工程科学学报
Chinese Journal of Engineering

面向边缘智能的协同训练研究进展

王睿 王岩 尹朴 齐建鹏 孙叶桃 李倩 张易达 张梅奎

Survey of edge–edge collaborative training for edge intelligence

WANG Rui, WANG Yan, YIN Pu, QI Jian–peng, SUN Ye–tao, LI Qian, ZHANG Yi–da, ZHANG Mei–kui

引用本文:
王睿, 王岩, 尹朴, 齐建鹏, 孙叶桃, 李倩, 张易达, 张梅奎. 面向边缘智能的协同训练研究进展[J]. 工程科学学报, 2023, 45(8): 1400–1416. doi: 10.13374/j.issn2095–9389.2022.09.26.004
WANG Rui, WANG Yan, YIN Pu, QI Jian–peng, SUN Ye–tao, LI Qian, ZHANG Yi–da, ZHANG Mei–kui. Survey of edge – edge collaborative training for edge intelligence[J]. *Chinese Journal of Engineering*, 2023, 45(8): 1400–1416. doi: 10.13374/j.issn2095–9389.2022.09.26.004

在线阅读 View online: https://doi.org/10.13374/j.issn2095–9389.2022.09.26.004

工程科学学报, 第 45 卷, 第 8 期: 1400−1416, 2023 年 8 月
Chinese Journal of Engineering, Vol. 45, No. 8: 1400−1416, August 2023
https://doi.org/10.13374/j.issn2095-9389.2022.09.26.004; http://cje.ustb.edu.cn

面向边缘智能的协同训练研究进展

王　睿[1)✉], 王　岩[1)], 尹　朴[1)], 齐建鹏[1)], 孙叶桃[1)], 李　倩[1)], 张易达[1)], 张梅奎[2)✉]

1) 北京科技大学计算机与通信工程学院, 北京 100083　2) 中国人民解放军总医院, 北京 100039
✉通信作者, 王睿, E-mail: wangrui@ustb.edu.cn; 张梅奎, E-mail: zmk301@126.com

摘　要　随着万物互联时代的快速到来, 海量的数据资源在边缘侧产生, 使得基于云计算的传统分布式训练面临网络负载大、能耗高、隐私安全等问题. 在此背景下, 边缘智能应运而生. 边缘智能协同训练作为关键环节, 在边缘侧辅助或实现机器学习模型的分布式训练, 成为边缘智能研究的一大热点. 然而, 边缘智能需要协调大量的边缘节点进行机器模型的训练, 在边缘场景中存在诸多挑战. 因此, 通过充分调研现有边缘智能协同训练研究基础, 从整体架构和核心模块两方面总结现有的关键技术, 围绕边缘智能协同训练在设备异构、设备资源受限和网络环境不稳定等边缘场景下进行训练的挑战及解决方案; 从边缘智能协同训练的整体架构和核心模块两大方面进行介绍与总结, 关注边缘设备之间的交互框架和大量边缘设备协同训练神经网络模型参数更新问题. 最后分析和总结了边缘协同训练存在的诸多挑战和未来展望.

关键词　云计算; 边缘智能; 协同训练; 边缘计算; 机器学习; 分布式训练
分类号　TP311

Survey of edge–edge collaborative training for edge intelligence

WANG Rui[1)✉], *WANG Yan*[1)], *YIN Pu*[1)], *QI Jian-peng*[1)], *SUN Ye-tao*[1)], *LI Qian*[1)], *ZHANG Yi-da*[1)], *ZHANG Mei-kui*[2)✉]

1) School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China
2) Chinese PLA General Hospital, Beijing 100039, China
✉ Corresponding author, WANG Rui, E-mail: wangrui@ustb.edu.cn; ZHANG Mei-kui, E-mail: zmk301@126.com

ABSTRACT　With the rapid arrival of the Internet of Everything era, massive data resources are generated on edge sides, causing problems such as large network load, high energy consumption, and privacy security in traditional distributed training based on cloud computing. Edge computing sinks computing power resources to the edge side, forming a collaborative computing system that integrates "cloud, edge, and end," which can meet the basic needs of real-time operations, intelligence, security, and privacy protection. With the help of edge computing capabilities, edge intelligence effectively promotes the intelligent development of the edge side, which has become a popular topic. Through our research, we found that edge collaborative intelligence is currently in a stage of rapid development. At this stage, several deep learning models are combined with edge computing, and many edge collaborative intelligent processing solutions have exploded, such as distributed training in edge computing scenarios, federated learning, and distributed collaborative reasoning based on technologies such as model cutting and early exit. The combination of a shallow breadth learning system and virtualization technology allows for quick implementation of edge intelligence, which considerably improves service quality and user experience and makes services more intelligent. As a key link of edge intelligence, edge intelligence collaborative training aims to assist or implement the distributed training of machine learning models on the edge side. However, in an edge computing scenario, the distributed training of the model must coordinate several edge nodes, and many challenges remain. Therefore, by fully investigating the existing research foundation of edge intelligent collaborative training, we focus on the challenges and solutions of edge intelligent collaborative training in edge scenarios such as equipment heterogeneity, limited equipment resources, and unstable network

收稿日期: 2022−09−26
基金项目: 国家自然科学基金资助项目(62173158,72004147); 生联网构建云端救治调度关键技术研究项目(221-CXCY-N101-07-18-01)

environments. This paper introduces and summarizes the overall architecture and core modules of edge intelligent collaborative training. The overall architecture mainly focuses on the interaction framework between edge devices. In terms of whether there is a central server role, it can be divided into two categories: parameter server centralized architecture and fully decentralized parallel architecture. The core module of edge intelligent collaborative training mainly focuses on the problem of collaborative training of a large number of edge devices for neural network models to update parameters. In terms of the role of parallel computing in model training, it is divided into data parallelism and model parallelism. Finally, the many challenges and prospects of edge collaborative training are analyzed and summarized.

KEY WORDS　cloud computing; edge intelligence; collaborative training; edge computing; machine learning; distributed training

边缘智能 (Edge intelligence, EI) 是使边缘设备能执行智能算法的能力[1], 是人工智能和边缘计算的结合, 在边缘侧辅助或实现机器学习模型训练与推理的一系列智能化方法[2]. 边缘智能充分利用边缘侧海量的数据和硬件资源, 将边缘计算和机器学习相结合, 使机器学习模型提供的智能服务更加高效、贴近用户、解决人工智能的“最后一公里”问题[3−7].

随着机器学习技术特别是深度学习技术的发展, 利用神经网络技术的智能服务给人类社会带来极大的便利. 然而机器学习算法的复杂度高, 大多数是串行执行, 而单台计算机的存储内存受限, 出现算法分析速度过慢、内存不足等现象, 导致数百万参数的模型可能需要几天的训练时间. 传统的解决方案是基于集群并行计算的云计算框架进行分布式计算, 以解决机器学习算法在处理海量数据时所面临的处理速度过慢和存储容量不足等问题[8]. 但是, 随着万物互联时代的到来, 基于云计算的智能服务面临严峻的挑战. 在数据方面, 由于越来越多的终端设备接入互联网, 海量的数据资源在边缘侧产生, 边缘终端设备产生泽字节 (Zettabyte, ZB) 级别海量数据[9]. 国际数据公司 (International data corporation, IDC) 预测 2025 年将有 416 亿边缘侧设备实现互联, 数据量达 79.4 ZB, 全球物联网数据的 70% 都要在网络边缘处理[10−11]. 在此情况下, 将海量的数据发送到云计算中心会面临实时性差、网络带宽受限和隐私安全问题[12−14]. 在计算方面, 如今物联网的边缘设备需要提供大量且多样的智能感知决策服务, 巨大的计算任务使云计算中心面临能耗高、维护费用高等问题[15].

在网络通信领域, 为提升网络的计算、存储能力和服务的多样性、可靠性、智能性, 边缘智能技术寻求将智能服务从网络核心扩展到边缘侧[16]. 对于终端侧的海量数据集和计算任务, 边缘计算提供最近端服务, 更快服务响应, 满足在实时、智能、安全与隐私保护等方面的基本需求. 实际上, 边缘计算正逐渐与人工智能结合, 在实现边缘智能和智能边缘方面相互受益, 其中边缘智能是目标, 智能边缘可以为边缘智能提供更高的服务吞吐量和资源利用率. 随着云计算能力从中心下沉到边缘, 边缘计算将推动形成“云、边、端”一体化的协同计算体系. 边缘智能利用广泛的边缘资源为人工智能应用提供支持, 而无需全部的云计算中心资源. 边缘计算将机器学习需求的计算、存储等资源下沉至终端侧, 以其低时延、动态性、移动性以及位置感知等特征解决传统云计算中心面临的问题[2], 在工业互联网、智能家居、智慧交通及智能医疗等[6, 17−20] 领域发挥着重要作用. 目前边缘协同智能正处于快速发展阶段[2], 这一时期大量的深度学习模型与边缘计算结合, 众多边缘协同智能处理方案爆发, 如边缘计算场景下的分布式训练[21] 和联邦学习[22], 基于模型切割、早期退出等技术的分布式协同推理[23−24], 浅层的宽度学习系统[25] 与虚拟化技术的结合使边缘协同智能的快速落地成为可能, 提高了服务质量和用户体验, 使服务更加智能化[26−27].

本文首先从现有的分布式机器学习、联邦学习以及边缘智能相关方面调研了边缘智能协同训练的关键技术, 围绕边缘智能协同训练讨论了在设备异构、设备资源受限和网络环境不稳定等边缘场景下进行训练的挑战及解决方案; 然后介绍了边缘智能协同训练的整体架构和核心模块两方面内容: 边缘智能协同的整体架构主要关注边缘设备之间的交互框架, 从有无中央服务器角色的角度分为参数服务器集中式架构和完全分散并行式架构两大类; 边缘智能协同的核心模块主要关注大量边缘设备协同训练神经网络模型参数更新的问题, 从并行计算的角度分为数据并行、模型并行. 此外, 还介绍了边缘场景在参数更新中很常见的数据非独立同分布相关工作和模型压缩中知识蒸馏相关工作; 最后分析和总结了边缘协同训练存在的诸多挑战和未来展望.

1 边缘智能协同训练概述

边缘智能主要研究在边缘侧辅助或实现机器学习模型的训练与推理两个阶段. 研究者在边缘协同推理阶段做了大量的研究, 但对边缘协同训练阶段的相关综述较少, 因而本文聚焦于边缘智能中的训练阶段, 即利用边缘节点间的协同在边缘侧辅助或实现机器模型的分布式训练有关研究进行总结.

