

多层超网络聚合的个性化联邦学习算法

李阳¹, 姜毅¹, 陈帅¹, 闫世超¹, 王磊², 马礼¹

(1. 北方工业大学信息学院, 北京 100144; 2. 河南省统计局数据处理中心, 河南 郑州 450018)

摘要: 个性化联邦学习(pFL)算法在处理非独立同分布(non-IID)数据集和客户端模型个性化方面有着巨大优势。而基于超网络的 pFL 利用客户端各自的超网络实现了客户端模型的个性化训练。然而, 针对客户端超网络参数和客户端数据的共享对客户端个性化模型准确率的影响, 提出多层超网络个性化联邦学习(pFedMHN)框架, 利用局部和全局超网络完成客户端超网络模型的加权聚合, 进而优化客户端模型。在服务器端学习全局超网络和每个客户端的多层局部超网络, 并加权聚合得到客户端超网络, 在客户端利用超网络参数迭代更新客户端模型, 超网络参数的共享实现了客户端更精确的个性化模型。实验结果表明, 在通用公开数据集上, pFedMHN 的准确率优于基准算法, 有效解决了在 non-IID 数据集上个性化联邦学习中数据异构性和模型准确性的问题, 利用超网络参数和客户端数据共享实现了客户端更精确的个性化模型。

关键词: 联邦学习; 联邦平均; 超网络; 个性化模型; 模型聚合

中图分类号: TP391

文献标志码: A

DOI: 10.19678/j.issn.1000-3428.0070679

Personalized Federated Learning Algorithm Based on Multi-Layer Hypernetwork Aggregation

LI Yang¹, JIANG Yi¹, CHEN Shuai¹, YAN Shichao¹, WANG Lei², MA Li¹

(1. School of Information, North China University of Technology, Beijing 100144, China;

2. Data Processing Center, Henan Province Bureau of Statistics, Zhengzhou 450018, Henan, China)

【Abstract】 Personalized Federated Learning (pFL) algorithms have significant advantages in handling non-Independent and Identically Distributed (non-IID) datasets and enabling client-side model personalization. Hypernetwork-based pFL utilizes the client's own hypernetwork to achieve a personalized client model. However, the effect of sharing client-side hypernetwork parameters and client-side data on the accuracy of client-side personalized models remains unclear. A personalized Federated learning with Multi-layer Hypernetwork (pFedMHN) framework is proposed to optimize client models through the weighted aggregation of local and global hypernetworks. The server learns a global hypernetwork and each client's multi-layer local hypernetworks and then aggregates them. Clients use these aggregated hypernetwork parameters to iteratively update their models, resulting in more accurate personalized models. The experimental results show that on general public datasets, the pFedMHN outperforms the four benchmark algorithms in terms of accuracy, effectively solving the problems of data heterogeneity and model accuracy faced during personalized federated learning on non-IID datasets and achieving a more accurate personalized model for clients by utilizing hypernetwork parameters and client data sharing.

【Key words】 federated learning; federated averaging; hypernetwork; personalization model; model aggregation

0 引言

随着智能设备的普及以及边缘计算的发展, 个性化联邦学习(pFL)成为了分布式学习领域重要的研究方向。pFL 机制允许每个客户端根据其本地数据分布训练一个专用的模型, 从而在保证数据隐私的同时提高模型在不同环境下的个性化能力。尽管 pFL 在实际应用中具有显著优势, 但在面对客

户端数据异构性和模型参数的协同优化时, 依然面临诸多挑战^[1-4]。

为了解决这些问题, 研究人员提出多种方法以促进个性化模型的生成。基于数据的方法, 即平滑客户端数据集之间的统计异质性^[5-6], 单模型方法, 例如正则化、元学习、参数解耦和多模型方式, 即为每个客户端训练个性化模型, 通过客户端模型的加权组合为每个客户端生成个性化模型。然而, 现有

基金项目: 北京市教育委员会科学研究计划(KM202410009003); 国家重点研发计划(2023YFC3107804, 2024YFE0200500); 北京市自然科学基金(4234083)。

作者简介: 李阳, 男, 讲师、博士, 主研方向为边缘计算、联邦学习、人工智能; 姜毅、陈帅、闫世超, 硕士研究生; 王磊, 高级工程师; 马礼(通信作者), 教授、博士。

收稿日期: 2024-12-04

修回日期: 2025-02-08

E-mail: mali@ncut.edu.cn

的 pFL 方法在整个模型参数或者不同客户端的损失值之间应用距离度量,这不足以利用它们的异构性,并可能导致组合权重不准确或贡献不平衡。为了进一步优化个性化联邦学习^[7],一些新兴的算法开始关注客户端之间的协作和聚类,如采用联邦注意力传递消息的 FedAMP 算法,使相似的客户端进行更多的协作。IFCA 算法^[8]采用迭代式聚类和簇类共享,服务器端在估计簇划分和最小化损失函数之间交替学习,并通过梯度下降优化客户端聚类的模型参数。CFL 算法^[9]是迭代式聚类和簇类不共享,簇的个数可以随着迭代次数变化,无需提前指定簇个数。FedRep 算法^[10]利用正则化方法促进相似客户端之间合作。但是上述工作通常需要在联邦学习过程中额外进行迭代聚类,通信开销和计算开销大,也没有考虑到层级之间的差异性。

在此背景下,基于超网络的方法逐渐成为一种新的研究方向。pFedHN^[11]利用超网络在服务器端直接生成一个客户端模型,每个客户端都有一个唯一的嵌入向量,作为输入传递给超网络以产生个性化的模型权重。pFedLA^[12]利用超网络与本地模型相结合最终为每个客户端训练个性化模型。然而这些都存在一些不足之处,特别是在如何有效融合局部知识和全局知识方面,当前方法未充分考虑到这两者之间的复杂关系,导致在一些场景下个性化效果的提升仍然有限。

本文提出一种基于多层超网络的个性化联邦学习(pFedMHN)算法,旨在利用局部超网络和全局超网络的局部知识和全局知识,以提升个性化模型的准确率。该框架可以实现联邦学习个性化的层级聚合,准确地识别客户端模型中每一层的效用以进行足够的个性化,从而提高非独立同分布(non-IID)模型的训练性能。在服务器端,为每个客户端引入三层局部超网络和一层全局超网络,在 pFL 训练过程中学习跨客户端的权重。服务器端学习全局超网络和每个客户端的多层局部超网络,并加权聚合得到客户端最终的超网络。客户端利用超网络参数迭代与客户端模型进行层级聚合,不断迭代更新客户端模型。利用超网络参数和客户端数据的共享实现了客户端更精确的个性化模型,并且更好地利用了局部知识和全局知识的共享,这被证明可以有效地促进对 non-IID 数据集的个性化学习,并且进行相应实验,证明了所提的 pFedMHN 在广泛使用的模型和数据集上可以达到更高的性能。本文的贡献主要如下:

