

# 模糊神经网络的二阶段变半径 随机搜索算法

安凯 邢进生<sup>1</sup> 邱祖廉

(西安交通大学自动控制系, 系统工程所<sup>1</sup> 西安 710049)

(E-mail: jsxing@xjtu.edu.cn)<sup>1</sup>

**摘要** 研究基于正态隶属函数的模糊神经网络的学习算法. 将模糊神经网络对一组样本的逼近误差表示为两组相互独立, 可分批学习的可调参数的非负函数之和. 其中一组可调参数可通过令相应的非负函数为零直接求得, 而与另一组可调参数相对应的非负函数就是用于这组参数学习的性能指标. 经对性能指标性质的分析给出了一种模糊神经网络的学习算法——二阶段变半径随机搜索法. 实例表明, 这种方法简便易行, 可使模糊神经网络达到较高的逼近精度.

**关键词** 二阶段变半径随机搜索法, 性能指标, 正态函数, 伪逆矩阵.

## A TWO-STAGE RANDOM SEARCH ALGORITHM OF FUZZY NEURAL NETWORK WITH VARIABLE RADIUS

AN Kai XING Jinsheng QIN Zulian

(Department of Automatic Control, Systems Engineering Institute,  
Xi'an Jiaotong University, Xi'an 710049)

**Abstract** Taking normal functions as membership functions of fuzzy variables the approaching error of a fuzzy neural network to a group of samples is denoted as the sum of two nonnegative functions of two independent and adjustable groups of parameters that can be trained one after another. One of the two parameter groups can be obtained directly by taking its corresponding nonnegative functions to be zero and another parameter group can be obtained through learning according to its corresponding nonnegative functions--performance index. Based on the analysis of the performance index a new algorithm, two-stage random search algorithm with variable radius, is put forward. Some examples show that the algorithm is simple and convenient and can make fuzzy neural network attain high precision.

**Key words** Two-stage random search algorithm with variable radius, performance index, normal function, pseudoinverse.

## 1 引言

目前模糊神经网络在理论和应用方面均已取得较大进展,但其学习算法似乎始终没有脱离梯度算法的思想束缚.复杂的公式,无休止的迭代、局部极小的困扰等一系列问题随着梯度算法的使用接踵而来,成为模糊神经网络学习中的一大弊端.因此找到一种收敛速度快、逼近能力强、运算简便、且可避免落入局部最小误区的模糊神经网络学习算法已成为模糊神经网络研究中的一个重要课题,这一问题的解决对推动模糊神经网络的进展具有极其深远的意义.

在模糊系统中,描述输入输出间关系的第  $i$  条模糊规则为

$$R_i: \text{if } x_1 \text{ is } A_1^{i_1} \text{ and } x_2 \text{ is } A_2^{i_2}, \dots \text{ and } x_n \text{ is } A_n^{i_n}, \text{ then } y \text{ is } B^i,$$

其中,  $i_k \in \{1, 2, \dots, m_k\}, k=1, 2, \dots, n; m = \prod_{i=1}^n m_i, i=1, 2, \dots, m; A_k^{i_k}, B^i$  为模糊集.本文中约定,规则序号  $i$  与模糊集上标号  $i_1, i_2, \dots, i_n$  的对应关系是:将  $\{1, 2, \dots, m_k\}$  视为有序数组,依  $k$  的大小按“字典排列法”排成  $m$  个数组,并依次与  $1, 2, \dots, m$  对应.于是对任何的输入向量  $\mathbf{x} = (x_1, x_2, \dots, x_n)$ ,若记

$$\mu_{A_j^{i_j}}(x_j) = \exp\left\{-\frac{(x_j - c_{j_i_j})^2}{\sigma_{j_i_j}}\right\} \hat{=} \hat{\mu}_j^{i_j}(x_j),$$

则第  $i$  条规则的适用度为  $\alpha_i(\mathbf{x}) = \mu_1^{i_1}(x_1)\mu_2^{i_2}(x_2)\cdots\mu_n^{i_n}(x_n)$ . 通过模糊推理并采用加权平均去模糊的方法可求得输出的清晰量为

$$y = \sum_{i=1}^m w_i \bar{\alpha}_i(\mathbf{x}),$$

其中  $\bar{\alpha}_i(\mathbf{x}) = \frac{\alpha_i(\mathbf{x})}{\sum_{i=1}^m \alpha_i(\mathbf{x})}$ ,  $w_i$  是模糊集  $B^i$  的隶属函数的中心值.

