计算机科学
COMPUTER SCIENCE

基于同态加密的神经网络模型训练方法

赵敏, 田有亮, 熊金波, 毕仁万, 谢洪涛

引用本文

赵敏, 田有亮, 熊金波, 毕仁万, 谢洪涛基于同态加密的神经网络模型训练方法[J]. 计算机科学, 2023, 50(5): 372-381.

ZHAO Min, TIAN Youliang, XIONG Jinbo, BI Renwan, XIE Hongtao. Neural Network Model Training Method Based on Homomorphic Encryption [J]. Computer Science, 2023, 50(5): 372-381.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.220300239

基于同态加密的神经网络模型训练方法

赵 敏[1,2,3] 田有亮[1,2,3] 熊金波[1,2,4] 毕仁万[4] 谢洪涛[5]
1 贵州大学公共大数据国家重点实验室 贵阳 550025
2 贵州大学计算机科学与技术学院 贵阳 550025
3 贵州大学密码学与数据安全研究所 贵阳 550025
4 福建师范大学计算机与网络空间安全学院 福州 350117
5 中国科学技术大学信息科学与技术学院 合肥 230000
(gs.zhaom20@gzu.edu.cn)

摘 要 针对云环境下数据隐私泄露与基于同态加密的隐私保护神经网络中精度不足的问题,文中提出了一种双服务器协作的隐私保护神经网络训练(PPNT)方案,在云服务器协同训练过程中实现了对数据传输、计算过程及模型参数的隐私保护。首先,为避免使用多项式近似方法实现指数和比较等非线性函数,并提高非线性函数的计算精度,基于Paillier半同态加密方案和加法秘密共享技术设计了一系列基础安全计算协议;其次,在已设计的安全计算协议基础上,构造了神经网络中的全连接层、激活层、Softmax层及反向传播相应的安全计算协议,以实现PPNT方案;最后,通过理论与安全性分析,证明了PPNT方案的正确性及安全性。性能实验结果显示,与PPMLaaS方案相比,PPNT方案的模型精度提高了1.7%,且在安全计算过程中支持客户端离线。

关键词:Paillier半同态加密;加法秘密共享;安全计算协议;隐私保护;模型训练
中图法分类号 TP309.2

Neural Network Model Training Method Based on Homomorphic Encryption

ZHAO Min[1,2,3],TIAN Youliang[1,2,3],XIONG Jinbo[1,2,4],BI Renwan[4] and XIE Hongtao[5]
1 State Key Laboratory of Public Big Date,Guizhou University,Guiyang 550025,China
2 College of Computer Science and Technology,Guizhou University,Guiyang,550025,China
3 Institute of Cryptography & Data Security,Guizhou University,Guiyang 550025,China
4 College of Computer and Cyber Security,Fujian Normal University,Fuzhou 350117,China
5 School of Information Science and Technology,University of Science and Technology of China,Hefei 230000,China

Abstract Aiming at the problem of data privacy leakage in cloud environment and insufficient accuracy in the privacy-preserving neural network based on homomorphic encryption,a privacy-preserving neural network training scheme(PPNT) is proposed for collaborative dual cloud servers,to achieve the goal of data transmission,computing security and model parameter under the collaborative training process of dual cloud servers. Firstly,in order to avoid using polynomial approximation method to realize nonlinear functions such as exponent and comparison,and improve the calculation accuracy of nonlinear function,a series of secure computing protocols are designed based on Paillier partially homomorphic encryption technology and additive secret sharing scheme. Furthermore,corresponding secure computing protocols of full connection layer,activation layer,softmax layer and back propagation in neural network are constructed to realize PPNT based on the designed secure computing protocols. Finally,theoretical and security analysis guarantees the correctness and security of PPNT. The actual performance results show that compared with the dual server scheme — privacy protection machine learning as a service(PPMLaaS),the model accuracy of PPNT improves by 1.7%,and supports the client offline in the process of secure computing.

到稿日期:2022-03-25 返修日期:2022-12-30
基金项目:国家重点研发计划(2021YFB3101100);国家自然科学基金(62272123,62272102);贵州省高层次创新型人才项目(黔科合平台人才[2020]6008);贵阳市科技计划项目(筑科合[2021]1-5,筑科合[2022]2-4);贵州省科技计划项目(黔科合平台人才[2020]5017,黔科合支撑[2022]一般065)
This work was supported by the National Key Research and Development Program of China(2021YFB3101100),National Natural Science Foundation of China(62272123,62272102),Project of High-level Innovative Talents of Guizhou Province([2020]6008),Science and Technology Program of Guiyang([2021]1-5,[2022]2-4) and Science and Technology Program of Guizhou Province([2020]5017,[2022]065).
通信作者:熊金波(jbxiong@fjnu.edu.cn)

Keywords　Paillier partially homomorphic encryption, Additive secret sharing, Secure computing protocol, Privacy-preserving, Model training

1　相关工作

基于神经网络(Neural Networks，NN)的机器学习算法已成为人脸识别[1]和语音识别[2]等领域的基石。随着云服务的不断增长，面向云服务的计算模式给用户提供了强大的数据处理平台和远程计算资源，用户和企业可将神经网络模型训练计算任务外包至云服务器，这不仅满足了用户与企业对大量数据样本训练的需求，也可节省存储与计算资源。然而，由于用户数据包含大量敏感信息，直接将用户数据传输至云服务器存在严峻的数据隐私泄露问题，此外，云服务器也不希望透露其训练数据集的任何细节或参数。因此，研究支持隐私保护训练的神经网络方案至关重要。

为了解决云环境下用户数据及模型参数隐私泄露问题，大多数学者采用同态加密(Homomorphic Encryption，HE)技术来构建密态神经网络模型[3-4]。Gilad-Bachrach 等[5]提出的 CryptoNets 模型是第一个使用同态加密来实现机器学习隐私保护的方案，云服务提供商利用用户的加密数据执行推理后返回加密结果。由于同态加密不支持比较计算及非多项式计算，因此 CryptoNets 方案将神经网络模型中的 ReLU 函数替换为平方激活函数，这影响了结果的准确性，且在牺牲准确性的前提下计算成本依然很大。Hesamifard 等[6]在加密数据上使用多项式近似激活函数方法来实现卷积神经网络(Convolutional Neural Networks，CNN)，所提方案实现了高效、准确和可扩展的隐私保护推理。Chou 等[7]提出的 Faster-CryptoNets 方案优化了模型简化方法，结合神经网络剪枝方法来减少原神经网络模型中参数的数量，从而缩减乘法操作的运算量，通过这种优化方式缩短了推理时间，但也因此造成了模型精度损失。此外，针对最大稀疏编码的处理，使用低阶多项式近似替代 ReLU 函数。相比原 CryptoNets 方案，FasterCryptoNets 方案在预测速率上提升了约 10 倍。为了实现复杂神经网络的隐私保护推理，Chabanne 等[8]使用低阶多项式近似计算激活函数，并基于全同态加密构造了安全批处理归一化层，使得神经网络模型适用于更复杂的隐私保护分类任务。由于采用低次多项式逼近非线性函数的方法存在模型精度损失问题，Juvekar 等[9]基于加法同态加密并结合混淆电路提出了神经网络框架 GAZELLE，但执行线性运算与非线性运算时需要在客户端和服务器端之间进行通信。为了打破上述模型仅支持 CPU 设置的局限性，Badawi 等[10]提出了兼容 GPU 设置的同态加密神经网络模型，有效加快了模型的运算速度。

上述方案主要关注神经网络模型的安全推理过程，但是，若用户将模型训练任务外包至云服务器，模型训练过程中将同样存在数据隐私泄露问题，对此，已有学者将同态加密方案应用于神经网络模型训练中。Han 等[11]基于同态加密首次实现了隐私保护神经网络训练模型，采用低阶多项式逼近方法实现了激活函数的安全计算，结合支持近似定点数计算的同态加密方案[12]来提高模型的训练精度。Zhang 等[13]利用 Paillier 半同态加密技术提出了 GELU-Net 方案，利用服务器与用户间的协作计算来避免多项式逼近激活函数所造成的精度损失，同时也可避免密文间的乘法同态计算。Bourse 等[14]对同态加密方案[15]的 Bootstrapping 过程进行了改进，缩短了神经网络的安全计算时间，但也因此损失了一些精度。Hesamifard 等[16]采用切比雪夫多项式来近似非线性函数，在一定程度上提高了 ReLU 激活函数的精度。此方案使用全同态加密对数据进行加密，为了避免使用自举技术，设置了噪声阈值，当乘法的噪声达到此阈值时需要服务器将加密模型返回给用户执行解密操作。Lou 等[17]提出了一种新的密码系统技术来实现 BGV 和 TFHE 之间的同态切换，使用 BGV 计算线性函数，基于 TFHE 构造电路门来计算 ReLU，Softmax 等非线性函数，虽然使用电路可执行快速计算，但构造电路过程较为复杂，且在算术电路与逻辑电路之间来回切换需要耗费大量的时间开销。

