

SESSC-DL-MTSK: 带有新型模糊化操作的 TSK 模糊系统

陈成杰, 梁 瑾

(华南师范大学人工智能学院, 广东 佛山 528225)

摘要: TSK 模糊系统是一种有效的分类器, 以其出色的可解释性和对不确定非线性系统的建模能力而著称。然而, 在经典 TSK 模糊系统中使用 K-means 和 FCM 聚类算法时, 存在模糊规则的激活饱和、处理特征维度超过 20 时聚类性能下降, 以及随着模糊规则数量增加模型复杂度升高的问题。为此, 提出一种基于监督增强软子空间聚类的新型 TSK 模糊系统 SESSC-DL-MTSK。首先, 使用监督增强软子空间聚类来训练模糊规则前件部分, 使用不同的特征空间构造模糊规则, 利用聚类中有价值的标签信息; 其次, 提出一种名为 MTSK 的创新型模糊化操作, 缓解了模糊规则的激活饱和问题; 最后, 结合 Batch Normalization、Dropout 等深度学习优化技术, 加速模型收敛并提高泛化性能。该系统在 UCI 中 10 个不同领域数据集上进行了实验, 平均准确率为 0.827 9, 系统参数数量为 15 093, 证实了其有效性。

关键词: TSK 模糊系统; 子空间聚类; 深度学习; 优化方法; 模糊规则

DOI: 10.11907/rjtk.242017

中图分类号: TP391

文献标识码: A

扫描二维码阅读全文:
文章编号: 1672-7800(2026)002-0065-09



SESSC-DL-MTSK: TSK Fuzzy System with New Fuzzification Operation

CHEN Chengjie, LIANG Jin

(School of Artificial Intelligence, South China Normal University, Foshan 528225, China)

Abstract: TSK fuzzy system is an effective classifier known for its excellent interpretability and modeling ability for uncertain nonlinear systems. However, when using K-means and FCM clustering algorithms in classic TSK fuzzy systems, there are issues such as activation saturation of fuzzy rules, decreased clustering performance when processing feature dimensions exceeding 20, and increased model complexity with increasing number of fuzzy rules. Therefore, a novel TSK fuzzy system SESSC-DL-MTSK based on supervised enhanced soft subspace clustering is proposed. The system first uses supervised enhanced soft subspace clustering to train the antecedent part of fuzzy rules, constructs fuzzy rules using different feature spaces, and utilizes valuable label information in clustering; Secondly, an innovative fuzzification operation called MTSK is proposed to alleviate the activation saturation problem of fuzzy rules; Finally, combining deep learning optimization techniques such as Batch Normalization and Dropout can accelerate convergence and improve generalization performance. The system was tested on 10 different domain datasets in UCI, with an average accuracy of 0.827 9 and a system parameter count of 15 093, confirming its effectiveness.

Key Words: TSK fuzzy system; subspace clustering; deep learning; optimization methods; fuzzy rule

0 引言

TSK (Takagi-Sugeno-Kang) 模糊系统是一种基于规则

的系统, 使用 IF-THEN 规则来实现不确定性的推理和模拟人类推理的过程, 在分类和回归问题上均有良好表现, 因为其输出为确定值, 易于解释与理解^[1-5]。目前, TSK 模糊系统集成了多种优化方法来微调模糊规则前件和后件的

收稿日期: 2025-01-13

基金项目: 广东省基础与应用基础研究基金省市联合基金重点项目(2020B1515120089); 广东省佛山市科技创新项目(2016AG100472)

作者简介: 陈成杰(1999-), 男, 华南师范大学人工智能学院硕士研究生, 研究方向为人工智能和模糊计算; 梁瑾(1975-), 男, 博士, 华南师范大学人工智能学院副教授、硕士生导师, 研究方向为人工智能和模糊计算。本文通讯作者: 梁瑾。

参数,常用的模糊规则前件优化方法包括 K-means、模糊 c-均值聚类(fuzzy C-Means, FCM)、均方层次模糊聚类等算法^[6-7];后件优化方法包括遗传算法、微分进化算法、修正支持向量机算法等^[8-10]。尽管上述方法可以提高 TSK 模糊系统的性能,但是随着数据维度的增加,模糊规则数量上升,系统复杂度随之增加,其可解释性会下降,准确性会降低。例如,自适应网络模糊推理系统(Adaptive-Neural-Network-Based Fuzzy Inference System, ANFIS)和多目标进化 TSK 模糊系统(Multi-Objective Evolutionary TSK Fuzzy System, METFS)使用的 FCM 使用全部特征来计算聚类中心,然而当数据维度超过 20 时,FCM 使用的欧氏距离逐渐变得不可区分,聚类中心会出现偏移^[11-12]。METFS 使用进化算法来调整模型参数,但需要保留大量候选解,并评估每个候选解的适应度,导致了高昂的计算成本和较大的内存需求,训练速度非常缓慢,会出现明显的过拟合和梯度消失等问题。为此,文献[13]通过深度卷积使模糊规则整体看起来像一个金字塔,从而减少了模糊规则数量,避免了规则爆炸问题。虽然模糊规则数量得到一定削减,但是每条模糊规则所使用的特征数量并没有变化,依然是使用全部特征来训练聚类中心,在处理维度较高的数据集时性能依然有所下降,弹性不足,模型复杂度亦没有得到优化。文献[14]通过在 TSK 模糊系统中引入深度神经网络改进了模糊规则的生成方式,但是存在饱和问题,且极大增加了模型复杂度。文献[15]在交叉熵损失函数中引入一个新的 UR(Uniform Regularization)项,提高了模型泛化性能,加快了训练的收敛速度,但依然没有解决模糊规则使用全部特征训练聚类中心以及模糊规则激活存在饱和的问题。

为解决上述问题,本文提出一种基于监督增强软子空间聚类的新型 TSK 模糊系统 SESSC-DL-MTSK(Maximized TSK Fuzzy System based on SESSC and Deep Learning)。首先,使用监督增强软子空间聚类(Supervised Enhanced Soft Subspace Clustering, SESSC)方法对每个模糊规则进行特征逼近,通过对不同模糊规则使用不同的特征子空间来构造模糊规则的前件部分^[16];其次,利用深度学习中的 BN(Batch Normalization)、MBGD(Mini-Batch Gradient Descent)、Dropout 和 Adabound 等技术来优化模糊规则的后件部分,提高处理高维数据集的性能,加快收敛速度,微调模糊规则后件参数^[17-23];最后,提出一种名为 MTSK(Maximized TSK)的新模糊化操作,用于解决模糊规则的激活饱和问题。

TSK 模糊系统的最终输出等同于 softmax 函数,会出现饱和问题,导致大值越来越大,小值越来越少,降低了模型的泛化程度^[24-25]。通过以上方法可得到一个更为精准的聚类中心,构建出简洁的模糊规则库。模糊规则库中的每条模糊规则使用的特征数量都不同,使得模型复杂度进一

