计算机科学
COMPUTER SCIENCE

面向联邦大语言模型训练的传输优化技术综述

顿婧博, 李卓

引用本文

顿婧博, 李卓. 面向联邦大语言模型训练的传输优化技术综述[J]. 计算机科学, 2025, 52(1): 42-55.

DUN Jingbo, LI Zhuo. Survey on Transmission Optimization Technologies for Federated Large Language Model Training [J]. Computer Science, 2025, 52(1): 42-55.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.240500095

面向联邦大语言模型训练的传输优化技术综述

顿婧博　李　卓
网络文化与数字传播北京市重点实验室(北京信息科技大学)　北京 100101
北京信息科技大学计算机学院　北京 100101
(dunjingbo@163.com)

摘　要　随着人工智能技术的快速发展,各类大型语言模型不断涌现。但是专用大语言模型的用户及数据集大多具有隐私性和安全性要求,数据安全隐私问题亟待解决。在此背景下,联邦大语言模型应运而生并得到越来越多的关注。由于大型语言模型庞大的数据量以及联邦学习的分布式架构,海量的参与节点与云服务器间进行大量的模型交换会产生较高的通信成本。为提升模型收敛速率,研究人员对面向联邦大语言模型训练的传输优化技术展开了研究。文章分析了联邦大语言模型所面临的挑战;综述了基于模型微调的传输优化方法、基于模型压缩的传输优化方法以及基于分布式并行处理的传输优化的优化问题;介绍了已有的开源联邦大语言模型以及所用到的传输优化技术,并对未来研究方向进行了展望。

关键词:联邦学习;大语言模型;传输优化;通信开销;模型压缩
中图分类号　TP393

Survey on Transmission Optimization Technologies for Federated Large Language Model Training

DUN Jingbo and LI Zhuo
Beijing Key Laboratory of Internet Culture and Digital Dissemination Research(Beijing Information Science & Technology University),Beijing 100101,China
School of Computer Science,Beijing Information Science & Technology University,Beijing 100101,China

Abstract　With the rapid development of artificial intelligence technology,various types of large language models are emerging. However,most users and datasets participating in dedicated large language models have certain requirements for privacy and security,the data security and privacy issues need to be solved urgently,and federated large language models have emerged and gained more and more attention. Due to the huge data volume of large language models and the distributed architecture of federated learning,a large number of model exchanges between a large number of participating nodes and cloud servers result in high communication costs. In order to improve the model convergence rate,researchers have investigated transmission optimization techniques for federated large language model training. This paper analyzes the challenges of federated large language models,reviews the optimization problems of transmission optimization methods based on model fine-tuning,transmission optimization methods based on model structure compression,and transmission optimization based on distributed parallel processing;introduces existing open-source federated large language models and the transmission optimization techniques used,and gives an outlook on future research directions.

Keywords　Federated learning,Large language models,Transmission optimization,Communication overhead,Model compression

1　引言

近年来,人工智能的蓬勃发展催生出一系列大型语言模型(Large Language Model,LLM),如 BERT,T5,OpenAI GPT,ChatGPT 等[1],其中具有代表性的大语言模型是由 OpenAI 开发的 GPT 系列,这类模型不仅可以生成流畅自然的文本,还能理解和推断输入的语境。除此之外,大型语言模型在特定领域的应用也层出不穷,如在医疗保健、金融银行等领域出现了专用大模型。然而,在特定领域训练大型语言模型涉及用户及其数据的隐私和安全问题,并且大型语言模型的训练通常需要庞大的中心化数据集和强大的计算能力,这些都给大型语言模型的训练带来了一定的挑战。

为了解决这些问题,研究人员引入联邦学习——一种分布式机器学习框架,其允许节点在本地进行训练,在实现分布式计算的同时可以有效解决用户数据隐私保护问题。因此可以考虑将联邦学习与大语言模型相结合(下称联邦大语言

到稿日期:2024-05-22　返修日期:2024-09-11
基金项目:北京市自然科学基金(4232024);国家重点研发计划(2022YFF0604502)
This work was supported by the Natural Science Foundation of Beijing,China(4232024) and National Key R&D Program of China(2022YFF0604502).
通信作者:李卓(lizhuo@bistu.edu.cn)

模型)。联邦大语言模型的出现代表着两个前沿领域的交汇,为分布式数据隐私保护和大规模自然语言处理任务提供了新的可能性。将二者相结合能够在分布式环境中训练大规模、高性能的语言模型并且为解决大模型中用户数据隐私问题提供新的思路。

联邦大语言模型与传统的两层联邦学习均会面临大量的模型更新传输,这就需要庞大的网络资源以及通信和计算成本,而它们的不同之处在于联邦大语言模型是将大语言模型放在分布式联邦学习架构下进行训练,挑战之一是如何在保持大模型性能的同时解决分布式训练以及去中心化学习的矛盾,并且 LLM 中庞大的参数量会使通信传输难度增大,单个设备节点无法承担大模型的通信及计算开销,联邦学习中传统的模型训练技术会导致大模型训练效率低下,这些都是可优化的方向。分层联邦学习是将边缘计算应用到联邦学习中,利用边缘服务器进行部分模型聚合。将大语言模型与分层联邦学习相结合可能会面临模型切分的问题,从而增加模型聚合的难度及复杂性。联邦学习应用场景差异比较如表 1 所列。

表 1 联邦学习应用场景差异比较

Table 1 Comparison of differences in federated learning application scenarios

应用场景	传统联邦学习	云-边-端分层联邦学习	联邦大语言模型
参数传输路径	客户端-中央服务器	云服务器-边缘设备-终端设备	客户端-中央服务器
通信成本	大量的数据传输,通信成本较高	数据传输路径较短,通信成本相对较低	大量的数据传输,通信成本较高
网络带宽需求	需求较低,仅设计两个层级的通信	需求较高,需处理多个层级的通信	需求取决于模型规模以及通信频率
适用场景	数据分布均匀、本地设备计算能力有限的场景	数据分布不均匀,需要实时响应的场景	大规模模型、本地设备计算能力有限的场景

本文从联邦大语言模型训练过程出发,深入分析联邦大语言模型的原理及优化技术,对相关研究的进展和现状进行讨论。第 2 章主要介绍联邦大语言模型的背景知识,明确其基本概念及原理,并归纳总结联邦大语言模型的训练流程;第 3 章归纳联邦大语言模型所需应对的挑战:海量的模型参数、庞大的显存需求以及巨大的通信及计算开销;第 4 章总结了联邦大语言模型的传输优化技术:通过基于 Adapter Tuning、Prompt Tuning 以及 LoRA 微调的优化方法、基于模型结构压缩的传输优化方法等,通过分布式训练技术中并行(parallelism)技术,包括模型并行、管道并行等来优化模型传输过程;第 5 章介绍已有的开源联邦大语言模型,包括 FATE-LLM、FedLLM、FederatedScope-LLM、PrimiHub 联邦语言大模型、OpenFedLLM 以及 FedLLM-Bench;第 6 章对该交汇领域提出研究展望,探讨未来的发展方向。

2 联邦大语言模型

2.1 联邦学习

联邦学习(Federated Learning,FL)是谷歌的 McMahan 等[2]于 2017 年首次提出并落地应用的一种分布式机器学习框架,整个框架由多个参与节点和一个云服务器组成,在参与节点的私有数据集上进行模型训练,并将本地模型参数而非本地数据上传至云服务器进行全局模型聚合,不必依赖云服务器来集中训练模型,形成了一个星型拓扑的网络通信结构。这种将数据在本地进行训练的方式有效地解决了用户数据隐私问题,在不破坏数据隐私的情况下实现了高效协作模型学习。联邦学习中有两种重要的角色,即产生和收集数据的参与节点以及进行全局模型聚合的云服务器。参与节点使用本地数据来训练本地模型并将其上传到云服务器,云服务器接收上传的本地模型进行全局聚合从而得到全局模型,经过多轮迭代后得到训练损失函数最小的模型参数 θ。研究人员提出一系列优化策略来加快联邦学习的模型收敛,提升模型质量。谷歌的 McMahan 等[2]于 2017 年提出 Federated Averaging(FedAVG)算法,该算法的基本原理是采用梯度下降的方式对模型参数进行迭代更新。目前已有大量的创新技术对联邦学习的发展做出了巨大贡献,在保持用户数据隐私和安全的同时实现了分布式的机器学习。

联邦学习模型训练的一次迭代过程如图 1 所示,其中 t 表示进行第 t 次迭代。

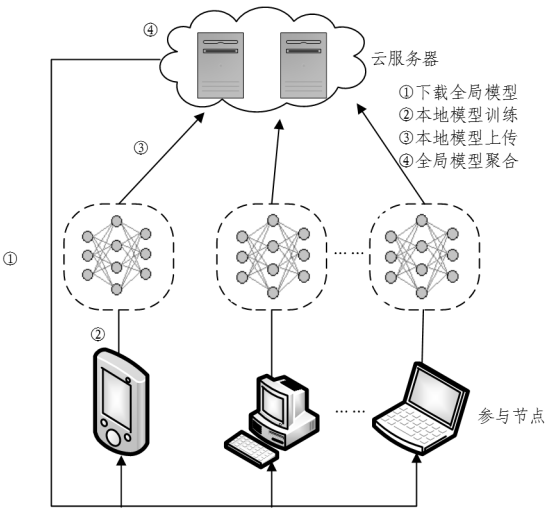

图 1 联邦学习模型架构

Fig. 1 Architecture of federated learning model

图中①表示参与节点从云服务器下载全局模型 $\theta(t-1)$。

图中②表示参与节点 m 使用本地数据进行本地模型训练,得到本地模型更新 $\theta_m(t-1)$。

图中③表示各参与节点将本地模型更新上传至云服务器。

图中④表示云服务器将接收的本地模型进行加权平均聚合,得到全局模型 $\theta(t)$。

2.2 大型语言模型

大型语言模型旨在认知、理解和生成人类语言。它们在大量的文本数据上进行训练,可以执行各种多样化的任务,包括文本总结、翻译、情感分析等。大型语言模型的特点是规模庞大,其包含数十亿的参数,用于帮助学习语言数据中的复杂形式。这些模型通常基于深度学习架构,如 Transformer 等,使得它们在自然语言处理任务上有良好的性能表现。

