

模糊格构造型形态神经网络

李 兵¹,董 俊²,刘鹏远¹,米双山¹

(1. 军械工程学院四系,河北石家庄 050003;2. 电子信息系统复杂电磁环境效应国家重点实验室,河南洛阳 471003)

摘 要: 针对构造型形态神经网络(CMNN)决策函数的局限性,提出了一种模糊格构造型形态神经网络(FL-CMNN);该模型在利用训练好的CMNN进行分类时,引入模糊格包容性测度计算测试样本属于各超盒的隶属度值.采用仿真数据集对提出的FL-CMNN模型进行了评价,并与原始的CMNN和传统的人工神经网络、支持向量机、最近邻分类器进行了对比;试验结果表明,FL-CMNN在测试精度上明显优于原始的CMNN,训练时间远远低于传统的神经网络和支持向量机,而分类精度丝毫不亚于传统的神经网络和支持向量机.

关键词: 数学形态学;形态神经网络;模糊格;模式识别

中图分类号: TP18 **文献标识码:** A **文章编号:** 0372-2112(2014)02-0319-09

电子学报 URL: <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2014.02.017

Fuzzy Lattice Constructive Morphological Neural Network

LI Bing¹, DONG Jun², LIU Peng-yuan¹, MI Shuang-shan¹

(1. Forth Department, Mechanical Engineering College, Shi Jia-zhuang, Hebei 050003, China

2. State Key Laboratory of Complex Electromagnetic Environmental Effects on Electronics&Information System, Luoyang, Henan 471003, China)

Abstract: A novel neural network model named fuzzy lattice constructive morphological neural network (FL-CMNN) is presented to overcome the deficiency of the original constructive morphological neural network(CMNN), which suffers for the problem of decision function in classification phase. The fuzzy lattice inclusion measure function is introduced to calculate the membership of testing sample belong to the hyper-boxes trained by the CMNN. Three standard datasets are employed to evaluate and compare the presented FL-CMNN with the CMNN, artificial neural network(ANN), support vector machine(SVM) and K nearest neighbor(KNN) classifiers. Experimental results have revealed that the presented FL-CMNN yields better performance than the original CMNN model. It also achieved comparative classification accuracies with much lower computational cost than traditional ANN and SVM model.

Key words: mathematical morphology; morphological neural network (MNN); fuzzy lattice; pattern recognition

1 引言

形态神经网络(MNN)是数学形态学与传统的神经网络模型相结合的一种全新的非线性神经网络,其主要思想是采用形态学算子取代传统神经网络模型所采用的乘法、加法算子.与传统的神经网络模型相比,形态神经网络不仅计算简单,易于硬件实现,而且能在数步内达到稳定,不存在收敛问题^[1~10].随着近年来形态神经网络的发展,有学者提出了基于构造型的形态神经网络模型(CMNN),该模型能够自动生成中间层隐层节点,而且训练效率极高^[11~13].

然而,本文研究发现,CMNN在对落在超盒之外的测试样本进行分类时,计算网络最终输出所采用的形态学算

子不能正确地反映测试样本隶属于各超盒的程度,因此造成了分类精度的下降.针对此问题,本文提出了模糊格构造型形态神经网络模型(FL-CMNN),引入了模糊格包容性测度的概念^[14~17],在利用训练好的形态神经网络进行分类时,采用模糊格包容性测度计算测试样本属于各超盒的隶属度值,从而避免了CMNN采用形态学算子而带来的错分问题,有效提高了形态神经网络的分类精度.

2 形态神经网络的理论基础

形态神经网络是建立在格计算和数学形态学理论基础之上的一种新型的神经网络模型.本节将简要介绍完备格理论基础上的数学形态学理论和形态神经网络格代数系统^[4,12,13,18~20].

2.1 完备格框架下的数学形态学理论

假设 L 是一个非空集合, 如果对于任何元素 $x, y, z \in L$, 满足以下三个条件:

- (1) 自反性, $x \leq x$;
- (2) 反对称性, $x \leq y, y \leq x \Rightarrow x = y$;
- (3) 传递性, $x \leq y, y \leq z \Rightarrow x \leq z$,

则称 L 为一个偏序集. 如果 L 中的任意两个元素 x, y 都存在最小上界 $x \vee y$ 和最大下界 $x \wedge y$, 则 L 是一个格; 如果 L 的任意一个子集都存在最小上界和最大下界, 则称 L 是一个完备格. 通常, 用 O 和 I 来表示一个完备格的最小和最大元素. 如果 L 中的任意两个元素都是可比的, 则称 L 是一个全序格, 比如实数集 R 就是一个全序格.

假设 L_1, L_2, \dots, L_N 为 N 个完备格, 定义 L 为 N 个格的笛卡儿积: $L^N = L_1 \times L_2 \times \dots \times L_N$, 则 L^N 也是一个完备格, L^N 上的偏序关系定义为:

$$(x_1, \dots, x_N) \leq (y_1, \dots, y_N) \Leftrightarrow x_i \leq y_i, \dots, x_N \leq y_N \quad (1)$$

假设 L 和 M 为两个完备格, 称映射 $\varphi: L \rightarrow M$ 为格映射, 如果 φ 满足:

$$\varphi(x \wedge y) = \varphi(x) \wedge \varphi(y), \varphi(x \vee y) = \varphi(x) \vee \varphi(y) \quad (2)$$

则称 φ 为格同构映射.

基于完备格理论, 可定义如下形态学算子. 假设 L 和 M 为两个完备格, 如果对于任何指标集 $J, \{x_j\}_{j \in J} \in L$, 定义映射 $\varepsilon, \delta, \bar{\varepsilon}, \bar{\delta}: L \rightarrow M$, 则定义:

- ① ε 为腐蚀算子 $\Leftrightarrow \varepsilon(\bigwedge_{j=1}^J x_j) = \bigwedge_{j=1}^J \varepsilon(x_j), \quad x_j \in L$
- ② δ 为膨胀算子 $\Leftrightarrow \delta(\bigvee_{j=1}^J x_j) = \bigvee_{j=1}^J \delta(x_j), \quad x_j \in L$
- ③ $\bar{\varepsilon}$ 为反腐蚀算子 $\Leftrightarrow \bar{\varepsilon}(\bigwedge_{j=1}^J x_j) = \bigvee_{j=1}^J \bar{\varepsilon}(x_j), \quad x_j \in L$
- ④ $\bar{\delta}$ 为反膨胀算子 $\Leftrightarrow \bar{\delta}(\bigvee_{j=1}^J x_j) = \bigwedge_{j=1}^J \bar{\delta}(x_j), \quad x_j \in L$

显然膨胀和腐蚀都是同构映射, 如果同时反转 L 和 M 上的序关系, 那么膨胀就成为腐蚀, 腐蚀成为膨胀.

