TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 15/18 pp174–185
DOI: 10.26599/TST.2021.9010045
Volume 27, Number 1, February 2022

SIGNGD with Error Feedback Meets Lazily Aggregated Technique: Communication-Efficient Algorithms for Distributed Learning

Xiaoge Deng, Tao Sun, Feng Liu, and Dongsheng Li*

Abstract: The proliferation of massive datasets has led to significant interests in distributed algorithms for solving large-scale machine learning problems. However, the communication overhead is a major bottleneck that hampers the scalability of distributed machine learning systems. In this paper, we design two communication-efficient algorithms for distributed learning tasks. The first one is named EF-SIGNGD, in which we use the 1-bit (sign-based) gradient quantization method to save the communication bits. Moreover, the error feedback technique, i.e., incorporating the error made by the compression operator into the next step, is employed for the convergence guarantee. The second algorithm is called LE-SIGNGD, in which we introduce a well-designed lazy gradient aggregation rule to EF-SIGNGD that can detect the gradients with small changes and reuse the outdated information. LE-SIGNGD saves communication costs both in transmitted bits and communication rounds. Furthermore, we show that LE-SIGNGD is convergent under some mild assumptions. The effectiveness of the two proposed algorithms is demonstrated through experiments on both real and synthetic data.

Key words: distributed learning; communication-efficient algorithm; convergence analysis

1 Introduction

The past few decades have witnessed an explosion of data in both the number of observations and parameters, resulting in significant interests in distributed algorithms for solving large-scale machine learning problems[1–7]. However, efficient implementations of the distributed optimization algorithms for machine learning applications are challenging. Both intensive computational workloads and the volume of data communication demand careful system designs. This paper is devoted to the algorithmic development of communication-efficient algorithms for distributed learning tasks, which can be formulated as

$$\min_{\boldsymbol{\omega}\in\mathbb{R}^d} f(\boldsymbol{\omega}) \quad \text{with} \quad f(\boldsymbol{\omega}) := \sum_{m\in\mathcal{M}} f_m(\boldsymbol{\omega}) \tag{1}$$

where $\boldsymbol{\omega} \in \mathbb{R}^d$ is the parameter vector to be learned. f and $\{f_m, m \in \mathcal{M}\}$ are smooth (may be nonconvex) functions, where $\mathcal{M} := \{1, \ldots, M\}$ denotes the set of workers. Problem (1) arises in a large number of distributed machine learning tasks, ranging from linear models to deep neural networks[2, 8, 9]. In distributed settings, f_m is also a sum of functions, i.e., $f_m(\boldsymbol{\omega}) := \sum_{n\in\mathcal{N}_m} \ell(\boldsymbol{x}_{m,n}; \boldsymbol{\omega})$, where $\ell(\boldsymbol{x}; \boldsymbol{\omega})$ is the loss function associated with parameter $\boldsymbol{\omega}$ and training sample $\boldsymbol{x}$, and $\mathcal{N}_m$ is the number of data samples at worker m.

Gradient Descent (GD) is the main workhorse for Problem (1), which is performed as

$$\boldsymbol{\omega}^{k+1} = \boldsymbol{\omega}^k - \gamma \cdot \sum_{m=1}^{M} \nabla f_m(\boldsymbol{\omega}^k) \tag{2}$$

where $\boldsymbol{\omega}^k$ is the parameter value at iteration k and γ denotes the step size. In the commonly used parameter server architecture, the implementation details of the GD method are as follows: At iteration k, the server

• Xiaoge Deng, Tao Sun, Feng Liu, and Dongsheng Li are with National Laboratory for Parallel and Distributed Processing (PDL), College of Computer, National University of Defense Technology, Changsha 410073, China. E-mail: nudtxgdeng@163.com; nudtsuntao@163.com; richardlf@nudt.edu.cn; dsli@nudt.edu.cn.
* To whom correspondence should be addressed.
Manuscript received: 2020-08-23; accepted: 2020-09-24

broadcasts $\boldsymbol{\omega}^k$ to all workers; every worker $m \in \mathcal{M}$ computes the local gradient $\nabla f_m(\boldsymbol{\omega}^k)$ and uploads it to the server; the server aggregates all the gradients, i.e., $\sum_{m\in\mathcal{M}} \nabla f_m(\boldsymbol{\omega}^k)$, and updates the parameters via Eq. (2). The server needs to communicate with all workers to obtain fresh gradients $\{\nabla f_m(\boldsymbol{\omega}^k)\}_{m=1}^M$ in each iteration. However, communication is much slower than computation in several settings[3]. Therefore, as the number of workers grows, or when incorporating popular deep learning-based models with high-dimensional parameters, worker server communications become a major bottleneck[10].

1.1 Related work

Communication-efficient distributed learning methods have gained popularity recently[10–14]. We briefly review two kinds of related work.

Communication bits. Gradient quantization is a simple but efficient method to reduce communication bits. It aims to compress gradients by limiting the number of bits that represent floating point numbers during communications. Multi-bit quantization schemes have been studied in Refs. [15, 16], where an adjustable quantization level can offer additional flexibility to control the trade-off between the per iteration communication cost and the convergence rate. One-bit quantization method (e.g., SIGNSGD) has been developed in Refs. [17–19], which reduces each component of the gradient to only its sign (one bit). References [18, 19] provided theoretical and empirical evidence that 1-bit signed gradient schemes converge well under some assumptions. However, Refs. [12, 20] show that naive use of this sign-based gradient compression scheme may lead to the divergence of SIGNSGD. To this end, they proposed SIGNSGD with Error Feedback (EF-SIGNSGD), which can fix possible divergence[20].

Communication rounds. Many communication-efficient schemes have been recently developed to reduce the number of communication rounds. Instead of the gradient information, higher-order information (Newton-type method) were leveraged to reduce the number of communication rounds[21–23]. Novel aggregation techniques, such as periodic aggregation[24, 25] and adaptive aggregation[16, 26–29], are used to skip some communications. Among these, the Lazily Aggregated Quantized gradients (LAQ) approach which was proposed in Ref. [16] is the first to quantize the computed gradients, and then skip less informative quantized gradient communications, which means that it saves both the communication bits and rounds. Reference [30] also reported a lower bound on communication rounds.

1.2 Our contribution

This paper focuses on developing the GD method Eq. (2) to reduce communication costs with a theoretical convergence of guarantees. Our contributions are as follows:

(1) To save the communication bits, we used the 1-bit gradient quantization method. The error feedback technique was employed for the convergence guarantee. We named this method as EF-SIGNGD and designed it for the distributed systems. In EF-SIGNSGD[20], only a single worker is considered. In this paper, we studied the more interesting distributed setting. Without relying on the unrealistic assumptions that have a large mini-batch and unimodal symmetric gradient noise in Refs. [18, 19], we show that EF-SIGNGD is convergent under mild assumptions.

(2) To reduce the communication rounds, we introduced a lazily aggregated rule and named this algorithm as LE-SIGNGD, which can save communication bits and rounds simultaneously without sacrificing the desired convergence properties. In particular, jointly adopting multiple techniques makes our theoretical analysis highly challenging.

(3) We tested these two algorithms both on real and synthetic data, and the numerical experiment results verified the effectiveness of the proposed algorithms.

Notation: Bold lowercase letters denote column vectors. For a vector $\boldsymbol{x} \in \mathbb{R}^d$, $\|\boldsymbol{x}\|$ and $\|\boldsymbol{x}\|_1$ denote the ℓ_2-norm and ℓ_1-norm of $\boldsymbol{x}$, respectively, $\boldsymbol{x}_i$ is the i-th component, and $\text{sign}(\boldsymbol{x})$ is the vector whose i-th element is $\text{sign}(\boldsymbol{x}_i)$.

2 SIGNGD Algorithm with Error Feedback

In this section, we intend to apply the 1-bit gradient compression approach for saving the communication bits of the basic GD method Eq. (2) with the error feedback technique.

2.1 Algorithm development

In the 1-bit gradient compression methodology, the m-th worker only uploads the sign of the gradient computed on its portion of the data, which suggests an update of the following form:

$$\boldsymbol{\omega}^{k+1} = \boldsymbol{\omega}^k - \gamma \cdot \sum_{m=1}^{M} \text{sign}(\nabla f_m(\boldsymbol{\omega}^k)) \qquad (3)$$

However, Ref. [20] presented some counterexamples

to substantiate that naively using such a sign-based algorithm may not generalize or even converge. The sign operator misses massive information about the local gradient's magnitude and direction.

Thus, we employ an elegant error feedback technique to fix the abovementioned problems. The error feedback technique is performed as follows: scaling the signed vector by the ℓ_1-norm of the gradient to ensure the magnitude of the gradient is not forgotten; locally storing the difference between the actual and compressed gradient, and adding it back into the next step so that the correct direction is not forgotten. We named this sign-based GD method with Error Feedback as EF-SIGNGD (Algorithm 1) and applied it to the distributed system.

More specifically, in Algorithm 1, $\boldsymbol{e}_m^k$ represents the accumulated error from all compression steps in the previous k iterations of worker m. This residual error is added to the gradient step $\nabla f_m(\boldsymbol{\omega}^k)$ to obtain the corrected direction $\boldsymbol{g}_m^k$, i.e.,

$$\boldsymbol{g}_m^k = \nabla f_m(\boldsymbol{\omega}^k) + \boldsymbol{e}_m^k \tag{4}$$

All workers will upload $\|\boldsymbol{g}_m^k\|_1$ and $\operatorname{sign}(\boldsymbol{g}_m^k)$ to the server, and the compression gradient is defined as the signed vector $\operatorname{sign}(\boldsymbol{g}_m^k)$ scaled by $\|\boldsymbol{g}_m^k\|_1/d$, i.e.,

$$\mathcal{C}(\boldsymbol{g}_m^k) := \frac{\|\boldsymbol{g}_m^k\|_1}{d}\operatorname{sign}(\boldsymbol{g}_m^k) \tag{5}$$

and stores information about the magnitude. d represents the parameter dimension. Therefore, the EF-SIGNGD algorithm is updated by

$$\boldsymbol{\omega}^{k+1} = \boldsymbol{\omega}^k - \gamma \cdot \sum_{m\in\mathcal{M}} \mathcal{C}(\boldsymbol{g}_m^k) \tag{6}$$

Our focus here is to reduce the number of worker-to-server uplink communications, which also refer to uploads (the same as Ref. [16]). Table 1 shows the communication bits per upload of various algorithms. Compared with GD and SIGNGD, we find a trade-off between the number of communication bits and convergence guarantee. In large-scale machine learning tasks, the dimension d of the parameters is usually very large, so the cost of the extra $32M$ bits is negligible.

