基于 BFV 同态加密神经网络参数设置实证研究

杨 涛[1], 董建锋[1,2]

[1]浙江工商大学, 计算机科学与技术学院, 杭州 中国 310012
[2]中国科学院信息工程研究所, 信息安全国家重点实验室, 北京 100093

摘要 基于 BFV 同态加密方案的隐私安全神经网络已经越来越被人们熟知。然而在将其应用到不同场景时, 用户对其众多参数设置的策略, 以及这些参数设置对网络模型预测速度和预测准确率的影响还比较模糊, 影响同态加密神经网络进一步推广和应用。本论文将以微软提出的 Cryptonets 为研究对象, 以实证研究的方式对 BFV 密码方案中各个参数进行设置与调试, 研究参数对加密解密速度、密文增长、网络预测速度、以及网络最终预测结果的影响, 并给出指导建议。从实验结果中发现, 1)多项式模数 N 的设定对网络模型预测的准确性影响最大。较大的多项式模数将带来更高的预测精度, 过小的多项式模数将使预测完全失真。BFV 中其余参数的设置只对运算效率产生影响, 对模型的准确性的影响不大; 2)时间复杂度、空间复杂度都随着多项式模数的增加而增加。密文与明文所占空间之比为 10∶1。随着多项式模数的增加, 神经网络计算的时间复杂度的增加要快于多项式模数的增长。3)在神经网络不同层级中, 池化层和卷积层是同态加密神经网络中计算耗时最长的层级, 增大卷积核的尺寸可以有助于提高效率。总之, 研究同态加密神经网络中的参数配置对于其在不同应用领域中的性能至关重要。本文对不同参数对计算效率和预测准确性的影响的研究, 使我们能够更明智地选择参数和设计网络。随着同态加密在隐私安全机器学习中的更广泛应用, 未来还需要进一步研究其他密码方案的参数配置及其对性能的影响。

关键词 同态加密; 神经网络; 参数配置;
中图法分类号 TP306+.2、TP391.4 DOI 号 10.19363/J.cnki.cn10-1380/tn.2023.05.04

Empirical Study on the effects of BFV Scheme Configuration on Secure Neural Networks Inference

YANG Tao[1], DONG Jianfeng[1,2]

[1] School of Computer Science and Technology, Zhejiang Gongshang University, Hangzhou, 310012, China
[2] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China

Abstract Homomorphic encryption has emerged as an effective solution for preserving privacy in neural network applications. However, the parameter configuration of the BFV homomorphic encryption scheme has not yet been thoroughly investigated for various application domains. This has resulted in difficulties in applying homomorphic encrypted neural networks to other scenarios, as a lack of understanding of how parameter configuration affects inference speed and accuracy can result in suboptimal performance. To address this issue, this paper conducted an investigation of parameter configuration not only in the BFV scheme but also in the homomorphic encrypted neural network model itself. The aim was to reveal the relationships between parameter settings and computation efficiency and inference accuracy. The results of the experiments showed that the polynomial modulus was the most important parameter influencing computation efficiency and accuracy. Larger polynomial moduli resulted in greater accuracy, while smaller polynomial moduli resulted in a significant loss of accuracy. Other parameters in the BFV scheme had no effect on inference accuracy and only affected computation efficiency. Additionally, the polynomial modulus was found to be strongly related to both space and time complexities. As the polynomial modulus increased, both time and space complexity increased, with time complexity increasing much faster than the polynomial modulus. The ciphertext was nearly ten times larger than the plaintext, indicating a significant increase in storage requirements when using homomorphic encryption. The experiments also revealed that pooling and convolutional layers were the most time-consuming layers in homomorphic encrypted neural networks due to the large volume of ciphertext computation. To mitigate this issue, the paper advised reducing the use of pooling layers or replacing them with other network structures. Moreover, increasing the size of the convolution kernel was found to reduce the time-complexity of inference computation. In summary, the investigation of parameter configuration in homomorphic encrypted neural networks is critical for achieving optimal performance in various application domains. The findings of this paper provide valuable insights into the impact of different parameters on computation efficiency and inference accu-

通讯作者: 董建锋, 博士, 研究员, Email: danieljf24@163.com。
本课题得到浙江省自然科学基金(No. LQ20F020008)与中科院信工所开放课题(No. 2021-MS-03)提供资助。
收稿日期: 2022-09-11; 修改日期: 2022-12-24; 定稿日期: 2023-03-24

racy, allowing for more informed decisions regarding parameter selection and network design. As homomorphic encryption continues to widely applied in privacy-preserving machine learning, further research is needed to explore additional aspects of parameter configuration and their impact on performance.

Key words homomorphic encryption; neural networks; parameter settings

1 引言

随着国内外对于隐私安全、数据安全关注的加深以及相关法律法规的出台, 越来越多的人工智能技术开始关注其所在应用场景的隐私安全问题[1]。如在依靠人脸识别信息的进行身份认证时, 人们关心自己的人脸信息有没有被滥用。在医院使用基于深度学习模型时, 人们关心自己的病例信息会不会被用来训练模型, 并造成隐私外泄。其中许多深度学习的隐私安全问题, 都涉及云计算场景中的隐私安全。在该场景下深度学习模型成为一种可调用的远程功能模块为用户提供如语音识别、目标检测、人脸识别等服务。用户只需将自己的数据上传到云端, 通过部署在云端的深度学习模型的处理, 即可得到自己所需要的结果。然而在该种场景下用户的数据需要上传到云端的服务器, 数据离开本地在传输的过程中或云端处理的过程中, 都存在隐私泄露的风险。部署于云端平台上的深度学习模型, 在为用户远程提供深度学习服务的同时, 如何能保证用户上传的数据的隐私不会被泄露, 成为了一个亟需解决问题。

为了构建适用于云端的隐私安全深度学习模型, 研究者尝试了同态加密、差分隐私和安全多方计算等多种方法。差分隐私原本是一种防范差分攻击的方法, 通过向数据集中添加随机噪声, 使得某一特定样本包括或不包括在数据集中, 对查询的结果不产生影响, 以此使得在数据集中的隐私数据不会泄露。当差分隐私技术应用于深度学习模型的训练时, 通过向每次反向传递的梯度中添加一定量的噪声, 使得训练得到的参数并不完全反映数据集的分布, 从而保护数据集的隐私, 构造满足差分隐私要求的深度学习训练系统[2]。多方安全计算则是面向一组计算的参与者, 每位参与者拥有自己的数据, 且不信任其他参与者与任何第三方, 如何对各自私密的数据计算出一个目标结果的过程。多方安全计算的关键在于设计一个能让各方参与者诚实执行的协议。然而在实际应用的过程中, 差分隐私和多方安全计算技术, 都存在一定的局限性。差分隐私不能保护数据在传输过程中的隐私暴露问题。如果未经加密的数据在传输过程中被截取, 则存在隐私暴露的风险。多方安全计算的应用需要精心设计安全协议, 甚至需要额外的硬件设备才能完成; 一些计算方法与没有使用多方安全计算时的方法差别甚大, 不利于将现有神经网络模型转化成符合多方安全计算的模型。此外, 多方安全计算的协议往往需要不同参与者之间的交互, 与现有神经网络训练和预测端到端一步完成的形式, 差别较大。因此在深度学习领域的实现还存在较大的难度。

