文章编号: 1001-0920 (2012) 07-0997-06

基于在线减法聚类的RBF神经网络结构设计

张昭昭[1,2], 乔俊飞[1]

(1. 北京工业大学 电子信息与控制工程学院，北京 100124；
2. 辽宁工程技术大学 电子与信息工程学院，辽宁 葫芦岛 125105)

摘 要: 以设计最小径向基函数 (RBF) 神经网络结构为着眼点, 提出一种在线 RBF 网络结构设计算法. 该算法将在线减法聚类能实时跟踪工况的特性与 RBF 网络参数学习过程相结合, 使得网络既能在线适应实时对象的变化又能维持紧凑的结构, 有效地解决了 RBF 神经网络结构自组织问题. 该算法只调整欧氏距离距实时工况最近的核函数, 大大提高了网络的学习速度. 通过对典型非线性函数逼近和混沌时间序列预测的仿真, 表明所提出的算法具有良好的动态特性响应能力和逼近能力.

关键词: RBF 神经网络；结构设计；在线减法聚类
中图分类号: TP273　　**文献标识码:** A

Design RBF neural network architecture based on online subtractive clustering

ZHANG Zhao-zhao[1,2], *QIAO Jun-fei*[1]

(1. College of Electronic and Control Engineering，Beijing University of Technology，Beijing 100124，China；2. Institute of Electronic and Information Engineering，Liaoning Technical University，Huludao 125105，China．Correspondent: ZHANG Zhao-zhao，E-mail：zzzhao123@126.com)

Abstract: This paper presents an online architecture design algorithm for radical basis function(RBF) neural network based on online subtractive clustering algorithm aiming at designing the minimal RBF neural network structure. The algorithm combines the characteristics that the online substractive clustering can track the real-time condition with the parameters learning process of the RBF neural network, which makes the RBF neural network adapt to the change of real-time condition dynamics while maintaining a compact network architecture. Therefore, this method can effectively solve the problem of self-organizing structure design of the RBF neural network. Only the kernel function whose Euclidean distance is nearest to the real-time conditions is adjusted, which greatly improves the learning speed of the network. The results of experiments on the typical function approximation and the chaotic time series prediction show that the algorithm owns favorable dynamic character response and approximating ability.

Key words: radical basis function neural network；architecture design；online subtractive cluster

1 引 言

径向基函数 (RBF) 神经网络由于其拓扑结构简单和通用的逼近能力, 以及收敛速度快、不易陷入局部极小点、鲁棒性好等优点, 在模式分类、信号处理、非线性系统的建模与控制等方面得到广泛的应用[1-3]. RBF 神经网络应用的关键问题是网络结构设计和网络参数学习. 所谓 RBF 神经网络的结构设计, 即是如何确定 RBF 网络中隐节点 (核函数) 的个数. 相对于参数学习而言, 神经网络结构设计要困难得多, 至今还没有确定的方法可循. 一个公认的指导原则是“在没有其他先验知识的情况下, 与给定样本一致的规模最小的网络便是最好的网络”[4], 从数学的角度看, 这相当于在样本点偏差允许范围之内, 用最平滑的函数去逼近未知的非线性映射. 因此, RBF 网

收稿日期: 2010-12-13；**修回日期:** 2011-04-24.
基金项目: 国家863计划项目(2009AA04Z155)；国家自然科学基金重点支持项目(61034008)；国家自然科学基金项目(60873043)；北京市自然科学基金项目(4092010)；教育部博士点基金项目(200800050004)；北京市属高等学校人才强教计划项目(PHR201006103).
作者简介: 张昭昭(1973−), 男, 博士生, 从事智能信息处理、神经网络结构设计与优化的研究；乔俊飞(1968−), 男, 教授, 博士生导师, 从事智能控制、智能信息处理等研究.

络设计的核心问题是设计能够满足精度要求且具有最少隐节点数(最小结构)的网络, 以保证网络的泛化性能.

