How to deal with non-IID data in Federated Learning

Bob Vo, Joss Rakotobe
IFT 6085 Project
Université de Montréal
Montréal, QC, Canada

Abstract

Federated Learning is an emerging machine learning setting that empowers many clients to learn a common model while preserving their training data local. Although this decentralized method brings unique benefits such as privacy and greater security, it intrinsically incorporates challenging problems in terms of data distribution across clients. In particular, unbalanced and non-IID properties are common in Federated Learning task because each client has its own characteristics and data pattern. In this work, we review strategies to mitigate these issues with details discussion and recent technique updates. Related convergence bounds and comparisons are also discussed, depending on the severity of data skewness. We further present open challenges and potential research directions in this domain after literature review. This exploration article could serve as a foundation for promising research in the trending field of Federated Learning.

1 Introduction

1.1 Federated learning

The surge of mobile phones and Internet of Things in recent years has enabled novel applications in machine learning domain. Such devices continuously generate tremendous amount of data that can be fed to learning models, and their computational power is greater than ever [1]. Meanwhile, conventional machine learning settings rely on centralize servers to perform such learning tasks. In this format, data are collected from sources and transferred to the computing clusters to form a unified data-set. This centralized method pose critical concerns in some practical settings. Many clients and end users prefer to store their data locally because of their privacy-sensitive applications. For example, institutions such as hospitals and legal firms maintain private clients' information that cannot be shared with any other entity. Therefore, a new learning framework has emerged to address those concerns in the recent years. Its high level definition is presented as below.

Federated learning is a machine learning setting where multiple entities (clients) collaborate in solving a machine learning problem, under the coordination of a central sersver or service provider. Each client's raw data is stored locally and not exchanged or transferred; instead, focused updates intended for immediate aggregation are used to achieve the learning objective.[2]

Though a similar framework, Distributed Learning, has been studied for many years, there are key differences between the two frameworks. Table1 outlines key points to differentiate Distributed Learning and Federated Learning. Crucially, in Distributed Learning setting, training data are either centrally stored or able to access by all clients. One key characteristic of FL is that the updates from

Preprint. Work in progress.

Table 1: Comparision between Distributed learning and Federated Learning [2]

	Distributed Learning	Federated Learning
Data distribution	Centrally stored	Generated locally. Remains decentralized
Clients	Compute nodes ∼1-1000 nodes	Large no. of mobile devices, up to 10^{10}
Reliability	Few failures	Highly unreliable

```
for $t = 0 \ldots T-1$ do
    (Server) Sample a subset of $K$ devices
    for $k \in [K]$ in parallel do
        $w_k^t \leftarrow w^t$
        $w_k^{t+1} \leftarrow \textbf{LocalUpdate}_k(w_k^t)$
    end for
    $w^{t+1} \leftarrow \textbf{Aggregate}\{w_k^{t+1}\}_{k \in S_t}$
end for
```

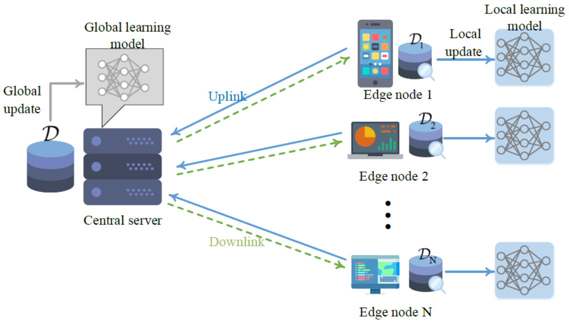

Figure 1: The typical Federated Learning setting [3]

clients are relatively unreliable because of stragglers or dropped out connections. Nevertheless, many results and methods from Distributed Learning are carried over to Federated Learning.

A typical Federated setting comprises a coordination server that orchestrates learning and updates activities of clients in the network. As illustrated in Figure 1, a subset of remote devices communicate with the central server at each communication round to perform local training and send back updates to the server via uplinks. After aggregating these updates, the server then push the new global model to a new subset of clients through downlinks. The whole process is repeatedly executed until the global model reaches a stopping criteria. This iterative process communicates only model updates instead of raw data information, so clients' privacy are preserved.

1.2 Our Focus and Contributions

Various challenging issues remain open for FL research. For example, clients can be massively distributed with limited communication resources. Therefore the compute nodes are generally unreliable because of unexpected offline and adversaries.

