On-Device Federated Learning via Second-Order Optimization with Over-the-Air Computation

Sheng Hua, Kai Yang, and Yuanming Shi
School of Information Science and Technology, ShanghaiTech University, Shanghai 201210, China
E-mail: {huasheng, yangkai, shiym}@shanghaitech.edu.cn

Abstract—**Federated learning becomes a promising approach for preserving privacy by keeping user data locally. The basic idea is that a central server iteratively aggregates distributed local models trained directly on mobile users' local datasets to form a high-quality global model by computing the weighted sum of the locally updated models. However, the communication cost becomes the main bottleneck as a large number of communication rounds are involved in the federated learning procedure. We propose to update local models by second-order optimization methods with fast convergence rates, thereby significantly reducing the communication rounds for global model updates. Furthermore, the over-the-air computation technique is adopted to improve communication efficiency for model aggregation by utilizing the superposition property of wireless channels. A nonconvex low-rank beamforming approach is then developed to support over-the-air computation via difference-of-convex-functions (DC) programming. Through extensive experiments, we reveal that the proposed DC algorithm is able to significantly minimize the aggregation error, and the second-order methods are quite robust to the model aggregation errors.**

I. Introduction

Recently distributed machine and deep learning techniques have achieved remarkable success in many applications, including image classification and speech recognition. Once mobile users collect or generate data like pictures or speech, they normally upload their data to a central server to perform such intelligent tasks. However, the sharing of data raises severe privacy concern since the dataset is typically personalized. A novel distributed learning paradigm called *Federated learning* is a promising approach to preserve user privacy by enabling users to train the model while keeping the dataset locally [1]. The recent proposed Federated Averaging (FedAvg) algorithm [2] serves as a framework for thousands of mobile devices interacting with a central server to collaboratively learn a high-quality global model.

In the original FedAvg algorithm, local models are normally updated by stochastic gradient descent (SGD) method. However, the slow convergence rate of SGD and limited bandwidth of wireless networks make the communication cost enormous, which may become the bottleneck of the whole system. The strong desire to alleviate the communication burden calls for algorithms with less communication rounds involved. To reduce the communication costs, various distributed second-order methods have recently been proposed to enjoy fast convergence rates [3], [4], [5], [6].

Notice that the global model is updated as the weighted sum of locally updated models. The global model aggregation procedure is a nomographic function and thus can be efficiently computed by over-the-air computation technique via concurrent transmission [7], [8]. The distortion between transmitted and received signal is inevitable and can be measured by mean-square-error (MSE). A corresponding min-max optimization problem is then formulated to minimize MSE, which turns out to be a nonconvex quadratically constrained quadratic programming (QCQP) problem. Although the semidefinite relaxation (SDR) technique [9] convexifies the lifting problem by dropping the rank-one constraint, its performance degrades dramatically as the problem setting size grows. Another state-of-the-art algorithm [10] can always find the global optimal solution, however the computational complexity is prohibitive. The difference-of-convex-functions (DC) [11] programming approach has recently been widely used in low-rank optimization [12], [13]. By introducing the DC representation for the nonconvex rank-one constraint, we develop a novel DC formulation which can be efficiently solved by the DC algorithm.

We conduct extensive experiments to compare the performance of SGD and second-order methods for federated learning tasks by considering the model aggregation error. Our major finding is that, with fast convergence rate, second-order method reveals a appealing property that the model aggregation errors will not accumulate in the learning procedure as the communication rounds are significantly reduced.

II. System model and Problem Formulation

In this section, we first present our on-device federated learning model, where the local models are updated by second-order optimization methods. We then present the over-the-air computation approach to minimize the global model aggregation errors via exploiting waveforms superposition in a wireless multiple access channel.

A. On-Device Federated Learning Model

The goal of federated learning is to collaboratively learn a global model, while keeping all training data locally to preserve user privacy. In this paper we mainly focus on the multiclass (C classes) classification problem, which is a common supervised learning problem in machine learning.

Specifically, consider a scenario with K single-antenna mobile devices and one N-antennas central base station (BS), as shown in Fig. 1. Let $(\boldsymbol{x}_i, y_i)$ denote a sample where $\boldsymbol{x}_i \in \mathbb{R}^d$ is the d-dimension feature vector and $y_i \in \{0, 1, \ldots, C-1\}$ is the corresponding label. Let $\boldsymbol{X} = [(\boldsymbol{x}_1, y_1); \ldots; (\boldsymbol{x}_n, y_n)] \in \mathbb{R}^{n\times(d+1)}$ denote the whole dataset with size $n = \sum_{k=1}^{K} n_k$,

This work was supported in part by the National Nature Science Foundation of China under Grant 61601290.

