AccFed：物联网中基于模型分割的联邦学习加速

曹绍华* 陈 辉 陈 舒 张汉卿 张卫山
(中国石油大学(华东)计算机科学与技术学院 青岛 266580)

摘 要：随着物联网(IoT)的快速发展，人工智能(AI)与边缘计算(EC)的深度融合形成了边缘智能(Edge AI)。但由于IoT设备计算与通信资源有限，并且这些设备通常具有隐私保护的需求，那么在保护隐私的同时，如何加速Edge AI仍然是一个挑战。联邦学习(FL)作为一种新兴的分布式学习范式，在隐私保护和提升模型性能等方面，具有巨大的潜力，但是通信及本地训练效率低。为了解决上述难题，该文提出一种FL加速框架AccFed。首先，根据网络状态的不同，提出一种基于模型分割的端边云协同训练算法，加速FL本地训练；然后，设计一种多轮迭代再聚合的模型聚合算法，加速FL聚合；最后实验结果表明，AccFed在训练精度、收敛速度、训练时间等方面均优于对照组。

关键词：边缘智能；联邦学习；端边云协同；模型分割
中图分类号：TN929.5; TP399 **文献标识码**：A **文章编号**：1009-5896(2023)05-1678-10
DOI: 10.11999/JEIT220240

AccFed: Federated Learning Acceleration Based on Model Partitioning in Internet of Things

CAO Shaohua CHEN Hui CHEN Shu
ZHANG Hanqing ZHANG Weishan
(*College of Computer Science and Technology, China University of Petroleum (East China), Qingdao* 266580, *China*)

Abstract: With the rapid development of Internet of Things (IoT), the deep integration of Artificial Intelligence (AI) and Edge Computing (EC) has formed Edge AI. However, since IoT devices are computationally and communicationally constrained and these devices often require privacy-preserving, it is still a challenge to accelerate Edge AI while protecting privacy. Federated Learning (FL), an emerging distributed learning paradigm, has great potential in terms of privacy preservation and improving model performance, but communication and local training are inefficient. To address the above challenges, a FL acceleration framework AccFed is proposed in this paper. Firstly, a Device-Edge-Cloud synergy training algorithm based on model partitioning is proposed to accelerate FL local training according to the different network states; Then, a multi-iteration and reaggregation algorithm is designed to accelerate FL aggregation; Finally, experimental results show that AccFed outperforms the control group in terms of training accuracy, convergence speed, training time, etc.

Key words: Edge Artificial Intelligence (AI); Federated Learning (FL); Device-edge-cloud synergy; Model partitioning

收稿日期：2022-03-08；改回日期：2022-05-11；网络出版：2022-05-20
*通信作者： 曹绍华 shaohuacao@upc.edu.cn
基金项目：国家自然科学基金(62072469)，研究生创新工程项目(YCX2021129)，中国科学院自动化研究所复杂系统管理与控制国家重点实验室开放课题(20210114)
Foundation Items: The National Natural Science Foundation of China (62072469), The Postgraduate Student Innovation Project (YCX2021129), The State Key Laboratory of Complex System Management and Control, Institute of Automation, Chinese Academy of Sciences, Open Project (20210114)

1　引言

目前，手机、可穿戴设备、工业生产设备和自动驾驶汽车等每天都产生大量数据，这些数据在物联网(Internet of Things, IoT)中具有巨大价值[1]。例如在生产环节，它们被用于人工智能(Artificial Intelligence, AI)，将第5代移动通信(the 5th Generation, 5G)与虚拟现实、增强现实、计算机视觉等技术相结合[2]，可以使企业实现对设备的远程操作、生产过程实时监测、设备的预测性维护等，有效提升了生产效率、减少了生产成本。但是传统的集中式计算在面临延敏感性场景时，对服务延迟提出了更高的要求。

随着边缘计算(Edge Computing, EC) 的兴起，它将传统的集中式计算能力下沉到靠近本地设备的地方，因而在本地存储数据并将计算推向边缘成为趋势[3]。通过将5G与边缘计算的融合，能够大幅降低业务时延，显著提升用户体验。而深度神经网络(Deep Neural Network, DNN)是当今许多AI应用背后的解决方案，DNN精确但资源需求密集，特别是对于IoT场景，设备处理能力有限。为了克服相关的资源限制，DNN的训练通常被卸载到边缘或云上，通过对模型进行分割并在两个不同的执行端来实现的[4]。

而在数据隐私方面，IoT设备生成的大量数据通常会涉及用户的私人信息，具有较高的隐私性[5,6]，并且容易受到恶意用户跟踪和撤销等安全问题，导致设备用户和服务供应者的相互不信任[7]。大量IoT设备的数据经常连接到互联网，可能包含巨大价值的信息，因而在保证数据安全的条件下，针对资源受限的IoT设备，如何进行安全、高效训练是一个亟待解决的问题。联邦学习(Federated Learning, FL)作为一种新兴的隐私保护框架，在隐私保护、提升模型性能等方面具有巨大潜力[8]。文献[9,10]通过将FL和EC的技术相结合，把本地模型训练的计算任务卸载到边缘设备上，并将数据保存在本地，通过利用分布式用户数据来提高模型性能，保障了数据安全。但由于设备异构等问题，传输成本较高、通信较慢等一系列通信问题严重限制了FL的效率[11]。

因此，针对资源受限的IoT设备，在执行边缘智能(Edge AI)应用场景(如图1所示)时，需要解决数据安全、低延迟和高效通信的难题，本文提出一种基于模型分割[12–14]的FL加速框架，具体如下：

(1) 针对IoT设备有限的计算与通信资源，难以高效地执行Edge AI任务的问题，同时考虑数据安全与模型性能，本文提出一种基于模型分割的FL加速框架AccFed。

(2) 根据网络状态的不同，提出一种基于模型分割的端边云协同训练算法，生成最佳卸载方案，加速FL本地训练。设计了一种多轮迭代再聚合的模型聚合算法，降低通信轮次，提高FL通信效率。

