计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2020.20190754
57(12): 2571-2582, 2020

一种面向边缘计算的高效异步联邦学习机制

芦效峰[1] 廖钰盈[1] Pietro Lio[2] Pan Hui[3]
[1](北京邮电大学网络空间安全学院 北京 100876)
[2](剑桥大学计算机实验室 英国剑桥 CB3 0FD)
[3](香港科技大学计算机科学与工程学院 香港 999077)
(luxf@bupt.edu.cn)

An Asynchronous Federated Learning Mechanism for Edge Network Computing

Lu Xiaofeng[1], Liao Yuying[1], Pietro Lio[2], and Pan Hui[3]
[1](*School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing* 100876)
[2](*Computer Laboratory, University of Cambridge, Cambridge CB3 0FD*)
[3](*Department of Computer Science & Engineering, Hong Kong University of Science and Technology, Hong Kong* 999077)

Abstract With the continuous improvement of the performance of the IoT and mobile devices, a new type of computing architecture, edge computing, came into being. The emergence of edge computing has changed the situation where data needs to be uploaded to the cloud for data processing, fully utilizing the computing and storage capabilities of edge IoT devices. Edge nodes process private data locally and no longer need upload a large amount of data to the cloud for processing, reducing the transmission delay. The demand for implementing artificial intelligence frameworks on edge nodes is also increasing day by day. Because the federated learning mechanism does not require centralized data for model training, it is more suitable for edge network machine learning scenarios where the average amount of data of nodes is limited. This paper proposes an efficient asynchronous federated learning mechanism for edge network computing (EAFLM), which compresses the redundant communication between the nodes and the parameter server during the training process according to the self-adaptive threshold. The gradient update algorithm based on dual-weight correction allows nodes to join or withdraw from federated learning during any process of learning. Experimental results show that when the gradient communication is compressed to 8.77% of the original communication times, the accuracy of the test set is only reduced by 0.03%.

Key words federated learning; edge computing; asynchronous distributed learning; gradient compression; privacy-preserving

摘 要 随着物联网和移动设备性能的不断提高,一种新型计算架构——边缘计算——应运而生.边缘计算的出现改变了数据需要集中上传到云端进行处理的局面,最大化利用边缘物联网设备的计算和存储能力.边缘计算节点对本地数据进行处理,不再需要把大量的本地数据上传到云端进行处理,减少了数据传输的延时.在边缘网络设备上进行人工智能运算的需求也在逐日增大,因为联邦学习机制不需要把数据集中后进行模型训练,所以更适合于节点平均数据量有限的边缘网络机器学习的场景.针对以上

收稿日期:2019-10-24;**修回日期**:2020-05-09
基金项目:国家自然科学基金项目(61472046);北京市科学技术协会种子基金项目;蚂蚁金服安全专项科研基金项目
This work was supported by the National Natural Science Foundation of China (61472046), the Beijing Association for Science and Technology Seed Fund, and the Ant Financial Security Special Research Fund.

挑战,提出了一种面向边缘网络计算的高效异步联邦学习机制(efficient asynchronous federated learning mechanism for edge network computing, EAFLM),根据自适应的阈值对训练过程中节点与参数服务器之间的冗余通信进行压缩.其中,双重权重修正的梯度更新算法,允许节点在学习的任何过程中加入或退出联邦学习.实验显示提出的方法将梯度通信压缩至原通信次数的 8.77%时,准确率仅降低 0.03%.

关键词 联邦学习;边缘计算;异步分布式学习;梯度压缩;隐私保护

中图法分类号 TP301.6

机器学习已经逐步地改变我们生活、学习、工作的方式,其在语音、图像和文本识别[1],语言翻译等方面都取得巨大突破.谷歌、Facebook 和苹果等大型公司从用户端收集大量训练数据,实现大规模的深度学习网络.虽然深度学习的实用性是不可否认的,但其使用的训练数据却可能涉及严重的隐私问题:数百万人的照片、视频被集中收集,这些数据被大公司永远地保存,用户既不能删除这些数据,也不能控制这些公司如何使用数据.其次,图像和视频中很可能包含大量敏感信息[2],如面孔、车牌、电脑屏幕、其他人的对话等.互联网巨头垄断这些"大数据",独享数据背后巨大的经济效益.

众所周知,随着训练数据量的增大,多样化的增多,机器学习所训练的模型会变得更好.然而,在许多领域,尤其与医学相关的领域,法律规定不允许共享与个人相关的数据.自 2017 年 6 月 1 日《中华人民共和国网络安全法》开始实施,个人信息安全被列入重点保护范围,国家对隐私条款提出了明确的要求,进一步完善了对个人信息的保护.因此,相关行业的研究人员只能对属于他们自己机构的数据集进行分析和挖掘.如果单个组织(如特定的医疗诊所)所拥有的数据量不是非常大,并且相似度较高,多样化不足,则在这样的数据集上进行机器学习,研究人员最终可能得到扩展性较差的模型,或者容易产生过拟合的结果.在这种情况下,数据的隐私和机密性的限制明显影响了机器学习的效果.

另一方面,目前全球物联网进入了第 3 次发展浪潮.2018 年全球物联网连接数约为 80 亿个[3],这些物联网设备产生大量的数据.在传统云计算架构中,这些数据需要集中传输到云端进行处理,这会加大网络负载,造成传输拥堵和数据处理的延迟.因此,传统云计算已经不适用于边缘物联网设备过多的情况.于是,一种新型的计算模型应运而生——边缘计算[4].边缘计算是指在靠近物或数据源头的一侧,就近提供计算、存储等服务.相比数据集中式的云计算模型,边缘计算在网络的边缘处理数据,这样能够降低网络带宽负载、减少请求响应时间、提升电池续航能力的同时保证数据的安全性和私密性.

由于物联网设备的长足发展,边缘设备具有了一定的计算能力和存储能力,这使得边缘设备与机器学习的结合不再只是一种假设.McMahan 等人[5]对联邦学习(federated learning)进行了一般性的描述,Bonawitz 等人[6]、Konečný 等人[7]和 McMahan 等人[8]也就该理论继续进行研究探索.联邦学习具有"联合学习"的含义——多台设备以协作的形式,共同训练预测模型.联邦学习可以被搭建在边缘设备(如智能电话、视频监控设备等)上.各个边缘节点在本地独立进行机器学习模型的训练,并通过中心服务器(如参数服务器)对全局模型进行优化合并.在整个联邦学习过程中,隐私数据不离开数据拥有者,且无须与其他节点共享数据,解决了隐私安全、数据安全等问题.

然而,面向边缘网络的联邦学习在实现中面临诸多挑战.首先,面向边缘网络的联邦学习中,因为任何一个独立的边缘设备所拥有的数据都是有限的,所以单独训练时,每个参与者的学习模型都容易陷入局部最优.其次,联邦学习中节点学习的梯度数据间接反映了训练样本的信息,攻击者能从有效的梯度信息反推出样本数据,需要降低梯度通信量来降低隐私泄露的可能性.

针对以上挑战,本文提出一种面向边缘计算的高效异步联邦学习机制(efficient asynchronous federated learning mechanism for edge network computing, EAFLM),可以实现在不共享数据的前提下对多方数据进行学习,实现更自由更高效的联邦学习.相比传统的分布式学习,EAFLM 各节点独立地在自己本地的数据集上进行训练,在不影响准确率的前提下赋予参与学习者更多的自由度和隐私保护.使用其他参与者学习的模型来优化本地学习的模型参数,可以有效地帮助每个参与者逃避局部

最优并使他们能够探索其他值,从而产生更准确的模型.

