2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Federated Learning in Wireless Networks via Over-the-Air Computations*

Halil Yigit Oksuz[1,2,3], Fabio Molinari[1], Henning Sprekeler[2,3], Jörg Raisch[1,2]

Abstract— **In a multi-agent system, agents can cooperatively learn a model from data by exchanging their estimated model parameters, without the need to exchange the locally available data used by the agents. This strategy, often called *federated learning*, is mainly employed for two reasons: (i) improving resource-efficiency by avoiding to share potentially large data sets and (ii) guaranteeing privacy of local agents' data. Efficiency can be further increased by adopting a beyond-5G communication strategy that goes under the name of Over-the-Air Computation. This strategy exploits the interference property of the wireless channel. Standard communication schemes prevent interference by enabling transmissions of signals from different agents at distinct time or frequency slots, which is not required with Over-the-Air Computation, thus saving resources. In this case, the received signal is a weighted sum of transmitted signals, with unknown weights (fading channel coefficients). State of the art papers in the field aim at reconstructing those unknown coefficients. In contrast, the approach presented here does not require reconstructing channel coefficients by complex encoding-decoding schemes. This improves both efficiency and privacy.**

I. INTRODUCTION

Over the last decade, topics related to machine learning have attracted a great deal of attention due to their success in many application areas (e.g., [1]–[3]). As computational power increases, we have smaller but more intelligent and powerful devices, which are able to handle big volumes of data and more complex computations.

Data is often distributed over these intelligent devices or agents, such as smartphones, personal computers and distributed data centers. In a centralized learning approach, all data are collected in a computationally powerful machine, such as a cloud-based unit or a server, on which the model is trained [4]. However, this approach may not be suitable for specific applications requiring privacy, resource efficiency, and low-latency. For instance, due to privacy concerns, smartphone users might not be willing to share their data; moreover, with an increasing number of agents, more communication resources must be allocated to share the agents' data with the central unit. However, it is also possible to train a model without centralized data collection. In this case, agents individually train models on their respective data and share their individually trained models (in the form of parameter vectors) with the central unit. Sharing parameter vectors is, in general, less expensive in terms of communication resources than sharing all the local data sets. This approach is often referred to as *federated learning*, and allows more private and communication efficient learning (see [5]–[8]).

Even though federated learning allows improved privacy by avoiding to share individual data sets, sharing local model parameters with the central unit might still reveal sensitive information [9]–[11]. It is possible to tackle this privacy problem with popular techniques like encryption or differential privacy [12], but they come at the price of lower efficiency and performance [9], [13].

As the number of agents and the number of model parameters increase, the communication load on the overall system increases as well [10], [11], [14]. To cope with this problem, one can carry out local training on agents for multiple steps or utilize *quantization* and/or *sparsification* on the model updates in order to accomplish an efficient compression; another approach is allowing only a subset of agents to transmit at specific time steps [15]–[19]. However, most of these techniques are not resource efficient in the sense that they increase the need of bandwidth or the number of communication rounds, which in general leads to a decrease in total throughput and learning speed as also observed in [20], [21].

Instead, we consider the so-called Over-The-Air computation approach, e.g., [22], to improve communication efficiency. When multiple agents transmit at the same time and in the same frequency band, signals are affected by the physical phenomenon of interference. Standard communication protocols prevent interference by transmitting signals orthogonally: in TDMA (Time Division Multiple Access), agents are assigned different time slots when they can transmit, whereas in FDMA (Frequency Division Multiple Access), different frequency bands are allocated to different users.

The philosophy of Over-The-Air Computation is to exploit interference rather than combat it, to increase communication efficiency. For example, in a system composed of N agents, each of which has to transmit an m-dimensional parameter vector $\theta \in \Re^m$ to a central unit, TDMA or FDMA would require mN orthogonal transmissions (multiplexed in time or frequency), whereas Over-The-Air Computation requires

* This work has been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2002/1 "Science of Intelligence" – project number 390523135.

[1] H.Y. Oksuz, F. Molinari, and J. Raisch are with the Control Systems Group at Technische Universität Berlin, Germany. {oksuz@tu-berlin.de}, {molinari,raisch}@control.tu-berlin.de.

[2] H.Y. Oksuz, H. Sprekeler, and J. Raisch are with Exzellenzcluster Science of Intelligence, Technische Universität Berlin, Marchstr. 23, 10587, Berlin, Germany.

[3] H.Y. Oksuz and H. Sprekeler are with the Modelling Cognitive Processes Group at Technische Universität Berlin, Germany.

979-8-3503-0123-6/23/$31.00 ©2023 IEEE

only m orthogonal transmissions. Due to its advantages, federated learning has been carried out by [23]–[25] via over-the air computation.

With this in mind, the main contributions of this paper can be summarized as follows:

- Unlike the studies by [23] and [25], we do not assume to know (nor to be able to reconstruct) channel coefficients, but we present an algorithm that can deal with their unknown nature. We will not need extra pre-processing to reconstruct the channel, which makes the proposed scheme more time and resource efficient.
- Privacy is inherently guaranteed by the unknown nature of the channel. The central unit will not be able to reconstruct information transmitted by individual agents.

The remainder of this paper is organized as follows: we describe the problem setup in Section II. In Section III, we present the proposed federated learning algorithm and prove its convergence. A numerical example is presented in Section IV. Concluding remarks are given in Section V.

