中国科学: 信息科学 2022 年 第 52 卷 第 12 期: 2306–2320
SCIENTIA SINICA Informationis
《中国科学》杂志社
SCIENCE CHINA PRESS
论文

面向异构数据的自适应个性化联邦学习 —— 一种基于参数分解和持续学习的方法

倪宣明[1], 沈鑫圆[1], 张海[2,3*]

1. 北京大学软件与微电子学院, 北京 100871
2. 西北大学数学学院, 西安 710127
3. 华东师范大学统计与数据科学前沿理论及应用教育部重点实验室, 上海 200062
* 通信作者. E-mail: zhanghai@nwu.edu.cn

收稿日期: 2021–05–07; 修回日期: 2021–08–25; 接受日期: 2021–11–01; 网络出版日期: 2022–12–06

国家自然科学基金委员会 — 广东省人民政府大数据科学研究中心项目 (批准号: U1811461) 资助

摘要 联邦学习允许资源受限的边缘计算设备协作训练机器学习模型, 同时能够保证数据不离开本地设备, 但也面临着异构数据下全局模型收敛缓慢甚至偏离最优解的挑战. 为解决上述问题, 本文提出一种自适应个性化联邦学习 (adaptive personalized federated learning, APFL) 算法, 在同时包括空间和时间维度的多任务学习框架下, 考虑面向异构数据的联邦优化问题. 首先, APFL 采用参数分解策略, 将待训练模型参数分解为全局共享参数和客户端特定参数, 在提取所有客户端公共知识的同时实现针对每个客户端的个性化建模. 进一步地, APFL 将每个客户端上执行的局部优化构建为顺序多任务学习, 通过对全局共享参数的更新施加弹性权重巩固 (elastic weight consolidation, EWC) 惩罚, 实现了全局共享模型中重要参数的记忆保留和非重要参数的快速学习. 多个联邦基准数据集上的对比实验验证了本文方法的有效性和优越性.

关键词 联邦学习, 边缘计算, 异构数据, 多任务学习, 持续学习, 参数分解, 个性化

1 引言

物联网和社交网络应用的快速发展导致网络边缘生成的数据量呈指数级增长 [1]. 随着边缘设备的存储容量和计算能力的增长, 在本地存储数据并将更多的网络计算推到边缘正成为一个富有吸引力的解决方案. 联邦学习 (federated learning) [2] 作为一种新兴的人工智能基础技术, 在 2016 年由谷歌 (Google) 最先提出, 它指多个客户端 (包括设备、节点、传感器、组织等) 在一个中央服务器的组织下, 协作训练模型且保证每个客户端的原始数据不离开本地的一种机器学习框架 [3]. 联邦学习能够有效

引用格式: 倪宣明, 沈鑫圆, 张海. 面向异构数据的自适应个性化联邦学习 —— 一种基于参数分解和持续学习的方法. 中国科学: 信息科学, 2022, 52: 2306–2320, doi: 10.1360/SSI-2021-0152
Ni X M, Shen X Y, Zhang H. Adaptive personalized federated learning for heterogeneous data: a method based on parameter decomposition and continual learning (in Chinese). Sci Sin Inform, 2022, 52: 2306–2320, doi: 10.1360/SSI-2021-0152

降低传统中心化机器学习方法带来的一些系统性隐私风险和计算存储成本 [4], 在推荐系统 [5]、金融风险管理、医疗大数据分析 [6] 等众多领域都有着巨大的应用前景, 因此其在研究和应用层面都受到了极大关注.

在面向边缘设备的联邦学习设置中, 数据以高度不均衡的方式分布在数以万计的边缘设备上, 这些设备具有较高的延迟和较低的吞吐量, 而且只能间歇地用于训练. 这些设置与传统的数据集均匀划分且设备连接稳定的分布式学习环境有很大的不同, 也因此面临着异构数据和设备、通信成本、隐私和公平, 以及大规模分布式网络实现等全新挑战 [7,8]. 异构数据挑战指各客户端数据非独立同分布 (非 IID), 这是联邦学习设置的一个典型特征, 因为在实际应用中, 每个客户端对应于特定的用户, 特定的地理位置或特定的时间窗口, 它们各自独立地收集训练数据, 而且这些数据不与其他客户端或中央服务器共享. 因此, 任何一个客户端的数据集分布都无法代表整体, 各客户端上的局部学习过程很可能会彼此分离. 对于客户端只能间歇用于训练的联邦学习来说, 数据异构问题不仅给理论分析带来困难, 也影响联邦学习算法的收敛速度和最终性能.

基于已有的研究成果, 我们注意到, 数据异构下的联邦优化问题可以从空间和时间两个维度共同加以考虑. 从空间维度来看, 如果可以为各客户端模型找到一个公共参数空间并抽象出全局模型, 然后以个性化模型对其加以补充, 就能有效减少客户端模型之间的相互干扰. 从时间维度来看, 如果进行新一轮协作训练时巩固根据之前的客户端数据所学到的知识, 就能将有益的全局知识持续传递下去.

基于上述考虑, 本文提出一种自适应个性化联邦学习 (adaptive personalized federated learning, APFL) 方法, 尝试同时引入持续学习 (continual learning) [9] 中的参数分解策略 [10] 和参数正则化策略 [11~13], 在空间和时间维度共同处理异构数据下的联邦优化问题. 首先, 我们在空间上利用参数分解策略 [10], 试图在简单训练一个初始化全局模型后, 通过构建一个迁移向量自适应地引导全局共享参数关注客户端上的训练, 并自然地分离出全局共享参数和客户端特定参数. 然后, 我们将每个客户端执行的局部优化过程构建为顺序多任务学习 (multi-task learning), 并将局部更新后偏离初始全局模型的现象类比为多任务学习中的灾难性遗忘 (catastrophic forgetting) 问题, 即根据新任务训练并更新模型参数后, 模型不再适应先前学习的旧任务. 为此, 本文利用参数正则化方法, 通过增加一个弹性的正则项实现参数的受限更新, 按照每个参数在先前轮次全局任务中的重要性权重调整其在下一轮的学习速度, 因此可以在每次局部更新时保护对全局任务重要的参数, 而对全局任务不重要的参数更新则给予客户端更大的改动自由. 通过引入重要性权重更新机制, 大大减少了重要性权重的通信和计算需求, 使得相比于基准联邦平均 (federated averaging, FedAvg) [2] 框架, APFL 几乎没有增加额外的通信开销.

本文的贡献如下. 第一, 在同时包括空间和时间维度的多任务学习框架下, 考虑面向异构数据的联邦优化问题, 一方面采用参数分解策略, 抽象出全局共享参数和客户端特定参数, 实现个性化联邦学习, 减少客户端模型间的干扰; 另一方面, 采用参数正则化手段, 通过弹性更新全局共享参数在时间维度上巩固和保留有益的全局知识. 第二, 通过引入重要性权重更新机制, 大大优化了本文方法的通信和计算效率. 第三, 在多个联邦学习基准数据集上对本文方法进行了充分的评估, 实验结果表明 APFL 在各实验数据集上均能快速稳健地收敛, 并且显著超越联邦学习的基准和先进方法. 相比 FedAvg, APFL 在 Vehicle, MNIST, FEMNIST 和 CIFAR-10 数据集上的平均测试正确率分别提升了 5.41, 6.01, 30.42 和 10.26 个百分点, 同时 APFL 对于超参数的取值具有稳健性.

2 相关工作

为了提高数据异构下全局模型的收敛能力, 联邦优化问题成为了近期众多研究工作的主题. FedAvg 算法 [2] 作为目前主流的联邦优化方法, 在数据异构环境中取得了经验上的成功, 为解决上述问题提供了一个通用的起点, 但仍然存在收敛缓慢且难以调整, 甚至偏离最优解的问题 [14]. 为了提高在数据异构环境中的收敛能力, 大量的改进方法相继被提出, 已有研究大致可以分为共享数据、压缩局部更新和个性化等几个方面. 共享数据策略通过构建在所有客户端之间全局共享的一小部分数据以获得更加同质的数据分布, 如客户端共同训练一个生成式对抗网络 (generative adversarial networks, GAN) 模型, 将本地数据扩展为一个 IID 数据集 [15], 或在初始化阶段将具有统一分布的全局数据中的少量数据子集部署到客户端中并利用其训练初始化模型. 显然这种策略不仅会加重通信负担, 而且会增加隐私泄露隐患 [15], 或者需要付出额外努力以生成或收集辅助数据. 压缩局部更新策略的代表性方法为 Li 等 [16] 提出的 FedProx 算法, 其为 FedAvg 添加了一个限制局部更新偏差量的近端项 (proximal term) 以避免局部模型的发散. 他们的实验结果显示, FedProx 在高度异构网络中相比 FedAvg 收敛更为稳健, 但在收敛速度上没有明显的突破.

