On the Benefits of Multiple Gossip Steps in Communication-Constrained Decentralized Federated Learning

Abolfazl Hashemi†, Anish Acharya*, Rudrajit Das*, Haris Vikalo, Sujay Sanghavi, and Inderjit Dhillon

Abstract—Federated learning (FL) is an emerging collaborative machine learning (ML) framework that enables training of predictive models in a distributed fashion where the communication among the participating nodes are facilitated by a central server. To deal with the communication bottleneck at the server, decentralized FL (DFL) methods advocate rely on local communication of nodes with their neighbors according to a specific communication network. In DFL, it is common algorithmic practice to have nodes interleave (local) gradient descent iterations with gossip (i.e. averaging over the network) steps. As the size of the ML models grows, the limited communication bandwidth among the nodes does not permit communication of full-precision messages; hence, it is becoming increasingly common to require that messages be *lossy, compressed* versions of the local parameters. The requirement of communicating compressed messages gives rise to the important question: *given a fixed communication budget, what should be our communication strategy to minimize the (training) loss as much as possible?* In this paper, we explore this direction, and show that in such compressed DFL settings, there are benefits to having *multiple* gossip steps between subsequent gradient iterations, even when the cost of doing so is appropriately accounted for, e.g. by means of reducing the precision of compressed information. In particular, we show that having $\mathcal{O}(\log \frac{1}{\epsilon})$ gradient iterations with constant step size - and $\mathcal{O}(\log \frac{1}{\epsilon})$ gossip steps between every pair of these iterations - enables convergence to within ϵ of the optimal value for a class of non-convex problems that arise in the training of deep learning models, namely, smooth non-convex objectives satisfying Polyak-Łojasiewicz condition. Empirically, we show that our proposed scheme bridges the gap between centralized gradient descent and DFL on various machine learning tasks across different network topologies and compression operators.

Index Terms—federated learning, decentralized learning, communication-constrained distributed optimization, compressed communication, nonconvex optimization.

1 Introduction

Collaborative machine learning (ML) methods such as federated learning (FL) [1] are among the fastest growing technological advances that find applications in numerous parallel and distributed systems. In such scenarios, there are a large number of clients (e.g. mobile phones or sensors) each with their own data and resources, and there is typically a central server (i.e., cloud) whose goal is to manage the training of a centralized model using the decentralized client data. Given the ever-increasing number of nodes in distributed systems, decentralized FL (DFL) schemes which allow each client to exchange messages only with their neighbors without exchanging their local data, show great potential in terms of scalability FL.

DFL can be thought of as an optimization task over a network with n client nodes where the objective function is possibly nonconvex [2]. Formally,

$$\min_{\mathbf{x}\in\mathbb{R}^d} \left[f(\mathbf{x}) := \sum_{i=1}^{n} f_i(\mathbf{x}) \right], \tag{1}$$

where $f_i : \mathbb{R}^d \to \mathbb{R}$ for $i \in [n] := \{1, ..., n\}$ is the local objective function of the i^{th} client. The goal of the clients in the network is to collaboratively solve the above optimization problem by passing messages over a graph that connects them [1], [3]. The optimization task in (1) arises in many collaborative ML tasks such as object and pedestrian detection in connected autonomous cars.

DFL is often facilitated by communication of clients' local model parameters over a network that governs their communication capabilities. Compared to a centralized methods, FL and DFL enable locality of data storage and model updates which in turn offers computational advantages by delegating computations to multiple clients, and further promotes preservation of privacy of user information [1].

As the size of ML models grows, exchanging information across the network becomes a major challenge in DFL and distributed optimization in general. It is therefore imperative to design communication-efficient strategies which reduce the amount of communicated data by performing compressed communication while at the same time, despite the use of compressed communication, achieve a convergence proper-

Manuscript received September 1, 2021. *Abolfazl Hashemi is with the School of Electrical and Computer Engineering, Purdue University. Anish Acharya, Haris Vikalo and Sujay Sanghavi are with the Department of Electrical and Computer Engineering, The University of Texas at Austin. Rudrajit Das and Inderjit Dhillon are with the Department of Computer Science, The University of Texas at Austin. * denotes equal contribution. † work done while at the Department of Electrical and Computer Engineering, The University of Texas at Austin.* *This work was supported in part by NSF grants ECCS-1809327, CCF-1564000, IIS-1546452 and HDR-1934932.*

ties that are on par with the performance of centralized and distributed methods utilizing uncompressed information.

1.1 Contribution

In this paper, we consider the task of DFL with nonconvex objective (i.e., training loss) functions in communication-constrained settings. In such scenarios, the clients may need to compress their local updates (using, e.g., quantization and/or sparsification) before transmitting them to their neighbors. In particular, our main goal is to answer the following question

Given a fixed communication budget, what should be our communication strategy to minimize the (training) loss as much as possible?

To this end, we demonstrate that in DFL, given a fixed communication budget per round, performing multiple gossiping/consensus steps – a common practice in decentralized optimization [2], [3] – in addition to aggressive compression, can yield a faster convergence rate (almost linear) compared to the standard approach of performing just one high-precision gossiping step. We argue that this faster convergence rate may result in a smaller loss/error as a function of the *total* number of communicated bits. Specifically, we will demonstrate that given a *fixed communication budget per iteration*, having multiple consensus (aka gossiping) steps with lower precision is a better alternative compared to having just one consensus step with higher precision. Motivated by this result, we theoretically study the effect of the number of gossiping steps on the rate of convergence of DFL in a communication-constrained setting. Specific contributions of this work can be summarized as follows:

- We propose **De**centralized **Li**near Learning with **Co**mmunication **Co**mpression (DeLi-CoCo), an iterative DFL algorithm with arbitrary communication compression (both biased and unbiased compression operators) that performs multiple gossip steps in each iteration for faster convergence.
- By employing $Q > 1$ steps of compressed communication after each local gradient update, DeLi-CoCo achieves a linear rate of convergence to a near-optimal solution for smooth nonconvex objectives satisfying the Polyak-Łojasiewicz condition (see Theorem 1). This rate matches the convergence rate of decentralized gradient descent (DGD) [4] – a DFL approach – with no communication compression under much milder conditions. The proposed Q-step gossiping further helps to arbitrarily decrease the sub-optimality radius of the near-optimal solution, thereby improving upon the results of DGD [4] (see Corollary 1.1).
- Our novel theoretical contributions enables us to demonstrate that given a fixed communication budget, increasing Q and decreasing the precision of compression theoretically improves the convergence properties of DeLi-CoCo (see Section 5.1).
- We verify our theoretical results and show the efficacy of the proposed communication strategy for DFL via extensive numerical experiments on both convex and nonconvex DFL tasks, including the task of decentralized classification using deep learning models.

1.2 Organization

The rest of the paper is organized as follows. Section 2 positions our contribution with respect to the related work. Section 3 discusses the notation and overviews the preliminary concepts on distributed optimization. In Section 4, we introduce the communication strategy for DFL based on multiple gossip steps. The theoretical analysis is discussed in Section 5. The empirical evaluation is provided in Section 6 while the concluding remarks are stated in Section 7.

2 Significance and Related Work

Designing efficient algorithms for federated learning is one of the most active area of research in the parallel and distributed system community in recent years [5], [6], [7]. Decentralized federated learning and optimization have drawn significant attention in the past few years due to the increasing importance of privacy and high data communication costs of centralized methods. Decentralized topologies overcome the aforementioned challenges by allowing each client to exchange messages only with their neighbors without exchanging their local data, showing great potential in terms of scalability and privacy-preserving capabilities.

2.1 Consensus with Compressed Communication

While DFL is an emerging topic, the study of decentralized optimization problems dates back to 1980s [8]. The main focus of early research in this area was on the task of average consensus where the goal of a network is to find the average of local variables (i.e., clients' model vectors) in a decentralized manner. Conditions for asymptotic and non-asymptotic convergence of the decentralized average consensus in a variety of settings including directed and undirected time-varying graphs have been established in the seminal works [9], [10]. Recently, [11] proposed a communication-efficient average consensus/gossip algorithm that achieves a linear convergence rate and improves the performance of existing quantized gossip methods [12]. In [11] a stochastic decentralized algorithm for strongly convex and smooth objectives is further developed. Such linearly convergent gossip methods have also recently been extended to the scenario where the communication graph of clients is directed and time-varying [13]. In our work, we aim to study the benefits of performing multiple quantized gossip steps in DFL to reduce the training error given a fixed communication budget, and consider nonconvex learning tasks in our theoretical analysis.

2.2 Decentralized Optimization with Compressed Communication

Distributed optimization is one of the richest topics at the intersection of machine learning, signal processing and control. Consensus/gossip algorithms have enabled distributed optimization of (non)convex objectives (e.g., empirical risk minimization) by modeling the task of decentralized optimization as noisy consensus. Examples include the celebrated distributed (sub)gradient descent algorithms (DGD) [2], [4].

