

引用格式:杨彦霞,王普,高学金,等. 基于混合双层自组织径向基函数神经网络的优化学习算法[J]. 北京工业大学学报, 2024, 50(1): 38-49.

YANG Y X, WANG P, GAO X J, et al. Optimization learning algorithm based on hybrid bilevel self-organizing radial basis function neural network[J]. Journal of Beijing University of Technology, 2024, 50(1): 38-49. (in Chinese)

基于混合双层自组织径向基函数神经网络的 优化学习算法

杨彦霞^{1,2}, 王普^{1,2}, 高学金^{1,2}, 高慧慧^{1,2}, 齐泽洋^{1,2}

(1. 北京工业大学信息学部, 北京 100124; 2. 北京工业大学计算智能与智能系统北京市重点实验室, 北京 100124)

摘要: 针对传统方法采用先训练后测试两阶段学习机制极易导致的过拟合或欠拟合问题, 提出一种基于混合双层自组织径向基函数神经网络的优化学习 (hybrid bilevel self-organizing radial basis function neural network optimization learning, Hb-SRBFNN-OL) 算法。首先, 将训练过程和测试过程集成到一个统一的框架中, 规避过拟合或欠拟合问题。其次, 基于进化学习机制, 提出上下2层的交互式优化学习算法, 上层基于网络复杂度和测试误差自组织调整网络结构, 下层采用列文伯格-马夸尔特 (Levenberg Marquardt, LM) 算法作为优化器对自组织径向基函数神经网络 (self-organizing radial basis function neural network, SO-RBFNN) 的连接权值进行优化。最后, 利用来自多个子网络的综合信息生成模型的最终输出, 加速网络全局收敛。为验证所提方法的可行性, 分别在多个分类和预测任务中进行了测试实验。结果表明, 在与传统神经网络结构相似甚至更好的测试和分类精度下, 该方法不仅能实现更快的训练收敛, 而且能进化成更精简紧凑的径向基函数神经网络 (radial basis function neural network, RBFNN) 模型。尤其在污水处理过程中总磷的质量浓度预测实验中, 测试集中均方根误差 (root mean squared error, RMSE) 最高可降低 48.90%, 实际场景实验结果验证了所提算法的精确性更佳且泛化能力更强。

关键词: 径向基函数神经网络 (radial basis function neural network, RBFNN); 自组织; 列文伯格-马夸尔特 (Levenberg Marquardt, LM) 算法; 混合双层; 优化学习; 泛化性能

中图分类号: TP 273

文献标志码: A

文章编号: 0254-0037(2024)01-0038-12

doi: 10.11936/bjutxb2022020006

Optimization Learning Algorithm Based on Hybrid Bilevel Self-organizing Radial Basis Function Neural Network

YANG Yanxia^{1,2}, WANG Pu^{1,2}, GAO Xuejin^{1,2}, GAO Huihui^{1,2}, QI Zeyang^{1,2}

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

2. Beijing Key Laboratory of Computational Intelligence and Intelligent System,
Beijing University of Technology, Beijing 100124, China)

Abstract: Traditional methods adopt the two-stage learning mechanism of training before testing, which can easily lead to over fitting or under fitting. To solve this problem, an optimization learning algorithm

收稿日期: 2022-02-19; 修回日期: 2022-04-24

基金项目: 国家自然科学基金资助项目(61640312, 61803005); 北京市自然科学基金资助项目(4172007, 4192011)

作者简介: 杨彦霞(1990—), 女, 博士研究生, 主要从事工业过程关键变量预测、故障监测方面的研究, E-mail: 2272103579@qq.com

通信作者: 高学金(1973—), 男, 教授, 主要从事工程过程状态监测与故障诊断方面的研究, E-mail: gaoxuejin@bjut.edu.cn

based on hybrid bilevel self-organizing radial basis function neural network (Hb-SRBFNN-OL) was proposed. First, the training process and testing process were integrated into a unified framework to effectively balance the overfitting and underfitting problems. Second, an interactive learning algorithm with two layers was proposed. The upper layer organized and adjusted the network structure based on the network complexity and test error, and the lower layer used Levenberg Marquardt (LM) algorithm as the optimizer to optimize the connection weight of self-organizing radial basis function neural network (SO-RBFNN). Finally, the final output of the model was generated by using the linear combination information from multiple SO-RBFNNs to accelerate the global convergence of the network. To verify the feasibility of the proposed method in practical problems, and test experiments in multiple classification and prediction tasks were conducted, respectively. Results show that this method can not only achieve faster training convergence, but also produce a more concise and compact radial basis function neural network (RBFNN) model. In particular, in the prediction experiment of total phosphorus concentration in wastewater treatment process, the root mean squared error (RMSE) of the test set is reduced by up to 48.90%, achieving better accuracy and stronger generalization ability.

Key words: radial basis function neural network (RBFNN); self-organizing; Levenberg Marquardt (LM) algorithm; hybrid bilevel; optimization learning; generalization performance

神经网络技术具有大规模并行处理能力、自适应能力和高灵活性等优点,已成为机器学习和数据处理中非常受欢迎的工具。3层的径向基函数神经网络(radial basis function neural network, RBFNN)和反向传播(back propagation, BP)学习算法可以用足够多的神经元逼近任意非线性连续函数且达到任意精度,在现实任务中得到了广泛的应用^[1-2]。在使用RBFNN模型来处理实际问题时,该模型通常包括选择合适的网络模型结构以及高效的学习算法^[3]。

RBFNN的结构可由网络层数、每层神经元个数、网络的所有连接以及神经元的传递函数表示。众所周知,RBFNN的性能对神经元的数量极其敏感,神经元数量过少会降低网络的逼近能力,而神经元数量过多又可能导致过拟合,因此,实现更好的网络性能和简化网络结构相互矛盾。多年实践证明,RBFNN结构的训练是一项具有挑战性的任务,而学习算法是RBFNN的核心。常用的学习算法是基于导数的优化算法^[4-6],如梯度下降(gradient descent, GD)方法、随机梯度下降(stochastic gradient descent, SGD)方法、共轭梯度(conjugate gradient, CG)方法等。列文伯格-马夸尔特(Levenberg Marquardt, LM)算法作为介于牛顿法与GD方法之间的一种非线性优化方法^[7-8],对过参数化问题不敏感,并能有效处理冗余参数问题,使代价函数陷入局部极小值的机会大大减小,在计算机视

觉、分类、控制等领域得到了广泛应用。RBFNN的学习过程是基于实际输出和期望输出之间的直接比较,通过迭代调整连接权值实现的,这种学习过程属于监督学习。其中,一种最常用的算法是基于GD的BP算法,即通过迭代调整权值,使误差最小化。

迄今为止,传统的RBFNN学习模型包括单目标优化模型和多目标优化模型。单目标优化模型是RBFNN训练中最常用、研究最广泛的模型,它只包含一个基于网络误差的目标。当网络结构固定时,RBFNN的权值只能通过训练算法进行优化,目标是使网络误差最小化^[9-11]。当同时优化网络结构和连接权值时,目标是使网络误差最小或网络误差和网络复杂度的组合最小^[12-13]。需要注意的是,多目标优化模型通常包含2个相互矛盾的目标,即最小化网络误差和最小化隐藏单元数^[14-15],并且多目标模型的目的是实现精度和网络结构之间的最优权衡。Goh等^[16]设计了一个多目标问题,同时,考虑了分类准确性和网络复杂性这2个相互冲突的目标。在文献[17]中,Almeida等构造了基于训练误差、结构复杂度和传递函数复杂度的双目标优化问题。其中:一个目标是测试误差、训练误差、隐藏层的数量、隐藏节点的数量、传递函数权重5项信息的组合;另一个目标是基于网络误差的测量。Loghmanian等^[18]同样提出了多目标策略,即最小化均方误差和由输入神经元个数、输出神经元个数、隐藏层神经个

