

DOI: 10.7641/CTA.2017.60309

基于自组织模糊神经网络的出水总磷预测

乔俊飞^{1,2†}, 周红标^{1,2,3}

(1. 北京工业大学 信息学部, 北京 100124; 2. 计算智能和智能系统北京市重点实验室, 北京 100124;

3. 淮阴工学院 自动化学院, 江苏 淮安 223003)

摘要: 针对污水处理过程出水总磷预测问题, 本文提出一种基于改进 Levenberg-Marquardt(improved Levenberg-Marquardt, ILM)学习算法和奇异值分解(singular value decomposition, SVD)的适于在线建模的自组织模糊神经网络(fuzzy neural network, FNN)预测方法. ILM-SVDFNN采用改进LM学习算法对隶属函数中心、宽度和输出权值进行训练. 在参数自适应学习的同时, 采用单边Jacobi变换实现规则层输出阵的奇异值分解, 根据奇异值定义增长和修剪指标实现规则层神经元在线动态调整. 此外, 证明了所提方法在网络结构固定和调整阶段的收敛性. 最后, 利用典型非线性系统辨识、Mackey-Glass时间序列预测和实际污水处理过程出水总磷预测实验进行验证. 仿真结果显示所设计的自组织模糊神经网络结构紧凑且预测精度较高, 较好地满足了污水处理厂对出水总磷检测精度和实时性的要求.

关键词: 出水总磷; 模糊神经网络; 自组织模糊神经网络; 改进Levenberg-Marquardt; 奇异值分解

中图分类号: TP183 **文献标识码:** A

Prediction of effluent total phosphorus based on self-organizing fuzzy neural network

QIAO Jun-fei^{1,2†}, ZHOU Hong-biao^{1,2,3}

(1. Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China;

2. Beijing Key Laboratory of Computational Intelligence and Intelligent System, Beijing 100124, China;

3. Faculty of Automation, Huaiyin Institute of Technology, Huai'an Jiangsu 223003, China)

Abstract: A novel online self-organizing fuzzy neural network (FNN) based on the improved Levenberg-Marquardt (ILM) learning algorithm and singular value decomposition (SVD) is proposed to predict the effluent total phosphorus (TP) in a wastewater treatment process. The centers and widths of membership functions and weights of output layer are trained by ILM learning algorithm. Meanwhile, the output matrix of the rule layer is decomposed with SVD, which is implemented by one-sided Jacobi's transformation. The neurons of rule layer are adjusted dynamically with growing and pruning algorithms, which are based on the singular values. In addition, the convergence of the proposed ILM-SVDFNN has been proved both in the structure fixed phase and the structure adjusting phase. Finally, the validity and practicability of the model are illustrated with three examples, including typical nonlinear system identification, Mackey-Glass time series prediction, and prediction of effluent TP. Simulation results demonstrate that the proposed ILM-SVDFNN generates a fuzzy neural network automatically and effectively with a highly accurate and compact structure, and it can well satisfy the detection accuracy and real-time requirements of the prediction of effluent TP.

Key words: effluent total phosphorus; fuzzy neural network; self-organizing fuzzy neural network; improved Levenberg-Marquardt; singular value decomposition

1 引言(Introduction)

近年来, 随着总量减排工作的持续推进, 污水处理过程(wastewater treatment process, WWTP)氨氮和化学需氧量(chemical oxygen demand, COD)排放总量

大幅下降, 但总磷(total phosphorus, TP)污染问题却日益严重. 研究表明, TP与水体富营养化密切相关, 是污水处理厂排放标准的核心指标之一, 加强对TP的检测有助于解决富营养化问题^[1]. 目前污水处理厂主要通

收稿日期: 2016-05-11; 录用日期: 2016-11-09.

†通信作者. E-mail: junfeiq@bjut.edu.cn; Tel.: +86 10-67391766.

本文责任编辑: 邓飞其.

国家杰出青年科学基金项目(61225016), 国家自然科学基金重点项目(61533002)资助.

Supported by National Science Foundation for Distinguished Young Scholars of China (61225016) and State Key Program of National Natural Science of China (61533002).

过化学分析法(如钼酸铵分光光度法)在专门实验室进行TP检测, 测量精度较高, 但操作繁琐、费时和费力, 不能满足TP测量的实时性需求^[2]. 由于基于数据驱动的软测量技术能够在线、快速和准确预测且结果不易受环境干扰, 克服了采用化学方法或在线仪表测量的固有缺陷, 受到研究者青睐^[3]. 其中神经网络凭借能够以任意精度逼近任何非线性系统的能力, 在污水处理过程^[4]关键变量软测量^[5]和智能控制^[6]中有着广泛应用.

模糊神经网络(fuzzy neural network, FNN)是模糊系统和神经网络的有机结合, 它不仅具有模糊逻辑处理不确定、非线性和不适定问题的能力, 而且具有神经网络并行计算、分布式信息存储和自适应学习等优势, 可实现隶属函数动态调整和规则在线优化, 故与传统的多层感知器(multi-layer perceptron, MLP)等前馈网络相比, 具有更好的性能^[7]. 然而传统FNN参数学习普遍采用误差反向传播(error back-propagation, EBP)算法, 存在收敛速度慢、易陷入局部极值点和精度低等缺陷. 同时由于出水总磷预测属于复杂非线性系统建模, 无法预先设定合适的规则数, 网络运行经常处于不饱和状态, 降低了网络泛化能力.

为了利用FNN实现污水处理出水总磷快速在线预测, 需要克服一阶算法的缺点. 作为典型的二阶算法, Levenberg-Marquardt(LM)算法是梯度下降法和牛顿法的结合, 收敛速度快且精度高, 但需要计算Jacobian矩阵的相乘以及存储类Hessian矩阵. 当样本数目较多时, 计算量和存储量较大, 网络运行时间长, 给快速建模带来不利影响^[8]. 改进LM算法只需计算Jacobian矩阵行向量相乘, 只与参数个数有关, 对样本数量无限制, 计算时间和存储空间大为减少, 尤其适合于解决快速在线建模问题^[9].

