

基于 Cholesky 分解的增量式 RELM 及其在时间序列预测中的应用

张 弦[†] 王宏力

(第二炮兵工程学院自动控制工程系, 西安 710025)

(2011 年 1 月 11 日收到; 2011 年 2 月 23 日收到修改稿)

针对应用于混沌时间序列预测的正则极端学习机 (RELM) 网络结构设计问题, 提出一种基于 Cholesky 分解的增量式 RELM 训练算法. 该算法通过逐次增加隐层神经元的方式自动确定最佳的 RELM 网络结构, 并以 Cholesky 分解方式计算其输出权值, 有效减小了隐层神经元递增过程的计算代价. 混沌时间序列预测实例表明, 该算法可有效实现最佳 RELM 网络结构的自动确定, 且计算效率高. 利用该算法训练后的 RELM 预测模型具有预测精度高的优点, 适用于混沌时间序列预测.

关键词: 神经网络, 极端学习机, 混沌时间序列, 时间序列预测

PACS: 02. 10. Yn, 05. 45. Ac, 05. 45. Tp

1. 引言

混沌是一种出现在确定性系统中的貌似无序、类似随机的复杂现象. 随着混沌理论研究的不断深入及其在通信、信号处理、自动控制与电子对抗等领域中的广泛应用, 混沌时间序列的建模与预测已成为混沌理论研究领域中一个非常重要且极为活跃的研究方向.

神经网络是一种由大量神经元互连而成的复杂非线性系统. 它是在现代神经生物学与认知科学对人脑信息处理能力研究的基础上提出的, 是对人脑生理结构的数学模拟形式, 具有较强的自适应性、自学习能力与非线性映射能力, 已被许多学者应用于混沌时间序列预测^[1-10]. 然而, 上述预测模型均基于传统神经网络, 其网络训练过程需根据训练目标与梯度信息对网络权值进行反复修改, 整个网络训练过程通常需几百甚至几千次迭代才可最终完成, 计算量较大.

极端学习机 (extreme learning machine, ELM)^[11] 是一种新颖的单隐层前馈神经网络. 它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, 整个训练过程一次完成且无须迭代. 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^[12-17]. 为改善 ELM 的泛化能力, 文献[17]借鉴统计学习理论中的结构风险最小化原理, 提出了正则极端学习机 (regularized extreme learning machine, RELM). RELM 通过引入参数 γ 以权衡结构风险与经验风险, 从而使 ELM 具有更好的泛化能力.

为实现对应用于混沌时间序列预测时的 RELM 网络结构的有效设计, 本文提出一种基于 Cholesky 分解的增量式 RELM 训练算法, 其有效性将通过六种典型混沌时间序列预测实例加以验证.

2. 正则极端学习机

对于时间序列训练样本集 $\{(\mathbf{x}_j, t_j)\}_{j=1}^k$, $\mathbf{x}_j = [x_j \cdots x_{j+n-1}]^T$, $t_j = x_{j+n}$, x_j 为时间序列数据, k 为训练样本的数量, n 为嵌入维数. 为较好权衡经验风

[†] E-mail: sltecas@sohu.com

险与结构风险, RELM 通过引入参数 γ 调节两种风险的比例, 用于回归的 RELM 可表示为约束优化

$$\min\left(\frac{1}{2}\boldsymbol{\beta}_L^T\boldsymbol{\beta}_L + \frac{\gamma}{2}\boldsymbol{\varepsilon}^T\boldsymbol{\varepsilon}\right)$$

$$\text{s. t. } t_j = \sum_{i=1}^L \beta_i f(\boldsymbol{\alpha}_i \mathbf{x}_j + c_i) - \varepsilon_j, \quad (1)$$

