基于 M-estimator 函数的加权深度随机配置网络

丁世飞[1),2)] 张成龙[1)] 郭丽丽[1),2)] 张 健[1),2)] 丁 玲[3)]

[1)](中国矿业大学计算机科学与技术学院 江苏 徐州 221116)

[2)](矿山数字化教育部工程研究中心(中国矿业大学) 江苏 徐州 221116)

[3)](天津大学智能与计算学部 天津 300350)

摘 要 深度随机配置网络（Deep Stochastic Configuration Network, DSCN）是一种增量式随机化学习模型，具有人为干预程度低、学习效率高和泛化能力强等优点. 但是，面向噪声数据回归与分析时，传统的 DSCN 易受到异常值影响，从而降低了模型的泛化性. 因此，为提高噪声数据回归的精度和鲁棒性，提出了基于 M-estimator 函数的加权深度随机配置网络（Weighted Deep Stochastic Configuration Networks, WDSCN）. 首先，选取 Huber 和 Bisquare 2 个常用的 M-estimator 函数计算样本权重，利用加权最小二乘法和 L_2 正则化策略替代最小二乘来更新 WDSCN 输出权重，以降低异常值对 WDSCN 的负面影响；其次，为提高 WDSCN 模型表征能力，设计了一种随机配置稀疏自编码器（Stochastic Configuration Sparse Autoencoder, SC-SAE），SC-SAE 基于 DSCN 其独有的监督机制随机分配输入参数，采用基于 L_1 正则化的目标函数，并利用交替方向乘子法（Alternating Direction Method of Multipliers, ADMM）计算 SC-SAE 输出权重；然后，为获取有效的特征表示，利用 SC-SAE 生成特征的随机性和多样性，采用多个 SC-SAE 进行特征学习并融合，用于 WDSCN 模型训练；最后，在真实数据集上的实验结果表明，WDSCN-Huber、WDSCN-Bisquare 相比于 DSCN、SCN 以及 RSC-KDE、RSC-Huber、RSC-IQR、RSCN-KDE、WBLS-KDE 和 RBLS-Huber 等加权模型具有更高的泛化性能和回归精度.

关键词 深度随机配置网络；异常数据；鲁棒性；回归；随机神经网络

中图法分类号 TP183 **DOI 号** 10.11897/SP.J.1016.2023.02476

Weighted Deep Stochastic Configuration Networks Based on M-estimator Functions

DING Shi-Fei[1),2)] ZHANG Cheng-Long[1)] GUO Li-Li[1),2)] ZHANG Jian[1),2)] DING Ling[3)]

[1)](*School of Computer Science and Technology, China University of Mining and Technology, Xuzhou, Jiangsu* 221116)

[2)](*Mine Digitization Engineering Research Center of Ministry of Education (China University of Mining and Technology), Xuzhou, Jiangsu* 221116)

[3)](*College of Intelligence and Computing, Tianjin University, Tianjin* 300350)

Abstract Deep stochastic configuration network (DSCN) is an randomized incremental learning model, it can start from a small structure, increase the nodes and hidden layers gradually. As the input weights and biases of nodes are assigned according to supervisory mechanism, meantime, all the nodes in hidden layer are fully connected to the outputs, the output weights of DSCN are determined through the least square method. Therefore, DSCN has the advantages of less manual intervention, high learning efficiency, strong generalization ability. However, although the randomized feedforward learning process of DSCN has faster efficiency, the feature learning ability is still insufficient. In the meantime, with the increase of nodes and

收稿日期：2022-10-18；在线发布日期：2023-03-17. 本课题得到国家自然科学基金（62276265, 61976216, 62206297, 61672522）资助.
丁世飞，博士，教授，中国计算机学会（CCF）高级会员，主要研究领域为智能信息处理、模式识别、机器学习、数据挖掘、粒计算. E-mail: dingsf@cumt.edu.cn. **张成龙**（通信作者），博士研究生，中国计算机学会（CCF）学生会员，主要研究领域为随机配置网络、随机化学习方法. E-mail: zhangchl1992@qq.com. **郭丽丽**，博士，讲师，中国计算机学会（CCF）会员，主要研究领域为多模态情感计算、深度学习. **张 健**（通信作者），博士，讲师，中国计算机学会（CCF）会员，主要研究领域为深度学习、多标签学习. E-mail: zhangjian-10231209@cumt.edu.cn. **丁 玲**，博士研究生，中国计算机学会（CCF）学生会员，主要研究领域为图机器学习、深度聚类.

hidden layers, it is easy to lead to overfitting phenomenon. When solving regression problems with noise, the performance of original DSCN is easily affected by outliers, which reduces the generalization ability of the model. Therefore, to improve the regression performance and robustness of DSCN, weighted deep stochastic configuration networks (WDSCN) based on M-Estimator functions are proposed. First of all, we adopt two common M-estimator functions (i.e., Huber and Bisquare) to acquire the sample weights for reducing the negative impact of outliers. When the sample has a smaller training error, give this sample a larger weight, while when the training error of sample is larger, it is determined to be outlier data and give this sample a smaller weight. The sample weight decreases monotonically with the increase of the absolute value of the error, thus reducing the influence of noisy data onto the model and improving the generalization of the algorithm. Meanwhile, the weighted least square method and L_2 regularization strategy are introduced to calculate output weight vector replace the least square method. It can not only solve the noisy data regression problems and avoid over-fitting problem of DSCN. In the second place, the model based on L_1 regularization is helpful to extract sparse features and improve the accuracy of supervised learning, for further improve the representation ability of WDSCN, a stochastic configuration sparse autoencoder (SC-SAE) is designed, SC-SAE use the supervision mechanism of DSCN to assign input parameters, at the same time, we adopt the L_1 regularization technique to objective function for getting sparse features, alternating direction method of multipliers (ADMM) approach is utilized to solve the objective function for determining the output weights of SC-SAE. And then, as the randomness encoding process of SC-SAE, we can obtain the diversity of features of different SC-SAE models, consequently effective feature representation can be acquired through fusion features from multiple SC-SAE for the training of WDSCN. Finally, experimental results on real-world datasets show that the proposed WDSCN-Huber and WDSCN-Bisquare have higher generalization performances and regression accuracies than DSCN, SCN, and other weighted models (e.g., RSC-KDE, RSC-Huber, RSC-IQR, RDSCN-KDE, WBLS-KDE and RBLS-Huber). But in the meantime, the results of ablation experiment show that WDSCN with fusion sparse features which exacted from multiple different SC-SAE models are superior to those models with fusion sparse feature. Therefore, it is verified that SC-SAE can extract effective sparse features and improve the learning ability of weighted models.

Keywords　deep stochastic configuration network; noisy data; robustness; regression; random neural network

1　引　言

近年来，随机神经网络（Random Neural Network, RNN）通过随机分配模型参数，利用最小二乘法计算输出权重，提高了神经网络学习效率[1-3]. 但是，通常情况下该类神经网络的通用逼近性与随机参数的设置范围和网络节点的数量密切相关[4-6]. 随机配置网络（Stochastic Configuration Network, SCN）由 Wang 于 2017 年提出，是一种快速、高效、精准的随机化建模方法[7]. SCN 采取增量学习方式，引入监督机制，根据训练样本配置隐含层节点参数，构建网络结构，保证了网络的学习精度和学习效率，受到国内外学者的青睐. 随后，Li 和 Wang 提出了具有矩阵输入的 2 维随机配置网络（2-D stochastic configuration network, 2DSCN）用于处理图像数据[8]. Dai 等提出了基于块增量学习的随机配置网络（Stochastic Configuration Network with Block Increments, Bi-SCN），实现了 SCN 隐含层节点的批量配置[9]. Zhu 等设计了 2 种新的不等式约束用于隐含层节点分配，减少了节点配置的时间[10]. Zhang 和 Ding 提出了一种基于混沌麻雀搜索算法的随机配置网络，使用群体智能优化算法选择最佳权重和偏置的控制因子，提高了网络的训练精度[11]. 鉴于 SCN 网络参数配置过程的人为干预程度低且具有较高的泛化性能，SCN 被广泛应用于电力大数据处理[12]、工业过程建模[13]、振动信号故障诊断[14]、在线测量

问题[15]等领域.

在实际工程应用中,传感器容易受到设备故障、人为干扰、工作环境等因素的影响，采集的数据中存在不同程度的噪声和异常值，从而降低了学习模型的泛化性[16]. 为了有效解决噪声数据回归问题，Wang 和 Li 首先提出了一种基于核密度估计的鲁棒随机配置网络（Robust Stochastic Configuration Network with Kernel Density Estimation, RSC- KDE）,利用核密度估计（Kernel Density Estimation, KDE）为训练样本赋予不同的惩罚权重，并采用加权最小二乘法来确定输出权值，以降低噪声数据或异常值的负面影响[17]. 随后，Xie 和 Zhou 利用 RSC-KDE 对高炉炼铁的铁水质量预测问题进行了多输出建模[18]. Li 等人提出了基于最大相关熵准则的鲁棒随机配置网络（Robust Stochastic Configuration Network with Maximum Correntropy Criterion, RSC-MCC），采用最大相关熵准则（Maximum Correntropy Criterion, MCC）作为损失函数，并通过半二次法优化每个训练样本的惩罚权值，以减弱噪声数据或异常值对整个训练过程的影响[19]. 为解决赤铁矿磨矿粒度估计问题，Dai 等基于 SCN 引入了 Huber、IQR、非参数核密度估计（Nonparametric Kernel Density Estimation, NKDE）作为损失函数，并分别提出了 RSC-Huber、RSC-IQR、RSC-NKDE 3 种鲁棒随机配置网络模型[20]. 此外,Wu 等人提出了用于处理不确定数据的贝叶斯随机配置网络（Bayesian Stochastic Configuration Network, BSCN），通过引入贝叶斯推理（Bayesian Inference, BI）计算 SCN 的输出权值，为处理噪声数据提供了一种新的解决方案[21].

