Personalized Cross-Silo Federated Learning on Non-IID Data

Yutao Huang[1], **Lingyang Chu**[2*], **Zirui Zhou**[2], **Lanjun Wang**[2],
Jiangchuan Liu[1], **Jian Pei**[1], **Yong Zhang**[2]
[1]Simon Fraser University, Burnaby, Canada
[2]Huawei Technologies Canada, Burnaby, Canada
[1]{yutaoh, jcliu, jpei}@{sfu, sfu, cs.sfu}.ca
[2]{lingyang.chu1, zirui.zhou, lanjun.wang, yong.zhang3}@huawei.com

Abstract

Non-IID data present a tough challenge for federated learning. In this paper, we explore a novel idea of facilitating pairwise collaborations between clients with similar data. We propose FedAMP, a new method employing federated attentive message passing to facilitate similar clients to collaborate more. We establish the convergence of FedAMP for both convex and non-convex models, and propose a heuristic method to further improve the performance of FedAMP when clients adopt deep neural networks as personalized models. Our extensive experiments on benchmark data sets demonstrate the superior performance of the proposed methods.

1 Introduction

Federated learning (Yang et al. 2019) facilitates collaborations among a set of clients and preserves their privacy so that the clients can achieve better machine learning performance than individually working alone. The underlying idea is to collectively learn from data from all clients. The initial idea of federated learning starts from aggregating models from clients to achieve a global model so that the global model can be more general and capable. The effectiveness of this global collaboration theme that is not differentiating among all clients highly depends on the data distribution among clients. It works well on IID data, that is, clients are similar to each other in their private data distribution.

In many application scenarios where collaborations among clients are needed to train machine learning models, data are unfortunately not IID. For example, consider the cases of personalized cross-silo federated learning (Kairouz et al. 2019), where there are tens or hundreds of clients and the private data of clients may be different in size, class distributions and even the distribution of each class. Global collaboration without considering individual private data often cannot achieve good performance for individual clients.

Some federated learning methods try to fix the problem by conducting an additional fine-tuning step after a global model is trained (Ben-David et al. 2010; Cortes and Mohri 2014; Mansour et al. 2020; Mansour, Mohri, and Rostamizadeh 2009; Schneider and Vlachos 2020; Wang et al. 2019).While those methods work in some cases, they cannot solve the problem systematically as demonstrated in our experimental results (e.g., data set CIFAR100 in Table 3).

*Lingyang Chu and Yutao Huang contribute equally in this work.

We argue that the fundamental bottleneck in personalized cross-silo federated learning with non-IID data is the misassumption of one global model can fit all clients. Consider the scenario where each client tries to train a model on customers' sentiments on food in a country. Different clients collect data in different countries. Obviously, customers' reviews on food are likely to be related to their cultures, life-styles, and environments. Unlikely there exists a global model universally fitting all countries. Instead, pairwise collaborations among countries that share similarity in culture, life-styles, environments and other factors may be the key to accomplish good performance in personalized cross-silo federated learning with non-IID data.

Carrying the above insight, in this paper, we tackle the challenging personalized cross-silo federated learning problem by a novel *attentive message passing mechanism* that adaptively facilitates the underlying pairwise collaborations between clients by iteratively encouraging similar clients to collaborate more. We make several technical contributions.

We propose a novel method *federated attentive message passing* (FedAMP) whose central idea is the attentive message passing mechanism. FedAMP allows each client to own a local personalized model, but does not use a single global model on the cloud server to conduct collaborations. Instead, it maintains a personalized cloud model on the cloud server for each client, and realizes the attentive message passing mechanism by attentively passing the personalized model of each client as a message to the personalized cloud models with similar model parameters. Moreover, FedAMP updates the personalized cloud model of each client by a weighted convex combination of all the messages it receives. This adaptively facilitates the underlying pairwise collaborations between clients and significantly improves the effectiveness of collaboration.

We prove the convergence of FedAMP for both convex and non-convex personalized models. Furthermore, we propose a heuristic method to further improve the performance of FedAMP on clients using deep neural networks as personalized models. We conduct extensive experiments to demonstrate the superior performance of the proposed methods.

2 Related Works

Personalized federated learning for clients with non-IID data has attracted much attention (Deng, Kamani, and Mahdavi 2020; Fallah, Mokhtari, and Ozdaglar 2020; Kulkarni, Kulkarni, and Pant 2020; Mansour et al. 2020). Particularly, our work is related to global federated learning, local customization and multi-task federated learning.

Global federated learning (Ji et al. 2019; McMahan et al. 2016; Wang et al. 2020; Yurochkin et al. 2019) trains a single global model to minimize an empirical risk function over the union of the data across all clients. When the data is non-IID across different clients, however, it is difficult to converge to a good global model that achieves a good personalized performance on every client (Kairouz et al. 2019; Li et al. 2020; McMahan et al. 2016; Zhao et al. 2018).

Local customization methods (Chen et al. 2018; Fallah, Mokhtari, and Ozdaglar 2020; Jiang et al. 2019; Khodak, Balcan, and Talwalkar 2019; Kulkarni, Kulkarni, and Pant 2020; Mansour et al. 2020; Nichol, Achiam, and Schulman 2018; Schneider and Vlachos 2020; Wang et al. 2019) build a personalized model for each client by customizing a well-trained global model. There are several ways to conduct customization. A practical way to customize a personalized model is local fine-tuning (Ben-David et al. 2010; Cortes and Mohri 2014; Mansour et al. 2020; Mansour, Mohri, and Rostamizadeh 2009; Schneider and Vlachos 2020; Wang et al. 2019), where the global model is fine-tuned using the private data of each client to produce a personalized model for the client. Similarly, meta-learning methods (Chen et al. 2018; Fallah, Mokhtari, and Ozdaglar 2020; Jiang et al. 2019; Khodak, Balcan, and Talwalkar 2019; Kulkarni, Kulkarni, and Pant 2020; Nichol, Achiam, and Schulman 2018) can be extended to customize personalized models by adapting a well-trained global model on the local data of a client (Kairouz et al. 2019). Model mixture methods (Deng, Kamani, and Mahdavi 2020; Hanzely and Richtárik 2020) customize for each client by combining the global model with the client's latent local model. SCAFFOLD (Karimireddy et al. 2019) customizes the gradient updates of personalized models to correct client-drifts between personalized models and a global model.

Most existing local customization methods use a single global model to conduct a global collaboration involving all clients. The global collaboration framework only allows contributions from all clients to a global model and customization of the global model for each client. It does not allow pairwise collaboration among clients with similar data, and thus may meet dramatic difficulty on non-IID data.

