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

Paul Pu Liang[♡*], **Terrance Liu**[♡*], **Liu Ziyin**[♠], **Ruslan Salakhutdinov**[♡], **Louis-Philippe Morency**[♡]
[♡]Carnegie Mellon University [♠]University of Tokyo
{pliang,terrancl,rsalakhu,morency}@cs.cmu.edu
zliu@cat.phys.s.u-tokyo.ac.jp

Abstract

Federated learning is an emerging research paradigm to train models on private data distributed over multiple devices. A key challenge involves keeping private all the data on each device and training a global model only by communicating parameters and updates. Overcoming this problem relies on the global model being sufficiently compact so that the parameters can be efficiently sent over communication channels such as wireless internet. Given the recent trend towards building deeper and larger neural networks, deploying these neural models in federated settings on real-world tasks is becoming increasingly difficult. To this end, we propose to augment federated learning with *local representation learning* on each device to learn useful and compact features from raw data. As a result, the global model can be smaller since it only operates on higher-level local representations instead of high-dimensional raw data. This reduces the number of global parameters that need to be communicated, thereby reducing the bottleneck in terms of communication cost. Finally, we show that our local models provide flexibility in dealing with online *heterogeneous* data and can be easily modified to learn *fair* representations that obfuscate protected attributes such as race, age, and gender, a feature crucial to preserving the privacy of on-device data.

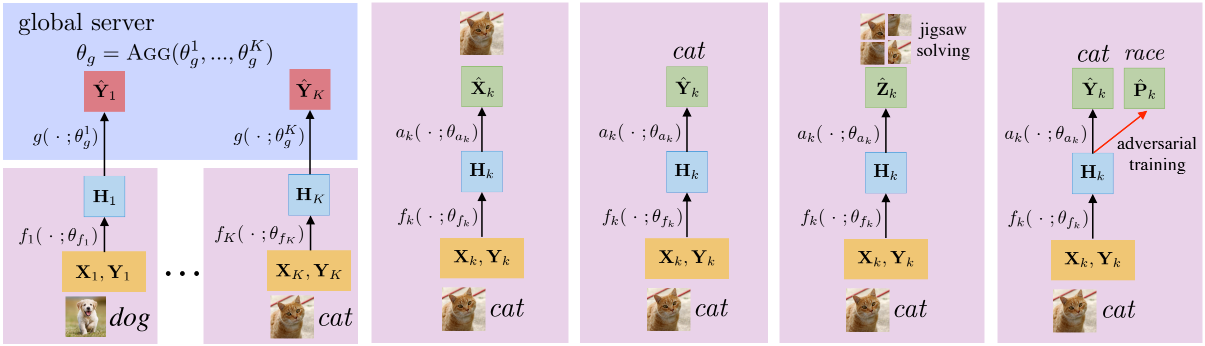
Introduction

Federated learning is an emerging research paradigm to train machine learning models on private data distributed in a potentially non-i.i.d. setting over multiple devices (McMahan et al. 2016). A key challenge in federated learning involves keeping private all the data on each device by training a global model only via communication of parameters and parameter updates to each device. This relies on the global model being sufficiently compact so that the parameters and updates can be sent efficiently over existing communication channels such as wireless networks (Nilsson et al. 2018). However, the recent demands towards building deeper and larger machine learning models (Devlin et al. 2018; He et al. 2015) poses a challenge for deploying federated learning on real-world tasks. This calls for new solutions to the traditional federated averaging frameworks. In this paper, we propose to augment traditional federated learning with *local representation learning* on each device. Each device is augmented with a local model which learns useful and compact representations of raw data. The single global model on the central server is then trained using federated averaging over the local representations from these devices. We call the resulting method Local Global Federated Averaging (LG-FEDAVG) and show that local representation learning is beneficial for the following reasons:

1) *Efficiency:* having local models extract useful, lower-dimensional semantic representations means that the global model now requires a fewer number of parameters. Our choice of local representation learning reduces the number of parameters and updates that need to be communicated to and from the global model, thereby reducing the bottleneck in terms of communication cost. Our proposed method also maintains superior or competitive results on a suite of publicly available real-world datasets spanning image recognition (MNIST, CIFAR) and multimodal learning (VQA).

2) *Heterogeneity:* real-world data is often heterogeneous (i.e. coming from different sources). A single mobile phone is likely to contain data across multiple modalities including images, text, videos, and audio files. In addition, a new device could contain sources of data that have never been observed before during training, such as text in another language, images of a different resolution, or audio in a different voice. Local representations allow us to process the data from new devices in different ways depending on their source modalities (Baltrusaitis, Ahuja, and Morency 2017) instead of using a single global model that might not generalize to never seen before modalities and distributions (Mohri, Sivek, and Suresh 2019). We show that our model can better deal with *heterogeneous* data never observed before during training as compared to recently proposed methods (Sahu et al. 2018).

3) *Fairness:* real-world data often contains sensitive attributes. While federated learning imposes a strict constraint that the data on each local device must remain private (McMahan et al. 2016), recent work has shown that it is possible to recover biases and protected attributes from data representations without having access to the data itself (Caliskan, Bryson, and Narayanan 2017; Bolukbasi et al. 2016). In light of this issue, we show that our local representations can be


global server
$\theta_g = \text{AGG}(\theta_g^1, ..., \theta_g^K)$
$\hat{\mathbf{Y}}_1$
$\hat{\mathbf{Y}}_K$
$g(\,\cdot\,;\theta_g^1)$
$g(\,\cdot\,;\theta_g^K)$
$\mathbf{H}_1$
$\mathbf{H}_K$
$f_1(\,\cdot\,;\theta_{f_1})$
$f_K(\,\cdot\,;\theta_{f_K})$
$\mathbf{X}_1, \mathbf{Y}_1$
$\mathbf{X}_K, \mathbf{Y}_K$
dog
cat
$\hat{\mathbf{X}}_k$
$a_k(\,\cdot\,;\theta_{a_k})$
$\mathbf{H}_k$
$f_k(\,\cdot\,;\theta_{f_k})$
$\mathbf{X}_k, \mathbf{Y}_k$
cat
$\hat{\mathbf{Y}}_k$
jigsaw solving
$\hat{\mathbf{Z}}_k$
race
$\hat{\mathbf{P}}_k$
adversarial training

