

基于参数优化的 RBF 神经网络结构设计算法

翟莹莹, 左 丽, 张恩德
(东北大学 计算机科学与工程学院, 辽宁 沈阳 110169)

摘 要: 针对径向基函数(RBF)神经网络的结构优化问题,提出了一种基于参数优化的 RBF 神经网络优化算法.首先,改进 K-means ++ 算法,使得聚类算法更精确,为 RBF 神经网络的隐含层节点找到一个合适的初始中心;然后,考虑数据分布和缩放因子选择的影响,采用方差度量法计算隐含层神经元基函数的宽度;最后,修正网络参数,提高网络的非线性逼近能力.实验结果表明,本文提出的基于参数优化的 RBF 神经网络具有良好的逼近效果和泛化能力.

关 键 词: 径向基函数;神经网络;K-means ++ 算法;核函数;数据密度估计
中图分类号: TP 183 **文献标志码:** A **文章编号:** 1005-3026(2020)02-0176-07

Algorithm for Structure Design of RBF Neural Network Based on Parameter Optimization

ZHAI Ying-ying, ZUO Li, ZHANG En-de
(School of Computer Science & Engineering, Northeastern University, Shenyang 110169, China; Corresponding author: ZUO Li, E-mail: zuoli@mail.neu.edu.cn)

Abstract: An algorithm based on parameter optimization (KV-RBF) is proposed for optimizing RBF(radial basis function) neural network structure. Firstly, the K-means ++ algorithm, which makes the clustering algorithm more accurate, is improved to find an appropriate initial center for the hidden layer node of RBF neural network. Then, considering the influence of data distribution and scaling factor selection, the width of neuron basis function in the hidden layer is calculated by variance measurement. Finally, the network parameters are modified to improve the nonlinear approximation ability of the network. The experimental results show that the proposed RBF neural network based on parameter optimization has good approximation effect and generalization ability.
Key words: radial basis function (RBF); neural network; K-means ++ algorithm; kernel function; data density estimation

径向基函数(radial basis function, RBF)神经网络是一种高效的三层前馈式神经网络,其拓扑结构简单,学习速度快,理论上可以任意精度逼近任何单值连续函数^[1-2],主要用于非线性系统建模与控制^[3-4]、模式识别^[5]和时间序列预测^[6-7]等方面.由于 RBF 神经网络基于多变量差值理论,所有的数据均可作为隐含层节点,使得网络的泛化性能低下.因此,RBF 神经网络结构设计的核心问题是在保证精度要求的情况下,提高网络的泛化性能^[8].

RBF 的网络结构优化是一个开放的问题,如何调整 RBF 网络结构是当前关注的主要问题,仍需要进一步研究.为了提升 RBF 神经网络的预测性能,很多学者提出了不同的改进方法^[9-13]. Yang 等^[10]应用输出敏感度方差重要性的度量方法优化 RBF 神经网络结构(VS-RBF),采用 K-means 聚类算法确定网络隐含层的节点中心,采用随机初始化函数的宽度,提高网络的自适应能力.为了提高聚类算法的有效性,本文改进 K-means ++ 聚类算法,得到 MKM ++ 算法,确定了

合适的网络初始中心;考虑数据分布和缩放因子的自适应选择,采用方差度量优化法求取隐含层神经元基函数的宽度.基于上述参数的优化,本文提出一种两阶段学习的 RBF 神经网络结构优化方法(KV-RBF).通过仿真实验,验证了 KV-RBF 的准确性和有效性.

1 基于聚类算法的 RBF 神经网络

RBF 神经网络由三层前馈神经网络构成,包括输入层、隐含层和输出层,如图 1 所示.设 RBF 网络结构为 $n-m-p$ (n 为输入层节点数, m 为隐含层节点数, p 为输出层节点数),输入层向量为 $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$, 隐含层第 j 个节点的输出为

$$h_j(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x} - \boldsymbol{\varphi}_j\|^2}{2\sigma_j^2}\right). \quad (1)$$

式中: $h_j(\mathbf{x})$ 为径向基函数; $\|\cdot\|$ 为欧式范数; 径向基函数的中心向量为 $\boldsymbol{\varphi}_j = (\varphi_{j1}, \varphi_{j2}, \dots, \varphi_{jm})^T$; σ_j 为第 j 个基函数的扩展宽度.

