2024 IEEE 63rd Conference on Decision and Control (CDC)
December 16-19, 2024. MiCo, Milan, Italy

Distributed Learning by Local Training ADMM

Xiaoxing Ren[1], Nicola Bastianello[2⋆], Karl H. Johansson[2], Thomas Parisini[1,3,4]

***Abstract*— In this paper, we focus on distributed learning over peer-to-peer networks. In particular, we address the challenge of expensive communications (which arise when *e.g.* training neural networks), by proposing a novel local training algorithm, LT-ADMM. We extend the distributed ADMM enabling the agents to perform multiple local gradient steps per communication round (local training). We present a preliminary convergence analysis of the algorithm under a graph regularity assumption, and show how the use of local training does not compromise the accuracy of the learned model. We compare the algorithm with the state of the art for a classification task, and in different set-ups. The results are very promising showing a great performance of LT-ADMM, and paving the way for future important theoretical developments.**

I. Introduction

The technological progress of the past decades has led to the spread of devices equipped with computing and communication resources in a range of applications, from power grids to robotics, from traffic to sensor networks, to name a few [1], [2]. These devices interconnect with each other thus creating multi-agent systems capable of collecting and processing data in a cooperative fashion [3]. Therefore, there is a pressing need for novel algorithms enabling efficient learning paradigms over multi-agent systems.

In decentralized learning we can distinguish two approaches, characterized by the topology of the agents interconnections: federated learning [3], where a central node coordinates all the other ones, and fully distributed learning, where each agent communicates with a subset of the others. In this paper, we focus on the latter set-up because of its inherent resilience since a single point of failure represented by the central node is not present in this case.

Let us first characterise our approach with respect to the literature on distributed optimization and learning [2], [4]. There are two main classes of distributed algorithms: gradient-based, which includes DGD and Gradient Tracking, and dual algorithms, which includes dual ascent and ADMM [4]. The algorithms of both classes are characterized by the alternation of a communication step with a training step, which employs the local data stored by an agent. However, in many learning applications we encounter a fundamental challenge: *communications are expensive*. Indeed, consider the task of training a (deep) neural network – the agents need to share the parameters of their local NN, which results in rather sizable packets. The objective therefore is to design decentralized learning algorithms in which *agents perform multiple training steps (or epochs) for each communication step*.

We call this paradigm *local training*, and in the following we briefly review the literature on the topic. Local training was proposed as a heuristic in the seminal federated learning paper [5] – however, the heterogeneity of the agents' data may worsen the accuracy [6]. Different federated algorithms have then been designed to employ local training without compromising accuracy [7], [8], [9], [6], [10]. Similarly, fully distributed algorithms with local training have been proposed in [11], [12], [13], [14]. In particular, [12], [14] are based on gradient tracking algorithms, [11] can be interpreted as a dual ascent method, and [13] as a gradient tracking method with communication rounds performed at random.

In this paper, we focus on a different algorithm from the class of dual methods: the distributed ADMM [15]. In particular, we provide the following contributions:

- We propose a novel algorithm based on the distributed ADMM which allows for local training, with communication rounds interspersed between multiple local gradient steps. We denote the proposed algorithm as LT-ADMM for Local Training ADMM.
- We analyze the convergence of LT-ADMM for strongly convex problems, and under the assumption that the communication graph is regular. In particular, the convergence is exact (hence accuracy is not impacted), and we characterize it in terms of the number of local training steps performed by the agents. We remark that the technical assumption of regularity will be relaxed in future works.
- We compare LT-ADMM to state-of-the-art alternatives for a classification task, and highlight its better performance. The comparison is then extended to set-ups with quantized communications and stochastic gradients. We further discuss the performance of LT-ADMM for different parameters choices and for different topologies (including non-regular ones).

The work of X. R and T. P. was partially supported by European Union's Horizon 2020 research and innovation programme under grant agreement no. 739551 (KIOS CoE).

The work of N.B. and K.H.J. was partially supported by the European Union's Horizon Research and Innovation Actions programme under grant agreement No. 101070162, and partially by Swedish Research Council Distinguished Professor Grant 2017-01078 Knut and Alice Wallenberg Foundation Wallenberg Scholar Grant.

[1]Department of Electrical and Electronic Engineering, Imperial College London, London, United Kingdom.

[2]School of Electrical Engineering and Computer Science, and Digital Futures, KTH Royal Institute of Technology, Stockholm, Sweden.

[3]Department of Electronic Systems, Aalborg University, Denmark.

[4]Department of Engineering and Architecture, University of Trieste, Trieste, Italy.

⋆Corresponding author. Email: nicolba@kth.se.

II. Problem Formulation and Algorithm Design

A. Problem Formulation

Consider a network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consisting of N agents tasked with solving the following consensus optimization problem

$$\min_{x_i \in \mathbb{R}^n,\ i \in \mathcal{V}} \sum_{i=1}^{N} f_i(x_i) = F(\mathbf{X}) \quad \text{s.t. } x_1 = \ldots = x_N \tag{1}$$

where $f_i : \mathbb{R}^n \to \mathbb{R}$ is the local cost function of agent i, and $\mathbf{X} = \operatorname{col}\{x_1, x_2, \cdots, x_N\} \in \mathbb{R}^{Nn}$. In the following, we denote the optimal solution to (1) by $\mathbf{X}^* = \mathbf{1}_N \otimes x^*$, where $\otimes$ denotes Kronecker product, and $x^* = \arg\min_{x \in \mathbb{R}^n} \sum_{i=1}^{N} f_i(x)$.

The following assumptions on the cost functions and network will hold throughout the paper.