Algorithm 1 EF-SIGNGD

Input: step size $\gamma > 0$, worker number M.
Initialize: $\boldsymbol{\omega}^0 = \mathbf{0}$; $\boldsymbol{e}_m^0 = \mathbf{0}$, $\forall m \in \mathcal{M}$.
for $k = 0, 1, \ldots, K$ **do**
  server broadcasts $\boldsymbol{\omega}^k$ to all workers.
  for worker $m = 1, \ldots, M$ **do**
    worker m computes $\boldsymbol{g}_m^k = \nabla f_m(\boldsymbol{\omega}^k) + \boldsymbol{e}_m^k$.
    $\mathcal{C}(\boldsymbol{g}_m^k) := (\|\boldsymbol{g}_m^k\|_1/d)\operatorname{sign}(\boldsymbol{g}_m^k)$.
    worker m updates $\boldsymbol{e}_m^{k+1} = \boldsymbol{g}_m^k - \mathcal{C}(\boldsymbol{g}_m^k)$.
    worker m uploads $\|\boldsymbol{g}_m^k\|_1$ and $\operatorname{sign}(\boldsymbol{g}_m^k)$.
  end for
  server update parameter $\boldsymbol{\omega}$ according to Eq. (6).
end for

Table 1 Communication bits of different algorithms when training a d-dimensional parameter with M workers.

Algorithm	Number of bits per upload
GD	$32Md$
SIGNGD	Md
EF-SIGNGD	$M(32+d)$

2.2 Theorem guarantee

This part shows the theoretical guarantee of the EF-SIGNGD method under several standard conditions. The detailed proof will be given in the Appendix.

Assumption 1 (smoothness) Function $f_m(\cdot)$ is L_m-smooth, and $f(\cdot)$ is L-smooth.

Assumption 2 (gradient boundedness) For a given $\boldsymbol{\omega} \in \mathbb{R}^d$, the local gradient $\nabla f_m(\cdot)$ and the global gradient $\nabla f(\cdot)$ are bounded, i.e., there exits constant $\sigma_m, \sigma \in \mathbb{R}$ such that

$$\|\nabla f_m(\boldsymbol{\omega})\| \leqslant \sigma_m, \|\nabla f(\boldsymbol{\omega})\| \leqslant \sigma \tag{7}$$

We use the definition of compressor given in Ref. [20].

Lemma 1 (in Ref. [20], Lemma 8) The operator $\mathcal{C}(\cdot)$ defined in Formula (5) is a δ-approximate compressor, i.e., there exits a constant $\delta \in (0, 1)$, such that

$$\|\mathcal{C}(\boldsymbol{x}) - \boldsymbol{x}\|^2 \leqslant (1-\delta)\|\boldsymbol{x}\|^2, \forall \boldsymbol{x} \in \mathbb{R}^d \tag{8}$$

With the gradient bound assumption, we are prepared to present a critical lemma, which is used to bound the residual error in Algorithm 1.

Lemma 2 Under Assumption 2, at any iteration k of EF-SIGNGD, the accumulated error in worker m, i.e., $\boldsymbol{e}_m^k$, is bounded:

$$\|\boldsymbol{e}_m^k\|^2 \leqslant \frac{4(1-\delta)}{\delta^2}\sigma_m^2 \tag{9}$$

Lemma 2 shows that the residual errors maintained in Algorithm 1 do not accumulate too much. Then we can estimate the object value descent by performing one-iteration of the EF-SIGNGD method, and the convergence theorem of our algorithm will also be given.

Lemma 3 (EF-SIGNGD descent) Under Assumption 1, $\boldsymbol{\omega}^{k+1}$ is generated by running the one-step EF-SIGNGD iteration Eq. (6) given $\boldsymbol{\omega}^k$ and step size $0 < \gamma < 2/(\rho + L)$. The objective values satisfy

$$f(\boldsymbol{\nu}^{k+1}) - f(\boldsymbol{\nu}^k) \leqslant \Delta_{\mathrm{EF}}^k \tag{10}$$

where $\rho > 0$ and

$$\boldsymbol{\nu}^k := \boldsymbol{\omega}^k - \gamma \cdot \sum_{m=1}^{M} \boldsymbol{e}_m^k \tag{11}$$

$$\Delta_{\mathrm{EF}}^k := -\gamma\left[1-\frac{(\rho+L)\gamma}{2}\right]\|\nabla f(\boldsymbol{\omega}^k)\|^2+\frac{L^2\gamma^2}{2\rho}\Big\|\sum_{m=1}^M \boldsymbol{e}_m^k\Big\|^2 \tag{12}$$

Theorem 1 Under Assumptions 1 and 2, let the sequence $\{\boldsymbol{\omega}^k\}_{k\geqslant 0}$ be generated by EF-SIGNGD with step size $0<\gamma<2/(\rho+L)$, then

$$\min_{0\leqslant k\leqslant K}\|\nabla f(\boldsymbol{\omega}^k)\|^2 \leqslant \frac{a(f^0-f^*)}{\gamma\cdot(K+1)}+b\cdot\gamma \tag{13}$$

where $f^0:=f(\boldsymbol{\omega}^0)$, f^* is the optimal value of Problem (1), $a:=\dfrac{2}{2-(\rho+L)\gamma}$, $b:=\dfrac{4L^2M\sigma^2(1-\delta)}{\rho\delta^2[2-(\rho+L)\gamma]}$, and $\sigma^2:=\sum_{m=1}^M\sigma_m^2$.

Remark 1 Lemma 1 shows that $\mathcal{C}(\cdot)$ is a δ-approximate compressor, which implies that a δ-fraction of the gradient information is sent at each iteration. The rest is added to the residual error to be transmitted later. Therefore, the sequence $\{\boldsymbol{v}^k\}_{k\geqslant 0}$ defined in Formula (11) is the result of error correction for $\{\boldsymbol{\omega}^k\}_{k\geqslant 0}$, and it has the property:

$$\boldsymbol{v}^{k+1}=\boldsymbol{\omega}^k-\gamma\sum_{m=1}^M \boldsymbol{g}_m^k=\boldsymbol{v}^k-\gamma\nabla f(\boldsymbol{\omega}^k) \tag{14}$$

3 Lazily Aggregated EF-SIGNGD Method

The EF-SIGNGD method reduces the number of communication bits per upload. At the same time, we can further reduce the number of uploads while ensuring the convergence of the algorithm. In this section, we employ the lazily aggregated technique to develop a smart lazy aggregation rule to skip certain communications. This approach can save communication bits and rounds simultaneously without sacrificing the desired convergence properties. Throughout the paper, one round of communication means one worker upload.

3.1 Lazily aggregated algorithm

The basic idea of the lazily aggregated technique is that if the difference of two consecutive locally compressed gradients is small, then the redundant uploads may be skipped and the previous one in the server can be reused. This idea comes from a simple rewriting of the EF-SIGNGD iteration Eq. (6) as

$$\boldsymbol{\omega}^{k+1}=\boldsymbol{\omega}^k-\gamma\sum_{m\in\mathcal{M}}\mathcal{C}(\boldsymbol{g}_m^{k-1})-\gamma\sum_{m\in\mathcal{M}}(\mathcal{C}(\boldsymbol{g}_m^k)-\mathcal{C}(\boldsymbol{g}_m^{k-1})) \tag{15}$$

The difference in two consecutive compressed gradients on worker m, i.e., $\mathcal{C}(\boldsymbol{g}_m^k)-\mathcal{C}(\boldsymbol{g}_m^{k-1})$, can be viewed as a refinement to $\mathcal{C}(\boldsymbol{g}_m^{k-1})$. Obtaining this refinement requires a round of communication between the server and the worker m. If this refinement is small enough, i.e.,

$$\|\mathcal{C}(\boldsymbol{g}_m^k)-\mathcal{C}(\boldsymbol{g}_m^{k-1})\|\ll\Big\|\sum_{m\in\mathcal{M}}\mathcal{C}(\boldsymbol{g}_m^{k-1})\Big\|,$$

then we can skip the communication between the server and the worker m to reduce the communication rounds.

Generalizing on this intuition, the lazily aggregated EF-SIGNGD algorithm, named LE-SIGNGD, will be updated by

$$\begin{aligned}&\boldsymbol{\omega}^{k+1}=\boldsymbol{\omega}^k-\gamma\sum_{m\in\mathcal{M}}\mathcal{C}(\tilde{\boldsymbol{g}}_m^{k-1})-\gamma\sum_{m\in\mathcal{M}^k}(\mathcal{C}(\boldsymbol{g}_m^k)-\mathcal{C}(\tilde{\boldsymbol{g}}_m^{k-1}))=\\&\boldsymbol{\omega}^k-\gamma\Big(\sum_{m\in\mathcal{M}^k}\mathcal{C}(\boldsymbol{g}_m^k)+\sum_{m\in\mathcal{M}_c^k}\mathcal{C}(\tilde{\boldsymbol{g}}_m^{k-1})\Big)=\\&\boldsymbol{\omega}^k-\gamma\sum_{m\in\mathcal{M}}\mathcal{C}(\boldsymbol{g}_m^k)+\gamma\sum_{m\in\mathcal{M}_c^k}\Big(\mathcal{C}(\boldsymbol{g}_m^k)-\mathcal{C}(\tilde{\boldsymbol{g}}_m^{k-1})\Big)\end{aligned} \tag{16}$$

where $\tilde{\boldsymbol{g}}_m^k=\nabla f_m(\tilde{\boldsymbol{\omega}}_m^k)+\boldsymbol{e}_m^k$, $\mathcal{M}^k$ and $\mathcal{M}_c^k$ are the sets of workers that do and do not communicate with the server in iteration k, respectively. We will only use the fresh compressed gradients from the selected workers in $\mathcal{M}^k$, and reuse the outdated compressed gradients from the rest of workers, which means

$$\begin{aligned}&\tilde{\boldsymbol{\omega}}_m^k:=\boldsymbol{\omega}^k,\forall m\in\mathcal{M}^k;\quad \tilde{\boldsymbol{\omega}}_m^k:=\tilde{\boldsymbol{\omega}}_m^{k-1},\forall m\in\mathcal{M}_c^k,\\&\tilde{\boldsymbol{g}}_m^k:=\boldsymbol{g}_m^k,\forall m\in\mathcal{M}^k;\quad \tilde{\boldsymbol{g}}_m^k:=\tilde{\boldsymbol{g}}_m^{k-1},\forall m\in\mathcal{M}_c^k\end{aligned} \tag{17}$$

