Federated Learning for Face Recognition with Gradient Correction

Yifan Niu, Weihong Deng

Beijing University of Posts and Telecommunications
nyf@bupt.edu.cn, whdeng@bupt.edu.cn

Abstract

With increasing appealing to privacy issues in face recognition, federated learning has emerged as one of the most prevalent approaches to study the unconstrained face recognition problem with private decentralized data. However, conventional decentralized federated algorithm sharing whole parameters of networks among clients suffers from privacy leakage in face recognition scene. In this work, we introduce a framework, FedGC, to tackle federated learning for face recognition and guarantees higher privacy. We explore a novel idea of correcting gradients from the perspective of backward propagation and propose a softmax-based regularizer to correct gradients of class embeddings by precisely injecting a cross-client gradient term. Theoretically, we show that FedGC constitutes a valid loss function similar to standard softmax. Extensive experiments have been conducted to validate the superiority of FedGC which can match the performance of conventional centralized methods utilizing full training dataset on several popular benchmark datasets.

Introduction

Face Recognition has been the prominent biometric technique for identity authentication and has been widely applied in many areas. Recently, a variety of data-driven approaches using Deep Convolutional Neural Networks (DCNNs) (Taigman et al. 2014; Kim, Park, and Shin 2020; Deng et al. 2020; Duan, Lu, and Zhou 2019; Marriott, Romdhani, and Chen 2021) have been proposed to improve the face identification and verification accuracy. A large scale dataset with diverse variance is crucial for discriminative face representation learning. Although existing datasets (Cao et al. 2018; Guo et al. 2016; Kemelmacher-Shlizerman et al. 2016; Parkhi, Vedaldi, and Zisserman 2015; Wang et al. 2018a; Yi et al. 2014; Zhu et al. 2021) were created aiming to study the unconstrained face recognition problem, they are still biased compared with the real world data distribution. Considering privacy issue, we are not authorized to get access to mass face data in real world. Thus, it is vital to train a model with private decentralized face data to study the unconstrained face recognition problem in real world scene.

Federated methods on object classification tasks are all under a common setting where a shallow network is adopted as backbone and a shared fully-connected layer is applied for final classification, which is likely to lead to privacy leakage. Therefore, these methods are not applicable to face recognition. Once the private class embedding is obtained, one client's private high-fidelity face images can be easily synthesized by other clients via optimizing random noise, such as DeepInversion (Yin et al. 2020). Moreover, lots of GAN-based face generation technics are also proposed to generate a frontal photorealethistic face image with face embeddings. On the other hand, existing federated methods are mainly focusing on shallow networks(*e.g.* 2 layer fully connected network), we found these methods may easily cause network collapsing when applied to deeper network structure on facial datasets. Thus, we rethink the federated learning problem of face recognition on privacy issues, and remodel conventional Federated Averaging algorithm (FedAvg) (McMahan et al. 2017) via ensuring each client holds a private fully-connected layer which not only guarantees higher privacy but also contributes to network convergence.

In general, each client commonly holds a small-scale non-IID local dataset. When we follow the above setting, onece the k-th client solves the optimization problem locally, the classification task is relatively uncomplicated and the network tends to overfit and suffers from degradation of generalization ability. It leads to a phenomenon that *class embeddings* (the parameters of last fully-connected layer) of the same client are almost orthogonal to each other, but part of class embeddings of different clients are highly similar.

To solve aforementioned problem, we should constitute a new training strategy to train a model with private decentralized non-IID (Non Identically and Independently Distributed) facial data. In this work, we first propose FedGC, a novel and poweful federated learning framework for face recognition, which combines local optimization and cross client optimization injected by our proposed softmax regularizer. FedGC is a privacy-preserving federated learning framework which guarantees that each client holds private class embeddings. In face recognition, several variants of softmax-based objective functions (Deng et al. 2019; Deng, Zhou, and Zafeiriou 2017; Simonyan and Zisserman 2014; Sun, Wang, and Tang 2015; Taigman et al. 2014; Wang et al. 2018b; Wolf, Hassner, and Maoz 2011) have been proposed in centralized methods. Hence, we propose a softmax-based regularizer aiming to correct gradients of local softmax and precisely introduce cross-client gradients to ensure that cross-clients class embeddings are fully spread out and it can can be readily extended to other forms. Additionally, we give a theoretical analysis to show that FedGC constitutes a valid loss function similar to standard softmax. Our contributions can be summarized as follows:

- We propose a federated learning framework, FedGC, for face recognition and guarantees higher privacy. It addresses the missing local optimization problems for face-specific softmax-based loss functions.
- We start from a novel perspective of back propagation to correct gradients and introduce cross-client gradients to ensure the network updates in the direction of standard softmax. We also give a theoretical analysis to show the effectiveness and significance of our method.
- Extensive experiments and ablation studies have been conducted and demonstrate the superiority of the proposed FedGC on several popular benchmark datasets.

Related Work

Face Recognition. Face Recognition (FR) has been the prominent biometric technique for identity authentication and has been widely applied in many areas (Wang and Deng 2018). Rcently, face reogintion has achieved a series of promising breakthrough based on deep face representation learning and perfromed far beyond human. Conventional face-recognition approaches are proposed such as Gabor wavelets (Liu and Wechsler 2002) and LBP (Ahonen, Hadid, and Pietikainen 2006). Schroff (Schroff, Kalenichenko, and Philbin 2015) proposed triplet loss to minimize intra-class variance and maximize inter-class variance. Various of softmax-based loss functions also emerged, such as L-Softmax (Liu et al. 2016), CosFace (Wang et al. 2018c), SphereFace (Liu et al. 2017), AM-Softmax (Wang et al. 2018b), Arcface (Deng et al. 2019). Circle Loss (Sun et al. 2020) proposed a flexible optimization manner via reweighting less-optimized similarity scores. GroupFace (Kim et al. 2020) proposed a novel face-recognition achitecture learning group-aware representations. However, these data-driven approaches aim to learn discriminative face representations on the premise of having the access to full private faical statistics. Public available training databases (Cao et al. 2018; Guo et al. 2016; Kemelmacher-Shlizerman et al. 2016; Parkhi, Vedaldi, and Zisserman 2015; Wang et al. 2018a; Yi et al. 2014) are mostly collected from the photos of celebrities due to privacy issue, it is still biased. Furthermore, with increasing appealing to privacy issues in society, existing public face datasets may turn to illegal.