步降低,且每条模糊规则都能得到很好地激活,处于一个较为均匀的激活状态,解决了模糊规则的激活饱和问题。此外,在合理减少特征和模糊规则数量后得到了更为简洁高效的 TSK 模糊系统,其在处理高维数据时具有更强的准确性和鲁棒性。

1 系统构建

1.1 TSK 模糊系统

TSK 模糊系统中的第 r 条模糊规则定义为^[26]:

$$\begin{aligned} \text{Rule}_r: \quad & \text{IF } x_1 \text{ is } A_{r,1} \text{ and } \cdots \text{ and } x_D \text{ is } A_{r,D}, \\ & \text{THEN } y_r^c(x) = b_{r,0}^c + \sum_{d=1}^D b_{r,d}^c \cdot x_d \end{aligned} \quad (1)$$

式中: $x = \{x_1, x_2, \cdots, x_D\}^T$, 为训练数据集中的 D 维输入向量; $A_{r,d} (r = 1, \cdots, R; d = 1, \cdots, D)$, 为第 r 条规则中与第 d 个特征相关的模糊子集; $b_{r,0}^c$ 和 $b_{r,d}^c$ 为第 c 个类别的模糊规则后件参数。

TSK 模糊系统的输出可表示为:

$$y^c(x) = \sum_{r=1}^R \tilde{f}_r(x) y_r^c(x) \quad (2)$$

$$\tilde{f}_r(x) = \frac{f_r(x)}{\sum_{r'=1}^R f_{r'}(x)} \quad (3)$$

$$f_r(x) = \prod_{d=1}^D \mu_{A_{r,d}}(x_d) = \exp \left(- \sum_{d=1}^D \frac{(x_d - m_{r,d})^2}{2\sigma_{r,d}^2} \right) \quad (4)$$

式中: $\tilde{f}_r(x)$ 为模糊规则的归一化激活水平; $f_r(x)$ 为第 r 条模糊规则的激活水平; $\mu_{A_{r,d}}$ 为特征 x_d 在模糊集 $A_{r,d}$ 上的隶属度值; $m_{r,d}$ 和 $\sigma_{r,d}$ 分别为通过聚类算法计算出来的聚类中心和标准偏差。如果使用 FCM 计算 $m_{r,d}$ 和 $\sigma_{r,d}$, 则计算公式为:

$$m_{r,d} = \sum_{j=1}^N \mu_{jr} x_{jd} / \sum_{j=1}^N \mu_{jr} \quad (5)$$

$$\sigma_{r,d}^2 = h \sum_{j=1}^N \mu_{jr} (x_{jd} - m_{r,d})^2 / \sum_{j=1}^N \mu_{jr} \quad (6)$$

$$\mu_{jr} = \frac{1}{\sum_{k=1}^c \left(\frac{\|x_{jd} - m_{r,d}\|}{\|x_{jd} - m_{r,k}\|} \right)} \quad (7)$$

式中: N 为输入样本的数量; x_{jd} 为输入数据 j 的第 d 个特征; μ_{jr} 为第 r 条规则上输入数据 j 的隶属度值; h 为一个可手动调节的缩放因子^[27]。为了优化 TSK 模糊分类器,需要对规则的前件和后件参数进行微调。

1.2 使用 SESSC 训练模糊规则前件

目前有许多软子空间聚类算法用于处理高维数据,但大多侧重于簇内紧凑性,而簇间可能难以区分开来。相比之下,SESSC 不仅注重簇内的紧凑性,而且注重簇间的可分离性,同时有效利用标签信息来获得更好的判别性能。对于输入数据 $X = \{x_1, x_2, \cdots, x_N\}$ 与其相对应的样本标签 $Y = \{y_1, y_2, \cdots, y_N\}$, SESSC 的损失函数为:

$$J_{\text{SESSC}}(U, V, W, Z) = \sum_{n=1}^N \sum_{r=1}^R U_{n,r}^m (x_n - v_r)^2_{\omega_r} + \gamma \sum_{r=1}^R \sum_{d=1}^D W_{r,d} \ln(W_{r,d}) - \eta \sum_{r=1}^R \left(\sum_{n=1}^N U_{n,r}^m \right) // v_r - v_0 //^2_{\omega_r} + \beta \sum_{n=1}^N \sum_{r=1}^R U_{n,r}^m \left[- \sum_{c=1}^C Y_{n,c} \ln(Z_{r,c}) \right] \quad (8)$$

$$s.t. U_{n,r} \in [0, 1], \quad W_{r,d} \in [0, 1]$$

$$\sum_{r=1}^R U_{n,r} = 1, \quad \sum_{d=1}^D W_{r,d} = 1, \quad \sum_{c=1}^C Z_{r,c} = 1$$

式中: $U_{n,r}^m$ 为 x_n 在第 r 条规则上的隶属度; m 为模糊索引值; v_r 为第 r 条规则的聚类中心; $v_0 = \frac{1}{N} \sum_{n=1}^N x_n$ 为所有输入样本的中心; $W_{r,d}$ 为第 r 条规则在第 d 个特征上的正则化权重; $Y_{n,c}$ 为 x_n 属于第 c 类别的样本标签; $Z_{r,c}$ 为聚类 r 属于 c 类的概率; γ, η, β 为自定义的正则化参数; N, R, D, C 分别为数据样本数量、聚类中心数量、数据特征维度、类别数量。

SESSC 的目标函数由 4 项组成, 分别对应加权簇内紧凑性、特征权重正则化、加权簇间分离性和类标签信息。其中, 第一项继承了属性加权模糊聚类 (Attribute Weighting Fuzzy Clustering, AWFCM) 的目标函数, 使用加权欧氏距离计算^[28]; 第二项继承了熵正则化加权模糊聚类 (Entropy-regularized Weighted Fuzzy C-means, EWFCM) 的目标函数, 使用熵作为正则项来控制权重^[29]; 第三项利用样本中心和簇内中心来控制簇之间的分离状态; 第四项利用标签信息辅助决策。SESSC 的目标函数可通过拉格朗日乘法求得 U, V, W, Z 的最优公式。表示为:

$$U_{n,r} = \frac{\{ // x_n - v_r //^2_{\omega_r} - \beta \sum_{c=1}^C Y_{n,c} \ln(Z_{r,c}) \}^{-\frac{1}{m-1}}}{\sum_{i=1}^R \{ // x_n - v_i //^2_{\omega_i} - \beta \sum_{c=1}^C Y_{n,c} \ln(Z_{i,c}) \}} \quad (9)$$

$$V_{r,d} = \frac{\sum_{n=1}^N U_{n,r} (X_{n,d} - \eta V_{0,d})}{\sum_{n=1}^N U_{n,r} (1 - \eta)} \quad (10)$$

$$W_{r,d} = \frac{\exp \left(- \left\{ \sum_{n=1}^N U_{n,r} \left[(X_{n,d} - V_{r,d})^2 - \eta (V_{r,d} - V_{0,d})^2 \right] \right\} / \gamma \right)}{\sum_{i=1}^D \exp \left(- \left\{ \sum_{n=1}^N U_{n,r} \left[(X_{n,i} - V_{r,i})^2 - \eta (V_{r,i} - V_{0,i})^2 \right] \right\} / \gamma \right)} \quad (11)$$

$$Z_{r,c} = \frac{\sum_{n=1}^N \beta U_{n,r}^m Y_{n,c}}{\sum_{i=1}^C \sum_{n=1}^N \beta U_{n,r}^m Y_{n,i}} \quad (12)$$