尽管标准联邦学习框架旨在学习单个全局模型, 但已有一些工作致力于实现客户端的个性化建模, 主要策略包括模型混合、多任务学习 (multi-task learning) 和局部微调 (fine-tuning) 3 类. 模型混合策略通常使用一个混合参数来寻求全局模型和局部模型之间的显式权衡. Smith 等 [17] 提出的 MOCHA 算法是多任务学习策略的代表性方法. MOCHA 通过多任务学习框架为每个客户端学习独立但相关的模型来实现个性化建模, 提高了处理异构网络的能力, 但统一优化的架构使其对于新的参与设备, 非凸问题和大型联邦网络的可伸缩性和泛化性较差. 局部微调是个性化的主流思路, 常见方法是将全局模型的生成和个性化视作两个独立的过程, 通过引入元学习 (meta learning)、迁移学习 (transfer learning) 等技术基于全局模型进行再训练, 形成联邦元学习 [18], 针对深度神经网络联邦训练的分层匹配 [19] 等算法框架. 局部微调策略有助于建立一个易于个性化的全局模型, 但是现有方法大多需要进行额外训练, 带来更大的计算负担, 而且微调过程常以最小化本地训练损失为目标, 在大量客户端数据不充分的联邦环境下, 存在对本地训练数据过拟合的风险. 总体来看, 若以学习单一全局模型为目标, 则不可避免地存在局部模型之间的相互影响和干扰, 因此个性化策略在面向异构数据的联邦学习中具有天然的优势和必要性, 但开发通用性, 可扩展性更强的个性化联邦学习算法以实现异构数据下的快速、稳定收敛并使得大多数客户端受益, 仍然是一个亟待解决的开放问题和关键挑战.

持续学习 (continual learning) 或者终身学习 (lifelong learning) [9] 的目的是使神经网络学习器从之前的任务中获取经验以更快更好地适应新的任务, 同时克服灾难性遗忘, 保留先前获得的知识. 目前持续学习已分别针对持续的监督学习 [20]、无监督学习 [21]、半监督学习 [22] 和强化学习 [23] 等任务形成了相应的发展方向, 但将持续学习应用于联邦学习领域以解决数据异构下的联邦优化问题尚且是一个较新的研究领域. 现有的相关工作基本采用的是局部持续训练的策略, 鼓励客户端在训练局部模型时保留全局知识, 试图通过克服局部训练中的灾难性遗忘缓解模型发散. 这类工作包括对局部模型增加持续学习正则项以避免与全局任务关联较大的参数更新幅度过大的 FedCurv [24] 和 FedCL [25] 方法, 以及基于知识蒸馏 (knowledge distillation) 挖掘全局模型知识的 FedLSD [26] 方法. 这些方法减少了客户端模型之间的差异, 一定程度上减轻了数据异构带来的影响, 但仍然无法避免协作过程中局部模型之间的相互干扰, 特别是在极端数据异质性下, 客户端间的知识相差很大, 即使有定义良好的正则化器, 聚合得到的全局模型依然是嘈杂的.

最近, 有研究者提出了联邦持续学习 (federated continual learning) 的概念 [27, 28], 假设多个模型在

分布式客户端上进行持续学习, 通过一个全局服务器交流客户端间的特定任务参数. Yoon 等 [27] 提出将模型参数分解为全局联邦参数, 特定客户端参数和特定任务参数, 每个客户端利用注意力机制有选择地使用从其他客户端那里获得的关于特定任务的知识. 他们的方法进一步推动了联邦学习和持续学习的融合性研究, 但从研究的问题来看仍然是特定场景下的持续学习, 可以看成是多智能体终身学习 (multi-agent lifelong learning) 的拓展, 与注重通信开销, 客户端数据隐私且面临异构设备和大规模分布式网络挑战的联邦优化问题的关注重点有所不同.

3 方法

本节提出一种自适应个性化联邦学习算法 APFL, 3.1 小节首先介绍了数据异构下的联邦学习的问题设置, 3.2 和 3.3 小节分别提出用于联邦学习的参数分解和全局共享参数弹性巩固策略, 并给出了本文算法.

3.1 问题设置

在一个典型的面向边缘设备的联邦学习训练过程中, 数以万计的远程客户端定期与服务器通信以获取全局模型. 在每一轮通信中, 服务器将当前的全局模型分发给一部分客户端, 客户端基于本地数据执行局部优化, 并将这些局部更新发送到中央服务器. 服务器将它们聚合以更新全局模型 (通常是平均), 之后根据新全局模型的表现, 选择停止训练或者开始新一轮的通信 [4].

不失一般性, 假设各客户端 (以 k 为索引) 均使用一个 L 层深度神经网络模型完成多分类任务. 记待求解的神经网络模型参数集为 $\boldsymbol{W}=\{\boldsymbol{w}^l\}_{l=1}^L$, 其中 $\boldsymbol{w}^l$ 表示第 l 层的模型参数, 联邦学习的目标通常是最小化如下目标函数以学习一个共享的全局模型, 供所有客户端使用:

$$\mathcal{L}(\boldsymbol{W})=\sum_{k=1}^{N}p_k\mathcal{L}_k(\boldsymbol{W}), \tag{1}$$

其中 N 为参与模型训练的客户端数量, $p_k\geqslant 0$ 且 $\sum_k p_k=1$, 通常可设 $p_k=n_k/\sum_k n_k$, 其中 n_k 表示客户端 k 上的可用训练样本数. 局部目标函数 $\mathcal{L}_k(\boldsymbol{W})$ 通常度量的是本地数据分布的经验风险 (empirical risk) 或称为经验损失, 即

$$\mathcal{L}_k(\boldsymbol{W};\mathcal{P}_k)=\frac{1}{n_k}\sum_{i\in\mathcal{P}_k}\ell_i(\boldsymbol{W}), \tag{2}$$

其中 $\mathcal{P}_k$ 为客户端 k 上的训练数据集. 不同于集中式学习中使用 IID 的样本数据, 联邦学习使用终端设备独立收集或生成的本地数据, 而且这些数据不与其他客户端或服务器共享, 因此各客户端的数据异构性是联邦学习的典型特征. 在数据严重异构的情形下, 各客户端的局部优化目标 $\mathcal{L}_k(\boldsymbol{W})$ 可能相差甚远, 导致局部更新发散, 不能保证良好的收敛效果. 在 3.2 和 3.3 小节中, 我们提出一个自适应个性化联邦学习框架, 通过使用自适应参数分解和参数正则化方法构建新的联邦优化目标, 以试图缓解数据异构造成的联邦学习模型收敛困难的问题.

3.2 参数分解

联邦学习受益于在客户端之间分享知识, 但标准的联邦学习需要所有参与设备就协作训练的共同模型达成一致, 因此各客户端不可避免地会受到来自其他参与设备的学习任务的影响和干扰, 而不相关的学习任务很可能会损害其模型性能. 本小节针对单一全局模型参数的缺点, 基于持续学习中的加

性参数分解 (additive parameter decomposition, APD) [10] 策略提出了一种联邦参数分解方法, 以抽象出全局共享参数, 并分解出客户端个性化参数.

