mathematics

Article

Stagewise Accelerated Stochastic Gradient Methods for Nonconvex Optimization

Cui Jia [1] and Zhuoxu Cui [2,3,*]

[1] School of Statistics and Data Science, Ningbo University of Technology, Ningbo 315211, China; cjia80@whu.edu.cn
[2] School of Mathematics and Statistics, Wuhan University, Wuhan 430072, China
[3] Research Center for Medical AI, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 51800, China
[*] Correspondence: zhuoxucui@whu.edu.cn; Tel.: +86-183429323267

Abstract: For large-scale optimization that covers a wide range of optimization problems encountered frequently in machine learning and deep neural networks, stochastic optimization has become one of the most used methods thanks to its low computational complexity. In machine learning and deep learning problems, nonconvex problems are common, while convex problems are rare. How to find the global minimum for nonconvex optimization and reduce the computational complexity are challenges. Inspired by the phenomenon that the stagewise stepsize tuning strategy can empirically improve the convergence speed in deep neural networks, we incorporate the stagewise stepsize tuning strategy into the iterative framework of Nesterov's acceleration- and variance reduction-based methods to reduce the computational complexity, i.e., the stagewise stepsize tuning strategy is incorporated into randomized stochastic accelerated gradient and stochastic variance-reduced gradient. The proposed methods are theoretically derived to reduce the complexity of the nonconvex and convex problems and improve the convergence rate of the frameworks, which have the complexity $O(1/\mu\epsilon)$ and $O(1/\mu\sqrt{\epsilon})$, respectively, where μ is the PL modulus and L is the Lipschitz constant. In the end, numerical experiments on large benchmark datasets validate well the competitiveness of the proposed methods.

Keywords: stagewise stepsize tuning strategy; variance reduction; Nesterov's acceleration; nonconvex

MSC: 90C26; 65K10

check for updates

Citation: Jia, C.; Cui, Z. Stagewise Accelerated Stochastic Gradient Methods for Nonconvex Optimization. *Mathematics* **2024**, *12*, 1664. https://doi.org/10.3390/math12111664

Academic Editors: Moudafi Abdellatif and Jüri Majak

Received: 2 April 2024
Revised: 1 May 2024
Accepted: 23 May 2024
Published: 27 May 2024

1. Introduction

In this thesis, we consider the following empirical risk minimization problem:

$$\min_{x\in\mathbb{R}^d} F(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x) \tag{1}$$

where x represents the model parameters and $f_i : \mathbb{R}^d \to \mathbb{R}$ denotes a smooth but possibly nonconvex function. In particular, $f_i(x) := \ell(x, a_i, b_i)$ often denotes the loss function on given training sample $\{(a_i, b_i) \in \mathbb{R}^{d\times 1}\}$. Therefore, in problem (1), for example, when $\ell(x, a_i, b_i) = \log(1 + \exp(-b_i x^T a_i))$, (1) reduces to the logistic regression [1]; or if we let $\ell(x, a_i, b_i) = (\sigma_l(w_l^T \ldots \sigma_1(w_1^T a_i)) - b_i)^2$ where $x := [w_1, \ldots, w_l]$ and σ_s, $s = 1, \ldots, l$ denote activation functions, we obtain the training model of DNNs [2].

To solve problem (1), one of the standard methods is the gradient descent (GD) that carries out the following updates:

$$x_{k+1} = x_k - \frac{\eta_k}{n}\sum_{i=1}^{n} \nabla f_i(x_k)$$

where η_k is the stepsize. Since the above recursion needs to evaluate derivatives n times at each iteration, it is impracticable for problems in larger scales. To break this bottleneck, there has been a growing interest on stochastic methods to reduce the computational cost, among which the method of stochastic gradient descent (SGD) [3–5] is a typical one. In particular, the SGD reads as

$$x_{k+1} = x_k - \eta_k \nabla f_{i_k}(x_k),$$

where i_k is an i.i.d. random variable taking value in $\{1, 2, \ldots, n\}$ uniformly. Obviously, the computational complexity of SGD is independent with the size n. Thereupon, SGD becomes one of the most popular first-order methods to solve large-scale optimization problems [6–8]. However, because of the random interference caused by the stochastic gradient, SGD can only tolerate a relatively small stepsize that makes the convergence rate of SGD slower than its non-stochastic counterpart (i.e., GD). Consequently, it has become a hot topic to develop accelerated algorithms.

1.1. Related Works

The accelerated methods for optimization can be traced back to the 1960s. Inspired by the motion of a heavy ball in a potential field, Polyak [9] sped up the convergence of GD numerically, but without a rigorous analysis. In [10], Nesterov devised a different iterative framework and obtained a rigorous result in that, by running the Nesterov algorithm with the complexity at most $O(1/\sqrt{\epsilon})$ iterations, one can find an ϵ-minimum, i.e., a point $\mathring{x}$ such that $F(\mathring{x}) - F(x^*) \le \epsilon$ where x^* denotes the minimizer of objective F over $\mathbb{R}^d$. Henceforth, various variations of Nesterov's acceleration-based methods emerged, see [11–15] and references therein. However, please note that the convexity has played a pivotal role in establishing the convergence of the above methods.

In this paper, we mainly focus on acceleration playing a role in stochastic nonconvex optimization. Different from convex optimization, it is impractical to find a global minimum for nonconvex optimization in general. Consequently, one aims to seek a weaker guarantee, i.e., an ϵ-stationary point, that is, a point $\mathring{x}$ with a sufficiently small gradient $\|\nabla F(\mathring{x})\| \le \epsilon$ or $\mathbb{E}[\|\nabla F(\mathring{x})\|] \le \epsilon$, to surrogate the local optimal. When the objective is assumed to meet Lipschitz continuously differentiable (l.c.d.) with constant L, Ghadimi and Lan [16] showed that SGD needs to be run $O(L^2/\epsilon + L\sigma/\epsilon^2)$ times to find an ϵ-stationary point. To improve convergence rate, naturally, a number of researchers [17,18] applied Nesterov's acceleration to the nonconvex case, but no theoretical guarantees for faster rate were given. Recently, Ghadimi and Lan [19] devised a new elaborate iterative framework, termed the randomized stochastic accelerated gradient (RSAG) method, and showed that the iterative complexity can be reduced to $O(L/\epsilon + L\sigma/\epsilon^2)$ to find an ϵ-stationary point, where σ denotes the upper bound of the standard deviation of the stochastic gradient.

The reason why SGD converges slowly is that, to avoid divergence caused the random interference, only a relatively small stepsize can be tolerated. Then, reducing the random interference in stochastic gradient becomes another method for acceleration. For this purpose, Johnson and Zhang [20] proposed an easy and feasible approach named VR; that is, after dividing the total iterations into a number of epochs firstly, once the iteration is performed through an epoch, the full gradient is calculated once to avoid the stochastic gradient deviating too far. Considering strongly convex objectives, Johnson and Zhang followed the ideal of VR to propose the stochastic variance reduced gradient (SVRG), for which a relatively large stepsize becomes acceptable. In particular, Johnson and Zhang showed that SVRG can achieve a linear convergence rate when the stepsize is chosen appropriately. However, in modern learning problems, the strong convexity is often unsatisfied. For this, Reddi et al. [21] extended the SVRG to handle nonconvex objectives and showed that SVRG can achieve iterative complexity $O(n^{2/3}L/\epsilon)$ to find an ϵ-stationary point. Shang et al. [22] proposed a simple stochastic variance reduction method for machine learning, termed as VR-SGD.

As discussed earlier, in SGD, a large stepsize may amplify the random interference to cause the iterates to diverge. On the other hand, a too-small stepsize may make it difficult for the iterates to escape saddle points. In the process of solving DNN and other nonconvex models [23–26], one empirically finds a phenomenon that the optimization algorithms equipped with SSTS, which start from a relatively large stepsize and decrease it geometrically after a number of iterations, can improve the convergence speed effectively. In terms of theoretical analysis, Xu et al. [27] showed that the convergence rate of stagewise SGD (i.e., SGD equipped with SSTS), abbreviated as S-SGD, for convex objectives can be significantly improved with different degrees under different local growth conditions. However, the convexity plays an important role in deriving the above result, which is not met for modern learning problems in general, such as training DNNs. Recently, the Polyak–Łojasiewicz (PL) condition has been observed and proved for training DNNs [28–32]. For the vanilla SGD under PL condition, Arjevani et al. [33] studied the lower bounds of ϵ^{-3} to find an ϵ -stationary point by using stochastic first-order methods in the certain condition. Horváath et al. [34] showed the iterative complexity to find an ϵ-approximate solution that can fall in between $O(1/(\mu^2\epsilon^2))$ and $O(1/(\mu\epsilon))$ where μ denotes the PL modulus. In learning DNNs, $\mu \ll 1$ generally, which dampens its performance severely. Wang et al. [35] proposed the momentum stochastic method to achieve an ϵ-stationary solution under a constant step size with $O(1/(\epsilon^2))$ computation complexity. Yuan et al. [36] considered SSTS work for the nonconvex objective meeting PL condition, and then answered the question in the affirmative in [37]. Particularly, they showed that the iterative complexity of SGD for nonconvex objective meeting PL condition can be reduced to $O(L/\mu\epsilon)$ (which is smaller than $O(1/(\mu^2\epsilon))$ significantly due to $\mu \ll 1$) when SSTS is adopted.