深度神经网络（Deep Neural Network, DNN）能够提取高阶或多级特征，相比于浅层模型具有更强的特征表征能力，在机器视觉、模式识别、自然语言处理等领域展现了巨大潜力. Wang等人将SCN扩展深度随机配置网络（Deep Stochastic Configuration Network, DSCN）用来进行图像数据建模分析，展现出其在图像处理领域的发展潜力[22]. 为提高 DSCN 性能，Felicetti 和 Wang 采用对称零均值化分布构建 DSCN，提高了模型泛化性[23]，与此同时，其还利用蒙特卡洛树搜索和随机搜索策略来寻找模型超参数[24]. 集成学习能够显著提高神经网络的精度以及泛化性能，Zhang 等人提出了一种基于 AdaBoost 算法的 DSCN 集成模型,提高了 DSCN 回归性能[25]. 目前,在噪声数据分析方面,除基于KDE的鲁棒深度随机配置网络 （Robust Deep Stochastic Configuration Network Based on Kernel Density Estimation, RDSCN-KDE）外，DSCN 尚未用于处理噪声数据回归[26].

除此之外，DSCN 作为随机神经网络模型，仍存在特征学习能力不足的缺点，且随着节点数量的增加,存在模型过拟合的风险. 因此,为挖掘 DSCN 在噪声数据分析中的潜力,有效地消除噪声的影响,提高 DSCN 的特征学习能力和泛化性能，本文提出了基于 M-estimator 函数的加权深度随机配置网络（Weighted Deep Stochastic Configuration Networks, WDSCN）. WDSCN 引入 L_2 正则化策略[27, 28]，采取具有鲁棒特性的M-estimator函数作为DSCN加权函数[29, 30]. 与此同时，为提高模型特征学习能力，设计了一种随机配置稀疏自编码器（Stochastic Configuration Sparse Autoencoder, SC-SAE）提取稀疏特征用于 WDSCN 模型训练[7, 31].

本文主要贡献如下：

（1）分别选取 Huber 和 Bisquare 2 个常用的 M-estimator 函数计算样本权重，利用加权最小二乘法和 L_2 正则化策略替代最小二乘法更新 WDSCN 输出权重，以降低异常值对 WDSCN 的负面影响；

（2）设计了一种新型随机配置稀疏自编码器（SC-SAE），基于 SCN 其独有的监督机制随机分配输入参数，采用基于 L_1 正则化的目标函数，并利用交替方向乘子法（Alternating Direction Method of Multipliers, ADMM）[32]计算 SC-SAE 输出权重；

（3）为获取有效的特征表示，利用 SC-SAE 生成特征的随机性和多样性，采取多个 SC-SAE 进行特征学习并融合，用于 WDSCN 模型训练；

（4）在不同离群值比例下的 KEEL 回归数据集上的实验结果表明，WDSCN 相比于 DSCN 具有更强的鲁棒性和泛化性，且优于 RSC-KDE[17]、RSC-Huber[20]、RSC-IQR[20]、RDSCN-KDE[26]、WBLS-KDE[33]和 RBLS-Huber[34]等鲁棒模型.

2 深度随机配置网络

DSCN 提出于 2018 年，是 SCN 的扩展，其节点配置流程与 SCN 相同，网络结构由一个隐含层的一个节点开始，逐渐增加节点数量和网络层次，并将所有节点全连接至输出，相比于 SCN 模型，具有更强的特征表征能力，其网络结构如图 1 所示，能够充分利用多级特征进行最终决策[22].

为简化描述，DSCN 隐含层数量设置为 M，各隐含层节点数量统一设置为 $L_{\max}$，则其构建过程可以描述如下：

给定训练数据 $\{\boldsymbol{X},\boldsymbol{Y}\}$，其中 $\boldsymbol{X}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N\}$ 代表样本特征，$\boldsymbol{x}_i=[x_{i,1},x_{i,2},\cdots,x_{i,d}]\in\mathbb{R}^d$；$\boldsymbol{Y}=\{\boldsymbol{y}_1,\boldsymbol{y}_2,\cdots,\boldsymbol{y}_N\}$ 代表样本标记，$\boldsymbol{y}_i=[y_{i,1},y_{i,2},\cdots,y_{i,m}]\in\mathbb{R}^m$；$i=1,2,\cdots,N$，$N$ 代表样本数量.

Step 1: 设定目标函数 $f:\mathbb{R}^d\to\mathbb{R}^m$，假设 M 层的 $L-1$ 个节点已经生成，根据式（1）计算当前残差：

$$\boldsymbol{e}_{L-1}^M=\boldsymbol{f}-\boldsymbol{f}_{L-1}^M(\boldsymbol{X})=[\boldsymbol{e}_{L-1,1}^M(\boldsymbol{X}),\boldsymbol{e}_{L-1,2}^M(\boldsymbol{X}),\cdots,\boldsymbol{e}_{L-1,m}^M(\boldsymbol{X})] \tag{1}$$

式中，$\boldsymbol{e}_{L-1,q}^M(\boldsymbol{X})=[\boldsymbol{e}_{L-1,q}^M(\boldsymbol{x}_1),\boldsymbol{e}_{L-1,q}^M(\boldsymbol{x}_2),\cdots,\boldsymbol{e}_{L-1,q}^M(\boldsymbol{x}_N)]^T\in\mathbb{R}^N$，$q=1,2,\cdots,m$.

Step 2: 若 $\|\boldsymbol{e}_{L-1}^M\|^2$ 不满足预设误差 ε 或未达到该层最大节点数 $L_{\max}$，随机分配节点 L 候选参数，根据式（3）计算当前节点输出 $\boldsymbol{h}_L^M$：

$$\begin{aligned}\boldsymbol{X}^{(M)}&=\Phi(\boldsymbol{X}^{(M-1)};\boldsymbol{W}^{(M-1)},\boldsymbol{B}^{(M-1)})\\&=[\phi_{M,1}(\boldsymbol{X}^{(M-1)}),\phi_{M,2}(\boldsymbol{X}^{(M-1)}),\cdots,\phi_{M,L}(\boldsymbol{X}^{(M-1)})],M\geqslant 1\end{aligned} \tag{2}$$

$$\boldsymbol{h}_L^M=[\phi_{M,L}(\boldsymbol{x}_1^{M-1}),\phi_{M,L}(\boldsymbol{x}_2^{M-1})\cdots,\phi_{M,L}(\boldsymbol{x}_N^{M-1})]^T \tag{3}$$

式中，$\phi(\cdot)$ 代表激活函数，$\boldsymbol{X}^{(0)}=\boldsymbol{X}=[\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N]$，$\boldsymbol{W}^{M-1}=[\boldsymbol{w}_1^{M-1},\boldsymbol{w}_2^{M-1},\cdots,\boldsymbol{w}_L^{M-1}]$，$\boldsymbol{B}^{M-1}=[\boldsymbol{b}_1^{M-1},\boldsymbol{b}_2^{M-1},\cdots,\boldsymbol{b}_L^{M-1}]$，$\boldsymbol{w}_L^M$ 和 $\boldsymbol{b}_L^M$ 分别代表 M 层节点 L 的输入权重和偏置.

Step 3: 根据式（4）选择满足 $\xi_L^M=\sum_{q=1}^m\xi_{L,q}^M\geqslant 0$ 最大值的候选参数为 M 层节点 L 的参数：

$$\xi_{L_n,q}^M=\frac{\left\langle\boldsymbol{e}_{L-1,q}^M,\boldsymbol{h}_L^M\right\rangle^2}{\left\langle\boldsymbol{h}_L^M,\boldsymbol{h}_L^M\right\rangle}-(1-r)\left\langle\boldsymbol{e}_{L-1,q}^M,\boldsymbol{e}_{L-1,q}^M\right\rangle \tag{4}$$

式中，$q=1,2,\cdots,m$；$r\in(0,1)$.