将上面的单输入单输出的系统推广为多输入多输出的系统,有<sup>[1]</sup>

$$\mathbf{y} = W\bar{\boldsymbol{\alpha}}(\mathbf{x}), \quad (1)$$

其中

$$\mathbf{y} = (y_1, y_2, \dots, y_r)^T, \bar{\boldsymbol{\alpha}}_m(\mathbf{x}) = (\bar{\alpha}_1(\mathbf{x}), \bar{\alpha}_2(\mathbf{x}), \dots, \bar{\alpha}_m(\mathbf{x}))^T,$$

$$W = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1m} \\ w_{21} & w_{22} & \cdots & w_{2m} \\ \cdots & & & \\ w_{r1} & w_{r2} & \cdots & w_{rm} \end{bmatrix}. \quad (2)$$

然而,当一组样本  $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) (i=1, 2, \dots, N)$  给定时,如何选择参数  $C_{ki_k}, \sigma_{ki_k}$ ,使模糊神经网络更精确地拟合这组样本以及“精确”的标准将由下文给出.

## 2 性能指标

对于上述模糊系统  $\mathbf{y} = W\bar{\boldsymbol{\alpha}}(\mathbf{x})$  及给定的样本  $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) (k=1, 2, \dots, N)$ ,训练网络的

目的是找到权矩阵和参数,使

$$\sum_{k=1}^N \|y^{(k)} - W\bar{\alpha}(x^{(k)})\|^2 = \text{tr}\{(y^{(1)} - W\bar{\alpha}(x^{(1)}), \dots, y^{(N)} - W\bar{\alpha}(x^{(N)}))^\top (y^{(1)} - W\bar{\alpha}(x^{(1)}), \dots, y^{(N)} - W\bar{\alpha}(x^{(N)}))\} \quad (3)$$

尽可能小,其中 $\|\cdot\|$ 表示欧几里得模, $x^{(k)} = (x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)})^\top$ ,记

$$A = (\bar{\alpha}(x^{(1)}), \bar{\alpha}(x^{(2)}), \dots, \bar{\alpha}(x^{(N)})), Y = (y^{(1)}, y^{(2)}, \dots, y^{(N)}),$$

则式(3)的右端可改写为

$$\text{tr}\{(Y - WA)^\top (Y - WA)\} = \text{tr}\{(Y - WA)(Y - WA)^\top\}.$$

用 $A^+$ 表示 $A$ 的伪逆矩阵则有

$$A^+ AA^\top = A^\top, \quad AA^\top(A^+)^\top = A.$$

因此

$$\begin{aligned} (Y - WA)(Y - WA)^\top &= (Y - WAA^\top(A^+)^\top)(Y^\top - A^+ AA^\top W^\top) = \\ &= YY^\top - YA^+ AA^\top W^\top - WAA^\top(A^+)^\top Y^\top + WAA^\top(A^+)^\top A^+ AA^\top W^\top = \\ &= Y(I - A^+ A)Y^\top + (YA^+ - W)AA^\top(YA^+ - W)^\top, \end{aligned}$$

因此

$$\begin{aligned} \text{tr}\{(Y - WA)(Y - WA)^\top\} &= \\ &= \text{tr}\{Y(I - A^+ A)Y^\top\} + \text{tr}\{(YA^+ - W)AA^\top(YA^+ - W)^\top\}. \end{aligned} \quad (4)$$

注意到

$$(YA^+ - W)AA^\top(YA^+ - W)^\top = (YA^+ A - WA)(YA^+ A - WA)^\top,$$

为一非负定矩阵,其特征根非负.因此, $\text{tr}\{(YA^+ - W)AA^\top(YA^+ - W)^\top\} \geq 0$ ,等号成立当且仅当 $W = YA^+$ .