式中, $j = 1, \dots, k; f(\cdot)$ 为隐层神经元函数; L 为隐层神经元的数量; $\boldsymbol{\alpha}_i = [\alpha_{i1} \ \dots \ \alpha_{in}]$ 为连接第 i 个隐层神经元的输入权值; $\boldsymbol{\beta}_L = [\beta_1 \ \dots \ \beta_L]^T, \beta_i$ 为连接第 i 个隐层神经元的输出权值; c_i 是第 i 个隐层神经元的偏差; $\boldsymbol{\varepsilon} = [\varepsilon_1 \ \dots \ \varepsilon_k]^T, \varepsilon_i$ 为回归误差. 为求解上述优化问题, 建立拉格朗日函数为

$$L(\mathbf{w}, \boldsymbol{\varepsilon}, \boldsymbol{\beta}_L) = \frac{1}{2}\boldsymbol{\beta}_L^T\boldsymbol{\beta}_L + \frac{\gamma}{2}\boldsymbol{\varepsilon}^T\boldsymbol{\varepsilon}$$

$$- \mathbf{w}(\mathbf{H}_L\boldsymbol{\beta}_L - \mathbf{T} - \boldsymbol{\varepsilon}), \quad (2)$$

式中, $\mathbf{w} = [w_1 \ \dots \ w_k]$ 为拉格朗日权值; $\mathbf{H}_L = \begin{bmatrix} f(\boldsymbol{\alpha}_1 \mathbf{x}_1 + c_1) & \dots & f(\boldsymbol{\alpha}_L \mathbf{x}_1 + c_L) \\ \vdots & & \vdots \\ f(\boldsymbol{\alpha}_1 \mathbf{x}_k + c_1) & \dots & f(\boldsymbol{\alpha}_L \mathbf{x}_k + c_L) \end{bmatrix}$ 为神经元矩阵;

$\mathbf{T} = [t_1 \ \dots \ t_k]^T$ 为输出向量. 对拉格朗日函数各变量求偏导并令偏导数为零可得

$$\frac{\partial L}{\partial \boldsymbol{\beta}_L} \rightarrow \boldsymbol{\beta}_L^T = \mathbf{w}\mathbf{H}_L, \quad (3a)$$

$$\frac{\partial L}{\partial \boldsymbol{\varepsilon}} \rightarrow \gamma\boldsymbol{\varepsilon}^T + \mathbf{w} = 0, \quad (3b)$$

$$\frac{\partial L}{\partial \mathbf{w}} \rightarrow \mathbf{H}_L\boldsymbol{\beta}_L - \mathbf{T} - \boldsymbol{\varepsilon} = 0. \quad (3c)$$

求解(3)式可得^[17]

$$\boldsymbol{\beta}_L = (\mathbf{H}_L^T\mathbf{H}_L + \frac{\mathbf{I}_L}{\gamma})^{-1}\mathbf{H}_L^T\mathbf{T}, \quad (4)$$

式中, \mathbf{I}_L 为单位矩阵. 利用 $\boldsymbol{\beta}_L$ 则可建立 RELM 时间序列预测模型

$$t = \sum_{i=1}^L \beta_i f(\boldsymbol{\alpha}_i x + c_i), \quad (5)$$

式中, x 为模型输入; t 为模型输出.

3. 基于 Cholesky 分解的输出权值求解

从(1)式至(4)式所示的 RELM 训练过程可知, 其实质为求解 $\boldsymbol{\beta}_L$. 然而从(4)式可见, 文献[17]给出的 $\boldsymbol{\beta}_L$ 的求解方式涉及矩阵求逆运算, 存在计算量大的问题, 从而降低了 RELM 时间序列预测模型的建模效率. 为有效解决上述问题, 本文提出一种基于 Cholesky 分解的 $\boldsymbol{\beta}_L$ 求解方法. 首先, 在(3)式的

基础上, 由(3a)式可得 $\boldsymbol{\beta}_L = \mathbf{w}^T\mathbf{H}_L^T$, 由(3b)式可得 $\boldsymbol{\varepsilon} = -\mathbf{w}^T/\gamma$, 将 $\boldsymbol{\beta}_L$ 与 $\boldsymbol{\varepsilon}$ 代入(3c)式可得

$$\mathbf{A}_L\boldsymbol{\beta}_L = \mathbf{b}_L. \quad (6)$$

式中, $\mathbf{A}_L = \mathbf{H}_L^T\mathbf{H}_L + \frac{\mathbf{I}_L}{\gamma}; \mathbf{b}_L = \mathbf{H}_L^T\mathbf{T}$. 因此, 求解 $\boldsymbol{\beta}_L$ 的过程可被转化为求解形如(6)式的线性方程组. 将 Cholesky 分解应用于求解线性方程组的前提是其系数矩阵必须为对称正定矩阵^[18], 因此需证明 \mathbf{A}_L 是对称正定矩阵.