Step 4: 根据式（5）计算网络输出权重：

$$\boldsymbol{\beta}=\arg\min_{\boldsymbol{\beta}}\|\boldsymbol{H}\boldsymbol{\beta}-\boldsymbol{Y}\|_2^2=\boldsymbol{H}^{+}\boldsymbol{Y} \tag{5}$$

式中，$\boldsymbol{H}^{+}$ 表示 $\boldsymbol{H}$ 的摩尔-彭若斯（Moore-Penrose）广义逆，$\boldsymbol{H}=[\boldsymbol{H}_L^1,\boldsymbol{H}_L^2,\cdots,\boldsymbol{H}_L^M]$ 表示所有隐含层节点输出，$\boldsymbol{H}_L^M=[\boldsymbol{h}_1^M,\boldsymbol{h}_2^M,\cdots,\boldsymbol{h}_L^M]$ 表示隐含层 M 的输出矩阵.

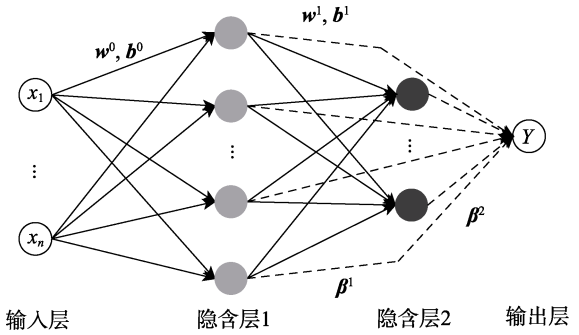

图 1　DSCN 模型结构[22]

前馈神经网络的随机化建模方法会随着网络层次和节点数量的增加，导致模型产生过拟合问题. 为提高模型泛化性，借鉴岭回归理论，文献[27]和文献[28]基于 SCN 节点配置方式，在目标函数中引入 L_2 正则化项. 为提高 DSCN 的泛化性，式（5）可被修正为

$$J(\boldsymbol{\beta})=\arg\min_{\boldsymbol{\beta}}\|\boldsymbol{H}\boldsymbol{\beta}-\boldsymbol{Y}\|_2^2+C_2\|\boldsymbol{\beta}\|_2^2 \tag{6}$$

通过对式（6）进行求偏导可得式（8）：

$$\frac{\partial J}{\partial\boldsymbol{\beta}}=2\boldsymbol{H}^T\boldsymbol{H}\boldsymbol{\beta}-2\boldsymbol{H}^T\boldsymbol{Y}+2C_2\boldsymbol{\beta}=0 \tag{7}$$

$$\boldsymbol{\beta}=(\boldsymbol{H}^T\boldsymbol{H}+C_2\boldsymbol{I})^{-1}\boldsymbol{H}^T\boldsymbol{Y} \tag{8}$$

式中，C_2 代表 L_2 正则化参数.

3　加权深度随机配置网络

3.1　WDSCN框架

面向噪声数据分析时，采用式（5）传统的最小二乘法计算训练误差易因离群值严重降低模型的泛化性. 为了降低异常值对模型泛化性的影响，本节引入加权最小二乘法和正则化策略作为 WDSCN 目标函数，采用基于 M-estimator 函数的加权函数计算样本权重，更新输出权重.

WDSCN 具体加权过程如下[27-30]：

Step 1: 给定目标函数 $f:\mathbb{R}^d\to\mathbb{R}^m$，根据式（1）当前误差和初始化加权矩阵 $\boldsymbol{\theta}=diag(\theta_1,\theta_2,\cdots,\theta_N)$，定义加权误差为

$$\boldsymbol{e}_{L-1}^{M\prime}=\boldsymbol{\theta}\boldsymbol{e}_{L-1}^M(\boldsymbol{X}) \tag{9}$$

式中，$\boldsymbol{e}_{L-1,q}^{M\prime}(\boldsymbol{X})=[\boldsymbol{e}_{L-1,q}^{M\prime}(\boldsymbol{x}_1),\boldsymbol{e}_{L-1,q}^{M\prime}(\boldsymbol{x}_2),\cdots,\boldsymbol{e}_{L-1,q}^{M\prime}(\boldsymbol{x}_N)]^T\in\mathbb{R}^N$，$q=1,2,\cdots,m$.

Step 2: 若 $\|\boldsymbol{e}_{L-1}^{M\prime}\|^2$ 不满足预设误差 ε 或未达到该层最大节点数 $L_{\max}$，随机分配节点 L 候选参数，根据式（15）节点输出 $\boldsymbol{h}_L^M$，定义加权输出为

$$\boldsymbol{h}_L^{M\prime}=\boldsymbol{\theta}\boldsymbol{h}_L^M(\boldsymbol{X}) \tag{10}$$

Step 3: 根据式（11）选择满足 $\xi_L^{M\prime}=\sum_{q=1}^m\xi_{L,q}^{M\prime}\geqslant 0$ 最大值的候选参数为 M 层节点 L 的参数：

$$\xi_{L,q}^{M\prime}=\frac{\left\langle\boldsymbol{e}_{L-1,q}^{M\prime},\boldsymbol{h}_L^{M\prime}\right\rangle^2}{\left\langle\boldsymbol{h}_L^{M\prime},\boldsymbol{h}_L^{M\prime}\right\rangle}-(1-r)\left\langle\boldsymbol{e}_{L-1,q}^{M\prime},\boldsymbol{e}_{L-1,q}^{M\prime}\right\rangle \tag{11}$$

式中，$q=1,2,\cdots,m$；$r\in(0,1)$.

Step 4: 根据式（12）计算网络输出权重：

$$\boldsymbol{\beta}=\arg\min_{\boldsymbol{\beta}}\|\boldsymbol{\theta}\boldsymbol{H}\boldsymbol{\beta}-\boldsymbol{\theta}\boldsymbol{Y}\|_2^2=(\boldsymbol{H}^T\boldsymbol{\theta}^2\boldsymbol{H})^{-1}\boldsymbol{H}^T\boldsymbol{\theta}^2\boldsymbol{Y} \tag{12}$$

式中，$\boldsymbol{H}=[\boldsymbol{H}_L^1,\boldsymbol{H}_L^2,\cdots,\boldsymbol{H}_L^M]$ 表示所有隐含层节点输出，$\boldsymbol{H}_L^M=[\boldsymbol{h}_1^M,\boldsymbol{h}_2^M,\cdots,\boldsymbol{h}_L^M]$ 表示隐含层 M 的输出矩阵.

与此同时，为避免模型过拟合，基于式（13）引入 L_2 正则化项，目标函数可被修正为[27, 28]

$$J(\boldsymbol{\beta})=\arg\min_{\boldsymbol{\beta}}\|\boldsymbol{\theta H\beta}-\boldsymbol{\theta Y}\|_2^2+C_2\|\boldsymbol{\beta}\|_2^2 \tag{13}$$

通过对式（13）求偏导置 0 可得式（15）：

$$\frac{\partial J}{\partial \boldsymbol{\beta}}=2\boldsymbol{H}^T\boldsymbol{\theta}^2\boldsymbol{H\beta}-2\boldsymbol{H}^T\boldsymbol{\theta}^2\boldsymbol{Y}+2C_2\boldsymbol{\beta}=0 \tag{14}$$

$$\boldsymbol{\beta}=(\boldsymbol{H}^T\boldsymbol{\theta}^2\boldsymbol{H}+C_2\boldsymbol{I})^{-1}\boldsymbol{H}^T\boldsymbol{\theta}^2\boldsymbol{Y} \tag{15}$$

式中，C_2 代表正则化参数，$\boldsymbol{I}$ 表示单位矩阵. 相比于文献[17, 19, 20, 26]权重加权方式，增加了正则化策略，有利于减少过拟合风险，$\boldsymbol{\theta}^2$ 项加权系数能够更好地降低噪声带来的负面影响.

此外，$\boldsymbol{\beta}$ 可通过式（16）交替优化策略迭代更新，选取最大以获取更高的回归精度.

$$\boldsymbol{\beta}_t=(\boldsymbol{H}^T\boldsymbol{\theta}_{t-1}^2\boldsymbol{H}+C_2\boldsymbol{I})^{-1}\boldsymbol{H}^T\boldsymbol{\theta}_{t-1}^2\boldsymbol{Y},\ t=1,2,\cdots,T \tag{16}$$

其中，t 表示迭代次数，T 代表最大迭代次数.

最后，WDSCN 加权流程图如图 2 所示.

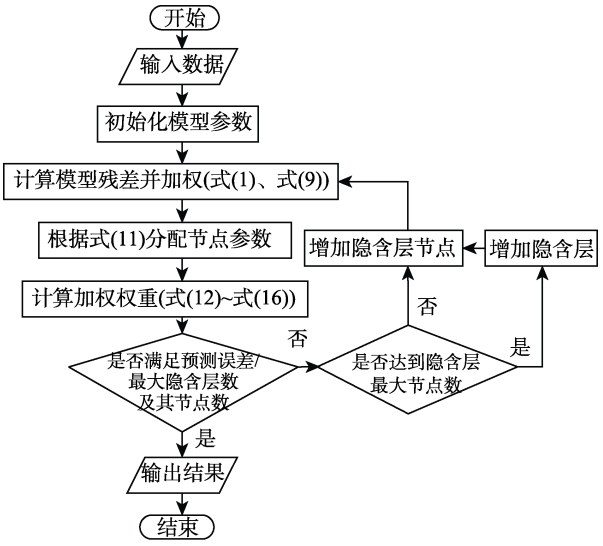

图 2 WDSCN 加权流程图

3.2 M-estimator加权惩罚因子

为保证 WDSCN 的鲁棒性，加权函数的选择需要保证权重系数能够伴随训练误差的增加而单调递减，模型训练误差过大时，赋予较小的权重系数，以降低学习模型对于离群值的敏感性. 因此，本节选取 Huber、Bisquare 2 个常用的基于 M-estimator 函数的加权函数，如表 1 所示，相关函数曲线见图 3[29, 30]. 如图 3 所示，(a)为 Huber 加权函数，(b)为 Bisquare 加权函数. 当样本训练误差较小时，能够赋予样本较大的权重；当样本训练误差较大时，判定其为离群数据，具有较小的加权系数. 样本权重随着误差的绝对值增加而单调递减，从而减小异常数据对模型的影响，提高算法泛化性.

