Federated Deep AUC Maximization for Heterogeneous Data with a Constant Communication Complexity

Zhuoning Yuan [*1] **Zhishuai Guo** [*1] **Yi Xu** [2] **Yiming Ying** [3] **Tianbao Yang** [1]

Abstract

Deep AUC (area under the ROC curve) Maximization (DAM) has attracted much attention recently due to its great potential for imbalanced data classification. However, the research on Federated Deep AUC Maximization (FDAM) is still limited. Compared with standard federated learning (FL) approaches that focus on decomposable minimization objectives, FDAM is more complicated due to its minimization objective is non-decomposable over individual examples. In this paper, we propose improved FDAM algorithms for heterogeneous data by solving the popular non-convex strongly-concave min-max formulation of DAM in a distributed fashion, which can also be applied to a class of non-convex strongly-concave min-max problems. A striking result of this paper is that the communication complexity of the proposed algorithm is a constant independent of the number of machines and also independent of the accuracy level, which improves an existing result by orders of magnitude. The experiments have demonstrated the effectiveness of our FDAM algorithm on benchmark datasets, and on medical chest X-ray images from different organizations. Our experiment shows that the performance of FDAM using data from multiple hospitals can improve the AUC score on testing data from a single hospital for detecting life-threatening diseases based on chest radiographs.

1. Introduction

Federated learning (FL) is an emerging paradigm for large-scale learning to deal with data that are (geographically) distributed over multiple clients, e.g., mobile phones, organizations. An important feature of FL is that the data remains at its own clients, allowing the preservation of data privacy. This feature makes FL attractive not only to internet companies such as Google and Apple but also to conventional industries such as those that provide services to hospitals and banks in the big data era (Rieke et al., 2020; Long et al., 2020). Data in these industries is usually collected from people who are concerned about data leakage. But in order to provide better services, large-scale machine learning from diverse data sources is important for addressing model bias. For example, most patients in hospitals located in urban areas could have dramatic differences in demographic data, lifestyles, and diseases from patients who are from rural areas. Machine learning models (in particular, deep neural networks) trained based on patients' data from one hospital could dramatically bias towards its major population, which could bring serious ethical concerns (Pooch et al., 2020).

One of the fundamental issues that could cause model bias is data imbalance, where the number of samples from different classes are skewed. Although FL provides an effective framework for leveraging multiple data sources, *most existing FL methods still lack the capability to tackle the model bias caused by data imbalance*. The reason is that most existing FL methods are developed for minimizing the conventional objective function, e.g., the average of a standard loss function on all data, which are not amenable to optimizing more suitable measures such as area under the ROC curve (AUC) for imbalanced data. It has been recently shown that directly maximizing AUC for deep learning can lead to great improvements on real-world difficult classification tasks (Yuan et al., 2020b). For example, Yuan et al. (2020b) reported the best performance by DAM on the Stanford CheXpert Competition for interpreting chest X-ray images like radiologists (Irvin et al., 2019).

However, the research on FDAM is still limited. To the best of our knowledge, Guo et al. (2020a) is the only work that was dedicated to FDAM by solving **the non-convex strongly-concave min-max** problem in a distributed manner. Their algorithm (CODA) is similar to the standard FedAvg method (McMahan et al., 2017) except that the periodic averaging is applied both to the primal and the

[*]Equal contribution [1]Department of Computer Science, University of Iowa [2]Machine Intelligence Technology, Alibaba Group [3]Department of Mathematics and Statistics, State University of New York at Albany. Correspondence to: Tianbao Yang <tianbao-yang@uiowa.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

Table 1. The summary of sample and communication complexities of different algorithms for FDAM under a μ-PL condition in both heterogeneous and homogeneous settings, where K is the number of machines and $\mu \leq 1$. NPA denotes the naive parallel (large mini-batch) version of PPD-SG (Liu et al., 2020) for DAM, where M denotes the batch size in the NPA. The $*$ indicate the results that are derived by us. $\widetilde{O}(\cdot)$ suppresses a logarithmic factor.

	Heterogeneous Data	Homogeneous Data	Sample Complexity
NPA ($M < \frac{1}{K\mu\epsilon}$)	$\widetilde{O}\left(\frac{1}{KM\mu^2\epsilon} + \frac{1}{\mu\epsilon}\right)$	$\widetilde{O}\left(\frac{1}{KM\mu^2\epsilon} + \frac{1}{\mu\epsilon}\right)$	$\widetilde{O}\left(\frac{M}{\mu\epsilon} + \frac{1}{\mu^2 K\epsilon}\right)$
NPA ($M \geq \frac{1}{K\mu\epsilon}$)	$\widetilde{O}\left(\frac{1}{\mu}\right)^*$	$\widetilde{O}\left(\frac{1}{\mu}\right)^*$	$\widetilde{O}\left(\frac{M}{\mu}\right)^*$
CODA+ (CODA)	$\widetilde{O}\left(\frac{K}{\mu} + \frac{1}{\mu\epsilon^{1/2}} + \frac{1}{\mu^{3/2}\epsilon^{1/2}}\right)$	$\widetilde{O}\left(\frac{K}{\mu}\right)^*$	$\widetilde{O}\left(\frac{1}{\mu\epsilon} + \frac{1}{\mu^2 K\epsilon}\right)$
CODASCA	$\widetilde{O}\left(\frac{1}{\mu}\right)$	$\widetilde{O}\left(\frac{1}{\mu}\right)$	$\widetilde{O}\left(\frac{1}{\mu\epsilon} + \frac{1}{\mu^2 K\epsilon}\right)$

dual variables. Nevertheless, their results on FDAM are not comprehensive. By a deep investigation of their algorithms and analysis, we found that (i) although their FL algorithm CODA was shown to be better than the naive parallel algorithm (NPA) with a small mini-batch for DAM, the NPA using a larger mini-batch at local machines can enjoy a smaller communication complexity than CODA; (ii) the communication complexity of CODA for homogeneous data becomes better than that was established for the heterogeneous data, but is still worse than that of NPA with a large mini-batch at local clients. These shortcomings of CODA for FDAM motivate us to develop better federated averaging algorithms and analysis with a better communication complexity without sacrificing the sample complexity.

This paper aims to provide more comprehensive results for FDAM, with a focus on improving the communication complexity of CODA for heterogeneous data. In particular, our contributions are summarized below:

- First, we provide a stronger baseline with a simpler algorithm than CODA named CODA+, and establish its complexity in both homogeneous and heterogeneous data settings. Although CODA+ has a slight change from CODA, its analysis is much more involved than that of CODA, which is based on the duality gap analysis instead of the primal objective gap analysis.
- Second, we propose a new variant of CODA+ named CODASCA with a much improved communication complexity than CODA+. The key thrust is to incorporate the idea of stochastic controlled averaging of SCAFFOLD (Karimireddy et al., 2020) into the framework of CODA+ to correct the client-drift for both local primal updates and local dual updates. A striking result of CODASCA under a PL condition for deep learning is that its communication complexity is independent of the number of machines and the targeted accuracy level, which is even better than CODA+ in the homogeneous data setting. The analysis of CODASCA is also non-trivial that combines the duality gap analysis of CODA+ for a non-convex strongly-concave min-max problem and the variance reduction analysis of SCAFFOLD. The comparison between CODASCA and CODA+ and the NPA for FDAM is shown in Table 1.
- Third, we conduct experiments on benchmark datasets to verify our theory by showing CODASCA can enjoy a larger communication window size than CODA+ without sacrificing the performance. Moreover, we conduct empirical studies on medical chest X-ray images from different hospitals by showing that the performance of CODASCA using data from multiple organizations can improve the performance on testing data from a single hospital.

