Understanding the Main Challenges of Federated Learning

TA: Debora Caldarola (debora.caldarola@polito.it)

OVERVIEW

Nowadays, a lot of data cannot be exploited by traditional Deep Learning (DL) methods due to its sensitive and privacy-protected nature. For instance, we can think of the data collected by the cameras or GPS sensors in our mobile phones, or produced by the Internet of Things (IoT). Introduced in 2016 by Google, **Federated Learning** (FL) is a machine learning scenario aiming to use privacy-protected data without violating the regulations in force. The goal is to learn a global model in privacy-constrained scenarios leveraging a client-server architecture. The central model is kept on the server-side and, unlike standard DL settings, has no direct access to the data, which is stored on multiple edge devices, *i.e.* the clients. Thanks to a paradigm based on the exchange of the model parameters (or gradient) between clients and server through multiple rounds, the global model is able to extract knowledge from data without breaking the users' privacy.

The goal of this project is to become familiar with the standard federated scenario and understand its main challenges: i) *statistical heterogeneity*, *i.e.* the non-i.i.d. distribution of the clients' data which leads to degraded performances and unstable learning; ii) *systems heterogeneity*, *i.e.* the presence of devices having different computational capabilities (*e.g.* smartphones VS servers); iii) *privacy concerns*, deriving from the possibility of obtaining information on clients' data from the updated model exchanged on the network. Specifically for the latter, you will investigate the *gradient inversion attack*, *i.e.* recovering the input from the gradient of the trained model. Once these issues are understood, you will have the opportunity to propose your solution or to implement an existing one among those selected.

GOALS

1. To become familiar with the standard federated scenario and its main algorithms. 2. To understand the real-world challenges related to data statistical heterogeneity, systems heterogeneity and privacy concerns (specifically the gradient inversion attack). 3. To replicate the experiments detailed in the following sections.
4. To implement and test your contribution for solving one of the highlighted issues.

STEPS

Become familiar with the literature and the state-of-the-art

Before starting the project, you should take some time to study and get familiar with the federated scenario, its algorithms, challenges and the main proposed solutions. This will also help you understand what to expect from the experiments you will be asked to do. Those are the main references you must study:

1. [1][2][3] to understand the standard FL setting
2. [4][5] to get a more comprehensive view of the current state of the literature
3. [6][7][8][9] for the effect of data statistical heterogeneity
4. [10][11] for systems heterogeneity
5. Gradient inversion attack [12]
6. [13][14] for possible countermeasures to the gradient inversion attack
7. ResNet [15]

If you wish to delve into this topic, you will find a more detailed bibliography at the end of this document, containing the references to the latest trends in the FL literature.

Centralized training: define the upper bound

The performance of any model trained in a federated fashion is upper bounded by the results obtained in the centralized setting, *i.e.* the standard one. Therefore, as a first step, you need to define that upper bound by training the model in a centralized way. The specifics are as follows:

- Model: ResNet-50 [15]
- Dataset: CIFAR10 (download here or use PyTorch Datasets)
- Task: classification on 10 classes
- Tune the values of learning rate, weight decay, momentum if necessary and the choice of the optimizer (SGD or Adam)
- Metrics: Accuracy
- Use the standard data augmentation pipeline: random crop, random horizontal flip, normalization.

In addition, as proved by [16], Group Normalization (GN) layers [17] are usually more effective than Batch Normalization (BN) ones in federated scenarios. So you are asked to repeat the previous experiment substituting BN layers with GN ones (number of groups: 2). Complete the following table with the obtained results:

Model	**Normalization Layers**	**Centralized Accuracy (%)**	**Number of model parameters**
ResNet-50	BN	66	23520842
ResNet-50	GN	74	23520842

Implement and test the federated baseline

You can now implement the federated framework for your baselines. You can find possible implementations here. Remember you are required to use **PyTorch** and be aware of possible bugs in any repo you might use as a starting point! Understand the code before using it. If you set up your own code, I suggest you look at this function for splitting the dataset among clients. The employed model is the same of the centralized experiments, both with BN and GN layers. It is now time to test your code! Run the experiment with FedAvg on the **IID** and **balanced** version of CIFAR10 (*i.e.* all clients likely see the same distribution of classes and number of samples) using the same architectures and hyperparameters of the centralized baseline. Report your results in the table below. If you find more rounds to be necessary, feel free to increase the quota and modify the table accordingly.

