文章编号:1671-4229(2022)03-0001-13

基于秘密共享的本地多节点联邦学习算法

王捍贫, 范耀榕
(广州大学 计算机科学与网络工程学院, 广东 广州 510006)

摘 要: 深度学习技术在各个领域中的应用越来越广泛,而深度学习模型的准确性需要依靠大量的训练数据。由于数据安全和法规限制,许多领域存在无法集中数据进行训练的情况,导致"数据孤岛"的现象。对此,谷歌提出能使大量客户端在数据保存本地的情况下与可信服务器联合训练模型的联邦学习。目前,联邦学习的研究主要集中在安全性和训练效率的问题上,针对跨数据库联邦学习场景,文章将分层联邦学习和基于安全多方计算的隐私保护机制结合,提出了一种基于秘密共享的本地多节点联邦学习算法 Mask-FL,以保证联邦学习安全性的同时提高训练效率。主要工作包括:①提出本地多节点的跨数据库联邦学习框架,客户端利用本地计算资源生成多个本地节点,并且根据基于计算能力的数据划分方法进行分配数据资源,每个客户端代表局部所有节点参与全局联邦学习训练,从而构成3层级的分层联邦学习;②提出基于秘密共享的自适应掩码加密协议,在前面提出的联邦学习框架基础上,通过秘密共享的方式生成可复用的安全参数掩码,本地节点在训练过程的上行通信中对模型添加掩码从而保护模型参数安全。经过安全性假设分析证明,该算法可保护客户端的数据隐私安全。在通用数据集的实验表明,该算法能够在保护隐私的前提下保持相对较高的准确率,并且显著减少了全局通信轮次,训练效率相比于传统联邦学习方法提高30%,有效地提高了跨数据库联邦学习中的模型训练速度。

关键词: 联邦学习; 安全多方计算; 分布式计算; 隐私保护
中图分类号: TP 183 **文献标志码**: A

Local multi-node federated learning algorithm based on secret sharing

WANG Han-pin, FAN Yao-rong
(School of Computer Science and Cyber Engineering, Guangzhou University, Guangzhou 510006, China)

Abstract: The application of deep learning technology in various fields is becoming more and more extensive. However, the accuracy of deep learning models needs to rely on a large amount of training data. Due to data security and regulatory restrictions, many fields cannot centralize data for training, resulting in the phenomenon of "data silos". In this regard, Google proposes federated learning that enables a large number of clients to jointly train models with trusted servers while the data is stored locally. At present, the research of federated learning mainly focuses on the issues of security and training efficiency. For the cross-database federated learning scenario, this paper combines hierarchical federated learning with a privacy protection mechanism based on secure multi-party computation, and proposes a local multi-node mask federated learning based on secret sharing. The algorithm Mask-FL can improve the training efficiency while ensuring the security of federated learning. The main work includes: ① This paper proposes a local multi-node cross-database federated learning framework. The client uses local computing resources to generate multiple local nodes, and allocates data resources according to the data division method based on computing power. Each client participates in global federated learning training on behalf of all local nodes, thus constituting a three-level hierarchical federated learning. ② An adaptive mask encryption protocol based on secret sharing is proposed. On the

作者简介: 王捍贫(1964—),男,教授,博士生导师. E-mail:wanghp@gzhu.edu.cn
引文格式: 王捍贫, 范耀榕. 基于秘密共享的本地多节点联邦学习算法[J]. 广州大学学报(自然科学版),2022,21(3):1-13.

basis of the local multi-node federated learning framework proposed above, a reusable security parameter mask is generated by secret sharing. The local node adds a mask to the model in the uplink communication of the training process to protect the model parameters. After the security hypothesis analysis, it is proved that the algorithm can protect the data privacy security of the client. Experiments on general data sets show that the algorithm can maintain a relatively high accuracy while protecting privacy, and significantly reduces the number of global communication rounds. Compared with the traditional federated learning method, the training efficiency is improved by 30%, effectively improve the training speed of federated learning models.

Key words: federated learning; secure multi-party computation; distributed computing; privacy protection

目前,深度学习技术在各个领域发展迅速,这离不开数据和算力的爆发。深度学习模型的准确度依赖于数据量,由于隐私问题不断出现,人们逐渐重视数据的隐私安全,与此同时,政府也颁布隐私保护法律法规(如:GDPR[1]、《中华人民共和国网络安全法》[2]等)进一步保护用户数据安全,导致数据不断分散出现"数据孤岛"现象,从而无法聚集数据来训练高精度模型。

对此,Google 于 2016 年提出联邦学习(Federated Learning, FL)理论[3]。在 FL 系统中,参与者通过在自己的私有本地数据上执行本地训练算法,并仅与中心服务器共享模型参数,此中心服务器用作中央聚合器,以适当方式聚合本地参数更新后与每个参与者共享聚合的更新。然而参与者与中心服务器之间存在高频通信及长传输延迟,这导致 FL 不得不面对通信效率问题[4]。

根据应用场景的不同,联邦学习可分为跨设备联邦学习(Cross-device federated learning)和跨数据库联邦学习(Cross-silo federated learning)[5]。联邦学习的通信拓扑图一般为星形拓扑,对于跨设备联邦学习而言,参与训练的客户端为数量庞大的 IoT 设备或者移动设备,并且具有本地数据量少、通信不稳定和客户端间不互信的特性,这往往会对中心服务器的通信造成巨大的压力。对此,最近的研究提出了分层联邦学习框架(Hierarchical Federated Learning, HFL)[6-7],在客户端-中心服务器的结构中加入边缘服务器,形成客户端-边缘服务器-中心服务器结构,在训练过程中相邻的客户端将模型发送到近端边缘服务器进行聚合,然后由边缘服务器发送局部聚合模型到中心服务器进行最终聚合,从而减少中心服务器的通信压力。目前的 HFL 主要是利用物理层的中间设备充当边缘服务器,如 Mehdi 等[8]利用物理层面的小型蜂窝基站(Small-cell Base Station, SBS)来充当边缘服务器,大型基站(Macro-cell Base Station, MBS)作为中心服务器,构建分层联邦学习训练架构,移动用户将与最近的 SBS 进行通信形成局部交流结构,SBS 聚合模型后再与 MBS 通信从而减少 MBS 的通信量。而在跨数据库联邦学习的场景中,其客户端数量通常在 100 个以内,每个客户端具有数据量大、通信可靠、计算资源丰富和客户端间不互信等特性,并且都参与每个轮次的训练。以往的 HFL 在该场景下反而可能会由于额外增加的边缘服务器聚合操作造成通信效率下降[6-8]。

本文关注在跨数据库联邦学习环境下客户端与服务器之间的通信效率问题。在该场景下,客户端内部通常拥有多个本地计算资源,如高性能计算机,且本地的内部通信相对于外部网络 WAN 而言具有速度更快、更可靠和可信任的优点。传统的联邦学习框架在该场景下,客户端上仅利用部分计算能力训练模型,并不能够充分地利用其计算资源训练模型。而 HFL 的结构则根据跨数据库联邦学习内部通信的特性与优点,可以很好地利用客户端本地的计算资源,通过客户端将本地的多个计算资源生成多个节点参与到全局联邦学习的训练中,形成本地节点-客户端-中心服务器的多级分层联邦学习结构,有效地利用跨数据库联邦学习的计算资源来加快训练速度,从而提高通信效率。

此外,虽然 FL 允许参与者将其原始数据保存在本地,为客户端的数据隐私提供了保护,但最近的工作表明,它不足以保护本地训练数据的隐私免受成员推理攻击[9]、属性推理攻击[10],训练过程中交换的模型参数与梯度更新仍然是重点攻击目标[10-11]。Gei 等[11]通过余弦相似性和对抗攻击策略从梯度信息中恢复训练时输入一批图像,并证明从梯度中重建输入图像与模型的深度架构无关。

