rTop-k: A Statistical Estimation Approach to Distributed SGD

Leighton Pate Barnes, Huseyin A. Inan, Berivan Isik, and Ayfer Özgür

***Abstract*—The large communication cost for exchanging gradients between different nodes significantly limits the scalability of distributed training for large-scale learning models. Motivated by this observation, there has been significant recent interest in techniques that reduce the communication cost of distributed Stochastic Gradient Descent (SGD), with gradient sparsification techniques such as top-k and random-k shown to be particularly effective. The same observation has also motivated a separate line of work in distributed statistical estimation theory focusing on the impact of communication constraints on the estimation efficiency of different statistical models. The primary goal of this paper is to connect these two research lines and demonstrate how statistical estimation models and their analysis can lead to new insights in the design of communication-efficient training techniques. We propose a simple statistical estimation model for the stochastic gradients which captures the sparsity and skewness of their distribution. The statistically optimal communication scheme arising from the analysis of this model leads to a new sparsification technique for SGD, which concatenates random-k and top-k, considered separately in the prior literature. We show through extensive experiments on both image and language domains with CIFAR-10, ImageNet, and Penn Treebank datasets that the concatenated application of these two sparsification methods consistently and significantly outperforms either method applied alone.**

I. Introduction

Stochastic Gradient Descent (SGD) and its variants have become the workhorse for training large machine learning models on ever growing datasets. Such large-scale training can be accelerated by partitioning datasets across multiple nodes and parallelizing the computation of the stochastic gradient; nodes can compute gradients in parallel based on their local datasets, and these local gradients can then be aggregated at a master node. However, for large models with millions of parameters, it is now well-understood that full-precision gradient aggregation is costly and the associated communication overhead can negate the savings in computation time [1]. Communication is an even more significant bottleneck when training is performed over potentially slow, unreliable and expensive wireless links such as in federated learning [2], where locally processing user data offers additional benefits such as privacy and personalization.

Motivated by these observations, there has been significant recent interest in developing communication-efficient SGD methods [1], [3]–[15], as well as other fundamental contributions to distributed first-order [16]–[20] and second-order [21]–[25] optimization procedures. These works show that the communication cost of SGD can be significantly reduced by using a variety of techniques, including thresholding and sparsification of the local gradients (e.g. communicating only the top-k gradients with largest magnitudes or k randomly selected gradients); quantization and compression of gradients (potentially with randomization) to a small number of bits; and reducing the number of communication rounds by performing multiple iterations on the local dataset of each node (e.g. federated averaging). The general methodology for many of these works is to first propose a technique for reducing the communication cost of SGD, and then demonstrate its effectiveness through experimental evaluations and/or prove its convergence under standard assumptions. Such convergence guarantees have been proven in [6], [15], [26], [27] for various communication efficient training techniques.

In this paper, we take a different approach. We ask the following question: can we develop suitable statistical models for the stochastic gradients and use these models to inform the design of more efficient training techniques? This statistical perspective has been the focus of a recent research line in distributed estimation theory [28]–[36]. Motivated by similar observations as above, these works focus on characterizing the impact of communication constraints on the estimation efficiency of common statistical models, e.g. Gaussian mean estimation and its sparse variants, and discrete distribution estimation. However, it is unclear how to leverage these results to inform new practical training methods with SGD; these works focus on a parameter estimation framework rather the training problem, and their emphasis is on canonical statistical models such as Gaussian mean estimation, which may not accurately reflect the distribution of the stochastic gradients.

In this work, we connect these two research lines by casting each communication round of distributed SGD as a communication-constrained parameter estimation problem and propose a statistical model for the stochastic gradients. This model aims to capture the skewed and sparse distribution of the local gradients observed in experiments, which underlies the experimental success of existing sparsification methods such as top-k and thresholding [1], [10]. These methods communicate only a few local gradients per node which have the largest magnitudes, and can reduce the communication cost to order $k \log d$, where d is the number of parameters.

By using the toolset of distributed estimation theory, we characterize the fundamental estimation efficiency for our proposed statistical model and the optimal communication and

All authors contributed equally to this work.

This work was supported in part by NSF award CCF-1704624 and by a Google faculty research award.

L. P. Barnes, H. A. Inan, B. Isik and A. Özgür are with Department of Electrical Engineering, Stanford University, Stanford, CA 94305, USA (e-mail: lpb@stanford.edu; hinan1@stanford.edu; berivan.isik@stanford.edu; aozgur@stanford.edu).

estimation schemes that achieve this performance. The study of this statistical model naturally leads to the idea of communicating a randomly chosen subset of a set of large magnitude gradients. Instead of simply choosing the top-k gradients, the optimal communication scheme chooses a random k-subset of the gradients with large magnitudes. Even though random-k and top-k sparsification have both been separately considered in the literature, selecting a random subset of the top magnitude gradients, which corresponds to their concatenated application, is novel. We call this new sparsification strategy rTop-k. It can be observed from [26], [27], [37] that rTop-k readily enjoys the same convergence guarantees as top-k and random-k since it is also a compression operator (Definition 3 of [26]). However, in extensive experiments on both image and language domains with CIFAR-10, ImageNet, and Penn Treebank datasets we observe that rTop-k significantly outperforms either top-k or random-k applied separately.

The contributions of our paper can be summarized as follows:

- We show that each communication round of distributed SGD can be cast as a communication-constrained statistical parameter estimation problem and that the study of such problems can inform the design of communication-efficient training schemes. To the best of our knowledge, our work is the first to connect distributed stastical estimation theory with communication efficient SGD methods.
- We propose and analyze novel statistical estimation models that capture the skew and sparse distribution of the stochastic gradients. We develop new communication and estimation schemes for these models and prove their optimality via information theoretic lower bounds.
- We introduce a new gradient sparsification scheme, rTop-k, and show that it outperforms either of its two previously known constituent schemes in extensive experiments.

II. Approach and Main Results

A. Distributed Statistical Parameter Estimation

We begin by formulating the distributed parameter estimation problem. Let

$$X_1, X_2, \cdots, X_n \overset{\text{i.i.d.}}{\sim} P_\theta,$$

where $\theta \in \Theta \subseteq \mathbb{R}^d$. We are interested in estimating the parameter θ from the samples $X_1, \ldots, X_n$. Unlike the traditional statistical setting where samples $X_1, \cdots, X_n$ are available to the estimator as they are, we consider a distributed setting where each observation X_i is available at a different node in a network and has to be communicated to a master node by using k bits. In other words, each node has to encode its sample X_i by a possibly randomized strategy Π_i to a k-bit string M_i independently of the other nodes and send it to the master processor. The goal of the master node is to produce an estimate $\hat{\theta}$ of the underlying parameter θ from the nk-bit transcript $M = (M_1, \ldots, M_n)$ it receives from the nodes so as to minimize the worst case squared ℓ^2 risk:

$$\inf_{(\{\Pi_i\}_{i=1}^n, \hat{\theta})} \sup_{\theta \in \Theta} \mathbb{E}_\theta \|\hat{\theta}(M) - \theta\|_2^2, \tag{1}$$

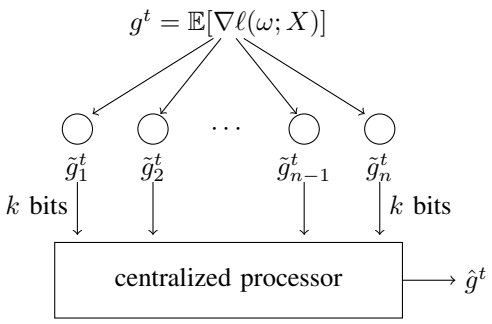

Fig. 1. Statistical estimation model for distributed training.

where $\hat{\theta}(M)$ is an estimator of θ based on the transcript M. Note that the encoding strategies Π_i for $i = 1, \ldots, n$ and the estimator $\hat{\theta}$ can be jointly designed to minimize the risk.