Smith *et al.* (Smith et al. 2017) model the pair-wise collaboration relationships between clients by extending distributed multi-task learning to federated learning. They tackle the problem by a primal-dual optimization method that achieves great performance on convex models. At the same time, due to its rigid requirement of strong duality, their method is not applicable when clients adopt deep neural networks as personalized models.

Different from all existing work, our study explores pairwise collaboration among clients. Our method is particularly effective when clients' data are non-IID, and can take the great advantage of similarity among clients.

3 Personalized Federated Learning Problem

In this section, we introduce the personalized federated learning problem that aims to collaboratively train personalized models for a set of clients using the non-IID private data of all clients in a privacy-preserving manner (Kairouz et al. 2019; Zhao et al. 2018).

Consider m clients $C_1, \dots, C_m$ that have the same type of models $\mathcal{M}$ personalized by m different sets of model parameters $\mathbf{w_1}, \dots, \mathbf{w_m}$, respectively. Denote by $\mathcal{M}(\mathbf{w_i})$ and D_i $(1 \leq i \leq m)$ the personalized model and the private training data set of client C_i, respectively. These data sets are non-IID, that is, $D_1, \dots, D_m$ are uniformly sampled from m distinct distributions $P_1, \dots, P_m$, respectively. For each client C_i, denote by $\mathcal{V}_i$ the performance of $\mathcal{M}(\mathbf{w_i})$ on the distribution P_i. Denote by $\mathcal{V}_i^*$ the best performance model $\mathcal{M}$ can achieve on P_i by considering all possible parameter sets.

The *personalized federated learning problem* aims to collaboratively use the private training data sets $D_1, \dots, D_m$ to train the *personalized models* $\mathcal{M}(\mathbf{w_1}), \dots, \mathcal{M}(\mathbf{w_m})$ such that $\mathcal{V}_1, \dots, \mathcal{V}_m$ are close to $\mathcal{V}_1^*, \dots, \mathcal{V}_m^*$, respectively, and no private training data of any clients are exposed to any other clients or any third parties.

To be concrete, denote by $F_i : \mathbb{R}^d \to \mathbb{R}$ the training objective function that maps the model parameter set $\mathbf{w_i} \in \mathbb{R}^d$ to a real valued training loss with respect to the private training data D_i of client C_i. We formulate the personalized federated learning problem as

$$\min_{W} \left\{ \mathcal{G}(W) := \sum_{i=1}^{m} F_i(\mathbf{w_i}) + \lambda \sum_{i<j}^{m} A(\|\mathbf{w_i} - \mathbf{w_j}\|^2) \right\}, \tag{1}$$

where $W = [\mathbf{w_1}, \dots, \mathbf{w_m}]$ is a d-by-m dimensional matrix that collects $\mathbf{w_1}, \dots, \mathbf{w_m}$ as its columns and $\lambda > 0$ is a regularization parameter.

The first term $\sum_{i=1}^{m} F_i(\mathbf{w_i})$ in Eq. (1) is the sum of the training losses of the personalized models of all clients. This term allows each client to separately train its own personalized model using its own private training data. The second term improves the collaboration effectiveness between clients by an attention-inducing function $A(\|\mathbf{w_i} - \mathbf{w_j}\|^2)$ defined as follows.

Definition 1 *$A(\|\mathbf{w_i} - \mathbf{w_j}\|^2)$ is an attention-inducing function of $\mathbf{w_i}$ and $\mathbf{w_j}$ if $A : [0, \infty) \to \mathbb{R}$ is a non-linear function that satisfies the following properties.*

1. *A is increasing and concave on $[0, \infty)$ and $A(0) = 0$;*
2. *A is continuously differentiable on $(0, \infty)$; and*
3. *For the derivative A' of A, $\lim_{t \to 0^+} A'(t)$ is finite.*

The attention-inducing function $A(\|\mathbf{w_i} - \mathbf{w_j}\|^2)$ measures the difference between $\mathbf{w_i}$ and $\mathbf{w_j}$ in a non-linear manner. A typical example of $A(\|\mathbf{w_i} - \mathbf{w_j}\|^2)$ is the negative exponential function $A(\|\mathbf{w_i} - \mathbf{w_j}\|^2) = 1 - e^{-\|\mathbf{w_i} - \mathbf{w_j}\|^2/\sigma}$ with a hyperparameter σ. Another example is the smoothly clipped absolute deviation function (Fan and Li 2001). One more example is the minimax concave penalty function (Zhang 2010). We adopt the widely-used negative exponential function for our method in this paper.

As to be illustrated in the next section, our novel use of the attention-inducing function realizes an attentive message passing mechanism that adaptively facilitates collaborations between clients by iteratively encouraging similar clients to collaborate more with each other. The pairwise collaborations boost the performance in personalized federated learning dramatically.

4 Federated Attentive Message Passing

In this section, we first propose a general method to tackle the optimization problem in Eq. (1) without considering privacy preservation for clients. Then, we implement the general method by a personalized federated learning method, *federated attentive message passing* (FedAMP), which collaboratively trains the personalized models of all clients and preserves their data privacy. Last, we explain why FedAMP can adaptively facilitate collaborations between clients and significantly improve the performance of the personalized models.

A General Method

Denote by $\mathcal{F}(W) := \sum_{i=1}^{m} F_i(\mathbf{w_i})$ and $\mathcal{A}(W) := \sum_{i<j}^{m} A(\|\mathbf{w_i} - \mathbf{w_j}\|^2)$ the first and the second terms of $\mathcal{G}(W)$, respectively. We can rewrite the optimization problem in Eq. (1) to

$$\min_{W} \left\{ \mathcal{G}(W) := \mathcal{F}(W) + \lambda \mathcal{A}(W) \right\}. \tag{2}$$

Based on the framework of incremental-type optimization (Bertsekas 2011), we develop a general method to iteratively optimize $\mathcal{G}(W)$ by alternatively optimizing $\mathcal{A}(W)$ and $\mathcal{F}(W)$ until convergence. In the k-th iteration, we first optimize $\mathcal{A}(W)$ by applying a gradient descent step to compute an intermediate d-by-m dimensional matrix

$$U^k = W^{k-1} - \alpha_k \nabla \mathcal{A}(W^{k-1}), \tag{3}$$

where $\alpha_k > 0$ is the step size of gradient descent, and W^{k-1} denotes the matrix W after the $(k-1)$-th iteration. Then, we use U^k as the prox-center and apply a proximal point step (Rockafellar 1976) to optimize $\mathcal{F}(W)$ by computing

$$W^k = \arg\min_{W} \ \mathcal{F}(W) + \frac{\lambda}{2\alpha_k} \|W - U^k\|^2. \tag{4}$$

This iterative process continues until a preset maximum number of iterations K is reached. As illustrated later in Section 5, we analyze the non-asymptotic convergence of the general method, and prove that it converges to an optimal solution when $\mathcal{G}(W)$ is a convex function, and to a stationary point when $\mathcal{G}(W)$ is non-convex.