由式(4)知,若 $W = YA^+$ 则 $\text{tr}\{(Y - WA)(Y - WA)^\top\} = \text{tr}\{Y(I - A^+ A)Y^\top\}$ .于是得如下定理:

**定理1.** 对于前述给定的 $N$ 个样本,欲使模糊神经网络更好地拟合这组样本,只要取 $c_{ki}, \sigma_{ki}$ 使得 $\{Y(I - A^+ A)Y^\top\}$ 达到最小值,然后令 $W = YA^+$ 即可.

取 $\text{tr}\{Y(I - A^+ A)Y^\top\}$ 作为性能指标,发现它是一个非负数,其值越小说明神经网络对样本的拟合程度越高.

### 3 性能指标的一些性质

若用 $y_i$ 表示 $Y$ 的第 $i$ 行,即所有输出样本的第 $i$ 个分量组成的向量,则性能指标化为

$$\text{tr}\{Y(I - A^+ A)Y^\top\} = \sum_{i=1}^r y_i(I - A^+ A)y_i^\top.$$

因此本文仅讨论 $r=1$ 的情形.此时将上述样本 $(x^{(k)}, y^{(k)})$ 记为 $(x^{(k)}, y^{(k)})$ , $(k=1, 2, \dots, N)$ 并记 $y = (y^{(1)}, y^{(2)}, \dots, y^{(N)})$ .

**引理1.** [2~4]若 $A_k = (A_{k-1} \cdots a_k)$ ,其中 $a_k$ 为一列向量,则有

$$A_k^+ = \begin{bmatrix} A_{k-1}^+ (I - a_k p_k^\top) \\ p_k^\top \end{bmatrix},$$

其中

$$p_k = \begin{cases} \frac{(I - A_{k-1}A_{k-1}^+)a_k}{\|(I - A_{k-1}A_{k-1}^+)a_k\|^2}, & \text{若分母} \neq 0, \\ \frac{(A_{k-1}^+)^T A_{k-1}^+ a_k}{1 + \|A_{k-1}^+ a_k\|^2}, & \text{其它.} \end{cases}$$

由引理1易得

引理2. 若  $B_k = \begin{bmatrix} B_{k-1} \\ b_k \end{bmatrix}$ , 其中  $b_k$  为一行向量, 则

$$B_k^+ = ((I - p_k b_k) B_{k-1}^+ \cdots p_k),$$

$$p_k = \begin{cases} \frac{(I - B_{k-1}^+ B_{k-1}) b_k}{\|(I - B_{k-1}^+ B_{k-1}) b_k^T\|^2}, & \text{若分母} \neq 0, \\ \frac{B_{k-1}^+ (B_{k-1}^+)^T b_k^T}{1 + \|b_k B_{k-1}^+\|^2}, & \text{其它.} \end{cases}$$

引理3. 令  $I_{ij}$  是单位矩阵经交换第  $i, j$  两行得到的矩阵, 则  $(I_{ij}A)^+ = A^+ I_{ij}$ .

由引理2和引理3可得

定理2. 令  $A$  为任一  $m \times n$  矩阵,  $y$  为任一  $n$  维列向量,  $A_k$  为将  $A$  的第  $k$  行  $a^{(k)}$  变为零向量得到的矩阵, 则  $y(I_n - A^+ A) y^+ \leq y(I_n - A_k^+ A_k) y^T$ . 等式成立当且仅当  $a^{(k)}(I_n - A_k^+ A_k) = 0$ .

(证明从略).