与传统架构一样,HFL 架构中传输的模型参数或梯度仍面临着潜在的隐私泄露风险,不足以保护训练数据的隐私免受推理攻击及数据重构攻击。为保护 FL 系统免受这些隐私攻击,目前已有学者提出解决方案,如

Abadi 等[12]在神经网络模型训练过程中添加差分隐私噪声来消除训练数据的隐私,后续的工作在此基础上进行适应改造,将该方法移植到联邦学习系统中,如 Truex 等[13]提出 LDP-Fed,用于实现客户端能自主定义本地差分隐私预算,在客户端上传模型时添加隐私噪声,实现相对于中心式差分隐私更为优秀的隐私保护功能。Lu 等[14]提出了一种在 HFL 场景中应用差分隐私的隐私保护方案 HFL-DP,在客户端上传模型时,添加满足局部差分隐私的噪声进行扰动,并且采用 Abadi 的时刻记账方式来跟踪累计的隐私损失。又如 Moreau 等[15]将 Abadi 的方法应用至跨数据库联邦学习中,并提出了一种混合策略,即客户端根据本地数据量选择固定或自适应的隐私预算策略。然而差分隐私方案会带来噪声,随着噪声变大模型精度也逐渐降低,从而导致模型难以收敛[4]。另外一个方向为采用安全多方计算来进行隐私保护,Bonawitz 等[16]提出了一种 FL 的安全聚合方法,通过使用伪随机数、Shamir 的秘密共享[17]和对称加密来禁止服务器直接访问客户端模型。然而该方法需要可信服务器,并且需要较高的通信代价。更进一步,David 等[18]基于 Bonawitz 等人的工作,将差分隐私与安全多方计算结合,用于联邦学习的安全训练过程,即客户端向训练好的模型参数添加差分隐私噪声以及基于加密原语生成的随机数,在中心服务器进行聚合操作时,对加密模型进行聚合,即可将随机数消除得到聚合模型。但是该方法依旧向模型添加了额外的差分隐私噪声。Duan 等[19]提出了一种采用秘密共享策略的深度模型隐私保护方法,各个客户端将本地模型的梯度更新进行秘密共享,由中心服务器聚合秘密,从而得到梯度聚合结果。然而,该方案并未处理客户端掉线问题。目前的隐私保护方案中,采用安全多方计算的方案并不会在训练过程中额外添加噪声,相比于差分隐私方案其能够得到准确的模型,但是也存在相应的缺点,如需要较高的通信代价,需要依赖可信服务器与没有处理客户端掉线情况等。因此,设计一种高效且保护隐私的 FL 方案,以防止数据的隐私泄露至关重要。

在对模型精度要求更高的需求下,安全多方计算方案能更好地发挥数据的价值。秘密共享作为安全多方计算中应用场景较为广泛的方法,相比于其他安全多方计算方法而言算法实现更为简单,且在联邦学习系统中产生的代价相对较小[20],故而本文采用秘密共享方法进行保护跨数据库联邦学习中的数据隐私安全,通过优化加密过程来减少秘密共享所产生的通信代价。

因此,针对跨数据库联邦学习的通信效率问题,以及目前联邦学习隐私保护方案中存在的问题,本文提出了一种基于秘密共享的本地多节点联邦学习算法 Mask-FL。本文假设用户数据分布在不同的客户端上,例如电商平台、银行或金融机构,它们拥有大量不同的用户数据。每个客户端将生成多个本地节点,然后将用户数据划分成多份分布在本地节点上,每个客户端进行本地训练时,由本地节点采用划分的数据训练模型。在训练过程中通过秘密共享方式解决模型上行传输的隐私泄露问题。实验结果表明,该算法在保护隐私的同时具有较高的通信效率。本文的主要工作如下:

(1)提出本地多节点跨数据库联邦学习框架。在客户端 - 服务器的结构上加入本地多节点结构,每个客户端根据自身的计算资源能力生成多个本地节点,并将本地数据进行切分后设置在各个本地节点上,每个本地节点并行参与到全局联邦训练。针对节点的数据量分配问题,设计了一种基于计算能力的数据切分算法,客户端根据各节点计算能力进行数据切分,以减小数据量不平衡带来的影响。

(2)提出基于秘密共享的自适应掩码加密协议。在本地多节点跨数据库联邦学习框架的基础中,通过秘密共享的方式得到可复用的安全自适应参数掩码,客户端通过对模型添加掩码以保护模型参数安全后,再发送至服务器进行聚合。在诚实且好奇的安全设置下,证明了本协议能够对抗来自客户端与服务器的威胁。

(3)将 Mask-FL 算法用于训练卷积神经网络模型过程,通过对 Mask-FL 的各个参数进行独立实验,以及对比 3 种不同联邦学习算法,证明了本文提出的 Mask-FL 在保护隐私的前提下能保持相对较高的准确率,并且减少了全局通信轮次,有效地提高了联邦学习模型训练速度。

1　背景知识

1.1　联邦学习

联邦学习提供了使用分布式数据训练机器学习模型的能力,参与实体之间无需共享原始数据。假设(D_1,D_2,…,D_n)是分布式数据集,分别分布在 n 个用户(O_1,O_2,…,O_n)上,在联邦学习中,每一个用户都独立拥有一个数据集,并且仅使用本地的数据独立训练一个 ML 模型,而不对外部公开本地数据。每个用户通过本地训练得到的模型参数被收集到服务器(一个中心实体/机构)中,该服务器聚合所有收集到的模型参数以生成全局模型。全局模型的精度 A_{fed} 应非常接近在服务器上使用所有数据集训练得到的模型精度 A_{ctr},这种关系可用公式

(1)表示,其中,δ是一个非负实数[20]。

$$|A_{fed}-A_{ctr}|<\delta \tag{1}$$

标准的FL训练算法在多轮训练中进行,典型的联邦学习步骤如下:

(1)服务器初始模型,下发到各个客户端;

(2)每个客户端根据各自的数据训练本地模型;

(3)每个客户端将其模型权重发送到受信服务器;

(4)服务器计算模型平均权重得到共享模型;

(5)服务器将共享模型返回给所有客户端;

(6)客户端从共享模型开始,重新训练本地模型。

在提供高度准确推断的同时,保护敏感用户信息非常重要。例如输入法提供商可以使用联邦学习来提高客户输入推荐词的精确度。各提供商不必采集客户设备上的隐私输入词来训练自己的推荐算法,而是结合其模型创建共享的高频词推荐机制,无需共享其个别客户的隐私输入词。然而,恶意方仍然有可能通过从训练模型的权重或参数中推断出训练数据集的细节来潜在地损害个人用户的隐私[9-11]。

1.2 安全多方计算

安全多方计算理论是姚期智先生为解决一组互不信任的参与方在保护隐私信息,以及没有可信第三方的前提下,协同计算问题而提出的理论框架。目前,主要通过3种不同的框架来实现:不经意传输、秘密共享和阈值同态加密。不经意传输协议和阈值同态加密方法都使用了秘密共享的思想[20]。

秘密共享(Secret Share, SS)是指通过将秘密值分割为随机多份,并将其分发到不同方来隐藏秘密值的一种概念。每一方只能拥有一个通过共享得到的值,即秘密值的一小部分。根据不同场景,需要所有或者一定数量共享值才能重新构造原始的秘密值[17]。图1给出了如何使用秘密共享的简单示例。

