第26卷 第3期
2000年5月

自 动 化 学 报
ACTA AUTOMATICA SINICA

Vol. 26, No. 3
May, 2000

二进制映射神经网络的几何学习算法及其应用[1)]

朱大铭* 马绍汉
（山东大学计算机科学系，济南 250100）
（* E-mail: zam@cs.sdu.edu.cn）

魏道政
（中国科学院计算所 北京 100080）

摘 要 提出一种一般二进制映射问题的前馈网络学习算法. 给出一种求解超平面以几何分割训练点的新方法，不仅相应地构造了隐层神经网络，而且使得只需再构造一个输出层网络便可实现训练样本所描述的映射. 该算法在学习收敛速度方面优于 BP 算法和 SC 算法，对样本数据的分布和密集程度变化适应性强，具有较好的容错能力.

关键词 神经网络，学习算法，训练样本，超平面.

A GEOMETRICAL LEARNING ALGORITHM FOR BINARY MAPPING NEURAL NETWORKS AND ITS APPLICATION

ZHU Daming MA Shaohan
(*Dept. of Computer Science, Shandong University, Jinan* 250100)
WEI Daozheng
(*Institute of Computing Technology, Chinese Academy of Sciences, Beijing* 100080)

Abstract A learning algorithm is presented for the general binary mapping problem. The algorithm employs a new method to compute hyper-planes to divide the training points into distinct areas so that the hidden layer of neural networks is correspondingly constructed. This makes it possible that only one output-layer needs construction to implement the mapping determined by the training points. The algorithm is better than BP and SC algorithms in respect of the convergence velocity, is more adaptive to the change of the distribution and density of the training points, and has a better error-tolerant ability.

Key words Neural networks, learning algorithm, training sample, hyper-plane.

1) 国家自然科学基金、国家“八六三-三〇六”计划项目、山东省自然科学基金资助课题.
收稿日期 1998-03-16 收修改稿日期 1999-05-19

1 引 言

近年来,前馈神经网络的高效学习算法一直受到许多学者的关注. STONE-WEIERSTRASS 定理[1]定性地保证了前馈神经网络逼近任意映射的能力,但并未给出确定神经网络结构的具体方法. BP 算法[2]是目前应用广泛且研究很深的前馈网络学习算法,标准 BP 算法在特殊情况下可有不同程度的改进[3]. 但 BP 算法本身具有收敛速度慢且学习不精确的缺陷. 由于 BP 算法要求每个神经元具有连续可导的作用函数,因而用于二进制映射学习问题时弱点表现得更明显. 造成 BP 算法收敛速度慢甚至不收敛的重要原因是该算法不能确定神经网络结构.

对于二进制映射二分问题,目前已有一些构造学习算法[4,5],但这些方法均不能推广到一般的二进制映射问题. 对于二进制映射 K 分类问题,目前见到的有 FP 算法[6]、顺序构造(SC)方法[7]和几何算法[8]. SC 算法仍需根据人为经验确定部分神经网络参数,收敛速度仍不能令人满意,且神经元突触权值指数增长速度很快. FP 算法产生的隐层神经元太多且存在拒识情况,当训练样本集合规模较大时并不实用. 我们给出的几何方法[8]可构造四层网络实现 K 分类,但对一般的二进制映射无能为力,且未考虑权值指数增长问题.

本文基本算法建立一个3层前馈神经网络实现训练样本所描述的映射. 每个神经元的突触权值和阈值均由算法一次确定,确定后便不再修改. 因而算法的学习复杂度只是训练样本个数的多项式函数,算法保证学习的收敛性. 分析表明,本文算法学习收敛速度快于 BP 和 SC 算法,训练样本的多少和分布对算法的干扰较小,在这方面优于 FP 算法,且具有较好的容错能力. 另外,算法所确定的所有神经元参数均为整数,算法的改进形式可十分有效地将神经元参数大小控制在较小的范围内,因而特别适合集成电路实现.

