ISSN 2959-6157

A Comparative Study of Federated Learning Aggregation Algorithms under Non-IID Conditions

Yicheng Cao

ZJU-UIUC Institute, Zhejiang University, Zhejiang, China
Corresponding author: yicheng.23@intl.zju.edu.cn

Abstract:

Privacy can be maintained in collaborative model training by enabling clients to contribute without exposing their local datasets, as supported by Federated Learning (FL). Performance-wise, however, in scenarios involving non-IID data across clients, training becomes significantly more complex and less reliable, as is often the case with real-world datasets. In this work, we provide a comprehensive comparative review of three popular aggregation algorithms in FL: Federated Averaging (FedAvg), Federated Proximal (FedProx) and Stochastic Controlled Averaging (SCAFFOLD). Among these baselines, FedAvg is the fundamental approach, FedProx adds the proximal term to suppress the drift of clients and SCAFFOLD utilizes control variates to adjust the updating directions of local clients. According to the related work, FedAvg, which is communication-efficient, is unstable and slow on non-IID data. FedProx stabilizes convergence, and SCAFFOLD has the best convergence performance, but it has to exchange control variates to solve them, thus consuming more communication. While no algorithm appears to dominate for all metrics, it underscores the significance of choosing FL strategy which can meet certain trade-off between accuracy, speed of convergence and communication expense.

Keywords: Non-IID Data; Federated Averaging; Federated Proximal; Stochastic Controlled Averaging.

1. Introduction

By allowing decentralized clients to jointly train a global model while keeping their raw datasets local, Federated Learning (FL) offers a promising solution to privacy concerns in collaborative machine learning. Federated learning can be adopted in many sensitive application domains, such as mobile devices or the health care industry, since the underlying data could be protected by privacy and regulations and therefore cannot be centralized [1, 2]. Unfortunately, for real-world FL systems, non-independent and

ISSN 2959-6157

identically distributed (non-IID) client data are common and degrade the model performance seriously due to the client drift, unreliable convergence, communication inefficiency, etc [3].

Even though they are commonly applied, systematic comparison of these algorithm under identical experimental setting is rare. Especially, there lacks study in evaluating their algorithmic mechanism, theoretical incentives and practical tradeoff against the non-IID client's data. This paper conducts a structured analysis of FedAvg, FedProx, and SCAFFOLD in federated learning, focusing on their characteristics, strengths, and limitations with respect to robustness, convergence, and communication efficiency. We would like to provide actionable information that helps select well-suited FL strategies on heterogeneity.

2. Background and Motivation

As an emerging approach to privacy-preserving machine learning, Federated Learning (FL) allows distributed clients to jointly build a global model while keeping their raw data local and confidential. This is of particular interest on applications where data privacy is an issue, or data use is subject to regulation, such as personalized service (e.g. user's mobile devices) prediction of the financial market, medical image analysis, and personalized medical diagnostics [1, 2]. In contrast to keeping data at a central location, with FL, not only can data sets be stored in distributed data silos, but edges can benefit from diverse data scale to train models.

Nevertheless, due to statistical heterogeneity, it is a significant challenge to practically deploy FL, especially when client data is non-identically and independently distributed (non-IID). In these circumstances, one client local data distribution can be very different from those of other clients, so that client drift, instability to convergence, and poor global performance will be induced [3]. All these problems are further aggravated due to the clients' heterogeneous computing capabilities, communication bandwidth limitations and infrequent participation.

In response, a succession of enhanced aggregation algorithms has been suggested. FedAvg, the "gold standard" FL algorithm works well for IID settings, but deteriorates dramatically when data heterogeneity exists [4]. FedAvg lacks any mechanism for counteracting the divergence caused by non-IID local updates [5]. To this end, FedProx further regularizes the client optimization objective to stabilize the local update by adding a proximal term, working better on moderate non-IID cases [6]. Recently, SCAFFOLD proposes a control variate approach which explicitly compensates for local update drift by maintaining the gradient estimates on both the client and the server to adapt better on severe heterogeneity [7].

However, because of the popularity of these algorithms, in existing literature they are usually compared to each other only through either standalone assessment or through different assumptions. This drives the need for a well-organized comparative study with a unified experimental design [8]. The objective of this paper is to compare FedAvg, FedProx, and SCAFFOLD directly and comparatively across theoretical, empirical, and practical perspectives. We aim to offer practical guidelines on the choices of suitable aggregation strategies for FL systems running in the realistic, non-IID regimes.

