智能系统学报
CAAI TRANSACTIONS ON INTELLIGENT SYSTEMS

数字孪生中混合知识蒸馏辅助的异构联邦类增量学习

张铭泉, 贾圆圆, 张荣华

引用本文:
张铭泉， 贾圆圆， 张荣华. 数字孪生中混合知识蒸馏辅助的异构联邦类增量学习[J]. 智能系统学报， 2025， 20(4): 905-915.
ZHANG Mingquan， JIA Yuanyuan， ZHANG Ronghua. Hybrid knowledge distillation-assisted heterogeneous federated class incremental learning for digital twins[J]. *CAAI Transactions on Intelligent Systems*, 2025， 20(4): 905-915.

在线阅读 View online: https://dx.doi.org/10.11992/tis.202406027

第 20 卷第 4 期
2025 年 7 月

智 能 系 统 学 报
CAAI Transactions on Intelligent Systems

Vol.20 No.4
Jul. 2025

DOI: 10.11992/tis.202406027
网络出版地址: https://link.cnki.net/urlid/23.1538.tp.20250221.1355.012

数字孪生中混合知识蒸馏辅助的异构联邦类增量学习

张铭泉[1,2], 贾圆圆[1], 张荣华[1,3]
（1. 华北电力大学 计算机系, 河北 保定 071003; 2. 华北电力大学 河北省能源电力知识计算重点实验室, 河北 保定 071003; 3. 华北电力大学 复杂能源系统智能计算教育部工程研究中心, 河北 保定 071003）

摘　要: 在数字孪生背景下, 联邦学习面临数据非独立同分布和类别动态变化的挑战, 即空间和时间范围内的数据异构问题。为解决这一问题, 本文构建了一个数字孪生背景下的联邦类增量学习整体框架, 并提出了一种混合知识蒸馏辅助的联邦类增量学习方法。具体来说, 与传统联邦学习本地更新方式不同, 本文方法通过自适应语义蒸馏损失和自适应注意力蒸馏损失集成的混合知识蒸馏方法提取旧全局模型中输出层的软标签语义知识和中间层的高维特征知识, 使客户端模型在拟合新数据的同时有效减少对旧数据的遗忘, 提升联邦类增量模型的性能。在相同的数据异构情况下, 与对比模型相比, 本文方法在 CIFAR100 数据集上精度提升 1.85%~2.56%, 在医学 CT 图像数据集 OrganAMNIST、OrganCMNIST、OrganSMNIST 上也取得了最优或次优的性能。
关键词: 数字孪生; 联邦类增量学习; 混合知识蒸馏; 数据异构; 图像分类; 灾难性遗忘; CT 图像; 联邦学习
中图分类号: TP399　　**文献标志码:** A　　**文章编号:** 1673−4785(2025)04−0905−11

中文引用格式: 张铭泉, 贾圆圆, 张荣华. 数字孪生中混合知识蒸馏辅助的异构联邦类增量学习 [J]. 智能系统学报, 2025, 20(4): 905–915.
英文引用格式: ZHANG Mingquan, JIA Yuanyuan, ZHANG Ronghua. Hybrid knowledge distillation-assisted heterogeneous federated class incremental learning for digital twins[J]. CAAI transactions on intelligent systems, 2025, 20(4): 905–915.

Hybrid knowledge distillation-assisted heterogeneous federated class incremental learning for digital twins

ZHANG Mingquan[1,2], JIA Yuanyuan[1], ZHANG Ronghua[1,3]
(1. Department of Computer, North China Electric Power University, Baoding 071003, China; 2. Hebei Key Laboratory of Knowledge Computing for Energy & Power, North China Electric Power University, Baoding 071003, China; 3. Engineering Research Center of Intelligent Computing for Complex Energy Systems, Ministry of Education, North China Electric Power University, Baoding 071003, China)

Abstract: In the context of digital twins, federated learning faces the challenge of identically nonindependent distribution data and dynamic changes of classes, which can be explained as the problem of data heterogeneity in the spatial and temporal scales. To solve this problem, this paper constructs an overall framework for federated class incremental learning for digital twins and proposes a federated class incremental learning method called hybrid knowledge distillation-assisted heterogeneous federated class incremental learning (FedKA). Specifically, different from the traditional federated learning approaches, FedKA employs a hybrid knowledge distillation method during the local update period. This method integrates adaptive semantic distillation loss with adaptive attention distillation loss. FedKA can distill the soft-labeled semantic knowledge in the output layer and the high-dimensional feature knowledge in the middle layer of the old global model. Consequently, the client model can effectively reduce the forgetfulness of the old data while fitting the new data and improve the performance of the federated class incremental model. Under the same data heterogeneity, the proposed FedKA method is utilized, and the accuracy on the CIFAR100 dataset remarkably increases from 1.85% to 2.56% compared with the SOTA model. Furthermore, FedKA achieves optimal or near-optimal performance on the medical CT image datasets, including OrganAMNIST, OrganCMNIST, and OrganSMNIST.
Keywords: digital twin; federated class incremental learning; hybrid knowledge distillation; data heterogeneity; image classification; catastrophic forgetting; CT images; federated learning

联邦学习是一种新兴的分布式机器学习框架, 以其“数据不动模型动, 数据可用不可见”的独特优势, 在保护数据隐私的同时打破数据孤岛, 将多方数据安全有效的连接起来, 力求多方联合训练一个性能好效率高安全可靠的全局模型。然而, 在实际应用中, 分布式数据往往具有

收稿日期: 2024−06−18.　**网络出版日期:** 2025−02−24.
基金项目: 中央高校基本科研业务费专项（2020MS122）.
通信作者: 张荣华. E-mail: zronghua88@aliyun.com.

异构性，即不同数据源的数据特征、分布和规模等存在差异。数据异构问题不仅使得理论分析变得困难，而且还降低了许多联邦学习算法的性能[1-4]。

数字孪生旨在将物理实体实时精细化建模为动态孪生体，物理实体和孪生体持续动态更新、迭代优化、动态实时交互，最终达到以虚控实的目的，即以虚拟孪生体的演变预测物理实体的未来状态和行为，从而指导现实世界的决策。而在数字孪生数字化建模过程中同样存在数据孤岛和数据异构问题，与联邦学习的结合是打破当前数字孪生瓶颈的有效方法之一。考虑到面向开放环境的数字孪生系统，任务的标记集合、特征空间、数据分布、学习目标可能随时间发生变化[5]。因此，在数字孪生背景下，数据异构是一个涵盖广阔空间维度与复杂时间变化的综合性挑战问题。目前，空间维度的数据异构问题是联邦学习的主要研究方向之一。例如 FedProx[6]、SCAFFOLD (stochastic controlled averaging for federated learning)[7]等算法，都是针对不同空间范围内客户端非独立同分布数据导致的性能下降问题对基础联邦学习算法 FedAvg[8] 的改进。时间维度上的数据异构问题，往往会随着对新数据的不断拟合导致对旧数据的遗忘，从而引起灾难性遗忘问题。由于开放环境中动态变化的因素多，问题较为复杂，本文主要研究客户端数据中新类别增量出现的情况，也就是异构情况下的联邦类增量问题。

在联邦类增量的设置下，本地客户端以流式的方式增量接收新类别的数据，而且收集新类别的客户端可以不定期地参加到整体的联邦训练。即不同客户端的数据不仅是非独立同分布的，数据类别是动态变化的，而且参与训练的客户端也是动态变化的。动态场景下，异构联邦增量学习不仅仅要解决由于单个客户端新类别的不断出现导致对旧类别的局部遗忘问题，还需要解决由于各个客户端非独立同分布数据引起的不同的遗忘速度，从而导致的全局遗忘问题。