Therefore, the iteration process of LE-SIGNGD can also be expressed as

$$\boldsymbol{\omega}^{k+1}=\boldsymbol{\omega}^k-\gamma\nabla^k,\quad \nabla^k:=\sum_{m\in\mathcal{M}}\mathcal{C}(\tilde{\boldsymbol{g}}_m^k) \tag{18}$$

The difference between two compressed gradients in worker m at the current iterate $\boldsymbol{\omega}^k$ and the old copy $\tilde{\boldsymbol{\omega}}^{k-1}$ is defined as

$$\Delta_m^k:=\mathcal{C}(\boldsymbol{g}_m^k)-\mathcal{C}(\tilde{\boldsymbol{g}}_m^{k-1}) \tag{19}$$

We find that

$$\nabla^k=\nabla^{k-1}+\sum_{m\in\mathcal{M}^k}\Delta_m^k \tag{20}$$

Combining Eqs. (18) and (20), we can observe that, instead of requesting all fresh compressed gradients in EF-SIGNGD, the lazily aggregated trick is to obtain ∇^k by refining the previously aggregated gradient ∇^{k-1}. If ∇^{k-1} is stored in the server, then we can scale down the per-iteration communication rounds from M to $|\mathcal{M}^k|$.

Designing a principled criterion to select a subset of workers $\mathcal{M}_c^k$ that does not communicate with the server at each iteration is critical. Our focus is on the trade-off

between the communication cost and the convergence guarantee of the LE-SIGNGD algorithm. Therefore, we compare the one-step descent amount of EF-SIGNGD and that of LE-SIGNGD. For EF-SIGNGD, as shown in Lemma 3, the one-step descent is Δ_{EF}^k. The one-step descent of LE-SIGNGD is Δ_{LE}^k, which will be specified in the following lemma.

Lemma 4 (LE-SIGNGD descent) Under Assumption 1, $\boldsymbol{\omega}^{k+1}$ is generated by running the one-step LE-SIGNGD iteration (Eq. (18)) given $\boldsymbol{\omega}^k$. The objective values satisfy

$$f(\boldsymbol{\nu}^{k+1}) - f(\boldsymbol{\nu}^k) \leqslant \Delta_{\mathrm{LE}}^k \tag{21}$$

$$\Delta_{\mathrm{LE}}^k := -\frac{\gamma}{2}\left\|\nabla f(\boldsymbol{\omega}^k)\right\|^2 + \frac{L^2\gamma^2}{2\rho}\Big\|\sum_{m=1}^M \boldsymbol{e}_m^k\Big\|^2 + \left(\frac{\rho+L}{2} - \frac{1}{2\gamma}\right)\|\boldsymbol{\nu}^{k+1} - \boldsymbol{\nu}^k\|^2 + \frac{\gamma}{2}\Big\|\sum_{m\in\mathcal{M}_c^k}\Delta_m^k\Big\|^2 \tag{22}$$

where $\rho > 0$.

Remark 2 Different from Eq. (14) in EF-SIGNGD, the sequence $\{\boldsymbol{\nu}^k\}_{k\geqslant 0}$ has the following property:

$$\boldsymbol{\nu}^{k+1} = \boldsymbol{\omega}^k - \gamma\sum_{m=1}^M \boldsymbol{g}_m^k + \gamma\sum_{m\in\mathcal{M}_c^k}\Delta_m^k = \boldsymbol{\nu}^k - \gamma\nabla f(\boldsymbol{\omega}^k) + \gamma\sum_{m\in\mathcal{M}_c^k}\Delta_m^k \tag{23}$$

Lemmas 3 and 4 estimate the objective value descent by performing one iteration of the EF-SIGNGD and LE-SIGNGD methods, respectively, conditioned on a common iterate $\boldsymbol{\omega}^k$. EF-SIGNGD finds Δ_{EF}^k by performing M rounds of communication with all the workers, while LE-SIGNGD yields Δ_{LE}^k by performing only $|\mathcal{M}^k|$ rounds of communication with a selected subset of workers. Our pursuit is to select a subset $\mathcal{M}^k$ to ensure that LE-SIGNGD enjoys larger per-communication descent than EF-SIGNGD; that is

$$\frac{\Delta_{\mathrm{LE}}^k}{|\mathcal{M}^k|} \leqslant \frac{\Delta_{\mathrm{EF}}^k}{M} \tag{24}$$

For simplicity, we choose the step size $\gamma = 1/(\rho+L)$, ignore the same residual error term, and obtain

$$\Big\|\sum_{m\in\mathcal{M}_c^k}\Delta_m^k\Big\|^2 \leqslant \frac{|\mathcal{M}_c^k|}{M}\|\nabla f(\boldsymbol{\omega}^k)\|^2 \tag{25}$$

If we can further show that

$$\|\Delta_m^k\|^2 \leqslant \frac{1}{M^2}\|\nabla f(\boldsymbol{\omega}^k)\|^2, \quad \forall m \in \mathcal{M}_c^k \tag{26}$$

then we can prove that Formula (25) holds. However, directly checking Formula (26) in the local worker is impossible because obtaining the fully aggregated gradient $\nabla f(\boldsymbol{\omega}^k)$ requires information from all the workers. It does not make sense to reduce uploads if the fully aggregated gradient has been obtained. Instead, we approximate $\|\nabla f(\boldsymbol{\omega}^k)\|^2$ in Formula (26) as follows:

$$\|\nabla f(\boldsymbol{\omega}^k)\|^2 \approx \frac{1}{\gamma^2}\sum_{d=1}^D \alpha_d\|\boldsymbol{\omega}^{k+1-d} - \boldsymbol{\omega}^{k-d}\|^2 \tag{27}$$

where $\{\alpha_d\}_{d=1}^D$ are constant weights. The fundamental reason here is that, as f is smooth, $\nabla f(\boldsymbol{\omega}^k)$ can be approximated by weighted previous gradients or parameter differences.

Building upon Formulas (26) and (27), we will include worker m in $\mathcal{M}_c^k$ if it satisfies

$$\|\Delta_m^k\|^2 \leqslant \frac{1}{\gamma^2 M^2}\sum_{d=1}^D \alpha_d\|\boldsymbol{\omega}^{k+1-d} - \boldsymbol{\omega}^{k-d}\|^2 \tag{28}$$

Although the intuition for Formula (28) is not mathematically strict, we will show that the convergence of the algorithm is guaranteed under this selection rule. In summary, LE-SIGNGD can be implemented as follows: At iteration k, the server broadcasts the learning parameter to all workers; each worker calculates the local gradient, adds the residual error, and compresses the local information; the worker in $\mathcal{M}^k$ selected by Formula (28) will upload the local information to the server; the server aggregates the fresh compressed gradient from the selected workers $\mathcal{M}^k$ and the outdated gradient information (stored in the server) from $\mathcal{M}_c^k$ to update the parameter. The LE-SIGNGD algorithm is summarized in Algorithm 2.

Algorithm 2 LE-SIGNGD

Input: step size $\gamma > 0$, and $\{\alpha_d\}_{d=1}^D$.
Initialize: $\boldsymbol{\omega}^0 = \mathbf{0}$; $\tilde{\boldsymbol{g}}_m^0, \boldsymbol{e}_m^0 = \mathbf{0}, \forall m \in \mathcal{M}$.
for $k = 0, 1, \ldots, K$ **do**
  server broadcasts $\boldsymbol{\omega}^k$ to all workers.
  for worker $m = 1, \ldots, M$ **do**
    worker m computes $\boldsymbol{g}_m^k = \nabla f_m(\boldsymbol{\omega}^k) + \boldsymbol{e}_m^k$.
    $\mathcal{C}(\boldsymbol{g}_m^k) := (\|\boldsymbol{g}_m^k\|_1/d)\,\mathrm{sign}(\boldsymbol{g}_m^k)$.
    if worker m violates Formula (28) **then**
      worker m uploads $\|\boldsymbol{g}_m^k\|_1$ and $\mathrm{sign}(\boldsymbol{g}_m^k)$.
      worker m updates $\boldsymbol{e}_m^{k+1} = \boldsymbol{g}_m^k - \mathcal{C}(\boldsymbol{g}_m^k)$.
      set $\tilde{\boldsymbol{g}}_m^k = \boldsymbol{g}_m^k$ for worker m.
    else
      worker m uploads nothing.
      set $\tilde{\boldsymbol{g}}_m^k = \tilde{\boldsymbol{g}}_m^{k-1}$ and $\boldsymbol{e}_m^{k+1} = \boldsymbol{e}_m^k$.
    end if
  end for
  server updates parameter $\boldsymbol{\omega}$ according to Eq. (18).
end for

3.2 Convergence analysis

In this section, we will establish the convergence of LE-SIGNGD under the following assumption.

