ANL/NSE-22/54

Homomorphic Encryption for Machine Learning and Artificial Intelligence Applications

Secure Application of Machine Learning and Artificial Intelligence using Homomorphic Encryption

Nuclear Science and Engineering Division

ANL/NSE-22/54

Homomorphic Encryption for Machine Learning and Artificial Intelligence Applications

Secure Application of Machine Learning and Artificial Intelligence using Homomorphic Encryption

prepared by
David Arnold[1,2], Jafar Saniie[2], Alexander Heifetz[1]

[1]Nuclear Science Engineering Division, Argonne National Laboratory, Lemont, IL
[2]Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL

August 31, 2022

Table of Contents

List of Figures

Abstract

Third-party and expert analysis is a cost-effective solution for solving specialized problems or processing large datasets related to reactor structural health monitoring and nondestructive evaluation. However, when handling proprietary information, third-party and expert analysts pose a privacy risk. To address this challenge, Homomorphic Encryption (HE) permits arithmetic operations on encrypted data without exposing the underlying data. Implementations of Machine Learning (ML) and Artificial Intelligence (AI) algorithms using HE greatly enhances the capabilities of third-party analysts while maintaining a low security risk. This paper details current success in applying Principal Component Analysis (PCA) and Fully Connected Neural Networks (NN) using the Microsoft SEAL implementation of the popular CKKS Fully Homomorphic Encryption (FHE) algorithm. The MNIST Handwritten Dataset is analyzed as a proof-of-concept demonstration of the implementations.

1. Introduction

Third-party and expert analysis is a cost-effective solution for solving specialized problems or processing large datasets related to reactor structural health monitoring [1-3] and nondestructive evaluation [4-7]. However, when handling proprietary information, third-party and expert analysts pose a privacy risk. To address this challenge, Homomorphic Encryption (HE) permits arithmetic operations on encrypted data without exposing the underlying data. As a result of their ability to modify the underlying data of a given ciphertext, HE cryptosystems are a powerful tool for maintaining privacy. During operation, the user or corporate data is encrypted using the HE algorithm and is transferred to the data processor. The processor then completes their desired operations on the ciphertext without revealing the sensitive data. An example of this exchange can be seen in Figure 1, where a client transmits their encrypted data to a third-party server. Potential applications of HE cryptosystems include the application of a filter on a user image, analysis of network traffic data from a government network, and cyberattack detection for critical infrastructure. For this report, we will apply several Machine Learning (ML) and Artificial Intelligence (AI) models on the MNIST Handwritten dataset for model accuracy and analysis of storage size and timing considerations. Future work will be focused on applying HE cryptosystems for defect detection on thermal imaging data.

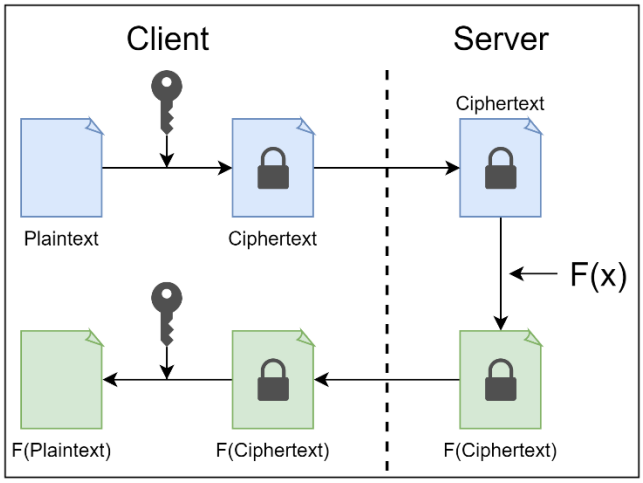

Figure 1 - A common application of Homomorphic Encryption technology. The client will encrypt their data using the HE algorithm and will transmit it to the server for processing. The server will apply their chosen operations and send the resulting ciphertext back to the client for decryption.

