ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-7605-5/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICASSP39728.2021.9414419

IMPROVED STEP-SIZE SCHEDULES FOR NOISY GRADIENT METHODS

Sarit Khirirat[1]*, Xiaoyu Wang*[1]*, Sindri Magnússon*[2]*, Mikael Johansson*[1]

[1]KTH Royal Institute of Technology, [2]Stockholm University

ABSTRACT

Noise is inherited in many optimization methods such as stochastic gradient methods, zeroth-order methods and compressed gradient methods. For such methods to converge toward a global optimum, it is intuitive to use large step-sizes in the initial iterations when the noise is typically small compared to the algorithm-steps, and reduce the step-sizes as the algorithm progresses. This intuition has been confirmed in theory and practice for stochastic gradient methods, but similar results are lacking for other methods using approximate gradients. This paper shows that the diminishing step-size strategies can indeed be applied for a broad class of noisy gradient methods. Unlike previous works, our analysis framework shows that such step-size schedules enable these methods to enjoy an optimal $\mathcal{O}(1/k)$ rate. We exemplify our results on zeroth-order methods and stochastic compression methods. Our experiments validate fast convergence of these methods with the step decay schedules.

***Index Terms*—** Optimization, machine learning, distributed algorithms, zeroth-order algorithms, quantization.

1. INTRODUCTION

Many problems in signal processing and machine learning can be cast as convex optimization problems. These problems are often solved by first-order methods such as (stochastic) gradient descent. It is well-known that gradient descent with a constant step-size enjoys a linear rate toward the optimal solution for strongly convex problems. However, we can only access noisy gradients for several applications. For example, in machine learning we often approximate the gradient from a few samples. In other scenarios, we approximate the gradient using zeroth-order information [1, 2]. In distributed optimization, we compress gradients so that they can be communicated efficiently over a digital channel [3, 4, 5, 6, 7, 8]. If we can only access noisy gradients then gradient methods with a constant step-size will converge to sub-optimal solution. To improve solution accuracy, we need to decrease the constant step-size, which leads to slower convergence.

To improve the convergence, we might use large step-sizes initially and then decrease the step-sizes as the algorithm progresses/converges. Such step-size schedules have been shown to improve the convergence of stochastic gradient methods in both theory and practice. These ideas were elegantly analyzed by Polyak (see [9, chapter 4]), and have been polished by many recent works, e.g., [10, 9, 11, 12, 13, 14, 15, 16]. However, most existing analysis usually considers the diminishing step-sizes only for stochastic gradient methods. Intuitively, these decaying schedules benefit noisy gradient methods in general. In this paper, we provide a unified theoretical justification that shows the benefit of using decreasing step-sizes for popular noisy gradient methods.

Contributions: We provide general step-size schedules for optimization methods with noise. In particular, we consider general Lyapunov functions that can be used to analyze many noisy optimization algorithms. For these algorithms, we show how to tune the step-sizes to ensure $\mathcal{O}(1/k)$ convergence rate. We illustrate how these results can be used to improve the convergence for a) zeroth-order methods and b) compressed gradient methods in both theory and practice.

Notation: We let $\mathbb{N}, \mathbb{N}_0, \mathbb{Z}$ be the set of natural numbers, the set of natural numbers including zero, and the set of integers, respectively. For $x \in \mathbb{R}^d$, $\|x\|$ is its ℓ_2 norm, while x^i is its i^{th} element. Finally, $g(k) = \mathcal{O}(h(k))$ implies that $g(k) \leq Mh(k)$ for all $k \in \mathbb{N}$ and some $M \in \mathbb{R}$, while $g(k) = o(h(k))$ means that

$$\lim_{k\to\infty} \frac{g(k)}{h(k)} = 0,$$

or, in other words, that $g(k)$ decreases much faster than $h(k)$.

