

# 一种回归神经网络的快速在线学习算法<sup>1)</sup>

韦 巍

(浙江大学电机系 杭州 310027)

**摘 要** 针对回归神经网络 BP 学习算法收敛慢的缺陷,提出了一种新的快速在线递推学习算法.本算法在目标函数中引入了遗忘因子,并借助于非线性系统的最大似然估计原理成功地解决了动态非线性系统回归神经网络模型权系数学习的实时性和快速性问题.仿真结果表明,该算法比传统的回归 BP 学习算法具有更快的收敛速度.

**关键词** 回归神经网络,在线学习,快速算法.

## 1 引言

传统的前向传播神经网络在动态时序信号处理、非线性动态系统控制等带有强时序行为系统的应用中存在相当大的困难.目前虽然有通过在网络中引入时滞环节来描述系统的动态性能<sup>[1]</sup>,但是能够更直接更生动地反映系统动态特性的网络应该是动态神经网络,它包含网络内部状态的反馈.回归神经网络就是利用网络的内部状态反馈来描述系统的非线性动力学行为.构成回归神经网络模型的方法有很多,但总的思想都是通过对前馈神经网络中加入一些附加的、内部的反馈通道来增加网络本身处理动态信息的能力.例如根据状态信息的反馈途径不同可构成两种不同的回归神经网络结构模型:Jordan 型(如图 1(a))和 Elman 型(如图 1(b)).

