

奇偶校验问题的二进神经网络学习算法

杨娟¹ 陆阳^{1,2} 朱晓娟¹ 邱述威¹

(合肥工业大学计算机与信息学院 合肥 230009)¹

(安徽省矿山物联网与安全监控技术重点实验室 合肥 230088)²

摘要 二进神经网络可以完备表达任意布尔函数,但对于孤立节点较多的奇偶校验问题却难以用简洁的网络结构实现。针对该问题,提出了一种实现奇偶校验等孤立节点较多的一类布尔函数的二进神经网络学习算法。该算法首先借助蚁群算法优化选择真节点及伪节点的访问顺序;其次结合几何学习算法,根据优化的节点访问顺序给出扩张分类超平面的步骤,从而减少隐层神经元的数目,同时给出了隐层神经元及输出元的表达形式;最后通过典型实例验证了该算法的有效性。

关键词 二进神经网络,布尔函数,奇偶校验问题,蚁群算法

中图分类号 TP18 **文献标识码** A

Learning Algorithm of Binary Neural Networks for Parity Problems

YANG Juan¹ LU Yang^{1,2} ZHU Xiao-juan¹ QIU Shu-wei¹

(School of Computer and Information, Hefei University of Technology, Hefei 230009, China)¹

(The Anhui Provincial Key Laboratory of Mine IoT and Mine Safety Supervisory Control, Hefei 230088, China)²

Abstract Binary neural network can completely express arbitrary Boolean function, but more isolated nodes such as parity problem are difficult to implement with simple network structure. According to this problem, we presented a learning algorithm to realize Boolean function such as parity problems with many isolated samples. By means of the ant colony algorithm, we obtained the optimized core nodes and the extension order of true and false nodes, by combing the geometrical algorithm, we gave the steps of how to expand the classifier hyperplanes with the optimized core nodes, so this algorithm can reduce the number of hidden neurons in network, and the expression of the hidden neurons and the output neuron are also given. Finally, this algorithm is validated to be effective through examples.

Keywords Binary neural networks, Boolean function, Parity problems, Ant colony algorithm

1 引言

对于布尔空间中函数映射问题,采用二进神经网络来实现比采用其他前向多层神经网络可靠性更强,学习速度更快。二进神经网络是采用阈值函数作为激发函数的三层前向网络,只要有足够数目的隐层神经元,即可以实现任意给定的布尔函数。

设 B_2 表示 $\{0, 1\}$ 二元域, F 是 $B_2^n \rightarrow B_2$ 的任意映射,即 F 为 n 元布尔函数,对所有 $X = (x_1, x_2, \dots, x_n) \in B_2^n$, 用 $F(x_1, x_2, \dots, x_n)$ 表示 F 在 X 处的取值,简记为 $F(X)$, 则实现 $F(X)$ 的二进神经网络的三层结构可由下式表达:

$$S_i = U(\sum_{j=1}^n w_j^i x_j - T_i) = \begin{cases} 1, & \sum_{j=1}^n w_j^i x_j \geq T_i \\ 0, & \sum_{j=1}^n w_j^i x_j < T_i \end{cases} \quad i=1, 2, \dots, L$$

$$O = U(\sum_{i=1}^L w_i^* S_i - \theta) = \begin{cases} 1, & \sum_{i=1}^L w_i^* S_i \geq \theta \\ 0, & \sum_{i=1}^L w_i^* S_i < \theta \end{cases}$$

记 $W^i = \{w_1^i, w_2^i, \dots, w_n^i\}$, $W^* = \{w_1^*, w_2^*, \dots, w_n^*\}$, 其中每个 $S_i = S_i(X, W^i, T_i)$ 为隐层神经元, $O = O(X, W^*, \theta)$ 为输出元,网络结构如图 1 所示。

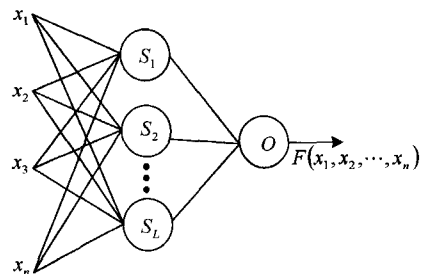


图 1 二进神经网络结构图

到稿日期:2012-03-09 返修日期:2012-07-06 本文受国家自然科学基金项目(61070220),国家“863”计划项目(2011AA060406),高等学校博士学科点专项科研基金(20090111110002)资助。