首先, 证明 \mathbf{A}_L 的对称性. 因为

$$\mathbf{A}_L^T = \left(\mathbf{H}_L^T\mathbf{H}_L + \frac{\mathbf{I}_L}{\gamma}\right)^T = \mathbf{H}_L^T\mathbf{H}_L + \frac{\mathbf{I}_L}{\gamma} = \mathbf{A}_L. \quad (7)$$

所以, \mathbf{A}_L 为对称矩阵. 其次, 证明 \mathbf{A}_L 的正定性. 对于任意向量 $\mathbf{v} = [v_1 \ \dots \ v_L]^T, \mathbf{A}_L$ 的二次型可表示为

$$\mathbf{v}^T\mathbf{A}_L\mathbf{v} = \mathbf{v}^T\left(\mathbf{H}_L^T\mathbf{H}_L + \frac{\mathbf{I}_L}{\gamma}\right)\mathbf{v}$$

$$= \sum_{i=1}^L (f(\boldsymbol{\alpha}_i \mathbf{x}_1 + c_i)v_i)^2 + \dots$$

$$+ \sum_{i=1}^L (f(\boldsymbol{\alpha}_i \mathbf{x}_k + c_i)v_i)^2 + \frac{1}{\gamma} \sum_{i=1}^L v_i^2. \quad (8)$$

从(8)式可见, 当 $\mathbf{v} \neq 0$ 时, 必然满足 $\mathbf{v}^T\mathbf{A}_L\mathbf{v} > 0$, 所以, \mathbf{A}_L 为正定矩阵. 因此, \mathbf{A}_L 为对称正定矩阵.

基于 Cholesky 分解的 $\boldsymbol{\beta}_L$ 求解过程为: 首先, 计算 \mathbf{A}_L 的 Cholesky 分解结果

$$\mathbf{A}_L = \mathbf{S}_L\mathbf{S}_L^T. \quad (9)$$

式中, \mathbf{S}_L 是一个具有正的对角线元素的下三角矩阵. \mathbf{S}_L 中不为零的元素 s_{ij} 可利用 \mathbf{A}_L 的元素 a_{ij} 根据(10)式计算而得

$$s_{ij} = \begin{cases} \sqrt{a_{ii} - \sum_{n=1}^{i-1} s_{in}^2} & i = j, \\ (a_{ij} - \sum_{n=1}^{j-1} s_{in}s_{jn})/s_{jj} & i > j. \end{cases} \quad (10)$$

式中, $i = 1, \dots, L; j = 1, \dots, L$. 将(9)式代入(6)式并在等式两端同时乘以 \mathbf{S}_L^{-1} 可得

$$\mathbf{S}_L^T\boldsymbol{\beta}_L = \mathbf{F}_L. \quad (11)$$

式中, $\mathbf{F}_L = \mathbf{S}_L^{-1}\mathbf{b}_L$. 因此求解 $\boldsymbol{\beta}_L$ 等价于求解(11)式. 由于 $\mathbf{F}_L = \mathbf{S}_L^{-1}\mathbf{b}_L$ 等价于 $\mathbf{S}_L\mathbf{F}_L = \mathbf{b}_L$, 因此通过比较等式两端的元素, 可得 \mathbf{F}_L 的元素 f_i 的计算公式为

$$f_i = \begin{cases} b_i/s_{ii} & i = 1, \\ (b_i - \sum_{n=1}^{i-1} s_{ni}f_n)/s_{ii} & i > 1. \end{cases} \quad (12)$$