(3)通过分析IoT设备的数据特征，本文使用CIFAR-10数据集。为了进一步减少IoT本地训练的压力，本文采用分支DNN对数据进行训练，并

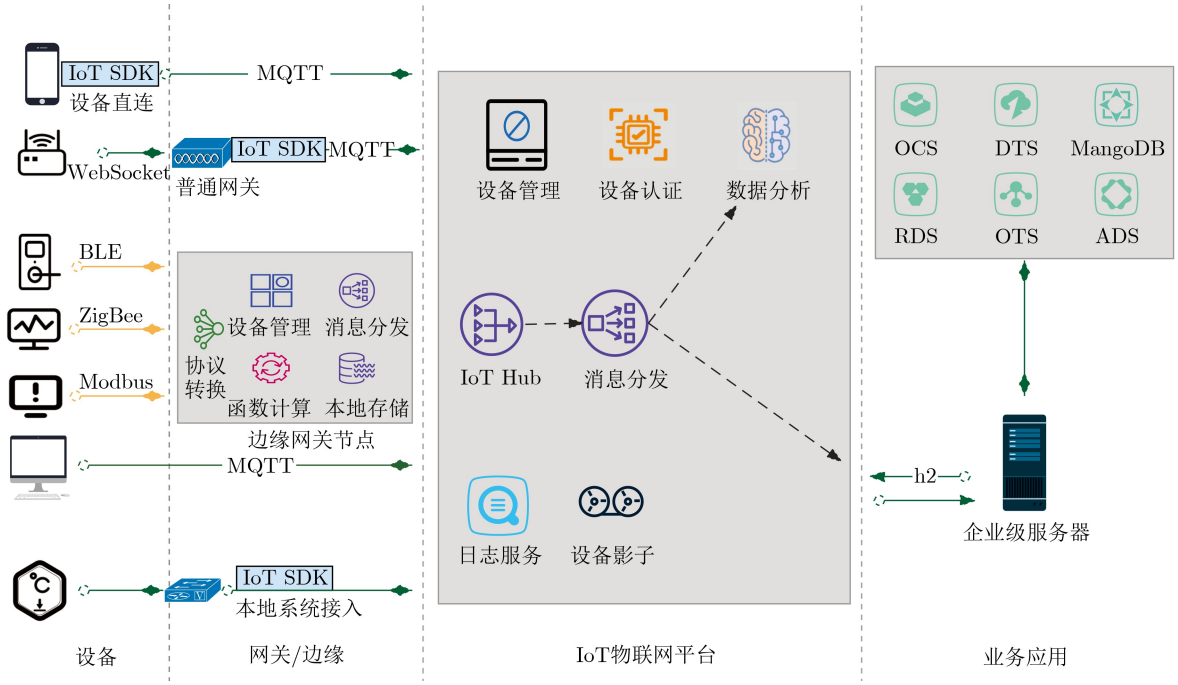

图 1 IoT场景中的Edge AI

且在树莓派、NVIDIA Jetson Xavier NX和服务器上实现了AccFed原型。最后验证了AccFed的有效性与高效性。

本文的其余部分如下：第2节回顾了Edge AI技术在IoT中应用的相关工作；第3节对AccFed框架进行概述与问题建模；第4节介绍了基于端边云协同的FL加速算法；第5节描述了AccFed的实验设置，并展示了实验结果；第6节为本文的结论。

2 相关工作

2.1 基于模型分割的IoT相关工作

在IoT领域中，越来越多的IoT设备以及AI算法被部署，以实现工业智能。然而，在IoT设备上应用计算密集型的深度学习(Deep Learning, DL)时，要满足工业制造的关键延迟要求具有挑战性。传统的集中式计算，由于沉重的传输延迟开销，仍然是低效或无效的。为了应对这一挑战，Edge AI[12,15,16]已然成为学术界与工业界的研究热点。

以DNN分割(DNN Partitioning)为代表的协同推理技术，在IoT设备和边缘服务器之间动态地分割DNN，实现Edge AI推理的即时性[4]。DNN Partitioning结合提前退出机制重塑DNN的计算量，从而减少DNN推理的总运行时间，最大限度地提高性能[16]。Serdab框架[17]将DNN计算划分到多个设备，利用只运行神经网络浅层，结合管道并行的策略，加快DNN推理的速度。上述DNN分割方案往往只是将DNN分成两部分，一部分在本地或边缘运行，另一部分在云端。他们仅仅考虑了降低推理时间，而忽略了训练时间。

2.2 基于联邦学习的IoT相关工作

由于神经网络模型的规模和复杂性不断增加，在资源受限的IoT设备上执行训练任务，以实现准确的模型推理，变得低效，甚至不可行。Guo等人[9]在FL的基础上，提出了一个联邦边缘学习(FEderated Edge Learning, FEEL)系统，设计了一个基于边缘训练的任务卸载策略来提高训练效率。在FEEL中，IoT设备向边缘服务器上传高维随机梯度，汇总后更新全局模型。虽然FEEL加速了本地训练效率，但也容易引起通信瓶颈问题。为了解决通信开销问题，很多学者采用梯度压缩的方式。Zhu等人[18]提出了一位带宽数字聚合(One-bit Broadband Digital Aggregation, OBDA)方案，在边缘设备上进行梯度量化，从而减缓通信压力。而Du等人[19]设计一个分层梯度量化框架，显著降低了通信开销，但以上方法在学习精度上均有一定的损失。

而切分学习(Split Learning, SL)与FL是最先进的分布式机器学习技术，两者均可以在不接触原始数据的情况下进行机器学习(Machine Learning, ML)。虽然相比FL, SL模型生成策略较慢，但是SL由于在客户端和服务器之间分割ML模型而提供了比FL更好的隐私，并且保证学习精度不损失[20]。同时，SL可以在资源受限的情况下进行ML训练，因为设备只训练分割后的ML网络模型的前几层，从而让IoT设备进行FL成为可能。通过将FL与SL这两种技术进行对比，本文对FL与SL的优势与不足进行了归纳与分析[20–22]，如表1所示。

