Performance of Hierarchical Transforms in Homomorphic Encryption:

A case study on Logistic Regression inference

Pedro Geraldo M. R. Alves, Jheyne N. Ortiz, Diego F. Aranha

Zama, University of Campinas, Aarhus University

Objectives and Contributions

This work brings some light to the discussion about the suitability of the Discrete Galois Transform (DGT) to replace the Number-Theoretic Transform (NTT) to accelerate polynomial multiplications on RLWE-based homomorphic encryption schemes.
We developed similar CUDA implementations of the CKKS cryptosystem, differing only in the underlying transform and related data structure. Also, we ran experiments to collect performance metrics in three different contexts:

- Direct comparison between DGT and NTT;
- Impact on CKKS' homomorphic multiplication; and
- Impact on the inference phase of Logistic Regression.

Introduction

Modern homomorphic encryption schemes rely on polynomial arithmetic. In this context, developers need to find efficient handling of costly operations such as polynomial multiplication. The literature has established the suitability of the NTT, a variant of the Discrete Fourier Transform (DFT) that operates over integers, to compute the polynomial multiplication with linear complexity within the transform domain. Nonetheless, some recent works suggest that the DGT may be a better candidate when the target hardware is a CUDA-enabled GPU [1].

CKKS Scheme

CKKS is a leveled homomorphic encryption scheme in which the plaintext domain is composed of complex numbers.
Consider that $\mathcal{C} = \{q_0, q_1, \ldots, q_\ell\}$ is a set of coprime integers, called a RNS basis, $q = \prod_{i=0}^{\ell} q_i$, and $R_q = \mathbb{Z}_q[x]/(x^N + 1)$ for N an integer. If $s \in R_{\mathcal{C}}$, then $\exists S \in R_q$ such that $s := \{[S]_{q_0}, [S]_{q_1}, \ldots, [S]_{q_\ell}\}$.
A CKKS ciphertext is a pair of elements in $R_{\mathcal{C}}$. Its relinearization procedure switches the ciphertext representation between $R^2_{\mathcal{C}}$ and $R^2_{\mathcal{C}+\mathcal{D}}$, for a secondary basis $\mathcal{D}$ coprime to $\mathcal{C}$. These basis conversion steps are done through approximate modulus switching functions called ModUp and ModDown.

Performance Evaluation of Hierarchical Transforms

We conceived two CUDA-based implementations of CKKS using the hierarchical versions of both DGT and NTT. Both implementations follow the same design decisions, except for their basic data type. We also explore an optimization on NTT's version of ModUp which increases arithmetic density per thread. We executed our experiments on a Google cloud instance featuring either an NVIDIA Tesla V100 or Tesla A100 GPUs.

Figure 1:Ratio of DGT and NTT execution times for different polynomial degrees and RNS basis' sizes.