978-1-7281-1220-6/19/$31.00 ©2019 IEEE

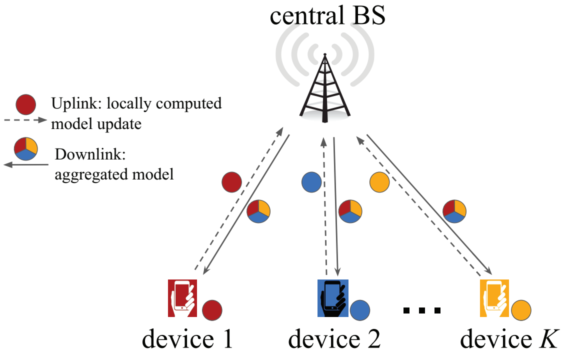

central BS
Uplink: locally computed model update
Downlink: aggregated model
device 1
device 2
device K

Fig. 1: The system model of on-device federated learning. The central base station first aggregate all locally computed models through uplink channels and then broadcast the aggregated one to devices through downlink channels.

Algorithm 1: General Federated Learning Framework Based on Federated Averaging Algorithm [2]

```
for each communication round $t$ do
    1. Model Distribution: broadcast the current global
       model $\boldsymbol{W}_t$ to all $|\mathcal{S}_t|$ active devices
    2. for each active device $k \in \mathcal{S}_t$ in parallel do
           $\boldsymbol{W}_k^{[t+1]} \leftarrow ClientUpdate(k, \boldsymbol{W}_t, \alpha_t)$
       end
    3. Model Aggregation:
       $\boldsymbol{W}^{[t+1]} \leftarrow \sum_{k \in \mathcal{S}_t} \frac{n_k}{n} \boldsymbol{W}_k^{[t+1]}$
end

ClientUpdate$(k, \boldsymbol{W}, \alpha)$:
for each local epoch do
    for each minibatch do
        compute update $\Delta \boldsymbol{W}$
        $\boldsymbol{W} \leftarrow \boldsymbol{W} - \alpha \Delta \boldsymbol{W}$
    end
end
```

which is distributed among K devices and n_k is the local dataset size of the k-th device. The goal of learning an optimal global classifier from local datasets can be achieved by minimizing an empirical risk function, i.e.,

$$\underset{\boldsymbol{W} \in \mathbb{R}^{d \times C}}{\text{minimize}} \quad F(\boldsymbol{W}) = \sum_{k=1}^{K} \frac{n_k}{n} f_k(\boldsymbol{W}), \tag{1}$$

$$\text{with } f_k(\boldsymbol{W}) = \frac{1}{n_k} \sum_{i=1}^{n_k} \ell(\boldsymbol{W}; \boldsymbol{x}_i, y_i) + \gamma \mathcal{R}(\boldsymbol{W}). \tag{2}$$

The first term in (2) is averaged data loss and the second term is the regularization loss with penalty factor γ. We assume f_k is smooth and twice differentiable. Since devices are in unreliable connections and disconnected from time to time, let $\mathcal{S}_t \subseteq \{1, \ldots, K\}$ denote the set of active devices at the t-th iteration. In order to improve the global model, the overall procedure for on-device federated learning consists of the following three phases shown in Algo. 1.

B. Second-Order Methods for Local Updates

The key step in *ClientUpdate* phase is to compute local updates with the minibatch data. In consideration of many successful applications based on stochastic gradient descent (SGD) algorithm or its variants, a natural choice is to implement SGD to perform local updates. This is exactly what is adopted in original Federated Averaging algorithm [2]. SGD or other similar first-order methods require low computational costs, however, the highly iterative nature of these methods calls for a great number of iterations thus incurring much communication overhead. In the federated learning setting devices are typically with limited bandwidth and relatively slow network connections [1], hence such overhead may become unacceptable.

Instead, we opt to use second-order methods to perform local updates for the following reasons. The communication burden is significantly decreased due to the fact that compared to first-order methods, second-order ones utilize the information of curvature and require much less communication rounds to converge. Moreover, faster convergence rate brings another appealing benefit that communication errors will not accumulate. In fact accumulated errors will dramatically hurt the quality of aggregated model, which is verified in our experiments. In this paper, we only focus on the canonical Newton's method, which has the similar form to $p = -\nabla^2 f_k^{-1} \nabla f_k$ [14]. Further improvements to the canonical Newton's method can be found in [5], [15].

