Think Locally, Act Globally: Federated Learning with Local and Global Representations

林佳

Pu Liang*, Terrance Liu*, Liu Ziyin, Ruslan Salakhutdinov, Louis-Philippe Morency
NeurIPS 2019 Workshop on Federated Learning

- Non-IID
  - Non-independent and identically distributed

Outline

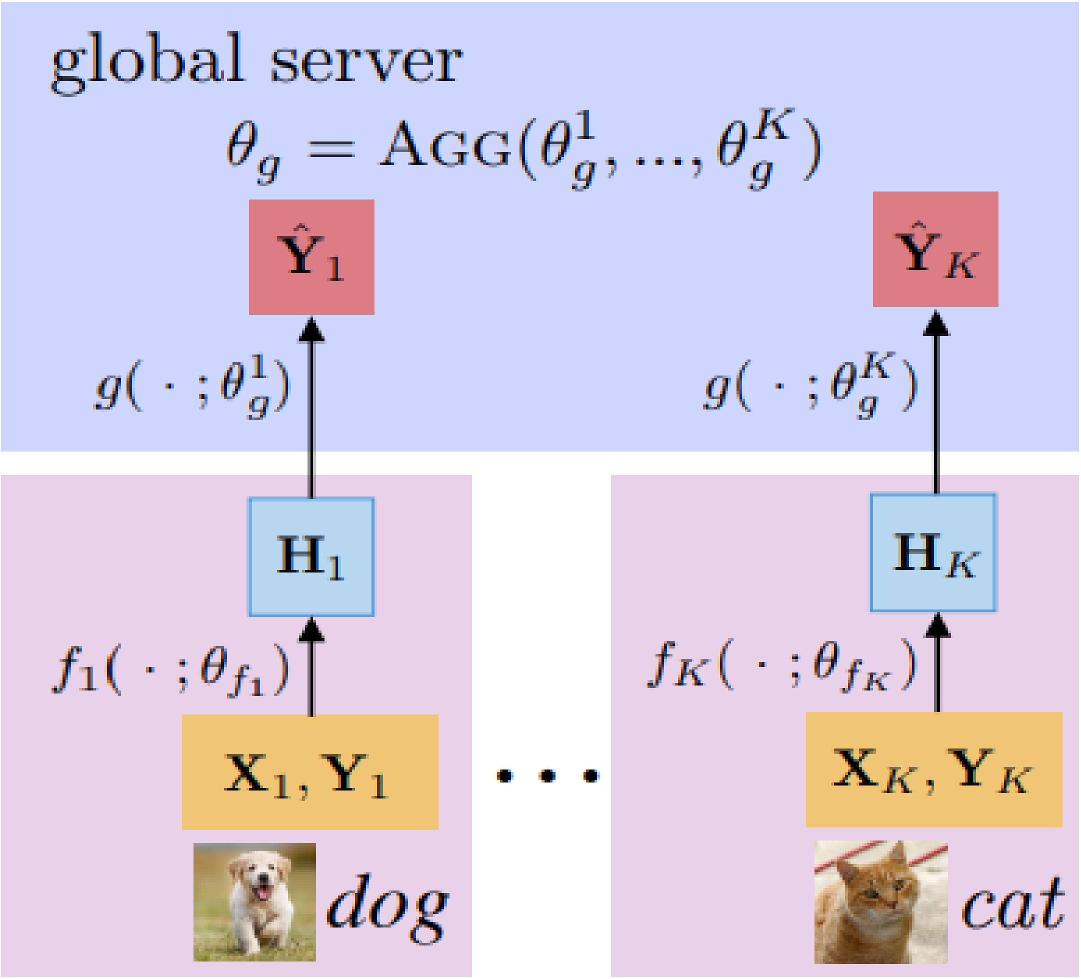
- Introduction
- Local Global Federated Averaging
- Experiments

