软件学报 ISSN 1000-9825, CODEN RUXUEW
[doi: 10.13328/j.cnki.jos.007639] [CSTR: 32375.14.jos.007639]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

面向概念漂移的在线神经网络深度自适应调整方法*

邹 迪[1], 周学文[2], 范玉雷[1], 高 楠[1], 喻 坚[3], 杨良怀[1]

[1](浙江工业大学 计算机科学与技术学院, 浙江 杭州 310023)

[2](中国人民解放军 31306 部队, 四川 成都 610000)

[3](Computer and Information Sciences Department, Auckland University of Technology, Auckland 1010, New Zealand)

通信作者: 杨良怀, E-mail: yanglh@zjut.edu.cn

摘 要: 非平稳场景下深度神经网络 (deep neural network, DNN) 模型的适应性是当前人工智能领域面临的一个重要挑战. 尤其在发生概念漂移的场景, 预设架构参数的模型将难以适应演化的数据分布. DNN 现有深度调整方法缺乏对深度拓展的有效性的评估, 忽视了深度调整过程中网络权重参数和深度的协同. 为此, 提出自适应在线深度神经网络 (adaptive online deep neural network, AODNN). AODNN 根据网络的损失变化趋势、分类器权重以及互信息变化组合分析对深度增长的有效性做出判断, 实现深度自适应增长; 并通过参数更新优化选择合适的中间分类器参与反向传播, 减少深层中间分类器对浅层分类器的干扰并加速收敛. AODNN 在多个包含概念漂移的真实和合成数据集上与当前最先进方法开展对比实验, 实验结果验证了 AODNN 深度自适应增长和参数更新优化策略的有效性, 能够有效捕捉数据分布变化并抑制概念漂移的影响. 在关键性能指标的比较中, AODNN 展现出显著优势, 在累计准确率上显著优于 HBP、ANSN 和 EODL, 在 *F*1 分数上显著优于 ATNN、ANSN 和 EODL, 在 *MCC* 上显著优于 ANSN 和 EODL. 此外, 在模型收敛速度等方面, AODNN 也超越了当前最先进方法.

关键词: 非平稳数据流; 概念漂移适应; 神经网络架构; 架构深度调整; 在线深度学习

中图法分类号: TP18

中文引用格式: 邹迪, 周学文, 范玉雷, 高楠, 喻坚, 杨良怀. 面向概念漂移的在线神经网络深度自适应调整方法. 软件学报. http://www.jos.org.cn/1000-9825/7639.htm

英文引用格式: Zou D, Zhou XW, Fan YL, Gao N, Yu J, Yang LH. Adaptive Depth Adjustment Scheme for Online Neural Networks Under Concept Drift. Ruan Jian Xue Bao/Journal of Software (in Chinese). http://www.jos.org.cn/1000-9825/7639.htm

Adaptive Depth Adjustment Scheme for Online Neural Networks Under Concept Drift

ZOU Di[1], ZHOU Xue-Wen[2], FAN Yu-Lei[1], GAO Nan[1], YU Jian[3], YANG Liang-Huai[1]

[1](College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China)

[2](Unit 3306 of the People's Liberation Army of China, Chengdu 610000, China)

[3](Computer and Information Sciences Department, Auckland University of Technology, Auckland 1010, New Zealand)

Abstract: The adaptability of deep neural network (DNN) models in non-stationary scenarios remains a significant challenge in the field of artificial intelligence. In particular, when concept drift occurs, models with predefined architectural parameters struggle to adapt to evolving data distributions. Existing depth adjustment methods for DNNs do not adequately evaluate the validity of depth expansion and overlook the synergy between network weights and depth during the adjustment process. To address this issue, this study proposes an adaptive online deep neural network (AODNN). AODNN assesses the effectiveness of depth expansion by jointly analyzing the network's loss trends, classifier weights, and changes in mutual information, thus enabling adaptive depth growth. In addition, it employs parameter update optimization to select intermediate classifiers for participation in backpropagation, reducing interference from deeper intermediate classifiers to shallower ones and accelerating convergence. Comparative experiments are conducted between AODNN and state-of-the-art

* 基金项目: 国家重点研发计划 (2022YFB3304100)
收稿时间: 2025-08-18; 修改时间: 2025-09-27, 2025-11-17, 2026-01-08; 采用时间: 2026-01-19; jos 在线出版时间: 2026-05-20

methods on multiple real-world and synthetic datasets containing concept drift. The experimental results validate the effectiveness of AODNN's adaptive depth growth and parameter update optimization strategies, demonstrating its capability to effectively capture changes in data distribution and mitigate the impact of concept drift. Across key performance metrics, AODNN demonstrates significant advantages, notably outperforming HBP, ANSN, and EODL in cumulative accuracy, surpassing ATNN, ANSN, and EODL in *F*1-score, and achieving significantly better *MCC* compared to ANSN and EODL. Furthermore, AODNN surpasses current state-of-the-art methods in aspects of model convergence speed.

Key words: non-stationary data stream; concept drift adaptation; neural network architecture; architecture depth adjustment; online deep learning

在许多实际应用中, 例如金融服务、传感器网络、气象测控等, 都会以流式方式生成大量数据[1]. 在流式数据处理的初始阶段, 大部分算法通常假定数据分布是平稳的, 但在真实世界的应用中, 数据流往往是非平稳的, 数据分布会因时间推移、环境变化或任务目标偏移而发生动态演化, 这一现象称为概念漂移[2]. 典型场景包括金融市场价格波动、短视频中用户兴趣迁移、自动驾驶车辆跨地区行驶、传感器网络环境变化等, 概念漂移可能会导致模型无法做出准确的预测, 导致性能下降[3]. 为应对概念漂移, 研究者提出了多种策略, 其中增量学习、持续学习和在线学习是 3 类主流范式. 其中, 增量学习关注数据或任务规模随时间扩展时的模型更新 (如特征/类别空间的增量式扩展); 持续学习则强调在保留旧知识的同时学习新知识, 以避免灾难性遗忘[4]. 然而, 在数据挖掘等对响应速度要求较高的流式数据处理场景中, 更重要的是快速适应当前的数据分布变化[5]. 在线学习通过逐样本或逐批次更新模型, 能够更及时地响应分布变化, 因而在处理概念漂移问题时更具优势.

在线学习框架下, 适应漂移的传统方法可以分为两类: 主动检测与被动适应. 主动检测方法通过确定概念漂移的时间和严重程度, 主动调整模型以适应漂移; 被动适应方法假设数据环境可能随时变化, 通过不断调整自身模型来适应数据分布的改变. 然而, 传统方法主要利用浅层模型, 无法适应大数据时代日益复杂多样的数据.

深度神经网络与传统的浅层模型相比, 具有更强大的表示学习能力, 可以学习更为复杂的模式和特征[6]. 现有的基于深度神经网络的概念漂移适应方法根据更新方式的不同主要可分为两类: 模型参数更新方法和模型结构更新方法. 模型参数更新方法保持网络结构固定, 仅调整网络参数以适应概念漂移. 模型结构更新方法主要通过调整网络结构来适应概念漂移, 根据调整方式的不同可以分为网络宽度调整 (如添加分支单元) 或深度调整 (如增减隐藏层). 然而, 现有的基于深度神经网络的方法仍存在以下问题: 首先, 预定义网络深度难以适配动态变化的样本复杂度, 深度过浅会限制模型表达能力, 而深度过深则会导致网络收敛缓慢和过拟合的问题; 其次, 现有深度调整方法虽能响应概念漂移的分布变化, 却缺乏对深度扩展有效性的评估, 容易导致盲目增加层数引发的过拟合或无效计算; 并且大多数工作仅关注如何增减网络层数, 却忽略在结构变化过程中对深浅层协同训练的优化, 导致新旧结构的融合效果不佳, 难以快速适应概念漂移.

为了解决深度神经网络在概念漂移数据流中的这些问题, 本文提出了自适应在线深度神经网络 (adaptive online deep neural network, AODNN), 旨在克服流数据分类任务中如何进行深度神经网络深度自适应增长以及如何提高模型面对概念漂移的适应性的问题. 在 AODNN 中, 每个隐藏层都连接着一个中间分类器, 每个中间分类器有其对应的分类器权重, 最终输出由这些中间分类器的结果加权整合而得. 为了解决在流式数据中难以为深度神经网络预设合适深度的问题, AODNN 采用了深度自适应增长机制, 该机制可以根据网络的损失变化趋势、分类器权重以及互信息变化来综合分析增加深度的合理性, 自适应地增加网络深度. 同时, 在概念漂移发生时, 数据分布的变化可能使现有网络深度不再最优. 具体而言, 模型中会存留一些因深度自适应增长机制而产生、但当前性能较差的深层分类器, 为了避免这些深层分类器在梯度的反向传播时干扰模型中浅层网络的收敛, AODNN 采取了参数优化方法, 通过调整参与反向传播的中间分类器, 减少无效梯度干扰并加快浅层网络的收敛.

本文主要贡献包括以下 3 个方面.

1) 提出自适应在线深度神经网络 AODNN, 可以快速适应不断变化的流数据分布, 能够实时监测数据分布变化并自适应更新模型的架构和权重参数.

2) 提出一种深度适应性调整策略, 自适应增加层数. 该策略根据网络的损失变化趋势、分类器权重以及互信

息变化组合分析对深度增长的有效性做出判断, 来决定是否增加网络深度, 从而找到最佳深度.

3) 提出一种权重参数更新优化策略. 根据中间分类器的分类器权重, 选出表现良好的中间分类器参与反向传播, 减少反向传播过程中深层中间分类器对浅层分类器的干扰并加速网络收敛.

本文第 1 节回顾相关技术现状. 第 2 节介绍本文所提算法. 第 3 节是性能评价. 最后, 进行总结并提出未来的研究方向.

1 相关工作

概念漂移是由流式数据随时间的变化或演变引起的, 给定一个时间段 $[0, t]$, 一个样本集合记为 $S_{0,t} = \{d_0, d_1,\ldots, d_t\}$, 其中 $d_i = (X_i, y_i)$ 是一个数据实例, X_i 是特征向量, y_i 是对应的标签, 且 $S_{0,t}$ 遵循某个联合分布 $F_{0,t}(X, y)$. 若在时间戳 t+1 处满足 $F_{0,t}(X, y) \neq F_{t+1,\infty}(X, y)$, 则称发生了概念漂移, 记为 $\exists t$: $P_t(X, y) \neq P_{t+1}(X, y)$. 概念漂移的出现使得训练好的模型不再有效, 如何解决概念漂移已成为一个关键问题, 现有的概念漂移处理方法可以划分为传统方法和基于深度神经网络的方法.

