MOMENTUM AS VARIANCE-REDUCED STOCHASTIC GRADIENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Stochastic gradient descent with momentum (SGD+M) is widely used to empirically improve the convergence behavior and the generalization performance of plain stochastic gradient descent (SGD) in the training of deep learning models, but our theoretical understanding for SGD+M is still very limited. Contrary to the conventional wisdom that sees the momentum in SGD+M as a way to extrapolate the iterates, this work provides an alternative view that interprets the momentum in SGD+M as a (biased) variance-reduced stochastic gradient. We rigorously prove that the momentum in SGD+M converges to the real gradient, with the variance vanishing asymptotically. This reduced variance in gradient estimation thus provides better convergence behavior and opens up a different path for future analyses of momentum methods. Because the reduction of the variance in the momentum requires neither a finite-sum structure in the objective function nor complicated hyperparameters to tune, SGD+M works on complicated deep learning models with possible involvement of data augmentation and dropout, on which many other variance reduction methods fail.

1 INTRODUCTION

Stochastic gradient descent (SGD) has become one of the most popular algorithms for training machine learning models due to its low per-iteration cost and astonishing practical performance, especially when large-scale data are involved (Bottou, 2010; Shalev-Shwartz et al., 2011; Bottou et al., 2018). When applied to training deep learning models, SGD is often combined with momentum that extrapolates the previous update step to obtain better practical performance in both the generalization accuracy and the empirical convergence behavior (Sutskever et al., 2013). On the other hand, current theoretical understanding for SGD with momentum (SGD+M) is incommensurate to its popularity in practice. Although some studies have analyzed the convergence of SGD+M under various schemes, their results do not reflect why momentum is applied by default to SGD especially when training deep learning models. This leads to the open question:

For SGD+M, can we show a convergence behavior better than that of SGD to corroborate its empirical superiority?

In this work, we affirmatively answer this question in part by utilizing the recently established result of iterate convergence of SGD+M under the scenario of the stochastic heavy-ball method. The difficulty in analyzing SGD+M mainly arises from the stochastic nature of the algorithm that complicates the accumulated error in the momentum in comparison to the analyses for their deterministic counterparts. Our analysis shows that, surprisingly, this accumulated error of in the momentum actually vanishes asymptotically when the iterates converge. The cancellation of the accumulated error can therefore be used to improve the local convergence speed analogous to variance-reduction stochastic methods popular for finite-sum problems, such as SAG (Schmidt et al., 2017), SVRG (Johnson & Zhang, 2013), SAGA (Defazio et al., 2014), just to name a few. This not only explains the prominent practical performance of SGD+M, but also justifies the success of the recently popular cyclical step size scheduling that adopts larger step sizes at later stages (Smith, 2017), as reduced step sizes and the accompanied slower convergence in the original SGD was a consequence of the non-vanishing variance in the stochastic gradients.

Convergence of the iterates of SGD+M was unknown for a long period of time. Past study focused on the non-convergence to saddle points under suitable assumptions, but whether the iterates converge remained an open problem. Recently, Gadat et al. (2018) have shown that the iterates generated by the stochastic version of heavy-ball, which can be seen as a form of SGD+M, generates iterates that converge almost surely to a critical point without the need of convexity, but convergence rates are established only under the additional assumption of strong convexity. Sebbouh et al. (2021) proved that when the optimization problem is convex, stochastic heavy-ball generates iterates that converge almost surely to a solution, but contrary to the deterministic case, the obtained convergence speed of the objective value is not faster than that of SGD alone.

When the real gradient without noise is used as in the deterministic case, SGD+M reduces to the well-known heavy-ball method of Polyak (1964). When the underlying problem is strongly convex, Polyak (1964) showed that as the iterates converge to the optimal solution, the heavy-ball method exhibits an accelerated local linear convergence rate of $1 - \kappa^{-1/2}$, which is significantly faster than the $1-\kappa^{-1}$ rate of gradient descent. We notice that the accelerated speed of the heavy-ball method is asymptotic, meaning that it happens only when the iterates are close enough to the optimal solution. This result ignites our motivation to follow the traditional optimization viewpoint to study the local behavior of SGD+M, instead of the more conservative and pessimistic global complexity.

In the stochastic setting where the objective function to minimize is the expectation over a given distribution of some parameterized loss function, we observe that convergent iterates provide a local acceleration for heavy-ball, or equivalently SGD+M, over vanilla SGD with an equally prominent improvement but through a much different perspective. Unlike the deterministic heavy-ball method that relies on the problem being strongly convex locally around the limit point, our analysis does not require convexity even locally. Our main observation is that together with a mild smoothness assumption on the parameterized loss function, a converging sequence of iterates implies that the sampled stochastic gradients can also be viewed as stochastic gradients at the limit point plus a bias term converging to zero. Therefore, the momentum that accumulates the previous stochastic gradients can then be treated as an estimator for the real gradient at the limit point with improving precision. This then leads to our main contribution of showing that the variance in the momentum term vanishes asymptotically albeit the persistent presence of the noise in the individual stochastic gradient even at the point of convergence. Therefore, although popular variance reduction methods that rely on the finite-sum structure of the underlying problem is not applicable to deep learning tasks that often incorporate data augmentation and dropout (Defazio & Bottou, 2019), SGD+M can be viewed as a different means for variance reduction in this stochastic approximation scenario.[1]

In existing analyses for SGD and SGD+M, a scenario that is particularly of interest is the so-called overparameterized or interpolation case such that the individual stochastic gradient estimations all become zero at the point of convergence (Vaswani et al., 2019a;b). The obtained convergence speed of SGD and SGD+M in such a scenario can be much faster than the normal case in which the variance of the stochastic gradient is only assumed to be upper-bounded. Moreover, when the objective function is a finite sum of losses, even in the presence of variance in the stochastic gradient, variance reduction methods are able to generate estimators of the real gradient with the variance decreasing to zero as the iterates converge to a point. Through the vanishing gradient, these methods thus provide convergence rates matching that of SGD in the interpolation scenario, which is much faster than SGD's convergence speed in this normal setting. This suggests that even just locally, the reduction of the variance in the stochastic gradient estimation can lead to remarkable improvement in the convergence speed. Our analysis thus suggests that even if the variance in the individual stochastic gradient does not change, the variance in the momentum can reduce to zero and therefore provide faster local convergence behavior mimicking that observed in both the noiseless interpolation case and the variance reduction methods. This work thus serves as the initial step of a different angle to justify the astonishing practical performance of SGD+M, and we hope that it can open up a new path and inspire future research that focuses not only on the global complexity bound but also other perspectives like the local or asymptotic behaviors of SGD+M, which might also turn out to be useful for global analysis just like what we have observed in the interpolation case and variance reduction methods.

[1] There are also recent developments for variance reduction in such an online or streaming setting, but those methods either require the knowledge of the variance upper bound of plain SGD at a stationary point, or have complicated parameters to tune; see our discussion in Section 4.

The remainder of this work is organized as follows. We present the problem setting and the algorithm in Section 2. The main analysis is given in Section 3. Section 4 then review existing works relevant to our results. Finally, numerical results in Section 5 verity our analysis, and Section 6 discusses limitations and future directions of this work.

