FEDERATED LEARNING WITH DATA-AGNOSTIC DISTRIBUTION FUSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated learning has emerged as a promising distributed machine learning paradigm to preserve data privacy. One of the fundamental challenges of federated learning is that data samples across clients are usually not independent and identically distributed (non-IID), leading to slow convergence and severe performance drop of the aggregated global model. In this paper, we propose a novel data-agnostic distribution fusion based model aggregation method called FedDAF to optimize federated learning with non-IID local datasets, based on which the heterogeneous clients' data distributions can be represented by the fusion of several virtual components with different parameters and weights. We develop a variational autoencoder (VAE) method to derive the optimal parameters for the fusion distribution using the limited statistical information extracted from local models, which optimizes model aggregation for federated learning by solving a probabilistic maximization problem. Extensive experiments based on various federated learning scenarios with real-world datasets show that FedDAF achieves significant performance improvement compared to the state-of-the-art.

1 INTRODUCTION

Federated learning (FL) has emerged as a novel distributed machine learning paradigm that allows a global deep neural network (DNN) model to be trained by multiple mobile clients collaboratively. In such a paradigm, mobile clients train local models based on datasets generated by edge devices such as sensors and smartphones, and the server is responsible to aggregate the parameters from local models to form a global model without transferring data to a central server. Federated learning has been drawn much attention in mobile-edge computing Konecný et al. (2016); Sun et al. (2017) with its advantages in preserving data privacy Zhu & Jin (2020); Keller et al. (2018) and enhancing communication efficiency Smith et al. (2018); McMahan et al. (2017); Wang et al. (2020). Besides, a lot of algorithms have been proposed to improve the resource allocation fairness, communication efficiency, and convergence rate for federated learning Kairouz et al. (2019); Lim et al. (2020), which include LAG Chen et al. (2018), Zeno Xie et al. (2019), AFL Mohri et al. (2019), q-FedSGD Li et al. (2020b), FedMA Wang et al. (2020), etc.

One of the fundamental challenges of federated learning is the non-IID data sampling from heterogeneous clients. In real-world federated learning scenarios, local datasets are typically non-IID, and the local models trained on them are significantly different from each other. Aggregating the local models with simple averaging may cause severe performance degradation in terms of model accuracy and communication rounds required for convergence. It was reported in Zhao et al. (2018) that the accuracy of a convolutional neural network (CNN) model trained by FedAvg reduces by up to 55% for a highly skewed non-IID dataset. The work in Wang et al. (2020) showed that the accuracy of VGG model trained with FedAvg and its variants dropped from 61% to under 50% when the client number increases from 5 to 20 on heterogeneous data partition.

Several works have been made to address the non-IID challenge. Li et al. (2020a) modified FedAvg by adding a dissimilarity bound on local datasets and a proximal term on the local model parameter to tackle heterogeneity. However, it poses restrictions on the local updates to be closer to the initial global model, which may lead to model bias. Zhao et al. (2018) proposed a data sharing strategy to improve training on non-IID data by creating a small subset of data to share between all clients. However, data sharing could weaken the privacy requirement of federated learning. Several works adopted clustering based approaches to tackle non-IID settings, where client models were partitioned into clusters and model aggregation in performed in cluster level Chen et al. (2020); Xie et al. (2020); Ghosh et al. (2020); Duan et al. (2020). However, clustered federated learning may suffer from privacy leakage with shared data to cluster clients, and its performance relied on the cluster number which is a hyperparameter needed to be manually adjusted from task to task.

In this paper, we propose a novel data-agnostic distribution fusion method called `FedDAF` for federated learning on non-IID data. We introduce a distribution fusion model to describe the global data distribution as a fusion of several visual components belonging to the same parametric family of distributions, which is ideal for representing non-IID data generated from heterogeneous clients. However, applying a distribution fusion for federated learning encounters several difficulties. First, due to the privacy policy of federated learning, the local datasets are inaccessible and their distributions are unknown to the server, so it is impossible for the server to form a global distribution based on observing to data samples. Second, the number of distribution components and their fusion weights are unspecified without the knowledge of local data, making it a challenging task to develop such a fusion model for federated learning.

To tackle these issues, we propose an efficient solution to optimize the distribution fusion federated learning problem with variational inference. Since the local data is inaccessible to the server, our method is based on the limited statistical information embedded in the normalization layers of DNN models, i.e., the means and standard deviations of the feature maps (the outputs of intermediate layers). Those information can be extracted from the received local model parameters, which can be used to infer a global distribution. We develop a variational autoencoder (VAE) model to derive the optimal parameters of distribution fusion components based on the observed information, and applied the derived parameters to optimize federated learning with non-IID data. Extensive experiments based on a variety of federated learning scenarios with non-IID data show that `FedDAF` significantly outperforms the state-of-the-arts.

The contributions of our work are summarized as follows.

- We propose a novel data-agnostic distribution fusion based model aggregation method called `FedDAF` to address the data heterogeneity problem in federated learning. It represents the global data by a fusion model of several virtual distribution components with different fusion weights, which is ideal to describe non-IID data generated from heterogeneous clients.
- We develop a variational autoencoder (VAE) method to derive the optimal parameters for the data-agnostic distribution fusion federated learning model. Without violating the privacy principle of federated learning, the proposed method uses limited statistical information embedded in DNN models to infer a target global distribution with a maximum probability. Based on the inferred parameters, an optimal model aggregation strategy can be developed for federated learning under non-IID data.
- We conduct extensive experiments using five mainstream DNN models based on four real-world datasets under non-IID conditions. Compared to FedAvg and the state-of-the-art for non-IID data (FedProx, FedMA, IFCA, FedGroup, etc), the proposed `FedDAF` has better convergence and training efficiency, improving the global model's accuracy up to 12%.

2 Related Work

Federated learning Konečnỳ et al. (2015) is an emerging distributed machine learning paradigm that aims to build a global model based on datasets distributing across multiple clients. One of the standard parameter aggregation methods is FedAvg McMahan et al. (2017), which combined local stochastic gradient descent (SGD) on each client with a server that performs parameter averaging. Later, the lazily aggregated gradient (LAG) method Chen et al. (2018) allowed clients to run multiple epochs before model aggregation to reduce communication costs. The q-FedSGD Li et al. (2020b) method improved FedAvg with a dynamic SGD update step using a scale factor to achieve fair resource allocation among heterogeneous clients. The FedMA Wang et al. (2020) method, derived from AFL Mohri et al. (2019) and PFNM Yurochkin et al. (2019), demonstrated that permutations of layers could affect the parameter aggregation results, and proposed a layer-wise parameter-permutation aggregation method to solve the problem. The FedDyn Acar et al. (2021) method proposed a dynamic regularizer for each client at each round of aggregation, so that different models are aligned to alleviate the inconsistency between local loss and global loss.

