A Convergence Analysis of Distributed SGD with Communication-Efficient Gradient Sparsification

Shaohuai Shi, **Kaiyong Zhao**, **Qiang Wang**, **Zhenheng Tang** and **Xiaowen Chu**
Department of Computer Science, Hong Kong Baptist University
{csshshi, kyzhao, qiangwang, zhtang, chxw}@comp.hkbu.edu.hk

Abstract

Gradient sparsification is a promising technique to significantly reduce the communication overhead in decentralized synchronous stochastic gradient descent (S-SGD) algorithms. Yet, many existing gradient sparsification schemes (e.g., Top-k sparsification) have a communication complexity of $O(kP)$, where k is the number of selected gradients by each worker and P is the number of workers. Recently, the gTop-k sparsification scheme has been proposed to reduce the communication complexity from $O(kP)$ to $O(k \log P)$, which significantly boosts the system scalability. However, it remains unclear whether the gTop-k sparsification scheme can converge in theory. In this paper, we first provide theoretical proofs on the convergence of the gTop-k scheme for non-convex objective functions under certain analytic assumptions. We then derive the convergence rate of gTop-k S-SGD, which is at the same order as the vanilla mini-batch SGD. Finally, we conduct extensive experiments on different machine learning models and data sets to verify the soundness of the assumptions and theoretical results, and discuss the impact of the compression ratio on the convergence performance.

1 Introduction

Stochastic gradient descent (SGD) algorithms are commonly used for training many machine-learning models. SGD minimizes the objective function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ with stochastic gradients $G(x_t)$ using the following update formula:

$$x_{t+1} = x_t - \alpha_t G(x_t), \tag{1}$$

where $x_t \in \mathbb{R}^d$ is a set of model parameters, and $\alpha_t \in \mathbb{R}$ is the step size at iteration t. With large-scale models (i.e., d is at the order of millions or even billions) and data sets, distributed synchronous SGD (S-SGD) with data-parallelism is the key technique to reduce the overall training time using multiple computational workers [Goyal *et al.*, 2017; Jia *et al.*, 2018]. Given a cluster with P workers, in the t^{th} iteration, the p^{th} worker calculates the gradients $G^p(x_t)$ with locally sampled data, and then all workers collaboratively update the model parameters with the aggregated gradients $\frac{1}{P}\sum_{p=1}^{P} G^p(x_t)$, i.e.,

$$x_{t+1} = x_t - \alpha_t \frac{1}{P}\sum_{p=1}^{P} G^p(x_t). \tag{2}$$

Ideally, S-SGD with P workers would accelerate the training process by P times. However, the aggregation of gradients requires tremendous data communications among workers, whose time cost becomes significant, especially when the network bandwidth is relatively low [Dean *et al.*, 2012; Shi *et al.*, 2018]. Efficient communication methods have been proposed to alleviate the communication overheads on the system level [Awan *et al.*, 2017; Zhang *et al.*, 2017; Shi *et al.*, 2019a; Chen *et al.*, 2019], while the Top-k sparsification scheme [Chen *et al.*, 2018; Lin *et al.*, 2018] has been proposed to sparsify the gradients to dramatically reduce the communication cost with little impact on the model accuracy on the algorithm level. In Top-k S-SGD, each worker only selects its top-k gradients (in terms of absolute magnitude) to be exchanged with other workers. The update formula becomes:

$$x_{t+1} = x_t - \alpha_t \frac{1}{P}\sum_{p=1}^{P} \widetilde{G}^p(x_t), \tag{3}$$

where $\widetilde{G}^p(x_t) = \text{TopK}(G^p(x_t))$ is the sparsified top-k gradients at the p^{th} worker. Specifically, for a vector $x \in \mathbb{R}^d$, $\text{TopK}(x) \in \mathbb{R}^d$, and the i^{th} ($i = 1, 2, ..., d$) element of $\text{TopK}(x)$ is defined by:

$$\text{TopK}(x)^{(i)} = \begin{cases} x^{(i)}, & \text{if } |x^{(i)}| > thr \\ 0, & \text{otherwise} \end{cases}, \tag{4}$$

where $x^{(i)}$ denotes the i^{th} element of x and thr is the k^{th} largest value of $|x|$. In practice, k can be two to three orders of magnitude smaller than d with little impact on the model accuracy [Aji and Heafield, 2017; Lin *et al.*, 2018; Alistarh *et al.*, 2018], which can dramatically reduce the communication overhead. Some work [Wangni *et al.*, 2018; Stich *et al.*, 2018; Alistarh *et al.*, 2018; Jiang and Agrawal, 2018] has provided the theoretical convergence analysis of Top-k S-SGD under different assumptions.

However, it is noted that the indices of non-zero elements of $\widetilde{G}_i^p(x_t)$ for different workers are generally inconsistent.

Therefore, the aggregation of sparsified gradients of Top-k S-SGD from P workers has a communication complexity of $O(kP)$ [Renggli *et al.*, 2018] [1]. Recently, an alternative gradient sparsification scheme named gTop-k [Shi *et al.*, 2019b] has been proposed to reduce the communication complexity from $O(kP)$ to $O(k\log P)$ by using a tree based approximate reduction algorithm. Their empirical studies show that gTop-k S-SGD achieves much better training performance than the Top-k scheme on GPU clusters; yet, there is no theoretical justification on the convergence of gTop-k S-SGD.