可以根据以上优化公式不断更新聚类中心, 得到最优结果。

1.3 创新型模糊化操作 MTSK

设式(4)中的 $-\sum_{d=1}^D \frac{(x_d - m_{r,d})^2}{2\sigma_{r,d}^2}$ 为 P_r , 则式(3)可更新为:

$$\tilde{f}_r(x) = \frac{f_r(x)}{\sum_{r'=1}^R f_{r'}(x)} = \frac{\exp(-P_r)}{\sum_{r'=1}^R \exp(-P_{r'})} \quad (13)$$

最终的解模糊化操作如式(13)所示, 等同于 softmax 函

数, 而该函数存在饱和问题^[30]。随着维度增加, P_r 会越来越小, 模糊规则激活的数量也会越来越小。例如, 有 4 条模糊规则 $P = [-0.1, -0.7, -0.3, -0.5]$, 那么 $\tilde{f} = [0.3292, 0.1807, 0.2695, 0.2207]$, 此时每条规则的激活水平差异很小, 可激活的规则数量也很多。当维度增加时, 有 $P = [-10, -70, -30, -50]$, $\tilde{f} = [1, 8.7565 \times 10^{-27}, 2.0612 \times 10^{-9}, 4.2484 \times 10^{-18}]$ 。此时, 只能激活第一条规则, 其余规则几乎不会触发。为避免模糊规则饱和的问题, 本文提出一种名为 MTSK 的新模糊化操作。表示为:

$$\tilde{P}_r = -\frac{1}{D} \left(\sum_{d=1}^D \frac{(x_d - m_{r,d})^2}{2\sigma_{r,d}^2} - P_{\max} \right) \quad (14)$$

式中: D 为特征维度大小; $P_{\max} = \max(P_1, P_2, \dots, P_r)$ 。

则式(2)可更新为:

$$y^c(x) = \sum_{r=1}^R \tilde{f}_r^{\text{MTSK}}(x) y_r^c(x) \quad (15)$$

$$\tilde{f}_r^{\text{MTSK}}(x) = \frac{\exp(\tilde{P}_r)}{\sum_{r'=1}^R \exp(\tilde{P}_{r'})} \quad (16)$$

如此一来, 可以避免 $\tilde{P}_r$ 的下溢问题, 同时缓解 $\tilde{P}_r$ 对特征维度的依赖性。

1.4 使用深度学习优化技术训练模糊规则后件

以往的优化算法, 例如遗传算法和微分进化算法, 需对整个种群进行突变与交叉操作, 并设置一些关键参数, 如种群大小和交叉概率, 这些操作都是计算密集型的, 需要耗费大量计算资源。为此, 本文使用深度学习优化技术, 通过权衡可用内存、训练速度与目标函数的预期泛化性能来评估和优化模糊规则后件部分。目标函数表示为:

$$J = -\frac{1}{N} \sum_{n=1}^N \sum_{c=1}^C y_{n,c} \log(\hat{y}_{n,c}) \quad (17)$$

$$\hat{y}_{n,c} = \sum_{r=1}^R \tilde{f}_r(x) \cdot \{ (b'_c)^T \cdot \Gamma(x_n) + b'_{c,0} \} \quad (18)$$

式中: $y_{n,c}$ 为 x_n 属于第 c 类的标签; $\hat{y}_{n,c}$ 为模糊规则后件的输出结果; $\Gamma(x_n)$ 可表示为:

$$\Gamma(x_n) = \phi(BN(x_n)) \quad (19)$$

本文使用 MBGD 训练模型, 但其容易陷入局部极小值, 导致梯度消失问题, 因此使用 BN 对输入数据进行归一化处理, 并通过减少内部协变量来加速模型训练。对于一个小批度 $\mathcal{B} = \{x_n\}_{n=1}^m$, BN 的输出为:

$$x'_n = BN(x_n) = \gamma \frac{x_n - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} + \beta \quad (20)$$

式中: μ_B 和 σ_B^2 分别为小批量样品的均值和标准差。

使用 sigmoid 函数作为非线性函数, 非线性变化后的数据为 $\hat{x}_n$ 。表示为:

$$\hat{x}_n = \phi(x'_n) = \frac{1}{1 + \exp(-x'_n)} \quad (21)$$

在训练过程中, 模糊规则会出现冗余, 降低 TSK 模糊系统的泛化程度, 因此引入 Dropout 技术, 选择性丢弃部分规则。例如, 丢弃第二条规则的效果如图 1 所示。

1.5 SESSC-DL-MTSK 模糊系统

综上所述, 使用 SESSC 生成和优化模糊规则的前件部分, 使用 MTSK 进行模糊化操作, 使用深度学习技术生成和优化模糊规则的后件部分。最终得到的 SESSC-DL-

