Received September 30, 2020, accepted October 12, 2020, date of publication October 21, 2020, date of current version November 2, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3032637

Distributed SignSGD With Improved Accuracy and Network-Fault Tolerance

LE TRIEU PHONG[1] **AND TRAN THI PHUONG**[2]
[1]National Institute of Information and Communications Technology (NICT), Tokyo 184-8795, Japan
[2]Faculty of Mathematics and Statistics, Ton Duc Thang University, Ho Chi Minh City, Vietnam

Corresponding author: Tran Thi Phuong (tranthiphuong@tdtu.edu.vn)

The work of Le Trieu Phong was supported in part by JST CREST under Grant JPMJCR19F6.

ABSTRACT This paper proposes DROPSIGNSGD, a communication-efficient and network-fault tolerant algorithm for training deep neural networks in a distributed and synchronous fashion. In DROPSIGNSGD, all numerical elements communicated between machines are either 1 or -1, represented by only one bit. More importantly, DROPSIGNSGD does not decline the benchmark accuracy on the ImageNet dataset when compared with the traditional distributed stochastic gradient descent algorithm, owing to a little trick in memorizing unused gradients. Experimental results are supported by a mathematical proof showing that DROPSIGNSGD converges under standard assumptions.

INDEX TERMS Network-fault tolerance, communication efficiency, distributed SGD, deep learning.

I. INTRODUCTION

A. BACKGROUND

In recent years, deep learning has produced excellent utilities in many practical applications. Along with the development of the field, it has also been realized that the scale of training data and neural network parameters can significantly enhance final leaning result.

Distributed stochastic gradient descent (SGD) is a key algorithm in deep learning. In distributed SGD, many distributed workers each possessing a local dataset constantly communicate with a central parameter server, thereby enabling the shared neural network model at the workers to learn from all local datasets.

The scale of training data and neural network parameters raises two system-level concerns regarding the original distributed SGD algorithm: communication efficiency and network-fault tolerance, as briefly illustrated in Figure 1. Many studies have observed that the communication between any worker and the central server can become a bottleneck in the entire system [1]–[7]. Moreover, the network link between any worker and the server may suffer from unexpected, possibly adversarial faults [8], [9].

As an attempt to simultaneously address both communication efficiency and network-fault tolerance, Bernstein *et al.* [11] have examined an algorithm called signSGD with majority vote, in which only the gradient signs are transmitted from the workers, and the signs of the gradient aggregate are sent from the central parameter server. Such an aggressive quantization of gradients for communication and fault tolerance is elegant, but (unfortunately) reduces the learning accuracy. Indeed, on the ImageNet dataset, the signSGD with majority vote algorithm in [11] suffers from an accuracy decline of approximately 4% compared with the baseline result of distributed SGD (see Table 1).

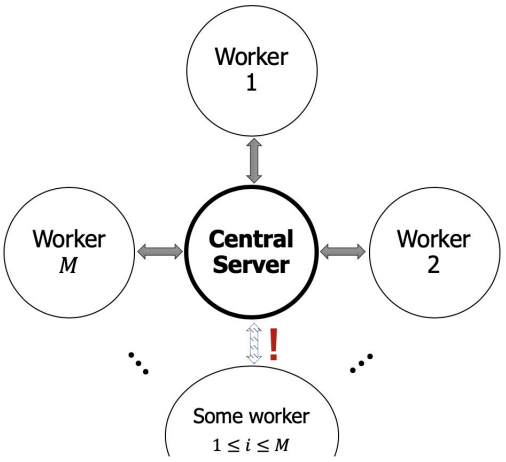

FIGURE 1. Synchronous distributed computation model with unexpectedly slow or faulty communication.

The associate editor coordinating the review of this manuscript and approving it for publication was Qingli Li.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

TABLE 1. Comparison with variants of distributed SGD.

Paper	Worker + Server Communication Cost	Network-Fault Tolerance?	ImageNet Top-1 Accuracy (%)
Baseline (distributed SGD)	$32d + 32d$	no	76.27
Zheng et al. [4]	$(d + 32) + (d + 32)$	no	76.77
Phuong-Phong [10]	$(0.3d + 32) + (d + 32)$	no	76.38
Distributed signSGD with major. vote [11]	$d + d$	yes	72.77
This paper (DROPSIGNSGD)	$0.3d + 0.3d$	yes	76.09
	$0.3d + 0.5d$	yes	76.64
	$0.3d + 0.7d$	yes	76.85

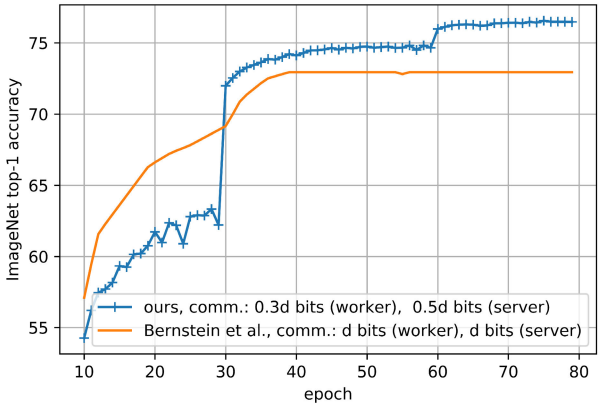

FIGURE 2. While both are robust against network-faults, our DROPSIGNSGD has better top-1 accuracy than that of Bernstein *et al.* [11] in the end. See also Table 1.

It has been also shown in [12] by counter-examples that the use of gradient signs may result in nonconvergence of the algorithm. In this work, we are interested in overcoming these drawbacks, and simultaneously improving the communication efficiency further.

B. OUR CONTRIBUTIONS

We propose DROPSIGNSGD as a variant of signSGD with majority vote [11]. In addition to inheriting the properties of communication efficiency and network-fault tolerance of the original algorithm in [11], DROPSIGNSGD has the following additional merits outlined in Table 1:

- Both workers and the central server in DROPSIGNSGD are allowed to further reduce their amounts of communication by partially dropping at random the gradient signs. For example, each worker can only send $0.3d$ bits to the server, and then receives $0.5d$ bits from the server in each iteration, where d is the number of parameters of a neural network. From a network latency viewpoint, this is advantageous because the network between any worker and the server may become unexpectedly slow at times, as often observed in real systems [3].
- The top-1 accuracy on the ImageNet dataset of DROPSIGNSGD is even better than the baseline result as seen in Table 1. This overcomes the demerit of the distributed signSGD with majority vote algorithm [11] with respect to top-1 accuracy. Figure 2 depicts the top-1 accuracy graph of both algorithms, which reveals that DROPSIGNSGD is better in the end. Indeed, DROPSIGNSGD can reach a top-1 accuracy of 76.64% (and top-5 accuracy of 92.91%) when the communication cost of worker and server is $0.3d + 0.5d$, whereas signSGD with majority vote [11] reaches an inferior top-1 accuracy of 72.77%. When the communication bits from the server vary to other values of $0.3d$ and $0.7d$, the corresponding top-1 (resp., top-5) accuracies change, as expected, to 76.09% (resp., 92.71%) and 76.85% (resp., 93.01%). It is worth noting that, further increasing the communication rates does not necessarily yield better accuracy results.

Technically, DROPSIGNSGD uses a method previously exploited in [4], [10], [12], remembering the unused gradient magnitudes and adding those to the subsequent training iteration. Nonetheless, because network-fault tolerance is one of our design goals, which is not exhibited by the algorithms in [4], [10], [12], care must be taken to maintain the network-fault tolerance property. A little trick that we introduce is to use an error-learning rate to update the local errors caused by unused gradient magnitudes, as detailed in a subsequent section.

The paper is organized as follows. The proposed DROPSIGNSGD is fully described in Algorithm 1. The convergence of the algorithm is ensured by Theorem 1. Finally, Section IV presents experimental results on the ImageNet dataset.

C. RELATED WORKS

Regarding communication, the standard distributed SGD algorithm is inefficient when compared with its subsequent variants, as illustrated in Table 1. Moreover, it is not network-fault-tolerant: if a gradient vector is rescaled by some large factor over the faulty network when transmitted from a worker to the server, the gradient average on the server is severely affected by the factor. Therefore, the parameter update becomes faulty to the extent of divergence.

Network faults can be handled by Byzantine fault-tolerance as in [13]–[15], but these do not consider communication efficiency as a design goal. In addition, signSGD with majority vote [11] is more dedicated to network faults, with relatively graceful tolerances as examined in [11].

Variants of distributed SGD having communication efficiency for workers were proposed in [4]–[6], [16]–[19], but without examining network-fault tolerance. The communication from the server to workers in [6], [17] is not compressed, and hence identical to traditional SGD.

Techniques dealing with slow (while not necessarily faulty) workers have been extensively reported in the literature. Replication-based techniques as in [20]–[24] make use of cloned workers or repeated communication; in contrast to our proposed system. Asynchronous optimization as in [1], [2], [25], [26] can handle slow workers effectively; nevertheless synchronous distributed SGD exhibits better accuracy as shown in [27], [28]. Code-based techniques as in [7], [29]–[32] can also be used when redundancy required in the codes can be satisfied.