1) 本文提出一个 pFedMHN 的训练框架,利用

三层局部超网络和一层全局超网络,并通过加权聚合的方式生成最终超网络。

2) 本文利用超网络参数和客户端嵌入向量的共享,证明其对模型准确率的提升是有效的,并且可以充分利用局部知识和全局知识进行层级聚合。

3) 本文对 4 种典型的图像分类任务进行相应的实验,结果表明,该算法在 non-IID 数据集上的准确率优于基准对比算法,可以得到更好的效果。

1 相关工作

1.1 联邦学习

XING 等^[13]提出一种基于 FedAvg 的新型联邦平均算法 N-FedAvg,旨在解决联邦学习中的随机选择客户端、静态学习率和模型压缩问题。该方法通过顺序选择客户端、余弦退火学习率和稀疏策略等改进了传统的 FedAvg 算法,提升模型准确性、降低损失函数值,并提高了通信效率和隐私保护能力。ZHANG 等^[14]提出一种新的联邦学习方法 CC-FedAvg,通过定制计算资源解决计算异构性挑战问题,理论和实验都验证了其有效性。ZHOU 等^[15]提出 Overlap-FedAvg,通过分离模型上传和下载阶段与训练阶段,减少通信干扰,加速联邦学习收敛速度。SUN 等^[16]提出 DFedAvgM 及其量化版本,在非凸假设下证明其理论收敛性。LAI 等^[17]提出一种轻量级联合学习方法,用于检测胸部 CT 图像中的 COVID-19,通过传输平均特征向量显著减少通信资源消耗。金彪等^[18]提出一种名为 GenFedRL 的联邦强化学习框架,通过模型共享技术保护智能物联网中深度强化学习智能体的私有数据,同时利用同步并行机制和 FedAvg 算法,实现不同网络结构智能体间的有效模型共享。虽然现有的 FedAvg 及其变种算法通过改进客户端选择、学习率调整和计算资源定制来提升联邦学习的效率和性能,但这些方法仍主要聚焦于整体模型的同步优化,缺乏针对个性化需求的深度优化。

1.2 个性化模型

ACHITUVE 等^[19]提出一种在模型中引入全局诱导点的方法,以增加客户端间的知识共享,使用高斯过程-伽马增强方法,并在联邦学习中应用诱导点的变种,实验验证了其在各种学习设置下的性能。MA 等^[12]提出层级个性化联邦学习 pFedLA 框架,利用非独立同分布数据中的客户端相似性生成个性化模型,并通过四个典型图像分类任务的实验证明其性能优于现有方法。YANG 等^[20]提出一种使用分层 Fisher 信息值的动态个性化策略,通过自适应

方法平衡客户端特征和全局知识,减少噪声影响。YANG 等^[21]提出特定于客户端的提示生成框架 pFedPG,通过交替优化个性化提示,提升异构客户端间的模型个性化。YUAN 等^[22]提出针对驾驶监控应用的联邦转移有序个性化学习 FedTOP 框架,解决了准确性降低、网络安全和通信资源不足等问题,实验结果表明,其在驾驶监控应用中显著提高了准确性和鲁棒性。朱素霞等^[23]提出一种基于相似度加速的自适应聚类联邦学习(ACFL)算法,通过客户端本地更新的几何特性和正向反馈实现自适应加速聚类,将数据分布相似的客户端划分到不同任务簇,从而提升模型性能。汤凌韬等^[24]提出一种隐私保护的数据增强方案,节点本地生成并共享虚拟样本,缓解因数据分布差异造成的模型偏移问题,结合生成对抗网络和差分隐私,设计隐私保护的样本生成和标签选取算法,确保生成的虚拟样本和标签满足差分隐私。上述方法虽然能够通过不同策略提升客户端模型的适应性,但大多依赖于有限的局部知识或通过局部聚合来优化模型个性化,缺乏对多层次知识的全面整合。

1.3 超网络

SHAMSIAN 等^[25]提出一种基于超网络的个性化联邦学习新方法,这种方法在以下方面具有更好的泛化能力,对于训练期间未出现的新客户端,对于具有不同计算资源的客户端,允许客户端拥有不同大小的模型。YANG 等^[26]提出一种 HyperFed 用于 CT 成像,HyperFed 由机构特定物理驱动超网络和全局共享成像网络两个模块组成,这是第一次基于 FL 的个性化 CT 成像,该框架灵活,可以更好地扩展到不同的 CT 任务,如后处理和重建。LI 等^[27]使用超网络在 Transformer 的自关注层生成投影矩阵,以生成客户端查询、键和值。然而上述方法都未充分利用局部知识和全局知识的共享,在处理复杂任务时可能面临计算和存储开销较大的问题。

2 算法设计

在本节中,先将个性化联邦学习问题形式化,然后展示了个性化联邦学习多层超网络方法。

2.1 问题描述

个性化联邦学习旨在多个客户端之间协作训练个性化模型,同时保持它们本地数据私密性。考虑到具有非独立同分布任务的 N 个客户端,对每个客户端分到的数据种类不同,假设 $\mathcal{D}_i = \{(x_j^{(i)}, y_j^{(i)})\}_{j=1}^{m_i} (1 \leq i \leq N)$ 是第 i 个客户端的数据集,其

中 x_j 是第 j 个输入数据样本, y_j 是对应的标签,第 i 个客户端上的数据集大小用 m_i 表示,设 θ_i 为客户端 i 的模型参数。pFedMHN 的优化问题可表示为:

$$\theta^* = \operatorname{argmin}_{\theta} \frac{1}{N} \sum_{i=1}^N \mathcal{L}_i(\theta_i) \quad (1)$$

$$\mathcal{L}_i(\theta_i) = \frac{1}{m_i} \sum_{j=1}^{m_i} \mathcal{L}_{\text{CE}}(\theta_i; x_j^{(i)}, y_j^{(i)}) \quad (2)$$

式中: $\theta = \{\theta_1, \dots, \theta_N\}$ 是所有客户端的个性化参数集; $\mathcal{L}_i$ 是第 i 个与数据集 $\mathcal{D}_i$ 相关的客户端损失函数; $\mathcal{L}_{\text{CE}}$ 是交叉熵损失函数。

2.2 pFedMHN 算法框架

在本节中,本文将介绍所提出的算法,该算法评估来自不同客户端每一层的重要性,以实现分层的个性化模型聚合。

pFedMHN 算法框架如图 1 所示,为每个客户端都应用了四层超网络。以往的 pFL 无论是在使用更新过程中 $\mathbf{v}_i$ 和 φ_i 的值都会改变的单层超网络,还是使用更新过程中客户端随机选择一个 $\mathbf{v}_i$ 作为嵌入向量,而 φ 的值共享单层超网络,这些都不足以充分利用全局知识和局部知识之间的共享。而本文算法采用的是三层局部超网络和一层全局共享的超网络。 $\text{HN}_i^l(\mathbf{v}_i; \varphi_i)$ 超网络层的网络在更新过程中 $\mathbf{v}_i$ 和 φ_i 值都会发生改变, $\mathbf{v}_i$ 和 φ_i 的值是客户端各自超网络的值,这一设计充分考虑了每个客户端本地数据的异构性,确保模型能根据客户端的个性化需求做出调整。 $\text{HN}_i^l(\mathbf{v}_i; \varphi)$ 超网络层的网络在更新过程中,客户端随机选择一个 $\mathbf{v}_i$ 作为嵌入向量,而 φ 的值则共享,这样可以促进客户端之间的合作,通过共享的参数加强类似客户端的协作,有助于提高相似数据分布客户端的聚合效果。 $\text{HN}_i^l(\mathbf{v}; \varphi_i)$ 超网络层的网络在更新过程中, $\mathbf{v}$ 的值是所有客户端嵌入向量的平均值,而 φ_i 的值为客户端自身的值,这一层通过汇聚所有客户端的知识,进一步提高知识的共享度,有助于在局部模型的基础上获得更全面的全局知识。 $\text{HN}_g^l(\mathbf{v}; \varphi)$ 超网络层的网络在更新过程中, $\mathbf{v}$ 和 φ 的值都是共享的,所有客户端用的是同一个值,全局超网络层提供了跨客户端的全局共享知识,确保所有客户端在一定程度上保持一致性,避免模型过度分散。

