동형암호를 적용한 CNN 추론을 위한 ReLU 함수 근사에 대한 연구

주유연[1], 남기빈[1], 이동주[1],백윤흥[1]
[1]서울대학교 전기·정보공학부, 서울대학교 반도체공동연구소

{yyjoo, kvnam}@sor.snu.ac.kr, ehdwn95713@naver.com, ypaek@snu.ac.kr

A Study on Approximation Methods for a ReLU Function in Homomorphic Encrypted CNN Inference

You-yeon Joo[1], Kevin Nam[1], Dong-ju Lee[1] , Yun-heung Paek[1]
[1]Dept. of Electrical and Computer Engineering and Inter-University Semiconductor Research Center(ISRC), Seoul National University

요 약

As deep learning has become an essential part of human lives, the requirement for Deep Learning as a Service (DLaaS) is growing. Since using remote cloud servers induces privacy concerns for users, a Fully Homomorphic Encryption (FHE) arises to protect users' sensitive data from a malicious attack in the cloud environment. However, the FHE cannot support several computations, including the most popular activation function, Rectified Linear Unit (ReLU). This paper analyzes several polynomial approximation methods for ReLU to utilize FHE in DLaaS.

1. Introduction

Deep learning has become an essential part of human lives nowadays. In general, the deeper the model architecture and the more training data, the higher the model accuracy. However, collecting big-data or training such a huge model would be limited to training individual users' local computers. For these reasons, several cloud service vendors provide deep learning as a service (DLaaS) for users to readily utilize the computational resources of vendors to build applicable deep learning models.

DLaaS is worth leveraging high-performance cloud environments' resources, however, it gives rise to fundamental concerns about the privacy of individual users' data. In DLaaS, each user queries his data and receives the inference result of the requested model. The users' data is encrypted when being sent through the network; however, it should be decrypted in the cloud environment to run the intended service, for instance, model inference in DLaaS, which implies that users' data is entirely revealed in the cloud and exposed to malicious insider threats. Several techniques have appeared to protect users' data from such threats, called privacy-preserving techniques.

Fully Homomorphic Encryption (FHE) is one of the promising privacy-preserving techniques in that it enables mathematical operations between encrypted data, does not require decryption, and protects data from aggressive attempts to access it. FHE has a basis on post-quantum cryptographic hardness, which denotes that even quantum computers cannot collapse the cryptosystem. The cryptographic keys are only stored in the user's device; therefore, users' private data keeps encrypted throughout the whole process of DLaaS so that the user can only decrypt processed data which offers a fundamental solution for privacy concerns raised when using traditional cryptography that requires decryption in the remote server.

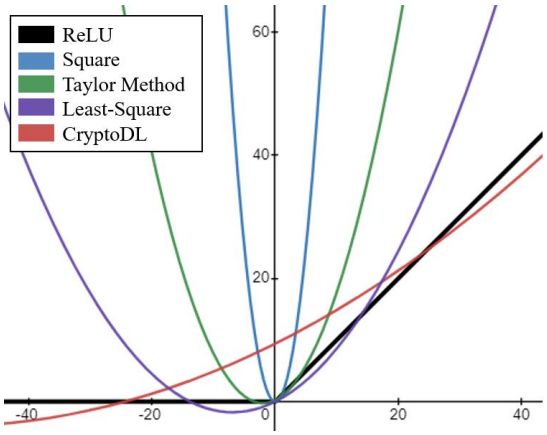
Meanwhile, FHE naturally provides just two primitive operations, addition and multiplication. It cannot compute activation functions in deep learning for non-linearity of the model, such as ReLU-like functions and comparison. Thus, previous works approximate non-linear functions into polynomial functions, which induces computational error versus original activation functions and causes an accuracy drop. In this paper, we analyze several approximation methods for Rectified Linear Unit (ReLU), a one of famous activation function in Convolutional Neural Network (CNN), and compare each in the concept of accuracy.

2. Approximation Methods for ReLU

ReLU is a function that outputs the bigger one between zero and a given number. It does not consist of addition or multiplication but comparison, so it cannot be computed naturally in FHE. There are two generally adopted approaches to approximating ReLU: One is by mimicking the behavior of

ReLU, and the other is by approximating a part of ReLU.

2-1. Mimic-based Methods



(Figure 1) Polynomial Approximation of ReLU

Once the first homomorphic encrypted neural network was proposed, much research conducts on the more precise polynomial approximation of ReLU. CryptoNets[1] initialized research on homomorphic encrypted neural networks, which replaced ReLU with a square function in a simple CNN model. It trained 9-layers CNN model that adopts a square function as an activation function, applied at two parts in the total model, after convolutional and fully-connected layers. The other widely adopted polynomial approximation methods are least square and Taylor expansion. The least-square-based approximation selects a set of points from ReLU and gives these as inputs of an approximated function. However, it is challenging to choose meaningful points to mimic ReLU because it is an infinitely ascending function in fields of positive numbers. For the Taylor series-based method, even if it can precisely express values in the small interval, values out of the interval especially negative numbers, would be divergent. These two methods are simple to approximate a given function, however, they are not suitable to approximate ReLU since they only can mimic the behavior of ReLU in a specific interval.