与上述工作类似，本文共同关注了Edge AI场景下IoT设备资源受限的问题，并从框架和算法实现的角度提出了可行的解决方案。但是本文的工作与上述工作最大的区别在于，他们大多数工作是在协同推理方面做出贡献，在端边云协同训练方面却少有研究，即使仅有的端边协同训练研究只是集中在FL单一过程优化(如本地训练)，没有考虑多方面优化(如通信开销)。因而将SL与FL的优势相结合便成为本文的研究方向。同时，FL与SL都面临通信成本较高的问题，如何提升通信效率也是本文的重要工作之一。

3 AccFed框架

3.1 框架概述

为了更好地描述AccFed框架工作机制，本文引入信任区(Trust Zone, TZ)的概念。信任区内部包括资源受限的IoT设备以及相应的边缘服务器(如图2中的TZ1与TZ2)，当进行FL本地训练时，根据网络状况与优化目标生成卸载策略，自适应进行DNN模型分割，通过端边协同计算，加速FL本地训练。信任区内部进行协同计算时，设备与边缘服务器之间默认相互信任，无需采用加密算法。而在信任区外部进行FL时，需要采用加密算法。信任区在功能上等同于计算较强的参与者。而针对非资源受限的设备，则直接参与FL。

AccFed执行FL的过程：首先，针对IoT设备是否受限，选择相应的卸载策略，进行FL的本地训练阶段；然后将训练参数经过多轮迭代之后再上传至云端，通过差分隐私技术进行保护上传参数，

表 1 AccFed与FL, SL各项指标对比

指标	FL	SL	AccFed
构建模型	快	慢	快
隐私性	中等	优秀	优秀
计算卸载	无	有	有
通信成本	中等	高	低

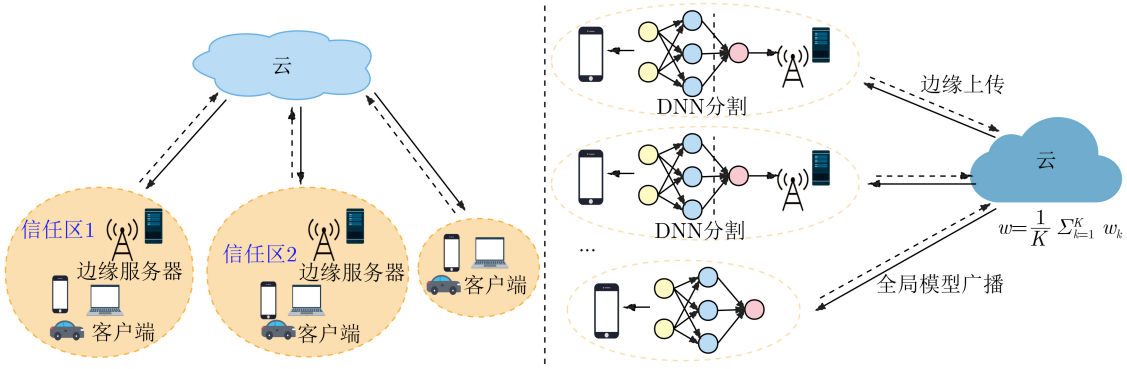

图 2 AccFed 框架

经过联邦平均算法聚合之后，再将更新后的全局模型下发至本地。左半部分为框架图，右边部分为FL细节示意图。其中，DNN Partitioning代表着DNN模型被切分，然后进行端边协同训练。

3.2 问题建模

本文根据IoT设备在FL中的真实网络场景，将资源受限情况下的端边协同计算的卸载决策分为静态决策与动态决策。其中，静态决策主要发生在网络情况良好，为了获得更好的服务体验，本文以总时延最小为优化目标。而当网络状态波动严重，一些FL的参与者往往会出现网络服务中断的情况，从而影响整个FL过程。为了提高FL效率，减少总时延，此时应将这些影响FL的时延尽可能地降到最低，从而加速FL。

3.2.1 静态网络

当IoT设备的网络状态比较稳定时，优化目标是最小化总体时延。总体时延主要由以下3个部分构成：客户端执行计算时延、客户端到边缘服务器传输时延与边缘服务器执行时延。

本文采用的AlexNet分支网络[23]有3个分支，即每种箭头颜色代表一个分支，每个分支与主干网络的交汇点为切分点。如图3所示，CONV1, CONV2, CONV5为3个切分点。每一个分支上的卷积层为候选退出点。

当IoT设备的网络状态比较稳定时，目标是最小化时延。时延主要由以下3个部分组成。

$$T = T_{\mathrm{d}} + T_{\mathrm{t}} + T_{\mathrm{e}} \tag{1}$$

其中，在IoT设备计算阶段

$$T_{\mathrm{d}} = \sum_{j=1}^{p-1} \mathrm{TD}_j \tag{2}$$

在边缘服务器计算阶段

$$T_{\mathrm{e}} = \sum_{j=p}^{N_i} \mathrm{TE}_j \tag{3}$$

传输阶段

$$T_{\mathrm{t}} = \frac{D_{\mathrm{in}} + D_{p-1}}{B} \tag{4}$$

其中，p代表切分点，B为带宽，N_i代表第i退出点所在分支包含的网络层数，D_{in}代表输入数据量。D_{p-1}代表第$p-1$层的输出数据量。TD_j与TE_j用回归模型来预测延迟[12]。因而在网络状况较好时，优化目标就是最小化T，从而找到最佳卸载方案，加速FL本地训练效率。

3.2.2 动态网络

当网络波动严重时，FL处于重度工作负载模式，此时应尽量减少最大时延(由于网络波动大，导致这些FL参与者传输时延较大，因此影响整个FL)，从而获得系统最大增益[4]。

给定B,TD_j,TE_j与D_{p-1}，通过求得切分点p与退出点ex，来最小化$\max\{T_{\mathrm{e}}, T_{\mathrm{t}}, T_{\mathrm{d}}\}$。即

$$T_{\mathrm{h}} = \frac{1}{\max\{T_{\mathrm{e}}, T_{\mathrm{t}}, T_{\mathrm{d}}\}} \tag{5}$$