针对云环境下的数据泄露和当前隐私保护训练精度优化问题，本文设计并实现了一种双服务器协作的隐私保护神经网络训练方案，旨在在保证全连接神经网络模型准确性及数据隐私性的同时，尽可能降低通信复杂度。为此，基于 Paillier 半同态加密技术和加法秘密共享方案，设计了一系列与全连接神经网络各层对应的安全交互协议，避免了密文多项式多轮迭代计算非线性函数。本文的具体贡献如下。

(1)基于 Paillier 半同态加密技术及加法秘密共享方案，设计了安全比较(Secure Comparison，SComp)、安全指数(Secure Exponent，SExp)、安全 ReLU(Secure ReLU，SRE)、安全乘法(Secure Multiplication，SMul)、安全 Softmax(Secure Softmax，SSF)和安全 ReLU 导数(Secure Derivative of ReLU，SDRE)等安全计算协议。相比多项式近似方法，本文方案具有精确计算非线性函数的优势。

(2)基于安全计算协议，设计并实现了一种双服务器协作的隐私保护神经网络训练方案 PPNT。用户不参与在线运算，训练过程中两个服务器均不能获得完整的计算结果，PPNT 既可保护推理阶段的输入数据和中间结果，也可保护训练阶段的模型参数。

(3)通过理论分析证明了本文所提安全计算协议和 PPNT 方案的正确性与安全性。实验结果显示，相比 PPMLaaS 方案，PPNT 方案的模型精度提高了 1.7%。

2　预备知识

本文所提的安全计算协议主要基于 Paillier 半同态加密方案、全连接神经网络和加法秘密共享技术，下文将对相关概念和基本知识进行介绍。

2.1　Paillier 半同态加密方案

Paillier 是一种基于合数剩余类问题的半同态加密(Partially Homomorphic Encryption，PHE)公钥密码方案[18]，只支持加法同态运算，因其安全性证明完备、效率较高的特点，是隐私计算场景中常用的 PHE 实例化方案，在实际应用中

得到了广泛的使用。

Paillier 半同态加密方案由加密算法 Enc、密钥生成算法 KeyGen 和解密算法 Dec 组成。

(1)密钥生成 $KeyGen(\cdot)\rightarrow(pk,sk)$:产生两个独立的随机大素数 p 与 q,且满足 $gcd(pq,(p-1)(q-1))=1$,计算 $\lambda=lcm(p-1,q-1)$ 与 $n=pq$,选取随机整数 $g\in Z_{n^2}^*$,则用于加密数据的公钥 $pk=(n,g)$ 和私钥 $sk=\lambda$。

(2)加密过程 $Enc_{pk}(m)\rightarrow c$:已知明文信息 m,随机选择 $r\in Z_n^*$,使用公钥 pk 对 m 进行加密,可得密文信息 c,满足 $c=g^m r^n \bmod n^2$。

(3)解密过程 $Dec_{sk}(c)\rightarrow m$:使用 sk 对 c 进行解密得到 m,令函数 $L(x)=(x-1)/n$,满足 $m=(L(c^\lambda \bmod n^2))/(L(g^\lambda \bmod n^2)) \bmod n$。

Paillier 半同态加密方案支持同态标量乘法以及同态加法运算,其具体定义如下。

(1)同态标量乘算法:已知明文信息 m 和标量 k,c 表示 m 对应的密文信息,满足 $Dec(c^k \bmod n^2)=m\cdot k$。

(2)同态加算法:已知明文信息 m_1 和 m_2,对应的密文信息分别为 c_1 和 c_2,满足 $Dec((c_1\cdot c_2) \bmod n^2)=m_1+m_2$。

2.2 全连接神经网络

全连接神经网络(Fully Connected Neural Network,FCNN)中每个结点和下一层所有结点都有运算关系,FCNN 通常有多个隐藏层,增加隐藏层可以更好地分离数据的特征。当原输入数据是线性不可分时,使用激活函数产生非线性输出,例如引入 Sigmoid 函数、Tanh 函数、ReLU 函数等作为激活函数。全连接神经网络训练分为前向传播、后向传播两个过程。

(1)前向传播。神经网络层与层之间的计算分为两部分:线性与非线性计算。第 L 层网络的线性变换输出结果可表示为 $z^L=a^{L-1}\omega^L+b^L$,z^L 即非线性计算的输入,其中 b^L 表示第 L 层的偏置项,ω^L 表示第 L 层和第 $L-1$ 层间的权值矩阵,$a^L=\sigma(z^L)$ 表示非线性计算的输出,即下一层网络的输入,其中 $\sigma(\cdot)$ 为激活函数。

(2)反向传播。通过链式法则计算目标函数关于模型参数的偏导数,以此更新优化模型参数,使得神经网络的误差减小,从而达到训练的目的。假设神经网络的目标损失函数为 $J=-\sum(\bar{y}_i\ln a_i)$,$i\in(0,9)$,其中 $\bar{y}_i$ 为实际标签值,a_i 为神经网络的输出值,第 L 层网络的误差为 $\delta^L=(\omega^{L+1})^{\mathrm{T}}\delta^{L+1}\sigma'(z^L)$,利用 δ^L 对模型参数进行更新,使得 $\frac{\partial J}{\partial \omega}=\delta^L(a^{L-1})^{\mathrm{T}}$,$\frac{\partial J}{\partial b}=\delta^L$。

2.3 加法秘密共享技术

秘密共享将秘密数据随机拆分为若干秘密份额,然后将其传输至多个计算方并由其掌管,可达到容忍入侵和分散风险的目的。Shamir 于 1979 年提出了 t-out-of-N 秘密共享方案[19],对秘密数据 x 进行随机拆分并在 N 个计算方中共享,使得由 t 个或多余 t 个计算方所掌握的部分秘密份额可恢复秘密数据 x,而任何少于 t 个计算方所掌握的部分秘密份额则无法恢复秘密数据 x。一般来说,该方案由秘密共享 Share 和秘密重构 Recon 两种算法组成。

(1)Share(x,t,N):已知原始秘密 x、计算方数 t 和秘密份额数 N,秘密 x 可被随机拆分为 N 部分,即 $\{x_1,x_2,\cdots,x_N\}$。

(2)Recon$(\bar{\omega},t)$:已知秘密份额子集为 $\bar{\omega}\in\{x_1,x_2,\cdots,x_N\}$,且满足 $|\bar{\omega}|\geqslant t$,则可恢复出原始秘密 x。

本文方案由双云服务器执行交互式协作计算任务,因此使用 2-out-of-2 秘密共享方案,也称作加法秘密共享。已知原始秘密 x,通过 Share$(x,2,2)$算法得到两个秘密份额 x_1 和 x_2。相应地,根据 x_1 和 x_2,通过 Recon$(2,\{x_1,x_2\})$算法恢复出原始秘密 x,且满足 $x=x_1+x_2$。

3 系统模型与安全模型

3.1 系统模型

本文旨在解决数据在信道传输、模型训练过程中和训练后结果的隐私安全问题,并兼顾方案的模型精度。系统模型如图 1 所示,PPNT 方案中的参与者包含客户端 C、可信第三方服务器 T、两台云服务器 S_1 和 S_2 及接收终端 R。

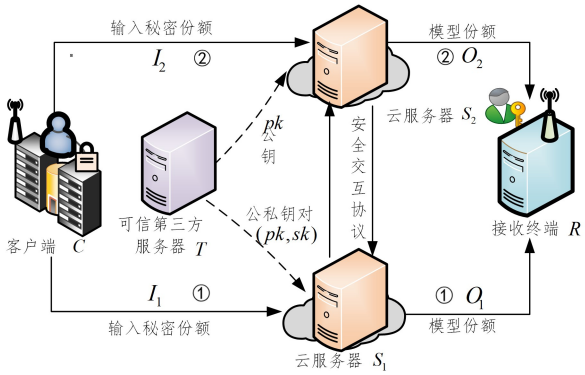

图 1 系统模型

Fig. 1 System model

(1)T 负责产生公私钥对(pk,sk)并将其发送给 S_1,同时将公钥 pk 发送给 S_2,T 不直接参与模型训练,因此可在离线阶段执行。

(2)为了保护图像数据的隐私,C 将图像数据 I 随机拆分为两份秘密份额 I_1 和 I_2,其中 $I=I_1+I_2$,并通过安全信道将 I_1 和 I_2 分别发送给 S_1 和 S_2。