杨娟(1983-),女,博士生,主要研究方向为人工智能、神经网络,E-mail: yangjuan6985@163.com;陆阳(1967-),男,教授,博士生导师,主要研究方向为人工智能、计算机控制;朱晓娟(1978-),女,博士生,讲师,主要研究方向为无线传感器网络;邱述威(1975-),男,博士生,副教授,主要研究方向为 Ipv6 网络。

减少二进神经网络中隐层神经元数目仍是目前二进神经网络学习算法研究的焦点^[1-4]。近年来随着智能算法的发展,借助遗传算法优化分类超平面分离节点的顺序^[5,6],取得了较好的结果,减少了隐层神经元的数目,文献[7]借助蚁群算法优化分类超平面分离节点的顺序,对于孤立节点较少或在卡诺图中连通性较高的问题,得到了较为满意的结果;但对于孤立节点较多或连通性较差的问题,如奇偶校验问题,每个真节点都是孤立节点,文献[8]采用枚举法使较为简洁的网络结构实现奇偶校验,但随着维数的加大,采用枚举法难以实现,文献[7]的算法可以避免枚举法,但隐层需要 2^{n-1} 个神经元。针对此问题,本文提出针对样本孤立节点较多的二进神经网络学习算法,该算法为文献[3,9]理论上分析的结果提供了一种切实有效的实现方法。

2 相关概念

定义 1 样本值为“1”的节点称为真节点,样本值为“0”的节点称为伪节点;由每个超平面所分离的第一个节点称为核心节点。

定义 2 $A \subseteq F^{-1}(1)$,若满足下列两个条件之一,则称集合 A 是 $F^{-1}(1)$ 中的一个连通集。

(1) $C(A)=1$;

(2) $C(A) \geq 2$,且 $\forall X^p, X^q \in A, X^p \neq X^q$,存在样本序列 $X^1, X^2, \dots, X^k \in A$,满足:

$$\begin{cases} d_H(X^p, X^1) = 1 \\ d_H(X^i, X^{i+1}) = 1 \quad i=1, \dots, (k-1) \\ d_H(X^k, X^q) = 1 \end{cases}$$

定义 3 若样本 $X^g \in F^{-1}(1)$,且 $\forall X', d_H(X', X^g) = 1$,满足 $X' \in F^{-1}(0)$,则称 X^g 为 $F^{-1}(1)$ 中的孤立节点。

以图 2 中的 5 变量布尔函数卡诺图为例,其中 01000, 10010, 10101 均为孤立节点, {11110, 11111, 01111, 01011, 11011, 01101} 是一个连通集。图 3 为 5 变量奇检验的卡诺图,根据定义可以看出,每个真节点都是孤立节点。

		x_3, x_2, x_1							
x_5, x_4		000	001	011	010	110	111	101	100
00		0	0	0	1	0	0	0	0
01		0	1	0	0	0	1	0	0
11		0	0	1	1	1	1	0	0
10		0	0	0	0	1	0	1	0

图 2 5 变量布尔函数卡诺图

		x_3, x_2, x_1							
x_5, x_4		000	001	011	010	110	111	101	100
00		1	0	1	0	1	0	1	0
01		0	1	0	1	0	1	0	1
11		1	0	1	0	1	0	1	0
10		0	1	0	1	0	1	0	1

图 3 5 变量奇检验卡诺图

3 学习算法

设 $F(x_1, x_2, \dots, x_n)$ 为待学习的布尔函数,设 $V_T = \{X^{s_1}, X^{s_2}, \dots, X^{s_q}\}$ 为真节点集,即有 $F^{s_1} = F^{s_2} = \dots = F^{s_q} = 1, V_F =$

$\{X^{l_1}, X^{l_2}, \dots, X^{l_z}\}$ 为伪节点集,即有 $F^{l_1} = F^{l_2} = \dots = F^{l_z} = 0$,且 $z+q=2^n$ 。

3.1 算法思想

该算法的基本思想是首先借助蚁群算法^[10]来选择一个真节点作为核心节点,然后再利用几何学习算法^[8]不断扩张获得隐层神经元对应的分类超平面;依次再通过真假节点变换,对该集合作进一步的扩张来获得下一个隐层神经元的分类超平面,直到真节点集或伪节点集为空。

