

非线性系统 RBF 网在线建模的 资源优化网络方法

魏海坤 宋文忠 李奇

(东南大学自动化研究所 南京 210096)
(E-mail: hkwei@seu.edu.cn)

摘要 提出了一种 RBF 网非线性动态系统在线建模的资源优化网络 (RON) 方法. RON 在资源分配网络的学习过程中引入了滑动窗口和网络结构在线优化的思想, 使网络能根据最近一段时间内的误差信息自动实现网络结构优化, 从而使 RBF 网既能在线适应对象的变化, 又能使网络规模维持在较小水平, 并保证了网络的泛化能力. 使用滑动窗口技术使 RON 对学习参数变化具有较好的鲁棒性, 并更易收敛. 三个标准例子演示了算法的有效性.

关键词 RBF 网, 资源优化网络, 滑动窗口, 泛化能力, 非线性系统, 在线建模

中图分类号 TP183

Resource-optimizing Networks for Nonlinear System On-line Modeling Using Radial Basis Function Networks

WEI Hai-Kun SONG Wen-Zhong LI Qi

(Institute of Automation, Southeast University, Nanjing 210096)
(E-mail: hkwei@seu.edu.cn)

Abstract An on-line radial basis function neural networks modeling method for nonlinear dynamics system, called resource-optimizing network (RON), is presented. RON introduces the ideal of sliding window and on-line structure optimization to the standard learning process of resource-allocating network. According to the newest error information, RON optimizes network structure on-line to adapt to the change of system dynamics, and maintains a compact network and satisfactory generalization. With the use of sliding window, RON is robust against the changing of learning parameters, and is easy to converge. Three benchmark examples demonstrate the effectiveness of the method.

Key words Radial basis function network, resource-optimizing network, sliding window, generalization ability, non-linear system, on-line modeling

1 绪论

径向基网络 (RBF 网) 在非线性系统在线建模中得到了较多的应用, 其中最著名的是资源分配网络 (RAN)^[1] 及 MRBF 方法^[2]. 这些方法的设计原则都是在线设计满足精度要求的神经网络, 以保证网络的泛化能力^[3]. RAN 在学习过程中循环地检查各样本输入输出

收稿日期 2003-9-19 收修改稿日期 2004-10-7

Received September 19, 2003; in revised form October 7, 2004

Copyright © 2005 by Editorial Office of Acta Automatica Sinica. All rights reserved.

对, 当新样本满足“新性”条件时, 则分配新节点, 否则用梯度法对网络进行权值修正; MRAN 对 RAN 做了改进: 除满足距离准则和误差准则外, 如果当前网络对过去连续多个样本偏差都过大则增加新隐节点, 而某隐节点对连续多个样本不被激活后就被删除. 尽管 RAN 和 MRAN 可以实现在线学习, 但这些方法的缺点是算法对学习参数变化的鲁棒性较差, 且网络的泛化能力不能保证.

因此, 本文提出一种设计具有鲁棒性的在线 RBF 网建模方法, 我们称之为资源优化网络 (Resource-optimizing network, 简称 RON). RON 根据最新的误差信息实现资源优化, 保证了网络结构的精简, 从而保证了 RBF 网的泛化能力. RON 实现资源优化的内容包括: 新隐节点的在线增加 (资源分配), 冗余隐节点的在线合并 (资源重组), 无用隐节点的在线删除 (资源释放). 采用滑动窗口思想使 RON 对学习参数变化具有较好的鲁棒性, 并更易收敛.

