Architecture-Based FedAvg for Vertical Federated Learning

Bruno Casella
bruno.casella@unito.it
University of Turin
Turin, Pidmont, Italy

Samuele Fonio
samuele.fonio@unito.it
University of Turin
Turin, Pidmont, Italy

ABSTRACT

Federated Learning (FL) has emerged as a promising solution to address privacy concerns by collaboratively training Deep Learning (DL) models across distributed parties. This work proposes an architecture-based aggregation strategy in Vertical FL, where parties hold data with different attributes but shared instances. Our approach leverages the identical architectural parts, i.e. neural network layers, of different models to selectively aggregate weights, which is particularly relevant when collaborating with institutions holding different types of datasets, i.e., image, text, or tabular datasets. In a scenario where two entities train DL models, such as a Convolutional Neural Network (CNN) and a Multi-Layer Perceptron (MLP), our strategy computes the average only for architecturally identical segments. This preserves data-specific features learned from demographic and clinical data. We tested our approach on two clinical datasets, i.e., the COVID-CXR dataset and the ADNI study. Results show that our method achieves comparable results with the centralized scenario, in which all the data are collected in a single data lake, and benefits from FL generalizability. In particular, compared to the non-federated models, our proposed proof-of-concept model exhibits a slight performance loss on the COVID-CXR dataset (less than 8%), but outperforms ADNI models by up to 12%. Moreover, communication costs between training rounds are minimized by exchanging only the dense layer parameters.

CCS CONCEPTS

• **Computing methodologies → Multi-task learning**; **Distributed artificial intelligence**; • **Applied computing → Life and medical sciences**; • **Security and privacy → Human and societal aspects of security and privacy**.

KEYWORDS

Federated Learning, Vertical Federated Learning, Computer Vision, Deep Learning, Personalized Federated Learning.

ACM Reference Format:
Bruno Casella and Samuele Fonio. 2023. Architecture-Based FedAvg for Vertical Federated Learning. In *2023 IEEE/ACM 16th International Conference on Utility and Cloud Computing (UCC '23), December 4–7, 2023, Taormina (Messina), Italy*. ACM, New York, NY, USA, Article XXX, 6 pages. https://doi.org/10.1145/3603166.3632559

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
UCC '23, December 4–7, 2023, Taormina (Messina), Italy
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0234-1/23/12.
https://doi.org/10.1145/3603166.3632559

1 INTRODUCTION

Nowadays, Artificial Intelligence (AI) is increasingly being utilized in various domains. AI techniques require training on data available in a single data lake. However, data are often distributed among different institutions, and aggregating them is not always feasible due to privacy and security concerns. The data's distributed location makes the training model's deployment local, with specific concerns on the heterogeneity of hardware capacities, connections and, in our case, data privacy. In this sense, FL has emerged as a promising solution for collaborative machine learning across distributed parties while preserving data privacy, and it represents an interesting challenge for Computing Continuum [18]. In particular, FL leverages the computing continuum to enable collaborative, distributed model training across a range of connected devices, ensuring seamless and continuous intelligent learning. The key innovation brought by FL is to benefit from data that usually are only used locally. FL is typically divided into two categories:

- **Horizontal (HFL)** [13] Federated Learning, where the federation participants (clients) hold data with the same features or attributes but different instances or examples.
- **Vertical (VFL)** [12] Federated Learning, where the different clients hold data with the same instance but different features.

HFL and VFL adopt very different training techniques. Indeed, in HFL, the most well-known and most used aggregation algorithm is *FederatedAveraging* (FedAvg), where each client of the federation trains a local model and exchanges its parameters with a central server, which aggregates them and sends the resulting global model back to each client. Besides FedAvg, other FL algorithms have been proposed, such as FedCurv [20], and SCAFFOLD [7], in order to deal with non-IID data, which has been shown to represent a challenge for FL systems [2]. On the other hand, in VFL both data and models are kept local while intermediate results are exchanged among clients. These intermediate messages consist of learning representations of local data and their gradients [22, 24].

Our work aims to explore an innovative aggregation strategy for VFL that leverages model architecture as a key criterion for weight aggregation. We propose to average the network parameters only for the parts of the models that share the same architectural structure, thus preserving the specificity of features learned from demographic and clinical data. Basically, this results in applying the HFL principles only for the shared identical architectural layers.

