第 48 卷 第 5 期　　自 动 化 学 报　　Vol. 48, No. 5
2022 年 5 月　　ACTA AUTOMATICA SINICA　　May, 2022

支持数据隐私保护的联邦深度神经网络模型研究

张泽辉[1]　富 瑶[1]　高铁杠[1]

摘　要　近些年, 人工智能技术已经在图像分类、目标检测、语义分割、智能控制以及故障诊断等领域得到广泛应用, 然而某些行业 (例如医疗行业) 由于数据隐私的原因, 多个研究机构或组织难以共享数据训练联邦学习模型. 因此, 将同态加密 (Homomorphic encryption, HE) 算法技术引入到联邦学习中, 提出一种支持数据隐私保护的联邦深度神经网络模型 (Privacy-preserving federated deep neural network, PFDNN). 该模型通过对其权重参数的同态加密保证了数据的隐私性, 并极大地减少了训练过程中的加解密计算量. 通过理论分析与实验验证, 所提出的联邦深度神经网络模型具有较好的安全性, 并且能够保证较高的精度.

关键词　联邦学习, 深度学习, 数据隐私, 同态加密, 神经网络

引用格式　张泽辉, 富瑶, 高铁杠. 支持数据隐私保护的联邦深度神经网络模型研究. 自动化学报, 2022, **48**(5): 1273−1284

DOI　10.16383/j.aas.c200236

Research on Federated Deep Neural Network Model for Data Privacy Preserving

ZHANG Ze-Hui[1]　FU Yao[1]　GAO Tie-Gang[1]

Abstract　In recent years, artificial intelligence technology has been widely used in the fields of image classification, object detection, semantic segmentation, intelligent control and fault diagnosis, etc.. However, in some industries, such as medical, it is difficult that multiple research institutions share data to train federated learning models due to data privacy. Therefore, homomorphic encryption (HE) algorithm technology is introduced into federated learning in this paper. A privacy-preserving federated deep neural network (PFDNN) model is proposed, which preserves data privacy by homomorphic encryption of model parameters and significantly reduces the amount of computation required for encryption and decryption. Theoretical analysis and experimental results show that the proposed federated deep neural network model has better security and can guarantee higher accuracy.

Key words　Federated learning, deep learning, data privacy, homomorphic encryption (HE), neural network

Citation　Zhang Ze-Hui, Fu Yao, Gao Tie-Gang. Research on federated deep neural network model for data privacy preserving. *Acta Automatica Sinica*, 2022, **48**(5): 1273−1284

近年来, 人工智能技术已经在图像分类、目标检测、语义分割、智能控制以及故障诊断等领域取得了优秀的成果[1−7]. 在解决某些特殊问题方面, 深度学习算法已经逼近甚至超过人类水平. 深度学习技术的快速发展主要依赖于丰富的数据集、算法的创新和运算设备性能的大幅提升[8]. 其中, 数据集的丰富程度对深度学习模型的性能水平产生直接影响[9−11]. 但是, 某些行业考虑到数据隐私泄露的问题, 难以共享数据进行集中式学习. 例如医疗行业在数据共享的过程中, 某些病人的信息可能会泄露到不法分子手中, 不法分子则利用患者信息推销非法药品、谋财害命[12−14]. 因此, 为进一步提升模型性能水平, 能够组织多个研究机构通过共享本地模型参数的方式, 实现协同训练全局模型的联邦学习算法被提出.

针对机器学习中数据隐私泄露的问题, 一些隐私保护的方法被提出, 主要可以分为以安全多方计算 (Secure multiparty computation, SMPC)、同态加密 (Homomorphic encryption, HE) 为代表的基于加密的隐私保护方法和以差分隐私 (Differential privacy, DP) 为代表的基于扰动的隐私保护方法.

安全多方计算是指两个或者多个持有私有数据的参与者通过联合计算得到输出, 并且满足正确性、隐私性、公平性等特性. Bonawitz 等[15] 提出一种基于秘密共享的安全多方计算协议, 旨在保证设备与服务端之间通信, 并可以用于联邦学习的参数聚合过程. 与传统密码学方法相比, 该协议的优点在于其计算代价并不高, 但由于通信过程涉及大量安全密钥及其他参数, 可能导致通信代价会高于计算代价.

同态加密方案能够保证对密文执行的特定数学

收稿日期 2020-04-21　录用日期 2020-07-27
Manuscript received April 21, 2020; accepted July 27, 2020
国家科技重大专项 (2018YFB0204304), 天津市研究生科研创新基金资助项目 (2019YJSB067) 资助
Supported by National Science and Technology Major Project of China (2018YFB0204304), and Tianjin Research Innovation Project for Postgraduate Students (2019YJSB067)
本文责任编委 吴建鑫
Recommended by Associate Editor WU Jian-Xin
1. 南开大学软件学院 天津 300071
1. College of Software, Nankai University, Tianjin 300071

运算会对其明文有着相同的影响. 贾春福等[16] 提出一种在同态加密数据集上训练机器学习算法的方案. 这类方法能够很好地解决隐私安全问题, 既可以将加密的数据聚合在一起进行模型训练, 也可以采用联邦学习进行模型训练. 然而, 该类方法[16−19] 需要根据所构建的机器学习模型, 选用或设计恰当的同态加密方案对训练数据进行加密, 对密码学知识有着较高的要求. 同时, 由于对数据加密需要大量的算力资源, 该类方法不适用于大数据环境下的深度学习模型训练. Phong 等[20] 提出通过对联邦学习过程中各训练者产生的梯度数据进行加密, 从而保证多个参与训练者的本地数据隐私安全. 这类方法通过对梯度参数进行加密保护, 能够很好地保护数据隐私安全. 然而, 该方法的加密运算量与训练数据量的大小直接相关, 会大大增加模型训练时间和计算成本, 并且没有对偏置项进行考虑.

差分隐私技术指在模型训练过程中引入随机性, 即添加一定程度的随机噪声, 使输出结果与真实结果存在着一定程度的偏差, 从而防止攻击者推理. Agrawal 等[21] 提出通过对训练数据集进行扰动, 实现联邦深度神经网络的隐私保护. Shokri 等[22] 通过在神经网络模型的梯度参数上添加噪声, 从而实现数据隐私的保护. Truex 等[23] 针对联邦学习模型, 提出一种结合差分隐私和安全多方计算的隐私保护方案, 能够在保护数据隐私的同时, 还有着较高的准确率. 然而在梯度参数上添加噪声, 可能会造成机器学习模型训练时收敛难度增大、预测精度下降, 降低模型的使用性能.

针对上述问题, 本文提出一种支持数据隐私保护的联邦深度神经网络模型. 本文主要贡献有两个: 1) 对多层神经网络的训练过程进行分析, 详细地论述模型权重参数与梯度参数是如何泄露数据集信息的. 2) 基于此, 将固定的偏置项参数改为随机数生成, 从而避免由于梯度参数信息泄露而导致数据信息的直接泄露; 并且将模型梯度参数加密替换为神经网络模型的权重参数加密, 从而减少了加解密运算量; 同时训练者可选择多种优化算法, 不再局限于随机梯度下降法, 使得提出的方法更加适用于真实场景.

1 问题陈述与预备知识

1.1 问题陈述

深度神经网络属于有监督学习算法, 设有数据集 $\{\vec{x}_i,\ y_i\}_{i=1}^N$, 其中数据 $\vec{x}_i=(x_{i1},\cdots,x_{id})$, d 为输入数据的维度, y_i 为第 i 个数据的标签值. 深度神经网络模型经过训练, 使得样本数据 $\vec{x}_i$ 映射到标签 y_i 上.

本文研究的联邦学习系统由两部分组成: 多个训练者和参数服务器. 所有训练者都预先同意训练一个确定框架的深度神经网络模型, 并使用相同的优化算法 (如随机梯度下降法) 和其他训练参数 (如学习率、mini-batch、迭代次数). 各训练者和参数服务器采用联邦学习的方式, 互相联合、协作训练一个深度神经网络, 在训练过程中各训练者的训练数据均在本地. 另外, 训练者和参数服务器之间需要认证与授权, 授权后的训练者才可以获得从参数服务器下载参数和上传参数的权限. 关于云环境下的用户认证和授权非本文研究重点, 具体可见文献 [24].

本研究使用 "诚实且好奇" 的半可信威胁模型[25]. 假设所有训练者和参数服务器都是 "诚实且好奇" 的半可信实体, 所有训练者和参数服务器都会遵守训练协议, 但是在训练模型过程中都想要通过推理得到其他训练者的数据信息. 本文研究旨在在联邦学习训练阶段保护各训练者的数据隐私, 使各训练者不能推理得到对方的数据信息, 同时也不能泄露信息给第三方服务器.