Techniques for handling errors in neural networks as surveyed in [33] can be locally used in each workers. These techniques are complementary to this paper, because they deal with errors inside but not outside the workers.

II. MATHEMATICAL ASSUMPTIONS FOR CONVERGENCE

Associated to a non-convex loss function $\ell : \mathbb{R}^d \times \Xi \to \mathbb{R}^+$, consider $f(x) = \mathbf{E}_\xi[\ell(x,\xi)]$, in which $x \in \mathbb{R}^d$ is the neural-network weight parameters, and $\xi \in \Xi$ is the data in computation. The following assumptions are standard and have been used in previous works [4], [10]. Below $\|\cdot\|$ denotes the Euclidean norm of a vector, and $\langle\cdot,\cdot\rangle$ the inner product.

Assumption 1: We have $f^\star = \inf_{x\in\mathbb{R}^d} f(x) < \infty$. In addition, f is L-smooth, namely f is differentiable, and for some $L \geq 0$

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x-y\|, \quad \forall x, y \in \mathbb{R}^d, \tag{1}$$

which implies the following

$$f(x) \leq f(y) + \langle \nabla f(y), x-y\rangle + \frac{L}{2}\|x-y\|^2. \tag{2}$$

Assumption 2: Let $\mathbf{E}_t$ denote the expectation at iteration t. Then $\mathbf{E}_t[g_{t,i}] = \nabla f(x_t)$ and $\exists\sigma$, $\mathbf{E}_t[\|g_{t,i} - \nabla f(x_t)\|^2] \leq \sigma^2$.

Assumption 3: There is a constant ω such that $\|\nabla f(x_t)\|^2 \leq \omega^2$.

Assumption 4: The vectors $\{g_{t,i} - \nabla f(x_t)\}_{1\leq i\leq M}$ are independently random.

Below are some useful and direct derivations from the assumptions used later in the mathematical proof of convergence. Because $\mathbf{E}_t[\|g_{t,i} - \nabla f(x_t)\|^2] \leq \sigma^2$ and $\|\nabla f(x_t)\|^2 \leq \omega^2$, we obtain

$$\mathbf{E}_t[\|g_{t,i}\|^2] \leq G^2 = \sigma^2 + \omega^2. \tag{3}$$

which implies

$$\mathbf{E}_t\left[\left\|\frac{1}{M}\sum_{i=1}^M g_{t,i}\right\|^2\right] \leq G^2. \tag{4}$$

By Assumption 4, we have

$$\mathbf{E}_t\left[\left\|\sum_{i=1}^M (g_{t,i} - \nabla f(x_t))\right\|^2\right] = \sum_{i=1}^M \mathbf{E}_t\left[\|g_{t,i} - \nabla f(x_t)\|^2\right] \leq M\sigma^2.$$

Therefore

$$\mathbf{E}_t\left[\left\|\frac{1}{M}\sum_{i=1}^M (g_{t,i} - \nabla f(x_t))\right\|^2\right] \leq \frac{\sigma^2}{M}. \tag{5}$$

III. ALGORITHM 1 AND ITS MATHEMATICAL CONVERGENCE

The design of Algorithm 1 follows those in [4], [10], but the communication between any worker and the server is gradient signs as in [11]. To make this combination not decreasing the top-1 accuracy on the ImageNet dataset, we introduce a little trick in lines 12 and 17 of Algorithm 1: the errors in each iteration are updated by error-learning rates c_t (at worker) and $\tilde{c}_t$ (at server). In the experiments, we simply select $c_t = \tilde{c}_t = O(10^{-3})$ as a small constant. These errors are bounded as in Lemma 1.

To realize the randomized method $\mathsf{sign}_{\beta_{t,i}}(p)$ at line 8 where each sign is kept as-is with probability $\beta_{t,i}$, it suffices for the workers and the server to agree on a method $\mathsf{maskgen}(t,i)$ generating a vector $\mathsf{mask}^{(t,i)} = (\mathsf{mask}_1^{(t,i)}, \ldots, \mathsf{mask}_d^{(t,i)}) \in \{0,1\}^d$ in which each component is 1 with probability $\beta_{t,i}$ and 0 with probability $1-\beta_{t,i}$. Each worker i, at iteration t, only transmits the gradient signs corresponding to the component of value 1 in the mask vector. For line 16, a similar method $\mathsf{maskgenServer}(t)$ is agreed between the server and the workers.

Lemma 1 (Error Bound for Lines 12 and 17 of Algorithm 1): Let p be a vector in $\mathbb{R}^d$. For some sufficiently small $c > 0$, there exists $0 < \delta_\beta < 1$ such that

$$\|c\cdot\mathsf{sign}_\beta(p) - p\|^2 \leq (1-\delta_\beta)\|p\|^2 \tag{6}$$

Proof: Expanding the left-hand side as follows:

$$\begin{aligned}
&\|c\cdot\mathsf{sign}_\beta(p) - p\|^2 \\
&= \langle c\cdot\mathsf{sign}_\beta(p) - p, c\cdot\mathsf{sign}_\beta(p) - p\rangle \\
&= \langle c\cdot\mathsf{sign}_\beta(p), c\cdot\mathsf{sign}_\beta(p)\rangle - 2c\langle\mathsf{sign}_\beta(p), p\rangle + \langle p, p\rangle \\
&= c^2\|\mathsf{sign}_\beta(p)\|^2 - 2c\langle\mathsf{sign}_\beta(p), p\rangle + \|p\|^2.
\end{aligned}$$

To ensure (6), it suffices to have

$$c^2\|\mathsf{sign}_\beta(p)\|^2 - 2c\langle\mathsf{sign}_\beta(p), p\rangle + \|p\|^2 \leq (1-\delta_\beta)\|p\|^2$$

which is equivalent to the following

$$\delta_\beta \leq \frac{2c\langle\mathsf{sign}_\beta(p), p\rangle - c^2\|\mathsf{sign}_\beta(p)\|^2}{\|p\|^2}. \tag{7}$$

To allow $\delta_\beta > 0$ in (7), it is necessary that

$$2c\langle\mathsf{sign}_\beta(p), p\rangle - c^2\|\mathsf{sign}_\beta(p)\|^2 > 0,$$

or equivalently,

$$\frac{2\langle\mathsf{sign}_\beta(p), p\rangle}{\|\mathsf{sign}_\beta(p)\|^2} > c \tag{8}$$

owing to the fact that $c > 0$. Note that, with vectors $p = (p_1, \ldots, p_d) \in \mathbb{R}^d$, $\mathsf{sign}_\beta(p) = (s_1, \ldots, s_d) \in \{-1, 0, 1\}^d$,

Algorithm 1 Distributed signSGD With Sign Dropouts (DROPSIGNSGD)

Input: Neural-net loss function ℓ, sequences $\{\eta_t\}$, $\{c_t\}$, $\{\tilde{c}_t\}$, momentum $0 \le \mu < 1$, keep-or-drop parameters $\beta_{t,i}$, $\tilde{\beta}_t$
Initialize: $x_0 \in \mathbb{R}^d$; $m_{-1,i} = e_{0,i} = 0 \in \mathbb{R}^d$; $\tilde{e}_0 = 0 \in \mathbb{R}^d$ initially
for $t \in \{0, \ldots, T-1\}$ **do**
  • **on each worker** i $(1 \le i \le M)$**:**
    Select data $\xi_{t,i}$ and compute $g_{t,i} = \nabla \ell(x_t, \xi_{t,i})$ ▷ stochastic gradient
    $m_{t,i} = \mu m_{t-1,i} + g_{t,i}$ ▷ stochastic momentum
    $p_{t,i} = \mu m_{t,i} + g_{t,i} + e_{t,i}$ ▷ gradient with Nesterov momentum plus the error from the previous iteration
    $\Delta_{t,i} = \mathsf{sign}_{\beta_{t,i}}(p_{t,i})$ ▷ the gradient signs, kept as-is (i.e., ± 1) with prob. $\beta_{t,i}$, and 0 with prob. $1 - \beta_{t,i}$
    push $\Delta_{t,i}$ to server
    pull $\tilde{\Delta}_t$ from server
    $x_{t+1} = x_t - \eta_t \tilde{\Delta}_t$ ▷ update the neural network weight
    $e_{t+1,i} = p_{t,i} - c_t \Delta_{t,i}$ ▷ update the local error, with error-learning rate c_t
  • **on central parameter server:**
    receive $\Delta_{t,i}$ for all $1 \le i \le M$
    compute $\tilde{p}_t = \frac{1}{M} \sum_{i=1}^{M} c_t \Delta_{t,i} + \tilde{e}_t$ ▷ average all workers $\Delta_{t,i}$, and adding possible server error
    push $\tilde{\Delta}_t = \mathsf{sign}_{\tilde{\beta}_t}(\tilde{p}_t)$ to each worker ▷ each component is ± 1 with prob. $\tilde{\beta}_t$, and 0 with prob. $1 - \tilde{\beta}_t$
    $\tilde{e}_{t+1} = \tilde{p}_t - \tilde{c}_t \tilde{\Delta}_t$ ▷ update the server error, with error-learning rate $\tilde{c}_t$
end for

and $\mathcal{I} = \{i : s_i \neq 0\}$, we have

$$\langle \mathsf{sign}_\beta(p), p \rangle = \sum_{i \in \mathcal{I}} |p_i| > 0$$

and thus the left-hand side of (8) is a positive number. Therefore, there always exists a small constant c satisfying (8), finishing the proof. ■