其中，T_{h}定义为系统收益，为了获得最大系统收益，即优化目标为最小化$\max\{T_{\mathrm{e}}, T_{\mathrm{t}}, T_{\mathrm{d}}\}$。

网络状态波动严重的情况下，要想找到最优解属于NP困难问题。在多项式时间内找到一个全局最优解是不现实的。因而本文可以通过改变层与层之间的连接成本来进行优化。V表示DNN的各层，(v_i, v_j)表示层之间的连接。e, d表示边缘服务器与设备。V_p表示切分点所在层。轻负载的工作条件下各层之间的成本为计算的通信时延，权重系数相同，均为1。重负载的工作条件下各层之间的成本需要添加不同的权重，以此来获得最大系统收益，则修改后的成本为

$$c(v_i, v_j) = \begin{cases} \alpha T_i^{\mathrm{d}}, & v_i \in \mathcal{V}, v_j = e \\ \beta T_i^{\mathrm{t}}, & v_i \in \mathcal{V}, v_j \in \mathcal{V} \cup \mathcal{V}_p \\ \gamma T_i^{\mathrm{e}}, & v_i = d, v_j \in \mathcal{V} \\ +\infty, & \end{cases} \tag{6}$$

其中，α, β和γ是非负系数。该方法是用几个不同

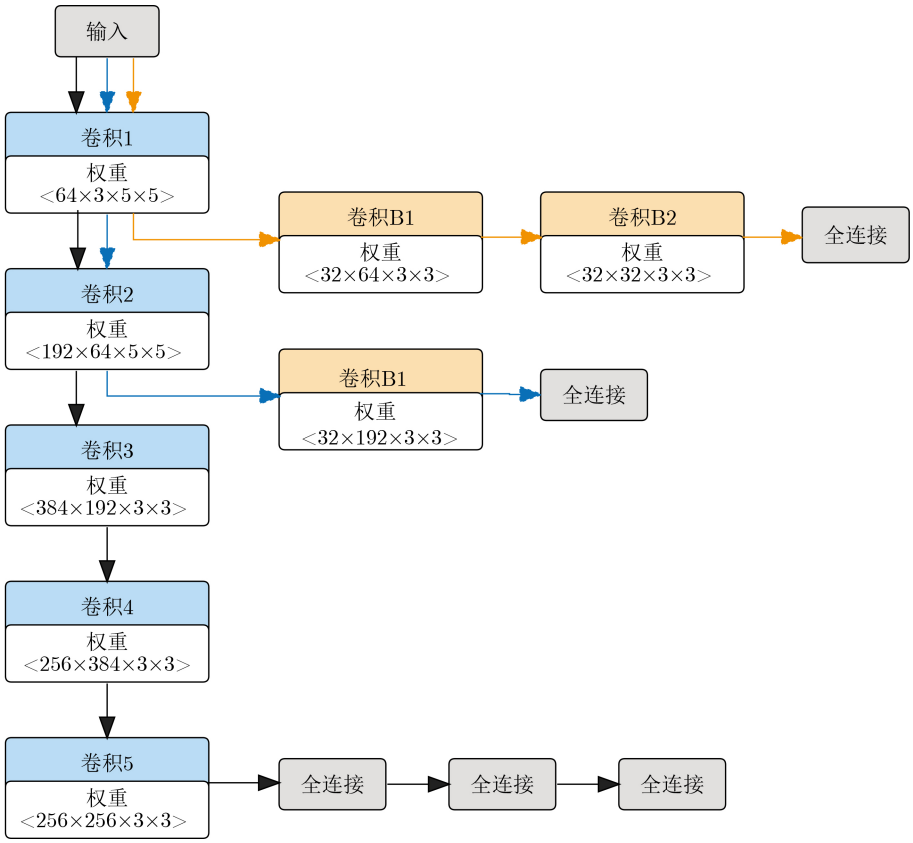

图 3 AlexNet分支网络结构示意图

的α, β和γ值来执行静态网络决策，从而最小化最大时延$T_{\max}$。通过这种方式，以优化具有特定α,β和γ元组。然后分析是否对重负载下也足够好。如果它比所有现有的解决方案更好，它就被视为重负载下的一个新的解决方案。通过固定3个参数中的1个，例如$\beta=1$，而只改变另外两个，产生一个关于α和γ的2维搜索空间。首先以粗粒度在2维平面上搜索，以找到最佳解决方案。然后，在最佳解决方案的附近使用更精细的粒度搜索来进一步改进。重复这些步骤，直到改进后的性能小于一个阈值。

4 基于端边云协同的FL加速算法

4.1 DPS算法

在DNN 切分点选择(DNN Partition Selection, DPS)算法中，使用“ping”工具，连续向边缘服务器发送两个不同大小的数据，并测量其响应时间。带宽B等于数据大小之差和响应时间之差之间的比率。由第3节已知，本文将计算卸载决策分为两种情况。即一种为网络情况稳定时，本文旨在寻找最小延迟。另一种情况是网络情况波动较为严重时，本文采用寻找系统最大收益为优化目标。根据用户输入延迟(latency)的需求，生成卸载方案。

如算法1所示：输入用户所需延迟latency，输入数据量D_{in}，分支网络拓扑(包括N_{ex},N_i)，返回提前退出点与切分点与最小时延。其中，每个分支具有的退出点数量记为N_{ex}，每个退出点具有的层数记为N_i，第i个退出点网络层记为L_j。$f(L_j)$是L_j层利用回归模型预测的运行时间。

首先，根据网络状态的不同，分为静态状态与动态网络状况；然后，根据用户输入延迟latency的需求，如果满足则根据3.2.1节进行优化，得到卸载方案(算法1的第(4)～(16)行)。否则，扩大搜索空间，进行静态卸载方案，更新$T_{\max}$，使其达到最小(算法1的第(18)～(24)行)。