Introduction

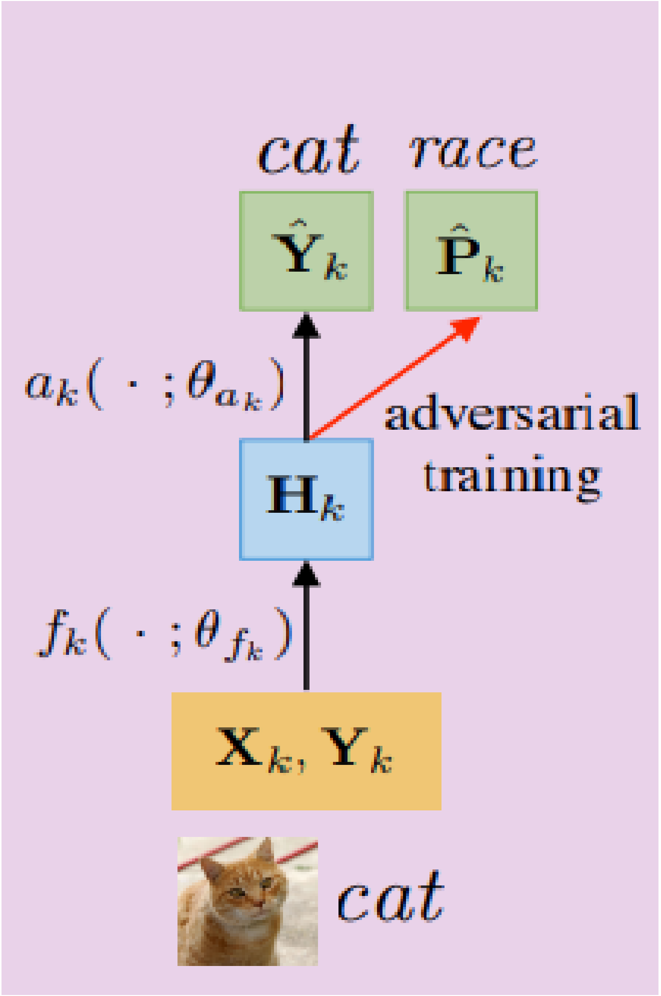
Problems

- Communication Cost
- Heterogeneous Data
- Fair Representation

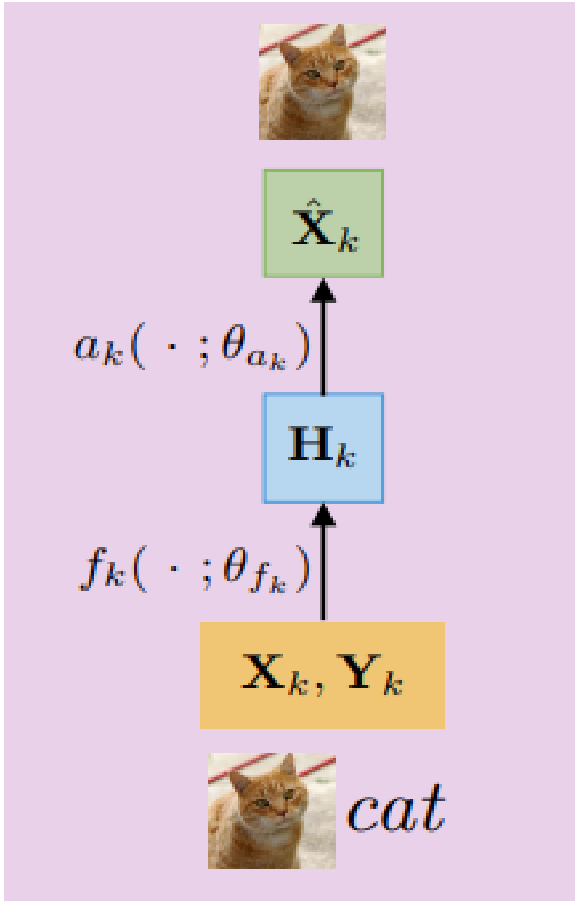
Local Global Federated Averaging



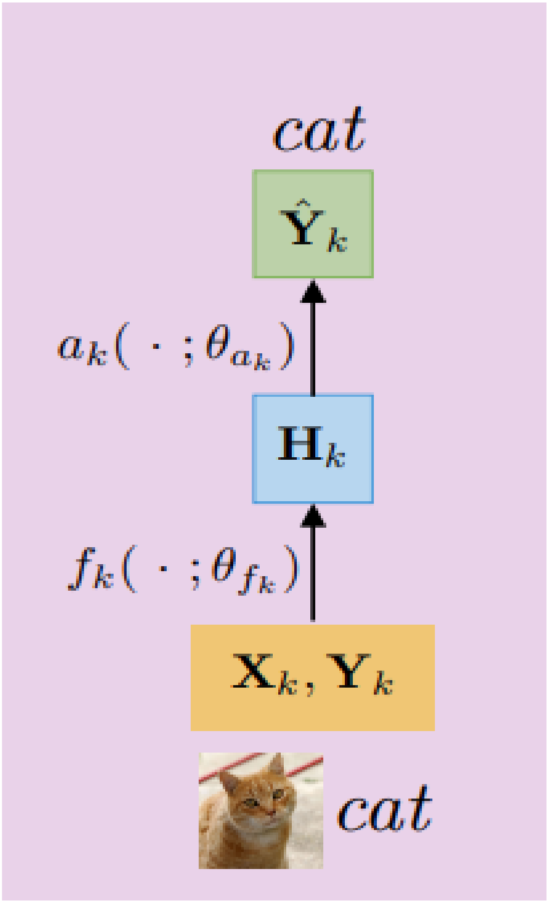
(a) Local Global Federated Averaging



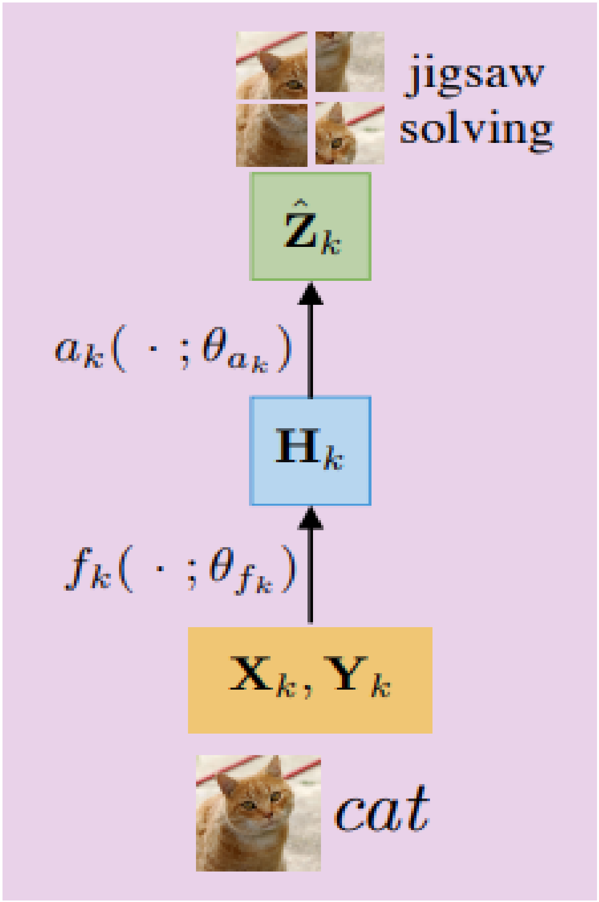
(e) Local Fair Representation Learning



(b) Local Unsupervised Autoencoding



(c) Local Supervised Prediction



(d) Local Self-supervised Prediction

$$\mathcal{L}_{f_k}(\theta_{f_k}, \theta_{a_k}) = \mathbb{E}_{\substack{\mathbf{x} \sim X_k \\ \mathbf{y} \sim Y_k | \mathbf{x}}} \left[-\log \sum_{\mathbf{h}} \left(p_{\theta_{a_k}}(\mathbf{y}|\mathbf{h})\, p_{\theta_{f_k}}(\mathbf{h}|\mathbf{x}) \right) \right]$$

$$\mathcal{L}_{r_k}(\theta_{f_k}, \theta_{r_k}) = \mathbb{E}_{\mathbf{h} \sim f(X_k; \theta_{f_k})} \mathbb{E}_{\mathbf{p} \sim P_k | \mathbf{h}} \left[-\log p_{\theta_{r_k}}(\mathbf{p}|\mathbf{h}) \right]$$