In this paper, we mainly consider whether we can further improve the convergence rate or reduce iterative complexity by incorporating the SSTS into the iterative framework of Nesterov's acceleration based methods and VR based methods. We will answer this question in the affirmative. Specifically, we will develop SSTS-equipped RSAG and SVRG algorithms respectively and give their corresponding their theoretical analysis.

1.2. Contributions

In this paper, we mainly develop and analyze two accelerated algorithms, namely, SSTS equipped RSAG and SVRG. Specifically, the main contributions of the paper are summarized as follows:

- Incorporating the SSTS into the iterative framework of RSAG, we propose the stagewise RSAG, abbreviated as S-RSAG, and show that the iterative complexities of it are at most $O(L/\mu\epsilon)$ to find an ϵ-stationary point for nonconvex objective and $O(1/\mu\epsilon)$ to find an ϵ-minimum for convex objective, which are significantly reduced with respect to its non-stagewise counterpart RSAG, the complexities of which are $O(L^2/\epsilon + L/\epsilon^2)$ and $O(L/\sqrt{\epsilon} + 1/\epsilon^2)$ respectively, where μ is the PL modulus and L is the Lipschitz constant. Compared to existing stagewise algorithm S-SGD, the complexities of our S-RSAG are more superior under the convex condition (i.e., $O(1/\mu\epsilon)$ with respect to $O(L/\mu\epsilon)$, where $L \gg 1$ in general) and at the same level under the nonconvex condition.
- With the same methodology, we propose the stagewise SVRG, abbreviated as S-SVRG and show that the iterative complexities of it are at most $O(Lm/(\mu\sqrt{\epsilon}))$ to find an ϵ-stationary point for nonconvex objective and $O(Lm/(\mu^2\sqrt{\epsilon}))$ to find an ϵ-minimum for convex objective, where m is an arbitrary constant and denotes the number of inner iterations for VR. It is worth mentioning that the iterative complexities of S-SVRG are both significantly superior to its non-stagewise counterpart SVRG and existing stagewise algorithm S-SGD under convex and nonconvex conditions.
- We also numerically evaluate our algorithms through the experiments on a number of benchmark datasets. The obtained results are consistent with our theoretical findings.

The remainder of this paper is organized as follows. Section 2 provides some notions and preliminaries. The accelerated methods based on SSTS are proposed and analyzed in Section 3. Experiments performed on several real-world datasets are presented in Section 4.

Discussion about the proposed methods are reported in Section 5. Lastly, Section 6 gives some concluding remarks. All the proofs are presented in the Appendix A.

2. Notions and Preliminaries

In this paper, we use $\|\cdot\|$ to denote a general norm without specific mention. Given any $X \subseteq \mathbb{R}^d$, we say f is Lipschitz continuously differentiable (l.c.d.) with Lipschitz constant $L > 0$ over X, if $\|\nabla f(y) - \nabla f(x)\| \leq L\|y - x\|$ for any $x, y \in X$. What is more, we can also verify the following inequality:

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

We say f is convex over X, if any intermediate value is at most the average value, i.e.,

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y)$$

for any $x, y \in X$ and $\lambda \in (0,1)$. We say f is a PL function or meets the PL condition over X with modulus $\mu > 0$, if

$$2\mu(f(x) - f(x^*)) \leq \|\nabla f(x)\|^2 \tag{3}$$

where x^* is the minimizer of f over X. Note that such a function f need not be convex. However, it is also easy to show that a λ-strongly convex function is a PL function with modulus $1/2\lambda$. Furthermore, for f meeting the PL condition, it holds that

$$\|x - x^*\|^2 \leq \frac{1}{2\mu}(f(x) - f(x^*)). \tag{4}$$

The proof of above inequality can be found in [38] directly.

3. Stagewise Accelerated Algorithms Development

In this section, we propose to consider the SSTS playing an accelerated role in Nesterov's acceleration-based method (RSAG) and VR-based method (SVRG). Particularly, we incorporate SSTS into the iterative framework of RSAG or SVRG to create a new way to accelerate the convergence rate further. Next, let us discuss stagewise RSAG and stagewise SVRG one by one.

3.1. Stagewise RSAG Development

Nesterov's acceleration is devised according to Polyak's heavy ball, the acceleration principle behind which is mainly based on the effectiveness of momentum on physics. Nesterov's acceleration has attracted much interest due to the increasing need to solve large-scale problems. However, Nesterov's acceleration requires explicitly convexity assumption for establishing convergence. Recently, Ghadimi and Lan [19], based on Nesterov's acceleration, redesigned an elaborate iterative framework RSAG. On the other hand, in the process of optimizing nonconvex models such as DNNs, sparse regularization, one empirically finds out a phenomenon that the optimization algorithms equipped with SSTS, which starts from a relatively large stepsize and decreases it geometrically after a number of iterations, can improve the convergence speed effectively. SSTS makes the objective function to fastly find the space of the optimal. Thereupon, SSTS has been viewed as another simply implemented way to accelerate convergence rate.

In this section, we incorporate the SSTS into the iterative framework of RSAG which carries out the following updates (Algorithm 1): In Algorithm 1, the number of total iterations has been divided into T epochs, and the iteration is implemented S_k times at the kth epoch. At the first epoch, we choose a relatively large η_0, and decrease it by half, i.e., $\eta_1 = \eta_0/2$, at the second epoch, and so on. The SSTS procedure contains the T epochs times S_k epoch, which starts from a relatively large stepsize and decreases it geometrically after a number of iterations. The stepsize is decreased in half through each epoch in Algorithm 1.

It is easy to verify that the convergence result also holds when the the stepsize is decayed slower somewhat, i.e., decayed by a factor $1 < c < 2$, $\eta_{k+1} = \eta_k / c$. In particular, the recursion shown in the 6*th* line of Algorithm 1 is the so called RSAG.

Algorithm 1 Stagewise RSAG (S-RSAG).

```
1:  Input: $T \geq 1$, parameters $\{S_k, \eta_k, \alpha_s\}_{k=1}^T \geq 1$ and random variables $R_{S_k}$ taking values in $\{1, \dots, S_k\}$;
2:  Initialize: $x_0 = 0$;
3:  for $k = 0, 1, \dots, T-1$ do
4:      $x_{k,0} = x_{k,0}^{md} = x_{k,0}^{ag} = x_k$;
5:      for $s = 0, 1, \dots, S_k - 1$ do
6:          $\begin{cases} x_{k,s+1}^{md} = (1-\alpha_s) x_{k,s}^{ag} + \alpha_s x_{k,s} \\ x_{k,s+1} = x_{k,s} - \lambda_{k,s} \nabla f_{i_s}(x_{k,s+1}^{md}) \\ x_{k,s+1}^{ag} = x_{k,s}^{md} - \eta_k \nabla f_{i_s}(x_{k,s+1}^{md}) \end{cases}$
7:
8:      end for
9:      $\begin{cases} x_{k+1} = x_{k,R_{S_k}}^{ag}, \text{ if } F \text{ is convex} \\ x_{k+1} = x_{k,R_{S_k}}^{md}, \text{ elsewise} \end{cases}$
10: end for
11: Output: $x_T$.
```

3.2. Theoretical Aspects of S-RSAG

The goal of this section is to show the results of the convergence rate of Algorithm 1. For DNNs, the PL condition has been satisfied and proved. So we assume the objective F meets the PL condition. In particular, we give the following theorem in which the convergence rate of the Algorithm 1 is characterized.

Theorem 1. *Suppose that F is l.c.d. with constant L and meets the PL condition with modulus μ, and its gradient is bounded by G uniformly.*

1. *If the parameters are chosen as $\alpha_s = \frac{2}{s+1}$, $\eta_k \leq \min\left\{\frac{1}{2L}, \frac{\epsilon_k}{8L\sigma^2}\right\}$, $\lambda_{k,s} \in \left[\eta_k, \frac{2s+3}{2(s+1)}\eta_k\right]$, $S_k = \frac{4}{\mu\eta_k}$, and probability mass function of S_k is chosen such that $\mathbb{P}(R_{S_k} = s) = \frac{C_{k,s}\lambda_{k,s}}{\sum_{i=1}^{S_k} C_{k,i}\lambda_{k,i}}$ for any $s = 1, \dots, S_k$, where $\epsilon_0 \geq \|\nabla F(x_0)\|^2$, $\epsilon_{k+1} = \epsilon_k/2$, and $C_{k,s} = 1 - L\left[\lambda_{k,s} + \frac{(\lambda_{k,s}-\eta_k)^2(s+1)^2}{8\lambda_{k,s}(S_k+1)}\right]$ we can find an ϵ-stationary point (a point $\mathring{x}$ such that $\mathbb{E}[\|\nabla F(\mathring{x})\|^2] \leq \epsilon$) by performing Algorithm 1 at most $O(L/(\mu\epsilon))$ times.*
2. *If we further assume F is convex and the parameters are chosen as $\alpha_s = \frac{2}{s+1}$, $\eta_k \leq \min\left\{\frac{1}{L}, \frac{\epsilon_k}{16\sigma^2}\sqrt{\frac{\mu}{3L}}\right\}$, $\lambda_{k,s} = \eta_k$, $S_k = \sqrt{\frac{12}{L\mu\eta_k^2}}$, and probability mass function of S_k is chosen such that $\mathbb{P}(R_{S_k} = s) = \frac{s(s+1)}{\sum_{i=1}^{S_k} i(i+1)}$ for any $s = 1, \dots, S_k$, where $\epsilon_0 \geq F(x_0) - F(x^*)$ and $\epsilon_{k+1} = \epsilon_k/2$, we can find an ϵ-minimizer (a point $\mathring{x}$ such that $\mathbb{E}[F(\mathring{x}) - F(x^*)] \leq \epsilon$) by performing Algorithm 1 at most $O(1/(\mu\epsilon))$ times.*

where σ denotes the upper bound of the standard deviation of the stochastic gradient.