These schemes consider small-scale problems where the clients can communicate uncompressed messages to their neighbors. Designing communication-efficient distributed optimization algorithms is an active area of research motivated by the desire to reduce the communication burden of multi-core and parallel optimization of ML models. Majority of the existing works consider distributed optimization tasks with master-slave architectures where the compression of communication is accomplished by using methods based on sparsification or quantization of gradients [14], [15], [16]. Divergent from these master-slave architectures, FL's properties such as high heterogeneity, partial participation, and periodic communication between the clients and the server, make FL a practically appealing, hard-to-analyze method [1]. Recent FL schemes that promote communication efficiency either focus on compressing the size of the client-to-cloud messages or decreasing the number of communication rounds [13], [16], [17]. In contrast to that line of work, we consider the more general and challenging setting of communication-constrained decentralized federated learning and exploit the error feedback mechanism of [14], [15], [16] as part of our proposed communication strategy to enable compressed message-passing while maintaining a linear convergence rate. More importantly, our focus in this paper is the importance of organizing the communication resources. That is, given a fixed communication budget in DFL, what is the best strategy for the number of consensus steps and the precision of compression in order to achieve a smaller error in terms of the number of communicated bits.

It is worth noting that unlike a majority of decentralized optimization and FL schemes including those with uncompressed communication that require strong convexity to a achieve linear rate, e.g. [3], [4], we only assume the Polyak-Łojasiewicz condition which enables us to analyze nonconvex learning tasks. Our proposed communication strategy results in a linear convergence rate for DFL with compressed communication under the Polyak-Łojasiewicz condition.

3 Preliminaries and Background

In this section, we briefly overview a few important concepts and definitions with regard to the communication network and characteristics of the loss function.

We consider the standard DFL setup [2] where n clients, each having a local function $f_i(.)$, aim to collaboratively reach $\mathbf{x}^* \in \mathcal{X}^* \subset \mathbb{R}^d$, an optimizer of (1). Problem (1) can be written equivalently as [2], [4], [11], [18]

$$\min_{\mathbf{x}_1=\cdots=\mathbf{x}_n} \left[F(X) := \sum_{i=1}^{n} f_i(\mathbf{x}_i) \right], \tag{2}$$

where $\mathbf{x}_i \in \mathbb{R}^d$ is the vector collecting the local parameters of client i, and $X \in \mathbb{R}^{d\times n}$ is a matrix having $\mathbf{x}_i$ as its i^{th} column. Therefore, the goal of the clients in the network is to achieve consensus such that $\mathbf{x}_i = \mathbf{x}^*$ for some $\mathbf{x}^* \in \mathcal{X}^*$; in matrix notation, $X = X^*$, where all the columns of X^* are equal to $\mathbf{x}^*$, i.e. $X^* = \mathbf{x}^*\mathbf{1}^\top$.

To solve (2), each client can communicate only with its neighbors, where the communication in the network is modeled by a graph. Specifically, we assume each node i associates a non-negative weight w_{ij} to any node j in the network, and $w_{ij} > 0$ if and only if node j can communicate with node i, and $w_{ii} > 0$ for all i. Let $W = [w_{ij}] \in [0,1]^{n\times n}$ be the matrix that collects these weights. We call W the mixing or gossip matrix and state some its properties (following [10]) below.

Assumption 1 (Mixing Matrix). *The gossip matrix $W = [w_{ij}] \in [0,1]^{n\times n}$ associated with a connected graph is non-negative, symmetric and doubly stochastic, i.e.*

$$W = W^\top, \qquad W\mathbf{1} = \mathbf{1}. \tag{3}$$

Under this condition, eigenvalues of W can be shown to satisfy $1 = |\lambda_1(W)| > |\lambda_2(W)| \geq \cdots \geq |\lambda_n(W)|$ [10]. Furthermore, $\delta := 1 - |\lambda_2(W)| \in (0,1]$ is the so-called spectral gap of W.

A large spectral gap implies a faster convergence rate of decentralized algorithms. When the graph is fully connected and $\deg(i) = n$, with $W = \mathbf{1}\mathbf{1}^\top/n$, it holds that $\delta = 1$ which in turn implies consensus can be achieved exactly after one iteration of message passing.

Designing the communication network and its associated mixing matrix W with a large spectral gap is an important task and an active area of research in multi-agent systems and DFL (see e.g. [2], [10]) which is beyond the scope of this work. Here, work under the standard consideration that W and its spectral gap δ are known and can be used as inputs of our proposed DFL algorithm.

We now define some commonly assumed properties of the objective function, i.e. the training loss in DFL.[1]

Assumption 2 (Smoothness). *Each local objective function is L_i-smooth, i.e., for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$*

$$f_i(\mathbf{x}) \leq f_i(\mathbf{y}) + (\mathbf{x}-\mathbf{y})^\top \nabla f_i(\mathbf{y}) + \frac{L_i}{2}\|\mathbf{x}-\mathbf{y}\|^2. \tag{4}$$

Also, define $L := \sum_i L_i/n$ and $\hat{L} := \max_i L_i$.

Assumption 3 (Polyak-Łojasiewicz Condition). *The objective function satisfies the Polyak-Łojasiewicz condition (PLC) with parameter μ, i.e. for all $\mathbf{x} \in \mathbb{R}^d$*

$$\|\nabla f(\mathbf{x})\|^2 \geq 2\mu(f(\mathbf{x}) - f^*), \quad \mu > 0, \quad f^* = \min_{\mathbf{x}} f(\mathbf{x}).$$

The Polyak-Łojasiewicz condition implies that when multiple global optima exist, each stationary point of the objective function is a global optimum [19]. This setting enables studies of modern large-scale ML tasks such as training of deep neural networks that are generally nonconvex but are fairly likely to satisfy PLC [20]. It is worth noting that μ-strongly convex functions satisfy PLC with parameter μ – thus, PLC is a weaker assumption than strong convexity.

Convergence of centralized gradient descent under PLC follows a very simple analysis [19]. However, in decentralized federated learning settings with compression, analysis of the existing algorithms, e.g. [3], [4], [11], relies on a key property of strongly convex objectives known as co-coercivity (see Theorem 2.1.11 in [21]). Unfortunately, the results of such analysis do not generalize to PLC settings. In this paper, by performing a novel convergence analysis, we establish convergence of DeLi-CoCo for decentralized

1. $\|.\|$ denotes the Euclidean norm.

nonconvex problems with compressed communication under PLC.

Finally, we characterize the compression operator $\mathcal{C}$ that we use in our DFL algorithm. The following assumption is standard and has been previously made by [11], [16], [22].

Assumption 4 (Contraction Compression). *The compression operator $\mathcal{C}$ satisfies*

$$\mathbb{E}_{\mathcal{C}}\left[\|\mathcal{C}(\mathbf{x})-\mathbf{x}\|^2 \mid \mathbf{x}\right] \leq (1-\omega)\|\mathbf{x}\|^2, \tag{5}$$

for all $\mathbf{x} \in \mathbb{R}^d$ where $0 < \omega \leq 1$ and the expectation is over the internal randomness of $\mathcal{C}$.

Note that $\mathcal{C}$ can be a biased or an unbiased compression operator including:

- Random selection of k out of d coordinates or k coordinates with the largest magnitudes. In this case $\omega = k/d$ [16]. We denote these two by $\mathrm{rand}(\omega)$ and $\mathrm{top}(\omega)$, respectively.
- Setting $\mathcal{C}(\mathbf{x}) = \mathbf{x}$ with probability p and $\mathcal{C}(\mathbf{x}) = 0$ otherwise. In this case $\omega = p$ [11]. We denote this by $\mathrm{rand2}(\omega)$.
- b-bit random quantization (i.e., the number of quantization levels is 2^b) from [23]

$$\mathrm{qsgd}_b(\mathbf{x}) = \frac{\mathrm{sign}(\mathbf{x})\|\mathbf{x}\|}{2^b w}\left\lfloor 2^b \frac{|\mathbf{x}|}{\|\mathbf{x}\|} + \mathbf{u}\right\rfloor, \tag{6}$$

where $w = 1 + \min\{\sqrt{d}/2^b, d/2^{2b}\}$, $\mathbf{u} \sim [0,1]^d$, and $\mathrm{qsgd}_b(\mathbf{0}) = \mathbf{0}$. In this case, $\omega = 1/w$.

4 Compressed Decentralized Learning

In this section, we present our proposed DFL algorithm for solving (2) iteratively in a decentralized manner where the clients are restricted to communicate compressed information (See Fig 1 for the block diagram of the proposed method) . In particular, we aim to develop a scheme that by relying on performing multiple low-precision compressed gossiping steps achieves a smaller error in terms of the number of communicated bits.

The proposed DFL algorithm, DeLi-CoCo (see Algorithm 1), consists of two main subroutines: (i) update of the local variable $\mathbf{x}_i$ via gradient descent, and (ii) exchange of compressed messages between neighboring clients by performing $Q \geq 1$ compressed gossiping steps via employing Choco-gossip [11].

Let $t = 1, \ldots, T$ denote the t^{th} iteration of Algorithm 1 and let $q = 0, \ldots .Q-1$ denote the q^{th} compressed gossiping/consensus step. Each client i maintains three local variables: $\mathbf{x}_{t,i}^{(q)}$, $\mathbf{z}_{t,i}^{(q)}$, and $\mathbf{s}_{t,i}^{(q)}$. Here, $\mathbf{x}_{t,i}^{(q)}$ denotes the vector of current local parameters of node i, while $\mathbf{z}_{t,i}^{(q)}$, and $\mathbf{s}_{t,i}^{(q)}$ are maintained locally to keep track of the compression noise and be used as an error feedback for subsequent iterations, respectively [11], [16]. Consider a matrix notation where we store these quantities as the i^{th} column of matrices $X_t^{(q)}$, $Z_t^{(q)}$, and $S_t^{(q)}$, respectively. At iteration t, each client updates its own parameters by performing a simple gradient descent update according to step 3, where $\eta > 0$ is a constant learning rate specified in Theorem 1. Following the gradient update, we propose to perform Q compressed gossiping steps in to update the local parameters as well as the error feedback variables. This Q-step procedure is a crucial part of Algorithm 1 that enables updated parameters $\mathbf{x}_{t,i}^{(0)}$ to converge to their average value.

