ADAPTIVE STEP-SIZE METHODS FOR COMPRESSED SGD

Adarsh M. Subramaniam, Akshayaa Magesh, Venugopal V. Veeravalli

ECE Department and Coordinated Science Lab, University of Illinois at Urbana-Champaign
Email: adarshm2, amagesh2, vvv@illinois.edu

ABSTRACT

Compressed Stochastic Gradient Descent (SGD) algorithms have been proposed to address the communication bottleneck in distributed and decentralized optimization problems such as federated machine learning. Many existing compressed SGD algorithms use non-adaptive step-sizes (constant or diminishing) to provide theoretical convergence guarantees. Since non-adaptive step-sizes typically involve unknown system parameters, the step-sizes are fine-tuned in practice to obtain good empirical performance for the given dataset and learning model. Such fine-tuning might be impractical in many scenarios. Thus, it is of interest to study compressed SGD using adaptive step-sizes. Motivated by prior work that use adaptive step-sizes for uncompressed SGD, we develop an Armijo rule based step-size selection method for compressed SGD. In particular, we introduce a scaling technique for the descent step, which we use to establish order-optimal convergence rates for convex-smooth and strong convex-smooth objectives under an interpolation condition, and for non-convex objectives under a strong growth condition. We present experimental results on deep neural networks trained on real-world datasets, and compare the performance of our proposed algorithm with state-of-the-art compressed SGD methods to demonstrate improved performance at various levels of compression.

Index Terms—Adaptive step-size, stochastic gradient descent, compressed SGD, Armijo rule, scaling, federated learning.

1. INTRODUCTION

Consider the optimization problem of minimizing the sum of $n$ functions:

$$\min_x f(x) = \min_x \frac{1}{n} \sum_{i=1}^{n} f_i(x).$$

This formulation is widely used in machine learning, where $f_i$ is the loss function corresponding to the $i$-th datapoint, $n$ is the size of the training set, and $x$ represents the parameters of the learning model.

A practical approach to solving the optimization problem in (1) is Stochastic Gradient Descent (SGD), where one of the component functions $f_i$ is chosen at random during each iteration, and the iterate $x$ is updated using the gradient of $f_i$. In this paper, we consider the setting where compressed gradients are used in place of the actual gradients. The use of compressed gradients helps alleviate the communication bottleneck in distributed and decentralized optimization settings such as federated learning [1].

There have been several recent papers on using compressed SGD algorithms [2–6] to address the communication bottleneck, where only the top $k$ (say 1%) values of the components of the gradient are used in the iteration update. The convergence properties of such compressed gradient methods in the distributed and decentralized setting with feedback were studied in a number of subsequent papers [4, 7, 8].

Existing works on compressed SGD use diminishing or constant step-sizes based on system parameters to show convergence properties. In machine learning applications, the step-sizes are fine-tuned to the dataset and machine learning model to provide good empirical performance. However, such fine tuning can be impractical, and can result in suboptimal rates of convergence [9–11]. Hence, it is of interest to study adaptive step-size methods for compressed SGD.

Adaptive step-size techniques for (uncompressed) SGD have been studied in several recent works [12, 13]. An efficient implementation of the classical Armijo step-search method [14] to train neural networks has been proposed in [12]. In order to establish convergence, these works assume that the objective satisfies an interpolation condition. The condition states that there exists a point $x^*$ which minimizes all functions $f_i$ in the optimization problem (1). Several other recent works [15, 16] have used the interpolation condition to establish convergence of non-compressed SGD, and the use of this condition has also been justified by theoretical works [17] in the context of modern neural networks, where the number of parameters is much larger than the number of datapoints in the training dataset.

Motivated by adaptive step-size techniques for uncompressed SGD, we develop an adaptive step-size technique for compressed SGD that results in faster convergence at the
same compression level as compared to existing non-adaptive step-size compressed SGD algorithms. To this end, we develop a scaling technique, which is crucial to the convergence of our adaptive step-size compressed SGD algorithm. To the best of our knowledge, we are not aware of any other adaptive step-size methods for compressed SGD with feedback with theoretical guarantees. An extended version of our work with the complete proofs can be found in [18].

