Accelerating Differential Privacy-Based Federated Learning Systems

Mirco Mannino, Alessio Medaglini, Biagio Peccerillo, Sandro Bartolini
Department of Information Engineering and Mathematics
University of Siena
Siena, Italy
e-mail: {mannino | medaglini | peccerillo | bartolini}@diism.unisi.it

***Abstract*—The number of mobile, wearable, and Internet of Things (IoT) devices we are using is increasingly growing, especially those implementing machine learning applications on-the-edge. Relying on a centralized server for processing and storing of this ever-increasing amount of data might not be the optimal solution, from both performance and privacy points of view. *Federated Learning* is a good solution to avoid sending user's data to a central server to train machine learning models. In order to guarantee privacy in a *Federated Learning* system, it is possible to leverage several techniques. *Differential Privacy* is one of the most popular, since it provides robust privacy protection. In this paper, we target mobile devices, proposing ideas on how to speed up training in *Differential Privacy*-based *Federated Learning systems* through a dedicated hardware accelerator.**

Keywords-differential privacy; federated learning; hardware accelerator.

I. Introduction and Background

Traditional Machine Learning (ML) approaches expect to collect data into a central computational system (e.g., server) and train models using such data. Nowadays, with the unstoppable diffusion of mobile devices (e.g., smartphones and wearable ones), more and more data is being collected locally, with a growing need to keep it private and unshared. Federated Learning (FL) was introduced by Google in 2017 [1] as a solution to implement a distributed training approach where individual model replicas are trained locally on different user's devices, and then global aggregated training strategy is orchestrated. In a FL system, there are two kinds of players: 1) a centralized server and 2) a group of N clients. Each client has a local database built with data collected locally that should not be shared with others. The training process can be summarized in the following steps:

- The server shares an untrained model among clients;
- Each client performs a local training procedure using its own data;
- Clients send trained models to the central server;
- The server aggregates them into an updated model;
- The server shares the updated model among clients.

The training process is iterative, continuing until it converges on the optimal model. During training, it is also possible that clients exchange parameters among themselves. Figure 1 shows an overview of a FL system.

One of the key aspects of FL is ensuring the privacy of data collected locally. Among all the techniques proposed to ensure privacy, Differential Privacy (DP) is one of the most promising [2]. Although DP can be achieved in different ways, the key-idea is to add noise to guarantee privacy. There are several research proposals that leverage DP [3]: 1) in *Local DP* techniques (e.g., [4]), the clients alter local data and send them to the server for centralized aggregation, protecting both the clients and the server from potential private information leaks; 2) DP based distributed Stochastic Gradient Descent (e.g., [5]) techniques aim to perturb the gradient during the training phase on the client devices; 3) DP meta learning [6] techniques aim to learn a model that can quickly adapt to new tasks using a few data points.

Another aspect concerning the world of FL, which is not usually taken into account in *traditional* ML applications, is the possibility of doing training at the edge. Indeed, usual ML models are trained on high-end platforms equipped with several types of hardware accelerators, e.g., Tensor Processing Unit (TPU). With the FL approach, client devices need to be readapted in order to guarantee efficiency and performance during the training phase. Indeed, training, compared to inference, involves the repetition of several steps: feed-forward, backpropagation, and weight gradient [7], [8]. For this reason, companies are encouraged to produce systems allowing training on the edge, especially introducing more storage and specialized hardware. In terms of specialized hardware, it is possible to distinguish three main categories [9]: Graphics Processing Unit (GPU), Field Programmable Gate Array (FPGA), and Application Specific Integrated Circuit (ASIC). GPU-based acceleration is the more flexible in terms of programmability, but it leads to a higher power consumption compared to the other two categories. On the other hand, FPGA- and ASIC-based accelerators allow to reach higher

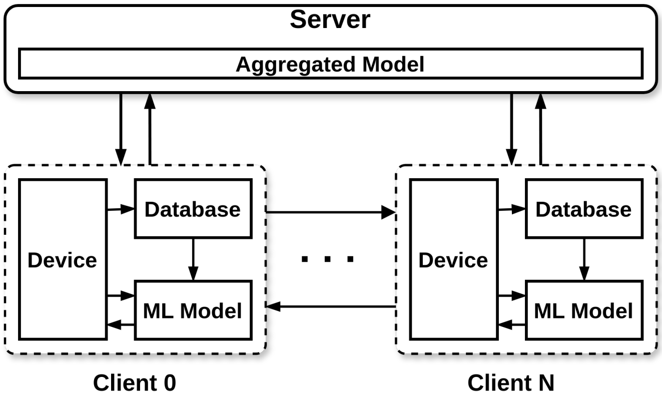

Figure 1. Overview of a Federated Learning system.