1.1 发展历程

传统分布式训练的代表是基于集群并行计算的云计算框架, 用于解决机器学习算法训练在处理海量数据时所面临的处理速度过慢和存储容量不足等问题, 如 UCBerkeleyAMP 实验室研究的开源类 Hadoop 并行框架 ApacheSpark[28] 以及现在主流的两种研究框架——以 TensorFlow[29] 为代表的数据流式架构和以 MXNet[30] 为代表的参数服务器架构(Parameter server, PS), 都在迭代优化计算方面有显著的应用效果[31]. 分布式机器学习 (Distributed machine learning, DML) 需要较少的计算资源, 可以有效地实现许多人工智能应用程序. 然而, 由于动态网络拓扑结构和边缘传输质量的波动, 工作节点的选择对 DML 的性能影响很大[32], 也正是因为多节点的部署、参数的相互协调和节点相互之间的通信, 造成了网络通信的大量开销. 随着物联网(Internet of things, IoT)的发展, 越来越多的终端设备接入, 而这些必须连到云计算中心才能处理数据, 网络负载消耗太大, 响应时间也随之拉长. 加之, 现在很多边缘设备存储的信息涉及用户的隐私数据, 如健康信息、身份密码验证信息等. 与此同时, 边缘设备的计算能力和存储能力都在提高, 在边缘上部署机器学习模型设想上可以很好地解决网络负载大的问题, 且由于在边缘的数据较少, 所以边缘计算可以解决隐私安全等问题. 但由于边缘设备比云的资源更有限、计算能力小, 且不能按需弹性扩展[33], 这种部署具有挑战性. 值得注意的是, 边缘智能并不是与云计算中心完全对立脱离了关系, 而是“边缘–云协同”的模式, 分布式边缘处理可以不与云中心交换底层数据实现训练工作, 减少了端到端的延迟和网络负载, 降低了云计算中心的运算压力; 而边缘的节点无法承受备份数据的负载时, 云中心又可以提供可扩展的存储[34]. Srivastava 等[35] 研究了 4 种硬件平台上最流行的基于深度神经网络的目标检测模型的利弊, 结果证实了 MobileNet 模型对边缘部署的适用性. Padmanandam 等[32] 提到在边缘使用的机器学习和深度学习算法可以通过对产品评级[36]的情感分析和对利润最大化[37] 的推荐系统来重新定义业务流程, 在边缘设备上收集的数据被分析和处理后用于学习靠近边缘的预测模型. 在边缘上进行人工智能协同训练, 除需要密集处理的数据传送到云服务端外, 其余数据均可在数据源附近进行分析, 具有明显的优势[38]: (1) 基于用户特定的行为及需求的模型个性化可以更有效与学习用户直接交互的设备进行可扩展; (2) 利用设备上的资源快速动态地调整模型, 可以更好地实现对用户行为和环境变化的响应; (3) 与许多用户共享的公共资源相比, 在用户设备上学习的用户特定信息的隐私可以得到更好的保护.

为了解决数据孤岛问题, 建立更好的训练模型, 同时保护用户隐私, 可以利用联邦学习来解决, 其重点在于保护隐私的同时使用更多的数据建立更全面模型[39]. 近年来, 联邦学习[40] 作为边缘智能协同训练的一项使能技术[3], 吸引了广大研究人员的注意. 联邦学习初步应用是在边缘设备上, 如谷歌的 FedAvg[41] 应用在手机键盘语言预测上, 随后边缘设备上的联邦学习应用逐渐得到扩展. 联邦学习是一种机器学习环境, 多方共同参与, 多个实体在一个中央服务器或服务提供商的协调下合作解决一个机器学习问题, 其中每个客户的原始数据都存储在本地, 不进行交换或传输, 且所有模型的训练在设备本地进行, 本地模型训练完毕后将得到的模型参数加密上传至云端, 云端模型接收所有上传的加密参数后, 统一聚合所有参数值得到新的结果, 然后将新的结果重新下发到本地更新得到一个全新的模型. 联邦学习中跨主机、跨设备训练与边缘智能协同训练有部分相似. 联邦学习强调模型训练过程中对数据拥有方的数据隐私保护, 是一种应对数据隐私保护的有效措施. 而边缘智能协同训练研究辅助或实现机器学习模型训练的一系列智能化方法, 不仅关注数据并行和隐私保护, 还关注模型并行、通信架构等多种技术. 联邦学习可分成横向联邦学习、纵向联邦学习和迁移联邦学习[39,40,42], 分别为边缘智能协同训练在不同机器学习设定下实现数据并行训练提供了对应训练框架. 其中数据源之间的关系可以从特征重叠和用户重叠来进行区分, 当数据源之间在特征上重叠较多, 但用户重叠较少时, 如不同地区的两家银行在对用户属性的记录上存在较多的相同之处, 但由于其处于不同地区, 导致两个

数据源面向的用户存在差异较大的现象，这种类型的数据分布模式称为横向联邦学习[43]；相反，当数据源之间出现特征重叠较少，但用户重叠较多的情况时，如位于同一地区的银行和超市，显然，营业类型的不同将导致两个数据源在特征属性上的不同，但同一地区这一因素使得两个数据源在用户上出现了较大比例的相交，称为纵向联邦学习[44]；联邦迁移学习[45]则是在两个数据源特征重叠较少以及用户重叠较少的情况下，研究人员考虑将迁移学习技术引入联邦学习框架中，通过迁移学习来缓解不同学科数据集之间的差异.

1.2　挑战总结

与以云计算为代表的传统分布式训练相比，边缘智能通过边缘计算与人工智能相结合，有显著的特点：

(1) 边缘智能训练更接近用户、数据源和设备[34]. 与云计算相比，从物理角度看，边缘服务器更接近人、数据源和设备；从智能化角度看，边缘计算比云计算有潜力提供更多样化的人工智能应用场景. 作为云计算向网络边缘和终端用户的扩展，边缘智能训练将计算资源和服务从原理用户的云端下沉到网络边缘侧，有效的降低网络延迟和带宽消耗，并且加强了隐私保护.

(2) 低延迟和能源消耗. 将海量的数据从边缘设备发送到云计算中心会面临带宽受限、网络不稳定的情况，将训练过程向边缘侧移动可以不与云中心交换底层数据来实现训练工作，由此减少了端到端的延迟和网络负载，降低了云计算中心的运算压力[34].

(3) 隐私安全[46–47]. 边缘智能协同训练，同样关注联邦学习所关注的隐私保护. 在边缘侧进行训练可以实现在数据不出边缘设备本地的情况下进行训练，保护了数据的隐私安全.

边缘智能协同训练要协调大量的边缘节点进行机器模型的训练，其挑战主要包括：

(1) 设备异构[19,20,48]. 边缘节点的设备异构性体现在设备类型异构性、设备资源异构性、设备数据异构性和设备模型异构性. 高效协同异构设备完成模型训练将面临巨大挑战，其中包括模型训练过程中如何管理异构设备的配置和状态、如何在非独立同分布的设备数据下训练高质量模型、如何训练自适应的个性化模型应对设备模型异构性等挑战.

(2) 设备资源受限[49–53]，边缘节点的设备资源受限体现在单个边缘节点的计算资源有限，需要多个边缘服务器协同训练工作. 设计高效、低开销的模型训练方案以实现在资源受限的边缘节点上完成大规模模型的训练，其中包含如何将一个机器学习模型进行切分并分配到各个协同工作的边缘服务器上、如何加速设备上的训练速度、如何降低开销等挑战.

(3) 网络环境不稳定[54–55]. 处于不同区域的边缘设备的网络拓扑结构、带宽、传播时延等参数存在较大差异，传输过程中会产生波动，造成网络环境的不稳定. 这也说明边缘智能协同训练需要设计高效的通信架构以面对不稳定的网络环境挑战.

2　边缘智能协同训练过程

在本地计算情景下，通过多个边缘设备协同进行全局模型训练有效缓解了传统集中式机器学习在存储、计算和通信上的超高负荷以及隐私安全等方面的痛点，得到了研究人员的广泛关注，但仍存在较多不足. 在边缘计算环境中包含大量不同类型的终端设备，因此各设备在数据、性能以及网络环境上存在较大的差异，同时，由于大量边缘设备呈地理分布，开放的网络环境易导致通信状态不稳定，从而影响协同训练的交互效率，且设备异构、资源受限以及网络环境不稳定对边缘协同的训练效率提出了挑战. 如何高效协调各设备之间的差异性，优化协同训练机制在不稳定环境的鲁棒性是当前亟待解决的问题，将从边缘协同训练的整体架构和核心模块两方面对当前的研究工作进行介绍与总结.

2.1　边缘智能协同训练整体架构

面向大量的边缘设备，设备之间的交互主要分为中央服务器统筹管理与设备之间点对点交互两种方式. 从有无中央服务器的角度，将边缘智能协同训练的整体架构分为参数服务器集中式架构和完全分散并行式架构两大类.

2.1.1　参数服务器集中式架构

参数服务器集中式架构在分布式机器学习以及联邦学习中是一种传统架构，如图 1. 在该架构中，参数服务器可以是云或高性能的边缘设备，对参与协同训练的边缘设备进行协调，并负责通信轮次的模型参数交互以及聚合更新. 参与协同训练的边缘设备，即设备终端，可以是一个个被分配数据的小数据中心或自行生成数据的移动设备. 基于本地数据集进行通信轮次中的本地模型更新，可以看出参数服务器集中式架构是在参数服务器的聚合模型更新和客户端的本地模型更新之

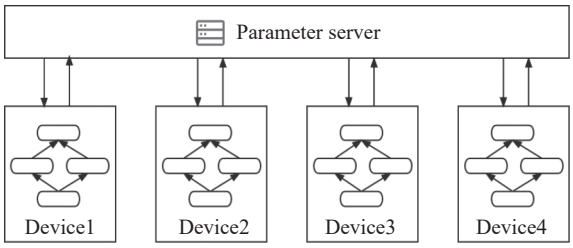

图 1　参数服务器架构

Fig.1　Parameter server architecture

间的多轮通信交互之下, 以得到最终模型的集中式架构, 因此参数服务器的集中式协调具备实现简单、易于运维以及资源相对集中等特点.

参数服务器集中式架构中, 参数服务器与客户端之间的具体交互方式对提高边缘协同训练的模型性能起到了关键作用. 根据不同的通信协调机制, 参数服务器与客户端之间的交互主要包括同步和异步两种. 针对协同训练场景下不同的实际问题, 已有多数研究人员对这两种不同的交互方式展开实验与研究. 这些研究可以概括为整体架构中的设备层次和通信层次两个方面: 设备层次包括了参数服务器以及参与聚合训练的客户端设备; 通信层次是指参数服务器与客户端交互过程中涉及到的网络环境以及通信成本等, 分别从参数服务器同步和参数服务器异步两种方式进行描述.