Another major challenge is the handling of non-IID data. More often than not, in the cross-device setting of FL we will encounter a heterogeneous setting. In fact, since each user has its own experience, the locally stored data is more likely to be different from one device to another, which brings the issue of non-identical distribution across devices. For example, suppose we have

- A next word prediction task, and we know that the sentence is starting by 'I want to'. Then the distribution of the label will be very different from one device to another.
- An animal recognition task, live from the device camera. Then, the label kangaroo will have a higher probability for devices in Australia than in Canada.

Moreover, the amount of data at each client varies. Heavy users are supposed to generate more data from their devices than light users, leading to unbalanced data. Finally, the unreliability of the communication is in general addressed by imposing constraints on the selection of the devices that participate in the updates. These constraints are typically for the devices to be idle, plugged, and connected to the Wi-Fi. The problem is that it will also induce a geographical dependence between the devices that are selected for that update. In fact, this type of situation only happens during the night.

The problems of non-IID data can be summed up in two big parts:

- To ensure convergence of the algorithms. The theoretical results of convergence were mostly developed based on the IID assumption.

- To generate a test set representative of the whole dataset becomes very difficult, if not unfeasible.

2 Review of strategies to overcome Non-IID data

In this short review, we will define $f(w, z)$ to be the loss function of a model w at an example $z = (x, y)$. For each device $k \in \{1, \ldots, N\}$, let P_k be its data distribution and define its local objective function

$$F_k(w) := \mathop{\mathbb{E}}_{z \sim \mathcal{P}_k} [f(w, z)]$$

The global objective function that we wish to minimize is defined as

$$F(w) := \frac{1}{N} \sum_{k=1}^{N} F_k(w)$$

Various approaches have recently come to light in order to overcome the problem of non-IID data.

2.1 Data Augmentation

Beside tweaking the core algorithms, innovative methods to enhance highly skewed non-I.I.D. datasets could also mitigate the statistical problem. Zhao et al. [4] has shown that by creating a small subset of data that is globally shared between all participants, the training accuracy can be increased by 30 % for the CIFAR-10 dataset with only 5 % globally shared data. Alternatively, each client can share a distilled data-set based on their local data [5]. The idea is to synthesize few[1] data points that can be used to train an approximate model that is similar to the model trained on original data. Incorporating public data-set is another technique to improve the performance of the global model. This approach is fairly common to pre-train deep neural networks. Data Augmentation can also be applicable when the original data-set is unbalanced [6], which is very common in FL. However, one should be cautious when applying such augmentation techniques because of data leakage potentials.

2.2 Employ Novel Learning Schemes

While most Federated Learning tasks focus on learning one global model, some learning schemes introduce multi-model concept to not only tackle non-IID data but also exploit the phenomenon to yield better models. Based on the clients' relationships, many clusters of similar devices can be created to learn common models. In Multi-Task Learning [7], such relationships can be either known a priori [8] or learned from data [9; 10]. Another learning format in this subgroup is Meta Learning [11]. In this way, a general model is trained on available data at the beginning of the training process. The global model is then fine-tuned with a few local gradient steps to form a new model [12]. Such approach may greatly improve the performance of clients with limited data. Similarly, the outcomes are personalized for each local user [13].

2.3 Evolution of the Algorithms

Federated Averaging (FedAvg, [14]) was the first algorithm to address the issue of privacy via a FL setting. Its *LocalUpdate* function (see Algorithm in Figure 1) is simple: for each device selected, in parallel perform a fixed number of steps of stochastic gradient descent (SGD) locally. Although the first experiments on non-IID data were promising, it was later shown in [15] and in [4] that under highly heterogeneous settings, because the updates that each device sends back to the server might be going too far into a particular direction proper to its surrogate objective, when aggregating the weights, FedAvg could become very unstable to the point of non-convergence. However, it was proved in [16] that FedAvg converges under some strong assumption on the local surrogate objective functions.

FedProx was proposed by Sahu et al. [15] as an upgrade of FedAvg in order to control its stability. The *LocalUpdate* optimizes a regularized local objective. The regularizer, called the *proximal term*,

[1] It was tested on MNIST that 10 data samples can actually represent the whole dataset, in the sense that when training on these 10 data samples, we can reach an accuracy of over 90%, given a particular initialization.

is preventing the weights from going too far from the initialization.

SCAFFOLD was proposed by Karimireddy et al. [17] as another attempt to control the instability of FedAvg using *control variates*. Instead of limiting the distance of the updated weights as in FedProx, they have regulated its direction by learning server and client control variate terms.

We detail the local updates of these three models in table 2. For FedProx, γ can vary for each client, and at each round. Its value can control the number of steps in the optimization. FedAvg can be viewed as a particular case of FedProx ($\mu = 0, \gamma$ constant) and of SCAFFOLD (by forcing $c_k = 0$).