2. CENTRAL LEMMAS: STEP-SIZE SCHEDULES FOR A NOISY LYAPUNOV FUNCTION

Many optimization algorithms can be analyzed by considering Lyapunov functions that satisfy a recursion on the following form (see Section 3):

$$V_{k+1} \leq (1 - A\gamma_k)V_k + \gamma_k^2 B. \tag{1}$$

This work was partially supported by the Wallenberg Artificial Intelligence, Autonomous Systems and Software Program (WASP) funded by Knut and Alice Wallenberg Foundation. This work is also supported in parts by the Swedish Research Council (Vetenskapsrådet) under grant 2020-03607. S. Magnússon is with Department of Computer and System Science, Stockholm University, while S. Khirirat, X. Wang, and M. Johansson are with the Division of Decision and Control Systems, Royal Institute of Technology (KTH), Stockholm, Sweden. Emails: sindri.magnusson@dsv.su.se, sarit@kth.se, wang10@kth.se, mikaelj@kth.se

Here, V_k is the Lyapunov function, and γ_k is the step-size/learning rate of the optimization algorithms. The main goal is to provide a general analysis for algorithms with given Lyapunov functions and show how to tune step-sizes to get improved convergence rate.

In an idealized setting, where A and B are known, and V_k can be monitored at each step, it would be natural to select γ_k to minimize the right-hand side of (1). This corresponds to letting $\gamma_k = AV_k/(2B)$. A short calculation reveals that V_k would then satisfy

$$V_k \leq \frac{4B}{A^2 k} \tag{2}$$

indicating that

$$\gamma_k = \frac{2A^{-1}}{k}$$

would be a good step-size choice. The next result validates that this is indeed the case.

Lemma 1. *Choose the step-size*

$$\gamma_k = \min\left\{\gamma, \frac{\alpha}{k+1}\right\},$$

where $\alpha > 0$, $\gamma \in (0, 1/A)$, and $A\alpha > 1$. If we set set $k^* = \max\{0, \alpha/\gamma - 1\}$ and $k > k^*$ then we have the following convergence rate:

$$V_k \leq \frac{B\alpha^2\nu}{(A\alpha-1)}\frac{1}{(k+1)} + \frac{V_0^*}{(k+1)^{A\alpha}} + \frac{B\alpha^2\nu}{(k+1)^2},$$

where $V_0^* = (k^*+2)^{A\alpha}((1-A\gamma)^{(k^*+1)}V_0 + B\gamma/A)$ and $\nu = (1+1/(k^*+2))^2$. Moreover, if we set $\alpha = 2/A$ we get

$$V_k \leq \frac{4B}{A^2(k+1)} + \frac{V_0^* + 2\ln(k+1) + 2}{(k+1)^2}. \tag{3}$$

Proof. Let $k^* = \max\{0, \alpha/\gamma - 1\}$. For $0 \leq k \leq k^*$, the step-size $\gamma_k = \gamma$. For $k > k^*$, we have $\gamma_k = \alpha/(k+1)$. Recursively applying the step size $\gamma_k = \gamma$ for $0 \leq k \leq k^*$ in Eq. (1) gives

$$V_{k^*+1} \leq (1-A\gamma)^{(k^*+1)}V_0 + \frac{B\gamma}{A}. \tag{4}$$

For $k > k^*$, plugging the step-size $\gamma_k = \alpha/(k+1)$ into Eq. (1) yields

$$V_{k+1} \leq \left(1 - \frac{A\alpha}{k+1}\right)V_k + \frac{B\alpha^2}{(k+1)^2}.$$

By recursively applying the inequality and utilizing $1 + x \leq \exp(x)$ for $x \in \mathbb{R}$, we get

$$V_{k+1} \leq \exp\left(-A\alpha\sum_{l=k^*+1}^{k}\frac{1}{l+1}\right)V_{k^*+1} + B\alpha^2\sum_{l=k^*+1}^{k}\frac{1}{(l+1)^2}\exp\left(-A\alpha\sum_{i=l+1}^{k}\frac{1}{i+1}\right). \tag{5}$$

Since $1/(s+1)$ is decreasing in s, we get

$$\sum_{i=l+1}^{k}\frac{1}{i+1} \geq \int_{i=l+1}^{k+1}\frac{di}{i+1} = \ln(k+2) - \ln(l+2).$$

By applying these inequalities in Eq. (5), we get

$$V_{k+1} \leq \frac{(k^*+2)^{A\alpha}V_{k^*+1}}{(k+2)^{A\alpha}} + \frac{B\alpha^2}{(k+2)^{A\alpha}}\sum_{l=k^*+1}^{k}\frac{(l+2)^{A\alpha}}{(l+1)^2}.$$

