Poster: Exploiting Data Heterogeneity for Performance and Reliability in Federated Learning*

Yuanli Wang
University of Minnesota, Twin cities
wang8662@umn.edu

Dhruv Kumar
University of Minnesota, Twin Cities
dhruv@umn.edu

Abhishek Chandra
University of Minnesota, Twin Cities
chandra@umn.edu

I. Introduction

Federated Learning [1] enables distributed devices to learn a shared machine learning model together, without uploading their private training data. It has received significant attention recently and has been used in mobile applications such as search suggestion [2] and object detection [3]. Federated Learning is different from distributed machine learning due to the following reasons: 1) **System heterogeneity:** federated learning is usually performed on devices having highly dynamic and heterogeneous network, compute, and power availability. 2) **Data heterogeneity (or statistical heterogeneity):** data is produced by different users on different devices, and therefore may have different statistical distribution (non-IID).

Prior work has tried to address the challenges arising due to system and data heterogeneity in federated learning [4]–[9]. As an example, TiFL [4] selects the set of devices participating in training in each iteration based on their computational speed to accelerate training and mitigate straggler effect. The prior work has mainly considered system heterogeneity in order to achieve good system metrics (e.g. training speed, energy consumption), without taking into account data heterogeneity issues. Also, fault tolerance of devices has not been looked at.

In this work, we look at the impact of data heterogeneity in selecting devices during training in federated learning, especially for edge devices. More specifically, we try to find answer to the question: which subset of devices should be selected for training in order to ensure high accuracy, while accelerating training speed and achieving fault tolerance? To this end, we conduct detailed experiments to simulate various scenarios for selecting devices on MNIST, MNIST-Variations Dataset and LEAF framework Our results indicate that federated learning is quite robust to intermittent dropping of devices. Additionally, the accuracy for a permanently dropped device will not drop as long as its data distribution is represented by some other devices participating in the training process. Based on these results, we conclude that utilizing statistical metrics (along with system metrics) can lead to significant improvement in the performance and reliability in federated learning without any loss in accuracy.

II. Preliminary Results

Datasets. We use two datasets. Each dataset has a different type of data heterogeneity:

- **Data heterogeneity on class labels.** In this case, different devices have different distribution of class labels. For this, we use MNIST dataset [10]. All the images are 28x28 pixels hand-writing numbers labeled from [0-9].
- **Data heterogeneity within the same class.** Here, different devices have different data distribution for the same class. For example, different user may write the same hand-writing number with different style. For this, we use MNIST Variations dataset [11], which has 5 different styles for each number.

For each of the above mentioned heterogeneity, we partition each dataset into 100 partitions. Each partition is assigned to one client. We randomly select 20 clients from these 100 clients for each training epoch.

Testbed. We extend the distributed federated learning framework LEAF [12] to simulate different drop patterns.

Model. We use the Convolutional Neural Network as in [12]. The overall accuracy is the average of the test accuracy on all devices. Wherever required, we also measure the accuracy on each device separately. We run the training for 1000 epochs.

We next look at the various scenarios for device selection.

A. Intermittent availability of devices

In a real environment, edge devices, like mobile phones, may participate in training at the beginning but leave after some epochs due to reasons such as lack of power supply, network disconnection etc. We simulate this by dropping some devices from some training epochs while ensuring that every device has participated in at least one epoch.

We use MNIST dataset for evaluation. We adopt the setting from [13] to ensure that each device contains data from at most two classes. We drop out 0, 5 or 15 clients out of 20 clients in each epoch, and plot the overall accuracy of the global model on 100 clients after each training epoch in Fig. 1.

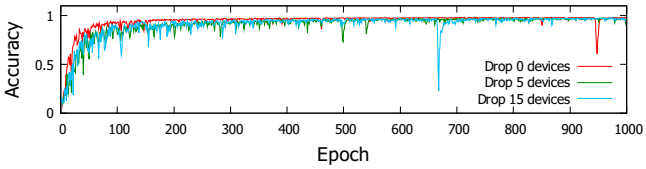

Fig. 1: Accuracy of each epoch when devices drop intermittently

Observation. In Fig. 1, we see that dropping devices intermittently does not significantly impact the overall accuracy.