FedAMP

The general method introduced above can be easily implemented by merging all clients' private training data together as the training data. To perform personalized federated learning without infringing the data privacy of the clients, we develop FedAMP to implement the optimization steps of the general method in a client-server framework by maintaining a personalized cloud model for each client on a cloud server, and passing weighted model-aggregation messages between personalized models and personalized cloud models.

Following the optimization steps of the general method, FedAMP first optimizes $\mathcal{A}(W)$ and implements the optimization step in Eq. (3) by computing the d-by-m dimensional matrix U^k on the cloud server.

Let $U^k = [\mathbf{u_1^k}, \ldots, \mathbf{u_m^k}]$, where $\mathbf{u_1^k}, \ldots, \mathbf{u_m^k}$ are the d-dimensional columns of U^k. Since $\mathcal{A}(W) := \sum_{i<j}^{m} A(\|\mathbf{w_i} - \mathbf{w_j}\|^2)$ and $A(\|\mathbf{w_i} - \mathbf{w_j}\|^2)$ is an attention inducing function, the i-th column $\mathbf{u_i^k}$ of matrix U^k computed

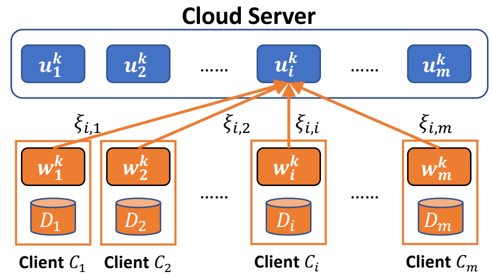


Figure 1: The message passing mechanism of FedAMP.

in Eq. (3) can be rewritten into a linear combination of the model parameter sets $\mathbf{w_1^{k-1}}, \ldots, \mathbf{w_m^{k-1}}$ as follows.

$$\begin{aligned}
\mathbf{u_i^k} =& \left(1 - \alpha_k \sum_{j \neq i}^{m} A'\left(\|\mathbf{w_i^{k-1}} - \mathbf{w_j^{k-1}}\|^2\right)\right) \cdot \mathbf{w_i^{k-1}} \\
&+ \alpha_k \sum_{j \neq i}^{m} A'\left(\|\mathbf{w_i^{k-1}} - \mathbf{w_j^{k-1}}\|^2\right) \cdot \mathbf{w_j^{k-1}} \\
=& \xi_{i,1}\mathbf{w_1^{k-1}} + \cdots + \xi_{i,m}\mathbf{w_m^{k-1}},
\end{aligned} \tag{5}$$

where $A'(\|\mathbf{w_i} - \mathbf{w_j}\|^2)$ is the derivative of $A(\|\mathbf{w_i} - \mathbf{w_j}\|^2)$ and $\xi_{i,1}, \ldots, \xi_{i,m}$ are the linear combination weights of the model parameter sets $\mathbf{w_1^{k-1}}, \ldots, \mathbf{w_m^{k-1}}$, respectively.

Often a small value is chosen as the step size α_k of gradient descent so that all the linear combination weights $\xi_{i,1}, \ldots, \xi_{i,m}$ are non-negative. Since $\xi_{i,1} + \cdots + \xi_{i,m} = 1$, $\mathbf{u_i^k}$ is actually a convex combination of the model parameter sets $\mathbf{w_1^{k-1}}, \ldots, \mathbf{w_m^{k-1}}$ of the personalized models of the clients.

As illustrated in Figure 1, the convex combination $\mathbf{u_i^k}$ can be modeled a *message passing mechanism* as follows. We treat $\mathbf{u_i^k}$ as the model parameter set of the *personalized cloud model* of client C_i and also a model aggregation that aggregates $\mathbf{w_1^{k-1}}, \ldots, \mathbf{w_m^{k-1}}$. Correspondingly, we can treat $\mathbf{w_1^{k-1}}, \ldots, \mathbf{w_m^{k-1}}$ as *model-aggregation messages* that are passed from all clients to client C_i to conduct the model aggregation and produce $\mathbf{u_i^k}$ at the cloud server.

The above message passing mechanism is the key step for FedAMP to perform inter-client collaboration. This mechanism solely depends on the model parameter sets $\mathbf{w_1^{k-1}}, \ldots, \mathbf{w_m^{k-1}}$, thus the cloud server can collect $\mathbf{w_1^{k-1}}, \ldots, \mathbf{w_m^{k-1}}$ from the clients and conduct the message passing mechanism to optimize $\mathcal{A}(W)$ without infringing the data privacy of all the clients.

After optimizing $\mathcal{A}(W)$ on the cloud server, FedAMP then optimizes $\mathcal{F}(W)$ and implements the optimization step in Eq. (4) by computing independently columns $\mathbf{w_1^k}, \ldots, \mathbf{w_m^k}$ of W^k for clients $C_1, \ldots, C_m$, respectively. Recall that $\mathbf{w_i^k}$ is the model parameter set of the personalized model owned by client C_i. Following Eq. (4), we compute $\mathbf{w_i^k}$ locally on C_i by

$$\mathbf{w_i^k} = \arg\min_{\mathbf{w} \in \mathbb{R}^d} F_i(\mathbf{w}) + \frac{\lambda}{2\alpha_k} \|\mathbf{w} - \mathbf{u_i^k}\|^2, \tag{6}$$

Here, we only use the private training data set D_i of client

Algorithm 1: FedAMP

Input: m clients, each holds a set of private training data and a personalized model to train.
Output: The trained model parameter sets $\mathbf{w_1^K}, \dots, \mathbf{w_m^K}$ and $\mathbf{u_1^K}, \dots, \mathbf{u_m^K}$.
Randomly initialize $\mathbf{w_1^0}, \dots, \mathbf{w_m^0}$ on the clients.
for $k = 1, 2, \dots, K$ **do**
  Optimize $\mathcal{A}(W)$: cloud server collects $\mathbf{w_1^{k-1}}, \dots, \mathbf{w_m^{k-1}}$ from the clients to compute $\mathbf{u_1^k}, \dots, \mathbf{u_m^k}$ by Eq. (5).
  Optimize $\mathcal{F}(W)$: each client C_i requests $\mathbf{u_i^k}$ from the cloud server to compute $\mathbf{w_i^k}$ by Eq. (6).
end

C_i to perform personalized training on model $\mathcal{M}(\mathbf{w_i})$ and, at the same time, consider the inter-client collaboration information carried by the personalized cloud model $\mathcal{M}(\mathbf{u_i^k})$ by requiring $\mathbf{w_i^k}$ and $\mathbf{u_i^k}$ to be close to each other.