By Eq. (4) and the above inequality, for $k > k^*$

$$V_{k+1} \leq \frac{V_0^*}{(k+2)^{A\alpha}} + \frac{B\alpha^2}{(k+2)^{A\alpha}}\sum_{l=k^*+1}^{k}\frac{(l+2)^{A\alpha}}{(l+1)^2}, \tag{6}$$

where $V_0^* = (k^*+2)^{A\alpha}((1-A\gamma)^{(k^*+1)}V_0 + B\gamma/A)$. Hence, for $A\alpha > 1$

$$\begin{aligned}\sum_{l=k^*+1}^{k}\frac{(l+2)^{A\alpha}}{(l+1)^2} &\leq \nu\sum_{l=k^*+1}^{k}(l+2)^{A\alpha-2} \\ &\leq \nu\int_{l=k^*+1}^{k}(l+2)^{A\alpha-2}dl + \nu(k+2)^{A\alpha-2} \\ &\leq \frac{\nu(k+2)^{A\alpha-1}}{A\alpha-1} + \nu(k+2)^{A\alpha-2},\end{aligned}$$

where $\nu = (1+1/(k^*+2))^2$. Incorporating the above inequality into Eq. (6), we have

$$V_{k+1} \leq \frac{V_0^*}{(k+2)^{A\alpha}} + \frac{B\alpha^2\nu}{(A\alpha-1)}\frac{1}{(k+2)} + \frac{B\alpha^2\nu}{(k+2)^2}.$$

Finally, if we choose $\alpha = 2/A$, then Eq. (6) can be improved by the follow procedure:

$$\begin{aligned}V_{k+1} &\leq \frac{V_0^*}{(k+2)^2} + \frac{B\alpha^2}{(k+2)^2}\sum_{l=k^*+1}^{k}\left(1 + \frac{2}{l+1} + \frac{1}{(l+1)^2}\right) \\ &\leq \frac{V_0^*}{(k+2)^2} \\ &\quad + \frac{B\alpha^2}{(k+2)^2}\left(k - k^* + 3 + 2\int_{l=k^*+1}^{k+1}\frac{dl}{l} + \int_{l=k^*+1}^{k+1}\frac{dl}{l^2}\right) \\ &\leq \frac{V_0^*}{(k+2)^2} + \frac{B\alpha^2(k+3+2\ln(k+1)+\frac{1}{k^*+1}-k^*)}{(k+2)^2} \\ &\leq \frac{V_0^* + 2\ln(k+1) + 2}{(k+2)^2} + \frac{B\alpha^2}{k+2}.\end{aligned}$$

We complete the proof. □

Lemma 1 provides a step-size schedule ensuring an $O(1/k)$ convergence rate for algorithms with a Lyapunov function on the form of (1). To apply this step-size schedule we do not need to know B or to monitor V_k from (1).

Nevertheless, as shown in Eq. (3), for $\alpha = 2/A$ we get the convergence rate

$$V_{k+1} \leq \frac{4B}{A^2(k+1)} + o\left(\frac{1}{k}\right),$$

which is comparable to what we would get if we knew B and could monitor V_k, see discussion around Eq. (2) above.

We can apply Lemma 1 directly to stochastic gradient iterations on strongly convex problems. This will give us a simple convergence proof with guarantees on par with the best known results, e.g., section 3 in [12], [16] or chapter 4 in [9]. However, Lemma 1 also allows us to apply similar step-size schedules to more general noisy gradient methods.

3. APPLICATIONS: STRONGLY CONVEX OPTIMIZATION WITH NOISE

We now illustrate how our step-size in the previous section can be used to improve complexity bounds of two popular noisy gradient methods. We consider problems on the form:

$$\underset{x\in\mathbb{R}^d}{\text{minimize}} \quad F(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x), \tag{7}$$

where each function $f_i(\cdot)$ satisfies the following assumptions.

Assumption 1. *$f_i(\cdot)$ is L-smooth, i.e. for all $x, y \in \mathbb{R}^d$*

$$\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x - y\|.$$

Assumption 2. *$F(\cdot)$ is strongly convex with a positive parameter μ, i.e. for all $x, y \in \mathbb{R}^d$*

$$F(y) \geq F(x) + \langle \nabla F(x), y - x\rangle + \frac{\mu}{2}\|y - x\|^2.$$