本文探讨了异步联邦学习的不同情况,对边缘节点异步联邦学习提出了双重权重的解决方案.异步联邦学习旨在为边缘学习节点提供更自由的学习方式,且降低高自由度带来的精度损失.本文的主要贡献有3个方面:

1) 区别于现有的分布式学习系统,提出一种更适合边缘网络中离散节点协作学习的异步联邦学习机制,使节点在不共享隐私数据的前提下从数据中进行学习.

2) 在前人工作的基础上设计了一种阈值自适应的梯度压缩算法,在将梯度通信次数压缩至原通信次数的8.77%时,测试集上准确率仅降低0.03%.

3) 为了适应边缘节点在进行模型训练时间上的随机特性,对异步联邦学习进行深入的探索,提出了双重权重的方法来解决异步学习带来的性能降低问题.

1 相关工作

许多研究者在分布式机器学习方面做了大量工作,已经有基于云的大规模分布式系统投入使用.许多系统支持多种数据处理方案,包括模型并行和数据并行[9-10].

常见的分布式架构中包含数据中心,该数据中心一般以参数服务器的形式存在.参数服务器允许大量的学习节点在共享的全局神经网络模型上进行协作学习[11-12].这一研究领域专注于如何设计高效的服务器架构,快速处理大小为$10^9 \sim 10^{12}$的向量.目前分布式学习的学习节点都是电脑等存储和计算较强且网络连接稳定的设备,对于诸如移动手机等性能较为波动的设备则少有关注.

联邦学习架构的出现吸引了许多学者的关注,其中关于Non-IID数据集和数据不平衡问题[13]是其中一大研究重点.而关于节点学习模式方面的研究一直都集中于同步学习算法,如McMahan等人[5]的联邦学习平均算法采取同步训练的方法;联邦学习中的隐私保护如差分隐私[8]和安全聚合[14]都需要在设备上进行同步操作,所以本质上还是属于同步训练的范畴.此外,研究人员开始尝试在车对车通信[15]和医疗应用[16]等领域实现联邦学习.

由于联邦学习需要与大量的学习节点进行梯度交互,而这些交互通信不仅带来巨大网络通信带宽,还存在安全隐患.有研究表明,可以通过从共享的梯度数据中获取隐私训练数据[17],研究者首先随机生成一对"虚拟的"输入和标签(dummy data and label),然后执行通常的前向传播和反向传播.从虚拟数据导出虚拟梯度之后,他们没有像传统优化那样更新模型权重,而是更新虚拟输入和标签,以最大程度地减小虚拟梯度和真实梯度之间的差异.当攻击结束后,私人数据便完全暴露了出来.值得注意的是,整个过程不需要训练数据集的任何额外信息.文中还给出一些能有效防止隐私泄露的方法,其中包括梯度量化和稀疏化.数据显示,当压缩率大于20%时,梯度压缩能有效地防止样本信息泄漏.

梯度量化和稀疏化不仅能保证样本数据的隐私安全,还能减轻网络的负载,克服分布式学习模型中的通信瓶颈[18].

梯度量化将梯度量化成低精度值以降低通信带宽.Frank等人[19]提出1-bit SGD(stochastic gradient descent)以减少梯度传输数据大小,并在传统语音应用中实现10倍加速.Dan等人[20]提出了另一种称为QSGD(quantized SGD)的方法,它平衡了准确度和梯度精度之间的权衡.类似于QSGD,Wen等人[21]开发了使用3-level梯度的TernGrad.这2项工作都展示了量化训练的收敛性.但是TernGrad仅对CNN进行了实验,QSGD也仅计算了RNN损失的精度.DoReFa-Net[22]尝试量化整个模型,包括梯度,使用1 b的权重和2 b的梯度量化.

Strom等人[23]使用阈值来实现梯度稀疏化——仅发送大于预先定义的恒定阈值的梯度.但是,在实践中阈值是难以选择的.因此,Dryden等人[24]分别选择了固定比例的正梯度和负梯度;Aji等人[25]提出了gradient dropping,使用单一绝对值阈值对梯度进行稀疏化.为了保持收敛速度,gradient dropping需要添加一个标准化层[26];gradient dropping可节省99%的梯度交换,同时在机器翻译任务中仅导致0.3%的BLEU损失;同时,Chen等人[27]提出根据局部梯度活动自动调整压缩率,并在数学上进行了证明,实现了对于完全连接层的200倍压缩,卷积层的40倍压缩,而在ImageNet数据集上降低的Top-1精度可忽略不计.Chen等人[28]提出了LAG(lazily aggregated gradient),自适应地对梯度进行计算并跳过部分梯度通信,以减少通信带宽并减轻服务器压力.基本原则是检测变化缓慢的梯度,并对这些梯度进行压缩.LAG很有参考价值,但对优化的问题

进行了限制:优化问题是凸问题且是 Lipschitz 平滑的.

本文提出的 EAFLM 中的梯度压缩属于梯度通信稀疏化的范畴,但与前文提到的方案不同,在本文中无需探索一个最优的梯度阈值,因为寻找最优阈值的工作是困难的,且不同问题对应的阈值也是不同的,限定梯度阈值也会使得异步联邦学习框架的扩展性降低.

如果将学习对象扩散到边缘节点,同步训练变得难以实现且不切合现实需求.虽然也有少数学者采用了异步学习方法,但仅是将异步作为一种正则化方法[29],缺少对异步联邦学习进行详细系统的研究,而且在大多数研究中各异步联邦学习参与节点所处的学习轮数是相差无几的,实质上还是当作同步联邦学习研究的.异步联邦学习中各节点学习轮数相差较大,甚至学习进程是完全错开的,这跟同步联邦学习情况是截然不同的.联邦学习过程由并联到串联的转变,不仅增加了总体学习时间,降低学习效率,还影响人工智能模型的精度,而这类型的异步联邦学习问题目前仍缺乏关注.本文对这类型的异步联邦学习问题进行了研究,降低了由异步联邦学习带来的精度损失.

2 一种面向边缘计算的高效异步联邦学习机制

2.1 联邦学习机制

联邦学习(federated learning)是一种新兴的人工智能基础技术,其设计目标是在保障终端数据和个人数据安全的前提下,在多参与方或多计算节点之间开展高效率的机器学习.其中,联邦学习可使用的机器学习算法不局限于神经网络,还包括随机森林等重要算法.

如图 1 所示,联邦学习机制由 1 个参数服务器和多个边缘节点组成,参数服务器负责收集各参加节点上传的梯度,根据优化算法对模型各参数进行更新,维护全局参数;参与节点独立地对本地拥有的敏感数据集进行学习.每轮学习结束后,节点将计算的梯度数据上传至参数服务器,由服务器进行汇总更新全局参数.然后节点从参数服务器下载更新后的参数,覆盖本地模型参数,进行下一轮迭代.整个学习过程中,节点只与参数服务器通信,除了共同维护的全局参数外节点无法获取有关其余节点的任何信息,保障隐私数据的机密性.