Notation

The set of real numbers is denoted by $\Re$, $\Re^m$ represents m-dimensional Euclidean space. $\mathbb{N}$ and $\mathbb{N}_0$ respectively denote the set of natural numbers and the set of nonnegative integers. For a vector $x \in \Re^m$, x^T denotes its transpose. The 2-norm of vector x is denoted by $||x||$. The expected value of a random variable p is denoted by $\mathbb{E}[p]$. Given a finite set S, its cardinality is denoted by $|S|$. For a differentiable function $f : \Re^m \to \Re$, $\nabla f(\mathbf{x})$ represents the gradient of the function f at $x \in \Re^m$. The projection of $x \in \Re^m$ onto a nonempty closed convex set $S \subset \Re^m$ is denoted by $\mathbf{P}_S(x)$, where $\mathbf{P}_S(x) = \arg\min_{s \in S} ||s - x||$.

II. Problem Description

A. *Federated Learning with Constraints*

Consider a system of N agents connected to a server or cloud based unit, whose objective is to carry out a machine learning task. Each agent can access different portions of the dataset, and has an individual local cost function. We denote the set of data available to the i-th agent by $D_i = \{d_i^n\}_{n=1}^{|D_i|}$ and use $\mathcal{L}_i(d_i^n, \theta)$ to represent the value of the cost function of a model with parameter $\theta \in \mathcal{R}^m$. In a supervised learning setting, the dataset D_i consists of pairs of inputs and targets, i.e., $d_i^n = (u_i^n, z_i^n)$, where u_i^n and z_i^n represent, respectively, input and target data. Moreover, the private local cost function of agent i can be expressed as

$$f_i(\theta) = \frac{1}{|D_i|} \sum_{n=1}^{|D_i|} \mathcal{L}_i(d_i^n, \theta), \tag{1}$$

If the global cost function is defined as the average of all local cost functions, i.e.,

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^{N} f_i(\theta), \tag{2}$$

then, the objective of the overall system is to cooperatively solve the following constrained optimization problem

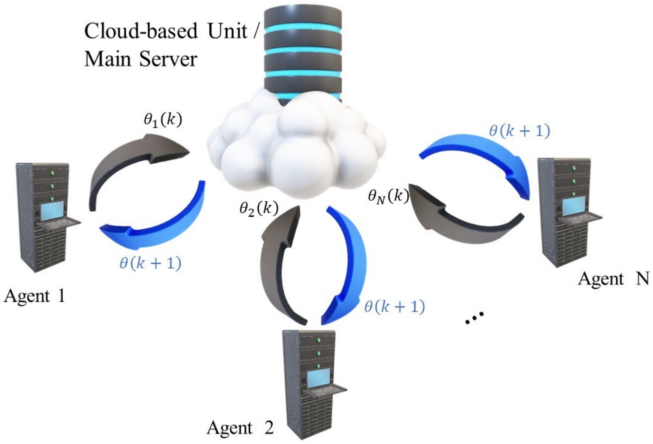

Fig. 1: An illustration of a federated learning setting.

$$\theta^* = \arg\min_{\theta \in \Theta} \mathcal{L}(\theta) = \arg\min_{\theta \in \Theta} \frac{1}{N} \sum_{i=1}^{N} f_i(\theta), \tag{3}$$

where $\Theta \subset \Re^m$ is a nonempty constraint set. In what follows, we denote $\mathcal{L}(\theta^*)$ by $\mathcal{L}^*$. Moreover, the set of optimal solutions is defined as

$$\Theta^* = \{\theta \in \Theta \mid \frac{1}{N} \sum_{i=1}^{N} f_i(\theta) = \mathcal{L}^*\}.$$

If the entire dataset, i.e.,

$$D = D_1 \cup D_2 \cup \cdots \cup D_N$$

were known to the server, one could employ a centralized gradient descent based optimization to solve the above global learning task [3]. However, in the considered federated learning approach, each agent can access only its own dataset, on which it trains its own model. After local optimization steps, the current versions of the local parameter estimates are transmitted to the server, where they are aggregated. The aggregated version is then transmitted to the agents for the next optimization step (see Fig. 1).

B. *Over-the-Air Communication*

In wireless communication systems, the wireless multiple access channel (WMAC) model has been extensively used to characterize communication between multiple transmitters and a single receiver over fading channels, e.g., [26], [27]. Throughout this paper, we employ the WMAC-based communication model described by [20], [28]. In this model, multiple (say N) agents simultaneously send information $\{\mathbf{s}_i(k) \in \Re^m\}_{i=1}^N$ in the same frequency band. This information is corrupted by the channel and superimposed by the receiver, i.e., the received information is

$$\mathbf{r}^{rec}(k) = \sum_{i=1}^{N} \alpha_i(k) \mathbf{s}_i(k), \tag{4}$$

where the $\alpha_i(k)$ are unknown time-varying positive channel coefficients.

Algorithm 1: FedCOTA

```
Initialization: $\theta(0) \in \Theta$.
Learning Loop:
for each time step $k \in \mathbb{N}_0$ do
    The server broadcasts $\theta(k)$
    Each agent $i$ updates its local variables:
        $\theta_i(k) = \theta(k) - \eta(k)\nabla f_i\big(\theta(k)\big)$    (5)
        $\rho_i(k) = 1$    (6)
    Each agent $i$ transmits $\theta_i(k)$ and $\rho_i(k)$
    The server receives:
        $\theta^{\text{rec}}(k) = \sum_{i=1}^{N} \alpha_i(k)\theta_i(k)$    (7)
        $\rho^{\text{rec}}(k) = \sum_{i=1}^{N} \alpha_i(k)\rho_i(k)$    (8)
    The server updates:
        $\theta(k+1) = \mathbf{P}_\Theta\left(\frac{\theta^{\text{rec}}(k)}{\rho^{\text{rec}}(k)}\right)$    (9)
end for
```

As indicated above, employing Over-the-Air computation for federated learning provides the following benefits:

- **privacy**: the channel coefficients $\{\alpha_i(k)\}_{i=1}^N$ in (4) are unknown, thus it is impossible for the central unit to individually reconstruct $\{s_i(k)\}_{i=1}^N$ from $\mathbf{r}^{rec}(k)$;
- **efficiency**: as shown by [21], Over-the-Air Computation can achieve better-than-linear scaling of the communication cost as the number of transmitters grows.