1.2 神经网络模型

一个典型的神经网络模型通常包含三层, 分别为输入层、隐藏层与输出层[26]. 深度神经网络模型可以包含多个隐藏层, 每个隐藏层具有一定数量的神经元, 并且不同隐藏层的神经元可以自动从训练数据中提取不同层次的特征. 两个神经元之间的连接, 即是神经网络模型的权重参数. 每层神经网络除了神经元权重参数以外, 还有偏置参数. 权重参数和偏置参数统称为神经网络的模型参数. 神经网络工作过程主要有两个: 正向传播与反向传播. 正向传播过程是指, 输入数据正向通过各层神经元的计算, 然后输出预测结果. 反向传播过程是指, 先计算预测结果与目标值的偏差值 $Loss$, 然后根据偏差值采用链式求导与学习算法 (例如梯度下降法) 对各层网络的权重进行更新.

1.3 联邦学习

当数据分散在不同组织中时, 联邦学习提供了一种多方协作训练深度学习模型的解决方案, 现在已成为深度学习领域的研究热点[27−29].

图 1 为联邦学习结构图. 首先, 远程参数服务器将初始化的深度学习模型发送至各个训练者. 然后, 各训练者使用自己本地的数据集对模型进行训练, 并将训练的梯度参数 G_i $(i=1,2,\cdots,n)$ 发送至参数服务器, 而不是将本地数据发送到云端训练. 接着, 参数服务器使用各训练者所上传的梯度参数,

将全局模型的权重参数 $W_{\rm old}$ 更新为 $W_{\rm new}$, 并将更新后的权重参数发送至各个训练者. 整个学习过程重复进行, 直至达到设定的学习次数或满足用户设定的停止训练的条件.

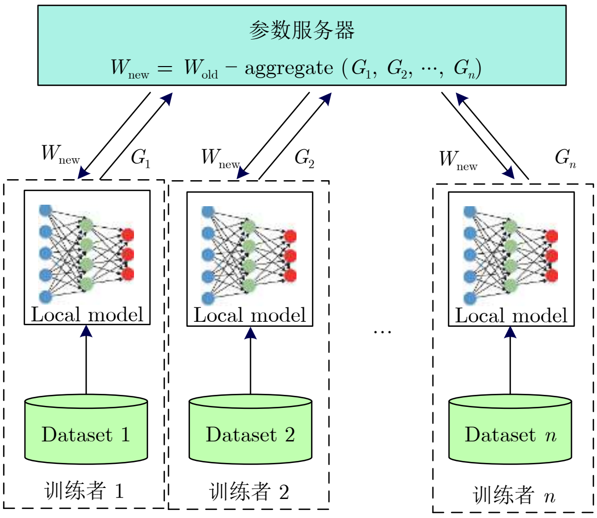


图 1　联邦学习结构

Fig. 1　Federated learning structure

可以看出, 多个训练者可以通过使用联邦学习框架协同训练深度学习模型, 从而提高模型性能. 但是, 该训练过程中主要存在两个缺点: 1) 通信过程不安全; 2) 模型权重参数与梯度参数可能存在隐私安全问题. 经典的信息加密方法能够有效解决第一类问题[30−32], 而第二类问题则需要根据深度学习具体问题建立一种隐私保护的机制. 并且, 由于隐私政策或法规的限制, 大多数客户都不愿共享其数据. 因此, 需要对考虑数据隐私保护的联邦深度学习模型进行深入研究.

1.4　同态加密

根据支持的数学运算种类与运算次数, 同态加密算法可以分为: 支持一种数学运算且不限制运算次数的部分同态加密算法、支持特定数学运算且有限运算次数的有限同态加密算法、支持不限制数学运算种类且不限制运算次数的全同态加密算法. 本研究采用的是 Paillier 同态加密算法, 该算法包括 4 个步骤 (密钥生成 KeyGen、加密算法 Encryption、解密算法 Decryption 和验证算法)[33−35].

1) 密钥生成 KeyGen $(\lambda) \rightarrow (pk,\ sk)$:

随机选取两个长度相同的大素数 p 和 q, 且满足 $gcd(pq,\ (p-1)(q-1)) = 1$, 计算 $N = pq$, $\lambda = lcm\ (p-1,\ \ q-1)$, 随机选取 $g \in \mathbf{Z}^*_{N^2}$, 则公钥 $pk = (N,\ \ g)$, 私钥 $sk = (\lambda)$.

2) 加密算法 Encryption$(pk,\ m) \rightarrow c$:

输入公钥 pk 和明文信息 m, 随机选取 $r \in \mathbf{Z}^*_N$, 计算密文 $c = g^m r^N\ ({\rm mod} N^2)$.

3) 解密算法 Decryption$(sk,\ c) \rightarrow\ m$:

输入私钥 sk 和密文信息 c, 其中 $L\,(x) = (x-1)/N$, 计算明文 $m = \left(L\left(c^\lambda\ ({\rm mod} N^2)\right)/L(g^\lambda\ ({\rm mod} N^2)\ t)\right) {\rm mod} N$.

4) 验证算法:

$$\begin{aligned}&E\,(m_1) \times E\,(m_2) = \\&\quad \left(g^{m_1} {r_1}^N\ ({\rm mod} N^2)\right) \times \left(g^{m_2} {r_2}^N\ ({\rm mod} N^2)\right) = \\&\quad g^{m_1+m_2} (r_1 \times r_2)^N\ ({\rm mod} N^2) = \\&\quad E\,(m_1 + m_2)\end{aligned}$$

2　深度学习模型的隐私泄露

经典的神经网络在训练过程中, 可能存在数据集隐私泄露的风险. 图 2 为一个典型的三层神经网络, 其中 x_1 与 x_2 为输入数据, w_i $(i = 1, \cdots, 8)$ 分别为各神经元的权重, b_i $(i = 1, \cdots, 4)$ 为各偏置项的权重, $target$ 为输出目标值. 下面以三种情况为例, 详细地对深度学习模型的隐私泄露问题进行说明.

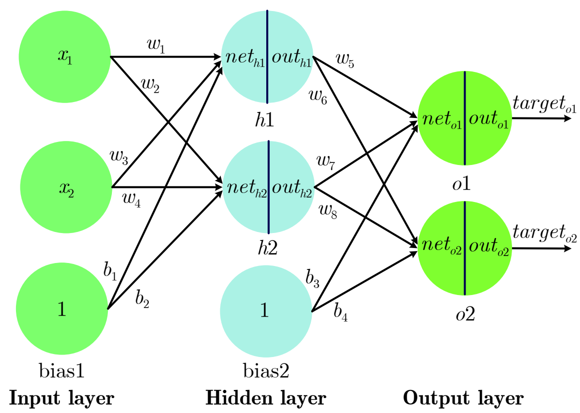


图 2　神经网络结构

Fig. 2　Neural network construction

1) 第一种情况

首先计算神经网络的输出值与目标值的偏差, 计算公式如下:

$$\begin{aligned}L_{\rm total} &= L_{o1} + L_{o2} = \\&\frac{1}{2}\left[(out_{o1} - target_{o1})^2 + (out_{o2} - target_{o2})^2\right]\end{aligned} \tag{1}$$

然后, 根据链式求导法则, 可以计算出 w_1、b_1 和 w_3 的梯度:

$$\begin{aligned}&\frac{\partial L_{\rm total}}{\partial w_1} = \frac{\partial L_{\rm total}}{\partial out_{h1}} \frac{\partial out_{h1}}{\partial net_{h1}} \frac{\partial net_{h1}}{\partial w_1} = \\&\left(\frac{\partial L_{o1}}{\partial out_{h1}} + \frac{\partial L_{o2}}{\partial out_{h1}}\right) out_{h1}(1 - out_{h1}) x_1\end{aligned} \tag{2}$$

$$\frac{\partial L_{\text{total}}}{\partial b_1}=\frac{\partial L_{\text{total}}}{\partial out_{h1}}\frac{\partial out_{h1}}{\partial net_{h1}}\frac{\partial net_{h1}}{\partial b_1}=$$
$$\left(\frac{\partial L_{o1}}{\partial out_{h1}}+\frac{\partial L_{o2}}{\partial out_{h1}}\right)out_{h1}(1-out_{h1})\times 1 \tag{3}$$

$$\frac{\partial L_{\text{total}}}{\partial w_3}=\frac{\partial L_{\text{total}}}{\partial out_{h1}}\frac{\partial out_{h1}}{\partial net_{h1}}\frac{\partial net_{h1}}{\partial w_3}=$$
$$\left(\frac{\partial L_{o1}}{\partial out_{h1}}+\frac{\partial L_{o2}}{\partial out_{h1}}\right)out_{h1}(1-out_{h1})x_2 \tag{4}$$