Our Contributions: We propose a computationally feasible and efficient adaptive step-size compressed SGD algorithm that uses the scaling technique and the biased topk operator for compression. Under the interpolation condition (see Definition 1), we prove that our proposed algorithm achieves a convergence rate of $O\left(\frac{1}{T}\right)$ when the objective function is convex and smooth, and $O\left(\beta^T\right)$ ($\beta < 1$) when the objective function is strongly-convex and smooth. We also show that under a strong growth condition (see Definition 2), the proposed algorithm achieves a convergence rate of $O\left(\frac{1}{\nu T}\right)$ in the non-convex setting. The above results hold for all values of the parameter $\sigma \in (0, 1)$ of the step-search.

We demonstrate that our algorithm for compressed SGD outperforms existing non-adaptive compressed SGD methods on neural network training tasks at an arbitrarily chosen compression rate of 1.5%, and show similar results for other rates of compression in [18].

2. PRELIMINARIES AND NOTATION

In this section, we define our notation, discuss the classical Armijo step-size search in Algorithm 1 (detailed explanation in [18]), the compression operator, and the conditions under which we prove our convergence results.

2.1. Notation

Let $x_t$ denote the iterate of the optimization algorithm at time $t$, and $i_t$ denote the datapoint or batch of data chosen at time $t$. The loss function of the batch $i_t$ chosen at time $t$ at an iterate $x_t$ is denoted by $f_{i_t}(x_t)$. The step-size used at time $t$ is denoted by $\eta_t$.

2.2. Armijo Step-Size Search in Gradient Descent

The algorithm proposed in this work makes use of a modified and efficient implementation of the classical Armijo step-size search method [14]. The pseudocode for Armijo step-size search is given in Algorithm 1.

2.3. Topk Compression Operator

In our compressed SGD algorithm, the gradient in each iteration is compressed using the topk compression operator [4,5]. Consider a vector $x \in \mathbb{R}^d$. Let $T_k$ be the set of indices of the $k$ elements of $x$ with maximum magnitude, and define $\gamma \triangleq \frac{k}{\sigma}$. The topk operator compresses a vector $x$ such that

$$
top_k(x)_i = \begin{cases} (x)_i & \text{if } i \in T_k \\ 0 & \text{otherwise} \end{cases}
$$

where $(x)_i$ and $top_k(x)_i$ are the $i$-th elements of the vectors.

2.4. Definitions

To establish the convergence of our algorithm, we require the following interpolation condition in the convex setting, and the strong growth condition in the non-convex setting.

Definition 1 (Interpolation [16]). A function $f(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x)$ satisfies the interpolation condition if

$$
\exists x^* \text{ s.t. } \nabla f_i(x^*) = 0, \forall i \in \{1, 2, \cdots, n\}.
$$

Definition 2 (Strong Growth Condition [12]). A function $f = \frac{1}{n} \sum_{i=1}^{n} f_i(x)$ satisfies the strong growth condition with constant $\nu$ if

$$
\frac{1}{n} \sum_{i=1}^{n} ||\nabla f_i(x)||^2 \leq \nu ||\nabla f(x)||^2.
$$

3. COMPRESSED SGD WITH ADAPTIVE STEP-SIZE

Our proposed compressed SGD algorithm is motivated by the idea of scaling, where the step-size returned by the Armijo step-size search is scaled by a factor $a$ in the descent step. We now discuss the intuition for the scaling, and propose an algorithm for compressed SGD with Armijo step-size search and scaling.