RBF网络设计的常用方法有: 增长方法, 修剪方法和增长与修剪相结合的方法. Platt提出的资源分配网络(RAN)是一种著名的在线学习增长方法[5], 主要思想是在网络学习过程中, 当发现"未建模"的样本满足距离准则和误差准则时, 为其分配一个新节点. RAN利用RBF网络核函数的局部响应特性, 权值调节时只调节新增加的隐节点, 加快了学习速度. 调节数据中心时使用了样本的输出信息以提高学习精度, 但在RAN中, 隐节点一旦添加便不能删除, 所以随着在线学习时间的加长, 网络的隐节点将越来越多, 导致网络泛化能力下降. 文献[6]提出一种修剪型RBF神经网络算法, 该算法是在训练完所有的样本后对神经网络隐层进行修剪而不是在学习过程中进行, 因此该算法不适合于实时系统. [7]在RAN的基础上提出了一种最小资源神经网络(MRAN)[7], MRAN利用增长和修剪策略对RBF神经网络结构进行调整[8], 若某个隐节点不能被多个连续输入样本激活, 则删除该隐节点. 所以MRAN属于增长和修剪相结合的方法, 能最终获得比较合适的RBF结构. [8]也提出了一种增长和修剪RBF神经网络结构设计算法(GGAP-RBF) [9], 该算法通过在线学习的方法判断隐含层神经元的重要性, 从而增减神经网络结构. 但是, GGAP-RBF需要根据全局样本数据设定初始值, 对于一个在线RBF神经网络结构设计算法, 预先获得全局样本数据集有时并不可能. 因此, 到目前为止, RBF神经网络结构优化设计方法依然是一个开放问题[10].

鉴于上述存在的问题, 本文提出了一种基于在线减法聚类的RBF神经网络结构优化设计方法(OLSC-RBF). 该方法将在线减法聚类与RBF神经网络的学习过程有机的结合, 能根据实时工况在线增长或删除RBF神经网络隐层神经元, 使RBF神经网络具有良好的自适应能力, 解决了RBF神经网络结构过大或过小的问题. 本文最后通过仿真实验验证了该算法的性能.

2 RBF神经网络

2.1 RBF神经网络结构

RBF网络是一种前馈网络, 一般分为3层, 分别为输入层、隐含层和输出层. 设RBF网络结构为$n-m-p$(n个输入节点, m个隐含层节点和p个输出节点), 则第k个输出节点的输出为

$$y_k=\sum_{i=1}^{m} w_i\varPhi_i(\|\boldsymbol{x}-c_i\|). \tag{1}$$

其中$\boldsymbol{x}=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}\in R^n$为网络输入向量, $W\in R^{p\times m}$为输出权矩阵, $y=[y_1,y_2,\cdots,y_p]^{\mathrm{T}}$为网络输出, $\varPhi(*)$为第i个隐节点的激活函数, 即

$$\varPhi_i(\|\boldsymbol{x}-c_i\|)=\mathrm{e}^{-\|\boldsymbol{x}-c_i\|^2/\delta_i^2}. \tag{2}$$

其中: c_i为核函数的数据中心, δ_i为该核函数的扩展宽度.

2.2 RBF神经网络梯度学习方法

采用带遗忘因子的RBF网络学习方法, 通过最小化目标函数实现对各隐节点数据中心、扩展常数和输出权值的调整.

设神经网络学习的目标函数为

$$E=\frac{1}{2}\sum_{j=1}^{N}\beta_j e_j^2. \tag{3}$$

其中: β_j为遗忘因子, 误差信号e_j定义为

$$e_j=y_j-F(\boldsymbol{x}_j)=y_j-\sum_{i=1}^{m}w_i\varPhi_i(\boldsymbol{x}_j). \tag{4}$$