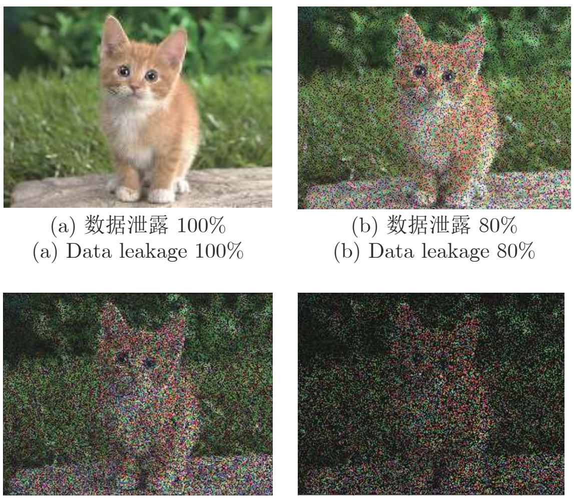
根据链式求导结果, 可以得出:

$$\frac{\partial L_{\text{total}}}{\partial w_1}\left(\frac{\partial L_{\text{total}}}{\partial b_1}\right)^{-1}=x_1 \tag{5}$$

$$\frac{\partial L_{\text{total}}}{\partial w_3}\left(\frac{\partial L_{\text{total}}}{\partial b_1}\right)^{-1}=x_2 \tag{6}$$

从式 (5)、式 (6) 看出, 能够从梯度参数反推出输入数据. 并且第一层网络的所有梯度参数, 都能够通过与之相对应的偏置项推导出数据信息, 例如 x_1 也可以通过 w_2 的梯度除以 b_2 的梯度得到.

图 3(a) ~ 图 3(d) 分别为信息泄露 100%、80%、50% 和 30% 的图片. 从图中可以看出, 当获取一定数量的输入数据时, 其关键信息就可能已经泄露.



(a) 数据泄露 100%
(a) Data leakage 100%

(b) 数据泄露 80%
(b) Data leakage 80%

(c) 数据泄露 50%
(c) Data leakage 50%

(d) 数据泄露 30%
(d) Data leakage 30%

图 3 不同比例的数据信息泄露

Fig.3 Different proportions of data information leakage

当求得训练数据中的输入数据后, 通过神经网络模型正向传播求出模型输出值. 然后, 根据式 (7)、式 (8) 可以求出数据集中的目标输出数据 $target_{o1}$, 类似可以依次求出所有的目标输出数据. 经过上述过程, 数据集的信息便泄露出来.

$$\frac{\partial L_{\text{total}}}{\partial w_5}=\frac{\partial L_{\text{total}}}{\partial out_{o1}}\frac{\partial out_{o1}}{\partial net_{o1}}\frac{\partial net_{o1}}{\partial w_5}=$$
$$-(target_{o1}-out_{o1})out_{o1}(1-out_{o1})out_{h1} \tag{7}$$

$$target_{o1}=$$
$$out_{o1}-\frac{\partial L_{\text{total}}}{\partial w_5(out_{o1}(1-out_{o1})out_{h1})} \tag{8}$$

2) 第二种情况

假如第一层的梯度参数 ($\partial L_{\text{total}}/\partial w_i$, $i=1,\cdots,4$) 被加密, 但是依然可以根据式 (9) ~ 式 (11) 推导出隐藏层的输出值 out_{h1} 与 out_{h2}. 然后通过解方程组式 (12), 则可求出输入数据 x_1 与 x_2. 再按照第一种情况正向传播求解目标数据值, 便可以得到数据集的所有信息.

$$\frac{\partial L_{\text{total}}}{\partial w_5}=\frac{\partial L_{\text{total}}}{\partial out_{o1}}\frac{\partial out_{o1}}{\partial net_{o1}}\frac{\partial net_{o1}}{\partial w_5}=$$
$$-(target_{o1}-out_{o1})out_{o1}(1-out_{o1})out_{h1} \tag{9}$$

$$\frac{\partial L_{\text{total}}}{\partial w_7}=\frac{\partial L_{\text{total}}}{\partial out_{o1}}\frac{\partial out_{o1}}{\partial net_{o1}}\frac{\partial net_{o1}}{\partial w_7}=$$
$$-(target_{o1}-out_{o1})out_{o1}(1-out_{o1})out_{h2} \tag{10}$$

$$\frac{\partial L_{\text{total}}}{\partial b_3}=\frac{\partial L_{\text{total}}}{\partial out_{o1}}\frac{\partial out_{o1}}{\partial net_{o1}}\frac{\partial net_{o1}}{\partial b_3}=$$
$$-(target_{o1}-out_{o1})out_{o1}(1-out_{o1})\times 1 \tag{11}$$

$$\begin{cases} x_1\times w_1+x_2\times w_3=net_{h1}-b_1\\ x_1\times w_2+x_2\times w_4=net_{h2}-b_2\end{cases} \tag{12}$$

3) 第三种情况

假如神经网络模型偏置项数值不是固定值 1, 而是由随机数生成, 可以通过式 (13) 求出各隐藏层输出值与偏置项的比值. 然后, 尝试可能使用的偏置项数值, 将推测的偏置项数值根据第一种情况, 计算出推测的输出值和推测的目标值, 最后将使用推测结果计算的梯度值与真实的梯度值进行比较. 重复上述过程, 暴力求解可能使用的偏置项数值. 从图 4 中可以看出, 当输入数据为图片时, 与真实偏置项数值的差距即为推测图片与真实图片亮度的差距. 因此, 当推测的偏置项数值与真实的偏置项数值比较接近时, 信息便可能出现泄露.

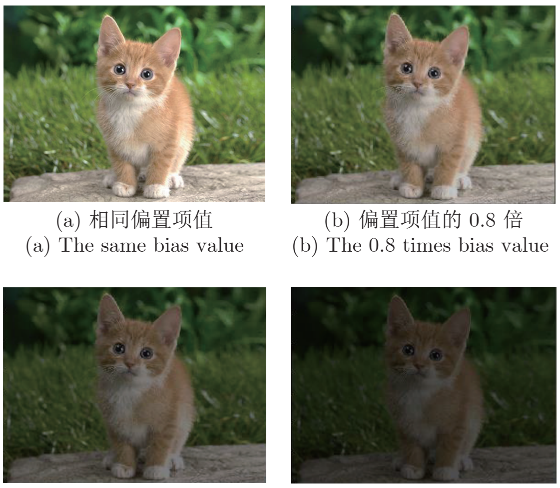
$$\frac{\partial L_{\text{total}}}{\partial w_5}\left(\frac{\partial L_{\text{total}}}{\partial b_3}\right)^{-1}=\frac{out_{h1}}{b_3} \tag{13}$$

3 支持数据隐私保护的联邦深度学习模型

支持数据隐私保护的联邦深度学习模型, 其中包括一个参数服务器与多个训练者. 模型运行过程主要分为三个部分: 模型初始化、模型训练和模型部署. 图 5 为联邦深度学习算法训练过程交互图.

采用文献 [20] 中提出的联邦学习隐私保护的通讯方案, 各训练者与参数服务器之间建立不同的 TLS/SSL (Transport layer security/Secure sockets layer) 安全通道. 各训练者之间采用经典的 AES (Advanced encryption standard) 加密算法, 对同态加密算法的公钥 pk 与私钥 sk、深度神经网络模型的结构参数、选用的优化算法及其学习率等进行加密传输. 收到信息后, 各训练者进行模型初始化、模型训练以及模型部署操作.

模型初始化. 训练者们预先确定负责模型初始化的训练者. 该训练者对深度神经网络模型进行搭建, 模型中的权重参数与偏置项参数使用随机数算法生成. 然后, 确定同态加密算法的参数, 根据同态加密算法的定义生成公钥 pk 与私钥 sk (具体过程见文中第 1.4 节或参考文献 [33−35]), 并将公钥、私钥与偏置项数值使用 AES 算法加密后发送至各训练者, 对参数服务器保密. 该训练者使用同态加密算法的私钥对深度神经网络模型的权重参数进行同态加密, 并将加密后的权重参数发送至参数服务器. 参数服务器将该训练者上传的加密参数广播至其他训练者. 其他训练者使用私钥对加密的模型权重参数进行解密, 将解密后的参数加载到自己的本地模型, 从而完成初始化过程.