本论文所关注的同态加密技术, 可以在密文状态下完成计算, 可以保证在数据传输时的安全性。同时现有的计算操作与同态加密操作的转换相对简单, 与神经网络技术结合紧密, 可以在无需用户交互的情况下完成同态加密神经网络的训练和预测。因此同态加密技术正越来越多地被应用到各个场景中。其中微软公司所提出的 Cryptonets[3-4], 以及之前公开的 SEAL 同态加密库[5], 是目前应用最广泛的模型和同态加密神经网络框架之一。Cryptonets 主要利用同态加密技术改造 Lenet5 神经网络模型, 并在密文的状态下完成对加密的手写字母图片完成识别, 以此作为一套原型系统来验证其对神经网络改造的有效性。

在使用同态加密技术时, 由于额外的加密解密计算、以及在密文状态下计算使得整个计算过程的时间复杂度与空间复杂度都大为增加, 这不但是人们在使用同态加密技术时最关心的问题, 而且也是限制同态加密技术进一步推广应用的原因之一。

我们认为对密码方案中参数设置, 以及神经网络中的一些关键参数的设置将会对时间复杂度与空间复杂度, 以及神经网络模型的预测准确率带来较大的影响。参数设置对加密神经网络性能的定量分析一直比较缺乏。尤其对最终神经网络识别和判断的准确率有多少影响, 一直没有一个清楚的认识。这使得用户在利用 BFV 密码方案时, 不但需要经过长时间的调参, 才能找到适应具体任务的参数配置, 而且一旦选取了不合适的参数, 不但影响模型的计算效率, 同时也将影响模型的准确率。

本论文将对应用于 SEAL 中已实现的 BFV 密码方案中的多个关键参数, 以及其在 Cryptonets 网络模型中的多个重要参数进行评测, 研究不同参数对于计算过程中密文增长、时间复杂度、以及网络预测

准确率的影响, 为普通用户正确地选择合适的 BFV 参数提供合理的建议。

本论文第二部分介绍同态加密、隐私安全神经网络、同态加密框架的相关背景。第三部分将讨论 BFV 密码方案与 Cryptonets 同态加密神经网络中多个重要参数的含义及原理。第四章介绍相关实验环境的搭建与实验结果。第五章为全文的总结和对未来 BFV 方案参数设置的建议。

2 相关工作

2.1 全同态与有限同态加密

同态加密是加密的一种形式, 其允许对加密后的数据进行一些特定的代数运算, 计算得到的结果解密后将与直接在明文数据上进行运算的结果相同。由于所有的运算都是在加密域中完成, 因此能够保证数据的隐私性。同态加密的概念最早由 Rives 等人在 1978 年提出[6], 之后 Elgamel[7]与 Paillier[8]分别提出了基于加法同态的加密方案。

假设有两个正整数 x_1、x_2, 经过公钥 pk 加密后可得密文态的 $\hat{x}_1$, $\hat{x}_2$。 $\hat{x}_1 \coloneqq E(pk,x_1),\hat{x}_2 \coloneqq E(pk,x_2)$。在最早的几年中, 在加法同态加密中的 $\oplus$ 操作已经可以完成 $E(pk,x_1+x_2) \leftarrow \hat{x}_1 \oplus \hat{x}_2$ 操作。在之后的几年里, 带常数的乘法同态加密也可以实现 $E(pk,a\cdot x_1) \leftarrow a\cdot \hat{x}_1$。在 2009 年时 Gentry 等人首先提出了基于理想格的可以完成任意次数的加法与乘法运算的全同态加密方案[9]以及基于最大近似公因子问题的变种方案 DGHV[10], 被称为第一代全同态加密方案。然而由于其密钥尺寸过大, 计算效率较低, 很快被第二代全同态加密取代。基于格上的“带错误学习(Learning With Error, LWE)”被称为第二代全同态加密, 该种全同态加密实现了在标准格上的同态加密, 但是由于密文是向量, 通过张量积的操作实现同态加密使得密文的维数急剧增加, 带来计算开销过大。第三代全同态加密方案是由 Gentry 等人提出的基于矩阵近似特征向量的 GSW 同态加密方案[11], 该方案的特点为密文有一个矩阵构成, 可以进行自然的乘法加法运算, 避免了密文维数膨胀的问题。

由于全同态加密的效率问题, 人们在提出全同态加密之后, 后续又提出了有限同态加密(somewhat homomorphic encryption)的概念。有限同态加密允许对密文进行有限次数多项式函数运算。其中具有代表性的有[12-14], 有限同态加密对被加密的多项式的最高次数有要求。高次的多项式在加密时需要大的参数, 不但带来更大的计算复杂度, 也会导致体量较大的密文, 使得整体效率降低。在同态加密技术的应用中, 运行效率问题一直人们比较关注的问题。其核心问题是如何在效率与安全性上找到一个平衡点, 在不牺牲太多运行效率的同时, 保证数据的安全性。

2.2 隐私安全的神经网络

近些年来, 神经网络的快速发展已经推动人工智能领域进入了一个新的阶段。人们已经根据实际的种种需要研发出了不同种类的神经网络, 如深度置信神经网络, 卷积神经网络, 循环神经网络。考虑到数据的隐私性和安全性, 在本世纪初研究人员提出了加密神经网络的概念。加密神经网络是指在密文上进行训练和预测的神经网络[3], 现有的方法都是基于安全多方计算或是同态加密的方法, 或是它们两者的结合。Barni 等人[15]结合了安全多方计算与同态加密的方法, 其中同态加密的方案使用的是 Paillier[8]基于加法的同态加密, 所用到的安全多方运算协议基于 Yao 百万富翁问题[16]。之后 Orlandi 等人[17]于次年进一步改进了[15]的方法, 采用了更为先进的同态加密方案[18], 同时中间结果将被隐藏直到最后才向客户端公开。[15]与[17]提出了 Oblivious neural network 的概念, 即为数据所有方与计算服务方互相之间不清楚对方的情况, 数据所有方只是使用计算服务方所提供的模型服务, 对其中的参数并不清楚。计算服务方只是提供计算服务给数据所有方使用, 对上传进来的数据具体情况并不知情。这样的神经网络保护了双方的隐私不被泄露, 为以后的安全神经网络做出了启示。类似[15]和[17]这样基于同态加密与多方安全计算的方法近年来还有很多, 如[19-20], 由于多方计算需要人们对协议的交互性输入, 系统整体的计算复杂度与传输损耗仍然很大。Gilad 等人提出了一种可应用于加密数据的近似神经网 CryptoNets[3]。它不借助客户端与服务器端的交互, 数据在客户端上加密之后, 传输到服务器端。服务器端运行的网络也是在原有网络基础上改进的可以处理加密数据的网络, 数据一直在加密域中处理, 直到最后传输给客户端后再由客户端解密。CryptoNets 在 MNIST 手写识别数据集上做了测试达到了较高的准确率, 然而由于它所采用的平方函数作为激活函数使得无法处理较深层的网络, 并将原有 max-pooling 函数改为 mean-pooling 使得识别的准确性有所降低, 并且在训练时需要积累数据达到一定数量后, 成批次训练, 无法满足可用性的需求, 因此仍然没有达到实用性的要求。