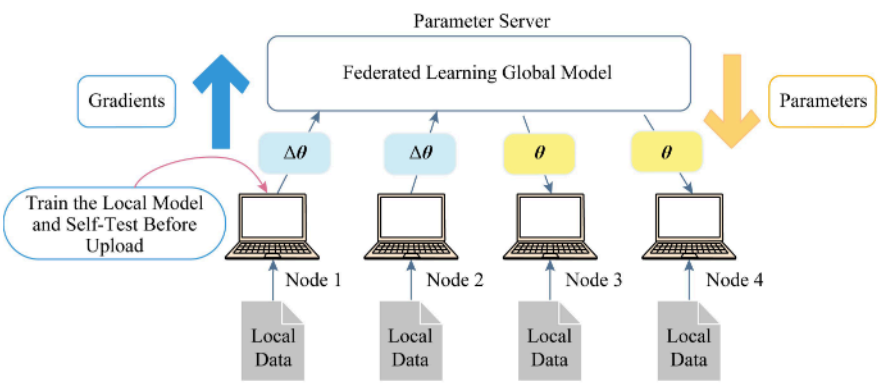


Fig. 1 Schematic diagram of the federal learning mechanism

图 1 联邦学习机制示意图

2.2 结构介绍

面向边缘网络的联邦学习需要与多个边缘学习节点进行模型数据的交互,这带来高昂的通信代价.经研究发现分布式 SGD 中 99.9%的梯度交换是冗余的,当参与学习的边缘节点数量增多时,这些冗余的梯度所需要的通信成本是不可忽视的,而且会给予参数服务器较大的通信压力,跳过节点的部分通信能有助于减轻通信负荷.且在联邦学习中,节点在执行梯度通信前会停止训练直至获得返回的最新参数,因此,梯度压缩还有助于减少联邦学习的总体模型训练时间.

图 2 所示为面向边缘网络计算的高效异步联邦学习机制框架图,框架分为参数服务器层和边缘节点层,其中阈值自适应模块处于边缘节点层,异步联邦学习模块横跨 2 层.

阈值自适应模块分为自适应阈值计算和梯度通信压缩 2 个子模块,分别负责根据最新参数变化计算阈值和使用阈值对不符合的梯度通信进行压缩.

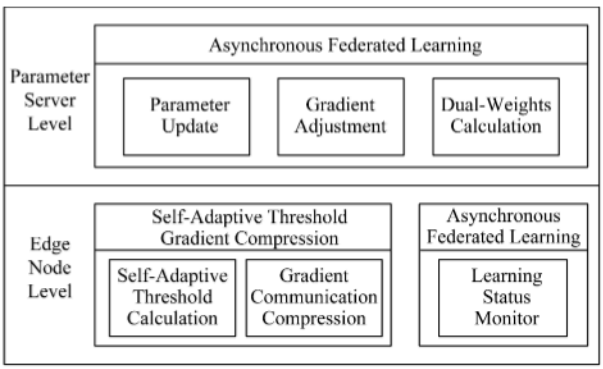


Fig. 2 Asynchronous federated learning mechanism for edge network computing

图 2 面向边缘网络计算的高效异步联邦学习框架

由于边缘节点具有高度自由性,因此强迫所有节点同时进行训练变得不切实际.异步联邦学习则分为 4 个子模块:参数更新、梯度修正、双重权重计算和节点学习状态监控.节点学习状态监控子模块负责监控节点学习状态如学习所处的轮数与样本数量等;双重权重计算子模块根据节点学习信息计算对应的样本权重和参数权重,具体计算方法在第 4 节进行介绍;梯度修正子模块将节点上传的梯度根据双重权重进行修正;而修正完的梯度则被用于参数更新子模块进行全局参数更新.

阈值自适应模块与异步联邦学习模块不是相互独立的,自适应的阈值计算需要获取更新后的参数,并与历史参数进行比较.而压缩后的梯度通信影响了节点学习状态,进而影响双重权重的计算.

3 阈值自适应梯度压缩

3.1 阈值自适应梯度压缩原理说明

梯度压缩是指对节点与服务器之间的梯度通信进行压缩,即压缩单个节点与参数服务器的通信次数.在前人的工作中,无论是简单地引入通信压缩率,还是根据固定阈值作为梯度检查的判别条件,都存在诸多不足之处.因为不同学习过程中梯度变化程度是不同的,简单地根据压缩率随机选取压缩的节点容易忽略拥有较大信息量的梯度,对全局模型训练造成影响;而固定阈值则会对梯度通信进行过度压缩,在训练的后期造成模型波动不易收敛的结果.

在 EAFLM 中,节点自动适应模型训练过程中每轮梯度的变化,计算出合适的阈值来对梯度通信进行压缩.只有符合条件的节点才能获取对应轮次的与参数服务器通信的资格,否则在本地累计梯度信息,进行下一轮学习迭代,最终梯度会累计足够的信息量上传到参数服务器.无论节点是否获得该轮的通信资格,在下一轮学习结束后都要进行梯度检查,即梯度检查是贯穿节点的整个学习过程.

3.2 梯度压缩自检式数学推导

本文对梯度压缩中的自检表达式进行研究,并在数学上进行了推导验证.

表 1 对本文所涉及的符号进行解释:

Table 1 The Meaning of the Symbol

表 1 公式中符号的表示含义

Symbol	Meaning
$\boldsymbol{\theta}^k$	The kth round parameter of the parameter server
$\nabla_i(\boldsymbol{\theta}^k)$	The kth round gradient calculated by node i based on the kth round parameter
∇_M^k	The sum of the kth round gradient of all nodes in the set M
M	The set contains all edge nodes
m	$card(M)$, the total number of all elements in the set M
D	Total number of samples owned by all nodes
D_i	Number of samples owned by node i
β_S^i	The sample weight of current epoch of node i
β_P^i	The parameter weight of current epoch of node i

在传统联邦学习机制中,存在一个参数服务器需要与 m 个学习节点通信以共同完成模型参数的更新:在第 k 轮迭代,参数服务器将当前模型 $\boldsymbol{\theta}^{k-1}$ 广播给所有学习节点;每个学习节点 $i \in M$ 计算 $\nabla_i(\boldsymbol{\theta}^{k-1})$ 并将其上传到参数服务器,各节点上传的梯度总和为 ∇_M^{k-1}.参数服务器从所有学习节点中接收聚合梯度 ∇_M^{k-1},执行优化算法更新模型参数.

本文参考了 Chen 等人[28]的工作,对通信中的梯度进行压缩,以减轻通信的带宽负担.算法的主要思想是忽略某轮次中“懒惰(lazy)”的节点,只与“勤奋(hard work)”的节点通信.这些被忽略的节点用 M_{L} 表示,而与服务器通信的节点的集合则表示为 M_{H},也即 $M=M_{\mathrm{L}}+M_{\mathrm{H}}$.因此 ∇_M^{k-1} 可改为

$$\nabla_M^{k-1}=\nabla_{M_{\mathrm{L}}}^{k-1}+\nabla_{M_{\mathrm{H}}}^{k-1}. \tag{1}$$

“懒惰”节点的定义可以说是阈值自适应梯度压缩的关键部分,不同的定义不仅会导致被忽略的信息量的不同和算法的压缩率,还会影响模型的准确率.在本文中,“懒惰”节点集合 M_{L} 满足:

$$\frac{\left\|\nabla_{M_{\mathrm{L}}}^{k-1}\right\|^2}{m_{\mathrm{L}}} \leqslant \frac{\left\|\nabla_M^{k-1}\right\|^2}{m}. \tag{2}$$