对  $n$  用数学归纳法可证明如下引理:

引理4.  $\sum_{i=1}^m \mu_1^{i_1}(x_1) \cdots \mu_n^{i_n}(x_n) = \prod_{j=1}^n \sum_{i=1}^{m_j} \mu_j^{i_j}(x_j)$ .

由引理4可得

$$\bar{\alpha}_i(x) = \frac{\mu_1^{i_1}(x_1) \cdots \mu_n^{i_n}(x_n)}{\sum_{j=1}^m \mu_1^{j_1}(x_1) \cdots \mu_n^{j_n}(x_n)} = \frac{\mu_1^{i_1}(x_1) \cdots \mu_n^{i_n}(x_n)}{\prod_{j=1}^k \sum_{i=1}^{m_j} \mu_j^{i_j}(x_j)} = \frac{\mu_1^{i_1}(x_1)}{\sum_{i=1}^{m_1} \mu_1^{i_1}(x_1)} \cdots \frac{\mu_n^{i_n}(x_n)}{\sum_{i=1}^{m_n} \mu_n^{i_n}(x_n)} \hat{=} \lambda_1^{i_1}(x_1) \cdots \lambda_n^{i_n}(x_n).$$

为表达方便给出两矩阵“点积”的定义

定义. 两个矩阵  $A_{m \times k}$  和  $B_{n \times k}$  的点积定义为这样的矩阵: 其第  $(i-1)n+1$  到第  $i \cdot n$  行为  $A$  的第  $i$  行与  $B$  的每一行的对应元素相乘所得的矩阵,  $i=1, 2, \dots, m$ . 记作  $A \odot B$ .

利用这一定义及引理4, 若令

$$\Lambda_k = \begin{bmatrix} \lambda_k^1(x_k^{(1)}) & \lambda_k^1(x_k^{(2)}) & \cdots & \lambda_k^1(x_k^{(N)}) \\ \lambda_k^2(x_k^{(1)}) & \lambda_k^2(x_k^{(2)}) & \cdots & \lambda_k^2(x_k^{(N)}) \\ \vdots & \vdots & \vdots & \vdots \\ \lambda_k^{m_k}(x_k^{(1)}) & \lambda_k^{m_k}(x_k^{(2)}) & \cdots & \lambda_k^{m_k}(x_k^{(N)}) \end{bmatrix}, k = 1, 2, \dots, n,$$

则  $A$  可表示为  $A = \Lambda_1 \odot \Lambda_2 \odot \dots \odot \Lambda_n$ . 由此可得结论.

**定理3.** 要寻求  $c_{ki_k}, \sigma_{ki_k}$  使  $y(I - A^+ A)y^T$  达到极小值, 仅需考虑那些使得  $\Lambda_k (k = 1, 2, \dots, n)$  中的每一向量均不为零向量的参数即可.

**引理5.** 若  $P$  为正交阵, 则  $(PA)^+ = A^+ P^{-1}$ .

**引理6.** 若  $A$  的行向量线性相关, 则  $y(I - A^+ A)y^T$  达不到极小值.

证明. 若  $A$  的行向量线性相关, 必有向量  $b$ , 使得  $bA = 0$ . 以  $\frac{b}{\|b\|}$  为第一行作正交阵  $P$ , 则

$$y(I - A^+ A)y^T = y(I - A^+ P^{-1}PA)y^T = y(I - (PA)^+ (PA))y^T.$$

而  $PA$  的第一行为零向量, 依定理2便得结论.

然而, 当  $A$  的行向量线性相关时,  $A^+ = A^T(A \cdot A^T)^{-1}$ , 因此有

**定理4.** 对于前述给定的  $N$  个样本, 欲使模糊神经网络更好地拟合这组样本, 只要取  $c_{ki_k}, \sigma_{ki_k}$  使得的行向量线性无关且  $y(I - A^T(A \cdot A^T)^{-1}A)y^T$  达到最小值, 然后令  $W = yA^+$  即可.