Since Eq. (6) only uses $F_i(\mathbf{w})$ and $\mathbf{u_i^k}$, where $F_i(\mathbf{w})$ is determined by the private training data D_i of client C_i, C_i can request its own model parameter set $\mathbf{u_i^k}$ from the cloud server and compute $\mathbf{w_i^k}$ locally without exposing its private training data D_i to any other clients or the cloud server. Furthermore, since $\mathbf{u_i^k}$ is a convex combination of $\mathbf{w_1^k}, \dots, \mathbf{w_m^k}$, a client C_j cannot infer the personalized models of any other clients or the private data of any other clients.

Algorithm 1 summarizes the pseudocode. FedAMP implements the optimization steps of the general method in a client-server framework, that is, iteratively optimizing $\mathcal{G}(W)$ by alternatively optimizing $\mathcal{A}(W)$ and $\mathcal{F}(W)$ until a preset maximum number of iterations K is reached. The non-asymptotic convergence of FedAMP is exactly the same as the general method.

Collaboration in FedAMP

FedAMP adaptively facilitates collaborations between similar clients, since the attentive message passing mechanism iteratively encourages similar clients to collaborate more with each other during the personalized federated learning process.

To analyze the attentive message passing mechanism of FedAMP, we revisit the weights $\xi_{i,1}, \dots, \xi_{i,m}$ of the convex combination in Eq. (5), where the weight

$$\xi_{i,j} = \alpha_k A' \left(\|\mathbf{w_i^{k-1}} - \mathbf{w_j^{k-1}}\|^2 \right), (i \neq j) \qquad (7)$$

is the contribution of message $\mathbf{w_j^{k-1}}$ sent from client C_j to the aggregated model parameter set $\mathbf{u_i^k}$ of the personalized cloud model owned by client C_i. $\xi_{i,i} = 1 - \sum_{j \neq i}^{m} \xi_{i,j}$ is simply a self-attention weight that specifies the proportion of the model parameter set $\mathbf{w_i^{k-1}}$ of client C_i's personalized model in its own personalized cloud model.

Due to Definition 1, A is an increasing and concave function on $[0, \infty)$. Thus, the derivative A' of A is a non-negative and non-increasing function on $(0, \infty)$. Therefore, function $A'(\|\mathbf{w_i^{k-1}} - \mathbf{w_j^{k-1}}\|^2)$ is a *similarity function* that measures the similarity between $\mathbf{w_i^{k-1}}$ and $\mathbf{w_j^{k-1}}$, such that their similarity is high if they have a small Euclidean distance.

From Eq. (7), if the model parameters $\mathbf{w_i^{k-1}}$ and $\mathbf{w_j^{k-1}}$ are similar with each other, they contribute more to the model parameters $\mathbf{u_j^k}$ and $\mathbf{u_i^k}$ of clients C_j and C_i, respectively. This further makes $\mathbf{u_i^k}$ and $\mathbf{u_j^k}$ more similar to each other. Since the optimization step in Eq. (6) forces $\mathbf{w_i^k}$ and $\mathbf{w_j^k}$ to be close to $\mathbf{u_i^k}$ and $\mathbf{u_j^k}$, respectively, $\mathbf{w_i^k}$ and $\mathbf{w_j^k}$ are more similar to each other as well.

In summary, FedAMP builds a positive feedback loop that iteratively encourages clients with similar model parameters to have stronger collaborations, and adaptively and implicitly groups similar clients together to conduct more effective collaborations.

5 Convergence Analysis of FedAMP

In this section, we analyze the convergence of FedAMP when $\mathcal{G}$ is convex or non-convex under suitable conditions. To begin with, similar to the analysis of many incremental and stochastic optimization algorithms (Bertsekas 2011; Nemirovski et al. 2009), we make the following assumption.

Assumption 1 *There exists a constant $B > 0$ such that $\max\{\|Y\| : Y \in \partial\mathcal{F}(W^k)\} \leq B$ and $\|\nabla\mathcal{A}(W^k)\| \leq B/\lambda$ hold for every $k \geq 0$, where $\partial\mathcal{F}$ is the subdifferential of $\mathcal{F}$ and $\|\cdot\|$ is the Frobenius norm.*

For our problem in Eq. (1), Assumption 1 naturally holds if both $\mathcal{F}(W)$ and $\mathcal{A}(W)$ are locally Lipschitz continuous and $\|W^k\|$ is bounded by a constant for all $k \geq 0$.

Now, we provide the guarantee on convergence for FedAMP when both $\mathcal{F}(W)$ and $\mathcal{A}(W)$ are convex functions.

Theorem 1 *Under Assumption 1 and assuming functions $\mathcal{F}(W)$ and $\mathcal{A}(W)$ in Eq. (1) are convex, if $\alpha_1 = \cdots = \alpha_K = \lambda/\sqrt{K}$ for some $K \geq 0$, then the sequence $W^0, \dots, W^K$ generated by Algorithm 1 satisfies*

$$\min_{0 \leq k \leq K} \mathcal{G}(W^k) \leq \mathcal{G}^* + \frac{\|W^0 - W^*\|^2 + 5B^2}{\sqrt{K}},$$

where W^* is an optimal solution of Eq. (1) and $\mathcal{G}^* = \mathcal{G}(W^*)$. Moreover, if α_k satisfies $\sum_{k=1}^{\infty} \alpha_k = \infty$ and $\sum_{k=1}^{\infty} \alpha_k^2 < \infty$, then

$$\liminf_{k \to \infty} \mathcal{G}(W^k) = \mathcal{G}^*.$$

Theorem 1 implies that for any $\epsilon > 0$, FedAMP needs at most $\mathcal{O}(\epsilon^{-2})$ iterations to find an ϵ-optimal solution $\widetilde{W}$ of Eq. (1) such that $\mathcal{G}(\widetilde{W}) - \mathcal{G}^* \leq \epsilon$. It also establishes the global convergence of FedAMP to an optimal solution of Eq. (1) when $\mathcal{G}$ is convex. The proof of Theorem 1 is provided in Appendix A (Huang et al. 2020).