(a) Local Global Federated Averaging (b) Local Unsupervised Autoencoding (c) Local Supervised Prediction (d) Local Self-supervised Prediction (e) Local Fair Representation Learning

Figure 1: (a) Local Global Federated Averaging (LG-FEDAVG) allows for *efficient* global parameter updates (smaller number of global parameters θ_g), *flexibility* in design across local and global models, the ability to handle *heterogeneous* data, and *fair* representation learning. (b) through (d) show various approaches of training local models including unsupervised, supervised, and self-supervised learning (e.g. jigsaw solving (Noroozi and Favaro 2016)). (e) shows adversarial training against protected attributes $\mathbf{P}_k$. Blue represents the global server and purple represents the local devices. $(\mathbf{X}_k, \mathbf{Y}_k)$ represents data on device k, $\mathbf{H}_k$ are learned local representations via local models $f_k(\,\cdot\,;\theta_{f_k}) : \mathbf{x} \rightarrow \mathbf{h}$ and auxiliary models $a_k(\,\cdot\,;\theta_{a_k}) : \mathbf{h} \rightarrow \mathbf{z}$. $g(\,\cdot\,;\theta_g) : \mathbf{h} \rightarrow \mathbf{y}$ is the global model. AGG is an aggregation function over local updates to the global model (e.g. FEDAVG).

modified to learn *fair* representations that obfuscate protected attributes such as race, age, and gender, a feature crucial to preserving the privacy of on-device data.

Related Work

Federated Learning aims to train models in massively distributed networks (McMahan et al. 2016) at a large scale (Bonawitz et al. 2019), over multiple sources of heterogeneous data (Sahu et al. 2018), and over multiple learning objectives (Smith et al. 2017). Recent methods aim to improve the efficiency of federated learning (Caldas et al. 2018a), perform learning in a one-shot setting (Guha, Talwalkar, and Smith 2019), propose realistic benchmarks (Caldas et al. 2018b), and reduce the data mismatch between local and global data distributions (Mohri, Sivek, and Suresh 2019). While several specific algorithms have been proposed for heterogeneous data, LG-FEDAVG is a more *general* framework that can handle heterogeneous data from new devices, reduce communication complexity, and ensure fair representation learning. We additionally compare with these existing baselines and show that LG-FEDAVG outperforms them in heterogeneous settings.

Distributed Learning is a related field with similarities and key differences: while both study the theory and practice involving partitioning of data and aggregation of model updates (Dean et al. 2012; Ben-Nun and Hoefler 2018; Suresh et al. 2017), federated learning is additionally concerned with data that is private and distributed in a *non-i.i.d.* fashion. Recent work has improved the communication efficiency of distributed learning by sparsifying the data and model (Wang et al. 2017), developing efficient gradient-based methods (Wang and Joshi 2018; Diakonikolas et al. 2017), and compressing the updates (Tsuzuku, Imachi, and Akiba 2018). We emphasize that these general purpose data and gradient compression techniques are *complementary* to our approach: our local and global models can be further sparsified, quantized or compressed.

Representation Learning involves learning informative features from data that are useful for generative and discriminative tasks. A recent focus has been on learning *fair* representations (Zemel et al. 2013), including using adversarial training (Goodfellow et al. 2014) to learn representations that are not informative of predefined private attributes (Feng et al. 2019; Celis and Keswani 2019) such as demographics (Elazar and Goldberg 2018) and gender (Wang et al. 2018). A related line of research is differential privacy which constraints statistical databases to limit the privacy impact on individuals whose information is in the database (Dwork 2006; Dwork and Roth 2014).

Our approach extends recent independent advances in adapting federated learning for heterogeneous data and fairness. LG-FEDAVG is a more *general* framework that can handle heterogeneous data from new devices, reduce communication complexity, and ensure fair representation learning at the same time.

Local Global Federated Averaging

Traditional federated learning algorithms maintain a global model on a server that makes predictions from high-dimensional raw data. A copy of the global model is transmitted to all devices which compute gradient updates using their own subset of the data. Gradient updates from each device are then transmitted back to the global server where all updates are aggregated to finally update the global model. The use of a single global model allows learning of a general model that can make predictions on all subsets of data across all devices. However, a global model operating on high-dimensional raw data requires many parameters that must be transmitted multiple times across devices and training epochs (McMahan et al. 2016).

At a high level, LG-FEDAVG combines local representation learning with global model learning in an *end-to-end* *manner*. Each device learns to extract higher-level representations from raw data before a global model operates on the representations (rather than raw data) from all devices. An overview of LG-FEDAVG is shown in Figure 1(a). The local and global learning procedures are designed to be complementary: local representation learning aims to extract *high level, compact features* important for predicting the label of the data from each device, thereby allowing the global model to save parameters by operating only on *lower dimensional representations*. At the same time, the global model objective ensures that the global model must be able classify data from *all* devices, thereby ensuring that the local representations are general enough instead overfitting to the subset of data on each device.

We begin by defining notation before describing how local and global representation learning is performed. We then explain how the local models can be adapted to learn fair representations before demonstrating how to perform test-time inference over trained local and global models.

