计算机研究与发展
Journal of Computer Research and Development
ISSN 1000-1239,CN 11-1777/TP

《计算机研究与发展》网络首发论文

题目：联邦学习开源框架综述
作者：林伟伟，石方，曾岚，李董东，许银海，刘波
收稿日期：2022-02-15
网络首发日期：2022-08-19
引用格式：林伟伟，石方，曾岚，李董东，许银海，刘波. 联邦学习开源框架综述[J/OL]. 计算机研究与发展.
https://kns.cnki.net/kcms/detail/11.1777.TP.20220819.0826.002.html

计 算 机 研 究 与 发 展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.20220148

联邦学习开源框架综述

林伟伟[1] 石方[1] 曾岚[2] 李董东[1] 许银海[1] 刘波[3]

[1]（华南理工大学计算机科学与工程学院 广州 510006）

[2]（吉林大学数学学院 长春 130012）

[3]（华南师范大学计算机学院 广州 510631）

（linww@scut.edu.cn）

A Survey of Federated Learning Open-Source Frameworks

Lin Weiwei[1], Shi Fang[1], Zeng Lan[2], Li Dongdong[1], Xu Yinhai[1], and Liu Bo[3]

[1]（*School of Computer Science & Engineering, South China University of Technology, Guangzhou 510006*）

[2]（*School of Mathematics, Jilin University, Changchun 130012*）

[3]（*School of Computer Science, South China Normal University, Guangzhou 510631*）

Abstract In recent years, FL (federated learning) has gained widespread attention as an effective solution to breaking down the barrier to data sharing and is being progressively applied in areas such as healthcare, finance, and smart cities. FL frameworks are the cornerstone of academic research and industrial applications. Although companies such as Google, OpenMined, WeBank, and Baidu have open-sourced their own FL frameworks and systems, there is a lack of in-depth research and comparison of the technical principles, applicability scenarios, and problems of these FL open-source frameworks. For this reason, according to the preference level of each open-source framework in the industry, we select the widely used open-source frameworks to analyze. For the different types of FL frameworks, firstly, the system architecture and system function are analyzed, respectively. Secondly, we compare and analyze each framework from the aspects of privacy mechanism, machine learning algorithm, computing paradigm, learning type, training architecture, communication protocol, visualization, etc. Moreover, this paper presents two FL experiments for different application scenarios to help the readers choose and use the open-source framework to implement federated learning applications. Finally, based on the openness of the current framework, the paper discusses the possible future research directions from the aspects of privacy security, incentive mechanism, cross-framework interaction, etc. This paper aims to provide references and ideas for developing and innovating an open-source framework, architecture optimization, security improvement, and algorithm optimization.

Key words federated learning; open-source framework; model training; machine learning; big data

摘要 近年来，联邦学习作为破解数据共享壁垒的有效解决方案被广泛关注，并被逐步应用于医疗、金融和智慧城市等领域.联邦学习框架是联邦学习学术研究和工业应用的基石.虽然 Google、OpenMined、微众银行和百度等企业开源了各自的联邦学习框架和系统，然而，目前缺少对这些联邦学习开源框架的技术原理、适用场景、存在问题等的深入研究和比较.为此，根据各开源框架在业界的受众程度，选取了目前应用较广和影响较大的联邦学习开源框架进行深入研究.针对不同类型的联邦学习框架，首先分别从系统架构和系统功能 2 个层次对

收稿日期：2022-02-15；**修回日期**：2022-08-08

基金项目：广东省重点领域研发计划项目（2021B0101420002）；国家自然科学基金项目（62072187, 61872084）；广东省基础与应用基础研究重大项目(2019B030302002)；广州市开发区国际合作项目（2021GH10, 2020GH10）

This work was supported by the Key-Area Research and Development Program of Guangdong Province (2021B0101420002), the National Natural Science Foundation of China (62072187, 61872084), the Guangdong Major Project of Basic and Applied Basic Research (2019B030302002), and the Guangzhou Development Zone Science and Technology (2021GH10, 2020GH10).

通信作者：刘波（liugubin530@126.com）

各框架进行剖析；其次从隐私机制、机器学习算法、计算范式、学习类型、训练架构、通信协议、可视化等多个维度对各框架进行深入对比分析.而且，为了帮助读者更好地选择和使用开源框架实现联邦学习应用，给出了面向 2 个不同应用场景的联邦学习实验.最后，基于目前框架存在的开放性问题，从隐私安全、激励机制、跨框架交互等方面讨论了未来可能的研究发展方向，旨在为开源框架的开发创新、架构优化、安全改进以及算法优化等提供参考和思路.

关键词 联邦学习；开源框架；模型训练；机器学习；大数据
中图法分类号 TP393

人工智能（artificial intelligence, AI）从 1995 年达特茅斯会议开始经过了两起两落的发展.第 1 个高峰期自动化算法的提出使得人们看到了 AI 的希望，但是受计算能力的限制大规模任务训练无法完成，AI 进入了第 1 个低谷.第 2 个高峰期霍普菲尔特神经网络（hopfiled neural network, HNN）和反向神经网络（back propagation, BP）的提出使得大规模网络训练成为可能.但是由于算力和数据不够导致 AI 进入了第 2 个低谷.随着深度神经网络的提出、硬件设备计算能力的提升以及大数据的出现，AI 迎来了第 3 个高峰.特别是近年来智能边缘设备的激增，海量的数据更是推动了边缘协同技术的发展.许多研究人员尝试将人工智能技术和边缘计算结合起来，挖掘庞大边缘设备的巨大潜力.其中，Neurosurgeon[1]可以说是较具代表性的研究之一. 它的基本思想是将一个完整的深度神经网络分割成几个更小的部分并将它们下放到边缘设备进行训练，依靠边缘设备和用户之间的低延迟，可以显著提高模型训练速度.

虽然大数据时代提供了海量数据，但是大部分行业中的数据都是以孤岛形式存在.由于隐私安全、公司制度等问题，即使同一个公司的不同部门之间实现数据整合也非常困难.因此，在现实中想要联合各地的各个机构进行数据交流是一项艰巨的任务.由此可见，“数据孤岛”[2]问题和数据隐私安全问题成为了制约人工智能发展的重要因素.

为了解决存在的问题，联邦学习（federated learning, FL）应运而生，并被成功应用于工业界和学术界.2016 年，谷歌公司在安卓手机终端研究机器学习时提出了 FL 这一概念和技术[3-4]，旨在保护隐私安全的前提下进行各参与方的数据交流.具体来讲，即多个数据拥有者（如移动设备）可以在中央服务器（如服务提供商）协调下训练模型.且在训练过程中，各参与方的原始数据始终保留在本地，服务器主要通过加密机制下的参数交换建立共有模型.

可以看出，联邦学习技术是一种“合作共赢”的模式，在这种联邦机制下，联邦系统帮助各参与方建立了一个“共同富裕”的策略.特别是对于各商业企业，联邦学习可以实现不同行业间的数据交流，打破数据壁垒，实现各行业间的协同发展.因此，随着联邦学习研究的不断深入，各科研团队与公司纷纷推出了适用于不同场景的联邦学习框架，如 FATE[5]和 TensorFlow Federated[6]等.

据中国信息通信研究院推出的报告显示，2020 年通过评测的联邦学习产品多达 18 款，拥有联邦学习框架和产品的企业超过 60 多家.除了 Google 开源的 TensorFlow Federated[7]、OpenMined 开源的 PySyft[8]、南加州大学团队的 FedML[9]和剑桥大学团队的 Flower[10]外，业界内较为成熟的联邦学习框架还有微众银行的 FATE[5]和百度的 PaddleFL[11].根据各框架的受众定位，其主要被应用于工业产品研究和学术研究，以帮助业界的研究人员了解联邦学习的原理并进一步促进联邦学习理论、算法以及隐私安全等方面的优化和创新.由此可见，联邦学习框架是学术研究和工业应用的基石，然而，尽管联邦学习的研究和开源框架的开发进展迅速，但目前仍鲜有文献针对各框架进行系统分析和比较.因此，为了帮助大家更系统、更快速地了解联邦学习框架，本文根据各开源框架在业界的受众程度，选取具有代表性的 PySyft，FATE，TensorFlow Federated，Paddle FL，FedML，Flower 框架进行详细分析和比较.针对不同类型的研究框架，本文首先从框架的系统架构和系统功能 2 个层次出发分别对各框架进行详细分析；其次从算法、隐私机制、计算范式、学习类型、工业支持、可视化、硬件类型等多个维度对不同框架进行对比分析.同时本文基于目前框架存在的问题，进一步讨论了未来可能的研究发展方向，为开源框架的建设创新、结构优化、安全优化以及算法优化等提供有效思路.

1 联邦学习概述

1.1 联邦学习的定义及分类

联邦学习是一种加密的分布式机器学习范式，一般由多个客户端（如移动设备）和一个中央服务器（如服务提供商）组成.其特点是各参与客户端的数据始终保持在用户本地，以最大限度保障客户端数据安全.

联邦学习常用的框架包含 2 种：中心化架构和去

中心化架构.中心化架构也被称作客户端-服务器架构，在该架构中，各参与客户端利用自己的本地数据和本地资源进行本地训练，待训练完成后再将脱敏参数上传到服务器进行整合，其具体架构如图 1 所示.中心化架构的基本流程大致可以分 3 步：1）分发全局模型.中央服务器初始化全局模型，并根据不同的客户端状态信息（如是否充电、是否为计费网络等）选择参与训练的客户端，并将初始化后的模型结构和参数分发给所选客户端. 2）训练本地模型并发回更新.客户端收到模型后利用本地数据执行模型训练，在训练一定次数之后，将更新的模型参数发送给服务器.3）聚合与更新.服务器对所选客户端参数进行聚合后更新全局模型，并将更新后的模型及参数发送给各客户端，通过重复以上步骤直到停止训练.

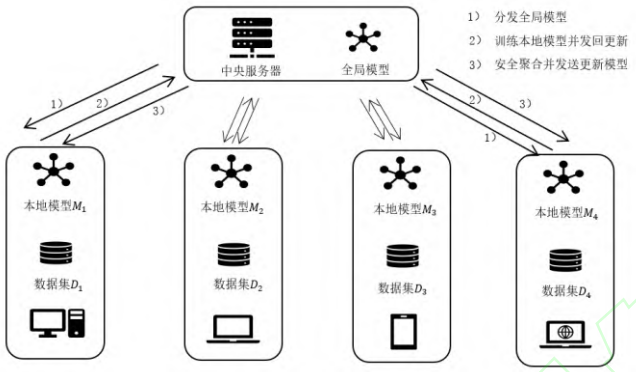


Fig. 1 Centralized architecture of federated learning system

图 1 联邦学习系统中心化架构

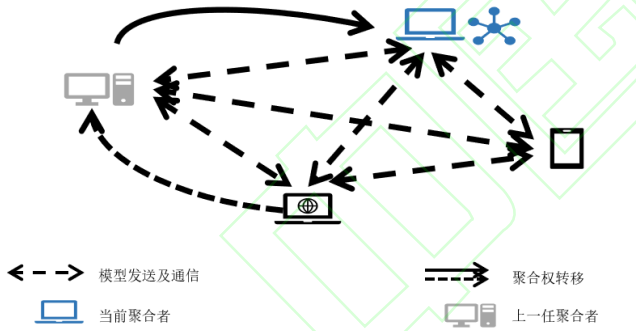


Fig. 2 Decentralized architecture of federated learning system

图 2 联邦学习系统去中心化架构

不同于中心化架构，在去中心化的联邦学习[12]架构中，由于各参与客户端可以直接通信，不需要借助第三方（服务器），因此也被称作对等网络架构，其架构示意图如图 2 所示.在该架构中，联邦学习的基本流程与去中心化架构相似，不同的地方在于训练全局模型的任务是由某一个参与方发起，且当其他参与方对模型进行训练后，各参与方需要将其本地模型加密传输给其余参与方.虽然该架构减少了第三方服务器的参与，但是由于所有模型参数的交互都是加密的，因此需要更多的操作进行加密和解密操作.目前在工业界和学术界研究更多的还是基于中心化的联邦学习架构.

此外，在联邦学习系统中，各参与方的数据又具有不同的分布特征.若根据参与方之间数据重叠程度的不同，联邦学习又可以分为横向联邦学习[13]、纵向联邦学习[14]以及迁移联邦学习[15].

如图 3 所示，横向联邦学习也称特征对齐的联邦学习，适用于各参与方之间数据特征空间重叠较多而用户空间重叠较少的情况.目前横向联邦学习经典框架是 FedAvg[3]，唤醒单词识别[16]和输入法下一词预测[17]是横向联邦的典型应用.纵向联邦学习（如图 4 所示），即样本对齐的联邦学习，适用于各参与方之间用户空间重叠较多，而特征空间重叠较少或没有重叠的场景.目前支持纵向联邦学习的经典框架包括 FATE，PaddleFL，FedML.联邦迁移学习（如图 5 所示）是对横向联邦学习和纵向联邦学习的补充.它用于克服数据或标签不足的情况，以解决单边数据规模小和标签样本少的问题，适用于各参与方用户空间和特征空间都重叠较少的场景.目前支持联邦迁移学习的框架主要为 FATE.

	ID	x_1	x_2	x_3	Y_1
数据集A	u_1				
	u_2				
	u_3				
	ID	x_1	x_2	x_3	Y_1
数据集B	u_4				
	u_5				
	u_6				

Fig. 3 Horizontal federated learning

图 3 横向联邦学习

ID	x_1	x_2	x_3	Y_1	ID	x_4	x_5	x_6	Y_2
u_1					u_1				
u_2					u_2				
u_3					u_3				

数据集A 数据集B

Fig. 4 Vertical federated learning

图 4 纵向联邦学习

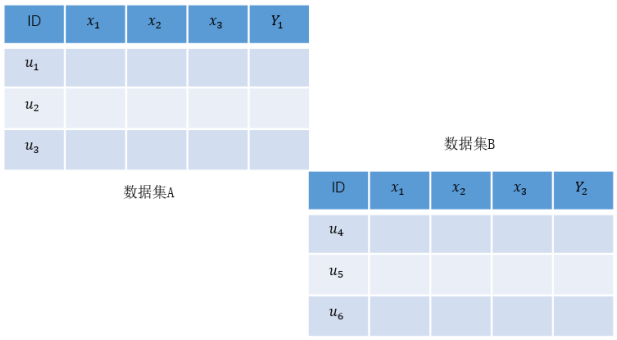


Fig. 5 Federated transfer learning

图 5 联邦迁移学习

1.2 联邦学习与传统分布式学习的区别

从系统架构上看，联邦学习与传统分布式学习都是由服务器和多个分布式节点组成，具有较高的相似性.但是相比于传统分布式学习，联邦学习在数据、通信以及系统构成上又具有自己的特点，其与传统分布式学习的主要区别如表 1 所示.

1.2.1 数据方面

联邦学习与传统分布式学习在数据方面的区别主要体现在 3 个方面：数据分布、数据量级和数据安全.1）数据分布：在分布式学习中，不同设备的数据通常是均匀、随机分布的，具有独立同分布的特点.而在联邦学习中，不同设备的数据是其独立产生的，由于设备拥有者的喜好、设备的地理位置、时间等的差异往往表现出不同的分布特征，具有非独立同分布的特点.2）数据量级：分布式学习为了提高训练效率，通常都会把训练数据均匀分布在每个参与设备上，实现负载均衡.然而在联邦学习中，每个参与设备拥有的数据量与设备拥有者的喜好以及设备自身有关，很难保证不同设备拥有相近的数据量.3）数据安全：分布式学习中的服务器对参与设备以及其中的数据具有较高的控制权，可以将训练数据分布在各个参与设备上，也可以对参与设备进行调度，让设备之间进行数据交换等操作.当数据具有隐私敏感属性时，传统分布式学习无疑会给用户数据带来极大的隐私泄露风险.而在联邦学习过程中，由于参与设备的数据始终保持在本地，服务器无法直接或间接操作设备上的数据，因此参与设备对数据具有绝对的控制权，可以最大限度地保障数据隐私和安全.