(a) 相同偏置项值
(a) The same bias value

(b) 偏置项值的 0.8 倍
(b) The 0.8 times bias value

(c) 偏置项值的 0.5 倍
(c) The 0.5 times bias value

(d) 偏置项值的 0.3 倍
(d) The 0.3 times bias value

图 4　不同偏置值的数据信息泄露

Fig. 4　Data information leakage of different bias values

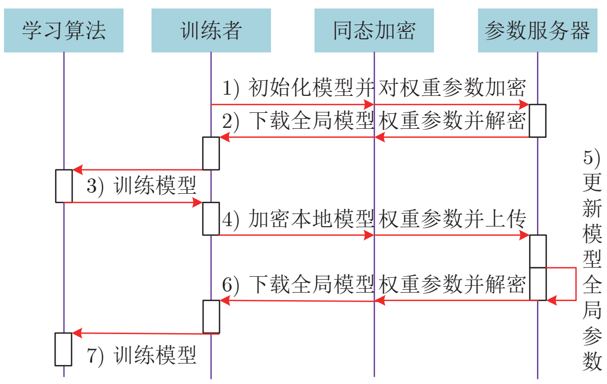


图 5　训练过程交互图

Fig. 5　Interaction in the training process

模型训练. 如图 6 所示, 各训练者将下载的全局模型权重参数解密后, 加载到本地模型上. 使用本地数据对模型进行训练, 完成一次学习后, 将更新的本地模型的权重参数经同态加密后上传至参数服务器. 进行加密的全局模型权重参数更新, 并将加密的全局模型权重参数广播给所有的训练者. 迭代上述学习过程, 直至达到训练设定条件. 具体训练过程见第 3.1 节和第 3.2 节.

模型部署. 当模型完成训练后, 各训练者不再上传深度神经网络的权重参数. 将训练好的深度神经网络全局模型的权重参数经私钥 sk 解密后, 加载至本地模型进行部署, 并断开与参数服务器的连接.

3.1　训练者

各训练者对存储在本地的数据集进行预处理, 然后对本地的深度神经网络模型进行训练. 具体步骤如下:

1) 各训练者首先根据具体项目内容, 预先确定好所采用的数据预处理方式, 然后对各自本地数据进行预处理;

2) 各训练者从参数服务器下载深度神经网络全局模型权重参数密文, 然后使用私钥 sk 对密文进行解密, 最后将全局模型权重参数加载至本地深度神经网络模型;

3) 各训练者根据预先设定好的 mini-batch 值、迭代次数、优化算法和学习率, 使用本地数据进行一次迭代学习, 并使用公钥 pk 对本地深度神经网络更新的模型权重参数进行同态加密, 最后将密文发送至参数服务器;

4) 重复步骤 2) 和步骤 3), 直至达到设定迭代次数.

3.2　参数服务器

如图 6 所示, 参数服务器将各训练者上传的模型参数求平均值, 并更新加密的全局模型权重参数. 图中密文计算公式的合法性是由同态加密算法的性质来保证的, 具体特性可见附录 A.

定理 1 (针对参数服务器的安全性). 如果采用的加密方案是满足 CPA (Chosen-plaintext attack) 安全的, 则训练者不会将数据集上的信息泄漏给“诚实但好奇”的参数服务器.

证明. 训练者仅将加密的模型权重参数发送到云服务器. 因此, 在联邦训练过程中, 如果加密方案是 CPA 安全的, 则不会泄漏有关训练者数据的任

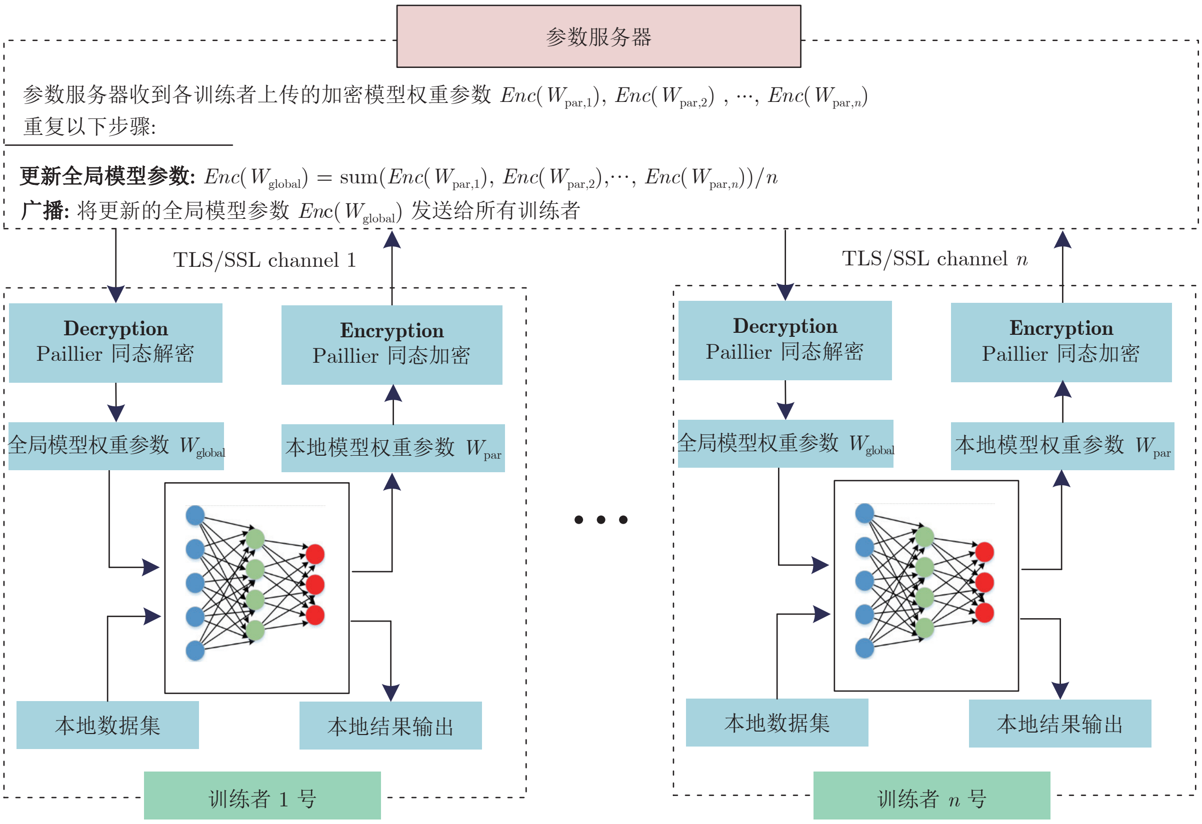


图 6 支持数据隐私保护的联邦学习训练过程

Fig.6 The training process of the date privacy-preserving federated learning

何信息. □

注 1. 当各参与训练者通过不同的 TLS/SSL 通道连接到参数服务器时, 假设参数服务器不会和任何训练者串通, 训练者仅能够知道全局模型的参数, 而不能知道其他参与训练者的梯度信息.

注 2. 深度神经网络中的偏置项不是固定的常数项 (例如数值为 1), 而是由某一训练者随机生成的常数项.

4 联邦深度学习算法分析

4.1 安全性分析

回顾上述联邦深度学习算法训练过程, 训练者与参数服务器获得的中间数据如表 1 所示. 从表 1 中可以看到, 在联邦深度学习训练过程中, 各训练者通过私钥解密获得的是全局模型权重参数, 无法获得其他训练者模型权重参数 W_{par}、梯度 G、预测结果 Prediction results 以及损失值 $Loss$. 同时, 参数服务器获得的是各训练者加密过的本地模型权重参数 $Enc(W_{\text{par}})$ 和加密过的全局模型权重参数 $Enc(W_{\text{global}})$, 而参数服务器没有私钥, 无法对参数数据进行解密.

表 1 训练者与参数服务器获得的数据信息

Table 1 Data information obtained by the participant and parameter server

名称	训练者 i	参数服务器
中间数据	$Enc(W_{\text{global}})$	$Enc(W_{\text{global}})$
	Prediction results	$Enc(W_{\text{par},1})$
	$Loss_i$	$Enc(W_{\text{par},2})$
	G_i	⋯
	$W_{\text{par},i}$	$Enc(W_{\text{par},n})$

联邦深度学习训练前, 模型的偏置项参数与同态加密算法的私钥、公钥使用现代成熟的加密方法在各训练者之间进行信息传输; 联邦深度学习训练过程中, 参数服务器与各训练者传输的是模型参数的密文, 即使模型参数信息被敌手截获, 其也无法进行解密.