In the experiments, we select c as small as $O(10^{-3})$ by the following intuition from formula (8). The left-hand side of formula (8) can be written as, using the same notations as above,

$$\frac{2 \sum_{i \in \mathcal{I}} |p_i|}{|\mathcal{I}|},$$

which is proportional to the average of gradient norm. Therefore, if the average is larger than the small constant c (e.g., $c = O(10^{-3})$), Lemma 1 holds true; and hence by the subsequent Theorem 1, Algorithm 1 continues to converge, which is desirable. Reversely, when the gradient average becomes smaller than the constant c, Algorithm 1 potentially reaches a stationary point which can be a minimum. As a concrete example, we plot in Figure 3 the graphs of accuracies when varying the error-learning rates $c_t = \tilde{c}_t = c$, using the ImageNet dataset and the ResNet-50 model for a few epochs. The top-1 accuracies with $c = c' \times 10^{-3}$ with small $c' \in \{2, 4, 6, 8, 10\}$ are close, and relatively more stable than the one with larger $c' = 100$. We exploit this observation in subsequent experiments in Section IV.

Lemma 2 (Total Error Bound): In Algorithm 1, let $\eta_t = \eta > 0$, *and* $\delta = \min\{\delta_{\beta_{t,i}}\}_{t,i}$ *and* $\tilde{\delta} = \min\{\delta_{\tilde{\beta}_t}\}$. *There exists a value* U *depending on* $\delta, \tilde{\delta}$ *such that*

$$\left\| \tilde{e}_t + \frac{1}{M} \sum_{i=1}^{M} e_{t,i} \right\|^2 \le \frac{G^2 U}{(1-\mu)^2}.$$

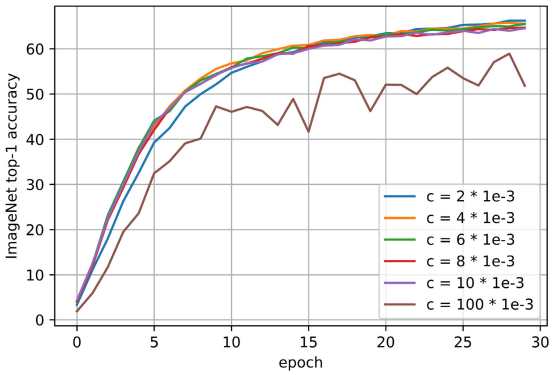

FIGURE 3. Searching for a suitable error-learning rate.

Proof: Given in the Appendix. ■

The mathematical convergence assurance of DROPSIGNSGD is in Theorem 1 in which Lemma 1 and Lemma 2 play an important role in estimating the errors incurred by not sending the real gradients but only their signs. Other lemmas play a supporting role, breaking the complexity of the proof into small parts to ease the presentation.

Theorem 1 (Convergence of DROPSIGNSGD): Suppose that Assumptions 1-4 hold. Let $\eta_t = \eta > 0$, $c_t = \tilde{c}_t = c > 0 \, \forall t \ge 0$, *then there exists a learning rate* η *such that*

$$\min_t \mathbf{E}[\|\nabla f(x_t)\|^2] \le O\left(\frac{1}{\sqrt{MT}}\right).$$

Proof of Theorem 1: We consider the following iterate as in Lemma 8

$$z_t = \tilde{x}_t - \frac{\eta \mu^2}{c(1-\mu)} \frac{1}{M} \sum_{i=1}^{M} m_{t-1,i},$$

where

$$\tilde{x}_t = x_t - \frac{\eta}{c}\left(\tilde{e}_t + \frac{1}{M}\sum_{i=1}^{M} e_{t,i}\right).$$

Using (2), under the smoothness assumption,

$$\begin{aligned}
&\mathbf{E}_t[f(z_{t+1})] \\
&\le f(z_t) + \langle \nabla f(z_t), \mathbf{E}_t[z_{t+1} - z_t]\rangle + \frac{L}{2}\mathbf{E}_t[\|z_{t+1} - z_t\|^2] \\
&= f(z_t) - \frac{\eta}{c(1-\mu)}\left\langle \nabla f(z_t), \mathbf{E}_t\left[\frac{1}{M}\sum_{i=1}^{M} g_{t,i}\right]\right\rangle \\
&\quad + \frac{L\eta^2}{2c^2(1-\mu)^2}\mathbf{E}_t\left[\left\|\frac{1}{M}\sum_{i=1}^{M} g_{t,i}\right\|^2\right] \qquad (9) \\
&= f(z_t) - \frac{\eta}{c(1-\mu)}\langle \nabla f(z_t), \nabla f(x_t)\rangle \\
&\quad + \frac{L\eta^2}{2c^2(1-\mu)^2}\mathbf{E}_t\left[\left\|\frac{1}{M}\sum_{i=1}^{M} g_{t,i}\right\|^2\right], \qquad (10)
\end{aligned}$$

where (9) is by Lemma 8 and (10) is by Assumption 2. In addition, given

$$\begin{aligned}
\mathbf{E}_t\left[\left\|\frac{1}{M}\sum_{i=1}^{M} g_{t,i}\right\|^2\right] &= \mathbf{E}_t\left[\|\nabla f(x_t)\|^2\right] \\
&\quad + \mathbf{E}_t\left[\left\|\frac{1}{M}\sum_{i=1}^{M} g_{t,i} - \nabla f(x_t)\right\|^2\right],
\end{aligned}$$

combining with (10), we have

$$\begin{aligned}
\mathbf{E}_t[f(z_{t+1})] &\le f(z_t) - \frac{\eta}{c(1-\mu)}\langle \nabla f(z_t), \nabla f(x_t)\rangle \\
&\quad + \frac{L\eta^2}{2c^2(1-\mu)^2}\mathbf{E}_t\|\nabla f(x_t)\|^2 \\
&\quad + \frac{L\eta^2}{2c^2(1-\mu)^2}\mathbf{E}_t\left[\left\|\frac{1}{M}\sum_{i=1}^{M} g_{t,i} - \nabla f(x_t)\right\|^2\right].
\end{aligned}$$

By (5), we obtain

$$\begin{aligned}
\mathbf{E}_t[f(z_{t+1})] &\le f(z_t) - \frac{\eta}{c(1-\mu)}\langle \nabla f(z_t), \nabla f(x_t)\rangle \\
&\quad + \frac{L\eta^2}{2c^2(1-\mu)^2}\mathbf{E}_t\|\nabla f(x_t)\|^2 + \frac{L\eta^2\sigma^2}{2c^2(1-\mu)^2 M}. \qquad (11)
\end{aligned}$$

In addition, we have

$$\begin{aligned}
&-\langle \nabla f(z_t), \nabla f(x_t)\rangle \\
&= \langle \nabla f(x_t) - \nabla f(z_t), \nabla f(x_t)\rangle - \langle \nabla f(x_t), \nabla f(x_t)\rangle \\
&= \langle \nabla f(x_t) - \nabla f(z_t), \nabla f(x_t)\rangle - \|\nabla f(x_t)\|^2 \\
&\le \frac{1}{2}\|\nabla f(x_t)\|^2 + \frac{1}{2}\|\nabla f(x_t) - \nabla f(z_t)\|^2 \\
&\quad - \|\nabla f(x_t)\|^2 \\
&= -\left(1 - \frac{1}{2}\right)\|\nabla f(x_t)\|^2 + \frac{1}{2}\|\nabla f(x_t) - \nabla f(z_t)\|^2 \\
&\le -\frac{1}{2}\|\nabla f(x_t)\|^2 + \frac{L^2}{2}\|x_t - z_t\|^2, \qquad (12)
\end{aligned}$$

where the last inequality is by Assumption 1. Using Lemma 7 and Lemma 8, we obtain

$$\begin{aligned}
&\|x_t - z_t\|^2 \\
&\le 2\|x_t - \tilde{x}_t\|^2 + 2\|\tilde{x}_t - z_t\|^2 \\
&= 2\frac{\eta^2}{c^2}\left\|\tilde{e}_t + \frac{1}{M}\sum_{i=1}^{M} e_{t,i}\right\|^2 + \frac{2\eta^2\mu^4}{c^2(1-\mu)^2}\left\|\frac{1}{M}\sum_{i=1}^{M} m_{t-1,i}\right\|^2.
\end{aligned}$$