二进制映射神经网络的学习问题由一组训练样本来描述:$U=\{u_1,u_2,\cdots,u_N\}$,其中每个训练样本 u_i 由输入向量 $\boldsymbol{x}_i\in\{0,1\}^n$ 和输出向量 $\boldsymbol{y}_i\in\{0,1\}^m$ 组成,记为 $u_i=(\boldsymbol{x}_i,\boldsymbol{y}_i)$. 训练样本 u_i 的输入和输出向量又分别记为 $IV(u_i)=\boldsymbol{x}_i$ 和 $OV(u_i)=\boldsymbol{y}_i$;并记 $IV(U)=\{IV(u_i)\,|\,u_i\in U\}$,$OV(U)=\{OV(u_i)\,|\,u_i\in U\}$,其中 $IV(U)$和 $OV(U)$分别称为学习问题的输入向量集和输出向量集. 下文算法描述中,总将输入向量看作 n 维空间中的点,因此样本的输入向量也称为该样本的训练点. 实际上训练集合 U 描述了一个二进制映射 $F:\{0,1\}^n\to\{0,1\}^m$. 学习问题的目的是建立一个多层神经网络,实现训练样本所描述的映射. 样本的输出向量表达了该样本或该样本输入向量的类别,因此二进制映射问题也称为二进制映射分类问题. 若训练样本的输出向量只取一位0或1,则该问题即为二分问题. 给定二进制映射问题的训练集合 U,若某前馈网络使对任意 $u\in U$,当以 $IV(u)$输入时,该网络总有输出 $OV(u)$,则该神经网络称为该给定问题的目标网络. 本文算法获得的网络神经元作用函数均采用硬限函数,即 $y=f(\sigma)=0$,若 $\sigma<0$;$f(\sigma)=1$,若 $\sigma\geqslant 0$,其中 $\sigma=\sum_{i=1}^{n}w_i\boldsymbol{x}_i-\theta$. 单个神经元如图1所示. 任一$(n-1)$维超平面对 n 维空间的分离作用与一个 n 输入的神经元作用等价.

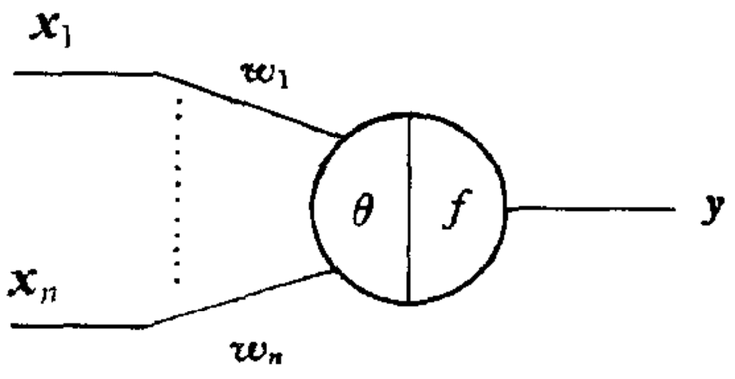

图1 神经元模型

2　隐层网络学习

隐层学习算法求解一组超平面实现所有训练点的几何分割. 由所得的超平面组形成等价隐层神经网络. 隐层学习要获得有效分割训练点的超平面.

2.1　候选样本集和核心顶点的选择

算法开始在训练样本集 U 中任意选择一个样本 u_c,并根据 u_c 确定 U 的子集 $U_c=\{u_k|u_k\in U, OV(u_k)=OV(u_c)\}$,$U_c$ 称为候选样本集合. 然后算法将 $IV(U_c)$ 中的训练点指定为真顶点;将 $IV(U)-IV(U_c)$ 中的训练点指定为假顶点.

定义1. 给定二进制映射学习问题的训练集合 U. 设训练点集 V_t 是真顶点集,若 V_t 与 $IV(U)-V_t$ 是线性可分的,则称 V_t 是真顶点分离集.

最初真顶点分离集 $V_t=\varnothing$,算法将通过真顶点分离集的不断扩展使 U 中的样本依次得到训练,并逐渐得到分离超平面. 但进入 V_t 中的训练点必须从 $IV(U_c)$ 中选择. U_c 确定后,算法便在 $IV(U_c)$ 中选择第一个真顶点 $\boldsymbol{v}_{\ker}\in IV(U_c)$ 加入 V_t,则 $V_t=\{\boldsymbol{v}_{\ker}\}$ 必然是真顶点分离集. $\boldsymbol{v}_{\ker}$ 称为核心顶点,核心顶点可以在 $IV(U_c)$ 中任意选择,也可以利用 k 最近相邻算法[9]选择. 设 $\boldsymbol{v}_{\ker}=(v_{k1},v_{k2},\cdots,v_{kn})^{\mathrm{T}}$,则分离 V_t 的 $(n-1)$ 维超平面为

$$\sum_{i=1}^{n}(2v_{ki}-1)\boldsymbol{x}_i-\sum_{i=1}^{n}v_{ki}=0. \tag{1}$$

2.2　真顶点分离集的扩展

真顶点分离集的扩展就是要选择新的真顶点进入 V_t,并保持 V_t 为真顶点分离集.

