

一种对时延敏感的去中心化联邦学习算法

彭姣, 常永娟, 严韬, 游张政, 宋美娜, 朱一凡, 张鹏飞, 贺月, 张博, 欧中洪

引用本文

彭姣, 常永娟, 严韬, 游张政, 宋美娜, 朱一凡, 张鹏飞, 贺月, 张博, 欧中洪. [一种对时延敏感的去中心化联邦学习算法](#)[J]. 计算机科学, 2025, 52(12): 314-320.

PENG Jiao, CHANG Yongjuan, YAN Tao, YOU Zhangzheng, SONG Meina, ZHU Yifan, ZHANG Pengfei, HE Yue, ZHANG Bo, OU Zhonghong. [Decentralized Federated Learning Algorithm Sensitive to Delay](#) [J]. Computer Science, 2025, 52(12): 314-320.

相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

[面向图垂直联邦学习的对抗攻击方法](#)

Adversarial Attack on Vertical Graph Federated Learning

计算机科学, 2025, 52(11A): 241200220-10. <https://doi.org/10.11896/jsjcx.241200220>

[基于知识蒸馏的联邦学习后门攻击方法](#)

Backdoor Attack Method for Federated Learning Based on Knowledge Distillation

计算机科学, 2025, 52(11): 434-443. <https://doi.org/10.11896/jsjcx.250100146>

[基于渐进原型匹配的文本-动态图片跨模态检索算法](#)

Text-Dynamic Image Cross-modal Retrieval Algorithm Based on Progressive Prototype Matching

计算机科学, 2025, 52(9): 276-281. <https://doi.org/10.11896/jsjcx.241200204>

[面向长尾异构数据的个性化联邦学习框架](#)

Personalized Federated Learning Framework for Long-tailed Heterogeneous Data

计算机科学, 2025, 52(9): 232-240. <https://doi.org/10.11896/jsjcx.240700116>

[基于自适应采样的超级传播者检测算法](#)

Super Spreader Detection Algorithm Based on Adaptive Sampling

计算机科学, 2025, 52(8): 393-402. <https://doi.org/10.11896/jsjcx.240900085>

一种对时延敏感的去中心化联邦学习算法

彭姣¹ 常永娟¹ 严韬² 游张政² 宋美娜² 朱一凡² 张鹏飞¹ 贺月¹ 张博¹
欧中洪³

1 国网河北省电力有限公司信息通信分公司 石家庄 050000

2 北京邮电大学计算机学院(国家示范性软件学院) 北京 100876

3 北京邮电大学网络与交换技术全国重点实验室 北京 100876

(P2010015645@163.com)

摘要 近年来,深度学习、移动设备及物联网技术的快速发展,导致在边缘设备上模型推理和数据存储的需求激增。传统的集中式模型训练方法受限于数据量、通信带宽及用户数据隐私等问题,无法有效应对新的挑战。为此,联邦学习技术应运而生。联邦学习允许边缘设备基于本地数据训练模型,并上传模型参数至中央服务器进行聚合与分发,保证数据在不出各方可信域的前提下进行联合建模,并进一步发展了分布式联邦学习以解决时延、带宽限制及单点故障风险等问题。受限于真实网络环境下的网络延迟和带宽等因素,联邦学习的训练效率受到严重影响,造成多方联合建模困难。针对这一问题,提出一种对时延敏感的去中心化联邦学习算法 DBFedAvg,通过动态选择算法选取平均时延较小的节点作为主节点,降低通信成本,提高全局模型训练性能,加速模型收敛。Sprint 网络等场景下的实验结果,验证了所提方法在通信成本和模型收敛性能等方面带来了巨大提升。

关键词: 联邦学习;去中心化;真实网络环境;时延敏感;通信成本

中图分类号 TP391

Decentralized Federated Learning Algorithm Sensitive to Delay

PENG Jiao¹, CHANG Yongjuan¹, YAN Tao², YOU Zhangzheng², SONG Meina², ZHU Yifan², ZHANG Pengfei¹, HE Yue¹, ZHANG Bo¹ and OU Zhonghong³

1 State Grid Hebei Information and Telecommunication Branch, Shijiazhuang 050000, China

2 School of Computer Science(National Pilot Software Engineering School), Beijing University of Posts and Telecommunications, Beijing 100876, China

3 State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China

Abstract In recent years, the rapid development of deep learning, mobile devices, and IoT technology has led to a surge in demand for model inference and data storage on edge devices. Traditional centralized model training methods are limited by data volume, communication bandwidth, and user data privacy issues and cannot effectively address the new challenges. Therefore, federated learning technology is born. Federated learning allows edge devices to train models based on local data and upload model parameters to a central server for aggregation and distribution, ensuring that joint modeling can be performed without data leaving the trusted domain of each party. Furthermore, distributed federated learning has been developed to overcome issues such as latency, bandwidth limitations, and single point of failure risks. However, the training efficiency of federated learning is severely affected by real-world network delay and bandwidth factors, making multi-party joint modeling difficult. To address this issue, this paper proposes a decentralized federated learning algorithm DBFedAvg that dynamically selects nodes with lower average delay as the main nodes to reduce communication costs and improve global model training performance, accelerating model convergence. Experimental results on the Sprint network and other scenarios have validated that the proposed method brings significant improvements in communication costs and model convergence.