The proof of Theorem 1 is presented in Appendix A.1.

Remark 1. *By the above Theorem 1, the iterative complexities of S-RSAG are at most $O(L/\mu\epsilon)$ to find an ϵ-stationary point for nonconvex objective and $O(1/\mu\epsilon)$ to find an ϵ-minimum for convex* *objective. For RSAG, the complexities are $O(L^2/\epsilon + L/\epsilon^2)$ and $O(L/\sqrt{\epsilon} + 1/\epsilon^2)$, respectively, where μ is the PL modulus and L is the Lipschitz constant. S-RSAG reduces the complexity $O(1/\epsilon^2)$ to $O(1/\epsilon)$ under convex and nonconvex conditions. Compared to existing stagewise algorithm S-SGD, the complexities of our S-RSAG are more superior under a convex condition (i.e., $O(1/\mu\epsilon)$ v.s. $O(L/\mu\epsilon)$, where $L \gg 1$ in general) and at the same level under a nonconvex condition. The detail comparison is reported in Table 1. Meanwhile, our algorithm S-RSAG does not deny the optimality of RSAG for first-order stochastic gradient methods, and further considers the PL condition.*

Table 1. Some recent results in the accelerated stochastic gradient methods.

Methods	PL Condition	Generally Convex	Nonconvex
SGD [16]		$O(L/\epsilon + \sigma/\epsilon^2)$	$O(L^2/\epsilon + L\sigma/\epsilon^2)$
SGD [39]	✓		$O(1/(\mu^2\epsilon))$
RSAG [19]		$O(L/\sqrt{\epsilon} + \sigma/\epsilon^2)$	$O(L^2/\epsilon + L\sigma/\epsilon^2)$
SVRG [21]			$O(n^{2/3}L/\epsilon)$
S-SGD [37]	✓	$O(L/\mu\epsilon)$	$O(L/\mu\epsilon)$
S-RSAG	✓	$O(1/\mu\epsilon)$	$O(L/\mu\epsilon)$
S-SVRG	✓	$O(Lm/(\mu^2\sqrt{\epsilon}))$	$O(Lm/(\mu\sqrt{\epsilon}))$

3.3. Stagewise SVRG Development

As discussed earlier, because SGD can only tolerate a relatively small stepsize, it suffers from a slow convergence rate consequently. Apart from Nesterov's method, another way (VR) is to implement acceleration by reducing the variance of random interference. The core ideal of the VR technique is realized by calculating a full gradient once at each epoch and incorporating it into the iteration to adjust the current stochastic gradient so that it does not deviate too far away from the full one. Benefiting from the VR technique, SVRG has been shown to possess strong competitiveness with respect to fast convergence rate.

In this section, we mainly consider whether the convergence rate can be further improved by combining the VR technique and SSTS together. In particular, we incorporate the SSTS into the iterative framework of SVRG, which carries out the following updates (Algorithm 2):

Algorithm 2 Stagewise SVRG (S-SVRG).

```
1:  Input: $T \geq 1$, $m \geq 1$, parameters $\{S_k, \eta_k\}_{k=1}^{T} \geq 1$ and random variables $R_{S_k}$ taking
    values in $\{1, \dots, S_k\}$;
2:  Initialize: $x_0 = 0$;
3:  for $k = 0, 1, \dots, T-1$ do
4:      $x_{k,0} = x_k$;
5:      for $s = 0, 1, \dots, S_k - 1$ do
6:          $x_{k,s}^0 = x_{k,s}$;
7:          for $t = 0, 1, \dots, m-1$ do
8:              $x_{k,s}^{t+1} = x_{k,s}^t - \eta_k[\nabla f_{i_t}(x_{k,s}^t) - \nabla f_{i_t}(x_{k,s}^0) + \nabla F(x_{k,s}^0)]$;
9:          end for
10:         $x_{k,s+1} = x_{k,s}^m$;
11:     end for
12:     $x_{k+1} = x_{k,R_{S_k}}$;
13: end for
14: Output: $x_T$.
```

In Algorithm 2, m is an arbitrary constant and denotes the number of inner iteration for VR, and the rest parameters are set as in Algorithm 1. In particular, the process of the so-called VR technique has been shown in the 7–9th line of Algorithm 2.

3.4. Theoretical Aspects of S-SVRG

The goal of this section is to show the results of the convergence rate of Algorithm 2. In particular, we give the following theorem in which the convergence rate of the Algorithm 2 is characterized.

Theorem 2. *Suppose that F is l.c.d. with constant L and meets the PL condition with modulus μ, and its gradient is bounded by G uniformly.*

1. *If the parameters are chosen as $\eta_k \le \min\left\{\frac{1}{2mGL}\sqrt{\frac{(1-m^2\eta_0^2L^2)\epsilon_k}{4+m\eta_0 L}}, \frac{1}{2mL}\right\}$, $S_k = \frac{4}{\mu m\eta_k(1-2\eta_0 mL)}$, and probability mass function of R_{S_k} is chosen such that $\mathbb{P}(R_{S_k} = s) = \frac{1}{S_k}$ for any $s = 0, \ldots, S_k - 1$, where $\epsilon_0 \ge \|\nabla F(x_0)\|^2$ and $\epsilon_{k+1} = \epsilon_k/2$, we can find an ϵ-stationary point (a point $\mathring{x}$ such that $\mathbb{E}[\|\nabla F(\mathring{x})\|^2] \le \epsilon$) by performing Algorithm 2 at most $O(Lm/(\mu\sqrt{\epsilon}))$ times.*
2. *If we further assume F is convex and the parameters are chosen as $S_k = \frac{1+\log(4)}{\mu m\eta_k}$, $\eta_k \le \min\left\{\frac{\mu\sqrt{(1-m^2\eta_0^2L^2)\epsilon_k}}{2mGL}, \frac{1}{mL}, \frac{1}{\mu m}\right\}$ and probability mass function are chosen such that $\mathbb{P}(R_{S_k} = S_k) = 1$ and $\mathbb{P}(R_{S_k} = s) = 0$ for $s = 1, \ldots, S_k - 1$, where $\epsilon_0 \ge \|x_0 - x^*\|^2$ and $\epsilon_{k+1} = \epsilon_k/2$, we can find an ϵ-minimizer (a point $\mathring{x}$ such that $\|\mathring{x} - x^*\|^2 \le \epsilon$) by performing Algorithm 1 at most $O(mL/(\mu^2\sqrt{\epsilon}))$ times.*

The proof of Theorem 2 is presented in Appendix A.2.

Remark 2. *From the above theorem, it easy to verify that the iterative complexities of S-SVRG have been significantly reduced, which are both more superior than its non-stagewise counterpart SVRG and existing stagewise algorithm S-SGD under convex and nonconvex conditions. In other words, the convergence rates of S-SVRG have been significantly improved by SSTS. A detailed comparison is reported in Table 1.*

So far, we have answered the main question considered in this paper in the affirmative; namely, the incorporation between SSTS and RSAG or SVRG does further improve the convergence rate.

4. Numerical Experiments

In the previous sections, we proposed two stagewise algorithms S-RSAG and S-SVRG, and analyzed the acceleration of their convergence rate. Now, we turn to consider their experimental performances.

4.1. Learning DNNs

In this subsection, we focus on testing our algorithm under the nonconvex condition, i.e., training DNNs. Particularly, we choose two familiar networks, MLP and VGG net, to examine the performances of our algorithms. Note that we are not trying to show that these two networks are the most efficient, but are attempting to show the superiority of our algorithms based on these two nonconvex models.

Firstly, we compare our stagewise algorithms S-RSAG and S-SVRG with their non-stagewise counterparts RSAG and SVRG. Then, we also compare them with the other state-of-art methods including stagewise SGD (S-SGD), VR-SGD [22], and Katyusha [40]. Experiments are performed on two commonly used datasets:

1. MNIST: This dataset contains 28×28 gray images from ten-digit classes. To improve learning efficiency, we load 10 samples per batch. We use 60,000 (6000×10) images for training, and the remaining 10,000 for testing. We adopt the 4-layer MLP network

$$784FC - 2048FC - 1024FC - 512FC - 256FC - 10SF$$

to training, where FC denotes a ReLU full-connected layer and SF denotes a softmax output layer, for which the "CrossEntropyLoss" is adopted.