3.1. Zeroth-order Methods

In real-life applications, it is common that we want to perform optimization but it is expensive or even impossible to access gradients. This is for example the case when the objective function is computed from simulations or in control systems where we only observe the objective function value for the actions that we take. In these applications it is common to use zeroth-order optimization, where a directional derivative (in a random direction) is estimated based on finite difference. We illustrate here the zeroth-order method in [1], which progresses as follows:

$$x_{k+1} = x_k - \gamma_k g_k, \quad \text{for } k \in \mathbb{N} \tag{8}$$

for a given initialization x_0 where

$$g_k = \frac{F(x_k + \tau_k u_k) - F(x_k)}{\tau_k} u_k, \tag{9}$$

is a finite difference estimation of the directional derivative in the direction $u_k \in \mathbb{R}^d$. Here $\tau_k > 0$ is a smoothness parameter, as τ_k converges to 0, g_k converges to the directional derivative of $F(\cdot)$ in the direction u_k. Moreover, we assume here that u_k is a random direction, in particular, a zero-mean and unit-variance Gaussian noise. We have the following result from [1, Lemma 3]

$$\|g_k - \nabla F(x_k)\|^2 \leq \tau_k^2 L^2 (d+3)^3/4. \tag{10}$$

We can show that if $\tau_k = \gamma_k$, then Algorithm (8) is shown to satisfy the central lemmas with the associated parameters shown next:

Theorem 1. *Consider the optimization problem* (7)*. Suppose that $\{x_k\}_{k\in\mathbb{N}}$ is generated by the zeroth-order method* (8) *with $\gamma_k \in (0, 1/L]$ and set*

$$V_k = F(x_k) - F(x^\star), \quad A = \mu, \quad B = L(d+3)^3/8.$$

Then V_k progresses according to Eq. (1)*.*

3.2. Compressed Gradient Methods

In distributed optimization, gradients are often communicated to enable a descent update. It is often desirable to compress these gradients, since communicating full-precision gradient is expensive for large dimensional problems. For example, we need 80 MB to communicate a $d = 10^7$ dimensional gradient, which is not an uncommon problem size for machine learning problems. We can formulate such compressed gradient algorithms with the following iterations

$$x_{k+1} = x_k - \gamma_k \frac{1}{n}\sum_{i=1}^{n} Q(\nabla f_i(x_k)), \quad \text{for } k \in \mathbb{N}, \tag{11}$$

where $Q(\cdot)$ is a stochastic compression satisfying the unbiased and variance-bounded properties, i.e. for $x \in \mathbb{R}^d$ and $q \in \mathbb{R}$

$$\mathbf{E}[Q(x)] = x, \quad \text{and} \quad \mathbf{E}\|Q(x)\|^2 \leq q\|x\|^2. \tag{12}$$

This means that stochastic compression has high accuracy when q is close to 1. Examples of compressions that satisfy these properties are stochastic sparsification:

$$[Q_p(v)]^i = (v^i/p^i)\xi^i, \quad \forall i \in \{1, 2, \ldots, d\} \tag{13}$$

where $\xi^i \sim$ Bernouli(p_i). There are many heuristics to choose p^i. For instance, if we set $p^i = |v^i|/\|v\|_q$ with $q = 2$, $q = \infty$ and $q \in (0, \infty]$, then we get, respectively, `QSGD` in [3] with $s = 1$, the `TernGrad` in [6], and the ℓ_q-quantizer in [17]. The probabilities p^i can also be fine-tuned adaptively to minimize the variance [17], or to maximize the communication efficiency [18].

We now show that the compression methods (11) can be analysed by Lyapunov functions on the form of Eq. (1).

Theorem 2. *Consider the optimization problem* (7)*. Suppose that* $\{x_k\}_{k\in\mathbb{N}}$ *are generated by the compression method* (11) *with* $\gamma_k \in (0, 1/(2qL)]$ *and set*

$$V_k = \mathbf{E}\|x_k - x^\star\|^2, \quad A = \mu, \quad B = \frac{2q}{n}\sum_{i=1}^{n}\|\nabla f_i(x^\star)\|^2.$$