3 BFV 密码方案与关键参数

3.1 BFV 同态加密方案

BFV(Brakerski-Fan-Vercauteren)密码方案是第一代全同态算法改良后的第二代全同态方案。BFV 方案基于带误差的环学习(RLWE)困难问题。与第一代算法相比, BFV 通过重线性化技术[21]来控制密文的维度, 进而通过维度-模数约减技术减少解密算法的复杂性, 提升运算效率。由于 BFV 方案的数学基础是环, 所以明文与密文都是由环上的多项式组成。其主要思想是将明文放在密文的高位, 在解密时通过整除的方式, 去除低位上的噪声, 得到明文。以下实现过程为生成公私钥, 加密, 解密, 和重线性化的一些步骤:

首先 p 为明文模, q 为一个大整数模, 明文空间为环 R/pR.

1) 生成私钥 sk: 选取 $s \leftarrow \chi$, 输出 $sk=(1,s)$. χ 为一个随机分布。私钥分布与噪声分布一般是一个分布, 实际中往往选择是离散高斯分布。

2) 生成公钥 pk: 设定 $s=sk$, 选取 $a \leftarrow R_q$, $e \leftarrow \chi$, 输出 $pk=(b,a)$, 其中 $b=-(as+e)$. a 由明文空间均匀随机采样得到, e 由噪声空间采样得到。

3) 加密函数 (pk,m): 对于一个明文首先需要将其编码到明文空间 R/pR.编码方法是需要一个整数 t, 一个数可以表示为 t 进制数相加的形式, 因此消息 $m \in R_t$, 加密得到密文 $ct=(c_0,c_1)$, 其中 $c_0=bu+e_1+\omega m$, $c_1=au+e_2$. e_1 和 e_2 为从噪声分布中两次采样取得的噪声, u 为从私钥空间取得的随机数。 ω 为 $\lfloor q/p \rfloor$。

4) 解密函数 (sk, ct): 设定 $s=sk$, 密文 $ct=(c_0,c_1)$ 解密为 $m=\lfloor (ct\cdot s)/\omega \rfloor$

5) 加法同态性: 假设 $ct_i(i\in 1,2)$ 为两个密文, 那么 $\mathrm{Add}(ct_1,ct_2) := (ct_1[0]+ct_2[0], ct_1[1]+ct_2[1])$.

6) 乘法同态性: 步骤为先将 $ct_i(i\in 1,2)$ 相乘, 再重线性化。假设 $ct_1=(a_0,a_1)$, $ct_2=(b_0,b_1)$, 相乘后密文为 $ct_1ct_2=(a_0b_0, a_0b_1+a_1b_0, a_1b_1)$, 其密文的私钥 $s=(1,2s,s^2)$, 因为 $cs\cdot cs=cc\cdot s^2$

乘法运算后密文由二维变为三维, 需要重线性化, 重线性化需要借助辅助密钥, 通过重新计算噪声, 能够将密文转为二维。

从以上的步骤中可以可以看出, 同态的加法和乘法都遵循不同于普通加法乘法运算的规则。由于在同态加密运算规则涉及更多的参数运算, 并且需要妥善处理随时可能增长的噪声, 因此采用以同态加密作为隐私保护方法的神经网络, 其时间复杂度和空间复杂将远超过普通神经网络。其中可以看出由于重线性化步骤的加入, 同态加密乘态加密乘法和加法对时间复杂度和密文增长的影响展开评测, 探究参数的变化对同态加密乘法和加法运算效率的影响。法的复杂度远大与加法运算。

3.2 编码

同态加密的运算对象为环上的多项式, 所有参与 BFV 同态加密运算的整数或浮点数都需要转化为多项式的形式才能参与运算, 这一个转化为多项式的过程被称为编码。

将整数转化为多项式的方法有多种, 以下是 SEAL 类库所支持的几种编码方式。

表 1 SEAL 中可用于 BFV 的编码方式

Table 1 Encoding method in BFV scheme provided by SEAL Library

编码方式	所需操作
标量编码	标量编码是最简单的一种编码形式。它将每个常数, 编码成一个多项式。虽然这在理论上可行, 但是效率较低, 大量计算资料被浪费
整数编码	整数编码的形式, 是将某一个整数 M, 写成某个整数 B 的几个整数次幂相加的形式。例如, 当整数 B 为 2 时, 整数 $M=a_1\cdot 2^n+a_2\cdot 2^{n-1}+\cdots+a_n\cdot 2^0$
批处理编码 CRT batching encoding	批处理编码是 SEAL 类库支持的最高效的编码方式。使用批处理编码方式可以将 N 个整数同时编码到一个 N 项的多项式中。其中 N 是 BFV 方案中的多项式模数, 它控制了有多少数据可同时编码, 生成的多项式有多少项 批处理编码中所用到的技术, 包括中国剩余定理 CRT 和 SIMD 技术, 即通过连乘的方式, 可以对 N 个整数进行编码, 且每个整数都可以编码成 N 项的多项式[21]

Cryptonets 在实现时采用了批处理编码的方式, 每次将 N 张图片同时处理, 以降低单张图片的时间和空间开销。然而当每次批处理时, 输入图片的数量没有达到足够的数量, 单张图片的时间空间开销仍然会很大。

3.3 矩阵旋转与 Galois 密钥

在对密文进行连乘连加操作时, 需要进行矩阵的旋转操作。矩阵的旋转操作就是将矩阵内的元素按行或列循环移位。在密文态下, 旋转需要先生成 *Galois Keys*。

在 SEAL 库中, Galois 密钥的生成被封装成函数, 可以自动生成。只有通过修改配置文件才能修改。在设置 *Galois Keys* 时, 可以对 *Galois Decomposition BitCount* 进行设置。

3.4 加密

在加密过程中影响时间和空间复杂性的参数包括图片的尺寸、参与预测的图片数量, 编码形式, 以及神经网络中批处理大小(一次有多少张图片参与运算)。在 Cryptonets 中, 因为使用了 SIMD 技术, 一次参与运算的图片量由 N 多项式模数决定。模型为了提高整体的处理效率, 一次将对 N 张图片进行预测。如果输入图片数量未达到 N 张, 将由 0 补齐。