In this paper, we provide a detailed theoretical analysis on the convergence performance of gTop-k S-SGD on non-convex problems. We summarize our main contributions as follows:

- Unlike the existing convergence analysis in [Stich *et al.*, 2018; Alistarh *et al.*, 2018] that uses some assumptions on $\sum_{p=1}^{P}\widetilde{G}^p(x_t)$ which may not hold in practice, we introduce a relatively weak assumption which can be easily verified through real-world experiments.
- We prove that gTop-k S-SGD provides convergence guarantees for non-convex problems under our analytic assumptions. We conduct extensive experiments on representative deep learning models and data sets to verify the soundness of the assumptions and theoretical results.
- We show that gTop-k S-SGD has the same theoretical convergence rate with vanilla mini-batch SGD with properly chosen learning rates. We also discuss the impact of the compression ratio on the convergence rate through experiments.

2 Related Work

There are two main types of communication reduction schemes in S-SGD: gradient quantization and sparsification.

In quantization methods, the exchanged gradients at every iteration can be quantified to a small number of bits (e.g., 2 bits) with error compensation [Alistarh *et al.*, 2017; Wen *et al.*, 2017; Bernstein *et al.*, 2018; Jiang and Agrawal, 2018; Wu *et al.*, 2018; Stich *et al.*, 2018; Wangni *et al.*, 2018; Haddadpour *et al.*, 2019] during communication while keeping the model accuracy nearly unchanged. However, even using only one bit for each gradient, the maximum communication compression ratio is 32× compared to the 32-bit counterpart.

In sparsification methods, one can only transmit a small portion of non-zero gradients [Aji and Heafield, 2017; Chen *et al.*, 2018; Lin *et al.*, 2018; Stich *et al.*, 2018; Jiang and Agrawal, 2018; Alistarh *et al.*, 2018; Wangni *et al.*, 2018; Wang *et al.*, 2018] for aggregation so that the communication size can be reduced significantly. Researchers [Aji and Heafield, 2017; Chen *et al.*, 2018; Lin *et al.*, 2018; Renggli *et al.*, 2018] first empirically show the effectiveness of the Top-k or the random-k sparsification in S-SGD with little impact on the model convergence, where k can be only 0.1% of the gradient dimension d. Some recent work [Stich *et al.*, 2018; Jiang and Agrawal, 2018; Alistarh *et al.*, 2018; Wangni *et al.*, 2018] provides the convergence analysis on Top-k S-SGD under different assumptions. However, even though the Top-k sparsification scheme can zero-out a large number of gradients, it generates the irregular indices among different workers such that the communication complexity is $O(kP)$ for P workers [Renggli *et al.*, 2018], which limits the system scalability. To further reduce the communication complexity, a communication-efficient sparsification scheme named gTop-k has been recently proposed [Shi *et al.*, 2019b]. gTop-k has a communication complexity of $O(k\log P)$ by leveraging a tree structure for gradient communications, and therefore it performs much better than Top-k on large clusters. However, there is no theoretical justification on the convergence of gTop-k S-SGD in [Shi *et al.*, 2019b]. Due to the biased gradients aggregation through the gTop-k sparsification, the theoretical convergence analysis is non-trivial. In this study, we provide the convergence proofs for gTop-k S-SGD, and we conclude that gTop-k S-SGD has the same convergence rate as vanilla S-SGD.

We want to highlight that the existing convergence analysis on Top-k S-SGD [Wangni *et al.*, 2018; Stich *et al.*, 2018; Alistarh *et al.*, 2018; Jiang and Agrawal, 2018] cannot be directly applied to prove the convergence of gTop-k S-SGD. First, gTop-k S-SGD is a biased stochastic compression scheme which is different with the unbiased one in [Wangni *et al.*, 2018]. Second, the analysis in [Stich *et al.*, 2018] is for convex problems, and it requires the sparsification on fully aggregated gradients, while gTop-k S-SGD has no such condition. Third, the analysis in [Jiang and Agrawal, 2018] requires the algorithm to exchange all parameter components in any certain T consecutive iterations, which could also not hold on gTop-k S-SGD since in every iteration only top-k gradients are selected and some very small gradients may not be chosen throughout the training process. Our analysis is closer to the work [Alistarh *et al.*, 2018], but there are three main technical differences. 1) We use a relatively weak analytic assumption on the top-k gradients (and also gTop-k gradients). 2) We eliminate the condition ($k > d/2$) that is required in [Alistarh *et al.*, 2018] to guarantee the convergence. 3) We prove the convergence of the gTop-k S-SGD algorithm, and derive the convergence rate, and empirically evaluate the impact of compression ratio on the convergence performance.

3 The Algorithm of gTop-k S-SGD

For completeness, in this section we briefly introduce the algorithm of communication-efficient global Top-k (gTop-k) S-SGD proposed in [Shi *et al.*, 2019b]. Before describing the algorithm, we define some notations. Let v_t and ϵ_t^p denote the local model of each worker and the local gradient residuals of worker p at iteration t, respectively. Note that all workers have the consistent model at any iteration. In gTop-k S-SGD, the model is updated by

$$v_{t+1} = v_t - \alpha_t \frac{1}{P}\mathrm{gTopK}_{p=1}^{P}(G_t^p(v_t) + \epsilon_t^p), \qquad (5)$$

where $\mathrm{gTopK}_{p=1}^{P}(x^p) = x^1 \top x^2 \top ... \top x^P$, and the operator $\top$ is defined as follows. For any two vectors $x^i \in \mathbb{R}^d$ and

[1] For all-reduce based aggregation, every worker has a communication complexity of $O(kP)$. For parameter-server based aggregation, the parameter server has a communication complexity of $O(kP)$.

