FedPerf: A Practitioners' Guide to Performance of Federated Learning Algorithms

Ajinkya Mulay[1]**, Ayush Manish Agrawal**[2]**, Tushar Semwal**[3]*
{[1]Purdue University, [2]University of Nebraska-Lincoln , [3]The University of Edinburgh, [1,2,3]OpenMined}
{[1]mulay@purdue.edu, [2]aagrawal@nebraska.edu, [3]tushar.semwal@ed.ac.uk}

Abstract

Federated Learning (FL) enables the edge devices to collaboratively train a joint model without sharing their local data. This decentralised and distributed approach improves user privacy, security, and trust. Different variants of FL algorithms have presented promising results on both IID and skewed Non-IID data. However, the performance of FL algorithms is found to be sensitive to the FL system parameters and hyperparameters of the used model. In practice, tuning the right set of parameter settings for an FL algorithm is an expensive task. In this preregister paper, we propose an empirical investigation on five prominent FL algorithms to discover the relation between the FL System Parameters (FLSPs) and their performance. The FLSPs adds extra complexity to FL algorithms over a traditional ML system. We hypothesise that choosing the best FL algorithm for the given FLSP is not a trivial problem. Further, we endeavour to formulate a single easy-to-use metric which can describe the performance of an FL algorithm, thereby making the comparison simpler.

1 Introduction

Data is naturally found to be decentralised and distributed across the edge devices. Conventional Machine Learning (ML) approaches involve first collecting this data into a central server and then training a global model using the aggregated dataset. Though this centralised form of training provides better control, however, it suffers from two paramount issues. The first is the privacy of the data owners as governed by the General Data Protection Regulation [1] and Health Insurance Portability and Accountability Act [2]. The other major concern with traditional ML approaches is the communication overhead. For instance, uploading the data from a resource-limited end-device to a central server depletes the battery.

Federated Learning (FL) [3] is a new paradigm which allows the edge devices called *clients* to train a global model collaboratively. FL involves multiple communication cycles. In each cycle, a set of clients train an ML model on their local data and share only the model updates (gradients) with the central server. The central server then aggregates these updates from a pool of selected clients and does a single update to the global model. Finally, the server shares the updated global model back to the clients thereby completing one cycle. Thus, instead of sharing the data, the clients only share the marginally smaller sized model updates, hence reducing communication overheads and avoiding a potential privacy leak.

Since its inception, FL has shown promising results on training decentralised and Non-Independent and Identically Distributed (Non-IID) datasets. Hard et al. [4] introduced a recurrent neural network model for the next-word prediction task in Google virtual keyboard (GBoard) on millions of mobile devices. Similar large-scale system design of FL is described in [5]. Nevertheless, FL has its own

[*] Names in alphabetical order. All co-authors equally contributed.

Pre-registration workshop NeurIPS (2020), Vancouver, Canada.

set of critical challenges and downsides, especially in scenarios where deep learning models are used. The performance of an FL algorithm is found to be highly sensitive to both the system- and hyper-parameters of the model (for instance, a deep neural network) [6]. In practice, exploring the right set of configuration settings for an FL algorithm is a costly and arduous task. The primary reason being that training FL algorithms in a simulation setting is much slower than the conventional DL approaches since it involves training multiple models sequentially.

In this paper, we present `FedPerf`, an empirical study on five prominent FL algorithms to capture the relation between the parameters and performance. To provide for the unique scenario created by FL, we define Federated Learning System Parameters (FLSPs). These FLSPs are an added complexity in FL over a traditional ML system. For example, one significant FLSP is the skewness in the Non-IID data distributed among the client devices. A highly skewed data could comprise a scenario with each client containing only a single label data in a multi-class classification problem. Other common FLSPs include the number of participating clients, the communication and processing budget of individual clients, and fraction of stragglers. We are inspired by the previous work from Zhang and Wallace [7], and Semwal et al. [8], which presents a similar analysis on convolutional neural networks and Transfer Learning [9], respectively. It is envisaged that this work will significantly reduce the efforts of practitioners and researchers expend in finding the right setting. Here we will explore and report the results of a large set of experiments on a total of five different state-of-the-art FL algorithms. Many of the recent work in FL either present a conceptual survey or only discuss the mean accuracies for their own set of selected hyperparameters values. However, we found that the performance of FL algorithms is highly susceptible to the choice of constant parameters. We hypothesise that choosing the right FL algorithm and hyperparameters for the given FLSPs while being able to tune the FLSPs is not a trivial task. Furthermore, we are interested in identifying the inherent limitations of the FL system. Developing a thorough understanding of the FL system will help us lay the groundwork for expanding FL with Secure Multi-Party Computation (SMPC), Homomorphic Encryption (HE), and Differential Privacy (DP).