对于神经网络模型的各层 (以 l 作为索引), 我们假设待训练的模型参数可以分解为全局共享参数 $\boldsymbol{w}^l_{S(k)}$ 和客户端特定参数 $\boldsymbol{w}^l_k$. 通过找到一个迁移向量 $\boldsymbol{m}^l_k$, 可以自适应地引导全局共享参数关注客户端 k 上的训练, 即

$$\boldsymbol{w}^l = \boldsymbol{w}^l_{S(k)} \odot \boldsymbol{m}^l_k + \boldsymbol{w}^l_k, \tag{3}$$

其中 $\odot$ 表示每个网络单元 (unit) 的对应元素相乘. $\boldsymbol{w}^l_{S(k)}$ 与 $\boldsymbol{w}^l_k$ 维度相同, $\boldsymbol{m}^l_k \in \mathbb{R}^{u^l \times 1}$, 其中 u^l 为第 l 层单元数. 模型训练时, $\boldsymbol{w}^l_{S(k)}$, $\boldsymbol{m}^l_k$, $\boldsymbol{w}^l_k$ 同时通过反向传播计算梯度以进行更新, $\boldsymbol{m}^l_k$ 额外通过 sigmoid 函数限制其元素 $m^l_k \in (0, 1)$. 局部更新结束后, 全局共享参数 $\boldsymbol{w}^l_{S(k)}$ 传回服务器计算平均值获得全局模型 $\boldsymbol{W}_S$, 迁移向量 $\boldsymbol{m}^l_k$ 和客户端特定参数 $\boldsymbol{w}^l_k$ 则保留在本地, 等待下一次本地模型训练时更新, 并供模型测试时调用.

为表示简便, 忽略网络层索引 l, 将式 (3) 重写为 $\boldsymbol{W} = \boldsymbol{W}_{S(k)} \odot \boldsymbol{M}_k + \boldsymbol{W}_k$, 其中全局共享参数 $\boldsymbol{W}_{S(k)} = \{\boldsymbol{w}^l_{S(k)}\}$ 试图抽象出客户端之间的公共知识, 客户端特定参数 $\boldsymbol{W}_k = \{\boldsymbol{w}^l_k\}$ 则着力于捕获客户端 k 的特定知识. 因此该策略不仅可以像标准联邦学习一样构建全局模型, 而且允许客户端保留个性化模型. 式 (3) 蕴含了迁移学习的思想, 试图通过迁移向量 $\boldsymbol{M}_k = \{\boldsymbol{m}^l_k\}$, 在缓解客户端之间模型干扰的同时实现客户端间的知识迁移. 此外, 由于仅需和服务器交换全局共享参数, 因此相比于直接交换参数的标准联邦学习框架, 本文方法不仅没有增加通信负担, 而且可以更好地保护数据隐私.

直觉上, 简单训练一个初始化全局模型后再进行参数分解, 会使模型在训练的初期有更合适的起点, 因此本文设置参数 z 控制开始分解的轮数. 实验结果显示这样的设置进一步提升了算法初期的表现, 这也体现了通过微调 (fine-tuning) 实现个性化建模的思想.

3.3 全局共享参数弹性巩固

引入 t 表示全局模型训练轮次. 在第 t 轮训练中, 客户端 k 以中央服务器分发的全局模型 $\boldsymbol{W}^t_S$ 作为初始参数, 并利用本地数据对其进行更新. 在各客户端数据异构且仅间歇用于训练的联邦学习中, 各轮的训练数据集都存在分布上的差异, 可以认为各轮的学习任务不断变化, 但模型参数的训练是连续的. 因此, 本文将上述局部更新过程构建为顺序多任务学习, 即将上一轮的全局模型 $\boldsymbol{W}^t_S$ 视为已经学习完毕的旧任务, 而将本轮基于本地数据学习的全局共享参数 $\boldsymbol{W}^{t+1}_{S(k)}$ 看作新任务. 我们的目的包括两个方面: 第一, 从之前任务中获取经验以更快更好地学习当前任务, 第二, 学习当前任务时, 不会忘记之前已经学会的任务.

由于模型参数的训练是连续的, 当基于本轮客户端数据更新全局模型时, 模型会调整它学到的关于旧数据的参数以适应新任务, 这样之前任务的知识就很有可能被覆盖, 可以将其类比为神经网络的灾难性遗忘问题. 持续学习中的参数正则化方法, 包括 EWC (elastic weight consolidation) [11], SI (synaptic intelligence) [12], MAS (memory aware synapses) [13] 等, 为解决这一问题提供了有效思路. 这种方法存在的基础是绝大多数神经网络都是过参数化 (over-parameterization) 的, 许多参数配置可以得到相同的性能, 即可能存在特定的模型参数对于新任务最优, 并接近旧任务的最优参数. 因此, 本小节以经典的 EWC [11] 算法为例, 进一步提出利用持续学习中的参数正则化方法, 克服联邦学习中的灾难性遗忘问题.

EWC [11] 算法从贝叶斯 (Bayes) 的视角看待神经网络的训练, 致力于通过约束参数更新, 在最大化新任务似然概率的同时保持旧任务的后验概率最大化. 其在新任务的训练过程中, 通过放慢在旧任务中重要参数的学习, 将参数约束在围绕旧任务最优参数的低误差区域内以克服灾难性遗忘. 为了度

量各参数分量对于旧任务的重要性, EWC 算法利用估计的 Fisher 信息度量各参数分量估计的准确度, 并将据其定义的新旧任务参数的距离作为正则化惩罚项添加到损失函数中.

本文在客户端的局部目标函数中添加上述 EWC 惩罚项, 以在学习全局共享参数时强调对全局任务重要参数的记忆保留, 并放宽对非重要参数的可塑性, 即非重要参数仍可以拥有较高的学习率. 在第 t 轮向客户端 k 施加的 EWC 惩罚项 $\mathcal{R}_k$ 如下所示:

$$\mathcal{R}_k(\boldsymbol{W}_{S(k)};\boldsymbol{W}_S^t)=\sum_{j\in\mathcal{S}}\hat{F}(\boldsymbol{W}_S^t)_{jj}(w_{S(k),j}-w_{S,j}^t)^2, \tag{4}$$

其中 $j\in\mathcal{S}$ 表示全局共享参数 $\boldsymbol{W}_S$ 分量的索引, $\hat{F}(\boldsymbol{W}_S^t)_{jj}$ 是 Fisher 信息矩阵中第 j 个对角元的估计, 衡量了参数估计的准确度, 代表全局共享参数 $\boldsymbol{W}_S$ 的分量 $w_{S,j}$ 的重要性权重. 越大的 $\hat{F}(\boldsymbol{W}_S^t)_{jj}$ 对应越高的参数估计准确度, 意味着 $w_{S,j}$ 在前 t 轮代表了越多的全局任务信息, 对全局任务越关键, 因此在第 $t+1$ 轮的局部更新中 $w_{S,j}$ 的更新会受到更大的限制, 反之则受到的限制更小.

EWC 算法会为每个历史任务上训练得到的最优参数都维护一个惩罚项, 所以惩罚项会随任务数量线性增长. 在联邦学习中, 为每一轮迭代形成的全局模型都维护一个惩罚项显然没有必要, 因此本文采用 online EWC [29] 算法, 在估计 $\boldsymbol{W}_S^{t+1}$ 时只维护一个针对 $\boldsymbol{W}_S^{\text{old}}$ 施加的惩罚项, 而重要性权重 $\hat{F}(\boldsymbol{W}_S^t)_{jj}$ 则采取带权累加的方式得到

$$\hat{F}(\boldsymbol{W}_S^t)_{jj}=\lambda\hat{F}(\boldsymbol{W}_S^{\text{old}})_{jj}+\sum_{k=1}^{K}p_k\Big[\big(\nabla\mathcal{L}_k(\boldsymbol{W}_S;\mathcal{P}_k)|_{\boldsymbol{W}_S^{t-1}}\big)_j\Big]^2, \tag{5}$$

其中 K 表示本轮参与联邦学习的客户端数, $\hat{F}(\boldsymbol{W}_S^{\text{old}})_{jj}$ 表示上一次计算得到的重要性权重, $\lambda\in[0,1]$ 为遗忘系数, 相当于根据之前任务计算的重要性权重对最终结果的贡献会逐渐降低. λ 代表了对之前通信轮次中全局模型的记忆能力, λ 越大, 越强调全局模型的累积记忆.

