计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
[doi: 10.15888/j.cnki.csa.009843] [CSTR: 32024.14.csa.009843]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于解耦自蒸馏的个性化联邦学习算法①

闵和祥, 朱子奇

(武汉科技大学 计算机科学与技术学院, 武汉 430065)
通信作者: 闵和祥, E-mail: 1554768198@qq.com

摘　要: 联邦学习 (federated learning, FL) 是一种新兴的分布式机器学习框架, 旨在解决数据隐私保护和高效分布式计算的问题. 它允许多个客户端在不共享数据的前提下协同训练全局模型, 但由于各客户端的数据分布存在异质性, 单一的全局模型往往难以满足不同客户端的个性化需求. 针对这一问题, 本文提出了一种结合自蒸馏和解耦知识蒸馏的联邦学习算法, 该算法通过保留客户端的历史模型作为教师模型, 对客户端本地模型的训练进行蒸馏指导, 得到新的本地模型后上传到服务端进行加权平均聚合. 在知识蒸馏中, 通过对目标类知识和非目标类知识进行解耦蒸馏, 实现了对个性化知识的更充分传递. 实验结果表明, 本文提出的方法在 CIFAR-10 和 CIFAR-100 数据集上的分类准确率均超过了现有的联邦学习方法.

关键词: 联邦学习; 个性化学习; 知识蒸馏; 解耦知识蒸馏; 异质数据

引用格式: 闵和祥,朱子奇.基于解耦自蒸馏的个性化联邦学习算法.计算机系统应用. http://www.c-s-a.org.cn/1003-3254/9843.html

Personalized Federated Learning Algorithm Based on Decoupled Self-distillation

MIN He-Xiang, ZHU Zi-Qi

(School of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan 430065, China)

Abstract: Federated learning (FL) is an emerging distributed machine learning framework aimed at addressing issues of data privacy protection and efficient distributed computing. It allows multiple clients to collaboratively train a global model without sharing their data. However, due to the heterogeneity in the data distribution of each client, a single global model often fails to meet the personalized needs of different clients. To address this issue, this paper proposes a federated learning algorithm that combines self-distillation and decoupled knowledge distillation. The algorithm retains the client's historical model as a teacher model to distill and guide the training of the local model, and after obtaining a new local model, it is uploaded to the server for weighted averaging and aggregation. In the knowledge distillation process, the decoupled distillation of target class knowledge and non-target class knowledge allows for a more thorough transmission of personalized knowledge. Experimental results show that the proposed method outperforms existing federated learning methods in classification accuracy on the CIFAR-10 and CIFAR-100 datasets.

Key words: federated learning (FL); personalized learning; knowledge distillation (KD); decoupled knowledge distillation; heterogeneous data

随着分布式计算技术的迅速发展, 越来越多的应用需要在多个节点间进行高效协同计算. 但在传统的集中式数据处理模式下, 数据需要汇集到中心服务器进行统一处理, 这种模式不仅增加了数据传输的成本, 还带来了严重的隐私泄露风险. 特别是在涉及个人隐私或敏感信息的场景中, 例如医疗和金融领域, 用户的

① 基金项目: 公安部科技计划 (2022JSM08)
收稿时间: 2024-10-22; 修改时间: 2024-11-07; 采用时间: 2024-12-04; csa 在线出版时间: 2025-02-28

数据通常无法直接外传. 在这种背景下, 谷歌提出了联邦学习 (federated learning, FL)[1]作为一种保护用户隐私的新型分布式模型训练方法, 允许多个客户端在不共享本地数据的情况下协同训练出一个通用的全局模型, 目前在医疗保健[2]、智能交通[3]、自然语言处理[4,5]等领域得到了广泛应用.

传统联邦学习的目标是为所有客户端生成一个通用的全局模型, 服务端在每轮训练中随机选择部分客户端, 向其下发全局模型. 客户端在接收到全局模型后, 基于其本地数据集进行模型训练, 并将更新后的模型参数上传至服务端. 服务端根据设定的算法对各客户端上传的模型参数进行聚合, 生成新一轮的全局模型, 随后进入下一轮迭代. 然而在实际场景中, 不同客户端的数据分布往往具有显著的异质性 (Non-IID)[6,7], 即客户端的数据不服从同一分布, 这导致在服务端的每轮聚合过程中, 各客户端的特定数据特征出现不同程度地丢失, 即产生个性化遗忘问题[8–10], 使得全局模型在某些客户端上的表现不佳, 难以满足客户端的个性化需求.