Algorithm 1 The proposed DFL Algorithm (DeLi-CoCo)

1: **Input:** stepsize η, consensus stepsize γ, number of gradient iterations T, number of consensus steps per gradient iteration Q, mixing matrix W; initialize $X_0^{(Q)}$.
2: **for** $t = 1, \ldots, T$ **do**
3: $X_t^{(0)} = X_{t-1}^{(Q)} - \eta \nabla F(X_{t-1}^{(Q)})$ (local gradient update) $Z_t^{(0)} = S_t^{(0)} = X_t^{(0)}$
4: **for** $q = 0, 1, \ldots, Q-1$ **do**
5: $S_t^{(q+1)} = S_t^{(q)} + \mathcal{C}(X_t^{(q)} - Z_t^{(q)})W$ (Exchanging messages)
6: $Z_t^{(q+1)} = Z_t^{(q)} + \mathcal{C}(X_t^{(q)} - Z_t^{(q)})$ (Compression error feedback)
7: $X_t^{(q+1)} = X_t^{(q)} + \gamma(S_t^{(q+1)} - Z_t^{(q+1)})$ (Local gossip update)
8: **end for**
9: **end for**

To perform the $(q+1)^{\text{st}}$ gossiping step, each client generates the compressed message $\mathcal{C}(\mathbf{x}_{t,i}^{(q)} - \mathbf{z}_{t,i}^{(q)})$ which in turn is communicated to update $\mathbf{s}_{t,i}^{(q)}$, and then it is further used by the transmitting client as an error feedback to update $\mathbf{z}_{t,i}^{(q)}$ (steps 5 and 6). Then, at $(q+1)^{\text{st}}$ gossiping step, each client performs a gossip update [10] in step 7 with a gossiping/consensus learning rate $0 < \gamma \leq 1$ whose exact value will be specified in Theorem 1. After performing compressed gossiping for Q steps, the t^{th} iteration of Algorithm 1 is complete.

Remark 1. *Let $Q = 1$, $\gamma = 1$, and assume there is no compression, i.e. $\mathcal{C}(\mathbf{x}_{t,i}^{(q)} - \mathbf{z}_{t,i}^{(q)}) = \mathbf{x}_{t,i}^{(q)} - \mathbf{z}_{t,i}^{(q)}$. Then Algorithm 1 reduces to the DGD [4]. If $Q = 1$, $\eta = \mathcal{O}(1/T)$, and clients perform local stochastic gradient updates, the proposed scheme reduces to Choco-SGD [11]. We will show in Section 4 that by performing $Q > 1$ gossiping steps and reducing the precision of compression, Algorithm 1 achieves a smaller training error compared to these schemes, given a fixed communication budged.*

4.1 Practical Considerations

In a scenario where there is negligible latency and synchronization among the clients,Algorithm 1 that relies on multiple compressed gossiping steps achieves a faster convergence rate and also requires fewer total number of bits for communication (see Section 6). With latency and synchronization considerations, decentralized federated learning schemes based on multiple **uncompressed consensus** steps are shown effective in training deep models [24], while the study of their benefits in terms of savings in communication resources has remained an open question until the present paper. Hence, even if synchronization constraints are taken into account, given that we employ compressed gossiping steps, the proposed algorithm leads to significant savings in the total number of communicated bits. In this case, certifying that our algorithm converges faster (with respect to wall-clock time) is difficult without knowing the actual time expended on synchronization and the latency of the communication

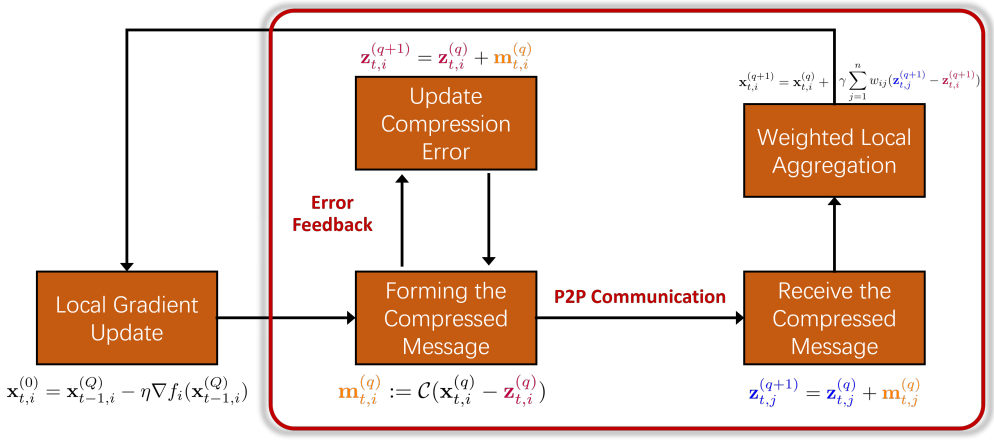
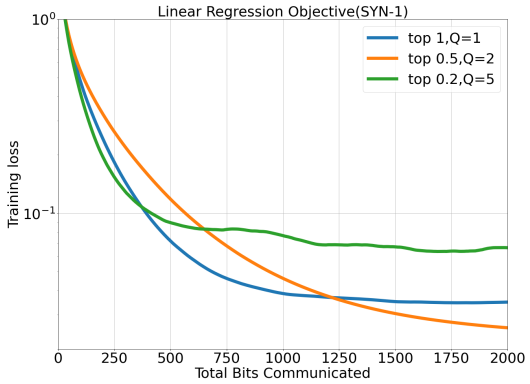
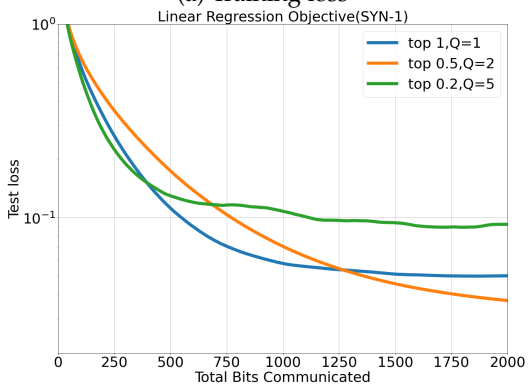


Fig. 1: The diagram of the proposed strategy. After computing the gradient and performing a local gradient update, each node communicates for Q steps with its neighbors using a compressed gossip mechanism while keeping track of the accumulated compress error.

structure, and is left for future work. Nonetheless, in order to simulate a network that may suffer from a high latency issues, and hence the emergence of straggler nodes, we consider a scenario where each client can communicate with its neighbors only 95% of times. That is, with probability 5%, each node may become a straggler at each gossiping step. This implies while it may receive messages from its immediate neighbors, it will not be able to transmit. We consider the linear regression task on SYN-1 using the torus topology with $n = 9$ (see Section 6) and show the test and training errors in Fig. 2. As the figure demonstrates, while $Q = 2$ results in a better performance, due to the straggler effect, increasing the number of gossiping steps to $Q = 5$ suffers from a slow convergence. Therefore, we conclude that for low latency networks moderate Q values, say $Q = 4, 5$ is preferred, while for high latency networks a smaller Q such as $Q = 1, 2$ should be chosen. Thus, in communication-constrained settings, Algorithm 1 with multiple gossiping steps ($Q > 1$) is indeed preferable.



(a) Training loss



(b) Test loss

Fig. 2: Training and test errors for the linear regression task on SYN-1 data with straggler nodes.

5 Convergence Analysis

In this section we analyze the convergence properties of DeLi-CoCo. First, We define the following quantities:

$$\Delta^2 := \max_{\mathbf{x}^* \in \mathcal{X}^*} \sum_{i=1}^{n} \|\nabla f_i(\mathbf{x}^*)\|^2, \quad R_0 := F(X_0^Q) - f^*. \tag{7}$$

The main results of our convergence analysis are summarized in the following theorem, whose proof is provided in the attached supplementary material due to space constraints.

Theorem 1. *Suppose Assumptions 1-4 hold. Define*

$$Q_0 := \left\lceil \log(\bar{\rho}/46) \Big/ \log\left(1 - \frac{\delta\gamma}{2}\right) \right\rceil, \quad \bar{\rho} := 1 - \frac{\mu}{n\hat{L}}, \\ \gamma = \frac{\delta\omega}{16\delta + \delta^2 - 8\delta\omega + (4 + 2\delta)\lambda_{\max}^2(I - W)}. \tag{8}$$

Then, if the nodes are initialized such that $X_0^{(Q)} = \mathbf{0}$, for any $Q > Q_0$ after T iterations the iterates of DeLi-CoCo with $\eta = \frac{1}{\hat{L}}$ satisfy

$$\mathbb{E}_{\mathcal{C}}[F(X_T^{(Q)})] - f^* = \mathcal{O}\Bigg(\frac{\Delta^2 e^{-\frac{\gamma\delta Q}{4}}}{1 - \bar{\rho}} \\ + \left[1 + \frac{nL}{\mu\bar{\rho}}\left(1 + e^{-\frac{\gamma\delta Q}{4}}\right)\right] R_0 \rho^T\Bigg). \tag{9}$$

Remark 2. *Note that Theorem 1 implies that there exists an implicit limit on the compression level since as $\omega \to 0$, the minimum value of Q (i.e. Q_0) tends to infinity. Further, note that as $\omega \to 0$, γ (the consensus learning rate) tends to 0; implying there is hardly any message-passing and mixing among the nodes.*

5.1 Implications

The result of Theorem 1 implies having multiple consensus (aka gossiping) steps with aggressive compression results in a smaller error in terms of the number of communicated bits. In fact, we can observe this via a simple experiment: Let us consider a decentralized federated learning scenario where we aim to collaborative solve a nonlinear regression task over a network of resource-constrained clients (see Section 6 for more details). We depict the training error versus the communicated bits in Figure 3. As the figure shows increasing the number of consensus steps (denoted by Q) with a lower quantization precision requires fewer communicated bits to achieve a target accuracy.