Federated Learning. Federated Learning (FL) is a machine learning setting where many clients collaboratively train a model under the orchestration of a central server, while keeping the training data decentralized, aims to transfer the traditional deep learning methods to a privacy-preserving way. Existing works seek to improve model performance, efficiency and fairness in training and communication stage. FegAvg (McMahan et al. 2017) was proposed as the basic algorithm of federated learning. FedProx (Li et al. 2018) was proposed as a generalization and re-parametrization of FedAvg with a proximal term. SCAFFOLD (Karimireddy et al. 2019) controls variates to correct the 'client-drift' in local updates. FedAC (Yuan and Ma 2020) is proposed to improve convergence speed and communication efficiency. FedAwS (Yu et al. 2020) investigated a new setting where each client has access to the positive data associated with only a single class. However, these methods are not applicable for face recognition. Most of them are mainly focusing on shallow networks and suffers from privacy leakage in face recognition. Recently, there also emerged some works (Bai et al. 2021; Aggarwal, Zhou, and Jain 2021) focusing on federated face recognition.

Methodology

In this section, we will first provide the formulation of the federated learning and its variant for face recognition. We start by analysing, and then illustrate how we are motivated to propose FedGC.

Problem Formulation

We consider a C class classification problem defined over a compact space $\mathcal{X}$ and a label space $\mathcal{Y}$. Let K be the number of clients, suppose the k-th client holds the data $\{x_i^k, y_i^k\}$ which distributes over $\mathcal{S}_k : \mathcal{X}_k \times \mathcal{Y}_k$, and ensure the identity mutual exclusion of clients $\mathcal{Y}_k \cap \mathcal{Y}_z = \varnothing$, where $k, z \in [K], k \neq z$, such that $\mathcal{S} = \cup_{k\in[K]}\mathcal{S}_k$. In this work, we consider the following distributed optimization model:

$$\min_w F(w) \triangleq \sum_{k=1}^{K} p_k F_k(w), \quad (1)$$

where p_k is the weight of the k-th client. Let the k-th client holds n_k training data and $\sum_{k=1}^{K} n_k = N$, where N is total number of data samples. We define p_k as $\frac{n_k}{N}$, then we have $\sum_{k=1}^{K} p_k = 1$.

Consider an "embedding-based" discriminative model, given an input data $x \in \mathcal{X}$, a neural network $G : \mathcal{X} \to \mathbb{R}^d$ parameterized by θ embeds the data x into a d-dimensional vector $G(x;\theta) \in \mathbb{R}^d$. Finally, the logits of an input data x in the k-th client $f_k(x) \in \mathbb{R}^{C_k}$ can be expressed as:

$$f_k(x) = W_k^T G(x;\theta), \quad (2)$$

where matrix $W_k \in \mathbb{R}^{d\times C_k}$ is the class embeddings of the k-th client. Then Eq. 1 can be reformulated as:

$$\min_{W,\theta} F(W,\theta) \triangleq \sum_{k=1}^{K} p_k \frac{1}{n_k} \sum_{i=1}^{n_k} \ell_k\left(f_k(x_i^k), y_i^k\right), \quad (3)$$

where $\ell_k(\cdot,\cdot)$ is the loss function of the k-th client, $W = [W_1, \cdots, W_K]^T$. To provide a more strict privacy guarantee, we modifed FedAvg (McMahan et al. 2017) via keeping last fully-connected layer private in each client. We term this privacy-preserving version of FedAvg as Federated Averaging with Private Embedding (FedPE). In FedPE, each client only have access to its own final class embeddings and the shared backbone parameters. Note that differential privacy

Algorithm 1: FedGC.

```
Input. The $K$ clients are indexed by $k$ and hold local data distributes over $\mathcal{S}_k$, $\eta$ is learning rate.
Server initializes model parameters $\theta^0$, $W^0$
for each round $t = 0, 1, ..., T-1$ do
    Server initializes $k$-th client model with $\theta^t$, $W_k^t$.
    for each client $k = 1, 2, ..., K$ do
        The $k$-th client computes local Softmax
        $\left(\theta_k^{t+1}, W_k^{t+1}\right) \leftarrow \left(\theta^t, W_k^t\right) - \eta\nabla\ell_k\left(x_i^k, y_i^k\right)$,
        and sends $\left(\theta_k^{t+1}, W_k^{t+1}\right)$ to the server.
    end for
    Server aggregates the model parameters:
    $\theta^{t+1} \leftarrow \sum_{k=1}^{K}\frac{n_k}{n}\theta_k^{t+1}$
    $\tilde{W}^{t+1} = \left[W_k^{t+1}, \ldots, W_K^{t+1}\right]^T$
    Server applys gradient correction:
    $W^{t+1} \leftarrow \tilde{W}^{t+1} - \lambda\eta\nabla_{\tilde{W}^{t+1}} Reg\left(\tilde{W}^{t+1}\right)$
end for
Output. $\theta^T$, $W^T$
```

(Abadi et al. 2016) for federated methods can be readily employed in FedPE by adding noise to the parameters from each client to enhance security.

Observation and Motivation

Softmax Loss. Softmax loss is the most widely used calssification loss function in face recognition. For convenience, we omit the bias b_j. In the k-th client, the optimization objective is to minimize the following cross-entropy loss:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{W_{k,y_i}^T X_{k,i}}}{\sum_{j=1}^{n} e^{W_{k,j}^T X_{k,i}}}, \tag{4}$$

where $X_{k,i} \in \mathbb{R}^d$ denotes the deep feature of the i-th sample, belonging to the y_i-th class. In each individual client, the optimization objective is to minimize inter-class similarity and maximize intra-class similarity over the local class space C_k. We define this optimization in FedPE within client as *local optimization*. However, centralized training on the full training set solves the problem over the global class space C. We define the centralized method as *global optimization*.