Keywords Federated learning, Decentralized, Real network environment, Delay-based, Communication cost

到稿日期:2024-11-13 返修日期:2025-02-21

基金项目:国网河北省电力有限公司(SGHEXT00SJS2310134)

This work was supported by the State Grid Hebei Information and Telecommunication(SGHEXT00SJS2310134).

通信作者:欧中洪(zhonghong.ou@bupt.edu.cn)

1 引言

近年来,随着深度学习技术的不断发展、现代移动设备及终端的大量使用,以及物联网等技术日渐成熟,在边缘设备(客户端)上进行模型推理、数据存储的需求日益增长。由于深度学习神经网络仍然受到学习效率的限制,它需要通过大量的数据训练才能得到更有效的模型,因此如谷歌、微软等公司在提供人工智能服务前,需要收集大量的数据,从而训练出有效的神经网络。然而,该方法中,边缘设备需要将日益膨胀的本地数据悉数上传到中央服务器上,然后由服务器依据多个终端的上传数据完成模型训练,以此往复,边缘设备的数据不断上传到服务器,服务器不断训练神经网络模型直到模型收敛,最终将神经网络模型分发到各设备终端上。这种数据集中式的模型训练方法首先导致的问题是,当数据量过大时模型训练过慢。

因此,为了加快模型训练速度,学术界和产业界相继提出分布式机器学习,该方法基于数据并行原理,在多个节点进行分布式训练,一定程度上加快了训练速度。但分布式机器学习仍然存在两个问题:1)边缘设备与服务器间传输大量数据,极大地消耗了边缘设备与服务器之间的通信带宽;2)无法保证边缘设备上最为关键的用户数据的隐私问题。

以此为背景,谷歌人工智能研究中心在2016年提出联邦学习^[1](Federated Learning)技术,它已成为目前人工智能领域的一个热门研究方向。联邦学习指在保证数据隐私安全及合法合规的基础上,在联邦学习体系中,边缘设备基于本地数据训练出本地模型,然后向中央服务器传输该模型而非本地数据,从而实现面向特定业务的模型训练。人们采用联邦学习技术,不同终端节点基于本地数据集训练模型后,将模型参数上传到中央服务器,最终由中央服务器完成模型聚合与模型分发,从而实现客户端与服务器协同完成模型训练。

这种特殊的训练体系的优点显而易见——边缘设备不再将数据上传至中央服务器,而是上传模型参数。但是在真实网络环境下,由于联邦学习自身的体系结构设计,所有终端节点均依赖单一的中央服务器协调完成训练,导致系统受限于时延、带宽,且系统承受恶意单点攻击的风险,即该系统完全无法应对中央服务器单点故障问题。面对以上问题,研究人员提出没有中央服务器的去中心化联邦学习方法(Decentralized Federated Learning),客户端与客户端之实现 p2p 通信^[2],并采取加密、降噪等方法加强隐私保护;也有些研究将联邦学习与区块链^[3]结合起来达到完全去中心化,利用区块链技术的上链操作解决隐私与安全问题的。然而,在真实网络环境中,不同客户端之间的网络延迟、带宽等因素会影响联邦学习的模型训练效率,现有的去中心化联邦学习方法并未充分考虑真实网络环境对联邦学习模型训练的影响。

为解决上述挑战,提升真实网络环境下的联邦学习模型训练的效率,本文基于 IPMininet 仿真 Sprint 骨干网络,提出对时延敏感的去中心化联邦学习算法 DBFedAvg。在 FedAvg 的基础上,DBFedAvg 基于时延动态选择平均时延较小的节点作为主节点,以降低联邦学习系统中传输模型参数的通信成本,从而提高联邦学习框架中全局模型的训练性能,加速模型收敛。

2 相关工作

在联邦学习的概念被引入学术界之前,已经出现许多与联邦学习相关的工作和研究。在多方拥有数据的场景下,一些密码学和隐私计算的研究团队的目标是,在充分保护用户数据隐私的前提下进行数据的加密计算。尽管联邦学习已经作为一个概念被提出,但确实没有任何独立的研究领域能够完全解决联邦学习涉及的所有问题与挑战。因此,广义的联邦学习并非与元学习、迁移学习或小样本学习等特定的深度学习技术相同,而是一个相对复杂且多元化的领域,涵盖了算法效率、安全性、公平性、隐私保护等多维度的综合性问题。

联邦学习的概念最早由 McMahan 等^[1]于2016年提出,具体指由中央服务器收集边缘设备(客户端)的本地模型而非数据,从而完成模型学习任务。同时该项工作提出了经典联邦学习算法 FedAvg。在传统联邦学习的基础上,部分学者提出能够进一步保护用户数据隐私、规避单点故障风险的去中心化联邦学习算法。下文将从传统联邦学习和去中心化联邦学习两个方面阐述国内外研究现状。本文的主要创新点如下:

1)基于 IPMininet 仿真 Sprint Network Topology 网络拓扑,并通过部署和测试经典联邦学习算法 FedAvg,开拓性地验证了该仿真设计的可行性;

2)提出对时延敏感的去中心化联邦学习算法框架 DBFedAvg,在训练过程中,集群基于时延动态选择平均时延较小的节点作为主节点,以降低联邦学习系统中传输模型参数的通信成本;