(3) S_1 和 S_2 收到随机拆分后的秘密份额后,协作执行隐私保护神经网络的训练任务,S_1 和 S_2 可在本地执行线性函数计算,通过 Paillier 半同态加密方案对数据加密后进行交互,以完成非线性的密文计算,根据本文提出的安全计算协议执行安全全连接、安全激活和安全 Softmax 等模块操作,然后分别将训练后的模型份额 O_1 和 O_2 发送给 R。

(4) R 负责对两份模型份额执行简单加法运算,即可恢复完整的模型结果 O,其中 $O=O_1+O_2$。

3.2 安全模型

类似于现有文献定义[20-21],本文中云服务器 S_1 和 S_2 均为半诚实服务器,即 S_1 和 S_2 遵循安全协议步骤完成计算任务,但好奇计算过程中的数据,并试图通过这些数据推测出其他信息。假设 S_1 和 S_2 是独立非共谋的,云服务器持有的

数据通过Paillier半同态方案加密后进行交互，Paillier半同态方案满足语义安全，即保证交互过程中密文不会泄露明文的信息。此外，S_1 和 S_2 之间通过安全通道传递信息，以保证传输数据的安全性。

同时，在安全模型中引入一个概率多项式时间敌手 $\mathcal{A}$，其在同态加密方案中常被认定为具有多项式时间内的解密能力，且其正确地得到密文的概率小于随机猜测的概率，假设敌手 $\mathcal{A}$ 具备下述攻击能力[22-24]：至多可窃取 S_1 和 S_2 中一个云服务器持有的信息，不能同时窃听传递输入和输出的通信链路（图1中的①和②），且敌手不能干扰客户端、两个云服务器以及接收终端之间的正常通信。

4　安全函数设计方法

为了实现云服务器 S_1 和 S_2 之间的安全交互，且考虑一般非线性函数不能直接做拆分操作的局限性，本节利用乘法与加法之间可进行安全等值转换的思想，对神经网络中涉及的指数运算、比较运算、乘法运算构造相应的安全计算协议，在保证协议安全性的同时降低通信开销，实现精确的非线性计算。

4.1　安全指数函数

自然指数运算是Softmax模块的主要组成部分，已知输入数据 x，需要计算 e^x，首先利用加法秘密共享技术将 x 随机拆分为 x_1 和 x_2，满足 $x=x_1+x_2$，由于 $e^x=e^{x_1}\cdot e^{x_2}$，显然指数结果不能直接用加法秘密共享方案做拆分计算。因此可将指数的乘法形式转换为加法形式，S_1 和 S_2 分别持有 x_1 和 x_2，接着利用Paillier加密方案将数据加密并进行交互，由于 S_2 不知道私钥 sk，因此无法解密获得 e^{x_1}。类似地，S_2 利用Paillier的同态标量乘算法计算 $c_1\cdot q_2$，并将计算结果加噪后传输给 S_1。最后两个云服务器分别获得计算结果 μ_1 和 μ_2，其中 $\mu_1+\mu_2=e^{x_1+x_2}$，具体过程如协议1所示。

协议1　SExp协议

输入：S_1 持有 x_1，S_2 持有 x_2

输出：S_1 返回 μ_1，S_2 返回 μ_2

1. S_1 计算 $q_1\leftarrow e^{x_1}$；
2. S_2 计算 $q_2\leftarrow e^{x_2}$；
3. S_1 计算 $c_1\leftarrow \mathrm{Enc}_{pk}(q_1)$，并将 c_1 发送给 S_2；
4. S_2 选择随机数 r，计算 $c_2\leftarrow r+c_1\cdot q_2$，并将 c_2 发送给 S_1；
5. S_1 计算并返回 $\mu_1\leftarrow \mathrm{Dec}_{sk}(c_2)$；
6. S_2 计算并返回 $\mu_2\leftarrow -r$。

4.2　安全比较函数

比较运算在神经网络中使用得十分普遍，例如计算ReLU激活函数及其导数。为了实现 x 与 y 的比较，首先将其随机拆分为 $x=x_1+x_2$ 和 $y=y_1+y_2$，使得 S_i 持有 (x_i,y_i)，因此 S_i 可本地计算 x_i-y_i，利用差值法比较两个数值的大小，即差值 $x-y=(x_1-y_1)+(x_2-y_2)$ 的正负值表示比较结果。但比较操作为非线性计算，同样不能直接利用加法秘密共享方案对其进行直接拆分，但其实质上可看作求解 $x-y$ 的符号。因此利用将 $x-y$ 的加法形式转换为乘法形式的思想来完成比较运算[25]。然后利用Paillier半同态加密方案将数据加密并进行交互来获得计算结果，具体过程如协议2所示。步骤1－步骤6将 $x-y$ 的差值份额转换乘法份额 $\rho_1\cdot\rho_2$，步骤7求得 ρ_i 的符号位 ℓ_i，$x-y$ 的符号位可看做 ℓ_1 和 ℓ_2 的单比特异或结果，执行步骤8－步骤11使得安全比较协议的输出满足加法秘密份额形式，最后输出 $f_1+f_2=\ell_1+\ell_2-2\cdot\ell_1\cdot\ell_2$ 把异或计算转成算术运算[26]，使得 S_1 和 S_2 分别持有 f_i 的秘密份额 f_i，满足 $f=\mathrm{sign}(x-y)=f_1+f_2$，若 $f=0$，则 $x\geqslant y$；否则 $f=1$，$x<y$。

协议2　SComp协议

输入：S_1 持有 (x_1,y_1)，S_2 持有 (x_2,y_2)

输出：S_1 返回 f_1，S_2 返回 f_2

1. S_2 计算 $z_2\leftarrow x_2-y_2$；
2. S_1 计算 $z_1\leftarrow x_1-y_1$；
3. S_2 选择随机数 r_1，并计算 r_1 的乘法逆元 $\kappa\leftarrow r_1^{-1}$；
4. S_1 计算 $c_1\leftarrow \mathrm{Enc}_{pk}(z_1)$，将 c_1 发送给 S_2；
5. S_2 计算 $c_2\leftarrow c_1+z_2$ 和 $\eta\leftarrow r_1\cdot c_2$，并将 η 发送给 S_1；
6. S_1 计算 $\rho_1\leftarrow \mathrm{Dec}_{sk}(\eta)$，$S_2$ 计算 $\rho_2\leftarrow\kappa$；
7. S_i 计算 $\ell_i\leftarrow \mathrm{sign}(\rho_i)$；
8. S_1 选择随机数 $\tilde{r}_1$，计算 $\tilde{c}_1\leftarrow \mathrm{Enc}_{pk}(\tilde{r}_1+?_1)$，并将 $(\tilde{c}_1,\tilde{r}_1)$ 发送给 S_2；
9. S_2 选择随机数 $\tilde{r}_2$，计算 $\tilde{c}_2\leftarrow\tilde{r}_2+\tilde{c}_1\cdot\ell_2$，并将 $\tilde{c}_2$ 发送给 S_1；
10. S_1 计算 $\tilde{w}_1\leftarrow \mathrm{Dec}_{sk}(\tilde{c}_2)$；
11. S_2 计算 $\tilde{w}_2\leftarrow -(\tilde{r}_1\cdot\ell_2+\tilde{r}_2)$；
12. S_i 计算并返回 $f_i\leftarrow\ell_i-2\cdot\tilde{w}_i$。

4.3　安全乘法函数

对于输入数据 x 与 y，云服务器 S_i 持有秘密份额 (x_i,y_i)，计算 $x\cdot y$，使其满足 $x\cdot y=(x_1+x_2)\cdot(y_1+y_2)$，云服务器 S_i 无法通过本地计算得到乘法项，因此需要执行交互计算。考虑到在神经网络模型训练中除了存在普通乘法运算，还存在点乘运算，因此 S_1 先将数据进行加密发送给 S_2，由于 S_2 不知道私钥，无法解密获得 x_1 与 y_1，收到加密的数据份额后，S_2 利用同态加算法重构 x 和 y，再将重构的加密数据加入噪声后发送给 S_1。接着 S_1 解密收到的加密数据，执行普通乘法或点乘运算后，使用随机数对结果进行加噪。随后 S_1 和 S_2 分别计算结果 ξ_1 和 ξ_2，满足 $\xi_1+\xi_1=(x_1+x_2)\odot(y_1+y_2)$，其中 $\odot$ 表示矩阵乘或点乘。具体过程如协议3所示。