Several works focused on optimizing federated learning under non-IID data. Zhao et al. used the earth mover's distance (EMD) to quantify data heterogeneity and proposed to use globally shared data for training to deal with non-IID Zhao et al. (2018). The RNN-based method Ji et al. (2019) adopted a meta-learning method to learn a new gradient from the received gradients and then applied it to update the global model. FedProx Li et al. (2020a) modified FedAvg by adding a heterogeneity bound on local datasets and a proximal term on the local model parameter to tackle heterogeneity. FedBN Li et al. (2021) suggested keeping the local Batch Normalization parameters not synchronized with the global model to mitigate feature shifts in non-IID data. FedGN Hsieh et al. (2020) replaced Batch Normalization with Group Normalization to avoids the accuracy loss induced by the skewed distribution of data labels. Yang et al. provided theoretical evidence on linear speedup for convergence of FedAvg under non-IID datasets with partial worker participation Yang et al. (2021). The federated cluster learning Chen et al. (2020) Xie et al. (2020) Ghosh et al. (2020) Duan et al. (2020) partitioned clients into clusters to address data heterogeneity, and aggregated different models for different clusters. For example, IFCA Ghosh et al. (2020) alternately estimated the cluster identities of the clients and optimized the model parameters for the clusters via gradient descent. FedGroup Duan et al. (2020) grouped the clients based on the similarities between their optimization directions to improve training efficiency. The personalized federated learning Smith et al. (2017) Khodak et al. (2019) Liang et al. (2020) Peng et al. (2020) further adopted multi-task learning and meta-learning to train personalized model for individual client. Different from clustered FL and personalized FL that form multiple personalized models, our work focus on training a single global model from heterogeneous clients. We propose a novel data agnostic fusion with variational inference to optimize the model aggregation process in federated learning under non-IID conditions, which has not yet been addressed in the literature.

3 Formulation of Federated Learning with Distribution Fusion

We consider the following federated learning scenario with non-IID data. We assume there are K clients involved in the learning system, and $\mathcal{D}_k$ $(k = 1, \cdots, K)$ indicates the data distribution of the kth client. The learning process repeats for multiple communication rounds. At the beginning of each round, each client downloads a learning model from the global server, and trains the model individually with its local data to minimize the local loss, i.e., $\min \mathcal{L}(\mathbf{w}, \mathcal{D}_k)$ $(k = 1, \cdots, K)$ where $\mathcal{L}(\cdot)$ is the loss function and $\mathbf{w}$ is the earnable model parameters. The optimal local model of the kth client is given by:

$$\mathbf{w}_k^* = arg \min_{\mathbf{w}\in\mathcal{W}} \mathcal{L}(\mathbf{w}, \mathcal{D}_k). \tag{1}$$

After receiving model parameters $\mathbf{w}_k$ from local clients, with $\tilde{\mathcal{D}}$ to be global distribution of all $\mathcal{D}_k$, the optimal global model in the server is given by:

$$\mathbf{w}^* = arg \min_{\mathbf{w}\in\mathcal{W}} \mathcal{L}(\mathbf{w} = aggr(\mathbf{w}_1, \cdots, \mathbf{w}_K), \tilde{\mathcal{D}}), \tag{2}$$

where $aggr(\mathbf{w}_1, \cdots, \mathbf{w}_K)$ is the strategy of the server to aggregate local models into a global model. Conventionally, the aggregation strategies are typically in the form of averaging or weighted-averaging in McMahan et al. (2017) Li et al. (2020a) Wang et al. (2020) Li et al. (2020b) Duan et al. (2020) Ghosh et al. (2020), etc. In this paper, we explore a distribution fusion method to derive the optimal model aggregation strategy.

Since both local data and their distributions are unknown to the server, we model the target global distribution $\tilde{\mathcal{D}}$ as a fusion of the distributions with M $(M \leq K)$ virtual components (M can be adaptively learned from the task): $\tilde{\mathcal{D}} = \sum_{m=1}^{M} \pi_m \bar{\mathcal{D}}_m$, where $\bar{\mathcal{D}}_m$ $(m = 1, \cdots, M)$ is the mth virtual distribution component and $\sum_{m=1}^{M} \pi_m = 1$ are the fusion weights. With the above model, each client's data distribution $\mathcal{D}_k$ can be allocated into several of the M components in $\{\bar{\mathcal{D}}_1, \ldots, \bar{\mathcal{D}}_M\}$. To formally describe the distribution fusion, we introduce a *distribution allocation vector* $\mathbf{c}_k$, that is defined as a zero-one vector $\mathbf{c}_k = [c_{km}|m = 1, \cdots, M]$, where $c_{km} = 1$ if the data of the kth client can be allocated to the mth virtual distribution component and otherwise $c_{km} = 0$. And $b_{km} = P(c_{km} = 1|\mathcal{D}_k)$ is normalized conditional probability of how much $\mathcal{D}_k$ been allocated to mth virtual component. With such notation, we consider allocating the distribution of K clients' data distributions to M virtual components as a probability event, and the distribution fusion model can be described as:

$$\tilde{\mathcal{D}} = \sum_{m=1}^{M} \pi_m \sum_{k=1}^{K} b_{km} \cdot c_{km} \cdot \mathcal{D}_k, \text{ s.t., } \sum_{m=1}^{M} b_{km} = 1. \tag{3}$$

Based on Eq. 3, the objective of model aggregation in Eq. 2 can be formulated as the following optimization problem:

$$(\pi^*, \mathbf{c}^*, \mathbf{b}^*) = arg \min_{\pi,\mathbf{c},\mathbf{b}} \mathcal{L}(\sum_{m=1}^{M} \pi_m \sum_{k=1}^{K} b_{km} \cdot c_{km} \cdot \mathbf{w}_k, \tilde{\mathcal{D}}), \text{ s.t.}, \sum_{k=1}^{K} \pi_k = 1, \sum_{m=1}^{M} b_{km} = 1. \quad (4)$$

The minimization problem in above formulation can be understood as finding the optimal fusion parameters π, $\mathbf{c}$ and $\mathbf{b}$ to maximize the probability of allocating the clients' data distribution to the most probable virtual distribution components, so that the expected global loss over the target distribution is minimized. Notice that in an extreme condition where the data are IID among all clients, the number of virtual components $M = 1$ and the objective in Eq. (4) equals to simple averaging, which makes the classical FedAvg McMahan et al. (2017) a special case of our model.