式中, $i = 1, \dots, L; b_i$ 为 \mathbf{b}_L 相应位置上的元素. 最终

在获得了 S_L 与 F_L 的基础上, β_L 可利用 S_L 与 F_L 的元素计算而得

$$\beta_i = \begin{cases} f_i/s_{ii} & i = L, \\ (f_i - \sum_{n=1}^{L-i} s_{i+n,i}\beta_{i+n})/s_{ii} & i < L. \end{cases} \quad (13)$$

相比于(4)式所示的 β_L 求解方法, 基于 Cholesky 分解的 β_L 求解不涉及矩阵求逆运算, 仅利用简单的四则运算就可实现. 更为重要的是, 当 RELM 的隐层神经元数量从 L 个增加为 $L + 1$ 个时, 神经元矩阵由 H_L 变化为

$$H_{L+1} = [H_L \mid h_{L+1}] = [h_1 \cdots h_L \mid h_{L+1}]. \quad (14)$$

式中, $h_i = [f(\alpha_i x_1 + c_i) \cdots f(\alpha_i x_k + c_i)]^T, i = 1, \dots, L + 1$. 此时

$$A_{L+1} = H_{L+1}^T H_{L+1} + \frac{I_{L+1}}{\gamma} = \begin{bmatrix} A_L & Q_{L+1} \\ Q_{L+1}^T & P_{L+1} \end{bmatrix}. \quad (15)$$

式中, $Q_{L+1} = [h_1^T h_{L+1} \cdots h_L^T h_{L+1}]^T; P_{L+1} = h_{L+1}^T h_{L+1} + \frac{1}{\gamma}; I_{L+1}$ 为单位矩阵. 由于 A_{L+1} 与 A_L 之间存在(15)式所示的关系, 因此从(10)式所示的 Cholesky 分解计算过程可知, A_{L+1} 的 Cholesky 分解结果 S_{L+1} 中, $s_{11}, s_{21}, \dots, s_{L+1,L}$ 这 $L(L + 1)/2$ 个不为零的元素与 S_L 中不为零的元素相等, 因此不需重新计算, 只需计算 $s_{L+1,1}$ 至 $s_{L+1,L+1}$ 这 $L + 1$ 个不为零的元素即可获得 S_{L+1} . 而且, 此时

$$b_{L+1} = H_{L+1}^T T = \begin{bmatrix} b_L \\ h_{L+1}^T T \end{bmatrix}. \quad (16)$$

因此, 从(12)式所示的 F_L 的计算方式可知

$$F_{L+1} = \begin{bmatrix} F_L \\ f_{L+1} \end{bmatrix}. \quad (17)$$

可见, 只需计算出 f_{L+1} 即可获得 F_{L+1} , 而不需重新计算 f_1 至 f_L . 基于 Cholesky 分解的 β_L 求解方法充分利用了计算 β_L 时所存储的信息, 使得 S_{L+1} 与 F_{L+1} 可分别在 S_L 与 F_L 的基础上获得, 因此计算 β_{L+1} 可在计算 β_L 的基础上进行. 如果在此条件下利用(4)式所示的方法计算 β_{L+1} , 则需重新以矩阵求逆运算的方式计算, 而无法在计算 β_L 的基础上进行.

4. 增量式 RELM 训练算法

研究表明^[11]: 隐层神经元的数量决定了 ELM 的学习精度与泛化能力, 同时也是在设计 ELM 网络

结构时必须事先确定的关键因素. 在 RELM 的网络结构设计时, 也存在相同的问题. 由于实际应用于 RELM 预测模型训练的时间序列训练样本的复杂性, 因此人为凭经验很难精准地给出最佳的隐层神经元数量, 使得 RELM 预测模型既具有足够多的隐层神经元以保证其学习精度, 同时又具有尽可能少的隐层神经元以保持其简约的网络结构. 为避免人为选取隐层神经元数量存在的弊端与困难, 本文提出一种基于 Cholesky 分解的增量式 RELM 训练算法, 使其可自动确定最佳的 RELM 隐层神经元数量, 其训练过程为:

步骤 1 令 RELM 的隐层神经元数量 $L = 1$, 并计算此时的 A_L 与 b_L ;

步骤 2 根据(10)式计算 A_L 的 Cholesky 分解结果 S_L , 并利用 S_L 与 b_L 根据(12)式计算 F_L ;