3)实现 DBFedAvg 算法框架,并部署到网络拓扑中,基于多种链路设置完成和 Gossip Learning 的对比实验,验证了该方法能显著提高联邦学习框架中全局模型的训练性能。

2.1 传统联邦学习

传统的联邦学习以 FedAvg 算法为代表,致力于训练出一个性能卓越的全局模型,以供中心服务器进行部署和应用。McMahan 等^[1]首次提出了基于服务器-客户端框架的联邦学习,在该框架中,服务器负责接收来自各参与方的模型参数,进行聚合和平均处理,随后将更新后的全局模型广播回各参与方。为了解决现实场景中设备、数据异质性问题,Li 等^[4]在 FedAvg 算法的基础上提出 FedProx 算法,通过引入近端项来平衡全局模型与本地模型之间的差异,提高算法在异构网络中的稳定性。Karimireddy 等^[5]提出 SCAFFOLD 方法,其旨在估计全局模型与客户端模型之间的更新方向差异,并通过减小这种差异来优化算法的更新过程。为了进一步提升联邦学习的性能,研究人员更加关注联邦学习过程中降低通信成本、加速模型收敛等优化问题。Jiang 等^[6]提出了周期性量化平均(PQASGD)策略,该策略在维持模型收敛性的同时,有效降低了通信成本。Yu 等^[7]通过增加本地迭代更新的次数来减少通信量,进一步提高了联邦学习的效率和性能。Basu 等^[8]和 Tang 等^[9]的研究都集中在通过压缩梯度来降低联邦学习每轮通信中的开销。

综上所述,传统的联邦学习方法都需要借助中央服务器

协调各方客户端来完成模型训练,主要聚焦于为服务器端构建一个泛化能力强的单一全局模型。但在真实网络环境下,由于带宽、时延受限,传统联邦学习方法的训练性能低下,并且完全无法规避服务器单点故障风险;此外,在数据异构性较强的场景下,尽管单一全局模型的泛化能力较强,但在客户端本地可能并不总是表现出色。

2.2 去中心化联邦学习

传统的联邦学习方法总是需要借助中央服务器协调各方来完成训练,因此学者们提出了去中心化联邦学习方法。Kuo 等^[10]首次提出了一个去中心化的机器学习框架,该框架巧妙地结合了区块链技术以保护用户数据隐私。Weng 等^[11]基于区块链技术设计了 DeepChain 框架,通过精心构建的激励机制,确保了多方协作训练过程中的数据隐私和公平性。Tran 等^[12]通过优化通信协议和降低通信带宽,提出无需依赖可信的第三方服务器的 SDTF 框架,该框架能在去中心化的网络环境中有效运行。Roy 等^[13]首次提出了一种无服务器的、基于 p2p 通信的联邦学习框架——BrainTorrent,但并未充分考虑用户数据隐私保护问题,导致客户端通信时存在隐私泄露风险。Warnat-Herresthal 等^[3]结合边缘计算,实现基于区块链的点对点网络,并提出一种去中心化的机器学习框架 Swarm Learning。尽管这些算法框架在多种去中心化场景中有效,但并不支持并行处理,客户端之间的模型更新几乎只能逐步进行,这在实际应用中可能导致系统存在较高延迟。为了降低联邦学习框架中的网络延迟,Wang 等^[14]提出了 MATCHA 框架,在客户端之间搭建相互通信的关键静态链接,这一定程度地提高了通信效率,但搭建静态链路的方法无法适用动态变化的真实网络环境。Lalitha 等^[15]提出了一种分布式学习算法,其中客户端通过聚合来自其单跳邻居的模型信息来更新全局模型,最终得到适合整个网络拓扑结构的神经网络模型。He 等^[16]提出的 SpreadGNN 模型实现了图神经网络上的联邦多任务学习,且完全摒弃了对中央服务器的单点依赖。Hegedűs 等^[17]提出不需要中央服务器或任何中央组件的 Gossip Learning 算法框架,完全避免了服务器单点故障问题,但通信效率较低。Tian 等^[18]通过模型分割打破异构联邦学习的壁垒,保障了训练进度和准确性。Zhang 等^[19]通过对联邦学习的潜在漏洞进行攻击,进一步表明了安全协议的重要性。

去中心化联邦学习算法通过剔除中央服务器的方式,有效避免了单点故障问题,但几乎不考虑真实网络环境下,由于链路时延、带宽受限等因素带来的高延迟问题。因此,现在仍需要一种充分考虑真实网络环境的去中心化联邦学习算法框架,并期望这种方法能有更低的通信成本和更高的训练效率。

3 方法

3.1 基于 IPMininet 的网络拓扑仿真及可行性分析

目前联邦学习方面的研究成果百花齐放,但仍然缺少对真实网络环境的考虑,业界缺少在相对真实的网络环境中部署联邦学习算法并验证的工作。目前主流的网络仿真工具是 OPNET, NS, GloMoSim 以及 Mininet 等,基于它们可以轻易搭建具有指定时延、带宽、丢包率的链路,构造交换机、路由