利用三层局部超网络和一层全局超网络加权聚合,为每一个客户端生成专用的超网络,并训练它们为每个不同客户端的每一个模型层生成聚合权重。将输出的每一个最终超网络层和各自客户端模型进行聚合并下发给客户端。客户端通过不断迭代得到

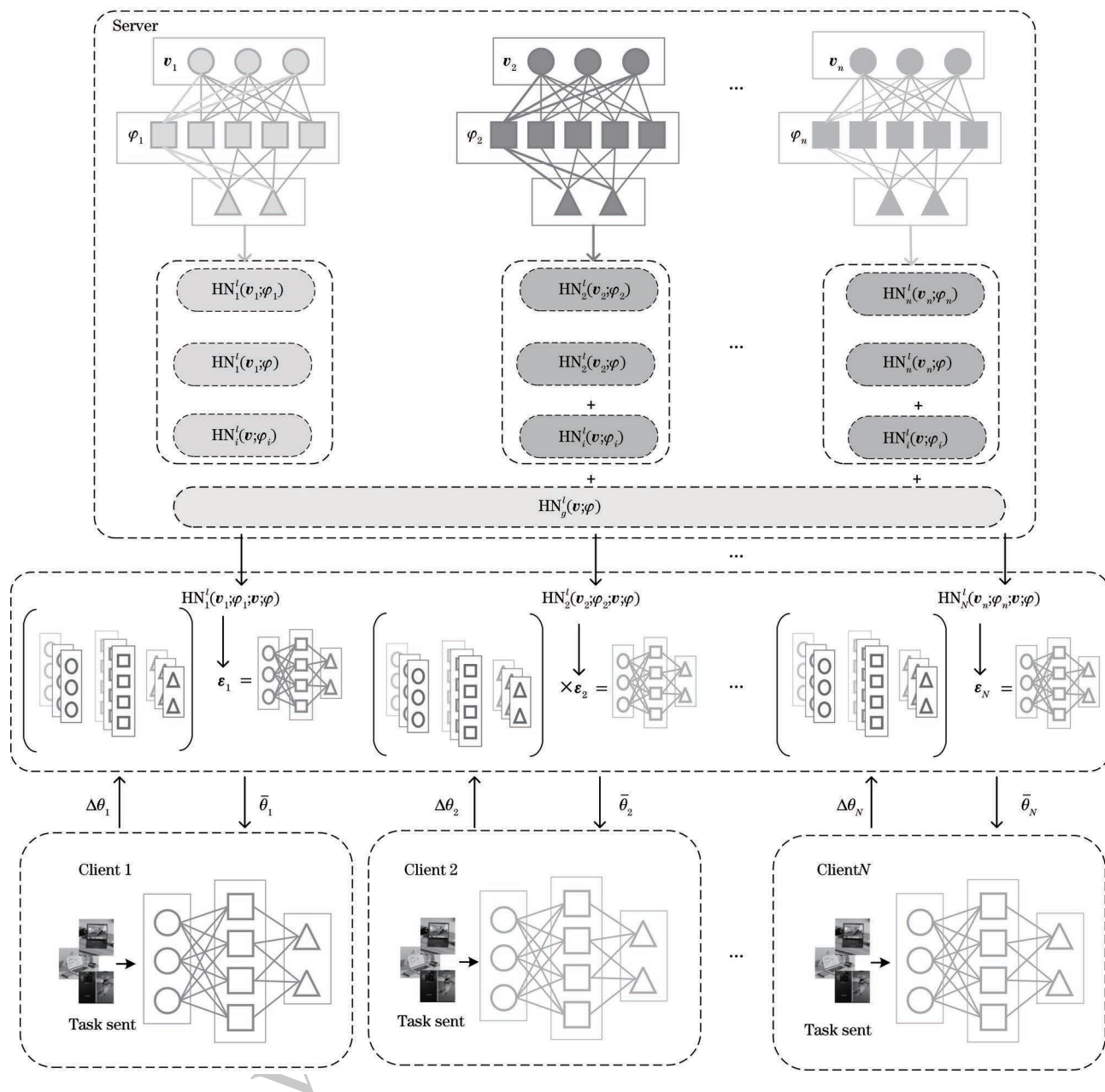


图 1 pFedMHN 算法的整体框架

Fig.1 Overall framework of pFedMHN algorithm

本地个性化模型,并返回模型的增量用于超网络更新。

该算法为每一个客户端维护了三层局部超网络,并且还维护了一层全局的超网络,充分利用局部知识和全局知识的共享,让具有相似数据分布的客户端具有较高的聚合权重,以加强彼此的相互贡献。本文选择三层局部超网络和一层全局超网络的架构,旨在通过平衡局部知识与全局知识的共享,实现高效的个性化模型聚合。客户端的数据分布通常具有显著的异质性,单层超网络难以充分捕捉这种异质性。因此,本文采用三层局部超网络为每个客户端提供个性化的模型聚合,而全局超网络则提供统一的全局知识,确保全局一致性。该设计有

效平衡了局部知识与全局知识的共享,提升了个性化模型的适应性和准确性。具体而言,三层局部超网络的层次化设计使得每一层能够处理不同维度的异质性:第一层局部超网络根据客户端独有的数据调整个性化参数;第二层通过共享部分超网络权重加强相似客户端间的协作;第三层通过汇聚所有客户端的嵌入向量进一步增强全局知识和局部知识共享。全局超网络层则确保全局一致性,提升模型的灵活性和适应能力。通过这种层次化设计,模型能够应对复杂的异质性需求,同时保持较强的表示能力。

本文算法在服务器端应用了一组聚合权重 ε_i ,以逐步利用层级别的用户间相似性, ε_i 定义为:

$$\mathbf{\epsilon}_i = [\mathbf{\epsilon}_i^{l1}, \mathbf{\epsilon}_i^{l2}, \dots, \mathbf{\epsilon}_i^{ln}] = \begin{bmatrix} \mathbf{\epsilon}_i^{l1,1} & \mathbf{\epsilon}_i^{l2,1} & \dots & \mathbf{\epsilon}_i^{ln,1} \\ \mathbf{\epsilon}_i^{l1,2} & \mathbf{\epsilon}_i^{l2,2} & \dots & \mathbf{\epsilon}_i^{ln,2} \\ \vdots & \vdots & & \vdots \\ \mathbf{\epsilon}_i^{l1,N} & \mathbf{\epsilon}_i^{l2,N} & \dots & \mathbf{\epsilon}_i^{ln,N} \end{bmatrix} \quad (3)$$

接下来,本文详细说明该算法的聚合权重矩阵 $\mathbf{\epsilon}_i$ 的更新过程 $\text{HN}_i^l(\mathbf{v}_i; \varphi) = \beta^1$, $\text{HN}_i^l(\mathbf{v}_i; \varphi_i) = \beta^2$, $\text{HN}_i^l(\mathbf{v}; \varphi_i) = \beta^3$, $\text{HN}_i^l(\mathbf{v}; \varphi) = \beta^4$ 。

设有 4 个字典 $\{\beta^1, \beta^2, \beta^3, \beta^4\}$, 每个字典包含相同的键集合 K 。对于每个键 $k \in K$, 其对应的值为向量 $\mathbf{\alpha}_k^i$ 。

对应每个字典 β^i 和键 k , 计算向量 $\mathbf{\alpha}_k^i$ 的 L2 范数, 如式(4)所示:

$$\|\mathbf{\alpha}_k^i\|_2 = \sqrt{\sum_j (\mathbf{\alpha}_{k,j}^i)^2} \quad (4)$$

