Efficient homomorphic multiplication of ternary weight matrix and data vector by number-theoretic transform of standard unit vectors

Min-Wook Jeong, Jina Choi, Byoungwoo Yoon, Jongho Shin

LG Electronics

minwook.jeong@lge.com, jina11.choi@lge.com, byoungwoo.yoon@lge.com, jongho0.shin@lge.com

표준 단위 벡터의 정수론적 변환을 통한 터너리 가중치 행렬과 데이터 벡터간의 효율적 동형 곱셈

정민욱, 최진아, 윤병우, 신종호
LG 전자

Abstract

We present an efficient method for multiplication between a ternary weight matrix and a homomorphically encrypted data vector. We rearrange the homomorphic multiplication process into additions and subtractions of the data vector rotations and subsequent elementwise multiplication with the standard unit vector via number-theoretic transform (NTT). This results in both reduced number of costly homomorphic multiplications and increased speed of the multiplication itself. Further performance improvement is attained when multiple products computed by the proposed method are eventually added or subtracted. Finally, our method does not suffer from multiplication level dropping of the encrypted data. This can lead to reduction in the number of bootstrappings required, particularly when the proposed method is embedded in to more complex systems involving multi-layered computations.

Ⅰ. Introduction

Privacy-preserving large language models (PP-LLM) enable all the services of large language models (LLM) without exposing any content of user data to the server. Fully homomorphic encryption (FHE), a technology that performs arithmetic operations directly on encrypted data [1], is considered an ideal tool for implementation of PP-LLM. However, huge computational burden, particularly with multiplication, prevents wide adoption of FHE into real world applications [2].

Transformer is an integral part of state-of-the-art LLMs. Multiplication of a weight matrix and a data vector accounts for large portion of the computation within a transformer.

A study [3] shows that the accuracy of LLM suffers little degradation if we quantize its weights to a ternary value in the set $\{-1,0,1\}$. This results in a big improvement in latency by reducing the number of multiplications.

As multiplication is even more costly in homomorphic computation, this finding is not only useful for FHE implementation of the transformer but also implies the possibility of faster PP-LLM.

In this work we present an efficient method for multiplication between a ternary weight matrix and an encrypted data vector. The proposed method reduces the number of homomorphic multiplications and increases the speed of multiplication itself, leading to improved latency for overall inference of the transformer and PP-LLM. The improvement is achieved by rearrangement of the multiplication process that first generates intermediate values, obtained by additions and subtractions of the data vector rotations, and then multiply the intermediate values elementwise with the standard unit vector.

Further performance boost is attained when multiple products computed by the proposed method are later added or subtracted. Finally, our method does not drop the multiplication level of the product and potentially decreases the number of bootstrappings in a more complex computation scenario.

Ⅱ. Method

We consider a plaintext weight matrix $A \in \{-1, 0, 1\}^{N \times N}$ and a CKKS-encrypted [4] data vector $b \in \mathbb{R}^N$.

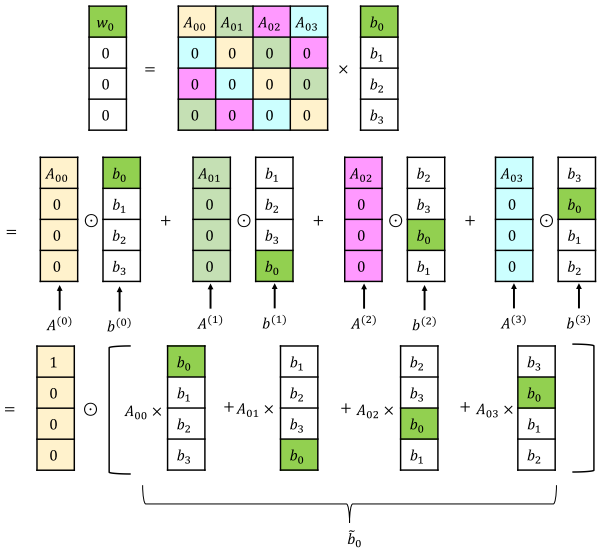

Figure 1. Computation of $(Ab)_0$ with the proposed method

In CKKS, the multiplication Ab is computed by $\sum_{k=0}^{N-1} A^{(k)} \odot b^{(k)}$, where $A^{(k)}$ is the kth upper diagonal vector of A, $b^{(k)}$ is the left rotation of b by k slots, and $\odot$ denotes the elementwise multiplication.