器、主机节点,并实现网络拓扑仿真。但本文研究不仅需要构建可以模拟真实网络环境的网络拓扑,还能在仿真网络上部署联邦学习系统,因此本文采用的仿真工具是 IPMininet^[20],它是 Mininet^[21]的扩展,经过大量的研究测试,验证基于 IPMininet 搭建的虚拟网络拓扑能够执行联邦学习模型训练脚本,同时也能仿真真实网络环境。基于 IPMininet 搭建 Sprint 网络^[22],其是由 25 个路由器节点、25 个主机节点和 53 条链路组成的骨干网拓扑,如图 1 所示。图 1 中,红色节点表示路由器节点,绿色线段表示链路,并且各链路的时延值都被标记。一系列实验都将基于此网络拓扑完成。

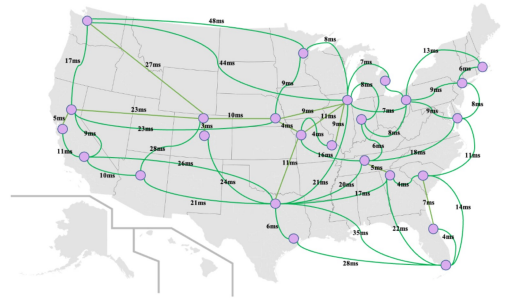


图 1 Sprint 网络拓扑结构(电子版为彩图)

Fig. 1 Topological structure of Sprint network

McMahan 提出的 FedAvg 开源代码并不能部署在 IPMininet 虚拟仿真的真实网络环境中,并且也缺乏能部署在真实网络环境中的 FedAvg 算法。因此,为了验证实验环境的可行性,本文基于 Flask,开发出能够部署在真实网络环境中的 FedAvg 算法,如图 2 所示,其是以 A, B, C 3 个客户端节点为例,同时以 A 节点作为中央服务器的 FedAvg 模型训练。由提交程序初始化全局模型并分发到 A, B, C 客户端节点,以触发集群训练,训练完毕的客户端将本地模型参数传输给中央服务器 A,模型聚合后,再由服务器发回各客户端,并开启下一轮模型训练。以此往复,最终完成联邦学习的模型训练。

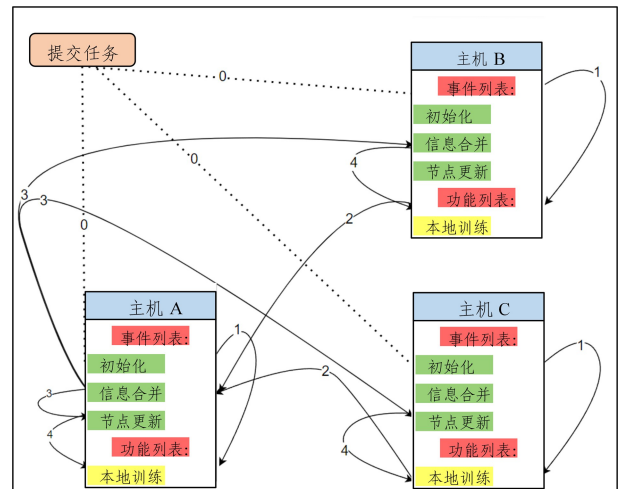


图 2 FedAvg 算法的训练框架

Fig. 2 Training framework of FedAvg algorithm

本工作基于 IPMininet 构建复杂且纵横交错的 Sprint 网络, Sprint Backbone Network 中共有 25 个主机节点,依次标记为 h1-h25,经过 pingall 指令,各节点确实能够成功连接并支持接发数据,并且客户端链路之间设置的时延有效。

客户端节点 k 到其他 $N-1$ 个客户端节点的平均总时延的计算式如下:

$$avg_delay_k = \frac{1}{N-1} \sum_{i=1}^N ping(node_i), i \neq k \quad (1)$$

从中选出 9 个主机节点作为客户端节点,分别是 h1, h3, h5, h7, h10, h12, h14, h16 和 h18。经过初步测试,各个主机节点的平均总时延如表 1 所列。

表 1 各主机节点的平均时延

Table 1 Average delay of each host node

主机节点	平均时延
h1	94
h3	88
h5	72
h7	79
h10	51
h12	68
h14	58
h16	62
h18	63

表 1 中各客户端的平均时延都是基于多次测试取平均值得到,可以发现,h1 的平均时延最大,h10 的平均时延最小。FedAvg 算法中存在固定且单一的中央服务器,其功能是检验 IPMininet 构建的 Sprint 网络拓扑能否模拟由时延、带宽等真实网络环境因素造成的各个客户端节点之间模型参数、平均时延等信息传输时对整体模型训练时间的影响。本文设计了两种方案,分别是 PingMax 和 PingMin,分别对应选择平均时延最大的 h1 节点作为中央服务器和选择平均时延最小的 h10 节点作为中央服务器,比较这两种选择的平均每轮训练所耗费的时长。

验证实验的数据集选用了 CIFAR10^[23],其最初由 CIFAR 研究者和合作伙伴在 20 世纪 90 年代创建,用于开展计算机视觉和模式识别方面的研究。当时,CIFAR 的科学家致力于推动图像识别的进展,并创建了这个数据集,以促进算法研究和评估。CIFAR10 是用于机器视觉领域的图像分类数据集,包含飞机、汽车、鸟类、猫、鹿、青蛙等 10 类别的 60 000 张彩色图像,尺寸均为 32×32 。将这 60 000 张图像划分为 50 000 张训练集和 10 000 张测试集,50 000 张训练集的划分采用随机独立抽取的方式,将 CIFAR10 独立同分布地划分到 10 个客户端上。