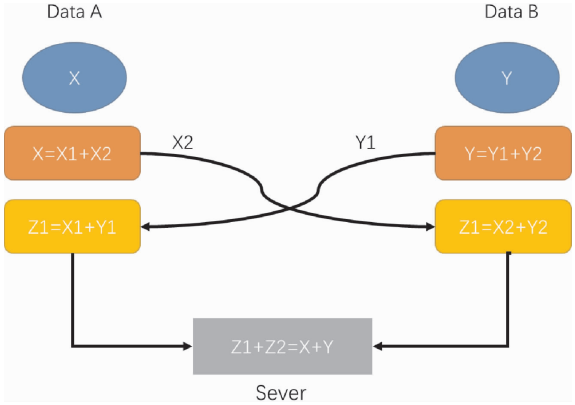


图1 秘密共享的简单示例

Fig. 1 Simple example of secret sharing

由图1所知,2个数据源分别拥有数字X和Y,服务器想要知道X+Y之和,但对X和Y一无所知。该过程可以描述如下:首先,将原始数据分解为2个子部分,一个子部分在双方之间交换,然后计算剩余子部分与另一方部分的和。最后,对计算结果进行汇总,得到原问题的解。在这个过程中,原始数据不会被公开,因此,可以在保护数据隐私的前提下完成求和运算。

2 基于秘密共享的本地多节点联邦学习算法

本章介绍本文提出的联邦学习算法Mask-FL,主要分为3个部分,第一部分为本地多节点跨数据库联邦学习训练框架,第二部分为基于秘密共享的自适应掩码加密协议,第三部分为联邦学习算法Mask-FL,将掩码加密协议嵌入本地多节点跨数据库联邦学习训练框架,并进行更为详细全面的设计。

2.1 本地多节点跨数据库联邦学习训练框架

跨数据库FL自然适合企业对企业(B2B)场景,其中每个数据库可以是公司或组织,而跨设备FL对应于企业对客户(B2C)模式。跨设备FL通常涉及大量用户,故通信成本可能是一个瓶颈,而跨数据库FL只有几个参与方(通常少于10个),因此,对于通信要求相对不大。本文基于跨数据库设置FL,在这种情况下应该考虑计算成本,因为作为企业的每一方都拥有比个人设备更为庞大的数据,并且计算能力也比单一设备要强,然而当前的联邦学习框架在数据库节点上仅仅训练单一的模型,并不能够充分地利用计算资源。虽然有分布式机器学习方法的辅助,但是在联邦学习中仍存在缺陷,即不能够直接进行快速的训练,因此,本节主要为了解决在FL中充分利用客户端的计算能力问题,设计了新型的训练结构框架LocalNodes-FL,以将数据和算力利用起来,加快联邦学习模型的训练,进而减少训练过程中的通信开销。

考虑在FedAvg联邦学习的框架上,通过改变其结构来有效提高资源利用效率和通信效率。框架如图2所示,结构采用本地节点-客户端-中心服务器的方式。通信过程存在于客户端内部、客户端与客户端、中心服务器与客户端。本地节点由于是处于同一个数据库客户端环境下,网络传输的延迟影响较小,其中的通信开销可忽略不计。而主要的通信开销产生于中心服务器与各个客户端之间。

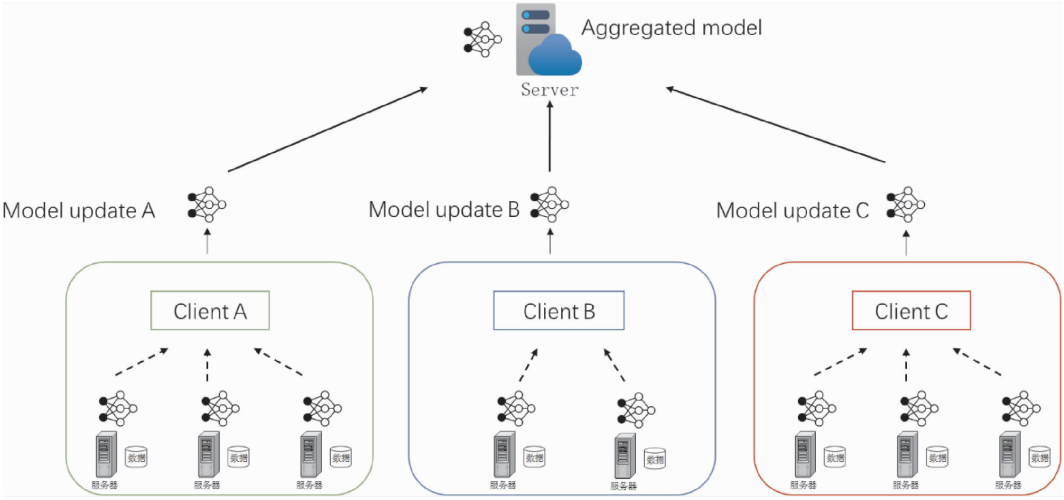


图 2　本地多节点跨数据库联邦学习示例

Fig. 2　Example of local multi-node cross-database federated learning

考虑 n 个客户端参与训练的联邦学习,本地数据集设为(D_1, D_2, …, D_n),各客户端根据本地的 m 个计算资源能力生成 m 个本地节点,将本地数据划分成 m 份并放置在本地节点上,有

$$D_i = \sum_{j=1}^{m} D_{i,j} \tag{2}$$

在本地节点训练模型过程中,各个对等节点的训练是同步进行的,那么在客户端等待收集各个节点训练完成时,最长等待时间为本地节点中训练时间最长的节点。而模型训练时间跟数据集大小成正比关系,跟节点计算能力成反比关系,因此,在数据划分时,为了能够减少客户端等待时间,本文设计了数据集切分算法——DataSplit,以让本地各个节点训练时间相近,从而能够提高整体的训练速度。伪代码如算法 1。

首先,客户端确认本地拥有的计算资源节点数量 m,从中心服务器获取初始化模型后,从本地数据集中采样 m 个 batch 大小为 B 的样本数据分配到各个节点上,每个节点采用该 batch 进行训练,记录节点训练所需时间 t_j。所有节点训练完成后,得到各个节点训练所耗时间序列。计算节点计算能力系数

$$c_j = \frac{t_1}{t_j * \sum_{k=1}^{m} \frac{t_1}{t_k}} \tag{3}$$

客户端根据每个节点的计算能力系数进行划分数据集

$$|D_j| = c_j |D| \tag{4}$$

在进行数据集划分时,各个节点所分配的数量为 $|D_j|$,但是划分的数据是从客户端数据集中随机采样的,并且每个节点内的样本都不重复。

算法 1 基于计算能力的数据切分算法——DataSplit

输入:客户端数据集 D,计算资源节点集 M,服务器下发模型 w_0

输出:节点数据集 $|D_j|$

Client executes:
  sample m batches from the dataset and distribute them to each node
  for each node $j \in M$ in parallel do
    $w_1 \leftarrow ClientUpdate(j, w_0)$
    Statistic time t_j　//统计训练时间
  End for
  calculate dataset rate of nodes:
  $c_j \leftarrow \frac{t_1}{t_j * \sum_{k=1}^{m} \frac{t_1}{t_k}}$
  for each node $j \in M$ do:
    $|D_j| = c_j |D|$
  End for

2.2　基于秘密共享的自适应掩码加密协议

在联邦学习训练过程中,按照参与训练的主体划分,存在着来自非诚实服务器以及其他参与训练方的威胁,所以为了解决来自这些主体的威胁,本节在 Local-Nodes-FL 框架上设计基于秘密共享的自适应掩码加密协议,加入了联邦加权平均算法的思想。该协议用于一组固定的客户端(P_1, P_2, …, P_n),还有一台服务器 S 的深度神经网络交互训练。

安全假设:①假设所有客户端与服务器都采用安全的通道进行通信,如 TLS/SSL;②所有客户端和中心服务器都是诚实且好奇的[21],诚实且好奇的定义如下:诚

实且好奇的客户端和中心服务器会根据协议执行相应步骤,但是,它们也会尝试推断出其他客户端的隐私数据;③每个客户端至少生成 $m(m\geqslant 2)$ 个节点参与联邦学习过程。

