG 变异率参数设置

Table 3 GBNG mutation parameter settings

权值变异率	连接变异率	阈值变异率	结点变异率
50%	15%	30%	5%

3.2 双边沿触发计数器实验

双边沿触发计数器是时序逻辑设计中一种常用的结构,可以用于计数、定时和分频[7]。本实验的目标是通过 GBNG 演化生成一个两位双边沿触发加法计数器。输入为一位二进制数,输出为两位二进制数,要求每当输入由 1→0,或者由 0→1 时,计数器的输出变为上次输出加 1。二位二进制可以表示的数为:00、01、10、11,所以这种计数器具有 4 种输出状态。通过多个二位加法计数器的组合,就能够实现任意多位的加法计数器。

3.2.1 计数器实验的适应度函数设计

设定网络输入节点数为 1,输出节点数为 2,使用训练样本(有导师学习)来评价个体适应度。顺序输入样本,记录下测试通过的样本数,显然通过的样本越多,个体适应度越大。设测试通过的训练样本个数为 n,个体为 g,则其适应度可表示为:

$$\text{fitness}(g) = n \tag{8}$$

本实验中训练样本数据共 8 组,所以在实验中个体的最大适应度值为 8,如表 4 所示。

表 4 计数器训练样本

Table 4 Training dataset for the counter problem

顺序	输入	期望输出	顺序	输入	期望输出
1	0	00	5	0	00
2	1	01	6	1	01
3	0	10	7	0	10
4	1	11	8	1	11

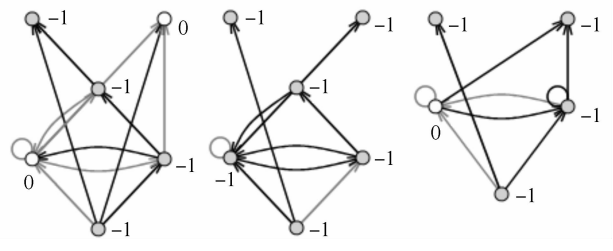
3.2.2 计数器实验结果

表 5 列出了 5 次实验的结果,这 5 次实验中,最大适应度个体都通过了所有训练样本,实际的仿真结果也说明这些网络具备计数能力。即通过 GBNG 能够演化出合适的网络结构,实现双边沿触发计数器。分析图 2 中的 3 个网络解,发现都具有一个类似的双节点互连环状结构。实验验证了这种环状结构可以实现一定的记忆存取功能,通过这种结构的组合,可以构造出高阶神经网络计数器,实现多位加法计数,如图 3 所示。

表 5 加法计数器实验结果

Table 5 Experimental results for the counter problem

演化代数	时间花费/ms	通过样本数	隐层节点个数
70	140	8	2
104	343	8	5
289	1171	8	5
553	2093	8	3
121	375	8	2



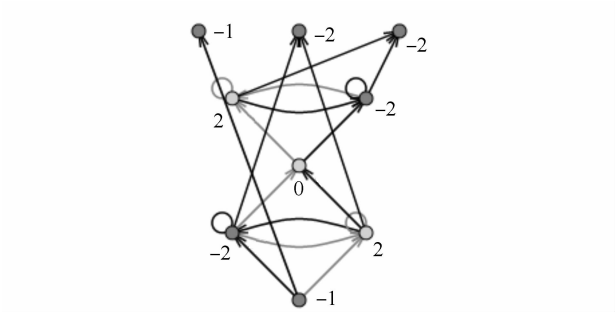
灰线表示权值为 -1;黑线表示权值 +1;节点旁的数字代表阈值

图 2 计数器问题的 3 个网络解

Fig. 2 Three network solutions for the counter problem

3.3 迷宫路径覆盖实验

迷宫覆盖问题是算法设计中的经典问题[8-9]。



灰线表示权值为 -1；黑线表示权值 +1；节点旁的数字代表阈值

图 3　三阶计数器的网络拓扑结构

Fig. 3　Topological structure of the third order counter network

规定一个起始点，要求智能体(Agent)在尽可能少的走步内尽可能多的覆盖迷宫的可行路径。智能体能够感知四周环境，即能够知道上下左右 4 个方向是墙壁还是通路，通过 4 位二进制数表示，由高到低各位分别代表上下左右 4 个方向，值若为 1 表示该方向是墙壁不可通行，若为 0 则表示该方向是通路可行。Agent 能够做出四种动作即上移、下移、左移、右移，分别用二进制数 00、01、10、11 来表示。

在这个问题中，可以把 Agent 的智能划分为 4 个等级。

第 1 等级：Agent 不能判断四周环境情况，仅做盲目的运动，有可能碰撞上墙壁，停滞不前。

第 2 等级：Agent 能够判断四周环境情况，选择合法的方向进行移动，但有可能经常走回头路，虽然合法，但是不能有效的拓展覆盖空间，浪费移动步数。

第 3 等级：Agent 具有短时记忆能力，能记住自己上一步的方位，决策下一步的移动时参考上一步的方位，尽量不走回头路，以节省步数。但是当第 2 次遇到岔路口时，往往做出跟上次一样的选择，导致在某一区域内循环，不能继续拓展空间。

第 4 等级：Agent 具有长时记忆能力，能够记录它在行走过程中的所有岔路点，每次经过这个岔路点时，都能选择一个与上次不同的方向。具有这个等级智能的 Agent，能够遍历迷宫。

3.3.1　迷宫实验的适应度函数设计

个体在仿真实验中能够覆盖的迷宫格数越多，行走步数越少，个体的适应度就应该越大。迷宫规模为 5 × 5，限制最多步数为 30 步。设网络个体为 g，在运行过程中覆盖的迷宫宫格数为 n，则其个体适应度可表示为

$$\text{fitness}(g)=\begin{cases}0, & \text{当行走过程中碰壁}\\ 0, & \text{当行走路径出现冗余}\\ n, & \text{当行走过程未出现异常}\end{cases}\quad(9)$$

