

# 具有选择与遗忘机制的极端学习机 在时间序列预测中的应用\*

张 弦<sup>†</sup> 王宏力

(第二炮兵工程学院自动控制工程系, 西安 710025)

(2010年10月29日收到; 2011年2月23日收到修改稿)

针对训练样本贯序输入时的极端学习机 (ELM) 训练问题, 提出一种具有选择与遗忘机制的极端学习机 (SF-ELM), 并研究了其在混沌时间序列预测中的应用. SF-ELM 以逐次增加新训练样本的方式实现在线训练, 通过引入遗忘因子以减弱旧训练样本的影响, 同时以泛化能力为判断依据, 对其输出权值进行选择性递推更新. 混沌时间序列在线预测实例表明, SF-ELM 是一种有效的 ELM 在线训练模式. 相比于在线贯序极端学习机, SF-ELM 具有更快的在线训练速度和更高的在线预测精度, 因此更适于混沌时间序列在线预测.

**关键词:** 混沌时间序列, 时间序列预测, 神经网络, 极端学习机

**PACS:** 05.45.Ac, 05.45.Tp

## 1. 引言

混沌时间序列是由非线性确定性系统产生的一种介于确定性与随机性之间的非线性动力学现象. 它是有序与无序、确定性与随机性的中间态, 在宏观上表现为无序无律的混乱运动, 微观上呈现出无穷嵌套几何自相似性, 具有短期的可预测性与长期的不可预测性. 随着混沌理论研究的不断深入及其在通信、信号处理、自动控制及电子对抗等领域中的广泛应用, 混沌时间序列的建模和预测已成为混沌信号处理领域中非常重要的研究内容<sup>[1-13]</sup>, 而神经网络方法则由于具有预测精度高的特点而被众多学者选择用于混沌时间序列的建模和预测.

极端学习机 (extreme learning machine, 简记为 ELM)<sup>[14]</sup> 是一种新颖的单隐层前馈神经网络. ELM 将传统的神经网络参数训练的迭代调整过程转化为求解线性方程组, 根据 Moore-Penrose 广义逆矩阵理论, 解析求得具有最小范数的最小二乘解作为网络权值, 其网络训练过程可一次完成而无需迭代. 相比于传统神经网络训练算法需经多

次迭代调整才可最终确定网络权值, ELM 的训练速度获得了显著提升. 这一优势使得 ELM 已在模式识别与回归估计等问题中获得了成功应用<sup>[15-19]</sup>. 文献[15]为解决 ELM 的在线训练问题, 提出了一种在线贯序极端学习机 (on-line sequential extreme learning machine, 简记为 OS-ELM). OS-ELM 首先在 ELM 的初始训练阶段计算出其初始网络权值, 之后当有新训练样本加入训练样本集时, 与之对应的网络权值则可在初始网络权值的基础上递推而得. 然而, OS-ELM 认为新、旧训练样本的价值是相等的, 对其采取等权处理, 未能突出新训练样本的作用. 并且, OS-ELM 只要获得了新训练样本, 就对网络权值进行递推更新. 这种机械的网络权值更新模式缺乏根据实际情况调整的灵活性, 同时容易增加不必要的计算量.

为有效解决 OS-ELM 中存在的问题, 本文提出一种具有选择与遗忘机制的极端学习机 (selective forgetting extreme learning machine, 简记为 SF-ELM), 并研究了 SF-ELM 在混沌时间序列在线预测中的应用. SF-ELM 的有效性将通过四种典型混沌时间序列在线预测实例加以验证.

\* 国防科技预研基金 (批准号: 51309060302) 资助的课题.

<sup>†</sup>E-mail: sltecas@sohu.com

## 2. ELM 与 OS-ELM

### 2.1. ELM

基于时间序列训练样本集  $S_k = \{(\mathbf{x}_i, t_i)\}_{i=1}^k$  ( $\mathbf{x}_i = [x_i \ x_{i+1} \ \dots \ x_{i+n-1}]^T, t_i = x_{i+n}, x_i$  为时间序列数据,  $n$  为嵌入维数), 一个包含  $L$  个隐层神经元函数  $f(\cdot)$  的 ELM 回归模型可表示为

$$\begin{aligned} \sum_{i=1}^L \beta_i f(\alpha_i \mathbf{x}_1 + b_i) &= t_1, \\ \sum_{i=1}^L \beta_i f(\alpha_i \mathbf{x}_2 + b_i) &= t_2, \\ &\vdots \\ \sum_{i=1}^L \beta_i f(\alpha_i \mathbf{x}_k + b_i) &= t_k, \end{aligned} \quad (1)$$