2 Preliminaries

We consider the following general optimization problem:

$$\min_{x\in\mathbb{R}^d} \quad F\left(x\right) \coloneqq \mathbb{E}_{\xi\sim\mathcal{D}}\left[f_\xi\left(x\right)\right], \tag{1}$$

where $d > 0$ is the dimension of the problem, $\mathcal{D}$ is a distribution over an arbitrary space Ω, $\mathbb{E}_{\xi\sim\mathcal{D}}$ is the expectation with respect to the random variable ξ distributed as $\mathcal{D}$, $f_\xi(x)$ is differentiable almost everywhere for all $\xi \in \Omega$. We further assume that (1) is bounded below, that the set of stationary points $\mathcal{Z} \coloneqq \{x \mid \nabla F(x) = 0\}$ is nonempty, and there exist global solutions to (1) inside $\mathcal{Z}$. In this work, $\|\cdot\|$ denotes the Euclidean norm for vectors and the corresponding operator norm for matrices, and the inner product is denoted by $\langle\cdot,\,\cdot\rangle$.

In most machine learning problems, instead of (1), the finite-sum structure such that Ω consists of finitely many points is often assumed, and an abundant amount of study for such a scenario under the name empirical risk minimization is available. We do not consider this scenario because as discussed by Defazio & Bottou (2019), in the practice of deep learning, data augmentation (Van Dyk & Meng, 2001; Kobayashi, 2018; Shorten & Khoshgoftaar, 2019) and dropout (Srivastava et al., 2014) are pervasive and the amount of possible random data points they can generate from a finite population is intractable for any utilization of the finite-sum structure.

The algorithm SGD+M being considered takes the following simple iterative form with $m^0 = 0, \alpha_0 = 1$, and some given initialization x^0:

$$\begin{cases} m^{t+1} \leftarrow \left(1-\alpha_t\right)m^t + \alpha_t \nabla f_{\xi_{t+1}}(x^t) \\ x^{t+1} = x^t - \eta_t m^{t+1} \end{cases}, \quad t \geq 0. \tag{2}$$

where ξ_t for all $t \geq 1$ are random variables that independently and identically distributed (i.i.d.) as $\mathcal{D}$, m^t is the momentum term that accumulates previous gradients, $\alpha_t \in [0,1], \eta_t \geq 0$ are algorithm-defining parameters decided in advance. The special case of $\alpha_t \equiv 1$ in (2) reduces to the ordinary SGD without momentum and its convergence is well-established.

In our analysis, we use $\{\mathcal{F}_t\}_{t\geq 0}$ to denote the natural filtration of $\{(m^t, x^t)\}_{t\geq 0}$. Namely, $\mathcal{F}_t$ records the information of x^0, $\{\alpha_i\}_{i=0}^{t-1}$, $\{\eta_i\}_{i=0}^{t-1}$, and $\{\xi_i\}_{i=1}^{t}$. The standard assumption for SGD is that it is an unbiased estimator for the real gradient

$$\mathbb{E}_{\xi\sim\mathcal{D}}\left[\nabla f_\xi(x^t) \mid \mathcal{F}_t\right] = \nabla F(x^t), \quad \forall t \geq 0,$$

which is already incorporated into our problem formulation. When there is no ambiguity, we use $\mathbb{E}$ with no subscript to denote expectation either over $\xi \sim \mathcal{D}$ or over $\xi_1,\ldots,\xi_t \overset{\text{i.i.d.}}{\sim} \mathcal{D}$. To relate the gradient change with the iterate convergence, we further make the following assumption.

Assumption 1. *For all $\xi \sim \mathcal{D}$, $f_\xi(x)$ is differentiable almost everywhere, and there exists L_ξ such that*

$$\|\nabla f_\xi(x) - \nabla f_\xi(y)\| \leq L_\xi \|x - y\|, \quad \forall x, y \in \mathbb{R}^d \tag{3}$$

almost surely. Moreover, let $L \coloneqq \sup_{\xi\sim\mathcal{D}} L_\xi$, we have $L < \infty$.

The almost surely part in our assumption above helps to tackle a broader class of problems that contain a measure zero set of nondifferentiability for each ξ. This happens often in machine learning applications such as the hinge loss in support vector machines and the ReLU activation function in neural networks.

To quantify the variance of the stochastic gradient, the following assumption is standard for SGD.

Assumption 2. *There exists $\sigma^2 \geq 0$ such that*

$$\mathbb{E}\|\nabla f_\xi\left(x^*\right)\|^2 \leq \sigma^2, \quad \forall x^* \in \mathcal{Z}. \tag{4}$$

Assumption 2 together with Assumption 1 provides a global upper bound for the variance of the gradient that grows with the distance between the iterate and $\mathcal{Z}$. If we restrict the iterates to a compact set, we can also replace such a global bound function with a constant.

When $\sigma^2 = 0$, this is the so-called interpolation case that implies $\nabla f_\xi(x^*) = 0$, and therefore $f_\xi(x^*) = \inf_x f_\xi(x)$ almost surely under the distribution of $\mathcal{D}$. When $\sigma^2 > 0$, this assumption accords with empirical observations and is the fundamental reason that SGD requires a diminishing step size to converge. On the other hand, we will show that under suitable parameter settings, even if $\sigma^2 > 0$, $\mathbb{E}\left\|m^{t+1} - \nabla F(x^t)\right\|^2 \longrightarrow 0$, and therefore a faster asymptotic convergence can be expected.

3 Analysis

We first state existing convergence results of the iterates generated by SGD+M. The main result of reduced variance is then built upon the assumption that the iterates are convergent.

Proposition 1 (Gadat et al., 2018, Theorem 2.1)**.** *Consider applying* (2) *to optimize* (1)*. If F is a $\mathcal{C}^2$ and coercive function with* $\inf_x F(x) = 0$*, there is $c_f \geq 0$ such that*

$$\sup_{x \in \mathbb{R}^d} \left\|\nabla^2 F(x)\right\| < \infty, \quad \|\nabla F(x)\| \leq c_f F(x),$$

there is $\sigma^2 > 0$ such that

$$\mathbb{E}\left[\left\|\nabla f_\xi\left(x^t\right)\right\|^2 \mid \mathcal{F}_t\right] \leq \sigma^2\left(1 + F\left(x^t\right)\right), \quad \forall t \geq 0,$$

there are $\eta > 0$ and $p \in (0, 1]$ such that α_t and η_t satisfy

$$\eta_t = t^{-p}\eta, \quad \sum_{t \geq 1} \alpha_t = \infty, \quad \sum_{t \geq 1} \alpha_t \eta_{t-1} < \infty, \quad \limsup_{t \to \infty}\ \alpha_{t+1}^{-1} - \eta_t^{-1}\eta_{t-1}\alpha_t^{-1} < 1,$$

and for any z, $\{x \mid F(x) = z\} \cap \mathcal{Z}$ is locally finite, then $\{x^t\}$ converges almost surely to a critical point $x^* \in \mathcal{Z}$.