2. CIFAR-10: This dataset contains 32×32 color images from ten object classes. Similarly, we load 4 samples per batch. We use 50,000 (12,500 $\times$ 4) images for training, and the remaining 10,000 for testing. We adopt the VGG-like architecture

$$\begin{aligned} &32 \times 32 \times 3C3 - MP2 - 16 \times 16 \times 64C3 - MP2 \\ &- 8 \times 8 \times 128C3 - MP2 - 4 \times 4 \times 512C3 - MP2 \\ &- 2 \times 2 \times 256C3 - MP2 - 10FC \end{aligned}$$

where $C3$ denotes a 3×3 ReLU convolution layer, and $MP2$ denotes a 2×2 max-pooling layer, and FC denotes a ReLU full-connected output layer, for which the "CrossEntropyLoss" is adopted.

In this section, the initial stepsize values for stagewise algorithms S-RSAG, S-SVRG and S-SGD are set as $\eta_0 = 0.5$ for MLP and $\eta_0 = 0.05$ for VGG. The iterations of these algorithms are divided into 5 epochs ($T = 5$) for MLP and 10 epochs ($T = 10$) for VGG. At each epoch, we run the iterations over the entire training set ergodicly, namely, the number of inner iterations at each epoch satisfies $S_k = 6000$ for MLP and S_k = 12,500 for VGG. At last, the stepsize for MLP decays by a factor of 2 and for VGG it decays by a factor of 1.5. In addition, the more the algorithm can tolerate large stepsize, the faster the convergence rate of the algorithm is, usually [22]. For other comparison algorithms, we tried several times to select a large as possible stepsize under the premise of ensuring convergence. We evaluate the performances of these algorithms in three aspects, i.e., value of loss, classification error rate on training set (training accuracy), and rate on testing set (testing accuracy). Next, we design the two following comparative experiments to verify our previous claims.

4.1.1. S-RSAG and S-SVRG vs. Their Non-Stagewise Counterparts

In this test, we attempt to verify the effectiveness of SSTS via comparing our stagewise algorithms S-RSAG and S-SVRG with their non-stagewise counterparts RSAG and SVRG. Figure 1 shows the behaviors of all the algorithms considered, in the three aspects (i.e., value of loss, training error, and testing error). Let us take a close look at the decay of loss function values: it is easy to find that the proposed S-RSAG and S-SVRG converge faster than their non-stagewise counterparts RSAG and SVRG, respectively. With respect to the training and testing accuracy, in most cases, our proposed RSAG and SVRG also can achieve the best accuracy more quickly.

4.1.2. S-RSAG and S-SVRG vs. Other Methods

In the above test, we have shown the effectiveness of SSTS via comparing the performances of our proposed algorithms with their non-stagewise counterparts. In this section, we verify whether the combination of SSTS and Nesterov's acceleration or VR can further accelerate the convergence rate. Specifically, for the full-gradient-free S-RSAG, we compare it with other full-gradient-free method S-SGD; for the full-gradient-calibration S-SVRG, we compare it with VR-SGD and Katyusha.

Figure 2 shows the behaviors of all the algorithms considered. For full-gradient-free methods, as can be seen, S-RSAG outperforms S-SGD obviously in all the three aspects we considered. For full-gradient-calibration methods, we can see that our proposed S-SVRG achieves the fastest convergence rate with respect to the value of loss. Since our S-SVRG takes the last iteration as the output, the stability of the model is susceptible to random noise interference. From Table 2, it can be seen that S-SVRG and S-RASG have lower computational times. Therefore, S-SVRG's performance in terms of training and testing accuracy is better than other methods.

In the end, from the above numerical results, it can be seen that the combination of SSTS with Nesterov's acceleration or VR does further accelerate the convergence rate under nonconvex optimization. Next, we will examine the performances of our proposed algorithms under the convex condition.

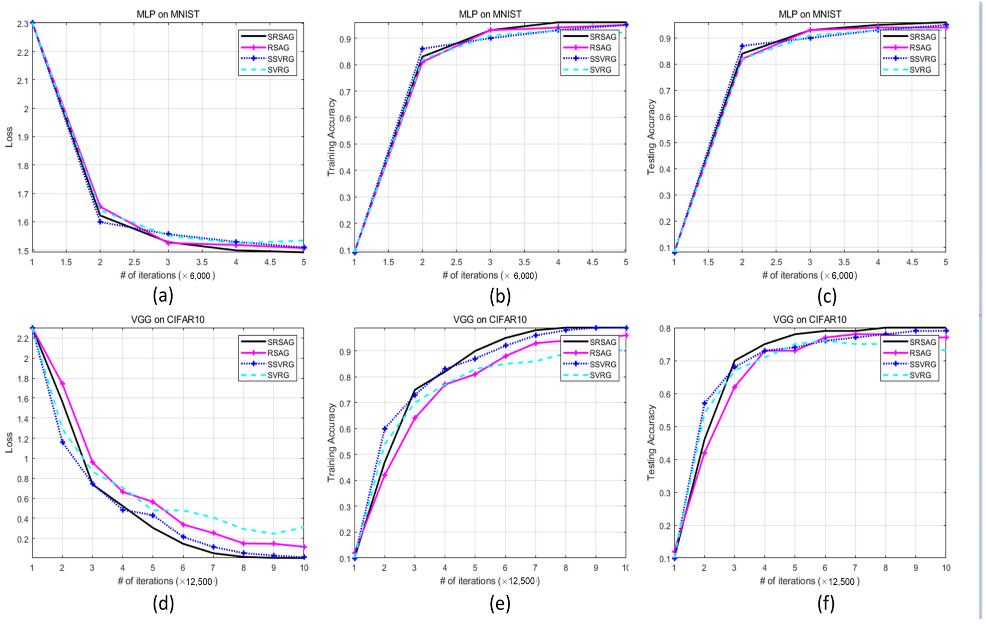

Figure 1. Performances of S-SVRG, S-RSAG, SVRG and RSAG on different nonconvex tasks. (**a**–**c**) respectively demonstrate the loss, training accuracy, and testing accuracy of different methods for MLP networks on MNIST dataset; (**d**–**f**) respectively plot the loss, training accuracy, and testing accuracy of VGG networks on CIFAR-10 dataset.

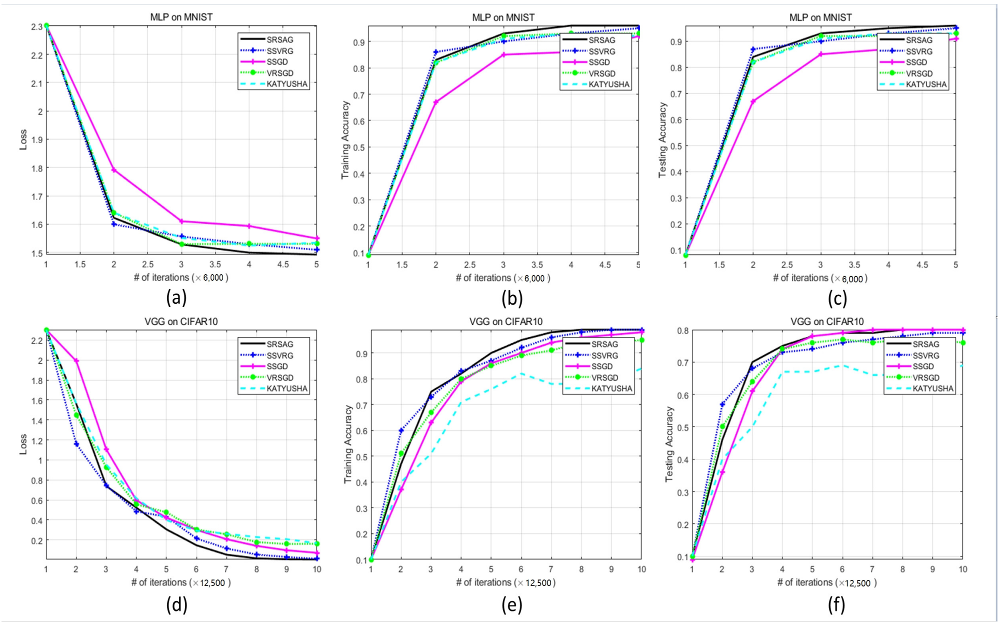

Figure 2. Performances of S-SVRG, S-RSAG, S-SGD, VR-SGD, and Katyusha for various nonconvex tasks. (**a**–**c**) respectively demonstrate the loss, training accuracy, and testing accuracy of different methods for MLP networks on MNIST dataset; (**d**–**f**) respectively plot the loss, training accuracy, and testing accuracy of VGG networks on CIFAR-10 dataset.

Table 2. The computational time (s) for different methods in the nonconvex condition.