为了解决这一问题, 本文提出了一种基于解耦自蒸馏的个性化联邦学习算法. 在该算法中, 客户端会存储上一轮的本地模型作为其个性化模型, 模型加权聚合后, 再通过知识蒸馏[11]将个性化模型中的知识传递给本轮的本地模型, 以缓解个性化遗忘问题. 同时, 在蒸馏过程中用更灵活的解耦知识蒸馏[12]代替传统知识蒸馏, 通过将目标类蒸馏和非目标类蒸馏进行解耦, 有效地从历史模型中提取个性化知识, 从而提升模型的个性化能力.

本文的主要贡献如下.

(1) 提出了一种基于解耦自蒸馏的个性化联邦学习算法, 对现有联邦学习框架做出改进, 通过保留客户端历史模型并将其作为“教师模型”, 有效缓解了传统方法中的个性化遗忘问题.

(2) 本文进一步在知识蒸馏中引入解耦机制, 通过目标类蒸馏帮助模型保持对目标类的难度感知能力, 同时通过非目标类蒸馏传递更广泛的类别区分信息, 优化个性化模型对本地数据特征的表达, 显著提升模型的个性化性能.

(3) 通过与多种现有方法的对比实验, 验证了本文方法在 CIFAR-10 和 CIFAR-100 数据集上具有更高的分类准确率, 尤其在数据分布异质性较大的情况下展现出显著优势.

1 相关工作

传统联邦学习算法的目标是构建一个适用于所有客户端的全局模型, 但由于各个客户端的数据分布存在显著差异, 导致全局模型难以适应每个客户端的特定需求. 解决数据异质性的联邦学习方法可以分为以下 3 类: 基于多任务学习的方法、基于元学习的方法、基于知识蒸馏的方法.

1.1 基于多任务学习的方法

多任务学习通过同时优化多个相关任务来提升模型性能. 在联邦学习中, 多任务学习视每个客户端的数据为独立的任务, 通过联合训练这些任务的方式来学习共享的表示. 基于这个框架, Smith 等人[13]提出了一种分布式优化方法, 采用交替优化的方式, 分阶段优化任务模型参数和任务关系矩阵, 通过为每个节点拟合独立的模型来处理非独立同分布的数据. 首先固定任务关系矩阵, 更新每个节点的模型参数, 然后在中央服务端固定模型参数, 优化任务关系矩阵. Mills 等人[14]提出了一种 MTFL 算法, 通过引入私有的 BN 层以应对数据异质性, 使每个客户端能够训练个性化的模型, 训练完成后, 客户端上传非私有的模型参数到服务端, 服务端对多个客户端的模型进行聚合.

1.2 基于元学习的方法

元学习方法[15]旨在学习一个“学习算法”, 使得模型能够通过少量的调整或学习迅速适应新的客户端数据分布. Jiang 等[16]在 FedAvg 算法的基础上, 结合了模型无关元学习 (MAML) 的微调阶段, 将联邦学习的训练过程视为元学习中的“元训练”, 将基于梯度下降的个性化过程视为“元测试”. 通过在每个客户端进行少量的本地更新来提高模型在数据异质性环境下的个性化性能. Fallah 等人[17]提出了 Per-FedAvg 算法, 通过一个元学习过程来找到适合所有客户端的初始模型, 然后每个客户端根据其本地数据进行微调, 对本地的数据执行一步或几步梯度下降来适应客户端的本地数据集.

1.3 基于知识蒸馏的方法

基于知识蒸馏的方法通过在客户端内部或客户端与服务端之间传递知识来增强模型的性能. 相比其他蒸馏方式, 基于 logit 的蒸馏在计算效率上更高, 因此成为大多数联邦蒸馏算法的首选方案. He 等人[18]提出

了 FedGKT, 在每个客户端使用本地数据训练模型, 生成特征图和 logit 并上传至服务端. 服务端基于这些信息更新服务端模型后将软标签返回给客户端, 通过这种双向蒸馏的方式应对数据异构问题. Jeong 等人[19]提出了一种联邦蒸馏方法, 通过将各个客户端的模型视为学生模型, 并利用其他所有客户端的平均模型输出作为教师模型的输出, 实现了分布式知识蒸馏. 相比共享模型参数, 通过交换模型输出来进行协同训练的方式降低了通信开销, 并保留了客户端对本地模型架构的自主设计权, 使得每个客户端能够根据自身数据特点优化模型. Jin 等人[20]提出了 pFedSD, 探讨了个性化遗忘的问题, 通过在客户端训练时引入个性化模型和本地模型之间的 *KL* 散度作为一种正则化策略来应对数据异质性, 以自蒸馏的形式缓解模型对历史个性化知识的遗忘. Lee 等人[21]提出了 FedNTD, 同样利用类似自蒸馏的概念来缓解数据异质性问题, 通过在本地训练时保留本地数据未出现的类别 (即非真实类别), 在训练损失中引入非真实蒸馏损失来防止知识遗忘. 本文同样利用自蒸馏机制以应对客户端数据异质性, 并针对目标类蒸馏和非目标类蒸馏进行解耦, 进一步加强个性化知识的传递.