2.2 形态神经网络的格代数系统

传统的人工神经网络(ANN)可以通过其网络拓扑结构节点特性和训练或学习规则来描述, 传统神经网络的代数系统是实数集 \mathbf{R} 上的加法、乘法及相关运算规则, 可记之为 $(\mathbf{R}, \times, +)$, 其单个神经元的计算公式为:

$$\tau_j(x) = \sum_{i=1}^n x_i \times w_{ij} - \theta_j \quad (3)$$

其中 $\mathbf{x} \in \mathbf{R}^n$, x_i 代表第 i 个神经元的值, w_{ij} 表示第 i 个神经元与第 j 个神经元的连接权值, θ_j 表示第 j 个神经元的阈值, τ_j 表示第 j 个神经元的输出, 图 1 是一个

传统的具有 n 维输入的神经元模型.

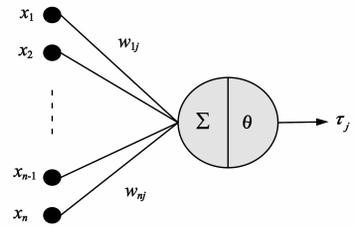


图1 传统神经网络的计算模型

形态神经网络是建立在格理论基础上的的一种新型神经计算方法, 其代数系统是实数集 \mathbf{R} 上的取大、取小、加法及相关运算规则, 记为格代数系统 $(G, \wedge, \vee, +, +')$, 其中 $G = \mathbf{R} \cup \{-\infty, +\infty\}$, \vee 代表取大算子, \wedge 代表取小算子, $+$ 和 $+$ ' 为定义在 G 上的加法算子, 二者只存在如下差别:

$$\infty + (-\infty) = (-\infty) + \infty = \infty \quad (4)$$

$$\infty + '(-\infty) = (-\infty) + ' \infty = -\infty$$

格代数系统中元素 x 的共轭 x^* 定义为:

$$x^* = \begin{cases} -x, & x \in G / \{-\infty, +\infty\} \\ -\infty, & x = +\infty \\ +\infty, & x = -\infty \end{cases} \quad (5)$$

则 G^n 上的负空间 v_* 可定义为:

$$(v_*(x))_i = x_i^*, i = 1, 2, \dots, n \quad (6)$$

格代数系统 $(G, \wedge, \vee, +, +')$ 的计算规则描述如下, 如果 $a, b, c \in \mathbf{R}_\infty$, 则:

$$\begin{aligned} a + (b \wedge c) &= (a + b) \wedge (a + c) \\ a \wedge \infty &= \infty \wedge a = a \\ a + \infty &= \infty + a = \infty \\ a + 0 &= 0 + a = a \end{aligned} \quad (7)$$

式(7)中只列出了取小算子 \wedge 的计算规则, 取大算子 \vee 的规则与取小算子 \wedge 完全一样.

传统神经计算的代数系统 $(\mathbf{R}, \times, +)$, 也满足相似的规则, 如果 $a, b, c \in \mathbf{R}$, 则有:

$$\begin{aligned} a \times (b + c) &= a \times b + a \times c \\ a + 0 &= 0 + a = a \\ a \times 0 &= 0 \times a = 0 \\ a \times 1 &= 1 \times a = 0 \end{aligned} \quad (8)$$

可以看出, 格代数系统中的 ∞ 类似于传统代数系统中的 0 , 0 类似于 1 , \wedge 类似于 $+$, 而 $+$ 类似于 \times .

除了上述列出的规则外, 这两种代数体系有很多其它的共同特征, 比如交换律和结合律. 从这意义上讲, 这两种体系在数学上是非常相似的, 所以, 在许多应用领域, 其中的一个很容易被另外一个所代替. 但是, 两者之间也存在着不同之处, 最主要的一点是取大、取小运算没有逆运算, 正是这种格运算使得形态神经元有着独一无二的特性.

与传统代数系统矩阵乘积类似,定义在有界格序组 G 上的矩阵运算算子分别称为最大积算子和最小积算子. 设矩阵 $A \in G^{m \times p}$ 和 $B \in G^{p \times n}$ 为定义在 G 上的两个矩阵, 则 A 和 B 的最大积 C 和最小积 D 定义为:

$$C = A \nabla B = \bigvee_{k=1}^p (a_{ik} + b_{kj}) \quad (9)$$

$$D = A \Delta B = \bigwedge_{k=1}^p (a_{ik} + b_{kj})$$

同理,可以定义矩阵与向量间的乘积,假设 $A \in G^{m \times p}$, $x \in G^p$, 分别定义 ϵ_A 和 δ_A 为:

$$\epsilon_A(x) = A \Delta x, \quad \delta_A(x) = A \nabla x \quad (10)$$

可以看出, ϵ_A 和 δ_A 分别定义了从格空间 G^p 到格空间 G^m 的腐蚀和膨胀算子.

同样,下式定义了从格空间 G^p 到格空间 G^m 的反腐蚀和反膨胀算子:

$$\bar{\epsilon}_A(x) = A \nabla x^*, \quad \bar{\delta}_A(x) = A \Delta x^* \quad (11)$$

3 构造型形态神经网络 (CMNN)

3.1 构造型形态神经网络的基本框架

形态学感知器是文献中首先出现的一种形态神经网络^[1], 只能实现二分类, 由于形态学神经计算函数是不可微的, 因此传统的反向传播算法不能用来训练形态神经网络. 因此, P. Sussner 提出了一种形态学感知器学习算法^[2], 并在此基础上进行了多次改进, 提出了一种新的构造型形态神经网络模型, 并实现了多类分类的功能^[11~13].

形态神经网络的基本操作算子计算是前述定义在格代数系统 G 上的四种形态学算子, 即 ϵ_w , δ_w , $\bar{\epsilon}_w$, $\bar{\delta}_w$. 给定输入向量 $x \in \mathbf{R}^n$, 连接权值向量为 $w \in \mathbf{R}_{\pm}^n$, 则第 j 个隐层节点的总输入可由如下公式计算:

$$\textcircled{1} \epsilon_w(x) = w \Delta x = \bigwedge_{i=1}^n (x_i + w_i)$$

$$\textcircled{2} \delta_w(x) = w \nabla x = \bigvee_{i=1}^n (x_i + w_i)$$

$$\textcircled{3} \bar{\epsilon}_w(x) = \delta_w(x) \circ v_*(x) = \bigwedge_{i=1}^n (x_i^* + w_i)$$

$$\textcircled{4} \bar{\delta}_w(x) = \epsilon_w(x) \circ v_*(x) = \bigvee_{i=1}^n (x_i^* + w_i)$$