在参数服务器同步方面, 参数服务器在每个通信轮次会对选中的客户端统一传递全局模型, 以保证客户端进行本地更新前的模型全局一致性, 并等待选中的客户端均将本地更新模型上传至参数服务器之后, 再进行聚合. McMahan 等[41]面向系统中的网络资源受限问题, 考虑在设备层次上从提高本地模型更新并行度的角度, 将客户端本地更新单迭代次数替换为适当的多迭代次数, 通过提高本地模型质量和减少整体通信轮次来加快全局模型的收敛速度. 同样针对有限的网络资源问题, 为保证资源约束下的模型收敛性, Hsieh 等[56]利用实验证明, 模型更新的参数并不是全部都具有代表性, 大部分更新对模型的性能贡献微小, 因此, 作者提出为模型上传聚合的更新度提供一个可调阈值, 仅当本地模型的更新变化达到指定阈值之上以后才可上传至参数服务器参与聚合, 从通信层次上加以限制, 以避免因网络传输数据的冗余性带来的通信资源上不必要的消耗. 增加本地模型的迭代次数实现了很好的实验效果, 但仅关注在单个客户端上的优化, 忽略了多个客户端之间的异构性, 在由大量移动设备组成的边缘计算环境中, 设备异构性是一个不可忽略的主要问题. 因此, 在文献 [41,57] 的基础上, 对设备之间的异构性进行了进一步的探索, 考虑到因较慢客户端造成的“落后者”问题, 均在设备层次上对较慢客户端采取一定的策略, 以避免因“落后者”而拉长通信回合长度带来的损失[58−59]. Wang 等[60]对边缘计算环境中通信资源受限和设备异构性两方面进行了综合考虑, 以在有限资源约束下克服设备异构性并提高全局模型收敛速度为目标, 通过协调局部更新与全局聚合之间的步长, 动态调整全局聚合频率, 对有效利用资源与提高通信效率进行权衡. 边缘计算环境中除资源受限和设备性能异构外, 数据资源不平衡以及网络环境不稳定等问题也影响着整体架构的性能, Hsieh 等[56]在通信层次上分别从低秩矩阵和稀疏矩阵两个方面提出两种解决策略, 以降低上行链路通信量为目标, 提高不稳定系统环境下整体架构的鲁棒性.

在参数服务器异步方面, 参数服务器不再对客户端进行子集同步等待, 而是面向整个客户端集合, 随机接受至一定数量的部分客户端即可进行一个通信轮次的聚合更新. 显然, 异步的通信方式可以有效避免参数服务器同步架构中的较慢“落后者”问题, 但也因此引入了一些新的问题. 由于异步通信不再对客户端的本地模型进行一致性更新, 使得不同客户端因不同的更新频次而出现模型参数新旧版本上的差异, 参数服务器异步架构中对梯度陈旧程度的控制显得尤为重要[61], 否则将对协同训练全局模型的收敛性能造成严重的影响, 该现象被研究者归为一种“散乱者”问题. 在参数服务器异步架构中的设备层次上对“散乱者”现象进行优化, 均从不同的角度对部分超参数进行动态调整, 以适应不同的客户端散乱现象[62−63]. Chen 等[64]除从设备层次上的超参数进行动态调整外, 为参数服务器的聚合阶段引入了一种特征学习新聚合方案, 在参数服务器与客户端的协同控制下, 降低“散乱者”带来的干扰. 另外, 在通信层次上, Dutta 等[65]认为应该将每轮通信时间长度与客户端的散乱程度同时考虑, 以二者权衡的优化角度对全局模型收敛的整体效率进行提升. 以上研究均面向一个静态的客户端全集, 参数服务器异步架构仍多被用于研究动态组成的客户端集合. 比较典型的应用场景为车联网, Lu 等[66]考虑到参数服务器的服务范围有一定的局限性, 在一定的局部服务范围内, 其所面向的参与车辆协同

训练集合将发生动态变化，因此作者根据时长对本地数据计算耗时、参数更新耗时以及优化耗时三方面进行综合考察，并与深度强化学习结合以研究优化规律，得出最近参与子集方案，保证协同训练的准确性．另外，客户端的不可靠性，如存在掉线、崩溃以及主动退出等现象，也将导致客户端子集的动态变化．Wu 等[67]分别从设备层次和通信层次研究出训练后客户端选择与延迟容忍的优化策略，有效提高了参数服务器异步架构的鲁棒性．

表 1 总结了参数服务器集中式架构的相关工作．

表 1　参数服务器集中式架构相关工作

Table 1　Related works of a parameter server with centralized architecture

Communication mechanism	Optimization level	Research questions	Optimization objective	Reference
Synchronization	Equipment level	Limited resources	Improve local model quality	[41]
	Communication level	Limited resources	Reduce traffic	[56]
	Equipment level	Heterogeneous equipment	Shorten communication time	[58–59]
	Equipment level	Comprehensive consideration	Architecture flexibility	[60]
	Communication level	Unstable environment	Architecture robustness	[56]
Asynchronization	Equipment level	Stale gradient	Architecture flexibility	[62–64]
	Communication level	Comprehensive consideration	Trade optimization	[65]
	Equipment level	Dynamic client	Time consuming optimization	[66]
	Overall architecture	Heterogeneous equipment	Architecture robustness	[67]

2.1.2　完全分散并行式架构

参数服务器集中式架构主要依赖于中央服务器来协调整个协同训练流程，但该架构也因中央服务器的存在面临资源瓶颈问题，即当该架构面向大量的客户端时，中央服务器作为一个唯一协调者，扩展性不佳，成为该架构无法顺利完成协同训练的主要瓶颈．分散并行式架构由此出现，如图 2 所示．分散并行式架构的核心思想在于其去除了中央服务器的角色，由各客户端与其他客户端之间协调交互，根据信息的互相传播来实现全局模型的最终收敛，因此又被称为 Gossip 架构或 Peer-to-Peer 架构．

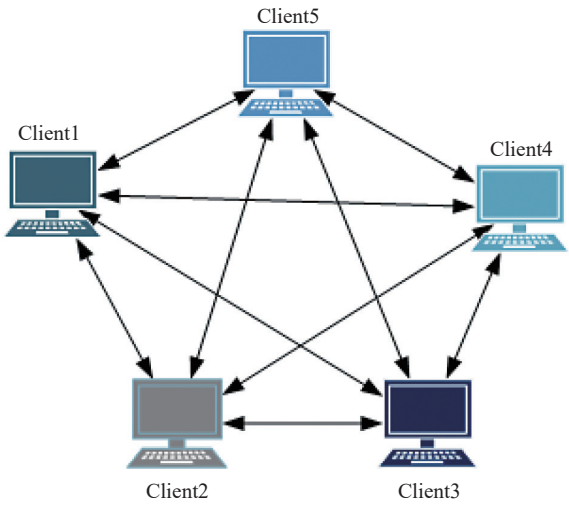

图 2　完全分散并行式架构

Fig.2　Fully decentralized parallel architecture

分散并行式架构可分为 AllReduce-stochastic gradient descent (AllReduce-SGD) 架构和 Dispersed parallel stochastic gradient descent (D-PSGD) 架构两种，二者的区别主要在于如何选择相互交互的客户端上．由于 AllReduce-SGD 架构[68]主要关注在各个客户端模型参数的同步一致性问题，因此该架构要求客户端与其他任一个客户端都完成握手才算完成一个通信轮次的交互，又被称为 all-to-all 方式．这种方式虽然降低了每轮各客户端之间的模型差异性，但交互过于冗余且频繁，忽略了资源受限以及环境不稳定等问题．D-PSGD 架构[69]削减了 AllReduce-SGD 中的冗余交互过程，要求客户端仅向一个或一组选定的节点进行交互即可，通过局部信息的扩散实现全局信息的一致，更适用于大量呈地理分布的边缘协同训练场景．

在分散并行式架构中，由于其去除了对全局协调控制的中央服务器角色，客户端如何与其他客户端之间进行交互以及如何选择最佳通信伙伴等问题也开始浮现，称为邻居交互选择问题．邻居交互选择问题对模型收敛效率起到了关键性作用，目前已有研究从多个不同的角度对邻居交互进行实现与探索，如对邻居交互的实现采用随机选择的方式，每个客户端在进行每轮更新前会随

机选择任一个客户端进行参数平均计算, 通过对称式交互进行协同训练[69−72]; 对于随机选择通信伙伴的通信方式, Daily 等[73] 认为其存在通信不平衡以及梯度不平衡传递等问题, 将导致通信性能差以及收敛速度下降, 作者提出一种分批轮换合作伙伴的邻居交互方式, 并通过理论加以验证, 获取了较好的效果. 同样考虑到邻居交互选择的重要性, 研究者开始从不同的标准对邻居交互方式进行改进. Vanhaesebrouck 等[74] 以寻找具有相似目标的其他客户端为标准, 分别从为客户端增加置信度标签和邻居行为学习两个方面提出两种邻居选择方案, 得到了很好的模型收敛效果. He 等[75] 认为传统的分散并行式多基于对称交互方式, 极其不符合实际场景下的环境不稳定情形, 因此从单信任集合上来实现邻居交互, 提高了分散并行式架构的稳定性. 与前面两者不同, Colin 等[76] 根据节点之间连接边的角度, 对连接边添加权重, 将环状结构扩展为环面结构, 按权重值选择交互客户端.

在边缘设备协同训练的环境中, 资源受限是整体架构面临的一个普遍问题, 分散并行式架构相较于参数服务器集中式架构来说, 由于其中各客户端之间的交互复杂性, 使得对通信资源消耗上的优化与改进显得十分重要. Nedić等[77] 认为对称交互的通信方式是影响通信资源消耗的主要原因, 因此提出一种非对称交互方式来使用多个设备完成协同训练, 并证明了该方式对模型收敛的保证. 在非对称交互方式想法的基础上, Assran 等[78] 引入允许通信和计算量重叠的思想, 对非对称交互的通信方式加以改进, 并在降低通信资源消耗上取得了很好的效果. 另外, 也有研究者开始从改变传递模型参数的角度对通信资源的消耗问题进行优化. Koloskova 等[79] 将模型压缩技术引入分散并行式架构中, 并研究了压缩因子的动态调整策略, 在模型质量以及节省资源上均得到了保证. Hu 等[80] 提出一种模型段级交互方式, 通过结合模型切割技术, 从段级模型参数交互上实现客户端之间的模型参数同步, 使得客户端之间的网络资源得到了充分利用.

表 2 总结了分散并行式架构的相关工作.

2.2 边缘协同训练的核心模块

随机梯度下降法 (Stochastic gradient descent, SGD) 是最常用的优化算法之一, 是迄今为止最常用的优化深度神经网络的方法[81], 也是研究边缘协同训练优化技术的基础算法. 从边缘协同智能模型训练方式的角度, 将边缘协同训练的核心模块分为数据并行和模型并行[82] 两个模块. 其中数据并行介绍了同步更新及异步更新研究现状和边缘协同场景均会遇到的数据非独立同分布问题以及现有的解决方案; 模型并行介绍了模型并行相关工作以及模型压缩中的知识蒸馏训练方式.

2.2.1 数据并行

数据并行指多个参与训练的边缘设备利用各自的本地数据集更新本地模型参数, 并且通过聚合本地模型参数来更新全局模型的训练方式. 数据并行训练过程需要参数服务器参与调控、聚合本地模型参数和更新全局模型等过程, 因此通常在参数服务器集中式架构下进行. 其中, 参与训练的设备节点被称为客户端, 每个客户端上存储着本地数据集和模型参数. 数据并行方式按照参数更新方式可分为同步更新与异步更新.