下面考虑何时更新重要性权重 $\hat{F}(\boldsymbol{W}_S^t)_{jj}$. EWC 算法假设训练数据来自各个边界明确的任务, 各个任务都被充分学习后才会进行下一个任务的学习, 因此可将训练过程划分为连续的阶段, 在训练阶段之间, 估计每个模型参数的重要性权重, 并惩罚 (放慢) 对旧任务而言重要参数的变化 [30]. 然而这一假设在联邦学习中并不成立, 因为在联邦学习的每轮训练中, 仅有很少量的客户端以分布式训练的形式参与, 这些客户端的数据分布并不相同, 每一轮的训练样本都不具有代表性甚至可能有较大差异, 这意味着聚合模型是嘈杂的, 尤其是在训练的早期阶段. 在这种情况下不适合将每轮更新都当作是一个任务的交替并逐轮更新重要性权重 $\hat{F}(\boldsymbol{W}_S^t)_{jj}$, 因为并非每轮的全局模型知识都是有效且值得记忆的, 而且与模型参数同样大小的 $\hat{F}(\boldsymbol{W}_S^t)_{jj}$ 的传输会带来两倍于 FedAvg 的通信负担.

借鉴 Aljundi 等 [30] 提出的任务无关 (task-free) 持续学习中的重要性权重更新机制, 本文进一步将全局损失先增加再减少随后平稳的平稳期作为更新重要性权重的时机. 全局损失增加表明联邦系统遇到了包含新知识的样本, 再减少意味着模型学到了这些新知识, 这样一升一降的阶段被称为一个 “峰”. 随后损失平稳时表明学习已经趋于平稳, 这时正应该巩固所学到的知识. 本文同样利用滑动窗口探测损失平稳期, 当窗口损失值均值高于前一个平稳窗口损失值的均值 + 标准差 ($\mu_L^{\text{old}}+\sigma_L^{\text{old}}$) 时, 说明检测到了 “峰”, 即遇到了新的需要被学习的知识, 当窗口内的平均训练损失值的均值与标准差都低于阈值 $(\delta_\mu,\delta_\sigma)$ 时, 触发重要性权重的更新.

随后的实验部分将证明, 该更新策略可以使得本文方法在几乎等同于标准 FedAvg 的通信开销的前提下, 仍然保证较好地缓解灾难性遗忘的问题.

最后, 结合式 (3) 和 (4), 忽略网络层索引 l, 得到了本文构建的联邦学习客户端局部目标函数:

$$\min_{\boldsymbol{W}_{S(k)},\boldsymbol{M}_k,\boldsymbol{W}_k} \mathcal{L}_k(\boldsymbol{W}_{S(k)} \odot \boldsymbol{M}_k + \boldsymbol{W}_k; \mathcal{P}_k) + \frac{\mu}{2}\sum_{j\in S} \hat{F}(\boldsymbol{W}_S^t)_{jj}(w_{S(k),j} - w_{S,j}^t)^2, \tag{6}$$

其中 μ 为正则化参数, 设置了对全局共享参数的 EWC 惩罚的强度. 算法 1 总结了本文所提自适应个性化联邦学习算法 APFL 的具体流程.

Algorithm 1 APFL: adaptive personalized federated learning

Input: Training datasets $\{\mathcal{P}_k\}_{k=1}^N$, number of clients trainined per round K (indexed by k), number of rounds to start parameter decomposition z, update thresholds for loss window $\delta_\mu, \delta_\sigma$, and model hyperparameters η, E, μ, λ.

//Run on the server
Initialize $\boldsymbol{W}_S^0$, $\{\hat{F}(\boldsymbol{W}_S^0)_{jj}\} = \boldsymbol{0}$, loss window $L_{\text{win}} = \{\}$, $\mu_L^{\text{old}} = 0$, $\sigma_L^{\text{old}} = 0$, peak indicator p = False, indicator for importance weight update flag = 0;
for each round $t = 0, \dots$ **do**
  $G_t \leftarrow$ (random subset of K devices)
  for each device $k \in G_t$ in parallel **do**
    $\{\boldsymbol{W}_{S(k)}^{t+1}, \nabla\mathcal{L}_k(\boldsymbol{W}_S)|_{\boldsymbol{W}_S^t}\} \leftarrow$ ClientUpdate$(k, \boldsymbol{W}_S^t, \{\hat{F}(\boldsymbol{W}_S^t)_{jj}\}, t-z, \text{flag})$;
  end for
  Aggregate the $\boldsymbol{W}_{S(k)}^{t+1}$'s as $\boldsymbol{W}_S^{t+1} \leftarrow \frac{\sum_k n_k \boldsymbol{W}_{S(k)}^{t+1}}{\sum_k n_k}$;
  Update loss window L_{win} and compute $\mu(L_{\text{win}})$, $\sigma(L_{\text{win}})$;
  if $p \wedge \mu(L_{\text{win}}) < \delta_\mu \wedge \sigma(L_{\text{win}}) < \delta_\sigma$ **then**
    $\mu_L^{\text{old}} = \mu(L_{\text{win}})$, $\sigma_L^{\text{old}} = \sigma(L_{\text{win}})$, p = False, flag = 1;
    Update $\{\hat{F}(\boldsymbol{W}_S^{t+1})_{jj}\}$ using Eq. (5); //Update the importance weights when plateaus of loss are detected
  end if
  if $\neg p \wedge \mu(L_{\text{win}}) > \mu_L^{\text{old}} + \sigma_L^{\text{old}}$ **then**
    p = True; //Detect peaks in the loss surface
  end if
end for
//Run on client k
ClientUpdate$(k, \boldsymbol{W}_S^t, \{\hat{F}(\boldsymbol{W}_S^t)_{jj}\}, t-z, \text{flag})$:
if $t - z > 0 \wedge$ ($\boldsymbol{M}_k$ and $\boldsymbol{W}_k$ not exist) **then**
  Initialize $\boldsymbol{M_k} = \boldsymbol{1}$, $\boldsymbol{W}_k = \boldsymbol{W}_S^t$;
end if
Perform E epochs of mini-batch SGD (stochastic gradient descent) on Eq. (6) with $\{\hat{F}(\boldsymbol{W}_S^t)_{jj}\}$ and learning rate η to obtain $\boldsymbol{W}_{S(k)}^{t+1}$, and update $\boldsymbol{M}_k, \boldsymbol{W}_k$ if $t - z > 0$;
return $\begin{cases} (\boldsymbol{W}_{S(k)}^{t+1}, \nabla\mathcal{L}_k(\boldsymbol{W}_S)|_{\boldsymbol{W}_S^t}), & \text{flag} = 1, \\ \boldsymbol{W}_{S(k)}^{t+1}, & \text{flag} = 0, \end{cases}$ to the server, save $\boldsymbol{M}_k$ and $\boldsymbol{W}_k$ locally if $\boldsymbol{M}_k$ and $\boldsymbol{W}_k$ exist.

4 实验

为了充分评估本文算法的有效性, 本节在不同的联邦基准数据集上进行综合实验, 4.1 小节提供了详细的实验设置信息, 4.2 小节首先展示了 APFL 与各对比方法在多个真实数据集上的实验结果, 然后检验了 APFL 对不同参数取值的稳健性.

4.1 实验设置

4.1.1 数据和模型

为了充分评估所提出的 APFL 算法对非 IID 联邦数据集的稳健性, 本文依据最近的联邦学习基

表 1 真实联邦数据集统计信息

Table 1 Statistics of real federated data sets

Dataset	#Clients	#Samples	#Samples/clients	
			Mean	Standard
Vehicle	23	43695	1899	349
MNIST	1000	69035	69	106
FEMNIST	200	18345	92	159
CIFAR-10	100	49757	498	344

准, 在 4 个真实数据集上进行了综合评估:

Vehicle. 使用和 Smith 等 [17] 相同的联邦 Vehicle 数据集, 包括由 23 个传感器组成的分布式网络中收集的声学、地震和红外车辆传感器数据, 用于根据路段对行驶的车辆进行分类 [31]. 每个样本由 100 维特征数据和一个二进制标签 (aav 型或 dwv 型车辆) 组成[1)]. 实验中将每个传感器作为一个客户端, 通过联邦学习协作训练一个线性支持向量机 (support vector machine, SVM) 模型完成二分类任务.