数组成的网络复杂度。神经网络训练的另一个常见困难是神经网络中的非线性激活函数使模型收敛到局部最优解而不是全局最优解,模型性能不能达到最优。为解决此问题,提高训练效果,近几十年来多位学者做了大量的工作,例如,将激活函数更改为线性函数,并保持输入和输出的方差近似,利用 Xavier 初始化网络权值^[19]。然而,这类解决方案高度依赖于数据集的数据分布。之后,也有学者提出更改网络结构,即非常著名的长短时记忆(long short term memory, LSTM)神经网络,通过使用多门结构控制状态存储器的流量,有效解决梯度消失^[20-21]问题。将激活函数更改为整流线性单元(rectified linear unit, ReLU)函数也有助于解决消失的梯度问题^[22-23],然而当输入为负数时,神经元呈死亡状态。简而言之,输出要么为0,要么为正,这极大限制了其实际应用。

1 问题描述

现有学习模型大都采用先通过训练模型架构和参数得到最优神经网络模型,再测试网络性能的思路。此种机制将训练过程和测试过程分裂成彼此独立的个体,前者不知后者,而后者亦看不到前者,在实际应用时,由此种机制导致的过拟合和欠拟合很难平衡。当一个模型开始记忆训练数据并以突触权重的形式存储知识,而不是从中学习归纳趋势或规则时,极易引起过拟合或欠拟合问题,从而导致泛化性能较差。

为了设计一种紧凑且具有良好泛化能力的 RBFNN 体系结构,本文构建了一种以网络结构为领导者、以连接权值为跟随者的双层进化学习模型,即基于混合双层自组织径向基函数神经网络的优化学习(hybrid bilevel self-organizing radial basis function neural network optimization learning, Hb-SRBFNN-OL)算法。该方法是一个线性叠加多层神经网络的双层进化学习系统,包括与网络复杂度和测试误差相关的上层结构优化部分,以及与训练误差相关的下层参数优化部分。其优势如下:

1) 不同于以往只使用元启发式算法来优化 RBFNN 的结构和权值,该算法兼顾了训练误差和测试误差。上层优化器用于优化网络架构,减少测试误差,下层优化器通过最小化训练误差优化给定的 RBFNN 权重。

2) 通过一个全局损失函数绕过消失梯度问题

和减少非凸优化的负面影响,在训练速度和准确性方面提高模型性能。

3) 将测试过程结果反馈给训练过程,通过持续交互、进化学习,有效解决了过拟合和欠拟合问题。

2 Hb-SRBFNN-OL 算法

2.1 混合双层优化学习模型结构

为有效平衡神经网络训练中的过拟合和欠拟合,提出一种组合多层神经网络学习框架,如图1所示。

结构调整模块表示上层优化器,基于测试误差和网络复杂度优化网络结构;参数优化模块表示下层优化器,基于训练误差优化网络权值。 \mathbf{X}_{te} 和 \mathbf{X}_{tr} 分别表示测试输入和训练输入数据, \hat{y}_{te} 和 \hat{y}_{tr} 分别为它们对应的预测输出。RBFNN 子网络具有相同的结构、不同的初始权值。

参数优化模块,即传统的权值学习算法通过 BP 输出误差 $e_{i,k}(t) = \hat{y}_k(t) - y_k(t)$ 更新第 k 个子网络中的所有参数,使预测输出 $\hat{y}_k(t)$ 在 $t \rightarrow \infty$ 时逐渐接近其真标签 $y_k(t)$ 。

在本文所设计的结构中,不再使用每个 $\hat{y}_k(t)$ 作为网络预测值,而是使用 k 个预测模型输出的线性组合作为 $\hat{y}(t)$ 在每个时间步 t 的估计,即

$$\hat{y}(t) = \sum_{i=1}^k \hat{\alpha}_i(t) \hat{y}_i(t) \quad (1)$$

$$\hat{y}_i(t) = \sum_{j=1}^J w_j(t) \varphi_j(t) \quad (2)$$

$$\varphi_j(t) = e^{-\|\mathbf{X}(t) - \mathbf{C}_j(t)\|^2 / (2\sigma_j(t)^2)} \quad (3)$$

式中: $w_j(t)$ 为 t 时刻第 j 个隐藏层神经元与输出之间的权重; $\varphi_j(t)$ 为第 j 个隐藏层神经元在 t 时刻的输出; $\mathbf{C}_j(t) = [c_{j1}(t), c_{j2}(t), \dots, c_{jI}(t)]$; $j=1, 2, \dots, J$ 为第 j 个隐藏层神经元在 t 时刻的中心向量; $\mathbf{X}(t) = [x_1(t), x_2(t), \dots, x_I(t)]^T$ 为 RBFNN 在 t 时刻的 I 维输入向量; $\|\cdot\|$ 表示欧氏距离; $\sigma_j(t)$ 为第 j 个隐藏层神经元在 t 时刻的宽度; $\hat{\alpha}_i(t)$ 为第 i 个子网络在时间步 t 的组合系数,目标系数 $\alpha_i(t)$ 满足准则

$$\sum_{i=1}^k \alpha_i(t), 0 \leq \alpha_i(t) \leq 1 \quad (4)$$

在任意时刻,当前系数的估计值满足

$$\sum_{i=1}^k \hat{\alpha}_i(t), 0 \leq \hat{\alpha}_i(t) \leq 1, \forall t \in T \quad (5)$$