步骤 3 利用 S_L 与 F_L 根据(13)式计算 β_L , 并在 β_L 的基础上建立(5)式所示的具有 L 个隐层神经元的 RELM 预测模型;

步骤 4 计算此时 RELM 预测模型的经验风险与结构风险之和

$$R_L = \frac{1}{2} \beta_L^T \beta_L + \frac{\gamma}{2} \epsilon^T \epsilon; \quad (18)$$

步骤 5 令 $L = L + 1$ 后, 根据第 3 节所示原理在 S_{L-1} 与 F_{L-1} 的基础上计算 S_L 与 F_L , 然后转至步骤 3. 从 $L = 5$ 开始判断是否同时满足 $| (R_{L-i} - R_{L-i-1}) / R_{\max} | \leq \xi$ (ξ 为学习精度, R_{\max} 为 R_1, \dots, R_L 中的最大值, $i = 0, \dots, 3$). 如果满足则训练过程结束, 确定 L 为最佳隐层神经元数量, 并建立相应的 RELM 预测模型, 否则继续增加 L 直至满足条件.

RELM 的隐层神经元从初始值逐次递增, 并使其在 R_L 不再显著减小时停止扩张. 此时即使继续增加隐层神经元, 代表 RELM 学习精度与泛化能力的 R_L 也不会有显著改善, 反而会导致 RELM 中存在大量冗余的隐层神经元. 因此, 此时的 RELM 具有的隐层神经元数量是最佳的.

5. 混沌时间序列预测实例

以六种典型混沌时间序列 Tent, Kent, Henon, Rossler, Logistic 与 Kawakami 为例, Tent 混沌时间序列由下式描述:

$$x_T(i+1) = \begin{cases} x_T(i)/0.4 & x_T(i) \leq 0.4, \\ (1 - x_T(i))/0.6 & x_T(i) > 0.4. \end{cases} \quad (19)$$

Kent 混沌时间序列由下式描述:

$$x_K(i+1) = 0.9 - 1.9 |x_K(i)|. \quad (20)$$

Henon 混沌时间序列由下式描述:

$$x_H(i+1) = 1 - 0.3x_H(i) - 1.4x_H(i)^2. \quad (21)$$

Rosler 混沌时间序列由下式描述:

$$\begin{cases} \frac{dx_R}{dt} = -y_R - z_R, \\ \frac{dy_R}{dt} = x_R + 0.15y_R, \\ \frac{dz_R}{dt} = 0.2 + z_R(x_R - 10). \end{cases} \quad (22)$$

Logistic 混沌时间序列由下式描述:

$$x_L(i+1) = 4x_L(i)(1 - x_L(i)). \quad (23)$$

Kawakami 混沌时间序列由下式描述:

$$x_K(i+1) = x_K(i)^2 - 0.1x_K(i) - 1.6. \quad (24)$$

选择嵌入维数 $n = 4$, 参数 $\gamma = 10^8$, 学习精度 ξ

$= 10^{-2}$, 隐层神经元函数为 $f(x) = 1/(1 + e^{-x})$. 每种时间序列包含 500 个数据, 利用前 300 个数据训练 RELM 预测模型, 对后 200 个数据进行预测验证,

以均方根误差 $E_L = \sqrt{\sum_{i=1}^{200} (\bar{x}_i - x_i)^2 / 200}$ 衡量预测精度, 其中 x_i 为混沌时间序列真实值, \bar{x}_i 为 x_i 的预测值.

图 1 为 RELM 预测模型在 L 递增条件下应用于六种混沌时间序列预测的 R_L 与 E_L 的变化规律(图中的 R_L 与 E_L 已被归一化至 $[0, 1]$ 区间内). 从图中可见, R_L 与 E_L 均随 L 的增加而表现出逐渐减小的变化趋势, 由于初始的 RELM 预测模型仅具有 1 个隐层神经元, 其学习精度与泛化能力极为有限, 因此 R_L 与 E_L 均在初期随 L 的增加而急剧减小, 表明通过增加隐层神经元可有效提高 RELM 的学习精度与泛化能力. 当 L 增加到一定程度时, R_L 与 E_L 的

