Triad of Split Learning: Privacy, Accuracy, and Performance

Dongho Lee
Dept. of Mobile Systems Eng.
Dankook University
Yong-in, South Korea
jeff3093@dankook.ac.kr

Jaeseo Lee
Dept. of Mobile Systems Eng.
Dankook University
Yong-in, South Korea
jaeseoleee@dankook.ac.kr

Hyunsung Jun
Dept. of Mobile Systems Eng.
Dankook University
Yong-in, South Korea
jun971103@dankook.ac.kr

Hongdeok Kim
Dept. of Mobile Systems Eng.
Dankook University
Yong-in, South Korea
qchdkim@dankook.ac.kr

Seehwan Yoo
Dept. of Mobile Systems Eng.
Dankook University
Yong-in, South Korea
seehwan.yoo@dankook.ac.kr

***Abstract*—Split learning is a new machine learning model, considering the distributed users' privacy. While preserving the privacy of user data, split learning can leverage an amount of training data from multiple users. This paper presents how split learning can efficiently trade privacy, prediction accuracy, and training overhead. We devise three practical implementation models of split learning with different levels of privacy. Our experiment shows that privacy, accuracy, and training overhead are differently presented according to the implementation model. The result supports that privacy-preserving layer in split learning enhances privacy with marginal processing overhead, and we can achieve reasonably high accuracy, compared with the local model with limited dataset.**

***Index Terms*—Machine learning, Privacy-preserving machine learning, split learning**

I. Introduction

Machine learning (ML) uses more and more privacy-sensitive data to learn human life. Smart phones, watches recognize the voice and face of the user [1], know favorite food and restaurants [2], and react with daily emotion [3]. With the ML, those smart devices listen your acoustic voices, memorize where you have been, what you have bought, and monitor your biological, psychological conditions for 24/7. With the increasing popularity of AI/ML applications on smart devices, attacks on machine learning such as model inversion, membership inference has become a major concern for users.

More seriously, machine learning techniques are used in mission-critical systems such as automotives, medical systems, that literally affects human lives. Owing to the advances in computational hardware performance, ML enables hospitals to accurately diagnose cancers [4], to quickly find new medications against disease [5], to prevent sudden heart failures [6]. Yet, it requires an amount of data, that is very private, patients' medical records. Because medical information is classified with high privacy level, regulations strictly prohibit sharing patients medical data over public Internet. Currently, each hospital usually conducts ML study with the limited data silo, limiting the accuracy.

Addressing the privacy preservation and limited data set issues, federated learning (FL) has been proposed [7]. In federate learning, users/participants share learned models, instead of raw data. Therefore, FL tries to avoid private data breaches. In a recent federated learning platform [8], each participating hospital constructs the local model from its own dataset. Then, they send the learned models to the central server. The server integrates the models from different hospitals, minimizing the overfitting, outliers in the model. Combining local models, FL tries to improve accuracy, which requires subtle adjustment. Usually the integration server calculates the weighted average from the models of participants. When there are numerous participants, the integration server carefully adjust the weights among different participants. Note that different participants have different measuring tools, different capability in analyzing data.

Yet, there are some cases where traditional federated machine learning is hard to be applied. Smart phones and watches, and tiny IoT devices have limited computing resources, and limited battery capacity. It is hard to imagine a user runs ML process, instead of inference process, in a smart watch. Currently, the smart devices send out the data gathered from the sensors under the permission of its user. That is, you always deliver your private information to Facebook, Google, Amazon, etc, so that they can learn about you.

This paper presents a practice of privacy-preserving machine learning technique, called split learning. Although split learning idea has been presented in [9], its privacy-performance trade-off relationship has not been thoroughly presented yet. Therefore, a goal of this paper is to search the ML space that best fits the split learning, and to present the trading relationship between the ML accuracy, data privacy, and processing overhead of ML participants.

Our experiment shows that the split learning preserves the

978-1-6654-2383-0/21/$31.00 ©2021 IEEE

privacy by separating the learning process in two different servers, narrowing the attack surface. In addition, it presents competitive accuracy and learning curve. Further, it allows much light-weighted participants than decent federated learning.

