Poster Abstract: Fair Training of Multiple Federated Learning Models on Resource Constrained Network Devices

Marie Siew
Carnegie Mellon University
msiew@andrew.cmu.edu

Shoba Arunasalam
Carnegie Mellon University
sarunasa@andrew.cmu.edu

Yichen Ruan
yicruan@gmail.com

Ziwei Zhu
ziweizhu@andrew.cmu.edu

Lili Su
Northeastern University
l.su@northeastern.edu

Stratis Ioannidis
Northeastern University
ioannidis@ece.neu.edu

Edmund Yeh
Northeastern University
eyeh@ece.neu.edu

Carlee Joe-Wong
Carnegie Mellon University
cjoewong@andrew.cmu.edu

ABSTRACT

Federated learning (FL) is an increasingly popular form of distributed learning across devices such as sensors and smartphones. To amortize the effort and cost of setting up FL training in real world systems, in practice multiple machine learning tasks may be trained during one FL execution. However, given that the tasks have varying complexities, naïve methods of allocating resource-constrained devices to work on each task may lead to highly variable performance across the tasks. We instead propose an α-fair based allocation algorithm that dynamically allocates tasks to users during multi-model FL training, based on the prevailing loss levels.

KEYWORDS

Federated Learning, Resource Allocation

ACM Reference Format:
Marie Siew, Shoba Arunasalam, Yichen Ruan, Ziwei Zhu, Lili Su, Stratis Ioannidis, Edmund Yeh, and Carlee Joe-Wong. 2023. Poster Abstract: Fair Training of Multiple Federated Learning Models on Resource Constrained Network Devices. In *The 22nd International Conference on Information Processing in Sensor Networks (IPSN '23), May 9–12, 2023, San Antonio, TX, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3583120.3589835

1 INTRODUCTION

Federated learning (FL) is a form of distributed learning, where a machine learning task is collaboratively trained across devices/users (called clients), under the coordination of a central server [4]. For instance, the devices may be smart home sensors [7]. In FL, clients perform local updates on local datasets, and upload model parameters to the central server, which aggregates these updates to obtain a global model. It sends the global model back to clients. The process iterates until convergence. Single model FL has been widely studied across different perspectives, such as optimizing client selection [6] and analyzing the impact of clients' data heterogeneity [5].

Setting up federated training in the real world can be expensive and effort-incurring, involving recruiting clients and coordinating a common training time for all clients. To reduce this operational expense and effort, Multiple Model Federated Learning (MMFL), where multiple machine learning tasks (models) can be trained during one federated learning execution, has been proposed [1]. Nevertheless, network devices such as mobile and Internet of Things devices are often resource constrained, having heterogeneous and possibly limited amount of CPU cycles and battery levels. Under these resource constraints, we will consider the case where each client is allocated one task during each local training iteration. [1] showed that when each client is allocated one task each local iteration, there is no sharp drop in accuracy compared to single model FL.

[1] studied the case where the multiple models (tasks) are of similar or equal complexity. Nevertheless, the multiple tasks in MMFL would often be different and have differing complexities. For example, clients might simultaneously train models of different sizes to predict the same quantity, allowing devices to choose which model they should deploy according to their available resources. Allocating clients to tasks via a round robin and other uniform strategies may result in simpler models being trained well, but more complex models being underfit. **Allocating tasks to users in a way which achieves *fairness* with respect to the tasks' performance, i.e., converged accuracy levels, is the main challenge we address.** We approach this problem through α-fairness, which is commonly used for fair resource allocation in networking [2] and has been used for fairness across clients in FL [3]. We propose an α-fair based algorithm where users are dynamically allocated tasks in MMFL, based on the prevailing loss levels of the tasks. Our algorithm helps to ensure that effort is spread across the MMFL tasks in accordance with their difficulty, helping to achieve fairness with respect to task performances. Preliminary results show that our algorithm achieves a higher minimum accuracy than the random allocation baseline, and a lower variance across converged task accuracy levels.

2 SYSTEM OVERVIEW

Multiple model federated learning (MMFL): Consider a network of K devices (clients) jointly training S machine learning tasks. As seen in Fig. 1, unlike traditional FL where the clients aim to train one task, in *MMFL*, the clients aim to train multiple (S) tasks. Each client k trains its local models, and uploads its model parameters $\mathbf{w}_{k,s}$ (with $s \in \mathbb{S}$ corresponding to the specific task) to a central aggregating server, which aggregates these updates to obtain a global model. The global model parameters would be sent back to clients. The process iterates until convergence. The clients collectively seek to obtain converged model weights that minimize the global loss. Letting client k's local loss for task s under model w_s be $F_{k,s}(w_s)$,

IPSN '23, May 9–12, 2023, San Antonio, TX, USA

ACM ISBN 979-8-4007-0118-4/23/05.
https://doi.org/10.1145/3583120.3589835

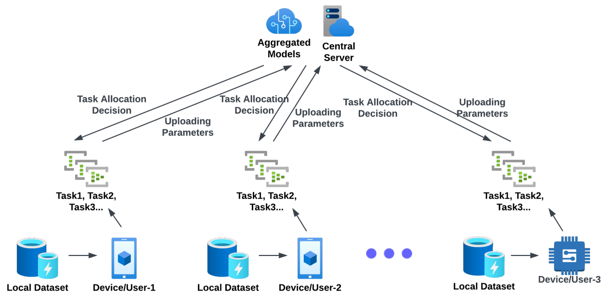

Figure 1: Multiple Model Federated Learning

the global loss for each task s will be $f_s(w_s) = \sum_{k=1}^{K} p_k F_{k,s}(w_s)$, where the weight p_k is the fraction of the size of client k's dataset, relative to the entire dataset over all clients. Our objective is to minimize the weighted sum of losses across all tasks:

$$\min_{w_1, w_2, \dots w_S} \sum_{s=1}^{S} q_s f_s(w_s), \tag{1}$$

where q_s is the weight given to task s.