Proposition 2 (Sebbouh et al., 2021, Theorem 13)**.** *Consider applying* (2) *to optimize* (1)*. Assume Assumption* 1 *holds and in addition for all $\xi \sim \mathcal{D}$, f_ξ is convex almost surely. If there is a decreasing positive sequence $\{\hat{\eta}_t\}$ such that*

$$\sum \hat{\eta}_t = \infty, \quad \sum \hat{\eta}_t^2 < \infty, \quad \sum_t \frac{\hat{\eta}_t}{\sum_j \hat{\eta}_j} = \infty,$$

we define

$$\lambda_0 := 0, \quad \lambda_t := \frac{\sum_{k=0}^{t-1} \hat{\eta}_k}{4\hat{\eta}_t}, \quad \hat{\alpha}_t := \frac{\hat{\eta}_t}{1 + \lambda_{t+1}}, \quad \hat{\beta}_t := \frac{\lambda_k}{1 + \lambda_{t+1}}, \quad \forall t \geq 0,$$

and the parameters setting for α_t and η_t in (2) *satisfies*

$$\eta_{-1} = 0, \quad \eta_t = \hat{\alpha}_t + \hat{\beta}_t \eta_{t-1}, \quad \forall t \geq 0, \quad \alpha_t = \frac{\hat{\alpha}_t}{\hat{\alpha}_t + \hat{\beta}_t \eta_{t-1}}, \forall t \geq 0,$$

then $\{x_t\}$ converges almost surely to some global solution $x^* \in \mathcal{Z}$.

The assumption of $\inf_x F(x) = 0$ is simply for the ease of description and can be substituted by any finite constant, and the results still hold after straightforward modifications. We also note that the constraints related to the (stochastic) gradient is implied by Assumptions 1 and 2 together with a mild assumption that the iterates stay within a bounded domain. The coerciveness of F is also a common scheme in machine learning when a certain regularization or weight decay is applied in model training.

The above results rely on specific parameters selection and do not directly imply that (2) generates convergent iterates in general. However, in the practice of deep learning, grid parameter search is conducted to select only those parameters that lead to convergent objective, and the variables are usually restricted to a bounded region (through the help of weight decay or ℓ_2-norm regularization), so even beyond the above parameter choices, it is often observed in practice that at least convergent subsequences of the iterates exist. Therefore, in our following analysis, we assume convergence of the iterates with probability one without applying the specific parameter setting required by Propositions 1 and 2.

The following is our main result that states that the momentum m^{t+1} converges to the real gradient $\nabla F(x^t)$, which verifies the variance reduction claim we made in the paper title.

Theorem 1. *Consider applying* (2) *to solve* (1) *with Assumptions* 1 *and* 2 *hold. Assume that there is $x^* \in \mathcal{Z}$ such that $x^t \longrightarrow x^*$ almost surely, then if*

$$\sum_{t \geq 0} \alpha_t = \infty, \quad \lim_{t \to \infty} \alpha_t = 0, \quad \frac{\left\|x^{t+1} - x^t\right\|}{\alpha_t} \xrightarrow{a.s.} 0, \tag{5}$$

where $\xrightarrow{a.s.}$ denotes converging almost surely, we have that

$$\mathbb{E}\left\|m^{t+1} - \nabla F\left(x^t\right)\right\|^2 \longrightarrow 0.$$

Moreover, the covariance matrix also converges to 0.

$$\mathbb{E}\left(m^{t+1} - \nabla F\left(x^t\right)\right)\left(m^{t+1} - \nabla F\left(x^t\right)\right)^\top \longrightarrow 0.$$

Proof sketch. First, Assumption 2 together with the convergence of the iterates will be used to provide an upper bound for the noise level of each individual stochastic gradient as well as a bound for the size of $\nabla F(x^t)$. The relative change in x^t and the smoothness assumption will then ensure that the change in the expected value of the stochastic gradient does not move too fast, so $m^{t+1} - \nabla F(x^t)$ can be expressed as a weighted sum of $m^t - \nabla F(x^{t-1})$, a noise term related to the change in ∇F, and a term related to the variance of the stochastic gradient. The conditions for α_t then ensure that the noise terms will be asymptotically negligible. □

The last requirement of the relatively decreasing update length in (5) seems hard to verify, as convergence of the iterates does not imply any information about the convergence speed in the iterate difference. Fortunately, in most practical schemes of SGD+M, the (periodically) exponentially decreasing step size scheduling together with the smoothness assumption can ensure the fulfillment of such a requirement.

Remark 1.

1. *The conditions in* (5) *does not require the square of α_t to be summable, which is an often-seen requirement in stochastic algorithms, thus our result allows for much slower decrease in α_t, which is closer to the practice of momentum usage. Indeed, if we assume summability of α_t^2, a stronger result of almost sure convergence of m^t can be proven. However, this does not provide much further insights in the algorithm, so we opted for weaker assumptions.*

2. *Theorem* 1 *indicates that although SGD+M reduces the variance of the gradient estimator to zero. This suggests that the momentum is a biased but in a sense consistent estimator for the real gradient. On the other hand, the stochastic gradient in SGD is an unbiased estimator but it is not consistent in the limit of time. Therefore, SGD+M outperforms SGD asymptotically because the stochastic gradient estimation from the momentum eventually becomes the real gradient and the algorithm behaves similar to gradient descent, so it can achieve convergence rates similar to what we have seen in variance reduction methods.*

3. *The function of momentum in variance reduction also explained why in the interpolation case, plain SGD without momentum is preferred and instead Nesterov's acceleration is considered by Vaswani et al. (2019a;b), as in this situation, variance reduction is automatically achieved in SGD, and the role of momentum becomes redundant.*

4. *Theorem* 1 *also provides an explanation for why cyclical step size scheduling (Smith, 2017) works. When the variance reduces to a small enough amount, the momentum becomes a high-quality approximation of the real gradient, and we can therefore afford to use a larger*

> *step size to improve the convergence speed. The followed-up decrease of the step size then serves to improve the approximation precision of the momentum by keeping the iterates moving slowly.*

The variance reduction aspect of the momentum we observed in Theorem 1 also suggests reasoning for why Nesterov's acceleration (Nesterov, 1983) is sometimes used together with SGD+M in practice, as such an acceleration technique has already been proven to provide better convergence speed for plain variance reduction methods. See, for example, (Nitanda, 2014; Allen-Zhu, 2017; Zhou et al., 2019).

4 Related Works

Understanding SGD+M is an ongoing active research topic recent years thanks to the great success of training deep learning models using SGD+M. Works that considered special settings such as quadratic or (strongly) convex problems and provided non-asymptotic convergence analyses are abundant. Representative works include Flammarion & Bach (2015); Yuan et al. (2016); Can et al. (2019); Needell et al. (2014). These works provide some insights and intuitions for why SGD+M works well, but the settings are still too restrictive as deep learning models are highly nonconvex. Non-asymptotic convergence rates of SGD+M (on the minimum of the expected norm of the stochastic gradients) are recently discussed in Yan et al. (2018); Defazio (2020); Liu et al. (2020); Mai & Johansson (2020), but the rates are only of the same order as vanilla SGD and therefore provide only limited insight on the success of SGD+M.

Another path for analyzing SGD+M considers its asymptotic convergence, which focuses on the iterates instead of the objective value or the gradient. Liu et al. (2018) used a diffusion argument to analyze the asymptotic behavior of SGD+M and argued that SGD+M is more likely than plain SGD to escape saddle points. This is further confirmed by Wang et al. (2020), who showed that under the additional assumption of correlated negative curvature, SGD+M escapes saddle points faster than SGD by a constant factor. These results help to explain the better generalization ability of the models trained by SGD+M, but is from an angle complementary to the convergence speed. Our work is inspired by the almost sure convergence of the iterates of SGD+M to a stationary point shown by Gadat et al. (2018). In addition to Proposition 1, they further showed that when the noise of the stochastic gradient satisfies an elliptic assumption, the iterates escape saddle points and converge almost surely to a local minimum, although no rate is shown in the nonconvex setting. When the problem is further assumed to be convex or even strongly convex, convergence rates are proven in Gadat et al. (2018); Sebbouh et al. (2021) but the rates are no better than that of plain SGD. We notice that the only local acceleration of SGD+M shown by Sebbouh et al. (2021) is the improvement from $O(t^{-p})$ to $o(t^{-p})$ where t is the iteration count and $p \in (0, 1/2)$ depends on the step size selection.