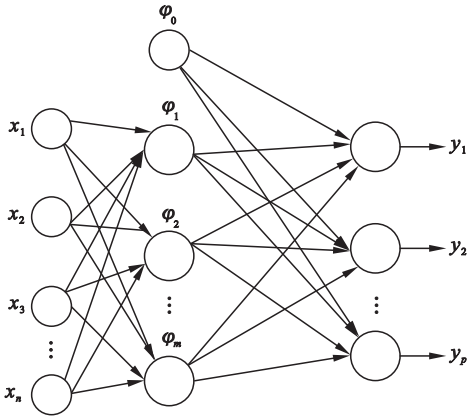


图 1 RBF 神经网络结构

Fig. 1 Structure of RBF neural network

输出层第 k 个节点的输出为

$$y_k(\mathbf{x}) = \sum_{i=1}^m w_{ki} h_i(\mathbf{x}) + w_{k0}. \quad (2)$$

式中: w_{ki} 为隐含层第 i 个节点到输出层第 k 个节点的连接权重, w_{k0} 为输出层第 k 个节点的偏差项. 输出层节点 k 的误差函数为

$$E_k = \frac{1}{2} (y_k(\mathbf{x}) - y_{d,k})^2. \quad (3)$$

式中, $y_{d,k}$ 为节点 k 的期望输出. RBF 网络的总误差函数为

$$E = \sum_{k=1}^m E_k. \quad (4)$$

在 RBF 神经网络结构中,决定网络收敛速度

的主要参数是隐含层节点中心和基函数宽度初值,合理选取这两个参数的初值能够提高收敛速度.本文采用聚类算法确定隐含层节点中心和基函数宽度.将得到的聚类中心 μ_j ($j = 1, 2, \dots, m$) 作为径向基函数的中心 μ , 径向基函数的宽度计算公式如下:

$$\sigma_1 = \sigma_2 = \dots = \sigma_m = \frac{d_{\max}}{\sqrt{2m}}. \quad (5)$$

式中: d_{\max} 为所有中心的最大距离.

使用最小二乘法求 RBF 神经网络的权值参数.式(2)可以写成矩阵形式:

$$\mathbf{Y} = \mathbf{H}\mathbf{w}. \quad (6)$$

$$\text{其中, } \mathbf{H} = \begin{bmatrix} 1 & h_{11} & h_{12} & \dots & h_{1m} \\ 1 & h_{21} & h_{22} & \dots & h_{2m} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & h_{p1} & h_{p2} & \dots & h_{pm} \end{bmatrix},$$

$$\mathbf{w} = (w_1, w_2, \dots, w_m)^T.$$

用 \mathbf{H} 的伪逆 $\mathbf{H}^\dagger = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T$ 求得

$$\mathbf{w} = \mathbf{H}^\dagger \mathbf{Y}. \quad (7)$$

该方法中径向基函数的中心和宽度可以在 RBF 神经网络训练前求出,训练中只要求出隐含层输出矩阵的伪逆 $\boldsymbol{\varphi}^\dagger$, 通过式(7)求出输出层和隐含层的权值.

基于聚类算法的 RBF 神经网络存在如下问题:算法对聚类初始条件敏感,初始中心选择不恰当会使聚类效果很差;宽度求取过程未考虑数据样本的分布.这种不稳定性是 RBF 神经网络实际应用中最大的瓶颈,因此,本文提出了基于参数优化的 RBF 算法,以改进算法的上述缺陷.

2 参数优化的 RBF 算法

2.1 K-means ++ 算法

K-means ++ 聚类算法是一种非监督学习方法.在样本中选取 k 个数据点作为初始聚类中心,形成 k 个分组;重复分配,更新分组,直到聚类中心的变化值在指定误差范围内,其算法步骤如下:

①随机选取整个样本数据集中的 k 个数作为初始聚类中心;

②计算样本点与聚类中心的欧氏距离,按照最小距离原则分配到最邻近聚类;

③用每个聚类中样本均值作为新聚类中心;

④重复步骤②和③,直到聚类中心变化值小于指定的误差;

⑤结束,得到 k 个聚类中心.