式中  $k$  为训练样本的数量;  $\alpha_i$  为连接第  $i$  个神经元的输入权值, 且

$$\alpha_i = [\alpha_{i1} \ \alpha_{i2} \ \dots \ \alpha_{in}];$$

$\beta_i$  为连接第  $i$  个神经元的输出权值;  $b_i$  为第  $i$  个神经元的偏差. 将(1)式写成矩阵形式

$$\mathbf{H}_k \boldsymbol{\beta}_k = \mathbf{T}_k, \quad (2)$$

式中  $\mathbf{H}_k$  为神经元矩阵, 且

$$\mathbf{H}_k = \begin{bmatrix} f(\alpha_1 \mathbf{x}_1 + b_1) & f(\alpha_2 \mathbf{x}_1 + b_2) & \dots & f(\alpha_L \mathbf{x}_1 + b_L) \\ f(\alpha_1 \mathbf{x}_2 + b_1) & f(\alpha_2 \mathbf{x}_2 + b_2) & \dots & f(\alpha_L \mathbf{x}_2 + b_L) \\ \vdots & \vdots & \ddots & \vdots \\ f(\alpha_1 \mathbf{x}_k + b_1) & f(\alpha_2 \mathbf{x}_k + b_2) & \dots & f(\alpha_L \mathbf{x}_k + b_L) \end{bmatrix};$$

$\boldsymbol{\beta}_k$  为输出权值, 且

$$\boldsymbol{\beta}_k = [\beta_1 \ \beta_2 \ \dots \ \beta_L]^T;$$

$\mathbf{T}_k$  为输出向量, 且

$$\mathbf{T}_k = [t_1 \ t_2 \ \dots \ t_k]^T.$$

由于多数情况下  $k$  远大于  $L$ , 因此求解(2)式可得输出权值

$$\boldsymbol{\beta}_k = (\mathbf{H}_k^T \mathbf{H}_k)^{-1} \mathbf{H}_k^T \mathbf{T}_k. \quad (3)$$

因此, 最终可得训练后的 ELM 时间序列预测模型

$$t = \sum_{i=1}^L \beta_i f(\alpha_i \mathbf{x} + b_i), \quad (4)$$

式中  $\mathbf{x}$  为预测模型输入,  $t$  为预测模型输出.

### 2.2. OS-ELM

下面给出 OS-ELM 的计算步骤.

**步骤 I** 在满足  $k \geq L$  的条件下, 利用  $S_k$  计算初始输出权值

$$\boldsymbol{\beta}_k = \mathbf{P}_k \mathbf{H}_k^T \mathbf{T}_k, \quad (5)$$

式中

$$\mathbf{P}_k = (\mathbf{H}_k^T \mathbf{H}_k)^{-1}.$$

**步骤 II** 将新训练样本  $(\mathbf{x}_{k+1}, t_{k+1})$  加入训练样本集, 并根据下式递推计算此时的  $\mathbf{P}_{k+1}$  与  $\boldsymbol{\beta}_{k+1}$ :

$$\mathbf{P}_{k+1} = \mathbf{P}_k - \frac{\mathbf{P}_k \mathbf{h}_{k+1}^T \mathbf{h}_{k+1} \mathbf{P}_k}{1 + \mathbf{h}_{k+1}^T \mathbf{P}_k \mathbf{h}_{k+1}}, \quad (6)$$

$$\boldsymbol{\beta}_{k+1} = \boldsymbol{\beta}_k + \mathbf{P}_{k+1} \mathbf{h}_{k+1}^T (t_{k+1} - \mathbf{h}_{k+1} \boldsymbol{\beta}_k),$$

式中

$$\mathbf{h}_{k+1} = [f(\alpha_1 \mathbf{x}_{k+1} + b_1) \ f(\alpha_2 \mathbf{x}_{k+1} + b_2) \ \dots \ f(\alpha_L \mathbf{x}_{k+1} + b_L)].$$

**步骤 III** 令  $k = k + 1$  后转至步骤 II.

从上述 OS-ELM 的计算步骤可知, 该方法对新、旧训练样本采取等权处理, 且对于  $\boldsymbol{\beta}_k$  采取单一的递推更新模式. 然而, 相比于由旧数据构成的旧训练样本, 由新数据构成的新训练样本更能反映出当前混沌时间序列的动态变化特性, 如果对新、旧训练样本进行加权处理, 以增强新训练样本的作用, 则可进一步提高预测精度. 同时, 利用新训练样本更新  $\boldsymbol{\beta}_k$  的目的是为了使基于  $\boldsymbol{\beta}_k$  的预测模型更加逼近当前时刻的混沌时间序列真实模型, 从而提高预测模型的预测精度. 如果此时的预测模型已具有较高的预测精度, 则暂时无需更新  $\boldsymbol{\beta}_k$ . 因此, 对于  $\boldsymbol{\beta}_k$  的更新应是可选择的而并非每次均要更新. 此外, 在 OS-ELM 的计算过程中,  $\mathbf{P}_k$  和  $\boldsymbol{\beta}_k$  的更新涉及  $\mathbf{P}_k, \mathbf{P}_{k+1}, \mathbf{H}_{k+1}$  和  $\mathbf{H}_{k+1}^T$  之间多次反复的矩阵乘法运算, 计算过程较为繁琐.