为了向服务器隐藏每个客户端的模型权重,协议采用安全多方计算技术中的秘密共享技术,在该协议中,客户端协同工作,以加密的方式向服务器发送各自的模型参数或者模型更新梯度。在模型从中心服务器下发到客户端后,每个客户端生成一个与模型参数形状相等的掩码 Mask,例如对于一个具有 10 M 个参数的模型,那么也相应生成一个 10 M 个参数的掩码,即

$$shape(Mask)=shape(w) \tag{5}$$

由于模型参数与掩码值可能相差过大,进而会出现一种危机:服务端接收到本地掩码模型后,通过比对原始模型,判断掩码模型参数是否出现异常值,那么将异常值去掉突出部分得到原始模型参数,某种程度上也会反映出本地真实模型,因此,本文提出自适应的掩码生成方案,将生成的掩码值界限设置为$[\min(w),\ \max(w)]$,使得模型值与掩码值在同一范围区间内,进而消除掩码与模型参数之间的差距,从而更好地防止服务器从客户模型中推断私人数据。模型掩码采用伪随机数发生器进行生成:

$$Mask=PRG(a)^{z} \tag{6}$$

其中,z 代表 $PRG(\cdot)$的输出维度(在本协议中,其维度等于模型的参数数量),$PRG(\cdot)$是伪随机数生成器,输出空间限定为$[\min(w),\ \max(w)]$,a 为模型参数值。

协议如图 3 所示,该图为 3 个客户端参与的掩码交换过程。

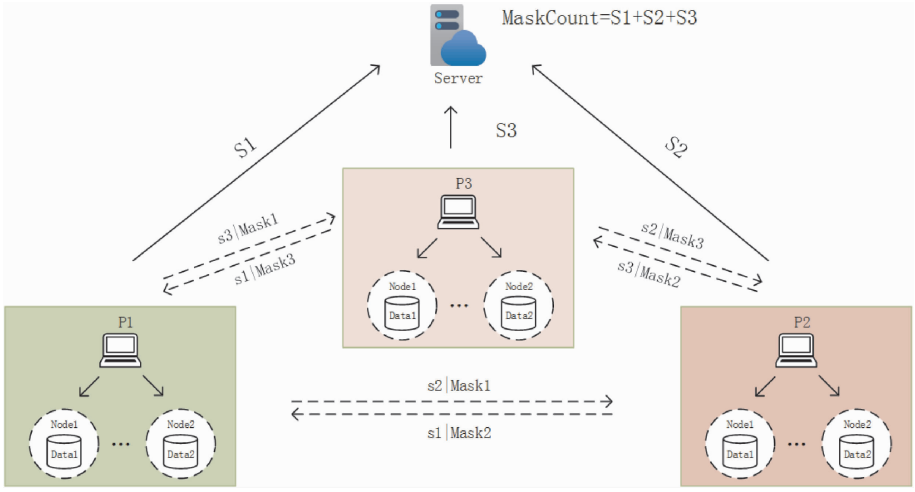


图 3 三方参与掩码加密协议过程

Fig. 3 Three parties participate in the mask encryption protocol process

协议具体流程如下:

(1)节点生成掩码

自适应掩码加密的目标是保护单个客户端的模型参数,每个参与训练的节点按照式(6)生成本地掩码。

(2)进行秘密共享求和

每个客户端的各个本地节点都生成本地掩码后,进行求和,即

$$Mask_i=\sum_{j=1}^{m}Mask_{ij} \tag{7}$$

得到客户端掩码和,每个客户端对掩码和执行秘密共享协议,将 $Mask_i$ 分别拆成 n 份 $Mask_i=\{s_{i,1},\ s_{i,2},\ \cdots,\ s_{i,n}\}$,分发给 n 个客户端。客户端收到各个客户端发来的部分秘密后,对部分秘密进行求和,即

$$S_i=\sum_{j=1}^{n}s_{j,i} \tag{8}$$

得到部分秘密和,接着将其发送到中心服务器。

(3)服务器计算掩码和

中心服务器接收来自各个客户端发送的部分秘密和之后,计算掩码和

$$MaskCount=\sum_{i=1}^{n}S_i \tag{9}$$

(4)模型添加掩码

节点在本地训练完模型后,向模型添加权值$|D_{i,j}|$,其代表客户端划分到本地节点的数据集大小,再向模型 w 添加自适应掩码。即对模型参数 w 做如下处理:

$$\widehat{w_{i,j}}=|D_{i,j}|*w_{i,j}+Mask_{i,j} \tag{10}$$

然后客户端将本地掩码模型聚合:

$$\widehat{w_i}=\sum_{j=1}^{m}\widehat{w_{i,j}} \tag{11}$$

将本地掩码模型发送至中心服务器,$|D_{i,j}|$为局部

节点划分到的数据集大小,并且客户端发送$|D_i|$至中心服务器。通过对模型参数添加自适应掩码,从而保护客户端的训练模型参数在交互过程中不被泄露。

(5)模型解码

中心服务器收到每个客户端发来的掩码模型和数据集值,然后执行模型解码:

$$Rate = \sum_{i=1}^{n} |D_i| \tag{12}$$

$$w` = \sum_{i=1}^{n} \widehat{w_i} - MaskCount \tag{13}$$

$$w` = \sum_{i=1}^{n}\sum_{j=1}^{m}(|D_{i,j}| * w_{i,j} + Mask_{i,j}) - \sum_{i=1}^{n}\sum_{j=1}^{m} Mask_{i,j} \tag{14}$$

$$w` = \sum_{i=1}^{n}\sum_{j=1}^{m}(|D_{i,j}| * w_{i,j}) \tag{15}$$

$$w = \frac{w`}{Rate} = \frac{1}{\sum_{i=1}^{n}|D_i|}\sum_{i=1}^{n}\left(|D_{i,j}| * \sum_{j=1}^{m} w_{i,j}\right) = \frac{1}{\sum_{i=1}^{n}\sum_{j=1}^{m}|D_{i,j}|}\sum_{i=1}^{n}\left(|D_{i,j}| * \sum_{j=1}^{m} w_{i,j}\right) \tag{16}$$

得到聚合模型。

本文协议能够让客户端在每次模型迭代中重用掩码,因此,客户端在协议开始时只需要与服务器和其他客户端进行一次通信。在随后的迭代中,每个客户端只需与服务器通信,当然,随着训练的轮次增大,模型参数的最值区间改变,那么相应的,掩码值也应更新,并且掩码值定期更新能为系统带来更高的安全性。

协议安全性分析:

本节证明了在诚实且好奇的客户端和中心服务器参与攻击下,本协议是安全的。

定理 1(抵御服务器攻击)　服务器不能推断出节点的模型参数。

证明　在上述联邦学习训练过程中,服务器可以获得客户端上传的加权聚合掩码模型。假设服务器可以推断出$w_{i,j}$,那么服务器需要推断出节点的参数掩码和权值。首先,推断出基于a的伪随机数生成的掩码$Mask_{i,j}$,a是节点的参数,服务器需要与节点进行相互串谋才能获取a,这与该攻击方式相矛盾。故而服务器无法推断出节点的模型掩码$Mask_{i,j}$;其次,服务器破解权值$|D_{i,j}|$,由于假设③,$|D_{i,j}| \neq |D_i|$,则需要推断出客户端给该节点分配的数据量,而客户端在本地内进行切分的方法是根据节点的计算能力进行分配的,服务器需要与客户端进行合谋才能破解,节点是属于客户端的,这与该攻击方式相矛盾,因此,服务器无法推断出节点的权值$|D_{i,j}|$。