为了解决上述问题，Dong 等[9] 提出的全局局部遗忘补偿（GLFC）算法，引入类感知梯度补偿损失和类语义关系蒸馏损失，防止局部遗忘，代理服务器用来选择最好的旧全局模型协助语义关系蒸馏。考虑到 GLFC 忽视了不同旧类别的遗忘速度不同，因此 Dong 等[10] 再次提出本地全局防遗忘（LGA）算法，提出类别平衡的梯度自适应补偿损失和类别梯度诱导的语义蒸馏损失，根据不同任务类别来自适应地重新加权梯度反向传播和语义关系蒸馏，从而平衡难以忘记和容易忘记的旧类别的不同遗忘速度。然而，GLFC 和 LGA 算法都是通过知识蒸馏来限制本地模型最后一个输出层的输出向全局模型最后一个输出层的输出靠拢，从而使本地模型的更新方向不偏离全局模型。虽然简单易懂，但是仅提取到全局模型中十分有限的标签语义知识，忽视了中间层蕴含的高维特征语义知识。

除此之外，这两种算法在设计和实施时并未充分考虑到数字孪生场景的特殊需求，因此在实际应用中可能存在局限性，比如数据更新策略、模型更新方法、通信开销、计算资源、隐私泄露等多方面限制。尽管现有工作已将数字孪生与联邦学习相结合[11-13]，但对关键的联邦类增量学习问题的研究仍显不足。为了提升算法的适应性和实用性，本文将数字孪生理念融入联邦类增量算法设计，提出了对应的模型更新策略，并探索其在智慧医疗中的应用，以实现更高效、更精准的数据分析和决策支持。

针对上述问题，本文研究数字孪生背景下的联邦类增量问题，首先给出了整体架构设计方案，其次提出联邦类增量学习方法（hybrid knowledge distillation-assisted heterogeneous federated class-incremental learning, FedKA），在本地更新阶段中引入了基于特征的注意力知识蒸馏的正则项，从全局模型中提取足量知识，并采用基于梯度的自适应系数调整不同类别数据的损失权重，从而在学习新知识的同时保留旧知识，有效防止遗忘。除此之外，针对医学图像分类领域的私密数据难以集中处理，数据分布类别等可能动态变化的现实情况，本文将联邦类增量用于医疗保健领域，可以在不影响隐私安全的情况下聚合来自不同联邦的通用信息，实现持续有效地进行全局模型的优化。最后，在 CIFAR100、OrganAMNIST、OrganCMNIST、OrganSMNIST 数据集上，通过实验验证了所提出算法的有效性和先进性。

1 相关工作

1.1 联邦学习

联邦学习这一概念由谷歌研究院于 2016 年提出[8]。FedAvg 是联邦学习最基础最具有代表的算法，核心思想是服务器通过将本地模型的参数进行平均来更新全局模型。然而当数据异构时，单纯的平均参数会导致权重发散、性能下降甚至根本无法收敛。为解决上述问题，很多基于 FedAvg 的变体被提出。变体可以分为两类，一类变体是通过规范局部模型与全局模型的偏差来进行

局部修改，比如 FedProx[6]、Scaffold[7]、FedNtd[14]。另一类则是服务器端的修改，改进聚合策略，从而提高服务器中本地模型聚合的效率。比如 Fedbe[15]、FedFTG[16]。尽管基于 FedAvg 的优化算法研究在联邦学习中占据重要地位，但并非本文的核心研究焦点。本文的主要研究内容聚焦于联邦学习过程中，客户端如何有效应对增量接收新类别数据的问题。为实现这一目标，本文采用了经典的 FedAvg 聚合策略作为基础框架。

1.2　类增量学习

目前，类增量学习方法可分为正则化、参数隔离、回放 3 类。

正则化方法通过添加显式正则项来约束模型参数的更新，比如基于参数重要性估计的弹性权值巩固（EWC）算法[17]、基于子空间投影的 OWM[18] 和 OGD（orthogonal gradient descene）[19] 算法、基于知识蒸馏的 LwF（learning without forgetting）[20]、LwM（learning without memorging）[21]、PODnet（pooled outputs distiuation）[22] 方法等。LwF[20]、iCaRL[23]、BiC[24] 算法主要是惩罚输出概率分布，具体来说，先得到旧模型在新任务上的预测值，在损失函数中引入新模型输出的蒸馏损失，然后用微调的方法在新任务上训练模型，从而避免新任务的训练过分调整旧模型的参数而导致新模型在旧任务上性能的下降。LwM[21] 算法则是惩罚重要特征变化，通过注意力损失使得新模型更关注重要的特征区域。基于正则化的方法实现简单，操作易于上手，它能够通过牺牲一定的计算能力来换取显著的性能提升，是一种高效且实用的优化策略。

参数隔离为每个任务指定不同的模型参数。比如 PackNet[25] 通过硬隔离新旧任务参数来处理学习任务。面对新任务时，它增量使用部分模型空间，并通过剪枝保留冗余空间。新任务的训练分为两步：固定旧任务参数进行训练并剪枝，在剩余空间重新训练。但这种方法限制了可处理的任务数量。PackNet 网络结构是固定的，而另一种基于参数隔离的方法（progressive neural networks，PNN）[26] 采取了动态的网络结构。参数隔离方法虽能有效管理不同任务的学习，但往往伴随着额外的参数和计算开销，这增加了学习新任务的成本。

回放方法在学习新任务时存储和重放先前的任务样本，比如 iCaRL[23] 和 BiC[24] 算法。iCaRL[23] 并未直接存储之前的任务样本，而是在训练新数据时为每个旧任务保留了一部分有代表性的旧数据，即越靠近类别特征均值的样本，从而更好地记忆模型在旧任务上学习到的数据特征。BiC[24] 在 iCaRL[23] 基础上使用线性偏移缓解新类和旧类之间由于样本不平衡导致的分类器偏移问题。相较于其他算法，回放的方法需要额外的计算资源和存储空间用于回忆旧知识，当任务种类不断增多时，要么训练成本会变高，要么代表样本的代表性会减弱，同时在实际生产环境中，这种方法还可能存在数据隐私泄露的问题[27]。

本文借鉴了 iCaRL 算法的核心思想，将距离类均值向量最近的样本保存在本地，在新类到达时，重新更新样例集。客户端本地更新过程中，同时使用样例集和本地数据进行训练。

1.3　联邦类增量学习

对联邦类增量学习中遗忘问题的处理比较复杂，不仅需应对单个客户端因新类别持续出现而对旧类别产生的局部遗忘问题，还需克服因各客户端数据非独立同分布特性所引发的遗忘速度差异，这种差异进一步导致了全局遗忘的复杂情况。为缓解联邦学习中的遗忘问题，众多学者从增量学习相关方法中得到灵感，将正则化、参数隔离、样本回放的方法加以改进进而应用到联邦学习中。

FedCurv[28] 在局部客户端采用正则化的方法，使用信息矩阵保护对旧任务重要的参数，限制其更新，从而实现局部防遗忘，该算法属于本地方法。然而，它需要在客户端之间计算和传达参数的重要性，这严重增加了学习过程的负担。CHFL[29] 使用参数隔离的思想，将每个客户端的网络分为公共网络和特有网络。其中，公共网络负责提取通用特征，特有网络负责提取特有特征。各个联邦参与方使用横向联邦的方法协同训练公共网络。特有网络则由局部训练更新。横向连接矩阵将两个网络连接，将旧任务的知识转移到新任务，从而防止灾难性遗忘问题。FLwF2T[30] 在 LwF[20] 的基础上，通过限制本地模型更新的方向，使其更趋向于本地旧模型和全局模型的方向，从而有效防止了局部遗忘和全局遗忘。但是当本地旧模型和全局模型差异较大时，可能会导致权重发散。FedWeIT[31] 是联邦类增量学习的代表性工作之一，在联邦学习过程中，为了抗局部遗忘，局部优化目标中加入了关于新类别和旧类别模型基本参数之间的差异正则项。为了抗全局遗忘，防止客户端之间互相干扰导致模型性能下降，使用了稀疏局部任务自适应参数，让每个模型有选择的更新全局任务共享参数，自适应的进行客户端间的知识转移。