2. Related Work

Federated Learning (FL). Many empirical studies (Povey et al., 2014; Su & Chen, 2015; McMahan et al., 2016; Chen & Huo, 2016; Lin et al., 2020b; Kamp et al., 2018; Yuan et al., 2020a) have shown that FL exhibits good empirical performance for distributed deep learning. For a more thorough survey of FL, we refer the readers to (McMahan et al., 2019). This paper is closely related to recent studies on the design of distributed stochastic algorithms for FL with provable convergence guarantee.

The most popular FL algorithm is Federated Averaging (FedAvg) (McMahan et al., 2017), also referred to as local SGD (Stich, 2019). Stich (2019) is the first that establishes the convergence of local SGD for strongly convex functions. Yu et al. (2019b;a) establishes the convergence of local SGD and their momentum variants for non-convex functions. The analysis in (Yu et al., 2019b) has exhibited the difference of communication complexities of local SGD in homogeneous and heterogeneous data settings, which is also discovered in recent works (Khaled et al., 2020; Woodworth et al., 2020b;a). These latter studies provide a tight analysis of local SGD in homogeneous and/or heterogeneous data settings, improving its upper bounds for convex functions and strongly convex functions than some earlier works, which sometimes improve over large mini-batch SGD, e.g., when the level of heterogeneity is sufficiently small.

Haddadpour et al. (2019) improve the complexities of local SGD for non-convex optimization by leveraging the Polyak-Łojasiewicz (PL) condition. (Karimireddy et al., 2020) propose a new FedAvg algorithm SCAFFOLD by introducing control variates (variance reduction) to correct

for the 'client-drift' in the local updates for heterogeneous data. The communication complexities of SCAFFOLD are no worse than that of large mini-batch SGD for both strongly convex and non-convex functions. The proposed algorithm CODASCA is inspired by the idea of stochastic controlled averaging of SCAFFOLD. However, the analysis of CODASCA for non-convex min-max optimization under a PL condition of the primal objective function is non-trivial compared to that of SCAFFOLD.

AUC Maximization. This work builds on the foundations of stochastic AUC maximization developed in many previous works. Ying et al. (2016) address the scalability issue of optimizing AUC by introducing a min-max reformulation of the AUC square surrogate loss and solving it by a convex-concave stochastic gradient method (Nemirovski et al., 2009). Natole et al. (2018) improve the convergence rate by adding a strongly convex regularizer into the original formulation. Based on the same min-max formulation as in (Ying et al., 2016), Liu et al. (2018) achieve an improved convergence rate by developing a multi-stage algorithm by leveraging the quadratic growth condition of the problem. However, all of these studies focus on learning a linear model, whose corresponding problem is convex and strongly concave. Yuan et al. (2020b) propose a more robust margin-based surrogate loss for the AUC score, which can be formulated as a similar min-max problem to the AUC square surrogate loss.

Deep AUC Maximization (DAM). (Rafique et al., 2018) is the first work that develops algorithms and convergence theories for weakly convex and strongly concave min-max problems, which is applicable to DAM. However, their convergence rate is slow for a practical purpose. Liu et al. (2020) consider improving the convergence rate for DAM under a practical PL condition of the primal objective function. Guo et al. (2020b) further develop more generic algorithms for non-convex strongly-concave min-max problems, which can also be applied to DAM. There are also several studies (Yan et al., 2020; Lin et al., 2020a; Luo et al., 2020; Yang et al., 2020) focusing on non-convex strongly concave min-max problems without considering the application to DAM. Based on Liu et al. (2020)'s algorithm, Guo et al. (2020a) propose a communication-efficient FL algorithm (CODA) for DAM. However, its communication cost is still high for heterogeneous data.

DL for Medical Image Analysis. In past decades, machine learning, especially deep learning methods have revolutionized many domains such as machine vision, natural language processing. For medical image analysis, deep learning methods are also showing great potential such as in classification of skin lesions (Esteva et al., 2017; Li & Shen, 2018), interpretation of chest radiographs (Ardila et al., 2019; Irvin et al., 2019), and breast cancer screening (Bejnordi et al., 2017; McKinney et al., 2020; Wang et al., 2016). Some works have already achieved expert-level performance in different tasks (Ardila et al., 2019; McKinney et al., 2020; Litjens et al., 2017). Recently, Yuan et al. (2020b) employ DAM for medical image classification and achieve great success on two challenging tasks, namely CheXpert competition for chest X-ray image classification and Kaggle competition for melanoma classification based on skin lesion images. However, to the best of our knowledge, the application of FDAM methods on medical datasets from different hospitals have not be thoroughly investigated.

3. Preliminaries and Notations

We consider federated learning of deep neural networks by maximizing the AUC score. The setting is the same to that was considered as in (Guo et al., 2020a). Below, we present some preliminaries and notations, which are mostly the same as in (Guo et al., 2020a). In this paper, we consider the following min-max formulation for distributed problem:

$$\min_{\substack{\mathbf{w}\in\mathbb{R}^d\\(a,b)\in\mathbb{R}^2}} \max_{\alpha\in\mathbb{R}} f(\mathbf{w},a,b,\alpha) = \frac{1}{K}\sum_{k=1}^{K} f_k(\mathbf{w},a,b,\alpha), \quad (1)$$

where K is the total number of machines. This formulation covers a class of non-convex strongly concave min-max problems and specifically for the AUC maximization, $f_k(\mathbf{w},a,b,\alpha)$ is defined below.

$$\begin{aligned} f_k(\mathbf{w},a,b,\alpha) &= \mathbb{E}_{\mathbf{z}^k}[F_k(\mathbf{w},a,b,\alpha;\mathbf{z}^k)] \\ &= \mathbb{E}_{\mathbf{z}^k}\big[(1-p)(h(\mathbf{w};\mathbf{x}^k)-a)^2\mathbb{I}_{[y^k=1]} \\ &\quad + p(h(\mathbf{w};\mathbf{x}^k)-b)^2\mathbb{I}_{[y^k=-1]} \\ &\quad + 2(1+\alpha)(ph(\mathbf{w};\mathbf{x}^k)\mathbb{I}_{[y^k=-1]} \\ &\quad - (1-p)h(\mathbf{w},\mathbf{x}^k)\mathbb{I}_{[y^k=1]}) - p(1-p)\alpha^2\big]. \end{aligned} \quad (2)$$

where $\mathbf{z}^k = (\mathbf{x}^k, y^k) \sim \mathbb{P}_k$, $\mathbb{P}_k$ is the data distribution on machine k, p is the ratio of positive data. When $\phi_k = \phi_l, \forall k \neq l$, this is referred to as the homogeneous data setting; otherwise heterogeneous data setting.

Notations. We define the following notations:

$$\begin{aligned} &\mathbf{v} = (\mathbf{w}^T, a, b)^T, \quad \phi(\mathbf{v}) = \max_{\alpha} f(\mathbf{v},\alpha), \\ &\phi_s(\mathbf{v}) = \phi(\mathbf{v}) + \frac{1}{2\gamma}\|\mathbf{v}-\mathbf{v}_{s-1}\|^2, \\ &f^s(\mathbf{v},\alpha) = f(\mathbf{v},\alpha) + \frac{1}{2\gamma}\|\mathbf{v}-\mathbf{v}_{s-1}\|^2 \\ &F_k^s(\mathbf{v},\alpha;\mathbf{z}_k) = F_k(\mathbf{v},\alpha;\mathbf{z}_k) + \frac{1}{2\gamma}\|\mathbf{v}-\mathbf{v}_{s-1}\|^2 \\ &\mathbf{v}_\phi^* = \arg\min_{\mathbf{v}} \phi(\mathbf{v}), \quad \mathbf{v}_{\phi_s}^* = \arg\min_{\mathbf{v}} \phi_s(\mathbf{v}). \end{aligned}$$

Assumptions. Similar to (Guo et al., 2020a), we make the following assumptions throughout this paper.