1.2.2 通信方面

联邦学习与传统分布式学习在通信方面的区别主要体现在 2 个方面：网络稳定性和通信代价.1）网络稳定性：传统分布式场景中的服务器与参与设备通常都位于专用的机房中，且用高速宽带进行互联，网络、运行环境都相对稳定.而参与联邦学习的设备大多数是手机、平板等移动设备，由于具有移动性，其所处的网络环境并不稳定，导致其稳定性较差，随时都可能与中心服务器断开连接.2）通信代价：由于传统分布式场景中的服务器与参与设备通常处于同一地理位置，且具有专用的信道进行通信，其通信代价往往较小.而在联邦学习中，由于参与训练的设备可能分布在不同的地理位置，与服务器一般处于远程连接的状态，受不同设备网络带宽的影响，还存在设备掉线等不稳定情况，因此联邦学习相比于传统的分布式学习通信代价要更高.

1.2.3 系统构成

联邦学习与传统分布式学习的系统构成较为相似，都是由服务器和多个分布式节点组成.不同的地方在于，在传统的分布式学习系统中，数据分布、计算以及模型更新都是由服务器进行统一控制，服务器具有绝对的控制权.而在联邦学习系统中，由于节点之间数据分布、数据量级、计算能力以及网络环境之间的差别，联邦学习的系统不但需要考虑数据安全性和非独立同分布的特点，还需要考虑数据传输时延等众多因素.

1.3 联邦学习计算范例

在联邦学习中，根据其目前的应用场景，我们将其计算范例分为单机模拟、基于拓扑结构的分布式训练和移动设备端训练[9].

单机模拟主要是为了帮助研究人员快速了解联邦学习框架以及测试算法等，比较适合小型研究使用.当项目内容较为简单时，可以将项目部署在一台服务器上，由该服务器模拟整个联邦学习框架.然而，单机对于大规模数据计算和存储的处理能力有限，当需要进行复杂联邦学习训练时，单机的硬件资源将无法满足需求.

基于拓扑结构的分布式训练类似于传统的分布式训练，由多方协同进行联邦计算，可以用于大型实验测试和真实环境部署（如不同机构组织之间）.在该计算范例中，中央服务器作为协调者负责分发和聚合模型，所有模型的训练都在客户端完成.因此在分布式训练范例中，通信起着至关重要的作用.在分布式通信中，应用层可以采用的协议有 HTTP 和 WebSocket 等，会话层则可以基于 RPC 进行通信.在通信过程中，联邦学习框架通常采用同态加密、差分隐私和安全多方计算等手段保护隐私安全，以求达到效率、精度和隐私的平衡.

联邦学习的重要计算范例之一是移动设备端训练，如移动电话、智能手环等设备.在学习过程中节点间需要频繁通信，非常消耗计算资源和传输资源，然而移动设备通常计算能力有限，且由于网络状况的不稳

定可能随时退出训练.因此，除了 PySyft，FedML，Flower 外，其他联邦学习框架并未过多关注移动设备训练，而更多考虑在服务器上训练好模型后,将存储的模型移植到终端，在终端推理模型.

Table 1 The Difference Between Traditional Distributed Learning and Federated Learning

表 1 传统分布式学习与联邦学习的区别

对比项目		学习类型	
		分布式学习	联邦学习
数据	数据分布	独立同分布	非独立同分布
	数据量级	相同	不同
	数据安全	数据控制权在服务器手中，具有较高的隐私泄露风险	数据控制权在参与设备手中，可以最大限度地保障数据隐私和安全
通信	网络稳定性	较强	较弱
	通信代价	较小	高
系统构成		数据分布、模型训练以及模型更新都是服务器进行统一控制，不需要考虑传输时延等因素.	模型训练和模型更新分离，需要考虑设备间异构性所带来的影响以及传输时延等众多因素.

2 联邦学习开源框架

为了更系统、更快速地了解联邦学习框架以及不同框架特点，本文根据开源框架在业界的受众程度，选取目前在工业界和学术界影响较大的开源框架进行深入分析和介绍，重点从各框架的架构设计和系统功能 2 个层次对各框架进行剖析.

2.1 PySyft

PySyft[8]是 OpenMined 在 2018 年提出，开源于 2020 年的一个基于 python 的隐私保护深度学习框架，主要借助差分隐私和加密计算等技术，对联邦学习过程中的数据和模型进行分离.PySyft 的设计主要依赖于客户端之间交换的张量链，特点是涵盖了多种隐私机制，如差分隐私、同态加密和安全多方计算；并以可扩展的方式进行设计，便于研究人员可以添加新的联邦学习方法或隐私保护机制.

2.1.1 PySyft 系统架构

由于 PySyft 的设计主要依赖于客户端之间交换的张量链，因此其系统架构的重点是基于张量的链抽象模型的设计.如图 6 所示，基于张量的链抽象模型的核心部分是一个称为_Syft 张量的抽象，主要用于表示数据的状态或变换，并且可以链接在一起. 链结构的头部始终有 PyTorch 张量，并使用子属性向下访问由_Syft 张量体现的变换或状态，使用父属性向上访问由_Syft 张量体现的变换或状态。

_Syft 张量有 2 个重要的子类.第 1 个是在实例化 Torch 张量时自动创建的 Local 张量，其作用是在 Torch 张量上执行与加载运算相对应的本机运算.第 2 个是当将张量发送给远程客户端时创建的 Pointer 张量.Pointer 张量发送和取回张量十分简便：当接收到命令时，整个链将被发送给客户端，并被替换为具有双节点的链（张量（空）和指定拥有数据和远程存储地址的 Pointer 张量）.此外，Pysyft 采用了类似指针的方式进行多方调度，当向客户端发送张量时，会返回一个指向该张量的指针，所有操作都将使用该指针执行.

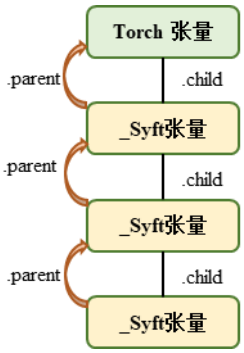


Fig. 6 Chain abstract model

图 6 链抽象模型

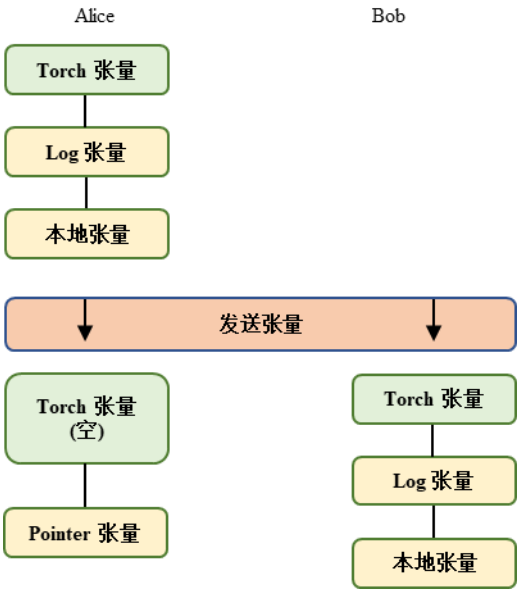


Fig. 7 Tensor sending schematic

图 7 张量发送示意图

PySyft 还建立了一个用于客户端间通信的标准化协议.在联邦学习环境中的客户端有 2 种实现方式：Network sockets 和 Web Sockets.Network sockets 客户端通过调用 Socket API 来完成应用层协议，实现不同客户端之间的通信.而 Web sockets 客户端可以从浏览器中实例化，每个客户端都被视为设备上的一个单独实体，并通过 WebSocket API 进行通信，为不同机器上的远程客户端之间的联邦学习提供了解决方案.此外，由于 WebSocket API 是纯事件驱动，因此可以使用异步事件在客户端监听连接生命周期的每个阶段.

2.1.2 PySyft 系统功能

作为注重隐私安全的深度学习框架，PySyft 重要的一项系统功能就是基于张量指针集成了 SyMPC 多方安全计算库以实现 SPDZ 协议.同时，除安全多方计算外，PySyft 还支持差分隐私，包括 DP-SGD，PATE，Moments Accountant，Laplace 和指数机制.同态加密方面由 TenSEAL 库负责完成，其主要依赖 Microsoft SEAL 中的 CKKS，允许各方加密它们的数据，以便让不受信任的第三方使用加密数据训练模型，而不泄露数据本身.除此之外，还有 PyDP，Petlib 等库提供了隐私保护.

对于联邦学习类型，PySyft 目前仅可用于横向联邦学习，涵盖联邦算法包括 FedAvg 等.虽然它可进行基于拆分神经网络的垂直学习，并利用 PSI 协议以保护数据集隐私，但仍未提供纵向联邦的解决方案.机器学习算法方面，该框架支持逻辑回归和神经网络，如 DCGAN 和 VAE 模型.除联邦学习的基本方法外，PySyft 还支持联邦的同步和异步机制.操作系统方面，PySyft 支持 Mac，Windows，Linux.研究人员能进行单机模拟、基于拓扑架构的分布式训练和移动端设备训练.

2.1.3 PySyft 版本变化

虽然 OpenMined 提供了多种隐私保护方式，但目前的版本仅支持横向联邦学习. 其中，2021 年 7 月发布的 PySyft 0. 2. 8 版本取消了以模型和数据为中心的概念，引入了动态和静态的联邦概念. 2021 年 9 月发布的 PySyft 0. 2. 9 版本改进了函数加密共享的执行速度，添加了对 BFV 方案的支持，以及监控基准. 直到 2021 年 12 月发布的最新版 PySyft 0. 6 暂时仅对功能进行一定程度上的维护. 相较于 FATE 和 PaddleFL，PySyft 尚未提供高效的部署方案及服务器 ing 端解决方案.

2.2 FATE

FATE（federated AI technology enabler）是微众银行在 2019 年开源的联邦学习框架，旨在解决各种工业应用实际问题.在安全机制方面，FATE 采用密钥共享、散列[18]以及同态加密技术，以此支持多方安全模式下不同种类的机器学习、深度学习和迁移学习.在技术方面，FATE 同时覆盖了横向、纵向、迁移联邦学习和同步、异步模型融合，不仅实现了许多常见联邦机器学习算法（如 LR[19-20]，GBDT，CNN[21]），还提供了一站式联邦模型服务解决方案，包括联邦特征工程、模型评估、在线推理、样本安全匹配等.此外，FATE 所提供的 FATE-Board 建模具有可视化功能，建模过程交互体验感强，具有较强的易用性.目前这一开源框架已在金融、服务、科技、医疗等多领域推动应用落地.为了让大家更清晰的了解 FATE，在接下来的内容中，我们将从系统架构以及系统功能 2 个层次出发，对 FATE 进行详细分析.

2.2.1 FATE 系统架构

FATE 主要包括离线训练和在线预测 2 部分，其系统架构如图 8 所示.其中，FATE Flow 为学习任务流水线管理模块，负责联邦学习的作业调度；Federation 为联邦网络中数据通信模块，用于在不同功能单元之间传输消息；Proxy 作为网络通信模块承担路由功能；元服务为集群元数据服务模块；Mysql 为元服务和 FATE-Flow 的基础组件，用于存放系统数据和工作日志；FATE 服务（FATE serving）为在线联合预测模块，提供联邦在线推理功能；FATE-Board 为联邦学习过程可视化模块；Egg 和 Roll 分别为分布式计算处理器管理模块和运算结果汇聚模块，负责计算和存储数据.

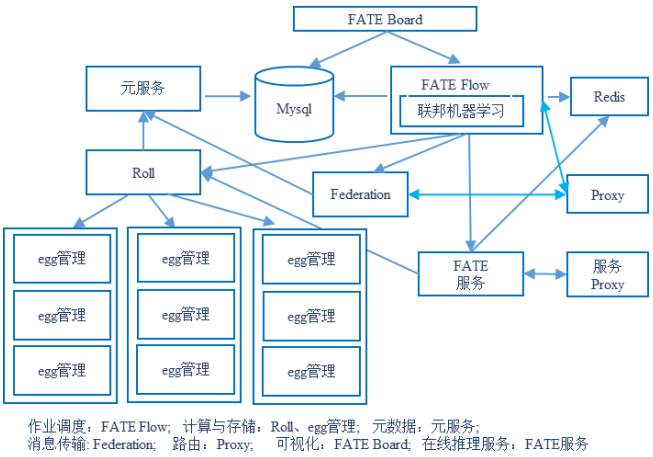


Fig. 8 FATE system architecture

图 8 FATE 系统架构图

2.2.2 FATE 系统功能

1） 离线训练框架.

对于离线训练框架（如图 9 所示），其架构主要分成基础设施层、计算存储层、核心组件层、任务执行层、任务调度层、可视化面板层以及跨网络交互层.在基础设施层，FATE 提供 KubeFATE 模式，使用云本地技术管理联邦学习工作负载，支持通过 Docker Compose 和 Kubernetes 进行部署.其中，KubeFATE 提供了丰富的 FATE 集群生命周期管理功能，可以方便地对集群进行增加或删减、修改配置等操作.利用 FATE 集群管理的任务框架，研发人员可以轻松定位与解决基础设施层的问题.

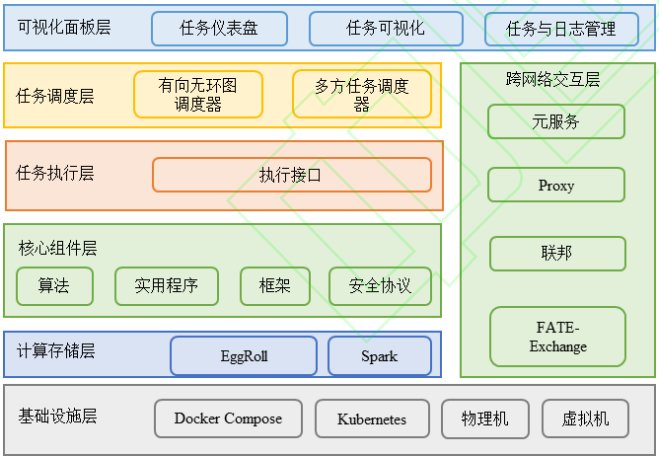


Fig. 9 Off-line training framework architecture

图 9 离线训练框架架构图

在计算存储层，离线训练框架使用 EggRoll 以及 Spark 作为分布式计算引擎.当使用 EggRoll 作为计算引擎时，簇管理器（cluster manager）负责提供服务入口以分配资源，节点管理器（node manager）执行实际计算和进行存储，Rollsite 负责数据传输.当使用 Spark 作为计算引擎时，需要使用 HDFS 来实现数据的持久化，而 pipeline 的同步和训练过程中消息的同步则要依赖于 Nginx 和 RabbitMQ 服务来完成.

核心组件层主要提供数据交互、算法和模型训练评估相关的实现.本层主要由 FederatedML 组成，其包括许多常见机器学习算法的联邦实现以及必要的实用工具.简单来说，可分为以下几个功能模块：①算法模块，用于数据预处理和联邦特征工程的机器学习算法；②实用程序模块，作为启用联邦学习的工具，例如加密算子，参数定义及传递变量自动生成器等；③框架模块，用于开发新算法模块的工具包和基础模型；④安全协议模块，包括 SPDZ，OT 等协议，以实现安全联邦学习.

任务执行层以及调度层主要由 FATE-Flow 构成.FATE-Flow 是实现联邦学习建模和任务协同调度的重要工具，主要包括：DAG 定义联邦学习 Pipeline、联邦任务协同调度、联邦任务生命周期管理、联邦模型管理、联邦任务输入输出实时追踪以及生产发布功能.

可视化面板层由 FATE-Board 构成，主要实现联邦建模过程的可视化.FATE-Board 由任务仪表盘、任务可视化、任务管理与日志管理等模块组成，可以对联邦学习过程中模型的训练状态及输出结果进行可视化展现.FATE-Board 提供了矩阵图、回归结果、树模型等的可视化反映，从而研究人员可以更及时地了解模型状态与调整参数，提升联邦学习的效果.

跨网络交互层由 Federated-Network 联邦多方通信网络构成，它的架构如下：元服务为元数据管理者和持有者，负责定位不同数据在不同机器的位置；Proxy 为应用程序层联邦学习路由，Federation 负责全局对象的抽象和实现，FATE-Exchange 提供通信功能.

2）在线预测框架功能.