$$\mathcal{L}_g^k(\theta_{f_k}, \theta_g^k) = \mathbb{E}_{\substack{\mathbf{x} \sim X_k \\ \mathbf{y} \sim Y_k | \mathbf{x}}} \left[-\log \sum_{\mathbf{h}} \left(p_{\theta_g^k}(\mathbf{y}|\mathbf{h})\, p_{\theta_{f_k}}(\mathbf{h}|\mathbf{x}) \right) \right]$$

$$p(f_k(\mathbf{x};\theta_{f_k}) = \mathbf{h}|\mathbf{p}) = p(f_k(\mathbf{x};\theta_{f_k}) = \mathbf{h}|\mathbf{p}')$$

$$r_k = p_{\theta_{r_k}}(\mathbf{p}|f(\mathbf{x};\theta_{f_k}) = \mathbf{h})$$

If $p(f_k(\mathbf{x};\theta_{f_k}) = \mathbf{h}|\mathbf{p})$ varies with p, then the corresponding correlation can be captured by adversary r_k

If $p(f_k(\mathbf{x};\theta_{f_k}) = \mathbf{h}|\mathbf{p})$ is indeed invariant with respect to p, then adversary r_k should perform as poorly as random choice

Experiments

Local Test: we know which device the data belongs to (i.e. predictions on existing devices) and choose that particular trained local model.
New Test: we do not know which device the data belongs to (i.e. predictions on new devices)

Table 1: Comparison of federated learning methods on MNIST (top 3 rows) and CIFAR-10 (bottom 3 rows) with non-iid splits. We report accuracy under settings local test and new test as well as the total number of parameters communicated during training. Best results in **bold**. LG-FedAvg outperforms FedAvg under local test and achieves similar performance under new test while using around 50% of the total communicated parameters. Mean and standard deviation are computed over 10 runs.

Method	Local Test Acc. (↑)	New Test Acc. (↑)	# FedAvg Rounds	# LG Rounds	# Params Communicated (↓)
FedAvg	98.15 ± 0.05	$\mathbf{98.15 \pm 0.05}$	725 ± 23.43	0	$5.05 \times 10^{10} \pm 0.16 \times 10^{10}$
Local only	97.17 ± 0.15	84.01 ± 7.42	0	0	0
LG-FedAvg	$\mathbf{98.66 \pm 0.06}$	97.81 ± 0.12	400 ± 14.11	50	$\mathbf{2.80 \times 10^{10} \pm 0.12 \times 10^{10}}$
FedAvg	59.94 ± 1.48	$\mathbf{59.94 \pm 1.48}$	1850 ± 157.10	0	$13.04 \times 10^{9} \pm 1.11 \times 10^{9}$
Local only	86.81 ± 1.20	54.83 ± 0.91	0	0	0
LG-FedAvg	$\mathbf{89.66 \pm 0.53}$	59.63 ± 1.41	1200 ± 244.50	60	$\mathbf{8.48 \times 10^{9} \pm 1.75 \times 10^{9}}$

3, 000 training and 500 test examples drawn independently from the MNIST dataset but rotated 90 degrees

Table 2: What happens when FEDAVG trained on 100 devices of normal MNIST sees a device with rotated MNIST? Catastrophic forgetting, unless one fine-tunes again on training devices and incur high communication cost. LG-FEDAVG relieves catastrophic forgetting by using local models to perform well on both online rotated and regular MNIST, with (C = 0.1) and without (C = 0.0) fine-tuning. Mean and standard deviation are computed over 10 runs.

Method	C	i.i.d. device data		non-i.i.d. device data	
		Normal (↑)	Rotated (↑)	Normal (↑)	Rotated (↑)
FEDAVG	0.0	$\boxed{32.01 \pm 6.24}$	91.83 ± 3.02	$\boxed{35.70 \pm 4.30}$	93.58 ± 0.29
LG-FEDAVG	0.0	$\mathbf{96.55 \pm 0.94}$	$\mathbf{92.92 \pm 2.73}$	$\mathbf{96.31 \pm 0.28}$	$\mathbf{94.12 \pm 0.70}$
FEDAVG	0.1	97.35 ± 0.34	89.29 ± 0.79	96.89 ± 0.54	89.62 ± 0.55
FEDPROX	0.1	94.82 ± 1.14	87.19 ± 0.69	97.86 ± 0.06	91.58 ± 0.19
LG-FEDAVG	0.1	$\mathbf{97.66 \pm 0.75}$	$\mathbf{93.16 \pm 1.24}$	$\mathbf{98.16 \pm 0.67}$	$\mathbf{93.88 \pm 1.36}$

Use the UCI adult dataset where the goal is to predict whether an individual makes more than 50K per year based on their personal attributes, such as age, education, and marital status.