K-means ++ 选择中心点依赖于前面选取的初始中心点,而随机选取初始中心点导致了算法的不稳定. K-means ++ 并没有克服对离群点的敏感性,离群点的存在仍然使最终聚类中心偏离真正的中心点;算法往往选择离群点和边缘点作为下一个初始中心点,使得聚类性能下降.

2.2 改进的 K-means ++ 算法 (MKM ++)

为了克服 K-means ++ 算法的聚类缺陷,引入 MMOD (mountain method based outlier detection) 算法^[14]. 基于数据密度优化的 MMOD 算法很好地计算衡量了每个样本的密度,将其引入到 K-means ++ 算法中,用密度最大的样本代替 K-means ++ 算法随机选择的样本,作为第一个初始中心点,减弱对初始中心点的敏感性,即本文提出的 MKM ++ 算法.

MMOD 算法采用样本点的数据密度估计方法,区分正常数据点与离群数据点的数据密度,其数据密度计算公式为

$$M(p) = \frac{1}{k} \sum_{x_i \in \text{KNN}(p)} \exp(-\frac{d(p, x_i)}{\delta_k(x_i)}) . \quad (8)$$

式中: k 为近邻点数量; $d(p, x_i)$ 为数据点 p 到 x_i 的距离; $\delta_k(x_i)$ 为数据点 x_i 到其第 k 个近邻点的欧式距离,是局部尺度化参数; $\text{KNN}(p)$ 为数据点 p 的 k 近邻集合. 离群度,即数据点的离群程度,取值区间为 $0 \sim 1$. 对于聚类内部的点,假设 p 的 k 近邻属于同一个类,则 $\delta_k(x_i) > d(p, x_i)$, $M(p)$ 的值会比较大;在 p 的 k 近邻集合中,数据点越靠近中心,密度越大.

为了保证聚类效果,使用传统欧式距离与数据密度的乘积优化 K-means ++ 算法的初始中心点选取策略,在满足初始中心点应当尽可能远的前提下,选择密度大的点作为下一个初始中心点,即基于距离尽可能远、密度尽可能大的原则选择其余的初始中心点.

2.3 RBF 径向基函数宽度的选取

基于样本数据的聚类中心 $\mu = \{\mu_1, \mu_2, \cdots, \mu_m\}$, 根据每个中心的数据密度分布来分配相应的缩放因子,使得每个中心的宽度体现其数据的空间分布. 类簇样本分布不同,相同宽度或者相邻两个中心点的距离无法反映簇内中心点的稠密情况;因此,宽度的选择应该考虑到每个类簇之间的距离和簇内样本的分布密度.

以每个聚类中心和其他聚类中心的距离的平均值作为距离基数:

$$\text{meanD}(\mu_i) = \frac{\sum_{i \neq j} \text{dist}(\mu_i, \mu_j)}{k - 1} . \quad (9)$$

方差代表样本分布的稠密程度,将每个聚类看作一个数据集,求得每个聚类的方差 S_i :

$$S_i = \frac{1}{\text{size}(C_i)} \sum_{x \in C_i} \text{dist}(x, \mu_i)^2 . \quad (10)$$

式中: $\text{size}(C_i)$ 为属于聚类中心 μ_i 的样本数; C_i 为子样本; $\text{dist}(x, \mu_i)$ 为欧式距离. 该中心宽度的缩放因子 ε_i :

$$\varepsilon_i = S_i / (\frac{1}{k} \sum_{i=1}^k S_i) . \quad (11)$$

根据式 (10) 和式 (11), 得到每个中心宽度:

$$\sigma_i = \varepsilon_i \cdot \text{meanD}(\mu_i) . \quad (12)$$

当类内分布稠密 (类内方差 S_i 比较小) 时, 缩放因子也会变小, 宽度也会减小; 同理, 当类内数据分布比较稀疏 (类内方差 S_i 比较大) 时, 缩放因子则会变大, 宽度就会适当扩大, 此时, RBF 神经网络的选择性则会减小.