FATE 服务是针对联邦学习模型的高性能工业化服务系统.在离线建模后，FATE-Flow 将模型推送至 FATE 服务，FATE 服务通过加载训练模型实现在线预测功能，主要支持动态加载联合学习模型、多级缓存、生产部署的预/后处理、联邦学习在线批量预测、各方并行预测等.其具体部署架构如图 10 所示.其中服务器服务（serving-service）为预测功能的核心，用于处理各种请求，提供基于 gRPC 的模型在线推理服务.服务器服务从 FATE-Flow 加载好模型后，将该模型相关的服务信息注入注册中心（zookeeper），以便网关服务从中拉取可用服务并调用.同时，训练好的模型信息将被发送给模型管理服务完成持久化存储，作为备份可在特殊情况下恢复.

服务代理（serving-proxy）主要是在多方交互时作为网关服务实现服务路由，实现客户端与主机之间的通信.它对外提供 gRPC 接口以及 HTTP 接口，维护一个各参与方 partId 的路由表，通过路由表中的信息

转发外部系统的请求.

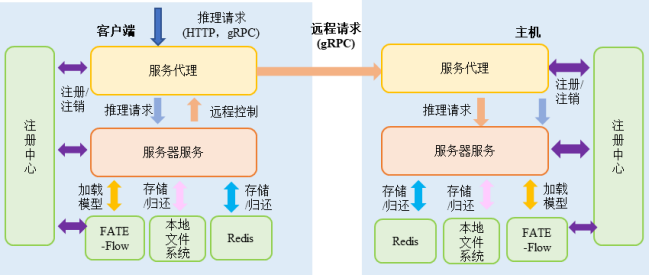


Fig. 10 FATE-serving deployment architecture

图 10 FATE-serving 部署架构

从 FATE 的系统架构和系统功能可以看出，FATE 的优势在于其具有丰富的算法组件，具有简单、开箱即用、易用性强的特点.作为目前唯一的一个可以同时支持横向联邦学习、纵向联邦学习以及联邦迁移学习的开源框架，FATE 得到了业界广泛的关注与应用.同时，FATE 还提供了一站式联邦模型解决方案，可以有效降低开发成本，相比于其他开源框架，在工业领域优势突出.

2.2.3 FATE 版本变化

2019 年首款工业级联邦学习框架 FATE 由微众银行推出，具有区别于其他开源框架的特色功能，包括可视化模块 FATE-Board，联邦学习建模 pipeline 调度和生命周期管理工具 FATE-Flow. 截至 2019 年 12 月发布的 FATE v1. 2 版本，覆盖横向联邦学习、纵向联邦学习、联邦迁移学习，得到了社区内广泛的关注与应用. 2020 年 10 月发布的 FATE 1. 5 版本对纵向联邦的通信效率进行了提升，支持不同的计算、存储和传输引擎的组合，并对用户的便捷使用做出改进，引入了 Pipeline 和 CLI v2 等. 2021 年 3 月，框架新增了本地文件系统目录路径虚拟存储引擎和消息队列 Pulsar 跨站传输引擎，对组网模式进一步扩充，并在后续加入了 1 对多的 FATE 服务的集群推送. 在 FATE 1. 7 版本中，开始支持 EggRoll、Spark-Local 计算引擎、Hive 存储，进一步提升 Spark-Local 和 Spark-Cluster 之间的异步融合. 2022 年 4 月发布的 FATE 1. 8 除了添加对加密性能评估 Paillier、性能评估 SPDZ 和管道 dsl 的转换工具外，还对纵向联邦的性能进行了提升，例如 SecureBoost 节省带宽 75%，提速 1. 5～5 倍.

2.3 TFF (TensorFlow federated)

2019 年，谷歌发布了基于 TensorFlow 构建的全球首个大规模移动设备端联邦学习系统[6]，该系统用于在移动智能设备执行机器学习和其他分布式计算，旨在促进联邦学习的开放性研究和实验.

2.3.1 TFF 系统架构

为协调客户端和中央服务器的交互，TFF 除了提供与 GKE（Google Kubernetes engine）和 Kubernetes 集群的集成，还提供容器映像来部署客户端并通过 gRPC 调用进行连接，其系统架构图 11 所示:

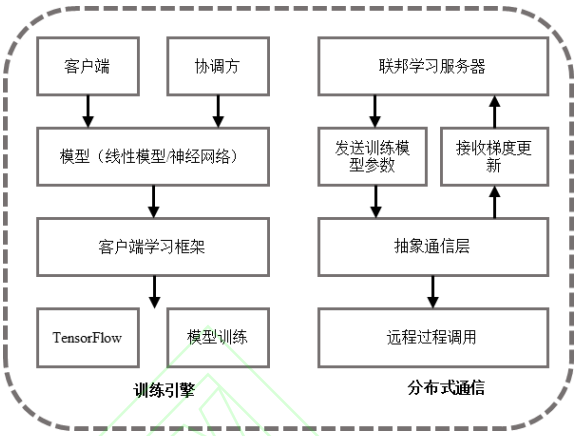


Fig. 11 TFF framework architecture

图 11 TFF 系统架构图

从系统架构图可以看出，TFF 的训练流程包括以下几步：1）服务器从所有设备端筛选出参与该轮联邦学习任务的设备，为了不影响用户体验，筛选标准包括是否充电、是否为计费网络等因素.2）服务器向训练设备发送数据，包括计算图以及执行计算图的方法.而在每轮训练开始时，服务器向设备端发送当前模型的超参数以及必要状态数据.设备端根据全局参数、状态数据以及本地数据集进行训练，并将更新后的本地模型发送到服务端.3）服务端聚合所有设备的本地模型，更新全局模型并开始下一轮训练.由此可见，设备端的功能主要包括连接服务器，获取模型和参数状态数据，模型训练，模型更新.TFF 的客户端架构设计如图 12 所示：

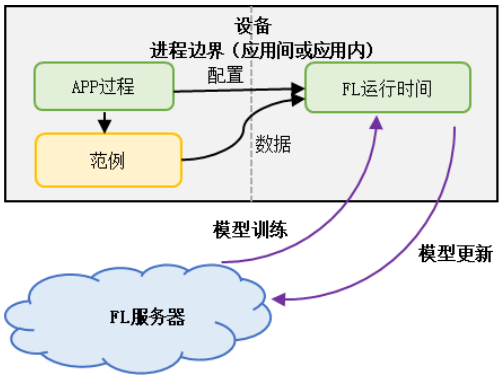


Fig. 12 TFF client architecture

图 12 TFF 客户端架构图

对于服务器端，TFF 围绕编程模型参与者模式（actor model）设计，使用消息传递作为唯一的通信机制，采用自顶向下的设计结构（如图 13 所示）.其中协调方（coordinator）是顶级参与者，负责全局同步和推送训练.多个协调方与多个联邦学习设备集群

一一对应，负责注册设备集群的地址. 协调方接收有关选择器的信息，并根据计划指示它们接受多少设备参与训练.而选择器负责接收和转发设备连接，同时定期从协调方接收有关联邦集群的信息，决定是否接受每台设备做出本地决策.主聚合器（master aggregator）负责管理每个联邦学习任务的回合数，它可以根据设备的数量做出动态决策，以生成聚合器（aggregator）实现弹性计算.

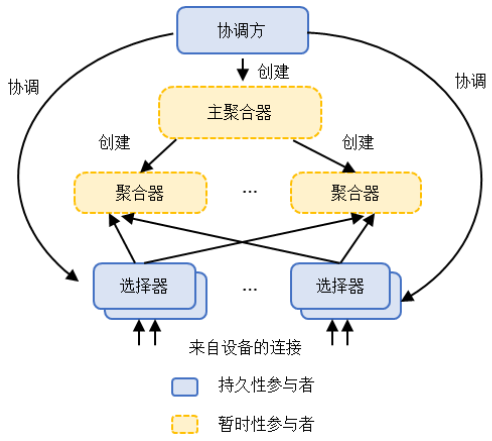


Fig. 13 TFF top-down structure

图 13 TFF 自顶向下结构图

2.3.2 TFF 系统功能

为实现联邦学习模型训练的实验环境和计算框架，TFF 构建了 FL API（federated learning API）和 FC API（federated core API）2 个级别的接口.

如图 14 所示，FL API 包括模型、联合计算构建器、数据集 3 个部分.模型部分提供封装完成的 tff.learning 函数，研究人员可以直接调用该函数实现各种联邦学习算法而无需自行构建，如可以使用 FedAvg 和 Fed-SGD 进行模型训练.联邦计算构建器的主要目的是使用现有模型为训练或评估构造联邦计算的帮助函数，主要用于辅助联邦学习的训练和计算过程.在数据集模块，通过 Tensorflow API 中提供的 LEAF[22]生成联邦学习特定训练数据集，给出了用于 TFF 仿真和模型训练的可直接下载和访问的罐装数据集.除了高级接口外，FC API 提供了底层联邦学习接口，它是联邦学习流程的基础.研究人员可以通过它方便地构建自定义联邦学习算法.

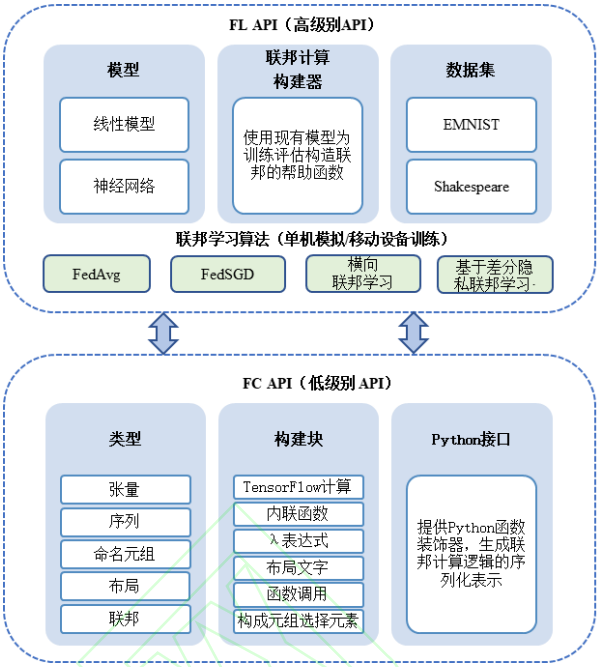


Fig. 14 TFF API architecture

图 14 TFF API 架构图

在联邦学习类型方面，TFF 目前只支持横向联邦学习，尚未提供纵向联邦及迁移学习的方案；模型方面，提供了 FedAvg，Fed-SGD 等算法，同时也支持神经网络和线性模型；在计算范式方面，TFF 支持单机模拟和移动设备训练，不支持基于拓扑结构的分布式训练；在隐私保护机制方面，TFF 采用差分隐私以保证数据安全.TFF 的主要受众目标是研究人员和从业者，他们可以采用灵活可扩展的语言来表达分布式数据流算法，定义自己的运算符，以实现联邦学习算法和研究联邦学习机制.

2.3.3 TFF 版本变化

谷歌于 2019 年 2 月首次发布 TFF 框架，12 月发布 TFF 0.11.与 PySyft 相同，该框架也仅支持横向联邦学习.用户可以通过 FL API 与 TensorFlow/Keras 交互，完成分类、回归等任务，也可以基于 FC API 构建自定义的联邦学习算法.其中，2020 年发布的 TFF 0.17 版本，支持计算跟踪控制，消除了客户机数量对分布式运行的影响.2022 年 TFF 0.20 版本除了对 API 功能进行了升级外，明确了对零客户端聚合的支持，增加了张量的流量.直至最新的 TFF 0.23 版本，该框架仍然仅提供了基于差分隐私的隐私保护，未添加同态加密和多方安全计算.

2.4 Paddle FL

2019 年，百度基于安全多方计算、差分隐私等领域的实践，开源了 PaddlePaddle 生态中的联邦学习框架 PaddleFL[11]，旨在为业界提供完整的安全机器学习开发生态.PaddleFL 提供多种联邦学习策略，因此该框架在不同领域都受到了广泛关注.

2.4.1 Paddle FL 系统架构

Paddle FL 架构的整体设计可以参考图 15.如图所示，Paddle FL 可以支持横向联邦和纵向联邦 2 种策略，对于横向联邦学习，其主要支持 FedAvg，DPSGD，SECAGG 等策略；对于纵向联邦学习，其主要支持 LR with PrivC 和 NN with MPC [23]的神经网络.PaddleFL 底层的编程模型采用的是飞桨训练框架，结合飞桨的参数服务器功能，其可以实现在 Kubernetes 集群中联邦学习系统的部署.训练策略方面，Paddle FL 可进行多任务学习[24]、迁移学习、主动学习等训练.

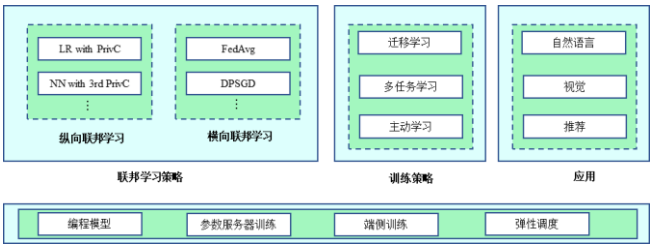


Fig. 15 Paddle FL architecture

图 15 Paddle FL 的架构设计

2.4.2 Paddle FL 系统功能

Paddle FL 主要提供 2 种实现联邦学习的方案：数据并行（data parallel, DP）和基于多方安全计算的联邦学习（paddle federated learning with MPC, PFM）.在 Data Parallel 中，模型训练过程主要分为 2 个阶段：编译（complie time）和分布式运行（distributed run-time）.

如图 16 所示，编译包含 4 个组件：联邦学习策略（FL-strategy）、用户自定义程序（user-defined srogram）、分布式配置（distributed-config）、联邦学习任务生成器（FL-job-generator）. 联邦学习策略主要提供横向和纵向联邦学习算法，如神经网络和逻辑回归等，用户也可以自己定义机器学习模型和训练策略. 用户自定义程序定义机器学习模型结构和训练策略，分布式配置的作用是设定分布式训练的节点信息. 联邦学习任务生成器负责为联邦服务器端和客户端生成联邦任务，并将它发送到联邦参数服务器进行联邦训练.分布式运行由服务器、客户端、联邦学习调度器 3 部分组成.服务器为联邦参数服务器，是联邦学习的管理者.客户端为联邦学习的各参与方，与联邦参数服务器进行通信，其中通信模式包括 Gloo 和 gRPC. 联邦学习调度器在训练过程中负责调度，选择每一训练回合中参与的客户端.

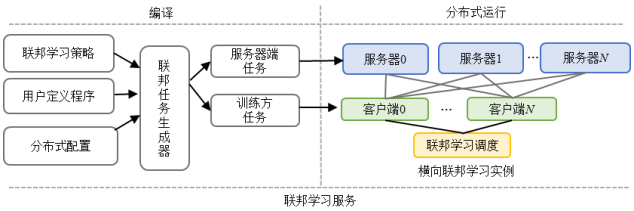


Fig. 16 Data parallel training framework

图 16 Data parallel 训练框架图

PFM 整个训练过程包括数据准备、训练推理、结果解析 3 个阶段.其中数据准备阶段主要包括 2 项工作：1）隐私集合求交.在保护数据隐私的前提下，求出各参与方的数据交集；2）数据加密及分发.采用 Secret Sharing 进行数据和模型加密，并分发给参与方. 数据准备完成后，根据用户设定的 FL 模型进行训练，同时基于多方安全计算保证训练安全.在训练结束后，对结果解密并发送给各参与方.PFM 提供了多方安全计算下的联邦学习，支持横向和纵向联邦学习.在 PFM 中，一共有 3 个参与者：输入方（IP）、计算方（CP）和结果方（RP）.输入方（数据或模型的拥有者）负责将数据或模型加密后发送给计算方；计算方（云上的虚拟机）接受加密后的数据或模型后基于特定的 MPC 协议进行训练或任务推理，完成后将结果发送给结果方；结果方接受加密计算结果后重构明文结果.三方参与者角色可以重叠.

