计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2020.20200463
57(10): 2241-2250, 2020

基于秘密分享和梯度选择的高效安全联邦学习

董 业[1,2] 侯 炜[1,2] 陈小军[1] 曾 帅[1]
[1](中国科学院信息工程研究所 北京 100195)
[2](中国科学院大学网络空间安全学院 北京 101408)
(dongye@iie.ac.cn)

Efficient and Secure Federated Learning Based on Secret Sharing and Gradients Selection

Dong Ye[1,2], Hou Wei[1,2], Chen Xiaojun[1], and Zeng Shuai[1]
[1](*Institute of Information Engineering*, *Chinese Academy of Sciences*, *Beijing* 100195)
[2](*School of Cyber Security*, *University of Chinese Academy of Sciences*, *Beijing* 101408)

Abstract In recent years, federated learning (FL) has been an emerging collaborative machine learning method where distributed users can train various models by only sharing gradients. To prevent privacy leakages from gradients, secure multi-party computation (MPC) has been considered as a promising guarantee recently. Meanwhile, some researchers proposed the Top-K gradients selection algorithm to reduce the traffic for synchronizing gradients among distributed users. However, there are few works that can balance the advantages of the two areas at present. We combine secret sharing with Top-K gradients selection to design efficient and secure federated learning protocols, so that we can cut down the communication overheads and improve the efficiency during the training phase while guaranteeing the users' privacy and data security. Also, we propose an efficient method to construct message authentication code (MAC) to verify the validity of the aggregated results from the servers. And the communication overheads introduced by the MAC is small and independent of the number of shared gradients. Besides, we implement a prototype system. Compared with the plaintext training, on the one hand, our secure techniques introduce small additional overheads in communication and computation; On the other hand, we achieve the same level of accuracy as the plaintext training.

Key words security; privacy; secret sharing; gradients selection; federated learning

摘 要 近年来,联邦学习已经成为一种新兴的协作式机器学习方法.在联邦学习中,分布式用户可以仅通过共享梯度来训练各种模型.但是一些研究表明梯度也会泄露用户的隐私信息,而安全多方计算被认为是一种保护隐私安全的有效工具.另一方面,一些研究人员提出了 Top-K 梯度选择算法,以减少用户之间同步梯度的通信开销.但是,目前很少有工作可以平衡这 2 个领域的优势.将秘密共享与 Top-K 梯度选择相结合,设计了高效且安全的联邦学习协议,以便在保证用户隐私和数据安全的同时,减少通信开销,并提高模型训练效率.此外,提出了一种高效的方法来构造消息验证码,以验证服务器返回的聚合结果的有效性,其中,验证码引入的通信开销与梯度的数量无关.实验结果表明:相比于同样条件下的明文训练,该文的安全技术在通信和计算方面都会引入少量额外的开销,但该方案取得了和明文训练同一水平的模型准确率.

收稿日期:2020-06-10;修回日期:2020-07-28
通信作者:陈小军(chenxiaojun@iie.ac.cn)

关键词 安全;隐私;秘密分享;梯度选择;联邦学习

中图法分类号 TP391; TP181

在互联网、大数据和机器学习的推动下,人工智能技术飞速发展,语音识别、图像处理等应用逐步改变着人类的生产、生活方式.然而,在这些应用背后,大规模的用户敏感数据,如医疗数据、生理特征、社交网络等,被各类企业、机构随意收集.这些数据的收集能够带动机器学习性能的提升,实现经济效益和公众效益的双赢,但却使得个人隐私和数据安全面临更大的风险.与此同时,个人隐私和数据安全愈发受到国内外的重视关注.2017 年 6 月施行的《中华人民共和国网络安全法》①、2018 年 3 月欧盟宣布的欧盟通用数据保护条例(General Data Protection Regulation, GDPR)②都对企业处理用户数据提出了明确要求.可见,企业在用户不知情时进行数据收集、共享和分析已经被视为一种违法行为.因此,如何在保证用户个人隐私和数据安全的情况下,充分发挥机器学习等人工智能方法的潜力是一项意义深远而又亟待解决的问题.

联邦学习(federated learning)[1-3]作为一种新兴的分布式隐私保护机器学习训练框架得到越来越多的重视.在联邦学习中,分布式用户在本地根据自己的私有数据训练机器学习模型,然后借助参数服务器仅共享训练得到的梯度来提升模型的性能.联邦学习避免了用户将自己的数据暴露给企业或者其他参与方,从而在一定程度上保护了自己的隐私和数据安全.但是,联邦学习还处在发展初期,现有的方案还存在较多的不足:1)直接分享梯度的方式面临许多的隐私威胁,敌手可以针对用户的梯度推测出用户数据中的敏感信息,甚至可以逆向推演得到用户的私有数据;2)简单地将隐私保护技术,例如安全多方计算(secure multi-party computation, MPC)、同态加密(homomorphic encryption, HE)等,应用到联邦学习中可能带来巨大的、甚至无法接受的额外开销.

为了解决现有联邦学习方案的问题和不足,本文基于梯度选择和安全多方计算中的秘密分享技术,设计了高效的安全联邦学习方案,在保护用户隐私的前提下大大减小了通信开销.进一步,我们还提出了一种验证方案用来防止服务器的恶意篡改.

本文的贡献主要包括 3 个方面:

1) 结合梯度选择和秘密分享,提出了一种高效的安全联邦学习方案.和现有的工作相比,我们在保证准确性的前提下,将通信开销减少到了原来的 1%~10%.

2) 基于梯度的索引和值的数量积,我们设计了一种验证梯度聚合结果有效性的方案.在合理的安全性假设下,我们的验证方案可以抵抗服务器的恶意篡改.

3) 我们在真实的数据集上验证了方案通信的效率提升和模型的准确性.实验结果表明本文提出的机制实现了隐私保护、通信开销和模型性能三者之间的有效平衡.

1 相关工作

本节主要介绍有关梯度选择算法和联邦学习隐私保护的方案.