Methods	**SGD**	**RASG**	**SVRG**	**S-SGD**	**Katyusha**	**S-RSAG**	**S-SVRG**
MLP on MNIST	247.57	685.94	292.72	293.61	435.22	293.15	326.78

4.2. Logistic Regression

In the above section, we have examined the performance of our proposed algorithms under the nonconvex condition (training DNNs). In this section, we test our algorithms on logistic regression problem , which is under the convex condition and can be viewed as a one-layer fully connected network with SoftMarginLoss. Experiments are performed on two commonly used datasets downloaded from the libsvm website (https://www.csie.ntu.edu.tw/~cjlin/libsvm/).

1. REAL-SIM: This contains 72,309 data points of 20,958 features from two object classes. We divide it into two sets, i.e., one for training and the other for testing.
2. RCV1: This contains 20,242 data points of 47,236 features from two object classes. Similarly, we also divide it into two sets, averagely, one for training and the other for testing.

In this section, the initial values of stepsize for stagewise algorithm S-RSAG, S-SVRG and S-SGD are chosen as $\eta_0 = 5$. The iterations are divided into 5 epochs ($T = 5$). At each epoch, we set the batch size as 10 and run the iterations over the entire training set ergodicly, namely, the number of inner iteration at each epoch satisfies $S_k = 1500$ for RCV1 and $S_k = 5000$ for REAL-SIM. For other comparison algorithms, we tried several times to select as large as possible a stepsize under the premise of ensuring convergence. We evaluate the performances of these algorithms in the aspects of loss, training, and testing accuracy.

4.2.1. S-RSAG and S-SVRG vs. Their Non-Stagewise Counterparts

In this test, we attempt to verify the effectiveness of SSTS under the convex condition. Figure 3 shows the behaviors of all the algorithms considered. With respect to the decay of loss function values, it is easy to find that the proposed S-RSAG and S-SVRG converge faster than their non-stagewise counterparts RSAG and SVRG, respectively. With respect to the training accuracy, S-RSAG and S-SVRG also outperform their non-stagewise counterparts. As shown in subfigure (f), with respect to the testing accuracy, our proposed methods are inferior to the comparison methods. This is due to overfitting; that is, the faster the algorithm converges, the weaker the generalization ability is.

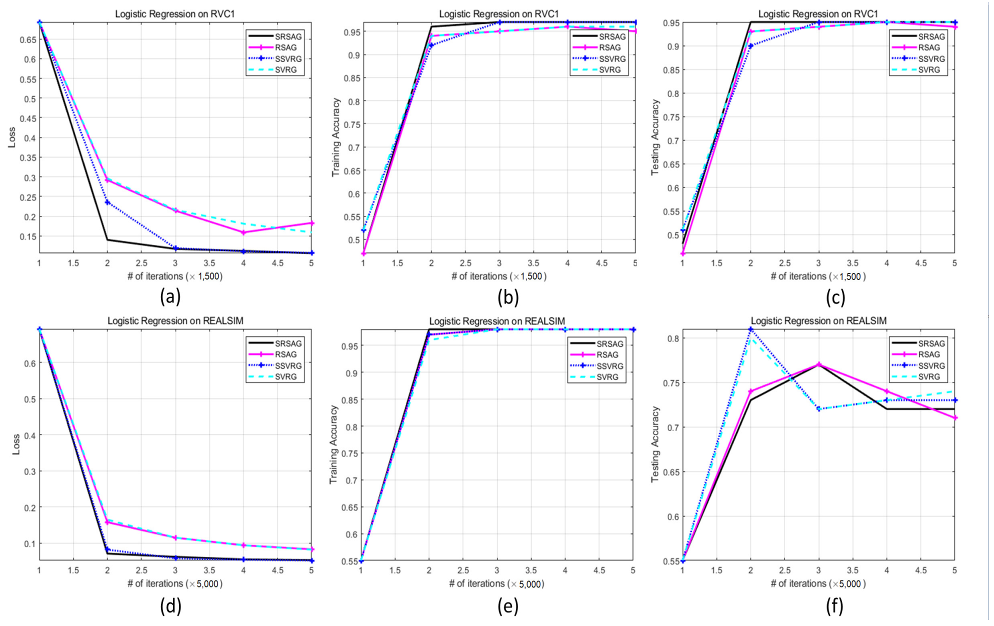

Figure 3. Performances of S-SVRG, S-RSAG, SVRG, and RSAG on convex condition. (**a**–**c**) demonstrate the values of loss, training accuracy, and testing accuracy of various methods for logistic regression on RVC1 dataset. (**d**–**f**) demonstrate the values of loss, training accuracy, and testing accuracy of various methods for logistic regression on REAL-SIM dataset.

4.2.2. S-RSAG and S-SVRG vs. S-SGD

Figure 4 shows the behaviors of all the algorithms considered. As can be seen, S-RSAG outperforms S-SGD, and S-SVRG achieves the best performance other than full-gradient-calibrated methods, in most cases. We have verified the effectiveness of SSTS under the convex condition.

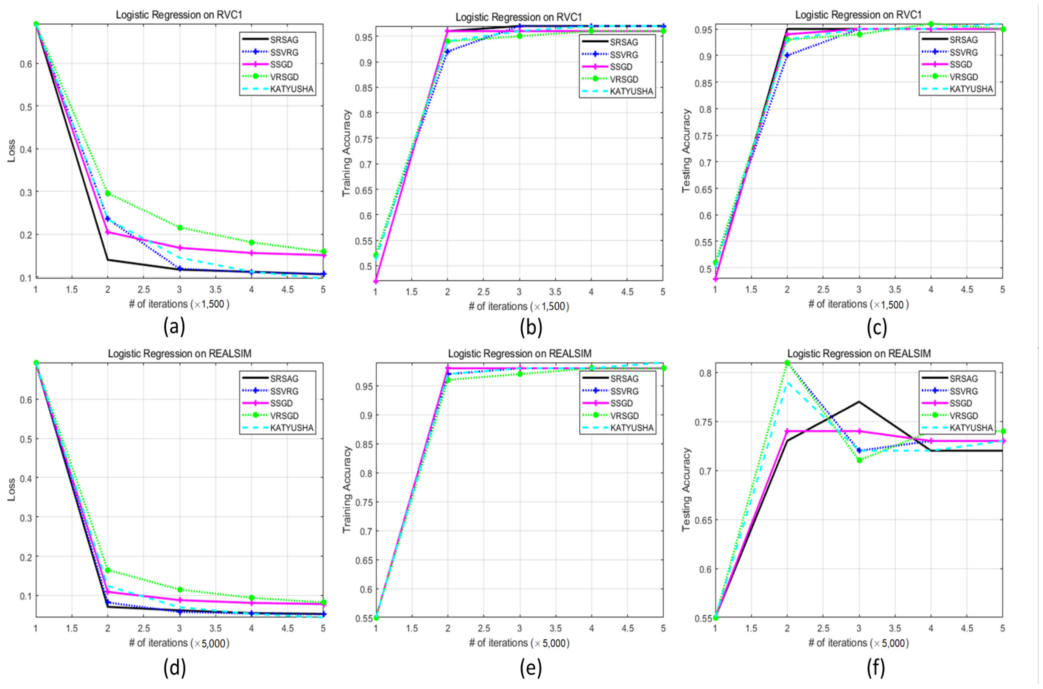

Figure 4. Performances of S-SVRG, S-RSAG, S-SGD, VR-SGD, and Katyusha on convex condition. (**a**–**c**) demonstrate the values of loss, training accuracy, and testing accuracy of various methods for logistic regression on RVC1 dataset. (**d**–**f**) demonstrate the values of loss, training accuracy, and testing accuracy of various methods for logistic regression on REAL-SIM dataset.

5. Discussion

Inspired by the phenomenon of optimization algorithms equipped into SSTS and the continuous strategy in nonconvex optimization, we incorporated SSTS into the iterative framework of RSAG and SVRG, and proposed S-RSAG and S-SVRG. We show that the iterative complexities of S-RSAG are significantly reduced with respect to its non-stagewise counterpart RSAG, and are more superior than the existing stagewise algorithm S-SGD under the convex condition and at the same level under the nonconvex condition, and the iterative complexities of S-SVRG are significantly superior than both its non-stagewise counterpart SVRG and the existing stagewise algorithm S-SGD under convex and nonconvex conditions. We will further discuss the lower bound for our proposed methods and explore the performance of gradient in higher-dimensional constrained optimization in the future.

6. Conclusions

In this paper, we mainly proposed a conjecture as to whether the incorporation between SSTS, which is a common method to train DNNs empirically, and Nesterov's acceleration or VR-based methods can further improve the convergence rate. Particularly, we propose two SSTS equipped accelerated algorithms and answered the above conjecture in the affirmative theoretically under both convex and nonconvex conditions, respectively. Furthermore, we examine the performance of our equipped algorithms by designing contrast experiments on training DNNs and logistic regression, which validates the competitiveness of our methods well.

Author Contributions: Conceptualization, C.J. and Z.C.; investigation, C.J.; methodology, C.J. and Z.C.; project administration, C.J. and Z.C.; validation, C.J. and Z.C.; writing—original draft, C.J.; writing and review and editing, C.J. and Z.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by General project of Zhejiang provincial Department of Education with No. Y202147627.

