

基于高斯函数与分级学习的 D-FNN 算法研究*

张 勇¹, 孙亚民², 吴建洪¹

(1. 佛山科学技术学院 电子与信息工程学院, 广东 佛山 528000;

2. 南京理工大学计算机科学与技术学院, 江苏 南京 210094)

摘要: D-FNN 基本思想是构造一个基于扩展的 RBF 神经网络, 它可以看成是一个 TSK 模糊系统, 也可以看做是基于归一化的高斯 RBF 神经网络。该文提出的算法, 学习前, 模糊神经网络不需要预先确定, 在学习的过程中, 参数估计与结构辨识同时进行, 并根据系统精度要求及模糊规则的重要性, 自动地产生或者删除一条模糊规则。在学习速度、系统结构和泛化能力方面进行了仿真实验, 仿真结果表明 D-FNN 具有更简洁的结构和优良的性能。

关键词: 动态模糊神经网络; 模糊规则; 高斯函数; 分级学习

中图分类号: TP183 **文献标志码:** A **文章编号:** 0529-6579(2014)03-0024-05

Research on D-FNN Algorithm Based on Gauss Function and Classification Learning

ZHANG Yong¹, SUN Yamin², WU Jianhong

(1. School of Electronic and Information Engineering, Foshan University, Foshan 528000, China;

2. School of Computer and Technology, Nanjing University of Science and Technology, Nanjing 210094, China)

Abstract: Dynamic fuzzy neural network (D-FNN), whose basic idea is to construct a RBF neural network based on extension, could be seen as a TSK fuzzy system, as well as a Gaussian RBF neural network based on normalized. In the algorithm proposed, fuzzy neural network does not need to be predetermined before learning. During the process of learning, parameter estimation and structure identification are done simultaneously, and a fuzzy rule would be automatically generated or deleted, according to the system accuracy requirement and importance of fuzzy rules. Simulated experiments are performed in terms of learning speed, system structure and the generalization ability. The results show that D-FNN has more concise structure and more excellent performance.

Key words: dynamic fuzzy neural network; fuzzy rules; Gaussian function; classification learning

从功能上来说, 模糊系统和神经网络都是无模型的函数逼近器, 更明确地说, 它们旨在从一个函数 f 中提取样本, 从而得到未知函数 $f: R^r \rightarrow R^n$ 的逼近^[1]。理论研究表明: 无论是神经网络还是模糊推理系统都是通用逼近器, 换言之, 只要给出足够的隐层神经元或者模糊规则, 它们就能以任意规

定的精度逼近任意函数。由于模糊神经网络不仅吸取了模糊逻辑和神经网络二者的优点, 还克服了各自具有的缺点, 即模糊神经网络有着明确的物理意义, 专家知识很容易结合到模糊神经网络中, 这对提高收敛速度、缩短训练时间是很有帮助的^[2]。正因如此, 模糊神经网络成为当前研究的一个热

* 收稿日期: 2013-10-28

基金项目: 国家自然科学基金资助项目(81272552)

作者简介: 张勇(1969年生), 女; 研究方向: 智能计算; E-mail: zhy@fosu.edu.cn

点，并形成了一个较为完善的体系。本文提出了一种动态模糊神经网络结构的快速学习算法，该模糊神经网络结构基于扩展的径向基神经网络。

1 D-FNN 的结构算法分析

1.1 D-FNN 的结构分析

本文所研究的动态模糊神经网络在结构逻辑上分 5 层。第 1 层称为输入层，每个节点分别表示一个输入的语言变量；第 2 层称为隶属函数层，每个节点分别代表一个隶属函数；第 3 层称为 T-范数层，每个节点分别代表一个可能的模糊规则中的 IF-部分；第 4 层称为归一化层，称这些节点为 N 节点。第 5 层称为输出层，该层中的每个节点分别表示一个输出变量，该输出是所有输入信号的叠加。模糊神经网络等同于一个扩展的 RBF 神经网络。多输入单输出的 RBF 网络结构如图 1 所示，图 2 是 RBF 神经网络逼近^[3]。