Next, we provide the convergence guarantee of FedAMP when $\mathcal{G}(W)$ is a smooth and non-convex function.

Theorem 2 *Under Assumption 1 and assuming functions $\mathcal{F}(W)$ and $\mathcal{A}(W)$ in Eq. (1) are continuously differentiable and the gradients $\nabla\mathcal{F}(W)$ and $\nabla\mathcal{A}(W)$ are Lipschitz continuous with modulus L, if $\alpha_1 = \cdots = \alpha_K = \lambda/\sqrt{K}$, then* *the sequence $W^0, \dots, W^K$ generated by Algorithm 1 satisfies*

$$\min_{0\le k\le K} \|\nabla \mathcal{G}(W^k)\|^2 \le \frac{18(\mathcal{G}(W^0) - \mathcal{G}^* + 20LB^2)}{\sqrt{K}} + \mathcal{O}\left(\frac{1}{K}\right)$$

where W^* and $\mathcal{G}^*$ are the same as in Theorem 1. Moreover, if α_k satisfies $\sum_{k=1}^{\infty} \alpha_k = \infty$ and $\sum_{k=1}^{\infty} \alpha_k^2 < \infty$, then

$$\liminf_{k\to\infty} \|\nabla \mathcal{G}(W^k)\| = 0.$$

Theorem 2 implies that for any $\epsilon > 0$, FedAMP needs at most $\mathcal{O}(\epsilon^{-4})$ iterations to find an ϵ-approximate stationary point $\widetilde{W}$ of Eq. (1) such that $\|\nabla \mathcal{G}(\widetilde{W})\| \le \epsilon$. It also establishes the global convergence of FedAMP to a stationary point of Eq. (1) when $\mathcal{G}$ is smooth and non-convex. The proof of Theorem 2 is in Appendix B (Huang et al. 2020).

6 HeurFedAMP: Heuristic Improvement of FedAMP on Deep Neural Networks

In this section, we tackle the challenge in the message passing mechanism when deep neural networks are used by clients, and propose a heuristic improvement of FedAMP.

As illustrated in Section 4, the effectiveness of the attentive message passing mechanism of FedAMP largely depends on the weights $\xi_{i,1}, \dots, \xi_{i,m}$ of the model aggregation messages. These message weights are determined by the similarity function $A'(\|\mathbf{w_i} - \mathbf{w_j}\|^2)$ that measures the similarity between the model parameter sets $\mathbf{w_i}$ and $\mathbf{w_j}$ based on their Euclidean distance $\|\mathbf{w_i} - \mathbf{w_j}\|$.

When the dimensionalities of $\mathbf{w_i}$ and $\mathbf{w_j}$ are small, Euclidean distance is a good measurement to evaluate their difference. In this case, the similarity function $A'(\|\mathbf{w_i} - \mathbf{w_j}\|^2)$ works well in evaluating the similarity between $\mathbf{w_i}$ and $\mathbf{w_j}$. However, when clients adopt deep neural networks as their personalized models, each personalized model involves a large number of parameters, which means the dimensionalities of both $\mathbf{w_i}$ and $\mathbf{w_j}$ are high. In this case, Euclidean distance may not be effective in evaluating the difference between $\mathbf{w_i}$ and $\mathbf{w_j}$ anymore due to the curse of dimensionality (Verleysen and François 2005; Wikipedia. 2020). Consequently, the message weights produced by $A'(\|\mathbf{w_i} - \mathbf{w_j}\|^2)$ may not be an effective attentive message passing mechanism. Thus, we need a better way to produce the message weights instead of using $A'(\|\mathbf{w_i} - \mathbf{w_j}\|^2)$.

To tackle the challenge, we propose HeurFedAMP, a heuristic revision of FedAMP when clients use deep neural networks. The key idea of HeurFedAMP is to heuristically compute the message weights in a different way that works well with the high-dimensional model parameters of deep neural networks. Specifically, HeurFedAMP follows the optimization steps of FedAMP exactly, except that, when computing message weights $\xi_{i,1}, \dots, \xi_{i,m}$ in the k-th iteration, HeurFedAMP first treats weight $\xi_{i,i}$ as a self-attention hyper-parameter that controls the proportion of the message $\mathbf{w_i^{k-1}}$ sent from client C_i to its own personalized cloud model, and then computes the weight of the message passed from a client C_j to client C_i by

$$\xi_{i,j} = \frac{e^{\sigma \cos(\mathbf{w_i^{k-1}}, \mathbf{w_j^{k-1}})}}{\sum_{h\neq i}^{m} e^{\sigma \cos(\mathbf{w_i^{k-1}}, \mathbf{w_h^{k-1}})}} \cdot (1 - \xi_{i,i}), \qquad (8)$$

where σ is a scaling hyper-parameter and $\cos(\mathbf{w_i^{k-1}}, \mathbf{w_j^{k-1}})$ is the cosine similarity between $\mathbf{w_i^{k-1}}$ and $\mathbf{w_j^{k-1}}$.

All the weights $\xi_{i,1}, \dots, \xi_{i,m}$ computed by HeurFedAMP are non-negative and sum to 1. Applying the weights computed by HeurFedAMP to Eq. (5), the model parameter set $\mathbf{u_i^k}$ of the personalized cloud model of client C_i is still a convex combination of all the messages that it receives.

Furthermore, according to from Eq. (8), if the model parameter sets $\mathbf{w_i^{k-1}}$ and $\mathbf{w_j^{k-1}}$ of two clients have a large cosine similarity $\cos(\mathbf{w_i^{k-1}}, \mathbf{w_j^{k-1}})$, their messages have large weights and contribute more to the personalized cloud models of each other. In other words, HeurFedAMP builds a positive feedback loop similar to that of FedAMP to realize the attentive message passing mechanism.

As to be demonstrated in Section 7, HeurFedAMP improves the performance of FedAMP when clients adopt deep neural networks as personalized models, because cosine similarity is well-known to be more robust in evaluating similarity between high dimensional model parameters than Euclidean distance.

7 Experiments

In this section, we evaluate the performance of FedAMP and HeurFedAMP and compare them with the state-of-the-art personalized federated learning algorithms, including SCAFFOLD (Karimireddy et al. 2019), APFL (Deng, Kamani, and Mahdavi 2020), FedAvg-FT and FedProx-FT (Wang et al. 2019). FedAvg-FT and FedProx-FT are two local fine-tuning methods (Wang et al. 2019) that obtain personalized models by fine-tuning the global models produced by the classic global federated learning methods FedAvg (McMahan et al. 2016) and FedProx (Li et al. 2020), respectively. To make our experiments more comprehensive, we also report the performance of FedAvg, FedProx and a naive separate training method named *Separate* that independently trains the personalized model of each client without collaboration between clients.