此外在保证精度的前提下, 网络结构越精简, 泛化能力越强, 因此FNN结构自组织设计受到研究者普遍关注. 神经网络结构优化方法主要有以交叉验证为代表的规则化法, 以瀑流关联构造为代表的自底向上法, 以修剪为代表的自顶向下法和以遗传算法为代表的群智能优化法等^[10]. 本文采用增长修剪相结合的算法, 能够从一个小规模的网络结构开始训练(一般取1-2个初始节点), 学习过程中动态调整网络结构, 从而避免了增长型神经网络易出现网络冗余的情况. 同时也可以设定一个适当大的初始网络结构(一般要稍大于输入节点个数的2倍), 在训练过程中逐渐删除冗余神经元, 解决神经网络结构过大的问题. 增长修剪型结构自组织优化算法最终能够获取结构精简、泛化能力强、学习速度快和性能稳定的神经网络. Wu等^[11]采用扩展的径向基函数构建动态模糊神经网络(dynamic fuzzy neural network, DFNN), 但其隶属函数中心和宽度不能在线调整, 且需设置参数过多. Wu等^[12]采

用椭圆基函数构建广义动态模糊神经网络(generalized dynamic fuzzy neural network, GDFNN), 但其规则的产生依赖全部训练样本, 限制其在线学习能力. De等^[13]采用最小二乘算法构建在线自组织模糊神经网络(online self-organizing fuzzy modified least-squares, SOFMLS), 但其规则的增长和修剪依赖于预先设定的半径, 而这个半径难以确定. Wang等^[14]采用误差下降率和卡尔曼滤波构建在线自组织模糊神经网络(fast and accurate online self-organizing parsimonious fuzzy neural network, FAOS-PFNN), 但其参数和结构优化的本质等同于DFNN. Han等^[15]采用敏感度分析构建增长修剪型自组织模糊神经网络(growing and pruning fuzzy neural network, GP-FNN), 但其采用梯度下降法训练网络参数, 收敛速度较慢. Han等^[16]将尖峰强度和互信息相结合构建自组织神经网络(self-organizing fuzzy-neural-network with adaptive computation algorithm, SOFNN-ACA), 但其采用一阶梯度算法, 收敛速度慢, 同时由于引入两个机制, 阈值设置存在困难. 此外还有基于进化算法^[17]采取时间换性能的方式构建自组织网络, 由于耗时太长, 不适宜在线建模.

因此, 针对上述问题, 本文采用单边Jacobi变换实现奇异值分解(singular value decomposition, SVD), 根据规则层输出阵奇异值的大小对规则层神经元进行增长和修剪, 并结合改进LM算法, 构建一种基于SVD的快速在线自组织模糊神经网络(ILM-SVDFNN), 并用于典型非线性系统辨识、时间序列预测和污水处理出水总磷预测.

2 模糊神经网络(Fuzzy neural network)

FNN主要有Mamdani和T-S(Takagi-Sugeno)两种模型, 本文采用Mamdani模型构建4层FNN^[11]. FNN网络结构如图1所示, 主要由输入层、隶属函数层、规则层和输出层组成.

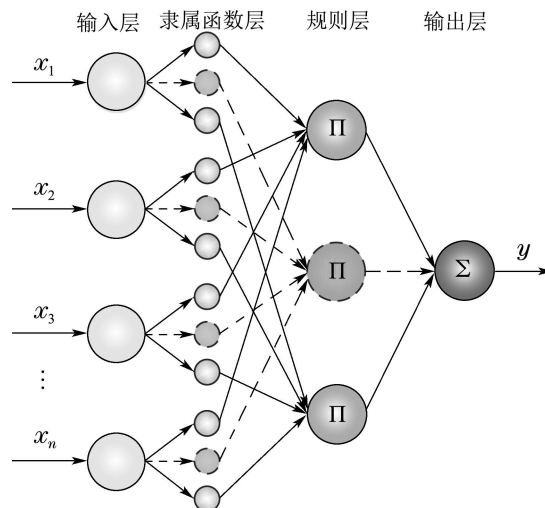


图1 模糊神经网络结构

Fig. 1 The structure of FNN

第1层为输入层,有 n 个输入神经元,该层各神经元直接与输入向量各分量 x_i 相连,其输出等于输入,即

$$u_i = x_i, i = 1, 2, \dots, n, \quad (1)$$

其中 u_i 是输入层第 i 个神经元的输出。

第2层为隶属函数层,由于每个输入神经元有 m 个隶属函数,故该层有 $n \times m$ 个神经元.学习过程中需要对高斯隶属函数的中心和宽度进行调整.该层神经元的输出为隶属度值,即^[11,15]

$$\varphi_{ij} = e^{-\frac{(x_i - c_{ij})^2}{\sigma_{ij}^2}}, j = 1, 2, \dots, m, \quad (2)$$

其中: c_{ij} 和 σ_{ij} 分别是第1层第 i 个神经元对应的第2层第 j 个神经元的高斯函数的中心和宽度, φ_{ij} 是相应隶属函数层神经元的输出。

第3层为规则层,该层有 m 个神经元,主要对第2层神经元执行 Π 和归一化操作.神经元输出表示模糊规则激活强度,代表了规则的前件部分,即

$$v_j = \prod_{i=1}^n \varphi_{ij} = e^{-\sum_{i=1}^n \frac{(x_i - c_{ij})^2}{\sigma_{ij}^2}}, \quad (3)$$

$$h_j = v_j / \sum_{j=1}^m v_j = e^{-\sum_{i=1}^n \frac{(x_i - c_{ij})^2}{\sigma_{ij}^2}} / \sum_{j=1}^m e^{-\sum_{i=1}^n \frac{(x_i - c_{ij})^2}{\sigma_{ij}^2}}, \quad (4)$$

其中 v_j 和 h_j 分别是规则层第 j 个神经元输出和归一化输出,结构优化时规则层神经元数将在线调整。

第4层为输出层,该层有1个神经元,对输入进行加权求和,实现清晰化计算,输出为

$$y = \sum_{j=1}^m h_j w_j, \quad (5)$$

其中: y 为神经元的输出, w_j 为规则层和输出层之间的连接权值。

3 改进LM学习算法的SVDFNN (SVDFNN with improved LM learning algorithm)

3.1 改进LM学习算法(Improved LM learning algorithm)

FNN需要学习的参数主要是第3层和第4层之间的连接权值 w_j 以及第2层隶属函数的中心 c_{ij} 和宽度 σ_{ij} .针对FNN在线建模,给定 P 对输入-输出样本 $\{\mathbf{x}(p), y_d(p), p = 1, 2, \dots, P\}$,单个样本送入网络训练,产生误差 e^p :

$$e^p = y_d^p - y^p, \quad (6)$$

其中: y_d^p 为期望输出, y^p 为网络实际输出, $p = 1, 2, \dots, P$ 为训练样本个数。

Jacobian矩阵行向量 \mathbf{j}_p 可以表达为^[8]

$$\mathbf{j}_p = \left[\frac{\partial e_p}{\partial w_1}, \dots, \frac{\partial e_p}{\partial w_m}, \frac{\partial e_p}{\partial c_{11}}, \dots, \frac{\partial e_p}{\partial c_{nm}}, \frac{\partial e_p}{\partial \sigma_{11}}, \dots, \frac{\partial e_p}{\partial \sigma_{nm}} \right]. \quad (7)$$

根据式(6)和(7)计算梯度子向量 $\boldsymbol{\eta}_p$ 和子矩阵 \mathbf{q}_p :

$$\boldsymbol{\eta}_p = \mathbf{j}_p e_p, \quad (8)$$

$$\mathbf{q}_p = \mathbf{j}_p^T \mathbf{j}_p. \quad (9)$$

从式(9)可以看出,改进LM算法避免了LM算法中Jacobian矩阵的相乘,其仅需对行向量 \mathbf{j}_p 相乘,减少了计算时间和存储空间.采用循环累加方式对当前滑动窗内所有训练样本计算梯度向量 \mathbf{g} 和类Hessian矩阵 \mathbf{Q} :

$$\mathbf{g} = \sum_{p=1}^P \boldsymbol{\eta}_p, \quad (10)$$

$$\mathbf{Q} = \sum_{p=1}^P \mathbf{q}_p. \quad (11)$$

改进LM算法参数更新公式为^[9]

$$\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) + (\mathbf{Q}(t) + \mu \mathbf{I})^{-1} \mathbf{g}(t), \quad (12)$$