Notation: We use uppercase letters X to denote random variables and lowercase letters x to denote their values. Upper case boldface letters $\mathbf{X}$ denote datasets consisting of multiple vector data points $\mathbf{x}$ which we represent by lowercase boldface letters. In the standard federated learning setting, we assume that we have data $\mathbf{X}_k \in \mathbb{R}^{n_k \times d}, k \in [K]$ and their corresponding labels $\mathbf{Y}_k \in \mathbb{R}^{n_k \times c}, k \in [K]$ across K devices. n_k denotes the number of data points on device k and $n = \sum_k n_k$ is the total number of data points, d represents the input dimension and c represents the number of classes for classification and $c = 1$ for regression. Intuitively, each source of data captures a different view $p(X_k, Y_k)$ of the global data distribution $p(X, Y)$. In our experiments, we consider settings where the individual data points in $\mathbf{X}_k, \mathbf{Y}_k$ are sampled i.i.d. with respect to $p(X, Y)$ as well as settings in which sampling is non i.i.d. (e.g. biased sampling with respect to the marginal $p(Y)$ implies that data is distributed unevenly with respect their labels: one device may have, in expectation, a lot more cat images, and another a lot more dog images). During training, we use parenthesized subscripts (e.g. $\theta_{(t)}$) to represent the training iteration t.

Local Representation Learning

For each source of data $(\mathbf{X}_k, \mathbf{Y}_k)$, we learn a high-level, compact representation $\mathbf{H}_k$. This general framework gives the user flexibility in learning $\mathbf{H}_k$, but in general the local representation should have the following properties: 1) be low-dimensional as compared to raw data $\mathbf{X}_k$, 2) capture important features related to $\mathbf{X}_k$ and $\mathbf{Y}_k$ that are useful towards the global model, and 3) not overfit to on-device data which may not align to the global data distribution.

To be more concrete, define some important features $\mathbf{z} \in \mathcal{Z}$ that should be captured using a good representation $\mathbf{h}$. Some choices of $\mathbf{z}$ can be 1) the data itself $\mathbf{x}$ (unsupervised autoencoder learning), 2) the labels $\mathbf{y}$ (supervised learning), or 3) some manually defined labels $\mathbf{z}$ (self-supervised learning). In Figure 1(b) through (d) we summarize the local representation learning methods from $\mathbf{X}_k$ to $\mathbf{H}_k$ and $\mathbf{Y}_k$ resulting in a trained local model f_k on each device. Given these features, each device consists of two components: the local model $f_k : \mathbf{x} \rightarrow \mathbf{h}$ with parameters θ_{f_k}, as well as the local auxiliary network $a_k : \mathbf{h} \rightarrow \mathbf{y}$ with parameters θ_{a_k}. These two networks allow us to infer features $\mathbf{H}_k = f_k(\mathbf{X}_k; \theta_{f_k})$ and auxiliary labels $\mathbf{Z}_k = a_k(\mathbf{H}_k; \theta_{a_k})$ from local device data. Given a suitably chosen local loss function ℓ_{f_k} over $\mathcal{Z} \times \mathcal{Z}$, the local model f_k can now be learned using (stochastic) gradient descent. The local training objective optimizes parameters θ_{f_k} and θ_{a_k} with respect to the local loss (for simplicity we choose supervised learning hence $Z = Y$).

$$\mathcal{L}_{f_k}(\theta_{f_k}, \theta_{a_k}) = \mathbb{E}_{\substack{\mathbf{x}\sim X_k \\ \mathbf{y}\sim Y_k|\mathbf{x}}} \left[-\log \sum_{\mathbf{h}} \left(p_{\theta_{a_k}}(\mathbf{y}|\mathbf{h})\; p_{\theta_{f_k}}(\mathbf{h}|\mathbf{x}) \right) \right]. \tag{1}$$

In practice, we do not have to compute the summation over $\mathbf{h}$ since we perform end-to-end training in a multitask fashion: $\mathbf{h}$ is simply a shared intermediate representation that will be trained to work well for local tasks $\mathbf{z}$ as well as the global model objective as we will discuss next.

Global Aggregation

The *non-i.i.d.* requirements of federated learning implies that simply learning the best possible local model $p(Y_k|X_k)$ is still insufficient for learning a good prediction model over the true joint distribution $p(X, Y)$. Therefore, it is important to learn a global model over the data from all devices $\{(\mathbf{X}_1, \mathbf{Y}_1), ..., (\mathbf{X}_K, \mathbf{Y}_K)\}$. To this end, we define a global model g with parameters θ_g which will be updated using data from all devices. The key difference now is that the global model $g : \mathbf{h} \rightarrow \mathbf{y}$ now operates on the learned local representations $\mathbf{H}_k$ which are already representative of the features required for prediction. Therefore, g can be a much smaller model which we will empirically show in our experiments (§). Contrast this with traditional federated learning where the global model g takes as input raw device data $\mathbf{X}_k$ and makes a prediction $\mathbf{Y}_k$. A model g operating on raw data will usually require multiple layers of representation learning to achieve good performance as shown from the recent trend of using large models for language understanding (Devlin et al. 2018) and visual recognition (Huang, Liu, and Weinberger 2016). This leads to significant communication costs when transmitting global parameters θ_g to local devices.