The performance of all the methods is evaluated by the *best mean testing accuracy* (BMTA) in percentage, where the *mean testing accuracy* is the average of the testing accuracies on all clients, and BMTA is the highest mean testing accuracy achieved by a method during all the communication rounds of training.

All the methods are implemented in PyTorch 1.3 running on Dell Alienware with Intel(R) Core(TM) i9-9980XE CPU, 128G memory, NVIDIA 1080Ti, and Ubuntu 16.04.

Settings of Data Sets

We use four public benchmark data sets, MNIST (LeCun, Cortes, and Burges 2010), FMNIST (Fashion-MNIST) (Xiao, Rasul, and Vollgraf 2017), EMNIST (Extended-MNIST) (Cohen et al. 2017) and CIFAR100 (Krizhevsky and Hinton 2009).

For each of the data sets, we apply three different data settings: 1) an IID data setting (McMahan et al. 2016) that uniformly distributes data across different clients; 2) a pathological non-IID data setting (McMahan et al. 2016) that partitions the data set in a non-IID manner such that each client contains two classes of samples and there is no group-wise similarities between the private data of clients; and 3) a practical non-IID data setting that first partitions clients into groups, and then assigns data samples to clients in such a way that the clients in the same group have similar data distributions, the clients in different groups have different data distributions, every client has data from all classes, and the number of samples per client is different for different groups.

Comparing with the pathological non-IID data setting, the practical non-IID data setting is closer to reality, since in practice each company participating in a personalized federated learning process often has data from most of the classes, and it is common that a subgroup of companies may have similar data distributions that are different from the data owned by companies outside the subgroup.

Let us take EMNIST as an example to show how we apply the practical non-IID data setting. First, we set up 62 clients numbered as clients $0, 1, \ldots, 61$ and divide them into three groups. Then, we assign the samples to the clients such that 80% of the data of every client are uniformly sampled from a set of dominating classes, and 20% of the data are uniformly sampled from the rest of the classes. Specifically, the first group consists of clients 0-9, where each client has 1000 training samples from the dominating classes with digit labels from '0' to '9'. The second group consists of clients 10-35, where each client has 700 training samples from the dominating classes of upper-case letters from 'A' to 'Z'. The third group consists of clients 36-61, where each client has 400 training samples from the dominating classes of lower-case letters from 'a' to 'z'. Every client has 100 testing samples with the same distribution as its training data.

Limited by space, we only report the most important experimental results in the rest of this section. Please see Appendix C (Huang et al. 2020) for the details of the practical non-IID data setting on MNIST, FMNIST and CIFAR100, the implementation details and the hyperparameter settings of all the methods, and also more extensive results about the convergence and robustness of the proposed methods.

Results on the IID Data Setting

Table 1 shows the BMTA of all methods being compared under the IID data setting. The performance of Separate is a good baseline to indicate the needs of collaboration on classifying the data sets, since Separate does not conduct collaboration at all. Separate achieves a performance comparable with all the other methods on the easy data set MNIST. However, on the more challenging data sets FMNIST, EMNIST and CIFAR100, the performance of Separate is significantly behind that of the others due to the lack of collaborations between clients.

The global federated learning methods FedAvg and FedProx achieve the best performance most of the time on IID data, because the clients are similar to each other and the global model fits every client well. Differentiating pairwise collaborations between different clients are not needed on IID data. APFL achieves a performance comparable with FedAvg and FedProx on all data sets, because it degenerates to FedAvg under the IID data setting (Deng, Kamani, and Mahdavi 2020). For this reason, under the IID data setting, we consider APFL a global federated learning method instead of a personalized federated learning method.

The personalized federated learning methods FedAvg-FT, FedProx-FT and SCAFFOLD do not perform as well as FedAvg and FedProx under the IID data setting. Although they achieve a performance comparable to FedAvg and FedProx on MNIST, their performances on the more challenging data sets FMNIST, EMNIST and CIFAR100 are clearly inferior to FedAvg and FedProx. The local fine-tuning steps of FedAvg-FT and FedProx-FT are prone to over-fitting, and the rigid customization on the gradient updates of SCAFFOLD limits its flexibility to fit IID data well.

FedAMP and HeurFedAMP perform much better than FedAvg-FT, FedProx-FT and SCAFFOLD under the IID data setting. The personalized models of clients are similar to each other under the IID data setting, thus the attentive message passing mechanism assigns comparable weights to all messages, which accomplishes a global collaboration among all clients similar to that of FedAvg and FedAMP in effect. FedAMP and HeurFedAMP achieve the best performance among all the personalized federated learning methods on all data sets, and also perform comparably well as FedAvg and FedProx on MNIST, FMNIST and EMNIST.

Table 1: BMTA for the IID data setting.

Methods	MNIST	FMNIST	EMNIST	CIFAR100
Separate	99.27	81.66	54.41	9.82
FedAvg	99.31	91.94	74.38	49.59
FedProx	98.81	90.19	73.14	46.50
FedAvg-FT	98.98	90.17	70.53	35.07
FedProx-FT	98.72	89.02	69.49	40.77
SCAFFOLD	98.89	89.04	72.51	43.06
APFL	98.93	91.03	73.95	49.02
FedAMP	99.22	92.05	74.07	45.68
HeurFedAMP	99.28	91.80	74.07	45.88

Table 2: BMTA for the pathological non-IID data setting.

Methods	MNIST	FMNIST	EMNIST	CIFAR100
Separate	98.73	97.67	99.15	92.67
FedAvg	98.39	77.88	19.44	2.70
FedProx	97.15	83.80	48.81	2.81
FedAvg-FT	99.66	98.07	99.24	95.00
FedProx-FT	99.63	98.00	99.27	94.36
SCAFFOLD	99.34	94.58	98.75	2.04
APFL	98.24	97.44	98.90	52.11
FedAMP	99.53	97.95	99.27	94.87
HeurFedAMP	99.38	98.17	99.26	94.74

Results on the Pathological Non-IID Data Setting

Table 2 shows the BMTA of all the methods under the pathological non-IID data setting. This data setting is pathological because each client contains only two classes of samples, which largely simplifies the classification task on every client (McMahan et al. 2016). The simplicity of client tasks is clearly indicated by the high performance of Separate on all the data sets.

However, the pathological non-IID data setting is not easy for the global federated learning methods. The performance of FedAvg and FedProx degenerates a lot on FMNIST and EMNIST, because taking the global aggregation of all personalized models trained on the non-IID data of different

Table 3: BMTA for the practical non-IID data setting.