虽然现有的算法一定程度上缓解了遗忘问题，但是有些需要用到一定的先验知识，比如新类出现的时间、位置等。GLFC[9] 和 LGA[10] 着重考虑了模型的类别预测信息和基于软标签的类别语义知识来进行本地更新，但是忽略了模型特征语义知识，未能最大限度的解决遗忘问题。本文在目前最先进的 LGA 算法的基础上，采用混合知识蒸馏的方式，提取旧模型中针对旧类的输出层的语义知识和中间层的高维特征知识，从而从旧全局模型中提取足量的知识，有效提升了算法性能。此外，本文综合考虑了数字孪生、类增量学习和联邦学习。据了解，现有工作大多数并没有综合考虑这种设置。

2 具体方法

本文方法主要分为两部分：1）基于数字孪生背景提出了一种联邦类增量学习算法的整体架构，将数字孪生、联邦学习、增量学习进行集成，为在分布式、隐私敏感的环境中进行增量学习提供了一种解决方案；2）基于混合知识蒸馏的联邦类增量学习方法，利用模型的软标签的语义信息和中间层的特征注意力图的知识，指导客户端本地更新，降低模型对旧类的遗忘速度，提升算法性能。

2.1 数字孪生背景下联邦类增量学习架构

本文提出的数字孪生背景下的联邦类增量学习算法整体架构如图 1 所示。

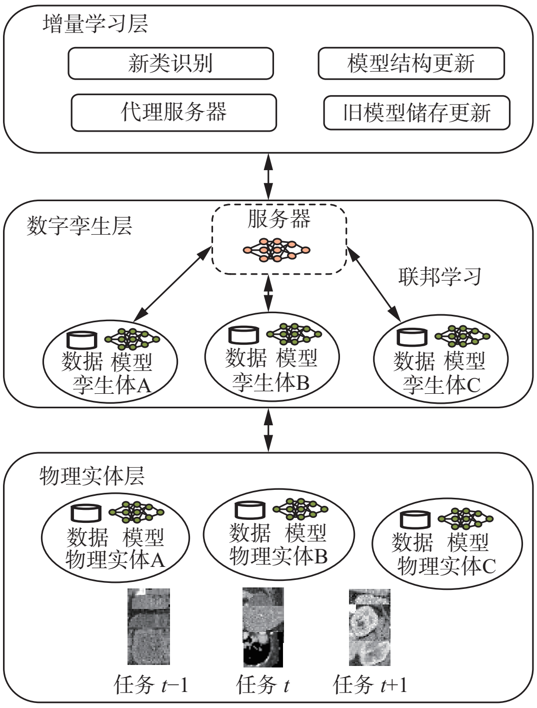

图 1　数字孪生背景下联邦类增量学习架构

Fig. 1　Architecture for federated class incremental learning in the context of digital twins

数字孪生背景下联邦类增量学习框架分为 3 层，分别为物理实体层、数字孪生层、增量学习层。

物理实体层中主要是众多物理实体，比如医院、银行等。每个物理实体会增量接收新类别数据。部署在物理实体上的传感器通过实时监测数据和运行状态能够动态更新模型。

数字孪生层是物理实体的模型和数据等对象的实时 1∶1 映射，在这一层中，服务器根据各个物理实体的孪生体进行联邦学习。联邦流程如下：1）模型初始化；2）每个孪生体使用本地数据训练模型，并将训练好的模型发送给服务器；3）服务器根据收到的模型，按一定的策略聚合为全局模型，并将其分发给各个联邦参与方；4）联邦参与方收到全局模型后，根据自身数据和全局模型迭代更新自己的模型，将更新后的模型发送给服务器，服务器再进行聚合、分发……直至达到收敛条件。

增量学习层主要负责新类识别、模型结构更新、代理服务器进行最佳旧模型选择、旧模型储存更新。由于不断接收新类，在数字孪生层中的联邦学习过程中需要增量学习层判断当前数据是否存在新类，如果存在新类，则需要调整模型结构，即改变输出层神经元个数，同时，储存当前旧模型，代理服务器从所有旧模型中挑选出最佳旧模型，各联邦参与方根据获得的最佳旧模型辅助模型更新。

两个物理实体的孪生体进行联邦类增量学习协同训练模型的大致流程如图 2 所示，主要分成 3 个部分。

1）服务器：负责将当前参与训练的客户端的本地模型聚合成全局模型。采用的是最基础的 FedAvg 聚合策略。

2）代理服务器：梯度解码网络负责将收到的梯度信息重建为扰乱的原型样本。扰乱的原型样本经过数据增强以后，根据旧全局模型的性能表现，从中选取最佳的全局旧模型，传输给客户端。

3）孪生体：每个客户端持续不断地接收流式数据，对应的实时动态更新的孪生体依据 Fed-KA 算法使用当前数据和收到的最佳全局旧模型训练本地模型。借鉴 iCaRL[23] 的做法，从本地数据选择最接近特征均值的样本作为原型样本，存储在本地。原型样本经过扰乱之后，送入梯度编码网络，将得到的梯度信息发送给代理服务器。

请注意，孪生体是参与联邦训练的实体，因此可以称之为客户端或参与方。若无特殊强调，下文的“客户端”即代指孪生体。

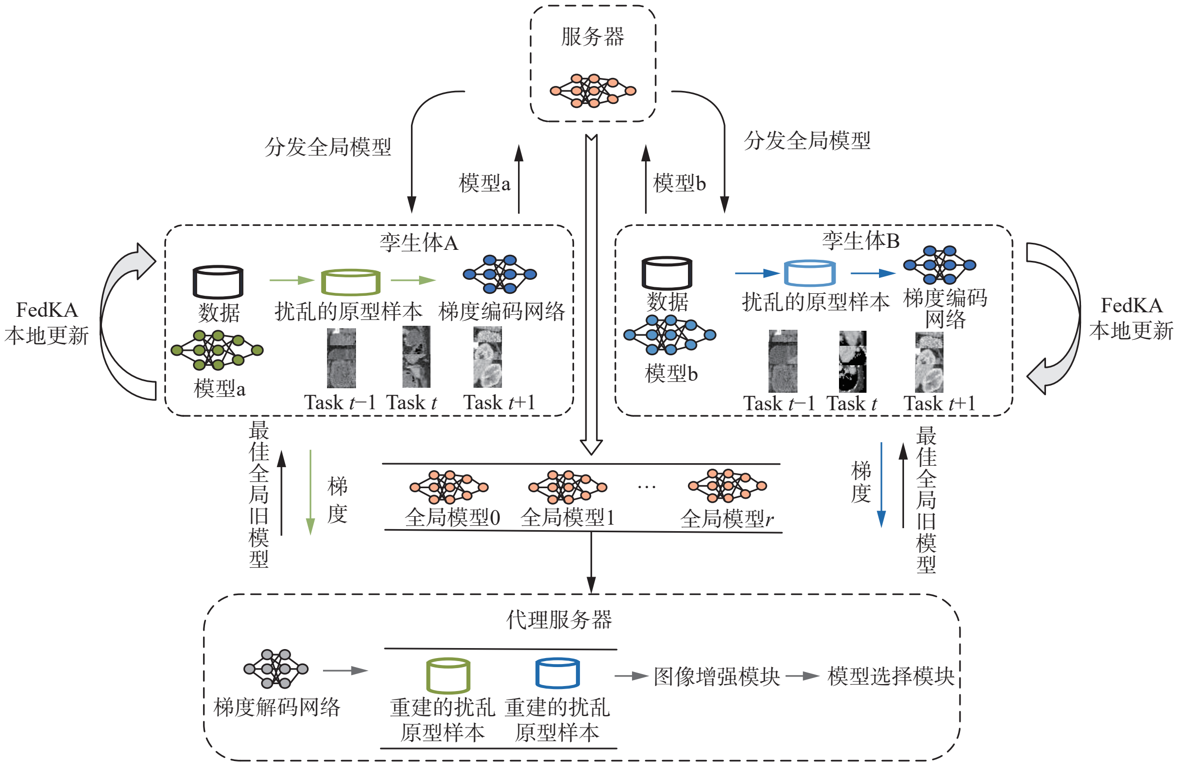

图 2　两个物理实体的孪生体进行联邦类增量学习协同训练模型的大致流程

Fig. 2　General process of federated class incremental learning model training for twins of two physical entities