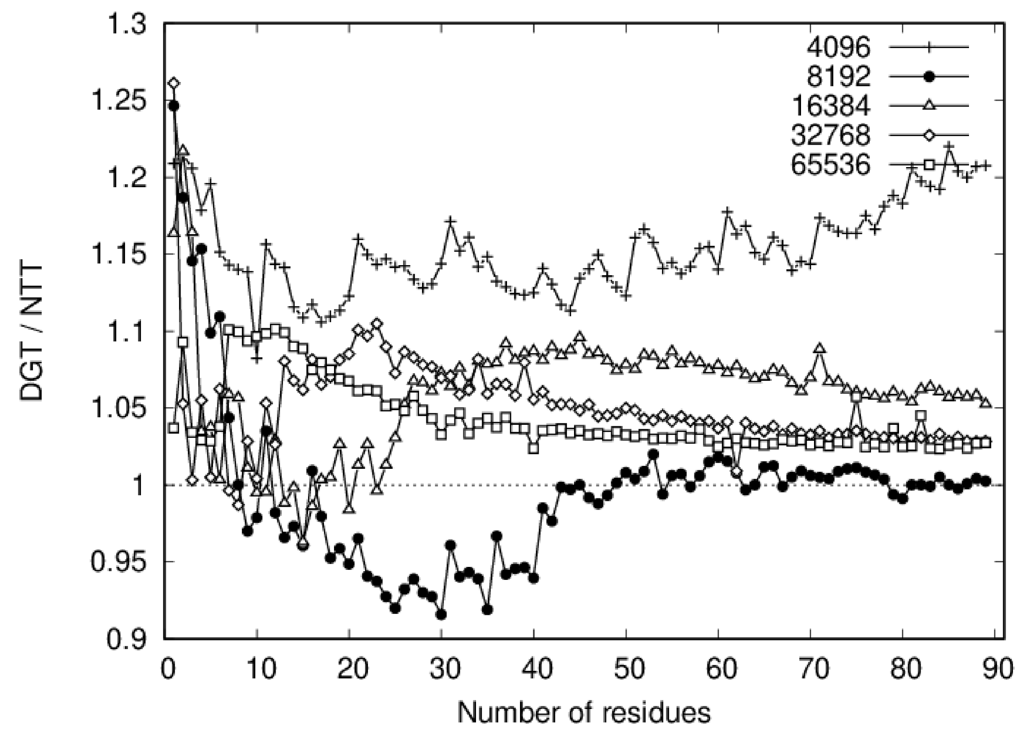
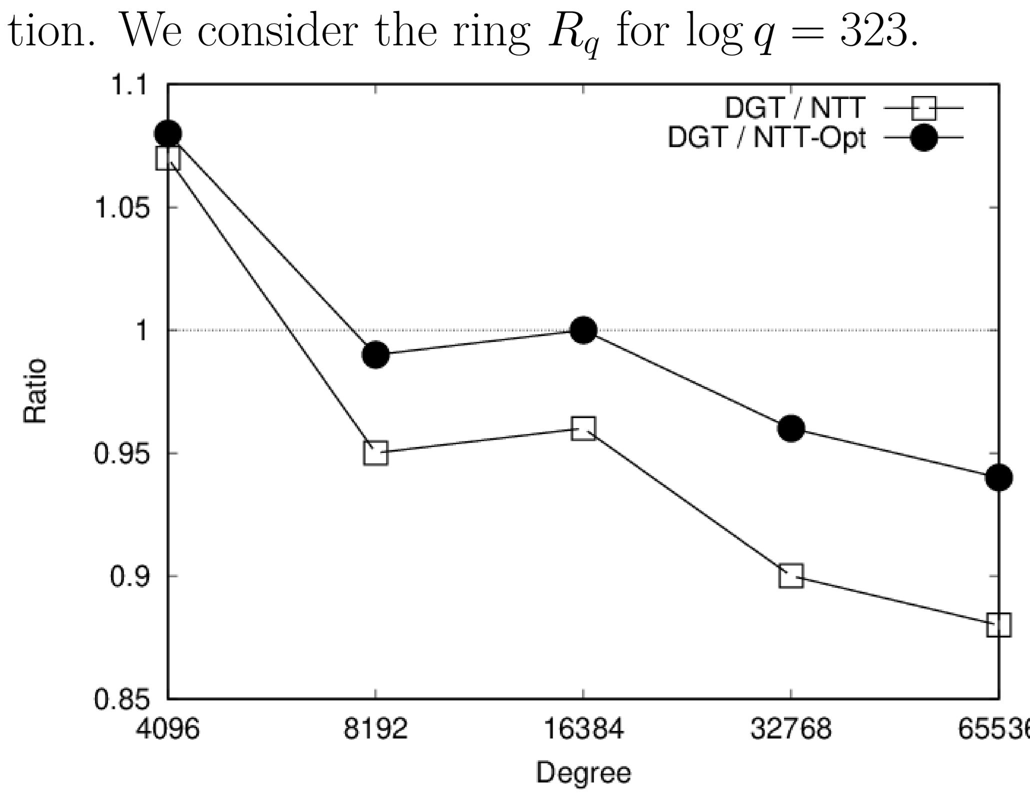


Figure 2:Comparison of CKKS' homomorphic multiplication running with DGT and two implementations of NTT: a schoolbook and an optimized implementation. We consider the ring R_q for $\log q = 323$.