神经网络函数$F(\boldsymbol{x})$对数据中心c_i, 扩展常数δ_i和输出权值w_i的梯度分别为

$$\nabla_{c_i}F(\boldsymbol{x})=\frac{2w_i}{\delta_i^2}\varPhi_i(\boldsymbol{x})(\boldsymbol{x}-c_i), \tag{5}$$

$$\nabla_{\sigma_i}F(\boldsymbol{x})=-\frac{2w_i}{\delta_i^3}\varPhi_i(\boldsymbol{x})\|\boldsymbol{x}-c_i\|^2, \tag{6}$$

$$\nabla_{w_i}F(\boldsymbol{x})=\varPhi_i(\boldsymbol{x}). \tag{7}$$

考虑所有训练样本和遗忘因子的影响, c_i, δ_i和w_i的调节量分别为

$$\varDelta_{c_i}=\eta\frac{w_i}{\delta_i^2}\sum_{j=1}^{N}\beta_j e_j\varPhi(\boldsymbol{x}_j)(\boldsymbol{x}_j-c_i), \tag{8}$$

$$\varDelta_{\delta_i}=-\eta\frac{w_i}{\delta_i^3}\sum_{j=1}^{N}\beta_j e_j\varPhi_i(\boldsymbol{x}_j)\|\boldsymbol{x}_j-c_i\|, \tag{9}$$

$$\varDelta_{w_i}=\eta\sum_{j=1}^{N}\beta_j e_j\varPhi_i(\boldsymbol{x}_j). \tag{10}$$

式中: $\varPhi_i(\boldsymbol{x}_j)$为第$i$个隐节点对$\boldsymbol{x}_j$的输出, η为学习率.

3 基于在线减法聚类的RBF网络结构设计

RBF神经网络结构设计是如何针对具体的学习任务, 确定合适的核函数个数n, 核函数的数据中心c_i和扩展宽度δ_i. 在RBF网络的学习过程中, 这些参数可以通过在线减法聚类算法和网络学习算法共同获得. 在线减法聚类算法聚类时以所有数据样本作为可能的聚类中心并计算他们的密度值, 密度值以该数据样本到所有其他数据样本的距离和作为度量标准[11-12], 其形式为

$P(x(i))=$

$$\frac{1}{1+\frac{1}{N-1}\sum_{j=1,j\neq i}^{N}(x(i)-x(j))^{\mathrm{T}}(x(i)-x(j))}. \quad (11)$$

其中: $i=1,2,\cdots,N$; $x(i)$ 为数据集 X 中的数据样本; $X=[x(1),x(2),\cdots,x(N)]$; N 为数据 X 中的数据样本量.

首先以第 1 个数据样本 $x(1)$ 为第 1 个聚类中心 (即RBF网络隐层核函数的数据中心), 并赋值$P_1(x(1))=1$. $P_k\,(k=1,2,\cdots)$ 表示第 k 时刻计算的密度值.

在 k 时刻, 假设已经存在 l 个聚类中心 $c_i(i=1,2,\cdots,l)$, 则需要计算当前数据样本 $x(k)$ 与原聚类中心 c_i 的密度值, 并进行比较. $x(k)$ 密度值的计算公式如下:

$$P_k(x(k))=\frac{1}{1+\frac{1}{k-1}\sum_{j=1}^{k-1}(x(k)-x(j))^{\mathrm{T}}(x(k)-x(j))}=\frac{k-1}{(k-1)(\gamma(k)+1)-2\eta(k)x(k)+\sigma(k)}. \quad (12)$$

其中

$$\gamma(k)=x^{\mathrm{T}}(k)x(k),\ \eta(k)=\sum_{j=1}^{k-1}x^{\mathrm{T}}(j),$$

$$\sigma(k)=\sum_{j=1}^{k-1}x^{\mathrm{T}}(j)x(j).$$

在式 (12) 右边, $\gamma(k)$ 可由当前数据样本 $x(k)$ 计算得到, 而其他参数存在以下递推关系:

$$\eta(k)=\eta(k-1)+x^{\mathrm{T}}(k-1), \quad (13)$$

$$\sigma(k)=\sigma(k-1)+x^{\mathrm{T}}(k-1)x(k-1). \quad (14)$$

由式 (12)~(14) 知, 在计算 $P_k(x(k))$ 时, 并不需要保留 k 时刻之前的所有数据, 只需保留 $\eta(k-1),\sigma(k-1),x(k-1)$, 即可由当前的数据 $x(k)$ 递推而求得.