Federated baseline:

- number of clients: 100
- images per client: 500 in the IID version; variable in the non-IID one
- rounds: number of rounds necessary to reach performance close to that of the centralized baseline
- local epochs: 1
- server optimizer: SGD with learning rate 1 (FedAvg)
- local training hyper-parameters: same as in the centralized baseline

Baseline	**Normalization Layers**	**IID Accuracy (%) @ 200** rounds
FedAvg ResNet-50 on CIFAR10	BN	62
FedAvg ResNet-50 on CIFAR10	GN	50

Understand the effect of data statistical heterogeneity in FL

Repeat the same experiment using the **unbalanced** and **non-IID** version of the dataset (*i.e.* clients have access to different data distributions and number of samples). Report your results in the table below. Use the same number of rounds of the previous experiment. Which are the most evident issues arising when the data distribution is heterogeneous?

Baseline	**Normalization Layers**	**Non-IID Accuracy (%) @ 200**rounds
FedAvg ResNet-50 on CIFAR10	BN	25
FedAvg ResNet-50 on CIFAR10	GN	49

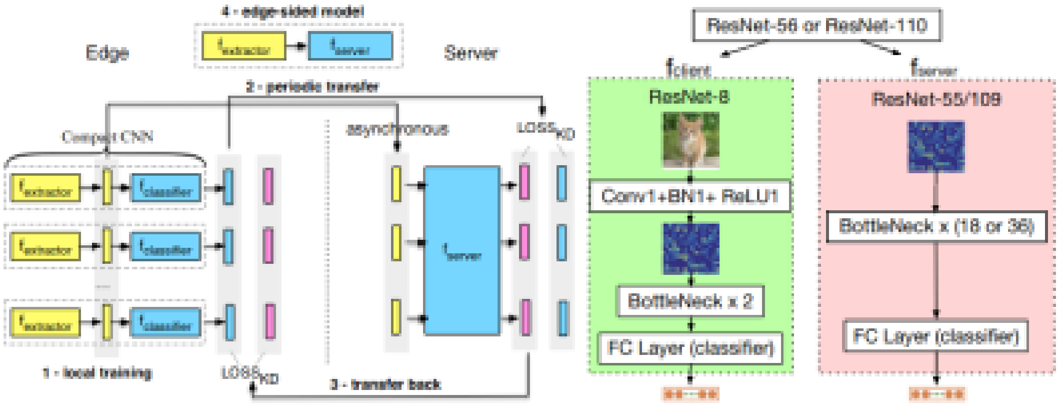
Simulate systems heterogeneity in FL

The usage of large neural networks usually leads to higher accuracies. Within a federated context, the large model size impedes training on resource-constrained edge devices. [11] proposes FedGKT to address such an issue, modifying the standard baseline into a Group

Knowledge Transfer algorithm. The clients train a smaller CNN and periodically send their knowledge to a larger server-side CNN. In this way, large architectures can be exploited without burdening the clients. In this step, you are required to implement the same algorithm proposed by [11]. For computational reasons, substitute their ResNet-56/110 with your ResNet-50. You can find the details on the client-side ResNet-8 in the paper (Table 7 Supplementary Material). Modify the server-side architecture accordingly. The figures below summarize FedGKT. Report your results in the table below in both IID and non-IID scenarios.

Training specifics:

- client-model: ResNet-8
- server-model: ResNet-49 (see Table 8 and 9 from [11] as reference for building the model)
- server epochs: from 10 to 40 (according to computational resources) - communication rounds: minimum number between the rounds used for standard FedAvg and the necessary rounds to reach convergence with FedGKT
- server optimizer: same as centralized baseline



(a) Alternating and periodical knowledge transfer (b) CNN architectures on the edge and server

Figure 1: Reformulation of Federated Learning: Group Knowledge Transfer

Algorithm	Normalization Layers	Data Distribution Accuracy (%)	
FedGKT	BN	IID	69
FedGKT	BN	Non-IID	53
FedGKT	GN	IID	72
FedGKT	GN	Non-IID	36

	Model Number of model parameters
ResNet-50	23,520,842
ResNet-8	10586

ResNet-49	23,503,842

Maintaining privacy in FL: gradient inversion attack

Gradient inversion attack is an emerging threat to the security of Federated Learning and its ability to preserve the users' privacy. An attacker eavesdropping on the communication with a client and the server can use the updated gradient to reconstruct the client's private data. [13] summarizes the attacks and defenses recently proposed. In this step, you are required to verify hands-on such an attack, using your architecture (ResNet-50) and dataset (you can limit the test to one client's data and updated model). You can find a possible implementation of the attack in a federated scenario here. Is the attacker able to reconstruct the original images?

No attacker can not exactly reconstruct the original image but can recover something near

Time for your personal contribution!

Last step: time to address one of the challenges seen so far! You can either choose to implement one of the solutions from the current literature listed below, or propose your own.