MNIST. 使用和 Li 等 [16] 相同的联邦 MNIST 数据集. MNIST 数据集 [32] 是手写数字识别问题的经典通用数据集, 其输入是一幅 28×28 (展平成 784 维) 的图像, 输出是 0∼9 之间的类标签[2)]. 为了模拟数据异构的联邦学习设置, 将数据分布在 1000 个客户端上, 大多数客户端只拥有 2 种数字样本, 每个客户端的样本数服从幂律分布 (power law) 以模拟不均衡的样本分布. 客户端协作训练一个多项逻辑回归 (multinomial logistic regression) 模型.

FEMNIST. 使用和 Li 等 [16] 相同的联邦 FEMNIST 数据集. FEMNIST 数据集 [33] 是 MNIST 的扩展, 除手写数字外还包含手写字母的分类任务, 具有和 MNIST 相同的图像结构和参数[3)]. 实验中从 FEMNIST 中提取 10 个小写字符 ("a"∼"j"), 为每个客户端分配 3 种字符样本, 总共设置 200 个客户端, 利用联邦学习协作训练多项逻辑回归模型.

CIFAR-10. 基于 CIFAR-10 数据集创建. CIFAR-10 数据集[4)]是一个广泛使用的计算机视觉数据集, 图像大小为 32×32 像素 (展平成 1024 维), 一共标注为 10 类. 实验中将数据分布在 100 个客户端中, 每个客户端拥有 5 个类的样本, 每个设备的样本数遵循幂律分布以模拟不均衡的样本分布. 客户端协作训练一个卷积神经网络 (convolutional neural network, CNN) 模型, 模型输入为单通道灰度图, CNN 网络包含两个卷积核大小为 $(5,5)$ 的卷积层, 卷积核数量分别为 20 和 50, 每个卷积层后接一个 ReLU 激励层和一个采用最大池化 (max pooling) 方式的池化层, 池化核大小为 $(2,2)$, 最后连接了两个全连接层, 分别拥有 800 和 10 个神经元.

数据集的汇总信息见表 1. 为了展示数据异构对联邦学习算法效果的影响, 我们还额外生成了上述各数据集的 IID 版本: 将分布在各客户端的所有数据集中起来后重新进行随机划分并分布在各客户端上, 然后以 mini-batch SGD ($E = 1$) 作为优化器, 在每轮都对所有客户端数据进行顺序训练, 以模拟集中式学习时的表现.

1) http://www.ecs.umass.edu/mduarte/Software.html.
2) http://yann.lecun.com/exdb/mnist/.
3) https://www.westernsydney.edu.au/bens/home/reproducible_research/emnist.
4) http://www.cs.toronto.edu/~kriz/cifar.html.

4.1.2 对比方法

• FedAvg [2]. 面向非 IID 数据提出的第 1 个同步联邦学习方法, 在数据非 IID 环境中取得了经验上的成功, 是目前联邦学习研究的标准基线. FedAvg 直接以客户端训练数据的经验损失作为局部目标函数. 在 FedAvg 方法中, 客户端 k 基于本地训练数据 $\mathcal{P}_k$ 以式 (2) 的训练数据经验损失为优化目标执行 mini-batch SGD 更新模型. 各客户端具有相同的学习率 η 和 epoch 数 E, 由服务器每隔一段时间对模型参数进行平均.

• FedProx [16]. 针对异构网络的 FedAvg 变体. FedProx 通过在每个局部目标函数中添加一个近端项限制待训练局部模型和全局模型 (初始局部模型) 之间的欧氏距离, 避免客户端学习目标偏离全局目标.

• FedCurv [24]. 同样引入持续学习中的 EWC 算法, 通过对局部模型增加持续学习正则项克服数据异构下联邦学习的灾难性遗忘. 其重要性权重仅由上一轮训练计算得到的 Fisher 信息矩阵确定.

• FedMeta [34]. 一种典型的以元学习方法实现局部微调的联邦学习方法. 各客户端使用模型不可知元学习 (model-agnostic meta-learning, MAML) 训练模型的初始参数, 使预先训练的模型在使用少量目标客户端上的数据进行快速适应后可以实现有效更新.

4.1.3 实验细节

实验环境. 实验模拟了具有 1 个中央服务器和 N 个客户端的联邦学习架构, 基于 Li 等 [16] 发布的框架[5)]实现了 FedAvg, FedProx, FedCurv, FedMeta 和 APFL 算法. 所有实验在搭载 2 个 Intel(R) Xeon(R) CPU 和 1 个 NVIDIA Tesla P100 GPU 的 Google Colaboratory 云端 IDE 上运行.

数据集及客户端设置. 所有客户端上的数据都被随机划分为训练集和测试集, 对于 Vehicle, MNIST, FEMNIST, 直接使用开源发布的训练集和测试集, 对于 CIFAR-10, 将每个客户端上 80% 的数据随机划分为训练集, 20% 划分为测试集. 特别地, 对于联邦元学习 FedMeta 方法, 每个客户端上的训练集又被进一步划分, 80% 作为支持集 (support set), 20% 作为查询集 (query set). Vehicle, MNIST, FEMNIST 和 CIFAR-10 数据集的最大通信轮次分别设置为 20, 150, 200 和 100. 所有实验中服务器统一用与 n_k 数量成比例的权重聚合客户端返回的更新参数, 每轮所选客户端的数量均设为 10. 在每个对比实验中, 固定所有运行轮次中随机选择的设备和数据批量顺序 (mini-batch orders).

超参数. 统一使用固定学习率的 mini-batch SGD 作为本地求解器, 批量大小 (batch size) 均设置为 10. 每轮 epoch 数 E 和学习率 η 是联邦学习方法共有的超参数, 为了进行公平的比较, 在性能对比部分将所有数据集上的 E 均设置为 5, 各方法在 Vehicle, MNIST, FEMNIST 和 CIFAR-10 的学习率分别设置为 0.03, 0.03, 0.003 和 0.01, FedMeta 算法中的元学习率在所有数据集上均设置为 0.01. FedProx, FedCurv 和 APFL 的正则化参数 μ 等价地设置为 1, 且 FedCurv 和 APFL 均以经验 (empirical) Fisher 信息矩阵的对角元代替 Fisher 信息矩阵的对角元. APFL 额外设置的超参数 λ (遗忘系数) 取值为 1, 重要性权重更新机制中的损失均值和标准差的阈值 $(\delta_\mu, \delta_\sigma)$ 在 MNIST 和 FEMNIST 数据集上设置为 $(1, 0.5)$, Vehicle 数据集上设置为 $(700, 400)$, CIFAR-10 数据集上设置为 (1, 0.2). z (开始参数分解的轮数) 设置为 2, 为了保证突然的参数分解不会使模型发生突变, 设置在开始参数分解的前后两轮内都更新重要性权重.

评价指标. 采用联邦学习文献中普遍使用的性能评估指标, 基于全局目标 $\mathcal{L}(\boldsymbol{W})$ 报告训练集的平均经验损失和测试集的平均正确率 (accuracy) 等指标. 由于假设每个通信回合都对应于特定的聚合时

5) https://github.com/litian96/FedProx.