Assumption 1: The cost function of each agent $i \in \mathcal{V}$ is L-smooth and μ-strongly convex, with $L, \mu > 0$. That is, $\forall x, y \in \mathbb{R}^n$, $\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x - y\|$, and $\langle \nabla f_i(x) - \nabla f_i(y), x - y \rangle \geq \mu \|x - y\|^2$.

Assumption 2: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a connected, undirected, and regular graph, with all agents having the same degree d.

Assumptions 1 and 2 are generally standard, with the exception of the requirement for a regular graph. This current limitation is technical in nature, as the numerical results of section IV show convergence for non-regular graphs as well. Relaxing the regularity assumption will be the objective of future research.

The next section will present the proposed algorithm for distributed learning. As motivated in section I, the guiding objective will be to *design an algorithm that allows for multiple local gradient steps per communication round.*

B. Algorithm design

The starting point of our design is the distributed ADMM of [15], which is characterized by the local updates

$$x_{i,k+1} = \operatorname{prox}_{f_i/\rho|\mathcal{N}_i|}\left(\frac{1}{\rho\,|\mathcal{N}_i|}\sum_{j \in \mathcal{N}_i} z_{ij,k}\right) \tag{2a}$$

$$z_{ij,k+1} = \frac{1}{2} z_{ij,k} - \frac{1}{2}\left(z_{ji,k} - 2\rho x_{j,k+1}\right) \tag{2b}$$

where $\rho > 0$ is a penalty parameter, and $\operatorname{prox}_{f_i/\rho|\mathcal{N}_i|}(z) = \arg\min_{x \in \mathbb{R}^n}\left\{ f_i(x) + \frac{\rho|\mathcal{N}_i|}{2}\|x - z\|^2 \right\}$. During the first step (2a), the agents perform local training using their cost functions, while during the second step (2b) they update the auxiliary variables after communicating with their neighbors.

However, update (2a) entails the solution of an optimization problem – which in general does not have a closed form, especially for learning problems. The agents then need to approximate the solution of (2a). We proposed to do so with $\tau \in \mathbb{N}$ steps of local gradient descent:

$$\begin{aligned} &\phi_k^{0,i} = x_{i,k} \\ &\phi_k^{t+1,i} = \phi_k^{t,i} + && t = 0, \ldots, \tau - 1 \\ &\quad - \gamma\left(\nabla f_i(\phi_k^{t,i}) + \rho\,|\mathcal{N}_i|\left(\phi_k^{t,i} - \frac{1}{\rho|\mathcal{N}_i|}\sum_{j \in \mathcal{N}_i} z_{ij,k}\right)\right) \\ &x_{i,k+1} = \phi_k^{\tau,i} \end{aligned} \tag{3}$$

where $\gamma < 1/(L + \rho|\mathcal{N}_i|)$ is the local step-size.

Applying local training (3) to the distributed ADMM (2) results in the proposed LT-ADMM Algorithm 1.

Algorithm 1 LT-ADMM.

Input: For each node i, initialize $x_{i,0}$ and $z_{ij,0}$, $j \in \mathcal{N}_i$. Set the penalty ρ, the number of local training steps τ, and the local step-size γ.

1: **for** $k = 0, 1, \ldots$ every agent i **do**
  // local training
2: update $x_{i,k+1}$ according to (3)
  // communication
3: transmit $z_{ij,k} - 2\rho x_{i,k+1}$ to each neighbor $j \in \mathcal{N}_i$, and receive the corresponding transmissions
  // auxiliary update
4: update $z_{ij,k+1}$ according to (2b)
5: **end for**

Remark 1: In this paper we approximate (2a) with τ steps of local gradient descent. But in principle we could use different solvers for local training, *e.g.* stochastic gradient descent or accelerated gradient descent. We also remark that the initialization $\phi_k^{0,i} = x_{i,k}$ in (3) is fundamental to guarantee convergence. Intuitively, this choice instates a feedback loop that controls to zero the approximation error.

III. Convergence Analysis

In this section, we analyze the convergence of the proposed LT-ADMM.

Theorem 1: Let Assumptions 1 and 2 hold. Let $\{\mathbf{X}_k\}_{k \in \mathbb{N}}$ be the trajectory generated by LT-ADMM. Then there exists $\bar{\gamma} > 0$ (cf. (16)) such that if $\gamma < \bar{\gamma}$ and $\tau > \frac{2}{N\mu}$, the states $\mathbf{X}_k$ converge linearly to the optimal solution $\mathbf{X}^*$.

Remark 2: We remark that to achieve convergence of LT-ADMM, we need to impose bounds on the local step-size γ and the number of local iterations τ. This is in line with previous distributed algorithms that employ local training [11], [12], [13], [14].

In the next section, we sketch the proof of Theorem 1.

A. Rewrite LT-ADMM in a compact form

We start by rewriting LT-ADMM in a compact form. To this end, define the matrix $\mathbf{A} = \operatorname{blk}\operatorname{diag}\{\mathbf{1}_d\} \otimes \mathbf{I}_n \in \mathbb{R}^{Mn \times Nn}$ where $M = \sum_i |\mathcal{N}_i|$, and the vectors $\nabla F(\mathbf{X}) = \operatorname{col}\{\nabla f_1(x_1), f_2(x_2), ..., f_N(x_N)\}$, $\mathbf{\Phi}_k^t = \operatorname{col}\{\phi_k^{t,1}, \phi_k^{t,2}, ..., \phi_k^{t,N}\}$, $\mathbf{Z} = \operatorname{col}\{z_{ij}\}_{i,j \in \mathcal{E}}$. Moreover, define $\mathbf{P} \in \mathbb{R}^{Mn \times Mn}$ as the permutation matrix that swaps z_{ij}

with z_{ji}. We can then rewrite LT-ADMM as

$$\mathbf{X}_{k+1} = \mathbf{X}_k - \gamma \sum_{t=0}^{\tau-1} (\nabla F(\mathbf{\Phi}_k^t) + \rho \mathbf{A}^T \mathbf{A} \mathbf{\Phi}_k^t - \mathbf{A}^T \mathbf{Z}_k) \quad (4a)$$

$$\mathbf{Z}_{k+1} = \frac{1}{2}\mathbf{Z}_k - \frac{1}{2}\mathbf{P}\mathbf{Z}_k + \rho \mathbf{P}\mathbf{A}\mathbf{X}_{k+1}. \quad (4b)$$