对于原聚类中心 c_i, 由式 (11) 可得到 $k-1$ 时刻的密度值为

$$P_{k-1}(c_i)=\frac{1}{1+\frac{1}{k-2}\sum_{j=1,j\neq i}^{k-1}(c_i-x(j))^{\mathrm{T}}(c_i-x(j))}, \quad (15)$$

则在 k 时刻 c_i 密度值的递推公式可以描述如下:

$$P_k(c_i)=\frac{1}{1+\frac{1}{k-1}\sum_{j=1,j\neq i}^{k}(c_i-x(j))^{\mathrm{T}}(c_i-x(j))}=\frac{(k-1)P_{k-1}(c_i)}{k-2+P_{k-1}(c_i)+\varsigma(k)P_{k-1}(c_i)}, \quad (16)$$

其中 $\varsigma(k)=(c_i-x(k))^{\mathrm{T}}(c_i-x(k))$ 可由当前数据 $x(k)$ 计算得到.

由式 (11) 知, 数据样本周围数据越多密度值越大, 因此, 数据样本的密度值反映了该数据在所有数据中的描述能力. 通过对 $P_k(x(k))$ 与 $P_k(c_i)(i=1,2,\cdots,l)$ 进行比较, 实现在线 RBF 神经网络结构的设计.

下面分 3 种情况进行讨论:

1) 隐节点的增加.

If $\min\limits_{i=1,2,\cdots,l}\|x(k)-c_i\|_2>r_1$ and $P_k(x(k))>\varepsilon_1$,

Then $c_{l+1}=x(k),\ l=l+1$.

其中 r_1 和 ε 为预先设定的值, 分别取

$$r_1=(0.3\sim0.5)\max_{i\neq j,i,j=1,2,\cdots,N}\parallel x(i)-x(j)\parallel_2, \quad (17)$$

$$\varepsilon_1=0.5\times\max_{i=1,2,\cdots,l}P_{k-1}(c_i). \quad (18)$$

这种情况表明, 该数据样本与最近的核函数数据中心的距离足够大, 且其密度值也大于给定的阈值, 表明当前 RBF 网络遇到一个无法学习的新工况, 且该工况周围数据点比较多, 应该在 RBF 网络隐层增加一个以该数据为中心的隐节点跟踪该工况.

新增隐节点的相关参数设置为

$$w_{l+1}=e_n,\ c_{l+1}=x(k),\ \delta_{l+1}=\kappa\|x(k)-c_{nr}\|. \quad (19)$$

其中: $e_n=y_n-f(x(k))$, c_{nr} 为距离当前数据样本 $x(k)$ 最近的 RBF 网络隐层核函数的数据中心.

2) 隐节点的合并.

If $\min\limits_{i\neq j,i,j=1,2,\cdots,l}\parallel c_i-c_j\parallel\leqslant r_2$ and$|\delta_i-\delta_j|\leqslant\varepsilon_2$,

Then $c_i=c_d,\ l=l-1$.

其中: 若 $P_k(c_j)>P_k(c_i)$, 则 $d=j$; 否则, $d=i$; $r_2\approx(0.5\sim0.7)r_1$; $\varepsilon_2\in(0,1)$ 为设定阈值.

这种情况表明, 当 RBF 网络隐层中两个核函数中心的距离足够近时, 为了使 RBF 网络结构更加紧凑, 应将这两个隐节点合并.

合并隐节点后相关参数设置为

$$w_i=w_i+w_j,\ c_i=(c_i+c_j)/2,\ \delta_i=\max(\delta_i,\delta_j). \quad (20)$$

3) 隐节点的删除.

网络在线跟踪实时工况的过程中, 如果有某隐节点长时间没有被任何工况数据激活, 则表明该隐节点已经不能跟踪实时工况. 为了使 RBF 网络维持紧凑的结构, 这些隐节点应该被删除.

综上所述, 可获得基于在线减法聚类的 RBF 神经网络结构设计算法如下:

Step 1: 开始时网络中无隐节点.

Step 2: 对每个在线输入的样本 $x(k)$, 若 $k=1$, 则分配一个隐节点, 该隐节点核函数的数据中心 $c_1=$

$x(1)$, 扩展宽度 $\delta_1 = 0.5$; 若 $k \neq 1$, 则通过式 (11) 和 (16) 分别递推计算该数据样本的密度值 $P_k(x(k))$ 和该数据样本到所有核函数数据中心 $c_i(i = 1, 2, \cdots, l)$ 的密度值 $P_k(c_i)$.

