A Study of Split Learning Model

Jihyeon Ryu
Department of Software
Sungkyunkwan University
Suwon, Korea
jhryu@security.re.kr

Dongho Won
Department of Software
Sungkyunkwan University
Suwon, Korea
dhwon@security.re.kr

Youngsook Lee*
ITSoftwareSecurity
Howon University
Gunsan, Korea
ysooklee@howon.ac.kr

***Abstract*—Split learning is considered a state-of-the-art solution for machine learning privacy that takes place between clients and servers. In this way, the model is split and trained, so that the original data does not move to the client from the server, and the model is properly split between the client and the server, reducing the burden of training. This paper introduces the concept of split learning, reviews traditional, novel, and state-of-the-art split learning methods, and discusses current challenges and trends.**

***Index Terms*—Split Learning, Machine Learning, Distributed Learning, Convolutional Neural Networks**

I. Introduction

Artificial intelligence, which is indispensable in modern society, uses a large amount of data in the learning process, raising issues related to information protection of the collected data. In 2021, the AI chatbot 'Iruda', developed by Scatter lab, out of about 9.4 billion sentences from 600,000 people collected during the learning process, data such as name, mobile phone number, and address is exposed as it is while using the chatbot. So it occurs resulting in an invasion of privacy [1]. In addition, Amazon's artificial intelligence voice assistant Alexa had a problem in which personal information such as users' addresses were exposed while listening to recorded commands from users [2]. There is a risk that such a privacy violation problem will result in artificial intelligence learning if a large amount of data is not reliably preprocessed. In addition, when users send their data to the cloud, the original data is often used as it is, so data with sensitive information can be easily extorted even if only the data sent by the user is stolen. As in the case above, the problem of personal information infringement in artificial intelligence can occur without difficulty.

In particular, a deep neural network, a field of artificial intelligence, occupies a large share in data processing in the classification and prediction fields of high-dimensional data such as images, videos, and bio-sensors. When processing sensitive and large amounts of data, such as medical and health, it can cause more problems due to privacy and ethical issues. Recently, artificial intelligence researchers are interested in developing new deep neural network algorithms, but the difficulty of keeping user data private is a problem.

When using a deep neural network, a number of methods have been proposed to prevent privacy violations. Among them, the method of learning the model by separating the client and the server has attracted a lot of attention. In particular, federated learning [3] and split learning [4] recently studied at Google and MIT are attracting global attention. Split learning is a learning method designed to protect privacy in general supervised machine learning models.

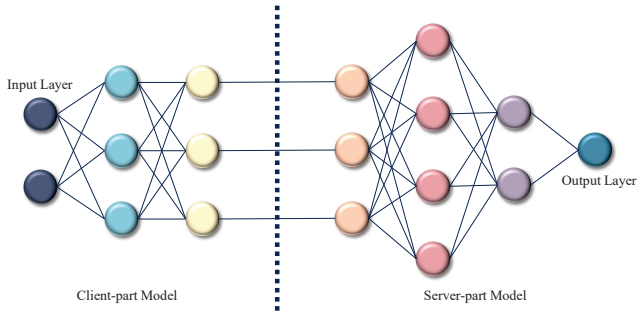

Fig. 1. Simple Vanilla Split Learning Model

This paper serves as an introduction to split learning as shown in Fig 1 and aims to examine the latest techniques in the field. We also aim to meet the challenges of split learning. The rest of the paper is structured as follows. Section II introduces split learning foundations before dealing with specific models of split learning. Section III deals with how to design a split learning model, and Section IV looks at three split learning methods. Finally, we conclude in Section V.

II. Split Learning Foundations

A. Federated Learning

Split learning is a method of federated learning, a technology in which a local client and a server cooperate to learn a global model in a situation where data is decentralized. In this case, the local client refers to a device that enables users to collect data, such as personal Internet of Things (IoT) devices and smartphones. Federated learning [3] is first introduced in 2015, and it is introduced in Google AI blog in 2017 and received a lot of attention [5]. In federated learning, the client has a copy of the entire model and updates the weights. Afterward the server receives the updated weight values from all clients and averages them to update the weights.