定义2.　设 $\boldsymbol{x},\boldsymbol{y}\in\{0,1\}^n$,将 $\boldsymbol{x}$ 和 $\boldsymbol{y}$ 之间的海明距离记为 $H(\boldsymbol{x},\boldsymbol{y})$. 另设 $V\subseteq\{0,1\}^n$ 为二进制向量集合,则将 $\boldsymbol{x}$ 与 V 中所有向量的海明距离之和记为 $d_H(\boldsymbol{x},V)=\sum_{\boldsymbol{y}\in V}H(\boldsymbol{x},\boldsymbol{y})$. 对 $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}\in\{0,1\}^n$ 和 $I=\{i_1,i_2,\cdots,i_m\}$,$m\leqslant n$,$1\leqslant i_j\leqslant n$,$1\leqslant j\leqslant m$,记 $\boldsymbol{x}_{[I]}=(x_{i_1},x_{i_2},\cdots,x_{i_m})^{\mathrm{T}}$.

由定义2和文[8]的分析易得如下结论.

定理1. 对于核心顶点 $\boldsymbol{v}_{\ker}$ 和候选训练集 U_c,若有正整数 d 和正整数集合 I 使 $V_t=\{\boldsymbol{v}_{\ker}\}\cup\{\boldsymbol{x}|\boldsymbol{x}\in V(U),H(\boldsymbol{x},\boldsymbol{v}_{\ker})\leqslant d-1\}\cup\{\boldsymbol{x}|\boldsymbol{x}\in V(U),H(\boldsymbol{x},\boldsymbol{v}_{\ker})=H(x_{[I]},\boldsymbol{v}_{\ker[I]})=d\}$ 中的训练点均为真顶点,则 V_t 是真顶点分离集. 其中 $I=\{i_1,i_2,\cdots,i_m\}$,$1\leqslant i_j\leqslant n$,$1\leqslant j\leqslant m$,$1\leqslant d\leqslant m\leqslant n$.

方法1. 1)确定最大的 d 使 $V_N=\{\boldsymbol{x}|\boldsymbol{x}\in IV(U),H(\boldsymbol{x},\boldsymbol{v}_{\ker})\leqslant d-1\}\subseteq IV(U_c)$,真顶点分离集扩展为 $V_{\mathrm{tnew}}=V_{told}+V_N$,候选样本集减小为 $U_{\mathrm{cnew}}=U_{\mathrm{cold}}-\{u|IV(u)\in V_N\}$;2)确定一个整数集合 I,使训练点集 $V_I=\{\boldsymbol{x}|\boldsymbol{x}\in IV(U),\ H(\boldsymbol{x},\boldsymbol{v}_{\ker})=H(\boldsymbol{x}_{[I]},\boldsymbol{v}_{\ker[I]})=d\}\subseteq IV(U_c)$,真顶点分离集扩展为 $V_{\mathrm{tnew}}=V_{\mathrm{told}}+V_I$,候选样本集减小为 $U_c=U_c-\{u|IV(u)\in V_I\}$. 确定分离超平面的具体方法见文[8].

方法1用来将真顶点分离集同时扩展多个顶点,但该方法只适合于特殊情况. 欲进一步扩展真顶点分离集,必须探讨更一般的真顶点分离集的判定方法.

定理2. 给定二进制映射学习问题的训练集合 U. 若 V_t 是真顶点集,且 $\max\{d_H(\boldsymbol{x},V_t)|\boldsymbol{x}\in V_t\}<\min\{d_H(\boldsymbol{x},V_t)|\boldsymbol{x}\in V(U)-V_t\}$,则 V_t 是真顶点分离集,分离超平面为 $P(\boldsymbol{x})=$

$\sum_{i=i}^{n}(2c_i-c_0)x_i-\theta=0$，其中 $c_0=|V_t|$，$c_i=\sum_{x\in V_t}\sum_{i=1}^{n}x_i-\max\{d_H(\boldsymbol{x},V_t)|\boldsymbol{x}\in V_t\}$.