Without loss of generality, we train a softmax classifier for our multiclass classification problem by adopting cross-entropy and ℓ_2 norm as the measurement of data loss and regularization loss, respectively. Specifically, $f_k(\boldsymbol{W})$ is given by

$$f_k(\boldsymbol{W}) = \frac{1}{n_k} \sum_{i=1}^{n_k} -\log\left(\frac{\exp(\boldsymbol{w}_{y_i}^{\mathsf{T}} \boldsymbol{x}_i)}{\sum_{j=1}^{C} \exp(\boldsymbol{w}_j^{\mathsf{T}} \boldsymbol{x}_i)}\right) + \gamma \sum_{j=1}^{C} \|\boldsymbol{w}_j\|_2^2, \tag{3}$$

where $\boldsymbol{W} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_C]$ and $\boldsymbol{w}_j \in \mathbb{R}^d$ is the parameter vector related to class j. Given the global model at the t-th round $\boldsymbol{W}^{[t]}$, each active device $k \in \mathcal{S}_t$ will first compute the local gradient direction with respect to $\boldsymbol{w}_j$ as

$$\boldsymbol{g}_{k,j}^{[t]} = \nabla_{\boldsymbol{w}_j} f_k\left(\boldsymbol{W}^{[t]}\right) = \frac{1}{n_k} \sum_{i=1}^{n_k} \boldsymbol{x}_i (p_j - \mathbf{1}_{y_i=j}) + 2\gamma \boldsymbol{w}_j \in \mathbb{R}^d, \tag{4}$$

where $p_j = \frac{\exp(\boldsymbol{w}_j^{\mathsf{T}} \boldsymbol{x}_i)}{\sum_{j=1}^{C} \exp(\boldsymbol{w}_j^{\mathsf{T}} \boldsymbol{x}_i)}$ is the predicted probability on class j and $\mathbf{1}_{(\cdot)}$ is an indicator function whose output is 1 if and only if the condition $(\cdot)$ is satisfied. The Hessian matrix at $\boldsymbol{W}^{[t]}$ with respect to $\boldsymbol{w}_j$ is computed as

$$\boldsymbol{H}_{k,j}^{[t]} = \nabla^2_{\boldsymbol{w}_j \boldsymbol{w}_j^{\mathsf{T}}} f_k\left(\boldsymbol{W}^{[t]}\right) = \frac{1}{n_k} \sum_{i=1}^{n_k} \boldsymbol{x}_i \boldsymbol{x}_i^{\mathsf{T}} (p_j^2 - p_j) + 2\gamma \boldsymbol{I}_d, \tag{5}$$

where $\boldsymbol{I}_d$ is the identity matrix. The k-th device will update its local model as $\boldsymbol{W}_k^{[t+1]} = [\boldsymbol{w}_j']_{j=1}^C$, where $\boldsymbol{w}_j' = \boldsymbol{w}_j - \alpha_t \boldsymbol{p}_{k,j}^{[t]}$, $\boldsymbol{p}_{k,j}^{[t]} = \left(\boldsymbol{H}_{k,j}^{[t]}\right)^{-1} \boldsymbol{g}_{k,j}^{[t]}$ is the Newton direction and α_t is the learning rate. After all active devices finishing updating local models in a parallel way, the central BS aggregates these models to form the new global model for the next round, i.e., $\boldsymbol{W}^{[t+1]} \leftarrow \sum_{k \in \mathcal{S}_t} \frac{n_k}{n} \boldsymbol{W}_k^{[t+1]}$.

C. Communication Model

In practice, the model aggregation process is accomplished by signal transmission through wireless channels. To simplify the notation, we will drop the communication round index t in the following context if it will not cause any ambiguity. Observe that the model aggregation expression is actually a nomographic function [16], which can be efficiently computed via concurrent transmission by using over-the-air computation technique [7]. To be more specific, we can rewrite the aggregation function as

$$\mathbf{w} = \psi\left(\sum_{k\in\mathcal{S}} \varphi_k(\mathbf{w}_k)\right), \tag{6}$$

where $\varphi_k{=}\frac{n_k}{n}$ is the pre-processing function at each device, $\psi = 1$ is the post-processing function at the BS, and

$$\mathbf{w}_k = [\underbrace{\mathsf{w}_k^{(1)},\ldots,\mathsf{w}_k^{(d)}}_{\boldsymbol{w}_1^{\mathsf{H}}},\ldots,\underbrace{\mathsf{w}_k^{(Cd-d+1)},\ldots,\mathsf{w}_k^{(Cd)}}_{\boldsymbol{w}_C^{\mathsf{H}}}]^{\mathsf{H}} \in \mathbb{C}^{Cd}$$

stretch the weight matrix $\boldsymbol{W}_k$ into a vector (in the field of wireless communcations, signals are discussed within complex domain. Therefore in the following we will consistently use the notation $\mathbf{w}_k \in \mathbb{C}^{Cd}$). The symbol vector $\boldsymbol{s}_k = \varphi_k(\mathbf{w}_k)$ is supposed to be normalized with unit variance, i.e., $\mathbb{E}(\boldsymbol{s}_k\boldsymbol{s}_k^{\mathsf{H}}) = \boldsymbol{I}$.