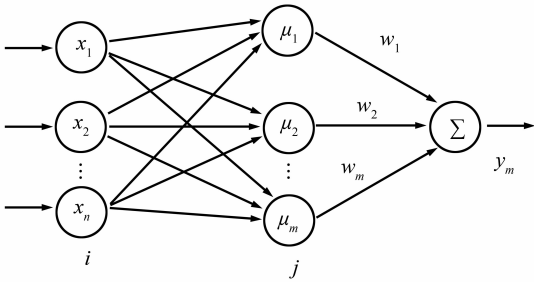


图 1 RBF 神经网络结构

Fig. 1 RBF neural network structure

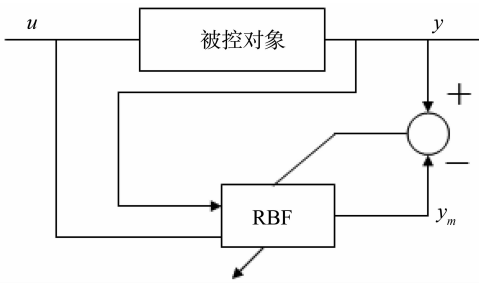


图 2 RBF 神经网络逼近

Fig. 2 RBF neural network approximation

在 RBF 网络结构中， $\mathbf{X} = [x_1, x_2, \dots, x_n]^T$ 为网络的输入向量。设 RBF 网络的径向基向量 $\boldsymbol{\mu} = [\mu_1, \mu_2, \dots, \mu_m]^T$ ，其中 μ_j 为高斯函数：

$$\mu_j = \exp\left(-\frac{X - c_j^2}{\sigma_j^2}\right), \quad j = 1, 2, \dots, m \quad (1)$$

其中 RBF 网络第 j 个节点的高斯函数的中心为 $c_j =$

$$[c_{j1}, \dots, c_{jm}]$$

设高斯函数宽度为

$$\boldsymbol{\sigma} = [\sigma_1, \sigma_2, \dots, \sigma_m]^T \quad (2)$$

σ_j 为节点 j 的高斯函数宽度，且大于零的数。网络的权向量为

$$\mathbf{W} = [w_1, w_2, \dots, w_m]^T \quad (3)$$

RBF 网络的输出为

$$y_m(t) = w_1\mu_1 + w_2\mu_2 + \dots + w_m\mu_m \quad (4)$$

RBF 网络性能指标函数^[5]：

$$J_1 = \frac{1}{2}(y(t) - y_m(t))^2 \quad (5)$$

RBF 网络输出权值、高斯函数中心及高斯函数宽度的迭代算法如下：

$$w_j(t) = w_j(t-1) + \eta(y(t) - y_m(t))\mu_j + \alpha(w_j(t-1) - w_j(t-2)) \quad (6)$$

$$\Delta\sigma_j = (y(t) - y_m(t))w_j\mu_j \frac{X - c_j^2}{\sigma_j^3} \quad (7)$$

$$\sigma_j(t) = \sigma_j(t-1) + \eta\Delta\sigma_j + \alpha(\sigma_j(t-1) - \sigma_j(t-2)) \quad (8)$$

$$\Delta c_{ji} = (y(t) - y_m(t))w_j \frac{x_i - c_{ji}}{\sigma_j^2} \quad (9)$$

$$c_{ji}(t) = c_{ji}(t-1) + \eta\Delta c_{ji} + \alpha(c_{ji}(t-1) - c_{ji}(t-2)) \quad (10)$$

其中， η 为学习速率， α 为动量因子^[4]。与一个标准的 RBF 相比，所谓的“扩展的 RBF 神经网络”这个术语是指：具有超过 3 层的网络结构；未考虑偏置量；权值可以是输入的函数而不是实常数^[5]。

1.2 提取模糊规则的方法

本文采用列主元 SVD-QR 方法作为修剪策略。Mouzouris 和 Mendel 用列主元 SVD-QR 方法从给定的规则库中提取重要的模糊规则^[6]。这种方法的基本思想是用 $\mathbf{H}^{(b)}\boldsymbol{\theta}^{(b)}$ 替换如下线性回归问题中的 $\mathbf{H}\boldsymbol{\theta}$

$$\mathbf{D} = \mathbf{H}\boldsymbol{\theta} + \mathbf{E} \quad (11)$$

其中， $\mathbf{H}^{(b)} \in \mathbf{R}^{n \times b}$ 由 \mathbf{H} 的 b 个列构成，这 b 个列在 \mathbf{H} 中的位置决定了用规则库中哪些规则来逼近向量 \mathbf{D} 。