II. Background and Related Work

A. Federated Learning and attacks on shared data

In 2016, google coined Federated learning (FL) to deal with the privacy issues in ML. FL enhances the privacy of ML data because the local site does not need to upload raw data to the central ML server. Instead, each local site delivers local model learned from its own dataset.

There are several FL models and architectures [10], but the common FL protocol is as follows. The learning process of FL goes through multiple rounds. Initially, each site conducts the local learning with its own dataset, updating the weights. Then, the learned model (or gradient values) is sent to the FL server. The server aggregates the local models from different sites. The aggregated model is distributed back to local sites for the next round.

In the FL learning process, privacy-sensitive raw data is not shared. However, the participants in the local sites and the central server exchange model and gradients over the public Internet. Although it seems safer than exchanging the raw data, model and gradients can also be used to attack the model or data privacy. Many FL attacks focus on updates of model or gradient values from the participants.

In a recent survey of FL attack [11], the authors identified malicious actors in FL, (a) malicious server, (b) insider adversary, and (c) outsider adversary. An untrusted participants can poison the data (data poisoning attack) or model (model poisoning). The goal of poisoning attacks are usually making bad model. Malicious participant could inject bad label or compromised data, which leads to ill-learning, dropping the accuracy.

In addition to the poisioning attack, FL is also known to be vulnerable to inference attacks. Inference attack assumes a passive attacker or a malicious server. A passive attack can observe the FL traffic, particularly on the changes of model or gradients. A study reveal that an adversary can learn about the data, membership information, properties in the local site, only by observing the gradient update in FL.

In the next section, we will present the attack surface and threat model of split learning, comparing with FL.

III. Split Learning

Split learning is a new distributed machine learning method with multiple participants and the split learning server. Although the split learning idea has been proposed in, its feasibility in terms of the accuracy and practical implementation has not been thoroughly studied.

In this section, we present a detailed model of split learning with some implementation options. Figure 1 is the overall structure of split learning. In the figure, each participant has its own local dataset. Each participant sends out data to the split learning server only after running through the privacy-preserving layer. The layer includes filters and activation function that sanitizes the privacy-sensitive information from the raw data. The split learning server merges the dataset from the participants, and then trains the internal DNN layers from the sanitized data. The entire training process is divided into two parts: local training, and remote training. Local training focuses on privacy sanitizing, and remote training

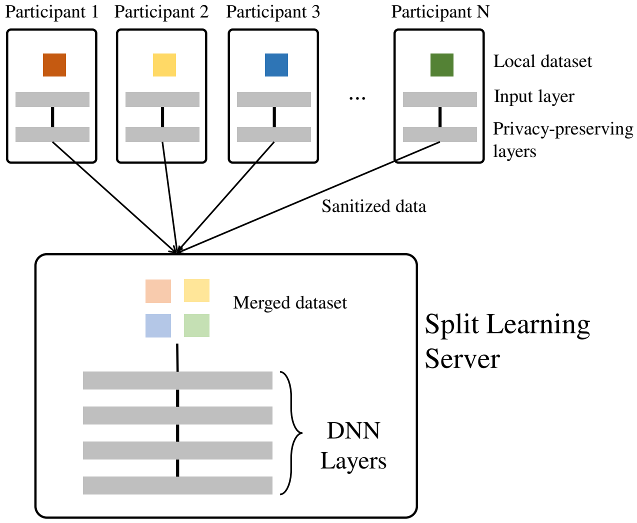


Fig. 1: Overall structure of Split Learning

Although split learning looks similar to federated learning, there is a key difference between them. In Federated learning, each participant builds its own model, and the model is integrated at the central server. On the other hand, participants in split learning does not build the entire model. Instead, the participants send out data to the split learning server. To address the privacy issue of the learning data, the sender sanitize it before sending over the Internet. The privacy-preserving layer includes some security protocol that minimize the privacy leakage.

A. Threat model and Security Properties of Split Learning

Similar to Federated Learning, split learning assumes actors in different roles. Firstly, we assume trusted participants. That is, participants are not intentionally sending compromised data. The participants should consent to providing his/her data; thus, we trust the participants are willing to provide correct information. Secondly, we assume trusted split learning server. The goal of the server is to generate the high accuracy model using a large volume of training dataset. If we assume a benign server, no participant will consent to provide data.