2.4 基于参数优化的 RBF 算法

根据上述分析得到基于参数优化的 RBF 神经网络结构算法 KV-RBF: 首先, 由 MKM ++ 求取 RBF 神经网络隐节点的中心; 然后, 采用基于方差的宽度求取算法求得 RBF 每个核函数的宽度; 最后, 采用最小二乘法求隐含层和输出层的权重. 算法实现步骤如下:

- ① 采用 MKM ++ 算法求得 RBF 神经网络隐节点的 k 个聚类中心;
- ② 使用式 (9) 计算每个聚类中心和其他聚类中心的距离的平均值;
- ③ 使用式 (10) 计算每个聚类的类内方差 S_i ;
- ④ 使用式 (11) 计算缩放因子 ε_i ;
- ⑤ 使用式 (12) 求得该核函数的宽度 σ_i , 求出 RBF 神经网络对应中心的宽度;
- ⑥ 基于中心 C_i 和宽度 σ_i , 应用最小二乘法求得隐含层与输出层的权值参数, 训练模型.

3 仿真实验

3.1 MKM ++ 算法实验

在聚类算法对比实验中, 用平方误差来度量不同算法的聚类效果. 其计算公式为

$$E = \sum_{i=1}^k \sum_{p \in C_i} \text{dist}(p, \mu_i)^2 . \quad (13)$$

式中: E 是数据集中所有对象的误差平方和; p 是空间中的点, 表示给定的数据对象; μ_i 是聚类中心.

选取 UCI 开源的数据集, 该数据集是 5 000 个矢量合成的二维数据, 由 15 个重叠程度不相同的

高斯集群组成. 进行 10 次模拟实验,得到两个算法的平方误差如图 2 所示.

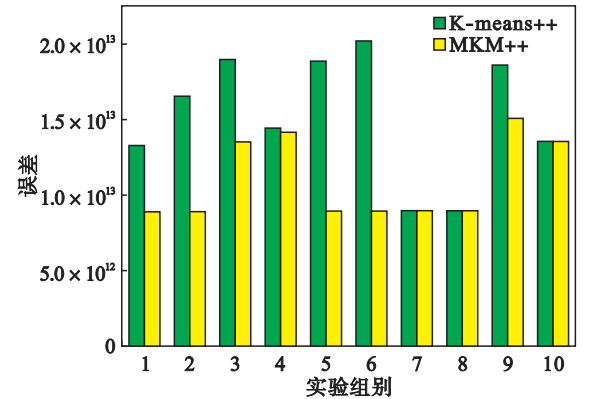


图 2 聚类平方误差对比

Fig. 2 Comparison of clustering squared error

由图 2 可知,K-means ++ 平方误差值大多为 10^{13} 数量级,而 MKM ++ 大多数为 10^{12} 数量级. 经过多次实验图像分析可以得出以下经验:误差值为 10^{12} 数量级的都能正确找到聚类中心,而误差为 10^{13} 数量级时,出现 1 个中心的聚类错误. MKM ++ 在稳定性和准确率上均比 K-means ++ 有一定的提升.

本文采用如下指标评定聚类效果:

①JC 系数(Jaccard coefficient):

$$JC = \frac{a}{a + b + c}; \tag{14}$$

②FMI 指数(Fowlkes and Mallows index):

$$FMI = \sqrt{\frac{a}{a + b} \times \frac{a}{a + c}}; \tag{15}$$

③RI 指数(Rand index):

$$RI = \frac{2(a + d)}{m(m - 1)}. \tag{16}$$

记数据集 $D = \{x_1, x_2, \cdots, x_m\}$, 聚类簇划分为 $C = \{c_1, c_2, \cdots, c_m\}$, 参考模型为 $C^* = \{C_1^*, C_2^*, \cdots, C_s^*\}$, λ 和 λ^* 分别为 C 和 C^* 对应的簇标记; 则

