Hybrid Synchronization Approach with Dynamic Weight Allocation for Secure Federated Learning

Abdus Samee
Department of CSE, BUET
1805021@ugrad.cse.buet.ac.bd

Dr. Muhammad Abdullah Adnan
Department of CSE, BUET
adnan@cse.buet.ac.bd

***Abstract*—Federated learning (also called collaborative learning) is a machine learning technique introduced by Google in 2016 that trains a global model at a server via local gradient updates collected from multiple independent training sessions in several clients. Since the local gradient updates could be sensitive data, it becomes imminent to preserve them while training to reduce privacy breaches. Multi-key homomorphic encryption encrypts client updates and ensures efficient gradient aggregation. This poses communication and computation overhead. The training time of a model increases when encrypted local updates are aggregated synchronously. The asynchronous aggregation is likely to reduce training time. Hence, we followed a hybrid approach to training the clients using both synchronous and asynchronous methods. We introduced dynamic weight allocation for each local update during aggregation to mitigate the effect of stale gradients. Our experiments demonstrate that converged test accuracy in hybrid training stays closer to that in fully synchronous training. In contrast, the training time is reduced significantly for encrypted gradients in the MNIST and FMNIST datasets compared to fully synchronous training.**

***Index Terms*—federated learning, homomorphic encryption, affine quantization**

I. Introduction

Federated learning (FL) [1] is a machine learning paradigm where multiple distributed clients collaboratively train a model by sharing model updates with a central server. The server aggregates these updates using algorithms like FedAvg [2] and sends them back to the clients for continued training. Horizontal Federated Learning (HFL) is a scenario where clients' data has little overlap in sample space but significant overlap in feature space.

Homomorphic Encryption [3] is used to secure model updates in FL. It allows operations on encrypted data, enabling secure aggregation of model updates. Hence, robust solutions like the Multi-Key Homomorphic Encryption scheme xMK-CKKS [4] have been adopted. However, this encryption scheme introduces significant overhead by complex operations like modular multiplication and exponentiation.

The choice of synchronization method in distributed training impacts performance. A synchronized approach requires all clients to be updated in a single iteration. An asynchronous approach allows flexible update timing, enabling faster convergence but potentially lower accuracy. To leverage the benefits of both methods, a hybrid synchronization scheme has been introduced to tackle the overhead due to homomorphic encryption. A previous work [5] has shown the benefit of using both BSP and ASP in training a deep neural network. The aforementioned hybrid scheme is built as an inspiration for their work. In our scheme, the training starts in synchronous mode. This process depends on the slowest client, so we switch to asynchronous mode. In asynchronous training, clients upload their model gradients to the server as soon as they finish, without waiting for others. Furthermore, a weighted aggregation is adopted based on client availability and data distribution.

II. Methodology

A. Hybrid Synchronization Approach

Hybrid federated learning training switching from synchronous to asynchronous mode. The switch occurs after completing 3.125%, 6.25%, 12.5%, and 50% of communication rounds in the server.

B. Multi-Key Homomorphic Encyption

Upgraded version of the MK-CKKS including only the additive feature based on Ring Learning with errors (RLWE). It includes the following steps: Client Initialization, Encryption, Aggregation, Decryption Share, and Decryption.

C. Affine Quantization

Reduce the precision of the client gradients.

D. Dynamic Weight Allocation

Depends on client availability and data distribution.

E. Gradient Sparsification

Reduce non-zero elements in the client updates.

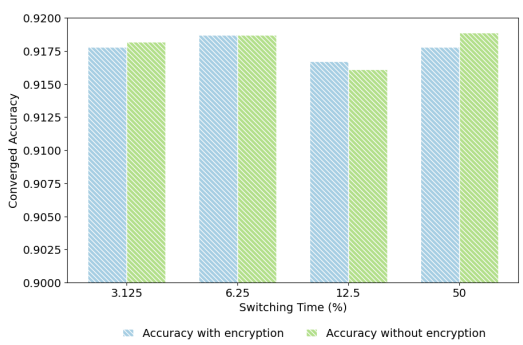
III. Experiment and Evaluation

A. Experiment Setup

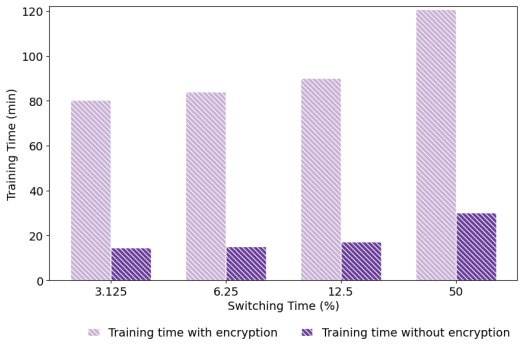
- **Machine** 1 server & 4 clients from Google Cloud
- **Storage** Google Cloud storage bucket
- **Dataset** MNIST & FMNIST
- **Model** 1 input, 2 Dense, and 1 output layer(s)

