

# 控制与决策

Control and Decision

## 基于SAPSO算法的RBF神经网络设计

张伟, 黄卫民

引用本文:

张伟, 黄卫民. 基于SAPSO算法的RBF神经网络设计[J]. 控制与决策, 2021, 36(9): 2305–2312.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2020.0176>

---

## 您可能感兴趣的其他文章

Articles you may be interested in

### 基于R2指标和目标空间分解的高维多目标粒子群优化算法

R2 indicator and objective space partition based many-objective particle swarm optimizer

控制与决策. 2021, 36(9): 2085–2094 <https://doi.org/10.13195/j.kzyjc.2020.0113>

### 基于滚动时域粒子群优化的视频去雾算法

Receding horizon particle swarm optimization based video defogging algorithm

控制与决策. 2021, 36(9): 2218–2224 <https://doi.org/10.13195/j.kzyjc.2019.1183>

### 基于小波变换与差分变异BSO–BP算法的大坝变形预测

Dam deformation prediction based on wavelet transform and differential mutation BSO–BP algorithm

控制与决策. 2021, 36(7): 1611–1618 <https://doi.org/10.13195/j.kzyjc.2019.1431>

### 区分交通流模式的混合服务路口信号控制策略

Signal control strategies of mixed service intersections to discriminate traffic flow patterns

控制与决策. 2021, 36(6): 1509–1515 <https://doi.org/10.13195/j.kzyjc.2019.1520>

### 基于强化学习的倒立摆分数阶梯度下降RBF控制

Reinforcement learning based fractional gradient descent RBF neural network control of inverted pendulum

控制与决策. 2021, 36(1): 125–134 <https://doi.org/10.13195/j.kzyjc.2019.0816>

# 基于SAPSO算法的RBF神经网络设计

张伟<sup>†</sup>, 黄卫民

(河南理工大学 电气工程与自动化学院, 河南 焦作 454003)

**摘要:** 针对径向神经网络结构和参数的动态优化问题, 提出一种基于敏感度分析和粒子群优化的RBF神经网络(SAPSO-RBF)优化算法. 算法通过初始化各粒子信息数, 基于粒子敏感度分析, 对算法学习阶段粒子信息进行增加和删减, 确定第一次收敛时网络结构大小; 算法达到收敛后, 对最优粒子进行敏感度分析, 删除冗余信息, 使算法重新发散; 根据算法发散和收敛次数提出一种惯性权重更新方法, 使算法在解空间内进行多次发散和收敛, 增强算法搜索能力的同时减小网络结构, 并给出SAPSO算法的收敛性证明. 仿真实验结果表明, SAPSO-RBF算法具有良好的自组织能力, 相较于其他自组织RBF神经网络优化算法, 在网络结构紧凑度和精度等方面有较大提升.

**关键词:** RBF神经网络; 敏感度分析; 粒子群优化; 多次收敛; 动态优化

中图分类号: TP183

文献标志码: A

DOI: 10.13195/j.kzyjc.2020.0176

开放科学(资源服务)标识码(OSID):



引用格式: 张伟, 黄卫民. 基于SAPSO算法的RBF神经网络设计[J]. 控制与决策, 2021, 36(9): 2305-2312.

## Design of RBF neural network based on SAPSO algorithm

ZHANG Wei<sup>†</sup>, HUANG Wei-min

(College of Electrical Engineering and Automation, Henan Polytechnic University, Jiaozuo 454003, China)

**Abstract:** Aiming at the dynamic optimization of structure and parameters of the radial basis function (RBF) neural network, an optimization algorithm based on sensitivity analysis (SA) and particle swarm optimization (PSO) for the RBF neural network (SAPSO-RBF) is proposed. Firstly, the number of particle information is randomly initialized, and the particle information is added and deleted by the sensitivity analysis in the learning phase, and the network structure of the algorithm in first convergence is determined. Then, after the algorithm reaches convergence, we analyze the sensitivity of the optimal particles, delete the redundant information, and make the algorithm re-divergent. An inertia weight update method is proposed to make the algorithm perform multiple divergence and convergence in the solution space, which enhances algorithm search ability while reducing network structure, and the convergence of SAPSO algorithm is proved. Finally, the results of experiments show that the proposed SAPSO-RBF algorithm has good self-organizing ability and has greatly improved the network structure compactness and accuracy compared with some other existing methods.

**Keywords:** RBF neural network; sensitivity analysis; particle swarm optimization; convergence several times; dynamic optimization

## 0 引言

径向基函数(radial basis function, RBF)神经网络被广泛用于模式识别、信号处理、非线性系统建模及控制等方面<sup>[1]</sup>. 相较于其他神经网络, RBF神经网络结构简单, 训练速度快. RBF神经网络的性能主要受网络参数和结构大小影响<sup>[2-3]</sup>, 优化网络参数和确定合适的网络结构以提高精度和减小计算复杂度是研究RBF神经网络的重要问题之一<sup>[4]</sup>.

为解决上述问题, 学者们相继提出了一些神经网络的参数学习和结构优化算法. Huang等<sup>[2]</sup>提出

一种增长修剪RBF神经网络(growing and pruning RBF, GAP-RBF); 文献[5]进一步提出一种改进的广义增长修剪RBF神经网络(generalized growing and pruning RBF, GGAP-RBF), 根据隐含层神经元的贡献度对网络结构进行调整, 并利用顺序学习算法对网络参数进行更新, 但这种方法需要完整的训练过程数据以获取网络初始参数, 在处理实际问题时很难满足条件. Han等<sup>[6]</sup>提出一种基于信息定位法的自组织回归RBF神经网络(information-oriented algorithm RRBF, IOA-RRBF), 网络有较为良好的性能以及紧凑

收稿日期: 2020-02-23; 修回日期: 2020-06-11.