In our approach, at each iteration t of global model training, the server sends a copy of the global model parameters $\theta_{g(t)}$ to each device which we now label as $\theta^k_{g(t)}$ to represent the asynchronous updates made to each local copy. Each device runs their local model $\mathbf{H}_k = f_k(\mathbf{X}_k; \theta_{f_k})$ and the global model $\hat{\mathbf{Y}}_k = g(\mathbf{H}_k; \theta^k_{g(t)})$ to obtain predicted labels. Given a suitable loss function ℓ_g on the label space $\mathcal{Y} \times \mathcal{Y}$, we can compute the overall loss of the global model on device k:

$$\mathcal{L}^k_g(\theta_{f_k}, \theta^k_g) = \mathbb{E}_{\substack{\mathbf{x}\sim X_k \\ \mathbf{y}\sim Y_k|\mathbf{x}}} \left[-\log \sum_{\mathbf{h}} \left(p_{\theta^k_g}(\mathbf{y}|\mathbf{h})\; p_{\theta_{f_k}}(\mathbf{h}|\mathbf{x}) \right) \right]. \tag{2}$$

We do not have to compute $\sum_{\mathbf{h}}$ since this gradient is a function of both the local and global model parameters (θ_{f_k} and θ^k_g respectively) so both can be updated in an end-to-end manner. The overall loss on device k is a weighted combination