$x^j \in \mathbb{R}^d$,

$$x^i \top x^j = mask \odot |\text{TopK}(x^i) + \text{TopK}(x^j)|, \tag{6}$$

where $mask = |\text{TopK}(x^i) + \text{TopK}(x^j)| > thr$ and thr is the k^{th} largest value of $|\text{TopK}(x^i) + \text{TopK}(x^j)|$. Assume that x is the aggregation result by $\text{gTopK}_{p=1}^{P}(x^p)$, it simultaneously generates a vector of $gMask^p \in \mathbb{R}^d$ which indicates the indices of the selected local values (i.e., $\text{TopK}(x^p)$) that contribute to the final x. Specifically, the i^{th} ($i = 1, 2, ..., d$) element of $gMask^p$ is defined as

$$gMask^{p,(i)} = \begin{cases} 1, & \text{If } \text{TopK}(x^p)^{(i)} \text{ contributes to } x^{(i)} \\ 0, & \text{otherwise} \end{cases}. \tag{7}$$

The pseudocode of gTop-k S-SGD is shown in Algorithm 1.

Algorithm 1 gTop-k S-SGD at worker p

```
Input: Stochastic gradients $G^p(\cdot)$ at worker $p$
Input: Configured value $k$ and the learning rate $\alpha$
Initialize $v_0 = \epsilon_0^p = 0$;
for $t = 1 \to T$ do
    $acc_t^p = \epsilon_{t-1}^p + \alpha G_t^p(v_{t-1})$; // Accumulate the residuals
    $g_t, gMask_t^p = \text{gTopK}_{p=1}^{P}(acc_t^p)$; // Global top-$k$ and mask
    $\epsilon_t^p = acc_t^p \odot \neg gMask_t^p$; // Store residuals
    $v_t = v_{t-1} - \frac{1}{P} g_t$; // Update the model
end for
```

Similar to [Alistarh *et al.*, 2018], we also use x_t to denote the auxiliary random variable at iteration t, and

$$x_{t+1} = x_t - \alpha G_t(v_t), \tag{8}$$

where $G_t(v_t) = \frac{1}{P}\sum_{p=1}^{P} G_t^p(v_t)$ and $x_0 = 0^d$. The difference between the auxiliary variable x_t and the model variable v_t can be represented by

$$\epsilon_t = v_t - x_t. \tag{9}$$

According to Algorithm 1, we have $\epsilon_t = \frac{1}{P}\sum_{p=1}^{P} \epsilon_t^p$.

4 Convergence Analysis

4.1 Notations and Assumptions

We mainly discuss the cases that all the computational workers have a full copy of data. We assume that the gTop-k S-SGD is applied to solve the non-convex objective function $f : \mathbb{R}^d \to \mathbb{R}$, which is L-Lipschitz smooth, i.e.,

$$||\nabla f(x) - \nabla f(y)|| \leq L||x - y||, \forall x, y \in \mathbb{R}^d. \tag{10}$$

The sampled stochastic gradients $G(\cdot)$ at every iteration are unbiased, i.e., $\mathbb{E}[G(v_t)] = \nabla f(v_t)$. We also assume that the second moment of the stochastic gradients is bounded, i.e.,

$$\mathbb{E}[||G_t^{p,(i)}(x)||^2] \leq M^2, \forall x \in \mathbb{R}^d, \forall t \in \mathbb{N}, \tag{11}$$

where $G_t^{p,(i)}(x)$ are the gradients of the i^{th} sample in a mini-batch and $||\cdot||$ is ℓ_2-norm. Let b denote the mini-batch size used per worker, and the total mini-batch size with P workers is $B = Pb$. The mini-batch setting has $G_t^p(x) = \frac{1}{b}\sum_{i=1}^{b} G_t^{p,(i)}(x)$. Thus, the second moment of the average gradients has a smaller bound, i.e., for any $t \in \mathbb{N}$,

$$\mathbb{E}[||\frac{1}{P}\sum_{p=1}^{P} G_t^p(x)||^2] \leq \frac{M^2}{Pb} = \frac{M^2}{B}, \forall x \in \mathbb{R}^d. \tag{12}$$

Assumption 1. *The* gTopK *operator is expected to select k larger values than randomly selecting k values from the accumulated vectors, i.e.,*

$$\mathbb{E}[||\frac{1}{P}\sum_{p=1}^{P} x^p - \frac{1}{P}\text{gTopK}_{p=1}^{P} x^p||^2] \leq$$
$$\mathbb{E}[||\frac{1}{P}\sum_{p=1}^{P} x^p - \text{randomK}(\frac{1}{P}\sum_{p=1}^{P} x^p)||^2], \tag{13}$$

where $\text{randomK}(x^p) \in \mathbb{R}^d$ *is a vector whose k elements are randomly selected from x^p following a uniform distribution, and the other $d - k$ elements are zeros.*

The assumption will be verified by experiments in Section 5. The key ideas of the proofs are 1) We first bound the difference between the model x_t without sparsification and the sparsified model v_t. It enables us to bound the expected sum-of-squares of gradients of f so that the convergence is guaranteed [Bottou *et al.*, 2018]. 2) Then we bound the expected average-squared gradients of f with some sufficient conditions to derive the convergence rate.