同步更新作为分布式训练最普遍的方法, 也

表 2 分散并行式架构 (D-PSGD) 相关工作

Table 2 Related works of dispersed parallel stochastic gradient descent

Research questions	Research protocol	Optimization objective	Reference
Neighbor interaction	Random selection	Reduce interaction complexity	[69–72]
	Cooperation by batch rotation	Improve model consistency	[73]
	Look for similar targets	Best communication partner	[74]
	Single trust set	Improve architecture robustness	[75]
	Weight comparison selection	Best communication partner	[76]
Communication consumption	Asymmetric interaction	Avoid redundant communication	[77]
	Overlapping communication computation	Avoid redundant communication and computing	[78]
	Model compression	Make full use of link resources	[79]
	Model cutting	Improve communication flexibility	[80]

是联邦平均算法[41]的基础，适用范围十分广泛. 同步更新要求参数服务器需要等待所有选定的客户端上传完本地模型更新梯度才开始聚合和更新全局模型，Chen 等[58]通过实验观察到在模型更新过程 80% 的梯度在 2 s 以内到达，而最后的梯度只有 30% 到达. 此外，为了等待梯度更新最慢的客户端完成更新，训练时间会呈指数增长以及存在闲置资源的浪费. 本文将不能在规定时间内将本地梯度上传至参数服务器的客户端称为延迟客户端；将在同步更新过程中为了等待延迟客户端造成训练增长的问题称为客户端延迟问题. 同时在边缘协同环境中，设备异构性和资源受限等问题导致客户端延迟问题更加严重. 因此如何设计一个更加鲁棒、稳定收敛的同步更新算法在边缘协同训练中十分重要. 解决客户端延迟问题的一种常见方法是让满足一定条件的客户端才能进行更新. Chen 等[58]提出了使用备份节点辅助同步更新过程，参数服务器只需收到一定数量的返回梯度就开始聚合并在后续丢弃超时梯度. 该方法在一定程度上缓解了资源设备所带来的训练时长问题. 但在边缘协同场景下备份机器的设置和数据异构的问题仍需要进一步研究. 一些研究专注于在同步更新过程主动选择满足训练需求的客户端以缓解客户端延迟问题. Nishio 等[59]中通过过滤响应时间慢的客户端解决了具有资源约束的客户端选择问题，允许服务器聚合尽可能多的客户端更新. 但这些方法适用范围仅考虑到了训练时间的加速，而在数据异构场景下丢弃延迟客户端的参与可能会导致模型的准确率降低. Chai 等[83]同时考虑到资源异构和数据异构对同步更新算法带来的挑战，认为删除较慢的客户端可能会使客户端上可用的某些数据分布无法用于训练全局模型，因此提出了一种基于层的同步更新算法，该算法首先利用分层模块按照训练性能将可用的客户端进行分层，每轮次将训练时间相似的客户端更新以缓解客户端延迟问题中的等待时间，之后利用自适应的层选择算法并根据动态变化的系统条件自适应地调整每轮训练选择各层的概率在训练时间和准确性之间取得平衡. Li 等[84]提出一种混合式更新算法，算法的两个关键组件同步内核和异步更新器分别缓解了在同步更新中的资源异构和数据异构，其中同步内核负责同步非延迟客户端的更新参数，异步更新器负责将落后者的模型更新合并到全局模型训练过程中，并且提出自适应延迟 SGD 将落后者的陈旧局部更新合并到全局模型中. Xu 等[85]认为通过异步边缘协作不能从根本上解决落后者问题，提出了一种基于异构边缘设备的资源感知更新框架，框架中的本地模型更新利用“软更新”进行，即为参与训练的边缘设备屏蔽相应比例的神经元数量进行本地模型更新以适应设备能满足训练需求的更新方式，并通过对应的参数聚合方案来平衡各设备“软更新”对全局模型的贡献，从而达到更好的边缘协同收敛.

异步更新允许客户端无需等待其他客户端完成全局模型的下载和本地模型的上传，同时参数服务器接收到上传的本地模型可以选择立即更新全局模型. 异步训练方法实现了更好的并行性[54]，可以消除同步更新的等待问题. 目前已有研究为异步 SGD 在凸问题和非凸问题上提供了收敛的理论保证[61,86–87]. Dean 等[88]实现了 DistBelief 的分布式并行训练框架以解决数万个 CPU 核训练一个具有数十亿个参数的深度神经网络问题，其中利用了一种异步 SGD 变体 Downpour SGD 算法对模型异步更新，通过评估发现在 2000 个或更少 CPU 核的计算资源时 Adagrad 与异步 SGD 能在深度神经网良好地协同工作. 随后，异步更新算法各种扩展及变体在各种工作[64,72,89–90]中实现. 然而，在异步更新方式下，客户端计算梯度使用的模型参数可能和最新的全局模型参数不同，这时计算出来的梯度称为陈旧梯度. 使用陈旧梯度更新全局模型会出现模型难收敛、准确性下降等现象，称为陈旧效应. 边缘协同的异构环境下，不稳定的网络环境与客户端的计算资源等因素加剧了陈旧效应. 有效地处理陈旧梯度对于异步更新训练算法实现良好的性能至关重要[91]. 一个简单而有效的技术来解决陈旧梯度问题就是惩罚陈旧梯度，梯度会根据时间衰减从而在更新模型时只使用陈旧梯度的一小部分以缓解陈旧梯度带来的不稳定[92]. Xie 等[90]将异步更新与联邦学习相结合，其中采用加权平均的方式更新全局模型权重，而过时梯度的权重通过设置为陈旧度的函数计算. 实验结果验证了该方法能容忍陈旧现象. 然而，惩罚陈旧梯度会使梯度变得任意小，在大规模训练中使模型收敛速率得不到保障. 另一个有效地对抗梯度陈旧问题的方法是对陈旧梯度调整学习率. Zhang 等[62]中提出了一种异步更新的陈旧感知算法，该算法根据全局模型更新轮次量化陈旧程度，从而自动调整学习率，通过图像分类实验验证了该异步更新算法在得到与同步 SGD 相当的模型精度的同时提供了接近线性的收敛加速. Odena 等[91]提出通过

统计梯度的平均移动量来量化梯度过时程度的方法, 并证明该方法在收敛速度和可扩展性方面均有显著提升. 因而可将异步更新运用在更丰富的领域中以适用于训练更多类型的机器学习模型, 如在 SGD 上使用动量能提高收敛速率和准确度[81,93]. Hakimi 等[94] 首次将动量运用在异步更新设置中以实现模型的高精度和快速收敛, 该方法在训练过程中预估模型参数未来位置上的梯度来减轻梯度的陈旧程度. 另外, 当各个客户端本地数据集之间在非独立同分布的情况下, 若引入异步会加剧陈旧效应[95]. Chen 等[95] 提出了一种两阶段的异步更新训练方法 FedSA, 考虑设备异构性和本地模型的相似性自适应调整训练超参数以应对数据非独立同分布和陈旧现象.

表 3 总结了数据并行相关工作.

表 3 数据并行相关工作

Table 3 Related works of data parallel

Parameter update method	Main problems	Solution	Reference
Synchronize updates	Client delay	Client filtering	[58–59]
		Client selection	[83]
		Hybrid update	[84]
		Partial update of model	[85]
Asynchronous update	Obsolescence effect	Astringency	[61,86]
		Penalty old gradient	[90,92]
		Adjust learning rate	[62,91]
		Use momentum	[94]
		Adjust super parameters	[95]

在数据并行中, 非独立同分布的数据十分常见, 非独立同分布会带来训练时间长、模型准确性低等问题, 因此边缘协同训练的一大挑战是边缘设备如何利用非独立同分布的数据进行协同训练.

Kairouz 等[40] 总结了非独立同分布数据的来源, 归结于边缘设备存储数据分布不同、边缘设备之间不独立和数据集偏差 (训练数据与测试数据不同分布). 其中边缘设备存储数据又可细分为: (1) 特征分布倾斜; (2) 标签分布倾斜; (3) 相同特征, 不同标签; (4) 相同标签, 不同特征; (5) 数据数量不平衡. 数据的非独立同分布符合联邦学习的设定, 因此近些年来涉及这方面的研究基于联邦学习. 理论上, 在强凸光滑问题中, 联邦平均算法能够在数据非独立同分布的设定下收敛[96–97], 但使用非独立同分布的数据进行训练会出现模型收敛速度不稳定和模型准确率下降问题[98–99]. Karimireddy 等[99] 推导了联邦平均算法的收敛速度, 并证明数据是非独立同分布时导致不稳定和缓慢的收敛源于本地模型更新偏离全局最优的现象, 称为“客户端漂移”现象.

现有应对数据的非独立同分布问题可以分成 3 类[40]:

(1) 修改现有的算法. Li 等[100] 中对联邦平均算法在本地更新过程的优化目标进行修改, 在优化目标中加入本地模型参数与全局模型参数的正则化项, 限制本地模型更新大小, 并在理论和实验上验证了该算法的稳定性和准确性. Karimireddy 等[99] 针对“客户端漂移”现象增加了一组控制变量, 目的是将本地模型更新方向修正到全局最优的位置上. Wang 等[101] 在更新全局模型的过程用一种归一化平均方法取代联邦平均算法的加权聚合操作, 避免全局模型收敛到不匹配目标函数点处. Hsu 等[98] 合成了不同程度的非独立同分布数据集用于评估算法性能, 结果表明随着数据分布的差异越来越大, 性能下降越多, 并通过服务器动量提出了一种缓解策略. Hsu 等[102] 中开发了两种新的算法 (FedVC、FedIR), 可以从工作节点中智能地重新采样和重新加权, 从而大大提高训练的准确性和稳定性.

(2) 边缘设备间共享信息. Zhao 等[103] 发现由非独立同分布引起的模型精度降低可以用权重分散来解释, 权重分散可以通过每个设备上的类分布和总体分布之间的搬土距离 (The earth mover's distance, EMD) 量化. 因此提出了一种数据共享策略, 通过创建一个在所有边缘设备之间全局共享的小数据集改进联邦平均算法. Yoshida 等[104] 考虑到了有极少数设备允许数据上传服务器的节点, 并在每次迭代更新中使用启发式算法询问并选择参与训练的客户端以及愿意被上传的数据集来减轻数据的非独立同分布程度. Shoham 等[105] 将终身学习的思想运用在解决数据非独立同分布上, 在客户端共享费歇耳信息矩阵保护对每个本地模型都重要的参数, 从而缓解“客户端漂移”现象.