3. Algorithm Overview

3.1 Federated Averaging

Federated Averaging (FedAvg) is the original aggregation algorithm in Federated Learning. The server dispatches the current global model to a portion of clients, who then engage in local updates over their private data using SGD across several training epochs. These updated models are subsequently aggregated on the server side through a data-weighted averaging process [4]:

$$w^{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} w_k^t. \tag{1}$$

Here, w_k^t represents the local model of client k at round t, and its contribution to aggregation is weighted according to the volume of local training data.

Although FedAvg is intuitive and communication-efficient, which lead to its popularity in early FL applications. However, under non-IID setting, FedAvg is faced with client drift due to different local update process causes global model drifts or converge slow [5]. These issues underscore the need for further algorithmic improvements to cope with statistical heterogeneity.

3.2 Federated Proximal

Federated Proximal (FedProx) further adapts FedAvg by altering the local optimization objective. A regularization term, known as the proximal component, is incorporated into the local objective to penalize updates that stray from the global model. The local training objective will be updated as follows [6]:

$$\min_{w}\left[F_k\left(w\right)+\frac{\mu}{2}\left|\,\left|w-w^t\right|\right|^2\right]. \tag{2}$$

This proximal term $\frac{\mu}{2}\left|\,\left|w-w^t\right|\,\right|^2$ constrains the local update, stabilizing optimization under heterogeneous data. FedProx improves robustness in non-IID settings without significantly increasing communication cost. However, its performance is sensitive to the choice of the regularization coefficient μ [6].

3.3 Stochastic Controlled Averaging

Stochastic Controlled Averaging (SCAFFOLD) addresses client drift by incorporating control variates into local updates. Every client holds its own correction term c_k, while the server keeps track of a global control variable c. During local training, the client updates its model as [7]:

$$w_k^{t+1} = w_k^t - \eta\left[\nabla F_k\left(w_k^t\right) - c_k + c\right]. \tag{3}$$

This correction term $-c_k + c$ helps align local updates with the global optimization direction, reducing variance caused by non-IID data.

SCAFFOLD can noticeably accelerate convergence and also be more stable compared with FedAvg in heterogeneous environments [7]. Nevertheless, SCAFFOLD needs to synchronize the control variates with other workers, which incurs more communications costs [9]. While theoretically appealing, SCAFFOLD's design makes it less suitable for devices with strict communication budgets or limited memory.

Table 1. Federated learning: Aggregation algorithms comparison.

Feature	FedAvg	FedProx	SCAFFOLD
Local Objective	$F_k(w)$	$F_k(w)+\frac{\mu}{2}\| \|w-w^t\|^2$	$F_k(w)$ with update correction
Update Correction	No	No	Yes (control variates)
Handles Non-IID	Poor	Moderate	Good
Implementation Complexity	Low	Moderate	High

4. Comparative Discussion

In this section, we contrast FedAvg, FedProx and SCAFFOLD from the perspective of (i) convergence performance, (ii) robustness to data heterogeneity, and (iii) communication and computation complexity, from both a theoretical and empirical perspective, as shown in Table 1.

As a baseline method, FedAvg is appreciated for its simplicity and limited communication overhead. When facing the data identically and independently distributed (IID) setting, it achieves a fairly reasonable performance. However, its performance sharply collapses under non-IID setting. It permits clients to minimize their local objective without any regularization or correction and clients' updates will go in different directions that are hard to converge the global model fast and reliably. By way of contrast, experiments in previous work indicate that FedAvg sometimes needs many more communication rounds than methods developed for non-IID settings, in order to reach comparable levels of accuracy [5, 6].

To address this issue, FedProx modifies the local objective function by incorporating a proximal component that discourages updates from straying too far from the global model. This can be regarded as a soft constraint that prevents clients from straying too far from the global direction. As a result, this soft constraint prevents client drift and stabilizes client training. In practice, FedProx smooths and accelerates convergence (compared with FedAvg) for moderately heterogeneous clients [6]. But FedProx adds another hyperparameter.

SCAFFOLD is based on a different idea of using control variates to compensate the directions of client updates. SCAFFOLD has a pair of control vectors for the server and client which estimate the gradient drift. When each client updates locally, it will modify its gradients with the control variates so that the client estimates of the parameter update can be made consistent with the global objective. SCAFFOLD achieves much faster and more robust convergence in a setting with extreme non-IID conditions as demonstrated in previous work [7]. But the overhead for the control variates communication comes at the price of paying additional rounds: The client and the server needs to exchange these vectors at every round or every few rounds, depending on their synchronization strategy. Additionally, due to extra state maintained on both sides and additional control flow on the SCAFFOLD update, overhead is also greater.