We explore the following aspects of FL:

1. How do we characterise the effect of FLSPs on the performance of an FL system?
2. How can we formulate a single easy-to-use metric that can explain the performance of an FL system?
3. Given FLSPs for a system, can we identify the best performing FL algorithm?

2 Related Work

Lim et al. .[10] discuss challenges in FL that occur in highly heterogeneous systems. These include communication costs, resource allocation, and privacy and security in the implementation of FL at scale. Similarly, Aledhari et al. [11] provide industry-specific obstacles in FL, along with detailed service use-cases. However, both of these manuscripts do not address the complexity involved in implementing and comparing different FL algorithms. Keeping these papers in mind, we further investigate the motivation behind FLSPs and Federated Learning Metric (FLM).

FLSPs: Understanding the every-increasing list of system parameters, which affect FL is hard to keep track off. Further, investigating every single FL system configuration is virtually and computationally infeasible. We do see an attempt in Hu et al. [12] to determine the important parameters in FL systems. They investigate unique parameters like dataset partitioning styles and diversity of datasets. Furthermore, the authors have made attempts to present metrics that could track these parameters separately. Similarly, in Li. et al. [13], we see a new parameter called *client fairness* being discussed, thus prompting the idea that there could be more undiscovered parameters which might be important and yet not being brought into the discussions. In, [14], the authors compare the performance of multiple FL algorithms for one FLSP – data distribution heterogeneity (either IID or non-IID). Based on the performance analysis reported in [14], the number of simulations for five algorithms, and a single FLSP, would be in the order of $O(mn)$ where m is the number of FL algorithms and n is the different variations of each FLSP. Thus, exhibiting the complexity of choosing FLSPs. Fig. 1 further showcases the complexity in tuning multiple FLSPs along with hyperparameters for an FL System. As can be seen from the figure, the number of experiments increases by polynomial times with the increase in the FLSPs. We believe understanding the

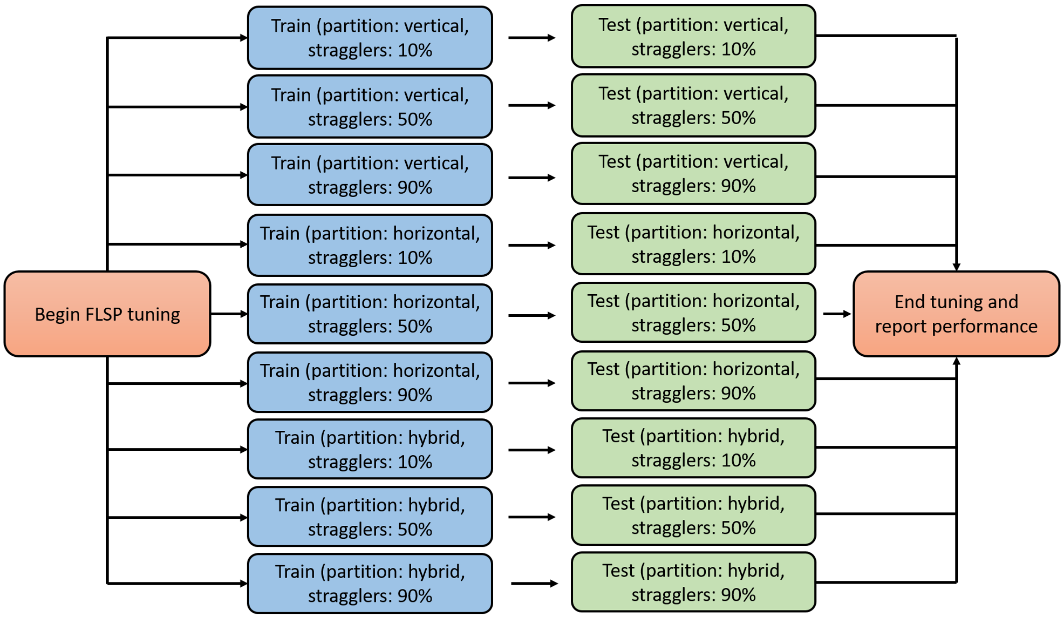
Begin FLSP tuning
Train (partition: vertical, stragglers: 10%)
Train (partition: vertical, stragglers: 50%)
Train (partition: vertical, stragglers: 90%)
Train (partition: horizontal, stragglers: 10%)
Train (partition: horizontal, stragglers: 50%)
Train (partition: horizontal, stragglers: 90%)
Train (partition: hybrid, stragglers: 10%)
Train (partition: hybrid, stragglers: 50%)
Train (partition: hybrid, stragglers: 90%)
Test (partition: vertical, stragglers: 10%)
Test (partition: vertical, stragglers: 50%)
Test (partition: vertical, stragglers: 90%)
Test (partition: horizontal, stragglers: 10%)
Test (partition: horizontal, stragglers: 50%)
Test (partition: horizontal, stragglers: 90%)
Test (partition: hybrid, stragglers: 10%)
Test (partition: hybrid, stragglers: 50%)
Test (partition: hybrid, stragglers: 90%)
End tuning and report performance

