A Federated Learning Monitoring Tool for Self-Driving Car Simulation (Student Abstract)

Taejoon Lee, Hyunsu Mun, Youngseok Lee

Dept. of Computer Science and Engineering, Chungnam National University
taejoonlee@o.cnu.ac.kr, munhyunsu@cnu.ac.kr, lee@cnu.ac.kr

Abstract

We propose CARLA-FLMon, which can monitor the progress of running federated learning (FL) training in the open-source autonomous driving simulation software, CARLA. The purpose of CARLA-FLMon is to visually present the status and results of federated learning training, and to provide an extensible FL training environment with which FL training can be performed repeatedly with updated learning strategies through analysis. With CARLA-FLMon, we can determine what factors have positive or negative influences on learning by visualizing training data. Then, we can optimize the parameters of the FL training model to improve the accuracy of FL. With preliminary experiments of CARLA-FLMon on lane recognition, we demonstrate that CARLA-FLMon can increase the overall accuracy from 80.33% to 93.82% by identifying lowly-contributing clients and excluding them.

Introduction

Federated learning (FL) is important to keep the user privacy by putting data on users' device while conducting machine learning training jobs in a distributed computing environment (Li et al. 2020). FL has been studied and applied in diverse areas such as health care, medical machine learning, financial user classification problems. Recently, FL is being studied in autonomous driving simulation such as CARLA (Dosovitskiy et al. 2017), because it is not easy to collect private driving image or moving picture data from large numbers of cars in the realistic environment.

It is essential to ensure the performance of safety functions such as lane recognition and obstacle detection in autonomous driving for accident prevention. However, it is highly risky and expensive to conduct experiments with cars in a real environment. Through autonomous driving simulation, we can achieve comparable experiments, and verify the machine learning algorithms before applying the machine learning model to actual vehicles.

Federated learning is an attractive way to conduct realistic experiments in virtual autonomous driving environment, because clients participate in learning while avoiding personal information leakage by sending only learning models, not data from clients. To improve the performance of FL models, we need to observe parameters or configuration of FL training, such as learning situation, data distribution, and contribution to learning per client. In particular, as the number of FL clients increases, we need an effective simulation tool that can collect, monitor and analyze the FL data from clients and a server.

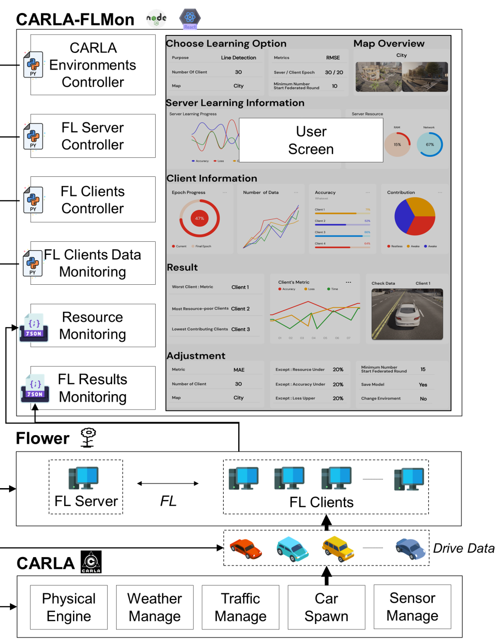

Figure 1: CARLA-FLMon overview

HFLens (Li et al. 2021) provides a visual method for finding abnormal clients that could not be identified in existing federated learning methods. FEDn (Ekmefjord et al. 2022) provides visualization tools as well as functions such as data pipelines, heterogeneous federated learning, and custom learning. Since the aforementioned FL visualization studies are designed to perform general purposes, they only generate the results of federated learning which is not convenient to integrate with a self-driving simulation software and to understand the insights of simulation results. Therefore, we propose CARLA-FLMon, a federated learning monitoring tool that can collect, monitor, analyze FL training data in CARLA environments.