Google 在 2018 年首次提出 Bidirectional Encoder Represen-

tations from Transformers[3]，即 BERT 模型，提出了预训练的思想并且使用 Transfomer 编码器作为基础架构。它使用了一系列自注意力机制来捕捉输入序列中不同位置之间的关系，同时还使用前向神经网络[4]和残差连接[5]等技术来提升模型的非线性能力。继 BERT 模型之后，OpenAI 从 2018 年起发布生成式预训练语言模型 GPT(Generative Pre-trained Transformer)[6]，称为 GPT-1，可用于生成文章、图像、机器翻译、问答等各类内容。研究人员将模型训练划分为两个阶段：首先，在预训练阶段，模型通过处理大规模文本语料库来学习高质量的语言表示，并在这个过程中从海量无标记文本中提取出有意义的文本表示；接着在微调阶段，使预训练模型适应特定领域的标记数据。通过预训练模型生成高质量文本表示，并在微调阶段针对特定任务进行微调，可以显著提升模型在特定任务上的性能。

随着模型规模的扩大以及数据量的不断增加，GPT-1 的准确度以及泛化能力也有待提升。所以，Radford 等[7]针对当时的现状提出了 GPT-2，GPT-2 在设计之初引入了 zero-shot 学习的思想，使模型在没有任何显式监督的情况下学习任务，采用更直观的自然语言处理方式来处理指定任务。

然而，GPT-2 在实验中的性能表现较差。尽管 GPT-2 未能带来显著的改进，但也起到了承上启下的重要作用。2020 年，Brown 等[8]基于 GPT-2 做了进一步的研究，引出 GPT-3，假设语言模型已经提供了自然语言指令和(或)几个任务演示，它可以通过输入文本的单词序列来生成测试用例的预期输出，而不需要额外的训练或梯度更新。GPT-3 表现出了较强的语境学习能力，同时也引入了 sparse transformer，其具有1 750 亿的参数量，并且在实验中性能表现较好。

目前，最新的 GPT-4 模型[9]已经发展为一个大规模的多模态系统，可以接受图像和文本输入并输出文本。GPT-4 是一个基于 Transformer 的模型，经过训练可以预测文档中的下一个 token。该模型优于现有的大型语言模型，其理解和生成自然语言文本的能力得到提升，特别是在复杂的场景中可以呈现出良好的性能。自最初的 GPT 模型发布以来，GPT 系列经历了多次迭代和优化，逐渐提升了语言理解与生成的能力。表 2 列出了 GPT 系列语言模型从最初版本到最新版本的发展历程及其特点。

大型语言模型是自然语言处理领域的一项重大突破，其训练主要包括预训练(Pre-training)和微调(Fine-tuning)两个阶段。预训练阶段[10]在目标学习任务之前，使用大规模数据集和无监督学习的方法对模型进行初始化训练，通过学习大规模数据特征来为训练模型提供初始化参数和特征表示，为后续微调阶段的具体任务提供更好的初始状态，在解决训练过程中模型参数巨大问题的同时提高模型的学习能力以及泛化能力。微调是在特定任务或领域上对大语言模型进一步的训练，使用预训练模型为起点，在特定领域或在领域的标记数据集上继续进行训练。微调可以使模型更好地适应目标任务的要求，并且显著提高训练效率。

表 2 大型语言模型——GPT 系列发展过程

Table 2 Large language model—the development process of GPT series

模型	作者	时间	特点
GPT-1	Nazir 等[6]	2018 年 6 月	首次使用 Transformer 框架架构的 GPT 模型
GPT-2	Radford 等[7]	2019 年 2 月	引入 zero-shot 学习，采用直观的自然语言处理方式来指定任务
GPT-3	Brown 等[8]	2020 年 5 月	引入稀疏 Transformer(sparse transformer)，具有较强的语境学习能力
GPT-4	Achiam 等[9]	2022 年 11 月	经过预训练可以预测文档中的下一个 token，优于现有的大型语言模型，理解和生成自然语言文本的能力得到提升

2.3 联邦大语言模型训练过程

本节将介绍联邦大语言模型的训练过程，根据 2.2 节所述建立一个联邦学习框架，如图 2 所示。该框架通常包括一个云服务器和多个参与节点，通过模型训练聚合算法实现大语言模型的分布式训练。具体训练过程如下：

1)初始模型部署：初始阶段，每个参与节点包含一个现有的开源基础模型，将各模型作为共享全局模型的起点。

2)本地模型训练：每个参与节点利用其本地数据进行模型训练。这包括对用户文本数据以及本地语料库的微调，使模型更好地适应各自特定领域的任务特征。

3)本地梯度计算：参与节点进行本地训练，每个参与节点计算其本地模型的梯度，然后上传到云服务器进行全局聚合。

4)上传梯度聚合：本地计算的梯度上传到云服务器进行模型聚合。使用模型聚合协议，以保护参与方设备的数据隐私。

5)全局模型更新：中央服务器使用聚合后的梯度更新全局模型。使用随机梯度下降(SGD)等优化算法，以最小化全局模型在整个联邦学习训练过程中的损失。

6)迭代训练：从本地模型训练到全局模型更新的过程进行迭代，直至全局模型达到收敛。每轮迭代都包括本地模型训练、本地梯度计算、上传梯度聚合和全局模型更新这 4 个步骤。

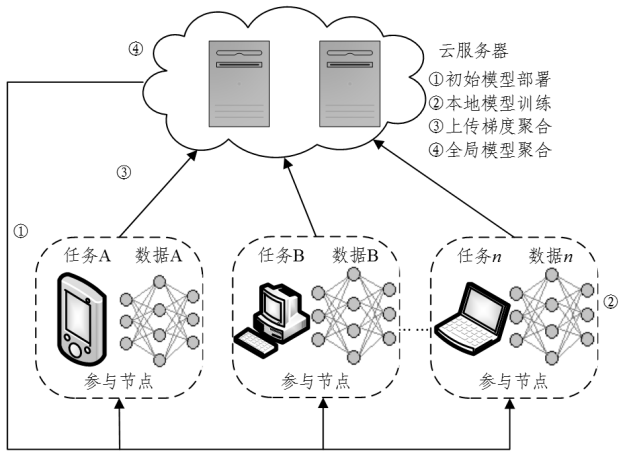

图 2 联邦大语言模型训练过程

Fig. 2 Training process of federated large language model

3 联邦大语言模型的传输挑战

联邦大语言模型面临海量的模型参数、庞大的显存需求以及巨大的计算和通信开销三大挑战。

3.1 海量的模型参数

在本地模型训练阶段，LLM 在大量无标注文本数据上训练语言模型，参数量达到数百亿、千亿甚至更多[11]。每一代 GPT 模型的参数量都呈爆炸式增长，表 3 列出了 GPT 系列参数量以及预训练数据量的对比结果。

表 3 GPT 系列主要参数量对比

Table 3 Comparison of main parameters of GPT series

模型	参数量	预训练数据量
GPT-1	1.17 亿	约 5 GB
GPT-2	15 亿	40 GB
GPT-3	1 750 亿	45 TB
GPT-4	千亿级	百 T 级

除 GPT 系列的大语言模型以外，Google[12] 提出的视觉语言模型 PaLM-E（Pathways Language Model with Embodied）是目前最大规模的多模态具身视觉语言模型（VLM），PaLM-E 的参数量高达 5 620 亿。与此同时，百度于 2021 年提出的 ERNIE3.0[13] 是一种强大的语义表示模型，该模型参数规模达到 2 600 亿，是目前全球最大的中文单体模型。ERNIE3.0 在文本分类、命名实体识别以及语义匹配等自然语言处理领域的任务中取得了优秀的性能。

因此用联邦学习框架来训练大语言模型的过程中会面临海量模型参数问题，在联邦学习中需要处理大规模的参数更新及梯度传输。例如，联邦学习中有 10 个客户端使用 BERT-base 对 30 个全局 epoch 进行学习，那么会产生大约 263 GB 的总传输量[14]。并且如果联邦学习中的参与节点数量较少，那么会导致每个参与节点承载庞大的模型参数量，而现有的参与节点（如手机、平板等小型移动设备）都无法存储如此大的模型参数量。

3.2 庞大的显存需求

在大模型推理阶段，显存的消耗主要来源于模型加载，即将 LLM 中所有的权重参数和偏差参数加载到显存中；在模型训练阶段，模型训练通常比模型推理需要更多的显存，因为涉及梯度计算和参数更新等大量计算，模型的全量参数需要加载比模型推理多 3 倍左右的显存大小。传统的数据并行模型下，模型训练会对每个参与方消耗固定大小的全量内存，而这部分内存并不会随着数据的并行而减小。这意味着即使参与节点仅负责部分数据的训练，它们仍需要较大的内存来存储整个模型的参数及其他状态。

在联邦学习架构下，参与节点存在内存资源不足等挑战。Rajbhandari 等[15] 对模型的内存消耗归纳总结为 3 种：优化器状态、梯度以及参数，这 3 种状态被称为 OGP 状态。在模型训练阶段，大部分内存被以下 3 种情况所消耗：激活、OGP 状态（即由优化器状态、参数梯度和参数本身组成的张量）以及临时缓冲区。同时，研究发现 LLaMA 7B 的总内存需求为 12 GB[16]，这类相对较小的模型对于一些手机、平板等移动设备来说都是难以满足的。那么对于一个拥有 15 亿参数的 GPT-2，至少需要 24 GB 的内存才能存储。表 4 列出了部分开源大语言模型的最低显存容量对比结果。并且参与节点在联邦学习架构下需要不断地与云服务器传输参数更新和梯度信息，也会消耗设备的内存和计算资源，进一步增加了显存消耗。因此，在联邦学习架构下进行大语言模型的训练，对于参与节点的显存性能都有着较高的要求，可以从模型结构压缩的角度解决显存在传输过程中的挑战。

表 4 开源 LLM 最低显存容量对比

Table 4 Minimum memory capacity comparison of open source LLM

开源大语言模型	最低显存容量/GB	推荐显卡
LLaMA-7B	12	RTX 4080
ChatGLM3-6B	14	RTX 4080
Qwen-14B	30	V100
Yi-34B	69	A100
Qwen-72B	145	多卡 A100 以上

