面向异构流式数据的高性能联邦持续学习算法

姜慧[1,2]，何天流[1,2]，刘敏[1,2,3]，孙胜[1]，王煜炜[1,2]
（1. 中国科学院计算技术研究所，北京 100190；
2. 中国科学院大学计算机科学与技术学院，北京 100190；
3. 中关村实验室，北京 100084）

摘 要：为了缓解提供智能服务的 AI 模型训练流式数据存在模型性能差、训练效率低等问题，在具有隐私数据的分布式终端系统中，提出了一种面向异构流式数据的高性能联邦持续学习算法（FCL-HSD）。为了缓解当前模型遗忘旧数据问题，在本地训练阶段引入结构可动态扩展模型，并设计扩展审核机制，以较小的存储开销来保障 AI 模型识别旧数据的能力；考虑到终端的数据异构性，在中央节点侧设计了基于数据分布相似度的全局模型定制化策略，并为模型的不同模块执行分块聚合方式。在不同数据集下多种数据增量场景中验证了所提算法的可行性和有效性。实验结果证明，相较于现有工作，所提算法在保证模型对新数据具有分类能力的前提下，可以有效提升模型对旧数据的分类能力。

关键词：异构数据；流式数据；联邦学习；联邦持续学习；灾难性遗忘
中图分类号：TP302
文献标志码：A
DOI: 10.11959/j.issn.1000−436x.2023102

High-performance federated continual learning algorithm for heterogeneous streaming data

JIANG Hui[1,2], HE Tianliu[1,2], LIU Min[1,2,3], SUN Sheng[1], WANG Yuwei[1,2]
1. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China
2. School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 100190, China
3. Zhongguancun Laboratory, Beijing 100084, China

Abstract: Aiming at the problems of poor model performance and low training efficiency in training streaming data of AI models that provide intelligent services, a high-performance federated continual learning algorithm for heterogeneous streaming data (FCL-HSD) was proposed in the distributed terminal system with privacy data. In order to solve the problem of the current model forgetting old data, a model with dynamically extensible structure was introduced in the local training stage, and an extension audit mechanism was designed to ensure the capability of the AI model to recognize old data at the cost of small storage overhead. Considering the heterogeneity of terminal data, a customized global model strategy based on data distribution similarity was designed at the central server side, and an aggregation-by-block manner was implemented for different modules of the model. The feasibility and effectiveness of the proposed algorithm were verified under various data increment scenarios with different data sets. Experimental results show that, compared with existing works, the proposed algorithm can effectively improve the model performance to classify old data on the premise of ensuring the capability to classify new data.

Keywords: heterogeneous data, streaming data, federated learning, federated continual learning, catastrophic forgetting

收稿日期：2023−02−06；**修回日期：**2023−04−28
通信作者：刘敏，liumin@ict.ac.cn
基金项目：国家重点研发计划基金资助项目（No.2021YFB2900102）；国家自然科学基金资助项目（No.62072436）
Foundation Items: The National Key Research and Development Program of China (No.2021YFB2900102), The National Natural Science Foundation of China (No.62072436)

0 引言

蓬勃发展的 5G 与人工智能（AI, artificial intelligence）技术赋予了分布式终端多元化的智能能力，推动了自动驾驶、虚拟现实、智慧医疗等各类智能服务落地与大规模应用[1]。对于分布式终端设备在网络边缘侧产生的海量数据，以隐私保护为前提的联邦学习（FL, federated learning）框架，为大规模终端提供了新型的 AI 模型训练方式[2-3]。在传统 FL 系统中，分布式终端根据本地数据进行模型训练，并周期性地与中央节点交互模型参数，最终得到一个对所有终端都适用的全局模型[4]。传统的 FL 假设在模型训练过程中，终端侧的数据已提前存储且保持不变，因此仅适用于静态数据集场景[5]。但是，在实际网络系统中，数据具有流式性质，终端数据量、数据类别会随着时间的推移而动态增加。由于隐私要求以及终端受限的存储能力，终端无法存储所有数据，因此产生的新数据会覆盖旧数据。然而，采用传统 FL 方法对动态变化的数据进行模型训练时，当前模型会偏向于对新数据进行识别分析，而遗忘对旧数据的识别分析能力，产生灾难性遗忘现象[6]。文献[7]指出，用 ResNet 模型[8]进行图像分类时，模型对当前数据的分类准确率可以达到 92%，同时对旧数据的分类准确率从 92%降到 21%，这表明模型遗忘了对旧数据的分类能力。因此传统 FL 方法不适用于具有流式数据的实际网络系统，无法为网络中的智能服务维护演进式、持续式的智能特性。

最近提出的持续学习（CL, continual learning），也称为增量学习，是解决灾难性遗忘问题的一种学习范式[9-10]。然而，现有的解决方案仅适用于集中式场景[11-13]，无法解决终端分散的 FL 场景下的灾难性遗忘问题。最主要的原因在于，在 FL 框架中，终端基于本地数据集训练得到本地模型，而不同终端的数据分布是异构的，因此不同终端的本地模型仅适用于本地数据分布。若在终端本地使用上述集中式持续学习方法，并采用原始的 FL 聚合方法对多个终端模型进行模型聚合，所得到的全局模型混淆了多个终端模型对本地异构数据的识别分析能力，无法在所有终端侧都具有良好的性能表现，甚至会产生性能降级、模型不收敛等问题。

为了能够在联邦场景下应用持续学习以缓解灾难性遗忘问题，部分工作开展了关于联邦持续学习（FCL, federated continual learning）的研究。文献[14]提出了一个运用 FL 框架实现网络流量分类的分布式协议 FLIC（federated learning on Internet classification），实现对网络数据包的应用分类。在 FLIC 中，当终端产生来自新应用的网络数据时，会向中央节点发送模型扩展请求，并将本地数据标签分布上传给中央节点。中央节点接收到标签分布后，根据新增的类别数，增加全局模型最后一层的输出维度。虽然 FLIC 考虑了终端数据动态增量问题，但是没有考虑到本地数据存储有限，新数据持续到来会覆盖旧数据信息，仅通过扩增全局模型输出维度的方式不能解决因数据类别持续增加而引起的模型灾难性遗忘问题。文献[15]指出在 FCL 场景中，本地和全局都存在灾难性遗忘问题，该研究提出的 GLFC（global local forgetting compensation）算法在本地模型中额外引入了 2 个正则项：梯度补偿损失项和蒸馏损失项，用于缓解本地灾难性遗忘问题。此外，为了缓解全局灾难性遗忘问题，中央节点通过收集本地模型的梯度值来构建代理数据集，通过代理数据集选择全局最佳旧模型。因此，中央节点既需要执行模型聚合操作，也需要额外训练一个用于生成数据的模型，存在巨大的计算开销。此外，GLFC 为每个终端下发全局统一的旧模型，是根据系统内所有类别数据的平均准确率进行挑选的，无法确保全局模型对每个终端本地异构的旧数据都有较好的分类能力[16]。综上所述，如何在终端数据异构情况下设计确保对新旧数据精准识别与分析的联邦持续学习算法是实现网络内生智能、演进智能亟待解决的关键问题。

该问题主要存在 2 个挑战。首先，流式数据使终端设备在训练过程中逐渐遗忘对旧数据的分析能力，影响了本地模型性能；其次，统一的全局聚合方法忽略了终端异构的增量数据，使全局模型在终端模型上的性能表现具有差异化。

为了解决上述挑战，本文提出了一种面向异构流式数据的联邦持续学习（FCL-HSD, federated continual learning for heterogeneous streaming data）算法，在保护数据隐私的前提下实现分布式终端的联邦学习，并考虑到终端数据的异构性与动态增量性，优化了联邦学习的本地训练和模型聚合阶段，提升模型在新旧数据下的整体准确率，实现网络内生的、演进的智能。

本文的主要贡献包括 4 个方面。

1) 考虑到网络智能服务中终端数据具有异构

性和流式增量性，针对分类任务，提出了 FCL-HSD 算法，缓解模型训练过程中因数据持续增量导致对旧数据的灾难性遗忘问题，从而保证终端对新旧数据都具有准确的识别与分类能力。

2) 为了缓解联邦学习中的灾难性遗忘问题，FCL-HSD 优化了本地训练方式。在本地训练阶段，为模型设计了可动态扩展的结构用于存储部分旧模型参数，并提出了扩展审核机制，以更少的存储开销为代价保障模型对旧数据的识别能力。

3) 为了解决传统聚合方法忽略终端的数据异构性、存在全局模型不适配问题，FCL-HSD 设计了全局模型定制化策略，针对模型不同模块采用分块聚合方式，并提出了基于余弦相似度的模型贡献度衡量方法，对终端之间的数据分布进行余弦相似度分析，为每个终端量化其他终端的模型贡献度，生成定制化的全局模型，提升终端对旧数据的分类能力和对新数据的识别能力。

4) 在不同的数据增量场景下验证了 FCL-HSD 算法的有效性。实验结果表明，相较于现有的联邦持续学习算法，FCL-HSD 算法在保证对新数据具有分类能力前提下，可以有效提升模型对旧数据的分类能力，使模型整体性能达到最优。