3.1. Armijo Step-Size Search with Scaling for GD

Optimization problems that arise in machine learning applications are usually asymmetric, and have gradients that do not point in the direction of a minimizer. For example, consider the function $f(x) = x_1^2 + \frac{x_2^2}{\beta}$. The negative gradient at a point is orthogonal to the tangent and does not necessarily point towards the minimizer $(0, 0)$. Hence, GD with Armijo step-size

---

**Algorithm 1** Armijo Step-Size Search

1: procedure ARMijo Step-Size Search($f, \alpha_{\text{max}}, x_t, t$)
2: $\alpha_t = \alpha_{\text{max}}$
3: repeat
4: $\alpha_t \leftarrow \alpha_t \rho$
5: $x_{t+1} \leftarrow x_t - \alpha_t \nabla f(x_t)$
6: until $f(x_{t+1}) \leq f(x_t) - \sigma \alpha_t ||\nabla f(x_t)||^2$
7: return $\alpha_t$

8: end procedure
Algorithm 2 Compressed SGD + Armijo Step-Size Search and Scaling (CSGD-ASSS)

1: for $t = 1, \cdots, T$ do
2: Sample batch $i_t$ of data
3: $\alpha_{\text{max}} = \omega \alpha_{t-1}$
4: $\alpha_t \leftarrow$ Armijo Step-Size Search($f_{i_t}, \alpha_{\text{max}}, x_t, t$)
5: $\eta_t = a \alpha_t$
6: $g_t = \text{topk}(m_t + \eta_t \nabla f_{i_t}(x_t))$
7: $x_{t+1} = x_t - g_t$
8: $m_{t+1} = m_t + \eta_t \nabla f_{i_t}(x_t) - \text{topk}(m_t + \eta_t \nabla f_{i_t}(x_t))$
9: end for

3.3. Convergence Analysis

We now study the convergence of the CSGD-ASSS algorithm in the convex and strong convex settings under the interpolation condition (3), and in the non-convex setting under the strong growth condition (2). These rates achieved under compression with the scaling technique are the same as in the uncompressed setting [12].

3.3.1. Main Results

Consider the minimization of the objective $f(x)$ in (1) with CSGD-ASSS algorithm. For the convex setting we prove the following convergence result under the interpolation condition.

**Theorem 1** (CSGD-ASSS convex). Let $f_i(x)$ be convex and $L_i$ smooth for $i \in [n]$, and assume that the interpolation condition (3) is satisfied. Then, there exists $\hat{a} > 0$ such that for $0 < a \leq \hat{a}$, the CSGD-ASSS algorithm with scaling $a$, parameter $\sigma \in (0, 1)$, and compression factor $\gamma = \frac{1}{\hat{a}}$, satisfies

$$E \left[ f \left( \frac{1}{T} \sum_{t=0}^{T-1} x_t \right) \right] - E[f(x^*)] \leq \frac{1}{\delta_1 T} (E[||x_0 - x^*||^2]),$$

where for any $0 < \epsilon < \zeta$, $\zeta \triangleq \frac{\sigma}{(2-\gamma)}$ and $L_{\text{max}} \triangleq \max_i L_i$,

$$\delta_1 \triangleq \rho \frac{2(1-\sigma)}{L_{\text{max}}},$$

$$\delta_2 \triangleq \left( 2a - \frac{a^2}{\sigma} - \frac{a^2}{\sigma p(\epsilon)} - (1 - \gamma) \left( 1 + \frac{1}{r(\epsilon)} \right)^2 \frac{a^2}{\sigma} \right),$$

$$\hat{a} \triangleq (\zeta - \epsilon).$$

The values of $p(\epsilon), r(\epsilon)$ are obtained from a convex problem described in [18].

Consider the minimization of the function $f(x)$ in the convex setting, with some $f_i(x)$ satisfying strong convexity with parameter $\mu_i > 0$. Then we can establish the following ‘linear’ convergence result.