在 Sprint 网络上,部署两种不同中央服务器选择的 FedAvg 算法,并比较两种方案的平均每轮训练所耗费的时长,具体结果如表 2 所列。

表 2 PingMax 和 PingMin 的对比实验

Table 2 Comparative experiments of PingMax and PingMin

链路参数设置	PingMax/s	PingMin/s	Improvement/%
10 MB+0.001 ms	266.91	254.67	4.59
10 MB	267.95	256.09	4.43
20 MB+0.001 ms	182.55	175.08	4.09
20 MB	184.33	175.40	4.85
40 MB+0.001 ms	146.28	139.22	4.83
40 MB	145.37	138.01	5.06
100 MB+0.001 ms	128.23	120.42	6.09
100 MB	129.36	121.68	5.94
Default	113.71	104.63	7.99

Sprint 网络中 25 个红色节点之间的 53 条绿色链路的时延值如图 1 所示,因此仅需设置 53 条链路之间的带宽、抖动值、丢包率即可。但在此前的一些实验表明,当设置所有链路的丢包率不为 0、链路抖动值较大时,PingMin 方案仍需要超过数小时才能完成一轮全局训练,这对验证实验毫无帮助,因此在链路设置中均设置 0 丢包率以及较小的抖动值。10 MB+0.001 ms 表示全链路的带宽值均设置为 10 MB,链路的抖动值均设置为 0.001 ms;10 MB 表示仅设置全链路带宽值为 10 MB;Default 表示不额外设置链路的网络属性;其余链路设置方案不再赘述。每组对比实验的提升率计算式如下:

$$improvement = \frac{PingMax - PingMin}{PingMax} \quad (2)$$

为更加清晰地认识到不同链路带宽设置对传输速率的影响,本节将给出不同链路带宽设置下,两个客户端节点之间的上传和下载速率。通过使用 IPMininet 自带的 iperf 命令,帮助快速获得任意两个客户端节点之间的传输速率。

当全局链路带宽设置为默认值时,iperf 大部分的两个节点之间的传输速率均不同,如图 3 所示,h9 和 h10 之间的链路传输速率达到 6.16 Gbps,h1 和 h18 之间的传输速率仅有 416 Mbps。但对比仅 20 MB 大小的模型参数,默认设置的传输速率足够大。

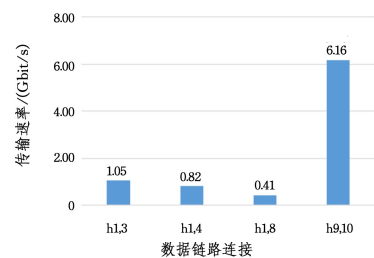


图 3 默认带宽的传输速率差异

Fig. 3 Difference between default bandwidth transfer rates

从表 2 中的实验结果可以发现,PingMin 方案选择平均时延最小的节点作为中央服务器,在不同的链路带宽设置下,平均每轮模型训练所耗费的时间均短于 PingMax 方案,相较于 PingMax 方案,最低有 4% 以上的提升。

在不同链路设置下,客户端本地模型迭代指定次数的训练时间相差不大,这是因为本地模型训练快慢仅与 GPU 等硬件性能相关,而与带宽大小的设置无关。当链路带宽较大时,平均每轮全局训练耗费的时间较短,这是因为在更大带宽下,传输 ResNet18 模型参数耗费的时间更短。从 10 MB 到 Default 链路带宽设置,带宽越来越大,提升也越来越高,这是因为带宽设置越大,模型参数经过每条链路的耗时越短,则一次参数传输的耗时更短,相较于不变的模型训练时间,一轮训练耗费的占比变大。

综上所述,基于 IPMininet 仿真的虚拟网络拓扑能够反映真实网络环境中链路时延、带宽等网络因素,因此基于 IPMininet 的一系列网络拓扑仿真具有很强的可行性。此外,选择平均时延较小的节点作为中央服务器,联邦学习模型训练的性能会更高。开拓性地基于 IPMininet 完成联邦学习部署与训练。

3.2 对时延敏感的去中心化联邦学习算法 DBFedAvg

以 FedAvg 为代表,传统联邦学习算法 FedAvg 中,单一固定的中央服务器是必需的,这一点也在 2.1 节中有所体现, PingMin 和 PingMax 两种方案都需要固定的中央服务器协调模型训练。而在真实网络环境中,采用固定单一的中央服务器导致系统无法承受服务器单点故障问题,此外受限于网络环境中多变的时延和带宽,固定的中央服务器到各个客户端的总时延较大,这将导致整体联邦学习模型的收敛速度较慢,模型训练性能低下。

传统联邦学习算法的中央服务器是必需的,但在现实中很难找到一个所有客户端都信任的、可靠的、公平的中央服务器。此外,固定的中央服务器凭借其在集群中的特殊性和唯一性,可能与一些客户端串通,泄露其他设备的隐私。因此,去中心化联邦学习在联邦学习领域引起了广泛研究。八卦学习(Gossip Learning)被提出作为联邦学习的替代,在基于 p2p 通信的八卦学习算法中,所有终端节点都是独立的个体,当某终端完成本地训练后,随机拉取集群中其他终端节点训练完成的模型参数,同时自身本地模型可被其他终端节点拉取。可以发现,在八卦学习的整个训练过程中,自始至终没有主节点协调和推进模型训练,这造成集群处于紊乱的状态。此外,八卦学习的收敛需要较长的时间,模型训练性能低下。