形态神经网络训练过程为采用给定的训练样本来确定输入层与隐层节点之间的连接权值, 从而构造多个能够覆盖同类样本的 n 维超盒, 实现样本的正确分类. 值得注意的是, 定义在 G^n 上的超盒可完全由其覆盖元素的上顶点 a 和下顶点 b 所确定. 假设元素 x 包含于超盒 $[a, b]$, 则必满足 $a \leq x \leq b$. 令 $v = a^*$, $w = b$, 则上述条件可转化为:

$$\epsilon_v(x) \wedge \bar{\delta}_w(x) \geq 0 \quad (12)$$

针对二分类问题, 在训练过程中, 形态学神经计算自动产生只包含第一类样本的 m 个超盒 $[a^j, b^j]$ ($j = 1,$

$2, \dots, m$), 与之相对应的形态学神经计算为形态腐蚀算子 ϵ_{v_j} 和形态反膨胀算子 $\bar{\delta}_{w_j}$, 其中 $v_j = a^{j*}$, $w_j = b^j$. 在分类过程中, 给定一个输入样本 x , 假如 x 包含在超盒 $[a^j, b^j]$ 中, 则将 x 判别为第一类. 换言之, 形态学感知器的输出由下式决定:

$$y = f\left(\bigvee_{j=1}^m (\epsilon_{v_j}(x) \wedge \bar{\delta}_{w_j}(x))\right) \quad (13)$$

其中 $f(x)$ 为硬阈值函数:

$$f(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (14)$$

给定任意输入向量 x , 当且仅当 $y = 1$ 时, 将 x 分为第一类, 否则为第二类.

可以证明, $f(x)$ 是一个格同构映射 $G^n \rightarrow \{0, 1\}$, 因此, 公式(13)可重写为:

$$y = \bigvee_{j=1}^m (f(\epsilon_{v_j}(x)) \wedge f(\bar{\delta}_{w_j}(x))) \quad (15)$$

由前述基于完备格理论的形态学操作算子的定义可以证明, $f(\epsilon_{v_j}(x))$ 和 $f(\bar{\delta}_{w_j}(x))$ 亦为定义在格上的腐蚀算子和反膨胀算子. 基于构造型的形态神经网络的结构如图 2 所示.

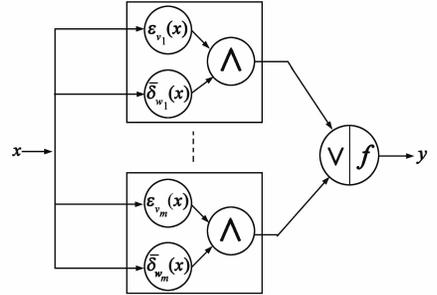


图2 构造型形态神经网络的结构图

给定一组训练样本集 $x^k \in \mathbf{R}^n$, 共包含两种类别 C^1 和 C^2 , 在训练过程中, 形态神经网络自动产生 m 个仅包含 C^1 样本的超盒, 其余的在超盒之外的样本均属于 C^2 类.

3.2 构造型形态神经网络训练算法

形态神经网络分类算法是建立在二分类基础之上的, 因此本节将重点阐述二分类的形态神经网络训练算法.

首先给出两个半空间 $H_i^-(c)$ 和 $H_i^+(c)$ 的定义, 假设 $x, c \in L^n$, 则:

$$H_i^-(c) = \{x \in L^n : x_i < c_i\} \quad (16)$$

$$H_i^+(c) = \{x \in L^n : x_i > c_i\}, i = 1, 2, \dots, n \quad (17)$$

假设训练集为 $X = \{x^1, x^2, \dots, x^k\} \in G^n$, 令 C^1 和 C^2 分别表示第一类和第二类训练样本. 形态神经网络训练的目的是找到两个超盒簇 $B^1 = \{[a_1^1, b_1^1], [a_2^1, b_2^1], \dots, [a_{m_1}^1, b_{m_1}^1]\}$ 和 $B^2 = \{[a_1^2, b_1^2], [a_2^2, b_2^2], \dots,$

$[\mathbf{a}_{m2}^2, \mathbf{b}_{m2}^2]$, 其中 \mathbf{B}^1 为包含所有第一类样本的超盒集合, \mathbf{B}^2 为包含所有第二类样本的超盒集合.

Sunsser 在文献^[12]中提出了一种构造型形态神经网络(CMNN)的训练方法, 本文方法与其类似, 但略有不同. 下面介绍本文提出的 CMNN 训练方法.

首先进行初始化, 令 $\mathbf{C} = \mathbf{X}, \mathbf{B}^1 = \Phi$, 执行如下操作步骤:

(1) 令 $\bar{\mathbf{C}}^1$ 表示集合 \mathbf{C} 中所有属于第一类的样本, 即 $\bar{\mathbf{C}}^1 = \mathbf{C} \cap \mathbf{C}^1$, 计算包含第一类样本的最小超盒:

$$[\bar{\mathbf{a}}^1, \bar{\mathbf{b}}^1] = [\wedge \bar{\mathbf{C}}^1, \vee \bar{\mathbf{C}}^1] \quad (18)$$

(2) 找出所有包含于超盒 $[\bar{\mathbf{a}}^1, \bar{\mathbf{b}}^1]$ 中的第二类样本集合, 记为 $\bar{\mathbf{C}}^2$;

(3) 如果 $\bar{\mathbf{C}}^2 = \Phi$, 则将 $[\bar{\mathbf{a}}^1, \bar{\mathbf{b}}^1]$ 加入到 \mathbf{B}^1 中去, 同时训练中止;

(4) 如果 $\bar{\mathbf{C}}^2 \neq \Phi$, 计算覆盖 $\bar{\mathbf{C}}^2$ 的最小超盒

$$[\bar{\mathbf{a}}^2, \bar{\mathbf{b}}^2] = [\wedge \bar{\mathbf{C}}^2, \vee \bar{\mathbf{C}}^2] \quad (19)$$

(5) 计算 $[\bar{\mathbf{a}}^2, \bar{\mathbf{b}}^2]$ 的半空间:

$$\mathbf{H}_i^-(\bar{\mathbf{a}}^2) = \{\mathbf{x} \mid x_i < \bar{a}_i^2\}, \mathbf{x} \in \bar{\mathbf{C}}^1, i = 1, 2, \dots, n \quad (20)$$

$$\mathbf{H}_i^+(\bar{\mathbf{b}}^2) = \{\mathbf{x} \mid x_i > \bar{b}_i^2\}, \mathbf{x} \in \bar{\mathbf{C}}^1, i = 1, 2, \dots, n \quad (21)$$

由此可以得到 $2n$ 个半空间 $\mathbf{H} = \{\mathbf{H}_1, \mathbf{H}_2, \dots, \mathbf{H}_{2n}\}$

(6) 如果 $\mathbf{H}_j \neq \Phi, j = 1, 2, \dots, 2n$, 则计算其超盒:

$$[\mathbf{a}_j^1, \mathbf{b}_j^1] = [\wedge \mathbf{H}_j, \vee \mathbf{H}_j], j = 1, 2, \dots, 2n \quad (22)$$

(7) 将超盒 $[\mathbf{a}_j^1, \mathbf{b}_j^1] (j = 1, 2, \dots, 2n)$ 加入到 \mathbf{B}^1 中去.