1　相关概念

本节介绍方法设计中涉及的关键技术概念以及原理，包括联邦学习和持续学习。

1.1　联邦学习

银行、医院等本地终端会产生丰富的数据，但是受限于规章制度和相关法律法规，这些富含大量隐私信息的原始数据不能被集中式地收集与分析。谷歌公司于 2018 年提出 FL 框架[4]，其核心思想是“数据不动模型动”，即在终端本地部署 AI 模型，利用本地数据训练模型，与中央节点周期性交互并更新模型参数。因此，FL 以保障数据隐私为前提，可以为分布式终端部署更普适的 AI 模型，提供更高质量、更安全的智能服务[17-18]。

FL 系统通常由一个中央节点和多个终端组成。传统 FL 的模型训练流程主要分为 2 个阶段：模型本地训练阶段和全局聚合阶段。

1) 模型本地训练阶段。在第 r 轮模型训练开始时，终端 k 接收来自中央节点的上一轮全局模型 w^{r-1}，并利用本地存储的数据集 $\mathcal{D}_k=\{x_{i,k},y_{i,k}\}_{i=1}^{D_k}$ 进行模型训练，训练目标为最小化模型的损失函数，即

$$\min_{w_k^r}\mathcal{L}(w_k^r)=\sum_{i=1}^{D_k}\frac{1}{D_k}\mathcal{L}(x_{i,k};w_k^r) \tag{1}$$

其中，$D_k=\left|\mathcal{D}_k\right|$ 表示终端 k 的本地数据量。在完成多次训练后将模型参数 w_k^r 上传给中央节点，从而进入全局聚合阶段。

2) 全局聚合阶段。中央节点收集来自多个终端的模型参数 $\{w_k^r\}_{k=1}^{S^r}$ 后，执行模型聚合策略，原始的加权聚合方式为

$$w^r=\sum_{k=1}^{S^r}\frac{D_k}{\sum\limits_k D_k}w_k^r \tag{2}$$

其中，S^r 表示参与第 r 轮聚合的终端数量。中央节点生成第 r 轮的全局模型后，选择部分终端下发更新后的全局模型参数，开启第 $r+1$ 轮模型训练。

通过迭代式地进行本地训练和全局聚合，最终 FL 系统生成一个对所有终端都适用的全局模型。

1.2　持续学习

通过以上描述可知，FL 假设在训练过程中，终端的本地数据集 $\mathcal{D}_k$ 保持不变。与之不同的是，持续学习适用于数据流式变化的场景。

在持续学习中，训练新数据视为新任务。因此，在整个持续学习的模型训练过程中，可以将流式数据划分成 T 个任务集 $\mathcal{T}=\{\mathcal{D}^t\}_{t=1}^{T}$，其中 $\mathcal{D}^t=\{x_i^t,y_i^t\}_{i=1}^{D^t}$ 表示第 t 个任务的数据集。

由于网络系统中存在新旧 2 种数据，持续学习面临“稳定性–可塑性”困境。稳定性表示当前模型具备对旧数据的数据挖掘能力，可塑性表示模型能够扩增对新数据的数据挖掘能力。现有的持续学习算法的目标是提高模型的稳定性，降低模型对旧数据的遗忘速度。但是过度保障旧数据的数据挖掘能力，如不训练新数据，会影响模型的可塑性，无法实现演进式智能。因此需要根据具体场景设定具体的算法目标，权衡模型的稳定性和可塑性。

2　高性能联邦持续学习算法

2.1　系统概述

本文在终端离散分布的大规模分布式网络场景中针对联邦持续学习展开研究。本文探索的联邦持续学习算法是针对分类任务设计的一种通用的演进式智能方法，可以运用在智慧医疗、智慧家庭、自动驾驶等需要进行数据分类的场景。

2.2 问题定义

定义网络中参与联邦学习的终端数量为 K，终端 k 在第 t 个任务下的数据集为 $\mathcal{D}_k^t=\{x_{i,k}^t,y_{i,k}^t\}_{i=1}^{D_k^t}$，类别集为 $\mathcal{Y}_k^t$，数据分布为 $\boldsymbol{D}_k^t=\{D_{k,\gamma}^t\}_{\gamma\in\mathcal{Y}_k^t}$，其中 $D_{k,\gamma}^t$ 表示终端 k 的第 t 个任务中属于类别 γ 的数据量。

为了实现对流式数据的精准分类，FCL-HSD 的训练目标是最大化每个终端模型对本地所有数据的分类准确率，即最小化整体分类误差，表示为

$$\min_{\{w_k\}_{k=1}^K}\sum_{(x_{i,k},y_{i,k})\in\bigcup_{t=1}^T\mathcal{D}_k^t}\frac{1}{\left|\bigcup_{t=1}^T\mathcal{D}_k^t\right|}\mathcal{L}(\boldsymbol{P}(x_{i,k};w_k),y_{i,k}) \quad (3)$$

其中，$\boldsymbol{P}(x_{i,k};w_k)$ 为模型 w_k 对样本 $x_{i,k}$ 的预测概率，$y_{i,k}$ 为样本 $x_{i,k}$ 的真实标签。

为了实现上述目标，本文提出了 FCL-HSD 算法，分别从 FL 的模型本地训练阶段和全局聚合阶段进行优化，提升模型性能。

2.3 基于结构可扩展模型的本地模型训练范式

在本地训练阶段，FCL-HSD 受到 DER（dynamically expandable representation）[13]的启发，根据分类模型的结构特点，将模型划分为特征提取器和分类器，通过动态扩展模型结构，执行存储旧特征提取器、生成新特征提取器、扩充分类器维度等操作，保障终端对旧数据的识别能力，扩增终端对新数据的识别与分类能力。DER 仅针对单个终端提出存储部分模型参数，不适用于联邦场景。FCL-HSD 研究了分布式场景下模型扩展方式，考虑更实际的数据增量问题，制定模型扩展审核机制，优化本地存储规模并统一模型扩展结构，使本地模型可以参与 FL 的全局聚合。具体内容描述如下。

本文采用的数据分类模型属于深度学习（DL, deep learning）模型。模型训练时，首先通过对输入数据进行特征提取，从而识别样本数据；然后通过分析提取到的特征对数据进行分类。因此，用于分类任务的 DL 模型主要由 2 个部分组成：特征提取器 $\mathcal{F}$ 和分类器 Φ。

模型扩展审核机制。FCL-HSD 算法中，本地模型结构扩展方式如图 1 所示。当终端 k 本地产生了新数据集 $\mathcal{D}_k^t$ 从而开启第 t 个任务的训练时，触发了模型结构扩展操作，需要扩充分类器维度并判断是否新增特征提取器。终端扩展模型时，需要在 FL 系统内进行结构统一扩展，以防不统一的模型结构导致多个终端模型之间无法进行全局聚合。因此，终端 k 首先统计第 t 个任务的数据分布 $\boldsymbol{D}_k^t=\{D_{k,\gamma}^t\}_{\gamma\in\mathcal{Y}_k^t}$ 并发送给中央节点。$\boldsymbol{D}_k^t$ 中只统计了终端中每个类别的数据量，不包含具有隐私性的原始训练数据，因此终端向第三方节点传输统计信息仍然满足 FL 的隐私保护前提。中央节点收集并记录来自所有终端的数据分布后，生成第 t 个任务下的全局数据分布 $\boldsymbol{D}^t=\bigcup_{k=1}^K\boldsymbol{D}_k^t$，以及对应的全局类别集合 $\mathcal{Y}^t=\bigcup_{k=1}^K\mathcal{Y}_k^t$，并将 $\mathcal{Y}^t$ 下发给所有终端。

终端 k 根据 $\mathcal{Y}^t$ 扩展模型分类器 Φ_k^t，使 Φ_k^t 的输出维度为 FL 系统内所有类别数 $\left|\bigcup_{\tau=1}^t\mathcal{Y}^\tau\right|$。此外，终端还需判断是否新增一个特征提取器。若需要新增，则复制旧特征提取器 $\mathcal{F}_k^{n-1}$，初始化新特征提取器 $\mathcal{F}_k^n$。考虑到实际场景中，终端产生的数据类型动态变化且逐渐增加，多个任务之间数据类别分布具有重叠性。若为每个任务 t 保存一个特征提取器，既造成了多个特征提取器之间功能冗余，又极大地增加了本地的存储开销。所以，本文设计了模型扩展审核机制，目标是在保证模型对新旧数据具有识别功能的前提下，尽量减少存储开销。具体审核方式为，依次衡量第 t 个任务的类别分布与前 $n-1$ 个

