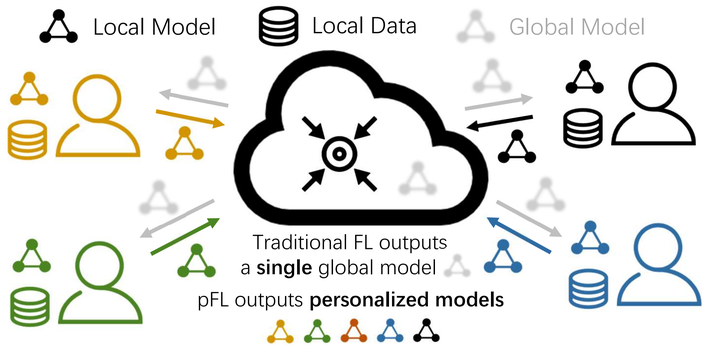
FedALA: Adaptive Local Aggregation for Personalized Federated Learning

Jianqing Zhang[1], Yang Hua[2], Hao Wang[3], Tao Song[1], Zhengui Xue[1], Ruhui Ma[1], Haibing Guan[1]

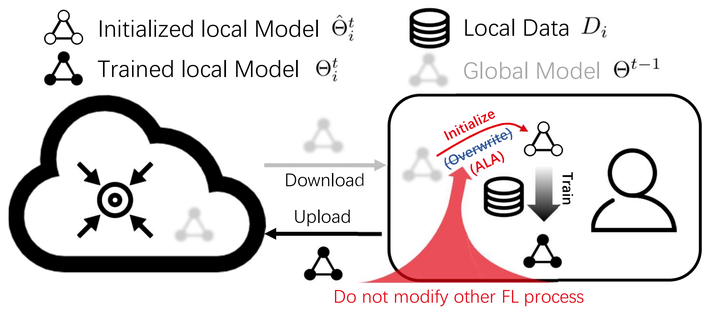
[1]Shanghai Jiao Tong University [2]Queen's University Belfast [3]Louisiana State University

Introduction

Background: Federated learning (FL) can leverage distributed user data while preserving privacy. A key challenge in FL is statistical heterogeneity (colorful icons), which results in poor generalization ability of the (blurred) global model on each client. To tackle this issue and achieve personalized requirements, personalized FL (pFL) was proposed and becomes popular.



Motivation: Most existing traditional FL and pFL methods overwrite local model with entire global model parameters. However, only the desired information that improves the quality of local model is beneficial for the client.

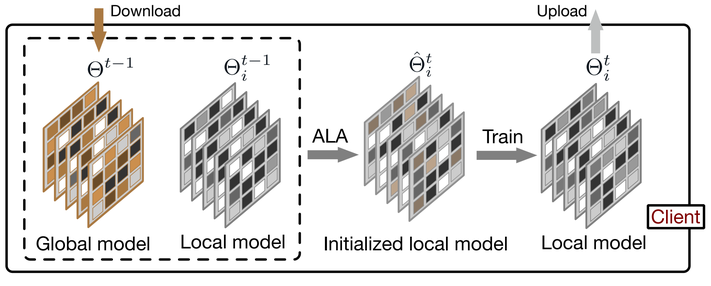


Goal: Replace overwriting with ***A**daptive **L**ocal **A**ggregation* (**ALA**) to precisely capture the desired information in the global model for each client without additional communication overhead in each iteration.

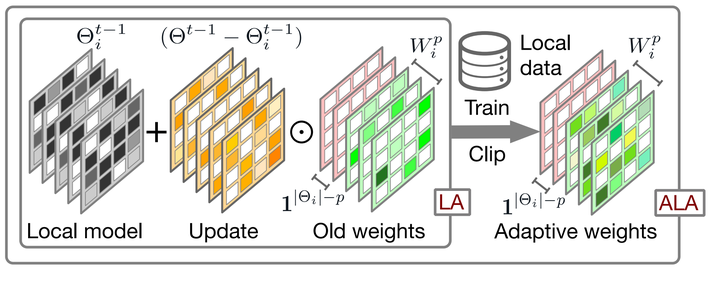
Key contributions:

- We propose a novel pFL method FedALA that adaptively aggregates the global model and local model towards the local objective to capture the desired information from the global model element-wisely.
- We empirically show the effectiveness of FedALA, which outperforms eleven SOTA methods by up to 3.27% in test accuracy without additional communication overhead in each iteration.
- Attributed to the minor modification of FL process, the ALA module in FedALA can be directly applied to existing FL methods to enhance their performance by up to 24.19% in test accuracy on Cifar100.

Overview: In FedALA, the local learning process on client i (t-th iteration).



The learning process in ALA ($p = 3$).