### 4 二阶段变半径随机搜索法

本算法基于如下分析:

1) 由定理3知应选参数  $c_{ki_k}, \sigma_{ki_k}$  使  $\frac{(x_i - c_{ki_k})^2}{\sigma_{ki_k}^2}$  不致于太大. 而隶属函数作为基函数, 对随机的一组样本而言中心分布以均匀为宜.

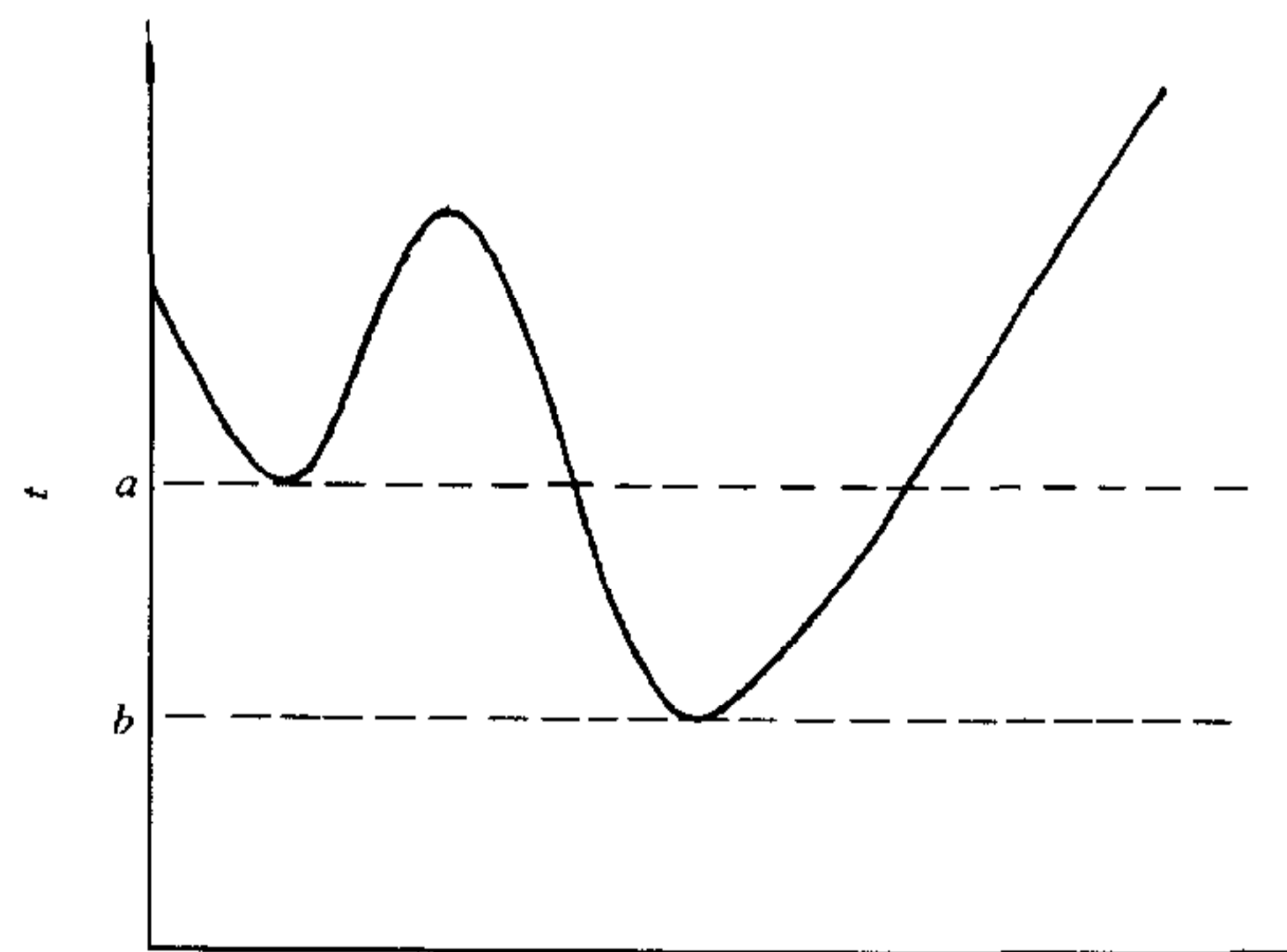


图1 临界值的选择示意图

2) BP 算法的缺点之一是不能保证达到全局最优. 以此为前车之鉴, 本算法采用分阶段搜索的方法: 对于要求的性能指标值  $e$ , 设定一精度临界值  $u$ , 在性能指标值未达到此临界值之前, 在1°所划定的范围内随机搜索, 直到找到一组使得性能指标达到  $u$  的  $\sigma_i (i = 1, 2, \dots, \sum m_i)$  为止. 此时以  $\sigma = (\sigma_1, \sigma_2, \dots, \sigma_{\sum m_i})$  为中心, 在其某邻域中随机选择新的  $\sigma$ . 临界值选得合适可以避免达到局部最优, 如图1所示. 图中  $\sigma$  为参数向,  $t(\sigma)$  为性能指标.  $u$  的选择不宜太大, 否则达不到避免局部

最优的目的, 如  $u > a$ ; 但也不宜太小, 否则会降低收敛速度, 如  $u$  和  $b$  较接近. 一般应由大到小作调整. 若选定一  $u$  值之后, 性能指标已达到  $u$ , 但在离要求精度  $e$  很远处便难以下降, 这很可能是参数掉进某一个“不合适”的邻域内或者说徘徊在局部极小范围内的缘故, 此时继续搜索很难奏效, 遇到这种情况精度临界值  $u$  便可发挥作用, 如果  $u < a$ , 性能指标达到  $u$  之后便不会掉入局部极小的邻域. 此时应减小临界值  $u$ . 由于本算法较简便, 运算时间很短, 临界值的修改不会带来多少麻烦, 但对于像 BP 算法那么复杂的算法采用此法未必行得通. 值得注意的是神经元个数一旦给定, 性能指标下限便是一定数, 尽管我们还

没有从理论上找到它. 因此单凭加大学习次数便想任意逼近某一对象是办不到的<sup>[6]</sup>. 正如图1所示, 若  $u < b$ , 算法将无休止地进行下去.