Assumption 3 For a given parameter $\boldsymbol{\omega} \in \mathbb{R}^d$, the local gradient in workers i and j satisfies

$$\langle \nabla f_i(\boldsymbol{\omega}), \nabla f_j(\boldsymbol{\omega}) \rangle \geqslant 0 \tag{29}$$

Although Assumption 3 is non-standard, it can be verified reasonably in real-world experiments. For example, in the logistic regression problem, i.e., $f_m(\boldsymbol{\omega}) = \log(1 + \exp(-\boldsymbol{y}_m^{\mathrm{T}} \boldsymbol{X}_m \boldsymbol{\omega}))$, where $\{\boldsymbol{X}_m \in \mathbb{R}^{n_m \times d}, \ \boldsymbol{y}_m \in \mathbb{R}^{n_m \times 1}\}$ are data in worker m, we then have

$$\langle \nabla f_i(\boldsymbol{\omega}), \nabla f_j(\boldsymbol{\omega}) \rangle = \frac{(\boldsymbol{X}_i^{\mathrm{T}} \boldsymbol{y}_i)^{\mathrm{T}} \boldsymbol{X}_j^{\mathrm{T}} \boldsymbol{y}_j}{(1 + \mathrm{e}^{\boldsymbol{y}_i^{\mathrm{T}} \boldsymbol{X}_i \boldsymbol{\omega}})(1 + \mathrm{e}^{\boldsymbol{y}_j^{\mathrm{T}} \boldsymbol{X}_j \boldsymbol{\omega}})} \tag{30}$$

If we distribute the data randomly and evenly to each worker, which means that the data in workers i and j are not significantly different, then $(\boldsymbol{X}_i^{\mathrm{T}} \boldsymbol{y}_i)^{\mathrm{T}} \boldsymbol{X}_j^{\mathrm{T}} \boldsymbol{y}_j \geqslant 0$ and Assumption 3 hold.

As f^* denotes the optimal value of Problem (1), we define a vital Lyapunov function as follows:

$$\mathcal{L}^k := f(\boldsymbol{\nu}^k) - f^* + \sum_{d=1}^{D} \beta_d \|\boldsymbol{\omega}^{k+1-d} - \boldsymbol{\omega}^{k-d}\|^2 + \tau \sum_{m=1}^{M} \|\boldsymbol{e}_m^k\|^2 \tag{31}$$

where τ and $\{\beta_d\}_{d=1}^D$ are constants that will be determined later. The Lyapunov function is coupled with the selection rule Formula (28) that contains the parameter difference terms. Compared with LAQ[16] and Lazily Aggregated Gradient (LAG)[26], our Lyapunov function introduces a residual term to cope with the difficulties caused by multiple technical combinations. We will start with an important descent lemma of the Lyapunov function $\mathcal{L}^k$.

Lemma 5 (descent lemma) Under Assumptions 1 and 3, if the step size γ and constant weights $\{\alpha_d\}_{d=1}^D$ are chosen properly, then the Lyapunov function satisfies

$$\mathcal{L}^{k+1} - \mathcal{L}^k \leqslant -c_f \|\nabla f(\boldsymbol{\omega}^k)\|^2 - c_e \sum_{m=1}^{M} \|\boldsymbol{e}_m^k\|^2 - \sum_{d=1}^{D} c_d \|\boldsymbol{\omega}^{k+1-d} - \boldsymbol{\omega}^{k-d}\|^2 \tag{32}$$

where constants $c_f, c_e, c_1, \ldots, c_D \geqslant 0$ depend on γ, τ, $\{\alpha_d\}$, and $\{\beta_d\}$ (see the Appendix for details).

In the proof of Lemma 5, we combined Lemma 2 to conduct a further analysis of the residual term. Moreover, we discussed the choice of parameters to satisfy Formula (32).

Theorem 2 Under Assumptions 1 and 3, if the step size γ and constant weights $\{\alpha_d\}_{d=1}^D$ are selected properly, let the sequence $\{\boldsymbol{\omega}^k\}_{k \geqslant 0}$ be generated by LE-SIGNGD, then

$$\begin{cases} \min\limits_{0 \leqslant k \leqslant K} \|\boldsymbol{\omega}^{k+1} - \boldsymbol{\omega}^k\|^2 = o\left(\dfrac{1}{K}\right); \\ \min\limits_{0 \leqslant k \leqslant K} \|\nabla f(\boldsymbol{\omega}^k)\|^2 = o\left(\dfrac{1}{K}\right) \end{cases} \tag{33}$$

Theorem 2 shows that LE-SIGNGD can achieve an order of convergence rate identical to the GD method with the judiciously designed lazy gradient aggregation rule Formula (28).

Remark 3 Note that Theorem 1 shows that the EF-SIGNGD algorithm only has a sub-linear convergence rate $\mathcal{O}(1/\sqrt{K})$ with the decreasing step size $\gamma = \mathcal{O}(1/\sqrt{K})$. Theorem 2 does not require additional assumptions, and is more general. If we assume Formula (29) holds, we can also easily prove that the EF-SIGNGD algorithm has the same convergence rate as the GD method.

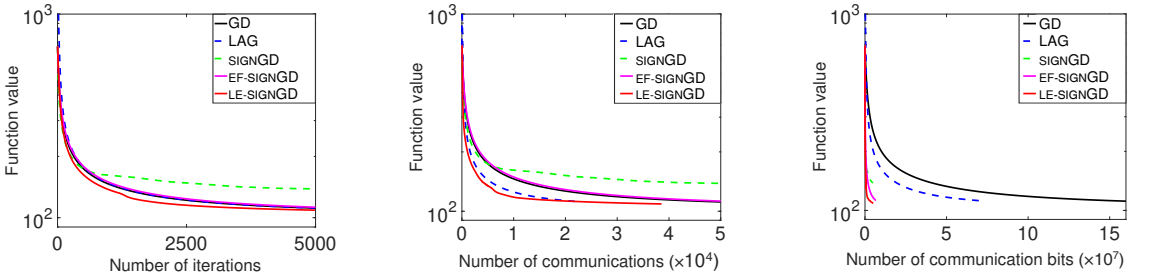
4 Numerical Result

This section contains some numerical experiments to demonstrate the effectiveness of the proposed two distributed algorithms: EF-SIGNGD and LE-SIGNGD. We evaluate the performance of these algorithms for the regularized logistic regression problems as

$$f_m(\boldsymbol{\omega}) = \log(1 + \exp(-\boldsymbol{y}_m^{\mathrm{T}} \boldsymbol{X}_m \boldsymbol{\omega})) + \frac{\lambda}{2} \|\boldsymbol{\omega}\|^2 \tag{34}$$

By default, we consider one server and ten workers in the distributed system, and the regularization parameter is set to $\lambda = 0.001$. We use the GD, SIGNGD, and LAG[26] algorithms as benchmarks. GD is described in Eq. (2) and SIGNGD is updated by Eq. (3). LAG improves the basic GD algorithm by introducing lazy aggregation techniques. To optimize performance and ensure stability, we choose the decreasing step size $\gamma_k = 1/(\sqrt{k}L)$ for SIGNGD. And step size for other algorithms is chosen as $\gamma = 1/L$. For LE-SIGNGD, the constant weights $\{\alpha_d\}$ are set to $\alpha_d = (D - d + 1)/D$ with $D = 10$.

We first consider a synthetic dataset $\{\boldsymbol{X}_m \in \mathbb{R}^{200 \times 100}\}_{m=1,\ldots,10}$, which is synthesized with increasing smoothness constants. As shown in Fig. 1, the proposed two algorithms achieve the same iteration complexity as GD and reduce the needed communication bits by several orders of magnitude compared with



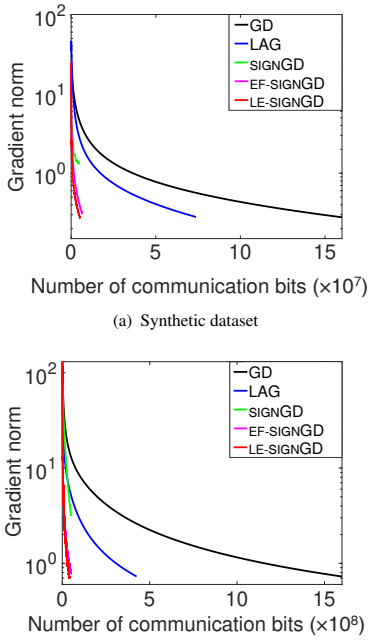
(a) Function value vs. number of iterations (b) Function value vs. number of communications (c) Function value vs. number of communication bits

Fig. 1 Objective function value vs. iteration and communication cost in a synthetic dataset.

LAG and GD. Moreover, the LE-SIGNGD algorithm saves communication bits and rounds simultaneously without sacrificing the desired convergence properties.

Performance is also tested on a real dataset, Gisette, which is constructed from the MNIST data[31]. The dataset contains 6000 samples, and each sample $\boldsymbol{x}_{m,n} \in \mathbb{R}^{5000}$. To reduce the computational cost, we randomly sampled 2000 samples and projected them to dimension 1000 by Principal Component Analysis (PCA). Figure 2 shows the test results in terms of iteration and communication cost. In Fig. 2a, we can see that our proposed two algorithms can achieve comparable performance as GD and outperform SIGNGD. As shown in Fig. 2b, LE-SIGNGD requires fewer communication rounds than EF-SIGNGD and GD because of the lazy selection rule, but more rounds than LAG due to gradient compression. Nevertheless, the total number of communication bits of our proposed algorithms is significantly smaller than that of LAG and GD, as demonstrated in Fig. 2c.

Figure 3 records the convergence performance of the gradient norm for different algorithms, which also shows that our algorithm can obtain better performance.



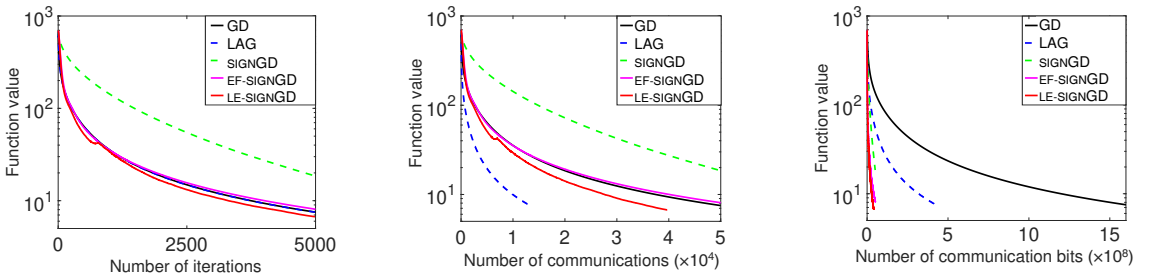
(a) Synthetic dataset

(b) Real dataset

Fig. 3 Gradient norm vs. communication cost in two datasets.

5 Conclusion

This paper proposed two communication-efficient algorithms for distributed learning tasks. EF-SIGNGD used the sign-based method to reduce the communication bits by several orders of magnitude. LE-SIGNGD further introduced the lazily aggregated technique to save both communication bits and rounds. Convergence guarantees have been provided under some mild assumptions. The effectiveness of the two algorithms has also been demonstrated by empirical performance on both synthetic and real datasets.