4.2 Main Results

Lemma 1. *For any vectors $x^p \in \mathbb{R}^d, p = 1, 2, ..., P$, and $0 < k \leq d$, it holds that*

$$\mathbb{E}[||\sum_{p=1}^{P} x^p - \text{gTopK}_{p=1}^{P} x^p||^2] \leq (1 - \frac{k}{d})||\sum_{p=1}^{P} x^p||^2 \tag{14}$$

Proof. In [Stich *et al.*, 2018], the authors have shown that for any vector $x \in \mathbb{R}^d$, it holds

$$\mathbb{E}[||x - \text{randomK}(x)||^2] = (1 - \frac{k}{d})||x||^2. \tag{15}$$

Combined with Assumption 1, we easily obtain

$$\mathbb{E}[||\sum_{p=1}^{P} x^p - \text{gTopK}_{p=1}^{P} x^p||^2] \leq (1 - \frac{k}{d})||\sum_{p=1}^{P} x^p||^2$$

□

Lemma 2. *For any iteration $t \geq 1$:*

$$\mathbb{E}[||v_t - x_t||^2] \leq \frac{1}{\eta}\sum_{i=1}^{t}(\gamma(1+\eta))^i \mathbb{E}[||x_{t-i+1} - x_{t-i}||^2], \tag{16}$$

where $\gamma = 1 - \frac{k}{d}$, $0 < k \leq d$ and $\eta > 0$.

Proof. We derive the difference between v_{t+1} and x_{t+1}, i.e.,

$$\mathbb{E}[||v_{t+1} - x_{t+1}||^2] = \mathbb{E}[||\frac{1}{P}\sum_{p=1}^{P}(\alpha_t G_t^p(v_t) + \epsilon_t^p)$$

$$
\begin{aligned}
&+ v_t - x_t - \epsilon_t - \frac{1}{P}\text{gTopK}_{p=1}^{P}(\alpha_t G_t^p(v_t) + \epsilon_t^p)||^2] = \\
&\mathbb{E}[||\frac{1}{P}\sum_{p=1}^{P}(\alpha_t G_t^p(v_t) + \epsilon_t^p) - \frac{1}{P}\text{gTopK}_{p=1}^{P}(\alpha_t G_t^p(v_t) + \epsilon_t^p)||^2] \\
&\leq \gamma||\frac{1}{P}\sum_{p=1}^{P}(\alpha_t G_t^p(v_t) + \epsilon_t^p)||^2 \text{ (by Lemma 1)} \\
&= \gamma||\alpha_t G_t(v_t) + v_t - x_t||^2 \\
&\leq \gamma(1+\eta)\mathbb{E}[||v_t - x_t||^2] + \gamma(1+\frac{1}{\eta})\mathbb{E}[||\alpha_t G_t(v_t)||^2] \\
&= \gamma(1+\eta)\mathbb{E}[||v_t - x_t||^2] + \gamma(1+\frac{1}{\eta})\mathbb{E}[||x_{t+1} - x_t||^2].
\end{aligned}
$$

Iterating the above inequality from $i = 0 \to t$ yields:

$$
\begin{aligned}
&\mathbb{E}[||v_t - x_t||^2] \\
\leq&\gamma(1+\frac{1}{\eta})\sum_{i=1}^{t}(\gamma(1+\eta))^{i-1}\mathbb{E}[||x_{t-i+1} - x_{t-i}||^2] \\
=&\frac{1}{\eta}\sum_{i=1}^{t}(\gamma(1+\eta))^{i}\mathbb{E}[||x_{t-i+1} - x_{t-i}||^2].
\end{aligned}
$$

□

Corollary 1.

$$
\mathbb{E}[||v_t - x_t||^2] \leq \frac{1}{\eta}\sum_{i=1}^{t}(\gamma(1+\eta))^{i}\alpha_{t-i}^2\frac{M^2}{B}. \qquad (17)
$$

Proof. Using Lemma 2 and the bound of (12), we have

$$
\begin{aligned}
\mathbb{E}[||v_t - x_t||^2] &\leq \frac{1}{\eta}\sum_{i=1}^{t}(\gamma(1+\eta))^{i}\mathbb{E}[||x_{t-i+1} - x_{t-i}||^2] \\
&= \frac{1}{\eta}\sum_{i=1}^{t}(\gamma(1+\eta))^{i}\mathbb{E}[||\alpha_{t-i}G_{t-i}(v_{t-i})||^2] \\
&\leq \frac{1}{\eta}\sum_{i=1}^{t}(\gamma(1+\eta))^{i}\alpha_{t-i}^2\frac{M^2}{B}.
\end{aligned}
$$

□

Theorem 1. *Assume that gTop-k S-SGD is applied to minimize the objective function f that satisfies the assumptions in Section 4.1. If one chooses a learning rate schedule such that for any iteration $t > 0$:*

$$
\sum_{i=1}^{t}(\gamma(1+\eta))^{i}\frac{\alpha_{t-i}^2}{\alpha_t} \leq D, \qquad (18)
$$

for some constant $D > 0$, then after running T iterations with Algorithm 1, we have

$$
\begin{aligned}
&\frac{1}{\sum_{t=1}^{T}\alpha_t}\sum_{t=1}^{T}\alpha_t\mathbb{E}[||\nabla f(v_t)||^2] \leq \\
&\frac{4(f(x_0) - f(x^*))}{\sum_{t=1}^{T}\alpha_t} + \frac{(L + \frac{2L^2D}{\eta})\frac{2M^2}{B}\sum_{t=1}^{T}\alpha_t^2}{\sum_{t=1}^{T}\alpha_t}, \qquad (19)
\end{aligned}
$$

where x^* is the optimal solution to the objective function f.