4.2 时间分析

与传统联邦深度学习算法相比, 本文所提出算法额外产生的计算开销和通信开销分别为: 模型参数加密/解密, 模型参数上传/下载. 本文使用 Python 开源平台上的 Paillier 同态加密算法库, 密钥长度为 1024 bit, 计算机 CPU 型号为 Intel i7-6500. 表 2 所示为不同个数的模型参数运算 100 次, 同态

表 2 算法执行时间

Table 2 Execution time of the algorithms

操作		参数量							
		10	16	32	64	160	512	2048	50176
Paillier 方案	加密	0.07 s	0.11 s	0.23 s	0.48 s	1.19 s	3.91 s	15.05 s	381.51 s
	解密	0.02 s	0.03 s	0.07 s	0.13 s	0.34 s	1.10 s	4.31 s	107.77 s
	加法	0.10 ms	0.19 ms	0.49 ms	1.09 ms	5.59 ms	8.07 ms	30.82 ms	0.81 s
直接相加		0.96 μs	0.99 μs	1.01 μs	1.03 μs	1.10 μs	1.49 μs	2.83 μs	33.30 μs

加密算法操作 (包括加密操作、密文加法操作和解密操作) 和无加密算法的直接相加的平均每次执行时间.

训练过程中, 迭代次数为 $epoch$, 训练者个数为 n, 网络模型参数个数为 $M_{\rm par}$, 则加密时间复杂度为 $\mathrm{O}(epoch \times n \times M_{\rm par} \times E_{\rm paillier})$, 解密时间复杂度为 $\mathrm{O}(epoch \times n \times M_{\rm par} \times D_{\rm paillier})$. 可以看出, 所提出的联邦深度学习算法加密/解密次数与数据集无关. 本文模型参数量为 53018 (模型结构如表 3 所示, Li 代表第 i 层神经网络), $n = 2$, 一个迭代训练过程加密/解密操作数为 $2 \times 53018 = 1.06 \times 10^5$ 次. 如果采用文献 [20] 的方法, 假设训练数据集个数为 60000 个, 则其一个迭代训练过程加密/解密操作数为 $2 \times 60000 \times 53018/50 = 1.27 \times 10^8$ 次. 通过对比能够看出, 直接对参数进行同态加密, 能够大大缩短模型训练时间.

表 3 深度神经网络模型结构

Table 3 Deep neural network model structure

model	深度神经网络
结构	L1 input = 784, output = 64
	L2 input = 64, output = 32
	L3 input = 32, output = 16
	L4 input = 16, output = 10

本文所提出算法的通信开销为 $T_{\rm cost} = 2(epoch \times n \times M_{\rm par} \times ct)$, 其中 ct 为一条密文大小, 空间复杂度为 $O(n \times M_{\rm par} \times ct)$. 可以看出, 所提出算法的空间复杂度与数据集无关, 主要受到模型参数量的影响. 本文模型参数量为 53018, $n = 2$, $ct = 256$ bit, 一个迭代训练过程通信开销为 $2 \times 53018 \times 256 = 25.88$ MB. 如果采用文献 [20] 的方法, 假设训练数据集个数为 60000 个, 则其一个迭代训练过程通信开销为 $2 \times 60000 \times 53018 \times 256/50 = 0.12$ GB. 通过对比能够看出, 直接对参数进行同态加密, 能够大大减小通信开销.

5 实验与分析

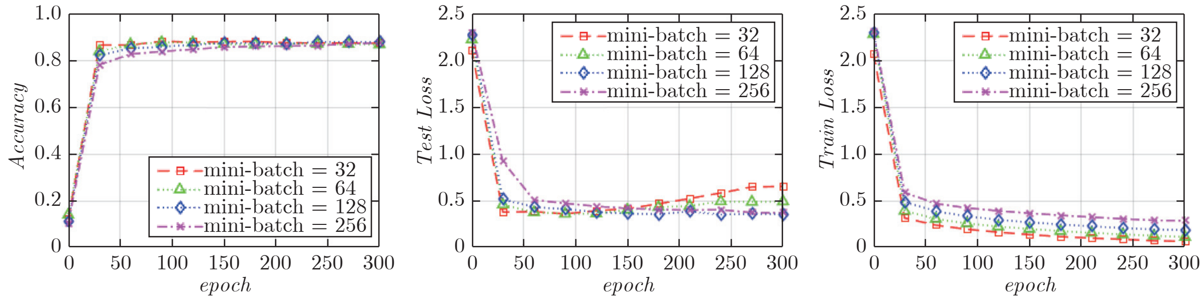
5.1 实验环境与数据集

电脑运行环境如下: CPU 为 Intel i7-6550, 内存 16 GB, 显卡 NVIDIA 1050Ti, Pytorch 1.1, CUDA 9.0, Win10 操作系统. 使用开源的 Paillier 库[36], 对神经网络模型的权重参数进行加密与解密. 实验数据集为 Zalando Research 在 GitHub 上推出的一个用于替代手写数字识别数据集 MNIST 的全新数据集 Fashion-MNIST, 该数据集已经被广泛用于机器学习算法的性能评估[37−39]. Fashion-MNIST 的训练集包含 60000 个样本, 测试集包含 10000 个样本, 共有 10 类 (Trouser, Pullover, Dress, Coat 等), 每个样本都是 28×28 的灰度图像. 本文以两个训练者为例, 实现支持隐私保护的联邦深度神经网络模型. 训练集中, 前 30000 个样本设定为数据集 1, 后 30000 个样本设定为数据集 2, 从而完成训练数据集的切割. 传统训练模型使用完整的训练数据集, 而联邦深度学习中的训练者 1 使用数据集 1 作为训练数据集, 训练者 2 使用数据集 2 作为训练数据集. 训练好模型后, 采用 Fashion-MNIST 测试数据集进行评估.

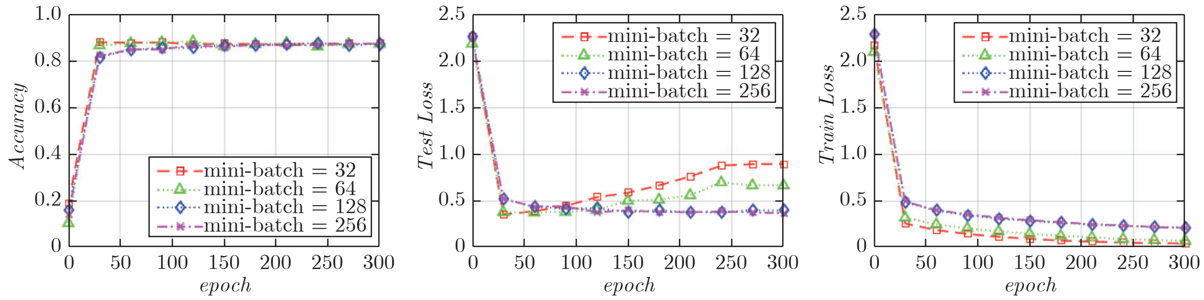
5.2 实验和结果分析

本节将传统深度神经网络模型与支持隐私保护的联邦深度神经网络模型采用如表 3 所示相同的结构, 采用 SGD (Stochastic gradient descent) 算法对深度神经网络 DNN (Deep neural network) 与联邦深度神经网络 PFDNN (Privacy-preserving federated deep neural network) 进行训练, 采用 ReLU 激活函数, 循环次数 $epoch$ 为 300 次. DNN-1 与 PFDNN-1 学习率设定为 0.005, DNN-2 与 PF-DNN-2 学习率设定为 0.01.