3)BP 算法的优点是从一点出发在其附近使能量函数变化最快的方向上找新的参数点, 本算法以此方式达到这一目的: 若一点使性能指标下降了, 则在这一方向上再前进一步, 直到遇到性能指标增大的点为止. 二阶段随机搜索法的算法步骤如下:

a) 选取模糊变量  $x_i$  的模糊划分  $m_i$ , ( $i = 1, 2, \dots, n$ ), 隶属函数采用正态函数  $\exp\{-\sigma^{-2}(x-c)^2\}$ , 隶属函数中心在论域中均匀分布, 最小和最大的中心值分别对应论域中的最小最大点.

b) 根据对精度的要求  $e$  设定临界值  $u$ , 选取初值向量  $\sigma^{(1)}$ , 置  $k=0$ .

c)  $k=k+1$ , 在  $\sigma^{(k)}$  的  $\delta$  邻域内随机选取新参数向量  $\sigma^{(k+1)}$ , 计算性能指标  $t(\sigma^{(k+1)})$ . 若  $t(\sigma^{(k+1)}) < u$ , 令  $u = t(\sigma^{(k+1)})$ , 转 d. 否则置  $\sigma^{(k+1)} = \sigma^{(k)}$ . 返回 c.

d) 记最先达到  $u$  的参数向量为  $\bar{\sigma}_1$ , 置  $k=0$ .

e)  $k=k+1$ . 在  $\bar{\sigma}_k$  的半径为  $L(t(\bar{\sigma}_k))$  的邻域内随机选取新参数向量  $\bar{\sigma}_{k+1}$ , 这里  $L(t)$  是  $t$  的单调下降的正值函数. 若  $t(\bar{\sigma}_{k+1}) > t(\bar{\sigma}_k)$ , 置  $\bar{\sigma}_{k+1} = \bar{\sigma}_k$ , 返回 e. 否则置  $\bar{\sigma}_{k+1} = 2\bar{\sigma}_{k+1} - \bar{\sigma}_k$ . 转 f.

f) 若  $t(\bar{\sigma}_{k+1}) \leq e$ , 停止, 否则转 e.

