FedLMT: Tackling System Heterogeneity of Federated Learning via Low-Rank Model Training with Theoretical Guarantees

Jiahao Liu[1], Yipeng Zhou[2], Di Wu[1], Miao Hu[1], Mohsen Guizani[3], Quan Z. Sheng[2]

[1]Sun Yat-sen University, [2]Macquarie University, [3]Mohamed bin Zayed University of Artificial Intelligence

ICML International Conference On Machine Learning

中山大學 SUN YAT-SEN UNIVERSITY

MACQUARIE University SYDNEY·AUSTRALIA

MOHAMED BIN ZAYED UNIVERSITY OF ARTIFICIAL INTELLIGENCE

Heterogeneous Federated Learning

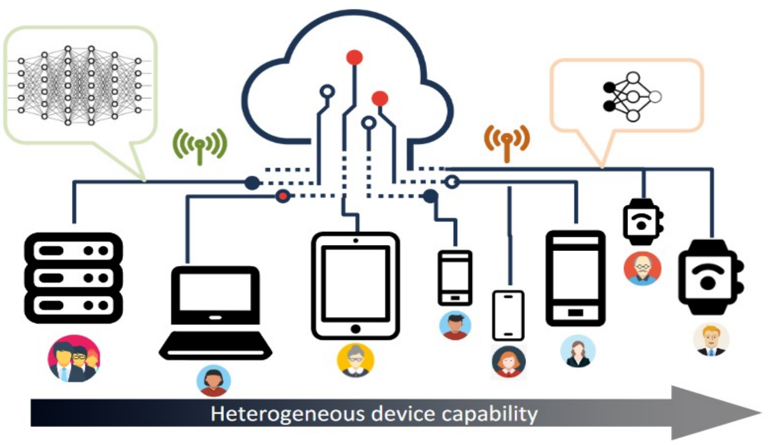
➢ **Problem Statement:**

Suppose there are N clients with non-identically and independently distributed data $D = \{D_1, \cdots, D_N\}$, Federated Learning with system heterogeneity aims to train a global model w by solving the following optimization problem:

$$\min_{w \in R^d} f(w) := \frac{1}{N}\sum_{i=1}^{N} f_i(w_i), w_i = h_i(w)$$

where the local sub-model w_i is trained in client i, and is obtained from w through a map function h_i, e.g., model pruning. The design of h_i depends on the heterogeneous memory capacity β_i of client i, ensuring that each client can load the sub-model for training.

➢ **Example:**



Challenges and Motivations

➢ **Challenges 1:** System heterogeneity limits the participation of clients with constrained memory but rich data, and hence degrades the performance of the global model in federated learning.

➢ **Challenges 2:** The global model trained in current mainstream approaches still suffer performance degradation due to heterogeneous sub-models aggregation, and lack of theoretical performance guarantees.

➢ **Goal:** Devise lightweight algorithms with theoretical guarantees and high flexibility.

Contributions

➢ Demonstrate that the global large model trained by *homogeneous* low-rank sub-models **(FedLMT)** can beat those trained by *heterogeneous* sub-models (current mainstream approaches), with less training costs.

➢ Point out one issue in the convergence analysis of FedHM [1].

➢ Theoretically prove that a converged large model can be reached by training it in the low-rank weight space under non-convex settings.

➢ Propose **pFedLMT**, allowing clients to obtain personalized local models flexibly according to their own resources.

Notations and Main Assumptions

➢ **L_s-smoothness:** $||\nabla f(x) - \nabla f(y)|| \le L_s||x - y||$

➢ **Bounded noise and gradient:**

$$\mathbb{E}_\xi||\nabla F_i(w;\xi) - \nabla f_i(w)||^2 \le \sigma^2, \mathbb{E}_\xi||\nabla F_i(w;\xi)||^4 \le G^4$$

➢ **κ_w-bounded model weight:** $||W||_F \le \kappa_w$

➢ **The weight matrix of low-rank model is of full-rank.**

Algorithms

Algorithm 4 FedLMT

Input: Local epoch E, total iteration T, learning rate γ, a set of randomly selected clients $\mathcal{N}^0$, the initial low-rank model $\mathbf{x}_i^0 = \mathbf{x}^0 = \{W_{i,t}^1, \cdots, W_{i,t}^\rho, U_{i,t}^{\rho+1}, V_{i,t}^{\rho+1}, \cdots, U_{i,t}^L, V_{i,t}^L\}$ according to $\beta_i, \forall i$.
Output: A global model $\mathbf{x}^t$.
for $t = 1$ **to** T **do**
  for client $i \in \mathcal{N}^{t-1}$ in parallel **do**
    $\mathbf{x}_i^t = \mathbf{x}_i^{t-1} - \gamma\nabla_{\mathbf{x}_i^{t-1}} G_i(\mathbf{x}_i^{t-1}, \xi_i^t)$
  end for
  if t divides E **then**
    Each client i in $\mathcal{N}^{t-1}$ sends $\mathbf{x}_i^t$ to the server
    Server updates $\mathbf{x}^t = \frac{1}{|\mathcal{N}^{t-1}|}\sum_{i=1}^{|\mathcal{N}^{t-1}|}\mathbf{x}_i^t$
    Server randomly samples a new client set $\mathcal{N}^t$
    Server broadcasts $\mathbf{x}^t$ to all chosen clients and replaces the local models
  end if
end for
(Optional) Generate $\mathbf{w}^T$ from $\mathbf{x}^T$.