Homomorphic Encryption (HE) is a family of public key encryption algorithms that permit operations on ciphertext without exposing the underlying data. Different levels of HE exists, with initial generations and partially homomorphic cryptosystems providing support for limited operations, arbitrary circuits, and circuits of bounded depth. Recently, Fully Homomorphic Encryption (FHE) cryptosystems were introduced that permit arbitrary circuits of unbounded depth. For our purposes, we selected the Microsoft SEAL implementation of the popular CKKS

algorithm [8]. This algorithm was introduced by Cheon et al. to support approximate addition and multiplication on ciphertexts and provides a rescaling method for managing the scale of the plaintext during operations [9]. Due to these capabilities, the selected Microsoft SEAL implementation was a clear choice for constructing our Machine Learning (ML) and Artificial Intelligence (AI) algorithms. It is important to note that our implementation will complete training of the models on the plaintext data before transferring the weights to the HE implementation for training. However, there are two primary challenges when working with FHE cryptosystems, namely the large increase in size when encrypting the plaintext into ciphertext and the inability to directly compute comparisons and division operations.

The remainder of this report will discuss our efforts towards implementing several Machine Learning and Artificial Intelligence models for Homomorphic Encryption. First, we will discuss challenges and solutions when working with HE. Next, we will discuss the ML and AI models that will be constructed for our application. Finally, we will discuss our results when building these models in plaintext and using the HE cryptosystem.

2. Homomorphic Encryption Challenges and Solutions

During initialization, the CKKS algorithm requires a poly-modulus degree, which sets the ciphertext space for operations, along with a set of coefficients for resizing. Whenever a multiplication operation is completed, the size of the ciphertext grows linearly and must be rescaled and re-linearized before completing any new operations [10]. This prevents our ciphertext from outgrowing the allotted space while also limiting the potential error size. Further, since this procedure uses one of the pre-selected coefficients, there is a hard limit on the number of operations depending on how many coefficients and the size of the poly-modulus degree. For instance, for a set of 2 operations we require 4 coefficients with a poly-modulus degree of 8192 while 4 operations require 6 coefficients and an even larger poly-modulus degree of 16384. When stored, 10 standardized images are stored as 68.4 Kilobytes whereas the encrypted data is stored as 2.45 Gigabytes for a poly-modulus degree of 8192 and 7.73 Gigabytes for a poly-modulus degree of 16384. In addition to increasing the storage size for our ciphertext, this also increases the memory space used by plaintext values that we intend on multiplying or adding to our ciphertext. A comparison between the storage size for each poly-modulus degree can be seen in Figure 2. Unfortunately, we do not present a solution for this challenge in this paper but will explore this challenge in future endeavors.

Type	Size	Size Ratio
Plaintext	68.4 KB	1
8192 poly-modulus degree	2.45 GB	35,818
16384 poly-modulus degree	7.73 GB	113,011

Figure 2 - Storage size comparison for 10 standardized images from the MNIST Handwritten dataset. The compared files were for the plaintext, the images encrypted with a poly-modulus degree of 8192, and images encrypted with a poly-modulus degree of 16384. As shown FHE greatly increases the required storage size when compared to the plaintext.

Since we are limited to only addition and multiplication operations, we are limited in the types of functions that we can directly implement using the HE toolkits. For instance, the ReLU, SoftMax, and sigmoid functions require comparisons and division operations. To work around this challenge, we can use polynomial approximation to develop approximate functions for these common activation functions [11-15]. During polynomial approximation, we will attempt to find the coefficients for the order of the equation that we desire. An example equation with order 2 can be seen in Equation 1. Determining the required coefficients is accomplished by taking the difference between the desired function and the approximate function over a given limit, as shown in Equation 2. By minimizing Equation 2, we achieve the necessary coefficients. For a ReLU function over the range -20 to +30, we receive Equation 3. A comparison between the ReLU function and its polynomial approximation is displayed in Figure **3**.

