基于函数加密的密文卷积神经网络模型

王琛，李佳润，徐剑
（东北大学软件学院，辽宁 沈阳 110167）

摘 要：目前，多数的外包卷积神经网络（CNN）模型采用同态加密、安全多方计算等方法来保护敏感数据的隐私性。然而，上述方法存在计算与通信开销过大而引起的系统效率较低的问题。利用函数加密的低开销特点，构建了基于函数加密的密文卷积神经网络模型。首先，设计了内积函数加密算法和基本运算函数加密算法，实现了密文数据的内积、乘法、减法等基本运算，降低了计算与通信开销；然后，设计了针对基本运算的安全卷积计算协议和安全损失优化协议，实现了卷积层的密文前向传播和输出层的密文反向传播；最后，给出了模型的安全训练和分类方法，通过将以上安全协议进行模块化顺序组合的方式实现 CNN 对密文数据的训练和分类，该方法可以同时保护用户数据和标签的机密性。理论分析和实验结果表明，所提模型能够在保证正确性和安全性的前提下实现密文数据的训练和分类。

关键词：卷积神经网络；密文数据；函数加密；隐私保护
中图分类号：TP309.2
文献标志码：A
DOI: 10.11959/j.issn.1000−436x.2024050

Convolutional neural network model over encrypted data based on functional encryption

WANG Chen, LI Jiarun, XU Jian
Software College, Northeastern University, Shenyang 110167, China

Abstract: Currently, homomorphic encryption, secure multi-party computation, and other encryption schemes are used to protect the privacy of sensitive data in outsourced convolutional neural network (CNN) models. However, the computational and communication overhead caused by the above schemes would reduce system efficiency. Based on the low cost of functional encryption, a new convolutional neural network model over encrypted data was constructed using functional encryption. Firstly, two algorithms based on functional encryption were designed, including inner product functional encryption and basic operation functional encryption algorithms to implement basic operations such as inner product, multiplication, and subtraction over encrypted data, reducing computational and communication costs. Secondly, a secure convolutional computation protocol and a secure loss optimization protocol were designed for each of these basic operations, which achieved ciphertext forward propagation in the convolutional layer and ciphertext backward propagation in the output layer. Finally, a secure training and classification method for the model was provided by the above secure protocols in a module-composable way, which could simultaneously protect the confidentiality of user data as well as data labels. Theoretical analysis and experimental results indicate that the proposed model can achieve CNN training and classification over encrypted data while ensuring accuracy and security.

Keywords: convolutional neural network, encrypted data, functional encryption, privacy protection

---

收稿日期：2023−11−02；**修回日期：**2023−12−29
通信作者：徐剑，xuj@mail.neu.edu.cn
基金项目：国家自然科学基金资助项目（No.62372096, No.62173101）
Foundation Items: The National Natural Science Foundation of China (No.62372096, No.62173101)

0　引言

卷积神经网络（CNN, convolutional neural network）作为一种重要的深度学习模型，已经被广泛应用于图像识别[1]、人脸识别[2]、文本分类[3]等任务中，给人们的生活带来了许多便利。然而，CNN 的训练和分类过程涉及许多复杂的运算，通常需要强大的计算和存储资源来支持，但由于一些用户接入设备的计算和存储能力十分有限，特别是一些轻量级的计算终端（如智能手机、智能手环等），无法完成 CNN 模型的构建。因此，越来越多的用户选择将 CNN 模型部署到云服务器上[4]。云计算不仅极大地降低了用户的计算开销，同时有利于用户将自己的数据共享给其他用户。然而，人们在享受云计算带来便利的同时，隐私泄露的问题随之而来，且愈发严重。通常外包的数据包括个人身份数据、人脸数据、语音数据、金融数据、生物医学数据等，一旦这些包含敏感信息的数据发生泄露，将会对用户的隐私安全造成极大危害[5]。因此，如何保护用户隐私成为 CNN 与云计算技术结合的一个挑战。

很多学者致力于研究卷积神经网络隐私保护方法，以满足数据隐私保护需求。当前方案主要分为安全多方计算、差分隐私以及同态加密三类[6]。然而，由于卷积神经网络的特点，当前的卷积神经网络隐私保护方案仍存在如下问题：1) 安全多方计算的隐私保护卷积神经网络需要高度的协调和信息交换，需要数据所有者保持在线参与，会产生较大的通信开销[7]；2) 差分隐私技术只能对受噪声干扰的明文数据进行处理，因此不具备语义安全性[8-9]；3) 同态加密可以保护数据隐私且支持在密文数据上进行计算，但由于使用同态加密的数据只能由数据所有者解密，因此基于同态加密的隐私保护卷积神经网络方法大多只能在训练好的模型上进行分类[10]。同态加密的计算结果对云服务器是保密的，无法直接用于训练反向传播过程中对标签的评估[11-12]。函数加密（FE, functional encryption）是一种新兴的加密原语[13]，能够使云服务器对密文数据进行计算并返回明文结果，这种特性使函数加密能够很好地在云计算环境中得到应用。与同态加密相比，函数加密省略了由数据所有者对密文进行解密操作的步骤。因此，可以利用函数加密构建云计算环境中新型的密文卷积神经网络模型。

针对卷积神经网络的自身特点，密文卷积神经网络模型除了满足保护数据隐私的要求之外，还需要具备如下特性。1) 较低计算和通信开销的密码学原语：同态加密、安全多方计算等密码学原语能够在某种程度上对密文数据进行相对灵活的处理，但是目前这些方案往往需要较高的计算和通信开销，不适用于卷积神经网络。2) 支持密文计算的基本算法：密文 CNN 的计算主要涉及卷积层的前向传播过程和误差在输出层的反向传播过程。为了支持密文 CNN 训练和分类过程的计算，需要设计密文数据的内积、乘法、减法等基本算法。

为此，本文重点研究基于函数加密的密文卷积神经网络模型（CNNM-ED, convolutional neural network model over encrypted data），主要贡献如下。

1) 设计了基于判定性 Diffie-Hellman（DDH, decisional Diffie-Hellman）困难假设的函数加密算法，包括内积函数加密算法和基本运算函数加密算法。通过该算法实现了有限域内密文整数上特定的基本运算，例如内积、乘法、减法等。对每个算法包含的 4 个概率多项式时间子算法进行了详细设计。在解密算法中，设计了一个有界哈希表来加速离散对数过程，降低了计算和通信开销。

2) 设计了针对基本运算的安全通信协议，包括安全卷积协议和安全损失优化协议。分别通过内积函数加密算法（IPFEA, inner product functional encryption algorithm）和基本运算函数加密算法（BOFEA, basic operations functional encryption algorithm）构建了安全卷积协议和安全损失优化协议，给出了协议的详细描述。并通过上述协议实现了卷积层的安全卷积计算及输出层的安全损失函数和梯度计算，完成密文前向传播和反向传播过程。

3) 利用上述协议给出了 CNNM-ED 构建方法，实现了密文数据的 CNN 训练和分类。基于通信协议在 CNN 各层中进行迭代计算，能够保护用户数据以及标签的隐私。安全和性能分析表明，所提模型在保证正确性和安全性的前提下实现了密文卷积神经网络的训练和分类。

1　相关工作

当前，众多学者开展了关于隐私保护深度神经网络模型的研究工作。

文献[14]提出了一种基于安全多方计算的卷积神经网络分类模型，实现了图像的安全分类。该模型对卷积神经网络中的浅层和深层进行分析，采用

不同的计算方法来提升卷积神经网络模型的分类速度。文献[15]提出了支持隐私保护训练和分类的SecureNN，为神经网络构建模块提供新的安全三方和四方计算协议，在算术电路上对激活函数进行计算，从而实现了神经网络的安全训练和分类。文献[16]提出了 DeepSecure 框架，利用混淆电路对数据进行加密，在计算过程中可以完成任意激活函数和池化函数的计算。文献[17]设计实现了 Delphi 安全分类系统，在不泄露任何一方数据的前提下，该系统允许双方安全地执行神经网络分类。文献[18]提出了 GELU-Net（globally encrypted, locally unencrypted deep neural network），该网络通过设计一种安全的两方计算协议，在密文数据上完成模型训练。然而，该方案采用了基于噪声注入的隐私保护机制，这种机制在保护隐私的同时降低了模型训练精度。

文献[8]提出了基于差分隐私的深度神经网络方法，对隐私损失进行定量分析，训练具有非凸目标函数的深度神经网络，平衡了模型训练效率与隐私预算。文献[9]提出了一种基于差分隐私的新型卷积神经网络优化算法，在每次计算迭代中动态分配隐私预算，而不是固定隐私预算。理论分析和实验结果表明，该算法可以保护训练数据的隐私性，并且在保证隐私预算下实现了更高的分类精度。

文献[19]提出了 CryptoNets 隐私保护深度学习模型，通过将简单卷积神经网络转换为 CryptoNets 模型，从而实现加密数据的分类。由于同态加密不支持非多项式和比较运算，该模型通过替换激活函数、池化函数并近似替换浮点数，实现高效的加密数据分类。文献[20]对 CryptoNets 进行了改进，在神经网络非线性层较多的情况下，该方案仍然保证了较高的分类准确率。文献[21]通过采用全同态加密（FHE, fully homomorphic encryption）提出了一个高效的隐私保护卷积神经网络系统，具有高准确率以及低时延性。文献[22]提出了一种隐私保护和可验证卷积神经网络的分类方法，平衡了安全性和效率问题。由于 FHE 的高计算成本，上述工作仅限于浅层和狭窄的神经网络和简单的任务。文献[23]提出了一种更有效的评估卷积神经网络的方法，无论卷积核大小如何，代价都保持不变，从而加快了训练速度。文献[24]主要研究 Bootstrapped 全同态加密方法，允许计算任意复杂的函数，设计逻辑电路来执行加法、乘法、最大值、平均值以及求幂等运算，同时保持语义安全。通过上述电路实现了卷积神经网络模型分类，结果证明了安全参数和逻辑电路对系统性能的影响。

文献[25]提出了一种函数加密方案来实现二次多项式，该加密方案可以有效地计算加密向量上的二次多项式，并将其用于二次网络对加密数据进行分类。在此基础上，文献[26]提出了一种隐私保护分类方案，用于在神经网络中有效地计算二次函数，并增加了对抗性训练。文献[27]提出了在多个数据所有者的加密数据上训练深度神经网络模型的隐私保护方法。文献[28]基于函数隐藏提出了一种计算神经网络中激活函数的隐私保护方法，提高了隐私保护深度学习中计算激活函数的性能。

目前，隐私保护深度神经网络模型的研究取得了较大进展，但仍存在以下问题：1) 安全多方计算方法由于较高的通信和计算复杂度，会导致在实际应用中效率过低；2) 差分隐私方法通过向分类结果中添加噪声，保证攻击者无法从分类结果中反推出用户数据信息，但仍需要显式地访问用户数据，仅能保护分类结果而无法保护分类过程；3) 同态加密方法以密文形式产生计算结果，无法直接应用于训练反向传播过程中对标签的评估。同时现有方法多用于分类阶段，对训练阶段的研究较少。而函数加密在深度神经网络模型的应用仍处于探索阶段，且在已有的函数加密方案中没有考虑卷积神经网络模型训练，也没有考虑到对数据中标签的隐私保护。