1.1 Top-K 梯度选择算法

Top-K 梯度选择算法在减小联邦学习通信开销方面发挥着重要作用,在学术界和工业界都受到了广泛的关注和重视.Strom[4]首先提出在训练过程中,客户端可以只上传绝对值大于某个阈值的梯度.之后,Aji 等人[5]提出了梯度丢弃算法.与 Strom 的方案不同的是,梯度丢弃算法不再依赖一个固定的阈值来选择梯度,而是将梯度按照绝对值的大小排序,然后按照从大到小的原则选择固定比例的梯度上传.进一步,Wang 等人[6]从梯度的稀疏性和方差 2 个方面综合考量来选择梯度.最近,Alistarh 等人[7]第一次在理论上给出了针对 Top-K 梯度选择算法的收敛性分析.然而,尽管 Top-K 梯度选择算法上传的只是稀疏的部分梯度值,暴露这些梯度的真实值仍然会威胁到用户的隐私安全.

1.2 联邦学习隐私保护

如引言所述,直接暴露梯度的真实值可能使得各参与方本地的数据被敌手逆向推理,从而造成隐私泄露.针对上述威胁,目前的方案主要从 2 方面考虑:1)差分隐私;2)密码学技术.

① http://www.cac.gov.cn/2016-11/07/c_1119867116.htm

② https://gdpr-info.eu/

利用差分隐私,可以在本地模型训练及全局模型整个过程中对相关参数进行扰动,从而令敌手无法获取真实模型参数[8-9].但是,与密码学技术相比,差分隐私无法保证参数传递过程中的机密性,从而增加了模型遭受隐私攻击的可能性.例如刘俊旭等人[10]针对联邦学习下差分隐私中存在的攻击方法进行了详细的调研.

目前在隐私保护联邦学习中常用的密码学技术包括安全多方计算、同态加密等.研究者们利用这些技术旨在实现联邦学习中的核心问题——安全聚合(secure aggregation).目前的方案[11-13]在不同的方面都取得了长足的进步,但是也存在一些问题亟待解决.例如文献[12-13]基于秘密分享构造了安全聚合协议,旨在保证用户梯度隐私的情况下高效地聚合梯度.其计算性能良好,但是其通信开销远远大于在明文方案和利用差分隐私的方案,这大大限制了其应用.

1.3　秘密分享

秘密分享技术是安全多方计算领域的一种常用的技术.秘密分享最早由 Shamir 和 Blakley 分别在文献[14]和文献[15]中提出,并广泛用于密钥管理、门限加密等方面.其后,针对秘密分享和恢复过程中正确性的问题,Feldman[16]构造了可验证秘密分享方案(Feldman VSS),从而防止秘密分享者和参与者作弊;进一步,为了克服 Feldman VSS 安全性对于秘密先验分布的依赖,Pedersen[17]将 Pedersen 承诺和秘密分享方案结合,构造了 Pedersen 可验证秘密分享方案(Pedersen VSS).

另一方面,加法秘密分享技术由于其高效性也受到大量研究者的关注,并被用于许多重要的安全两方(多方)计算方案,例如 Sharemind[18],SPDZ[19]等.本文出于对安全和性能的综合考虑,将采用加法秘密分享构造安全方案.

2　理论基础

本节主要介绍联邦学习、Top-K 梯度选择算法、秘密分享技术和消息验证码.

2.1　联邦学习

在联邦学习中,用户 C_i 持有本地私密数据集 D_i,所有的用户共享同一个模型架构 $\boldsymbol{\theta}$.每个用户都和参数服务器 S 建立安全信道.用户的数目记作 m.

训练阶段如图 1 所示,形式化描述如下:

1) 用户 C_i 基于本地私密数据集训练模型 $\boldsymbol{\theta}$,计算得到梯度向量 $\boldsymbol{g}_i$:

$$\boldsymbol{g}_i = Train(\boldsymbol{\theta}, D_i). \tag{1}$$

2) 用户 C_i 将 $\boldsymbol{g}_i$ 上传到服务器 S.

3) 服务器 S 聚合所有用户上传的梯度向量.在本文中采用加法聚合:

$$\boldsymbol{g}_S = \sum_{i=1}^{m} \boldsymbol{g}_i. \tag{2}$$

4) 服务器 S 将聚合结果 $\boldsymbol{g}_S$ 返回给所有用户,用户计算均值,更新本地模型:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \frac{\alpha}{m}\boldsymbol{g}_S, \tag{3}$$

其中,α 是学习率.

一轮更新完成之后,用户检测本地模型的准确率是否达到要求.如果满足要求则停止训练;否则,准备进行下一轮的训练.

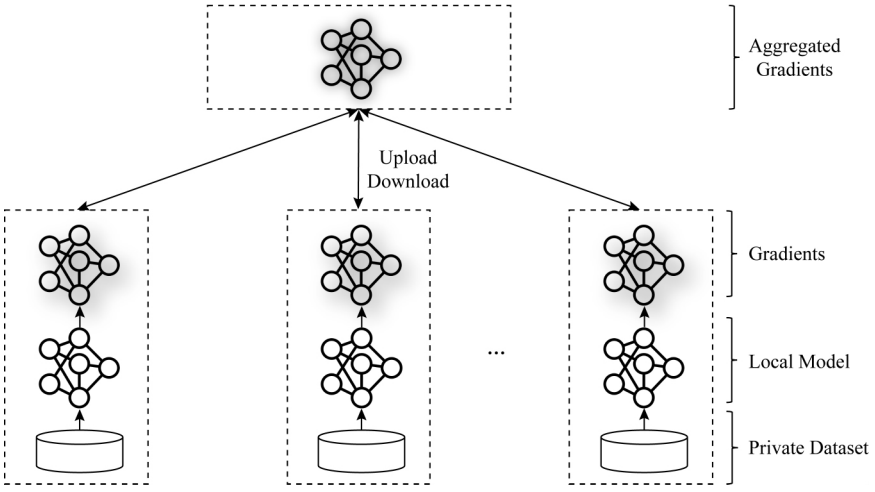

Fig. 1　The system architecture of federated learning

图 1　联邦学习系统结构

2.2 Top-K 梯度选择

在 2.1 节中,我们介绍了联邦学习的基础训练框架.在基础框架中,用户每次需要上传所有的梯度.对于大型的网络来说,上传、下载梯度所需要的通信开销可能成为系统的瓶颈.因此,梯度选择成为一个改善通信性能的有效方法.在本文中,我们参考 Aji 等人[5]提出的梯度选择方案来选择上传的 K 个梯度值.