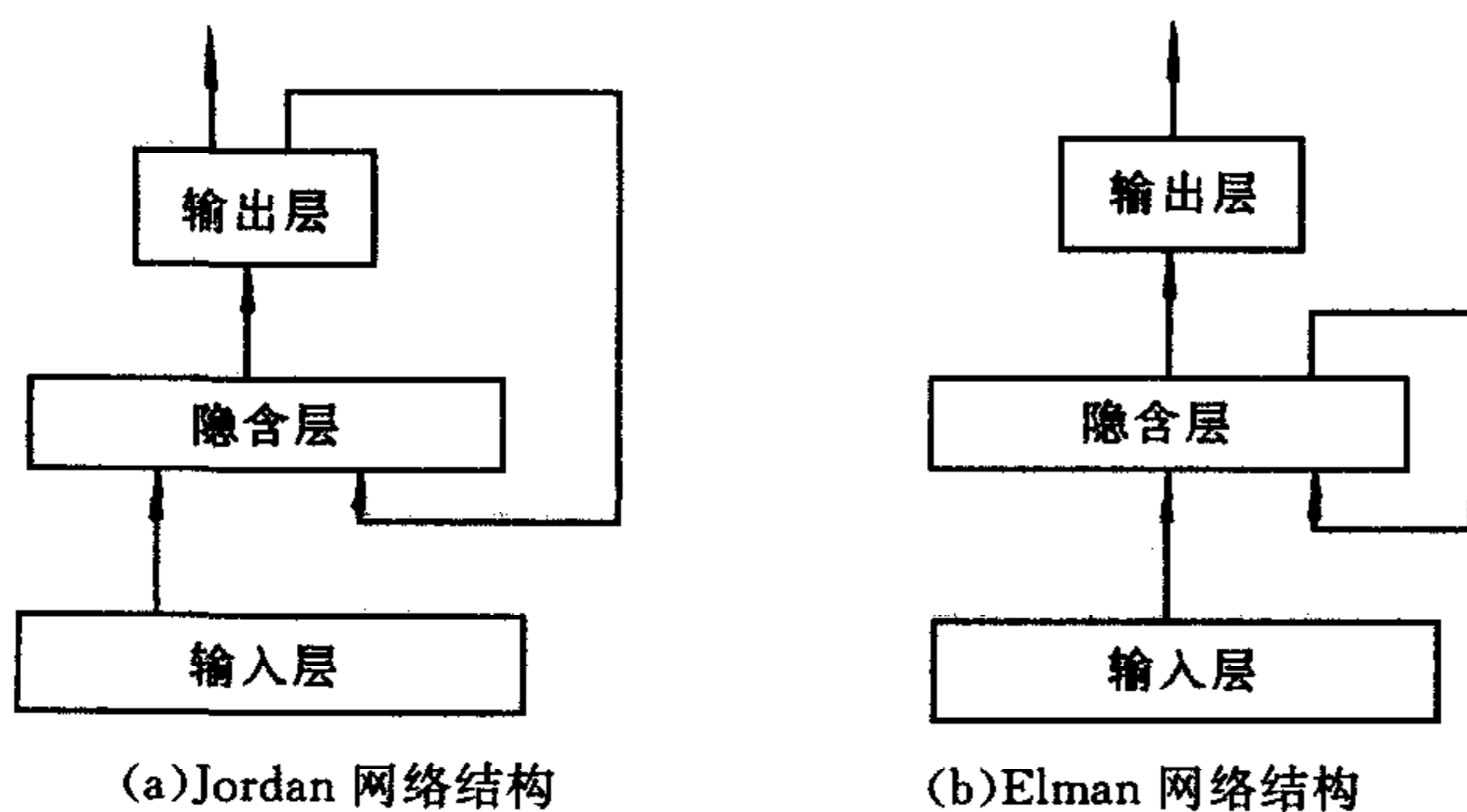


图 1 回归神经网络结构模型

由于在回归神经网络中存在内部反馈,因此传统前馈网络的学习算法——BP 法已不能直接应用于回归神经网络的学习.1987 年 Pineda 首先提出了回归神经网络的 BP 学习算法<sup>[2]</sup>.遗憾的是 Pineda 提出的方法仍然是依据梯度法来进行的,因此不可避免地会产生局部收敛和收敛速度慢等缺陷.如何寻求一种快速的寻优算法来解决回归神经网络的学习问题已引起许多专家学者的关注.本文将非线性系统的最小二乘递推算法引到回归神经网络权阵的学习中来,利用此算法的实时性和快速性来解决神经网络的学习

<sup>1)</sup>国家教委留学回国人员科研启动基金资助.

习问题. 依据回归神经网络 BP 学习算法的原理和方法来推导出回归神经网络的快速在线学习算法. 仿真实验进一步证明了此学习算法的有效性和快速性.

## 2 回归神经网络模型及其 BP 学习算法

对于一个动态的目标序列  $t_k, t_{k-1}, t_{k-2}, \dots$ , 传统的神经网络动态系统描述法是通过引入系统的先前输出  $y_{k-1}, y_{k-2}, \dots, y_{k-n}$  等作为前向传播神经网络模型的输入信号, 从而来描述系统的动力学特性的. 最典型的此类模型结构就是 Narendra K. S. 等人提出的带时滞环节的 MLP 网络结构<sup>[1]</sup>. 显然, 引入过少的时滞环节即  $n$  太小则不能描述具有高阶滞后系统的动态时序特性. 另一方面, 引入过大的时滞环节即  $n$  太大, 则需要系统更多的记忆单元来保持所有的先验信息, 因此也是不可取的. 回归神经网络就是针对这一缺陷而提出的. 它能利用很少的记忆单元来描述任何系统的动力学特性.

设  $Y, H, U, R$  分别表示  $m$  维输出矢量、 $h$  维隐含单元的状态矢量、 $n$  维输入矢量和  $r$  维内部反馈状态矢量, 则回归神经网络模型可以用如下方程来描述

$$Y_k = \Phi(b + \beta' H_k), \quad H_k = \Psi(c + \gamma' U_k + \delta' R_k), \quad R_k = \Lambda(U_k, R_{k-1}, W) \quad (1), (2), (3)$$

其中  $k$  表示第  $k$  时刻,  $\Phi, \Psi, \Lambda$  分别为输出神经元、隐含神经元和反馈神经单元的矢量函数,  $W$  为  $s$  维的神经网络连接权系数阵, 包括  $(b, \beta', c, \gamma', \delta')$ . 当取  $R_k = Y_{k-1}, H_{k-1}, 0$  时, 则此网络分别为 Jordan、Elman、MLP 网络结构.