(a) Function value vs. number of iterations (b) Function value vs. number of communications (c) Function value vs. number of communications bits

Fig. 2 Objective function value vs. iteration and communication cost in a real dataset.

Appendix

Proof of Lemma 2

As shown in Lemma 1, $\mathcal{C}(\cdot)$ defined in Formula (5) is a δ-approximate compressor. Combined with the definition of the error sequence, we have

$$\begin{aligned}&\|\boldsymbol{e}_m^{k+1}\|^2 = \|\boldsymbol{g}_m^k - \mathcal{C}(\boldsymbol{g}_m^k)\|^2 \leqslant (1-\delta)\|\boldsymbol{g}_m^k\|^2 = \\ &(1-\delta)\|\nabla f_m(\boldsymbol{\omega}^k) + \boldsymbol{e}_m^k\|^2 \leqslant \\ &(1-\delta)\big[(1+\eta)\|\boldsymbol{e}_m^k\|^2 + (1+1/\eta)\|\nabla f_m(\boldsymbol{\omega}^k)\|^2\big] \quad (35)\end{aligned}$$

where we used Young's inequality (for any $\eta > 0$). Notice that $\boldsymbol{e}_m^0 = \boldsymbol{0}$ and $\nabla f_m(\boldsymbol{\omega}^k)$ is bounded in Assumption 2, so a simple computation yields

$$\begin{aligned}&\|\boldsymbol{e}_m^{k+1}\|^2 \leqslant (1-\delta)(1+\eta)\|\boldsymbol{e}_m^k\|^2 + (1-\delta)(1+1/\eta)\cdot \\ &\|\nabla f_m(\boldsymbol{\omega}^k)\|^2 \leqslant (1-\delta)(1+\eta)^{k+1}\|\boldsymbol{e}_m^0\|^2 + \\ &\sum_{t=0}^{k}\big[(1-\delta)(1+\eta)\big]^{k-t}(1-\delta)(1+1/\eta)\|\nabla f_m(\boldsymbol{\omega}^t)\|^2 \leqslant \\ &\sum_{t=0}^{\infty}\big[(1-\delta)(1+\eta)\big]^{k-t}(1-\delta)(1+1/\eta)\sigma_m^2 = \\ &\frac{(1-\delta)(1+1/\eta)}{1-(1-\delta)(1+\eta)}\sigma_m^2 \quad (36)\end{aligned}$$

Let us select $\eta = \frac{\delta}{2(1-\delta)}$ such that $1 + 1/\eta = (2-\delta)/\delta \leqslant 2/\delta$. Thus, we have

$$\|\boldsymbol{e}_m^{k+1}\|^2 \leqslant \frac{2(1-\delta)(1+1/\eta)}{\delta}\sigma_m^2 \leqslant \frac{4(1-\delta)}{\delta^2}\sigma_m^2 \quad (37)$$

■

Proof of Lemma 3

From Formula (11), the sequence $\{\boldsymbol{\nu}^k\}$ has the property

$$\begin{aligned}\boldsymbol{\nu}^{k+1} &= \boldsymbol{\omega}^{k+1} - \gamma\sum_{m=1}^{M}\boldsymbol{e}_m^{k+1} = \boldsymbol{\omega}^k - \gamma\sum_{m=1}^{M}\mathcal{C}(\boldsymbol{g}_m^k) - \\ &\gamma\sum_{m=1}^{M}\boldsymbol{e}_m^{k+1} = \boldsymbol{\omega}^k - \gamma\sum_{m=1}^{M}\boldsymbol{g}_m^k = \\ &\boldsymbol{\omega}^k - \gamma\sum_{m=1}^{M}\boldsymbol{e}_m^k - \gamma\sum_{m=1}^{M}\nabla f_m(\boldsymbol{\omega}^k) = \\ &\boldsymbol{\nu}^k - \gamma\nabla f(\boldsymbol{\omega}^k) \quad (38)\end{aligned}$$

Given that the function f is L-smooth, i.e.,

$$\begin{aligned}&\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\|_2 \leqslant L\|\boldsymbol{x}-\boldsymbol{y}\|_2, \\ &f(\boldsymbol{x}) - f(\boldsymbol{y}) \leqslant \langle\nabla f(\boldsymbol{y}), \boldsymbol{x}-\boldsymbol{y}\rangle + \frac{L}{2}\|\boldsymbol{x}-\boldsymbol{y}\|^2 \quad (39)\end{aligned}$$

We have

$$\begin{aligned}&f(\boldsymbol{\nu}^{k+1}) - f(\boldsymbol{\nu}^k) \leqslant \langle\nabla f(\boldsymbol{\nu}^k), \boldsymbol{\nu}^{k+1} - \boldsymbol{\nu}^k\rangle + \\ &\frac{L}{2}\|\boldsymbol{\nu}^{k+1} - \boldsymbol{\nu}^k\|^2\langle\nabla f(\boldsymbol{\omega}^k), \boldsymbol{\nu}^{k+1} - \boldsymbol{\nu}^k\rangle + \\ &\langle\nabla f(\boldsymbol{\nu}^k) - f(\boldsymbol{\omega}^k), \boldsymbol{\nu}^{k+1} - \boldsymbol{\nu}^k\rangle + \\ &\frac{L}{2}\|\boldsymbol{\nu}^{k+1} - \boldsymbol{\nu}^k\|^2 \leqslant -\gamma\Big(1 - \frac{L\gamma}{2}\Big)\|\nabla f(\boldsymbol{\omega}^k)\|^2 + \\ &\frac{\rho}{2}\|\boldsymbol{\nu}^{k+1} - \boldsymbol{\nu}^k\|^2 + \frac{1}{2\rho}\|\nabla f(\boldsymbol{\nu}^k) - f(\boldsymbol{\omega}^k)\|^2 \leqslant \\ &-\gamma\Big[1 - \frac{(L+\rho)\gamma}{2}\Big]\|\nabla f(\boldsymbol{\omega}^k)\|^2 + \frac{L^2}{2\rho}\|\boldsymbol{\nu}^k - \boldsymbol{\omega}^k\|^2 \leqslant \\ &-\gamma\Big[1 - \frac{(\rho+L)\gamma}{2}\Big]\|\nabla f(\boldsymbol{\omega}^k)\|^2 + \frac{L^2\gamma^2}{2\rho}\Big\|\sum_{m=1}^{M}\boldsymbol{e}_m^k\Big\|^2 \quad (40)\end{aligned}$$

The third inequality follows from the mean-value inequality and holds for any $\rho > 0$. ■

Proof of Theorem 1

The one-step descent of EF-SIGNGD is given in Lemma 3, and the residual error is bounded in Lemma 2. Summing the terms in Formula (40) over k yields

$$\begin{aligned}&f(\boldsymbol{\nu}^{K+1}) - f(\boldsymbol{\nu}^0) \leqslant -\gamma\left[1 - \frac{(\rho+L)\gamma}{2}\right]\cdot \\ &\sum_{k=0}^{K}\|\nabla f(\boldsymbol{\omega}^k)\|^2 + \frac{2L^2\gamma^2(1-\delta)}{\rho\delta^2}(K+1)M\sigma^2 \quad (41)\end{aligned}$$

where we used

$$\begin{aligned}\|\sum_{m=1}^{M}\boldsymbol{e}_m^k\|^2 \leqslant M\sum_{m=1}^{M}\|\boldsymbol{e}_m^k\|^2 \leqslant M\frac{4(1-\delta)}{\delta^2}\cdot \\ \sum_{m=1}^{M}\sigma_m^2 := \frac{4(1-\delta)}{\delta^2}M\sigma^2,\end{aligned}$$

where $\sigma^2 := \sum_{m=1}^{M}\sigma_m^2$.

Noticed that $\boldsymbol{\nu}^0 = \boldsymbol{\omega}^0$ and f^* is the optimal value. Given step size $\gamma < 2/(\rho+L)$, rearranging the terms in Formula (41) and averaging over k can lead to

$$\begin{aligned}&\frac{1}{K+1}\sum_{k=0}^{K}\|\nabla f(\boldsymbol{\omega}^k)\|^2 \leqslant \\ &\frac{f(\boldsymbol{\nu}^0) - f(\boldsymbol{\nu}^{K+1})}{\gamma(K+1)\big[1 - \frac{(\rho+L)\gamma}{2}\big]} + \frac{2L^2\gamma(1-\delta)M\sigma^2}{\rho\delta^2\big[1 - \frac{(\rho+L)\gamma}{2}\big]} \leqslant \\ &\frac{2(f^0 - f^*)}{\gamma(K+1)[2-(\rho+L)\gamma]} + \frac{4L^2\gamma(1-\delta)M\sigma^2}{\rho\delta^2[2-(\rho+L)\gamma]} \quad (42)\end{aligned}$$

Therefore, we finished this proof. ■

Proof of Lemma 4

The sequence $\{\boldsymbol{\nu}^k\}$ in LE-SIGNGD has the following property:

$$\begin{aligned}\boldsymbol{\nu}^{k+1} &= \boldsymbol{\omega}^{k+1} - \gamma\sum_{m=1}^{M}\boldsymbol{e}_m^{k+1} = \boldsymbol{\omega}^k - \gamma\sum_{m=1}^{M}\mathcal{C}(\boldsymbol{g}_m^k) + \\ &\gamma\sum_{m\in\mathcal{M}_c^k}\Delta_m^k - \gamma\sum_{m=1}^{M}\boldsymbol{e}_m^{k+1} =\end{aligned}$$

$$\boldsymbol{\omega}^k-\gamma\sum_{m=1}^{M}\boldsymbol{g}_m^k+\gamma\sum_{m\in\mathcal{M}_c^k}\Delta_m^k=\boldsymbol{\omega}^k-\gamma\sum_{m=1}^{M}\boldsymbol{e}_m^k-$$
$$\gamma\sum_{m=1}^{M}\nabla f_m(\boldsymbol{\omega}^k)+\gamma\sum_{m\in\mathcal{M}_c^k}\Delta_m^k=$$
$$\boldsymbol{\nu}^k-\gamma\nabla f(\boldsymbol{\omega}^k)+\gamma\sum_{m\in\mathcal{M}_c^k}\Delta_m^k \tag{43}$$