式中: $\mathbf{\alpha}_{k,j}^i$ 表示向量 $\mathbf{\alpha}_k^i$ 的第 j 个分量。

计算所有字典中每个键 k 的 L2 范数总和, 如式(5)所示:

$$S_k = \sum_{i=1}^N \|\mathbf{\alpha}_k^i\|_2 \quad (5)$$

根据 L2 范数的比例对每个键 k 进行加权求和, 得到加权平均后的向量 $\bar{\mathbf{\alpha}}_k$:

$$\bar{\mathbf{\alpha}}_k = \sum_{i=1}^N \left(\mathbf{\alpha}_k^i \cdot \frac{\|\mathbf{\alpha}_k^i\|_2}{S_k} \right) \quad (6)$$

最终, 加权聚合的字典为:

$$\bar{\mathbf{\alpha}} = \{k: \bar{\mathbf{\alpha}}_k | k \in K\} \quad (7)$$

得到 $\mathbf{\epsilon}_i = \bar{\mathbf{\alpha}}$ 。算法中的聚合权重 $\mathbf{\epsilon}_i$ 也采用传统联邦平均方法。此处主要介绍基于 L2 范数的方法。

在超网络中, $\mathbf{v}$ 是嵌入向量, φ 是客户端 i 的超网络参数。设 $\{\theta^{l1}, \theta^{l2}, \dots, \theta^{ln}\}$ 是所有客户端在本地训练完成的中间模型参数集合, 其中每一个 $\theta^{ln} = \{\theta_1^{ln}, \theta_2^{ln}, \dots, \theta_N^{ln}\}$ 是所有客户端的第 n 层的集合, 其中 θ_N^{ln} 是客户端 N 在第 n 层的模型参数。在算法中, 客户端 i 的模型参数是通过加权聚合 $\mathbf{\epsilon}_i$:

$$\bar{\theta}_i = \{\bar{\theta}_i^{l1}, \bar{\theta}_i^{l2}, \dots, \bar{\theta}_i^{ln}\} = \{\theta^{l1}, \theta^{l2}, \dots, \theta^{ln}\} * \mathbf{\epsilon}_i \quad (8)$$

其中, $\bar{\theta}_i^{ln}$ 也可以表示为:

$$\bar{\theta}_i^{ln} = \sum_{j=1}^N \theta_j^{ln} \mathbf{\epsilon}_i^{ln,j} \quad (9)$$

$$\Delta \mathbf{v}_i = (\nabla_{\mathbf{v}_i} \bar{\theta}_i)^T \Delta \theta_i = [\{\theta^{l1}, \theta^{l2}, \dots, \theta^{ln}\} \times (\nabla_{\mathbf{v}_i} \text{HN}_i^l(\mathbf{v}_i; \varphi_i) + \nabla_{\mathbf{v}_i} \text{HN}_i^l(\mathbf{v}_i; \varphi))]^T \Delta \theta_i$$

$$\Delta \varphi_i = (\nabla_{\varphi_i} \bar{\theta}_i)^T \Delta \theta_i = [\{\theta^{l1}, \theta^{l2}, \dots, \theta^{ln}\} \times (\nabla_{\varphi_i} \text{HN}_i^l(\mathbf{v}_i; \varphi_i) + \nabla_{\varphi_i} \text{HN}_i^l(\mathbf{v}; \varphi))]^T \Delta \theta_i$$

$$\Delta \mathbf{v} = (\nabla_{\mathbf{v}} \bar{\theta}_i)^T \Delta \theta_i = [\{\theta^{l1}, \theta^{l2}, \dots, \theta^{ln}\} \times (\nabla_{\mathbf{v}} \text{HN}_i^l(\mathbf{v}; \varphi_i) + \nabla_{\mathbf{v}} \text{HN}_i^l(\mathbf{v}; \varphi))]^T \Delta \theta_i$$

$$\Delta \varphi = (\nabla_{\varphi} \bar{\theta}_i)^T \Delta \theta_i = [\{\theta^{l1}, \theta^{l2}, \dots, \theta^{ln}\} \times (\nabla_{\varphi} \text{HN}_i^l(\mathbf{v}_i; \varphi) + \nabla_{\varphi} \text{HN}_i^l(\mathbf{v}; \varphi))]^T \Delta \theta_i \quad (12)$$

2.3 pFedMHN 算法流程

基于多超网络的 pFedMHN 算法的输入为数

因此, 算法的目标函数可更新为:

$$\underset{\mathbf{V}, \varphi, \mathbf{v}; \varphi}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^N \mathcal{L}_i(\{\theta^{l1}, \theta^{l2}, \dots, \theta^{ln}\} * \text{HN}_i^l(\mathbf{V}; \varphi; \mathbf{v}; \varphi)) \quad (10)$$

式中: $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_N\}$ 是每一个客户端的嵌入向量; $\varphi = \{\varphi_1, \varphi_2, \dots, \varphi_N\}$ 是每个客户端的超网络参数; $\mathbf{v}$ 是全局的嵌入向量; φ 是全局的超网络参数。因此, pFedMHN 算法将客户端参数 θ_i 的优化问题转换成超网络的嵌入向量 $\mathbf{v}_i, \varphi_i, \mathbf{v}$ 和 φ 。嵌入向量和超网络参数更新规则为:

$$\begin{aligned} \nabla_{\mathbf{v}_i} \mathcal{L}_i &= (\nabla_{\mathbf{v}_i} \bar{\theta}_i)^T \nabla_{\theta_i} \mathcal{L}_i = \\ &[\{\theta^{l1}, \theta^{l2}, \dots, \theta^{ln}\} \times \nabla_{\mathbf{v}_i} \text{HN}_i^l(\mathbf{V}; \varphi; \mathbf{v}; \varphi)]^T \nabla_{\theta_i} \mathcal{L}_i \\ \nabla_{\varphi_i} \mathcal{L}_i &= (\nabla_{\varphi_i} \bar{\theta}_i)^T \nabla_{\theta_i} \mathcal{L}_i = \\ &[\{\theta^{l1}, \theta^{l2}, \dots, \theta^{ln}\} \times \nabla_{\varphi_i} \text{HN}_i^l(\mathbf{V}; \varphi; \mathbf{v}; \varphi)]^T \nabla_{\theta_i} \mathcal{L}_i \\ \nabla_{\mathbf{v}} \mathcal{L}_i &= (\nabla_{\mathbf{v}} \bar{\theta}_i)^T \nabla_{\theta_i} \mathcal{L}_i = \\ &[\{\theta^{l1}, \theta^{l2}, \dots, \theta^{ln}\} \times \nabla_{\mathbf{v}} \text{HN}_i^l(\mathbf{V}; \varphi; \mathbf{v}; \varphi)]^T \nabla_{\theta_i} \mathcal{L}_i \\ \nabla_{\varphi} \mathcal{L}_i &= (\nabla_{\varphi} \bar{\theta}_i)^T \nabla_{\theta_i} \mathcal{L}_i = \\ &[\{\theta^{l1}, \theta^{l2}, \dots, \theta^{ln}\} \times \nabla_{\varphi} \text{HN}_i^l(\mathbf{V}; \varphi; \mathbf{v}; \varphi)]^T \nabla_{\theta_i} \mathcal{L}_i \end{aligned} \quad (11)$$

上述公式描述了如何计算嵌入向量 $\mathbf{v}_i$ 关于客户端 i 的损失函数 $\mathcal{L}_i$ 的梯度 $\nabla_{\mathbf{v}_i} \mathcal{L}_i$ 以及如何计算超网络参数 φ_i 关于客户端 i 的损失函数 $\mathcal{L}_i$ 的梯度 $\nabla_{\varphi_i} \mathcal{L}_i$ 。pFedMHN 的超网络框架如图 2 所示。