1.1 传统方法

传统方法主要分为主动检测与被动适应两类. 主动检测方法通过统计或窗口机制识别漂移并触发模型更新, 例如 DDM[7]基于分类错误率及其标准差生成警告信号, EDDM[8]则通过监测连续错误分类的平均距离变化提升检测灵敏度. HDDM[9]使用 Hoeffding 不等式来设置窗口平均值之间差值的上限进行漂移检测, 但需要对渐变漂移和突变漂移分别设定不同的阈值. 为解决这个问题, FHDDM[10]融合了统计和窗口方法, 使用滑动窗口和 Hoeffding 不等式进行概念漂移检测; FHDDMS[11]改进了 FHDDM, 通过设置一长一短两个窗口进行漂移检测, 短窗口用来应对突变概念漂移, 长窗口用于减少噪声对漂移检测的影响, 降低了漂移检测延迟和噪声的影响. DCFHT[12]基于 Hoeffding 树, 通过动态复用和重构树结构实现非线性建模与漂移适应, 并采用备份学习器策略, 检测到概念漂移时训练备用模型直至其可替代主模型, 实现漂移适应. Yu 等人[13]借助 DDM 进行概念漂移检测, 若检测到漂移, 则利用高斯混合模型评估新旧概念的分布, 并通过最大化条件概率计算新实例的重要性权重, 以适应新概念. 被动适应方法则持续更新模型以隐含适应漂移, 可以分为单学习器方法和集成学习方法. 单学习器方法中代表性工作包括基于决策树的 VFDT[14]、CVFDT[15]、RLDT[16]、基于贝叶斯模型的 WNB-CD[17]、RGNBC[18]、PGNBC[19]、基于极限学习机的算法如 OS-ELM[20]、IDS-ELM[21]、EOS-ELM[22]等. 这些算法采用单学习器进行漂移适应, 通常具有较低的复杂性和计算量, 但其泛化能力不足. 而集成方法相对于单学习器方法具有更强的泛化能力, 因此也更受关注. SEA[23]算法是最早的基于块的集成方法之一. SEA 在训练数据的连续块上构建单独的分类器, 当新分类器满足条件时则替换旧的分类器, 作者还使用了决策树进行块的构建, 使其可以更快地适应概念漂移. CDAM-APIE[24]使用在线增量集成策略构建被动集成模型, 并利用增量学习和概念漂移检测技术构建主动基模型, 提升模型在平稳数据流状态下的鲁棒性和漂移后的泛化性. IncA-DES[25]通过增量训练生成局部专家模型, 结合概念漂移检测器和重叠分类过滤器, 动态适应数据分布变化并提升计算效率, 并引入在线 K-d 树加速 k 近邻搜索, 支持快速删除旧数据并处理数据流不平衡问题. 然而, 这些传统方法大多依赖浅层模型, 在面对日益复杂的数据流时, 由于其结构限制难以捕捉复杂非线性模式, 且在处理高维动态数据时效率受限.

1.2 基于深度神经网络的方法

深度神经网络 (deep neural network, DNN) 具有强大的学习能力, 以及在适当正则化条件下实现低泛化误差的能力. 根据应对概念漂移方式的不同, 可以将基于深度神经网络的方法分为模型参数更新方法和模型结构更新方法[26,27]. 模型参数更新方法指的是不改变网络结构只对网络参数进行更新来使得网络适应概念漂移. Sahoo 等人[28]提出了 HBP 网络, HBP 通过将每个隐藏层都连接到一个中间分类器并对中间分类器进行加权融合被动适应漂移, 但在网络层数较多时收敛速度缓慢. EODL[29]改进了 HBP 算法, 通过中间分类器的预测结果动态的选择和训练子网络, 提高了模型训练的收敛速度, 并实时调整位于不同深度的子网络的学习率, 从而减轻了反向传播中因中间分类器预测分歧而导致的优化冲突. SEOA[30]则将浅层特征与深层特征相结合构建自适应深度单元缓解特征退化问

题, 并根据流数据的波动动态选择基分类器, 进行选择性集成. BODL[31]采用了双层学习机制适应概念漂移, 当检测到概念漂移时, BODL 通过初始化记忆权重来重放记忆中的知识, 并应用这些权重来更新模型参数, 以适应新环境. 然而, 此类方法仍受限于固定网络结构, 难以适配复杂的动态数据.

模型结构更新方法主要通过调整网络结构来适应概念漂移, 根据调整方式的不同可以分为网络宽度调整和网络深度调整. 网络宽度调整方法在面对概念漂移时网络深度保持不变, 通过添加新的分支或单元来修改网络宽度进行增量学习, 以实现概念漂移适应. Kauschke 等人[32]提出了神经网络修补, 通过训练一个错误分类器来识别分类器发生误分类的错误区域, 并利用先前训练的网络的内层及其输出来学习增强分类的补丁来适应概念漂移. DEVDAN[33]在去噪自动编码器 (DAE) 基础上进行构建, 拥有灵活的网络结构, 通过对隐藏单元进行插入或删除来实现对概念漂移的适应. 以上这两种方法都是通过调整网络宽度来进行概念漂移适应, 但根据 Elda 等人[34]研究表明, 增加网络深度比增加网络宽度更能增强神经网络的泛化性能. 因此, 在面对概念漂移时网络深度调整方法拥有更好的表现.

网络深度调整方法通过动态增减网络的层数以增强模型对概念漂移的适应性. 自主深度学习 (autonomous deep learning, ADL)[35]在检测到概念漂移时增加隐藏层, 并根据网络显著性 (network significance, NS) 公式改变隐藏节点的数量. 堆叠式自动编码器深度神经网络 (SAE-DNN)[36]使用随机向量功能链接结构来对神经网络进行扩展, 并且扩展层的参数由使用组套索正则化 (group lasso regularization) 和 L2 正则化的新传入数据决定. NADINE[37]则通过结合了自适应窗口机制和 Hoeffding 界检测方法进行概念漂移检测, 当检测到概念漂移时, NADINE 通过增加新的隐藏层来进行适应, 使网络有更多的局部区域来捕捉数据中的复杂模式, 从而提高泛化能力. 然而仅依据概念漂移检测结果来触发深度扩展, 在频繁漂移的数据流中可能导致网络深度盲目增加, 进而引发过拟合. ANSN[38]在检测到概念漂移时增加网络层数, 当检测到多层权重低于阈值的隐藏层时则会对隐藏层进行合并, 并对权重最大的隐藏层进行宽度调整, 由 NS 决定是否增加或删除隐藏节点. DeDNN[39]在 NADINE 基础上进行改进, 通过 JS 散度进行深度增长判定, 并通过局部误差和 JS 散度进行神经元增长, 使其在处理非平稳数据流时具有更强的适应性, 并采用了自适应记忆机制抵抗灾难性遗忘, 但当模型的结构发生较大变化时, 特别是当新的数据分布与旧的分布差异较大时, 自适应记忆机制可能无法完全保留所有重要的历史信息. IADM[40]通过注意力机制来实现深度自适应, 并通过 Fisher 正则化来减轻灾难性遗忘. ATNN[41]则是在 HBP 的基础上通过多分支结构来适应概念漂移, ATNN 使用两个大小不同的滑动窗口来检测概念漂移, 较大的窗口用于计算整体损失, 而较小的窗口用于计算当前损失, 损失最低的分支会作为活动分支输出预测结果. 如果当前损失显著高于整体损失, 则认为发生了概念漂移, 当所有分支的损失都达到漂移水平时则认为出现了新概念, 将会添加一条新的分支到主干网络, 这种策略使得网络在面对重复漂移时能有出色的表现. 这些方法通过是否发生概念漂移来决定是否进行深度或分支的增长, 当概念漂移频繁发生时模型会不断添加新的分支和隐藏层以适应这些变化, 网络规模会显著增加, 导致模型运行时间大幅延长, 难以有效地控制模型的规模.

为了避免因盲目增加网络深度引发过拟合的风险, 以及模型参数更新与结构更新协同不足的问题, AODNN 在神经网络深度控制方面提出了更为精细的决策机制. 与现有方法通常仅依赖概念漂移检测结果直接触发深度扩展不同, AODNN 采用损失趋势、分类器权重和互信息组合判断深度增长的合理性, 该策略有效抑制了因频繁漂移而导致的深度无效扩张, 从而在结构复杂性、泛化能力与计算效率之间取得更好平衡. 此外, AODNN 通过参数更新优化动态控制实际参与更新的中间分类器规模, 加快了模型收敛速度, 实现深度增长与参数调整的协同优化.

2 自适应在线深度神经网络 AODNN

AODNN 是基于动态深度调整的在线深度神经网络模型, 其架构如图 1 所示. 模型具有 L 层隐藏层, 其中每个隐藏层都连接到一个分类器, 也称为中间分类器, 用于输出该层的预测结果. 每个中间分类器分配一个动态调整的分类器权重 α. 当 t 时刻数据流中数据 x_t 到来时, 通过对中间分类器的预测结果 $f^{(l)}(x_t)$ (l = 1, 2,…, L) 加权求和得到最终预测结果 $F(x_t)$. 在反向传播时对分类器权重 $\alpha_t^{(l)}$、隐藏层权重 $w_{h,t}^{(l)}$ 以及中间分类器的权重 $w_{o,t}^{(l)}$ 进行更新. 可知, AODNN 的中间分类器的权重不是固定不变的, 而是动态调整的.

AODNN 前向传播公式如下所示:

$$F(x_t)=\sum_{j=1}^{L}\alpha_t^{(j)}f^{(j)}(x_t) \tag{1}$$

$$f^{(l)}(x_t)=Softmax\left(h^{(l)}w_{o,t}^{(l)}\right) \tag{2}$$

$$h^{(l)}=\sigma\left(w_{h,t}^{(l)}h^{(l-1)}\right) \tag{3}$$

$$h^{(0)}=x_t \tag{4}$$

其中, $l=1,2,\ldots,L$, L 为网络深度. $h^{(0)}$ 为输入层, $h^{(l)}$ 是第 l 层隐藏层, 函数 $\sigma(\cdot)$ 表示隐藏层使用的激活函数, $\alpha_t^{(l)}$ 表示第 l 层中间分类器的分类器权重. 最终预测结果 $F(x_t)$ 由每个中间分类器的预测结果 $f^{(l)}(x_t)$ 和分类器权重 $\alpha_t^{(l)}$ 加权求和而得到的.

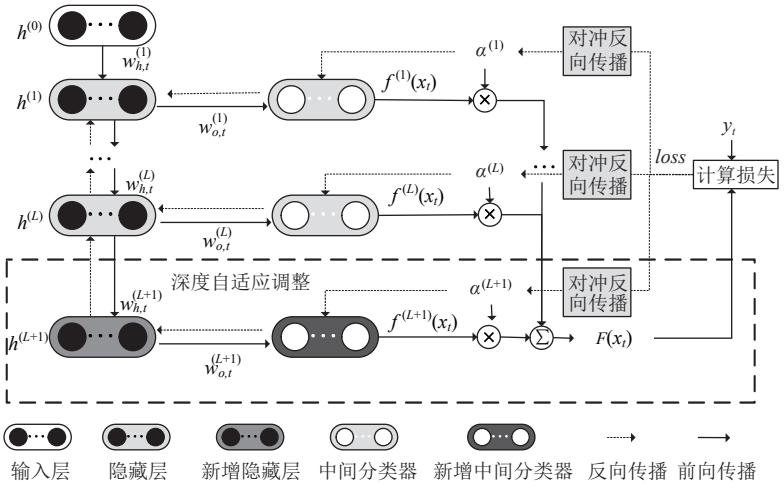

图 1 AODNN 模型框架图

自适应在线深度网络 AODNN 的适应能力和在线学习能力通过深度自适应增长机制和参数更新优化策略来实现.

2.1 深度自适应增长机制

不同深度的神经网络具有不同的收敛和数据拟合能力, 当数据更复杂时, 多层的网络结构比浅层更为有效[42]. 在非平稳的数据流中, 概念漂移的发生导致事先预设深度的网络难以适应数据分布的不断改变, 因此需要有一种动态调整神经网络架构的机制来保障模型推理性能. 是否增加网络深度的决策关键在于判断网络的性能瓶颈是否在模型的容量上. AODNN 通过损失变化的趋势分析判断网络性能是否不再继续增长, 即网络达到瓶颈期; 一旦确认瓶颈期, 则通过分类器权重、特征互信息分析判断增加网络规模能否进一步提高网络性能来决定是否进行深度增长.

2.1.1 损失变化趋势分析

损失函数作为模型预测误差的直接量化指标, 能够实时反映当前网络结构对数据分布的拟合能力, 损失变化趋势分析的任务是判断模型是否进入性能瓶颈期. 当损失值在一段时间内趋于稳定且无明显变化趋势时 (即进入平台期), 通常表明现有网络已达到局部最优或是当前网络架构可能不足以捕捉数据的复杂模式, 此时的一种改进方向是进一步分析网络深度增加是否可以提升网络性能. 然而, 在非平稳数据流的场景中, 损失值源于多种因素而发生波动: 一方面, 概念漂移会导致数据分布发生变化, 引发损失的短期震荡; 另一方面, 随机噪声或数据的局部特性也可能造成损失发生波动. 因此, 需要采用合理的方式来对损失变化趋势进行分析, 避免在模型仍有优化空间时过早增加深度.

