Homomorphic Encryption in Machine Learning

Narendra N
Lead Consultant, AI Research
Wipro Limited

***Abstract*—Data driven analysis using Machine Learning (ML) and Deep Learning (DL) has gained a lot of popularity in recent years. The success of these techniques in solving tasks in the field of image processing, computer vision and natural language processing has made these techniques almost indispensable. These techniques are now being explored in the areas of Finance, Medicine etc. Application in these areas pose a specific problem of privacy. Privacy in financial domain might be the hesitancy of a client to share his/her financial information for data analysis. In the medical domain, the patient records are bound by confidentiality. Data encryption could be used to preserve the privacy of such data. However, can we do analysis on encrypted data? Homomorphic Encryption (HE) might be an answer to this question. HE is a form of encryption which allows us to do computation on encrypted data. It is a lattice based cryptographic technique which is gaining popularity in the ML research community to see if it can be incorporated to solve privacy preserving ML tasks.**

HOMOMORPHIC ENCRYPTION[1] is a form of encryption that allows computation on ciphertexts, generating an encrypted result which, when decrypted, matches the result of the operations as if they had been performed on the plaintext. Consider a simple example of adding two numbers a and b. Let Enc and Dec denote the encryption and decryption operations respectively. HE says that $a + b$ is equivalent to $Dec(Enc(a) + Enc(b))$.

HE schemes are specified by four steps:

- **Key Generation**: In cryptography, key generation is the process of generating the public and secret key
- **Encryption**: It is a process used to convert messages m_1 and m_2 into a ciphertexts c_1 and c_2 using a public key.
- **Decryption**: It is the process of converting the ciphertexts c_1 and c_2 back to messages m_1 and m_2 with a secret key
- **Eval**: It is a HE specific operation, which takes ciphertexts as input and outputs a ciphertext corresponding to a functioned plaintext. Eval performs the function $f()$ over the ciphertexts c_1 and c_2 without seeing the messages m_1 and m_2.

The Homomorphic encryption schemes can be categorized into three types:

- **Partially homomorphic encryption** encompasses schemes that support the evaluation of circuits consisting of only one type of operation, e.g., addition or multiplication. The RSA encryption scheme is a good example for the same.
- **Somewhat homomorphic encryption** schemes can evaluate both addition and multiplication but only for a finite operations.
- **Fully homomorphic encryption** (FHE) allows the evaluation of arbitrary operations of unbounded depth and is the strongest notion of homomorphic encryption. This is achieved

through a process called Bootstrapping which allows HE to perform any number of operations.

Homomorphic Encryption in Machine Learning

To understand the applications of HE in Machine Learning, let us consider a simple example of linear regression. Linear Regression allows us to estimate the value of a target variable as a linear combination of the input parameters. Let y be the target variable and $x_1, x_2, x_3, ..., x_n$ be the input parameters. Linear regression estimate of y is given by

$$y = \alpha_1 \times x_1 + \alpha_2 \times x_2 + ... + \alpha_n \times x_n \quad (1)$$

where $\alpha_1, \alpha_2, ..., \alpha_n$ are the learnt weights or coefficients of corresponding input parameters.

Equation 1 involves just multiplications and additions which is supported by HE. Thus, linear regression is a straightforward application of HE. If we now look at neural networks which are the basic building blocks of state of the art Deep Learning architectures, we see that output of each layer of the network is a linear combination of the inputs followed by a non-linear activation function. With non-linearity handled by approximations, HE can be a suitable candidate for privacy preserving ML tasks. An example can be inference of ML models on encrypted data as depicted in Figure 1. CryptoNets [2] was one of the first papers to tackle the inference problem on an image classification task. The encrypted MNIST data was classified by passing it through a Convolutional Neural Network. Several works [3], [4], [5], [6], [7] followed which tackled the problems of Bootstrapping, Training of Neural Networks to name a few.

Limitations of Homomorphic Encryption

Even though a novel solution, HE faces its own set of problems.

- **Time Complexity**: The time taken to perform an operation on encrypted data is much higher compared to plaintext. For example, a simple single layer Neural Network can push out the result within a fraction of a second which is not the case when doing an inference on encrypted data.

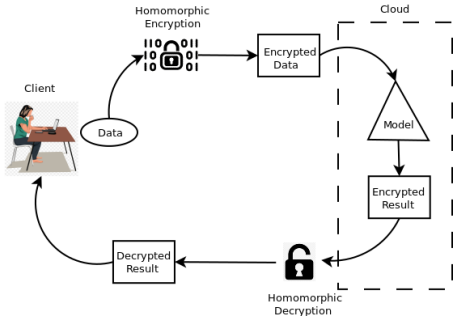

Figure 1. Application of HE in ML: An example

- **Arbitrary Function Evaluation**: HE supports only addition and multiplication. Hence, arbitrary functions have to be approximated to their polynomial counterparts. For example, in a neural network, the sigmoid can be approximated to a third degree polynomial.
- **Number of Computations**: A dense network involves large number of multiplications and additions. With Somewhat Homomorphic Encryption, the number of operations which can be performed is limited and using an FHE is not practically feasible due to the high time complexity of the bootstrapping operation.

Conclusion

The article gives a brief of Homomorphic Encryption and its application to Machine Learning. FHE as mentioned earlier is the ultimate prize one is looking for as this allows users to share data without inhibition. It holds a lot of promise for a plethora of real-world applications. However, the aim is to find a way to solve the limitations of the technology and to make it usable across industries in a larger scale

REFERENCES

1. C. Gentry, *A Fully Homomorphic Encryption Scheme*. PhD thesis, Stanford, CA, USA, 2009. AAI3382729.
2. 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 *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pp. 201–210, JMLR.org, 2016.
3. I. Chillotti, N. Gama, M. Georgieva, and M. Izabachne, "Faster fully homomorphic encryption: Bootstrapping in

less than 0.1 seconds." Cryptology ePrint Archive, Report 2016/870, 2016. https://eprint.iacr.org/2016/870.

4. K. Han, S. Hong, J. H. Cheon, and D. Park, "Logistic regression on homomorphic encrypted data at scale," 2019.
5. H. Chen, I. Chillotti, and Y. Song, "Improved bootstrapping for approximate homomorphic encryption," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 34–54, Springer, 2019.
6. K. Nandakumar, N. Ratha, S. Pankanti, and S. Halevi, "Towards deep neural network training on encrypted data," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 0–0, 2019.
7. R. Xu, J. B. D. Joshi, and C. Li, "Cryptonn: Training neural networks over encrypted data," 2019.