To see the verification of this result from Theorem 1, in Lemma 3 in the supplementary we show for $\omega > 10^{-3}$, which is practical lower bound for the compression/quantization rate in practice, the convergence rate depends on

$$e^{-\frac{\gamma\delta Q}{4}} \leq e^{-\frac{\delta^2 Q\omega^{3/4}}{656}}.$$

We shall analyze this upper bound to motivate the benefit of advocating a higher Q by the proposed communication strategy. Consider two pairs of (Q_1, ω_1) and $(Q_1 \times c, \omega_1/c)$ where $c > 0$ is an integer that determines the allocation of communication resources, and (Q_1, ω_1) satisfies the conditions stated in Theorem 1. The proposed scheme for both of these pairs require the same amount of communication budget. Upon defining

$$g(c) := e^{-\delta^2 cQ(\omega/c)^{3/4}} = e^{-\delta^2 c^{1/4} Q\omega^{3/4}},$$

in Figure 4 we depict the value of $g(c)$ versus c for various values of the spectral gap δ. As the figure shows $g(c)$ is decreasing in c meaning that for a fixed communication budget, increasing the number of gossiping steps Q and decreasing the compression parameter ω **theoretically** results in improved convergence properties given that both terms in (9) incur smaller values. Intuitively, this is expected since the rate depends on the product $Q\omega^{3/4}$. This theoretical result hence shows the advantage of Algorithm 1 that advocates the use of multiple gossiping steps to achieve a smaller error in terms of the number of communicated bits.

5.2 Further Discussions

We further highlight the following remarks:

1. Comparison to DGD: We compare our result to the prior work in [4], [25] that assume exact communication. First, in contrast to [4], [25], our analysis is carried out under PLC without assuming (restricted) strong convexity. The radius of the near-optimal neighborhood in [4] (see Theorem 4 there) is proportional to Δ/δ while in our case, by using the proposed Q- step compressed gossiping procedure, the radius is proportional to $\Delta^2(1-\delta)^{\frac{Q}{2}}$; in fact, we can make the bound arbitrarily small by performing a sufficiently large number of gossiping steps Q (see Corollary 1.1).

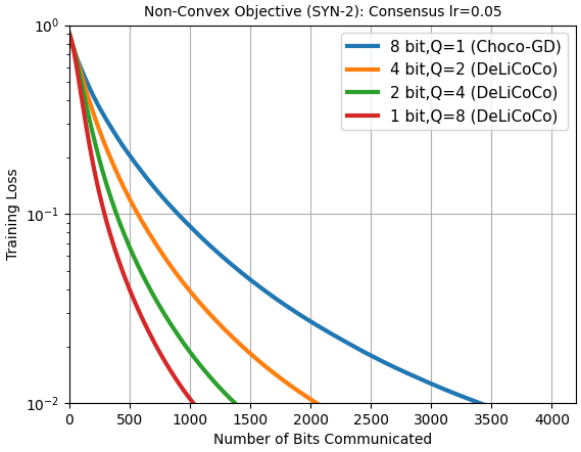


Fig. 3: Empirical effect of increasing the number of gossiping steps on a non-convex nonlinear regression task given a fixed communication budget per iteration.

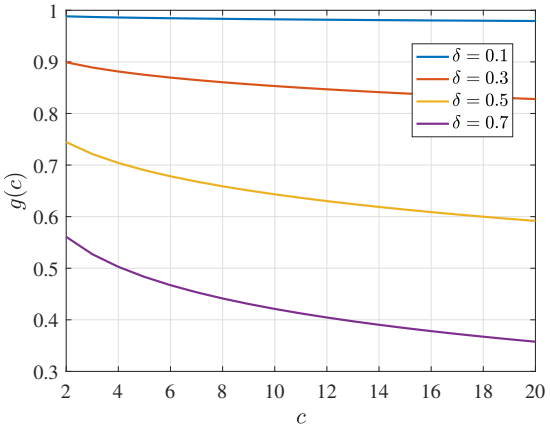


Fig. 4: Variation of $g(c)$ vs. c for different values of δ.

2. Effect of Compression: Our results reveal that compression of messages using contraction operators can be thought of as weakening the connectivity property of the communication graph by inducing spectral gap $\delta' = \delta\omega^{1/4}$. As ω approaches zero, the consensus learning rate decreases. Hence, as per intuition, a larger Q is required to satisfy the conditions in the statement of Theorem 1.

3. Almost Linear Convergence: Our analysis further reveals that at the cost of increased number of rounds of communication, the suboptimality radius can be arbitrarily reduced. In particular, $\mathbb{E}_{\mathcal{C}}[F(X_t^{(Q)})] - f^* \leq \epsilon$ accuracy can be achieved after $\mathcal{O}(\log^2(1/\epsilon))$ rounds of communication by setting $Q = T = \log(1/\epsilon)$. However, in practice it suffices to use a small Q to achieve a competitive performance compared to centralized and decentralized schemes with no compression.

4. Overparameterization: Consider the case that (1) corresponds to a decentralized regression or classification task wherein the model architecture is expressive enough to completely fit or *interpolate* the training data distributed among the clients [26], e.g. in the case of over-parameterized neural networks or functions satisfying a certain growth condition [27]. Then any stationary point of f will also be a stationary point of each of the f_i's and thus $\Delta^2 = 0$. Therefore, in

this setting and under PLC, Deli-CoCo converges exactly at a linear rate of $\mathcal{O}(\log(1/\epsilon))$ by setting Q to be a constant independent of ϵ.

Corollary 1.1. *Instate the notation and hypotheses of Theorem 1. In order to achieve $\mathbb{E}_{\mathcal{C}}[F(X_T^{(Q)})] - f^* \leq \epsilon$, Deli-CoCo requires $\tau = \mathcal{O}(\log^2(1/\epsilon))$ rounds of communication if $\Delta \neq 0$, and $\tau = \mathcal{O}(\log(1/\epsilon))$ if $\Delta = 0$.*

We emphasize that this result is new and to our knowledge, DeLi-CoCo is the first algorithm attaining a linear convergence rate for decentralized nonconvex FL with compressed communication in the interpolation regime. Notice that linear convergence even in the centralized setting necessitates $T = \mathcal{O}(\log 1/\epsilon)$. In the decentralized setting under strong convexity (SC), without using techniques such as gradient tracking [18], DGD based FL schemes use either $\eta = \mathcal{O}(\frac{1}{t})$ to have $\mathcal{O}(1/\epsilon)$ rounds of communication (e.g. Choco-GD or DGD [2], [11]), or a fixed stepsize (independent of T) to achieve linear convergence to a *near-optimal* solution [4]. Corollary 1.1 states without over-parameterization $Q = \mathcal{O}(\log 1/\epsilon)$ enables our algorithm to converge to an ϵ-accurate solution under PLC with $\mathcal{O}(\log^2 1/\epsilon)$ rounds of communication, which is a significant improvement over $\mathcal{O}(1/\epsilon)$ for DGD based schemes with decaying step-size.

5. Results Under Strong Convexity: Since PLC is implied by strong convexity, Theorem 2 provides a convergence rate for strongly convex and smooth objectives. In Theorem 2 by explicitly exploiting the strong convexity of the individual f_i's, we provide an alternative result that improves the dependency of the rates on Q and n.

Theorem 2. *Suppose Assumptions 1,2, and 4 hold. Further, assume each f_i is strongly convex with parameter μ_i, and define $\mu = \sum_i \mu_i/n$, $\hat{\mu} = \min_i \mu_i$ and $D_0 := \|X_0^{(Q)} - X^*\|^2$. Define*

$$Q_0 := \left\lceil \log(\ell/46) \Big/ \log\left(1 - \frac{\delta\gamma}{2}\right) \right\rceil, \quad \ell := 1 - \frac{\hat{\mu}}{\hat{L}}, \qquad \gamma = \frac{\delta\omega}{16\delta + \delta^2 - 8\delta\omega + (4+2\delta)\lambda_{\max}^2(I-W)}. \tag{10}$$

Then, if the nodes are initialized such that $X_0^{(Q)} = \mathbf{0}$, for any $Q > Q_0$ after T iterations the iterates of DeLi-CoCo with $\eta = \frac{1}{\hat{L}}$ satisfy

$$\mathbb{E}_{\mathcal{C}}\|X_T^{(Q)} - X^*\|^2 = \mathcal{O}\Big(\frac{\Delta^2 e^{-\frac{\gamma\delta Q}{2}}}{\hat{\mu}^2} + \left[1 + \frac{Te^{-\frac{\gamma\delta Q}{4}}}{\ell^2(\hat{L}-\hat{\mu})}\right] D_0 \ell^T\Big). \tag{11}$$

5.3 Proof Outline

Here, we briefly discuss the technical difficulties and the main ideas of the proof. Due to space limitations, the details are in the attached supplementary material.