具体来说,每次训练中用户 C 计算得到梯度向量 g 后,首先将求 g 中每个元素 $g[j]$ 的绝对值.然后根据绝对值的大小将所有 $g[j]$ 排序.排序之后,我们需要选择出绝对值最大的 K 个梯度值.然后将这 K 个梯度值上传到服务器 S.图 2 给出了梯度选择的具体描述.算法细节如算法 1 所示.

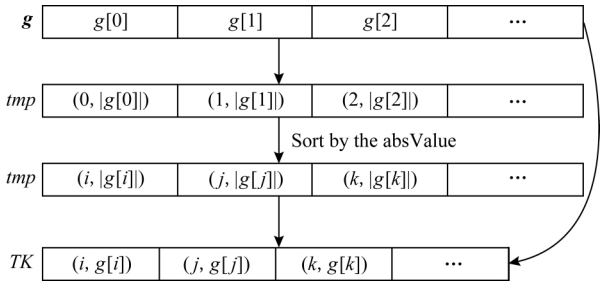

Fig. 2 Top-K gradients selection algorithm

图 2 Top-K 梯度选择算法

算法 1. Top-K 梯度选择算法.

输入:梯度向量 g,K;

输出:Top-K 梯度值和对应的索引信息.

① $tmp=[]$, $TK=[]$; /* tmp 存储梯度绝对值和索引信息, TK 存储最终选择的梯度 */

② for $g[j]$ in g

③ $tmp.append(\{j, abs(g[j])\})$; /* tmp 中的每个元素包含索引 ind 和梯度绝对值 abs */

④ end for

⑤ Sort tmp by tmp.absValue; /* 从大到小排序 */

⑥ for $d=1$ to K

⑦ $TK.append(\{tmp[d].ind, g[tmp[d].ind]\})$; /* TK 中每个元素包含索引 ind 和梯度值 */

⑧ end for

在算法 1 的行②~④,用户将每个梯度值的索引和绝对值作为一个整体存储到 tmp 中.行⑤,用户基于梯度绝对值的大小将 tmp 中所有元素降序排序.行⑥~⑧,用户依据 tmp 中排序之后的索引信息选择绝对值最大的 K 个梯度值,并将相应的索引信息作为一个整体存入 TK.最后,用户得到 TK,其中包含了绝对值最大的 K 个梯度值和对应的索引信息.

与上传所有梯度的方案相比,本文采用的方法大大减小了训练过程的通信开销;和基于梯度阈值的梯度选择方案[4]相比,本文采用的方法能将通信开销限制在固定的范围.

2.3 秘密分享

秘密分享将秘密 x 分成 n 个份额 $\{x_1, x_2, \cdots, x_n\}$,然后将 x_i 安全地保存在第 i 个秘密持有者 P_i.所有的份额满足任意少于 t 个份额无法揭示任何关于秘密 x 的信息,而任何不少于 t 个份额可以恢复原来的秘密 x,其中 $t \leqslant n$.出于对系统整体性能的考虑,在本文中我们采用高效的加法秘密分享[20],其中 $t=n$.

定义 1. 加法秘密分享包含分享算法 $Sharing$、恢复算法 $Reconst$、分享的份额数目 n.

$Sharing$:将秘密 x、份额数目 n 作为输入,输出 n 个关于秘密 x 的份额如下:

1) 随机选取 $n-1$ 个随机数

$$\{x_i \xleftarrow{R} [0, 2^l]\}_{i=1,2,\cdots,n-1}, \tag{4}$$

作为 $n-1$ 个秘密份额.

2) 计算第 n 个秘密份额:

$$x_n = x - \sum_{i=1}^{n-1} x_i \pmod{2^l}. \tag{5}$$

$Reconst$:输入 n 个秘密份额 $\{x_1, x_2, \cdots, x_n\}$,恢复秘密 x:

$$x = \sum_{i=1}^{n} x_i \pmod{2^l}. \tag{6}$$

从而,任意的少于 n 个份额不能泄露秘密 x 的任何私密信息.而任意获得 n 份秘密份额的用户便可以根据式(6)恢复秘密.

除此之外,秘密分享方案还具有加同态性质.

定理 1. 给定秘密 x 和 y 的秘密分享份额,份额持有者 P_i 可以在秘密分享的状态下计算得到秘密 $x+y$ 的分享份额.

证明. 根据定义 1,对 x 进行加法秘密分享,得到:

$$\{x_1, x_2, \cdots, x_n\} \leftarrow Sharing(x, n). \tag{7}$$

同理,对 y 的秘密分享,有:

$$\{y_1, y_2, \cdots, y_n\} \leftarrow Sharing(y, n). \tag{8}$$

对于 $\forall i \in \{1, 2, \cdots, n\}$,$P_i$ 可以在本地计算:

$$z_i = x_i + y_i \pmod{2^l}. \tag{9}$$

同时,根据式(6),有:

$$\sum_{i=1}^{n} z_i = \sum_{i=1}^{n} (x_i + y_i) = \sum_{i=1}^{n} x_i + \sum_{i=1}^{n} y_i = x + y \pmod{2^l}. \quad (10)$$

故$\{z_1, z_2, \cdots, z_n\}$是$x+y$的加法秘密分享份额.$P_i$可以根据式(9)计算$x+y$的秘密分享份额.

证毕.

2.4　消息验证码

消息验证码(message authentication code, *MAC*)是一种确认消息完整性并认证的技术[21],是一种与密钥相关联的单向 Hash 函数.消息验证码的形式化定义如下.