表 1 M-estimator 函数

函数名称	加权函数 $\boldsymbol{\theta}(\boldsymbol{e}_i)$	默认 *tuning*
Huber	$\min\left(1,\frac{k}{\lvert\boldsymbol{e}_i\rvert}\right)$	1.345
Bisquare	$\left[1-\left(\frac{\boldsymbol{e}_i}{k}\right)^2\right]^2,\lvert\boldsymbol{e}_i\rvert\leqslant k$	4.685

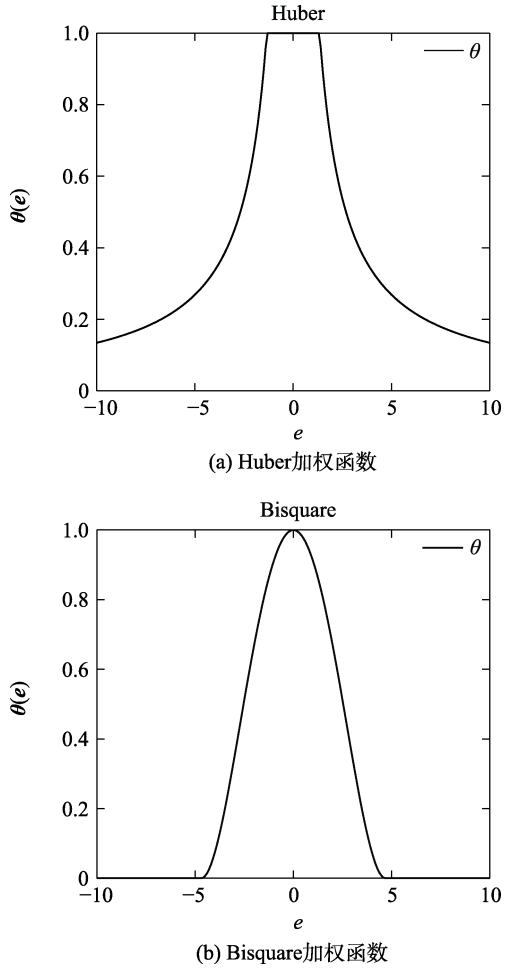

图 3 M-estimator 函数图

与此同时，为确保模型能够充分学习和训练，提高噪声数据回归性能，通常采用式（17）计算阈值 k：

$$k=tuning\times\sigma \tag{17}$$

式中，*tuning* 表示加权函数的默认调节参数；σ 表示训练误差 e 的标准差，在异常值情况下可采用式（18）表示：

$$\sigma=med(|\boldsymbol{e}|)/0.6745 \tag{18}$$

式中，med(·)代表取误差值的中位数，同时引入常量保证了 σ 在高斯误差条件下的无偏性.

3.3　基于SC-SAE的稀疏特征融合

3.3.1　随机配置稀疏自编码器

良好的特征表示能够提高分类和回归问题的性能，为了提取有效的特征，基于稀疏特征的学习模型得到了广泛的应用[35, 36]. SCN 采取增量学习方式，利用不等式约束随机配置隐含层节点参数，网络结构由一个隐含层节点开始逐渐增加，直至完成网络构建，具有良好的通用逼近特性[7]. 因此，利用 SCN 的通用逼近特性，设计了一种随机配置稀疏自编码器(Stochastic Configuration Sparse Autoencoder, SC-SAE ）来提取原始数据的稀疏特征. 其中，SC-SAE 结构如图 4 所示，分为编码和解码过程，编码过程采用 SCN 监督机制随机分配输入节点参数，基于 L_1 正则化参数的目标函数进行特征重构，获取稀疏特征.

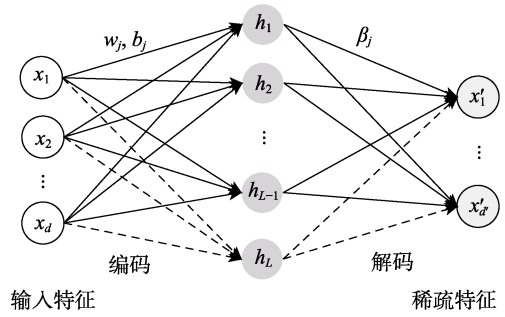

图 4　随机配置稀疏自编码器

SC-SAE 网络结构如图 4 所示，其构建过程可以描述如下：

给定训练样本特征 $\boldsymbol{X}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N\}$，其中 $\boldsymbol{x}_i=[x_{i,1},x_{i,2},\cdots,x_{i,d}]\in\mathbb{R}^d$，$d$ 表示原始特征维度；$i=1,2,\cdots,N$，N 代表样本数量.

Step 1：给定目标函数 $f:\mathbb{R}^d\to\mathbb{R}^{d'}$，$d'$ 表示重构特征维度，假设 SC-SAE 的 $L-1$ 个隐含层节点已经配置完成，则当前网络输出 $\boldsymbol{f}_{L-1}(\boldsymbol{X})$ 可由式（19）计算：

$$\boldsymbol{f}_{L-1}(\boldsymbol{X})=\sum_{j=1}^{L-1}g_j(\boldsymbol{X}\boldsymbol{w}_j+\boldsymbol{b}_j)\boldsymbol{\beta}_j\ (L=1,2,\cdots,f_0=0) \tag{19}$$

式中，$\boldsymbol{\beta}_j$ 代表节点 j 的输出权重；g_j 代表激活函数；$\boldsymbol{w}_j$ 和 $\boldsymbol{b}_j$ 分别代表节点 j 的输入权重和偏置，其范围分别在 $[-\lambda,\lambda]^d$ 和 $[-\lambda,\lambda]$，λ 代表参数范围；$j=1,2,\cdots,L_{\max}$.

Step 2：当前网络残差向量可通过式（20）计算：

$$\boldsymbol{e}_{L-1}=\boldsymbol{f}-\boldsymbol{f}_{L-1}(\boldsymbol{X})=[\boldsymbol{e}_{L-1,1}(\boldsymbol{X}),\boldsymbol{e}_{L-1,2}(\boldsymbol{X}),\cdots,\boldsymbol{e}_{L-1,d'}(\boldsymbol{X})] \tag{20}$$

Step 3：根据预设误差 ε 或设定最大节点数 $L_{\max}$，判断是否满足预设条件.如未满足预设条件，自适应增加节点 L，并根据式（21）配置节点参数：

$$\xi_{L,q}=\frac{\left\langle \boldsymbol{e}_{L-1,q},\boldsymbol{h}_L\right\rangle^2}{\left\langle \boldsymbol{h}_L,\boldsymbol{h}_L\right\rangle}-(1-r-\mu_L)\left\langle \boldsymbol{e}_{L-1,q},\boldsymbol{e}_{L-1,q}\right\rangle \tag{21}$$

$$\boldsymbol{h}_L=[g_L(\boldsymbol{w}_L{}^T\boldsymbol{x}_1+\boldsymbol{b}_L),\cdots,g_L(\boldsymbol{w}_L{}^T\boldsymbol{x}_N+\boldsymbol{b}_L)]^T \tag{22}$$

式中，$q=1,2,\cdots,d'$，代表输出维度；$\boldsymbol{h}_L$ 代表节点 L 的输出，由式（22）计算；$\boldsymbol{w}_L$ 和 $\boldsymbol{b}_L$ 分别代表待确定的节点 L 参数；$T_{\max}$ 代表候选参数数量；$r\in(0,1)$；$\{\mu_L\}$ 代表非负实数序列，其中 $\mu_L\leqslant 1-r$，$\lim_{L\to+\infty}\mu_L=0$；根据式（21），满足 $\xi_L=\sum_{q=1}^{m}\xi_{L,q}\geqslant 0$ 最大值的候选节点参数作为节点 L 的最终参数.

Step 4：根据式（23）目标函数，确定 SC-SAE 输出权重 $\boldsymbol{\beta}$：

$$\boldsymbol{\beta}=\arg\min_{\boldsymbol{\beta}}\|\boldsymbol{H}\boldsymbol{\beta}-\boldsymbol{X}\|^2+C_1\|\boldsymbol{\beta}\|_1 \tag{23}$$

式中，$\boldsymbol{\beta}=[\boldsymbol{\beta}_1,\boldsymbol{\beta}_2,\cdots,\boldsymbol{\beta}_L]^T$；$\boldsymbol{H}=[\boldsymbol{h}_1,\boldsymbol{h}_2,\cdots,\boldsymbol{h}_L]$ 表示所有隐含层节点输出.