3.3 巨大的计算及通信开销

联邦大语言模型不仅在海量模型参数传输方面存在挑战，而且在训练和分发的过程中会导致较大的计算和通信开销。在联邦学习训练过程中，大量的参与节点与云服务器之间需要交换更新模型梯度，这会导致巨大的通信开销[17]。这一过程受到网络带宽的限制并且会增加参与节点的掉线率，大语言模型的通信延迟也会受到一定的影响，因为庞大的参数量和复杂的模型结构需要更多的通信传输和更高的计算要求。研究表明，现有的联邦学习训练算法如 FedAVG，FedProx 和 FedNova 等存在一定的局限性[18]，无法有效满足大语言模型的训练需求。这些算法在处理海量模型参数和复杂模型结构时会导致训练效率低下和通信开销巨大的问题。因此，研究人员正致力于改进模型训练技术，以应对分散数据以及模型架构的复杂性，提高联邦学习架构下大语言模型的训练效率，降低模型训练和梯度传输过程中的通信和计算开销。

一种潜在的解决方案是将分布式模型训练技术应用于联邦大语言模型，以更好地适应大语言模型的分布式训练需求。这些分布式训练技术可以通过优化数据传输方式、改进模型分发策略以及设计更有效的模型并行机制来降低传输过程中的能耗。例如，一些研究人员提出了基于梯度压缩和稀疏化的算法，通过减少传输的梯度来降低通信开销，同时还可以保持模型性能。除此之外，还有一些研究致力于优化通信和计算资源的利用方式，例如，利用深度学习模型的稀疏性和低精度计算技术，可以减少通信和计算的负载，从而降低整体模型的传输开销。

4 联邦大语言模型传输优化技术

本章将着重介绍联邦大语言模型有关传输优化技术的已有研究和进展，将其按照方法原理进行分类梳理，并列举出关键优化技术（见图 3），包括模型参数微调、模型结构压缩以及

分布式并行处理3个方面。

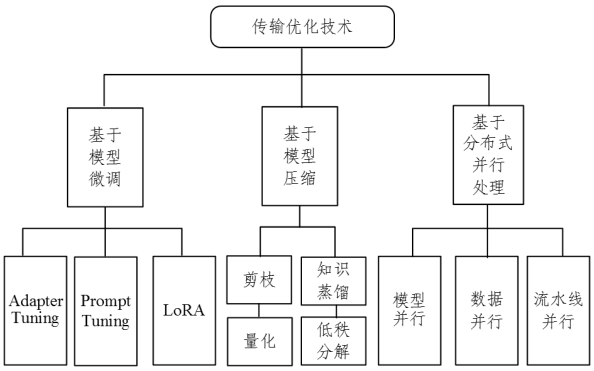

图3 典型的传输优化技术

Fig. 3 Typical transmission optimization techniques

4.1 基于模型微调的参数量传输优化方法

为解决海量模型参数量大的问题，研究人员提出高效参数微调(Parameter-Efficient Fine-Tuning，PEFT)技术[19]，该技术通过在预训练阶段添加Transformer模块来产生一个紧凑且可拓展的模型，这样能够只为每个任务添加一些可训练的参数并且无需重新访问旧任务即可添加新任务。实验表明，该技术大大减少了微调阶段需要处理的参数量，从而提高预训练模型在新任务上的性能并且显著降低模型训练的成本。目前，PEFT技术主要有Adapter Tuning[20]，Prefix Tuning[21]，Prompt Tuning[22]，P-Tuning[23]和LoRA[24]，因此研究人员从模型微调的角度出发，考虑将PEFT技术与联邦大语言模型相结合来解决模型参数量问题，基于模型微调的传输优化研究对比如表5所列。

表5 基于模型微调的传输优化

Table 5 Transmission optimization based on model fine-tuning

主要文献	针对数据场景		优化方法			优化目标			个性化框架
	IID	Non-IID	Adapter Tuning	Prompt Tuning	LoRA	提高效率	提高模型精度	降低能耗	
[26]	√	√	√	√			√		
[28]	√	√	√			√			
[29]	√	√	√			√		√	
[30]	√	√		√		√		√	
[31]		√		√		√			√
[32]		√		√		√	√		
[33]	√	√			√	√	√		√
[34]		√	√		√	√	√	√	√
[35]					√	√		√	
[36]		√			√	√		√	
[37]	√	√			√		√	√	
[38]	√	√			√		√	√	

4.1.1 基于Adapter Tuning的参数量传输优化方法

上述基于Transformer的预训练模型改变了大语言模型的性能，FedNLP(首个联邦学习赋能NLP的开源框架)的出现解决了下游任务数据隐私性问题，但由于庞大的模型参数量以及高昂的计算、通信成本，FedNLP的通信延迟比一般的联邦学习模型至少高出一个数量级[25]。该优化的主要目的是减少模型训练参数量，降低通信延迟以达到目标精度。首先将Transformer确定为预训练阶段中唯一可调的模块，并且冻结其他剩余的模型参数，大大减少了模型中的可训练参数。尽管Transformer显著减少了训练过程中的模型参数量，但不会自动选择Transformer的配置空间使其达到最佳的收敛时间，所以文献[26]提出FedAdapter框架，该框架的关键思想是冻结整个原始模型，在不同位置插入几个小模块。

FedAdapter框架中引入两个关键设计来加快模型收敛速度。1)Progressive training(渐进式训练)：FedAdapter在训练的初始阶段，只在接近模型输出处插入小容量的Transformer，以较低的训练成本学习浅层知识。当模型收敛趋于平稳时，FedAdapter会将底层Transformer添加到训练中并且增加其宽度，这与curriculum learning[27]有相似之处。通过构建一个Adapter，在整个训练过程中自动调整深度和宽度，加速模型收敛。2)Sideline trials(副业实验)：FedAdapter除了对当前训练模型进行配置外，还要探测下一个配置并且抉择何时进行切换。所以，FedAdapter中要用不同的Transformer配置来训练模型并比较训练的进度，从多个试验组中选出收敛速度最快的Transformer配置作为下一个切换的候选配置。实验证明了FedAdapter框架可显著减少模型训练参数量并加快模型收敛。

Kim等[28]使用Adapter通过最少的参数量来调整预训练模型，研究人员通过冻结预训练的语言模型仅训练模型中的adapter layers和classification head来提高传输效率。首先准备并下载预训练模型，使每个参与节点具有相同的模型结构和参数量；之后每个参与节点将经过训练的adapter layers和classification head上传到云服务器进行模型聚合；最后参与节点下载聚合后的全局模型开始新一轮的学习。该方法与未使用Adapter的预训练大语言模型相比，在不降低性能的情况下传输时延减少了约98%，训练时延减少约20%。文献[29]引入了FedPEFT，该方法冻结大多数模型权重，通过调整为特定下游任务定制的最小参数子集来减少传输过程中的模型参数量。FedPEFT评估了4种PEFT技术的效率，包括Head-tuning，Bias，Adapter和ViT-B。研究表明，这些有针对性的大模型微调方法显著减少了通信开销，达到99%以上。

基于Adapter Tuning的参数量优化传输均是通过冻结部分参数来减少模型传输时的参数量，从而降低模型传输成本。

4.1.2 基于Prompt Tuning的参数量传输优化方法

基于Prompt Tuning的传输优化方法是将联邦学习与Prompt Tuning相结合的一种调整参数量的优化方法，即分

布式环境中在不修改预训练模型的情况下调优 soft prompt。文献[30]通过冻结参数量巨大的预训练模型(PLM)，仅聚合和调整一些 soft prompt 来减少模型参数量并降低通信成本。

在一般 Prompt Tuning 中，用 F 和 P 分别表示 PLM 和 soft prompt 中具有可训练的参数，那么第 t 轮的全局模型参数可以表示为：$\theta_t=F_t+P_t$。在 FedPrompt 中，研究人员用固定的 PLM 参数来学习数据集 D 上的一组全局模型 θ，目的是降低参与方 k 的经验损失。

$$Lk(P)=E_{(x^{(i)},y^{(i)})\in D_k}\ell_k(f(x^{(i)},P,P),y^{(i)}) \tag{1}$$

首先服务器初始化整个模型，然后分发给各个参与方。在第 t 轮开始时，服务器按比例 C 选择本轮的参与方，将全局 soft prompt 参数 P_t 分发给客户端，每个被选中的客户端 k 将本地 P_{t1}^k 替换为 P_t，即 $P_t^k=P_t$。由于 PLM 是固定的，因此 $F_t^k=F_{t1}{}^k$。然后每个客户端用 optimizer 只针对 P_t^k 进行本地训练，并将其更新后得到的 soft prompt 参数并行发送回服务器，服务器进行如下聚合操作：

$$P_{t+1}\leftarrow\sum_{k=1}^{\lceil C\cdot K\rceil}\frac{n_k}{N_t}P_t^k \tag{2}$$

文献[30]中的实验证明了 FedPrompt 能将通信成本压缩到全参数微调的近 0.01%，它在大大降低通信成本的同时精度仅下降 1%左右。所以 FedPrompt 适用于通信和存储约束受限的设备，只需要少量的局部训练和通信轮次就能获得良好的全局模型。

近年来，研究人员针对非独立同分布(Non-Independently Identically Distribution Non-IID)的数据场景，展开针对减少模型传输参数量的方法研究。文献[31]引入一种客户端特定提示生成的个性化 FL 框架——pFedPG，该框架在云服务器上部署个性化提示生成器以生成特定于客户端的可视化提示，优化了本地个性化提示适配和本地个性化提示生成这两个阶段。前者使模型适应每个客户端，后者则为每个客户端生成个性化模型，该框架可以提高个性化模型传输过程中的计算和通信效率。同时，文献[32]针对参与方数据分散的问题，引入一种参数高效的自适应优化提示调优方法——FedPepTAO。该方法提出一种评分机制来分析各层与大模型输出的相关性，用来代表各层的重要程度。在不降低模型准确率的情况下评估每一层的分数，为 FL 选择合适的层，使用其中的提示参数进行模型更新，其他参数仅在参与方内更新，这样可以减少模型更新传输时的参数量，从而在有限的通信成本下获得较好的性能。此外，设计了一种新颖的自适应优化方法在服务器端控制变量，FedPepTAO 显著减少了传输时的模型参数量并且效率提高了 97.59%，准确率提高了 60.8%。