Paddle FL 提供的联邦学习策略涵盖了横向联邦学习和纵向联邦学习，相比于其他的联邦学习框架，其适合在大规模分布式集群中部署.其中，横向联邦学习主要涵盖了 FedAvg[3]，DP-SGD[25]等算法，纵向联邦方面主要包括了基于 PrivC 的逻辑回归和基于 ABY3 协议的神经网络.目前 Paddle FL 开源了横向联邦学习场景，可以用于具有相同类型任务的多个组织进行联合训练.对于纵向联邦学习场景，百度指出将会在未来开源纵向联邦学习编程框架，并在实现不同联邦学习类型的编程接口统一.同时，在性能方面也将提升训练的速度和精度、跨地域的稀疏通信、通信的稳定性等.

2.4.3 Paddle FL 版本变化

自 2019 年 11 月 PaddleFL 开源开始，百度随后在 2020 年 5 月发布的 Paddle 1.8 版本深度优化了命令式编程模式功能. 对原生推理库性能进行了显著优化，推出了可以轻量化部署的推理引擎 PaddleLite，并发布了前端推理引擎 Paddle. js.且 Paddle-服务器ving 全面升级，提供强大简单化的部署功能，对应的开发库和工具组件也进一步进行了丰富完善. 在 2021 年 2 月推出的 Paddle 2. 0 版本，百度针对之前的编程

范式，对支持动态图转静态图的方式进行了模型部署和训练加速，同时对支持的组件性能也进行了大范围提升. 由于开源时间较短，其算子丰富程度逊于上述框架，但它通过与百度机器学习开源框架 PadddlePaddle 的交互，也收获了不少研究人员的关注.

2.5 FedML

FedML 是由美国南加州大学联合 MIT、Stanford、MSU、UW-Madison、UIUC、腾讯、微众银行等众多高校与公司联合发布的一个联邦学习开源框架.FedML 不但支持 3 种计算范例（单机模拟、基于拓扑结构的分布式训练和移动设备训练），还通过灵活且通用的 API 设计和参考基准实现促进了各种算法研究，并针对非独立同分布（non-independent identically distributed, Non-IID）数据设置了精选且全面的基准数据集用于公平比较.

2.5.1 FedML 系统架构

FedML 主要包含 FedML-API 和 FedML-core 这 2 个组件，分别对应高级别接口（high-level API）和低级别接口（low-level API），系统架构如图 17 所示：

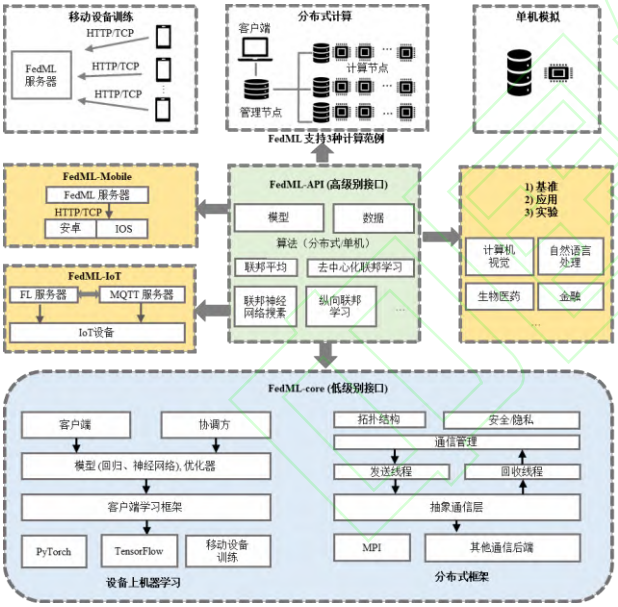


Fig. 17 FedML system architecture

图 17 FedML 系统架构图

其中 FedML-core 将分布式通信和模型训练分为 2 个单独的模块.分布式通信模块负责不同客户端之间的底层通信，使用统一的通信管理来完成算法通信协议.目前，FedML-core 支持 MPI，RPC，MQTT 通信后端.其中 MPI 主要用于满足单个集群中的分布式训练需求；RPC 主要用于满足跨数据中心的通信需求（例如 cross-silo FL）；MQTT 主要用于满足移动设备的联邦学习训练.由于 FedML 默认采用 MPI 通信，因此下面简要介绍其通信过程，针对于移动设备的 MQTT 通信则在“FedML-Mobile 和 FedML-IoT”部分介绍.

在 FedML 中，MPI 通信主要由通信管理和其维护的发送线程和接收线程实现.其中发送线程和接收线程各自维护一个缓冲队列，发送线程每隔 0.003s 轮询一次自己的队列，如果有新消息放入，就将其发送.对于收到的消息，通信管理每隔 0.3s1 次对其自身进行轮询，有新消息收到则通知观察者，观察者利用回调机制处理信息.

具体通信过程如图 18 所示：①服务器启动，发送初始化信息给客户端；②客户端收到服务器端发送的消息，触发 handler 函数；③训练方进行本地模型的训练；④每轮训练结束后，将训练好的参数放入发送队列；⑤发送线程将队列中的数据传回服务器；⑥服务器收到客户端端发送的消息，触发 handler 函数；⑦进行全局模型参数的更新；⑧将更新后的全局参数传入发送队列进行下发，开始下一轮迭代训练.此外，FedML 还支持用户自定义通信协议.如果需要使用不同的通信协议，用户只需替换底层的通信管理即可.

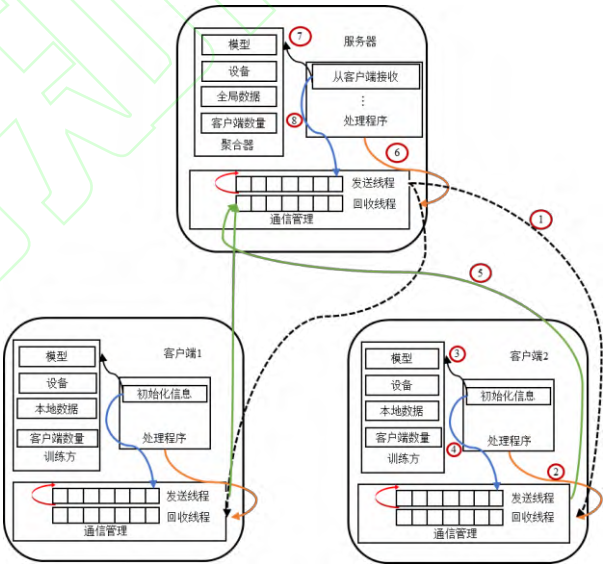


Fig. 18 MPI communication diagram

图 18 MPI 通信示意图

在分布式通信模块内部，TopologyManager 支持多种网络结构，如基于数据中心的 FL、去中心化 FL、分层 FL[26-27]等.

FedML-API 建立在 FedML-core 之上，包括模型，数据集和算法.FedML 支持的算法包括线性模型（逻辑回归）、神经网络（CNN，RNN）等；每个算法包括服务器 Manager 和客户端 Manager 这 2 个对象，用于集成 FedML-core 的通信模块的 ComManager 和机器学习组件的客户端与 coordinator，完成分布式算法协议（如 FedAvg，FedNAS，FedNova，FedOpt 等）和分布式训练.FedML-API 采用模型、数据集和算法

相互分开的系统设计，以实现代码重用和公平比较，避免由于不同实现方式导致的算法之间的统计或系统级差距.此外，通过这种设计，研究人员无需了解不同的分布式算法的细节，就可以开发新模型并提交更现实的数据集.

2.5.2 FedML 系统功能

从 FedML 架构图可以看出，FedML 的系统功能主要包含 FedML-API，FedML-core，FedML-Mobile，FedML-IoT.因此，接下来我们将分别介绍这几部分的功能.

1) FedML-core 功能.

自定义客户端及信息交换：FedML-core 采取了面向客户端的编程设计模式，当设计 FL 算法中的训练过程时，用户可以通过继承客户端 Manager 类并使用其预定义的 API 来定义接收和发送消息，从而在 FL 网络中自定义其客户端.另外，消息交换方面，FedML 还支持梯度或模型之外的消息交换（如特定的中间结果），每个客户端都可以从发送的角度定义消息类型.客户端 Manager 用于处理各客户端定义的消息，并发送其自身定义的消息.

拓扑管理：FL 具有各种拓扑定义，例如垂直 FL[28]、拆分学习[29]、去中心化 FL 和分层 FL[27].为了满足这种多样化的要求，FedML 提供了 TopologyManager 来管理拓扑，并允许客户端在培训期间将消息发送给任意相邻客户端.在完成 TopologyManager 的初始设置后，网络中的每个客户端都可以通过 TopologyManager 查询相邻客户端的 ID、权重等信息.

隐私安全和鲁棒性：除了保护数据隐私外，联邦学习框架（尤其是移动设备训练时）的另一个关键要求是应对用户退出的鲁棒性.为了增强安全性和隐私性，FedML-core 实现了常见的密码原语，可支持多方安全计算（秘密共享）和同态加密（密钥协议和数字签名）.针对联邦学习中对抗攻击的防御，FedML 涵盖了规范差异裁剪[30]、弱差分隐私[30]、RFA[31]、KRUM 和 MULTIKRUM[32].同时，为了帮助科研人员研究新的攻击，该框架还提供了通用的对抗性攻击 API，该 API 通过实施模型替换攻击[33]和边缘案例后门攻击[34]来支持后门.

2) FedML-API 功能.

算法：FedML 提供了网络拓扑结构不同、交换信息不同（如交换模型参数或特定中间结果）的各种联邦学习算法.不同计算范式支持的算法如下：单机模拟支持 FedAvg、FedOpt、FedNova、FedNAS、去中心化联邦、纵向联邦、分层联邦；分布式计算支持去中心化联邦、FedAvg、FedRobust、FedNAS[35]、FedSEG、FedGKT、纵向联邦、Split Learning；移动设备/IoT 仅支持 FedAvg.同时，这些算法也可以用作实现示例和基准，以帮助用户开发和评估自己的算法.FedML 还在不断添加新的 FL 算法，如 Adaptive Federated optimizer，FedProx，FedMA.

模型和数据集：由于不同实验中模型和数据集的用法不一致，因此很难公平比较 FL 算法的性能.为了加强公平比较，FedML 采取了模型和数据集合二为一的方法，明确规定了数据集和模型的组合.FedML 提供的模型和数据集具体可以分为以下 3 类：线性模型（凸优化）、轻型浅层神经网络（非凸优化，一般用于 cross-device 设置）以及深神经网络（非凸优化，用于 cross-silo 训练大型 DNN）.其中线性模型的数据集包括 MNIST[36]，联邦 EMNIST[37]和 Synthetic（α，β）[38]；浅层神经网络的数据集包括联邦 EMNIST[22]、CIFAR-100[39]、莎士比亚[40]和 StackOverflow[41]；深度神经网络的数据集有 CIFAR-10，CIFAR-100，CINIC-10，StackOverflow.具体模型和数据集的组合如表 2 所示：

Table 2 Combination of FedML Specific Models and Datasets

表 2 FedML 具体模型和数据集的组合

模型	学习类型			
	FL (FedAvg, FedOpt, FedNova, 等) 或去中心化 FL	**FedNAS**	**纵向 FL**	**分割学习**
Cross-device	**CV:** Federated EMNIST + CNN，CIFAR100 + ResNet18 (Group Normalization) **NLP:** shakespeare/stackoverflow (NWP) + RNN (bi-LSTM)		lending_club_loan+ VFL, NUS_WIDE + VFL	
Cross-silo	**CV:** CIFAR10, CIFAR100,	**CV:**CIFAR10,CIF		**CV:** CIFAR10,

	CINIC10 + ResNet/MobileNet	AR100,CINIC10+ ResNet/MobileNet	CIFAR100, CINIC10 + ResNet/MobileNet
Linear	MNIST/Synthetic+ Logistic Regression		

3) FedML-Mobile 和 FedML-IoT.

FedML 的一项主要功能是其在实际硬件框架上对联邦学习的支持.具体来说，FedML 包括 FedML-Mobile 和 FedML-IoT，这是 2 个基于实际硬件框架构建设备上的 FL 测试框架.当前，FedML-Mobile（包括 FedML-服务器和 Android 客户端 Simulator 这 2 个 API）支持在 Android/iOS 智能手机进行设备上培训.而 FedML-IoT 支持 Raspberry PI 4 和 NVIDIA Jetson Nano.借助建立在实际硬件框架上的测试框架，研究人员可以评估实际系统性能，例如训练时间，通信和计算成本.FedML 的架构设计可以将分布式计算代码平稳地移植到 FedML-Mobile 和 FedML-IoT 框架上，从而几乎重用了分布式计算范式中的所有算法.

根据训练设备的异构性，FedML 采用了当前最流行的物联网协议 MQTT，实现 FedML-服务器与移动/IoT 设备之间的通信.在发布/订阅体系结构中，FedML-服务器为订阅者，识别设备何时准备好开始培训以及接收模型；而参与训练的客户端为发布者.图 19 为服务器和设备之间的工作流程.在步骤 1～3 中，服务器和设备与代理建立连接，然后由代理进行通信.同时，服务器为自己订阅一个特定的主题，以便从设备端接收状态更新.步骤 4 和步骤 5 将训练模型发送到设备.在这个步骤之前，服务器分配并准备所有对通信阶段有用的参数（资源分配、模型序列化等）.在步骤 6 和步骤 7 中，参与训练的设备开始训练并返回模型更新.

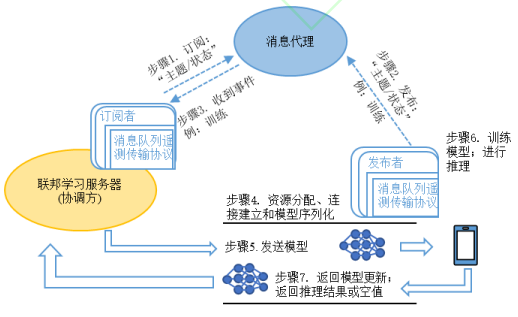


Fig. 19 Workflow between FedML servers and devices

图 19 FedML 服务器与设备之间的工作流程

2.5.3 FedML 版本变化

2020 年 9 月由南加州大学联合多所科研院所联合发布了 FedML 联邦学习开源框架，针对现有软件框架不能充分支持多样化算法开发以及实验对比存在不一致的问题，提供了一个开放的研究库以及基准. 2020 年 10 月 7 日版本提供了面向研究的数据集和模型. 2020 年 10 月 28 日版本对单一的 FedAvg 聚合算法进行扩充，加入了更多的联合优化算法. 2020 年 11 月 5 日的版本针对物联网设备的联邦学习模式进行支持. 直至 2022 年 5 月发布了最新的版本，本次升级对原有的 MPI 训练进行了扩充，加入了 NCCL 的模式. 支持跨组织的跨仓联合训练. 相较于 FATE，Paddle FL 等工业联邦学习框架，FedML 更适合开发人员作为学术研究使用.

2.6 Flower

Flower[10]是由英国牛津大学在 2020 年发布的一款联邦学习框架，其优点在于 Flower 可以模拟真实场景下的大规模联邦训练.且基于其跨平台的兼容性、跨设计语言的易用性、对已有机器学习框架的支持以及抽象的框架封装，用户可以快速高效搭建所需的联邦学习训练流程.Flower 综合计算资源、内存空间和通信资源等因素，高效实现了移动和无线客户端下异构资源的使用.

2.6.1 Flower 系统架构

Flower 包含 Flower 客户端（Flower client）、Flower 服务端（Flower service）、联邦策略（Federation strategy）、Flower 协议（Flower protocol）、Flower 数据集（Flower datasets）、Flower 基准（Flower baselines）、Flower 工具（Flower tools）这 7 部分，系统架构如图 20 所示：

Flower 客户端允许用户借助接口实现相应的操作，如 SDK 主要为用户处理连接管理、Flower 协议和序列化等操作.在目前的版本中，SDK 提供了对 Android 和 Python 的支持，研究团队指出将陆续开放对 iOS 端等移动终端的支持.

Flower 服务端在整个框架中负责节点之间的连接、客户端生命周期的管理、联邦学习执行的定制、验证、聚合和度量等处理.联邦策略中，用户可以借助抽象的策略来实现新的联邦学习算法，如 FedAvg，FedProx，QFedAvg，FedOpt.