Harnessing the ternary nature of the elements of A, intermediate values $\tilde{b}_i := \sum_{k=0}^{N-1} A_i^{(k)} b^{(k)}$ can be computed only with sums and differences of $b^{(k)}$s, the rotations of b. As the ith element of $\tilde{b}_i$ is equal to the ith element of Ab, we have

$$Ab = \sum_{i=0}^{N-1} e_i \odot \tilde{b}_i = \sum_{i=0}^{N-1} e_i \odot (Ab)_i \qquad (1)$$

Here, e_i is the ith standard unit vector and $(Ab)_i$ denotes the ith element of Ab. The procedure is illustrated in Figure 1.

The computation of $e_i \odot \tilde{b}_i$ in (1) with CKKS involves elementwise multiplication of the NTTs of e_i and b_i, denoted by $\mathcal{E}_i$ and $\tilde{\mathcal{B}}_i$ respectively. Unlike in ordinary CKKS multiplication, e_i is a known value and $\mathcal{E}_i$ can be computed in advance, resulting in improved multiplication latency. Therefore, we attain computational efficiency by replacing ordinary CKKS multiplications with less time-consuming multiplication with e_is.

Further performance improvement is achieved when products from several multiplications of the type in (1) are later added/subtracted. For a concrete example, we consider the computation of $Ab + Cd + Gh$, where A, C and D are N-by-N ternary weight matrices and b, d and h are encrypted data vector in $\mathbb{R}^N$. We have

$$Ab + Cd + Gh = \sum_{i=0}^{N-1} e_i \odot \tilde{b}_i + e_i \odot \tilde{d}_i + e_i \odot \tilde{h}_i$$

$$= \sum_{i=0}^{N-1} e_i \odot (\tilde{b}_i + \tilde{d}_i + \tilde{h}_i) \qquad (2)$$

In this case, intermediate values $\tilde{b}_i, \tilde{d}_i$ and $\tilde{h}_i$ in (2) can be added before multiplication with e_i and the number of multiplications can be reduced to one-third.

Finally, the proposed method does not decrease the multiplication level of encrypted data after multiplication. The value of e_i is known in advance and $\mathcal{E}_i$ can be scaled ahead of multiplication. This eliminates the need for post-multiplication rescaling and decreases the number of bootstrappings, particularly when the method is adopted in complex systems with multiple layers of computations.

Ⅲ. Conclusion

Our work was motivated by a finding that quantization of weight matrices in LLM significantly decreases inference latency at little cost of accuracy. The speed improvement mainly comes from reducing the number of multiplications.

As multiplication is even more costly in homomorphic computation, we leveraged this finding to propose an efficient method for homomorphic multiplication between a plaintext ternary weight matrix and an encrypted data vector.

The merits of the proposed method imply that its adoption in complex models like transformer could decrease the overall inference time of the system, which in turn could help with wider deployment of services like PP-LLM.

REFERENCES

[1] Gentry, Craig. A fully homomorphic encryption scheme. Stanford university, 2009.

[2] Pambudi, Doni Setio, et al. "Performance Analysis of Leading Homomorphic Encryption Libraries: A Benchmark Study of SEAL, HElib, OpenFHE, and Lattigo." Proceedings of the 2025 4th International Conference on Cyber Security, Artificial Intelligence and the Digital Economy. 2025.

[3] Ma, Shuming, et al. "The era of 1-bit llms: All large language models are in 1.58 bits." arXiv preprint arXiv:2402.17764 1.4 (2024).

[4] Cheon, Jung Hee, et al. "Homomorphic encryption for arithmetic of approximate numbers." International conference on the theory and application of cryptology and information security. Cham: Springer International Publishing, 2017.