Applying Lemma 6 and Lemma 2, we get

$$\begin{aligned}
\|x_t - z_t\|^2 &\le \frac{2\eta^2 G^2 U}{c^2(1-\mu)^2} \\
&\quad + \frac{2\eta^2\mu^4}{c^2(1-\mu)^3}\sum_{k=0}^{t-1}\mu^{t-1-k}\left\|\frac{1}{M}\sum_{i=1}^{M} g_{k,i}\right\|^2. \qquad (13)
\end{aligned}$$

Substituting (13) into (12) gives us

$$\begin{aligned}
&-\langle \nabla f(z_t), \nabla f(x_t)\rangle \\
&\le -\frac{1}{2}\|\nabla f(x_t)\|^2 + \frac{\eta^2\, G^2\, L^2\, U}{c^2(1-\mu)^2} \\
&\quad + \frac{\eta^2\mu^4 L^2}{c^2(1-\mu)^3}\sum_{k=0}^{t-1}\mu^{t-1-k}\left\|\frac{1}{M}\sum_{i=1}^{M} g_{k,i}\right\|^2.
\end{aligned}$$

Therefore, by (11), we have

$$\begin{aligned}
&\mathbf{E}_t[f(z_{t+1})] \\
&\le f(z_t) - \left(\frac{\eta}{2c(1-\mu)} - \frac{L\eta^2}{2c^2(1-\mu)^2}\right)\mathbf{E}_t\|\nabla f(x_t)\|^2 \\
&\quad + \frac{L\eta^2\sigma^2}{2c^2(1-\mu)^2 M} + \frac{\eta^3\, G^2\, L^2\, U}{c^3(1-\mu)^3} \\
&\quad + \frac{\eta^3\mu^4 L^2}{c^3(1-\mu)^4}\sum_{k=0}^{t-1}\mu^{t-1-k}\left\|\frac{1}{M}\sum_{i=1}^{M} g_{k,i}\right\|^2.
\end{aligned}$$

Rearranging the terms, taking total expectation give us

$$\begin{aligned}
&\left(\frac{\eta}{2c(1-\mu)} - \frac{L\eta^2}{2c^2(1-\mu)^2}\right)\mathbf{E}[\|\nabla f(x_t)\|^2] \\
&\le \mathbf{E}[f(z_t) - f(z_{t+1})] + \frac{L\eta^2\sigma^2}{2c^2(1-\mu)^2 M} + \frac{\eta^3\, G^2\, L^2\, U}{c^3(1-\mu)^3} \\
&\quad + \frac{\eta^3\mu^4 L^2}{c^3(1-\mu)^4}\sum_{k=0}^{t-1}\mu^{t-1-k}\mathbf{E}\left[\left\|\frac{1}{M}\sum_{i=1}^{M} g_{k,i}\right\|^2\right].
\end{aligned}$$

Because

$$\begin{aligned}
\sum_{t=0}^{T-1}\mathbf{E}[f(z_t) - f(z_{t+1})] &= f(z_0) - f(z_T) \\
&= f(x_0) - f(z_T) \le f(x_0) - f^\star,
\end{aligned}$$

we have

$$\begin{aligned}
&\left(\frac{\eta}{2c(1-\mu)} - \frac{L\eta^2}{2c^2(1-\mu)^2}\right)\sum_{t=0}^{T-1}\mathbf{E}[\|\nabla f(x_t)\|^2] \\
&\le f(x_0) - f^\star + \frac{L\eta^2\sigma^2 T}{2c^2(1-\mu)^2 M} + \frac{\eta^3\, G^2\, L^2\, UT}{c^3(1-\mu)^3} \\
&\quad + \frac{\eta^3\mu^4 L^2}{c^3(1-\mu)^4}\sum_{t=0}^{T-1}\sum_{k=0}^{t-1}\mu^{t-1-k}\mathbf{E}\left[\left\|\frac{1}{M}\sum_{i=1}^{M} g_{k,i}\right\|^2\right].
\end{aligned}$$

$$= f(x_0) - f^\star + \frac{L\eta^2\sigma^2 T}{2c^2(1-\mu)^2 M} + \frac{\eta^3\, G^2\, L^2\, UT}{c^3(1-\mu)^3} + \frac{\eta^3\mu^4 L^2}{c^3(1-\mu)^4}\sum_{t=0}^{T-1}\sum_{k=0}^{t-1}\mu^{t-1-k}\mathbf{E}\left[\|\nabla f(x_k)\|^2\right] + \frac{\eta^3\mu^4 L^2}{c^3(1-\mu)^4}\sum_{t=0}^{T-1}\sum_{k=0}^{t-1}\mu^{t-1-k}\mathbf{E}\left\|\frac{1}{M}\sum_{i=1}^{M} g_{k,i} - \nabla f(x_k)\right\|^2.$$

By (5), we obtain

$$\sum_{k=0}^{t-1}\mu^{t-1-k}\mathbf{E}\left[\left\|\frac{1}{M}\sum_{i=1}^{M} g_{k,i} - \nabla f(x_k)\right\|^2\right] \le \frac{\sigma^2}{M}\sum_{k=0}^{t-1}\mu^{t-1-k} \le \frac{\sigma^2}{M(1-\mu)}.$$

Moreover, we have

$$\begin{aligned}
&\sum_{t=0}^{T-1}\sum_{k=0}^{t-1}\mu^{t-1-k}\mathbf{E}\left[\|\nabla f(x_k)\|^2\right] \\
&= \sum_{k=0}^{T-2}\sum_{t=k+1}^{T-1}\mu^{t-1-k}\mathbf{E}\left[\|\nabla f(x_k)\|^2\right] \\
&= \sum_{k=0}^{T-2}\mathbf{E}\left[\|\nabla f(x_k)\|^2\right]\sum_{t=k+1}^{T}\mu^{t-1-k} \\
&\le \frac{1}{1-\mu}\sum_{k=0}^{T-2}\mathbf{E}\left[\|\nabla f(x_k)\|^2\right] \\
&\le \frac{1}{1-\mu}\sum_{k=0}^{T-1}\mathbf{E}\left[\|\nabla f(x_k)\|^2\right].
\end{aligned}$$

Therefore

$$\left(\frac{\eta}{2c(1-\mu)} - \frac{L\eta^2}{2c^2(1-\mu)^2}\right)\sum_{t=0}^{T-1}\mathbf{E}[\|\nabla f(x_t)\|^2] \le f(x_0) - f^\star + \frac{L\eta^2\sigma^2 T}{2c^2(1-\mu)^2 M} + \frac{\eta^3\, G^2\, L^2\, UT}{c^3(1-\mu)^3} + \frac{\eta^3\mu^4 L^2}{c^3(1-\mu)^5}\sum_{k=0}^{T-1}\mathbf{E}\left[\|\nabla f(x_k)\|^2\right] + \frac{\eta^3\mu^4 L^2\sigma^2\, T}{c^3(1-\mu)^5 M}.$$

Let

$$V = \frac{\eta}{2c(1-\mu)} - \frac{L\eta^2}{2c^2(1-\mu)^2} - \frac{\eta^3\mu^4 L^2}{c^3(1-\mu)^5},$$

we obtain

$$V\sum_{t=0}^{T-1}\mathbf{E}[\|\nabla f(x_t)\|^2] \le f(x_0) - f^\star + \frac{L\eta^2\sigma^2 T}{2c^2(1-\mu)^2 M} + \frac{\eta^3\, G^2\, L^2\, UT}{c^3(1-\mu)^3} + \frac{\eta^3\mu^4 L^2\sigma^2 T}{c^3(1-\mu)^5 M}.$$

Let $\eta \le \frac{c(1-\mu)^2}{2L}$. Then

$$\begin{aligned}
V &= \frac{\eta}{2c(1-\mu)} - \frac{L\eta^2}{2c^2(1-\mu)^2} - \frac{\eta^3\mu^4 L^2}{c^3(1-\mu)^5} \\
&= \frac{\eta}{2c(1-\mu)}\left(1 - \frac{L\eta}{c(1-\mu)} - \frac{2\eta^2\mu^4 L^2}{c^2(1-\mu)^4}\right) \\
&\ge \frac{\eta}{2c(1-\mu)}\left(1 - \frac{1-\mu}{2} - \frac{\mu^4}{2}\right) \\
&\ge \frac{\eta}{4c(1-\mu)}.
\end{aligned}$$

Therefore

$$\frac{\eta}{4c(1-\mu)}\sum_{t=0}^{T-1}\mathbf{E}[\|\nabla f(x_t)\|^2] \le f(x_0) - f^\star + \frac{L\eta^2\sigma^2 T}{2c^2(1-\mu)^2 M} + \frac{\eta^3\, G^2\, L^2\, UT}{c^3(1-\mu)^3} + \frac{\eta^3\mu^4 L^2\sigma^2 T}{c^3(1-\mu)^5 M}.$$

Multiplying both sides with $\frac{4c(1-\mu)}{\eta T}$, we get

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbf{E}[\|\nabla f(x_t)\|^2] \le \frac{4c(1-\mu)(f(x_0)-f^\star)}{\eta T} + \frac{2L\eta\sigma^2}{c(1-\mu)M} + \frac{4\eta^2\mu^4 L^2\sigma^2}{c^2(1-\mu)^4 M} + \frac{4\eta^2 G^2\, L^2\, U}{c^2(1-\mu)^2},$$

which implies

$$\min_t \mathbf{E}[\|\nabla f(x_t)\|^2] \le \frac{4c(1-\mu)(f(x_0)-f^\star)}{\eta T} + \frac{2L\eta\sigma^2}{c(1-\mu)M} + \frac{4\eta^2\mu^4 L^2\sigma^2}{c^2(1-\mu)^4 M} + \frac{4\eta^2 G^2\, L^2\, U}{c^2(1-\mu)^2},$$

and thereby the theorem statement is obtained by simply selecting $\eta = \sqrt{M}/\sqrt{T}$. ■

IV. EXPERIMENTS

We conduct experiments using ResNet-50 [34], trained with the large-scale ImageNet dataset [35]. We slightly change the PyTorch codes given in [11], [36] with necessary adaptation to DROPSIGNSGD.