其中 $\boldsymbol{\theta} = [w_1, \dots, w_m, c_{11}, \dots, c_{nm}, \sigma_{11}, \dots, \sigma_{nm}]$.将式(10)和(11)代入式(12)实现参数的在线学习。

3.2 单边Jacobi实现SVD(SVD based on one-sided Jacobi)

设规则层输出阵 $\mathbf{H} \in \mathbb{R}^{P \times m}$ 和权系数矩阵 $\mathbf{W} \in \mathbb{R}^{m \times 1}$,则有

$$\mathbf{Y} = \mathbf{H} \times \mathbf{W}, \quad (13)$$

$$\tilde{\mathbf{E}} = \|\mathbf{Y}_d - \mathbf{Y}\|, \quad (14)$$

目标是寻找一个最优的权系数向量 \mathbf{W}^* 使得误差能量 $\tilde{\mathbf{E}}^T \tilde{\mathbf{E}}$ 最小化.考虑式(13)和(14),构建线性回归模型

$$\mathbf{Y}_d = \mathbf{H} \boldsymbol{\theta} + \mathbf{E}. \quad (15)$$

对 \mathbf{H} 可以用SVD方法分解为^[18]

$$\mathbf{H} = \mathbf{U} \boldsymbol{\Sigma} \mathbf{S}^T, \quad (16)$$

其中: $\mathbf{U} \in \mathbb{R}^{P \times P}$ 和 $\mathbf{S} \in \mathbb{R}^{m \times m}$ 为正交矩阵, $\boldsymbol{\Sigma} = \text{diag}\{\xi_1, \xi_2, \dots, \xi_m\}$ 为对角矩阵,对角元素 $\xi_1 \geq \xi_2 \geq \dots \geq \xi_m \geq 0$ 即为矩阵 \mathbf{H} 的奇异值。

根据 $\text{rank}(\mathbf{H}) = \text{rank}(\boldsymbol{\Sigma})$,设 $b = \text{rank}(\mathbf{H})$,则式(16)可以表示成如下形式:

$$\mathbf{H} = \sum_{i=1}^b \xi_i \mathbf{u}_i \mathbf{s}_i^T, \quad (17)$$

其中: $\xi_1, \xi_2, \dots, \xi_b$ 是矩阵 \mathbf{H} 的 b 个非零奇异值, \mathbf{u}_i 和 \mathbf{s}_i 分别是矩阵 \mathbf{U} 和 \mathbf{S} 的第 i 列.把式(17)代入式(15),可以得到:

$$\boldsymbol{\theta} = \sum_{i=1}^b \frac{\mathbf{u}_i^T \mathbf{Y}_d}{\xi_i} \mathbf{s}_i, \quad (18)$$

可以证明这个解最小化 $\|\mathbf{Y}_d - \mathbf{H} \boldsymbol{\theta}\|_2$ 的同时具有最小的2-范数^[11]. $\boldsymbol{\theta}$ 可以近似为

$$\hat{\boldsymbol{\theta}} = \sum_{i=1}^b \frac{\mathbf{u}_i^T \mathbf{Y}_d}{\xi_i} \mathbf{s}_i, \quad (19)$$

其中 \hat{b} 由 ξ_i 来决定。

显然, 当 $\hat{b} \leq b$ 时, $\hat{\theta}$ 最小化 $\|Y_d - \hat{H}\theta'\|_2$, 可以用 \hat{H} 代替 H :

$$\hat{H} = \sum_{i=1}^{\hat{b}} \xi_i \mathbf{u}_i \mathbf{s}_i^T. \quad (20)$$

上述理论反映了对较小奇异值的过滤, 这在图像去噪和主成分分析等场合有着广泛应用。奇异值的大小反映与之对应的规则层神经元对整个网络的贡献程度。较小奇异值对应的是冗余规则, 可以通过剔除冗余规则增强网络泛化能力。MATLAB 中SVD函数得到的奇异值从大到小排列, 且该函数底层采用Fortran编程, 内部封装, 无法对其进行编辑。本文采用单边Jacobi变换实现SVD, 以确定奇异值在 H 阵中的原始位置。

单边Jacobi变换^[19]的核心思想是采用一系列Jacobi平面旋转变换对矩阵 H 进行正交化 $B = H(J_1 J_2 J_3 \dots)$, 使得 B 中任意两列向量满足 $\mathbf{b}_i^T \mathbf{b}_j = 0$, 然后对 B 归一化得到 $B = U\Sigma$ 。其中 Σ 对角阵中对角元素 $\xi_i = \mathbf{b}_i^T \mathbf{b}_i$ 即为奇异值。

对矩阵 H 进行正交化时, 每次仅有 i 与 j 两列元素受到影响, 具体变换过程如式(21):

$$\begin{pmatrix} \mathbf{b}_i^T & \mathbf{b}_j^T \end{pmatrix} = \begin{pmatrix} \mathbf{h}_i^T & \mathbf{h}_j^T \end{pmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix}. \quad (21)$$

对式(21)计算得到 $\mathbf{b}_i^T = c\mathbf{h}_i^T - s\mathbf{h}_j^T$, $\mathbf{b}_j^T = s\mathbf{h}_i^T + c\mathbf{h}_j^T$, 最终解得 c 和 s 分别为

$$c = 1/\sqrt{1+t^2}, \quad (22)$$

$$s = t \times c, \quad (23)$$

其中 $t = \frac{\text{sgn } \tau}{|\tau| + \sqrt{1 + \tau^2}}$, $\tau = \frac{\mathbf{h}_j^T \mathbf{h}_j - \mathbf{h}_i^T \mathbf{h}_i}{2\mathbf{h}_i^T \mathbf{h}_j}$ 。

3.3 基于SVD的自组织FNN(SOFNN based on SVD)

利用SVD分解FNN规则层输出阵, 比较奇异值的大小并结合网络学习过程训练集均方根误差实现规则层神经元的生长和修剪, 从而实现网络结构的在线优化^[15-16]。

增长和修剪两个评价指标及均方根误差分别定义如下:

$$I_g = \max(\xi) / \sum_{s=1}^{n_s} \xi_s, \quad (24)$$

$$I_d = \sum_{s=1}^d \xi_s / \sum_{s=1}^{n_s} \xi_s, \quad (25)$$

$$E(\theta(t)) = \sqrt{\frac{1}{P} \sum_{p=1}^P (y_d^p(t) - y^p(t))^2}, \quad (26)$$

其中: I_g 是增长指标, I_d 是修剪指标, 奇异值向量 ξ 从小到大排列, I_d 是当前 d 个较小奇异值之和占全部奇异值的比值, n_s 是当前奇异值的个数。