This strategy offers several potential advantages. Firstly, it enables collaboration among parties that do not hold the same type of data, i.e., one client holding images and another holding tabular data, increasing the model's generalizability. Secondly, it reduces cost communication among parties by lowering the data size to be shared during the aggregation process. Indeed, a reduction in

communication costs during the training of our federated model is achieved by transferring only a smaller subset of layers of the local models, while typical HFL systems exchange all the layers.

VFL is often used in scenarios where different organizations or entities want to collaborate to train a model on data with different attributes without violating the privacy of the data itself. A typical example is the medical application. In healthcare, patient data can be highly heterogeneous. Different institutions may collect patient data with different attributes, such as age, gender, medical history, and test results. For this reason, in our experiments, we employed two medical datasets containing both images and clinical parameters, namely the COVID-CXR dataset [21] and the ADNI study [23].

The main contributions of our work are:

- we extend the principles of HFL to VFL by applying FedAvg only for the parts of the models that share the same architectural structure.
- We perform extensive experiments to provide a systematic study of FedAvg in a VFL setting, analyzing the performance from a learning and computational point of view.
- As a result of experimentation, we show that our proposed framework aligns with the centralized results achieved by relying only on a single input data type and allows for exploiting FL generalizability even when institutions have different input sources.
- We release the code to replicate our experiments.

The rest of the paper is organized as follows. Section 2 reviews the most recent related works. Section 3 presents our proposed method. Section 4 describes the experimental settings and discusses the results. In Section 5, conclusions are drawn.

2 RELATED WORKS

VFL has gained attention in the last few years, tackling the problem of decentralization both for examples and for the feature space. The goal is to create an aggregated architecture efficient on different data without the possibility of considering all of the examples together at training time.

The developed research has become very effective for cases in which privacy concerns are particularly challenging, and data can be very heterogeneous. For example, the healthcare domain can benefit consistently from this setting.

However, in the first place, privacy concerns have not been taken into account, enabling the development of centralized multi-modal architectures for data with heterogeneous feature space (for example, images and tabular data). In particular, the need for multi-modal architectures in the healthcare domain has brought many advancements in designing methods to accomplish specific tasks. In a recent work [9], the authors used the representations extracted by a CNN for training a second linear model that fits tabular data. This method has been outperformed by single architecture methods, which solve the problem of redundant representations. In fact, approaches involving a singular architecture connect the latent representation of image and tabular information before the last layer, overcoming the problem of redundant information [5, 8, 16, 17]. However, they exploit only linear relations between image representation and tabular data. This limitation was overcome using an MLP, which takes into consideration the non-linearities between different data involving the same patient, as done in some recent works [4, 11, 14].

All these approaches present architectures to accomplish a certain medical task with different types of data without considering possible privacy restrictions. Our approach aims at providing a framework where data are owned by institutions sensible to data privacy, which makes the task harder to complete. Indeed, few works have addressed this problem concerning the FL setting. Some FL works in the Internet of Things (IoT), and healthcare sectors are present in the literature. Most of these works design multi-modal architectures for solving specific tasks [1, 19, 25]. Moreover, the increasing proliferation of IoT applications requires a seamless interconnection of resources of edge and cloud devices, leading to the ecosystem of Computing Continuum. FL is a Computing Continuum strategy of Machine Learning applications relying on data coming from IoT and edge devices [15].

Another recent work [3] in the medical domain proposes MERGE, a multi-input NN leveraging multiple input sources, i.e., images and tabular data. The basic assumption of MERGE is that each federation participant has both data types, locally available and accessible. However, in a real federation, each client can hold various types of datasets, such as images, tabular features, or text reports. Our proposed method aims to overcome this FL limitation by aggregating only the identical architectural parts of the DL models. In addition, we focus on the interpretation of the aggregation, shedding light on the interpretation of the extracted features by different architectures on different data and the combination of the two.

3 METHOD

In this section, we present our approach for VFL with Architecture-Based Aggregation. Our method aims to address the unique challenges of collaborating on machine learning tasks when data sources possess heterogeneous attributes that refer to the same instance (e.g. an institution owning scans of a patient, while another institution owns his clinical parameters).

We consider a typical VFL scenario where two distinct institutions hold data related to the same set of instances but with different attributes. In particular, in our experiments, the first institution trains a CNN on image data, while the second party trains an MLP on tabular data. We selectively aggregate the weights of these models based on their architectural similarity. By design, and in order to be aggregated, the dense layers of the CNN adopted by the first organization are the same as the MLP of the second institution. Figure 1 shows how our proposed method works.

