A split-training approach to JoVe-FL

M. Hartmann[1], G. Danoy[12], M. Alswaitti[1], and P. Bouvry[12]

[1] SnT, University of Luxembourg
[2] FSTM/DCS, University of Luxembourg
{maria.hartmann, gregoire.danoy, mohammed.alswaitti, pascal.bouvry}@uni.lu

1 Introduction

Federated Learning is a relatively novel concept in the field of distributed machine learning, developed to allow the joint training of a common machine learning model by many participants without revealing their respective training data to each other. This is generally accomplished by having each participant separately train a local machine learning model, using only the dataset known to itself, then sharing the results of the training process with others in the form of model parameters or model outputs. These results are then utilised to obtain a global model that implicitly incorporates each participant's local information.
Since its inception in 2016 [1], great strides have been made in the field; however, much of the related research has been limited to the problem known as horizontal federated learning (HFL), where all participants possess samples from the same feature space. Significantly less research exists on the more complex scenario where participants may observe different features and train models with different architectures [2]. This type of problem is generally referred to as vertical federated learning (VFL).
Furthermore, application of such research has generally been focused on scenarios where privacy of data is the driving cause for keeping participants' datasets separate. In contrast, our research is targeted towards a scenario where the available communication channels between participants are severely constrained, thereby limiting the amount of information that can be shared. Federated learning is a natural solution approach for such application scenarios, but this requires somewhat different considerations than the privacy-focused scenario, e.g. with respect to robustness. A highly relevant example for such scenarios is that of autonomous satellite swarms, consisting of multiple miniature satellites working towards a common goal, with communication capacity limited by a strict energy budget [4]. An effective joint learning strategy in such a scenario would represent a significant leap forward on the path towards realising truly autonomous satellite swarms.

In some previous work [3], we introduced the Joint-embedding Vertical Federated Learning (JoVe-FL) framework as a first step towards developing a VFL scheme tailored to such applications. This article introduces a novel extension to JoVe-FL which consists in the introduction of separate training phases for the local embedding and prediction models, intended to encourage the learning of a joint embedding space while further reducing the amount of computation required.

2 The JoVe-FL framework

JoVe-FL is based on the idea of transforming the underlying vertical federated learning problem into a horizontal one. The framework is designed such that each client maintains a complete local model, allowing independent functioning of all participants outside of the joint training process, with the server acting in an auxiliary role only to facilitate model aggregation.

Each client C_i maintains a local model consisting of two parts: the embedding model ξ_i and the prediction model θ_i (see Figure 1 (a)). These are concatenated in the given order to obtain the complete local model $\xi_i \circ \theta_i$. Each individual embedding model may take any type of input, independent of the models of other clients. The only requirement w.r.t. the architecture is that all participating clients' embedding models must return an output vector of the same dimensions. In contrast, the architecture of prediction models is uniform across all clients. The purpose of this setup is to train all clients to map their differing feature spaces to a common embedding space by training the individual embedding models; on this embedding space the prediction models can be

trained in classical HFL fashion.
We have demonstrated in preliminary experiments that participants in this framework are indeed capable of learning such an embedding, and that doing so enables them to share information with each other, thus improving their performance compared to the no-communication setting. However, these experiments still showed room for improvement when comparing the results of the federated models to those trained on the centralised dataset; hence we experimentally consider possible modifications of the framework to improve its performance.

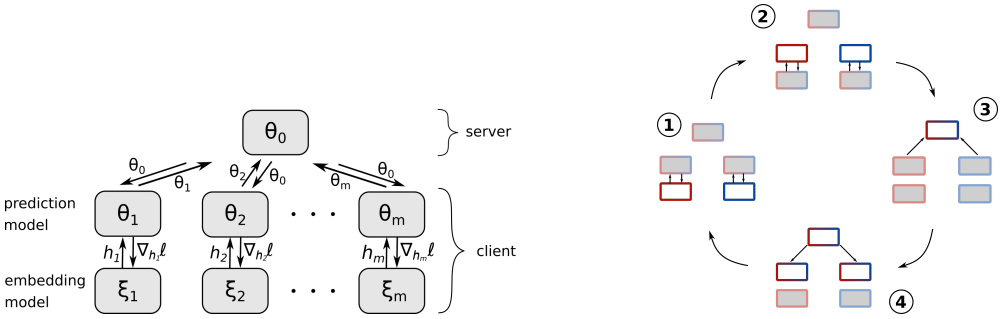

Fig. 1: (a) Architecture of the JoVe-FL framework, (b) Process with separate model training phases.