## 3. SF-ELM

为解决 OS-ELM 应用于混沌时间序列在线预测时存在的问题, 同时进一步提高 ELM 在线训练的计算效率和基于 ELM 的预测精度, 本文提出一种具有遗忘机制且  $\mathbf{P}_k$  和  $\boldsymbol{\beta}_k$  的更新模式可调的 SF-ELM.

假设当前已利用  $k$  个训练样本  $(\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2), \dots, (\mathbf{x}_k, t_k)$  计算获得了(3)式所示的  $\boldsymbol{\beta}_k$ . 当有新训练样本  $(\mathbf{x}_{k+1}, t_{k+1})$  加入时, 此时对应的  $\boldsymbol{\beta}_{k+1}$  可表示为

$$\begin{aligned} \boldsymbol{\beta}_{k+1} &= \left( \begin{bmatrix} \mathbf{H}_k \\ \mathbf{h}_{k+1} \end{bmatrix}^T \begin{bmatrix} \mathbf{H}_k \\ \mathbf{h}_{k+1} \end{bmatrix} \right)^{-1} \begin{bmatrix} \mathbf{H}_k \\ \mathbf{h}_{k+1} \end{bmatrix}^T \begin{bmatrix} \mathbf{T}_k \\ t_{k+1} \end{bmatrix} \\ &= (\mathbf{H}_k^T \mathbf{H}_k + \mathbf{h}_{k+1}^T \mathbf{h}_{k+1})^{-1} (\mathbf{H}_k^T \mathbf{T}_k + \mathbf{h}_{k+1}^T t_{k+1}). \end{aligned} \quad (7)$$

由于  $\boldsymbol{\beta}_{k+1}$  中的  $\mathbf{H}_k^T \mathbf{H}_k$  和  $\mathbf{H}_k^T \mathbf{T}_k$  均由旧训练样本

构成,因此对上述两项进行加权处理,从而将(7)式重新表示为

$$\beta_{k+1} = (w\mathbf{H}_k^T\mathbf{H}_k + \mathbf{h}_{k+1}^T\mathbf{h}_{k+1})^{-1}(w\mathbf{H}_k^T\mathbf{T}_k + \mathbf{h}_{k+1}^T t_{k+1}), \quad (8)$$

式中  $w$  为遗忘因子且满足  $0 < w < 1$ , 其作用是削弱旧训练样本的影响,从而间接增强新训练样本的作用. 令

$$\mathbf{P}_{k+1} = (w\mathbf{H}_k^T\mathbf{H}_k + \mathbf{h}_{k+1}^T\mathbf{h}_{k+1})^{-1}, \quad (9)$$

对(9)式两端同时求逆可得

$$\mathbf{P}_{k+1}^{-1} = w\mathbf{P}_k^{-1} + \mathbf{h}_{k+1}^T\mathbf{h}_{k+1}. \quad (10)$$

将(10)式代入(8)式,可得  $\beta_k$  的递推表达式为

$$\begin{aligned} \beta_{k+1} &= \mathbf{P}_{k+1}(w\mathbf{H}_k^T\mathbf{T}_k + \mathbf{h}_{k+1}^T t_{k+1}) \\ &= \mathbf{P}_{k+1}(w\mathbf{P}_k^{-1}\beta_k + \mathbf{h}_{k+1}^T t_{k+1}) \\ &= \mathbf{P}_{k+1}((\mathbf{P}_{k+1}^{-1} - \mathbf{h}_{k+1}^T\mathbf{h}_{k+1})\beta_k + \mathbf{h}_{k+1}^T t_{k+1}) \\ &= \beta_k + \mathbf{P}_{k+1}\mathbf{h}_{k+1}^T(t_{k+1} - \mathbf{h}_{k+1}\beta_k). \end{aligned} \quad (11)$$

将 Sherman-Morrison 矩阵求逆引理<sup>[20]</sup>应用于(9)式,可得  $\mathbf{P}_k$  的递推表达式

$$\mathbf{P}_{k+1} = \frac{\mathbf{P}_k}{w} - \frac{\mathbf{Q}_k\mathbf{Q}_k^T}{w(w + \mathbf{h}_{k+1}^T\mathbf{Q}_k)}, \quad (12)$$

式中

$$\mathbf{Q}_k = \mathbf{P}_k\mathbf{h}_{k+1}^T.$$

#### 4. 基于 SF-ELM 的混沌时间序列在线预测

基于 SF-ELM 的混沌时间序列在线预测过程分为五步进行.