In local optimization, the local softmax is to force $W_{k,y_i}^T X_{k,i} > \max_{j\in C_k, j\neq y_i}\left(W_{k,j}^T X_{k,i}\right)$. However, in global optimization, the softmax is to force $W_{y_i}^T X_i > \max_{j\in C, j\neq y_i}\left(W_j^T X_i\right)$. Thus, it is obvious that the model solving the classification problem as Eq. 3 only apply within-client optimization and omit cross-client optimization, lacking constraint $W_{k,y_i}^T X_{k,i} > \max_{j\in C_z, z\neq k}\left(W_{z,j}^T X_{k,i}\right)$.

Therefore, this objective function as Eq. 3 leads the model to a convergence state where class embeddings of the same client are almost orthogonal to each other, but part of class embeddings of different clients may highly similar. And it results in overlapping of feature space among cross-client classes. Furthermore, only applying local optimization is more likely to cause overfitting on small-scale local datasets.

Cross-Client Separability with Gradient Correction

It is hard to mimic the global softmax with a set of local softmax. To address the missing optimization, as illustrated in Fig. 1, a heuristic approach to minimize similarity among cross-client class embeddings is to constrain the cross-client embeddings with a regularization term. Considering the additivity of gradients and the unique properties of sofmax loss gradient, we are motivated to address this issue from a new perspective of back propagation. Following the form of softmax, we define a regularization term, namely *softmax regularizer*, on the class embeddings $W \in \mathbb{R}^{d\times C}$ as:

$$Reg\left(W\right) = \sum_{k=0}^{K}\sum_{i=0}^{C_k} -\log\frac{e^{W'^T_{k,i}W'_{k,i}}}{e^{W'^T_{k,i}W'_{k,i}} + \sum_{z\neq k}\sum_{j=0}^{C_z} e^{W^T_{z,j}W'_{k,i}}}, \tag{5}$$

where $W_{k,i}$ is the i-th class embedding of the k-th client, and $(\cdot)'$ indicates the vector dosen't require gradient (the gradient is set to be zero). We precisely limit the gradient of loss function with softmax regularizer in order to push the network update towards the direction of standard Softmax.

In addition to FedPE, the server performs an additional optimization step on the class embedding matrix $\mathbf{W} \in \mathbb{R}^{d\times C}$ to ensure that cross-client class embeddings are separated from each other. The Federated Averaging with Gradient correction (FedGC) algorithm is summarized in Algorithm 1. In communication round t, Server broadcast model parameters $\left(\theta^t, W_k^t\right)$ to the k-th clients. Then clients locally compute an update with respect to local softmax loss function with their local data asynchronously, and send the new model $\left(\theta^{t+1}, W_k^{t+1}\right)$ back. Finally, Server collects an aggregate of the client updates and applys cross-client optimization. Note that differential privacy can also be applied to FedGC to prevent privacy leakage, like FedPE.

We will theoretically analyze how FedGC works and how it pushes the network to update in the direction of global standard softmax. Note that FedGC effectively seeks to collaboratively minimize the following objective with softmax regularizer $Reg(W)$:

$$F(W,\theta) \triangleq \sum_{k=1}^{K} p_k \frac{1}{n_k}\sum_{i=1}^{n_k}\ell_k\left(f_k(x_i^k), y_i^k\right) + \lambda\cdot Reg\left(W\right). \tag{6}$$

For convenience, we assume that every client holds $n_1 = \cdots = n_C = \frac{N}{K}$ data and $c_1 = \cdots = c_K = \frac{C}{K}$, every class holds $a_1 = \cdots = a_C = \frac{N}{C}$ images, and $\lambda = \frac{1}{C}$, the objective function can be reformulated as:

$$\begin{aligned} F(W,\theta) &= \frac{1}{N}\sum_{k=1}^{K}\sum_{(x_i,y_i)\in\mathcal{S}_k}\ell_{eq}\left(f_k(x_i^k), y_i^k\right) \\ &= -\frac{1}{N}\sum_{k=1}^{K}\sum_{(x_i,y_i)\in\mathcal{S}_k}\left(\log\frac{e^{W_{k,y_i}^T X_{k,i}}}{\sum_{j=1}^{C} e^{W_{k,j}^T X_{k,i}}}\right. \\ &\left. + \log\frac{e^{W'^T_{k,y_i}W'_{k,y_i}}}{e^{W'^T_{k,y_i}W'_{k,y_i}} + \sum_{z\neq k}\sum_{j=0}^{C_z} e^{W^T_{z,j}W'_{k,y_i}}}\right). \end{aligned} \tag{7}$$

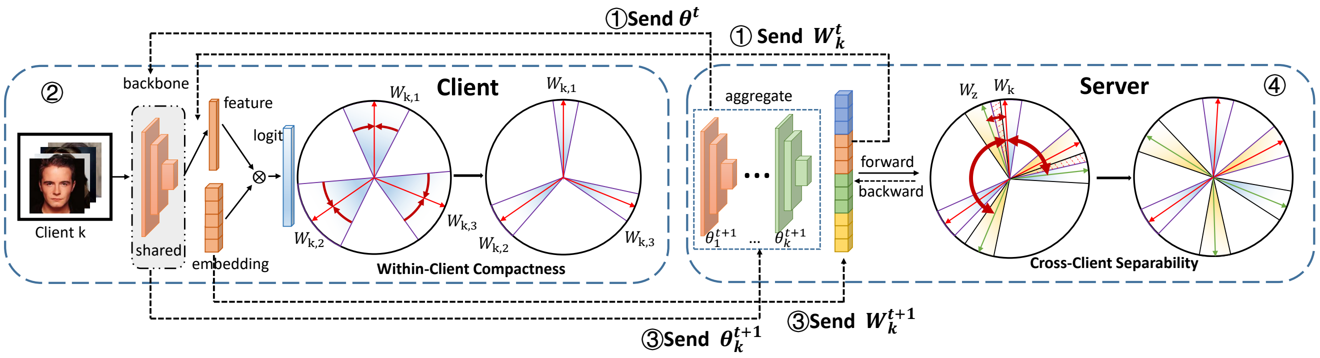


Figure 1: An illustration of our method. In communication round t, Server broadcast model parameters (θ^t, W_k^t) to the selected clients. Then clients locally compute an update to the model with their local data asynchronously, and send the new model $\left(\theta^{t+1}, W_k^{t+1}\right)$ back. Finally, Server collects an aggregate of the client updates and applys cross-client optimization. (a)Client Optimization: clients seek to get more discriminative and more compact features. (b)Server Optimization: correct gradients and make cross-client embeddings spreadout.

