计算机科学1998Vol. 25№. 6

基于逐层优化递推最小二乘算法的反馈神经网络*)

A Recurrent Neural Network Based on the OLL-RLS Algorithm

帅藕莲 周三川 郑咸义 尹俊勋 冯久超

(华南理工大学电子与通信工程系 广州510641)

摘 要 This paper has developed a novel training algorithm for recurrent neural networks, Optimization Layer-by-Layer Recursive Least Squares, OLL-RLS. This algorithm overcomes the numerical instability and covergence installing resulted from common RLS in the training, thus the covergence is accelerated and the weights construction is optimal. Simulation results have demonstrated that the network trained by our developed algorithm possesses strong generality.

关键词 Recurrent Neural Networks (RNNs); Optimization Layer-by-Layer Recursive Least Squares, OLL-RLS; Numerical Unstability; Covergence Installing

计算智能(CI)是当前智能科学研究的新热点，其积极意义在于推出新的、功能更强大的、具有更普遍意义的计算智能模型或方法。神经网络的学习能力及并行计算结构，成为其中一个主要方面。前馈神经网络中的神经元以不同方式嵌入反馈连接，则可构成不同的反馈神经网络结构，这样，过去事件或状态以不同方式记忆在网络中。显然，就本身结构特点而言，反馈神经网络比前馈神经网络更能描述系统的动态特性[1]。事实上，大多数物理系统呈现出动态特性。由此，近十年来，此方面的研究吸引了不少的学者与专家，且对于不同的应用设计出不同的结构及有效的训练算法[2-9]。最近，文献[10-11]中，线性逐层优化最小二乘法为加速前馈神经网络训练过程提供了重要突破。随后，文献[12]推广了逐层优化算法，并运用到混沌时间序列预测中。然而，以上算法在训练样本较大时要求大量的计算存储且不能提供确定的收敛结果。本文把 OLL-RLS 推广到反馈神经网络的训练。

1. 学习算法

1.1 OLL-RLS 算法的由来

为讨论的简单性，我们考虑只有一隐层且只在隐层进行互连接(反馈连接)的反馈神经网络结构。假设网络有 r 个输入，s 个隐节点，m 个输出。则反馈神经网络的一般形式可表示为：

$$y_j=\sigma_j\Big\{\sum_{i=1}^{r}w_{ji}^{s}x_i+\sum_{q=1}^{s}w_{jq}^{F}z^{-1}y_q+w_j^{B}\Big\}\qquad 1\leqslant j\leqslant s;\ 1\leqslant i\leqslant r \tag{1.1a}$$

$$z_k=\sum_{j=1}^{n}v_{kj}^{s}y_j+v_k^{B}\qquad 1\leqslant k\leqslant m \tag{1.1b}$$

其中，z_k 表示第 k 个输出节点的输出，v_{kj}^{s} 为从第 j 个节点到第 k 个节点的连接权值，v_k^{B} 为第 k 个输出节点的偏置。$\sigma_j\{\cdot\}$ 表示非线性单调函数，常取为 *Sigmoid* 函数形式或正切函数形式，x_i 表示第 i 个输入节点，y_j 表示第 j 个隐节点的输出，$z^{-1}y_q$ 表示第 q 个隐节点输出的单位延迟量，w_{ji}^{s} 表示从第 i 个输入节点到第 j 个隐节点的连接权值，w_{jq}^{F} 表示第 q 个隐节点到第 j 个隐节点的反馈连接权值，w_j^{B} 为第 j 个隐节点的偏置量。