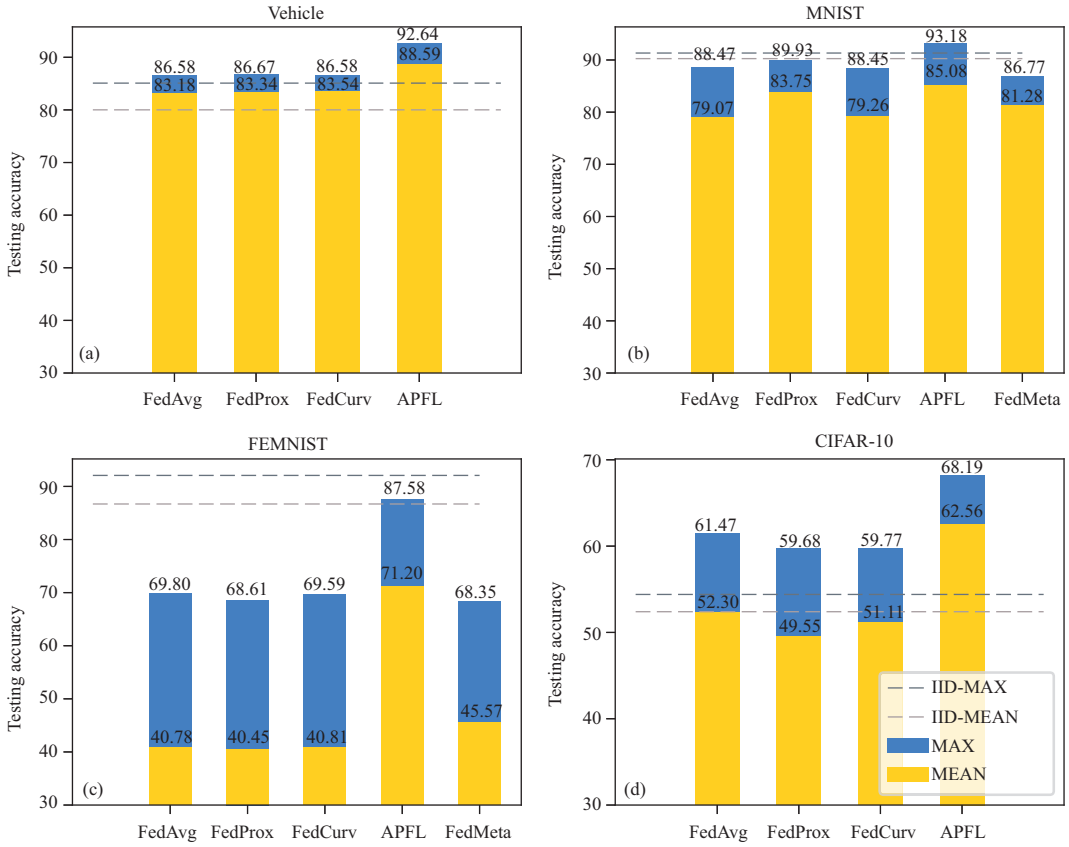

图 1 (网络版彩图) 不同数据集上各方法的平均 & 最高测试正确率 (%)

Figure 1 (Color online) Average & the highest testing accuracy (%) of each method on different datasets. (a) Vehicle; (b) MNIST; (c) FEMNIST; (d) CIFAR-10

间戳, 因此以回合形式报告结果.

4.2 结果和讨论

4.2.1 性能比较

图 1 汇总了各实验数据集上所有通信轮次的平均测试正确率[6)]和最高测试正确率. 可以观察到, 由于数据异构性的存在, 相比 ID 数据下利用 mini-batch SGD ($E = 1$) 进行训练的结果, 基准的 FedAvg 算法在 MNIST 和 FEMNIST 数据集上的精度有大幅下降, 其他对比算法对其的改善并不明显, 但 APFL 在所有实验数据集上都取得了最高的平均测试正确率, 相比基准 FedAvg 方法分别提升了 5.41 (Vehicle), 6.01 (MNIST), 30.42 (FEMNIST) 和 10.26 (CIFAR-10) 个百分点. 而且 APFL 的算法精度非常接近甚至在 Vehicle 和 CIFAR-10 上远超了 IID 数据下的集中式学习的表现, 这说明了数据异构下的联邦学习如果恰当使用个性化策略, 可以得到优于集中式学习的算法表现. 上述结果表明 APFL 可以建立具有更高全局正确率的联邦模型, 且适用于多种数据和待训练模型.

图 2 直观地展示了所有实验数据集上各对比方法的平均经验损失和测试正确率随通信回合的变化. 可以看出, FedAvg 和 FedProx 收敛相对缓慢, 在一些客户端上无法将经验损失降至低值, 而且全

6) 均从第 0 轮开始算起.

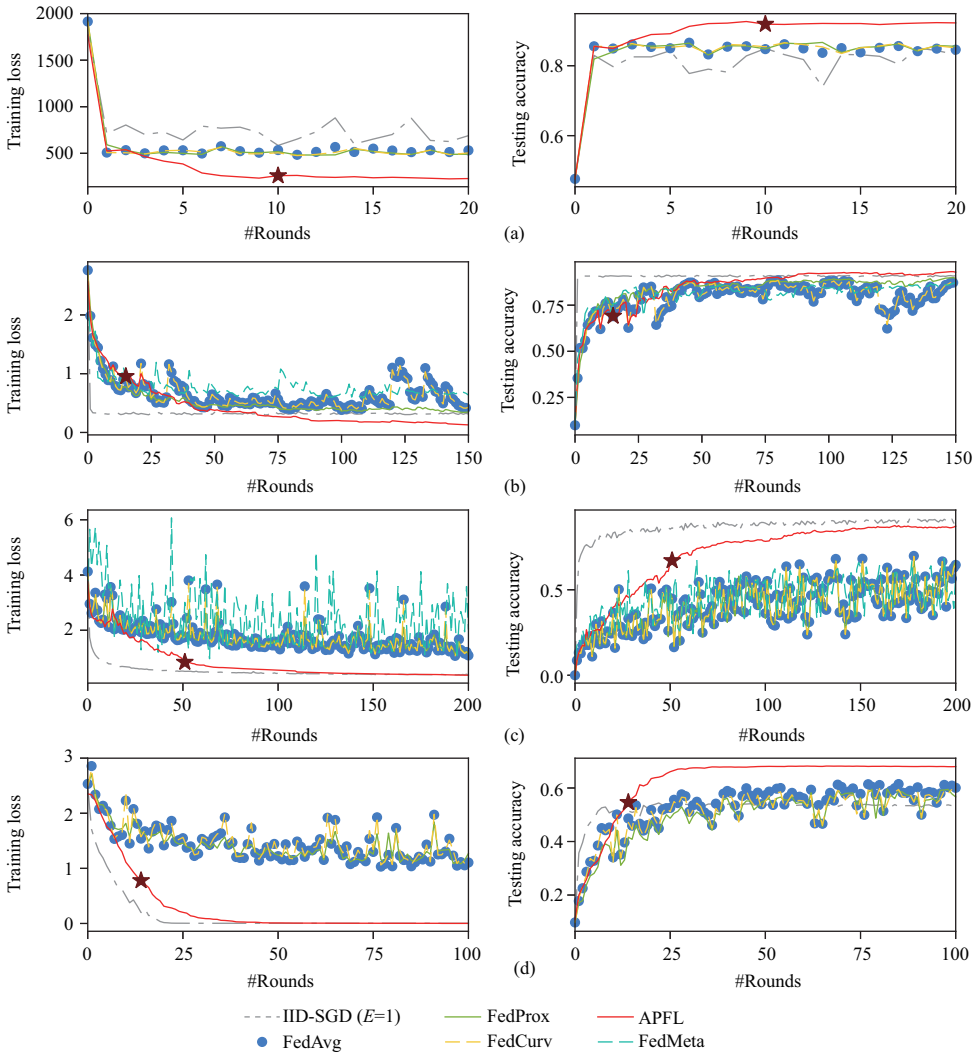

图 2　(网络版彩图) 不同数据集上各方法训练损失和测试正确率随通信回合的变化

Figure 2　(Color online) Training loss and testing accuracy vs. communication rounds of each method on different datasets. (a) Vehicle; (b) MNIST; (c) FEMNIST; (d) CIFAR-10

局模型的测试正确率处于一种震荡的状态. FedCurv 的表现基本与基准算法相当, 虽然它使用 EWC 算法通过缓解局部模型训练时的灾难性遗忘一定程度上减轻了数据异构带来的影响, 但由于没有将全局参数和客户端个性化模型相互分离, 仍然无法避免协作过程中局部模型之间的相互干扰. 而且 FedCurv 的重要性权重完全由上一轮的训练结果确定, 一方面在客户端仅能间歇参与训练的联邦学习中不能很好地起到保留全局重要知识的作用, 另一方面每轮对重要性权重的计算和传输会造成两倍于 FedAvg 的通信成本. FedMeta 由于使用了元学习技术, 有利于联邦学习模型的初期训练, 但在训练的中后期效果较为不稳定, 而且二阶导的计算也带来了大量的计算负担.