$a = |SS|, SS = \{(x_i, x_j) | \lambda_i = \lambda_j, \lambda_i^* = \lambda_j^*, i < j\};$

$b = |SD|, SD = \{(x_i, x_j) | \lambda_i = \lambda_j, \lambda_i^* \neq \lambda_j^*, i < j\};$

$c = |DS|, DS = \{(x_i, x_j) | \lambda_i \neq \lambda_j, \lambda_i^* = \lambda_j^*, i < j\};$

$d = |DD|, DD = \{(x_i, x_j) | \lambda_i \neq \lambda_j, \lambda_i^* \neq \lambda_j^*, i < j\};$

$a + b + c + d = m(m - 1)/2.$

上述指标度量结果均在 $[0, 1]$ 内,值越大,聚类越好.

对开源数据集 中的 5 个数据集 Iris, seeds, Pima Indians (P-I), wine, new-thyroid (n-th) 进行 30 次模拟实验,使用上述指标进行 K-means ++ (K-M) 和 MKM ++ 算法性能对比,实验结果如表

1 所示.

表 1 聚类评价指标对比

Table 1 Comparison of clustering evaluation indexes

数据集	JC		FMI		RI	
	K-M	MKM++	K-M	MKM++	K-M	MKM++
Iris	0. 69	0. 724	0. 817	0. 84	0. 877	0. 894
seeds	0. 679	0. 679	0. 809	0. 809	0. 873	0. 874
P-I	0. 458	0. 458	0. 631	0. 631	0. 551	0. 551
wine	0. 415	0. 432	0. 507	0. 606	0. 563	0. 666
n-th	0. 585	0. 614	0. 734	0. 776	0. 728	0. 687

由表 1 可知,在 Iris 数据集中指标提升比较明显;而在 seeds 数据集中 JC 系数和 FMI 指数提升较少,但 RI 指数还是略有提升. P-I 数据集的指标没有变化,但 n-th 数据集 3 个指标提升明显. 因此,MKM ++ 优于 K-means ++ 算法.

3. 2 基于参数优化的 RBF 对比实验

选取一维非线性函数:

$$y = 1 + (x + 5x^2) \sin(-2x^2), x \in [-4, 4], \tag{17}$$

随机产生 201 个数据(取值区间为 $[-4, 4]$) 的样本. 分别用传统 RBF, VS-RBF^[10] 和本文的 KV-RBF 算法对上述离散样本进行模拟,实验结果如图 3 所示.

由图 3 可知,KV-RBF 算法优于 RBF 及 VS-RBF 算法;RBF 和 VS-RBF 算法在中心点个数不足时很难达到理想的拟合效果,为了规避聚类带来的不稳定性,对比不同算法的均方误差,如图 4 所示.

KV-RBF 算法的均方误差小于 RBF 和 VS-RBF 算法,应用改进的 K-means ++ 方法后,能够提高模型的稳定性.