Step 5：分别根据式(24)和式(25),获取 SC-SAE 提取特征和输出 $\boldsymbol{f}$：

$$\boldsymbol{X}'=\boldsymbol{X}\boldsymbol{\beta} \tag{24}$$

$$\boldsymbol{f}=\boldsymbol{X}'\boldsymbol{\beta}^+ \tag{25}$$

式中，$\boldsymbol{\beta}^+$ 表示 $\boldsymbol{\beta}$ 的 Moore-Penrose 广义逆.

为了对式（23）进行求解，采用 ADMM 求解过程如下[32]：

将式（23）转化为式（26）ADMM 形式：

$$\arg\min_{\boldsymbol{\beta}}u(\boldsymbol{\beta})+v(\boldsymbol{o}),\ \text{s.t.}\ \boldsymbol{\beta}-\boldsymbol{o}=0 \tag{26}$$

式中，$u(\boldsymbol{\beta})=\|\boldsymbol{H}\boldsymbol{\beta}-\boldsymbol{X}\|^2$，$v(\boldsymbol{o})=C_1\|\boldsymbol{\beta}\|_1$.

式（26）可通过下述迭代过程进行求解：

$$\begin{cases}\boldsymbol{\beta}_{k+1}=(\boldsymbol{H}^T\boldsymbol{H}+\rho\boldsymbol{I})^{-1}(\boldsymbol{H}^T\boldsymbol{X}+\rho(\boldsymbol{o}^k-\boldsymbol{z}^k))\\ \boldsymbol{o}_{k+1}=S_{\frac{C_1}{\rho}}(\boldsymbol{\beta}_{k+1}+\boldsymbol{z}_k)\\ \boldsymbol{z}_{k+1}=\boldsymbol{z}_k+(\boldsymbol{\beta}_{k+1}-\boldsymbol{o}_{k+1})\end{cases} \tag{27}$$

式中，$\rho>0$；S 表示软阈值运算符，见式（28）.

$$S_\kappa(a)=\begin{cases}a-\kappa, & a>\kappa\\ 0, & |a|\leqslant\kappa\\ a+\kappa, & a<\kappa\end{cases} \tag{28}$$

3.3.2 基于 SC-SAE 的稀疏特征融合的 WDSCN

由于 SC-SAE 随机化特征学习过程，为获取有

效的特征表示，利用 SC-SAE 生成特征的随机性和多样性，采取多个 SC-SAE 进行特征学习并融合，用于 WDSCN 模型训练，基于 SC-SAE 的稀疏特征融合的 WDSCN 流程如图 5 所示.

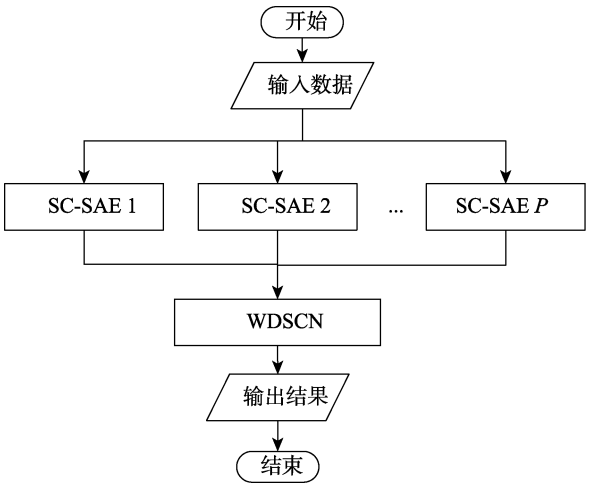

图 5 基于 SC-SAE 的稀疏特征融合的 WDSCN 流程

其中，P 表示 SC-SAE 模型数量，p 为各 SC-SAE 提取特征维度，为更好地进行特征学习，P 和 p 需要根据数据集进行设定.

4 实验与分析

所有实验在 Windows10 64 位操作系统 MATLAB R2019b 环境下进行，计算机配置为：Intel (R) Core (TM) i7-9750H CPU@2.60GHz, 64.00GB RAM，并将 WDSCN-Huber 和 WDSCN-Bisquare 和 DSCN[22]、SCN[7]以及 RSC-KDE[17]、RSC-Huber[20]、RSC-IQR[20]、RDSCN-KDE[26]、WBLS-KDE[33]和 RBLS-Huber[34]等模型进行了对比.

4.1 实验设计

4.1.1 实验数据

为了验证 WDSCN 对于噪声数据回归问题的性能，实验选取 5 个 KEEL（Knowledge Extraction based on Evolutionary Learning）①真实数据集，数据集描述见表 2. 为降低不同特征数据范围对模型的影响，首先，将原始数据集的输入和输出向量归一化为[0, 1]，其中 75%的样本随机作为训练数据，其余样本作为测试数据. 其次，在训练数据的输出值中随机加入 10%、20%、30%和 40%的均匀分布的异常值，范围为[–0.5, 0.5]，因此，具有噪声的训练数据的输出值范围被转化为[–0.5, 1.5].

① Knowledge Extraction based on Evolutionary Learning, http://www. keel. es

表 2 KEEL 实验数据集描述

数据集	特征数	样本数
Abalone	8	4177
Wizmir	9	1461
Concrete	8	1030
Stock	9	950
Mortgage	15	1049

4.1.2 参数设置

为保证实验公平性，6 种模型均采用式（29）所示激活函数，所有实验独立运行 20 次，以式（30）均方根误差（Root Mean Square Error, RMSE）作为评价指标对模型进行评价.

$$S(\Theta)=\frac{1}{1+e^{-\Theta}} \tag{29}$$

$$\mathrm{RMSE}=\sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i-\hat{y}_i)^2} \tag{30}$$

式中，y_i 表示样本实际值；$\hat{y}_i$ 表示回归值；N 表示样本个数.

模型参数设置分别如下:

所有模型节点参数范围为{25, 50, 75, 100}；其中 DSCN、WDSCN-Huber 和 WDSCN-Bisquare 等模型采取 2 个隐含层；参数 r 和控制因子 λ 分别由集合{0.9, 0.99, 0.999, 0.9999, 0.99999, 0.999999}和{0.5, 1, 5, 10, 30, 50, 100, 150, 200, 250}确定；最大候选池大小 $T_{\max}$ 设置为 100；预设误差设置为 0.001.

与 DSCN 和 RDSCN-KDE 基础的深度模型不同，WDSCN-Huber 和 WDSCN-Bisquare 引入了基于 SC-SAE 的稀疏特征学习与融合，特征学习阶段，SC-SAE 模型数量和各 SC-SAE 提取特征维度从{1, 2, 3, …, 20}范围内进行筛选. 与此同时，WBLS-KDE 和 RBLS-Huber 特征映射节点数量和映射次数仍在{1, 2, 3, …, 20}范围内进行筛选.

需要特别指出的是，为避免模型过拟合，DSCN、RDSCN-KDE、WDSCN-Huber 和 WDSCN-Bisquare、WBLS-KDE 和 RBLS-Huber 采用了正则化参数，参数范围为{2^{-30}, 2^{-25}, 2^{-20}, 2^{-15}, 2^{-10}, 2^{-5}, 2^{0}}，由网格搜索确定，其余模型保持与参考文献同样设置.

为区分对比模型差异，表 3 对各模型差异化进行了分析：

4.2 实验结果分析

WDSCN-Huber、WDSCN-Bisquare 和相关对比算法在 5 个真实数据集上的平均测试误差分别见表 4~表 8.

表 3　各模型差异化分析

对比模型	正则化策略	稀疏特征学习融合	监督机制参数分配
SCN [7]	×	×	√
RSC-IQR [20]	×	×	√
RSC-Huber [20]	×	×	√
RSC-KDE [17]	×	×	√
DSCN [22]	√	×	√
RDSCN-KDE [26]	√	×	√
WBLS-KDE [33]	√	√(SAE)	×
RBLS-Huber [34]	√	√(SAE)	×
WDSCN-Bisquare	√	√(SC-SAE)	√
WDSCN-Huber	√	√(SC-SAE)	√

注：为避免模型过拟合，本文为 DSCN 和 RDSCN-KDE 增加正则化，文献[22]和文献[26]未涉及.