Introducing the following variables

$$\mathbf{Y}_k = \mathbf{A}^T \mathbf{Z}_k - \nabla F(\bar{\mathbf{X}}_k) - \rho \mathbf{D}\mathbf{X}_k$$
$$\tilde{\mathbf{Y}}_k = \mathbf{A}^T \mathbf{P}\mathbf{Z}_k + \nabla F(\bar{\mathbf{X}}_k) - \rho \mathbf{D}\mathbf{X}_k,$$

where $\bar{\mathbf{X}}_k = \mathbf{1}_N \otimes \bar{x}_k, \bar{x}_k = \frac{1}{N}\mathbf{1}^T \mathbf{X}_k$, and $\mathbf{D} = \text{diag}\{d\mathbf{I}_n\}$, (4) can be further rewritten as

$$\begin{bmatrix} \mathbf{X}_{k+1} \\ \mathbf{Y}_{k+1} \\ \tilde{\mathbf{Y}}_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \gamma\tau\mathbf{I} & \mathbf{0} \\ \rho\tilde{\mathbf{L}} & \rho\tilde{\mathbf{L}}\gamma\tau + \frac{1}{2} & -\frac{1}{2}\mathbf{I} \\ \mathbf{0} & -\frac{1}{2}\mathbf{I} & \frac{1}{2}\mathbf{I} \end{bmatrix} \otimes \mathbf{I}_n \begin{bmatrix} \mathbf{X}_k \\ \mathbf{Y}_k \\ \tilde{\mathbf{Y}}_k \end{bmatrix} - \mathbf{h}_k, \quad (5)$$

where $\tilde{\mathbf{L}} = \mathbf{A} - \mathbf{D}$ and

$$\mathbf{h}_k = [\gamma \sum_{t=0}^{\tau-1} (\nabla F(\mathbf{\Phi}_k^t) - \nabla F(\bar{\mathbf{X}}_k) + \rho D \mathbf{\Phi}_k^t - \rho \mathbf{D}\mathbf{X}_k);$$
$$\gamma\rho\tilde{\mathbf{L}} \sum_{t=0}^{\tau-1} (\nabla F(\mathbf{\Phi}_k^t) - \nabla F(\bar{\mathbf{X}}_k) + \rho \mathbf{D}\mathbf{\Phi}_k^t - \rho \mathbf{D}\mathbf{X}_k)$$
$$+ \nabla F(\bar{\mathbf{X}}_{k+1}) - \nabla F(\bar{\mathbf{X}}_k); -\nabla F(\bar{\mathbf{X}}_{k+1}) + \nabla F(\bar{\mathbf{X}}_k)].$$

We remark that (5) can be interpreted as a linear dynamic system, with the non-linearity of the gradients as input in $\mathbf{h}_k$.

With this reformulation in place, the goal now is to bound the distance of $\mathbf{X}$ from the optimal solutions. Specifically, we do this by first bounding the distance from the average, and then bounding the distance of the average from the optimal solution.

B. Deviation from the average

Under Assumption 2, we define the weight matrix of $\mathcal{G}$ as $\mathbf{W} = \frac{1}{2}(\frac{\mathbf{A}+\mathbf{I}}{d+1} + \mathbf{I})$, which can be decomposed as $\mathbf{W} = \begin{bmatrix} \frac{1}{\sqrt{N}}\mathbf{1} & \widehat{\mathbf{Q}} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \widehat{\Lambda} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{N}}\mathbf{1}^T \\ \widehat{\mathbf{Q}}^T \end{bmatrix}$ [16], where $\widehat{\Lambda} = \text{diag}\{\lambda_i\}_{i=2}^N$, $0 < \lambda_2 \leq \ldots \leq \lambda_N < 1$, and $\widehat{\mathbf{Q}} \in \mathbf{R}^{N\times(N-1)}$ satisfies $\widehat{\mathbf{Q}}\widehat{\mathbf{Q}}^T = I_N - \frac{1}{N}\mathbf{1}\mathbf{1}^T$, $\widehat{\mathbf{Q}}^T\widehat{\mathbf{Q}} = I_{N-1}$, $\mathbf{1}^T\widehat{\mathbf{Q}} = 0$ and $\widehat{\mathbf{Q}}^T\mathbf{1} = 0$. Multiplying both sides of (5) by $\widehat{\mathbf{Q}}^T$ we obtain

$$\begin{bmatrix} \widehat{\mathbf{Q}}^T X_{k+1} \\ \widehat{\mathbf{Q}}^T Y_{k+1} \\ \widehat{\mathbf{Q}}^T \tilde{Y}_{k+1} \end{bmatrix} = (\mathbf{\Theta} \otimes \mathbf{I}_n) \begin{bmatrix} \widehat{\mathbf{Q}}^T X_k \\ \widehat{\mathbf{Q}}^T Y_k \\ \widehat{\mathbf{Q}}^T \tilde{Y}_k \end{bmatrix} - \widehat{\mathbf{Q}}^T \mathbf{h}_k \quad (6)$$

where $\mathbf{\Theta} = \begin{bmatrix} \mathbf{I} & \gamma\tau\mathbf{I} & \mathbf{0} \\ \rho\,\text{diag}\left\{\tilde{\lambda}_i\right\} & \rho\,\text{diag}\left\{\tilde{\lambda}_i\right\}\gamma\tau\mathbf{I} + \frac{1}{2}\mathbf{I} & -\frac{1}{2}\mathbf{I} \\ \mathbf{0} & \frac{1}{2}\mathbf{I} & \frac{1}{2}\mathbf{I} \end{bmatrix}$, $\tilde{\lambda}_i = 2(d+1)(\lambda_i - 1)$.