Unlike the algorithm suggested by [25], our approach does not require any knowledge of the channel coefficients, nor do they need to be reconstructed at each time step, thus removing one source of complexity.

III. Federated Constrained Over-the-Air Learning (FedCOTA) Algorithm

In this section, we first introduce the proposed algorithm that allows agents to carry out federated learning via over-the-air communication. Then, we will investigate the convergence properties of the proposed algorithm.

A. Algorithm

The FedCOTA algorithm is summarized in Algorithm 1. At the beginning, the server broadcasts to agents $\theta(0) \in \Theta$. Then, through iterations, each agent computes its own parameter vector $\theta_i(k)$ by using the local update rule (5). Afterwards, all agents transmit simultaneously (and in the same frequency band) their local parameter vectors $\theta_i(k)$ to the central unit. Then, again simultaneously and in the same frequency band a constant $\rho_i(k) = 1$ is transmitted by all agents. Because of the superposition property of the wireless channel (see Section II-B), the server receives (7) and (8) at time step k. Finally, the server computes (9) thus obtaining $\theta(k+1)$, which can be written as

$$\begin{aligned}\theta(k+1) &= \mathbf{P}_\Theta\left(\frac{\theta^{\text{rec}}(k)}{\rho^{\text{rec}}(k)}\right) = \mathbf{P}_\Theta\left(\sum_{i=1}^{N}\frac{\alpha_i(k)}{\sum_{i=1}^{N}\alpha_i(k)}\theta_i(k)\right)\\ &= \mathbf{P}_\Theta\left(\sum_{i=1}^{N} h_i(k)\theta_i(k)\right) \end{aligned}\tag{10}$$

where $\mathbf{P}_\Theta$ is the projection operator onto the set Θ, the $\alpha_i(k)$ are the unknown time-varying positive real channel coefficients, and the $h_i(k) = \frac{\alpha_i(k)}{\sum_{i=1}^{N}\alpha_i(k)}$ are the corresponding normalized channel coefficients. By construction, the $h_i(k)$ are positive and

$$\sum_{i=1}^{N} h_i(k) = 1, \tag{11}$$

for all $k \geq 0$.

B. Preliminaries

The following assumptions, also made in similar papers in the field, will hold throughout the paper.

Assumption 1: The constraint set $\Theta \subset \Re^m$ is convex and compact. As a consequence (see [29, Theorem 2.41, p.40]), Θ is then also closed and bounded.

Assumption 2: The cost functions $f_i(\theta)$ are continuously differentiable and strongly convex with L-Lipschitz continuous gradients, i.e., for any $\theta_1, \theta_2 \in \Theta \subset \Re^m$ and for all $i = 1, 2, \ldots, N$, the following inequalities hold:

(i) $\exists \mu_i > 0$ such that

$$f_i(\theta_2) - f_i(\theta_1) \geq \nabla f_i(\theta_1)^T(\theta_2 - \theta_1) + \frac{\mu_i}{2}||\theta_2 - \theta_1||^2, \tag{12}$$

(ii) $\exists L_i > 0$ such that

$$||\nabla f_i(\theta_1) - \nabla f_i(\theta_2)|| \leq L_i||\theta_1 - \theta_2||. \tag{13}$$

Remark 1: Note that the global cost function $\mathcal{L}(\theta)$ is then also differentiable and strongly convex with L-Lipschitz continuous gradient, i.e.,

$$\begin{gathered}\mathcal{L}(\theta_2) - \mathcal{L}(\theta_1) \geq \nabla\mathcal{L}(\theta_1)^T(\theta_2 - \theta_1) + \frac{\mu}{2}||\theta_2 - \theta_1||^2\\ ||\nabla\mathcal{L}(\theta_1) - \nabla\mathcal{L}(\theta_2)|| \leq L||\theta_1 - \theta_2||,\end{gathered}$$

where $\mu = \frac{1}{N}\sum_{i=1}^{N}\mu_i$ and $L = \frac{1}{N}\sum_{i=1}^{N}L_i$ (see [30]).

Remark 2: Note that Assumptions 1 and 2 imply that the local and global cost functions have bounded gradients on Θ.

Note that Assumptions 1 and 2 have been widely utilized in the federated learning and distributed optimization literature to illustrate the existence of a solution to problem (3) and convergence properties of the proposed algorithms (see [3], [25], [31]–[35], and the references therein).

Assumption 3: For all $k \geq 0$, the step size in Algorithm 1 is chosen as $\eta(k) = \frac{\eta_c}{\sqrt{k+1}}$, where $0 < \eta_c \leq \frac{1}{L}$.

Remark 3: Under Assumption 3, the step size $\eta(k)$ is decreasing and satisfies $\sum_{k=0}^{\infty}\eta(k) = \infty$ and $\lim_{k\to\infty}\eta(k) = 0$.

Conditions similar to Assumption 3 on the step size have also been utilized in many previous studies (e.g., [3],

[34]–[40], and the references therein) to show the exact convergence to an optimal solution.

Lemma 1: If $f(\cdot)$ is continuously differentiable, convex with L-Lipschitz continuous gradients, then for any $\theta_1, \theta_2 \in \Re^m$, the following inequality holds:

$$0 \leq f(\theta_2) - f(\theta_1) - \nabla f(\theta_1)^T(\theta_2 - \theta_1) \leq \frac{L}{2}||\theta_1 - \theta_2||^2.$$

Proof: The result is a direct consequence of [35, Theorem 2.1.5]. ■

Remark 4: From Lemma 1 and the definition of strong convexity (12), we have $\mu \leq L$ if a function is continuously differentiable, strongly convex with L-Lipschitz continuous gradients. Hence, $\eta_c \leq \frac{1}{L} \leq \frac{1}{\mu}$ holds by Assumption 3.