Copyright (c) IARIA, 2024. ISBN: 978-1-68558-184-8

performance and lower power consumption. ASICs suffer from a lower flexibility in terms of design and development, compared to FPGAs.

Nevertheless, it is worth mentioning that, in a FL system, there is a huge degree of heterogeneity and it is not possible to assume that all the client nodes incorporate the same hardware resources. Finding solutions that can be optimal for all the categories is the only way to achieve an efficient training phase from a system perspective.

The remainder of the paper is organized as follows. In Section II, we summarise the key points of differential privacy-based FL acceleration from our point of view, and in Section III we give an overview of a possible hardware accelerator design. Finally, in section IV we conclude.

II. Acceleration of Differential Privacy Training

It is clear that FL brings new challenges to the client devices, both from an algorithmic and architectural points of view. Differential Privacy adoption needs the addition of *noise* to the local data before sending them to the server, and the training process performed on-the-edge needs to be as much performant and efficient as possible.

One of the main challenges is to leverage, in an efficient way, the heterogeneity of client platforms and their interaction with the server node. We believe that a hardware/software (HW/SW) co-design is the key to find an optimal solution to the problem. In particular, there is the need to design and implement solutions that can be adopted by all the client platforms. The key points can be summarized as follows:

- Robust framework allowing the orchestration of all the players in the system.
- Algorithmic improvement for DP both on client and server side.
- Introduction of specialized hardware in heterogeneous architectures to accelerate common operations in FL systems, ensuring energy efficiency.

The increasing interest in FL led to the creation of several open-source frameworks, such as Tensorflow Federated [10] and FATE [11]. The open-source nature of these frameworks provides engineers with robust tools that are continuously developed and improved. For this reason, we focus our attention on the other two points of the previous list.

Algorithmic improvements and design of specialized hardware can be developed together as a HW/SW co-design process. Since we are targeting DP-based FL, the algorithmic aspects include both DP and deep learning training operations.

Our proposal is to design and evaluate a dedicated circuit, called *Federated Learning Processing Unit* (FLPU). It should be integrated in the current architectures, providing highly specialization in DP operations (e.g., noise addition, encryption) and deep learning operations carried out during training (e.g., backpropagation). The novel module should be included in any client platform: as an autonomous module on GPUs, an IP-block for Systems-on-a-Chip or FPGAs, adapting to the compatibility needs of each of them. This way, the programming side would also benefit.

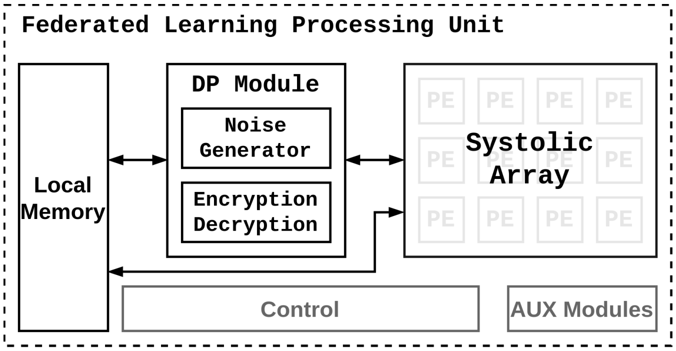

Figure 2. Overview of the Federated Learning Processing Unit (FLPU).

Under DP conditions, the training process performed among several clients may need encryption/decryption operations, and the generation of noise according to a certain distribution. For this reason, the FLPU should be equipped with an engine capable of efficiently carrying out these operations. At the same time, provisions should be made for deep learning-related tasks, including specialized hardware to accelerate deep learning processes (e.g., systolic arrays) and dedicated memory for storing weights during the backpropagation phase. The adoption of novel and existing algorithmic optimizations (e.g., reduction of memory consumption during backpropagation [8]) should be evaluated in order to reach an optimal design.