**Algorithm 2** Compressed SGD + Armijo Step-Size Search and Scaling (CSGD-ASSS)

1: for $t = 1, \cdots, T$ do
2: Sample batch $i_t$ of data
3: $\alpha_{\text{max}} = \omega \alpha_{t-1}$
4: $\alpha_t \leftarrow$ Armijo Step-Size Search($f_{i_t}, \alpha_{\text{max}}, x_t, t$)
5: $\eta_t = a \alpha_t$
6: $g_t = \text{topk}(m_t + \eta_t \nabla f_{i_t}(x_t))$
7: $x_{t+1} = x_t - g_t$
8: $m_{t+1} = m_t + \eta_t \nabla f_{i_t}(x_t) - \text{topk}(m_t + \eta_t \nabla f_{i_t}(x_t))$
9: end for

3.3. Convergence Analysis

We now study the convergence of the CSGD-ASSS algorithm in the convex and strong convex settings under the interpolation condition (3), and in the non-convex setting under the strong growth condition (2). These rates achieved under compression with the scaling technique are the same as in the uncompressed setting [12].

3.3.1. Main Results

Consider the minimization of the objective $f(x)$ in (1) with CSGD-ASSS algorithm. For the convex setting we prove the following convergence result under the interpolation condition.

**Theorem 1** (CSGD-ASSS convex). Let $f_i(x)$ be convex and $L_i$ smooth for $i \in [n]$, and assume that the interpolation condition (3) is satisfied. Then, there exists $\hat{a} > 0$ such that for $0 < a \leq \hat{a}$, the CSGD-ASSS algorithm with scaling $a$, parameter $\sigma \in (0, 1)$, and compression factor $\gamma = \frac{1}{\hat{a}}$, satisfies

$$E \left[ f \left( \frac{1}{T} \sum_{t=0}^{T-1} x_t \right) \right] - E[f(x^*)] \leq \frac{1}{\delta_1 T} (E[||x_0 - x^*||^2]),$$

where for any $0 < \epsilon < \zeta$, $\zeta \triangleq \frac{\sigma}{(2-\gamma)}$ and $L_{\text{max}} \triangleq \max_i L_i$,

$$\delta_1 \triangleq \rho \frac{2(1-\sigma)}{L_{\text{max}}},$$

$$\delta_2 \triangleq \left( 2a - \frac{a^2}{\sigma} - \frac{a^2}{\sigma p(\epsilon)} - (1 - \gamma) \left( 1 + \frac{1}{r(\epsilon)} \right)^2 \frac{a^2}{\sigma} \right),$$

$$\hat{a} \triangleq (\zeta - \epsilon).$$

The values of $p(\epsilon), r(\epsilon)$ are obtained from a convex problem described in [18].

Consider the minimization of the function $f(x)$ in the convex setting, with some $f_i(x)$ satisfying strong convexity with parameter $\mu_i > 0$. Then we can establish the following ‘linear’ convergence result.
**Theorem 2** (CSGD-ASSS strong convex). Let $f_i$ be convex and $L_i$ smooth for all $i \in [n]$, and $\mu_i$ strongly convex with $\mu_i > 0$ for some $i \in [n]$. Assume that the interpolation condition (3) is satisfied. Then, there exists $\hat{\alpha} > 0$ such that for $0 < \alpha \leq \hat{\alpha}$, the CSGD-ASSS algorithm with scaling $a$ and $\sigma \in (0, 1)$ satisfies

$$E[\|x_{t} - x^*\|^2] \leq 2(\hat{\beta})^t E[\|x_0 - x^*\|^2],$$

for some $\hat{\beta} < 1$, where for all $0 < \epsilon < \zeta$,

$$\hat{\beta} \triangleq \max \{\beta_1(p(\epsilon), r(\epsilon)), (1 - \frac{E_t[\mu_i]}{L_{\max}}(1 - \sigma))^t\},$$

$$\beta_1(p(\epsilon), r(\epsilon)) \triangleq \left(\mu_{\max} a \alpha_{\max} + p(\epsilon) + (1 - \gamma)(1 + r(\epsilon))\right),$$

$$\hat{\alpha} = \zeta - \epsilon.$$

The values $p(\epsilon), r(\epsilon)$ and $\mu_{\max}$ are defined in [18].

To show convergence guarantees in the non-convex case, we use the strong growth condition (4).