Methods	MNIST	FMNIST	EMNIST	CIFAR100
Separate	86.30	86.73	61.78	39.99
FedAvg	81.82	79.50	72.27	35.21
FedProx	81.46	78.71	70.55	37.31
FedAvg-FT	91.79	89.73	78.93	49.00
FedProx-FT	94.10	87.51	77.31	50.24
SCAFFOLD	**98.50**	40.20	77.98	21.29
APFL	85.05	84.08	59.07	16.45
FedAMP	97.59	90.97	81.22	53.04
HeurFedAMP	97.36	**91.37**	**81.47**	**53.27**

clients introduces significant unstableness to the gradient-based optimization process (Zhang et al. 2020).

On the most challenging CIFAR100 data set, the unstableness catastrophically destroys the performance of the global models produced by FedAvg and FedProx, and also significantly damages the performance of SCAFFOLD and APFL because the global models are destroyed such that the customized gradient updates of SCAFFOLD and the model mixtures conducted by APFL can hardly tune it up.

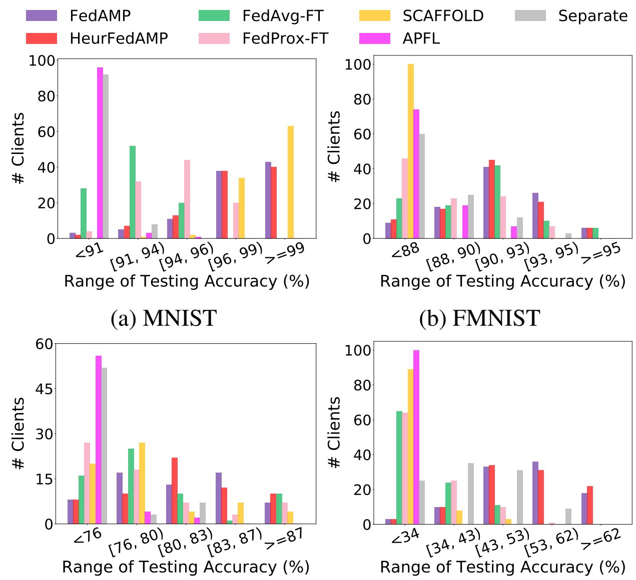
The other personalized federated learning methods FedAvg-FT, FedProx-FT, FedAMP and HeurFedAMP achieve comparably good performance on all data sets. FedAvg-FT and FedProx-FT achieve good performance by taking many fine-tuning steps to tune the poor global models back to normal. The good performance of FedAMP and HeurFedAMP is achieved by adaptively facilitating pairwise collaborations between clients without using a single global model. Since the personalized cloud models of FedAMP and HeurFedAMP only aggregate similar personalized models of clients, they stably converge without suffering from the unstableness caused by the global aggregation of different personalized models.

Results on the Practical Non-IID Data Setting

Table 3 evaluates all methods in BMTA under the practical non-IID data setting. FedAMP and HeurFedAMP perform comparably well as SCAFFOLD on MNIST, and they significantly outperform all other methods on FMNIST, EMNIST and CIFAR100.

To evaluate the personalization performance of all methods in detail, we analyze the testing accuracy of the personalized model owned by each client (Figure 2). Both FedAMP and HeurFedAMP have more clients with higher testing accuracy on FMNIST, EMNIST and CIFAR100. We also conduct Wilcoxon signed-rank test (Wilcoxon 1992) to compare FedAMP/HeurFedAMP against the other methods on FMNIST, EMNIST and CIFAR100, a pair on a data set at a time. In all those tests, the p-values are all less than 10^{-4} and thus the non-hypotheses are all rejected. FedAMP and HeurFedAMP outperform the other methods in testing accuracies of individual clients with statistical significance.

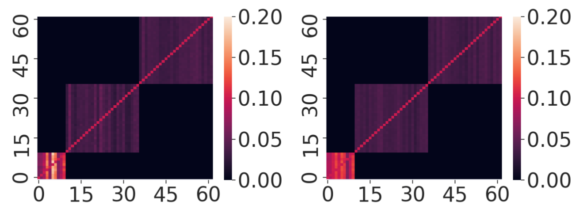
The superior performance of FedAMP and HeurFedAMP is contributed by the attentive message passing mechanism that adaptively facilitates the underlying pair-wise collaborations between clients. Figure 3 the visualizes the collaboration weights $\xi_{i,j}$ computed by FedAMP and HeurFedAMP. The pair-wise collaborations between clients



(a) MNIST (b) FMNIST

(c) EMNIST (d) CIFAR100

Figure 2: The distribution of the testing accuracy of all clients under the practical non-IID data setting.



(a) FedAMP (b) HeurFedAMP

Figure 3: The visualization of the collaboration weights $\xi_{i,j}$ computed by FedAMP and HeurFedAMP on EMNIST under the practical non-IID data setting. X-axis and y-axis show the IDs of clients.

are accurately captured by the three blocks in the matrix, where the three ground-truth collaboration groups are clients 0-9, 10-35 and 36-61. The other methods, however, are not able to form those collaboration groups because using a single global model cannot describe the numerate pairwise collaboration relationships between clients when the data is non-IID across different clients.

8 Conclusions

In this paper, we tackle the challenging problem of personalized cross-silo federated learning and develop FedAMP and HeurFedAMP that introduce a novel attentive message passing mechanism to significantly facilitate the collaboration effectiveness between clients without infringing their data privacy. We analyze how the attentive message passing mechanism iteratively enables similar clients to have stronger collaboration than clients with dissimilar models, and empirically demonstrate that this mechanism significantly improves the learning performance.

Acknowledgements

Yutao Huang's, Jiangchuan Liu's and Jian Pei's research is supported in part by the NSERC Discovery Grant program. All opinions, findings, conclusions and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agencies. Most of the work of the author Zirui Zhou was done when he was affiliated with the Department of Mathematics of Hong Kong Baptist University (HKBU) and was supported in part by an HKBU Start-up Grant.

Ethics Statement

The ever-growing regulations and laws on protecting data privacy, such as the General Data Protection Regulation (GDPR 2016) of Europe, strictly restricts user data transmission between different sources. The restrictions on data transmission have become one of the biggest challenges for many data-intensive machine learning tasks. To tackle this challenge, we propose FedAMP and HeurFedAMP to securely and efficiently train a high performance AI model by legally using the private data held by multiple data owners without infringing the data privacy of any data owner.