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 $\theta_g$; initialize $K$ local models with weights $\theta_{f_k}$ and auxiliary model weights $\theta_{a_k}$
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
        $\theta^k_{g(t+1)} \leftarrow$ 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 $(k, \theta^k_g)$:                                   // 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 with respect to local loss
        $\theta_{a_k} \leftarrow \theta_{a_k} - \eta \nabla_{\theta_{a_k}} \mathcal{L}_{a_k}(\theta_{f_k}, \theta_{a_k})$      // update auxiliary local model with respect to local loss
        $\theta_{f_k} \leftarrow \theta_{f_k} - \eta \nabla_{\theta_{f_k}} \mathcal{L}^k_g(\theta_{f_k}, \theta^k_g)$      // update local model with respect to global loss
        $\theta^k_g \leftarrow \theta^k_g - \eta \nabla_{\theta^k_g} \mathcal{L}^k_g(\theta_{f_k}, \theta^k_g)$      // update (local copy of) global model with respect to global loss
    end for
end for
return global parameters $\theta^k_g$ to server
```

of local loss and global losses:

$$\mathcal{L} = \mathcal{L}_{f_k}(\theta_{f_k}, \theta_{a_k}) + \lambda \mathcal{L}^k_g(\theta_{f_k}, \theta^k_g), \tag{3}$$

where λ is a hyperparameter that balances the updates to local model θ_{f_k} from both local and global objectives. We argue that this synchronizes the training of the local models. While local models can flexibly fit the small amounts of data on their device, the global objective acts as a regularizer to synchronize the local representations learnt from all devices. Each local model cannot overfit to local data because otherwise, the global objective would be much higher.

After the joint local and global updates, each device now returns updated global parameters $\theta^k_{g(t+1)}$ back to the server which aggregates these updates using FEDAVG: a weighted average over the fraction of data points in each device, $\theta_{g(t+1)} = \sum_{k=1}^{K} \frac{n_k}{n} \theta^k_{g(t+1)}$. We also found that weighting the updates by the norm of the global gradient $\left(\text{i.e. } \left\| \nabla_{\theta^k_g} \mathcal{L}^k_g(\theta_{f_k}, \theta^k_g) \right\|_2^2 \right)$ sped up convergence, a technique proposed by Alain et al. (2015).

The overall training procedure for LG-FEDAVG is shown in Algorithm 1. Communication only happens between the global server and local devices when training the global model, which as we will show in our experiments, can be much smaller given good local representations.

In the following subsection, we outline an example of such a modification in the following section where we aim to learn fair and privacy-preserving local representations via an auxiliary adversarial loss in each local model.

Fair Representation Learning

In this section we detail one example of local representation learning with the goal of removing information that might be indicative of protected attributes. In this setting, suppose the data on each device is now data a triple $(\mathbf{X}_k, \mathbf{Y}_k, \mathbf{P}_k)$ drawn non-i.i.d. from a joint distribution $p(X, Y, P)$ (instead of $p(X, Y)$ as we had previously considered) where $\mathbf{p} \in \mathcal{P}$ are some protected attributes in which the model should not pick up on when making a prediction from $\mathbf{x}$ to $\mathbf{y}$. For example, although there exist correlations between race and income (Lassiter 1965) which could help in income prediction (Chen, Johansson, and Sontag 2018), it would be undesirable for our models to rely on these correlations since these would exacerbate racial biases especially when these models are deployed in the real world.

To learn fair local representations, we follow a similar procedure to (Louppe, Kagan, and Cranmer 2017) which uses adversarial training to remove protected attributes (Figure 1 (e)). More formally, we aim to learn a local model f_k such that the distribution of $f_k(\mathbf{x}; \theta_{f_k})$ conditional on $\mathbf{h}$ is invariant with respect to parameters $\mathbf{p}$:

$$p(f_k(\mathbf{x}; \theta_{f_k}) = \mathbf{h} | \mathbf{p}) = p(f_k(\mathbf{x}; \theta_{f_k}) = \mathbf{h} | \mathbf{p}') \tag{4}$$

for all $\mathbf{p}, \mathbf{p}' \in \mathcal{P}$ and outputs $\mathbf{h} \in \mathcal{H}$ of $f_k(\,\cdot\,; \theta_{f_k})$, thereby implying that $f(\mathbf{x}; \theta_{f_k})$ and $\mathbf{p}$ are independent and f is a pivotal quantity with respect to $\mathbf{p}$. Louppe, Kagan, and Cranmer (2017) showed that we can use adversarial networks in order to constrain model f_k to satisfy Equation (4). f_k is pit against an adversarial model $r_k = p_{\theta_{r_k}}(\mathbf{p} | f(\mathbf{x}; \theta_{f_k}) = \mathbf{h})$

with parameters θ_{r_k} and loss $\mathcal{L}_{r_k}(\theta_{f_k}, \theta_{r_k})$. Intuitively, the adversarial network r_k is trained to predict the distribution of $\mathbf{p}$ as much as possible given the local representation $\mathbf{h}$ from f_k. If $p(f_k(\mathbf{x}; \theta_{f_k}) = \mathbf{h}|\mathbf{p})$ varies with $\mathbf{p}$, then the corresponding correlation can be captured by adversary r_k. On the other hand, if $p(f_k(\mathbf{x}; \theta_{f_k}) = \mathbf{h}|\mathbf{p})$ is indeed invariant with respect to $\mathbf{p}$, then adversary r_k should perform as poorly as random choice. Therefore, we train f_k to both minimize the local loss $\mathcal{L}_{f_k}(\theta_{f_k}, \theta_{a_k})$ and to maximize the adversarial loss $\mathcal{L}_{r_k}(\theta_{f_k}, \theta_{r_k})$. In practice, f_k, a_k, and r_k are simultaneously updated by the following value function:

$$E(\theta_{f_k}, \theta_{a_k}, \theta_{r_k}) = \mathcal{L}_{f_k}(\theta_{f_k}, \theta_{a_k}) - \mathcal{L}_{r_k}(\theta_{f_k},\theta_{r_k}). \quad (5)$$

and solving for the minimax solution

$$\hat{\theta}_{f_k}, \hat{\theta}_{a_k}, \hat{\theta}_{r_k} = \arg \min_{\{\theta_{f_k},\theta_{a_k}\}} \max_{\theta_{r_k}} E(\theta_{f_k}, \theta_{a_k}, \theta_{r_k}). \quad (6)$$

$\mathcal{L}_{f_k}$ and $\mathcal{L}_{r_k}$ are computed using the expected value of the log likelihood through inference networks f_k, a_k, and r_k. We optimize Equation (6) by treating it as a coordinate descent problem and alternately solving for $\hat{\theta}_{f_k}, \hat{\theta}_{a_k}, \hat{\theta}_{r_k}$ using gradient-based methods (details in appendix). Proposition 1 shows that this adversarial training procedure learns an optimal local model f_k that is pivotal (invariant) with respect to $\mathbf{p}$ under local device data distribution $p(X_k, Y_k, P_k)$.