(3) 个性化模型. Huang 等[106] 认为使用非 IID 数据进行联合学习的根本瓶颈是假设一个全局模型可以适合所有节点, 因此提出了允许每个节点设备都拥有一个本地的个性化模型, 并且通过一种消息传递机制对更新每个客户端的个性化模型, 节点之间的底层成对协作, 并证明了此方法对

于凸和非凸模型的收敛性. Wu 等[107] 提出了一种个性化边缘云协同的框架，该框架先为边缘设备协同训练出一个全局模型，然后每个设备将基于全局模型和本地数据训练一个个性化模型，并提出训练个性化模型可采用迁移学习、元学习和知识蒸馏等方法.

表 4 总结数据非独立同分布相关工作.

表 4　数据非独立同分布问题相关工作

Table 4　Related works of data non-independent and identical distribution issues

Research classification	Research direction	Reference
Theoretical proof	Source and classification	[40]
	Astringency	[96–97,99]
	Negative effect	[98–99]
Solution	Modify existing algorithm	[98–102]
	Share information	[103–105]
	Personalization model	[106–107]

2.2.2　模型并行

模型并行是一种将模型拆分到多个边缘设备进行并行化训练的方式，适用于模型架构太大而单个节点不能独立存储模型或计算输出的情况[82,108]. 在边缘协同训练场景下，模型并行可以通过模型的分割减轻节点的异构性、平衡计算量与通信量以达到更好的协同. 但模型并行也存在亟待解决的问题[109]：(1) 深度神经网络参数的更新包含前向传播和反向传播两个过程，并且网络层与层之间有严格的先后关系，这两点导致传统的模型并行方式效率低下，当一个工作节点处于工作状态的情况下，后续的所有节点均处于空闲的等待状态；(2) 模型并行效率依赖于模型拆分的质量，因此需要寻求高质量和高可靠性的模型切割技术.

模型并行与数据并行的混合式并行可以进一步提高训练过程的并行性[110–111]. Harlap 等[111] 提出一种流水线模型并行方法来解决传统的模型并行方式效率问题，允许传输和计算同时进行，和数据并行计算进一步提高训练速度. 这种方法首先以多个小批量数据作为输入，保持流水线处于稳定状态，并通过交替执行前向和反向传播机制进行调度，但这种训练方式会引起某一小批量数据反向传播更新的参数可能与前向传播时不匹配的梯度陈旧问题，因此采用了保存版本模型参数来确保反向传播的正确性. Chen 等[112] 提供了一种利用平滑的梯度进行权值预测的方法以解决流水线模型并行中梯度陈旧问题，并通过实验证明该方法实现的流水线模型并行吞吐量平均比数据并行吞吐量高 98.5%. Huang 等[113] 提出了一种模型并行框架 GPipe，GPipe 将同步更新与流水线模型并行进行结合. 该框架可扩展以层为序列的深度神经网络模型训练，并成功训练一个 5 亿多个参数的图像分类模型和一个包含 60 多亿个参数的 Transformer 模型.

在模型并行训练方式中，模型切割技术是一项重要技术. Harlap 等[111] 首先预测各层的处理时间，并且通过动态规划的方法切割模型. Mirhoseini 等[114] 将强化学习运用在模型并行训练中的模型切割中，使用一个强化学习模型为深度神经网络进行切割和卸载，其中执行时间被用作奖励信号来训练. 这些方法均为基于多 GPU 数据中心的分布式训练方法，然而在边缘协同训练场景，由于节点异构性、动态性以及训练节点的退出等因素，如何自适应切割模型以及正确完成模型参数的调度和更新仍需要进一步研究.

表 5 总结模型并行相关工作.

近年来，神经网络模型的规模变得越来越大，拥有数十亿参数的深度神经网络不再是例外，比如用于自然语言处理的 Megatron-LM 模型[115] 就拥有 83 亿个参数，这让神经网络模型部署在边缘设备上变得十分困难. 其次，无论是同步更新还是异步更新等分布式模型参数更新方式需要传输的梯度信息与模型规模成正相关，因此这给边缘计算

表 5　模型并行相关工作

Table 5　Related works of model parallelism

Research direction	Main problems	Solution	Reference
Parallel pipeline	Gradient obsolescence problem	Save version model parameters	[111]
		Weight prediction	[112]
		Synchronize updates	[113]
Model cutting training	Adaptive cutting and unloading	Dynamic programming	[111]
		Intensive learning	[114]

场景的通信成本带来阻碍, 大规模网络通常是过参数化的, 往往在部署推断期间可以通过压缩网络减小规模但不损失精度. 然而, 在训练阶段, 随着深度神经网络模型规模的增大, 模型变得更加容易训练, 泛化能力也在不断增强, 并且已有理论证明直接训练一个小规模模型的效果不如大型神经网络[116], 而且随着边缘设备上数据量的不断增加, 神经网络在计算和存储资源有限的情况下, 很难及时完成一些任务[117]. 目前知识蒸馏是将迁移学习运用在边缘协同训练场景的一种基于模型压缩的训练方式[3,48,118]. 不同于模型并行利用模型切割将一个大规模神经网络部署在多个边缘节点进行训练, 知识蒸馏则是通过解决小规模模型不易训练的问题来训练出和大规模网络推理性能相差不大的小型网络. 知识蒸馏首先在云端和边缘服务器上利用基本数据集进行训练一个大型神经网络, 或者利用现有公开的大型预训练神经网络作为教师网络. 教师网络可以指导边缘设备利用其私有数据集训练得到一个规模较小但性能不错的小型网络(称为学生网络)方便边缘设备使用. 基于知识蒸馏的训练方式不仅能得到一个将来方便在边缘设备上进行部署以提供推理服务的小规模神经网络, 还能相比于直接训练小型网络得到更好的模型准确率[119], 是边缘协同训练的一个具有潜力的研究方向.

Sharma 等[38] 通过大量实验研究验证在边缘设备中进行基于知识蒸馏的深度学习的有效性, 结果表明现有的知识蒸馏技术都有助于依赖学生模型更快地收敛, 但知识蒸馏过程的知识转移方式对模型准确率影响较大. Gou 等[120] 通过总结了一些现有的典型知识方法在两个流行的图像分类数据集上 (CIFAR10 和 CIFAR100) 的分类性能来展示知识蒸馏的有效性. Phuong 等[121] 首先利用线性分类器从理论上证明了知识蒸馏的有效性, 并指出决定知识蒸馏成功的三个因素: 数据分布的几何特性、优化偏差和训练集大小. 这是第一个给出基于知识蒸馏的训练有效性的定量分析工作. Anil 等[122] 从量化 DNN 中间层编码的知识角度来解释知识蒸馏的有效性, 并且验证了知识蒸馏确保 DNN 学习更多有用的概念, 具有更高的学习速度.

知识蒸馏是一种有效的模型压缩技术, 如何将其运用在边缘协同训练场景下是一个值得研究的方向. 首先, 传统的知识蒸馏需要预训练教师网络的参与, 这给知识蒸馏在分布式环境下的进行带来了挑战. 共蒸馏 (Co-distillation)[122] 是一种在线蒸馏在分布式环境下的变种, 将多个设备上的学生网络的 logit 输出进行平均来代替传统教师网络的监督信息, 因此各边缘设备上模型更新不仅受到训练样本的监督, 还要约束本地模型输出与其余多个设备输出的平均差异大小. 共蒸馏的设定无需预训练教师网络, 但需要边缘设备之间共享训练样本, 这不利于通信传输和隐私保护. Jeong 等[123] 提出了一种无需在边缘设备上共享数据集的联邦蒸馏训练方法, 该方法利用边缘设备在各类别上的 logit 平均作为监督信息, 将全局的 logit 平均作为每个类别的知识进行蒸馏学习.

联邦蒸馏为联邦学习提供了一种新的训练范式, 不仅为边缘计算场景下提供一个协同训练小规模高性能模型的方法, 而且还能适应边缘计算场景下有限的无线信道和时变网络拓扑. 近年来研究者在联邦蒸馏的基础上进行算法优化以达到通信效率或准确率的提高, 但对于联邦蒸馏的理论研究仍存在不足[124−126].

表 6 为知识蒸馏的相关工作.

表 6 知识蒸馏相关工作

Table 6 Related works of data non-independent and identical distribution issues

Research classification	Research means	Reference
Theoretical research	Effectiveness	[38,120–122]
	Critical factor	[38,121]
Distributed knowledge distillation	Co-distillation	[122]
	Federal distillation	[123–126]

近几年, 用户服务质量、用户体验优化以及数据隐私的要求不断提升, 边缘协同训练已经开始受到人们的广泛关注, 由于边缘环境由大量呈地理分布的边缘设备组成, 环境的复杂性使得边缘协同训练的研究仍需要进一步的研究与优化.

在整体框架中, 从有无中央服务器角色的角度, 将边缘协同训练的整体架构分为参数服务器集中式架构和完全分散并行式架构两大类. 在参数服务器集中式架构中, 按照设备间的交互主要包括同步和异步两种, 并从设备层次和通信层次对架构灵活性、架构鲁棒性和耗时优化等进行优化. 在完全分散并行式架构中, 面临选择最佳通信伙伴困难和资源受限挑战, 从邻居交互和通信消耗两个角度进行研究优化.

在核心模块中, 数据并行对多个边缘设备数据并行协同训练, 模型并行则将模型拆分到多个

边缘设备进行训练, 模型并行与数据并行的混合式并行可以进一步提高边缘协同训练效率. 在不稳定的边缘计算场景下, 数据并行过程中需要解决同步更新中的客户端延迟问题以及异步更新的陈旧效应; 而模型并行需要解决流水线并行上的梯度陈旧问题和如何切割模型以适应训练要求等问题. 另外, 无论是数据并行还是模型并行, 都会遭受到边缘设备上非独立同分布数据带来的负面影响. 因此, 边缘协同训练的核心模块如何适应边缘计算的异构环境值得进一步研究.

3　边缘协同智能在动态场景下的挑战与展望

本文从边缘协同训练出发, 简要描述了其发展过程, 着重从边缘协同智能的角度对模型的训练涉及到的关键技术进行了归纳总结, 并从动态场景的角度分别进行了分析. 但边缘协同训练还存在诸多挑战.

(1) 异步收敛. 在具有不稳定特点的边缘计算环境中, 参数服务器异步架构在通信性能上相较于参数服务器同步架构取得了良好的优化效果, 但在模型收敛性能的保证上却多因数据不平衡等问题而劣于同步架构. 异步通信机制不再指定每轮参与客户端, 而是使参数服务器被动接收所到达的客户端模型参数来进行聚合, 由于客户端在数据以及模型等方面的差异性, 每轮不同的组合对聚合更新质量就会有着不同的影响, 若客户端子集配合不协调, 则在很大程度上将会对模型聚合的收敛速度造成负面影响.