Proof. Under the Assumption of L-smooth of f, we have

$$
\begin{aligned}
f(x_{t+1}) - f(x_t) &\leq \nabla f(x_t)^\top(x_{t+1} - x_t) + \frac{L}{2}||x_{t+1} - x_t||^2 \\
&= -\alpha_t\nabla f(x_t)^\top G_t(v_t) + \frac{\alpha_t^2 L}{2}||G_t(v_t)||^2. \qquad (20)
\end{aligned}
$$

Taking the expectation at iteration t, we have

$$
\begin{aligned}
&\mathbb{E}[f(x_{t+1})] - f(x_t) \\
\leq& -\alpha_t\nabla f(x_t)^\top\mathbb{E}[G_t(v_t)] + \frac{\alpha_t^2 L}{2}\mathbb{E}||G_t(v_t)||^2 \\
=& -\alpha_t\nabla f(x_t)^\top\nabla f(v_t) + \frac{\alpha_t^2 L}{2}\mathbb{E}[||G_t(v_t)||^2] \\
=& -\frac{\alpha_t}{2}||\nabla f(x_t)||^2 - \frac{\alpha_t}{2}||\nabla f(v_t)||^2 \\
&+ \frac{\alpha_t}{2}||\nabla f(x_t) - \nabla f(v_t)||^2 + \frac{\alpha_t^2 L}{2}\mathbb{E}[||G_t(v_t)||^2] \\
\leq& -\frac{\alpha_t}{2}||\nabla f(x_t)||^2 + \frac{\alpha_t L^2}{2}||v_t - x_t||^2 + \frac{\alpha_t^2 L}{2}\mathbb{E}[||G_t(v_t)||^2] \\
=& -\frac{\alpha_t}{2}(||\nabla f(x_t)||^2 + L^2||v_t - x_t||^2) \\
&+ \alpha_t L^2||v_t - x_t||^2 + \frac{\alpha_t^2 L}{2}\mathbb{E}[||G_t(v_t)||^2] \\
\leq& -\frac{\alpha_t}{2}(||\nabla f(x_t)||^2 + L^2||v_t - x_t||^2) \\
&+ \alpha_t L^2||v_t - x_t||^2 + \frac{\alpha_t^2 L M^2}{2B}.
\end{aligned}
$$

Taking the expectation before t, it yields

$$
\begin{aligned}
&\mathbb{E}[f(x_{t+1})] - \mathbb{E}[f(x_t)] \leq \alpha_t L^2\mathbb{E}[||v_t - x_t||^2] \\
&\quad + \frac{\alpha_t^2 L M^2}{2B} - \frac{\alpha_t}{2}\mathbb{E}[(||\nabla f(x_t)||^2 + L^2||v_t - x_t||^2)] \\
\leq&\frac{\alpha_t L^2}{\eta}\sum_{i=1}^{t}(\gamma(1+\eta))^{i}\alpha_{t-i}^2\frac{M^2}{B} + \frac{\alpha_t^2 L M^2}{2B} \\
&\quad - \frac{\alpha_t}{2}\mathbb{E}[(||\nabla f(x_t)||^2 + L^2||v_t - x_t||^2)] \\
=&\frac{\alpha_t^2 L^2}{\eta}\sum_{i=1}^{t}(\gamma(1+\eta))^{i}\frac{\alpha_{t-i}^2}{\alpha_t}\frac{M^2}{B} + \frac{\alpha_t^2 L M^2}{2B} \\
&\quad - \frac{\alpha_t}{2}\mathbb{E}[(||\nabla f(x_t)||^2 + L^2||v_t - x_t||^2)].
\end{aligned}
$$

Apply (18) to the above inequality, we have

$$
\begin{aligned}
&\mathbb{E}[f(x_{t+1})] - \mathbb{E}[f(x_t)] \leq (L + \frac{2L^2D}{\eta})\frac{M^2\alpha_t^2}{2B} \\
&\quad - \frac{\alpha_t}{2}\mathbb{E}[(||\nabla f(x_t)||^2 + L^2||v_t - x_t||^2)].
\end{aligned}
$$

Then we can obtain

$$
\begin{aligned}
&\alpha_t\mathbb{E}[(||\nabla f(x_t)||^2 + L^2||v_t - x_t||^2)] \leq \\
&2(\mathbb{E}[f(x_t)] - \mathbb{E}[f(x_{t+1})]) + (L + \frac{2L^2D}{\eta})\frac{M^2\alpha_t^2}{B}. \qquad (21)
\end{aligned}
$$

Using the L-smooth property of f, we have

$$\begin{aligned}||\nabla f(v_t)||^2 &= ||\nabla f(v_t) - \nabla f(x_t) + \nabla f(x_t)||^2\\ &\leq 2||\nabla f(v_t) - \nabla f(x_t)||^2 + 2||\nabla f(x_t)||^2\\ &\leq 2L^2||v_t - x_t||^2 + 2||\nabla f(x_t)||^2.\end{aligned}$$

Combine with (21), we obtain

$$\begin{aligned}&\alpha_t\mathbb{E}[||\nabla f(v_t)||^2] \leq 2\alpha_t\mathbb{E}[L^2||v_t - x_t||^2 + ||\nabla f(x_t)||^2]\\ &\leq 4(\mathbb{E}[f(x_t)] - \mathbb{E}[f(x_{t+1})]) + 2(L + \frac{2L^2D}{\eta})\frac{M^2\alpha_t^2}{B}.\end{aligned}$$

Summing up the above inequality for $t = 1, 2, ..., T$, we have

$$\sum_{t=1}^{T}\alpha_t\mathbb{E}[||\nabla f(v_t)||^2] \leq$$

$$4(f(x_0) - f(x^*)) + 2(L + \frac{2L^2D}{\eta})\frac{M^2}{B}\sum_{t=1}^{T}\alpha_t^2. \quad (22)$$