AODNN 采用滑动窗口分位数差异分析来检测平台期. 分位数统计量的优势在于能够捕捉损失分布的整体偏移趋势, 并且对极端值不敏感, 更适用于非平稳数据环境[43]. 例如, 若新窗口的损失分布整体右移 (即高损失值比例增加), 即使均值不变, 分位数差异仍能反映性能下降. 若仅通过均值差异进行判断, 在损失分布形态变化但均值稳定时则可能失效. 具体而言, AODNN 通过维护如图 2 所示的两个大小固定为 W 的滑动窗口 L_{new} 与 L_{old} 用于记录新、旧损失值, 每个窗口存储固定数量的损失值, 窗口未满时可加入新损失值, 窗口满时加入新损失值的同时移出最旧的一个损失值. 新旧窗口均未满时则认为未进入平台期, 窗口均满后则计算其损失分位数差异:

$$\Delta Q_p = Q_p(L_{\text{new}}) - Q_p(L_{\text{old}}) \tag{5}$$

其中, $Q_p(\cdot)$ 为 p 分位数, 动态阈值 τ_t^Q 通过指数加权移动平均 (exponentially weighted moving average, EWMA) 平滑历史分位数差异来平衡短期波动与长期趋势, 具体公式如下:

$$\tau_t^Q = \gamma\tau_{t-1}^Q + (1-\gamma)\Delta Q_p \tag{6}$$

其中, $\gamma\in(0, 1)$ 为平滑因子, 用于控制阈值更新的灵敏度. 若最新窗口的损失分位数未显著高于旧窗口, 即损失分位数差异 $\Delta Q_p < \tau_t^Q$, 表明损失未发生实质性变化, 进入平台期, 则进行分类器权重分析; 否则损失分位数差异 $\Delta Q_p \geqslant \tau_t^Q$, 未进入平台期, 继续进行损失变化趋势分析.

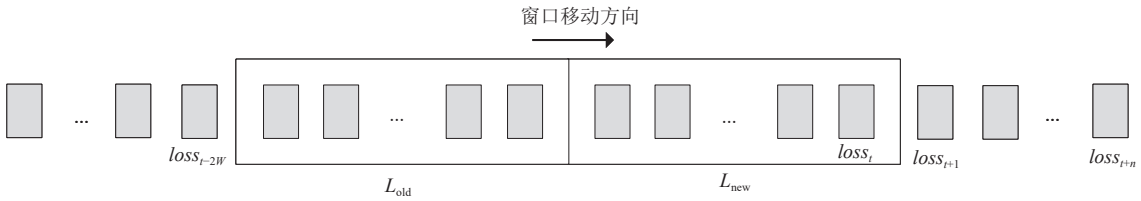

图 2 AODNN 新旧滑动窗口

2.1.2 分类器权重分析

根据 Sahoo 等人[28]的观察, 在概念漂移发生的初期, 当数据流规模较小且分布尚未稳定时, HBP 浅层网络因其参数规模小、优化路径短, 能够快速收敛, 此时浅层中间分类器的分类器权重 α 显著偏高. 当数据流平稳时, 样本量的积累使得数据内在复杂度逐渐显现, 浅层的中间分类器由于网络结构简单学习能力受到限制, 此时深层中间分类器的性能将会逐渐提升, 其分类器权重 α 随性能提升而升高, 推动分类器权重分布的峰值向深层迁移. 值得注意的是, 这种迁移并非无限制地向末端层偏移, 而是最终在中间层类似于正态分布的形式, 这种形态表明模型通过分类器权重对冲反向传播更新机制自发筛选出与当前数据规模匹配的深度区间. 然而, 这种方法要求模型预设足够的深度, 以确保分类器权重能在对冲反向传播的过程中自发收敛到合适的深度区间. 随着预设深度的增加, 模型反向传播更新的耗时也将成倍增长, 难以满足在线学习对效率的要求. 尽管如此, 分类器权重 α 的分布变化规律仍可作为判断当前网络深度是否充足的重要依据.

因此, AODNN 通过对当前最后一层分类器权重是否超过动态阈值 τ_t^α 来判断分类器权重的分布是否已经移动到深层, 动态阈值 τ_t^α 通过如下方式计算, 首先计算当前所有分类器权重 $\alpha_t = \left(\alpha_t^{(1)}, \alpha_t^{(2)}, \ldots, \alpha_t^{(L)}\right)^{\mathrm{T}}$ 的均值 μ_t^α 与标准差 σ_t^α:

$$\mu_t^\alpha = \frac{1}{L}\sum_{l=1}^{L}\alpha_t^{(l)} \tag{7}$$

$$\sigma_t^\alpha = \sqrt{\frac{1}{L}\sum_{l=1}^{L}\left(\alpha_t^{(l)} - \mu_t^\alpha\right)^2} \tag{8}$$

动态阈值 τ_t^α 设置为:

$$\tau_t^\alpha = \mu_t^\alpha + c^\alpha \cdot \sigma_t^\alpha \tag{9}$$

其中, c^α 为敏感度常数, $c^\alpha\in[0, 3]$, 同时 $\tau_t^\alpha = \min(\tau_t^\alpha, 0.5)$. 若 $\alpha_t^{(L)} > \tau_t^\alpha$, 表明深层分类器已承担主要预测责任, 但模型性能进入平台期且无法进一步提高, 说明现有深度可能不足以捕捉数据复杂度, 此时触发互信息分析.

2.1.3 模型互信息分析

互信息已被应用到深度神经网络架构的优化设计中[44,45]. 深度神经网络深度的优化可以归结为关于输入 X 与隐藏层互信息的最小化和关于期望输出 Y 与隐藏层互信息的最大化. 在非平稳数据流中, 深度需要依据数据分布的变化自适应调整, 本节提出了一种基于互信息变化趋势的启发式方法. 本节先引出 DNN 架构互信息分析的原理, 然后给出我们的解决方法.

一个示意性前馈 DNN 如图 3 所示, 其中 $X\in\mathbb{X}$ 是输入, $Y\in\mathbb{Y}$ 是期望输出, $h^{(1)}$, $h^{(2)}$,…, $h^{(L)}$ 是 L 个隐藏层, $\hat{Y}$ 是网络的输出. DNN 模型学习的本质是获取紧凑数据表征, 这与互信息紧密相关, 其理论可归结到 Tishby 等人[46]提出的信息瓶颈 (information bottleneck, IB) 原理. 信息瓶颈原理用于从输入随机变量 X 中提取与输出随机变量 Y 相关的信息. 假设 X 和 Y 之间存在统计相关性, 给定 X 与 Y 的联合分布 $p(X, Y)$, 相关信息被定义为互信息 $I(X;Y)$. 此时, Y 隐式地决定了 X 中的相关特征与无关特征. 关于 X 的最优表征应能捕捉相关特征, 并通过舍弃那些对预测 Y 无贡献的无关部分来实现对 X 的压缩. 深度神经网络学习过程可通过信息瓶颈理论进行分析, 每一层都力求最大化保留与输出标签相关的信息, 同时最小化其与前一表征层之间的信息. 因此, 每一层都能确保既对输出具有最强的判别性, 又能对输入实现最具代表性的表征.

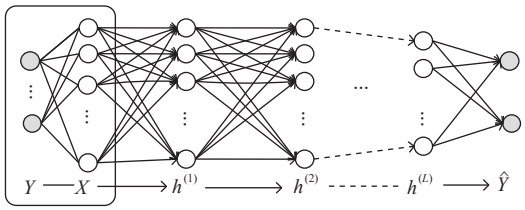

图 3 一个示意性深度神经网络

由多层神经网络结构可知, 两个隐藏层之间直接依赖, 即 $h^{(l)}$ 仅依赖于上一层 $h^{(l-1)}$, DNN 的各层形成了马尔可夫级联中间表征 $x=h^{(0)}\rightarrow h^{(1)}\rightarrow\ldots\rightarrow h^{(L)}$. 显然它满足数据处理不等式 (data processing inequality, DPI)[47]的条件, 从而依据数据处理不等式: 对于任何构成马尔可夫链 $X\rightarrow Y\rightarrow Z$ 的 3 个变量有 $I(X;Y)\geqslant I(X;Z)$, 可以得到 $I(X;h^{(1)})\geqslant I(X;h^{(2)})\geqslant\ldots\geqslant I(X;h^{(L)})\geqslant I(X;\hat{Y})$. 另外, 根据互信息链式规则可以证明对于任何构成马尔可夫链 $X\rightarrow Y\rightarrow Z$ 的 3 个变量有 $I(X;Y)\geqslant I(Y;Z)$; 应用到 DNN 中可以得到 $I(h^{(0)};h^{(1)})\geqslant I(h^{(1)};h^{(2)})\geqslant\ldots\geqslant I(h^{(L-1)};h^{(L)})\geqslant I(h^{(L)};\hat{Y})$. 因此, 互信息在深度神经网络中有以下结论成立.

命题 1. 对于一个深度神经网络, 其输入层 $h^{(0)}=X$, 隐藏层为 $h^{(1)}$, $h^{(2)}$,…, $h^{(L)}$, X 是输入, Y 是期望输出, $\hat{Y}$ 是网络的输出. 则输入与各层之间的互信息逐层递减 $I(X;h^{(1)})\geqslant I(X;h^{(2)})\geqslant\ldots\geqslant I(X;h^{(L)})\geqslant I(X;\hat{Y})$, 各层间的互信息随层数增加递减 $I(h^{(0)};h^{(1)})\geqslant I(h^{(1)};h^{(2)})\geqslant\ldots\geqslant I(h^{(L-1)};h^{(L)})\geqslant I(h^{(L)};\hat{Y})$.

在非平稳数据流中, 随着数据分布的变化, 原来优化的深度可能此时已非最优, 需要捕捉调整深度的指示量, 最后两个隐藏层间互信息的变化情况正是一个能反映数据表征紧凑程度的关键指标. 根据前述命题 1, 各层与输入之间的互信息应随层级加深而递减, 即越靠后的层应形成更为紧凑的表征. 因此, 如果某一时刻的最深层 $h^{(L)}$ 相对于上一层 $h^{(L-1)}$ 仍能形成更紧凑的数据表征, 则该层在当前数据分布下仍然是有价值的; 否则说明该层可能是冗余的, 未带来结构优化.

基于这一观察, 所提方法 AODNN 采用启发式的方法, 将 DNN 的各层视为马尔可夫级联中间表征 $x_t=h^{(0)}\rightarrow h^{(1)}\rightarrow\ldots\rightarrow h^{(L)}$, 其中 x_t 是当前 t 时刻的输入, 通过考察最后两层输出的互信息差$\Delta I_t=I(x_t;h^{(L-1)})-I(x_t;h^{(L)})$ 变化的趋势判定当前深度合理性, 若ΔI_t 较大, 表示深层 $h^{(L)}$ 相对 $h^{(L-1)}$ 仍然实现了显著的信息压缩, 即深层表征更加紧凑; 若该值持续偏小, 则说明最深层无法带来更多有效表征, 可能已成为结构冗余.