Proposition 1 (Optimality of f_k, adapted from Proposition 1 in (Louppe, Kagan, and Cranmer 2017))**.** *Suppose we compute losses $\mathcal{L}_{f_k}$ and $\mathcal{L}_{r_k}$ using the expected log likelihood through the inference networks f_k, a_k, and r_k,*

$$\mathcal{L}_{f_k}(\theta_{f_k}, \theta_{a_k}) = \mathbb{E}_{\substack{\mathbf{x}\sim X_k \\ \mathbf{y}\sim Y_k|\mathbf{x}}} \left[-\log \left(\sum_{\mathbf{h}} p_{\theta_{a_k}}(\mathbf{y}|\mathbf{h}) \; p_{\theta_{f_k}}(\mathbf{h}|\mathbf{x}) \right) \right], \quad (7)$$

$$\mathcal{L}_{r_k}(\theta_{f_k}, \theta_{r_k}) = \mathbb{E}_{\mathbf{h}\sim f(X_k;\theta_{f_k})} \mathbb{E}_{\mathbf{p}\sim P_k|\mathbf{h}} [-\log p_{\theta_{r_k}}(\mathbf{p}|\mathbf{h})]. \quad (8)$$

If there is a minimax solution $(\hat{\theta}_{f_k}, \hat{\theta}_{a_k}, \hat{\theta}_{r_k})$ for Equation (6) such that $E(\hat{\theta}_{f_k}, \hat{\theta}_{a_k}, \hat{\theta}_{r_k}) = H(Y_k|X_k) - H(P_k)$, $f_k(\,\cdot\,; \hat{\theta}_{f_k})$ is an optimal classifier and a pivotal quantity.

The proof is adapted from (Louppe, Kagan, and Cranmer 2017) to account for local data distributions and intermediate representations $\mathbf{h}$. We provide details in the appendix and also explain adversarial training for the global model.

Inference at Test Time

Given a new test sample $\mathbf{x}'$, FedAvg simply passes $\mathbf{x}'$ to the trained global model g^* for inference. However, LG-FedAvg requires inference through both local and global models. How do we know which trained local model f_k^* fits $\mathbf{x}'$ best? We consider two settings: (1) **Local Test** where we assume we know which device the test data belongs to (e.g. training a personalized text completer from phone data). Using that particular local model works best for the best match between train and test data distribution. (2) **New Test** where we relax this assumption such that it is possible to have an entirely new device during testing with new data sources and distributions. To address the new device scenario, we view each local model f_k^* as trained on a different view of the global data distribution. We can then pass $\mathbf{x}'$ through all the trained local models f_k^* and *ensemble* the outputs. Alternatively, we can train on the new device in an online setting: first train a new local model f_{K+1} on device $K + 1$ and then (optionally) fine tune the global model. We now describe these settings and their performance in detail.

Experiments

We designed our experiments to evaluate the novel components of our proposed method for federated learning. We 1) evaluate how local representations can *efficiently* reduce the number of parameters required in the global model while retaining performance, 2) consider data from *heterogeneous* sources where local models help to prevent catastrophic forgetting in the global model, and 3) demonstrate how to learn fair representations that obfuscate private attributes. Anonymized code is included in the supplementary and implementation details are in the appendix.

Model Performance & Communication Efficiency

Image Recognition on MNIST and CIFAR-10: We begin by studying properties of local and global models on MNIST and CIFAR-10. We focus on a highly *non-i.i.d.* setting and follow the experimental design in (McMahan et al. 2016). We partition the training data by sorting the dataset by labels and dividing it into 200 shards of size 300 (MNIST) and size 250 (CIFAR-10). We randomly assign 2 shards to 100 devices so that each device has at most examples of two classes (highly unbalanced). Similarly, we divide the test set into 200 shards of size 50 and assign 2 shards to each device. Each device has matching train and test distributions.

We consider two settings during testing: 1) **Local Test** where we know which device the data belongs to (i.e. new predictions on an existing device) and choose that particular trained local model. For this setting, we split each device's data into train, validation, and test data, similar to (Smith et al. 2017). 2) **New Test** in which we do not know which device the data belongs to (i.e. new predictions on new devices) (McMahan et al. 2016), so we use an ensemble approach by averaging all trained local model logits before choosing the most likely class (Breiman 1996)[1]. For this setting, we evaluate on the CIFAR-10 test set of 10,000 examples. We choose LeNet-5 (Lecun et al. 1998) as our base model which allows us to draw comparisons between LG-FedAvg and FedAvg. We use the same hyperparameters as the baseline, $C = 0.1$, $E = 1$, $B = 50$, and use the two convolutional layers as our global model, which make up only 4.48% $(2872/64102)$ of the total parameters. We train LG-FedAvg with global updates until we reach a goal accuracy (97.5% for MNIST, 57% for CIFAR-10) before training for additional rounds to jointly update local and global models.

The results in Table 1 show that **LG-FedAvg gives strong performance with low communication cost** on both

[1] For ensembling, all local models have to be sent to the global server *only once* after training. We include this overhead when computing the total number of parameters communicated.

Table 1: Comparison of federated learning methods on MNIST (top 3 rows) and CIFAR-10 (bottom 3 rows) with non-iid split over devices. 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.

Dataset	Method	Local Test Acc. (↑)	New Test Acc. (↑)	# FedAvg Rounds	# LG Rounds	# Params Communicated (↓)
MNIST	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	$\mathbf{2.80 \times 10^{10} \pm 0.12 \times 10^{10}}$
CIFAR-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	$\mathbf{8.48 \times 10^{9} \pm 1.75 \times 10^{9}}$

Table 2: Comparison of FedAvg and LG-FedAvg methods on Visual Question Answering on non-i.i.d. device split setting. We report the number of rounds required to reach our goal accuracy of 40%. LG-FedAvg achieves strong performance using fewer communicated parameters.

Dataset	Method	Local Test Acc. (↑)	# FedAvg Rounds	# LG Rounds	# Params Communicated (↓)
VQA	FedAvg	40.02	47	0	13.97×10^{10}
	LG-FedAvg	**40.94**	32	17	$\mathbf{9.99 \times 10^{10}}$

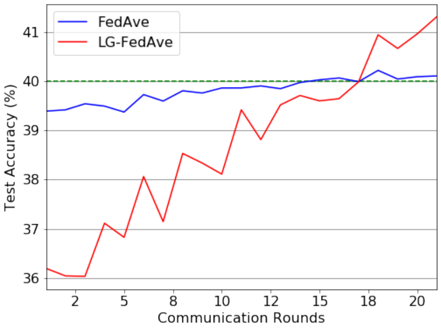


Figure 2: Test accuracy of FedAvg and LG-FedAvg on VQA across 20 rounds (dotted green line marks the goal accuracy of 40% used in Table 2). LG-FedAvg reaches a maximum accuracy of 41.30% compared to that of 40.22% for FedAvg while using only 9.53% of the parameters.

MNIST and CIFAR. For CIFAR local test, LG-FedAvg significantly outperforms FedAvg since local models allow us to better model the local device data distribution. For new test, LG-FedAvg achieves similar performance to FedAvg while using around 50% of the total parameters communicated during updates to the global model. Therefore, LG-FedAvg can learn good local representations for strong global performance under test settings.