4.1.3 基于 LoRA 的参数量传输优化方法

除了利用 Adapter Tuning 和 Prompt Tuning 这两种参数微调方式外，LoRA 微调常用于大语言模型的个性化联邦学习框架中。文献[33]提出基于 LoRA 微调的模型异构个性化联邦学习框架——FedLoRA。该方法为每个参与节点的本地个性化模型分配一个同构的小型低秩线性 Adapter，提出了局部异构模型和同构适配器的迭代学习方法支持全局知识与局部知识的双向迁移。文献[34]结合了两种 PEFT 技术，即 Adapter 和 LoRA 来减少模型传输时的参数量，参与方的 Adapter 参数发送至云服务器进行全局聚合，而 LoRA 参数保留在参与节点中维护本地模型的个性化。它们引入了两种个性化联邦微调算法，即个性化联邦指令调优算法(PFIT)，采用强化学习来微调本地 LLM 以实现个性化；以及个性化联邦任务调优算法(PFTT)，可以利用全局适配和本地低秩适配(LoRA)协同微调本地 LLM，无需聚合来降低传输过程中产生的能耗。该方法实现了在减少需要更新的参数量的同时也可以保持参与方的本地模型个性化。与 FedLoRA 相比，该方法在降低通信和计算能耗的同时提高了模型精度。

除了在个性化联邦学习框架中应用 LoRA 技术外，文献[35]引入低参数联邦学习(LP-FL)框架，利用 LoRA 构建紧凑的可学习参数、高效的局部模型微调以及高效的全局模型聚合。首先，每个参与节点使用各自的局部标记数据对全局模型进行微调，全局模型是一个带有 LoRA 模块的 LLM，通过训练 LoRA 的参数以减少训练时的模型参数量。训练完成后，每个参与节点将自身的 LoRA 参数传递给云服务器进行聚合后重传到各个参与节点；每个参与节点使用接收到的 LoRA 参数来更新自身的局部模型，并从自身局部未标记数据集中选择一部分数据进行标注，从而扩展已标记的数据集来进行进一步的训练；通过迭代的 soft prompt 分配实现基于大语言模型的低参数量联邦学习。该方法与全参数联邦学习相比具有更好的性能，同时降低了单个设备上的计算成本和通信开销。

以上方法考虑更多的是独立同分布(Independently Identically Distribution，IID)数据场景，文献[36]引入 SLoRA 方法，这种方法利用一种数据驱动的初始化技术，成功解决了 LoRA 在 Non-IID 数据场景下的关键限制。它通过比率来调节每个参与节点的更新影响，从而抵消 Non-IID 数据导致的“漂移”。SLoRA 可以在最小的通信、时间成本下保持与全参数微调相同的模型性能。文献[37]引入 FLoSS 方法，该方法实现了在训练时上传和下载模型梯度阶段采用 LoRA 技术，在降低通信能耗的同时保留模型的实用性，这是第一个将非结构化稀疏性应用于 LoRA 以实现高效联邦微调的方法。该方法最多降低 90%的通信能耗，同时能够提高模型精度。文献[38]提出基于 LoRA 和 P-Tuning 的分段微调方法，并通过将 LLM 按层拆分以优化传输成本、降低计算能耗，然后提出将 LoRA 与 Sparsification Parameter Fine-tuning(SPF)技术相结合来进一步提高下游任务的准确性。除此之外，该工作还将 TEE 与轻量级加密相结合，以确保训练过程中的安全性及隐私性。

4.2 基于模型压缩的传输优化方法

模型压缩可以在不完全损失模型性能的情况下压缩模型结构，将大型数据密集型模型转化为适合在资源受限移动设备上存储的数据紧凑型模型，从而使它们在联邦学习中以较低的存储成本进行部署[39]。因此，研究人员考虑将常见的模型压缩技术与联邦大语言模型相结合。基于模型压缩的传输优化研究对比如表 6 所列。

剪枝(Pruning)指移除模型中不必要或多余的部分，以使模型结构更加简洁高效[40]。通过识别和移除对模型预测贡献较小的冗余参数，可以显著减少模型训练的计算量，从而加快

大模型在联邦学习中的推理速度，减少节点中的内存消耗。剪枝后的模型占用更少的存储，便于在资源有限的设备上部署和运行。剪枝分为非结构化剪枝和结构化剪枝。文献[41]出非结构化剪枝即对独立权重或神经元进行剪枝，旨在移除个别权重，模型压缩比较高，但缺点在于精度不可控，要通过专门的稀疏矩阵运算库或硬件来实现，且大部分的运算操作是由激活函数映射产生，因此在模型推理阶段很难压缩存储内存。文献[42]引入 SparseGPT，将非结构化剪枝问题转化为一系列大规模稀疏回归问题，并用一种创新的近似求解器来解决问题。该文中提出的 SparseGPT 在几乎不影响模型准确性的情况，能够实现高达 60%的稀疏度。文献[43]提出可以在训练好的模型中修剪较小权重的通道，再通过微调模型使其达到目标精度，不需要专用硬件来实现，可保留部分模型结构但实现算法较复杂。文献[44]提出一个专为 FL 框架设计的模型剪枝过程——PruneFL，首先在“预热”阶段选择一个有能力的参与方，使用其本地数据对模型进行修剪；其次在“自适应修剪”阶段，服务器通过在多个轮次中删除或重新引入参数来周期性地调整模型。文献[45]引入了与 PruneFL 类似的方法，但该方法在初始化阶段利用参与方数据的批处理规范化值作为共享初始化的基础，从而增强对不同参与方数据分布的适应性。文献[46]引入 LLM-Pruner，利用梯度信息和 Hessian 矩阵来为 LLMs 中的耦合结构（如注意力头）做出剪枝决策。修剪后的 LLMs 通过 LoRA 进行微调，以恢复模型性能。通过结构化修剪的压缩 LLMs 可以直接部署并在标准计算框架上执行，而无需额外调整，因为结构化修剪移除了 LLMs 中的整个结构。文献[47]引入了一种一次性剪枝策略，该方法将剪枝视为稀疏回归问题，并使用近似稀疏回归求解器来解决，实现了非结构化稀疏性。文献[48]提出了一种新的剪枝度量，它根据每个参数的权重大小以及相应输入激活的范数的乘积进行评估，该乘积通过一个小型校准数据集来近似计算。该度量用于线性层输出内的局部比较，使得可以从大语言模型中移除优先级较低的权重。因此，剪枝技术可以有效解决大语言模型庞大的显存需求方面的问题。

表 6 基于模型压缩的传输优化

Table 6 Transmission optimization based on model compression

主要文献	针对挑战		优化方法				优化目标			
	海量模型参数	庞大显存需求	剪枝	知识蒸馏	量化	低秩分解	提高效率	提高模型精度	降低能耗	降低内存占用
[42]		√	√							√
[43]		√	√				√		√	√
[45]		√	√					√	√	√
[46]		√	√				√			√
[47]		√	√				√		√	√
[48]	√	√	√				√			√
[52]	√	√		√						√
[53]	√	√		√				√		√
[54]		√		√				√		√
[57]		√			√			√		√
[58]		√			√				√	√
[59]		√			√		√			√
[60]	√	√			√		√	√		√
[61]	√	√			√		√			√
[62]		√			√		√			√
[63]		√			√		√			√
[68]		√				√	√			√
[69]		√				√	√			√
[70]	√	√				√	√			√

知识蒸馏（Knowledge Distillation，KD），通过从一个复杂的模型（称为教师模型）向一个简化的模型（称为学生模型）转移知识来实现。将大模型上的知识提炼为一个小模型，然后用训练大模型相同的方式对小模型进行泛化。其核心思想是使用复杂模型的预测分布或软标签来训练简化的模型，从而提升简化模型的性能。文献[49]提到几乎所有通过训练的深度神经网络集合都可以被提炼成一个相同大小的神经网络，这样更容易进行模型部署。知识蒸馏的方式一般分为 3 种：离线蒸馏（Offline Distillation）、在线蒸馏（Online Distillation）以及自我蒸馏（Self-Distillation）[50]。离线蒸馏主要集中于改进知识转移的不同部分，优点在于简单易行，例如，教师模型使用位于不同节点上的不同数据集训练一组模型，可以提取知识并将其存储在缓存中。但缺点在于复杂的高容量教师模型训练时间较长，而离线蒸馏中对学生模型的训练通常在教师模型的指导下是有效的。此外，学生模型很大程度上依赖于教师模型。为了克服离线蒸馏的局限性，文献[51]提出了在线蒸馏以进一步改善学生模型的性能，特别是在没有高性能教师模型的情况下。在线蒸馏使教师模型和学生模型同时更新，是一种具有高效并行计算功能的端到端训练技术。自我蒸馏是在线蒸馏的特殊情况，教师模型和学生模型采用相同的模型结构，每个学生模型都可以使用教师优化来蒸馏先前模型的知识。因此可以将联邦学习与知识蒸馏技术相结合并应用于大语言模型，这种基于知识蒸馏的高效联邦学习方法被称为 FedKD[52]。在文献[52]中，参与方与云服务器不直接通过大模型通信，而是由一个小模型和一个大模型相互学习来提炼知识，其中只有小模型被传递到不同客户端共享并协同学习，这样做可以显著降低模型参数量，并且能够更好地适应参与节点局部数据的特点。文献[53]提出一种联邦代理微调——FedPFT，这是一种通过两种关键模块——子调频构建模块和子调频对准模块——来提高联邦代理适应下游任务的

能力的新方法。在子调频对准模块采用分层压缩方法,通过强调关键神经元来促进所有层的全面调频微调;在子调频对准模块中对模型微调前和微调过程中分别进行两步蒸馏——层级蒸馏和神经元级蒸馏,FedPFT 性能接近使用全模型微调的 FedPETuning 的性能。文献[54]引入 FedBiOT 算法,该算法通过对部分模型微调来降低计算和通信能耗。将模型微调与模型压缩技术相结合,通过模型压缩将模型分为两部分,一部分复制未压缩的 LLM 以实现模型性能,另一部分侧重于学习特定领域的语言知识,解决了参与方无法加载完整 LLM 的问题,使参与方不需要在访问完整模型的情况下进行微调。该算法在计算和通信开销以及最终模型精度方面都有显著的改善。

量化(Quantization)是将传统方法中的浮点数转换为整数或其他离散形式,以减轻模型的存储和计算负担。量化通过权值共享来降低模型参数的大小,从而进一步简化模型结构和降低计算量[55]。依据量化压缩模型的应用阶段,可分为以下 3 种方法:

1)量化感知训练(Quantization-Aware Training,QAT),主要目标是将量化目标集成到模型的训练过程中,这种方法使 LLM 在训练过程中适应低精度表示,提高处理由量化引起的精度损失的能力[56]。深度学习通常以全精度(32 位)运行,但现在通常不需要如此高的计算精度。文献[57]提出 LLM-QAT,利用预训练模型生成的结果来实现无数据蒸馏。LLM-QAT 能够将带有量化权重和 KV 缓存的大型 LLaMA 模型蒸馏为仅有 4 位的模型,这一突破性的结果证明了制造准确的 4 位量化 LLM 的可行性。文献[58]引入一种具有周期性平均和量化的通信高效联邦学习方法——FedPAQ,通过量化参与方在上传到云服务器之前的更新来减少传输时的开销。文献[59]引入 FedOBD 框架,通过将大模型分割成语义块使参与节点与云服务器能够选择性地交换量化块。FedOBD 评估块根据单个参数的重要性来选择性地挑选块,并且结合先进的神经网络自适应确定性量化(NNADQ),显著降低了通信开销,同时保持了模型性能。与 FedPAQ 相比,FedOBD 可以降低近 50%的通信成本。

2)量化感知微调(Quantization-Aware Fine-tuning,QAF),QAF 是在微调过程中对 LLM 进行量化,主要目标是确保经过微调的 LLM 在量化为最低位宽后仍保持原始模型的推理准确性、任务特定的表现和整体性能。通过将量化整合到微调中,旨在在模型压缩和保持性能之间取得平衡。Kim 等提出的 PEQA[60] 和 Dettmers 等提出的 QLORA[61] 都属于量化感知参数高效微调(PEFT),这些技术侧重于促进模型压缩和加速推理。其中,文献[60]采用了双阶段:在第一阶段,全连接层的每个参数矩阵被量化为低位整数矩阵和标量向量;在第二阶段,对每个特定下游任务的标量进行微调。文献[61]引入了新的数据类型、双重量化和分页优化器等创新概念。通过在不影响性能的情况下节省内存,QLORA 使得大型模型可以在单个 GPU 上进行微调,同时在 Vicuna 基准测试上实现了最优的结果。

3)后训练量化(Post-Training Quantization,PTQ),PTQ 是在 LLM 的训练阶段完成后对其参数进行量化,主要目标是降低 LLM 的存储和计算复杂性,无需对 LLM 架构进行修改或重新训练。PTQ 的主要优势在于简单性和高效性,但在量化过程中可能会有一定程度的精度损失。在 PTQ 中,有些方法仅对 LLM 的权重进行量化,可在提高效率的同时减少计算需求。文献[62]提出 LLM.int8()对 LLM Transformers 中的矩阵乘法采用 8 位量化,在推理过程中有效地减少了 GPU 内存使用量,同时保持较高的性能精度。该方法采用矢量量化和混合精度分解来处理异常值以实现高效的推理。文献[63]发现,对于 LLM 的性能,权重不是最主要的,保护 1% 的权重可以大大减少量化误差。在此基础上 Lin 等[63] 提出了 AWQ,AWQ 采用了激活感知方法,考虑与较大激活幅度对应的权重通道的重要性,这在处理重要特征时起着关键作用。该方法采用逐通道缩放技术来确定最佳缩放因子,从而在量化所有权重的同时最小化量化误差。

但量化技术也存在一定的缺点,对于除了传统的卷积算子以外的其他复杂度很高的算子,量化技术无法适应复杂多变的连接结构以及运算操作。例如在 8 比特量化中,一些 BERT-base 基础模型反而会有性能损失[64],并且会给模型带来明显的量化噪声。

低秩分解(Low-Rank Factorization) 指通过合并维数和施加低秩约束的方式稀疏化卷积核矩阵。由于权值向量大多分布在低秩子空间,因此可以用少数的基向量来重构卷积核矩阵,达到缩小存储空间的目的[65]。低秩分解的核心思想是对一个大的权重矩阵 $\boldsymbol{W}$ 进行分解,得到两个矩阵 $\boldsymbol{U}$ 和 $\boldsymbol{V}$,使得 $\boldsymbol{W}\approx\boldsymbol{U}\times\boldsymbol{V}$,其中 $\boldsymbol{U}$ 是一个 $m\times n$ 矩阵,$\boldsymbol{V}$ 是一个 $n\times k$ 矩阵,其中 n 远远小于 m 和 k。$\boldsymbol{U}$ 和 $\boldsymbol{V}$ 的乘积近似于原始的权重矩阵,从而大幅减少了参数数量和计算开销。Jaderberg 等[66] 将 $w\times h$ 的卷积核分解为 $w\times 1$ 和 $1\times h$ 的卷积核,将学习到的字典权重线性组合重构,得到输出 feature map。Lebedev 等[67] 提出 CP 分解,即位 tensor 分解,将四维卷积核分解成 4 个 1×1,$w\times 1$,$1\times h$ 和 1×1 的卷积,即将 1 层网络分解为 5 层低复杂度的网络层,可以显著缩小存储空间。Wu 等提出的 Zeroquant-fp[68] 以及 Zhang 等提出的 LoRAPrune[69] 都能在实现压缩模型的同时保持模型的良好性能。文献[70]引入的 FwdLLM 框架使模型能在存储约束更严格的移动设备上使用。该框架将无反向传播(BP)与参数高效训练方法相结合,不依靠传统的反向传播来计算精确的梯度。无 BP 方法引入微小和自生成的扰动,评估这些扰动与原始未扰动模型相比如何影响模型预测。实验证明 FwdLLM 有显著的性能增强,并且在内存占用方面减少了近 14/15 倍。

由于低秩分解是通过将原始数据矩阵表示为低秩近似来实现降维,部分原始数据的信息会被舍弃或者近似表示,尤其是那些对模型中微小变化敏感的数据特征,因此低秩分解可能会导致关键信息的损失,从而影响分析或预测的准确性[71]。此外,低秩分解的稳定性也是一个需要考虑的问题。特别是在模型包含大量噪声或者异常值时,使用低秩分解技术会导致模型的性能表现下降。

4.3 基于分布式并行处理的传输优化方法

在大规模参数量的模型训练中,分布式并行处理对于联邦大语言模型来说十分关键,这些并行技术能解决大语言

模型在传输过程中出现的问题及面临的挑战。在2021年，NVIDIA提出了分布式训练超大规模模型的3种并行技术[72]：模型并行(Tensor Model Parallelism)、数据并行(Data Parallelism)、流水线并行(Pipeline Model Parallelism)。这些技术可以高效应对联邦大语言模型传输过程中参数量以及模型结构方面的挑战。基于分布式并行处理的传输优化研究对比如表7所列。

表7 基于分布式并行处理的传输优化

Table 7 Transmission optimization based on distributed parallel processing

主要文献	优化方法			优化目标		
	模型并行	数据并行	流水线并行	提高效率	提高模型精度	降低能耗
[75]	√			√		√
[76]	√			√		
[77]	√			√		√
[80]			√	√	√	√
[81]			√	√		√

模型并行[73]是一种对模型资源的分配问题，可以考虑将庞大的模型架构按层分成若干份，把每一层载入不同的节点中，即每个不同的节点计算模型的不同层的梯度。通过这种方法，可以减轻单个节点的训练压力并提高模型传输效率。但分布式并行处理需要着重考虑通信开销，模型并行带来的通信开销远超数据并行。所以模型并行一般只适用于单机内存无法容纳的大模型，并且具有局部连接结构的模型比全连接结构更适合模型并行，因为这类模型的通信需求较低。同时，对于模型的切分主要有两种[74]：横向切分和纵向切分。横向切分指将模型的每一层都拆分到多个参与设备上，相当于这一层在每个设备上占用的内存量减少；纵向切分指将模型按层次分到多个参与设备上，一个设备可以放一层或者连续多层。文献[75]引入一个分布式训练框架——TITANIC，首先选择最佳的客户端子集，接着将大语言模型划分为不同层并分发给多个客户端进行模型训练及微调，可以在训练过程中达到损失最小化。文献[76]在自注意块和一个多层感知器(MLP)中引入模型并行性，突破了传统的单个GPU训练模型的限制。文献[77]提出一个负载均衡的模型并行框架——Sky Computing，用于实现权重分配的自适应性，在训练64个节点的160层BERT模型时，Sky Computing的训练时间缩短了55%。

数据并行[78]是一种用于加速大规模模型训练的分布式计算策略，其核心原则是Single Program Multiple Data (SPMD)。在数据并行的环境中，整个模型任务被分割成多个子任务，每个子任务在不同的数据子集上执行相同的计算操作。数据并行能够充分利用多个处理单元的计算能力，从而显著提高训练速度和效率，在处理大规模数据时尤为明显。

数据并行适用于各种大规模数据处理场景，如大语言模型的训练过程。在分布式环境中，每个参与训练的节点(通常是不同的服务器或GPU)负责处理数据集的一个子集，并独立地执行模型的计算操作。一旦每个处理单元完成了其计算任务，它们的计算结果可以被汇总或者合并，以产生最终的模型参数更新。尽管数据并行具有显著的优势，能够提高计算效率并且优化模型传输过程，但也存在一些缺点。首先，数据并行要求每个参与节点上都存储完整的模型副本，这意味着每个节点的显存需求较高，尤其是在处理大规模模型时会限制系统的可扩展性。其次，在数据分布不均匀或者网络带宽有限的情况下，数据并行会影响整体训练的效率和稳定性[79]。

Huang等[80]引入了一个名为GPipe的管道并行(又称为流水线并行)，用于训练大型神经网络。将小批量流水线用于大模型的训练，通过在单独的加速器上对不同层的子序列进行管道化，GPipe扩展了各种灵活性来高效地实现大规模网络。并且GPipe采用新颖的批处理流水线算法，使训练可以得到更好的模型质量以及更快的收敛速度。在异构设备的分散大模型训练中使用管道并行技术是可行的，管道并行技术可以将模型训练过程划分为多个阶段，每个阶段在不同的设备上并行进行计算，从而提高训练效率及吞吐量，降低通信及计算开销。此外，Harlap等[81]引入PipeDream作为针对GPU的并行训练系统，通过在多台机器上流水线执行来实现并行计算。流水线并行计算避免了在大模型训练时导致较高的通信-计算比。实验证明，PipeDream在通信开销方面最多可减少95%，能有效均衡通信与计算成本。但流水线并行技术依赖较复杂的硬件结构，对节点的硬件资源要求较高。在联邦大语言开源平台中，FedLLM[82]利用FedML平台在单个数据中心的专用GPU集群中处理训练数据，使用DeepSpeed在多节点多GPU环境中实现高效并行训练。