At each time slot $l \in \{1,\ldots,Cd\}$, each device sends the signal $s_k^{(l)} \in \mathbb{C}$ to the BS. The target signal received by the BS at the l-th time slot is denoted as $\mathsf{w}^{(l)} = \sum_{k\in\mathcal{S}} s_k^{(l)}$, with $s_k^{(l)} = \varphi_k(\mathsf{w}_k^{(l)})$. For ease of notation, we will omit the time slot index l by writing $s_k^{(l)}$ and $\mathsf{w}_k^{(l)}$ as s_k and w_k, respectively. The received signal after concurrent transmissions of all devices is given as $\boldsymbol{y} = \sum_{k\in\mathcal{S}} \boldsymbol{h}_k b_k s_k + \boldsymbol{n}$, where $b_k \in \mathbb{C}$ is the transmitter scalar, $\boldsymbol{h}_k \in \mathbb{C}^N$ is the channel vector between the k-th device and the BS, and $\boldsymbol{n} \sim \mathcal{CN}(\boldsymbol{0}, \sigma^2\boldsymbol{I})$ is the noise vector. Given P_0 as the maximum transmit power, each device has the transmit power constraint, i.e., $\mathbb{E}(|b_k s_k|^2) = |b_k|_2 \le P_0$. By introducing a normalizing factor η, the estimated target signal received at the BS is given by

$$\hat{\mathsf{w}} = \frac{1}{\sqrt{\eta}}\boldsymbol{a}^{\mathsf{H}}\boldsymbol{y} = \frac{1}{\sqrt{\eta}}\boldsymbol{a}^{\mathsf{H}}\sum_{k\in\mathcal{S}} \boldsymbol{h}_k b_k s_k + \frac{1}{\sqrt{\eta}}\boldsymbol{a}^{\mathsf{H}}\boldsymbol{n}, \tag{7}$$

where $\boldsymbol{a} \in \mathbb{C}^N$ is the receiver beamforming vector. The distortion between $\hat{\mathsf{w}}$ and w, which quantifies the over-the-air computation performance, can be measured by mean square error (MSE)

$$\begin{aligned} &\mathsf{MSE}\left(\hat{\mathsf{w}}, \mathsf{w}; \mathcal{S}, \boldsymbol{a}\right) = \mathbb{E}\left(|\hat{\mathsf{w}} - \mathsf{w}|^2\right) \\ &= \textstyle\sum_k |\boldsymbol{a}^{\mathsf{H}}\boldsymbol{h}_k b_k/\sqrt{\eta} - 1|^2 + \sigma^2\|\boldsymbol{a}\|^2/\eta. \end{aligned} \tag{8}$$

Suppose each device has its local CSI, and the BS has global CSI. Given a receiver beamforming vector $\boldsymbol{a}$, it's obvious that the MSE is minimized by the transmit scalar $b_k = \sqrt{\eta}\frac{(\boldsymbol{a}^{\mathsf{H}}\boldsymbol{h}_k)^{\mathsf{H}}}{\|\boldsymbol{a}^{\mathsf{H}}\boldsymbol{h}_k\|^2}$. We also have $|b_k|^2 <= P_0$, which is the transmit power constraint, therefore η is given as $\eta = P_0\min_{k\in\mathcal{S}}\|\boldsymbol{a}^{\mathsf{H}}\boldsymbol{h}_k\|^2$. We can rewrite the MSE as

$$\mathsf{MSE}\left(\hat{\mathsf{w}}, \mathsf{w}; \mathcal{S}, \boldsymbol{a}\right) = \frac{\|\boldsymbol{a}\|^2\sigma^2}{\eta} = \frac{\|\boldsymbol{a}\|^2\sigma^2}{P_0\min\limits_{k\in\mathcal{S}}\|\boldsymbol{a}^{\mathsf{H}}\boldsymbol{h}_k\|^2}. \tag{9}$$

D. Problem Formulation

In order to maintain the distortion at a low level, we need to design an optimal receiver beamforming vector for the following MSE minimization problem:

$$\underset{\boldsymbol{a}\in\mathbb{C}^N}{\text{minimize}} \left(\max_{k\in\mathcal{S}} \frac{\|\boldsymbol{a}\|^2}{\|\boldsymbol{a}^{\mathsf{H}}\boldsymbol{h}_k\|^2}\right), \tag{10}$$

which can be recast as [17]:

$$\begin{aligned} &\underset{\boldsymbol{a}\in\mathbb{C}^N}{\text{minimize}} && \|\boldsymbol{a}\|^2 \\ &\text{subject to} && \|\boldsymbol{a}^{\mathsf{H}}\boldsymbol{h}_k\|^2 \ge 1, \forall k \in \mathcal{S}. \end{aligned} \tag{11}$$