然而, 仅依赖瞬时的互信息差值并不足以稳定反映层间压缩能力. 在非平稳数据流中, 数据分布会持续漂移, 互信息的单次估计容易受到噪声、局部异常及样本波动的影响, 从而产生短时的大幅起伏. 如果直接根据某一时刻的ΔI_t 做深度调整, 将导致模型结构对瞬时扰动过于敏感, 甚至可能频繁且不必要地改变深度. 为更加稳健地对

深层压缩表征能力进行判断, AODNN 引入滑动窗口机制对互信息差进行平滑, 并基于其统计特征构建自适应阈值. 通过大小为 W 的窗口记录历史ΔI_t 值, 计算窗口中的历史均值 μ_t^I 与方差 σ_t^I, 计算公式如下:

$$\mu_t^I = \frac{1}{W}\sum_{j=1}^{W}\Delta I_{t-j+1} \tag{10}$$

$$\sigma_t^I = \sqrt{\frac{1}{W}\sum_{j=1}^{W}(\Delta I_{t-j+1}-\mu_L)^2} \tag{11}$$

据此构建ΔI_t 的自适应阈值为 $\tau_t^I = \mu_t^I + c^I\cdot\sigma_t^I$, 用于判断深层压缩表征能力, 其中 c^I 为敏感度系数, 当 $\Delta I_t > \tau_t^I$ 时, 表明在当前数据分布下深层有效紧凑表征了输入特征, 但是否应当增加模型深度以提升紧凑度, 则需要进入综合决策阶段来决定.

2.1.4 模型深度增长

经过前面损失变化趋势分析、分类器权重分析、网络模型互信息分析后, 若同时出现模型损失进入平台期(即模型整体损失无法下降)、最后一层分类器权重超过动态阈值 τ_t^α (表明深层拥有较高的预测性能) 以及最后两层互信息的差异 ΔI_t 大于阈值 τ_t^I (表明当前深度下模型能捕捉有效特征), 则说明现有模型的表征能力趋于饱和, 表达能力需要进一步增强, 输入变量表征紧凑度需要进一步提升. 此时, 可以通过增加新的网络层来增强模型紧凑表征输入特征, 从而降低损失. AODNN 基于这个思想扩展模型深度, 在 L 层后增加新的隐藏层及其附加的中间分类器, 新隐藏层的参数 $w_h^{(L+1)}$ 和中间分类器的参数 $w_o^{(L+1)}$ 由随机函数随机初始化, 同时新的分类器权重 $\alpha_t^{(L+1)}$ 初始化为 $\frac{1}{L+1}$ 并归一化, 并更新最大层数.

2.2 参数更新优化

深度自适应增长方法有效解决了数据流平稳阶段浅层网络学习能力受限、难以拟合复杂数据的问题, 也避免了为深度神经网络预设合适深度的困难. 然而, 在非平稳数据流中, 概念漂移的发生会导致已习得的分类器权重与网络参数失效, 网络深度也可能不再适应新的数据分布. 由于新分布下的训练样本有限, 浅层分类器往往收敛更快, 在模型优化中能获得较高权重, 使得深层分类器在新环境中的贡献有限. 同时, 过深的网络结构不仅容易引发过拟合, 还会降低模型泛化性能[35]. 此外, 网络中可能存在多个分类性能较弱的深层分类器, 其梯度在反向传播过程中可能会干扰浅层网络的收敛过程[41]. 针对上述问题, 本文提出一种参数更新优化策略, 通过动态调整参与反向传播的中间分类器, 减少深层分类器对浅层网络的干扰, 从而加快模型收敛速度并降低运行时间开销.

在算法 AODNN 中, 当获取到数据的标签 y_t 后, 通过损失函数计算最终预测结果 $F(x_t)$ 与标签 y_t 之间的损失, 并采用在线梯度下降法更新网络参数. 为了避免深层分类器对浅层网络的干扰, AODNN 选择贡献大的深层分类器参与反向传播. 若某深层分类器的权重 $\alpha_t^{(l')}$ 大于预设阈值 $\xi\in(0, 1)$, 则认为其预测性能良好, 将其纳入反向传播过程. 隐藏层的参数 $w_h^{(l)}$ 只取决于从 l 层开始分类器权重大于阈值 ξ 的中间分类器 $f^{(l')}$, 具体公式如下:

$$w_{h,t+1}^{(l)} = w_{h,t}^{(l)} - \eta\sum_{l'\in L^*}\alpha_t^{(l')}\nabla_{w_{h,t}^{(l)}}\mathcal{L}\left(f^{(l')}(x_t), y_t\right) \tag{12}$$

其中, $\mathcal{L}(\cdot,\cdot)$ 表示损失函数, $L_l^* = \{l' \mid \alpha_t^{(l')} > \xi, l'\in[l, L]\}$, 特别地, 当 L_l^* 为空时, 即没有中间分类器权重 $\alpha_t^{(l)}$ 大于给定阈值 ξ 时, $L_l^* = \{l_{\text{best}}, l_{\text{best}}+1\}$, 其中 l_{best} 为 $\alpha_t^{(l)}$ 最大的中间分类器的层数, 若 l_{best}=L, 则令 $L_l^* = \{l_{\text{best}}\}$. 当 L_l^* 只有包含一个元素 l' 且 $l'\neq L$ 时, 令 $L_l^* = \{l', l'+1\}$, 以保证 L_l^* 中至少包含两个元素, 避免中间某一层分类器权重过高导致权重分布无法向深层移动. 中间分类器的参数 $w_o^{(l)}$ 的更新取决于当前层的损失, 具体公式如下:

$$w_{o,t+1}^{(l)} = w_{o,t}^{(l)} - \eta\alpha_t^{(l)}\nabla_{w_{o,t}^{(l)}}\mathcal{L}\left(f^{(l)}(x_t), y_t\right) \tag{13}$$

分类器权重 $\alpha_t^{(l)}$ 根据对应的中间分类器的预测结果 $f^{(l)}$ 与标签之间的损失采用对冲反向传播算法进行更新, 具体公式如下:

$$\alpha_{t+1}^{(l)} = \alpha_t^{(l)}\beta^{\mathcal{L}\left(f^{(l)}(x_t), y_t\right)} \tag{14}$$

其中, $\beta\in[0, 1]$ 是衰减因子. 为避免将表现较差的分类器的权重降至太低影响网络参数更新, 引入平滑参数 s, 用于

为每个分类器设置最小权重. 每次迭代更新分类器的权重后, 将权重设置为:

$$\alpha_{t+1}^{(l)} = \max\left(\alpha_t^{(l)}, \frac{s}{L}\right) \tag{15}$$

调整完以后, 再对 $\alpha_{t+1}^{(l)}$ 进行归一化, 即使得 $\sum_{l=1}^{L}\alpha_{t+1}^{(l)} = 1$.

与 HBP 相比, AODNN 的动态适应性体现在两方面: 其一, 通过多维度监控机制实现对网络深度自动增长. AODNN 基于实时损失变化趋势判断、分类器权重分析与特征互信息分析进行联合决策, 仅在增加网络深度能明确提升网络表现时扩展网络深度. 其二, 反向传播过程采用选择性参数更新策略, 通过权重阈值筛选预测能力强的中间分类器, 仅让选中的中间分类器参与浅层参数梯度更新, 从而在保证收敛速度的同时降低计算开销. 通过对参与反向传播的中间分类器的选择, 不仅加快了浅层网络收敛, 也减少了网络计算量, 提高了网络计算速度. 在概念漂移发生时, 因浅层分类器收敛速度更快, AODNN 可更早地适应新出现的概念. 而在数据分布稳定的情况下, 深层分类器可以更好地拟合复杂的数据分布, 提升模型预测性能, 会逐步增大深层分类器的权重, 所提深度自适应增长机制会逐步增加网络深度, 直至达到合适的网络深度.

3 实 验

3.1 实验数据与参数设置

在实验中, 本文使用了 6 个合成数据集和 6 个真实数据集, 具体信息如表 1 所示, 其中合成数据集使用 MOA 数据生成器[48]生成, 包含突变、渐变和增量这 3 种类型的概念漂移, 真实数据集均来自 UCI[49], 其概念漂移类型、漂移位置、漂移数都是未知的.

表 1 实验使用的数据集

数据集	特征数	类别数	数据量 (k)	漂移位置	漂移类型	漂移数
Agrawal	9	2	100	25k-50k-75k	突变	3
Sea	3	2	100	25k-50k-77k	突变	3
Hyperplane	10	2	100	—	增量	—
LED_abrupt	24	10	100	25k-50k-75k	突变	3
LED_gradual	24	10	100	25k-50k-76k	逐渐	3
RBF	10	10	100	25k-50k-78k	重复	3
Weather	8	2	18.1	—	—	—
Electricity	6	2	45.3	—	—	—
Kddcup99	41	23	494	—	—	—
Covtype	54	7	581	—	—	—
Airlines	7	2	539	—	—	—
Poker-hand	11	10	1 025	—	—	—

为了验证 AODL 模型的性能相对其他方法的表现, 需要与当前最先进的方法进行各方面的比较, 对比方法除了包括模型结构调整方法以外还应包括模型参数更新方法, 本文选取了 4 种目前最优的方法 EODL[29]、ATNN[41]、ANSN[38]、HBP[28]进行对比, 而对于其他基于深度神经网络的方法 ADL[35]、IADM[40]、DeDNN[39]、SEOA[30]和基于传统方法的 BELS[50]、AiRStream[51]、CAND[52]都已在文献 [29,41] 中进行比较, 因此本文不再进行重复实验.

所有算法均采用 ReLU 函数作为激活函数, Softmax 作为输出层函数, 采用交叉熵函数作为损失函数, 算法的超参数设置如下.

● EODL[29]通过模型参数更新适应概念漂移, 根据中间分类器的分类器权重 α 的方差是否大于阈值 θ 来判断是否发生了概念漂移, 根据漂移检测结果来动态选择和训练子网络, 并实时调整位于不同深度的子网络的学习率, 使其能够适应概念漂移, α 的衰减率采用指数加权移动平均进行更新. 模型深度设置为 20, 隐藏层的神经元节点数设置为 100, 衰减因子 β=0.8, 平滑因子 s=0.2, 学习率 η=0.01, 指数移动平均的加权系数 p=0.99, 概念漂移阈值

θ=0.01.

● ATNN[41]则是通过模型结构更新适应概念漂移, 通过新旧窗口检测整体损失和当前分支损失偏差判断是否发生概念漂移, 在概念漂移发生时通过增加分支来学习新出现的概念, 保证每个分支都只在其对应的概念上进行训练, 同时采用 Fisher 正则化约束主干网络参数更新, 克服灾难性遗忘的发生; 其新增分支只有一层隐藏层, 通过判断最后一层中间分类器是否超过阈值来决定是否进行深度增长. 衰减因子 β=0.99, 平滑因子 s=0.2, 学习率 η=0.02, 阈值 δ_1=0.85, δ_2=0.7, 整体损失窗口大小 W_o=500, 当前损失窗口大小 W_r=50, 正则化参数 λ=5000.

● ANSN[38]通过检测数据块间准确率变化识别概念漂移, 并自适应地增加/合并隐藏层、增减隐藏节点, 动态调整网络深度与宽度来适应漂移. 概念漂移警告参数 α_W=0.0005, 概念漂移发生参数 α_D=0.0001, 权重更新步长 ξ=0.001, 数据块大小 S=500.

● HBP[28]提出将每一层隐藏层连接到中间分类器, 并对每层中间分类器的输出结果进行加权集成来进行分类. 模型深度设置为 5, 隐藏层的神经元节点数设置为 100, 衰减因子 β=0.99, 平滑因子 s=0.2, 学习率 η=0.01.

● AODNN, 本文提出的方法. 隐藏层的神经元节点数设置为 100, 模型深度自适应增长, 互信息的计算至少需要两层隐藏层, 因此初始隐藏层数量设置为 2, 衰减因子 β=0.99, 平滑因子 s=0.2, 学习率 η=0.01, 阈值 τ_t^O 初始值为 0, 阈值 ξ 初始值为 0.3, 平滑因子 γ=0.9, 分位数 p=0.75, 敏感度常数 c^{α}=1.5, c^{I}=1, 滑动窗口大小 W=100. 为避免互信息计算开销过大的问题, 在 AODNN 中采用 MINE 方法[53]进行计算.