By dividing the summation of learning rates, we have:

$$\frac{1}{\sum_{t=1}^{T}\alpha_t}\sum_{t=1}^{T}\alpha_t\mathbb{E}[||\nabla f(v_t)||^2] \leq$$

$$\frac{4(f(x_0) - f(x^*))}{\sum_{t=1}^{T}\alpha_t} + \frac{2(L + \frac{2L^2D}{\eta})\frac{M^2}{B}\sum_{t=1}^{T}\alpha_t^2}{\sum_{t=1}^{T}\alpha_t} \quad (23)$$

□

The condition (18) holds if $\gamma(1+\eta) < 1$ for both fixed learning rates and diminishing learning rates. To derive the bound of η, we have

$$\gamma(1+\eta) = (1 - \frac{k}{d})(1+\eta) < 1.$$

Therefore, one should choose $\eta < \frac{k}{d-k}$ to satisfy the above inequality. Theorem 1 implies that Algorithm 1 converges to 0 if T is large enough, when α_t is set to satisfy the following conditions:

$$\lim_{T\to\infty}\sum_{t=1}^{T}\alpha_t = \infty \text{ and } \lim_{T\to\infty}\frac{\sum_{t=1}^{T}\alpha_t^2}{\sum_{t=1}^{T}\alpha_t} = 0. \quad (24)$$

Corollary 2. *Under the assumptions in Theorem 1, if $\tau = \gamma(1+\eta)$ and $\alpha_t = \theta\sqrt{B/T}, \forall t > 0$, where $\theta > 0$ is a constant, we have the convergence rate for Algorithm 1:*

$$\begin{aligned}&\mathbb{E}[\frac{1}{T}\sum_{t=1}^{T}||\nabla f(v_t)||^2]\\ &\leq \frac{4\theta^{-1}(f(x_0) - f(x^*)) + 2\theta LM^2}{\sqrt{BT}} + \frac{4\frac{\tau}{(1-\tau)\eta}L^2M^2\theta^2}{T}.\end{aligned} \quad (25)$$

Proof. First we prove that $\alpha_t = \theta\sqrt{B/T}$, which is a constant step size (or learning rate), satisfies the condition in (18). We set $\alpha_t = \alpha$ for simplification (i.e., $\alpha = \theta\sqrt{B/T}$). We have

$$\sum_{i=1}^{t}(\gamma(1+\eta))^i\frac{\alpha_{t-i}^2}{\alpha_t} = \sum_{i=1}^{t}\tau^i\frac{\alpha_{t-i}^2}{\alpha_t} = \alpha\sum_{i=1}^{t}\tau^i = \alpha\frac{\tau(1-\tau^t)}{1-\tau}.$$

Since $0 \leq \tau < 1$, we obtain

$$\lim_{t\to\infty}\alpha\frac{\tau(1-\tau^t)}{1-\tau} = \frac{\alpha\tau}{1-\tau}.$$

Therefore, (18) holds by choosing $D = \frac{\alpha\tau}{1-\tau}$. Applying Theorem 1, we obtain the inequality of the expected average-squared gradients of f, i.e.,

$$\begin{aligned}&\mathbb{E}[\frac{1}{T}\sum_{t=1}^{T}||\nabla f(v_t)||^2]\\ &\leq \frac{4(f(x_0) - f(x^*))}{\alpha T} + 2(L + \frac{2L^2D}{\eta})\frac{M^2\alpha}{B}\\ &= \frac{4\theta^{-1}(f(x_0) - f(x^*))}{\sqrt{BT}} + \frac{2LM^2\theta}{\sqrt{BT}} + \frac{4\frac{\tau}{(1-\tau)\eta}L^2M^2\theta^2}{T},\end{aligned}$$

which concludes the proof. □

From Corollary 2, we can seen that with a properly set learning rate, the gTop-k S-SGD algorithm has a convergence rate of $O(\frac{1}{\sqrt{BT}})$, which is the same as that of mini-batch SGD [Dekel *et al.*, 2012]. It also indicates that k has small impact on the convergence rate if T is large enough.

4.3 Discussion

In Corollary 2, there are two terms to determine the convergence rate of gTop-k S-SGD. The first term indicates that the convergence rate is affected by the constant θ and the mini-batch size, and the second term indicates that the convergence rate is also affected by both θ (related to the learning rate) and τ (related to the compression ratio $\frac{d}{k}$). The second term will be dominated by the first term if T is large enough. However, it is not uncommon that a fixed number of iterations is used for training deep neural networks (DNNs) in practice. As a result, although a larger compression ratio leads to less communications overhead, it would enlarge the bound of the convergence rate.

To understand the details, we expand the second term on the right-hand side of (25). Let $c = d/k$ denote the compression ratio, then $\gamma = 1 - 1/c$ and $\tau = (1 - 1/c)(1+\eta)$. Since η should satisfy the condition of $\eta < k/(d-k)$, we choose $\eta = k/d = 1/c$. Thus,

$$\frac{\tau}{(1-\tau)\eta} = \frac{(1-1/c)(1+\eta)}{\eta - \eta(1-1/c)(1+\eta)} = c^3 - c. \quad (26)$$

Therefore, inequality (25) becomes

$$\begin{aligned}&\mathbb{E}[\frac{1}{T}\sum_{t=1}^{T}||\nabla f(v_t)||^2]\\ &\leq \frac{4\theta^{-1}(f(x_0) - f(x^*)) + 2\theta LM^2}{\sqrt{BT}} + \frac{4L^2M^2(c^3-c)\theta^2}{T}.\end{aligned} \quad (27)$$

The above inequality indicates that given a fixed iteration budget (i.e., T), a higher compression ratio (c) causes a larger bound of the convergence rate. In summary, to achieve a better convergence with a given time budget, one should balance the communication cost and the convergence rate. We will further evaluate the impact of the compression ratio on the convergence performance through experiments in Section 5.