因此，本文重点研究基于函数加密的卷积神经网络模型，实现密文卷积神经网络的训练和分类。

2 预备知识

2.1 函数加密

函数加密是在 Shamir[29]提出的身份基加密（IBE, identity-based encryption）和 Sahai 等[30]提出的属性基加密（ABE, attribute-based encryption）的基础上发展而来的一种新型公钥加密算法。从 2 个方面拓展了传统公钥加密算法体系：一是支持访问控制，只有满足特定条件的一方才可以解密；二是允许通过对密文进行选择性的计算从而直接得到计算结果。传统公钥解密中对密文的解密结果只有 2 种，一种是解密获取全部的明文信息，另一种是完全不能获取任何信息。

在函数加密中，记需要执行的函数为 f，其中持有主密钥的一方可以向解密方分发一个解密密

钥 sk_f，也称为函数密钥，只有拥有解密密钥 sk_f 的一方，才可以对密文数据进行解密，从而获取到该密文数据对应的函数结果 $f(x)$，其中，x 为明文消息。函数加密能够在不泄露相应明文的情况下获得密文上的函数结果。下面是函数加密的形式化定义。

定义 1　$F(K, X)$是一个定义在密钥空间 K 和明文空间 X 中的函数集合，其中每个函数 $f \in F$ 的密钥都与密钥空间 K 中的某个 $k \in K$ 值相关联。针对 F 的函数加密方案由 4 个概率多项式时间（PPT, probabilistic polynomial time）算法组成，可表示为 FE= (Setup, KeyDerive, Encrypt, Decrypt)，方案中的 4 个算法定义如下。

1) Setup (1^{λ})：初始化算法，其中 λ 为给定的安全参数，基于 λ 生成主公钥 mpk 和主私钥 msk。

2) KeyDerive (msk,k)：密钥生成算法，通过主私钥 msk 及与函数 $f \in F$ 相关联的 $k \in K$ 值，来生成一个对函数 $f \in F$ 的解密密钥 sk_f。

3) Encrypt (mpk, x)：加密算法，通过主公钥 mpk 和加密明文消息 $x \in X$ 生成密文 ct。

4) Decrypt (sk_f,ct)：解密算法，通过解密密钥 sk_f 去解密密文 ct，得到解密的函数结果 $f(x)$。

2.2　判定性 Diffie-Hellman 困难假设

DDH 困难假设主要关注离散对数问题的难度。Diffie-Hellman 协议是一种广泛应用于安全通信的密钥交换协议，而 DDH 困难假设为该协议提供了安全性基础。DDH 困难假设建立在数学上的计算不可区分性，尤其是模运算下的离散对数问题。

DDH 困难假设。给定一个阶为大素数 p 的乘法循环群 $\mathbb{G}$ 上的一个生成元 g，随机选取 3 个正整数 $a,b,c \in \mathbb{Z}_p$，则四元组 $D=(g, g^a, g^b, g^c)$与随机四元组 $R=(g, g^a, g^b, g^{ab})$在计算上是不可区分的，称为 DDH 假设。具体来说，对于任何一个敌手 $\mathcal{A}$，其在任意概率多项式时间 t 内区分四元组 D 与随机四元组 R 的优势是可以忽略的。

3　模型构建

3.1　总体框架

CNNM-ED 模型包括三方实体：可信第三方机构（TPA, trusted third-party authority）、客户端（C, client）和云服务器（CS, cloud server），如图 1 所示。

各方实体在 CNNM-ED 执行的过程如下。

1) 可信第三方机构。TPA 是 CNNM-ED 模型中其他部分都信任的实体，负责通过设置安全参数初始化模型，为 C 和 CS 提供密钥分发服务，并处理 CS 在执行 CNN 训练或分类过程中的函数密钥请求，为其生成函数密钥。TPA 仅相当于现有公钥基础设施中可信证书颁发机构的角色，此后，不参与任何训练或者分类计算过程。

2) 客户端。C 对需要提供的数据进行预处理，其中数据包括样本数据和标签，并根据设计的通信

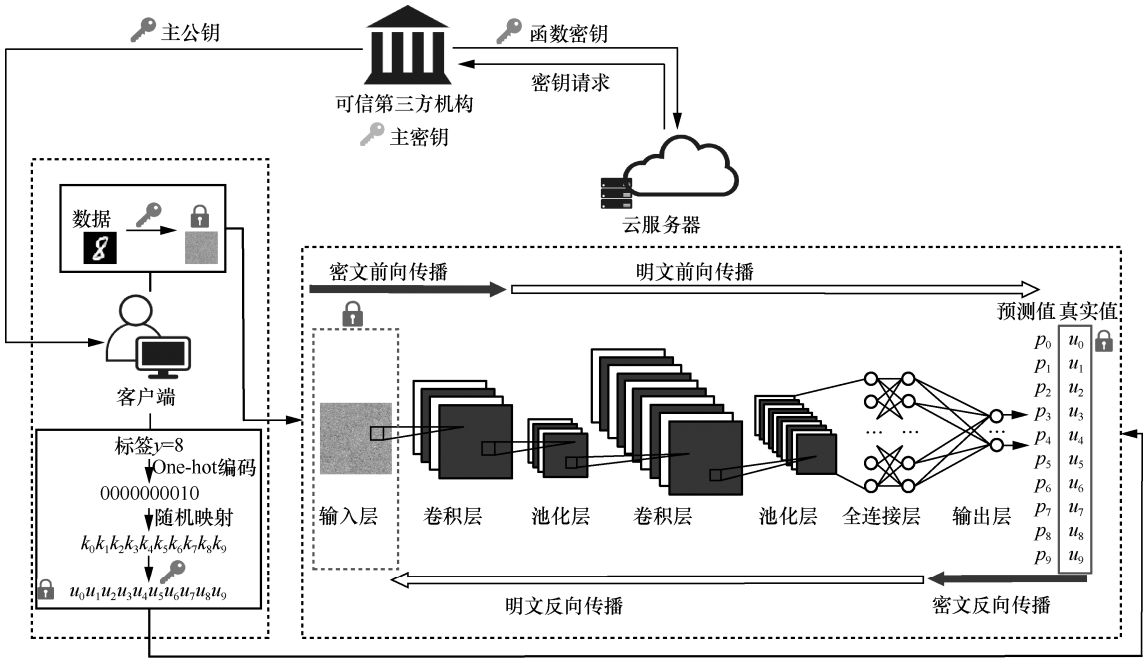

图 1　CNNM-ED 模型

协议将所有预处理的数据利用 IPFEA 和 BOFEA 加密算法进行加密发送给 CS，CS 在密文下为 C 提供有监督的卷积神经网络的训练或分类任务。

3) 云服务器。CS 负责将 C 提供的加密数据在 CNN 中进行训练或分类。在 CNN 训练或分类的前向传播过程中，CS 通过安全卷积协议与 TPA 进行交互实现第一个卷积层中密文数据与卷积核权重参数的卷积计算，完成密文前向传播过程；在训练的反向传播过程中，CS 通过安全损失优化协议与 TPA 进行交互实现输出层中损失函数和梯度的计算，完成密文反向传播过程；密文前向传播和反向传播获得的明文结果，在中间其余层执行明文前向传播和反向传播，实现 CNN 的训练或分类，最终将得到公开的明文模型或明文分类结果，在此过程中云服务器不会获得任何关于客户端中敏感数据的信息。

3.2 威胁模型

假设存在一个可信且独立的第三方（即 TPA），受到客户端和服务器中所有数据源的信任。TPA 类似于现有公钥基础设施中可信证书颁发机构的角色。

在本文中，考虑以下威胁模型。

1) 诚实但好奇的服务器：这是大多数现有方法的共同假设。在这里，服务器诚实地执行协议或算法，但可能会对加密数据集和训练阶段解密的函数结果好奇，通过收集数据来学习私有信息。

2) 共谋的客户端：一些客户端可能试图共谋，通过检查其外包的加密数据来推断其他非串通客户端的任何私人信息。

定义 2 （IPFEA 和 BOFEA 的 CPA 安全性）如果概率多项式时间内的敌手 $\mathcal{A}$ 赢得游戏的概率在设置的安全参数下是可以忽略的，即式(1)成立，则 IPFEA 和 BOFEA 具有选择明文攻击（CPA, chosen plaintext attack）安全性。

$$\left|\Pr\left[\mathrm{Exp}_0^{\mathrm{IPFEA/BOFEA}}(\mathcal{A})=1\right]-\Pr\left[\mathrm{Exp}_1^{\mathrm{IPFEA/BOFEA}}(\mathcal{A})=1\right]\right|=\mathrm{negl}(\lambda) \tag{1}$$

下面，基于 DDH 假设构建一个包含 4 个阶段的挑战者 $\mathcal{C}$ 与概率多项式时间内的任意敌手 $\mathcal{A}$ 之间的游戏来定义 IPFEA 和 BOFEA 的 CPA 安全性。具体的游戏 $\mathrm{Exp}_{\mathcal{A}}^{\mathrm{IPFEA/BOFEA}}$ 描述如下。

1) 初始化阶段：敌手 $\mathcal{A}$ 选择并向挑战者 $\mathcal{C}$ 发起 2 个特定的挑战消息 $w_0^*, w_1^* \in X$，对于内积函数加密算法，挑战者 $\mathcal{C}$ 将安全参数 λ 和消息长度 l 作为输入，运行 IPFEA 中的 Setup() 算法后将生成的主私钥 $\mathrm{msk}_{\mathrm{IP}}$ 自己保管，将公钥 $\mathrm{pk}_{\mathrm{IP}}$ 发布给敌手 $\mathcal{A}$；对于基本运算函数加密算法，挑战者 $\mathcal{C}$ 输入安全参数 λ 运行 BOFEA 中的 Setup() 算法，同样保留主私钥 $\mathrm{msk}_{\mathrm{BO}}$，发布公钥 $\mathrm{pk}_{\mathrm{BO}}$ 给敌手 $\mathcal{A}$。

2) 询问阶段 1：敌手 $\mathcal{A}$ 向挑战者 $\mathcal{C}$ 发起任意多项式次关于 $k_i \in Y(i=1,2,\cdots)$ 的函数密钥请求，其中对于内积函数加密算法，必须满足 $<w_0^*, k_i> = <w_1^*, k_i>$，挑战者 $\mathcal{C}$ 将 k_i 作为输入，运行 IPFEA 中的 KenGen() 算法后生成与其对应的函数密钥 sk_{k_i}，将其发布给敌手 $\mathcal{A}$；对于基本运算函数加密算法，挑战者 $\mathcal{C}$ 输入要执行具体运算的 k_i 运行 BOFEA 中的 KenGen() 算法，同样生成对应的函数密钥 sk_{k_i} 并发布给敌手 $\mathcal{A}$。