协议3　SMul协议

输入：S_1 持有 (x_1,y_1)，S_2 持有 (x_2,y_2)

输出：S_1 返回 ξ_1，S_2 返回 ξ_2

1. S_1 计算 $(\rho_1,\rho_2)\leftarrow(\mathrm{Enc}(x_1),\mathrm{Enc}(y_1))$，并将 (ρ_1,ρ_2) 发送给 S_2；
2. S_2 计算 $(\alpha_1,\alpha_2)\leftarrow(x_2+\rho_1,y_2+\rho_2)$；
3. S_2 随机选取 r_1 和 r_2，并计算 r_1 的乘法逆元 γ 及 $(\vartheta_1,\vartheta_2)\leftarrow(\alpha_1\cdot r_1,\alpha_2\cdot\gamma)$，将 $(\vartheta_1,\vartheta_2,r_2)$ 发送给 S_1；
4. S_1 计算并返回 $\xi_1\leftarrow \mathrm{Dec}_{sk}(\vartheta_1)\odot\mathrm{Dec}_{sk}(\vartheta_2)-r_2$；
5. S_2 计算并返回 $\xi_2\leftarrow r_2$。

5　PPNT方案

为了实现全连接神经网络训练过程与推理过程的隐私保护，本文基于SComp协议、SExp协议和SMul协议构造了全连接神经网络相应的安全交互协议。PPNT方案结构如图2所示，包含安全全连接（Secure Fully Connected，SFC）、安全ReLU、安全Softmax等模块的前向传播与反向传播过程。在

安全前向传播的计算过程中，S_i 分别接收到图像秘密份额 I_1 和 I_2、模型参数份额 (ω_1, b_1) 和 (ω_1, b_1)，协同执行 SMul 协议，实现 SFC 层中参数份额与图像秘密份额模型的安全计算。接下来通过执行非线性的 SRE 协议来增强模型的表达能力，安全激活层将份额的负特征值设置为零。随后 S_1 和 S_2 协同执行 SSF 协议，完成 Softmax 层的安全计算。在反向传播的计算过程中，S_1 和 S_2 协同计算并返回 SFC 层、激活层及 Softmax 层的梯度份额，接着逐层求出损失函数对各神经元的梯度份额，然后分别完成对模型参数的安全更新。在前向传播及反向传播计算过程中，输入数据及中间结果均以秘密份额的形式存在，因此能够保证整个隐私计算过程中用户数据、中间结果及模型参数的隐私性。

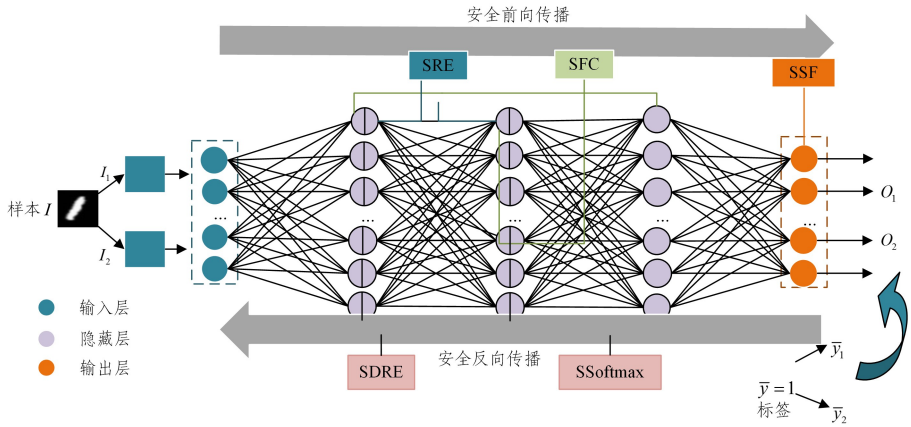

图 2 PPNT 方案的结构

Fig. 2 Architecture of PPNT scheme

5.1 SFC 层

SFC 层负责线性计算，其运算可表示为 $z=\omega \cdot a+b$，其中 z 表示神经网络的输出，a 表示神经网络的输入。为了保护计算过程的隐私性，将权重系数 ω 及偏置值 b 进行随机拆分，满足 $\omega=\omega_1+\omega_2$，$b=b_1+b_2$。因此 S_1 和 S_2 协同执行 SMul 协议计算得到 $\omega \cdot a$ 的秘密份额，随后 S_1 和 S_2 本地执行加法操作得到 $z_1=\omega_1 \cdot x_1+b_1$ 和 $z_2=\omega_2 \cdot x_2+b_2$，满足 $z=z_1+z_2$。

5.2 安全激活层

激活层的目的是引入非线性网络，现有方案多采用低次多项式逼近激活函数，但用此逼近方式会存在计算误差。因此，本文设计了一种安全的 SRE 协议，用于实现 ReLU 函数的精确计算。首先计算 ReLU 函数的导函数 $DReLU(x)=\begin{cases}0, & x\leqslant 0\\ 1, & x>0\end{cases}$，因此 $ReLU(x)=DReLU(x) \cdot x$，S_1 和 S_2 协同调用 SComp 协议计算输入值 x 与 0 的比较结果份额，再结合 ReLU 导函数的性质可计算得到导数的结果份额，具体过程如协议 4 所示。

协议 4 SDRE 协议

输入：S_1 持有 x_1，S_2 持有 x_2

输出：S_1 返回 υ_1，S_2 返回 υ_2

1. S_1 和 S_2 协同计算 $(g_1, g_2) \leftarrow SComp(x_1, x_2)$；
2. S_2 产生随机数 r_1 和 r_2，满足 $r_1+r_2=1$，并将 r_1 发送给 S_1；
3. S_1 计算并返回 $\upsilon_1 \leftarrow r_1-g_1$；
4. S_2 计算并返回 $\upsilon_2 \leftarrow r_2-g_2$。

在此基础上，S_1 和 S_2 协同执行 SDRE 协议来构建 SRE 协议，如协议 5 所示。激活函数的输入值 x 被 S_1 和 S_2 共享，即 $x=x_1+x_2$，S_1 和 S_2 协同重构可得到导数值 ζ。若 $\zeta=1$，则有 $S_i(i=1,2)$ 返回 x 的加法份额；否则返回 0 的加法份额。

协议 5 SRE 协议

输入：S_1 持有 x_1，S_2 持有 x_2

输出：S_1 返回 ψ_1，S_2 返回 ψ_2

1. S_1 和 S_2 协同计算 $(\zeta_1, \zeta_2) \leftarrow SDRE(x_1, x_2)$；
2. S_1 和 S_2 重构 (ζ_1, ζ_2) 得到 $\zeta \leftarrow \zeta_1+\zeta_2$；
3. S_1 计算并返回 $\psi_1 \leftarrow x_1 \cdot \zeta$；
4. S_2 计算并返回 $\psi_2 \leftarrow x_2 \cdot \zeta$。

5.3 安全 Softmax 层

Softmax 层将最后一层全连接模块的输出值进行归一化处理，使得多分类的输出结果转换为范围在 $[0,1]$、和为 1 的概率分布，Softmax 函数的定义为 $softmax(x_i)=e^{x_i}/\sum e^{x_i}$。特别地，针对 Softmax 函数提出 SSF 协议，已知输入数据 x_1 和 x_2，S_1 和 S_2 协同执行 SExp 协议分别获得输入特征向量的指数结果份额 g_1 和 g_2，并协作计算输出 Softmax 函数的分母结果 $g_1+g_2=\sum e^{x_i}$。最后 S_1 和 S_2 分别得到 ρ_1 和 ρ_2，满足 $\rho_1+\rho_2=\rho$，即为 Softmax 函数输出的归一化特征。具体过程如协议 6 所示。

协议 6 SSF 协议

输入：S_1 持有 x_1，S_2 持有 x_2

输出：S_1 返回 ρ_1，S_2 返回 ρ_2

1. S_1 和 S_2 协同计算 $(g_1, g_2) \leftarrow SExp(x_1, x_2)$；
2. S_1 和 S_2 协同计算 $g \leftarrow g_1+g_2$；
3. S_1 计算并返回 $\rho_1 \leftarrow g_1/g$；
4. S_2 计算并返回 $\rho_2 \leftarrow g_2/g$。