Statistical Heterogeneity

- FedDyn [9]
- SCAFFOLD [8]
- FedProx [10]
- FedVC [7]
- FedIR [7]
- Classifier Calibration [18]

Systems Heterogeneity

- FedProx with stragglers [10]
- Sageflow [19]
- Adaptive Delayed-SGD [20]

Privacy and Security in FL

- Differential Privacy [13][14]
- Encoding inputs [13]: Mixup [21] and InstaHide [22]

DELIVERABLES

Once completed all the steps, you need to submit:

1. PyTorch scripts with code for all the steps.
2. This file with completed tables.
3. A complete PDF report (paper-style). The report should contain the following sections: a brief introduction, description of the main related works, a methodological section describing the used algorithms and their purposes, an experimental section with all the results and discussions, and a final brief conclusion. Follow this link to open and create the template for the report.

EXAMPLES OF QUESTIONS YOU SHOULD BE ABLE TO ANSWER AT THE END OF THE PROJECT

- How is the standard federated framework setup?
- What was FL born for?
- Which are the main issues arising when the clients' data distribution is unbalanced and heterogeneous?
- Can you mention a few solutions addressing the problem of statistical heterogeneity? Which are their pros and cons?
- How does the systems heterogeneity affect training in FL?
- What is the gradient inversion attack? How can we try to prevent it in a federated context?
- Can you think of a few examples of real-world applications of FL?

REFERENCES

[1] Google AI Blog, Federated Learning: Collaborative Machine Learning without Centralized Training Data

[2] McMahan, Brendan et al. "Communication-Efficient Learning of Deep Networks from Decentralized Data" *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, PMLR* 54:1273-1282, (2017).

[3] Reddi, Sashank, et al."Adaptive federated optimization."ICLR (2021).

[4] Li, Tian, et al. "Federated Learning: Challenges, Methods, and Future Directions" *IEEE Signal Processing Magazine* 37.3 (2020): 50-60.

[5] Kairouz, Peter, et al. "Advances and Open Problems in Federated Learning" *arXiv preprint arXiv:1912.04977* (2019).

[6] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. "Measuring the effects of non-identical data distribution for federated visual classification", 2019

[7] Hsu TM.H. et al. "Federated Visual Classification with Real-World Data Distribution" *European Conference on Computer Vision . ECCV 2020.* Lecture Notes in Computer Science, vol 12355. Springer, Cham.

[8] Sai Praneeth Karimireddy, et al. “Scaffold: Stochastic controlled averaging for federated learning.” In *International Conference on Machine Learning*, pages 5132–5143. PMLR, 2020

[9] Acar, D. A. E., Zhao, Y., Navarro, R. M., Mattina, M., Whatmough, P. N., & Saligrama, V. (2021). “Federated learning based on dynamic regularization.” ICLR Oral (2021).

[10] Tian Li, et al. “Federated optimization in heterogeneous networks.” In I. Dhillon, D. Papailiopoulos, and V. Sze, editors, *Proceedings of Machine Learning and Systems*, volume 2, pages 429–450, 2020.

[11] He, Chaoyang, Murali Annavaram, and Salman Avestimehr. "Group knowledge transfer: Federated learning of large CNNs at the edge." *Advances in Neural Information Processing Systems* 33 (2020): 14068-14080.

[12] Geiping, Jonas, et al."Inverting gradients-how easy is it to break privacy in federated learning?." *Advancesin Neural Information Processing Systems* 33 (2020): 16937-16947.

[13] Huang, Yangsibo, et al."Evaluating gradient inversion attacks and defenses in federated learning." *Advancesin Neural Information Processing Systems* 34 (2021).

[14] Wei, Kang, et al."Federated learning with differential privacy: Algorithms and performance analysis."*IEEE Transactions on Information Forensics and Security* 15 (2020): 3454-3469.

[15] He, Kaiming, et al."Deep residual learning for image recognition." *arXiv preprint arXiv:1512.03385* (2015).

[16] Hsieh, Kevin, et al."The non-iid data quagmire of decentralized machine learning." *International Conference on Machine Learning*. PMLR, 2020.

[17] Wu, Yuxin, and Kaiming He."Group normalization." *Proceedings of the European conference on computer vision (ECCV)*. 2018.

[18] Luo, Mi, et al."No fear of heterogeneity: Classifier calibration for federated learning with non-iid data." *Advancesin Neural Information Processing Systems* 34 (2021).

[19] Park, Jung Wuk, et al."Sageflow: Robust Federated Learning against Both Stragglers and Adversaries." Advances in Neural Information Processing Systems 34 (2021).

[20] Li, Xingyu, et al."Stragglers are not disaster: A hybrid federated learning algorithm with delayed gradients." *arXiv preprint arXiv:2102.06329* (2021).

[21] Zhang, Hongyi, et al."mixup: Beyond empirical risk minimization." *ICLR* (2018).

[22] Huang, Yangsibo, et al."Instahide: Instance-hiding schemes for private distributed learning." *International Conference on Machine Learning*. PMLR, 2020.