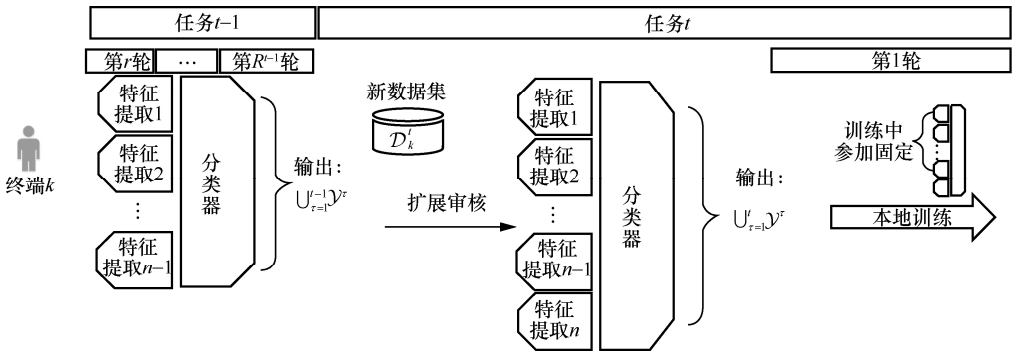


图 1 本地模型结构扩展方式

特征提取器对应的类别分布的相似度 $\varsigma_k^{\eta,t}$，即

$$\varsigma_k^{\eta,t}=\frac{\mathcal{Y}^{\mathcal{F}_k^\eta}\mathcal{Y}^t}{\left\|\mathcal{Y}^{\mathcal{F}_k^\eta}\right\|\left\|\mathcal{Y}^t\right\|},\eta=1,\cdots,n-1 \tag{4}$$

若所有的相似度 $\varsigma_k^{\eta,t}$ 均小于阈值 λ_l，则当前任务的类别分布与所有的旧特征提取器对应的类别分布相似程度低，第 t 个任务包含较多新类别的数据，因此需要增加新特征提取器 $\mathcal{F}_k^n$；若有相似度 $\varsigma_k^{\eta,t}$ 超出阈值 λ_l，则用最相似的旧特征提取器进行第 t 个任务的模型训练。

本地模型训练过程。当新增特征提取器 $\mathcal{F}_k^n$ 后，终端 k 用于训练第 t 个任务的模型为 $w_k^t=\{\mathcal{F}_k,\Phi_k^t\}=\{\mathcal{F}_k^1,\cdots,\mathcal{F}_k^{n-1},\mathcal{F}_k^n,\Phi_k^t\}$，从而开始使用第 t 个任务的本地数据集 $\mathcal{D}_k^t$ 进行模型训练，更新第 t 个任务对应的特征提取器和分类器 $\{\mathcal{F}_k^n,\Phi_k^t\}$。

本地训练具体流程如图 2 所示。在模型训练过程中，每个样本 $x_{i,k}$ 需要通过 n 个特征提取器生成 n 个多维特征图。然后将每个特征图进行平铺，展成一维特征图，进而输入分类器 Φ_k^t。在执行反向梯度传播以更新模型参数时，为了保存对旧数据的识别能力，终端需要固定前 $n-1$ 个特征提取器的模型参数，只更新 $\{\mathcal{F}_k^n,\Phi_k^t\}$ 的模型参数从而提升模型对新数据的识别与分类能力。

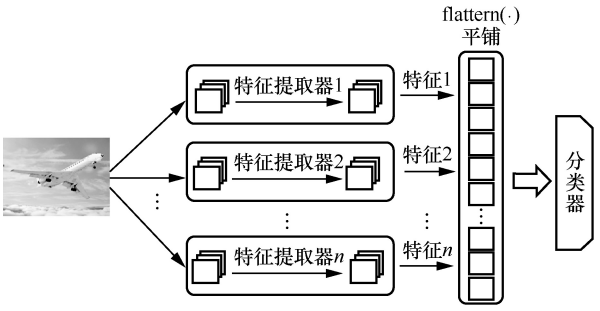


图 2　本地训练具体流程

当完成第 r 轮本地训练后，终端 k 将更新后的模型参数 $\{\mathcal{F}_k^{n,r},\Phi_k^{t,r}\}$ 上传至中央节点，等待中央节点下发聚合后的模型参数。

上述过程的具体步骤如算法 1 所示。

算法 1　终端 k 本地结构可扩展模型训练算法

输入　第 t 个任务的模型训练轮数 R^t、数据集 $\mathcal{D}_k^t$、相似度阈值 λ_l

输出　完整模型参数 $w_k^t=\{\mathcal{F}_k,\Phi_k^t\}$

1) 统计本地数据分布 $\boldsymbol{D}_k^t$ 并上传给服务器
2) 接收服务器下发的全局类别集合 $\mathcal{Y}^t$
3) 将分类器 Φ_k^t 维度扩展为 $\left|\bigcup_{\tau=1}^t\mathcal{Y}^\tau\right|$
4) 初始化最大相似度 max_sim=0，定义最相似特征提取器 $\tilde{\eta}=n$
5) for $\eta=1,\cdots,n-1$
6) 根据式(4)计算 $\mathcal{Y}^t$ 与 $\mathcal{Y}^{\mathcal{F}_k^\eta}$ 的余弦相似度 $\varsigma_k^{\eta,t}$
7) if $\varsigma_k^{\eta,t}>$ max_sim
8) max_sim= $\varsigma_k^{\eta,t}$, $\tilde{\eta}=\eta$
9) end if
10) end for
11) if max_sim $<\lambda_l$
12) 新增第 n 个特征提取器
13) else 采用第 $\tilde{\eta}$ 个特征提取器进行本地模型训练
14) end if
15) for $r=1,\cdots,R^t$
16) 基于数据集 $\mathcal{D}_k^t$ 训练模型 $\{\mathcal{F}_k^r,\Phi_k^r\}$，且只更新部分模型参数 $\{\mathcal{F}_k^{t,r},\Phi_k^{t,r}\}$
17) 上传参数 $\{\mathcal{F}_k^{t,r},\Phi_k^{t,r}\}$ 至服务器
18) 接收全局模型参数 $\{\mathcal{F}_{g,k}^{t,r},\Phi_{g,k}^{t,r}\}$，更新本地完整模型 $\{\mathcal{F}_k^{r+1},\Phi_k^{r+1}\}$
19) end for

通过算法 1 可知，相较于存储旧数据的方法，FCL-HSD 通过存储部分模型参数来减缓模型遗忘旧数据的速度，此存储开销与特征提取器大小、特征提取器个数有关。不同的“特征提取器–分类器”划分方式会产生不同的存储开销。定义单个特征提取器大小为 $M^{\mathcal{F}}$，极端数据增量情况下，任务之间数据类别分布相似度均低于阈值 λ_l，此时本文所提 FCL-HSD 算法会为每个任务新增一个特征提取器，因此算法 1 的空间复杂度为 $O(TM^{\mathcal{F}})$。而 GLFC[15] 的终端侧需要存储完整的旧模型，存储开销与模型大小 M 有关，空间复杂度为 $O(M)$。不同数据增量场景、不同 AI 模型下，算法 1 在总任务数少、特征提取器小的场景中存储开销低于 GLFC。算法 1 的时间复杂度与每次训练开销、每个任务的训练轮数 R、总任务数 T 有关。假设每次训练开销为 $O(\zeta)$，该开销受到终端数据量 D 和模型大小 M 影响，则算法 1 的时间复杂度为 $O(\zeta TR)$。算法 1 的时间复杂度与 GLFC、传统 FL 算法 FedAvg[4]一致。

2.4 基于数据分布相似度的全局模型定制化策略

为了解决数据异构性造成的传统聚合方法中全局模型不适配问题，FCL-HSD 提出了全局模型定制化策略，根据模型结构特点采用分块聚合方式，并设计了一种基于数据分布相似度的模型贡献度衡量方法，使中央节点依次对单个终端历史数据分布与其他终端当前数据分布进行余弦相似度分析，量化每个终端的模型贡献度，用于生成定制化的全局模型，进一步提升终端对本地新旧数据的识别与分类能力，详细方案描述如下。

中央节点收到来自多个终端的本地部分模型参数 $\{\mathcal{F}_k^{t,r},\boldsymbol{\Phi}_k^{t,r}\}_{k=1}^{S^r}$ 后，执行模型聚合操作。考虑到接收到的终端模型参数中，特征提取器 $\{\mathcal{F}_k^{t,r}\}_{k=1}^{S^r}$ 用于识别当前数据，而分类器 $\{\boldsymbol{\Phi}_k^{t,r}\}_{k=1}^{S^r}$ 既需要实现对当前数据的分类能力，又不能遗忘对旧数据的分类能力。因此特征提取器和分类器负责的数据范围不同，故本文提出的 FCL-HSD 算法在聚合阶段采用了分块聚合方式。具体体现为分类器聚合和特征提取器聚合算法。

分类器聚合。考虑到分类器会偏向于对当前数据类别进行分类，从而遗忘旧数据的分类能力，因此聚合分类器参数时需要解决灾难性遗忘问题。由于不同终端之间数据分布异构但类别分布具有相关性，例如，终端 2 当前任务中的数据类别可能是终端 1 旧任务中的数据类别，为了降低终端 1 对旧类别的遗忘速度，此时应提高终端 2 对终端 1 的模型贡献度。以此类推，中央节点需要计算全局分类器的重要性参数向量 $\boldsymbol{\mathcal{A}}^{\Phi}=\{\boldsymbol{\mathcal{A}}_k^{\Phi}\}_{k=1}^{K}$。终端 k 的全局分类器重要性参数向量 $\boldsymbol{\mathcal{A}}_k^{\Phi}=[a_{k,1}^{\Phi},\cdots,a_{k,j}^{\Phi},\cdots,a_{k,K}^{\Phi}]$，其中 $a_{k,j}^{\Phi}$ 表示终端 j 的分类器对终端 k 分类器的贡献度。贡献度的具体衡量方式为中央节点记录终端 k 的旧数据分布，表示为

$$\boldsymbol{D}_k^{\text{old}}=\bigcup_{\tau=1}^{t-1}\boldsymbol{D}_k^{t}=\{D_{k,\gamma}^{\text{old}}\}_{\gamma\in\bigcup_{\tau=1}^{t}\mathcal{Y}^{\tau}} \tag{5}$$