首先引入如下记号:

m 表示蚁群中蚂蚁的数量。

L_{\max} 表示每次循环中所分离的节点的最大数目。

$d_{ij}, (i, j=1, 2, \dots, q)$ 表示真样本点间的汉明距离, q 为真节点个数。

$\overline{d_{ij}}, (i, j=1, 2, \dots, z)$ 表示伪样本点间的汉明距离, z 为伪节点个数。

$\tau_{ij}(t), (i, j=1, 2, \dots, q)$ 表示 t 时刻在 ij 连线上残留的信息素量,初始时,各条边上的信息素量相等,设 $\tau_{ij}(0) = \tau_0, (i, j=1, 2, \dots, q)$ 。

$\overline{\tau_{ij}}(t), (i, j=1, 2, \dots, z)$ 表示 t 时刻在 ij 连线上残留的信息素量,初始时,各条边上的信息素量相等,设 $\overline{\tau_{ij}}(0) = \overline{\tau_0}, (i, j=1, 2, \dots, z)$ 。

随着时间的推移, ij 连线上残留的信息素量逐渐降低,用参数 ρ 表示持久性,用 $1-\rho$ 表示衰减程度,即以前留下的信息逐渐消失,蚂蚁们完成一次循环,各条边上的信息素量根据下式进行调整:

$$\begin{aligned} \tau_{ij}(t+1) &= (1-\rho)\tau_{ij}(t) + q \sum_{i=1}^m \Delta\tau_{ij}^k \\ \overline{\tau_{ij}}(t+1) &= (1-\rho)\overline{\tau_{ij}}(t) + z \sum_{i=1}^m \overline{\Delta\tau_{ij}^k} \end{aligned}$$

式中, $\Delta\tau_{ij}^k$ 表示第 k 只蚂蚁在 t 时刻时对于真节点 i, j 之间信息量的改变, $\overline{\Delta\tau_{ij}^k}$ 表示第 k 只蚂蚁在 t 时刻时对于伪节点 i, j 之间信息量的改变,

$$\begin{aligned} \Delta\tau_{ij}^k &= \begin{cases} L(k), & \text{蚂蚁 } k \text{ 经过真节点 } i, j \\ 0, & \text{else} \end{cases} \\ \overline{\Delta\tau_{ij}^k} &= \begin{cases} L(k), & \text{蚂蚁 } k \text{ 经过真节点 } i, j \\ 0, & \text{else} \end{cases} \end{aligned}$$

式中, $L(k)$ 表示第 k 只蚂蚁分离所能分离的真、伪节点的数目。

$p_{ij}^k(t)$ 表示 t 时刻时,对于真节点,蚂蚁 k 由位置 i 转移到位置 j 的概率,可设为如下形式:

$$p_{ij}^k(t) = \begin{cases} \frac{(\tau_{ij}(t))^\alpha \left(\frac{1}{d_{ij}}\right)^\beta}{\sum_{s \in allowed_k} (\tau_{is}(t))^\alpha \left(\frac{1}{d_{is}}\right)^\beta}, & j \in allowed_k \\ 0, & \text{else} \end{cases}$$

式中, $allowed_k$ 表示蚂蚁 k 目前尚未访问过的真节点集合, α, β 分别为控制真节点间信息素量及其距离的相对重要程度。

$\overline{p_{ij}^k}(t)$ 表示 t 时刻时,对于伪节点,蚂蚁 k 由位置 i 转移到位置 j 的概率,可设为如下形式:

$$\overline{p_{ij}^k}(t) = \begin{cases} \frac{(\overline{\tau_{ij}}(t))^\alpha \left(\frac{1}{\overline{d_{ij}}}\right)^\beta}{\sum_{s \in allowed_k} (\overline{\tau_{is}}(t))^\alpha \left(\frac{1}{\overline{d_{is}}}\right)^\beta}, & j \in \overline{allowed_k} \\ 0, & \text{else} \end{cases}$$