B. Gradient Aggregation in Distributed Training

We next formulate the gradient aggregation problem in distributed training with SGD as a distributed parameter estimation problem. Let $g^t = \mathbb{E}[\nabla \ell(\omega_t; X)]$ denote the gradient of the population risk in $\mathbb{R}^d$ where ℓ denotes the loss function, $\omega_t \in \mathbb{R}^d$ denotes the weights of the network at the current iteration t, and the expectation is with respect to the true distribution of the samples X. Assume there are n distributed nodes and each node i for $i = 1, \ldots, n$ computes the (stochastic) gradient of the loss function with respect to a small batch of i.i.d. samples $\{X_j^{(i)}\}_{j=1}^B$ available at this node,

$$g_i^t = \frac{1}{B} \sum_{j=1}^{B} \nabla \ell(\omega_t; X_j^{(i)}).$$

Note that when disjoint subsets of the dataset are assigned to different nodes, $\{X_j^{(i)}\}_{j=1}^B$ can be modeled as independent and identically distributed for different i. Hence, the local gradients $g_1^t, g_2^t, \ldots, g_n^t \in \mathbb{R}^d$ are generated independently from the same (unknown) distribution and have mean equal to g^t, the true gradient with respect to the population risk. [1] To model the communication bottleneck, assume each node has k bits to communicate its local gradient vector g_i^t to the master node. The goal of the master node is to generate an estimate $\hat{g}^t$ of the true gradient g^t under squared ℓ^2 risk as per (1) from the nk-bit transcript M it obtains at the end of this communication round. The g^t can be regarded as the parameter of the underlying distribution that we wish to estimate. In order to complete the model description, we need to specify the statistical model P_θ, with $\theta = g^t$, according to which the samples $g_1^t, g_2^t, \ldots, g_n^t$ are generated and the set $\Theta \subseteq \mathbb{R}^d$ in which the parameter lies.

C. Statistical Models

The distributed parameter estimation problem in (1) has been studied for specific classes of statistical models P_θ in the literature. One canonical model that has been of significant

[1] The same model can be extended to the case where nodes perform multiple steps of stochastic gradient descent on their local dataset such as in federated learning [2], in which case g_i^t is the resultant model update generated by node i.

interest in the literature is the Gaussian location model $P_\theta = \mathcal{N}(\theta, \sigma^2 I_d)$ with $\Theta = \mathbb{R}^d$ which is studied in [28]–[30], [32]. Adopting this Gaussian model for the gradient aggregation problem would correspond to modeling the local gradients g_i^t computed at each node as i.i.d. observations of the true gradient vector g_t under spherically-symmetric additive Gaussian noise. However, this leads to dense observation vectors g_i^t, while the distribution of the local gradients is often skewed and sparse like in experiments, i.e. often relatively few entries of g_i^t have large magnitudes and many entries have magnitudes close to zero.

A sparse variation of the Gaussian mean estimation problem has been studied in [29], [30], [32], where $\Theta = \{\theta \in \mathbb{R}^d : \|\theta\|_0 \leq s \leq d\}$. Adopting this statistical model for the gradient aggregation problem would correspond to modeling the true gradient vector g_t as sparse, with only s non-zero components, while the local gradients g_i^t computed at each node would still correspond to i.i.d. observations of the true gradient vector g_t under additive Gaussian noise. Even though the true gradient vector g_t is assumed to be sparse, qualitatively this model behaves similarly to the original Gaussian model. In particular the number of bits k needed to achieve the centralized estimation performance (with no communication constraints) scales linearly with the ambient dimension d as in the original Gaussian model, i.e. it is not impacted by the sparsity s. Similarly the optimal communication scheme independently quantizes each gradient component as in the case of the dense Gaussian model.

In this work, we instead propose to focus on the following sparse Bernoulli model. Suppose each node i has a sample

$$X_i \sim \prod_{j=1}^{d} \text{Bern}(\theta_j) \tag{2}$$

with

$$\Theta = \left\{ \theta \in [0,1]^d \, : \, \sum_{j=1}^{d} \theta_j \leq s \right\} .$$

This model imposes a soft sparsity constraint on the parameter vector θ; when $s \ll d$ not all entries of θ can be large, i.e. close to '1'. The observation vectors X_i are highly skewed with entries either equal to '1' or '0'. Note also that the parameter vector θ corresponds to the mean of the observation vectors X_i. We adopt this model for the gradient aggregation problem with the understanding that $\theta = |g_t|$ and the '0' values from the Bernoulli random variables model entries of g_i^t with small magnitudes, while the '1' values model entries with large magnitudes. Note that a large value for a given component of $\theta = |g_t|$ makes it more likely to observe '1', i.e. a large magnitude entry in the corresponding location of the stochastic gradient vectors g_i^t for $i = 1, \ldots, n$. However, it does not exclude the possibility of observing a '0' i.e. a small magnitude entry, in the same location.

While this is a highly simplified model for real stochastic gradient vectors, it allows us to focus our attention on the impact of a highly skewed distribution for the magnitudes of the gradients. This model and our main results in the next section can be easily extended to more accurately reflect the distribution of the stochastic gradients, and doing so does not change the optimal encoding or estimation schemes. In particular, we could consider the following refinements:

(i) **Signed values:** The mean parameters θ_j could be in $[-1, 1]$ subject to $\sum_{j=1}^{d} |\theta_j| \leq s$, so that θ_j's can be positive or negative, with the jth component of the X_i now being $\mathsf{Sign}(\theta_j)\mathsf{Bern}(|\theta_j|)$.

(ii) **Scaling:** The Bernoulli random variables could be scaled by any $M > 0$ with the goal of estimating the scaled Bernoulli parameters $M\theta$.

(iii) **Continuous Perturbations:** We could add a vector of continuous independent zero-mean random variables Z_i to each X_i, where X_i is distributed according to (2) and each component of Z_i is supported in $[-\frac{1}{2}, \frac{1}{2}]$. As a result, each component of the resultant $Y_i = Z_i + X_i$ will now take continuous values.

With none of these refinements changing the fundamental encoding and estimation schemes, we find that the simple sparse Bernoulli model captures the issues that are central to our distributed estimation problem. Note that by including just the sign and scaling changes, the resulting model also makes a good representation of coarsely quantized stochastic gradients such as with ternary quantization.

D. Theoretical Results and Discussion

We now turn to analyzing the distributed estimation problem under this model.

Theorem 1 (sparse Bernoulli upper bound)**.** *For the sparse Bernoulli model (2) above,*

$$\inf_{(\hat{\theta}, \Pi_i)} \sup_{\theta \in \Theta} \mathbb{E}_\theta \|\hat{\theta} - \theta\|_2^2 \leq C \frac{s^2 \log d}{nk}$$

for $2 \log d \leq k \leq s \log d$ *and some constant* C *that is independent of* n, k, d, s.

We prove this theorem in Section V by describing an explicit independent encoding scheme at each node and a centralized estimator that together achieve at most this error. The scheme is built on the idea of subsampling the non-zero entries in each sample vector and communicating the locations of the subsampled components. With signed values as in (i) above, the scheme is modified to communicate also the sign of each subsampled component. Since this requires only a single bit for each subsampled component it does change the scalings in Theorem 1. Similarly, the constant scaling in (ii) also does not impact the scheme and the result of Theorem 1. When continuous perturbations are present as in (iii), the scheme is modified to include a pre-processing step where Y_i is quantized to '0' if $|Y_i| \leq 1/2$, and quantized to '1' if $|Y_i| > 1/2$. This converts the observations to the original Bernoulli model, on which we apply the idea of random subsampling. This recovers the result in Theorem 1. Note that this strategy is different than simply communicating the components of Y_i with largest magnitude or the indexes of these large magnitude components.

We also prove the following lower bound which shows that our proposed scheme is order optimal up to logarithmic factors.

Theorem 2 (Sparse Bernoulli lower bound)**.** *For the sparse Bernoulli model (2) above, if* $nk \geq d \log \frac{d}{s}$ *and* $s \leq \frac{d}{2}$ *then*

$$\sup_{\theta \in \Theta} \mathbb{E}_\theta \|\hat{\theta} - \theta\|_2^2 \geq c \max \left\{ \frac{s^2 \log \frac{d}{s}}{nk}, \frac{s}{n} \right\} \tag{3}$$

for any estimator $\hat{\theta}(M)$*, communication strategies* Π_i*, and some constant* c *that is independent of* n, k, d, s*.*

The proof is given in Section VI. It builds on a geometric characterization of Fisher information from quantized samples, a framework introduced in [35]. The same lower bound applies trivially to extensions (i) and (ii) above, while the extension to (iii) follows simply from the data processing inequality for Fisher information. The two theorems together show that the number of bits k (per node) needed to achieve the centralized performance under the sparse Bernoulli model is of the order of $s \log d$. (The centralized performance is given by the second term in the maximization on the right side of (3).) This model is interesting from a statistical estimation perspective; in contrast to the sparse Gaussian location model, this model suggests that the centralized performance can be achieved with much fewer than d bits when the underlying vector is sparse with $s \ll d$.