(8) 令 $\mathbf{C} = \bar{\mathbf{C}}^2$, 从步骤(1)继续.

图3给出了上述算法在二维空间上的示意图. 其中“ \circ ”代表第一类样本, “ $+$ ”代表第二类样本, 图中 $[\bar{\mathbf{a}}^1, \bar{\mathbf{b}}^1]$ 为第一类样本的超盒, $[\bar{\mathbf{a}}^2, \bar{\mathbf{b}}^2]$ 为包含在 $[\bar{\mathbf{a}}^1, \bar{\mathbf{b}}^1]$ 中第二类样本的超盒. 图中只给出了 $\bar{\mathbf{a}}^2$ 的半空间 $\mathbf{H}_1^-(\bar{\mathbf{a}}^2)$ 和 $\mathbf{H}_2^-(\bar{\mathbf{a}}^2)$, $[\mathbf{a}_1^1, \mathbf{b}_1^1]$ 和 $[\mathbf{a}_2^1, \mathbf{b}_2^1]$ 分别为两个半空间上的超盒, 可以看出 $[\mathbf{a}_1^1, \mathbf{b}_1^1]$ 和 $[\mathbf{a}_2^1, \mathbf{b}_2^1]$ 只包含了第一类样本; 同理, 可以计算 $\bar{\mathbf{b}}^2$ 的半空间及其只包含第

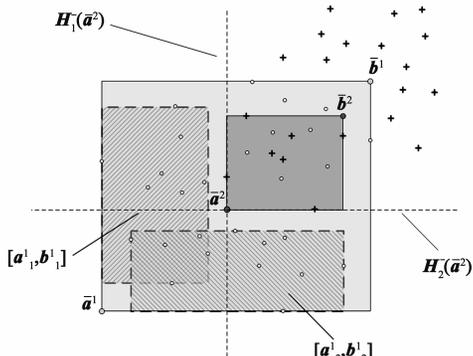


图3 CMNN训练算法示意图

一类样本的超盒, 并将其加入到 \mathbf{B}^1 中去.

在训练算法收敛后, 我们可以得到一个只包含第一类样本点的超盒簇 \mathbf{B}^1 . 为得到第二类样本点的超盒簇 \mathbf{B}^2 , 只需将两类样本的类别标签互换, 并重复上述训练算法一次即可.

3.3 多类分类构造型形态神经网络

前述的训练算法可非常容易的扩展至多类分类任务. 假如 \mathbf{X} 中包含了 S 类样本, 每类样本用 $\mathbf{C}^s (s = 1, 2, \dots, S)$ 表示. CMNN 的目的是构建 S 个超盒簇 $\mathbf{B}^1, \mathbf{B}^2, \dots, \mathbf{B}^S$, 其中第 s 个超盒簇 $\mathbf{B}^s (s = 1, 2, \dots, S)$ 只包含第 s 类样本. 在训练时, 我们只需令 $\mathbf{C}^1 = \mathbf{C}^s (s = 1, 2, \dots, S), \mathbf{C}^2 = \{\mathbf{C}^i, i \neq s, i = 1, 2, \dots, S\}$, 然后采用上述的训练算法即可得到第 s 类样本的超盒, 重复前述步骤 S 次, 即可得到 S 个包含各类样本的超盒簇.

在测试阶段, 给定一个测试样本 \mathbf{x} , 首先将 \mathbf{x} 带入第 s 个 CMNN 模型, 并计算其输出 $y_s (s = 1, 2, \dots, S)$:

$$y = \bigvee_{j=1}^m (\epsilon_{v_j}(\mathbf{x}) \wedge \delta_{w_j}(\mathbf{x})) \quad (23)$$

图4给出了基于第 s 个超盒簇的 CMNN 计算模型, 其中权重 $\mu_j^s = (\mathbf{a}_j^s)^*$, $w_j^s = \mathbf{b}_j^s, j = 1, 2, \dots, m_s$.

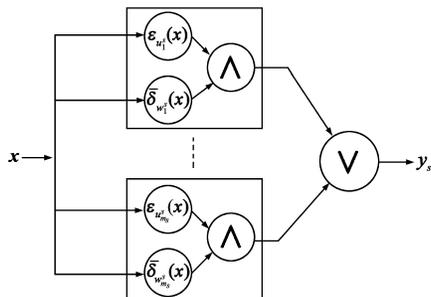


图4 多类分类中的第s个CMNN模型

计算出各模型的输出值 $y_s (s = 1, 2, \dots, S)$ 后, 根据图5给出的最大隶属度原则将 \mathbf{x} 归为输出值最大的一类.

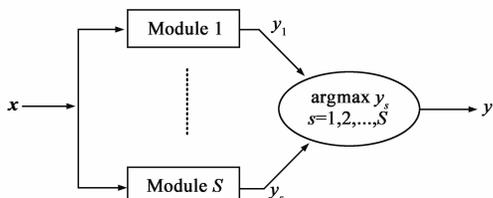


图5 CMNN多类分类规则

4 模糊格构造型形态神经网络

尽管前述的构造型形态神经网络具有很多传统神经网络所不具备的优势, 然而, 本文研究发现, 该构造型形态神经网络在对落在超盒之外的测试样本进行分类时, 计算网络最终输出所采用的形态学算子不能正

确地反映测试样本隶属于各超盒的程度,因此造成了分类精度的下降.针对此问题,本文提出了模糊格形态神经网络模型(FL-CMNN),引入了模糊格包容性测度的概念^[14-17],在利用训练好的形态神经网络进行分类时,计算测试样本属于各超盒的隶属度值,从而避免了原始形态神经网络采用形态学算子而带来的错分问题,有效提高了形态神经网络的分类精度.

下面首先介绍有关模糊格和区间上的模糊格理论.

4.1 模糊格理论

在完备格理论的基础上,V. Petridis^[21]提出了一个新的格学习框架——模糊格(fuzzy lattice),其定义为:

假设 L 为一个格, $S = \{(x, y) : x, y \in L\}$ 为一个分布, $\mu_p : S \rightarrow [0, 1]$ 是一个模糊隶属度函数,且满足当且仅当 $x \leq y$ 时 $\mu_p(x, y) = 1$, 则称 $(L, \mu_p(x, y))$ 为一个模糊格,其中实值函数 $\mu_p(x, y)$ 可以解释为 x 包含于 y 的程度.