**步骤 I** 选择嵌入维数为  $n$ , 将初始的  $N$  个时间序列数据  $x_1, x_2, \dots, x_N$  转化为训练样本  $(\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2), \dots, (\mathbf{x}_k, t_k), \mathbf{x}_i = [x_i \ x_{i+1} \ \dots \ x_{i+n-1}]^T$  为预测模型输入,  $t_i = x_{i+n}$  为预测模型输出,  $k = N - n \geq L$ .

**步骤 II** 利用  $(\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2), \dots, (\mathbf{x}_k, t_k)$  计算初始输出权值

$$\beta_k = \mathbf{P}_k\mathbf{H}_k^T\mathbf{T}_k. \quad (13)$$

这里

$$\mathbf{P}_k = \left(\mathbf{H}_k^T\mathbf{H}_k + \frac{\mathbf{I}_k}{C}\right)^{-1},$$

其中  $\mathbf{I}_k$  为单位矩阵,  $C$  为岭回归系数.

**步骤 III** 以  $\mathbf{x}_{k+1} = [x_{N-n+1} \ x_{N-n+2} \ \dots \ x_N]^T$  为输入, 计算神经元矩阵输入向量  $\mathbf{h}_{k+1}$ , 从而获得对  $x_{N+1}$  的一步预测值

$$\bar{x}_{N+1} = \mathbf{h}_{k+1}\beta_k. \quad (14)$$

**步骤 IV** 在获得  $x_{N+1}$  的真实值之后, 首先根据下式对  $\mathbf{P}_k$  进行选择更新:

$$\mathbf{P}_{k+1} = \begin{cases} \frac{\mathbf{P}_k}{w} - \frac{\mathbf{Q}_k\mathbf{Q}_k^T}{w(w + \mathbf{h}_{k+1}^T\mathbf{Q}_k)} & (E_N > \varepsilon), \\ \mathbf{P}_k & (E_N \leq \varepsilon), \end{cases} \quad (15)$$

式中  $E_N = \sqrt{(\bar{x}_{N+1} - x_{N+1})^2}$  为均方根误差,  $\varepsilon$  为事先设定的阈值. 然后利用更新后的  $\mathbf{P}_k$  根据下式更新  $\beta_k$ :

$$\beta_{k+1} = \beta_k + \mathbf{P}_{k+1}\mathbf{h}_{k+1}^T(t_{k+1} - \mathbf{h}_{k+1}\beta_k), \quad (16)$$

式中  $t_{k+1} = x_{N+1}$ .

**步骤 V** 令  $k = k + 1, N = N + 1$  后转至步骤 III.

#### 5. 混沌时间序列在线预测实例

以 Mackey-Glass 混沌时间序列(简记为 CTS-1)、Tent 混沌时间序列(简记为 CTS-2)、Logistic 混沌时间序列(简记为 CTS-3)和 Hénon 混沌时间序列(简记为 CTS-4)为例进行在线预测. CTS-1 由下式描述:

$$\frac{dx_M(t)}{dt} = \frac{0.2x_M(t-20)}{1 + [x_M(t-20)]^{10}} - 0.1x_M(t). \quad (17)$$

CTS-2 由下式描述:

$$x_T(n+1) = \begin{cases} x_T(n)/0.4 & (x_T(n) \leq 0.4), \\ (1 - x_T(n))/0.6 & (x_T(n) > 0.4). \end{cases} \quad (18)$$

CTS-3 由下式描述:

$$x_L(n+1) = 4x_L(n)(1 - x_L(n)). \quad (19)$$

CTS-4 由下式描述:

$$x_H(n+1) = 1 - 0.3x_H(n) - 1.4x_H(n)^2. \quad (20)$$

将 SF-ELM, OS-ELM 用于在线预测, 同时将其预测结果与基于离线模式训练且不更新输出权值的 ELM 进行比较. 由于隐层神经元的数量  $L$  是影响神经网络学习精度和泛化能力的重要因素, 因此分别选择  $L = 20, L = 50$  和  $L = 100$  时的 SF-ELM, OS-ELM 和 ELM 用于预测. 同时, 令  $L = 20$  时, SF-ELM, OS-ELM 和 ELM 初始训练阶段的训练样本数量  $k = 50; L = 50$  时, 相应的  $k = 100; L = 100$  时, 相应的  $k = 200$ . 选择嵌入维数  $n = 4$ , SF-ELM 中的岭