Figure 1: **FLSP Tuning**: A graphical demonstrations of the complexity of FLSP tuning to identify the performance of FL algorithms. Here, we have to discretise continuous FLSP spaces necessarily, (here the percentage of stragglers in the system) to reduce the parameter search complexity. Even then, we have nine configurations. Furthermore, adding in the hyperparameter search leads to a doubling of the parameter tuning complexity.

relationship between FLSPs and the performance of FL Systems, could help reduce the time required to simulate FL algorithms.

In this paper, we, therefore, attempt to provide a broad definition inclusive of all current and potential future FLSPs. Furthermore, we provide comprehensive coverage of FLSPs known through the literature and our findings. Finally, we provide an empirical correlation between FLPSs and FL performance.

FLM: As reported in Hu et al. [12], tracking performance is increasingly complex since there could be multiple metrics for the *same* FL system. Liang et al. [15] provides an industry-specific empirical coverage of the parameters that are valuable in real datasets, such as algorithm robustness and fairness between corporations. As evidenced above, the literature is not yet complete enough to provide simple easy-to-track metrics in FL. This paper attempts to provide clarity on these metrics. We investigate how we can reduce the number of metrics to be tracked and come up with a simple easy-to-track metric (denoted by FLM).

3 Methodology and experimental protocol

Since we wish to understand the impact of FLSPs on the system performance, we will begin with a simple experiment over baseline federated datasets from LEAF [16]. Table 1 presents a list of currently identified FLSPs. We tune the five algorithms with respect to the FLSPs, by tuning one FLSP at a time while keeping the rest constant. The algorithm performance is measured over the metrics suggested in Table 2. We expect this tuning to get harder with increasing FLSPs, as demonstrated by Fig. 1. Through this empirical analysis of the FL algorithms over varying FLSP configurations, we expect to find matching between the optimal algorithm to be used for the given FLSPs. Furthermore, to improve performance measurement, we will understand the various suggested metrics in literature and provide an algorithm-matching based on the right FLM and the given FLSPs. Our experiments will primarily consist of using the LEAF datasets combined with five prominent FL algorithms in the literature to come up with this matching. Precisely, the experiments will be based on:

FLSP	Types
Datasets	LEAF datasets and Synthetic datasets
Data Partitioning	Vertical, Horizontal and Hybrid
Data Variety	Text, audio and video
Data Skewness	Difference in data size across clients
Data Distribution Heterogeneity	IID and Non-IID
Communication	Network bitrate, Number of global rounds
Number of clients	Vary over possible number of clients
Stragglers	Percentage varying from 0%-95%
ML Models	Choosing multiple ML models appropriate for the task
Synchronicity	Synchronous and Asynchronous
Client Fairness	Maximum performance difference (in %) the over clients
Computational power	number of local rounds

Table 1: FLSP: List of system parameters which affect the performance of FL algorithms

- the variation in the FLSPs,
- performance measurement over multiple metrics and determining the right FLM,
- computing over a multitude of baseline FL datasets and algorithms.