3) 挑战阶段：挑战者 $\mathcal{C}$ 随机选择一个 $\beta \in \{0,1\}$，对于内积函数加密算法，将消息 w_β^* 和公钥 $\mathrm{pk}_{\mathrm{IP}}$ 作为输入，运行 IPFEA 中的 Encrypt() 算法，生成挑战密文 ct_*，然后将其发送给敌手 $\mathcal{A}$；对于基本运算函数加密算法，与内积函数加密算法操作一样，公钥为 $\mathrm{pk}_{\mathrm{BO}}$，执行的算法为 BOFEA 中的 Encrypt() 算法。

4) 询问阶段 2：敌手 $\mathcal{A}$ 继续发起与询问阶段 1 相同的函数密钥请求。

5) 猜测阶段：敌手 $\mathcal{A}$ 对 β 进行猜测生成 β'，最后输出 β'。

定义 3 （CNNM-ED 的机密性）在密文训练和分类过程中，CNNM-ED 通过函数加密算法保证数据与标签隐私，提供了机密性保证。

3.3 关键算法

3.3.1 内积函数加密算法

基于 DDH 假设构造了一种支持有限域内的向量内积计算的内积函数加密算法。

定义 4 IPFEA 包括 4 个概率多项式时间算法，其形式化定义为 IPFEA=(Setup, Encrypt, KeyGen, Decrypt)，具体如下。

1) 初始化算法 $\mathrm{Setup}(1^\lambda, \mu) \to (\mathrm{pk}_{\mathrm{IP}}, \mathrm{msk}_{\mathrm{IP}})$，用来计算生成公私钥对，输入给定的安全参数 λ 和向量长度 μ，输出公钥 $\mathrm{pk}_{\mathrm{IP}}$ 和主私钥 $\mathrm{msk}_{\mathrm{IP}}$。

2) 密钥生成算法 $\mathrm{KeyGen}(\mathrm{msk}_{\mathrm{IP}}, \vec{y}) \to \mathrm{sk}_{\mathrm{IP},\vec{y}}$，用来生成具有内积函数功能的函数密钥，输入主私钥 $\mathrm{msk}_{\mathrm{IP}}$ 和计算内积的向量 $\vec{y}$，输出函数密钥 $\mathrm{sk}_{\mathrm{IP},\vec{y}}$。

3) 加密算法 $\text{Encrypt}(\text{pk}_{\text{IP}},\vec{x})\to \text{ct}_{\vec{x}}$，用来对明文向量 $\vec{x}$ 进行加密，输入公钥 pk_{IP} 对明文消息 $\vec{x}$ 进行加密，输出密文 $\text{ct}_{\vec{x}}$。

4) 解密算法 $\text{Decrypt}(\text{pk}_{\text{IP}},\text{sk}_{\text{IP},\vec{y}},\text{ct}_{\vec{x}},\vec{y})\to<\vec{x},\vec{y}>$，用来解密出内积函数的结果，而不泄露明文 $\vec{x}$，输入公钥 pk_{IP}、函数密钥 $\text{sk}_{\text{IP},\vec{y}}$、密文 $\text{ct}_{\vec{x}}$ 和向量 $\vec{y}$，输出函数计算的明文结果 $<\vec{x},\vec{y}>$。

下面给出 IPFEA 的执行过程。

Step1　$\text{Setup}(1^{\lambda},\mu)\to(\text{pk}_{\text{IP}},\text{msk}_{\text{IP}})$。初始化算法首先根据给定的安全参数 λ，生成一个安全的循环群 $\mathbb{G}$，定义 $\mathbb{G}$ 的阶为 λ 位长度的大素数 p，并令 g 为循环群 $\mathbb{G}$ 的生成元，形成一个三元组 $(\mathbb{G},p,g)$，在 DDH 假设中，(g,g^a,g^b,g^{ab})与(g,g^a,g^b,g^c)在计算上是不可区分的，其中 $a,b,c\in\mathbb{Z}_p$ 是独立且均匀随机选取的，然后根据向量长度 μ 独立且均匀地随机选择 $(s_1,\cdots,s_\mu)\leftarrow\mathbb{Z}_p^{\mu}$，并计算 $h_i=g^{s_i}$，其中 $i\in[1,\cdots,\mu]$，将其分别设置为主公钥 $\text{pk}_{\text{IP}}=(\mathbb{G},p,g,\{h_i\}_{i\in[1,\cdots,\mu]})$和主私钥 $\text{msk}_{\text{IP}}=(s_1,\cdots,s_\mu)$。

Step2　$\text{KeyGen}(\text{msk}_{\text{IP}},\vec{y})\to\text{sk}_{\text{IP},\vec{y}}$。密钥生成算法基于输入的主私钥 $\text{msk}_{\text{IP}}=(s_1,\cdots,s_\mu)$ 和向量 $\vec{y}=(y_1,\cdots,y_\mu)\in\mathbb{Z}_p^{\mu}$，计算出 $\vec{y}$ 对应的函数密钥 $\text{sk}_{\text{IP},\vec{y}}=\sum_{i=1}^{\mu}s_iy_i$。

Step3　$\text{Encrypt}(\text{pk}_{\text{IP}},\vec{x})\to\text{ct}_{\vec{x}}$。加密算法根据给定的主公钥 mpk_{IP} 和输入的待加密明文向量 $\vec{x}=(x_1,\cdots,x_\mu)\in\mathbb{Z}_p^{\mu}$，选取一个随机数 $r\leftarrow\mathbb{Z}_p$。首先计算 $\text{ct}_0=g^r$，然后对于每个 $i\in[1,\cdots,\mu]$，计算 $\text{ct}_i=h_i^rg^{x_i}$，输出密文 $\text{ct}_{\text{IP},\vec{x}}=(\text{ct}_0,\{\text{ct}_i\})$。

Step4　$\text{Decrypt}(\text{pk}_{\text{IP}},\text{sk}_{\text{IP},\vec{y}},\text{ct}_{\vec{x}},\vec{y})\to<\vec{x},\vec{y}>$。解密算法基于输入的 pk_{IP}、$\text{sk}_{\text{IP},\vec{y}}$、$\text{ct}_{\text{IP},\vec{x}}$、$\vec{y}$，通过式(2)计算得到 $g^{<\vec{x},\vec{y}>}$，然后通过求解 $g^{<\vec{x},\vec{y}>}$ 的离散对数得到解密后的明文函数值 $<\vec{x},\vec{y}>$，即向量 $\vec{x}$ 和向量 $\vec{y}$ 的内积值。

$$g^{<\vec{x},\vec{y}>}=\frac{\prod_{i=1,\cdots,\mu}\text{ct}_i^{\,y_i}}{\text{ct}_0^{\,\text{sk}_{\text{IP},\vec{y}}}}\tag{2}$$

3.3.2　基本运算函数加密算法

为了支持元素基本运算，即加法、减法、乘法和除法，构造一种基本运算函数加密算法，其安全性基于 DDH 假设，在计算过程中假设无法区分加密数据的计算结果与未加密数据的计算结果是否相等，可以在保护数据隐私的同时，实现对加密数据的特定函数运算。

定义 5　BOFEA 包括 4 个概率多项式时间算法，其形式化定义为 BOFEA=(Setup, Encrypt, KeyGen, Decrypt)，具体如下。

1) 初始化算法 $\text{Setup}(1^{\lambda})\to(\text{pk}_{\text{BO}},\text{msk}_{\text{BO}})$，用来计算生成公私钥对，输入给定的安全参数 λ，输出公钥 pk_{BO} 和主私钥 msk_{BO}。

2) 密钥生成算法 $\text{KeyGen}(\text{pk}_{\text{BO}},\text{msk}_{\text{BO}},\text{cmt},\bigcirc,y)\to\text{sk}_{\text{BO},y}$，用来生成具有内积函数功能的函数密钥，输入给定的公钥 pk_{BO}、主私钥 msk_{BO}、承诺 cmt、执行的运算 $\bigcirc\in[+,-,\times,\div]$ 和元素 y，输出与运算 $\bigcirc$ 相对应的函数密钥 $\text{sk}_{\text{BO},y}$。

3) 加密算法 $\text{Encrypt}(\text{pk}_{\text{BO}},x)\to\text{ct}_x,\text{cmt}$，用来对明文元素 x 进行加密，输入公钥 pk_{BO} 对明文消息 x 进行加密，输出密文 ct_x 和一个承诺 cmt。

4) 解密算法 $\text{Decrypt}(\text{pk}_{\text{BO}},\text{sk}_{\text{BO},y},\text{ct}_x,\bigcirc,y)\to x\bigcirc y$，用来解密出基本运算的函数结果，输入公钥 pk_{BO}、函数密钥 $\text{sk}_{\text{BO},y}$、密文 ct_x、执行的运算 $\bigcirc\in[+,-,\times,\div]$ 和元素 y，经过计算输出函数计算的明文结果 $x\bigcirc y$。

下面给出 BOFEA 的执行过程。

Step1　$\text{Setup}(1^{\lambda})\to(\text{pk}_{\text{BO}},\text{msk}_{\text{BO}})$。BOFEA 的初始化算法中三元组 $(\mathbb{G},p,g)$ 的生成方式与 IPFEA 相同，同样基于 DDH 假设，这里不再赘述，生成 $(\mathbb{G},p,g)$ 后从 $\mathbb{Z}_p$ 中独立且均匀地随机选择 s，计算 $h=g^s$，将其分别设置为公钥 $\text{pk}_{\text{BO}}=(g,h)$和主私钥 $\text{msk}_{\text{BO}}=s$。

Step2　$\text{KeyGen}(\text{pk}_{\text{BO}},\text{msk}_{\text{BO}},\text{cmt},\bigcirc,y)\to\text{sk}_{\text{BO},y}$。密钥生成算法基于公钥 pk_{BO}、主密钥 msk_{BO}、输入元素 $y\in\mathbb{Z}_p$ 以及承诺 cmt，根据具体要执行的基本运算 $\bigcirc\in[+,-,\times,\div]$，计算出函数密钥 sk_y。如果 $\bigcirc=+$，计算 $\text{sk}_{\text{BO},y}=\text{cmt}^sg^{-y}$；如果 $\bigcirc=-$，计算 $\text{sk}_{\text{BO},y}=\text{cmt}^sg^{y}$；如果 $\bigcirc=\times$，计算 $\text{sk}_{\text{BO},y}=(\text{cmt}^s)^y$；如果 $\bigcirc=\div$，计算 $\text{sk}_{\text{BO},y}=(\text{cmt}^s)^{y^{-1}}$，输出 $\text{sk}_{\text{BO},y}$。

Step3　$\text{Encrypt}(\text{pk}_{\text{BO}},x)\to\text{ct}_x,\text{cmt}$。加密算法根据给定的主公钥 pk_{BO} 和输入的明文元素 $x\in\mathbb{Z}_p$，