图 2 显示 APFL 能在少量的训练轮次中将各数据集上的经验损失降到更小值, 实现更平稳、快速的收敛, 在测试集上也表现出了优异的泛化性能, 在所有实验数据集上都能接近甚至超越 IID 数据下集中式训练基线的表现. 实验结果说明 APFL 通过建立个性化模型使得大多数客户端受益, 而其

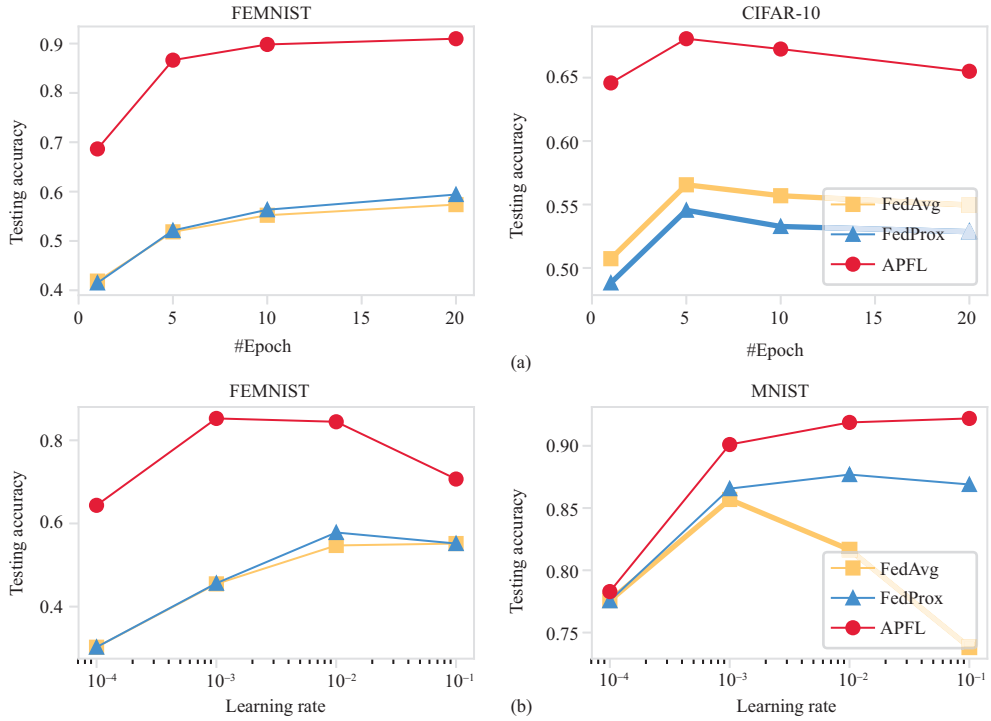

图 3 (网络版彩图) E, η 的值对 APFL 测试正确率的影响

Figure 3 (Color online) Impact of E and η on the average testing accuracy of APFL. (a) Impact of E on FEMNIST and CIFAR-10; (b) impact of η on FEMNIST and MNIST

他对比算法仅训练单一全局模型, 在数据异质性的情况下, 各客户端的模型训练相互之间受到了干扰. APFL 通过对全局知识和客户端特定知识的分解, 抽取到了更为通用的全局知识, 具有更强的泛化能力, 而且借助于持续学习正则项, 全局模型收敛更加稳健. 此外, 我们用 "$\star$" 号标记出了在损失平稳期触发重要性权重更新的时机, 结果显示除了预设的前 3 轮外, 在各个数据集上 APFL 仅需进行一次重要性权重更新, 可见其相比基准 FedAvg 方法, 基本不会带来额外的通信开销.

4.2.2 参数敏感性分析

4.2.1 小节中 FedCurv 和 APFL 算法的对比结果可以大致体现 APFL 对正则化参数 μ 的稳健性. 本小节分析 APFL 对其他重要超参数 E, η 和 λ 的敏感性.

E 和 η 的影响. 全局 epoch 数 E 和学习率 η 是联邦学习算法的两个重要参数. 一个 epoch 指代所有训练数据送入神经网络中完成一次前向计算及反向传播的过程, 因此更大的 E 表示每个通信回合在客户端上执行更多次的梯度更新, 也就意味着更大的局部计算量. 这既可能大大提高通信受限网络的整体收敛速度, 也有可能由于客户端的数据异构导致训练时更关注局部目标的最优, 而损害全局模型的收敛和准确性. 而学习率 η 控制模型的学习进度, 学习率越大, 参数更新的就越快, 同时模型受到异常数据的影响也就越大, 局部模型发散的可能性越大.

为了验证 APFL 算法对于 E 和 η 的敏感性, 分别利用 FEMNIST 和 CIFAR-10 数据集以及 FEMNIST 和 MNIST 数据集展示 E 和 η 的取值对 APFL 算法测试正确率的影响. 分别固定 η 如 4.1.3 小节所示, 设置 E 的备选集为 $\{1, 5, 10, 20\}$; 固定 $E = 5$, 设置 η 的备选集为 $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$. 以基准算法 FedAvg 和 FedProx 的结果作为对比, 实验结果见图 3. 由图可知, APFL 在所有 E

表 2　λ 的值对 APFL 平均测试正确率 (%) 的影响

Table 2　Impact of λ on the average testing accuracy (%) of APFL

Dataset	E	0	0.25	0.5	0.75	1	Mean±Standard
Vehicle	5	88.65	88.69	88.67	**88.71**	88.59	88.66±0.05
	20	88.29	88.24	88.22	88.31	**88.32**	88.28±0.04
MNIST	5	84.62	84.66	84.78	84.93	**85.08**	84.81±0.19
	20	84.81	84.92	85.16	85.42	**85.67**	85.20±0.35

和 η 设置中均能实现最高的预测正确率, 体现出了 APFL 对于全局 epoch 数和学习率选取的稳健性.

λ 的影响. 遗忘系数 λ 代表了对之前通信轮次中全局模型的记忆能力, λ 越大, 越强调对全局参数的记忆. 设置 λ 的备选集为 {0, 0.25, 0.5, 0.75, 1}, 基于 Vehicle 和 MNIST 数据集分析 $E = 5$ 和 $E = 20$ 时, 不同 λ 取值下 APFL 的性能表现, 平均测试正确率的对比结果见表 2. 可以看到 APFL 在很大的参数取值范围内可以取得相近的良好结果, 表明其对 λ 值选取具有稳健性. 而且一般调大 λ 会得到更佳的效果, 尤其是在 E 取较大值时, 因为此时局部计算量较大, 在异构数据下局部模型的学习更有可能偏离全局模型, 因此累积记忆全局参数信息更为必要.

5　结论

本文提出了一种个性化联邦学习算法 APFL, 同时从空间和时间维度优化数据异构下的联邦学习. APFL 基于参数分解策略抽象出全局共享参数并分解出客户端个性化参数, 以缓解客户端之间的模型干扰, 并进一步基于持续学习中的 EWC 算法对全局共享参数进行弹性巩固, 以实现对全局知识的有效保留. 大量联邦学习基准数据集上的实验结果表明, APFL 具有优越的性能, 可以实现快速稳定的收敛, 构建具有高全局正确率的联邦模型. 相比基准 FedAvg 方法, APFL 的平均测试正确率分别提升了 5.41 (Vehicle), 6.01 (MNIST), 30.42 (FEMNIST) 和 10.26 (CIFAR-10) 个百分点, 且其在参数的取值上具有稳健性. 本文工作同时以时间和空间维度的多任务学习看待联邦学习, 通过引入持续学习的思想提升数据异构下联邦学习的算法表现, 为联邦优化问题提供了一种有效探索. 考虑到真实联邦学习环境中受限的计算和通信资源, 下一步研究准备以持续学习思想设计高效的异步联邦学习机制, 并开发超参数自适应优化方案.