Data Availability Statement: We choose to exclude this statement because the study did not report any data.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

GD	gradient descent
SGD	stochastic gradient descent
SSTS	stagewise stepsize tuning strategy
DNN	deep neural network
VR	variance reduction
RASG	randomized stochastic accelerated gradient
SVRG	stochastic variance reduced gradient
S-SGD	stagewise stochastic gradient descent
S-RSAG	stagewise randomized stochastic accelerated gradient
S-SVRG	stagewise stochastic variance reduced gradient
l.c.d.	Lipschitz continuously differentiable
PL	Polyak–Łojasiewicz
MLP	Mmultilayer Perceptron
VGG	Visual Geometry Group
FC	a ReLu fully connected layer
SF	softmax output layer

Appendix A

Appendix A.1. Proof of Theorem 1

Proof. At first, let us review the following convergence result about RSAG in the proof of Corollary 3 [19]. Suppose F and parameters coincide with the assumptions made in Theorem 1; then, the iterate generated by Algorithm 1 satisfies the following inequality

$$\mathbb{E}[\|\nabla F(x_{k+1})\|^2 | x_k] \leq \frac{2[F(x_k) - F(x^*)]}{\eta_k S_k} + 2L\sigma^2\eta_k. \tag{A1}$$

When F is further assumed to be convex, we have

$$\mathbb{E}[F(x_{k+1}) - F(x^*) | x_k] \leq \frac{12\|x_k - x^*\|}{L\eta_k^2 S_k^2} + 2L\sigma^2\eta_k^2 S_k \tag{A2}$$

where σ denotes the upper bound of the standard deviation of the stochastic gradient.

We firstly show part (1). We prove the result $\mathbb{E}[\|\nabla F(x_{k+1})\|^2] \leq \epsilon_{k+1}$ by induction, where $\epsilon_{k+1} := \epsilon_k/2$, which is true for $k = 0$ as long as the initial value ϵ_0 is chosen as $\epsilon_0 := \|\nabla F(x_0)\|^2$. We assume $\mathbb{E}[\|\nabla F(x_k)\|^2] \leq \epsilon_k$ is true and propose to prove this inequality holds at $k+1$. Plugging PL inequality (3) into (A1), we have

$$\begin{aligned}\mathbb{E}[\|\nabla F(x_{k+1})\|^2] \leq & \frac{\mathbb{E}[\|\nabla F(x_k)\|^2]}{\mu\eta_k S_k} + 2L\sigma^2\eta_k \\ & \frac{\epsilon_k}{\mu\eta_k S_k} + 2L\sigma^2\eta_k\end{aligned}$$

Since the parameters are chosen as in Theorem 1, we obtain $\mathbb{E}[\|F(x_{k+1})\|^2] \le \epsilon_{k+1}$. By induction, after $T = \lceil \log_2(\epsilon_0/\epsilon) \rceil$ stages, we have $\mathbb{E}[\|F(x_T)\|^2] \le \epsilon$, with total iterative complexity $\sum_{k=1}^{T} S_k = \sum_{k=1}^{T} O(L/(\mu\epsilon_k)) \le O(L/(\mu\epsilon))$.

We now show part (2). With the same method used above, we prove result $\mathbb{E}[F(x_{k+1}) - F(x^*)] \le \epsilon_{k+1}$ by induction. We assume $\mathbb{E}[F(x_k) - F(x^*)] \le \epsilon_k$ is true. Plugging PL inequality (4) into (A2), we have

$$\begin{aligned}\mathbb{E}[F(x_{k+1}) - F(x^*)] \le& \frac{6\mathbb{E}[F(x_k) - F(x^*)]}{L\mu\eta_k^2 S_k^2} + 2L\sigma^2\eta_k^2 S_k \\ \le& \frac{3\epsilon_k}{L\mu\eta_k^2 S_k^2} + 2L\sigma^2\eta_k^2 S_k\end{aligned}$$

Since the parameters are chosen as in Theorem 1, we obtain $\mathbb{E}[F(x_{k+1}) - F(x^*)] \le \epsilon_{k+1}$. Similar to the proof of the above part, it is easy to verify that the iterative complexity of Algorithm under convex condition is $O(1/(\mu\epsilon))$. □

Appendix A.2. Proof of Theorem 2

Proof. We first show part (1) by following two steps:

First step: we will show that the following inequality holds for any $0 \le k \le T$:

$$\begin{aligned}&\mathbb{E}[\|\nabla F(x_{k+1})\|^2 | x_k] \\ \le& \frac{2}{1 - 2\eta_k mL}\left[\frac{F(x_k) - F^*}{m\eta_k S_k} + \frac{m^2\eta_k^2 G^2 L^2 (4 + m\eta_k L)}{1 - m^2\eta_k^2 L^2}\right].\end{aligned} \tag{A3}$$

Now we begin proving the above inequality (A3). Following the recursion (line 8 in Algorithm 2) directly, we have

$$\begin{aligned}x_{k,s+1} = x_{k,s}^m =& x_{k,s}^{m-1} - \eta_k[\nabla f_{i_{m-1}}(x_{k,s}^{m-1}) - \nabla f_{i_{m-1}}(x_{k,s}^0) \\ &+ \nabla F(x_{k,s}^0)] \\ &\vdots \\ =& x_{k,s}^j - \eta_k \sum_{t=1}^{m-j} [\nabla f_{i_{m-t}}(x_{k,s}^{m-t}) - f_{i_{m-t}}(x_{k,s}^0) \\ &+ \nabla F(x_{k,s}^0)] \\ &\vdots \\ =& x_{k,s}^0 - \eta_k \sum_{t=0}^{m-1} [\nabla f_{i_t}(x_{k,s}^t) - \nabla f_{i_t}(x_{k,s}^0) \\ &+ \nabla F(x_{k,s}^0)].\end{aligned}$$

We further define $g_{k,s}^t(x_{k,s}^t) := \nabla f_{i_t}(x_{k,s}^t) - \nabla f_{i_t}(x_{k,s}^0)$; then the above recursion can be rewritten as

$$x_{k,s+1} = x_{k,s} - \eta_k m \nabla F(x_{k,s}) - \eta_k \sum_{t=0}^{m-1} g_{k,s}^t(x_{k,s}^t), \tag{A4}$$

where we make use of the notion $x_{k,s}^0 = x_{k,s}$.

Since f is Lipschitz continuously differentiable with constant L, it holds that

$$\begin{aligned}F(x_{k,s+1}) \le& F(x_{k,s}) + \langle \nabla F(x_{k,s}), x_{k,s+1} - x_{k,s} \rangle \\ &+ \frac{L}{2}\|x_{k,s+1} - x_{k,s}\|^2.\end{aligned}$$

By the recursion (A4) and above inequality, we have

$$\begin{aligned}
&F(x_{k,s+1}) \\
\leq &F(x_{k,s}) + \frac{\eta_k^2 L}{2}\left\| m\nabla F(x_{k,s}) + \sum_{t=0}^{m-1} g_{k,s}^t(x_{k,s}^t)\right\|^2 \\
&- \eta_k \left\langle \nabla F(x_{k,s}), m\nabla F(x_{k,s}) + \sum_{t=0}^{m-1} g_{k,s}^t(x_{k,s}^t)\right\rangle \\
\leq &F(x_{k,s}) - \left(\frac{\eta_k m}{2} - \eta_k^2 m^2 L\right)\|\nabla F(x_{k,s})\|^2 \\
&+ \left(\frac{4\eta_k}{m} + \eta_k^2 L\right)\left\|\sum_{t=0}^{m-1} g_{k,s}^t(x_{k,s}^t)\right\|^2,
\end{aligned} \tag{A5}$$

where the last inequality follows the Cauchy–Schwarz inequality.

Now, we aim to give the upper bound for the term $\left\|\sum_{t=0}^{m-1} g_{k,s}^t(x_{k,s}^t)\right\|^2$. Via direct computation, we have

$$\begin{aligned}
\left\|\sum_{t=0}^{m-1} g_{k,s}^t(x_{k,s}^t)\right\|^2 \leq &m\sum_{t=0}^{m-1}\left\|g_{k,s}^t(x_{k,s}^t)\right\|^2 \\
\leq &mL^2\sum_{t=0}^{m-1}\|x_{k,s}^t - x_{k,s}^0\|^2 \\
= &mL^2\sum_{t=0}^{m-1}\left\|\sum_{j=1}^{t}(x_{k,s}^j - x_{k,s}^{j-1})\right\|^2 \\
\leq &mL^2\sum_{t=0}^{m-1} t\sum_{j=1}^{t}\|x_{k,s}^j - x_{k,s}^{j-1}\|^2 \\
= &mL^2\sum_{j=1}^{m-1}\sum_{t=j}^{m-1} t\|x_{k,s}^j - x_{k,s}^{j-1}\|^2 \\
\leq &\frac{m^3L^2}{2}\sum_{j=1}^{m}\|x_{k,s}^j - x_{k,s}^{j-1}\|^2
\end{aligned} \tag{A6}$$

where we make use of the notion

$$\sum_{t=j}^{m-1} t \leq \sum_{t=1}^{m-1} t = \frac{m(m-1)}{2} < \frac{m^2}{2}$$