References

Ben-David, S.; Blitzer, J.; Crammer, K.; Kulesza, A.; Pereira, F.; and Vaughan, J. W. 2010. A theory of learning from different domains. *Machine Learning* 79(1-2): 151–175.

Bertsekas, D. P. 2011. Incremental gradient, subgradient, and proximal methods for convex optimization: A survey. *Optimization for Machine Learning* 2010(1-38): 3.

Chen, F.; Dong, Z.; Li, Z.; and He, X. 2018. Federated meta-learning for recommendation. *arXiv preprint arXiv:1802.07876* .

Cohen, G.; Afshar, S.; Tapson, J.; and Van Schaik, A. 2017. EMNIST: Extending MNIST to handwritten letters. In *IEEE International Joint Conference on Neural Networks*, 2921–2926.

Cortes, C.; and Mohri, M. 2014. Domain adaptation and sample bias correction theory and algorithm for regression. *Theoretical Computer Science* 519: 103–126.

Deng, Y.; Kamani, M. M.; and Mahdavi, M. 2020. Adaptive personalized federated learning. *arXiv preprint arXiv:2003.13461* .

Fallah, A.; Mokhtari, A.; and Ozdaglar, A. 2020. Personalized federated learning: A meta-learning approach. *arXiv preprint arXiv:2002.07948* .

Fan, J.; and Li, R. 2001. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association* 96(456): 1348–1360.

GDPR. 2016. General Data Protection Regulation (https://gdpr.eu/) .

Hanzely, F.; and Richtárik, P. 2020. Federated learning of a mixture of global and local models. *arXiv preprint arXiv:2002.05516* .

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Huang, Y.; Chu, L.; Zhou, Z.; Wang, L.; Liu, J.; Pei, J.; and Zhang, Y. 2020. Personalized Cross-Silo Federated Learning on Non-IID Data. *arXiv preprint arXiv:2007.03797* .

Ji, S.; Pan, S.; Long, G.; Li, X.; Jiang, J.; and Huang, Z. 2019. Learning private neural language modeling with attentive aggregation. In *IEEE International Joint Conference on Neural Networks*, 1–8.

Jiang, Y.; Konečnỳ, J.; Rush, K.; and Kannan, S. 2019. Improving federated learning personalization via model agnostic meta learning. *arXiv preprint arXiv:1909.12488* .

Kairouz, P.; McMahan, H. B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A. N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; et al. 2019. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977* .

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

Khodak, M.; Balcan, M.-F. F.; and Talwalkar, A. S. 2019. Adaptive gradient-based meta-learning methods. In *Advances in Neural Information Processing Systems*, 5915–5926.

Kingma, D. P.; and Ba, J. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images. *Technical Report, University of Toronto* .

Kulkarni, V.; Kulkarni, M.; and Pant, A. 2020. Survey of personalization techniques for federated learning. *arXiv preprint arXiv:2003.08673* .

LeCun, Y.; Cortes, C.; and Burges, C. 2010. MNIST handwritten digit database .

Li, T.; Sahu, A. K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; and Smith, V. 2020. Federated optimization in heterogeneous networks. In *Machine Learning and Systems*, 429–450.

Li, X.; Zhu, Z.; So, A. M.-C.; and Lee, J. D. 2019. Incremental methods for weakly convex optimization. *arXiv preprint arXiv:1907.11687* .

Mansour, Y.; Mohri, M.; Ro, J.; and Suresh, A. T. 2020. Three approaches for personalization with applications to federated learning. *arXiv preprint arXiv:2002.10619* .

Mansour, Y.; Mohri, M.; and Rostamizadeh, A. 2009. Domain adaptation: Learning bounds and algorithms. *arXiv preprint arXiv:0902.3430* .

McMahan, H. B.; Moore, E.; Ramage, D.; Hampson, S.; et al. 2016. Communication-efficient learning of deep networks from decentralized data. In *International Conference on Artificial Intelligence and Statistics*, 1273–1282.

Nemirovski, A.; Juditsky, A.; Lan, G.; and Shapiro, A. 2009. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization* 19(4): 1574–1609.

Nesterov, Y. 2013. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media.

Nichol, A.; Achiam, J.; and Schulman, J. 2018. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999* .

Reddi, S. J.; Kale, S.; and Kumar, S. 2018. On the convergence of adam and beyond. In *International Conference on Learning Representations*.

Rockafellar, R. T. 1976. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization* 14(5): 877–898.

Schneider, J.; and Vlachos, M. 2020. Mass personalization of deep learning. In *International Data Science Conference*.

Smith, V.; Chiang, C.-K.; Sanjabi, M.; and Talwalkar, A. S. 2017. Federated multi-task learning. In *Advances in Neural Information Processing Systems*, 4424–4434.

Verleysen, M.; and François, D. 2005. The curse of dimensionality in data mining and time series prediction. In *International Conference on Artificial Neural Networks: Computational Intelligence and Bioinspired Systems*, 758–770.

Wang, H.; Yurochkin, M.; Sun, Y.; Papailiopoulos, D.; and Khazaeni, Y. 2020. Federated learning with matched averaging. In *International Conference on Learning Representations*.

Wang, K.; Mathews, R.; Kiddon, C.; Eichner, H.; Beaufays, F.; and Ramage, D. 2019. Federated evaluation of on-device personalization. *arXiv preprint arXiv:1910.10252* .

Wikipedia. 2020. Curse of dimensionality. https://en.wikipedia.org/wiki/Curse_of_dimensionality.

Wilcoxon, F. 1992. Individual comparisons by ranking methods. In *Breakthroughs in Statistics*, 196–202. Springer.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747* .

Yang, Q.; Liu, Y.; Chen, T.; and Tong, Y. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology* 10(2): 1–19.

Yurochkin, M.; Agarwal, M.; Ghosh, S.; Greenewald, K.; Hoang, N.; and Khazaeni, Y. 2019. Bayesian nonparametric federated learning of neural networks. In *International Conference on Machine Learning*, 7252–7261.

Zhang, C.-H. 2010. Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics* 38(2): 894–942.

Zhang, J.; Karimireddy, S. P.; Veit, A.; Kim, S.; Reddi, S. J.; Kumar, S.; and Sra, S. 2019. Why ADAM beats SGD for attention models. *arXiv preprint arXiv:1912.03194* .

Zhang, X.; Hong, M.; Dhople, S.; Yin, W.; and Liu, Y. 2020. FedPD: A Federated Learning Framework with Optimal Rates and Adaptivity to Non-IID Data. *arXiv preprint arXiv:2005.11418* .

Zhao, Y.; Li, M.; Lai, L.; Suda, N.; Civin, D.; and Chandra, V. 2018. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582* .