2.2 上层目标:网络结构优化

对于一个典型的3层 RBFNN,输入神经元的个

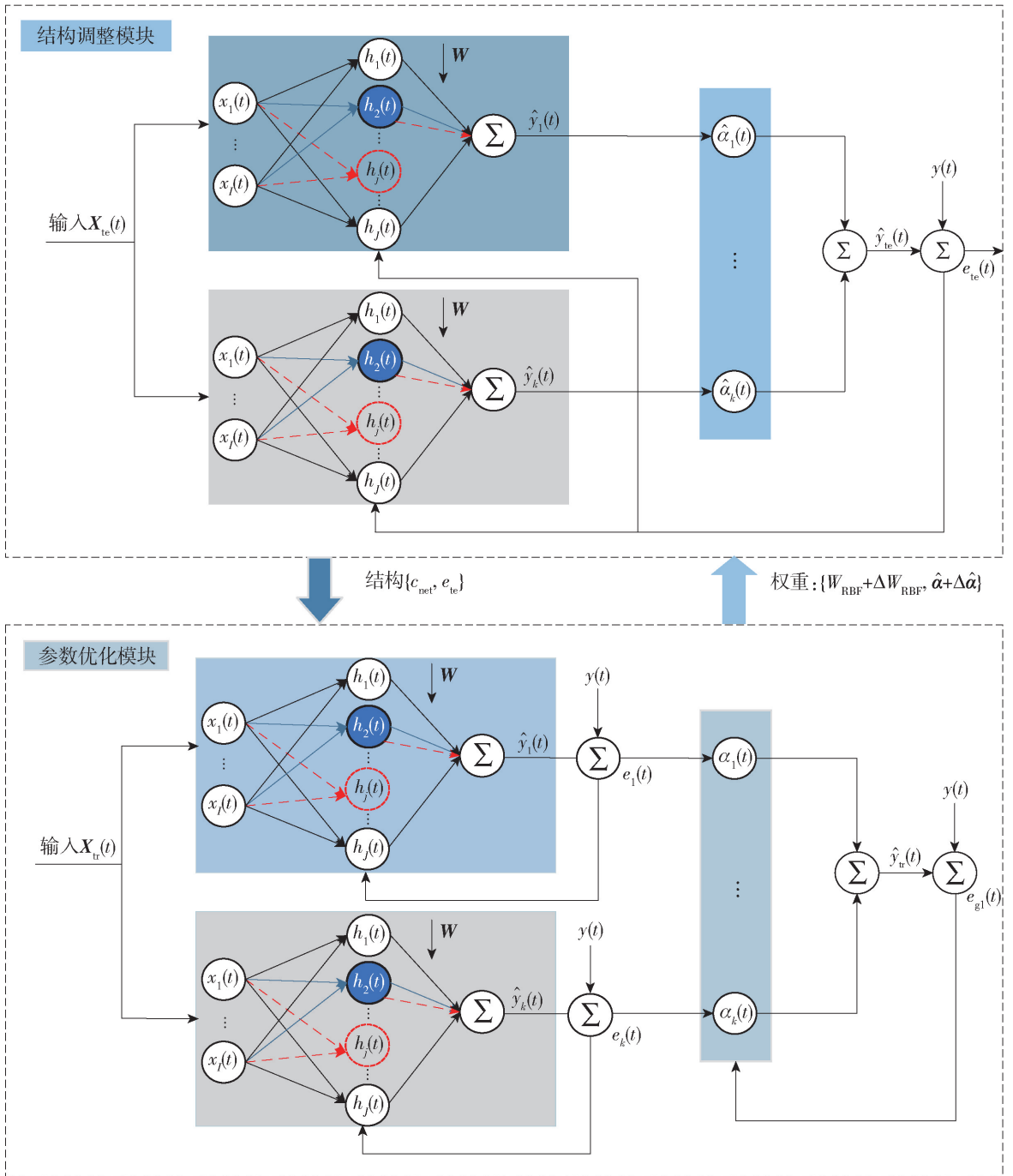


图1 Hb-SRBFNN-OL 框架
Fig. 1 Framework of Hb-SRBFNN-OL

数和输出神经元的个数分别取决于训练模式和训练样本,而核心问题在于优化过程获得最优数量的隐藏层神经元个数。因此,一个3层的多输入单输出RBFNN结构可以表示为 $I-J-1$ 。 I 表示输入维度, J 表示隐藏层神经元个数。假设最大隐藏层神经元数量为 J_{max} , 隐藏层结构表示为 $\mathbf{h} = [h_1, h_2, \dots, h_{J_{max}}]$, $h_{J_{max}}$ 表示第 J_{max} 个隐藏层神经元的状态。

RBFNN)的拓扑结构见图2。对于保留神经元,它与输入层的每个神经元和输出层的每个神经元都有连接。蓝色神经元表示新添加神经元,而红色虚线神经元为删除神经元,即断开网络中该神经元的所有连接。网络结构的自适应调整过程即本文所提模型框架的上层网络优化过程。

上层目标主要考虑2个指标:网络复杂度 c_{net} 和测试误差 e_{te} ,并分别优化这2个指标。基于此,

自组织 RBFNN (self-organizing RBFNN, SO-

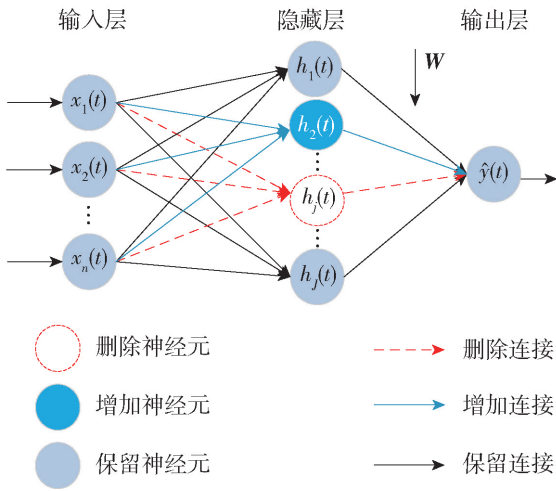


图2 SO-RBFNN 的拓扑结构

Fig. 2 Topology of SO-RBFNN

对网络结构进行自适应调整,目的是获得紧凑且泛化能力良好的 RBFNN 结构。上层目标函数定义为

$$L_u = c_1 \times c_{\text{net}} + c_2 \times e_{\text{te}} \quad (6)$$

式中: $c_1, c_2 \in (0, 1)$ 表示权重系数; c_{net} 用来描述网络拓扑结构。网络复杂度的值越小,说明网络越简单、越紧凑。 c_{net} 定义为

$$c_{\text{net}} = \frac{A_c}{T_c} \quad (7)$$

式中: A_c 表示上层 SO-RBFNN 得到的实际网络连接数; T_c 表示包括所有输入神经元、隐藏层神经元和输出神经元的网络连接总数。

对于具有最大隐藏层神经元个数 J_{max} 的 3 层 RBFNN 结构, T_c 和 A_c 的计算方式分别为

$$T_c = I \times J_{\text{max}} + J_{\text{max}} \times 1 = (I + 1) \times J_{\text{max}} \quad (8)$$

$$A_c = I \times J + J \times 1 = (I + 1) \times J \quad (9)$$

在获得紧凑且泛化能力良好的 RBFNN 体系结构后,基于目标函数对网络结构进行自调整,调整规则为

$$J(W) = \begin{bmatrix} \frac{\partial e_k(t)}{\partial c_{1,1}} & \frac{\partial e_k(t)}{\partial c_{1,2}} & \dots & \frac{\partial e_k(t)}{\partial c_{1,J}} & \frac{\partial e_k(t)}{\partial \sigma_1} & \frac{\partial e_k(t)}{\partial w_1} & \frac{\partial e_k(t)}{\partial c_{2,1}} & \frac{\partial e_k(t)}{\partial c_{2,2}} & \dots & \frac{\partial e_k(t)}{\partial c_{2,J}} & \frac{\partial e_k(t)}{\partial \sigma_2} & \frac{\partial e_k(t)}{\partial w_2} & \frac{\partial e_k(t)}{\partial c_{J,1}} & \frac{\partial e_k(t)}{\partial c_{J,2}} & \dots & \frac{\partial e_k(t)}{\partial c_{J,I}} & \frac{\partial e_k(t)}{\partial \sigma_J} & \frac{\partial e_k(t)}{\partial w_J} \end{bmatrix} \quad (14)$$

式中: $e_k(t)$ 表示第 k 个子网络在 t 时刻的网络误差; $c_{J,1}$ 为第 J 个神经元中心的第 1 个元素,并且

$$H(W) = \sum_{p=1}^P e_{k,p} \nabla^2 e_{k,p} \quad (15)$$

假设 $H(W)$ 很小,则 Hessian 矩阵近似为

$$J_{t+1} = \begin{cases} J+1, & 0 < c_{\text{net}} \leq \varepsilon_1 \\ J, & \varepsilon_1 < c_{\text{net}} \leq \varepsilon_2 \\ J-1, & \varepsilon_2 < c_{\text{net}} \leq 1 \end{cases} \quad (10)$$

式中 ε_1 和 ε_2 表示阈值因子, ε_1 设定为 0.3, ε_2 设定为 0.7。

2.3 下层目标:网络权值优化

SO-RBFNN 训练的主要困难是获取最优的一组参数,即网络的中心、宽度、输出层与隐藏层之间的权向量,统一记为 W 。神经元之间的信息是单向传递的,而信息强度主要依靠神经元之间的连接权值,即实数加权来表示,因此,将网络中所有的连接权向量 W 视为较低层次的决策变量,表达形式如图 2 所示。算法的下层目标是对给定网络结构的 RBFNN 权值进行优化,为了保证算法的快速收敛,本文提出了基于局部优化和全局优化的独立优化机制。