参考文献

1　Wang S, Tuor T, Salonidis T, et al. Adaptive federated learning in resource constrained edge computing systems. IEEE J Sel Areas Commun, 2019, 37: 1205–1221

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

3　Wu J J, Liu G N, Wang J Y, et al. Data intelligence: trends and challenges. Syst Eng Theory Practice, 2020, 40: 2116–2149 [吴俊杰, 刘冠男, 王静远, 等. 数据智能: 趋势与挑战. 系统工程理论与实践, 2020, 40: 2116–2149]

4　Yang Q, Liu Y, Chen T J, et al. Federated machine learning. ACM Trans Intell Syst Technol, 2019, 10: 1–19

5　Hu Y C, Niu D, Yang J M, et al. FDML: a collaborative machine learning framework for distributed features. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2019. 2232–2240

6　Li W Q, Milletarì F, Xu D G, et al. Privacy-preserving federated brain tumour segmentation. In: Proceedings of the 10th International Workshop on Machine Learning in Medical Imaging and the 22nd International Conference on

Medical Image Computing and Computer-Assisted Intervention, Shenzhen, 2019. 133–141
7 Feng J, Cai Q Z, Jiang Y. Towards training time attacks for federated machine learning systems. Sci Sin Inform, 2021, 51: 900–911 [冯雾, 蔡其志, 姜远. 联邦学习下对抗训练样本表示的研究. 中国科学: 信息科学, 2021, 51: 900–911]
8 Gao S, Yuan L P, Zhu J M, et al. A blockchain-based privacy-preserving asynchronous federated learning. Sci Sin Inform, 2021, 51: 1755–1774 [高胜, 袁丽萍, 朱建明, 等. 一种基于区块链的隐私保护异步联邦学习. 中国科学: 信息科学, 2021, 51: 1755–1774]
9 Thrun S, Mitchell T M. Lifelong robot learning. Robot Autonom Syst, 1995, 15: 25–46
10 Yoon J, Kim S, Yang E, et al. Scalable and order-robust continual learning with additive parameter decomposition. In: Proceedings of the 8th International Conference on Learning Representations, Addis Ababa, 2020
11 Kirkpatrick J, Pascanu R, Rabinowitz N, et al. Overcoming catastrophic forgetting in neural networks. Proc Natl Acad Sci USA, 2017, 114: 3521–3526
12 Zenke F, Poole B, Ganguli S. Continual learning through synaptic intelligence. In: Proceedings of the 34th International Conference on Machine Learning, Sydney, 2017. 6072–6082
13 Aljundi R, Babiloni F, Elhoseiny M, et al. Memory aware synapses: learning what (not) to forget. In: Proceedings of the 15th European Conference on Computer Vision, Munich, 2018. 144–161
14 Li X, Huang K X, Yang W H, et al. On the convergence of FedAvg on non-IID data. In: Proceedings of the 8th International Conference on Learning Representations, Addis Ababa, 2020
15 Jeong E, Oh S, Kim H, et al. Communication-efficient on-device machine learning: federated distillation and augmentation under non-IID private data. In: Proceedings of the 32nd Conference on Neural Information Processing Systems, Workshop on Machine Learning on the Phone and other Consumer Devices, Montréal, 2018
16 Li T, Sahu A K, Zaheer M, et al. Federated optimization in heterogeneous networks. In: Proceedings of Conference on Machine Learning and Systems, Austin, 2020. 429–450
17 Smith V, Chiang C-K, Sanjabi M, et al. Federated multi-task learning. In: Proceedings of the 31st Annual Conference on Neural Information Processing Systems, Long Beach, 2017. 4425–4435
18 Lin S, Yang G, Zhang J S. A collaborative learning framework via federated meta-learning. In: Proceedings of the 40th IEEE International Conference on Distributed Computing Systems, Singapore, 2020. 289–299
19 Wang H Y, Yurochkin M, Sun Y K, et al. Federated learning with matched averaging, In: Proceedings of the 8th International Conference on Learning Representations, Addis Ababa, 2020
20 Ruvolo P, Eaton E. ELLA: an efficient lifelong learning algorithm. In: Proceedings of the 30th International Conference on Machine Learning, Atlanta, 2013. 507–515
21 Liu Q, Liu B, Zhang Y L, et al. Improving opinion aspect extraction using semantic similarity and aspect associations. In: Proceedings of the 30th AAAI Conference on Artificial Intelligence, Phoenix, 2016. 2986–2992
22 Mitchell T, Cohen W, Hruschka E, et al. Never-ending learning. In: Proceedings of the 29th AAAI Conference on Artificial Intelligence, AAAI 2015 and the 27th Innovative Applications of Artificial Intelligence Conference, Austin, 2018. 2302–2310
23 Ammar H B, Eaton E, Ruvolo P, et al. Online multi-task learning for policy gradient methods. In: Proceedings of the 31st International Conference on Machine Learning, Beijing, 2014. 2949–2957
24 Shoham N, Avidor T, Keren A, et al. Overcoming forgetting in federated learning on non-IID data. In: Proceedings of NeurIPS Workshop on Federated Learning for Data Privacy and Confidentiality, Vancouver, 2019
25 Yao X, Sun L F. Continual local training for better initialization of federated models. In: Proceedings of IEEE International Conference on Image Processing, Virtual, Abu Dhabi, 2020. 1736–1740
26 Lee G, Shin Y, Jeong M, et al. Preservation of the global knowledge by not-true self knowledge distillation in federated learning. 2021. ArXiv:2106.03097
27 Yoon J, Jeong W, Lee G, et al. Federated continual learning with weighted inter-client transfer. In: Proceedings of the 38th International Conference on Machine Learning, Virtual Event, 2021. 12073–12086
28 Casado F E, Lema D, Iglesias R, et al. Concept drift detection and adaptation for robotics and mobile devices in federated and continual settings. In: Proceedings of the 21st International Workshop of Physical Agents, Alcalá de Henares, Madrid, 2020. 79–93
29 Huszár F. On quadratic penalties in elastic weight consolidation. 2017. ArXiv:1712.03847
30 Aljundi R, Kelchtermans K, Tuytelaars T. Task-free continual learning. In: Proceedings of the 32nd IEEE/CVF

Conference on Computer Vision and Pattern Recognition, Long Beach, 2019. 11246–11255
31 Duarte M F, Hu Y H. Vehicle classification in distributed sensor networks. J Parallel Distrib Comput, 2004, 64: 826–838
32 Lecun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition. Proc IEEE, 1998, 86: 2278–2324
33 Cohen G, Afshar S, Tapson J, et al. EMNIST: extending MNIST to handwritten letters. In: Proceedings of International Joint Conference on Neural Networks, Anchorage, 2017. 2921–2926
34 Chen F, Luo M, Dong Z H, et al. Federated meta-learning with fast convergence and efficient communication. 2019. ArXiv:1802.07876

Adaptive personalized federated learning for heterogeneous data: a method based on parameter decomposition and continual learning

Xuanming NI[1], Xinyuan SHEN[1] & Hai ZHANG[2,3*]

1. *School of Software and Microelectronics, Peking University, Beijing* 100871*, China*;
2. *School of Mathematics, Northwest University, Xi'an* 710127*, China*;
3. *Key Laboratory of Advanced Theory and Application in Statistics and Data Science-MOE, East China Normal University, Shanghai* 200062*, China*
* Corresponding author. E-mail: zhanghai@nwu.edu.cn

Abstract Federated learning allows resource-constrained edge computing devices to cooperatively train machine learning models while keeping data locally. However, federated learning faces the challenge that the global model slowly converges and even deviates from the optimal solution under heterogeneous data. To solve this problem, this paper proposes an adaptive personalized federated learning (APFL) algorithm, considering the federation optimization problem for heterogeneous data under a multi-task learning framework that includes spatial and temporal dimensions. First, a parameter decomposition strategy is adopted to decompose the model parameters into globally shared parameters and client-specific parameters to achieve model personalization for each client while extracting general knowledge for all clients. Then, APFL models the local optimization as sequential multi-task learning performed on each client. An elastic weight consolidation penalty term is imposed on the update of the globally shared parameters to realize the memory retention of important parameters and the fast learning of unimportant parameters for the globally shared model. Comparative experiments on multiple federated benchmark datasets verify the effectiveness and superiority of the proposed method.

Keywords federated learning, edge computing, heterogeneous data, multi-task learning, continual learning, parameter decomposition, personalization