3.3.1 结构增长算法(Structure growing algorithm)

当 $E(\theta(t)) > E(\theta(t-1))$ 且 $I_g > I_{gth}$ (I_{gth} 为预设的增长阈值)时, 对当前奇异值最大的神经元 g 进行分裂, 从而调整网络结构以提高性能。新增神经元及神经元 g 的参数为

$$c_{\text{new}}(t) = \frac{1}{2}(c_g(t) + x(t)), \quad (27)$$

$$\sigma_{\text{new}}(t) = \sigma_0(t), \quad (28)$$

$$w_{\text{new}}(t) = \frac{y_d(t) - y(t)}{e^{-\sum_{i=1}^n \frac{(x_i(t) - c_{i\text{new}}(t))^2}{\sigma_{i\text{new}}^2(t)}}}, \quad (29)$$

$$c'_g(t) = c_g(t), \quad (30)$$

$$\sigma'_g(t) = \sigma_g(t), \quad (31)$$

$$w'_g(t) = w_g(t), \quad (32)$$

其中: $c_{\text{new}}(t)$ 和 $\sigma_{\text{new}}(t)$ 分别是新增神经元对应隶属函数层神经元的中心和宽度; $w_{\text{new}}(t)$ 是新增神经元输出权值; $x(t)$, $y_d(t)$ 和 $y(t)$ 分别是当前训练样本的输入、期望输出和网络输出; $\sigma_0(t)$ 是网络初始化时预设的一组宽度; $c_g(t)$ 和 $c'_g(t)$, $\sigma_g(t)$ 和 $\sigma'_g(t)$ 分别是神经元 g 对应隶属函数层神经元结构调整前后的中心、宽度; $w_g(t)$ 和 $w'_g(t)$ 分别是神经元 g 结构调整前后的权值。

3.3.2 结构修剪算法(Structure pruning algorithm)

当 $E(\theta(t)) < E(\theta(t-1))$ 且 $I_d < I_{dth}$ (I_{dth} 为预设的修剪阈值), 当前 d 个较小奇异值对应的神经元 s ($s = 1, 2, \dots, d$)将被删除。假定奇异值最大的神经元为 g , 则神经元 s 和 g 的参数为

$$c'_g(t) = c_g(t), \quad (33)$$

$$\sigma'_g(t) = \sigma_g(t), \quad (34)$$

$$w'_g(t) =$$

$$\left[w_g(t) e^{-\sum_{i=1}^n \frac{(x_i(t) - c_{ig}(t))^2}{\sigma_{ig}^2(t)}} + \sum_{s=1}^d w_s(t) e^{-\sum_{i=1}^n \frac{(x_i(t) - c_{is}(t))^2}{\sigma_{is}^2(t)}} \right] / e^{-\sum_{i=1}^n \frac{(x_i(t) - c_{ig}(t))^2}{\sigma_{ig}^2(t)}}, \quad (35)$$

$$c'_s(t) = 0, \quad (36)$$

$$\sigma'_s(t) = 0, \quad (37)$$

$$w'_s(t) = 0, \quad (38)$$

其中: $c_g(t)$ 和 $c'_g(t)$, $\sigma_g(t)$ 和 $\sigma'_g(t)$, $w_g(t)$ 和 $w'_g(t)$ 如增长过程所述; $c'_s(t)$ 和 $\sigma'_s(t)$ 分别是 d 个被修剪规则层神经元对应的隶属函数层神经元调整后的中心和宽度; $w_s(t)$ 和 $w'_g(t)$ 是分别是 d 个被修剪神经元调整前后的输出权值; $x(t)$ 是当前输入样本。

基于以上分析, ILM-SVDFNN算法运行的具体步骤如下^[15]:

步骤1 给定一个规则数为任意自然数的模糊神经网络, 初始化中心、宽度和权值参数; 设置目标误差

E_d 、阻尼因子 μ 、增长阈值 I_{gth} 和修剪阈值 I_{dth} ;对网络进行训练.

步骤2 根据式(1)–(5)计算滑动窗内当前样本输出;根据式(6)–(9)计算梯度子向量 η_p 和子矩阵 q_p ;根据式(10)和(11),计算梯度向量 g 和类Hessian矩阵 Q .

步骤3 如果滑动窗内所有样本都计算完毕,则根据式(12)计算更新后的网络参数值;否则,跳往步骤2计算窗内下一个样本.

步骤4 利用当前网络对所有训练集样本进行测试,根据式(26)计算均方根误差 $E(\theta(t))$.

步骤5 每隔一定步数,利用单边Jacobi方法计算 H 阵奇异值.根据增长判断条件对规则层神经元进行分析,如需分裂,利用式(27)–(29)对新增神经元进行参数调整,且利用式(30)–(32)对被分裂神经元进行参数设置;否则,跳往步骤6.

步骤6 根据修剪条件对规则层神经元进行分析,如需修剪,根据式(33)–(35)对最大奇异值对应的神经元进行参数调整,并同时利用式(36)–(38)对被修剪神经元进行参数设置.

步骤7 直到全部训练样本都学习完毕或满足停止条件 $E(\theta(t)) < E_d$;否则,跳往步骤2重复训练.

3.4 收敛性分析(Convergence analysis)

ILM-SVDFNN收敛性证明过程主要分两部分:结构固定阶段和结构调整阶段,而结构自组织调整阶段又分神经元增长和删减两个方面^[15-16].假设在 t 时刻规则层有 m 个神经元,当前逼近误差为 $\varepsilon(t)$.

3.4.1 网络结构固定阶段(Structure fixed phase)

定义一个离散Lyapunov方程

$$V(\theta(t)) = \frac{1}{2} \varepsilon^T(t) \varepsilon(t), \quad (39)$$

其中 $\varepsilon(t) = y_d^p - y^p$. Lyapunov函数 $V(\theta(t))$ 的变化量为

$$\begin{aligned} \Delta V(\theta(t)) &= \\ V(\theta(t+1)) - V(\theta(t)) &= \\ -\nabla E^T(\theta(t)) \Delta \theta(t) + \\ \frac{1}{2} \Delta \theta^T(t) \nabla^2 E(\theta(t)) \Delta \theta(t), \end{aligned} \quad (40)$$

其中

$$\begin{cases} \Delta \theta(t) = (Q(\theta(t)) + \mu I)^{-1} g(\theta(t)), \\ \nabla E(\theta(t)) = g(\theta(t)), \\ \nabla^2 E(\theta(t)) = Q(\theta(t)) + \mu I. \end{cases} \quad (41)$$

如果

$$\|\Delta \theta(t)\| \leq \min\left\{\|\Delta \theta(t-1)\|, \frac{g(\theta(t-1))}{Q(\theta(t-1))}\right\}, \quad (42)$$

则根据式(41)(42),可得出

$$\Delta V(\theta(t)) = -\frac{1}{2} \Delta \theta^T(t) \nabla^2 E(\theta(t)) \Delta \theta(t), \quad (43)$$

当式(42)成立时,二阶矩阵 $\nabla^2 E(\theta(t))$ 为正定矩阵,可得 $\Delta V(\theta(t)) < 0$.根据Lyapunov定理, $\lim_{t \rightarrow \infty} \varepsilon(t) = 0$,故网络结构固定时能够保证收敛性.