LocalUpdate

Algorithm 1: FedAvg

```
Input: $E, w^t, \alpha$
for $i = 1, \ldots, E$ do
    $w^t \leftarrow w^t - \alpha \nabla F_k(w^t)$
end for
```

Algorithm 2: FedProx

```
Input: $w^t, \gamma$
$h(w) := F_k + \frac{\mu}{2}\|w - w^t\|^2$
$w_k \leftarrow w^t$
while $\nabla h(w_k) > \gamma \nabla h(w^t)$ do
        optimize $h(w_k)$
end while
```

Algorithm 3: SCAFFOLD

```
Input: $E, w^t, \alpha, c$
for $i = 1, \ldots, E$ do
    $w_k \leftarrow w^t$
    $w_k \leftarrow w_k - \alpha(\nabla F_k(w_k) - c_k + c)$
end for
$c_k \leftarrow \nabla F_k(w^t)$
return $(w_k, c_k)$
```

Table 2: Local update function of each algorithm for client k. For SCAFFOLD, the clients control variates c_k are also aggregated in the server to form the server control variate c.

2.4 Convergence results

We summarize in table 3 the convergence results derived from the previously described models and others in the heterogeneous setting (non-IID data across clients).

Inter-client assumptions[2]		Other assumptions[1]	
BCGV	$\mathbb{E}_k\|\nabla F_k(w) - \nabla F(w)\|^2 \leq \eta^2$	BLGV	$\mathbb{E}_k\left[\mathbb{E}_{z\sim P_k}\|\nabla f(w,z) - \nabla F(w)\|^2\right] \leq \eta^2$
BOOD	$\|F(w^*) - \mathbb{E}_k[F_k(w_k^*)]\| \leq \eta^2$	BLGN	$\mathbb{E}_{z\sim P_k}\|\nabla f(w^t,z)\|^2 \leq \eta^2$
BGD	$\mathbb{E}_k\|\nabla F_k(w)\|^2/\|\nabla F(w)\|^2 \leq \eta^2$	BNCVX	$\nabla^2 F_k(w) \succeq -\eta^2 I$
BMGA	$\|\theta_k^t\| \leq \frac{\pi}{2}$	SCVX	$\nabla^2 F_k(w) \succeq \eta^2 I$

Convergence Rates

Method	Non-IID	Other assumptions	Variant	Rate
Parallel SGD [18]	BCGV	BLGV	$K = N; E = 1$	$\mathcal{O}(1/T) + \mathcal{O}(1/\sqrt{NT})$
MATCHA [19]	BCGV	BLGV	No Stragglers	$\mathcal{O}(1/\sqrt{TKE}) + \mathcal{O}(K/ET)$
FedAvg [16]	BOOD	SCVX BLGV BLGN	–	$\mathcal{O}(E/T)$
FedProx [15]	BGD	BNCVX	Proximal	$\mathcal{O}(1/\sqrt{T})$
MFL[3] [3]	BMGA	SCVX	Momentum; $K = N$	$\mathcal{O}(1/T)$
SCAFFOLD [17]	–	SCVX BLGV	CV[1]	$\mathcal{O}\left(1/TEM\right) + \mathcal{O}\left(e^{-T}\right)$

Table 3: Convergence rates of $F(w^T) - F(w^*)$ for convex functions, $\|\nabla F(w^T)\|$ for non-convex, with T the number of steps. All local surrogate F_k are assumed to be smooth and have Lipschitz gradient. We describe each method as a variant of FedAvg. This table was adapted from [2].

[2] BCGV: Bounded inter-Client Gradient Variance; BOOD: Bounded Optimal Objective Difference
BGD: Bounded Gradient Dissimilarity; BLGV: Bounded Local Gradient Variance
BLGN: Bounded Local Gradient Norm; BNCVX: Bounded Non-Convexity; SCVX: Strongly Convex; CV: Control Variate

[3] Momentum Federated Learning

The convergence rates derived in non-IID are similar in terms of order of convergence with respect to T to the results on IID with smooth and convex functions (see [20], [21] and [22]), but the main difference is the need of these assumptions. Basically, the Inter-client assumptions in table 3 are all bounding in various ways the difference between the clients' data distributions. Take notice that all of them would be always satisfied if we have a homogeneous setting. It is important to notice the upgrade in these assumptions from FedAvg to FedProx, which does not require convexity, and SCAFFOLD which does not need any inter-client assumptions.