其中，$D_{k,\gamma}^{\text{old}}$ 表示终端 k 的历史数据中属于类别 γ 的数据量。对于系统内参与聚合的终端 j，中央节点根据终端 j 当前任务的数据分布 $\boldsymbol{D}_j^t$，计算其与终端 k 的历史数据分布的相似度，即

$$\varsigma_{k,j}^{\Phi}=\frac{\boldsymbol{D}_k^{\text{old}}\boldsymbol{D}_j^t}{\left\|\boldsymbol{D}_k^{\text{old}}\right\|\left\|\boldsymbol{D}_j^t\right\|},j=1,\cdots,S^r,j\neq k \tag{6}$$

从而得到终端 k 的全局分类器相似度向量 $\boldsymbol{\delta}_k^{\Phi}=[\varsigma_{k,1}^{\Phi},\cdots,\varsigma_{k,j}^{\Phi},\cdots,\varsigma_{k,K}^{\Phi}]$。模型聚合时，本地模型与其他终端模型的全局贡献度衡量方式不同，因此在相似度向量中 $\varsigma_{k,k}^{\Phi}=0$。将全局相似度向量归一化后，得到全局分类器重要性参数向量 $\boldsymbol{\mathcal{A}}_k^{\Phi}=[a_{k,1}^{\Phi},\cdots,a_{k,j}^{\Phi},\cdots,a_{k,k}^{\Phi}]$，其中 $a_{k,j}^{\Phi}$ 的计算方式为

$$a_{k,j}^{\Phi}=\frac{\varsigma_{k,j}^{\Phi}}{\sum_{\mathcal{J}}\varsigma_{k,\mathcal{J}}^{\Phi}} \tag{7}$$

特征提取器聚合。特征提取器负责对当前任务数据进行特征提取，终端 j 当前数据分布 $\boldsymbol{D}_j^t$ 与终端 k 当前数据分布 $\boldsymbol{D}_k^t$ 越相近，则终端 j 对终端 k 的特征提取器的模型贡献度越高，从而终端 k 对当前数据的特征提取能力越强。终端 k 的全局特征提取器重要性参数向量可以表示为 $\boldsymbol{\mathcal{A}}_k^{\mathcal{F}}=[a_{k,1}^{\mathcal{F}},\cdots,a_{k,j}^{\mathcal{F}},\cdots,a_{k,K}^{\mathcal{F}}]$，其中

$$\varsigma_{k,j}^{\mathcal{F}}=\frac{\boldsymbol{D}_k^t\boldsymbol{D}_j^t}{\left\|\boldsymbol{D}_k^t\right\|\left\|\boldsymbol{D}_j^t\right\|} \tag{8}$$

$$a_{k,j}^{\mathcal{F}}=\frac{\varsigma_{k,j}^{\mathcal{F}}}{\sum_{\mathcal{J}}\varsigma_{k,\mathcal{J}}^{\mathcal{F}}} \tag{9}$$

其中，$\varsigma_{k,k}^{\mathcal{F}}=0$。同一个任务中，终端数据集不变，因此全局重要性参数向量 $\boldsymbol{\mathcal{A}}_k^{\bullet}$ 只需计算一次。

至此中央节点计算得到了所有终端的全局特征提取器/分类器重要性参数向量。考虑到终端之间的数据异构性，为了提升每个终端本地模型的性能，FCL-HSD 采用分块聚合方式，为每个终端定制个性化的全局特征提取器 $\mathcal{F}_{g,k}^t$ 和全局分类器 $\boldsymbol{\Phi}_{g,k}^t$。

$$\mathcal{F}_{g,k}^t=\lambda_1^{\mathcal{F}}\mathcal{F}_k^t+\lambda_2^{\mathcal{F}}\sum_{j\in K,j\neq k}a_{k,j}^{\mathcal{F}}\mathcal{F}_j^t \tag{10}$$

$$\boldsymbol{\Phi}_{g,k}^t=\lambda_1^{\Phi}\boldsymbol{\Phi}_k^t+\lambda_2^{\Phi}\sum_{j\in K,j\neq k}a_{k,j}^{\Phi}\boldsymbol{\Phi}_j^t \tag{11}$$

其中，$\lambda_1^{\bullet}+\lambda_2^{\bullet}\sum_{j\in K,j\neq k}a_{k,j}^{\bullet}=1$，$\lambda_1^{\bullet}$ 和 $\lambda_2^{\bullet}$ 是可调整的全局超参数。

然后，中央节点将本轮更新后的 $\{\mathcal{F}_{g,k}^t,\boldsymbol{\Phi}_{g,k}^t\}$ 下发给终端 k，从而开启下一轮模型训练。

上述过程的具体步骤如算法 2 所示。

算法 2 中央节点侧全局模型定制化算法

输入 全局超参数 $\lambda_1^{\mathcal{F}},\lambda_2^{\mathcal{F}},\lambda_1^{\Phi},\lambda_2^{\Phi}$

输出 部分模型参数 $\{\mathcal{F}_{g,k}^{t,r},\boldsymbol{\Phi}_{g,k}^{t,r}\}_{k=1}^{S^r}$

1) 接收终端的本地数据分布 $\{\boldsymbol{D}_k^t\}_{k=1}^K$
2) 更新全局数据类别分布信息表
3) 向所有终端下发全局类别集合 $\mathcal{Y}^t$
4) for $r=1,\cdots,R^t$
5) 选择参与模型聚合的终端 S^r
6) 接收来自终端的 $\{\mathcal{F}_k^{t,r},\boldsymbol{\Phi}_k^{t,r}\}_{k=1}^{S^r}$
7) for $k=1,\cdots,S^r$
8) 分别根据式(10)和式(11)计算 $\mathcal{F}_{g,k}^{t,r}$ 和 $\boldsymbol{\Phi}_{g,k}^{t,r}$
9) 向终端 k 下发 $\{\mathcal{F}_{g,k}^{t,r},\boldsymbol{\Phi}_{g,k}^{t,r}\}$
10) end for
11) end for

通过算法 2 可知，在本文设计的全局模型聚合算法中，中央节点需要额外存储一个全局数据类别分布信息表。随着数据的持续增加，该表逐渐包含所有终端不同任务下的数据分布信息，因此该表最大维度为 $O(TK)$，即算法 2 的空间复杂度。但是中央节点具有较丰富的存储资源，而全局数据类别分布信息表只记录了终端不同任务下的统计信息。相较于 GLFC[15] 的中央节点需要存储具有高维数据特征的模型参数，本文只存储一张表的空间开销是可以容忍的。此外，中央节点在联邦学习的一轮模型训练中只负责执行简单的乘加操作进行模型聚合，单次模型聚合的时间开销较低。定义中央节点侧一次模型聚合所需的时间开销为 $O(1)$，则算法 2 的时间复杂度为 $O(TRK)$。相较于 FedAvg[4]，即中央节点为所有终端下发统一的全局模型，其时间复杂度为 $O(TR)$，本文采用全局模型定制化策略，时间复杂度为 FedAvg 的 K 倍。然而，在 GLFC 中，中央节点除了执行模型聚合操作外，还需要额外训练一个数据生成模型，每一次模型训练需要进行数万次乘加操作。相较于 GLFC 服务器侧的算法时间复杂度，算法 2 的时间复杂度几乎可以忽略不计。

3 仿真与性能分析

本节介绍 FCL-HSD 的仿真实验设置，展示实验结果，并从准确率变化趋势、新旧类别最终准确率、每个终端的平均准确率等方面对所实现的算法进行性能评估。

3.1 实验设置

1) 数据集

为了验证算法的可行性与有效性，本文采用了在真实环境下采样生成的 CIFAR-100 数据集[19]、CIFAR-10 数据集[19]和 ISCX 数据集[20]进行性能评估。CIFAR-100 数据集共包含 100 个类别（即细粒度类别），分属于 20 个大类（即粗粒度类别）。每个细粒度类别有 600 张图像，其中 500 张作为训练集，100 张作为测试集。每张图像是 32 像素×32 像素的彩色图像。每张图像有 2 个标签：粗粒度标签和细粒度标签。本文采用细粒度标签来指导模型训练。CIFAR-10 数据集共有 60 000 张彩色图像，涵盖 10 类数据，类别编号为 0～9。每类数据包含 6 000 张图像，其中 5 000 张作为训练集，1 000 张作为测试集。采用 CIFAR 系列数据集进行性能测试是为了体现本文算法可以在涉及图像分类任务的场景中使用，包括智慧相册、人脸识别、智慧医疗、自动驾驶等分布式场景。ISCX 是 New Brunswick 大学发布的网络流量数据集，包含多个版本，本文选择具有不同应用数据的 ISCXVPN2016 数据集进行实验。通过 wireshark 软件和 tcpdump 命令捕获不同类型的会话流量，最终生成包括 15 类应用的加密流量数据集，类别编号为 0～14。ISCXVPN2016 数据集中的每个样本是一条 pcap 格式的数据流。实验前，需要对样本进行预处理，通过矢量化操作将 pcap 格式的数据流转换成二维图像。每一类应用数据选择 800 张图像作为训练集，200 张图像作为测试集。采用 ISCX 系列数据集进行性能测试是为了体现即使在非图像场景中，对数据进行格式转换后，本文算法依然可以向具有流式数据的终端提供诸如业务分类、恶意流量检测、个性化推荐等各种智能服务。