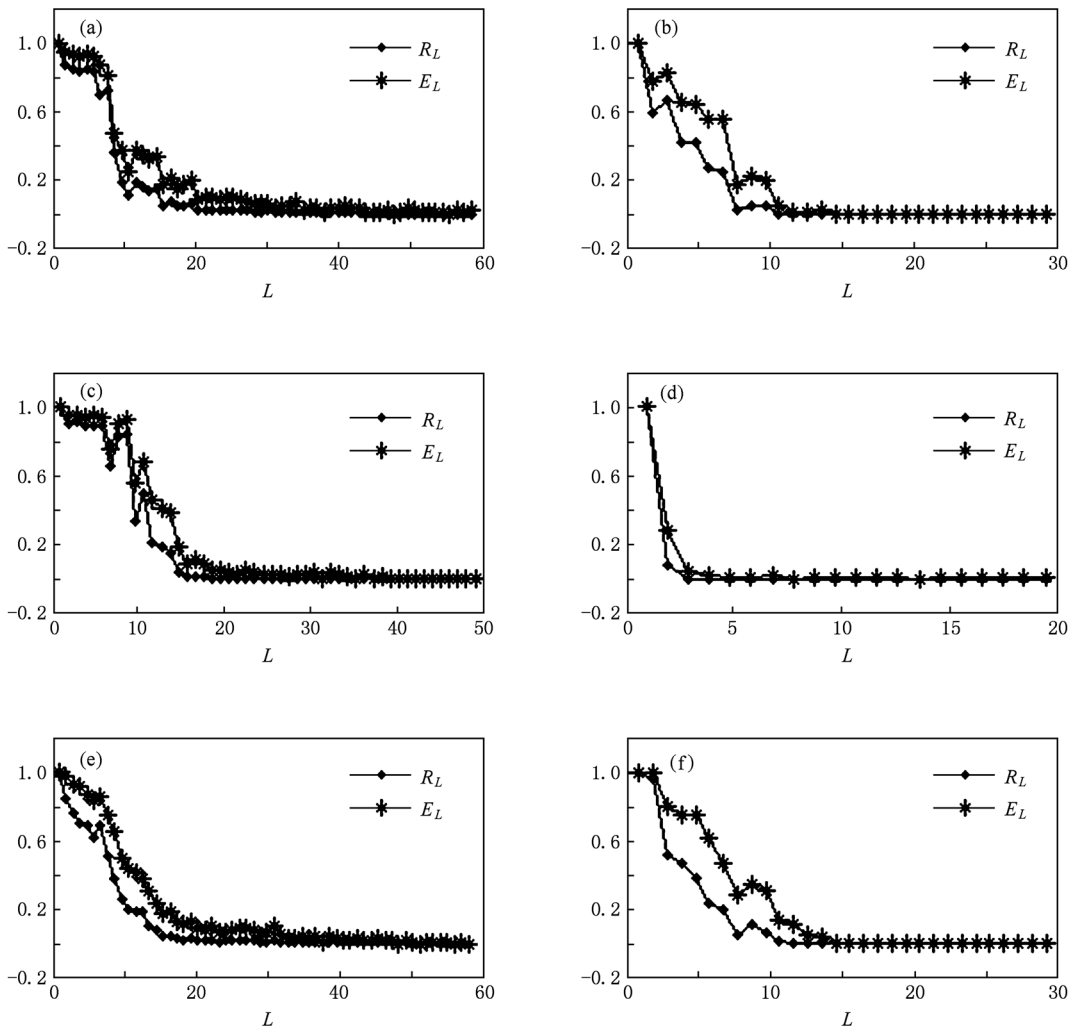


图 1 在 L 递增条件下 R_L 与 E_L 的变化规律 (a)Tent; (b)Kent; (c)Henon; (d)Rosler; (e)Logistic; (f)Kawakami

变化开始逐渐减小,此时即使继续增加 L, R_L 与 E_L 也几乎不再继续减小,表明此时的 RELM 已具有了足够多的隐层神经元,继续增加 L 只会增加 RELM 的网络结构而无法继续提高其学习精度与泛化能力.此时停止增加隐层神经元则可使 RELM 具有最佳的隐层神经元数量.