基金项目: 国家自然科学基金项目(61703145); 河南省高校科技创新团队项目(20IRTSTHN019).

<sup>†</sup>通讯作者. E-mail: zwei1563@126.com.

的结构,但算法的参数和结构调整方法较复杂,且收敛性得不到保证.文献[7]提出一种自组织RBF神经网络(self-organizing RBF, SORBF),通过隐含层神经元之间的交互信息对神经网络的参数和结构进行调整.文献[8-9]也分别提出了对RBF神经网络的参数和结构进行调节的方法.但是,大部分SORBF算法都是通过不同的方法分别进行参数的更新和结构的调整,算法呈现出相对独立的几个部分,不能很好地解决在动态网络结构下对网络各个参数进行调整的问题<sup>[10]</sup>.

进化类算法因其在寻优方面的优势,受到学者们的关注<sup>[3,11-13]</sup>.粒子群优化(particle swarm optimization, PSO)算法快速的收敛过程和简单的初始参数设置<sup>[14]</sup>,使PSO算法有效应用于神经网络结构动态优化<sup>[1]</sup>.Feng等<sup>[3]</sup>提出一种基于PSO算法的自组织RBF神经网络来加速网络的学习过程;Alexandridis等<sup>[11]</sup>提出一种基于模糊规则和PSO算法的自组织RBF神经网络.这些研究都表明基于PSO算法的RBF神经网络具有较高的精度和较为紧凑的网络结构,但因为PSO算法易陷入局部最优,造成早熟收敛等问题,所以仍然存在许多问题需要解决.文献[12]提出一种基于自适应PSO算法的自组织RBF神经网络(adaptive PSO self-organizing RBF, APSO-SORBF);文献[13]进一步提出一种基于梯度多目标粒子群优化算法的自组织RBF神经网络(adaptive gradient multiobjective PSO self-organizing RBF, AGMOPSO-SORBF),通过粒子的适应度函数值构造回归函数确定算法的惯性权重,在一定程度上改善了PSO算法易陷入局部最优的问题.但是,上述几种算法并未明确给出应进行增加和删减的粒子信息选取规则,且当算法达到收敛时,不能确定算法最终精度和网络结构的合适性.

鉴于上述存在的问题,本文提出一种基于敏感度分析(sensitivity analysis, SA)和PSO的自组织RBF神经网络(SAPSO-RBF)优化算法.利用敏感度分析,实现算法学习过程中粒子信息的增加和删减,确定算法学习初期合适的网络结构.当算法达到收敛时,删除最优粒子的冗余信息,并对其余粒子进行位置调整,使算法重新发散,跳出收敛的极小值点,在解空间内继续搜索并再次收敛,给出根据算法发散和收敛次数提出的权重更新方法下的收敛性证明.所提算法能够实现网络参数和结构的同步优化,利用多次收敛的方法减小网络结构,加快算法收敛速度,并最终收敛在更有效的位置.仿真实验验证了所提算法的有效

性.

## 1 SAPSO-RBF神经网络

### 1.1 RBF神经网络

RBF神经网络是由输入层、隐含层和输出层构成的3层网络结构,其结构如图1所示.网络的激活函数为高斯核函数,RBF神经网络的输出为

$$y = \sum_{i=1}^H \omega_i \Phi_i(x) + b = \sum_{i=1}^H \omega_i e^{-\frac{\|x - \mu_i\|^2}{\sigma_i^2}} + b. \quad (1)$$

其中: $x$ 为网络输入, $\Phi_i(x)$ 为隐含层第*i*个神经元的输出, $\omega_i$ 为网络的第*i*个连接权值, $\mu_i$ 、 $\sigma_i$ 分别为第*i*个隐含层神经元的中心和宽度, $b$ 为网络偏置, $i = 1, 2, \dots, H$ , $H$ 为网络隐含层神经元个数.

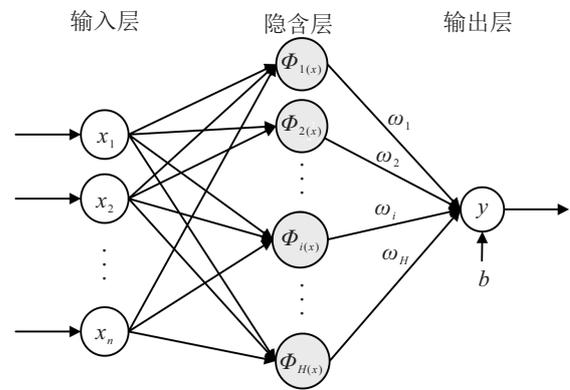


图1 RBF神经网络结构

### 1.2 SAPSO算法

为动态调整RBF神经网络的结构和参数,算法的粒子位置信息由网络的中心 $\mu_i(t)$ 、宽度 $\sigma_i(t)$ 、连接权值 $\omega_i(t)$ 和偏置 $b(t)$ 组成,粒子信息数即为隐含层神经元个数.

粒子位置 $a_i(t)$ 可表示为

$$a_i(t) = [\mu_{i,1}(t), \sigma_{i,1}(t), \omega_1(t), \dots, \mu_{i,H}(t), \sigma_{i,H}(t), \omega_H(t), b(t)], \quad (2)$$

其中 $i = 1, 2, \dots, s$ , $s$ 为种群中粒子个数.