从本质上来说，该方法同特征值分解方法是紧密相关的，它可以看作是把 $\mathbf{H} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{S}^T$ 式代入 $\boldsymbol{\Phi}_{pp}$

$$\boldsymbol{\Phi}_{pp} = \mathbf{H}^T\mathbf{H} = (\mathbf{U}\boldsymbol{\Sigma}\mathbf{S}^T)^T\mathbf{U}\boldsymbol{\Sigma}\mathbf{S}^T = \mathbf{S}\boldsymbol{\Sigma}^2\mathbf{S}^T \quad (12)$$

比较式 (12) 和 $\boldsymbol{\Phi}_{HH} = \mathbf{B}\mathbf{A}\mathbf{B}^T$ ，注意到 $\boldsymbol{\Sigma}^2$ 等价于 \mathbf{A} （因为 $\boldsymbol{\Phi}_{pp}$ 是对称且非负的），可以知道这两个

算法是等价的。它们的区别在于如何确定应保留的 θ (或等价于 H 的列): SVD-QR 使用交换矩阵 Π , 而 ED 使用测量指数向量 I_{ED} 。但是, 回顾列主元 QR 的含义, 是在矩阵 $[S_{11}^T S_{21}^T]$ 的 QR 分解的每个正交化的步骤中将列相互交换使得有最大 2-范数的列最先被正交化, 最大主元被移动到 $[R_{11} R_{12}]$ 的左上角。注意到将列交换的效果等价于做 QR 变换之前对 $[S_{11}^T S_{21}^T]$ 做相应的列交换, 因此, 交换矩阵 Π 实际可以通过计算测量指数向量来确定^[7]:

$$\begin{aligned} I_{\text{SVD-QR}} &= [I_{\text{SVD-QR}1}, I_{\text{SVD-QR}2}, \dots, I_{\text{SVD-QR}v}] = \\ &[\|s_1\|_2, \|s_2\|_2, \dots, \|s_v\|_2]^T = \\ &[(s_1 s_1^T)^{1/2}, (s_2 s_2^T)^{1/2}, \dots, (s_v s_v^T)^{1/2}] \end{aligned} \quad (13)$$

其中, $s_i \in \mathbf{R}^b$ 是 $[s_{11}^T s_{21}^T]$ 的第 i 列。矩阵 Π 的每一列元素“1”对应于 $I_{\text{SVD-QR}}$ 指数中最大的一个, 例如, Π 的第一列元素“1”对应于 $I_{\text{SVD-QR}}$ 的最大指数, 它的行等同对应于 $I_{\text{SVD-QR}}$ 的最大指数的位置^[8]。换句话说, 如果 $I_{\text{SVD-QR}}$ 的第 i 个元素最大, Π 的第 $(i, 1)$ 元素就为 1; 如果 $I_{\text{SVD-QR}}$ 的第 j 个元素为第二大, Π 的第 $(j, 2)$ 元素就为 1, 等等。特别地, 如果用

$$\begin{aligned} I_{ED} &= [I_{ED1}, I_{ED2}, \dots, I_{EDv}]^T = \\ &\text{diag}(B_b B_b^T) = \\ &\text{diag}(b_1 b_1^T + b_2 b_2^T + \dots + b_b b_b^T) \end{aligned} \quad (14)$$

式 (14) 替换式 (13) 或者用式 (13) 替换式 (14), 则这两种算法本质上是相同的。SVD-QR 算法不需要明确地构造相关矩阵, 因此, 避免了不必要的数字和舍入误差。如果在学习进行时, 用 SVD-QR 算法检测到不活跃的模糊规则并加以剔除, 则可获得更为紧凑的 D-FNN 结构^[9]。

2 实验结果与分析

分级学习算法能自动地确定模糊规则并能达到系统的特定性能。模糊规则是否加入, 加入的条件是什么? 本文引入了分级学习算法。增加一条新的模糊规则条件: 一是系统误差, 二是可容纳边界。

设

$$\|e_i\| = \|t_i - y_i\| \quad (15)$$

当

$$\|e_i\| > k_e \quad (16)$$

时需要增加一条新规则。其中 t_i 是期望的输出, y_i 是 D-FNN 系统的输出, e_i 是系统误差, k_e 是预先选定的精度。

设

$$d_i(j) = \|X_i - C_j\| \quad j = 1, 2, \dots, u \quad (17)$$

$$d_{\min} = \text{argmin}(d_i(j)) \quad (18)$$

当

$$d_{\min} > k_d \quad (19)$$

时需要增加一条新模糊规则。其中 $d_i(j)$ 是输入向量 X_i 和高斯函数中心 C_j 之间的距离, u 是模糊规则数或 RBF 单元数。 k_d 是可容纳边界有效半径。

