

边缘场景下动态联邦学习优化方法

王志良¹, 何刚^{1,2*}, 俞文心¹, 许康¹, 文军¹, 刘畅¹

(1. 西南科技大学 计算机科学与技术学院, 四川 绵阳 621010;

2. 国家卫生健康委核技术医学转化实验室(绵阳市中心医院), 四川 绵阳 621010)

摘要:边缘计算(Edge Computing)是一种新的计算方式,通过在网络边缘提供计算服务,与传统的云计算模式相比,具有高可信度和低延迟等特点。联邦学习(FL)作为一种分布式机器学习方法,尽管具备保护隐私和数据安全的特性,却仍然面临设备异构和数据不均衡等问题,导致出现部分参与者(边缘端)训练时间长、训练效率低下等问题。为了解决上述问题,该文提出了一种名为FlexFL的动态联邦学习优化算法。该算法引入了两层联邦学习策略,通过在同一参与者部署多个联邦学习训练服务和一个联邦学习聚合服务,将本地数据集平均分配给各个联邦学习训练服务,并每回合激活一定数量的训练服务。未激活的服务将休眠,不占用计算资源,并将资源平均分配给激活的服务,以加快训练速度。该算法能够平衡参与者设备异构和数据不均衡性带来的训练时间差异,从而提高整体训练效率。在MINST数据集和CIFAR数据集上与FedAvg联邦学习算法进行了对比实验,结果显示,FlexFL算法在减少时间消耗的同时,不降低模型性能。

关键词:边缘计算;机器学习;联邦学习;服务动态缩放;数据不均衡;设备异构

中图分类号:TP181

文献标识码:A

文章编号:1673-629X(2024)02-0098-07

doi:10.3969/j.issn.1673-629X.2024.02.015

Dynamic Federated Learning Optimization Method in Edge Scenarios

WANG Zhi-liang¹, HE Gang^{1,2*}, YU Wen-xin¹, XU Kang¹, WEN Jun¹, LIU Chang¹

(1. School of Computer Science and Technology, Southwest University of Science and Technology,
Mianyang 621010, China;

2. National Health Commission Laboratory for Translational Medicine of Nuclear Technology
(Mianyang Central Hospital), Mianyang 621010, China)

Abstract: Edge computing is a new computing paradigm that provides computational services at the network edge. Compared to traditional cloud computing, edge computing offers advantages such as high reliability and low latency. However, federated learning (FL), a distributed machine learning method, still faces challenges related to device heterogeneity and data imbalance, leading to issues like prolonged training time and low training efficiency for certain participants (edge devices). To address these challenges, we propose a dynamic federated learning optimization algorithm called FlexFL. The algorithm introduces a two-tier federated learning strategy by deploying multiple federated learning training services and a federated learning aggregation service on the same edge device. It evenly partitions the local dataset among the federated learning training services and activates a certain number of training services per round. Inactive services go into a dormant state, freeing up computing resources and redistributing them evenly among the active services to accelerate training. The algorithm balances the discrepancies in training time caused by device heterogeneity and data imbalance, thereby improving overall training efficiency. Experimental comparisons between the FlexFL algorithm and the FedAvg federated learning algorithm were conducted on the MINST dataset and CIFAR dataset, and the results demonstrate that FlexFL reduces time consumption without compromising model performance.

Key words: edge computing; machine learning; federated learning; service dynamic scaling; data imbalance; device heterogeneity

0 引言

随着物联网和智慧城市等应用的快速发展,越来越多的设备(如传感器、摄像头和智能手机等)加入到

无线网络中。然而,云计算已经无法满足海量边缘数据爆炸式增长^[1-2]。为了解决这一问题,边缘计算^[3]作为一种新的计算形式出现,通过在网络边缘的边缘

收稿日期:2023-05-20

修回日期:2023-09-21

基金项目:四川省科技项目(2020YFS0454, 2020YFS0318);国家卫生健康委核技术医学转化重点实验室开放课题资助(2021HYX031)

作者简介:王志良(1998-),男,硕士研究生,从事机器学习方面的研究工作;通信作者:何刚(1984-),男,副教授,博士,研究方向为深度学习、医学图像分析与处理、医疗仪器研发。

端服务器上处理数据,避免了将数据发送到远程云端所需的长传播延迟。然而,由于通信资源和数据隐私安全等限制,将用户数据发送到云端的方法通常被认为不切实际。因此,Google提出了一种分布式的机器学习方法,称为联邦学习(Federated Learning, FL)^[4-5]。在传统的集中式机器学习模型中,通信成本相对较小,计算成本占主导地位。然而,在联邦学习中,通信成本占主体地位。综合以往的研究,可以总结出以下三个挑战:

- (1)通信问题:服务器与客户端之间的网络连接相对较慢。
- (2)设备异构:不同参与者的计算资源可能不同。
- (3)数据不均衡:不同客户端之间的数据分布不一致^[6]。

为了解决通信问题,已有研究者提出了模型压缩和模型蒸馏来降低链路通信成本^[7-8]。许多文献着重于处理数据特征分布不均衡,以减少通信回合来提高联邦学习的测试准确率^[9-11],但每个回合的计算时间也很重要。针对数据类别不平衡,可以修改训练数据或调整学习策略^[12]。而样本总数在参与者之间分布不均衡会影响每回合联邦学习的时间消耗。

该文重点解决每回合时间消耗大导致整体训练效率低的问题。联邦学习中,每回合的时间消耗由耗时最长的参与者决定,而参与者的时间消耗差异通常由设备异构和数据不均衡引起。已有国内外研究关注如何缩短每回合的通信时间,其中基于博弈论的优化研究较多^[13-15]。Sarikaya等人^[15]基于斯塔克伯格博弈模型提出了一项研究,在预算有限的情况下,通过权衡参与者多样性和完成训练的延迟来选择参与者。然而,这种选择方式使得某些参与者无法参与联邦学习,不公平。贾云健^[16]提出了一种基于博弈论的激励机制,在分层联邦学习框架下考虑了连接终端设备数量差异。该机制设计了两层主从博弈,分配激励预算,刺激终端设备更积极参与联邦学习任务,减小“掉队效应”,最小化全局模型训练时间。然而,仍存在不公平行为和某些参与者需要贡献更多计算资源的问题。

综上所述,设计一种方法以充分考虑每回合中参与者的设备异构和数据不均衡,缩短每回合时间消耗并确保公平性变得重要。为此,该文提出了FlexFL算法,旨在缩短联邦学习每回合的时间消耗。主要贡献如下:

(1)引入联邦学习训练服务动态缩放策略,通过两层联邦学习架构,在同一参与者上部署多个联邦学习训练服务和一个联邦学习聚合服务。通过动态缩放服务,加快计算速度。

(2)提出FlexFL算法,将本地数据集平均分配给

各联邦学习训练服务,并每回合激活一定数量的训练服务。未激活的服务休眠,不占用计算资源,并将计算资源平均分配给激活的服务,加快训练速度。

通过以上贡献,FlexFL算法能有效解决参与者设备异构和数据不均衡导致的训练时间差异问题,提高整体训练效率。

1 相关工作

1.1 服务动态缩放

服务动态缩放常常在互联网高并发架构中使用,其本质是通过合理配置资源以及增加或缩减服务来提高服务吞吐量和降低服务延迟^[17-19]。目前,尚未有研究者考虑将服务动态缩放与联邦学习相结合。该文旨在将服务动态缩放与联邦学习相结合,以平衡每回合参与者在设备异构和数据不均衡性带来的时间消耗差异。

1.2 联邦学习算法

时间消耗的优化一直是FL的主要研究方向之一。McMahan等人^[20]提出了一种名为FedAvg(Federated Averaging)的联邦学习算法,该算法通过对各参与方的局部模型进行平均来更新全局模型,并通过增加聚合期间局部模型的训练次数来减少通信开销。然而,该算法忽略了减少每回合时间消耗的问题。

田有亮^[21]提出了一种基于激励机制的联邦学习优化算法,利用博弈论和拍卖理论设计了拍卖机制。参与者通过向雾节点拍卖本地训练任务,并委托高性能雾节点训练本地数据,以提升本地训练效率并解决客户端间性能不均衡的问题。然而,这种方案破坏了联邦学习的隐私性,因为本地数据被发送到其他节点。程帆^[22]提出了一种基于动态权重的联邦学习优化方法。在极端情况下,计算时间消耗较大的参与者可以选择不参与聚合。尽管这种方案可以减少每回合的时间消耗,但对于部分计算时间消耗较大的参与者来说存在不公平性的问题。现有的研究常常通过剔除训练延迟大的参与者来提高效率,但这对某些参与者不公平。为解决这个问题,该文提出了FlexFL算法,能够平衡参与者设备异构和数据不均衡性对训练时间的影响,提高整体训练效率。实验结果表明,FlexFL算法在减少时间消耗的同时,不降低模型性能。