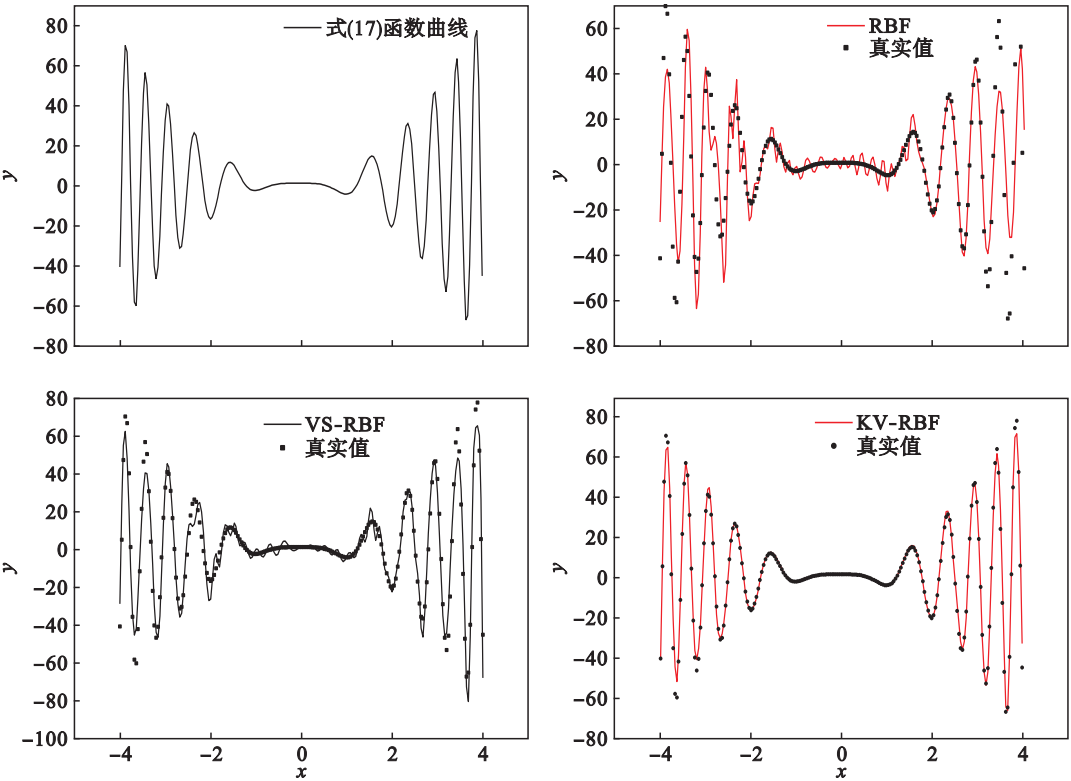
为了分析 KV-RBF 算法的有效性,选取二维三角函数:

$$y = \cos(2\pi k_1 x_1 + 2\pi k_2 x_2) \sin(2\pi k_2 x_2). \tag{18}$$

式中: y 为期望输出; $k_1 = 0. 05, k_2 = 0. 02$; 输出样本满足 $0 < x_1, x_2 < 36$; 随机产生训练样本 600 个. 对上述样本进行训练,期望误差为 0. 003. 在训练过程中,应用 KV-RBF 算法的逼近效果和拟合误差曲面如图 5 和图 6 所示.

由图 6 可知,KV-RBF 算法对函数的拟合误差在 0. 03 范围内,误差曲面比较平坦,具有很强的泛化性能.

图 7 为基于 KV-RBF 算法的网络隐含层节点数量的动态变化. 可以看出,随着训练时间的增加,隐含层神经元个数也在逐渐增加,该网络能够很好地自适应调整结构.



图中真实值为式(17)函数曲线的离散点
图 3 不同算法对一维函数的逼近效果

Fig. 3 Approximation of different algorithms to one dimensional function

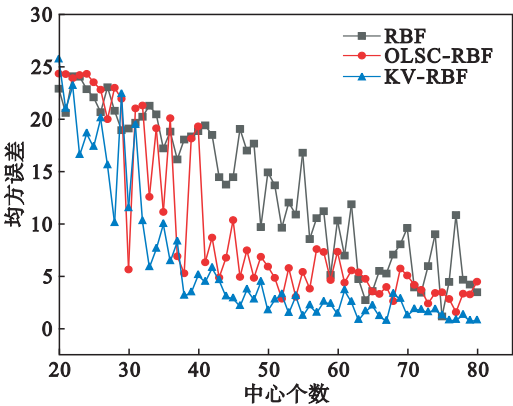


图 4 均方误差随核函数中心变化

Fig. 4 Mean square error varying with center of kernel function

抽取每个数据集的 3/4 作为训练样本,剩下 1/4 样本用于测试. 分别采用传统 RBF 算法和 KV-RBF 算法进行对比,为了避免聚类的随机性,隐含层中心的数量从 20 到 80 递增,得到的不同数据集的均方误差变化如图 8 所示.