2.3.1 局部网络权值优化

局部优化的任务是训练 RBF 子网络的连接权值,在这里,选择 LM 作为底层局部优化器,原因是它是一种优秀的基于导数的方法,收敛速度快,稳定性好。对给定的网络架构 h ,设置第 k 个子网络目标函数为

$$L_k = \frac{1}{2} \sum_{p=1}^P \|y_{k,p} - \hat{y}_{k,p}\|^2 \quad (11)$$

式中: P 为训练集中样本个数; $\hat{y}_{k,p}$ 为第 k 个子网络中第 p 个样本的网络输出; $y_{k,p}$ 为相应的目标输出。最小化目标函数,即

$$\Delta W = [\nabla^2 E(W)]^{-1} \cdot \nabla E(W) \quad (12)$$

式中: $W = [w_1, w_2, \dots, w_J]$ 表示所有权值的向量; $\nabla E(W)$ 表示权值的梯度; $\nabla^2 E(W)$ 为 Hessian 矩阵。 $\nabla E(W)$ 和 $\nabla^2 E(W)$ 的公式分别为

$$\begin{cases} \nabla E(W) = J^T(W)E(W) \\ \nabla^2 E(W) = J^T(W)J(W) + H(W) \end{cases} \quad (13)$$

$J(W)$ 为雅可比矩阵,公式为

$$\nabla^2 E(W) \approx J^T(t)J(t) \quad (16)$$

将式(13) (16)代入式(12),得到

$$\Delta W = [J^T(t)J(t)]^{-1} J^T(t)e_k(t) \quad (17)$$

$J^T(t)J(t)$ 可能是不可逆的,因此,使用近似 Hessian 矩阵

$$\Delta \mathbf{W} = [\mathbf{J}^T(t)\mathbf{J}(t) + \mu \mathbf{I}]^{-1} \mathbf{J}^T(t) e_k(t) \quad (18)$$

式中: \mathbf{I} 为单位矩阵; μ 表示一个大于 0 的参数。 μ 较大时, 算法变为最陡下降法, 而 μ 较小时, 算法变为高斯牛顿法。在本文中, 初始化 μ 为 0.01。

2.3.2 全局组合系数优化

全局损失函数 e_{gl} 定义为真实输出与预测输出之间的差值, 公式为

$$e_{gl}(t) = y(t) - \hat{y}_{tr}(t) \quad (19)$$

将式(3)代入式(19), 可以将全局损失函数改写为

$$\begin{aligned} e_{gl}(t) = & y(t) - \hat{y}_{tr}(t) = \\ & y(t) - [\hat{\alpha}_1(t)\hat{y}_1(t) + \hat{\alpha}_2(t)\hat{y}_2(t) + \dots + \\ & \hat{\alpha}_k(t)\hat{y}_k(t)] = \\ & \sum_{i=1}^{k-1} \hat{\alpha}_i(t)e_i(t) + \hat{\alpha}_k(t)e_k(t) \end{aligned} \quad (20)$$

式中 $e_i(t)$ 定义为第 i 个子网络预测输出与目标输出的差值, 即

$$e_i(t) = y_i(t) - \hat{y}_i(t) \quad (21)$$

则

$$\begin{aligned} e_{gl}(t) = & \sum_{i=1}^{k-1} \hat{\alpha}_i(t)e_i(t) + \left(1 - \sum_{i=1}^{k-1} \hat{\alpha}_i(t)\right)e_k(t) = \\ & \sum_{i=1}^{k-1} \hat{\alpha}_i(t)\hat{e}_i(t) + e_k(t) \end{aligned} \quad (22)$$

式中 $\hat{e}_i(t)$ 表示第 i 个模型的误差和第 k 个模型误差的差值, 定义为

$$\hat{e}_i(t) = e_i(t) - e_k(t) \quad (23)$$

由式(20)推导出的损失函数可进一步简化为

$$\tilde{e}_{gl}(t) = \hat{\mathbf{e}}^T(t)\hat{\boldsymbol{\alpha}}(t) \quad (24)$$

式中: $\tilde{e}_{gl}(t) = e_{gl}(t) - e_k(t)$; $\hat{\mathbf{e}}(t)$ 和 $\hat{\boldsymbol{\alpha}}(t)$ 分别定义为

$$\begin{cases} \hat{\mathbf{e}}(t) = [\hat{e}_{1,1}(t), \hat{e}_{1,2}(t), \dots, \hat{e}_{l,k-1}(t)]^T \\ \hat{\boldsymbol{\alpha}}(t) = [\hat{\alpha}_1(t), \hat{\alpha}_2(t), \dots, \hat{\alpha}_{k-1}(t)]^T \end{cases} \quad (25)$$

在式(24)的两边乘以 $\hat{\mathbf{e}}(t)$, 得到

$$\hat{\mathbf{e}}(t)\tilde{e}_{gl}(t) = \hat{\mathbf{e}}(t)\hat{\mathbf{e}}^T(t)\hat{\boldsymbol{\alpha}}(t) \quad (26)$$

将方程左边的元素移到右边, 令 $\Delta\hat{\boldsymbol{\alpha}}(t) = \hat{\boldsymbol{\alpha}}(t+1) - \hat{\boldsymbol{\alpha}}(t)$, 得到线性系数的变化量

$$\Delta\hat{\boldsymbol{\alpha}}(t) = -\hat{\mathbf{e}}(t)\hat{\mathbf{e}}^T(t)\hat{\boldsymbol{\alpha}}(t) + \hat{\mathbf{e}}(t)\tilde{e}_{gl}(t) \quad (27)$$

注1 全局叠加系数 $\hat{\boldsymbol{\alpha}}$ 的更新是一个线性差分方程, 全局损失通过从多个子网络中收集信息提高收敛速度和性能。

注2 通过定义并更新全局损失函数, 系统无须等待每个子网络在系统收敛之前收敛, 有效避免了梯度消失问题。

注3 将训练过程和测试过程集成到一个系统中, 上层框架用来自调整网络结构, 降低网络复杂度, 下层框架用来优化子网络的连接权值和线性组合的系数。通过持续交互、进化学习, 有效解决了过拟合和欠拟合问题。

3 实验结果与分析

为了证明所提 Hb-SRBFNN-OL 的可行性, 本文给出了多个分类任务和预测任务的测试结果与分析。下面将 Hb-SRBFNN-OL 算法用于从 UCI 机器学习库中选择的 4 个基准分类问题, 并将其用于 Lorenz 混沌时间序列预测和污水处理过程中出水总磷 (total phosphorus, TP) 质量浓度预测的实际问题。为了进一步说明本文所提方法的优越性, 将其与现有多种算法进行了比较。

所有实验均在 Microsoft Windows 10.0 操作系统, 时钟速度为 3.6 GHz、RAM 为 32 GB 的计算机上使用 MATLAB 2020b 软件运行并通过均方根误差 (root mean square error, RMSE) 测量其性能。

3.1 Hb-SRBFNN-OL 算法用于分类任务的实验结果与分析

3.1.1 实验设置

在本节实验中, 下层优化器使用的 LM 方法中的参数设置如下: 预定阈值为 0.01, 最大迭代步数为 20, $\mu = 0.0015$ 。所有数据集的输入标准化为 $[-1, 1]$, 采用“赢家通吃”策略, 输出激活度最高的类别, 实验数据集及详细参数如表 1 所示。在本实验中, 所有数据集被分成两部分: 随机选择 85% 作为训练集, 剩余 15% 作为测试集。在上层优化器提供网络结构的基础上, 下层优化器使用训练集训练子网络权值, 并将权值结果反馈至上层优化器, 用以测试集测试 Hb-SRBFNN-OL 的泛化性能。