分级学习主要思想式 (17) 和式 (18) 中的 k_e 、 k_d 不是常数, 而是动态地调节的:

$$k_e = \max[e_{\max} \times \beta^i, e_{\min}] \quad (20)$$

$$k_d = \max[d_{\max} \times \gamma^i, d_{\min}] \quad (21)$$

其中, e_{\max} 是最大误差, e_{\min} 是期望的精度, β ($0 < \beta < 1$) 是收敛常数, d_{\max} 是输入空间的最大长度, d_{\min} 是最小长度, γ ($0 < \gamma < 1$) 是衰减常数。

在这里, 被逼近的函数为如下的 Hermite 多项式:

$$f(x) = 1.1(1 - x + 2x^2) \exp\left(-\frac{x^2}{2}\right) \quad (22)$$

为了学习这个被逼近的函数, 使用区间 $[-4, 4]$ 内的随机样本函数来产生 250 个输入输出数据作为训练集。预先确定的参数如下, 且跟文献 [10] 和文献 [11] 所选一样: $d_{\max} = 2$, $d_{\min} = 0.2$, $\gamma = 0.977$, $\beta = 0.9$, $\sigma_0 = 2$, $e_{\max} = 1.1$, $e_{\min} = 0.02$, $k = 1.1$, $k_w = 1.1$, $k_{err} = 0.0015$ 采用列主元 SVD-QR 方法修剪策略与分级学习仿真程序输出的模糊规则及均方根误差如图 3 和图 4 所示。

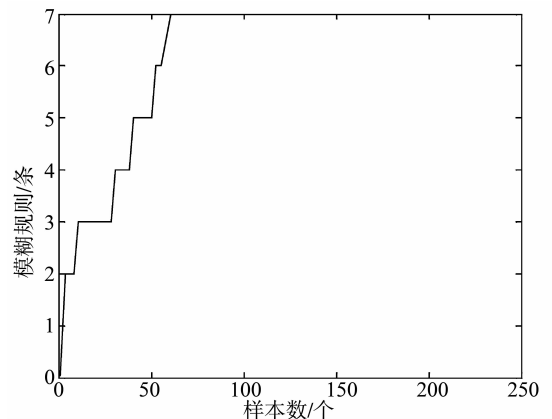


图 3 模糊规则产生过程

Fig. 3 Fuzzy rule generation

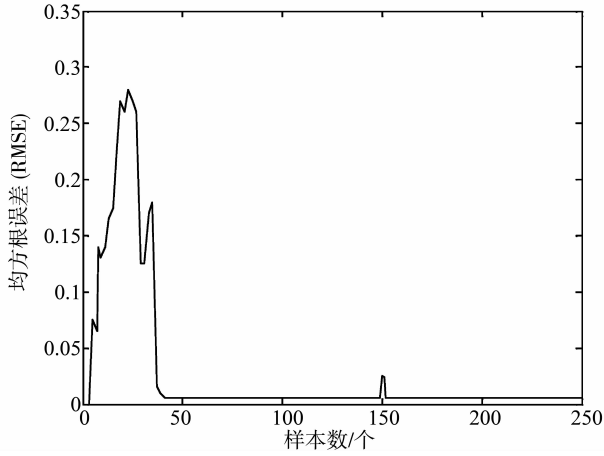


图 4 训练数据均方根误差

Fig. 4 Training data root mean square error

表 1 列出了使用不同算法的均方根误差输出结果对比。输入 49 个训练数据后，系统规则数趋于稳定，如图 3 所示。由于 D-FNN 系统结构是动态变化的，当模糊规则处于不活跃的状态时将被剔除，使得系统规则处于不断更新状态。在本次实验中，当第 151 个训练数据到来时，系统产生了一条新规则，原来不活跃模糊规则被剔除，如图 4 所示，产生了陡峭的均方根误差。

表 1 不同算法的均方根误差输出结果对比

Table 1 Comparison of different algorithms of RMSE output

	D-FNN	OLS	M-RAN	RANEKF
模糊规则数	7	8	9	14
可调参数数量	24	22	25	40
训练次数	250	250	250	250
均方根误差	0.005 5	0.009 1	0.043 8	0.026 3