2 本文方法

2.1 联邦自蒸馏

个性化联邦学习的核心问题在于如何找到个性化与泛化之间的平衡. 个性化的极端情况是没有与其他客户端的协作而完全依赖本地数据的独立训练. 而泛化的极端情况则是传统的联邦学习算法, 这类算法忽略了不同客户端之间的个体差异. 本文在服务端采用传统联邦学习中的加权平均策略对客户端上传的模型进行聚合, 以确保模型保持良好的泛化能力. 对于客户端的个性化遗忘问题, 使用知识蒸馏将历史模型的个性化知识传递给下一轮的本地模型. 图 1 为本文工作流程图.

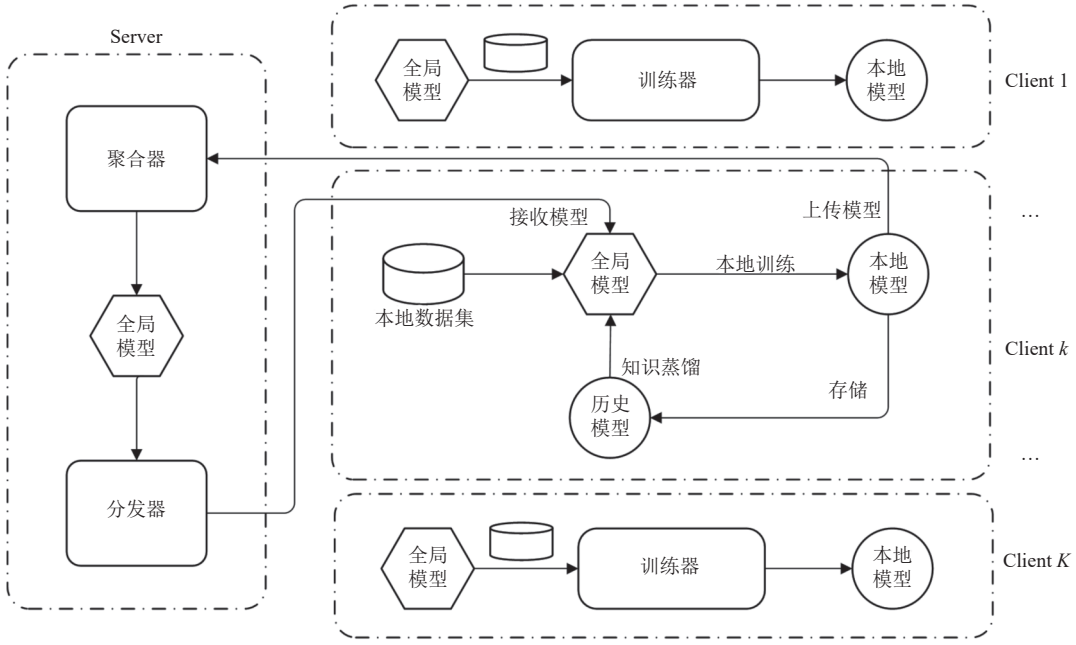

图 1 本文方法工作流程

如图 1 所示, 算法工作流程基于 Server-Client 框架, 服务端与多个客户端协同训练, 其中 Client 1、Client *k* 和 Client *K* 表示 *K* 个并行的客户端, 它们拥有各自的本地数据集并在本地独立训练. 每轮开始时, 服务端通过聚合器对客户端上传的模型进行聚合, 随后使用分发器将聚合后的全局模型分发给参与训练的客户端. 客户端接收到全局模型后, 通过历史模型进行蒸馏指导其训练, 训练结束后客户端将更新后的模型存储在本地并上传至服务端, 进行下一轮迭代. 其中, 聚合器负责接收客户端上传的模型, 并使用加权平均策略对这些客户端模型进行聚合, 生成新的服务端模型, 即全局模型, 从而实现各客户端的协同优化. 分发器则将聚合后的全局模型通过广播的方式发送至每个客户端, 确保各客户端能够在新一轮中使用包含全局信息的模

型进行本地训练.