表 4　10 种模型在 Abalone 数据集上的测试性能

算法	不同比例噪声下的测试性能 (RMSE)				
	0	10%	20%	30%	40%
SCN [7]	0.0762	0.0780	0.0785	0.0783	0.0808
RSC-IQR [20]	0.0780	0.0773	0.0776	0.0779	0.0778
RSC-Huber [20]	0.0768	0.0772	0.0763	0.0761	0.0783
RSC-KDE [17]	0.0783	0.0779	0.0782	0.0775	0.0775
DSCN [22]	0.0754	0.0762	0.0769	0.0760	0.0773
RDSCN-KDE [26]	0.0759	0.0750	0.0764	0.0765	0.0771
WBLS-KDE [33]	0.0756	0.0751	0.0759	0.0760	0.0764
RBLS-Huber [34]	0.0752	0.0747	0.0754	0.0754	0.0758
WDSCN-Bisquare	**0.0736**	**0.0740**	**0.0743**	**0.0744**	0.0754
WDSCN-Huber	**0.0736**	**0.0741**	0.0744	**0.0744**	**0.0746**

表 5　10 种模型在 Wizmir 数据集上的测试性能

算法	不同比例噪声下的测试性能 (RMSE)				
	0	10%	20%	30%	40%
SCN [7]	0.0193	0.0295	0.0350	0.0452	0.0507
RSC-IQR [20]	0.0185	0.0194	0.0204	0.0210	0.0228
RSC-Huber [20]	0.0194	0.0202	0.0210	0.0222	0.0255
RSC-KDE [17]	0.0185	0.0195	0.0203	0.0223	0.0231
DSCN [22]	0.0185	0.0226	0.0240	0.0254	0.0268
RDSCN-KDE [26]	0.0180	0.0182	0.0190	0.0196	0.0212
WBLS-KDE [33]	0.0178	0.0188	0.0191	0.0200	0.0219
RBLS-Huber [34]	0.0177	0.0183	0.0189	0.0205	0.0241
WDSCN-Bisquare	**0.0172**	**0.0172**	**0.0178**	**0.0183**	**0.0196**
WDSCN-Huber	0.0173	0.0173	0.0179	0.0184	**0.0196**

通过表 4~表 8 可得，在噪声比例为 0%的情况下，本文提出的 WDSCN 在 Abalone、Wizmir、Concrete 和 Mortgage 等 4 个数据集上取得了最优结果，在 Stock 数据集中，WDSCN-Bisquare 测试性能和 DSCN 相同，均为最佳，说明了基于稀疏特征融合的 WDSCN 适用于无异常值的原始回归数据集.

表 6　10 种模型在 Concrete 数据集上的测试性能

算法	不同比例噪声下的测试性能 (RMSE)				
	0	10%	20%	30%	40%
SCN [7]	0.0882	0.0982	0.1039	0.1051	0.1080
RSC-IQR [20]	0.0923	0.0923	0.0996	0.1057	0.1115
RSC-Huber [20]	0.0875	0.0939	0.0995	0.1036	0.1100
RSC-KDE [17]	0.0909	0.0946	0.0987	0.1007	0.1066
DSCN [22]	0.0805	0.0927	0.0967	0.0999	0.1006
RDSCN-KDE [26]	0.0788	0.0833	0.0889	0.0930	0.0948
WBLS-KDE [33]	0.0829	0.0873	0.0921	0.0958	0.0996
RBLS-Huber [34]	0.0827	0.0861	0.0909	0.0954	0.1001
WDSCN-Bisquare	0.0750	**0.0788**	0.0838	0.0909	0.0953
WDSCN-Huber	**0.0748**	0.0799	**0.0835**	**0.0890**	**0.0941**

表 7　10 种模型在 Stock 数据集上的测试性能

算法	不同比例噪声下的测试性能 (RMSE)				
	0	10%	20%	30%	40%
SCN [7]	0.0318	0.0483	0.0518	0.0554	0.0618
RSC-IQR [20]	0.0349	0.0361	0.0398	0.0462	0.0557
RSC-Huber [20]	0.0340	0.0391	0.0436	0.0478	0.0544
RSC-KDE [17]	0.0352	0.0386	0.0417	0.0479	0.0513
DSCN [22]	**0.0290**	0.0437	0.0477	0.0517	0.0559
RDSCN-KDE [26]	0.0297	0.0340	0.0370	0.0409	0.0447
WBLS-KDE [33]	0.0321	0.0369	0.0395	0.0417	0.0458
RBLS-Huber [34]	0.0320	0.0351	0.0388	0.0430	0.0485
WDSCN-Bisquare	**0.0290**	**0.0314**	**0.0348**	0.0396	0.0448
WDSCN-Huber	0.0293	0.0317	0.0349	**0.0385**	**0.0425**

表 8　10 种模型在 Mortgage 数据集上的测试性能

算法	不同比例噪声下的测试性能 (RMSE)				
	0	10%	20%	30%	40%
SCN [7]	0.0053	0.0229	0.0292	0.0379	0.0436
RSC-IQR [20]	0.0055	0.0072	0.0085	0.0105	0.0152
RSC-Huber [20]	0.0054	0.0092	0.0104	0.0136	0.0194
RSC-KDE [17]	0.0055	0.0081	0.0095	0.0122	0.0164
DSCN [22]	0.0048	0.0156	0.0185	0.0198	0.0222
RDSCN-KDE [26]	0.0048	0.0060	0.0077	0.0094	0.0120
WBLS-KDE [33]	0.0049	0.0063	0.0074	0.0094	0.0111
RBLS-Huber [34]	0.0049	0.0064	0.0081	0.0113	0.0156
WDSCN-Bisquare	**0.0047**	**0.0059**	**0.0073**	**0.0088**	0.0110
WDSCN-Huber	**0.0047**	0.0061	0.0074	0.0089	**0.0109**

随着训练数据集中噪声比例的不断提高，SCN 和 DSCN 的测试性能存在较为明显的下降，说明模型泛化性能易受到离群值的影响. 然而，本文提出的 2 种 WDSCN 模型在不同噪声比例下的 5 个数据

集中均取得了最好结果，随着噪声比例的提高，测试误差的增幅有效降低，下降趋势得到了缓解，提高了模型的鲁棒性和泛化性.

与此同时，通过对比基于不同 M-estimator 函数的 WDSCN，WDSCN-Huber 和 WDSCN-Bisquare 2 种模型均能够提高 DSCN 模型的鲁棒性，且能在不同噪声比例下的测试数据中取得最佳结果，其中 WDSCN-Bisquare 取得 17 次最佳结果（含并列最佳结果）、WDSCN-Huber 获得 13 次最佳结果（含并列最佳结果）. 根据上述实验结果可得，在噪声比例为 30%和 40%时，WDSCN-Bisquare 取得 4 次最佳结果（含并列最佳结果），WDSCN-Huber 获得 8 次最佳结果（含并列最佳结果），可得相比于 Bisquare 函数，Huber 函数能够对误差大的样本进行加权，随着噪声比例的增多，有效提高模型鲁棒性.

4.3 消融实验

为了说明提出模型的有效性，本节进行了消融实验，其中 WDSCN-Bisquare 和 WDSCN-Huber 代表最终模型；WDSCN-Bisquare*和 WDSCN-Huber*采用文献[17, 19, 20, 26]权重加权方式，且未采用 SC-SAE 进行稀疏特征学习（为避免多层模型过拟合，采取了正则化参数）；WDSCN-Bisquare**、WDSCN-Huber**的加权方式和 WDSCN-Bisquare、WDSCN-Huber 相同，但未采用 SC-SAE 进行稀疏特征学习，具体实验结果见表 9~表 13.

通过表 9~表 13 可得，WDSCN-Bisquare 和 WDSCN-Huber 在 Abalone、Wizmir、Concrete 和 Stock 中取得最优结果；在 Mortgage 数据集中，噪声比例为 0 时，WDSCN-Huber*最优；噪声比例为 40%时，WDSCN-Huber**和 WDSCN-Huber 最优.

表 9 6 种模型在 Abalone 数据集上的测试性能

算法	不同比例噪声下的测试性能 (RMSE)				
	0	10%	20%	30%	40%
WDSCN-Bisquare*	0.0750	0.0753	0.0753	0.0756	0.0766
WDSCN-Huber*	0.0746	0.0751	0.0762	0.0756	0.0760
WDSCN-Bisquare**	0.0753	0.0757	0.0759	0.0761	0.0766
WDSCN-Huber**	0.0747	0.0750	0.0761	0.0750	0.0762
WDSCN-Bisquare	**0.0736**	**0.0740**	**0.0743**	**0.0744**	0.0754
WDSCN-Huber	**0.0736**	0.0741	0.0744	**0.0744**	**0.0746**

表 10 6 种模型在 Wizmir 数据集上的测试性能

算法	不同比例噪声下的测试性能 (RMSE)				
	0	10%	20%	30%	40%
WDSCN-Bisquare*	0.0180	0.0182	0.0189	0.0198	0.0208
WDSCN-Huber*	0.0179	0.0190	0.0196	0.0205	0.0215
WDSCN-Bisquare**	0.0179	0.0185	0.0191	0.0194	0.0206
WDSCN-Huber**	0.0182	0.0185	0.0189	0.0196	0.0203
WDSCN-Bisquare	**0.0172**	**0.0172**	**0.0178**	**0.0183**	**0.0196**
WDSCN-Huber	0.0173	0.0173	0.0179	0.0184	**0.0196**