Assumption 1.
(i) There exist $\mathbf{v}_0, \Delta_0 > 0$ *such that* $\phi(\mathbf{v}_0) - \phi(\mathbf{v}_\phi^*) \leq \Delta_0$. *(ii) PL condition:* $\phi(\mathbf{v})$ *satisfies the* μ*-PL condition, i.e.,* $\mu(\phi(\mathbf{v}) - \phi(\mathbf{v}_*)) \leq \frac{1}{2}\|\nabla\phi(\mathbf{v})\|^2$*; (iii) Smoothness: For any* $\mathbf{z}$*,* $f(\mathbf{v}, \alpha; \mathbf{z})$ *is* ℓ*-smooth in* $\mathbf{v}$ *and* α*.* $\phi(\mathbf{v})$ *is* L*-smooth, i.e.,* $\|\nabla\phi(\mathbf{v}_1) - \nabla\phi(\mathbf{v}_2)\| \leq L\|\mathbf{v}_1 - \mathbf{v}_2\|$.
(iv) Bounded variance:

$$\begin{aligned}\mathbb{E}[\|\nabla_{\mathbf{v}} f_k(\mathbf{v}, \alpha) - \nabla_{\mathbf{v}} F_k(\mathbf{v}, \alpha; \mathbf{z})\|^2] \leq \sigma^2 \\ \mathbb{E}[|\nabla_{\alpha} f_k(\mathbf{v}, \alpha) - \nabla_{\alpha} F_k(\mathbf{v}, \alpha; \mathbf{z})|^2] \leq \sigma^2.\end{aligned} \tag{3}$$

To quantify the drifts between different clients, we introduce the following assumption.

Assumption 2. *Bounded client drift:*

$$\begin{aligned}\frac{1}{K}\sum_{k=1}^{K}\|\nabla_{\mathbf{v}} f_k(\mathbf{v}, \alpha) - \nabla_{\mathbf{v}} f(\mathbf{v}, \alpha)\|^2 \leq D^2 \\ \frac{1}{K}\sum_{k=1}^{K}\|\nabla_{\alpha} f_k(\mathbf{v}, \alpha) - \nabla_{\alpha} f(\mathbf{v}, \alpha)\|^2 \leq D^2.\end{aligned} \tag{4}$$

Remark. D quantifies the drift between the local objectives and the global objective. $D = 0$ denotes the homogeneous data setting that all the local objectives are identical. $D > 0$ corresponds to the heterogeneous data setting.

4. CODA+: A stronger baseline

In this section, we present a stronger baseline than CODA (Guo et al., 2020a). The motivation is that (i) the CODA algorithm uses a step to compute the dual variable from the primal variable by using sampled data from all clients; but we find this step is unnecessary by an improved analysis; (ii) the complexity of CODA for homogeneous data is not given in its original paper. Hence, CODA+ is a simplified version of CODA but with much refined analysis.

We present the steps of CODA+ in Algorithm 1. It is similar to CODA that uses stagewise updates. In s-th stage, a strongly convex strongly concave subproblem is constructed:

$$\min_{\mathbf{v}}\max_{\alpha} f(\mathbf{v}, \alpha) + \frac{\gamma}{2}\|\mathbf{v} - \mathbf{v}_0^s\|^2, \tag{5}$$

where $\mathbf{v}_0^s$ is the output of the previous stage.

CODA+ improves upon CODA in two folds. First, CODA+ algorithm is more concise since the output primal and dual variables of each stage can be directly used as input for the next stage, while CODA needs an extra large batch of data after each stage to compute the dual variable. This modification not only reduces the sample complexity, but also makes the algorithm applicable to a boarder family of nonconvex min-max problems. Second, CODA+ has a smaller communication complexity for homogeneous data than that for heterogeneous data while the previous analysis of CODA yields the same communication complexity for homogeneous data and heterogeneous data.

Algorithm 1 CODA+

Initialization: $(\mathbf{v}_0, \alpha_0, \gamma)$.
for $s = 1, ..., S$ **do**
  $\mathbf{v}_s, \alpha_s = \text{DSG+}(\mathbf{v}_{s-1}, \alpha_{s-1}, \eta_s, I_s, \gamma)$;
end for
Return $\mathbf{v}_S, \alpha_S$.

Algorithm 2 DSG+$(\mathbf{v}_0, \alpha_0, \eta, T, I, \gamma)$

Each machine does initialization: $\mathbf{v}_0^k = \mathbf{v}_0, \alpha_0^k = \alpha_0$,
for $t = 0, 1, ..., T-1$ **do**
  Each machine k updates its local solution in parallel:
  $\mathbf{v}_{t+1}^k = \mathbf{v}_t^k - \eta(\nabla_{\mathbf{v}} F_k(\mathbf{v}_t^k, \alpha_t^k; \mathbf{z}_t^k) + \gamma(\mathbf{v}_t^k - \mathbf{v}_0))$,
  $\alpha_{t+1}^k = \alpha_t^k + \eta\nabla_{\alpha} F_k(\mathbf{v}_t^k, \alpha_t^k; \mathbf{z}_t^k)$,
  if $t+1 \bmod I = 0$ **then**
    $\mathbf{v}_{t+1}^k = \frac{1}{K}\sum_{k=1}^{K}\mathbf{v}_{t+1}^k$, ◇ communicate
    $\alpha_{t+1}^k = \frac{1}{K}\sum_{k=1}^{K}\alpha_{t+1}^k$, ◇ communicate
  end if
end for
Return $\left(\bar{\mathbf{v}} = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{T}\sum_{t=1}^{T}\mathbf{v}_t^k, \bar{\alpha} = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{T}\sum_{t=1}^{T}\alpha_t^k\right)$.

We have the following lemma to bound the convergence for the subproblem in each s-th stage.

Lemma 1. *(One call of Algorithm 2) Let* $(\bar{\mathbf{v}}, \bar{\alpha})$ *be the output of Algorithm 2. Suppose Assumption 1 and 2 hold. By running Algorithm 2 with given input* $\mathbf{v}_0, \alpha_0$ *for* T *iterations,* $\gamma = 2\ell$*, and* $\eta \leq \min(\frac{1}{3\ell + 3\ell^2/\mu_2}, \frac{1}{4\ell})$*, we have for any* $\mathbf{v}$ *and* α

$$\begin{aligned}\mathbb{E}[f^s(\bar{\mathbf{v}}, \alpha) - f^s(\mathbf{v}, \bar{\alpha})] \leq \frac{1}{\eta T}\|\mathbf{v}_0 - \mathbf{v}\|^2 + \frac{1}{\eta T}(\alpha_0 - \alpha)^2 \\ + \underbrace{\left(\frac{3\ell^2}{2\mu_2} + \frac{3\ell}{2}\right)(12\eta^2 I\sigma^2 + 36\eta^2 I^2 D^2)\mathbb{I}_{I>1}}_{A_1} + \frac{3\eta\sigma^2}{K},\end{aligned}$$

where $\mu_2 = 2p(1-p)$ *is the strong concavity coefficient of* $f(\mathbf{v}, \alpha)$ *in* α.