回归系数  $C = 10^4$ , 阈值  $\varepsilon = 10^{-3}$ , 遗忘因子  $w = 0.98$ . 表 1、表 2 和表 3 分别列出了  $L = 20, L = 50$  和  $L = 100$  时, SF-ELM, OS-ELM 和 ELM 进行 100, 500, 1000 和 2000 步在线预测时的预测均方根误差  $E_{RMS}$ .

表 1  $L = 20$  时 SF-ELM, OS-ELM 和 ELM 的预测误差

混沌时间序列	预测步长	SF-ELM	OS-ELM	ELM
CTS-1	100	0.0132	0.0173	0.2625
	500	0.0070	0.0110	0.5615
	1000	0.0054	0.0093	0.7004
	2000	0.0043	0.0081	0.8882
CTS-2	100	0.1665	0.1670	0.2327
	500	0.0921	0.1404	0.5055
	1000	0.0290	0.1282	0.6399
	2000	0.0239	0.1391	0.7106
CTS-3	100	0.0452	0.0474	0.0893
	500	0.0288	0.0438	0.2521
	1000	0.0200	0.0340	0.2883
	2000	0.0141	0.0270	0.2982
CTS-4	100	0.0098	0.0213	0.0881
	500	0.0045	0.0320	0.2321
	1000	0.0036	0.0273	0.2457
	2000	0.0025	0.0459	0.3800

表 2  $L = 50$  时 SF-ELM, OS-ELM 和 ELM 的预测误差

混沌时间序列	预测步长	SF-ELM	OS-ELM	ELM
CTS-1	100	0.0069	0.0081	0.0114
	500	0.0031	0.0045	0.0155
	1000	0.0026	0.0045	0.0232
	2000	0.0022	0.0037	0.0328
CTS-2	100	0.0096	0.0107	0.0558
	500	0.0120	0.0161	0.0725
	1000	0.0395	0.0607	0.0820
	2000	0.0214	0.1188	0.1860
CTS-3	100	0.0096	0.0106	0.0195
	500	0.0052	0.0114	0.0523
	1000	0.0038	0.0133	0.1215
	2000	0.0025	0.0082	0.1932
CTS-4	100	0.0016	0.0016	0.0022
	500	0.0018	0.0019	0.0048
	1000	0.0015	0.0018	0.0068
	2000	0.0010	0.0016	0.0096

表 3  $L = 100$  时 SF-ELM, OS-ELM 和 ELM 的预测误差

混沌时间序列	预测步长	SF-ELM	OS-ELM	ELM
CTS-1	100	0.0007	0.0007	0.0016
	500	0.0010	0.0019	0.0033
	1000	0.0007	0.0019	0.0055
	2000	0.0008	0.0015	0.0077
CTS-2	100	0.0119	0.0138	0.0827
	500	0.0453	0.0518	0.1059
	1000	0.0461	0.0883	0.1106
	2000	0.0393	0.1027	0.2566
CTS-3	100	0.0053	0.0057	0.0069
	500	0.0027	0.0051	0.0117
	1000	0.0021	0.0074	0.0242
	2000	0.0011	0.0045	0.0349
CTS-4	100	0.00014	0.00014	0.00017
	500	0.00015	0.00025	0.00038
	1000	0.00012	0.00021	0.00054
	2000	0.00008	0.00020	0.00077

$$E_{RMS} = \sqrt{\sum_{i=1}^{n_p} (\bar{x}_i - x_i)^2 / n_p},$$

其中  $x_i$  为混沌时间序列的真实值,  $\bar{x}_i$  为  $x_i$  的预测值,  $n_p$  为预测数据的长度.

从表 1—表 3 中的数据可知, 由于 ELM 在初始训练阶段结束后, 不利用新训练样本更新预测模型, 因此随预测步数的增多, 用于训练 ELM 的训练样本逐渐远离当前时刻, 从而导致其跟踪混沌时间序列动态变化特性的能力逐渐减弱, 其预测精度也会随之逐渐降低. SF-ELM, OS-ELM 由于不断利用包含了当前时刻信息的新训练样本更新预测模型, 因此其预测均方根误差均小于 ELM 的预测均方根误差. 此外, 由于新训练样本距离当前时刻最近, 比旧训练样本更接近于当前时刻真实的混沌时间序列, 因此更具价值. 相比于 OS-ELM 对待新旧训练样本的公平原则, SF-ELM 更加重视新训练样本的贡献, 以遗忘旧训练样本的方式间接突出新训练样本的作用, 从而使更新后的预测模型中来自新训练样本的信息多于旧训练样本, 因此 SF-ELM 比 OS-ELM 更加接近当前时刻真实的混沌时间序列, 其预测均方根误差也明显小于 OS-ELM 的预测均方根误差.