4.2 端边云协同训练算法

4.2.1 本地训练

首先根据DPS算法，在满足用户延迟需求的情况下，得到最佳切分点，使得资源受限的IoT设备能够参与到FL中去，加快FL速度，提升模型性能。

为了还原真实的IoT设备FL的场景，考虑到设备的异构性，本文只针对资源受限的设备进行计算卸载，这样能力较强的终端设备就无需经过边缘服务器而直接参与训练。因此本文设计了跨信任区的联邦边缘学习模型，如AccFed框架，图2所示。其

算法1　DPS算法

输入：用户所需延迟latency，输入数据量D_{in}，分支网络拓扑(包括N_{ex},N_i)，$f(L_j)$

输出：切分点p，最小时延 T

(1) **while** true do
(2)　通过“ping”监视网络状态
(3)　**if** 需要进行计算卸载 **then**
(4)　　**if** 网络动态为静态**then**
(5)　　**for** $i=1:N_{\text{ex}}$ **do**
(6)　　选择第i个退出点
(7)　　**for** $j=1:N_i$ **do**
(8)　　$j=1:N_i\text{TE}_j \leftarrow f_{\text{e}}(L_j)$
(9)　　$\text{TD}_j \leftarrow f_{\text{d}}(L_j)$
(10)　　**end for**
(11)　　$T_{i,p}=\arg\min_p(T_{\text{d}}+T_{\text{t}}+T_{\text{e}})$
(12)　　**if** $T_{i,p}\le$latency **then**
(13)　　Return $i,p,T_{i,p}$
(14)　　**end if**
(15)　　**end for**
(16)　　Return NULL
(17)　　**else**
(18)　　$T_{\max}\leftarrow+\infty$
(19)　　**for** $\alpha=0:\dfrac{T}{\min(T_i)};\alpha\leftarrow\alpha+\sigma$ **do**
(20)　　**for** $\gamma=0:\dfrac{T}{\min(T_i)};\gamma\leftarrow\gamma+\sigma$**do**
(21)　　执行4～16行，更新$T_{\max}$
(22)　　**end for**
(23)　　若发现小于阈值，则缩小搜索空间
(24)　　**end for**
(25)　　**end if**
(26)　**end if**
(27) **end while**

算法2　Device-Edge-Cloud Synergy FL算法

输入：客户端数量N，参与者数量K，网络带宽B

输出：全局模型

(1) 从N个客户端中随机选取K个客户端进行FL
(2) 根据B，执行DPS()得到p
Procedure Device
(3) **for** each epoch **do**
(4)　**for** each batch b_i **do**
(5)　$O_p\leftarrow\text{Output}(b_i,W_{\text{d}})$
(6)　将前p层的输出O_p与激活函数发送给边
(7)　从边接收$\nabla L(O_p)$
(8)　$W_{\text{d}}\leftarrow W_{\text{d}}-\eta\cdot\nabla L(O_p)\cdot\nabla O_p(W_{\text{d}})$
(9)　将W_{d}的变化进行参数裁剪
(10)　**end for**
(11)　计算W_{d}平均变化量$\delta_{W_{\text{d}}}$，如果$\delta_{W_{\text{d}}}$变小，则增加本地迭代次数
Procedure Edge
(12) 从云获取最新全局模型W_{c}
(13) $W_{\text{e}}\leftarrow W_{\text{c}}$
(14) **while** true **do**
(15)　从设备接收O_p与激活函数
(16)　$W_{\text{e}}\leftarrow W_{\text{e}}-\eta\cdot\nabla L(W_{\text{e}})$
(17)　将$\nabla L(O_p)$发给设备
(18) **end while**
Procedure Cloud
(19) 初始化W_{c}
(20) **for** each round **do**
(21)　将W_{c}发送给边
(22)　从设备接收W_{d}
(23)　执行联邦平均算法更新W_{c}
(24)　对W_{c}进行裁剪，求取高斯噪声方差σ
(25)　$W_{\text{c}}\leftarrow W_{\text{c}}+N(0,\sigma^2)$
(26) **end for**

中，客户端和边缘服务器的协同计算如算法2所示。其中，$W_{\text{d}},W_{\text{e}},W_{\text{c}}$分别代表IoT设备、边缘服务器和云相应的权重值。$\eta$为学习率，$L$为损失函数。

首先运行DPS算法，得到最佳切分点p，然后在设备本地执行DNN前p层，并将得到的模型参数发送给同一个信任区内的边缘服务器，边缘服务器执行DNN后P层，并将结果返给设备；然后，设备接收边缘服务器返回的结果，继续执行梯度后向传播，完成本轮训练。最后将本地训练的权重参数上传给云，并计算每一轮训练的权重变化量。如果变化量减少，则增加本地训练迭代次数，从而减少上传间隔，提高通信效率。整个执行步骤如算法2的第(1)～(18)行所示。

4.2.2　模型聚合

由于跨信任区之间进行联邦学习，IoT设备具有异构性，在模型聚合时需要考虑两种情形。一种是资源受限的设备，需要进行端边协同训练；另一种是计算能力较强的设备，无需进行计算卸载，直接进行本地计算，然后上传模型参数。

本文通过FedAvg算法来进行全局模型更新

$$G^{t+1}=G^t+\frac{1}{K}\sum_{i=1}^{m}(L_i^{t+1}-G_i^t)\tag{7}$$

其中，G^t表示第t轮聚合之后的全局模型，L_i^{t+1}表示第i个客户端在$t+1$轮本地更新后的模型，G^{t+1}表示第$t+1$轮聚合之后的全局模型。

为了保护数据隐私安全，本文使用了这DP-FedAvg[24,25]算法。该算法是将联邦学习中经典的FedAvg[8]算法与差分隐私[26]训练相结合，并将其应用在语言模型的预测上，取得了不错的效果。IoT设备与边缘服务器融合得到的模型最终经过差分隐私的方式上传到云端，保障数据的隐私性。