(2) 聚合优化. 完全分散并行式架构以其去除中央服务器的核心思想, 显著优化了参数服务器集中式架构的资源瓶颈问题. 但在参数聚合方面, 集中式架构的聚合面向全局的客户端来进行更新, 无论同步或异步, 在大规模的分布式系统中, 参与的客户端个数在采取一定的措施下, 可以得到很好的保障, 分散并行式架构中, 客户端面向的为邻近的局部客户端, 且多因各种标准使得可通信客户端个数进一步缩减, 因此对于每个客户端, 每轮的更新质量是无法得到保障的, 尽管可以通过大量的通信迭代得到很好的广播, 但这将导致大量的通信次数, 从而造成收敛效率低.

(3) 训练时长. 无论是数据并行还是模型并行, 均会遇到边缘设备因资源受限、设备异构和环境不稳定场景而导致模型更新速度慢的问题. 现有的研究工作为了加速模型收敛, 提出了选择满足一定条件的客户端、减轻陈旧梯度对全局模型参数的影响等方法. 但在边缘协同训练中, 某些资源受限的边缘设备可能存储着重要数据, 在训练过程中不能忽略或者降低这些低速的边缘设备在协同训练中的作用, 从而导致训练时间的增长.

(4) 模型性能. 模型性能往往受限于参与训练的边缘设备数据分布情况. 由于在边缘协同场景下, 数据的隐私性、异构性以及动态性使边缘设备节点的数据可感知性下降. 另外目前还没有一种现有的算法在数据所有的非独立同分步类型下都优于其他算法[90]. 因此, 在不易感知数据的分布情况下, 非独立同分布的数据给边缘协同智能中的训练与模型的准确性带来了挑战.

(5) 自适应性. 训练过程的效率受到多方面的制约, 包括集中式训练中客户端子集的动态变化、分散并行式训练的聚合策略、数据分布与计算资源分布不协调、由于服务类型多样和资源地理分布等特点导致的非独立同分布数据等. 这些问题的解决将会使边缘协同智能更加适用于动态场景.

(6) 验证评估. 已有的研究在验证思路时多数通过实际的生产场景, 或较为简单的代价评估模型模拟, 缺乏动态场景特点上的考虑. 分析原因: 一方面是生产场景对于广泛的研究人员并不容易可及; 二是模拟场景考虑的影响因素不够全面. 在边缘协同智能的有效性评估上还存在困难, 给边缘协同智能的落地带来一定阻碍.

针对边缘协同训练存在的诸多挑战, 对未来研究方向做出展望:

(1) 客户端子集的动态变化. 为解决异步通信机制带来的新问题, 以扩展参数服务器异步架构来更好完成边缘协同训练, 目前已有研究多从设备参数层面的局部角度和通信层面的全局角度对其进行优化, 但却忽略了一个关键步骤, 即每个通信轮次动态组合的客户端子集. 客户端子集的动态变化对参数服务器异步架构的协同训练性能起到了重要的作用, 仍需进一步探索未来发展的方向.

(2) 分散并行式架构的聚合. 对于该架构的优化, 研究者多认为分散并行式架构与集中式架构的不同在于邻居交互上的协调与控制, 但分散并行式架构在聚合上与集中式架构也有着很大的不同. 因此, 能否采取一种适用于分散并行式架构的聚合策略, 是优化该架构的一个很好的突破口, 如在少量的局部客户端情况下, 如何进行模型参数的聚合.

(3) 异构设备间的协同机制. 在边缘计算环境

下, 由于资源受限的低速边缘设备可能存储着重要数据, 因此整个训练过程不能忽略或减轻低速边缘设备参与训练的重要性. 如何在保证隐私安全和模型训练时长的条件下, 高速的边缘设备协助低速设备共同参与到模型训练或许是一个解决方案, 但异构设备间的协同机制仍需未来进一步探索方向.

(4) 面向动态变化数据的训练机制. 在非独立同分布数据下训练会带来的训练时间长、模型准确性降低等问题, 且由于现实需求, 模型也需要在更多样的智能学习设置中训练, 如强化学习、在线学习和集成学习等, 这些机器学习设置下可能面临边缘设备上数据的动态增长问题. 因此, 如何在非独立同分布且动态变化数据中训练更多样的高性能模型也是边缘协同训练中的未来一个方向.

(5) 在线学习与边缘计算的结合. 目前多数研究或工作将训练与推理分开, 界线清晰. 该类较适合于资源丰富且动态性不强的场景. 但对于涉及计算、网络等资源存在限制且动态的场景, 单独的训练过程并不适用, 如何利用有限的资源对模型进行在线更新值得研究, 如感知计算、触觉网络等.

(6) 边缘协同智能中的动态场景建模. 动态场景下的边缘计算普遍存在硬件故障、系统及软件的负载变化、人为与环境因素的影响、地理分布广泛以及服务状态复杂多变、易受影响等特点. 因此, 提供一个可信的动态场景仿真场景值得研究.

参 考 文 献

[1] Zhang X Z, Wang Y F, Lu S D, et al. OpenEI: an open framework for edge intelligence // 2019 *IEEE* 39*th International Conference on Distributed Computing Systems* (ICDCS). Dallas, 2019: 1840

[2] Wang R, Qi J P, Chen L, et al. Survey of collaborative inference for edge intelligence. *J Comput Res Dev*, 2023, 60(2): 398
(王睿, 齐建鹏, 陈亮, 等. 面向边缘智能的协同推理综述. 计算机研究与发展, 2023, 60(2): 398)

[3] Zhou Z, Chen X, Li E, et al. Edge intelligence: Paving the last Mile of artificial intelligence with edge computing. *Proc IEEE*, 2019, 107(8): 1738

[4] Li K L, Liu C B. Edge intelligence: State-of-the-art and expectations. *Big Data Res*, 2019, 5(3): 69
(李肯立, 刘楚波. 边缘智能: 现状和展望. 大数据, 2019, 5(3): 69)

[5] Tan H S, Guo D, Ke Z C, et al. Development and challenges of cloud edge collaborative intelligent edge computing. *CCCF*, 2020(1): 16
(谈海生, 郭得科, 张弛, 等. 云边端协同智能边缘计算的发展与挑战. 中国计算机协会通讯, 2020(1): 16)

[6] Zhang X Z, Lu S D, Shi W S. Research on collaborative computing technology in edge intelligence. *AI-View*, 2019, 6(5): 55
(张星洲, 鲁思迪, 施巍松. 边缘智能中的协同计算技术研究. 人工智能, 2019, 6(5): 55)

[7] Wang X F. Intelligent edge computing: From internet of everything to internet of everything empowered. *Frontiers*, 2020(9): 6
(王晓飞. 智慧边缘计算: 万物互联到万物赋能的桥梁. 人民论坛·学术前沿, 2020(9): 6)

[8] Fang A D, Cui L, Zhang Z W, et al. A parallel computing framework for cloud services // 2020 *IEEE International Conference on Advances in Electrical Engineering and Computer Applications* (*AEECA*). Dalian, 2020: 832

[9] Lanka S, Aung Win T, Eshan S. A review on Edge computing and 5G in IOT: Architecture & Applications // 2021 5*th International Conference on Electronics, Communication and Aerospace Technology* (*ICECA*). Coimbatore, 2021: 532

[10] Carrie M, David R, Michael S. The growth in connected IoT devices is expected to generate 79.4ZB of data in 2025, according to a new IDC forecast. (2019-06-18) [2022-09-26]. https://www.businesswire.com/news/home/20190618005012

[11] Zwolenski M, Weatherill L. The digital universe rich data and the increasing value of the internet of things. *J Telecommun Digital Economy*, 2014, 2(3): 47.1

[12] Jin H, Jia L, Zhou Z. Boosting edge intelligence with collaborative cross-edge analytics. *IEEE Internet Things J*, 2021, 8(4): 2444

[13] Jiang X L, Shokri-Ghadikolaei H, Fodor G, et al. Low-latency networking: Where latency lurks and how to tame it. *Proc IEEE*, 2019, 107(2): 280

[14] Xiao Y H, Jia Y Z, Liu C C, et al. Edge computing security: State of the art and challenges. *Proc IEEE*, 2019, 107(8): 1608

[15] Huang T, Liu J, Wang S, et al. Survey of the future network technology and trend. *J Commun*, 2021, 42(1): 130
(黄韬, 刘江, 汪硕, 等. 未来网络技术与发展趋势综述. 通信学报, 2021, 42(1): 130)

[16] Jennings A, Copenhagen van R, Rusmin T. *Aspects of Network Edge Intelligence*. Maluku Technical Report, 2001

[17] Song C H, Zeng P, Yu H B. Industrial Internet intelligent manufacturing edge computing: State-of-the-art and challenges. *ZTE Technol J*, 2019, 25(3): 50
(宋纯贺, 曾鹏, 于海斌. 工业互联网智能制造边缘计算: 现状与挑战. 中兴通讯技术, 2019, 25(3): 50)

[18] Risteska Stojkoska B L, Trivodaliev K V. A review of Internet of Things for smart home: Challenges and solutions. *J Clean Prod*, 2017, 140: 1454

[19] Varghese B, Wang N, Barbhuiya S, et al. Challenges and opportunities in edge computing // 2016 *IEEE International Conference on Smart Cloud* (*SmartCloud*). New York, 2016: 20

[20] Shi W S, Zhang X Z, Wang Y F, et al. Edge computing: State-of-the-art and future directions. *J Comput Res Dev*, 2019, 56(1): 69
(施巍松, 张星洲, 王一帆, 等. 边缘计算: 现状与展望. 计算机研究与发展, 2019, 56(1): 69)

[21] Teerapittayanon S, McDanel B, Kung H T. Distributed deep neural networks over the cloud, the edge and end devices // 2017 *IEEE 37th International Conference on Distributed Computing Systems* (*ICDCS*). Atlanta, 2017: 328

[22] Wang X F, Han Y W, Wang C Y, et al. In-edge AI: Intelligentizing mobile edge computing, caching and communication by federated learning. *IEEE Netw*, 2019, 33(5): 156

[23] Kang Y P, Hauswald J, Gao C, et al. Neurosurgeon. *SIGOPS Oper Syst Rev*, 2017, 51(2): 615

[24] Li E, Zhou Z, Chen X. Edge intelligence: On-demand deep learning model co-inference with device-edge synergy // *Proceedings of the* 2018 *Workshop on Mobile Edge Communications*. Budapest, 2018: 31

[25] Li Y K, Zhang T, Chen J L. Broad Siamese network for edge computing applications. *Acta Autom Sin*, 2020, 46(10): 2060
(李逸楷, 张通, 陈俊龙. 面向边缘计算应用的宽度孪生网络. 自动化学报, 2020, 46(10): 2060)

[26] Al-Rakhami M, Alsahli M, Hassan M M, et al. Cost efficient edge intelligence framework using docker containers // 2018 *IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing*, *16th Intl Conf on Pervasive Intelligence and Computing*, *4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress* (*DASC/PiCom/DataCom/CyberSciTech*). Athens, 2018: 800

[27] Al-Rakhami M, Gumaei A, Alsahli M, et al. A lightweight and cost effective edge intelligence architecture based on containerization technology. *World Wide Web*, 2020, 23(2): 1341