若可获得 P 个样本，算法的目的是得到网络的优化权值，从而使输出误差的平方最小。这里，定义误差代价函数为：

$$E_T=\frac{1}{2}\sum_{t=1}^{P}\beta^{P-t}\sum_{k=1}^{m}\|e_{vk}\|^2$$

$$e_{vk}=d_{vk}-Y_{vk} \tag{1.2}$$

其中，$\|e\|$ 表示向量 e 的欧几里得范数，β 为遗忘因子，选为小于但接近于1的正数。

$d_{vk}=[d_{vk1},d_{vk2},\cdots\cdots,d_{vkP}]^T$ 为第 k 个输出节点相应于 P 个样本的理想输出。

*)此课题得到国家自然科学基金资助。

$v_k=[v_{k1}^S, v_{k2}^S, \cdots\cdots, v_{ks}^S, v_k^B]^T$ 为隐节点到第 k 个输出节点的权向量。

$$Y=\begin{bmatrix} \sigma_{11}\{x_1w_1\} & \sigma_{21}\{x_1w_2\} & \cdots & \sigma_{s1}\{x_1w_s\} & 1 \\ \sigma_{12}\{x_2w_1\} & \sigma_{22}\{x_2w_2\} & \cdots & \sigma_{s2}\{x_2w_s\} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \sigma_{1P}\{x_Pw_1\} & \sigma_{2P}\{x_Pw_2\} & \cdots & \sigma_{sP}\{x_Pw_s\} & 1 \end{bmatrix}$$ 为隐层的输出所构成的矩阵。

$x_t=[x_1, x_2, \cdots, x_t, z^{-1}y_1, z^{-1}y_2, \cdots, z^{-1}y_s, 1]$ 表示由网络输入及 t 时刻所有隐节点输出的单位延迟量所构成的向量(或者可说是 t 时刻某一隐节点的等价输入向量)，$w_j=[w_{j1}^S, w_{j2}^S, \cdots, w_{jr}^S, w_{j1}^F, w_{j2}^F, \cdots, w_{js}^F, w_j^B]^T$ 表示某一时刻所有隐节点及所有输入到第 j 个隐节点的连接权向量(或者可说是某时刻第 j 个隐节点等价输入向量的加权向量)，须注意的是：x_t 及 w, 均考虑了反馈连接及偏置。

为计算网络的优化权值，我们将 E_T 对权向量 v_k 求偏微分且令所得微分值为0，即：

$$\frac{\partial E_T}{\partial v_k}=\sum_{t=1}^{P}\beta^{P-t}(Y^TYv_k-Y^Td_{vk})=0 \qquad (1.3)$$

从而可得权向量的优化值 v：

$$V(n)=\Phi_v(n)^{-1}\theta_v(n) \qquad (1.4)$$

式中，$V=[v_1, v_2, \cdots, v_m]$ 为隐层到输出层的权连接。$\Phi_v(n)=\sum_{t=1}^{P}\beta^{P-t}Y(t)^TY(t)$ 表示网络输出 Y(t)的自相关矩阵，大小为(s+1)×(s+1)；$\theta_v(n)=\sum_{t=1}^{P}\beta^{P-t}Y(t)^Td_v(t)$ 表示网络输出 Y(t)与理想输出 $d_v(t)$ 的互相关矩阵，大小为 m×(s+1)。

类似地，我们可以得到从输入层到隐层的连接权值的优化值，即：

$$\frac{\partial E_T}{\partial w_j}=\sum_{t=1}^{P}\beta^{P-t}\frac{\partial Y}{\partial w_j}\frac{\partial E_T}{\partial Y}=0 \qquad (1.5)$$

若定义

$$\frac{\partial E_T}{\partial Y}=-M_ke_{vk} \qquad (1.6a)$$

以及

$$\frac{\partial Y}{\partial w_j}=X^TF'\{XW\} \qquad (1.6b)$$

其中，$M_k=[v_k \quad v_k \cdots v_k]_{P\times(s+1)}$，$X=[x_1, x_2, \cdots, x_P]^T$ 表示由 P 个训练样本输入模式所构成的矩阵，

$$F'\{XW\}=\begin{bmatrix} \sigma_{11}'\{x_1w_1\} & \sigma_{21}'\{x_1w_2\} & \cdots & \sigma_{s1}'\{x_1w_s\} \\ \sigma_{12}'\{x_2w_1\} & \sigma_{22}'\{x_2w_2\} & \cdots & \sigma_{s2}'\{x_2w_s\} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1P}'\{x_Pw_1\} & \sigma_{2P}'\{x_Pw_2\} & \cdots & \sigma_{sP}'\{x_Pw_s\} \end{bmatrix}$$