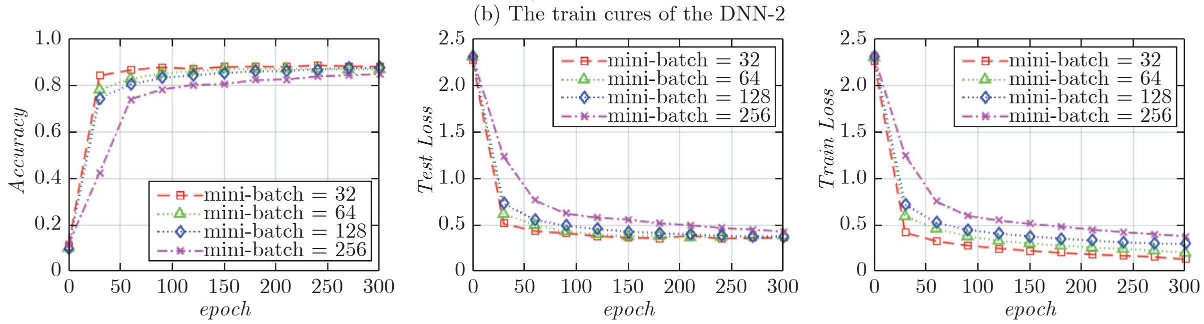
不同 mini-batch 值下, 各模型训练过程曲线如图 7 所示, 其中包含测试数据集识别准确率 $Accuracy$ 曲线、测试数据集 $Loss$ 曲线和训练数据集 $Loss$ 曲线. 从图 7(a) 和图 7(b) 中可以看出, DNN 模型在迭代前 50 次时, 训练集 $Loss$ 值与测试集 $Loss$ 值迅速下降, 识别准确率 $Accuracy$ 快速提高. DNN-1 模型迭代至 150 次左右时, $Loss$ 值与识别准确率趋近于稳定. 由于学习率高于 DNN-1, DNN-2 模型迭代至 100 次左右时 $Loss$ 值与识别准确率 $Accuracy$ 趋近于稳定. 从图 7(c) 和图 7(d) 的 $Loss$



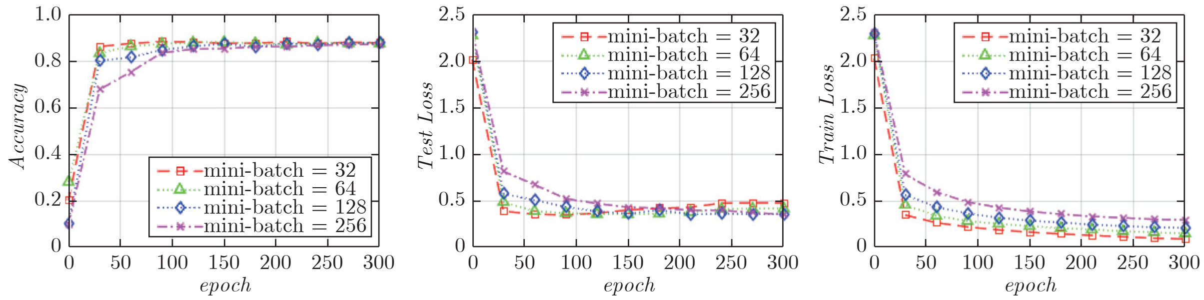
(a) DNN-1 模型训练曲线
(a) The train cures of the DNN-1



(b) DNN-2 模型训练曲线
(b) The train cures of the DNN-2



(c) PFDNN-1 模型训练曲线
(c) The train cures of the PFDNN-1



(d) PFDNN-2 模型训练曲线
(d) The train cures of the PFDNN-2

图 7 各模型训练过程曲线

Fig. 7 Training process curves of each models

值曲线可以看出, PFDNN-1 与 PFDNN-2 模型在训练时模型收敛速度相比于 DNN 模型较慢一些, 迭代至 200 次后 *Loss* 值与识别准确率趋近于稳定. 并且从各模型的测试集 *Loss* 曲线可以观察到, 使用较小 mini-batch 值会加快模型收敛速度. 在 mini-batch = 32 时, 各模型的测试集 *Loss* 曲线都出现了不同程度的过拟合现象. 可以看出, 所提出的 PFDNN 模型在不同参数下, 识别准确率 *Accuracy*、训练集 *Loss* 值与测试集 *Loss* 值都与 DNN 模型总体保持一致.

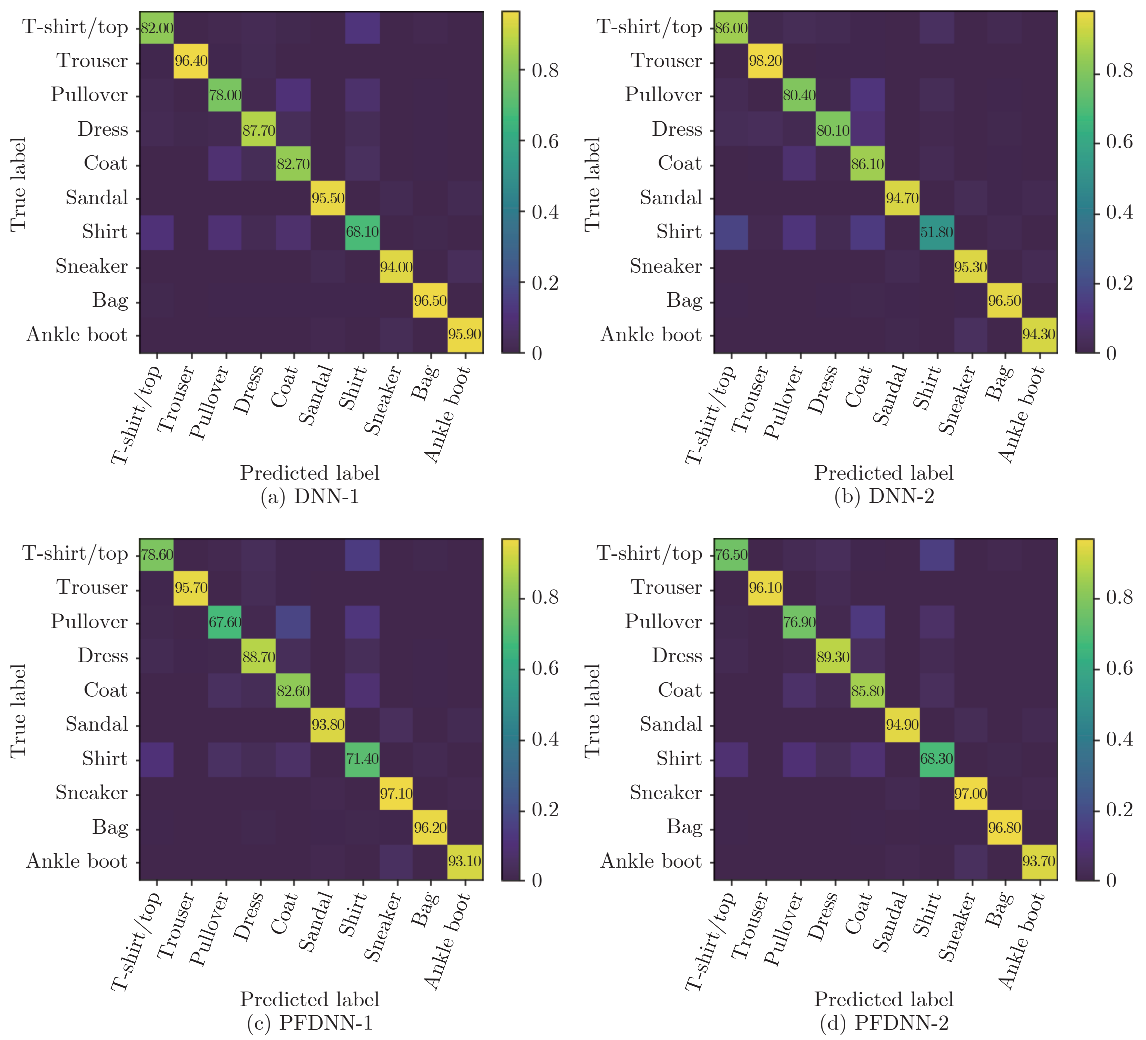
为观察 DNN 与 PFDNN 具体物品分类的情况, 4 种模型在测试集识别结果的混淆矩阵如图 8 所示. 可以看出, DNN 与 PFDNN 对测试集中的 Shirt 识别效果都不理想, 并且都主要误识别为 T-shirt、Pullover 和 Coat. 以混淆矩阵的结果和训练过程测试集识别准确率对 DNN 与 PFDNN 的偏差进行评估. 评估计算公式为:

$$dev_{\text{avg}} = \frac{1}{cn}\sum_{i=1}^{cn}|\text{DNN}_{\text{acc},\ i} - \text{PFDNN}_{\text{acc},\ i}| \tag{14}$$

$$dev_{\max} = |\text{DNN}_{\text{acc},\ \max} - \text{PFDNN}_{\text{acc},\ \max}| \tag{15}$$

式中, cn 表示物品种类个数, $\text{DNN}_{\text{acc},i}$ 表示 DNN 识别第 i 类物品的准确率, $\text{PFDNN}_{\text{acc},i}$ 表示 PFDNN 识别第 i 类物品的准确率, $\text{DNN}_{\text{acc,max}}$ 表示 DNN 在训练过程中测试集最高的准确率, $\text{PFDNN}_{\text{acc,max}}$ 表示 PFDNN 在训练过程中测试集最高的准确率. 如表 4 所示, 随着 mini-batch 值的增大, DNN 与 PFDNN 在具体物品分类上的偏差加大. 在 mini-batch 值较大时, PFDNN-1 由于欠拟合, 与 DNN 最高准确率的偏差最大达到了 2.82%; 而当 PFDNN 充分训练后, 与 DNN 最高准确率的偏差较小. 总体可以看出, 所提出的 PFDNN 模型能够保证很高的精确度.