D-FNN 和这些系统结构上主要不同之处在于模糊规则的结论部分。D-FNN 的结论是一个线性函数而不像 OLS、RANEKF 和 M-RAN 的结论那样是个常数。如果这些系统具有相同的复杂性，当非线性参数以相同的方法选择时，D-FNN 中的更多参数就得到了全局估计。这也许就是 TSK 模糊推理和神经网络的结合可以更加有利于设计智能系统的原因之一。从仿真结果来看（表 1），也可以得出这样的结论。

RBF 单元的误差指数和有效半径是根据式 (20) 和式 (21) 动态调节的。开始时，误差指数 k_e 和有效半径 k_d 设置较大，称为初学习；随着不

断学习，误差指数 k_e 和有效半径 k_d 的逐渐减小，开始进入细学习。由于分级学习策略的使用使得学习速度快而有效，网络结构不会持续增长，本算法应用于实时学习和控制成为可能。

图 5、图 6 显示了用 Hermite 函数逼近的仿真结果，在这里没有使用分级学习方法，而其他的参数选择和使用分级学习方法一样。表 2 列出了使用和不使用分级学习的 D-FNN 性能比较。由图 5、图 6 以及表 2 可以看出，不使用分级学习的 D-FNN，由于 RBF 单元的有效半径 k_d 和误差指数 k_e 总是很小，因此具有更小的输出误差和 RMSE。然而，尽管两者的 RMSE 几乎相同，但采用分级学习的算法能保证更小的 D-FNN 结构和快速的学习速度。

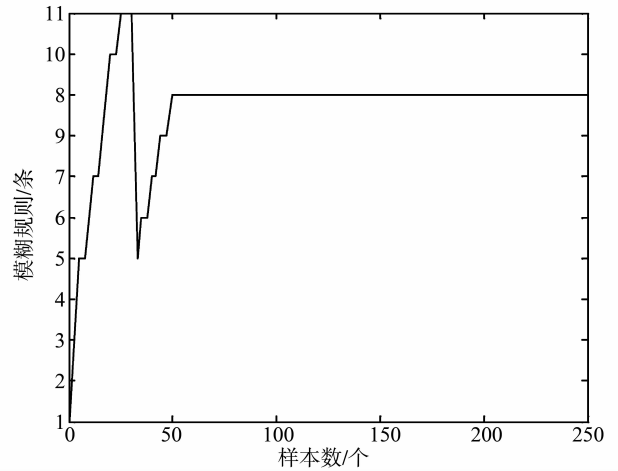


图 5 模糊规则产生

Fig. 5 Fuzzy rule generation

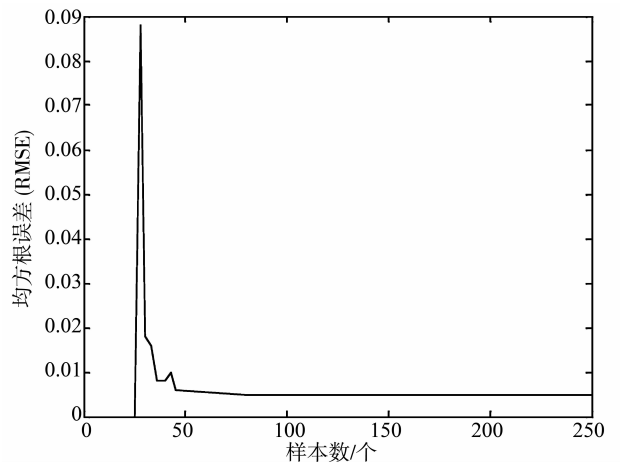


图 6 训练数据均方根误差

Fig. 6 Training data root mean square error

表 2 使用和不使用分级学习的 D-FNN 性能比较

Table 2 Performance comparison of use and do not use the classification to learn D-FNN

方法	规则数	训练时均方根误差	运行时间/s
采用分级学习的 D-FNN 系统	7	0.005 5	5.77
未采用分级学习的 D-FNN 系统	9	0.005 4	6.56