978-1-6654-2678-7/22/$31.00

B. How to Split Learning protect User's Privacy?

The split learning method trains a model divided into a client and a server to protect the user's personal information. If the split learning model is not used, when the client wants to train his/her data on the server's deep learning model, the raw data is sent to the server. In this case, if the data containing sensitive information is stolen, the attacker can possess the original data, resulting in a privacy violation problem. In order to prevent such a privacy violation problem, the client trains data in advance through some models and sends the learned data to the server. It is difficult to recover the original data unless we know the structure of the model and the weights of the model.

C. Split Learning and Challenges

Split learning has not yet conducted much research that constitutes a model, and its performance is not superior to that of a general model. As the convolutional neural network is mainly researched, it is necessary to study voice data, image data, or string data. In addition, when an attacker steals data between a client and a server, a white box attack is possible if the attacker knows the model structure of the client and knows some input data and output data. In this case, since model reconstruction attacks are possible, the original data can be calculated. Even in this case, we need a way to defend against attacks.

III. Building a Split Learning Model

In this section, we describe in detail how to build a split learning model. In the process of creating the model, the convolutional neural network models are taken as an example based on the previous paper [4], [6], [7], [9] that create the split learning model. In this section, which convolutional neural network models are used, the three structures of the split model, and how the client and server learn are explained.

A. Classic Convolutional Neural Network Model

So far, many papers have studied effective building block construction methods with convolutional neural network models. This section briefly introduces the most popular classic networks [6], [7], [9] that underlie computer vision.

1) LeNet-5: LeNet [6] is the name of the first convolutional neural network algorithm developed in 1998 by Y. LeCun's research team, who first developed the convolutional neural network. LeNet was developed to recognize the handwriting of postal codes and checks and is composed as shown in Fig 2.

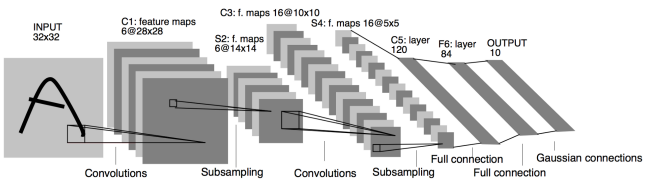

Fig. 2. Architecture of a Convolutional Neural Network LeNet-5 [6]

2) AlexNet: AlexNet [7] is a convolutional neural network structure that won the ILSVRC (ImageNet Large Scale Visual Recognition Challenge) competition held in 2012. The structure of AlexNet, named after the first author, Alex Krizhevsky, is as follows in Fig 3.

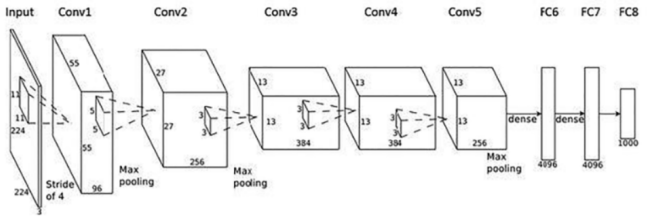

Fig. 3. Architecture of a Convolutional Neural Network AlexNet [7], [8]

3) VGG16: In Fig 4, VGG16 is shown. VGGNet [9] is a model developed by VGG, a research team at Oxford University, and it is the model that won the 2014 ImageNet image recognition contest. VGG16 means a model composed of 16 layers in VGGNet.

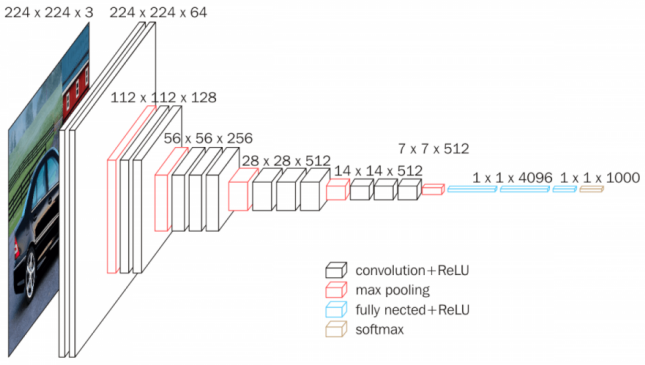

Fig. 4. Architecture of a Convolutional Neural Network VGG16 [9], [10]

B. Making a Split Model

Split learning [4] is literally related to how to learn a split model. Therefore, it is necessary to design a model that is basically divided. In this section, the structure of the client model and the server model is different depending on the method of sharing the label or not, and the method of processing the data of multiple clients at once.