针对客户端 i 所对应的超网络 $\text{HN}_i^l(\mathbf{v}_i; \varphi_i; \mathbf{v}; \varphi)$, $\mathbf{v}_i, \varphi_i, \mathbf{v}$ 和 φ 的更新方式如式(12)所示:

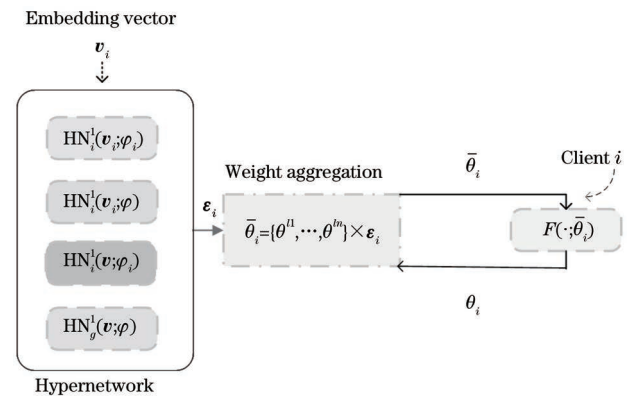


图 2 pFedMHN 的超网络框架

Fig.2 Framework of pFedMHN with hypernetwork

据集 $\{\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_N\}$, 客户端学习率 η , 超网络学习率 γ , 服务器端轮次 R , 客户端训练轮次 E , 初始化

模型参数,超网络参数,输出为训练后个性化模型 $\{\bar{\theta}_1, \bar{\theta}_2, \dots, \bar{\theta}_N\}$ 。服务器的计算步骤为:

- 1) 定义服务器端训练次数 R ;
- 2) 更新聚合超网络 $\text{HN}_i^l(\mathbf{V}; \psi; \mathbf{v}; \varphi)$;
- 3) 每个客户端的模型与最终超网络 $\text{HN}_i^l(\mathbf{V}; \psi;$

$\mathbf{v}; \varphi)$ 聚合得到模型 $\bar{\theta}_i^{(t+1)}$;

- 4) $\Delta\theta_i \leftarrow \text{ClientUpdate}(\bar{\theta}_i^{(t+1)})$;
- 5) 根据式(12)更新 $\mathbf{v}_i, \varphi_i, \mathbf{v}$ 和 φ 。

客户端的计算步骤为:

- 1) 接收服务器传送的模型;
- 2) 进行本地训练 E 次;
- 3) 返回模型之间增量 $\Delta\theta_i$ 。

首先,服务器初始化四层超网络模型,并进行加权聚合输出最终的超网络,再与客户端模型进行相应的层级聚合之后,将模型下发给各个客户端。客户端接收模型,并在本地进行迭代更新,给服务器返回模型增量。服务器接收到每个客户端的模型增量后,更新每个客户端对应的局部超网络和全局超网络。

2.4 pFedMHN 算法时间复杂度

表 1 定义了相关符号以及对应含义。

表 1 相关符号以及对应含义

Table 1 Related symbols and their corresponding meanings

参数	含义
N	客户端的数量
R	服务器端的训练轮数
E	客户端训练轮数
C	客户端本地每次训练的计算复杂度
B	客户端模型与超网络聚合的计算复杂度
A	更新聚合超网络的计算复杂度
U	更新参数 $\mathbf{v}_i, \varphi_i, \mathbf{v}$ 和 φ 的计算复杂度
P	客户端与服务器之间传输数据的复杂度

服务器端时间复杂度为 $O(A) + O(N \times B) + O(U) = O(A + N \times B + U)$, 其计算步骤:

- 1) 更新聚合超网络 $O(A)$;
- 2) 客户端模型与超网络聚合 $O(B)$;
- 3) 每个客户端的模型与最终超网络聚合得到模型 $O(N \times B)$;
- 4) 更新参数 $\mathbf{v}_i, \varphi_i, \mathbf{v}$ 和 φ 得到模型 $O(U)$ 。

客户端时间复杂度的计算步骤为:

- 1) 接收服务器传送的模型 $O(P)$;
- 2) 进行本地训练 E 次 $O(E \times C)$;
- 3) 返回模型之间增量 $O(P)$ 。

每个客户端的总时间复杂度为:

$$O(P) + O(E \times C) + O(P) = O(2P + E \times C) = O(E \times C + P)$$

因为有 N 个客户端,所以所有客户端的总时间复杂度为:

$$N \times O(E \times C + P) = O(N \times (E \times C + P))$$

总时间复杂度为:

$$R(O(A + N \times B + U) + O(N \times (E \times C + P))) = O(R \times A + R \times N \times B + R \times U + R \times N \times E \times C + R \times N \times P)$$

3 实验测试

3.1 实验设置

本文在 CIFAR-10^[28]、CIFAR-100^[29]、MNIST^[30] 和 FMNIST^[31] 4 个数据集上评估 pFedMHN 框架,训练客户端上所有数据集的分布是非独立同分布的。为了模拟不同 non-IID 程度的客户端数据,本文将 Dirichlet 分布函数的参数向量设置 $\alpha = 0.1$,该值越小说明异构性越大,所有数据都分为 70% 的训练集和 30% 的测试集。

对比算法: FedAvg 在服务器端进行模型加权聚合^[32]; pFedBN 保持每个客户端的批量归一化 (BN) 层在本地更新^[33]; pFedHN 是一种使用超网络直接产生个性化模型的 pFL 算法^[11]; pFedLA 是一种使用分层的个性化联邦学习算法^[12]; FedTP 是一种基于 Transformer 的新型联邦学习框架^[27]。

在实验中,使用与 pFedLA 相同的卷积神经网络 (CNN) 架构。在相同设计下,所有模型在不同客户端之间具有相同的结构。对于 CIFAR-10 和 CIFAR-100,在卷积层之后加了 BN 层。对于 MNIST 和 FMNIST,模型中没有 BN 层,用于计算逐层聚合权重的超网络是几个全连接层的简单结构。目标客户端每一层的权重由超网络中的相应全连接层计算。本文评估了 pFedLA 算法的性能,分别设置了 10 个与 20 个客户端以 100% 参与度开展全量训练。在 100 轮迭代 (MNIST 数据集设置为 50 轮迭代) 的训练中,统计并得到所有客户端的模型平均精度。对于所有实验,本研究使用批量大小为 32 的交叉熵损失和 SGD 优化器。10 个客户端案例的本地 Epoch 数为 10 轮。CIFAR-10 和 CIFAR-100 的学习率为 0.01, MNIST 和 FMNIST 的学习率为 0.005。

实验环境: 使用 RTX 3070 GPU, 3.6 GHz AMD 5500 和 32 GB RAM 在工作站上模拟所有客户端和服务器。

3.2 结果分析

1) 模型准确率分析。

在随机不同数据集(非独立同分布)上不同客户端下模型的平均准确率如表 2 所示,加粗表示最优数据,表 2 中数据格式为“平均准确率±标准差”,其中标准差反映了算法在多次独立实验中受随机因素(如客户端采样、数据划分、初始化等)影响导致的性能波动,数值越小表明算法越稳定。从表 2 可以看出,本文所提算法在 10 个客户端和 20 个客户端的实验中,均显示了相似的效果,此处将主要描述 10 个客户端的情况。对比传统的 FedAvg、FedBN,以及基于超网络的个性化算法,在不同数据集下所提模型的准确性基本都优于上述算法。在 CIFAR-10 数据集中,pFedMHN 算法得到的模型准确率为