Homomorphic Logistic Regression

Logistic Regression is a learning algorithm widely used to solve classification problemss. We follow the approach of similar works in the literature and apply learning algorithms to the MNIST dataset. The images on the MNIST dataset represent handwritten digits $(0 - 9)$ that were split into a training and a test set with 60 000 and 10 000 records, respectively. We trained a model using a Python script applying the `scikit-learn` library's LR implementation.

Data Structure

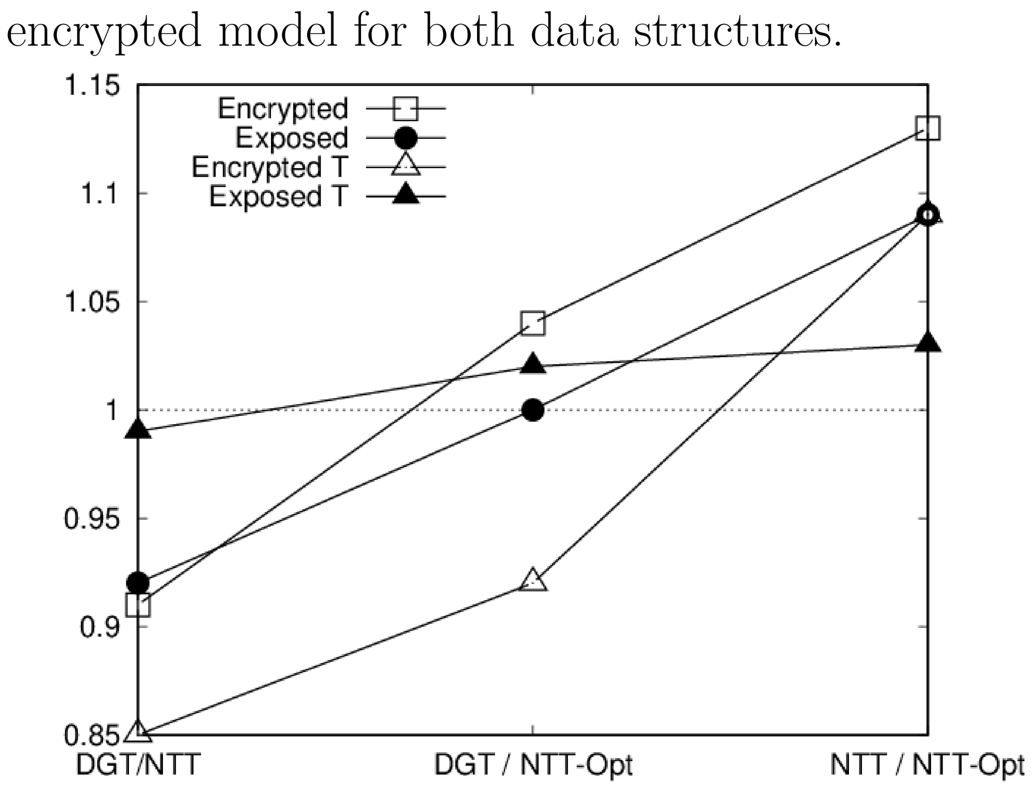
We evaluated two ciphertext designs:

- **Direct**: Each ciphertext stores all the columns related to a particular digit. In this case, we have as many ciphertexts as images.
- **Transposed**: A ciphertext stores one pixel of each image in the dataset. More ciphertext slots may be needed compared to the direct approach.

The direct version requires smaller CKKS parameters and consumes less memory. However, it depends on more expensive operations to compute the inner product within the inference phase.

Result

Figure 3:Comparison of the Logistic Regression inference performing using the DGT, the NTT, and the optimized NTT. We consider the encrypted and not encrypted model for both data structures.



Conclusion

The data structure related to the DGT naturally benefits the performance of the homomorphic multiplication over the NTT since it increases the arithmetic density of ModUp. However, the NTT may be optimized to achieve similar performance. This result goes toward Badawi et al.'s claim that the DGT better fits CUDA's processing paradigm.

References

[1] Badawi et al.
High-performance FV somewhat homomorphic encryption on gpus: An implementation using CUDA.
IACR Trans. Cryptogr. Hardw. Embed. Syst., 2018(2):70–95, 2018.