综上所述,在中心服务器攻击中,服务器无法推断出节点的模型参数。

定理 2(抵御节点攻击)　恶意客户端不能推断出节点的模型参数。

证明　假设恶意客户端在每次训练迭代过程中,根据聚合模型结果推断出其他客户端的本地节点模型参数,但是在协议中,模型聚合结果是由每个客户端本地节点的模型加权聚合平均得到的,那么恶意客户端需要推断其他客户端生成的本地节点个数m_i及划分到本地节点的权值$|D_{i,j}|$。首先,推断所有其他客户端生成的本地节点个数m_i,m是每个客户端根据本身计算资源生成的节点,其他客户端并不知晓,恶意客户端需要与所有客户端相互勾结才能获取m_i,这与该类攻击方式不相符,故恶意客户端无法推断出其他客户端生成的本地节点个数m_i;其次,恶意客户端推断本地节点的权值$|D_{i,j}|$,$|D_{i,j}|$是每个客户端的隐私数据集大小,只有其本身知晓,恶意客户端需要所有客户端相互勾结才能获取,与该类攻击方式不符,因此,恶意客户端无法推断出其他客户端的本地节点权值$|D_{i,j}|$。综上所述,无法从聚合结果推断出其他客户端本地节点的模型参数。

定理 3(抵御$h \leqslant n-1$个客户端和服务器的共谋攻击)　服务器和客户端不能推断出其他客户端本地节点的模型参数。

证明　首先从服务器角度证明。假设中心服务器可以推断出其他客户端本地节点的模型参数$w_{i,j}$。服务器收到来自客户端发送的加权聚合掩码模型,服务器需要推断出本地节点添加的掩码$Mask_{i,j}$和权值$|D_{i,j}|$,服务器要求与之勾结的h个客户端上传各个节点的掩码值,服务器拥有$MaskCount$,当$h = n-1$时,则仅能推断出客户端内所有本地节点的模型掩码和,无法获得本地节点的单一模型掩码。通过消除加权聚合掩码模型的掩码,得到客户端的加权聚合模型,而服务器要破解本地节点的权值$|D_{i,j}|$,需要与客户端进行合谋,与攻击方式不符。当$h < n-1$时,无法推断出客户端的模型掩码和。所以,服务器不能推断出其他节点的模型参数。

其次,从客户端角度证明。①客户端可以从聚合结果进行推断。假设h个客户端相互合谋,从聚合结果推断出其他节点的模型参数。h个客户端需要推断其他客户端生成的本地节点个数m_i以及划分到本地节点的权值$|D_{i,j}|$。与服务器合谋获知其他客户端发送的客户端

总权值$|D_i|$,然而h个客户端仍需与所有客户端进行勾结才能获取$|D_i|$和m_i,与该攻击不符,故客户端不能从聚合结果中推断出节点的模型参数。②客户端可以从加权聚合掩码模型中进行推断。与上述服务器角度证明相同,故客户端不能推断出其他客户端本地节点的模型参数。综上所述,诚实且好奇的客户端无论从聚合结果进行推断还是从客户端上传的结果进行推断,都不能推断出其他客户端的节点模型参数。

总之,服务器和客户端不能推断出其他客户端本地节点的模型参数。以上3个定理证明了基于秘密共享的自适应掩码加密协议能够有效地抵御服务器攻击、客户端攻击以及服务器与客户端相互合谋的共谋攻击。因此,本文提出的结合自适应掩码加密的本地多节点联邦学习隐私保护方案是安全的。该方案能够保证诚实且好奇的服务器和客户端都不能获取其他诚实客户端的隐私数据。

2.3 Mask-FL

集成本地多节点训练框架和自适应掩码加密协议,考虑客户端掉线问题与全联邦学习训练流程,在本节给出完整的Mask-FL联邦学习算法。训练分为2个阶段,第一阶段为初始化阶段,第二阶段为联邦学习训练阶段。具体如算法2所示。

算法2 Mask-FL

输入:客户端$\{P_1,\cdots,P_n\}$,服务器S,初始模型参数w_0,全局训练轮次R,掩码更新阈值T,本地训练迭代次数E,客户端数据集$\{D_1,D_2,\cdots,D_n\}$,客户端本地节点集$\{m_1,m_2,\cdots,m_n\}$

输出:结果模型w

S initialization model w_0 and deliver the model to each client.
Mask-FL. Setup:
  for each client $i \in P_n$ do:
    $\{D_{i,j}\},\{w_{i,j}\} \leftarrow DataSplit(D_i, w_0)$ //详见算法1
    $Mask_{i,j} \leftarrow PRG(w_{i,j})$ //采用伪随机数生成器生成与模型相同的模型掩码
    $Mask_i \leftarrow \sum_{j=1}^{m} Mask_{i,j}$
    Split $\{s_{i,1}, s_{i,2}, \cdots, s_{i,n}\} \leftarrow Mask_i$
    Send share $s_{i,n}$ to P_n
    receive share and merge them:
      $S_i \leftarrow \sum_{j=1}^{n} s_{j,i}$
    Send S_i, $|D_i|$ to server
  End for
Server do:
  $MaskCount \leftarrow \sum_{i=1}^{n} S_i$
Mask-FL. Train:
  While $r \leqslant R$ do:
    for each client $i \in P_n$ in parallel do:
      for each node $j \in m_i$ in parallel do:
        $w_{r+1}^{j} \leftarrow ClientUpdate(j, w_r)$, do it for E time
        $\widehat{w_{r+1}^{j}} \leftarrow |D_{i,j}| * w_{r+1}^{j} + Mask_{i,j}$
      End for
      $\widehat{w_{r+1}^{i}} = \sum_{j=1}^{m} \widehat{w_{r+1}^{j}}$ and send to server
    End for
    Server do:
      $\widehat{w_{r+1}} \leftarrow \sum_{i=1}^{n} \widehat{w_{r+1}^{i}}$
      $w_{r+1} \leftarrow \frac{1}{\sum_{i=1}^{n} |D_i|}(\widehat{w_{r+1}} - MaskCount)$
      Send w_{r+1} to all clients
      $r++$
      if $r \% T == 0$ or client dropout:
        Run Mask-FL. Setup (w_{r+1})
  End while
ClientUpdate(k, w): // Executed on client k
  for each local epoch i from 1 to E do:
    batches$\leftarrow$(data D split into batches of size B)
    for batch b in batches do:
      $w \leftarrow w - \eta \nabla((w;b)$
    End for
  End for

一个完整的训练周期如下:

(1)初始化模型参数。中心服务器初始化模型,并且生成模型初始化参数,分别为客户端本地节点的训练迭代次数、全局通信轮次以及学习率,并且将模型和这些参数发送至所有参与训练的客户端,客户端根据本地计算资源能力将本地数据集进行随机划分,每一个计算资源生成一个节点并拥有一个划分后的数据集。

(2)节点采用一个batch的隐私数据集进行单次训练,每个节点根据模型参数生成模型掩码,采用秘密共享机制与其他节点分享模型掩码秘密,各节点再将获得的掩码秘密求和后发送到服务器,服务器结合各部分掩码秘密和获得总掩码。

(3)节点采用隐私数据集进行本地训练,获得模型参数后加权,并且添加本地掩码得到掩码模型,发送至客户端,客户端聚合后再上传至中心服务器并且发送客户端本地参与训练的数据集数量。

(4)中心服务器检测是否有离线客户端,①没有离线客户端,直接将所有客户端上传的掩码模型相加并采用总掩码和解密后进行平均得到全局模型;②存在离线客户端,则其他客户端重启步骤 2 ~ 步骤 5,然后计算全局模型,下发全局模型至各个客户端。当全局迭代次数达到掩码更新阈值时,各个客户端按照步骤 2 更新掩码。

如果训练过程中,有客户端中途退出,则采用步骤 2 重新生成新的掩码并将其保存下来,若在下一次迭代掉线客户端重新上线,则可启用上一次掩码值而不用重新运行步骤 2。反复迭代,直到模型收敛或达到最大训练轮数。