## 5 实例

**例1.** 利用二阶段变半径随机搜索法对

$$f(x, y) = \cos\pi x (\sin\pi y)^2, \quad (x, y) \in [-1, 1]^2$$

做仿真, 取441个样本点  $(x_i, y_j) = \left(\frac{i-1}{10} - 1, \frac{j-1}{10} - 1\right)$ , 其中  $1 \leq i, j \leq 21$ , 两个输入变量都划分为5个模糊集合, 中心值都分别取为  $-1, -0.5, 0, 0.5, 1$ , 经100步左右的迭代便可使性能指标低于0.02(即均方差小于0.0067). 图2是经学习得到的函数及其隶属函数的图形.

**例2.** 与其它方法的比较.

在文献[5]中“Table 1”列出40个数据点, 经对前20个数据点进行了接近1 000次的学习之后得到模糊神经网络模型. 用这20个数据点对模型作评价, 其精度为  $J_1 = 0.84(\%)$ . 用另外20个数据点对模型作评价, 精度为  $J_2 = 1.22(\%)$ .

采用本文的方法, 三个模糊变量的划分数均取3, 隶属函数的中心值分别取1, 3, 5. 三个模糊变量的隶属函数中依次取  $\sigma^{-2} = 0.2, 0.3, 0.4$ . 对于上述前20个数据点经一步学习后得一模型. 仍采用文献[5]中定义的精度, 用前20个数据点作评价,  $J_1 = 10^{-3}(\%)$ ; 用后20个数据点作评价,  $J_2 = 0.5404(\%)$ . 足见无论是从逼近效果还是从所用时间上作比较, 后者都占绝对优势.