Assumption 4: The unknown time-varying positive real channel coefficients are assumed to be independent realizations (across time and agents) of the same probability distribution, i.e., $\forall k \in \mathbb{N}_0$, $\alpha_i(k) \sim \mathcal{D}\big(\bar{\alpha}, Var(\alpha)\big)$, where $\bar{\alpha}$ and $Var(\alpha)$ respectively denote the mean and variance of the distribution $\mathcal{D}$.

As in [20], [28], we refer here to [41, Ch 2.3, Ch 2.4] and [42, Ch 5.4], thus considering channel coefficients independent realizations of the same probability distribution (see [43]).

Lemma 2: Under Assumption 4, $\mathbb{E}[h_i(k)] = \frac{1}{N}$ holds for all $i = 1, 2, \ldots, N$ and $k \geq 0$.

Proof: From (11), we have $h_i(k) = \frac{\alpha_i(k)}{\sum_{i=1}^N \alpha_i(k)}$ and $\sum_{i=1}^N h_i(k) = 1$. Since each $\alpha_i(k)$ is sampled from the same distribution as stated in Assumption 4, the underlying distribution of each $h_i(k)$ is the same, i.e., $\mathbb{E}[h_i(k)] = \mathbb{E}[h_j(k)]$ holds for all $i, j \in \{1, 2, \cdots, N\}$ and $\forall k \geq 0$. Moreover, due to the linearity of expectation, we have

$$1 = \mathbb{E}\Big[\sum_{i=1}^N h_i(k)\Big] = \sum_{i=1}^N \mathbb{E}[h_i(k)] = N\mathbb{E}[h_i(k)],$$

which implies $\mathbb{E}[h_i(k)] = \frac{1}{N}$ for all $i = 1, 2, \ldots, N$. ■

C. Convergence Analysis of the FedCOTA Algorithm

First, we present preparatory results needed to show the convergence of the FedCOTA algorithm. Consider

$$C(k) = C_k = 1 - \eta(k)\mu. \tag{14}$$

Note that, under Assumption 3, $C_k \geq 0$ holds $\forall k \in \mathbb{N}_0$. As $\eta(k)$ is decreasing, C_k is increasing in k, hence it suffices to show $C_0 \geq 0$. This immediately follows from $C_0 = 1 - \eta_c\mu$ and $\eta_c \leq \frac{1}{\mu}$ (see Remark 4), hence $C_0 \geq 0$. Note furthermore that, as C_k is increasing, $C_k > 0$ holds for all $k \geq 1$.

Lemma 3: Under Assumption 3, $\lim_{k\to\infty}(C_k)^k = 0$.

Proof: By letting $C_k = 1 - \frac{Q}{\sqrt{k+1}}$, where $Q = \eta_c\mu > 0$, we write

$$\lim_{k\to\infty}(C_k)^k = \lim_{k\to\infty}\Big(1 - \frac{Q}{\sqrt{k+1}}\Big)^k. \tag{15}$$

Note that $\forall x \in \Re$, we have $1 + x \leq e^x$. Hence, $\forall k \in \mathbb{N}_0$

$$C_k = 1 - \frac{Q}{\sqrt{k+1}} \leq e^{-\frac{Q}{\sqrt{k+1}}}. \tag{16}$$

Moreover, as noted above, $C_k \geq 0$, hence

$$0 \leq (C_k)^k \leq e^{-\frac{Qk}{\sqrt{k+1}}}. \tag{17}$$

As the term on the right hand side of (17) goes to zero for $k \to \infty$, we have established that $\lim_{k\to\infty}(C_k)^k = 0$. ■

Lemma 4: Under Assumption 3, $\prod_{k=0}^{\infty} C_k = 0$.

Proof: By using the relation $1 + x \leq e^x$, $\forall x \in \Re$, we write

$$\begin{aligned}\prod_{t=0}^{k} C_t &= \prod_{t=0}^{k}\big(1 - \eta(t)\mu\big)\\ &\leq \prod_{t=0}^{k} \mathrm{e}^{-\eta(t)\mu} = \mathrm{e}^{-\sum_{t=0}^{k}\eta(t)\mu}\end{aligned} \tag{18}$$

Taking the limit as $k \to \infty$ on both sides gives

$$\lim_{k\to\infty}\prod_{t=0}^{k} C_t \leq \mathrm{e}^{-\mu\sum_{t=0}^{\infty}\eta(t)} = 0 \tag{19}$$

since by Assumption 3, $\mu\sum_{t=0}^{\infty}\eta(t) = \infty$. ■

Lemma 5: Under Assumption 3, for $k \to \infty$,

$$\sum_{t=0}^{k-2}\Big(\prod_{l=t+1}^{k-1} C_l\Big)\eta^2(t) + \eta^2(k-1)$$

converges to an arbitrarily small positive value ε.

Proof: See Appendix. ■

We are now ready to present the main result.

Theorem 1: Let Assumptions 1, 2, 3, and 4 hold. Then, $\exists\theta^* \in \Theta^*$ such that for $k \to \infty$, $\mathbb{E}\big[||\theta(k) - \theta^*||^2\big]$ is arbitrarily small.