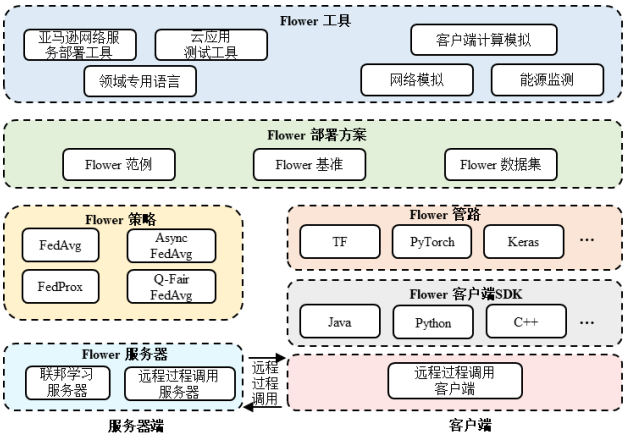


Fig. 20 Flower system architecture

图 20 Flower 系统架构图

此外，Flower 协议降低了对联邦学习设备的要求和对特定编程语言的依赖程度，提高了使用上的客制化程度.Flower 数据集中考虑到不同实验数据的使用以及数据的划分，其提供了一组内置的数据集和相应的分区函数，满足客户机之间数据的动态分配.Flower 基准提供了端到端的联邦学习实现，用户借助特定领域语言（domain-specific language，DSL）,设置好相应的实验条件就可创建.Flower 工具提供了一套部署、模拟、检测参数的系统级工具.主要包括 1）部署实例去模拟客户端和服务端；2）模拟不同性能的客户端设备；3）改变客户端和服务端之间的网络带宽，模拟真实场景下的网络状态.

Flower 的通信模块位于 Flower 服务端组件中，该组件可大致分为 4 层（如图 21 所示）.目前 Flower 仅支持 gRPC 的通信方式，但是也支持用户自定义通信协议的替换.gRPC 的交互过程主要通过连接管理（connection management）和 gRPC 网桥（gRPC bridge）这 2 个模块实现.

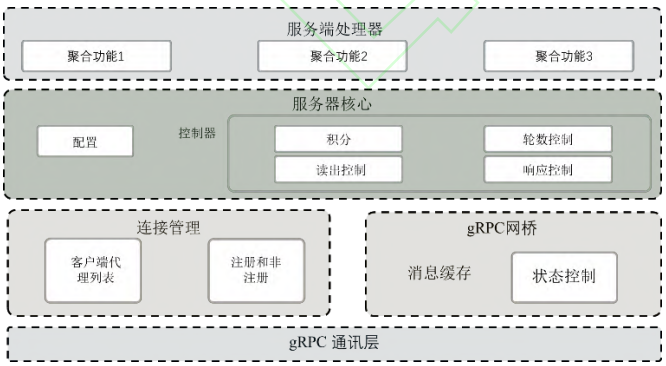


Fig. 21 Flower server architecture

图 21 Flower 服务端结构图

其中连接管理模块负责维护当前的所有 gRPC 连接.当 gRPC 服务器第 1 次收到请求时，会触发注册函数进行客户端连接的注册管理，将该客户端的信息存放到一个数组中，每一个 gRPC 对应一个客户端.服务器借助这个模块来获得指定客户端的 gRPC 连接，之后进行通信，获取模型信息.在完成通信或者因为等待超时等情况导致 gRPC 断开，则调用非注册函数将断开连接的客户端从当前的记录中删去.

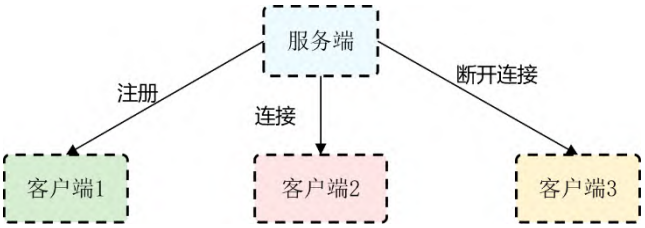


Fig. 22 Connection management diagram

图 22 连接管理示意图

客户端和服务端的通信由 gRPC 网桥进行，该模块负责缓存二者的 gRPC 信息.首先由客户端向其中放入信息，客户端再从中获取；客户端本地训练完模型后，将模型信息上传至网桥，服务端从网桥获取客户端所上传的本地模型信息进行全局模型的更新.该过程通过状态转换的方法来保证其中储存的都是相同的信息.大致的状态转换如图 23 所示：

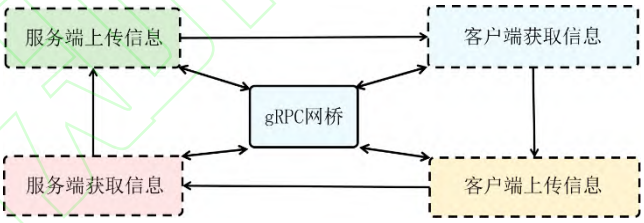


Fig. 23 gRPC Bridge state transition diagram

图 23 gRPC 网桥状态转换图

2.6.2 Flower 系统功能模块

Flower 主要提供可复现实验，机器学习算法，网络扰动，跨平台接入，大规模接入这 5 个的功能模块.

1)可复现实验模块.

一个完整的联邦学习框架需要多个组件来实现，Flower 中提供了一套可靠、成熟的组件来实现这些部分，研究人员可以在一套组件下快速进行实验验证.此外，已有算法库可以让研究人员快速的与现有方案进行对比.

2)机器学习算法模块.

联邦计算的方式弥补了传统单机模式下的不足，但是仍旧有许多的 ML 算法尚未迁移过来.Flower 通过对现有 ML 框架的链接，允许用户在现有 ML 代码库的基础之上将 ML 算法快速应用于联邦模型的训练中.

3)网络扰动模块.

通信网络贯穿联邦学习的整个始末，在现实场景中，网络通信状态直接会影响到模型训练的效率和结果.Flower 中提供了对网络带宽约束的功能，方便量

化网络波动对整个联邦学习过程的影响.

4)跨平台接入模块.

现有联邦学习框架对于异构设备的支持十分有限，更多的是倾向于服务端的定义和客户端的计算，或是倾向于对服务器集群而忽视对移动客户端的关注.Flower 中提供了对不同架构的支持，因此可以测试异构环境下不同算法的表现.此外，在设计上抽象类的设计，使得 Flower 对于特定编程语言的依赖降到最低.

5) 大规模接入模块.

在真实场景下的联邦学习训练需要大量的设备进行参与.然而，在实验场景中往往不会以真实的参与规模进行设计，对于大规模设备参与的可扩展性有待考察.Flower 在设计之初就考虑到了大量并发连接客户机的场景，具有很好的可扩展性.

2.6.3 Flower 版本变化

相比于其他几款联邦学习框架，Flower 具有可扩展性、兼容性、易扩展等特性，使得使该框架不仅可以用于项目研究，还能方便地进行生产部署. 除了 2020 年 11 月发布的首个版本，2021 年 1 月 Flower 0. 13. 0 版本实现了由集中式训练向联合式训练的转变. 2021 年 9 月的 Flower 0. 17 版本引入了虚拟客户端引擎，支持在单台机器或计算集群中实现大规模客户端的模拟. 2022 年 2 月 Flower0. 18. 0 版本实现了对 Android 移动端的支持.

3 其他联邦学习框架

除了目前业内较常用的几款开源框架外，其他的公司也根据自己的业务场景设计、开源了其框架.其中开源框架包括 Fedlearner，FedNLP，FederatedScope；闭源框架有 ClaraFL 和蜂巢.由于闭源框架的封闭性，后续的对比仅对开源框架进行.

3.1 其他开源框架

3.1.1 Fedlearner 框架

字节跳动在 2020 年初开源了联邦学习框架 Fedlearner，该框架可以支持各类联邦学习模式，包括模型管理、训练任务管理等模块[42].与微众银行等开源框架不同，Fedlearner 实行产品化工作（如图 24 所示），将模块部署于平台侧和广告主侧，注重于在推荐行业开展联邦学习.

Fedlearner 的主要架构如图 24 所示，主要流程可以概括为 3 个部分：数据求交、模型训练和部署.

1）数据求交.

在进行联邦学习前，首先要对训练双方数据求交集，找到共有的数据特征，数据求交的方法有以下 2 种：流式数据求交和 PSI 数据求交.①流式数据求交.流式数据通常是指由共同在线流量产生的数据，它们的数据落盘时间、样本存储可靠性都不能做到一致，且不同的训练方还存在样本缺失和样本顺序不统一的问题.因此，Fedlearner 针对这些数据采取了流式数据求交的方法.②PSI 数据求交.对于各方独自持有的数据，例如不同机构的用户信息数据，Fedlearner 提供了隐私保护集合交集（private set intersection，PSI）加密数据求交的方式，通过该方式完成数据求交后，双方不会得到除交集信息之外的其他任何信息.

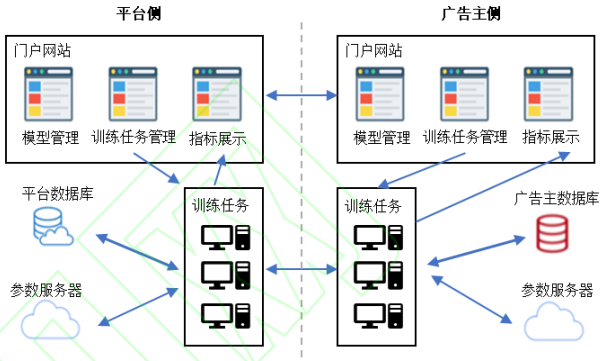


Fig.24 Fedlearner

图 24 Fedlearner

2）模型训练.

由于字节跳动在推荐场景有丰富的技术资源，因此模型训练部分包括神经网络模型训练（支持横向和纵向联邦学习）、树模型训练（SecureBoost 算法）.

3）部署.

为了实现一键部署，Fedlearner 的设计团队推出了 Kubernetes+HDFS/MySQL/Elasticsearch 的部署模式，其中 Kubernetes 管理集群和任务，HDFS 负责数据存放，MySQL 负责存储系统数据.由此，用户可以利用 Helm Chart 轻松完成大规模部署.

3.1.2 FedNLP 框架

此外，除了工业界，南加大 Lin 等人[43]也开源了首个以研究为导向的自然语言处理联邦学习框架（federated learning natural language processing, FedNLP）.其具体框架如图 25 所示，主要由应用程序层、算法层和基础架构层 3 层组成.

1）应用程序层.

应用程序层定义了数据管理、模型定义和 NLP 训练器 3 个功能模块.①数据管理.在数据管理中，4 种不同类型的数据管理负责控制从加载数据到返回训练函数的整个工作流程.用户可以根据需要开展的联邦学习任务，实现自定义数据管理.②模型定义.FedNLP 提供了 Transformer 和 LSTM 模型.其特点是与 HuggingFace Transformers 库兼容，研究人员可以直接应用现有 NLP 生态中各种类型的 Transformer，

无需重新设计.此外，Fedlearner 也支持 LSTM 模型，以实现一些特定的联邦学习案例.③NLP 训练器（单进程角度）：该 NLP 训练器不需要研究人员了解分布式系统的内容即可完成设定，即用户只需完成单进程代码编写.为实现联邦训练，用户需要继承应用层的训练方类来实现以下操作：获取本地模型参数并传输至服务器、获取服务器聚合后模型并更新本地模型参数.

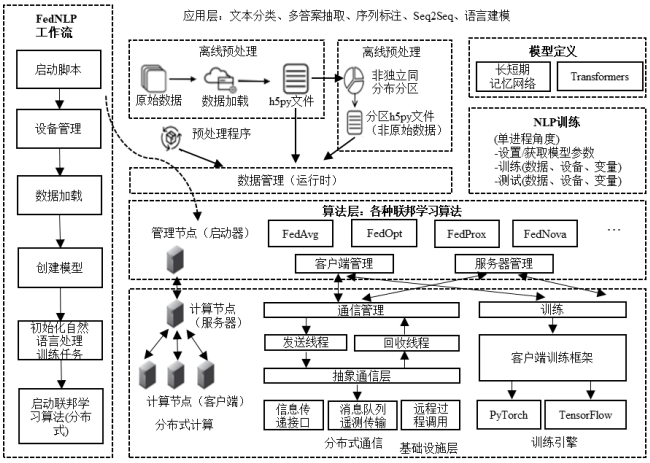


Fig.25 FedNLP architecture

图 25 FedNLP 框架图

2）算法层.

算法层由服务器管理和客户端管理核心对象组成，其作用是集成基础结构层的通信模块，以完成分布式算法协议（如 FedAvg，FedProx，FedOPT 等）和分布式训练.同时，用户也可以通过将自定义的训练方传递给算法 API 来自定义训练方，API 的参数包括模型、数据和单进程训练器.

3）基础架构层.

该层包括分布式计算、分布式传输、训练引擎 3 个模块.其中分布式计算模块主要负责管理各联邦学习参与端，进行 GPU 资源分配.训练传输模块中使用统一抽象的通信管理来完成复杂的算法通信协议.当前，该架构支持 MPI，RPC，MQTT 通信后端.其中 MPI 主要用于满足单个集群中的分布式训练需求.RPC 主要用于满足跨数据中心的通信需求（例如，跨孤岛联邦学习）.MQTT 主要用于满足智能手机或物联网设备的通信需求.训练引擎模块的主要作用是通过训练方类重用现有的深度学习训练引擎.虽然该模块的当前版本是基于 PyTorch，但它可以轻松支持 TensorFlow 等框架.同时，研究团队也指出未来该框架可能会考虑在此级别上支持通过编译器技术优化的轻量级边缘训练引擎.

3.1.3 FederatedScope

FederatedScope 是由阿里巴巴达摩院研发、开源的框架[44].该框架采用事件驱动的编程范式，用于支持现实场景中联邦学习应用的异步训练.并借鉴分布式机器学习的相关研究成果，集成了异步训练策略来提升训练效率.具体而言，FederatedScope 将联邦学习看成是参与方之间收发消息的过程，通过定义消息类型以及处理消息的行为来描述联邦学习过程.

如图 26 所示，FederatedScope 通信模块由消息和通信器组成，使用类型区分不同类型的消息，包括发送方、接收方、数据载体等.

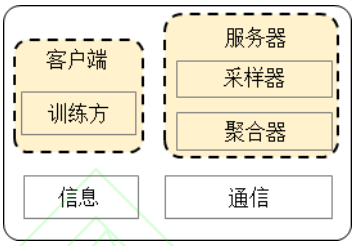


Fig.26 FederatedScope framework

图 26 FederatedScope 架构

FederatedScope 的优点在于不需要开发者将联邦学习的过程用顺序执行的视角来完整描述，而只需采用事件驱动的方式增加新的消息类型和消息处理行为，系统协助完成自动调参和高效异步训练，降低了所需的开发量以及复杂度.

而后该团队开发了 FS-G (FederatedScope-GNN)，是基于 FederatedScope 面向于图神经网络的联邦学习（GFL）的联邦学习框架[45].FS-G 是一个提供现成的图神经网络的联邦学习的框架，开发了一个 FGL 包构建可配置、统一、全面的基准.其系统架构如图 27 所示：

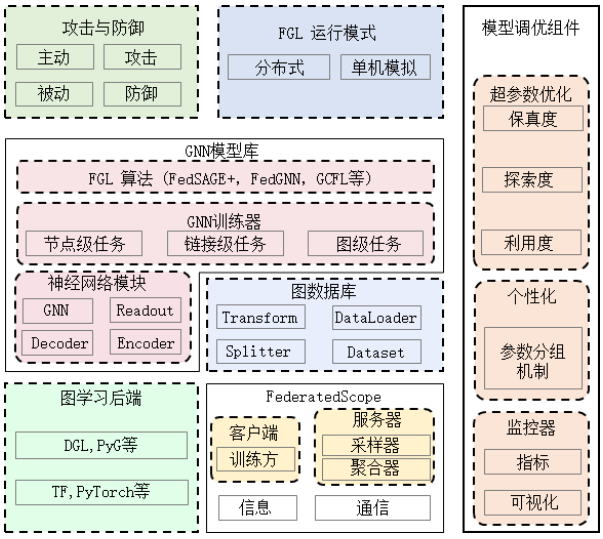


Fig.27 FS-G framework

图 27 FS-G 架构图

FS-G 系统组件功能如下:

1）GNN 模型库模块用于处理不同级别任务；神经网络模块构建不同结构的 GNN；GNN 训练方封装本地训练方法接口；图数据库模块允许用户通过配置

数据集、分割器、转化器、以及数据下载器来构建 FL 数据集.