2.2　FedKA: 基于混合知识蒸馏的联邦类增量方法

2.2.1　问题设置

联邦学习设置: 假设共有 L 个客户端, 可表示为 $\{S_l\}_{l=1}^{L}$, 每次从中随机挑选一部分, 在第 r(r=1, 2, ⋯, R)轮参与梯度聚合和模型更新, 其中 R 代表全局迭代次数。被挑选的客户端 S_l 根据本地数据构建本地模型 θ_l^r, 然后将梯度信息传输给中央服务器 S_g, S_g 聚合梯度后更新全局模型 θ^r, 并将其传输给选择的客户端, 客户端根据全局模型进行本地更新, 得到新的本地模型 θ_l^{r+1}。

联邦类增量学习设置: 每个客户端将持续不断的接收一系列的增量任务 $\{T^t\}_{t=1}^{T}$, 其中 T 表示增量任务的数目。对于第 t 个增量任务, 其中 x_i^t 和 y_i^t 分别表示第 i 张图像及其对应的标签, N^t 表示 T^t 中样本数目。假设每个增量任务中标签种类不重叠, 即 $\forall t \forall j$, $Y^t \cap Y^j = \varnothing (t \neq j)$, 其中 Y^t 和 Y^j 分别代表第 t 个和第 j 个增量任务的标签空间。

2.2.2　算法

在本节中, FedKA 示意如图 3 所示。伪代码如算法 1 所示。本文在 LGA[10] 模型的基础上, 引入了基于注意力图的特征知识蒸馏损失。在本地更新时, 除了必要的分类损失和基于标签的知识蒸馏损失, FedKA 还包括了特征注意力知识蒸馏损失, 并根据当前模型梯度设置了自适应系数, 动态调节每个类别的遗忘速度。在知识蒸馏过程中, 本文采用 KL 散度衡量输出层概率分布差异, L2 损失度量中间层特征差异。之所以采用这两种损失函数, 主要是根据不同损失函数的特性, 任务需求的匹配度以及实验验证的有效性。KL 散度其特性在于能够精准捕捉概率分布间的细微差别, 而且与 Softmax 函数兼容, 特别适用于处理分类任务中的输出层蒸馏。L2 损失简单有效, 能有效度量中间层特征差异。经实验验证, 两者结合能实现更高效、更精准的知识传递与模型性能提升。

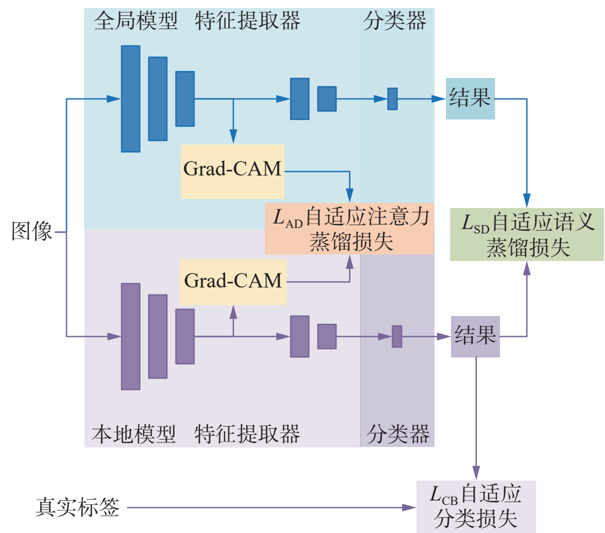

图 3　FedKA 示意

Fig. 3　FedKA diagram

算法 1 混合知识蒸馏辅助的联邦类增量学习方法 (FedKA)

输入 通信轮次 r, 当前增量任务 t, 本地更新轮数 E, L 个客户端 $\{S_1\}_{l=1}^{L}$, 数据集 $\{D_l\}_{l=1}^{L}$, 客户端样例集 $\{M_l\}_{l=1}^{L}$, 学习率 η, 权重参数 α、β、γ, 客户端 S_l 的最佳旧模型 θ_1^{t-1}

输出 全局模型 θ^{r+1}

for $\{S_l\}_{l=1}^{L}$ 中每一个客户端 S_l do

更新样例集 M_l

客户端随机抽样 $\{S_{l1}, S_{l2}, \cdots, S_{ln}\}$

for 每个被抽到的客户端 S_l do

for e=0 to e=E−1 do

for M_l 和 D_l 中样本 $\left\{x_l^{t,i}, y_l^{t,i}\right\}$do

$L_{\text{CB}} = L_{\text{CB}}(\theta_l^{r,t}, x_l^{t,i}, y_l^{t,i})$

$L_{\text{SD}} = L_{\text{SD}}(\theta_l^{r,t}, \theta_1^{t-1}, x_l^{t,i}, y_l^{t,i})$

$L_{\text{AD}} = L_{\text{AD}}(\theta_l^{r,t}, \theta_1^{t-1}, x_l^{t,i}, y_l^{t,i})$

$L = \alpha L_{\text{CB}} + \beta L_{\text{SD}} + \gamma L_{\text{AD}}$

$\theta_l^{r,t} = \theta_l^{r,t} - \eta \dfrac{\partial L}{\partial \theta_l^{r,t}}$

end for

end for

上传 $\theta_l^{r,t}$ 到服务器

end for

$\theta_l^{r+1} = \dfrac{1}{n} \sum\limits_{S_l \in \{S_{l1}, S_{l2}, \cdots, S_{ln}\}} \theta_l^{r,t}$

假设当前为第 t 个增量任务, 第 l 个客户端 S_l 接收到新类, r 表示当前训练全局轮次, 本地数据集 (包含所有采集到的数据和原型样本数据) 表示为 $D_l^t = \left\{x_l^{t,i}, y_l^{t,i}\right\}_{i=1}^{N_l^t}$, 其中 N_l^t 表示 S_l 本地数据总量, $x_l^{t,i}$ 和 $y_l^{t,i}$ 分别表示第 i 个图像和对应标签。

1)L_{CB} 自适应分类损失

在多分类任务中, 最常用的损失函数为

$$L_C = \frac{1}{B} \sum_{i=0}^{i=B} D_{\text{CE}} \left(y_i^{\text{pred}}, y_i^{\text{true}}\right)$$

式中: B 表示批大小, D_{CE} 表示标准交叉熵损失, y_i^{true} 和 y_i^{pred} 分别表示数据真实标签和当前模型的预测值。

假设当前客户端 S_l 的本地模型 $\theta_l^{r,t}$, 则在当前联邦环境中上述交叉熵损失可表示为

$$L_{\text{CE}} = \frac{1}{B} \sum_{i=0}^{i=B} D_{\text{CE}} \left(P(\theta_l^{r,t}, x_l^{t,i}), y_l^{t,i}\right)$$

式中 $P(\theta_l^{r,t}, x_l^{t,i})$ 表示数据 $x_l^{t,i}$ 通过本地模型 $\theta_l^{r,t}$ 的预测值。

考虑到每个类别的遗忘速度不同, 在基础的交叉熵损失中加入了自适应损失项进行加权。自适应项主要与当前模型的梯度相关。

假设当前网络模型的分类器, 即最后一层神经元 N_l 的输出为 $P(\theta_l^{r,t}, x_l^{t,i})$, 则最后一层神经元的梯度可表示为

$$\varDelta = \frac{\partial D_{\text{CE}} \left(P(\theta_l^{r,t}, x_l^{t,i}), y_l^{t,i}\right)}{\partial N_l} = P(\theta_l^{r,t}, x_l^{t,i}) - y_l^{t,i}$$