5.4 安全反向传播

安全反向传播计算过程中主要包含乘法运算与激活函数的求导运算。S_1 和 S_2 协同执行 SMul 协议来完成乘法运算，协同执行 SDRE 协议来完成 ReLU 函数的导数运算。对于反向传播中损失安全传递的问题，若假设 Softmax 函数为 $softmax(x_i)=e^{x_i}/\sum e^{x_i}$，交叉熵损失函数为 $J=-\sum(\bar{y}_i \ln a_i)$，$i \in (0,9)$，其中 x_i 表示全连接层的输出，$\bar{y}_i$ 为实际标签，a_i 即 Softmax 归一化特征。则可依链式求导法则得 Softmax 交叉

熵损失函数求导结果为$\frac{\partial J}{\partial z_i}=a_i-\bar{y}_i$。在安全反向传播过程中，首先计算输出层误差$\delta^L$，$S_1$和$S_2$分别拥有数据$(\bar{y}_1,a_1)$和$(\bar{y}_2,a_2)$，满足$\bar{y}=\bar{y}_1+\bar{y}_2$，$a=a_1+a_2$，本地计算得到输出层的误差为$\delta_1^L=a_1-\bar{y}_1$和$\delta_2^L=a_2-\bar{y}_2$，其中$\delta^L=\delta_1^L+\delta_2^L$；计算隐藏层的误差时涉及安全乘法，$S_1$和$S_2$分别收到上一层的误差、上一层的权重和当前层激活值输入$(\omega_1^{l+1},\delta_1^{l+1},\sigma_1^l(z))$及$(\omega_2^{l+1},\delta_2^{l+1},\sigma_2^l(z))$，且满足$\omega^{l+1}=\omega_1^{l+1}+\omega_2^{l+1}$，$\delta^{l+1}=\delta_1^{l+1}+\delta_2^{l+1}$，$\sigma^l(z)=\sigma_1^l(z)+\sigma_2^l(z)$，执行安全乘法协议计算权重和误差的乘积，接着协同执行SDRE协议得到当前层激活值的导数份额，S_1和S_2继续协同执行SMul协议得到隐藏层误差δ_1^l和δ_2^l；最后计算权重参数的变化率$\frac{\partial J}{\partial \omega^l}=\delta^l a^{l-1}$，$S_1$和$S_2$分别收到当前层的误差、下一层的激活值输入$(\delta_1^l,a_1^{l-1})$和$(\delta_2^l,a_2^{l-1})$，$S_1$和$S_2$对输入协同执行SMul协议得到权重变化率份额，安全反向传播协议(Secure Back Propagation，SBP)的具体执行过程如协议7所示。

协议7　SBP协议

输入：模型参数$(\omega^1,\omega^2,\cdots,\omega^L)$和$(b^1,b^2,\cdots,b^L)$及前向传播计算的激活值$(a^1,a^2,\cdots,a^L)$以及实际标签份额$\bar{y}_i$

输出：梯度值$\left(\frac{\partial J}{\partial \omega^1},\frac{\partial J}{\partial \omega^2},\cdots,\frac{\partial J}{\partial \omega^L}\right)$和$\left(\frac{\partial J}{\partial b^1},\frac{\partial J}{\partial b^2},\cdots,\frac{\partial J}{\partial b^L}\right)$的秘密份额

1. S_i的本地输出层误差为$\delta_i^L\leftarrow a_i-\bar{y}_i$；

2. S_1和S_2协同计算权重变化率$\left(\left(\frac{\partial J}{\partial \omega^L}\right)_1,\left(\frac{\partial J}{\partial \omega^L}\right)_2\right)\leftarrow \mathrm{SMul}(\delta_1^L,\delta_2^L,a_1^{L-1},a_2^{L-1})$；

3. S_i本地计算偏置变化率$\left(\frac{\partial J}{\partial b^L}\right)_i\leftarrow(\delta^L)_i$；

4. for l=1…(L−1)：

5. S_1和S_2协同计算$(t_1,t_2)\leftarrow \mathrm{SMul}(\omega_1^{l+1},\omega_2^{l+1},\delta_1^{l+1},\delta_2^{l+1})$；

6. S_1和S_2协同计算$(\sigma_1'(z^l),\sigma_2'(z^l))\leftarrow \mathrm{SDRE}(\sigma_1(z^l),\sigma_2(z^l))$；

7. S_1和S_2协同计算$(\delta_1^l,\delta_2^l)\leftarrow \mathrm{SMul}(t_1,t_2,\sigma_1'(z^l),\sigma_2'(z^l))$；

8. S_1和S_2协同执行步骤2、步骤3，计算当前层$\left(\left(\frac{\partial J}{\partial \omega^L}\right)_1,\left(\frac{\partial J}{\partial b^L}\right)_1\right)$和$\left(\left(\frac{\partial J}{\partial \omega^L}\right)_2,\left(\frac{\partial J}{\partial b^L}\right)_2\right)$；

9. end for

10. S_1和S_2结合梯度下降法本地更新梯度；

11. S_1计算并返回$\left(\left(\frac{\partial J}{\partial \omega^1}\right)_1,\left(\frac{\partial J}{\partial \omega^2}\right)_1,\cdots,\left(\frac{\partial J}{\partial \omega^L}\right)_1\right)$和$\left(\left(\frac{\partial J}{\partial b^1}\right)_1,\left(\frac{\partial J}{\partial b^2}\right)_1,\cdots,\left(\frac{\partial J}{\partial b^L}\right)_1\right)$；

12. S_2计算并返回$\left(\left(\frac{\partial J}{\partial \omega^1}\right)_2,\left(\frac{\partial J}{\partial \omega^2}\right)_2,\cdots,\left(\frac{\partial J}{\partial \omega^L}\right)_2\right)$和$\left(\left(\frac{\partial J}{\partial b^1}\right)_2,\left(\frac{\partial J}{\partial b^2}\right)_2,\cdots,\left(\frac{\partial J}{\partial b^L}\right)_2\right)$。

6　理论分析

6.1　正确性分析

已知原始图像I的两个份额I_1和I_2，S_1和S_2执行PPNT后输出模型结果O_1和O_2，其正确性依赖于安全计算协议的正确性。

已知输入x_1和x_2，SExp协议输出$x=x_1+x_2$的指数结果为$\mu_1+\mu_2=q_1\cdot q_2=e^{x_1}\cdot e^{x_2}$，满足$\mu=e^x$。针对SMul协议，$sign(\rho_1)$解密加噪后的$x'$和$y'$，执行乘法操作后$S_i$输出$\xi_i$，满足$x\cdot y=\xi_1+\xi_2$。针对SComp协议，$S_1$和$S_2$首先计算得到$(x_1-y_1)+(x_2-y_2)$的秘密份额$\rho_1$和$\rho_2$，满足$\rho=\rho_1\cdot\rho_2$，计算它们相应的符号位$\ell_i=sign(\rho_i)\in\{0,1\}$，即$\ell_i$为单比特值，因此$\ell_1\oplus\ell_2=sign(x-y)$，由于单比特异或形式可表示为算术形式：$\ell_1\oplus\ell_2=\ell_1+\ell_2-2\cdot\ell_1\cdot\ell 2$，而$\tilde{\omega}_1+\tilde{\omega}_2=\ell_1\cdot\ell_2$，因此最终输出的秘密份额$f_1+f_2=\ell_1+\ell_2-2(\tilde{\omega}_1+\tilde{\omega}_2)=\ell_1+\ell_2-2(\ell_1\cdot\ell_2)$，因此SExp协议、SMul协议和SComp协议是正确的。

神经网络中的全连接层实质上是执行线性乘法操作，对于输入$x=x_1+x_2$，有相应的输出$z=(\omega_1+\omega_2)\cdot(x_1+x_2)+(b_1+b_2)$，满足$z=z_1+z_2$，利用SMul协议可以正确计算出$(\omega_1+\omega_2)\cdot(x_1+x_2)$。接着由SDRE协议和SRE协议完成安全激活层的计算。在SDRE协议中，利用SComp协议来比较输入数据x与0的大小得到g_1和g_2，满足$g=g_1+g_2$，根据ReLU函数的导数特征可知，ReLU函数的导数结果值为$1-g$。在SRE协议中，S_1和S_2协作调用SDRE协议获得$(\varsigma_1,\varsigma_2)=SDRE(x_1,x_2)$，其中$\varsigma=\varsigma_1+\varsigma_2$，ReLU函数值为$\varsigma\cdot(x_1+x_2)$。当$\varsigma=1$时，$S_1$和$S_2$输出$x$的秘密份额，当$\varsigma=0$时，输出0的秘密份额。在SSF协议中，$S_1$和$S_2$协作执行SExp协议计算得到指数结果$g_1+g_2=e^{x_1}\cdot e^{x_2}$，并重构可得Softmax函数的分母$g=g_1+g_2$，$S_1$和$S_2$分别输出$\rho_1$和$\rho_2$，满足$\rho=\rho_1+\rho_2$，故$\rho$为Softmax处理后的归一化特征，可见SDRE协议、SRE协议和SSF协议是正确的。在反向传播计算过程中，输出层误差通过本地计算得到，接着S_1和S_2协同执行SMul协议和SDRE协议计算得到隐藏层梯度。显然，这一系列安全计算协议可以保证PPNT方案在理论上是完全正确的。