3.5 解密和解码

解密和解码过程中所涉及的参数与编码和加密中, 只是解密与加密不同的是, 解密过程所用到的密钥是私钥。解密通过私钥将密文解密为一个明文多项式。最后将编码的多项式还原为预测结果。

Cryptonets 模型的基本流程如下图 1 所示。N 为多项式模数, q 为系数模量, T 为选取的基准。

4 Cryptonets 神经网络及其关键参数

Cryptonets 网络结构与 Lenet-5 手写数字识别模型类似, 只是通过 BFV 同态加密密码方案将普通运算如乘法运算, 替换为符合同态加密规则的运算。其中一些非线性的函数如激活函数, 还将被替换为线性函数, 并将其转化为符合同态加密规则的函数运算。经过修改后的 Lenet-5 的网络模型可以处理加密后的数字图片, 并将识别后的结果以加密数据的形式返回给用户。用户可以将加密后的手写数字图片上传, 并能够在密文状态下识别手写数字图片。

本节将对加密神经网络中多个可能影响性能的参数进行分析。

4.1 Cryptonets 神经网络基本架构

表 2 中介绍 Cryptonets 网络各个网络层的参数。在训练时各个层级都会发挥作用, 但是当预测时, 只有第一层、第二层、第三层、第七层、第九层发挥了作用。

对于一个加密神经网络来说, 训练阶段往往是在明文的状态下, 由大规模的明文数据训练得到。普通用户更关心经过同态加密改造后的模型在使用加密数据预测时的运行效率。

在第五节实验部分我们将对预测模型不同层级的时间复杂度和空间复杂度进行评测, 为普通用户在设计同态加密神经网络架构时提供建议。

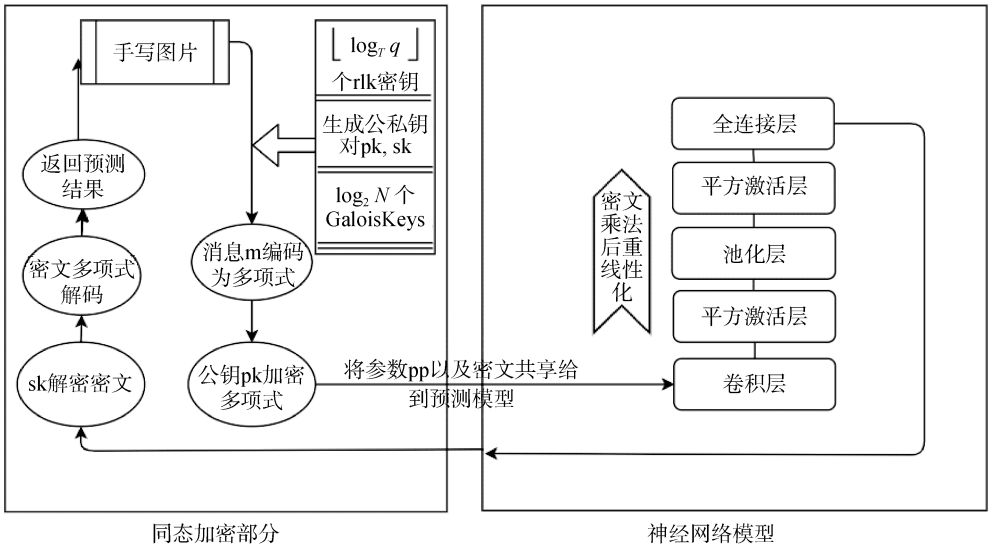

图 1 Cryptonets 加解密与预测示意图

Figure 1 Cryptonets encryption/decryption and inference process

表 2 Cryptonets 网络结构

Table 2 Cryptonets Network Structure

层数	名称	描述
1	卷积层	输入图片的大小为 28×28, 卷积核的大小为 5×5, 步长为 2, 共有 5 个卷积核, 所以输出的特征图的大小为 5×13×13
2	平方激活层	对每一个输入进行平方运算
3	平均池化层	经过一个 1×3×3 卷积得到 5×13×13 的特征图
4	卷积层	卷积核大小 5×5, 步长为 2, 共有 10 个卷积核, 最后输出的特征图的大小为 50×5×5
5	平均池化层	经过一个 1×3×3 卷积得到 50×5×5 的特征图
6	全连接层	全连接层连接 50×5×5 个输入节点和 100 个输出节点
7	平方激活层	对每一个输入进行平方运算
8	全连接层	全连接层将 100 个输入节点和 10 个输出节点
9	Sigmoid 激活层	对每个输入的节点的值计算 sigmoid 的值, 最后得到识别结构

4.2 卷积

在卷积神经网络中卷积层中的卷积操作是提取图像特征的重要方法。其中所涉及的参数有卷积核尺寸、填充(padding)、步长、卷积核数量等。这些参数参与卷积层的运算, 并对整体的时间复杂度和空间复杂度产生影响。

卷积核尺寸: 在卷积时卷积核的尺寸控制了感受野的大小, 标志在多大范围内的信息参与运算。不同尺寸的卷积核影响了同时有多少像素值参与乘法运算与加法运算。

填充(padding):padding 的作用是调整输入特征图的大小。通过在图片周围添加填充物以满足不同卷积核的计算。

步长: 步长和填充一起对最后生成的特征图尺寸产生影响。一般情况下, 步长越大输出特征图相对于输入特征图缩小得越多, 后续所需的计算量也将减少。

卷积核数量: 参与卷积运算的卷积核一般随机生成, 卷积核数量对应着最后生成的特征图通道的数量。所需要的特征图通道越多, 所需的计算量越大, 时间复杂度也相应的增加。

4.3 层级

不同网络层级经过同态加密函数转换之后, 各层所需要的计算时间有所不同。本论文将对预测时各层所需的时间加以评测, 确定各层的时间复杂度。为用户设计网络架构提供效率方面的建议。

4.4 激活函数

在 Lenet-5 的网络模型中, 激活函数为 ReLU 函数。由于 ReLU 函数不属于线性函数, 不能被同态加密转化, 因此在 Cryptonets 的实现中将 ReLU 函数替换为平方函数。平方函数所涉及到的同态加密中的乘法运算所需要的时间将多于 ReLU 函数。

然而平方函数在后续的实验中也发现收敛效果不佳, 如果换成其他激活函数, 其时间复杂度也将是需要考察的因素。

4.5 池化层

池化层的作用是将特征做一个筛选, 降低特征的维度。降低特征的维度不但是防止模型过拟合、增加模型不变性的需要, 而且也是减少计算量的要求。

在卷积神经网络中, 池化层所用到的常见方法有平均池化法、最大值池化法。由于最大值函数不是线性函数, 因此在 Cryptonets 同态加密神经网络中, 使用了平均池化法。