*This research was supported in part by NSF under grant CNS-1717834.

978-1-7281-5943-0/20/$31.00 ©2020 IEEE
DOI 10.1109/SEC50012.2020.00023

During the initial phases of training, there is more variation in the accuracy for scenarios where the devices are dropped. But as the training progresses, the variation continues to reduce, eventually becoming insignificant.

Conclusion. This result shows that federated learning is robust to intermittent dropping of devices. Hence, we may not need specially designed fault tolerance policy for such cases.

B. Permanent dropping of devices

In real environments, due to system heterogeneity, some devices might be significantly slower than others. Therefore, existing systems such as TiFL [4] partition and select devices based on system metrics (e.g. training speed). It is possible that the slower devices may not get any opportunity to participate in training. To simulate this scenario, we pre-select the set of devices which can participate in the training. Then, in each training epoch, we only select the devices from this pre-selected set. Effectively, this leads to some devices not participating in the entire training.

1) Data heterogeneity on class labels: We adopt the setting from [13] to partition 100 clients to 10 groups. Each group contains 10 clients and will be assigned 2 classes. We ensure that a device in any group will have training data only from the 2 classes assigned to that group as shown in Table I. We implement two dropping policies: 1) randomly pre-select some clients to drop 2) pre-select an entire group of devices to drop. For each policy, we drop 80 out of 100 devices, and measure the accuracy of trained global model on the local test dataset of each device. The results are shown in Fig. 2.

TABLE I: Partition of training data on 100 devices

Device Group No.	0	1	2	3	4
Classes	6,7	1,4	5,9	2,3	0,4
Device Group No.	5	6	7	8	9
Classes	2,5	6,8	0,9	7,8	1,3

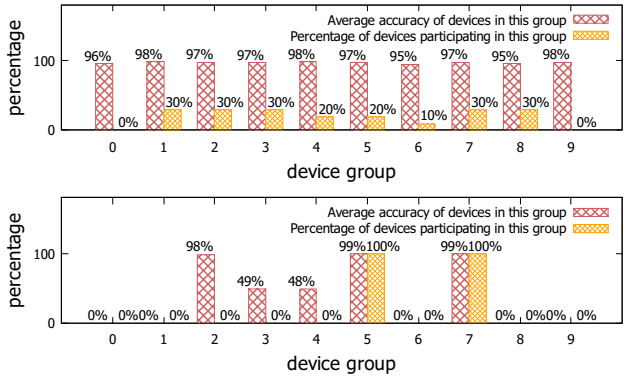

Fig. 2: Top: Devices are randomly dropped permanently Bottom: Drop entire groups of devices permanently

Observation. In Fig. 2 Top, there is no drop in accuracy for any group. We conclude that if at least one client from each group participates in training, the accuracy of all devices in this group can be guaranteed. In Fig. 2 Bottom, we see that the groups which have been dropped completely experience a significant drop in accuracy. The accuracy drop for dropped groups which have some of their class labels in participating groups is less as compared to the groups whose class labels are not present in any of the participating groups.

2) Data heterogeneity within the class labels: We further consider data heterogeneity within the same class. In this experiment, we partition 100 clients to 5 groups. Each group contains 20 clients. All devices have all the class labels but the devices are grouped by different image styles from MNIST Variations Dataset (See Table II).

TABLE II: Partition of training data on 100 devices

Device Group No.	0	1	2	3	4
Style	basic MNIST	Rotated MNIST + background images	MNIST + background images	Rotated MNIST	MNIST + random background

Observation. We dropped 90 clients, and only leave 10 clients from basic MNIST group. The evaluation result is shown in Fig. 3. The result shows that only the basic MNIST group achieved good accuracy while other groups experience varying drops in accuracy. We conclude that data heterogeneity within the same class label have similar impact as previous experiments.