ISSN 2959-6157

Table 2. Performance Comparison among Different Aggregation Algorithms in the Case of Non-IID Setting.

Feature	FedAvg	FedProx	SCAFFOLD
Convergence Speed	Poor	Moderate	Good
Communication Efficiency	Slow	Moderate	Fast
Tuning Sensitivity	High	High	Lower (extra sync)

A comparative summary is shown in Table 2. We observe that: (1) FedAvg is better in the low heterogeneity and low communication-resource cases; (2) FedProx can work in a medium IID non-IID environment and resource-constrained systems; (3) SCAFFOLD is optimal when fast convergence and robustness are critical, and when the system can tolerate increased communication and implementation cost. Overall, it is subject to the nature of application scenario such as the data distribution, system limitation, and accuracy constraints which strategy of aggregation should be selected [10].

5. Conclusion

This paper focused on a controlled experimental comparison of three established aggregation methods in FL: FedAvg, FedProx and SCAFFOLD. They reflect varying approaches to overcoming the IID client data assumption in FL, which is regularly found in the field.

We show that while efficient and straightforward to code, FedAvg performs poorly in heterogeneous settings, as a result of client drift and the absence of any corrective mechanism. FedProx addressed this shortcoming by adding a proximal term that can stabilize the process; however, the regularization parameter of FedProx must be tuned. SCAFFOLD takes this further, using control variates to correct the local updates, leading to faster convergence, albeit with greater communication demands and added intricacy in system design.

By describing both the algorithms at high level and comparing them, we noted the trade-off for accuracy vs convergence speed vs communication efficiency vs tuning sensitivity. Notably, comprehensive reviews emphasize that existing studies often evaluate aggregation methods under inconsistent settings, lacking a unified experimental protocol. As we discuss in Tables 1 and 2, we find no algorithm better than others, it all depends on the scenario. FedAvg is most appropriate in settings of communication constrained or close-IID data, FedProx is the best choice in settings of general-heterogeneous data to stabilize the algorithms and maintain simplicity, and SCAFFOLD is most applicable in settings of high heterogeneous data when performance in convergence is the priority.

Our future work might study adaptiveness techniques which can dynamically select which kind of aggregation algorithm to use according to current working conditions in system, or even mixture algorithms that leverage various types of algorithms' advantages. In general, we expect that this research may become a useful reference to choose FL aggregation algorithms suitable for various deployments.

References

[1] P. M. Mammen. Federated Learning: Opportunities and Challenges. arXiv: arXiv:2101.05428.

[2] J. Alekseenko, A. Karargyris, and N. Padoy. Distance-Aware Non-IID Federated Learning for Generalization and Personalization in Medical Imaging Segmentation. Medical Imaging with Deep Learning, Paris, France, 03-05 juillet 2024.

[3] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor. Tackling the Objective Inconsistency Problem in Heterogeneous Federated Optimization. arXiv: arXiv:2007.07481.

[4] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. arXiv: arXiv:1602.05629.

[5] W. Huang, Z. Cai, Y. Shi, and T. Suzuki. Understanding Convergence and Generalization in Federated Learning through Feature Learning Theory. The Twelfth International Conference on Learning Representations. 2023.

[6] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith. Federated Optimization in Heterogeneous Networks. arXiv: arXiv:1812.06127.

[7] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh. SCAFFOLD: Stochastic Controlled Averaging for Federated Learning. arXiv: arXiv:1910.06378.

[8] W. Guo, F. Zhuang, X. Zhang, Y. Tong, and J. Dong. A Comprehensive Survey of Federated Transfer Learning: Challenges, Methods and Applications. arXiv: arXiv:2403.01387.

[9] F. Varno, M. Saghayi, L. R. Sevyeri, S. Gupta, S. Matwin, and M. Havaei. AdaBest: Minimizing Client Drift in Federated Learning via Adaptive Bias Estimation. European Conference on Computer Vision, vol. 13683, 2022, pp. 710–726.

[10] D. M. J. G et al. Non-IID data in Federated Learning: A Systematic Review with Taxonomy, Metrics, Methods, Frameworks and Future Directions. arXiv: arXiv:2411.12377.