在 Cryptonets 中, 平均池化需要对 1×3×3 卷积核中的每一个特征值进行平均运算, 运算量较大, 对整体时间复杂度影响较大。

5 实验测试与讨论

5.1 实验设置

本论文所涉及的实验都在阿里云服务器上完成。所租用的服务器, 具体的软硬件配置如下:

为了能对各个参数进行动态的调试, 我们开发了一个用户可自主测试模型的测试平台, 用来向服务器端的模型传递参数, 并记录返回结果, 最终生成图表。

实验中所使用的数据集为 MNIST 手写数字数据集。实验中我们随机选择了 10000 张图片参与 Cryptonets 模型的预测实验。

表 3 所测试系统的硬件配置

Table 3 Our testing system's hardware configuration

软硬件配置	服务器
操作系统	Windows sever2019
内存	4G
硬盘容量	40G
处理器	Intel(R) Xeon(R) Platinum 8163
核数	2 核

5.2 BFV 方案参数对性能的影响

论文中所涉及的 BFV 同态加密方案来自微软 SEAL 同态加密库[3]。我们对多个参数进行的测试。具体如表 4 所示。其中最主要的参数是多项式模数 N, 如 3.1 中的介绍, 它不但表征了输入图片每个像素值会被编码成有多少项的多项式, 而且也表征了会多少个像素值同时加密。如表 5 所示, 在 SEAL 中有三个多项式模数可以选择, 并为每个多项式模数自动生成了对应的明文模量。用户选择不同的多项式模数 N, SEAL 将自动选用相应的明文模量对输入信息进行编码。

5.2.1 多项式模与识别准确率

本实验通过调整不同的多项式模数, 测试其对最终神经网络识别准确率的影响。表 6 展示的是不同多项式模数对最终手写识别率的影响。当多项式模数为 2048 时, 识别率为 10%~11%。只有十分之一的数字被正确识别, 约等于平均分布。由此可见, 此时网络基本不具备识别数字的能力, 可以推测此时过小的多项式模数使得神经网络在做乘法时产生的噪声过大, 进而在解密时不能回到正常的数值区间。由于返回的数值为一些随机的噪声, 使得最后只十分之一的数值识别正确。

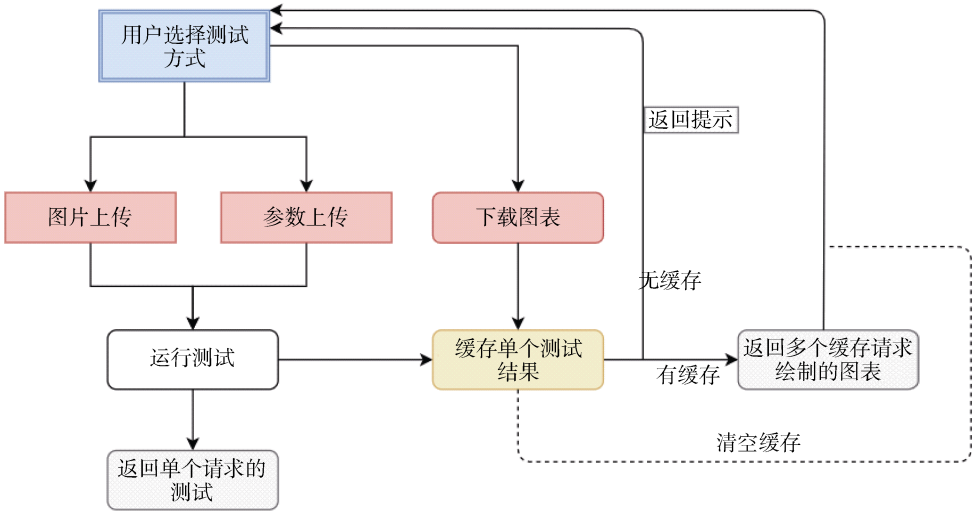

图 2 测试平台与云服务器数据流图

Figure 2 The dataflow between testing platform and the server in the cloud

表 4 SEAL 中 BFV 方案所涉及的重要参数

Table 4 The key parameters in BFV scheme provided by SEAL

名称	含义
Galois Decomposition BitCount	影响 Galoiskey 生成的因素
poly_modulus_degree (batchsize)	多项式模数
relin_Scale	密文重线性化程度
Decomposition BitCount	影响重线性化密钥生成的因素
编码方式(*Integer/batch*)	整数编码与批处理编码

当多项式模超过 4096 时, 识别准确率较高, 已有 97%的数字识别正确。当多项式模为 8192 时识别准备率达到 98%~99%, 已经达到未加密的普通神经网络的水平。因此可以推测 1)当多项式模足够大时可以提高神经网络识别的精度。当多项式模数较小时, 网络模型无法正常工作。2)更大的多项式模数可以有助提高精度。当神经网络结构较复杂, 层数较多时选择较大的多项式模数可以保证网络的准确性。

5.2.2 其他参数与识别准确率

在实验中我们也对 BFV 其他参数进行了调试, 其中 *weightscale* 旨在对 *cryptonets* 中的权重加以缩放, *scale* 是 BFV 中每次重线性化所使用的参数。在

表 5 测试中所用到的多项式模量与明文模量

Table 5 Polynomial modulus and plaintext modulus in our experiments

多项式模数 N	明文模量 p
2048	{ 65537, 40961, 114689, 147457, 188417 }
4096	{ 40961, 65537, 114689, 147457, 188417 }
8192	{549764251649, 549764284417}

表 6 多项式模 *N* 与网络识别准确率的影响

Table 6 The relationship between polynomial modulus and recognition accuracy

多项式模 *N*	识别准确率
2048	10%~11%
4096	97%
8192	98%~99%

神经网络的运行过程中会对多项式密文做乘法, 加法运算, 乘法使得密文的噪声翻倍, 为防止密文噪声太大导致解密误差很大, 需要对乘法结果进行重线性化运算。*Scale* 大小决定了密文还原回原大小所要约减的程度。

表 7 中展示了不同 *weightscale* 和 *Scale* 对识别准确率带来的影响。从实验结果可以看到, 当多项式模数固定时, 不同的 *weightscale* 和 *Scale* 的组合, 对识别准确率的影响不大。

表 7 *Weightscale* 与 *Scale* 参数对识别准确率的影响

Table 7 The recognition accuracy inferenced by *weightscale* and *scale*

多项式模数	*Weightscale*	*Scale*	识别准确率
4096	6	12	97%
4096	6	6	97%
8192	20	12	99%
8192	20	6	99%
8192	40	6	99%
8192	48	6	99%