The smoothness of f is

$$f(\boldsymbol{\nu}^{k+1})-f(\boldsymbol{\nu}^k)\leqslant\langle\nabla f(\boldsymbol{\nu}^k),\boldsymbol{\nu}^{k+1}-\boldsymbol{\nu}^k\rangle+$$
$$\frac{L}{2}\|\boldsymbol{\nu}^{k+1}-\boldsymbol{\nu}^k\|^2\leqslant$$
$$\Big\langle\nabla f(\boldsymbol{\omega}^k),-\gamma\nabla f(\boldsymbol{\omega}^k)+\gamma\sum_{m\in\mathcal{M}_c^k}\Delta_m^k\Big\rangle+$$
$$\Big\langle\nabla f(\boldsymbol{\nu}^k)-f(\boldsymbol{\omega}^k),\boldsymbol{\nu}^{k+1}-\boldsymbol{\nu}^k\Big\rangle+\frac{L}{2}\|\boldsymbol{\nu}^{k+1}-\boldsymbol{\nu}^k\|^2\overset{\text{(a)}}{\leqslant}$$
$$-\gamma\|\nabla f(\boldsymbol{\omega}^k)\|^2+\gamma\Big\langle\nabla f(\boldsymbol{\omega}^k),\sum_{m\in\mathcal{M}_c^k}\Delta_m^k\Big\rangle+$$
$$\frac{\rho+L}{2}\|\boldsymbol{\nu}^{k+1}-\boldsymbol{\nu}^k\|^2+\frac{1}{2\rho}\|\nabla f(\boldsymbol{\nu}^k)-f(\boldsymbol{\omega}^k)\|^2\overset{\text{(b)}}{\leqslant}$$
$$-\frac{\gamma}{2}\|\nabla f(\boldsymbol{\omega}^k)\|^2+\frac{\gamma}{2}\Big\|\sum_{m\in\mathcal{M}_c^k}\Delta_m^k\Big\|^2+$$
$$\Big(\frac{\rho+L}{2}-\frac{1}{2\gamma}\Big)\|\boldsymbol{\nu}^{k+1}-\boldsymbol{\nu}^k\|^2+\frac{L^2}{2\rho}\|\boldsymbol{\nu}^k-\boldsymbol{\omega}^k\|^2\leqslant$$
$$-\frac{\gamma}{2}\Big\|\nabla f(\boldsymbol{\omega}^k)\Big\|^2+\frac{L^2\gamma^2}{2\rho}\Big\|\sum_{m=1}^{M}\boldsymbol{e}_m^k\Big\|^2+$$
$$\Big(\frac{\rho+L}{2}-\frac{1}{2\gamma}\Big)\|\boldsymbol{\nu}^{k+1}-\boldsymbol{\nu}^k\|^2+\frac{\gamma}{2}\Big\|\sum_{m\in\mathcal{M}_c^k}\Delta_m^k\Big\|^2 \tag{44}$$

where (a) uses Young's inequality (for any $\rho>0$), and (b) uses $2\langle\boldsymbol{a},\boldsymbol{b}\rangle=\|\boldsymbol{a}\|^2+\|\boldsymbol{b}\|^2-\|\boldsymbol{a}-\boldsymbol{b}\|^2$, i.e.,

$$\gamma\Big\langle\nabla f(\boldsymbol{\omega}^k),\sum_{m\in\mathcal{M}_c^k}\Delta_m^k\Big\rangle=\frac{\gamma}{2}\|\nabla f(\boldsymbol{\omega}^k)\|^2+$$
$$\frac{\gamma}{2}\Big\|\sum_{m\in\mathcal{M}_c^k}\Delta_m^k\Big\|^2-\frac{1}{2\gamma}\|\boldsymbol{\nu}^{k+1}-\boldsymbol{\nu}^k\|^2 \tag{45}$$

■

Proof of Lemma 5

By the definition of $\mathcal{L}^k$ in Formula (31), it follows that

$$\mathcal{L}^{k+1}-\mathcal{L}^k=$$
$$f(\boldsymbol{\nu}^{k+1})-f(\boldsymbol{\nu}^k)+\sum_{d=1}^{D}\beta_d\|\boldsymbol{\omega}^{k+2-d}-\boldsymbol{\omega}^{k+1-d}\|^2-$$
$$\sum_{d=1}^{D}\beta_d\|\boldsymbol{\omega}^{k+1-d}-\boldsymbol{\omega}^{k-d}\|^2+$$
$$\tau\Big(\sum_{m=1}^{M}\|\boldsymbol{e}_m^{k+1}\|^2-\sum_{m=1}^{M}\|\boldsymbol{e}_m^k\|^2\Big)\leqslant$$
$$-\frac{\gamma}{2}\Big\|\nabla f(\boldsymbol{\omega}^k)\Big\|^2+\frac{L^2\gamma^2}{2\rho}\Big\|\sum_{m=1}^{M}\boldsymbol{e}_m^k\Big\|^2+\frac{\gamma}{2}\Big\|\sum_{m\in\mathcal{M}_c^k}\Delta_m^k\Big\|^2+$$
$$\left(\frac{\rho+L}{2}-\frac{1}{2\gamma}\right)\|\boldsymbol{\nu}^{k+1}-\boldsymbol{\nu}^k\|^2+\beta_1\|\boldsymbol{\omega}^{k+1}-\boldsymbol{\omega}^k\|^2+$$
$$\sum_{d=1}^{D-1}(\beta_{d+1}-\beta_d)\|\boldsymbol{\omega}^{k+1-d}-\boldsymbol{\omega}^{k-d}\|^2-$$
$$\beta_D\|\boldsymbol{\omega}^{k+1-D}-\boldsymbol{\omega}^{k-D}\|^2+$$
$$\tau\Big(\sum_{m=1}^{M}\|\boldsymbol{e}_m^{k+1}\|^2-\sum_{m=1}^{M}\|\boldsymbol{e}_m^k\|^2\Big) \tag{46}$$

Using Young's inequality, for any η_1 and $\eta_2>0$,

$$\|\boldsymbol{\omega}^{k+1}-\boldsymbol{\omega}^k\|^2=\Big\|\boldsymbol{\nu}^{k+1}-\boldsymbol{\nu}^k+\gamma\sum_{m=1}^{M}(\boldsymbol{e}_m^{k+1}-\boldsymbol{e}_m^k)\Big\|^2\leqslant$$
$$(1+\eta_1)\|\boldsymbol{\nu}^{k+1}-\boldsymbol{\nu}^k\|^2+(1+1/\eta_1)\gamma^2\Big\|\sum_{m=1}^{M}(\boldsymbol{e}_m^{k+1}-\boldsymbol{e}_m^k)\Big\|^2 \tag{47}$$

$$\|\boldsymbol{\nu}^{k+1}-\boldsymbol{\nu}^k\|^2=\Big\|-\gamma\nabla f(\boldsymbol{\omega}^k)+\gamma\sum_{m\in\mathcal{M}_c^k}\Delta_m^k\Big\|^2\leqslant$$
$$(1+\eta_2)\gamma^2\||\nabla f(\boldsymbol{\omega}^k)\|^2+(1+1/\eta_2)\gamma^2\Big\|\sum_{m\in\mathcal{M}_c^k}\Delta_m^k\Big\|^2 \tag{48}$$

Plugging Formulas (47) and (48) into Formula (46), we arrive at

$$\mathcal{L}^{k+1}-\mathcal{L}^k\leqslant$$
$$-\frac{\gamma}{2}\|\nabla f(\boldsymbol{\omega}^k)\|^2+\left[\beta_1(1+\eta_1)+\frac{\rho+L}{2}-\frac{1}{2\gamma}\right]\cdot$$
$$\|\boldsymbol{\nu}^{k+1}-\boldsymbol{\nu}^k\|^2+\frac{\gamma}{2}\Big\|\sum_{m\in\mathcal{M}_c^k}\Delta_m^k\Big\|^2+\frac{L^2\gamma^2}{2\rho}\Big\|\sum_{m=1}^{M}\boldsymbol{e}_m^k\Big\|^2+$$
$$\sum_{d=1}^{D-1}(\beta_{d+1}-\beta_d)\|\boldsymbol{\omega}^{k+1-d}-\boldsymbol{\omega}^{k-d}\|^2-\beta_D\|\boldsymbol{\omega}^{k+1-D}-\boldsymbol{\omega}^{k-D}\|^2+$$
$$(1+1/\eta_1)\beta_1\gamma^2M\Big(\sum_{m=1}^{M}\|\boldsymbol{e}_m^{k+1}\|^2+\sum_{m=1}^{M}\|\boldsymbol{e}_m^k\|^2\Big)+$$
$$\tau\Big(\sum_{m=1}^{M}\|\boldsymbol{e}_m^{k+1}\|^2-\sum_{m=1}^{M}\|\boldsymbol{e}_m^k\|^2\Big)\leqslant$$
$$-\Big[\frac{\gamma}{2}-\Big(\beta_1(1+\eta_1)+\frac{\rho+L}{2}-\frac{1}{2\gamma}\Big)(1+\eta_2)\gamma^2\Big]\|\nabla f(\boldsymbol{\omega}^k)\|^2+$$
$$\Big[\frac{\gamma}{2}+\Big(\beta_1(1+\eta_1)+\frac{\rho+L}{2}-\frac{1}{2\gamma}\Big)(1+1/\eta_2)\gamma^2\Big]\Big\|\sum_{m\in\mathcal{M}_c^k}\Delta_m^k\Big\|^2+$$

$$\sum_{d=1}^{D-1}(\beta_{d+1}-\beta_d)\|\boldsymbol{\omega}^{k+1-d}-\boldsymbol{\omega}^{k-d}\|^2-\beta_D\|\boldsymbol{\omega}^{k+1-D}-\boldsymbol{\omega}^{k-D}\|^2+\Big[(1+1/\eta_1)\beta_1\gamma^2M+\tau\Big]\sum_{m=1}^{M}\|\boldsymbol{e}_m^{k+1}\|^2+\Big[(1+1/\eta_1)\beta_1\gamma^2M+\frac{L^2\gamma^2}{2\rho}M-\tau\Big]\sum_{m=1}^{M}\|\boldsymbol{e}_m^{k}\|^2 \quad (49)$$

Noticed that in the proof of Lemma 2, Formula (35) shows that $\|\boldsymbol{e}_m^{k+1}\|^2$ satisfies

$$\|\boldsymbol{e}_m^{k+1}\|^2 \leqslant \tilde{c}_1\|\boldsymbol{e}_m^{k}\|^2+\tilde{c}_2\|\nabla f_m(\boldsymbol{\omega}^k)\|^2 \quad (50)$$

where $\tilde{c}_1 := (1-\delta)(1+\eta)$ and $\tilde{c}_2 = (1-\delta)(1+1/\eta)$. In Lemma 2, η is chosen as $\dfrac{\delta}{2(1-\delta)}$, then we can see that $\tilde{c}_1 = \dfrac{2-\delta}{2} < 1$.