表 5 为 4 种模型在测试集上, 对不同种类物品的识别精确率 *Precision* 与召回率 *Recall*. 识别精确率是指正确预测为正占全部预测为正的比例, 而召回率是指正确预测为正占全部为正的比例. 识别精确率与召回率越接近 1, 说明分类器的分类效果越好. 从表 5 中可以看出, 传统深度学习模型与所提出的联邦深度学习模型对各类物品识别的结果相差不大. 这说明本文所提出的 PFDNN 模型能在没有明显精度损失的前提下, 保护数据的隐私安全.



(a) DNN-1　(b) DNN-2　(c) PFDNN-1　(d) PFDNN-2

图 8　测试集预测结果的混淆矩阵

Fig. 8　The confounding matrix of the test dataset prediction results

表 4 不同模型偏差结果
Table 4 The deviation results of the different models

模型		参数			
		mini-batch = 32	mini-batch = 64	mini-batch = 128	mini-batch = 256
DNN-1	dev_{avg}	2.04%	2.62%	2.68%	6.25%
PFDNN-1	$dev_{\max}$	0.47%	0.67%	1.07%	2.82%
DNN-2	dev_{avg}	2.30%	3.00%	4.39%	5.26%
PFDNN-2	$dev_{\max}$	0.36%	0.57%	0.03%	0.05%

表 5 不同类别物品的预测结果
Table 5 Prediction results of the different items

Method	DNN-1 mini-batch = 128 $lr = 0.005$, $epoch = 300$		DNN-1 mini-batch =128 $lr = 0.01$, $epoch = 300$		FDNN-1 mini-batch = 128 $lr = 0.005$, $epoch = 300$		FDNN-2 mini-batch = 128 $lr = 0.01$, $epoch = 300$	
Type	*Recall*	*Precision*	*Recall*	*Precision*	*Recall*	*Precision*	*Recall*	*Precision*
T-shirt	82.00%	83.42%	86.00%	78.18%	78.60%	84.24%	76.50%	86.50%
Trouser	96.40%	98.07%	98.20%	92.99%	95.70%	98.76%	96.10%	99.17%
Pullover	78.00%	78.23%	80.40%	76.14%	67.60%	83.25%	76.90%	78.47%
Dress	87.70%	87.96%	80.10%	91.23%	88.70%	85.62%	89.30%	86.03%
Coat	82.70%	77.80%	86.10%	70.40%	82.60%	74.08%	85.80%	75.59%
Sandal	95.50%	95.50%	94.70%	95.75%	93.80%	95.81%	94.90%	96.25%
Shirt	68.10%	71.16%	51.80%	80.06%	71.40%	64.09%	68.30%	69.48%
Sneaker	94.00%	95.05%	95.30%	92.08%	97.10%	90.58%	97.00%	92.21%
Bag	96.50%	95.64%	96.50%	93.78%	96.20%	95.82%	96.80%	96.80%
Boot	95.90%	93.84%	94.30%	95.54%	93.10%	96.38%	93.70%	96.80%
Average	87.68%	87.67%	86.34%	86.62%	86.48%	86.86%	87.53%	87.69%

6 结论

针对联邦学习中存在的数据隐私信息泄露问题, 提出了一种支持隐私保护的联邦深度神经网络模型. 本文以多层神经网络为研究目标, 详细地分析模型梯度参数可能会造成数据隐私泄露的三种情况. 基于此, 初始化模型参数时, 偏置项参数不再使用固定数值而使用随机数, 以避免梯度参数泄露导致数据信息直接泄露. 然后, 引入同态加密算法, 各训练者通过对深度神经网络模型的权重参数进行同态加密, 从而保障了自身本地数据隐私安全. 与传统联邦学习隐私保护方法相比, 使用模型权重参数共享, 能够将模型参数更新过程转移到训练者端 (即边缘端), 降低了参数服务器 (即云端) 性能要求, 并且训练者不再局限于随机梯度下降法. 理论分析与实验表明, 所提出的联邦深度神经网络在没有明显损失模型精度的前提下, 能够对用户数据的隐私进行有效的保护.

所提出的联邦深度学习模型收敛速度较慢, 并且在非独立同分布数据集和较大 mini-batch 值上会出现一定程度的精度损失. 虽然提高全局模型更新频率能够进一步提高精度, 但是会消耗大量的计算资源与训练时间. 因此, 下一步拟通过使用 GPU 对同态加密过程进行加速, 以进一步缩短模型训练时间. 并且, 拟对参数服务器中的模型权重参数更新机制 (例如, 全局模型参数更新频率和 mini-batch 值) 进行深入研究, 从而增加支持训练者的个数, 并提高模型收敛速度和计算精度.

附录 A Paillier 加密算法同态特性

对明文 m_1、m_2 采用 Paillier 进行同态加密, 加密后密文为 $E(m_1)= g^{m_1} {r_1}^N \; (\mathrm{mod} N^2)$, $E(m_2)= g^{m_2} {r_2}^N \; (\mathrm{mod} N^2)$, 其中, r_1、r_2 是两个随机数.

对密文 $E(m_1)$、$E(m_2)$ 进行乘法运算得到:

$$\begin{aligned} E(m_1)\cdot E(m_2) = {} & g^{m_1} r_1^N \cdot g^{m_2} r_2^N (\mathrm{mod} N^2) = \\ & g^{m_1+m_2} (r_1 \times r_2)^N (\mathrm{mod} N^2) = \\ & E(m_1+m_2) \end{aligned} \quad \text{(A1)}$$

可以看出, $E(m_1)$ 和 $E(m_2)$ 的乘法结果相当于明文消息 $m_1 + m_2$ 在公钥 pk 和随机数 r_1、r_2 下进行加密, 因此, Pallier 加密算法满足加法同态特性.

当对密文进行指数运算时, 根据加密算法有:

$$E(m_1)^{m_2} = [g^{m_1} r_1^N (\text{mod} N^2)]^{m_2} = g^{m_1 m_2} r_1^{N \cdot m_2} (\text{mod} N^2) = E(m_1 \cdot m_2) \tag{A2}$$

从结果可以看出, 对密文的指数运算等于明文之间的乘法运算.

References

1 Kebria P M, Khosravi A, Salaken S M, Nahavandi S. Deep imitation learning for autonomous vehicles based on convolutional neural networks. *IEEE/CAA Journal of Automatica Sinica*, 2020, **7**(1): 82−95

2 Gong W F, Chen H, Zhang Z H, Zhang M L, Wang R H, Guan C, et al. A novel deep learning method for intelligent fault diagnosis of rotating machinery based on improved CNN-SVM and multichannel data fusion. *Sensors* (*Basel, Switzerland*), 2019, **19**(7): Article No. 1693

3 Roy P, Mahapatra G S, Dey K N. Forecasting of software reliability using neighborhood fuzzy particle swarm optimization based novel neural network. *IEEE/CAA Journal of Automatica Sinica*, 2019, **6**(6): 1365−1383

4 Tian Juan-Xiu, Liu Guo-Cai, Gu Shan-Shan, Ju Zhong-Jian, Liu Jin-Guang, Gu Dong-Dong. Research and challenges of deep learning methods for medical image analysis. *Acta Automatica Sinica*, 2018, **44**(3): 401−424
(田娟秀, 刘国才, 谷珊珊, 鞠忠建, 刘劲光, 顾冬冬. 医学图像分析深度学习方法研究与挑战. 自动化学报, 2018, **44**(3): 401−424)

5 Chen Jia, Zhang Yu-Qi, Song Peng, Wei Yan-Tao, Wang Yu. Application of deep learning to 3D object reconstruction from a single image. *Acta Automatica Sinica*, 2019, **45**(4): 657−668
(陈加, 张玉麒, 宋鹏, 魏艳涛, 王煜. 深度学习在基于单幅图像的物体三维重建中的应用. 自动化学报, 2019, **45**(4): 657−668)

6 Zhang Z H, Guan C, Liu Z. Real-time optimization energy management strategy for fuel cell hybrid ships considering power sources degradation. *IEEE Access*, 2020, **8**: 87046−87059