证明. 任给 $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}\in\{0,1\}^n$，有

$$P(\boldsymbol{x})=\sum_{y\in V_t}\sum_{i=1}^{n}(2y_i-1)x_i-\theta=\sum_{y\in V_t}\sum_{i=1}^{n}y_i-d_H(\boldsymbol{x},V_t)-\theta. \tag{2}$$

若 $\boldsymbol{x}\in V_t$，则 $P(\boldsymbol{x})=\max\{d_H(\boldsymbol{x},V_t)|\boldsymbol{x}\in V_t\}-d_H(\boldsymbol{x},V_t)\geqslant 0$；若 $x\in IV(U)-V_t$，则 $P(\boldsymbol{x})\leqslant\max\{d_H(\boldsymbol{x},V_t)\ |\boldsymbol{x}\in V_t\}-\min\{d_H(\boldsymbol{x},V_t)|\boldsymbol{x}\in V(U)-V_t\}<0$. 证毕.

方法2. 设 U_c 为候选训练集，V_t 为真顶点分离集. 则在 $IV(U_c)-V_t$ 中选择训练点 $\boldsymbol{v}_n$，使 $d_H(\boldsymbol{v}_n,V_t)=\min\{d_H(\boldsymbol{x},V_t)|\boldsymbol{x}\in IV(U_c)-V_t\}$. 若 $\max\{d_H(\boldsymbol{x},V_t)|\boldsymbol{x}\in V_t+\{\boldsymbol{v}_n\}\}<\min\{d_H(\boldsymbol{x},V_t)|\boldsymbol{x}\in(IV(U)-V_t-\{\boldsymbol{v}_n\}\}$，则真顶点分离集扩展为 $V_{\mathrm{tnew}}=V_{\mathrm{told}}+\{\boldsymbol{v}_n\}$，候选训练集减小为 $U_{\mathrm{cnew}}=U_{\mathrm{cold}}-\{u|IV(u)=\boldsymbol{v}_n\}$，分离超平面由定理2确定；否则真顶点分离集扩展失败.

算法选择核心顶点后，首先按照方法1，然后再按方法2循环扩展真顶点分离集 V_t. 当方法2扩展失败时，算法得到第一个分离超平面，对应得到第一个隐层神经元.

2.3 候选训练集的替换与真顶点分离集再扩展

算法在方法2扩展失败时，选择另一类样本形成新的候选训练集.

2.3.1 候选训练集替换

1)将 $IV(U_c)-V_t$ 中的真顶点恢复为假顶点，U_c 中的样本恢复为普通样本.

2)在当前训练集中选择样本 u_n，$\boldsymbol{v}_n=IV(u_n)$，使 $d_H(\boldsymbol{v}_n,V_t)=\min\{d_H(\boldsymbol{x},V_t)|\boldsymbol{x}\in IV(U)-V_t\}$；

3)形成新候选训练集 $U_c=\{u|OV(u)=OV(u_n)\}$，$IV(U_c)$ 中的点全部设置成真顶点.

候选集替换后，$V_t+IV(U_c)$ 中的训练点均为真顶点. 按照2.2所述，算法再次按照方法1首先扩展 V_t 一次，再按照方法2循环扩展 V_t. V_t 再次扩展失败时，得到第2个分离超平面，对应得到第2个隐层神经元. 继续这一过程，每当按照方法2扩展真顶点分离集失败时，得到一个分离超平面，对应得到一个隐层神经元. 若这个过程可以一直进行，直到所有训练点均已进入真顶点分离集，则所有隐层神经元全部产生，隐层网络学习结束.

若候选训练集替换后，真顶点分离集仍不能继续扩展，则算法进入阻塞状态. 设算法遇到阻塞状态时，真顶点分离集 $V_t=IV(U_z)$，$U_z\subset U$. 此时 U_z 中的训练样本是已经被训练过的样本，称为无关样本，V_t 中的训练点称为无关训练点. 建立向量集 V_r 表示无关训练点集. 算法开始时，无关训练点集为空，算法遇到阻塞状态时，如下处理.

2.3.2 阻塞处理

在训练集 U 中清除无关样本 $U_{\mathrm{new}}=U_{\mathrm{old}}-U_z$ 和无关训练点集 $V_{\mathrm{rnew}}=V_{\mathrm{rold}}+V_t$；并将真顶点分离集清空 $V_t=\varnothing$.