式中: $P(\theta_l^{r,t}, x_l^{t,i})$ 是样本 $x_l^{t,i}$ 在当前模型 $\theta_l^{r,t}$ 经过 softmax 激活函数处理后的预测值, $y_l^{t,i}$ 表示样本对应的真实标签的独热编码值。由于 $y_l^{t,i}$ 只有对应的真实类别的位置为 1, 其余为 0, 因此 $\varDelta$ 可以理解为当前模型的预测值在对应真实类别的位置减 1 处理后的值。假设 $y_l^{t,i}$ 只有第 m 个位置为 1, 则最后一层第 t 个神经元的梯度表示为

$$\varDelta_t = P(\theta_l^{r,t}, x_l^{t,i})_t - I_{t=m}$$

式中: $I_{(\cdot)}$ 是指示函数, $I_{\text{False}} = 0$, $I_{\text{True}} = 1$。

当 t=m 时,

$$\varDelta_m = P(\theta_l^{r,t}, x_l^{t,i})_m - 1$$

对于第 k 个增量任务中的小批量数据 $\left\{x_l^{t,i}, y_l^{t,i}\right\}_{i=1}^{B}$, 则最后一层神经元的梯度均值 ν_k 可表示为

$$\nu_k = \frac{1}{\sum\limits_{i=1}^{B} I_{y_l^{t,i} \in y_l^k}} \sum_{i=1}^{B} |\varDelta_m| \cdot I_{y_l^{t,i} \in y_l^k}$$

式中: m 为真实标签 $y_l^{t,i}$ 热编码位置为 1 的下标位置, y_l^k 表示客户端 S_l 在增量任务 k 中旧类的标签空间。

然而, 上述梯度均值限定在不同增量任务的遗忘速度, 对于类级别的遗忘速度并未做出限制, 为平衡不同类别的遗忘速度, 在上述 ν_k 基础上添加了与类别相关的指数项, 可表示为

$$\nu_k^n = \frac{1}{\sum\limits_{i=1}^{B} I_{y_l^{t,i} \in y_l^k}} \sum_{i=1}^{B} |\varDelta_m|^{\frac{C_l^o}{C_l^o + C_l^t}} \cdot I_{y_l^{t,i} \in y_l^k}$$

式中: C_l^o 表示客户端 S_l 的旧类别总数目, C_l^t 表示第 t 个增量任务中客户端 S_l 的新类别总数目。通过 ν_k 对交叉熵损失 L_{CE} 进行加权, 自适应分类损失 L_{CB} 可表示为

$$L_{\text{CB}} = \frac{1}{B} \sum_{i=0}^{i=B} \frac{|\varDelta_m|^{\frac{C_l^o}{C_l^o + C_l^t}}}{\sum\limits_{k=1}^{t} \nu_k^n \times I_{y_l^{t,i} \in y_l^k}} \cdot D_{\text{CE}} \left(P(\theta_l^{r,t}, x_l^{t,i}), y_l^{t,i}\right)$$

2)L_{SD} 自适应语义蒸馏损失

设客户端 S_l 的当前最佳全局旧模型为 θ_1^{t-1}, 样本数据 $x_l^{t,i}$ 通过最佳全局旧模型的输出为 $P(\theta_1^{t-1}, x_l^{t,i})$, 数据 $x_l^{t,i}$ 通过本地模型 $\theta_l^{r,t}$ 的预测值可表示为 $P(\theta_l^{r,t}, x_l^{t,i})$。

自适应语义蒸馏损失可表示为

$$L_{\mathrm{SD}}=\frac{1}{B}\sum_{i=0}^{i=B}\frac{|\varDelta_m|^{\frac{C_l^o}{C_l^o+C_l^t}}}{\sum_{k=1}^{t}\nu_k^n\times I_{y_l^{t,i}\in y_l^k}}\cdot$$

$$\sum_{k=1}^{t}D_{\mathrm{KL}}\left(P\left(\theta_1^{r-1},x_l^{t,i}\right)_{[C^{k-1}+1,C^k]}\|P\left(\theta_l^{r,t},x_l^{t,i}\right)_{[C^{k-1}+1,C^k]}\right)$$

式中: D_{KL} 表示 KL 散度, 下标 $[C^{k-1}+1,C^k]$ 来选择模型输出概率中增量任务 k 的新类别的概率值。

3) L_{AD} 自适应注意力蒸馏损失

梯度加权类激活映射(GradCAM)[32] 是深度学习中一种常用的可视化技术, 旨在深入解析和理解卷积神经网络(convolutional neural network, CNN)的决策机制。它通过网络梯度来解码每个特征图对特定类别的重要性, 有效地捕获了不同层级的特征激活信息及其与预测结果之间的关联。这一技术不仅将 CNN从黑盒模型转变为可解释性强的工具, 还提供了丰富的视觉信息, 以揭示网络在决策过程中真正关注的图像区域。GradCAM 可以用在知识蒸馏领域, 通过从教师模型中提取的注意力图将深层网络的特征知识传递给学生模型, 提示学生模型关注有助于网络预测的重要部分[33]。

在联邦类增量学习中, 本文新增了基于 GradCAM 的注意力蒸馏损失, 主要是计算当前全局最佳旧模型的注意力图和本地模型的注意力图向量之间的曼哈顿距离, 即 Ⅱ 范数。

样本数据 $x_l^{t,i}$ 通过最佳全局旧模型 θ_1^{r-1} 处理后, 中间层 layer 提取到的注意力图为

$$Q_{l,\mathrm{layer}}^{t-1,i}=\mathrm{vector}\left(\mathrm{GradCAM}\left(\theta_1^{r-1},x_l^{t,i}\right)\right)$$

数据 $x_l^{t,i}$ 通过本地模型 $\theta_l^{r,t}$ 处理后, 中间层 layer 提取到的注意力图可表示为

$$Q_{l,\mathrm{layer}}^{t,i}=\mathrm{vector}\left(\mathrm{GradCAM}\left(\theta_l^{r,t},x_l^{t,i}\right)\right)$$

设 $Q_{l,\mathrm{layer}}^{t-1,i}$ 和 $Q_{l,\mathrm{layer}}^{t,i}$ 的长度均为 q, 则注意力蒸馏损失为

$$L_Q=\sum_{j=1}^{q}\left\|\frac{Q_{l,\mathrm{layer}}^{t-1,i,j}}{\|Q_{l,\mathrm{layer}}^{t-1,i}\|_2}-\frac{Q_{l,\mathrm{layer}}^{t,i,j}}{\|Q_{l,\mathrm{layer}}^{t,i}\|_2}\right\|_1$$

式中: $\|Q_{l,\mathrm{layer}}^{t-1,i}\|_2$ 和 $\|Q_{l,\mathrm{layer}}^{t,i}\|_2$ 分别表示注意力图 $Q_{l,\mathrm{layer}}^{t-1,i}$ 和 $Q_{l,\mathrm{layer}}^{t,i}$ 的 L2 范数。

考虑到不同类别的数据遗忘速度不一致, 在 L_Q 的基础上又添加基于梯度的自适应权重, 得到自适应注意力蒸馏损失 L_{AD}:

$$L_{\mathrm{AD}}=\frac{1}{B}\sum_{i=0}^{i=B}\frac{|\varDelta_m|^{\frac{C_l^o}{C_l^o+C_l^t}}}{\sum_{k=1}^{t}\nu_k^n\times I_{y_l^{t,i}\in y_l^k}}\cdot L_Q$$