III. rTop-k Algorithm

The main algorithmic idea that emerges from the theoretical analysis of the sparse Bernoulli model is to communicate a random subset of the large magnitude entries of the local gradient vectors. Motivated by this observation, we propose a new sparsification operator rTop-k for sparsifying the local gradient vectors, so as to reduce the overall communication cost of distributed SGD. This new sparsification operator can be viewed as a concatenation of top-r and random-k sparsification operators from the prior literature.

Definition 1 (top-r sparsification [1], [10], [26], [27])**.** *For a parameter* $1 \leq r \leq d$*, the operator* $\mathsf{top}_r : \mathbb{R}^d \to \mathbb{R}^d$ *is defined for* $\omega \in \mathbb{R}^d$ *as*

$$\left(\mathsf{top}_r(\omega)\right)_i := \begin{cases} (\omega)_i, & \text{if } i \in \{\pi(1), \ldots, \pi(r)\} \\ 0 & \text{otherwise}, \end{cases}$$

where π *is a permutation of* $[d] := \{1, \ldots, d\}$ *such that* $|(\omega)_{\pi(i)}| \geq |(\omega)_{\pi(i+1)}|$ *for* $i = 1, \ldots, d-1$*.*

Definition 2 (random-k sparsification [9], [26], [27])**.** *For a parameter* $1 \leq k \leq d$*, let* $\mathcal{W}_k$ *denote the set of all* k *element subsets of* $[d]$ *and* W *be uniformly at random chosen element of* $\mathcal{W}_k$*. The operator* $\mathsf{random}_k : \mathbb{R}^d \to \mathbb{R}^d$ *is defined for* $\omega \in \mathbb{R}^d$ *as*

$$\left(\mathsf{random}_k(\omega)\right)_i := \begin{cases} (\omega)_i, & \text{if } i \in W \\ 0 & \text{otherwise.} \end{cases}$$

Definition 3 (rTop-k sparsification)**.** *Let* π *be a permutation of* $[d]$ *such that* $|(\omega)_{\pi(i)}| \geq |(\omega)_{\pi(i+1)}|$ *for* $i = 1, \ldots, d-1$*. For two parameters* $1 \leq k \leq r \leq d$*, let* $\mathcal{U}_k$ *denote the set of all* k *element subsets of* $\{\pi(1), \ldots, \pi(r)\}$ *and* U *be uniformly at random chosen element of* $\mathcal{U}_k$*. The operator* $\mathsf{rTop}_k : \mathbb{R}^d \to \mathbb{R}^d$ *is defined for* $\omega \in \mathbb{R}^d$ *as*

$$\mathsf{rTop}_k(\omega) := \begin{cases} (\omega)_i, & \text{if } i \in U \\ 0 & \text{otherwise.} \end{cases}.$$

The rTop-k sparsification operator first chooses the r entries with largest magnitudes of a vector $\omega \in \mathbb{R}^d$ and then communicates k randomly chosen gradients among these r entries. Note that while both top-r and random-k strategies have been studied extensively in the prior literature, their concatenated application as in rTop-k has not been investigated in prior works. As we demonstrate in the next section, rTop-k consistently outperforms either top-r or random-k applied alone in experiments. Intuitively, top-r sparsification allows to focus only on the most significant entries of the gradient vector and random-k sparsification allows to reduce the bias introduced by top-r sparsification, hence concatenation combines the best of the two approaches.

Note that using this algorithm, k is the final number of components that must be communicated from each node to the centralized processor. Since the index for each component can be referred to with $\log d$ bits, and the value of each component can be encoded with a constant number of bits of precision, this is also up to logarithmic factors the number of bits needed for communication. For this reason we have used k to refer to both the number of bits allowed in the distributed statistical estimation framework, and the number of gradient components that are communicated in distributed training.

The convergence of distributed SGD with sparsification operators top-r and random-k has been analyzed in the prior literature. Together with some form of error compensation, these methods have been shown to converge as fast as vanilla SGD in [26], [27], [37]. Therefore, when applying rTop-k sparsification to distributed SGD we also employ error compensation. The algorithm we use for our simulations in the subsequent section is given in Algorithm 1. In the next subsection, we prove that Algorithm 1 converges at the same rate as vanilla SGD for smooth but non-convex functions by building on the results of [26]. We note that this convergence result differs somewhat from the standard proofs of SGD convergence in that the gradient estimates can be unbiased, as is the case for rTop-k, however this does not affect the convergence of the algorithm when error compensation is employed.

A. Convergence of rTop-k

In this section we demonstrate that a result from [26] implies the convergence of Algorithm 1 to a fixed point under certain assumptions on the empirical risk. For nodes $i = 1, \ldots, n$, let $\mathcal{D}_i$ be the local dataset at node i. Further define

$$f^{(i)}(\omega) = \mathbb{E}_{l \sim \mathcal{D}_i}[\ell(\omega; X_l)]$$

to be the local loss function at node i where $\mathbb{E}_{l \sim \mathcal{D}_i}$ denotes expectation over a random sample l chosen from $\mathcal{D}_i$. We consider minimization of the empirical risk

$$f(\omega) = \frac{1}{n} \sum_{i=1}^{n} f^{(i)}(\omega)$$

using Algorithm 1.

We make the following assumptions:

Algorithm 1 SGD with rTop-k and Error Compensation

Hyperparameters: number of components communicated k, subsampling ratio r/k, learning rate η, minibatch size B
Inputs: local datasets $\mathcal{D}_i \quad i = 1, \ldots, n$
Output: weights ω^T

Broadcast randomly initialized weights, ω^0, to all nodes.
$m_i^0 \leftarrow 0$
for $t = 0, \ldots, T-1$ **do**
  On Nodes:
  for $i = 1, \ldots, n$ **do**
    receive ω^t from the centralized processor.
    $g_i^t \leftarrow \frac{1}{B}\sum_{j=1}^{B} \nabla \ell(\omega_t; X_j^{(i)})$; $\{X_j^{(i)}\}_{j=1}^{B}$ is uniformly chosen from $\mathcal{D}_i$
    $g_i^t \leftarrow g_i^t + m_i^t$
    $\hat{g}_i^t \leftarrow \mathsf{rTop}_k\left(g_i^t\right)$
    $m_i^{t+1} \leftarrow g_i^t - \hat{g}_i^t$
    send $\hat{g}_i^t$ to the centralized processor.
  end for
  On Centralized Processor:
  receive $\hat{g}_i^t$'s from n nodes.
  $\hat{g}^t \leftarrow \frac{1}{n}\sum_{i=1}^{n} \hat{g}_i^t$
  $\omega^{t+1} \leftarrow \omega^t - \eta \cdot \hat{g}^t$
  broadcast ω^{t+1} to all nodes.
end for

$\mathsf{rTop}_k\left(g_i^t\right)$:
  $I \leftarrow$ (indices of the r largest mag. components of g_i^t)
  $I \leftarrow$ (a random subset of k out of the r elements of I)
  $\hat{g}_i^t \leftarrow$ (g_i^t with indices not in I set to 0)
  return $\hat{g}_i^t$

(i) **Smoothness:** The local loss functions $f^{(i)} : \mathbb{R}^d \to \mathbb{R}$ are L-smooth for each $i = 1, \ldots, n$ in the sense that

$$f^{(i)}(\omega) \leq f^{(i)}(\omega') + \langle \nabla f^{(i)}(\omega'), \omega - \omega' \rangle + \frac{L}{2}\|\omega - \omega'\|_2^2$$

for any $\omega, \omega' \in \mathbb{R}^d$.

(ii) **Bounded second moment of gradients:** For each $i = 1, \ldots, n$ and $\omega \in \mathbb{R}^d$, and some constant $0 \leq G < \infty$, we have

$$\mathbb{E}_{l \sim \mathcal{D}_i} \|\nabla \ell(\omega; X_l)\|_2^2 \leq G^2 \, .$$

This also implies a bound on the variance $\sigma_i^2 = \mathbb{E}_{l \sim \mathcal{D}_i} \|\nabla \ell(\omega; X_l) - \nabla f^{(i)}(\omega)\|_2^2 \leq G^2$.