无关样本和无关训练点清除后，算法只需重新选择候选训练集和核心顶点，便可继续真顶点分离集的扩展过程. 阻塞处理后的无关训练点集对于真顶点分离集的扩展仍然是有意义的. 阻塞后核心顶点当然不能选择无关训练点，但核心顶点选定后，利用方法1和方法2继续扩展 V_t 时，$IV(U_c)+V_r$ 中的训练点均视为真顶点，但 V_r 中的点不会在候选集替换时变为假顶点. 显然，算法阻塞一次至少有一个非无关样本得到训练，因而隐层学习算法最多阻塞 N 次即可结束，隐层学习结束时得到所有隐层神经元. 也可以完全不考虑无

关训练点,这样会使真顶点分离集每扩展一次便有一个样本得到训练,因而学习复杂度较低.但对这种方法获得的隐层网络,我们目前找不到能够控制输出层网络权值大小的有效算法.

3　输出层网络学习

假设隐层神经网络学习得到 M 个隐层神经元,记为 $Nh_1,Nh_2,\cdots,Nh_M$,神经元 Nh_i 由其输入权值和阈值($\boldsymbol{w}(Nh_i),\theta(Nh_i)$)唯一地确定,其中 $\boldsymbol{w}(Nh_i)=(w_1(Nh_i),\cdots,w_n(Nh_i))^{\mathrm{T}}$,$1\leqslant i\leqslant M$. 当训练点 $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}$ 作为网络输入时,Nh_i 的输出为 $h_i(x)=f(\boldsymbol{w}^{\mathrm{T}}(Nh_i)\cdot\boldsymbol{x}-\theta(Nh_i))$. 隐层神经网络实际实现了一个映射 $F_1:\{0,1\}^n\to\{0,1\}^M$.

定义3. 在隐层学习中,设当隐层神经元 $Nh_i(1\leqslant i\leqslant M)$ 产生时,最后一个进入真顶点分离集的训练点 $\boldsymbol{v}_{[Nh_i]}=IV(u_{[Nh_i]})$,则 $OV(u_{[Nh_i]})$ 称为 Nh_i 的输出标志,记 $D(Nh_i)=OV(u_{[Nh_i]})$.

3.1　基本算法

引理1. 给定二进制映射学习问题的训练集合 $U=U_{i0}+U_{i1}$,其中 $U_{i0}=\{u\,|\,\mathrm{bit}(OV(u),i)=0,\ u\in U\}$,$U_{i1}=\{u\,|\,\mathrm{bit}(OV(u),i)=1,u\in U\}$,$1\leqslant i\leqslant m$,$\mathrm{bit}(OV(u),i)$ 表示 $OV(u)$ 的第 i 位. 若由隐层学习产生 M 个隐层神经元,则在 M 维空间中,集合 $V_0=\{(h_1(\boldsymbol{x}),h_2(\boldsymbol{x}),\cdots,h_M(\boldsymbol{x}))^{\mathrm{T}}\,|\,\boldsymbol{x}\in IV(U_{i0})\}$ 和 $V_1=\{(h_1(\boldsymbol{x}),h_2(\boldsymbol{x}),\cdots,h_M(\boldsymbol{x}))^{\mathrm{T}}\,|\,\boldsymbol{x}\in IV(U_{i1})\}$ 是线性可分的.

证明. 考虑如下算法构造的 M-1维超平面.

构造超平面 $\mathrm{CP}(i,w,\theta)$:

1) $w_1=w_2=\cdots=w_M=0$;

2) 若 $\mathrm{bit}(D(Nh_M),i)=1$,则取 $\theta=0$,否则 $\theta=1$;

3) for $k=M-1$ to 1 step -1 do

(3.1)若 $\mathrm{bit}(D(Nh_k),i)=\mathrm{bit}(D(Nh_{k+1}),i)$,则取 $w_k=0$,否则

(3.2)若 $\mathrm{bit}(D(Nh_k),i)=1$,则取 $w_k=\max\{\theta-\sum_{j=k+1}^{M}w_jx_j\,|\,x_j\in\{0,1\}\}$,否则

(3.3)$w_k=\min\{\theta-\sum_{j=k+1}^{M}w_jx_j\,|\,x_j\in\{0,1\}\}-1$.

设 $\boldsymbol{x}\in IV(U)$ 对应有隐层输出向量 $(h_1(\boldsymbol{x}),h_2(\boldsymbol{x}),\cdots,h_M(\boldsymbol{x}))^{\mathrm{T}}$. 再设 $\boldsymbol{x}$ 进入真顶点分离集后,隐层学习产生的第一个隐层神经元为 $Nh_{k(x)}$. 根据隐层学习算法不难得到 $h_{k(x)}(\boldsymbol{x})=1$. 但对于 k,$1\leqslant k<k(\boldsymbol{x})$,$h_k(\boldsymbol{x})=0$. 取 $l(\boldsymbol{x})$ 是满足 $w_{l(x)}\neq0$ 且 $l(\boldsymbol{x})\geqslant k(\boldsymbol{x})$ 的最小正整数,则

$$\sum_{k=1}^{M}w_kh_k(\boldsymbol{x})-\theta=w_{l(x)}+\sum_{k=l(x)+1}^{M-1}w_kh_k(\boldsymbol{x})-\theta. \tag{3}$$

根据算法 $\mathrm{CP}(i,\boldsymbol{w},\theta)$ 的(3.2)和(3.3)不难得(3)式对应的超平面分割 V_0 和 V_1. 证毕.