2 动态联邦学习优化方法

在本章中,首先提出了联邦学习训练服务动态缩放策略,并引入了两层联邦学习架构来实现该策略。随后,基于联邦学习训练服务动态策略,提出了FlexFL算法,旨在降低联邦学习每回合的时间消耗。

使用的符号解释如表1所示。

表 1 符号解释

符号	含义
m	参与者标号
c	参与者中联邦学习训练服务标号
D	参与者本地数据
k	通信回合数目
E	本地训练轮数
a	激活的联邦学习训练服务数目
v	参与者的计算资源
\tilde{t}	参与者本回合的预计时间消耗
φ_1	神经网络前向传播时间
φ_2	神经网络后向传播时间

2.1 联邦学习训练服务动态缩放策略

针对联邦学习中每回合不同参与者由于时间消耗不同而导致全局模型聚合时间长,从而影响整体联邦学习训练效率的问题,重点考虑设备异构和数据不均衡这两个因素,因为它们在实际环境中是主要影响参与者时间消耗的因素。基于实际环境的考虑,提出以下设定:

设定一:参与者的设备异构主要通过模拟计算资源的不同来实现,而数据不均衡主要通过模拟数据分布的不均衡来实现。

设定二:参与者在训练开始之前能够收集到自身的数据量和计算资源的情况,并且对中央服务器具有信任,能够将这些信息上传至中央服务器。

设定三:参与者的数据量不均衡,并且在联邦学习过程中不再发生变化,即不会增加或减少本地数据量。

设定四:参与者的计算资源因复杂的环境而处于动态变化中,例如参与了其他计算任务,占用了计算资源。

基于以上设定,并结合服务动态缩放的思想^[17],提出了联邦学习训练服务动态缩放策略。该策略在每个参与者上部署了多个联邦学习训练服务和一个联邦学习模型聚合服务。首先,将参与者的本地数据平均划分给对应的联邦学习训练服务进行训练,得到本地模型。然后,参与者的模型聚合服务将本地模型聚合后上传到中央服务器进行进一步的模型聚合。中央服务器完成模型聚合后,将更新后的模型发送回参与者的联邦学习训练服务,进行下一回合的训练。在每回合训练中,根据参与者计算资源的变化,随机选择激活参与者部署的联邦学习训练服务参与下一轮的训练,未被选中的服务将休眠并释放计算资源,而被选中的服务将获得分配的计算资源,从而加快计算速度。图 1 展示了该策略的结构。

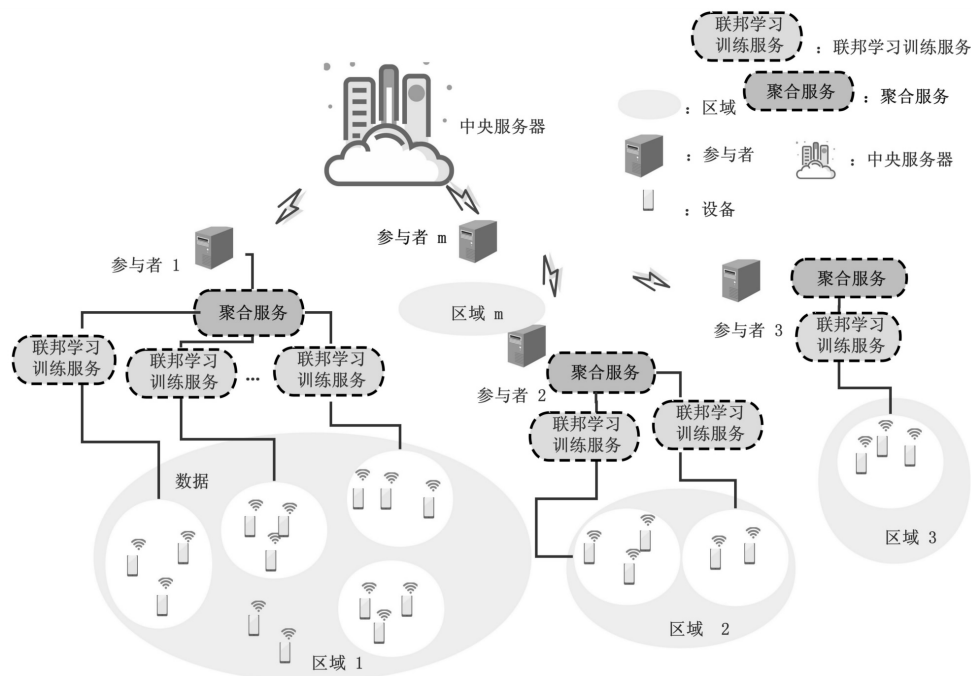


图 1 联邦学习训练服务动态缩放策略

如图 1 所示, m 个参与者在中央服务器的协助下共同训练一个全局模型。在该设定中,参与者的本地数据是非独立同分布(Non-IID)且不均衡的,这符合大多数真实场景的情况。参与者的本地数据集用集合 $\{D_1, D_2, \dots, D_m\}$ 表示,参与者的联邦学习训练服务