服务端算法如算法 1 所示, 服务端初始化一个初始全局模型, 随后系统运行 T 轮通信. 在每轮通信中, 服务端首先选择部分客户端参与本轮训练, 将全局模型w^t分发给该轮参与训练的客户端, 然后各客户端在本地数据集上独立训练, 得到更新后的模型w_k^t并上传回服务端, 最后服务端对各客户端上传的模型参数进行加权平均, 从而得到新的全局模型w^{t+1}.

算法 1. 服务端算法

1) 初始化全局模型w^0
2) for t in T, do:
3) 选择 n 个客户端参与本轮训练
4) 分发器将w^t给参与训练的客户端
5) for k in n, do:
6) $w_k^t \leftarrow client\_train(w^t)$
7) 上传w_k^t到服务端
8) end
9) 聚合器对 n 个客户端上传的模型进行聚合:

$$w^{t+1} \leftarrow \sum_{k=1}^{n} \frac{|D_k|}{\sum_{i=1}^{n}|D_i|} w_k^t$$

10) end

为了实现客户端个性化知识的保留, 每个客户端都会存储一个个性化模型v_k^t, 用于在本地训练过程中指导当前轮次的模型更新. 客户端算法如算法 2 所示, 客户端每轮首先接收来自服务端的全局模型作为本轮的本地模型, 训练若干轮后, 将得到的模型w_k^t作为下一轮的个性化模型v_k^{t+1}, 并将w_k^t发送至服务端. 客户端在训练中不仅要最小化自己的交叉熵损失, 还需要考虑个性化模型v_k^t的蒸馏损失:

$$\Phi_k(w_k^t) = L_{\mathrm{CE}}(w_k^t) + \lambda(t) L_{\mathrm{KD}}(v_k^t, w_k^t) \tag{1}$$

$$\lambda(t) = \min\left(\frac{t}{N}, 1\right) \cdot \lambda_{\max} \tag{2}$$

其中, L_{CE}是客户端的交叉熵损失, 用于当前本地任务的学习. L_{KD}表示个性化模型的预测与当前本地模型的预测之间的蒸馏损失, 通过捕捉并保持二者之间的知识相似度, 使学生模型能够有效继承个性化模型的特征偏好. $\lambda(t)$是 warmup 函数, N 轮后达到$\lambda_{\max}$, 它在训练初期使整体损失较小, 使模型在训练过程中更平滑地收敛. 该损失函数结合交叉熵损失和知识蒸馏损失, 在实现各客户端协同训练的同时, 有效保留了客户端的本地个性化知识, 提高了模型的个性化性能.

本地权重w_k将按照以下方式通过随机梯度下降方法进行更新:

$$w_k^{t+1} = w_k^t - \eta \nabla \Phi_k(w_k^t, v_k^t) \tag{3}$$

其中, η是学习率.

算法 2. 客户端算法

1) $w_k^t \leftarrow w^t$
2) 计算本地迭代次数$\left\lceil \frac{|D_k|}{b} \right\rceil$, b 是批次大小
3) for i in 本地训练轮数, do:
4) for j in 本地迭代次数, do:
5) $w_k^t \leftarrow w_k^t - \eta \nabla (L_{\mathrm{CE}}(w_k^t) + \lambda(t) L_{KD}(v_k^t, w_k^t))$
7) end
8) end
8) 存储个性化模型$v_k^{t+1} \leftarrow w_k^t$
9) 将w_k^t上传至服务端

2.2 基于解耦自蒸馏的个性化联邦学习

如前文所述, 本文在联邦学习中采用了基于传统 logit 的自蒸馏进行个性化知识传递, 使其更好地适应本地需求. 然而传统的 logit 蒸馏仅通过计算教师模型与学生模型输出之间的 KL 散度来实现分布的拟合, 并未关注 logit 中目标类和非目标类在蒸馏过程中的不同作用, 导致客户端个性化知识传递不够充分. 因此, 本文引入解耦机制, 分别赋予目标类蒸馏和非目标类蒸馏不同的权重系数, 为客户端提供一种更高效、灵活的知识传递方法.

假设训练样本共有 C 个类别, 每个样本的分类概率表示为$p = [p_1, p_2, \cdots, p_t, \cdots, p_C]$, 其中第 t 类是目标类别, 通过 Softmax 函数得到每个类别的概率.

$$p_i = \frac{\exp(s_i)}{\sum_{j=1}^{c} \exp(s_j)} \tag{4}$$

其中, s_i表示第 i 类的预测输出.