4) 客户端总优化目标

结合以上 3 个损失函数, 可以得到客户端总的优化目标为

$$L=\alpha L_{\mathrm{CB}}+\beta L_{\mathrm{SD}}+\gamma L_{\mathrm{AD}}$$

式中 α、β、γ 表示超参数。

3 实验评估

3.1 数据集介绍

CIFAR100 数据集[34]: 包含 100 个不同的类别, 每个类别都包含 600 张 32 像素×32 像素的彩色图像, 其中 500 张图像用于训练, 100 张图像用于测试。这些图像涵盖了各种各样的主题, 包括动物、植物、食物、交通工具等。该数据集被广泛用于评估图像分类算法的性能。

MedMNIST v2 数据集[35] 是一个包含多个医学公开数据集的集合, 包含 X 射线、OCT、超声、CT 等不同成像模式下不同病灶的数据, 主要用于医学图像分析研究。其包括 12 个 2D 数据集 (共 708 069 例, 28 像素×28 像素) 和 6 个 3D 数据集 (共 10 214 例, 28 像素×28 像素×28 像素)。在联邦类增量设置中, 客户端需要增量接收类别数据, 因此最好选择多分类任务进行实验。因此本文选择了基于 3D 计算机断层扫描 (CT) 的医学图像数据集 OrganAMNIST、OrganCMNIST 和 OrganSMNIST。

OrganAMNIST、OrganCMNIST、OrganSMNIST 分别是轴位面/冠状面/矢状面的基于肝脏肿瘤分割基准 (LiTS) 的三维腹部 CT 图像的采样切片。每个数据集中都包含 11 个类别, 分别代表 11 个器官。所有数据都是 28 像素×28 像素×1 像素的灰度图。OrganAMNIST 共有 58 850 张图像, 其中 34 581 张作为训练集, 6 491 张作为验证集, 17 778 张作为测试集。OrganCMNIST 共有 23 660 张图像, 其中 13 000 张作为训练集, 2 392 张作为验证集, 8 268 张作为测试集。OrganSMNIST 共有 25 221 张图像, 其中 13 940 张作为训练集, 2 452 张作为验证集, 8 829 张作为测试集。

3.2 实验设置

为保证实验结果的可复现性和分析, 实验相关环境配置如表 1 所示。

表 1　实验环境配置

Table 1　Experimental environment configuration

名称	配置信息
操作系统	Ubuntu 18.04
开发语言	Python 3.8.17
框架	PyTorch 1.7.0+cuda 10.2
CPU	Intel(R) Core(TM) i7-9700
GPU	GeForce RTX 2080 Ti
显存/GB	10

实验中采用了与 LGA[10] 相似的实验设置。具体的实验信息如下:

模型设置　实验中,全局服务器和每个客户端均采用 Resnet18 作为特征提取器。分类器将根据当前数据总类别动态调整输出层神经元个数。初始学习率设置成 2,所有客户端模型均采用 SGD 优化器。梯度编码网络和梯度解码网络采用了含有 3 个卷积层和一个线性层的 4 层 LeNet 网络模型。

联邦类增量设置　初始客户端设置为 30,每次增量任务中添加 5 个新的客户端。每轮全局训练中从现有客户端中随机选取 5 个进行训练,被选择的客户端根据 SGD 算法对本地模型进行迭代优化,一次 SGD 记为一次局部训练,总共进行 5 次局部训练。全局服务器进行聚合时采用 FedAvg 算法进行聚合。聚合一次全局模型算作一轮全局训练。每个增量任务全局训练 10 次。

数据设置　为模拟不同客户端的异构数据,实验中采取了对数据类别进行随机采样的思想。主要通过随机采样的个数来控制异构程度。比如,在 CIFAR100 数据集中,假设一次实验中每个增量任务新增类别数为 10,在异构程度 (iid) 为 0.6 的情况下,新的增量任务中,每个客户端将新增 6 个新类别。

参数设置　根据平均交叉熵判断客户端是否接收到新类,参照 LGA[10] 的做法,在 CIFAR100 数据集中,阈值设置为 1.2。大小 Batch Size 均设为 32,原型样本内存设置为 2000。随机数种子设置为 2021。本地训练的局部训练目标中 $\alpha=2$, $\beta=1$, $\gamma=1$。

3.3　CIFAR100 数据集对比实验

图 4 和图 5 分别给出了包括本文方法在内的 5 种算法在新增类别数取 10 和 5 的情况下准确率变化情况。表 2 和表 3 分别给出了 5 种算法在不同增量任务下的平均准确率对比。结果显示,在不同情况下 FedKA 算法均取得最优,相较于其他算法,精度达到了 1.85%~46.99%的提升。

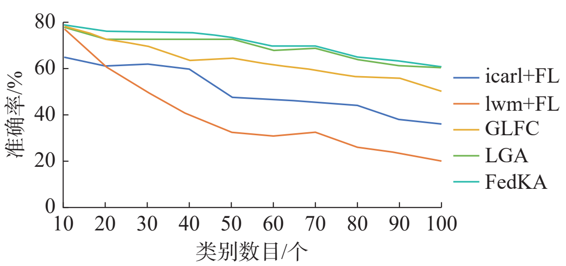

图 4　CIFAR100 数据集新增类别数为 10 时各算法准确率对比

Fig. 4　Comparison of accuracy among different algorithms on CIFAR100 dataset with task size set 10

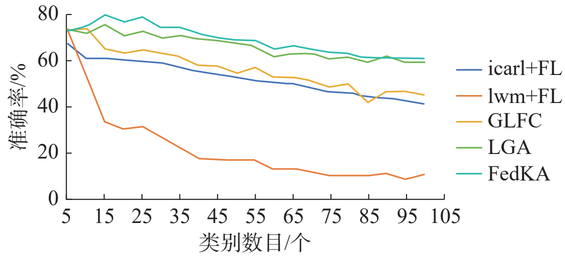

图 5　CIFAR100 数据集新增类别数为 5 时各算法准确率对比

Fig. 5　Comparison of accuracy among different algorithms on CIFAR100 dataset with task size set 5

表 2　CIFAR100 数据集新增类别数为 10 时各算法平均准确率对比

Table 2　Comparison of average accuracy of different algorithms on CIFAR100 dataset with task size set 10　%

方法	平均值	提升
icarl+FL	50.57	20.38
lwm+FL	39.29	31.66
GLFC	63.24	7.71
LGA	69.10	1.85
FedKA	70.95	—

表 3　CIFAR100 数据集新增类别数为 5 时各算法平均准确率对比

Table 3　Comparison of average accuracy of different algorithms on CIFAR100 dataset with task size set 5　%

方法	平均值	提升
icarl+FL	53.12	15.80
lwm+FL	21.94	46.99
GLFC	56.38	12.55
LGA	66.36	2.56
FedKA	68.93	—

3.4　MedMNIST 数据集对比实验

在数据集 OrganAMNIST、OrganCMNIST 和 OrganSMNIST 中,由于类别数目较少,只有 11 种,因此每个增量任务新增类别数设置为 3,异构程度为 0.67,即每个客户端新增 2 个类别,M 改成 1 000。另外,实验发现,平均交叉熵阈值 1.2 不再适用数据集,将其调整为 0.4。实验结果由表 4~9 所示。

表 4　OrganAMNIST 数据集上各个算法准确率对比

Table 4　Comparison of accuracy among different algorithms on OrganAMNIST dataset　%

方法	task 1	task 2	task 3	平均值	提升
icarl+FL	82.31	87.12	86.90	85.44	11.14
lwm+FL	75.45	53.02	50.34	59.60	36.98
GLFC	86.01	96.89	96.40	93.10	3.48
LGA	92.68	82.58	94.04	89.77	6.82
FedKA	98.84	93.77	97.14	96.58	—

表 5　OrganCMNIST 数据集上各个算法准确率对比
Table 5　Comparison of accuracy among different algorithms on OrganCMNIST dataset　%

方法	任务 1	任务 2	任务 3	平均值	提升
icarl+FL	63.82	73.44	90.04	75.77	12.98
lwm+FL	64.85	41.22	44.44	50.17	38.58
GLFC	76.09	94.32	96.71	89.04	−0.29
LGA	72.75	94.09	96.46	87.77	0.98
FedKA	73.26	95.68	97.31	88.75	—