粒子的速度 $v_i(t)$ 可表示为

$$v_i(t) = [v_{i,1}(t), v_{i,2}(t), \dots, v_{i,m}(t)], \quad (3)$$

其中 $m$ 为解空间的维度.为记录粒子在运动过程中的最优位置,引入个体最优粒子 $P_i(t)$ 和全局最优粒子 $G(t)$ .

粒子速度更新公式为

$$v_{i,d}(t+1) = W(t)v_{i,d}(t) + c_1 r_1 (P_{i,d}(t) - a_{i,d}(t)) + c_2 r_2 (G_d(t) - a_{i,d}(t)). \quad (4)$$

其中: $W(t)$ 为算法的惯性权重; $c_1$ 和 $c_2$ 为算法的学习因子; $r_1$ 和 $r_2$ 为 $[0,1]$ 之间的随机数; $d = 1, 2, \dots, m$ .

根据粒子速度对粒子位置进行更新,有

$$a_{i,d}(t+1) = a_{i,d}(t) + v_{i,d}(t+1). \quad (5)$$

个体最优粒子  $P_i(t)$  和全局最优粒子  $G(t)$  的更新规则分别为

$$P_i(t+1) = \begin{cases} P_i(t), & f(a_i(t+1)) \geq f(P_i(t)); \\ a_i(t+1), & f(a_i(t+1)) < f(P_i(t)). \end{cases} \quad (6)$$

$$G_i(t+1) = \arg \min_{P_i} f(P_i(t+1)). \quad (7)$$

其中  $f(\cdot)$  为 PSO 算法的适应度函数. 使用网络输出的均方根误差  $E_i(t)$  作为适应度函数,有

$$f(a_i(t)) = E_i(t) = \sqrt{\frac{1}{K} \sum_{i=1}^K (y(t) - y_d(t))^2}. \quad (8)$$

其中:  $y(t)$  和  $y_d(t)$  分别为  $t$  时刻 RBF 神经网络的实际输出和期望输出,  $K$  为样本数据个数.

基于所提算法在学习过程中收敛和发散的次数,

提出一种惯性权重  $W(t)$  取值方案

$$W(t) = W_s - (W_s - W_e) \left( \frac{t - A((t - t^*) + T_\alpha)}{M} \right)^2. \quad (9)$$

其中:  $M$  为算法最大迭代次数;  $A = \alpha/\alpha + k$ ,  $\alpha$  为算法当前收敛的次数,  $k$  为大于等于 1 的正整数;  $t^*$  为算法上一次收敛时的迭代次数;  $T_\alpha$  为正常数;  $W_s$  和  $W_e$  分别为算法的起始惯性权重和终止惯性权重, 其取值为  $[0, 1]$  之间的常数.

RBF 神经网络隐含层神经元个数  $H$  的初始值为取值范围内的随机数构成的向量,  $H$  的取值范围由处理问题的复杂程度决定, 粒子信息数的更新方式为

$$H_i = \begin{cases} H_i - 1, & H_i > H_{\text{best}}; \\ H_i + 1, & H_i < H_{\text{best}}. \end{cases} \quad (10)$$

其中:  $H_i$  为第  $i$  个粒子的信息数;  $H_{\text{best}}$  为当前最优粒子的信息数. 图 2 以 3 个粒子的信息更新为例, 给出了粒子信息数的调整过程.

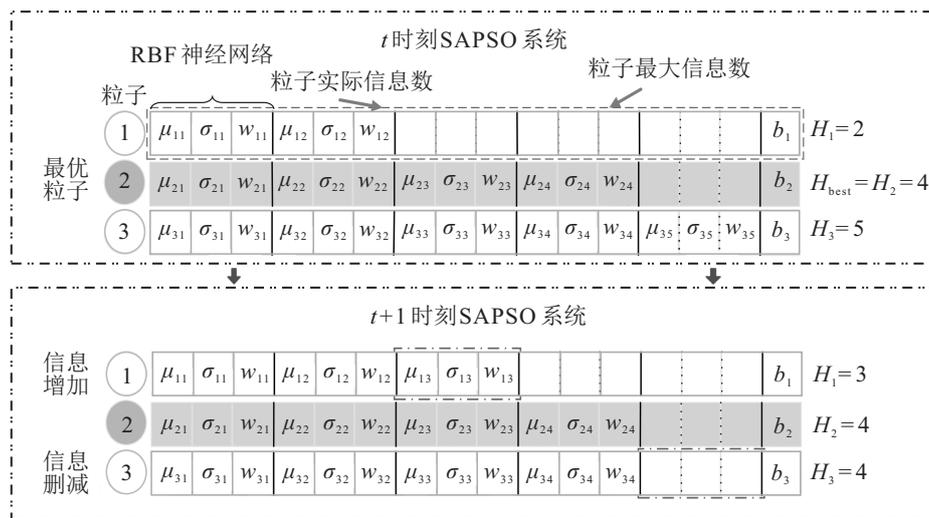


图 2 粒子信息数调整过程

在算法学习阶段, 若各粒子的信息数不同, 则不能根据式 (4) 和 (5) 对粒子位置和速度进行更新. 为解决这一问题, 在参数初始化时设定粒子最大信息数<sup>[3]</sup>, 若粒子信息数小于当前全局最优粒子的信息数, 则除粒子实际信息外, 用 0 补充粒子信息; 若粒子信息数大于当前全局最优粒子信息数, 则舍弃多余信息, 使粒子满足更新位置和速度的条件.