Table 3: Results on enforcing independence with respect to protected attributes *race* and *gender* on income prediction. LG-FEDAVG+Adv uses local models with adversarial (adv) training to remove information about protected attributes, at the expense of a small drop in classifier (class) accuracy of around 4%. Mean and standard deviation are computed over 10 runs.

Method	i.i.d. device data			non-i.i.d. device data		
	Class Acc (↑)	Class AUC (↑)	Adv AUC (↓)	Class Acc (↑)	Class AUC (↑)	Adv AUC (↓)
FEDAVG	83.7 ± 3.1	89.4 ± 1.9	65.5 ± 1.6	83.7 ± 1.8	88.7 ± 1.2	64.1 ± 2.1
LG-FEDAVG−Adv	84.3 ± 2.4	89.0 ± 2.2	63.3 ± 3.7	81.1 ± 1.6	84.4 ± 2.4	62.7 ± 2.5
LG-FEDAVG+Adv	82.1 ± 1.0	85.7 ± 1.7	**50.1 ± 1.3**	80.1 ± 2.0	84.1 ± 2.3	**49.8 ± 2.2**

Algorithm 1 LG-FEDAVG: Local Global Federated Averaging. The K clients are indexed by k; B is the local minibatch size, E is the number of local epochs, and η is the learning rate.

Server executes:

initialize global model with weights θ_g

initialize K local models with weights θ_{f_k} and auxiliary model weights θ_{a_k}

for each round $t = 1, 2, \ldots$ **do**

  $m \leftarrow \max(C \cdot K, 1)$

  $S_t \leftarrow$ (random set of m clients)

  for each client $k \in S_t$ **in parallel do**

    $\theta^k_{g(t+1)} \leftarrow \text{ClientUpdate}(k, \theta_{g(t)})$

  end for

  $\theta_{g(t+1)} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} \theta^k_{g(t+1)}$ // aggregate updates

end for

ClientUpdate (k, θ^k_g)**:** // run on client k

$\mathcal{B} \leftarrow$ (split local data $(\mathbf{X}_k, \mathbf{Y}_k)$ into batches of size B)

for each local epoch i from 1 to E **do**

  for batch $(\mathbf{X}, \mathbf{Y}) \in \mathcal{B}$ **do**

    $\mathbf{H} = f_k(\mathbf{X}; \theta_{f_k})$, $\hat{\mathbf{Z}} = a_k(\mathbf{X}; \theta_{a_k})$, $\hat{\mathbf{Y}} = g(\mathbf{H}; \theta^k_{g(t)})$ // inference steps

    $\theta_{f_k} \leftarrow \theta_{f_k} - \eta \nabla_{\theta_{f_k}} \mathcal{L}_{f_k}(\theta_{f_k}, \theta_{a_k})$ // update local model

    $\theta_{a_k} \leftarrow \theta_{a_k} - \eta \nabla_{\theta_{a_k}} \mathcal{L}_{a_k}(\theta_{f_k}, \theta_{a_k})$ // update auxiliary local model