Remark. Note that the term A_1 on the RHS is the drift of clients caused by skipping communication. When $D = 0$, i.e., the machines have homogeneous data distribution, we need $\eta I = O\left(\frac{1}{K}\right)$, then A_1 can be merged with the last term. When $D > 0$, we need $\eta I^2 = O\left(\frac{1}{K}\right)$, which means that I has to be smaller in heterogeneous data setting and thus the communication complexity is higher.

Remark. The key difference between the analysis of CODA+ and that of CODA lies at how to handle the term $(\alpha_0 - \alpha)^2$ in Lemma 1. In CODA, the initial dual variable α_0 is computed from the initial primal variable $\mathbf{v}_0$, which reduces the error term $(\alpha_0-\alpha)^2$ to one similar to $\|\mathbf{v}_0-\mathbf{v}\|^2$, which is then bounded by the primal objective gap due to the PL condition. However, since we do not conduct the extra computation of α_0 from $\mathbf{v}_0$, our analysis directly deals with such error term by using the duality gap of f^s. This technique is originally developed by (Yan et al., 2020).

Theorem 1. *Define* $\hat{L} = L + 2\ell, c = \frac{\mu/\hat{L}}{5+\mu/\hat{L}}$. *Set* $\gamma = 2\ell$, $\eta_s = \eta_0 \exp(-(s-1)c)$, $T_s = \frac{212}{\eta_0 \min(\ell,\mu_2)} \exp((s-1)c)$. *To return* $\mathbf{v}_S$ *such that* $\mathbb{E}[\phi(\mathbf{v}_S) - \phi(\mathbf{v}^*_\phi)] \leq \epsilon$, *it suffices to choose* $S \geq O\left(\frac{5\hat{L}+\mu}{\mu} \max\left\{\log\left(\frac{2\Delta_0}{\epsilon}\right), \log S + \log\left[\frac{2\eta_0}{\epsilon}\frac{12(\sigma^2)}{5K}\right]\right\}\right)$. *The iteration complexity is* $\widetilde{O}\left(\max\left(\frac{\Delta_0}{\mu\epsilon\eta_0 K}, \frac{\hat{L}}{\mu^2 K\epsilon}\right)\right)$ *and the communication complexity is* $\widetilde{O}\left(\frac{K}{\mu}\right)$ *by setting* $I_s = \Theta(\frac{1}{K\eta_s})$ *if* $D = 0$, *and is* $\widetilde{O}\left(\max\left(\frac{K}{\mu} + \frac{\Delta_0^{1/2}}{\mu(\eta_0\epsilon)^{1/2}}, \frac{K}{\mu} + \frac{\hat{L}^{1/2}}{\mu^{3/2}\epsilon^{1/2}}\right)\right)$ *by setting* $I_s = \Theta(\frac{1}{\sqrt{K\eta_s}})$ *if* $D > 0$, *where* $\widetilde{O}$ *suppresses logarithmic factors.*

Remark. Due to the PL condition, the step size η decreases geometrically. Accordingly, I increases geometrically due to Lemma 1, and I increases with a faster rate when the data are homogeneous than that when data are heterogeneous. In result, the total number of communications in homogeneous setting is much less than that in heterogeneous setting.

5. CODASCA

Although CODA+ has a highly reduced communication complexity for homogeneous data, it is still suffering from a high communication complexity for heterogeneous data. Even for the homogeneous data, CODA+ has a worse communication complexity with a dependence on the number of clients K than the NPA algorithm with a large batch size.

Can we further reduce the communication complexity for FDAM for both homogeneous and heterogeneous data without using a large batch size?

The main reason for the degeneration in the heterogeneous data setting is the data difference. Even at global optimum $(\mathbf{v}_*, \alpha_*)$, the gradient of local functions in different clients could be different and non-zero. In the homogeneous data setting, different clients still produce different solutions due to stochastic error (cf. the $\eta^2\sigma^2 I$ term of A_1 in Lemma 1). These together contribute to the client drift.

To correct the client drift, we propose to leverage the idea of stochastic controlled averaging due to (Karimireddy et al., 2020). The key idea is to maintain and update a control variate to accommodate the client drift, which is taken into account when updating the local solutions. In the proposed algorithm CODASCA, we apply control variates to both primal and dual variables. CODASCA shares the same stagewise framework as CODA+, where a strongly convex strongly concave subproblem is constructed and optimized in a distributed fashion approximatly in each stage. The steps of CODASCA are presented in Algorithm 3 and Algorithm 4. Below, we describe the algorithm in each stage.

Each stage has R communication rounds. Between two rounds, there are I local updates, and each machine k does the local updates as

$$\begin{aligned}\mathbf{v}^k_{r,t+1} &= \mathbf{v}^k_{r,t} - \eta_l(\nabla_{\mathbf{v}} F^s_k(\mathbf{v}^k_{r,t}, \alpha^k_{r,t}; \mathbf{z}^t_{r,t}) - c^k_{\mathbf{v}} + c_{\mathbf{v}}) \\ \alpha^k_{r,t+1} &= \alpha^k_{r,t} + \eta_l(\nabla_{\alpha} F^s_k(\mathbf{v}^k_{r,t}, \alpha^k_{r,t}; \mathbf{z}^k_{r,t}) - c^k_{\alpha} + c_{\alpha}),\end{aligned}$$

where $c^k_{\mathbf{v}}, c_{\mathbf{v}}$ are local and global control variates for the primal variable, and c^k_{α}, c_{α} are local and global control variates for the dual variable. Note that $\nabla_{\mathbf{v}} F^s_k(\mathbf{v}^k_{r,t}, \alpha^k_{r,t}; \mathbf{z}^t_{r,t})$ and $\nabla_{\alpha} F^s_k(\mathbf{v}^k_{r,t}, \alpha^k_{r,t}; \mathbf{z}^k_{r,t})$ are unbiased stochastic gradient on local data. However, they are biased estimate of global gradient when data on different clients are heterogeneous. Intuitively, the term $-c^k_{\mathbf{v}} + c_{\mathbf{v}}$ and $-c^k_{\alpha} + c_{\alpha}$ work to correct the local gradients to get closer to the global gradient. They also play a role of reducing variance of stochastic gradients, which is helpful as well to reduce the communication complexity in the homogeneous data setting.