式中, $allowed_k$ 表示蚂蚁 k 目前尚未访问过的伪节点集合, α , β 分别为控制伪节点间信息素量及其距离的相对重要程度。

每个隐层神经元系数, 可由定理 1 得到。

定理 1^[7] 若集合 $A = \{X^1, X^2, \dots, X^P\} \subseteq F^{-1}(1)$ 可以用一个神经元表达, 其中 $X^k = (x_1^k, x_2^k, \dots, x_n^k)$, $(k=1, 2, \dots, P)$, 如果: $W = \{w_1, w_2, \dots, w_n\}$, $w_i = 2 \sum_{k=1}^P x_i^k - P$, $i=1, 2, \dots, n$, $T = \min_{X \in A} (\sum_{i=1}^n w_i x_i)$, 则神经元 $S = S(X, W, T)$ 表达集合 A 。

对于输出元系数, 可由定理 2 得到。

定理 2^[9] 在几何学习算法中, 若首先分离真节点, 则第奇数个隐层神经元与输出元的连接权为 1, 第偶数个隐层神经元与输出元的连接权为 -1; 若隐层共有奇数个神经元, 则输出元的阈值为 1, 若隐层共有偶数个神经元, 则输出元的阈值为 0。

3.2 算法步骤

Step1 初始化神经元数目 $J=1$;

Step2 $L_{max}=0$, 若神经元数目 J 为奇数时, 执行 Step3, 若 J 为偶数时, 执行 Step14;

Step3 若伪节点集非空, 即伪节点个数 $z \neq 0$, 执行 Step4, 否则转到 Step25;

Step4 初始化迭代次数 $NC=1$;

Step5 将 m 只蚂蚁分布到 q 个真节点上;

Step6 以样本点间的汉明距离及外信息素为变量的概率函数选择下一个真节点;

Step7 规定蚂蚁走合法路线, 即在一次循环中, 每只蚂蚁不允许转到已访问的真节点, 由禁忌表控制;

Step8 在 m 只蚂蚁的每次循环中, 得到真节点的访问顺序表 $R^{[i]} = \{R_1^{[i]}, R_2^{[i]}, \dots, R_q^{[i]}\}$, $i=1, 2, \dots, m$;

Step9 针对蚂蚁 i , 第 J 个隐层神经元所分离的真节点数目初始化为 $j^{[i]}=0$;

Step10 按照真节点访问顺序找到第 J 个神经元所分离的第一个真节点 $M^{[i]} = \{M_{S_1}^{[i]}\}$, 并将顺序表 $R^{[i]}$ 中的 $M_{S_1}^{[i]}$ 删除, 且 $j^{[i]} = j^{[i]} + 1$;

Step11 遍历顺序表 $R^{[i]}$, 判断是否可以将顺序表中剩余真节点逐个加入到集合 $M^{[i]}$ 中, 且与未分离的真节点及未分离的伪节点构成线性可分集, 若可以加入, 则将已经分离的真节点角标在顺序表 $R^{[i]}$ 中删除, 并且 $j^{[i]} = j^{[i]} + 1$;

Step12 第 i 只蚂蚁所分离的真节点数目即为 $L(i) = j^{[i]}$, 蚂蚁 i 在它每条访问的边上留下外激素 $L(i)$, 并且记录当前 $L(i)$ 中的最大值 L_{max} , 即目前最好解;

Step13 根据新得到的外激素 $L(i)$ 更新转移概率, $NC = NC + 1$, 若 NC 小于预定的迭代次数, 则转到 Step5; 否则输出目前最好解, 并 $q = q - L_{max}$, 更新剩余的真节点集, $J = J + 1$;

Step14 若真节点集非空, 即真节点个数 $q \neq 0$, 执行 Step15, 否则转到 Step25;

Step15 初始化迭代次数 $NC=1$;

Step16 将 m 只蚂蚁分布到 z 个伪节点上;

Step17 以样本点间的汉明距离及外信息素为变量的概率函数选择下一个伪节点;

Step18 规定蚂蚁走合法路线, 即在一次循环中, 每只蚂

蚁不允许转到已访问的伪节点, 由禁忌表控制;

Step19 在 m 只蚂蚁的每次循环中, 得到伪节点的访问顺序表 $Z^{[i]} = \{Z_1^{[i]}, Z_2^{[i]}, \dots, Z_z^{[i]}\}$, $i=1, 2, \dots, m$;