定理3. 给定二进制映射学习问题的训练集合 U. 若由隐层学习产生 M 个神经元的隐层网络,则必存在 m 个神经元作为输出层,形成该给定问题的目标网络.

证明. 输出层网络学习算法确定神经元 $Nout_i:(\boldsymbol{w}(Nout_i),\theta(Nout_i))$,$1\leqslant i\leqslant m$. 方法

如下:1)调用 CP(i,$\boldsymbol{w}$,θ)计算得到 $\boldsymbol{w}$,θ;2)$\boldsymbol{w}(Nout_i)=\boldsymbol{w}$;$\theta(Nout_i)=\theta$. 由引理1不难得到对任意 $u\in U$,有 $out(IV(u))=OV(u)$. 证毕.

引理1和定理3给出了输出层神经网络学习算法并保证了算法的正确性.

3.2 改进算法

输出层学习使每个神经元权值在计算过程中指数增长. 已有的一些构造学习算法[5,7,8]都存在这一问题. 下面的改进算法主要考虑如何有效地降低神经元权值的指数增长速度. 设 Nh_i,Nh_j 是由隐层学习算法产生的两个神经元. 若 Nh_i,Nh_j 产生时对应的真顶点分离集有相同的核心顶点,则称隐层神经元 Nh_i,Nh_j 同核. 不加证明地给出如下结论.

引理2. 给定二进制映射学习问题的训练集 U. 隐层学习得到 M 个隐层神经元 Nh_1,Nh_2,…,Nh_M. 任给 $u\in U$,设 Nh_i 是 $IV(u)$进入真顶点分离集后产生的第一个隐层神经元,则1) $h_i(IV(u))=1$;2)任意 $j<i$,$h_j(IV(u))=0$;任意 $j>i$,若 Nh_i 和 Nh_j 同核,则 $h_j(IV(u))=1$;3) 任给同核神经元 Nh_i 和 Nh_j,若 $1\leqslant i<j\leqslant M$,且 $h_i(IV(u))=1$,则 $h_j(IV(u))=1$.

由引理2,可将算法 CP(i,$\boldsymbol{w}$,θ)改进为

S-CP(i,$\boldsymbol{w}$,θ):

1),2)同算法 CP(i,$\boldsymbol{w}$,θ)的1),2);

3)weight$=1$;max_$w=0$;min_$w=-1$;

4)for $k=M-1$ down to 1 step -1 do

(4.1)若 Nh_k 与 Nh_{k+1}同核,则

(4.1.1)若 bit($D(Nh_k)$,i)$=$bit($D(Nh_{k+1})$,i),则取 $w_k=0$;否则

(4.1.2)若 bit($D(Nh_k)$,i)$=1$,则取 $w_k=$weight;否则,$w_k=-$weight;

(4.2)若 Nh_k 与 Nh_{k+1}不同核,则

(4.2.1)若 bit($D(Nh_k)$,i)$=1$,则取 $w_k=-$min_w;否则 $w_k=-$max_$w-1$;

(4.2.2) $\text{max\_}w_{\text{new}}=\text{max\_}w_{\text{old}}-\text{min\_}w_{\text{old}}$; $\text{min\_}w_{\text{new}}=\text{min\_}w_{\text{old}}-\text{max\_}w_{\text{old}}-1$;

weight$=-$min_$\boldsymbol{w}$.

定理4. 给定二进制映射学习问题的训练集 U. 隐层学习算法结束后,输出层网络形成算法可调用 S-CP(i,$\boldsymbol{w}$,θ)确定输出层神经元. 所得神经网络必为该问题的目标网络.