表 6　OrganSMNIST 数据集上各个算法准确率对比
Table 6　Comparison of accuracy among different algorithms on OrganSMNIST dataset　%

方法	任务 1	任务 2	任务 3	平均值	提升
icarl+FL	64.38	75.55	79.75	73.23	6.42
lwm+FL	62.33	45.67	34.32	47.44	32.21
GLFC	72.61	69.85	59.21	67.22	12.43
LGA	73.39	71.57	83.26	76.07	3.58
FedKA	72.87	79.40	86.68	79.65	—

表 7　OrganAMNIST 数据集上各个算法 F1 score 对比
Table 7　Comparison of F1 score among different algorithms on OrganAMNIST dataset　%

方法	任务1	任务2	任务 3	平均值	提升
icarl+FL	58.13	74.54	86.55	73.07	23.49
lwm+FL	56.01	37.60	29.48	41.03	55.53
GLFC	83.59	97.03	96.37	92.33	4.23
LGA	92.50	82.53	91.41	88.81	7.75
FedKA	97.14	95.37	97.18	96.56	—

表 8　OrganCMNIST 数据集上各个算法 F1 score 对比
Table 8　Comparison of F1 score among different algorithms on OrganCMNIST dataset　%

方法	任务1	任务 2	任务3	平均值	提升
icarl+FL	32.89	51.60	70.74	51.74	30.84
lwm+FL	35.72	31.54	18.41	28.56	54.03
GLFC	64.20	93.95	95.68	84.61	−2.02
LGA	54.25	93.65	95.67	81.19	1.40
FedKA	55.26	95.51	96.99	82.59	—

表 9　OrganSMNIST 数据集上各个算法 F1 score 对比
Table 9　Comparison of F1 score among different algorithms on OrganSMNIST dataset　%

方法	task 1	task 2	task 3	平均值	提升
icarl+FL	31.89	52.71	59.60	48.07	20.20
lwm+FL	32.53	31.22	14.07	25.94	42.33
GLFC	54.36	59.20	43.66	52.41	15.86
LGA	65.13	58.84	70.49	64.82	3.45
FedKA	54.59	72.52	77.70	68.27	—

实验结果表明，本文方法与其他联邦类增量学习方法相比，在准确性和 F1 分数两个评价指标中均取得最优或者次优的表现。可见，本方法在解决医疗领域联邦类增量问题的优越性。

3.5　不同异构条件下 CIFAR100 对比实验

表 10 给出了包括本文方法在内的两种算法在 CIFAR100 数据集上异构程度分别为 0.4、0.5、0.6 和 0.7 的全局模型测试精度。从表 10 中可以看出，在不同异构程度下，相较于 LGA 算法，FedKA 算法的精度均有所提升，提升范围为 0.78%~3.01%。当异构程度越小，提升越明显。

表 10　CIFAR100 数据集在不同异构程度 ({0.4,0.5,0.6,0.7}) 下准确率对比
Table 10　Comparison accuracy of heterogeneous class distribution ({0.4,0.5,0.6,0.7}) on CIFAR100　%

异构程度	方法	10	20	30	40	50	60	70	80	90	100	平均值	提升
0.40	LGA	54.50	60.85	61.30	60.83	61.48	49.73	50.60	43.79	45.76	40.27	52.91	3.01
	FedKA	54.80	62.15	61.57	62.72	63.08	54.70	53.99	49.95	48.96	47.31	55.92	
0.50	LGA	64.10	66.55	69.80	66.43	63.24	67.20	63.31	61.30	60.23	57.12	63.93	2.26
	FedKA	64.20	70.50	72.33	71.22	66.02	68.23	64.50	63.24	62.98	58.66	66.19	
0.60	LGA	77.70	72.60	73.13	72.62	72.86	67.80	68.83	63.85	61.26	60.31	69.10	1.85
	FedKA	78.90	76.35	75.63	75.70	73.58	69.62	69.89	65.31	63.33	61.15	70.95	
0.70	LGA	86.90	80.20	78.77	74.12	73.02	70.32	66.90	63.9	64.03	61.17	71.93	0.78
	FedKA	87.40	82.80	79.73	74.72	72.96	70.90	68.09	64.18	64.74	61.56	72.71	

3.6　消融实验

为验证本文提出的混合知识蒸馏辅助的联邦类增量学习算法 FedKA 的有效性，本文对损失函数的 3 个组成部分在 CIFAR100 数据集上新增类别数取 10 场景下进行了充分的消融实验。其中，FedKA(w/oCBL) 表示将自适应分类损失 L_{CB} 改为交叉熵损失 L_{CE}，FedKA(w/oSDL) 表示损失函数 L 中去掉 L_{SD}，FedKA(w/oADL) 表示损失函数 L 中

去掉 L_{AD}，FedKA(w/oWADL) 表示将损失函数 L_{AD} 改为 L_Q。消融实验结果如表 11 所示。

表 11　CIFAR100 数据集新增类别数取 10 的消融实验
Table 11　Ablation experiments on the CIFAR100 dataset with task size set 10　%

方法	平均准确率	提升
FedKA(w/oCBL)	66.43	4.52
FedKA(w/oSDL)	66.15	4.80
FedKA(w/oADL)	68.11	2.84
FedKA(w/oWADL)	70.50	0.45
FedKA	70.95	—

可以发现，当损失函数 L 完整地包含其 3 个组成部分时，模型的性能将达到最优状态。自适应语义蒸馏损失和基于注意力图的自适应注意力蒸馏损失的混合蒸馏策略能够有效地提取教师模型的知识，减少在模型更新过程中对旧类的遗忘，提高面对联邦类增量学习环境中模型的性能。只有 3 个组成部分共同协作，才能确保模型在数据分散且类别动态增加的环境中能够稳定且高效地学习和优化。

4　结束语

在数字孪生技术的背景下，本文设计了一种联邦类增量学习框架，并创新性地提出了 FedKA 算法。本文深入探讨了数字孪生环境中联邦学习面临的数据异构问题，强调联邦增量学习本质上是应对时间维度上的数据异构导致的灾难性遗忘问题。FedKA 算法以知识蒸馏为核心思想，将损失函数分成 3 部分，自适应分类损失、基于软标签的自适应语义蒸馏损失、基于注意力图的自适应注意力蒸馏损失。这种混合知识蒸馏的策略使本地模型能够更有效地学习到旧全局模型的决策信息，从而显著提升了本地模型的泛化能力和性能表现。在 CIFAR100 和真实医学数据集 Organamnist、Organcmnist 和 Organsmnist 上的对比实验结果表明，相较于当前的联邦类增量学习算法，本文提出的 FedKA 算法能够取得最优或次优的表现，在医疗领域 CT 图像分类应用中也展现了其优越性。然而，本文实际上是考虑数字孪生受限环境，即只考虑动态类别数据流情形下的联邦类增量学习，没有考虑开放环境下的其他动态情形。未来我们将对这一问题进行更为深入和全面的研究，以推动联邦类增量学习在数字孪生领域的具体应用与发展。

参考文献：

[1]　张红艳, 张玉, 曹灿明. 一种解决数据异构问题的联邦学习方法[J]. 计算机应用研究, 2024, 41(3): 713−720.
ZHANG Hongyan, ZHANG Yu, CAO Canming. Effective method to solve problem of data heterogeneity in federated learning[J]. Application research of computers, 2024, 41(3): 713−720.

[2]　徐奕成, 戴超凡, 马武彬, 等. 基于粒子群优化的面向数据异构的联邦学习方法[J]. 计算机科学, 2024, 51(6): 391−398.
XU Yicheng, DAI Chaofan, MA Wubin, et al. Particle swarm optimization-based federated learning method for heterogeneous data[J]. Computer science, 2024, 51(6): 391−398.