Below, we show that the rTop_k operator in Definition 3 is a "compression operator," and that this implies certain convergence guarantees for the empirical-risk minimization. We note that the convergence of distributed SGD with compression operators has been established in the literature for strongly convex [26], [27] and smooth but non-convex [26] loss functions under the bounded second moment assumption for the gradients (Assumption (ii) above). Assumption (ii) is a standard assumption in [26], [27], [38]–[41]. However, we note that the strongly convex assumption contradicts with Assumption (ii) if the domain of the optimization problem is assumed to be all of $\mathbb{R}^d$ as is done in [27]. Assumption (ii) has been subsequently removed in follow-up work [37]. In this paper, we focus on the non-convex case where Assumption (ii) is standard and does not lead to a contradiction.

Definition 4 (Compression operator)**.** *A (potentially randomized) function $\mathsf{Comp}_k : \mathbb{R}^d \to \mathbb{R}^d$ is a compression operator if there exists a constant $\gamma \in (0, 1]$ (potentially depending on k and d) such that*

$$\mathbb{E}_C \|\omega - \mathsf{Comp}_k(\omega)\|_2^2 \leq (1 - \gamma)\|\omega\|_2^2$$

for each $\omega \in \mathbb{R}^d$, where the expectation is over the randomness in the operator.

Proposition 1. *rTop_k is a compression operator with $\gamma = k/d$.*

Proof. Let $\omega_1, \ldots, \omega_d$ be an ordering of the d components of ω such that

$$|\omega_1| \geq |\omega_2| \geq \ldots \geq |\omega_d| \, .$$

Then

$$\begin{aligned}
\mathbb{E}_C \|\omega - \mathsf{rTop}_k(\omega)\|_2^2 &= \mathbb{E}_C \left[\sum_{j=1}^{d} (\omega_j - (\mathsf{rTop}_k(\omega))_j)^2\right] \\
&= \left(1 - \frac{k}{r}\right) \sum_{j=1}^{r} \omega_j^2 + \sum_{j=r+1}^{d} \omega_j^2 \\
&\leq \sum_{j=1}^{d} \omega_j^2 - \frac{k}{r}\frac{r}{d}\sum_{j=1}^{d} \omega_j^2 \\
&= \left(1 - \frac{k}{d}\right) \|\omega\|_2^2 \, .
\end{aligned}$$

□

Using this proposition, we get the below convergence result that follows as a special case from Theorem 1 in [26]. Let $f^* = \min_{\omega \in \mathbb{R}^d} f(\omega)$.

Theorem 3. *Under assumptions (i) and (ii) above, let the learning rate be set to $\eta = \frac{\widehat{C}}{\sqrt{T}}$ where $\widehat{C}$ is a constant that satisfies $\frac{\widehat{C}}{\sqrt{T}} \leq \frac{1}{2L}$. Let the ω^t be generated according to Algorithm 1. Then*

$$\begin{aligned}
\mathbb{E}\|\nabla f(z_T)\|_2^2 \leq &\left(\frac{\mathbb{E}[f(\omega^0)] - f^*}{\widehat{C}} + \widehat{C}L\left(\frac{G^2}{Bn}\right)\right)\frac{4}{\sqrt{T}} \\
&+ 8\left(4\frac{\left(1 - \frac{k^2}{d^2}\right)}{\frac{k^2}{d^2}} + 1\right)\frac{\widehat{C}^2 L^2 G^2}{T}
\end{aligned}$$

where z_T is a random variable that is sampled uniformly from the ω^t each with probability $1/T$.

Theorem 3 is exactly the same as Theorem 1 in [26], except with $H = 1$, $\gamma = k/d$, and using $\sum_{i=1}^{n} \sigma_i^2 \leq nG^2$. Additionally, since we aggregate the parameter updates on every time step, the iterates ω^t are the same at every node and the random variable z_T only varies over T possible values instead of nT possible values. Corollary 2 in [26] clarifies the order of these terms, and we reproduce the corresponding result below for convenience.

Corollary 1. *Suppose* $\mathbb{E}[f(\omega^0)] - f^* \leq J^2$ *and pick* $\widehat{C}^2 = \frac{Bn(\mathbb{E}[f(\omega^0)]-f^*)}{G^2 L}$ *which will satisfy* $\frac{\widehat{C}}{\sqrt{T}} \leq \frac{1}{2L}$ *if* T *is sufficiently large. Then*

$$\mathbb{E}\|\nabla f(z_T)\|_2^2 \leq O\left(\frac{JG}{\sqrt{BnT}}\right) + O\left(\frac{J^2 Bnd^2}{k^2 T}\right) .$$

Theorem 3 provides non-asymptotic guarantees for Algorithm 1, where we observe that sparsification does not affect the first order term. Here, we are required to decide the horizon T before running the algorithm. Therefore, in order for the algorithm to converge to a fixed point, the learning rate needs to follow a piecewise schedule (i.e. the learning rate would have to be reduced once in a while throughout the training process), which is what we do in our experiments in the next section. By similarly building on results of [26] (Theorem 2), one can bound the convergence rate of Algorithm 1 with a decaying learning rate as well as when nodes perform local iterations.

IV. Experiments

A. Experiment Settings

We validate our approach over a wide range of experiments including both image and language domains with CIFAR-10, ImageNet, and Penn Treebank (PTB) datasets and using two training methods where each node communicates the gradient vector after local training: (1) on one batch of the local data (distributed setting) and (2) for one epoch over the local data (i.e. one iteration over the local data adapting the federated setting in [7], [8]). Note that in the federated setting the number of communication rounds (per epoch) is much smaller than that in the distributed setting. In the federated setting one epoch always corresponds to one communication round while in the distributed setting we perform multiple communications in each epoch (78 communication rounds/epoch for CIFAR-10 and 265 for PTB). In our CIFAR-10 and ImageNet experiments, the dataset is distributed to each user in an i.i.d. fashion, which corresponds to a homogeneous data distributions among the nodes. On the other hand, the experiment on PTB dataset assigns one chapter of the corpus to each user, resulting in a more heterogeneous scenario for the data distributions among the nodes. We have 5 nodes in our experiments.

In Algorithm 1, we combine rTop-k with distributed SGD and error compensation. Here, we add the error, i.e. the gradients that are not communicated in the current iteration, to the gradients computed in the next iteration so that all important gradients are communicated eventually as in [1]. We fix the hyperparameter r such that $k/r = 1/n$ in all of our experiments. This choice is motivated by the following intuition: if the gradients corresponding to a given parameter are among the top k largest magnitude gradients at all n nodes, this choice ensures that in expectation this parameter will be updated by one node, since each node chooses to update this parameter with probability $k/r = 1/n$. As an example, a compression ratio of 99.9% requires only $k = 0.1\%$ of the entries of each gradient vector to be communicated by the corresponding node. For $n = 5$ nodes this is achieved by taking $\frac{r}{d} = 0.5\%$ of the entries of each gradient vector (d is the number of parameters in the model), which have the largest magnitudes, and then communicating a random subset of $\frac{k}{r} = \frac{1}{n} = 20\%$ of them. The central node calculates the global update vector by averaging the updates it receives for each component.

We compare our results with the baseline setting where there is no compression, the setting proposed in [1], which uses the gradient top-k selection method, and the random-k sparsification strategy where the gradients to be communicated by each node are chosen uniformly at random [9]. We employ the warm-up strategy and exponentially increase the sparsity in the warm-up period as in [1]. We further employ the local gradient accumulation strategy in [1], which provides substantial improvement in the performance for all sparsification methods. We do not compare our strategy to stochastic quantization techniques such as [6] as our main goal is to compare the performance of different sparsification methods. Also, [27] observes that top-k selection outperforms stochastic quantization in experiments and theoretically it can reduce the communication cost to at most order $\sqrt{d}$ bits, while sparsification methods can achieve order $k \log d$, where d is the number of parameters.

B. Image Domain

For image classification tasks, we trained ResNet-18 [42] on CIFAR-10 [43] and ResNet-34 [42] on ImageNet [44]. CIFAR-10 dataset consists of 50,000 training images, 5000 for each class, and 10000 test images, 1000 for each class. ImageNet dataset contains over 1 million training images and 50,000 validation images in 1000 classes. In all experiments, we use momentum SGD as the optimizer and cross entropy as the loss function. We split the data into batches of size 128 for CIFAR-10 and 32 for ImageNet experiments. We set the warm-up period as 5 epochs.

In our first two experiments, we train ResNet-18 on CIFAR-10 in both distributed and federated settings and show the results in Figure 2 and Figure 3 respectively. Figure 4 shows the results of training ResNet-34 on ImageNet dataset. Figures depict the performance of baseline with no compression and our rTop-k strategy with compression ratios 99% and 99.9%. To compare our method with [1], we present the performance of top-k strategy with compression ratios 99% and 99.9%. Final accuracies and compression ratios are summarized in Tables I, II, and III.