### 1.3 基于敏感度的粒子信息调整

#### 1.3.1 粒子信息敏感度

粒子增加和删减的信息位置由敏感度给定. 敏感度表示隐含层神经元输出对神经网络输出所做的贡献, 表征隐含层节点的有效度, 其描述为

$$S_i = E_{\text{WithoutUnit}_i} - E_{\text{WithUnit}_i}. \quad (11)$$

其中:  $E_{\text{WithUnit}_i}$  为网络输出误差,  $E_{\text{WithoutUnit}_i}$  为不包含第  $i$  个隐节点的网络输出误差. 引入一个重视程度系数  $\beta_i$ , 其表示意义为: 若  $\beta_i = 0$ , 则隐节点  $i$  对网络的其余部分没有影响; 若  $\beta_i = 1$ , 则隐节点  $i$  为一个传统的节点. 因此式 (11) 可改写为

$$S_i = E_{\beta_i=0} - E_{\beta_i=1}. \quad (12)$$

构造关于误差目标函数和  $\beta_i$  的函数逼近  $S_i$ , 即

$$\lim_{\xi \rightarrow 1} \left( \frac{E_{\beta_i=\xi} - E_{\beta_i=1}}{\xi - 1} \right) = \left. \frac{\partial E}{\partial \beta_i} \right|_{\beta_i=1}. \quad (13)$$

对式 (13) 做近似处理, 即当  $\xi = 0$  时仍然成立, 有

$$\frac{E_{\beta_i=0} - E_{\beta_i=1}}{-1} \approx \left. \frac{\partial E}{\partial \beta_i} \right|_{\beta_i=1}. \quad (14)$$

得到  $S_i$  的逼近表达式

$$S_i = E_{\beta_i=0} - E_{\beta_i=1} \approx \left. \frac{\partial E}{\partial \beta_i} \right|_{\beta_i=1}. \quad (15)$$

由于式(14)中近似的影响,  $S_i$  直接计算时波动较大, 使用一阶惯性滤波得到较平稳的敏感度值.

$$S_i(t+1) = 0.8S_i(t) + 0.2 \frac{\partial E}{\partial \beta_i}. \quad (16)$$

对式(16)做归一化处理, 有

$$\hat{S}_i(t) = \frac{S_i(t)}{\sum_{i=1}^H |S_i(t)|}. \quad (17)$$

### 1.3.2 基于敏感度分析的粒子信息分裂和删减

在算法迭代过程中, 粒子需要增加信息时, 计算当前神经网络各隐含层神经元敏感度, 使敏感度最大的隐含层神经元分裂. 若分裂前隐含层神经元数为  $H$ ,  $t$  时刻敏感度最大的隐含层神经元为  $j$ , 则为保证神经元分裂前后, 网络输出误差差不发生跳变, 神经元  $j$  和新增神经元的参数<sup>[15]</sup> 分别设置为

$$\begin{cases} \mu_{i,j}(t) = \mu_{i,j}(t), \\ \sigma_{i,j}(t) = \sigma_{i,j}(t), \\ \omega_{i,j}(t) = (1 - \delta)\omega_{i,j}(t); \end{cases} \quad (18)$$

$$\begin{cases} \mu_{i,H+1}(t) = \mu_{i,j}(t), \\ \sigma_{i,H+1}(t) = \sigma_{i,j}(t), \\ \omega_{i,H+1}(t) = \delta\omega_{i,j}(t). \end{cases} \quad (19)$$

其中  $\delta$  为区间  $[0.2, 0.5]$  中的常数.

粒子需要进行信息删减时, 计算当前神经网络各隐含层神经元敏感度, 删除敏感度最小的隐含层神经元.

当算法达到收敛后, 通过对全局最优粒子进行敏感度分析. 若存在  $\hat{S}_{G,j} < \eta$ , 则判断最优粒子的第  $j$  个信息冗余, 将其删除, 其中  $\eta$  为设定的敏感度阈值. 最优粒子信息和信息数应调整为

$$G(t) = [g_1(t), \dots, g_{j-1}(t), g_{j+1}(t), \dots, g_H(t)], \quad (20)$$

$$H_{\text{best}} = H_{\text{best}} - 1. \quad (21)$$

由式(10)对其他粒子携带的信息进行调整, 改变粒子在解空间内的位置, 使粒子在解空间内由收敛变为发散, 算法继续搜索并再次达到收敛. 通过上述方法使算法进行多次收敛, 不仅可以删除冗余隐含层神经元, 而且可以使算法跳出局部极小值重新搜索, 有效避免算法陷入局部极小值.

## 2 SAPSO 算法收敛分析

为便于计算, 定义下列变量:

$$\theta_1 = c_1 r_1, \theta_2 = c_2 r_2, \theta = \theta_1 \theta_2. \quad (22)$$

为证明所提算法的收敛性, 给出3个基本假设.

**假设1** 粒子最优位置  $P_i(t)$  和最优全局位置  $G(t)$  满足条件  $\{P_i(t), G(t)\} \in \Gamma$ , 其中  $\Gamma$  为解空间.  $P_i(t)$  和  $G(t)$  存在最优取值  $P^*$  和  $G^*$ .

**假设2** 存在  $0 < T_\alpha < t, k > 0$  使得  $AT_\alpha < t^*$ .

**假设3** 存在  $\theta_1 \geq 0, \theta_2 \geq 0$  以及合适的  $\theta$  取值, 使得  $0 \leq \theta \leq 2(1 + W(t))$  成立.

基于假设1~假设3, 当  $\theta_1 \geq 0, \theta_2 \geq 0$  时, 粒子位置  $a_i(t)$  将收敛至  $(\theta_1 P^* + \theta_2 G^*) / (\theta_1 + \theta_2)$ , 粒子的速度  $v_i(t)$  将收敛至0. 下面给出其证明过程.