Then V_k *progresses accroding to Eq.* (1)*.*

By Theorem 2 and Lemma 1, we can establish the iteration complexity of the stochastic compression methods using the diminishing step-size with $\alpha = 2/A$ and $\gamma = 1/(2qL)$. To reach the ϵ-accuracy, the methods then need to run at least

$$\max\left(\frac{16q\sigma^2}{\mu^2}\cdot\frac{c_1}{\epsilon}, (\eta+1)\sqrt{\frac{2c_2\epsilon_0 + \sigma^2/(\mu L)}{\epsilon}}\right) \quad \text{iterations},$$

where $c_1 = (\eta+2)^2/(\eta+1)^2$, $c_2 = (1-\mu/(2qL))^\eta$ and $\eta = 4qL/\mu$. On the other hand, the stochastic compression methods with the fixed step-size $\gamma_k = 0.5/(q(\beta+L))$ where $\beta = 2\sigma^2/(\mu\epsilon)$ in [18, Theorem 3] require at least

$$\frac{2qL}{\mu}\log\left(\frac{2\epsilon_0}{\epsilon}\right) + \frac{4q\sigma^2}{\mu^2}\cdot\frac{1}{\epsilon}\log\left(\frac{2\epsilon_0}{\epsilon}\right) \quad \text{iterations}.$$

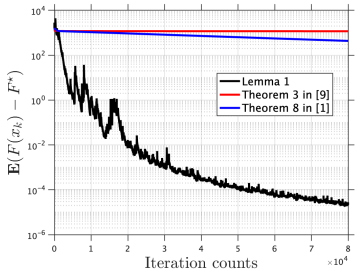
Clearly, the diminishing step-size achieves $\mathcal{O}(1/\epsilon)$ iteration complexity, while the fixed step-size in [18, Theorem 3] attains $\mathcal{O}((1/\epsilon)\log(1/\epsilon))$ complexity. The stochastic compression methods with the decaying step-size policy hence converge toward the optimum faster than those with the fixed step-size. The benefit of using the step decay schedule is also confirmed empirically in Section 4.

4. EXPERIMENTS

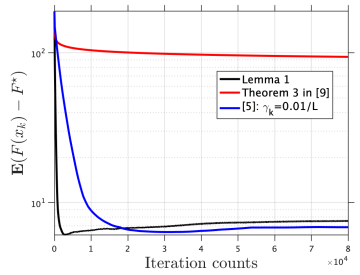
In this section we present numerical experiments for solving ℓ_2-regularized least-squares problems, *i.e.* minimization problems on the form (7) with

$$F(x) = \sum_{i=1}^{m}(a_i^T x - b_i)^2 + \frac{\mu}{2}\|x\|^2,$$

where μ is a positive regularization parameter and (a_1, b_1), $\ldots$, (a_m, b_m) represent the n training samples. Here, $a_i \in \mathbb{R}^d$ is the i^{th} training input and $b_i \in \mathbb{R}$ is the associated output. We implemented zeroth-order methods and compressed gradient methods using `QSGD` with $s = 1$ and 10 workers in Julia, and generated each training example a_i and b_i, respectively, according to the uniform distribution in the range $[0, 1)$ and to the normal distribution with zero mean and unit variance. Throughout the experiments, we normalized each a_i by its Euclidean norm, and set $m = 8,000$, $d = 100$, $\mu = 1$, and also $x_0 \sim \mathcal{N}(0, 1)$. The results are averaged over three Monte Carlo runs.



(a) Zeroth-order Methods



(b) Compressed Gradient Methods

Fig. 1: Performance of noisy gradient methods.

From Figure 1(a), zeroth-order methods with the step-size in Lemma 1 outperform those with existing step-size strategies, in terms of both convergence rate and accuracy. At $k = 40,000$, the step-size in Lemma 1 has attained an accuracy improvement of more than six orders of magnitude compared to $\gamma_k = 1/(L\cdot k)$ from [9, Theorem 3] and $\gamma_k = 0.25/((d+4)L)$ from [1, Theorem 8]. Figure 1(b) similarly shows fast convergence of stochastic compression methods with the step-size in Lemma 1, compared to existing policies. To reach $\mathbf{E}(F(x_k) - F^\star) \leq 10$, the step-size in Lemma 1 leads to a roughly tenfold decrease in required iteration counts, compared to $\gamma_k = 0.01/L$ in [19].

5. CONCLUSIONS

Decaying step-size schedules were shown extensively on stochastic gradient methods to enjoy convergence toward the global optimum. However, these results are lacking for other methods operating on noisy gradients. This paper provides a unified theoretical analysis which shows the benefit of diminishing step-sizes on general noisy gradient methods. In essence, we prove the $\mathcal{O}(1/k)$ rate of many popular methods using these step-sizes such as zeroth-order methods and compressed gradient methods. We exemplify these methods in numerical experiments, highlighting that our decreasing step-size choices have superior practical performance over existing strategies in both convergence rate and solution accuracy.