When the underlying problem is convex and randomness is not involved in the algorithm, such locally accelerated from big-O to small-o due to the convergence of the iterates is prevalent in first-order optimization algorithms, as evidenced by, for example, (Bertsekas, 2016; Attouch & Peypouquet, 2016; Peng et al., 2020; Lee & Wright, 2019). Moreover, examples in Attouch & Peypouquet (2016); Lee & Wright (2019) have shown that further improvement is impossible without additional assumptions on the problem class, so such an incremental improvement in SGD+M seems reasonable. On the other hand, as mentioned in Section 1, the heavy-ball method (Polyak, 1964) has significant convergence speed improvement locally when the iterates get close enough to the point of convergence in comparison to gradient descent, and this is in stark contrast to the stochastic case in which the known rates of SGD and SGD+M are equivalent and the amount of local acceleration is negligible. This work partially fills this gap to demonstrate that the local convergence improvement of SGD+M over vanilla SGD can be much greater than expected, although it can be hard to quantify.

Variance reduction methods that decrease the variance of the stochastic gradient to zero as the iterates converge to a stationary point have been an extremely powerful tool to accelerate SGD. The most widely-considered algorithms in this category, including SAG (Schmidt et al., 2017), SVRG (Johnson & Zhang, 2013), and SAGA (Defazio et al., 2014), require the objective function to possess a finite-sum structure so that stale versions of the (stochastic) gradients can be incorporated in the algorithm design. A major improvement of these methods from SGD is that with the vanishing variance, a decreasing step size is unnecessary, and a fixed step size can thus help to achieve significantly faster convergence rates that track the speed of gradient descent. Similar speed improvement of SGD is also observed in the interpolation scheme where the variance of the stochastic gradient vanishes when approaching a stationary point (Vaswani et al., 2019a;b; Sebbouh et al., 2021). Furthermore, for both variance reduction methods and SGD in the interpolation case, Nesterov's acceleration is shown to be effective in further improving the convergence speed (Vaswani et al., 2019a; Nitanda, 2014; Allen-Zhu, 2017; Zhou et al., 2019) by another order, but this technique is not useful for plain SGD in the bounded-variance scenario. These results suggest that vanishing variance is indeed the key to accelerating stochastic algorithms, and our main result in Theorem 1 shows that the momentum term has a similar effect in reducing the variance without the need of the finite-sum structure in the objective function or additional gradient evaluations. This provides a different view point for explaining the improved practical performance of SGD+M over SGD. Another implication of our result is the justification of the recently popular cyclical step size setting proposed by Smith (2017). This scheme periodically increases the step size to a certain value and then gradually reduces it. Our result in Theorem 1 indicates that the momentum term eventually gets close to the real gradient, and therefore larger step sizes can be applied to improve the convergence. However, a large and fixed step size violates the assumptions in Theorem 1, and therefore one needs to subsequently gradually decrease the step size again to improve the quality of approximation to the real gradient, which accords with the setting in cyclical step size scheduling.

Another line of research devises variance reduction algorithms beyond the finite-sum setting in order to improve convergence speed for (1), mostly with the motivation being the online or streaming scenario instead of the time-consuming training of deep learning models. The earliest work is probably the regularized dual averaging algorithm (Xiao, 2010), although its role as a variance-reduction algorithm is shown much later by Lee & Wright (2012) and it has no known convergence guarantee in the nonconvex case. Other works including Allen-Zhu (2018); Fang et al. (2018); Wang et al. (2019); Nguyen et al. (2021); Pham et al. (2020) apply multiple loops with periodical checkpoints for computing a stochastic gradient with a relatively large batch size to attain variance reduction. These algorithms either require knowledge of the hard-to-estimate σ^2 in Assumption 2 and the Lipschitz constant, or otherwise have multiple hyperparameters to tune. In average, these algorithms also require a computational cost at least thrice of plain SGD, making hyperparameter tuning even more daunting. Therefore, these algorithms are mainly of theoretical interests without much popularity in practice. The most relevant works are Cutkosky & Orabona (2019); Tran-Dinh et al. (2019) that do not require multiple loops and their algorithms mainly just replace the momentum update in (2) with

$$m^{t+1} = (1-\alpha_t)\, m^t + \alpha_t \nabla f_{\xi_t}\left(x^t\right) + (1-\alpha_t)\left(\nabla f_{\xi_t}\left(x^t\right) - \nabla f_{\xi_t}\left(x^{t-1}\right)\right) \tag{6}$$

and provide specific choices for α_t and η_t. Cutkosky & Orabona (2019) showed that this m^t achieves variance reduction, and with properly chosen α_t and η_t together with some additional assumptions, a global convergence rate matching the optimal speed of other variance reduction methods can be obtained, although in practice there are still three hyperparameters to tune. As argued by the authors, when $\{x^t\}$ converges, the last term in (6) becomes very small, and thus their new update becomes almost the same as the ordinary momentum update in SGD+M. Our result and their argument can be seen to complement each other in confirming the function in variance reduction of the momentum term. Tran-Dinh et al. (2019) requires an initial large batch to start the first iterate from a place close enough to a stationary point to obtain the same optimal convergence rate as Cutkosky & Orabona (2019), and the remaining algorithm fits in (6). However, their hyperparameters need to change with the pre-specified epoch number, meaning that early stopping and other practices cannot be combined with the algorithm easily.

5 Experiments

In this section, we provide preliminary numerical experiments to exemplify our theoretical results that the variance of the momentum term vanishes asymptotically. Although in the analysis we considered the general case in which the objective is the expectation over a possibly infinite sample size so that data augmentation and dropout are possible, to make the computation of the variance tractable, we have to restrict our setting to finite data points in the experiments. Therefore, the problem in (1) simplifies to

$$\min_{x \in \mathbb{R}^d} \quad F(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x) \tag{7}$$

for some individual functions $f_i, i = 1, \dots, n$, that depends on the data points, and obtaining a stochastic gradient amounts to deciding an index set $\mathcal{I} \subset \{1, \dots, n\}$ and compute

$$\nabla f_{\xi_{t+1}}\left(x^t\right) = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \nabla f_i\left(x^t\right).$$

In modern deep learning platforms, the setting is further simplified such that the index set at an iteration of the t-th epoch is selected uniformly and randomly from a pre-defined partition $\{\mathcal{I}_t^j\}_{j=1}^N$ of $\{1, \dots, n\}$ such that

$$\bigcup_{j=1}^N \mathcal{I}_t^j = \{1, \dots, n\}, \quad \mathcal{I}_t^{j_1} \cap \mathcal{I}_t^{j_2} = \emptyset, \quad \forall j_1 \neq j_2,$$

where the partition can change with the epochs if reshuffling is conducted. We also note that the computational and spatial cost of the covariance matrix of either the stochastic gradient or the momentum is quadratic to the number of variables in the optimization problem, which is prohibitively large in modern machine learning tasks. Therefore, we consider the variance in the norm of the difference between the stochastic estimation and the real gradient only. This variance in the norm at the t-th epoch given the iterate x is thus calculated by either of the following.

$$\sigma_G^2 := \frac{1}{N} \sum_{j=1}^N \left\| \frac{1}{|\mathcal{I}_t^j|} \sum_{i \in \mathcal{I}_t^j} \nabla f_i(x) - \nabla F(x) \right\|^2,$$

$$\sigma_M^2 := \frac{1}{N} \sum_{j=1}^N \left\| (1 - \alpha_t) m^t + \alpha_t \frac{1}{|\mathcal{I}_t^j|} \sum_{i \in \mathcal{I}_t^j} \nabla f_i(x) - \nabla F(x) \right\|^2.$$

The second equation is for computing the variance in the momentum term m^{t+1}, while the first one is for the variance of the stochastic gradient.