Thus, FedGC objective Eq. 6 equals the empirical risk with respect to the loss function $\ell_{eq}\left(f_k(x_i^k), y_i^k\right)$. Our analysis easily extends to unbalanced distribution by involving a weighted form.

Considering the collaborative effect of all the terms in ℓ_{eq}, we give a interpretation from the perspective of backward propagation. For standard softmax in global optimization, the computation of gradients $\frac{\partial L}{\partial W_{y_i}}$ and $\frac{\partial L}{\partial W_j}$ are listed as follows:

$$\frac{\partial L}{\partial W_{y_i}} = \left(\frac{e^{W_{y_i}^T X_i}}{\sum_{j=1}^{C} e^{W_j^T X_i}} - 1\right) X_i, \tag{8}$$

$$\frac{\partial L}{\partial W_j} = \frac{e^{W_j^T X_i}}{\sum_{j'=1}^{C} e^{W_{j'}^T X_i}} X_i, where j \neq y_i. \tag{9}$$

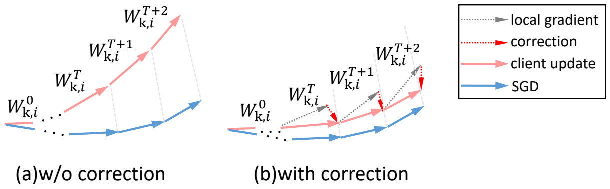
Similarly, for FedGC we also calculate the gradient of ℓ_{eq}. Then, $\frac{\partial \ell_{eq}}{\partial W_{k,j}}$, $\frac{\partial \ell_{eq}}{\partial W_{z,j}}$ and $\frac{\partial \ell_{eq}}{\partial W_{k,y_i}}$ can be expressed as:

$$\frac{\partial \ell_{eq}}{\partial W_{k,j}} = \frac{e^{W_{k,j}^T X_{k,i}}}{\sum_{j'=1}^{C_k} e^{W_{k,j'}^T X_{k,i}}} X_{k,i}, \tag{10}$$

$$\frac{\partial \ell_{eq}}{\partial W_{z,j}} = \frac{e^{W_{z,j}^T W_{k,y_i}} W_{k,y_i}}{e^{W_{k,y_i}^T W_{k,y_i}} + \sum_{z\neq k} \sum_{j'=0}^{C_z} e^{W_{z,j'}^T W_{k,y_i}}}, \tag{11}$$

$$\frac{\partial \ell_{eq}}{\partial W_{k,y_i}} = \left(\frac{e^{W_{k,y_i}^T X_{k,i}}}{\sum_{j=1}^{C_k} e^{W_{k,j}^T X_{k,i}}} - 1\right) X_{k,i}. \tag{12}$$

Let $D_{k,i}$ denote the distance between $W_{k,i}$ and X^k, $D_{k,i} = W_{k,y_i} - X_{k,i}$. We assume a well trained feature on local data due to it's easy convergence on local data, *i.e.* $D_{k,i} \to 0$, then we have $W_{k,y_i} \to X_{k,i}$. We can approximate:



(a)w/o correction

(b)with correction

Figure 2: Update steps of class embedings on a single client. (a) The divergence between FedPE and SGD becomes much larger w/o correction. (b) Gradient correction term ensures the update moves towards the true optimum.

$$\frac{\partial \ell_{eq}}{\partial W_{z,j}} \approx \frac{e^{W_{z,j}^T X_{k,i}} X_{k,i}}{e^{W_{k,y_i}^T X_{k,i}} + \sum_{z\neq k} \sum_{j'=0}^{C_z} e^{W_{z,j'}^T X_{k,i}}}. \tag{13}$$

The parameters are updated by SGD as $w_k' = w_k - \eta \frac{\partial \ell_{eq}}{\partial w_k}$, where η is step-size. Here for simplicity, we simplify Eq. 8 as $\frac{\partial L}{\partial W_{y_i}} = \alpha X_i$, Eq. 9 as $\frac{\partial L}{\partial W_j} = \beta X_i$, and Eq. 12 as $\frac{\partial \ell_{eq}}{\partial W_{k,y_i}} = \alpha' X_{k,i}$. Eq. 10 as $\frac{\partial \ell_{eq}}{\partial W_{k,j}} = \beta' X_{k,i}$ Eq. 13 as $\frac{\partial \ell_{eq}}{\partial W_{z,j}} = \gamma' X_{k,i}$. We consider the direction of gradients, thus Eq. 12 will act as a substitute for Eq. 8 in within-client optimization, Eq. 10 will act as a substitute for Eq. 9 in within-client optimization. The collaborative effect of both terms act as local gradient in Fig. 2. The mismatch of the magitude can be alleviated by adjusting the learning rate of class embeddings.

More importantly, Eq. 13 performs cross-client optimization and act as a correction term in Fig. 2 to correct gradient in cross-client optimization, introducing a gradient of cross-client samples to Eq. 10. And Eq. 13 has the same direction as Eq. 9. And for magitude, the denominator of Eq. 13 lacks term $\sum_{j=0,j\neq i}^{C_k} e^{W_{k,j}^T X_{k,i}}$ compared to standard SGD, but with a well done local optimization, we have

$\sum_{j=0,j\neq i}^{C_k} e^{W_{k,j}^T X_{k,i}} \ll \sum_{z\neq k}\sum_{j=0}^{C_z} e^{W_{z,j}^T X_{k,i}}$. Therefore, magitude of Eq. 13 and Eq. 9 are approximately equal. Thus, Eq. 13 together with Eq. 11 can act as a substitute for Eq. 9, and add a missing cross-client item. Therefore, FedGC can push the class embeddings toward the similar dirction as standard SGD and guarantees higher privacy.

