PerFedMask: Personalized Federated Learning with Optimized Masking Vectors

Mehdi Setayesh, Xiaoxiao Li, and Vincent W.S. Wong

Department of Electrical and Computer Engineering

The University of British Columbia, Vancouver, Canada

May 2023

INTRODUCTION

- Federated learning (FL) allows multiple edge devices to train a single model collaboratively under the orchestration of a central server.
- In this work, we study both data and device heterogeneity issues in federated learning using model personalization and masking vectors.

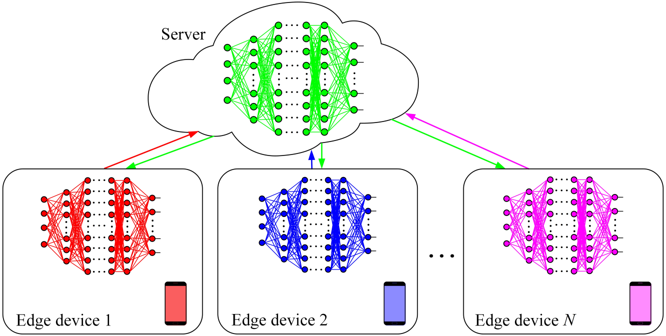

Figure: Illustration of a typical federated learning system.

Federated Learning under Data Heterogeneity

- In practical federated learning systems, the local data samples at the devices are usually non-IID.
- Different personalized federated learning algorithms (e.g. FedBABU) have been proposed to tackle the data heterogeneity issue.

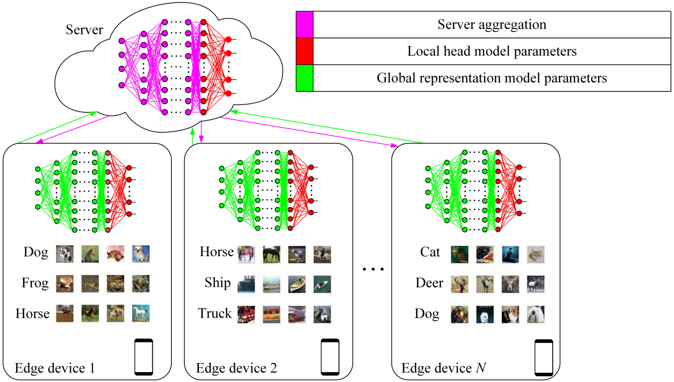

Figure: Illustration of a federated learning system using FedBABU (Oh et al., 2022).

Federated Learning under Device Heterogeneity

- In practical federated learning systems, the devices may have diverse and limited computational and communication capabilities.
- To tackle the device heterogeneity issue, masking vectors can be used to train only a sub-network of the learning model for each device.
- Some works (e.g., HeteroFL, Split-Mix FL) have utilized masking vectors to perform static pruning at initialization (i.e., before training).

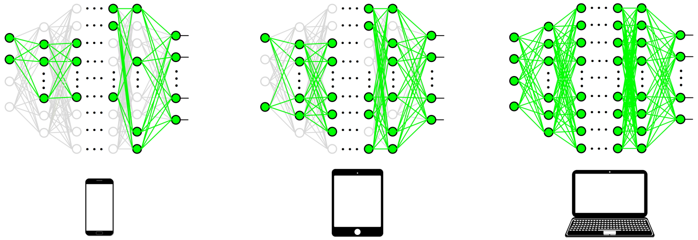

Figure: Using masking vectors to prune the learning model for each device based on its computational capability.

Freezing Method in Federated Learning

- Freezing method is another approach to address the device heterogeneity issue without changing the learning model architecture.
- Unlike pruning, the masked parameters are not removed but are frozen during local updates.

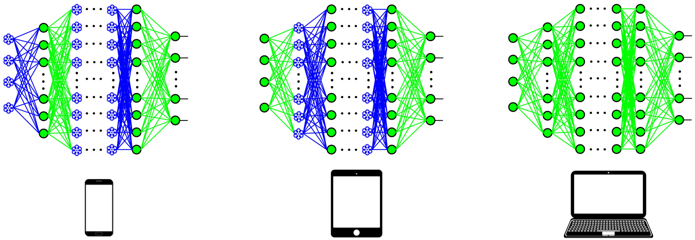

Figure: Using masking vectors to freeze some parts of the learning model for each device based on its computational capability.