We introduce several aggregation strategies to explore the benefits of selective aggregation:

- **Full**: in this scenario, we perform weight aggregation across all the dense layers of the models.
- **Half**: in this scenario, we perform weight aggregation across half of the dense layers of the models. According to the layers we choose to combine, we can have different cases:
  - Aggregating the first half: we aggregate the weights of the first layers. This strategy focuses on capturing the models' initial feature extraction and processing stages.

Figure 1: A generic representation of our architecture-based FedAvg for VFL.

Aggregating the second half: we aggregate the weights of the last layers. This strategy focuses on capturing the last processing stages and the nal output decision-making.

In our experiments, we tested only the second case, i.e., we performed weight aggregation of only the last model layers. So, it is possible further to distinguish several subcases within the Half Aggregation approach to re ne our aggregation strategy. Speci cally, when averaging the last layers, the question arises regarding the weights of the rst layers. We investigated three di erent possibilities:

Random initialization : the rst layers of the models are randomly initialized. This approach allows for exploring the relationship between random weights and the features extracted in the initial layers. Moreover, it allows to study the in uence and interaction among random weights and subsequent layers.
Alignment with CNN dense layers : the weights of the rst layers are set to the corresponding values in the CNN. This strategy allows for emphasizing the feature extraction and initial processing stages of the CNN.
Alignment with MLP dense layers : the weights of the rst layers are set to the corresponding values in the MLP. This approach allows for emphasizing the feature extraction and initial processing stages of the MLP.

4 EXPERIMENTS AND RESULTS

Federation setup: the entities of our simulated federation are a server and two clients. For each federation round, each client executes one training, one validation, and one testing stage. Then, aggregation is performed according to one of the previously discussed strategies, and the resulting models are validated on the test data. All experiments are executed on a simulated federation deployed on a dedicated server with an Intel Xeon CPU (8 cores per CPU) and one Tesla T4 GPU. We release the code required to reproduce our experiments at the following link: OMITTED FOR ANONYMOUS SUBMISSION.

4.1 Datasets

We tested our aggregation strategy on two tasks:

Prognosis of COVID-19 disease from chest X-rays (CXR) data, using the COVID-CXR dataset [21]
Detection of Alzheimer's disease from neuroimaging data, using the ADNI study [23].

More details about the datasets are provided below.

COVID-CXR dataset. This dataset consists of both CXR and clinical parameters. Data were collected from six hospitals (A, B, ..., F) in emergency conditions during the rst COVID-19 outbreak in Northern Italy in collaboration with Centro Diagnostico Italiano and Bracco Imaging. A sample from each hospital is shown in Fig. 2

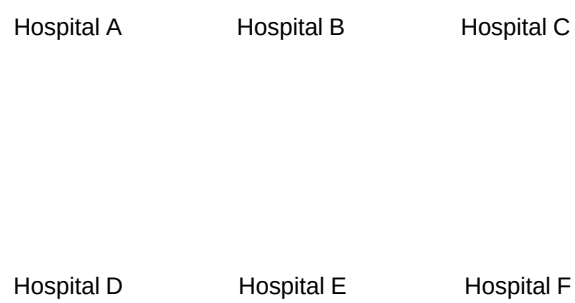

Figure 2: One sample coming from each of the six hospitals.

As clearly shown in Fig. 2, this dataset exhibits a clear feature distribution skew due to the di erent data collection procedures of the six hospitals. This leads to the well-known problem of non-iidness in FL [2, 10]. Each patient of the dataset is provided a CXR and some clinical parameters (namely age, sex, positivity at admission, temperature, days of fever, cough, di culty in breathing, WBC, RBC, CRP, glucose, LDH, INR, PaO2, PaCO2, pH, high blood pressure, diabetes, dementia, BPCO, cancer, CKD and respiratory failure). The statistics of the dataset are summarized in Table 1.

Table 1: Statistics of the CoViD-19 CXR dataset with class-balance percentages.

Hospital	Samples	Positives (%)	Negatives (%)
A	120	85 (70.83%)	35 (29.17%)
B	104	59 (56.73%)	45 (43.27%)
C	151	81 (53.64%)	70 (46.36%)
D	139	76 (54.68%)	63 (45.32%)
E	101	55 (54.46%)	46 (45.54%)
F	974	546(56.06%)	428 (43.94%)