将式(22)代入(4), 粒子位置可写为

$$\begin{aligned} a_{i,d}(t+1) = & (1 + W(t) - \theta)a_{i,d}(t) - \\ & W(t)a_{i,d}(t-1) + \theta_1 P_{i,d}(t) + \theta_2 G_d(t). \end{aligned} \quad (23)$$

式(23)可用矩阵表示为

$$\begin{bmatrix} a_{i,d}(t+1) \\ a_{i,d}(t) \\ 1 \end{bmatrix} = \Psi(t) \begin{bmatrix} a_{i,d}(t) \\ a_{i,d}(t-1) \\ 1 \end{bmatrix}. \quad (24)$$

其中系数矩阵  $\Psi(t)$  为

$$\Psi(t) = \begin{bmatrix} 1 + W(t) - \theta & -W(t) & \theta_1 P_{i,d}(t) + \theta_2 G_d(t) \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (25)$$

$\Psi(t)$  的特征多项式为

$$(\lambda - 1)(\lambda^2 - (1 + W(t) - \theta)\lambda + W(t)) = 0. \quad (26)$$

求解得到  $\Psi(t)$  的特征值为

$$\begin{cases} \lambda_1 = 1, \\ \lambda_2 = \frac{1 + W(t) - \theta + \sqrt{(1 + W(t) - \theta)^2 - 4W(t)}}{2}, \\ \lambda_3 = \frac{1 + W(t) - \theta - \sqrt{(1 + W(t) - \theta)^2 - 4W(t)}}{2}. \end{cases} \quad (27)$$

则粒子位置  $a_{i,d}(t)$ <sup>[4]</sup> 可表示为

$$a_{i,d}(t) = l_1 + l_2 \lambda_2^t + l_3 \lambda_3^t, \quad (28)$$

其中  $l_1, l_2, l_3$  均为常数. 由文献[3]可以得到, 当假设1~假设3均成立时, 算法收敛的充要条件为

$$\begin{cases} 0 \leq W(t) < 1, \\ 0 < \theta < 2(1 + W(t)). \end{cases} \quad (29)$$

为保证算法收敛,  $W(t)$  应满足  $0 \leq W(t) < 1$ , 所

以式(9)应满足

$$-M < t - A(t - t^* + T_\alpha) < M. \quad (30)$$

当  $\alpha = 0$  时,  $A = 0, W(t)$  满足  $0 \leq W(t) < 1$ . 当  $\alpha \geq 1$  时, 有  $t = t_\alpha + t^*$ , 其中  $t_\alpha$  为算法第  $\alpha$  次收敛后重新计算的迭代次数. 式(30)可写为

$$-M < (t^* - AT_\alpha) + (1 - A)t_\alpha < M. \quad (31)$$

因为  $A < 1$ , 由假设2可知, 存在  $T_\alpha$  使  $(t^* - AT_\alpha) < t^*, (1 - A)t_\alpha < t_\alpha$ , 即

$$0 < (t^* - AT_\alpha) + (1 - A)t_\alpha < t. \quad (32)$$

综上, 在算法有限次收敛后仍然满足条件  $0 \leq W(t) < 1$ . 再由假设3可知, 算法满足式(29)的收敛条件. 由式(28)得到最终粒子收敛位置为

$$\lim_{t \rightarrow \infty} a_{i,d}(t) = l_1. \quad (33)$$

令  $t = 0, 1, 2$ , 分别代入式(24), 得到  $l_1$  的值为

$$\lim_{t \rightarrow \infty} a_{i,d}(t) = \lim_{t \rightarrow \infty} (\theta_1 P_{i,d}(t) + \theta_2 G_d(t)) / (\theta_1 + \theta_2). \quad (34)$$

由假设1可以得到<sup>[16]</sup>

$$\lim_{t \rightarrow \infty} P_{i,d}(t) = P^*, \quad \lim_{t \rightarrow \infty} G_d(t) = G^*. \quad (35)$$

粒子位置  $a_i(t)$  最终收敛至

$$\lim_{t \rightarrow \infty} a_i(t) = (\theta_1 P^* + \theta_2 G^*) / (\theta_1 + \theta_2). \quad (36)$$

由文献[13]可得到, 若假设1~假设3均成立, 则粒子速度  $v_i(t)$  将收敛至0, 即

$$\lim_{t \rightarrow \infty} v_i(t) = 0. \quad (37)$$

综上所述, 所提算法在式(9)的惯性权重更新方法下, 通过粒子信息删减并调整粒子位置, 使算法能够在学习阶段进行多次发散和收敛, 最终收敛在更优的位置.

### 3 仿真实验

为验证所提出 SAPSO-RBF 算法的有效性, 分别设计了时间序列预测和污水处理过程中出水氨氮浓度软测量的实验研究. 实验基本参数设置如下: 种群中粒子个数  $s = 100$ , 惯性权重参数  $k = 2, T_\alpha = 200$ , 惯性权重起始值和终止值分别为  $W_s = 0.9$  和  $W_e = 0.4$ , 学习因子为  $c_1 = c_2 = 1.49$ , 敏感度阈值  $\eta = 0.006$ . 本文采用多种算法进行对比, 各算法仿真实验所采用的训练数据和测试数据均为相同的数据样本, 算法最大评价次数均为5000次, 每种算法独立运行30次, 取平均值作为最终实验结果.