实验中还对神经网络中 *normalization factor*、卷积核大小等参数进行的测试, 这些对最终的识别准确率没有影响。

5.2.3 加解密过程时间与空间复杂度

我们对选取不同的多项式模数时, 加密解密所消耗的时间, 模型预测的时间、明文密文在内存中所占空间的大小进行的评测。实验结果如表 7 所示。

从表 8 中可以看出, 随着多项式模数的增长, 明文所占空间大小和密文所占空间大小呈现出明显的翻倍的关系。明文和密文所占空间大小约为 1∶100。换句话说, 加密之后密文是明文的 100 倍。此处, 明文是以批处理编码(batch encode)的形式编码。从明文(编码后)和消息(编码前)所占空间大小的对比可以看出, 在批处理编码之后, 明文所占空间反而缩小了一些, 说明批处理编码的高效。

对比加密和解密的耗时, 可以发现加密的耗时约为解密耗时的 8~10 倍。加密过程的时间复杂度远大于解密过程。

表 8 多项式 *N* 与加密网络时间与空间复杂度的关系

Table 8 The relationship between polynomial modulus, time complexity, and spatial complexity

	N=2048	*N*=4096	*N*=8192
加密耗时	0.892s	1.35s	1.611s
解密耗时	0.117s	0.159s	0.188s
模型预测耗时	5.7s	14.77s	24.374s
明文	1.5M	3.1M	6.1M
密文	147.0M	294M	588.8M
消息大小	1.9M	3.8M	7.5M

从表 8 中可以看出, 加密和解密的耗时虽然随着 *N* 的翻倍而增加, 但是没有呈现翻倍的情况。然而, 在 *N*=8192 时模型预测耗时 24.374s, 而在 *N*=2048 是, 模型预测耗时 5.7s。当 *N* 增长 4 倍的时候, 模型预测的耗时增长快于 4 倍。由此可以推测随着 *N* 的增加会使得网络的复杂度呈快速增加, 并将影响模型整体的时间复杂度。

5.2.4 重线性化参数 *relinKeys*

在对影响重线性化 *relinKeys* 的配置参数 *DecompositionBitCount* 的测试中我们发现, 若取值小, 则重线性化过程的计算速度较慢, 反之, 取值大, 则计算速度较快。在池化层中, 模型涉及大量的乘法运算, 每次乘法运算都需要运用重线性化将密文重新回到 *size* 为 2 的状态。从图 3a 和图, 3b 中可以看出, 当 *DecompositionBitCount* 参数从 1 增加到 10, 模型预测时池化层耗时也从 1.6s 减少到 0.9s, 说明当 *DecompositionBitCount* 较大时将提高重线性化的速度。

然而较大的重线性化因子会导致每次运算中 *relinkey* 占据更大的噪音预算, 使得最大乘法次数降低了。因此仍需要选择合适的 *DecompositionBitCount*.

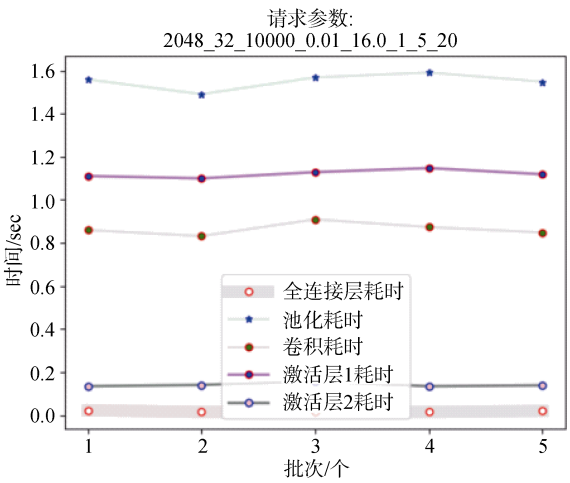

图 3a *DecompositionBitCount*=1 时网络各层耗时

Figure 3a Time consumption in each layer when *DecompositionBitCount*=1

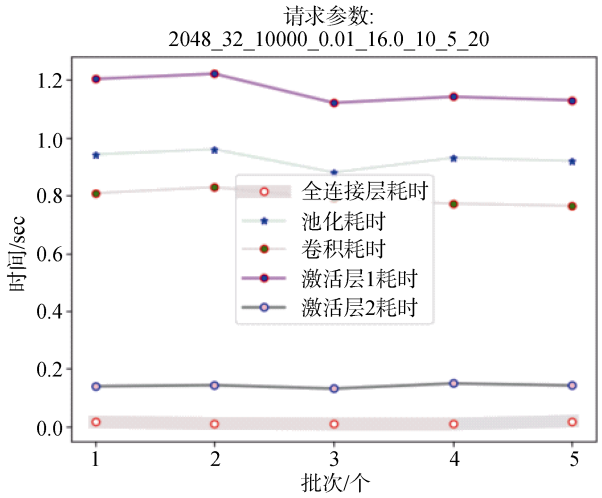

图 3b　*DecompositionBitCount*=10 时网络各层耗时

Figure 3b　Time consumption in each layer when *DecompositionBitCount*=10

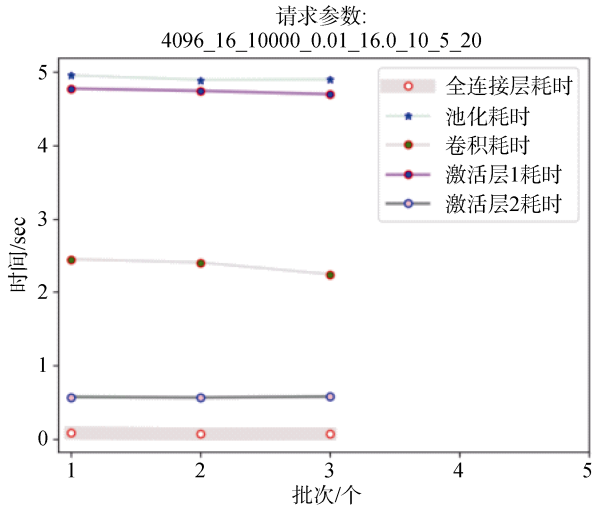

图 4b　*N*=4096 时网络各层耗时

Figure 4b　Time consumption in each layer when *N*=4096

5.3　神经网络参数对性能的影响

5.3.1　加密神经网络各层时间复杂度

我们同时对不同多项式模数下，Cryptonets 网络各层中计算耗时的多少进行了测量。由图 4a，图 4b，图 4c 可以看出，在同态加密化后，前三层(卷积层、池化层、激活层 1)耗时最多，且耗时相差很大，后两层(激活层 2、全连接层)耗时较少，且耗时相差不大都在 1s 之内。在选择不同的多项式模数的情况下，卷积层，第一个平方激活层，和池化层的耗时随 N 增大而增加明显，而第二个平方激活层与全连接层则受影响较小。