5 联邦大语言模型典型系统

随着联邦大语言模型的不断发展，目前已经出现了一些开源平台，如FATE-LLM、FedLLM、FederatedScope-LLM、PrimiHub联邦语言大模型、OpenFedLLM以及FedLLM-Bench。开源平台与传输优化技术相对应的总结如表8所列。

表8 开源平台及传输优化应用

Table 8 Open source platforms and transmission optimization application

开源平台	模型微调			模型压缩			分布式并行处理			应用领域	
	Adapter Tuning	Prompt Tuning	LoRA	剪枝	知识蒸馏	量化	低秩分解	模型并行	数据并行	流水线并行	工业界	学业界
FATE-LLM[83]		√	√		√	√					√	
FedLLM[82]	√	√	√							√	√	
FederatedScope-LLM[84]		√	√								√	
PrimiHub联邦语言大模型	√	√	√								√	
OpenFedLLM[85]			√			√						√
FedLLM-Bench[86]												√

5.1 FATE-LLM

2023 年 4 月,联邦学习隐私开源平台 FATE(Federated AI Technology Enabler)推出了 FATE-LLM[83],这是一个用于大型语言模型的工业级联邦学习框架。该框架促进了大型语言模型(FedLLM)的联邦学习,利用高效参数微调(PEFT)方法促进大语言模型的高效训练并且通过隐私保护机制在训练和推理过程中保护数据隐私。

FATE-LLM 作为 FATE 的子开源模块,包含 3 个组件:Communication-Efficient Hub, FedLLM Model Hub 和 FedLLM Privacy Hub。其中,Communication-Efficient Hub 中是将多种 PEFT 方法集成到 FedLLM 中,包括 Adapter, Prompt,KD 和 Quantization 等,可以降低联邦学习系统中客户端的局部模型参数以及模型的存储大小,从而达到传输优化效果。FedLLM Model Hub 中集成了多种主流 LLM,包括 BERT,GPTs,LLaMA 等,这些 LLM 具有不同的架构和大小,可以应用于不同的场景。FedLLM Privacy Hub 中集成了各种隐私和安全保护技术,包括安全聚合、差分隐私和多方计算等,用来保护用户的数据隐私和模型安全。作为一个开源平台,FATE-LLM 的出现为研究和相关行业社区之间的合作提供了一个高效的学习平台。

5.2 FedLLM

2023 年 4 月,FedML 平台(FedML, Inc. 是由华人主导的国际化团队,起源于美国南加州大学,是全球范围内研究联邦学习的早期机构之一)发布了 FedLLM[82],使用该平台在专有数据上构建自己的大型语言模型。FedLLM 是一个 MLOps 支持的训练 pipeline,允许在专有数据上构建特定于领域的大语言模型,该平台支持数据协作、计算协作和模型协作,并支持在集中式和地理式分布 GPU 集群上进行训练,也可以以联邦学习方式对数据孤岛进行训练。

FedLLM 使用了 3 种创新方法:1)FedLLM 利用 FedML 平台在单个数据中心的专用 GPU 集群中处理训练数据,其中使用 DeepSpeed 用于在多节点多 GPU 环境中实现高效并行性。FedML 作为后台负责处理各种工作,可作为调度器、优化器以及质量控制器。2)FedLLM 利用 FedML 功能在异地分布式群集中运行训练或提供作业,当单个数据中心没有足够的 GPU 节点来处理训练数据时,可以合并现有的 GPU 资源来训练大型 LLM。并且 FedML 平台中有许多优化方法可以用来降低计算成本以及模型传输成本,如模型压缩、PEFT 等优化方法。3)FedLLM 也利用 FedML 的联邦学习平台,实现联邦学习功能。通过联邦学习将数据在节点本地进行训练,从而缓解传统机器学习固有的隐数据迁移以及传输成本较高的问题。

5.3 FederatedScope-LLM

2023 年 9 月,Kuang 等[84]实现了开源联邦大语言模型平台——FederatedScope-LLM(FS-LLM)。该平台支持在各种 FL 场景下联合微调 LLM,自动化数据集预处理、联邦微调执行或模拟的过程并且提供了全面和现成的 PEFT 算法实现和通用编程接口。

FS-LLM 有 3 个创新点:1)FS-LLM 封装了来自不同领域的各种联邦微调数据集的集合,有一个完整的流水线,具备可调整的异构级别,并包含一系列评估任务,用于评估在联邦学习场景中微调 LLM 算法的性能表现。2)FS-LLM 为 LLM 提供了全面的联邦微调算法,具有较低的通信成本和计算成本以及通用的编程接口,能在支持客户端可以或不能访问完整模型的两种场景的同时降低模型训练传输时的通信开销。3)FS-LLM 配备了一个优化的联邦微调训练范例,用于 LLM 实现可定制的效率提升(如内存消耗减少和多 GPU 并行)和跨学科研究潜力(如 pFL 和 FedHPO),并通过大量的实验证明该开源平台的实用性。

5.4 PrimiHub 联邦学习大模型

2023 年 4 月,原语科技在 PrimiHub 上开源了联邦学习大模型,实现了基于联邦学习的大模型训练和预测,可以理解和生成文本,为用户提供更丰富、精确和个性化的内容和服务。PrimiHub 可以让用户在自己的设备上参与联邦学习,同时体验大模型的智能服务。

PrimiHub 联邦学习大模型有 3 个创新点:1)PrimiHub 联邦学习大模型基于 ChatGLM6B,其参数量大,效果好。它具有超过 60 亿个参数,是目前最大的中文预训练模型之一,也是目前最先进的多模态预训练模型之一。在自然语言处理、计算机视觉、语音识别等任务上都取得了良好的性能。2)结合了 PEFT 技术,可以在保持大部分参数固定的情况下,通过调整部分参数来实现和调整全部参数一样效果的模型优化,从而显著降低了模型训练和传输时所需的通信和计算开销,让用户在消费级的显卡(如 NVIDIA GeForce RTX 3070)上就能体验联邦大语言模型的流程,轻松地进行联邦学习,降低了用户参与联邦学习的门槛和成本。3)该模型基于新的 PrimiHub SDK,这是一个开源的、高效的联邦学习软件开发工具包,用户输入一行指令即可自动完成大模型在联邦学习中的分布式训练和更新,提升了用户体验感以及便捷性。

5.5 OpenFedLLM

2024 年 2 月,Ye 等[85]构建了一个简洁集成易于研究的联邦大语言模型框架——OpenFedLLM,该框架涵盖联合指令调优(FedIT)、联邦值对齐(FedVA)、7 个经典的联邦学习基线以及 8 个语言训练数据集来支持不同领域的训练,提供了 30 多种评估指标来实现全面的评估,进而实现通过联邦学习在去中心化私有数据上训练大型语言模型。OpenFedLLM 有 3 个创新点:1)联邦指令调优(FedIT)中,每个参与方有多个数据样本,其中每个样本是一个(instruction, ground-truth response)对。在局部模型训练期间,模型预测给定具有指令的模板响应,其中基础 LLM 被冻结,而只有少数可学习参数被更新(如使用 LoRA 微调),从而减少模型参数量并优化模型传输过程。2)联邦值对齐(FedVA)中,每个客户机持有多个数据样本,其中每个样本由一条指令、一个首选响应和一个非首选响应组成。在局部训练期间,训练模型与首选响应保持一致,同时远离非首选响应,其中基础 LLM 被冻结,只引入少量可学习参数(如使用 LoRA 微调)。在通信过程中,只有一组可学习的参数被通信和聚合。3)研究表明,该框架对 LLM 和 FL 社区均友好,且基于 OpenFedLLM 方法训练出的模型始终优于单独训练的模型,提升了模型训练质量及传输效率。

5.6 FedLLM-Bench

Ye 等[86]于 2024 年 6 月提出的 FedLLM-Bench 是首个为联邦学习大语言模型(FedLLM)提供的真实基准,旨在解决现有 FedLLM 研究中缺乏真实数据集和基准测试的问题。FedLLM-Bench 基准包含 8 种代表性基线方法、4 个训练数据集以及 6 个评估指标,为 FedLLM 社区提供了一个全面的测试平台。

FedLLM-Bench 的核心贡献在于数据集,包含 3 个用于联邦指令调优的数据集(用户注释的多语言数据集)和一个用于联邦偏好对齐的数据集(如用户注释的偏好数据集),这些数据集继承了以下多样性:

1)语言:数据集涵盖了来自不同语言的数据,模拟了多语言协作的真实场景。

2)质量和数量:客户端数据集的质量和数量各不相同,这是真实场景中的一个常见属性。

3)长度:客户端数据的序列长度可能会有很大的不同,代表了一种新的数据异构类型。

4)偏好:不同的客户端有不同的偏好,通过指令调优数据集中不同的偏好指令和偏好校准数据集中不同的偏好响应来验证。.以上数据集多样性反映了现实数据场景的复杂性,使 FedLLM- Bench 成为 FedLLM 社区的综合性基准,目前已开源供社区使用。

6 研究展望

联邦大语言模型作为当前一个热门的发展方向,已经有很多应用场景,但该领域也存在着许多更深层次的、具有研究意义的传输优化问题。

1)面向异构设备及网络的联邦大语言模型传输优化研究

异构网络环境下的传输优化问题主要涉及如何在不同参与节点的资源条件下实现模型参数的高效传输。由于不同参与节点的算力和带宽各不相同,传统的模型聚合方法可能会导致某些节点负担过重或通信延迟过长,从而影响整体模型的训练效率和质量[87]。未来研究可以聚焦资源动态调度策略,即根据当前网络和参与节点的状态实时调整任务分配和梯度传输策略。例如,可以通过训练神经网络预测出各参与节点下一轮次的训练时间,从而动态调整模型训练任务的分配,并通过算子亲和性衡量各参与节点中算子的优先级来划分模型训练任务[88]。通过有效地平衡各设备的负载来缩短传输过程中的通信延迟,并提高模型性能和稳定性。