#### 3.1 Mackey-Glass时间序列预测

Mackey-Glass 时间序列预测被认为是评估算法学习性能的基准问题之一<sup>[4,6,13]</sup>. Mackey-Glass 时间

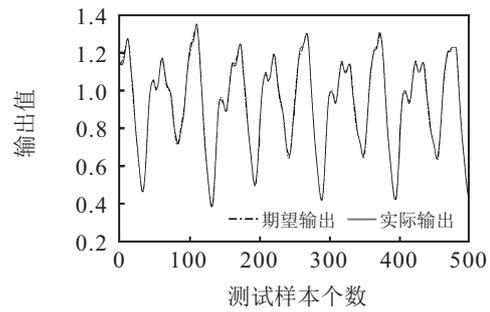
序列方程为

$$x(t+1) = (1-a)x(t) + \frac{bx(t-\tau)}{1+x^{10}(t-\tau)}. \quad (38)$$

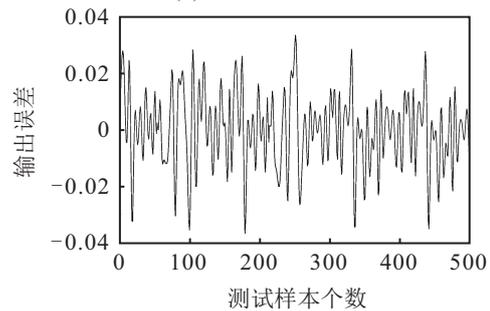
取  $a = 0.1, b = 0.2, \tau = 17$  进行仿真实验, 其预测模型为

$$x(t + \Delta t) = f[x(t), x(t - 6), x(t - 12), x(t - 18)]. \quad (39)$$

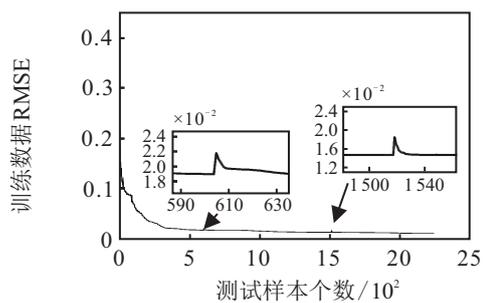
进行算法训练和测试时, 初始条件为  $x(0) = 1.2$ . 应用式(38)和(39)在  $t = 1 \sim 1000$  之间产生1000组数据, 取前500组数据作为实验训练样本, 后500组数据作为实验测试样本.



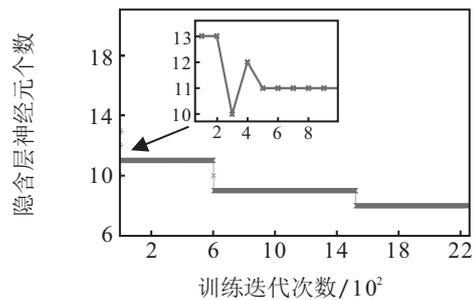
(a) 测试数据输出结果



(b) 测试数据输出误差



(c) 训练数据RMSE变化过程



(d) 隐含层神经元个数变化过程

图3 SAPSO-RBF算法对Mackey-Glass时间序列的预测

图3(a)和图3(b)分别给出了SAPSO-RBF算法的Mackey-Glass时间序列预测输出结果和误差,可以看出,SAPSO-RBF算法的预测输出能够较好地跟踪期望输出,相对误差小于0.04.图3(c)和图3(d)为SAPSO-RBF算法在学习过程中网络结构的动态调整过程,算法在学习过程初期通过较少次数的迭代确定算法第1次收敛时的网络结构大小;算法在整个学习阶段中共达到3次收敛,在约第600次迭代时算法达到第1次收敛,删除网络冗余隐含层神经元,算法发散;迭代至1520步左右,算法达到第2次收敛,网络中仍然存在冗余隐含层神经元,算法再次发散并最终收敛在更优的位置.经过多次收敛,算法获得了更高

的精度和更加紧凑的网络结构.

表1给出了几种不同算法对Mackey-Glass时间序列预测的实验结果.由表1可以看出,相较于PSO-RBF<sup>[4]</sup>、AI-RBF<sup>[8]</sup>、GA-RBF<sup>[17]</sup>、APSO-SORBF<sup>[13]</sup>、AI-PSO-RBF<sup>[18]</sup>以及SAIW-PSO-RBF<sup>[19]</sup>算法,SAPSO-RBF算法经过3次收敛后,算法的训练阶段RMSE、测试阶段RMSE和MAE更小,而且有更简单的网络结构.虽然GGAP-RBF<sup>[6]</sup>算法的测试阶段RMSE相较于SAPSO-RBF算法更小,但是在精度相差不多的情况下,SAPSO-RBF算法的网络结构更简单.同时,相较于其他对比算法,SAPSO-RBF算法的训练时间和测试时间更短.