On the other hand, we assume a hostile network environment. We assume an eavesdropper, a passive attacker who tries to overhear the network traffic. In the federated learning, the eavesdropper can overhear gradient update or model update, so that it can leak privacy of training data.

In addition, we assume an active attacker who tries to mount forgery or replay attack. In the federated learning, the active attacker can poison the model and data, so that the server cannot achieve high accuracy. The attacker can play a forged server, so that the participants can send its data to the malicious server, instead of the split learning server.

To mitigate the basic threats that federated learning assumes, we introduce authentication and integrity check for all network communication, and randomization on shared data.

The privacy preserving layer prepares for sending privacy-insensitive data over the network. In the process, the participant authenticates the split learning server. This mitigates the forged server attack, and participants now can establish secure communication channel to the split learning server.

Then, the data is sent over the network. At this time, the server also authenticates the participant, and the server checks for the integrity of the received data. Thus, we can mitigate the data and model poisoning attacks because the participants does not receive ML data from server, and the server checks the integrity of the received data.

Finally, the data on the network is randomized; thus eavesdropper cannot easily learn about training dataset. To randomize the data, the privacy-preserving layer establish the communication channel to the split learning server. The server generates a nonce, and then it is used as a seed for randomizing the training data.

B. The Privacy-Preserving Layer

The privacy-preserving layer plays a key role in split learning. It preserves the privacy of training data with some filtering and randomization, and it transfers only the sanitized data.

Note that the sanitized data also should be used as training input at the split learning server. Therefore, just randomization does not work, and it should be feasible to use as a machine learning layer.

To sanitize, we propose three techniques for implementing the privacy-preserving layer.

1) Case 1: Random Vector: The first technique is to use a random vector as a privacy-preserving layer. The split learning server generates a random nonce, and it is shared with participants over secure communication channel. The reason for using the same random value among all the participants is that the sanitized data should be used as training input. If participants use different randomized values, the training should not be successful.

Note that we assume a trusted split learning server and trusted participants; but none in the network are trusted. This case assumes secure communication between the server and participants. Only authenticated participants can get the vector; thus, the privacy preserving layer can encrypt the training data.

Because all participants share the same random vector for the same machine learning session, the eavesdropper can mount differential attack. Also note that once the vector is leaked for a single node, all training data from all participants are leaked.

2) Case 2: Random Vector with an Activation Layer: Using an activation layer along with random vector can be another way to implement a privacy preserving layer. ReLU (Rectified Logical Unit) function is the most commonly used activation function in neural network. ReLU function zeros out the value if the value is under some threshold. Because the ReLU serves as a non-linear filter, which has no inverse function, we may eternally lose some value if it is filtered out.

The filter generates some noise to the existing value, making harder to derive the original random vector. Each participant has its local dataset. The participant passes original private data through privacy-preserving layer, likewise the forwarding path in the machine learning pipeline. The output of the privacy-preserving layer is sanitized data.

The sanitized data loses some privacy-sensitive information, due to the randomization and filtering in the activation layer. Yet, the activation layer selectively chooses the live neuron from the others, making the sanitized data better to be trained.

In contrast to method of previous method, ReLU function has tolerance for inverse calculation. Therefore, this technique is less vulnerable to differential attack. However, the initial random vector is still used; thus, once the random vector is leaked, all training data is easily breached.

3) Case 3: Pre-training in clients: The third technique is to use pre-training with local dataset. In this technique, each participant makes privacy-preserving layer as a small learning layer with its own local dataset. From the initial random vector of split learning server, all participants trains its own local model. The local model can be simply constructed with single Conv2D, Activation, Dense layers. Participants have different dataset; thereby, participants should use different local models.

Once the local training is completed, the participants passes original private data through privacy-preserving layer. The sanitized data adds more noise than Case 2 because the data set passes through locally-biased prediction model. Besides, Case 3 has another advantage over the other cases because it does not use shared randomization value. Differential attack is almost impossible to be mounted. At the same time, the split learning server also cannot reason about the original training set because the server does not have any information about the final local model.