为隐节点非线性激励函数微分所构成的矩阵形式。

把式(1.6a)与(1.6b)代入(1.5)式，有

$$\frac{\partial E_T}{\partial w_j}=-\sum_{t=1}^{P}\beta^{P-t}X^TF'\{XW\}M_ke_{vk}=0 \qquad (1.7)$$

进一步可得

$$e_{wj}=F'\{XW\}M_ke_{vk} \qquad (1.8)$$

式中，e_{wj} 定义为隐层中网络输出与理想输出间误差的列向量。若定义 $d_{wj}=Xw_j+e_{wj}$，其中，$d_{wj}=[d_{wj1}, d_{wj2}, \cdots, d_{wjP}]^T$ 表示第 j 个隐节点理想值所构成的向量。这样，(1.7)式可表达成：

$$\partial E_T/\partial w_j=-\sum_{t=1}^{P}\beta^{P-t}X^T(d_{wj}-X_{wj})=$$

$$\sum_{t=1}^{P}\beta^{P-t}(X^TXw_j-X^Td_{wj})=0 \qquad (1.9)$$

因而，(1.9)式可写成(1.4)式相似的形式。可见，从输入层到隐层的连接权矩阵的最优值可通过逐层优化最小二乘法计算出来，即：

$$W(n)=\Phi_w(n)^{-1}\theta_w(n) \qquad (1.10)$$

其中，$W=[w_1, w_2, \cdots, w_s]$ 表示从输入层到隐层的连接权矩阵。$\Phi_w(n)=\sum_{t=1}^{P}\beta^{P-t}X(t)^TX(t)$ 为网络输入 X(t)的自相关矩阵，大小为(r+1)×(r+1)，$\theta_w(n)=\sum_{t=1}^{P}\beta^{P-t}X(t)^Td_w(t)$ 为网络输入 X(t)与隐层理想值 $d_w(t)$ 的互相关矩阵，大小为 s×(r+1)。

由式(1.4)及(1.10)可见，为求得最优权矩阵 V(n)与 W(n)需要确定相关矩阵 $\Phi_v(n)$ 与 $\Phi_w(n)$ 的逆矩阵。由于逆矩阵的计算很复杂，且消耗时间多，特别在权数很大的情况下尤其明显。这样，事实上常避免矩阵求逆运算。更则，当 P<m 或 P<s 时相关矩阵可能为奇异矩阵，详述请参见文献[13]。因此，采用 RLS 确定相关矩阵。

若设 n≈p，则$(\Phi_v(n)\theta_v(n))$及$(\Phi_w(n), \theta_w(n))$的更新形式为：

$$\Phi_v(n)=\beta\Phi_v(n-1)+Y(n)^TY(n)$$

$$\theta_v(n)=\beta\theta_v(n-1)+Y(n)^Td_v(n)$$

$$\Phi_w(n)=\beta\Phi_w(n-1)+X(n)^TX(n)$$

$$\theta_w(n)=\beta\theta_w(n-1)+X(n)^Td_w(n)$$

$P_v(n)=\Phi_v(n)^{-1}$及 $P_w(n)=\Phi_w(n)^{-1}$，利用矩阵求逆引理[14]，可得以下递推方程：

$$K_v(n)=\frac{1}{\beta}P_v(n-1)Y(n)^T\Big[1+\frac{1}{\beta}\sum_{t=1}^{P}y_t(n)P_v(n-1)y_t(n)^T\Big]^{-1} \qquad (1.11)$$

$$P_v(n)=\frac{1}{\beta}[P_v(n-1)-K_v(n)Y(n)P_v(n-1)]$$

(1.12)

$$K_w(n)=\frac{1}{\beta}P_w(n-1)X(n)^T\Big[1-\frac{1}{\beta}\sum_{i=1}^{p}x_i(n)P_w(n-1)x_i(n)^T\Big]^{-1} \quad (1.13)$$