Since each block of $\mathbf{\Theta}$ is diagonal matrices, there exists a permutation matrix $\mathbf{P}_0$ such that $\mathbf{P}_0\mathbf{\Theta}\mathbf{P}_0^T = \text{blkdiag}\{\mathbf{D}_i\}_{i=2}^N$, where $\mathbf{D}_i = \begin{bmatrix} 1 & \gamma\tau & 0 \\ \rho\tilde{\lambda}_i & \rho\tilde{\lambda}_i\gamma\tau + \frac{1}{2} & -\frac{1}{2} \\ 0 & -\frac{1}{2} & \frac{1}{2} \end{bmatrix}$, diagonalize $\mathbf{D}_i = \mathbf{V}_i\mathbf{\Delta}_i\mathbf{V}_i^{-1}$. We conclude that $\mathbf{\Theta} = \mathbf{P}_0^T\mathbf{V}\mathbf{\Delta}\mathbf{V}^{-1}\mathbf{P}_0$ where $\mathbf{V} = \text{blkdiag}\{\mathbf{V}_i\}_{i=2}^N$ and $\mathbf{\Delta} = \text{blkdiag}\{\mathbf{\Delta}_i\}_{i=2}^N$. Multiplying both sides of (6) by $\widehat{\mathbf{V}}^{-1}$ where $\widehat{\mathbf{V}} = P_0^T\mathbf{V}$ gives

$$\widehat{\mathbf{d}}_{k+1} = \mathbf{\Delta}\widehat{\mathbf{d}}_k - \widehat{\mathbf{h}}_k, \quad (7)$$

where $\widehat{\mathbf{d}}_k = \widehat{\mathbf{V}}^{-1}\begin{bmatrix} \widehat{\mathbf{Q}}^T\mathbf{X}_k \\ \widehat{\mathbf{Q}}^T\mathbf{Y}_k \\ \widehat{\mathbf{Q}}^T\tilde{\mathbf{Y}}_k. \end{bmatrix}$, $\widehat{\mathbf{h}}_k = \widehat{\mathbf{V}}^{-1}\widehat{\mathbf{Q}}^T\mathbf{h}_k$. Under the condition that $-\frac{8}{3} < \tilde{\lambda}_i\rho\tau\gamma < 0$, *i.e.*, $2(d+1)(1-\lambda_2)\rho\tau\gamma < \frac{8}{3}$, $\mathbf{\Delta}$ is a matrix with norm $\delta = \|\mathbf{\Delta}\| < 1$ [17]. Additionally, when $\gamma < \min\{\frac{1}{\tau}, \frac{2(d+1)(1-\lambda_N)\rho}{2\tau}\}$, the magnitude of all elements of V_i is less than or equal to 1, it follows that

$$\|\bar{\mathbf{X}}_k - \mathbf{X}_k\|^2 \leq 9\|\widehat{\mathbf{d}}_k\|^2, \quad \|\bar{\mathbf{Y}}_k - \mathbf{Y}_k\|^2 \leq 9\|\widehat{\mathbf{d}}_k\|^2. \quad (8)$$

Define now $\|\widehat{\mathbf{\Phi}}^k\|^2 = \sum_{i=1}^N \sum_{t=0}^{\tau-1} \left\|\phi_k^{t,i} - \bar{x}^k\right\|^2 = \sum_{t=0}^{\tau-1} \left\|\mathbf{\Phi}_k^t - \bar{\mathbf{X}}_k\right\|^2$. Suppose that $\tau > 2$, using similar analysis as [11] we obtain that when $4\gamma^2\tau(L^2 + \rho^2 d^2) < \frac{1/4}{\tau-1}$, both the following inequalities hold

$$\|\widehat{\mathbf{\Phi}}_k\|^2 \leq (36\tau + 16\tau^3\gamma^2(18 + 18\rho^2 d^2))\|\widehat{\mathbf{d}}_k\|^2 + 32\tau^3\gamma^2 LN(F(\bar{x}_k) - F(x^*)), \quad (9)$$

$$\|\widehat{\mathbf{\Phi}}_k\|^2 \leq (36\tau + 16\tau^3\gamma^2(18 + 18\rho^2 d^2))\|\widehat{\mathbf{d}}_k\|^2 + 16\tau^3\gamma^2 L^2 N\|\bar{x}_k - x^*\|^2. \quad (10)$$

From the definition of $\widehat{\mathbf{h}}_k$ it holds that

$$\left\|\widehat{\mathbf{h}}_k\right\|^2 \leq (6L^2\|\widehat{\mathbf{V}}^{-1}\|^2\gamma^2\tau(2L^2 + \rho^2 d^2) + 3\gamma^2\tau(1 + 2\|\tilde{L}\|^2)\|\widehat{\mathbf{V}}^{-1}\|^2(L^2 + \rho^2 d^2))\|\widehat{\mathbf{\Phi}}^k\|^2 + 27\gamma^2(1 + 2\|\tilde{L}\|^2)\|\widehat{\mathbf{V}}^{-1}\|^2\tau^2\rho^2 d^2\|\widehat{\mathbf{d}}^k\|^2 + 12L^4\|\widehat{\mathbf{V}}^{-1}\|^2\gamma^2\tau^2 N\|\bar{x}_k - x^*\|^2, \quad (11)$$

while from (7) and using Jensen's inequality

$$\|\widehat{\mathbf{d}}_{k+1}\|^2 \leq \delta\|\widehat{\mathbf{d}}_k\|^2 + \frac{1}{1-\delta}\left\|\widehat{\mathbf{h}}_k\right\|^2 \quad (12)$$

where recall that $\delta = \|\Delta\| < 1$.