FLSPs: Prompted from previous literature works [12], [15] we first list the possible FLSPs in Table 1. Due, to the limited scope of this article, we decided to solely focus on the parameters related to FL. We do understand that a lot of supplementary parameters exist for DP, SMPC, and HE. We however, defer this work to future articles.
Metrics: Several metrics are suggested to keep track of how an FL system performs. We list the proposed metrics in Table 2. We realise that some metrics target the budget available while others define the performance of the algorithm and appropriately segregate them. Therefore, our experiments involve tracking each one of these metrics to understand how the FLSPs and the choice of algorithms affect them. Similar to FLSPs, we postpone discussion of metrics relating to DP, SMPC, and HE to the future work.

Metrics	Description
Model	Accuracy and Loss
Fairness	Similar model performance over clients
Communication Cost	Number of global rounds, data transmitted
Computational Power	Number of local rounds, Convergence

Table 2: FL Metrics

FL algorithms and datasets: We decide to start with five baseline Federated Algorithms - FedAvg [3], FedProx [17], FSVRG [18], CO-OP [19] and q-FedAvg [13]. These cover different optimisation methods, synchronicity and fairness in FL. We use the baseline datasets from the LEAF.
Embracing Open-Science: Finally, besides the FLSMs, we realise that a number of articles in FL, lack implementations of the proposed algorithm, therefore requiring heavy hyperparameter tuning and adjustments to the proposed ML model. We, therefore, will provide implementations of all our code, where new implementations will continually be added.

References

[1] P. Voigt and A. Von dem Bussche, "The eu general data protection regulation (gdpr)," *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 2017.

[2] G. J. Annas, "Hipaa regulations — a new era of medical-record privacy?" *New England Journal of Medicine*, vol. 348, no. 15, pp. 1486–1490, 2003.

[3] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*, 2017, pp. 1273–1282.

[4] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage, "Federated learning for mobile keyboard prediction," *arXiv preprint arXiv:1811.03604*, 2018.

[5] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečnỳ, S. Mazzocchi, H. B. McMahan *et al.*, "Towards federated learning at scale: System design," *arXiv preprint arXiv:1902.01046*, 2019.

[6] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," ser. Proceedings of Machine Learning Research, A. Singh and J. Zhu, Eds., vol. 54. Fort Lauderdale, FL, USA: PMLR, 20–22 Apr 2017, pp. 1273–1282.

[7] Y. Zhang and B. Wallace, "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification," in *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Taipei, Taiwan: Asian Federation of Natural Language Processing, Nov. 2017, pp. 253–263.

[8] T. Semwal, P. Yenigalla, G. Mathur, and S. B. Nair, "A practitioners' guide to transfer learning for text classification using convolutional neural networks," in *Proceedings of the 2018 SIAM International Conference on Data Mining*. SIAM, 2018, pp. 513–521.

[9] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[10] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y. C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surveys Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020.

[11] M. Aledhari, R. Razzak, R. M. Parizi, and F. Saeed, "Federated learning: A survey on enabling technologies, protocols, and applications," *IEEE Access*, vol. 8, pp. 140 699–140 725, 2020.

[12] S. Hu, Y. Li, X. Liu, Q. Li, Z. Wu, and B. He, "The oarf benchmark suite: Characterization and implications for federated learning systems," *arXiv preprint arXiv:2006.07856*, 2020.

[13] T. Li, M. Sanjabi, A. Beirami, and V. Smith, "Fair resource allocation in federated learning," *arXiv preprint arXiv:1905.10497*, 2019.

[14] A. Nilsson, S. Smith, G. Ulm, E. Gustavsson, and M. Jirstrand, "A performance evaluation of federated learning algorithms," in *Proceedings of the Second Workshop on Distributed Infrastructures for Deep Learning*, 2018, pp. 1–8.

[15] Y. Liang, Y. Guo, Y. Gong, C. Luo, J. Zhan, and Y. Huang, "An isolated data island benchmark suite for federated learning," *arXiv preprint arXiv:2008.07257*, 2020.

[16] S. Caldas, P. Wu, T. Li, J. Konečnỳ, H. B. McMahan, V. Smith, and A. Talwalkar, "Leaf: A benchmark for federated settings," *arXiv preprint arXiv:1812.01097*, 2018.

[17] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *arXiv preprint arXiv:1812.06127*, 2018.

[18] J. Konečnỳ, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," *arXiv preprint arXiv:1610.02527*, 2016.

[19] Y. Wang, "Co-op: Cooperative machine learning from mobile devices," 2017.