设更新参数的优化算法为梯度下降算法(gradient descent),即:

$$\boldsymbol{\theta}^k=\boldsymbol{\theta}^{k-1}-\alpha\ \nabla_M^{k-1}, \tag{3}$$

其中,α 是学习率.将式(3)代入式(2)可得:

$$\|\nabla_{M_L}^{k-1}\|^2 \leqslant \frac{m_L}{\alpha^2 m}\|\boldsymbol{\theta}^k-\boldsymbol{\theta}^{k-1}\|^2, \tag{4}$$

其中,$\|\nabla_{M_L}^{k-1}\|^2=\left\|\sum_{i\in M_L}\nabla_i(\boldsymbol{\theta}^{k-1})\right\|^2$,由均值不等式可知 $\|\nabla_{M_L}^{k-1}\|^2$ 满足:

$$\|\nabla_{M_L}^{k-1}\|^2 \leqslant m_L\sum_{i\in M_L}\|\nabla_i(\boldsymbol{\theta}^{k-1})\|^2. \tag{5}$$

若节点 $i\in M_L$ 满足式(6),则式(2)一定能被满足:

$$\|\nabla_i(\boldsymbol{\theta}^{k-1})\|^2 \leqslant \frac{1}{\alpha^2 m m_L}\|\boldsymbol{\theta}^k-\boldsymbol{\theta}^{k-1}\|^2. \tag{6}$$

由于集合 M_L 的总数无法事先获取,所以为了简化问题,我们引入比例系数 β 来衡量集合 M_L 的节点总数,即 $m_L=\beta m$.整理式(6)可得:

$$\|\nabla_i(\boldsymbol{\theta}^{k-1})\|^2 \leqslant \frac{1}{\alpha^2\beta m^2}\|\boldsymbol{\theta}^k-\boldsymbol{\theta}^{k-1}\|^2. \tag{7}$$

$\boldsymbol{\theta}^k-\boldsymbol{\theta}^{k-1}$ 的获取是困难的,但由于学习过程中参数变化趋于平滑,因此本文将 $\boldsymbol{\theta}^k-\boldsymbol{\theta}^{k-1}$ 近似为

$$\boldsymbol{\theta}^k-\boldsymbol{\theta}^{k-1}\approx\sum_{d=1}^{D}\xi_d(\boldsymbol{\theta}^{k-d}-\boldsymbol{\theta}^{k-1-d}), \tag{8}$$

其中,ξ_d 与 D 都是常数系数,简单地,可以选择 $\xi_d=\frac{1}{D}$.在本文相关实验中,$D=1$.

将式(8)代入式(7)可得:

$$\|\nabla_i(\boldsymbol{\theta}^{k-1})\|^2 \leqslant \frac{1}{\alpha^2\beta m^2}\left\|\sum_{d=1}^{D}\xi_d(\boldsymbol{\theta}^{k-d}-\boldsymbol{\theta}^{k-1-d})\right\|^2. \tag{9}$$

式(9)为节点进行梯度检查的自检表达式,即节点在 1 轮学习结束之后执行自检操作.需注意的是,不满足式(9)则与参数服务器通信;满足则跳过本轮通信,本地累计梯度,继续执行下一轮学习.

相较于传统联邦学习,本文的方法对网络带宽等设备配置的依赖性较小.因为带宽影响了节点传输时间,较低的带宽会导致节点每轮学习时间的增大,从而延长总体学习时间,因此若要控制总体学习时间,则需对带宽配置进行限制.但经梯度压缩后的联邦学习中梯度通信次数减少,受带宽影响的轮数也随之降低,因此理论上,本文的方法对带宽等配置的需求与未压缩的传统方法来比较小,能更好地适应带宽有限的网络.

4 面向边缘网络的高效异步联邦学习

EAFLM 面向的对象是高度自由的边缘节点,有许多因素都会引起异步学习,如加入学习的时间不同、节点的计算力不同(相同训练任务所需的时间不相同)、梯度压缩、由各种外部因素导致的学习中断等.本文提出参数权重来解决异步联邦学习问题.

4.1 异步联邦学习存在的问题

异步联邦学习中节点存在学习样本不均、学习进度各异等问题.如图 3 所示,图 3 中直线表示当前时间,浅色部分表示已经完成的学习任务,深色带底纹部分表示还未进行的学习任务.所有节点的学习总轮数是相同的,与时间线交差的部分表示当前所处的迭代轮次.图中的各个节点是处于学习的不同阶段,如节点 1 处于学习进程的 50%,若总轮数为 1 000 轮,则节点 1 处于第 501 轮.

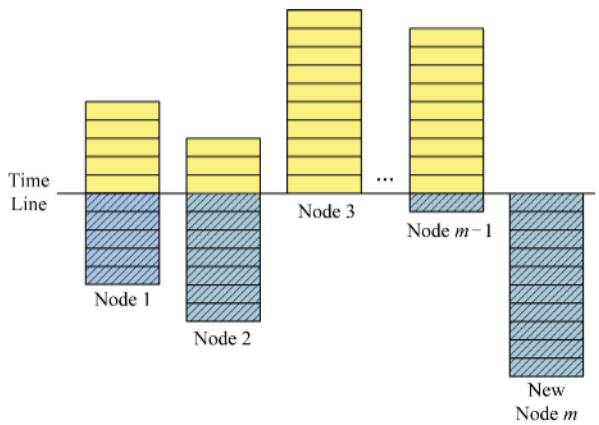


Fig. 3 Asynchronous federated learning

图 3 异步联邦学习

样本数量可以一定程度上反映样本多样性,基于高复杂度数据训练的模型具有更好的扩展性.而模型训练的过程可以理解为模型的“学习”的过程,一般地,随着时间的推移,模型越接近问题的最优解.如图 3 中,这些节点与学习问题最优解的距离是不同的,因此这些差异较大的的节点对参数服务器中的全局参数进行平等的更新显然是不合理的.我们引入双重权重来解决异步学习中节点学习状态不均衡的问题.

4.2 双重权重修正的异步联邦学习

面向边缘网络计算的高效异步联邦学习机制的权重分为 2 部分:样本权重和参数权重.样本权重是由节点的样本数量与总样本数量的占比决定,而参数权重则受梯度的目标参数与当前全局参数在时间上的相近程度所影响.

定义 1. 样本权重.代表一个节点所拥有的样本与总节点的样本总数的占比大小.

n 个学习节点的集合为 $N=\{N_1,N_2,\cdots,N_n\}$,

其中每个节点对应的样本数用 D_i 表示.节点 i 的样本权重可由节点的样本数与总样本数求得:

$$\beta_s^i=\frac{D_i}{D}, \tag{10}$$

其中,$D=\sum_{j=0}^{n}D_j$ 为 n 个节点总样本数.

定义 2. 参数权重.表示节点优化的参数与当前的全局参数在时间先后顺序上相差的程度.