Step 3: 若满足隐节点增加准则, 则增加一个隐节点, 并以式 (19) 给该新增加的隐节点赋以初始参数, 转 Step 2; 否则, 执行下一步.

Step 4: 以式 (8)～(10)调整距离 $x(k)$ 最近的核函数的参数 c_{nr}, δ_{nr} 和 w_{nr}, 每次网络参数调节后, 累计各隐节点未被激活的次数.

Step 5: 删除长期未被激活的隐节点.

Step 6: 检查距离实时工况数据 $x(k)$ 欧式距离最近的隐节点是否满足与其他隐节点合并的条件, 若满足, 则合并相应的隐节点, 并以式 (20) 给合并后的隐节点赋以相关参数值. 转 Step 2.

该算法将在线减法聚类算法和 RBF 网络参数学习算法有机地结合, 利用在线减法聚类算法能够根据在线数据准确判断实时工况的特性, 结合 RBF 神经网络的局部响应的物理特性, 准确跟踪实时工况. 算法中对网络参数的调整, 每次只调整距离实时工况数据 $x(k)$ 欧式距离最近的核函数的参数, 极大地提高了网络的学习速度.

4　仿真实验

4.1　非线性函数逼近

为检验本文提出的基于在线减法聚类的 RBF 神经网络结构设计算法对非线性系统的建模性能, 选取典型非线性函数进行逼近, 该函数已被证明非常适合用来检测神经网络的性能[13].

$$f(x,y) = -10\Big(\frac{x}{5} - x^3 - y^5\Big)\mathrm{e}^{-x^2-y^2} - \frac{1}{3}\mathrm{e}^{-(x+1)^2-y^2} + 3(1-x)^2\mathrm{e}^{-x^2-(y+1)^2}. \tag{21}$$

对式 (21) 进行训练, 训练前随机产生 900 个训练样本对 $((x,y), f(x,y))$, 其中 $x, y \in [-3, 3]$. 测试样本对为 $x, y \in \{-3, -2.8, -2.6, \cdots, 3\}$, 共 961 个. 仿真时学习率 $\eta = 0.01$, 遗忘因子 $\beta = 0.7$, 网络结构调整参数 $\varepsilon_2 = 0.2$.

对函数 $f(x,y)$ 的逼近效果如图 1 所示. 逼近 $f(x,y)$ 时, RBF 网络隐节点变化动态及跟踪实时工况误差变化如图 2 所示. 函数 $f(x,y)$ 的逼近误差曲面如图 3 所示.

由图 1 训练后的神经网络对典型非线性函数测试样本的逼近效果可以看出, 网络输出值与函数期望值基本重合, 说明 OLSC-RBF 神经网络具有很强的逼近能力. 由图 2 网络对学习样本在线学习过程中的隐

