6.2050 Final Project Proposal
Team: Ruth Lu (ruthluvu)*, Shreya Chaudhary (shreyach)*
Working Title: FPGA Accelerator for Fully Homomorphic Encryption for Inference Models

Abstract Fully homomorphic encryption (FHE) is a post-quantum-secure algorithm allowing computation on encrypted data such that an adversarial algorithm can compute an output without learning any information about the input or output [2, 3]. FHE is an especially powerful algorithm for ensuring data privacy for inference models. Users can send encrypted data to an untrusted party's model for training or predictions, ensuring privacy for sensitive information like medical data. Despite these benefits, FHE is not used in practice due to its slowness: it involves many matrix computations for encryption, decryption, and operations on encrypted data. Specialized hardware, like that of an FPGA, could be used to accelerate these specific and highly parallelizable operations [1].

Consequently, we aim to build an accelerator for running machine learning inference models with fully homomorphic encryption (FHE). Specifically, we aim to at least run secure inference with a linear regression model on the FPGA. Our stretch goal is to be able to run a model with high complexity, like a small neural net, to do inference using the encrypted data. In building this project, our first priority will be correctness of FHE and the operations that can be executed on it. Secondly, we want it to be fast, ideally faster than a CPU implementation written in Python.

We will first create CPU code of FHE linear regression as a baseline. The actual FPGA implementation will require modules for generating random numbers, modular arithmetic, and matrix multiplication. Next we'll need to implement that actual algorithm. We plan to implement FHE with bootstrapping so it can work for an arbitrary-depth computation. Consequently, we will need to implement four functions: KeyGen, Encrypt, Decrypt, and Eval. KeyGen generates a random number and Encrypt and Decrypt will use the learning with errors (LWE) lattice-based encryption scheme to encrypt and decrypt data [4]. Eval will consist of evaluating both additions and multiplications (with bootstrapping). Finally, we will use the FHE module to add encryption for inference for linear regression and hopefully a simple neural network.

Division of Work We plan to both collaborate on building the CPU code. Ruth will focus on optimizing matrix multiplication and Shreya will focus on building modular arithmetic modules to implement LWE. From these building blocks, we will both work on implementing the main functions (KeyGen, Encrypt, Decrypt, and Eval for Addition and Multiplication). Similarly, once FHE is implemented, we will both work on implementing the linear regression and maybe the neural network.

References

[1] R. Agrawal, L. de Castro, G. Yang, C. Juvekar, R. Yazicigil, A. Chandrakasan, V. Vaikuntanathan, and A. Joshi. Fab: An fpga-based accelerator for bootstrappable fully homomorphic encryption, 2022.

[2] C. Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009.

[3] C. Gentry, A. Sahai, and B. Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *Advances in Cryptology–CRYPTO 2013: 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, pages 75–92, Santa Barbara, CA, USA, 2013. Springer.

[4] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM*, 2009.