[28] Zaharia M, Xin R S, Wendell P, et al. Apache spark. *Commun ACM*, 2016, 59(11): 56

[29] Abadi M, Barham P, Chen J M, et al. TensorFlow: A system for large-scale machine learning [J/OL]. *ArXiv Preprint* (2016-05-31) [2022-09-26]. https://arxiv.org/abs/1605.08695

[30] Chen T Q, Li M, Li Y T, et al. MXNet: A flexible and efficient machine learning library for heterogeneous distributed systems [J/OL]. *ArXiv Preprint* (2015-12-03) [2022-09-26]. https://arxiv.org/abs/1512.01274

[31] Jin A L, Xu W C, Guo S, et al. PS: A simple yet effective framework for fast training on parameter server. *IEEE Trans Parallel Distributed Syst*, 2022, 33(12): 4625

[32] Padmanandam K, Lingutla L. Practice of applied edge analytics in intelligent learning framework // 2020 *21st International Arab Conference on Information Technology* (*ACIT*). Giza, 2021: 1

[33] Ross P, Luckow A. EdgeInsight: characterizing and modeling the performance of machine learning inference on the edge and cloud // 2019 *IEEE International Conference on Big Data* (*Big Data*). Los Angeles, 2020: 1897

[34] Shi W S, Sun H, Cao J, et al. Edge computing–An emerging computing model for the Internet of everything era. *J Comput Res Dev*, 2017, 54(5): 907
(施巍松, 孙辉, 曹杰, 等. 边缘计算: 万物互联时代新型计算模型. 计算机研究与发展, 2017, 54(5): 907)

[35] Srivastava A, Nguyen D, Aggarwal S, et al. Performance and memory trade-offs of deep learning object detection in fast streaming high-definition images // 2018 *IEEE International Conference on Big Data* (*Big Data*). Seattle, 2018: 3915

[36] Sindhu C, Vyas D V, Pradyoth K. Sentiment analysis based product rating using textual reviews // 2017 *International Conference of Electronics, Communication and Aerospace Technology* (*ICECA*). Coimbatore, 2017: 727

[37] Hosein P, Rahaman I, Nichols K, et al. Recommendations for long-term profit optimization // *Proceedings of ImpactRS@ RecSys*. Copenhagen, 2019

[38] Sharma R, Biookaghazadeh S, Li B X, et al. Are existing knowledge transfer techniques effective for deep learning with edge devices? // 2018 *IEEE International Conference on Edge Computing* (*EDGE*). San Francisco, 2018: 42

[39] Bonawitz K, Eichner H, Grieskamp W, et al. Towards federated learning at scale: System design // *Proceedings of Machine Learning and Systems*. Palo Alto, 2019, 1: 374

[40] Kairouz P, McMahan H B, Avent B, et al. Advances and open problems in federated learning. *FNT Machine Learning*, 2021, 14(1-2): 1

[41] McMahan H B, Moore E, Ramage D, et al. Communication-efficient learning of deep networks from decentralized data [J/OL]. *ArXiv Preprint* (2017-02-28) [2022-09-26]. https://arxiv.org/abs/1602.05629

[42] Zhu J M, Zhang Q N, Gao S, et al. Privacy preserving and trustworthy federated learning model based on blockchain. *Chin J Comput*, 2021, 44(12): 2464
(朱建明, 张沁楠, 高胜, 等. 基于区块链的隐私保护可信联邦学习模型. 计算机学报, 2021, 44(12): 2464)

[43] Wei S Y, Tong Y X, Zhou Z M, et al. *Efficient and Fair Data Valuation for Horizontal Federated Learning*. Berlin: Springer, 2020

[44] Khan A, Thij M, Wilbik A. Communication-efficient vertical federated learning. *Algorithms*, 2022, 15(8): 273

[45] Chen Y Q, Qin X, Wang J D, et al. FedHealth: A federated transfer learning framework for wearable healthcare. *IEEE Intell Syst*, 2020, 35(4): 83

[46] Yang J, Zheng J, Zhang Z, et al. Security of federated learning for cloud-edge intelligence collaborative computing. *Int J Intell Syst*, 2022, 37(11): 9290

[47] Zhang X J, Gu H L, Fan L X, et al. No free lunch theorem for security and utility in federated learning [J/OL]. *ArXiv Preprint* (2022-09-05) [2022-09-26].https://arxiv.org/abs/2203.05816

[48] Deng S G, Zhao H L, Fang W J, et al. Edge intelligence: The confluence of edge computing and artificial intelligence. *IEEE*

Internet Things J, 2020, 7(8): 7457

[49] Feng C, Han P C, Zhang X, et al. Computation offloading in mobile edge computing networks: A survey. *J Netw Comput Appl*, 2022, 202: 103366

[50] Qiao D W, Guo S T, He J, et al. Edge intelligence: Research progress and challenges. *Radio Commun Technol*, 2022, 48(1): 34
(乔德文, 郭松涛, 何静, 等. 边缘智能: 研究进展及挑战. 无线电通信技术, 2022, 48(1): 34)

[51] Fortino G, Zhou M C, Hassan M M, et al. Pushing artificial intelligence to the edge: Emerging trends, issues and challenges. *Eng Appl Artif Intell*, 2021, 103: 104298

[52] Qiu X C, Fernández-Marqués J, Gusmão P, et al. ZeroFL: Efficient on-device training for federated learning with local sparsity [J/OL]. *ArXiv Preprint* (2022-08-04) [2022-09-26]. https://arxiv.org/abs/2208.02507

[53] Long S Q, Long W F, Li Z T, et al. A game-based approach for cost-aware task assignment with QoS constraint in collaborative edge and cloud environments. *IEEE Trans Parallel Distributed Syst*, 2021, 32(7): 1629

[54] Zhu H R, Yuan G J, Yao C J, et al. Survey on network of distributed deep learning training. *J Comput Res Dev*, 2021, 58(1): 98
(朱泓睿, 元国军, 姚成吉, 等. 分布式深度学习训练网络综述. 计算机研究与发展, 2021, 58(1): 98)

[55] Rafique Z, Khalid H M, Muyeen S M. Communication systems in distributed generation: A bibliographical review and frameworks. *IEEE Access*, 2020, 8: 207226

[56] Hsieh K, Harlap A, Vijaykumar N, et al. Gaia: Geo-distributed machine learning approaching LAN speeds // *Proceedings of the 14th USENIX Conference on Networked Systems Design and Implementation*. New York, 2017: 629

[57] Konečný J, McMahan H B, Yu F X, et al. Federated learning: Strategies for improving communication efficiency [J/OL]. *ArXiv Preprint* (2017-10-30) [2022-09-26]. https://arxiv.org/abs/1610.05492

[58] Chen J M, Pan X H, Monga R, et al. Revisiting distributed synchronous SGD [J/OL]. *ArXiv Preprint* (2017-03-21) [2022-09-26]. https://arxiv.org/abs/1604.00981

[59] Nishio T, Yonetani R. Client selection for federated learning with heterogeneous resources in mobile edge // *ICC* 2019–2019 *IEEE International Conference on Communications* (*ICC*). Shanghai, 2019: 1

[60] Wang S Q, Tuor T, Salonidis T, et al. When edge meets learning: Adaptive control for resource-constrained distributed machine learning // *IEEE INFOCOM* 2018-*IEEE Conference on Computer Communications*. Honolulu, 2018: 63

[61] Lian X R, Huang Y J, Li Y C, et al. Asynchronous parallel stochastic gradient for nonconvex optimization // *Proceedings of the 28th International Conference on Neural Information Processing Systems*. Montreal, 2015: 2737

[62] Zhang W, Gupta S, Lian X R, et al. Staleness-aware async-SGD for distributed deep learning [J/OL]. *ArXiv Preprint* (2014-04-05) [2022-09-26]. https://arxiv.org/abs/1511.05950

[63] Lu X F, Liao Y Y, Lio P, et al. Privacy-preserving asynchronous federated learning mechanism for edge network computing. *IEEE Access*, 2020, 8: 48970

[64] Chen Y J, Ning Y, Slawski M, et al. Asynchronous online federated learning for edge devices with non-IID data // 2020 *IEEE International Conference on Big Data* (*Big Data*). Atlanta, 2021: 15

[65] Dutta S, Wang J Y, Joshi G. Slow and stale gradients can win the race. *IEEE J Sel Areas Inf Theory*, 2021, 2(3): 1012

[66] Lu Y L, Huang X H, Zhang K, et al. Blockchain empowered asynchronous federated learning for secure data sharing in Internet of vehicles. *IEEE Trans Veh Technol*, 2020, 69(4): 4298

[67] Wu W T, He L G, Lin W W, et al. SAFA: A semi-asynchronous protocol for fast federated learning with low overhead. *IEEE Trans Comput*, 2021, 70(5): 655

[68] Luehr N. Fast multi-GPU collectives with NCCL [J/OL]. *NVIDIA Developer* (2016-04-07) [2022-09-26]. https://developer.nvidia.com/blog/fast-multi-gpu-collectives-nccl

[69] Lian X R, Zhang W, Zhang C, et al. Asynchronous decentralized parallel stochastic gradient descent [J/OL]. *ArXiv Preprint* (2018-09-25) [2022-09-26]. https://arxiv.org/abs/1710.06952

[70] Lalitha A, Kilinc O C, Javidi T, et al. Peer-to-peer federated learning on graphs [J/OL]. *ArXiv Preprint* (2019-01-31) [2022-09-26].https://arxiv.org/abs/1901.11173

[71] Blot M, Picard D, Cord M, et al. Gossip training for deep learning [J/OL]. *ArXiv Preprint* (2016-11-29) [2022-09-26]. https://arxiv.org/abs/1611.09726

[72] Jin P H, Yuan Q C, Iandola F, et al. How to scale distributed deep learning? [J/OL]. *ArXiv Preprint* (2016-11-14) [2022-09-26]. https://arxiv.org/abs/1611.04581

[73] Daily J, Vishnu A, Siegel C, et al. GossipGraD: Scalable Deep Learning using Gossip Communication based asynchronous gradient descent [J/OL]. *ArXiv Preprint* (2018-03-15) [2022-09-26]. https://arxiv.org/abs/1803.05880

[74] Vanhaesebrouck P, Bellet A, Tommasi M. Decentralized collaborative learning of personalized models over networks // *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. Florida, 2017: 509

[75] He C Y, Tan C H, Tang H L, et al. Central server free federated learning over single-sided trust social networks [J/OL]. *ArXiv Preprint* (2020-08-01) [2022-09-26]. https://arxiv.org/abs/1910.04956

[76] Colin I, Bellet A, Salmon J, et al. Gossip dual averaging for decentralized optimization of pairwise functions[J/OL]. *ArXiv Preprint* (2016-06-08) [2022-09-26]. https://arxiv.org/abs/1606.02421

[77] Nedić A, Olshevsky A. Stochastic gradient-push for strongly convex functions on time-varying directed graphs. *IEEE Trans*