表 1 数据集的描述

Table 1 Description of data sets

数据集	属性	类别	样本个数
Iris	4	3	150
Wine	13	3	178
Heart	13	2	270
Cancer	9	2	699

3.1.2 实验结果与分析

每个实例的所有实验具有相同的初始参数, 通过对所有数据集多次独立运行, 计算连续 30 次实验

的统计结果。表 2 为使用 Hb-SRBFNN-OL 算法进行训练和测试的错误率、隐藏层神经元数量以及运行时间的统计结果,表中给出了训练和测试错误率

的平均值和标准差(standard deviation, SD)。此外,还给出了隐藏层神经元个数的最小值和 SD 以及平均运行时间。

表 2 Hb-SRBFNN-OL 算法的训练和测试结果

Table 2 Training and test results of Hb-SRBFNN-OL algorithm

数据集	训练误差率/%		测试误差率/%		隐藏层神经元个数		平均运行时间/s
	平均值	SD	平均值	SD	最小值	SD	
Iris	0.016	0.013	0.006	0.012	2	1.465	27.353
Wine	0.006	0.012	0.002	0.001	2	1.056	42.482
Heart	0.117	0.041	0.104	0.049	3	1.561	41.630
Cancer	0.189	0.006	0.014	0.006	3	0.893	67.619

分析表 2,不难得出以下结论:1) 与以往建模方式不同,本文将训练过程和测试过程整合成一个总系统,通过上下 2 层交互学习,得到最优的网络结构和网络参数。测试误差率比训练误差率小得多,可见本文所提方法可以有效地避免过拟合。Hb-SRBFNN-OL 算法在测试误差率足够小的情况下,对 4 种分类问题都具有较好的泛化能力,这也进一步表明,Hb-SRBFNN-OL 算法能够在过拟合和欠拟合之间取得很好的平衡。2) Hb-SRBFNN-OL 算法同时考虑训练集和测试集,基于此,自适应动态调整

RBFNN 结构和连接权值,从而得到针对每个分类任务的最紧凑的网络结构。

为了进一步证明 Hb-SRBFNN-OL 算法的优越性,基于表 1 中的 3 个数据集,与单层学习算法进行了比较。其中,粒子群优化算法(particle swarm optimization, PSO)、鲸鱼优化算法(whale optimization algorithm, WOA)^[9]等元启发式方法均采用固定网络结构,隐藏层神经元的数量根据 $2 \times I + 1$ 选择, I 表示数据集中属性的个数,实验结果如表 3 所示。

表 3 与其他单层模型在分类任务上的对比结果

Table 3 Comparison results with other single-layer model on classification task

数据集	算法	分类准确率/%			隐藏层神经元个数	平均运行时间/s
		平均值	最佳值	SD		
Wine	WOA ^[9]	0.849 6	0.963 5	0.026 3	22	40.17
	PSO	0.803 4	0.910 3	0.037 9	22	46.29
	Hb-SRBFNN-OL	0.984 3	1.000 0	0.003 1	4	39.25
Heart	WOA ^[9]	0.714 6	0.792 4	0.001 6	7	52.31
	PSO	0.753 4	0.775 9	0.004 7	7	75.16
	Hb-SRBFNN-OL	0.812 5	0.817 3	0.009 1	3	56.93
Cancer	WOA ^[9]	0.975 1	0.986 4	0.005 9	16	41.38
	PSO	0.972 6	0.946 7	0.004 1	16	49.15
	Hb-SRBFNN-OL	0.980 1	0.995 4	0.004 2	5	42.96

从表 3 中可以看出,在 Wine、Heart 和 Cancer 这 3 个基准数据集上,Hb-SRBFNN-OL 算法在分类准确率平均值和最佳值方面明显优于 PSO、WOA 等元启发式方法且所需要的隐藏层神经元数量最少。需要特别说明的是,由于本文所提算法的下层优化器选择 LM 方法,在网络权值调整过程中需要计算并存储 Hessian 矩阵,增加了计算量。因此,在除 Wine 数据集之外的 2 个实例中,Hb-SRBFNN-OL 算法的运行时间都稍长于 WOA 算

法,并且无论在网络结构和分类准确率方面都充分证明了其在分类任务中的优越性。

3.2 Hb-SRBFNN-OL 算法用于 Lorenz 混沌时间序列预测任务的实验结果与分析

在本节中,将 Hb-SRBFNN-OL 算法应用于 Lorenz 混沌时间序列的预测。Lorenz 时间序列系统是一种大气对流数学模型,并且是一个高度非线性的三维系统,已被广泛用作评价学习算法性能^[24-25],定义为

$$\begin{cases} d\mathbf{X}(t)/dt = a_1\mathbf{Y}(t) - a_1\mathbf{X}(t) \\ d\mathbf{Y}(t)/dt = a_2\mathbf{X}(t) - \mathbf{X}(t)\mathbf{Z}(t) - \mathbf{Y}(t) \\ d\mathbf{Z}(t)/dt = \mathbf{X}(t)\mathbf{Y}(t) - a_3\mathbf{Z}(t) \end{cases} \quad (28)$$

式中: a_1 、 a_2 、 a_3 为 Lorenz 时间序列中的参数, $a_1 = 10$, $a_2 = 8$, $a_3 = 8/3$; $\mathbf{X}(t)$ 、 $\mathbf{Y}(t)$ 和 $\mathbf{Z}(t)$ 表示三维空间向量。

在本实验中,使用由步长为 0.01 的四阶 Runge-Kutta 方法产生的 9 000 个数据样本,前 6 000 个样本作为训练数据,后 3 000 个样本作为测试数据,比率为 2:1。不失一般性地,网络结构均初始化为 3-15-1,学习因子设置为 0.5,中心 \mathbf{c} 初始化为 $[-10, 10]$ 的随机数,宽度 σ 和隐藏层-输出层权值 \mathbf{W} 分别初始化为 $[0, 8]$ 和 $[0, 2]$ 的随机数,最大迭代次数设置为 200,预测结果如图 3 ~ 6 所示。图 3 为 Hb-SRBFNN-OL 算法的预测结果。图 4 为 Hb-SRBFNN-OL 算法与 RBFNN、混合双层 RBFNN (hybrid-bilevel RBFNN, Hb-RBFNN)、SO-RBFNN 的预测误差。图 5 为用 Lorenz 时间预测时 Hb-SRBFNN-OL 模型中 4 个子网络的隐藏层神经

元个数。从图 4、5 可以看出, Hb-SRBFNN-OL 算法的预测性能良好,通过在迭代过程中不断调整网络结构,得到网络结构为 7 ~ 9 时,性能最好,并且测试误差保持在 $[-0.45, 0.45]$ 。图 6 为线性系数变化相应的子模型误差结果,通过调整线性系数,避免了子网络中的梯度消失问题,加速了网络收敛。