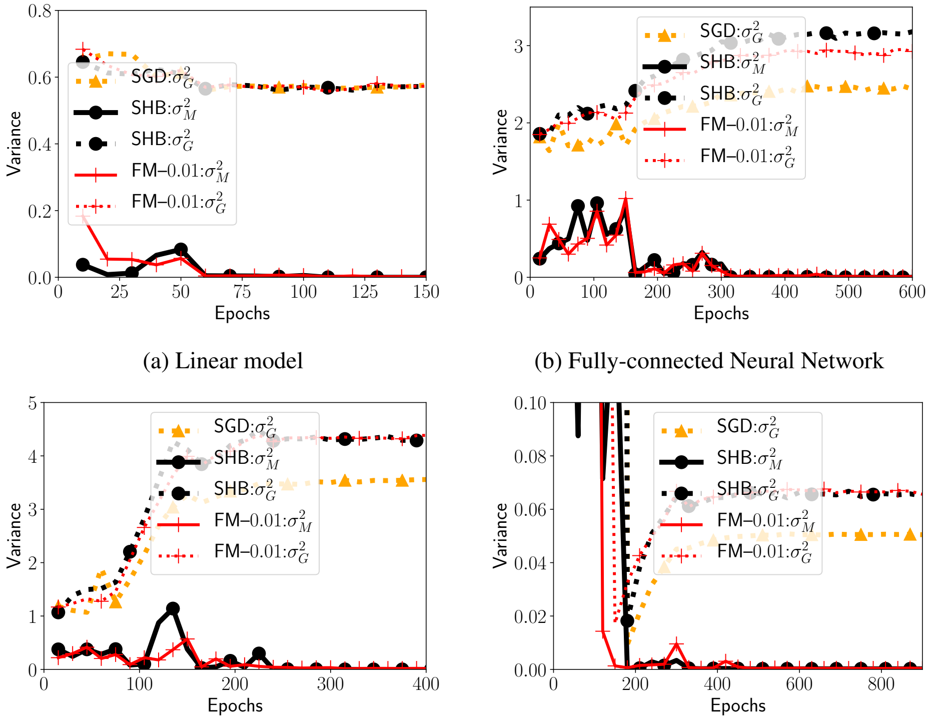
As our purpose is not to propose a new algorithm nor to showcase the performance of a certain algorithm, we do not spend much effort on tuning hyperparameters of the algorithm or the model to achieve the best performance. Instead, we use rather simple settings in the algorithm and consider smaller neural network models that allow for traversing all data points to compute the in-sample variance in an acceptable amount of time. Moreover, to avoid the interpolation case in which σ_G^2 is guaranteed to decrease to zero, so that we can still simulate the situation in the presence of data augmentation and dropout, we have to let the number of parameters in the model be smaller than the number of data points. This requirement is another reason for us to select smaller neural networks. In particular, we consider a simple linear logistic regression model, a fully-connected neural network with one hidden layer, and a convolutional neural network (CNN) model (LeNet-5 LeCun et al., 1998). We also experiment with a modern CNN network (VGG-11 Simonyan & Zisserman, 2015) and run it with fewer epochs before interpolation kicks in, so that we can observe how the variance changes in a neural network closer to what are empirically used. For the first model, we use the MNIST dataset (LeCun et al., 1998). For the second one, we use the CIFAR-10 data (Krizhevsky, 2009) with horizontal flips to double the data size to train the model. When training LeNet-5, we experiment with the FashionMNIST dataset (Xiao et al., 2017) and apply both horizontal and vertical flips to it to obtain a dataset of $240,000$ instances so that we can avoid interpolation. For VGG-11, we use the original CIFAR-10 without any data augmentation.

In the experiments, we compare plain SGD and the following schemes of SGD+M.

- Stochastic heavy-ball (SHB): Setting $\alpha_t = \eta t^{-(1/2+\epsilon)}$ for some given $\eta, \epsilon > 0$ as suggested by Sebbouh et al. (2021, Corollary 17). We fix $\eta = 1$ and $\epsilon = 10^{-2}$.
- Fixed momentum (FM): We follow the default setting of SGD in PyTorch to use a fixed value $\alpha_t \equiv \alpha$. In particular, we use $\alpha = 10^{-2}$.

Notice that the first scheme satisfies the requirements of Theorem 1 but the second one does not, but we will see in the experiments that in practice this still works well in reducing the momentum. For all algorithms, we follow common practice to apply the same multi-step scheduling that exponentially decreases the step size periodically.

The results are shown in Fig. 1. We can clearly see that for all methods, σ_G^2 remains at a constant level even at the later stage where the training objective has converged and the step size has decreased to nearly zero. This means that as expected, even if the iterates barely move, the variance of the stochastic gradient does not vanish. On the other hand, we see that for the SHB scheme satisfying the requirements of Theorem 1, σ_M^2 indeed soon decreases to very close to zero, confirming the theory. Moreover, even for the FM scheme not supported by our theory, we see that the variance of its momentum also vanishes. This shows that this popular practical setting for momentum indeed exhibits a similar effect of variance reduction. It thus explains from another angle both the practical performance of SGD+M over plain SGD and the effectiveness of cyclical step sizes.



(a) Linear model

(b) Fully-connected Neural Network

(c) LeNet-5

(d) VGG-11

Figure 1: Change of σ_G^2 and σ_M^2 with epochs on different problems.

6 Discussions

In this work, we have shown that when the iterates are convergent, SGD+M provides a means for variance reduction in the stochastic gradient estimation, and the variance vanishes asymptotically. This explains the practical superiority of SGD+M over vanilla SGD and the popularity of cyclical step sizes in the training of deep learning models. The major limitation of the proposed work is that the reduction speed in the variance cannot be directly quantified as it depends on the convergence speed of the iterates, and thus additional assumptions might be needed to obtain an explicit rate. However, we hope that this work serves as the first step towards a different but equally important perspective for analyzing and understanding SGD+M beyond the pessimistic worst-case guarantee and global complexity. In the near future, we plan to further extend the results to show faster local convergence rates of SGD+M and propose principled parameters scheduling by incorporating techniques used in analyzing the interpolation case and the variance reduction methods, which also reduce the variance of the gradient estimation only locally. This work is also far from being comprehensive in terms of the algorithms covered. Instead of considering the more sophisticated adaptive algorithms like Adagrad and Adam, we focused on the simpler case of SGD+M to illustrate our main observation, and it will also be an interesting direction to extend our analysis to such adaptive algorithms.

Reproducibility Statement

The assumptions and the theoretical result are stated in Section 3, with the proofs in Appendix A. For the experiments, our source code will be made available in the supplementary materials and the detailed experiment setting for data preprocessing is described in Appendix B.