2）图学习后端模块可对接包括 TensorFlow，Pytorch，JAX 在内的多种主流机器学习框架，以及 SQL 前端；

3）FGL 运行模式模块为 FS-G 的运行调用接口，可接收配置参数信息并返回训练结果指标集合.

4）模型调优组件模块为模型调整组件.超参数优化模块用于自动调整参数，通过减半算法（SHA）加快超参数评估和搜索，降低资源消耗.监控器模块通过可视化损失函数以及准确率等指标监控训练过程.个性化模块允许用户在 Non-iid 的条件下调整模型参数或者实现 GNN 的超参数个性化.

5）攻击与防御模块主要继承了现成的各种主动和被动隐私攻击方式，包括成员推理攻击、属性推理攻击和标签推荐攻击等.隐私安全保护策略，包括差分隐私、多方安全计算.

3.2 闭源框架

英伟达也推出了一款主要应用目标为医院和医疗机构的联邦学习框架 ClaraFL[46].该框架的特点是客户端可以部署于面向边缘的英伟达服务器上，在本地进行模型训练，并通过联邦学习的方式实现数据交流，从而与多参与方共同训练出更精准的全局模型.ClaraFL 可以将患者数据保存在医院内部，实现隐私保护的同时，帮助医生进行高速而准确的诊断.由于单个医疗机构的数据量有限，基于 ClaraFL 进行联邦学习可以有效汇总海量医疗数据，打破数据壁垒，提高医疗救治水平.

此外，平安科技也推出了一款主要应用于物流行业的联邦智能框架“蜂巢”.该框架的特点是为支持国密级加密的企业框架，采用了国密 SM2、国密 SM4 以及差分隐私和同态加密等不同的加密方式，以满足不同场景所需的不同保密级别.

3.3 框架对比分析

在 3.1 节介绍的开源框架中，框架在隐私泄露风险和数据加密都有各自的方案. 其中 FedNLP 和 FederatedScope 共同支持差分隐私的加密技术.在此基础之上，FederatedScope 除了支持多方安全加密和同态加密之外，还提供了主流的隐私评估算法.Fedlearner 则是采用嵌入式的保护框架.例如采用 PSI 加密数据或是采用 Paillier 算法对梯度加密.在算法级别，三者均支持传统的机器学习算法、深度学习网络，但是在学习类型中 FederatedScope 和 FedNLP 都支持横向联邦和纵向联邦学习，而 Fedlearner 暂时仅支持横向联邦学习.在计算范式方面，三者均支持单机模拟和分布式训练，其中 FederatedScope 为单机模拟和分布式部署提供了统一的算法描述和接口.除了单机和分布式 2 种模式外，FedNLP 还支持移动设备端的训练.针对可视化系统的支持，Fedlearner 自身有可视化图表功能，FefNLP 借助 wandb 完成对框架的监控.

4 框架对比分析

为了更好地根据应用场景以及应用需求选择相应的开源框架，本节从各框架支持的隐私机制、机器学习算法、计算范式、联邦学习类型、训练架构以及可视化等几方面对目前应用较广的几个开源框架进行了深入分析和对比，具体如表 3 所示：

降低隐私泄露风险、提升私密数据安全性是联邦学习的初衷之一，因此各框架均采用多样的加密技术（如同态加密、多方安全计算和差分隐私）以保障参与方的隐私安全.其中同态加密主要是利用具有同态性质的加密函数对数据加密，实现对加密后的数据处理和保证隐私安全.Deffie-Hellman 算法的通信双方通过交换信息生成共同密钥，利用密钥进行对称加密通信.RSA 加密算法则是非对称加密技术，由 1 对公钥和私钥组成密钥.

多方安全计算是指在无可信第三方情况下，通过多方共同参与安全完成协同计算.SPDZ 是许多联邦学习框架采用的协议，它包含混淆电路、秘密共享和不经意传输等技术.ABY3 协议则综合采用算术分享、布尔分享和混淆电路协议.

基于差分隐私的数据保护，是通过向数据或模型参数注入随机噪声以实现隐私保证，同时防止对模型的推理攻击.目前的联邦学习框架主要采用 DP-SGD 算法，对随机梯度下降算法进行改进而使其具有差分私有性

PySyft 具有相对完善的隐私保护机制，同时应用差分隐私、基于 CKKS 的同态加密和基于 SPDZ 协议的安全多方计算来保障数据安全；而 TFF 和 Flower 仅采用了差分隐私技术.与 PySyft 类似，FedML 同样提供了较为全面的隐私保障，同时它还能抵御对抗攻击，采取 RFA 和 KRUM 等技术让联邦学习具有鲁棒性.Paddle FL 利用差分隐私和基于 PrivC 和 ABY3 的安全多方计算来实现安全联邦学习.FATE 在隐私机制上主要采用同态加密技术和安全多方计算，其中前者包括 Paillier，RSA，Affine，IterativeAffine 等加密算法，后者基于 SPDZ、混淆电路、不经意传输等密码学协议实现.

Table 3 Framework Comparison

表 3 框架对比分析

对比项目		框架					
		FATE	PySyft	TFF	PaddleFL	FedML	Flower
编程语言		Python	Python	Python	Python/C++	Python	Python/Java/C++
隐私机制	同态加密	DHKE Paillier，RSA	CKKS	不支持	不支持	RSA	不支持
	多方安全计算	SPDZ，OT， Feldman VSS	SPDZ	不支持	ABY3 PrivC	Secret sharing	不支持
	差分隐私	不支持	DP-SGD PATE	DP-SGD	DP-SGD	DP-SGD	DP-SGD
机器学习算法		逻辑回归 集成学习 深度学习 迁移学习等	逻辑回归 深度学习等	逻辑回归 深度学习等	逻辑回归 深度学习等	逻辑回归 深度学习等	逻辑回归 深度学习
拓扑自定义		不支持	支持	不支持	不支持	支持	不支持
计算范式		单机模拟 基于拓扑结构的分布式训练	单机模拟 基于拓扑结构的分布式训练 移动设备端训练	单机模拟 基于拓扑结构的分布式训练	单机模拟 基于拓扑结构的分布式训练	单机模拟 基于拓扑结构的分布式训练 移动设备端训练	单机模式 基于拓扑结构的分布式训练 移动设备端训练
训练架构	中心化	支持	支持	支持	支持	支持	支持
	去中心化	不支持	不支持	不支持	不支持	支持	不支持
联邦学习类型		横向联邦 纵向联邦 联邦迁移	横向联邦	横向联邦	横向联邦 纵向联邦	横向联邦 纵向联邦	横向联邦
通信后端		gRPC	自定义	gRPC	gRPC	gRPC、MQTT MPI、自定义	gRPC 自定义
可视化		支持	不支持	不支持	不支持	支持	不支持
受众定位		工业/学术	学术	学术	工业/学术	学术	学术
跨系统		Linux/Mac	Windows/Linux/ Mac	Windows/Linux /Mac	Windows/Linux /Mac	Linux/Mac/Andr oid/iOS	Windows/Linux/ Mac/Android/iOS
网络拥塞模拟		不支持	不支持	不支持	不支持	不支持	支持
聚合算法		Fed-SMPC FedAVG 等	Fed-MPC Fed-DP Fed-HE 等	FedAvg Fed-SGD 等	FedAvg DPSGD SECAGG PFM LR With Privc NN With MPC 等	FedAvg FedOpt FedGKT FedNAS FedNova 等	FedAvg FedProx QFedAvg FedOptfed 等
场景	**异步聚合**	支持	不支持	不支持	不支持	不支持	不支持
	用户掉线	不支持	不支持	不支持	不支持	不支持	不支持

在算法级别，FATE 框架最为全面，对于横向、纵向联邦和联邦迁移学习均支持许多机器学习算法.该框架集成了各种线性模型和 DNN，RNN，CNN 等神经网络.此外，除了 FATE 提供了 SecureBoost 安全树外，其他框架均尚未支持决策树相关算法.PaddleFL 在横向联邦方面，提供了 DP-SGD 等算法；Flower 在横向联邦方面，除了对常规神经网络有着很好的支

持外，对 Sklearn 中提供的算法有着很好的支持.对于纵向联邦，支持关于神经网络和逻辑回归的相关算法.TFF，PySyft，FedML 可以实现线性模型和神经网络算法.总而言之，FATE 和 PaddleFL 倾向于提供现成算法供用户直接使用，而 PySyft，TFF，FedML 更则侧重用户构建自定义联邦学习算法.

计算范式方面，针对科学研究、测试开发和工业生产等不同使用场景，需要有不同的计算范式，因此是否拥有多样化的计算范式是衡量联邦学习框架的一个重要因素.目前较多学者主要采用支持 3 种范式的 PySyft，FedML，Flower 开展学术研究；此外，Flower 相较于其他框架，提供了一套用于模拟真实场景下网络拥塞和大规模并发的机制，使得场景更贴近现实场景.FATE 作为工业级联邦学习框架，可进行单机模拟和分布式计算，虽然在工业应用上较有优势，但暂未支持移动设备端训练.PaddleFL 目前主要支持单机模拟与基于拓扑结构的分布式训练，但研究团队指出在下一版本中将开源手机端的联邦学习模拟器.

联邦学习中服务器与客户端间存在大量的通信，需要保证通信的效率和可靠性.多数开源框架都利用 Google 提供的 gRPC 来实现 RPC 通信，包括 TFF，FATE，Paddle FL，Flower.RPC 框架可以有效连接跨数据中心的设备，适用于相互之间数据传输频繁的 cross-silo 联邦学习场景.FedML 支持的通信后端最为丰富，包括 MPI，gRPC，MQTT，这是 FedML 框架的一个突出优势.此外，Flower 和 FedML 的通信模块采用了灵活的设计，可以通过替换底层的通信管理模块来使用自定义的通信协议.

对于工业生产应用，利用 FATE 可以轻松构建端到端的联邦学习 Pipeline ，包括生产服务、建模训练、模型管理、生产发布和在线推理等方面.PySyft 尚未提供大规模或工业部署方案，更适合作为学术研究的工具.Flower 提供了大规模的部署方案以及网络模拟，此外也在移动、无线端提供了跨平台支持，适合作为模拟大规模场景下的学术研究工具.TensorFlow Federated 和 FedML 同样缺少对线上生产的完善支撑.

可视化方法可以帮助研究人员形象了解复杂模型的本质和过程[47] ，进行模型设计和模型调试等任务[48].截至目前，只有 FATE 和 FedML 提供了可视化功能，其他框架尚未推出相关功能.相对于其他框架，FATE 设计了针对模型训练过程的可视化组件，可以记录联邦学习的全流程.对于联邦学习了解较少的研究人员来说，使用 FATE 可以更好地跟进联邦学习过程，完成模型调试等操作.

异步聚合方面，FATE 提供异步聚合的接口，其他框架未直接提供异步聚合接口，但可以基于框架进行 2 次开发实现异步聚合的功能.现有原生的框架本身都提供了比较健全、完善的方法接口，用户可以根据需求编写自己的算法来满足不同的联邦学习场景的需要，如用户掉线场景.

5 基于开源框架的联邦学习实验

为了帮助研究人员更好地根据应用需求选择合适的联邦学习框架，本节以 2 个不同的场景为例，阐述如何选择以及搭建联邦学习框架，并基于搭建的框架进行实验.

5.1 应用场景 1：基于 IOT 的图像识别

随着智能设备感知能力和计算能力的不断提升，越来越多的设备具备了微型计算机的能力.而如何利用这些智能设备的本地数据和计算资源创造更有益的技术成为了目前的研究热点之一.因此，我们选择的第 1 个应用场景是：在 IOT 网络中，通过在分散的 IOT 设备中训练一个图像分类模型，使得其可以准确识别、分类图片.

对于此应用场景，第 1 个需求是需要在 IOT 网络中部署联邦学习框架，分析已有的开源框架，目前 FedML 和 Flower 支持该需求.我们以 FedML 为例搭建联邦学习框架，利用树莓派搭建 IOT 设备集群，在 MNIST 和 CIFAR10 数据集上实现图片的分类任务.

5.1.1 实验基本设置

该系统主要包含 4 个树莓派，并将其中 1 台作为中心服务器，用于模型聚合；另外 3 台作为客户机，基于本地数据集进行训练并上传模型.

算法：采用常规的联邦学习算法 Fedavg.

其他参数：采用 Adam 优化器，学习率为 0.01.

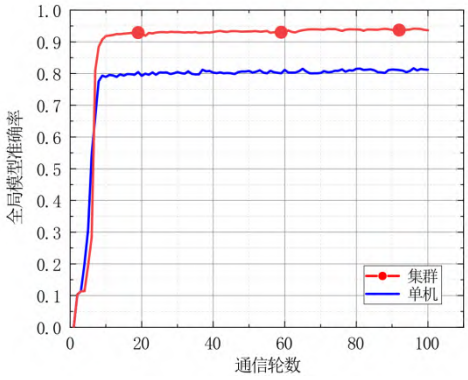
数据集：为了测试搭建框架在不同数据集以及不同模型上的训练效果，分别基于真实数据集 MNIST 及 CIFAR10 进行实验.

对比方法：①集群训练.联合 3 台客户机的本地数据进行分布式联邦学习模型训练.②单机本地训练.单台客户机基于本地数据集进行模型训练.

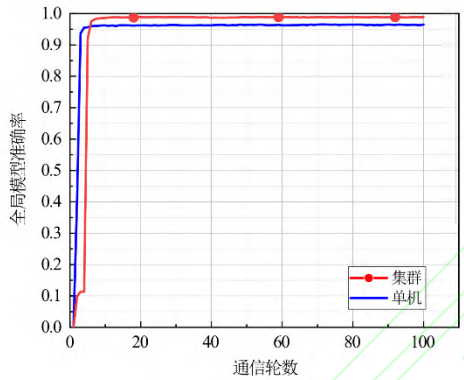
5.1.2 基于 MNIST 数据集实验

MNIST 数据集包含了 60000 个样本（50000 个训练集，10000 个测试集）.在 MNIST 数据集中的每张图片由 28×28 个像素点构成，每个像素点用 1 个灰度值表示.采用 MobileNet 模型，模型由 13 个 Depthwise Separable 卷积层、1 个二元自适应均值汇聚层和 1 个全连接层构成.图 28 分别分析了客户端数据为 500 和 5000 这 2 种情况下的联邦学习和单机的模型精度收

敛曲线.相比于单机本地训练，图 28（a）下联邦学习可以提升 12.4%的精度；图 28（b）下联邦学习可以提升 2.3%的精度.可见，在数据缺乏的情况下，联邦学习可以带来的更大的收益.



（a）500 样本数据的客户机



（b）5000 样本数据的客户机

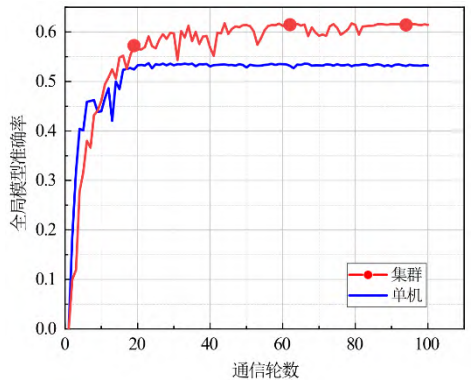
Fig.28 Model convergence curves under MNIST dataset

图 28 MNIST 数据集下模型收敛曲线

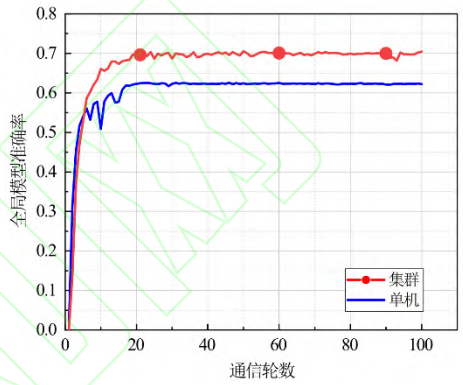
5.1.3 基于 CIFAR10 数据集实验

CIFAR-10 是 1 个包含 60000 张图片的数据集.其中每张照片为 32×32 的彩色照片，每个像素点包括 RGB3 个数值，数值范围 0~255.所有照片分属 10 个不同的类别，分别是“airplane”“automobile”“bird”“cat”“deer”“dog”“frog”“horse”“ship”“truck”.其中 50000 张图片被划分为训练集，剩下的 10000 张图片属于测试集.