5 Experiments

5.1 Experimental Settings

Our experimental settings cover three deep learning applications. 1) Image classification: Two popular DNNs, VGG-16 [Simonyan and Zisserman, 2014] and ResNet-20 [He *et al.*, 2016], are used for evaluation on the data set of Cifar-10[2] which consists of 50000 training images. 2) Language model: A 2-layer LSTM model (LSTM-PTB) with 1500 hidden units per layer is adopted for evaluation on the data set of PTB [Marcus *et al.*, 1993], which contains 923000 training words. 3) Speech recognition: A 5-layer LSTM model (LSTM-AN4) with 800 hidden units per layer is used for evaluation on AN4 [Acero, 1990], which contains 948 training utterances. In all training models, we exploit the warmup strategy in gTop-k S-SGD on the 4-worker distributed environment. The baselines are evaluated using S-SGD without gradient sparsification. The main hyper-parameters adopted in evaluation are shown in Table 1.

DNN	B	Initial α	# of epochs
VGG-16	512	0.1	140
ResNet-20	128	0.1	140
LSTM-PTB	400	30	40
LSTM-AN4	32	0.0002	80

Table 1: Hyper-parameters for different DNNs

5.2 Verification of Assumption and Convergences

We verify Assumption 1 empirically by training DNNs with gTop-k S-SGD. During the training process, we measure

$$\delta = \frac{\mathbb{E}[||\frac{1}{P}\sum_{p=1}^{P} x^p - \frac{1}{P}\text{gTopK}_{p=1}^{P} x^p||^2]}{\mathbb{E}[||\frac{1}{P}\sum_{p=1}^{P} x^p - \text{randomK}(\frac{1}{P}\sum_{p=1}^{P} x^p)||^2]},$$

where $x^p = G_t^p(v_t) + \epsilon_t^p$ and $k = 0.001 \times d$ (i.e., $c = 1000$). Assumption 1 holds if $\delta \leq 1$. The measurements of δ corresponded with the training losses on the evaluated DNNs are shown in Fig. 1. It can be seen that we always have $\delta < 1$, which verifies the soundness of Assumption 1. In Fig. 1, the convergences of gTop-k S-SGD are nearly consistent with S-SGD, which validates our theoretical results and shows that gTop-k S-SGD can converge as fast as S-SGD.

5.3 Convergence Rate v.s. Compression Ratio

The second term in inequality (27) indicates that the convergence rate may be degraded by the compression ratio c. We evaluate the sensitivity of the convergence rates to c on training DNNs without changing hyper-parameters including the total number of iterations (i.e., a fixed number of epochs).

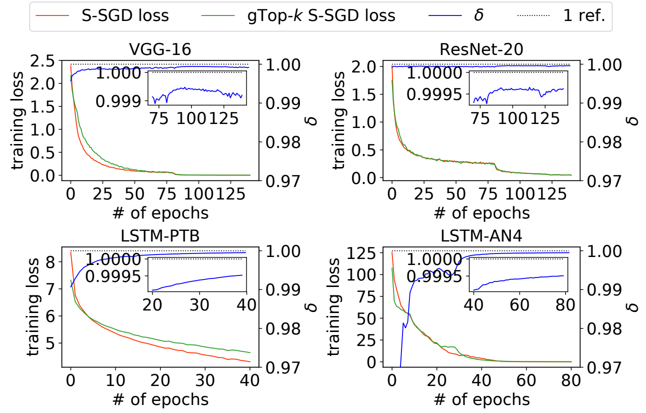

Figure 1: The measurements of δ and convergences on different DNNs with a compression ratio $c = 1000$

The results are shown in Fig. 2, which shows that with larger c, the convergence of the models would slowdown. Therefore, with large compression ratios, there is a trade-off between the communication size, which is directly related to the iteration time, and the convergence rate.

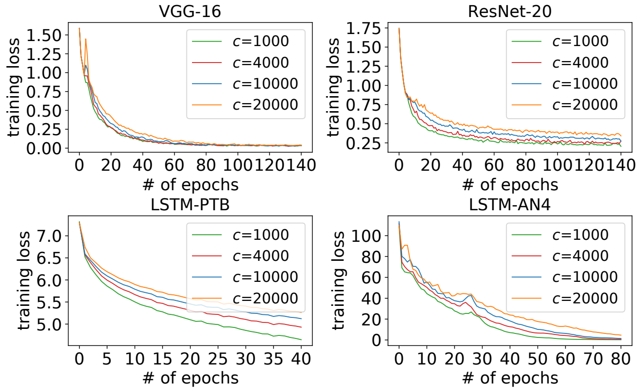

Figure 2: The convergences with different compression ratios

6 Conclusion

Top-k gradient sparsification is crucial for reducing the communication size in distributed S-SGD. The gTop-k scheme is a more communication efficient scheme than Top-k for gradient sparsification. In this study, we present a detailed convergence analysis for gTop-k S-SGD under some analytical assumptions, and we derive its convergence rate. Our theoretical results conclude that gTop-k S-SGD provides convergence guarantees for non-convex objective functions and it has the same convergence rate with vanilla mini-batch SGD with properly chosen learning rates. We derive and evaluate the impact of compression ratios on the convergence performance. We finally conduct experiments to verify the soundness of the analytical assumption and theoretical results.

Acknowledgements

The research was supported by Hong Kong RGC GRF grant HKBU 12200418.

[2] https://www.cs.toronto.edu/~kriz/cifar.html

References

[Acero, 1990] Alejandro Acero. Acoustical and environmental robustness in automatic speech recognition. In *ICASSP*, 1990.