表1 不同算法预测Mackey-Glass时间序列的性能指标比较

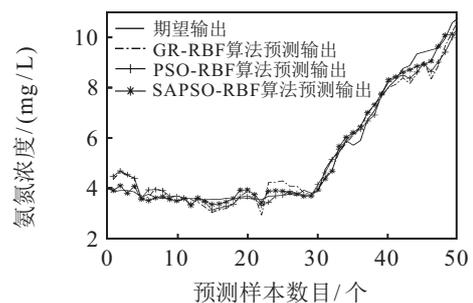
算法	训练阶段		测试阶段		隐含层神经元个数	训练时间/s	测试时间/s
	收敛次数	收敛位置(RMSE)	RMSE	MAE			
SAPSO-RBF	1	0.0189	0.0127	0.0089	11	804.5	0.0032
	2	0.0147			9		
	3	0.0126			8		
PSO-RBF <sup>[4]</sup>	1	0.0197	0.0208	0.0249	12	859.6	0.0047
GGAP-RBF <sup>[6]</sup>	—	0.0102*	0.0121*	0.0103	12*	—	0.0042
AI-RBF <sup>[8]</sup>	1	0.0143	0.0151*	0.0109	11*	892.9*	0.0045
GA-RBF <sup>[17]</sup>	1	0.0131	0.0133	0.0092	16	1862.2	0.0050
APSO-SORBF <sup>[14]</sup>	1	0.0129	0.0135*	0.0095	11*	832.7*	0.0039*
AI-PSO-RBF <sup>[18]</sup>	1	0.0164	0.0189*	0.0132	11*	1183.6*	0.0043*
SAIW-PSO-RBF <sup>[19]</sup>	1	0.0132	0.0166*	0.0145	11*	1135.9*	0.0053*

注: \*数据与原文献中一致

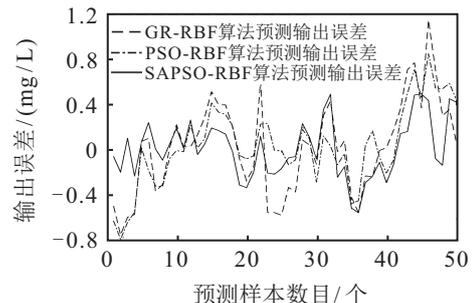
### 3.2 污水处理——出水氨氮浓度软测量

水体中氨氮(NH<sub>4</sub>-N)含量是衡量水体受污染程度的重要指标,预测出水氨氮浓度是污水处理过程中的一项重要工作.污水处理过程由于进水流量、水质成分及污染浓度波动剧烈,系统总是运行在非平稳状态.本文利用SAPSO-RBF算法对污水处理过程中出水氨氮浓度进行建模预测.采用北京某污水处理厂实际测量数据进行建模,实验前对采集的数据进行预处理,共得到190组数据,取前140组数据作为训练样本,取后50组数据作为测试样本.输入变量为温度(T)、好氧前端溶解氧浓度(DO)、总固体悬浮物(TSS)、出水酸碱度(pH)、出水氧化还原电位(ORP)、污泥龄(SRT),输出变量为出水氨氮浓度.

为验证所提算法在预测污水处理过程中出水氨氮浓度的性能,图4(a)和图4(b)分别给出了SAPSO-RBF、GR-RBF以及PSO-RBF<sup>[3]</sup>算法的预测输出和误差.由实验结果可以看出,SAPSO-RBF算法的预测值



(a) 不同算法的预测结果对比



(b) 不同算法的预测误差对比

图4 SAPSO-RBF算法预测NH<sub>4</sub>-N浓度的实验结果

能够基本跟随实测值, 相对误差小于0.6. 相较于GR-RBF和PSO-RBF算法, 所提算法的预测误差更小, 表明该模型是有效的, 且所提算法在出水氨氮浓度软测量问题上具有较好的性能.

表2给出了几种不同算法在预测污水处理过程中出水氨氮浓度时的训练阶段数据、测试阶段的

RMSE和MAE以及隐含层神经元个数和算法的训练时间及测试时间. 由表2可知, 相较于GR-RBF、SSORBF<sup>[20]</sup>、PSO-RBF以及AI-PSO-RBF<sup>[18]</sup>等算法, SAPSO-RBF算法不仅训练阶段RMSE以及测试阶段RMSE和MAE更小, 而且网络隐含层神经元个数更少, 并且SAPSO-RBF算法相较于其他几种算法的测试时间更短.

表2 不同算法对NH<sub>4</sub>-N浓度预测的性能指标比较

算法	训练阶段		测试阶段		隐含层神经元个数	训练时间/s	测试时间/s
	收敛次数	收敛位置(RMSE)	RMSE	MAE			
SAPSO-RBF	1	0.0607			10		
	2	0.0493	0.1791	0.1033	7	943.4	0.0021
	3	0.0488			6		
GR-RBF	—	0.0565	0.2906	0.1648	12	—	0.0038
PSO-RBF <sup>[3]</sup>	1	0.0698	0.2445	0.1325	12	923.2	0.0027
SSORBF <sup>[20]</sup>	—	0.0739	0.2351	0.1214	8	—	0.0029
AI-PSO-RBF <sup>[18]</sup>	1	0.0617	0.1973	0.1145	10	1206.7	0.0035

### 4 结论

本文提出一种基于敏感度分析和粒子群优化的自组织RBF神经网络(SAPSO-RBF)优化算法, 实现了RBF神经网络的结构和参数同步动态调整. 算法能够根据问题确定学习初期合适的网络结构, 并在学习阶段进行多次发散和收敛, 使算法能够跳出当前收敛的极小值继续搜索, 避免算法产生早熟收敛和陷入局部极小值, 增强了算法的全局搜索能力. 给出了算法在所提出惯性权重更新方法下的收敛性证明. 结合仿真实验研究, 并与其他几种自组织RBF神经网络进行比较, 实验结果表明, 所提算法最终获得的神经网络逼近能力更强, 结构更加紧凑.

### 参考文献(References)

[1] 杨刚, 王乐, 戴丽珍, 等. 基于AQPSO的RBF神经网络自组织学习[J]. 控制与决策, 2018, 33(9): 1631-1636. (Yang G, Wang L, Dai L Z, et al. AQPSO-based self-organization learning of RBF neural network[J]. Control and Decision, 2018, 33(9): 1631-1636.)