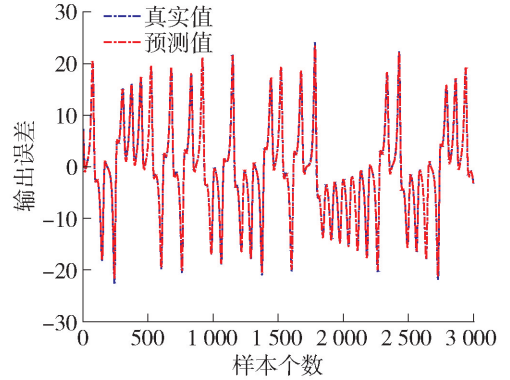


图 3 Hb-SRBFNN-OL 算法用于 Lorenz 时间序列预测的实验结果

Fig. 3 Experimental results of the Hb-SRBFNN-OL algorithm for Lorenz time series prediction

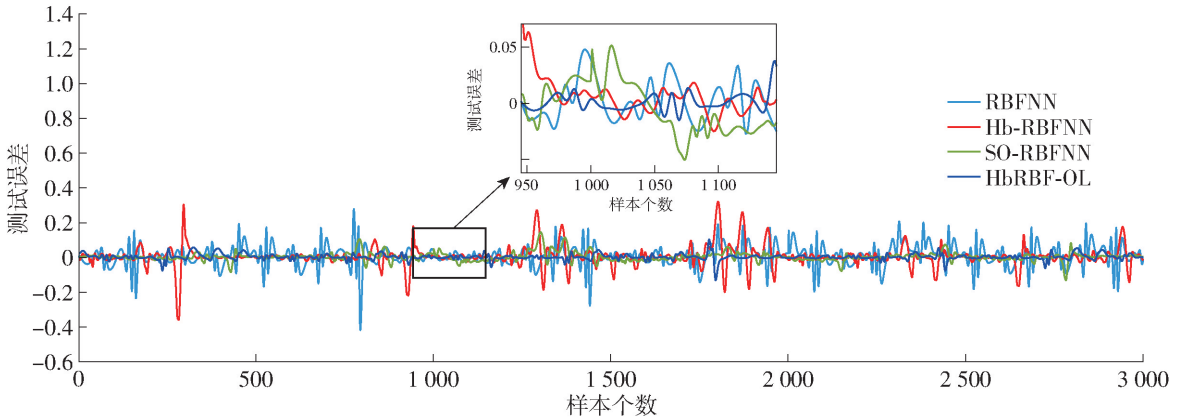


图 4 不同方法用于 Lorenz 时间序列预测的测试误差

Fig. 4 Test errors of Lorenz time series prediction through different methods

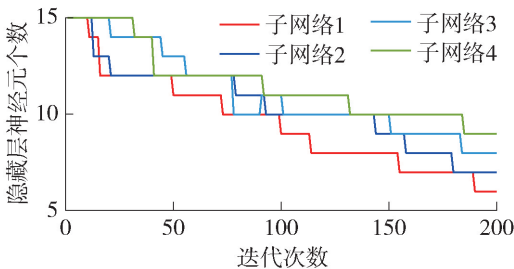


图 5 子模型的隐藏层神经元个数(Lorenz 时间序列预测)

Fig. 5 Number of hidden layer neurons of sub-model (Lorenz time series prediction)

为进一步验证 Hb-SRBFNN-OL 算法的有效性,将所得结果与 RBFNN、Hb-RBFNN、SO-RBFNN 这三种算法进行比较。以 80 次仿真实验结果的平均值作为最终结果,详细对比结果如表 4 所示。可以看出, Hb-SRBFNN-OL 算法中第 1 个子网络的平均运行时间为 36.16 s,在所有对比算法中时间是最短的,训练时间提高了 30.49% ~ 54.18%,充分说明更新全局损失函数可以加快网络的学习速度。主要原因是通过调整全局线性系数代替子网络参数,使得网络不必等待所有子网络收敛。此外,测试集的

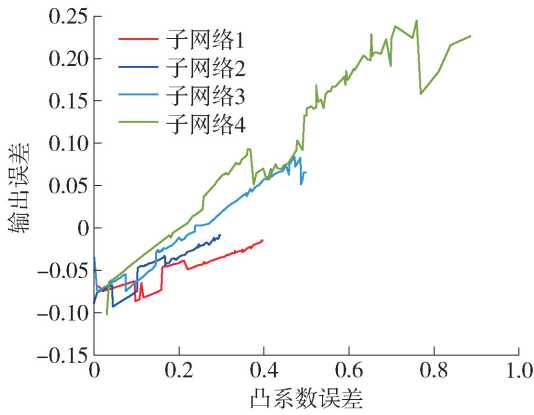


图6 子模型误差与系数关系(Lorenz时间序列预测)

Fig. 6 Relation of error and coefficient of sub-model (Lorenz time series prediction)

表4 Lorenz时间序列预测的不同算法比较

Table 4 Comparison of different algorithms for Lorenz time series prediction

算法	子网序号	神经元个数	迭代次数	训练时间/s	RMSE	
					训练集	测试集
RBFNN	1	15	1 000	76.17	0.032 5	0.032 1
	1	15	1 000	78.91	0.019 1	0.031 7
Hb-RBFNN	2	15			0.033 5	
	3	15			0.035 7	
SO-RBFNN	4	15			0.016 8	
	1	9	500	52.02	0.030 2	0.031 4
Hb-SRBFNN-OL	1	6	200	36.16	0.021 6	0.019 4
	2	7			0.010 2	
	3	8			0.023 6	
	4	9			0.011 2	

对未处理的数据直接建模,必然会导致系统性能较差。为了保证预测结果的可靠性和准确性,先对异常数据进行剔除。最终,随机选取其中的 10 000 组数据作为训练样本,3 000 组数据作为测试样本。

在本实验中,利用主成分分析方法选取预测模型输入:温度(T)、氧化还原电位(oxidation reduction potential, ORP)、溶解氧(dissolved oxygen, DO)、总悬浮固体(total suspended solids, TSS)以及 pH。模型训练分为两部分:1) 下层模型用于训练子网络权值,利用 LM 方法训练 RBFNN 子网络中的参数,利用全局损失函数训练组合线性系数 $\hat{\alpha}$; 2) 上层模型用于训练网络结构,利用网络复杂度和测试误差组成的损失函数对隐藏层神经元个数进行自适应调整,获得紧凑且泛化性能较好的预测模型,实验结果如图 7~10 所示。

从图 7 中可以看出,本文所提算法有很好的拟合效果。图 8 和图 9 分别为 Hb-SRBFNN-OL 与

RMSE 保持在 $[0.001 5 \sim 0.005 0]$,与其他 3 种算法相比,降低了 38.22%~39.56%。基于表 4 的分析结果,不难得出结论,本文所提出的 Hb-SRBFNN-OL 算法能够以更高的精度逼近 Lorenz 系统,并且具有更紧凑的网络和更好的泛化能力。

3.3 污水处理过程出水 TP 质量浓度预测

出水 TP 质量浓度是评价污水处理过程性能的一个重要参数。然而,由于活性污泥法的生物特性,出水 TP 的测定非常困难。因此,本实验采用所提出的 Hb-SRBFNN-OL 算法预测出水 TP 质量浓度。

实验采用北京某小型污水处理厂 2016 年 1 月至 6 月的 60 000 组数据,由于测量精度、操作方法等因素的影响,所采集的数据存在一定的误差。若

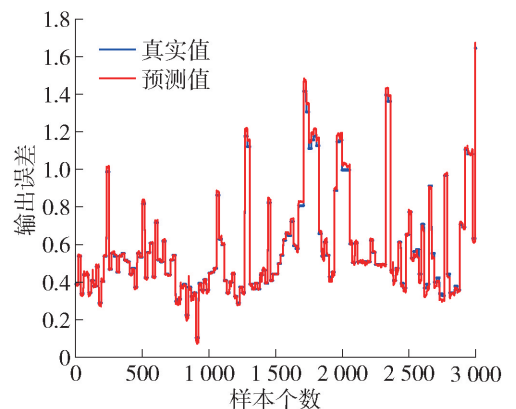


图7 Hb-SRBFNN-OL算法用于TP质量浓度预测的实验结果

Fig. 7 Experimental results of the Hb-SRBFNN-OL algorithm for TP mass concentration prediction