3.4.2 网络结构增长阶段(Structure growing phase)

若在 t 时刻规则层需要增加神经元(网络结构增长后变成 $m+1$),则增长后网络输出逼近误差变为

$$\begin{aligned} \varepsilon'(t) &= y_d(t) - y'(t) = \\ y_d(t) - \mathbf{h}(t) \mathbf{W}'(t) &= \\ y_d(t) - \sum_{j=1}^m w_j(t) h_j(t) - w_{m+1}(t) h_{m+1}(t), \end{aligned} \quad (44)$$

其中: $y_d(t)$ 是期望输出, $y(t)$ 是规则层有 m 个神经元时的输出, $y'(t)$ 是规则层有 $m+1$ 个神经元的输出,将式(27)–(32)代入式(44),可得

$$\begin{aligned} \varepsilon'(t) &= \\ y_d(t) - y(t) - \frac{y_d(t) - y(t)}{e^{-\sum_{i=1}^n \frac{(x_i(t) - c_{i\text{new}}(t))^2}{\sigma_{i\text{new}}^2(t)}}} h_{m+1}(t) &= \\ y_d(t) - y(t) - (y_d(t) - y(t)) &= \\ \varepsilon(t) - \varepsilon(t) &= 0. \end{aligned} \quad (45)$$

可见,新增神经元并没有对网络输出造成突变,网络逼近误差反而得到补偿,因此采用式(27)–(32)的参数设置方式并未改变网络收敛性.

3.4.3 网络结构删减阶段(Structure pruning phase)

若在 t 时刻规则层需要删减的神经元为 d 个神经元($s(s=1, 2, \dots, d)$),参数需要补偿的神经元为 g ,则删减后网络输出逼近误差将变为

$$\begin{aligned} \varepsilon'(t) &= y_d(t) - y'(t) = \\ y_d(t) - \sum_{j=1, j \neq s}^m w_j(t) h_j(t). \end{aligned} \quad (46)$$

根据式(33)–(38)给定的参数调整规则,式(46)变为

$$\begin{aligned} \varepsilon'(t) &= \\ y_d(t) - \sum_{j=1, j \neq s, j \neq g}^m w_j(t) h_j(t) - w_g(t) h_g(t) &= \\ y_d(t) - \sum_{j=1, j \neq s, j \neq g}^m w_j(t) h_j(t) - \\ [w_g(t) e^{-\sum_{i=1}^n \frac{(x_i(t) - c_{ig}(t))^2}{\sigma_{ig}^2(t)}} + \\ \sum_{s=1}^d w_s(t) e^{-\sum_{i=1}^n \frac{(x_i(t) - c_{is}(t))^2}{\sigma_{is}^2(t)}}] e^{-\sum_{i=1}^n \frac{(x_i(t) - c_{ig}(t))^2}{\sigma_{ig}^2(t)}}. \end{aligned}$$

$$h_g(t) = y_d(t) - \sum_{j=1}^m w_j(t) h_j(t) = \varepsilon(t). \quad (47)$$

可见,删除神经元时网络输出误差与未删减神经元时相等,网络结构删减并未改变网络收敛性.

4 实验研究(Experimental studies)

4.1 非线性系统辨识 (Nonlinear dynamic system modeling)

选取的典型非线性系统由式(48)表示^[12,15]:

$$y(t+1) = \frac{y(t)y(t-1)[y(t)+2.5]}{1+y^2(t)+y^2(t-1)} + u(t), \quad (48)$$

其中: $y(0) = 0, y(1) = 0, u(t) = \sin(2\pi t/25)$.

非线性系统的模型如下:

$$\hat{y}(t+1) = \hat{f}(y(t), y(t-1), u(t)). \quad (49)$$

根据式(49), 分别构建网络结构为3-6-2-1和3-60-20-1的辨识模型. 根据式(48)产生500个样本, 其中前400个用于训练, 剩余100个用于测试. 实验时, 目标误差 $E_d = 0.001$, 阻尼因子 $\mu = 0.1$, 增长阈值 $I_{gth} = 0.65$, 修剪阈值 $I_{dth} = 0.01$ (3个实验都相同, I_{gth} 和 I_{dth} 通过多次实验选取).

图2给出了训练过程规则层神经元数变化情况, 最终神经元稳定在5个左右, 表明网络能够在线优化规则数, 而且一次可以修剪多个冗余规则, 网络性能得到提高. 图3给出了训练过程均方根误差RMSE变化曲线, 表明网络能够较快收敛.

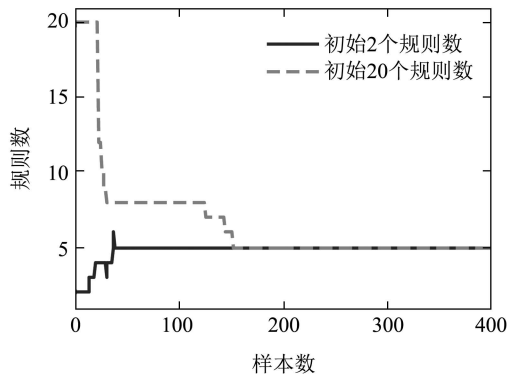


图2 规则层神经元数

Fig. 2 Number variation of normalized neurons

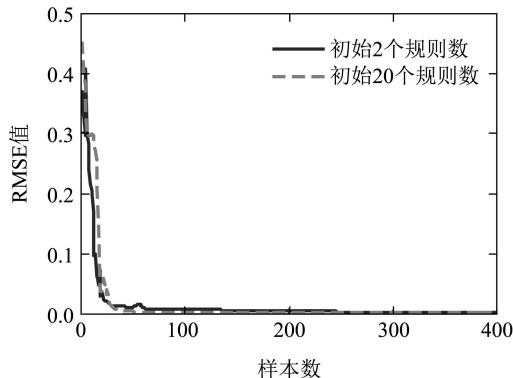


图3 训练RMSE值

Fig. 3 Training RMSE values

预测结果和预测误差分别如图4和图5所示, ILM-SVDFNN与其他算法性能比较如表1所示.