Next, we derive the solution of the optimization problem with a variational inference method.

4 Data-Agnostic Distribution Fusion Based on Variational Inference

Since local datasets are only accessible by their owners in federated learning for privacy protection, the local distributions $\mathcal{D}_1, \cdots, \mathcal{D}_K$ are unknown to the server, making derivation of target distribution $\tilde{\mathcal{D}}$ difficult. To confront this challenge, we propose a novel idea to use limited statistical information during model aggregation to approximate the optimal distribution fusion parameters.

In each communication round of federated learning, the clients will update their model parameters based on local data and report the updated models to the server for model aggregation. Although the private data is unknown, there are some statistical information embedded in the reported model parameters which can be used by the server to infer the local distributions. For example, in a DNN model, the statistical information can be extracted from the *normalization layers* such as batch normalization Ioffe & Szegedy (2015), layer normalization Ba et al. (2016), instance normalization Ulyanov et al. (2016), and group normalization Wu & He (2018), which typically contain the following statistical variables:

- $\hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\sigma}}_k$: the means and standard deviations of the feature maps (the outputs of intermediate layers) of the kth client's DNN model.
- $\hat{\boldsymbol{\beta}}_k, \hat{\boldsymbol{\gamma}}_k$: the *shifted means* and *scaled standard deviations* Ioffe & Szegedy (2015) of the feature maps of the kth client's DNN model.

We use $\mathbf{d}_k = \{\hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\sigma}}_k, \hat{\boldsymbol{\beta}}_k, \hat{\boldsymbol{\gamma}}_k\}$ to denote the observed statistical variable of the kth client. Since the real distribution $\mathcal{D}_k$ is unknown, we can approximate the objective in Eq. (4) by maximizing the probability of distribution allocation given the observed models' statistical information, which can be expressed as:

$$\begin{aligned} &(\pi^*, \mathbf{c}^*, \mathbf{b}^*) = arg \min_{\pi,\mathbf{c},\mathbf{b}} \mathcal{L}(\sum_{m=1}^{M} \pi_m \sum_{k=1}^{K} b_{km} \cdot c_{km} \cdot \mathbf{w}_k, \tilde{\mathbf{d}}), \\ &\text{s.t., } b_{km} = P(c_{km} = 1 | \mathbf{d}_k), \sum_{k=1}^{K} \pi_k = 1, \sum_{m=1}^{M} b_{km} = 1. \end{aligned} \quad (5)$$

Hence we convert a data-dependent optimization problem into a data-agnostic problem based on observable statistical variables to the server. Notice that the proposed method exchanges exactly the same information between server and clients as conventional FedAvg McMahan et al. (2017), which will not violate privacy protection in federated learning. Next, we introduce a variational autoencoder method to derive the optimal model parameters.

4.1 Variational AutoEncoder

Since the probabilities in Eq. (5) are hard to be expressed in mathematical form, we adopt a variational autoencoder (VAE) method to derive the optimal parameters π_m and $\mathbf{c}_k$ of the fusion based on variational inference. The plate notions of the VAE are shown in Fig. 1.

• $\mathbf{c}_k \in \{0, 1\}^M$ is a zero-one vector representing distribution allocation, where $c_{km} = 1$ represents allocating the distribution of the kth client to the mth virtual component. We assume that $\mathbf{c}_k$ is sampled from a Bernoulli distribution which is parameterized by $\boldsymbol{\lambda}_k = \{\lambda_{km} | m = 1, \cdots, M\}$, and $\boldsymbol{\lambda}_k$ is sampled from a Beta distribution $Beta(\boldsymbol{\zeta}_m, \boldsymbol{\kappa}_m)$ which is parameterized by $\boldsymbol{\zeta}_m, \boldsymbol{\kappa}_m$.

• $\mathbf{b}_k = \{b_{km} \in (0, 1) | m = 1, \cdots, M\}, \sum_{m=1}^{M} b_{km} = 1$, where b_{km} represents the allocation weight of the kth client to mth virtual component, and the sum of weights is 1. We assume that $\mathbf{b}_k$ is sampled from a Gaussian prior distribution $\mathcal{N}(\boldsymbol{\nu}_m, \varsigma_m)$ which is parameterized by $\boldsymbol{\nu}_m$ and ς_m.

• $\mathbf{z}_k = \sum_{m=1}^{M} b_{km} \cdot \tilde{\mathbf{z}}_m$ is a latent variable used by the decoder $\boldsymbol{\theta}$ to reconstruct the observed $\mathbf{d}_k$, where $\odot$ is the inner product of two vectors, and $\tilde{\mathbf{z}}_k$ means the sampled latent vector from every allocated distribution for kth client from the Gaussian prior distribution $\mathcal{N}(\boldsymbol{\nu}^{'}_m, \varsigma^{'}_m)$.

As illustrated in Fig. 1, the parameters of $Beta(\boldsymbol{\zeta}_m, \boldsymbol{\kappa}_m)$, $\mathcal{N}(\boldsymbol{\nu}_m, \varsigma_m)$ and $\mathcal{N}(\boldsymbol{\nu}^{'}_m, \varsigma^{'}_m)$ can be inferred with an variational encoder $\boldsymbol{\phi}$ based on the observed information $\mathbf{d}_k$, i.e., $\{\boldsymbol{\nu}_m, \varsigma_m, \boldsymbol{\nu}^{'}_m, \varsigma^{'}_m, \boldsymbol{\zeta}_m, \boldsymbol{\kappa}_m\} = \boldsymbol{\phi}(\mathbf{d}_k)$. In the meanwhile, the variables of $\mathbf{b}_k$ and $\tilde{\mathbf{z}}_m$ are used to compute a latent variable $\mathbf{z}_k$, which is further fed to a decoder $\boldsymbol{\theta}$ to reconstruct the observed data $\mathbf{d}_k$ with nonlinear transformation. By optimizing the parameters of the encoder-decoder, the optimal allocation vector $\mathbf{c}_k$ and the weight vector $\mathbf{b}_k$ can be derived, which can be further used to derived the fusion weights π_m.

The details of allocation encoder-decoder are explained as follows. As $\tilde{\mathbf{z}}_m$ is not related with allocation, we will not discuss here.