C. Distance from the average to x^*

Multiplying both sides of (4) by $\mathbf{1}^T\mathbf{A}^T$, under Assumption 2, $\mathbf{1}^T\mathbf{A}^T\mathbf{Z}_k = \rho\mathbf{1}^T\mathbf{D}\mathbf{X}_k = \rho\mathbf{1}^T\mathbf{D}\bar{\mathbf{X}}_k$ for all k. Denote

$$\epsilon_k = \sum_{t=0}^{\tau-1}(\rho\mathbf{1}^T\mathbf{D}\mathbf{\Phi}_k^t - \mathbf{1}^T\mathbf{A}^T\mathbf{Z}_k) = \rho d\sum_{t=0}^{\tau-1}\sum_{i=1}^{N}(\phi_k^{t,i} - \bar{x}_k),$$

we have $\epsilon_k^2 \leq \rho^2 d^2\tau N\|\widehat{\mathbf{\Phi}}^k\|^2$. From (4a) $\bar{x}_{k+1} - x^* = \bar{x}_k - x^* - \frac{\gamma}{N}\sum_{t=0}^{\tau-1}\sum_{i=1}^{N}\nabla f_i(\phi_k^{t,i}) - \frac{\gamma}{N}\epsilon_k$, which leads to

$$\|\bar{x}_{k+1} - x^*\|^2 = \frac{\gamma^2}{N^2}\left\|\sum_{t=0}^{\tau-1}\sum_{i=1}^{N}\nabla f_i(\phi_k^{t,i}) + \epsilon_k\right\|^2 + \|\bar{x}_k - x^*\|^2 - 2\frac{\gamma}{N}\langle\bar{x}_k - x^*, \sum_{t=0}^{\tau-1}\sum_{i=1}^{N}\nabla f_i(\phi_k^{t,i}) + \epsilon_k\rangle.$$

Now, any L-smooth and μ-strongly convex function g verifies $\langle (z-y), \nabla g(x)\rangle \geq g(z) - g(y) + \frac{\mu}{4}\|y-z\|^2 - L\|z-x\|^2$, $\forall x,y,z \in R^n$ [18]; moreover, $\|\frac{1}{N}\sum_i \nabla f_i(\bar{x}^k)\|^2 \leq 2L(F(\bar{x}^k) - F(x^*))$ [19]. Thus, using these inequalities we derive that when $32\tau^3\gamma^2 L\left(2\gamma L + 4\gamma^2\tau L^2 + \rho^2 d^2 \gamma\tau N + 2\gamma^2\rho^2 d^2\tau\right) + 8\gamma^2\tau^2 L - 2\gamma\tau < 0$,

$$\left\|\bar{x}^{k+1} - x^\star\right\|^2 \leq \left(1 - \frac{\mu\tau\gamma}{2} + \frac{4\gamma}{N}\right)\left\|\bar{x}^k - x^\star\right\|^2 + \\ + \left(\frac{2\gamma L}{N} + \frac{4\gamma^2\tau L^2}{N} + 2\rho^2 d^2\gamma\tau + \frac{4\gamma^2\rho^2 d^2\tau}{N}\right) \cdot \\ \cdot \left(36\tau + 16\tau^3\gamma^2(18 + 18\rho^2 d^2)\right)\|\widehat{\mathbf{d}}_k\|^2. \tag{13}$$

D. Main convergence result

Drawing (11), (12) and (13) together, we derive that

$$\begin{bmatrix} \left\|\bar{x}^{k+1} - x^\star\right\|^2 \\ \|\widehat{\mathbf{d}}_{k+1}\|^2 \end{bmatrix} = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \begin{bmatrix} \left\|\bar{x}^{k} - x^\star\right\|^2 \\ \|\widehat{\mathbf{d}}_{k}\|^2 \end{bmatrix} \tag{14}$$

where

$$a_1 = 1 - \frac{\mu\tau\gamma}{2} + \frac{\gamma}{N} \tag{15a}$$

$$a_2 = \left(\frac{2\gamma L}{N} + \frac{4\gamma^2\tau L^2}{N} + 2\rho^2 d^2\gamma\tau + \frac{4\gamma^2\rho^2 d^2\tau}{N}\right)\cdot \\ \cdot \left(36\tau + 16\tau^3\gamma^2(18 + 18\rho^2 d^2)\right) \tag{15b}$$

$$a_3 = \frac{12L^4\|\widehat{\mathbf{V}}^{-1}\|^2\gamma^2\tau^2 N}{1-\delta} + \\ + 16\tau^3\gamma^2 L^2 N\left(\frac{6L^2\|\widehat{\mathbf{V}}^{-1}\|^2\gamma^2\tau(2L^2 + \rho^2 d^2)}{1-\delta} + \right. \\ \left. + \frac{3\gamma^2\tau(1 + 2\|\tilde{L}\|^2)\|\widehat{\mathbf{V}}^{-1}\|^2(L^2 + \rho^2 d^2)}{1-\delta}\right) \tag{15c}$$

$$a_4 = \delta + \frac{12L^4\|\widehat{\mathbf{V}}^{-1}\|^2\gamma^2\tau^2 N}{1-\delta} + \\ + 3\gamma^2\tau\left(\frac{2L^2\|\widehat{\mathbf{V}}^{-1}\|^2(2L^2 + \rho^2 d^2)}{1-\delta} + \right. \\ \left. + \frac{(1 + 2\|\tilde{L}\|^2)\|\widehat{\mathbf{V}}^{-1}\|^2(L^2 + \rho^2 d^2)}{1-\delta}\right)\cdot \\ \cdot \left(36\tau + 16\tau^3\gamma^2(18 + 18\rho^2 d^2)\right) \tag{15d}$$