数目 c_m 为:

$$c_m = \frac{D_m}{\min\{D_1, D_2, \dots, D_m\}} \quad (1)$$

然后,参与者 m 将本地数据集随机平均划分成 c_m 份,对应 c_m 个联邦学习训练服务。接着,每回合重复

以下步骤直至模型收敛。

步骤一:用集合 $\{v_1^k, v_2^k, \dots, v_m^k\}$ 表示第 k 回合参与者的计算资源。参与者的预计时间消耗用集合 $\{\tilde{t}_1^k, \tilde{t}_2^k, \dots, \tilde{t}_m^k\}$ 表示。因此:

$$\tilde{t}_m^k = \frac{D_m}{v_m^k} \quad (2)$$

步骤二:确定参与者选择激活的参与者的数目为:

$$a = c_m \frac{\min\{\tilde{t}_1^k, \tilde{t}_2^k, \dots, \tilde{t}_m^k\}}{\tilde{t}_m^k} \quad (3)$$

步骤三:随机激活 a 个联邦学习训练服务,并将计算资源平均分配给这些激活的联邦学习训练服务,使其参与本回合的计算。未被激活的服务不进行计算,不占用计算资源。

表2 通信开销和计算开销

开销	FedAvg	FlexFL
中央服务器的聚合计算开销	$O(mn)$	$O(mn)$
参与者的聚合计算开销	无	$O(\alpha n)$
中央服务器的通信开销	$O(m)$	$O(m)$
参与者的通信开销	$O(1)$	$O(1)$
参与者的训练计算开销	$O(E(\varphi_1 + \varphi_2))$	$O(\frac{a}{c}E(\varphi_1 + \varphi_2))$

表2展示了每一回合中两种算法的计算开销对比。传统的 FedAvg 算法中,中央服务器基于参与者上传的 m 个模型计算全局模型,每个模型包含 n 个参数,对于每个参数计算其平均值,因此时间复杂度为 $O(mn)$ 。提出的 FlexFL 算法的时间复杂度也是 $O(mn)$ 。传统的 FedAvg 算法中,参与者基于本地数据优化模型,本地训练轮数为 E ,每一轮中,通过神经网络优化算法更新模型参数,包括一次前向传播和反向传播,因此时间复杂度为 $O(E(\varphi_1 + \varphi_2))$ 。所提的 FlexFL 算法在此基础上,每回合只激活 a 个联邦学习训练服务参与计算,即 $O(\frac{a}{c}E(\varphi_1 + \varphi_2))$ 。表2中还

展示了两种算法的通信开销。对于传统的 FedAvg 算法,服务器需要与 m 个参与者通信,因此其通信开销为 $O(m)$,每个参与者仅需与服务器通信,因此其通信开销为 $O(1)$,所提的 FlexFL 算法也是如此。

2.2 动态联邦学习训练算法 FlexFL

在参与者的训练中,如图1所示,每回合包括每个参与者中 c 个联邦学习训练服务和相应聚合服务之间的训练,以及 m 个参与者和中央服务器聚合器间的全局聚合。第 k 回合训练时,中央服务器将向多个参与者服务器发送初始模型参数 ω^{k-1} ,每个联邦学习训练服务将从对应的参与者获取这些参数。接下来,在接收到 ω^{k-1} 后,联邦学习训练服务使用其局部数据集来更新局部模型 ω_c^k 。模型参数的计算如下:

$$g = \nabla F(\omega_c^{k-1}) \quad (4)$$

$$\omega_c^k \leftarrow \omega_c^{k-1} + \mu g \quad (5)$$

其中, ∇F 是损失函数, g 是梯度, ω 表示回合 k 中的模型参数, μ 表示学习率, c 是联邦学习训练服务标号。

在联邦学习训练服务上计算更新值后,更新后的模型参数将传到参与者的聚合服务。聚合服务将模型参数 ω_c^k 进行加权平均计算,得到更新参数 ω_m^k , m 表示参与者标号:

$$\omega_m^k \leftarrow \sum_{c=1}^c \frac{n_c}{n} \omega_c^k \quad (6)$$

参与者 m 将更新后的模型参数 ω_m^k 发送到中央服务器进行加权平均计算,得到全局聚合的更新参数 ω_k ,返回参与者进行 $k+1$ 回合的训练。

$$\omega_k \leftarrow \sum_{m=1}^m \frac{n_m}{n} \omega_m^k \quad (7)$$

如图2所示,在开始训练之前,参与者需进行本地数据划分以对应联邦学习训练服务。中央服务器需要进行一些初始化工作,包括选择算法,超参数,初始化全局模型等,然后重复下述步骤直至模型收敛。

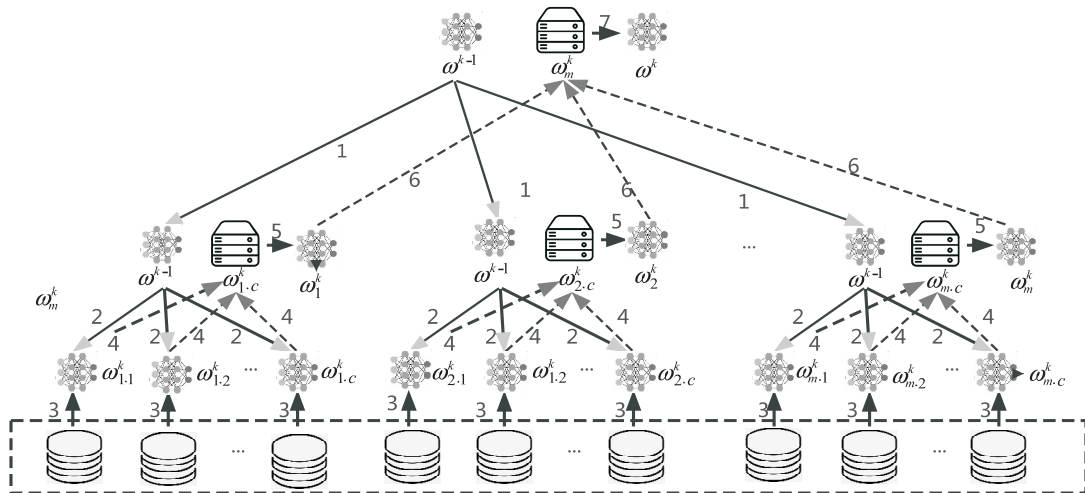


图2 FlexFL 训练架构

第1步:训练开始,中央服务器下发本轮全局模型给参与者。

第2步:参与者接收到中央服务器的全局模型 ω_{k-1} ,并将其发送给联邦学习训练服务。

第3步:联邦学习训练服务利用其本地数据进行训练,得到更新后的模型 ω_c^k 。

第4步:联邦学习训练服务将本地模型 ω_c^k 发送给对应的参与者的聚合服务。

第5步:参与者的聚合服务将分布在该参与者的联邦学习训练服务产生的本地模型进行初次聚合得到参与者的本地模型 ω_m^k 。

第6步:参与者将其本地模型发送到中央服务器。

第7步:中央服务器收到所有本轮参与训练的参与者的本地模型后,聚合更新后得到全局模型 ω^k 。

第8步:返回第1步,直至模型收敛。

3 实验结果及分析

3.1 数据集与模型

为了验证提出的 FlexFL 算法的性能,设计了一个由 200 个参与者和 1 个中央服务器组成的网络。每个回合中央服务器随机选择 20% 的参与者参与训练。

在深度学习中使用两个常见的基准数据集来实现算法。第一个数据集是 MNIST 手写数字识别,包含 60 000 个训练样本和 10 000 个测试样本。每张图片代表 0 到 9 中的一个数字,位于中央。为了每个参与者被随机分配训练样本,并且反映出联邦学习数据非独立同分布的特点,数据集划分采用了非平衡方式,首先按数字标签进行随机打乱,然后分为 3 000 个分片,每个分片包含 20 张数据。这种划分方式确保可以通过控制分片数目来控制样本数量差距。第二个数据集是 CIFAR-10,包含 50 000 个训练样本和 10 000 个测试样本,共有 10 个类别。每个类别包含 6 000 个 32×

