Supporting Efficient Workflow Deployment of Federated Learning Systems across the Computing Continuum

Cédric Prigent*, Gabriel Antoniu*, Alexandru Costan*, Loïc Cudennec†
*University of Rennes, Inria, CNRS, IRISA - Rennes, France
{cedric.prigent, gabriel.antoniu, alexandru.costan}@irisa.fr
†DGA Maîtrise de l'Information, Rennes, France
{loic.cudennec}@intradef.gouv.fr

***Abstract*—IoT devices produce ever growing amounts of data. Traditional cloud-based approaches for processing data are facing some limitations: bandwidth might become a bottleneck and sensitive data should not leave user devices as stated by data protection regulators such as GDPR. Federated Learning (FL) is a distributed Machine Learning paradigm aiming to collaboratively learn a shared model while considering privacy preservation. Clients do the training process locally with their private data while a central server updates the global model by aggregating local models. In the Computing Continuum context (edge-fog-cloud ecosystem), FL raises several challenges such as supporting very heterogeneous devices and optimizing massively distributed applications. We propose a workflow to better support and optimize FL systems across the Computing Continuum by relying on formal descriptions of the underlying infrastructure, hyperparameter optimization and model retraining in case of performance degradation. We motivate our approach by providing preliminary results using a human activity recognition dataset showing the importance of hyperparameter optimization and model retraining in the FL scenario.**

***Index Terms*—Computing Continuum, Federated Learning, Workflow, Hyperparameter optimization**

I. Context

With the ever growing amounts of data generated by IoT devices, traditional techniques such as cloud-based processing come to some limitations. Additionally with new data protection regulations such as GDPR, sensitive data can not be processed on such infrastructure. Federated Learning is a Machine Learning (ML) paradigm focusing on privacy preservation in which several clients collaboratively train a global model without exchanging their private data. In this setup, clients train the model locally with their private data while only model weights are aggregated to a central server to update the global model, bringing the benefit of privacy preservation and reduced bandwidth pressure.

The Computing Continuum [1] is an ecosystem in which data flows operate over edge, fog and cloud resources. Deploying applications in this massively distributed environment requires to support very heterogeneous devices with limited bandwidth and computing resources. FL fits well to this ecosystem where the cloud takes the role of a central server and clients are very heterogeneous edge devices. However, applying FL in this context, where many conflicting objectives (*e.g.* accuracy of the model, battery usage, bandwidth utilization, data security) need to be optimized brings several challenges.

II. Problem Statement

Many tools and frameworks were proposed to implement or deploy FL systems [2]. However they do not provide mechanisms to describe or interact with IoT objects relying on different architectures [3]. This results in the difficult integration of such heterogeneous devices.

FL introduces several hyperparameters such as federated optimization strategy, client learning rate or number of local epochs to process before aggregating the local models. This increases complexity regarding hyper parameter optimization. Only few studies focus on hyperparameter optimization in the FL context [4] and they do not consider model retraining in case of performance degradation due to data drift (unexpected change in data distribution).

Applying FL across the Computing Continuum involves working with heterogeneous devices with very different usage resulting in very different data distributions (Non-IID data). Diverging data distributions might end up in good global performance of the model while underperforming on specific clients.

Based on these statements, several questions arise:

1) **How to support frequent deployments and updates in heterogeneous and highly distributed environments?**
2) **How to efficiently combine hyperparameter optimization and model retraining in case of data drift?**
3) **How to deal with clients with very different data distribution?**

III. Our Approach

To better support the optimization and deployment of FL workloads, we propose a workflow which can be divided in 3 main steps.

Resource Description and Exploration Strategy:
In this phase, the user describes the infrastructure by listing accessible nodes and their properties as well as how to interact

with them. The user also selects an exploration strategy (*e.g.,* Grid Search, Random Search, Bayesian Optimization) to state which configuration to chose during the Model Optimization phase.

Model Optimization:
In this phase, several FL workloads are deployed in parallel using the selected exploration strategy to sample hyperparameters to explore. For each deployment, a FL training is done and the model is evaluated on several metrics (*e.g.,* accuracy, power consumption, bandwidth usage). If the trained model gets better performance than the current production model, the latter is updated.

Reacting to Performance Degradation:
In this phase, the model is used in production and its performance is monitored and evaluated according to several metrics (*e.g.,* accuracy, battery usage). If any degradation in model prediction is detected (data drift) on a given client, the local model is retrained with the new data distribution. If model predictions does not get better, another parameter exploration is started.

To support this workflow we base our approach on 3 concepts.

Improving interoperability across IoT platforms:
Thing Description [5] (TD) is a W3C standard for IoT devices. It enables better interoperability for IoT platforms by using a uniform description of properties and interactions of a device. TDs can be viewed as entry points for IoT devices and help in their management.

