Dynamic Attention-based Communication-Efficient Federated Learning

Zihan Chen[1,2], Kai Fong Ernest Chong[1], Tony Q.S. Quek[1]

[1]Singapore University of Technology and Design (SUTD), [2]National University of Singapore (NUS)

Abstract

To address data heterogeneity and communication limitation in federated learning (FL)[1,2,3,5], we propose a new adaptive training algorithm `AdaFL`, which comprises:

- an attention-based client selection mechanism for a fairer training scheme among the clients;
- a dynamic fraction method to balance the trade-off between performance stability and communication efficiency.

Experimental results show that our `AdaFL` algorithm outperforms the usual `FedAvg` algorithm, and can be incorporated to further improve various state-of-the-art FL algorithms, with respect to three aspects: model accuracy, performance stability, and communication efficiency.

Introduction

A good choice for this **fraction** is not clear.

- A small constant fraction method is widely used in existing work in FL.
- Large fractions methods are more stable and bring a slight convergence acceleration[6], at the expense of a larger communication cost.

To obtain training stability with relatively low communication cost, we shall consider a dynamic fraction method that captures the advantages of both small and large fractions.

The **selection probability** for each client is a measure of the "importance" of that client in a heterogeneous network. The selection probability distribution used in the usual FL is typically fixed. However, the relative contribution of each client is fluid. The "importance" of the clients may vary during training.

Overview of a typical round of `FedAvg`

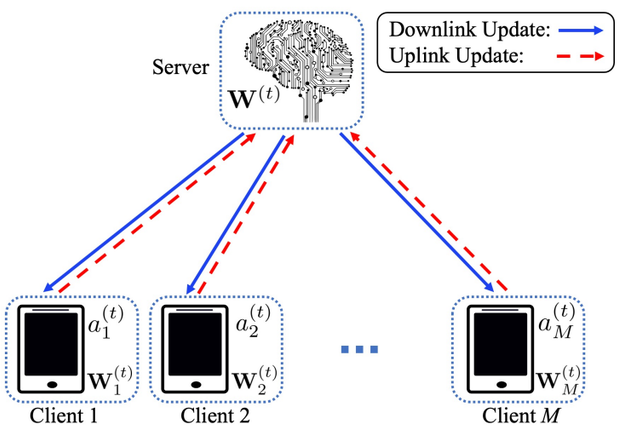
$$w_{t+1} \leftarrow w_t - \eta \nabla f(w_t) \qquad \nabla f(w_t) = \sum_{k\in\mathcal{S}_t} \frac{n_k}{n_{\mathcal{S}_t}} g_k$$

$$K = |\mathcal{S}_t| \qquad K = \gamma \cdot M$$

$$\mathbf{n} := \left[\frac{n_1}{n}, \ldots, \frac{n_M}{n}\right]$$

$$\mathbf{p} = [p_1, p_2, \ldots, p_M]$$

Fixed throughout training



System model with attention

We use $\boldsymbol{a}^t$ and γ^t to denote the attention vector and fraction respectively in communication round t.

Proposed Method

Attention mechanism

We use Euclidean distance as a measure of the model divergence of each local model, relative to the global model.

For **selected clients** in round *t*, the attention score would be updated as follows: Euclidean distance for model divergence

$$d_i^{(t)} = \left\| \mathbf{w}^{(t+1)} - \mathbf{w}_i^{(t)} \right\|_2$$

$$a_i^{(t+1)} = \alpha a_i^{(t)} + (1-\alpha) \cdot \frac{d_i^{(t)}}{\sum_{k\in\mathcal{S}_t} d_k^{(t)}} \sum_{k\in\mathcal{S}_t} a_k^{(t)}$$

For **unselected clients**, the attention score will have no change.

$$a_j^{(t+1)} = a_j^{(t)}$$

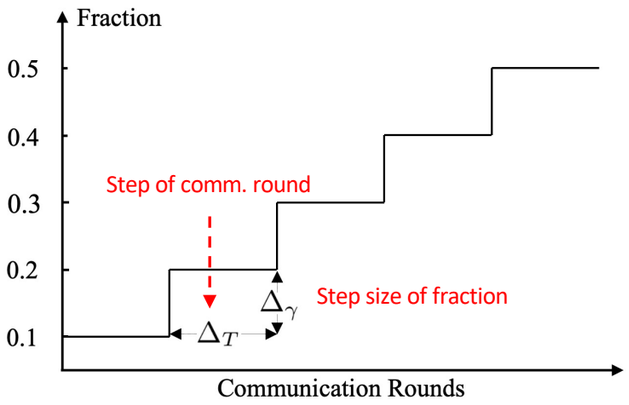
Client selection in round *t + 1* then follows the updated probability distribution, which equals to the updated attention scores $\boldsymbol{a}^{t+1}$. Here, α represents the decay rate of previous attention score.

stochastic vector

Dynamic fraction

The choice of constant fraction represents the trade-off between communication efficiency and performance stability.