RBFNN、Hb-RBFNN、SO-RBFNN 的预测误差及 Hb-SRBFNN-OL 模型中 4 个子网络的隐藏层神经元个

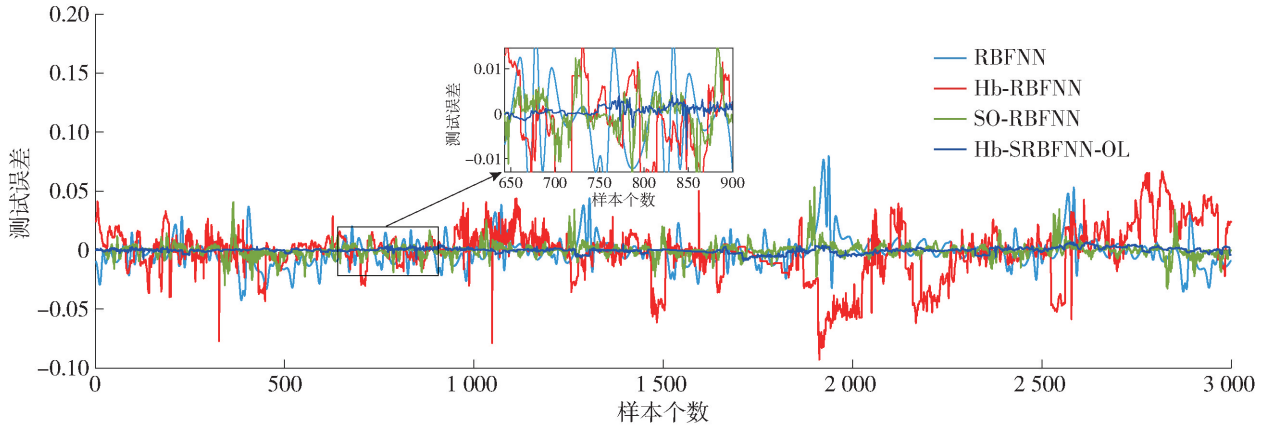


图 8 不同方法用于 TP 质量浓度预测的测试误差

Fig. 8 Test errors of TP mass concentration prediction through different methods

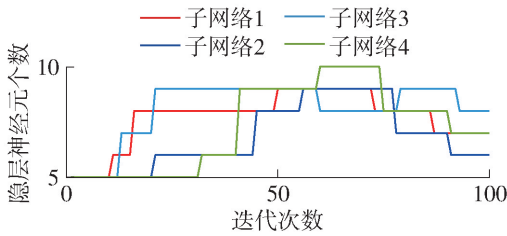


图 9 子模型的隐藏层神经元个数 (TP 质量浓度预测)

Fig. 9 Number of hidden layer neurons of sub-model (mass concentration prediction of TP)

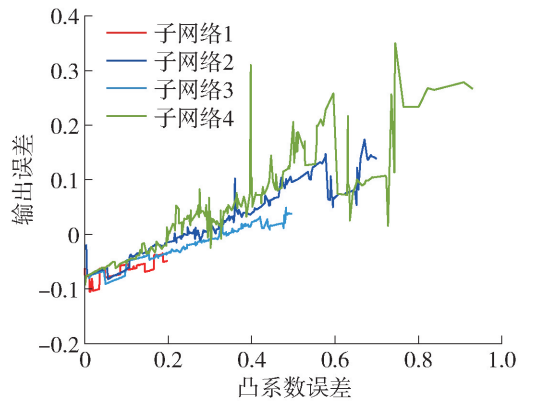


图 10 子模型误差与系数的关系 (TP 质量浓度预测)

Fig. 10 Relation of error and coefficient of sub-model (mass concentration prediction of TP)

数。从图中可以看出, Hb-SRBFNN-OL 算法可以较好地预测 TP 质量浓度, 预测误差较小, 保持在 $[-0.02 \sim 0.02]$ 。这主要是由于该算法基于结构复杂度和测试误差将参数调整和结构设计结合在一起, 把模型更新转换成优化问题, 并根据优化结果确定网络的最佳结构, 有效地平衡了过拟合和欠拟合问题, 使其泛化性能得到提高。图 10 为线性系数 $\hat{\alpha}$ 的更新过程。从图中可以看出, 随着系数 $\hat{\alpha}$ 的不断调整, 子网络误差不断更新, 通过始终保持总和不变

降低了运算量, 保证了 Hb-SRBFNN-OL 算法的快速收敛和成功应用。

为避免实验结果带来的随机性, 同时, 为了验证算法的整体性能, 将 4 种算法独立运行 60 次, 并求取平均值, 详细对比结果如表 5 所示。可以看出,

表 5 TP 质量浓度预测的不同算法比较

Table 5 Comparison of different algorithms for TP mass concentration prediction

算法	子网序号	神经元个数	迭代次数	训练时间/s	RMSE 平均值	
					训练集	测试集
RBFNN	1	5	500	24.80	0.029 1	0.021 3
	1	5	300	22.95	0.052 5	0.027 2
Hb-RBFNN	2	5			0.033 2	
	3	5			0.021 3	
	4	5			0.020 2	
SO-RBFNN	1	10	100	20.61	0.020 7	0.026 2
	1	8	100	13.89	0.021 3	0.013 9
Hb-SRBFNN-OL	2	6			0.010 1	
	3	7			0.013 5	
	4	6			0.012 9	

Hb-SRBFNN-OL 算法能较好地预测目标函数的值,并且具有最小的 RMSE 平均值。同时,该算法可以随着样本数据的变化自适应调整网络结构,使得 RBF 网络结构更加紧凑。最重要的是通过调整线性组合的全局线性系数 $\hat{\alpha}$ 而非网络参数,加速了网络收敛,这直接导致了 Hb-SRBFNN-OL 算法的预测时间最短。从表中也可看出,网络的训练时间与 SO-RBFNN 算法和 RBFNN 算法相比分别提高了 32.61% 和 43.99%。上述结果表明,Hb-SRBFNN-OL 算法在预测出水 TP 质量浓度方面比其他对比方法更合适、更有效。

4 讨论

本文的目的是开发一种有效的进化学习机制来有效解决过拟合和欠拟合问题。实验结果表明,Hb-SRBFNN-OL 算法的性能优于现有的一些方法。其主要贡献如下:

1) 学习速度快。Hb-SRBFNN-OL 的关键问题是学习速度,本文提出以全局损失函数代替局部损失函数策略以调整网络参数,无须等待所有 RBFNN 收敛,有效避免了网络训练中的梯度消失问题。此外,系数 $\hat{\alpha}$ 的更新公式为差分方程,计算简单。表 4、5 中的结果表明,Hb-SRBFNN-OL 算法比现有的对比方法有更快的学习速度。

2) 泛化能力强。泛化性能是 Hb-SRBFNN-OL 算法的另一个重要组成部分。混合双层进化学习机制可以兼顾训练过程和测试过程,通过持续交互更新网络参数和网络结构,克服了传统方法先训练后测试的弊端,有效解决了过拟合和欠拟合问题。