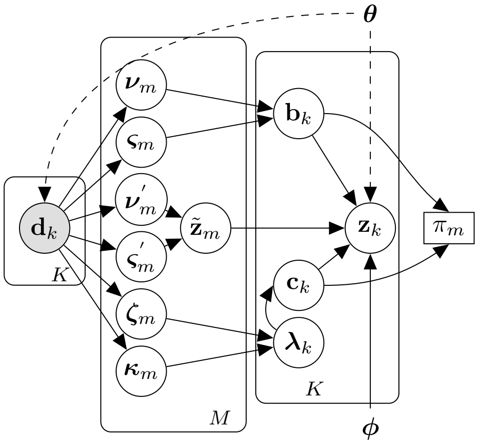


Figure 1: The variational Bayesian autoencoder using plate notations, where $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ are global variables representing the encoder's parameters and the decoder's parameters respectively.

Encoder: As shown in Fig. 1, in order to infer the latent vector $\mathbf{z}_k$, we should derive the variational posterior $q_{\boldsymbol{\phi}}(\boldsymbol{\lambda}_k, \mathbf{c}_k, \mathbf{b}_k)$. We employ a multi-head nonlinear model to infer the approximation of true posterior $p(\boldsymbol{\lambda}_k, \mathbf{c}_k, \mathbf{b}_k | \mathbf{d}_k)$ with variational posteriors, and apply the stochastic gradient variational Bayes (SGVB) Kingma & Welling (2014) algorithm to learn the model.

From Fig. 1 we know that variables in variational posterior are conditionally independent with the priori $p(\mathbf{d}_k)$. So we can decouple the variables as: $q_{\boldsymbol{\phi}}(\boldsymbol{\lambda}, \mathbf{c}, \mathbf{b}) = \prod_{k=1}^{K} \prod_{m=1}^{M} q_{\boldsymbol{\phi}}(b_{km}) \cdot q_{\boldsymbol{\phi}}(c_{km} | \lambda_{km}) \cdot q_{\boldsymbol{\phi}}(\lambda_{km})$, where the variational posterior distributions Nalisnick & Smyth (2017) can be derived by:

$$\begin{aligned} q_{\boldsymbol{\phi}}(\mathbf{b}_k) &\sim \mathcal{N}(\boldsymbol{\nu}_m, \varsigma_m), \\ q_{\boldsymbol{\phi}}(\boldsymbol{\lambda}_k) &\sim Beta(\boldsymbol{\zeta}_m, \boldsymbol{\kappa}_m), \\ q_{\boldsymbol{\phi}}(\mathbf{c}_k) &\sim Bernoulli(\prod_{m=1}^{M} \boldsymbol{\lambda}_{km}). \end{aligned} \tag{6}$$

Decoder: The decoder $\boldsymbol{\theta}$ takes the latent variable $\mathbf{z}_k$ as input to reconstruct the original observed data. According to Fig. 1, the derivation of $\mathbf{z}_k$ relies on three variables $\mathbf{b}_k$, $\boldsymbol{\lambda}_k$, and $\mathbf{c}_k$, whose variational posteriors are Gaussian, Beta, and Bernoulli distribution accordingly, as shown in Eq. (6). We infer the three latent variables as follows.

Since the posterior of $\mathbf{b}_k$ is a Gaussian distribution with differentiable Monte Carlo expectations, it can be easily inferred with the Stochastic Gradient Variational Bayes (SGVB) estimator Kingma & Welling (2014).

The posterior of $\boldsymbol{\lambda}_k$ is a Beta distribution, which is hard to be inferred with conventional variational inference algorithms. We approximate the posterior Beta with the Kumaraswamy distribution Nalisnick & Smyth (2017); Kumaraswamy (1980), a two-parameter continuous distribution also on the unit interval with a density function defined as:

$$Kumaraswamy(x; \boldsymbol{\zeta}_k, \boldsymbol{\kappa}_k) = \boldsymbol{\zeta}_k \boldsymbol{\kappa}_k x^{\boldsymbol{\zeta}_k - 1} (1 - x^{\boldsymbol{\zeta}_k})^{\boldsymbol{\kappa}_k - 1}, \tag{7}$$

where $\boldsymbol{\zeta}_k$ and $\boldsymbol{\kappa}_k$ are parameters of the distribution. It was proved that the Kumaraswamy approaches to the Beta albeit with high entropy, and it satisfies the differentiable and non-centered parameterization (DNCP) property with its closed-form inverse CDF Nalisnick & Smyth (2017). Therefore the samples of $\boldsymbol{\lambda}_k$ can be drawn via the inverse transform of Kumaraswamy:

$$\boldsymbol{\lambda}_k \sim (1-\xi^{\frac{1}{\boldsymbol{\kappa}_k}})^{\frac{1}{\boldsymbol{\zeta}_k}}, \text{ where } \xi \sim Uniform(0,1). \tag{8}$$

For the zero-one vector $\mathbf{c}_k$, we reparameterize it with the Beta distribution as in Eq. (6). Using the Gumbel-Max trick to draw samples $\mathbf{c}_k$ from a Bernoulli distribution with binary probabilities Jang et al. (2017), we have:

$$\mathbf{c}_{km} = arg\max_i (g_i + \log\prod_{i=1}^{2} \boldsymbol{\lambda}_{ki}), \tag{9}$$

where g_i is an IID sample drawn from $Gumbel(0,1)$. After deriving $\mathbf{b}_k$ and $\mathbf{c}_k$ and sampling latent vector $\tilde{\mathbf{z}}_k$ from every component which client k been allocated, we can compute the latent variable $\mathbf{z}_k$ with $\mathbf{z}_k = \sum_{m=1}^{M} b_{km} \cdot \tilde{\mathbf{z}}_m$. Then we use $\mathbf{z}_k$ to reconstruct the original observed data $\mathbf{d}_k$ with $p_{\boldsymbol{\theta}}(\hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\sigma}}_k, \hat{\boldsymbol{\beta}}_k, \hat{\boldsymbol{\gamma}}_k|\mathbf{z}_k)$. The decoder $\boldsymbol{\theta}$ can be parameterized by using a deep neural network to learn the model.