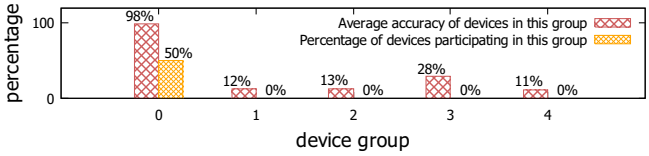

Fig. 3: Average accuracy and participation rate of each group

Conclusion. The accuracy for a permanently dropped device will not drop as long as its data distribution is represented by some other devices participating in the training process.

III. Conclusion and Proposed Solutions

In this work, we discuss the impact of data heterogeneity in federated learning. Our findings are as follows:

- Dropping clients intermittently (devices participating only in some epochs) will not decrease the accuracy.
- Dropping clients permanently might decrease the overall accuracy of the global model. That depends on the data heterogeneity of training data on different devices.

Based on these findings, we propose the following:

- Partition the devices such that the devices in each partition have similar data distribution (identified from prior training cycles). Then select the fastest devices from each partition for each training round, or give higher selection priority to fastest devices. This will improve the training speed without any compromise in accuracy.
- Reliable devices (i.e. devices that drop less frequently) having similar data distribution as that of less reliable devices, should be given higher selection priority for training.

REFERENCES

[1] K. A. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. M. Kiddon, J. Konečný, S. Mazzocchi, B. McMahan, T. V. Overveldt, D. Petrou, D. Ramage, and J. Roselander, "Towards federated learning at scale: System design," in *SysML 2019*, 2019. [Online]. Available: https://arxiv.org/abs/1902.01046

[2] T. Yang, G. Andrew, H. Eichner, H. Sun, W. Li, N. Kong, D. Ramage, and F. Beaufays, "Applied federated learning: Improving google keyboard query suggestions," 2018. [Online]. Available: https://arxiv.org/abs/1812.02903

[3] Y. Liu, A. Huang, Y. Luo, H. Huang, Y. Liu, Y. Chen, L. Feng, T. Chen, H. Yu, and Q. Yang, "Fedvision: An online visual object detection platform powered by federated learning," in *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, 2020, pp. 13 172–13 179. [Online]. Available: https://aaai.org/ojs/index.php/AAAI/article/view/7021

[4] Z. Chai, A. Ali, S. Zawad, S. Truex, A. Anwar, N. Baracaldo, Y. Zhou, H. Ludwig, F. Yan, and Y. Cheng, "Tifl: A tier-based federated learning system," in *Proceedings of the 29th International Symposium on High-Performance Parallel and Distributed Computing*, ser. HPDC '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 125–136. [Online]. Available: https://doi.org/10.1145/3369583.3392686

[5] L. Liu, J. Zhang, S. H. Song, and K. B. Letaief, "Client-edge-cloud hierarchical federated learning," 2019. [Online]. Available: https://arxiv.org/abs/1905.06641

[6] Y. Zhan, P. Li, and S. Guo, "Experience-driven computational resource allocation of federated learning by deep reinforcement learning," in *2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2020, pp. 234–243.

[7] Z. Xu, L. Li, and W. Zou, "Exploring federated learning on battery-powered devices," in *Proceedings of the ACM Turing Celebration Conference - China*, ser. ACM TURC '19. New York, NY, USA: Association for Computing Machinery, 2019.

[8] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "When edge meets learning: Adaptive control for resource-constrained distributed machine learning," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, 2018, pp. 63–71.

[9] D. Kumar, A. A. Ramkumar, R. Sindhu, and A. Chandra, "Decaf: Iterative collaborative processing over the edge," in *HotEdge*, 2019.

[10] *The MNIST database of handwritten digits*, http://yann.lecun.com/exdb/mnist/.

[11] *MNIST Variations*, https://sites.google.com/a/lisa.iro.umontreal.ca/public_static_twiki/variations-on-the-mnist-digits.

[12] S. Caldas, S. M. K. Duddu, P. Wu, T. Li, J. Konečný, H. B. McMahan, V. Smith, and A. Talwalkar, "Leaf: A benchmark for federated settings," 2018. [Online]. Available: https://arxiv.org/abs/1812.01097

[13] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," 2018. [Online]. Available: https://arxiv.org/abs/1806.00582