2)面向单模态及多模态异构数据的联邦大语言模型传输优化研究

联邦大语言模型需要处理来自多个地理位置和组织的异构数据,这些异构数据包括单模态异构数据及多模态异构数据。

在分布式环境中,对于单模态异构数据,采用多级联的模型融合能够显著提高整体模型的性能和泛化能力。可以采用分层模型聚合的方法,将各个节点训练得到的局部模型加权融合,或者利用迁移学习的技术。这种多级联的模型融合不仅可以提高模型的整体性能,还能够降低在数据安全性和隐私保护方面的风险[89]。因此,未来可以聚焦多级联模型融合的模型聚合算法,不仅能够在训练过程中实时调整参与节点的参数权重,还可以动态调整各个节点的贡献度,从而实现全局模型的最优性能。

在实际应用中,传统的单一模态学习往往难以全面捕捉到这些数据的丰富性和多样性,在分布式环境中多模态异构数据之间存在着丰富的关联性和复杂的信息交互。多模态学习将来自不同模态(如文本、图像、视频等)的信息有效地整合在一起,以增强模型对复杂数据的理解和处理能力。对于多模态学习而言,一个关键挑战是如何设计有效的信息融合策略,以最大化不同模态数据的互补性和增强模型的整体性能[90]。这包括但不限于跨模态特征的对齐与集成、模态间信息的传递与共享,以及多模态数据下的联合学习和推理机制的设计。例如,在视觉与语言理解的分布式多任务中,研究人员可以探索如何将图像中的视觉信息与文本描述中的语义信息有效结合,从而实现更精确和全面的任务解决方案。

3)面向领域自适应的联邦大语言模型传输优化研究

联邦大语言模型还有望推动领域自适应的传输优化研究[91]。具体地,设计基于领域特定数据的预训练模型,通过对目标领域参数的理解与调整,提升模型在特定领域的适应能力。领域自适应的核心在于模型如何利用目标领域的特定数据来调整参数,从而提升模型性能。这可以通过多种方式实现,包括但不限于对特定领域数据的再训练、特征选择与适应,甚至对模型结构进行微调[92]。例如,针对医疗领域的自然语言处理任务,研究人员可以使用医学文本数据对模型进行再训练,以提高在医疗实体识别或病例分类等任务上的准确性和鲁棒性。

4)基于联邦大语言模型的新应用场景

文献[93]指出,在医疗领域,可以通过联邦学习技术共享本地病例数据,例如医学影像数据或患者记录,从而形成更精确的诊断模型和个性化治疗方案。通过大语言模型的语义理解能力,医疗设备能够更好地理解医疗专业术语和患者描述,提供更精准的医疗方案,同时有效保护患者隐私数据;并且该文作者在 AMIA Annual Symposium 会议上也对临床语言模型的联邦学习框架展开研讨,深入探讨了联邦大语言模型在临床语言模型中的应用,并强调了其对推进医疗保健领域 AI 的潜在影响。

在金融领域,智能边缘计算和联邦大语言模型的结合正在推动风险评估、反欺诈和客户服务等关键领域的创新。智能边缘计算通过在接近数据源的位置执行计算和分析,减少了数据传输延迟和带宽需求,适用于实时处理金融领域[94]。文献[95]指出,联邦大语言模型利用分布式学习技术,可以在保护数据隐私的同时,对多个地区、多种金融产品的数据进行分析,生成智能决策支持系统。技术融合可以有效地处理金融领域的复杂数据挖掘和预测问题,为机构提供更准确的风险评估并增强反欺诈能力,同时优化客户服务和个性化推荐系统的效果。通过将本地数据和全局模型相结合,联邦大语言模型可以推动数据安全共享、个性化决策支持和模型训练的创新发展,助力社会各领域迈向智能化、高效率和可持续发展的新阶段。

结束语 研究联邦大语言模型中的传输优化问题,对大

语言模型的训练在实际场景中的应用具有重要的意义。本文首先基于联邦学习的概念以及大语言模型训练的算法原理，重点关注联邦大语言模型传输过程中可优化的问题，分别从减少数据传输量、模型压缩和分布式模型并行优化技术 3 个方面对联邦大语言模型优化的已有研究展开介绍，并总结了已有开源大语言模型及所用到的传输优化技术，最后展望了联邦大语言模型的发展前景。

参 考 文 献

[1] ZHAO W X, ZHOU K, LI J, et al. A survey of large language models[J]. arXiv: 2303.18223, 2023.

[2] 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.

[3] DEVLIN J, CHANG M W, LEE K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv: 1810.04805, 2018.

[4] BEBIS G, GEORGIOPOULOS M. Feed-forward neural networks[J]. IEEE Potentials, 1994, 13(4): 27-31.

[5] LUO J H, WU J. Neural network pruning with residual-connections and limited-data[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 1458-1467.

[6] NAZIR A, WANG Z. A Comprehensive Survey of ChatGPT: Advancements, Applications, Prospects, and Challenges[J]. Meta-radiology, 2023, 1(2): 100022.

[7] RADFORD A, WU J, CHILD R, et al. Language models are unsupervised multitask learners [J]. OpenAI Blog, 2019, 1(8): 9-33.

[8] BROWN T, MANN B, RYDER N, et al. Language models are few-shot learners[J]. Advances in Neural Information Processing Systems, 2020, 33: 1877-1901.

[9] ACHIAM J, ADLER S, AGARWAL S, et al. Gpt-4 technical report[J]. arXiv: 2303.08774, 2023.

[10] ERHAN D, BINGIO Y, COURVILLE A, et al. Why does unsupervised pre-training help deep learning? [C]// Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. 2010: 201-208.

[11] SHAHID O, POURIYEH S, PARIZI R M, et al. Communication efficiency in federated learning: Achievements and challenges [J]. arXiv: 2107.10996, 2021.

[12] DRIESS D, XIA F, SAJJADI M S M, et al. Palm-e: An embodied multimodal language model[J]. arXiv: 2303.03378, 2023.

[13] SUN Y, WANG S, FENG S, et al. Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation[J]. arXiv: 2107.02137, 2021.

[14] CHEN M, SHLEZINGER N, POOR H V, et al. Communication-efficient federated learning [J]. Proceedings of the National Academy of Sciences, 2021, 118(17): e2024789118.

[15] RAJBHANDARI S, RASLEY J, RUWASE O, et al. Zero: Memory optimizations toward training trillion parameter models [C]// SC20: International Conference for High Performance Computing, Networking, Storage and Analysis. IEEE, 2020: 1-16.

[16] VM K, WARRIER H, GUPTA Y. Fine Tuning LLM for Enterprise: Practical Guidelines and Recommendations [J]. arXiv: 2404.10779, 2024.

[17] CHEN C, FENG X, ZHOU J, et al. Federated large language model: A position paper[J]. arXiv: 2307.08925, 2023.

[18] WANG J, LIU Q, LIANG H, et al. A novel framework for the analysis and design of heterogeneous federated learning [J]. IEEE Transactions on Signal Processing, 2021, 69: 5234-5249.

[19] HOULSBY N, GIURGIU A, JASTRZEBSKI S, et al. Parameter-efficient transfer learning for NLP[C]// International Conference on Machine Learning. PMLR, 2019: 2790-2799.

[20] HE R, LIU L, YE H, et al. On the effectiveness of adapter-based tuning for pretrained language model adaptation [J]. arXiv: 2106.03164, 2021.

[21] LI X L, LIANG P. Prefix-tuning: Optimizing continuous prompts for generation[J]. arXiv: 2101.00190, 2021.

[22] LESTER B, AL-RFOU R, CONSTANT N. The power of scale for parameter-efficient prompt tuning [J]. arXiv: 2104.08691, 2021.

[23] LIU X, JI K, FU Y, et al. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks[C]// Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). 2022: 61-68.

[24] HU E J, SHEN Y, WALLIS P, et al. Lora: Low-rank adaptation of large language models[J]. arXiv: 2106.09685, 2021.

[25] LIN B Y, HE C, ZENG Z, et al. Fednlp: Benchmarking federated learning methods for natural language processing tasks[J]. arXiv: 2104.08815, 2021.

[26] CAI D, WU Y, WANG S, et al. FedAdapter: Efficient Federated Learning for Modern NLP[J]. arXiv: 2205.10162, 2022.

[27] BENGIO Y, LOURADOUR J, COLLOBERT R, et al. Curriculum learning[C]// Proceedings of the 26th Annual International Conference on Machine Learning. 2009: 41-48.

[28] KIM G, YOO J, KANG S. Efficient federated learning with pre-trained large language model using several adapter mechanisms [J]. Mathematics, 2023, 11(21): 4479.

[29] SUN G, MENDIETA M, YANG T, et al. Exploring parameter-efficient fine-tuning for improving communication efficiency in federated learning[J]. arXiv: 2210.01708, 2024.

[30] ZHAO H, DU W, LI F, et al. FedPrompt: Communication-Efficient and Privacy-Preserving Prompt Tuning in Federated Learning[C]// ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2023: 1-5.

[31] YANG F E, WANG C Y, WANG Y C F. Efficient model personalization in federated learning via client-specific prompt generation[C]// Proceedings of the IEEE/CVF International Conference on Computer Vision. 2023: 19159-19168.

[32] CHE T, LIU J, ZHOU Y, et al. Federated learning of large language models with parameter-efficient prompt tuning and adaptive optimization[J]. arXiv: 2310.15080, 2023.