Proof: For any $\theta^* \in \Theta^*$, by using (5), (10), (11), and the non-expansive[1] property of the projection $\mathbf{P}_\Theta$, we write

$$\begin{aligned}||\theta(k+1) - \theta^*||^2 &= \Big|\Big|\mathbf{P}_\Theta\Big(\sum_{i=1}^N h_i(k)\theta_i(k)\Big) - \mathbf{P}_\Theta\big(\theta^*\big)\Big|\Big|^2\\ &\leq \Big|\Big|\sum_{i=1}^N h_i(k)\big(\theta(k) - \eta(k)\nabla f_i\big(\theta(k)\big)\big) - \theta^*\Big|\Big|^2\\ &= \big|\big|\theta(k) - \theta^*\big|\big|^2\\ &\quad - 2\eta(k)\sum_{i=1}^N h_i(k)\nabla f_i^T\big(\theta(k)\big)\big(\theta(k) - \theta^*\big)\\ &\quad + \Big|\Big|\eta(k)\sum_{i=1}^N h_i(k)\nabla f_i\big(\theta(k)\big)\Big|\Big|^2.\end{aligned} \tag{20}$$

[1] $||\mathbf{P}_\Theta(x) - \mathbf{P}_\Theta(y)|| \leq ||x - y||$ holds for all $x, y \in \Re^m$ if Θ is a nonempty closed convex set (see [44]).

Note that boundedness of the constraint set Θ and Lipschitz continuity of $\nabla f_i\big(\theta(k)\big)$ imply that $\exists D_\Theta, M > 0$ such that

$$\big|\big|\theta(k) - \theta^*\big|\big| \leq D_\Theta, \tag{21}$$

$$\big|\big|\nabla f_i\big(\theta(k)\big)\big|\big| \leq M. \tag{22}$$

Then, an upper-bound for the last term on the right hand side of (20) can be written as

$$\begin{aligned}\Big|\Big|\eta(k)\sum_{i=1}^{N} h_i(k)\nabla f_i\big(\theta(k)\big)\Big|\Big|^2 &= \eta^2(k)\Big|\Big|\sum_{i=1}^{N} h_i(k)\nabla f_i\big(\theta(k)\big)\Big|\Big|^2\\ &\leq \eta^2(k)\sum_{i=1}^{N} h_i(k)\Big|\Big|\nabla f_i\big(\theta(k)\big)\Big|\Big|^2\\ &\leq \eta^2(k)M^2,\end{aligned} \tag{23}$$

which follows from (11), (22), and the convexity of the function $||\cdot||^2$. Subsequently, taking the expectations of both sides of (20) and using the linearity of expectation results in

$$\begin{aligned}\mathbb{E}\big[||\theta(k+1) - \theta^*||^2\big] \leq \mathbb{E}\big[\big|\big|\theta(k) - \theta^*\big|\big|^2\big]&\\ -2\eta(k)\mathbb{E}\big[\sum_{i=1}^{N} h_i(k)\nabla f_i^T\big(\theta(k)\big)\big(\theta(k) - \theta^*\big)\big]&\\ +\eta^2(k)M^2,&\end{aligned} \tag{24}$$

where the second term on the right hand side can be written as

$$\begin{aligned}&-2\eta(k)\mathbb{E}\big[\sum_{i=1}^{N} h_i(k)\nabla f_i^T\big(\theta(k)\big)\big(\theta(k) - \theta^*\big)\big]\\ &= -2\eta(k)\sum_{i=1}^{N}\mathbb{E}\big[h_i(k)\nabla f_i^T\big(\theta(k)\big)\big(\theta(k) - \theta^*\big)\big].\end{aligned} \tag{25}$$

Note that the statistics of $h_i(k)$ $(i = 1, 2, \cdots, N)$ are independent of $h_i(t)$ for $t < k$, which implies that $\theta(k)$ and $h_i(k)$ are statistically independent at time k since the statistics of $\theta(k)$ are dependent only of $h_i(t)$ for $t < k$ and $i = 1, 2, \cdots, N$. Hence, by using (2), Lemma 2, and the linearity of the expectation, we can further write (25) as

$$\begin{aligned}&-2\eta(k)\sum_{i=1}^{N}\mathbb{E}\big[h_i(k)\nabla f_i^T\big(\theta(k)\big)\big(\theta(k) - \theta^*\big)\big]\\ &= -2\eta(k)\sum_{i=1}^{N}\mathbb{E}\big[h_i(k)\big]\mathbb{E}\big[\nabla f_i^T\big(\theta(k)\big)\big(\theta(k) - \theta^*\big)\big]\\ &= -2\eta(k)\sum_{i=1}^{N}\frac{1}{N}\mathbb{E}\big[\nabla f_i^T\big(\theta(k)\big)\big(\theta(k) - \theta^*\big)\big]\\ &= -2\eta(k)\mathbb{E}\big[\sum_{i=1}^{N}\frac{1}{N}\nabla f_i^T\big(\theta(k)\big)\big(\theta(k) - \theta^*\big)\big]\\ &= -2\eta(k)\mathbb{E}\big[\nabla\mathcal{L}^T\big(\theta(k)\big)\big(\theta(k) - \theta^*\big)\big].\end{aligned} \tag{26}$$

Moreover, by Assumption 2 (strong convexity of $f_i(\theta)$ and $\mathcal{L}\big(\theta(k)\big)$), we have

$$\begin{aligned}-2\eta(k)\nabla\mathcal{L}^T\big(\theta(k)\big)\big(\theta(k) - \theta^*\big) \leq -2\eta(k)\big(\mathcal{L}\big(\theta(k)\big) - \mathcal{L}\big(\theta^*\big)\big)&\\ -\eta(k)\mu||\theta(k) - \theta^*||^2.&\end{aligned} \tag{27}$$