D-FNN 算法中, 非线性参数 (前提参数) 是由训练样本 (中心) 和启发式方法 (高斯宽度) 直接决定的, 而没有用优化算法来确定。虽然高斯宽度在学习时可以自适应地调整, 但学习规则却很简单。另一方面, 线性参数 (结论参数) 在每一步由 LLS 或 KF 进行修正, 用这种方法得到的解是全局最优的, 因此, 只需一步训练就可达到而目标而不需要迭代学习。这种方法的学习速度显然比文献 [12-14] 中采用的 BP 算法快得多。运行非线性系统辨识、函数逼近和 Mackey-Glas 时间序列预测仿真时间列于表 3 中。

表 3 仿真运行时间

Table 3 Simulation run time

例子	样本长度	运行时间 / s
非线性动态系统的辨识 ^[9]	250	22.45
函数逼近 ^[9]	250	6.75
Mackey-Glas 时间序列预测 ^[9]	250	175.27

值得一提的是, D-FNN 学习的快速性是以它不能获得最优为代价的。D-FNN 采用在线学习方法, 每个样本只学习一次, 而不像文献 [12-14] 那样, 一个学习过程是基于整个学习样本, 并对整个神经网络参数优化处理, 这一点可以从 Mackey-Glas 时间序列预测中看出来。

3 结 论

仿真结果表明, D-FNN 具有以下明显优点:

1) 学习速度更快。线性参数在每一步由 LLS 或 KF 进行修正, 用这种方法得到的解是全局最优的, 因此, 只需一步训练就可达到而目标而不需要迭代学习。这种方法的学习速度显然比采用的 BP 算法快得多。

2) 系统结构紧凑。仿真结果均表明: 分级学习的思想能保证更简洁的结构和更短的学习时间。

3) 泛化能力强大。从泛化能力的角度来看, 仿真结果均表明: 对测试数据的误差与对训练数据的训练误差很接近。这就是说, D-FNN 具有很强的泛化能力。

参考文献:

- [1] HONG Z. Algebraic feature extraction of image for recognition[J]. *Patt Recog*, 2009, 24(2): 211-219.
- [2] TURK M A, PENTLAND A P. Eigenfaces for recognition[J]. *Cognitive Neuroscience*, 2010, 3(1): 71-86.
- [3] CHELLAPPA R, WILSON C L, SIROHEY S. Human and machine recognition of faces: a survey[J]. *Proc. IEEE*, 2011, 93(2): 705-740.
- [4] 刘金琨. 机器人控制系统的设计与 MATLAB 仿真[M]. 北京:清华大学出版社, 2008.
- [5] POLYCARPOU M M, CONWAY J Y. Indirect adaptive nonlinear control of drug delivery systems[J]. *IEEE Trans Automat Contr*, 2009, 43: 849-856.
- [6] 何正风, 张德丰, 孙亚民. 高斯激活函数特征值分解修剪技术的 D-FNN 算法研究[J]. *中山大学学报:自然科学版*, 2013, 52(1): 34-39.
- [7] WANG L X, MENDEL J M. Fuzzy basis function, universal approximation, and orthogonal least squares learning[J]. *IEEE Trans Neural Networks*, 2012, 3: 907-914.
- [8] 任爱红. 模糊随机过程函数列均方差一致 Henstock 积分的可积性[J]. *中山大学学报:自然科学版*, 2010, 51(4): 41-44.
- [9] WU S Q, ER M J. Dynamic fuzzy neural networks: a novel approach to function approximation[J]. *IEEE Trans Syst, Man, Cybern: Part B*, 2011, 30: 358-36.
- [10] LU Y, SUNDARARAJAN N, SARATCHANDRAN P. A sequential learning scheme for function approximation by Using minimal radial basis function networks[J]. *Neural Computation*, 2012, 19(2): 461-478.
- [11] KADIRKAMANATHAN V, NIRANJAN M. A function estimation approach to sequential learning with neural networks[J]. *Neural Computation*, 2011, 16(4): 954-975.
- [12] CHO C B, WANG B H. Radial basis function based adaptive fuzzy systems and their application to system identification and prediction[J]. *Fuzzy Sets and Systems*, 2011, 83: 325-339.
- [13] CHAO C T, CHEN Y J, TENG C C. Simplification of fuzzy neural systems using similarity analysis[J]. *IEEE Trans Syst, Man, Cybern: Part B, Cybern*, 2011, 26: 344-354.
- [14] JANG J-S R. ANFIS: adaptive-net work-based fuzzy inference system[J]. *IEEE Trans Syst, Man, Cybern*, 2012, 23: 665-684.