Denoting $\boldsymbol{\Xi} = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix}$, we know that when the spectral radius of $\boldsymbol{\Xi}$ verifies $\text{sr}(\boldsymbol{\Xi}) < 1$, then $\mathbf{X}_k$ generated by LT-ADMM converges to the optimal solution $\mathbf{X}^*$ linearly. Note that $\text{sr}(\boldsymbol{\Xi}) \leq \|\boldsymbol{\Xi}\|_1 = \max\{a_1 + a_3, a_2 + a_4\}$. Thus, imposing $\|\boldsymbol{\Xi}\|_1 < 1$ together with the previous conditions on γ, we can derive the following set of conditions that guarantee linear convergence:

$$32\tau^3\gamma^2 L\left(2L + 4\gamma\tau L^2 + \rho^2 d^2\tau N + 2\gamma\rho^2 d^2\tau\right) + 8\gamma\tau^2 L < 2\tau, \tag{16a}$$

$$\gamma < \frac{4}{3(d+1)(1-\lambda_2)\rho\tau}, \quad (\mu\tau)/2 - 1/N > 0, \tag{16b}$$

$$4\gamma^2\tau(L^2 + \rho^2 d^2) < \frac{1/4}{\tau - 1}, \gamma < \min\left\{\frac{1}{\tau}, \frac{2(d+1)(1-\lambda_N)\rho}{2\tau}\right\} \tag{16c}$$

$$a_1 + a_3 < 1, \ a_2 + a_4 < 1. \tag{16d}$$

These conditions are satisfied by choosing $\tau > \frac{2}{N\mu}$ and a sufficiently small γ, and linear convergence with rate $\text{sr}(\boldsymbol{\Xi})$ follows. □

IV. Numerical Results

In this section we present numerical results comparing the proposed LT-ADMM with state-of-the-art alternative methods, and evaluate its performance. We apply the algorithms on a *classification task*, characterized by the local costs

$$f_i(x) = \sum_{h=1}^{m_i} \log\left(1 + \exp(-b_i^h a_i^h x)\right) + \frac{\epsilon}{2}\|x\|^2 \tag{17}$$

with $a_i^h \in \mathbb{R}^n$ and $b_i^h \in \{-1, 1\}$, and regularization weight $\epsilon = 0.075$. We use a ring network with $N = 10$, and assign $m_i = 150$ randomly generated data points to each agent. All simulations are implemented in `tvopt` [20].

A. Comparison

We start by comparing LT-ADMM with other distributed optimization methods that allow for local training, namely LED [11], LU-GT [12], K-GT [14], and RandCom [13]. We chose $\tau = 5$ for all algorithms, and hand-tuned their step-sizes to get optimal performance.

In Figure 1 we plot the error trajectories $\{\|X_k - X^*\|\}_{k\in\mathbb{N}}$ of the different algorithms, with the x-axis marking the communication rounds. As we can see, LT-ADMM achieves

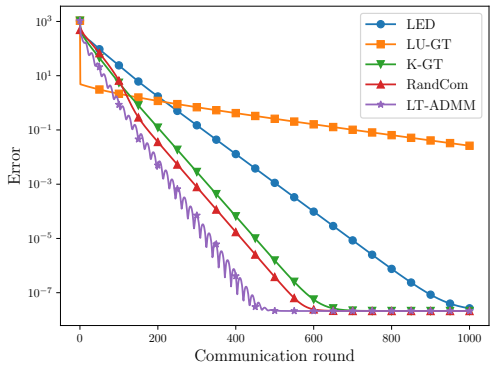

Fig. 1. Comparison of distributed optimization algorithms with local training.

better speed of convergence than all other methods. This is further illustrated by Table I, where we report the empirical convergence rates.

TABLE I

Empirical convergence rates of the compared algorithms.

Algorithm [ref.]	Convergence rate
LED [11]	0.9757
LU-GT [12]	0.9949
K-GT [14]	0.9629
RandCom [13]	0.9595
LT-ADMM [this work]	0.950852

B. LT-ADMM performance

In this section we explore the performance of LT-ADMM for different choices of its parameters, and over networks with different topologies.

The parameters that characterize LT-ADMM are the penalty ρ, the number of local epochs τ, and the local step-size γ. First of all, we remark that during local training, the agents apply τ steps of gradient descent to the regularized cost (cf. (3))

$$\tilde{f}_i(x) := f_i(x) + \frac{\rho|\mathcal{N}_i|}{2}\left\|x - \sum_{j\in\mathcal{N}_i} z_{ij,k}/(\rho|\mathcal{N}_i|)\right\|^2.$$

By Assumption 1, this implies that $\tilde{f}_i(x)$ is $(\mu+\rho|\mathcal{N}_i|)$-strongly convex and $(L+\rho|\mathcal{N}_i|)$-smooth. Therefore, for each $\rho > 0$ choosing the step-size $\gamma = 2/(\mu + L + 2\rho|\mathcal{N}_i|)$ we optimize the convergence rate of the local training gradient descent [19, Theorem 2.1.15]. The remaining two parameters can be freely tuned, and Table II reports how their values affects the (empirical) convergence rate. Let us focus first on

TABLE II

EMPIRICAL CONVERGENCE RATES OF LT-ADMM FOR DIFFERENT CHOICES OF PARAMETERS. "N.C." INDICATES THAT THE ALGORITHM DID NOT CONVERGE.