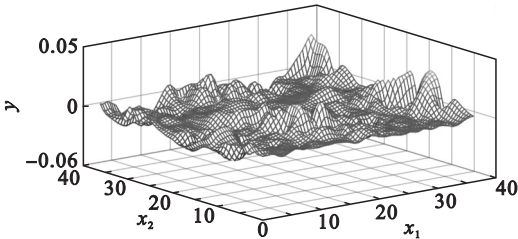


图 6 KV-RBF 算法逼近二维三角函数的拟合误差
Fig. 6 Fitting error of KV-RBF algorithm approximating to 2D trigonometric function

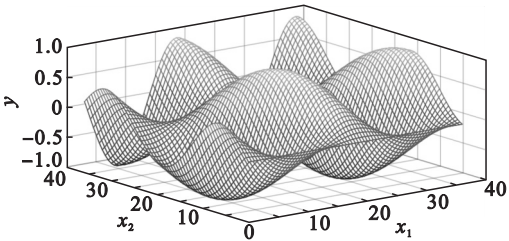


图 5 KV-RBF 算法对二维三角函数的逼近效果
Fig. 5 Approximation of KV-RBF algorithm to 2D trigonometric function

对 UCI 开源数据集中的 Boston housing, CCPP, Pollutant concentration 进行实验分析. 随机

由图 8 可知,RBF 神经网络的均方误差随着核函数聚类中心数量的增加而递减,符合 RBF 神经网络的原理. KV-RBF 算法的均方误差要小于传统 RBF 算法. 在 Boston housing 数据集中,KV-RBF 算法优于传统 RBF 算法,只有个别情况出现了均方误差比传统 RBF 算法大的现象;而对于 CCPP 数据集,KV-RBF 算法的精度明显提高,误差降低了 1 左右. 由于 Pollutant concentration 数据集中的数据有较强的离散性,而 CCPP 数据集中的数据具有连续性,且 RBF 核函数是基于欧式

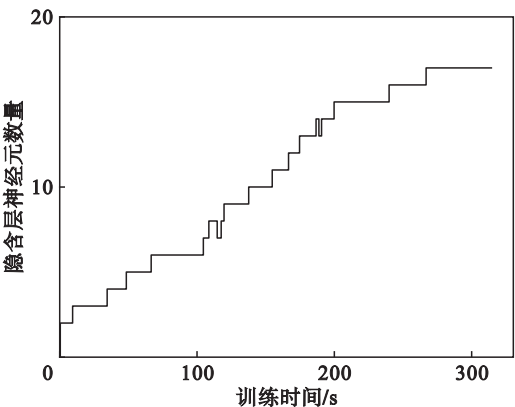


图 7 KV-RBF 训练中隐含层神经元数量的变化
Fig. 7 Variation of the number of neurons in KV-RBF hidden layer during training