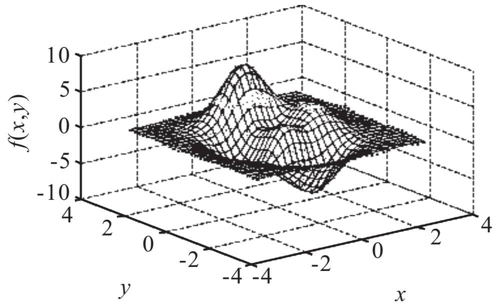

图 1　$f(x,y)$ 逼近效果

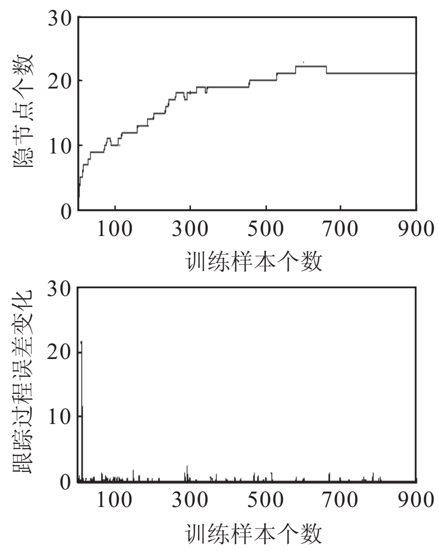

图 2　逼近 $f(x,y)$时隐节点和误差变化

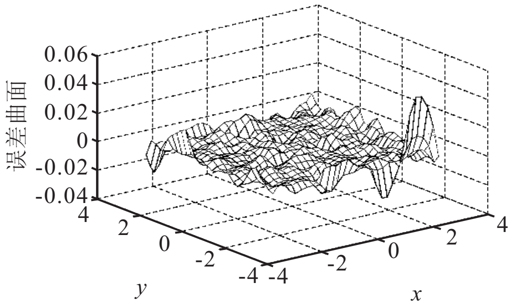

图 3　逼近 $f(x,y)$的误差曲面

节点变化动态可看出, OLSC-RBF 神经网络在跟踪实时工况的过程中, 能够根据实时工况数据自适应地调整网络结构. 从跟踪实时工况的误差变化曲线也可以看出, 虽然在网络构建的初始阶段, 跟踪误差波动较大, 但在随后的学习过程中, 跟踪误差很快得到有效抑制, 因此 OLSC-RBF 神经网络确实可以有效跟踪实时工况. 由图 3 对函数的逼近误差曲面也可以清晰看出, 检测误差基本上在 0.02 以内, 且除个别点外整个误差曲面比较平坦, 表明 OLSC-RBF 神经网络对任务的学习较好且最终网络具有很强的泛化性能.

为体现本文提出的 OLSC-RBF 算法较之其他动态RBF神经网络结构设计算法具有更好的性能, 将 OLSC-RBF 算法与 MRAN 算法和 GGAP-RBF 算法

进行比较, MRAN 算法曾一度是 RBF 神经网络结构设计的通用算法[14]. 上述各算法, 在相同的条件下对同样的问题达到相同的期望误差运行 20 次的平均结果比较如表 1 所示.

表 1　3 种算法性能比较

函 数	算 法	期望误差	检测误差	最终网络(隐节点数)	训练时间/min
	OLSC-RBF	0.01	0.015 1	21.2	1.39
$f(x,y)$	GGAP-RBF	0.01	0.021 8	24.3	1.42
	MRAN	0.01	0.033 6	30.4	2.17

由表 1 可以看出, 3 种算法中 OLSC-RBF 算法设计出的 RBF 神经网络结构最为紧凑, 且检测误差最小, 一定程度上表明 OLSC-RBF 神经网络的泛化能力较优. OLSC-RBF 算法在线调整网络参数和结构时, 充分考虑了 RBF 了神经网络的物理特性与在线减法聚类的性质, 只调节距离实时工况数据最近的核函数, 因此 OLSC-RBF 算法在运行时间方面占优.

4.2　混沌时间序列预测

混沌现象是一种由确定的非线性动力系统生成的复杂行为, 混沌时间序列预测已经在许多领域得到广泛应用[15]. 为检验该算法对混沌系统预测的能力, 选取典型混沌时间序列 Mackey-Glass 作为仿真实例进行预测研究. Mackey-Glass 混沌时间序列可由如下时延微分方程产生:

$$\frac{\mathrm{d}x}{\mathrm{d}t}=\frac{ax(t-\tau)}{1+x^{10}(t-\tau)}-bx(t). \tag{22}$$

用 4 阶 Runge-Kutta 法求解该微分方程, 方程中的参数设置为: $a=0.2$, $b=0.1$, $\tau=17$, 初始条件取 $x(0)=1.2$, 且当 $t<0$ 时, $x(t)=0$. 产生 3 000 个样本进行训练 ($t=1,2,\cdots,3\,000$), 用 3000 个样本进行测试($t=3\,001,\cdots,6\,000$). 以嵌入矢量 $[x(t)\ x(t-6)\ x(t-12)\ x(t-18)]$ 预测 $x(t+20)$ 的值. 仿真时学习率 $\eta=0.02$, 遗忘因子 $\beta=0.8$, 网络结构调整参数 $\varepsilon_2=0.12$. 预测曲线如图 4 所示, 训练过程中网络隐节点变化动态与训练误差曲线如图 5 所示.