总而言之，虽然将输入数据随机拆分为两部分，但在PPNT方案中的每一层及最终输出始终保持了可加性，这确保了接收终端能够正确地恢复模型结果。

6.2　安全性分析

使用通用可组合框架来证明本文提出的计算协议的安全性，在本文的半诚实模型中，敌手$\mathcal{A}$至多可以破坏两个云服务器中的一个。在给定输入和输出的情况下，只要证明被破坏的云服务器的视图是可模拟的，就说明本文提出的计算协议是安全的，需要引入下述定义[6]及引理[21,27-29]。

定义1　若存在一个概率多项式时间的模拟器Sim，该模拟器可为真实世界的敌手$\mathcal{A}$模拟生成一组模拟器视图，且$\mathcal{A}$无法区分该模拟视图与其真实视图，则说明本文所提计算协议是安全的。

引理1　在概率多项式时间内，如果一个协议所依赖的所有子协议是完全可模拟的，则认为该协议是完全可模拟的。

引理2　如果随机元素u和v在Z_m上是均匀分布的，且u和v之间相互独立，则$u\pm v$也是均匀随机的并且与v相互独立。

由引理1可知，PPNT方案的安全性可归结于其依赖的安全计算协议的安全性证明，因此本节主要对这些安全计算协议的安全性进行证明。

定理1　在半诚实模型中，SExp协议和SComp协议是安全的。

证明：对于 SExp 协议，S_1 的真实视图为$\{q_1, x_1, c_1, \mu_1\}$，其中，q_1 和 μ_1 为本地计算所得结果，计算过程中不需要任何交互，因此敌手 $\mathcal{A}$ 不能窃取到 q_1 和 μ_1 的任何信息，将加密后的 c_1 发送给 S_2，由于 S_2 不知道私钥 sk，因此不能推测得到秘密值 x_1，模拟器 Sim 为 S_1 生成均匀分布的模拟视图，在多项式时间内与其真实视图是不可区分的。同理，S_2 的真实视图为$\{q_2, x_2, r, c_2, \mu_2\}$，依据引理 2 可得，$r$ 是均匀分布的随机数，S_1 无法推测得到 S_2 的秘密值 x_2。同样地，S_2 的真实视图及其模拟视图在多项式时间内是不可区分的。在 SComp 协议中，S_1 的真实视图为$\{x_1, z_1, c_1, \rho_1, f_1, \ell_1, \tilde{r}_1, \tilde{c}_1, \tilde{w}_1\}$，其中 x_1 为输入数据，$\{z_1, \rho_1, f_1, \ell_1, \tilde{w}_1\}$可在本地计算得到，因此敌手 $\mathcal{A}$ 无法得到任何信息，S_1 使用 pk 对秘密值 z_1 进行加密得到 c_1，故未持有私钥的 S_2 收到 c_1 后不能推测得到 S_1 的秘密值 z_1，$\tilde{r}_1$ 为随机值，$\tilde{c}_1$，$\tilde{w}_1$ 中均加了随机值，由引理 2 可知，$\tilde{c}_1$，$\tilde{w}_1$ 是均匀分布的，模拟器 Sim 为 S_1 生成均匀随机分布的模拟视图，且与其真实视图在多项式时间内无法进行区分。同理，也无法对 S_2 的真实视图进行区分。由定义 1 可知，SExp 协议与 SComp 协议是安全的。

证毕。

定理 2 在半诚实模型中，SMul 协议和 SSF 协议是安全的。

证明：在 SMul 协议中，S_2 的真实视图为$\{x_2, \alpha_1, \alpha_2, r_1, r_2, \gamma, \vartheta_1, \vartheta_2, \xi_2\}$，其中 r_1, r_2 是均匀分布的随机数，依据引理 2 可得，$\gamma, \vartheta_1, \vartheta_2, \xi_2$ 是均匀分布的，当 S_1 接收到 $\gamma, \vartheta_1, \vartheta_2$ 后，不能正确解密出 x_1+x_2 和 y_1+y_2 的值。S_1 的真实视图为$\{x_1, \rho_1, r_2, \rho_2, \xi_1\}$，由于 r_2 是均匀分布的，S_1 不能推测得到 $x \cdot y$ 的秘密信息，模拟器 Sim 为 S_1 和 S_2 生成均匀随机分布的模拟视图，在多项式时间内与其真实视图是不可区分的。在 SSF 协议中，S_i 的真实视图为$\{x_i, g_i, g, \rho_i\}$，g_i 为 SExp 协议的计算结果，由于 SExp 协议在定理 1 中已被证明是安全的，依据引理 1 可得，SSF 协议的安全性可由这些子协议来保证，模拟器 Sim 为 S_i 模拟生成均匀随机分布的模拟视图，敌手 $\mathcal{A}$ 在多项式时间内无法对两者进行区分。由定义 1 可知，SMul 协议和 SSF 协议是安全的。证毕。

定理 3 在半诚实模型中，SRE 协议、SBP 协议和 SDRE 协议是安全的。

证明：在 SDRE 协议中，S_1 和 S_2 的真实视图分别为$\{x_1, g_1, r_1, \upsilon_1\}$和$\{x_2, g_2, r_2, \upsilon_2\}$，其中，$r_1$ 和 r_2 是均匀分布的随机数，依据引理 2 可得，υ_1 和 υ_2 是均匀分布的，S_1 和 S_2 均无法推测得到 $sign(x-y)$的秘密信息。此外，g_1 和 g_2 为 SComp 协议的输出，在定理 1 中已证明 SComp 协议是安全的，依据引理 1 可得，模拟器可以为 S_i 生成均匀随机分布的模拟视图，在多项式时间内与其真实视图是不可区分的。在 SBP 协议中，S_i 的真实视图为$\{\omega_i, a_i, b_i, \bar{y}_i, \delta_i^{\prime}, \sigma_i', t_i\}$，其中$\{\omega_i, a_i, b_i, \bar{y}_i\}$为输入数据，$\sigma_i'$ 是 SDRE 协议的计算结果，$\delta_i^{\prime}$和 t_i 是 SMul 协议的计算结果，SDRE 协议和 SMul 协议已被证明是安全的，依据引理 1 可得，模拟器 Sim 可以为 S_i 生成均匀随机分布的模拟视图，且与其真实视图在多项式时间内是不可区分的。在 SRE 协议中，S_i 的真实视图为$\{x_i, \zeta_i, \psi_i, \zeta\}$，其中 ζ_i 为 SDRE 协议的输出结果，SDRE 协议已被证明是安全的，ζ 为 S_i 协同计算的部分结果值，故敌手 $\mathcal{A}$ 无法推测得到完整信息，依据引理 1 可得，模拟器 Sim 可以为 S_i 生成均匀随机分布的模拟视图，且与其真实视图在多项式时间内是不可区分的。由定义 1 可知，SRE 协议、SBP 协议与 SDRE 协议是安全的。证毕。

6.3 复杂性分析

由于 PPNT 方案是基于 SExp 协议、SComp 协议、SMul 协议、SDRE 协议、SSF 协议和 SRE 协议等构建的，本文主要分析这些安全计算协议的计算复杂度及通信复杂度，以此来评估所提 PPNT 方案中安全计算协议的效率。

(1)计算复杂度。对于计算复杂度而言，首先假设输入数组长度为 n，Enc 表示一次加密计算，$PAdd$ 表示一次同态加计算，$PMul$ 表示一次同态标量乘计算，Dec 表示一次解密计算，安全计算协议的计算复杂度如表 1 所列。由于 SDRE 协议与 SRE 协议主要依赖 SComp 协议完成计算任务，因此计算复杂度与 SComp 协议保持一致。同样地，SSF 协议的实现依赖于 SExp 协议。安全计算协议的大部分运算可在本地执行，只有在涉及数据交互计算时使用 Paillier 半同态加密方案对交互数据进行加密传递，且避免了复杂的密文乘密文操作。在实际的训练过程中，安全反向传播使用到的 ReLU 导数值在安全前向传播中已被记录，因此不需要再次执行加密计算。