Remark: Another simple way to introduce cross-client constraint is to minimize: $\sum_{z\neq k}\sum_{j=0}^{C_z} W_{z,j}^T W_{k,i}$, we call it cosine regularizer. For particular $W_{z,j}$, cosine regularizer introduce gradient $\frac{\partial \ell}{\partial W_{z,j}} = \sum_{z\neq k} W_{k,i}$. We show that our proposed softmax regularizer can act as a correction term for local softmax and also can be regarded as a weighted version of $\sum_{z\neq k}\sum_{j=0}^{C_z} W_{z,j}^T W_{k,i}$ from the perspective of backward propagation. Our proposed softmax regularizer generate gradient of larger magitude for more similar embeddings (hard example), thus it can also be regarded as a regularization term with hard example mining. In addition, we defined the softmax regularizer following the form of softmax. Thus, several loss functions which are the variants of softmax (*e.g.* ArcFace, CosFace) can be obtained with minor modification on softmax regularizer.

Extend FedGC to More General Case

In the above analysis, we adopt identity mutual exclusion assumption $\mathcal{Y}_k \cap \mathcal{Y}_z = \varnothing$. In fact, FedGC is to solve the problem of missing cross-client optimization. FedGC can also be applied to general case. We generalize the above mentioned situations, that is, some IDs are mutually exclusive and some IDs are shared. For example, there is an identity l shared by a client group K_l. After each round of communication, server takes the average of $W_{n,l}^t, n \in K_l$ and apply our proposed softmax regularizer (only exclusive clients are introduced, in this case client $K - K_l$) to correct its gradient. In this way, we can get W_l updated in the direction similar to the standard softmax. With minor modifications to above analysis, we can prove the applicability of FedGC in general case.

Relation to Other Methods

Multi-task learning. Multi-task learning combines several tasks to one system aiming to improve the generalization ability (Seltzer and Droppo 2013). Considering a multi-task learning system with input data x_i, the overall objective function is a combination of several subobject loss functions, written as $L = \sum_j L_j\left(\theta, W_j, x_i\right)$, where θ is generic parameters and $W_j, j \in [1, 2, \cdots]$ are task-specific parameters. While in FedGC, Eq. 3 can also be regarded as a combination of many class-dependent changing tasks $L_k = \frac{1}{n_k}\sum_{j=1}^{n_k} \ell_k\left(f_k(x_j^k), y_j^k\right), k \in [1, \cdots, K]$. In general, multi-task learning is conducted end-to-end and training on a single device. While in FedGC, the model is trained with class-exclusive decentralized non-IID data. Thus, our method can be also regarded as a decentralized version of multi-task learning.

Generative Adversarial Nets (GAN). Based on the idea of game theory, GAN is essentially a two players minimax problem, $\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$, which converges to a Nash equilibrium. In FedGC, client optimization and server optimization can be regarded as a process of adversary learning, where clients tend to minimize the similarity of within-client class embeddings, $L_k = \frac{1}{n_k}\sum_{j=1}^{n_k} \ell_k\left(f_k(x_j^k), y_j^k\right)$. But server tends to minimize the similarity of cross-client class embeddings and encourages within-client class embeddings to be more compact, $Reg(W)$. By performing adversary learning similar to GAN, the network can learn more discriminative representations of class embeddings.

Experiments

Implementation Details

Datasets. Considering that federated learning is extremely time-consuming, we employ CASIA-WebFace (Yi et al. 2014) as training set. CASIA-WebFace is collected from internet and contains about 1,000 subjects and 500,000 images. To simulate federated learning setting, we randomly divide training set into 36 clients. For test, we explore the verification performance of proposed FedGC on benchmark datasets (LFW (Huang et al. 2008), CFP-FP (Sengupta et al. 2016), AgeDB-30 (Moschoglou et al. 2017), SLLFW (Deng et al. 2017), CPLFW (Zheng and Deng 2018), CALFW (Zheng, Deng, and Hu 2017), and VGG2-FP (Cao et al. 2018)). We also explore on large-scale image-datasets (MegaFace (Kemelmacher-Shlizerman et al. 2016), IJB-B (Whitelam et al. 2017) and IJB-C (Maze et al. 2018)).

Experimental Settings. In data preprocessing, we use five facial landmarks for similarity transformation, then crop and resize the faces to (112×112). We employ the ResNet-34 (He et al. 2016) as backbone. We train the model with 2 synchronized 1080Ti GPUs on Pytorch. The learning rate is set to a constant of 0.1. The learning rate is kept constant without decay which is similar to the recent federated works. The batch size is set as 256. For fair comparison, the learning rate is also kept 0.1 in centralized standard SGD. We set momentum as 0.9 and weight decay as 5e-4.

Ablation Study

Fraction of participants. We compare the fraction of participants $C \in [0, 1]$. In each communication round, there are $C \cdot K$ clients conduct optimization in parallel on LFW. Table 1 shows the impact of varying C for face recognition models. We train the models with the guidance of Softmax. It is shown that with the increasing of client participation C, the performance of the model also increased. And FedGC still ourperforms the baseline model by a notable margin.

Regularizer multiplier λ. We perform an analysis of the learning rate multiplier of the softmax regularizer λ on LFW. As shown in Table 2, FedGC achieves the best performance when λ is 20. It is shown that a large multiplier also cause network collapsing, as it makes within-client class embeddings collapse to one point. When λ is very small, then the model degenerates into baseline model FedPE.

Balanced *v.s.* Unbalanced Partition. We compare the verification performance according to the partition of datasets. Here we constructed a unbalanced partition by logarithmic normal distribution: $\ln X \sim N\left(0, 1\right)$. We perform an analysis on the model with softmax loss functions on

Method	$C = 0.25$	$C = 0.5$	$C = 0.75$	$C = 1$
Softmax-FedPE	93.12	93.83	94.32	94.77
Softmax-FedGC	**97.07**	**97.98**	**98.13**	**98.40**

Table 1: Verification performance on LFW of different participance fraction C with softmax loss function.