随机从 $\mathbb{Z}_p$ 中选取一个只用于生成承诺 cmt 和密文 ct_x 的临时随机数 r，计算 $\mathrm{cmt}=g^r$，$\mathrm{ct}_x=g^x h^r$，输出 cmt 和 ct_x。

Step4　$\mathrm{Decrypt}(\mathrm{pk}_{\mathrm{BO}},\mathrm{sk}_{\mathrm{BO},y},\mathrm{ct}_x,\bigcirc,y)\rightarrow x\bigcirc y$。解密算法根据执行的基本运算 $\bigcirc\in[+,-,\times,\div]$ 在密文 ct_x 上利用函数密钥 $\mathrm{sk}_{\mathrm{BO},y}$，进行对应的计算得到 $g^{x\bigcirc y}$，如果 $\bigcirc=\pm$，计算 $g^{x\bigcirc y}=\dfrac{\mathrm{ct}_x}{\mathrm{sk}_{\mathrm{BO},y}}$；如果 $\bigcirc=\times$，计算 $g^{x\bigcirc y}=\dfrac{\mathrm{ct}_x{}^y}{\mathrm{sk}_{\mathrm{BO},y}}$；如果 $\bigcirc=\div$，计算 $g^{x\bigcirc y}=\dfrac{\mathrm{ct}_x{}^{y^{-1}}}{\mathrm{sk}_{\mathrm{BO},y}}$，通过求解 $g^{x\bigcirc y}$ 的离散对数输出 $x\bigcirc y$，即 x 与 y 执行某具体基本运算的结果。

3.4　通信协议

CNN-CMED 的通信协议包括安全卷积计算协议和安全损失优化协议。安全卷积计算协议实现线性计算，例如卷积层；安全损失优化协议实现非线性计算，例如激活函数 ReLU 函数和 Softmax 函数。

在卷积神经网络的训练和分类过程中有三类计算，包括卷积计算、损失函数计算、误差反向传播计算。其中，卷积计算可以转换为内积计算，损失函数计算和误差反向传播计算经过推导可以转换为乘法和减法运算。卷积计算需要样本特征的参与，所以保护样本特征的隐私采用内积函数加密算法。损失函数计算和误差反向传播计算是标签参与的相关计算，所以保护标签的隐私采用基本运算函数加密算法。

在执行提出的协议前，需要对模型的密钥进行初始化。由于 CNNM-ED 需要同时保护数据和标签的隐私，并且数据参与的函数计算与标签参与的函数计算不同，因此根据不同的函数计算，初始化生成不同的密钥。

密钥初始化过程由 TPA 实现，通过输入安全参数 λ 和向量长度 μ 调用 IPFEA.Setup 和 BOFEA.Setup，输出密钥对 $(\mathrm{pk}_{\mathrm{IP}},\mathrm{msk}_{\mathrm{IP}})$ 和 $(\mathrm{pk}_{\mathrm{BO}},\mathrm{msk}_{\mathrm{BO}})$，TPA 在本地保存私钥 $\mathrm{msk}_{\mathrm{IP}}$ 和 $\mathrm{msk}_{\mathrm{BO}}$，并将 $\mathrm{pk}_{\mathrm{IP}}$ 和 $\mathrm{pk}_{\mathrm{BO}}$ 分发给 C 和 CS，完成密钥初始化。

3.4.1　安全卷积计算协议

1) 设计思想

安全卷积计算协议用于在卷积神经网络训练或分类过程中实现在不泄露样本特征信息的前提下完成与卷积核间的卷积计算，安全卷积计算协议包括样本特征的加密以及安全卷积计算。

图 2 为安全卷积计算协议中各实体间的通信时序图。

在不泄露 C 原始数据的情况下，CS 能够对密

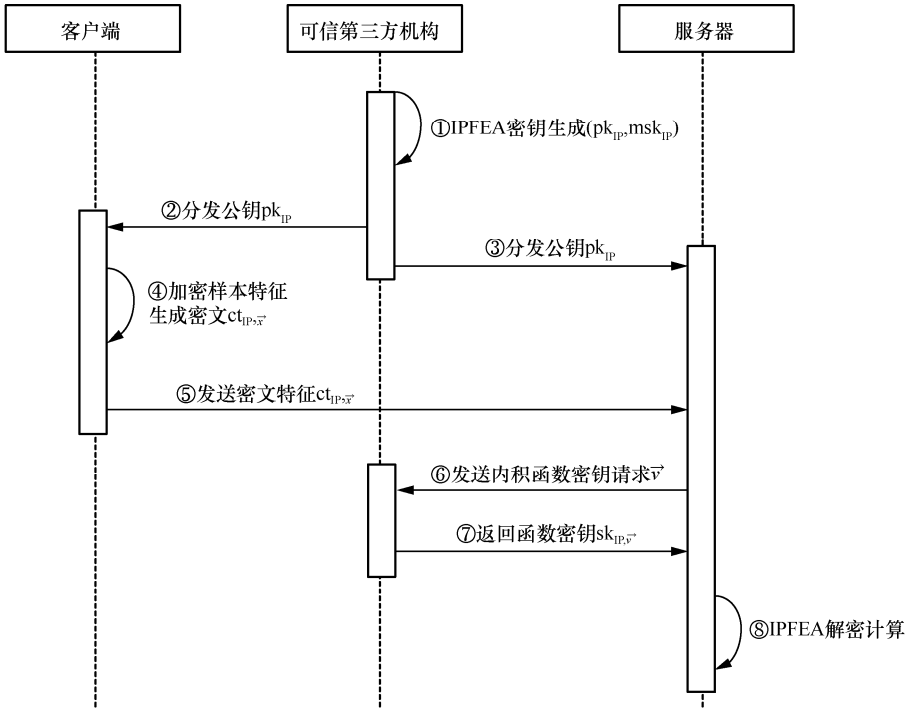

图 2　安全卷积计算协议中各实体间的通信时序图

文数据进行卷积操作，需要由 C 将预处理填充后的明文特征矩阵 $\boldsymbol{X}'$ 加密为适用于卷积计算的密文后发送给 CS，CS 再将 C 的密文样本特征与其卷积核参数进行卷积计算。在卷积计算中，CS 需要与 TPA 交互以获得用于解密计算的函数密钥。

在 C 与 CS 的安全卷积计算中，如图 3 所示，经过填充的明文特征矩阵 $\boldsymbol{X}'$ 与卷积核矩阵 $\boldsymbol{V}$ 之间的卷积计算可以分解为卷积核在矩阵 $\boldsymbol{X}'$ 上滑动所产生的滑动窗口矩阵和卷积核矩阵间的内积计算。

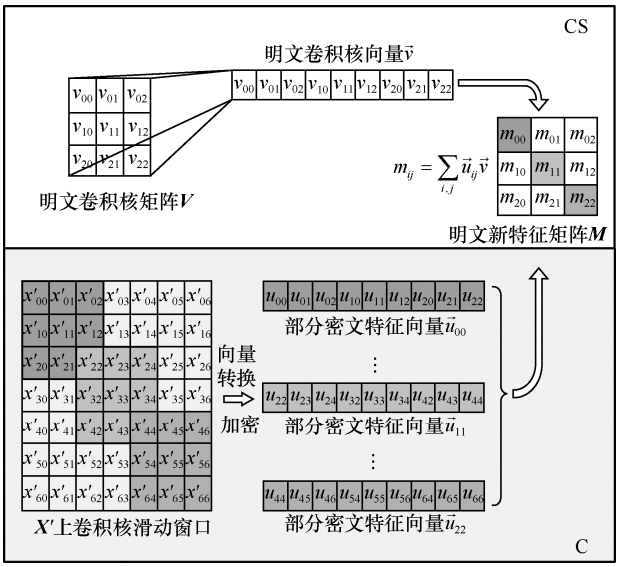

图 3　安全卷积计算过程

在此过程中，分别将所有的滑动窗口矩阵转化为一组向量 $\vec{u}_{00},\cdots,\vec{u}_{ij}$。然后，对 $\vec{u}_{00},\cdots,\vec{u}_{ij}$ 进行加密，并将卷积核矩阵转换为向量 $\vec{v}$，然后计算式 $m_{ij}=\sum_{i,j}\vec{u}_{ij}\vec{v}$。最后，将 m_{ij} 填充到新特征矩阵的对应位置，完成安全卷积计算。

2) 协议描述

安全卷积计算协议过程中的内积计算通过内积函数加密算法实现。首先 C 调用 IPFEA.Encrypt 对 $\vec{u}_{00},\cdots,\vec{u}_{ij}$ 进行加密，将加密后的密文发送给 CS，CS 接收密文的同时将持有的卷积核向量 $\vec{v}$ 发送给 TPA，向其申请与自身 $\vec{v}$ 相关联的函数密钥用于计算与密文的内积值，TPA 调用 IPFEA.KeyGen 生成与 $\vec{v}$ 相关联的函数密钥 $\mathrm{sk}_{\mathrm{IP},\vec{v}}$ 并返还给 CS，CS 接收 $\mathrm{sk}_{\mathrm{IP},\vec{v}}$ 并调用 IPFEA.Decrypt 解密得到密文与向量 $\vec{v}$ 的明文内积结果。在此协议的过程中，C 与 CS 只存在单向通信，即非交互式通信，既保证了 C 输入数据的隐私，也实现了在 CS 上的内积计算。协议 1 是对安全卷积计算协议的具体描述。

协议 1　安全卷积计算协议 Secure-Convolution ()

输入方 C　待加密特征矩阵 $\boldsymbol{X}'$，公钥 $\mathrm{pk}_{\mathrm{IP}}$

输入方 CS　卷积核权重矩阵 $\boldsymbol{V}$,公钥 $\mathrm{pk}_{\mathrm{IP}}$

输入方 TPA　公钥 $\mathrm{pk}_{\mathrm{IP}}$,主私钥 $\mathrm{msk}_{\mathrm{IP}}$

输出方 CS　卷积结果 $\boldsymbol{M}$

C: 初始化一个空窗口列表 CT

C: 在 $\boldsymbol{X}'$ 上抽象一个卷积核大小的窗口矩阵并转换为向量 $\vec{x}'$

C: 加密 $\vec{x}'$，得到 $\mathrm{ct}_{\mathrm{IP},\vec{x}}$

C: 赋值 $\mathrm{ct}_{\mathrm{IP},\vec{x}}$ 给 CT，发送 CT 给 CS

CS: 初始化矩阵 $\boldsymbol{M}$

CS: 将矩阵 $\boldsymbol{V}$ 转换为向量 $\vec{v}$，发送给 TPA

TPA: 生成私钥 $\mathrm{sk}_{\mathrm{IP},\vec{v}}$，并发送给 CS

CS: for i←1 to len(CT)

CS:　　解密 $\mathrm{CT}_{[i]}$，得到 m

CS: 填充卷积核矩阵 $\boldsymbol{M}$

CS: 得到 $\boldsymbol{M}$

安全卷积计算协议可以确保服务器在对加密数据进行卷积计算的同时无法访问输入数据，且内积函数加密算法也保证了服务器无法从内积结果中推断出原始数据，从而保证了数据的机密性。