Technical challenges

To show the advantage of employing multiple gossip steps with compressed communication in the nonconvex setting under PLC – a setting that is being analyzed for the first time (in decentralized FL) – we develop a novel analysis technique. In this technique – divergent from the existing works, e.g., [3], [11], [28] – we model the task at hand as a constrained optimization problem with a specific inexact projection tailored towards decentralized optimization (i.e., approximating $\mathcal{P}_{\mathcal{L}}(.)$, the projection onto $\mathcal{L}$, the linear subspace of $d\times n$ matrices having identical columns, see Section 1 of the supplementary material). Note that because of inexact projection we cannot rely on the existing convergence proof of projection-free first-order methods under PLC, or proximal methods under proximal-PLC stated in [29]. Instead, we utilize the specific structure of the inexact projection that we define and its implications, such as $\mathcal{P}_{\mathcal{L}}(\nabla F(X^*)) = \mathbf{0} \in \mathcal{L}$, to carry out the proof.

We now discuss the main steps of the proof.

Perturbed iterate analysis

Our proof relies on analyzing the (virtual) average iterates

$$\bar{X}_{t+1} = \mathcal{P}_{\mathcal{L}}(\bar{X}_t - \eta\nabla F(\bar{X}_t)), \quad \bar{X}_t = [\bar{\mathbf{x}}_t, \ldots, \bar{\mathbf{x}}_t], \qquad \mathcal{P}_{\mathcal{L}}(\nabla F(\bar{X}_t)) := \frac{1}{n}[\nabla f(\bar{\mathbf{x}}_t), \ldots, \nabla f(\bar{\mathbf{x}}_t)]. \tag{12}$$

where $\bar{\mathbf{x}}_t = \frac{1}{n}\sum_{i=1}^n \mathbf{x}_{t,i}^{(Q)}$ is the average parameters at iteration t. Then, we show the iterates of DeLi-CoCo satisfy

$$X_{t+1}^{(q)} = \mathcal{P}_{\mathcal{L}}(\bar{X}_t - \eta\nabla F(\bar{X}_t)) + E_t^{(q)} \tag{13}$$

for some error matrix $E_t^{(q)} \in \mathbb{R}^{d\times n}$.

Using a new result (Lemma 1) we show that the average iterates converge linearly under PLC, i.e.,

$$F(\bar{X}_t) - f^* \leq [F(\bar{X}_0) - f^*]\left(1 - 2\frac{\mu}{n}\eta(1 - \frac{L\eta}{2})\right)^t. \tag{14}$$

despite the projection and the fact that the global objective is nonconvex. Evidently, given this result, we further need to derive an upper bound on the error term $E_t^{(q)}$.

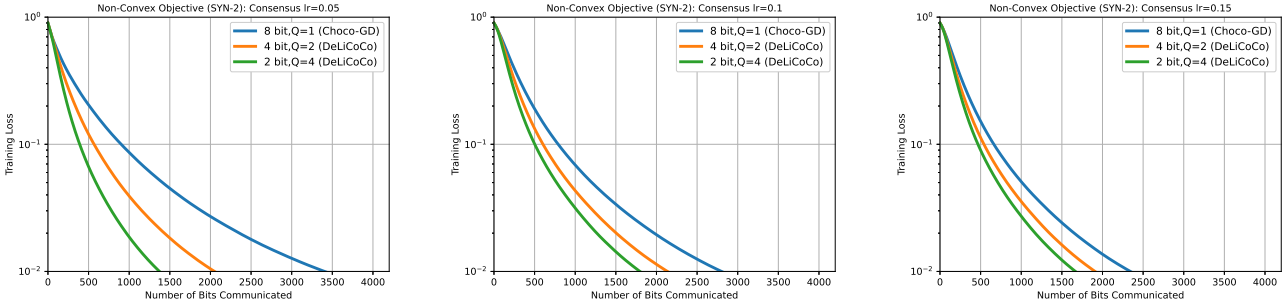
Bounding the error: We first bound the error term of the t^{th} iteration according to

$$\mathbb{E}_{\mathcal{C}}\|E_t^{(Q)}\|^2 \leq e_t^2 := \mathbb{E}_{\mathcal{C}}\|X_t^{(Q)} - \bar{X}_t\|^2 + \mathbb{E}_{\mathcal{C}}\|X_t^{(Q)} - Z_t^{(Q)}\|^2. \tag{15}$$

To analyze e_t^2 in Lemma 4 we leverage the linear convergence of $\{\bar{X}_t\}$ and the gossiping steps with error feedback [10], [11] to establish in Lemma 3 that with γ as in Theorem 1,

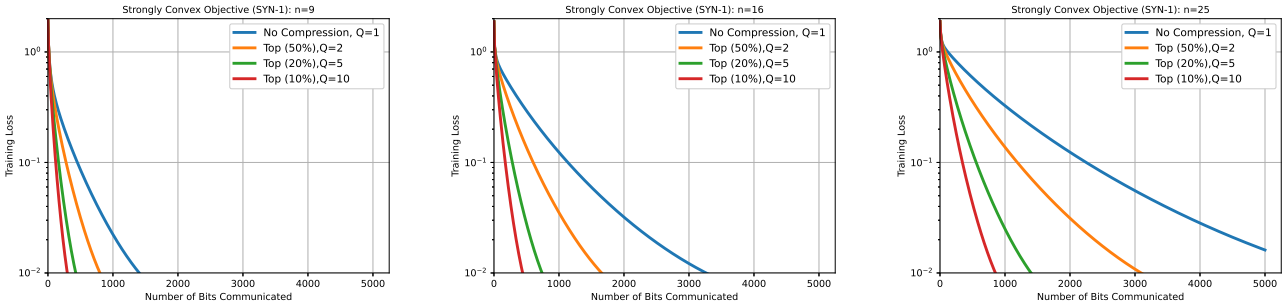
$$\begin{aligned} e_t^2 &\leq e^{-\frac{\delta\gamma Q}{2}}\left(\mathbb{E}_{\mathcal{C}}\|X_t^{(0)} - \bar{X}_t\|^2 + \mathbb{E}_{\mathcal{C}}\|X_t^{(0)} - Z_t^{(0)}\|^2\right) \\ &\leq e^{-\frac{\delta^2\omega Q}{82}}\left(\mathbb{E}_{\mathcal{C}}\|X_t^{(0)} - \bar{X}_t\|^2 + \mathbb{E}_{\mathcal{C}}\|X_t^{(0)} - Z_t^{(0)}\|^2\right). \end{aligned}$$

Upon establishing this last result we argue that the sequence e_t^2 and in turn the error term $E_t^{(Q)}$ converge linearly to $\mathcal{O}(\eta^2\Delta^2 e^{-\frac{\delta\gamma Q}{2}})$ with the same rate as in (14). Finally, we employ the smoothness and PLC assumptions on F to establish a recursive expression on the function sub-optimality $F(X_t^{(Q)}) - f^*$. However, given the inexact projection in the iterates $X_t^{(Q)}$, this recursive expression is involved. Nonetheless, we show that using the choice $\eta = 1/\hat{L}$ together with specific properties of $\mathcal{P}_{\mathcal{L}}(.)$, by using the Young's inequality and the variational characterization of the projection [21] we can judiciously bound the additional cross terms and establish the proof.



(a) $\gamma = 0.05$ (b) $\gamma = 0.1$ (c) $\gamma = 0.15$

Fig. 5: Effect of different (Q, b) pairs (where b denotes the number of bits in qsgd) such that $Qb = 8$, on the total number of bits communicated for SYN-2, with three different consensus learning rates γ. In all three plots, torus topology is used with $n = 16$, ℓ_2 regularization value = 0.001, and $\eta = 0.1$.



(a) $n = 9$ (b) $n = 16$ (c) $n = 25$

Fig. 6: Effect of different (Q, ω) pairs (where ω denotes the percentage of largest magnitude co-ordinates retained in the top-k quantization) such that $Q\omega = 100$, on the total number of bits communicated for SYN-1. We consider the torus topology with three different values of n. In all three plots, $\gamma = 0.05$, ℓ_2 regularization value = 0.001, and $\eta = 0.1$.

6 Numerical Experiments

We start our extensive empirical analysis by verifying our theoretical results on common regression and classification problems. Note that for these tasks Assumptions 1,2, and 4 hold. Afterwards, we show the efficacy of our method in a federated learning setting with partial participation and periodic communication, which can be thought of as a decentralized DL setting with a time-varying communication graph (see Figure 11).

6.1 Verifying the Theory

Following [11], for all the experiments we plot the sub-optimality, i.e. $f(\bar{\mathbf{x}}_t) - f^*$ against the number of local gradient computations (or steps). Here, f^* is the optimal value obtained by running vanilla gradient descent with the entire data on a single machine – we shall refer to this setting as "Centralized GD". We consider the top(k) and qsgd compression schemes and consider the ring and torus topologies to represent the communication graph of the network (see Fig. 11 for an example of a torus graph with 16 nodes). All plots are averaged over 3 independent runs. Before describing our experimental set-up, we describe the tasks and the datasets.

6.1.1 Tasks and Datasets

Let $\{s_1^{(i)}, \ldots, s_{n_i}^{(i)}\}$ denote the samples being processed in the i^{th} node where n_i is the total number of samples in the i^{th} node. Then, $f_i(\mathbf{x}) = \frac{1}{n_i}\sum_{j=1}^{n_i} \ell(\mathbf{x}, s_j^{(i)})$, where $\ell(.)$ denotes the loss function of the tasks that we explain next.