模糊格是对传统格理论的拓展,它考虑了格中两个元素之间的关系,为定量描述模糊格中两个元素之间的包含关系,V. Petridis 提出了包容性测度的概念,其定义为:

假设 L 为一个完备格,其最小元素与最大元素分别为 0 和 1 , 则包容性测度定义为 $\sigma(x, y) : S = \{(x, y) : x, y \in L\} \rightarrow [0, 1]$, 且 $\sigma(x, y) \equiv \sigma(x \leq y)$ 满足以下条件:

- (1) $\sigma(u \leq 0) = 0, u \neq 0$
- (2) $\sigma(u \leq u) = 1, u \in L$
- (3) $u \leq w \Rightarrow \sigma(x \leq u) \leq \sigma(x \leq w), u, w, x \in L$

由条件(2)和(3)可以很容易地得出以下结果: $u \leq w \Rightarrow 1 = \sigma(u \leq u) \leq \sigma(u \leq w) \leq 1 \Rightarrow \sigma(u \leq w) = 1$, 进一步可以得出 $\sigma(u \leq 1) = 1$. 因此, $\sigma(x \leq u)$ 表示了格关系 $x \leq u$ 的真实程度,即 x 包容于 u 的程度.

为了进一步量化包容性测度,定义实值函数 $h : L \rightarrow \mathbf{R}$, 它主要满足以下几个条件:

- (1) $h(0) = 0, 0$ 为格 L 中的最小元素;
- (2) $u < w \Rightarrow h(u) < h(w)$;
- (3) $u \leq w \Rightarrow h(x \vee w) - h(x \vee u) \leq h(w) - h(u)$

函数 h 在格 L 上不一定一定存在,假如函数 h 存在,则函数 $k(x, y) = h(y)/h(x \vee y)$ 是一个定义在 L 上的包容性测度.可以看出, $x \leq u \Leftrightarrow u = x \vee u \Leftrightarrow k(x \leq u) = h(u)/h(x \vee u) = 1$, 因此, $(L, k(x \leq u))$ 定义了一个模糊格.在本文中,简单地设函数 h 为: $h(x) = x$.

假设 L 为 N 个格的笛卡儿积: $L = L_1 \times L_2 \times \cdots \times L_N$, 其偏序关系定义为:

$$(x_1, \cdots, x_N) \leq (y_1, \cdots, y_N) \Leftrightarrow x_1 \leq y_1, \cdots, x_N \leq y_N \quad (24)$$

则在 L 上的函数 h 可定义为:

$$h(x_1, \cdots, x_N) = h(x_1) + \cdots + h(x_N) \quad (25)$$

可以证明 $h(x_1, \cdots, x_N)$ 满足前述的三个条件.

4.2 区间上的模糊格理论

设 L 是一个全序格,则定义在 L_i 上的一个区间 $V_L = \{[a, b] : a, b \in L\}$ 是 L 的一个子集,称之为区间格,其交与并分别定义为: $[a, b] \wedge [c, d] = [a \vee c, b \wedge d]$, $[a, b] \vee [c, d] = [a \wedge c, b \vee d]$, 相应的偏序关系即为如果在 $\tau(L_i)$ 上有 $[a, b] \leq [c, d]$, 则 $a \geq c$ 且 $b \leq d$.

由于前述的包容性测度是定义在传统的格 L 上的,而关于区间格的包容性测度是没有定义的.然而,直接在区间格 V_L 上定义评价函数是不可能的,因此需要将区间格转换到传统格来定义评价函数.我们需要找到一个将区间格转换到格的映射 $\psi : V_L \rightarrow L \times L$.

引入 L 上的对偶映射 θ , 其定义为 $\theta : L \rightarrow L$ 且满足 $x \leq y \Leftrightarrow \theta(x) \geq \theta(y)$, 则可定义映射 $\psi : [a, b] \in V_L \rightarrow (\theta(a), b) \in L \times L$. 根据这个映射,假如 V_L 上的两个元素满足关系 $[a, b] \leq [c, d] \Rightarrow a \geq c, b \leq d$, 则在 $L \times L$ 满足 $(\theta(a), b) \leq (\theta(c), d)$, 因此区间格 V_L 上的包容性测度定义为:

$$\begin{aligned} \sigma([a, b] \leq [c, d]) &= \sigma((\theta(a), b) \leq (\theta(c), d)) \\ &= \frac{h(\theta(a), b)}{h((\theta(a), b) \vee (\theta(c), d))} \\ &= \frac{h\theta(a) + h(b)}{h((\theta(a) \vee \theta(c)) + h(b \vee d))} \end{aligned} \quad (26)$$

下面举例来说明上述的计算过程,假设 $I = [0, 1]$ 为一个全序格,其最大元素为 1, 最小元素为 0, 定义 $\theta(x) = 1 - x$ 为格 I 上的对偶映射函数,可以看出如果 $a \leq b (a, b \in I)$, 则 $\theta(a) \geq \theta(b) (\theta(a), \theta(b) \in I)$.

设 $[0.2, 0.4]$ 和 $[0.3, 0.6]$ 是区间格上的两个元素,即两个格区间,则映射到传统格上分别为 $(1 - 0.2, 0.4) = (0.8, 0.4) \in I \times I$ 和 $(1 - 0.3, 0.6) = (0.7, 0.6) \in I \times I$, 则区间 $[0.2, 0.4]$ 包容于区间 $[0.3, 0.6]$ 的测度为:

$$\begin{aligned} \sigma([0.2, 0.4] \leq [0.3, 0.6]) &= \sigma((\theta(0.2), 0.4) \leq (\theta(0.3), 0.6)) \\ &= \sigma((0.8, 0.4) \leq (0.7, 0.6)) \\ &= \frac{h(0.7, 0.6)}{h((0.8, 0.4) \vee (0.7, 0.6))} \\ &= 0.929 \end{aligned}$$

同理,区间 $[0.3, 0.6]$ 包容于区间 $[0.2, 0.4]$ 的测度为 $\sigma([0.3, 0.6] \leq [0.2, 0.4]) = 0.857$.

将区间推广至二维空间,即单位盒 $U = [0, 1] \times [0, 1]$, 假设单位盒中的两个区域为 $[0.1, 0.5] \times [0.2, 0.3]$ 和 $[0.3, 0.6] \times [0.1, 0.6]$, 其在 V_U 中表示为两个元素 $[0.1, 0.5, 0.2, 0.3]$ 和 $[0.3, 0.6, 0.1, 0.6]$, 映射到格上即为

(0.9,0.5,0.8,0.3)和(0.7,0.6,0.9,0.6),则区域 $[0.1,0.5] \times [0.2,0.3]$ 包含于区域 $[0.3,0.6] \times [0.1,0.6]$ 的测度为:

$$\begin{aligned} \sigma([0.1,0.5,0.2,0.3] \leq [0.3,0.6,0.1,0.6]) \\ = \frac{h(0.7,0.6,0.9,0.6)}{h((0.9,0.5,0.8,0.3) \vee (0.7,0.6,0.9,0.6))} \\ = \frac{2.8}{3} \approx 0.933 \end{aligned}$$

同理区域 $[0.3,0.6] \times [0.1,0.6]$ 包含于区域 $[0.1,0.5] \times [0.2,0.3]$ 的测度为: $\sigma([0.3,0.6,0.1,0.6] \leq [0.1,0.5,0.2,0.3]) = 2.5/3 \approx 0.833$.