3.4.2　安全损失优化协议

1) 设计思想

安全损失优化协议用于反向传播过程中发生在输出层的分类标签与真实标签间损失函数的乘法计算，以及计算梯度过程中涉及的减法计算。而计算过程中的真实标签来自客户端提供的需要隐私保护的标签数据，即安全损失优化协议包括对标签数据的加密、损失函数的安全计算以及损失函数对输出层的参数向量的偏导数的安全计算。

假设客户端 C 的真实标签经 One-hot 编码后表示为向量 $\vec{y}=(y_1,\cdots,y_n)$，服务器 CS 的分类标签表示为向量 $\vec{p}=(p_1,\cdots,p_n)$。由 C 对其真实标签向量进行加密后发送给 CS，CS 根据要执行的具体计算向 TPA 申请对应的函数密钥，CS 利用函数密钥对密文进行与其自身持有向量参数 $\vec{p}=(p_1,\cdots,p_n)$ 的特定计算，图 4 为安全损失优化协议中各实体间的通信时序图。

在安全损失优化协议中，C 对其真实标签向量 $\vec{y}=(y_1,\cdots,y_n)$ 使用随机映射进行混淆，通过构建

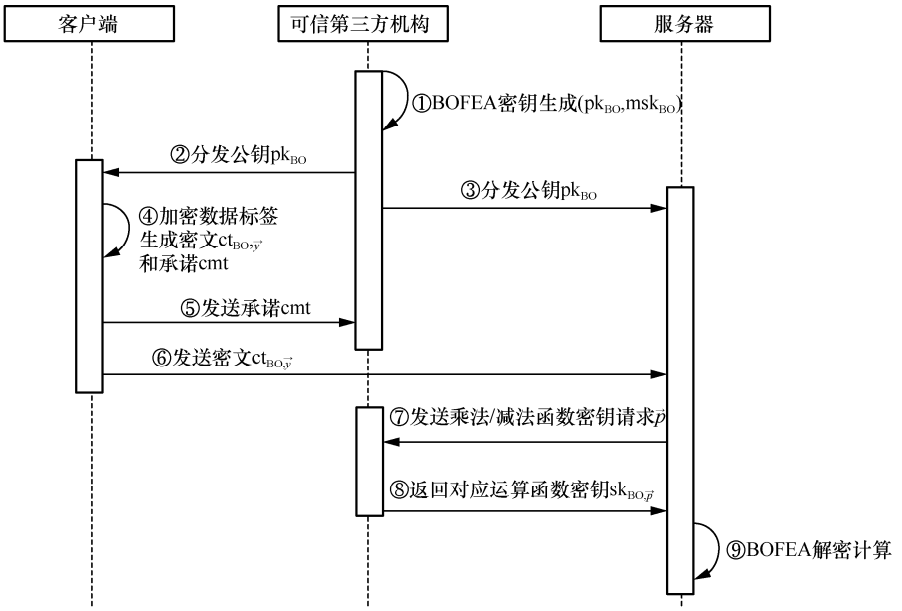

图 4 安全损失优化协议中各实体间的通信时序图

一个维度为 $n\times n$ 的映射矩阵 $\boldsymbol{D}$，将映射矩阵 $\boldsymbol{D}$ 与 $\vec{y}=(y_1,\cdots,y_n)$ 相乘，其标签向量将会被映射到一个新的向量中。

2) 协议描述

对映射到的新标签向量 $\vec{y}=(y_1,\cdots,y_n)$ 中每个元素 y_i 逐一执行 BOFEA.Encrypt 算法生成密文 $\mathrm{ct}_{\mathrm{BO},\vec{y}}$ 和一个承诺 cmt，承诺 cmt 发送给 TPA，将密文 $\mathrm{ct}_{\mathrm{BO},\vec{y}}$ 发送给 CS。在 CS 的解密计算中，首先接收密文 $\mathrm{ct}_{\mathrm{BO},\vec{y}}$ 并将与该密文进行计算的分类标签向量 $\vec{p}=(p_1,\cdots,p_n)$ 以及执行的运算操作 op 发送给 TPA 用于申请关联的函数密钥。TPA 接收来自 C 的承诺 cmt 和来自 CS 的 $\vec{p}=(p_1,\cdots,p_n)$ 以及 op，对 $\vec{p}=(p_1,\cdots,p_n)$ 中的每个元素 p_i 执行 BOFEA.KeyGen 算法生成对应 op 运算的函数密钥 $\mathrm{sk}_{\mathrm{BO},\vec{p}}$ 并发送给 CS。CS 接收 $\mathrm{sk}_{\mathrm{BO},\vec{p}}$ 并执行 BOFEA.Decrypt 算法得到对应的运算结果 $\vec{b}=(b_1,\cdots,b_n)$。损失函数的计算得到一个由每个类别的损失值组成的向量，而优化算法得到一个损失函数中每个元素梯度组成的向量，协议 2 是对安全损失优化协议的具体描述。

协议 2 安全损失优化协议 Secure-LossSGD ()

输入方 C 真实标签向量 $\vec{y}=(y_1,\cdots,y_n)$，公钥 $\mathrm{pk}_{\mathrm{BO}}$

输入方 CS 分类标签向量 $\vec{p}=(p_1,\cdots,p_n)$，运算操作 op，公钥 $\mathrm{pk}_{\mathrm{BO}}$

输入方 TPA 公钥 $\mathrm{pk}_{\mathrm{BO}}$，主私钥 $\mathrm{msk}_{\mathrm{BO}}$

输出方 CS 运算结果 b_i

C: 初始化矩阵 $\boldsymbol{D}$

C: for $i\leftarrow 1$ to n

C: 执行 BOFEA.Encrypt 算法生成密文 $\mathrm{ct}_{\mathrm{BO},y_i}$和$\mathrm{cmt}_i$

C: 将 $\mathrm{ct}_{\mathrm{BO},y_i}$和$\mathrm{cmt}_i$ 分别发送给 CS 和 TPA

CS: 将分类标签向量 $\vec{p}$ 发送给 TPA

TPA: 初始化列表 SK

TPA: for $i\leftarrow 1$ to n

TPA: 执行分类标签向量生成函数密钥 $\mathrm{sk}_{\mathrm{BO},p_i}$

TPA: 添加密钥 $\mathrm{sk}_{\mathrm{BO},p_i}$ 到列表 SK 中

TPA: 发送 SK 给 CS

CS: for $i\leftarrow 1$ to lengh ($\vec{p}$)

CS: 解密，得到 b_i

安全损失优化协议保护了标签的隐私，通过对标签进行随机映射后，再通过元素基本运算函数加密算法进行加密，避免了对标签的直接分类，而服务器只能得到数据与标签之间的映射关系。但由于数据与标签都受函数加密算法的保护，即使有些标签在数据中具有高相似性（例如一个标签映射到多个训练数据），加密的结果在密文空间中仍然是随机均匀分布的，服务器无法根据加密后的数据推断出任何关于原始数据和标签的私有信息。

3.5　安全训练

由于函数加密计算得到的是明文结果，将这一性质应用于 CNN 训练过程中需特别关注 2 个方面。首先是第一个卷积层的前向传播过程，即密文前向传播。其次是输出层的损失函数计算评估过程以及误差在输出层的反向传播过程，即密文反向传播。在其他层中，将通过从第一个卷积层和输出层计算出的明文结果，继续进行明文前向传播和反向传播过程，训练出完整的模型。

1) 密文前向传播

密文前向传播发生在 CNN 的第一个卷积层，在此层的前向传播过程中需要进行卷积计算，但在卷积计算过程中为更好地识别边缘，一般需要在输入的数据上进行填充操作。卷积计算过程中产生的滑动窗口如图 5 所示。

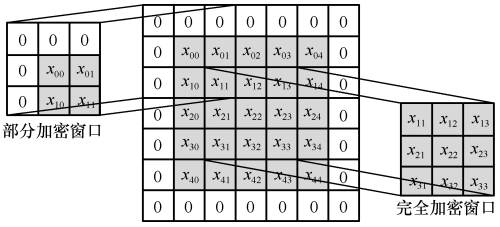

图 5　卷积计算过程中产生的滑动窗口

假设C从一张图片中提取出一个由特征向量构成的矩阵 $\boldsymbol{X}$，矩阵 $\boldsymbol{X}$ 经函数加密算法加密后形成阴影区域中的密文，如果由 CS 在该密文上进行填充操作会形成一个由明文“0”和密文组成的混合矩阵，之后再与卷积层相连的卷积核矩阵执行卷积计算，会产生 2 种类型的滑动窗口，即部分加密窗口和完全加密窗口。而与部分加密窗口进行卷积计算并非实现安全卷积计算，因此填充过程由 C 来执行。C 将其数据填充好后再对数据进行加密，发送给 CS。

在卷积计算过程中，涉及卷积核与滑动区域中密文矩阵之间的乘法计算，为了既保证输入数据的隐私性又实现卷积计算的功能，调用协议 1 解决卷积层中前向传播过程的安全问题。

2) 密文反向传播

密文反向传播过程发生在输出层，输出层的激活函数采用 Softmax 函数将每个神经元的输出转换为对应类别的概率，使整个输出向量呈概率分布的形式，这里 Softmax 函数如式(3)所示。

$$p_h = \frac{\mathrm{e}^{a_h}}{\sum_{k=1}^{N} \mathrm{e}^{a_k}} \tag{3}$$

其中，p_h 是对应输出层中第 h 个神经元的输出，即输入数据 $\boldsymbol{X}$ 属于类别 i 的概率；a_h 是前一层输出向量 $\vec{a}$ 的第 h 个元素；N 为输出层中神经元的个数，也是分类任务中类别的总数。然后，计算与真实标签相关的损失函数，使用式(4)的交叉熵损失函数以实现下一步对输出层梯度的计算。

$$\mathrm{Loss} = -\sum_{i=1}^{N} y_i \log p_i \tag{4}$$

其中，y_i 是真实标签向量的第 i 个元素，构成真实标签向量 $\vec{y} = (y_1, \cdots, y_n)$；$p_i$ 是经 Softmax 函数得到的输出，即标签向量中的第 i 个元素，构成标签向量 $\vec{p} = (p_1, \cdots, p_n)$。根据式(5)计算出损失函数对于输出层中第 i 个神经元的梯度 δ_i。

$$\begin{aligned}
&\frac{\partial \mathrm{Loss}}{\partial a_h} = \frac{\partial \mathrm{Loss}}{\partial p_i}\frac{\partial p_i}{\partial a_h} \\
&\frac{\partial p_i}{\partial a_i} = \begin{cases} p_i(1-p_h), & i = h \\ -p_h p_i, & i \neq h \end{cases} \\
&\delta_i = \frac{\partial p_i}{\partial a_h} = p_i - y_i
\end{aligned} \tag{5}$$