Linear Regression: We train a linear regression model on $m = 10000$ synthetic data samples $\{(\mathbf{a}_i, y_i)\}_{i=1}^m$ generated according to $y_i = \langle \boldsymbol{\theta}^*, \mathbf{a}_i \rangle + e_i$, where $\boldsymbol{\theta}^* \in \mathbb{R}^{2000}$, the i^{th} input $\mathbf{a}_i \sim \mathcal{N}(\mathbf{0}, I_{2000})$, and noise $e_i \sim \mathcal{N}(0, 0.05)$. We refer to this dataset as SYN-1. Here, we use the squared loss function with ℓ_2-regularization.

Non-Convex Non-Linear Regression: We train a non-linear regression model on $m = 10000$ synthetic data samples $\{(\mathbf{a}_i, y_i)\}_{i=1}^m$ generated as $y_i = \text{relu}(\langle \boldsymbol{\theta}^*, \mathbf{a}_i \rangle) + e_i$, where $\boldsymbol{\theta}^* \in \mathbb{R}^{2000}$, the i^{th} input $\mathbf{a}_i \sim \mathcal{N}(\mathbf{0}, I_{2000})$, $e_i \sim \mathcal{N}(0, 0.05)$ and $\text{relu}(z) = \max(z, 0)$ (i.e. the standard ReLU function). We call this synthetic dataset SYN-2 henceforth. We model this task as training a one-layer neural network having ReLU activation with the squared loss function and ℓ_2-regularization.

Logistic Regression: We use a binary version of MNIST [30] where the first five classes are treated as class 0 and the rest as class 1. We train a classifier with the binary cross-entropy loss. We consider a decentralized setting where the data is evenly distributed among all the nodes in a challenging sorted setting (sorted based on labels) where at most one node acquires examples from both classes.

Using the above tasks, next we study the effect of the following considerations.

6.1.2 Setup: Fixed Communication Budget Per Iteration

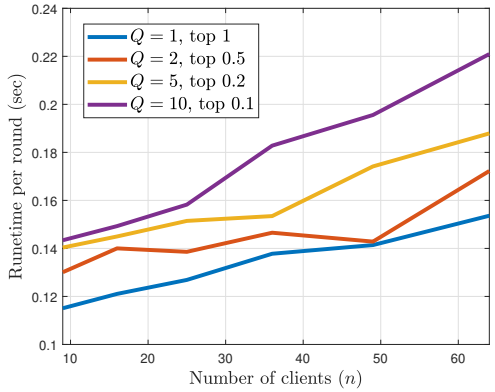
In order to illustrate the value of having more gossiping steps (i.e. larger Q), we consider a simple setting where our communication budget in every iteration (involving one gradient computation step and Q gossiping steps) is fixed. So for top-k/rand-k, we keep Qk constant, whereas for b-bit qsgd, Qb is kept constant. Since Qk/Qb is kept constant, the total number of bits communicated will be proportional to the number of iterations T (which is the horizontal axis of the plots in Figures 5 and 6). In Figure 5, we consider a DFL setting with $n = 16$ clients forming a torus topology, and plot the training loss on the vertical axis (in log-scale) vs. the number of bits (order wise) on the horizontal axis for SYN-2 (non-convex non-linear regression) with qsgd. We maintain $Qb = 8$ and consider 3 different consensus learning rates $\gamma = \{0.05, 0.1, 0.15\}$ (keeping everything else the same).

In Figure 6, we show similar plots for SYN-1 (strongly convex linear regression task) with top-k. Let $\omega = (k/d)*100$ (d being the dimension of the vectors). We keep $Q\omega = 100$ (note that this is the same as maintaining Qk constant) and consider 3 different values of the number of nodes $n = \{9, 16, 25\}$ (keeping everything else the same).

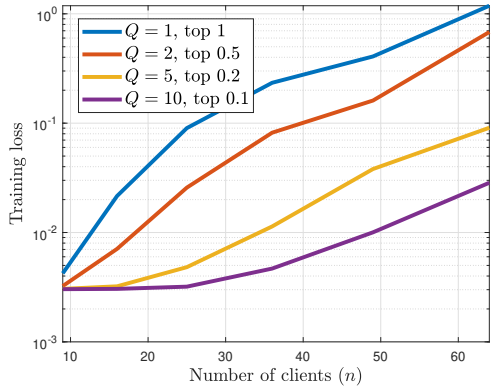
In Figure 8, we show results for the logistic regression task on MNIST with rand(ω). Let $\omega = (k/d)*100$ (d being the dimension of the vectors). We keep $Q\omega = 100$ (note that this is the same as maintaining Qk constant) and consider the two most commonly used topologies, ring and torus with $n = 9$. Everything else is kept the same. **Significance of large Q.** In both Figure 5 and Figure 6, observe that higher Q at the expense of more aggressive compression leads to fewer gross total number of bits communicated – as predicted by the results established in the beginning of Section 5.1. Consistent with the results of Figure 5 and Figure 6, observe that in Figure 8, using a higher Q at the expense of more aggressive compression leads to fewer gross total number of bits communicated – for both ring and torus topologies.

Note that if latency/synchronization time between the nodes is negligible, then having higher Q also leads to faster convergence (since the total number of bits is proportional to T in our setting). Further, in Figure 6 (for SYN-1, which has a strongly convex objective), observe that higher Q results in almost straight line curves (recall that the training loss is plotted in log-scale) – implying linear convergence. Further note that for $Q = 1$ the curves are not straight lines. This verifies in turn Corollary 1.1.

Extreme compression effect. As we discussed in Section 5.1, as long as the compression rate is not too severe, our theoretical results establishes that increasing Q is beneficial. Too verify this result, we consider the SYN-1 dataset for the linear regression task and show the results of training and test performance in Fig. 10. In this scenario we used the torus topology with $n = 9$ and kept the learning rates η and γ fixed for all curves. As the figure demonstrates, increasing Q helps improving the performance, but up to a point (i.e. a compression rate higher than 2%). When the compression rate is too small, i.e. 1%, the algorithm diverges. This observation is consistent with our preceding argument that a higher Q helps as long as the compression rate is not too small. Note that the smallest compression rate may depend on how difficult learning the parameters of the model are for a given task.



(a) Runtime vs network size



(b) Training loss v.s. network size given 30 sec runtime budget

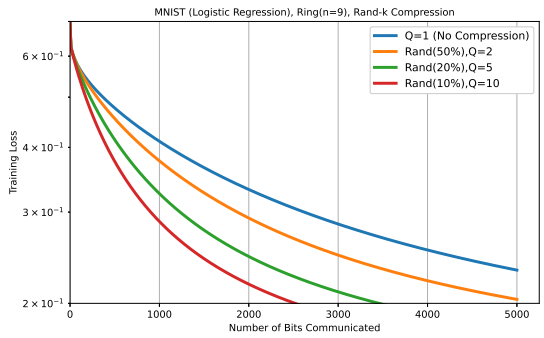
Fig. 7: (a) Runtime v.s. network size for the linear regression task on SYN-1 dataset with the torus topology. All schemes experience a slower convergence as the size of the network increases. (b) Training loss vs network size given 30 sec runtime budget for the linear regression task on SYN-1 dataset with the torus topology. The figure shows the benefit of the proposed approach.

Slightly convex curves. Note that the curves in Figure 5, Figure 6, and Figure 8 are slightly convex. This phenomenon stems from the fact that the tested Q might be smaller than Q_0 specified in Theorem 1. To further investigate this we ran a test on SYN-1 in Figure 9. As we see, given a fixed ω, with increasing Q the convergence curves approach that of the centralized algorithm (which attains a linear rate).

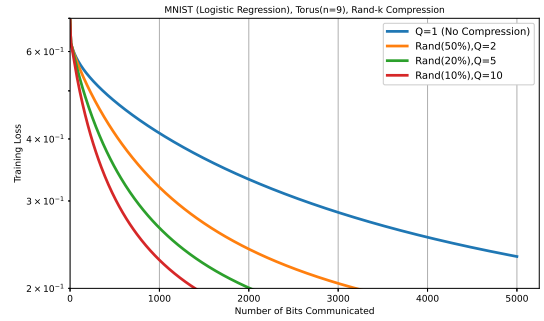
Runtime comparison. Finally, we aim to determine the effect of the network size and a fixed runtime budget on the accuracy. Fig. 7 summarizes the result of this study for the linear regression task on SYN-1 dataset with the torus topology. Fig. 7(a) shows all schemes experience a slower convergence as the size of the network increases. Fig. 7(b) shows given a fixed runtime budget of 30 sec the proposed approach finds higher quality models.

6.2 Deep Learning Experiments

Having verified our theoretical contribution via linear, nonlinear, and logistic regression tasks in the previous section, we now resort to large-scale experiments with deep learning models to identify parameters of a predictive model in



(a) Ring



(b) Torus

Fig. 8: Effect of different (Q, ω) pairs (where ω denotes the percentage of random co-ordinates picked in the rand quantization) such that $Q\omega = 100$, on the total number of bits communicated for MNIST logistic regression task. We consider the ring and torus topology with $n = 9$. In both plots, $\gamma = 0.05$, ℓ_2 regularization value = 0.001, and $\eta = 0.2$.