表 1 安全计算协议的计算复杂度

Table 1 Computational complexity of secure computing protocols

协议	Enc	PAdd	PMult	Dec
SExp	n	0	n	n
SComp	$2n$	$2n$	$2n$	$2n$
SMul	$2n$	0	$2n$	$2n$
SDRE	$2n$	$2n$	$2n$	$2n$
SSF	n	0	n	n
SRE	$2n$	$2n$	$2n$	$2n$

(2)通信复杂度。安全计算协议的通信开销与传输数据大小及通信轮数相关，假设明文空间中单位明文大小为 $\|M\|$，密文空间中的单位密文大小为 $\|C\|$。表 2 列出了本文安全计算协议与相关工作的通信轮次，本文的 SComp 协议仅需要一轮通信来相互传递一份密文数据，SMul 协议在一轮通信中相互传递两份密文数据及一份均匀分布的随机值。文献[29]提出的 SComp 协议需要对待比较的输入数据逐比特地进行计算，S_1 和 S_2 需要 $2+\log l$ 轮通信。文献[22]提出的 SMul 协议同样需要一轮通信来进行交互，而 SComp 协议迭代执行 SMul 协议，因此需要 $2l+4$ 轮通信。文献[24]提出的 SExp 协议迭代执行 h 次安全乘法协议，至少需要进行 $h+1$ 轮通信，而本文的 SExp 协议仅需要一轮通信。在 SSF 协议中，S_1 和 S_2 协同执行一次 SExp 协议，并交换指数结果份额，SDRE 协议调用一次 SComp 协议，且 S_2 将产生的随机值发送给 S_1，因此 SSF 协议需要进行 2 轮通信，SDRE 协议需要进行 3 轮通信。在 SRE 协议中首先调用一次 SDRE 协议，随后两个云服务器执行一次交互计算，则需进行 4 轮通信。文献[29]提出的 SDRE 协议和 SRE 协议均调用 SComp 协议和额外的协议，因此需要执行 $5+\log l$ 轮通信。相比之下，PPNT 方案具有通信优势。本文安全计算协议

的通信开销如表 3 所列。

表 2　安全计算协议的通信轮次比较

Table 2　Comparison of communication rounds of secure computing protocols

协议	文献[29]	文献[22]	文献[24]	Ours
SExp	—	—	$h+1$	1
SComp	$2+\log l$	$2l+4$	—	2
SMul	—	1	—	1
SDRE	$5+\log l$	—	—	3
SSF	—	—	—	2
SRE	$5+\log l$	—	—	4

注：l 表示二进制位长度，h 表示子协议的迭代次数

表 3　安全计算协议的通信开销

Table 3　Communication cost of secure computing protocols

协议	通信开销	协议	通信开销
SComp	$4\|C\|+\|M\|$	SDRE	$4\|C\|+2\|M\|$
SMul	$4\|C\|+\|M\|$	SSF	$2\|C\|+2\|M\|$
SExp	$2\|C\|$	SRE	$4\|C\|+4\|M\|$

7　性能分析

本文实验使用 A100-SXM4-40 GB 的 CPU，其硬件配置为 132 GB 内存、32 核的处理器和 940 GB 的硬盘。采用具有 784 个特征的 MNIST 手写体数字集，包含 10 个类别，训练和测试样本数分别为 60 000 和 10 000，训练或推理时将每个样本转换为向量并将其作为输入，利用 Numpy 工具执行计算。考虑到使用 Paillier 半同态加密方案不能直接对浮点数执行加密运算，因此需要将输入数据编码为对应的大整数，随后对大整数进行加密得到密文。对于实数 x，截断保留 x 的 $k(5\leqslant k\leqslant 10)$ 位小数，计算 $x'=[x\cdot 10^k]$，其中 $[\cdot]$ 为取整。

7.1　安全计算协议性能

协议的实际通信开销与云服务器之间传输数据大小、明文密文空间大小以及通信轮数相关。在两个云服务器交互过程中传输的明文为随机数，在内存中占用 4 字节，加密后数据在内存中占用 16 字节，协议的通信轮数均保持为常数，因此输入数据的大小是影响通信开销的主要原因。以最大输入长度的数组 $n=10^5$ 为例，由图 3 可知，安全计算协议的通信开销与输入数据长度呈正比关系，当输入数组长度 $n<10^3$ 时，协议的通信开销均小于 0.076 MB。当输入数组长度为 10^5 时，SDRE 协议和 SRE 协议的通信开销保持在 7.63 MB 范围内，SMul 协议的通信开销也可控制在 6.5 MB 以内，由表 4 可知，S_1 与 S_2 各自的通信开销基本维持在 3.81 MB 以内。

表 4　S_1 与 S_2 的通信开销

Table 4　Communication cost of S_1 and S_2

（单位：MB）

参与方	SExp	SComp	SMul	SDRE	SSF	SRE
S_1	1.52	3.43	3.05	3.43	1.90	3.81
S_2	1.52	3.05	3.43	3.43	1.90	3.81

7.2　实验结果

本文选择 2 种全连接神经网络模型进行实验，2 个神经网络模型具有 2 个隐藏层且有不同的神经元个数，表示为：Net1($784\times128\times128\times10$)，Net2($784\times128\times32\times10$)。使用 He 等的初始化方式[30]随机初始化网络模型中的权重系数和偏置，在激活层选择 ReLU 函数，并采用交叉熵损失函数来评价模型预测值和实际数值的差异程度，将学习率置为 0.1。

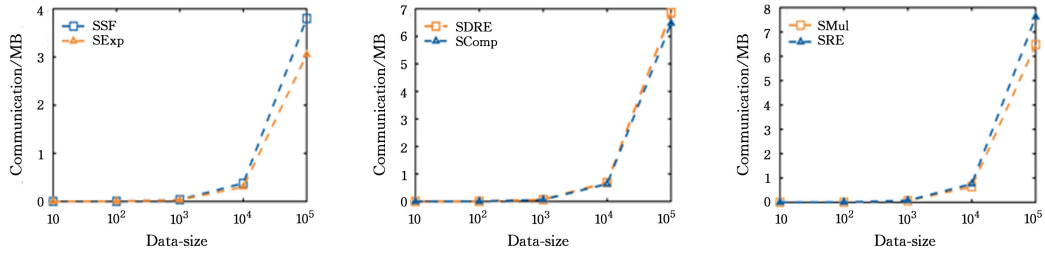

图 3　安全计算协议的通信开销

Fig. 3　Communication cost of secure computing protocols

模型精度是检验隐私保护、训练模型质量的重要指标，表 5 列出了 PPNT 方案与不同方案的模型精度比较。在 Net1 网络中，SecureML 方案[31]中线性函数的计算使用加法秘密共享方案，利用 ReLU 函数替代了 Softmax 公式中的指数部分，经过 15 代训练后模型精度达到 93.4%；PPMLaaS 方案[32]采用多项式近似计算激活函数，达到 95.15% 的模型精度需要训练 5 代；类似地，在 Net2 网络中，Glaph 方案[17]在第 5 代训练后得到 96.4% 的精度。上述方案中计算激活函数均是采用多项式近似的方法，使得模型精度存在计算误差，且多项式函数在一定程度上会影响模型的收敛速度，在 PPNT 方案中采用加法秘密共享和 Paillier 半同态加密方案设计安全计算协议，实现了非线性函数的精确计算。

表 5　PPNT 方案与其他方案的模型精度比较

Table 5　Comparison of model accuracy between PPNT scheme and other schemes

方案	网络	迭代轮数	模型精度/%
SecureML[31]	Net1	15	93.40
PPMLaaS[32]		5	95.15
Ours		5	96.89
Glaph[17]	Net2	5	96.40
Ours		5	96.80

此外，PPNT 方案与相关方案的隐私保护方法的比较结果如表 6 所列，CryptoNets 方案[5]与 MiniONN 方案[33]均是服务器基于使用明文训练得到的模型执行加密推理任务，但在 MiniONN 方案[33]中，对于神经网络中的每一层均需要服务器和用户协作计算，而得到的结果也是明文计算的两方

秘密分享。SecureML 方案[31]在训练过程中需要两个云服务器直接重构模型参数，这些信息可能会被泄露给敌手。在 PPMLaas 方案[32]中，为了避免同态加密代价高昂的自举操作，服务器在每次操作后检查密文中的噪声级别，如果噪声水平高于阈值，服务器将密文传输给用户并将其解密，紧接着加密并将新的密文传回至服务器。本文方案对训练过程中产生的相关计算数据及原始输入数据一直保持随机拆分状态，使得参与计算的云服务器只能持有部分数据，单个云服务器不能获得完整信息，可实现对模型参数、用户数据及推理结果的隐私保护。