Unfortunately, this is a nonconvex quadratically constrained quadratic programming (QCQP) problem, which is NP hard. By adopting the matrix lifting technique, i.e., lifting the beamforming vector $\boldsymbol{a}$ to the positive semidefinite matrix $\boldsymbol{A} = \boldsymbol{a}\boldsymbol{a}^{\mathsf{H}}$ with $\operatorname{rank}(\boldsymbol{A}) = 1$, problem (11) can be recast as the following low-rank optimization problem:

$$\begin{aligned} \mathscr{P}: \quad &\underset{\boldsymbol{A}\in\mathbb{C}^{N\times N}}{\text{minimize}} && \operatorname{Tr}(\boldsymbol{A}) \\ &\text{subject to} && \operatorname{Tr}(\boldsymbol{A}\boldsymbol{H}_k) \ge 1, \forall k \in \mathcal{S}, \\ &&& \boldsymbol{A} \succeq \boldsymbol{0}, \operatorname{rank}(\boldsymbol{A}) = 1, \end{aligned}$$

where $\boldsymbol{H}_k = \boldsymbol{h}_k\boldsymbol{h}_k^{\mathsf{H}}$. The above problem is still nonconvex due to the nonconvex rank-one constraint. Dropping the rank-one constraint to relax this problem into a convex one yields the celebrated semidefinite relaxation (SDR) technique [9], which has been successfully adopted as an effective approach in many multicast beamforming applications [18], [19]. However, the solution to the convex relaxation problem does not always satisfy the rank-one constraint. If this happens, the SDR algorithm will generate an approximation solution by using Gaussian randomization technique. As shown in [17], the SDR algorithm works well when the problem size is small; however, its performance degrade dramatically when N and K increase. Another state-of-the-art algorithm is arugment cut based relaxation and branch-and-bound (ACR-BB) proposed in [10], which is a global approach based on the branch-and-bound algorithm. However, the computational complexity is exponential in the worst case. To overcome the drawbacks of both algorithms, in this paper we will develop a novel difference-of-convex-functions (DC) approach for problem $\mathscr{P}$, which has low computational complexity but still yields good performance.

III. Algorithm Description

In this section, we propose a DC program for problem $\mathscr{P}$, followed by an efficient DC algorithm to successively solve the convex subproblem.

We first propose a DC representation of the troublesome rank-one constraint, i.e.,

$$\operatorname{rank}(\boldsymbol{A}) = 1 \Leftrightarrow \operatorname{Tr}(\boldsymbol{A}) - \|\boldsymbol{A}\|_2 = 0, \text{ with } \operatorname{Tr}(\boldsymbol{A}) > 0. \tag{12}$$

The trace norm and spectral norm are defined as $\operatorname{Tr}(\boldsymbol{A}) = \sum_{i=1}^N \sigma_i(\boldsymbol{A})$ and $\|\boldsymbol{A}\|_2 = \sigma_1(\boldsymbol{A})$ respectively, where $\sigma_i(\boldsymbol{A})$ stands for the i-th largest singular value of matrix $\boldsymbol{A}$. ($\Rightarrow$) If $\boldsymbol{A}$ has rank one, we have $\sigma_1 \neq 0$ and $\sigma_i = 0, i = 2,\ldots,N$. Then the trace norm will be the same as spectral norm. ($\Leftarrow$) $\operatorname{Tr}(\boldsymbol{A}) - \|\boldsymbol{A}\|_2 = 0$ indicates that $\sigma_i = 0, i = 2,\ldots,N$, and we know $\sigma_1(\boldsymbol{A}) > 0$ from $\mathrm{Tr}(\boldsymbol{A}) > 0$. Therefore $\mathrm{rank}(\boldsymbol{A}) = 1$ holds true.

With the help of DC representation (12), we can optimize $\mathscr{P}$ by solving the following DC program:

$$\begin{aligned}\mathscr{P}_{\mathrm{DC}}: \quad & \underset{\boldsymbol{A}\in\mathbb{C}^{N\times N}}{\text{minimize}} && \mathrm{Tr}(\boldsymbol{A}) + \beta(\mathrm{Tr}(\boldsymbol{A}) - \|\boldsymbol{A}\|_2) \\ & \text{subject to} && \mathrm{Tr}(\boldsymbol{A}\boldsymbol{H}_k) \geq 1, \forall k \in \mathcal{S}, \\ & && \boldsymbol{A} \succeq \boldsymbol{0}, \mathrm{Tr}(\boldsymbol{A}) > 0, \end{aligned} \tag{13}$$

where the second term in the objective is the regularizer corresponding to the rank-one constraint with penalty factor β. If the optimal solution to the above problem $\boldsymbol{A}^\star$ satisfies $\mathrm{Tr}(\boldsymbol{A}^\star) - \|\boldsymbol{A}^\star\|_2 = 0$, we can conclude that $\boldsymbol{A}$ is an exactly rank-one PSD matrix and a feasible beamforming vector $\boldsymbol{a}^\star$ can be obtained by matrix decomposition $\boldsymbol{A}^\star = \boldsymbol{a}^\star(\boldsymbol{a}^\star)^{\mathsf{H}}$. Since the objective in $\mathscr{P}_{\mathrm{DC}}$ is still nonconvex, we develop the DC algorithm (DCA) for this problem.