Method	$\lambda = 1$	$\lambda = 20$	$\lambda = 50$
Softmax-FedGC	95.20	98.40	97.42

Table 2: Verification performance on LFW of different learning rate multiplier λ with softmax loss function.

LFW. In table 3, it shows that unbalanced partition even improve the performance of network to some extent. We find that the clients which holds larger scale dataset than average contribute significantly to network and make it generate more discriminative representations. And FedGC still outperforms baseline model on both balanced and unbalanced datasets.

Regularizer *v.s.* Fixed. It has been proved that random initialized vectors in high dimensional spaces (512 in this case) are almost orthogonal to each other. A naïve way to prevent the class embeddings from collapsing into a overlapping space is keep the class embeddings fixed to initialization. Table 5 shows that proposed FedGC outperforms model with fully-connected layer fixed ("-Fixed"). For softmax loss function, simply fixing the last fully-connected layer leads to a better accuracy. However, for Arcface and Cosface which introduce a more strict constraint, the performance of the model is even worse than baseline model. Intuitively, random initialized orthogonal vectors lack semantic information, and it confuses the network in a more difficult classification task. Thus, it is shown that the performance is further increased with adaptive optimization (FedGC).

Cosine *v.s.* Softmax Regularizer. We replace softmax regularizer with cosine regularizer, namely FedCos: $\sum_{z\neq k}\sum_{j=0}^{C_z} W_{z,j}^T W_{k,i}$, and guided by softmax loss function. We show the verification result on LFW in Table 4. Although cosine regularizer shows a better accuracy than FedPE, it is still worse than FedGC. Because softmax regularizer can be regarded as a hard sample mining version of cosine regularizer, and also match the gradient in standard softmax. Thus, the superiority of softmax regularizer is proved experimentally.

Visualization

To show the effectiveness of FedGC, the visualization comparisons are conducted at feature level. We select four pairs of classes to compare FedGC and FedPE. In each pair, the two classes are from different clients and their corresponding class embeddings are highly similar in FedPE model. The features are extracted from softmax model and visualized by t-SNE (Maaten and Hinton 2008), as shown in Fig. 3(a) and Fig. 3(b), the representations of the 4 pairs tends to gather to a point and form 4 clusters in FedPE, but the representations tends to spreadout and clustered by themselves in FedGC. We also illustrate the angle distributions of all 8 selected cross-client classes. For each pair, we calculate pair-wise cosine similarity of two classes' samples. In Fig. 3(c) and Fig. 3(d), we can clearly find that the cross-client class similarity significantly decreases in FedGC which encourage a larger cross-client class angle.

Method	LFW	CFP-FP	AgeDB
Balanced-FedPE	94.77	81.90	78.38
Balanced-FedGC	**98.40**	**90.20**	**85.85**
Unbalanced-FedPE	96.27	85.26	81.22
Unbalanced-FedGC	**98.80**	**91.56**	**88.78**

Table 3: Verification performance on LFW of different data partition with softmax loss function.

	FedPE	FedCos	FedGC
LFW	94.77	96.63	**98.40**

Table 4: Verification performance on LFW of different form of regularization with softmax loss function.

Evaluations

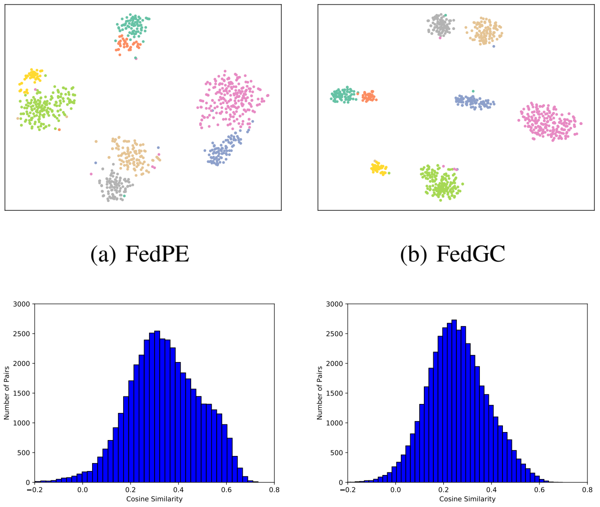
LFW, CALFW, CPLFW, CFP-FP, VGG2-FP SLLFW and AgeDB-30. In this section, we explore the performance of different loss functions (Softmax, Cosface (Wang et al. 2018c), Arcface (Deng et al. 2019)). We set the margin of Cosface (Wang et al. 2018c) at 0.35. For Arcface (Deng et al. 2019), we set the feature scale s to 64 and choose the angular margin m at 0.5. The performance of a model trained with federated learning algorithms is inherently upper bounded by that of model trained in the centralized fashion. Table 5 shows the experiments results, where "-X" means the dataset is trained by method "X". FedGC achieves the highest average accuracy for all loss functions (Softmax, Arcface , Cosface) and performs better on all of the above datasets. For ArcFace($m = 0.5$), the centralized method even achieves a poor performance worse than FedPE. And FedGC can also match the performance of conventional centralized methods.

MegaFace. The MegaFace dataset (Kemelmacher-Shlizerman et al. 2016) includes 1M images of 690K different individuals as the gallery set and 100K photos of 530 unique individuals from FaceScrub (Ng and Winkler 2014) as the probe set. It measures TPR at 1e-6 FPR for verification and Rank-1 retrieval performance for identification. In Table 6, adopting FaceScrub as probe set and using the wash list provided by DeepInsight (Deng et al. 2019), FedGC outperforms the baseline model FedPE by a large margin in different loss functions on both verification and identification tasks. Some centralized methods (Softmax, ArcFace($m = 0.5$)) even show a poor performance when the learning rate is 0.1. It shows that FedGC can match the performance of conventional centralized methods.