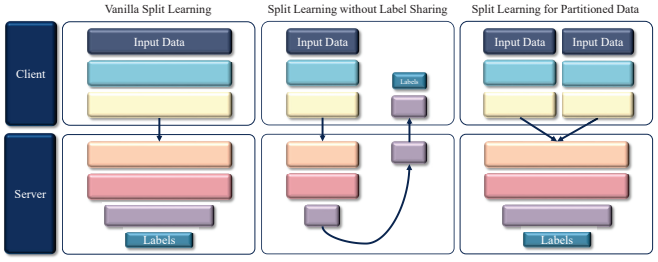

Fig. 5. Three Configurations of Split Learning

1) Vanilla Configuration for Split Learning: The structure of the simplest split learning model is when a whole model is shared only one client and one server, and also shares a label. In this setting, each client learns up to a layer close to the split data. When training to the split layer is completed, the data of that part is transmitted to the server. At this time, it is not necessary to share the original data, and the server learns the received data in the remaining layers without knowing the original data. This method is shown on the far left of the Fig 5.

2) U-Shaped Configurations for Split Learning without Label Sharing: In the method in the middle of the Fig 5, U-shaped structure is when a model is shared by only one client and one server, and the label is not shared. Because the client does not share the label, the learning is finally completed by the client. This can be solved with a U-shaped configuration.

3) Vertically Partitioned Data for Split Learning: This is a way to train a split model on multiple devices with different data without sharing data. In the method on the far right of the Fig 5, each client trains a partial model up to the split layer. Afterward, the learned data is sent to the server to complete the rest of the learning.

C. Client Model Training

Basically, the client model is trained before the server model. It learns in advance from the data that the client will not use and sends the preprocessed data to the server. At the same time, for the server to learn well, the data it sends must be completely random. In the case of split learning with a convolutional neural network, the client uses a convolutional neural network model used in the above section to learn data different from the data to be used. (For example, when learning animal images, numerical images are used for model training in advance.) When properly trained, only a part of the used model is used as the client model. In this case, a small number of layers are used to prevent data loss, and a value that is not smaller than the original data must be sent to the server.

D. Server Model Training

The server model trains the data that the client ultimately wants to learn after the client model is trained. In the case of Vanilla configuration for split learning, it receives the value received from the client, learns it with the label, and sends the result. In the case of U-shaped configurations for split learning without label sharing, unlabeled data are trained and sent to the client for finalization. Finally, in the case of vertically partitioned data for split learning, we receive labeled data from multiple clients and train them on one model at the same time.

IV. Split Learning Methods

A. Split Learning with Differential Privacy

ARDEN, a method that adds differential privacy to partition learning, is the method introduced in [11]. ARDEN depends on the mobile cloud environment and learns by dividing the deep neural network into local (client) and cloud (server).

Algorithm 1 Differential Privacy Transformation

Require: Sensitive data x_s, Local layer $\mathcal{L}$, Nullification matrix I_n, Noise scale σ, Boundary B, Injection layer l
Ensure: Noisy intermediate output x_r.

1: $x_s^{'} \leftarrow x_s I_n$
2: $x_l \leftarrow \mathcal{L}(x_s^{'})$
3: $x_{l,}^{'} \leftarrow x_l / max(1, ||x||_\infty / B)$
4: $x_l^{'} \leftarrow x_l^{'} + Lap(B/\sigma I)$
5: $x_r \leftarrow \mathcal{L}(x_l^{''})$

The pre-trained local neural network used performs feature extraction and perturbation in the learning stage. In the inference phase, perturbing the image in the local model. The data that appears disturbed in the local model is extracted with general features and transmitted to the cloud. In the training phase, the cloud additionally uses a noisy training method to ensure more safety. The following describes differential privacy and noisy training in more detail.