References

Zeyuan Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. *Journal of Machine Learning Research*, 18(1):8194–8244, 2017.

Zeyuan Allen-Zhu. Natasha 2: Faster non-convex optimization than SGD. In *Advances in Neural Information Processing Systems*, volume 31, pp. 2675–2686, 2018.

Hedy Attouch and Juan Peypouquet. The rate of convergence of nesterov's accelerated forward-backward method is actually faster than $1/k^2$. *SIAM Journal on Optimization*, 26(3):1824–1834, 2016.

Dimitri P. Bertsekas. *Nonlinear programming*. Athena scientific Belmont, third edition, 2016.

Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *COMPSTAT'2010*, pp. 177–186. Springer, 2010.

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

Bugra Can, Mert Gurbuzbalaban, and Lingjiong Zhu. Accelerated linear convergence of stochastic momentum methods in Wasserstein distances. In *International Conference on Machine Learning*, pp. 891–901, 2019.

Ashok Cutkosky and Francesco Orabona. Momentum-based variance reduction in non-convex SGD. In *Advances in Neural Information Processing Systems*, volume 32, pp. 15236–15245, 2019.

Aaron Defazio. Momentum via primal averaging: Theoretical insights and learning rate schedules for non-convex optimization. Technical report, 2020. arXiv:2010.00406.

Aaron Defazio and Leon Bottou. On the ineffectiveness of variance reduced optimization for deep learning. In *Advances in Neural Information Processing Systems*, volume 32, pp. 1755–1765, 2019.

Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in neural information processing systems*, volume 27, pp. 1646–1654, 2014.

Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. SPIDER: near-optimal non-convex optimization via stochastic path integrated differential estimator. In *Advances in Neural Information Processing Systems*, volume 31, pp. 687–697, 2018.

Nicolas Flammarion and Francis Bach. From averaging to acceleration, there is only a step-size. In *Conference on Learning Theory*, pp. 658–695, 2015.

Sébastien Gadat, Fabien Panloup, and Sofiane Saadane. Stochastic heavy ball. *Electronic Journal of Statistics*, 12(1):461–529, 2018.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, volume 26, pp. 315–323, 2013.

Sosuke Kobayashi. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 452–457, 2018.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Ching-pei Lee and Stephen J. Wright. First-order algorithms converge faster than $O(1/k)$ on convex problems. In *International Conference on Machine Learning*, 2019.

Sangkyun Lee and Stephen J. Wright. Manifold identification in dual averaging for regularized stochastic online learning. *Journal of Machine Learning Research*, 13:1705–1744, 2012.

Tianyi Liu, Zhehui Chen, Enlu Zhou, and Tuo Zhao. Towards deeper understanding of nonconvex stochastic optimization with momentum using diffusion approximations. Technical report, 2018. arXiv:1802.05155.

Yanli Liu, Yuan Gao, and Wotao Yin. An improved analysis of stochastic gradient descent with momentum. Technical report, 2020. arXiv:2007.07989.

Vien Mai and Mikael Johansson. Convergence of a stochastic gradient method with momentum for non-smooth non-convex optimization. In *International Conference on Machine Learning*, pp. 6630–6639, 2020.

Deanna Needell, Rachel Ward, and Nathan Srebro. Stochastic gradient descent, weighted sampling, and the randomized kaczmarz algorithm. In *Advances in neural information processing systems*, volume 27, pp. 1017–1025, 2014.

Yurii Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady*, 27:372–376, 1983.

Lam M. Nguyen, Katya Scheinberg, and Martin Takáč. Inexact SARAH algorithm for stochastic optimization. *Optimization Methods and Software*, 36(1):237–258, 2021.

Atsushi Nitanda. Stochastic proximal gradient descent with acceleration techniques. In *Advances in Neural Information Processing Systems*, volume 27, pp. 1574–1582, 2014.

Wei Peng, Hui Zhang, Xiaoya Zhang, and Lizhi Cheng. Global complexity analysis of inexact successive quadratic approximation methods for regularized optimization under mild assumptions. *Journal of Global Optimization*, 78(1):69–89, 2020.

Nhan H. Pham, Lam M. Nguyen, Dzung T. Phan, and Quoc Tran-Dinh. ProxSARAH: An efficient algorithmic framework for stochastic composite nonconvex optimization. *Journal of Machine Learning Research*, 21:1–48, 2020.

Boris T. Polyak. Some methods of speeding up the convergence of iteration methods. *Ussr computational mathematics and mathematical physics*, 4(5):1–17, 1964.

Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1-2):83–112, 2017.

Othmane Sebbouh, Robert M. Gower, and Aaron Defazio. Almost sure convergence rates for stochastic gradient descent and stochastic heavy ball. In *Conference on Learning Theory*, pp. 3935–3971. PMLR, 2021.

Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro, and Andrew Cotter. Pegasos: Primal estimated sub-gradient solver for SVM. *Mathematical programming*, 127(1):3–30, 2011.

Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):1–48, 2019.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. 2015.

Leslie N. Smith. Cyclical learning rates for training neural networks. In *IEEE winter conference on applications of computer vision*, pp. 464–472. IEEE, 2017.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pp. 1139–1147. PMLR, 2013.

Quoc Tran-Dinh, Nhan H. Pham, Dzung T. Phan, and Lam M. Nguyen. Hybrid stochastic gradient descent algorithms for stochastic nonconvex optimization. Technical report, 2019. arXiv:1905.05920.

David A. Van Dyk and Xiao-Li Meng. The art of data augmentation. *Journal of Computational and Graphical Statistics*, 10(1):1–50, 2001.

Sharan Vaswani, Francis Bach, and Mark Schmidt. Fast and faster convergence of SGD for over-parameterized models and an accelerated perceptron. In *International Conference on Artificial Intelligence and Statistics*, pp. 1195–1204. PMLR, 2019a.

Sharan Vaswani, Aaron Mishkin, Issam Laradji, Mark Schmidt, Gauthier Gidel, and Simon Lacoste-Julien. Painless stochastic gradient: Interpolation, line-search, and convergence rates. In *Advances in neural information processing systems*, volume 32, pp. 3732–3745, 2019b.

Jun-Kun Wang, Chi-Heng Lin, and Jacob Abernethy. Escaping saddle points faster with stochastic momentum. In *International Conference on Learning Representations*, 2020.

Zhe Wang, Kaiyi Ji, Yi Zhou, Yingbin Liang, and Vahid Tarokh. Spiderboost and momentum: Faster variance reduction algorithms. In *Advances in Neural Information Processing Systems*, volume 32, pp. 2406–2416, 2019.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. Technical report, 2017. arXiv:1708.07747.

Lin Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11(88):2543–2596, 2010.

Yan Yan, Tianbao Yang, Zhe Li, Qihang Lin, and Yi Yang. A unified analysis of stochastic momentum methods for deep learning. In *International Joint Conference on Artificial Intelligence*, 2018.

Kun Yuan, Bicheng Ying, and Ali H. Sayed. On the influence of momentum acceleration on online learning. *Journal of Machine Learning Research*, 17(1):6602–6667, 2016.

Kaiwen Zhou, Qinghua Ding, Fanhua Shang, James Cheng, Danli Li, and Zhi-Quan Luo. Direct acceleration of saga using sampled negative momentum. In *International Conference on Artificial Intelligence and Statistics*, pp. 1602–1610, 2019.