表 2 给出了 OLSC-RBF 算法与 MRAN 和 GGAP-RBF 对 Mackey-Glass 混沌时间序列预测结果的对比.

由图 4 和表 2 可看出, 3 种算法设计出的 RBF 神经网络都能取得一定的预测效果, 然而无论在网络结构的紧凑性和网络的预测精度方面, OLSC-RBF 神经网络均占优, 特别是算法的运行时间, OLSC-RBF 神经网络优势较大. 由图 5 跟踪过程的误差曲线变化可以看出, 虽然在 $t=2\,500$ 以后, 训练误差出现了较大的波动, 但是误差一直控制在 10^{-4} 以内, 而且很快得到了抑制. OLSC-RBF 算法, 一方面在网络的学习过程中只根据最近工况数据调整网络参数和结构, 因此运行速度较快; 另一方面, 由在线减法聚类获得的 RBF 神经网络的隐节点的数据中心具有更为丰富的信息量, 不仅包括当前时刻工况数据的信息, 而且包含历史数据的信息. 因此, 虽然最终网络隐节点数较少, 但在预测精度上还是优于另外两种算法.

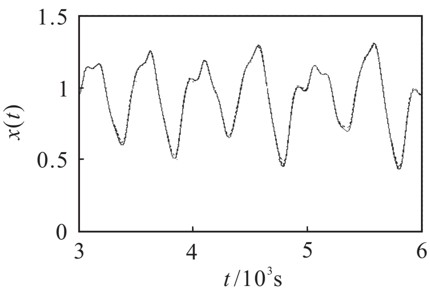

图 4　MG 时间序列预测图

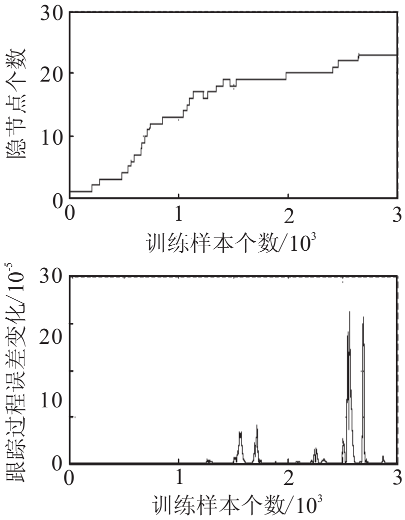

图 5　训练过程中隐节点数和误差变化

表 2　3 种算法预测效果对比

算 法	最终网络(隐节点数)	预测精度	时间/s
OLSC-RBF	23	0.010 3	16.375
GGAP-RBF	24	0.014 0	18.659
MRAN	30	0.049 2	26.166

表 3 为 3 种算法对 Mackey-Glass 混沌时间序列预测的相对误差及样本分布情况的对比. 从表 3 可以看出, 用 OLSC-RBF 神经网络预测, 94.8% 的样本测试误差在 10% 以内, 83.7% 的样本测试误差在 5% 以内, 测试均方差为 1.017; 而用 GGAP-RBF 神经网络和 MRAN 神经网络预测误差在 10% 以内的样本数分别为 91.4% 和 84.2%, 测试均方差分别为 1.702 和 2.833. 因此, 在一定的精度范围内 OLSC-RBF 神经网络总体预测效果要优于其他两种算法.

表 3　3 种算法预测相对误差与样本分布

算 法	样本个数			
OLSC-RBF	711	1 506	2 511	2 844
GGAP-RBF	627	1 395	2 481	2 742
MRAN	498	1 209	2 295	2 526
相对误差/%	⩽ 1	⩽ 2	⩽ 5	⩽ 10

5 结 论

本文以设计最小RBF神经网络结构为着眼点,提出了一种基于在线减法聚类的RBF神经网络结构设计算法. 该算法利用在线减法聚类算法能有效跟踪实时工况的特性以及RBF神经网络隐层神经元的物理意义, 确定要增加或要删除的隐节点. 该算法属于在线RBF神经网络结构设计算法, 充分考虑了算法的实时特性, 网络参数学习和结构调整只考虑距离实时工况数据最近的RBF神经网络隐层神经元, 因此该算法不仅能够根据工况的实时特性自适应在线调整RBF神经网络结构, 而且神经网络参数学习和结构调整速度较快. 通过对典型非线性函数的逼近和Mackey-Glass混沌时间序列预测以及与其他动态RBF神经网络结构设计算法进行比较, 表明该算法设计出的RBF神经网络确实能有效跟踪实时工况, 且设计出的神经网络具有较强的泛化性能.