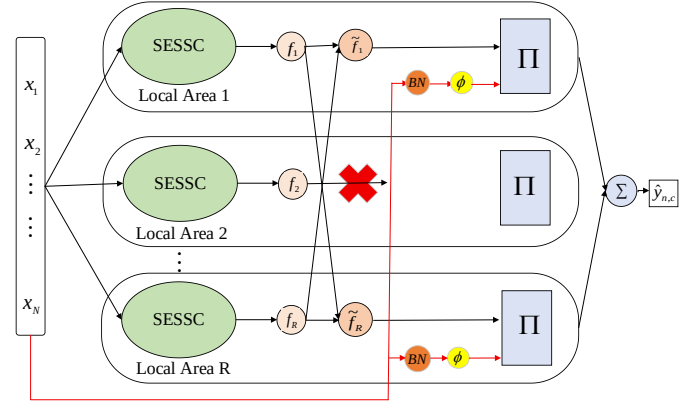


Fig. 1 The effect of discarding the second rule

图1 丢弃第二条规则的效果

MTSK 系统结构如图 2 所示。

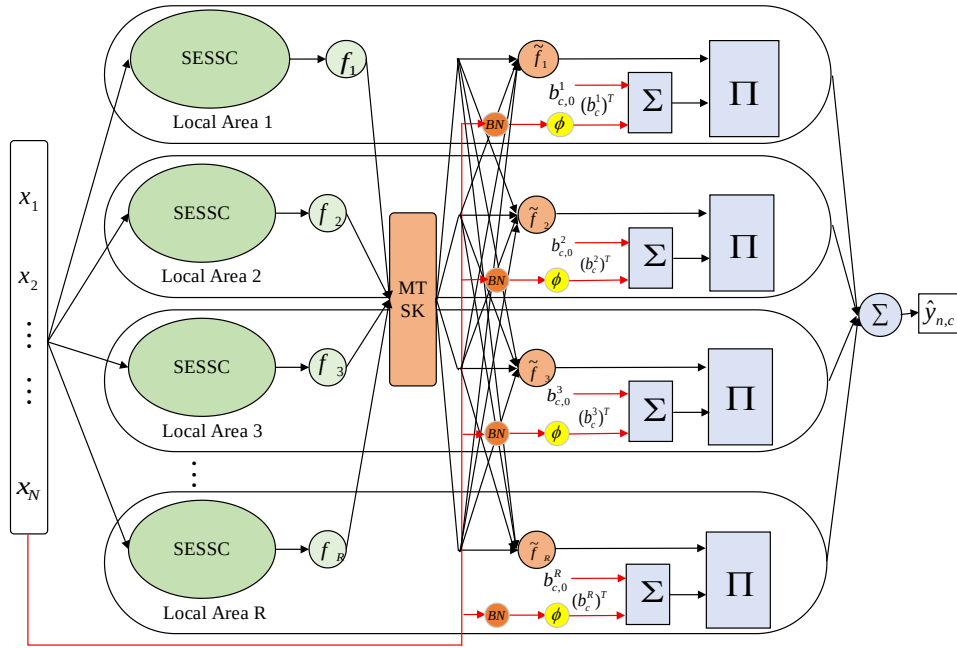


Fig. 2 The architecture of SESSC-DL-MTSK system

图2 SESSC-DL-MTSK 系统结构

SESSC-DL-MTSK 模糊系统的训练步骤为:

输入: 训练样本 (X_{train}), 训练样本标签 (Y_{train}), 测试样本 (X_{valid}), 测试样本标签 (Y_{valid}), 模糊规则数量 (R), 特征数量 (D), 类别数量 (C), SESSC 迭代的最大次数 (E), 模糊系统训练迭代的最大次数 (T), 停止训练的错误阈值 (ϵ), 正则化参数 (γ, η, β), 模糊化调节因子 (h), 批度大小 (B), 训练样本数据大小 (N), 最好分类准确率 (BAC) 输出: 新的测试用例

输出: 最佳性能状态的 SESSC-DL-MTSK

1. 在 X_{train} 上使用 FCM ($k = R$) 初始化一个原始的中心聚类 V

2. $\hat{V} \leftarrow V$

3. $v_0 \leftarrow \frac{1}{N} \sum_{n=1}^N x_n, x_n \in X_{train}$

4. $W_{r,d} \leftarrow \frac{1}{D}, r = 1, \dots, R, d = 1, \dots, D$

5. $Z_{r,c} \leftarrow \frac{1}{C}, r = 1, \dots, R, d = 1, \dots, C$

6. FOR $i = 1, 2, \dots, E$ DO

7. 用式 (9) 更新 U , 用式 (10) 更新 V , 用式 (11) 更新 W , 用式 (12) 更新 Z

8. IF $\| \hat{V} - V \|_2 < \epsilon$ THEN

9. BREAK

10. ELSE

11. $\hat{V} \leftarrow V$

12. END IF

13. END FOR

14. 用式 (6) 计算 σ^2

15. 用式 (20) 计算 x'_n , 用式 (21) 计算 $\hat{x}_n$

16. BAC $\leftarrow 0$

17. FOR $j = 1, 2, \dots, T$ DO

18. FOR $k = 1, 2, \dots, \left\lfloor \frac{N}{B} \right\rfloor$

```
19.      用式(14)计算 $\widetilde{P}_r$ 
20.      用式(16)计算 $\widetilde{f}_r^{MTSK}$ 
21.      用式(18)计算预测值 $\hat{y}$ 
22.      用式(17)计算拟合误差 Loss
23.      使用后向传播计算梯度 $\frac{\partial Loss}{\partial b}$ 
24.      END FOR
25.       $Acc_v \leftarrow Accuracy(X_{valid})$ 
26.      IF  $Acc_v > BAC$  THEN
27.           $BAC \leftarrow Acc_v$ 
28.          保留最好的性能模型 SESSC-DL-MTSK
29.      ELSE
30.          CONTINUE
31.      END IF
32. END FOR
33. RETURN SESSC-DL-MTSK model
```

2 实验方法与结果分析

2.1 实验环境

本文基于 PyTorch 深度学习框架构建算法模型,并且使用 torch 自带的函数功能进行数据增强。实验代码在 PyCharm 中调试,模型训练和测试亦在 PyCharm 下进行。硬件使用一块 8 GB 显存的 GTX3070 显卡,采用 AdaBound 策略逐渐衰减学习率,共训练 300 个 epoch,选取训练过程中性能最优的模型作为实验模型。

2.2 数据集

从 UCI 机器学习数据库 Repository 和 NTU 机器学习数据库 Lib 中选择 10 个分类数据集,每个数据集随机选取 70% 的样本作为训练集,10% 作为验证子集,其余 20% 作为测试集。表 1 为数据集的详细信息。

Table 1 Detailed information of the datasets

表 1 数据集详细信息

数据集	样本数	特征数	类别数	来源
Abalone	4 177	10	3	https://archive.ics.uci.edu/dataset/1/abalone
Clave	10 798	16	4	https://archive.ics.uci.edu/dataset/324/firm+teacher+clave+direction+classification
DRD	1 151	19	2	https://archive.ics.uci.edu/dataset/329/diabetic+retinopathy+debreceen
Waveform21	5 000	21	3	https://archive.ics.uci.edu/dataset/107/waveform+database+generator+version+1
Steel	1 941	27	7	https://archive.ics.uci.edu/dataset/198/steel+plates+faults
WDBC	569	30	2	https://archive.ics.uci.edu/dataset/17/breast+cancer+wisconsin+diagnostic
Student	4 424	36	3	https://archive.ics.uci.edu/dataset/697/predict+students+dropout+and+academic+success
Satellite	6 435	36	6	https://archive.ics.uci.edu/dataset/146/statlog+landsat+satellite
Biodeg	1 055	41	2	https://archive.ics.uci.edu/dataset/254/qsar+biodegradation
Usps	9 298	256	10	https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass.html#usps