模型采用 2 个 5×5 的卷积网络（每层有 64 个通道），2×2 最大池化层，2 个完全连接层，分别包含 384 个和 192 个单元，最后是一个 softmax 输出层.该实验设置单机随机 1500 个数据（图 29 （a））和 5000 个数据（图 29（b））作为对比，观测在联邦学习和单机模式下训练模型准确率的收敛曲线.2 组对照实验中，联邦学习的精度整体提升了 8%，相较于传统的单机模式性能有所提升.



（a）1500 样本数据的客户机



（b）5000 样本数据的客户机

Fig.29 Model convergence curves under CIFAR-10 dataset

图 29 CIFAR-10 数据集下模型收敛曲线

5.2 应用场景 2：医疗系统中的辅助病理诊断

人工病理诊断一般都要求医院具有较强的技术支持和医生具有较丰富的经验.而在一些医疗资源相对欠缺的区域，如果医生无法通过一些仪器进行辅助诊断，则可能存在误诊与漏诊情况.目前，人工智能的发展大大推动了智能药物研发、辅助医疗诊断、基因特性分析的发展，使得不同医疗机构之间共享医疗资源成为可能.因此，第 2 个应用场景是：通过学习不同医疗机构的医疗数据，训练一个可以辅助医生进行病理诊断的模型.

对于此应用场景，我们最基本的设计需求是需要搭建一个支持数据隐私保护的分布式学习架构.虽然已有开源框架几乎都支持该需求，但是由于医院对其医疗数据的安全性要求特别高，且目前工业支持和应用较为成熟的只有 FATE，因此，我们选择以 FATE 为例搭建病理诊断模型的联邦学习系统，选用 UCI 数据库中 Wisconsin 州乳腺癌数据集[49]进行模型训练.

5.2.1 配置文件

dsl 文件：用来描述任务模块，以有向无环图

（DAG）的形式组合任务模块；

conf 文件：设置各个组件的参数，如输入模块的数据表名；算法模块的学习率、batch 大小、迭代次数等.

5.2.2 参与各方角色

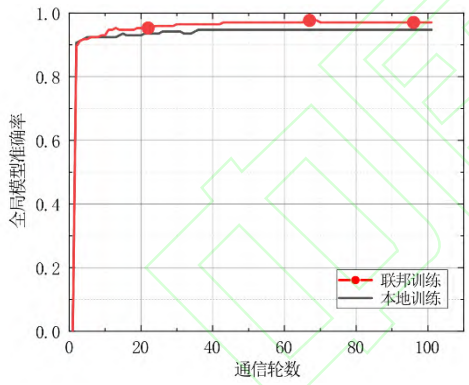
Guest：任务的发起者，代表数据应用方；

Host：数据提供方，为 Guest 提供数据；

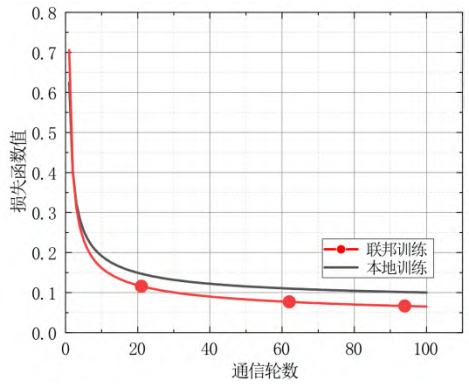
Arbiter：辅助多方完成联合建模，主要作用是聚合梯度或者模型.比如聚合各方本地模型，各方将自身梯度模型参数发送给 Arbiter，Arbiter 进行联合优化等.

5.2.3 横向联邦场景

数据集：采用 Breast Cancer（乳腺癌数据集）作为实验数据.Breast Cancer 数据集有 569 组 31 维实例数据，30 维的诊断属性包括 radius 半径（从中心到边缘上点的距离的平均值），texture 纹理（灰度值的标准偏差）等，1 维的标签类分为 WDBC-Malignant 恶性和 WDBC-Benign 良性.前 469 条数据作为训练样本，后面的 100 条数据作为测试样本.从 469 条训练样本中，选取前 200 条样本作为参与方 A（host）的本地数据；将剩余的 269 条样本作为参与方 B（guest）的本地数据；测试数据共享.



（a）模型准确率收敛曲线



（b）损失函数收敛曲线

Fig.30 Breast Cancer horizontal federated learning

图 30 Breast Cancer 横向联邦学习

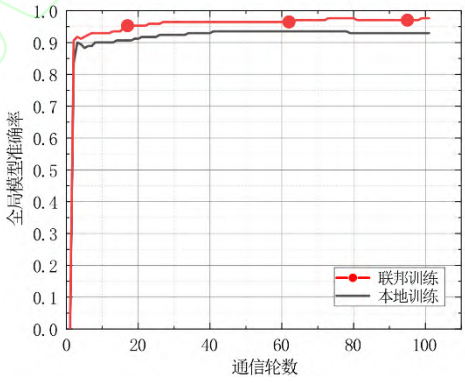
任务：采用乳腺癌数据集在 Logistic Regression 模型下进行医疗病例诊断.

对比方法：1)联邦训练.联合参与方 A，B 进行联邦学习模型训练.2)本地训练.参与方 B 基于本地数据进行模型训练.

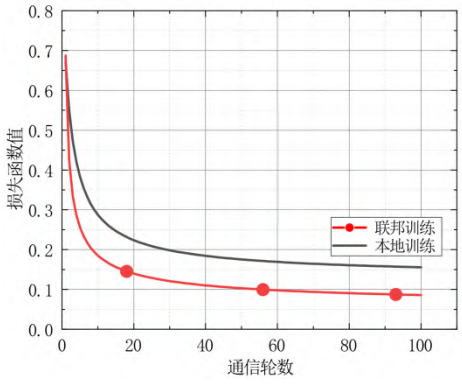
实验结果如图 30 所示，图 30（a）为模型准确率的收敛曲线，在整个训练过程中，双方都达到了不错的效果，但是采用联邦学习的方式，相比于单机模式提高了 2.3%的精度.在图 30（b）模型损失函数的收敛曲线中，联邦学习的方式使得损失的以更快的速度达到最小.相比于本地训练，联邦学习下模型的准确率有所提升.横向联邦学习可以联合多个参与者的具有相同特征的数据样本，提高样本的丰富性.

5.2.4 纵向联邦场景

数据集：采用 Breast Cancer 数据集作为实验数据.前 400 条数据作为训练样本，后 169 条数据作为评估测试样本.从 400 条训练样本中，选取乳腺癌数据集前 20 个特征作为参与方 A（host）的本地数据；后 10 个特征以及标签作为参与方 B（guest）的本地数据；测试数据参与方 B 独享.



（a）模型准确率收敛曲线



（b）损失函数收敛曲线

Fig.31 Breast Cancer vertical federated learning

图 31 乳腺癌纵向联邦学习

任务：采用乳腺癌数据集在逻辑回归模型下进行医疗病例诊断.

对比方法：1）联邦训练.联合参与方 A，B 进行联邦学习模型训练.2）本地训练.参与方 B 基于本地数据进行模型训练.

实验结果如图 31 所示.图 31（a）为模型准确率的收敛曲线.在准确率方面：联邦学习的结果相较于本地训练这种方式，准确率提升了 4.7%.在训练速度方面：联邦学习的结果也比本地训练加快了约 10 轮.图 31（b）为模型损失函数的收敛曲线，可以明显看出联邦学习的损失函数以更快的速度达到收敛.因此，从实验结果可以看出纵向联邦学习可以联合多个参与者的共同样本的不同数据特征，丰富样本特征，提高模型的识别精度.

总之，上述实验表明，横向和纵向联邦学习都可以有效提高模型的训练效果，同时，联邦学习的计算范式保证了用户本地数据的隐私安全.此外，在开展联邦学习时，需要根据场景选择合适的联邦学习模式，缺少样本数据可以选择横向联邦学习；缺少样本特征可以选择纵向联邦学习.

5.3 框架训练效率对比

为了进一步对比不同框架的效率，我们基于 FedML 和 FATE 框架分别对乳腺癌数据集进行横向联邦学习模型训练和纵向联邦学习模型训练，从时间角度来比较 2 个框架间的效率差异.服务器配置清单：CPU 为 i5-4460，1.86GHz，6 核；内存容量为 16GB；网络带宽为 30Mbps；操作系统为 CentOS Linux release 7.9.2009.

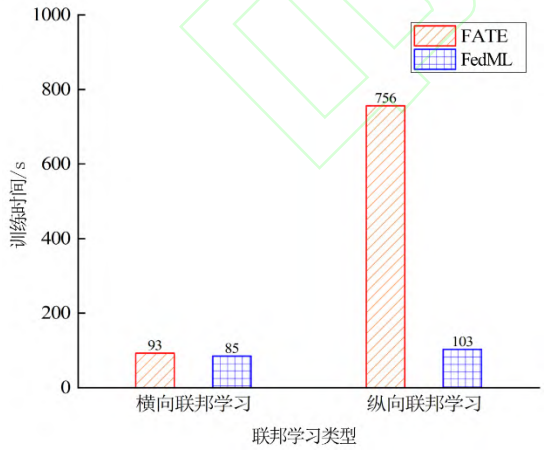


Fig. 32 Comparison of training time of different frame models

图 32 不同框架模型训练时间对比图

实验结果如图 32 所示，横向联邦到达指定目标准确率，FedML 框架比 FATE 框架耗时少 8s，有 8%左右的速度优势.究其原因，FedML 相比于 FATE 更轻量级，时间优势主要是在框架启动过程中，但整体上 2 个框架在横向联邦学习训练效率上比较相近.纵向联邦到达指定目标准确率，FedML 框架只需要 103s，相比于 FATE 框架速度提升了 86%.这主要原因是 FATE 框架中的纵向联邦采用了多方安全的样本对齐算法，算法基于 RSA 加密算法和散列函数实现，在增强隐私保护的同时严重降低了模型的训练效率.

FATE 框架相比于 FedML 有隐私保护机制支持，因而不可避免的造成效率上的损失.其他框架也是如此，追求数据的高度隐私安全会造成训练效率的急剧下降，但可以通过并行加密计算等技术加速训练，提高效率.

6 挑战与展望

尽管联邦学习已有实际落地的项目，但是其仍然处于发展初期，对于各种复杂的学习场景还有很多待解决的挑战问题以及待提升的技术，通过对目前联邦学习开源框架研究和应用现状的分析，本文总结了以下几点挑战与展望.

1）隐私安全.

隐私安全问题是联邦学习研究的动因.虽然通过训练数据不离开本地进行训练可以保护数据隐私信息的安全性，但在实际应用过程中，由于联邦学习的训练过程包含多个复杂环节，因而每个环节依然面临许多安全与隐私挑战，如投毒攻击[50-51]、基于 GAN[52-53]的攻击、推理攻击[54]、通信节点攻击和中央服务器攻击等.且多数实验证明了即使只上传模型的相关参数，仍然能反推出终端的隐私数据[55-56].针对以上威胁，联邦学习框架在安全和隐私保护方面采取了一系列的措施，如差分隐私[57-58]、同态加密[59-60]和安全多方计算[61-62]等.虽然上述技术能实现安全联邦学习，但仍然存在一些易受攻击的漏洞需要填补.此外，采用隐私技术的同时也会造成较大的通信开销，因此如何平衡模型安全和通信也是一个相当大的挑战.

2）效率、准确性和隐私的权衡.

增强联邦学习的隐私保护，在一定程度上牺牲了模型训练效率与结果准确性.在联邦学习场景下，一个模型训练包括大规模的数据样本和复杂计算，此时运算效率是一个重要问题.联邦学习框架采用复杂的加密系统以保护隐私，而这需要庞大的计算量和通信量.网络带宽有限、设备掉线等通信问题都会使传输的数据量和传输速度下降[63-64]，从而导致联邦学习的效率不高.结果准确性方面，如果加密级别太高或添加噪声过多，无疑会降低模型的准确性.因此，如何在保障隐私安全的前提下提高联邦学习的训练效率

和结果准确性，实现安全、效率和准确性之间的平衡[64-66]，是联邦学习框架需要解决的问题.

3）激励机制.

联邦学习是一个多方数据联盟的技术，只有提高参与用户的数量才能训练出更精确有效的模型.如果没有采取合适的激励机制，那么获取的数据和训练的模型则质量受限.同时，各方可能因利益冲突、互不信任等问题导致最终合作失败.如何制定合理的激励机制，是联邦学习框架面临的一大挑战.未来，联邦学习框架不仅需要考虑隐私保护，更要考虑如何通过共识机制实施公平激励[67-68]，以实现联邦集体利益最大化.各框架可以通过计算各参与方对于模型的贡献，建立数据记录机制（例如记录于区块链中[69-70]）以激励不同程度参与方，从而形成最优联邦组织.

4）异构性与个性化.

现有的联邦学习框架要求全体参与方训练出一致的全局模型，而这在实际复杂的物联网应用中是不现实的.由于设备、统计和模型的异构性[71]，现有的联邦学习模型不能直接在物联网设备中有效应用.为了解决异构性挑战，联邦框架需要考虑个性化处理[72-74]，让每个设备获得高质量的个性化模型，如采取多任务和元学习的方法.

5）跨框架交互.

随着联邦学习逐渐进入大众视野，不同行业的公司都推出了各自的联邦学习框架.而这在丰富市场选择中，也出现了新的问题：由于各框架技术实现的差异和安全协议的区别，不同框架所托管的数据在实际应用中无法跨框架交互.不同联邦学习技术框架之间互联的阻碍，限制了跨行业数据的交流和行业间融合互通，制约了数据价值的释放.因此，联邦学习框架应朝着数据跨框架交流、算法跨框架部署、任务跨框架执行的方向发展，使不同行业可以在统一的标准下进行联邦学习，实现行业互联互通，推进数字化融合发展.

7 结论

作为破解“数据孤岛”问题和保障隐私安全的有效手段，联邦学习的重要性日益凸显.而联邦学习的有效应用主要依托于开源框架的研究和建设.因此，考虑到联邦学习开源框架的重要性，本文重点从系统架构、系统功能、版本变化 3 方面介绍开源框架 FATE，PySyft，TensorFlow Federated，Paddle FL，FedML，Flower.并从隐私机制、机器学习算法、计算范式、学习类型、训练架构、通信协议、可视化等方面对比总结了各框架的优劣势.为了更好地帮助读者搭建开源框架，文章给出了 2 个不同应用场景框架搭建的实例.并基于目前框架存在的开放性问题，从隐私安全、激励机制与置信规则、跨框架交互等方面讨论了未来可能的研究发展方向.总之，文章对于联邦学习未来的研究具有较好的参考意义，可为开源框架的建设创新、结构优化、安全优化以及算法优化等提供有效思路.

作者贡献声明：林伟伟提出文章的整体思路和框架；石方和曾岚负责撰写论文；李董东和许银海负责完成实验；刘波提出指导意见并修改论文.