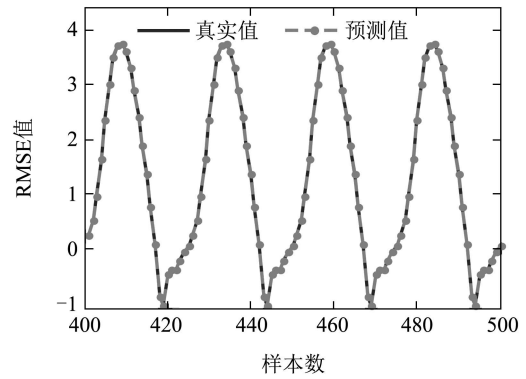


图4 预测结果

Fig. 4 Prediction results

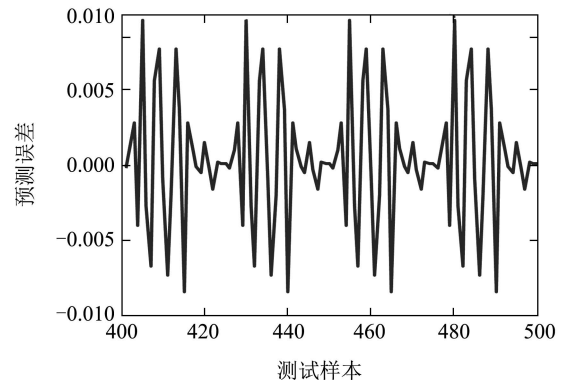


图5 预测误差

Fig. 5 Prediction errors

表1 不同算法性能比较(独立运行30次求平均值)
Table 1 Performance comparison of different algorithms (all results were averaged on 30 independent runs)

算法	2个初始规则层神经元			
	No.	Time/s	RMSE	APE
ILM-SVDFNN	5	6.93	0.0062	0.0107
LM-SVDFNN	5	9.33	0.0063	0.0109
DFNN ^[11]	6	17.23	0.0283	0.0104*
GDFNN ^[12]	8	18.12	0.0108	0.0040*
SOFMLS ^[13]	6	16.22	0.0290	0.0107*
FAOS-PFNN ^[14]	6	7.25	0.0252	0.0093
GP-FNN ^[15]	6	16.14	0.0107	0.0039
SOFNN-ACA ^[16]	6	9.56	0.0105	0.0039
SOFNNGA ^[17]	4	63.31	0.0146	0.0054*

注: No.表示最终规则层神经元数; Time表示运行时间; “*”表示在原文中不存在, 参考文献[15]给出的结果.

性能评价指标分别采用最终规则层神经元数、运行时间、均方根误差 RMSE 和平均百分比误差 APE, 其中RMSE如式(26), APE如式(50)^[15]:

$$APE = \frac{1}{P} \sum_{p=1}^P \frac{|y_d^p(t) - y^p(t)|}{|y_d^p(t)|} \times 100\%. \quad (50)$$

从表1可以看出, ILM-SVDFNN的测试RMSE为0.0062, 明显小于其他算法(APE稍高). 最终规则层神

神经元数为5个,仅多于SOFNNGA算法.改进LM算法的运行时间为6.93 s,比LM算法减少了2.4 s,也普遍快于其他算法.结果表明了ILM-SVDFNN不仅能够获得一个较为精简的网络结构,而且性能较之其他算法也有所提升.

4.2 Mackey-Glass 时间序列预测 (Mackey-Glass time-series prediction)

Mackey-Glass时间序列可由式(51)描述^[11, 15]:

$$x(t+1) = (1-a)x(t) + \frac{bx(t-\tau)}{1+x^{10}(t-\tau)}, \quad (51)$$

其中: $a = 0.1, b = 0.2, \tau = 17, x(0) = 1.2$. 预测问题就是采用嵌入矢量 $\{x(t), x(t-\Delta t), \dots, x(t-(n-1)\Delta t)\}$ 来预测 $x(t+\hat{H})$ 的值.

本文 $\hat{H} = \Delta t = 6, n = 4$, 故预测模型如下:

$$x(t+\hat{H}) = \bar{f}(x(t), x(t-6), x(t-12), x(t-18)). \quad (52)$$

根据式(52),分别构建网络结构为4-8-2-1和4-80-20-1的预测模型.根据式(51)产生1000个样本,其中前500个样本用于训练,剩余500个样本用于测试.

图6给出了训练过程规则层神经元数变化情况,最终神经元数稳定在6个左右,表明能获得一个较为精简的网络结构;图7给出了训练过程均方根误差RMSE变化曲线.

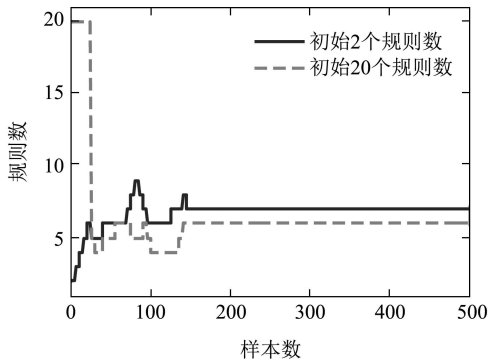


图6 规则层神经元

Fig. 6 Number variation of normalized neurons

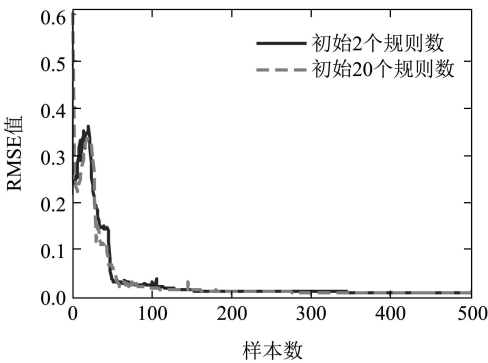


图7 训练RMSE值

Fig. 7 Training RMSE values

预测结果和预测误差分别如图8和图9所示,ILM-SVDFNN与其他算法性能比较如表2所示.

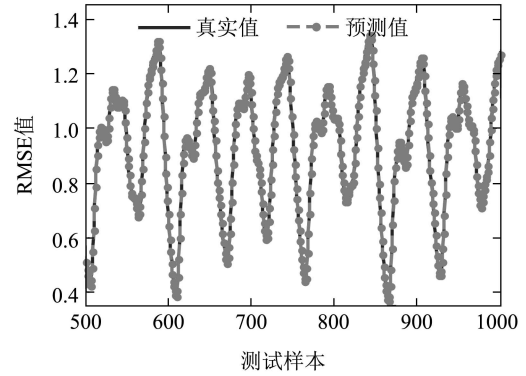


图8 预测结果

Fig. 8 Prediction results

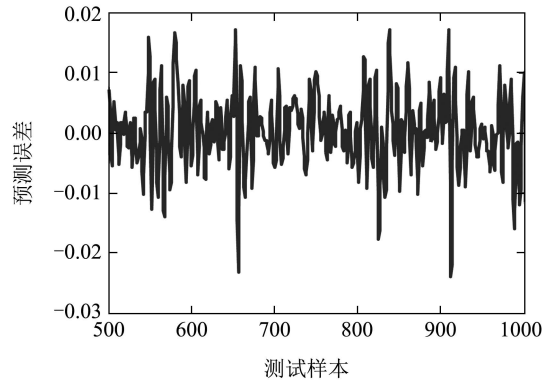


图9 预测误差

Fig. 9 Prediction errors

表2 不同算法性能比较(独立运行30次求平均值)
Table 2 Performance comparison of different algorithms(all results were averaged on 30 independent runs)

算法	2个初始规则层神经元			
	No.	Time/s	RMSE	APE
ILM-SVDFNN	6	15.04	0.0086	0.0072
LM-SVDFNN	6	19.78	0.0086	0.0075
DFNN ^[11]	7	92.23	0.0131	0.0093*
GDFNN ^[12]	11	87.12	0.0118	0.0084*
SOFMLS ^[13]	7	52.35*	0.0471	0.0335*
FAOS-PFNN ^[14]	11	18.18	0.0127	0.0090
GP-FNN ^[15]	9	56.14	0.0107	0.0076
SOFNN-ACA ^[16]	7	27.33	0.0107	0.0076
SOFNNGA ^[17]	7	168.35*	0.0132	0.0094*

注: No. 表示最终规则层神经元数; Time表示运行时间; “*”表示在原文中不存在,参考文献[15]给出的结果.