Adaptive Local Aggregation (ALA)

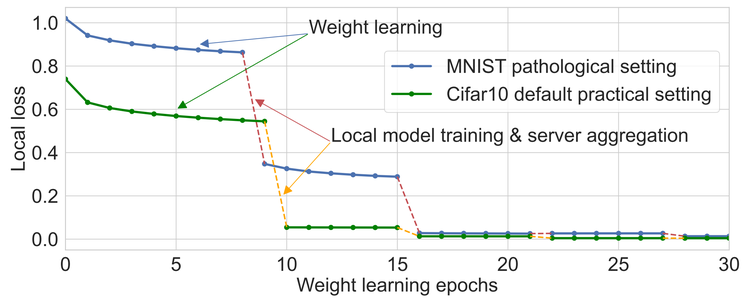
After downloading the global model Θ^{t-1}, we update the ALA weights W_i^p on only $s\%$ local data $D_i^{s,t}$ for adaptation:

$$W_i^p \leftarrow W_i^p - \eta \nabla_{W_i^p} \mathcal{L}(\hat{\Theta}_i^t, D_i^{s,t}; \Theta^{t-1}), \tag{1}$$

where the initialized local model $\hat{\Theta}_i^t$ is the aggregation of the global model and the local model Θ_i^{t-1}:

$$\hat{\Theta}_i^t := \Theta_i^{t-1} + (\Theta^{t-1} - \Theta_i^{t-1}) \odot [\mathbf{1}^{|\Theta_i|-p}; W_i^p], \tag{2}$$

we call the term $(\Theta^{t-1} - \Theta_i^{t-1})$ as "*update*", and we introduce a hyperparameter p to control the range of ALA by applying it on p higher layers and overwriting the parameters in lower layers. $\hat{\Theta}_i^t$ is simultaneously updated with W_i^p.



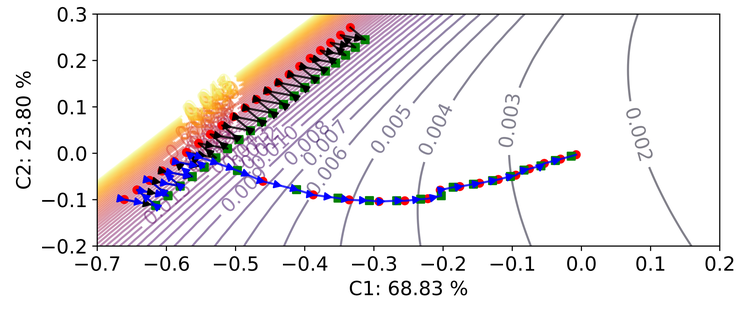
We observed that once we train W_i^p to converge in the start stage, it hardly changes later. In other words, W_i^p can be **reused**.
Besides, we can view updating W_i as updating $\hat{\Theta}_i^t$ in ALA:

$$\hat{\Theta}_i^t \leftarrow \hat{\Theta}_i^t - \eta(\Theta^{t-1} - \Theta_i^{t-1}) \odot (\Theta^{t-1} - \Theta_i^{t-1}) \odot \nabla_{\hat{\Theta}_i} \mathcal{L}_i^t. \tag{3}$$

In contrast to local model training (or fine-tuning) that only focuses on the local data, ALA can introduce the generic information in the global model.

Capture Client Desired Information

Compared to overwriting, ALA can correct the update direction for the local model with the desired information in the global model. We visualize the learning trajectory of the local model on client #4. We deactivate the ALA for FedALA in early iterations and activate it in subsequent iterations.



Without capturing the desired information in the global model, the update direction of local model is misled by the global model in FedAvg, as shown by the **black trajectory**. Once the ALA is activated, the update direction of local model is corrected to the local loss reducing direction, as shown by the **blue trajectory**.

Reduce Computation Overhead for ALA

The goal of introducing s (training W_i^p with $s\%$ local data) and p (applying ALA on p higher layers) is to reduce computation overhead for ALA. We can balance the performance and the computational cost by choosing a reasonable value for s (*e.g.*, $s = 80$). FedALA also performs well with $s = 5$. As for p, we can shrink the range of ALA with negligible accuracy decrease (*e.g.*, $p = 1$).

Items	$p=1$							$s=80$					
	$s=5$	$s=10$	$s=20$	$s=40$	$s=60$	$s=80$	$s=100$	$p=6$	$p=5$	$p=4$	$p=3$	$p=2$	$p=1$
Acc.	39.53	40.62	40.02	40.23	41.11	41.94	**42.11**	41.71	41.54	41.62	41.86	**42.47**	41.94
Param.	0.005							11.182	11.172	11.024	10.499	8.399	0.005

Computation and Communication Overhead

FedALA requires less total time and costs a little more time per iteration than FedAvg, which means that FedALA converges faster than FedAvg. Meanwhile, the FedALA does not introduce additional communication overhead in each iteration compared to FedAvg, but has less overall communication overhead due to faster convergence. Note that, $\alpha_f < 1$ and $M = 20$.