此外,在进行网络结构自组织时,Hb-SRBFNN-OL 算法基于网络复杂度和测试误差调整隐藏层神经元个数,而在选择系数 c_1 、 c_2 时,始终没有成熟的理论依据,因为测试误差最小为最终目标,故尝试 $\{\varepsilon_1, \varepsilon_2\} = \{(0.5; 0.4; 0.3; 0.2; 0.1), (0.5; 0.6; 0.7; 0.8; 0.9)\}$ 时的网络预测结果。结果表明,当 $\varepsilon_1 = 0.3, \varepsilon_2 = 0.7$ 时,所提出的 Hb-SRBFNN-OL 算法能够获得比其他方法更好的预测精度。另外,在上述 TP 质量浓度预测实验中,本文提出的 Hb-SRBFNN-OL 算法可以获得比同类算法更快的学习速度和更小的测试误差。

5 结论

1) 针对神经网络训练过程中过拟合和欠拟合难以取得平衡的问题,本文构建了一个混合双层进

化学习模型。针对该双层学习模型,基于进化学习机制,提出了一种交互进化学习算法 Hb-SRBFNN-OL。上层基于网络复杂度和测试误差优化网络结构,以便得到该场景下最精简紧凑的网络结构;下层基于训练误差对网络参数进行调整,得到一组最优的连接权值,以便提高准确率和泛化能力。此外,提出了线性叠加多层神经网络结构的思路,用系数组合的全局损失函数代替各个子网络的局部损失函数的策略进行进化学习迭代更新,加速了学习过程。

2) 通过在 4 个分类任务和 2 个预测问题上进行实验,并与其他现存的多种算法进行比较,结果表明,文中所提算法获得的 SO-RBFNN 结构更紧凑,泛化能力更强,并且预测准确率最高可提升 30%,为工业过程中的预测和分类任务提供了一个更加灵活、开放、准确可靠的框架。

3) 未来将进一步开展以下两方面工作:进一步提高模型精度,考虑使用粒子群优化、遗传算法等启发式策略,寻找最优的网络结构;进一步降低双层学习算法的计算成本,开发极限学习机等非迭代学习算法,得到神经网络的最优连接权值。

参考文献:

- [1] HORNIK K, STINCHCOMBE M, WHITE H. Multilayer feedforward networks are universal approximators [J]. *Neural Networks*, 1989, 2(5): 359-366.
- [2] LEE C M, KO C N. Time series prediction using RBF neural networks with a nonlinear time-varying evolution PSO algorithm[J]. *Neurocomputing*, 2009, 73 (1/2/3): 449-460.
- [3] MENG X, ZHANG Y, QIAO J F. An adaptive task-oriented RBF network for key water quality parameters prediction in wastewater treatment process [J]. *Neural Computing and Applications*, 2021, 33 (17): 11401-11414.
- [4] HAN H G, MA M L, QIAO J F. Accelerated gradient algorithm for RBF neural network [J]. *Neurocomputing*, 2021, 441(21): 237-247.
- [5] 刘威,付杰,周定宁,等. 基于反时限混沌郊狼优化算法的 BP 神经网络参数优化[J]. *控制与决策*, 2021, 36(10): 2339-2349.
LIU W, FU J, ZHOU D N, et al. Parameter optimization of BP neural network based on coyote optimization algorithm with inverse time chaotic [J]. *Control and Decision*, 2021, 36(10): 2339-2349. (in Chinese)
- [6] 乔俊飞,董敬娇,李文静. 改进的小脑模型神经网络及其在时间序列预测中的应用[J]. *北京工业大学学报*,

- 2021, 47(6): 598-606.
- QIAO J F, DONG J J, LI W J. Improved cerebellar model neural network and its application in time series prediction [J]. Journal of Beijing University of Technology, 2021, 47(6): 598-606. (in Chinese)
- [7] HAGAN M T, MENHAJ M B. Training feedforward networks with the Marquardt algorithm [J]. IEEE Transactions on Neural Networks, 1994, 5(6): 989-993.
- [8] LV C, XING Y, ZHANG J Z, et al. Levenberg-Marquardt backpropagation training of multilayer neural networks for state estimation of a safety-critical cyber-physical system [J]. IEEE Transactions on Industrial Informatics, 2018, 14(8): 3436-3446.
- [9] ALJARAH I, FARIS H, MIRJALILI S. Optimizing connection weights in neural networks using the whale optimization algorithm [J]. Soft Computing, 2018, 22(1): 1-15.
- [10] TSOULOS I G, TZALLAS A, TSALIKAKIS D. Evolutionary based weight decaying method for neural network training [J]. Neural Processing Letters, 2018, 47(2): 463-473.
- [11] YU J B, WANG S J, XI L F. Evolving artificial neural networks using an improved PSO and DPSO [J]. Neurocomputing, 2008, 71(4/5/6): 1054-1060.
- [12] QIAO J F, LI S Y, HAN H G, et al. An improved algorithm for building self-organizing feedforward neural networks [J]. Neurocomputing, 2017, 262(1): 28-40.
- [13] SHENG W G, SHAN P X, MAO J F, et al. An adaptive memetic algorithm with rank-based mutation for artificial neural network architecture optimization [J]. IEEE Access, 2017, 5(1): 18895-18908.
- [14] MARKUS S, DAN C, MIKE B. A memetic Pareto evolutionary approach to artificial neural networks [C] // Advances in Artificial Intelligence. Berlin: Springer, 2001: 1-12.
- [15] ABBASS H A. Speeding up backpropagation using multi-objective evolutionary algorithms [J]. Neural Computing, 2003, 15(11): 2705-2726.
- [16] GOH C K, TEOH E J, TAN K C. Hybrid multi-objective evolutionary design for artificial neural networks [J]. IEEE Transactions on Neural Networks, 2008, 19(9): 1531-1548.
- [17] ALMEIDA L M, LUDERMIR T B. A multi-objective memetic and hybrid methodology for optimizing the parameters and performance of artificial neural networks [J]. Neurocomputing, 2010, 73(7/8/9): 1438-1450.
- [18] LOGHMANIAN S M R, JAMALUDDIN H, AHMAD R, et al. Structure optimization of neural network for dynamic system modeling using multi-objective genetic algorithm [J]. Neural Computing, 2012, 21(6): 1281-1295.
- [19] GLOROT X, BENGIO Y. Understanding the difficulty of training deep feedforward neural networks [J]. Journal of Machine Learning Research, 2010, 9(1): 249-256.
- [20] HOCHREITER S, SCHMIDHUBER J. Long short-term memory [J]. Neural Computing, 1997, 9(8): 1735-1780.
- [21] 李文静, 王潇潇. 基于简化型 LSTM 神经网络的时间序列预测方法 [J]. 北京工业大学学报, 2021, 47(5): 480-488.
- LI W J, WANG X X. Time series prediction method based on simplified LSTM neural network [J]. Journal of Beijing University of Technology, 2021, 47(5): 480-488. (in Chinese)
- [22] LAU M M, LIM K H. Investigation of activation functions in deep belief network [C] // 2017 2nd International Conference on Control and Robotics Engineering. Piscataway: IEEE, 2017: 201-205.
- [23] XU B, WANG N Y, CHEN T Q, et al. Empirical evaluation of rectified activations in convolutional network [EB/OL]. [2022-02-16]. <https://arxiv.org/abs/1505.00853>.
- [24] HAN H G, ZHOU W D, QIAO J F, et al. A direct self-constructing neural controller design for a class of nonlinear systems [J]. IEEE Transactions Neural Network and Learning Systems, 2015, 26(6): 1312-1322.
- [25] QIAO J F, LI F, YANG C L, et al. A self-organizing RBF neural network based on distance concentration immune algorithm [J]. Journal of Automatica Sinica, 2020, 7(1): 276-291.

(责任编辑 梁洁)