定义 2. 一个完成的消息验证码方案由三元组$(G, Sign, Verify)$构成.其中,G是密钥生成算法,给定安全参数κ,生成密钥$sk \leftarrow G(1^\kappa)$;$Sign$是认证算法,给定密钥$sk$和消息$x$生成验证码$MAC = Sign(sk, x)$;$Verify$是验证算法,给定验证码$MAC$、密钥$sk$和消息$x$,判断$Verify(sk, x, MAC) = \text{Accept}$是否成立;并且算法$Sign$和$Verify$需要满足:

$$Pr\begin{bmatrix} sk \leftarrow G(1^\kappa), \\ MAC = Sign(sk, x), \\ V(sk, x, MAC) = \text{Accept} \end{bmatrix} = 1. \quad (11)$$

常见的消息验证码基于分组密码和 Hash 函数构造方案,例如 HMAC[22].

本文中,我们将消息验证码的思想和梯度选择的索引信息结合,设计了具有加同态性质的验证码,从而验证聚合结果的有效性.

3　方案设计

本文基于秘密分享和梯度选择,针对联邦学习中的隐私安全和通信开销,提出了高效、安全的训练协议.具体来说,我们首先提出了针对半诚实(semi-honest 或 honest-but-curious)敌手的协议——$\pi_{\text{semi}}^{\text{ESFL}}$,基于合理的安全性假设,可以保证敌手无法获得诚实用户的梯度值.

进一步,为了抵抗被恶意敌手腐化的服务器,防止其对聚合结果进行恶意修改,保证聚合梯度的有效性(validity),我们提出了协议$\pi_{\text{mali}}^{\text{ESFL}}$.在合理的安全假设下,可以在保护诚实用户梯度隐私的前提下抵抗恶意修改攻击.

我们引入了n个($n \geqslant 2$)服务器对用户的梯度进行安全聚合.用户的数目$m \geqslant 3$.

安全假设:在协议$\pi_{\text{semi}}^{\text{ESFL}}$中最多有$n-1$个服务器被半诚实敌手腐化;在协议$\pi_{\text{mali}}^{\text{ESFL}}$中,最多有$n-1$个服务器可以被恶意敌手腐化.在 2 个协议中,我们均假设最多有$m-2$个用户可以被半诚实敌手腐化.

3.1　协议$\pi_{\text{semi}}^{\text{ESFL}}$

在协议$\pi_{\text{semi}}^{\text{ESFL}}$中,我们的目标是在半诚实敌手威胁下保护单个用户梯度的隐私性.被半诚实敌手腐化的服务器和用户,会按照协议进行训练,但是却企图通过接收到的数据推测出某个诚实用户的梯度隐私信息,进而获得其私有数据集的信息.协议流程如图 3 所示:

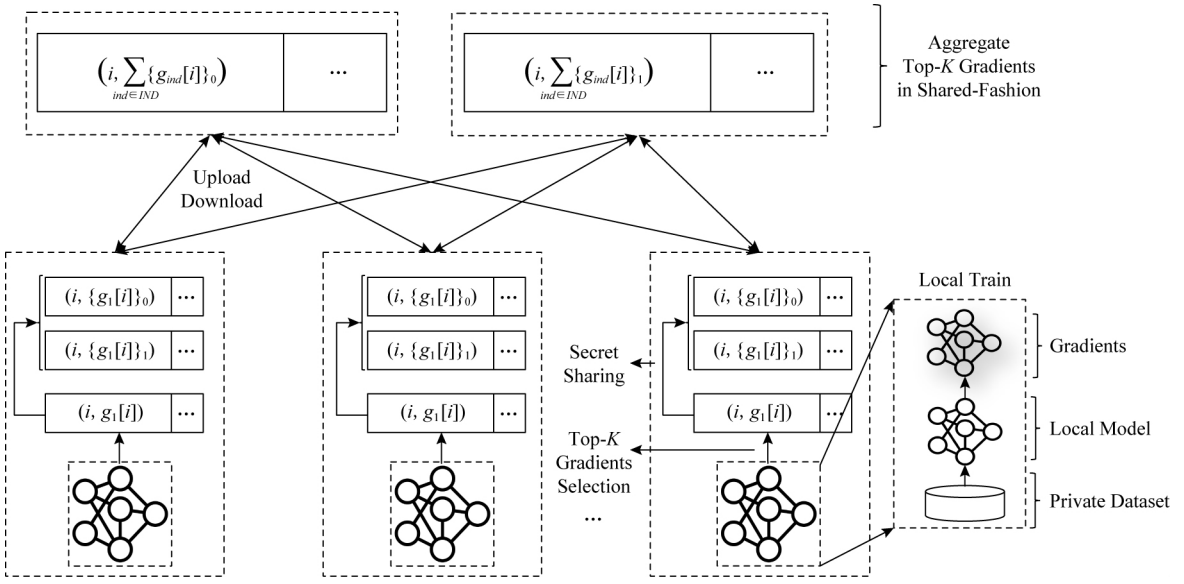

Fig. 3　The illustration of protocol $\pi_{\text{semi}}^{\text{ESFL}}$

图 3　协议$\pi_{\text{semi}}^{\text{ESFL}}$流程图

首先,用户按照式(1)训练模型 $\boldsymbol{\theta}$ 得到梯度向量 $\boldsymbol{g}$.然后,用户调用 Top-K 梯度选择算法选出绝对值最大的 K 个梯度.进一步,用户调用加法秘密分享的 $Sharing$ 算法分享选择出的梯度值.然后将分享份额上传服务器.需要注意的是,梯度的索引信息也需要上传.

另一方面,服务器在收到用户上传的梯度分享份额之后,根据索引信息将对应的份额按式(9)聚合,得到聚合结果 $\boldsymbol{g}_S$ 的秘密分享.然后将 $\boldsymbol{g}_S$ 的秘密分享和所有用户上传的索引信息的并集 IND 返回给用户.

最终,用户调用秘密恢复算法 $Reconst$ 恢复聚合结果 $\boldsymbol{g}_S$.并且,根据索引信息更新本地模型.