图 1、图 2 和图 3 分别展示了  $L = 20, L = 50$  和  $L = 100$  时, SF-ELM, OS-ELM 和 ELM 进行 1 步至

2000步预测时的计算时间.由此可见,SF-ELM的计算时间远少于OS-ELM的计算时间,且预测步长越长,隐层神经元的数量越多,SF-ELM相对于OS-ELM的计算优势就越明显.特别是当 $L = 50$ 和 $L = 100$ 时,利用SF-ELM在线预测CTS-3,CTS-4时的计

算时间仅略多于不更新 $\beta_k$ 的离线ELM的计算时间.这主要是由于SF-ELM采用(15)和(16)式所示的新模式更新 $P_k$ 和 $\beta_k$ ,且对于 $P_k$ 采取选择性更新策略,而并非像OS-ELM那样每增加一个新训练样本就更新一次 $P_k$ ,因此有效减小了计算量.

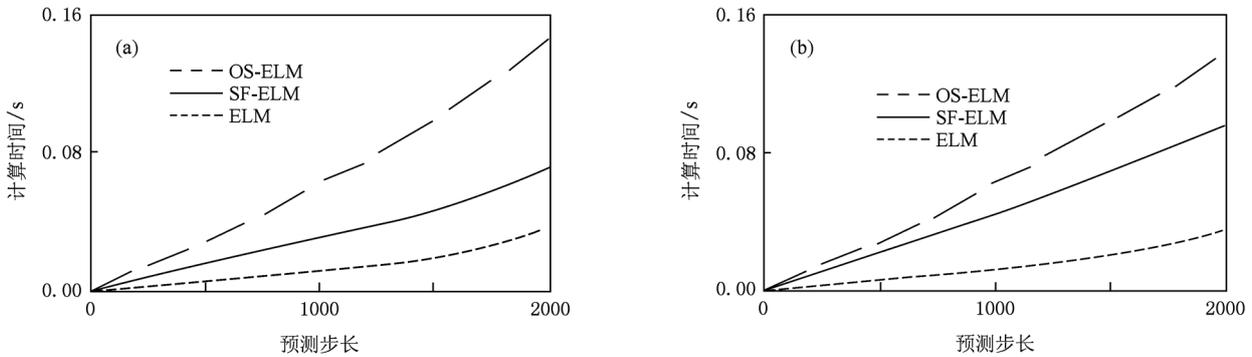


图1  $L = 20$ 时SF-ELM,OS-ELM和ELM对CTS-1,CTS-2的计算时间 (a)CTS-1,(b)CTS-2

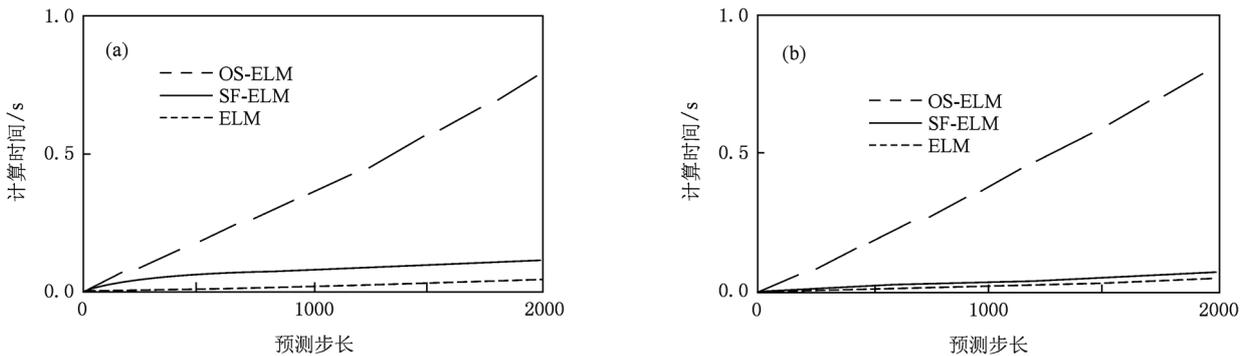


图2  $L = 50$ 时SF-ELM,OS-ELM和ELM对CTS-3,CTS-4的计算时间 (a)CTS-3,(b)CTS-4

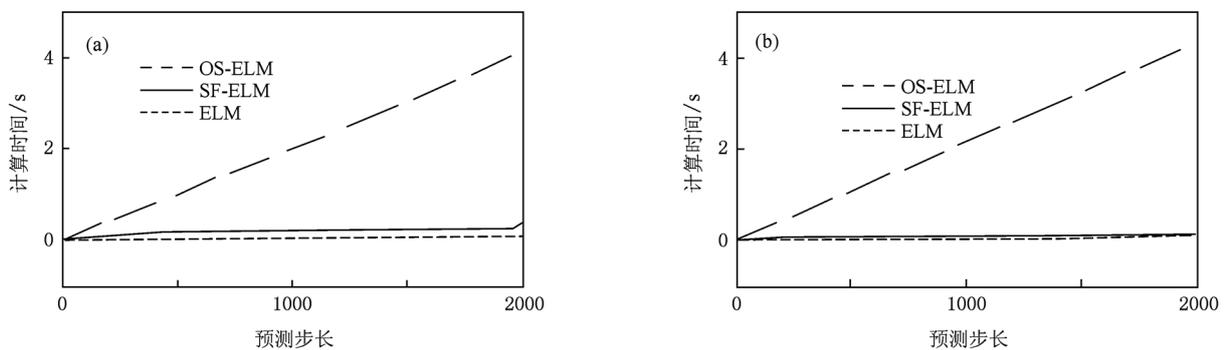


图3  $L = 100$ 时SF-ELM,OS-ELM和ELM对CTS-3,CTS-4的计算时间 (a)CTS-3,(b)CTS-4