从参数服务器的角度看整个联邦学习是一次次的参数优化更新,不间断地有节点向参数服务器提出下载最新参数的请求,又不停地有节点上传新的梯度以更新参数,简化的过程如图 4 所示:

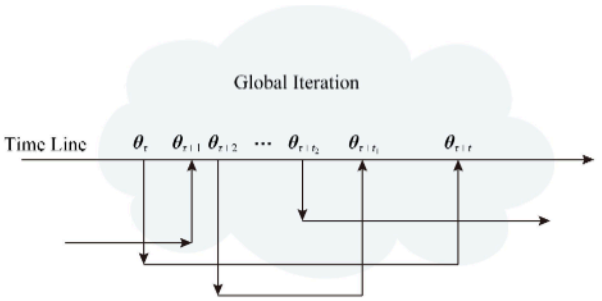


Fig. 4 Asynchronous federated learning and parameters staleness

图 4 异步联邦学习与参数陈旧度

每个节点在下载最新参数到上传对应的梯度之间的时间内都有其他节点的上传操作穿插其中.在图 4 描述的参数优化过程中梯度具有一定的陈旧性,令参数陈旧度为

$$\mu_{\text{staleness}}=I_{\text{upload}}-I_{\text{download}}. \tag{11}$$

在图 4 的例子中,在节点下载和上传梯度的时间间隔内,其他节点会上传新的梯度,参数服务器会进行梯度的更新,参数陈旧度表示了节点在完成一次迭代学习中参数服务器经过了几轮更新,这一定程度地反映了节点的计算能力.为了使陈旧度越大的节点其参数权重越小,且衰减的过程相对平缓,本文选择底数小于 1 的指数函数作为参数权重的衰减函数:

$$\beta_p^i=a^{\left(\frac{1}{m-1}\mu^i_{\text{staleness}}\right)-1}, \tag{12}$$

其中,$\mu^i_{\text{staleness}}$ 表示节点 i 的陈旧度,$a\in(0,1)$ 是参数权重中可调节的参数,决定了衰减的速度.随着 a 的增大,参数权重的衰减速度会增大;相反地,参数权重的衰减速度则会降低.简单地,在本文的实验中,$a=0.9$.

双重权重修正为

$$\boldsymbol{\theta}'=\boldsymbol{\theta}\times\beta_s^i\times\beta_p^i, \tag{13}$$

其中,$\boldsymbol{\theta}$ 是原始模型参数,$\boldsymbol{\theta}'$ 是修正后的模型参数.

在异步学习过程中,边缘节点提交的梯度在参数服务器上要经过双重权重的修正才能参与全局模型优化.修正后的梯度根据具体的优化算法更新全局参数,优化结束后,节点获得最新的参数并将其覆盖本地参数,进行下一轮迭代学习.

异步联邦学习对于网络性能的波动也具有很好的适应性,当某个节点的网络质量变差时,势必会影响节点学习的进程,造成各个节点学习进度各异.相较于传统联邦学习,基于双重权重修正的异步联邦学习对这些进度各异的节点贡献的梯度根据其学习程度进行修正,减小如网络性能波动造成的学习精度的影响,对性能不稳定的网络具有一定的鲁棒性.

5 实验结果

5.1 实验配置

为了模拟实际场景,我们搭建如下实验环境:计算力较强的 GPU 服务器作为参数服务器,负责大部分计算工作;其余多台电脑模拟边缘网络中的单个学习节点,各自独立地进行联邦学习,带宽为 1 Mbps.参数服务器与节点之间的通信使用 Thrift 框架.每台电脑本地存放部分数据(在我们模拟的实验中,每个节点的数据占总数据量的 0.2%),独自根据本地数据训练神经网络模型.本文研究的对象是多个边缘设备节点,这些节点的计算力各不相同,通过加入停顿间隔时间来模拟这些算力不同的设备.为了更好地模拟实验场景,这些电脑上的联邦学习都由一台独立的电脑统一控制.

实验环境如图 5 所示,其中 *Order* 表示 m 个节点启动顺序,*Interval* 表示节点从相邻 2 轮学习间隔停顿的时间.在初始化阶段,管理节点根据节点总数生成乱序序列 *Order*(如图 5 中的{4,1,6,9,…,5}),并对应生成随机序列 *Interval*(如图 5 中的{16.20,46.46,62.97,325.62,…,24.048}(s)).初始结束后,管理节点根据 *Order*:*Interval* 序列对启动对应节点.

5.2 实验指标

实验部分采用的评价指标为准确率(accuracy, *Acc*),压缩率(compression ratio, *CR*)和压缩平衡指数(compression balance index, *CBI*),其中准确率 *Acc* 是指 Top-1 Accuracy.

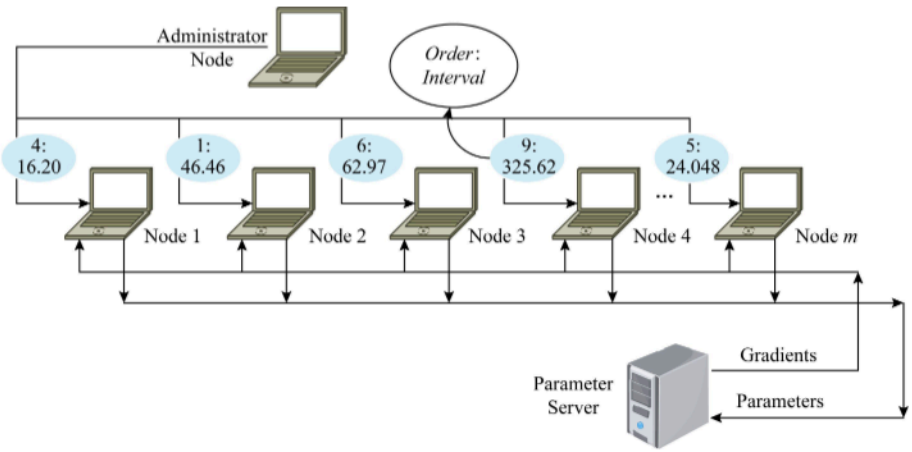


Fig. 5 Experiment configurations

图 5 实验配置

压缩率 CR 反映梯度压缩的压缩程度,压缩率越小,压缩程度越高.

$$CR=\frac{\text{压缩后的通信次数}}{\text{压缩前的通信次数}}\times 100\%. \tag{14}$$

一般地,压缩率的减小会导致准确率的逐渐降低.如何在 2 个指标中权衡,做出最佳决策成为难题.因此本文引入压缩平衡指数 CBI,表示梯度压缩综合效果,

$$CBI=\alpha_1\times Acc+\alpha_2\times(1-CR), \tag{15}$$

其中,α_1,α_2 是用于衡量 Acc 和 CR 重要性的 2 个可调节参数,$\alpha_1+\alpha_2=1$,$\alpha_1>0$,$\alpha_2>0$.若在现实情况中,Acc 优先级高于 CR,则可设置 $\alpha_1>\alpha_2$;反之,则 $\alpha_1<\alpha_2$;若两者优先级相同,则 $\alpha_1=\alpha_2$.CBI 值越高表示梯度压缩综合效果越好.

5.3 梯度压缩实验

我们在 MNIST 数据集上评估 EAFLM 的梯度压缩性能,MNIST 数据集是一个手写体数据集,共有 6 万个训练样本和 1 万个测试样本.本文对格式为 32×32 的图像进行归一化,使得数字位于图像的中心.

在这部分实验中共有 3 个学习节点,节点训练的模型结构参考了 Tensorflow 官网中的 Tutorial,我们未对模型结构进行调整.模型结构为 3 层 MLP 模型,每层的神经元个数分别为 256,256,10.本次实验将数据集平均分为 3 份,学习节点随机获得其中的 1 份.在每次实验中,节点都会重新获取新的数据.我们对不同参数 β 分别进行实验,并对最终结果求平均值.