61.21%,而基于 L2 范数的 pFedMHN2 算法得到的模型准确率为 60.31%。采用单个超网络 $HN_i^l(\mathbf{v}; \varphi_i)$ 的 pFedHO 算法得到的模型准确率为 57.83%。在 CIFAR-100 数据集中,pFedHO、pFedMHN 和 pFedMHN2 算法得到模型准确率分别为 58.46%、60.27% 和 58.59%,都高于 FedAvg、FedBN。在 MNIST 数据集中,pFedHO 算法得到的模型准确率为 99.41%,准确率最优,高于其他算法。在 FMNIST 数据集中,pFedHO、pFedMHN 和 pFedMHN2 算法得到的模型准确率为 93.79%、93.77% 和 94.06%,都高于 pFedHN、pFedLA、FedBN。实验结果表明,利用局部知识和全局知识的共享,能够为不同的客户端训练个性化模型。

表 2 在 CIFAR-10 和 CIFAR-100 数据集(non-IID)上不同客户端数下的平均准确率

Table 2 Average accuracy under different numbers of clients on CIFAR-10 and CIFAR-100 datasets(non-IID) %

算法	CIFAR-10		CIFAR-100		MNIST		FMNIST	
	客户端数	客户端数	客户端数	客户端数	客户端数	客户端数	客户端数	客户端数
	为 10 个	为 20 个	为 10 个	为 20 个	为 10 个	为 20 个	为 10 个	为 20 个
FedAvg	52.36±0.18	57.83±0.23	31.36±0.15	31.45±0.13	97.91±0.21	98.01±0.15	93.88±0.14	93.97±0.12
FedBN	52.35±0.23	57.67±0.28	31.39±0.23	31.45±0.19	97.52±0.29	97.58±0.31	80.83±0.21	80.91±0.19
FedTP	59.47±0.35	59.51±0.42	60.13±0.29	60.21±0.28	99.31±0.22	99.38±0.25	93.89±0.45	93.95±0.37
pFedHN	50.42±1.55	50.35±1.69	30.05±1.56	29.95±1.69	98.64±0.17	98.50±0.23	93.54±0.87	93.43±0.92
pFedLA	58.03±1.35	58.11±1.39	55.78±2.13	55.86±2.37	99.26±0.13	99.35±0.19	92.33±0.32	92.41±0.35
pFedHO	57.83±2.43	57.88±2.41	58.46±1.37	58.55±1.59	99.41±0.17	99.48±0.25	93.79±0.26	93.82±0.25
pFedMHN	61.21±1.77	61.32±1.68	60.27±1.33	61.37±1.41	99.21±0.08	99.29±0.12	93.77±0.72	93.81±0.75
pFedMHN2	60.31±1.93	61.43±1.88	58.59±1.25	58.65±1.35	99.22±0.07	99.31±0.13	94.06±0.59	94.10±0.61

图 3 是不同数据集下各算法的测试准确率。图 3(a)是在 CIFAR-10 数据集下的测试准确率,可以分析出本文提出的算法在准确率方面从一开始就高于对比算法,充分证明了利用局部知识和全局知识的共享,能够提升模型的准确率。图 3(b)是在 CIFAR-100 数据集下的测试准确率,本文所提算法在准确率方面都高于对比算法。图 3(c)是在 FMNIST 数据集下的测试准确率,本文所提算法的准确率高于对比算法。图 3(d)是在 MNIST 数据集下的测试准确率,所提算法在准确率优于对比算法。

2) 消融实验分析。

本节中,由于在 FMNIST 和 MNIST 数据集下,模型的准确率变化细微,因此在 CIFAR-10 和 CIFAR-100 数据集上进行消融实验,以评估模型中各个部分对其性能的影响。通过逐步移除不同的超网络,能够识别出哪些超网络对模型的最终性能至

关重要。在 CIFAR-10、CIFAR-100 数据集(非独立同分布)上,10 个客户端的平均模型准确率如表 3 所示。从表 3 可以看出,移除 $HN_i^l(\mathbf{v}; \varphi_i)$ 后的三层超网络,客户端模型的准确率与 pFedMHN 相比下降比较明显,在 CIFAR-10 数据集下,准确率平均下降了 11.66 百分点。实验结果表明, $HN_i^l(\mathbf{v}; \varphi_i)$ 在整个模型收敛以及准确率起到了至关重要的作用,并且该单层超网络在现有算法中没有使用过。随着超网络的层数减少至两层,相应的准确率也有所下降。移除了 $HN_i^l(\mathbf{v}; \varphi_i)$ 和 $HN_i^l(\mathbf{v}; \varphi_i)$ 之后的两层超网络,在客户端模型的准确率方面也低于其他两层超网络,这也同样验证了这四层超网络在模型中起到了一个互补的作用,通过互相分享局部知识,不仅可以加速不同客户端的收敛速度,还可以保留本地客户端的个性化,为客户端训练个性化的模型。实验结果验证了所提算法在模型准确率方面有着一定的优势。

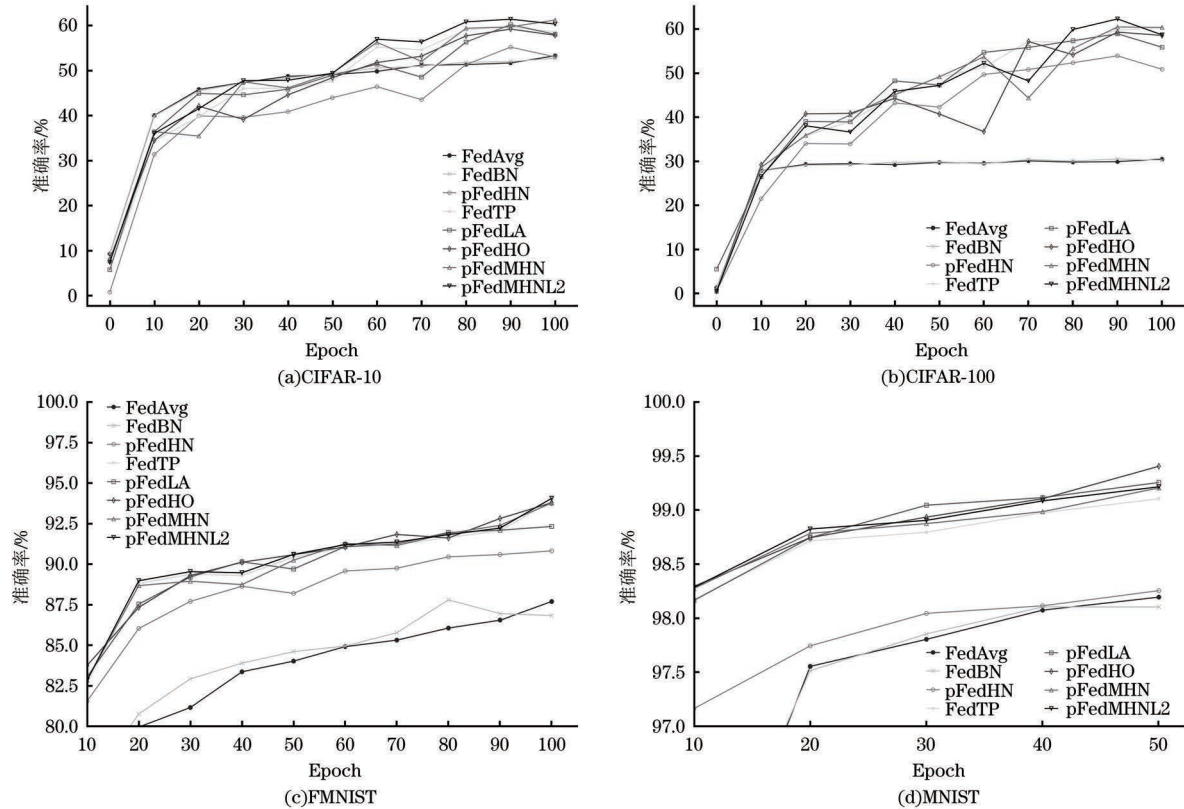


图 3 不同数据集下不同算法的测试准确率

Fig.3 The test accuracy among different algorithms under different datasets

表 3 在 CIFAR-10、CIFAR-100 数据集(non-IID)上的
平均模型准确率Table 3 The average model accuracy on the CIFAR-10 and
CIFAR-100 datasets (non-IID) %