We note that in all experiments rTop-k strategy has substantially better performance compared to top-k and random-k strategies under the same compression ratio. In the federated settings in Table II and III, we observe that the accuracy of rTop-k with 99.9% compression ratio is better than the performance of top-k 99% compression ratio, corresponding to more than an order of magnitude improvement in compression ratio. Another interesting observation is that under 99% compression ratio rTop-k outperforms the baseline.

C. Language Domain

We use the Penn Treebank corpus (PTB) dataset, which consists of 923,000 training, 73,000 validation and 82,000

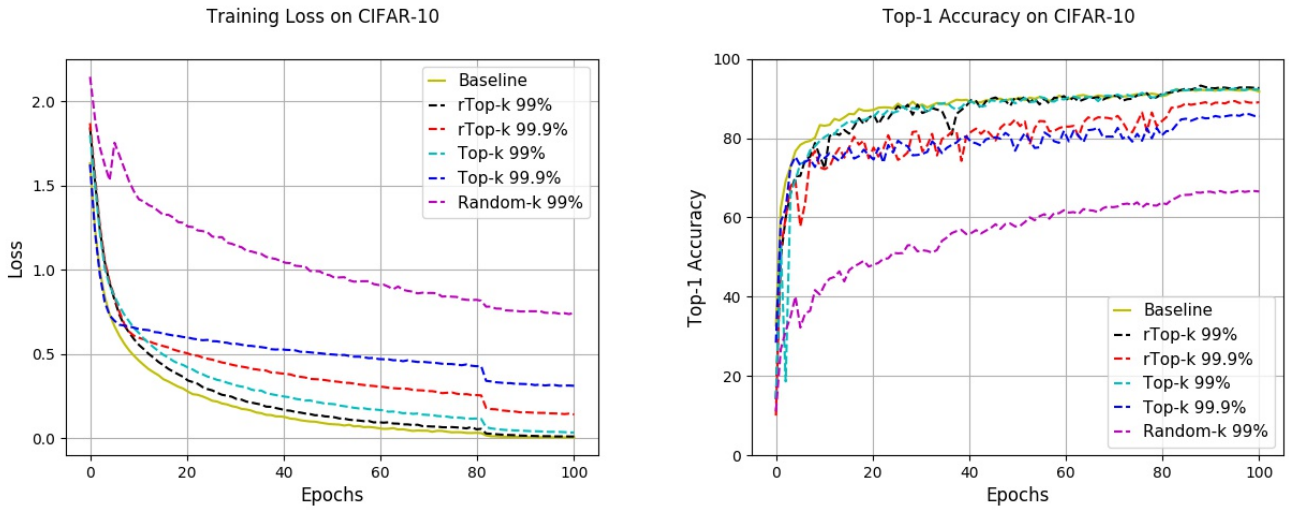

Fig. 2. Training Loss and Top-1 Test Accuracy on CIFAR-10 (distributed setting).

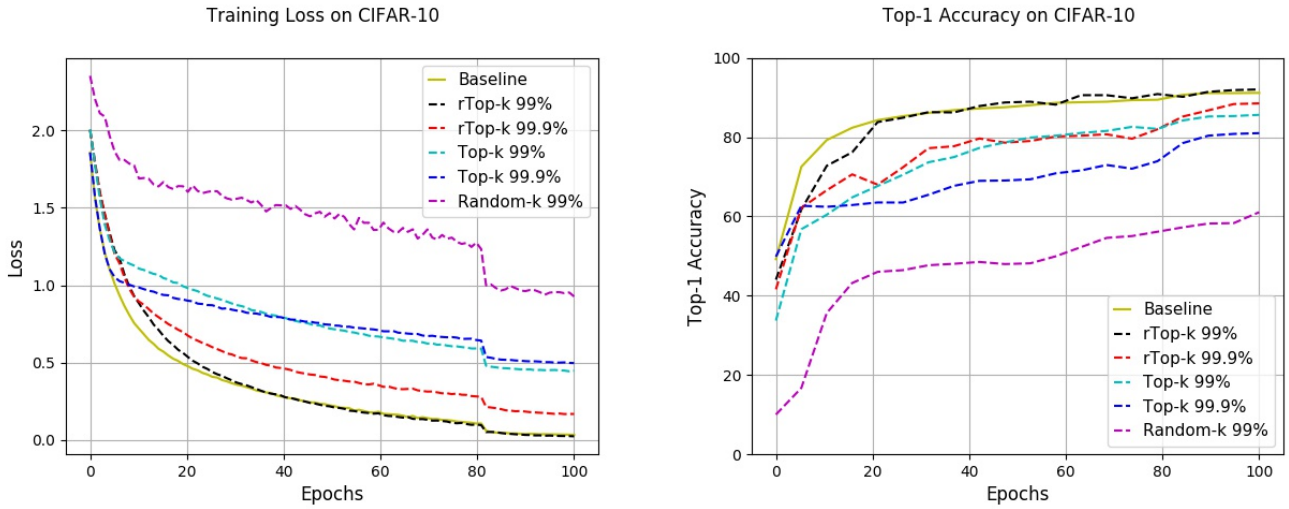

Fig. 3. Training Loss and Top-1 Test Accuracy on CIFAR-10 (federated setting).

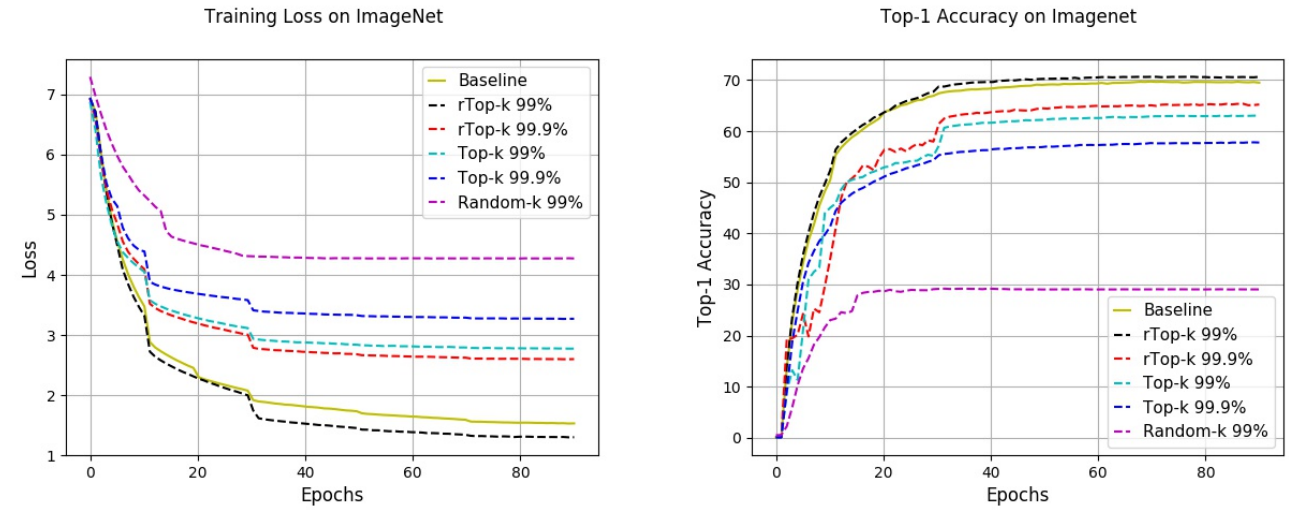

Fig. 4. Training Loss and Top-1 Test Accuracy on ImageNet (federated setting).

test words [45]. We train the 2-layer LSTM language model architecture with 1500 hidden units per layer [46] and tie the input and the output embeddings [47]. We use the same train/validation/test set split and vocabulary as [48]. We use

TABLE I
RESNET-18 TRAINED ON CIFAR-10 (DISTRIBUTED SETTING).

Method	Top-1 Accuracy	Compression
Baseline	92.40%	-
rTop-k	**93.25%**	99%
rTop-k	89.34%	99.9%
Top-k	**92.46%**	99%
Top-k	86.12%	99.9%
Random-k	66.81%	99%

TABLE II
RESNET-18 TRAINED ON CIFAR-10 (FEDERATED SETTING).