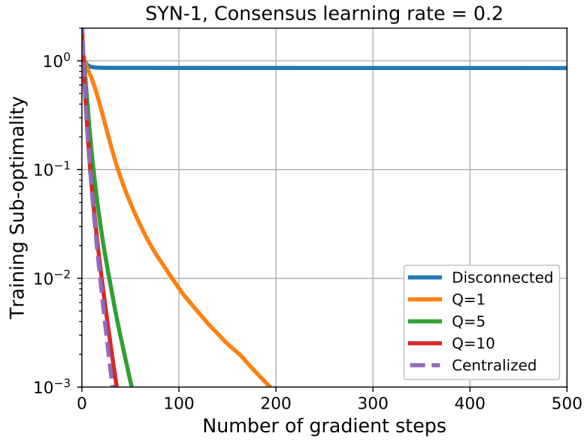
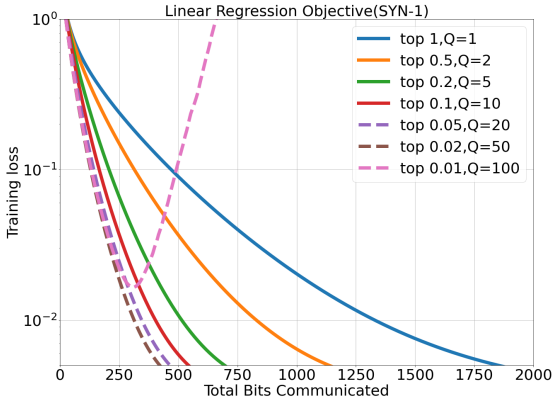


Fig. 9: As Q increases the convergence curves approach to a linear curve.

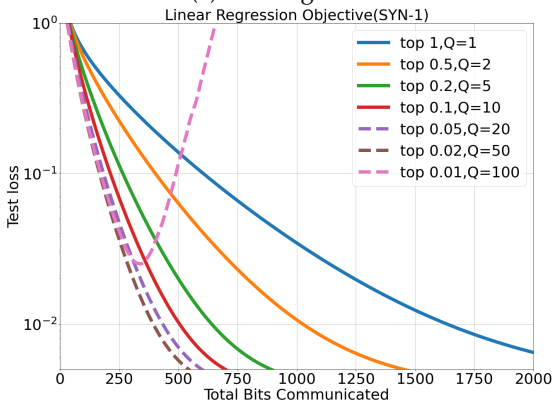
a federated learning scenario, thereby demonstrating the efficacy of the proposed communication strategy.

6.2.1 *Classification on CIFAR-10*

To show the benefit of the multi-step gossiping in large-scale non-convex optimization tasks, we consider the task of distributed classification of the CIFAR-10 dataset. We assume a DFL scenario where $n = 10$ clients form a time-varying undirected communication graph (see Figure 11 for



(a) Training loss



(b) Test loss

Fig. 10: Training and test errors for linear regression task on SYN-1 data. Increasing Q helps improving the performance as long as the compression rate is not too small.

a simple illustration). Specifically, at each time step a random subset of $r = 0.5n$ (= 5 for this case) nodes form a fully connected component and perform compressed gossiping among themselves. Note that the overall graph at each time step is disconnected. However, the union of these graphs over all time steps is going to be connected with high probability and therefore decentralized learning methods are expected to converge [31], [32]. The described setting models a federated learning task with periodic communications [1] (see [33]), where in each communication round only r out of n clients share their local models with the server.

The model in each client is a two-layer neural network with ReLU activation with 500 neurons in each hidden layer. Furthermore, the clients employ stochastic gradients with local batch size of 256 (as opposed to full gradients considered in our theoretical results). We consider a heterogeneous setting where each client can have data from at most five (out of 10) classes.

We now describe the procedure that we have used to generate heterogeneous data. The entire training data is first sorted based on labels and then divided into 50 equal data-shards in the sorted order, i.e. for CIFAR-10 each data shard is assigned 1000 samples. Further, this way of splitting ensures that each shard can have data belonging to just one class. Each client is then assigned 5 shards chosen uniformly at random without replacement to cover the whole dataset. Thus, each client can have data from at most five classes.

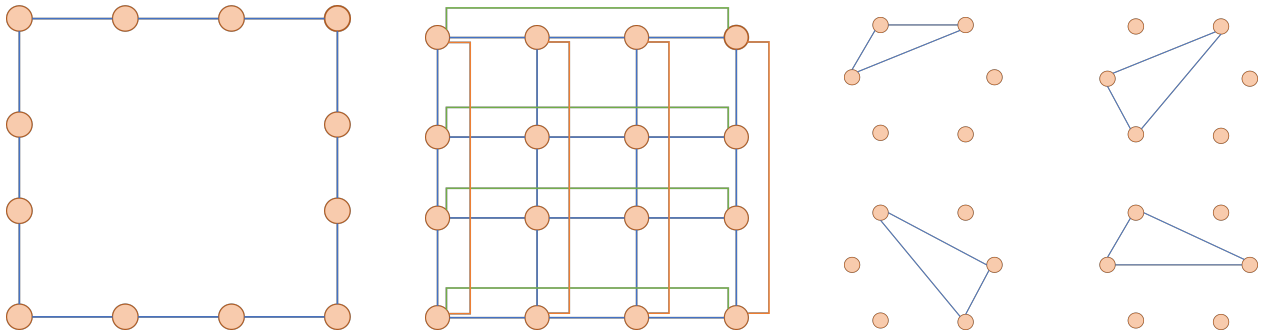


Fig. 11: **Left:** DFL on a ring topology with $n = 16$ clients used in SYN-1 and SYN-2 experiments. The nonzero weights in the mixing matrix are equal to $1/(\text{degree} + 1) = 1/3$. The plus one is to account for a "self-loop" as each node always communicates with itself. **Middle:** DFL on a Torus graph with $n = 16$ used in SYN-1 and SYN-2 experiments. The nonzero weights in the mixing matrix are equal to $1/(\text{degree} + 1) = 1/5$. Note that we use different colors for edges only for clarity. **Right:** An example of DFL with a time-varying undirected communication graph with uniformly at random client selection used in distributed classification with neural networks on CIFAR-10 and Fashion MNIST. While $n = 10$ in the experiments, for the ease of demonstration we only show six clients in the figure.

We train the models using the categorical cross-entropy loss with ℓ_2-regularization. The weight decay value in PyTorch for applying ℓ_2-regularization is set to be 1e-4. The experiments are run on one NVIDIA TITAN Xp GPU.

We follow the same experimental setting as that in the previous section, i.e., keeping the communication budget in every iteration fixed to $Qb = 16$, where b denotes the number of bits in qsgd (i.e., we use the QSGD operator with b bits to compress the communication in the network). The initial learning rate for all pairs of (Q, b) is set to 10^{-2}.

The results are shown in Figure 12a. As the figure shows, similar to the results on regression tasks, the proposed approach benefits from increasing Q while reducing b, and given a fixed communication budget the achievable error reduces by adopting the proposed multiple-step approach. We note that due to the fact that the data distribution is heterogeneous, a phenomenon known as client drift [34] slows the convergence of the average model. This is captured in Theorem 1 by the quantity Δ^2 which is a notion quantifying to what extent the clients' data distribution is different. As Theorem 1 shows, by increasing Q, we can reduce the effect of a high Δ^2, thereby reducing the effect of client drift.

Note that as predicted by Theorem 1, the convergence curves are relatively linear, even though we use stochastic gradients for each client.

6.2.2 Classification on Fashion-MNIST

We consider the same task as the one in Section 6.2.1, now using the Fashion-MNIST dataset [35] instead. We use two-layer neural network in each node with ReLU activation and 300 neurons in each hidden layer. Here, we use $n = 20$ clients each having data from at most two classes – note that this is a high degree of heterogeneity. We use a similar procedure as described in Section 6.2.1 for generating heterogeneous data here. The only change we make here is that the number of shards is set to 40 due to which each client gets the data of two shards (each of which has data from just one class as before). The graph topology is the same as that described in Section 6.2.1. We also use the same weight-decay value and initial learning rate as Section 6.2.1.

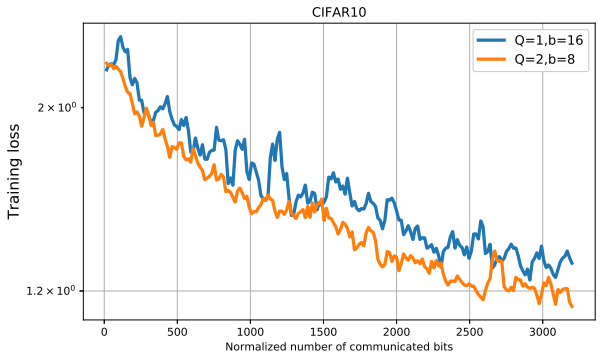
Keeping the communication budget in every iteration fixed ($Qb = 16$) by using qsgd with b bits, we show the training loss curves corresponding to different pairs of (Q, b) in Figure 12b. The result shows yet again the benefit of the proposed multiple-step approach in reducing the the client drift phenomenon in FL and thereby lowering the the training error under a fixed communication budget per (stochastic) gradient computation.