3 Open challenges and possible directions

3.1 Extending beyond supervised learning

Most of the reviewed articles were based on Supervised Learning. Though, in the Federated Learning context, it seems that other learning schemes such as Online Learning could be equally important. In fact, clients that participate in federated setting continuously generate fresh data after each round. So the objective of the learning should be minimizing the regret instead of minimizing the overall loss. This is the fundamental idea behind Online Learning [23]. With this idea in mind, we can try to derive regret bounds based on various assumptions as mentioned in Table 3. Some pioneering works using Online Learning in FL was surfaced recently [24; 25].

3.2 Fairness and Adaptation in FL

The objectives for FL problems could extend beyond convergence speed or overall accuracy. One prominence alternative goal is fairness among clients [26], and this fairness can be measured based on resource allocation or experience of each user. In a non-IID setting, when some clients have more data and resources than others, it may be unfair to demand that client to run more computation rounds. Similarly, instead of targeting the overall model accuracy, we can encourage a fair accuracy target for every client [27]. Such fairness indicators can be considered as constrains to incorporate into existing assumptions.

While the majority of the mentioned works in Table 2 discuss the convergence rate in terms of iterations, It is worth considering such speeds in terms of wall-clock time when it comes to practical applications. In this way, adaptive parameter can be considered to optimize total training plus communicating time. This direction seems to attract more attention lately[28; 24]

3.3 Heterogeneity Diagnostic

Recent algorithms have aimed to quantify statistical heterogeneity through metrics such as local dissimilarity (in FedProx [15]) and earth mover's distance (in [4]) to restrict it. However, the heterogeneity of the distributions is a natural sign of personalization before being a noise. One direction one could follow is to try to use such metrics to cluster the clients. However, these metrics cannot be easily calculated over the federated network before training occurs. We propose a new direction involving a dynamic clustering, and making use of the idea proposed in SCAFFOLD and MFL. However, instead of controlling the variates, we will use them for clustering. We detail the model in Algorithm 4. Since the server needs to keep and communicate as many sets of weights as the number of clusters, we propose a minimal clustering approach. We propose two ways of clustering:

- Using the direction of the updates c_k. In this case, the clustering is done using the angle between the $c_k, k \in S_t$. We make a minimal number of cluster such that within each cluster, all absolute angles between pairs of c_k are less than C. In this setting, each client needs to keep track of its cluster id.
- Using the updates w_k. We find a minimal clustering such that the distance between pairs of w_k is less than C.

In both cases, the hyperparameter C helps control the number of clusters, which is related to the similarity of the distributions across clients.

Algorithm 4 Proposition of a Dynamic Clustering Federated Learning

```
**Hyperparameters**:$K, T, E \in \mathbb{N}, C > 0, \alpha > 0, w^0, W =$ set()
**Initialization**: $W.add(w^0)$
**for** $t = 1, \dots, T$ **do**
  Select a sample $S_t$ of $K$ devices
  **for** $k \in S_t$ **in parallel do**
    $w_k^0 \leftarrow \arg\min_{w \in W} F_k(w)$    # Find the best model for device $k$
    **for** $i = 1, \dots, E$ **do**
      $w_k \leftarrow w_k^0$
      $w_k \leftarrow w_k - \alpha \nabla F_k(w_k)$
    **end for**
    $c_k \leftarrow w_k - w_k^0$                  # The direction of the new update
  **end for**
  $\mathcal{C}_t \leftarrow$ **Cluster**$_C(S_t$ w.r.t. $w_k$ or $c_k)$
  $W.add(\{\frac{1}{|S|} \sum_{k \in S} w_k | S \in \mathcal{C}_t\})$
**end for**
```

4 Contribution Statement

Bob and Joss contributed equally to the literature review, to the open challenges section, and to the writing of the manuscript.

5 Discussion and Conclusion

This project investigated the literature of Federated Learning - in the context of Non I.I.D, and strategies to deal with such challenge. We also propose potential research directions based on our understanding and reasoning about the relevant of the topic ideas.

While there are many sources that cause Non-IID in FL, various solutions were proposed to tackle the issue. The list of solutions ranging from data augmentation, multitask learning to convergence bounds based on inter-client assumptions. Although we did not explicitly cover all major scenarios, we believe that the convergence discussion was fairly broad to an extent because we referred multiple recent works and extensive review papers [2] [1]. Though, some limitations in this article might be related to the coverage of existing solutions because the broad application of the federated learning concept. Besides, some promising potential directions were also proposed above.

References

[1] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated Learning: Challenges, Methods, and Future Directions," pp. 1–21, 2019.

