CHAPTER 4

Horizontal Federated Learning

In this chapter, we introduce horizontal federated learning (HFL), covering the concept, architecture, application examples, and related works, as well as open research challenges.

4.1 THE DEFINITION OF HFL

HFL, a.k.a. sample-partitioned federated learning, or example-partitioned federated learning [Kairouz et al., 2019], can be applied in scenarios in which datasets at different sites share overlapping feature space but differ in sample space, as illustrated in Figure 4.1. It resembles the situation that data is horizontally partitioned inside a tabular view. In fact, the word "horizontal" comes from the term "horizontal partition," which is widely used in the context of the traditional tabular view of a database (e.g., rows of a table are horizontally partitioned into different groups and each row contains complete data features). For example, two regional banks may have very different user groups from their respective regions, and the intersection set of their users is very small. However, their business models are very similar. Hence, the feature spaces of their datasets are the same. Formally, we summarize the conditions for HFL as:

$$\mathcal{X}_i = \mathcal{X}_j, \quad \mathcal{Y}_i = \mathcal{Y}_j, \quad I_i \neq I_j, \quad \forall \mathcal{D}_i, \mathcal{D}_j, i \neq j, \tag{4.1}$$

where the data feature space and label space pair of the two parties, i.e., $(\mathcal{X}_i, \mathcal{Y}_i)$ and $(\mathcal{X}_j, \mathcal{Y}_j)$, are assumed to be the same, whereas the user identifiers I_i and I_j are assumed to be different; $\mathcal{D}_i$ and $\mathcal{D}_j$ denote the datasets of the ith party and the jth party, respectively.

Security of an HFL system. An HFL system typically assumes honest participants and security against an honest-but-curious server [Phong et al., 2018, Bonawitz et al., 2017]. That is, only the server can compromise the user privacy and data security of the participants.

Shokri and Shmatikov [2015] proposed a collaborative deep learning (DL) scheme where participants train models independently and share only subsets of model parameter updates, which is a special form of HFL. In 2016, researchers at Google proposed an HFL-based solution for Android smartphone model updates [McMahan et al., 2016a]. In this framework, a single Android smartphone updates the model parameters locally and uploads the model parameters to the Android cloud, thus jointly training the federated model together with other Android smartphones.

A secure aggregation scheme for protecting the privacy of the user model updates under this federated learning framework was introduced in Bonawitz et al. [2017]. More recently,

Q. Yang et al., *Federated Learning*

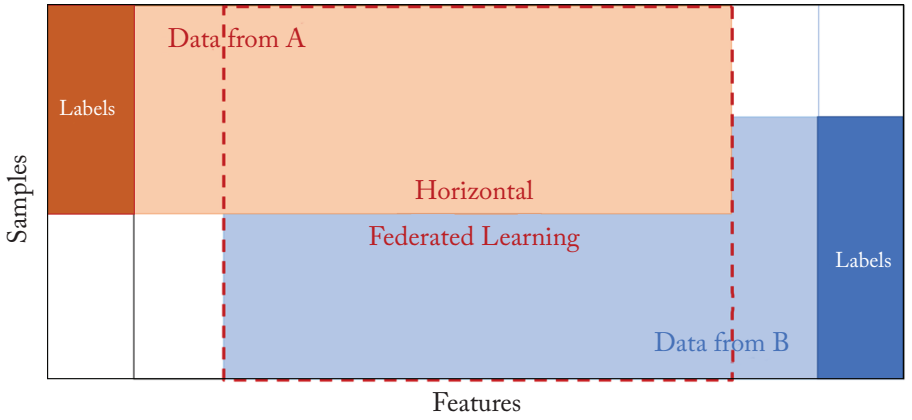

Figure 4.1: Illustration of HFL, a.k.a. sample-partitioned federated learning [Yang et al., 2019].

Phong et al. [2018] applied additively homomorphic encryption for model parameter aggregation to provide security against an untrustworthy central server.

In Smith et al. [2017], a multi-task style federated learning system is proposed to allow multiple sites to complete different tasks, while sharing knowledge and preserving security. Their proposed multi-task learning model can also address the issues of high communication costs, stragglers, and fault tolerance.

In McMahan et al. [2016a], the authors proposed a secure client-server structure where the federated learning system partitions data by users, and allows models built at client devices to collaborate at the server site to build a global federated model. The process of model building ensures that there is no data leakage. Likewise, in Konecný et al. [2016b], the authors proposed methods to reduce the communication cost to facilitate the training of federated models based on data distributed over mobile clients. More recently, a compression approach called Deep Gradient Compression [Lin et al., 2018] is proposed to greatly reduce the communication bandwidth in large-scale distributed model training.

Security proof has been provided in these works. Recently, another security model considering malicious user [Hitaj et al., 2017] is also proposed, posing additional privacy challenges. At the end of federated training, the aggregated model and the entire model parameters are exposed to all participants.

4.2 ARCHITECTURE OF HFL

In this section, we describe two popular architectures for HFL systems, namely the client-server architecture and the peer-to-peer architecture.