上述的计算方法可以很容易地推广至 N 维空间,在 N 维空间中,单位盒变为超立方体.

4.3 模糊格构造型形态神经网络

由 3.2 节的形态神经网络训练方法可知,形态神经网络生成了一系列包含各类别样本的超盒,原始的形态神经网络采用式(23)计算测试样本落入到各超盒内的隶属度值,然而,本文研究发现,对于落在超盒之外的测试样本,CMNN 分类效果较差,造成分类效果下降,因此,本文引入前述的模糊格包容性测度来计算测试样本隶属于各超盒的隶属度值,然后采用最大隶属度原则进行分类,避免了原始 CMNN 的缺陷.

前述的包容性测度是定义在两个区间上的,对于本文的分类问题来说给定的样本是空间中的一个点 $x \in G^n$,因此在计算一个点包容于一个区间的测度时,采用如下方法将 x 扩展为一个区间.以三维空间为例,假设 $x \in G^3$,则其区间为:

$$I(x) = [x_1, x_1, x_2, x_2, x_3, x_3] \quad (27)$$

同样,对于 CMNN 训练所得到的超盒也需要做一定的转换来计算包容性测度,给定超盒 $[a, b] \in G^3$,其区间为:

$$I[a, b] = [a_1, b_1, a_2, b_2, a_3, b_3] \quad (28)$$

图 6 给出了在二维空间上计算模糊格包容性测度的示意图.其中 $x_1 = [0.2,0.3]$, $x_2 = [0.3,0.4]$,两个超盒 $w_1 = [0.1,0.2,0.4,0.35]$, $w_2 = [0.5,0.2,0.8,0.8]$.

则 x_1 包容于超盒 w_1 的测度 $\sigma(I(x_1) \leq I(w_1))$ 为:

$$\begin{aligned} \sigma([0.2,0.2,0.3,0.3] \leq [0.1,0.4,0.2,0.35]) \\ = \frac{h(0.9,0.4,0.8,0.35)}{h((0.8,0.2,0.7,0.3) \vee (0.9,0.4,0.8,0.35))} \\ = \frac{2.45}{2.45} = 1 \end{aligned}$$

同理 x_1 包容于超盒 w_2 的测度为:

$$\begin{aligned} \sigma([0.2,0.2,0.3,0.3] \leq [0.5,0.8,0.2,0.8]) \\ = \frac{h(0.5,0.8,0.8,0.8)}{h((0.8,0.2,0.7,0.3) \vee (0.5,0.8,0.8,0.8))} \\ = \frac{2.9}{3.2} \approx 0.9062 \end{aligned}$$

由上述结果可以看出,当测试样本落入到某个超

盒之内时,该样本包容于该超盒的测度为最大值 1,而包容于其它超盒的测度小于 1.

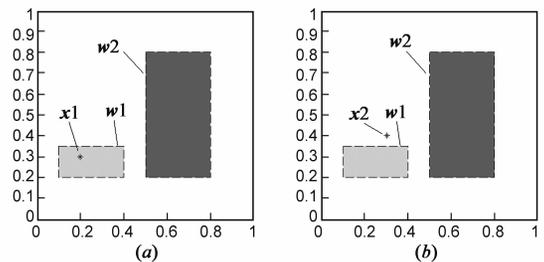


图6 FL-CMNN中测试样本包容性测度计算示例

下面考虑测试样本在所有超盒之外的情况,如图 6 (b)所示,给定样本 $x_2 = [0.3,0.4]$,可以看出 x_2 落在两个超盒之外,采用上述的方法分别计算 x_2 相对于超盒 B_1 和 B_2 的测度分别为 $\sigma(I(x_2) \leq I(w_1)) = 0.98$ 和 $\sigma(I(x_2) \leq I(w_2)) = 0.9355$.由此可以看出,样本离超盒越近,则相对于该超盒的包容性测度越大.

FL-CMNN 的训练过程与 CMNN 一样,两者最大的不同在于测试阶段,FL-CMNN 计算输入测试样本的单个 CMNN 模型输出值方法如图 7 所示,可以看出,图 7 与图 4 的结构非常类似,但两者最大的区别在于给定输入样本 x ,计算各超盒的输出时所采用的计算方法不同,CMNN 采用的是式(23)所定义的形态学算子,而 FL-CMNN 采用的是定义在模糊格理论上的包容性测度.

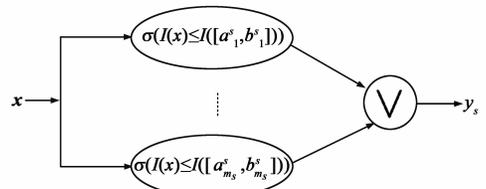


图7 多类分类中的第 m 个 FL-CMNN 模型

计算出输入样本隶属于每类超盒簇的包容性测度后,依然根据图 5 中的最大隶属度原则进行分类.

5 实验分析

为验证本文提出的 FL-CMNN 的有效性,本节采用标准数据集和仿真数据集对 FL-CMNN、CMNN 以及传统神经网络、 K 近邻和支持向量机的计算效率和分类精度进行对比试验分析.这三种分类器均在 Matlab 上采用模式识别工具箱(PRTTools 4.1)^[22]实现,这三种分类器的参数均为工具箱预设的缺省值.试验所采用的计算机配置为 CPU Core2 2.8G, 2G 内存, Matlab 版本为 R2007a.

5.1 Ripley 数据集

Ripley 数据集^[23]是由混合高斯分布产生的一个两类二维数据集,其中训练样本集由 250 个数据组成,每

类 125 个数据,测试样本集由 1000 个数据组成,每类 500 个数据.

图 8 给出了采用构造型形态神经网络训练后所得到的超盒.图中“+”代表第一类样本,“·”代表第二类样本;深色背景的超盒为第一类样本的超盒,浅色背景为第二类样本的超盒.在训练两类样本的超盒过程中,本文提出的训练算法分别在 8 步和 9 步内达到收敛,并相应生成了 32 和 33 个超盒来覆盖两类样本.从图中可以看出,所有的训练样本均可被正确地覆盖.需要注意的是,图中存在着部分的孤点,这些孤点也是一种特殊的超盒.

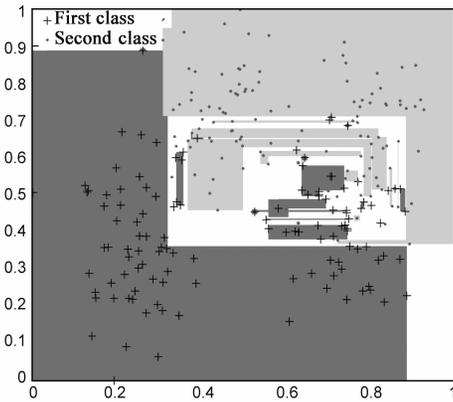


图8 CMNN模型在Ripley数据集上生成的超盒

表 1 给出了分别采用 K 近邻分类器(KNNC)、传统的 BP 神经网络(BP-NN)、支持向量机(SVM)、构造型神经网络(CMNN)和模糊格构造型神经网络(FL-CMNN)的训练时间(T_{train})、测试时间(T_{test})、训练样本分类精度(P_{train})和测试样本分类精度(P_{test}).