2.3 对照方法

为评估 SESSC-DL-MTSK 的有效性,将其与其他 5 种算法进行比较。其中,决策树(Decision Tree, DT)、随机森林(Random Forest, RF)是基于树的方法;TSK-PCA 是通过传统方法优化的 TSK 模糊系统;TSK-BN-UR^[31]和 LogTSK^[32]是近年来提出的高效 TSK 模糊系统。所有算法的参数均采用五折交叉验证和网格搜索策略进行优化。具体如表 2 所示。

Table 2 Parameter setting of the algorithms

表 2 算法参数设置

参数	数值
R	20
m	$\min(N, D-1)/(\min(N, D-1)-2)$ if $\min(N, D-1)>2$, otherwise 2
h	{0.01, 0.1, 1, 10, 100}
γ	{0.01, 0.1, 1, 10, 100}
η	{0.01, 0.05, 0.1, 0.3, 0.5}
β	{0.01, 0.1, 1, 10, 100}
α	{0.01, 0.05, 0.1, 0.15, 0.2}

2.4 评价指标

从分类性能和模型复杂度两个角度对模型性能进行评估。

2.4.1 分类性能

采用原始分类准确率(Raw Classification Accuracy,

RCA)、平衡分类准确率(Balance Classification Accuracy, BCA)和 Jaccard-measure 3 个指标评估五折交叉验证的分类性能。其中,RCA 为正确分类的测试样本总数占测试样本总数的比例;BCA 为每个类别的 RCA 平均值,用于评估模型在不平衡数据集上的性能;Jaccard-measure 用于评估模型的分类性能和泛化性。3 个指标的计算公式为:

$$RCA = \frac{TP + TN}{TP + TN + FP + FN} \tag{22}$$

$$BCA = \frac{\sum_i^c RCA}{C} \tag{23}$$

$$Jaccard = \frac{TP}{TP + FP + FN} \tag{24}$$

式中:TP 表示属于正类并且被模型正确预测为正类的样本;TN 表示属于负类并且被模型正确预测为负类的样本;FP 表示被预测为正类但实际上为负类的样本;FN 表示被预测为负类但实际上为正类的样本。

2.4.2 模型复杂度

TSK-PCA、TSK-UR-BN、LogTSK、SESSC-DL-MTSK 的模型复杂度通过参数量来评估。训练得到的模糊系统包含前件参数 m 和 σ^2 ,以及模糊成员函数中的后件参数 $b_{c,d}^r$,前件参数和非零的后件参数决定了 TSK 模糊系统的最终复杂度。DT 的模型复杂度由模型内部结点数量、特征数

量、叶子节点数量、类别数量决定。RF 的模型复杂度由树的数量、每棵树的深度、分支节点数量、叶节点数量、特征数量决定。

2.5 实验结果与分析

2.5.1 分类性能

6 种算法在 10 个数据集上的 RCA 结果如表 3 所示,最优结果用粗体标出(下同)。可以看出,本文算法在 10 个数据集上的平均 RCA 高于其他算法;在 3 个特征维度小于 20 的数据集上,有 2 个数据集的性能优于其他算法;在 7 个特征维度大于 20 的数据集上,有 5 个数据集的性能优于其他算法。总体来说,不管是低维还是高维数据,本文算法均具有较好的分类性能。

Table 3 RCA results of 6 algorithms on 10 datasets

表 3 6 种算法在 10 个数据集上的 RCA 结果

数据集	DT	RF	TSK- PCA	TSK- UR-BN	LogTSK	SESSC- DL-MTSK
Abalone	0.494 5	0.546 3	0.550 2	0.566 9	0.565 5	0.576 9
Clave	0.741 7	0.811 5	0.801 9	0.854 0	0.853 7	0.867 2
DRD	0.617 0	0.690 3	0.744 6	0.723 8	0.718 0	0.722 9
Waveform21	0.753 1	0.847 2	0.849 0	0.874 8	0.863 1	0.869 2
Steel	0.720 7	0.770 0	0.766 1	0.746 0	0.737 4	0.745 6
WDBC	0.924 3	0.962 0	0.956 1	0.963 7	0.968 2	0.985 7
Student	0.678 1	0.765 1	0.724 3	0.768 1	0.766 8	0.776 9
Satellite	0.854 1	0.912 6	0.855 5	0.901 3	0.905 7	0.913 6
Biodeg	0.824 0	0.867 8	0.862 6	0.872 2	0.879 4	0.880 7
Usps	0.850 5	0.926 8	0.909 8	0.917 7	0.923 3	0.939 8
平均	0.745 8	0.810 0	0.802 0	0.818 8	0.818 1	0.827 9

6 种算法在 10 个数据集上的 BCA 结果如表 4。可以看出,本文算法在 10 个数据集上的平均 BCA 优于其他算法;在 3 个特征维度小于 20 的数据集上,有 2 个数据集的性能优于其他算法;在 7 个特征维度大于 20 的数据集上,有 5 个数据集的性能优于其他算法。说明本文算法在处理不平衡数据方面具有较好性能。

Table 4 BCA results of 6 algorithms on 10 datasets

表 4 6 种算法在 10 个数据集上的 BCA 结果

数据集	DT	RF	TSK- PCA	TSK- UR-BN	LogTSK	SESSC- DL-MTSK
Abalone	0.497 2	0.548 7	0.554 6	0.567 0	0.565 3	0.575 3
Clave	0.611 3	0.601 3	0.564 0	0.714 2	0.695 0	0.756 0
DRD	0.616 1	0.691 4	0.750 4	0.724 8	0.719 2	0.726 3
Waveform21	0.753 2	0.847 0	0.848 6	0.874 8	0.863 0	0.869 6
Steel	0.740 3	0.773 6	0.782 5	0.758 2	0.748 6	0.771 2
WDBC	0.923 7	0.957 5	0.944 4	0.964 1	0.964 4	0.984 3
Student	0.619 7	0.670 2	0.625 8	0.662 4	0.673 1	0.689 7
Satellite	0.831 4	0.882 0	0.827 4	0.875 6	0.881 4	0.890 6
Biodeg	0.804 6	0.843 0	0.829 2	0.838 9	0.861 8	0.865 3
Usps	0.837 8	0.919 0	0.902 8	0.911 4	0.917 4	0.934 3
平均	0.723 5	0.773 4	0.763 0	0.789 1	0.788 9	0.806 3

6 种算法在 10 个数据集上的 Jaccard 结果如表 5。可以看出,本文算法在 10 个数据集上的平均 Jaccard 优于其他算法,在其中 5 个数据集上的表现明显优于其他算法,说