Method	Top-1 Accuracy	Compression
Baseline	91.16%	-
rTop-k	**92.02%**	99%
rTop-k	88.51%	99.9%
Top-k	85.62%	99%
Top-k	81.00%	99.9%
Random-k	61.07%	99%

TABLE III
RESNET-34 TRAINED ON IMAGENET (FEDERATED SETTING).

Method	Top-1 Accuracy	Compression
Baseline	69.70%	-
rTop-k	**70.63%**	99%
rTop-k	65.37%	99.9%
Top-k	63.06%	99%
Top-k	57.80%	99.9%
Random-k	29.19%	99%

vanilla SGD with gradient clipping. We use the same hyper-parameters (i.e. weight initialization, learning rate schedule, batch size) as in [49]. We set the warm-up period to 5 epochs.

In our first experiment, we study the distributed setting where each node communicates the gradient vector after the forward-backward pass on a single batch of local data. Figure 5 shows the perplexity and training loss of the trained language model in this setting.

TABLE IV
TRAINING RESULTS OF LANGUAGE MODELING ON PTB DATASET (DISTRIBUTED SETTING).

Method	Perplexity	Compression
Baseline	84.63	-
rTop-k	**82.49**	99.9%
Top-k	91.84	99.9%
Top-k	**84.31**	99%
Random-k	281.61	99%

In the second experiment, we study the federated setting where each node communicates the gradient vector after local training for one epoch over the local data. Figure 6 shows the perplexity and training loss of the trained language model in this setting. We did not use the local gradient accumulation strategy in this setting since we did not observe an improvement in the performance.

In our first experiment, we observe from Figure 5 that the validation perplexity of our rTop-k strategy matches the baseline with 99.9% compression ratio. The top-k strategy achieves the same performance with 99% compression ratio. We also note that the random-k strategy has substantially worse performance in this experiment. A similar set of results is obtained in the second experiment with slightly less aggressive compression levels compared to the first experiment as shown in Figure 6. In all cases, the rTop-k strategy has substantially better performance compared with the other methods at the same compression ratio. In both experiments, we observe from training losses and the corresponding test results (Figure 5 and Table IV for the first experiment and Figure 6 and Table V for the second experiment) that the rTop-k strategy obtains a better perplexity compared to baseline while its training loss is slightly larger. This suggests that training with rTop-k strategy results in better generalization.

TABLE V
TRAINING RESULTS OF LANGUAGE MODELING ON PTB DATASET (FEDERATED SETTING).

Method	Perplexity	Compression
Baseline	82.14	-
rTop-k	**82.02**	95%
Top-k	97.05	95%
Top-k	**81.97**	75%
Random-k	130.91	95%

V. PROOF OF THEOREM 1

To prove the desired upper bound, we describe an explicit encoding scheme $\Pi_i : \{0,1\}^d \to \{0,1\}^k$ and estimator $\hat{\theta}$ that is a function of the $M_i = \Pi_i(X_i)$ that achieves at most this error. The encoding functions Π_i are defined as follows:

(i) The first $\log d$ bits of $M_i = \Pi_i(X_i)$ encode the number of nonzero entries in X_i, i.e., $\|X_i\|_1$.

(ii) We form a codebook for the remaining $k - \log d$ bits such that each element of $\{0,1\}^d$ with at most k' ones maps to a unique $k - \log d$ bit string. Note that we can set $k' \geq \frac{k - \log d}{\log d}$.

(iii) Take X_i, and if $\|X_i\|_1 > k'$, we form $\tilde{X}_i$ by uniformly at random keeping only k' out of the original $\|X_i\|_1$ ones. If $\|X_i\|_1 \leq k'$, then $\tilde{X}_i = X_i$. We encode $\tilde{X}_i$ using the codebook from step (ii).

With this encoding scheme, the estimator $\hat{\theta}$ has access to both $\tilde{X}_i$ and $\|X_i\|_1$ for $i = 1, \ldots, n$. For convenience, define the subsampling fraction S_i by

$$S_i = \begin{cases} \frac{k'}{\|X_i\|_1} & \text{if } \|X_i\|_1 > k' \\ 1 & \text{otherwise .} \end{cases}$$

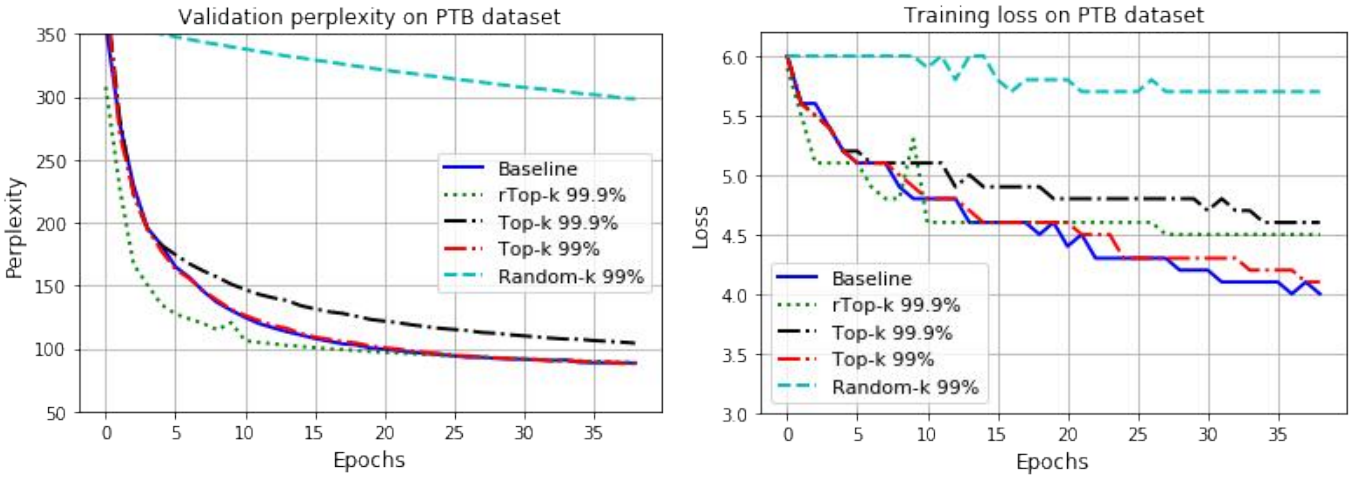

Fig. 5. Perplexity and training loss of LSTM language model on PTB dataset (distributed setting).

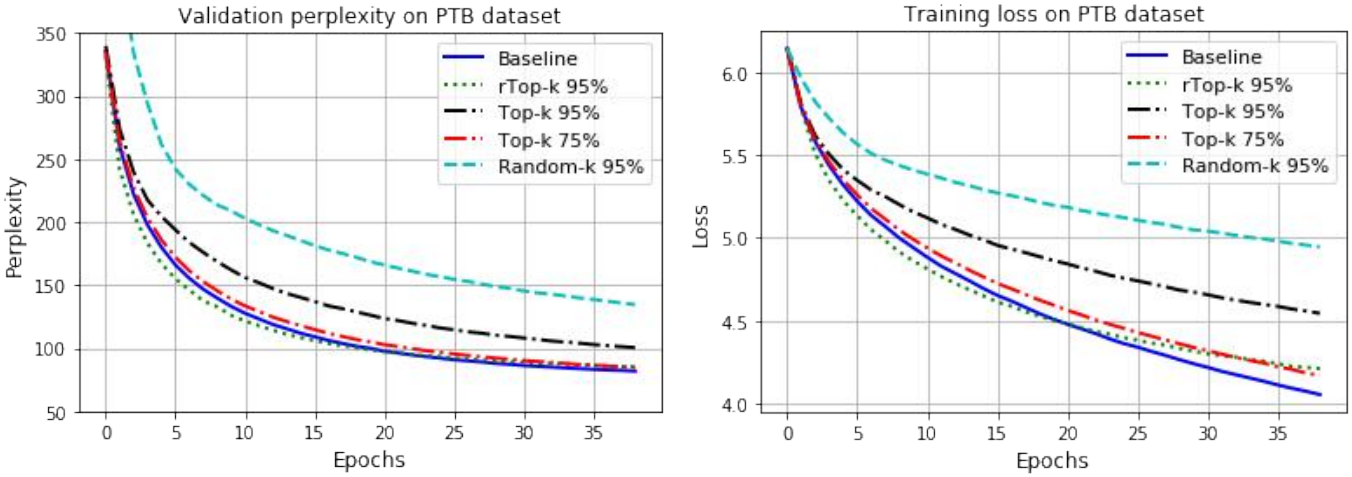

Fig. 6. Perplexity and training loss of LSTM language model on PTB dataset (federated setting).