每个实验重复 10 次并计算平均评价指标, 以确保结果的可靠性. 所有实验均在 Windows 11 操作系统上运行, CPU 为 Intel Core i5-13500 2.5 GHz, 内存 32 GB.

3.2 评价指标

模型评价指标采用 3 项: $F1$ 分数 ($F1$)、马修斯相关系数 (MCC) 与累计准确率 ($Cacc$). $F1=2\times\dfrac{\text{精确率}\times\text{召回率}}{(\text{精确率}+\text{召回率})}$, 其中召回率=$TP$/($TP$+$FN$), 精确率=$TP$/($TP$+$FP$), 真阳性 ($TP$) 为模型将属于某个类别的样本正确分类为该类别的次数, 假阳性 (FP) 为模型将不属于某个类别的样本错误分类为该类别的次数, 假阴性 (FN) 为模型将属于某个类别的样本错误分类为非该类别的次数.

马修斯相关系数 (MCC): 考虑了所有 4 个类别 (TP、FP、TN、FN) 的计数, 并对它们进行综合评估, 计算方法为: $MCC=\dfrac{TP\times TN-FP\times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$, 其中真阴性 ($TN$) 为模型将属于某个类别的样本错误分类为该类别的次数.

累计准确率 ($Cacc$): 累计准确率用于评估算法在数据集上的整体性能, 其计算方法为: $Cacc=\dfrac{1}{T\cdot n}\sum_{t=1}^{T}n_t$, 其中 n 为当前时间 T 的样本总数, n_t 表示 t 时间戳上预测正确的样本总数.

3.3 实验结果及分析

3.3.1 分类性能对比

表 2 展示了 AODNN 以及其他方法的累计准确率、$F1$ 分数和 MCC 值, 其中加粗为最优值,下划线为次优值. 累计准确率可以展示方法在数据集上的整体表现, AODNN 在除了 RBF、Airlines 和 Covtype 以外的数据集中均取得最高累计准确率. AODNN 在除了 Electricity 数据集以外均取得了最高的 $F1$ 分数. AODNN 在大多数数据集上都取得了最佳表现. 为了计算所有方法之间的显著性差异, 使用 Bonferroni-Dunn 检验[54]计算所有方法之间的临界值差异 (critical difference, CD), 当两种方法在所有数据集上平均排名的差异大于 CD 值时, 说明这两种方法存在显著差异. 以 α=0.05 作为显著性水平, 计算得到 CD 值均为 1.76, 结果如图 4–图 6 所示. 实验结果表明在累计准确率上 AODNN 显著优于 HBP、ANSN 和 EODL, 在 $F1$ 分数上 AODNN 显著优于 ATNN、ANSN 和 EODL, 在 MCC 上显著优于 ANSN 和 EODL. 这表明 AODNN 在非平稳数据流中具有良好的适应性学习能力, 通过自适应深度增长与参数更新优化相互协同, 能够有效捕捉数据分布变化并抑制概念漂移的影响.

表 2 AODNN 及其他方法的累计准确率、*F*1 分数和 *MCC* 值

Datasets	累计准确率 (%)					*F*1分数					*MCC*值				
	HBP	ANSN	EODL	ATNN	AODNN	HBP	ANSN	EODL	ATNN	AODNN	HBP	ANSN	EODL	ATNN	AODNN
Agrawal	88.4	87.3	83.9	87.7	**89.4**	0.88	0.88	0.83	0.88	**0.89**	0.77	0.74	0.66	0.75	**0.79**
Sea	88.1	88.3	88.1	86.8	**88.6**	0.87	0.87	0.86	0.86	**0.88**	0.74	0.74	0.72	0.72	**0.75**
Hyperplane	91.3	90.1	91.5	90.6	**91.6**	**0.92**	0.91	0.91	0.91	**0.92**	0.81	0.80	0.82	0.81	**0.83**
LED_abrupt	72	64.5	70.7	**72.9**	**72.9**	**0.73**	0.63	0.66	**0.73**	**0.73**	0.67	0.59	0.63	0.69	**0.70**
LED_gradual	70.7	69.4	69.9	**71.6**	**71.6**	0.71	0.69	0.66	**0.72**	**0.72**	0.65	0.66	0.63	0.67	**0.68**
RBF	78.3	78.4	82.7	**88**	85.2	0.49	0.73	**0.74**	0.67	**0.74**	0.81	0.74	0.78	**0.87**	0.83
Weather	78.4	78.6	77.7	75.6	**78.8**	0.88	0.85	0.85	0.83	**0.89**	0.47	0.49	0.50	**0.52**	0.48
Electricity	88.8	76.4	86.5	88.9	**89.2**	0.6	0.76	0.39	**0.84**	0.58	0.77	0.52	0.69	0.77	**0.78**
Kddcup99	99.8	94.8	99.3	**99.9**	**99.9**	0.77	0.14	0.52	0.62	**0.89**	0.99	0.91	0.99	**1.00**	**1.00**
Covtype	88.1	72.7	84.5	**94.1**	93.7	0.88	0.72	0.83	0.7	**0.89**	0.89	0.56	0.61	0.88	**0.90**
Airlines	61.6	61.8	57.5	**62.2**	62	0.55	0.57	0.5	0.56	**0.58**	0.2	0.22	0.11	**0.25**	0.21
Poker-hand	98.7	57.2	97.9	98.8	**98.9**	0.7	0.13	0.45	0.65	**0.73**	0.97	0.21	0.96	0.97	**0.98**

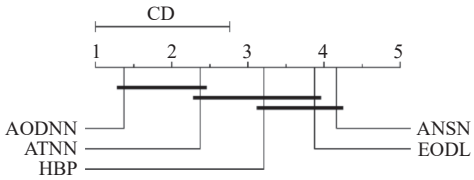

图 4 累计准确率 CD 图

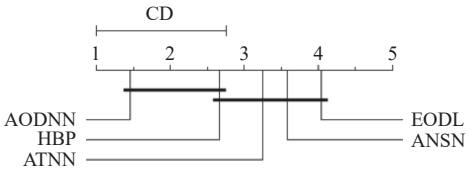

图 5 *F*1 分数 CD 图

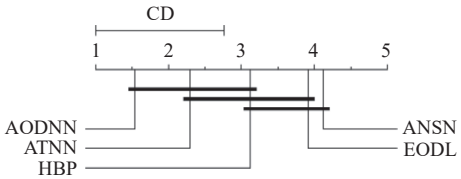

图 6 *MCC* 值 CD 图

图 7 展示了所有方法在不同数据集上的实时准确率, 横坐标为样本量, 纵坐标为实时准确率, 实时准确率可以用于跟踪方法的分类过程. 在 Agrawal、LED_abrupt 和 LED_gradual 数据集上, 当概念漂移发生时, 所有方法的准确率均出现显著下滑, AODNN 能够较其他方法更快地恢复至较高水平; 在真实数据集上, AODNN 准确率始终保持在较高水平, 相较其他方法表现更为平稳. 而在 RBF 数据集上, AODNN 的累计准确率比 ATNN 低 2.8%, 可以从图 7(g) 上看出原因, ATNN 在流数据前期 (前 20k 样本左右) 准确率上升迅速, 并始终高于 AODNN; 但随着样本量继续增大, 其准确率逐渐回落, 逐渐与 AODNN 持平. 这主要源于 ATNN 仅通过判断最后一层分类器权重是否是最大值来决定是否进行深度增长, 而 AODNN 的深度自适应增长策略需要综合网络损失变化趋势、分类器权重和互信息来决定是否添加层数, 这个过程更依赖于一定量的样本累积与参数稳定, 所以在最早期, AODNN 尚未完全完成深度增长, 其曲线比 ATNN 略低. ATNN 的深度增长策略使其在数据流前期表现较好, 但这也导致其深度增长缺乏合理性判断, 在数据集 Agrawal、Sea 和 Hyperplane 等数据集的后期阶段表现不佳, 而 AODNN 则凭借多维度评估进行深度增长决策和参数更新优化策略使得准确率始终保持在较高水平, 展示了对非平稳数据流的持续适应能力.

3.3.2 运行时间对比

为了评估算法的运行效率, 我们对比了所有算法的运行时间, 结果如表 3 所示. 除了 LED_abrupt 和 LED_gradual, AODNN 在其他数据集上的运行时间均低于对比算法. ATNN 的多分支自增长机制虽能提升漂移适应能力, 但其

频繁生成独立分支导致计算成本激增, 在 Covtype、Airlines、Poker-hand 等大规模数据集上表现尤为显著. HBP 和 EODL 则是由于固定的网络结构导致模型适应能力受限, 需要依靠大量参数迭代更新适应概念漂移. 值得注意的是, AODNN 在保持高效计算的同时也未牺牲分类精度, 实现了精度与效率的平衡, 这主要源于深度自适应增长与参数更新优化的协同作用, 通过深度自适应增长逐步扩展网络深度以维持特征抽象能力, 同时利用参数更新优化策略减少计算量并加快模型收敛速度.