Algorithm 3 pFedLMT

Input: Local epoch E, total iteration T, learning rate γ, a set of randomly selected clients $\mathcal{N}^0$, the initial low-rank model $\mathbf{x}_i^0 = (\mathbf{p}^0, \mathbf{q}_i^0)$ according to $\beta_i, \forall i$. $\mathbf{p}^0$ are the *common* layers and $\mathbf{p}^0 = \{W_{i,0}^1, \cdots, W_{i,0}^\rho\}$. $\mathbf{q}_i^0$ are the *custom* layers of client i and $\mathbf{q}_i^0 = \{U_{i,0}^{\rho+1}, V_{i,0}^{\rho+1}, \cdots, U_{i,0}^L, V_{i,0}^L\}$.
Output: Personalized models $\{\mathbf{x}_1^t, \cdots, \mathbf{x}_N^t\}$.
for $t = 1$ **to** T **do**
  for client $i \in \mathcal{N}^{t-1}$ in parallel **do**
    $\mathbf{q}_i^t = \mathbf{q}_i^{t-1} - \gamma\nabla_{\mathbf{q}_i^{t-1}} G_i(\mathbf{x}_i^{t-1}, \xi_i^t)$
    $\mathbf{p}^t = \mathbf{p}^{t-1} - \gamma\nabla_{\mathbf{p}^{t-1}} G_i(\mathbf{x}_i^{t-1}, \xi_i^t)$
  end for
  if t divides E **then**
    Each client i in $\mathcal{N}^{t-1}$ sends $\mathbf{p}_i^t$ to the server
    Server updates $\mathbf{p}^t = \frac{1}{|\mathcal{N}^{t-1}|}\sum_{i=1}^{|\mathcal{N}^{t-1}|}\mathbf{p}_i^t$
    Server randomly samples a new client set $\mathcal{N}^t$
    Server broadcasts $\mathbf{p}^t$ to all chosen clients and replaces the *common* layers of clients' local models
  end if
end for

Convergence Analysis

Theorem 1. Under the main assumptions, let q_0 be a constant and $1 < q_0 < 2$, and the learning rate satisfies $\gamma \le \min\{\phi^{\frac{2}{q_0-1}}, \frac{1}{L}, 1\}$, for a full model w, by training its corresponding low-rank model x using Algorithm 1, we have:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}||\nabla f(w^{t-1})||^2 \le \frac{2}{\gamma^{q_0}T}(f(w^0) - f^\star) \quad +\gamma^{2-q_0}(\frac{L_s\sigma^2}{2N} + \frac{3}{2}(L-\rho)G^2L_s(\kappa_u^4 + \kappa_v^4))$$
$$+\gamma^{2-q_0}\frac{(L-\rho)G^4}{N^2}(\kappa_{uv}^2 + \kappa_u^2\kappa_v^2(N-1)^2) + \mathcal{O}(\gamma^{3-q_0}).$$

Here ρ denotes the number of layers that are not factorized, and $f^\star$ is the minimum value under the full model weight space. κ_u, κ_v and κ_{uv} are constants bounding the low rank model weight, respectively.

➢ **Properties**

- **Linear Speedup.** By setting $\gamma = N^{\frac{1}{q_0}-\frac{1}{2}}/\sqrt{T}$, FedLMT can achieve a linear speed-up with respect to the number of participating clients.
- **Communication efficiency.** By setting $\gamma = 1/\sqrt{T}$, the convergence rate of FedLMT to obtain the full model w is $\mathcal{O}(1/\sqrt{T})$, which is the same as that of previous works which trains the full model directly under non-convex settings [2].
- **Effect of ρ.** There is a trade-off between the model convergence and the model compression. As ρ gets larger, the error bound gets smaller while the size of the low-rank model is larger.

Experimental Evaluation

1. Performance Comparison with SOTA Baselines

Table 2. The performance of different methods under 'β_4 - β_3 - β_2' settings. ACC means top-1 test accuracy, COMM means the total communication cost including download and upload among all clients, and FLOPs denotes the total floating operations during FL training.