定义p_t和p_t'分别表示目标类概率和非目标类的概率, 同时定义$\hat{p} = [\hat{p}_1, \hat{p}_2, \cdots, \hat{p}_{t-1}, \hat{p}_{t+1}, \cdots, \hat{p}_C]$来表示不考虑第 t 类时, 各个非目标类别的独立模型.

$$\begin{cases} p_t = \dfrac{\exp(s_t)}{\sum\limits_{j=1}^{C} \exp\left(s_j\right)} \\ p_t' = \dfrac{\sum\limits_{k=1,k\neq t}^{C} \exp(s_k)}{\sum\limits_{j=1}^{C} \exp\left(s_j\right)} \\ \hat{p}_i = \dfrac{\exp(s_i)}{\sum\limits_{j=1,j\neq t}^{C} \exp\left(s_j\right)} \end{cases} \tag{5}$$

令 T 和 S 分别表示教师模型和学生模型, 使用 *KL* 散度作为损失函数, 其表示如下:

$$L_{\mathrm{KD}} = KL\left(p^{\mathrm{T}} \| p^{\mathrm{S}}\right) = p_t^{\mathrm{T}} \log\left(\frac{p_t^{\mathrm{T}}}{p_t^{\mathrm{S}}}\right) + \sum_{i=1, i\neq t}^{c} p_i^{\mathrm{T}} \log\left(\frac{p_i^{\mathrm{T}}}{p_i^{\mathrm{S}}}\right) \quad (6)$$

根据式 (4)、式 (5) $\hat{p}_i = p_i / p_{t'}$, 故将式 (6) 改写为:

$$\begin{aligned} L_{\mathrm{KD}} &= p_t^{\mathrm{T}} \log\left(\frac{p_t^{T}}{p_t^{S}}\right) + p_{t'}^{\mathrm{T}} \sum_{i=1, i\neq t'}^{C} \hat{p}_i^{\mathrm{T}} \left(\log\left(\frac{\hat{p}_i^{\mathrm{T}}}{\hat{p}_i^{\mathrm{S}}}\right) + \log\left(\frac{p_{t'}^{\mathrm{T}}}{p_{t'}^{\mathrm{S}}}\right)\right) \\ &= p_t^{\mathrm{T}} \log\left(\frac{p_t^{\mathrm{T}}}{p_t^{\mathrm{S}}}\right) + p_{t'}^{\mathrm{T}} \log\left(\frac{p_{t'}^{\mathrm{T}}}{p_{t'}^{\mathrm{S}}}\right) + p_{t'}^{\mathrm{T}} \sum_{i=1, i\neq t}^{C} \hat{p}_i^{\mathrm{T}} \log\left(\frac{\hat{p}_i^{\mathrm{T}}}{\hat{p}_i^{\mathrm{S}}}\right) \end{aligned} \quad (7)$$

式 (7) 可写为:

$$L_{\mathrm{KD}} = KL(t^{\mathrm{T}} \| t^{\mathrm{S}}) + (1 - p_t^{\mathrm{T}}) KL(\hat{p}_t^{\mathrm{T}} \| \hat{p}_t^{\mathrm{S}}) \quad (8)$$

由式 (6) 可知, KD 损失可以视为两项的加权和, 其中 $KL(t^{\mathrm{T}} \| t^{\mathrm{S}})$ 表示目标类在教师与学生之间的二元概率相似度, 称为目标类知识蒸馏 (*TCKD*), 而 $KL(\hat{p}_i^{\mathrm{T}} \| \hat{p}_i^{\mathrm{S}})$ 表示非目标类在教师与学生之间的概率相似度, 称为非目标类知识蒸馏 (*NCKD*). DKD[12]指出, *TCKD* 处理困难数据集时更加有效, 而 *NCKD* 则包含着模型的暗知识, 这些暗知识在 logit 蒸馏中起着关键作用. 当教师模型对目标类别的置信度较高时, *NCKD* 的权重会相应减小, 这样的耦合关系大大抑制了蒸馏的效果. 因此分别赋予二者新的系 α 和 β, 得到解耦知识蒸馏的公式:

$$L_{\mathrm{DKD}} = \alpha \cdot TCKD + \beta \cdot NCKD \quad (9)$$

完整的蒸馏过程如图 2 所示, 教师模型和学生模型对输入图片进行特征提取并输出对应的 logit. 随后将 logit 按目标类和非目标类进行分类, 分别计算目标类和非目标类的蒸馏损失, 最终对二者加权求和得到最终的蒸馏损失. 所以客户端的损失函数被优化为:

$$\Phi_k(w_k^t) = L_{\mathrm{CE}}(w_k^t) + \lambda(t) L_{\mathrm{DKD}}(v_k^t, w_k^t) \quad (10)$$

其中, DKD 损失是 *TCKD* 损失和 *NCKD* 损失的加权和. 在式 (1) 基础上引入解耦机制, 通过将目标类蒸馏和非目标类蒸馏分别进行优化, 使客户端模型在训练中不仅能够从目标类蒸馏中获取辨别难度的信息, 还能更充分地利用解耦后的非目标类蒸馏所包含的深层次知识, 从而学习到更多的本地数据特征, 以应对数据异质性的影响.