补充说明:本文的研究重点是联邦学习框架中的通信,未对模型结构和优化算法进行过多研究,因此本文实验中出现的过拟合等问题其背后的原因及解决方案不属于本文研究范围.同样地,准确率等实验指标仅是为了对比各方法的性能,不对模型的优劣进行评估.且由于不同方法的对比实验采用相同的配置,实验中的指标仍具有可用于比较的价值.EAFLM 不对学习模型进行限制,因此在现实运用中使用者可以根据实际学习问题对模型结构和实验参数进行调整.

5.3.1 梯度压缩实验分析——压缩率及准确率

表 2 表示不同 β 取值对压缩率及准确率的影响.从整体上来看,随着 β 值的减小,压缩率也逐渐减小,即压缩程度增大.而随着梯度通信逐渐被压缩,准确率也随之降低.从表 2 中也可看出数据存在一定的波动,这是由于不同对比组的实验设置(如样本序号和初始模型参数)的不同导致模型训练结果

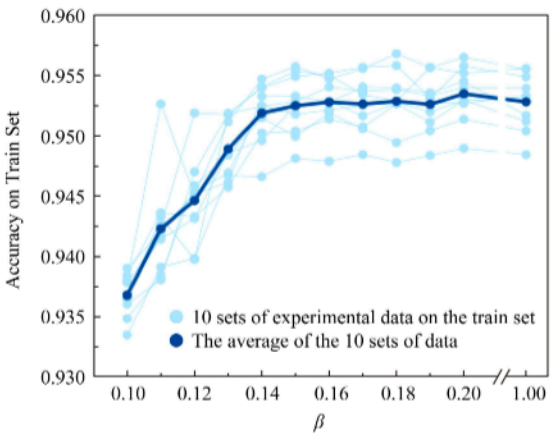
Table 2 Compression and Accuracy Under Different β

表 2 不同 β 值下的压缩率及准确率

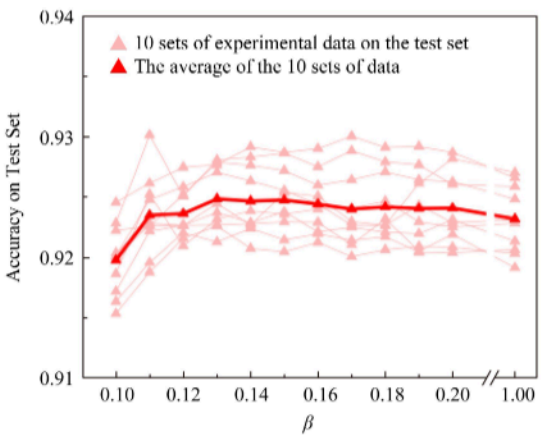
β	Average Communication Times After Compression	Accuracy on Train Set	Accuracy on TestSet	Compression Ratio/%
0.1	19	0.9368	0.9198	6.33
0.2	214	0.9540	0.9240	71.3
0.3	270	0.9530	0.9250	90
0.4	283	0.9534	0.9252	94.3
0.5	290	0.9546	0.9252	96.7
0.6	292.5	0.9544	0.9244	97.5
0.7	296.5	0.9545	0.9240	98.83
0.8	296.5	0.9531	0.9235	98.83
0.9	297	0.9541	0.9242	99
1.0	300	0.9541	0.9241	100

存在个体差异,且实验模型参数初始值不同也是造成数据小范围抖动的原因之一.但明显地,在 $\beta\in[0.1,0.2]$ 区间时,压缩率存在明显的上升,而在 $\beta\in[0.2,1]$ 区间内,压缩率上升幅度较小,因此在图 8 中,我们着重展示的是 $\beta\in[0.1,0.2]$ 区间.

图 6 为 10 组不同超参数(样本序号和停顿间隔时间)下对比实验的结果,图 6(a)浅色细线条和图 6(b)浅色细线条分别为 10 次实验中在测试集和训练集上的具体数据波动,对应的图 6(a)深色加粗线条和图 6(b)深色加粗线条为测试集和训练集 10 次实验的平均数据.



(a) Train set



(b) Test set

Fig. 6 Different β values and model accuracy on data sets

图 6 梯度压缩中不同 β 取值在数据集上的准确率

在图 6 中,10 组实验数据虽然在个体上存在波动,但总体上呈现出随着压缩率的增大,准确率随之上升,这也可从对应的深色加粗线条看出.有趣的是,图 6(a)中可看出,在 $\beta\in[0.14,1]$ 区间内(对应图 6(b)的区间则为 $\beta\in[0.13,1]$),梯度压缩对准确率的影响不大,且出现了波动,这是因为所选的学习模型已达到最大的学习效果.明显地,冗余梯度通信对达到饱和学习的模型的增益较小,梯度压缩有助于提高联邦学习的性能.在图 7 中,随着 β 的增大,梯度通信次数在不断增大,压缩率也随之增大.

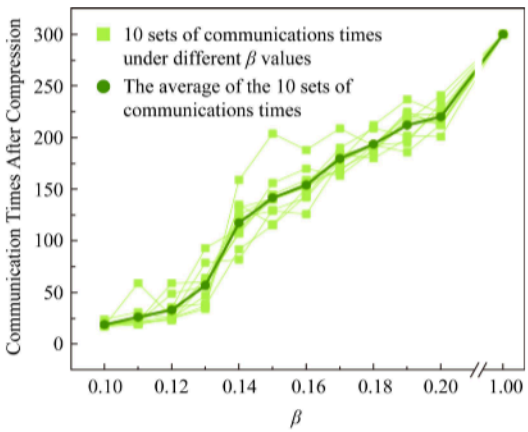


Fig. 7 Different β and communications times

图 7 梯度压缩中不同 β 取值对通信次数的影响

5.3.2 梯度压缩实验分析——压缩平衡指数

为了选择最佳 β 值,本文对 $\beta\in[0.1,0.2]$ 区间中的实验数据计算 CBI 值,计算结果如图 8 所示.结合图 8 可以发现本文实验中最佳 β 值为 0.1.

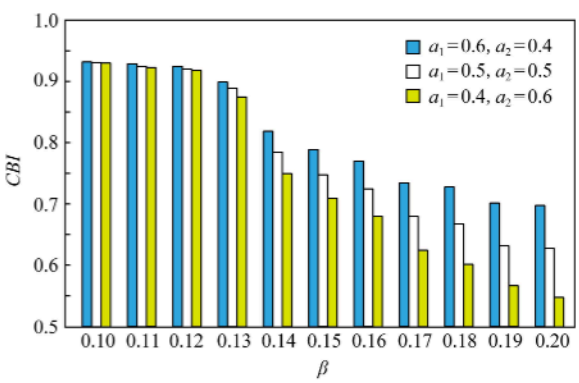


Fig. 8 CBI of different β values in gradient compression

图 8 梯度压缩中不同 β 取值下的 CBI 值

我们对 Chen 等人[28]的工作(LAG)做了对比实验,表 3 从准确率、压缩率、压缩平衡指数 3 个方面展示了 EAFLM 中阈值自适应梯度压缩算法与 LAG 算法的性能比较.从表 3 可看出 LAG 的压缩率比 $\beta=0.1$ 和 $\beta=0.11$ EAFLM 的压缩率低,压缩程度较高.但无论是在训练集还是测试集上,准确率都比 EAFLM 低.为了进一步比较两者性能,我们取 3 组不同的 a_1,a_2 组合计算 CBI.从表 3 可看出,

除了在 $a_1=0.4, a_2=0.6$ 时，LAG 的 CBI 比 $\beta=0.11$ EAFLM 高，其余两者情况下，EAFLM 性能都高于 LAG，$\beta=0.1$ 的 EAFLM 更是 3 种情况下都优于 LAG.且在 EAFLM 中，根据不同的实际需求调节 β 的取值，以达到最佳的压缩程度.