The DCA iteratively solves a convex subproblem by replacing the third component in $\mathscr{P}_{\mathrm{DC}}$ with its linearization, i.e.,

$$\begin{aligned} & \underset{\boldsymbol{A}\in\mathbb{C}^{N\times N}}{\text{minimize}} && (1+\beta)\mathrm{Tr}(\boldsymbol{A}) - \beta\langle\partial\|\boldsymbol{A}^t\|_2, \boldsymbol{A}\rangle \\ & \text{subject to} && \mathrm{Tr}(\boldsymbol{A}\boldsymbol{H}_k) \geq 1, \forall k \in \mathcal{S}, \\ & && \boldsymbol{A} \succeq \boldsymbol{0}, \mathrm{Tr}(\boldsymbol{A}) > 0. \end{aligned} \tag{14}$$

Here $\partial\|\boldsymbol{A}^t\|_2$ is one subgradient of the spectral norm at point $\boldsymbol{A}^t$, and $\langle\cdot\rangle$ is the inner product of two matrices defined as $\langle\boldsymbol{X}, \boldsymbol{Y}\rangle = \mathrm{real}(\boldsymbol{X}^{\mathsf{H}}\boldsymbol{Y})$. The subdifferential of $\|\boldsymbol{A}\|_2$ is given by [20]

$$\partial\|\boldsymbol{A}\|_2 = \mathrm{conv}\{\boldsymbol{U}\mathrm{diag}(\boldsymbol{q})\boldsymbol{U}^{\mathsf{H}} : \boldsymbol{q} \in \partial\|\boldsymbol{\sigma}(\boldsymbol{A})\|_\infty\}, \tag{15}$$

where conv denotes the convex hull, $\boldsymbol{A} = \boldsymbol{U}\mathrm{diag}(\boldsymbol{\sigma}(\boldsymbol{A}))\boldsymbol{U}^{\mathsf{H}}$ is the singular value decomposition of $\boldsymbol{A}$ and $\boldsymbol{\sigma}(\boldsymbol{A}) = [\sigma_1(\boldsymbol{A}), \ldots, \sigma_N(\boldsymbol{A})]^{\mathsf{T}}$ is the vector formed by all singular values. Notice that we can easily compute an element of $\partial\|\boldsymbol{A}\|_2$ as $\boldsymbol{u}_1\boldsymbol{u}_1^{\mathsf{H}}$, where $\boldsymbol{u}_1$ is the singular vector associated with the largest singular value of $\boldsymbol{A}$. The DCA for $\mathscr{P}_{\mathrm{DC}}$ is summarized in Algorithm 2.

Algorithm 2: DC Algorithm for Solving Problem $\mathscr{P}_{\mathrm{DC}}$

```
Choose $\boldsymbol{A}^0 \succeq \boldsymbol{0}$ and initialize $t = 1$
while $Tr(\boldsymbol{A}^{t-1}) - \|\boldsymbol{A}^{t-1}\|_2$ is not zero do
    Find a component of $\partial\|\boldsymbol{A}^{t-1}\|_2$ according to (15)
    Solve (14) to obtain $\boldsymbol{A}^t$
    $t \leftarrow t + 1$
end
```

IV. Experimental Results

In order to evaluate the effectiveness of DC algorithm and second-order method in a practical federated learning problem, we conduct extensive experiments on the CIFAR-10 dataset ($n = 50000$, $d = 3072$) [21]. In the first part of experiments, we reveal the superiortiy of proposed DC algorithm in over-the-air computation. And in the second part, we compare models updated by first-order method (SGD) and second-order method (Newton). Throughout our experiments, the channel vectors $\boldsymbol{h}_k$'s between the central BS and devices follow the complex normal distribution $\boldsymbol{h}_k \sim \mathcal{CN}(\boldsymbol{0}, \boldsymbol{I})$, the average channel signal-to-noise-ratio (SNR) P_0/σ^2 is fixed at 5dB, and the penalty factor β in $\mathscr{P}_{\mathrm{DC}}$ is set as 1.