A. TOLERANCE OF RESCALING ADVERSARY

A rescaling adversary captures the network faults in which the vector Δ in communication between any worker and the server is multiplied element-wise with an adversarial vector v of positive components, written as $v > 0$ for short. Given Δ as in lines 8 and 16 of DROPSIGNSGD, it can be seen that the algorithm tolerates this type of adversary almost for free, because $\Delta = \mathsf{sign}(\Delta) = \mathsf{sign}(\Delta \cdot v)$ in which "$\cdot$" is element-wise multiplication. Practically, the server and any worker can detect the faults by inspecting the component values, and then fix them just by taking the sign of the communicated vector if necessary. Experimental results with respect to this type of network fault are given in Table 1 and Figure 2.

TABLE 2. Network-fault (sign inversion) tolerance. The number of network parameters $d = 25,557,032$.

Paper	Communication (in bits)		Number of Sign-Inverting Workers	ImageNet Top-1 Accuracy
	Worker	Server		
Bernstein et al. [11]	d	d	0 (i.e., 0%)	72.77%
	d	d	1 (i.e., 14%)	71.99%
	d	d	2 (i.e., 29%)	66.82%
	d	d	3 (i.e., 43%)	48.94%
This paper (DROPSIGNSGD)	$0.3d$	$0.5d$	0 (i.e., 0%)	76.64%
	$0.3d$	$0.5d$	1 (i.e., 14%)	75.93%
	$0.3d$	$0.5d$	2 (i.e., 29%)	75.01%
	$0.3d$	$0.5d$	3 (i.e., 43%)	70.41%

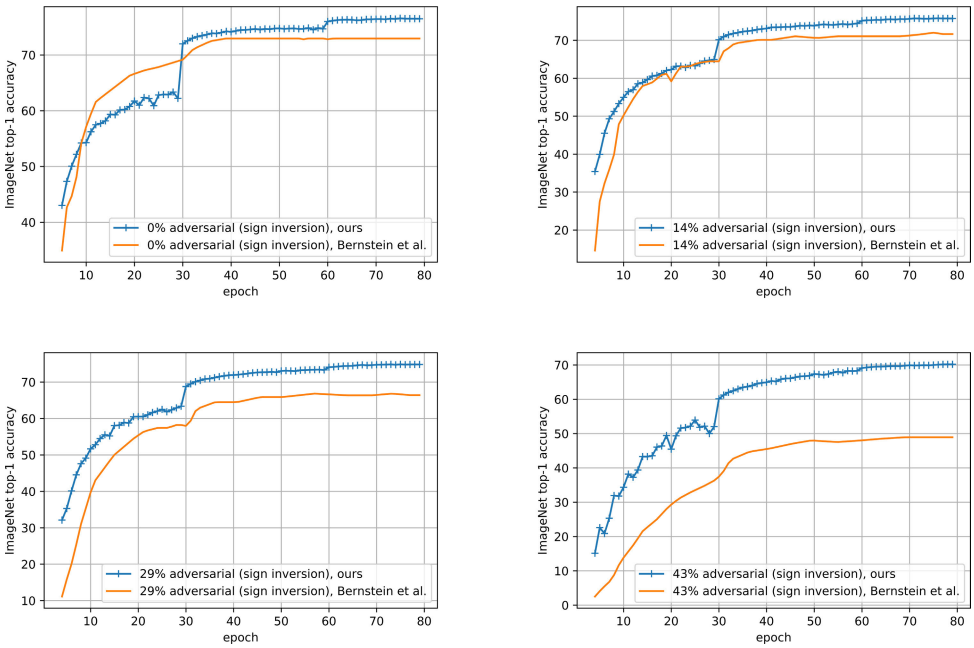

FIGURE 4. Comparisons on ImageNet top-1 testing accuracies with those in Bernstein *et al.* [11], when a fraction (respectively 0%, 14%, 29%, 43%) of workers behave adversarially, inverting their gradient signs.

B. TOLERANCE OF SIGN-INVERTING ADVERSARY

A sign-inverting adversary, as the name suggested, inverts the signs of the vector in communication. That is, the vector Δ before transmission becomes $-\Delta$ after transmission. This is the most devastating adversary considered in [11], intuitively because it forces the learning algorithm to move the weight parameters sharply against the minimum of the loss function. In Table 2 and Figure 4, we provide the experimental results with respect to the tolerance of this adversary type, for both DROPSIGNSGD and signSGD with majority vote [11]. When there is no sign-inverting adversary, DROPSIGNSGD obtains the top-1 accuracy of 76.64% (and top-5 of 92.91%), which is higher than those of signSGD with majority vote [11]. When the number of adversaries is $1, 2, 3$, the top-1 accuracies $(75.93\%, 75.01\%, 70.41\%)$ are given in comparison with those in [11], and additionally the top-5 accuracies of DROPSIGNSGD are $92.65\%, 92.20\%, 89.66\%$, respectively. Among $M = 7$ workers, in each iteration, we consider scenarios in which at most 3 (i.e., 43%) workers have faulty communication. The choice of faulty communication link is random at every iteration, capturing the idea that the network fault is unforeseen. We simply select the error-learning rates $c_t = \tilde{c}_t = 6 \times 10^{-3}$ for all workers and the server. For all iteration t and worker i, the rate $\beta_{t,i}$ is set to 0.3, meaning 30% of gradient signs are transmitted from each worker to the server. The rate $\tilde{\beta}_t$ is set to 0.5, meaning 50% of aggregated gradient signs are transmitted from the server to the workers. These rates make DROPSIGNSGD more communication-efficient than the counterpart in [11], and we intentionally set the rates small to reduce the bad effects of sign inversion. Indeed, the fact that 70% of worker gradients are zeroed makes sign inversion fault less severe as confirmed in the experiments. Putting it all together, DROPSIGNSGD is able to achieve better testing accuracy than the counterpart in [11]. The use of communication rates in DROPSIGNSGD is perhaps similar to the well-known technique of dropouts in deep neural networks: training with fewer and randomly-selected neural nodes (cf., with less and randomly-selected communication in DROPSIGNSGD) may give better results than with all and fixed neural nodes (cf., with all and fixed communication from the server and the workers in DROPSIGNSGD).

V. CONCLUSION

We design and evaluate DROPSIGNSGD as a variant of signSGD with majority vote, with better tolerance of network faults while having less communication for both workers and server. Under standard assumptions on the non-convex loss function, we show that DROPSIGNSGD converges mathematically. In addition, compared with state-of-the-art, DROPSIGNSGD experimentally exhibits superior performance with respect to rescaling and sign-inverting adversaries which models network faults. We believe that communication efficiency and robustness such as network-fault tolerance in distributed systems are important in order to scale the system, and suggest pushing the state-of-the-art to a new stage as a future research direction.

AUXILIARY LEMMAS

Below are necessary lemmas for proving the convergence of DROPSIGNSGD. Some are borrowed from [10] repeated here for completeness and thus without proofs, while the others are dedicated to DROPSIGNSGD.

Lemma 3 (Lemma 1 of [10]): Let $\{a_t\}$ is a non-negative sequence in $\mathbb{R}$ such that $a_0 = 0$ and, for all $t \geq 0$, and non-negative numbers $\alpha, \beta \in \mathbb{R}$ such that $a_{t+1} \leq \alpha a_t + \beta$. Then $a_{t+1} \leq \beta \sum_{j=0}^{t} \alpha^j$.