A Proofs

In this section, we provide proofs for the theoretical results in Section 3 for completeness.

We first provide a lemma showing that the variance of $\nabla f_\xi(x^t)$ is bounded.

Lemma 1. *Under Assumptions* 1 *and* 2*, if $x^t \xrightarrow{a.s.} x^*$, there exits $C > 0$ such that*

$$\mathbb{E}\left\|\nabla f_{\xi_{t+1}}(x^t) - \nabla F(x^t)\right\|^2 \le C, \quad \forall t \ge 0. \tag{8}$$

Proof. From the convergence of x^t to x^* we know that $\{x^t\}$ is bounded. Therefore, we can find a compact set S such that $\{x^t\} \subset S$ almost surely, with

$$\sup_{x \in S} \|x\| =: R_S < \infty. \tag{9}$$

From (3) and (9), we then get

$$\left\|\nabla F(x^t) - \nabla F(x^*)\right\| \le L\left\|x^t - x^*\right\| \le 2LR_S, \quad \forall t \ge 0 \tag{10}$$

almost surely. Moreover, $\nabla F(x^*) = 0$ as $x^* \in \mathcal{Z}$, so the bound above suggests that $\|\nabla F(x^t)\| \le 2LR_S$ for all t almost surely.

On the other hand, from the Cauchy-Schwarz inequality, Assumptions 1 and 2 and (9), we get

$$\begin{aligned}
&\mathbb{E}\left\|\nabla f_{\xi_{t+1}}\left(x^t\right)\right\|^2 \\
=\ &\mathbb{E}\left\|\nabla f_{\xi_{t+1}}\left(x^t\right) - \nabla f_{\xi_{t+1}}\left(x^*\right) + \nabla f_{\xi_{t+1}}\left(x^*\right)\right\|^2 \\
\le\ &2\left(\mathbb{E}\left\|\nabla f_{\xi_{t+1}}\left(x^t\right) - \nabla f_{\xi_{t+1}}\left(x^*\right)\right\|^2 + \mathbb{E}\left\|\nabla f_{\xi_{t+1}}\left(x^*\right)\right\|^2\right) \\
\le\ &2\left(\left(2LR_S\right)^2 + \sigma^2\right).
\end{aligned} \tag{11}$$

We can then bound the distance between $\nabla f_{\xi_{t+1}}(x^t)$ and $\nabla F(x^t)$ again by the Cauchy-Schwarz inequality:

$$\left\|\nabla f_{\xi_{t+1}}(x^t) - \nabla F(x^t)\right\|^2 \le 2\left(\left\|\nabla f_{\xi_{t+1}}(x^t)\right\|^2 + \left\|\nabla F(x^t)\right\|^2\right).$$

By further taking expectation on both sides and using (10) and (11), we obtain

$$\mathbb{E}\left\|\nabla f_{\xi_{t+1}}(x^t) - \nabla F(x^t)\right\|^2 \le 2\left(\left(2LR_S\right)^2 + 2\sigma^2 + \left(2LR_S\right)^2\right) < \infty,$$

proving (8) with $C = 2\left(\left(2LR_S\right)^2 + 2\sigma^2 + \left(2LR_S\right)^2\right)$. □

Now we are ready to prove Theorem 1.

Proof of Theorem 1. Consider $\left\|m^{t+1} - \nabla F(x^t)\right\|^2$, we have from (2) that

$$\begin{aligned}
&\left\|m^{t+1} - \nabla F(x^t)\right\|^2 \\
&= \left\|(1-\alpha_t)m^t + \alpha_t \nabla f_{\xi_{t+1}}(x^t) - \nabla F(x^t)\right\|^2 \\
&= \left\|(1-\alpha_t)\left(m^t - \nabla F(x^t)\right) + \alpha_t\left(\nabla f_{\xi_{t+1}}(x^t) - \nabla F(x^t)\right)\right\|^2 \\
&= (1-\alpha_t)^2\left\|m^t - \nabla F(x^t)\right\|^2 + \alpha_t^2\left\|\nabla f_{\xi_{t+1}}(x^t) - \nabla F(x^t)\right\|^2 \\
&\qquad + 2\alpha_t(1-\alpha_t)\langle m^t - \nabla F(x^t), \nabla f_{\xi_{t+1}}(x^t) - \nabla F(x^t)\rangle \\
&= (1-\alpha_t)^2\left\|\left(m^t - \nabla F\left(x^{t-1}\right)\right) + \left(\nabla F\left(x^{t-1}\right) - \nabla F\left(x^t\right)\right)\right\|^2 + \alpha_t^2\left\|\nabla f_{\xi_{t+1}}(x^t) - \nabla F(x^t)\right\|^2 \\
&\qquad + 2\alpha_t(1-\alpha_t)\langle m^t - \nabla F(x^t), \nabla f_{\xi_{t+1}}(x^t) - \nabla F(x^t)\rangle.
\end{aligned} \tag{12}$$

By defining $V_t := \left\|m^t - \nabla F(x^{t-1})\right\|^2$ and taking expectation over (12) conditional on $\mathcal{F}_t$, we obtain from $\mathbb{E}\left[\nabla f_{\xi_{t+1}}(x^t) \mid \mathcal{F}_t\right] = \nabla F(x^t)$ that

$$\begin{aligned}
&\mathbb{E}\left[V_{t+1} \mid \mathcal{F}_t\right] \\
&= (1-\alpha_t)^2 \left\|\left(m^t - \nabla F\left(x^{t-1}\right)\right) + \left(\nabla F\left(x^{t-1}\right) - \nabla F\left(x^t\right)\right)\right\|^2 \\
&\qquad + \alpha_t^2 \mathbb{E}\left[\left\|\nabla f_{\xi_t}(x^t) - \nabla F(x^t)\right\|^2 \mid \mathcal{F}_t\right]. \qquad (13)
\end{aligned}$$

From the assumed conditions in (5) and the Lipschitz continuity of ∇F, there are random variables $\{\epsilon_t\}$ and $\{u_t\}$ such that $\|u_t\| = 1$, $\epsilon_t \geq 0$, and $\nabla F(x^{t-1}) - \nabla F(x^t) = \alpha_t \epsilon_t u_t$ for all $t > 0$, with $\epsilon_t \downarrow 0$ almost surely. We thus obtain that

$$\begin{aligned}
&\left\|m^t - \nabla F(x^{t-1}) + \nabla F(x^{t-1}) - \nabla F(x^t)\right\|^2 \\
&= \left\|m^t - \nabla F(x^{t-1}) + \alpha_t \epsilon_t u_t\right\|^2 \\
&= (1+\alpha_t)^2 \left\|\frac{1}{1+\alpha_t}\left(m^t - \nabla F\left(x^{t-1}\right)\right) + \frac{\alpha_t}{1+\alpha_t}\epsilon_t u_t\right\|^2 \\
&\leq (1+\alpha_t)^2 \left(\frac{1}{1+\alpha_t} V_t + \frac{\alpha_t}{1+\alpha_t}{\epsilon_t}^2\right), \qquad (14)
\end{aligned}$$

where we used Jensen's inequality and the convexity of $\|\cdot\|^2$ in the last inequality. By substituting (14) back into (13), we obtain

$$\begin{aligned}
&\mathbb{E}\left[V_{t+1} | \mathcal{F}_t\right] \\
&\leq (1-\alpha_t)^2 (1+\alpha_t) V_t + (1-\alpha_t)^2(1+\alpha_t)\alpha_t {\epsilon_t}^2 + {\alpha_t}^2 \mathbb{E}\left[\left\|\nabla f_{\xi_t}(x^t) - \nabla F(x^t)\right\|^2 \mid \mathcal{F}_t\right] \\
&\leq (1-\alpha_t)(V_t + \alpha_t {\epsilon_t}^2) + {\alpha_t}^2 \mathbb{E}\left[\left\|\nabla f_{\xi_t}(x^t) - \nabla F(x^t)\right\|^2 \mid \mathcal{F}_t\right] \\
&\leq (1-\alpha_t) V_t + \alpha_t {\epsilon_t}^2 + {\alpha_t}^2 \mathbb{E}\left[\left\|\nabla f_{\xi_t}(x^t) - \nabla F(x^t)\right\|^2 \mid \mathcal{F}_t\right]. \qquad (15)
\end{aligned}$$