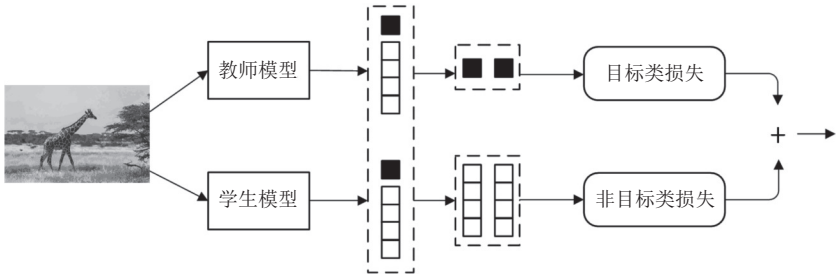

图 2 解耦知识蒸馏工作流程

3 实验数据分析

3.1 数据集介绍

本文在图像分类任务中评估所提出的方法, 使用 CIFAR-10 和 CIFAR-100 数据集[22]. CIFAR-10 数据集常用于小规模图像分类任务的测试, 包含 10 个类别、总计 60 000 张 32×32 像素的彩色图像, 其中训练集 50 000 张, 测试集 10 000 张. 由于其小尺寸和均匀的类别分布, CIFAR-10 已成为低分辨率图像分类的经典基准. 本实验中使用该数据集来评估模型在资源受限环境下的性能.

CIFAR-100 数据集则适用于更复杂的多类别任务评估, 包含 100 个类别的 60 000 张 32×32 像素彩色图像, 每个类别包含 500 张训练图像和 100 张测试图像. 实验中, 本文使用 CIFAR-100 来测试模型在处理数据稀缺和复杂类别结构时的表现. 通过模拟多种复杂的数据分布情况, 评估本文方法在细粒度分类任务中的能力, 进一步验证其相对其他联邦学习算法的优势和适用性.

本文采用 *Top*-1 *Accuracy*、计算成本和通信成本来评价方法的性能. 这 3 个指标能够全面、直观地反映算法在不同数据集上的性能, 便于与其他研究进行对比.

$$Top\text{-}1\ Accuracy = \frac{\text{正确预测的数量}}{\text{预测的总数量}} \times 100\% \quad (11)$$

3.2 实验参数

本文基于 PyTorch 深度学习框架, 使用 3 张 NVIDIA GeForce RTX 3090 显卡在 Ubuntu 18.04 平台下进行实验. 实验使用 5 层 CNN 作为基础网络结构, 设置了两种不同的 FL 场景:

(1) 20 个客户端, 参与率为 60%.

(2) 100 个客户端, 参与率为 10%.

每轮通信时, 每个客户端执行 50 个训练周期, 记录每个客户端的平均准确率.

本文使用 Dirichlet 分布$Dir(\alpha)$来为每个客户端创建不相交的异质性数据集. 超参数α控制标签分布的程度, 设置为 0.1. 较小的α值代表更多的 Non-IID 数据分布.

3.3 消融实验

为了研究自蒸馏和解耦知识蒸馏对个性化性能的影响, 本文使用 5 层卷积神经网络作为基础网络结构, 在 CIFAR-10 和 CIFAR-100 数据集上进行消融实验. 在消融实验中, 通过在客户端训练中分别使用传统联邦学习、联邦学习结合传统自蒸馏、联邦学习结合解耦自蒸馏进行消融分析, 评估自蒸馏机制和解耦机制在联邦学习中的作用. 实验结果如表 1 所示, 实验结果表明, 自蒸馏机制在训练过程中引导客户端利用个性化模型进行知识传递, 缓解了全局模型聚合过程中产生的个性化知识遗忘问题. 解耦机制通过区分目标类与非目标类的蒸馏, 深度挖掘了 logit 中的暗知识, 有效提升了联邦学习系统的性能.