密文反向传播的计算过程涉及乘法和减法运算，通过调用协议 2 实现安全反向传播过程。

3) 密文训练过程

密文训练包括密文前向传播、明文前向传播、密文反向传播、明文反向传播 4 个过程，对密文训练集进行多次的迭代训练，即上述 4 个过程顺序地进行迭代，直到满足训练条件。

假设 CS 训练一个包含 L 层的 CNN 模型，该 CNN 模型包括一个输入层、至少一个卷积层、一个输出层以及数量不定的激活层、池化层和全连接层，并将激活层的激活函数表示为 ReLU()，将池化层的函数表示为 Maxpool()，将输出层的激活函数表示为 Softmax()，使用的损失函数选用交叉熵损失函数 Loss()，然后将训练终止条件设置为达到一定迭代次数 e。

算法 1 是对 CNNM-ED 安全训练的详细描述。

算法 1　CNNM-ED 安全训练

输入方 C　待加密数据 $\boldsymbol{X}$,待加密标签 $\vec{y}$,公钥 $\mathrm{pk}_{\mathrm{IP}}$,公钥 $\mathrm{pk}_{\mathrm{BO}}$

输入方 CS　网络层数 $l \in [1, L]$,迭代次数 e,公钥 $\mathrm{pk}_{\mathrm{IP}}$,公钥 $\mathrm{pk}_{\mathrm{BO}}$

输出方 CS　模型权重 $W^{[l]}$ 和偏置 $b^{[l]}$

CS: 根据网络层数初始化每个网络层权重 $W^{[l]}$ 和偏置 $b^{[l]}$

CS: for 1 to e

CS: for l =1 to L

CS: if $l=1$

C、CS: 调用协议 1，执行密文前向传播，得到 $M^{[l]}$

CS: 计算 $\pi^{[l]} \leftarrow M^{[l]} + b^{[l]}$

CS: if $l \neq 1 \wedge l \neq L$

CS: 执行明文前向传播，得到 $\pi^{[l]}$

CS: if $l = L$

CS: 执行 Softmax 函数，计算 $\vec{k}$

CS: 使用 $\log \vec{k}$ 函数与真实标签比较得到 $\hat{y}$

C、CS: 调用协议 1 执行密文反向传播，计算损失值 L

CS: for $l = L-1$ to 2

CS: 执行明文反向传播更新各层权重 $W^{[l]}$ 和偏置参数 $b^{[l]}$

3.6 安全分类

在CNNM-ED中实现对加密数据的分类与安全训练阶段中密文前向传播的过程基本一致，客户端与服务器需要执行安全卷积计算协议，并在已训练好的模型上对加密数据进行分类。

在执行 CNN 安全分类之前，CS 要加载训练好的分类模型的参数，完成模型的初始化。其中，C 是分类过程的请求者，拥有待分类数据 $\boldsymbol{X}$ 和加密其数据的公钥 $\mathrm{pk_{IP}}$；CS 是分类任务的执行者，拥有已训练好模型中各网络层的参数 $W^{[l]}$，$b^{[l]}$（其中 $l \in [1, L]$），以及公钥 $\mathrm{pk_{IP}}$。在分类过程中，需要通过安全卷积计算协议实现 C 数据的加密与 CS 中卷积层参数即 $W^{[l]}$ 在 C 加密数据上的安全卷积计算，算法 2 是对 CNNM-ED 安全分类过程的详细描述。

算法 2 CNNM-ED 安全分类

输入方 C 待分类数据 $\boldsymbol{X}$，公钥 $\mathrm{pk_{IP}}$

输入方 CS 模型各层权重参数 $W^{[l]}$ 和偏置参数 $b^{[l]}$,公钥 $\mathrm{pk_{IP}}$

输出方 CS 分类标签 t

CS: 获取模型第一层权重参数

C、CS: 调用协议 1 执行安全卷积计算，得到第一层的结果 $\pi^{[l]} \leftarrow M^{[l]} + b^{[l]}$

CS: if $l \neq 1$

CS: for $2 \leqslant l \leqslant L$

CS: if $l \neq L$

CS: 根据 ReLU 函数计算非输出层的结果 $\pi^{[l]}$

CS: else

CS: 根据 Softmax 函数计算输出层的结果 $\vec{t}$

CS: 向量 $\vec{t}$ 中的最大索引为分类标签 t

4 安全性分析

本节分析 IPFEA 和 BOFEA 的安全性，并使用以下定理证明这 2 个算法是安全的。

定理 1 基于 DDH 假设构造的 IPFEA 具有 CPA 安全性。

证明

游戏 0 在游戏中，挑战者 $\mathcal{C}$ 首先生成公私钥对，然后将公钥 $\mathrm{pk_{IP}}$ 发送给敌手 $\mathcal{A}$，在挑战阶段，敌手 $\mathcal{A}$ 选择 2 个不同的向量 $\vec{x}_0$ 和 $\vec{x}_1$，可以向挑战者提交关于向量 $\vec{y}$ 的查询，要求满足 $f(\vec{x}_0, \vec{y}) = f(\vec{x}_1, \vec{y})$，然后挑战者 $\mathcal{C}$ 随机选择一个比特 $\beta \in \{0,1\}$，并使用公钥 $\mathrm{pk_{IP}}$ 加密向量 $\vec{x}_{\mathrm{IP},\beta}$，生成密文，并将密文发送给敌手 $\mathcal{A}$，游戏结束时，敌手 $\mathcal{A}$ 输出一个比特 β'，定义事件 S_0 为 $\beta = \beta'$ 时的事件。

游戏 1 修改挑战密文 $\mathrm{ct}_{\mathrm{IP},\vec{x}_\beta} = (\mathrm{ct}_0, \{\mathrm{ct}_i\}_{i\in[1,\cdots,\mu]})$ 的生成。挑战者首先随机选择 $r \leftarrow \mathbb{Z}_p$，计算 $\mathrm{ct}_0 = g^r$，然后，利用主私钥 $\mathrm{msk_{IP}} = (s_1, \cdots, s_\mu)$ 对于每个 $i \in [1,\cdots,\mu]$，计算 $\mathrm{ct}_i = g^{x_{\beta,i}} \mathrm{ct}_0^{s_i}$，显然 $\mathrm{ct}_{\mathrm{IP},\vec{x}_\beta}$ 的分布同游戏 0 中挑战密文的分布完全一致，因此 $\Pr[S_1] = \Pr[S_0]$。

游戏 2 再次修改挑战密文 $\mathrm{ct}_{\mathrm{IP},\vec{x}_\beta} = (\mathrm{ct}_0, \{\mathrm{ct}_i\}_{i\in[1,\cdots,\mu]})$ 的生成方式。挑战者随机选择 $r, r' \leftarrow \mathbb{Z}_p$，令 $\mathrm{ct}_0 = g^r$，计算 $\mathrm{ct}_i = g^{x_{\beta,i}} (\mathrm{ct}_0 g^{r'})^{s_i}$。在 DDH 困难假设下，此修改没有对敌手 $\mathcal{A}$ 的视图造成显著的影响。因此，有 $\Pr[S_2] - \Pr[S_1] \leqslant \mathrm{Adv}_\beta^{\mathrm{DDH}}(\lambda)$。继续证明在游戏 2 中挑战密文 $\mathrm{ct}_{\mathrm{IP},\vec{x}_\beta} = (\mathrm{ct}_0, \{\mathrm{ct}_i\}_{i\in[1,\cdots,\mu]})$ 隐藏了 $\beta \in \{0,1\}$，用于表明 $\Pr[S_1] = \dfrac{1}{2}$。由上述 ct_i 的计算，可得

$$g^{x_{\beta,i}} (\mathrm{ct}_0 g^{r'})^{s_i} = g^{x_{\beta,i}} g^{(r+r')s_i} = g^{x_{\beta,i}+r's_i} h_i^{s_i} \tag{6}$$

式(6)表示一个具备无限计算能力的敌手只能够从密文 $\mathrm{ct}_{\mathrm{IP},\vec{x}_\beta} = (\mathrm{ct}_0, \{\mathrm{ct}_i\}_{i\in[1,\cdots,\mu]})$ 中推断出

$\vec{z}_\beta=(x_{\beta,1}+r's_1,\cdots,x_{\beta,\mu}+r's_\mu)\in\mathbb{Z}_p^\mu$。为了证明$\vec{x}_{\mathrm{IP},\beta}$对于任意合法的密文来说没有泄露任何关于$\beta\in\{0,1\}$的信息，定义$\vec{x}=\vec{x}_0-\vec{x}_1$，然后以确定性的方式基于$\mu-1$维的子空间$\vec{x}^\top=\{\vec{y}\in\mathbb{Z}_p^\mu\mid f(\vec{x},\vec{y})\}$中生成矩阵$\boldsymbol{Y}_{\mathrm{top}}\in\mathbb{Z}_p^{(\mu-1)\times\mu}$，令向量$\vec{x}'\in\mathbb{Z}_p^\mu$为子空间$\vec{x}^\top$中以确定性方式选择出的向量，定义可逆矩阵为

$$\boldsymbol{Y}=\frac{\boldsymbol{Y}_{\mathrm{top}}}{\vec{x}'^\top}\in\mathbb{Z}_P^{\mu\times\mu}\tag{7}$$

由于$\boldsymbol{Y}$的行向量由$\vec{x}\in\mathbb{Z}_p^\mu$中生成，且被敌手$\mathcal{A}$所知。因此，如果能够证明$\boldsymbol{Y}\vec{z}_\beta\in\mathbb{Z}_p^\mu$与$\beta$独立且安全，则能够证明$\vec{x}_\beta$没有泄露任何关于$\beta\in\{0,1\}$的信息。显然$\boldsymbol{Y}\vec{z}_\beta\in\mathbb{Z}_p^\mu$的前$\mu-1$行与$\beta$独立，因此只需要证明$\boldsymbol{Y}\vec{z}_\beta\in\mathbb{Z}_p^\mu$的最后一行，即$\vec{x}'\vec{z}_\beta=<\vec{x}'\vec{x}_\beta>+r'<\vec{x}',(s_1,\cdots,s_\mu)>$。令$s_0\in\mathbb{Z}_p^\mu$为任意向量，满足$(h_1,\cdots,h_\mu)=g^{s_0}$，且对所有的查询$\vec{y}$，令$\mathrm{sk}_{\mathrm{IP},\vec{y}}=<(s_1,\cdots,s_\mu),\vec{y}>$，由于所有的查询$\vec{y}$都在$\vec{x}^\top$中，因此，在敌手$\mathcal{A}$的视图下，主私钥$\mathrm{msk}_{\mathrm{IP}}=(s_1,\cdots,s_\mu)$的分布为$\{s_0+\omega\vec{x}\mid\omega\leftarrow\mathbb{Z}_p\}$，其中$\omega$为从$\mathbb{Z}_p$中均匀选择的随机数。因此$r'<\vec{x}',(s_1,\cdots,s_\mu)>$的条件分布是$\{r'<\vec{x}',(s_1,\cdots,s_\mu)>+r'\omega<\vec{x}',\vec{x}>\mid\omega\leftarrow\mathbb{Z}_p\}$。根据上述构造，$<\vec{x}',\vec{x}>\neq0$，因此$r'<\vec{x}',(s_1,\cdots,s_\mu)>$的分布是均匀分布，同时，由于$r'\neq0$，这意味着$r'<\vec{x}',(s_1,\cdots,s_\mu)>$隐藏了$\vec{x}'\vec{z}_\beta$内积中的$<\vec{x}'\vec{x}_\beta>$项。综上所述，定理 1 成立，IPFEA 是 CPA 安全的。