III. Federated Learning Processing Unit

The FLPU is in charge of accelerating the training phase on the heterogeneous client devices under DP conditions. Figure 2 shows an overview of its architecture. The systolic array in the architecture is used to accelerate deep learning computations, e.g., matrix multiplication. The systolic array can be implemented using different dataflow strategies. For example, input-, weight-, and output-stationary dataflows can be utilized [12]. A more detailed workload analysis is essential to determine the optimal design choice. Moreover, a series of auxiliary modules are included in the architecture. In particular, these modules are useful to accelerate common operations in deep learning algorithms, such as activation function and quantization.

The FLPU is equipped with an on-chip local memory. It is used to store input data, weights, and output data. The specific design of the memory will be established after an accurate assessment of the workloads involved. Among the potential solutions, one option is to use a single memory unit to store all data types, or alternatively, multiple local memories, each dedicated to storing specific types of data.

One of the key components is the *DP module*, responsible for ensuring the implementation of differential privacy mechanisms. The main role of DP module within the FLPU is the possibility to add noise and encrypt data within the accelerator itself. Indeed, this design introduces an additional layer of security that a conventional accelerator (e.g., TPU) would not have. The DP module is mainly composed of two components: *noise generator* and *encryption/decryption* *module*. The noise generator exploits some random physical signals that can be read from the device (e.g., temperature). The encryption/decryption module can be implemented by exploiting cryptographic accelerator designs [13].

IV. Conclusion and Future Work

Federated Learning is a promising approach to exploit computation on-the-edge and preserving users' privacy. In this paper, we focus on *Differential Privacy*, discussing how it can be implemented in FL systems. Moreover, we point out the key points needed to obtain a performant and energy efficient heterogeneous FL system. In the future, we will explore these aspects further, starting with the design and implementation of ad-hoc modules, either as novel chips or integrated into existing architectures.

Another area of focus will be the HW/SW co-design required to efficiently implement both DP and deep learning training operations. In this scenario, a deeper investigation of both DP and deep learning training algorithms is needed to jointly understand and optimize them.

References

[1] 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*, ser. Proceedings of Machine Learning Research, PMLR, 2017.

[2] A. El Ouadrhiri and A. Abdelhadi, "Differential privacy for deep and federated learning: A survey," *IEEE access*, vol. 10, pp. 22 359–22 380, 2022.

[3] K. Wei *et al.*, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE transactions on information forensics and security*, vol. 15, pp. 3454–3469, 2020.

[4] S. Wang *et al.*, "Local differential private data aggregation for discrete distribution estimation," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 9, 2019.

[5] N. Wu, F. Farokhi, D. Smith, and M. A. Kaafar, "The value of collaboration in convex machine learning with differential privacy," in *2020 IEEE Symposium on Security and Privacy (SP)*, IEEE, 2020, pp. 304–317.

[6] J. Li, M. Khodak, S. Caldas, and A. Talwalkar, "Differentially private meta-learning," *arXiv preprint arXiv:1909.05830*, 2019.

[7] Y. Wang *et al.*, "Trainer: An energy-efficient edge-device training processor supporting dynamic weight pruning," *IEEE Journal of Solid-State Circuits*, vol. 57, no. 10, 2022.

[8] N. Kukreja *et al.*, "Training on the Edge: The why and the how," in *2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, IEEE, 2019, pp. 899–903.

[9] H. G. Abreha, M. Hayajneh, and M. A. Serhani, "Federated learning in edge computing: A systematic survey," *Sensors*, vol. 22, no. 2, p. 450, 2022.

[10] "TensorFlow Federated: Machine Learning on Decentralized Data," [Online]. Available: https://www.tensorflow.org/federated (visited on 09/18/2024).

[11] "An Industrial Grade Federated Learning Framework," [Online]. Available: https://fate.fedai.org/ (visited on 09/18/2024).

[12] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295–2329, 2017.

[13] N. Samardzic *et al.*, "F1: A fast and programmable accelerator for fully homomorphic encryption," in *MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture*, 2021, pp. 238–252.

Copyright (c) IARIA, 2024. ISBN: 978-1-68558-184-8