$$P_w(n)=\frac{1}{\beta}[P_w(n-1)-K_w(n)X(n)P_w(n-1)] \quad (1.14)$$

其中，I 为单位矩阵 P，$P_v(n)$和 $K_w(n)$表示 RLS 中的卡尔曼增益。

权矩阵 V(n)、W(n)的更新方程分别为：

$$V(n)=V(n-1)+K_v(n)[d_v(n)-Y(n)V(n-1)] \quad (1.15)$$

$$W(n)=W(n-1)+K_w(n)[d_w(n)-X(n)W(n-1)] \quad (1.16)$$

递推算法中所要求的初始值分别取为：$P_v(0)=\delta I_v$，$P_w(0)=\delta I_w$，δ 为一很小的正数，使 $P_v(0)$与 $P_w(0)$为正交矩阵；V(0)与 W(0)可为非零随机数。

1.2 克服 OLL-RLS 算法的数字不稳定性及收敛拖延性

由上述的 OLL-RLS 算法的由来可知，训练反馈神经网络中，没有计算动态微分及矩阵求逆，从而得到快速收敛率。然而，OLL-RLS 当用于实际网络训练时，常出现数字不稳定性及收敛拖延性。根据文献[14]，RLS 算法中的不稳定性及收敛拖延性类似于卡尔曼算法，这些问题的出现应追踪式(1.11)与(1.13)中的矩阵 $P_v(n)$与 $P_w(n)$，这两个矩阵是由两个非负定矩阵的差而计算出来。根据文献[14,15]中的结果可知，数字不稳定性及收敛拖延性出现于训练过程中权值不变化的情况中，特别在 $P_v(n)$与 $P_w(n)$都不具备正定性且很小时出现，这样乘因子 $P_v(n)$与 $P_w(n)$相当于乘因子零矩阵。相应地，用一启发性机理克服以上困难，即附加一人为噪声于递推方程中。这样，(1.12)变为：

$$P_v(n)=\frac{1}{\beta}[P_v(n-1)-K_v(n)Y(n)P_v(n-1)]+\Gamma_v(n) \quad (1.17)$$

式中 $\Gamma_v(n)$为双角矩阵，其元素很小且为非负，一般取为高斯分布随机数($10^{-6}\sim10^{-2}$)。相应地，权矩阵 $\hat{V}(n)$更新方程为：

$$\hat{V}(n)=V(n-1)+K_v(n)[d_v(n)-Y(n)V(n-1)]+\alpha_v\Delta_v \quad (1.18)$$

式中 α_v 表示一很小的正数且渐渐增加一很小量直至权值重新开始变化，Δ_v 为 v 维空间中 N(0,1)高斯噪声矩阵。同样地，式(1.14)变为：

$$P_w(n)=\frac{1}{\beta}[P_w(n-1)-K_w(n)X(n)P_w(n-1)]+\Gamma_w(n) \quad (1.19)$$

$\Gamma_w(n)$与 $\Gamma_v(n)$服从同样的规则。权矩阵 $\hat{W}(n)$变为：

$$\hat{W}(n)=W(n-1)-K_w(n)[d_w(n)-X(n)W(n-1)]-\alpha_w\Delta_w \quad (1.20)$$

α_w、Δ_w 分别与 α_v、Δ_v 遵循同样规则。

可见，OLL-RLS 算法的改善是通过引入人为噪声与渐增参数而实现的，必须注意到渐增参数的重要性。如果渐增参数增加太大，有可能失去脱离障碍的机会；若增加得太小，则收敛变慢而使算法失效。根据经验，渐增参数可按初始值的0.5%增加，直至权值重新开始调整。

2 OLL-RLS 算法讨论

2.1 OLL-RLS 算法收敛性分析

用于神经网络训练的 RLS 中，着重考虑的是所估计的权值($\hat{V}(n)$、$\hat{W}(n)$)随迭代次数 $n\to\infty$时的逼近特性，因而，有必要讨论 OLL-RLS 算法的收敛性。由于 OLL-RLS 算法是逐层优化，这样，为讨论的简单，关于 OLL-RLS 算法的收敛性分析只针对一层而言，下标"V"及"W"均省略。