证毕。

定理 2　基于 DDH 假设构造的 BOFEA 具有 CPA 安全性。

证明　假设存在一个敌手$\mathcal{A}$，能够以不可忽略的优势破坏 BOFEA 的 CPA 安全性，若无法以不可忽略的优势破坏，则该算法是 CPA 安全的。假设敌手$\mathcal{A}$在时间t上执行，并且具有优势ε，此时通过构造一个敌手$\mathcal{B}$来攻击 DDH 假设，该敌手$\mathcal{B}$在时间$t+O(1)$上执行，并且也具有优势ε，该敌手$\mathcal{B}(g,h_1=g^a,h_2=g^r,h_3=g^c)$的构造如下。

初始化阶段　设置公钥 pk 为(g,h_1)，从$\{0,1\}$中随机选取b，并将密文c设置为$(h_2,h_3g^{x\bigcirc y})$。

询问阶段 1　加密算法挑战者$\mathcal{C}$输入要执行具体运算的k_i，运行 BOFEA 中的 KenGen() 算法，同样将生成对应的函数密钥$\mathrm{sk}_{\vec{k}_i}$发布给敌手$\mathcal{A}$。

挑战阶段　挑战者$\mathcal{C}$随机选择一个$\beta\in\{0,1\}$，将公钥$\mathrm{pk}_{\mathrm{BO}}$作为输入，执行的算法为 BOFEA 中的 Encrypt() 算法。

询问阶段 2　敌手$\mathcal{A}$继续发起与询问阶段 1 相同的函数密钥请求。

猜测阶段　利用敌手$\mathcal{A}$执行$\mathcal{A}(\mathrm{pk}_{\mathrm{BO}},c)$得到输出的猜测$b'$。如果$b=b'$，则敌手$\mathcal{B}$猜测该输入的密文为有效的 DDH 元组，否则为随机的 DDH 元组。

如果$c=ra$，那么密文c就是$g^{x\bigcirc y}$的合法加密，但如果c是均匀分布的，与a和r无关，那么密文就是原始输出。在这种情况下敌手$\mathcal{B}$也具有优势ε打破 DDH 假设，这违反了预先的安全假设，因此敌手$\mathcal{A}$并不具有不可忽视的优势来破坏 BOFEA，则定理 2 成立，在 DDH 假设下，BOFEA 是 CPA 安全的。证毕。

综上所述，IPFEA 和 BOFEA 具有 CPA 安全性。CNNM-ED 的安全性分析主要涵盖了 IPFEA 和 BOFEA 的安全性，以及在隐私与效率之间的权衡，使服务器可以在加密的数据和标签上执行服务器中卷积神经网络模型的训练，且在整个过程中都无法解密出用户数据的原始内容。因此，本文所提的 CNNM-ED 模型能够保证数据与标签的机密性。

5　实验及分析

5.1　实验设置

本文开展实验的相关环境信息如表 1 所示。实验过程中主要利用 Charm-crypto-0.43 库生成循环群，实现大整数运算、处理模运算等，利用 Numpy 库在实验设置的指定范围内随机生成用于实验测试的整数数据。

表 1　　**实验环境**

环境	设备	参数
硬件	CPU	Intel(R) Xeon(R) Silver 4110 CPU@ 2.10 GHz 64 GB
软件	操作系统	Windows
	编译环境库	Charm-crypto-0.43、Numpy

采用 MNIST 数据集评估 CNNM-ED 中训练和分类的准确率以及效率。该数据集是一组标准的手写数字数据集[31]，含有 6 0000 张训练样本以及

10 000 张测试样本，其中每张样本都是一个 28 像素×28 像素的灰度图像，且像素值在 0 到 255 之间，代表一个手写数字。每张样本都有一个对应的标签，表示图像中手写数字的真实类别（0～9），在卷积神经网络中意味着每张图像样本具有 784 个特征值。

由于本文提出的函数加密算法只支持有限域内整数的数值运算，因此在加密数据前需要将输入的数据转换为整数形式。考虑将原输入的浮点数转换成一定精度的浮点数，这里设置了精度参数 ε，然后将需要加密计算的数据乘以 10^{ε}，最后经四舍五入得到整数。在不损失太多精度的情况下，将浮点数转换为有限域内的整数，进行解密计算之后再将结果除以精度参数 ε 即可得到实际计算结果。

利用 Python 的 Numpy 科学计算库搭建了一个用于多重分类的卷积神经网络，其结构如表 2 所示，网络结构中除输出层以外其他层的激活函数均采用 ReLU 函数，输出层采用 Softmax 函数。在训练完成后，将明文和密文数据训练出来的模型使用测试集分别进行评估，并记录准确率和训练时间。

表 2 卷积神经网络结构

CNN 层	详细描述
输入层	图像输入大小为 32× 32× 1
卷积层 1	6 个 5× 5 大小的卷积核，步长为 1，输出尺寸为 28× 28× 6
池化层 1	2× 2 大小的最大池化窗口，步长为 2，输出尺寸为 14× 14× 6
卷积层 2	16 个 5× 5 大小的卷积核，步长为 1，输出尺寸为 10× 10× 16
池化层 2	2× 2 大小的最大池化窗口，步长为 2，输出尺寸为 5× 5× 16
卷积层 3	120 个 5× 5 大小的卷积核，步长为 1，输出尺寸为 1× 1× 120
全连接层 1	120 个神经元，输入尺寸为 120，输出尺寸为 120
全连接层 2	84 个神经元，输入尺寸为 120，输出尺寸为 84
输出层	10 个神经元，输入尺寸为 84，输出尺寸为 10

5.2 准确率分析

本节将 CNNM-ED 模型与明文数据在没有任何隐私保护的原始 CNN 上进行对比，在具有相同结构的 CNN 上进行训练，研究所提模型的准确率。

使用随机梯度下降法将全部 60 000 张训练样本在构造的 CNN 上训练 2 个 epoch，其中 batch 大小为 64，学习率为 0.000 5，精度 ε 为 3，函数加密算法中输入的安全参数为 128。通过 CNNM-ED 训练出的模型与原始 CNN 训练出的模型准确率和训练时间对比如表 3 所示，各模型的平均批处理准确率如图 6 所示。

表 3 CNNM-ED 与原始 CNN 的准确率和训练时间对比

模型	准确率（epoch1）	准确率（epoch2）	训练时间/h
原始 CNN	93.35%	95.79%	4.8
CNNM-ED	92.93%	95.3%	59

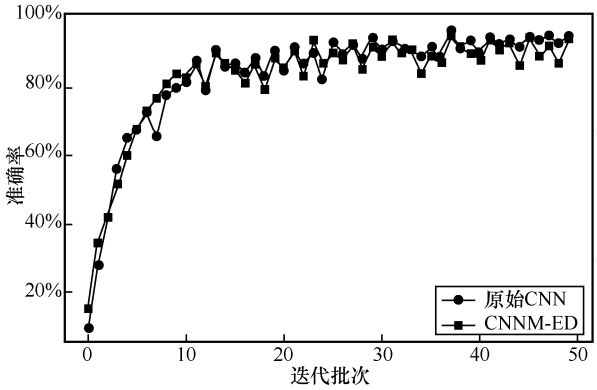

图 6 各模型的平均批处理准确率

根据上述实验可以得出结论，CNNM-ED 与原始 CNN 相比具有相似的准确率，均达到 90%以上。由于卷积层的密文前向传播和输出层的密文反向传播中的解密计算比较耗时，训练时间比原始 CNN 要更久，但实际上这种耗时是在解密算法中计算离散对数时产生的，耗费时间比在明文上训练模型多了 12 倍左右。

5.3 效率分析

通过设置一个预先存储的有界哈希表来加速解密过程中耗时的计算小整数的离散对数过程，以缩短模型训练的时间，从而实现不同的实验设置，生成的有界哈希表部分数据如表 4 所示。

表 4 有界哈希表部分数据

区间	公钥参数 g	$<\vec{x},\vec{y}>$	$g^{<\vec{x},\vec{y}>}$
{−25 000 000, 25 000 000}	1 045 899 731	−25 000 000	3 445 551 739
		−24 999 999	759 032 512
		−24 999 998	305 764 829
		⋮	⋮
		−1	2 496 402 232
		1	1 045 899 731
		⋮	⋮
		24 999 998	1 723 447 672
		24 999 999	2 319 590 661
		25 000 000	2 023 263 626

表 4 存储了加密算法中的指定公钥参数 g 和所有的 $(<\vec{x},\vec{y}>, g^{<\vec{x},\vec{y}>})$ 以及 $(x\bigcirc y, g^{x\bigcirc y})$。其中，$<\vec{x},\vec{y}>$ 表示特征向量与卷积核参数的内积结果，$x\bigcirc y$ 表示真实标签与分类标签的计算结果，2 个结果都在一个有限的区间范围内。根据构造的网络和数据的精度，选择将该区间设置为{−25 000 000, 25 000 000}，且将加密算法的安全参数 λ 设置为 32 bit，生成有界哈希表的大小约为 1.1 GB，预计算该哈希表的时间为 239.6 s。

为了评估训练过程中各个阶段的执行时间和通信开销，在 MNIST 数据集中随机选择一个样本作为训练样本，在构造的 CNN 上进行训练，分别记录客户端预处理加密传输、TPA 生成函数密钥、云服务器申请函数密钥并解密计算所消耗的时间和产生的通信带宽，如表 5 和表 6 所示。

表 5　CNNM-ED 单个样本训练的各阶段执行时间

类别	预处理加密传输/s	函数密钥申请和生成/s	解密/s
图像	0.089	0.018	0.251
标签	0.072	0.009	0.021

表 6　CNNM-ED 单个样本训练的通信开销

通信方	阶段	通信带宽/KB
客户端	预处理加密传输	18.72
云服务器	申请函数密钥并解密	0.146
第三方可信机构	生成函数密钥	0.569
	总计	19.435

由表 6 中数据可以观察到，在 CNNM-ED 训练过程中主要产生的通信开销是客户端在本地预处理加密数据后传输密文到服务器所产生的开销，而这种开销在数据传输中不可避免，且在可以接受的范围内。而在模型训练过程中，云服务器与第三方可信机构间函数密钥申请与分发所消耗的带宽相对较少，这是因为申请函数密钥仅需利用卷积核参数与分类结果参数作为申请执行线性计算，并没有因交互产生大量额外的通信开销。