[33] YI L, YU H, WANG G, et al. Fedlora: Model-heterogeneous

personalized federated learning with lora tuning[J]. arXiv: 2310.13283, 2023.
[34] JIANG F, DONG L, TU S, et al. Personalized wireless federated learning for large language models[J]. arXiv: 2404.13238, 2024.
[35] JIANG J, LIU X, FAN C. Low-parameter federated learning with large language models[J]. arXiv: 2307.13896, 2023.
[36] BABAKNIYA S, ELKORDY A R, EZZELDIN Y H, et al. SLoRA: Federated parameter efficient fine-tuning of language models[J]. arXiv: 2308.06522, 2023.
[37] RAJE A. Communication-Efficient LLM Training for Federated Learning[D]. Pittsburgh: Carnegie Mellon University, 2024.
[38] HUANG W, WANG Y, CHENG A, et al. A Fast, Performant, Secure Distributed Training Framework For LLM[C]// ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2024: 4800-4804.
[39] ZHUANG W, CHEN C, LYU L. When foundation model meets federated learning: Motivations, challenges, and future directions [J]. arXiv: 2306.15546, 2023.
[40] REED R. Pruning algorithms-a survey[J]. IEEE Transactions on Neural Networks, 1993, 4(5): 740-747.
[41] HAN S, POOL J, TRAN J, et al. Learning both weights and connections for efficient neural network[J]. arXiv: 1506.02626, 2015.
[42] FRANTAR E, ALISTARH D. Sparsegpt: Massive language models can be accurately pruned in one-shot[C]// International Conference on Machine Learning. PMLR, 2023: 10323-10337.
[43] LI H, KADAV A, DURDANOVIC I, et al. Pruning Filters for Efficient ConvNets[J]. arXiv. 1608.08710, 2016.
[44] JIANG Y, WANG S, VALLS V, et al. Model pruning enables efficient federated learning on edge devices[J]. IEEE Transactions on Neural Networks and Learning Systems, 2023, 34(12): 10374-10386.
[45] HUANG H, ZHANG L, SUN C, et al. Distributed pruning towards tiny neural networks in federated learning[C]// 2023 IEEE 43rd International Conference on Distributed Computing Systems (ICDCS). IEEE, 2023: 190-201.
[46] MA X, FANG G, WANG X. Llm-pruner: On the structural pruning of large language models[J]. Advances in neural information processing systems, 2023, 36: 21702-21720.
[47] FRANTAR E, ALISTARH D. Sparsegpt: Massive language models can be accurately pruned in one-shot[C]// International Conference on Machine Learning. PMLR, 2023: 10323-10337.
[48] SUN M, LIU Z, BAIR A, et al. A Simple and Effective Pruning Approach for Large Language Models[J]. arXiv: 2306.11695, 2023.
[49] HINTON G, VINYALS O, DEAN J. Distilling the knowledge in a neural network[J]. arXiv: 1503.02531, 2015.
[50] GOU J, YU B, MAYBANK S J, et al. Knowledge distillation: A survey[J]. International Journal of Computer Vision, 2021, 129(6): 1789-1819.
[51] ANIL R, PEREYRA G, PASSOS A, et al. Large scale distributed neural network training through online distillation[J]. arXiv: 1804.03235, 2018.
[52] WU C, WU F, LYU L, et al. Communication-efficient federated learning via knowledge distillation[J]. Nature Communications, 2022, 13(1): 2032.
[53] PENG Z, FAN X, CHEN Y, et al. FedPFT: Federated Proxy Fine-Tuning of Foundation Models[J]. arXiv: 2404.11536, 2024.
[54] WU F J, LI Z T, LI Y L, et al. FedBiOT: LLM Local Fine-tuning in Federated Learning without Full Model[J]. arXiv: 2406.17706, 2024.
[55] HAN S, MAO H, DALLY W J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding[J]. arXiv: 1510.00149, 2015.
[56] KIRTAS M, OIKONOMOU A, PASSALIS N, et al. Quantization-aware training for low precision photonic neural networks [J]. Neural Networks, 2022, 155: 561-573.
[57] LIU Z, OGUZ B, ZHAO C, et al. LLM-QAT: Data-Free Quantization Aware Training for Large Language Models[J]. arXiv: 2305.17888, 2023.
[58] REISIZADEH A, MOKHTARI A, HASSANI H, et al. Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization[C]// International Conference on Artificial Intelligence and Statistics. PMLR, 2020: 2021-2031.
[59] CHEN Y, CHEN Z, WU P, et al. FedOBD: Opportunistic block dropout for efficiently training large-scale neural networks through federated learning[J]. arXiv: 2208.05174, 2022.
[60] KIM J, LEE J H, KIM S, et al. Memory-efficient fine-tuning of compressed large language models via sub-4-bit integer quantization[J]. Advances in Neural Information Processing Systems, 2024, 36.
[61] DETTMERS T, PAGNONI A, HOLTZMAN A, et al. Qlora: Efficient finetuning of quantized llms[C]// Proceedings of the 37th International Conference on Neural Information Processing System. 2024: 36187-36207.
[62] DETTMERS T, LEWIS M, BELKADA Y, et al. Gpt3.int8(): 8-bit matrix multiplication for transformers at scale[J]. Advances in Neural Information Processing Systems, 2022, 35: 30318-30332.
[63] LIN J, TANG J, TANG H, et al. AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration [J]. arXiv: 2306.00978, 2023.
[64] BONDARENKO Y, NAGEL M, BLANKEVOORT T. Understanding and overcoming the challenges of efficient transformer quantization[J]. arXiv: 2109.12948, 2021.
[65] WEN Z, YIN W, ZHANG Y. Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm[J]. Mathematical Programming Computation, 2012, 4(4): 333-361.
[66] JADERBERG M, VEDALDI A, ZISSERMAN A. Speeding up convolutional neural networks with low rank expansions[J]. arXiv: 1405.3866, 2014.
[67] LEBEDEV V, GANIN Y, RAKHUBA M, et al. Speeding-up convolutional neural networks using fine-tuned cp-decomposition[J]. arXiv: 1412.6553, 2014.
[68] WU X, YAO Z, HE Y. Zeroquant-fp: A leap forward in llms

post-training w4a8 quantization using floating-point formats[J]. arXiv:2307.09782,2023.
[69] ZHANG M,SHEN C,YANG Z,et al. Pruning Meets Low-Rank Parameter-Efficient Fine-Tuning[J]. arXiv:2305.18403,2023.
[70] XU M,CAI D,WU Y,et al. Fwdllm:Efficient fedllm using forward gradient[J]. arXiv:2308.13894,2023.
[71] QIU Q,CHENG X,SAPIRO G. DCFNet:Deep neural network with decomposed convolutional filters[C]// International Conference on Machine Learning. PMLR,2018:4198-4207.
[72] NARAYANAN D,SHOEYBI M,CASPER J,et al. Efficient large-scale language model training on gpu clusters using megatron-lm[C]// Proceedings of the International Conference for High Performance Computing,Networking,Storage and Analysis. 2021:1-15.
[73] JIA Z,ZAHARIA M,AIKEN A. Beyond Data and Model Parallelism for Deep Neural Networks[J]. Proceedings of Machine Learning and Systems,2019,1:1-13.
[74] ANDROUTSOPOULOS K,CLARK D,HARMAN M,et al. State-based model slicing:A survey[J]. ACM Computing Surveys(CSUR),2013,45(4):1-36.
[75] SU N,HU C,LI B,et al. TITANIC:Towards Production Federated Learning with Large Language Models[C]// IEEE INFOCOM. 2024.
[76] SHOEYBI M,PATWARY M,PURI R,et al. Megatron-lm:Training multi-billion parameter language models using model parallelism[J]. arXiv:1909.08053,2019.
[77] ZHU J,LI S,YOU Y. Sky Computing:Accelerating Geo-distributed Computing in Federated Learning[J]. arXiv:2202.11836,2022.
[78] NAGRECHA K. Systems for parallel and distributed large-model deep learning training[J]. arXiv:2301.02691,2023.
[79] LI S,ZHAO Y,VARMA R,et al. Pytorch distributed:Experiences on accelerating data parallel training[J]. arXiv:2006.15704,2020.
[80] HUANG Y,CHENG Y,BAPNA A,et al. Gpipe:Efficient training of giant neural networks using pipeline parallelism[J]. Advances in Neural Information Processing Systems,2019,32(10):103-112.
[81] HARLAP A,NARAYANAN D,PHANISHAYEE A,et al. Pipedream:Fast and efficient pipeline parallel dnn training[J]. arXiv:1806.03377,2018.
[82] HE C,LI S,SO J,et al. Fedml:A research library and benchmark for federated machine learning[J]. arXiv:2007.13518,2020.
[83] FAN T,KANG Y,MA G,et al. FATE-LLM:A Industrial Grade Federated Learning Framework for Large Language Models[J]. arXiv:2310.10049,2023.
[84] KUANG W,QIAN B,LI Z,et al. Federatedscope-llm:A comprehensive package for fine-tuning large language models in federated learning[J]. arXiv:2309.00363,2023.
[85] YE R,WANG W,CHAI J,et al. OpenFedLLM:Training Large Language Models on Decentralized Private Data via Federated Learning[J]. arXiv:2402.06954,2024.
[86] YE R,GE R,ZHU X,et al. FedLLM-Bench:Realistic Benchmarks for Federated Learning of Large Language Models[J]. arXiv:2406.04845,2024.
[87] XIA Q,YE W,TAO Z,et al. A survey of federated learning for edge computing:Research problems and solutions[J]. High-Confidence Computing,2021,1(1):100008.
[88] ZOU W,LIU X,HOU S,et al. Affinity-Based Resource and Task Allocation in Edge Computing Systems[C]// 2023 IEEE 22nd International Conference on Trust,Security and Privacy in Computing and Communications(TrustCom). IEEE,2023.
[89] LIU Z,HUANG T,LI B,et al. Epnet++:Cascade bi-directional fusion for multi-modal 3d object detection[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence,2022,45(7):8324-8341.
[90] JI W,WEI Y,ZHENG Z,et al. Deep multimodal learning for information retrieval[C]// Proceedings of the 31st ACM International Conference on Multimedia. 2023:9739-9741.
[91] LIU F,ZHANG T,DAI W,et al. Few-shot Adaptation of Multimodal Foundation Models:A Survey[J]. arXiv:2401.01736,2024.
[92] FARAHANI A,VOGHOEI S,RASHEED K,et al. A brief review of domain adaptation[J]. arXiv:2010.03978,2021.
[93] PENG L,LUO G,ZHOU S,et al. An in-depth evaluation of federated learning on biomedical natural language processing for information extraction[J]. NPJ Digital Medicine,2024,7(1):127.
[94] CHEN X,SHI Q,YANG L,et al. ThriftyEdge:Resource-efficient edge computing for intelligent IoT applications[J]. IEEE network,2018,32(1):61-65.
[95] NGUYEN D C,DING M,PATHIRANA P N,et al. Federated learning for internet of things:A comprehensive survey[J]. IEEE Communications Surveys & Tutorials,2021,23(3):1622-1658.

DUN Jingbo, born in 2001, postgraduate. Her main research interests include federated large language model and so on.

LI Zhuo, born in 1983, Ph.D, Ph.D professor, is a senior member of CCF(No. 29832S). His main research interests include edge computing, distributed machine learning and mobile wireless networks.

(责任编辑:何杨)