2 RON 在线建模方法

考虑一个正在进行在线学习的 $m-h-1$ 结构的 RBF 网, 其中 m 和 h 分别为网络的输入节点和隐节点数. 假定第 i 个隐节点的激活函数为 $\phi_i(\mathbf{x}) = \exp(-\|\mathbf{x} - \mathbf{c}_i\|^2/r_i^2)$, 其中 $\mathbf{x} = [x_1, x_2, \dots, x_m]^T \in R^m$ 为 RBF 网输入, $\mathbf{c}_i \in R^m$ 为该隐节点的中心, r_i 为该节点的扩展常数 (或称为宽度). 于是 RBF 网模型为 $f(\mathbf{x}) = \sum_{i=1}^h w_i \phi_i(\mathbf{x}) + b$, 其中, $f(\mathbf{x}) \in R$ 为网络的输出, 输出权值 $\mathbf{w} = [w_1, w_2, \dots, w_h]$, w_i 为第 i 个隐节点的输出权值, b 为输出偏移.

对于在线输入的样本 (\mathbf{x}_n, y_n) , RAN 和 MRAN 只用最新的单个样本训练网络, 容易陷入局部最优, 且抗劣质样本能力较差. 另外, 在节点中心调节过程中也未考虑隐节点的重合现象. 因此, 比较合理的在线建模方法是在学习过程使用最近的多个样本, 并动态调节网络参数: 在当前网络对样本偏差过大时在线生成隐节点, 偏差较小时则调节节点中心、输出权值和偏移, 当网络中产生重合的隐节点时则合并这些冗余的隐节点. 隐节点参数的梯度调节及隐节点删除策略请参考文献 [1,2], 隐节点合并的原则是当两个节点中心和宽度都非常接近则进行合并. 下面主要介绍滑动窗口思想及 RON 的算法实现.

2.1 滑动窗口

滑动窗口是一个固定长度的先进先出的队列, 队列中的元素则是在线输入的样本, 按进入窗口的时间顺序排列, (\mathbf{x}_1, y_1) 和 (\mathbf{x}_L, y_L) 分别是最老和最新的样本. 长度为 L 的滑动窗口可以表示为 $[(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_L, y_L)]$. 当新输入一个样本 (\mathbf{x}_n, y_n) 后, 该样本成为窗口中最新的样本, 同时如果窗口内样本数大于 L , 则最老的一个样本便滑出窗口. 窗口中的所有样本都是神经网络的训练样本.

RBF 网学习的目标函数为 $\sum_{i=1}^L \beta_i e_i^2$, 其中 e_i 为窗口中第 i 个样本的训练误差, β_i 为加权系数. β_i 的取值采用“遗忘因子”的思想: 如果采用线性遗忘方式, 加权系数 $\beta_i = 2i/[L(L+1)]$, 如采用指数遗忘方式, 则 $\beta_i = (1-\mu)/[(1-\mu^L)\mu^{L-i}]$, 其中 μ 为遗忘因子.

不管那种方式, 均有 $\sum_{i=1}^L \beta_i = 1$. 在线学习时, 通常认为最新的样本中含较多信息, 因此最新的样本具有较大的加权系数, 而较老的样本加权系数则较小. 本文的滑动窗口采用线性遗忘方式.

滑动窗口使用了最近一批样本的信息. 由权系数的分布可见, 信息越新就越重要. 与

仅使用单个样本相比, 使用滑动窗口的好处是可以使算法具有较好的鲁棒性, 并使参数容易整定.

2.2 RON 算法实现

综上所述, RON 在线建模算法实现如下:

开始时网络中无隐节点, 令偏移为第一个样本的输出: $b = y_0$

对每个在线输入的样本 (\mathbf{x}_n, y_n) :

将该样本移入滑动窗口, 同时移出最老的样本 (如果有的话);

累计各隐节点未被激活次数, 删除长期未被激活的隐节点;

计算 $\phi_i(\mathbf{x}_n)$, $f(\mathbf{x}_n) = \sum_{i=1}^h w_i \phi_i(\mathbf{x}_n) + b$, $e_n = y_n - f(\mathbf{x}_n)$, $d_n = \|\mathbf{x}_n - \mathbf{c}_{nearest}\|$;

如果距离准则和误差准则同时满足, 即 $d_n > \delta_{\min}$ 且 $\|e_n\| > \varepsilon_{\min}$, 则分配新隐节点:

$\mathbf{c}_{h+1} = \mathbf{x}_n$; $w_{h+1} = y_n - f(\mathbf{x}_n)$; $r_{h+1} = \kappa d_n$;

否则, 按下面方式调节网络参数, 直至达到给定训练次数, 或滑动窗口内样本的训练误差小于给定值. 最后检查是否有冗余的隐节点, 合并这些隐节点;

$\nabla_{\mathbf{c}_i} f(\mathbf{x}) = 2w_i \phi_i(\mathbf{x})(\mathbf{x} - \mathbf{c}_i)/r_i^2$, $\nabla_{r_i} f(\mathbf{x}) = 2w_i \phi_i(\mathbf{x})\|\mathbf{x} - \mathbf{c}_i\|^2/r_i^3$;

$\Delta \mathbf{c}_i = \eta \beta_i e_i \sum_{j=1}^L \nabla_{\mathbf{c}_i} f(\mathbf{x}_j)$, $\Delta r_i = \eta \beta_i e_i \sum_{j=1}^L \nabla_{r_i} f(\mathbf{x}_j)$;

$\mathbf{c}_i(n) = \mathbf{c}_i(n-1) + \Delta \mathbf{c}_i(n) + \alpha \Delta \mathbf{c}_i(n-1)$;

$r_i(n) = r_i(n-1) + \Delta r_i(n) + \alpha \Delta r_i(n-1)$;

$\hat{A}^+ = (\hat{A} \hat{A}^T)^{-1}$, $W = \hat{A}^+ \mathbf{y}$;

$\mathbf{w}(n) = [W(1), W(2), \dots, W(M)]$, $b(n) = W(M+1)$;

3 仿真研究

3.1 静态函数逼近问题

该问题用于验证提出的算法能否设计出与实际一致的最小的 RBF 网^[2]. 非线性目标函数可由 6 个隐节点的 RBF 网实现, 6 个隐节点的参数见表 1. 产生 121 个训练样本 $((x_1, y_2), y)$, 其中 (x_1, y_2) 为输入, y 为输出, $x_i \in \{0, 0.1, \dots, 1\}$, $i \in \{1, 2\}$.

表 1 目标函数的实际值和估计值对比
Table 1 True and estimated centers, widths and weights

实际中心	(0.3,0.2)	(0.7,0.2)	(0.1,0.5)	(0.9,0.5)	(0.3,0.8)	(0.7,0.8)
估计中心	(0.301,0.200)	(0.699,0.198)	(0.097,0.500)	(0.900,0.500)	(0.300,0.800)	(0.699,0.802)
实际宽度	0.1	0.1	0.144	0.144	0.1	0.1
估计宽度	0.098	0.099	0.139	0.149	0.099	0.099
实际权值	1.0	1.0	1.0	1.0	1.0	1.0
估计权值	1.038	1.008	1.013	0.966	1.040	1.004

表 1 显示了学习后得到节点中心、扩展常数 (宽度) 及输出权值的估计值与实际值的对比, 可见两者非常接近. 由表 1 可见, RON 能较好地实现给定的最小 RBF 网. 另外, 我们的结果也远优于文献 [2] 的结果.

3.2 时变非线性系统辨识

一个 SISO 的时变非线性系统的例子^[4]

$$y(k) = \frac{29\beta(k)}{40} \sin\left(\frac{16u(k-1) + 8y(k-1)}{\beta(k)(3 + 4u(k-1)^2 + 4y(k-1)^2)}\right) + \frac{2}{10}u(k-1) + \frac{2}{10}y(k-1)$$

其中 $u(k)$ 为输入信号, 取 $[-1, 1]$ 内均匀分布的随机值. 时变参数 $\beta(k)$ 开始取 1.0, 在 $k = 1501$ 时刻取为 0.9, 在 $k = 2501$ 时刻取为 0.8. 令 $y(0) = 0$, 按上式产生 5000 个训练样本.