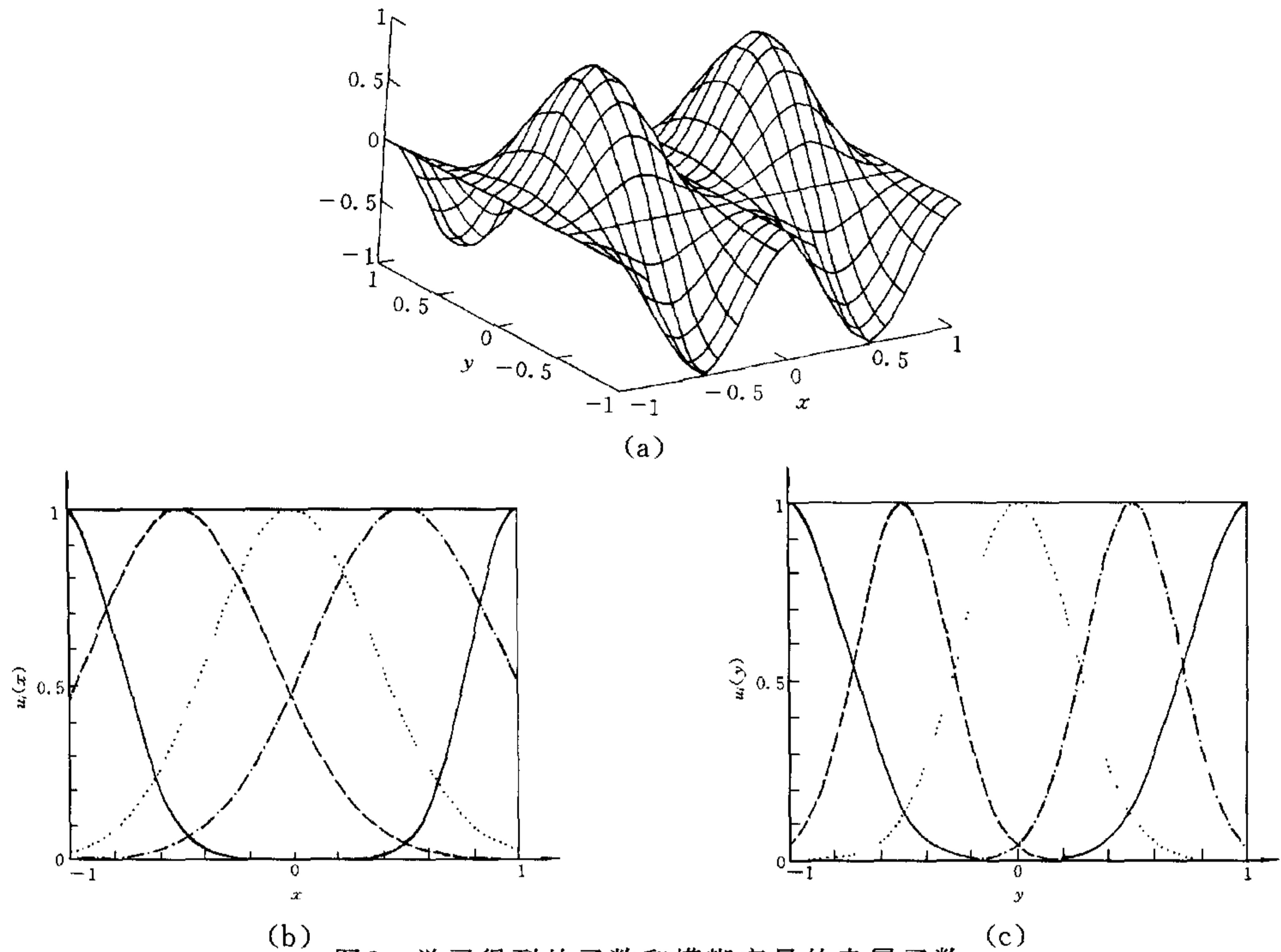


图2 学习得到的函数和模糊变量的隶属函数

## 6 结论

本算法的特点之一是不必求导但却可以达到 BP 算法通过求导才能达到的目的;之二是虽然不学习权矩阵  $W$ , 减少了运算量, 但  $W$  却取得最优值;之三是由于分两个阶段搜索, 只要临界值取得合适, 可以避免局部最优. 由于本文篇幅所限, 算法的收敛性将在另文中讨论.

## 参 考 文 献

- 1 安凯. PPDA 中的几个结论. 系统科学与数学, 1997, 17(1): 42~47
- 2 Teuvo Kohonen. Self-Organization and Associative Memory, Berlin: Springer-Verlag, 1984
- 3 Robb J Muirhead. Aspects of Multivariate Statistical Theory. New York: John Wiley and Sons Inc. 1982
- 4 Horn R A, Johnson C R. Matrix Analysis. Cambridge: Cambridge University, 1985
- 5 Horikawa S I, Furuhashi T, Uchikawa Y. On fuzzy modeling using fuzzy neural networks. *IEEE Trans. on Neural Network*, 1992, 3(5): 801~806
- 6 孙增圻. 智能控制的理论与技术. 北京: 清华大学出版社, 1997年

**安 凯** 1957年生于山西代县. 1982年获山西大学应用数学专业学士学位, 1989年获西北工业大学应用数学专业硕士学位, 1997年起在西安交通大学自动控制系攻读博士学位. 主要从事神经网络和智能开展等方面的研究工作.

**邱祖廉** 1937年生. 西安交通大学自动控制系教授、博士生导师. 主要研究方向为大时间滞后系统、智能控制和生产过程计算机控制.