DP-FedAvg在服务器侧的基本流程主要包括：

(1) 随机选取参与训练的客户端集K_t。

(2) 对挑选的客户端$k \in K_t$，执行本地训练。

(3) 服务端接受每一个客户端k的模型参数W_k^t，执行聚合操作，得到W^t。

(4) 求取高斯噪声分布的方差σ，利用高斯分布$N(0, I\sigma^2)$生成噪声数据。

(5) 在全局模型聚合操作中添加噪声数据，得到新的全局模型参数θ_t。

(6) 重复上述步骤，直至模型收敛。

即关键操作步骤

$$\theta_t = \theta_{t-1} + W^t + N(0, I\sigma^2) \tag{8}$$

模型聚合算法步骤：首先执行全局模型参数初始化；然后接收设备发来的模型参数，并行执行；最后通过DP-FedAvg更新全局模型，并下发给设备。模型聚合的步骤如算法2的第(19)～(26)行所示。

5 实验搭建与结果分析

5.1 实验搭建

首先，本文采用Python3.7, PyTorch1.4.0，基于开源FL框架PySyft部署多客户端场景，数据集采用CIFAR-10，并且对数据随机做了切分，保证每个IoT设备拥有数据的唯一性。采用经典的CNN中的AlexNet，并对其进行改造，生成分支AlexNet。然后，本文搭建了AccFed原型，原型系统主要包括5台树莓派、2台NVIDIA Jetson Xavier NX和1台服务器。其中有3台树莓派能力较弱，当作资源受限的IoT设备；NVIDIA Jetson Xavier NX能力较强，具有很强的Edge AI处理能力，用来当作边缘服务器；服务器性能最好，用来当作云。原型系统配置如表2所示。

5.2 性能评估

为了验证本文所提AccFed的优势，将其与文献[20]中的SplitFed算法和文献[8]中的FedAvg算法做比较。首先分析FedAvg, SplitFed与AccFed随着客户端数量增多时，学习性能的改变。其中，学习性能指标包括最终的模型精度、训练的耗时，模型收敛的速度。

在图4—图6中，对FedAvg, SplitFed与AccFed进行了模型训练精度的对比，其中k为客户端的数量。如图4所示，当$k=3$时，AccFed收敛速度最快，在迭代次数为49时，达到了最高精度76.69%；SplitFed收敛速度次之，但是最高精度仅为69.66%；而FedAvg在迭代次数为150时才逐渐收敛。如图5所示，当$k=5$时，AccFed在迭代次数为42时，达到了最高精度80.79%；SplitFed有与$k=3$时相似的趋势，最终精度达73.46%；FedAvg将在第130次迭代时逐渐收敛。如图6所示，当$k=7$时，AccFed在迭代次数为35时，达到了最高精度81.66%；SplitFed则是在第41次迭代时逐渐收敛到74.24%；FedAvg在迭代次数为100时趋于收敛。

综上所述，AccFed在模型收敛速度是最快的，并且训练精度是最好的。此外，当客户端数量增多时，三者的模型精度均有不同程度的提升。由此可以看出我们的AccFed算法的高效性。

在图7中，分析了当客户端数量逐渐增加时，AccFed收敛速度的变化。当客户端数量$k=5$或者$k=7$时，迭代次数达到25轮后，AccFed已经接近

表 2 各设备参数表

设备	内存(GB)	数量	计算能力
树莓派 3B+	1	3	较弱
树莓派 4B	8	2	一般
Jetson Xavier NX	16	2	较强
服务器	32	1	最强

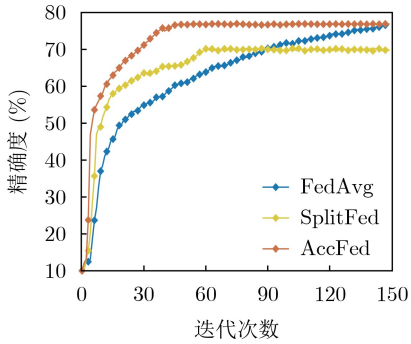

图 4 当$k=3$, FedAvg, SplitFed与AccFed的训练精度

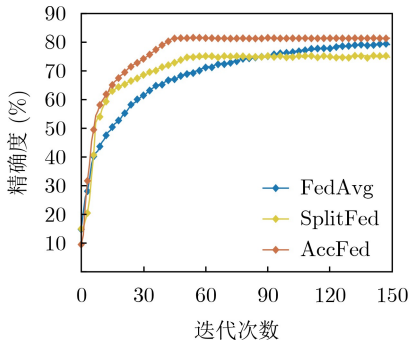

图 5 当$k=5$, FedAvg, SplitFed与AccFed的模型精度

收敛时的精度，当迭代次数达到50轮后，均已达到收敛时的精度。客户端数量$k=3$时，从第10轮～25轮之间精确度涨幅明显，到了50轮后，也已经达到收敛时准确度。总体而言，所有的客户端在迭代次数已经完成了收敛，对比FedAvg至少需要120轮收敛，收敛速度更快。

训练用时如图8所示。随着客户端数量的上升，相应的训练时间也有所增加。但是相同数量的客户端参与训练时，AccFed的平均训练用时大约只有FedAvg 的1/2。这充分说明了AccFed在计算卸载时的高效性。下面将通过损失值的收敛性，进一步阐明AccFed的优越性。

如图9所示，当$k=3$时，AccFed迭代50轮之前，损失值达到了最小值；SplitFed的收敛速度次之，最终损失值为0.7774；而FedAvg需要接近150轮时才能收敛。同理，如图10所示，当$k=5$时，AccFed也是领先FedAvg 45轮迭代之前，损失值达到了最小值；SplitFed有与$k=3$时相似的趋势；而FedAvg在120轮之后才达到收敛。AccFed对比FedAvg，以近似3倍的速度到达收敛，这充分说明了AccFed算法的优越性。

如图11所示，当$k=7$时，AccFed在迭代次数为35时，损失值达到了最小值0.612，SplitFed在第41轮时逐渐收敛，而FedAvg需要接近120轮时才能收敛。综上所述，AccFed能够保证最终模型精度与FedAvg基本一致，同时收敛速度达到FedAvg的3倍之多，每轮耗时却只有FedAvg的1/2。由于DPS算法与端边云训练算法的高效性，不仅能使资源受限的IoT设备参与训练，提高FL训练与通信效率，还能保证模型安全性传输。同时，在模型精