**Theorem 3** (CSGD-ASSS non-convex). Let $f_i$ be non-convex, $L_i$ smooth and let $f_i, i \in [n]$ satisfy the strong growth condition (4). Then, there exists $\bar{\alpha}, \bar{\alpha}$ such that for $0 < \alpha \leq \bar{\alpha}$ and $\alpha_{\max} \leq \bar{\alpha}$,

$$\frac{1}{T} \sum_{t=0}^{T-1} E[\|\nabla f(x_t)\|^2] \leq \frac{E[f(x_0)] - E[f(\hat{x}_T)]}{\delta T},$$

where $\hat{x}_T$ is a perturbed iterate [19] obtained from $\{x_t\}_{t=1}^T$. A discussion on the choice of $\bar{\alpha}$, and $\bar{\alpha}$ is presented in [18].

**Remark 1.** Proofs for the uncompressed SGD setting [12] constrain the values of both $\alpha$ and $\alpha_{\max}$ in the non-convex setting, and the bounds are dependent on the function parameters $L_i$. Interestingly, the scaling technique allows us to eliminate the bound on $\alpha$.

**Remark 2.** Similar convergence results can be obtained in the distributed setting, as presented in [18].

4. EXPERIMENTAL RESULTS

We test our CSGD-ASSS algorithm on neural network architectures ResNet-18, ResNet-34 and DenseNet-121, on CIFAR-100 and CIFAR-10 datasets. Complete simulations can be found in [18]. We use a batch size of 64 and set the initial value of $\alpha_{\max} = 0.1$. With $\omega = 1.2$ and $\rho = 0.8$, in each subsequent iteration, the value of $\alpha_{\max}$ is updated as $\alpha_{\max} = \omega \alpha_{\max} - 1$, where $\omega_{\max}$ is the step-size returned by the Armijo step-size search (Step 4) in CSGD-ASSS in the previous iteration. This renders the algorithm computationally efficient for neural network training tasks, as discussed in our work in [18]. For the Armijo step-size search parameter $\alpha$, we use the value 0.1 as in uncompressed SGD with Armijo step-size search [12]. Motivated by the analysis of scaled GD presented in [18], we set the scaling factor $\alpha$ to be a multiple of $\sigma$. In our simulations, we set $\alpha = 3\sigma$ for all neural network training tasks.

We show in Figure 1, that at 1.5% compression ($\frac{c}{d} = 1.5\%$), the CSGD-ASSS algorithm outperforms non-adaptive SGD at commonly used step-sizes of 0.1, 0.05, 0.01. The reason for using the algorithm in [3] as a baseline for comparison is that prevalent non-adaptive compressed SGD algorithms build on the topk gradient compression operator with memory feedback introduced in [3]. In the distributed setting, we simulate ResNet-18, ResNet-34 and DenseNet-121 neural network architectures on CIFAR-10 dataset at 1.5% compression on 3 worker nodes. Figure 1d demonstrates the convergence of distributed CSGD-ASSS algorithm to a local minima.

5. CONCLUSION

We have motivated and presented a scaling technique for Armijo step-size search for compressed SGD, and used it to establish convergence in convex and non-convex settings. In our experiments, we have shown that the CSGD-ASSS algorithm outperforms non-adaptive compressed SGD on ResNet-18 and ResNet-34 networks trained on CIFAR-100 and CIFAR-10 datasets at 1.5% compression. We have also shown the convergence of the CSGD-ASSS algorithm to local minima in distributed settings [18].
6. REFERENCES

[1] P. Kairouz, H. B. McMahan, B. Avent, and et.al, “Advances and open problems in federated learning,” CoRR, vol. abs/1912.04977, 2019. [Online]. Available: http://arxiv.org/abs/1912.04977

[2] N. Dryden, T. Moon, S. A. Jacobs, and B. Van Essen, “Communication quantization for data-parallel training of deep neural networks,” in 2016 2nd Workshop on Machine Learning in HPC Environments (MLHPC), 2016, pp. 1–8.