为了评估网络结构的变化对模型训练时间的影响，在具有相同数量的 MNIST 数据集上训练具有不同网络结构的 CNN 模型。基于表 2 中的网络结构，增加第一个卷积层中卷积核的数量，记为 No1CNN；增加一个新的卷积层，记为 No2CNN。这 2 个网络结构中输入的数据大小均为 $32\times32\times1$。不同网络结构的CNNM-ED模型与明文模型的训练时间对比如图 7 所示。从图 7 可以看出，在未加密的数据上训练不同的卷积神经网络模型，其训练时间与卷积核数量成反比，与卷积层数量成正比，当卷积核数量较多时，训练时间也较短；当卷积层数量增多时，其模型训练时间也增加。而 CNNM-ED 在加密数据上通过不同网络结构训练的时间与在未加密数据上训练的时间规律一致，则可以得出结论，当网络结构发生变化时，CNNM-ED 与没有隐私保护设置的正常神经网络相比，训练时间没有因在加密数据上训练而产生其他规律的变化。

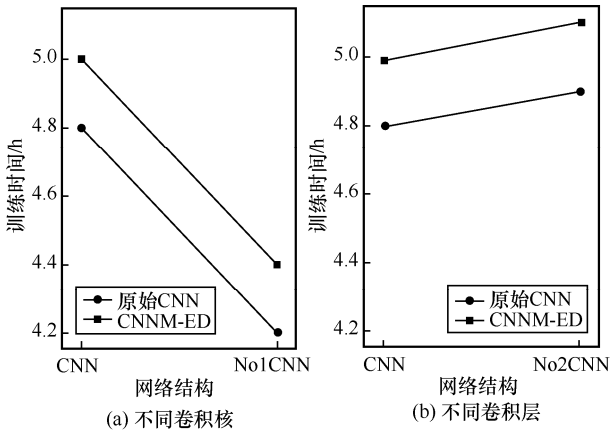

图 7　不同网络结构的 CNNM-ED 模型与明文模型的训练时间对比

尽管在不同结构的网络上进行训练，CNNM-ED 模型与未加密数据仍具有相似的准确率，如图 8 所示，该实验结果也进一步验证了本文提出的 CNNM-ED 模型的性能。

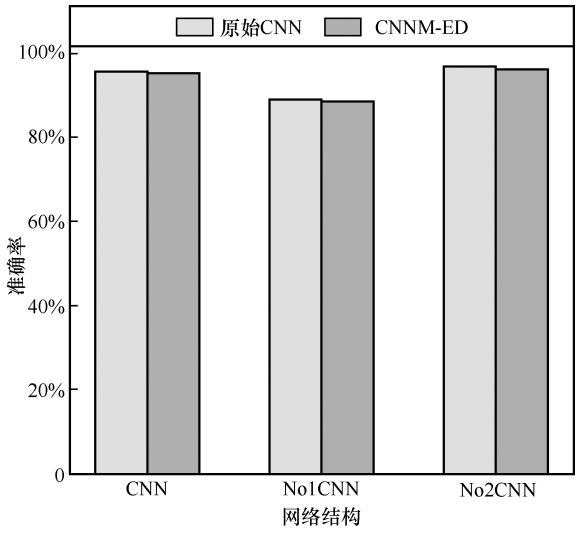

图 8　不同网络结构的 CNNM-ED 模型与明文模型的准确率对比

6 结束语

本文提出了一种面向密文数据的卷积神经网络模型 CNNM-ED，实现了密文卷积神经网络的训练和分类。为了在密文下进行卷积神经网络计算，设计了内积函数和基本运算函数加密算法。基于 2 种函数加密算法，设计了安全卷积计算协议和安全损失优化协议，实现了在卷积神经网络中卷积层的安全卷积计算和输出层的安全损失函数和梯度计算。基于上述协议构建了 CNNM-ED，能够保护数据以及数据中标签的隐私。实验结果表明，本文提出的模型可以在密文数据下实现 CNN 的训练和分类，同时保证了训练模型和分类结果的准确率。

参考文献：

[1] KUMAR R, JOSHI S, DWIVEDI A. CNN-SSPSO: a hybrid and optimized CNN approach for peripheral blood cell image recognition and classification[J]. International Journal of Pattern Recognition and Artificial Intelligence, 2021, 35(5): 2157004.

[2] PRASETYO M L, WIBOWO A T, RIDWAN M, et al. Face recognition using the convolutional neural network for barrier gate system[J]. International Journal of Interactive Mobile Technologies (IJIM), 2021, 15(10): 138-153.

[3] WAN C X, LI B. Financial causal sentence recognition based on BERT-CNN text classification[J]. The Journal of Supercomputing, 2022, 78(5): 6503-6527.

[4] YANG X A, CHEN J, HE K, et al. Efficient privacy-preserving inference outsourcing for convolutional neural networks[J]. IEEE Transactions on Information Forensics and Security, 2023, 18: 4815-4829.

[5] FALCETTA A, ROVERI M. Privacy-preserving deep learning with homomorphic encryption: an introduction[J]. IEEE Computational Intelligence Magazine, 2022, 17(3): 14-25.

[6] HUANG K, LIU X M, FU S J, et al. A lightweight privacy-preserving CNN feature extraction framework for mobile sensing[J]. IEEE Transactions on Dependable and Secure Computing, 2021, 18(3): 1441-1455.

[7] 朱宗武, 黄汝维. 基于高效全同态加密的安全多方计算协议[J]. 计算机科学, 2022, 49(11): 345-350.
ZHU Z W, HUANG R W. Secure multi-party computing protocol based on efficient fully homomorphic encryption[J]. Computer Science, 2022, 49(11): 345-350.

[8] 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 Press, 2016: 308-318.

[9] HUANG X X, GUAN J, ZHANG B, et al. Differentially private convolutional neural networks with adaptive gradient descent[C]//Proceedings of the 2019 IEEE Fourth International Conference on Data Science in Cyberspace (DSC). Piscataway: IEEE Press, 2019: 642-648.

[10] ALMUTAIRI N, COENEN F, DURES K. K-means clustering using homomorphic encryption and an updatable distance matrix: secure third party data clustering with limited data owner interaction[C]//Proceedings of the International Conference on Big Data Analytics and Knowledge Discovery. Berlin: Springer, 2017: 274-285.

[11] JÄSCHKE A, ARMKNECHT F. Unsupervised machine learning on encrypted data[C]//International Conference on Selected Areas in Cryptography. Berlin: Springer, 2019: 453-478.

[12] BOOMIJA M D, KASMIR RAJA S V. Securing medical data by role-based user policy with partially homomorphic encryption in AWS cloud[J]. Soft Computing, 2023, 27(1): 559-568.

[13] BONEH D, SAHAI A, WATERS B. Functional encryption: definitions and challenges[C]//Theory of Cryptography Conference. Berlin: Springer, 2011: 253-273.

[14] 于诗文. 基于安全多方计算的卷积神经网络模型研究与实现[D]. 西安: 西安电子科技大学, 2021.
YU S W. Research and implementation of convolutional neural network model based on secure multi-party computation[D]. Xi'an: Xidian University, 2021.

[15] WAGH S, GUPTA D, CHANDRAN N. SecureNN: 3-party secure computation for neural network training[J]. Proceedings on Privacy Enhancing Technologies, 2019(3): 26-49.

[16] ROUHANI B D, RIAZI M S, KOUSHANFAR F. DeepSecure: scalable provably-secure deep learning[C]//Proceedings of the 2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC). Piscataway: IEEE Press, 2018: 1-6.

[17] MISHRA P, LEHMKUHL R, SRINIVASAN A, et al. Delphi: a cryptographic inference system for neural networks[C]//Proceedings of the 2020 Workshop on Privacy-Preserving Machine Learning in Practice. New York: ACM Press, 2020: 27-30.

[18] ZHANG Q, WANG C, WU H Y, et al. GELU-net: a globally encrypted, locally unencrypted deep neural network for privacy-preserved learning[C]//Proceedings of the 27th International Joint Conference on Artificial Intelligence. New York: ACM Press, 2018: 3933-3939.

[19] DOWLIN N, GILAD-BACHRACH R, LAINE K, et al. CryptoNets: applying neural networks to encrypted data with high throughput and accuracy[C]//Proceedings of the 33rd International Conference on International Conference on Machine Learning. New York: ACM Press, 2016: 201-210.

[20] SEI Y, OKUMURA H, OHSUGA A. Privacy-preserving publication of deep neural networks[C]//Proceedings of the 2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS). Piscataway: IEEE Press, 2016: 1418-1425.

[21] BADAWI A A, JIN C, LIN J, 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.

[22] WENG J S, WENG J, TANG G, et al. pvCNN: privacy-preserving and verifiable convolutional neural network testing[J]. IEEE Transactions on Information Forensics and Security, 2023, 18: 2218-2233.

[23] KIM D, GUYOT C. Optimized privacy-preserving CNN inference with fully homomorphic encryption[J]. IEEE Transactions on Information Forensics and Security, 2022, 18: 2175-2187.

[24] HERNANDEZ M N J, MOLLER M, HANSEN S, et al. On fully

homomorphic encryption for privacy-preserving deep learning[C]//Proceedings of the 2019 IEEE Globecom Workshops (GC Wkshps). Piscataway: IEEE Press, 2019: 1-6.

[25] SANS E D, GAY R, POINTCHEVAL D. Reading in the dark: classifying encrypted digits with functional encryption[J]. IACR Cryptology ePrint Archive, 2018, 2018: 206.

[26] RYFFEL T, DUFOUR-SANS E, GAY R, et al. Partially encrypted machine learning using functional encryption[C]//Proceedings of the 33rd International Conference on Neural Information Processing Systems. Piscataway: IEEE Press, 2019: 4517-4528.

[27] XU R H, JOSHI J, LI C. NN-EMD: efficiently training neural networks using encrypted multi-sourced datasets[J]. IEEE Transactions on Dependable and Secure Computing, 2022, 19(4): 2807-2820.

[28] PANZADE P, TAKABI D. Towards faster functional encryption for privacy-preserving machine learning[C]//Proceedings of the 2021 Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA). Piscataway: IEEE Press, 2021: 21-30.

[29] SHAMIR A. Identity-based cryptosystems and signature schemes[C]//Advances in Cryptology-CRYPTO. Berlin: Springer, 1985: 47-53.

[30] SAHAI A, WATERS B. Fuzzy identity-based encryption[C]//Advances in Cryptology–EUROCRYPT 2005. Berlin: Springer, 2005: 457-473.

[31] LECUN Y, BOTTOU L, BENGIO Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.

[作者简介]

王琛（1996− ），女，辽宁锦州人，东北大学博士生，主要研究方向为隐私计算、密码学等。

李佳润（1998− ），女，辽宁沈阳人，东北大学硕士生，主要研究方向为机器学习、隐私保护等。

徐剑（1978− ），男，辽宁沈阳人，博士，东北大学教授、博士生导师，主要研究方向为网络与信息安全、隐私计算、AI 安全等。