To derive the component weight π_m, we use a variant of the EM algorithm Dempster et al. (1977) with a softmax function:

$$\pi_m = \frac{\exp(\frac{1}{K}\sum_{k=1}^{K} q_{\boldsymbol{\phi}}(c_{km}) \cdot b_{km})}{\sum_{m=1}^{M}\exp(\frac{1}{K}\sum_{k=1}^{K} q_{\boldsymbol{\phi}}(c_{km}) \cdot b_{km})}. \tag{10}$$

4.2 Optimizing the Variational AutoEncoder

We optimize the proposed variational autoencoder as follows. The dashed lines in Fig. 1 denote the generative model $p_{\boldsymbol{\theta}}(\mathbf{z}_k)p_{\boldsymbol{\theta}}(\mathbf{d}_k|\mathbf{z}_k)$, and the solid lines denote the variational approximation $q_{\boldsymbol{\phi}}(\mathbf{z}_k|\mathbf{d}_k)$ to the intractable posterior $p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{d}_k)$. We approximate $p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{d}_k)$ with $q_{\phi}(\mathbf{z}_k|\mathbf{d}_k)$ by minimizing their KL-divergence Joyce (2011):

$$\boldsymbol{\phi}^*, \boldsymbol{\theta}^* = arg\min_{\boldsymbol{\theta},\boldsymbol{\phi}} \mathbb{D}_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}_k|\mathbf{d}_k)\,||\,p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{d}_k)). \tag{11}$$

To derive the optimal value of the parameters $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$, we compute the marginal likelihood of $\mathbf{d}_k$:

$$\log p(\mathbf{d}_k) = \mathbb{D}_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}_k|\mathbf{d}_k)\,||\,p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{d}_k)) + \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}_k|\mathbf{d}_k)}\left[\log\frac{p_{\boldsymbol{\theta}}(\mathbf{z}_k,\mathbf{d}_k)}{q_{\boldsymbol{\phi}}(\mathbf{z}_k|\mathbf{d}_k)}\right], \tag{12}$$

where the first term is the KL-divergence of the approximate distribution and the posterior distribution; and the second term is called the ELBO (Evidence Lower BOund) on the marginal likelihood of the k-th client's dataset.

Since $\log p(\mathbf{d}_k)$ is non-negative, the minimization problem of Eq. (11) can be converted to maximizing the corresponding ELBO. To solve the problem, we change the form of ELBO as:

$$\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}_k|\mathbf{d}_k)}\left[\log\frac{p_{\boldsymbol{\theta}}(\mathbf{z}_k,\mathbf{d}_k)}{q_{\boldsymbol{\phi}}(\mathbf{z}_k|\mathbf{d}_k)}\right] = \underbrace{\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}_k|\mathbf{d}_k)}\left[log\frac{p(\mathbf{z}_k)}{q_{\boldsymbol{\phi}}(\mathbf{z}_k|\mathbf{d}_k)}\right]}_{\text{Encoder}} + \underbrace{\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}_k|\mathbf{d}_k)}[\log p_{\boldsymbol{\theta}}(\mathbf{d}_k|\mathbf{z}_k)]}_{\text{Decoder}}. \tag{13}$$

The above form is a variational encoder-decoder structure: the model $q_{\boldsymbol{\phi}}(\mathbf{z}_k|\mathbf{d}_k)$ can be viewed as a probabilistic encoder that given an observed statistics $\mathbf{d}_k$ it produces a distribution over the possible values of the latent variable $\mathbf{z}_k$; The model $p_{\boldsymbol{\theta}}(\mathbf{s}_k|\mathbf{z}_k)$ can be refered to as a probabilistic decoder that reconstructs the value of $\mathbf{d}_k$ based on the latent variable $\mathbf{z}_k$. According to the theory of variational inference Kingma & Welling (2014), the problem in Eq. (13) can be solved with the SGD method using a nonlinear deep neural network (DNN) to optimize the mean squared error loss function. The overall `FedDAF` model aggregation process is summarized in Algorithm 1.

5 Experiments

In this section, we evaluate the performance of the proposed `FedDAF` method for federated learning.

5.1 Experimental Setup

Implementation: We implement the proposed `FedDAF` algorithm and the considered baselines in PyTorch Paszke et al. (2019). We train the models in a simulated federated learning environment

Algorithm 1: The `FedDAF` model aggregation algorithm.

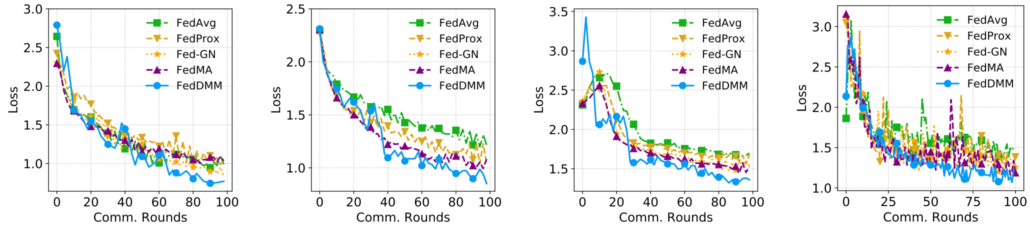
```
Initialize $\mathbf{w}^0$.
for each communication round $t = 0, 1, \ldots, T-1$ do
    $\mathbf{w}_k^{t+1}$ := the model received from client k
    $\mathbf{d_k}$ := $(\hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\sigma}}_k, \hat{\boldsymbol{\beta}}_k, \hat{\gamma}_k)$  // statical info from client k
    // Periodically conduct the following variational inference process:
    repeat
        Inference $\boldsymbol{\kappa}_m$, $\boldsymbol{\zeta}$,, $\boldsymbol{\nu}_m$, $\boldsymbol{\varsigma}_m$, $\boldsymbol{\nu}_m^{'}$ and $\boldsymbol{\varsigma}_m^{'}$ based on encoder $\boldsymbol{\phi}$
        $\mathbf{b}_k$, $\boldsymbol{\lambda}_k$, $\mathbf{c}_k$ := sampling from distributions with Eq. 6, 8, 9
        $\tilde{\mathbf{z}}_m$ := sampling from $\mathcal{N}(\boldsymbol{\nu}_m^{'}, \boldsymbol{\varsigma}_m^{'})$
        $\mathbf{z}_k := \sum_{m=1}^{M} b_{km} \cdot \tilde{\mathbf{z}}_m$
        Recover $\mathbf{z_k}$ to $\mathbf{d_k}$ based on decoder $\boldsymbol{\theta}$ with Eq. 13
    until VAE converge;
    $\mathbf{w}^{t+1} := \sum_{m=1}^{M} \pi_m \sum_{k=1}^{K} b_{km} \cdot c_{km} \cdot \mathbf{w}_k^{t+1}$  // model aggregation
    broadcast $\mathbf{w}^{t+1}$ to all clients
```

consisting of one server and a set of mobile clients with wireless network connections. Unless explicitly specified, the default number of clients is 50, and the learning rate $\beta = 0.01$. We conduct experiments on a GPU-equipped personal computer (CPU: Intel Core i7-8700 3.2GHz, GPU: Nvidia GeForce RTX 2070, Memory: 32GB DDR4 2666MHz, and OS: 64-bit Ubuntu 16.04).