Now we further take expectation on (15) and apply Lemma 1 to obtain

$$\mathbb{E}V_{t+1} \leq (1-\alpha_t)\mathbb{E}V_t + \alpha_t {\epsilon_t}^2 + {\alpha_t}^2 C = (1-\alpha_t)\,\mathbb{E}V_t + \alpha_t\left(\epsilon_t^2 + \alpha_t C\right). \qquad (16)$$

Note that the third implies $\epsilon_t \downarrow 0$, so this together with the second condition that $\alpha_t \downarrow 0$ means $\epsilon_t^2 + \alpha_t C \downarrow 0$ as well, and thus for any $\delta > 0$, we can find $T_\delta \geq 0$ such that $\epsilon_t^2 + \alpha_t C \leq \delta$ for all $t \geq T_\delta$. Thus, (16) further leads to

$$\mathbb{E}V_{t+1} - \delta \leq (1-\alpha_t)\mathbb{E}V_t + \alpha_t\delta - \delta = (1-\alpha_t)\left(\mathbb{E}V_t - \delta\right), \forall t \geq T_\delta. \qquad (17)$$

This implies that $(\mathbb{E}V_t - \delta)$ becomes a decreasing sequence starting from $t \geq T_\delta$, and since $V_t \geq 0$, this sequence is lower bounded by $-\delta$, and hence it converges to a certain value. By recursion of (17), we have that

$$\mathbb{E}V_t - \delta \leq \prod_{i=T_\delta}^{t}(1-\alpha_i)\left(\mathbb{E}V_{T_\delta} - \delta\right),$$

and from the well-known inequality $(1+x) \leq \exp^x$ for all $x \in \mathcal{R}$, the above result leads to

$$\mathbb{E}V_t - \delta \leq \exp\left(-\sum i = {T_\delta}^t \alpha_i\right)\left(\mathbb{E}V_{T_\delta} - \delta\right).$$

By letting t approach infinity and noting that (5) indicates

$$\sum_{t=k}^{\infty} \alpha_t = \infty$$

for any $k \geq 0$, we see that

$$-\delta \leq \lim_{t\to\infty} \mathbb{E}V_t - \delta \leq \exp\left(-\sum_{i=T_\delta}^{\infty}\alpha_i\right)\left(\mathbb{E}V_{T_\delta} - \delta\right) = 0. \qquad (18)$$

Data set	MNIST
Data augmentation	None
Data points	$60,000$
Model	Linear logistic regression
Number of parameters	$7,840$
Weight decay	0
Loss function	Cross entropy
Batch size	128
Step size	$10^{-(1+\lfloor \text{epoch}/50 \rfloor)}$
total epochs	150
Final training accuracy	93.9%
Final validation accuracy	SGD & SHB: 92.6%; FM: 92.7%

Table 1: Details of the experimental setting of training logistic regression on MNIST.

As δ is arbitrary, by taking $\delta \downarrow 0$ in (18) and noting the nonnegativity of V_t, we conclude that $\lim \mathbb{E} V_t = 0$, as desired.

Finally, we note that

$$\mathbb{E}\left\|m^{t+1} - \nabla F\left(x^t\right)\right\|^2 = \text{trace}\left(\Sigma^t\right), \quad \Sigma^t := \mathbb{E}\left(m^{t+1} - \nabla F\left(x^t\right)\right)\left(m^{t+1} - \nabla F\left(x^t\right)\right)^\top,$$

and we also know that the covariance matrix Σ^t is always positive semidefinite and symmetric, which means that all principal minors should be nonnegative. By taking any i, j that satisfy $1 \leq i, j \leq d, i \neq j$, we have from the above argument that

$$\Sigma^t_{i,i} \geq 0, \quad \Sigma^t_{j,j} \geq 0, \quad \Sigma^t_{i,i}\Sigma^t_{j,j} \geq \left(\Sigma^t_{i,j}\right)^2. \tag{19}$$

Moreover, since

$$\sum_{i=1}^{d} \Sigma^t_{i,i} = \text{trace}\left(\Sigma^t\right) \longrightarrow 0,$$

we have from the nonnegativity of $\Sigma^t_{i,i}$ that

$$\Sigma^t_{i,i} \longrightarrow 0, \quad i = 1, \ldots, d.$$

Therefore, from (19) and the bound above, we conclude that $\Sigma^t_{i,j} \longrightarrow 0$ for all i, j. This proves the last result in Theorem 1. □

B Experiment Details

In this section, we provide the detailed model, data, and algorithm settings. The setting for the linear model, the multi-layer fully-connected neural network, LeNet-5, and VGG-11 are shown respectively in Tables 1 to 4.

Data set	CIFAR-10
Data augmentation	Horizontal flip
Data points	$100,000$
Model	Fully-connected neural network
Number of hidden layers	1
Number of neurons in the hidden layers	10
Number of parameters	$30,820$
Weight decay	10^{-5}
Activation	ReLU
Loss function	Cross entropy
Batch size	256
Step size	$10^{-(1+\lfloor \text{epoch}/150 \rfloor)}$
total epochs	600
Final training accuracy	SHB: 49.6%; FM: 49.5%; SGD: 49.1%
Final validation accuracy	SHB: 42.5%; FM: 42.4%; SGD: 42.4%

Table 2: Details of the experimental setting of training the multi-layer fully-connected neural network on CIFAR-10.

Data set	FashionMNIST
Data augmentation	Horizontal flip, vertical flip, horizontal & vertical flip
Data points	$240,000$
Model	LeNet-5
Number of parameters	$60,850$
Weight decay	10^{-3}
Activation	ReLU
Loss function	Cross entropy
Batch size	256
Step size	$10^{-(1+\lfloor \text{epoch}/75 \rfloor)}$
total epochs	400
Final training accuracy	SHB: 98.0%; FM: 97.7%; SGD: 97.8%
Final validation accuracy	SHB: 91.3%; FM: 91.5%; SGD: 91.2%

Table 3: Details of the experimental setting of training LeNet-5 on augmented FashionMNIST.

Data set	CIFAR-10
Data points	$50,000$
Model	VGG-11
Number of parameters	133 millions
Weight decay	10^{-3}
Activation	ReLU
Loss function	Cross entropy
Batch size	128
Step size	$10^{-(1+\lfloor \text{epoch}/150 \rfloor)}$
total epochs	900
Final training accuracy	100%
Final validation accuracy	SHB: 78.6%; FM: 78.1%; SGD: 77.8%

Table 4: Details of the experimental setting of training VGG-11 on CIFAR-10.