Step20 针对蚂蚁 i , 第 J 个隐层神经元所分离的真节点数目初始化为 $j^{[i]}=0$;

Step21 按照访问伪节点顺序找到第 J 个神经元所分离的第一个伪节点 $Z^{[i]} = \{Z_{S_1}^{[i]}\}$, 并将顺序表 $Z^{[i]}$ 中的 $Z_{S_1}^{[i]}$ 删除, 且 $j^{[i]} = j^{[i]} + 1$;

Step22 遍历顺序表 $Z^{[i]}$, 判断是否可以将顺序表中剩余伪节点逐个加入到集合 $Z^{[i]}$ 中, 且与未分离的伪节点及未分离的真节点构成线性可分集, 若可以加入, 则将已经分离出的伪节点角标在顺序表 $Z^{[i]}$ 中删除, 同时 $j^{[i]} = j^{[i]} + 1$;

Step23 第 i 只蚂蚁所分离的伪节点数目即为 $L(i) = j^{[i]}$, 蚂蚁 i 在它每条访问的边上留下外激素 $L(i)$, 并且记录当前 $L(i)$ 中的最大值 L_{max} , 即目前最好解;

Step24 根据新得到的外激素 $L(i)$ 更新转移概率, $NC = NC + 1$, 若 NC 小于预定的迭代次数, 则转到 Step5; 否则输出目前最好解, 并 $z = z - L_{max}$, 更新剩余的伪节点集, $J = J + 1$, 转到 Step2;

Step25 输出结果。

算法流程图如图 4 所示。

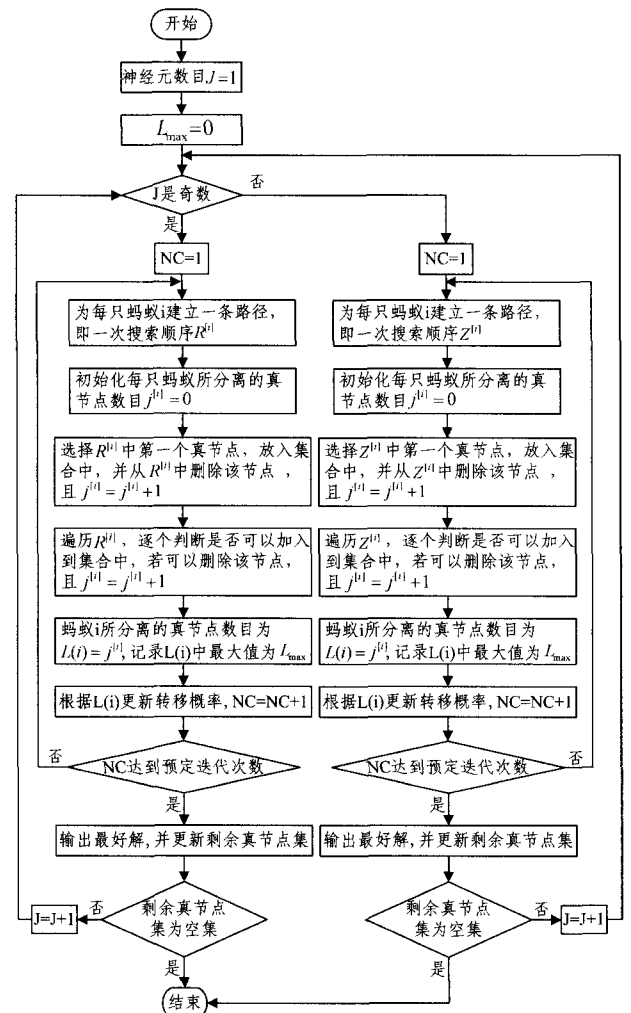


图 4 算法流程图

3.3 算法复杂度分析及改进

对于任意布尔函数 $F(x_1, x_2, \dots, x_n)$, 采用二进制神经网络

来实现,若理论上最少需要 L 个隐层神经元,则上述算法所需进行的迭代次数约为 $L * NC$,若算法中使用 m 只蚂蚁,则该算法总的运行次数约为 $L * NC * m$,较基于几何学习算法^[9]等一类算法采用枚举法在运算效率上有大幅度的提高,但该算法效率仍然不高,在具体算法中,随着真节点和伪节点逐渐被分离出去,根据运行情况分析,所需的蚂蚁数目以及迭代次数都可以相应地减少,进而可以减少总的运行次数,减少运行时间。另外,在几何学习算法中仅在核心节点选择上采用枚举法,故将上述算法进一步改进,仅在每次选择核心节点时运用蚁群算法,在选择核心节点后超平面扩张时从邻近节点开始,以减少相应的迭代次数,具体步骤如下。