从相关矩阵 $\Phi(n)$的定义，可得：

$$\Phi'(n)=\Phi(n)+\delta\beta^pI,=\sum_{i=1}^{p}\beta^{p-i}X(i)^TX(i)+\delta\beta^pI, \quad (2.1)$$

式中 $\delta\beta^pI$ 由初始化引起，现假设 $d(i)=X(i)W_0+e(i)$，其中 W_0为权值的最优解。

这样，所估计的权值 $\hat{W}(n)$为

$$\begin{aligned}\hat{W}(n)&=\Phi'(n)^{-1}\theta(n)\\&=[\Phi(n)+\delta\beta^pI]^{-1}\cdot\sum_{i=1}^{p}\beta^{p-i}X(i)^Td(i)\\&=[\Phi(n)+\delta\beta^pI]^{-1}\sum_{i=1}^{p}\beta^{p-i}X(i)^T[X(i)W_0+e(i)]\\&\vdots\\&=[I+\delta\beta^p\Phi(n)^{-1}]^{-1}W_0\end{aligned} \quad (2.2)$$

基于上式有关符号意义，我们有以下关于 OLL-RLS 算法的一般收敛性定理：

定理 若令 R 为集合平均(Ensemble-Averaged)矩阵，$R\approx\frac{1}{n}\sum_{i=1}^{p}X(i)^TX(i)=\frac{1}{n}\Phi(n)$，并设所估计的权值与其最优值的差为权值误差 $\varepsilon_w(n)\approx\hat{W}(n)-W_0$，则有

$$E\{\varepsilon_w(n)\}=-\frac{\delta}{n}R^{-1}W_0 \quad (2.3)$$

可见，随迭代次数 $n\to\infty$权值误差的均值 $E\{\varepsilon_w(n)\}\to0$，换言之，当 $n\to\infty$时有 $E\{\hat{W}(n)\}\to W_0$成立，因

而本文所推广的OLL-RLS算法是收敛的。

证明：根据式(2.1)且设β接近于1，有

$$\begin{aligned}E\{\varepsilon_w(n)\}&=E\{\hat{W}(n)-W_0\}\\&=E\{[I+\delta\beta^F\Phi(n)^{-1}]^{-1}W_0-W_0\}\\&=E\{([I+\delta\Phi(n)^{-1}]^{-1}-I)W_0\}\\&=E\{[I+\delta\Phi(n)^{-1}]^{-1}[I-\\&\quad I-\delta\Phi(n)^{-1}]\}W_0\\&\approx-\delta\Phi(n)^{-1}W_0,\text{for }\delta<<1\end{aligned}\quad(2.4)$$

由(2.3)式及矩阵R的定义可推得(2.2)式是成立的，从而收敛定理得证。

2.2 OLL-RLS算法与其它算法有关计算复杂性的比较

关于RLS、EKF及DEKF算法的计算复杂性主要取决于矩阵P(n)所需的计算量及存储要求。由于本文所推广的OLL-RLS算法是采用逐层优化法，这样，所需的计算量及存储要求远远小于EKF与DEKF算法，并且OLL-RLS算法没有计算动态梯度，这更有利于减少计算复杂度。表1中列出了OLL-RLS算法、动态BP算法、EKF算法和DEKF算法的计算复杂度(以乘法次数作为度量)。由表可见，OLL-RLS算法所需代价高于动态BP算法但低于EKF算法和DEKF算法。

3 仿真实验

3.1 用于标准的非线性系统辨识

考虑一高阶非线性系统模型为：

$$\begin{aligned}x_1(t+1)&=-0.4x_2(t)+x_3(t)+0.3x_4(t)\\x_2(t+1)&=tanh(0.5x_1(t)+x_3(t)+(1+0.2x_2(t))u(t)+0.7x_4(t))\\x_3(t+1)&=tanh(-0.6x_1(t)+0.4x_2(t)+0.3x_1(t)x_4(t))\\x_4(t+1)&=tanh(x_3(t)+0.6x_2(t)x_4(t))\\y(t+1)&=(x_1(t))^2+0.8x_4(t)\end{aligned}\quad(3.1)$$