协议的形式化描述如算法 2 所示.

算法 2. 协议 $\pi_{\text{semi}}^{\text{ESFL}}$.

输入:用户私密数据集 D_i、统一的初始化模型 $\boldsymbol{\theta}$;

输出:训练得到的模型 $\boldsymbol{\theta}$.

① 每个用户和每个服务器之间建立安全信道;

② 每个用户在本地生成随机数;/* ①和②在预处理阶段完成 */

③ 用户在本地训练模型,计算梯度;

④ 用户调用 Top-K 梯度选择算法,选择出 Top-K 的梯度元素;

⑤ 用户针对 Top-K 梯度元素调用秘密分享,得到梯度分享份额;

⑥ 用户将索引信息和梯度分享上传到对应的服务器;

⑦ 服务器依照索引信息,根据秘密分享的加同态性质聚合梯度分享;

⑧ 用户下载服务器的聚合梯度分享份额,并调用秘密恢复算法恢复聚合梯度;

⑨ 用户更新本地模型;

⑩ 进行下一轮训练跳转③,或者停止训练.

在我们的安全性假设下,协议 $\pi_{\text{semi}}^{\text{ESFL}}$ 能够保证诚实用户的梯度隐私.

定理 2. 在不超过 $n-1$ 个服务器和 $m-2$ 个用户被半诚实敌手腐化的前提下,协议 $\pi_{\text{semi}}^{\text{ESFL}}$ 能够保证敌手无法获得关于诚实用户梯度值的任何私密信息.

证明. 首先考虑敌手腐化了 $n-1$ 个服务器.在这种条件下,敌手能够获得 $n-1$ 个梯度秘密分享份额.但根据加法秘密分享的性质,任意不超过 $n-1$ 个秘密份额无法泄露任何关于秘密真实值的私密信息.因此,在这种条件下,敌手不能获得关于梯度真实值的任何私密信息.

其次,我们考虑在敌手腐化 $m-2$ 个用户的情况.在这种情况下,敌手可以得到聚合结果的真实值.除此之外,敌手还可以得到所有腐化用户的梯度真实值.因此,敌手能够按式(12)获得所有诚实用户梯度值之和:

$$\sum_{i\in honest}\boldsymbol{g}_i=\boldsymbol{g}_S-\sum_{j\in corrupted}\boldsymbol{g}_j, \tag{12}$$

其中,$honest$ 为诚实用户的集合,$corrupted$ 为被腐化用户的集合.

基于至少有 2 个诚实用户的假设,敌手在这种情况下也无法获得针对某个用户的梯度真实值.

结合上述 2 种情况,我们可以保证在不超过 $n-1$ 个服务器和 $m-2$ 个用户被半诚实敌手腐化的前提下,协议 $\pi_{\text{semi}}^{\text{ESFL}}$ 能够保证诚实用户的梯度隐私. 证毕.

3.2 协议 $\pi_{\text{mali}}^{\text{ESFL}}$

协议 $\pi_{\text{mali}}^{\text{ESFL}}$ 能够保护用户的梯度隐私,然而却不能检测聚合结果的有效性.具体来说,如果恶意敌手腐化了某些服务器,那敌手则可以任意修改其持有的秘密份额.

假设敌手腐化服务器 S_1.记 S_1 按照式(9)聚合完成的结果为 $\boldsymbol{g}_{S,1}$.敌手生成任意的噪声向量 $\boldsymbol{r}$.之后,敌手便可以计算:

$$\boldsymbol{g}'_{S,1}=\boldsymbol{g}_{S,1}+\boldsymbol{r}\ (\text{mod}\ 2^l). \tag{13}$$

最终用户端恢复聚合结果,得到:

$$\boldsymbol{g}'_S=\boldsymbol{g}'_{S,1}+\sum_{i=2}^{n}\boldsymbol{g}_{S,i}=\boldsymbol{g}_{S,1}+\boldsymbol{r}+\sum_{i=2}^{n}\boldsymbol{g}_{S,i}=\boldsymbol{g}_S+\boldsymbol{r}\ (\text{mod}\ 2^l). \tag{14}$$

从而使得最终聚合结果受到噪声 $\boldsymbol{r}$ 的干扰,模型的训练过程和性能受到影响.

为了抵抗上述类型的攻击,我们利用梯度的索引和梯度值构造了验证码 MAC.以此检测服务器是否对聚合结果进行恶意修改,从而抵抗恶意敌手的攻击.

和协议 $\pi_{\text{semi}}^{\text{ESFL}}$ 一样,用户需要首先在本地训练和选择 Top-K 梯度.用户选择出的上传梯度记作:

$$TK=[\{ind,g[ind]\}_k]. \tag{15}$$

进一步,利用索引信息 ind 和梯度的真实值 $g[ind]$ 计算验证码 MAC:

$$MAC=\sum_{ind}ind\times g[ind]\ (\text{mod}\ 2^l). \tag{16}$$

然后,用户上传 MAC 到所有的服务器.服务器在收到所有用户上传的 MAC 之后,将对所有 MAC 求和得到 MAC_S:

$$MAC_S=\sum_{i=1}^{m}MAC_i\ (\text{mod}\ 2^l). \tag{17}$$

关于梯度值,则按照协议 $\pi_{\text{semi}}^{\text{ESFL}}$ 的方法,采用秘密分享的方式按照式(9)聚合.

聚合完成之后,服务器将聚合结果的秘密份额和其对应的索引信息 IND_S,以及 MAC_S 返回给用户.用户在本地计算恢复聚合梯度真实值 g_S,并进行验证:

1) 所有的 MAC_S 都相等;

2) 用户计算:

$$MAC'_S=\sum_{ind\in IND_S} ind\times g_S[ind]\ (\mathrm{mod}\ 2^l),\quad (18)$$

并检验 $MAC'_S=MAC_S$.

如果上述 1)2)都验证通过,那么用户则接受 g_S 并更新模型;否则,则广播错误并终止协议.具体协议如算法 3.