7 Chen H, Zhang Z H, Guan C, Gao H B. Optimization of sizing and frequency control in battery/supercapacitor hybrid energy storage system for fuel cell ship. *Energy*, 2020, **197**: Article No. 117285

8 Liu W B, Wang Z D, Liu X H, Zeng N Y, Liu Y R, Alsaadi F E. A survey of deep neural network architectures and their applications. *Neurocomputing*, 2017, **234**(19): 11−26

9 Shorten C, Khoshgoftaar T M. A survey on image data augmentation for deep learning. *Journal of Big Data*, 2019, **6**(1): 1−48

10 Roh Y, Heo G, Whang S E. A survey on data collection for machine learning: A big data —— AI integration perspective. *IEEE Transactions on Knowledge and Data Engineering*, DOI: 10.1109/TKDE.2019.2946162

11 Zhang Q, Yang L T, Chen Z, Li P. A survey on deep learning for big data. *Information Fusion*, 2018, **42**: 146−157

12 Zhang Chao, Li Qiang, Chen Zi-Hao, Li Zu-Rui, Zhang Zhen. Medical Chain: Consortium medical blockchain system. *Acta Automatica Sinica*, 2019, **45**(8): 1495−1510
(张超, 李强, 陈子豪, 黎祖睿, 张震. Medical Chain: 联盟式医疗区块链系统. 自动化学报, 2019, **45**(8): 1495−1510)

13 Huang L, Shea A L, Qian H, Masurkar A, Deng H, Liu D B. Patient clustering improves efficiency of federated machine learning to predict mortality and hospital stay time using distributed electronic medical records. *Journal of Biomedical Informatics*, 2019, **99**: Article No. 103291

14 Xu Jian, Wang An-Di, Bi Meng, Zhou Fu-Cai. K-nearest neighbor classifier supporting privacy protection. *Journal of Software*, 2019, **30**(11): 3503−3517
(徐剑, 王安迪, 毕猛, 周福才. 支持隐私保护的 k 近邻分类器. 软件学报, 2019, **30**(11): 3503−3517)

15 Bonawitz K, Ivanov V, Kreuter B, Marcedone A, McMahan H B, Patel S, et al. Practical secure aggregation for privacy-preserving machine learning. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. New York, USA: Association for Computing Machinery, 2017. 1175−1191

16 Jia Chun-Fu, Wang Ya-Fei, Chen Yang, Sun Meng-Jie, Ge Feng-Yi. Application of machine learning algorithms on homomorphic encrypted data sets. *Journal of Tsinghua University (Natural Science Edition)*, 2020, **60**(6): 456−463
(贾春福, 王雅飞, 陈阳, 孙梦洁, 葛凤仪. 机器学习算法在同态加密数据集上的应用. 清华大学学报 (自然科学版), 2020, **60**(6): 456−463)

17 Bost R, Popa R A, Tu S, Goldwasser S. Machine learning classification over encrypted data. In: Proceedings of the 2014 Network and Distributed System Security Symposium. San Diego, USA: 2014.

18 Barni M, Failla P, Lazzeretti R, Sadeghi A R, Schneider T. Privacy-preserving ECG classification with branching programs and neural networks. *IEEE Transactions on Information Forensics and Security*, 2011, **6**(2): 452−468

19 Yu Y, Li H, Chen R, Zhao Y, Yang H, Du X. Enabling secure intelligent network with cloud-assisted privacy-preserving machine learning. *IEEE Network*, 2019, **33**(3): 82−87

20 Phong L T, Aono Y, Hayashi T, Wang L H. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Transactions on Information Forensics and Security*, 2017, **13**(5): 1333−1345

21 Agrawal R, Srikant R. Privacy-preserving data mining. In: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data. New York, USA: Association for Computing Machinery, 2000. 439−450

22 Shokri R, Shmatikov V. Privacy-preserving deep learning. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. New York, USA: Association for Computing Machinery, 2015. 1310−1321

23 Truex S, Baracaldo N, Anwar A, Steinke T, Ludwig H, Zhang R, et al. A hybrid approach to privacy-preserving federated learning. *Informatik Spektrum*, 2019, **42**(5): 356−357

24 Zhu Y, Hu H, Ahn G J, Huang D J, Wang S B. Towards temporal access control in cloud computing. *Proceedings IEEE Infocom*, 2012, **131**(5): 2576−2580

25 Canetti R, Feige U, Goldreich O, Naor M. Adaptively secure multi-party computation. In: Proceedings of the 28th Annual ACM Symposium on Theory of Computing. Philadelphia, USA: ACM, 1996. 639−648

26 Cireşan D, Meier U, Masci J, Schmidhuber J. Multi-column deep neural network for traffic sign classification. *Neural Networks*, 2012, **32**: 333−338

27 Yin B, Yin H, Wu Y L, Jiang Z X. FDC: A secure federated deep learning mechanism for data collaborations in the internet of things. *IEEE Internet of Things Journal*, 2020, **7**(7): 6348−6359

28 Wang X F, Han Y W, Wang C Y, Zhao Q Y. In-Edge AI: Intelligentizing mobile edge computing, caching and communication by federated learning. *IEEE Network*, 2019, **33**(5): 156−165

29 Yang K, Jiang T, Shi Y, Ding Z. Federated learning via over-the-air computation. *IEEE Transactions on Wireless Communications*, 2020, **19**(3): 2022−2035

30 Engelenburg S V, Janssen M, Klievink B. Design of a software architecture supporting business-to-government information sharing to improve public safety and security. *Journal of Intelligent Information Systems*, 2019, **52**(3): 595−618

31 Qiu M J, Sun P, Zhang B, Yu J H, Fu Y, Yu X, et al. Reliable information encryption and digital display applications based on multistate smart windows. *Advanced Optical Materials*, 2018, **6**(22): Article No. 1800338

32 Hua Z, Zhou Y. Image encryption using 2D logistic-adjusted-sine map. *Information Sciences*, 2016, **339**: 237−253

33 Wang R B, Li Y N, Xu H Y, Feng Y, Zhang Y G. Electronic

scoring scheme based on real paillier encryption algorithms. *IEEE Access*, 2019, **7**: 128043−128053

34 Ganesan I, Balasubramanian A A A, Muthusamy R. An efficient implementation of novel paillier encryption with polar encoder for 5G systems in VLSI. *Computers & Electrical Engineering*, 2018, **65**: 153−164

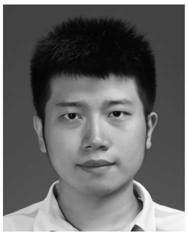
35 Wu H T, Cheung Y M, Huang J W. Reversible data hiding in Paillier cryptosystem. *Journal of Visual Communication and Image Representation*, 2016, **40**: 765−771

36 CSIRO′s Data61. Python paillier library [Online], available: https://github.com/data61/python-paillier, April 12, 2022

37 Seo Y, Shin K S. Hierarchical convolutional neural networks for fashion image classification. *Expert Systems With Applications*, 2019, **116**: 328−339

38 Sun Y, Chen J, Liu Q, Liu G. Learning image compressed sensing with sub-pixel convolutional generative adversarial network. *Pattern Recognition*, 2020, **98**: Article No. 107051

39 Liao Z, Couillet R. A large dimensional analysis of least squares support vector machines. *IEEE Transactions on Signal Processing*, 2019, **67**(4): 1065−1074



张泽辉 南开大学软件学院博士研究生. 2019 年获得武汉理工大学硕士学位. 主要研究方向为联邦学习, 故障诊断和智能船舶控制.
E-mail: zhangtianxia918@163.com
(**ZHANG Ze-Hui** Ph.D. candidate at the College of Software, Nankai University. He received his master degree from Wuhan University of Technology in 2019. His research interest covers federated learning, fault diagnosis, and intelligent ship control.)

富 瑶 南开大学软件学院硕士研究生. 主要研究方向为云端数据完整性验证, 信息安全.
E-mail: FuYao_TJ@163.com
(**FU Yao** Master student at the College of Software, Nankai University. Her research interest covers cloud data integrity verification and information security.)

高铁杠 南开大学软件学院教授. 1991 年获得华中理工大学应用数学专业硕士学位, 2005 年获得南开大学博士学位. 主要研究方向为联邦学习, 图像水印, 信息隐藏和云端数据安全. 本文通信作者.
E-mail: gaotiegang@nankai.edu.cn
(**GAO Tie-Gang** Professor at the College of Software, Nankai University. He received his master degree in applied mathematics from Huazhong University of Science and Technology in 1991, and Ph.D. degree from Nankai University in 2005. His research interest covers federated learning, image watermarking, information hiding, and cloud data security. Corresponding author of this paper.)