[Aji and Heafield, 2017] Alham Fikri Aji and Kenneth Heafield. Sparse communication for distributed gradient descent. In *EMNLP*, pages 440–445, 2017.

[Alistarh *et al.*, 2017] Dan Alistarh, Demjan Grubic, Jerry Li, et al. QSGD: Communication-efficient SGD via gradient quantization and encoding. In *NIPS*, pages 1709–1720, 2017.

[Alistarh *et al.*, 2018] Dan Alistarh, Torsten Hoefler, Mikael Johansson, et al. The convergence of sparsified gradient methods. In *NeurIPS*, pages 5977–5987, 2018.

[Awan *et al.*, 2017] Ammar Ahmad Awan, Khaled Hamidouche, Jahanzeb Maqbool Hashmi, et al. S-Caffe: Co-designing MPI runtimes and Caffe for scalable deep learning on modern GPU clusters. In *PPoPP*, pages 193–205. ACM, 2017.

[Bernstein *et al.*, 2018] Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, et al. signSGD: compressed optimisation for non-convex problems. In *ICML*, pages 559–568, 2018.

[Bottou *et al.*, 2018] Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311, 2018.

[Chen *et al.*, 2018] Chia-Yu Chen, Jungwook Choi, Daniel Brand, et al. Adacomp: Adaptive residual gradient compression for data-parallel distributed training. In *AAAI*, pages 2827–2835, 2018.

[Chen *et al.*, 2019] Chen Chen, Wei Wang, and Bo Li. Round-robin synchronization: Mitigating communication bottlenecks in parameter servers. In *IEEE INFOCOM*, 2019.

[Dean *et al.*, 2012] Jeffrey Dean, Greg Corrado, Rajat Monga, et al. Large scale distributed deep networks. In *NIPS*, pages 1223–1231, 2012.

[Dekel *et al.*, 2012] Ofer Dekel, Ran Gilad-Bachrach, Ohad Shamir, et al. Optimal distributed online prediction using mini-batches. *Journal of Machine Learning Research*, 13(Jan):165–202, 2012.

[Goyal *et al.*, 2017] Priya Goyal, Piotr Dollár, Ross Girshick, et al. Accurate, large minibatch SGD: training ImageNet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

[Haddadpour *et al.*, 2019] Farzin Haddadpour, Mohammad Mahdi Kamani, Mehrdad Mahdavi, et al. Trading redundancy for communication: Speeding up distributed SGD for non-convex optimization. In *ICML*, pages 2545–2554, 2019.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, et al. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[Jia *et al.*, 2018] Xianyan Jia, Shutao Song, Shaohuai Shi, et al. Highly scalable deep learning training system with mixed-precision: Training ImageNet in four minutes. In *NeurIPS Workshop MLSys*, 2018.

[Jiang and Agrawal, 2018] Peng Jiang and Gagan Agrawal. A linear speedup analysis of distributed deep learning with sparse and quantized communication. In *NeurIPS*, pages 2530–2541, 2018.

[Lin *et al.*, 2018] Yujun Lin, Song Han, Huizi Mao, et al. Deep gradient compression: Reducing the communication bandwidth for distributed training. In *ICLR*, 2018.

[Marcus *et al.*, 1993] Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics*, 19(2):313–330, 1993.

[Renggli *et al.*, 2018] Cèdric Renggli, Dan Alistarh, and Torsten Hoefler. SparCML: High-performance sparse communication for machine learning. *arXiv preprint arXiv:1802.08021*, 2018.

[Shi *et al.*, 2018] Shaohuai Shi, Wang Qiang, and Xiaowen Chu. Performance modeling and evaluation of distributed deep learning frameworks on GPUs. In *IEEE DataCom*, pages 949–957, 2018.

[Shi *et al.*, 2019a] Shaohuai Shi, Xiaowen Chu, and Bo Li. MG-WFBP: Efficient data communication for distributed synchronous SGD algorithms. In *IEEE INFOCOM*, 2019.

[Shi *et al.*, 2019b] Shaohuai Shi, Qiang Wang, Kaiyong Zhao, et al. A distributed synchronous SGD algorithm with global Top-k sparsification for low bandwidth networks. In *IEEE ICDCS*, 2019.

[Simonyan and Zisserman, 2014] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[Stich *et al.*, 2018] Sebastian U Stich, Jean-Baptiste Cordonnier, and Martin Jaggi. Sparsified SGD with memory. In *NeurIPS*, pages 4452–4463, 2018.

[Wang *et al.*, 2018] Hongyi Wang, Scott Sievert, Shengchao Liu, et al. Atomo: Communication-efficient learning via atomic sparsification. In *NeurIPS*, pages 9850–9861, 2018.

[Wangni *et al.*, 2018] Jianqiao Wangni, Jialei Wang, Ji Liu, et al. Gradient sparsification for communication-efficient distributed optimization. In *NeurIPS*, pages 1306–1316, 2018.

[Wen *et al.*, 2017] Wei Wen, Cong Xu, Feng Yan, et al. Terngrad: Ternary gradients to reduce communication in distributed deep learning. In *NIPS*, pages 1509–1519, 2017.

[Wu *et al.*, 2018] Jiaxiang Wu, Weidong Huang, Junzhou Huang, et al. Error compensated quantized SGD and its applications to large-scale distributed optimization. In *ICML*, pages 5321–5329, 2018.

[Zhang *et al.*, 2017] Hao Zhang, Zeyu Zheng, Shizhen Xu, et al. Poseidon: an efficient communication architecture for distributed deep learning on GPU clusters. In *USENIX ATC*, pages 181–193, 2017.