Lemma 4 (Lemma 3 of [10]): Let $0 < M \in \mathbb{N}$ and $x_i \in \mathbb{R}^d$. Then

$$\left\| \frac{1}{M} \sum_{i=1}^{M} x_i \right\|^2 \leq \frac{1}{M} \sum_{i=1}^{M} \|x_i\|^2 .$$

Lemma 5 (Lemma 4 of [10]): For any $1 \leq i \leq M$,

$$\mathbf{E}[\|\mu m_{t,i} + g_{t,i}\|^2] \leq \frac{G^2}{(1-\mu)^2}.$$

Lemma 6 (Lemma 5 of [10]):

$$\left\| \frac{1}{M} \sum_{i=1}^{M} m_{t-1,i} \right\|^2 \leq \frac{1}{1-\mu} \left(\sum_{k=0}^{t-1} \mu^{t-1-k} \left\| \frac{1}{M} \sum_{i=1}^{M} g_{k,i} \right\|^2 \right).$$

Lemma 7: Let $\eta_t = \eta > 0, c_t = \tilde{c}_t = c > 0$. The error-corected iterate

$$\tilde{x}_t = x_t - \frac{\eta}{c} \left(\tilde{e}_t + \frac{1}{M} \sum_{i=1}^{M} e_{t,i} \right),$$

where $x_t, \tilde{e}_t$ and $e_{t,i}$ are generated from Algorithm 1, satisfies

$$\tilde{x}_{t+1} = \tilde{x}_t - \frac{\eta}{cM} \sum_{i=1}^{M} (\mu m_{t,i} + g_{t,i}).$$

Proof: The following equations are by definition

$$\begin{aligned}
\tilde{x}_{t+1} &= x_{t+1} - \frac{\eta}{c} \left(\tilde{e}_{t+1} + \frac{1}{M} \sum_{i=1}^{M} e_{t+1,i} \right) \\
&= x_t - \eta \cdot \mathsf{sign}_{\tilde{\beta}_t}(\tilde{p}_t) \\
&\quad - \frac{\eta}{c} (\tilde{p}_t - c \cdot \mathsf{sign}_{\tilde{\beta}_t}(\tilde{p}_t)) - \frac{\eta}{c} \frac{1}{M} \sum_{i=1}^{M} e_{t+1,i} \\
&= x_t - \frac{\eta}{c} \left(\frac{c}{M} \sum_{i=1}^{M} \mathsf{sign}_{\beta_{t,i}}(p_{t,i}) + \tilde{e}_t \right) \\
&\quad - \frac{\eta}{c} \frac{1}{M} \sum_{i=1}^{M} e_{t+1,i} \\
&= x_t - \frac{\eta}{c} \frac{1}{M} \sum_{i=1}^{M} (c \cdot \mathsf{sign}_{\beta_{t,i}}(p_{t,i}) + e_{t+1,i}) - \frac{\eta}{c} \tilde{e}_t \\
&= x_t - \frac{\eta}{c} \frac{1}{M} \sum_{i=1}^{M} p_{t,i} - \frac{\eta}{c} \tilde{e}_t \\
&= x_t - \frac{\eta}{c} \tilde{e}_t - \frac{\eta}{c} \frac{1}{M} \sum_{i=1}^{M} \left(\mu m_{t,i} + g_{t,i} + e_{t,i} \right) \\
&= x_t - \frac{\eta}{c} \left(\tilde{e}_t + \frac{1}{M} \sum_{i=1}^{M} e_{t,i} \right) \\
&\quad - \frac{\eta}{c} \frac{1}{M} \sum_{i=1}^{M} (\mu m_{t,i} + g_{t,i}) \\
&= \tilde{x}_t - \frac{\eta}{c} \frac{1}{M} \sum_{i=1}^{M} (\mu m_{t,i} + g_{t,i})
\end{aligned}$$

which ends the proof. ∎

Lemma 8: Let $\eta_t = \eta > 0, c_t = \tilde{c}_t = c > 0$. With the sequence $\{\tilde{x}_t\}$ in Lemma 7, consider the following iterate

$$z_t = \tilde{x}_t - \frac{\eta \mu^2}{c(1-\mu)} \frac{1}{M} \sum_{i=1}^{M} m_{t-1,i}.$$

Then

$$z_{t+1} = z_t - \frac{\eta}{c(1-\mu)} \frac{1}{M} \sum_{i=1}^{M} g_{t,i}.$$

Proof: We have

$$\begin{aligned}
z_{t+1} &= \tilde{x}_{t+1} - \frac{\eta \mu^2}{c(1-\mu)} \frac{1}{M} \sum_{i=1}^{M} m_{t,i} \\
&= \tilde{x}_t - \frac{\eta}{cM} \sum_{i=1}^{M} (\mu m_{t,i} + g_{t,i}) - \frac{\eta \mu^2}{c(1-\mu)} \frac{1}{M} \sum_{i=1}^{M} m_{t,i} \\
&= \tilde{x}_t - \frac{\eta \mu}{c(1-\mu)} \frac{1}{M} \sum_{i=1}^{M} m_{t,i} - \frac{\eta}{cM} \sum_{i=1}^{M} g_{t,i} \\
&= \tilde{x}_t - \frac{\eta \mu^2}{c(1-\mu)} \frac{1}{M} \sum_{i=1}^{M} m_{t-1,i} \\
&\quad - \frac{\eta \mu}{c(1-\mu)} \frac{1}{M} \sum_{i=1}^{M} g_{t,i} - \frac{\eta}{cM} \sum_{i=1}^{M} g_{t,i} \\
&= z_t - \frac{\eta}{c(1-\mu)} \frac{1}{M} \sum_{i=1}^{M} g_{t,i}
\end{aligned}$$

as claimed in the lemma statement. ∎

PROOF OF LEMMA 2

Proof: We have

$$\begin{aligned}\left\|\tilde{e}_{t+1}+\frac{1}{M}\sum_{i=1}^{M}e_{t+1,i}\right\|^2 &\le 2\|\tilde{e}_{t+1}\|^2+2\|\frac{1}{M}\sum_{i=1}^{M}e_{t+1,i}\|^2\\ &\le 2\|\tilde{e}_{t+1}\|^2+\frac{2}{M}\sum_{i=1}^{M}\|e_{t+1,i}\|^2, \qquad (14)\end{aligned}$$

where the first inequality is by the fact that $(a+b)^2 \le 2a^2+2b^2$, $\forall a, b$, and the second inequality is by Lemma 4. We will separately bound the two terms of (14). We have

$$\begin{aligned}&\frac{1}{M}\sum_{i=1}^{M}\|e_{t+1,i}\|^2\\ &= \frac{1}{M}\sum_{i=1}^{M}\|c_t\mathsf{sign}_{\beta_{t,i}}(p_{t,i})-p_{t,i}\|^2 \qquad (15)\\ &\le \frac{1}{M}\sum_{i=1}^{M}(1-\delta_{\beta_{t,i}})\|p_{t,i}\|^2 \qquad (16)\\ &\le \frac{1-\delta}{M}\sum_{i=1}^{M}\|p_{t,i}\|^2 \qquad (17)\\ &= \frac{1-\delta}{M}\sum_{i=1}^{M}\|e_{t,i}+\mu m_{t,i}+g_{t,i}\|^2 \qquad (18)\\ &\le \frac{(1-\delta)(1+\gamma)}{M}\sum_{i=1}^{M}\|e_{t,i}\|^2 \qquad (19)\\ &\quad+\frac{(1-\delta)(1+1/\gamma)}{M}\sum_{i=1}^{M}\|\mu m_{t,i}+g_{t,i}\|^2\\ &\le (1-\delta)(1+\gamma)\left(\frac{1}{M}\sum_{i=1}^{M}\|e_{t,i}\|^2\right)\\ &\quad+(1-\delta)(1+1/\gamma)\frac{G^2}{(1-\mu)^2}, \qquad (20)\end{aligned}$$

where equality (15) and (18) is by the setting of $e_{t+1,i}$ and $p_{t,i}$ in Algorithm 1; (17) is by $\delta=\min\{\delta_{\beta_{t,i}}\}$; (16) is by Lemma 1; (19) is by Young inequality with any $\gamma>0$; and (20) is by Lemma 5. Note that inequality (20) is of the form

$$a_{t+1} \le \alpha a_t+\beta, \qquad (21)$$

where

$$\begin{aligned}a_{t+1} &= \frac{1}{M}\sum_{i=1}^{M}\|e_{t+1,i}\|^2\\ \alpha &= (1-\delta)(1+\gamma)\\ \beta &= (1-\delta)(1+1/\gamma)\frac{G^2}{(1-\mu)^2}.\end{aligned}$$

Applying Lemma 3, we obtain

$$a_{t+1} \le \beta\sum_{j=0}^{t}\alpha^j, \qquad (22)$$

By choosing $\gamma=\frac{\delta}{2(1-\delta)}$, we get $\beta=\frac{(1-\delta)(2-\delta)G^2}{\delta(1-\mu)^2}$ and $\alpha=1-\frac{\delta}{2}$. Since $0<\alpha<1$, we have $\sum_{j=0}^{t}\alpha^j \le \sum_{j\ge 0}\alpha^j=\frac{1}{1-\alpha}$. Therefore (22) becomes

$$\begin{aligned}\frac{1}{M}\sum_{i=1}^{M}\|e_{t+1,i}\|^2 &\le \frac{\beta}{1-\alpha}\\ &= \frac{2(1-\delta)(2-\delta)G^2}{\delta^2(1-\mu)^2}. \qquad (23)\end{aligned}$$