IJB-B and IJB-C. The IJB-B dataset (Whitelam et al. 2017) contains 1, 845 subjects with 21.8K still images and 55K frames from 7, 011 videos. In total, there are 12, 115

Method	LFW	CFP-FP	AgeDB	CALFW	CPLFW	SLLFW	VGG2-FP	Average
Softmax*	99.84	89.39	87.62	84.83	76.08	92.33	88.18	88.32
-FedPE	94.77	81.90	78.38	74.15	64.40	80.42	80.32	79.19
-FedPE+Fixed	96.11	83.67	80.28	77.95	66.27	84.23	82.70	81.60
-FedGC	**98.40**	**90.20**	**85.85**	**81.47**	**71.88**	**90.38**	**87.64**	**86.55**
CosFace($m = 0.35$)*	99.10	90.79	91.37	89.53	80.20	95.95	89.10	90.86
-FedPE	98.17	86.90	86.28	83.68	72.67	91.15	85.24	86.30
-FedPE+Fixed	96.35	73.01	81.77	79.25	62.15	86.57	75.16	79.18
-FedGC	**98.83**	**88.60**	**90.00**	**87.82**	**76.72**	**94.02**	**85.74**	**88.82**
ArcFace($m = 0.5$)*	97.62	90.50	83.37	77.33	70.95	86.28	89.40	85.06
-FedPE	98.18	87.23	86.13	82.47	71.77	91.05	85.70	86.08
-FedPE+Fixed	95.85	64.43	79.15	77.53	58.63	85.67	66.70	75.42
-FedGC	**98.65**	**87.77**	**89.27**	**86.47**	**75.17**	**93.58**	**84.80**	**87.96**

Table 5: Verification results (%) of different loss functions (Softmax, Cosface, Arcface) and method on 7 verification datasets. FedGC surpass others and enhance the average accuracy. * indicates the re-implementation by our code.



(a) FedPE

(b) FedGC

(c) FedPE

(d) FedGC

Figure 3: Visualization of selected 8 classes from training set. (a)(b) t-SNE (Maaten and Hinton 2008) data distribution; (c)(d) Histogram of pairwise cosine similarity.

Method	Ver.(%)	Id.(%)
Softmax*	61.21	59.65
-FedPE	36.83	34.08
-FedGC	**69.87**	**61.26**
CosFace($m = 0.35$)*	83.30	79.09
-FedPE	62.62	57.91
-FedGC	**72.82**	**70.96**
ArcFace($m = 0.5$)*	50.51	35.18
-FedPE	64.53	58.12
-FedGC	**71.96**	**68.75**

Table 6: Verification TPR (@FPR=1e-6) and identification Rank-1 on the MegaFace Challenge 1.

Method	IJB-B		IJB-C	
	Ver.(%)	Id.(%)	Ver.(%)	Id.(%)
Softmax*	72.60	74.81	75.06	76.05
-FedPE	54.33	64.44	57.85	65.35
-FedGC	**69.23**	**78.52**	**71.33**	**79.52**
CosFace($m = 0.35$)*	76.79	78.35	79.45	79.90
-FedPE	74.24	78.10	77.12	79.10
-FedGC	**80.28**	**82.10**	**83.40**	**83.44**
ArcFace($m = 0.5$)*	56.64	60.14	59.38	59.79
-FedPE	73.42	76.40	75.74	76.82
-FedGC	**75.11**	**78.33**	**78.13**	**79.28**

Table 7: Verification TPR (@FPR=1e-3) and identification Rank-1 on the IJB-B and IJB-C benchmarks.

templates with 10, 270 genuine matches and 8M impostor matches. The IJB-C dataset (Maze et al. 2018) is a further extension of IJB-B, having 3, 531 subjects with 31.3K still images and 117.5K frames from 11, 779 videos. In total, there are 23, 124 templates with 19, 557 genuine matches and 15, 639K impostor matches. The verification TPR at 1e-3 FPR and identification Rank-1 are reported in Table 7. FedGC shows significant improvements and surpasses all candidates by a large margin. Compared with centralized method on all of three loss functions, FedGC can match the performance of conventional centralized methods on both IJB-B and IJB-C datasets.

Conclusion

In this paper, we rethink the federated learning problem for face recognition on privacy issues, and introduce a novel face-recognition-specialized federated learning framework, FedGC, that consists of a set of local softmax and a softmax-based regularizer to effectively learn discriminative face representtations with decentralized face data. FedGC can effectively enhance the discriminative power of cross-client class embeddings and enable the network to update towards the same direction as standard SGD. FedGC not only guarantees higher privacy but also contributes to network convergence. Extensive experiments have been conducted over popular benchmarks to validate the necessity and effectiveness of FedGC that can match the performance of conventional centralized methods.

References

Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H. B.; Mironov, I.; Talwar, K.; and Zhang, L. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 308–318.

Aggarwal, D.; Zhou, J.; and Jain, A. K. 2021. FedFace: Collaborative Learning of Face Recognition Model. *arXiv preprint arXiv:2104.03008*.

Ahonen, T.; Hadid, A.; and Pietikainen, M. 2006. Face description with local binary patterns: Application to face recognition. *IEEE transactions on pattern analysis and machine intelligence*, 28(12): 2037–2041.

Bai, F.; Wu, J.; Shen, P.; Li, S.; and Zhou, S. 2021. Federated Face Recognition. *arXiv preprint arXiv:2105.02501*.

Cao, Q.; Shen, L.; Xie, W.; Parkhi, O. M.; and Zisserman, A. 2018. Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, 67–74. IEEE.

Deng, J.; Guo, J.; Liu, T.; Gong, M.; and Zafeiriou, S. 2020. Sub-center arcface: Boosting face recognition by large-scale noisy web faces. In *European Conference on Computer Vision*, 741–757. Springer.

Deng, J.; Guo, J.; Xue, N.; and Zafeiriou, S. 2019. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4690–4699.

Deng, J.; Zhou, Y.; and Zafeiriou, S. 2017. Marginal loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 60–68.

Deng, W.; Hu, J.; Zhang, N.; Chen, B.; and Guo, J. 2017. Fine-grained face verification: FGLFW database, baselines, and human-DCMN partnership. *Pattern Recognition*, 66: 63–73.

Duan, Y.; Lu, J.; and Zhou, J. 2019. Uniformface: Learning deep equidistributed representation for face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3415–3424.