Models and datasets: Our experiments are based on 5 mainstream deep neural network models: ResNet18 He et al. (2016), LeNet Lecun et al. (1998), DenseNet121 Huang et al. (2017), MobileNetV2 Sandler et al. (2018), and BiLSTM.

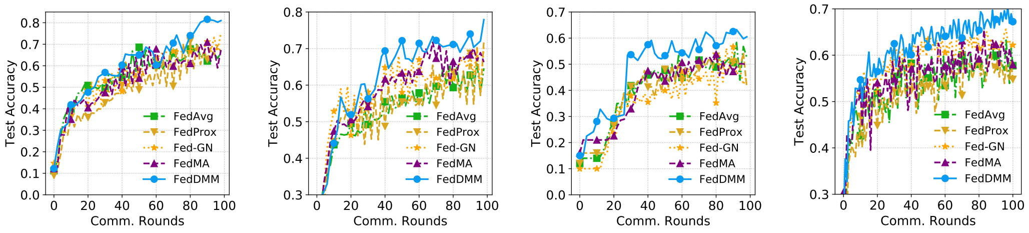
We use 4 real world datasets: MNIST LeCun et al. (2010), Fashion-MNIST Xiao et al. (2017), CIFAR-10 Krizhevsky (2009), and Sentiment140 Go et al. (2009). MNIST is a dataset for hand written digits classification with 60000 samples of 28×28 greyscale image. Fashion-MNIST is an extended version of MNIST for benchmarking machine learning algorithms. CIFAR-10 is a large image dataset with 10 categories, each of which has 6000 samples of size 32×32. Sentiment140 is a natural language process dataset containing 1,600,000 extracted tweets annotated in scale 0 to 4 for sentiment detection.

We generate non-IID data partition according to the work McMahan et al. (2017). For each dataset, we use 80% as training dada to form non-IID local datasets as follows. We sort the data by their labels and divide each class into 200 shards. Each client draw samples from the shards to form a local dataset with probability $pr(x) = \begin{cases} \eta \in [0,1], & \text{if x} \in class_j, \\ \mathcal{N}(0.5, 1), & \text{otherwise.} \end{cases}$ It means that the client draws samples from a particular class j with a fixed probability η, and from other classes based on a Gaussian distribution. The larger η is, the more likely the samples concentrate on a particular class, and the more heterogeneous the datasets are. By default we set $\eta = 0.5$.



(a) (b) (c) (d)

Figure 2: Convergence of different algorithms. (a) ResNet18 on CIFAR10, (b) DenseNet121 on CIFAR10, (c) MobileNetV2 on CIFAR10, (d) BiLSTM on Sent140.



(a) (b) (c) (d)

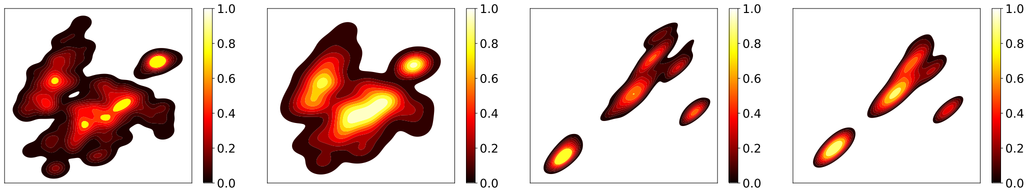
Figure 3: Training efficiency of different algorithms. (a) ResNet18 on CIFAR-10, (b) DenseNet121 on CIFAR-10, (c) MobileNetV2 on CIFAR-10, (d) BiLSTM on Sent140.

5.2 Performance Comparison

We compare the performance of FedDAF with 4 state-of-the-art methods: FedAvg McMahan et al. (2017), FedProx Li et al. (2020a), Fed-GN Hsieh et al. (2020), and FedMA Wang et al. (2020). The results are analyzed as follows.

Convergence: In this experiment, we study the convergence of the compared algorithms by showing the total communication epochs versus train loss. Fig. 2 shows the convergence of different algorithms for different models on different datasets. It is shown that the loss of all algorithms tends to be stable after a number of communication rounds. Clearly, FedDAF has the lowest loss, and converges the fastest among all algorithms.

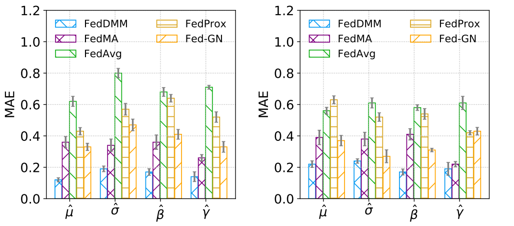
Training Efficiency: In this experiment, we study the test accuracy versus time during the training of a DNN model with federated learning. Fig. 3 shown the results of training different models on different datasets. It is shown that FedDAF trains much faster than the baseline algorithms, and it reaches higher accuracy in a shorter period.



(a) (b) (c) (d)

Figure 4: Visualization of data distribution (only a subset of the original data is illustrated). (a) the original distribution of MNIST, (b) the inferred distribution of MNIST with FedDAF, (c) the original distribution of CIFAR-10, (d) the inferred distribution of CIFAR-10 with FedDAF.

Visualization of Data Distribution: To intuitively illustrate how well the proposed FedDAF can approximate the original data distribution, we use t-SNE van der Maaten & Hinton (2008) to visualize the original distribution of MNIST and CIFAR-10 and the distribution fusion inferred with the proposed VAE. The results are shown in Fig. 4. According to the figure, the inferred distribution fusion looks very close to the original distribution, which implies that the federated server can well approximate the distribution parameters without accessing to local data.



(a) (b)

Figure 5: Comparison of parameter bias. (a) ResNet18 on CIFAR-10, (b) BiLSTM on Sent140.

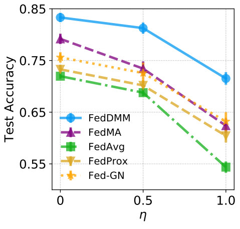


Figure 6: Test accuracy with different heterogeneity η (ResNet18 on CIFAR-10).