Note that $\mathcal{L}\big(\theta(k)\big) - \mathcal{L}\big(\theta^*\big) \geq 0$ holds for all $k \geq 0$ since $\theta^* \in \Theta^*$ (optimal point in the constraint set), which together with taking the expectations of both sides of (27) results in

$$-2\eta(k)\mathbb{E}[\nabla\mathcal{L}^T\big(\theta(k)\big)\big(\theta(k) - \theta^*\big)] \leq -\eta(k)\mu\mathbb{E}[||\theta(k) - \theta^*||^2]. \tag{28}$$

By using (25)-(28), (24) can be rewritten as

$$\begin{aligned}\mathbb{E}\big[||\theta(k+1) - \theta^*||^2\big] &\leq \Big(1 - \eta(k)\mu\Big)\mathbb{E}\big[\big|\big|\theta(k) - \theta^*\big|\big|^2\big]\\ &\qquad + \eta^2(k)M^2\\ &= C_k\mathbb{E}\big[||\theta(k) - \theta^*||^2\big] + \eta^2(k)M^2.\end{aligned} \tag{29}$$

The recursive relation (29) can be written as

$$\begin{aligned}\mathbb{E}\big[\big|\big|\theta(k) - \theta^*\big|\big|^2\big] \leq \Big(\prod_{t=0}^{k-1} C_t\Big)\mathbb{E}\big[\big|\big|\theta(0) - \theta^*\big|\big|^2\big]&\\ + M^2\Bigg(\sum_{t=0}^{k-2}\Big(\prod_{l=t+1}^{k-1} C_l\Big)\eta^2(t) + \eta^2(k-1)\Bigg).&\end{aligned} \tag{30}$$

Note that by Lemma 4, $\prod_{t=0}^{\infty} C_t = 0$ holds. Moreover, by Lemma 5, Assumption 3, taking the limits of both sides of (30) completes the proof. ■

IV. Numerical Example

We now apply the FedCOTA algorithm to a federated logistic regression problem, where a system of agents, each with access to only its own local dataset, tries to accomplish a global binary classification task. Let the dataset available to the i-th agent be $D_i = \{d_i^n\}_{n=1}^{|D_i|}$, where $d_i^n = (u_i^n, z_i^n) \in \Re^m \times \{0, 1\}$, and u_i^n and z_i^n respectively represent input data and labels available to i-th agent. Notice that all the agents have 2 different classes of data, labeled by 0 or 1, and their objective is to find a separating hyperplane in $\Re^m$ by using the existing data so that the agent is able to separate some unseen data from different classes. In order to accomplish this in a federated manner, each agent utilizes cross-entropy as its local cost function, which can be expressed as

$$\begin{aligned}f_i(\theta, d_i) = \lambda||\theta||_2^2 - \frac{1}{|D_i|}\Bigg(\sum_{n=1}^{|D_i|} z_i^n \log\big(S(\theta^T u_i^n)\big)&\\ + (1 - z_i^n)\log\big(1 - S(\theta^T u_i^n)\big)\Bigg)&\end{aligned} \tag{31}$$

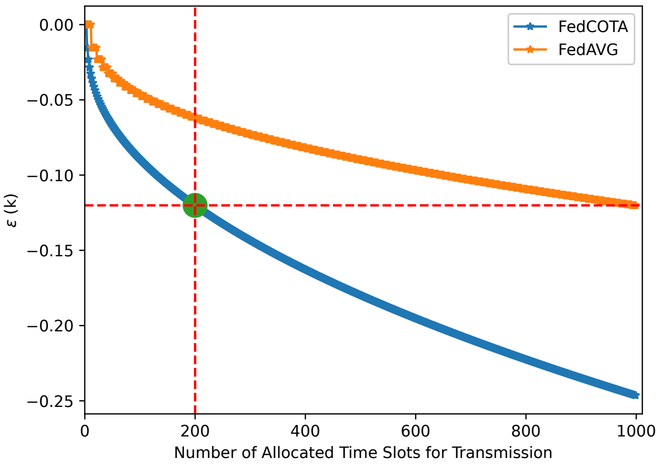

Fig. 2: Comparison of a global logistic regression and federated logistic regression models. While the former is trained over the entire dataset, the latter is trained via FedCOTA and FedAVG.

where $\hat{z}_i^n = S(\theta^T u_i^n)$ is the local estimate of the label z_i^n by the i-th agent, $S(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function, $||\theta||_2$ is the L_2-norm of the parameter vector θ, and λ is the regularization hyper-parameter, where $\lambda = 0$ means no regularization while $\lambda = 1$ represents maximum level of regularization. Note that $\lambda = 0.0001$ has been chosen in simulations, and the overall cost function given in (31) is strongly convex. Moreover, since the parameter vector θ is always in a closed and bounded (constraint) set Θ, the second order derivative of (31) is bounded and therefore has a Lipschitz continuous gradient. Additionally, for each agent, the step size is identically chosen as $\eta(k) = \frac{1}{\sqrt{k+1}}$, and the overall cost function is then

$$\mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N} f_i(\theta, d_i) \tag{32}$$

where N is the number of agents in the system. We consider a system of 10 agents, each having a total number of $|D_i| = 100$ training samples. The parameter vector has dimension $m = 3$, i.e., $\theta \in \Re^3$, which also includes a bias term. The objective is to find $\theta^* = \arg\min_{\theta\in\Theta} \mathcal{L}(\theta)$, where $\Theta = \{\theta \in \Re^3 |\ ||\theta(k)|| \le 15, \forall k \ge 0\}$ is the constraint set.

In order to obtain a desired parameter vector θ_d, we have trained a global logistic regression model, which has access to the entire dataset. We would expect that the parameter vector $\theta(k)$ tends to converge θ_d. This can be assessed by computing $\varepsilon(k) = \log_{10}\left(\frac{\theta(k)-\theta_d}{\theta(0)-\theta_d}\right)$ for each communication round $k \ge 0$. As it can be seen in Fig. 2, the parameter vectors of agents utilizing the FedCOTA algorithm tend to converge to θ_d.