为了模拟更加真实的终端流式数据增量场景，本文对终端数据集进行如下动态构建：定义 2 种类型训练任务，大任务 B 和小任务 b；每个大任务包含多个小任务；大任务之间的数据类别不重叠；同一个大任务下小任务之间的数据类别有部分重叠。因此根据 FCL-HSD 的模型扩展审核机制，终端的特征提取器个数即大任务数。本文以“$\{\mathcal{B}_\beta\}_{\beta=1}^B$-$b$-$\mathscr{b}$”标记实验的数据设置，其中 $\mathcal{B}_\beta$ 表示第 β 个大任务中的类别数量，b 表示每个大任务下的小任务数量，$\mathscr{b}$ ($\mathscr{b}\leqslant\mathcal{B}_\beta$)表示每个小任务中类别数量。例如，“{5,5}-3-2”表示将包含 10 类数据的完整数据集划分为 2 个分别有 5 类数据的大任务，每个大任务包含 3 个小任务，从 5 类数据中任意选择两类数据作为每个小任务的训练集。

2) 模型

本文采用经典的 DL 模型 AlexNet[21]作为终端待训

练的模型，AlexNet 主要由 5 层卷积层和 3 层全连接层构成，在本文实验中，将 5 层卷积层和前两层全连接层作为特征提取器，最后一层全连接层作为分类器。

3) 实验环境

本文在搭载有 Nvidia RTX 3090Ti GPU 的服务器上进行了模拟实验。在模拟环境下，本文实现了 FL 系统和多个联邦持续学习算法，系统包含一个中央节点与 $K(K \in \mathbb{N}^+)$ 个终端。在本文实验中，分别取 $K = 5,10$ 来验证算法性能。

4) 基准方法

为了展示本文所提算法的优越性，本文实现了多个现有工作进行性能对比，具体如下。① GLFC[15]。该研究首次在 FL 中研究类别增量的工作，中央节点需要收集本地模型的梯度信息，用于训练数据生成模型并重建代理数据集，通过代理数据集选择全局最佳旧模型。② FLIC[14]。该研究运用 FL 框架实现网络流量分类的分布式协议，通过扩充模型最后一层的输出维度来实现对新数据的分类。③ Fed-Bic。Bic[7]设计了矫正模型来修正原始模型的数据偏置现象，本文将其扩展成联邦形式，即采用传统的加权聚合算法[4]生成全局模型。④ Fed-iCaRL。iCaRL[12]是首次提出通过存储部分旧数据来缓解灾难性遗忘问题的工作，本文将其扩展成联邦形式。

5) 实验指标

本文采用模型准确率作为评估算法性能的指标。在每个终端每一轮训练后进行一次模型测试，并记录测试准确率。每一轮测试集包含的数据类别为当前终端本地出现过的所有类别。

3.2 性能评估

3.2.1 FCL-HSD 纵向实验与结果分析

本文在 CIFAR 系列数据集和 ISCX 数据集上验证了所提算法 FCL-HSD 的可行性，训练不同数据集时模型测试准确率变化趋势如图 3 和图 4 所示。图 3 和图 4 展示的是每一轮训练后模型准确率的统计值，统计的是所有终端在每一轮的平均准确率。每个子图的图名表示“数据集–{每个大任务包含的类别数}–大任务下的小任务数–每个小任务包含的类别数–终端总数–每轮参与聚合的终端数量”。不同子图展示的是不同数据增量场景的模型准确率变化趋势。由于数据集增量方式不一样，任务总数具有差异化，因此不同子图中的总聚合轮数也具有差异化。

准确率量化了模型对当前产生过的所有类别数据的分类能力。从图 3 和图 4 中可以看出，不同增量场景中，每新增一个小任务后，模型准确率都会有不同程度的降级，原因在于开启新任务后，模型尚未训练新数据，无法扩充对新数据的分类能力，导致准确率产生了阶跃现象。尤其是新增第二个小任务后，新类别在所有已出现过的类别中占有较高比重，因此准确率降级明显。在图 3 所有实验中，每个小任务下的模型需要聚合 20 轮。所以在图 3(a)、图 3(b)中第 20 轮处，即模型在训练第二个任务的初始时刻，其准确率相较于第一个任务的准确率产生了大幅度降级。随着模型训练对新数据的分类能力，以及 FCL-HSD 对旧类遗忘问题的处理，每个任务内的模型准确率逐渐升高。

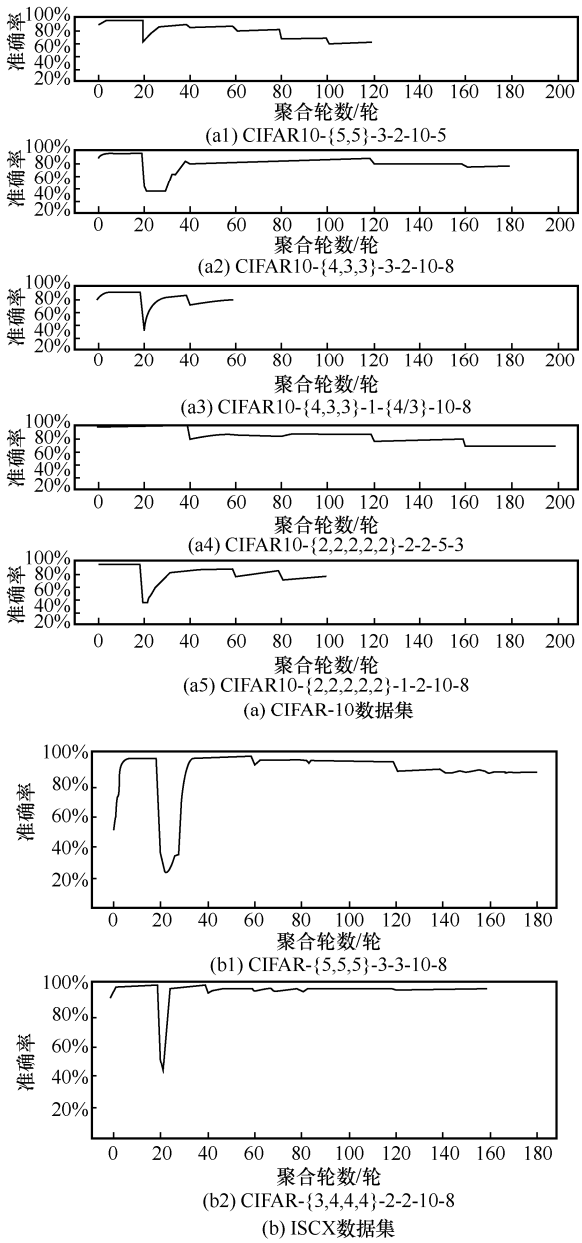
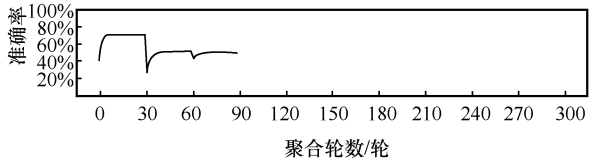
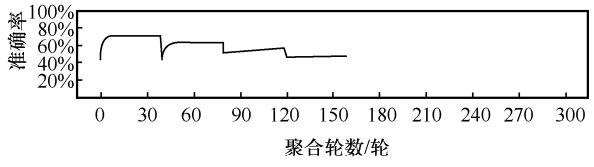