证明. 取 $l(\boldsymbol{x})=\min\{l\,|\,Nh_{k(x)+l}$与 $Nh_{k(x)}$不同核$\}$. 根据引理2和算法 S-CP(i,w,θ)不难得

$$\text{min\_}w_{[k(x)+l(x)]}\leqslant\sum_{i=k(\boldsymbol{x})+l(\boldsymbol{x})}^{M}w_ih_i(\boldsymbol{x})-\theta\leqslant\text{max\_}w_{[k(\boldsymbol{x})+l(\boldsymbol{x})]},\tag{4}$$

其中 $\text{min\_}w_{[k(x)+l(x)]}$,$\text{max_}w_{[k(x)+l(x)]}$分别表示算法 S-CP($i$,$\boldsymbol{w}$,$\theta$)计算到 $k=k(\boldsymbol{x})+l(\boldsymbol{x})$时 max_$w$ 和 min_w 值. 类似于引理1的分析,有

$$\sum_{k=1}^{M}w_kh_k(\boldsymbol{x})-\theta=\sum_{k=k(x)}^{k(x)+l(x)-1}w_kh_k(\boldsymbol{x})+\sum_{k=k(x)+l(x)}^{M-1}w_kh_k(\boldsymbol{x})-\theta.\tag{5}$$

分析 $\boldsymbol{x}$ 和 bit($D(Nh_{k(x)+l(x)-1})$,i)四种取值情况,再由(4)和(5)式可得定理4成立. 证毕.

由算法 S-CP(i,$\boldsymbol{w}$,θ)生成的神经元突触权值每当遇到阻塞隐层神经元时会有一次倍增. 倍增速度已远远小于 CP(i,$\boldsymbol{w}$,θ)所得权值的倍增速度.

3.3　附加层方法

考虑在隐层和输出层之间再增加一层神经元 $Na_1, Na_2, \cdots, Na_{M-1}$.

附加层学习算法:确定神经元$(\theta(Na_i), \boldsymbol{w}(Na_i))$,$1 \leqslant i \leqslant M-1$.

1)$\theta(Na_i)=1$;2)若 $j=i$,则 $w_j(Na_i)=1$;若 $j>i$ 或 $j<i$ 且 Nh_j 非阻塞神经元,则 $w_j(Na_i)=0$;若 $j<i$ 且 Nh_j 是阻塞神经元,则 $w_j(Na_i)=-1$,其中$1 \leqslant j \leqslant M-1$.

定理5. 给定二进制映射学习问题的训练集 U. 隐层网络确定后,附加层网络由附加层学习算法确定,输出层网络调用 S-CP$(i, \boldsymbol{w}, \theta)$确定,则 S-CP$(i, \boldsymbol{w}, \theta)$可按照所有隐层神经元同核计算. 所得神经网络必为该问题的目标网络.

引入附加层后,附加层和输出层神经元的突触权值和阈值只取-1,0,1三种情况.

4　性能分析及应用

隐层学习必结束,输出层学习显然也必结束,因此算法是收敛的. 下面考虑收敛速度. 隐层学习算法从核心顶点选择到遇到阻塞,复杂度为 $O(N^2)$. 而真顶点分离集扩展过程中,阻塞次数最多为 N. 故隐层学习的复杂度为 $O(N^3)$. 算法 CP$(i, \boldsymbol{w}, \theta)$和 S_CP$(i, \boldsymbol{w}, \theta)$的复杂度均为 $O(M)$,因此输出层网络学习的复杂度为 $O(mM)$. 故本文学习算法的最坏复杂度为 $O(N^3+mM)$. 因此算法的复杂度是训练样本个数的多项式函数. BP 算法的复杂度是网络结构参数的指数函数,这是导致 BP 算法收敛速度慢的重要原因. SC 算法仍依赖于多个不可控制的网络结构参数,其复杂度本质上也是指数的. 关于容错能力,先给出如下结论.

定理6. 给定二进制映射问题的训练集合 U. 若某超平面将一组训练点 $V \subseteq IV(U)$与其他训练点分开,则必存在一个超球面也可将这组训练点与其他训练点分开. 反之亦然.

由定理6知,每个训练样本都有一个超球面吸引域与之对应. 显然相对密集的训练点将有相同的吸引域,训练点的多少与训练点间的密集程度对本文算法的容错能力影响不大. 若训练样本个数很多或训练点之间的海明距离很小,FP 算法无法找到包围样本点的超球面,容错能力较差. 在这种情况下,本文算法仍有较好的容错效果. 另外,本文算法构造的网络不存在拒识点.