图 1 演示了某次训练中隐节点数的变化. 可见, $\beta(k) = 1.0$ 时, 大约在 $k = 600$ 左右, RON 就用 8 个隐节点基本拟合了目标函数; $\beta(k) = 0.9$ 时, 神经网络隐节点数变为 9, 以适应对象参数的变化; $\beta(k) = 0.8$ 时, 神经网络隐节点数增加到 10, 此后不再变化.

我们使用的误差准则为^[4] $I_d(i) = \frac{1}{L} \sum_{p=0}^{M-1} |u(i-p) - \hat{y}(i-p)|$, 并绘制了训练过程中 I_d 的演变情况, 如图 2 所示. 由图 2 可见, 由于对象参数的变化, 在 $k = 1501$ 和 $k = 2501$ 时 I_d 都有明显的增加, 但都很快被 RON 抑制. 训练结束后 MRAN 和 RON 的隐节点数分别为 11 和 10, 而两者在 $k = 1500$ 到 $k = 5000$ 间 I_d 的均值分别为 0.0326 和 0.0311. 可见, RON 的收敛速度, 训练误差和最终的隐节点数均优于 MRAN.

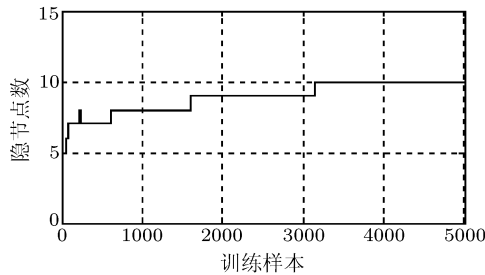


图 1 训练过程中隐节点数变化情况

Fig. 1 Evolution of hidden units

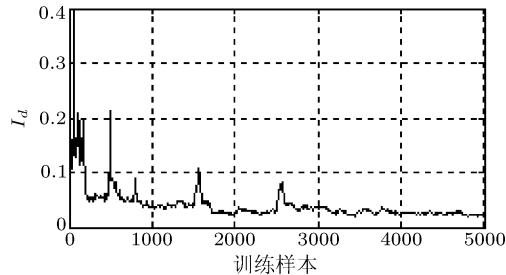


图 2 训练过程中误差 I_d 变化情况

Fig. 2 Evolution of error (I_d)

另外, 由于训练样本是随机的, 因此训练结果也具有一定的随机性. 我们作了 10 次试验, 隐节点数均在 10~13 间, I_{dav} 则在 0.0291~0.0407 间, 这是一个相当稳定的结果.

3.3 混沌时间序列预测

该试验中, 我们用 RON 对 MG 混沌时间序列进行建模和预测, 并研究了 RON 对学习参数变化的鲁棒性. 产生 MG 时间序列的微分方程为 $\frac{dx(t)}{dt} = \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t)$, 用四阶龙格-库塔方法求解该方程, 方程中的参数设置为: 时延 $\tau = 17$, 初值取 $x(0) = 1.2$, 且当 $t < 0$ 时 $x(t) = 0$. 预测的任务是根据输入 $[x(t-18) x(t-12)x(t-6) x(t)]$ 预测 $x(t+50)$ 的值. 我们用 1500 个样本进行训练 ($t = 201$ 到 1700), 用 500 个样本进行测试 ($t = 3001$ 到 3500).