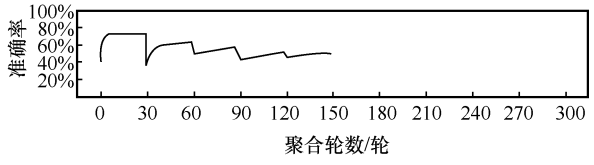
图 3 训练不同数据集时模型测试准确率变化趋势



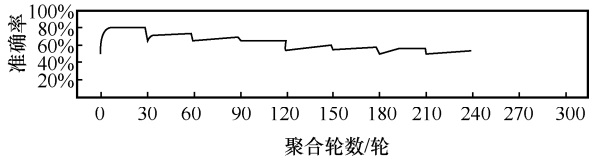
(a) CIFAR100-{30,35,35}-1-30-5-3



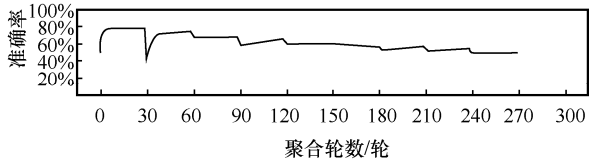
(b) CIFAR100-{50,50}-2-30-5-3



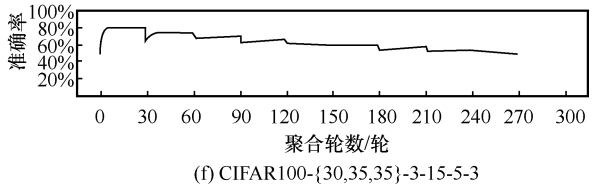
(c) CIFAR100-{20,20,20,20,20}-1-20-5-3



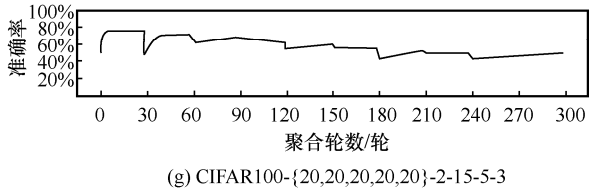
(d) CIFAR100-{25,25,25,25}-1-15-5-3



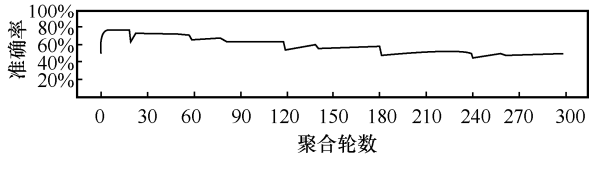
(e) CIFAR100-{40,30,30}-3-15-5-3



(f) CIFAR100-{30,35,35}-3-15-5-3



(g) CIFAR100-{20,20,20,20,20}-2-15-5-3



(h) CIFAR100-{20,20,20,20,20}-3-15-5-3

图 4　训练 CIFAR-100 数据集时模型测试准确率变化趋势

对比不同数据增量场景中模型准确率的变化可以发现，最终准确率既与参与聚合的终端比例有关，也与总任务数有关。通过图 3(b)可以发现，在终端数量相同的情况下，总任务数越多，最终准确率越低，这是由于模型训练的任务数越多，分类器训练新数据时越容易遗忘更“旧”的历史数据，故而影响了模型的整体准确率。但是，在图 3(a)中，“CIFAR10- {5,5}-3-2-10-5”（总任务数为 6）场景中模型最终准确率低于“CIFAR10-{4,3,3}-3-2-10-8”（总任务数为 9）。除了前者具有较低的终端参与聚合率这一因素外，另一个潜在原因是，实验采用的数据集的总数据量是固定的，因此将整个数据集划分成包含多个小任务的大任务时，总任务数越多的实验场景中模型因为同类数据训练次数更多而产生更高的准确率。在“CIFAR10-{5,5}-3-2”中，每个大任务下有 5 类数据，从中任意选择两类数据作为一个小任务的训练集，重复选择 3 次，平均每类数据的训练次数较“CIFAR10-{4,3,3}-3-2”低。因此上述 2 个因素导致“CIFAR10-{5,5}-3- 2-10-5”场景中模型的最终准确率低于“CIFAR10-{4,3,3}- 3-2-10-8”。

在图 4 展示的多个子图中，总任务数不一致，但不同任务下总终端数以及每轮参与训练的终端数保持一致。从图 4 整体可以看出，训练的任务数越多，模型准确率变化幅度越大，模型最终准确率越低。此现象进一步验证了总任务数会影响模型最终准确率。此外，在图 4 中，“CIFAR100-{50,50}-2-30-5-3”场景下每 40 轮新增一个小任务，“CIFAR100- {20,20,20,20, 20}-3-15-5-3”场景每 20 轮新增一个小任务，其余数据增量场景中每 30 轮新增一个小任务，因此图 4 展示的模型准确率每隔 30 轮（或 20 轮、40 轮）时，会由于训练增量任务而产生准确率阶跃现象。对比“CIFAR100-{30,35,35}-1-30- 5-3”、“CIFAR100-{20,20, 20,20,20}-1-20-5-3”和其他数据增量场景，可以发现，当每个大任务中只包含一个小任务时，准确率阶跃现象明显。原因在于，若每个大任务中只包含一个小任务，则小任务间数据类别相似度太低，甚至趋近于 0。此时，会导致模型在训练增量任务时，无法利用已有的分类能力对新数据进行分类。因此，在此数据增量场景下，当产生增量任务，尤其是新增第二个任务时，准确率下降幅度大。此外，在“CIFAR100-{20,20,20,20,20}-1-20-5-3”、“CIFAR100-{20,20,20,20,20} -2-15-5-3”和“CIFAR100-{20,20,20,20,20}-3-15-5-3”数据增量场景中，虽然总任务数从 5 增加到 10、15，但由

于本文设计了模型扩展审核机制，因此终端侧在上述 3 个数据增量场景下均只存储 5 个特征提取器和一个分类器，极大地降低了在多任务增量场景下终端本地的存储开销。

综合对比图 3 和图 4 可以发现，模型训练 ISCX 数据集的测试准确率比训练 CIFAR 系列数据集时更高。原因在于，ISCX 数据集中记录的是网络流量的统计特征，如分组到达间隔时间、流持续时间、分组长度分布等[22-24]，数据信息较简单，因此通过矢量化操作后生成灰色图像即可表达原始数据的所有信息。而 CIFAR 系列数据集是彩色图像。相较于灰色图像，彩色图像富含的信息更多，导致模型训练速度慢。相同训练次数下，训练灰色图像得到的准确率会高于训练彩色图像得到的准确率。此外，CIFAR-100 数据集包含的类别数量远高于 CIFAR-10 数据集和 ISCX 数据集中的类别数量，采用不同数据集训练相同模型时，模型对 CIFAR-100 数据集的分类性能会低于在其他 2 个数据集上的分类性能。但是，在不同数量的终端参与训练的场景下，FCL-HSD 在 CIFAR-10 数据集和 ISCX 数据集上的模型准确率大部分能达到 60%以上，在 CIFAR-100 数据集上的模型准确率大部分能达到 50%以上。

3.2.2 FCL-HSD 横向实验与结果分析

为了展示 FCL-HSD 算法的优越性，本节分别在 CIFAR 系列数据集和 ISCX 数据集上与基准算法进行模型性能对比。

图 5 展示了 CIFAR-100 数据集下不同算法的模型准确率变化趋势，数据增量场景为“cifar100-{25, 25,25,25}-3-10-5-3”，且每 20 轮新增一个小任务。由图 5 可知，相较于所有基准算法，本文所提算法 FCL-HSD 具有最优的性能表现。采用 FCL-HSD 算法训练的模型，在每新增一个小任务时，模型准确率下降幅度最小，且最终准确率能达到 56.2%。而所有基准算法下模型最终准确率均低于 50%。Bic 和 iCaRL 作为集中式的持续学习方法，将其扩展成联邦学习形式后，最终终端的平均准确率达到了 47.3%和 30.7%，没有超过 50%，说明集中式持续学习算法结合传统的加权聚合算法形成的联邦持续学习算法无法解决联邦持续学习中产生的灾难性遗忘问题。原因在于，终端之间数据分布存在异构性，传统的聚合算法忽略了本地个性化的数据分布情况，生成的全局模型扰乱了终端对本地旧数据的分析能力。值得注意的是，GLFC 在 CIFAR-100 数据集上性能完全失效。原因在于，GLFC 在训练阶段引入了两类损失函数，修改了传统的交叉熵损失函数，导致 GLFC 算法中的模型收敛速度慢。在相同训练次数下，采用 GLFC 训练的模型无法及时提升模型准确率。此外，FLIC 即使没有对旧类遗忘问题采取任何解决措施，其最终准确率也能达到 25.5%。原因在于，CIFAR-100 数据集中 100 类数据隶属于 20 个粗粒度标签。属于同一个粗粒度标签的数据，即使不属于同一类别，但具有相似特征，因此 FLIC 在训练新数据时，也小幅提升了模型对同种标签其他类别数据的特征提取能力，在一定程度上缓解了数据遗忘问题。

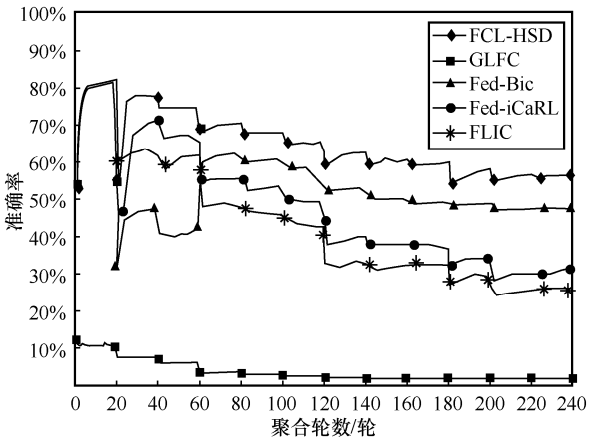


图 5 CIFAR-100 数据集下不同算法的测试准确率变化趋势

CIFAR-10 数据集下不同算法的测试准确率变化趋势如图 6 所示。从图 6 可以看出，相较于基准算法，本文算法 FCL-HSD 在模型训练后期仍然保持着较高的模型准确率，最终准确率达到 61.2%。而 Fed-Bic 和 Fed-iCaRL 在训练简单的 CIFAR-10 数据集时，最终终端的平均准确率只达到 27.6%和 27.7%。原因在于，CIFAR-10 数据集中各类数据之间差异性大，不具有共同特征，因此无法在训练某类数据的同时提升对其他类数据的特征提取能力。综合对比图 5 和图 6 可以发现，Fed-Bic 和 Fed-iCaRL 的可扩展性不佳，无法在多样化的数据集上保持一致的性能表现。FLIC 根据产生的新数据修改模型最后一层的输出维度，使模型可以具备对新数据的分类能力，但是没有提出应对模型灾难性遗忘问题的解决方案，因此模型的平均准确率越来越低，最终只有 11.8%。GLFC 在较简单的数据集上