To circumvent this trade-off, we drop the assumption on constant fraction, and propose a dynamic fraction method, which adopts different fractions during different training stages, with the fraction increasing progressively.



An example of dynamic fraction

In this work, we only consider fixed steps for fraction updates. It should be noted that, our methods work more generally for monotonically increasing fractions. The largest fraction $\gamma^T = 0.5$ is an arbitrary choice, which balanced the trade-off of stability and communication cost for large fraction case.

Proposed Algorithm - `AdaFL`

Our proposed algorithm `AdaFL` combines attention mechanism and dynamic fraction methods, which yields better communication efficiency with better performance stability.

The key difference and improvements of `AdaFL` are:

- It adaptively adjusts parameters during training;
- It complements most of the existing communication efficient FL algorithms;
- It can be incorporated to enhance the performance of existing popular FL optimization algorithms[2,3,4].

Algorithm 1 Adaptive Federated Learning (`AdaFL`)

Inputs: $M, T, \boldsymbol{\gamma}, \alpha, \mathbf{W}^{(1)}, \mathbf{n}$

$\mathbf{a}^{(1)} \leftarrow \mathbf{n}$
for $t = 1$ **to** T **do**
  $\mathbf{p} \leftarrow \mathbf{a}^{(t)}$ and $K \leftarrow \boldsymbol{\gamma}^{(t)} \cdot M$
  Server selects a subset of clients $\mathcal{S}_t$ of size $|\mathcal{S}_t| = K$ using probability distribution $\mathbf{p}$
  // local computation at clients
  for selected client $k \in \mathcal{S}_t$ **do**
    Client k downloads global model $\mathbf{W}^{(t)}$
    Client k computes local model $\mathbf{W}_k^{(t)}$
  // global computation at server
  Server computes a new global model by aggregation:

$$\mathbf{W}^{(t+1)} \leftarrow \textstyle\sum_{k\in\mathcal{S}_t} \frac{n_k}{n_{\mathcal{S}_t}} \mathbf{W}_k^{(t)}$$

  for selected client $i \in \mathcal{S}_t$ **do**
    Server updates $d_i^{(t)} \leftarrow \left\| \mathbf{w}^{(t+1)} - \mathbf{w}_i^{(t)} \right\|_2$ and

$$a_i^{(t+1)} \leftarrow \alpha a_i^{(t)} + (1-\alpha) \cdot \frac{d_i^{(t)}}{\sum_{k\in\mathcal{S}_t} d_k^{(t)}} \sum_{k\in\mathcal{S}_t} a_k^{(t)}$$

  for unselected client $j \notin \mathcal{S}_t$ **do**
    Server updates $a_j^{(t+1)} \leftarrow a_j^{(t)}$

Algorithm summary

Experiments

We evaluate our `AdaFL` algorithm on MNIST (Non-IID data partition with MLP model), CIFAR-10 (IID data partition with CNN model). We use $M = 100$ clients, $\alpha = 0.9$, starting fraction 0.1 and ending fraction 0.5. We report performance comparison on accuracy and communication efficiency[2,3,4].