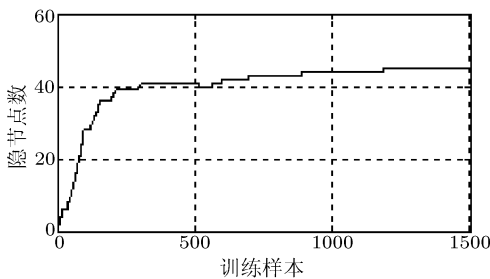


图 3 训练过程中隐节点数变化情况

Fig. 3 Evolution of hidden units

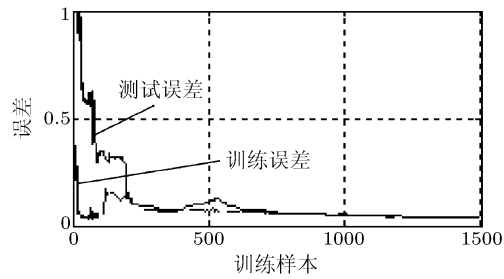


图 4 训练过程中训练误差和测试误差变化情况

Fig. 4 Evolution of training error and normalized RSME

学习过程中, RBF 网隐节点数的变化情况如图 3 所示, 图 4 记录了窗口内样本训练误差 (虚线) 和测试误差 (实线) 的变化情况. 1500 个样本学习结束后, 网络中隐节点数为

45, 对 500 个样本的测试误差为 0.0427. 由图 3 和图 4 可见, 我们仅用了不到 1500 个样本便对 MG 时间序列实现了正确学习, 且无论是隐节点数还是测试误差均优于资源分配网络的结果^[1].

前面提过, RON 对学习参数变化有较好的鲁棒性. 为了验证这一点, 我们改变主要学习参数的值, 如 ε_{\min} , δ_{\min} , 窗口宽度和最大训练次数 MaxEpoch, 并重新试验, 结果如表 2 所示. 由表可见, 当这些学习参数变化时, RON 的隐节点数和测试误差波动远小于传统的 RAN 方法, 说明 RON 具有较好的鲁棒性, 因此 RON 的参数也相对容易整定.

表 2 不同学习参数下的结果

Table 2 Result comparison with different learning parameters

ε_{\min}	δ_{\min}	L	MaxEpoch	隐节点数	测试误差
0.010	0.15	300	200	45	0.0427
0.005	0.15	300	200	62	0.0668
0.015	0.15	300	200	46	0.0622
0.010	0.18	300	2007	35	0.0540
0.010	0.12	300	200	62	0.0393
0.010	0.15	250	200	46	0.0443
0.010	0.15	350	200	45	0.0430
0.010	0.15	300	100	45	0.0427

4 讨论

本文提出了一种利用滑动窗口的 RBF 网在线建模方法, 讨论了网络中结构和权值参数的在线自适应调节问题, 包括隐节点的增加、合并、删除等操作, 及节点中心和权值等其它参数的在线调整, 目的是使在线设计满足精度要求的较小结构的神经网络, 以保证其泛化能力. 三个 benchmark 仿真例子分别演示了 RON 的最小 RBF 网建模能力、适应被建模对象参数时变的能力, 及算法对学习参数变化的鲁棒性.

References

- 1 Platt J. A resource-allocating network for function interpolation, *Neural Computation*, 1991, **2**(3): 213~225
- 2 Yingwei L, Sundararajan N, Saratchandran P. Identification of time-varying nonlinear systems using minimal radial basis function neural networks, *IEE Proc.-Control Theory Application*, 1997, **144**(2): 202~208
- 3 Wei H, Xu S, Song W. Generalization theory and generalization methods for neural networks, *Acta Automatica Sinica*, 2001, **27**(6): 806~815
- 4 Li Y, Sundararajan N, Saratchandran P. Analysis of minimal radial basis function neural networks algorithm for real-time identification of nonlinear dynamic systems, *IEE Proc.-Control Theory Application*, 2000, **147**(4): 476~484

魏海坤 副教授, 研究领域为神经网络理论及应用.

(**WEI Hai-Kun** Associate professor. His research interests include neural network modeling.)

宋文忠 教授, 博士生导师, 研究领域为复杂系统辨识及控制.

(**SONG Wen-Zhong** Professor. His research interests include system identification and control.)

李奇 教授, 博士生导师, 研究领域为过程控制及优化.

(**LI Qi** Professor. His research interests include process control and optimization.)