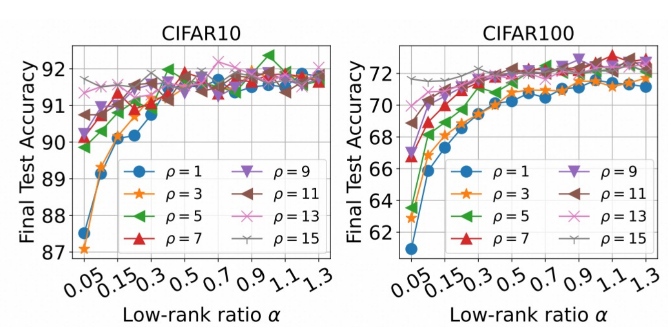
Task		FedAvg	FedDropout	HeteroFL	FedHM	FedRolex	DepthFL	FLANC	FedLMT
CIFAR10	Acc	91.91	73.31	85.02	83.33	89.11	86.79	75.83	**91.03**
	Comm(GB)	223.5	134.7	134.7	122.6	134.7	132.9	132.0	**28.62**
	FLOPs(1E12)	11.18	6.75	6.75	8.77	6.75	11.01	9.22	**2.80**
CIFAR100	Acc	72.20	64.84	63.59	66.10	68.56	69.35	58.32	**71.08**
	Comm(GB)	335.2	201.5	201.5	183.3	201.5	198.6	197.6	**42.93**
	FLOPs(1E12)	16.77	10.10	10.10	13.12	10.10	16.49	13.80	**4.20**
SVHN	Acc	94.39	93.68	92.08	94.26	94.62	92.41	88.05	**95.35**
	Comm(GB)	223.5	134.7	134.7	122.6	134.7	132.9	132.0	**28.62**
	FLOPs(1E12)	11.18	6.75	6.75	8.77	6.75	11.01	9.22	**2.80**
TINY	Acc	42.71	30.38	28.88	36.30	32.82	44.84	31.53	**48.53**
	Comm(GB)	335.2	201.5	201.5	183.3	201.5	198.6	197.6	**42.93**
	FLOPs(1E12)	67.02	40.36	40.36	52.50	40.36	65.94	55.20	**16.74**
WikiText2	Perplexity	3.52	4157.1	3.06	—	3.14	—	—	**2.93**
	Comm(GB)	10.36	7.50	7.50	—	7.50	—	—	**2.65**
	FLOPs(1E12)	0.39	0.275	0.275	—	0.275	—	—	**0.106**

Table 3. Impact of client model heterogeneity distribution on model accuracy using CIFAR10 dataset.

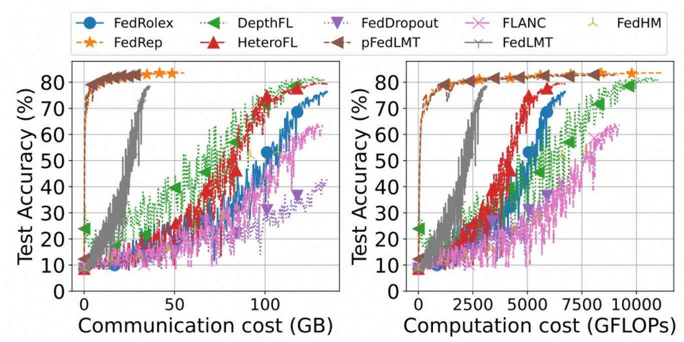
Model Distribution	FedAvg	FedDropout	HeteroFL	FedHM	FedRolex	DepthFL	FLANC	FedLMT (ours)
β_4	91.91	89.79	91.91	91.56	91.90	88.99	90.65	—
β_4 - β_3	—	85.08	88.40	82.11	91.75	88.78	82.41	**91.98**
β_4 - β_3 - β_2	—	73.31	85.02	83.33	89.11	86.79	75.83	**91.03**
β_4 - β_3 - β_2 - β_1	—	61.13	82.05	83.64	84.80	82.77	65.95	**86.27**
β_3	—	80.57	29.94	79.73	86.03	87.79	90.96	**91.98**
β_3 - β_2	—	63.90	32.11	81.87	83.54	84.20	82.35	**91.03**
β_3 - β_2 - β_1	—	47.30	31.95	81.83	72.49	80.54	75.32	**86.27**
β_2	—	55.04	15.92	79.45	61.15	81.49	90.56	**91.03**
β_2 - β_1	—	33.71	19.61	81.92	52.20	77.72	82.88	**86.27**
β_1	—	20.59	12.92	82.38	36.67	73.85	**88.08**	86.27

2. Ablation Study

➢ **Effect of Hyper-parameters**



➢ **Personalization Study**



➢ **FedLMT vs. SOTA Baselines:**

- Obtain better performance with less communication and computation costs.
- The performance is more robust in various system heterogeneous scenarios.
- FedLMT is more flexible and can be easily extended to personalized version (pFedLMT) to settle both data heterogeneity and system heterogeneity.

References:

[1] FedHM: Efficient Federated Learning for Heterogeneous Models via Low-rank Factorization (Arxiv2021)

[2] Parallel restarted SGD with faster convergence and less communication: Demystifying why model averaging works for deep learning. (AAAI2019)