3.4 对比实验

为了验证本文方法的有效性和先进性, 本文与多种现有的联邦学习方法对比实验, 这些方法可以分为:

(1) 传统的联邦学习, 如 FedAvg、FedProx.

(2) 针对数据异质性的改进联邦学习, 如 FedPer、pFedMe、FedFomo、pFedSD.

表 1 消融实验结果 (%)

方法	CIFAR-10	CIFAR-100
FL	41.58	30.11
FL+KD	80.53	57.01
FL+DKD	81.14	57.49

实验结果表明, 本文方法在准确率上相比这些方法取得了一定程度的提高. 如表 2 所示, 在参与率为 0.6 的实验中, 本文方法在 CIFAR-10 数据集上, 相比最优方法提高了 0.52 个百分点, 在 CIFAR-100 数据集上相比最优方法提高了 0.44 个百分点. 如表 3 所示, 当参与率降低至 0.1、客户端数量增加至 100 时, 本文方法在 CIFAR-10 数据集上, 相比最优方法提高了 0.32 个百分点, 在 CIFAR-100 数据集上相比最优方法提高了 0.26 个百分点. 实验结果验证了本文方法在数据异质性的场景下的有效性和鲁棒性.

表 2 20 个客户端、参与率为 0.6 的对比实验结果 (%)

数据集	FedAvg[1]	FedProx[6]	FedPer[23]	pFedMe[24]	FedFomo[25]	pFedSD[20]	本文方法
CIFAR-10	41.58	41.82	79.83	79.16	79.88	80.62	**81.14**
CIFAR-100	30.11	30.78	53.07	41.15	46.37	57.05	**57.49**

表 3 100 个客户端、参与率为 0.1 的对比实验结果 (%)

数据集	FedAvg[1]	FedProx[6]	FedPer[23]	pFedMe[24]	FedFomo[25]	pFedSD[20]	本文方法
CIFAR-10	39.41	37.93	72.77	68.37	68.73	73.31	**73.63**
CIFAR-100	23.72	13.17	43.89	25.17	24.70	47.39	**47.65**

如表 4 所示, 实验测量了客户端与服务端的单轮平均通信成本以及客户端单轮训练的计算成本. 通信成本包括下载和上传的总字节数, 对比其他方法, 本文方法没有增加额外开销. 计算成本通过 FLOPs 技术工具 pytorch-OpCounter 进行统计, 由于客户端需要重新计算个性化模型v_k的预测值, 本文方法每轮的计算成本相比有稍微提高, 但所需的通信轮次更少, 这部分的计算开销被通信效率的提升所弥补.

各对比实验结果表明, 本文方法相比传统联邦学习方案, 在缓解个性化遗忘问题方面具有显著贡献. 通过保留客户端历史模型并引入解耦知识蒸馏, 本文方法有效融合了全局模型和个性化模型的特征, 在客户端数据异质性场景下能够保持更高的分类准确率, 同时保持了较低的通信和计算成本, 在实际应用中更具实用性和适应性.

4 结束语

本文针对联邦学习中的客户端数据分布异质性问题, 提出了一种基于解耦自蒸馏的个性化联邦学习算法, 通过使用解耦知识蒸馏, 客户端能够从自身历史模

型中提取有价值的个性化知识, 在应对个性化需求和低开销的平衡上做出了重要贡献, 实验结果表明, 该方法在多个标准数据集上的性能表现优越, 在数据异质和资源受限的场景中具有较强的应用潜力.

表 4 CIFAR-10 下的通信开销和计算开销

方法	通信开销 (MB)	计算开销 (GFLOPs)
FedAvg[1]	3.76	1 132.75
FedProx[6]	3.76	1 132.75
FedPer[23]	3.70	1 132.75
pFedMe[24]	3.76	1 132.75
FedFomo[25]	9.16	1 248.86
pFedSD[20]	3.76	1 132.75
本文方法	**3.76**	**1 187.38 (4.82%)**

在未来的工作中, 可以探索更加灵活的个性化策略. 目前使用的静态解耦参数可能无法适应所有客户端的需求, 可以考虑根据各客户端的数据特征, 动态地调整目标类蒸馏和非目标类蒸馏的权重, 进一步提升模型在不同客户端上的适应能力.

参考文献

1 McMahan B, Moore E, Ramage D, *et al*. Communication-efficient learning of deep networks from decentralized data. Proceedings of the 20th International Conference on Artificial Intelligence and Statistics. Fort Lauderdale: PMLR, 2017. 1273–1282.