从图 4a，图 4b 和图 4c 中可以看出，池化层耗时明显高于第一个平方激活层和卷积层，最后的全

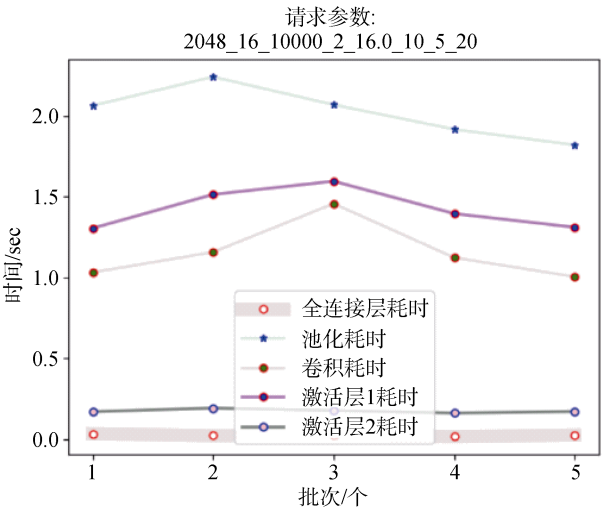

图 4c　*N*=2048 时网络各层耗时

Figure 4c　Time consumption in each layer when *N*=2048

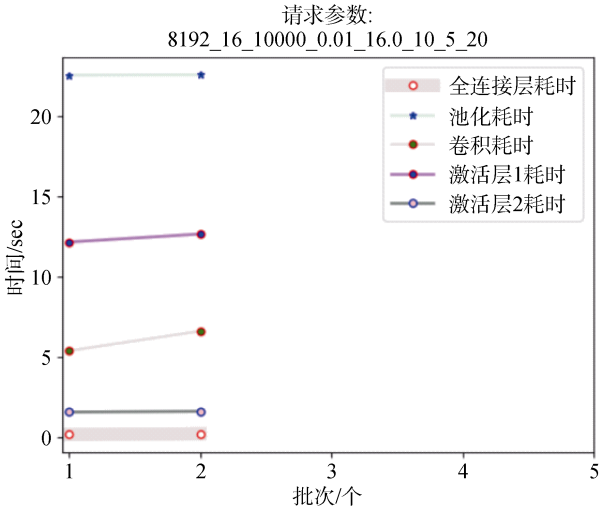

图 4a　*N*=8192 时网络各层耗时

Figure 4a　Time consumption in each layer when *N*=8192

连接层与第二个激活层耗时极小。池化层耗时最多的原因，推测是由于需要将一个卷积核范围内的特征层求平均值，涉及大量的密文乘法运算。全连接层与第二个激活层耗时较少，符合它们输入的神经元较少的规律，池化层压缩了特征值后将变少的神经元输出给后两层，使得后者密文计算量变小了所以需要的计算时间较少。

5.3.2　不同卷积核大小的影响

在对加密神经网络的评测中我们也测试了不同的卷积核大小对时间复杂度的影响。由图 5a，图 5b，图 5c 的对比中可以看出，当使用 1×1 卷积核，卷积层耗时 2s 左右，池化层耗时 6s 左右，激活层 1 耗时 5.2s 左右。当使用 3×3 卷积核，卷积层耗时 0.8s，池

化层耗时 5.7s 左右, 激活层 1 耗时 6s 左右。当使用 5×5 卷积核时, 卷积层耗时 1s 左右, 池化层耗时 2s 左右, 激活层 1 耗时 1.4s 左右。当 1×1, 3×3 大小的卷积核与 5×5 的卷积核相比时, 使用 5×5 的卷积核三个层级平均减少了 4s 左右的运算时间, 它的卷积耗时, 池化耗时和激活层耗时都更少。在三个卷积核的准确率几乎相等的情况下, 5×5 的卷积核显得更高效。

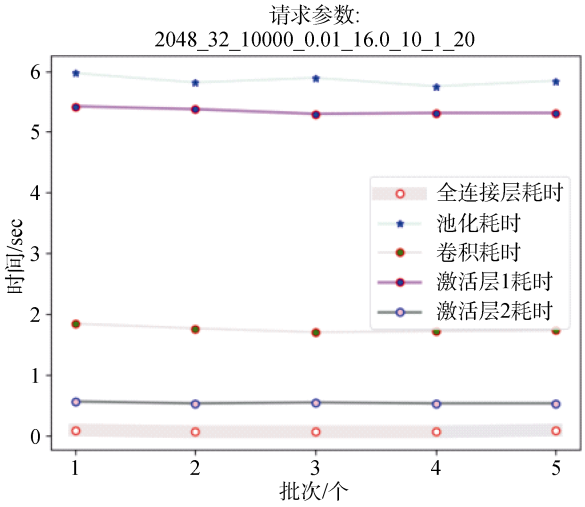

图 5a 卷积核 *size*=1 时网络各层耗时

Figure 5a Time consumption in each layer when *kernelsize* =1

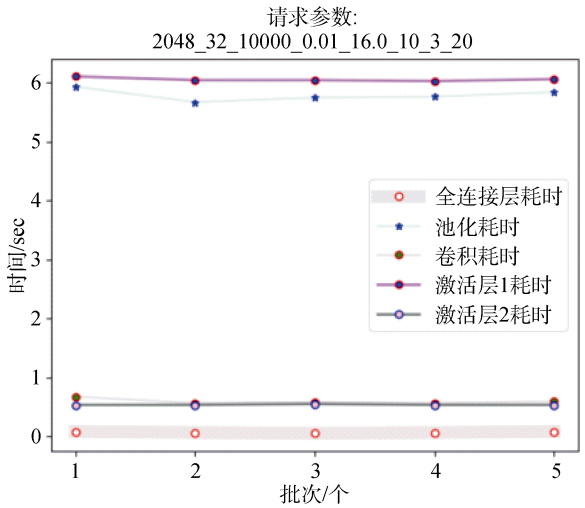

图 5b 卷积核 *size*=3 时网络各层耗时

Figure 5b Time consumption in each layer when *kernelsize* =3

在同样的准确性的前提下, 使用 5×5 的卷积核显得更高效一些。在卷积网络的设计中, 往往出于对计算量的考虑将 5×5 这样的大卷积核换成 3×3 的卷积核。然而在同态加密的神经网络中, 将大卷积核替换为小卷积核的优势被加密方案的开销所取代。反而使用大卷积核增大感受野, 所得到的特征图尺寸更小, 对之后的池化层, 激活层的计算都带来很大的便利。