可以正常运行。然而，在新增任务后，GLFC 性能急剧下降，体现出 GLFC 虽然为每个终端存储了旧模型，但仍存在灾难性遗忘问题。原因在于，GLFC 本地只存储一个全局旧模型，该旧模型需要对整个 FL 系统中所有旧数据具有较高的平均准确率，因此无法对所有终端异构的本地数据都有良好的性能表现。此外，GLFC 中数据生成模型的训练过程是非常耗时的，例如，若每轮更新旧模型前训练 20 次数据生成模型，则保持其他实验设置相同情况下，GLFC 的总训练时间是 FCL-HSD 的 4 倍。因此，在有限的时间里，GLFC 无法训练出合格的数据生成模型用于选择全局最佳旧模型，最终导致终端产生灾难性遗忘现象。

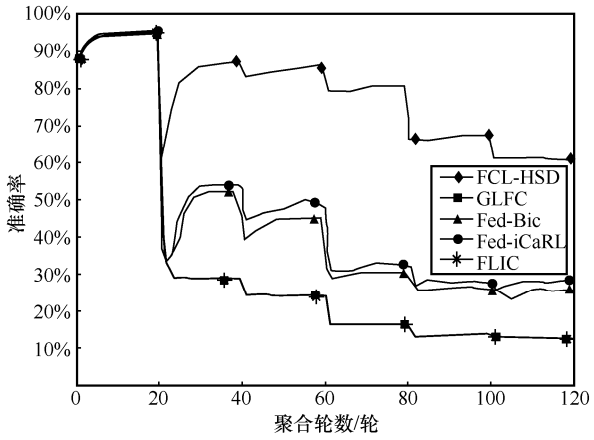


图 6　CIFAR-10 数据集下不同算法的测试准确率变化趋势

CIFAR-10 数据集下不同算法的每类最终准确率如图 7 所示，其中类别编号“0～4”和“5～9”分别属于不同的大任务。从图 7 中可以看出，所有算法在后一个大任务中均有分类能力，然而只有本文算法 FCL-HSD 仍然可以对前一个大任务的数据具有分类能力，体现出 FCL-HSD 对缓解灾难性遗忘问题的有效性。值得注意的是，FCL-HSD 在后一个大任务上的准确率表现并不是最好的，原因在于，FCL-HSD 算法更注重模型的稳定性：一方面，在新任务中，FCL-HSD 通过旧特征提取器约束了分类器训练新数据，减缓了模型对新数据的训练进程；另一方面，FCL-HSD 中采用基于数据分布相似度的全局模型定制化聚合方案，与本地历史数据分布越相似的其他终端，在聚合时贡献程度越高，因此分类器提升了对旧数据的分类能力同时也影响了对新数据的分类能力。

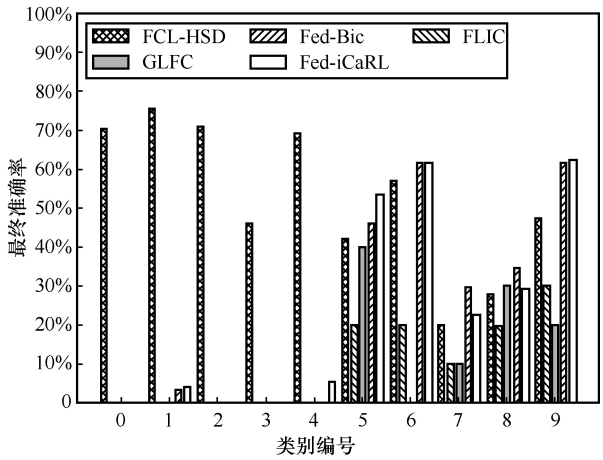


图 7　CIFAR-10 数据集下不同算法的每类最终准确率

ISCX 数据集下不同算法的准确率变化趋势以及每类数据的最终准确率如图 8 和图 9 所示。实验场景为“ISCX-{5,5,5}-3-3-10-8”。从图 8 可以看出，训练 ISCX 数据集时，采用本文提出的 FCL-HSD 算法训练得到的模型，最终准确率可以达到 89.8%，远高于所有基准算法。Fed-Bic 训练得到的模型性能次优，能达到 68.5%，而其他 3 个基准算法中，GLFC 准确率为 7.9%，Fed-iCaRL 和 FLIC 准确率为 7.1%。对此具体分析如下。

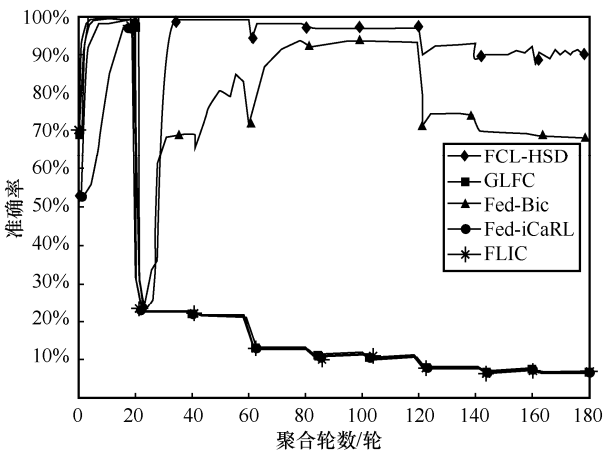


图 8　ISCX 数据集下不同算法的测试准确率变化趋势

首先，Fed-Bic 修改了模型训练流程，通过矫正模型的方式在一定程度上减缓了模型对旧数据的遗忘速度，因此训练简单的灰色图像时模型准确率能达到 68.5%。但是 Fed-Bic 仅在本地训练阶段提出灾难性遗忘的解决方案。而本文算法 FCL-HSD 不仅在终端侧通过存储旧特征提取器保证对旧类的识别能力，并在中央节点侧基于数据分布相似度的方法进行全局模型聚合，提升

分类器对旧类的分类能力。FCL-HSD 优化了联邦学习的本地训练方式和全局聚合方式，极大地减缓了模型对旧数据的遗忘速度，因此能达到最高的模型准确率。

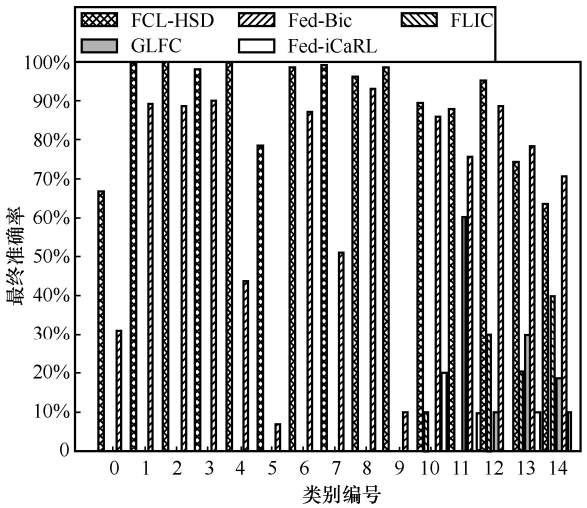


图 9 ISCX 数据集下不同算法的每类数据的最终准确率

与 FCL-HSD 和 Fed-Bic 不同，在“ISCX-{5,5,5}- 3-3”场景下，Fed-iCaRL 的模型性能与训练 CIFAR-10 数据集时相比产生了降级，最终准确率从 27.7%降到了 7.1%。主要原因在于，iCaRL 通过存储旧数据来避免对旧类的遗忘，而“ISCX-{5,5,5}-3-3”场景下共有 9 个任务，在训练第 9 个任务时，Fed-iCaRL 中每个终端需要存储前 8 个任务的旧数据，对于存储空间有限的终端来说，新数据的到来占用了旧数据的存储空间，且训练的任务数越多，每个旧类能存储的数据量越少。因此 Fed-iCaRL 不适用于任务数量较多的场景。图 9 所示为每类数据的最终准确率，其中类别编号“10～14”属于最后一个大任务的数据类别。图 9 也进一步验证了，存储较少旧数据，且新旧数据相互挤兑存储空间，导致 Fed-iCaRL 既遗忘了旧数据的分类能力，又未能有效扩展模型对新数据的分类能力。

3.2.3 FCL-HSD 消融实验与结果分析

在完成 FCL-HSD 算法的可行性和有效性探索后，本文做了消融实验，比较不同聚合策略下每个终端的最终准确率，如图 10 所示。数据增量设置为“ISCX- {5,5,5}-3-3”。FCL-HSD-FedAvg 表示终端采用结构可扩展的模型进行训练，而中央节点采用传统的加权聚合方法生成全局统一的模型。从图 10 可以看出，相较于 FCL-HSD-FedAvg，FCL-HSD 算法在终端上的最终准确率最低提升了 0.46%。最高提升了 8.96%，平均提升了 3%。实验结果证明了 FCL-HSD 采用基于数据分布相似度的全局模型定制化策略是有效的。