At each communication round, the primal and dual variables on all clients get aggregated, averaged and broadcast to all clients. The control variates c at r-th round get updated as

$$\begin{aligned}c^k_{\mathbf{v}} &= c^k_{\mathbf{v}} - c_{\mathbf{v}} + \frac{1}{I\eta_l}(\mathbf{v}_{r-1} - \mathbf{v}^k_{r,I}) \\ c^k_{\alpha} &= c^k_{\alpha} - c_{\alpha} + \frac{1}{I\eta_l}(\alpha^k_{r,I} - \alpha_{r-1}),\end{aligned} \tag{6}$$

which is equivalent to

$$\begin{aligned}c^k_{\mathbf{v}} &= \frac{1}{I}\sum_{t=1}^{I} \nabla_{\mathbf{v}} f^s_k(\mathbf{v}^k_{r,t}, \alpha^k_{r,t}; \mathbf{z}^k_{r,t}) \\ c^k_{\alpha} &= \frac{1}{I}\sum_{t=1}^{I} \nabla_{\alpha} f^s_k(\mathbf{v}^k_{r,t}, \alpha^k_{r,t}; \mathbf{z}^k_{r,t}).\end{aligned} \tag{7}$$

Notice that they are simply the average of stochastic gradients used in this round. An alternative way to compute the control variates is by computing the stochastic gradient with a large batch of extra samples at each client, but this would bring extra cost and is unnecessary. $c_{\mathbf{v}}$ and c_{α} are averages of $c^k_{\mathbf{v}}$ and c^k_{α} over all clients. After the local primal and dual

Algorithm 3 CODASCA

1: Initialization: $(\mathbf{v}_0, \alpha_0, \gamma)$.
2: **for** $s = 1, ..., S$ **do**
3: $\quad \mathbf{v}_s, \alpha_s = \text{DSGSCA+}(\mathbf{v}_{s-1}, \alpha_{s-1}, \eta_l, \eta_g, I_s, R_s, \gamma)$;
4: **end for**
5: Return $\mathbf{v}_S, \alpha_S$.

variables are averaged, an extrapolation step with $\eta_g > 1$ is performed, which will boost the convergence.

In order to establish the convergence of CODASCA, we first present a key lemma below.

Lemma 2. *(One call of Algorithm 4) Under the same setting as in Theorem 2, with* $\tilde{\eta} = \eta_l\eta_g I \leq \frac{\mu_2}{40\ell^2}$, *for* $\mathbf{v}' = \arg\min_{\mathbf{v}} f^s(\mathbf{v}, \alpha_{\tilde{r}}), \alpha' = \arg\max_{\alpha} f^s(\mathbf{v}_{\tilde{r}}, \alpha)$ *we have*

$$\mathbb{E}[f^s(\mathbf{v}_{\tilde{r}}, \alpha') - f^s(\mathbf{v}', \alpha_{\tilde{r}})] \leq \frac{2}{\eta_l\eta_g T}\|\mathbf{v}_0 - \mathbf{v}'\|^2 + \frac{2}{\eta_l\eta_g T}(\alpha_0 - \alpha')^2 + \underbrace{\frac{10\eta_l\sigma^2}{\eta_g}}_{A_2} + \frac{10\eta_l\eta_g\sigma^2}{K}$$

where $T = I \cdot R$ *is the number of iterations for each stage.*

Remark. Compared the above bound with that in Lemma 1, in particular the term A_2 vs the term A_1, we can see that CODASCA will not be affected by the data heterogeneity $D > 0$, and the stochastic variance is also much reduced. As will seen in the next theorem, the value of $\tilde{\eta}$ and R will keep the same in all stages. Therefore, by decreasing local step size η_l geometrically, the communication window size I_s will increase geometrically to ensure $\tilde{\eta} \leq O(1)$.

The convergence result of CODASCA is presented below.

Theorem 2. *Define* $\hat{L} = L + 2\ell$, $c = 4\ell + \frac{248}{53}\hat{L}$. *Set* $\eta_g = \sqrt{K}$, $I_s = I_0 \exp\left(\frac{2\mu_1}{c+2\mu_1}(s-1)\right)$, $R = \frac{1000}{\tilde{\eta}\mu_2}$, $\eta_l^s = \frac{\tilde{\eta}}{\eta_g I_s} = \frac{\tilde{\eta}}{\sqrt{K} I_0}\exp\left(-\frac{2\mu}{c+2\mu}(s-1)\right)$, $\tilde{\eta} \leq \min\{\frac{1}{3\ell + 3\ell^2/\mu_2}, \frac{\mu_2}{40\ell^2}\}$. *After* $S = O(\max\left\{\frac{c+2\mu}{2\mu}\log\frac{4\epsilon_0}{\epsilon}, \frac{c+2\mu}{2\mu}\log\frac{160\hat{L}S}{(c+2\mu)\epsilon}\frac{\tilde{\eta}\sigma^2}{KI_0}\right\})$ *stages, the output* $\mathbf{v}_S$ *satisfies* $\mathbb{E}[\phi(\mathbf{v}_S) - \phi(\mathbf{v}_\phi^*)] \leq \epsilon$. *The communication complexity is* $\widetilde{O}\left(\frac{1}{\mu}\right)$. *The iteration complexity is* $\widetilde{O}\left(\max\{\frac{1}{\mu\epsilon}, \frac{1}{\mu^2 K\epsilon}\}\right)$.

Remark. (i) The number of communications is $\widetilde{O}\left(\frac{1}{\mu}\right)$, independent of number of clients K and the accuracy level ϵ. This is a significant improvement over CODA+, which has a communication complexity of $\widetilde{O}\left(K/\mu + 1/(\mu^{3/2}\epsilon^{1/2})\right)$ in heterogeneous setting. Moreover, $\widetilde{O}\left(1/(\mu)\right)$ is a nearly optimal rate up to a logarithmic factor, since $O(1/\mu)$ is the lower bound communication complexity of distributed strongly convex optimization (Karimireddy et al., 2020; Arjevani & Shamir, 2015) and strongly convexity is a stronger condition than the PL condition.

Algorithm 4 DSGSCA+$(\mathbf{v}_0, \alpha_0, \eta_l, \eta_g, I, R, \gamma)$

Each machine does initialization: $\mathbf{v}_{0,0}^k = \mathbf{v}_0, \alpha_{0,0}^k = \alpha_0$, $c_{\mathbf{v}}^k = \mathbf{0}, c_\alpha^k = 0$
for $r = 1, ..., R$ **do**
  for $t = 0, 1, ..., I-1$ **do**
    Each machine k updates its local solution in parallel:
    $\mathbf{v}_{r,t+1}^k = \mathbf{v}_{r,t}^k - \eta_l(\nabla_{\mathbf{v}} F_k^s(\mathbf{v}_{r,t}^k, \alpha_{r,t}^k; \mathbf{z}_{r,t}^k) - c_{\mathbf{v}}^k + c_{\mathbf{v}})$,
    $\alpha_{r,t+1}^k = \alpha_{r,t}^k + \eta_l(\nabla_\alpha F_k^s(\mathbf{v}_{r,t}^k, \alpha_{r,t}^k; \mathbf{z}_{r,t}^k) - c_\alpha^k + c_\alpha)$,
  end for
  $c_{\mathbf{v}}^k = c_{\mathbf{v}}^k - c_{\mathbf{v}} + \frac{1}{I\eta_l}(\mathbf{v}_{r-1} - \mathbf{v}_{r,I}^k)$
  $c_\alpha^k = c_\alpha^k - c_\alpha + \frac{1}{I\eta_l}(\alpha_{r,I}^k - \alpha_{r-1})$
  $c_{\mathbf{v}} = \frac{1}{K}\sum_{k=1}^K c_{\mathbf{v}}^k, c_\alpha = \frac{1}{K}\sum_{k=1}^K c_\alpha^k$ $\quad \diamond$ communicate
  $\mathbf{v}_r = \frac{1}{K}\sum_{k=1}^K \mathbf{v}_{r,I}^k, \alpha_r = \frac{1}{K}\sum_{k=1}^K \alpha_{r,t}^k$ $\quad \diamond$ communicate
  $\mathbf{v}_r = \mathbf{v}_{r-1} + \eta_g(\mathbf{v}_r - \mathbf{v}_{r-1})$,
  $\alpha_r = \alpha_{r-1} + \eta_g(\alpha_r - \alpha_{r-1})$
  Broadcast $\mathbf{v}_r, \alpha_r, c_{\mathbf{v}}, c_\alpha$ $\quad \diamond$ communicate
end for
Return $\mathbf{v}_{\tilde{r}}, \alpha_{\tilde{r}}$ where $\tilde{r}$ is randomly sampled from $1, ..., R$

(ii) Each stage has the same number of communication rounds. However, I_s increases geometrically. Therefore, the number of iterations and samples in a stage increase geometrically. Theoretically, we can also set η_l^s to the same value as the one in the last stage, correspondingly I_s can be set as a fixed large value. But this increases the number of required samples without further speeding up the convergence. Our setting of I_s is a balance between skipping communications and reducing sample complexity. For simplicity, we use the fixed setting of I_s to compare CODASCA and the baseline CODA+ in our experiment to corroborate the theory.