1) Differential Privacy: [11] inject perturbation that satisfies differential privacy, the main function of ARDEN, before learning. At this time, the method of calculating the privacy budget is as follows Theorem 1.

Theorem 1: Given the sensitive data x_s and the local neural network $\mathcal{L}$,

$$\epsilon = ln[(1-\mu)e^{2\sigma/\gamma} + \mu]$$

where $\gamma = ||\nabla_{x_l^{'}} \mathcal{L}||_\infty$

With epsilon created using the formula to calculate the privacy budget shown in theorem 1, we can convert differential privacy data using the following algorithm 1. For each sensitive data x_s, some values are masked by the nullification operation. This data is then used for training in a local neural network.

2) Noisy Training: A neural network using differential privacy in the client guarantees data privacy, but this process alone incurs a performance sacrifice in the server. In addition, due to the increase in loss, the accuracy of the neural network is relatively lower than when no noise is added. To alleviate this decrease in accuracy, ARDEN proposed a noisy training method. For this loss, compute the loss with the value a of the algorithm 2, and update the weights with the algorithm to minimize the loss of clean representation a, loss of noisy training $a^{'}$, and loss of perturbed noisy training $a^{''}$.

B. Split learning with 1D Convolutional Neural Network Model

[12] showed that split learning can be used in 1D convolutional neural network models. [12] implemented split learning in a 1D convolutional neural network model and applied time-series sequential data using ECG signals to detect cardiac anomalies that inherently avoid sharing medical data with other parties but still achieve the same accuracy as non-segmentation models. They also suggested a privacy evaluation framework and observed that the direct use of segmentation learning for 1D convolutional neural networks resulted in high

Algorithm 2 Noisy Training in Each Batch

Require: Clean representation x, Cloud-side neural network $\mathcal{N}$, Batch size N, Noise scale σ, Boundary B, Controllers λ and μ, Learning rate α;

Ensure: Weight $w \leftarrow w - \alpha d/N$

1: $a \leftarrow Loss(y, C(w, x))$
2: $x^{'} \leftarrow x + Lap(B/\sigma I)$
3: $a^{'} \leftarrow Loss(y, C(w, x))$
4: $g \leftarrow \nabla_x a^{'}$
5: $r \leftarrow \mu g / ||g||_2$
6: $a^{''} \leftarrow Loss(y, C(w, x^{'} + r))$
7: $a_{sum} \leftarrow \lambda a + (1 - \lambda)(a^{'} + a^{''})$
8: $d \leftarrow d + \nabla_w a_{sum}$

privacy leaks in applications with sensitive data such as ECG signals.

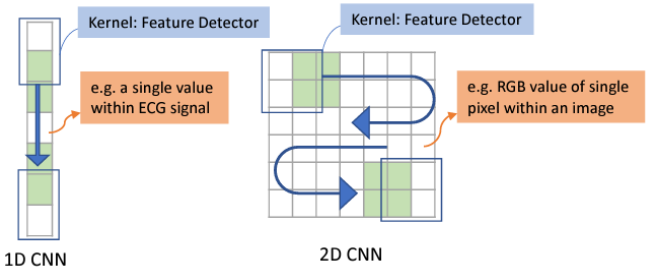

Fig. 6. Difference of 1D Convolutional Neural Network and 2D Convolutional Neural Network [12]

C. Split Learning with Spatio-Temporal

The following method in Fig 7 was used for [13], which performed split learning by dividing temporally. Individual results of the first hidden layer are sent to the central server, and the original data is not shared. When the central server shares the results of the first hidden layer, the original data is not exposed. The centralized server needs a queue while collecting the results of the hidden layer from the geographically dispersed end system. can In this case, the learning performance may be biased due to the difference in arrival time, so parameter scheduling should be defined.

V. Conclusion

Split learning separates the client and server models to ensure the privacy of user data in a neural network. This method ensures the privacy of user data by training the neural network model without confirming the client's original data directly on the server. In this paper, we described what we need to know to study split learning. In addition, our paper described how to create a split learning model. Finally, we covered the latest research using split learning.