移除的网络	准确率	
	CIFAR-10	CIFAR-100
$HN_i^l(v_i; \varphi)$	58.45 ± 1.18	56.27 ± 1.15
$HN_i^l(v_i; \varphi_i)$	57.35 ± 1.23	58.51 ± 1.23
$HN_i^l(v; \varphi_i)$	49.55 ± 1.65	48.57 ± 2.06
$HN_g^l(v; \varphi)$	55.88 ± 1.35	57.47 ± 1.13
$HN_i^l(v_i; \varphi) + HN_i^l(v_i; \varphi_i)$	56.78 ± 1.43	57.46 ± 1.87
$HN_i^l(v_i; \varphi) + HN_i^l(v; \varphi_i)$	50.24 ± 1.87	49.23 ± 1.83
$HN_i^l(v_i; \varphi) + HN_g^l(v; \varphi)$	57.26 ± 1.93	56.59 ± 1.75
$HN_i^l(v_i; \varphi_i) + HN_i^l(v; \varphi_i)$	49.53 ± 1.69	48.79 ± 1.73
$HN_i^l(v_i; \varphi_i) + HN_g^l(v; \varphi)$	49.21 ± 1.58	47.78 ± 1.64
$HN_i^l(v; \varphi_i) + HN_g^l(v; \varphi)$	55.93 ± 2.06	56.83 ± 1.86

3) 时间复杂度分析。

在本节中,针对 CIFAR-10、CIFAR-100、MNIST 和 FMNIST 等多个经典数据集上的各类算法进行时间消耗评估。在不同数据集下各算法在一轮中的时间消耗如图 4 所示。

从图 4 可以看出,本文提出的算法在追求更高准确率的同时,确实表现出了一定时间复杂度的增加。然而,这种增加幅度相对较小,在实际应用中

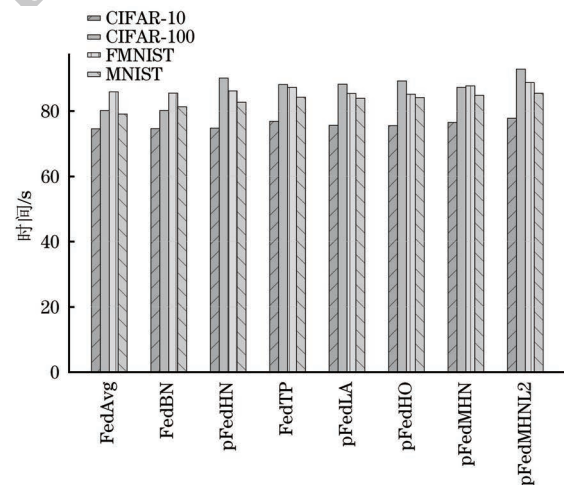


图 4 不同数据集下不同算法的时间消耗

Fig.4 Time consumption among different algorithms
under different datasets

合理接受。这种轻微的时间成本增加是为了在性能和准确率之间达成平衡,并对于提升算法整体效果具有显著意义。未来的研究可以尝试通过模型压缩、异步训练或更高效的参数更新策略来降低计算复杂度。此外,探索分布式计算和并行化技术,结合硬件加速(如 GPU 或 TPU),也有望能显著提升算法的运行效率,同时保持较高的准确率。

4) 模型存储大小分析。

在本节中,通过对不同算法的模型存储大小进

行详细评估, FedAvg、FedBN、pFedHN、FedTP、pFedLA 和 pFedMHN 算法的模型存储大小分别为 55 514.5、55 527.5、55 670.5、55 662.5、55 842.5 和 55 842.5 KB。不同算法的模型存储大小对比如图 5 所示。从图 5 可以看出, 本文所提算法与 5 种基准算法的模型存储需求相比略微增加。这种存储大小的略微增加反映了本文所提算法在处理某些特定任务或数据结构上具有更丰富的表达能力, 为了支持更复杂的模型更新和参数聚合的机制。

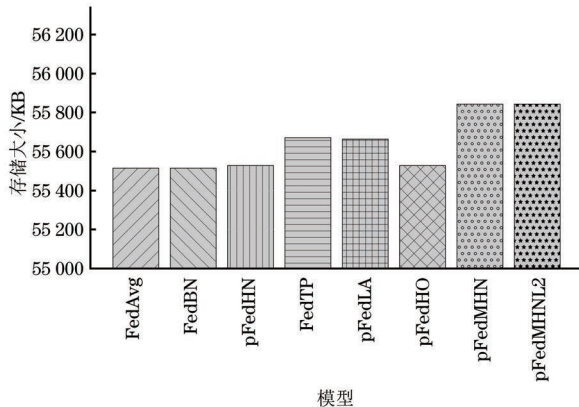


图 5 不同算法的模型存储大小对比

Fig. 5 Comparison of model storage sizes for different algorithms

未来的存储优化可以通过模型剪枝、参数量化等技术来减少存储需求, 从而降低整体的存储开销。同时, 采用增量式更新和分布式存储方法, 有望减少每轮训练过程中的存储负担, 并提高存储效率。

4 结束语

针对客户端超网络参数和客户端数据的共享对客户端个性化模型准确率的影响, 本文提出一种创新的个性化联邦学习训练框架。该框架通过结合三层局部超网络和一层全局超网络, 并通过加权聚合方式输出最终的超网络, 再与本地模型进行聚合以生成最终的本地模型, 可以充分利用局部知识和全局知识的共享, 为每个客户端训练个性化的模型。

实验结果表明, 充分利用局部和全局的知识进行逐层聚合的策略能够逐步增强相似客户端之间的协作, 并在 non-IID 数据集上生成更加个性化的模型, 其表现优于传统方法。此外, 还引入一种基于 L2 范数聚合的超网络, 在几乎不影响性能的情况下显著提高了准确率。通过对 4 种不同数据集进行评估, 在基于 CIFAR-10、CIFAR-100、FMNIST 和 MNIST 的 CNN 模型上验证了 pFedMHN 在提高模型性能方面的有效性。

尽管 pFedMHN 框架展示了出色的性能, 但仍存在一些局限性值得进一步研究。由于引入多层次的超网络和逐层聚合策略, 因此 pFedMHN 在计算和存储开销方面相对较高。下一步将在保证模型性能的前提下, 优化这些开销, 提升系统的效率, 通过对这些局限性进行改进和新研究方向的探索, pFedMHN 框架能推动个性化联邦学习的发展, 成为解决现实问题的有效工具。