表 11 6 种模型在 Concrete 数据集上的测试性能

算法	不同比例噪声下的测试性能 (RMSE)				
	0	10%	20%	30%	40%
WDSCN-Bisquare*	0.0784	0.0831	0.0915	0.0942	0.1005
WDSCN-Huber*	0.0767	0.0828	0.0868	0.0937	0.0948
WDSCN-Bisquare**	0.0772	0.0826	0.0876	0.0916	0.0996
WDSCN-Huber**	0.0790	0.0833	0.0876	0.0919	0.0967
WDSCN-Bisquare	0.0750	**0.0788**	0.0838	0.0909	0.0953
WDSCN-Huber	**0.0748**	0.0799	**0.0835**	**0.0890**	**0.0941**

表 12 6 种模型在 Stock 数据集上的测试性能

算法	不同比例噪声下的测试性能 (RMSE)				
	0	10%	20%	30%	40%
WDSCN-Bisquare*	0.0294	0.0328	0.0379	0.0432	0.0492
WDSCN-Huber*	0.0291	0.0349	0.0397	0.0433	0.0483
WDSCN-Bisquare**	0.0305	0.0324	0.0366	0.0417	0.0472
WDSCN-Huber**	0.0307	0.0333	0.0362	0.0412	0.0447
WDSCN-Bisquare	**0.0290**	**0.0314**	**0.0348**	0.0396	0.0448
WDSCN-Huber	0.0293	0.0317	0.0349	**0.0385**	**0.0425**

表 13 6 种模型在 Mortgage 数据集上的测试性能

算法	不同比例噪声下的测试性能 (RMSE)				
	0	10%	20%	30%	40%
WDSCN-Bisquare*	0.0047	0.0062	0.0078	0.0098	0.0113
WDSCN-Huber*	**0.0045**	0.0073	0.0093	0.0110	0.0134
WDSCN-Bisquare**	0.0047	0.0060	0.0076	0.0088	0.0111
WDSCN-Huber**	0.0049	0.0063	0.0078	0.0089	**0.0109**
WDSCN-Bisquare	0.0047	**0.0059**	**0.0073**	**0.0088**	0.0110
WDSCN-Huber	0.0047	0.0061	0.0074	0.0089	**0.0109**

与此同时，WDSCN-Bisquare**、WDSCN-Huber**相比于 WDSCN-Bisquare*和 WDSCN-Huber*整体性能最优，在二类模型比较中，WDSCN-Bisquare**和 WDSCN-Huber**取得 16 次最佳结果（含并列最佳结果），WDSCN-Bisquare*和 WDSCN-Huber*取得 11 次最佳结果（含并列最佳结果），说明了 θ^2 项加权系数相比于 θ 项加权能够更好地降低高比例噪声的负面影响.

综上分析，本文采用加权方式及其基于 SC-SAE 的稀疏特征提取与融合方法能够更好地处理噪声数据，提高模型鲁棒性.

4.4 显著性检验

为展现所提出的 WDSCN-Bisquare 和 WDSCN-

Huber 模型与其他相关模型之间的显著差异，选取表 4~表 8 中 10 个模型在不同噪声比例下的 5 种数据集的测试误差进行 Friedman 和 post hoc Nemenyi 检验[37, 38]，其中所有算法实验结果的 Friedman 检验结果 p 值远小于 0.05，说明实验结果整体存在显著差异. 此外，WDSCN-Bisquare 和 WDSCN-Huber 模型与对比模型的 post hoc Nemenyi 显著性检验结果如表 14 所示. 从表 14 可以看出，除 RDSCN-KDE 外，WDSCN-Bisquare 和 WDSCN-Huber 模型与其它算法的统计检验结果均小于 0.05，证明提出模型与其他模型在 5%显著性水平上存在显著性差异.

需要特别指出的是，在实验中，为避免 RDSCN-KDE 过拟合，为其增加了正则化参数. 同时，尽管 WDSCN-Bisquare 和 WDSCN-Huber 模型与 RDSCN-KDE 模型不存在显著性差异，但是实验结果均优于 RDSCN-KDE.

表 14　显著性检验

	SCN	RSC-IQR	RSC-Huber	RSC-KDE	DSCN	RDSCN-KDE	WBLS-KDE	RBLS-Huber
WDSCN-Bisquare	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	0.244	**0.013**	**0.023**
WDSCN-Huber	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	0.383	**0.029**	**0.048**

4.5　案例分析

为进一步验证提出模型的有效性，本节 UCI（UCI Machine Learning Repository）①联合循环发电厂（Combined Cycle Power Plant, CCPP）数据对电厂每小时净电能输出进行预测.

该数据集采集了联合循环发电厂 2006 年~2011 年间的 9568 个样本数据，主要包括每小时的平均发电厂温度、平均环境压力、平均相对湿度、平均排气真空等 4 个特征变量，其中温度在 1.81 ℃~37.11 ℃之间，环境压力在 992.89~1033.30 mbar 之间，相对湿度在 25.56%~100.16%之间，排气真空范围 25.36~81.56 CM 汞柱. 每小时净电能输出在 420.26~495.76 MW 之间.

数据处理方式以及相关模型设置方式与 4.2 节相同，实验选择 75%的样本点作为训练数据，25%的样本点作为测试数据，WDSCN-Huber、WDSCN-Bisquare 和相关对比算法在 CCPP 数据集上的平均测试误差分别见表 15.

通过表 15 可得，WDSCN-Bisquare 和 WDSCN-Huber 对于联合循环发电厂每小时净电能输出预测结果均优于其它对比算法，WDSCN-Huber 性能更佳. 其中，随着噪声比例的增加，尽管为 RDSCN-KDE 引入了正则化策略，随着隐含层数量的增加，相比于 RSC-KDE 其性能提升有限. WDSCN- Bisquare 和 WDSCN-Huber 采用稀疏特征集成与融合使得 WDSCN 模型保持了良好的预测性能与鲁棒性，进一步验证了提出算法的有效性.

综上所述，WDSCN 能够有效应用于正常数据和异常数据处理，基于 M-estimator 函数加权和正则化策略的引入提高了模型的泛化性和鲁棒性.

表 15　10 种模型在 CCPP 数据集上的测试性能

算法	不同比例噪声下的测试性能 (RMSE)				
	0	10%	20%	30%	40%
SCN [7]	0.0535	0.0555	0.0558	0.0565	0.0569
RSC-IQR [20]	0.0541	0.0548	0.0549	0.0551	0.0556
RSC-Huber [20]	0.0543	0.0549	0.0552	0.0557	0.0556
RSC-KDE [17]	0.0542	0.0546	0.0549	0.0552	0.0558
DSCN [22]	0.0540	0.0823	0.1244	0.1722	0.2190
RDSCN-KDE [26]	0.0528	0.0545	0.0544	0.0550	0.0555
WBLS-KDE [33]	0.0532	0.0532	0.0537	0.0540	0.0550
RBLS-Huber [34]	0.0518	0.0520	0.0527	0.0532	0.0539
WDSCN-Bisquare	**0.0510**	0.0515	0.0520	0.0526	0.0530
WDSCN-Huber	**0.0510**	**0.0514**	**0.0519**	**0.0524**	**0.0529**

5　结束语

本文针对含有异常值的不确定数据回归问题，提出了基于 M-estimator 函数的加权深度随机配置网络，其中 WDSCN 采用多个不同的随机配置稀疏自编码器（SC-SAE）随机提取稀疏特征并融合，并在 KEEL 回归数据集上进行了实验验证. 实验结果表明，基于 Huber 和 Bisquare 加权和正则化策略的 WDSCN 能够有效处理不同噪声比例的不确定数据回归问题，克服 DSCN 因参数随机配置、对训练样本依赖性强而导致的模型泛化性能低、过拟合等问题，相比于 DSCN、SCN、RSC-KDE、RSC-IQR、RSC-Huber 、RSCN-KDE 、BLS-Huber 和 BLS-Bisquare 等鲁棒模型具有更高的回归性能. 与此同时，消融实验结果表明，引入稀疏特征的 WDSCN 相比于缺少 SC-SAE 特征提取的模型具有更高的回归精度，验证了基于 SC-SAE 提取特征并融合的有效性.

① UCI Machine Learning Repository, https://archive.ics.uci.edu/ml/index.php

此外，正则化策略是克服模型过拟合问题的核心，进一步优化正则化策略，选择 WDSCN 模型正则化参数，克服模型过拟合，将成为未来的研究重点.