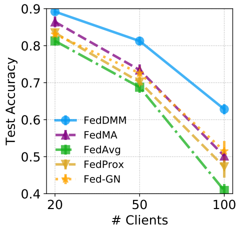


Figure 7: Test accuracy with different number of clients (ResNet18 on CIFAR-10).

Bias of Model Parameters: To show the power of the proposed VAE method for parameter optimization, we calculate the mean absolute error (MAE) of the statistical parameters $(\hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\sigma}}_k, \hat{\boldsymbol{\beta}}_k, \hat{\boldsymbol{\gamma}}_k)$ compared to a centrally-trained model based on global dataset, and the results are illustrated in Fig. 5(a) and Fig. 5(b). It is shown that FedDAF has a much lower bias in the statistical parameters than that of the other algorithms, which means that FedDAF provides a better approximation to the global data distribution.

Table 1: Comparison of average test accuracy on non-IID datasets.

Dataset	CIFAR-10			FMNIST	MNIST	Sent140
Model	ResNet18	DenseNet121	MobileNetV2	LeNet	LeNet	BiLSTM
FedAvg	68.78 (± 0.89)	63.33 (± 0.67)	54.69 (± 3.92)	79.20 (± 1.15)	97.32 (± 0.04)	58.33 (± 2.03)
FedProx	70.18 (± 0.45)	66.85 (± 0.93)	55.03 (± 2.77)	80.03 (± 0.98)	97.55 (± 0.02)	59.73 (± 1.38)
Fed-GN	72.57 (± 0.78)	70.02 (± 1.36)	56.43 (± 1.92)	81.11 (± 0.74)	97.88 (± 0.02)	63.41 (± 1.94)
FedMA	73.43 (± 1.03)	70.13 (± 1.71)	59.61 (± 2.01)	81.02 (± 1.35)	98.06 (± 0.03)	60.86 (± 2.42)
FeSEM	67.78 (± 2.58)	62.65 (± 0.82)	53.82 (± 3.69)	78.18 (± 1.45)	96.24 (± 0.17)	59.57 (± 3.41)
IFCA	73.04 (± 1.45)	70.85 (± 2.03)	58.93 (± 2.45)	80.82 (± 1.29)	97.09 (± 0.11)	60.82 (± 2.74)
FedCluster	72.57 (± 0.78)	68.77 (± 1.38)	58.18 (± 1.22)	79.11 (± 0.74)	97.88 (± 0.02)	63.41 (± 1.94)
FedGroup	74.38 (± 1.92)	71.63 (± 0.74)	59.86 (± 2.09)	81.32 (± 2.07)	97.37 (± 0.61)	63.61 (± 3.26)
FedDAF	**81.26** (± 0.82)	**75.92** (± 1.25)	**62.88** (± 1.21)	**83.16** (± 0.74)	**98.49** (± 0.04)	**67.51** (± 1.71)

Global Model Accuracy: In this experiment, we compare the global model accuracy of different federated parameter aggregation algorithms after training to converge. For thorough comparison, we include 4 clustered and personalized FL algorithms FeSEM Xie et al. (2020), IFCA Ghosh et al. (2020), FedCluster Chen et al. (2020), and FedGroup Duan et al. (2020) as additional baselines. Since clustered and personalized FL methods output multiple models, we show the average results of those models. We repeat each experiment for 20 rounds and show the average performance in Table 1. Comparing the global model accuracy of different federated learning methods, FedDAF significantly outperforms the other algorithms for all DNN models. It outperforms FedMA by 7.83%, 5.79%, and 3.27% for accuracy in ResNet18, DenseNet121, and MobileNetV2 respectively on CIFAR-10; achieves 2.14% improvement in LeNet on F-MNIST; 0.37% improvement in LeNet on MNIST; and 6.65% improvement in BiLSTM on Sent140 accordingly. Compared to FedAvg, the performance improvement of FedDAF is significant, which achieves up to 12.59% higher in DenseNet121 on CIFAR-10. In comparison to clustered/personalized FL, FedDAF outperforms the state-of-the-art method FedGroup by 6.88%, 1.84%, 1.12%, and 3.90% separately in the 4 datasets. In summary, FedDAF achieves the highest accuracy among all baseline algorithms.

Hyperparameter Analysis: We further analyze the influence of two hyperparameters: the heterogeneity of local datasets and the number of clients involved.

The heterogeneity of local datasets is represented by η, the probability that a client tends to sample from a particular class. The more η approaches to 1, the more heterogeneous the local datasets are. Fig. 6 shows the test accuracy under different levels of heterogeneity. As η increases, the test accuracy of all models decreases. FedDAF yields the highest test accuracy and slowest performance drop among all compared algorithms, which shows more robust against η.

Fig. 7 compares the test accuracy of the global model for a different numbers of involved clients. When the number of clients increases from 20 to 100, the accuracy of FedDAF decreases much slower than that of the baselines, and it achieves the highest test accuracy among all compared federated learning algorithms in all cases.

6 Conclusion

Developing efficient model aggregation methods against the performance drop in non-IID data is a key research problem in federated learning. In this paper, we proposed a novel data agnostic fusion called FedDAF to optimize federated learning with data heterogeneity. In the proposed method, each client minimized their local model the same as conventional federated learning, and the server aggregated the local models by allocating the clients' data distributions into several virtual components with different mixture weights. The optimal parameters of the distribution fusion federated learning model were derived by a variational autoencoder (VAE) method. Based on variational inference, an efficient algorithm was proposed to optimize federated learning on non-IID data by solving a probabilistic maximization problem. Extensive experiments showed that FedDAF significantly outperforms the state-of-the-art on a variety of federated learning scenarios.

REFERENCES

Durmus Alp Emre Acar, Yue Zhao, Ramon Matas, Matthew Mattina, Paul Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. In *International Conference on Learning Representations (ICLR'21)*, 2021.

Jimmy Ba, J. Kiros, and Geoffrey E. Hinton. Layer normalization. *ArXiv*, abs/1607.06450, 2016.

Cheng Chen, Ziyi Chen, Yi Zhou, and Bhavya Kailkhura. Fedcluster: Boosting the convergence of federated learning via cluster-cycling. In *IEEE International Conference on Big Data (Big Data'20)*, pp. 5017–5026, 2020.

