The Simulation Study on the Effect of Transmission Error in Split Computing Approach

Jaewook Lee*, Keunsoo Ko†, Haneul Ko‡
*The Department of Information and Communication Engineering, Pukyong National University, Busan, Korea
E-mail: jlee0315@pknu.ac.kr
†The Department of Artificial Intelligence, Catholic University of Korea, Bucheon, Korea
E-mail: ksko@catholic.ac.kr
‡ The Department of Electronic Engineering, Kyung Hee University, Yongin, Korea
E-mail: heko@khu.ac.kr

Abstract—**The split computation service has gained wide attention to reduce the artificial intelligence-based service completion time. In a split computation service, the mobile device and edge server need to exchange the intermediate data to cooperatively process the AI-based service (i.e., process the deep model in AI-based service). In this case, the performance of split computation (e.g., classification accuracy) is degraded due to the transmission error. However, most of the works did not consider the transmission error effect of the split computation service. In this paper, we conduct various simulations to analyze the effect of transmission error on the split computation service. From the simulation results, we find some general trends that the later deep layer and convolution layer have better error-tolerant capability.**

I. Introduction

With the remarkable growth of artificial intelligence (AI), traditional service applications in various fields are replaced by AI-application that are generally implemented by the deep neural network (DNN) [1]. Also, DNN in applications becomes deeper and deeper to improve the service quality and thus the AI application needs more computation power to the computation node (e.g., edge server, mobile device, and cloud server) [1]. Due to this trend, a higher computation burden is taken into the mobile devices having low computation power, and thus the service computation time (i.e., inference time) is prolonged. Especially, this problem is harmful to real-time service (e.g., AR navigation service).

To mitigate this problem, the split computation service has gained wide attention, In the split computation service, the deep model is split into two sub-deep models (i.e., a former deep model and later deep model), a former deep model and a later deep model are provisioned to the mobile device and the edge server, respectively [1]. Then, the mobile device processes the former deep model by feeding the input data and transmits the output data of the former model (i.e., intermediate data) to the edge server via a wireless network. After that, the edge server processes the later deep model with the received intermediate data and returns back the processing results to the mobile device. Since the mobile device processes the part of the deep model (i.e., the former deep model), the computation burden at the mobile device can be significantly reduced. Also, since the later deep model is processed by the edge server having higher computation power, the inference time of the deep model can be significantly reduced. Owing to these advantages, the split computation service is considered as one of the represented services of the next-generation network (e.g., private 5G/6G mobile network) [2], [3].

To optimize the performance of split computation service, lots of research have been conducted [1], [4], [5] and most of the works focused on the split layer determination to minimize the service completion time [1], [4]. Kang *et al.* proposed Neurosugeon to select the best-split point optimizing for the service completion time or the energy consumption of mobile devices. On the other hand, Chen *et al.* proposed dynamic split points selection scheme to minimize the service completion time. This scheme splits the deep model into two or three sub-models based on the determined split points. However, these works did not consider the transmission error. Since the intermediate data is transmitted from the mobile device to the edge server via a wireless network, a transmission error occurs, and thus the edge server cannot always receive the original intermediate data from the mobile device. Due to this transmission error, the performance of the split computation service (e.g., accuracy) is worsened. To handle this problem, Wang and Zhang proposed NeuroMessenger to reduce communication overhead while guaranteeing the target service quality (i.e., service accuracy). In NeuroMessemnger, pruning deep layers and coding of the intermediate data, and determination on the number of retransmission are proposed to make the error tolerance split computation service. However, these works did not determine the optimal split point to reduce the service completion time while guaranteeing the target service quality when the transmission error occurs. To this end, the relationship between the transmission error and split points should be first researched.

In this paper, we conduct various simulations to analyze the relationship between the transmission error and split points (i.e., the effect of transmission error on the split computation service). The simulation results provide insight into the error-tolerance split point selection scheme and we describe this scheme as future works.

979-8-3503-1327-7/23/$31.00 ©2023 IEEE

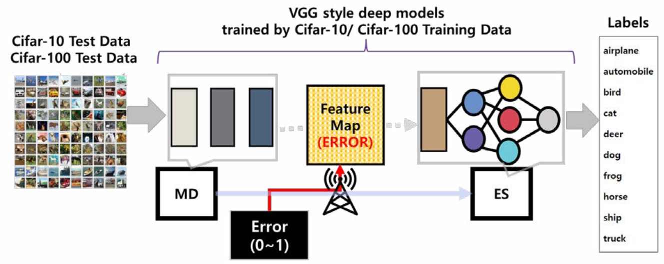


Fig. 1. Simulation environment

II. Effect of Transmission Error to the Split Computation Service

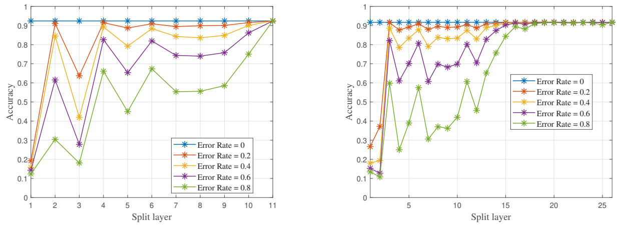
In this Section, we describe the simulation environment and results to analyze the effect of transmission error on the split computation service.