for the last inequality. On the other hand, we have

$$\begin{aligned}
&\|x_{k,s}^t - x_{k,s}^{t-1}\|^2 \\
= &\eta_k^2\|\nabla f_{i_{t-1}}(x_{k,s}^{t-1}) - \nabla f_{i_{t-1}}(x_{k,s}^0) + \nabla F(x_{k,s}^0)\|^2 \\
\leq &2\eta_k^2\|\nabla f_{i_{t-1}}(x_{k,s}^{t-1}) - \nabla f_{i_{t-1}}(x_{k,s}^0)\|^2 + 2\eta_k^2\|\nabla F(x_{k,s}^0)\|^2 \\
\leq &2\eta_k^2 L^2\|x_{k,s}^{t-1} - x_{k,s}^0\|^2 + 2\eta_k^2 G^2
\end{aligned}$$

where we make use of the assumption $\|\nabla F(x)\| \leq G$ for the last inequality. Taking summation over t from 1 to m on both sides, we obtain

$$\begin{aligned}\sum_{t=1}^{m}\|x_{k,s}^{t}-x_{k,s}^{t-1}\|^2 \le& 2\eta_k^2L^2\sum_{t=1}^{m}\|x_{k,s}^{t-1}-x_{k,s}^{0}\|^2+2S_k\eta_k^2G^2\\ \le& m^2\eta_k^2L^2\sum_{t=1}^{m}\|x_{k,s}^{t}-x_{k,s}^{t-1}\|^2+2S_k\eta_k^2G^2\end{aligned}$$

which derives

$$\sum_{t=1}^{m}\|x_{k,s}^{t}-x_{k,s}^{t-1}\|^2 \le \frac{2m\eta_k^2G^2}{1-m^2\eta_k^2L^2}. \tag{A7}$$

Substituting (A7) into (A6), we have

$$\left\|\sum_{t=0}^{m-1}g_{k,s}^{t}(x_{k,s}^{t})\right\|^2 \le \frac{m^4\eta_k^2G^2L^2}{1-m^2\eta_k^2L^2}. \tag{A8}$$

Then, substituting (A8) into (A5), we have

$$\begin{aligned}&\left(\frac{\eta_k m}{2}-\eta_k^2m^2L\right)\|\nabla F(x_{k,s})\|^2\\ \le& F(x_{k,s})-F(x_{k,s+1})+\frac{m^3\eta_k^3G^2L^2(4+m\eta_kL)}{1-m^2\eta_k^2L^2}.\end{aligned}$$

Taking summation over s from 0 to S_k-1, we have

$$\begin{aligned}&\left(\frac{\eta_k m}{2}-\eta_k^2m^2L\right)\sum_{s=0}^{S_k-1}\|\nabla F(x_{k,s})\|^2\\ \le& F(x_{k,0})-F(x_{k,S_k})+\frac{S_km^3\eta_k^3G^2L^2(4+m\eta_kL)}{1-m^2\eta_k^2L^2}.\end{aligned}$$

Dividing both sides of the above inequality by $(\eta_k m/2-\eta_k^2m^2L)$ and noting that

$$\begin{aligned}&\mathbb{E}[\|\nabla F(x_{k+1})\|^2]\\ =&\mathbb{E}[\|\nabla F(x_{k,R_{S_k}})\|^2]=\frac{1}{S_k}\sum_{s=0}^{S_k-1}\|\nabla F(x_{k,s})\|^2,\end{aligned}$$

we conclude

$$\begin{aligned}\mathbb{E}[\|\nabla F(x_{k+1})\|^2|x_k] \le& \frac{2}{1-2\eta_kmL}\left[\frac{F(x_k)-F(x_{k,S_k})}{m\eta_kS_k}\right.\\ &\left.+\frac{m^2\eta_k^2G^2L^2(4+m\eta_kL)}{1-m^2\eta_k^2L^2}\right].\end{aligned}$$

By using the notion $F(x_k)-F(x_{k,S_k})\le F(x_k)-F^*$, the inequality (A3) is yielded.

Second step: We prove $\mathbb{E}[\|\nabla F(x_{k+1})\|^2]\le\epsilon_{k+1}$ by induction, where $\epsilon_{k+1}:=\epsilon_k/2$, which is true for $k=0$ as long as the initial value ϵ_0 is chosen as $\epsilon_0:=\|\nabla F(x_0)\|^2$. We assume $\mathbb{E}[\|\nabla F(x_k)\|^2]\le\epsilon_k$ is true and propose to prove this inequality holds at $k+1$. Plugging PL inequality (3) into (A3), we have

$$\begin{aligned}&\mathbb{E}[\|\nabla F(x_{k+1})\|^2]\\ \le&\frac{2}{1-2\eta_kmL}\left[\frac{\mathbb{E}[\|\nabla F(x_k)\|^2]}{2\mu m\eta_kS_k}+\frac{m^2\eta_k^2G^2L^2(4+m\eta_kL)}{1-m^2\eta_k^2L^2}\right]\\ \le&\frac{2}{1-2\eta_kmL}\left[\frac{\epsilon_k}{2\mu m\eta_kS_k}+\frac{m^2\eta_k^2G^2L^2(4+m\eta_kL)}{1-m^2\eta_k^2L^2}\right]\end{aligned}$$

Since the parameters are chosen as in Theorem 2, we obtain $\mathbb{E}[\|\nabla F(x_{k+1})\|^2] \leq \epsilon_{k+1}$. By induction, after $T = \lceil \log_2(\epsilon_0/\epsilon) \rceil$ stages, we have $\mathbb{E}[\|\nabla F(x_T)\|^2] \leq \epsilon$, with total iterative complexity $\sum_{k=1}^{T} mS_k = \sum_{k=1}^{T} O(mL/(\mu\sqrt{\epsilon_k})) \leq O(mL/\sqrt{\mu^2\epsilon})$.

We now show part (2). By recursion (A4), let $d_{k,s} := x_{k,s} - x^*$, and we have

$$\begin{aligned}&\|d_{k,s+1}\|^2\\ =&\left\|d_{k,s} - \eta_k m\nabla F(x_{k,s}) - \eta_k \sum_{t=0}^{m-1} g_{k,s}^t(x_{k,s}^t)\right\|^2\\ \leq&\frac{1}{\gamma}\|d_{k,s} - \eta_k m\nabla F(x_{k,s})\|^2 + \frac{\eta_k^2}{1-\gamma}\left\|\sum_{t=0}^{m-1} g_{k,s}^t(x_{k,s}^t)\right\|^2\end{aligned}$$

where we make use of the notion $(a+b)^2 \leq \frac{1}{\gamma}a^2 + \frac{1}{1-\gamma}b^2$ with $\gamma \in (0,1)$ for the inequality. The upper bound for the second term of the right side of the above inequality can be found in (A8). We now propose to bound the first term as follows:

$$\begin{aligned}&\|d_{k,s} - \eta_k m\nabla F(x_{k,s})\|^2\\ =&\|d_{k,s}\|^2 + \eta_k^2 m^2\|\nabla F(x_{k,s})\|^2 - 2\eta_k m\langle d_{k,s}, \nabla F(x_{k,s})\rangle\\ \leq&(1 - 2\mu m\eta_k)\|d_{k,s}\|^2 - \left(\frac{\eta_k m}{L} - \eta_k^2 m^2\right)\|\nabla F(x_{k,s})\|^2\\ \leq&(1 - \mu m\eta_k)^2\|d_{k,s}\|^2 - \left(\frac{\eta_k m}{L} - \eta_k^2 m^2\right)\|\nabla F(x_{k,s})\|^2\end{aligned}$$

where we make use of the notion $\langle\nabla F(x) - \nabla F(y), x - y\rangle \geq \frac{1}{L}\|\nabla F(x) - \nabla F(y)\|^2$, and $-\langle d_{k,s}, \nabla F(x_{k,s})\rangle \leq f(x^*) - f(x_{k,s}) \leq -2\mu\|x^* - x_{k,s}\|^2$ for the first inequality. Let $\gamma = 1 - \mu m\eta_k$, we have

$$\|d_{k,s+1}\|^2 = (1 - \mu m\eta_k)\|d_{k,s}\|^2 + \frac{m^3\eta_k^3 G^2 L^2}{\mu(1 - m^2\eta_k^2 L^2)}$$

With a simple derivation, we have

$$\begin{aligned}\|d_{k,S_k}\|^2 =&(1 - \mu m\eta_k)^{S_k}\|d_{k,0}\|^2\\ &+ \frac{2m^3\eta_k^3 G^2 L^2}{\mu(1 - m^2\eta_k^2 L^2)}\sum_{s=0}^{S_k-1}(1 - \mu m\eta_k)^{S_k-s-1}\\ \leq&\exp(1 - \mu m\eta_k S_k)\|d_{k,0}\|^2 + \frac{m^2\eta_k^2 G^2 L^2}{\mu^2(1 - m^2\eta_k^2 L^2)}.\end{aligned}\tag{A9}$$