从表2可以看出,ILM-SVDFNN的最终神经元数为6个,运行时间为15.04 s,测试RMSE和APE分别为0.0086和0.0072,所有指标都优于其他算法.相比LM算法,改进LM算法运行时间减少了4.74 s.结果表明ILM-SVDFNN通过改进LM参数学习算法和网络结构在线自组织调整,网络运算能力和泛化性能得到极大提高.

4.3 污水处理出水总磷预测 (Predicting effluent total phosphorus in WWTP)

大量的磷排入受纳水体, 将使得藻类生物大量繁殖, 从而导致水体富营养化且极易造成严重的生态后果, 因此国家在《城镇污水处理厂污染物排放标准 (GB18918-2002)》中对磷的排放有严格的标准. 随着排放标准的提高, 为避免超标引起罚款, 污水处理厂对总磷检测精度的要求也将会越来越高. 因此本文利用ILM-SVDFNN构建出水总磷预测模型, 以期满足总磷检测需求.

污水处理生化反应过程关键可测变量及其意义如表3所示. 从北京市某小型污水处理厂获取2015年6-8月共492组数据, 首先采用小波包(sym8小波、2层分解、软阈值方式)进行降噪处理, 处理效果如图10所示. 每隔3个样本选取1个样本共123个样本组成测试集, 其余369个样本作为训练集, 构建9-18-2-1的预测模型.

表 3 可测变量及意义

Table 3 Measurable variables and significance

参数	意义	单位	取样点
T	水温	$^{\circ}\text{C}$	厌氧池
ORP_1	氧化还原电位	mV	厌氧池
DO_1	溶解氧浓度	mg/L	好氧前端
DO_2	溶解氧浓度	mg/L	好氧末端
TSS	总固体悬浮物浓度	g/L	好氧末端
pH	酸碱度	—	出水端
ORP_2	氧化还原电位	mV	出水端
$\text{NH}_4\text{-N}$	铵态氮浓度	mg/L	出水端
$\text{NO}_3\text{-N}$	硝态氮浓度	mg/L	出水端

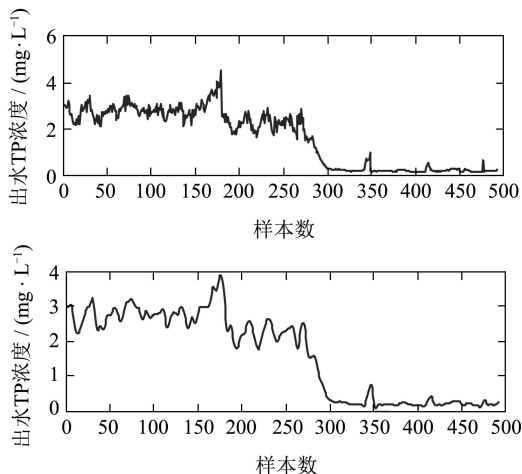


图 10 总磷数据小波包去噪

Fig. 10 Wavelet packet denoising of TP

图11给出了训练过程规则层神经元数变化情况, 最终神经元数稳定在10个左右; 图12和图13分别给出了测试样本的预测结果和预测误差; 表4给出了ILM-SVDFNN与LM-SVDFNN, ILM-FNN, EBP-FNN以及其他自组织FNN的结果比较.

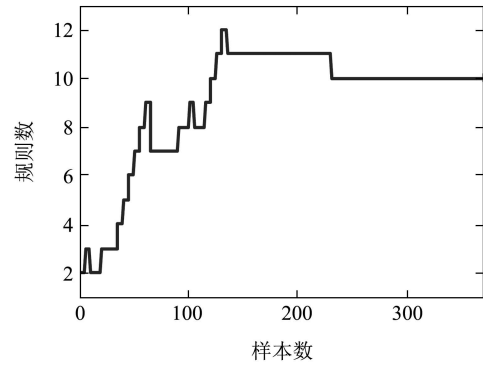


图 11 规则层神经元数

Fig. 11 Number variation of normalized neurons

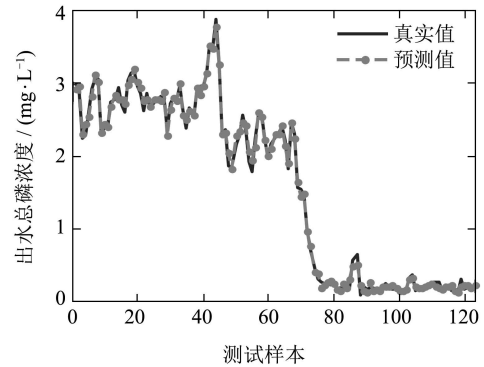


图 12 预测结果

Fig. 12 Prediction results

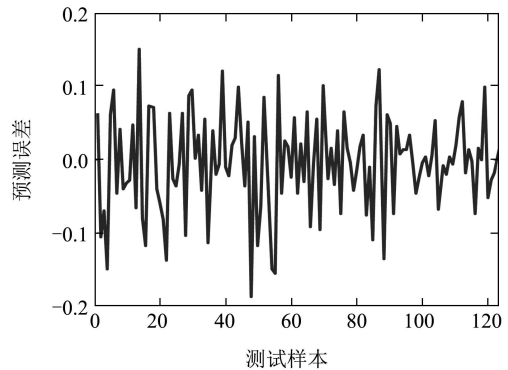


图 13 预测误差

Fig. 13 Prediction errors

从图11可以看出, ILM-SVDFNN在修剪阶段能够删减冗余神经元, 并且一次可以删减多个冗余神经元, 较之一般的自组织算法^[11-14]一次只能删减一个神经元更加快速有效. 此外增长修剪指标的定义是根据规则层输出阵的奇异值分解, 神经元的增长和修剪并不是取决于当前样本, 而是一段时间内样本的分布情况, 较之其他自组织算法^[11-12]以当前值作为判断依据更加客观. 从图12和图13可以看出, 污水处理出水总磷的预测精度较高, 预测误差主要集中在 ± 0.1 mg/L之间, 远小于实际污水处理厂总磷检测允许误差 ± 0.5 mg/L.