参 考 文 献

[1] Pao Y H, Takefuji Y. Functional-link net computing: Theory, system architecture, and functionalities. Computer, 1992, 25(5): 76-79

[2] Scardapane S, Wang D H. Randomness in neural networks: An overview. WIREs Data Mining and Knowledge Discovery, 2017, 7(2): e1200

[3] Wang D H. Editorial: Randomized algorithms for training neural networks. Information Sciences, 2016, 364: 126-128

[4] Tyukin I, Prokhorov D. Feasibility of random basis function approximators for modeling and control//IEEE International Conference on Control Applications/International Symposium on Intelligent Control. St. Petersburg, Russia, 2009, 1391-1396

[5] Gorban A, Tyukin I, Prokhorov D, et al. Approximation with random bases: Pro et contra. Information Sciences, 2016, 364: 129-145

[6] Li M, Wang D H. Insight s into randomized algorithms for neural networks: Practical issues and common pitfalls. Information Sciences, 2017, 382: 170-178

[7] Wang D H, Li M. Stochastic configuration networks: Fundamentals and algorithms. IEEE Transactions on Cybernetics, 2017, 47(10): 3466-3479

[8] Li M, Wang D H. 2-D stochastic configuration networks for image data analytics. IEEE Transactions on Cybernetics, 2021, 51(1): 359-372

[9] Dai W, Li D P, Zhou P, et al. Stochastic configuration networks with block increments for data modeling in process industries. Information Sciences, 2019, 484: 367-386

[10] Zhu X L, Feng X C, Wang W W, et al. A further study on the inequality constraints in stochastic configuration networks. Information Sciences, 2019, 487: 77-83

[11] Zhang C L, Ding S F. A stochastic configuration network based on chaotic sparrow search algorithm. Knowledge-Based Systems, 2021, 220: 106924

[12] Huang C Q, Huang Q H, Wang D H. Stochastic configuration networks based adaptive storage replica management for power big data processing. IEEE Transactions on Industrial Informatics, 2020, 16(1): 373-383

[13] Wang Q J, Dai W, Ma X P, et al. Driving amount based stochastic configuration network for industrial process modeling. Neurocomputing, 2020, 394: 61-69

[14] Liu J N, Hao R J, Zhang T L, et al. Vibration fault diagnosis based on stochastic configuration neural networks. Neurocomputing, 2021, 434: 98-125

[15] Wang W, Jia Y, Yu W, et al. On-line ammonia nitrogen measurement using generalized additive model and stochastic configuration networks. Measurement, 2021, 170: 108743

[16] Gribonval R, Jenatton R, Bach F. Sparse and spurious: Dictionary learning with noise and outliers. IEEE Transactions on Information Theory, 2015, 61(11): 6298-6319

[17] Wang D H, Li M. Robust stochastic configuration networks with kernel density estimation for uncertain data regression. Information Sciences, 2017, 412-413: 210-222

[18] Xie J, Zhou P. Robust stochastic configuration network multi-output modeling of molten iron quality in blast furnace ironmaking. Neurocomputing, 2020, 387, 139-149

[19] Li M, Huang C Q, Wang D H. Robust stochastic configuration networks with maximum correntropy criterion for uncertain data regression. Information Sciences, 2019, 473: 73-86

[20] Dai W, Li D P, Chen Q X, et al. Data driven particle size estimation of hematite grinding process using stochastic configuration network with robust technique. Journal of Central South University, 2019, 26(1): 43-62

[21] Wu R W, Lv B Y, Dai C M, et al. Bayesian stochastic configuration networks for robust data modeling. Concurrency and Computation-Practice and Experience, 2022, 34: e6495

[22] Wang D H, Li M. Deep stochastic configuration networks with universal approximation property//2018 International Joint Conference on Neural Networks (IJCNN). Rio de Janeiro, Brazil, 2018: 1-8

[23] Felicetti M J, Wang D H. Deep stochastic configuration networks with different random sampling strategies. Information Sciences, 2022, 607: 819-830

[24] Felicetti M J, Wang D H. Deep stochastic configuration networks with optimised model and hyper-parameters. Information Sciences, 2022, 600: 431-441

[25] Zhang C L, Ding S F, Ding L. An AdaBoost based-deep stochastic configuration network. IFIP Advances in Information and Communication Technology, 2022, 643: 3-14

[26] Guo J C, Yan A J. Robust deep stochastic configuration network modeling method based on kernel density estimation//2021 33rd Chinese Control and Decision Conference (CCDC). Kunming, China, 2021, 575-579

[27] Zhao L J, Zou S D, Guo S, et al. Ball mill load condition recognition model based on regularized stochastic configuration networks. Control Engineering of China, 2020, 27(1): 1-7. (in Chinese)
(赵立杰, 邹世达, 郭烁, 等. 基于正则化随机配置网络的球磨机工况识别. 控制工程, 2020, 27(1): 1-7)

[28] Wang Q J, Yang C Y, Ma X P, et al. Underground airflow quantity modeling based on SCN. Acta Automatica Sinica, 2021, 47(8): 1963-1975. (in Chinese)
(王前进, 杨春雨, 马小平, 等. 基于随机配置网络的井下供给风量建模. 自动化学报, 2021, 47(8): 1963-1975)

[29] Rousseeuw P J, Croux C. The bias of k-step M-estimators. Statistics & Probability Letters, 1994, 20(5): 411-420

[30] Fan J, Yan A L, Xiu N H. Asymptotic properties for M-estimators in linear models with dependent random errors. Journal of Statistical Planning & Inference, 2014, 148(148): 49-66

[31] Olshausen B A, Field D J. Sparse coding with an overcomplete basis set: A strategy employed by V1?. Vision Research, 1997, 37(23): 3311-3325

[32] Boyd S, Parikh N, Chu E, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends in Machine learning, 2011, 3(1): 1-122

[33] Chu F, Liang T, Chen C L P, et al. Weighted broad learning system and its application in nonlinear industrial process modeling.

IEEE Transactions on Neural Networks and Learning Systems, 2020, 31(8): 3017-3031

[34] Guo W, Xu T. M-estimator-based robust broad learning system. Control and Decision, 2022. DOI: 10.13195/j.kzyjc. 2021.1479 (郭威, 徐涛. 基于 M-estimator 的鲁棒宽度学习系统. 控制与决策, 2022. DOI: 10.13195/j.kzyjc.2021.1479)

[35] Yang W G, Gao Y, Shi YH, et al. MRM-lasso: A sparse multiview feature selection method via low-rank analysis. IEEE Transactions Neural Networks and Learning Systems, 2015, 26(11): 2801-2815

[36] Gong M G, Liu J, Li H, et al. A multiobjective sparse feature learning model for deep neural networks. IEEE Transactions Neural Networks and Learning Systems, 2015, 26(12): 3263-3277

[37] Demšar J. Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research, 2006, 7: 1-30

[38] Reyes O, Altalhi AH, Ventura S. Statistical comparisons of active learning strategies over multiple datasets. Knowledge-Based Systems, 2018, 145: 274-288

DING Shi-Fei, Ph. D., professor and Ph. D. supervisor. His research interests include intelligent information processing, pattern recognition, machine learning, data mining, and granular computing.

ZHANG Cheng-Long, Ph. D. candidate. His research interests include stochastic configuration network and randomized learning method.

GUO Li-Li, Ph. D., lecturer. Her research interests include multimodal emotional computing, deep learning.

ZHANG Jian, Ph. D., lecturer. His research interests include deep learning, multi-label learning.

DING Ling, Ph. D. candidate. Her research interests include graph machine learning, deep clustering.

Background

Stochastic configuration network (SCN) is an incremental neural network, It assigns the input weights and biases of nodes in hidden layer according to supervision mechanism. Deep stochastic configuration network (DSCN) is a deep version of SCN, it can start from a small structure, increase the nodes and hidden layers gradually. All the nodes in hidden layer are fully connected to the outputs, the output weights of DSCN are determined through the least square method.

However, although the randomized feedforward learning process of DSCN has faster efficiency, the feature learning ability is still insufficient. In the meantime, with the increase of nodes and hidden layers, it is easy to lead to overfitting phenomenon. When solving regression problems with noise, the performance of original DSCN is easily affected by outliers, which reduces the generalization ability of the model.

Nowadays, the existing research of robust stochastic configuration network (RSCN) and deep stochastic configuration (RDSCN) mainly focus on the weight mode, ignore the risk of over-fitting and lack of feature learning ability. Aiming at enhancing uncertain data regression accuracy and robustness of DSCN, weighted deep stochastic configuration networks (WDSCN) based on M-Estimator functions are proposed. First of all, we adopt two commonly M-estimator functions (i.e., Huber and Bisquare) to acquire the sample weights for reducing the negative impact of outliers. Meanwhile, the weighted least square method and L_2 regularization strategy are introduced to calculate output weight vector replace the least square method. It can not only solve the noisy data regression problems and avoid over-fitting problem of DSCN. In the second place, for further improve the representation ability of WDSCN, a stochastic configuration sparse autoencoder (SC-SAE) is designed, SC-SAE use the supervision mechanism of DSCN to assign input parameters, at the same time, we apply the L_1 regularization technique to objective function for getting sparse features, alternating direction method of multipliers (ADMM) approach is utilized to solve the objective function for determining the output weights of SC-SAE. And then, as the randomness encoding process of SC-SAE, we can obtain the diversity of features through different SC-SAE models, consequently effective feature representation can be acquired through fusion features from multiple SC-SAE for the training of WDSCN. Finally, experimental results on real-world datasets show that the proposed WDSCN-Huber and WDSCN-Bisquare have higher generalization performances and regression accuracies than DSCN, SCN and other weighted models (e.g., RSC-KDE, RSC-Huber, RSC-IQR, RDSCN-KDE, WBLS-KDE and RBLS-Huber).

This work was supported by the National Natural Science Foundation of China under Grant No.62276265, No.61976216, No.62206297 and No.61672522.