若行走过程未出现异常，个体的适应度即为其覆盖的迷宫宫格数。本实验采用随机生成的 5 × 5 宫格的迷宫，因此个体最大适应度值为 25。

3.3.2　迷宫路径覆盖实验结果

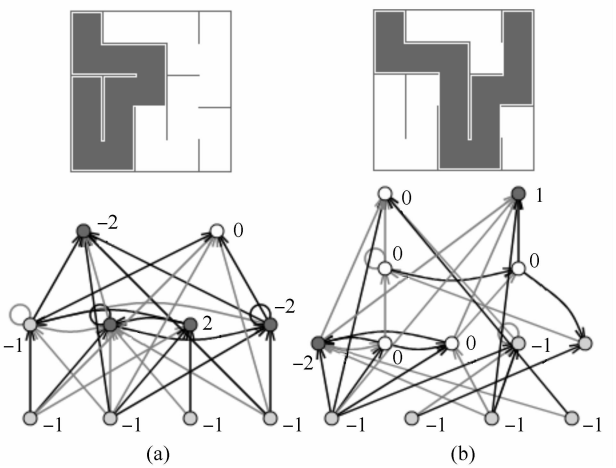
表 6 给出了 5 次实验的结果，迷宫具有 5 × 5 = 25 个宫格，因为在 5 次实验中，在规定的最大演化代数内都未能覆盖全部宫格，因此表中的演化代数和时间花费都取自演化过程中刚好能达到最大宫格数时的数据。

表 6　迷宫路径覆盖实验结果

Table 6　Results of maze path covering problem

演化代数	时间花费/ms	覆盖宫格数	隐层节点个数
1023	4468	13	6
1592	8421	11	4
407	2156	13	4
424	1921	13	5
1267	5578	13	3

图 4 中，(a)、(b)分别是迷宫覆盖 13 宫格和 11 宫格的两种情况，以及相应的网络解。通过仿真实验跟踪这两个网络在迷宫中的行为，能够判定，这两个解网络的智能都达到了第三等级，具有短时记忆能力，不会走重复路径，除非遇到死胡同。但是在返回到分叉路口处时，不能选择与上次不同的方向，因而未能达到第 4 等级的要求。



灰线表示权值为 -1；黑线表示权值 +1；节点旁的数字代表阈值

图 4　迷宫路径覆盖实验的网络解

Fig. 4　Network solutions for the maze path covering

4 结论

在计数器实验中,GBNG 演化出了一种环状结构,经过分析,这种环状结构具有状态记忆功能,通过组合可以实现更高阶的神经网络计数器;在迷宫覆盖实验中,GBNG 演化出的网络控制 Agent 的行为,使其表现出了一定的智能。通过这两个实验的实验结果可以验证,GBNG 能够演化出一定的记忆逻辑结构,具备一定的时序逻辑问题处理能力。GBNG 在神经网络结构自动化设计以及时序逻辑自动化设计等领域都具备应用前景。

参考文献:

[1] Starzyk J, Pang J. Evolvable binary artificial neural network for data classification: Proc. of Int. Conf. on Parallel and Distributed Processing Techniques and Applications (PDPTA)[C]. MA: MIT Press, 2000.

[2] Tsakonas A, Dounias G, Doumpos M, et al. Bankruptcy prediction with neural logic networks by means of grammar-guided genetic programming[J]. Expert Systems with Applications, 2006(3): 449 – 461.

[3] Hitzler P, Hølldobler S, Seda A. Logic programs and connectionist networks[J]. Journal of Applied Logic, 2004(3): 245 – 272.

[4] Chen F, Chen G, He G, et al. Universal perceptron and DNA-like learning algorithm for binary neural networks: LSBF and PBF implementations[J]. IEEE Transactions on Neural Networks, 2009(10): 1645 – 1658.

[5] Yu S, Zhu K, Diao F. A dynamic all parameters adaptive BP neural networks model and its application on oil reservoir prediction[J]. Applied Mathematics and Computation, 2008(1): 66 – 75.

[6] Lehman J, Stanley K. Exploiting open-endedness to solve problems through the search for novelty: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems(Artificial Life XI)[C]. MA: MIT Press, 2008.

[7] 谢建华, 叶卫东, 韩跃峰. 双边沿触发计数器的设计和应用[J]. 兵工自动化, 2006(4): 71 – 72.
Xie J H, Ye W D, Han Y F. Design and application of double edge trigger counter[J]. Ordnance Industry Automation, 2006 (4): 71 – 72. (in Chinese)

[8] Kim C, Ogata T, Sugano S. Self-organizing algorithm for logic circuit based on local rulues[J]. Transactions-Society of Instrument and Control Engineers, 2006 (4): 334 – 341.

[9] Stanley K, Bryant B, Miikkulainen R. Real-time neuroevolution in the NERO video game[J]. IEEE Transactions on Evolutionary Computation, 2005 (9): 653 – 668.

A genetic algorithm-based binary neural network generator

LV YangPeng[1] LI DaZi[1] JIN QiBing[1] TAN TianWei[2]

(1. College of Information Science and Technology; 2. College of Life Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China)

Abstract: Current research on Binary Neural Networks is focused on the Feed Forward Model. However, the Forward Model cannot form backward links, which are a necessary part of memory structure and, therefore, it cannot be used to solve sequential logic problems. This paper presents a Genetic Algorithm-based Binary Neural Network Generator, whose capability of creating sequential logic neural networks is verified by two experiments: a double-edge triggered counter experiment and a maze path covering experiment.

Key words: binary neural network; sequential logic; genetic algorithm; maze path covering