明本文算法在分类性能和泛化性上有较好表现。

Table 5 Jaccard results of 6 algorithms on 10 datasets

表 5 6 种算法在 10 个数据集上的 Jaccard 结果

数据集	DT	RF	TSK- PCA	TSK- UR-BN	LogTSK	SESSC- DL-MTSK
Abalone	0.326 7	0.376 9	0.379 5	0.400 5	0.400 2	0.402 0
Clave	0.587 1	0.677 9	0.669 2	0.745 2	0.744 7	0.765 6
DRD	0.437 5	0.512 1	0.593 1	0.579 8	0.560 5	0.566 5
Waveform21	0.605 8	0.734 7	0.737 6	0.777 5	0.774 9	0.769 3
Steel	0.563 6	0.626 5	0.620 8	0.607 9	0.596 8	0.594 5
WDBC	0.851 5	0.923 9	0.916 0	0.969 1	0.938 4	0.965 6
Student	0.513 1	0.619 7	0.567 8	0.630 2	0.628 4	0.635 3
Satellite	0.747 1	0.840 2	0.747 5	0.820 4	0.830 3	0.841 0
Biodeg	0.681 7	0.765 7	0.758 3	0.773 6	0.801 1	0.787 0
Usps	0.739 5	0.863 5	0.848 0	0.832 8	0.875 0	0.950 2
平均	0.605 4	0.694 1	0.683 8	0.713 7	0.715 0	0.727 7

2.5.2 模型复杂度

6 种算法在 10 个数据集上的模型复杂度见表 6,所有对照算法的模型复杂度均在表 4 中 BCA 值最优的情况下计算得出。可以看出,本文算法在 10 个数据集上的模型复杂度均低于其他算法。

2.5.3 消融实验

Table 6 Model complexity of 6 algorithms on the 10 datasets

表 6 6 种算法在 10 个数据集上的模型复杂度

数据集	DT	RF	TSK- PCA	TSK- UR-BN	LogTSK	SESSC- DL-MTSK
Abalone	23 678	15 531 349	1 320	1 860	1 860	250
Clave	99 747	123 298 017	3 440	3 920	3 920	3 128
DRD	7 811	625 561	2 200	2 320	2 320	187
Waveform21	21 948	3 886 573	3 480	3 840	3 840	3 065
Steel	18 062	1 369 873	11 060	11 480	11 480	791
WDBC	4 205	11 239	3 520	3 640	3 640	2 592
Student	52 653	8 847 531	6 360	6 540	6 540	3 630
Satellite	36 329	3 523 008	12 000	13 080	13 080	10 461
Biodeg	8 609	233 964	4 840	4 960	4 960	3 580
Usps	219 029	4 171 901	152 600	153 800	153 800	123 248
平均	49 207	16 149 901	20 082	20 544	20 544	15 093

在保持其他参数不变的情况下分析添加和未添加 SESSC 的情况下本文算法的模型复杂度和分类性能。结果如图 3 所示。可以看出,在添加 SESSC 的情况下,算法在每个数据集上的模型复杂度均有不同程度的降低。当数据维度较大时,如 Satellite 和 Usps 数据集,添加 SESSC 算法的分类性能优于未添加 SESSC 时的性能。然而,添加 SESSC 的模型在 DRD 和 Steel 数据集上的 RCA 值有明显下降,这是由于使用的特征太少。由此可以得出结论,在处理高维数据时,SESSC 可有效减少参数量,并具有更好的分类性能,有助于构建简洁高效的模糊系统。

为验证引入深度学习优化技术 Dropout 的有效性,进行以下实验。以 DRD 数据集为例, x 轴表示规则库中所有规则的索引,一开始有 20 条规则。 y 轴表示规则的长度,

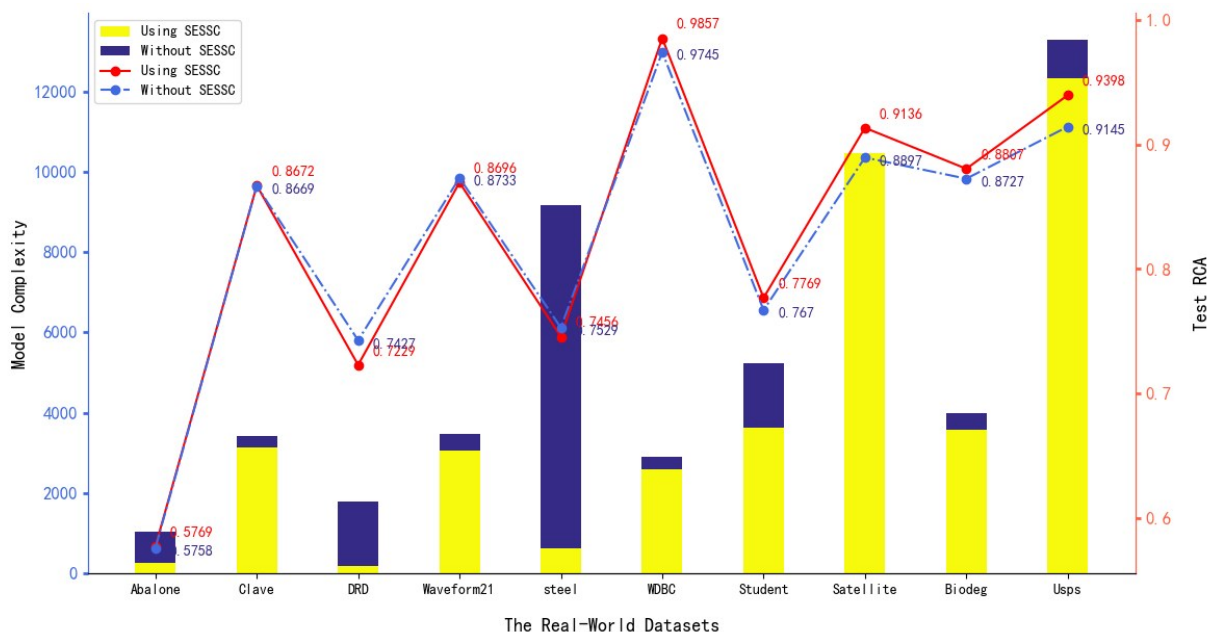


Fig. 3 The model complexity and classification performance of the proposed algorithm with and without SESSC added

图3 添加和未添加SESSC时本文算法的模型复杂度和分类性能

其由每个规则所选特征的数量决定。从图4可以看出,应用SESSC后只有部分特征被使用,即规则长度减少。考虑到人类认知能力有限,可以认为所提算法产生了更具有可解释性的结果。此外,应用Dropout后,一些冗余规则被舍弃,图中被丢弃的规则指数分别为5、6、7、9、12、16。因此,SESSC-DL-MTSK可以构建模型复杂度更低的模糊系统,并提高其可解释性。