表 6 不同方案的隐私保护方法比较

Table 6 Comparison of privacy-preserving method of different schemes

方案	客户端离线	数据隐私	模型参数隐私	推理	训练
MiniONN[33]	×	√	×	√	×
CryptoNets[5]	√	√	—	√	×
SecureML[31]	×	√	×	√	√
PPMLaaS[32]	×	√	√	√	√
Ours	√	√	√	√	√

结束语 本文基于 Paillier 半同态加密方案和加法秘密共享技术提出了一种神经网络隐私保护训练方案，针对现有方案使用多项式近似计算非线性函数存在计算误差的问题，利用乘法与加法之间的安全等值转换思想，设计了一系列的安全计算协议，用于非共谋双服务器协作训练神经网络模型，实现了线性函数与非线性函数的精确运算。理论分析证实了安全计算协议和 PPNT 方案的正确性和安全性，在整个隐私计算过程中均可确保用户数据、中间结果与模型参数的隐私性。实验结果显示，PPNT 方案兼备良好的通信开销和训练精度，且整个计算过程不需要客户端在线参与。在未来的研究工作中，将实现存在恶意服务器情况下的隐私保护训练方案，并构造兼容 GPU 设置的隐私保护网络模型。

参 考 文 献

[1] MA Z, LIU Y, LIU X, et al. Lightweight privacy-preserving ensemble classification for face recognition[J]. IEEE Internet of Things Journal, 2019, 6(3): 5778-5790.

[2] LUO X, LI L, WAN H, et al. Phone keypad voice recognition: an integrated experiment for digital signal processing education [C]// Proceedings of the 2020 IEEE Frontiers in Education Conference. Piscataway: IEEE Press, 2020: 1-4.

[3] LI Z Y, GUI X L, GU Y J, et al. Survey on homomorphic encryption algorithm and its application in the privacy-preserving for cloud computing[J]. Journal of Software, 2018, 29(7): 1830-1851.

[4] TAN Z W, ZHANG L F. Survey on privacy preserving techniques for machine learning[J]. Journal of Software, 2020, 31(7): 2127-2156.

[5] GILAD-BACHRACH R, DOWLIN N, LAINE K, et al. Cryptonets: applying neural networks to encrypted data with high throughput and accuracy[C]// International Conference on Machine Learning. New York: ACM Press, 2016: 201-210.

[6] HESAMIFARD E, TAKABI H, GHASEMI M. Cryptodl: Deep neural networks over encrypted data[J]. arXiv: 1711. 05189, 2017.

[7] CHOU E, BEAL J, LEVY D, et al. Faster cryptonets: leveraging sparsity for real-world encrypted inference[J]. arXiv: 1811. 09953, 2018.

[8] CHABANNE H, DE W A, MILGRAM J, et al. Privacy-preserving classification on deep neural network[J/OL]. Cryptology ePrint Archive, 2017, 1-35. http://eprint.iacr.org/2017/035.

[9] JUVEKAR C, VALKUNTANATHAN V, CHANDRAKASAN A. {GAZELLE}: A low latency framework for secure neural network inference[C]// 27th USENIX Security Symposium ({USENIX} Security 18). Berkeley: USENIX Association, 2018: 1651-1669.

[10] BADAWI A, CHAO J, JIE L, et al. Towards the alexnet moment for homomorphic encryption: hcnn, the first homomorphic cnn on encrypted data with gpus[J]. IEEE Transactions on Emerging Topics in Computing, 2021, 9(3): 1330-1343.

[11] HAN K, HONG S, CHEON J H, et al. Logistic regression on homomorphic encrypted data at scale[C]// Proceedings of the AAAI Conference on Artificial Intelligence. Menlo Park: AAAI Press, 2019: 9466-9471.

[12] BRAKERSKI Z, GENTRY C, VAIKUNTANATHAN V. (Leveled) fully homomorphic encryption without bootstrapping[J]. ACM Transactions on Computation Theory(TOCT), 2014, 6(3): 1-36.

[13] ZHANG Q, WANG C, WU H, et al. GELU-Net: a globally encrypted, locally unencrypted deep neural network for privacy-preserved learning[C]// Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence. Stockholm: IJCAI. 2018: 3933-3939.

[14] BOURSE F, MINELLI M, MINIHOLD M, et al. Fast homomorphic evaluation of deep discretized neural networks[C]// Annual International Cryptology Conference. Berlin: Springer, 2018: 483-512.

[15] CHILLOTTI I, GAMA N, GEORGIEVA M, et al. Faster fully homomorphic encryption: Bootstrapping in less than 0. 1 seconds [C]// International Conference on the Theory and Application of Cryptology and Information Security. Berlin: Springer, 2016: 3-33.

[16] HESAMIFARD E, TAKABI H, GHASEMI M, et al. Privacy-preserving machine learning in cloud[C]// Proceedings of the 2017 on Cloud Computing Security Workshop. New York: ACM Press, 2017: 39-43.

[17] LOU Q, FENG B, CHARLES F G, et al. Glyph: fast and accurately training deep neural networks on encrypted data[J/OL]. Advances in Neural Information Processing Systems, 2020, 33: 9193-9202. https://proceedings.neurips.cc/paper/2020/hash/685ac8cadc1be5ac98da9556bc1c8d9e-Abstract.html.

[18] PAILLIER P. Public-key cryptosystems based on composite degree residuosity classes[C]// Proceedings of the International Conference on the Theory and Dpplications of Cryptographic

Techniques. Berlin:Springer,1999:223-238.

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

[20] LIU Y,MA Z,LIU X,et al. Privacy-preserving object detection for medical images with faster R-CNN[J/OL]. IEEE Transactions on Information Forensics and Security,2022,17:69-84. https://doi.org/10.1109/TIFS.2019.2946476.

[21] XIONG J B,BI R W,TIAN Y L,et al. Towards lightweight,privacy-preserving cooperative object classification for connected autonomous vehicles[J]. IEEE Internet of Things Journal,2021,9(4):2787-2801.

[22] HUANG K,LIU X,FU S,et al. A lightweight privacy-preserving CNN feature extraction framework for mobile sensing[J]. IEEE Transactions on Dependable and Secure Computing,2019,18(3):1441-1455.

[23] XIONG J B,ZHOU Y J,BI R W,et al. Towards edge-collaborative,lightweight and privacy-preserving classification framework[J]. Journal on Communications,2022,43(1):127-137.

[24] MA Z,LIU Y,LIU X,et al. Privacy-preserving outsourced speech recognition for smart IoT devices[J]. IEEE Internet of Things Journal,2019,6(5):8406-8420.

[25] BI R W,CHEN Q X,XIONG J B,et al. Design method of secure computing protocol for deep neural network[J]. Chinese Journal of Network and Information Security,2020,6(4):130-139.

[26] WAGH S,TOPLE S,BENHAMOUDA F,et al. Falcon:honest-majority maliciously secure framework for private deep learning[J]. Privacy Enhancing Technologies,2021,2021(1):188-208.

[27] BOGDANOV D,NIITSOO M,TOFT T,et al. High-performance secure multi-party computation for data mining applications[J]. International Journal of Information Security,2012,11(6):403-418.

[28] XIONG J,BI R,ZHAO M,et al. Edge-assisted privacy-preserving raw data sharing framework for connected autonomous vehicles[J]. IEEE Wireless Communications,2020,27(3):24-30.

[29] XIONG J B,BI R W,CHEN Q X,et al. Towards edge-collaborative,lightweight and secure region proposal network[J]. Journal on Communications,2020,41(10):188-201.

[30] HE K,ZHANG X,REN S,et al. Delving deep into rectifiers:surpassing human-level performance on imagenet classification[C]// Proceedings of the IEEE International Conference on Computer Vision. Los Alamitos:IEEE Computer Society,2015:1026-1034.

[31] MOHASSEL P,ZHANG Y. Secureml:a system for scalable privacy-preserving machine learning[C]// 2017 IEEE Symposium on Security and Privacy(SP). Piscataway:IEEE Press,2017:19-38.

[32] HESAMIFARD E,TAKABI H,GHASEMI M,et al. Privacy-preserving machine learning as a service[J]. Proceedings on Privacy Enhancing Technologies,2018,2018(3):123-142.

[33] LIU J,JUUTI M,LU Y,et al. Oblivious neural network predictions via minionn transformations[C]// Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. New York:ACM Press,2017:619-631.

ZHAO Min, born in 1995, postgraduate. Her main research interests include secure machine learning and privacy protection.

XIONG Jinbo, born in 1981, Ph.D, professor, Ph.D supervisor. His main research interests include secure deep learning, mobile crowdsensing security and privacy protection.

(责任编辑:喻藜)