同时从图 1 可看出, R_L 与 E_L 的变化规律极为相似,当 R_L 随 L 的增加不再继续减小时, E_L 几乎同时达到最小值,而 E_L 代表的是 RELM 预测模型的真实预测误差,表明具有最佳的隐层神经元数量的 RELM 不仅具有最简洁的网络结构,同时还可取最小的预测误差.

图 2 展示了 L 从 1 逐次增加至 50 个时,利用 Cholesky 分解计算输出权值的 RELM 预测模型 (CF-RELM) 与利用矩阵求逆运算计算输出权值的 RELM 预测模型训练所需时间.

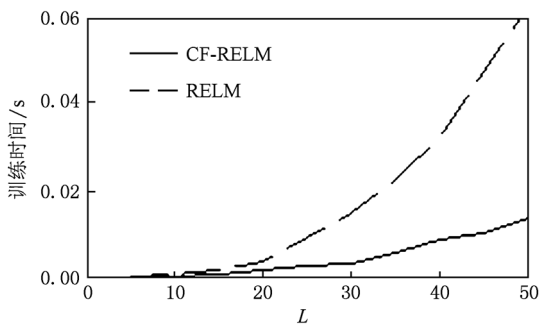


图 2 在 L 递增条件下 CF-RELM 与 RELM 训练所需时间

从图中可见, CF-RELM 的训练时间远少于 RELM,且 L 越多,其优势越明显. CF-RELM 利用

Cholesky 分解有效减小了单次计算输出权值的计算量,且在 L 递增条件下,其输出权值的计算还可在前一次计算的基础上进行,从而进一步减小了计算量,因此相比于 RELM,CF-RELM 的计算效率更高.

表 1 展示了利用本文提出的增量式 RELM 算法所确定的应用于六种混沌时间序列预测的 CF-RELM 预测模型的最佳隐层神经元数量及其对应的预测误差与训练时间,并将其与两种典型的增量式 ELM 训练算法做比较:一种为文献[12]提出的利用矩阵分块运算计算输出权值的增量式 ELM 训练算法 (EM-ELM),另一种为文献[13]提出的利用函数逼近原理计算输出权值的增量式 ELM 训练算法 (EI-ELM).表中的预测误差为均方根误差 E_L ,训练时间为隐层神经元数量 L 分别从 1 个增长至最佳隐层神经元数量时的训练时间.

从表中数据可见,CF-RELM 的预测误差明显小于 EM-ELM 与 EI-ELM,且训练时间最短,因此其综合性能最优. EM-ELM 虽在 L 递增的条件下也避免了通过矩阵逆运算的方式计算 ELM 输出权值,但其计算过程仍涉及较为复杂的分块矩阵乘法运算,因此其训练时间几乎是 CF-RELM 与 EI-ELM 的 10 倍,计算效率最低;EI-ELM 是一种基于函数逼近原理的增量式 ELM 训练算法,以逐次增加隐层神经元的方式不断减小 EI-ELM 与待逼近函数间的误差,但应用于时间序列建模预测时,其预测误差却明显大于 CF-RELM 与 EM-ELM.因此,CF-RELM 兼具预测精度高、计算速度快与自动确定最佳网络结构这三大优点,适用于混沌时间序列预测.

表 1 CF-RELM,EM-ELM 与 EI-ELM 对于六种混沌时间序列的预测结果

混沌时间序列	最佳隐层神经元数量	预测误差			训练时间/ms		
		CF-RELM	EM-ELM	EI-ELM	CF-RELM	EM-ELM	EI-ELM
Tent	25	0.0803	0.1331	0.2067	2.2761	24.4497	2.6283
Kent	21	0.0644	0.0671	0.1358	1.6987	20.8494	2.1785
Henon	20	0.0008	0.0013	0.0040	1.3757	18.9657	2.0184
Rosler	10	0.0024	0.0024	0.0164	0.3154	7.9173	0.9350
Logistic	23	0.0177	0.0194	0.0492	1.8761	22.1499	2.3453
Kawakami	16	0.0087	0.0089	0.0262	1.2625	14.4736	1.6483