(iii) The local step size η_l of CODASCA decreases similarly as the step size η in CODA+. But $I_s = O(1/(\sqrt{K}\eta_l^s))$ in CODASCA increases faster than that $I_s = O(1/(\sqrt{K}\eta_s))$ in CODA+ on heterogeneous data. It is noticeable that different from CODA+, we do not need Assumption 2 which bounds the client drift, meaning that CODASCA can be applied to optimize the global objective even if local objectives arbitrarily deviate from the global function.

6. Experiments

In this section, we first verify the effectiveness of CODASCA compared to CODA+ on various datasets, including two benchmark datasets, i.e., ImageNet, CIFAR100 (Deng et al., 2009; Krizhevsky et al., 2009) and a constructed large-scale chest X-ray dataset. Then, we demonstrate the effectiveness of FDAM on improving the performance on a single domain (CheXpert) by using data from multiple sources. For notations, K denotes the number of "clients" (# of machines, # of data sources) and I denotes the communication window size. The code used for the experiments are available

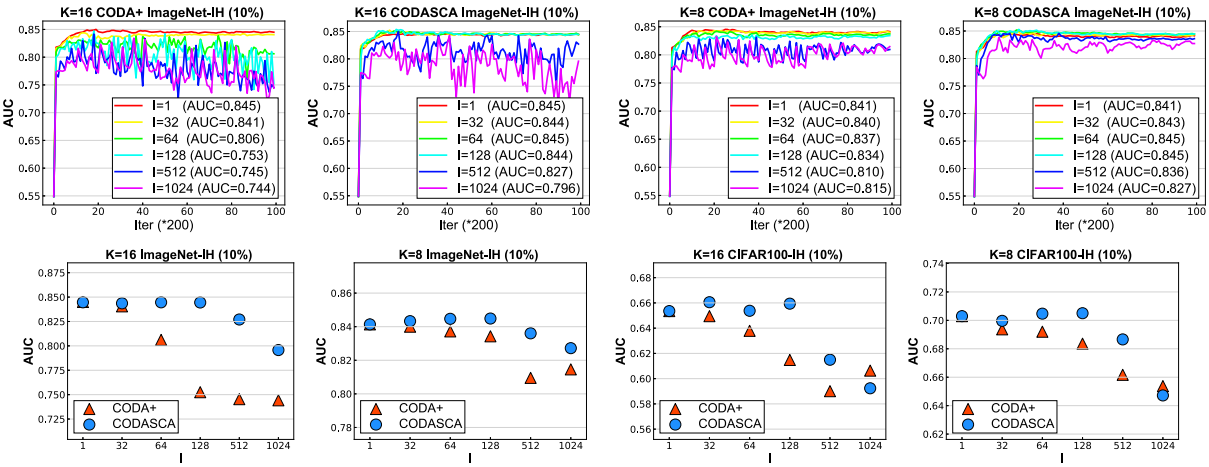

Figure 1. Top row: the testing AUC score of CODASCA vs # of iterations for different values of I on ImageNet-IH and CIFAR100-IH with imratio = 10% and K=16, 8 on Densenet121. Bottom row: the achieved testing AUC vs different values of I for CODASCA and CODA+. The AUC score in the legend in top row figures represent the AUC score at the last iteration.

Table 2. Statistics of Medical Chest X-ray Datasets.

Dataset	Source	Samples
CheXpert	Stanford Hospital (US)	224,316
ChestXray8	NIH Clinical Center (US)	112,120
PadChest	Hospital San Juan (Spain)	110,641
MIMIC-CXR	BIDMC (US)	377,110
ChestXrayAD	H108 and HMUH (Vietnam)	15,000

at https://github.com/yzhuoning/LibAUC.

Chest X-ray datasets. Five medical chest X-ray datasets, i.e., CheXpert, ChestXray14, MIMIC-CXR, PadChest, ChestXray-AD (Irvin et al., 2019; Wang et al., 2017; Johnson et al., 2019; Bustos et al., 2020; Nguyen et al., 2020) are collected from different organizations. The statistics of these medical datasets are summarized in Table 2. We construct five binary classification tasks for predicting five popular diseases, Cardiomegaly (C0), Edema (C1), Consolidation (C2), Atelectasis (C3), P. Effusion (C4), as in CheXpert competition (Irvin et al., 2019). These datasets are naturally imbalanced and heterogeneous due to different patients' populations, different data collection protocols and etc. We refer to the whole medical dataset as ChestXray-IH.

Imbalanced and Heterogeneous (IH) Benchmark Datasets. For benchmark datasets, we manually construct the imbalanced heterogeneous dataset. For ImageNet, we first randomly select 500 classes as positive class and 500 classes as negative class. To increase data heterogeneity, we further split all positive/negative classes into K groups so that each split only owns samples from unique classes without overlapping with that of other groups. To increase data imbalance level, we randomly remove some samples from positive classes for each machine. Please note that due to this operation, the whole sample set for different K is different. We refer to the proportion of positive samples in all samples as imbalance ratio ($imratio$). For CIFAR100, we follow similar steps to construct imbalanced heterogeneous data. We keep the testing/validation set untouched and keep them balanced. For imbalance ratio (imratio), we explore two ratios: 10% and 30%. We refer to the constructed datasets as ImageNet-IH (10%), ImageNet-IH (30%), CIFAR100-IH (10%), CIFAR100-IH (30%). Due to the limited space, we only report imratio=10% with DenseNet121 and defer the other results to supplement.

Parameters and Settings. We train Desenet121 on all datasets. For the parameters in CODASCA/CODA+, we tune $1/\gamma$ in [500, 700, 1000] and η in [0.1, 0.01, 0.001]. For learning rate schedule, we decay the step size by 3 times every T_0 iterations, where T_0 is tuned in [2000, 3000, 4000]. We experiment with a fixed value of I selected from [1, 32, 64, 128, 512, 1024] and we include experiments with increasing I_s in the supplement. We tune η_g in [1.1, 1, 0.99, 0.999]. The local batch size is set to 32 for each machine. We run a total of 20000 iterations for all experiments.

6.1. Comparison with CODA+

We plot the testing AUC on ImageNet (10%) vs # of iterations for CODASCA and CODA+ in Figure 1 (top row) by varying the value of I for different values of K. Results on CIFAR100 are shown in the Supplement. In the bottom row of Figure 1, we plot the achieved testing AUC score vs different values of I for CODASCA and CODA+. We have the following observations:

• **CODASCA enjoys a larger communication window size**. Comparing CODASCA and CODA+ in the bottom panel of Figure 1, we can see that CODASCA enjoys a larger communication window size without hurting the performance than CODA+, which is consistent with our theory.