Next, we consider the term $\|\tilde{e}_{t+1}\|^2$ of (14). By definition, we have

$$\begin{aligned}\|\tilde{e}_{t+1}\|^2 &= \|\tilde{c}_t\mathsf{sign}_{\tilde{\beta}_t}(\tilde{p}_t)-\tilde{p}_t\|^2\\ &\le (1-\delta_{\tilde{\beta}_t})\|\tilde{p}_t\|^2\\ &= (1-\delta_{\tilde{\beta}_t})\left\|\frac{1}{M}\sum_{i=1}^{M}c_t\mathsf{sign}_{\beta_{t,i}}(p_{t,i})+\tilde{e}_t\right\|^2\\ &\le (1-\tilde{\delta})(1+\lambda)\|\tilde{e}_t\|^2\\ &\quad+(1-\tilde{\delta})(1+1/\lambda)\left\|\frac{1}{M}\sum_{i=1}^{M}c_t\mathsf{sign}_{\beta_{t,i}}(p_{t,i})\right\|^2\end{aligned}$$

where the last inequality is by Young inequality for any $\lambda>0$, and the fact that $\tilde{\delta}=\min\{\delta_{\tilde{\beta}_t}\}$. We have

$$\begin{aligned}&\left\|\frac{1}{M}\sum_{i=1}^{M}c_t\mathsf{sign}_{\beta_{t,i}}(p_{t,i})\right\|^2\\ &\le \frac{1}{M}\sum_{i=1}^{M}\|c_t\mathsf{sign}_{\beta_{t,i}}(p_{t,i})\|^2 \qquad (24)\\ &\le \frac{1}{M}\sum_{i=1}^{M}\left(2\|c_t\mathsf{sign}_{\beta_{t,i}}(p_{t,i})-p_{t,i}\|^2+2\|p_{t,i}\|^2\right) \qquad (25)\\ &\le \frac{1}{M}\sum_{i=1}^{M}\left(2(1-\delta_{\beta_{t,i}})\|p_{t,i}\|^2+2\|p_{t,i}\|^2\right) \qquad (26)\\ &= \frac{1}{M}\sum_{i=1}^{M}2(2-\delta_{\beta_{t,i}})\|p_{t,i}\|^2\\ &\le 2(2-\delta)\frac{1}{M}\sum_{i=1}^{M}\|p_{t,i}\|^2, \qquad (27)\end{aligned}$$

where (24) is by Lemma 4; (25) is by the Young inequality; (26) is by Lemma 1; and (27) is by $\delta=\min\{\delta_{\beta_{t,i}}\}$. Therefore

$$\begin{aligned}\|\tilde{e}_{t+1}\|^2 &\le (1-\tilde{\delta})(1+\lambda)\|\tilde{e}_t\|^2\\ &\quad+2(2-\delta)(1-\tilde{\delta})(1+1/\lambda)\frac{1}{M}\sum_{i=1}^{M}\|p_{t,i}\|^2\end{aligned}$$

Moreover, (17), (20), and (23) yield

$$\begin{aligned}\frac{1-\delta}{M}\sum_{i=1}^{M}\|p_{t,i}\|^2 &\le (1-\delta)(1+\gamma)\left(\frac{1}{M}\sum_{i=1}^{M}\|e_{t,i}\|^2\right)\\ &\quad+(1-\delta)(1+1/\gamma)\frac{G^2}{(1-\mu)^2}\\ &\le (1-\delta)(1+\gamma)\frac{2(1-\delta)(2-\delta)G^2}{\delta^2(1-\mu)^2}\\ &\quad+(1-\delta)(1+1/\gamma)\frac{G^2}{(1-\mu)^2}\end{aligned}$$

which reduces to the following inequality, because $\gamma = \frac{\delta}{2(1-\delta)}$:

$$\frac{1}{M}\sum_{i=1}^{M}\|p_{t,i}\|^2 \le \frac{2(2-\delta)G^2}{\delta^2(1-\mu)^2}.$$

Therefore

$$\begin{aligned}\|\tilde{e}_{t+1}\|^2 &\le (1-\tilde{\delta})(1+\lambda)\|\tilde{e}_t\|^2\\ &\quad+\frac{4(2-\delta)^2(1-\tilde{\delta})(1+1/\lambda)G^2}{\delta^2(1-\mu)^2}.\end{aligned}$$

Choosing $\lambda = \frac{\tilde{\delta}}{2(1-\tilde{\delta})}$, we obtain

$$\begin{aligned}(1-\tilde{\delta})(1+\lambda) &= 1-\frac{\tilde{\delta}}{2}\\ 1+1/\lambda &= \frac{2-\tilde{\delta}}{\tilde{\delta}}.\end{aligned}$$

Therefore

$$\|\tilde{e}_{t+1}\|^2 \le \left(1-\frac{\tilde{\delta}}{2}\right)\|\tilde{e}_t\|^2+\frac{4(1-\tilde{\delta})(2-\tilde{\delta})(2-\delta)^2G^2}{\tilde{\delta}\delta^2(1-\mu)^2}. \tag{28}$$

Note that inequality (28) is of the form

$$a_{t+1} \le \alpha a_t + \beta, \tag{29}$$

where

$$\begin{aligned}a_{t+1} &= \|\tilde{e}_{t+1}\|^2\\ \alpha &= 1-\frac{\tilde{\delta}}{2}\\ \beta &= \frac{4(1-\tilde{\delta})(2-\tilde{\delta})(2-\delta)^2G^2}{\tilde{\delta}\delta^2(1-\mu)^2}.\end{aligned}$$

Applying Lemma 3, we obtain

$$\begin{aligned}\|\tilde{e}_{t+1}\|^2 &\le \beta\sum_{j=0}^{t}\alpha^j\\ &\le \frac{8(1-\tilde{\delta})(2-\tilde{\delta})(2-\delta)^2G^2}{(\tilde{\delta})^2\delta^2(1-\mu)^2}\end{aligned} \tag{30}$$

where (30) is by the fact that

$$\sum_{j=0}^{t}\alpha^j \le \sum_{j\ge 0}\alpha^j = \frac{1}{1-\alpha} = \frac{2}{\tilde{\delta}}.$$

Substituting (23) and (30) into (14) gives us

$$\begin{aligned}&\left\|\tilde{e}_{t+1}+\frac{1}{M}\sum_{i=1}^{M}e_{t+1,i}\right\|^2\\ &\le \frac{4(1-\delta)(2-\delta)G^2}{\delta^2(1-\mu)^2}+\frac{16(1-\tilde{\delta})(2-\tilde{\delta})(2-\delta)^2G^2}{(\tilde{\delta})^2\delta^2(1-\mu)^2}\\ &= \frac{4(2-\delta)G^2}{\delta^2(1-\mu)^2}\left(1-\delta+\frac{4(1-\tilde{\delta})(2-\tilde{\delta})(2-\delta)}{(\tilde{\delta})^2}\right).\end{aligned}$$

Let

$$U = \frac{4(2-\delta)}{\delta^2}\left(1-\delta+\frac{4(1-\tilde{\delta})(2-\tilde{\delta})(2-\delta)}{(\tilde{\delta})^2}\right)$$

we have

$$\left\|\tilde{e}_{t+1}+\frac{1}{M}\sum_{i=1}^{M}e_{t+1,i}\right\|^2 \le \frac{G^2\,U}{(1-\mu)^2}$$

and the claim follows. ■

ACKNOWLEDGMENT

The authors are grateful to the anonymous reviewers whose comprehensive comments greatly help improve this manuscript.

REFERENCES

[1] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, V. Q. Le, Z. M. Mao, M. Ranzato, W. A. Senior, A. P. Tucker, K. Yang, and Y. A. Ng, "Large scale distributed deep networks," in *Proc. 26th Annu. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1232–1240.

[2] B. Recht, C. Re, S. Wright, and F. Niu, "Hogwild: A lock-free approach to parallelizing stochastic gradient descent," in *Proc. Adv. Neural Inf. Process. Syst.*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, K. Q. Weinberger, Eds. New York, NY, USA: Curran Associates, 2011, pp. 693–701.

[3] J. Dean and L. A. Barroso, "The tail at scale," *Commun. ACM*, vol. 56, no. 2, pp. 74–80, Feb. 2013.