6. 结 论

RELM 通过引入参数 γ 权衡 ELM 的结构风险与

经验风险,其泛化能力较传统神经网络有显著提高.作为一种可自动确定最佳网络结构的 RELM 改进模式,CF-RELM 同样具备了这一优势,并以 Cholesky 分解方式计算其输出权值,有效减小了隐层神经元递增

过程所带来的计算代价. 六种混沌时间序列预测实例证明, CF-RELM 可有效确定 RELM 的最佳网络结构,

且具有预测精度高与计算速度快的优点, 可为混沌时间序列预测提供一条新颖且高效的解决途径.

- [1] Muhammad A F, Zolfaghari S 2010 *Neurocomputing* **73** 2540
- [2] Vairappan C, Tamura H, Gao S C 2009 *Neurocomputing* **72** 1870
- [3] Han M, Wang Y 2009 *Expert. Syst. Appl.* **36** 1280
- [4] Lee C C, Chiang Y C, Shih C Y, Tsai C L 2009 *Expert. Syst. Appl.* **36** 4717
- [5] Mirzaee H 2009 *Chaos Soliton Fract.* **41** 2681
- [6] Haniyas M P, Karras D A 2009 *ENG Appl. Artif. Intel.* **22** 32
- [7] Xiu C B, Xu M 2010 *Acta Phys. Sin.* **59** 7650 (in Chinese) [修春波、徐 勳 2010 物理学报 **59** 7650]
- [8] Ma Q L, Zheng Q L, Peng H, Qin J W 2009 *Acta Phys. Sin.* **58** 1410 (in Chinese) [马千里、郑启伦、彭 宏、覃姜维 2009 物理学报 **58** 1410]
- [9] Zhang J F, Hu S S 2007 *Acta Phys. Sin.* **56** 713 (in Chinese) [张军峰、胡寿松 2007 物理学报 **56** 713]
- [10] Li J, Liu J H 2005 *Acta Phys. Sin.* **54** 4569 (in Chinese) [李军、刘君华 2005 物理学报 **54** 4569]
- [11] Huang G B, Zhu Q Y, Siew C K 2006 *Neurocomputing* **70** 489
- [12] Feng G, Huang G B, Lin Q P, Gay R 2009 *IEEE Neural Netw.* **20** 1352
- [13] Huang G B, Chen L 2008 *Neurocomputing* **71** 3460
- [14] Liu N, Wang H 2010 *IEEE Signal Pro. Lett.* **17** 754
- [15] Lan Y, Soh C Y, Huang G B 2010 *Neurocomputing* **73** 3191
- [16] Malathi V, Marimuthu N S, Baskar S 2010 *Neurocomputing* **73** 2160
- [17] Deng W Y, Zheng Q H, Chen L, Xu X B 2010 *Chin. J. Comp.* **33** 279 (in Chinese) [邓万字、郑庆华、陈 琳、许学斌 2010 计算机学报 **33** 279]
- [18] 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]

Incremental regularized extreme learning machine based on Cholesky factorization and its application to time series prediction

Zhang Xian[†] Wang Hong-Li

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

(Received 11 January 2011; revised manuscript received 23 February 2011)

Abstract

In order to solve the hidden-layer neuron determination problem of regularized extreme learning machine (RELM) applied to chaotic time series prediction, a new algorithm based on Cholesky factorization is proposed. First, an RELM-based prediction model with one hidden-layer neuron is constructed and then a new hidden-layer neuron is added to the prediction model in each training step until the generalization performance of the prediction model reaches its peak value. Thus, the optimal network structure of the prediction model is determined. In the training procedure, Cholesky factorization is used to calculate the output weights of RELM. Experiments on chaotic time series prediction indicate that the algorithm can be effectively used to determine the optimal network structure of RELM, and the prediction model trained by the algorithm has excellent performance in prediction accuracy and computational cost.

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

PACS: 02.10.Yn, 05.45.Ac, 05.45.Tp

[†] E-mail: sltecas@sohu.com