3　实　验

在本章中对 Mask-FL 进行实验以评估其性能指标,同时设置对照实验组:联邦平均算法(FedAvg)[3]、结合差分隐私的联邦学习(DP-FL)[12]、结合秘密共享方案的联邦学习(SS-FL)[19]。为了进行相同背景的对比,3 个对照实验组的超参数、模型及数据集都设置成与 Mask-FL 一致。

本文在分布式数据集上使用深度神经网络 AletNet 进行训练来模拟,模型参数数量为 3.87 M。实验数据集采用 MNIST 手写图像数据集,该数据集由 28 * 28 像素的 60 000 张训练图片和 10 000 张测试图片组成,一共 10 个数字类,其中,每类各有 6 000 张训练集和 1 000 张测试集。实验运行环境为一台配有 Tesla P100 PCIe 16GB GPU 的 PC,内存为 32 GB。各节点训练模型使用 SGD 作优化器。

在实验中默认训练数据集 batch 为 64,测试集 batch 为 1 000,学习率为 0.01,全局训练轮次 round 为 100。对于 Mask-FL 中客户端的本地节点进行本地迭代训练次数 epoch 设置为 1,其他框架的客户端本地训练次数 epoch 为 1。在本文 DP-FL 对照实验中,设置隐私预算松弛度 $\delta = 1e-5$,训练抽样集比例 $q = 0.01$,clip = 8,由于抽样集比例为 0.01,为能与其他算法在同等情况下进行比较,控制其模型训练的样本数与其他算法相等,因此,设置 DP-FL 的本地训练轮次 epoch = 100。

Mask-FL、FedAvg、SS-FL 和 DP-FL 均采用 1 个中心服务器、3 个客户端的结构进行训练。对 60 000 张图片的 MNIST 训练集进行打乱后随机采样切分成 3 个训练集[20 000,20 000,20 000],每个客户端拥有一个训练集参与联邦学习训练。测试集只放置在中心服务器,在一个轮次训练结束后,中心服务器使用测试集进行测试以观察聚合模型的效果。

本文分别从协议通信成本、训练时间、模型训练精度等角度对 Mask-FL 进行评估与分析。

3.1　协议时间消耗分析

通过实验来评估协议,可以构建一个精确的过程,以确定在实际场景中运行协议需要多长时间。为了实现这一点,本文统计了 Mask-FL 各个节点及服务器交流过程中协议各个步骤的计时结果,包括节点上的掩码初始化平均时间、掩码交换时间、加密平均时间、训练平均时间,服务器上的模型聚合平均用时、解码平均用时、模型下发平均用时,并且统计了 FedAvg 算法各个训练过程的平均耗时。虽然实验是单线程的,但它确实跟踪每个节点每个动作的独立时间,并确保实验不允许在同一时间内执行多个活动。因此,可以断言这些时间应该是对协议完整、分布式实现的合理估计。

实验采集了 100 个通讯轮次 FedAvg 和 Mask-FL 的各阶段平均用时,如图 4 所示。对于 Mask-FL 算法,基于秘密共享的自适应掩码协议的掩码初始化过程耗时为 11.027 9 ms,在训练过程中,模型在本地训练的耗时为1 056.182 6 ms,相比之下,本协议对模型的加解密过程所耗费的时间仅为 2.771 4 ms,说明本协议加密过程所产生的耗时并未对 FL 系统产生明显的时间开销。并且通过 2.2 节中对协议进行的安全性证明分析,说明基于秘密共享的自适应掩码协议消耗的时间不仅极小,而且能够有效地保护客户端数据隐私安全。除此之外,从图中可以看出,Mask-FL 客户端的模型训练时间大大缩小,证明本文提出的本地多节点结构设计能够有效地减小客户端模型训练的时间。

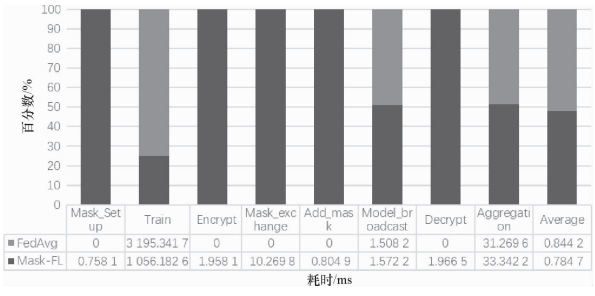


图 4　Mask-FL 和 FedAvg 训练中各阶段耗时

Fig. 4　Time-consuming stages in Mask-FL and FedAvg training

3.2　通信成本评估

由于本次实验是在单机上运行的,所以并未对客户

端-客户端、客户端-服务器的上下行传输时间进行模拟,并且在实际应用中,各种设备的通信带宽、数据传输效率和网络状态各不相同,所以在联邦学习上下行过程中,考虑的评价指标为通信数据量。本文对各种联邦学习安全协议的训练过程进行了数据传输量的比较。

表1比较了4种联邦学习协议全局训练过程中的通信成本,$|w|$为传输的模型大小,n为参与训练的客户端,k为服务器聚合模型次数。相比于其他协议,Mask-FL协议需要在训练开始进行初始化设置,在该过程中所有客户端间需传输固定数据量$(n-1)*n*|w|$,客户端与服务器需传输$3|w|$数据量。在整个训练流程中,初始化产生的时间损耗仅发生一次,而在训练中掩码的更新频率T小于全局训练次数k,其数据量相对于整个训练过程的数据量仅占少部分,考虑协议安全性与时间损耗,Mask-FL仍具有很大的优势。

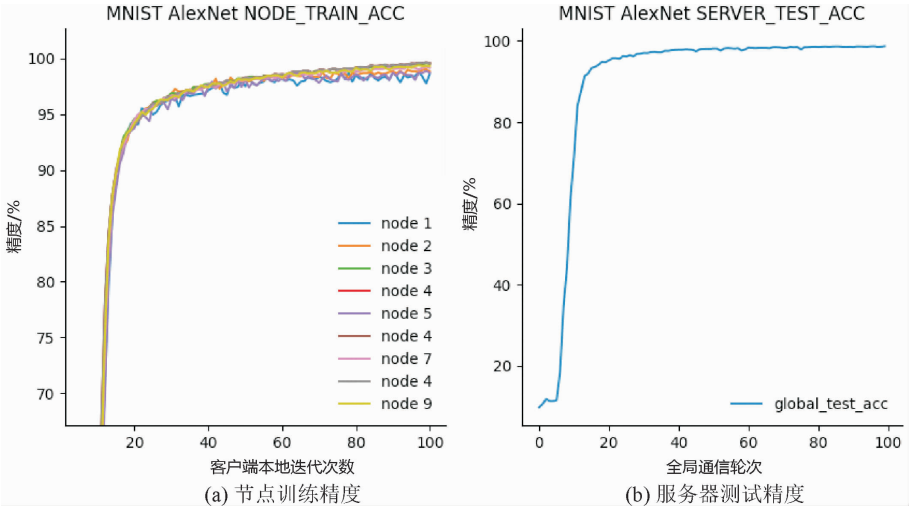
表1 不同联邦学习算法之间的通信成本比较

Table 1 Comparison of communication costs between different federated learning algorithms

方法	初始化		训练	
	Node-node	Node-server	Node-node	Node-server
Mask-FL	$(n-1)n\|w\|$	$3\|w\|$	-	$2k*n*\|w\|$
SS-FL	-	-	$(n-1)kn\|w\|$	$2k*n*\|w\|$
DP-FL	-	-	-	$2k*n*\|w\|$
FedAvg	-	-	-	$2k*n*\|w\|$