6. REFERENCES

[1] Yurii Nesterov and Vladimir Spokoiny, "Random gradient-free minimization of convex functions," *Foundations of Computational Mathematics*, vol. 17, no. 2, pp. 527–566, 2017.

[2] Ahmad Ajalloeian and Sebastian U Stich, "Analysis of SGD with biased gradient estimators," *arXiv preprint arXiv:2008.00051*, 2020.

[3] Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic, "QSGD: communication-efficient sgd via gradient quantization and encoding," in *Advances in Neural Information Processing Systems*, 2017, pp. 1709–1720.

[4] Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar, "signSGD: Compressed optimisation for non-convex problems," in *International Conference on Machine Learning*, 2018, pp. 560–569.

[5] Alyazeed Albasyoni, Mher Safaryan, Laurent Condat, and Peter Richtárik, "Optimal gradient compression for distributed and federated learning," *arXiv preprint arXiv:2010.03246*, 2020.

[6] Wei Wen, Cong Xu, Feng Yan, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li, "Terngrad: Ternary gradients to reduce communication in distributed deep learning," in *Advances in Neural Information Processing Systems*, 2017, pp. 1509–1519.

[7] Sindri Magnússon, Chinwendu Enyioha, Na Li, Carlo Fischione, and Vahid Tarokh, "Convergence of limited communication gradient methods," *IEEE Transactions on Automatic Control*, vol. 63, no. 5, pp. 1356–1371, 2017.

[8] Sindri Magnússon, Hossein Shokri-Ghadikolaei, and Na Li, "On maintaining linear convergence of distributed learning and optimization under limited communication," in *2019 53rd Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2019, pp. 432–436.

[9] Boris T Polyak, "Introduction to optimization," *Inc., Publications Division, New York*, vol. 1, 1987.

[10] Eric Moulines and Francis R Bach, "Non-asymptotic analysis of stochastic approximation algorithms for machine learning," in *Advances in Neural Information Processing Systems*, 2011, pp. 451–459.

[11] Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro, "Robust stochastic approximation approach to stochastic programming," *SIAM Journal on optimization*, vol. 19, no. 4, pp. 1574–1609, 2009.

[12] Alexander Rakhlin, Ohad Shamir, and Karthik Sridharan, "Making gradient descent optimal for strongly convex stochastic optimization," in *Proceedings of the 29th International Coference on International Conference on Machine Learning*, 2012, pp. 1571–1578.

[13] Simon Lacoste-Julien, Mark Schmidt, and Francis Bach, "A simpler approach to obtaining an $\mathcal{O}(1/t)$ convergence rate for the projected stochastic subgradient method," *arXiv preprint arXiv:1212.2002*, 2012.

[14] Ohad Shamir and Tong Zhang, "Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes," in *International Conference on Machine Learning*, 2013, pp. 71–79.

[15] Robert Mansel Gower, Nicolas Loizou, Xun Qian, Alibek Sailanbayev, Egor Shulgin, and Peter Richtárik, "SGD: General analysis and improved rates," in *International Conference on Machine Learning*, 2019, pp. 5200–5209.

[16] Lam Nguyen, Phuong Ha Nguyen, Marten Dijk, Peter Richtárik, Katya Scheinberg, and Martin Takác, "SGD and hogwild! convergence without the bounded gradients assumption," in *International Conference on Machine Learning*. PMLR, 2018, pp. 3750–3758.

[17] Hongyi Wang, Scott Sievert, Shengchao Liu, Zachary Charles, Dimitris Papailiopoulos, and Stephen Wright, "ATOMO: Communication-efficient learning via atomic sparsification," in *Advances in Neural Information Processing Systems*, 2018, pp. 9850–9861.

[18] Sarit Khirirat, Sindri Magnússon, Arda Aytekin, and Mikael Johansson, "Communication efficient sparsification for large scale machine learning," *arXiv preprint arXiv:2003.06377*, 2020.

[19] Sarit Khirirat, Hamid Reza Feyzmahdavian, and Mikael Johansson, "Distributed learning with compressed gradients," *arXiv preprint arXiv:1806.06573*, 2018.

[20] Rie Johnson and Tong Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," in *Advances in Neural Information Processing Systems*, 2013, pp. 315–323.