在同态加密神经网络的设计中, 可以尽量使用大卷积核减少池化层和激活层的乘法的计算压力。

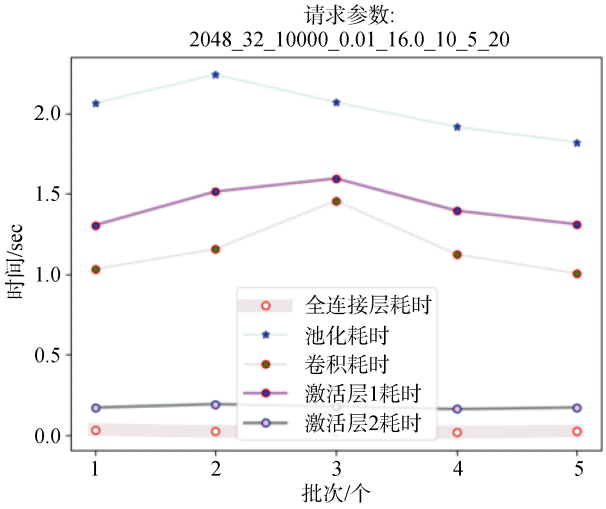

图 5c 卷积核 *size*=5 时网络各层耗时

Figure 5c Time consumption in each layer when *kernelsize* =5

6 总结

本文通过对 BFV 密码方案与 Cryptonets 同态加密神经网络中各个参数的配置与评测, 研究 BFV 同态加密方案里各个参数对神经网络识别准确率以及时间空间复杂性的影响, 为 BFV 在其他神经网络中的应用提供配置的建议与提示。经过测试, 本论文发现多项式模数是对性能和效率影响最大的参数。一方面较大的多项式模数可以带来更精确的预测结果, 另一方面随着多项式模数的增长, 时间复杂度和空间复杂度都在增长, 且时间复杂度的增长要快于空间复杂度的增长。因此, 在硬件条件允许的情况下尽量使用较大的多项式模数。在评价不同网络层级时, 池化层在不同多项式模数的条件下, 一直是耗时最长的层级, 甚至超过了卷积层。为了减少池化层对预测时间的影响, 应该减少或不使用池化层, 并用其他形式如空洞卷积代替池化层。在卷积层中较大的卷积核对提高计算效率有较大的帮助。不但可以获得更大的感受野, 而且可以使特征图尺寸更小, 减少后续层级的计算量。因此, 在设计同态加密神经网络时, 应尽量使用大的卷积核。

致 谢 本课题得到浙江省自然科学基金(LQ20F020008)与中科院信工所开放课题(2021-MS-03)提供

资助。

参考文献

[1] Zhao Z D, Chang X L, Wang Y X. A Survey of Privacy Preserving in Machine Learning[J]. *Journal of Cyber Security*, 2019, 4(5): 1-13.
(赵镇东, 常晓林, 王逸翔. 机器学习中的隐私保护综述[J]. *信息安全学报*, 2019, 4(5): 1-13.)

[2] Abadi M, Chu A, Goodfellow I, et al. Deep Learning with Differential Privacy[C]. *The 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016: 308-318.

[3] 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*, 2016:201-210.

[4] Brutzkus A, Elisha O, Gilad-Bachrach R. Low Latency Privacy Preserving Inference[EB/OL]. 2018: arXiv: 1812.10659. https://arxiv.org/abs/1812.10659

[5] Chen H, Laine K, Player R. Simple Encrypted Arithmetic Library - SEAL v2.1[C]. *International Conference on Financial Cryptography and Data Security*, 2017: 3-18.

[6] L, Adleman L, Dertouzos M L, On data banks and privacy homomorphisms[J]. *Foundations of secure computation*, 1978, 4(11): 169-180.

[7] Elgamal T. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms[J]. *IEEE Transactions on Information Theory*, 1985, 31(4): 469-472.

[8] Paillier P. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes[C]. *The 17th international conference on Theory and application of cryptographic techniques*, 1999: 223-238.

[9] Gentry C. Fully Homomorphic Encryption Using Ideal Lattices[C]. *The forty-first annual ACM symposium on Theory of computing*, 2009: 169-178.

[10] Van Dijk M, Gentry C, Halevi S, et al. Fully homomorphic encryption over the integers[C]. *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2010: 24-43.

[11] Gentry C, Sahai A, Waters B. Homomorphic Encryption from Learning with Errors: Conceptually-Simpler, Asymptotically-Faster, Attribute-Based[C]. *Annual Cryptology Conference*, 2013: 75-92.

[12] Bos J W, Lauter K, Loftus J, et al. Improved Security for a Ring-Based Fully Homomorphic Encryption Scheme[C]. *The 14th IMA International Conference on Cryptography and Coding - Volume 8308*, 2013: 45-64.

[13] López-Alt A, Tromer E, Vaikuntanathan V. On-the-Fly Multiparty Computation on the Cloud via Multikey Fully Homomorphic Encryption[C]. *The forty-fourth annual ACM symposium on Theory of computing*, 2012: 1219-1234.

[14] Stehlé D, Steinfeld R. Making NTRU as Secure as Worst-Case Problems over Ideal Lattices[C]. *The 30th Annual international conference on Theory and applications of cryptographic techniques*: *advances in cryptology*, 2011: 27-47.

[15] Barni M, Orlandi C, Piva A. A Privacy-Preserving Protocol for Neural-Network-Based Computation[C]. *The 8th workshop on Multimedia and security*, 2006: 146-151.

[16] Yao A C. Protocols for Secure Computations[C]. *23rd Annual Symposium on Foundations of Computer Science*, 2008: 160-164.

[17] Orlandi C, Piva A, Barni M. Oblivious Neural Network Computing via Homomorphic Encryption[J]. *EURASIP Journal on Information Security*, 2007, 2007(1): 037343.

[18] Damgård I, Jurik M. A Generalisation, a Simplification and some Applications of Paillier's Probabilistic Public-Key System[C]. *The 4th International Workshop on Practice and Theory in Public Key Cryptography*: *Public Key Cryptography*, 2001: 119-136.

[19] Mohassel P, Zhang Y P. SecureML: A System for Scalable Privacy-Preserving Machine Learning[C]. *2017 IEEE Symposium on Security and Privacy*, 2017: 19-38.

[20] Liu J, Juuti M, Lu Y, et al. Oblivious Neural Network Predictions via MiniONN Transformations[C]. *The 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017: 619-631.

[21] Chen Z G, Song X X, Zhang Y H. A Fully Homomorphic Encryption Scheme Based on Binary-LWE and Analysis of Security Parameters[J]. *Journal of Sichuan University* (*Engineering Science Edition*), 2015, 47(2): 75-81.
(陈智罡, 宋新霞, 张延红. 基于Binary-LWE噪音控制优化的全同态加密方案与安全参数分析[J]. *四川大学学报(工程科学版)*, 2015, 47(2): 75-81.)

[22] Hesamifard E, Takabi H, Ghasemi M. CryptoDL: Deep Neural Networks over Encrypted Data[EB/OL]. 2017: arXiv: 1711.05189. https://arxiv.org/abs/1711.05189.

杨涛 于 2015 年在美国克拉克森大学计算机专业获得工程科学博士学位。现任浙江工商大学副教授。研究领域为人工智能安全、计算机视觉。研究兴趣包括: 对抗性训练、隐私计算。Email: yangt@zjgsu.edu.cn

董建锋 于 2018 年在浙江大学计算机科学与技术专业获得博士学位。现任浙江工商大学计算机科学与技术学院研究员。研究领域为多媒体理解与检索。