32 像素点的彩色图像。CIFAR-10 将识别任务迁移到普适物体,具有大量噪声和不同比例的物体识别,因此对图像识别更具挑战性。数据集的划分方式与 MNIST 类似,采用了非平衡划分。该文使用全连接神经网络(也称为多层感知机)进行训练。

3.2 实验设置

FlexFL 算法是基于 Pytorch 的客户端/服务器端联邦学习架构框架实现。该文主要考虑由于设备异构和数据不均衡的影响,为了排除参与者和中央服务器之间网络通信时间的影响,实验中的中央服务器的聚合和客户端的本地训练都在同一台具有 NVIDIA1080Ti 的机器上进行。实验相关参数设置如下:本地训练中,学习率为 0.01, batch 设置为 50, epoch 设置为 5。设备的异构性通过给予设备不同的计算资源来模拟,并在每一回合随机、动态地更新不同参与者的计算资源。数据的不均衡通过设置不同的环境来实现:

环境 A:随机选取 5% 的参与者平均分得 50% 的数据。其余参与者平均分得 50% 的数据。

环境 B:随机选取 10% 的参与者平均分得 50% 的数据。其余参与者平均分得 50% 的数据。

环境 C:随机选取 30% 的参与者平均分得 50% 的数据。其余参与者平均分得 50% 的数据。

3.3 性能比较和结果分析

(1) 准确率上升速率。

为了对 FlexFL 算法的性能有直观的认识,图 3 比较了两种算法在不同环境设置下的性能表现。可以观察到,在通信初始阶段, FedAvg 以较少回合达到了同样的准确率,这是因为,当数据分布不均衡、参与者计算资源随机变化的时候, FedAvg 在通信初期,每回合能够有较多的数据参与训练,所以准确率上升较快。而 FlexFL 对本地数据进行划分,对联邦学习训练服务

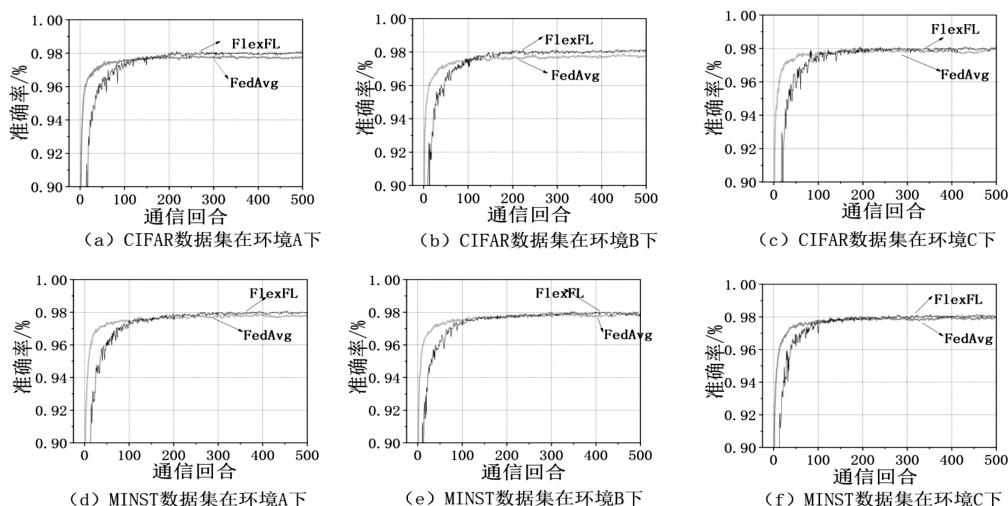


图3 不同设置下测试集准确率和通信回合的关系

随机激活,每次参与训练的数据比 FedAvg 少。当通信回合大于 150 的时候, FlexFL 以较少的通信回合达到同样的准确率,并且在相同的通信回合中,其准确率优于 FedAvg。这证明了 FlexFL 相较于 FedAvg 性能并没有降低。