算法 3. 协议 $\pi_{\text{mali}}^{\text{ESFL}}$.

输入:用户私密数据集 D_i、统一的初始化模型 $\boldsymbol{\theta}$;

输出:训练得到的模型 $\boldsymbol{\theta}$.

①～② 同协议 $\pi_{\text{semi}}^{\text{ESFL}}$ ①～②;

③～⑤ 同协议 $\pi_{\text{semi}}^{\text{ESFL}}$ ③～⑤,用户计算梯度、选择梯度、分享梯度;

⑥ 用户按式(16)计算 MAC;

⑦ 服务器聚合梯度,并按式(17)对 MAC 求和得到 MAC_S;

⑧ 用户下载并按协议 $\pi_{\text{semi}}^{\text{ESFL}}$ 恢复聚合梯度,同时下载所有服务器的 MAC_S;

⑨ 用户验证所有的 MAC_S 相等,并按式(18)验证聚合结果的有效性;

⑩ 如果所有的验证通过,用户更新本地模型;否则广播错误并终止训练;

⑪ 进行下一轮训练,或者停止训练.

首先,我们分析验证过程的正确性.

1) 如果服务器没有进行任何的恶意修改,由于所有的服务器收到的 MAC 都是相同的,所以按照式(16)计算得到的 MAC_S 都相等.

2) 为了简单起见,我们首先考虑 2 个用户的情况(此处暂时不考虑隐私保护).2 个用户记作 C_1,C_2,记 C_1,C_2 选择梯度向量为 g_1 和 g_2,对应的索引集合为 IND_1 和 IND_2.根据协议,我们有 $IND_S=IND_1\cup IND_2$.

因此,对于 $\forall\, ind\in IND_S$,有 3 种情况:

$$\begin{cases} ind\in IND_1, ind\notin IND_2;\\ ind\notin IND_1, ind\in IND_2;\\ ind\in IND_1, ind\in IND_2. \end{cases}\quad (19)$$

进而,式(18)可以分解为

$$\begin{aligned} MAC'_S&=\sum_{ind\in IND_S} ind\times g_S[ind]=\\ &\sum_{\substack{ind\in IND_1\\ ind\notin IND_2}} ind\times g_1[ind]+\sum_{\substack{ind\notin IND_1\\ ind\in IND_2}} ind\times g_2[ind]+\\ &\sum_{\substack{ind\in IND_1\\ ind\in IND_2}} ind\times (g_1[ind]+g_2[ind])=\\ &\sum_{ind\in IND_1} ind\times g_1[ind]+\sum_{ind\in IND_2} ind\times g_2[ind]=\\ &MAC_1+MAC_2(\mathrm{mod}\ 2^l). \end{aligned}\quad (20)$$

另一方面,根据式(17)有 $MAC_S=MAC_1+MAC_2$.因此,$MAC_S=MAC'_S$.

上述推理很容易可以推广到多个用户的情形.

我们从隐私性和有效性 2 方面证明协议 $\pi_{\text{mali}}^{\text{ESFL}}$ 的安全性.

1) 隐私性.和协议 $\pi_{\text{semi}}^{\text{ESFL}}$ 相比,在协议 $\pi_{\text{mali}}^{\text{ESFL}}$ 中敌手的进一步获得了 MAC 的信息.根据式(16),MAC 是索引 ind 和梯度值 $g[ind]$ 的数量积.虽然敌手获得了索引信息 ind 和 MAC,但是敌手确 K 个梯度值等价于在环内求解一个 K 元一次不定方程.然而,对于一个诚实用户,每轮训练敌手只能得到一个方程,这并不能让其确定 K 个未知数.因此,协议 $\pi_{\text{mali}}^{\text{ESFL}}$ 不会泄露诚实用户的关于梯度的隐私信息.

2) 有效性.首先,基于至多有 $n-1$ 个服务器被腐化的假设,如果敌手在计算 MAC_S 的过程中加入了噪声,那么诚实用户将在检测所有 MAC_S 是否相等时检测到敌手作恶.进一步,如果敌手仅仅在计算 g_S 的过程中作恶,诚实用户也将在检验 $MAC_S=MAC'_S$ 的时候发现敌手作恶.因此,协议 $\pi_{\text{mali}}^{\text{ESFL}}$ 可以保证聚合结果的有效性.

4　实验与结果

本文在手写体数字识别数据集 MNIST① 上的卷积神经网络(CNN)进行实验. 数据集 MNIST 由 4 部分——Training Set Images, Training Set Labels, Test Set Images 和 Test Set Labels 组成.各部分如表 1 所示.

我们在 Intel Xeon E5-2650 CPU with 2.30 GHz 126 GB RAM 服务器上、在局域网内模拟 2 个参数服务器和 10 用户端.我们采用 Pytorch 作为底层的机器学习训练库,并基于 Python3 实现我们的秘密分享方案和 Top-K 梯度选择算法.在实验中我们

① http://yann.lecun.com/exdb/mnist/

迭代训练 100 次,每次训练迭代中用户端需要和服务器交互计算 1 次完成梯度的安全聚合.

Table1 An Overview of MNIST

表 1 数据集 MNIST 概览

Catalog	Number
Training Set Images	60 000
Training Set Labels	60 000
Test Set Images	10 000
Test Set Labels	10 000

卷积神经网络的结构如图 4 所示:

① Conv2d (1, 32, 3, 1)
② Conv2d (32, 64, 3, 1)
③ Dropout (0.25)
④ Dropout (0.5)
⑤ Fully Connected (9 216, 128)
⑥ Fully Connected (128, 10)

Fig. 4 The architecture of CNN

图 4 卷积神经网络结构

我们分别从模型的准确率、训练的通信开销和时间开销 3 方面分析我们的方案.

4.1 模型准确率

模型的准确率是衡量模型性能的一个很重要的指标.通过测量模型在训练过程中的准确率变化,表明我们的方案对于准确率的影响是可以接受的.因为验证部分对于模型的准确率没有影响,这一部分我们只分析明文和协议 $\pi_{\text{semi}}^{\text{ESFL}}$ 两种情况.