A. Simulation Environment

For performance evaluation, we conduct various simulations with different split points and different transmission error rates. Figure 1 represents our simulation environment. In this environment, first, we prepare the VGG-based deep model with a different number of layers [6]. The detailed deep model information is described in Table I. The deep model consists of layer blocks including convolution layers and one pooling layer and 3 dense layers. The layer blocks and 3 dense layers are sequentially connected. For instance, the VGG9 models consist of 5 layer blocks and 3 dense layers. In the layer block, convolution layers with the same number of channels and one pooling layer are sequentially connected. In the last layer block, 2 convolution layers with 512 channels and one pooling layer are sequentially connected. The deep models are trained by Cifar-10 and Cifar-100 datasets. Cifar-10 and Cifar-100 datasets provide 60000 data for 10 and 100 classes, respectively. With the trained deep models, we make the split models (i.e., a former deep model and a later deep model) based on all layers. Then, we assume two transmission error models (i.e., a random error and a burst error) [5]. In the random error model, the values in the intermediate data (i.e., feature map) are randomly set to 0 according to the transmission error. On the other hand, the near values in the intermediate data (i.e., feature map) are set to 0 an in the burst error model. During the simulation, we use 10000 data in Cifar-10 or Cifar-100 dataset [7] and obtain the average classification accuracy.

TABLE I
Deep Model Information. (A): The number of layers. (B) The number of convolution layers in each layer block. (C) The number of filters of the convolution layer in each layer block. (D) The number of pooling layers in each layer block. (E) The number of dense layers.

Model	(A)	(B)	(C)	(D)	(E)
$VGG6$	10	[1,1,1]	[64,256,512]	[1,1,1]	3
$VGG9$	16	[1,1,1,1,2]	[64,128,256,512,512]	[1,1,1,1,1]	3
$VGG11$	18	[1,1,2,2,2]	[64,128,256,512,512]	[1,1,1,1,1]	3
$VGG13$	20	[2,2,2,2,2]	[64,128,256,512,512]	[1,1,1,1,1]	3
$VGG16$	23	[2,2,3,3,3]	[64,128,256,512,512]	[1,1,1,1,1]	3
$VGG19$	26	[2,2,4,4,4]	[64,128,256,512,512]	[1,1,1,1,1]	3



(a)

(b)

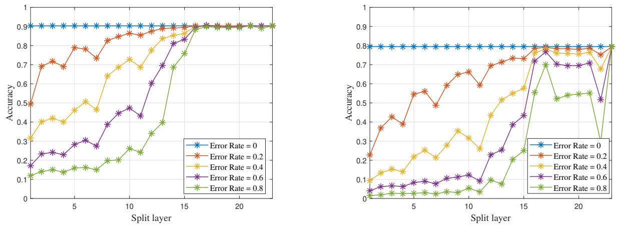
Fig. 2. Effect of Deep Model Architecture. (a) VGG6 model with Cifar-10. (b) VGG19 model with Cifar-10.

B. Simulation Results

1) Effect of Deep Model Architecture: Figures 2 show the effect of the deep model architecture. Especially, Figures 2 (a) and (b) represent the inference accuracy of VGG6 and VGG19 models according to the split layers and error rates. In this simulation, deep models are trained by the Cifar-10 dataset and a random error model is applied. From this result, when the error rate increases, the inference accuracy is degraded because the edge server processes the later deep model by the intermediate data with more error information. Meanwhile, we can find the trend that the accuracy is less degraded when the split point is determined as the later deep layer. This is because the important feature from the input image for classification is extracted from the former layers and the later layers only conduct the classification operation with the extracted feature.

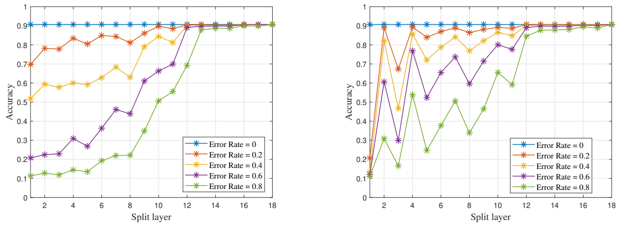
2) Effect of Dataset Type: Figures 3 (a) and (b) represent the inference accuracy of the VGG16 model trained by Cifar-10 and Cifar-100 datasets, respectively, when a burst error model is applied. The number of classes in the Cifar-100 dataset (i.e., 100 classes) is larger than that of the Cifar-10 dataset(i.e., 10 classes), i.e., classification of the Cifar-100 dataset is a more complex task than the Cifar-10 dataset. Figure 3 shows that the VGG16 model for the Cifar-10 dataset has more error-tolerance layer than the VGG16 model for the Cifar-100 dataset. This is because the VGG16 model is not sufficiently deep for the complex classification task (i.e., classifying the Cifar-100 dataset) but enough deep for the simple classification task (i.e., classifying the Cifar-100 dataset).