$\tau =$	1	5	10	50	100
$\rho = 10^{-3}$	0.9942	0.9858	0.9855	0.9895	0.9901
$\rho = 10^{-2}$	0.9915	0.9770	0.9825	0.9875	0.9880
$\rho = 10^{-1}$	0.9883	0.9599	0.9286	0.9318	0.9317
$\rho = 1$	0.9832	0.9662	0.9634	0.9157	0.9157
$\rho = 5$	n.c.	0.9875	0.9900	0.9781	0.9782

the choice of the number of local training steps τ. Intuitively, one may think that the larger τ is, the faster the convergence, as the local proximal computation is approximated to a better degree. However, this does not seem to be the case; indeed, the convergence rates along the first and second row do not decrease monotonically. A similar behavior can be observed reading along the columns in Table II, which indicate that there is an optimal value for ρ. Finally, we observe that for some (ρ, τ) combinations ($(5, 1)$ in the table) the algorithm diverges,in accordance with Theorem 1 which limits the parameters for which convergence is verified.

The previous results were derived for a ring graph, which satisfies Assumption 2 by being regular. We conclude this section by evaluating the performance of LT-ADMM for different topologies. In Figure 2 we report the error trajectory for different networks, and in Table III we report the empirical convergence rate. We remark that LT-ADMM convergence also on non-regular graph topologies, which will be a worthwhile future direction of research. Moreover, we notice that the graph being more connected does not automatically translate in a smaller convergence rate; this aspect will also be explored in future research.

C. Additional comparisons

In this section, we turn back to comparing LT-ADMM with the state-of-the-art alternatives, in different challenging

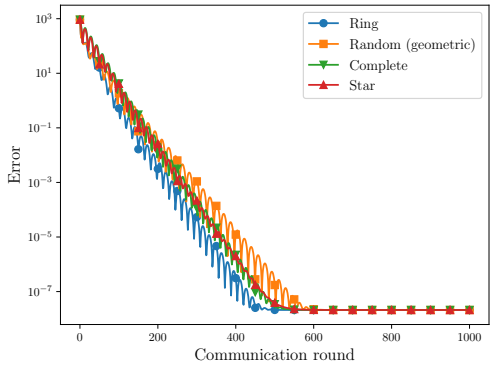

Fig. 2. Performance of LT-ADMM applied to different network topologies.

TABLE III

EMPIRICAL CONVERGENCE RATES OF LT-ADMM WITH DIFFERENT NETWORK TOPOLOGIES.

Topology	Convergence rate
Ring	0.950890
Random (geometric)	0.960377
Complete	0.952770
Star	0.953407

scenarios.

In decentralized learning, communications are often expensive due to the size of the locally trained models being exchanged between agents [21]. For this reason, quantization or compression is often applied in practice to reduce the packet size, and it is important to evaluate performance in such scenarios. To this end, we run a comparison of the algorithms over a network where agents quantize communications with $q(x) = \Delta\left\lfloor \frac{x}{\Delta} \right\rceil$, where $\lfloor\cdot\rceil$ rounds to the nearest integer and $\Delta = 10^{-4}$. In Figure 3 we report the error trajectory for the different algorithms. Owing to the imperfect

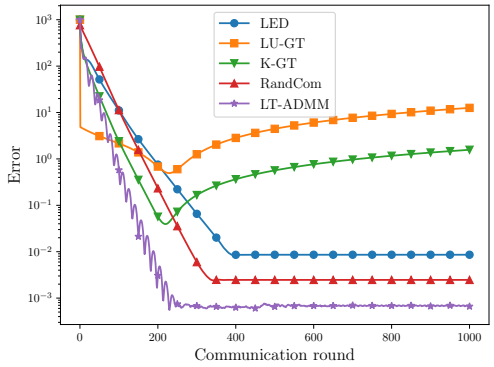

Fig. 3. Comparison over a network with quantized communications.

communications, the algorithms can at best reach a neighborhood of the optimal solution, as expected from the theory [22]. LT-ADMM, alongside LED and RandCom exhibit this behavior. On the other hand, the gradient tracking-based LU-GT and K-GT seem to diverge in the presence of quantized communications; this is a known issue for some gradient tracking schemes [23].

Another challenging feature of learning problems is that the local costs are often defined on large data-sets [3], with $m_i \gg 1$ in (17). Therefore, the computation of full local gradients may be too expensive, and agents need to resort to computing stochastic gradients, which use a random subset of the local data-set. In Figure 4 we report the error trajectory when the agents use stochastic gradients computed on a batch of $B = m_i/5$ data points. We can see that all

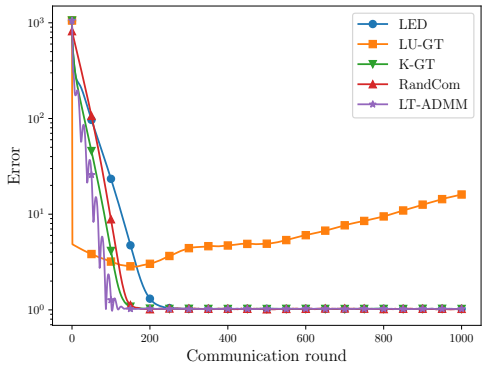

Fig. 4. Comparison using stochastic gradients with a batch of $B = m_i/5$ data points.

algorithms excluding LU-GT converge to a neighborhood of the optimal solution, while LU-GT seems to suffer from the same sensitivity to perturbations exhibited in the presence of quantization.

