Research Diary

Distributed and Federated Learning

Dr Shashank Vatedka
Assistant Professor
Department of Electrical Engineering

KID: 20220207

"Any sufficiently advanced technology is indistinguishable from magic" - Arthur C Clarke

If you watched cartoons during the 90's, then you would remember a futuristic sitcom called the Jetsons, which had flying cars, jetpacks, autonomous navigation, and robots doing household chores. While we are far from being in this technological utopia, the future of navigation is certainly heading towards connecting vehicles, autonomous navigation and UAV-based systems. Achieving these would require significant progress in various fields ranging from the design of specialized sensors and hardware, AI/ML, and communication systems to engines, aerodynamics and design. There has been a lot of exciting research in recent years, particularly by groups at TiHAN and IITH, towards making these dreams a reality.

Let us take machine learning, which has emerged as one of the hottest topics in recent years. There have been great strides made in the areas of object detection, computer vision and natural language processing. However, there are new challenges that are unique to the problem of autonomous and connected vehicles. If we are to make transportation "smarter," vehicles have to be fitted with a variety of sensors, including cameras, RADAR, LIDAR, GPS, etc., and the data generated by these are used to train deep learning algorithms for navigation, route planning, traffic control and infotainment services. However, each car can now generate several gigabytes of data in just a few hours, which is a lot to communicate and process and can contain sensitive information that needs to be kept private.

A majority of the work on machine learning today assumes that all the training data is available at one place (aka a centralized server), which is not true in the setting above. The training data is actually generated by various devices (or "clients") that are distributed, whereas the machine learning model is to be obtained at a centralized server which is connected to each of the clients typically through noisy/communication-constrained links. The need of the hour is to design algorithms for learning reliable models at the centralized server in a setting where client data is to be kept private, and the amount of communication between the clients and the server is limited. To solve this problem, there has been a flurry of recent work in emerging fields of distributed and federated learning [1,2]. This has led to several new problems and cross-fertilization of ideas in various areas, including distributed optimization, machine learning, cryptography, security, information theory, and statistics, just to name a few.

To give a very brief overview of the main challenges, contrast this with the standard supervised machine learning problem where we have labelled data, and the goal is to typically train a model such as a neural network using this data. In the setting mentioned previously, the data is instead distributed across a large number of clients. A naive solution would be to ask the clients to send all their data to the server and then use this to train the model.

However, this approach would be extremely communication-intensive, insecure, and violate privacy requirements.

The fundamental idea behind federated learning is that clients can instead train "local" models (say, neural networks) using their personal data and share this with the server, who now somehow aggregates the models and sends them back to the clients.

This is a very simplistic description that does not capture the nuances involved in the process, but the typical cycle used in federated learning is as follows:

1. The server selects a subset of the clients and broadcasts a coarse/current "global" model to these clients,
2. The clients use part of their personal data and the model shared by the server to train separate updated "local" models,
3. The local models are then compressed and sent to the server
4. The server aggregates these local models to update the current model
5. Go back to step 1, and refine the model

There are a number of challenges involved in this process: the data of the various users can be very heterogeneous, the local models must not leak information about the user data, clients are not always available, and the communication links between the clients and the server can be very noisy, the systems are susceptible to attacks, and so on. In fact, each step of the process gives rise to very interesting and challenging fundamental problems.

Our group has been recently studying problems of compression and communication-efficient aggregation in the above context. However, the field has a very rich set of open problems, and we can only scratch the surface.

References:

[1] Kairouz, Peter, et al. "Advances and open problems in federated learning." Foundations and Trends® in Machine Learning 14.1–2 (2021): 1-210.

[2] Elbir, Ahmet M., Burak Soner, and Sinem Coleri. "Federated learning in vehicular networks." arXiv preprint arXiv:2006.01412 (2020).

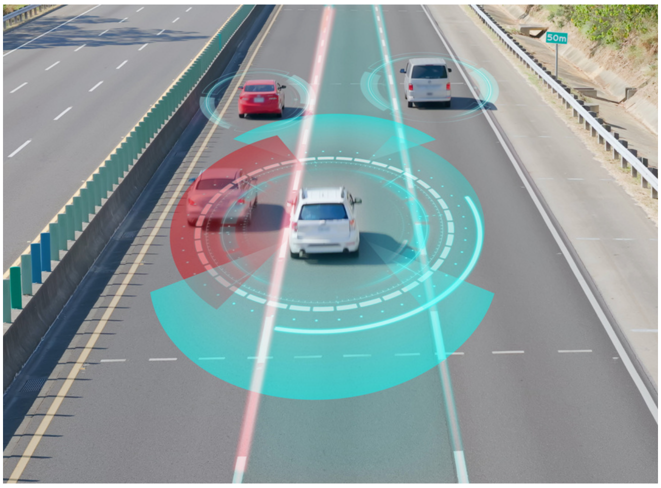

Figure 25: Representation Image of Lidar Technology