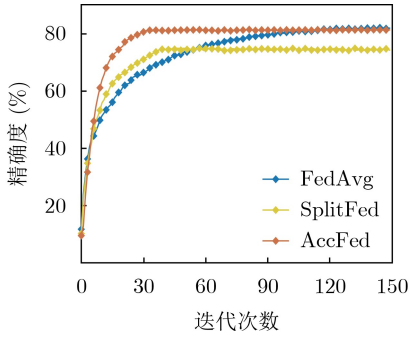

图 6 $k=7$, FedAvg, SplitFed与AccFed的模型精度

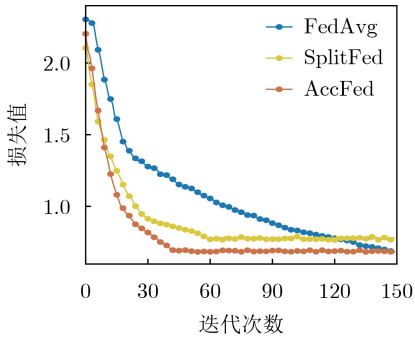

图 9 $k=3$, FedAvg, SplitFed与AccFed的损失值对比

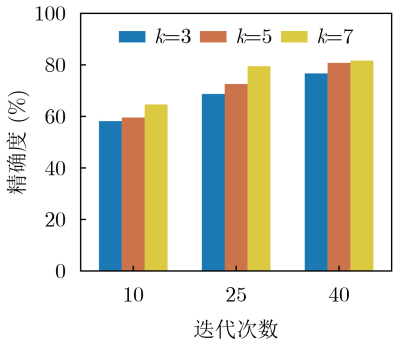

图 7 AccFed 50轮迭代之前的模型精度

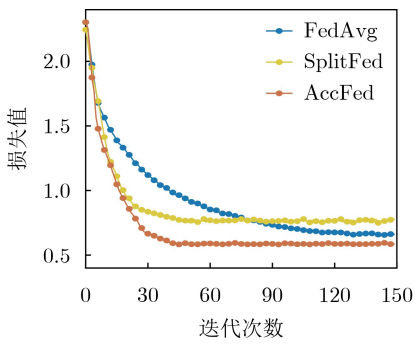

图 10 $k=5$, FedAvg, SplitFed与AccFed的损失值对比

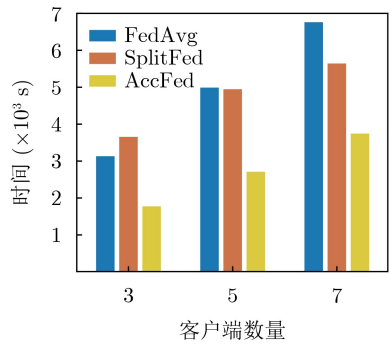

图 8 当迭代次数为150轮时，FedAvg, SplitFed与AccFed的训练用时

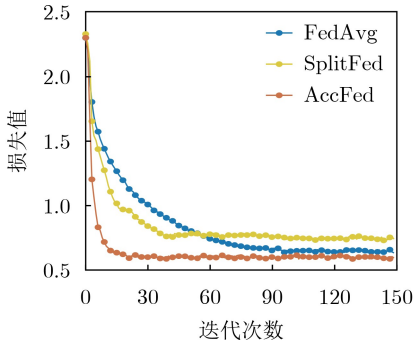

图 11 $k=7$, FedAvg, SplitFed与AccFed的损失值对比

度、收敛速度和平均每轮消耗时间等指标中，也充分说明了AccFed的有效性与高效性。

6 结论

在IoT场景下，Edge AI成为最近研究的热点。但IoT设备通信与计算资源有限，并且极易遇到数据泄露或者被篡改的风险。FL作为一种新兴的隐私保护框架，在数据安全、提升模型性能等方面具有巨大潜力，但是通信及本地训练效率低。为了解决上述难题，本文提出一种FL加速框架AccFed。首先，根据网络状态的不同，提出一种基于模型分割的端边云协同训练算法，加速FL本地训练；然后，设计一种多轮迭代再聚合的模型聚合算法，加速FL聚合；最后，通过分析IoT设备的数据特征，本文使用CIFAR-10数据集，采用分支深度神经网络对数据进行训练，在树莓派、NVIDIA Jetson Xavier NX和服务器上实现了AccFed原型。实验结果表明，与FedAvg以及SplitFed相比，AccFed在训练精度、收敛速度、训练时间等方面均优于对照组。AccFed能够保证最终模型训练精度与FedAvg基本一致，同时收敛速度达到FedAvg的3倍之多，平均每轮耗时却只有FedAvg的1/2，这充分说明了AccFed的有效性与高效性。

参 考 文 献

[1] AAZAM M, ZEADALLY S, and HARRAS K A. Deploying fog computing in industrial internet of things and industry 4.0[J]. *IEEE Transactions on Industrial Informatics*, 2018, 14(10): 4674–4682. doi: 10.1109/TII.2018.2855198.

[2] UR REHMAN M H, AHMED E, YAQOOB I, *et al*. Big data analytics in industrial IoT using a concentric computing model[J]. *IEEE Communications Magazine*, 2018, 56(2): 37–43. doi: 10.1109/MCOM.2018.1700632.

[3] SHI Weisong, CAO Jie, ZHANG Quan, *et al*. Edge computing: vision and challenges[J]. *IEEE Internet of Things Journal*, 2016, 3(5): 637–646. doi: 10.1109/JIOT.2016.2579198.

[4] MOHAMMED T, JOE-WONG C, BABBAR R, *et al*. Distributed inference acceleration with adaptive DNN partitioning and offloading[C]. Proceedings of 2020 IEEE Conference on Computer Communications, Toronto, Canada, 2020: 854–863. doi: 10.1109/INFOCOM41043.2020.9155237.

[5] ZHANG Peiying, WANG Chao, JIANG Chunxiao, *et al*. Deep reinforcement learning assisted federated learning algorithm for data management of IIoT[J]. *IEEE Transactions on Industrial Informatics*, 2021, 17(12): 8475–8484. doi: 10.1109/TII.2021.3064351.