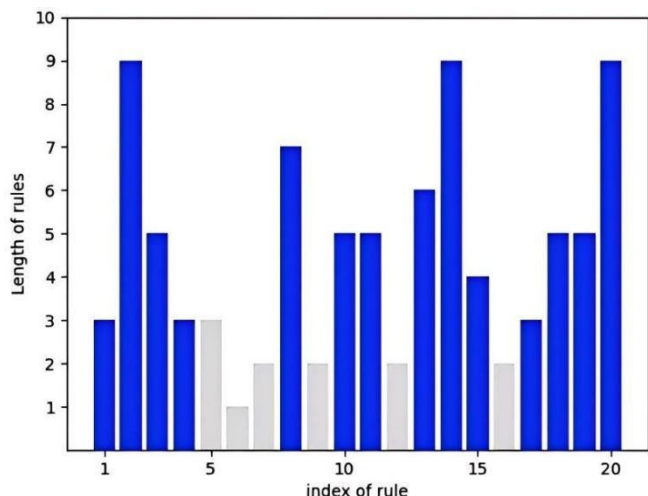


Fig. 4 Fuzzy rule result with and without Dropout

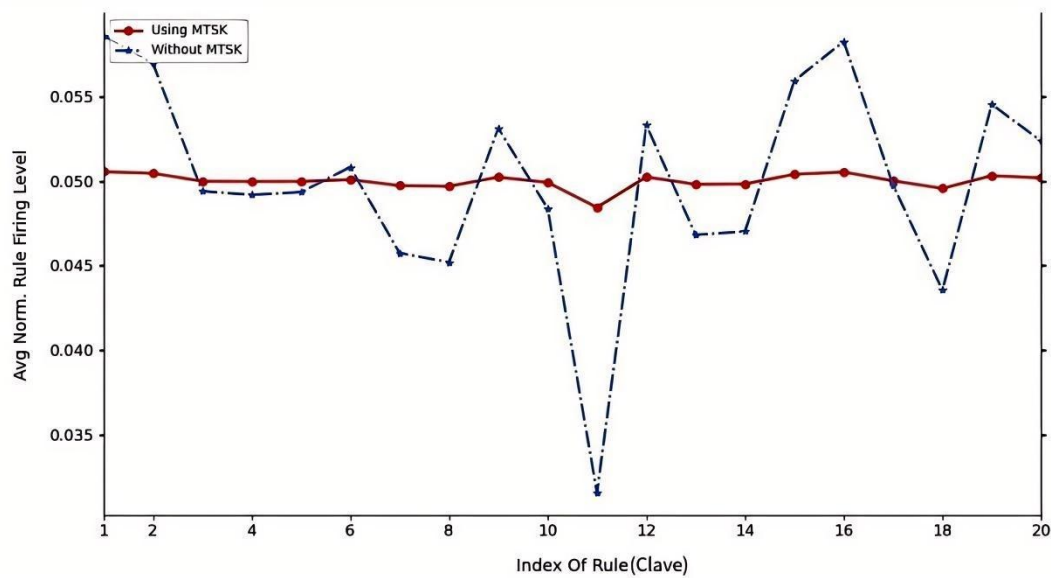
图4 使用Dropout前后的模糊规则结果

图5为在Clave和WDBC数据集上使用和未使用MTSK的规则平均归一化触发水平。可以看出,在未使用MTSK的模型中,饱和规则的平均触发水平大于其他规则,因此这些规则对输出的贡献大不相同。而使用MTSK的模型的平均归一化规则触发水平比未使用MTSK的模型更均匀,说明MTSK模糊化操作可以缓解去模糊化操作存在的饱和问题,这也实现了更好的分类性能。

3 结语

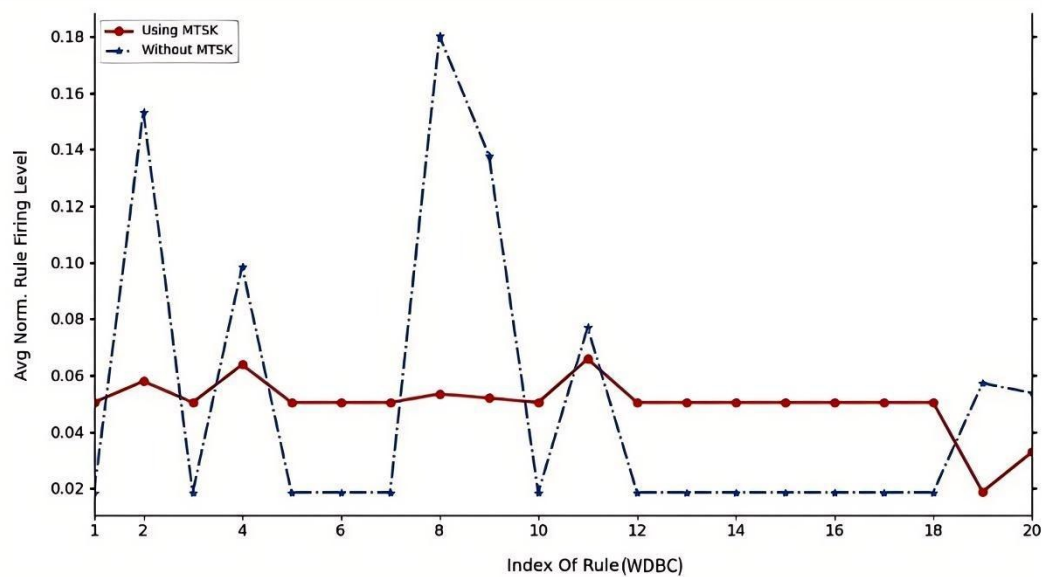
当数据维度较高时,以往常用的聚类算法在聚类时逐渐变得不可区分,无法训练出较为精准的聚类中心,并且模糊规则的训练使用全部特征,无法精简模型复杂度;模糊规则库的模糊规则激活不均匀,会出现少数几条被使用的饱和问题。以往的模型训练方法容易出现内存占用大、局部最优、梯度消失、模糊规则库冗余等训练问题。为此,本文提出一种简洁高效的训练模型算法SESSC-DL-MTSK。首先,使用SESSC对训练数据的输入特征进行划分,迭代训练出一个更精准的聚类中心,使用不同的特征空间生成弹性的模糊规则,其中每个规则只包含部分重要的特征,进一步降低了模型复杂度;其次,提出一个新型的模糊化操作MTSK,通过消除数据维度的增加对规则激活度的影响,解决了模糊规则的激活饱和问题;最后,使用深度学习的BN、MBGD、Dropout等优化方法对模型进行训练,解决了局部最优、梯度消失等问题。实验结果表明,所提算法在多个不同领域的数据集上均有较好的性能。

然而,该算法也存在一些局限性。首先,对于大规模的图像数据而言,由于TSK模型属于机器学习范畴,模型层次较为简单,可能很难学习到潜在的关键特征信息;其次,如何在提高TSK模型层次的同时保证其可解释性,这需要找到一个平衡点。在后续研究中,将尝试针对大规模图像数据构建“深度特征预提取+TSK模糊规则推理”的分层架构,在模型前端引入轻量化卷积神经网络作为特征编码器,利用其多层卷积结构自动提取图像的纹理、边缘、形状等分层特征;同时采用树状分层规则库结构替代传统扁平规则集合,实现层次化可解释性。