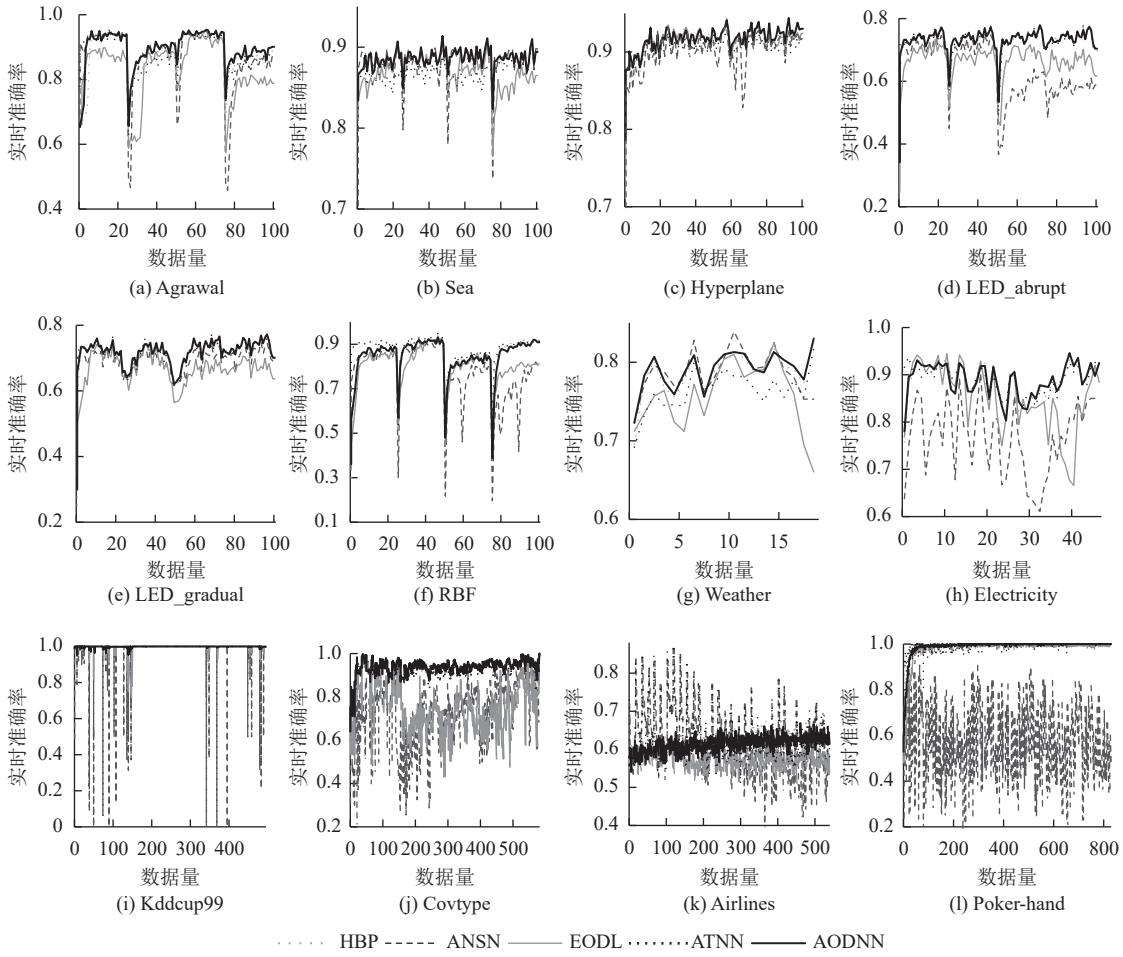

图 7 实时准确率

3.3.3 参数敏感性实验

在这个实验中对衰减因子 β、学习率 η、隐藏层宽度、阈值 ξ、敏感度系数 c^l、c^α、滑动窗口大小 W、分位数 p 以及平滑因子 γ 在不同数据集上的累计准确率进行了分析. 在进行某一超参数的敏感性分析时, 其他所有超参数均固定为第 3.1 节中的默认值. 此举旨在隔离单个参数的影响, 确保观察到的性能变化可归因于该参数本身. 如表 4 和表 5 所示, 分别比较了 β 和 η 不同取值时的累计准确率, 实验结果表明当学习率设定过高 (如 0.05、0.1) 易导致训练的模型准确率下降; 当学习率设定过低 (如 0.001) 时, 也会导致模型收敛到局部极小值使其性能受限. 当 η=0.02 时模型在除了 Kddcup99、Covtype 以外的数据集上都取得了较好的分类性能, 但在 Kddcup99、Covtype 等特征复杂的数据集上, 较大学习率容易引起性能波动与下降. 因此, 在大多数情况下将 η 设置为 0.01 更为合适. 与 η 相比,β 仅决定分类器权重 α 的更新, 对模型性能影响较小, 在大多数情况下将 β 设置为 0.99 可取得最佳性能.

表 3 运行时间对比 (s)

Dataset	HBP	ANSN	EODL	ATNN	AODNN
Agrawal	373	305	278	114	**106**
Sea	377	329	272	**107**	**107**
Hyperplane	380	129	274	106	**102**
LED_abrupt	376	338	276	**87**	100
LED_gradual	370	501	274	**85**	100
RBF	344	191	282	115	**103**
Weather	63	35	49	21	**20**
Electricity	155	907	126	78	**47**
Kddcup99	1 662	1 118	1 444	760	**537**
Covtype	1942	76 340	1 637	825	**589**
Airlines	2 139	2 087	1 919	777	**560**
Poker-hand	3 362	4 236	2 828	1 679	**837**

表 4 不同学习率 η 下 AODNN 的累计准确率 (%)

Dataset	0.001	0.005	0.01	0.02	0.05	0.1
Agrawal	78.3	87.6	89.4	**90.2**	89.6	80.6
Sea	87.3	**88.5**	**88.5**	**88.5**	88.4	86.7
Hyperplane	87.3	91.0	91.6	**91.8**	91.3	73.0
LED_abrupt	67.6	72.5	**72.9**	**72.9**	71.9	68.8
LED_gradual	67.7	71.4	**71.6**	71.5	70.4	67.1
RBF	62.4	82.5	85.2	**85.9**	83.7	46.5
Weather	74.8	78.6	78.8	**79.2**	78.8	73.6
Electricity	81.1	87.2	89.2	90.5	**91.2**	91.0
Kddcup99	99.6	99.8	**99.9**	77.9	21.1	20.0
Covtype	86.5	92.9	**93.7**	89.8	53.1	46.7
Airlines	59.5	61.2	**61.6**	**61.6**	61.4	58.9
Poker-hand	92.7	98.1	98.9	**99.4**	**99.4**	58.2

表 5 不同衰减因子 β 下 AODNN 的累计准确率 (%)

Dataset	0.5	0.7	0.9	0.99	0.999	0.9999
Agrawal	87.7	88.5	88.8	**89.4**	89.2	88.5
Sea	**88.6**	88.4	88.5	88.5	**88.6**	88.5
Hyperplane	90.8	91.2	**91.6**	**91.6**	91.5	91.5
LED_abrupt	72.1	72.4	72.4	**72.9**	**72.9**	72.8
LED_gradual	71.3	71.4	**71.6**	**71.6**	**71.6**	71.5
RBF	84.1	82.4	84.9	**85.2**	85.1	84.7
Weather	78.5	**79.0**	78.8	78.8	78.9	78.9
Electricity	**89.3**	89.0	**89.3**	89.2	89.1	88.4
Kddcup99	**99.9**	99.8	**99.9**	**99.9**	**99.9**	**99.9**
Covtype	91.9	92.2	**93.7**	**93.7**	93.6	93.6
Airlines	61.5	**61.8**	61.7	61.6	61.7	61.3
Poker-hand	98.9	**99.0**	**99.0**	98.9	98.9	98.9

当隐藏层宽度分别为 10、50、100、200 时, AODNN 的累计准确率如图 8 所示. 实验结果表明, 当隐藏层宽度减少时, 模型拟合能力会下降, 在除了 RBF 以外的数据集上, AODNN 性能下降幅度较小, 表明 AODNN 的深度自适应增长与参数更新优化策略可以在一定程度上减轻模型宽度减少带来的模型性能下降问题.

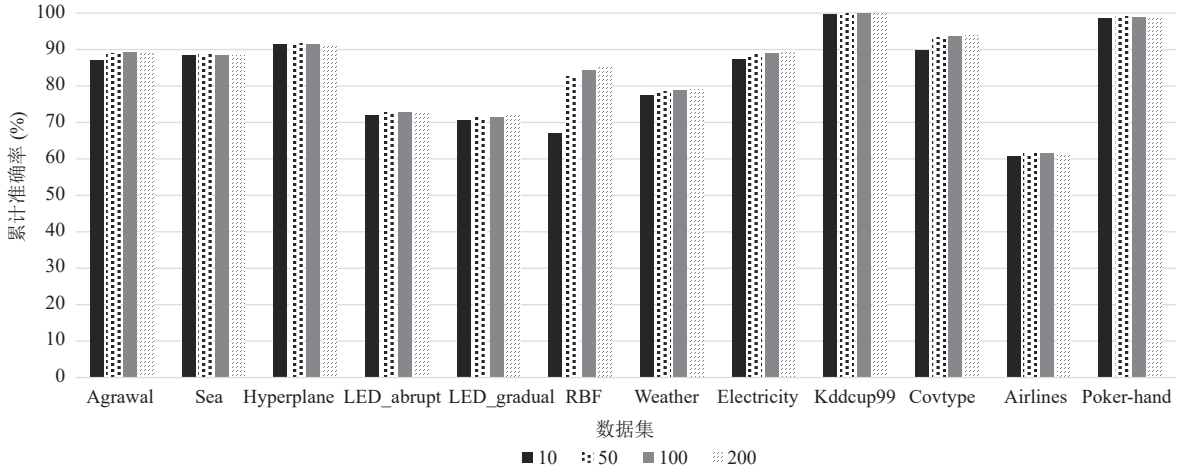

图 8 不同隐藏层宽度下 AODNN 的累计准确率

分位数 p 和平滑因子 γ 是平台期判断的超参数, p 和 γ 共同决定了平台期判断的灵敏度, 如表 6 所示, 评估了 p 为 0.25、0.75、0.9 和 γ 为 0.1、0.5、0.9 不同组合下模型的累计准确率, 实验结果表明在多数情况下, 参数组合 p=0.75, γ=0.9 能够取得最佳性能. 表 7、表 8 分别测试了敏感度系数 c^{α}、c^{I} 为 0.1、0.5、1、1.5、2、3 时的累计准确率, c^{α} 用于计算分类器权重分析的阈值, 决定了何时进行互信息分析, 因此对最终的累计准确率影响较小, 取 c^{α}=1.5 可以适用于大多数情况. 而 c^{I} 用于计算互信息分析的阈值, 作为深度自适应增长的最后一步直接决定了是否要进行层数增长, 实验表明在大多数情况下 c^{I}=1 时可取得最佳性能. 表 9 测试了滑动窗口大小 W 分别为 10、50、100、200 时的累计准确率, 滑动窗口过大会导致算法对损失和互信息的变化不敏感, 窗口过小则会使得算法易受噪声影响, 取 W=100 可以适用于大多数情况.

表 6　不同 (p, γ) 下 AODNN 的累计准确率 (%)

Dataset	(0.25, 0.1)	(0.25, 0.5)	(0.25, 0.9)	(0.75, 0.1)	(0.75, 0.5)	(0.75, 0.9)	(0.9, 0.1)	(0.9, 0.5)	(0.9, 0.9)
Agrawal	89.2	89.3	**89.7**	89.2	89.2	89.4	89.5	89.5	89.5
Sea	88.4	88.5	88.5	88.5	88.5	**88.6**	**88.6**	88.4	88.4
Hyperplane	91.4	91.4	91.5	91.4	91.4	**91.6**	91.5	91.5	**91.6**
LED_abrupt	72.7	72.7	**72.9**	72.8	72.8	**72.9**	72.8	72.8	72.8
LED_gradual	71.4	71.4	71.5	71.4	71.4	**71.6**	71.5	71.5	71.4
RBF	84.2	84.3	**85.2**	84.3	84.3	**85.2**	**85.2**	85.1	85.1
Weather	78.7	78.7	**79.1**	78.7	78.7	78.8	79.0	**79.1**	79.0
Electricity	89.0	89.1	89.2	89.0	89.0	89.2	89.2	**89.4**	89.2
Kddcup99	99.7	**99.9**	99.5	99.8	99.7	**99.9**	**99.9**	**99.9**	**99.9**
Covtype	93.6	93.4	93.6	93.6	**93.7**	**93.7**	**93.7**	**93.7**	**93.7**
Airlines	61.7	**61.8**	61.7	61.7	**61.8**	61.7	61.6	61.6	61.4
Poker-hand	**98.9**	**98.9**	98.8	98.8	98.8	98.9	**99.0**	98.9	98.9

表 7　不同敏感度系数 c^{α} 下 AODNN 的累计准确率 (%)

Dataset	0.1	0.5	1	1.5	2	3
Agrawal	**89.6**	89.5	89.5	**89.6**	89.5	89.4
Sea	88.4	88.4	**88.6**	**88.6**	88.5	**88.6**
Hyperplane	91.6	91.7	91.5	**91.6**	91.5	**91.6**
LED_abrupt	73.0	73.0	**73.1**	73.0	72.9	72.9
LED_gradual	**71.7**	**71.7**	71.6	**71.7**	71.6	71.6
RBF	85.2	85.1	**85.3**	85.2	85.1	85.2
Weather	79.1	78.8	**79.2**	79.0	79.1	79.1
Electricity	89.2	**89.3**	**89.3**	**89.3**	89.2	89.1
Kddcup99	99.7	99.7	99.8	**99.9**	99.8	99.8
Covtype	93.5	**93.7**	**93.7**	**93.7**	93.6	93.5
Airlines	61.7	61.7	**61.8**	61.7	61.6	61.6
Poker-hand	98.9	**99.0**	98.9	98.9	**99.0**	**99.0**

表 8　不同敏感度系数 c^{I} 下 AODNN 的累计准确率 (%)

Dataset	0.1	0.5	1	1.5	2	3
Agrawal	89.4	89.3	**89.6**	89.4	89.4	89.5
Sea	**88.6**	**88.6**	**88.6**	88.5	88.5	88.4
Hyperplane	91.5	91.5	**91.6**	**91.6**	91.5	91.5
LED_abrupt	72.9	72.9	**73.0**	**73.0**	72.9	72.9
LED_gradual	71.6	71.6	**71.7**	**71.7**	71.6	71.6
RBF	85.1	**85.2**	**85.2**	85.1	85.1	85.1
Weather	**79.2**	78.8	79.0	79.0	78.9	78.9
Electricity	89.3	89.3	89.3	89.2	89.2	**89.5**
Kddcup99	**99.9**	**99.9**	**99.9**	99.8	98.7	**99.9**
Covtype	**93.8**	93.7	93.7	93.6	93.7	93.7
Airlines	61.7	61.6	**61.8**	61.6	61.6	61.6
Poker-hand	**99.0**	98.9	98.9	98.9	98.9	**99.0**

阈值 ξ 用于决定哪些分类器可以参与反向传播, 在表 10 中, 对比了 AODNN 在 ξ 分别取 0.1、0.3、0.5、0.7 时的累计准确率, 当阈值过低 (0.1) 时, 几乎所有中间分类器都参与反向传播, 反而干扰网络的收敛; 而当阈值过高时 (0.5、0.7), 只有两层中间分类器会参与反向传播, 会导致分类器权重 α 过于集中, 影响模型对概念漂移的适应. 因此, 在绝大多数情况下将 ξ 设置为 0.3 可以取得最佳性能.