图4、图5和图6分别展示了 $L = 20, L = 50$ 和 $L = 100$ 时,SF-ELM和OS-ELM进行2000步在线预测时对 $P_k$ 的更新次数.由此可见,SF-ELM对 $P_k$ 的

更新次数均少于OS-ELM的更新次数,说明SF-ELM中的选择机制发挥了作用,有效减少了对 $P_k$ 的不必要更新次数.

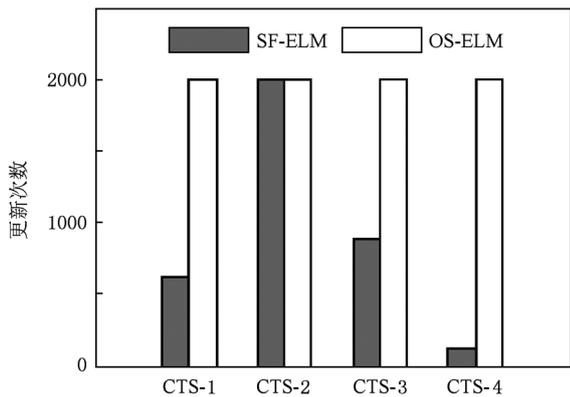


图4 L = 20 时 SF-ELM 和 OS-ELM 的更新次数比较

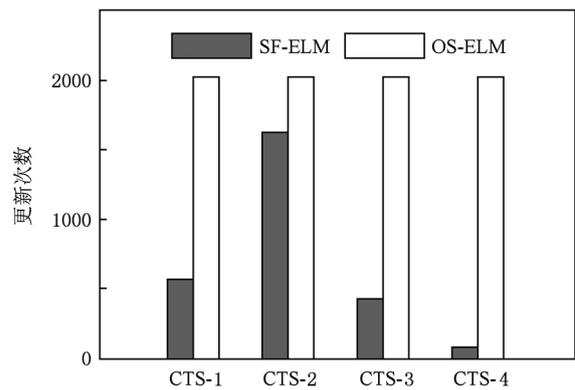


图6 L = 100 时 SF-ELM 和 OS-ELM 的更新次数比较

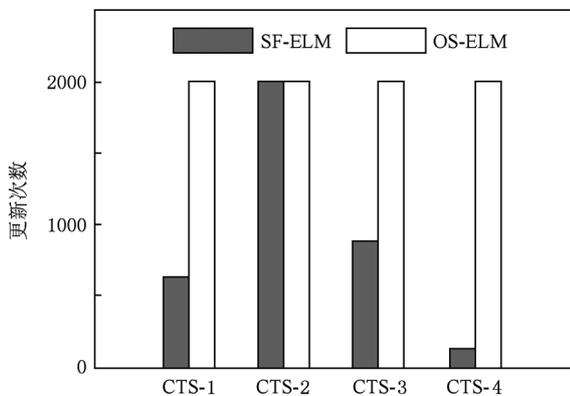


图5 L = 50 时 SF-ELM 和 OS-ELM 的更新次数比较

## 6. 结 论

传统的 ELM 训练算法仅适用于离线训练,一旦有独立于训练样本集之外的新训练样本贯序加入时,需重复网络训练过程. SF-ELM 采用递推模式求解 ELM 的输出权值,使 ELM 具备了在线训练能力,以引入遗忘因子的方式减弱旧训练样本的影响,从而突出新训练样本的作用,并根据其泛化能力对输出权值进行选择性递推更新. 四种混沌时间序列在线预测实例表明,相比 OS-ELM, SF-ELM 中独特的选择与遗忘机制使其更符合混沌时间序列在线预测对于计算快速性和预测准确性的要求.