CryptoDL[2] tried to approximate ReLU from simulating the structure of derivation of the ReLU. The derivation of the ReLU is 0 in negative and 1 in positive, which is a step function. They considered it has a structure like sigmoid function in the large intervals, thus they used the integral of the approximated sigmoid function as their activation function. SecureDL[3] generates the well-approximated low-order polynomial using the least square approximation and for given error bound ε, running iterative verifying algorithm to construct the low-degree polynomial function.

2-2. Approximating a part of ReLU-based Methods

[4] approximates ReLU by approximating sign function using a composition of minimax approximate polynomials of small degree. A minimax composite polynomial $p_k \circ p_{k-1} \circ \ldots \circ p_1$ on $[-b,a] \cup [a,b]$ for $\{d_i\}_{1 \leq i \leq k}$ of sign of x satisfies two properties: (1) p_1 is the minimax approximate polynomial of degree at most d_1 on D (2) For $2 \leqq i \leqq k$, p_i is the minimax approximate polynomial of degree at most di on $p_{i-1} \circ p_{i-2} \circ \ldots \circ p_1(D)$. As ReLU can be expressed with sign function, $\frac{x+x*sign(x)}{2}$, they replaced sign function with proposed approximated function. Though this method produces more precise approximate value, it composes two functions so that the expected multiplicative depth of approximated function is much larger than other methods. [5] proposed an algorithm to compute the Max function using approximated square root function between two values. Given two variables, a and b, it outputs the bigger one what the exact ReLU function does. However, this algorithm is validated for only inputs in the range [0, 1). The input values of the activation function in CNN usually get out of this range, which is unsuitable for CNN.

3. Evaluation

We evaluate the accuracy of each approximation method that applied CNN inference on the MNIST dataset. For a fair comparison, our backbone model consists of only one activation throughout the model and we trained the backbone model with the ReLU function and replaced it with each approximated function at inference time. Note that we normalized input data to scale down [0, 1], and the approximation range of some functions is also narrowed down. The model architecture used is follows:

Convolution – Activation – Fully Connected – Softmax

The accuracy of the baseline model (ReLU activation function) is 96.60%. The method proposed in [4] shows the highest accuracy in Table 1 and requires a much higher degree of a polynomial, for example, 29. However, even if other methods approximate ReLU with such a high degree, they cannot achieve such high accuracy. Also, if we implement ReLU with the method in [4], we need a much more multiplicative depth of homomorphically encrypted ciphertext to compute complicated computations.

Table 1. The Accuracy of the CNN on MNIST using different Approximation Methods

Method	Accuracy
ReLU	96.60%
Square	72.04%
Least-square-error	58.27%
Taylor Series	42.54%
[2]	73.73%
[4] small	95.53%
[4] large	96.60%

4. Conclusion

In this paper, we analyze several approximation methods for Rectified Linear Unit (ReLU) and compare each in the concept of accuracy. There is a tradeoff between the accuracy and the latency of approximated ReLU applied FHE model. The way to choose a better approximation method, the users should consider the size of the trained model and dataset, and the impact of model accuracy.

5. ACKNOWLEDGEMENT

This work was supported by the BK21 FOUR program of the Education and Research Program for Future ICT Pioneers, Seoul National University in 2023. This work was supported by Inter-University Semiconductor Research Center (ISRC). This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2023-2020-0-01602) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation) This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. 2022-0-00516, Derivation of a Differential Privacy Concept Applicable to National Statistics Data While Guaranteeing the Utility of Statistical Analysis)

참고문헌

[1] Gilad-Bachrach, Ran, et al. “Cryptonets: Applying Neural Networks to Encrypted Data with High Throughput and Accuracy”, International Conference on Machine Learning, 2016.

[2] Ehsan Hesamifard, et al. “CryptoDL: Deep Neural Networks over Encrypted Data”, arXiv, 2017.

[3] Xu, Guowen, et al. “Secure and Verifiable Inference in Deep Neural Networks”, Annual Computer Security Applications Conference, 2020.

[4] Lee, Junghyun, et al. “Precise Approximation of Convolutional Neural Networks for Homomorphically Encrypted Data”, arXiv, 2021.

[5] Jung Hee Cheon, et al. “Numerical Method for Comparison on Homomorphically Encrypted Numbers”, ASIACRYPT 2019, Lecture Notes in Computer Science(), vol 11922, 2019.