[2] Huang G B, Saratchandran P, Sundararajan N. An efficient sequential learning algorithm for growing and pruning RBF(GAP-RBF) networks[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B, 2004, 34(6): 2284-2292.

[3] Feng H M. Self-generation RBFNs using evolutionary PSO learning[J]. Neurocomputing, 2006, 70(1/2/3): 241-251.

[4] Semenov M A, Terkel D A. Analysis of convergence of an evolutionary algorithm with self-adaptation using a stochastic Lyapunov function[J]. Evolutionary Computation, 2003, 11(4): 363-379.

[5] Huang G B, Saratchandran P, Sundararajan N. A generalized growing and pruning RBF(GGAP-RBF) neural network for function approximation[J]. IEEE Transactions on Neural Networks, 2005, 16(1): 57-67.

[6] Han H G, Guo Y N, Qiao J F. Self-organization of a recurrent RBF neural network using an information-oriented algorithm[J]. Neurocomputing, 2017, 225: 80-91.

[7] 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 on Neural Networks and Learning Systems, 2015, 26(6): 1312-1322.

[8] Li F, Yang C L, Qiao J F. A novel RBF neural network design based on immune algorithm system[C]. Proceedings of the 36th Chinese Control Conference. Dalian: IEEE, 2017: 4598-4603.

[9] Zhang L, Li K, He H B, et al. A new discrete-continuous algorithm for radial basis function networks construction[J]. IEEE Transactions on Neural Network and Learning Systems, 2013, 24(11): 1785-1798.

[10] Xie T 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.

[11] Alexandridis A, Chondrodima E, Sarimveis H. Radial basis function network training using a non-symmetric partition of the input space and particle swarm optimization[J]. IEEE Transactions on Neural Networks and Learning Systems, 2013, 24(2): 219-230.

[12] Han H G, Lu W, Hou Y, et al. An adaptive-PSO-based self-organizing RBF neural network[J]. IEEE Transactions on Neural Networks and Learning Systems, 2018, 29(1): 104-117.

[13] Han H G, Wu X L, Zhang L, et al. Self-organizing RBF neural network using an adaptive gradient multiobjective particle swarm optimization[J]. IEEE Transactions on Cybernetics, 2019, 49(1): 69-82.

[14] 张强, 李盼池. 一种自适应多策略行为粒子群优化算法[J]. 控制与决策, 2020, 35(1): 115-122.  
(Zhang Q, Li P C. An adaptive multi-strategy behavior particle swarm optimization algorithm[J]. Control and Decision, 2020, 35(1): 115-122.)

[15] 乔俊飞, 韩红桂. RBF神经网络的结构动态优化设计[J]. 自动化学报, 2010, 36(6): 865-872.  
(Qiao J F, Han H G. Optimal structure design for RBFNN structure[J]. Acta Automatica Sinica, 2010, 36(6): 865-872.)

[16] Jiang M, Luo Y P, Yang S Y. Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm[J]. Information Processing

Letters, 2007, 102(1): 8-16.

[17] Duvvuri S P, Anmala J. Fecal coliform predictive model using genetic algorithm-based radial basis function neural networks (GA-RBFNNs)[J]. Neural Computing and Applications, 2019, 31(12): 8393-8409.

[18] Nickabadi A, Ebadzadeh M M, Safabakhsh R. A novel particle swarm optimization algorithm with adaptive inertia weight[J]. Applied Soft Computing, 2011, 11(4): 3658-3670.

[19] Taherkhani M, Safabakhsh R. A novel stability-based adaptive inertia weight for particle swarm optimization[J]. Applied Soft Computing, 2016, 38: 281-295.

[20] 卢超, 杨翠丽, 乔俊飞. 基于尖峰自组织径向基网络的氨氮软测量方法[J]. 信息与控制, 2017, 46(6): 752-758.  
(Lu C, Yang C L, Qiao J F. Soft-computing method for ammonia nitrogen prediction based on spiking self-organizing RBF neural network[J]. Information and Control, 2017, 46(6): 752-758.)

作者简介

张伟(1978—), 女, 副教授, 博士, 从事智能特征建模、神经网络结构优化、污水处理过程优化控制等研究, E-mail: zwei1563@126.com;

黄卫民(1997—), 男, 硕士生, 从事神经网络结构优化和算法设计的研究, E-mail: hweimin97520@163.com.

(责任编辑: 齐 霖)

下 期 要 目

车辆与无人机组合配送研究综述 ..... 任 璇, 等

迁移学习引导的变源域长短时记忆网络建筑负荷预测 ..... 张 勇, 等

基于反时限混沌郊狼优化算法的BP神经网络参数优化 ..... 刘 威, 等

带峰值能耗约束流水线调度的协同群智能优化 ..... 王 凌, 等

基于云模型的煤矿安全大数据多粒度表示方法及应用 ..... 代 劲, 等

多尺度决策系统中代价敏感的最优尺度组合 ..... 张清华, 等

$l_p$ -范数约束下MKL-OC-ELM的装备故障检测 ..... 刘 星, 等

有向切换拓扑条件下多航天器分组姿态协同控制 ..... 周绍磊, 等

基于迁移学习灰支持向量回归机的交互式进化计算 ..... 郭广颂, 等

基于虚拟力移动锚节点的3D-DVHop-ACR定位算法 ..... 张 晶, 等

一种基于免疫机理的确定性移动机器人路径规划算法 ..... 范俊岩, 等

基于2D-OSTU图像边缘检测的回转窑工况识别方法 ..... 徐逸峰, 等