[1] Song Q S, Feng Z R 2010 *Expert Syst. Appl.* **37** 1776

[2] Fu Y Y, Wu C J, Jeng J T, Ko C N 2010 *Expert Syst. Appl.* **37** 4441

[3] Jeng J T, Chuang C C, Tao C W 2010 *Neurocomputing* **73** 1686

[4] Muhammad A F, Zolfaghari S 2010 *Neurocomputing* **73** 2540

[5] Song Q S, Feng Z R 2010 *Neurocomputing* **73** 2177

[6] Han M, Wang Y 2009 *Expert Syst. Appl.* **36** 1280

[7] Mirzaee H 2009 *Chaos Solitons Fract.* **41** 2681

[8] Lau K W, Wu Q H 2008 *Pattern Recogn.* **41** 1539

[9] Lin C J, Chen C H, Lin C T 2008 *IEEE Trans. Syst. Man Cybernet.* **39** 55

[10] Zhang C T, Ma Q L, Peng H 2010 *Acta Phys. Sin.* **59** 7623 (in Chinese) [张春涛、马千里、彭宏 2010 物理学报 **59** 7623]

[11] Liu J H, Zhang H G, Feng J 2010 *Acta Phys. Sin.* **59** 4472 (in Chinese) [刘金海、张化光、冯健 2010 物理学报 **59** 4472]

[12] Song Q S, Feng Z R, Li R H 2009 *Acta Phys. Sin.* **58** 5057 (in Chinese) [宋青松、冯祖仁、李人厚 2009 物理学报 **58** 5057]

[13] Mao J Q, Yao J, Ding H S 2009 *Acta Phys. Sin.* **58** 2220 (in Chinese) [毛剑琴、姚健、丁海山 2009 物理学报 **58** 2220]

[14] Huang G B, Zhu Q Y, Siew C K 2007 *Neurocomputing* **70** 489

[15] Liang N Y, Huang G B, Saratchandran P, Sundararajan N 2006 *IEEE Trans. Neur. Net.* **17** 1411

[16] Feng G, Huang G B, Lin Q P, Gay R 2009 *IEEE Trans. Neur. Net.* **20** 1352

[17] Miche Y, Soriamaa A, Bas P, Simula O, Jutten C, Lendasse A 2010 *IEEE Trans. Neur. Net.* **21** 158

[18] Lan Y, Soh C Y, Huang G B 2010 *Neurocomputing* **73** 3191

[19] Malathi V, Marimuthu N S, Baskar S 2010 *Neurocomputing* **73** 2160

[20] Zhang X D 2005 *Matrix Analysis and Applications* (Beijing: Tsinghua University Press) p64 (in Chinese) [张贤达 2005 矩阵分析与应用 (北京:清华大学出版社) 第 64 页]

# Selective forgetting extreme learning machine and its application to time series prediction \*

Zhang Xian<sup>†</sup> Wang Hong-Li

(Department of Automatic Control Engineering, The Second Artillery Engineering College, Xi'an 710025, China)

(Received 29 October 2010; revised manuscript received 23 February 2011)

## Abstract

To solve the problem of extreme learning machine (ELM) on-line training with sequential training samples, a new algorithm called selective forgetting extreme learning machine (SF-ELM) is proposed and applied to chaotic time series prediction. The SF-ELM adopts the latest training sample and weights the old training samples iteratively to insure that the influence of the old training samples is weakened. The output weight of the SF-ELM is determined recursively during on-line training procedure according to its generalization performance. Numerical experiments on chaotic time series on-line prediction indicate that the SF-ELM is an effective on-line training version of ELM. In comparison with on-line sequential extreme learning machine, the SF-ELM has better performance in the sense of computational cost and prediction accuracy.

**Keywords:** chaotic time series, time series prediction, neural networks, extreme learning machine

**PACS:** 05.45.Ac, 05.45.Tp

---

\* Project supported by the Advanced Research Foundation for National Defense Science and Technology of China (Grant No. 51309060302).

<sup>†</sup> E-mail: sltecas@sohu.com