Multimodal Learning on Visual Question Answering (VQA): We perform experiments on VQA (Antol et al. 2015), a large-scale multimodal benchmark with 0.25M images, 0.76M questions, and 10M answers. We split the dataset in a non-i.i.d. manner and evaluate the accuracy under the local test setting. We use LSTM (Hochreiter and Schmidhuber 1997) and ResNet-18 (He et al. 2015) unimodal encoders as our local models and a global model which performs early fusion (Srivastava and Salakhutdinov 2012) of text and image features for answer prediction (details in appendix). In Table 2, we observe that LG-FedAvg reaches a goal accuracy of 40% while requiring lower communication costs. In Figure 2, we plot the convergence of test accuracy across communication rounds. LG-FedAvg outperforms FedAvg after 20 rounds while requiring only 9.53% of the number of parameters in FedAvg and continues to improve.

Heterogeneous Data in an Online Setting

We test whether LG-FedAvg can handle heterogeneous data from a new source introduced during testing. We split MNIST across 100 devices in both an i.i.d. and non-i.i.d. setting. We then introduce a new device with 3,000 training and 500 test examples drawn independently from the MNIST dataset but *rotated* 90 *degrees*. This simulates a drastic change in data distribution which may happen in federated learning settings.

We consider 2 methods: 1) FedAvg: train on the original 100 devices using FedAvg, and when a new device comes, update the global model using FedAvg. 2) FedProx (Sahu et al. 2018), a method designed specifically for heterogeneous data by reducing overfitting to local devices. 3) LG-FedAvg: train on the original 100 devices using FedAvg, and when a new device comes, use LG-FedAvg to learn local representations before fine-tuning together with the global model. We hypothesize that good local models can help to "unrotate" the images from the new device to better match the data distribution seen by the global model. In all our experiments, we first train on the original 100 devices until we reach an average goal accuracy of 98% on the devices' test sets. We then train for additional 500 rounds after the new device is introduced by using the new device in addition to a fraction C of the original training devices for fine-tuning: $C = 0.0$ implies no fine-tuning and $C = 0.1$ implies some fine-tuning. Note that $C = 1.0$ implies completely retraining on all data each round, which is impractical.

We report results in Table 3 and draw the following conclusions: 1) **FedAvg suffers from catastrophic forgetting** (Serra et al. 2018) without fine-tuning ($C = 0.0$), in which the global model can perform well on the new device's rotated MNIST (92%) but completely forgets how to classify regular MNIST (32%). Only after fine-tuning ($C = 0.1$) does

Table 3: 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.

Dataset	Method	C	i.i.d. device data		non-i.i.d. device data	
			Normal (↑)	Rotated (↑)	Normal (↑)	Rotated (↑)
MNIST	FedAvg	0.0	32.01 ± 6.24	91.83 ± 3.02	35.70 ± 4.30	93.58 ± 0.29
	LG-FedAvg	0.0	**96.55 ± 0.94**	**92.92 ± 2.73**	**96.31 ± 0.28**	**94.12 ± 0.70**
MNIST	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	**97.66 ± 0.75**	**93.16 ± 1.24**	**98.16 ± 0.67**	**93.88 ± 1.36**

Table 4: Enforcing independence with respect to protected attributes *race* and *gender* on income prediction with the UCI dataset. 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.

Dataset	Method	i.i.d. device data			non-i.i.d. device data		
		Class Acc (↑)	Class AUC (↑)	Adv AUC (↓)	Class Acc (↑)	Class AUC (↑)	Adv AUC (↓)
UCI	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**

the performance on both regular and rotated MNIST improve, but this requires more communication over the 100 training devices. 2) **LG-FedAvg with local models relieves catastrophic forgetting**. Augmenting local models indeed helps to improve online performance on rotated MNIST (93%) while allowing the global model to retain performance on regular MNIST (97%), outperforming both FedAvg and FedProx. We believe LG-FedAvg achieves these results by learning a strong local representation which therefore requires fewer updates from the trained global model.

Learning Fair Representations

The purpose of this experiment is to examine whether local models can be trained adversarially to protect private attributes before local representations pass through the global model. We use the UCI adult dataset (Kohavi 1996) 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. However, we would want our models to be invariant to the sensitive attributes of *race* and *gender* instead of picking up on correlations between $\{$race, gender$\}$ and income that could potentially exacerbate biases. The dataset contains $15,470$ instances each in training and testing which we take the first $15,000$ for easier splitting in a federated setting. We set the number of devices to be 10 and split the dataset in two ways. For the i.i.d. setting we uniformly sample a device for each train and test point, and for the non-i.i.d. setting we choose 100 shards of 150 data points each to obtain imbalanced devices.

We use adversarial learning to remove protected attributes Louppe, Kagan, and Cranmer (2017). Specifically, we aim to learn local representations from which a fully trained adversarial network should *not* be able to predict the protected attributes. We use neural networks for all models with details in the appendix. We report three methods: 1) FedAvg with only a global model and global adversary both updated using FedAvg. The global model is not trained with the adversarial loss since it is simply not possible: once local device data passes through the global model, privacy is potentially violated. 2) LG-FedAvg−Adv which is a local-global model without penalizing the adversarial network, and 3) LG-FedAvg+Adv which jointly trains local, global, and adversary models for a minimax equilibrium.

We report results according to the following metrics: 1) classifier binary accuracy, 2) classifier ROC AUC score, and 3) adversary ROC AUC score. The classifier metrics should be as close to 100% as possible while the adversary should be as close to 50% as possible. From the results in Table 4, we are able to enforce independence using LG-FedAvg+Adv (~ 50% adversary AUC) with a small drop in accuracy (~ 4%) for the global model. In order to ensure that poor adversary AUC was indeed due to fair representations instead of a poorly trained adversary classifier, we also fit a post-fit classifier on local representations to protected attributes and achieve similar close to random results.