A. The Effectiveness of DC Algorithm

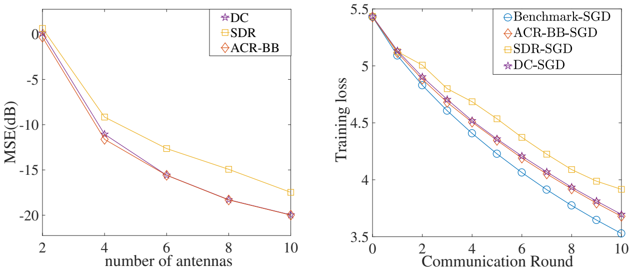
In our first experiment, we compare our proposed DC approach for solving $\mathscr{P}$ with two state-of-the-art approaches: **SDR** [9] and **ACR-BB** [10]. Since the performance of SDR deteriorates quickly as the problem size increases, we consider a small network with 5 mobile devices and the number of BS's antennas ranges from 2 to 10. Fig. 2a shows the mean-square-error obtained by three algorithms averaged on 100 independent channel realizations. The simulation results indicate that SDR fails to achieve satisfying MSE when the number of receiver antennas is large, while our proposed DC approach yields near-optimal performance.

In our second experiment, we want to demonstrate how MSE attained in model aggregation phase will influence the quality of global model. We assume the central BS is equipped with 10 antennas. The training dataset is distributed across 20 devices in a balanced way, i.e., each device has 2500 samples as their local dataset. In each communication round 5 random devices are selected as active, i.e., $|\mathcal{S}_t| = 5$. Local models are updated with SGD method and the learning rate α and regularization factor γ are well-tuned and determined by cross-validation. We compare the aggregated models (i.e., aggregation through over-the-air computation, termed as *DC-SGD*, *SDR-SGD* and *ACR-BB-SGD*) to the benchmark (i.e., ideal aggregation without any error, termed as *Benchmark-SGD*). Experiment results averaged over 5 channel realizations are shown in 2b. Gaps between all of three over-the-air computation methods and the benchmark enlarge as the communication process goes, which indicates that aggregation errors accumulate and gradually slow down the training process. By comparaing SDR to DC, we draw the conclusion that a higher aggregation error results in a larger training loss gap.

B. The Effectiveness of Newton Method

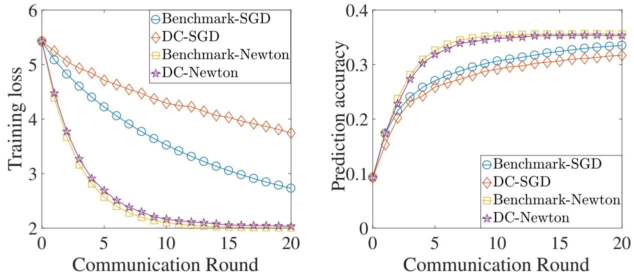
In this part, we aim to evaluate how the global model performance will benefit from local updates with Newton. Since our emphasis is on the comparison of SGD and Newthon in the federated learning with model aggregation error considered, DC approach is chosen as the underlying over-the-air computation technique, and the benchmarks are models obtained through ideal aggregation. All parameters are the same as those in the second experiment expect that in each communication round $|\mathcal{S}_t| = 15$. Training loss and test accuracy are illustrated in Fig. 3a and Fig. 3b, respectively. Both figures demonstrate that Newton converges much faster to a better point than the SGD does, which is the same as expected. Another major finding is that in both figures, the gap between DC and the benchmark for SGD is getting larger, while the gap is much smaller and more stable for Newton. This indicates that for SGD aggregation errors accumulate very fast and significantly hurt the model quality, while errors have merely neglectable influence on Newton. That's mainly because Newton converges to a relatively flat region within the very first few iterations, and in this region the model is less sensitive to errors.

In the subsequent experiments, we take a step further to demonstrate how an unbalanced data distribution will affect the performance, which is often the case in federated learning



(a) Gap in MSE (b) Training Loss

Fig. 2: (a) MSE obtained over the number of receiver antennas (b) Training loss obtained by different over-the-air algorithms. Local models are updated by SGD.



(a) Training Loss (b) Test accuracy

Fig. 3: Comparison between models updated by SGD and Newton in terms of training loss and test accuracy

problems. Relative gaps between aggregation via over-the-air computation and the benchmark after 20 communication rounds are summarized in Table I. In all scenarios that we have tested, Newton is relatively robust to aggregation errors and yields a close optimal performance in terms of training loss and prediction accuracy.