Now, we prove $\|x_{k+1} - x^*\|^2 \leq \epsilon_{k+1}$ by induction, where $\epsilon_{k+1} := \epsilon_k/2$, which is true for $k = 0$ as long as the initial value ϵ_0 is chosen as $\epsilon_0 := \|x_0 - x^*\|^2$. We assume $\|x_k - x^*\|^2 \leq \epsilon_k$ is true and propose to prove this inequality holds at $k+1$. Making use of the notion $d_{k,S_k} = x_{k,S_k} - x^* = x_{k+1} - x^*$ and $d_{k,0} = x_{k,0} - x^* = x_k - x^*$, following (A9) directly we have

$$\begin{aligned}&\|x_{k+1} - x^*\|^2\\ \leq&\exp(1 - \mu m\eta_k S_k)\|x_k - x^*\|^2 + \frac{m^2\eta_k^2 G^2 L^2}{\mu^2(1 - m^2\eta_k^2 L^2)}\\ \leq&\exp(1 - \mu m\eta_k S_k)\epsilon_k + \frac{m^2\eta_k^2 G^2 L^2}{\mu^2(1 - m^2\eta_k^2 L^2)}.\end{aligned}$$

Since the parameters are chosen as in Theorem 2, we obtain $\|x_{k+1} - x^*\|^2 \leq \epsilon_{k+1}$. By induction, after $T = \lceil \log_2(\epsilon_0/\epsilon) \rceil$ stages, we have $\|x_T - x^*\|^2 \leq \epsilon$, with total iterative complexity $\sum_{k=1}^{T} mS_k = \sum_{k=1}^{T} O(mL/(\mu^2\sqrt{\epsilon_k})) \leq O(mL/(\mu^2\sqrt{\epsilon}))$. □

References

1. Neter, J.; Khutner, M.H.; Nachtsheim, C.J.; Wasserman, W. *Applied Linear Statistical Models*; Irwin: Chicago, IL, USA, 1996; Volyme 4.
2. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436. [CrossRef] [PubMed]
3. Kushner, H.J.; Yin, G.G. *Stochastic Approximation and Recursive Algorithms and Applications*; Springer Science & Business Media: New York, NY, USA , 2003; Volume 35.
4. Bottou, L. Stochastic Gradient Descent Tricks. In *Neural Networks: Tricks of the Trade, Reloaded*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 421–436.
5. He, W.; Liu, Y. To regularize or not: Revisiting SGD with simple algorithms and experimental studies. *Expert Syst. Appl.* **2018**, *112*, 1–14. [CrossRef]
6. He, W.; Kwok, J.T.; Zhu, J.; Liu, Y. A Note on the Unification of Adaptive Online Learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *28*, 1178–1191. [CrossRef] [PubMed]
7. Bottou, L.; Bousquet, O. The Tradeoffs of Large Scale Learning. In Proceedings of the Neural Information Processing Systems, Vancouver, BC, Canada, 8–10 December 2008; pp. 161–168.
8. Bottou, L.; Curtis, F.E.; Nocedal, J. Optimization Methods for Large-Scale Machine Learning. *SIAM Rev.* **2018**, *60*, 223–311. [CrossRef]
9. Polyak, B.T. Some methods of speeding up the convergence of iteration methods. *USSR Comput. Math. Math. Phys.* **1964**, *4*, 1–17. [CrossRef]
10. Nesterov, Y. A method of solving a convex programming problem with convergence rate O($1/k^2$). *Sov. Math. Dokl.* **1983**, *27*, 372–376.
11. Nesterov, Y. *Introductory Lectures on Convex Optimization: A Basic Course*; Kluwer: Boston , MA, USA, 2004.
12. Nesterov, Y. Smooth minimization of non-smooth functions. *Math. Program.* **2005**, *103*, 127–152. [CrossRef]
13. Auslender, A.; Teboulle, M. Interior Gradient and Proximal Methods for Convex and Conic Optimization. *SIAM J. Optim.* **2006**, *16*, 697–725. [CrossRef]
14. Nesterov, Y. Primal-dual subgradient methods for convex problems. *Math. Program.* **2009**, *120*, 221–259. [CrossRef]
15. Lan, G.; Lu, Z.; Monteiro, R.D.C. Primal-dual first-order methods with $O(1/\epsilon)$ iteration-complexity for cone programming. *Math. Program.* **2011**, *126*, 1–29. [CrossRef]
16. Ghadimi, S.; Lan, G. Stochastic First- and Zeroth-Order Methods for Nonconvex Stochastic Programming. *SIAM J. Optim.* **2013**, *23*, 2341–2368. [CrossRef]
17. Sutskever, I.; Martens, J.; Dahl, G.E.; Hinton, G.E. On the importance of initialization and momentum in deep learning. In Proceedings of the International Conference on Machine Learning, Atlanta, GA, USA, 16–21 June 2013; pp. 1139–1147.
18. Ochs, P.; Chen, Y.; Brox, T.; Pock, T. iPiano: Inertial Proximal Algorithm for Nonconvex Optimization. *SIAM J. Imaging Sci.* **2014**, *7*, 1388–1419. [CrossRef]
19. Ghadimi, S.; Lan, G. Accelerated gradient methods for nonconvex nonlinear and stochastic programming. *Math. Program.* **2016**, *156*, 59–99. [CrossRef]
20. Johnson, R.; Zhang, T. Accelerating Stochastic Gradient Descent using Predictive Variance Reduction. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–10 December 2013; pp. 315–323.
21. Reddi, S.J.; Hefny, A.; Sra, S.; Poczos, B.; Smola, A. Stochastic Variance Reduction for Nonconvex Optimization. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; pp. 314–323.
22. Shang, F.; Zhou, K.; Liu, H.; Cheng, J.; Tsang, I.W.; Zhang, L.; Tao, D.; Jiao, L. VR-SGD: A Simple Stochastic Variance Reduction Method for Machine Learning. *IEEE Trans. Knowl. Data Eng.* **2020**, *32*, 188–202. [CrossRef]
23. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1106–1114.
24. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
25. Cui, Z.X.; Fan, Q. A "Nonconvex + Nonconvex" approach for image restoration with impulse noise removal. *Appl. Math. Model.* **2018**, *62*, 254–271. [CrossRef]
26. Fan, Q.; Jia, C.; Liu, J.; Luo, Y. Robust recovery in 1-bit compressive sensing via Lq-constrained least squares. *Signal Process.* **2021**, *179*, 107822. [CrossRef]
27. Xu, Y.; Lin, Q.; Yang, T. Stochastic Convex Optimization: Faster Local Growth Implies Faster Global Convergence. In Proceedings of the International Conference on Machine Learning, Ningbo, China, 9–12 July 2017; pp. 3821–3830.
28. Hardt, M.; Ma, T. Identity matters in deep learning. *arXiv* **2016**, arXiv:1611.04231.
29. Xie, B.; Liang, Y.; Song, L. Diversity leads to generalization in neural networks. *arXiv* **2016**, arXiv:1611.03131v2.
30. Li, Y.; Yuan, Y. Convergence analysis of two-layer neural networks with relu activation. In Proceedings of the Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 597–607.

31. Zhou, Y.; Liang, Y. Characterization of gradient dominance and regularity conditions for neural networks. *arXiv* **2017**, arXiv:1710.06910.
32. Charles, Z.; Papailiopoulos, D.S. Stability and Generalization of Learning Algorithms that Converge to Global Optima. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 744–753.
33. Arjevani, Y.; Carmon, Y.; Duchi, J.C.; Foster, D.J.; Srebro, N.; Woodworth, B. Lower bounds for non-convex stochastic optimization. *Math. Program.* **2023**, *199*, 165–214. [CrossRef]
34. Horváth, S.; Lei, L.; Richtárik, P.; Jordan, M.I. Adaptivity of stochastic gradient methods for nonconvex optimization. *SIAM J. Math. Data Sci.* **2022**, *4*, 634–648. [CrossRef]
35. Wang, Z.; Zhang, J.; Chang, T.H.; Li, J.; Luo, Z.Q. Distributed stochastic consensus optimization with momentum for nonconvex nonsmooth problems. *IEEE Trans. Signal Process.* **2021**, *69*, 4486–4501. [CrossRef]
36. Yuan, K.; Ying, B.; Zhao, X.; Sayed, A.H. Exact Diffusion for Distributed Optimization and Learning—Part I: Algorithm Development. *IEEE Trans. Signal Process.* **2019**, *67*, 708–723. [CrossRef]
37. Yuan, Z.; Yan, Y.; Jin, R.; Yang, T. Stagewise training accelerates convergence of testing error over SGD. In Proceedings of the Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; pp. 2604–2614.
38. Bolte, J.; Nguyen, T.P.; Peypouquet, J.; Suter, B.W. From error bounds to the complexity of first-order descent methods for convex functions. *Math. Program.* **2017**, *165*, 471–507. [CrossRef]
39. Karimi, H.; Nutini, J.; Schmidt, M. Linear convergence of gradient and proximal-gradient methods under the Polyak-Łojasiewicz condition. In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Riva del Garda, Italy, 19–23 September 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 795–811.
40. Allen-Zhu, Z. Katyusha: The First Direct Acceleration of Stochastic Gradient Methods. *J. Mach. Learn. Res.* **2018**, *18*, 1–51.