Resource constrained devices: Network devices may have other background tasks, and some devices may have lower battery levels or less CPU cycles available. We respect these constraints by allocating only one task to each client, per local training iteration.

Fairness over tasks: We want to maintain good performance across all tasks. If round robin allocation is used, simpler models (tasks) will be trained well (lower loss) and at a faster rate, at the expense of insufficiently trained more difficult models. Instead, at each local iteration t we will *dynamically allocate* tasks across clients, based on the prevailing global losses of the various tasks. This helps to allocate effort according to the prevailing performance levels of the different tasks: a task with current higher loss will be allocated to more users.

In the α-fair utility function, the parameter α controls the efficiency-fairness tradeoff. In our MMFL problem, we use the concept of α-fairness to attain a fair performance across tasks. Here, our α-fair global loss function for task s will be f_s^{α}. Our α-fair objective is

$$\min_{w_1, w_2, \dots w_S} \sum_{t=1}^{S} q_s f_s^{\alpha}(w_t), \tag{2}$$

where q_s is the weight given to task s. Our objective is to minimize the weighted sum of the alpha-fair global losses $f_s^{\alpha}(w_s)$ across all tasks, such that the converged accuracy levels across tasks are as uniform as possible.

Dynamic Client-Task Allocation: At each local training iteration, clients will be allocated tasks based on which task *currently* has the highest global loss:

$$\operatorname*{argmax}_{s \in \mathbb{S}} f_s^{\alpha}(w_s) \tag{3}$$

To prevent the multiple model federated learning training process from being stuck at local minimum points, we will add randomness in the client-task allocation process. At each local iteration, tasks will be selected for clients according to the following probabilities:

$$\mathbf{p} = \left[\frac{f_1^{\alpha}}{\sum_{s \in \mathbb{S}} f_s^{\alpha}}, \frac{f_2^{\alpha}}{\sum_{s \in \mathbb{S}} f_s^{\alpha}}, \dots, \frac{f_S^{\alpha}}{\sum_{s \in \mathbb{S}} f_s^{\alpha}} \right] \tag{4}$$

	Average Accuracy	Min Accuracy	Variance
Baseline	87.06%	83.05%	21.34
$\alpha = 1$	87.20%	**84.40%**	**12.87**
$\alpha = 2$	87.00%	**84.30%**	**12.41**
$\alpha = 3$	86.32%	**84.33%**	**9.90**

Table 1: We compare our proposed algorithm to the baseline (corresponding to $\alpha = 0$, with tasks being randomly allocated to clients). Our client-task allocation strategy achieves a fairer training over tasks, achieving a higher minimum accuracy, and lower variance in accuracies across tasks.

3 EVALUATION

We evaluate our proposed algorithm using an MMFL setting, with 30 clients undertaking 3 tasks of varying complexities: a single layer perceptron, a multiple layer perceptron, and a convolutional neural network, all classifying images from the Mnist dataset.

We compare our α-fairness based algorithm (where $\alpha = 1, 2, 3$) to the baseline (corresponding to $\alpha = 0$) in which all users have equal probability of working on each task. Table 1 illustrates our results: Our algorithm achieves a more fair result, in terms of a *higher minimum accuracy* amongst tasks, than the baseline. This is because, our algorithm dynamically allocates tasks to clients based on which task has a higher prevailing α-fair loss value f_s^{α}, allocating effort according to the relative difficulty of the task. We see that the difference in average accuracy achieved by our algorithm and the baseline is minimal, with our algorithm even outperforming the baseline when $\alpha = 1$. It can be seen that our algorithm achieves a *lower variance* with respect to the converged model accuracies over tasks, hence attaining fairer training performances over the tasks.

4 CONCLUSION

In this work, we proposed a strategy for fair training across multiple machine learning tasks, in multiple model FL, when devices have resource constraints. Our strategy consists of dynamically allocating tasks to clients each iteration, based on which task currently has a higher α-fair loss. In future work, we will consider dynamic allocation in light of heterogeneous resources across clients.

ACKNOWLEDGMENTS

This work is supported by the SUTD PPF, under the Singapore MOE START Scheme, and by NSF CNS-2106891 and CNS 2107062.

REFERENCES

[1] Neelkamal Bhuyan, Sharayu Moharir, and Gauri Joshi. 2022. Multi-Model Federated Learning with Provable Guarantees. *arXiv preprint arXiv:2207.04330* (2022).
[2] Tian Lan, David Kao, Mung Chiang, and Ashutosh Sabharwal. 2010. *An axiomatic theory of fairness in network resource allocation.* IEEE.
[3] Tian Li, Maziar Sanjabi, Ahmad Beirami, and Virginia Smith. 2019. Fair resource allocation in federated learning. *arXiv preprint arXiv:1905.10497* (2019).
[4] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics.* PMLR, 1273–1282.
[5] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. 2020. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295* (2020).
[6] Wenchao Xia, Tony QS Quek, Kun Guo, Wanli Wen, Howard H Yang, and Hongbo Zhu. 2020. Multi-armed bandit-based client scheduling for federated learning. *IEEE Transactions on Wireless Communications* 19, 11 (2020), 7108–7123.
[7] Yuhang Yao, Mohammad Mahdi Kamani, Zhongwei Cheng, Lin Chen, Carlee Joe-Wong, and Tianqiang Liu. 2023. FedRule: Federated Rule Recommendation System with Graph Neural Networks. In *Accepted to ACM/IEEE IoTDI.*