IV. Experiments

This section presents the experimental results of the proposed split learning. We used two dataset to present the performance of split learning. First, we used a medical dataset from SNUH (Seoul National University Hospitl). The dataset contains anonymized height, weight, sex and cholesterol values(TC, HDL-C, TG, LDL-C) of about 400,000 patients. We used linear regression model for training LDL-C values from the rest of information.

Second, we used CIFAR-10 [12], that is one of the popular image classificiation training dataset. CIFAR-10 contains 60000 32x32 colour images in 10 classes, with 6000 images per class. We used RNN-variant training model for split learning.

We used TensorFlow v2.4, that supports Keras API. Regarding the hardware used for split learning, the split learning server is equipped with RTX 2070 Super GPU.

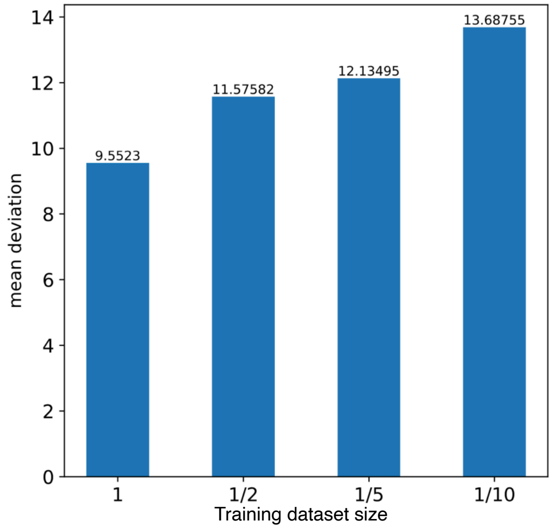


Fig. 2: Mean deviation according to the dataset size

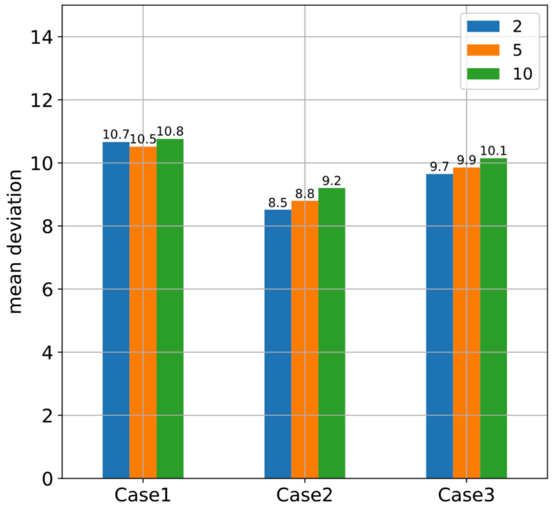


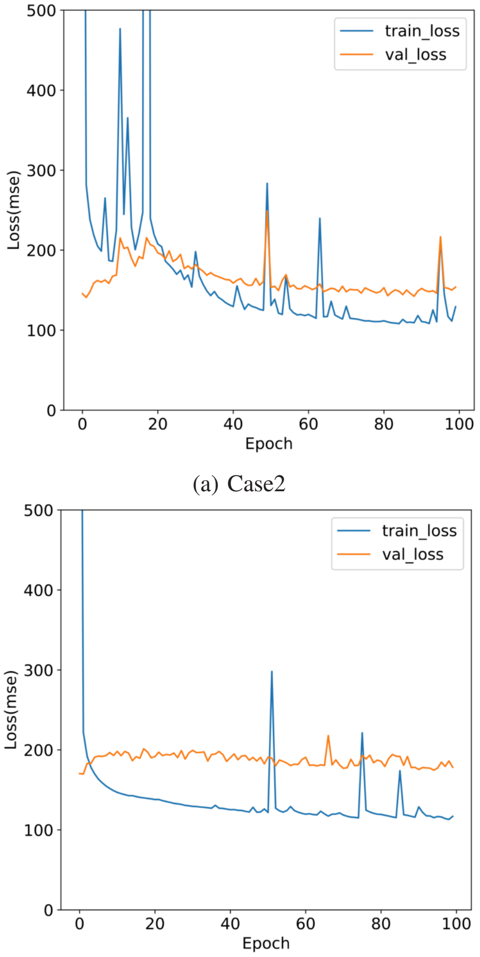
Fig. 3: Comparison of three implementation options

A. *Split Learning for Regression Model*

Figure 2 is a graph of mean deviations according to the dataset size. In the graph, as we reduce the dataset size to half, 1/5, 1/10, the mean deviation increases from 9.5 to 13.6. The graph shows that the accuracy of ML is dependent upon the dataset size. To obtain high-accuracy model, we need more data.