由于引入了回归单元函数(3), 可以实现对所有先验输入数据的纪录; 换言之, 回归神经网络的回归变量  $R$  能够依据网络输出  $Y$  或隐含单元的状态  $H$  的信息用紧凑的形式来保留系统所有以前的信息. 由式(3)递推可得

$$R_k = \Lambda(U_k, R_{k-1}; W) = \Lambda(U_k, \Lambda(U_{k-1}, R_{k-2}; W); W) = \dots = \Gamma(U^k; W), \quad (4)$$

其中  $U^k = \{U_0, U_1, \dots, U_k\}$  为所有以前的输入数据信息.

从(4)式可以看出, 回归神经网络模型是最节约的网络模型, 它无需存储所有的输入信息但又能在网络中反映出系统所有的历史信号对当前系统响应的影响. 正是由于回归神经网络的这一特性使得其在动态领域的应用具有很大的吸引力.

由(1,2)式可得

$$Y_k = \Phi(b + \beta' \Psi)(c + \gamma' U_k + \delta' R_k(W)) = F(U_k, R_k(W); W). \quad (5)$$

将式(4)代入式(5)可以看出, 网络的输出  $Y_k$  也是系统过去所有输入  $U^{k-1}$  的函数. 要使这样的回归网络真正能在实际中得到应用, 首先需要解决此类网络的学习问题. 与传统的前向传播网络一样, 回归神经网络的学习指标函数仍然选用误差平方和函数, 即给定的样本输出空间  $Z_k, k=0, 1, 2, \dots$

$$J_k = \sum_{l=0}^k J_l = \frac{1}{2} \sum_{l=0}^k (Z_l - Y_l)^2. \quad (6)$$

对于动态序列  $Z_k$  的网络权系数阵  $W$  的最优求解可归结为求

$$W^* = \min \left\{ \frac{1}{2} \lim_{k \rightarrow \infty} E[Z_k - F(U_k, R_k(W); W)]^2 \right\}, \quad (7)$$

其中  $E[\cdot]$  表示数学期望.

假设式(7)的极限存在,则由最大似然估计法可知权系数阵的更新公式为

$$W_{k+1} = W_k - \eta_{k+1} P_{k+1} \nabla e_{k+1}^T \cdot e_{k+1}, \quad (8)$$

其中

$$e_{k+1} = Z_{k+1} - F(U_{k+1}, R_{k+1}; W_k), \quad (9)$$

$\nabla e_{k+1}$  为  $e_{k+1}$  对参数阵  $W$  的梯度矢量.

当  $P_{k+1} = I$  时,即为梯度法;当  $P_{k+1} = \nabla E_{k+1}^T \cdot \nabla E_{k+1}^T$  时,即为牛顿法. 其中  $\nabla E_{k+1} = \frac{\partial J_{k+1}}{\partial W}$ .

必须注意的是,由于方程(9)描述的误差矢量既是  $W$  的显函数又是  $R_{k+1}$  的函数即  $W$  的隐函数. 因此在求梯度矢量  $\nabla e_{k+1}$  时不但要考虑  $e_{k+1}$  对  $W$  的显式求导而且要考虑它的隐式求导. 考虑了这些因素以后,不难得出回归神经网络模型(1-3)的 BP 学习算法(见文献[3]).

### 3 回归网络的在线学习算法

由于考虑的是动态系统,因此在处理样本数据时应尽量加重最新的输入输出数据在网络建模中的作用. 与回归神经网络的传统 BP 学习算法指标函数不同的是,本文提出的改进算法采用遗忘因子对  $J_k$  中的各项拟合残差平方进行加权办法来提高动态模型的实时性. 即取目标函数

$$J_k(W) = \sum_{l=0}^k J_l = \frac{1}{2} \sum_{l=0}^k \alpha^{k-l} (Z_l - Y_l)^2 = \frac{1}{2} \sum_{l=0}^k \alpha^{k-l} e(l, W)^2. \quad (10)$$

如果  $W_k$  是满足目标函数(10)的极小解,则有

$$\left. \frac{\partial J_k(\hat{W})}{\partial W} \right|_{W_k} = \sum_{l=1}^k \alpha^{k-l} \frac{\partial e(k, \hat{W}_k)}{\partial W} e(k, \hat{W}_k) = 0. \quad (11)$$