7 Conclusion

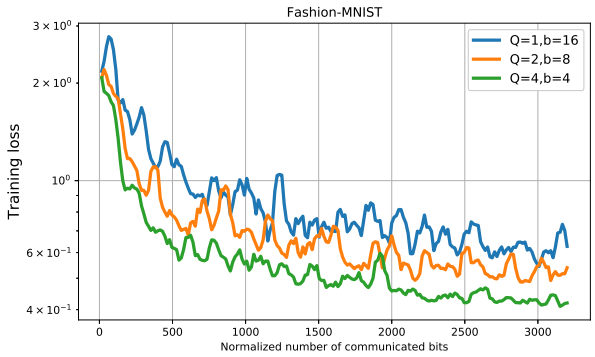
In this work, we considered the problem of communication-constrained decentralized federated learning to learn parameters of a deep predictive model in a collaborative fashion. We proposed a communication strategy that under a fixed communication budget aims to minimize the (training) loss as much as possible. The key insight behind the proposed strategy is using multiple gossip steps – given a fixed communication budget per iteration, having multiple gossip steps with lower precision communication is preferable to having just one gossip step with higher precision communication, in terms of the total number of bits communicated.

In particular, we showed that having $\mathcal{O}(\log \frac{1}{\epsilon})$ gradient iterations with constant step size - and $\mathcal{O}(\log \frac{1}{\epsilon})$ gossip steps between every pair of these iterations - enables convergence to within ϵ of the optimal value for smooth non-convex objectives satisfying Polyak-Łojasiewicz condition that arise in the training of deep learning models. Our extensive empirical study on a range of machine learning tasks such as regression, and collaborative classification via deep learning models across different network topologies and compression operators validates our theoretical contribution and shows the efficacy of the proposed scheme.

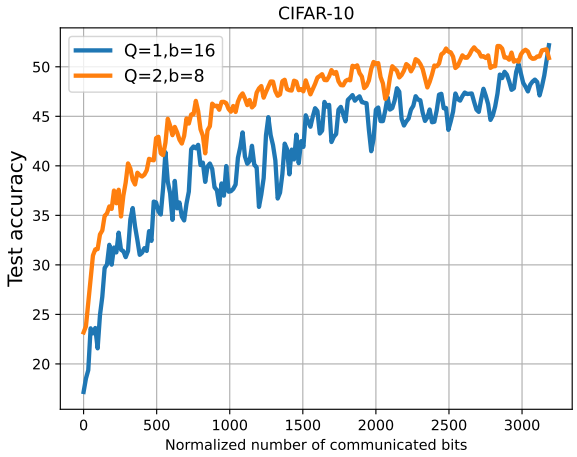
As part of the future work, it would be of interest to consider other practical extensions and considerations including communication strategies for directed and time-varying networks, dealing with communication-dropout and noisy communication channels, and use of stochastic local gradients and momentum. Another extension would be incorporating momentum to have an accelerated version of the proposed method.



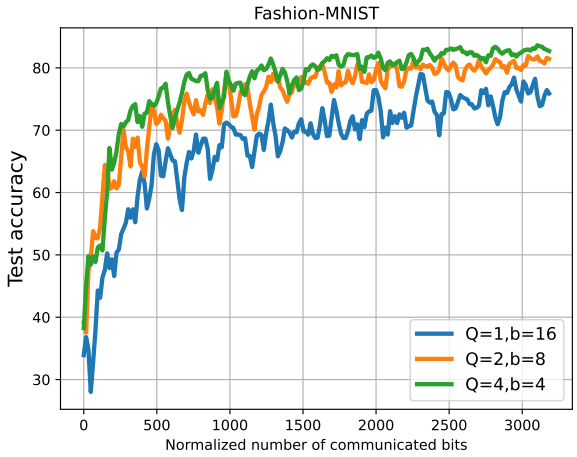
(a) Classification on CIFAR-10: Training loss



(b) Classification on Fashion-MNIST: Training loss



(c) Classification on CIFAR-10: Test accuracy



(d) Classification on Fashion-MNIST: Test accuracy

Fig. 12: Training loss and Test accuracy for distributed classification with neural networks via the considered DFL setting with a time-varying graph on CIFAR-10 (left) and Fashion MNIST (right). The x-axis is the normalized number of communicated bits where the normalization factor is $0.5nd$ (d being the dimension of the parameters). Note that for CIFAR-10, $(Q = 4, b = 4)$ does not do better than $(Q = 1, b = 16)$ due to which we have not shown it in the plot. We suspect that with just 4 bits, the learned model parameters are not "precise" enough to classify CIFAR-10 very well – even with multiple gossip steps. Note that for the CIFAR-10 experiment $Q = 4$, $b = 4$ is not shown since given the large number of parameters in this experiments, $b = 4$ will be a sever quantization level that results in slow convergence.

REFERENCES

[1] 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*, pp. 1273–1282, 2017.

[2] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[3] K. Scaman, F. Bach, S. Bubeck, Y. T. Lee, and L. Massoulié, "Optimal algorithms for smooth and strongly convex distributed optimization in networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 3027–3036, JMLR. org, 2017.

[4] 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.

[5] W. Liu, L. Chen, Y. Chen, and W. Zhang, "Accelerating federated learning via momentum gradient descent," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 8, pp. 1754–1766, 2020.

[6] Y. Zhou, Q. Ye, and J. C. Lv, "Communication-efficient federated learning with compensated overlap-fedavg," *IEEE Transactions on Parallel and Distributed Systems*, 2021.

[7] W. Wu, L. He, W. Lin, and R. Mao, "Accelerating federated learning over reliability-agnostic clients in mobile edge computing systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 7, pp. 1539–1551, 2020.

[8] J. N. Tsitsiklis, "Problems in decentralized decision making and computation.," tech. rep., Massachusetts Inst of Tech Cambridge Lab for Information and Decision Systems, 1984.

[9] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-based computation of aggregate information," in *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, pp. 482–491, IEEE, 2003.

[10] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.

[11] A. Koloskova, S. Stich, and M. Jaggi, "Decentralized stochastic optimization and gossip algorithms with compressed communication," in *International Conference on Machine Learning*, pp. 3478–3487, 2019.

[12] T. Li, M. Fu, L. Xie, and J.-F. Zhang, "Distributed consensus with limited communication data rate," *IEEE Transactions on Automatic Control*, vol. 56, no. 2, pp. 279–292, 2010.

[13] Y. Chen, A. Hashem, and H. Vikalo, "Communication-efficient algorithms for distributed optimization over directed graphs," *arXiv preprint arXiv*, 2020.

[14] F. Seide, H. Fu, J. Droppo, G. Li, and D. Yu, "1-bit stochastic gradient descent and its application to data-parallel distributed training of speech DNNs," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[15] N. Strom, "Scalable distributed DNN training using commodity

GPU cloud computing," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[16] S. U. Stich, J.-B. Cordonnier, and M. Jaggi, "Sparsified SGD with memory," in *Advances in Neural Information Processing Systems*, pp. 4447–4458, 2018.

[17] A. Koloskova, T. Lin, S. U. Stich, and M. Jaggi, "Decentralized deep learning with arbitrary communication compression," *arXiv preprint arXiv:1907.09356*, 2019.

[18] W. Shi, Q. Ling, G. Wu, and W. Yin, "EXTRA: An exact first-order algorithm for decentralized consensus optimization," *SIAM Journal on Optimization*, vol. 25, no. 2, pp. 944–966, 2015.

[19] B. T. Polyak, "Gradient methods for minimizing functionals," *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, vol. 3, no. 4, pp. 643–653, 1963.

[20] C. Liu, L. Zhu, and M. Belkin, "Toward a theory of optimization for over-parameterized systems of non-linear equations: the lessons of deep learning," *arXiv preprint arXiv:2003.00307*, 2020.

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

[22] S. P. Karimireddy, Q. Rebjock, S. U. Stich, and M. Jaggi, "Error feedback fixes SignSGD and other gradient compression schemes," *arXiv preprint arXiv:1901.09847*, 2019.

[23] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "QSGD: Communication-efficient SGD via gradient quantization and encoding," in *Advances in Neural Information Processing Systems*, pp. 1709–1720, 2017.

[24] M. Liu, W. Zhang, Y. Mroueh, X. Cui, J. Ross, T. Yang, and P. Das, "A decentralized parallel algorithm for training generative adversarial nets," *arXiv preprint arXiv:1910.12999*, 2019.

[25] A. Rogozin and A. Gasnikov, "Projected gradient method for decentralized optimization over time-varying networks," *arXiv preprint arXiv:1911.08527*, Feb. 2020.

[26] S. Ma, R. Bassily, and M. Belkin, "The power of interpolation: Understanding the effectiveness of SGD in modern over-parametrized learning," *arXiv preprint arXiv:1712.06559*, 2017.

[27] M. Schmidt and N. L. Roux, "Fast convergence of stochastic gradient descent under a strong growth condition," *arXiv preprint arXiv:1308.6370*, 2013.

[28] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu, "Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent," in *Advances in Neural Information Processing Systems*, pp. 5330–5340, 2017.

[29] H. Karimi, J. Nutini, and M. Schmidt, "Linear convergence of gradient and proximal-gradient methods under the Polyak-Łojasiewicz condition," in *European Conference on Machine Learning and Knowledge Discovery in Databases-Volume 9851*, pp. 795–811, 2016.

[30] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[31] A. Nedic, A. Olshevsky, and W. Shi, "Achieving geometric convergence for distributed optimization over time-varying graphs," *SIAM Journal on Optimization*, vol. 27, no. 4, pp. 2597–2633, 2017.

[32] 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.

[33] J. Wang, A. K. Sahu, Z. Yang, G. Joshi, and S. Kar, "Matcha: Speeding up decentralized SGD via matching decomposition sampling," *arXiv preprint arXiv:1905.09435*, 2019.

[34] 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," *arXiv preprint arXiv:1912.04977*, 2019.

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