$$p_n(x) = a_0 + a_1 x + a_2 x^2 \tag{1}$$

$$g(x) = \int_{-L}^{L} (f(x) - p_n(x))^2 dx \tag{2}$$

$$p(x) = 2.593 + 0.4752x + 0.0173x^2 \tag{3}$$

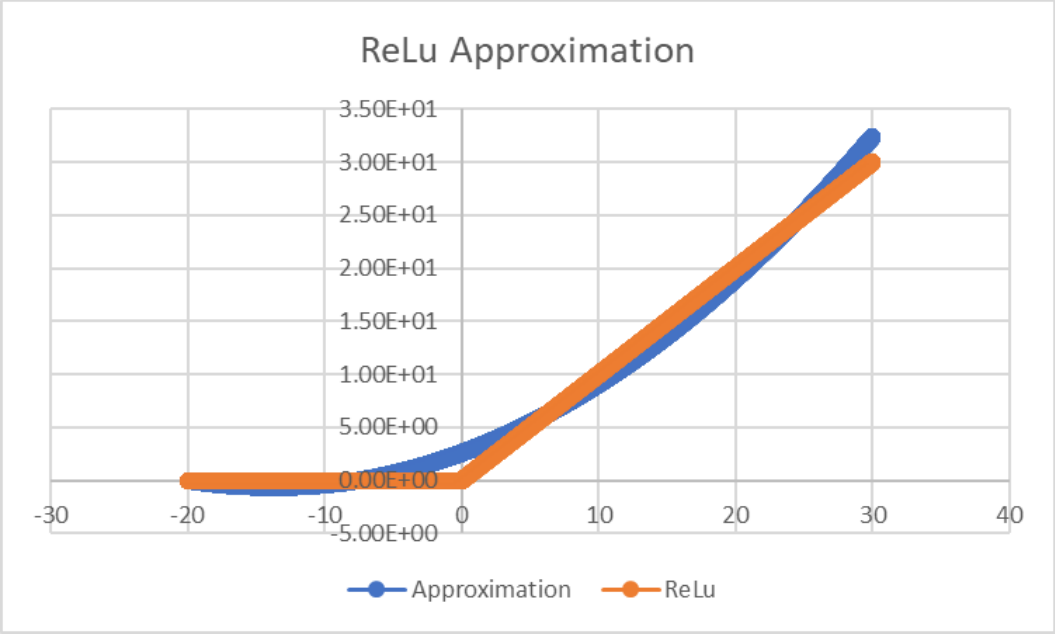

Figure 3 - Comparison between the ReLU activation function and its polynomial approximation.

3. Machine Learning and Artificial Intelligence Models

Due to their flexibility in solving a wide range of problems, Machine Learning (ML) and Artificial Intelligence (AI) were selected for implementation using the Homomorphic Encryption (HE) toolkit. As noted, many of these algorithms rely on non-arithmetic operations that require polynomial approximation. First, we decided to examine Fully Connected Neural Networks due to their simplicity and application in other AI models. Shown in Figure 4, a small NN with two layers is displayed. The first hidden layer uses the ReLU activation function and has 100 output nodes while the second layer uses the SoftMax function with 10 output nodes. Polynomial approximation was used for the ReLU function, but since the largest output of the second layer will be the largest value of the SoftMax function, we did not approximate the SoftMax function.

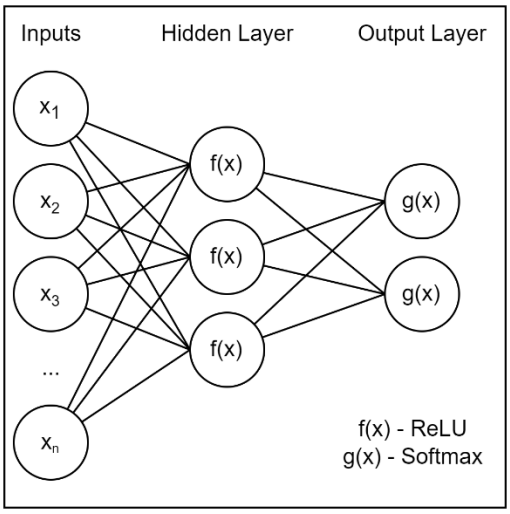

Figure 4 - Architecture of a small Fully Connected Neural Network. For our implementation, the first hidden layer used the ReLU activation function with 100 output nodes while the final layer used the SoftMax function and had 10 output nodes.