(2) 平均时间消耗。

时间消耗在评价联邦学习算法的性能上也是一个重要指标。如图 4 所示, FlexFL 算法无论是准确率还是最后收敛时间都优于 FedAvg 算法。

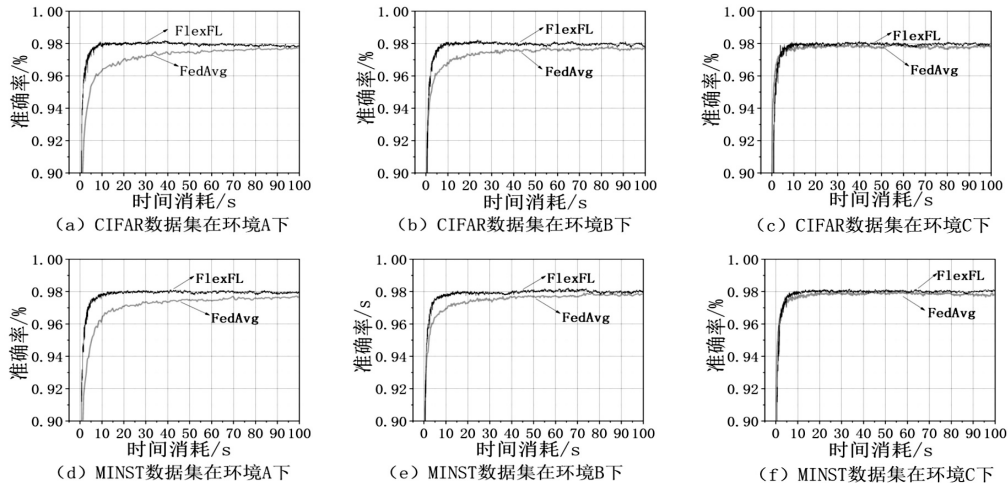


图4 不同设置下测试集准确率和时间消耗的关系

表3列出了不同环境设置下(对应图4中(a)~(f))使用两种算法达到目标准确率97.5%所需要的通信回合,平均每回合所需时间和总的时间消耗。在

表3中几组实验均显示, FlexFL 的每回合平均时间消耗和达到目标准确率总的时间消耗均比 FedAvg 的低。

表3 不同环境下 FlexFL 算法与联邦学习算法 FedAvg 在测试集上的表现对比

数据集	方案	数据分布	目标准确率/%	达到目标准确率所需通信回合数	平均每回合时间消耗/s	达到目标准确率所需时间消耗/s
MNIST	FlexFL	环境 A	97.5	110	0.047 8	4.495 8
		环境 B	97.5	78	0.044 0	3.432 0
		环境 C	97.5	92	0.048 1	4.425 2
	FedAvg	环境 A	97.5	81	0.576 4	46.688 4
		环境 B	97.5	110	0.314 3	24.515 4
		环境 C	97.5	46	0.112 4	5.170 4
CIFAR	FlexFL	环境 A	97.5	105	0.042 2	4.431 0
		环境 B	97.5	94	0.046 2	4.342 8
		环境 C	97.5	96	0.046 9	4.502 4
	FedAvg	环境 A	97.5	79	0.626 4	49.485 6
		环境 B	97.5	94	0.338 7	31.837 8
		环境 C	97.5	54	0.110 9	5.988 6

由表4显示,所提算法在不同环境下,用 MNIST 数据集测试,分别将训练时间消耗降低了 90.37%, 96.00% 和 14.41%。用 CIFAR 数据集测试,分别将训

练时间消耗降低了 91.04%, 86.35% 和 24.82%。这证实了在参与者数据分布不平衡、计算资源动态变化的时候, FlexFL 具有较低的时间消耗。

表4 不同环境下 FlexFL 达到目标准确率(97.5%)以 FedAvg 为基准的时间消耗减少率

指标	MNIST			CIFAR		
	环境 A	环境 B	环境 C	环境 A	环境 B	环境 C
FedAvg 时间消耗/s	46.688 4	24.515 4	5.170 4	49.485 6	31.837 8	5.988 6
FlexFL 时间消耗/s	4.495 8	3.432 0	4.425 2	4.431 0	4.342 8	4.502 4
减少率/%	90.37	86.00	14.41	91.04	86.35	24.82

值得说明的是,当数据分布均衡的时候, FlexFL 算法对本地数据不进行划分,参与者仅仅部署一个联邦学习服务,这本质上与 FedAvg 算法没有区别。

4 结束语

提出了一种基于动态缩放策略的联邦学习算法 FlexFL,旨在减少联邦学习每回合的时间消耗。该算法引入了两层联邦学习架构,通过将本地数据进行划分,并对应部署联邦学习训练服务,结合每轮参与者的计算资源变化,对部署在该参与者的联邦学习训练服务进行动态激活,使得参与者的每回合的时间消耗减少。实验结果表明,与联邦学习算法 FedAvg 相比,该算法有效降低了联邦学习的训练时间消耗。不足之处在于,需要在参与者部署多个联邦学习训练服务和一个聚合服务,这会带来一些额外的资源消耗; FlexFL 每一回合都需要对联邦学习服务进行动态激活,在复杂实际环境中,可能会有部分激活失败的情况。