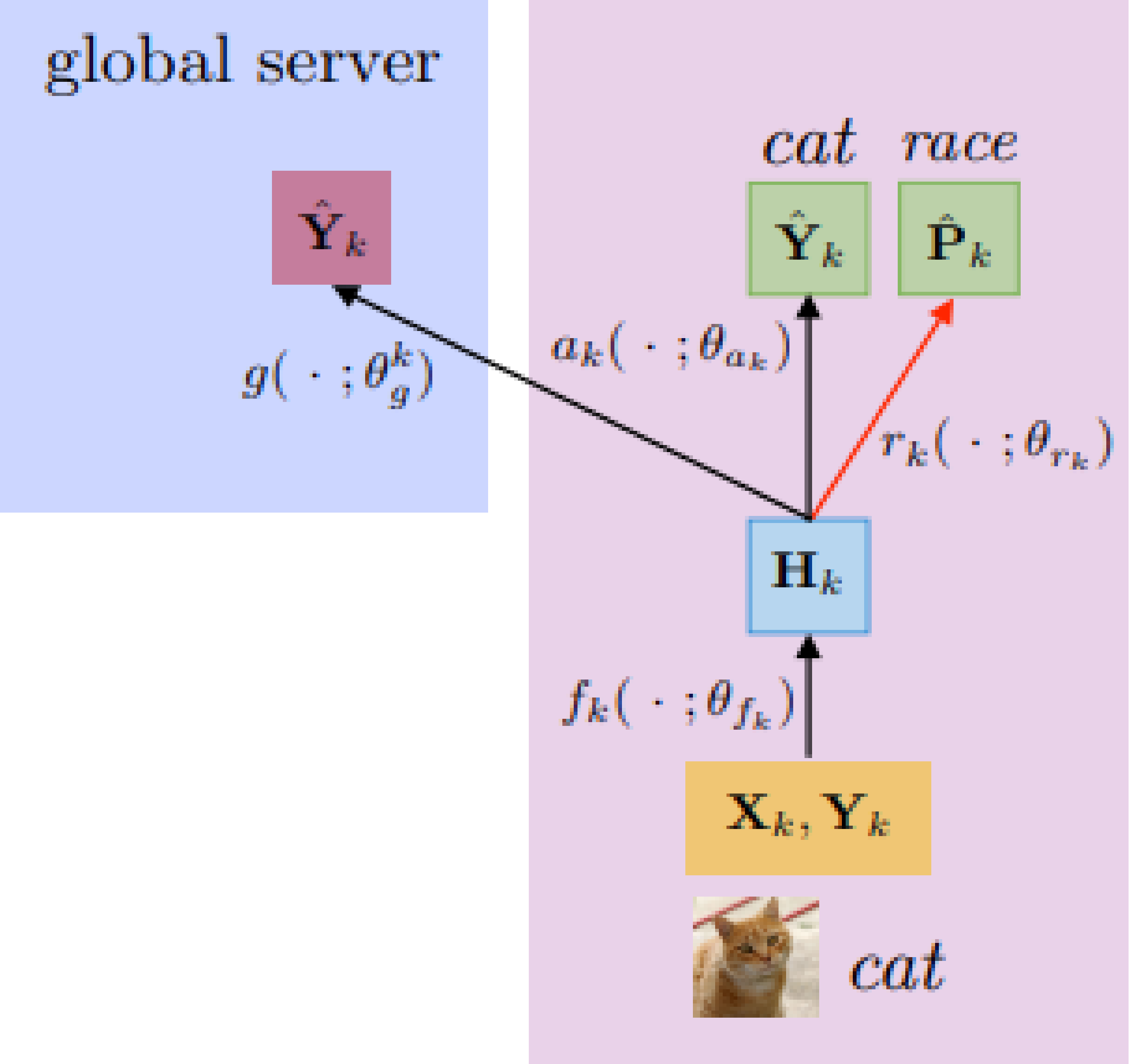
    $\theta_{f_k} \leftarrow \theta_{f_k} - \eta \nabla_{\theta_{f_k}} \mathcal{L}^k_g(\theta_{f_k}, \theta^k_g)$ // update local model

    $\theta^k_g \leftarrow \theta^k_g - \eta \nabla_{\theta^k_g} \mathcal{L}^k_g(\theta_{f_k}, \theta^k_g)$ // update (local copy of) global model

  end for

end for

return global parameters θ^k_g to server


global server
$\hat{\mathbf{Y}}_k$
$g(\,\cdot\,;\theta_g^k)$
cat
race
$\hat{\mathbf{Y}}_k$
$\hat{\mathbf{P}}_k$
$a_k(\,\cdot\,;\theta_{a_k})$
$r_k(\,\cdot\,;\theta_{r_k})$
$\mathbf{H}_k$
$f_k(\,\cdot\,;\theta_{f_k})$
$\mathbf{X}_k, \mathbf{Y}_k$
cat