[4] S. Zheng, Z. Huang, and T. James Kwok, "Communication-efficient distributed blockwise momentum SGD with error-feedback," in *Proc. Adv. Neural Inf. Process. Syst., Annu. Conf. Neural Inf. Process. Syst., NeurIPS*, 2019, pp. 11446–11456. [Online]. Available: https://arxiv.org/abs/1905.10936

[5] H. Tang, C. Yu, X. Lian, T. Zhang, and J. Liu, "DoubleSqueeze: Parallel stochastic gradient descent with double-pass error-compensated compression," in *Proc. 36th Int. Conf. Mach. Learn.*, vol. 97, K. Chaudhuri and R. Salakhutdinov, Eds. 2019, pp. 6155–6165.

[6] D. Basu, D. Data, C. Karakus, and N. Suhas Diggavi, "Qsparse-local-SGB: Distributed SGD with quantization, sparsification and local computations," in *Proc. Adv. Neural Inf. Process. Syst., Annu. Conf. Neural Inf. Process. Syst., NeurIPS*, 2019, pp. 14668–14679.

[7] C. Karakus, Y. Sun, S. Diggavi, and W. Yin, "Redundancy techniques for straggler mitigation in distributed optimization and learning," *J. Mach. Learn. Res.*, vol. 20, no. 72, pp. 1–47, 2019.

[8] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-Robust distributed learning: Towards optimal statistical rates," in *Proc. 35th Int. Conf. Mach. Learn. (ICML)*, vol. 80, Jul. 2018, pp. 5650–5659.

[9] E. M. E. Mhamdi, R. Guerraoui, and S. Rouault, "The hidden vulnerability of distributed learning in Byzantium," in *Proc. 35th Int. Conf. Mach. Learn.*, J. Dy and A. Krause, Eds. Stockholm Sweden: Stockholmsmässan, vol. 80, Jul. 2018, pp. 3521–3530.

[10] T. T. Phuong and L. T. Phong, "Distributed SGD with flexible gradient compression," *IEEE Access*, vol. 8, pp. 64707–64717, 2020.

[11] J. Bernstein, J. Zhao, K. Azizzadenesheli, and A. Anandkumar, "signSGD with majority vote is communication efficient and fault tolerant," in *Proc. 7th Int. Conf. Learn. Represent., ICLR*, 2019, pp. 1–20.

[12] S. P. Karimireddy, Q. Rebjock, U. Sebastian Stich, and M. Jaggi, "Error feedback fixes signSGD and other gradient compression schemes," in *Proc. 36th Int. Conf. Mach. Learn., ICML*, 2019, pp. 3252–3261. [Online]. Available: https://arxiv.org/abs/1901.09847

[13] P. Blanchard, E. M. E. Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant Gradien descent," in *Proc. Adv. Neural Inf. Process. Syst., Annu. Conf. Neural Inf. Process. Syst.*, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds. Long Beach, CA, USA, Dec. 2017, pp. 119–129.

[14] D. Yin, Y. Chen, K. Ramchandran, and L. Peter Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *Proc. 35th Int. Conf. Mach. Learn., ICML*, vol. 80, J. G. Dy and A. Krause, Eds. Stockholm, Sweden: Stockholmsmässan, Jul. 2018, pp. 5636–5645.

[15] D. Alistarh, Z. Allen-Zhu, and J. Li, "Byzantine stochastic gradient descent," in *Proc. Adv. Neural Inf. Process. Syst., Annu. Conf. Neural Inf. Process. Syst., NeurIPS*, S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Ed. Montreal, QC, Canada, Dec. 2018, pp. 4618–4628.

[16] U. Sebastian Stich, J. Cordonnier, and M. Jaggi, "Sparsified SGD with memory," in *Proc. Adv. Neural Inf. Process. Syst., Annu. Conf. Neural Inf. Process. Syst.*, 2018, pp. 4452–4463.

[17] T. Vogels, S. P. Karimireddy, and M. Jaggi, "PowerSGD: Practical low-rank gradient compression for distributed optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. New York, NY, USA: Curran Associates, 2019, pp. 14236–14245.

[18] X. Liu, Y. Li, J. Tang, and M. Yan, "A double residual compression algorithm for efficient distributed learning," in *Proc. 23rd Int. Conf. Artif. Intell. Statist., AISTATS*, S. Chiappa and R. Calandra, Eds. Palermo, Italy, vol. 108, Aug. 2020, pp. 133–143.

[19] T. T. Phuong and L. T. Phong, "Communication-efficient distributed SGD with error-feedback, revisited," 2020, *arXiv:2003.04706*. [Online]. Available: https://arxiv.org/abs/2003.04706

[20] K. Gardner, S. Zbarsky, S. Doroudi, M. Harchol-Balter, and E. Hyytia, "Reducing latency via redundant requests: Exact analysis," in *Proc. ACM SIGMETRICS Int. Conf. Meas. Modeling Comput. Syst. SIGMETRICS*, 2015, pp. 347–360.

[21] G. Ananthanarayanan, A. Ghodsi, S. Shenker, and I. Stoica, "Effective straggler mitigation: Attack of the clones," in *Proc. 10th USENIX Symp. Netw. Syst. Design Implement., NSDI*, N. Feamster and J. C. Mogul, Eds. Lombard, IL, USA: USENIX Association, Apr. 2013, pp. 185–198.

[22] N. B. Shah, K. Lee, and K. Ramchandran, "When do redundant requests reduce latency?" *IEEE Trans. Commun.*, vol. 64, no. 2, pp. 715–722, Feb. 2016.

[23] D. Wang, G. Joshi, and G. Wornell, "Using straggler replication to reduce latency in large-scale parallel computing," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 43, no. 3, pp. 7–11, Nov. 2015.

[24] N. J. Yadwadkar, B. Hariharan, E. Joseph Gonzalez, and R. H. Katz, "Multi-task learning for straggler avoiding predictive job scheduling," *J. Mach. Learn. Res.*, vol. 17, pp. 106:1–106:37, Jan. 2016.

[25] A. Agarwal and J. C. Duchi, "Distributed delayed stochastic optimization," in *Proc. IEEE 51st IEEE Conf. Decis. Control (CDC)*, Dec. 2012, pp. 873–881.

[26] M. Li, G. D. Andersen, J. W. Park, J. A. Smola, A. Ahmed, V. Josifovski, J. Long, J. E. Shekita, and B. Su, "Scaling distributed machine learning with the parameter server," in *Proc. 11th USENIX Symp. Operating Syst. Design Implement., OSDI*, J. F. H. Levy, Ed. Broomfield, CO, USA: USENIX Association, Oct. 2014, pp. 583–598.

[27] J. Chen, R. Monga, S. Bengio, and R. Jozefowicz, "Revisiting distributed synchronous SGD," in *Proc. Int. Conf. Learn. Represent. Workshop Track*, 2016, pp. 1–10

[28] S. Dutta, G. Joshi, S. Ghosh, P. Dube, and P. Nagpurkar, "Slow and stale gradients can win the race: Error-runtime trade-offs in distributed SGD," in *Proc. Int. Conf. Artif. Intell. Statist., AISTATS*, Apr. 2018, pp. 803–812.

[29] R. Tandon, Q. Lei, G. Alexandros Dimakis, and N. Karampatziakis, "Gradient coding: Avoiding stragglers in distributed learning," in *Proc. 34th Int. Conf. Mach. Learn., ICML*, vol. 70, D. P. Y. W. Teh, Ed. Sydney, NSW, Australia, Aug. 2017, pp. 3368–3376.

[30] K. Lee, M. Lam, R. Pedarsani, D. Papailiopoulos, and K. Ramchandran, "Speeding up distributed machine learning using codes," *IEEE Trans. Inf. Theory*, vol. 64, no. 3, pp. 1514–1529, Mar. 2018.

[31] W. Halbawi, N. Azizan, F. Salehi, and B. Hassibi, "Improving distributed gradient descent using Reed–Solomon codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2018, pp. 2027–2031.

[32] J. Sohn, D. Han, B. Choi, and J. Moon, "Election coding for distributed learning: Protecting SignSGD against Byzantine attacks," 2019, *arXiv:1910.06093*. [Online]. Available: https://arxiv.org/abs/1910.06093

[33] C. Torres-Huitzil and B. Girau, "Fault and error tolerance in neural networks: A review," *IEEE Access*, vol. 5, pp. 17322–17341, 2017.

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[35] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[36] Z. Huang. *Source Code for Dist-Ef-SGDM*. Accessed: Jul. 1, 2020. [Online]. Available: https://github.com/ZiyueHuang/dist-ef-sgdm/tree/master/imagenet

LE TRIEU PHONG received the Ph.D. degree from the Tokyo Institute of Technology, in 2009. He was an expert and the editor of multiple ISO/IEC information security standard and documents. He is currently a Senior Researcher with the National Institute of Information and Communications Technology (NICT), Japan. He has (co) authored more than 30 scientific articles, including several published by the IEEE. His current research interests include deep learning and computer science in general.

TRAN THI PHUONG received the Ph.D. degree in mathematics from Meiji University, in 2012. She has (co) authored more than ten scientific articles, including several published by the IEEE. Her current research interests include mathematics and deep learning, especially mathematical foundations of deep leaning and mathematical convergence of learning algorithms.