Figure 3 comparatively presents the mean deviation for the three cases of split learning. For each case, we evenly divide the dataset into 2, 5, and 10 different nodes, and run the split learning. Then, we present different privacy-preserving layer schemes effectively works as a part of machine learning layer. In case 1, each participant multiplies the dataset by a random vector. In case 2, each participant uses ReLU activation layer as a privacy-preserving layer. In case 3, each participant uses local model, and sanitizes the training data with local model.

In the graph, the mean deviation of case 1 is the highest among the three cases. In case 2, the mean deviation is the smallest among the three cases. In case 2, the split learning server gets training input after an activation layer. It seems like the activation layer effectively enhances the data quality of training at split learning. In case 3, the mean deviation has increased a bit, compared with case 2. In the graph, case 3 presents mean deviation about 9.9, which is comparable to the case when we use full dataset without split learning (in Figure 1).



(a) Case2

(b) Case3

Fig. 4: Training & validation loss (MSE) per epoch

The participants of case 3 runs through a local prediction before sending to the split learning server. Because the local prediction model is subject to the local dataset of the specific participant, the sanitized data is also subject to the sending participant. Thus, the man in the middle attack is hard to be mounted. In short, it preserves some privacy as well as achieves high accuracy.

In all cases, the impact of the number of participants are negligible, meaning that split learning can be scalable.

Figure 4a and 4b show how the training and validation loss(mean square error) change in case 2 and case 3 during 100 epochs. In Figure 4a, we can observe that the training loss decreases as epoch, which explains the model updates the weights, trying to optimize the costs. In Figure 4b, we can observe a stable curve of training loss, compared with the case

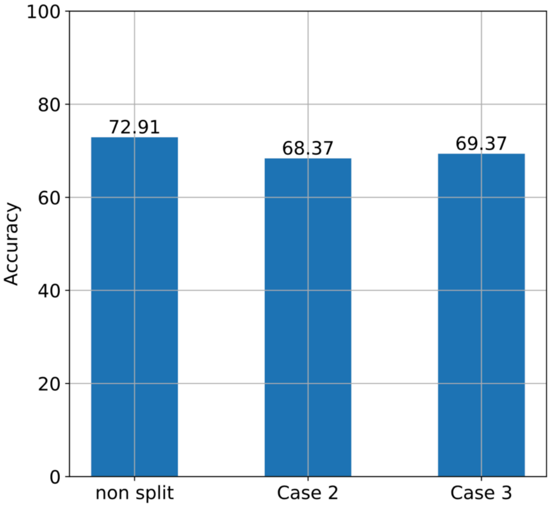


Fig. 5: Split learning performance for CIFAR-10 classification

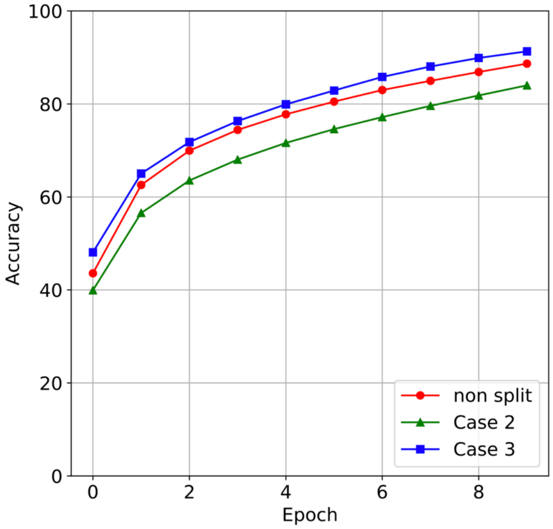


Fig. 6: training accuracy of split learning

2. It explains that the training process case 3 is much stable than case 2.

B. Split Learning for Classification Model

Figure5 shows the accuracy of split learning for the cifar-10 classification. We compare split learning case 2 and case 3 with non-split learning. The non-split learning assumes that the server has the entire dataset, and trains with DNN model. The split learning assumes 10 participants. For case 2, each participant uses ReLU activation function for the privacy-preservation layer. For case 3, each participant has a locally pre-trained model, that is used as a privacy-preservation layer.