[3]　王健宗, 张旭龙, 姜桂林, 等. 基于分层联邦框架的音频模型生成技术研究[J]. 智能系统学报, 2024, 19(5): 1331−1339.
WANG Jianzong, ZHANG Xulong, JIANG Guilin, et al. Research on audio model generation technology based on a hierarchical federated framework[J]. CAAI transactions on intelligent systems, 2024, 19(5): 1331−1339.

[4]　窦勇敢, 袁晓彤. 基于隐式随机梯度下降优化的联邦学习[J]. 智能系统学报, 2022, 17(3): 488−495.
DOU Yonggan, YUAN Xiaotong. Federated learning with implicit stochastic gradient descent optimization[J]. CAAI transactions on intelligent systems, 2022, 17(3): 488−495.

[5]　ZHOU Zhihua. Open-environment machine learning[J]. National science review, 2022, 9(8): nwac123.

[6]　LI Tian, SAHU A K, ZAHEER M, et al. Federated optimization in heterogeneous networks[EB/OL]. (2018−12−14)[2024−06−18]. https://arxiv.org/abs/1812.06127.

[7]　KARIMIREDDY S P, KALE S, MOHRI M, et al. SCAFFOLD: stochastic controlled averaging for federated learning[C]//Proceedings of the 37th International Conference on Machine Learning. New York: PMLR, 2020: 5132−5143.

[8]　MCMAHAN H B, MOORE E, RAMAGE D, et al. Communication-efficient learning of deep networks from decentralized data[EB/OL]. (2016−05−1)[2024−06−18]. https://arxiv.org/abs/1602.05629v4.

[9]　DONG Jianhua, WANG Lixu, FANG Zhen, et al. Federated class-incremental learning[EB/OL]. (2022−06−30)[2024−06−18]. https://github.com/conditionWang/FCIL/blob/main/README.md.

[10]　DONG Jiahua, LI Hongliu, CONG Yang, et al. No one left behind: real-world federated class-incremental learning[J]. IEEE transactions on pattern analysis and machine intelligence, 2024, 46(4): 2054−2070.

[11]　WANG Xiucheng, CHENG Nan, MA Longfei, et al. Digital twin-assisted knowledge distillation framework for heterogeneous federated learning[J]. China communications, 2023, 20(2): 61−78.

[12]　PANG Junjie, HUANG Yan, XIE Zhenzhen, et al. Collaborative city digital twin for the COVID-19 pandemic: a federated learning solution[J]. Tsinghua science and technology, 2021, 26(5): 759−771.

[13]　ZHAO Yunming, LI Li, LIU Ying, et al. Communication-efficient federated learning for digital twin systems of industrial Internet of Things[J]. IFAC-PapersOnLine, 2022, 55(2): 433−438.

[14] LEE G, JEONG M, SHIN Y, et al. Preservation of the global knowledge by not-true distillation in federated learning[EB/OL]. (2021−06−06)[2024−06−18]. https://arxiv.org/abs/2106.03097v5.

[15] CHEN Hongyou CHAO Weilun. Fedbe: making bayesian model ensemble applicable to federated learning[EB/OL]. (2020−09−04)[2024−06−18]. https://arxiv.org/abs/2009.01974.

[16] ZHANG Lin, SHEN Li, DING Liang, et al. Fine-tuning global model via data-free knowledge distillation for non-IID federated learning[C]//2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition. New Orleans: IEEE, 2022: 10164−10173.

[17] KIRKPATRICK J, PASCANU R, RABINOWITZ N, et al. Overcoming catastrophic forgetting in neural networks[J]. Proceedings of the national academy of science, 2017, 114(13): 3521−3526.

[18] ZENG Guanxiong, CHEN Yang, CUI Bo, et al. Continual learning of context-dependent processing in neural networks[J]. Nature machine intelligence, 2019, 1: 364−372.

[19] FARAJTABAR M, AZIZAN N, MOTT A, et al. Orthogonal gradient descent for continual learning[C]//Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics. New York: PMLR, 2020: 3762–3773.

[20] LI Zhizhong, HOIEM D. Learning without forgetting[J]. IEEE transactions on pattern analysis and machine intelligence, 2018, 40(12): 2935−2947.

[21] DHAR P, SINGH R V, PENG Kuanchuan, et al. Learning without memorizing[C]//2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach: IEEE, 2019: 5133−5141.

[22] DOUILLARD A, CORD M, OLLION C, et al. PODNet: pooled outputs distillation for small-tasks incremental learning[C]//Computer Vision-ECCV 2020. Cham: Springer International Publishing, 2020: 86−102.

[23] REBUFFI S A, KOLESNIKOV A, SPERL G, et al. iCaRL: incremental classifier and representation learning[C]//2017 IEEE Conference on Computer Vision and Pattern Recognition. Honolulu: IEEE, 2017: 5533−5542.

[24] WU Yue, CHEN Yinpeng, WANG Lijuan, et al. Large scale incremental learning[C]//2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach: IEEE, 2019: 374−382.

[25] MALLYA A, LAZEBNIK S. PackNet: adding multiple tasks to a single network by iterative pruning[C]//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018: 7765−7773.

[26] RUSU A A, RABINOWITZ N C, DESJARDINS G, et al. Progressive neural networks[EB/OL]. (2016−06−15)[2024−06−18]. https://arxiv.org/abs/1606.04671v4.

[27] 程虎威. 面向边缘计算资源受限场景的增量深度学习[D]. 北京: 北京交通大学, 2022.
CHENG Huwei. Incremental deep learning for marginal computing resource-constrained scenarios[D]. Beijing: Beijing Jiaotong University, 2022.

[28] SHOHAM N, AVIDOR T, KEREN A, et al. Overcoming forgetting in federated learning on Non-IID data[C]//2019 Workshop on Federated Learning for Data Privacy and Confidentiality. Vancouver: NeurIPS, 2019.

[29] MORI J, TERANISHI I, FURUKAWA R. Continual horizontal federated learning for heterogeneous data[C]//2022 International Joint Conference on Neural Networks. Padua: IEEE, 2022: 1−8.

[30] USMANOVA A, PORTET F, LALANDA P, et al. A distillation-based approach integrating continual learning and federated learning for pervasive services[EB/OL]. (2021−09−09)[2024−06−18]. https://arxiv.org/abs/2109.04197v1.

[31] YOON J, JEONG W, LEE G, et al. Federated continual learning with weighted inter-client transfer[C]//International Conference on Machine Learning. New York: PMLR, 2021: 12073−12086.

[32] SELVARAJU R R, COGSWELL M, DAS A, et al. Grad-CAM: visual explanations from deep networks via gradient-based localization[C]//2017 IEEE International Conference on Computer Vision. Venice: IEEE, 2017: 618−626.

[33] ZAGORUYKO S, KOMODAKIS N. Paying more attention to attention: improving the performance of convolutional neural networks via attention transfer[J]. (2016−12−12)[2024−06−18]. https://arxiv.org/abs/1612.03928.

[34] KRIZHEVSKY A, HINTON G. Learning multiple layers of features from tiny images[EB/OL]. (2012−05−18)[2024−06−18]. https://www.researchgate.net/publication/265748773_Learning_Multiple_Layers_of_Features_from_Tiny_Images.

[35] YANG Jiancheng, SHI Rui, WEI Donglai, et al. MedMNIST v2-A large-scale lightweight benchmark for 2D and 3D biomedical image classification[J]. Scientific data, 2023, 10(1): 41.

作者简介：

张铭泉，副教授，博士，主要研究方向为机器学习、模式识别和系统结构。发表学术论文 20 余篇。E-mail：mqzhang@ncepu.edu.cn

贾圆圆，硕士研究生，主要研究方向为联邦学习、数字孪生和增量学习。E-mail：jiayuans888@163.com

张荣华，高级工程师，主要研究方向为计算机图形学、3D AIGC 和数字孪生。E-mail：zronghua88@aliyun.com。