Guo, Y.; Zhang, L.; Hu, Y.; He, X.; and Gao, J. 2016. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *European conference on computer vision*, 87–102. Springer.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Huang, G. B.; Mattar, M.; Berg, T.; and Learned-Miller, E. 2008. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments.

Karimireddy, S. P.; Kale, S.; Mohri, M.; Reddi, S. J.; Stich, S. U.; and Suresh, A. T. 2019. Scaffold: Stochastic controlled averaging for federated learning. *arXiv preprint arXiv:1910.06378*.

Kemelmacher-Shlizerman, I.; Seitz, S. M.; Miller, D.; and Brossard, E. 2016. The megaface benchmark: 1 million faces for recognition at scale. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4873–4882.

Kim, Y.; Park, W.; Roh, M.-C.; and Shin, J. 2020. GroupFace: Learning Latent Groups and Constructing Group-based Representations for Face Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5621–5630.

Kim, Y.; Park, W.; and Shin, J. 2020. BroadFace: Looking at Tens of Thousands of People at Once for Face Recognition. In *European Conference on Computer Vision*, 536–552. Springer.

Li, T.; Sahu, A. K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; and Smith, V. 2018. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*.

Liu, C.; and Wechsler, H. 2002. Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. *IEEE Transactions on Image processing*, 11(4): 467–476.

Liu, W.; Wen, Y.; Yu, Z.; Li, M.; Raj, B.; and Song, L. 2017. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 212–220.

Liu, W.; Wen, Y.; Yu, Z.; and Yang, M. 2016. Large-margin softmax loss for convolutional neural networks. In *ICML*, volume 2, 7.

Maaten, L. v. d.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(Nov): 2579–2605.

Marriott, R. T.; Romdhani, S.; and Chen, L. 2021. A 3D GAN for Improved Large-pose Facial Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13445–13455.

Maze, B.; Adams, J.; Duncan, J. A.; Kalka, N.; Miller, T.; Otto, C.; Jain, A. K.; Niggel, W. T.; Anderson, J.; Cheney, J.; et al. 2018. Iarpa janus benchmark-c: Face dataset and protocol. In *2018 International Conference on Biometrics (ICB)*, 158–165. IEEE.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, 1273–1282. PMLR.

Moschoglou, S.; Papaioannou, A.; Sagonas, C.; Deng, J.; Kotsia, I.; and Zafeiriou, S. 2017. Agedb: the first manually collected, in-the-wild age database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 51–59.

Ng, H.-W.; and Winkler, S. 2014. A data-driven approach to cleaning large face datasets. In *2014 IEEE international conference on image processing (ICIP)*, 343–347. IEEE.

Parkhi, O. M.; Vedaldi, A.; and Zisserman, A. 2015. Deep face recognition.

Schroff, F.; Kalenichenko, D.; and Philbin, J. 2015. Facenet: A unified embedding for face recognition and clustering. In

Proceedings of the IEEE conference on computer vision and pattern recognition, 815–823.

Seltzer, M. L.; and Droppo, J. 2013. Multi-task learning in deep neural networks for improved phoneme recognition. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 6965–6969. IEEE.

Sengupta, S.; Chen, J.-C.; Castillo, C.; Patel, V. M.; Chellappa, R.; and Jacobs, D. W. 2016. Frontal to profile face verification in the wild. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 1–9. IEEE.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Sun, Y.; Cheng, C.; Zhang, Y.; Zhang, C.; Zheng, L.; Wang, Z.; and Wei, Y. 2020. Circle loss: A unified perspective of pair similarity optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6398–6407.

Sun, Y.; Wang, X.; and Tang, X. 2015. Deeply learned face representations are sparse, selective, and robust. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2892–2900.

Taigman, Y.; Yang, M.; Ranzato, M.; and Wolf, L. 2014. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1701–1708.

Wang, F.; Chen, L.; Li, C.; Huang, S.; Chen, Y.; Qian, C.; and Change Loy, C. 2018a. The devil of face recognition is in the noise. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 765–780.

Wang, F.; Cheng, J.; Liu, W.; and Liu, H. 2018b. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 25(7): 926–930.

Wang, H.; Wang, Y.; Zhou, Z.; Ji, X.; Gong, D.; Zhou, J.; Li, Z.; and Liu, W. 2018c. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5265–5274.

Wang, M.; and Deng, W. 2018. Deep Face Recognition: A Survey. *arXiv preprint arXiv:1804.06655*.

Whitelam, C.; Taborsky, E.; Blanton, A.; Maze, B.; Adams, J.; Miller, T.; Kalka, N.; Jain, A. K.; Duncan, J. A.; Allen, K.; et al. 2017. Iarpa janus benchmark-b face dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 90–98.

Wolf, L.; Hassner, T.; and Maoz, I. 2011. Face recognition in unconstrained videos with matched background similarity. In *CVPR 2011*, 529–534. IEEE.

Yi, D.; Lei, Z.; Liao, S.; and Li, S. Z. 2014. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*.

Yin, H.; Molchanov, P.; Alvarez, J. M.; Li, Z.; Mallya, A.; Hoiem, D.; Jha, N. K.; and Kautz, J. 2020. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8715–8724.

Yu, F. X.; Rawat, A. S.; Menon, A. K.; and Kumar, S. 2020. Federated Learning with Only Positive Labels. *arXiv preprint arXiv:2004.10342*.

Yuan, H.; and Ma, T. 2020. Federated Accelerated Stochastic Gradient Descent. *arXiv preprint arXiv:2006.08950*.

Zheng, T.; and Deng, W. 2018. Cross-pose lfw: A database for studying cross-pose face recognition in unconstrained environments. *Beijing University of Posts and Telecommunications, Tech. Rep*, 5.

Zheng, T.; Deng, W.; and Hu, J. 2017. Cross-age lfw: A database for studying cross-age face recognition in unconstrained environments. *arXiv preprint arXiv:1708.08197*.

Zhu, Z.; Huang, G.; Deng, J.; Ye, Y.; Huang, J.; Chen, X.; Zhu, J.; Yang, T.; Lu, J.; Du, D.; et al. 2021. WebFace260M: A Benchmark Unveiling the Power of Million-Scale Deep Face Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10492–10502.