V. Conclusions

In this paper we have proposed a novel algorithm for distributed learning, LT-ADMM. The algorithm is designed with the goal of allowing multiple local training steps per each communication round. This is especially important in many learning applications where communications are expensive. We have analyzed the convergence of the algorithm under graph regularity assumption, and shown how the use of local training does not impact the learned model accuracy. We have further compared LT-ADMM to the state of the art in different challenging scenarios, and for different graph topologies (including non-regular ones). These results show the promising performance of LT-ADMM, and point the way for future theoretical developements, starting with the relaxation of graph regularity.

References

[1] D. K. Molzahn, F. Dorfler, H. Sandberg, S. H. Low, S. Chakrabarti, R. Baldick, and J. Lavaei, "A Survey of Distributed Optimization and Control Algorithms for Electric Power Systems," *IEEE Transactions on Smart Grid*, vol. 8, no. 6, pp. 2941–2962, Nov. 2017.

[2] A. Nedić and J. Liu, "Distributed Optimization for Control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, no. 1, pp. 77–103, May 2018.

[3] T. Gafni, N. Shlezinger, K. Cohen, Y. C. Eldar, and H. V. Poor, "Federated Learning: A signal processing perspective," *IEEE Signal Processing Magazine*, vol. 39, no. 3, pp. 14–41, May 2022.

[4] G. Notarstefano, I. Notarnicola, and A. Camisa, "Distributed Optimization for Smart Cyber-Physical Networks," *Foundations and Trends® in Systems and Control*, vol. 7, no. 3, pp. 253–383, 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 *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, A. Singh and J. Zhu, Eds., vol. 54. Fort Lauderdale, FL, USA: PMLR, Apr. 2017, pp. 1273–1282.

[6] M. Grudzień, G. Malinovsky, and P. Richtarik, "Can 5th Generation Local Training Methods Support Client Sampling? Yes!" in *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, F. Ruiz, J. Dy, and J.-W. van de Meent, Eds., vol. 206. PMLR, Apr. 2023, pp. 1055–1092.

[7] X. Zhang, M. Hong, S. Dhople, W. Yin, and Y. Liu, "FedPD: A Federated Learning Framework With Adaptivity to Non-IID Data," *IEEE Transactions on Signal Processing*, vol. 69, pp. 6055–6070, 2021.

[8] K. Mishchenko, G. Malinovsky, S. Stich, and P. Richtarik, "ProxSkip: Yes! Local Gradient Steps Provably Lead to Communication Acceleration! Finally!" in *Proceedings of the 39th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds., vol. 162. PMLR, Jul. 2022, pp. 15 750–15 769.

[9] A. Mitra, R. Jaafar, G. J. Pappas, and H. Hassani, "Linear Convergence in Federated Learning: Tackling Client Heterogeneity and Sparse Gradients," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 14 606–14 619.

[10] L. Condat, I. Agarský, G. Malinovsky, and P. Richtárik, "TAMUNA: Doubly Accelerated Federated Learning with Local Training, Compression, and Partial Participation," May 2023.

[11] S. A. Alghunaim, "Local Exact-Diffusion for Decentralized Optimization and Learning," Feb. 2023.

[12] E. D. Hien Nguyen, S. A. Alghunaim, K. Yuan, and C. A. Uribe, "On the Performance of Gradient Tracking with Local Updates," in *2023 62nd IEEE Conference on Decision and Control (CDC)*. Singapore, Singapore: IEEE, Dec. 2023, pp. 4309–4313.

[13] L. Guo, S. A. Alghunaim, K. Yuan, L. Condat, and J. Cao, "RandCom: Random Communication Skipping Method for Decentralized Stochastic Optimization," Oct. 2023.

[14] Y. Liu, T. Lin, A. Koloskova, and S. U. Stich, "Decentralized Gradient Tracking with Local Steps," Jan. 2023.

[15] N. Bastianello, R. Carli, L. Schenato, and M. Todescato, "Asynchronous Distributed Optimization Over Lossy Networks via Relaxed ADMM: Stability and Linear Convergence," *IEEE Transactions on Automatic Control*, vol. 66, no. 6, pp. 2620–2635, Jun. 2021.

[16] R. A. Horn, R. A. Horn, and C. R. Johnson, *Matrix analysis*. Cambridge university press, 1990.

[17] O. N. Bria, "John g. proakis and dimitris g. manolakis, digital signal processing. principles, algorithms, and aplications," *Journal of Computer Science and Technology*, vol. 1, no. 1, p. 1, 1999.

[18] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *International conference on machine learning*. PMLR, 2020, pp. 5132–5143.

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

[20] N. Bastianello, "tvopt: A Python Framework for Time-Varying Optimization," in *2021 60th IEEE Conference on Decision and Control (CDC)*, 2021, pp. 227–232.

[21] Z. Zhao, Y. Mao, Y. Liu, L. Song, Y. Ouyang, X. Chen, and W. Ding, "Towards efficient communications in federated learning: A contemporary survey," *Journal of the Franklin Institute*, p. S0016003222009346, Jan. 2023.

[22] N. Bastianello, L. Madden, R. Carli, and E. Dall'Anese, "A Stochastic Operator Framework for Optimization and Learning with Sub-Weibull Errors," Jun. 2023, arXiv:2105.09884 [cs, eess, math]. [Online]. Available: http://arxiv.org/abs/2105.09884

[23] M. Bin, I. Notarnicola, and T. Parisini, "Stability, Linear Convergence, and Robustness of the Wang-Elia Algorithm for Distributed Consensus Optimization," in *2022 IEEE 61st Conference on Decision and Control (CDC)*. Cancun, Mexico: IEEE, Dec. 2022, pp. 1610–1615.