Combined with the selection rule Formula (28), i.e.,

$$\|\Delta_m^k\|^2 \leqslant \frac{1}{\gamma^2M^2}\sum_{d=1}^{D}\alpha_d\|\boldsymbol{\omega}^{k+1-d}-\boldsymbol{\omega}^{k-d}\|^2,\ \forall m\in\mathcal{M}_c^k \quad (51)$$

and Assumption 3, i.e.,

$$\|\nabla f(\boldsymbol{\omega}^k)\|^2 = \Big\|\sum_{m=1}^{M}\nabla f_m(\boldsymbol{\omega}^k)\Big\|^2 \geqslant \sum_{m=1}^{M}\|\nabla f_m(\boldsymbol{\omega}^k)\|^2 \quad (52)$$

we have (with $c_\delta := \dfrac{\gamma}{2}+\Big(\beta_1(1+\eta_1)+\dfrac{\rho+L}{2}-\dfrac{1}{2\gamma}\Big)(1+1/\eta_2)\gamma^2$)

$$\begin{aligned}
&\mathcal{L}^{k+1}-\mathcal{L}^k \leqslant \\
&-\Big[\frac{\gamma}{2}-\Big(\beta_1(1+\eta_1)+\frac{\rho+L}{2}-\frac{1}{2\gamma}\Big)(1+\eta_2)\gamma^2- \\
&((1+1/\eta_1)\beta_1\gamma^2M+\tau)\tilde{c}_2\Big]\|\nabla f(\boldsymbol{\omega}^k)\|^2- \\
&\Big[\tau-((1+1/\eta_1)\beta_1\gamma^2M+\tau)\tilde{c}_1-(1+1/\eta_1)\beta_1\gamma^2M- \\
&\frac{L^2\gamma^2M}{2\rho}\Big]\sum_{m=1}^{M}\|\boldsymbol{e}_m^k\|^2+ \\
&\Big[\frac{\gamma}{2}+\Big(\beta_1(1+\eta_1)+\frac{\rho+L}{2}-\frac{1}{2\gamma}\Big)(1+1/\eta_2)\gamma^2\Big]\cdot \\
&\Big\|\sum_{m\in\mathcal{M}_c^k}\Delta_m^k\Big\|^2+\sum_{d=1}^{D-1}(\beta_{d+1}-\beta_d)\|\boldsymbol{\omega}^{k+1-d}-\boldsymbol{\omega}^{k-d}\|^2- \\
&\beta_D\|\boldsymbol{\omega}^{k+1-D}-\boldsymbol{\omega}^{k-D}\|^2 \leqslant \\
&-\Big[\frac{\gamma}{2}-\Big(\beta_1(1+\eta_1)+\frac{\rho+L}{2}-\frac{1}{2\gamma}\Big)(1+\eta_2)\gamma^2- \\
&\big((1+1/\eta_1)\beta_1\gamma^2M+\tau\big)\tilde{c}_2\Big]\Big\|\nabla f(\boldsymbol{\omega}^k)\Big\|^2- \\
&\Big[(1-\tilde{c}_1)\tau-(1+\tilde{c}_1)(1+1/\eta_1)\beta_1\gamma^2M-\frac{L^2\gamma^2M}{2\rho}\Big]\cdot \\
&\sum_{m=1}^{M}\|\boldsymbol{e}_m^k\|^2-\sum_{d=1}^{D-1}\Big(\beta_d-\beta_{d+1}-\frac{c_\delta\alpha_d|\mathcal{M}_c^k|^2}{\gamma^2M^2}\Big)\cdot \\
&\|\boldsymbol{\omega}^{k+1-d}-\boldsymbol{\omega}^{k-d}\|^2-\Big(\beta_D-\frac{c_\delta\alpha_D|\mathcal{M}_c^k|^2}{\gamma^2M^2}\Big)\cdot \\
&\|\boldsymbol{\omega}^{k+1-D}-\boldsymbol{\omega}^{k-D}\|^2
\end{aligned} \quad (53)$$

Furthermore, if the step size γ, parameters τ, $\{\beta_d\}$, and trigger constants $\{\alpha_d\}$ satisfy

$$\begin{cases}
c_f := \dfrac{\gamma}{2}-\Big(\beta_1(1+\eta_1)+\dfrac{\rho+L}{2}-\dfrac{1}{2\gamma}\Big)(1+\eta_2)\gamma^2- \\
\qquad \big((1+1/\eta_1)\beta_1\gamma^2M+\tau\big)\tilde{c}_2 \geqslant 0; \\
c_e := (1-\tilde{c}_1)\tau-(1+\tilde{c}_1)(1+1/\eta_1)\beta_1\gamma^2M- \\
\qquad \dfrac{L^2\gamma^2M}{2\rho} \geqslant 0; \\
c_d := \beta_d-\beta_{d+1}-\Big(\dfrac{\gamma}{2}+\Big(\beta_1(1+\eta_1)+\dfrac{\rho+L}{2}- \\
\qquad \dfrac{1}{2\gamma}\Big)(1+1/\eta_2)\gamma^2\Big)\dfrac{\alpha_d|\mathcal{M}_c^k|^2}{\gamma^2M^2} \geqslant 0, \\
\qquad d=1,\ldots,D-1; \\
c_D := \beta_D-\Big(\dfrac{\gamma}{2}+\Big(\beta_1(1+\eta_1)+\dfrac{\rho+L}{2}-\dfrac{1}{2\gamma}\Big)\cdot \\
\qquad (1+1/\eta_2)\gamma^2\Big)\dfrac{\alpha_D|\mathcal{M}_c^k|^2}{\gamma^2M^2} \geqslant 0
\end{cases} \quad (54)$$

We can see that the Lyapunov function is non-increasing, i.e., $\mathcal{L}^{k+1}\leqslant\mathcal{L}^k$, and the proof is completed. ■

Choice of parameter: We then will show that simple parameter selection can satisfy Formula (54). We can choose $\beta_1 = \frac{1-(\rho+L)\gamma}{2\gamma(1+\eta_1)}$, so that $\beta_1(1+\eta_1)+\frac{\rho+L}{2}-\frac{1}{2\gamma}=0$ (with $\gamma<\frac{1}{\rho+L}$), after rearranging terms, Formula (54) is equivalent to

$$\tau \leqslant \frac{\gamma}{2\tilde{c}_2}\Big[1-\frac{\big(1-(\rho+L)\gamma\big)M\tilde{c}_2}{\eta_1}\Big] \quad (55a)$$

$$\tau \geqslant \frac{\gamma}{2(1-\tilde{c}_1)}\Big[\frac{\big(1-(\rho+L)\gamma\big)(1+\tilde{c}_1)M}{\eta_1}+\frac{L^2M\gamma}{\rho}\Big] \quad (55b)$$

$$\beta_d-\beta_{d+1}-\frac{\alpha_d|\mathcal{M}_c^k|^2}{2\gamma M^2}\geqslant 0,\ d=1,\ldots,D-1 \quad (55c)$$

$$c_D := \beta_D-\frac{\alpha_D|\mathcal{M}_c^k|^2}{2\gamma M^2}\geqslant 0 \quad (55d)$$

Notably, $\tilde{c}_2<1$, ρ, $\eta_1>0$, we can choose the sufficiently large ρ and η_1 to satisfy Formulas (55a) and (55b) for the theoretical guarantee. In practice, the step size γ and the trigger constants $\{\alpha_d\}$ can be chosen as follows:

$$\gamma<\frac{1}{\rho+L};\ \alpha_D\leqslant\frac{2\gamma M^2\beta_D}{|\mathcal{M}_c^k|^2};\ \alpha_d\leqslant\frac{2\gamma M^2(\beta_d-\beta_{d+1})}{|\mathcal{M}_c^k|^2},\quad d=1,\ldots,D-1 \quad (56)$$

Proof of Theorem 2

Lemma 5 implies that

$$\mathcal{L}^{k+1}-\mathcal{L}^{k}\leqslant -c_f\|\nabla f(\boldsymbol{\omega}^k)\|^2- c_e\sum_{m=1}^{M}\|\boldsymbol{e}_m^k\|^2-\sum_{d=1}^{D}c_d\|\boldsymbol{\omega}^{k+1-d}-\boldsymbol{\omega}^{k-d}\|^2\leqslant -c(\gamma;\tau;\{\alpha_d\})\Big(\|\nabla f(\boldsymbol{\omega}^k)\|^2+\sum_{m=1}^{M}\|\boldsymbol{e}_m^k\|^2+\sum_{d=1}^{D}\|\boldsymbol{\omega}^{k+1-d}-\boldsymbol{\omega}^{k-d}\|^2\Big) \tag{57}$$

where the constant $c(\gamma;\tau;\{\alpha_d\})>0$ is defined as $c(\gamma;\tau;\{\alpha_d\}):=\min\{c_f,c_e,c_1,\ldots,c_D\}$.