Our estimator $\hat{\theta}$ is now defined by $\hat{\theta} = \frac{1}{n}\sum_{i=1}^{n}\frac{\tilde{X}_i}{S_i}$. This estimator is unbiased in that

$$\begin{aligned}
\mathbb{E}[\hat{\theta}] &= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[\frac{\tilde{X}_i}{S_i}\right] \\
&= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[\mathbb{E}\left[\frac{\tilde{X}_i}{S_i}\middle| S_i\right]\right] \\
&= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[\mathbb{E}\left[X_i|S_i\right]\right] \\
&= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[X_i\right] \\
&= \theta\ .
\end{aligned}$$

Finally, we compute the error:

$$\begin{aligned}
\mathbb{E}[(\hat{\theta}_j-\theta_j)^2] =& \mathbb{E}[\hat{\theta}_j^2]-\theta_j^2 \\
=& \frac{1}{n^2}\sum_{i=1}^{n}\mathbb{E}\left[\frac{\tilde{X}_{i,j}^2}{S_i^2}\right] \\
&+\frac{1}{n^2}\sum_{i\neq k}\mathbb{E}\left[\frac{\tilde{X}_{i,j}\tilde{X}_{k,j}}{S_iS_k}\right]-\theta_j^2 \\
=& \frac{1}{n^2}\sum_{i=1}^{n}\mathbb{E}\left[\mathbb{E}\left[\frac{\tilde{X}_{i,j}}{S_i^2}\middle| S_i\right]\right] \\
&+\frac{1}{n^2}\sum_{i\neq k}\mathbb{E}\left[\frac{\tilde{X}_{i,j}}{S_i}\right]\mathbb{E}\left[\frac{\tilde{X}_{k,j}}{S_k}\right]-\theta_j^2 \\
=& \frac{1}{n^2}\sum_{i=1}^{n}\mathbb{E}\left[\mathbb{E}\left[\frac{X_{i,j}}{S_i}\middle| S_i\right]\right] \\
&+\frac{n(n-1)}{n^2}\theta_j^2-\theta_j^2 \\
\leq& \frac{1}{n^2}\sum_{i=1}^{n}\mathbb{E}\left[\mathbb{E}\left[\frac{X_{i,j}}{S_i}\middle| S_i\right]\right]\ ,
\end{aligned}$$

and then summing over each component j,

$$\begin{aligned}\mathbb{E}\|\hat{\theta}-\theta\|_2^2 &\le \frac{1}{n^2}\sum_{i=1}^{n}\mathbb{E}\left[\mathbb{E}\left[\frac{\|X_i\|_1}{S_i}\middle|S_i\right]\right]\\ &\le \frac{1}{n^2}\sum_{i=1}^{n}\left(\mathbb{E}\left[\frac{\|X_i\|_1^2}{k'}\right]+k'\right) \qquad (4)\\ &\le \frac{1}{n^2}\sum_{i=1}^{n}\left(\mathbb{E}\left[\frac{\|X_i\|_1^2}{k'}\right]+C_1 s\right) \qquad (5)\\ &\le C_2\frac{s^2\log d}{nk}\,. \qquad (6)\end{aligned}$$

In the displays above, (4) follows by separating out the cases $S_i = 1$ and $S_i < 1$. In any case where $S_i < 1$, the ratio inside the conditional expectation is exactly k', and in any case where $S_i = 1$, the fraction $\frac{\|X_i\|_1}{S_i} = \|X_i\|_1 \le k'$. The step in (5) follows because $k' \le C_1\frac{k}{\log d}$ and we are assuming $\frac{k}{\log d} \le s$, and (6) uses the second moment formula for a Poisson binomial distribution.

VI. Proof of Theorem 2

To prove Theorem 2, we use part of Theorem 3 from [50] (see also [36]), reproduced below, which is proved using a Fisher information argument. Recall that in the context of Fisher information, the score function vector is the gradient of the log-likelihood, i.e $S_\theta(x) = \nabla_\theta \log f(x|\theta)$. Recall also that the Ψ_2-Orlicz norm of a random variable X is defined as

$$\|X\|_{\Psi_2} = \inf\{K \in (0,\infty) \mid \mathbb{E}[\Psi_2(|X|/K)] \le 1\}$$

where $\Psi_2(x) = \exp(x^2) - 1$, and that a random variable with finite Ψ_2-Orlicz norm is sub-Gaussian [51].

Theorem 4 (Barnes, Han, Özgür 2019)**.** *Suppose $\Theta = [-B, B]^d$. For any estimator $\hat{\theta}(M)$ and communication strategies Π_i, if $S_\theta(X)$ satisfies $\|\langle u, S_\theta(X)\rangle\|_{\Psi_2} \le N$ for any unit vector $u \in \mathbb{R}^d$, then*

$$\sup_{\theta\in\Theta}\mathbb{E}[\|\hat{\theta}-\theta\|^2] \ge \frac{d^2}{CN^2 n + \frac{d\pi^2}{B^2}}\,.$$

Proof of Theorem 2. The $\frac{s}{n}$ lower bound comes from considering the centralized case where there is no communication constraint. We will focus on the other bound, and will restrict our attention to a subset $\Theta' \subset \Theta$ and then use the fact that

$$\sup_{\theta\in\Theta}\mathbb{E}_\theta\|\hat{\theta}-\theta\|_2^2 \ge \sup_{\theta\in\Theta'}\mathbb{E}_\theta\|\hat{\theta}-\theta\|_2^2\,.$$

In particular, let $\Theta' = \left[\frac{s}{2d}, \frac{s}{d}\right]^d$. The distribution $f(x|\theta)$ is the product of Bernoulli distributions with parameters θ_i, so the score function for each component θ_i is

$$S_{\theta_i}(x_i) = \frac{\partial}{\partial\theta_i}\log f(x_i|\theta_i) = \begin{cases}\frac{1}{\theta_i} & \text{, if } x_i = 1\\ \frac{-1}{1-\theta_i} & \text{, if } x_i = 0\,.\end{cases}$$

If we set

$$N = \max\left\{\frac{1}{\theta_i\sqrt{\log\frac{1}{\theta_i}}}\,,\ \frac{1}{(1-\theta_i)\sqrt{\log\frac{1}{(1-\theta_i)}}}\right\}\,,$$

then in can be checked that

$$\begin{aligned}\mathbb{E}\left[e^{\left(\frac{S_{\theta_i}(X_i)}{N}\right)^2}\right] &= \theta_i e^{\left(\frac{1}{\theta_i N}\right)^2} + (1-\theta_i)e^{\left(\frac{1}{(1-\theta_i)N}\right)^2}\\ &\le 2\end{aligned}$$

and thus $\|S_{\theta_i}(X_i)\|_{\Psi_2} \le N$. By taking sums of scaled independent sub-Gaussian random variables [51] we get

$$\begin{aligned}\|\langle u, S_\theta(X)\rangle\|_{\Psi_2} \le& c_1\max\left\{\frac{1}{\theta_i\sqrt{\log\frac{1}{\theta_i}}}\,,\right.\\ &\left.\frac{1}{(1-\theta_i)\sqrt{\log\frac{1}{(1-\theta_i)}}}\right\}\\ &\le c_1\frac{1}{\theta_i\sqrt{\log\frac{1}{\theta_i}}} \le \frac{c_2 d}{s\sqrt{\log\frac{d}{s}}}\end{aligned}$$

where in the last line we have used the fact that $x^2\log x \ge (1-x)^2\log(1-x)$ for $0 \le x \le \frac{1}{2}$ and $s \le \frac{d}{2}$ so that $\frac{s}{d} \le \frac{1}{2}$. Then by Theorem 4 above,

$$\begin{aligned}\sup_{\theta\in\Theta'}\mathbb{E}_\theta\|\hat{\theta}-\theta\|_2^2 &\ge \frac{d^2}{c_3 nk\frac{d^2}{s^2\log\frac{d}{s}} + c_4\frac{d^3}{s^2}}\\ &\ge c_5\frac{s^2\log\frac{d}{s}}{nk}\end{aligned}$$

where the last inequality uses $nk \ge d\log\frac{d}{s}$. □

References

[1] Y. Lin, S. Han, H. Mao, Y. Wang, and W. J. Dally, "Deep gradient compression: Reducing the communication bandwidth for distributed training," *6th International Congress on Learning Representations (ICLR)*, 2018.