表 1 Ripley 数据集测试结果

分类器	T_{train} (s)	T_{test} (s)	P_{train}	P_{test}
KNNC	-	0.3687	87.2	80.8
BP-NN	24.4906	0.1053	86.8	81.2
SVM	12.9543	0.1389	85.6	86.1
CMNN	0.0774	0.1662	100	78.3
FL-CMNN	0.0774	0.2843	100	81.7

由表 1 可以看出,在 Ripley 数据集上,SVM 获得了最高的测试精度 86.1%,其次为 FL-CMNN,测试精度为 81.7%.但是从表中可以发现,构造型形态神经网络的训练效率极高,其所需的训练时间要远远低于传统的 BP-NN 和 SVM,同时可以发现,两种形态神经网络在训练集上的分类精度均为 100%,这说明形态神经网络具有很强的学习能力,而 FL-CMNN 的推广能力要明显优于 CMNN.

5.2 Spiral 数据集

Spiral 数据集^[24]是一个经典的二分类问题,它由以

下公式所生成:

$$R_i = 0.4 \cdot (105 - n_i) / 104 \quad (29)$$

$$\alpha_i = \pi \cdot (n_i - 1) / 16 \quad (30)$$

$$x_i = R_i \cdot \sin(\alpha_i + q \cdot \text{randn}(1,1) / 100) + 0.5 \quad (31)$$

$$y_i = R_i \cdot \cos(\alpha_i + q \cdot \text{randn}(1,1) / 100) + 0.5 \quad (32)$$

其中 $n_i = 0.5i$ ($i = 1, 2, \dots, 200$), $\text{randn}(1,1)$ 为由标准高斯分布产生的随机数, q 是一个调节类间距离的参数,本文中设 $q = 5$.数据集中第一类样本由 $(x_i, y_i)_{i=1}^{200}$ 组成,第二类样本由 $(1 - x_i, 1 - y_i)_{i=1}^{200}$ 组成,由此可得到 400 个训练样本.同理,可由公式 29 ~ 32 产生 400 个测试样本.

图 9 给出了由本文提出的 CMNN 训练算法在 Spiral 数据集上所得到的超盒.图中“+”代表第一类样本,“·”代表第二类样本;深色背景的超盒为第一类样本的超盒,浅色背景为第二类样本的超盒.在训练中,算法均在 13 步内达到收,并均生成了 48 个超盒来覆盖两类样本.

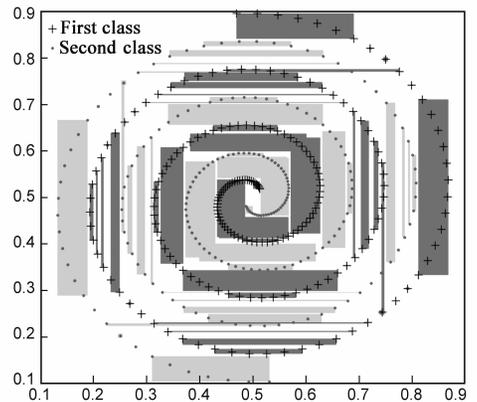


图9 CMNN模型在Spiral数据集上生成的超盒

表 2 给出了分别采用 K 近邻分类器(KNNC)、传统的 BP 神经网络(BP-NN)、支持向量机(SVM)、构造型神经网络(CMNN)和模糊格构造型神经网络(FL-CMNN)在 Spiral 数据集上的训练时间(T_{train})、测试时间(T_{test})、训练样本分类精度(P_{train})和测试样本分类精度(P_{test}).

表 2 Spiral 数据集测试结果

分类器	T_{train} (s)	T_{test} (s)	P_{train}	P_{test}
KNNC	-	0.1282	77.5	74.8
BP-NN	14.176	0.1324	75.25	74
SVM	5.916	0.1285	82	81.75
CMNN	0.1366	0.3352	100	90.75
FL-CMNN	0.1366	0.4843	100	98.5

由表 2 可以看出,在 Spiral 数据集上,本文提出的 FL-CMNN 取得了明显优于其它方法的测试精度和训练精度.传统的 KNN、BP-NN 和 SVM 在此数据集上无论是训练精度还是测试精度均较差,虽然 CMNN 取得了很

高的训练精度,但测试精度明显低于 FL-CMNN,这再次证明了 FL-CMNN 的推广能力优于 CMNN.同时从表中依然可以发现,构造型形态神经网络的训练效率极高,其所需的训练时间要远远低于传统的 BP-NN 和 SVM.

5.3 仿真数据集

图 10 为随机产生的三组数据,其中“·”代表第一类样本,“*”代表第二类样本,“+”代表第三类样本,每类数据 100 个样本点,共 300 个样本点.图中三种不同颜色的区域代表了采用本文提出的训练算法得到的覆盖三类样本的超盒集.可以看出,训练出的超盒可完全覆盖所有的样本点.在训练三类样本的超盒过程中,本文提出的训练算法分别在 3 步,8 步和 4 步内达到收敛,并相应生成了 8,25 和 12 个超盒来覆盖三类样本.

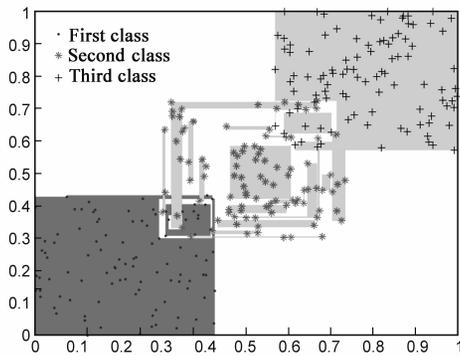


图10 CMNN模型在仿真数据集上生成的超盒

在测试阶段,依然按训练样本相同的条件随机生成三组数据,每类数据 100 个样本点.为保证测试结果的可靠性,本文将前述的训练和测试过程重复 20 次,即每次均随机产生 300 个训练样本和 300 个测试样本,取 20 次结果的平均值作为最终测试结果.

表 3 给出了前述五种分类器在仿真数据集上的训练时间 (T_{train})、测试时间 (T_{test})、训练样本分类精度 (P_{train}) 和测试样本分类精度 (P_{test}).