Accuracy

Algorithm	MNIST		CIFAR-10	
	Average	Best	Average	Best
`AdaFL`	**91.13**	**91.64**	74.38	**76.17**
`Attn.-0.1`	88.92	91.30	73.13	74.91
`Attn.-0.5`	91.07	91.58	**74.42**	75.96
`Dyn.FedAvg`	90.33	91.19	74.33	75.04
`FedAvg-0.1`	88.68	91.05	72.88	74.82
`FedAvg-0.5`	90.40	91.21	73.67	75.31

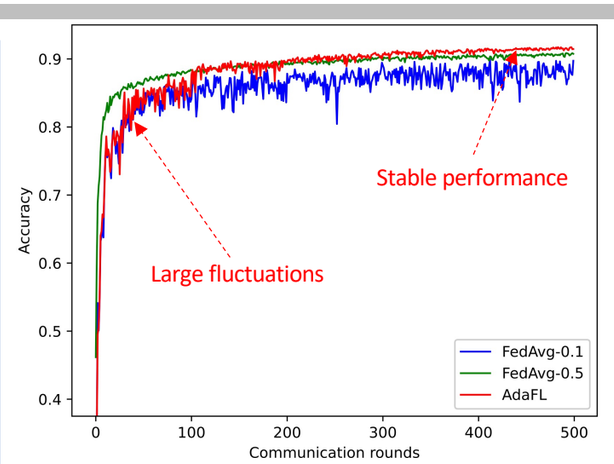
Ablation study

Ours
Attention
Dynamic
Baseline

Required comm. rounds — communication cost

Algorithm	MNIST		CIFAR-10
	90%	91%	73%
`AdaFL`	**423 (6690)**	**761 (18440)**	**683 (15320)**
`Attn.-0.1`	939 (9390)	1952 (19520)	1571 (15710)
`Attn.-0.5`	420 (21000)	741 (37050)	635 (31570)
`Dyn.FedAvg`	951 (20040)	1485 (44250)	1103 (26120)
`FedAvg-0.1`	1008 (10080)	2528 (25280)	1957 (19570)
`FedAvg-0.5`	570 (28500)	1232 (61600)	892 (44600)

Algorithm	MNIST		CIFAR-10	
	Average	Best	Average	Best
`AdaFL+FedProx`	**91.67**	**92.42**	**74.94**	**76.24**
`FedProx-0.1`	89.15	91.46	72.88	75.90
`FedProx-0.5`	90.81	91.55	73.57	76.12
`AdaFL+FedMix`	**90.52**	**91.30**	**73.27**	**75.05**
`FedMix-0.1`	88.37	90.61	71.53	73.43
`FedMix-0.5`	89.91	91.08	72.42	74.12
`AdaFL+SCAFFOLD`	**90.30**	**91.52**	**74.98**	**75.53**
`SCAFFOLD-0.1`	87.82	89.96	71.62	74.12
`SCAFFOLD-0.5`	89.73	90.82	73.50	74.77



Accuracy comparison of `AdaFL` with `FedAvg` on Non-IID MNIST

$\sum_{t=1}^{T^*} \boldsymbol{\gamma}^{(t)} \cdot M$ Calculation for total communication cost

Algorithm	MNIST	CIFAR-10
	91%	73%
`AdaFL+FedProx`	**821 (21600)**	721 (**16840**)
`FedProx-0.1`	2439 (24390)	1762 (17620)
`FedProx-0.5`	1084 (54200)	**658** (32900)
	90%	72%
`AdaFL+FedMix`	**852 (22600)**	**698 (15920)**
`FedMix-0.1`	2275 (22750)	1903 (19030)
`FedMix-0.5`	1241 (62050)	732 (36600)
	89%	72%
`AdaFL+SCAFFOLD`	**794 (19760)**	**672 (15600)**
`SCAFFOLD-0.1`	2252 (22520)	1981 (19810)
`SCAFFOLD-0.5`	1034 (51700)	725 (36250)

Conclusion and Further Work

`AdaFL` is a simple algorithm that can be easily incorporated into various state-of-the-art FL algorithms to obtain improvements on several aspects: model accuracy, performance stability, and communication efficiency.

Further work may include general dynamic fraction method and attention mechanism with imbalanced data.

Acknowledgement

This research is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG-RP-2019-015).

References

[1]. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In Artificial Intelligence and Statistics, pages 1273–1282. PMLR, 2017.

[2]. Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In Proceedings of Machine Learning and Systems, volume 2, pages 429–450, 2020.

[3]. Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In International Conference on Machine Learning, pages 5132–5143. PMLR, 2020.

[4]. Tehrim Yoon, Sumin Shin, Sung Ju Hwang, and Eunho Yang. Fedmix: Approximation of mixup under mean augmented federated learning. In International Conference on Learning Representations, 2021.

[5]. Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. IEEE Signal Processing Magazine, 37(3):50–60, 2020.

[6]. Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. In International Conference on Learning Representations, 2019.