Note that Fig. 2 also includes a comparison of the FedCOTA algorithm and the standard FedAVG algorithm [5], for which the TDMA scheme is used for communication between agents and the central unit. In this case, individual time slots are allocated for each agent to transmit its parameter vector at each communication round. After receiving parameter vectors, the server computes the average of them and sends it back to the agents. Since we consider a system with $N = 10$ agents, it takes 10 time slots per communication round for each agent to transmit its updated parameter vector while only 2 are needed for the FedaOTA algorithm (see (7) and (8)), which makes it 5 times faster than the FedAVG algorithm (see Fig. 2).

V. Conclusion

In this paper we have investigated the federated learning problem via Over-the-Air Computation, which provides significant improvements in terms of communication efficiency and privacy. We have investigated the convergence properties of the proposed gradient-based algorithm by considering time-varying step sizes. Subsequently, we have presented some numerical examples to illustrate our theoretical results.

Our current research is on the use of stochastic gradient descent, which is more efficient when we have large number of training samples. In addition, future work will also include cases where the communication between agents is fully distributed and data distribution among users are not independent and identically distributed (non-iid).

VI. Appendix: Proof of Lemma 5

Due to Assumption 3, C_k is increasing and nonnegative. Then, we have

$$\sum_{t=0}^{k-2}\Big(\prod_{l=t+1}^{k-1} C_l\Big)\eta^2(t) + \eta^2(k-1) \le \sum_{t=0}^{k-1}(C_{k-1})^{k-t-1}\eta^2(t). \tag{33}$$

Moreover, again by Assumption 3, $\eta(k)$ is decreasing and $\lim_{k\to\infty}\eta(k) = 0$. Thus, for an arbitrarily given $\varepsilon > 0$, there exists a time step $k_0 > 0$ such that $\eta(k) \le \varepsilon$, $\forall k \ge k_0$. Since $\eta(k)$ is decreasing, we have

$$\eta(k) < \eta(k-1) = \frac{1 - C_{k-1}}{\mu}. \tag{34}$$

Multiplying both sides of (34) with $\eta(k)$ provides for all $k \ge k_0$

$$\eta^2(k) < \frac{(1 - C_{k-1})\varepsilon}{\mu}. \tag{35}$$

For $k > k_0 + 1$, the right hand side of (33) can be bounded as

$$\sum_{t=0}^{k-1}(C_{k-1})^{k-t-1}\eta^2(t) \le \sum_{t=0}^{k_0}(C_{k-1})^{k-t-1}\eta^2(t) + \sum_{t=k_0+1}^{k-1}(C_{k-1})^{k-t-1}\eta^2(t). \tag{36}$$

Decreasingness of $\eta(k)$ implies $\eta^2(0) \geq \eta^2(k)$ for $k \geq 0$. From (14), $C_{k-1} \leq 1$ for all $k \geq 1$. Hence, the first term on the right hand side of (36) can be written as

$$\begin{aligned}\sum_{t=0}^{k_0}\big(C_{k-1}\big)^{k-t-1}\eta^2(t) &\leq \eta^2(0)\sum_{t=0}^{k_0}\big(C_{k-1}\big)^{k-t-1}\\ &= \eta^2(0)\sum_{t=0}^{k_0}\big(C_{k-1}\big)^{k-k_0+t-1}\\ &= \eta^2(0)\big(C_{k-1}\big)^{-k_0}\big(C_{k-1}\big)^{k-1}\sum_{t=0}^{k_0}\big(C_{k-1}\big)^{t}\\ &\leq \eta^2(0)\big(C_{k-1}\big)^{-k_0}\big(C_{k-1}\big)^{k-1}(k_0+1),\end{aligned} \tag{37}$$

where $\lim_{k\to\infty}(C_{k-1})^{k-1} = 0$ holds by Lemma 3. Thus, we have

$$\lim_{k\to\infty}\sum_{t=0}^{k_0}\big(C_{k-1}\big)^{k-t-1}\eta^2(t) = 0. \tag{38}$$

In addition, (35) holds for $k > k_0 + 1$, which allows us to find an upper-bound for the second term on the right hand side of (36) as

$$\begin{aligned}\sum_{t=k_0+1}^{k-1}\big(C_{k-1}\big)^{k-t-1}\eta^2(k) &\leq \frac{(1-C_{k-1})\varepsilon}{\mu}\sum_{t=k_0+1}^{k-1}\big(C_{k-1}\big)^{k-t-1}\\ &= \frac{(1-C_{k-1})\varepsilon}{\mu}\left(\frac{1-\big(C_{k-1}\big)^{k-k_0-1}}{1-C_{k-1}}\right)\\ &\leq \frac{\varepsilon}{\mu}.\end{aligned} \tag{39}$$

By using (37)-(39), and taking the limit of both sides of (36) results in

$$\lim_{k\to\infty}\sum_{t=0}^{k-1}\big(C_{k-1}\big)^{k-t-1}\eta(t) \leq \frac{\varepsilon}{\mu}. \tag{40}$$

Since ε is arbitrary, we complete the proof.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[3] A. Nedic, "Distributed gradient methods for convex machine learning problems in networks: Distributed optimization," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 92–101, 2020.

[4] J. Chen and X. Ran, "Deep learning with edge computing: A review," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1655–1674, 2019.

[5] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[6] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," *Advances in neural information processing systems*, vol. 30, 2017.

[7] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[8] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečnỳ, S. Mazzocchi, B. McMahan *et al.*, "Towards federated learning at scale: System design," *Proceedings of Machine Learning and Systems*, vol. 1, pp. 374–388, 2019.