算法 1 DBFedAvg 算法

```

1. /* 去中心化联邦学习算法 */
2. 初始化:  $(t_k, \omega_k) \leftarrow \emptyset$ 
3. LocalTrain( $t_k, \omega_k$ ):
   /* 本地训练,  $D_k$  是结点  $\epsilon_k$  上的数据 */
4. 参数更新:  $(t'_k, \omega'_k) \leftarrow \text{update}((t_k, \omega_k), D_k)$ 
5.   if IsLeader() = True then
6.     receive( $N_i, \omega'_i$ ) from  $\epsilon_i$ 
       /*  $N_i$  是节点  $\epsilon_k$  上的数据的数目 */
7.      $\omega' \leftarrow \sum_{i=1}^N \frac{N_i}{N} \omega'_i$ 
8.     send( $t'_k, \omega'$ ) to  $\epsilon_i$ 
9.   else
10.    send( $N_k, \omega'_k$ ) to LeaderNode
11. IsLeader():
12.  avg_delay  $\leftarrow \frac{1}{N} \sum_{i=1}^N \text{ping}(\epsilon_i)$ 
13.  vote  $\leftarrow 0$ 
14.  for  $\epsilon_i$  in NodeList do
15.    if getvote(avg_delay,  $\epsilon_i$ ) = True then
16.      vote  $\leftarrow$  vote + 1
17.    if vote = 1 then
18.      return True
19. else
20.  return False

```

去中心化联邦学习算法通过剔除中央服务器的方式有效避免了单点故障问题,部分研究提出基于区块链的去中心化联邦学习算法,根据 CAP 理论,虽然区块链很大程度上能够确保客户端数据的安全性、全局模型的一致性,但是区块链本身缺少可用性,这会导致整体训练性能低下。因此,现在仍需要一种充分考虑真实网络环境的去中心化联邦学习算法

框架,而且期望这种方法可以有更低的通信成本和更高的训练效率。

综上所述,本文提出了对时延敏感的去中心化联邦学习算法 DBFedAvg,客户端本地模型训练部分和 FedAvg 相同,不同的是每个客户端都会自我选举为中央服务器,将本地客户端到集群中其他客户端的平均总时延发送给协调服务器。每轮全局训练由协调服务器开启,它会根据集群中所有客户端的时延数据,比较出最小平均时延的客户端,并通知该客户端充当负责完成本轮训练的中央服务器,这样集群中的所有客户端都有机会成为中央服务器。此外,集群中各个客户端都在协调服务器的调整下,以相同的速度推进模型训练,避免了八卦学习中集群长期处于紊乱的状态。

对时延敏感的去中心化联邦学习算法有诸多优点:1)不存在固定单一的中央服务器;2)在真实网络环境中,动态选举平均时延最小的中央服务器,使得模型训练性能较高;3)在模型训练的过程中,借助协调服务器,始终会选举出中央服务器节点协调和推进模型训练。在后续第 4 章的实验与分析中,会介绍 DBFedAvg 相关实验,从而验证 DBFedAvg 算法的诸多优点。

4 实验与分析

4.1 Gossip Learning 实验

本文已在 2.1 节中介绍 Sprint 网络拓扑,因此不再赘述。本节是本职工作复现前人提出的 Gossip Learning 算法部署在 Sprint 网络上的测试,同样取 10 个节点 h1, h3, h5, h7, h10, h12, h14, h16, h18, h25, 分别部署 Gossip Learning 脚本,在模型训练过程中,八卦学习的各客户端节点不断进行本地模型训练,完成本地训练后,导出本地模型训练参数,随机地发送给集群中的某个客户端节点。当本地客户端获取到其他客户端节点发送的模型参数后,与先前本地模型参数进行聚合,完成聚合后作为新一轮的本地模型参数,再随机发送到集群中某节点。如此往复,直到各个客户端上的本地模型收敛。

从描述中可以发现,八卦学习不存在固定单一的中央服务器,但是在模型训练过程中,始终没有唯一确定的全局模型,各个客户端在训练过程中,本地模型不受控制地慢慢收敛,并且收敛速度各不相同。训练 300 轮之后,部分客户端上最优的本地模型的最小损失值与最高准确率如表 3 所列。

表 3 部分客户端的最优模型

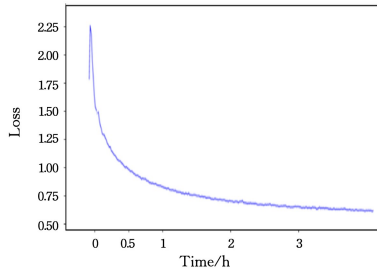
Table 3 Optimal model for some clients

	h1	h5	h10	h12	h16	h25
Min_loss	0.553	0.540	0.549	0.553	0.547	0.550
Max_acc	81.45	81.73	81.25	81.39	81.42	81.17

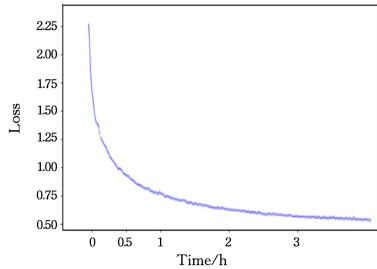
表 3 中可以观测到部分客户端的最优解不一致,反映出在当前学习框架下,去中心化的异步迭代易引发训练紊乱。一旦掉队,客户端就重新并网,其陈旧参数的扩散将降低其余节点的有效进度并延缓全局收敛。