• **CODASCA is consistently better for different values of K**. We compare the largest value of I such that the performance does not degenerate too much compared with $I = 1$,

Table 3. Performance on ChestXray-IH testing set when K=16.

Method	I	C0	C1	C2	C3	C4
	1	0.8472	0.8499	0.7406	0.7475	0.8688
CODA+	512	0.8361	**0.8464**	0.7356	0.7449	0.8680
CODASCA	512	**0.8427**	0.8457	**0.7401**	**0.7468**	**0.8680**
CODA+	1024	0.8280	**0.8451**	0.7322	0.7431	0.8660
CODASCA	1024	**0.8363**	0.8444	**0.7346**	**0.7481**	**0.8674**

Table 4. Performance of FDAM on Chexpert validation set for DenseNet121.

#of sources	C0	C1	C2	C3	C4	AVG
K=1	0.9007	0.9536	0.9542	0.9090	0.9571	0.9353
K=2	0.9027	0.9586	0.9542	0.9065	0.9583	0.9361
K=3	0.9021	0.9558	**0.9550**	0.9068	0.9583	0.9356
K=4	0.9055	**0.9603**	0.9542	0.9072	**0.9588**	0.9372
K=5	**0.9066**	0.9583	0.9544	**0.9101**	0.9584	**0.9376**

which is denoted by $I_{\max}$. From the bottom figures of Figure 1, we can see that the $I_{\max}$ value of CODASCA on ImageNet is 128 (K=16) and 512 (K=8), respectively, and that of CODA+ on ImageNet is 32 (K=16) and 128 (K=8). This demonstrates that CODASCA enjoys consistent advantage over CODA+, i.e., when $K = 16$, $I_{\max}^{\text{CODASCA}}/I_{\max}^{\text{CODA+}} = 4$, and when $K = 8$, $I_{\max}^{\text{CODASCA}}/I_{\max}^{\text{CODA+}} = 4$. The same phenomena occur on CIFAR100 data.

Next, we compare CODASCA with CODA+ on the ChestXray-IH medical dataset, which is also highly heterogeneous. We split the ChestXray-IH data into $K = 16$ groups according to the patient ID and each machine only owns samples from one organization without overlapping patients. The testing set is the collection of 5% data sampled from each organization. In addition, we use train/val split = 7:3 for the parameter tuning. We run CODASCA and CODA+ with the same number of iterations. The performance on testing set are reported in Table 3. From the results, we can observe that CODASCA performs consistently better than CODA+ on C0, C2, C3, C4.

6.2. FDAM for improving performance on CheXpert

Finally, we show that FDAM can be used to leverage data from multiple hospitals to improve the performance at a single target hospital. For this experiment, we choose CheXpert data from Stanford Hospital as the target data. Its validation data will be used for evaluating the performance of our FDAM method. Note that improving the AUC score on CheXpert is a very challenging task. The top 7 teams on CheXpert leaderboard differ by only 0.1% [1]. Hence, we consider any improvement over 0.1% significant. Our procedure is following: we gradually increase the number of data resources, e.g., $K = 1$ only includes the CheXpert training data, $K = 2$ includes the CheXpert training data and ChestXray8, $K = 3$ includes the CheXpert training data and ChestXray8 and PadChest, and so on.

[1] https://stanfordmlgroup.github.io/competitions/chexpert/

Table 5. Performance of FDAM on Chexpert validation set for DenSenet161.

of sources	C0	C1	C2	C3	C4	AVG
K=1	0.8946	0.9527	0.9544	0.9008	0.9556	0.9316
K=2	0.8938	0.9615	0.9568	0.9109	0.9517	0.9333
K=3	**0.9008**	0.9603	0.9568	0.9127	0.9505	0.9356
K=4	0.8986	**0.9615**	0.9561	0.9128	**0.9564**	0.9367
K=5	0.8986	0.9612	**0.9568**	**0.9130**	0.9552	**0.9370**

Parameters and Settings. Due to the limited computing resources, we resize all images to 320x320. We follow the two stage method proposed in (Yuan et al., 2020b) and compare with the baseline on a single machine with a single data source (CheXpert training data) (K=1) for learning DenseNet121, DenseNet161. More specifically, we first train a base model by minimizing the Cross-Entropy loss on CheXpert training dataset using Adam with a initial learning rate of 1e-5 and batch size of 32 for 2 epochs. Then, we discard the trained classifier, use the same pretrained model for initializing the local models at all machines and continue training using CODASCA. For the parameter tuning, we try I=[16, 32, 64, 128], learning rate=[0.1, 0.01] and we fix γ=1e-3, T_0=1000 and batch size=32.

Results. We report all results in term of AUC score on the CheXpert validation data in Table 4 and Table 5. We can see that using more data sources from different organizations can efficiently improve the performance on CheXpert. For DenseNet121, the average improvement across all 5 classification tasks from $K = 1$ to $K = 5$ is over 0.2% which is significant in light of the top CheXpert leaderboard results. Specifically, we can see that CODASCA with K=5 achieves the highest validation AUC score on C0 and C3, and with K=4 achieves the highest on C1 and C4. For DenseNet161, the improvement of average AUC is over 0.5%, which doubles the 0.2% improvement for DenseNet121.

7. Conclusion

In this work, we have conducted comprehensive studies of federated learning for deep AUC maximization. We analyzed a stronger baseline for deep AUC maximization by establishing its convergence for both homogeneous data and heterogeneous data. We also developed an improved variant by adding control variates to the local stochastic gradients for both primal and dual variables, which dramatically reduces the communication complexity. Besides a strong theory guarantee, we exhibit the power of FDAM on real world medical imaging problems. We have shown that our FDAM method can improve the performance on medical imaging classification tasks by leveraging data from different organizations that are kept locally.

Acknowledgements

We are grateful to the anonymous reviewers for their constructive comments and suggestions. This work is partially supported by NSF #1933212 and NSF CAREER Award #1844403.

References

Ardila, D., Kiraly, A. P., Bharadwaj, S., Choi, B., Reicher, J. J., Peng, L., Tse, D., Etemadi, M., Ye, W., Corrado, G., et al. End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography. *Nature medicine*, 25(6):954–961, 2019.

Arjevani, Y. and Shamir, O. Communication complexity of distributed convex learning and optimization. In *Advances in Neural Information Processing Systems 28 (NeurIPS)*, pp. 1756–1764, 2015.

Bejnordi, B. E., Veta, M., Van Diest, P. J., Van Ginneken, B., Karssemeijer, N., Litjens, G., Van Der Laak, J. A., Hermsen, M., Manson, Q. F., Balkenhol, M., et al. Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *Jama*, 318(22):2199–2210, 2017.

Bustos, A., Pertusa, A., Salinas, J.-M., and de la Iglesia-Vayá, M. Padchest: A large chest x-ray image dataset with multi-label annotated reports. *Medical Image Analysis*, 66:101797, 2020.

Chen, K. and Huo, Q. Scalable training of deep learning machines by incremental block training with intra-block parallel optimization and blockwise model-update filtering. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSSP)*, pp. 5880–5884, 2016.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In *Proceedings of the IEEE annual Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., and Thrun, S. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115–118, 2017.

Guo, Z., Liu, M., Yuan, Z., Shen, L., Liu, W., and Yang, T. Communication-efficient distributed stochastic AUC maximization with deep neural networks. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pp. 3864–3874, 2020a.

Guo, Z., Yuan, Z., Yan, Y., and Yang, T. Fast objective and duality gap convergence for non-convex strongly-concave min-max problems. *arXiv preprint arXiv:2006.06889*, 2020b.