[2] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *arXiv preprint arXiv:1912.04977*, 2019.

[3] M. Li, D. G. Andersen, A. Smola, and K. Yu, "Communication efficient distributed machine learning with the parameter server," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'14. Cambridge, MA, USA: MIT Press, 2014, pp. 19–27.

[4] F. Seide, H. Fu, J. Droppo, G. Li, and D. Yu, "1-bit stochastic gradient descent and application to data-parallel distributed training of speech dnns," in *Interspeech 2014*, September 2014.

[5] N. Strom, "Scalable distributed dnn training using commodity gpu cloud computing," in *INTERSPEECH*, 2015.

[6] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "Qsgd: Communication-efficient sgd via gradient quantization and encoding," in *Advances in Neural Information Processing Systems 30*, 2017, pp. 1709–1720.

[7] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *AISTATS*, 2017.

[8] J. Konecný, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," *CoRR*, vol. abs/1610.02527, 2016.

[9] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtarik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," in *NIPS Workshop on Private Multi-Party Machine Learning*, 2016.

[10] A. F. Aji and K. Heafield, "Sparse communication for distributed gradient descent," *arXiv preprint arXiv:1704.05021*, 2017.

[11] W. Wen, C. Xu, F. Yan, C. Wu, Y. Wang, Y. Chen, and H. Li, "Terngrad: Ternary gradients to reduce communication in distributed deep learning," in *NIPS*, 2017.

[12] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar, "signsgd: compressed optimisation for non-convex problems," in *ICML*, 2018.

[13] H. Wang, S. Sievert, S. Liu, Z. B. Charles, D. S. Papailiopoulos, and S. Wright, "Atomo: Communication-efficient learning via atomic sparsification," in *NeurIPS*, 2018.

[14] J. Wangni, J. Wang, J. Liu, and T. Zhang, "Gradient sparsification for communication-efficient distributed optimization," in *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 1299–1309.

[15] V. Gandikota, R. K. Maity, and A. Mazumdar, "vqsgd: Vector quantized stochastic gradient descent," *arXiv preprint arXiv:1911.07971*, 2019.

[16] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *in COMPSTAT*, 2010.

[17] B. Recht, C. Re, S. Wright, and F. Niu, "Hogwild!: A lock-free approach to parallelizing stochastic gradient descent," in *Advances in Neural Information Processing Systems*, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Q. Weinberger, Eds., vol. 24. Curran Associates, Inc., 2011, pp. 693–701. [Online]. Available: https://proceedings.neurips.cc/paper/2011/file/218a0aefd1d1a4be65601cc6ddc1520e-Paper.pdf

[18] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Transactions on Automatic Control*, vol. 31, no. 9, pp. 803–812, 1986.

[19] A. Elgabli, J. Park, A. S. Bedi, M. Bennis, and V. Aggarwal, "Q-gadmm: Quantized group admm for communication efficient decentralized machine learning," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 8876–8880.

[20] J. Wang, A. K. Sahu, Z. Yang, G. Joshi, and S. Kar, "Matcha: Speeding up decentralized sgd via matching decomposition sampling," in *2019 Sixth Indian Control Conference (ICC)*, 2019, pp. 299–300.

[21] O. Shamir, N. Srebro, and T. Zhang, "Communication-efficient distributed optimization using an approximate newton-type method," in *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ser. ICML'14. JMLR.org, 2014, p. II–1000–II–1008.

[22] M. Jahani, X. He, C. Ma, A. Mokhtari, D. Mudigere, A. Ribeiro, and M. Takác, "Efficient distributed hessian free algorithm for large-scale empirical risk minimization via accumulating sample strategy," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 2634–2644.

[23] J. Ba, R. Grosse, and J. Martens, "Distributed second-order optimization using kronecker-factored approximations," 2016.

[24] R. Crane and F. Roosta, "Dingo: Distributed newton-type method for gradient-norm optimization," in *Advances in Neural Information Processing Systems*, 2019, pp. 9498–9508.

[25] M. Jahani, M. Nazari, S. Rusakov, A. S. Berahas, and M. Takáč, "Scaling up quasi-newton algorithms: Communication efficient distributed sr1," *arXiv preprint arXiv:1905.13096*, 2019.

[26] D. Basu, D. Data, C. Karakus, and S. Diggavi, "Qsparse-local-sgd: Distributed sgd with quantization, sparsification and local computations," in *Advances in Neural Information Processing Systems*, 2019, pp. 14 668–14 679.

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

[28] Y. Zhang, J. Duchi, M. I. Jordan, and M. J. Wainwright, "Information-theoretic lower bounds for distributed statistical estimation with communication constraints," in *Advances in Neural Information Processing Systems*, 2013, pp. 2328–2336.

[29] A. Garg, T. Ma, and H. Nguyen, "On communication cost of distributed statistical estimation and dimensionality," in *Advances in Neural Information Processing Systems*, 2014, pp. 2726–2734.

[30] M. Braverman, A. Garg, T. Ma, H. L. Nguyen, and D. P. Woodruff, "Communication lower bounds for statistical estimation problems via a distributed data processing inequality," in *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*. ACM, 2016, pp. 1011–1020.

[31] A. Xu and M. Raginsky, "Information-theoretic lower bounds on bayes risk in decentralized estimation," *IEEE Transactions on Information Theory*, vol. 63, no. 3, pp. 1580–1600, 2017.

[32] Y. Han, A. Özgür, and T. Weissman, "Geometric lower bounds for distributed parameter estimation under communication constraints," in *Proceedings of Machine Learning Research*, 75, 2018, pp. 1–26.

[33] I. Diakonikolas, E. Grigorescu, J. Li, A. Natarajan, K. Onak, and L. Schmidt, "Communication-efficient distributed learning of discrete probability distributions," *Advances in Neural Information Processing Systems*, pp. 6394–6404, 2017.

[34] J. Acharya, C. L. Canonne, and H. Tyagi, "Inference under information constraints: Lower bounds from chi-square contraction," *Proceedings of Machine Learning Research vol*, vol. 99, pp. 1–15, 2019.

[35] L. P. Barnes, Y. Han, and A. Özgür, "A geometric characterization of fisher information from quantized samples with applications to distributed statistical estimation," *Proceedings of the 56th Annual Allerton Conference on Communication, Control, and Computing*, 2018.

[36] L. P. Barnes, Y. Han, and A. Ozgur, "Lower bounds for learning distributions under communication constraints via fisher information," *arXiv preprint arXiv:1902.02890*, 2019.

[37] S. U. Stich and S. P. Karimireddy, "The error-feedback framework: Better rates for sgd with delayed gradients and compressed communication," 2019.

[38] S. Shalev-Shwartz, Y. Singer, and N. Srebro, "Pegasos: Primal estimated sub-gradient solver for svm," 2007, pp. 807–814.

[39] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, "Robust stochastic approximation approach to stochastic programming," *SIAM Journal on Optimization*, vol. 19, pp. 1574–1609, 4 2009.

[40] F. Niu, B. Recht, C. Ré, and S. Wright, "Hogwild: A lock-free approach to parallelizing stochastic gradient," *NIPS*, vol. 24, pp. 693–701, 06 2011.

[41] D. Alistarh, T. Hoefler, M. Johansson, S. Khirirat, N. Kostantinov, and C. Renggli, "The convergence of sparsified gradient methods," *NeurIPS*, 2018.

[42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 770–778.

[43] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009.

[44] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[45] M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini, "Building a large annotated corpus of english: The penn treebank," *Comput. Linguist.*, vol. 19, no. 2, pp. 313–330, Jun. 1993.

[46] O. Press and L. Wolf, "Using the output embedding to improve language models," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 157–163.

[47] H. Inan, K. Khosravi, and R. Socher, "Tying word vectors and word classifiers: A loss framework for language modeling," *ArXiv*, vol. abs/1611.01462, 2016.

[48] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur, "Recurrent neural network based language model." in *INTERSPEECH*. ISCA, 2010, pp. 1045–1048.

[49] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," *CoRR*, vol. abs/1409.2329, 2014.

[50] L. P. Barnes, Y. Han, and A. Özgür, "Fisher information for distributed estimation under a blackboard communication protocol," *Proceedings of the IEEE International Symposium on Information Theory (ISIT), Paris, France*, 2019.

[51] R. Vershynin, "Introduction to the non-asymptotic analysis of random matrices," *arXiv preprint* arXiv*:1011.3027*, 2010.