图 5 分别给出了在明文和协议 $\pi_{\text{semi}}^{\text{ESFL}}$ 下模型训练过程中的准确率变化.从图 5 可以得到,秘密分享过程中由于将浮点数编码为整数带来的截断误差对于模型训练的影响微乎其微.一方面,模型的准确率降低几乎是忽略不计的.例如表 2 所示,和 100% 明文训练相比,我们的安全方案在 1% 梯度选择下带来的准确率下降小于 2%;另一方面,在 2 种条件下模型的收敛速率也基本一致,模型均大约在第 60 次迭代准确率稳定在 90% 以上.

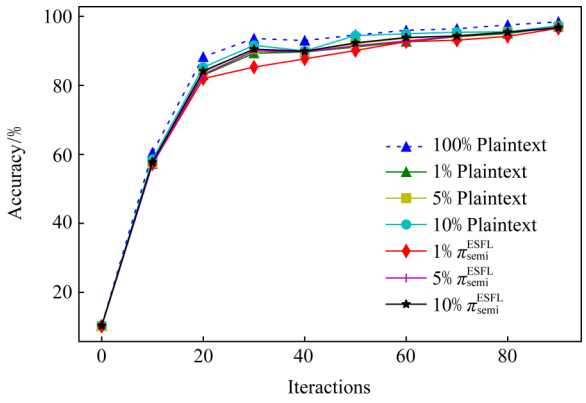

Fig. 5 The accuracy changes

图 5 准确率变化

Table 2 The Accuracy Results

表 2 模型最终准确率

Top-K/%	Protocol	Accuracy/%
100	plaintext[5]	98.41
1	plaintext[5]	96.98
	$\pi_{\text{semi}}^{\text{ESFL}}$	96.55
5	plaintext[5]	97.09
	$\pi_{\text{semi}}^{\text{ESFL}}$	96.58
10	plaintext[5]	97.28
	$\pi_{\text{semi}}^{\text{ESFL}}$	96.75

进一步,我们还探索不同的梯度选择比率对于模型训练的影响.在图 5 中,我们选择 1%,5% 和 10% 的梯度分享上传,并和明文下 100% 上传的方式比较.从图 5 和表 2 中可以看到,增加上传分享的梯度对于准确率的提升和收敛速度的提升并不显著.这与之前有关 Top-K 梯度选择工作的结论相符.

4.2 通信开销

通信开销是联邦学习的瓶颈之一,在安全的联邦学习中,由于在密文状态下无法使用压缩算法等优化技术,这个问题更为严重.为了提升通信性能,我们在 Top-K 梯度选择的基础上,结合秘密分享提出了高效的解决方案.

在实验中,每次迭代训练需要交互计算一次。我们分别将 Top-K 梯度选择的比率控制在 1%,5% 和 10%,并分别在明文、协议 $\pi_{\text{semi}}^{\text{ESFL}}$ 和协议 $\pi_{\text{mali}}^{\text{ESFL}}$ 下测量了在一次训练迭代中的通信开销.实验结果如表 3 所示:

Table 3 Communication Overhead in One Training Iteration

表 3 一次迭代训练中的通信开销

Top-K/%	Protocol	Communication/KB
1	plaintext[5]	233.045
	$\pi_{\text{semi}}^{\text{ESFL}}$	233.028×2
	$\pi_{\text{mali}}^{\text{ESFL}}$	233.065×2
5	plaintext[5]	1 089.387
	$\pi_{\text{semi}}^{\text{ESFL}}$	1 089.234×2
	$\pi_{\text{mali}}^{\text{ESFL}}$	1 089.301×2
10	plaintext[5]	2 280.128
	$\pi_{\text{semi}}^{\text{ESFL}}$	2 280.094×2
	$\pi_{\text{mali}}^{\text{ESFL}}$	2 280.164×2

实验结果表明,我们采用 Top-K 梯度选择算法,可以极大地优化在安全联邦学习中的通信开销问题.例如,我们将 Top-K 比率从 10%优化到 1%,通信性能优化了大约 9.78 倍.而 4.1 节的实验结果表明在这个范围内减少 Top-K 的选择比率并不会对模型性能造成很大的影响,因此 Top-K 梯度选择在安全联邦学习领域也具有很大的发展潜力.

另一方面,增加验证信息并没有额外增加太多的通信开销.这是因为在协议 $\pi_{\text{mali}}^{\text{ESFL}}$ 中,用户对所有的梯度信息做了一个全局的验证值 MAC.因而 MAC 所带来的开销独立于梯度量.

4.3　时间开销

在本节,我们分析在训练过程中的计算开销.针对协议 $\pi_{\text{semi}}^{\text{ESFL}}$,在线训练阶段我们测量了用户端梯度计算、分享和恢复的时间开销,传输梯度的时间开销,以及服务器端分享聚合的时间开销.我们进一步测量了预处理时间开销,主要包括建立安全连接信道和用户生成足够的随机数引入的时间开销.

对于协议 $\pi_{\text{mali}}^{\text{ESFL}}$,我们进一步在用户端加入了生成 MAC 和本地验证的时间开销,在服务器端则额外加入了聚合 MAC 的时间开销.

我们分别在 Top-K 梯度选择比率为 1%,5%和 10%下进行实验,并且与明文实验对比.结果如表 4 所示.

表 4 的实验结果表明:和明文训练对比,我们的安全协议在在线训练并没有引入太多的时间开销.用户端的计算时间和服务器的计算时间和明文训练对比,增加幅度很小,这种增幅在实际应用中是可以接受的.而在预处理阶段,通过 1%,5%和 10%的梯度选择比率对比,可以验证预处理的时间和需要的随机数的数量呈正相关.

另一方面,我们也可以观察到主要的开销来自于传输梯度的时间开销.例如,1%和 10%的梯度选择比率对比,后者需要的传输时间大约是前者的 10 倍.这也是目前安全联邦学习面临的主要性能瓶颈之一.