Table 3 Performance Comparison Between Different β Values of EAFLM and LAG

表 3 不同 β 值下的 EAFLM 与 LAG 的性能对比

Experimental Indicators	EAFLM		LAG
	$\beta=0.1$	$\beta=0.11$	
Acc(Train Set)	0.936 8	0.942 3	0.899 0
Acc(Test Set)	0.919 8	0.923 5	0.891 3
CR/%	6.33	8.77	5.11
$CBI(a_1=0.4, a_2=0.6)$	0.933 3	0.920 6	0.927 4
$CBI(a_1=0.5, a_2=0.5)$	0.932 5	0.922 6	0.922 0
$CBI(a_1=0.6, a_2=0.4)$	0.931 6	0.924 7	0.916 7

5.4 异步联邦学习

为了使异步学习实验效果更加显著，异步学习这部分的实验数据采用 Cifar10 数据集，CIFAR-10 数据集由 10 个类的 60 000 个 32×32 彩色图像组成，每个类有6 000 个图像.有 50 000 个训练图像和 10 000 个测试图像.我们将训练集平均分为 500 份，每次实验开始前，节点随机从中挑选一份数据进行训练为了模拟异步联邦学习，学习节点的数量增加到 10 个.除此之外，本实验在图 5 的基础上增加停顿时间，即每个节点在 1 轮训练结束后停顿一段时间.停顿时间序列由管理节点生成.同样地，本次实验采用了 Tensor-flow 官网 Tutorial 中的卷积网络代码，模型结构为 5 层卷积网络模型.由于异步联邦学习对象是高自由度的边缘设备，这些节点无法像服务器等设备一样稳定、长时间地训练网络模型.因此为了模拟这种情况，我们将训练轮数设置为 500 轮.学习轮数的减小使得实验中总体的准确率都不高，这可以通过调参等方法来提高准确率，这里不予过多讨论.实验中设置的超参数数值仅供参考，实际运用中仍需根据具体问题进行设置调整.

我们分别随机选取 5 组不同的超参数组(子样本集序号、停顿时间)进行 3 个对比实验，分别是节点训练中无停顿(对应表 4 中的 Synchronous)、节点训练中有停顿(Asynchronous)和在双重权重下的训练中有停顿(EAFLM).在我们查阅的文献范围内，没有发现其他学者针对异步联邦学习进行算法上的改进，因此本文无法与更多方法进行对比实验.

表 4 为 3 种算法在训练集和测试集中的 5 组不同超参数组合实验的平均准确率.从表 4 可看出异步学习比同步学习具有较好的性能，这可以理解为异步类似于一项正则化处理，能一定程度防止学习的过拟合.在训练集上，对比同步联邦学习和异步联邦学习，EAFLM 机制实现了 5.1%和 4%的准确率提升；而在测试集上，EAFLM 的准确率分别提升了 0.36%和 0.15%.

Table 4 Asynchronous Experiment Performance on Data Sets

表 4 异步实验在数据集上的准确率

Experimental Indicators	Synchronous	Asynchronous	EAFLM
Acc(Train Set)	0.642 0	0.653 0	0.693 0
Acc(Test Set)	0.477 3	0.479 4	0.480 9

5.5 综合实验分析——梯度压缩与异步联邦学习结合

综合实验部分仍然采用与异步联邦学习部分相同的实验配置，我们选取了 5 种不同 β 值下的 EAFLM 与异步联邦学习进行对比实验，其中异步联邦学习是无梯度压缩和权重修正.表 5 从准确率、压缩率、压缩平衡指数 3 个方面展示了 EAFLM 与异步学习算法的综合性能比较.从表 5 可以看出，当 $\beta=0.5$ 时，在准确率方面，异步联邦学习与 EAFLM 基本相当，但异步联邦学习在压缩和综合性能方面都比 EAFLM 差.在压缩率方面，$\beta\in[0.2, 0.5]$下的

Table 5 Performance Comparison Between Different β Values of Comprehensive EAFLM and Asynchronous Federated Learning

表 5 不同 β 值下的综合 EAFLM 与异步学习的性能对比

Experimental Indicators	EAFLM					Asynchronous Learning
	$\beta=0.5$	$\beta=0.4$	$\beta=0.3$	$\beta=0.2$	$\beta=0.1$	
Acc(Train Set)	0.728	0.725	0.746	0.687	0.616	0.721
Acc(Test Set)	0.487	0.477	0.483	0.479	0.460	0.490
CR/%	97.5	97.4	96.7	91.5	75.1	100
$CBI(a_1=0.4, a_2=0.6)$	0.258	0.256	0.266	0.284	0.364	0.242
$CBI(a_1=0.5, a_2=0.5)$	0.316	0.313	0.324	0.334	0.393	0.303
$CBI(a_1=0.6, a_2=0.4)$	0.374	0.371	0.382	0.384	0.422	0.363

EAFLM 都相差不大,但 EAFLM 在 $\beta=0.1$ 时,$CR=75.1\%$,即相对于异步联邦学习算法来说,EAFLM 进一步节省了 24.9%的通信成本,但在训练集和测试集上的准确率仅降低 10.5%和 3%.在 3 组不同的 a_1,a_2 组合计算的 CBI 中,不同 β 取值下的 EAFLM 都比异步联邦学习高,因此,本次实验结果显示 EAFLM 的综合性能比异步联邦学习有所提高,其中 $\beta=0.1$ 时,EAFLM 的综合性能达到最优.

6 总 结

本文提出一种面向边缘网络计算的高效异步联邦学习机制(EAFLM)可以满足在不共享隐私数据的前提下对多方数据进行学习的这个现实需求,以实现更自由高效的联邦学习.EAFLM 在不影响准确率的情况下赋予参与学习者更多的自由度和隐私保护.参与者在本地对拥有的数据集进行学习,在 1 轮训练之后,参与者根据自检条件来检查本轮是否满足与参数服务器通信的条件,若满足则将梯度上传至参数服务器.所有节点都是异步地执行以上步骤,不用等待其余节点,也无须同步学习轮次.整个学习过程中,节点只与参数服务器通信,除了共同维护的全局参数外节点无法获取有关其余节点的任何信息.

本文从阈值自适应梯度压缩和异步学习 2 大方面对 EAFLM 进行阐述,并进行相关实验.我们提出的阈值自适应梯度压缩算法,可以自适应模型训练过程中每轮梯度的变化,计算出合适的阈值来对梯度通信进行压缩.由于梯度数据间接反映了节点样本信息,攻击者能从有效的梯度信息反推出样本数据,减少梯度通信能有效地降低隐私泄露的可能性.测试集上,本文实现了将梯度通信压缩至原通信次数的 8.77%时,准确率仅降低 0.03%.