Contributions

- We theoretically show that using the masking vectors to freeze the model parameters for the devices may lead to a bias in the convergence bound.
- We propose PerFedMask, which aims to mitigate the performance degradation caused by bias through:
  - Designing the masking vectors via an optimization framework;
  - Fine-tuning the local head models.

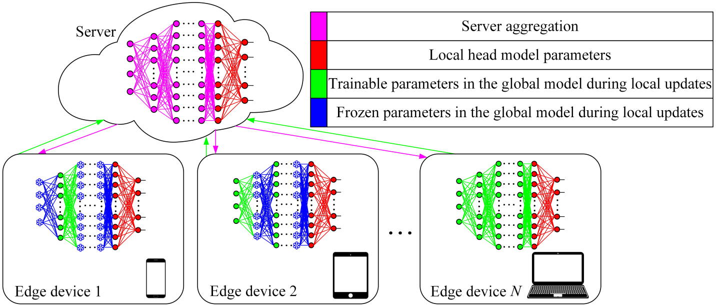

Figure: Illustration of a federated learning system using PerFedMask.

PerFedMask Algorithm

- The learning model $\boldsymbol{\theta}_n$ is decoupled for each device $n \in [N]$ into a global model $\boldsymbol{w}_g$ and a device-specific head model $\boldsymbol{\phi}_n$.
- The server determines the masking vector $\boldsymbol{m}_n$ for each device n before training by solving an optimization problem.
- In each communication round $t \in [T]$,
  - After performing τ local update iterations, each device n sends its final local model to the server.

$$\boldsymbol{w}_n^{\tau+1}(t) = \boldsymbol{w}_g(t) - \eta(t)\boldsymbol{m}_n \odot \sum_{i=1}^{\tau} \nabla f_n(\boldsymbol{w}_n^i(t), b_n^i(t)).$$

  - The server determines the new global model through aggregation of unfrozen parameters.

$$\boldsymbol{w}_g(t+1) = \sum_{n \in [N]} \boldsymbol{k}_n \odot \boldsymbol{w}_n^{\tau+1}(t), \quad \text{where} \quad (\boldsymbol{k}_n)_l = \frac{(\boldsymbol{m}_n)_l}{\sum_{n' \in [N]} (\boldsymbol{m}_{n'})_l}.$$

- After training, a personalized model is obtained for each device by fine-tuning.

Convergence Bound

- When the masking vectors are determined based on the computational capability of the devices, for non-convex and L-smooth loss functions, we have:

- **Theorem.** If the total number of communication rounds T is pre-defined and the learning rate $\eta(t)$ is small enough such that $\eta(t) = \eta \leq \frac{1}{LN^2\tau}$, we have

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla F(\boldsymbol{w}_g(t))\|^2 \leq \frac{2}{\eta\tau T}(F(\boldsymbol{w}_g(1)) - F^*) + LN\tau\eta\sum_{n=1}^{N}\xi_n^2 \\ + L^2\eta^2G^2\frac{(\tau-1)(2\tau-1)}{6} \\ + \boxed{2\Psi\sum_{n=1}^{N}\left(d_{\boldsymbol{w}}\gamma_n - \sum_{l=1}^{d_{\boldsymbol{w}}}(\boldsymbol{k}_n)_l\right)},\end{aligned}$$

Bias due to
device heterogeneity

where Ψ is a constant and $\gamma_n = \max_l\,(\boldsymbol{k}_n)_l$.

Designing Masking Vectors

- Let ψ_n denote the maximum number of parameters that can be trained by device $n \in [N]$.
- We use layer-wise masking to formulate the optimization problem that determines the masking vectors.

$$
\begin{aligned}
\mathcal{P}^{\text{mask}}: \underset{\tilde{\boldsymbol{m}}_n,\, \epsilon_n,\, n\in[N]}{\text{minimize}} \quad & \sum_{n=1}^{N} \left(d_{\boldsymbol{w}} \max_{j\in\Lambda} (\tilde{\boldsymbol{k}}_n)_j - \sum_{j'\in\Lambda} \overbrace{|\pi_{j'}|}^{\text{Number of parameters in layer } j'} (\tilde{\boldsymbol{k}}_n)_{j'} + \epsilon_n \right) \\
\text{subject to} \quad & (\tilde{\boldsymbol{k}}_n)_j = \frac{(\tilde{\boldsymbol{m}}_n)_j}{\sum_{n'=1}^{N} (\tilde{\boldsymbol{m}}_{n'})_j}, \ j \in \Lambda, \ n \in [N], \\
& \sum_{j\in\Lambda} |\pi_j| (\tilde{\boldsymbol{m}}_n)_j = \psi_n - \epsilon_n, \ n \in [N], \\
& (\tilde{\boldsymbol{m}}_n)_j \in \{0,1\}, \ j \in \Lambda, \ n \in [N], \\
& \underbrace{\epsilon_n}_{\text{This variable prevents to train more than } \psi_n \text{ parameters for each device } n} \geq 0, \ n \in [N].
\end{aligned}
$$