(a) With and without MTSK on Clave

(a) 在 Clave 上是否使用 MTSK



(b) With and without MTSK on WDBC

(b) 在 WDBC 上是否使用 MTSK

Fig. 5 Average normalized rule firing levels with and without MTSK on Clave and WDBC

图 5 在 Clave 和 WDBC 上使用和不使用 MTSK 的平均归一化规则触发水平

参考文献:

- [1] TAKAGI T, SUGENO M. Fuzzy identification of systems and its applications to modeling and control [J]. IEEE Transactions on Systems, Man, and Cybernetics, 1985(1): 116-132.
- [2] SUGENO M, TANAKA K. Successive identification of a fuzzy model and its applications to prediction of a complex system [J]. Fuzzy Sets and Systems, 1991, 42(3): 315-334.
- [3] SUGENO M, YASUKAWA T. A fuzzy-logic-based approach to qualitative modeling [J]. IEEE Transactions on Fuzzy Systems, 1993, 1(1): 7.
- [4] CHANG P C, LIU C H. A TSK type fuzzy rule based system for stock price prediction [J]. Expert Systems with Applications, 2008, 34(1): 135-144.
- [5] LOUVERDIS G, ANDREADIS I, TSALIDES P. New fuzzy model for morphological colour image processing [J]. IEE Proceedings-Vision, Image and Signal Processing, 2002, 149(3): 129-139.
- [6] BEZDEK J C, EHRLICH R, FULL W. FCM: The fuzzy c-means clustering algorithm [J]. Computers & Geosciences, 1984, 10(2-3): 191-203.
- [7] CHEN K, MA Y, JUN L. Segmentation by fusion of self-adaptive SFCM cluster in multi-color space components [J]. International Journal of Image Processing, 2012, 6(2): 157-166.
- [8] CORD O. Genetic fuzzy systems: evolutionary tuning and learning of fuzzy knowledge bases [M]. Singapore: World Scientific, 2001.
- [9] HAN M F, LIN C T, CHANG J Y. Differential evolution with local information for neuro-fuzzy systems optimisation [J]. Knowledge-Based Sys-

- tems, 2013, 44: 78–89.
- [10] LAUER F, GUERMEUR Y. MSVMpack: a multi-class support vector machine package[J]. *Journal of Machine Learning Research*, 2011, 12: 2269–2272.
- [11] JANG J S R. ANFIS: adaptive-network-based fuzzy inference system [J]. *IEEE Transactions on Systems, Man, and Cybernetics*, 1993, 23 (3): 665–685.
- [12] GACTO M J, GALENDE M, ALCALÁ R, et al. METSK-HDe: a multi-objective evolutionary algorithm to learn accurate TSK-fuzzy systems in high-dimensional and large-scale regression problems [J]. *Information Sciences*, 2014, 276: 63–79.
- [13] WANG H, YAO J. An improved deep convolutional fuzzy system for classification problems [C]//2020 IEEE International Conference on Fuzzy Systems, 2020: 1–6.
- [14] GU X, CHENG X. Distilling a deep neural network into a Takagi-Sugeno-Kang fuzzy inference system [DB/OL]. <https://arxiv.org/abs/2010.04974>.
- [15] CUI Y, WU D, HUANG J. Optimize TSK fuzzy systems for classification problems: minibatch gradient descent with uniform regularization and batch normalization [J]. *IEEE Transactions on Fuzzy Systems*, 2020, 28 (12): 3065–3075.
- [16] CUI Y, WANG H, WU D. Supervised enhanced soft subspace clustering (SESSC) for TSK fuzzy classifiers [DB/OL]. <https://arxiv.org/pdf/2002.12404>.
- [17] MATHEW A, AMUDHA P, SIVAKUMARI S. Deep learning techniques: an overview [C]//International Conference on Advanced Machine Learning Technologies and Applications, 2021: 599–608.
- [18] SHANG C, YANG F, HUANG D, et al. Data-driven soft sensor development based on deep learning technique [J]. *Journal of Process Control*, 2014, 24(3): 223–233.
- [19] GHOSH S, DAS N, DAS I, et al. Understanding deep learning techniques for image segmentation [J]. *ACM Computing Surveys (CSUR)*, 2019, 52 (4): 1–35.
- [20] IOFFE S. Batch normalization: accelerating deep network training by reducing internal covariate shift [DB/OL]. <https://arxiv.org/abs/1502.03167>.
- [21] HINTON G, SRIVASTAVA N, SWERSKY K. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent [EB/OL]. <https://www.cs.toronto.edu/~hinton/coursera/lecture6/lec6.pdf>.
- [22] SRIVASTAVA N, HINTON G, KRIZHEVSKY A, et al. Dropout: a simple way to prevent neural networks from overfitting [J]. *The Journal of Machine Learning Research*, 2014, 15(1): 1929–1958.
- [23] LUO L, XIONG Y, LIU Y, et al. Adaptive gradient methods with dynamic bound of learning rate [DB/OL]. <https://arxiv.org/abs/1902.09843>.
- [24] MORSE J M. The significance of saturation [J]. *Qualitative Health Research*, 1995, 5(2): 147–149.
- [25] GUEST G, BUNCE A, JOHNSON L. How many interviews are enough? an experiment with data saturation and variability [J]. *Field Methods*, 2006, 18(1): 59–82.
- [26] DENG Z, JIANG Y, ISHIBUCHI H, et al. Enhanced knowledge-leverage-based TSK fuzzy system modeling for inductive transfer learning [J]. *ACM Transactions on Intelligent Systems and Technology*, 2016, 8(1): 1–21.
- [27] DENG Z, JIANG Y, CHOI K S, et al. Knowledge-leverage-based TSK fuzzy system modeling [J]. *IEEE Transactions on Neural Networks and Learning Systems*, 2013, 24(8): 1200–1212.
- [28] KELLER A, KLAUWONN F. Fuzzy clustering with weighting of data variables [J]. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 2000, 8(6): 735–746.
- [29] ZHOU J, CHEN L, CHEN C L P, et al. Fuzzy clustering with the entropy of attribute weights [J]. *Neurocomputing*, 2016, 198: 125–134.
- [30] CHEN B, DENG W, DU J. Noisy softmax: improving the generalization ability of dcnn via postponing the early softmax saturation [C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017: 5372–5381.
- [31] CUI Y, WU D, HUANG J. Optimize TSK fuzzy systems for classification problems: minibatch gradient descent with uniform regularization and batch normalization [J]. *IEEE Transactions on Fuzzy Systems*, 2020, 28 (12): 3065–3075.
- [32] DU G, WANG Z, LI C, et al. A TSK-type convolutional recurrent fuzzy network for predicting driving fatigue [J]. *IEEE Transactions on Fuzzy Systems*, 2020, 29(8): 2100–2111.

(责任编辑:尹晨茹)