Step1 初始化神经元数目 $J=1$;
 Step2 初始化迭代次数 $NC=1$;
 Step3 将 m 只蚂蚁分布到未分离的真节点上;
 Step4 以样本点间的汉明距离及外信息素为变量的概率函数选择下一个真节点;

Step5 规定蚂蚁走合法路线,即在一次循环中,每只蚂蚁不允许转到已访问的真节点,由禁忌表控制;

Step6 在 m 只蚂蚁的每次循环中,得到真节点的访问顺序表 $R^{[j]} = \{R_1^{[j]}, R_2^{[j]}, \dots, R_q^{[j]}\}, i=1, 2, \dots, m$;

Step7 针对蚂蚁 i ,第 J 个隐层神经元所分离的真节点数目初始化为 $j^{[i]}=0$;

Step8 按照真节点访问顺序找到第 J 个神经元所分离的第一个真节点 $M^{[i]} = \{M_{S_1}^{[i]}\}$,并将顺序表 $R^{[i]}$ 中的 $M_{S_1}^{[i]}$ 删除,且 $j^{[i]} = j^{[i]} + 1$;

Step9 遍历顺序表 $R^{[i]}$,判断是否可以将顺序表中剩余真节点逐个加入到集合 $M^{[i]}$ 中,且与未分离的真节点及未分离的伪节点构成线性可分集,若可以加入,则将已经分离的真节点角标在顺序表 $R^{[i]}$ 中删除,并且 $j^{[i]} = j^{[i]} + 1$;

Step10 第 i 只蚂蚁所分离的真节点数目即为 $L(i) = j^{[i]}$,蚂蚁 i 在它每条访问的边上留下外激素 $L(i)$,并且记录当前 $L(i)$ 中的最大值 L_{max} ,及取最大值所分离的节点集合 H 与分离的第一个节点 v_c ;

Step11 根据新得到的外激素 $L(i)$ 更新转移概率, $NC=NC+1$,若 NC 小于预定的迭代次数,则转到 Step2;否则输出目前最好的核心节点 v_c 及分离节点集合 $V=H$;

Step12 交换 V_T 与 V_F 中节点的函数值,即真节点变为伪节点,伪节点变为真节点, $J=J+1$;

Step13 在保证 (V, \bar{V}) 线性可分的前提下,向 V 中逐个加入尽可能多的与 V 邻近的真节点;

Step14 交换 V_T 与 V_F 中节点的函数值, $J=J+1$;

Step15 若在保证 (V, \bar{V}) 线性可分的前提下, \bar{V} 中没有任何节点可以加入到 V 的红色节点,在执行 Step16,否则执行 Step13;

Step16 若 \bar{V} 中所有节点函数值相同,则执行 Step18,否则将已经分离开的 V 中节点标记为无关节点,在后续分离过程中不考虑这些节点;

Step17 执行 Step2;