参考文献(References)

[1] 宋彦坡, 彭小奇. 一种结构自适应神经网络及其训练方法[J]. 控制与决策, 2010, 25(8): 1265-1268.
(Song Y P, Peng X Q. New structure adapting neural network and its training method[J]. Control and Decision, 2010, 25(8): 1265-1268.)

[2] Chen S, Wang X X, Brown D J. Sparse incremental regression modeling using correlation criterion with boosting search[J]. IEEE Signal Processing Letters, 2005, 12(3): 198-201.

[3] 厉英, 王正, 敖志广, 等. BP神经网络漏钢预测系统优化[J]. 控制与决策, 2010, 25(3): 453-456.
(Li Y, Wang Z, Ao Z G, et al. Optimization for breakout prediction system of BP neural network[J]. Control and Decision, 2010, 25(3): 453-456.)

[4] 张昭昭, 乔俊飞, 韩红桂. 一种基于神经网络复杂度的修剪算法[J]. 控制与决策, 2010, 25(6): 178-182.
(Zhang Z Z, Qiao J F, Han H G. A pruning algorithm based on nueual complexity[J]. Control and Decision, 2010, 25(6): 821-824.)

[5] Platt J. A resource-allocating network for function interpolation[J]. Neural Computation, 1991, 3(2): 213-225.

[6] Xiuju Fu, Lipo Wang. Data dimensionality reduction with application to simplifying RBF network structure and improving classification performance[J]. IEEE Trans on System Man Cybernetics B, 2003, 33(3): 399-400.

[7] Yingwei L, Sundararajan N, Saratchandran P. A sequential learning scheme for function approximation using minimal radial basis function(RBF) neural networks[J]. Neural Computation, 1997, 9(2): 461-478.

[8] Chitra Panchapakesan, Marimuthu Palaniswami, Daniel Ralph, et al. Effects of moving the centers in an RBF network. IEEE Trans on Neural Networks, 2002, 13(6): 1299-1307.

[9] Huang G B, Saratchandran P, Sundararajan N. A generalized growing and pruning RBF(GGAP-RBF) neural network for function approximation[J]. IEEE Trans on Neural Networks, 2005, 16(1): 57-67.

[10] Xabier Barandiaran, Alvaro Moreno. On the nature of neural information: A critique of the received view 50 years later[J]. Neurocomputing, 2008, 71(4/5/6): 681-692.

[11] Angelov P P, Filev D P. An approach to online identification of Takagi-Sugeno fuzzy models[J]. IEEE Trans on Systems, Man, and Cybernetics, 2004, 34(1): 484-498.

[12] 潘天红, 薛振框, 李少远. 基于减法聚类的多模型在线辨识算法[J]. 自动化学报, 2009, 35(2): 220-224.
(Pan T H, Xue Z K, Li S Y. An online multi-model identification algorithm based on subtractive clustering[J]. Acta Automatica Sinica, 2009, 35(2): 220-224.)

[13] Muzhou Hou, Xuli Han. Constructive approximation to multivariate function by decay RBF neural network[J]. IEEE Trans on Neural Networks, 2010, 12(9): 1517-1523.

[14] Panchapakesan C, Palaniswami M, Ralph D, et al. Effects of moving the centers in an RBF network[J]. IEEE Trans on Neural Networks, 2002, 13(6): 1299-1307.

[15] Honggui Han, Junfei Qiao. A self-organizing fuzzy neural network based on a growing-and-pruning algorithm[J]. IEEE Trans on Fuzzy Systems, 2010, 18(6): 1129-1143.