在接下来的工作中,将对该算法进行改进。例如,在联邦学习训练服务进行缩放和本地数据集进行划分的时候不够细致,可能并没有达到最好的效果。其次,在中央服务器选择参与者的过程中采取的随机的方式,这可以结合其他研究进行优化。

参考文献:

- [1] ZHOU Y Z, ZHANG D. Near-ENF cloud computing: opportunities and challenges in the post-cloud computing era[J]. Chinese Journal of Computers, 2019, 42(4): 677-700.
- [2] CUI Y, SONG J, MIAO C C, et al. Mobile cloud computing research progress and trends[J]. Chinese Journal of Computers, 2017, 40(2): 273-295.
- [3] ABBAS N, ZHANG Y, TAHERKORDI A, et al. Mobile edge computing: a survey[J]. IEEE Internet of Things Journal, 2017, 5(1): 450-465.
- [4] KONECNY J, MCMAHAN H B, YU F X, et al. Federated learning: strategies for improving communication efficiency[J]. arXiv:1610.05492, 2016.
- [5] HANNAH L A. Stochastic optimization[J]. International Encyclopedia of the Social & Behavioral Sciences, 2015, 2: 473-481.
- [6] ZHAO Y, LI M, LAI L, et al. Federated learning with non-iid data[J]. arXiv:1806.00582, 2018.
- [7] SATTTLER F, WIEDEMANN S, MÜLLER K R, et al. Robust and communication-efficient federated learning from non-iid data[J]. IEEE Transactions on Neural Networks and Learning Systems, 2019, 31(9): 3400-3413.
- [8] CALDAS S, KONECNY J, MCMAHAN H B, et al. Expanding the reach of federated learning by reducing client resource requirements[J]. arXiv:1812.07210, 2018.
- [9] XIA W, WEN W, WONG K K, et al. Federated-learning-based client scheduling for low-latency wireless communications[J]. IEEE Wireless Communications, 2021, 28(2): 32-38.
- [10] 张婷, 吴宇. 面向 Non-IID 场景的联邦学习客户端选择算法研究[J]. 东莞理工学院学报, 2023, 30(3): 24-31.
- [11] YAO X, HUANG T, WU C, et al. Federated learning with additional mechanisms on clients to reduce communication costs[J]. arXiv:1908.05891, 2019.
- [12] KRAWCZYK B. Learning from imbalanced data: open challenges and future directions[J]. Progress in Artificial Intelligence, 2016, 5(4): 221-232.
- [13] KHAN L U, PANDEY S R, TRAN N H, et al. Federated learning for edge networks: resource optimization and incentive mechanism[J]. IEEE Communications Magazine, 2020, 58(10): 88-93.
- [14] ZHAN Y, LI P, QU Z, et al. A learning-based incentive mechanism for federated learning[J]. IEEE Internet of Things Journal, 2020, 7(7): 6360-6368.
- [15] SARIKAYA Y, ERCETIN O. Motivating workers in federated learning: a stackelberg game perspective[J]. IEEE Networking Letters, 2019, 2(1): 23-27.
- [16] 贾云健, 黄宇, 梁靓, 等. 基于主从博弈的分层联邦学习激励机制研究[J]. 电子与信息学报, 2023, 45(4): 1366-1373.
- [17] KHALEQ A A, RA I. Intelligent autoscaling of microservices in the cloud for real-time applications[J]. IEEE Access, 2021, 9: 35464-35476.
- [18] YAN M, LIANG X M, LU Z H, et al. HANSEL: adaptive horizontal scaling of microservices using Bi-LSTM[J]. Applied Soft Computing, 2021, 105: 107216.
- [19] ABDULLAH M, IQBAL W, MAHMOOD A, et al. Predictive autoscaling of microservices hosted in fog microdata center[J]. IEEE Systems Journal, 2020, 15(1): 1275-1286.
- [20] MCMAHAN H B, MOORE E, RAMAGE D, et al. Communication-efficient learning of deep networks from decentralized data[J]. arXiv:1602.05629, 2016.
- [21] 田有亮, 吴柿红, 李沓, 等. 基于激励机制的联邦学习优化算法[J]. 通信学报, 2023, 44(5): 169-180.
- [22] 程帆, 王瑞锦, 张凤荔. 边缘场景下动态权重的联邦学习优化方法[J]. 计算机科学, 2022, 49(12): 53-58.