B. Evaluation

- **Hybrid synchronization with encryption**
- **Full Synchronous with encryption**
- **Hybrid synchronization without encryption**

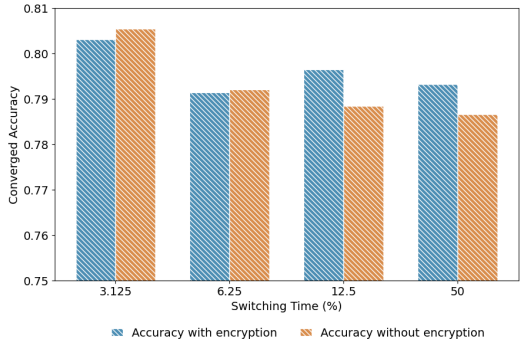


(a) Impact of switching times on the converged accuracy of the model

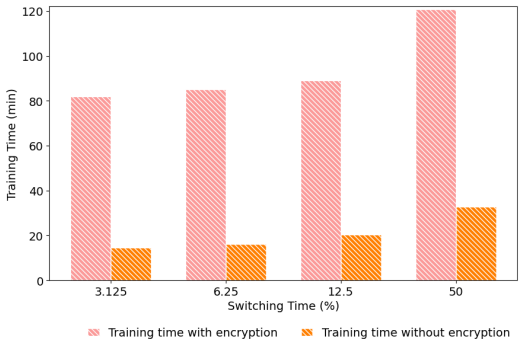


(b) Impact of switching times on the overall training time of the model

Fig. 1: Hybrid synchronization training on MNIST dataset



(a) Impact of switching times on the converged accuracy of the model



(b) Impact of switching times on the overall training time of the model

Fig. 2: Hybrid synchronization training on FMNIST dataset

C. Results

Figures 1a, 1b show results for MNIST dataset, and 2a, 2b for FMNIST. Figure 3 represents usage of quantization for MNIST. Table I shows results under various partitions for FMNIST.

TABLE I: Accuracy and Training Time under various partitions of the FMNIST dataset

Partition	Accuracy	Training Time
[25%, 25%, 25%, 25%]	78.05%	89.2 min
[6.6%, 80%, 6.7%, 6.7%]	76.43%	91.11 min
[10%, 70%, 10%, 10%]	75.59%	93.29 min

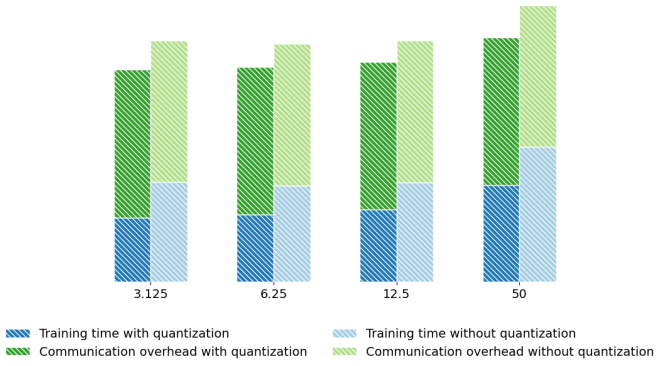


Fig. 3: Training time and communication overhead with & without quantization for MNIST dataset

References

[1] J. Konecny, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency, https://arxiv.org/abs/1610.05492, 2016.

[2] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data, in Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS), vol. 54, 2017

[3] S. S. Sathya, P. Vepakomma, R. Raskar, R. Ramachandra, and S. Bhattacharya, "A review of homomorphic encryption libraries for secure computation, https://arxiv.org/abs/1812.02428, 2018

[4] J. Ma, S.-A. Naas, S. Sigg, and X. Lyu, "Privacy-preserving federated learning based on multi-key homomorphic encryption, in International Journal of Intelligent Systems, 2022

[5] S. Li, O. Mangoubi, L. Xu, and T. Guo, "Sync-switch: Hybrid parameter synchronization for distributed deep learning, in 2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS), 2021, pp. 528–538