Next, we examined Convolutional Neural Networks (CNN) as they are a powerful tool at categorizing and analyzing images. For a simple implementation, we developed our CNN to use two convolutional layers, a max pool layer, and two fully connected layers. This yielded mediocre results and further testing will be conducted before it is converted into a fully HE implementation.

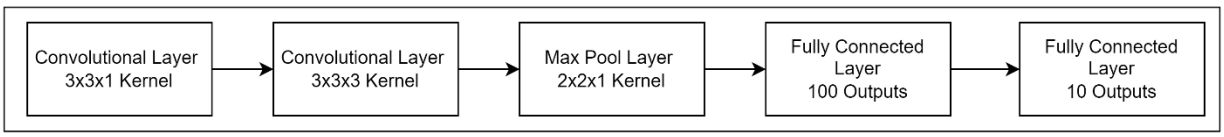

Figure 5 - Architecture of a simple Convolutional Neural Network. This CNN included two convolutional layers, a max pool layer, and two fully connected layers.

Finally, the only Machine Learning model that is currently being considered is Principal Component Analysis (PCA). This unsupervised model allows us to reduce our features based on their contribution. We have successfully implemented the plaintext version and still need to implement the HE version. PCA will be a great tool in decreasing the feature space, and will also be used later during defect detection for thermal imaging.

4. Results

We used the MNIST Handwritten Dataset to validate our plaintext and Homomorphic Encryption models. The MNIST Handwritten Dataset is composed of 28x28 black and white images that represent handwritten versions of the digits 0-9. In terms of our models, we successfully implemented the Fully Connected Neural Network described in Figure 4 and the Convolutional Neural Network described in Figure 5. Additionally, we successfully implemented an Approximated Neural Network that replaced the ReLU function with the polynomial approximation. Further, an additional model that preprocesses the dataset using PCA was also implemented in plaintext but not in HE. Figure 6 presents the accuracy results after training the models on 1000 images and testing the models on 100 images. Based on these results, we still need to conduct further work on different CNN architectures as the current implementation can only achieve an accuracy of 59.1%. The Neural Network performed well with 83.8% accuracy while the Approximate ReLU function decreased the accuracy by around 7.7%, so different orders of approximation will be explored to decrease the change. Finally, the PCA implementation achieved the best result with 85.9% accuracy, which is promising as a method for improving accuracy and decreasing runtime.

Model	Accuracy
Neural Network	**83.8%**
Approximated Neural Network	**76.1%**
Principal Component Analysis + Neural Network	85.9%
Convolutional Neural Network	59.1%

Figure 6 - Testing accuracy when categorizing the MNIST Handwritten dataset. The models included a simple Neural Network, pre-processing using Principal Component Analysis, a Convolutional Neural Network, and an Approximated Neural Network.

While we only implemented the Neural Network using Homomorphic Encryption, we had clear results regarding the large impact on timing that is experienced. While we receive an identical result between the NN and Approximated NN models, the Approximate version took around 150,000 times longer to run on average. Figure 7 presents the average runtime per image when running our different models. For the Approximated model, it took around 5 minutes per prediction whereas the plaintext version only took 2 milliseconds on average.

Model	Time
Plaintext Operations	2.001×10^{-3} seconds
Homomorphic Encryption Operations	295.3 seconds

Figure 7 - Average Timing analysis of the Approximated Neural Network when completed using plaintext operations and when completed using Homomorphic Encryption operations. As shown, the HE implementation takes far longer to complete.

5. Conclusions

Overall, we were successful at exploring the potential use of Machine Learning and Artificial Intelligence within the scope of Homomorphic Encryption. Polynomial Approximation was explored as a potential solution to the limited scope of operations presented by the HE toolkits. Neural Networks, Convolutional Neural Networks, and Principal Component Analysis were implemented in plaintext while an Approximate Neural Network was implemented in HE. Based on our results, the Approximate Neural Network was accurate, but experienced very poor runtime characteristics which need to be analyzed in future work. We plan on exploring the PCA implementation further along with other Machine Learning Models, such as Independent Component Analysis, Sparse Coding, and Exploratory Factor Analysis. These will be applied towards defect detection in thermal imaging.