	Gap in Loss		Gap in Accuracy	
SD.(n_k)	SGD	Newton	SGD	Newton
0	37.10%	1.30% (28×)	5.49%	0.73% (8×)
513	40.64%	1.56% (26×)	6.44%	0.17% (38×)
1026	39.71%	1.13% (35×)	5.89%	0.43% (14×)
1539	31.23%	0.62% (50×)	3.03%	0.31% (10×)
2115	21.17%	0.48% (44×)	1.56%	0.20% (8×)

TABLE I: Relative gaps after 20 communication rounds between aggregation via over-the-air computation and the benchmark with unbalanced local dataset consumption. **SD.**(n_k) is the standard deviation of local dataset size, and the number in parenthesis indicates the performance gain of Newton compared to SGD.

V. Conclusions

In this paper, we investigated the federated learning problem over wireless networks, i.e., the central BS aggregates the locally updated models from distributed devices via the over-the-air computation technique. To alleviate the communication burden, we proposed to update local models by Newton's method which requires less communication rounds. To reduce the aggregation error in global model aggregation, we developed a novel DC approach for minimizing the mean-square-error (MSE). Simulation results verified the effectiveness of our DC approach for MSE minimization. Furthermore, compared to the stochastic gradient descent method, models updated by Newton's method can not only converge faster but also show much better robustness to aggregation errors.

References

[1] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[2] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, *et al.*, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artificial Intell. Stat.*, vol. 54, pp. 1273–1282, 2017.

[3] Y. Zhang and X. Lin, "DiSCO: Distributed optimization for self-concordant empirical loss," in *Proc. Int. Conf. Mach. Learn. (ICML)*, pp. 362–370, 2015.

[4] O. Shamir, N. Srebro, and T. Zhang, "Communication-efficient distributed optimization using an approximate newton-type method," in *Proc. Int. Conf. Mach. Learn. (ICML)*, pp. 1000–1008, 2014.

[5] S. Wang, F. Roosta-Khorasani, P. Xu, and M. W. Mahoney, "GIANT: Globally improved approximate newton method for distributed optimization," in *Proc. Neural Inf. Process. Syst. (NeurIPS)*, pp. 2338–2348, 2018.

[6] P. Xu, F. Roosta-Khorasani, and M. W. Mahoney, "Second-order optimization for non-convex machine learning: An empirical study," *arXiv preprint arXiv:1708.07827*, 2017.

[7] G. Zhu and K. Huang, "MIMO over-the-air computation for high-mobility multi-modal sensing," *IEEE Internet Things J.*, 2019.

[8] J. Dong, Y. Shi, and Z. Ding, "Blind over-the-air computation and data fusion via provable Wirtinger flow," *arXiv preprint arXiv:1811.04644*, 2018.

[9] N. D. Sidiropoulos, T. N. Davidson, and Z.-Q. Luo, "Transmit beamforming for physical-layer multicasting," *IEEE Trans. Signal Process.*, vol. 54, no. 6-1, pp. 2239–2251, 2006.

[10] C. Lu and Y.-F. Liu, "An efficient global algorithm for single-group multicast beamforming," *IEEE Trans. Signal Process.*, vol. 65, no. 14, pp. 3761–3774, 2017.

[11] P. D. Tao and L. T. H. An, "Convex analysis approach to DC programming: Theory, algorithms and applications," *Acta mathematica vietnamica*, vol. 22, no. 1, pp. 289–355, 1997.

[12] K. Yang, Y. Shi, and Z. Ding, "Low-Rank optimization for data shuffling in wireless distributed computing," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, pp. 6343–6347, IEEE, 2018.

[13] K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated learning via over-the-air computation," *arXiv preprint arXiv:1812.11750*, 2018.

[14] J. Nocedal and S. Wright, *Numerical optimization*. Springer Science & Business Media, 2006.

[15] F. Roosta-Khorasani and M. W. Mahoney, "Sub-sampled newton methods I: globally convergent algorithms," *arXiv preprint arXiv:1601.04737*, 2016.

[16] M. Goldenbaum, H. Boche, and S. Stańczak, "Harnessing interference for analog function computation in wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 61, no. 20, pp. 4893–4906, 2013.

[17] L. Chen, X. Qin, and G. Wei, "A uniform-forcing transceiver design for over-the-air function computation," *IEEE Wireless Commun. Lett.*, vol. 7, no. 6, pp. 942–945, 2018.

[18] Y. Shi, J. Zhang, and K. B. Letaief, "Robust group sparse beamforming for multicast green Cloud-RAN with imperfect CSI," *IEEE Trans. Signal Process.*, vol. 63, no. 17, pp. 4647–4659, 2015.

[19] Y. Shi, J. Cheng, J. Zhang, B. Bai, W. Chen, and K. B. Letaief, "Smoothed l_p-minimization for green Cloud-RAN with user admission control," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 4, pp. 1022–1036, 2016.

[20] G. A. Watson, "Characterization of the subdifferential of some matrix norms," *Linear Algebra Appl.*, vol. 170, pp. 33–45, 1992.

[21] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," tech. rep., 2009.