Step18 输出结果。

图 5 是具体流程图,图 6 是调用蚁群算法的流程图。

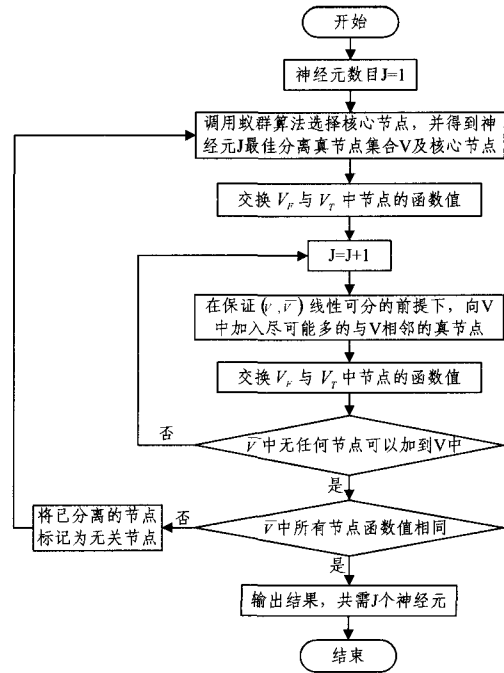


图 5 改进算法的主流程图

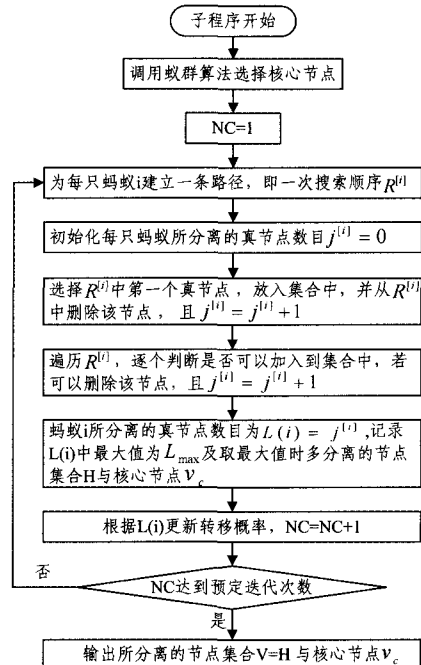


图 6 蚁群算法选择核心节点(调用程序)

4 数值实例

该算法主要用来解决存在多孤立样本的情况,对此,以 5 元及 6 元奇偶校验为例来说明本文算法的有效性。

表 1 神经元逐个分离的节点顺序(5 元奇校验)

神经元	每次所分离的节点
S_1	01010;
S_2	11010, 00010, 01110, 01000, 01011;
S_3	10010, 11110, 11000, 11011, 00110, 00000, 00011, 01100, 01111, 01001;
S_4	10110, 10000, 10011, 11100, 11111, 11001, 00100, 00111, 00001, 01101;
S_5	10100, 10111, 10001, 11101, 00101;

例1 5元奇校验,即当 $(x_1, x_2, x_3, x_4, x_5)$ 中有奇数个“1”时, $F(x_1, x_2, x_3, x_4, x_5)=0$,偶数个“1”时, $F(x_1, x_2, x_3, x_4, x_5)=1$,该算法的运行结果如表1、表2所列。

表2 分类超平面的系数及阈值(5元奇校验)

神经元	$\{w_1, w_2, w_3, w_4, w_5\}$	阈值
S_1	$\{-1, 1, -1, 1, -1\}$	2
S_2	$\{-4, 4, -4, 4, -4\}$	4
S_3	$\{-6, 6, -6, 6, -6\}$	0
S_4	$\{-4, 4, -4, 4, -4\}$	-4
S_5	$\{-1, 1, -1, 1, -1\}$	-2

其二进神经网络结构图如图7所示。

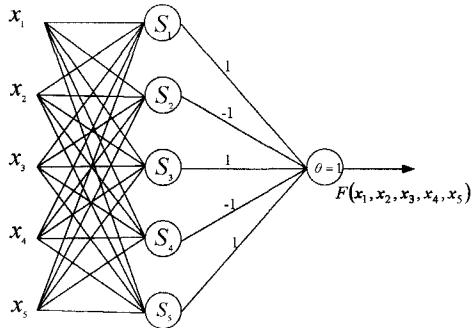


图7 5元奇校验网络结构图

例2 6元偶校验,即当 $(x_1, x_2, x_3, x_4, x_5, x_6)$ 中有奇数个“1”时, $F(x_1, x_2, x_3, x_4, x_5, x_6)=1$,偶数个“1”时, $F(x_1, x_2, x_3, x_4, x_5, x_6)=0$,该算法的运行结果如表3、表4所列。

表3 神经元逐个分离的节点顺序(6元偶校验)

神经元	每次所分离的节点
S_1	101100;
S_2	001100, 111100, 100100, 101000, 101110, 101101;
S_3	011100, 000100, 001000, 001110, 001101, 110100, 111000, 111110, 111101, 100000, 100110, 100101, 101010, 101001, 101111;
S_4	010100, 011000, 011110, 011101, 000000, 000110, 000101, 001010, 001001, 001111, 110000, 110110, 110101, 111010, 111001, 111111, 100010, 100001, 100111, 101011;
S_5	100011, 111011, 110111, 110001, 110010, 001011, 000111, 000001, 000010, 011111, 011001, 011010, 010101, 010110, 010000;
S_6	110011, 000011, 011011, 010111, 010001, 010010;