各客户端均有模型训练曲线图,图 4 为 h1, h25 的 loss-time 曲线图。

继续以该组实验为例,记录了部分客户端本地模型准确率第一次达到 81% 所用的时间,如表 4 所列。



(a)h1 节点 loss-time 图



(b)h25 节点 loss-time 图

图 4 h1 和 h25 节点 loss-time 曲线对比

Fig. 4 Comparison of curves between h1 and h25 nodes

表 4 部分客户端耗费时间

Table 4 Time spent by some clients

host	arr_time/h
h1	7.41
h3	7.52
h10	7.46
h12	7.36
h16	7.48
h18	7.46
h25	7.30

八卦学习在整个集群完成 300 轮次的训练后,10 个客户端的结果模型参数均不相同,且各个客户端达到 81% 准确率所用的时间各不相同。以上实验均能证明,八卦学习在模型训练过程中,由于没有中央服务器协调和推进模型训练,集群会处于紊乱的状态。在实际应用中,如果需要应用八卦学习的结果模型,单独选择任何一个客户端的结果模型都是不公平的。

4.2 DBFedAvg 与 Gossip Learning 对比实验

本节将实现第 2 章提出的 DBFedAvg 在 Sprint 网络上的部署,为保证对比实验的公平性,数据集划分、客户端集群节点、本地模型、学习率、链路设置等参数均保持一致。

DBFedAvg 和 Gossip Learning 的一组对比实验曲线如图 5 所示,以集群学习开始时间为初始点,记为 0,每轮全局模型训练完成时间与开始时间的偏移量记为每轮训练的完成时间,经过多轮训练,记录每轮模型训练对应的准确率与损失值。实际上,八卦学习有 10 条曲线图,但为凸显 DBFedAvg 算法在模型收敛方面的高效性,仅取收敛速度最快,即达到 81% 准确率所用时间最短的客户端 h25。

图 5 中,红色曲线对应 DBFedAvg,蓝色曲线对应 Gossip Learning,从图中可以清晰地观察到,DBFedAvg 算法在模型

收敛方面的高效性。由于本组对比实验中两种算法均进行了 300 轮次的训练,DBFedAvg 大概耗时 3.2 h,Gossip Learning 耗时 8.4 h,即使在模型准确率方面二者接近,但在模型收敛速度方面 DBFedAvg 提升显著。

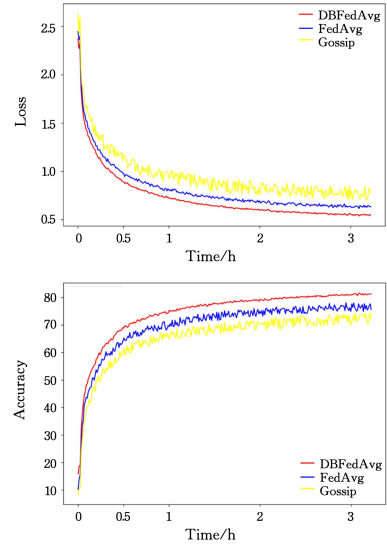


图 5 DBFedAvg 与 Gossip Learning, Fedavg 的收敛速度对比曲线 (电子版为彩图)

Fig. 5 Comparison curves between DBFedAvg, Gossip_learning and Fedavg

仍然基于该组对比实验,DBFedAvg 在中央服务器的协调下,以更高的性能完成模型训练,全局模型准确率第一次达到 81% 时,耗费 2.76 h,相较于八卦学习中最快的客户端节点 h25 的 7.3 h,缩短了 62% 的训练时间,在这一方面 DBFedAvg 的提升显著。

在上述实验基础上,本文基于不同链路设置进行了多次实验,具体结果如表 5 所列。由表 5 可知 DBFedAvg 在模型训练方面的高效性。

表 5 DBFedAvg 与 Gossip Learning 的对比实验

Table 5 Experiments between DBFedAvg and Gossip Learning

链路参数设置	Gossip/h	DBFedAvg/h	Improvement/%
10×10^7	12.12	3.96	67.33
20×10^7	10.85	3.54	67.37
40×10^7	9.63	3.32	65.52
1×10^8	8.74	3.07	64.87
Default	7.30	2.76	62.19

4.3 复杂网络环境

使用 3 台服务器作为客户端节点,衡量现实环境下算法的表现,DBFedAvg 表现出 2% 的性能提升。关于该算法在更实际复杂网络环境下的稳定性和泛化能力,后续将进行进一步验证。

结束语 本文针对传统联邦学习面临的单点故障风险以及高通信成本和低训练效率问题,创新地提出了动态选择平均时延最小的客户端完成中央服务器职能,在每轮本地训练结束后,由通信成本最低的客户端完成模型聚合和分发。实验结果表明,该方法能够降低通信成本,加速模型收敛。此外,还验证了基于 IPMininet 仿真网络拓扑并部署联邦学习算法的可行性,该方法是开拓性的。