[9] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," *arXiv preprint arXiv:1710.06963*, 2017.

[10] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[11] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[12] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography conference*. Springer, 2006, pp. 265–284.

[13] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 1175–1191.

[14] 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.

[15] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "Qsgd: Communication-efficient sgd via gradient quantization and encoding," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[16] D. Alistarh, T. Hoefler, M. Johansson, N. Konstantinov, S. Khirirat, and C. Renggli, "The convergence of sparsified gradient methods," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[17] A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, and R. Pedarsani, "Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 2021–2031.

[18] M. Chen, N. Shlezinger, H. V. Poor, Y. C. Eldar, and S. Cui, "Communication-efficient federated learning," *Proceedings of the National Academy of Sciences*, vol. 118, no. 17, 2021.

[19] A. Mitra, R. Jaafar, G. Pappas, and H. Hassani, "Linear convergence in federated learning: Tackling client heterogeneity and sparse gradients," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[20] F. Molinari, N. Agrawal, S. Stańczak, and J. Raisch, "Max-consensus over fading wireless channels," *IEEE Transactions on Control of Network Systems*, vol. 8, no. 2, pp. 791–802, 2021.

[21] M. Frey, I. Bjelaković, and S. Stańczak, "Over-the-air computation in correlated channels," *IEEE Transactions on Signal Processing*, vol. 69, pp. 5739–5755, 2021.

[22] G. Zhu, J. Xu, K. Huang, and S. Cui, "Over-the-air computing for wireless data aggregation in massive iot," *IEEE Wireless Communications*, vol. 28, no. 4, pp. 57–65, 2021.

[23] K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated learning via over-the-air computation," *IEEE Transactions on Wireless Communications*, vol. 19, no. 3, pp. 2022–2035, 2020.

[24] M. M. Amiri and D. Gündüz, "Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air," *IEEE Transactions on Signal Processing*, vol. 68, pp. 2155–2169, 2020.

[25] T. Sery, N. Shlezinger, K. Cohen, and Y. C. Eldar, "Over-the-air federated learning from heterogeneous data," *IEEE Transactions on Signal Processing*, vol. 69, pp. 3796–3811, 2021.

[26] R. Ahlswede, "Multi-way communication channels," in *Second International Symposium on Information Theory: Tsahkadsor, Armenia, USSR, Sept. 2-8, 1971*, 1973.

[27] A. Giridhar and P. Kumar, "Toward a theory of in-network computation in wireless sensor networks," *IEEE Communications Magazine*, vol. 44, no. 4, pp. 98–107, apr 2006.

[28] F. Molinari, N. Agrawal, S. Stańczak, and J. Raisch, "Over-the-air max-consensus in clustered networks adopting half-duplex communication technology," *IEEE Transactions on Control of Network Systems*, 2022.

[29] W. Rudin *et al.*, *Principles of mathematical analysis*. McGraw-hill New York, 1976, vol. 3.

[30] K. Yuan, Q. Ling, and W. Yin, "On the convergence of decentralized gradient descent," *SIAM Journal on Optimization*, vol. 26, no. 3, pp. 1835–1854, 2016.

[31] C. Xi, V. S. Mai, R. Xin, E. H. Abed, and U. A. Khan, “Linear convergence in optimization over directed graphs with row-stochastic matrices,” *IEEE Transactions on Automatic Control*, vol. 63, no. 10, pp. 3558–3565, 2018.

[32] R. Xin, S. Pu, A. Nedić, and U. A. Khan, “A general framework for decentralized optimization with first-order methods,” *Proceedings of the IEEE*, vol. 108, no. 11, pp. 1869–1889, 2020.

[33] V. S. Mai and E. H. Abed, “Distributed optimization over weighted directed graphs using row stochastic matrix,” in *2016 American Control Conference (ACC)*. IEEE, 2016, pp. 7165–7170.

[34] H. Li, Q. Lü, and T. Huang, “Distributed projection subgradient algorithm over time-varying general unbalanced directed graphs,” *IEEE Transactions on Automatic Control*, vol. 64, no. 3, pp. 1309–1316, 2018.

[35] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*. Springer Science & Business Media, 2003, vol. 87.

[36] A. Nedić and A. Olshevsky, “Distributed optimization over time-varying directed graphs,” *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 601–615, 2014.

[37] T. Yang, X. Yi, J. Wu, Y. Yuan, D. Wu, Z. Meng, Y. Hong, H. Wang, Z. Lin, and K. H. Johansson, “A survey of distributed optimization,” *Annual Reviews in Control*, vol. 47, pp. 278–305, 2019.

[38] P. Wang, P. Lin, W. Ren, and Y. Song, “Distributed subgradient-based multiagent optimization with more general step sizes,” *IEEE Transactions on Automatic Control*, vol. 63, no. 7, pp. 2295–2302, 2017.

[39] P. Xie, K. You, R. Tempo, S. Song, and C. Wu, “Distributed convex optimization with inequality constraints over time-varying unbalanced digraphs,” *IEEE Transactions on Automatic Control*, vol. 63, no. 12, pp. 4331–4337, 2018.

[40] V. S. Mai and E. H. Abed, “Distributed optimization over directed graphs with row stochasticity and constraint regularity,” *Automatica*, vol. 102, pp. 94–104, 2019.

[41] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge University Press, 2012.

[42] A. F. Molisch, *Wireless communications*. John Wiley & Sons, 2012.

[43] B. Sklar, “Rayleigh fading channels in mobile digital communication systems. i. characterization,” *IEEE Communications magazine*, vol. 35, no. 7, pp. 90–100, 1997.

[44] D. Bertsekas, A. Nedic, and A. Ozdaglar, *Convex analysis and optimization*. Athena Scientific, 2003, vol. 1.