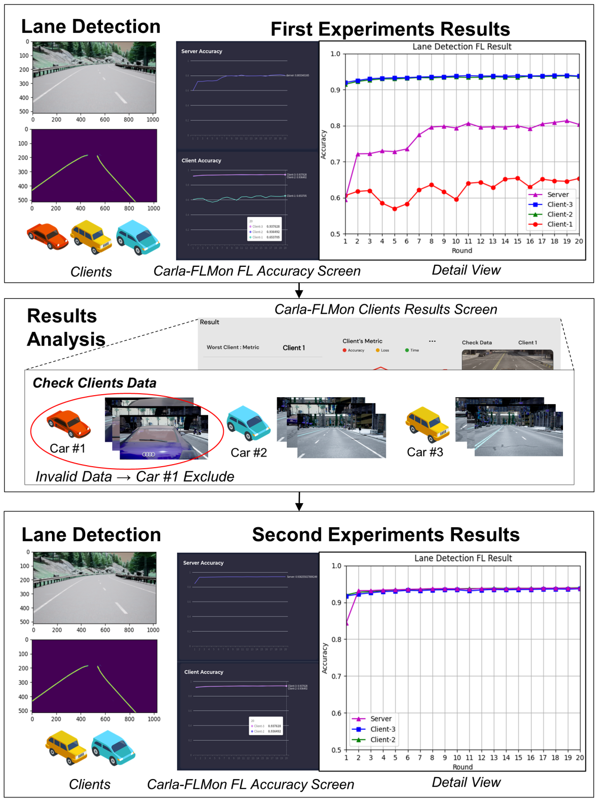

Figure 2: Lane detection experiments with CARLA-FLMon

CARLA-FLMon

Architecture: Assuming the federated learning framework called Flower (Beutel et al. 2020) on CARLA, we design the FL monitoring tool, CARLA-FLMon, as shown in Fig. 1. We present FL control and monitor modules for CARLA and implement the real-time view of monitoring FL data in CARLA. CARLA-FLMon consists of environments controller, learning option controller, FL strategy controller, and client data monitoring functions. Through the FL control module, we call CARLA simulation environment variables and learning-related settings in Python API and change them, and with the FL monitor module, we display FL simulation data in the web browser. In addition, CARLA-FLMon shows client information regarding resource utilization.

In CARLA-FLMon, we first set experimental environments, learning strategies, and model parameters. Through CARLA's API, we configure the environment parameters and start the data collection process. After collecting data, we set the learning strategy and model parameters through Flower's API, and carry out the learning procedure. Finally, we visualize the accuracy of each FL training round and client-specific learning results. We conduct lane recognition experiments to examine the usefulness of CARLA-FLMon.

Preliminary Experiment: For preliminary experiments, we examine a lane-aware model (Theers and Singh 2020) with CARLA-FLMon. Fig. 2 shows two experiment runs and the results. We select a sunny weather and a city map for the simulation environment, configure the FedAvg strategy for the FL server, 20 global rounds, and 10 rounds for FL clients. With CARLA-FLMon, we send an experimental configuration to CARLA and collect driving data to be used for lane recognition. We can evaluate the amount of data per FL client and the effective data ratio with a visualization function with CARLA-FLMon. After collecting driving data, we start federated learning with CARLA-FLMon. In CARLA-FLMon, a user can observe the learning progresses such as server and client model accuracy, loss, and epoch in real time. In the first simulation run, we found that the local model accuracy of client 1 is lower than client 2 and 3. With CARLA-FLMon's data viewer, we look into the reason why client 1 has low local model accuracy, finding that client 1 has a lot of invalid data. After detecting the FL performance degradation, we conduct the next simulation run, resulting in improving the accuracy of the FL global model except for client 1. When excluding client 1, we achieve the accuracy of the global model from 80.33% to 93.82%.

Conclusion and Future Work

In this work, we present CARLA-FLMon that can collect, monitor, analyze, and configure the FL parameters for self-driving car simulation software, CARLA. With the preliminary experiment on lane recognition, we demonstrate that the performance of FL global model can be improved by diagnosing the lowly-contributing FL clients. For the future work, we need to implement and extend FL functions such as resource utilization, clients data analysis, and driving data viewer. In addition, we plan to extend the distributed CARLA simulation environment and enhanced learning functions, such as object recognition and safe driving combined with traffic management and traffic light manipulation in CARLA simulations.

References

Beutel, D. J.; Topal, T.; Mathur, A.; Qiu, X.; Parcollet, T.; and Lane, N. D. 2020. Flower: A Friendly Federated Learning Research Framework. arXiv:2007.14390.

Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; and Koltun, V. 2017. CARLA: An Open Urban Driving Simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, 1–16. Mountain View, Calif.: PMLR.

Ekmefjord, M.; Ait-Mlouk, A.; Alawadi, S.; Åkesson, M.; Stoyanova, D.; Spjuth, O.; Toor, S.; and Hellander, A. 2022. Scalable federated machine learning with FEDn. arXiv:2103.00148.

Li, Q.; Wei, X.; Lin, H.; Liu, Y.; Chen, T.; and Ma, X. 2021. Inspecting the Running Process of Horizontal Federated Learning via Visual Analytics. *IEEE Transactions on Visualization and Computer Graphics*, 1–1.

Li, T.; Sahu, A. K.; Talwalkar, A.; and Smith, V. 2020. Federated Learning: Challenges, Methods, and Future Directions. *IEEE Signal Processing Magazine*, 37(3): 50–60.

Theers, M.; and Singh, M. 2020. Algorithms for Automated Driving. https://thomasfermi.github.io/Algorithms-for-Automated-Driving/Introduction/intro.html. Accessed: 2022-09-15.