在下一步的工作中,将继续关注和提高联邦学习模型的训练效率,并进一步提高联邦学习算法框架的安全性。

参 考 文 献

- [1] MCMAHAN B, MOORE E, RAMAGE D, et al. Communication-efficient learning of deep networks from decentralized data [C] // Artificial Intelligence and Statistics. PMLR, 2017: 1273-1282.
- [2] VANHAESEBROUCK P, BELLET A, TOMMASI M. Decentralized collaborative learning of personalized models over networks [C] // Artificial Intelligence and Statistics. PMLR, 2017: 509-517.
- [3] WARNAT-HERRESTHAL S, SCHULTZE H, SHASTRY K L, et al. Swarm learning for decentralized and confidential clinical machine learning [J]. Nature, 2021, 594(7862): 265-270.
- [4] LI T, SAHU A K, ZAHEER M, et al. Federated optimization in heterogeneous networks [C] // Proceedings of Machine Learning and Systems. 2020: 429-450.
- [5] KARIMIREDDY S P, KALE S, MOHRI M, et al. Scaffold: Stochastic controlled averaging for federated learning [C] // International conference on machine learning. PMLR, 2020: 5132-5143.
- [6] JIANG P, AGRAWAL G. A linear speedup analysis of distributed deep learning with sparse and quantized communication [C] // Proceedings of the 32nd International Conference on Neural Information Processing Systems. Red Hook, NY: Curran Associates Inc., 2018: 2530-2541.
- [7] YU H, YANG S, ZHU S. Parallel restarted SGD with faster convergence and less communication: Demystifying why model averaging works for deep learning [C] // Proceedings of the AAAI Conference on Artificial Intelligence. 2019: 5693-5700.
- [8] BASU D, DATA D, KARAKUS C, et al. Qsparse-local-SGD: Distributed SGD with quantization, sparsification and local computations [J]. IEEE Journal on Selected Areas in Information Theory, 2020, 1(1): 217-226.
- [9] TANG H, YU C, LIAN X, et al. Doublesqueeze: Parallel stochastic gradient descent with double-pass error-compensated compression [C] // International Conference on Machine Learning. PMLR, 2019: 6155-6165.
- [10] KUO T T, OHNO-MACHADO L. Modelchain: Decentralized privacy-preserving healthcare predictive modeling framework on private blockchain networks [J]. arXiv:1802.01746, 2018.
- [11] WENG J, WENG J, ZHANG J, et al. Deepchain: Auditable and privacy-preserving deep learning with blockchain-based incentive [J]. IEEE Transactions on Dependable and Secure Computing, 2019, 18(5): 2438-2455.
- [12] TRAN A T, LUONG T D, KARNJANA J, et al. An efficient approach for privacy preserving decentralized deep learning models based on secure multi-party computation [J]. Neurocomputing, 2021, 422: 245-262.
- [13] ROY A G, SIDDIQUI S, PÖLSTERL S, et al. Braintorrent: A peer-to-peer environment for decentralized federated learning [J]. arXiv:1905.06731, 2019.
- [14] WANG J, SAHU A K, YANG Z, et al. MATCHA: Speeding up decentralized SGD via matching decomposition sampling [C] // 2019 Sixth Indian Control Conference (ICC). IEEE, 2019: 299-300.
- [15] LALITHA A, KILINC O C, JAVIDI T, et al. Peer-to-peer federated learning on graphs [J]. arXiv:1901.11173, 2019.
- [16] HE C, CEYANI E, BALASUBRAMANIAN K, et al. Spreadgnn: Serverless multi-task federated learning for graph neural networks [J]. arXiv:2106.02743, 2021.
- [17] HEGEDÜS I, DANNER G, JELASITY M. Gossip learning as a decentralized alternative to federated learning [C] // IFIP International Conference on Distributed Applications and Interoperable Systems. Cham: Springer, 2019: 74-90.
- [18] TIAN C, LI L, TAM K, et al. Breaking the Memory Wall for Heterogeneous Federated Learning via Model Splitting [J]. IEEE Transactions on Parallel and Distributed Systems, 2024, 35(12): 2513-2526.
- [19] ZHANG Y, BEHNIA R, YAVUZ A A, et al. Uncovering Attacks and Defenses in Secure Aggregation for Federated Deep Learning [J]. arXiv:2410.09676, 2024.
- [20] OlivierTilmans. IPMininet's documentation! [EB/OL] (2022-05-28)[2024-03-01]. <https://ipmininet.readthedocs.io/en/latest/>.
- [21] DE OLIVEIRA R L S, SCHWEITZER C M, SHINODA A A, et al. Using mininet for emulation and prototyping software-defined networks [C] // 2014 IEEE Colombian conference on communications and computing (COLCOM). IEEE, 2014: 1-6.
- [22] MILLS J, HU J, MIN G. Communication-efficient federated learning for wireless edge intelligence in IoT [J]. IEEE Internet of Things Journal, 2019, 7(7): 5986-5994.
- [23] KRIZHEVSKY A, HINTON G. Learning Multiple Layers of Features from Tiny Images [EB/OL]. <http://www.cs.utoronto.ca/~kriz/learning-features-2009-TR.pdf>.



PENG Jiao, born in 1991, master, engineer. Her main research interests include NLP, image processing and big data analysis.



OU Zhonghong, born in 1982, Ph.D., professor, Ph.D supervisor, is a senior member of CCF (No. 69730S). His main research interests include few shot learning, cross domain adaptation and small object detection.

(责任编辑:柯颖)