表4 分类超平面的系数及阈值(6元偶校验)

神经元	$\{w_1, w_2, w_3, w_4, w_5, w_6\}$	阈值
S_1	$\{1, -1, 1, 1, -1, -1\}$	3
S_2	$\{5, -5, 5, 5, -5, -5\}$	10
S_3	$\{10, -10, 10, 10, -10, -10\}$	10
S_4	$\{10, -10, 10, 10, -10, -10\}$	0
S_5	$\{5, -5, 5, 5, -5, -5\}$	-5
S_6	$\{1, -1, 1, 1, -1, -1\}$	-2

其二进神经网络结构图如图8所示。

从实例1、2可以看出,针对所有节点均为孤立节点的奇偶校验问题,该算法可以较为快速地使用简洁的网络结构来实现,解决了文献[7,11]等较为困难的问题。

结束语 本文提出了一种将蚁群算法与几何学习算法相结合的二进神经网络学习算法,其可以实现具有较多孤立节

点的布尔函数,以奇偶校验问题为例,该算法采用较为简单的结构即可实现,并且避免了几何学习算法中需要枚举搜索核心节点的缺点,而且所有的分类超平面及阈值均为整数,较易在硬件上实现,因而该算法成功地解决了奇偶校验等孤立节点较多的布尔函数实现问题。

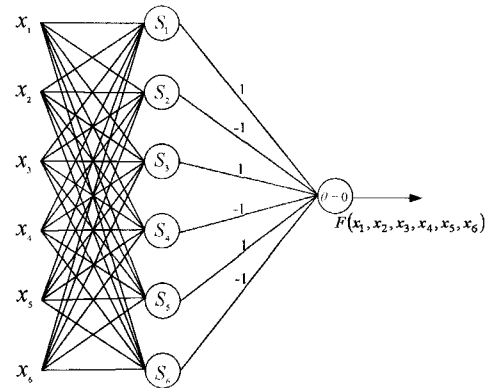


图8 6元偶校验网络结构图

参考文献

- [1] Chen Fang-yue, Chen Guan-rong, He Guo-long. Universal Perceptron and DNA-Like Learning Algorithm for Binary Neural Networks; LSBF and PBF Implementations [J]. IEEE Transactions on Neural network, 2009, 20(10): 1645-1658
- [2] Chen Fang-yue, Chen Guan-rong, He Qin-bin. Universal Perceptron and DNA-Like Learning Algorithm for Binary Neural Networks; Non-LSBF Implementation [J]. IEEE Transactions on Neural network, 2009, 20(8): 1293-1301
- [3] Lu Yang, Yang Juan, Wang Qiang, et al. The upper bound of the minimal number of hidden nodes in parity problem by neural networks [J]. Science in China Ser. F Information Sciences, 2007 (7)
- [4] Chaudhari N S, Tiwari A. Binary Neural Network classifier and it's bound for the number of hidden layer neurons [C]// International conference on Control Automation Robotics and Vision, Singapore, Dec. 2010
- [5] Shimada M, Satio T. A GA-based learning algorithm for binary neural networks [J]. IEICE Trans. Fundamentals, 2002, E85-A (11): 2544-2546
- [6] Hirane T, Toryu T, Nakano H, et al. Analysis for Characteristics of GA-Based Learning Method of Binary Neural Networks [J]. ICANN, 2005, LNCS 3696: 323-329
- [7] 杨娟, 陆阳, 方欢, 等. 基于蚁群算法的二进神经网络学习算法 [J]. 电路与系统学报
- [8] Kim J H, Park S K. The geometrical learning of binary neural networks [J]. IEEE Trans Neural Networks, 1995, 6(1): 237-247
- [9] Fung H K, Li L K. Minimal feedforward parity networks using threshold gates [J]. Neural Computation, 2001, 13(2): 319-326
- [10] 高尚, 杨静宇. 群智能算法及其应用 [M]. 北京: 中国水利水电出版社, 2006
- [11] 华强, 郑启伦. 二进神经网络的汉明图学习算法 [J]. 计算机学报, 2001, 24(11): 1250-1255