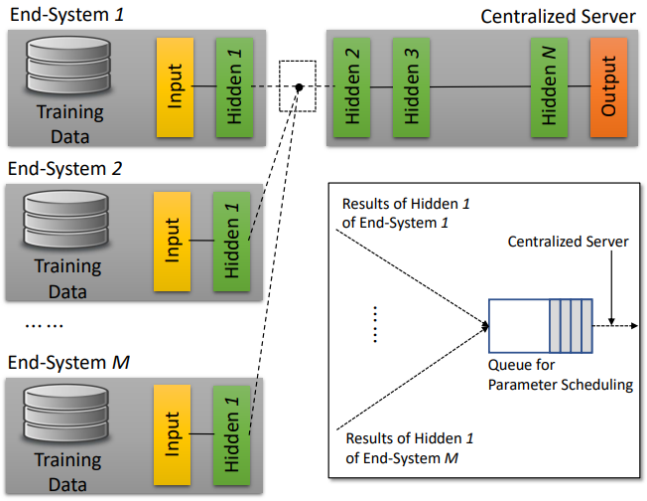

Fig. 7. Architecture of Spatio-Temporal [13]

References

[1] W. Lee, "South Korean ai developer shuts down chatbot following privacy violation probe.", MLex Market Insight, 13 Jan. 2021, Retrieved 18. Nov. 2021, https://mlexmarketinsight.com/news-hub/editors-picks/area-of-expertise/data-privacy-and-security/south-korean-ai-developer-shuts-down-chatbot-following-privacy-violation-probe

[2] N. Statt, "Amazon sent 1,700 Alexa voice recordings to the wrong user following data request", THE VERGE, 20 Dec. 2018, Retrieved 18. Nov. 2021, https://www.theverge.com/2018/12/20/18150531/amazon-alexa-voice-recordings-wrong-user-gdpr-privacy-ai

[3] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data", In Artificial intelligence and statistics, pp. 1273-1282, 2017.

[4] P. Vepakomma, O. Gupta, T. Swedish, and R. Raskar, "Split learning for health: Distributed deep learning without sharing raw patient data", Accepted to ICLR 2019 Workshop on AI for social good, 2018.

[5] B. McMahan, D. Ramage, and R. Scientists, "Federated Learning: Collaborative Machine Learning without Centralized Training Data", Google AI Blog, 6 Apr. 2017, Retrieved 18. Nov. 2021, https://ai.googleblog.com/2017/04/federated-learning-collaborative.html

[6] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition". Proceedings of the IEEE, 86(11), pp. 2278-2324, 1998.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", Advances in neural information processing systems, vol. 25, pp. 1097-1105, 2012.

[8] A. Khan, A. Eker, A. Chefranov, and H. Demirel, "White blood cell type identification using multi-layer convolutional features with an extreme-learning machine", Biomedical Signal Processing and Control, 69, 2021.

[9] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition", Int. Conference on Learning Representations, San Diego, CA, 2015.

[10] M. Hassan, "VGG16 - Convolutional Network for Classification and Detection", Neurohive, 24 Feb. 2021, Retrieved 18. Nov. 2021, https://neurohive.io/en/popular-networks/vgg16/

[11] J. Wang, J. Zhang, W. Bao, X. Zhu, B. Cao, and P. S. Yu, "Not just privacy: Improving performance of private deep learning in mobile cloud", In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery Data Mining, pp. 2407-2416, 2018.

[12] S. Abuadbba, K. Kim, M. Kim, C. Thapa, S. A. Camtepe, Y. Gao, H. Kim, and S. Nepal, "Can we use split learning on 1d cnn models for privacy preserving training?", In Proceedings of the 15th ACM Asia Conference on Computer and Communications Security, pp. 305-318, 2020.

[13] J. Kim, S. Park, S. Jung, and S. Yoo, "Spatio-temporal split learning", In 2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks-Supplemental Volume (DSN-S), pp. 11-12, 2021.