3.3.4　消融实验

(1) 深度自适应增长有效性

为了验证深度自适应增长的有效性, 在删除 AODNN 深度自适应增长机制的情况下, 分别测试模型深度为 3–10 层时在不同数据集上的累计准确率来找到模型的最佳深度, 然后将最佳深度与在同一数据集中训练 AODNN 模型

的累计准确率和最终深度进行对比, 具体结果如表 11 所示, 其中最佳层数比=AODNN 层数/最佳层数. 根据实验结果可以看出, 模型深度过深或者过浅都会导致模型准确率下降, AODNN 的深度自适应增长可以减轻这个问题, 其累计准确率在大多数数据集上都接近最佳层数的准确率, 根据最佳层数比可以看出 AODNN 在大多数数据集上的层数都接近最佳层数, 表明 AODNN 的深度自适应增长机制有效.

表 9 不同滑动窗口大小 W 下 AODNN 的累计准确率 (%)

Dataset	10	50	100	200
Agrawal	89.5	89.4	**89.6**	89.5
Sea	**88.6**	**88.6**	**88.6**	**88.6**
Hyperplane	91.6	91.6	**91.7**	91.6
LED_abrupt	72.9	72.9	**73.0**	72.9
LED_gradual	71.7	71.6	**71.7**	**71.7**
RBF	85.2	85.2	**85.3**	85.2
Weather	79.0	79.0	**79.1**	79.0
Electricity	**89.4**	**89.4**	89.3	89.2
Kddcup99	**99.9**	**99.9**	**99.9**	99.8
Covtype	92.8	93.3	**93.7**	93.6
Airlines	61.5	61.5	**61.6**	61.5
Poker-hand	**99.0**	98.9	**99.0**	98.9

表 10 不同阈值 ξ 下 AODNN 的累计准确率 (%)

Dataset	0.1	0.3	0.5	0.7
Agrawal	89.2	**89.4**	89.2	89.0
Sea	88.5	**88.6**	88.5	88.5
Hyperplane	**91.6**	**91.6**	91.5	91.5
LED_abrupt	**72.9**	**72.9**	72.3	72.1
LED_gradual	**71.6**	**71.6**	70.2	70.6
RBF	84.3	84.3	84.8	**85.0**
Weather	78.7	**78.8**	**78.8**	78.7
Electricity	89.1	89.1	**89.4**	89.3
Kddcup99	**99.9**	**99.9**	99.8	**99.9**
Covtype	93.6	93.7	**93.9**	93.8
Airlines	**61.7**	61.6	**61.7**	**61.7**
Poker-hand	99.0	99.0	**99.2**	99.1

表 11 AODNN 模型和 AODNN 删除深度自适应后不同预设深度的层数、最佳层数比及累计准确率

Dataset	AODNN删除深度自适应后在不同预设深度的累计准确率 (%)								AODNN模型		
	3	4	5	6	7	8	9	10	累计准确率 (%)	层数	最佳层数比
Agrawal	89.2	88.9	88.2	87.8	86.8	84.7	85.5	87.0	**89.3**	3.1	1.03
Sea	**88.6**	**88.6**	88.5	88.4	88.5	88.1	87.7	87.7	**88.6**	4	1
Hyperplane	**91.6**	91.4	91.1	91.3	91.2	90.8	90.6	90.7	**91.6**	3	1
LED_abrupt	**72.9**	72.0	70.8	69.5	69.9	68.1	65.5	59.1	**72.9**	3	1
LED_gradual	**71.6**	70.3	69.1	70.1	70.0	69.4	64.4	70.3	**71.6**	3	1
RBF	84.3	**84.4**	83.5	83.6	81.6	80.1	67.7	74.2	84.3	3.9	0.975
Weather	**78.8**	78.3	78.3	77.7	77.3	77.4	78.2	76.0	**78.8**	3	1
Electricity	89.0	89.2	**89.3**	89.0	88.5	88.7	88.9	88.7	89.2	3	0.6
Kddcup99	**99.9**	99.5	99.8	99.8	99.8	93.4	97.0	99.7	**99.9**	3.6	1.2
Covtype	93.6	**93.8**	89.9	88.5	93.3	93.4	92.8	93.3	**93.7**	3.6	0.9
Airlines	61.7	**61.8**	61.6	61.7	61.7	61.4	61.0	61.3	61.7	4	1
Poker-hand	98.9	**99.1**	**99.1**	**99.1**	98.8	98.7	98.7	98.6	98.9	3	0.6

(2) 参数更新优化有效性

为了验证参数更新优化策略的有效性, 对比了删除 AODNN 参数更新优化策略的算法 AODNN_DEL 和 AODNN 在累计准确率和运行时间上的效果, 具体结果如表 12 所示. 实验结果表明, 在大多数数据集上累计准确率都有提升, 运行时间的消耗平均减少 4%, 特别是在 Covtype 这种大规模的数据集上表现明显, 由于 Covtype 数据复杂度较高, 模型的中间分类器之间容易产生干扰, 导致模型性能下降, 参数更新优化在一定程度上缓解了这个问题.

4 总 结

针对非平稳数据流中深度神经网络预设深度困难、现有方法缺少深度增长有效性判断机制以及参数和结构协同调整不足的问题, 本文提出自适应在线深度神经网络调整方法 AODNN. AODNN 通过分位数对网络的损失

变化趋势进行分析, 判断网络性能是否达到瓶颈, 通过动态阈值对分类器权重和互信息变化综合分析判断增加深度的合理性和有效性, 当出现性能瓶颈且深度增长被判定有效时, AODNN 会增加深度来提升模型容量; 而当概念漂移发生时, AODNN 通过参数更新优化选择合适的中间分类器参与反向传播, 有效减轻深层对浅层的负向干扰, 加速模型收敛, 从而实现参数与结构的协同调整. 实验部分在真实与合成数据集上验证了 AODNN 深度自适应增长和参数更新优化策略的有效性; 与前沿在线深度学习方法相比, AODNN 展现出更高的分类精度和更快的概念漂移适应速度. 值得注意的是, 当前 AODNN 缺少记忆机制, 在面对重复出现的概念时需要重新学习, 并易受灾难性遗忘的影响, 这将是未来研究的重点.

表 12　AODNN 和 AODNN_DEL 在累计准确率和运行时间上的对比

Dataset	累计准确率 (%)		运行时间 (s)	
	AODNN	AODNN_DEL	AODNN	AODNN_DEL
Agrawal	**89.3**	89.2	104	**102**
Sea	**88.6**	88.5	**105**	110
Hyperplane	**91.6**	**91.6**	**102**	111
LED_abrupt	**72.9**	**72.9**	**100**	103
LED_gradual	**71.6**	**71.6**	**100**	106
RBF	84.3	**84.9**	**103**	104
Weather	**78.8**	78.7	**20**	21
Electricity	**89.1**	**89.1**	**47**	48
Kddcup99	**99.9**	**99.9**	**532**	537
Covtype	**93.7**	90.7	**589**	647
Airlines	**61.7**	**61.7**	**560**	617
Poker-hand	**98.9**	**98.9**	**837**	877

References

[1] Chen ZQ, Han M, Li MH, Wu HX, Zhang XL. Survey of concept drift handling methods in data streams. Computer Science, 2022, 49(9): 14–32 (in Chinese with English abstract). [doi: 10.11896/jsjkx.210700112]

[2] Hinder F, Vaquet V, Hammer B. One or two things we know about concept drift—A survey on monitoring in evolving environments. Part A: Detecting concept drift. Frontiers in Artificial Intelligence, 2024, 7: 1330257. [doi: 10.3389/frai.2024.1330257]

[3] Wen YM, Liu S, Miao YQ, Yi XH, Liu CJ. Survey on semi-supervised classification of data streams with concept drifts. Ruan Jian Xue Bao/Journal of Software, 2022, 33(4): 1287–1314 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6476.htm [doi: 10.13328/j.cnki.jos.006476]

[4] Yu E, Lu J, Zhang GQ. Generalized incremental learning under concept drift across evolving data streams. arXiv:2506.05736, 2025.

[5] Korycki L, Krawczyk B. Class-incremental experience replay for continual learning under concept drift. In: Proc. of the 2021 IEEE/CVF Conf. on Computer Vision and Pattern Recognition Workshops. Nashville: IEEE, 2021. 3644–3653. [doi: 10.1109/CVPRW53098.2021.00404]

[6] Guo YW, Yao AB, Chen YR. Dynamic network surgery for efficient DNNs. In: Proc. of the 30th Int'l Conf. on Neural Information Processing Systems. Barcelona: Curran Associates Inc., 2016. 1387–1395.

[7] Gama J, Medas P, Castillo G, Rodrigues P. Learning with drift detection. In: Proc. of the 17th Brazilian Symp. on Artificial Intelligence Advances in Artificial Intelligence. Sao Luis: Springer, 2004. 286–295. [doi: 10.1007/978-3-540-28645-5_29]

[8] Baena-García M, del Campo-Ávila J, Fidalgo R, Bifet A, Gavaldà R, Morales-Bueno R. Early drift detection method. In: Proc. of the 4th Int'l Workshop on Knowledge Discovery from Data Streams, 2006. 77–86.