[6] GAO Yansong, KIM M, ABUADBBA S, *et al*. End-to-end evaluation of federated learning and split learning for internet of things[C]. Proceedings of 2020 International Symposium on Reliable Distributed Systems (SRDS), Shanghai, China, 2020. doi: 10.1109/SRDS51746.2020.00017.

[7] YU Keping, TAN Liang, ALOQAILY M, *et al*. Blockchain-enhanced data sharing with traceable and direct revocation in IIoT[J]. *IEEE Transactions on Industrial Informatics*, 2021, 17(11): 7669–7678. doi: 10.1109/TII.2021.3049141.

[8] MCMAHAN B, MOORE E, RAMAGE D, *et al*. Communication-efficient learning of deep networks from decentralized data[C]. Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, USA, 2017: 1273–1282.

[9] GUO Yeting, LIU Fang, CAI Zhiping, *et al*. FEEL: A federated edge learning system for efficient and privacy-preserving mobile healthcare[C]. Proceedings of the 49th International Conference on Parallel Processing-ICPP. Edmonton, Canada, 2020: 9. doi: 10.1145/3404397.3404410.

[10] CAO Xiaowen, ZHU Guangxu, XU Jie, *et al*. Optimized power control for over-the-air federated edge learning[C]. ICC 2021-IEEE International Conference on Communications, Montreal, Canada, 2021: 1–6. doi: 10.1109/ICC42927.2021.9500664.

[11] LO S K, LU Qinghua, WANG Chen, *et al*. A systematic literature review on federated machine learning: From a software engineering perspective[J]. *ACM Computing Surveys*, 2022, 54(5): 95. doi: 10.1145/3450288.

[12] LI En, ZHOU Zhi, and CHEN Xu. Edge intelligence: On-demand deep learning model co-inference with device-edge synergy[C]. Proceedings of 2018 Workshop on Mobile Edge Communications, Budapest, Hungary, 2018: 31–36. doi: 10.1145/3229556.3229562.

[13] 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. doi: 10.1145/3093337.3037698.

[14] ESHRATIFAR A E, ABRISHAMI M S, and PEDRAM M. JointDNN: an efficient training and inference engine for intelligent mobile cloud computing services[J]. *IEEE Transactions on Mobile Computing*, 2021, 20(2): 565–576. doi: 10.1109/TMC.2019.2947893.

[15] TANG Xin, CHEN Xu, ZENG Liekang, *et al*. Joint multiuser DNN partitioning and computational resource allocation for collaborative edge intelligence[J]. *IEEE Internet of Things Journal*, 2021, 8(12): 9511–9522. doi: 10.1109/JIOT.2020.3010258.

[16] LI En, ZENG Liekang, ZHOU Zhi, *et al*. Edge AI: On-demand accelerating deep neural network inference via edge

computing[J]. *IEEE Transactions on Wireless Communications*, 2020, 19(1): 447–457. doi: 10.1109/TWC.2019.2946140.

[17] ELGAMAL T and NAHRSTEDT K. Serdab: An IoT framework for partitioning neural networks computation across multiple enclaves[C]. Proceedings of the 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID), Melbourne, Australia, 2020: 519–528. doi: 10.1109/CCGrid49817.2020.00-41.

[18] ZHU Guangxu, DU Yuqing, GÜNDÜZ D, *et al*. One-bit over-the-air aggregation for communication-efficient federated edge learning: Design and convergence analysis[J]. *IEEE Transactions on Wireless Communications*, 2021, 20(3): 2120–2135. doi: 10.1109/TWC.2020.3039309.

[19] DU Yuqing, YANG Sheng, and HUANG Kaibin. High-dimensional stochastic gradient quantization for communication-efficient edge learning[J]. *IEEE Transactions on Signal Processing*, 2020, 68: 2128–2142. doi: 10.1109/TSP.2020.2983166.

[20] THAPA C, CHAMIKARA M A P, CAMTEPE S, *et al*. Splitfed: When federated learning meets split learning[J]. arXiv: 2004.12088, 2020. doi: 10.48550/arXiv.2004.12088.

[21] VEPAKOMMA P, GUPTA O, SWEDISH T, *et al*. Split learning for health: Distributed deep learning without sharing raw patient data[J]. arXiv: 1812.00564, 2018. doi: 10.48550/arXiv.1812.00564.

[22] ROMANINI D, HALL A J, PAPADOPOULOS P, *et al*. PyVertical: A vertical federated learning framework for multi-headed SplitNN[J]. arXiv: 2104.00489, 2021. doi: 10.48550/arXiv.2104.00489.

[23] TEERAPITTAYANON S, MCDANEL B, and KUNG H T. Branchynet: Fast inference via early exiting from deep neural networks[C]. Proceedings of the 23rd International Conference on Pattern Recognition (ICPR), Cancun, Mexico, 2016: 2464–2469. doi: 10.1109/ICPR.2016.7900006.

[24] MCMAHAN H B, ANDREW G, ERLINGSSON U, *et al*. A general approach to adding differential privacy to iterative training procedures[J]. arXiv: 1812.06210, 2018. doi: 10.48550/arXiv.1812.06210.

[25] MCMAHAN H B, RAMAGE D, TALWAR K, *et al*. Learning differentially private language models without losing accuracy[J]. arXiv: 1710.06963, 2018. doi: 10.48550/arXiv.1710.06963.

[26] ABADI M, CHU A, GOODFELLOW I, *et al*. Deep learning with differential privacy[C]. Proceedings of 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 2016: 308–318. doi: 10.1145/2976749.2978318.

曹绍华：男，副教授，硕士生导师，研究方向为SDN、云计算和边缘计算等.

陈　辉：男，硕士生，研究方向为边缘智能、联邦学习和SDN等.

陈　舒：女，硕士生，研究方向为智能城市和5G等.

张汉卿：男，硕士生，研究方向为边缘计算中的计算卸载和数据缓存等.

张卫山：男，教授，博士生导师，研究方向为大数据平台、普适性云计算、面向服务计算和联邦学习等.

责任编辑：余　蓉