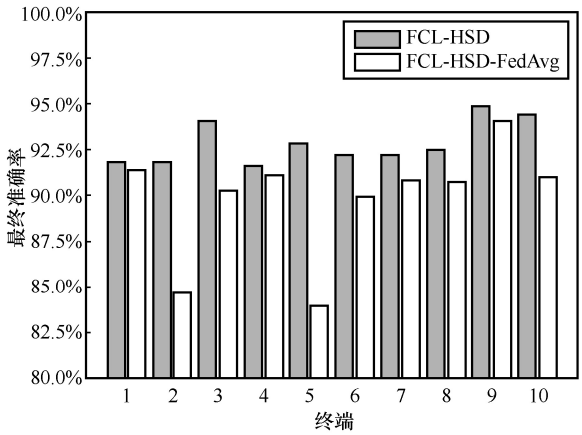


图 10 不同聚合策略下每个终端的最终准确率

3.2.4 FCL-HSD 超参数取值分析

此外，本文探索了 FCL-HSD 中关于 $\{\lambda_1^{\cdot},\lambda_2^{\cdot}\}$ 超参数的设置，不同超参数设定下的测试准确率变化趋势如图 11 所示。数据增量设置为“CIFAR10-{5,5}-3-2”。从图 11 中可以看出，全局分类器超参数 λ_2^{φ} 取较大值时，即与本地历史数据分布相似的其他终端的分类器在模型聚合时贡献较多，则有效减缓了当前分类器对旧数据的遗忘速度；且较大的全局特征提取器超参数 $\lambda_2^{\mathcal{F}}$ 强化了新特征提取器对新数据的识别能力，二者共同作用，提升了终端模型的最终准确率。实验结果证明了 FCL-HSD 中基于数据分布相似度的全局模型定制化策略是有效的。

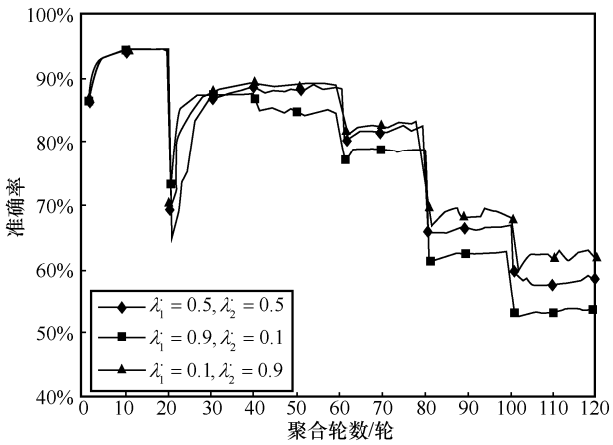


图 11 不同超参数设定下的测试准确率变化趋势

4　结束语

本文考虑到各种万物智联场景中终端数据具有流式增量特点，提出一种面向异构流式数据的联邦持续学习算法 FCL-HSD，缓解因数据持续增加造成模型对旧数据灾难性遗忘问题。FCL-HSD 改进了联邦学习的模型训练与聚合方式，在模型训练阶段，设计了模型结构动态扩展方式和模型扩展审核机制，保存对旧数据的识别能力，同时降低模型存储开销；在全局聚合阶段，针对模型不同模块采用分块聚合形式，并提出了基于数据分布相似度的全局模型定制化策略，缓解因模型聚合引发的旧数据分类能力衰退问题。本文在多种数据增量场景下，验证了 FCL-HSD 的可行性和有效性。相较于现有工作，本文算法在保证对新数据具有分类能力的前提下，有效提升了模型对旧数据的分类能力，实现新旧数据的整体分类能力达到最优。

此外，通过实验数据分析，发现采用所提算法 FCL-HSD 训练得到的模型对新数据的分类能力没有达到最优，即 FCL-HSD 虽然提升了模型对旧数据的“稳定性”，但是在新数据的“可塑性”方面，算法仍然存在优化空间，将在未来展开进一步研究。

参考文献：

[1] 麻省理工科技评论. 2021 年中国数字经济时代人工智能生态白皮书[R]. 2022.
MIT Technology Review. 2021 white paper on artificial intelligence ecology in China's digital economy era[R]. 2022.

[2] BILOGREVIC I, JADLIWALA M, KALKAN K, et al. Privacy in mobile computing for location-sharing-based services[C]//Inter national Symposium on Privacy Enhancing Technologies Symposium. Berlin: Springer, 2011: 77-96.

[3] LIANG X H, LI X, LUAN T H, et al. Morality-driven data forwarding with privacy preservation in mobile social networks[J]. IEEE Transactions on Vehicular Technology, 2012, 61(7): 3209-3222.

[4] KONEČNÝ J, MCMAHAN H B, RAMAGE D, et al. Federated optimization: distributed machine learning for on-device intelligence[J]. arXiv Preprint, arXiv: 1610.02527, 2016.

[5] LE J Q, LEI X Y, MU N K, et al. Federated continuous learning with broad network architecture[J]. IEEE Transactions on Cybernetics, 2021, 51(8): 3874-3888.

[6] SERRA J, SURIS D, MIRON M, et al. Overcoming catastrophic forgetting with hard attention to the task[C]//Proceedings of the International Conference on Machine Learning. New York: ACM Press, 2018: 4548-4557.

[7] WU Y, CHEN Y P, WANG L J, et al. Large scale incremental learning[C]//Proceedings of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2020: 374-382.

[8] HE K M, ZHANG X Y, REN S Q, et al. Deep residual learning for image recognition[C]//Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2016: 770-778.

[9] CASTRO F M, MARÍN-JIMÉNEZ M J, GUIL N, et al. End-to-end incremental learning[C]//European Conference on Computer Vision. Berlin: Springer, 2018: 241-257.

[10] MASANA M, LIU X L, TWARDOWSKI B, et al. Class-incremental learning: survey and performance evaluation on image classification[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023, 45(5): 5513-5533.

[11] LI Z Z, HOIEM D. Learning without forgetting[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018, 40(12): 2935- 2947.

[12] REBUFFI S A, KOLESNIKOV A, SPERL G, et al. iCaRL: incremental classifier and representation learning[C]//Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2017: 5533-5542.

[13] YAN S P, XIE J W, HE X M. D: dynamically expandable representation for class incremental learning[C]//Proceedings of 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2021: 3013-3022.

[14] MUN H, LEE Y. Internet traffic classification with federated learning[J]. Electronics, 2020, 10(1): 27.

[15] DONG J H, WANG L X, FANG Z, et al. Federated class-incremental learning[C]//Proceedings of 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2022: 10154-10163.

[16] LI T, SANJABI M, BEIRAMI A, et al. Fair resource allocation in federated learning[J]. arXiv Preprint, arXiv: 1905.10497, 2019.

[17] ABAD M S H, OZFATURA E, GUNDUZ D, et al. Hierarchical federated learning ACROSS heterogeneous cellular networks[C]//Proceed ings of 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Piscataway: IEEE Press, 2020: 8866-8870.

[18] LONG G, XIE M, SHEN T, et al. Multi-center federated learning: clients clustering for better personalization[J]. arXiv Preprint, arXiv: 2005.01026, 2020.

[19] KRIZHEVSKY A, SUTSKEVER I, GEOFFREY E H. Learning mul-

tiple layers of features from tiny images[J]. Communications of the ACM, 2012, 60(6): 84-90.

[20] DRAPER-GIL G, LASHKARI A H, MAMUN M S I, et al. Characterization of encrypted and VPN traffic using time-related features[C]//Proceedings of the 2nd International Conference on Information Systems Security and Privacy. [S.l]: Scite Press, 2016: 407-414.

[21] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet classification with deep convolutional neural networks[J]. Communications of the ACM, 2017, 60(6): 84-90.

[22] 骆子铭, 许书彬, 刘晓东. 基于机器学习的 TLS 恶意加密流量检测方案[J]. 网络与信息安全学报, 2020, 6(1)77-83.
LUO Z M, XU S B, LIU X D. Scheme for identifying malware traffic with TLS data based on machine learning[J]. Chinese Journal of Network and Information Security, 2020, 6(1)77-83.

[23] PACHECO F, EXPOSITO E, GINESTE M, et al. Towards the deployment of machine learning solutions in network traffic classification: a systematic survey[J]. IEEE Communications Surveys & Tutorials, 2019, 21(2): 1988-2014.

[24] XIE G R, LI Q, JIANG Y. Self-attentive deep learning method for online traffic classification and its interpretability[J]. Computer Networks, 2021, 196: 108267.

[作者简介]

姜慧（1995– ），女，江苏扬州人，中国科学院大学博士生，主要研究方向为联邦学习、边缘智能、分布式机器学习等。

何天流（1999– ），男，江西吉安人，中国科学院大学硕士生，主要研究方向为联邦学习、边缘智能、分布式机器学习等。

刘敏（1976– ），女，河南偃师人，博士，中国科学院计算技术研究所研究员、博士生导师，主要研究方向为移动计算和边缘智能。

孙胜（1990– ），女，河北衡水人，博士，中国科学院计算技术研究所助理研究员，主要方向为联邦学习、移动计算和边缘智能。

王煜炜（1980– ），男，河北唐山人，博士，中国科学院计算技术研究所高级工程师、硕士生导师，主要研究方向为联邦学习、移动边缘计算和下一代网络架构。