假设最大似然估计  $\hat{W}_k$  具备良好的渐近性质,则当  $k$  比较大时  $\hat{W}_k$  可望与真值  $W$  很接近. 因此在  $\hat{W} = \hat{W}_k$  点的附近  $e(l, \hat{W})$  可在  $\hat{W}_k$  点进行泰勒展开,并取一次项得

$$e(l, W) \approx e(l, \hat{W}_k) + \frac{\partial e^T(l, \hat{W}_k)}{\partial W} (\hat{W} - \hat{W}_k). \quad (12)$$

若再取得一组新的观察数据  $Z_{k+1}$ ,则可根据  $l=k+1$  时刻以前的全部观察,得出使目标函数

$$J_{k+1}(W) = \sum_{l=0}^{k+1} J_l = \frac{1}{2} \sum_{l=0}^{k+1} \alpha^{k+1-l} e(l, W)^2 = \alpha J_k(W) + \frac{1}{2} e^2(k+1, W) \quad (13)$$

的极小解  $\hat{W}_{k+1}$ . 此时还应有

$$\left. \frac{\partial J_{k+1}(\hat{W})}{\partial W} \right|_{W_{k+1}} = \alpha \left. \frac{\partial J_k(\hat{W})}{\partial W} \right|_{W_{k+1}} + \frac{\partial e(k+1, \hat{W}_{k+1})}{\partial W} e(k+1, \hat{W}_{k+1}) = 0. \quad (14)$$

由于  $k$  很大时,  $\hat{W}_{k+1} \approx \hat{W}_k$ , 故按式(12)有

$$e(l, \hat{W}_{k+1}) \approx e(l, \hat{W}_k) + \frac{\partial e^T(l, \hat{W}_k)}{\partial W} (\hat{W}_{k+1} - \hat{W}_k), \quad (15)$$

$$\frac{\partial e(l, \hat{W}_{k+1})}{\partial W} = \frac{\partial e(l, W)}{\partial W} \Big|_{W=\hat{W}_{k+1}} \approx \frac{\partial e(l, \hat{W}_k)}{\partial W}. \quad (16)$$

利用式(11), (15), (16), 对目标函数(10)式进行求导得

$$\begin{aligned} \frac{\partial J_k(\hat{W})}{\partial W} \Big|_{\hat{W}_{k+1}} &= \sum_{l=1}^k \alpha^{k-l} \frac{\partial e(k, \hat{W}_{k+1})}{\partial W} e(k, \hat{W}_{k+1}) \approx \\ &\sum_{l=1}^k \alpha^{k-l} \frac{\partial e(k, \hat{W}_k)}{\partial W} \frac{\partial e^T(k, \hat{W}_k)}{\partial W} (\hat{W}_{k+1} - \hat{W}_k). \end{aligned} \quad (17)$$

将式(15)—(17)代入式(14)中可得

$$\begin{aligned} \frac{\partial J_{k+1}(\hat{W})}{\partial W} \Big|_{\hat{W}_{k+1}} &\approx \sum_{l=1}^{k+1} \alpha^{k+1-l} \frac{\partial e(k, \hat{W}_k)}{\partial W} \frac{\partial e^T(k, \hat{W}_k)}{\partial W} (\hat{W}_{k+1} - \hat{W}_k) + \\ \frac{\partial e(k+1, \hat{W}_k)}{\partial W} e(k+1, \hat{W}_k) &= \sum_{l=1}^{k+1} \alpha^{k+1-l} \nabla e(k, \hat{W}_k) \cdot \nabla e^T(k, \hat{W}_k) (\hat{W}_{k+1} - \hat{W}_k) + \\ \nabla e(k+1, \hat{W}_k) e(k+1, \hat{W}_k) &= 0. \end{aligned} \quad (18)$$

由式(18)可得

$$\hat{W}_{k+1} = \hat{W}_k - \left[ \sum_{l=1}^{k+1} \alpha^{k+1-l} \nabla e(k, \hat{W}_k) \cdot \nabla e^T(k, \hat{W}_k) \right]^{-1} \nabla e(k+1, \hat{W}_k) e(k+1, \hat{W}_k). \quad (19)$$

$$\text{引入变量 } P_k = \left[ \sum_{l=1}^k \alpha^{k-l} \nabla e(k, \hat{W}_k) \cdot \nabla e^T(k, \hat{W}_k) \right]^{-1}, \quad (20)$$