References

1. X. Huang, J. Saniie, D. Arnold, T. Fang, A. Heifetz, S. Bakhtiari, “Software-Defined Ultrasonic Communication System with OFDM for Secure Video Monitoring,” *IEEE Access*, vol. 10, pp. 47309 – 47321, 2022.
2. A. Heifetz, D. Shribak, X. Huang, B. Wang, J. Saniie, R. Poncirolli, E.R. Koehl, S. Bakhtiari, R.B. Vilim, “Transmission of Images on High-Temperature Nuclear-Grade Metallic Pipe with Ultrasonic Elastic Waves,” *Nuclear Technology*, vol. 207, no. 4, pp. 604-616, 2021.
3. A. Heifetz, D. Shribak, X. Huang, B. Wang, J. Saniie, J. Young, S. Bakhtiari, R. Vilim, “Transmission of Images with Ultrasonic Elastic Shear Waves on a Metallic Pipe using Amplitude Shift Keying Protocol,” *IEEE Transactions in Ultrasonics, Ferroelectrics and Frequency Control*, vol. 67, no. 6, pp. 1192-1200, 2020.
4. X. Zhang, J. Saniie, S. Bakhtiari, A. Heifetz, “Compression of Pulsed Infrared Thermography Data with Unsupervised Learning for Nondestructive Evaluation of Additively Manufactured Metals,” *IEEE Access*, vol. 10, pp. 9094 – 9107 (2022).
5. X. Zhang, J. Saniie, A. Heifetz, “Detection of Defects in Additively Manufactured Stainless Steel 316L with Compact Infrared Camera and Machine Learning Algorithms,” *JOM*, vol. 72, no. 12, pp. 4244-4253, 2020.
6. X. Zhang, J. Saniie, W. Cleary, A. Heifetz, “Quality Control of Additively Manufactured Metallic Structures with Machine Learning of Thermography Images,” *JOM*, vol. 72, no. 12, pp. 4682-4694, 2020.
7. A. Heifetz, D. Shribak, X. Zhang, J. Saniie, Z.L. Fisher, T. Liu, J.G. Sun, T. Elmer, S. Bakhtiari, W. Cleary, “Thermal Tomography 3D Imaging of Additively Manufactured Metallic Structures,” *AIP Advances*, vol. 10, no.10, pp. 105318, 2020.
8. Microsoft Research, *Microsoft Seal*, Redmond, Washington, 2022.
9. J. H. Cheon, A. Kim, M. Kim and Y. Song, "Homomorphic Encryption for Arithmetic of Approximate Numbers," in *International conference on the theory and application of cryptology and information security*, 2017.
10. J. Fan and F. Vercauteren, "Somewhat Practical Fully Homomorphic Encryption," Katholieke Universitat Leuven , 2012.
11. J. Chiang, "On Polynomial Approximation of Activation Function," Cornell University , 2022.
12. N. Dowlin, R. Gilad-Bachrach, K. Laine, K. Lauter, M. Naehrig and J. Wernsing, "CryptoNets: Applying Neural Networks to Encrypted Data with High Throughput and Accuracy," in *International conference on machine learning*, 2016.
13. E. Lee, J.-W. Lee, Y.-S. Kim and J.-S. No, "Optimization of Homomorphic Comparison Algorithm on RNS-CKKS Scheme," *IEEE Access*, no. 10, pp. 26163-26176, 2022.
14. J. Lee, E. Lee, J.-W. Lee, Y. K. Y.-S. Kim and J.-S. Noo, "Precise Approximation of Convolutional Neural Networks for Homomorphically Encrypted Data," Cornell University, 2021.

15. J.-W. Lee, H. Kang, Y. Lee, W. Choi, J. Eom, M. Deryabin, E. Lee, J. Lee, D. Yoo, Y.-S. Kim and J.-S. No, "Privacy-Preserving Machine Learning with Fully Homomorphic Encryption for Deep Neural Network," IEEE Access, no. 10, pp. 30039-30054, 2022.

Nuclear Science and Engineering (NSE) Division
Argonne National Laboratory
9700 South Cass Avenue, Bldg. 208
Argonne, IL 60439

www.anl.gov