在异步联邦学习中,节点存在学习样本不均、学习进度各异等问题.这些差异较大的节点对参数服务器中的全局参数进行平等的更新显然是不合理的,因此我们引入双重权重来解决异步学习中节点学习状态不均衡的问题.实验表示在训练集上,对比同步联邦学习和异步联邦学习,EAFLM 机制实现了 5.1%和 4%的准确率提升;而在测试集上,EAFLM 的准确率分别提升了 0.36%和 0.15%.

我们的工作也存在不足之处,在阈值自适应梯度压缩中,执行梯度检查是牺牲单个节点本地计算时间来减少全局通信时间的典型例子.在众多神经网络学习中,学习迭代的轮数都是较大,轮数上万也并不少见,因此每轮结束都对梯度进行检查带来的计算时间无疑也是不容忽视的.为了缓解本地计算量,我们可以在上文描述的梯度压缩的基础上增加一个“免检”机制,这有助于减轻梯度检查带来的计算工作.在未来的工作中,我们将进一步对“免检”机制进行研究.

参 考 文 献

[1] He K, Zhang Xiangyu, Ren Shaoqing, et al. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification [C] //Proc of the IEEE Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2015: 1026-1034

[2] Shultz D. When your voice betrays you [J]. Science, 2015, 347(6221): 494-494

[3] Tian Jian. Analysis on the application of Internet of things in Chengdu broadcasting network [J]. Telecom World, 2019, 26(5): 35-36 (in Chinese)
(田健. 浅析物联网在成都广电网络中的运用[J]. 通讯世界, 2019, 26(5): 35-36)

[4] Shi Weisong, Cao Jie, Zhang Quan, et al. Edge computing: Vision and challenges [J]. IEEE Internet of Things Journal, 2016, 3(5): 637-646

[5] McMahan B, Ramage D. Federated learning: Collaborative machine learning without centralized training data [J/OL]. [2018-12-23]. http://www.googblogs.com/federated-learning-collaborative-machine-learning-without-centralized-training-data/

[6] Bonawitz K, Eichner H, Grieskamp W, et al. Towards federated learning at scale: System design [J]. arXiv preprint, arXiv:1902.01046, 2019

[7] Konečný J, McMahan H B, Yu F X, et al. Federated learning: Strategies for improving communication efficiency [J]. arXiv preprint, arXiv:1610.05492, 2016

[8] McMahan B, Ramage D, Talwar K, et al. Learning differentially private recurrent language models [J]. arXiv preprint, arXiv:1710.06963, 2017

[9] Dean J, Corrado G, Monga R, et al. Large scale distributed deep networks [C/OL] //Advances in Neural Information Processing Systems. 2012: 1223-1231. https://proceedings.neurips.cc/paper/2012/hash/6aca97005c68f1206823815f66102863-Abstract.html

[10] Low Y, Gonzalez J, Kyrola A, et al. Distributed graphlab: A framework for machine learning in the cloud [J]. arXiv preprint, arXiv:1204.6078, 2012

[11] Li Mu, Andersen D G, Park J W, et al. Scaling distributed machine learning with the parameter server [C] //Proc of the 11th USENIX Symp on Operating Systems Design and Implementation (OSDI'14). Berkeley, CA: USENIX Association, 2014: 583-598

[12] Abadi M, Agarwal A, Barham P, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems [J]. arXiv preprint, arXiv: 1603.04467, 2016

[13] Lim W Y B, Luong N C, Hoang D T, et al. Federated learning in mobile edge networks: A comprehensive survey [J]. arXiv preprint, arXiv: 1909.11875, 2019

[14] 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

[15] Samarakoon S, Bennis M, Saad W, et al. Federated learning for ultra-reliable low-latency V2V communications [C] // Proc of the 2018 IEEE Global Communications Conf (GLOBECOM). Piscataway, NJ: IEEE, 2018: 1-7

[16] Brisimi T S, Chen R, Mela T, et al. Federated learning of predictive models from federated electronic health records [J]. International Journal of Medical Informatics, 2018, 112: 59-67

[17] Zhu Ligeng, Liu Zhijian, Han Song. Deep leakage from gradients [C] //Advances in Neural Information Processing Systems. New York: Curran Associates, 2019: 14747-14756

[18] Lin Yujun, Han Song, Mao Huizi, et al. Deep gradient compression: Reducing the communication bandwidth for distributed training [J]. arXiv preprint, arXiv: 1712.01887, 2017

[19] Seide F, Fu Hao, Droppo J, et al. 1 bit stochastic gradient descent and its application to data-parallel distributed training of speech DNNs [C/OL] //Proc of the 15th Annual Conf of the Int Speech Communication Association. 2014 [2019-10-24]. https://www.researchgate.net/publication/290231878_1-bit_stochastic_gradient_descent_and_its_application_to_data-parallel_distributed_training_of_speech_DNNs

[20] Alistarh D, Grubic D, Li Jerry, et al. Qsgd: Randomized quantization for communication-optimal stochastic gradient descent [J]. arXiv preprint, arXiv: 1610.02132, 2016

[21] Wen Wei, Xu Cong, Yan Feng, et al. Terngrad: Ternary gradients to reduce communication in distributed deep learning [C] //Advances in Neural Information Processing Systems. New York: Curran Associates, 2017: 1509-1519

[22] Zhou Shuchang, Wu Yuxin, Ni Zekun, et al. DoReFa-Net: Training low bitwidth convolutional neural networks with low bitwidth gradients [J]. arXiv preprint, arXiv: 1606.06160, 2016

[23] Strom N. Scalable distributed DNN training using commodity GPU cloud computing [C/OL] //Proc of the 16th Annual Conf of the Int Speech Communication Association. 2015 [2019-10-24]. https://isca-speech.org/archive/interspeech_2015/i15_1488.html

[24] Dryden N, Moon T, Jacobs S A, et al. Communication quantization for data-parallel training of deep neural networks [C] //Proc of the 2nd Workshop on Machine Learning in HPC Environments (MLHPC). Piscataway, NJ: IEEE, 2016: 1-8

[25] Aji A F, Heafield K. Sparse communication for distributed gradient descent [J]. arXiv preprint, arXiv: 1704.05021, 2017

[26] Ba J L, Kiros J R, Hinton G E. Layer normalization [J]. arXiv preprint, arXiv: 1607.06450, 2016

[27] Chen C, Choi J, Brand D, et al. Adacomp: Adaptive residual gradient compression for data-parallel distributed training [C] //Proc of the 32nd AAAI Conf on Artificial Intelligence. Menlo Park, CA: AAAI, 2018

[28] Chen Tianyi, Giannakis G, Sun Tao, et al. LAG: Lazily aggregated gradient for communication-efficient distributed learning [C] //Advances in Neural Information Processing Systems. New York: Curran Associates, 2018: 5050-5060

[29] Shokri R, Shmatikov V. Privacy-preserving deep learning [C] //Proc of the 22nd ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2015: 1310-1321

Lu Xiaofeng, born in 1976. PhD, associate professor. His main research interests include cyberspace security, information security and artificial intelligence.

Liao Yuying, born in 1994. Master. Her main research interests include federated learning, edge computing, privacy preservation and AI security.

Pietro Lio, born in 1966. PhD, professor. His main research interests include bio-inspired design of wireless networks; epidemiological networks.

Pan Hui, born in 1978. PhD, professor. His main research interests include delay tolerant networking, mobile networking and systems, planetscale mobility measurement, social networks, and the application of complex network science in communication system design.