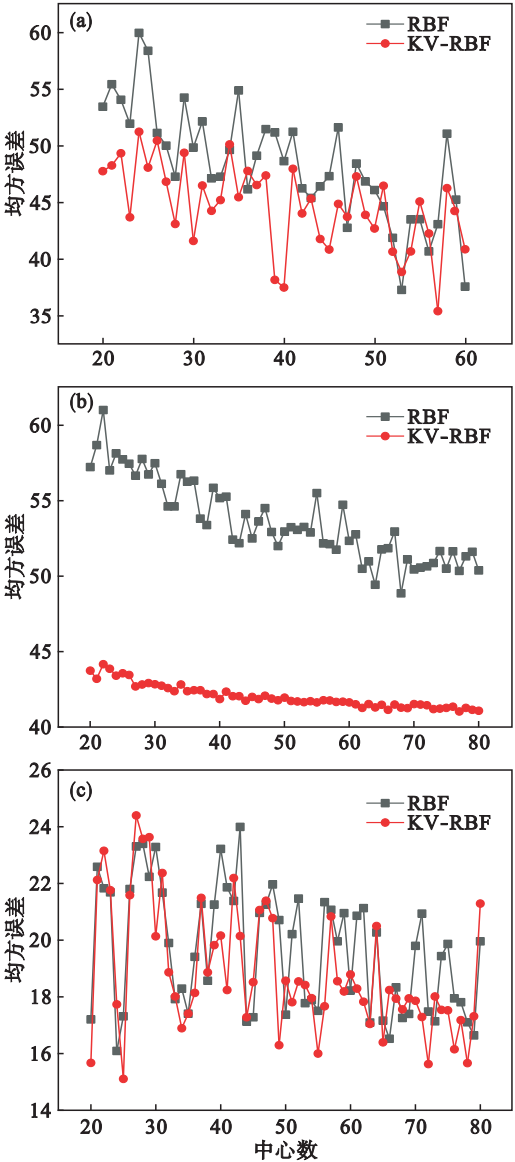


图 8 针对不同数据集均方误差随核函数中心数的变化

Fig. 8 Mean square error varying with the number of kernel function center for different datasets

(a)—Boston housing; (b)—CCPP;
(c)—Pollutant concentration.

距离的,处理离散性数据时均方误差会随着数据的波动而波动;因此,Pollutant concentration 的均方误差抖动比较明显,而 CCPP 基本上很稳定.由前述可知,基于参数优化的 KV-RBF 算法提升了网络的预测性能.

4 结 论

1) 基于 KV-RBF 算法的神经网络找到的聚类中心更稳定,并根据不同的数据集自适应调整隐含层的节点数.

2) 参数优化后的 RBF 网络具有较好的非线性逼近能力和收敛速度;本文提出的基于 KV-RBF 算法的神经网络表现出较好的训练结果和测试结果,说明其具有较好的泛化性能.

参考文献:

[1] Ferrari S, Bellocchio F, Puri V, et al. A hierachical RBF online learning algorithm for real-time 3-D scanner[J]. *IEEE Transactions on Neural Networks*, 2010, 21(2): 275 – 285.

[2] Kokshenev I, Braga A P. An efficient multi-objective learning algorithm for RBF neural network [J]. *Neurocomputing*, 2010, 73(16/17/18): 2799 – 2808.

[3] 张琨,崔胜民,王剑锋. 基于自适应 RBF 网络补偿的智能车辆循迹控制[J]. *控制与决策*, 2014, 29(4): 627 – 631. (Zhang Kun, Cui Sheng-min, Wang Jian-feng. Intelligent vehicle' s path tracking control based on self-adaptive RBF network compensation [J]. *Control and Decision*, 2014, 29(4): 627 – 631.)

[4] Kumar P, Kumar N, Panwar V. BF neural control design for SISO nonaffine nonlinear systems [J]. *Procedia Computer Science*, 2018, 125: 25 – 33.

[5] Skala V. RBF interpolation with CSRBF of large data sets [J]. *Procedia Computer Science*, 2017, 108: 2433 – 2437.

[6] Liu Y B, Zhao J Z, Xiao Y P. C-RBFNN: a user retweet behavior prediction method for hotspot topics based on improved RBF neural network [J]. *Neurocomputing*, 2018, 275: 733 – 746.

[7] 张军峰,胡寿松. 基于一种新型聚类算法的 RBF 神经网络混沌时间序列预测[J]. *物理学报*, 2007, 56(2): 713 – 719. (Zhang Jun-feng, Hu Shou-song. Chaotic time series prediction based on RBF neural networks with a new clustering algorithm [J]. *Acta Physica Sinica*, 2007, 56(2): 713 – 719.)

[8] Barandiaran X, Moreno A. On the nature of neural information: a critique of the received view 50 years later[J]. *Neurocomputing*, 2008, 71(4/5/6): 681 – 692.

[9] Platt J. A resource-allocating network for function interpolation [J]. *Neural Computation*, 1991, 3(2): 213 – 225.

[10] 杨一,高社生,胡高歌. 基于敏感度方差重要性的 RBF 神经网络结构优化算法[J]. *控制与决策*, 2015, 30(8): 1393 – 1398. (Yang Yi, Gao She-sheng, Hu Gao-ge. Optimal algorithm for RBF neural network structure based on variance significance in output sensitivity [J]. *Control and Decision*, 2015, 30(8): 1393 – 1398.)

(下转第 187 页)