其中，$\tanh(x)=\frac{1-e^{-x}}{1+e^{-x}}$，u(t)为系统的输入，y(t)为输出，$x_i(t)$，i=1，2，3，4为系统的内部状态。输入信号{u(t)}为均匀分布在[-1，1]的随机序列，训练中采取200个样本。为测试所构造的网络模型的推广性，我们采用300个样本作为测试集，但测试样本分别对应不同的输入响应：高斯随机输入，阶跃输入及正弦输入信号；即：

$$u(t)=\begin{cases}N(0,1) & 1\leqslant t\leqslant 100\\0.5 & 101\leqslant t\leqslant 200\\10\sin\left(\frac{t-200}{10}\right) & 201\leqslant t\leqslant 300\end{cases}\quad(3.2)$$

其中，N(0，1)表示均值为0，方差为1的高斯随机信号。

本文所推广的OLL-RLS算法，带固定学习率的动态BP算法及带自调节学习率的快速BP算法用于训练反馈神经网络，网络权值初始化为均匀分布在[-1.0，1.0]之间的随机数。文中所提出的算法均用PC Matlab软件编写，而其它BP算法则可以从PC Matlab平台的神经网络工具箱中获得，在PC-pentium/120MHz的计算机上运行，把均方根误差(RMSE)小于或等于0.05作为训练过程终止准则，其中RMSE定义为：

$$RMSE=\sqrt{\frac{\sum_{i=1}^{P}\sum_{k=1}^{m}(d_{ok}-z_{ik})^2}{P\cdot m}}\quad(3.3)$$

所报导的结果均为20次不同初始环境所得结果的平均。图1、图2分别展示了RMSE随迭代次数变化的关系及相应的测试误差响应；不同算法对不同网络、不同驱动信号的结果比较列于表2中。可见，基于OLL-RLS算法的反馈神经网络有较好的推广性，并且以最快的速度收敛。

3.2 用于铁路运输系统辨识

此部分实验所用的振动信号是通过安装在大约30mph速率运行的铁路运输系统中一特殊轴承上的测试仪(Accelerometer)所测得的。在有负载(33000lbs)与无负载(8000lbs)情况下以27kHz的采样频率分别取1秒钟记录作为每一数据文件，即每一文件长度为27k，为得到辨识模型，我们利用序列号1～256作为训练样本，必须强调的是，我们的实验是基于振动信号的频域(采取512点FFT)而言的。为测试所构造模型的特性，我们采用不同工作状态下的两组测试集：序列号从512～1024及12800～13312。所构造的网络结构为1个输入单元，10个隐单元，1个输出单元。2000次迭代的不同训练算法的辨识测试结果见图3(略)。可见，基于OLL-RLS算法的反馈神经网络模型具有很强的辨识性及推广性，并且根据不同工作状态的辨识误差，可初步确定无负载时输出误差范围为[-0.2，0.3]，而有负载输出误差在[-0.4，0.4]内。另外，我们叠加一均值为有载时正常信号基频、方差为0.5的频域高斯噪声信号所得的信号作为故障信号，其相应的两组测试误差信号分别展示于图4(略)中，误差范围分别为[-2.5，+1.0]与[-1.5，+1.5]。可见，误差范围远远高于正常信号的误差指标。因此，基于OLL-RLS算法的反馈神经网络模型不仅可用于实际系统辨识，还可用于系统错误诊断。

结论 本文改进了用于训练反馈神经网络的一

种新算法:OLL-RLS.此算法是采用标准的RLS对网络权值进行逐层优化,这样,与其它RLS或卡尔曼滤波算法相比,大大减少了计算复杂性,并且,通过引入人为噪声的启发性机理,克服了RLS用于网络训练中的数字不稳定性及收敛拖延性,加速了收敛进程,得到最优权值结构,仿真结果表明,基于OLL-RLS算法的反馈神经网络具有较强的辨识性及推广性;还可用于实际物理系统辨识。(参考文献共15篇,略)