考虑算法构造的神经网络规模,BP 算法根本不能确定神经网络规模,因而与 BP 算法比较意义不大. FP 算法使得隐层神经元个数与训练样本个数相同,在实际应用时局限性太大. 本文算法可利用超平面将同类且海明距离较近的训练点分割在同一区域中,因而目标网络的隐层神经元个数较少. 对于训练样本的分布和密集程度变化,本文算法具有较强的适应能力. 一些实验表明,本文算法得到的隐层神经元个数与 SC 算法相当.

奇偶函数可由特殊的神经网络实现,但学习算法只能根据训练样本决定神经网络. 利用奇偶函数可以检测学习算法的性能. 下面考虑偶函数,设 $n=10, m=1$. 当 n 位输入有偶数个1时输出为1,否则输出为0. 随机选择 N 个样本训练神经网络,然后检测所有

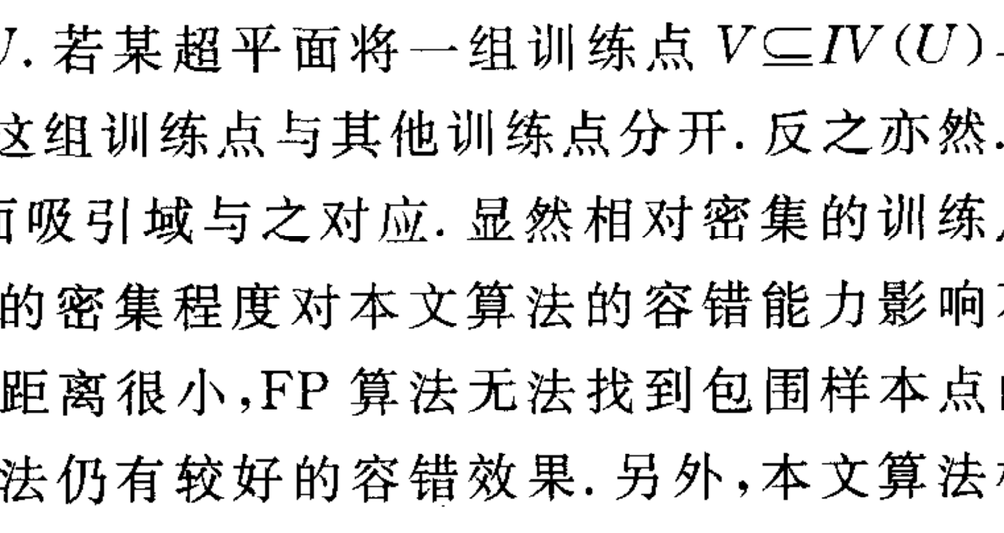

图2　利用奇偶函数检测网络容错能力

1024个向量输入神经网络时的神经网络输出，统计其正确率. 结果与 SC 算法比较如图2所示. 任意正规串可以由一个确定的有限状态自动机来描述. 可将有限自动机的状态转换表作为样本集训练前馈神经网络. 加上信息反馈形成一个 Recurrent 神经网络，该网络可用于识别正规串. 另外，本文算法在电力负荷预测应用中获得了较好的效果.

参 考 文 献

1 Cotter D E. The Stone-Weistrass theorem and its apllication to neural networks. *IEEE Trans. Neural Networks*, 1990, **1**(12)

2 Rumelhart D E *et al*, Parallel Distributed Processing, Cambridge, MA:MIT Press,1987

3 Vogl T P, Mabgis J K, Rigler A K *et al*. Accelerating the convergence of the back-propagation method, *Biol. Cybern.*, 1988,**59**:257～263

4 Gray D L, Michel A N. A training algorithm for binary feed-forward neural networks. *IEEE Trans. Neural Networks*, 1993, **4**(3):176～194

5 Kim J H, Park S K. The geometrical learning of binary neural networks. *IEEE Trans. Neural Networks*, 1995,**6**(1):91～105

6 张铃等. 多层前馈神经网络的学习与综合方法. 软件学报,1995, **6**(7)

7 Musselli M. On sequential construction of binary neural networks. *IEEE Trans. Neural Networks*, 1995, **6**(3)

8 朱大铭，马绍汉. 二进制神经网络分类问题的几何学习算法. 软件学报,1997, **8**(8):622～629

9 Fukunaga K. Introduction to statistical pattern recgnition,(2nd edition), New York:Academic,1990

朱大铭 1964年生，副教授，博士生. 从事神经网络、算法分析与设计研究工作.

马绍汉 1938年生，教授，博士导师. 从事算法分析与设计、人工智能研究工作.

魏道政 研究员，博士导师. 从事电路故障诊断、算法分析与设计研究工作.