[2] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, R. G. L. D'Oliveira, S. E. Rouayheb, D. Evans, J. Gardner, Z. Garrett, A. Gascón, B. Ghazi, P. B. Gibbons, M. Gruteser, Z. Harchaoui, C. He, L. He, Z. Huo, B. Hutchinson, J. Hsu, M. Jaggi, T. Javidi, G. Joshi, M. Khodak, J. Konečný, A. Korolova, F. Koushanfar, S. Koyejo, T. Lepoint, Y. Liu, P. Mittal, M. Mohri, R. Nock, A. Özgür, R. Pagh, M. Raykova, H. Qi, D. Ramage, R. Raskar, D. Song, W. Song, S. U. Stich, Z. Sun, A. T. Suresh, F. Tramèr, P. Vepakomma, J. Wang, L. Xiong, Z. Xu, Q. Yang, F. X. Yu, H. Yu, and S. Zhao, "Advances and Open Problems in Federated Learning," pp. 1–105, dec 2019.

[3] W. Liu, L. Chen, Y. Chen, and W. Zhang, "Accelerating federated learning via momentum gradient descent," 2019.

[4] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated Learning with Non-IID Data," 2018.

[5] T. Wang, J.-Y. Zhu, A. Torralba, and A. A. Efros, "Dataset Distillation," pp. 1–14, 2018.

[6] J. Wang and L. Perez, “The effectiveness of data augmentation in image classification using deep learning,”

[7] Y. Zhang and Q. Yang, “A Survey on Multi-Task Learning,” pp. 1–20, 2017.

[8] A. Argyriou, T. Evgeniou, and M. Pontil, “Convex multi-task feature learning,” *Machine learning*, vol. 73, no. 3, pp. 243–272, 2008.

[9] V. Zantedeschi, A. Bellet, and M. Tommasi, “Fully decentralized joint learning of personalized models and collaboration graphs,” 2019.

[10] V. Smith, C.-K. Chiang, M. Sanjabi, and A. Talwalkar, “Federated multi-task learning,” 2017.

[11] J. Li, M. Khodak, S. Caldas, and A. Talwalkar, “Differentially private meta-learning,” 2019.

[12] C. Finn, P. Abbeel, and S. Levine, “Model-agnostic meta-learning for fast adaptation of deep networks,” in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML’17, p. 1126–1135, JMLR.org, 2017.

[13] Y. Jiang, J. Konečný, K. Rush, and S. Kannan, “Improving federated learning personalization via model agnostic meta learning,” 2019.

[14] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, “Federated learning of deep networks using model averaging,” *CoRR*, vol. abs/1602.05629, 2016.

[15] A. K. Sahu, T. Li, M. Sanjabi, M. Zaheer, A. Talwalkar, and V. Smith, “On the convergence of federated optimization in heterogeneous networks,” *CoRR*, vol. abs/1812.06127, 2018.

[16] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, “On the convergence of fedavg on non-iid data,” 2019.

[17] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh, “Scaffold: Stochastic controlled averaging for federated learning,” 2019.

[18] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu, “Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent,” 2017.

[19] J. Wang, A. K. Sahu, Z. Yang, G. Joshi, and S. Kar, “MATCHA: speeding up decentralized SGD via matching decomposition sampling,” *CoRR*, vol. abs/1905.09435, 2019.

[20] S. U. Stich, “Local sgd converges fast and communicates little,” 2018.

[21] J. Wang and G. Joshi, “Cooperative SGD: A unified framework for the design and analysis of communication-efficient SGD algorithms,” *CoRR*, vol. abs/1808.07576, 2018.

[22] H. Yu, S. Yang, and S. Zhu, “Parallel restarted SGD with faster convergence and less communication: Demystifying why model averaging works for deep learning,” 2018.

[23] S. Shalev-Shwartz *et al.*, “Online learning and online convex optimization,” *Foundations and Trends® in Machine Learning*, vol. 4, no. 2, pp. 107–194, 2012.

[24] P. Han, S. Wang, and K. K. Leung, “Adaptive gradient sparsification for efficient federated learning: An online learning approach,” 2020.

[25] C. He, C. Tan, H. Tang, S. Qiu, and J. Liu, “Central server free federated learning over single-sided trust social networks,” 2019.

[26] M. Mohri, G. Sivek, and A. T. Suresh, “Agnostic federated learning,” *36th Int. Conf. Mach. Learn. ICML 2019*, vol. 2019-June, pp. 8114–8124, 2019.

[27] T. Li, M. Sanjabi, A. Beirami, and V. Smith, “Fair resource allocation in federated learning,” in *International Conference on Learning Representations*, 2020.

[28] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, “Adaptive Federated Learning in Resource Constrained Edge Computing Systems,” *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205–1221, 2019.