2 Sui DB, Chen YB, Zhao J, *et al*. FedED: Federated learning via ensemble distillation for medical relation extraction. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). ACL, 2020. 2118–2128.

3 Kaleem S, Sohail A, Tariq MU, *et al*. An improved big data analytics architecture using federated learning for IoT-enabled urban intelligent transportation systems. Sustainability, 2023, 15(21): 15333.

4 Liu M, Ho S, Wang MQ, *et al*. Federated learning meets natural language processing: A survey. arXiv:2107.12603, 2021.

5 Lin BY, He CY, Zeng ZH, *et al*. FedNLP: Benchmarking federated learning methods for natural language processing tasks. Findings of the Association for Computational Linguistics: NAACL 2022. Seattle: ACL, 2022. 157–175.

6 Zhao Y, Li M, Lai LZ, *et al*. Federated learning with non-IID data. arXiv:1806.00582, 2018.

7 Li T, Sahu AK, Zaheer M, *et al*. Federated optimization in heterogeneous networks. Proceedings of the 3rd Conference on Machine Learning and Systems. Austin, 2020. 429–450.

8 Tang XY, Guo S, Guo JC. Personalized federated learning with contextualized generalization. Proceedings of the 31st International Joint Conference on Artificial Intelligence. Vienna: IJCAI, 2022. 2241–2247.

9 Aljahdali M, Abdelmoniem AM, Canini M, *et al*. Flashback: Understanding and mitigating forgetting in federated learning. arXiv:2402.05558, 2024.

10 Xu YC, Ma WB, Dai CF, *et al*. Generalized federated learning via gradient norm-aware minimization and control variables. Mathematics, 2024, 12(17): 2644. [doi: 10.3390/math12172644]

11 Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network. arXiv:1503.02531, 2015.

12 Zhao BR, Cui Q, Song RJ, *et al*. Decoupled knowledge distillation. Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition. New Orleans: IEEE, 2022. 11943–11952.

13 Smith V, Chiang CK, Sanjabi M, *et al*. Federated multi-task learning. Proceedings of the 31st International Conference on Neural Information Processing Systems. Long Beach: Curran Associates Inc., 2017. 4427–4437.

14 Mills J, Hu J, Min GY. Multi-task federated learning for personalised deep neural networks in edge computing. IEEE Transactions on Parallel and Distributed Systems, 2022, 33(3): 630–641. [doi: 10.1109/TPDS.2021.3098467]

15 Finn C, Abbeel P, Levine S. Model-agnostic meta-learning for fast adaptation of deep networks. Proceedings of the 34th International Conference on Machine Learning. Sydney: PMLR, 2017. 1126–1135.

16 Jiang YH, Konečný J, Rush K, *et al*. Improving federated learning personalization via model agnostic meta learning. arXiv:1909.12488, 2019.

17 Fallah A, Mokhtari A, Ozdaglar A. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. Proceedings of the 34th International Conference on Neural Information Processing Systems. Vancouver: Curran Associates Inc., 2020. 300.

18 He CY, Annavaram M, Avestimehr S. Group knowledge transfer: Federated learning of large CNNs at the edge. Proceedings of the 34th International Conference on Neural Information Processing Systems. Vancouver: Curran Associates Inc., 2020. 1180.

19 Jeong E, Oh S, Kim H, *et al*. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-IID private data. arXiv:1811.11479,

2018.

20 Jin H, Bai DS, Yao DZ, *et al*. Personalized edge intelligence via federated self-knowledge distillation. IEEE Transactions on Parallel and Distributed Systems, 2023, 34(2): 567–580. [doi: 10.1109/TPDS.2022.3225185]

21 Lee G, Shin Y, Jeong M, *et al*. Preservation of the global knowledge by not-true self knowledge distillation in federated learning. arXiv:2106.03097, 2021.

22 Krizhevsky A, Hinton G. Learning multiple layers of features from tiny images. https://www.researchgate.net/publication/265748773_Learning_Multiple_Layers_of_Features_from_Tiny_Images. [2024-10-12].

23 Arivazhagan MG, Aggarwal V, Singh AK, *et al*. Federated learning with personalization layers. arXiv:1912.00818, 2019.

24 Dinh CT, Tran N, Nguyen J. Personalized federated learning with Moreau envelopes. Proceedings of the 34th International Conference on Neural Information Processing Systems. Vancouver: Curran Associates Inc., 2020. 1796.

25 Zhang M, Sapra K, Fidler S, *et al*. Personalized federated learning with first order model optimization. Proceedings of the 9th International Conference on Learning Representations. OpenReview.net, 2021.

(校对责编: 张重毅)