3.3 模型准确性分析

构造1个中心服务器、3个客户端及每个客户端3个节点的结构,每个客户端拥有20 000张数据,由于在相同的机器上训练,因此,按照Datasplit数据切分算法划分,得到每个客户端节点中的数据为[6 666,6 666,6 666]。按照以上采用默认参数设置,每个节点本地迭代训练1epoch,全局通信轮次为100,进行Mask-FL训练,训练结果如图5所示。



(a) 节点训练精度 (b) 服务器测试精度

图5 Mask-FL的模型训练精度

Fig. 5 Model training accuracy of Mask-FL

由于采用的是平均聚合方式,所以在模型下发各个节点后,再次进行本地训练导致训练精度突然改变,从而出现了图5(a)中的毛刺,9个参与训练的节点的训练效果各有差异,但是在联邦学习的平均聚合效果下,最终都能达到整体的最优精度,在图5(b)中,随着全局迭代的不断增加,模型逐渐收敛,最大准确率达98.69%,即Mask-FL充分地利用各个节点数据达到训练目标。

由于Mask-FL客户端的训练速度提高,那么在相同的通信轮次内,本地节点可进行更多本地迭代训练的次数,根据上述实验,Mask-FL客户端训练时间约为FedAvg客户端训练时间的1/3,因此,设置节点本地迭代次数为FedAvg的3倍以进行相同条件的对比。训练结果如图6所示。

图6中,在相同的训练时间条件下,Mask-FL训练模型的收敛速度比FedAvg显著提高。达到相同的98%准确度条件下,Mask-FL需要进行14次通信,FedAvg需要进行20次通信,相比之下本方案所需的通信轮次更低,提高训练效率比为30%。Mask-FL最大收敛准确率为98.85%,FedAvg为98.83%,2种方案训练的模型均收敛。

在默认参数设置下,针对Mask-FL不同的本地迭代训练次数进行实验,将epoch设为1、3、5、10分别进行训练,研究节点本地迭代训练次数对训练结果的影响,见图7。

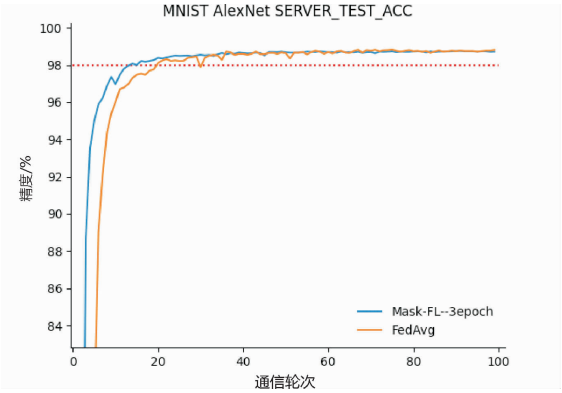


图 6　Mask-FL 本地节点训练 3 epoch 与 FedAvg 训练模型精度比较

Fig. 6　Comparison of the accuracy of Mask-FL local node training 3 epoch and FedAvg training model

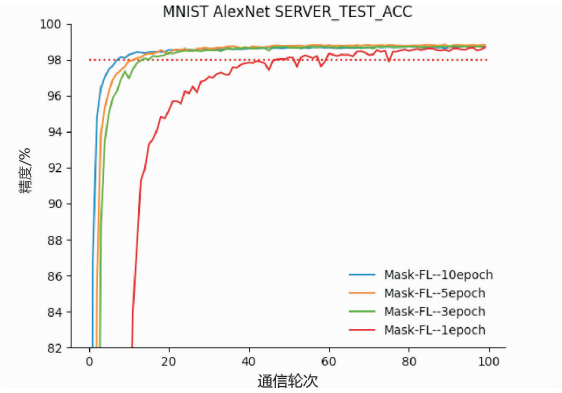


图 7　Mask-FL 中不同 epoch 对训练的影响

Fig. 7　Effects of different epochs on training in Mask-FL

图 7 为训练模型达到 98% 精度,Mask-FL 在本地迭代 1epoch 方式需要进行通信 47 round,3 epoch 方式需要进行通信 14 round, 5 epoch 方式需要进行通信 12 round,10 epoch 方式需要进行通信 8 round。1 epoch 方式达到最大精度为 98.69% ,3 epoch 方式最大精度为 98.85% ,5 epoch 方式最大精度为 98.85% ,10 epoch 方式最大精度为 98.86% 。随着本地节点迭代次数的增加,模型能更快地收敛,从而联邦学习训练收敛时所需的通信轮次更少。

采用默认参数设置,针对 Mask-FL 每个客户端生成的节点数分别为 1、2、3、5 进行实验,分析客户端节点数对模型训练的影响,结果见图 8 及表 2。

如图 8 所示,在这次实验中 Mask-FL 节点数为 1 时,等价于 FedAvg 方法,但是不符合 Mask-FL 的安全性假设。随着客户端节点数的增加,模型收敛时所需的通信轮次增多,训练时间如表 2 所示,训练耗时大大降低,并且能够达到收敛精度。考虑上述实验图 6 的结果,可在提高客户端节点数的同时,提高节点本地迭代次数,以此来提高训练效果。

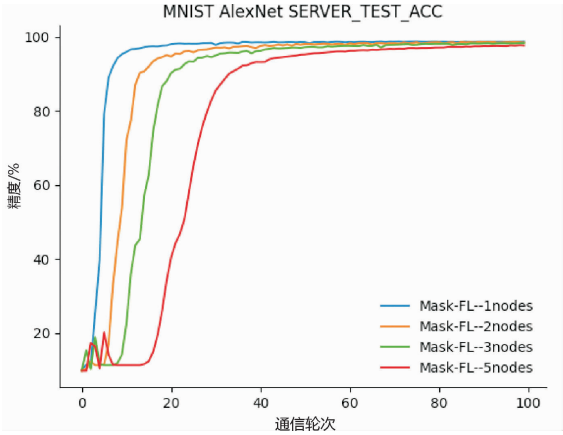


图 8　Mask-FL 客户端生成不同节点数对训练的影响

Fig. 8　The effect of different number of nodes generated by the Mask-FL client on training

表 2　Mask-FL 客户端多节点训练 100 round 耗时和精度

Table 2　Time-consuming and accuracy of Mask-FL client multi-node training for 100 round

Mask-FL	训练总耗时/s	平均精度/%	最大精度/%
nodes = 1	1 135.1	94.102 1	98.83
nodes = 2	669.1	89.339 7	98.74
nodes = 3	416.1	84.797 1	98.68
nodes = 5	283.6	76.044 2	97.80

采用默认参数设置,根据前面实验的结论节点数与训练耗时成反比,考虑在相同的时间内比较 Mask-FL 的训练效果,即客户端生成 2 节点的实验进行本地迭代 2 epoch,客户端生成 3 节点的实验进行本地迭代 3 epoch,客户端生成 5 节点的实验进行本地迭代 5 epoch。实验结果如图 9 及表 3。

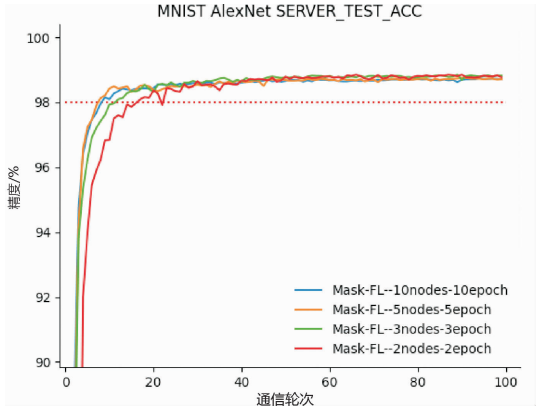


图 9　Mask-FL 多节点多 epoch 对训练的影响

Fig. 9　The impact of Mask-FL multi-node multi-epoch on training

表3 Mask-FL多节点多epoch训练100 round耗时和精度

Table 3 Time-consuming and accuracy of Mask-FL multi-node multi-epoch training for 100 round