则回归神经网络的联结系数矩阵的更新公式为

$$\hat{W}_{k+1} = \hat{W}_k - P_{k+1} \nabla e(k+1, \hat{W}_k) e(k+1, \hat{W}_k), \quad (21)$$

$$\begin{aligned} \text{其中 } P_{k+1} &= \left[ \sum_{l=1}^{k+1} \alpha^{k+1-l} \nabla e(l, \hat{W}_k) \cdot \nabla e(l, \hat{W}_k) \right]^{-1} = \\ &\left[ \alpha P_k^{-1} + \nabla e(k+1, \hat{W}_k) \nabla e^T(k+1, \hat{W}_k) \right]^{-1}. \end{aligned} \quad (22)$$

利用矩阵反演公式可将(22)式进一步简化, 从而可得出整个回归神经网络的快速在线学习算法

1) 随机选取一个初始估计系数阵  $\hat{W}_0, R_0, \hat{\Delta}_0, P_0 = \rho I_{s \times s}$ ,  $\rho$  表示在第一步迭代时权系数阵按梯度方向进行更新的步长.

2)  $k \leftarrow 0$ .

3) 取当前最新的样本数据  $(U_{k+1}, Z_{k+1})$ , 按下列各式更新权系数阵

$$R_{k+1} = \Lambda(U_{k+1}, R_k; W_k), \quad (23)$$

$$e(k+1, \hat{W}_k) = Z_{k+1} - F(U_{k+1}, R_{k+1}, \hat{W}_k), \quad (24)$$

$$\hat{\Delta}_{k+1} = \Lambda_W(U_{k+1}, R_k; W_k) + \hat{\Delta}_k \cdot \Lambda_R(U_{k+1}, R_k; W_k), \quad (25)$$

$$\nabla e(k+1, \hat{W}_k) = -F_W(U_{k+1}, R_{k+1}; W_k) - \hat{\Delta}_{k+1} F_R(U_{k+1}, R_{k+1}; W_k), \quad (26)$$

$$\gamma_{k+1} = [\alpha I_{m \times m} + \nabla e^T(k+1, \hat{W}_k) P_k \nabla e(k+1, \hat{W}_k)]^{-1}, \quad (27)$$

$$P_{k+1} = [\alpha P_k - P_k \nabla e(k+1, \hat{W}_k) \gamma_{k+1} \nabla e^T(k+1, \hat{W}_k) P_k] / \alpha, \quad (28)$$

$$\hat{W}_{k+1} = \hat{W}_k - P_{k+1} \nabla e(k+1, \hat{W}_k) e(k+1, \hat{W}_k). \quad (29)$$

4)  $k \leftarrow k+1$ , 转3).

## 4 仿真研究

通过一个例子的仿真结果来评价本文提出的快速在线学习算法的有效性. 仿真模型选用文献[4]中给出的双线性 DGP 模型, 即

$$z_k = 0.5 - 0.4z_{k-1} + 0.4z_{k-1}u_{k-1} + u_k, \quad k = 1, 2, \dots, k_{\max}, \quad (30)$$

其中  $u_k$  选择独立的随机序列  $N(0, 2)$ .

不难看出, 此系统的输出依赖于系统过去的所有信息. 在例中选用 Elman 回归神经网络结构, 并取隐含单元为4. 隐含神经元的激励函数为 Sigmoid 函数、输出单元为线性函数. 在例子的仿真研究中取  $\eta=0.02, \alpha=0.95, \rho=0.68$ .  $W_0, R_0, \hat{\Delta}_0$  为0—1之间的随机数. 输入单元为  $z_{k-1}, u_k$ . 当选用 Elman 回归神经网络结构时其连接系数总共有33个. 利用此神经网络结构和给定的动态模型进行仿真研究, 其仿真结果如图2, 图3所示.