Haddadpour, F., Kamani, M. M., Mahdavi, M., and Cadambe, V. R. Local SGD with periodic averaging: Tighter analysis and adaptive synchronization. In *Advances in Neural Information Processing Systems 32 (NeurIPS)*, pp. 11080–11092, 2019.

Irvin, J., Rajpurkar, P., Ko, M., Yu, Y., Ciurea-Ilcus, S., Chute, C., Marklund, H., Haghgoo, B., Ball, R. L., Shpanskaya, K. S., Seekins, J., Mong, D. A., Halabi, S. S., Sandberg, J. K., Jones, R., Larson, D. B., Langlotz, C. P., Patel, B. N., Lungren, M. P., and Ng, A. Y. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI)*, pp. 590–597, 2019.

Johnson, A. E. W., Pollard, T. J., Berkowitz, S., Greenbaum, N. R., Lungren, M. P., Deng, C.-y., Mark, R. G., and Horng, S. Mimic-cxr: A large publicly available database of labeled chest radiographs. *arXiv preprint arXiv:1901.07042*, 2019.

Kamp, M., Adilova, L., Sicking, J., Hüger, F., Schlicht, P., Wirtz, T., and Wrobel, S. Efficient decentralized deep learning by dynamic model averaging. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD)*, pp. 393–409. Springer, 2018.

Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S. J., Stich, S. U., and Suresh, A. T. Scaffold: Stochastic controlled averaging for on-device federated learning. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.

Khaled, A., Mishchenko, K., and Richtárik, P. Tighter theory for local SGD on identical and heterogeneous data. In *The 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 4519–4529, 2020.

Krizhevsky, A., Nair, V., and Hinton, G. CIFAR-10 and CIFAR-100 datasets. *URl: https://www. cs. toronto. edu/kriz/cifar. html*, 6:1, 2009.

Li, Y. and Shen, L. Skin lesion analysis towards melanoma detection using deep learning network. *Sensors*, 18(2): 556, 2018.

Lin, T., Jin, C., and Jordan, M. I. On gradient descent ascent for nonconvex-concave minimax problems. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119, pp. 6083–6093, 2020a.

Lin, T., Stich, S. U., Patel, K. K., and Jaggi, M. Don't use large mini-batches, use local SGD. In *8th International Conference on Learning Representations (ICLR)*, 2020b.

Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A. W. M., van Ginneken, B., and Sánchez, C. I. A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42:60–88, 2017.

Liu, M., Zhang, X., Chen, Z., Wang, X., and Yang, T. Fast stochastic auc maximization with O (1/n)-convergence rate. In *Proceedings of 35th International Conference on Machine Learning (ICML)*, pp. 3195–3203, 2018.

Liu, M., Yuan, Z., Ying, Y., and Yang, T. Stochastic AUC maximization with deep neural networks. In *8th International Conference on Learning Representations (ICLR)*, 2020.

Long, G., Tan, Y., Jiang, J., and Zhang, C. Federated learning for open banking. In *Federated Learning*, pp. 240–254. Springer, 2020.

Luo, L., Ye, H., Huang, Z., and Zhang, T. Stochastic recursive gradient descent ascent for stochastic nonconvex-strongly-concave minimax problems. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, 2020.

McKinney, S. M., Sieniek, M., Godbole, V., Godwin, J., Antropova, N., Ashrafian, H., Back, T., Chesus, M., Corrado, G. S., Darzi, A., et al. International evaluation of an AI system for breast cancer screening. *Nature*, 577 (7788):89–94, 2020.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 1273–1282, 2017.

McMahan, H. B., Moore, E., Ramage, D., Hampson, S., et al. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2016.

McMahan, H. B. et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1), 2019.

Natole, M., Ying, Y., and Lyu, S. Stochastic proximal algorithms for auc maximization. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pp. 3707–3716, 2018.

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

Nesterov, Y. E. *Introductory Lectures on Convex Optimization - A Basic Course*, volume 87 of *Applied Optimization*. Springer, 2004.

Nguyen, H. Q., Lam, K., Le, L. T., Pham, H. H., Tran, D. Q., Nguyen, D. B., Le, D. D., Pham, C. M., Tong, H. T., Dinh, D. H., et al. Vindr-cxr: An open dataset of chest x-rays with radiologist's annotations. *arXiv preprint arXiv:2012.15029*, 2020.

Pooch, E. H., Ballester, P., and Barros, R. C. Can we trust deep learning based diagnosis? the impact of domain shift in chest radiograph classification. In *International Workshop on Thoracic Image Analysis*, pp. 74–83. Springer, 2020.

Povey, D., Zhang, X., and Khudanpur, S. Parallel training of dnns with natural gradient and parameter averaging. *arXiv preprint arXiv:1410.7455*, 2014.

Rafique, H., Liu, M., Lin, Q., and Yang, T. Non-convex min-max optimization: Provable algorithms and applications in machine learning. *arXiv preprint arXiv:1810.02060*, 2018.

Rieke, N., Hancox, J., Li, W., Milletari, F., Roth, H. R., Albarqouni, S., Bakas, S., Galtier, M. N., Landman, B. A., Maier-Hein, K., et al. The future of digital health with federated learning. *NPJ digital medicine*, 3(1):1–7, 2020.

Stich, S. U. Local SGD converges fast and communicates little. In *7th International Conference on Learning Representations (ICLR)*, 2019.

Su, H. and Chen, H. Experiments on parallel training of deep neural network using model averaging. *arXiv preprint arXiv:1507.01239*, 2015.

Wang, D., Khosla, A., Gargeya, R., Irshad, H., and Beck, A. H. Deep learning for identifying metastatic breast cancer. *arXiv preprint arXiv:1606.05718*, 2016.

Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., and Summers, R. M. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2097–2106, 2017.

Woodworth, B. E., Patel, K. K., and Srebro, N. Minibatch vs local SGD for heterogeneous distributed learning. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, 2020a.

Woodworth, B. E., Patel, K. K., Stich, S. U., Dai, Z., Bullins, B., McMahan, H. B., Shamir, O., and Srebro, N. Is local SGD better than minibatch sgd? In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pp. 10334–10343, 2020b.

Yan, Y., Xu, Y., Lin, Q., Liu, W., and Yang, T. Optimal epoch stochastic gradient descent ascent methods for min-max optimization. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, 2020.

Yang, J., Kiyavash, N., and He, N. Global convergence and variance reduction for a class of nonconvex-nonconcave minimax problems. *Advances in Neural Information Processing Systems 33 (NeurIPS)*, 2020.

Ying, Y., Wen, L., and Lyu, S. Stochastic online auc maximization. In *Advances in Neural Information Processing Systems 29 (NeurIPS)*, pp. 451–459, 2016.

Yu, H., Jin, R., and Yang, S. On the linear speedup analysis of communication efficient momentum SGD for distributed non-convex optimization. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pp. 7184–7193, 2019a.

Yu, H., Yang, S., and Zhu, S. Parallel restarted sgd with faster convergence and less communication: Demystifying why model averaging works for deep learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 5693–5700, 2019b.

Yuan, Z., Guo, Z., Yu, X., Wang, X., and Yang, T. Accelerating deep learning with millions of classes. In *16th European Conference on Computer Vision (ECCV)*, pp. 711–726, 2020a.

Yuan, Z., Yan, Y., Sonka, M., and Yang, T. Robust deep auc maximization: A new surrogate loss and empirical studies on medical image classification. *arXiv preprint arXiv:2012.03173*, 2020b.