从表4可以看出, ILM-SVDFNN的测试RMSE和APE分别为0.1061和0.1161, 不仅小于EBP-FNN,

而且与其他算法相比也有一定程度提高. ILM-SVDFNN 的运行时间为 22.15 s, 比 ILM-FNN 减少了 12.23 s, 比 LM-SVDFNN 算法减少了近一倍. 这主要是因为该自组织模糊神经网络初始时模糊规则数较少, 计算速度较快, 另外总磷预测数据的输入量个数和训练样本数相对较多, 更能发挥改进 LM 算法的快速在线建模优势. 结果表明本文构建的 ILM-SVDFNN 预测模型较好地满足了出水总磷在线快速精确预测的需求, 有助于及时发现并采取措施控制水体富营养化.

表 4 不同算法性能比较(独立运行30次求平均值)
Table 4 Performance comparison of different algorithms(all results were averaged on 30 independent runs)

算法	No.	Time/s	RMSE	APE
ILM-SVDFNN	10	22.15	0.1061	0.1161
LM-SVDFNN	10	40.05	0.1078	0.1167
ILM-FNN	10 ^a	34.38	0.1296	0.1412
EBP-SVDFNN	14	85.76	0.1766	0.1690
EBP-FNN	10 ^a	114.37	0.1944	0.1840
DFNN	12	56.87	0.1763	0.1644
GDFNN	9	46.83	0.1539	0.1628

注: No. 表示最终规则层神经元数; Time 表示运行时间;

EBP 表示采用误差反向传播算法;

^a 表示取固定数目神经元.

5 结论(Conclusions)

本文基于改进 LM 算法和 SVD 构建了一种在线自组织模糊神经网络, 并应用于典型非线性系统建模和污水处理过程出水总磷预测. 改进 LM 算法收敛速度快, 计算量和存储空间小, 易于寻找到全局最优值. 通过利用单边 Jacobi 变换实现 SVD, 可以确定奇异值与规则层神经元之间的对应关系, 结合增长和修剪指标, 从而能够实现网络结构在线自组织调整. 出水总磷预测、非线性系统建模以及 Mackey-Glass 时间序列预测实验的仿真结果验证了所构建的 ILM-SVDFNN 不仅能够获得一个较精简的网络结构, 而且网络建模速度、预测精度和泛化能力都有较大程度提高, 对污水处理过程总磷参数的实时检测具有一定的现实意义. 需要指出的是, 针对本文小样本实验, 神经元动态调整的时间间隔可以取较小的整数, 以充分挖掘样本自身分布式信息. 然而对于大样本实验, 为了减少运行时间, 需要取较大的间隔步数, 这可能会导致有用信息的丢失和模型泛化能力的降低, 需要进一步研究解决.

参考文献(References):

- [1] DUPAS R, DELMAS M, DORIOZ J M, et al. Assessing the impact of agricultural pressures on N and P loads and eutrophication risk [J]. *Ecological Indicators*, 2015, 48(1): 396 – 407.
- [2] JIN L Y, ZHANG G M, TIAN H F. Current state of sewage treatment in China [J]. *Water Research*, 2014, 66(12): 85 – 98.

- [3] DÜRRENMATT D J, GUJER W. Data-driven modeling approaches to support wastewater treatment plant operation [J]. *Environmental Modelling & Software*, 2012, 30(4): 47 – 56.
- [4] BAGHERI M, MIRBAGHERI S A, EHTESHAMI M, et al. Modeling of a sequencing batch reactor treating municipal wastewater using multi-layer perceptron and radial basis function artificial neural networks [J]. *Process Safety and Environmental Protection*, 2015, 93(1): 111 – 123.
- [5] HEDDAM S, LAMDA H, FILALI S. Predicting effluent biochemical oxygen demand in a wastewater treatment plant using generalized regression neural network based approach: a comparative study [J]. *Environmental Processes*, 2016, 3(1): 153 – 165.
- [6] QIAO J F, HAN G, HAN H G. Neural network on-line modeling and controlling method for multi-variable control of wastewater treatment processes [J]. *Asian Journal of Control*, 2014, 16(4): 1213 – 1223.
- [7] PIZZILEO B, LI K, IRWIN G W, et al. Improved structure optimization for fuzzy-neural networks [J]. *IEEE Transactions on Fuzzy Systems*, 2012, 20(6): 1076 – 1089.
- [8] WILAMOWSKI B M, YU H. Improved computation for Levenberg-Marquardt training [J]. *IEEE Transactions on Neural Networks*, 2010, 21(6): 930 – 937.
- [9] XIE T, YU H, HEWLETT J, et al. Fast and efficient second-order method for training radial basis function networks [J]. *IEEE Transactions on Neural Networks and Learning Systems*, 2012, 23(4): 609 – 619.
- [10] QIAO Junfei, HAN Honggui. Dynamic optimization structure design for neural networks: review and perspective [J]. *Control Theory & Applications*, 2010, 27(3): 350 – 357.
(乔俊飞, 韩红桂. 神经网络结构动态优化设计的分析与展望 [J]. 控制理论与应用, 2010, 27(3): 350 – 357.)
- [11] WU S Q, ER M J. Dynamic fuzzy neural networks—a novel approach to function approximation [J]. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 2000, 30(2): 358 – 364.
- [12] WU S Q, ER M J, Gao Y. A fast approach for automatic generation of fuzzy rules by generalized dynamic fuzzy neural networks [J]. *IEEE Transactions on Fuzzy Systems*, 2001, 9(4): 578 – 594.
- [13] DE JESÚS RUBIO J. SOFMLS: online self-organizing fuzzy modified least-squares network [J]. *IEEE Transactions on Fuzzy Systems*, 2009, 17(6): 1296 – 1309.
- [14] WANG N, ER M J, MENG X Y. A fast and accurate online self-organizing scheme for parsimonious fuzzy neural networks [J]. *Neurocomputing*, 2009, 72(16): 3818 – 3829.
- [15] HAN H G, QIAO J F. A self-organizing fuzzy neural network based on a growing-and-pruning algorithm [J]. *IEEE Transactions on Fuzzy Systems*, 2010, 18(6): 1129 – 1143.
- [16] HAN H G, WU X L, QIAO J F. Nonlinear systems modeling based on self-organizing fuzzy-neural-network with adaptive computation algorithm [J]. *IEEE Transactions on Cybernetics*, 2014, 44(4): 554 – 564.
- [17] LENG G, MCGINNITY T M, PRASAD G. Design for self-organizing fuzzy neural networks based on genetic algorithms [J]. *IEEE Transactions on Fuzzy Systems*, 2006, 14(6): 755 – 766.
- [18] LIU J, LIU X Q, MA X L. First-order perturbation analysis of singular vectors in singular value decomposition [J]. *IEEE Transactions on Signal Processing*, 2008, 56(7): 3044 – 3049.
- [19] DRMAC Z, VESELIC K. New fast and accurate Jacobi SVD algorithm, I [J]. *SIAM Journal on Matrix Analysis and Applications*, 2008, 29(4): 1322 – 1342.

作者简介:

乔俊飞 (1968–), 男, 教授, 博士生导师, 目前研究方向为污水处理过程知识自动化, E-mail: junfeiq@bjut.edu.cn;

周红标 (1980–), 男, 博士研究生, 目前研究方向为神经网络分析与设计, E-mail: hyitzhb@163.com.