参考文献

- [1] FALLAH A, MOKHTARI A, OZDAGLAR A. Personalized federated learning with theoretical guarantees: a model-agnostic meta-learning approach [J]. Advances in Neural Information Processing Systems, 2020, 33: 3557-3568.
- [2] 顾永根, 高凌轩, 吴小红, 等. 非独立同分布下联邦半监督学习的数据分享研究[J]. 计算机工程, 2024, 50(6): 188-196. GU Y G, GAO L X, WU X H, et al. Research on data sharing for federated semi-supervised learning with non-IID [J]. Computer Engineering, 2024, 50(6): 188-196. (in Chinese)
- [3] HUANG Y T, CHU L Y, ZHOU Z R, et al. Personalized cross-silo federated learning on non-IID data [EB/OL]. [2024-11-01]. <https://arxiv.org/pdf/2007.03797>.
- [4] LI T, HU S Y, BEIRAMI A, et al. Ditto: fair and robust federated learning through personalization [EB/OL]. [2024-11-01]. <https://arxiv.org/pdf/2012.04221>.
- [5] MCMAHAN H B, MOORE E, RAMAGE D, et al. Communication-efficient learning of deep networks from decentralized data [EB/OL]. [2024-11-01]. <https://arxiv.org/pdf/1602.05629>.
- [6] DUAN M M, LIU D, CHEN X Z, et al. Self-balancing federated learning with global imbalanced data in mobile systems [J]. IEEE Transactions on Parallel and Distributed Systems, 2020, 32(1): 59-71.
- [7] 张少波, 张激勇, 朱更明, 等. 基于 Bregman 散度和差分隐私的个性化联邦学习方法 [J]. 软件学报, 2024, 35(11): 5249-5262. ZHANG S B, ZHANG J Y, ZHU G M, et al. Personalized federated learning method based on Bregman divergence and differential privacy [J]. Journal of Software, 2024, 35(11): 5249-5262. (in Chinese)
- [8] GHOSH A, CHUNG J C, YIN D, et al. An efficient framework for clustered federated learning [J]. IEEE Transactions on Information Theory, 2022, 68(12): 8076-8091.
- [9] SATTLER F, MÜLLER K R, SAMEK W. Clustered federated learning: model-agnostic distributed multitask optimization under privacy constraints [J]. IEEE Transactions on Neural Networks and Learning Systems, 2020, 32(8): 3710-3722.
- [10] COLLINS L, HASSANI H, MOKHTARI A, et al. Exploiting shared representations for personalized federated learning [C] // Proceedings of International Conference on Machine Learning. [S.l.]: AAAI Press, 2021: 2089-2099.
- [11] SHAMSIAN A, NAVON A, FETAYA E, et al. Personalized federated learning using hypernetworks [EB/OL]. [2024-11-01]. <https://arxiv.org/pdf/2103.04628>.
- [12] MA X S, ZHANG J, GUO S, et al. Layer-wise model aggregation for personalized federated learning [C] // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Washington D. C., USA: IEEE Press, 2022: 10092-10101.

- [13] XING S K, NING Z H, ZHOU J, et al. N-FedAvg: novel federated average algorithm based on FedAvg [C] // Proceedings of the 14th International Conference on Communication Software and Networks (ICCSN). Washington D.C., USA: IEEE Press, 2022: 187-196.
- [14] ZHANG H, WU T T, CHENG S Y, et al. CC-FedAvg: computationally customized federated averaging [J]. IEEE Internet of Things Journal, 2023, 11(3): 4826-4841.
- [15] ZHOU Y H, YE Q, LÜ J C. Communication-efficient federated learning with compensated overlap-FedAvg [J]. IEEE Transactions on Parallel and Distributed Systems, 2021, 33(1): 192-205.
- [16] SUN T, LI D S, WANG B. Decentralized federated averaging [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 45(4): 4289-4301.
- [17] LAI W L, YAN Q. Federated learning for detecting COVID-19 in chest CT images: a lightweight federated learning approach [C] // Proceedings of the 4th International Conference on Frontiers Technology of Information and Computer (ICFTIC). Washington D.C., USA: IEEE Press, 2022: 146-149.
- [18] 金彪, 李逸康, 姚志强, 等. GenFedRL: 面向深度强化学习智能体的通用联邦强化学习框架 [J]. 通信学报, 2023, 44(6): 183-197.
JIN B, LI Y K, YAO Z Q, et al. GenFedRL: a general federated reinforcement learning framework for deep reinforcement learning agents [J]. Journal of Communications, 2023, 44(6): 183-197. (in Chinese)
- [19] ACHITUVE I, SHAMSIAN A, NAVON A, et al. Personalized federated learning with Gaussian processes [EB/OL]. [2024-11-01]. <https://arxiv.org/pdf/2106.15482>.
- [20] YANG X Y, HUANG W K, YE M. Dynamic personalized federated learning with adaptive differential privacy [C] // Proceedings of Conference on Neural Information Processing Systems (NeurIPS). New York, USA: [s.n.], 2023: 72181-72192.
- [21] YANG F E, WANG C Y, WANG Y C F. Efficient model personalization in federated learning via client-specific prompt generation [C] // Proceedings of the IEEE/CVF International Conference on Computer Vision. Washington D.C., USA: IEEE Press, 2023: 19159-19168.
- [22] YUAN L Q, SU L, WANG Z R. Federated transfer-ordered-personalized learning for driver monitoring application [J]. IEEE Internet of Things Journal, 2023, 10(20): 18292-18301.
- [23] 朱素霞, 顾玢珂, 孙广路. 基于相似度加速的自适应聚类联邦学习 [J]. 通信学报, 2024, 45(3): 197-207.
ZHU S X, GU F K, SUN G L. Adaptive clustering federated learning via similarity acceleration [J]. Journal of Communications, 2024, 45(3): 197-207. (in Chinese)
- [24] 汤凌韬, 王迪, 刘盛云. 面向非独立同分布数据的联邦学习数据增强方案 [J]. 通信学报, 2023, 44(1): 164-176.
TANG L T, WANG D, LIU S Y. Data augmentation scheme for federated learning targeting non-IID data [J]. Journal of Communications, 2023, 44(1): 164-176. (in Chinese)
- [25] SHAMSIAN A, NAVON A, FETAYA E, et al. Personalized federated learning using hypernetworks [EB/OL]. [2024-11-01]. <https://arxiv.org/pdf/2103.04628>.
- [26] YANG Z Y, XIA W J, LU Z X, et al. Hypernetwork-based physics-driven personalized federated learning for CT imaging [J]. IEEE Transactions on Neural Networks and Learning Systems, 2023, 99: 1-15.
- [27] LI H X, CAI Z Y, WANG J Y, et al. FedTP: federated learning by transformer personalization [J]. IEEE Transactions on Neural Networks and Learning Systems, 2023, 35(10): 13426-13440.
- [28] DARLOW L N, CROWLEY E J, ANTONIOU A, et al. CINIC-10 is not ImageNet or CIFAR-10 [EB/OL]. [2024-11-01]. <https://arxiv.org/pdf/1810.03505>.
- [29] SINGLA S, SINGLA S, FEIZI S. Improved deterministic l2 robustness on CIFAR-10 and CIFAR-100 [EB/OL]. [2024-11-01]. <https://arxiv.org/pdf/2108.04062>.
- [30] COHEN G, AFSHAR S, TAPSON J, et al. EMNIST: extending MNIST to handwritten letters [C] // Proceedings of International Joint Conference on Neural Networks (IJCNN). Washington D.C., USA: IEEE Press, 2017: 2921-2926.
- [31] XIAO H, RASUL K, VOLLGRAF R. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms [EB/OL]. [2024-11-01]. <https://arxiv.org/pdf/1708.07747>.
- [32] MCMAHAN B H, MOORE E, RAMAGE D, et al. Communication-efficient learning of deep networks from decentralized data [EB/OL]. [2024-11-01]. <https://arxiv.org/pdf/1602.05629>.
- [33] LI X X, JIANG M R, ZHANG X F, et al. FedBN: federated learning on non-IID features via local batch normalization [EB/OL]. [2024-11-01]. <https://arxiv.org/pdf/2102.07623>.

文字编辑 薛晋栋
栏目编辑 宋 圆