[3] A. F. Aji and K. Heafield, “Sparse communication for distributed gradient descent,” CoRR, vol. abs/1704.05021, 2017. [Online]. Available: http://arxiv.org/abs/1704.05021

[4] S. U. Stich, J. Cordonnier, and M. Jaggi, “Sparsified SGD with memory,” in NeurIPS - Advances in Neural Information Processing Systems, 2018, pp. 4448–4459.

[5] 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, H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alch´e-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019.

[6] V. Gandikota, D. Kane, R. Kumar Maity, and A. Mazumdar, “VQSGD: Vector quantized stochastic gradient descent,” in Proceedings of The 24th International Conference on Artificial Intelligence and Statistics, A. Banerjee and K. Fukumizu, Eds., vol. 130. PMLR, 09–15 Apr 2021. pp. 2197–2205. [Online]. Available: https://proceedings.mlr.press/v130/gandikota21a.html

[7] D. Alistarh, T. Hoefler, M. Johansson, N. Konstantinov, S. Khrirat, and C. Renggli, “The convergence of sparsified gradient methods,” in Advances in Neural Information Processing Systems, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018.

[8] A. Koloskova, S. Stich, and M. Jaggi, “Decentralized stochastic optimization and gossip algorithms with compressed communication,” in Proceedings of the 36th International Conference on Machine Learning, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019. pp. 3478–3487. [Online]. Available: https://proceedings.mlr.press/v97/koloskova19a.html

[9] B. Barazandeh, D. A. Tarzanagh, and G. Michailidis, “Solving a class of non-convex min-max games using adaptive momentum methods,” in ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021. pp. 3625–3629.

[10] A. R. Mahmood, R. S. Sutton, T. Degris, and P. M. Pilarski, “Tuning-free step-size adaptation,” in 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2012, pp. 2121–2124.

[11] A. Aghazadeh, A. Ayremlou, D. D. Calderon, T. Goldstein, R. Patel, D. Vats, and R. G. Baraniuk, “Adaptive step size selection for optimization via the ski rental problem,” in 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, 2013, pp. 5383–5387.

[12] S. Vaswani, A. Mishkin, I. H. Laradji, M. Schmidt, G. Gidel, and S. Lacoste-Julien, “Painless stochastic gradient: Interpolation, line-search, and convergence rates,” CoRR, vol. abs/1905.09997, 2019. [Online]. Available: http://arxiv.org/abs/1905.09997

[13] B. Dubois-Taine, S. Vaswani, R. Babanazhad, M. Schmidt, and S. Lacoste-Julien, “SVRG meets adagrad: Painless variance reduction,” CoRR, vol. abs/2102.09645, 2021. [Online]. Available: https://arxiv.org/abs/2102.09645

[14] L. Armijo, “Minimization of functions having Lipschitz continuous first partial derivatives.” Pacific Journal of Mathematics, vol. 16, no. 1, pp. 1 – 3, 1966. [Online]. Available: https://doi.org/

[15] R. Bassily, M. Belkin, and S. Ma, “On exponential convergence of sgd in non-convex over-parametrized learning,” arXiv preprint arXiv:1811.02564, 2018.

[16] C. Liu and M. Belkin, “Mass: an accelerated stochastic method for over-parametrized learning,” CoRR, vol. abs/1810.13395, 2018. [Online]. Available: http://arxiv.org/abs/1810.13395

[17] A. Montanari and Y. Zhong, “The interpolation phase transition in neural networks: Memorization and generalization under lazy training,” arXiv preprint arXiv:2007.12826, 2020.

[18] A. M. Subramaniam, A. Magesh, and V. V. Veeravalli, “Adaptive step-size methods for compressed sgd,” arXiv preprint arXiv:2207.10046, 2022.

[19] H. Mania, X. Pan, D. Papailiopoulos, B. Recht, K. Ramchandran, and M. I. Jordan, “Perturbed iterate analysis for asynchronous stochastic optimization,” SIAM J. Optim., vol. 27, pp. 2202–2229, 2017.