Conclusion

This paper proposed LG-FedAvg as a general method that augments FedAvg with *local representation learning* on each device to learn useful features from raw data for prediction. Our approach unifies and extends several recent independent advances in adapting federated learning for heterogeneous data and fair learning. On a suite of publicly available real-world datasets, LG-FedAvg achieves strong performance while reducing communication costs, deals with heterogeneous data in an online setting, and can be easily modified to learn fair representations that obfuscate protected attributes such as race, age, and gender, a feature crucial to preserving the privacy of on-device data. We hope that our work will inspire future research on efficient and privacy-preserving federated learning.

References

[2015] Alain, G.; Lamb, A.; Sankar, C.; Courville, A. C.; and Bengio, Y. 2015. Variance reduction in sgd by distributed importance sampling. *CoRR* abs/1511.06481.

[2015] Antol, S.; Agrawal, A.; Lu, J.; Mitchell, M.; Batra, D.; Zitnick, C. L.; and Parikh, D. 2015. VQA: Visual Question Answering. In *ICCV*.

[2017] Baltrusaitis, T.; Ahuja, C.; and Morency, L. 2017. Multimodal machine learning: A survey and taxonomy. *CoRR* abs/1705.09406.

[2018] Ben-Nun, T., and Hoefler, T. 2018. Demystifying parallel and distributed deep learning: An in-depth concurrency analysis. *CoRR* abs/1802.09941.

[2016] Bolukbasi, T.; Chang, K.-W.; Zou, J. Y.; Saligrama, V.; and Kalai, A. T. 2016. Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. In *NIPS*.

[2019] Bonawitz, K.; Eichner, H.; Grieskamp, W.; Huba, D.; Ingerman, A.; Ivanov, V.; Kiddon, C.; Konecný, J.; Mazzocchi, S.; McMahan, H. B.; Overveldt, T. V.; Petrou, D.; Ramage, D.; and Roselander, J. 2019. Towards federated learning at scale: System design. *CoRR* abs/1902.01046.

[1996] Breiman, L. 1996. Bagging predictors. *Machine learning* 24(2):123–140.

[2018a] Caldas, S.; Konecný, J.; McMahan, H. B.; and Talwalkar, A. 2018a. Expanding the reach of federated learning by reducing client resource requirements. *CoRR* abs/1812.07210.

[2018b] Caldas, S.; Wu, P.; Li, T.; Konecný, J.; McMahan, H. B.; Smith, V.; and Talwalkar, A. 2018b. LEAF: A benchmark for federated settings. *CoRR* abs/1812.01097.

[2017] Caliskan, A.; Bryson, J. J.; and Narayanan, A. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science*.

[2019] Celis, L. E., and Keswani, V. 2019. Improved adversarial learning for fair classification. *CoRR* abs/1901.10443.

[2018] Chen, I.; Johansson, F. D.; and Sontag, D. 2018. Why is my classifier discriminatory? In *NIPS*.

[2012] Dean, J.; Corrado, G.; Monga, R.; Chen, K.; Devin, M.; Mao, M.; aurelio Ranzato, M.; Senior, A.; Tucker, P.; Yang, K.; Le, Q. V.; and Ng, A. Y. 2012. Large scale distributed deep networks. In *NIPS*.

[2018] Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR* abs/1810.04805.

[2017] Diakonikolas, I.; Grigorescu, E.; Li, J.; Natarajan, A.; Onak, K.; and Schmidt, L. 2017. Communication-efficient distributed learning of discrete distributions. In *NIPS*.

[2014] Dwork, C., and Roth, A. 2014. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.* 9(3&#8211;4):211–407.

[2006] Dwork, C. 2006. Differential privacy. In *ICALP*.

[2018] Elazar, Y., and Goldberg, Y. 2018. Adversarial removal of demographic attributes from text data. *CoRR* abs/1808.06640.

[2019] Feng, R.; Yang, Y.; Lyu, Y.; Tan, C.; Sun, Y.; and Wang, C. 2019. Learning fair representations via an adversarial framework. *CoRR* abs/1904.13341.

[2014] Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *NIPS*.

[2019] Guha, N.; Talwalkar, A.; and Smith, V. 2019. One-shot federated learning. *CoRR* abs/1902.11175.

[2015] He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Deep residual learning for image recognition. *CoRR* abs/1512.03385.

[1997] Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

[2016] Huang, G.; Liu, Z.; and Weinberger, K. Q. 2016. Densely connected convolutional networks. *CoRR* abs/1608.06993.

[1996] Kohavi, R. 1996. Scaling up the accuracy of naive-bayes classifiers: a decision-tree hybrid. In *PROCEEDINGS OF THE SECOND INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING*, 202–207. AAAI Press.

[1965] Lassiter, R. L. 1965. The association of income and education for males by region, race, and age. *Southern Economic Journal* 32(1):15–22.

[1998] Lecun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, 2278–2324.

[2017] Louppe, G.; Kagan, M.; and Cranmer, K. 2017. Learning to pivot with adversarial networks. In *NIPS*.

[2016] McMahan, H. B.; Moore, E.; Ramage, D.; and y Arcas, B. A. 2016. Federated learning of deep networks using model averaging. *CoRR* abs/1602.05629.

[2019] Mohri, M.; Sivek, G.; and Suresh, A. T. 2019. Agnostic federated learning. *CoRR* abs/1902.00146.

[2018] Nilsson, A.; Smith, S.; Ulm, G.; Gustavsson, E.; and Jirstrand, M. 2018. A performance evaluation of federated learning algorithms. In *DIDL*.

[2016] Noroozi, M., and Favaro, P. 2016. Unsupervised learning of visual representations by solving jigsaw puzzles. *CoRR* abs/1603.09246.

[2018] Sahu, A. K.; Li, T.; Sanjabi, M.; Zaheer, M.; Talwalkar, A.; and Smith, V. 2018. On the convergence of federated optimization in heterogeneous networks. *CoRR* abs/1812.06127.

[2018] Serra, J.; Suris, D.; Miron, M.; and Karatzoglou, A. 2018. Overcoming catastrophic forgetting with hard attention to the task. In *ICML*.

[2017] Smith, V.; Chiang, C.; Sanjabi, M.; and Talwalkar, A. 2017. Federated multi-task learning. *CoRR* abs/1705.10467.

[2012] Srivastava, N., and Salakhutdinov, R. R. 2012. Multimodal learning with deep boltzmann machines. In *NIPS*.

[2017] Suresh, A. T.; Yu, F. X.; Kumar, S.; and McMahan, H. B. 2017. Distributed mean estimation with limited communication. In *ICML*.

[2018] Tsuzuku, Y.; Imachi, H.; and Akiba, T. 2018. Variance-based gradient compression for efficient distributed deep learning. *CoRR* abs/1802.06058.

[2018] Wang, J., and Joshi, G. 2018. Cooperative sgd: A unified framework for the design and analysis of communication-efficient sgd algorithms. *CoRR* abs/1808.07576.

[2017] Wang, J.; Kolar, M.; Srebro, N.; and Zhang, T. 2017. Efficient distributed learning with sparsity. In *ICML*.

[2018] Wang, T.; Zhao, J.; Yatskar, M.; Chang, K.-W.; and Ordonez, V. 2018. Adversarial removal of gender from deep image representations. *CoRR* abs/1811.08489.

[2013] Zemel, R.; Wu, Y.; Swersky, K.; Pitassi, T.; and Dwork, C. 2013. Learning fair representations. In *ICML*.