Tianyi Chen, Georgios Giannakis, Tao Sun, and Wotao Yin. Lag: Lazily aggregated gradient for communication-efficient distributed learning. *Advances in Neural Information Processing Systems (NIPS'18)*, pp. 5050–5060, 2018.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

Moming Duan, Duo Liu, Xinyuan Ji, Renping Liu, Liang Liang, Xianzhang Chen, and Yujuan Tan. Fedgroup: Ternary cosine similarity-based clustered federated learning framework toward high accuracy in heterogeneous data. *CoRR*, abs/2010.06870, 2020.

Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning. In *Advances in Neural Information Processing Systems (NeurIPS'20)*, 2020.

Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision, 2009. URL `http://help.sentiment140.com/home`.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'16)*, pp. 770–778, 2016.

Kevin Hsieh, Amar Phanishayee, Onur Mutlu, and Phillip B. Gibbons. The Non-IID data quagmire of decentralized machine learning. *Proceedings of the 37th International Conference on Machine Learning (ICML'20)*, 2020.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17)*, pp. 2261–2269, 2017.

Sergey Ioffe and Christian Szegedy. Batch normalization: accelerating deep network training by reducing internal covariate shift. *Internation Conference on Machine Learning (ICML'15)*, 1: 448–456, 2015.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations (ICLR'17)*, 2017.

Jinlong Ji, Xuhui Chen, Qianlong Wang, Lixing Yu, and Pan Li. Learning to learn gradient aggregation by gradient descent. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI'19)*, pp. 2614–2620, 2019.

James M. Joyce. *Kullback-Leibler Divergence*, pp. 720–722. 2011.

P. Kairouz, H. McMahan, B. Avent, Aurélien Bellet, Mehdi Bennis, A. Bhagoji, Keith Bonawitz, and et al. Advances and open problems in federated learning. *ArXiv*, abs/1912.04977, 2019.

Marcel Keller, Valerio Pastro, and Dragos Rotaru. Overdrive: Making SPDZ great again. In *Advances in Cryptology (EUROCRYPT'18)*, volume 10822, pp. 158–189, 2018.

Mikhail Khodak, Maria-Florina F Balcan, and Ameet S Talwalkar. Adaptive gradient-based meta-learning methods. In *Advances in Neural Information Processing Systems (NeurIPS'19)*, volume 32, 2019.

Diederik Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR'14)*, 2014.

Jakub Konečnỳ, H. Brendan McMahan, and Daniel Ramage. Federated optimization: Distributed optimization beyond the datacenter. *NIPS Optimization for Machine Learning Workshop 2015*, pp. pp.5, 2015.

Jakub Konecný, H. Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *ArXiv*, abs/1610.02527, 2016.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Ponnambalam Kumaraswamy. A generalized probability density function for double-bounded random processes. *Journal of Hydrology,*, 1980.

Yann Lecun, LÃl'on Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pp. 2278–2324, 1998.

Yann LeCun, Corinna Cortes, and CJ Burges. MNIST handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems (MLSys'20)*, pp. 429–450. 2020a.

Tian Li, Maziar Sanjabi, and Virginia Smith. Fair resource allocation in federated learning. In *International Conference on Learning Representations (ICLR'20)*, 2020b.

Xiaoxiao Li, Meirui JIANG, Xiaofei Zhang, Michael Kamp, and Qi Dou. FedBN: Federated learning on non-IID features via local batch normalization. In *International Conference on Learning Representations (ICLR'21)*, 2021.

Paul Pu Liang, Terrance Liu, Liu Ziyin, Ruslan Salakhutdinov, and Louis-Philippe Morency. Think locally, act globally: Federated learning with local and global representations. *arXiv preprint arXiv:2001.01523*, 2020.

Wei Yang Lim, Nguyen Cong Luong, D. Hoang, Y. Jiao, Ying-Chang Liang, Qiang Yang, D. Niyato, and Chunyan Miao. Federated learning in mobile edge networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 22:2031–2063, 2020.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS'17)*, 54:1273–1282, 2017.

Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. In *Proceedings of the 36th International Conference on Machine Learning (ICML'19)*, volume 97, pp. 4615–4625, 2019.

Eric T. Nalisnick and Padhraic Smyth. Stick-breaking variational autoencoders. In *5th International Conference on Learning Representations (ICLR'17)*, 2017.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, and Zachary DeVito. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS'19)*, pp. 8024–8035, 2019.

Xingchao Peng, Zijun Huang, Yizhe Zhu, and Kate Saenko. Federated adversarial domain adaptation. In *International Conference on Learning Representations (ICLR'20)*, 2020.

M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'18)*, pp. 4510–4520, 2018.

V. Smith, S. Forte, C. Ma, M. Takac, M. I. Jordan, and M. Jaggi. Cocoa: A general framework for communication-efficient distributed optimization. *Journal of Machine Learning Research*, 18 (230):1–47, 2018.

Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. Federated multi-task learning. In *Advances in Neural Information Processing Systems (NIPS'17)*, volume 30, pp. 4424–4434, 2017.

Shizhao Sun, Wei Chen, Jiang Bian, Xiaoguang Liu, and Tie-Yan Liu. Ensemble-compression: A new method for parallel training of deep neural networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-KDD'17)*, pp. 187–202, 2017.

Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. Instance normalization: The missing ingredient for fast stylization. *CoRR*, abs/1607.08022, 2016.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. In *International Conference on Learning Representations (ICLR'20)*, 2020.

Yuxin Wu and Kaiming He. Group normalization. *CoRR*, abs/1803.08494, 2018.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. 2017.

Cong Xie, Sanmi Koyejo, and Indranil Gupta. Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance. In *Proceedings of the 36th International Conference on Machine Learning (ICML'19)*, volume 97, pp. 6893–6901, 2019.

Ming Xie, Guodong Long, Tao Shen, Tianyi Zhou, Xianzhi Wang, and Jing Jiang. Multi-center federated learning. *CoRR*, abs/2005.01026, 2020.

Haibo Yang, Minghong Fang, and Jia Liu. Achieving linear speedup with partial worker participation in non-IID federated learning. In *International Conference on Learning Representations (ICLR'21)*, 2021.

Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Nghia Hoang, and Yasaman Khazaeni. Bayesian nonparametric federated learning of neural networks. In *Proceedings of the 36th International Conference on Machine Learning (ICML'19)*, volume 97, pp. 7252–7261, 2019.

Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and V. Chandra. Federated learning with Non-IID data. *ArXiv*, abs/1806.00582, 2018.

H. Zhu and Y. Jin. Multi-objective evolutionary federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 31(4):1310–1322, 2020.