参 考 文 献

[1] Kang Yiping, Hauswald J, Gao Cao, et al. Neurosurgeon: Collaborative intelligence between the cloud and mobile edge [J]. ACM SIGARCH Computer Architecture News, 2017, 45(1): 615-629

[2] McMahan B, Ramage D. Federated learning: Collaborative machine learning without centralized training data, [CP/OL] [2021-12-26]. https://ai.googleblog.com/2017/04/federated-learning collaborative.html

[3] McMahan H B, Moore E, Ramage D, et al. Federated learning of deep networks using model averaging [J]. arXiv preprint, arXiv:1602.05629, 2016

[4] McMahan B, Moore E, Ramage D, et al. Communication-efficient learning of deep networks from decentralized data [G] //Artificial Intelligence and Statistics. New York: PMLR, 2017: 1273-1282

[5] Liu Yang, Fan Tao, Chen Tianjian, et al. FATE: An industrial grade platform for collaborative learning with data protection [J]. Journal of Machine Learning Research, 2021, 22(226): 1-6

[6] Bonawitz K, Eichner H, Grieskamp W, et al. Towards federated learning at scale: System design [J]. Proceedings of Machine Learning and Systems, 2019, 1: 374-388

[7] Ingerman A, Ostrowski K. TensorFlow Federated [CP/OL]. (2019-03-06)[2021-12-28]. https://blog.tensorflow.org/2019/03/introducing-tensorflow-federated.html

[8] Ryffel T, Trask A, Dahl M, et al. A generic framework for privacy preserving deep learning [J]. arXiv preprint, arXiv:1811.04017, 2018

[9] He Chaoyang, Li Songze, So Jinhyun, et al. FedML: A research library and benchmark for federated machine learning [J]. arXiv preprint, arXiv:2007.13518, 2020

[10] Beutel J, Daniel J, et al. Flower: A friendly federated Learning Research Framework [J]. arXiv preprint, arXiv:2007.14390, 2020

[11] Ma Yanjun, Yu Dianhai, Wu Tian, et al. PaddlePaddle: An open-source deep learning platform from industrial practice [J]. Frontiers of Data and Computing, 2019, 1(1): 105-115

[12] He Chaoyang, Tan Conghui, Tang Hanlin, et al. Central 服务器 free federated learning over single-sided trust social networks [J]. arXiv preprint, arXiv:1910.04956, 2019

[13] Yang Qiang, Liu Yang, Chen Tianjian, et al. Federated machine learning: Concept and applications [J]. ACM Transactions on Intelligent Systems and Technology , 2019, 10(2): 1-19

[14] Vaidya J, Clifton C. Privacy preserving association rule mining in

vertically partitioned data [C] //Proc of the 8th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. New York:ACM, 2002: 639–644

[15] Pan S J, Yang Qiang. A survey on transfer learning [J]. IEEE Transactions on knowledge and data engineering, 2010, 22(10): 1345–1359

[16] Leroy D, Coucke A, Lavril T, et al. Federated learning for keyword spotting [C] //Proc of the 2019 IEEE Int Conf on Acoustics, Speech and Signal Processing. Piscataway, NJ: IEEE, 2019: 6341–6345

[17] Hard A, Rao K, Mathews R, et al. Federated learning for mobile keyboard prediction [J]. arXiv preprint, arXiv:1811.03604,2018

[18] Li Qinbin, Wen Zeyi, He Bingsheng. Practical federated gradient boosting decision trees [J]. arXiv preprint, arXiv:1911.04206, 2019

[19] Yang Kai, Fan Tao, Chen Tianjian, et al. A quasinewton method based vertical federated learning framework for logistic regression [J]. arXiv preprint, arXiv:1912.00513, 2019

[20] Yang Shengwen, Ren Bing, Zhou Xuhui, et al. Parallel distributed logistic regression for vertical federated learning without third-party coordinator [J]. arXiv preprint, arXiv:1911.09824, 2019

[21] Zhu Xinghua, Wang Jianzong, Hong Zhenhou, et al. Federated learning of unsegmented Chinese text recognition model [C] //Proc of the 31st IEEE Int Conf on Tools with Artificial Intelligence. Piscataway, NJ: IEEE, 2019: 1341-1345

[22] Caldas S, Duddu S M K, Wu P, et al. LEAF: A benchmark for federated Settings [J]. arXiv preprint, arXiv:1812.01097, 2018

[23] Mohassel P, Rindal P. ABY3: A mixed protocol framework for machine learning [C] //Proc of the 2018 ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2018: 35-52

[24] Smith V, Chiang C K, Sanjabi M, et al. Federated multi-task learning [C/OL] //Proc of the 31st Int Conf on Neural Information Processing Systems. 2017: 4427–4437 [2021-12-29]. https://proceedings.neurips.cc/paper/2017/hash/6211080fa89981f66b1a0c9d55c61d0f-Abstract.html

[25] Abadi M, Chu A, Goodfellow I, et al. Deep learning with differential privacy [C] //Proc of the 2016 ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2016: 308–318 （没有届）

[26] Wainakh A, Guinea A S, Grube T. Enhancing privacy via hierarchical federated learning [J]. arXiv preprint, arXiv:2004.11361, 2020

[27] Liao Feng, Zhuo H H, Huang Xiaoling, et al. Federated hierarchical hybrid networks for clickbait detection [J]. arXiv preprint, arXiv:1906.00638, 2019

[28] Hardy S, Henecka W, Ivey-Law H, et al. Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption [J]. arXiv preprint, arXiv:1711.10677, 2017

[29] Gupta O, Raskar R. Distributed learning of deep neural network over multiple agents [J]. Journal of Network and Computer Applications, 2018, 116: 1-8

[30] Sun Z, Kairouz P, Suresh A T, et al. Can you really backdoor federated learning? [J]. arXiv preprint, arXiv:1911.07963, 2019

[31] Pillutla K, Kakade S M, Harchaoui Z. Robust aggregation for federated learning [J]. arXiv preprint, arXiv:1912.13445, 2019

[32] Blanchard P, Mhamdi E, Guerraoui R, et al. Machine learning with adversaries: Byzantine tolerant gradient descent [C/OL] //Proc of the 2017 Advances in Neural Information Processing Systems. 2017: 119–129[2021-12-28]. https://proceedings.neurips.cc/paper/2017/hash/f4b9ec30ad9f68f89b29639786cb62ef-Abstract.html

[33] Bagdasaryan E, Veit A, Hua Yiqing, et al. How to backdoor federated learning [C/OL] //Proc of the 23rd Int Conf on Artificial Intelligence and Statistics. 2018: 2938–2948[2021-12-29]. https://proceedings.mlr.press/v108/bagdasaryan20a.html

[34] Wang Hongyi, Sreenivasan K, Rajput S, et al. Attack of the tails: Yes, you really can backdoor federated learning[J]. arXiv preprint, arXiv:2007.05084, 2020

[35] He Chaoyang, Annavaram M, Avestimehr S. FedNAS: Federated deep learning via neural architecture search [J]. arXiv preprint, arXiv:2004.08546, 2020

[36] LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition [J]. Proceeding of the IEEE, 1998, 86(11):2278–2324

[37] Reddi S, Charles Z, Zaheer M, et al. Adaptive federated optimization [J]. arXiv preprint, arXiv:2003.00295, 2020

[38] Li Tian, Sahu A K, Zaheer M, et al. Federated optimization in heterogeneous networks[J]. arXiv preprint, arXiv:1812.06127, 2018

[39] Krizhevsky A, Hinton G. Learning multiple layers of features from tiny images [J/OL]. 2009[2022-01-08]. https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.222.9220

[40] Liu Yang, Ma Zhuo, Liu Ximeng, et al. Boosting privately: Privacy-preserving federated extreme boosting for mobile crowdsensing [J]. arXiv preprint, arXiv:1907.10218, 2019

[41] Agarwal N, Kairouz P, Liu Z. The skellam mechanism for differentially private federated learning [C/OL] //Proc of the 2021 Advances in Neural Information Processing Systems. 2021:5052-5264 [2022-01-08]. https://proceedings.neurips.cc/paper/2021/hash/285baacbdf8fda1de94b19282acd23e2-Abstract.html

[42] A multi-party collaborative machine learning framework [EB/OL]. (2020-10-26)[2022-01-08]. https://github.com/bytedance/fedlearner

[43] Lin Yuchen, He Chaoyang, Zeng Zihang, et al. FedNLP: A research platform for federated learning in natural language processing [J]. arXiv preprint, arXiv:2104.08815, 2021

[44] Xie Yuexiang, Wang Zhen, Chen Daoyuan, et al. Federatedscope: A flexible federated learning platform for heterogeneity[J]. arXiv preprint, arXiv:2204.05011, 2022

[45] Wang Zhen, Kuang Weirui, Xie Yuexiang, et al. FederatedScope-GNN: Towards a unified, comprehensive and efficient package for federated graph learning[J]. arXiv preprint, arXiv:2204.05562, 2022

[46] Powell K. NVIDIA Clara Federated Learning to Deliver AI to Hospitals While Protecting Patient Data [CP/OL]. (2019-12-01)[2021-01-08]. https://blogs.nvidia.com/blog/2019/12/01/clara-federated-learning/

[47] Yosinski J, Clune J, Nguyen A, et al. Understanding neural networks through deep visualization [J]. arXiv preprint, arXiv:1506.06579, 2015

[48] Du Memgnan, Liu Ninghao, Hu Xia. Techniques for interpretable machine learning [J]. arXiv preprint, arXiv:1808.00033, 2018

[49] Street W N, Wolberg W H, Mangasarian O L. Nuclear feature extraction for breast tumor diagnosis [G] //SPIE 1905: Proc of the 1993 Biomedical Image Processing and Biomedical Visualization. Bellingham: SPIE, 1993: 861-870

[50] Bhagoji A N, Chakraborty S, Mittal P, et al. Analyzing federated learning through an adversarial lens [C/OL] //Proc of the 36th Int Conf

on Machine Learning. 2019:634-643[2022-01-08]. https://proceedings.mlr.press/v97/bhagoji19a.html

[51] Fang Minghong, Cao Xiaoyu, Jia Jinyuan, et al. Local model poisoning attacks to byzantine-robust federated learning [C] //Proc of the 29th USENIX Security Symp. Berkeley, CA: USENIX Association, 2020: 1605-1622

[52] Hitaj B, Ateniese G, Perez-cruz F. Deep models under the GAN: Information leakage from collaborative deep learning [C] //Proc of the 2017 ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2017: 603-618

[53] Zhang Jiale, Chen Junjun, Wu Di, et al. Poisoning attack in federated learning using generative adversarial nets [C] //Proc of the 18th IEEE Int Conf on Trust, Security and Privacy in Computing and Communications 13th IEEE Int Conf on Big Data Science and Engineering. Piscataway, NJ: IEEE, 2019: 374–380

[54] Melis L, Song Congzheng, De Cristofaro E, et al. Exploiting unintended feature leakage in collaborative learning [C] //Proc of the 2019 IEEE Symp on Security and Privacy. Piscataway, NJ: IEEE, 2019: 691–706

[55] Phong L T, Aono Y, Hayashi T, et al. Privacy-preserving deep learning via additively homomorphic encryption[J]. IEEE Transactions on Information Forensics and Security, 2017, 13(5): 1333-1345

[56] Zhu Ligeng, Liu Zhijian, Han Song. Deep leakage from gradients[C/OL] //Proc of the 2019 Advances in Neural Information Processing Systems. 2019 [2022-01-08]. https://proceedings.neurips.cc/paper/2019/hash/60a6c4002cc7b29142def8871531281a-Abstract.html

[57] Wei Kang, Li Jun, Ding Ming, et al. Federated learning with differential privacy: Algorithms and performance analysis [J]. IEEE Transactions on Information Forensics and Security, 2020, 15: 3454-3469

[58] Bhowmick A, Duchi J, Freudiger J, et al. Protection against reconstruction and its applications in private federated learning [J]. arXiv preprint, arXiv:1812.00984, 2018

[59] Xu Guowen, Li Hongwei, Liu Sen, et al. VerifyNet: Secure and verifiable federated learning [J]. IEEE Transactions on Information Forensics and Security, 2019, 15: 911-926

[60] Yuan Jiawei, Yu Shucheng. Privacy preserving back-propagation neural network learning made practical with cloud computing [J]. IEEE Transactions on Parallel and Distributed Systems, 2013, 25(1): 212-221.

[61] Wan Li, Ng W K, Han Shuguo, et al. Privacy-preservation for gradient descent methods [C] //Proc of the 13th ACM SIGKDD Int Conf Knowledge Discovery and Data Mining. New York: ACM, 2007: 775-783

[62] Bonawitz K, Ivanov V, Kreuter B, et al. Practical secure aggregation for privacy-preserving machine learning [C] //Proc of the 2017 ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2017: 1175-1191

[63] Hamer J, Mohri M, Suresh A T, FedBoost: A communication-efficient algorithm for federated learning [C/OL] //Proc of the 37th Int Conf on Machine Learning. 2020: 3973-3983[2022-01-08]. https://proceedings.mlr.press/v119/hamer20a.html

[64] Gogineni V C, Werner S, Huang Y F, et al. A. Communication-efficient online federated learning framework for nonlinear regression [C/OL] //Proc of the 2022 IEEE Int Conf on Acoustics, Speech and Signal Processing. 2022: 5228-5232[2022-01-08]. https://ieeexplore.ieee.org/abstract/document/9746228

[65] Song Jincheng, Wang Weizheng, Gadekallu T R, et al. EPPDA: An efficient privacy-preserving data aggregation federated learning scheme [J/OL]. IEEE Transactions on Network Science and Engineering, 2022[2022-01-08]. https://ieeexplore.ieee.org/abstract/document/9721557

[66] Chen Hao, Huang Shaocheng, Zhang Deyou, et al. Federated learning over wireless IoT networks with optimized communication and resources [J/OL]. IEEE Internet of Things Journal, 2022[2022-01-08]. https://ieeexplore.ieee.org/abstract/document/9712615

[67] Yu Han, Liu Zelei, Liu Yang, et al. A sustainable incentive scheme for federated learning[J]. IEEE Intelligent Systems, 2020, 35(4): 58–69

[68] Zhan Yufeng, Zhang Jiang, Li Peng. Crowdtraining: Architecture and incentive mechanism for deep learning training in the internet of things[J]. IEEE Network, 2019, 33(5): 89–95

[69] Kim H, Park J, Bennis M, et al. Blockchained on-device federated learning[J]. IEEE Communications Letters, 2020, 24(6): 1279-1283

[70] Lu Yunlong, Huang Xiaohong, Dai Yueyue, et al. Blockchain and federated learning for privacy-preserved data sharing in industrial IoT[J]. IEEE Transactions on Industrial Informatics, 2020, 16(6): 4177-4186

[71] Xie Cong, Koyejo S, Gupta I. Asynchronous federated optimization[J]. arXiv preprint, arXiv:1903.03934, 2019

[72] Wu Qiong, He Kaiwen, Chen Xu. Personalized federated learning for intelligent IoT applications: A cloud-edge based framework [J]. IEEE Open Journal of the Computer Society, 2020, 1: 35-44

[73] Tan A Z, Yu Han, Cui Lizhen, et al. Towards personalized federated learning [J/OL]. IEEE Transactions on Neural Networks and Learning Systems. 2022 [2022-01-18]. https://ieeexplore.ieee.org/abstract/document/9743558

[74] Pei Jiaming, Zhong Kaiyang, Jan M A, et al. Personalized federated learning framework for network traffic anomaly detection [J/OL]. Computer Networks, 2022, 209 [2022-05-24]. https://www.sciencedirect.com/science/article/abs/pii/S1389128622001001

Lin Weiwei, born in 1980. PhD, professor, PhD supervisor. Senior member of CCF. His main research interests include cloud computing, big data, and application technology of artificial intelligence.

林伟伟，1980 年生.博士，教授，博士生导师， CCF 高级会员. 主要研究方向为云计算、大数据、人工智能应用技术.

Shi Fang, born in 1993. PhD candidate. student member of CCF. Her main research interests include cloud computing, federate

learning.
石方，1993 年生.博士研究生, CCF 学生会员.主要研究方向为联邦学习、云计算.
E-mail: 978772638@qq.com

Zeng Lan, born in 2001. Undergraduate. Her main research interest is federate learning.
E-mail: 1227460497@qq.com
曾岚，2001 年生.本科生.主要研究方向为联邦学习.

Li Dongdong, born in 1994. PhD candidate. His main research interests include big data, and federate learning.
李董东，1994 年生.博士研究生.主要研究方向为大数据和联邦学习.
E-mail: dongdonglee1994@foxmail.com>

Xu Yinhai, born in 1998. Master candidate. His main research interests include big data, and federate learning.
许银海，1998 年生.硕士研究生.主要研究方向为大数据和联邦学习.
E-mail: xu13584593584@163.com

Liu Bo, born in 1968. PhD, professor. His main research interests include distributed computing, and artificial intelligence.
刘波， 1968 年生.博士，教授.主要研究方向为分布式计算和人工智能.