Autom Control, 2016, 61(12): 3936

[78] Assran M, Loizou N, Ballas N, et al. Stochastic gradient push for distributed deep learning // *Proceedings of the* 36th *International Conference on Machine Learning*. California, 2019: 344

[79] Koloskova A, Stich S, Jaggi M. Decentralized stochastic optimization and gossip algorithms with compressed communication // *Proceedings of the* 36*th International Conference on Machine Learning*. California, 2019: 3478

[80] Hu C H, Jiang J Y, Wang Z. Decentralized federated learning: A segmented gossip approach [J/OL]. *ArXiv Preprint* (2019-08-21) [2022-09-26]. https://arxiv.org/abs/1908.07782

[81] Ruder S. An overview of gradient descent optimization algorithms [J/OL]. *ArXiv Preprint* (2017-06-15) [2022-09-26]. https://arxiv.org/abs/1609.04747

[82] Chahal K S, Grover M S, Dey K, et al. A hitchhiker's guide on distributed training of deep neural networks. *J Parallel Distributed Comput*, 2020, 137: 65

[83] Chai Z, Ali A, Zawad S, et al. TiFL: A tier-based federated learning system // *Proceedings of the* 29*th International Symposium on High-Performance Parallel and Distributed Computing*. Stockholm, 2020: 125

[84] Li X Y, Qu Z, Tang B, et al. Stragglers are not disaster: A hybrid federated learning algorithm with delayed gradients[J/OL]. *ArXiv Preprint* (2021-02-12) [2022-09-26]. https://arxiv.org/abs/2102.06329

[85] Xu Z R, Yang Z, Xiong J J, et al. ELFISH: Resource-aware federated learning on heterogeneous edge devices[J/OL]. *ArXiv Preprint* (2021-03-01) [2022-09-26]. https://arxiv.org/abs/1912.01684

[86] Agarwal A, Duchi J C. Distributed delayed stochastic optimization // *Proceedings of the* 24*th International Conference on Neural Information Processing Systems*. Granada, 2011: 873

[87] Sahu A N, Dutta A, Tiwari A, et al. On the convergence analysis of asynchronous SGD for solving consistent linear systems [J/OL]. *ArXiv Preprint* (2020-04-05) [2022-09-26]. https://arxiv.org/abs/2004.02163

[88] Dean J, Corrado G S, Monga R, et al. Large scale distributed deep networks // *Proceedings of the* 25*th International Conference on Neural Information Processing Systems*. Lake Tahoe, 2012: 1223

[89] Zhang S X, Choromanska A, LeCun Y. Deep learning with elastic averaging SGD // *Proceedings of the* 28*th International Conference on Neural Information Processing Systems*. Montreal, 2015: 685

[90] Xie C, Koyejo S, Gupta I. Asynchronous federated optimization [J/OL]. *ArXiv Preprint* (2020-12-05) [2022-09-26]. https://arxiv.org/abs/1903.03934

[91] Odena A. Faster asynchronous SGD [J/OL]. *ArXiv Preprint* (2016-01-15) [2022-09-26]. https://arxiv.org/abs/1601.04033

[92] Chan W, Lane I. Distributed asynchronous optimization of convolutional neural networks // *Proceedings of Fifteenth Annual Conference of the International Speech Communication Association*. Singapore, 2014: 1073

[93] Sutskever I, Martens J, Dahl G, et al. On the importance of initialization and momentum in deep learning // *Proceedings of the* 30*th International Conference on International Conference on Machine Learning*. Atlanta, 2013: 1139

[94] Hakimi I, Barkai S, Gabel M, et al. Taming momentum in a distributed asynchronous environment [J/OL]. *ArXiv Preprint* (2020-10-14) [2022-09-26]. https://arxiv.org/abs/1907.11612

[95] Chen M, Mao B C, Ma T Y. FedSA: A staleness-aware asynchronous federated learning algorithm with non-IID data. *Future Gener Comput Syst*, 2021, 120: 1

[96] Li X, Huang K X, Yang W H, et al. On the convergence of FedAvg on non-IID data [J/OL]. *ArXiv Preprint* (2020-06-25) [2022-09-26]. https://arxiv.org/abs/1907.02189

[97] Khaled A, Mishchenko K, Richtárik P. First analysis of local GD on heterogeneous data [J/OL]. *ArXiv Preprint* (2020-03-18) [2022-09-26]. https://arxiv.org/abs/1909.04715

[98] Hsu T M H, Qi H, Brown M. Measuring the effects of non-identical data distribution for federated visual classification [J/OL]. *ArXiv Preprint* (2019-09-13) [2022-09-26]. https://arxiv.org/abs/1909.06335

[99] Karimireddy S P, Kale S, Mohri M, et al. SCAFFOLD: Stochastic controlled averaging for on-device federated learning [J/OL]. *ArXiv Preprint* (2021-04-09) [2022-09-26]. https://arxiv.org/abs/1910.06378

[100] Li T, Sahu A K, Zaheer M, et al. Federated optimization in heterogeneous networks [J/OL]. *ArXiv Preprint* (2020-04-21) [2022-09-26]. https://arxiv.org/abs/1812.06127

[101] Wang J Y, Liu Q H, Liang H, et al. Tackling the objective inconsistency problem in heterogeneous federated optimization [J/OL]. *ArXiv Preprint* (2020-07-15) [2022-09-26]. https://arxiv.org/abs/2007.07481

[102] Hsu T M H, Qi H, Brown M. Federated visual classification with real-world data distribution [J/OL]. *ArXiv Preprint* (2020-07-17) [2022-09-26]. https://arxiv.org/abs/2003.08082

[103] Zhao Y, Li M, Lai L Z, et al. Federated learning with non-IID data [J/OL]. *ArXiv Preprint* (2022-07-21) [2022-09-26]. https://arxiv.org/abs/1806.00582

[104] Yoshida N, Nishio T, Morikura M, et al. Hybrid-FL for wireless networks: Cooperative learning mechanism using non-IID data // *ICC* 2020 –2020 *IEEE International Conference on Communications* (*ICC*). Dublin, 2020: 1

[105] Shoham N, Avidor T, Keren A, et al. Overcoming forgetting in federated learning on non-IID data [J/OL]. *ArXiv Preprint* (2019-10-17) [2022-09-26]. https://arxiv.org/abs/1910.07796

[106] Huang Y T, Chu L Y, Zhou Z R, et al. Personalized cross-silo federated learning on non-IID data. *Proc AAAI Conf Artif Intell*, 2021, 35(9): 7865

[107] Wu Q, He K W, Chen X. Personalized federated learning for intelligent IoT applications: A cloud-edge based framework.

IEEE Open J Comput Soc, 2020, 1: 35

[108] Günther S, Ruthotto L, Schroder J B, et al. Layer-parallel training of deep residual neural networks [J/OL]. *ArXiv Preprint* (2019-07-25) [2022-09-26]. https://arxiv.org/abs/1812.04352

[109] Mayer R, Jacobsen H-A. Scalable deep learning on distributed infrastructures: Challenges, techniques, and tools. *ACM Comput Surv*, 2020, 53(1): 1

[110] Jia Z H, Zaharia M, Aiken A. Beyond data and model parallelism for deep neural networks [J/OL]. *ArXiv Preprint* (2018-07-14) [2022-09-26]. https://arxiv.org/abs/1807.05358

[111] Harlap A, Narayanan D, Phanishayee A, et al. PipeDream: Fast and efficient pipeline parallel DNN training [J/OL]. *ArXiv Preprint* (2018-06-08) [2022-09-26]. https://arxiv.org/abs/1806.03377

[112] Chen C C, Yang C L, Cheng H Y. Efficient and robust parallel DNN training through model parallelism on multi-GPU platform [J/OL]. *ArXiv Preprint* (2019-10-28) [2022-09-26]. https://arxiv.org/abs/1809.02839

[113] Huang Y P, Cheng Y L, Bapna A, et al. GPipe: Efficient training of giant neural networks using pipeline parallelism [J/OL]. *ArXiv Preprint* (2019-07-25) [2022-09-26]. https://arxiv.org/abs/1811.06965

[114] Mirhoseini A, Pham H, Le Q V, et al. Device placement optimization with reinforcement learning // *Proceedings of the* 34*th International Conference on Machine Learning*. Sydney, 2017: 2430

[115] Shoeybi M, Patwary M, Puri R, et al. Megatron-LM: Training multi-billion parameter language models using model parallelism [J/OL]. *ArXiv Preprint* (2020-03-13) [2022-09-26]. https://arxiv.org/abs/1909.08053

[116] Frankle J, Carbin M. The lottery ticket hypothesis: Finding sparse, trainable neural networks [J/OL]. *ArXiv Preprint* (2019-03-04) [2022-09-26]. https://arxiv.org/abs/1803.03635

[117] Wang Z D, Liu X X, Huang L, et al. QSFM: Model pruning based on quantified similarity between feature maps for AI on edge. *IEEE Internet Things J*, 2022, 9(23): 24506

[118] Wang J, Zhang J G, Bao W D, et al. Not just privacy: Improving performance of private deep learning in mobile cloud // *Proceedings of the* 24*th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. London, 2018: 2407

[119] Zhang L F, Tan Z H, Song J B, et al. Scan: A scalable neural networks framework towards compact and efficient models // 33*rd Conference on Neural Information Processing Systems* (*NeurIPS* 2019). Vancouver, 2019: 32

[120] Gou J P, Yu B S, Maybank S J, et al. Knowledge distillation: A survey [J/OL]. *ArXiv Preprint* (2021-03-20) [2022-09-26]. https://arxiv.org/abs/2006.05525

[121] Phuong M, Lampert C H. Towards understanding knowledge distillation [J/OL]. *ArXiv Preprint* (2021-03-27) [2022-09-26].https://arxiv.org/abs/2105.13093

[122] Anil R, Pereyra G, Passos A, et al. Large scale distributed neural network training through online distillation [J/OL]. *ArXiv Preprint* (2020-08-20) [2022-09-26]. https://arxiv.org/abs/1804.03235

[123] Jeong E, Oh S, Kim H, et al. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-IID private data [J/OL]. *ArXiv Preprint* (2018-11-28) [2022-09-26]. https://arxiv.org/abs/1811.11479

[124] Shen T, Zhang J, Jia X K, et al. Federated mutual learning [J/OL]. *ArXiv Preprint* (2020-09-17) [2022-09-26]. https://arxiv.org/abs/2006.16765

[125] Sattler F, Marban A, Rischke R, et al. Communication-efficient federated distillation [J/OL]. *ArXiv Preprint* (2020-12-01) [2022-09-26]. https://arxiv.org/abs/2012.00632

[126] Ahn J H, Simeone O, Kang J. Wireless federated distillation for distributed edge learning with heterogeneous data // 2019 *IEEE* 30*th Annual International Symposium on Personal, Indoor and Mobile Radio Communications* (*PIMRC*). Istanbul, 2019: 1