Table 4　The Time in One Training Iteration

表 4　一轮迭代训练中的时间开销　　s

Top-K/%	Protocol	User_time	Transfer_time	Server_time	Pre_time	Total_time
1	plaintext[5]	1.546	2.021	0.317	2.496	6.380
	$\pi_{\text{semi}}^{\text{ESFL}}$	1.603	2.198	0.334	2.695	6.830
	$\pi_{\text{mali}}^{\text{ESFL}}$	1.613	2.198	0.337	2.698	6.846
5	plaintext[5]	1.546	10.298	0.378	2.496	14.718
	$\pi_{\text{semi}}^{\text{ESFL}}$	1.615	11.786	0.398	3.231	17.030
	$\pi_{\text{mali}}^{\text{ESFL}}$	1.621	11.805	0.401	3.331	17.158
10	plaintext[5]	1.548	20.269	0.431	2.496	24.744
	$\pi_{\text{semi}}^{\text{ESFL}}$	1.627	21.380	0.431	5.331	28.769
	$\pi_{\text{mali}}^{\text{ESFL}}$	1.633	21.403	0.440	5.332	28.808

5　总　　结

联邦学习中的通信开销是其重要的瓶颈之一,而这个问题在安全联邦学习中带来的影响更为严重.本文将 Top-K 梯度选择和秘密分享的方法结合起来,在保证用户隐私和数据安全的情况下大幅提升了通信效率.进一步,我们利用索引信息构造了验证码,在保护隐私的基础上验证了服务端聚合结果的有效性,而且验证码所带来的额外通信负载是常数级别并独立于梯度的数量.这对于进一步的相关研究具有重要的参考意义.目前,也有许多工作通过对梯度量化编码的方式来提升通信性能,我们将在未来的工作中结合相关领域的方案进行下一步的探索.

参　考　文　献

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

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

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

[4] Strom N. Scalable distributed DNN training using commodity GPU cloud computing [C] //Proc of the 16th Annual Conf of the Int Speech Communication Association. Dresden: ISCA, 2015: 1488-1492

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

[6] Wangni Jianqiao, Wang Jialei, Liu Ji, et al. Gradient sparsification for communication-efficient distributed optimization [C] //Proc of the 32nd Neural Information Processing Systems. Cambridge, MA: MIT Press, 2018: 1299-1309

[7] Alistarh D, Hoefler T, Johansson M, et al. The convergence of sparsified gradient methods [C] //Proc of the 32nd Neural Information Processing Systems. Cambridge, MA: MIT Press, 2018: 5973-5983

[8] 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

[9] Liu Ruixuan, Cao Yang, Yoshikawa M, et al. FedSel: Federated SGD under local differential privacy with top-*k* dimension selection [J]. arXiv preprint, arXiv: 2003.10637, 2020

[10] Liu Junxu, Meng Xiaofeng. Survey on privacy-preserving machine learning [J]. Journal of Computer Research and Development, 2020, 57(2): 346-362 (in Chinese)
(**刘俊旭, 孟小峰. 机器学习的隐私保护研究综述**[J]. **计算机研究与发展**, 2020, 57(2): 346-362)

[11] Bonawitz K, Ivanov V, Kreuter B, et al. Practical secure aggregation for privacy-preserving machine learning [C] // Proc of the 24th ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2017: 1175-1191

[12] Corrigan-Gibbs H, Boneh D. Prio: Private, robust, and scalable computation of aggregate statistics [C] //Proc of the 14th USENIX Symp on Networked Systems Design and Implementation. Berkeley, CA: USENIX Association, 2017: 259-282

[13] Dong Ye, Chen Xiaojun, Shen Liyan, et al. Privacy-preserving distributed machine learning based on secret sharing [C] //Proc of the 21st Int Conf on Information and Communications Security. Berlin: Springer, 2019: 684-702

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

[15] Blakley G R. Safeguarding cryptographic keys [C]//Proc of 1979 Int Workshop on Managing Requirements Knowledge. Piscataway, NJ: IEEE, 1979: 313-318

[16] Feldman P. A practical scheme for non-interactive verifiable secret sharing [C] //Proc of the 28th Annual Symp on Foundations of Computer Science. Piscataway, NJ: IEEE, 1987: 427-438

[17] Pedersen T P. Non-interactive and information-theoretic secure verifiable secret sharing [C] //LNCS 576: Advances in Cryptology (CRYPTO 1991). Berlin: Springer, 1991: 129-140

[18] Bogdanov D, Laur S, Willemson J. Sharemind: A framework for fast privacy-preserving computations [G] //LNCS 5283: Computer Security (ESORICS 2008). Berlin: Springer, 2008: 192-206

[19] Damgård I, Keller M, Larraia E, et al. Practical covertly secure MPC for dishonest majority – or: Breaking the SPDZ limits [G]// LNCS 8134: Computer Security (ESORICS 2013). Berlin: Springer, 2013: 1-18

[20] Cramer R, Damgård I, Maurer U. General secure multi-party computation from any linear secret-sharing scheme [G] //LNCS 1807: Advances in Cryptology (EUROCRYPT 2000). Berlin: Springer, 2000: 316-334

[21] Bellare M, Kilian J, Rogaway P. The security of the cipher block chaining message authentication code [J]. Journal of Computer and System Sciences, 2000, 61(3): 362-399

[22] Bellare M. New proofs for NMAC and HMAC: Security without collision-resistance [G] //LNCS 4117: Advances in Cryptology (CRYPTO 2006). Berlin: Springer, 2006: 602-619

Dong Ye, born in 1995. PhD candidate. His main research interests include applied cryptography, privacy preserving, and machine learning.

Hou Wei, born in 1995. Master. His main research interests include privacy preserving and deep learning. (houwei@iie.ac.cn)

Chen Xiaojun, born in 1979. PhD, professorate senior engineer. Member of CCF. His main research interests include big data, privacy preserving and data security.

Zeng Shuai, born in 1985. PhD, engineer. Her main research interests include big data, strategy optimization, and knowledge automation. (zengshuai@iie.ac.cn)