Automatically deploying and monitoring FL solutions:
E2Clab [6] is a framework for reproducible deployment of experiments on the cloud-fog-edge Continuum using large scale testbeds. It allows researchers to deploy their experiments by describing important steps of a deployment: resource reservation, configuration and installation of required libraries and running necessary scripts. E2Clab comes with monitoring tools and an optimization manager which allows to deploy in parallel several experiments with different configurations and retrieve experiment results.

Adapting solutions to local data distributions:
Personalized FL [7] aims at providing personalized models for each client. Personalized models are built in a 2 phase process. First, clients collaboratively train a global model like in the standard FL process. Then, each client performs additional training steps on the model to adapt it to their local data distribution. We believe that, following this strategy, personalized models would provide better predictions for each client. Moreover, in case of data drift and performance degradation on a specific client, only retraining the model on this specific client would be sufficient.

IV. Early Results

To motivate our approach, we perform preliminary experiments on a Human Activity Recognition dataset [8]. In this dataset, smartphone data from 30 participants performing daily living activities are provided. We investigate two problems:

I) Impact of hyperparameter tuning in FL scenarios.
First we apply FL to this use case by affecting to 8 clients data from different participants. Model accuracy is evaluated on non-encountered data from the same participants used for training. We investigate the impact of client learning rate and number of local epochs on accuracy of the resulting model. Preliminary experiments show that performing several local epochs before aggregating updated model weights to the server might improve the learning. For instance, for the same number of total epochs (100) and same learning rate (0.03), performing 1 local epoch resulted in 52.9% accuracy against 98.6% accuracy while performing 5 local epochs. Moreover client learning rate need to be carefully selected as using a learning rate of 0.01 resulted in poor performance (31.4% accuracy) while using a learning rate of 0.03 resulted in 98.6% accuracy. Using a higher learning rate also resulted in lower accuracy (87.1% for a learning rate of 0.05). These early results show that in addition to standard ML hyperparameters, the ones specific to FL also need to be optimized. As a consequence, this complexify optimization of the model.

II) Impact of data drift on the production model.
As a second step, we investigate the impact of data drift on the production model. In this step, the model is first evaluated using data from the same participants used for training. Then it is evaluated on data from other participants (*e.g.,* walking style might affect the prediction). It results in degradation of model performance, dropping from 96% to 74% of accuracy, revealing that the model is not adapted to data coming from participants with different moving styles. We believe that using personalized FL would reduce the cost of retraining a model for a specific client, by adapting an already-trained central model in case of data drift.

V. Next Objectives

We motivated our approach with preliminary experiments. The next step is to implement our approach by describing our experimental infrastructure with Thing Description and deploy it with E2Clab. We also need to investigate how to best integrate model retraining strategy by using personalized FL. For experiments we will take advantage of Grid'5000 testbed [9]. Moreover, we are interested in investigating other open questions such as:

How to determine which model is performing better considering several conflicting metrics and clients with Non-IID data?

Considering constrained-devices with limited battery life, how often can we perform the FL training step?

References

[1] ETP4HPC, "Strategic research agenda." https://www.etp4hpc.eu/sra.html, 2020.

[2] I. Kholod, E. Yanaki, D. Fomichev, E. Shalugin, E. Novikova, E. Filippov, and M. Nordlund, "Open-source federated learning frameworks for iot: A comparative review and analysis," *Sensors*, vol. 21, no. 1, 2021.

[3] T. Zhang, L. Gao, C. He, M. Zhang, B. Krishnamachari, and S. Avestimehr, "Federated learning for internet of things: Applications, challenges, and opportunities," *CoRR*, vol. abs/2111.07494, 2021.

[4] M. Khodak, R. Tu, T. Li, L. Li, M. Balcan, V. Smith, and A. Talwalkar, "Federated hyperparameter tuning: Challenges, baselines, and connections to weight-sharing," *CoRR*, vol. abs/2106.04502, 2021.

[5] S. Kaebisch, T. Kamiya, M. McCool, V. Charpenay, and M. Kovatsch, "Web of things (wot) thing description." https://www.w3.org/TR/2020/REC-wot-thing-description-20200409/, 2020.

[6] D. Rosendo, P. Silva, M. Simonin, A. Costan, and G. Antoniu, "E2clab: Exploring the computing continuum through repeatable, replicable and reproducible edge-to-cloud experiments," in *IEEE International Conference on Cluster Computing, CLUSTER 2020, Kobe, Japan, September 14-17, 2020*, pp. 176–186, IEEE, 2020.

[7] Y. Jiang, J. Konecný, K. Rush, and S. Kannan, "Improving federated learning personalization via model agnostic meta learning," *ArXiv*, vol. abs/1909.12488, 2019.

[8] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones," in *ESANN*, 2013.

[9] F. Cappello, E. Caron, M. Dayde, F. Desprez, Y. Jegou, P. Primet, E. Jeannot, S. Lanteri, J. Leduc, N. Melab, G. Mornet, R. Namyst, B. Quetier, and O. Richard, "Grid'5000: a large scale and highly reconfigurable grid experimental testbed," in *The 6th IEEE/ACM International Workshop on Grid Computing, 2005.*, pp. 8 pp.–, 2005.