表 3 仿真数据集测试结果

分类器	T_{train} (s)	T_{test} (s)	P_{train}	P_{test}
KNNC	-	0.0952	94.33	88.37
BP-NN	24.37	0.0310	96	91
SVM	12.07	0.0175	92.67	91.67
CMNN	0.0786	0.0662	100	87.33
FL-CMNN	0.0786	0.0843	100	92.67

由表 3 可以看出,CMNN 的训练时间只有支持向量机的 1/100,传统 BP 神经网络训练时间的 1/200 左右;在测试阶段,两种构造型神经网络的计算时间要略高于 BP-NN 和 SVM,与 KNNC 的测试时间相当.在训练样本的分类精度上,两种构造型神经网络均可达到

100%;在测试阶段,传统的支持向量机和 BP 神经网络均获得了比 CMNN 高的分类精度,而本文提出的 FL-CMNN 仍然获得了最高的分类精度.

6 结论

(1)深入分析了构造型形态神经网络的结构特点,在文献中已有方法的基础上提出了一种构造型形态神经网络的训练算法,该算法能够在数步内达到收敛,具有极高的训练效率.

(2)针对构造型形态神经网络所采用的形态学算子不能正确地反映测试样本隶属于各超盒程度的问题,本文提出了一种模糊格构造型形态神经网络模型 FL-CMNN,该模型引入了模糊格包容性测度的概念,在利用训练好的形态神经网络进行分类时,计算测试样本属于各超盒的隶属度值,从而避免了原始形态神经网络决策函数的不足,有效提高了形态神经网络的分类精度.

(3)采用标准数据集对本文提出的 CMNN 和 FL-CMNN 模型进行了评价,并与传统的最近邻分类器、人工神经网络和支持向量机进行了对比,结果表明,本文提出的 FL-CMNN 在测试精度上明显优于原始的 CMNN,训练时间仅仅是传统的神经网络和支持向量机的数百分之一,而且分类精度也丝毫不亚于传统的神经网络和支持向量机.

参考文献

- [1] G X Ritter, P Sussner. An introduction to morphological neural networks[A]. International Conference on Pattern Recognition [C]. Vienna, IEEE Xplore, 1996. 709 - 717.
- [2] P Sussner. Morphological perceptron learning[A]. IEEE International Symposium on Intelligent Control Proceedings[C]. Gaithersburg, MD, IEEE Xplore, 1998. 477 - 482.
- [3] M Grana, B Raducanu. Some applications of morphological neural networks[A]. Proceedings of the International Joint Conference on Neural Networks[C]. Washington, DC, IEEE Xplore, 2001. 4:2518 - 2523.
- [4] G X Ritter, G Urcid. Lattice algebra approach to single-neuron computation[J]. IEEE Transactions on Neural Networks, 2003, 14(2): 282 - 295.
- [5] P Sussner, M E Valle. Gray-scale morphological associative memories[J]. IEEE Transactions on Neural Networks, 2006, 17(3): 559 - 570.
- [6] I Villaverde, M Grana, A d' Anjou. Morphological neural networks and vision based mobile robot navigation[A]. Lecture Notes in Computer Science(including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics [C]. Berlin, Springer Berlin Heidelberg, 2006. 878 - 887.

- [7] C W Han. Range facial recognition with the aid of eigenface and morphological neural networks [A]. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) [C]. Berlin, Springer Berlin Heidelberg, 2008. 217 – 224.
- [8] A M da Silva, P Sussner. A brief review and comparison of feedforward morphological neural networks with applications to classification[A]. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)[C]. Berlin, Springer Berlin Heidelberg, 2008. 783 – 792.
- [9] 王敏, 王士同, 吴小俊. 新模糊形态学联想记忆网络的初步研究[J]. 电子学报, 2003, 31(5): 690 – 693.
Wang Min, Wng Shi tong, Wu Xiao jun. Initial results on fuzzy morphological associative memories[J]. Acta Electronica Sinica, 2003, 31(5): 690 – 693. (in Chinese)
- [10] 景晓军, 余农. 基于形态学变权神经网络的数据精炼[J]. 电子学报, 2005, 33(3): 397 – 401.
Jing Xiao-jun, Yu Nong. Data abstraction based on morphological adjusted-weight neural network [J]. Acta Electronica Sinica, 2005, 33(3): 397 – 401. (in Chinese).
- [11] P Sussner, E L Esmi. An introduction to morphological perceptrons with competitive learning[A]. Proceedings of the International Joint Conference on Neural Networks[C]. Atlanta, GA, IEEE Xplore, 2009. 3024 – 3031.
- [12] P Sussner, E L Esmi. Constructive morphological neural networks: Some theoretical aspects and experimental results in classification[A]. Studies in Computational Intelligence[C]. Berlin, Springer Berlin Heidelberg, 2009. 258/2009; 123 – 144.
- [13] P Sussner, E L Esmi. Morphological perceptrons with competitive learning: Lattice-theoretical framework and constructive learning algorithm[J]. Information Sciences, 2011, 181(10): 1929 – 1950.
- [14] V Petridis, V G Kaburlasos. Learning in the framework of fuzzy lattices[J]. IEEE Transactions on Fuzzy Systems, 1999, 7(4): 422 – 440.
- [15] V G Kaburlasos, V Petridis. Fuzzy lattice neurocomputing (FLN) models[J]. Neural Networks, 2000, 13(10): 1145 – 1170.
- [16] V G Kaburlasos, I N Athanasiadis, P A Mitkas. Fuzzy lattice reasoning (FLR) classifier and its application for ambient ozone estimation[J]. International Journal of Approximate Reasoning, 2007, 45(1): 152 – 188.
- [17] V G Kaburlasos, S E Papadakis. A granular extension of the fuzzy-ARTMAP (FAM) neural classifier based on fuzzy lattice reasoning (FLR)[J]. Neurocomputing, 2009, 72(10-12): 2067 – 2078.
- [18] E R Dougherty, D Sinha. Computational gray-scale mathematical morphology on lattices (A comparator-based image algebra) Part 1: Architecture[J]. Real-Time Imaging, 1995, 1(1): 69 – 85.
- [19] P Maragos. Lattice image processing: A unification of morphological and fuzzy algebraic systems[J]. Journal of Mathematical Imaging and Vision, 2005, 22(2): 333 – 353.
- [20] M Nachtgaeel, P Sussner, T M Ullange, E E Kerre. On the role of complete lattices in mathematical morphology: From tool to uncertainty model[J]. Information Sciences, 2011, 181(10): 1971 – 1988.
- [21] V Petridis, V G Kaburlasos. Fuzzy Lattice Neural Network (FLNN): A hybrid model for learning[J]. IEEE Transactions on Neural Networks, 1998, 9(5): 877 – 890.
- [22] PRTOOLS 4.20[EB/OL]. <http://prtools.org>, 2001.
- [23] B D Ripley. Pattern Recognition and Neural Networks[M]. Cambridge University Press, 1996.
- [24] K J Lang, D J Witbrock. Learning to tell two spirals apart [A]. Proceedings of Connectionist Models Summer School [C]. Mountain View, CA: Morgan Kaufmann. 1989. 52 – 59.

作者简介



李 兵 男, 1982 年 9 月出生于湖北随州, 军械工程学院讲师, 主要研究方向为信号处理与模式识别, 状态监测与故障诊断。