In the graph, the accuracy is the highest when we do not use split-learning. Yet, we can show that the case 2 and 3 achieves comparable accuracy to non-split learning, meaning that split learning can achieve high-accuracy for prediction as well as classification.

Figure 6 shows the training curve for the above three cases. In all three cases, accuracy increases by a similar margin, and stable learning curve.

V. Conclusion

Split learning preserves the data privacy in the multi-party machine learning. This paper presents three feasible ways implement privacy-preserving layer in split learning. In addition to the data privacy, the split learning achieves comparable accuracy to local training. Throughout an extensive experiments, we observed that the split learning can stably conduct training, achieving high accuracy. Besides, our experiments support that the split learning is scalable as the number of participants. With large number of small participants, we could generate high-accuracy classification and regression models. Furthermore, the split learning requires small training overhead at participants side. It does not require huge GPU memory and processing power; with only a few computational layers, split learning achieves high accuracy, privacy at the same time. As future work, we are going to develop a theoretic analysis on privacy such as differential privacy in split learning, and the in-depth understanding of trade-off relationship in computation overhead and balancing between the split learning server and participants.

References

[1] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, “Face recognition: A literature survey,” *ACM Comput. Surv.*, vol. 35, no. 4, p. 399–458, Dec. 2003. [Online]. Available: https://doi.org/10.1145/954339.954342

[2] X. Li, W. Jia, Z. Yang, Y. Li, D. Yuan, H. Zhang, and M. Sun, “Application of intelligent recommendation techniques for consumers’ food choices in restaurants,” *Frontiers in Psychiatry*, vol. 9, p. 415, 2018. [Online]. Available: https://www.frontiersin.org/article/10.3389/fpsyt.2018.00415

[3] O. Mitruț, G. Moise, L. Petrescu, A. Moldoveanu, M. Leordeanu, and F. Moldoveanu, “Emotion classification based on biophysical signals and machine learning techniques,” *Symmetry*, vol. 12, p. 21, 12 2019.

[4] K. Kourou, T. P. Exarchos, K. P. Exarchos, M. V. Karamouzis, and D. I. Fotiadis, “Machine learning applications in cancer prognosis and prediction,” *Computational and Structural Biotechnology Journal*, vol. 13, pp. 8–17, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2001037014000464

[5] A. Rajkomar, J. Dean, and I. Kohane, “Machine learning in medicine,” *New England Journal of Medicine*, vol. 380, pp. 1347–1358, 04 2019.

[6] R. Katarya and P. Srinivas, “Predicting heart disease at early stages using machine learning: A survey,” in *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, 2020, pp. 302–305.

[7] X. Yin, Y. Zhu, and J. Hu, “A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions,” *ACM Comput. Surv.*, vol. 54, no. 6, Jul. 2021. [Online]. Available: https://doi.org/10.1145/3460427

[8] Q. Yang, Y. Liu, T. Chen, and Y. Tong, “Federated machine learning: Concept and applications,” *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, Jan. 2019. [Online]. Available: https://doi.org/10.1145/3298981

[9] P. Vepakomma, O. Gupta, T. Swedish, and R. Raskar, “Split learning for health: Distributed deep learning without sharing raw patient data,” *CoRR*, vol. abs/1812.00564, 2018. [Online]. Available: http://arxiv.org/abs/1812.00564

[10] I. Kholod, E. Yanaki, D. Fomichev, E. Shalugin, E. Novikova, E. Filippov, and M. Nordlund, “Open-source federated learning frameworks for iot: A comparative review and analysis,” *Sensors*, vol. 21, no. 1, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/1/167

[11] L. Lyu, H. Yu, and Q. Yang, “Threats to federated learning: A survey,” *CoRR*, vol. abs/2003.02133, 2020. [Online]. Available: https://arxiv.org/abs/2003.02133

[12] A. Krizhevsky and G. Hinton, “Learning multiple layers of features from tiny images,” University of Toronto, Toronto, Ontario, Tech. Rep. 0, 2009.