	Computation		Communication
	Total time	Time/iter.	Param./iter.
FedAvg	365 min	1.59 min	$2 * \Sigma$
FedProx	325 min	1.99 min	$2 * \Sigma$
FedAvg-C	607 min	24.28 min	$2 * \Sigma$
FedProx-C	711 min	28.44 min	$2 * \Sigma$
Per-FedAvg	121 min	3.56 min	$2 * \Sigma$
FedRep	471 min	4.09 min	$2 * \alpha_f * \Sigma$
pFedMe	1157 min	10.24 min	$2 * \Sigma$
Ditto	318 min	11.78 min	$2 * \Sigma$
FedAMP	92 min	1.53 min	$2 * \Sigma$
FedPHP	264 min	4.06 min	$2 * \Sigma$
FedFomo	193 min	2.72 min	$(1 + M) * \Sigma$
APPLE	132 min	2.93 min	$(1 + M) * \Sigma$
PartialFed	693 min	2.13 min	$2 * \Sigma$
FedALA	7+116 min	1.93 min	$2 * \Sigma$

Applicability of ALA

We directly apply ALA ($s = 80$ and $p = 1$) to the SOTA traditional FL and pFL methods without modifying other learning processes. The accuracy improvement for most of them is remarkable.

	Datasets	Tiny-ImageNet		Cifar100	
	Methods	Acc.	Imps.	Acc.	Imps.
Traditional FL	FedAvg**+ALA**	40.54±0.17	**21.08**	55.92±0.15	**24.03**
	FedProx**+ALA**	40.53±0.26	**21.16**	56.18±0.65	**24.19**
Personalized FL	Per-FedAvg**+ALA**	30.90±0.28	**5.83**	48.68±0.36	**4.40**
	FedRep**+ALA**	37.89±0.31	**0.62**	53.02±0.11	**0.63**
	pFedMe**+ALA**	27.30±0.24	**0.37**	47.91±0.21	**0.57**
	Ditto**+ALA**	40.75±0.06	**8.60**	56.33±0.07	**3.46**
	FedAMP**+ALA**	28.18±0.20	**0.19**	48.03±0.23	**0.34**
	FedPHP**+ALA**	40.16±0.24	**4.47**	54.28±0.21	**3.76**
	PartialFed**+ALA**	35.40±0.02	**0.14**	48.99±0.05	**0.18**

Evaluate in Various Scenarios

FedALA outperforms **11 SOTA** traditional FL and pFL methods as well as **2 fine-tuning-based** pFL methods in the **pathological** and **practical heterogeneous settings** with **5** datasets (including **CV** and **NLP** domains) by up to **3.27%**.

Settings	Pathological heterogeneous setting			Practical heterogeneous setting				
Methods	MNIST	Cifar10	Cifar100	Cifar10	Cifar100	TINY	TINY*	AG News
FedAvg	97.93±0.05	55.09±0.83	25.98±0.13	59.16±0.47	31.89±0.47	19.46±0.20	19.45±0.13	79.57±0.17
FedProx	98.01±0.09	55.06±0.75	25.94±0.16	59.21±0.40	31.99±0.41	19.37±0.22	19.27±0.23	79.35±0.23
FedAvg-C	99.79±0.00	92.13±0.03	66.17±0.03	90.34±0.01	51.80±0.02	30.67±0.08	36.94±0.10	95.89±0.25
FedProx-C	99.80±0.04	92.12±0.03	66.07±0.08	90.33±0.01	51.84±0.07	30.77±0.13	38.78±0.52	96.10±0.22
Per-FedAvg	99.63±0.02	89.63±0.23	56.80±0.26	87.74±0.19	44.28±0.33	25.07±0.07	21.81±0.54	93.27±0.25
FedRep	99.77±0.03	91.93±0.14	67.56±0.31	90.40±0.24	52.39±0.35	37.27±0.20	39.95±0.61	96.28±0.14
pFedMe	99.75±0.02	90.11±0.10	58.20±0.14	88.09±0.32	47.34±0.46	26.93±0.19	33.44±0.33	91.41±0.22
Ditto	99.81±0.00	92.39±0.06	67.23±0.07	90.59±0.01	52.87±0.64	32.15±0.04	35.92±0.43	95.45±0.17
FedAMP	99.76±0.02	90.79±0.16	64.34±0.37	88.70±0.18	47.69±0.49	27.99±0.11	29.11±0.15	94.18±0.09
FedPHP	99.73±0.00	90.01±0.00	63.09±0.04	88.92±0.02	50.52±0.16	35.69±3.26	29.90±0.51	94.38±0.12
FedFomo	99.83±0.00	91.85±0.02	62.49±0.22	88.06±0.02	45.39±0.45	26.33±0.22	26.84±0.11	95.84±0.15
APPLE	99.75±0.01	90.97±0.05	65.80±0.08	89.37±0.11	53.22±0.20	35.04±0.47	39.93±0.52	95.63±0.21
PartialFed	99.86±0.01	89.60±0.13	61.39±0.12	87.38±0.08	48.81±0.20	35.26±0.18	37.50±0.16	85.20±0.16
FedALA	**99.88±0.01**	**92.44±0.02**	**67.83±0.06**	**90.67±0.03**	**55.92±0.03**	**40.54±0.02**	**41.94±0.05**	**96.52±0.08**

Due to limited space, we only show selected results here, please refer to our paper for more results and details (*e.g.*, scalability).

Full paper: https://arxiv.org/abs/2212.01197
Code: https://github.com/TsingZ0/FedALA