Embedding-prediction model training phases. The local model training is split into two phases, so that the prediction model and the embedding model are trained entirely separately - this process is also illustrated in Figure 1 (b). After the global aggregation phase (step 3 in Figure 1 (b)), once the local prediction model has been updated (step 4 in the figure), the client fixes the parameters of its local prediction model and trains only the embedding model for a certain number of steps (see step 1 in the figure). Then, the local prediction model is held fixed for a certain duration while the embedding model is trained (step 1 in the figure). Then these states are reversed: the parameters of the embedding model are frozen while the prediction model is trained (step 2 in the figure). The duration of each of the two training phases is varied across experiments.

3 Experiments

Experiments were carried out on the CIFAR10 dataset for image classification [5], with a vertically distributed dataset created by assigning different partial slices of each image to different clients. For each experimental configuration, we compare the performances of (1) the federated model under consideration, (2) the original federated model without separation of training phases, and (3) the same model without communication between clients as a lower bound. The models in (3) were trained with the same local scheme as the ones in (2), without the separation of training phases that is present in the modified federated learning scheme, as this is not expected to benefit the performance of non-federated clients.
Each experiment is repeated five times and is stopped upon convergence, i.e. once the gradient of the test loss falls below a certain threshold. The results of one such experiment are presented in Figure 2. For this experiment, the final top-1 accuracies achieved by the two clients were $(73.5, 84.4)$ and $(74.2, 84.7)$ for the modified and original scheme, respectively. Both present a clear improvement over the accuracies of $(71.4, 83.1)$ that were obtained without cooperation between clients. We observe a similar pattern across all but one experiment, showing that the performance of the modified JoVe-FL scheme outperforms the non-federated training. Indeed, in all experiments the final accuracy results accomplished by the modified learning scheme with lower computational effort nearly match those of the original version.

4 Conclusion

This article presented a possible modification of the JoVe-FL framework, designed to reduce the computational cost of the local training performed by participating clients. Experiments were

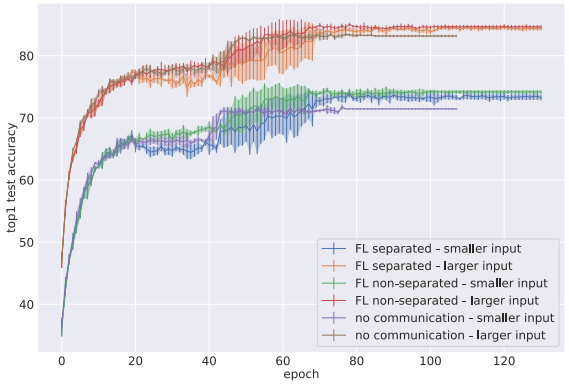

Fig. 2: Results for two clients with differently-sized datasets performing (1) federated learning with separate training phases, (2) federated learning without separate training phases, and (3) model training without communication.

carried out on the CIFAR10 image classification dataset, separated into partial views to obtain a vertical distribution. We conclude that the modified version of the framework is capable of producing results similar to the ones obtained by the original JoVe-FL scheme, while significantly reducing the amount of computation required in the local training phase.

Acknowledgements

This work is partially funded by the joint research programme UL/SnT–ILNAS on Technical Standardisation for Trustworthy ICT, Aerospace, and Construction. The experiments presented in this paper were carried out using the HPC facilities of the University of Luxembourg [6] – see https://hpc.uni.lu.

References

1. McMahan, H. B., Moore, E., Ramage D., Hampson, S., and y Arcas, B. A., Communication-Efficient Learning of Deep Networks from Decentralized Data. CoRR, Feb. 28, 2017. doi: 10.48550/arXiv.1602.05629.
2. Kairouz, P., et al., Advances and Open Problems in Federated Learning. CoRR, Available: http://arxiv.org/abs/1912.04977 (2021)
3. Hartmann, M., Danoy, G., Alswaitti, M., and Bouvry, P. JoVe-FL – A Joint-embedding Vertical Federated Learning Framework In Proc. of the 15th International Conference on Agents and Artificial Intelligence (ICAART 2023), Lisbon, Portugal. SCITEPRESS. (2023).
4. Arnold, S. S., Nuzzaci, R., and Gordon-Ross, A., Energy budgeting for cubesats with an integrated fpga. In 2012 IEEE Aerospace Conference, pages 1–14. (2012).
5. Krizhevsky, A., and Hinton, G. Learning Multiple Layers of Features from Tiny Images. Technical report, University of Toronto.
6. Varrette, S., Cartiaux, H., Peter, S., Kieffer, E., Valette, T., and Olloh, A. Management of an Academic HPC & Research Computing Facility: The ULHPC Experience 2.0. In Proc. of the 6th ACM High Performance Computing and Cluster Technologies Conf. (HPCCT 2022), Fuzhou, China. Association for Computing Machinery (ACM). (2022).