表1 动态BP,EKF,DEKF与OLL-RLS算法复杂度的比较(M,N分别为输出层、隐层中权值数,在此仅假设三层网络)

算法	计算复杂度	动态微分的计算
动态BP	$\Theta(M+N)$	要求
EKF	$\Theta((M+N)^2)$	要求
DEKF	$\Theta(\sum_{i=1}^{g}(m_i+n_i)^2)$	要求
OLL-RLS	$\Theta(M^2+N^2)$	不要求

表2 不同算法对不同网络响应于高阶系统不同驱动信号的结果比较
(输入元、隐单元、输出元分别记为NI,NH,NO;NI=4,NO=1)

算法		OLL-RLS		动态BP		快速动态BP	
网络大小		NH=12	NH=22	NH=12	NH=22	NH=12	NH=22
到达收敛准则所需的次数		171	89	>2000	>2000	>2000	>2000
训练完的RMSE		0.0771	0.0352	0.173	0.1014	0.173	0.0976
对不同驱动信号样本的测试RMSE	高斯信号	0.111	0.0801	0.165	0.146	0.164	0.134
	阶跃信号	0.1409	0.0255	0.143	0.0949	0.137	0.0961
	正旋信号	0.1407	0.0519	0.148	0.102	0.145	0.0965

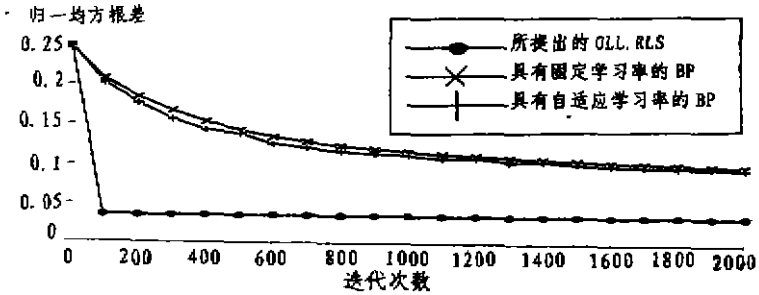

图1 由动态BP、快速BP、OLL-RLS算法训练的反馈神经网络对高阶系统响应的收敛曲线

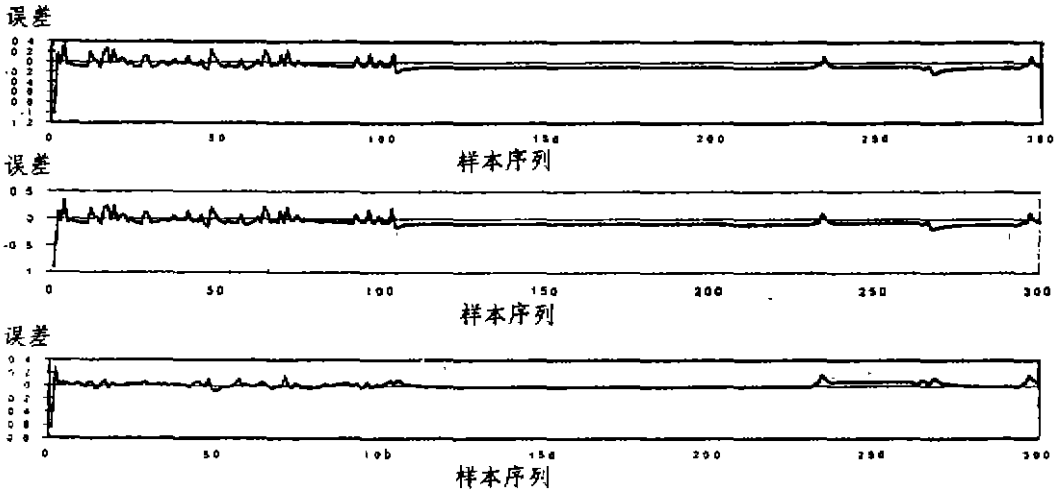

图2 从上到下分别为动态BP、快速BP、OLL-RLS算法对高阶系统的辨识结果