Rearranging the terms in Formula (57) and summing up both sides over k, we have

$$c(\gamma;\tau;\{\alpha_d\})\sum_{k=0}^{K}\Big(\|\nabla f(\boldsymbol{\omega}^k)\|^2+\sum_{m=1}^{M}\|\boldsymbol{e}_m^k\|^2+\sum_{d=1}^{D}\|\boldsymbol{\omega}^{k+1-d}-\boldsymbol{\omega}^{k-d}\|^2\Big)\leqslant\mathcal{L}^0-\mathcal{L}^{K+1} \tag{58}$$

Given that the Lyapunov function (Formula (31)) is lower bounded by $\mathcal{L}^k\geqslant 0,\forall k$, and $\mathcal{L}^0\leqslant\infty$. Taking $K\to\infty$, we have that

$$c(\gamma;\tau;\{\alpha_d\})\lim_{K\to\infty}\sum_{k=0}^{K}\Big(\|\nabla f(\boldsymbol{\omega}^k)\|^2+\sum_{m=1}^{M}\|\boldsymbol{e}_m^k\|^2+\sum_{d=1}^{D}\|\boldsymbol{\omega}^{k+1-d}-\boldsymbol{\omega}^{k-d}\|^2\Big)\leqslant\mathcal{L}^0 \tag{59}$$

which implies that

$$\sum_{k=0}^{\infty}\|\nabla f(\boldsymbol{\omega}^k)\|^2\leqslant\infty;\quad\sum_{k=0}^{\infty}\|\boldsymbol{\omega}^{k+1}-\boldsymbol{\omega}^k\|^2\leqslant\infty \tag{60}$$

Using the implications of summable sequences in Ref. [32], the theorem follows. ■

Acknowledgment

This work was supported in part by the Core Electronic Devices, High-End Generic Chips, and Basic Software Major Special Projects (No. 2018ZX01028101), and the National Natural Science Foundation of China (Nos. 61907034, 61932001, and 61906200).

References

[1] A. Ahmed, N. Shervashidze, S. Narayanamurthy, V. Josifovski, and A. J. Smola, Distributed large-scale natural graph factorization, in *Proc. 22nd Int. Conf. World Wide Web*, Rio de Janeiro, Brazil, 2013, pp. 37–48.

[2] J. Dean, G. S. Corrado, R. Monga, K. Chen, M. Devin, Q. V. Le, M. Z. Mao, M. Ranzato, A. Senior, P. Tucker, et al., Large scale distributed deep networks, in *Proc. 25th Int. Conf. Neural Information Processing Systems*, Red Hook, NY, USA, 2012, pp. 1223–1231.

[3] M. Li, D. G. Andersen, A. Smola, and K. Yu, Communication efficient distributed machine learning with the parameter server, in *Proc. 27th Int. Conf. Neural Information Processing Systems*, Cambridge, MA, USA, 2014, pp. 19–27.

[4] D. S. Li, Z. Q. Lai, K. S. Ge, Y. M. Zhang, Z. N. Zhang, Q. L. Wang, and H. M. Wang, Hpdl: Towards a general framework for high-performance distributed deep learning, presented at 2019 IEEE 39th Int. Conf. Distributed Computing Systems (ICDCS), Dallas, TX, USA, 2019, pp. 1742–1753.

[5] K. M. Nan, S. C. Liu, J. Z. Du, and H. Liu, Deep model compression for mobile platforms: A survey, *Tsinghua Science and Technology*, vol. 24, no. 6, pp. 677–693, 2019.

[6] J. Q. Huang, W. T. Han, X. Y. Wang, and W. G. Chen, Heterogeneous parallel algorithm design and performance optimization for WENO on the Sunway Taihulight supercomputer, *Tsinghua Science and Technology*, vol. 25, no. 1, pp. 56–67, 2020.

[7] L. Guan, T. Sun, L. B. Qiao, Z. H. Yang, D. S. Li, K. S. Ge, and X. C. Lu, An efficient parallel and distributed solution to nonconvex penalized linear SVMs, *Frontiers of Information Technology & Electronic Engineering*, vol. 21, no. 4, pp. 587–603, 2020.

[8] 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.

[9] G. B. Giannakis, Q. Ling, G. Mateos, I. D. Schizas, and H. Zhu, Decentralized learning for wireless communications and networking, in *Splitting Methods in Communication*, *Imaging*, *Science*, *and Engineering*, R. Glowinski, S. Osher, and W. Yin, eds. Cham, Germany: Springer, 2016, pp. 461–497.

[10] M. I. Jordan, J. D. Lee, and Y. Yang, Communication-efficient distributed statistical inference, *Journal of the American Statistical Association*, vol. 114, no. 526, pp. 668–681, 2019.

[11] A. Nedić, A. Olshevsky, and M. G. Rabbat, Network topology and communication-computation tradeoffs in decentralized optimization, *Proceedings of the IEEE*, vol. 106, no. 5, pp. 953–976, 2018.

[12] S. Zheng, Z. Y. Huang, and J. T. Kwok, Communication-efficient distributed blockwise momentum SGD with error-feedback, arXiv preprint arXiv: 1905.10936, 2019.

[13] Z. X. Guo and S. H. Zhang, Sparse deep nonnegative matrix factorization, *Big Data Mining and Analytics*, vol. 3, no. 1, pp. 13–28, 2020.

[14] F. Ablayev, M. Ablayev, J. Z. Huang, K. Khadiev, N. Salikhova, and D. M. Wu, On quantum methods for machine learning problems part II: Quantum classification algorithms, *Big Data Mining and Analytics*, vol. 3, no. 1, pp. 56–67, 2020.

[15] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, QSGD: Communication-efficient SGD via gradient quantization and encoding, presented at Advances in Neural Information Processing Systems, Long Beach, CA, USA, 2017, pp. 1709–1720.

[16] J. Sun, T. Y. Chen, G. B. Giannakis, and Z. Y. Yang, Communication-efficient distributed learning via lazily

aggregated quantized gradients, presented at Advances in Neural Information Processing Systems, Vancouver, Canada, 2019, pp. 3365–3375.

[17] 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 *Proc. 15th Annu. Conf. Int. Speech Communication Association*, Singapore, 2014, pp. 1058–1062.

[18] J. Bernstein, Y. X. Wang, K. Azizzadenesheli, and A. Anandkumar, signSGD: Compressed optimisation for non-convex problems, in *Proc. 35th Int. Conf. Machine Learning*, Stockholm, Sweden, 2018, pp. 560–569.

[19] J. Bernstein, J. W. Zhao, K. Azizzadenesheli, and A. Anandkumar, signSGD with majority vote is communication efficient and fault tolerant, in *Proc. 7th Int. Conf. Learning Representations*, New Orleans, LA, USA, 2019.

[20] S. P. Karimireddy, Q. Rebjock, S. U. Stich, and M. Jaggi, Error feedback fixes signSGD and other gradient compression schemes, in *Proc. 36th Int. Conf. Machine Learning*, Long Beach, CA, USA, 2019, pp. 3252–3261.

[21] O. Shamir, N. Srebro, and T. Zhang, Communication-efficient distributed optimization using an approximate newton-type method, in *Proc. 31st Int. Conf. Machine Learning*, Beijing, China, 2014, pp. 1000–1008.

[22] Y. C. Zhang and X. Lin, Disco: Distributed optimization for self-concordant empirical loss, in *Proc. 32nd Int. Conf. Machine Learning*, Lille, France, 2015, pp. 362–370.

[23] A. Mokhtari, Q. Ling, and A. Ribeiro, Network newton distributed optimization methods, *IEEE Transactions on Signal Processing*, vol. 65, no. 1, pp. 146–161, 2017.

[24] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, Communication-efficient learning of deep networks from decentralized data, in *Proc. 20th Int. Conf. Artificial Intelligence and Statistics*, Fort Lauderdale, FL, USA, 2017, pp. 1273–1282.

[25] S. X. Zhang, A. E. Choromanska, and Y. LeCun, Deep learning with elastic averaging SGD, in *Proc. 28th Int. Conf. Neural Information Processing Systems*, Cambridge, MA, USA, 2015, pp. 685–693.

[26] T. Y. Chen, G. Giannakis, T. Sun, and W. T. Yin, Lag: Lazily aggregated gradient for communication-efficient distributed learning, in *Proc. 32nd Int. Conf. Neural Information Processing Systems*, Red Hook, NY, USA, 2018, pp. 5050–5065.

[27] J. Y. Wang and G. Joshi, Cooperative SGD: A unified framework for the design and analysis of communication-efficient SGD algorithms, arXiv preprint arXiv: 1808.07576, 2018.

[28] T. Y. Chen, Y. J. Sun, and W. T. Yin, LASG: Lazily aggregated stochastic gradients for communication-efficient distributed learning, arXiv preprint arXiv: 2002.11360, 2020.

[29] Y. Dong, J. Chen, Y. H. Tang, J. J. Wu, H. Q. Wang, and E. Q. Zhou, Lazy scheduling based disk energy optimization method, *Tsinghua Science and Technology*, vol. 25, no. 2, pp. 203–216, 2020.

[30] Y. Arjevani and O. Shamir, Communication complexity of distributed convex learning and optimization, in *Proc. 28th Int. Conf. Neural Information Processing Systems*, Cambridge, MA, USA, 2015, pp. 1756–1764.

[31] 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.

[32] D. Davis and W. T. Yin, Convergence rate analysis of several splitting schemes, in *Splitting Methods in Communication, Imaging, Science, and Engineering*, R. Glowinski, S. Osher, and W. Yin, eds. Cham, Germany: Springer, 2016, pp. 115–163.

Xiaoge Deng received the BS degree in mathematics from the University of Science and Technology of China (USTC), Hefei, China in 2018. He is currently pursuing the MS degree at the School of Computer, National University of Defense Technology (NUDT). His research interests include optimization for machine learning and distributed systems.

Dongsheng Li received the BSc (Hons.) and PhD (Hons.) degrees in computer science from the National University of Defense Technology, Changsha, China in 1999 and 2005, respectively. He is currently a full professor at the National Laboratory for Parallel and Distributed Processing, National University of Defense Technology. His research interests include parallel and distributed computing, cloud computing, and large-scale data management. He was awarded the prize of the National Excellent Doctoral Dissertation of China by the Ministry of Education of China in 2008.

Tao Sun received the BS, MS, and PhD degrees in mathematics from the National University of Defense Technology, Changsha, China in 2012, 2015, and 2018, respectively. He is currently an assistant professor at the National Laboratory for Parallel and Distributed Processing, National University of Defense Technology. His research interests include optimization for machine learning, image processing, distributed systems, and neural networks.

Feng Liu received the BS, MS, and PhD degrees in computer science and technology from National University of Defense Technology, Changsha, China in 1999, 2002, and 2006, respectively. He is currently an associate research fellow at the National Laboratory for Parallel and Distributed Processing, National University of Defense Technology, China. His research interests include distributed computing and big data.