3) Effect of error model: Figures 4 show the effect of the transmission error model. Figures 4 (a) and (b) represent the inference accuracy of the VGG11 models trained by the Cifar-10 dataset when a burst error model and a random error model are applied, respectively. Generally, the burst error model affects a more negative effect to the split computation than the random error. In the burst error, the blank hole in the intermediate data occurs and thus the deep layers hardly reconstruct this blank hole with other information. On the other hand, in the error model, the error is sparsely applied to



(a) (b)

Fig. 3. Effect of the dataset. (a) VGG16 model with Cifar-10. (b) VGG16 model with Cifar-100.



(a) (b)

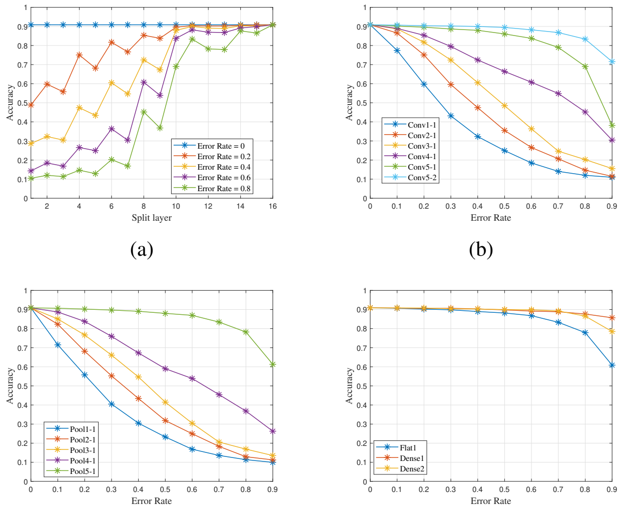
Fig. 4. Effect of the error model. (a) VGG11 model with Cifar-10 at burst error model. (b) VGG11 model with Cifar-10 at random error model.

the intermediate data and this information with error can be constructed by the deep layers with near information.

4) Effect of split layer: Figure 5 shows the effect of the split layer. In Figures 5, Figure 5 (a) represents the inference accuracy of all split layers in the VGG9 model. On the other hand, Figures 5 (a), (b), and (c) show the inference accuracy when the convolution layers, the pooling layers, and the dense layers are selected as the split layer. In this simulation, the Cifar-10 dataset is used for training the VGG9 model. Also, a burst error model is applied for this simulation. From Figure 5 (a), when the convolution layer is determined as a split point, i.e., the intermediate data after the convolution layer is transmitted to the edge server, the split computation service has a better error-tolerance feature than when the pooling layer is a split point. This is because the pooling layer makes the intermediate data with only important information in the results of the previous layer (i.e., the intermediate data of the convolution layer). Thus, although the pooling layer provides a small size of intermediate data, this intermediate data is more critical to transmission error.

III. Conclusion

In this paper, we conducted various simulations to analyze the transmission error effect on the split computation service. From the simulation results, some general characteristics have been found. First, when the later deep layer is selected to the split point, the split computation service has more error-tolerant capability. Also, when the convolution layer is selected to the split point, better error tolerance can be achieved. However, when the split point is selected to a later split point or convolution layer, the higher computation burden and transmission burden to the mobile device are inevitable and thus it prolongs the service completion time. Therefore, in future work, we will propose the performance-guaranteed split point (PGPS) selection scheme that can guarantee the target accuracy and service completion time while minimizing the computation burden of mobile devices.



(a) (b)

(c) (d)

Fig. 5. Effect of the error model. (a) All layers in the VGG9 model with Cifar-10 at burst error model. (b) convolution layers in the VGG9 model. (c) pooling layers in the VGG9 model. (d) dense layers in the VGG9 model.

Acknowledgement

This work was supported by the Pukyong National University Research Fund in 2023 (202303830001).

References

[1] Y. Kang *et al.*, "Neurosurgeon: Collaborative Intelligence Between the Cloud and Mobile Edge," in *Proc. ACM ASPLOS 2017*, Apr. 2017.

[2] "The Next Hyper-Connected Experience for All," [Online]. Available: https://news.samsung.com/global/samsungs-6g-white-paper-lays-out-the-companys-vision-for-the-next-generation-of-communications-technology

[3] 3GPP Technical Requirement (TR) 22.874, Study on Traffic Characteristics and Performance Requirements for AI/ML Model Transfer, v18.2.0, Dec. 2021.

[4] J. Chen *et al.*., "Accelerating DNN Inference by Edge-Cloud Collaboration," in *Proc. IEEE IPCCC 2021*, Oct. 2021.

[5] S. Wang, and X. Zhang, "NeuroMessenger: Towards Error Tolerant Distributed Machine Learning Over Edge Networks," in *Proc. IEEE INFOCOM 2022*, May 2022.

[6] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *Proc. ICLR 2015*, May 2015.

[7] A. Krizhevsky, "Learning multiple layers of features from tiny images," M.S. Thesis, Dept. Comp. Sci., Univ. Toronto, Toronto, Canada, Apr. 2009.