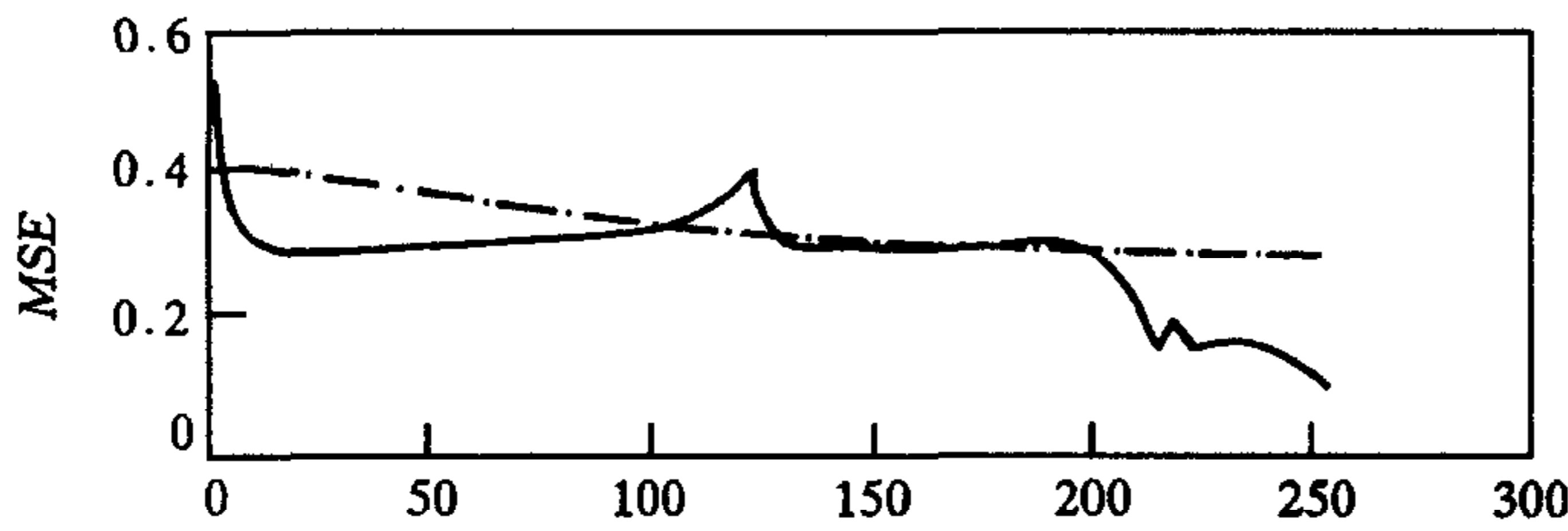


图2 两种学习算法的平方误差曲线

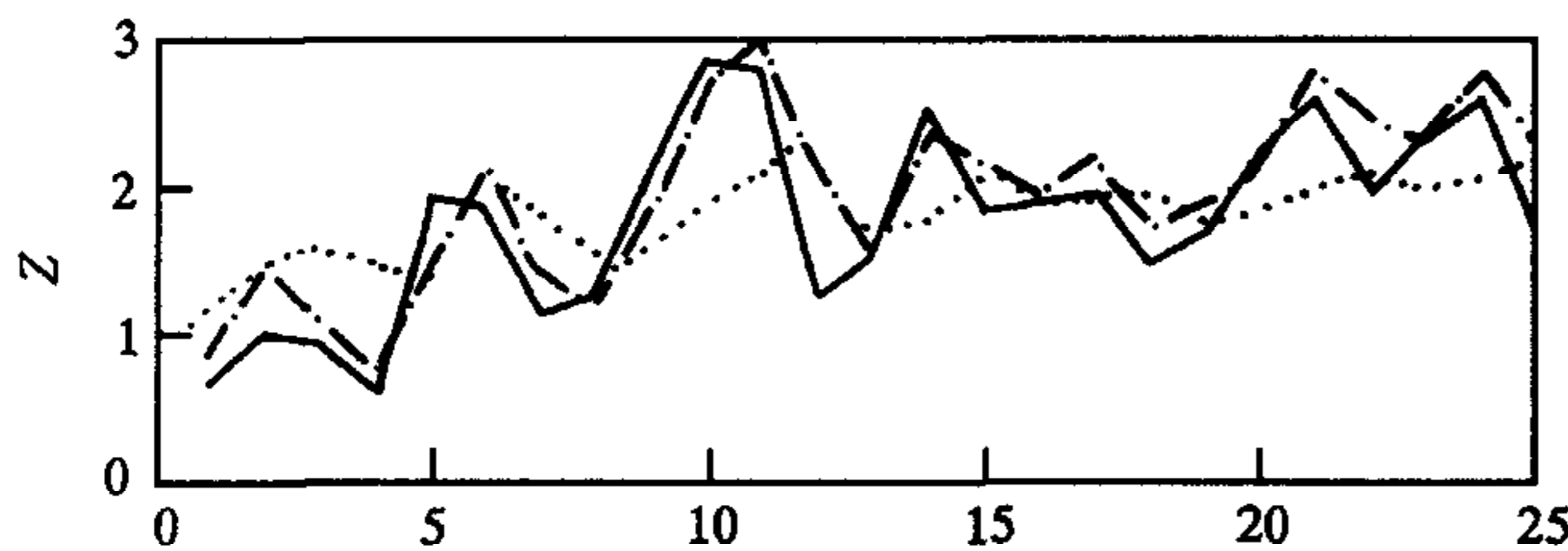


图3 两种学习算法的辨识结果

图2中的横座标表示迭代的次数, 纵座标表示最近25个点的平方误差均值. 其中实线表示利用本算法进行学习的平方误差收敛曲线、虚线表示用传统回归网络 BP 学习算法训练的平方误差收敛曲线; 图3表示最后一次迭代学习结束后的最近25个点的系统响应曲线. 其中横座标表示时间相对值、纵座标为系统的输出即实线表示动态模型的真实输出值、点划线表示回归网络在线学习算法得出的辨识结果、点线表示传统回归网络 BP 学习算法得出的辨识结果. 从仿

真结果可以看出, 回归网络在线学习算法具备比传统回归 BP 算法更快的收敛性和更高的辨识精度. 回归网络在线学习算法在学习过程中由于  $P_0$  的正定性难以时时刻刻得以满足, 因此不可避免地出现一些波动. 在本算法中采取了一些措施来抑制算法可能出现的发散, 如当相邻两次的目标函数上升幅度过大时权系数的更新将沿梯度方向进行. 矩阵  $P_0$  将重置为对角阵. 从而保证了回归神经网络在线学习算法能够正常运行下去. 此外, 在初始学习阶段, 由于  $W_k$  远离真值  $W$ , 因此式(12)的近似度受到严重影响. 在仿真研究中本文采取在初始的几步中选择负梯度方向进行搜索, 而后转入在线递推算法的训练和学习.

## 参 考 文 献

- 1 Narendra K S, Parthasarathy K. Identification and control of dynamical systems using neural networks. *IEEE Trans. on N.N.* 1990, 1(1):4—27
- 2 Pineda F J. Generalization of back-propagation to recurrent neural networks. *Physical Rev. Lett.*, 1987, 59: 2229—2232
- 3 Kuan C M, Hornik K, White H. A convergence results for learning in recurrent neural networks. *Neural Computation*, 1994, 6(2):420—440

## A NEW ON-LINE RECURSIVE LEARNING ALGORITHM FOR RECURRENT NEURAL NETWORK

WEI WEI

(Department of Electrical Engineering, Zhejiang University, Hangzhou 310027)

**Abstract** In this paper a new on-line recursive learning algorithm for recurrent neural network is proposed. It overcomes the disadvantage of the slow convergence of the recurrent BP algorithm. The real-time learning ability and the fast convergence of the recurrent network model of nonlinear dynamical system have been obtained by introducing the forgetting factor in the objective function and the maximum likelihood estimation principle. Simulation results show that the proposed algorithm performs better than the traditional recurrent BP algorithm.

**Key words** Recurrent neural network, on-line learning, fast algorithm.

**韦巍** 1964年出生. 1986年在浙江大学获硕士学位, 1994年获博士学位. 现为浙江大学电机系副教授. 近年来发表学术论文20余篇. 目前主要研究方向为智能控制理论和应用、机器人控制.