[9] Frias-Blanco I, del Campo-Ávila J, Ramos-Jimenez G, Morales-Bueno R, Ortiz-Diaz A, Caballero-Mota Y. Online and non-parametric drift detection methods based on Hoeffding's bounds. IEEE Trans. on Knowledge and Data Engineering, 2015, 27(3): 810–823. [doi: 10.1109/tkde.2014.2345382]

[10] Pesaranghader A, Viktor HL. Fast Hoeffding drift detection method for evolving data streams. In: Proc. of the 2016 Joint European Conf. on Machine Learning and Knowledge Discovery in Databases. Riva del Garda: Springer, 2016. 96–111. [doi: 10.1007/978-3-319-46227-1_7]

[11] Pesaranghader A, Viktor H, Paquet E. Reservoir of diverse adaptive learners and stacking fast Hoeffding drift detection methods for evolving data streams. Machine Learning, 2018, 107(11): 1711–1743. [doi: 10.1007/s10994-018-5719-z]

[12] Zhao RR, You YQ, Sun JB, Gama J, Jiang J. Online learning from drifting capricious data streams with flexible Hoeffding tree. Information Processing & Management, 2025, 62(6): 104221. [doi: 10.1016/j.ipm.2025.104221]

[13] Yu E, Lu J, Zhang B, Zhang GQ. Online boosting adaptive learning under concept drift for multistream classification. In: Proc. of the 38th AAAI Conf. on Artificial Intelligence. Vancouver: AAAI, 2024. 16522–16530. [doi: 10.1609/aaai.v38i15.29590]

[14] Domingos P, Hulten G. Mining high-speed data streams. In: Proc. of the 6th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. Boston: ACM, 2000. 71–80. [doi: 10.1145/347090.347107]

[15] Hulten G, Spencer L, Domingos P. Mining time-changing data streams. In: Proc. of the 7th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. San Francisco: ACM, 2001. 97–106. [doi: 10.1145/502512.502529]

[16] Blake C, Ntoutsi E. Reinforcement learning based decision tree induction over data streams with concept drifts. In: Proc. of the 2018 IEEE Int'l Conf. on Big Knowledge. Singapore: IEEE, 2018. 328–335. [doi: 10.1109/ICBK.2018.00051]

[17] Krawczyk B, Woźniak M. Weighted naïve Bayes classifier with forgetting for drifting data streams. In: Proc. of the 2015 IEEE Int'l Conf. on Systems, Man, and Cybernetics. Hong Kong: IEEE, 2015. 2147–2152. [doi: 10.1109/SMC.2015.375]

[18] Kishore Babu D, Ramadevi Y, Ramana KV. RGNBC: Rough gaussian Naïve Bayes classifier for data stream classification with recurring concept drift. Arabian Journal for Science and Engineering, 2017, 42(2): 705–714. [doi: 10.1007/s13369-016-2317-x]

[19] Babu DK, Ramadevi Y, Ramana KV. PGNBC: Pearson Gaussian Naïve Bayes classifier for data stream classification with recurring concept drift. Intelligent Data Analysis, 2017, 21(5): 1173–1191. [doi: 10.3233/IDA-163020]

[20] Liang NY, Huang GB, Saratchandran P, Sundararajan N. A fast and accurate online sequential learning algorithm for feedforward networks. IEEE Trans. on Neural Networks, 2006, 17(6): 1411–1423. [doi: 10.1109/TNN.2006.880583]

[21] Xu SL, Wang JH. A fast incremental extreme learning machine algorithm for data streams classification. Expert Systems with Applications, 2016, 65: 332–344. [doi: 10.1016/j.eswa.2016.08.052]

[22] Lan Y, Soh YC, Huang GB. Ensemble of online sequential extreme learning machine. Neurocomputing, 2009, 72(13–15): 3391–3395. [doi: 10.1016/j.neucom.2009.02.013]

[23] Street WN, Kim YS. A streaming ensemble algorithm (SEA) for large-scale classification. In: Proc. of the 7th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. San Francisco: ACM, 2001. 377–382. [doi: 10.1145/502512.502568]

[24] Qi XB, Chen JM, Shi Y, Qi H, Guo HS, Wang WJ. Concept drift adaptive method based on active-passive incremental ensemble. Acta Automatica Sinica, 2025, 51(5): 1131–1144 (in Chinese with English abstract). [doi: 10.16383/j.aas.c240503]

[25] Barboza EVL, de Almeida PRL, de Souza Britto A Jr, Sabourin R, Cruz RMO. IncA-DES: An incremental and adaptive dynamic ensemble selection approach using online K-d tree neighborhood search for data streams with concept drift. Information Fusion, 2025, 123: 103272. [doi: 10.1016/j.inffus.2025.103272]

[26] Yuan LH, Li H, Xia BH, Gao GY, Liu MY, Yuan W, You X. Recent advances in concept drift adaptation methods for deep learning. In: Proc. of the 31st Int'l Joint Conf. on Artificial Intelligence. Vienna, 2022. 5654–5661. [doi: 10.24963/ijcai.2022/788]

[27] Guo JF, Chen CLP, Liu ZL, Yang XX. Dynamic neural network structure: A review for its theories and applications. IEEE Trans. on Neural Networks and Learning Systems, 2025, 36(3): 4246–4266. [doi: 10.1109/TNNLS.2024.3377194]

[28] Sahoo D, Pham Q, Lu J, Hoi SCH. Online deep learning: Learning deep neural networks on the fly. In: Proc. of the 27th Int'l Joint Conf. on Artificial Intelligence. Stockholm: AAAI Press, 2018. 2660–2666.

[29] Su R, Guo HS, Wang WJ. Elastic online deep learning for dynamic streaming data. Information Sciences, 2024, 676: 120799. [doi: 10.1016/j.ins.2024.120799]

[30] Guo HS, Zhang S, Wang WJ. Selective ensemble-based online adaptive deep neural networks for streaming data with concept drift. Neural Networks, 2021, 142: 437–456. [doi: 10.1016/j.neunet.2021.06.027]

[31] Han YN, Liu JW, Xiao BB, Wang XT, Luo XL. Bilevel online deep learning in non-stationary environment. In: Proc. of the 30th Int'l Conf. on Artificial Neural Networks. Bratislava: Springer, 2021. 347–358. [doi: 10.1007/978-3-030-86340-1_28]

[32] Kauschke S, Lehmann DH, Fürnkranz J. Patching deep neural networks for nonstationary environments. In: Proc. of the 2019 Int'l Joint Conf. on Neural Networks. Budapest: IEEE, 2019. 1–8. [doi: 10.1109/IJCNN.2019.8852222]

[33] Ashfahani A, Pratama M, Lughofer E, Ong YS. DEVDAN: Deep evolving denoising autoencoder. Neurocomputing, 2020, 390: 297–314. [doi: 10.1016/j.neucom.2019.07.106]

[34] Eldan R, Shamir O. The power of depth for feedforward neural networks. In: Proc. of the 29th Conf. on Learning Theory. New York: JMLR. org, 2016. 907–940.

[35] Ashfahani A, Pratama M. Autonomous deep learning: Continual learning approach for dynamic environments. In: Proc. of the 2019

SIAM Int'l Conf. on Data Mining. Calgary: Society for Industrial and Applied Mathematics, 2019. 666–674. [doi: 10.1137/1.9781611975673.75]

[36] Zhang X, Zou YY, Li SY. Enhancing incremental deep learning for FCCU end-point quality prediction. Information Sciences, 2020, 530: 95–107. [doi: 10.1016/j.ins.2020.04.013]

[37] Pratama M, Za'in C, Ashfahani A, Ong YS, Ding WP Automatic construction of multi-layer perceptron network from streaming examples. In: Proc. of the 28th ACM Int'l Conf. on Information and Knowledge Management. Beijing: ACM, 2019. 1171–1180. [doi: 10.1145/3357384.3357946]

[38] Zhou XZ, Liu X, Wen YM. Depth and width adaption of DNN for data stream classification with concept drifts. In: Proc. of the 24th Int'l Conf. on Intelligent Data Engineering and Automated Learning. Évora: Springer, 2023. 406–417. [doi: 10.1007/978-3-031-48232-8_37]

[39] Zhong Y, Zhou J, Li P, Gong J. Dynamically evolving deep neural networks with continuous online learning. Information Sciences, 2023, 646: 119411. [doi: 10.1016/j.ins.2023.119411]

[40] Yang Y, Zhou DW, Zhan DC, Xiong H, Jiang Y. Adaptive deep models for incremental learning: Considering capacity scalability and sustainability. In: Proc. of the 25th ACM SIGKDD Int'l Conf. on Knowledge Discovery & Data Mining. Anchorage: ACM, 2019. 74–82. [doi: 10.1145/3292500.3330865]

[41] Wen YM, Liu X, Yu H. Adaptive tree-like neural network: Overcoming catastrophic forgetting to classify streaming data with concept drifts. Knowledge-based Systems, 2024, 293: 111636. [doi: 10.1016/j.knosys.2024.111636]

[42] Thomas AJ, Petridis M, Walters SD, Gheytassi SM, Morgan RE. Two hidden layers are usually better than one. In: Proc. of the 18th Int'l Conf. on Engineering Applications of Neural Networks. Athens: Springer, 2017. 279–290. [doi: 10.1007/978-3-319-65172-9_24]

[43] Koenker R, Hallock KF. Quantile regression. Journal of Economic Perspectives, 2001, 15(4): 143–156. [doi: 10.1257/jep.15.4.143]

[44] Hu SZ, Lou ZZ, Yan XQ, Ye YD. A survey on information bottleneck. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2024, 46(8): 5325–5344. [doi: 10.1109/TPAMI.2024.3366349]

[45] Goldfeld Z, van den Berg E, Greenewald K, Melnyk I, Nguyen N, Kingsbury B, Polyanskiy Y. Estimating information flow in deep neural networks. In: Proc. of the 36th Int'l Conf. on Machine Learning. Long Beach: PMLR, 2019. 2299–2308.

[46] Tishby N, Zaslavsky N. Deep learning and the information bottleneck principle. In: Proc. of the 2015 IEEE Information Theory Workshop. Jerusalem: IEEE, 2015. 1–5. [doi: 10.1109/ITW.2015.7133169]

[47] Nalewajski RF. Elements of information theory. In: Perspectives in Electronic Structure Theory. Berlin, Heidelberg: Springer, 2012. 371–395. [doi: 10.1007/978-3-642-20180-6_8]

[48] Bifet A, Holmes G, Pfahringer B, Kranen P, Kremer H, Jansen T, Seidl T. MOA: Massive online analysis, a framework for stream classification and clustering. In: Proc. of the 1st Workshop on Applications of Pattern Analysis. Cumberland Lodge: PMLR, 2010. 44–50.

[49] Wang YZ, Qian JX, Hassan M, Zhang XY, Zhang T, Yang C, Zhou XX, Jia FJ. Density peak clustering algorithms: A review on the decade 2014–2023. Expert Systems with Applications, 2024, 238: 121860. [doi: 10.1016/j.eswa.2023.121860]

[50] Bakhshi S, Ghahramanian P, Bonab H, Can F. A broad ensemble learning system for drifting stream classification. IEEE Access, 2023, 11: 89315–89330. [doi: 10.1109/ACCESS.2023.3306957]

[51] Halstead B, Koh YS, Riddle P, Pears R, Pechenizkiy M, Bifet A, Olivares G, Coulson G. Analyzing and repairing concept drift adaptation in data stream classification. Machine Learning, 2022, 111(10): 3489–3523. [doi: 10.1007/s10994-021-05993-w]

[52] Gunasekara N, Gomes HM, Pfahringer B, Bifet A. Online hyperparameter optimization for streaming neural networks. In: Proc. of the 2022 Int'l Joint Conf. on Neural Networks. Padua: IEEE, 2022. 1–9. [doi: 10.1109/IJCNN55064.2022.9891953]

[53] Belghazi MI, Baratin A, Rajeshwar S, Ozair S, Bengio Y, Courville A, Hjelm RD. Mutual information neural estimation. In: Proc. of the 35th Int'l Conf. on Machine Learning. Stockholm: PMLR, 2018. 531–540.

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

附中文参考文献

[1] 陈志强, 韩萌, 李慕航, 武红鑫, 张喜龙. 数据流概念漂移处理方法研究综述. 计算机科学, 2022, 49(9): 14–32. [doi: 10.11896/jsjkx.210700112]

[3] 文益民, 刘帅, 缪裕青, 易新河, 刘长杰. 概念漂移数据流半监督分类综述. 软件学报, 2022, 33(4): 1287–1314. http://www.jos.org.cn/1000-9825/6476.htm [doi: 10.13328/j.cnki.jos.006476]

[24] 祁晓博, 陈佳明, 史颖, 亓慧, 郭虎升, 王文剑. 基于主动-被动增量集成的概念漂移适应方法. 自动化学报, 2025, 51(5): 1131–1144. [doi: 10.16383/j.aas.c240503]

作者简介

邹迪, 硕士生, 主要研究领域为机器学习.

周学文, 高级工程师, 主要研究领域为数据挖掘.

范玉雷, 博士, 讲师, CCF 专业会员, 主要研究领域为数据科学, 机器学习.

高楠, 博士, 副教授, CCF 专业会员, 主要研究领域为机器学习, 自然语言处理, 医学图像处理.

喻坚, 博士, 副教授, 主要研究领域为计算理论, 人工智能, 服务计算.

杨良怀, 博士, 教授, CCF 专业会员, 主要研究领域为数据科学, 人工智能.