Benchmark Experiments

- PerFedMask has comparable performance to FedBABU and outperforms the other baselines in terms of test accuracy after fine-tuning.
- Using masking vectors enable PerFedMask, HeteroFL, and Split-Mix FL algorithms to decrease the number of trainable parameters.

Table: Test accuracy after fine-tuning and number of trainable parameters of PerFedMask and the baseline algorithms for CIFAR-10 and CIFAR-100 datasets

				Test accuracy after fine-tuning				
Dataset	c	PerFedMask (Ours)	FedBABU	FedProx	FedNova	HeteroFL	Split-Mix FL	FedAvg
CIFAR-10	1	**88.43**	88.20	84.96	84.26	87.33	85.56	84.99
	0.1	83.60	**84.27**	74.55	71.88	73.34	77.76	71.19
CIFAR-100	1	**72.40**	69.01	64.63	65.24	68.65	65.95	65.27
	0.1	**67.47**	66.32	59.36	60.42	65.87	62.35	59.12
				Number of trainable parameters				
Dataset		PerFedMask (Ours)	FedBABU	FedProx	FedNova	HeteroFL	Split-Mix FL	FedAvg
CIFAR-10		6.138M	11.167M	11.172M	11.172M	5.674M	0.793M	11.172M
CIFAR-100		1.803M	3.207M	3.309M	3.309M	1.774M	0.223M	3.309M

Benchmark Experiments cont.

- PerFedMask can easily be combined with Split-Mix FL or HeteroFL to further reduce the number of backward FLOPs and the number of trainable parameters.
- Although PerFedMask has reduced the number of trainable parameters and backward FLOPs, it can achieve higher test accuracy than FedBABU.

Table: Performance comparison on CIFAR-10 dataset when $c = 1$.

Algorithm	Test accuracy		# of trainable parameters	# of FLOPs	
	Before fine-tuning	After fine-tuning		Forward	Backward
PerFedMask + Split-Mix FL	51.88	87.74	0.691M	0.178G	0.514G
PerFedMask + HeteroFL	69.44	87.79	5.473M	1.111G	1.721G
PerFedMask	70.14	88.43	6.138M	2.182G	2.697G
Split-Mix FL	57.96	85.56	0.793M	0.178G	0.541G
HeteroFL	62.58	87.33	5.674M	1.111G	1.749G
FedBABU	69.27	88.20	11.167M	2.182G	3.466G

Ablation Studies

- Let ν denote the ratio of devices which can completely update the entire global model during the local update iterations.
- By increasing ν, the test accuracy before fine-tuning is improved.
- PerFedMask can provide a comparable test accuracy after fine-tuning even for $\nu = 0.2$, when compared with the case in which $\nu = 1$.

Table: Results of increasing ν for CIFAR-100 dataset when $c = 1$.

Algorithm	ν	Test accuracy		# of trainable parameters	# of backward FLOPs
		Before fine-tuning	After fine-tuning		
PerFedMask	0.2	29.29	72.07	0.941M	0.617G
	0.4	32.31	74.33	1.518M	0.675G
	0.6	32.79	72.82	2.095M	0.741G
	0.8	33.59	72.64	2.647M	0.803G
	1.0	34.73	73.76	3.207M	0.863G

Conclusion

- We showed that using the masking vectors to address the device heterogeneity issue in federated learning leads to a bias term in the convergence bound.
- We proposed a flexible and easy to implement personalized federated learning algorithm called PerFedMask.
- PerFedMask provides a systematic approach based on minimizing the bias term in the convergence bound to design the masking vectors.
- In PerFedMask, fine-tuning is performed by each device after training to improve the final test accuracy.
- A future direction is to consider freezing priority for different layers in the neural network architecture based on their impact on the final accuracy.