实验	Mask-FL	训练总耗时/s	平均精度/%	最大精度/%
①	2nodes-2epoch	1 155.3	95.971 1	98.85
②	3nodes-3epoch	1 176.6	96.783 0	98.85
③	5nodes-5epoch	1 225.7	97.011 9	98.84
④	10nodes-10epoch	1 461.1	97.172 9	98.85

如图9所示,实验①达到98%精度时,所需通信轮次为17,实验②达到98%精度时,所需通信轮次为14,实验③达到98%精度时,所需通信轮次为9,实验④达到98%精度时,所需通信轮次为10。各个实验的模型通过100轮次的训练最终达到收敛。根据表3所示,得出实验结论:在近似的时间内,投入的计算资源越多,则模型训练速度越快,但是计算资源投入越多所获得的收益呈下降趋势。

采用默认设置,Mask-FL的每个客户端生成3个节点,节点本地迭代训练3 epoch,与其他3个框架进行100个通信轮次对比实验,比较模型准确度,结果见图10及表4。

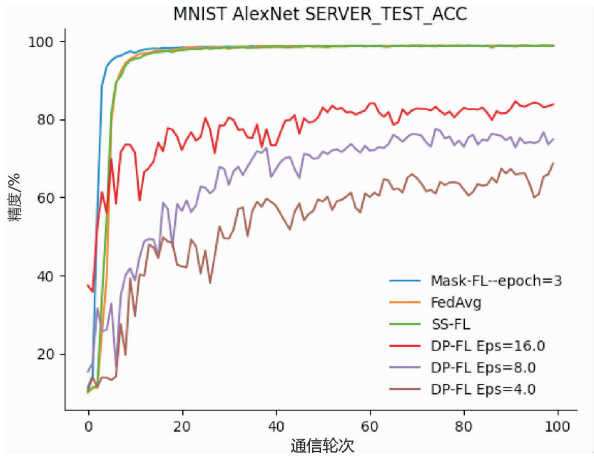


图10 多种框架训练模型100 round的准确度

Fig. 10 Accuracy of 100 rounds of training models with multiple frameworks

表4 各个框架训练100 round耗时和精度

Table 4 100 round training time and accuracy of each framework

方法	训练总耗时/s	平均精度/%	最大精度/%
Mask-FL epoch=3	1 176.6	96.783 0	98.85
FedAvg	1 128.4	94.096 6	98.83
SS-FL	1 236.1	94.252 3	98.83
DP-FL Eps=16.0	27 729.7	77.277 4	84.62
DP-FL Eps=8.0	27 334.1	64.756 3	77.54
DP-FL Eps=4.0	27 642.6	53.234 4	68.67

如图10所示,在Mask-FL、SS-FL及FedAvg训练中,模型均获得较高精度,在DP-FL中,其结果模型准确度随着差分隐私预算减小而减小,即添加的扰动过大,导致模型精度下降,模型效果变差,并且表4显示,由于在训练过程中需要进行小批次采样训练和梯度切割而导致训练时间极大增加。Mask-FL训练所需耗时与SS-FL及FedAvg相近,但实验平均精度均比其他框架高,说明Mask-FL所训练的模型收敛更快。实验证明了Mask-FL训练过程中,在保证多方安全计算的前提下没有引入新的噪声,因此,得出Mask-FL能够在深度神经网络训练过程中不会造成精度的损失,并且能够更快地收敛模型的结论。

4 总 结

针对联邦学习目前面临着成员推理、重构攻击等隐私推断攻击,以及跨数据库联邦学习各节点训练时间长的问题,本文将分层联邦学习和基于安全多方计算的隐私保护机制相结合,提出了一种新型联邦学习训练算法Mask-FL,通过安全性分析证明,客户端的模型参数能不被服务器和其他客户端所获取。实验结果证明,相比于结合秘密共享的联邦学习,Mask-FL的通信成本较少;相比于添加差分隐私噪声的联邦学习训练方式,Mask-FL的训练准确度更高;相比于FedAvg算法,Mask-FL更安全且训练速度更快。本算法在保证数据隐私安全的前提下,拥有较高的模型准确度,以及具备优秀的模型训练速度。

参考文献:

[1] Zhuo R, Huffaker B, Greenstein S. The impact of the General Data Protection Regulation on internet interconnection[J]. Telecommunications Policy, 2021, 45(2): 102083.

[2] 潘婧. 保障网络安全维护公共利益——《中华人民共和国网络安全法》正式实施[J]. 金融电子化,2017(6):92.

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

[4] 周传鑫, 孙奕, 汪德刚, 等. 联邦学习研究综述[J]. 网络与信息安全学报, 2021,7(5): 77-92.

[5] Kairouz P, McMahan H B, Avent B, et al. Advances and open problems in federated learning[J]. Foundations and Trends in Machine Learning, 2021, 14(1/2): 1-210.

[6] Liu L M, Zhang J, Song S H, et al. Client-edge-cloud hierarchical federated learning[C]//ICC 2020 - 2020 IEEE International Conference on Communications (ICC). Piscataway: IEEE, 2020: 1-6.

[7] Luo S Q, Chen X, Wu Q, et al. HFEL: Joint edge association and resource allocation for cost-efficient hierarchical federated edge learning[J]. IEEE Transactions on Wireless Communications, 2020,19(10): 6535-6548.

[8] Abad M S H, Ozfatura E, Gunduz D, et al. Hierarchical federated learning across heterogeneous cellular networks[C]//ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Piscataway: IEEE, 2020: 8866-8870.

[9] Shokri R, Stronati M, Song C, et al. Membership inference attacks against machine learning models[C]//2017 IEEE Symposium on Security and Privacy (SP). Piscataway: IEEE, 2017: 3-18.

[10] Melis L, Song C, Cristofaro E D, et al. Exploiting unintended feature leakage in collaborative learning[C]//2019 IEEE Symposium on Security and Privacy (SP). Piscataway: IEEE, 2019: 691-706.

[11] Geiping J, Bauermeister H, Dröge H, et al. Inverting gradients—How easy is it to break privacy in federated learning[C]//Advances in Neural Information Processing Systems 33(NeurIPS 2020). Cambridge: MIT Press, 2020: 16937-16947.

[12] Abadi M, Chu A, Goodfellow I, et al. Deep learning with differential privacy[C]//Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM, 2016: 308-318.

[13] Truex S, Liu L, Chow K H, et al. LDP-Fed: Federated learning with local differential privacy[C]//Proceedings of the Third ACM International Workshop on Edge Systems, Analytics and Networking. New York: ACM, 2020: 61-66.

[14] Shi L, Shu J, Zhang W, et al. HFL-DP: Hierarchical federated learning with differential privacy[C]//2021 IEEE Global Communications Conference (GLOBECOM). Piscataway: IEEE, 2021: 1-7.

[15] Moreau M, Benkhelif T. DPSGD strategies for cross-silo federated learning[C]//2021 International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI). Piscataway: IEEE, 2021: 1-5.

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

[17] Shamir A. How to share a secret[J]. Communications of the ACM, 1979, 22(11): 612-613.

[18] Byrd D, Polychroniadou A. Differentially private secure multi-party computation for federated learning in financial applications[C]//Proceedings of the First ACM International Conference on AI in Finance. New York: ACM, 2020:1-9.

[19] Duan J, Zhou J, Li Y. Privacy-preserving distributed deep learning based on secret sharing[J]. Information Sciences, 2020, 527: 108-127.

[20] 杨强, 刘洋, 程勇, 等. 联邦学习[M]. 北京:电子工业出版社, 2020.

[21] Li H, Liu D, Dai Y, et al. Engineering searchable encryption of mobile cloud networks: When QoE meets QoP[J]. IEEE Wireless Communications, 2015, 22(4): 74-80.

【责任编辑: 卓祯雨】