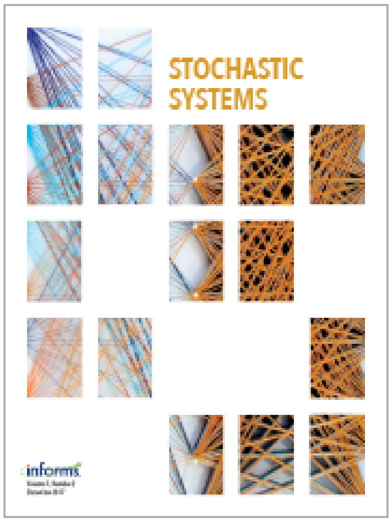


Stochastic Systems

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

A Diffusion Approximation Theory of Momentum Stochastic Gradient Descent in Nonconvex Optimization

Tianyi Liu, Zhehui Chen, Enlu Zhou, Tuo Zhao

informs
http://pubsonline.informs.org/journal/stsy

STOCHASTIC SYSTEMS
Vol. 11, No. 4, December 2021, pp. 265–281
ISSN 1946-5238 (online)

A Diffusion Approximation Theory of Momentum Stochastic Gradient Descent in Nonconvex Optimization

Tianyi Liu,[a] Zhehui Chen,[a] Enlu Zhou,[a] Tuo Zhao[a]

[a] School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia 30318

Contact: tianyiliu@gatech.edu, https://orcid.org/0000-0002-5573-5093 (TL); zchen451@gatech.edu (ZC); enlu.zhou@isye.gatech.edu, https://orcid.org/0000-0001-5399-6508 (EZ); tourzhao@gatech.edu (TZ)

Received: October 9, 2019
Revised: March 2, 2021
Accepted: June 10, 2021
Published Online in Articles in Advance: October 21, 2021

https://doi.org/10.1287/stsy.2021.0083

Copyright: © 2021 The Author(s)

Abstract. Momentum stochastic gradient descent (MSGD) algorithm has been widely applied to many nonconvex optimization problems in machine learning (e.g., training deep neural networks, variational Bayesian inference, etc.). Despite its empirical success, there is still a lack of theoretical understanding of convergence properties of MSGD. To fill this gap, we propose to analyze the algorithmic behavior of MSGD by diffusion approximations for nonconvex optimization problems with strict saddle points and isolated local optima. Our study shows that the momentum helps escape from saddle points but hurts the convergence within the neighborhood of optima (if without the step size annealing or momentum annealing). Our theoretical discovery partially corroborates the empirical success of MSGD in training deep neural networks.

Funding: T. Liu and E. Zhou were supported by the National Science Foundation [Grant CAREER CMMI-1453934] and the Air Force Office of Scientific Research [Grant FA9550-19-1-0283]. T. Zhao was partially supported by the National Science Foundation [Grant III-1717916].

Supplemental Material: The online companion is available at https://doi.org/10.1287/stsy.2021.0083.

Keywords: nonconvex optimization • SGD with momentum • saddle points and optima • diffusion approximation

1. Introduction

Nonconvex stochastic optimization naturally arises in many machine learning problems. Taking training deep neural networks as an example, given n samples denoted by $\{(x_i, y_i)\}_{i=1}^n$, where x_i is the ith input feature and y_i is the response, we solve the following optimization problem:

$$\min_{\theta} \mathcal{F}(\theta) := \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f(x_i, \theta)), \tag{1}$$

where ℓ is a loss function, f denotes the decision function based on the neural network, and θ denotes the parameter associated with f. Stochastic gradient descent (SGD), which has been known for a long time as stochastic approximation in the control and simulation literature (Robbins and Monro 1951, Borkar and Meyn 2000, Kushner and Yin 2003, Borkar 2009, Fu et al. 2015), has been applied to solve machine learning problems such as Problem (1) (Newton et al. 2018). Momentum stochastic gradient descent (MSGD; Polyak 1964) is one of the most popular variants of SGD. Specifically, at the tth iteration, we uniformly sample i from $(1, \dots, n)$. Then, we take

$$\theta^{(t+1)} = \theta^{(t)} - \eta\nabla\ell(y_i, f(x_i, \theta^{(t)})) + \mu(\theta^{(t)} - \theta^{(t-1)}), \tag{2}$$

where η is the step size parameter and $\mu \in [0, 1)$ is the parameter for controlling the momentum. Note that when $\mu = 0$, iterate (2) is reduced to vanilla stochastic gradient descent (VSGD).

Although SGD-type algorithms have demonstrated significant empirical success for training deep neural networks, their convergence properties for nonconvex optimization are still largely unknown. For VSGD, existing literature (Ghadimi and Lan 2013) shows that it is guaranteed to converge to a first-order optimal solution (i.e., $\nabla\mathcal{F}(\theta) = 0$) under general smooth nonconvex optimization.

The theoretical investigation of MSGD is even more limited than that of VSGD. The momentum in iterate (2) has been observed to significantly accelerate computation in practice. To the best of our knowledge, we are only aware of Ghadimi and Lan (2016) in existing literature, which shows that MSGD is guaranteed to converge to a first-order optimal solution for smooth nonconvex problems. Their analysis, however, does not justify the advantage of the momentum in MSGD over VSGD.

To fill the gap between the significant empirical success and the lack of theoretical understanding of MSGD, we are interested in answering a natural and fundamental question in this paper.

What is the role of the momentum in nonconvex stochastic optimization?

The major technical bottleneck in analyzing MSGD and answering the question comes from the nonconvex optimization landscape of these highly complicated problems (e.g., training large recommendation systems and deep neural networks). We propose to analyze MSGD for nonconvex optimization problems under the assumption of isolated local optima and strict saddle points. This allows us to make progress toward understanding MSGD and gaining new insights on more general problems. Specifically, we consider the following problem:

$$\min_{x\in\mathbb{R}^d} \mathcal{F}(x) = \mathbb{E}[f(x,\xi)],$$

where ξ is a random variable representing the noiseand $f(x,\xi)$ is nonconvex in x given any realization of ξ. We assume that the nonconvex landscape has the following structures. (1) Every local optimum has positive curvatures along all directions; (2) there always exist negative curvatures around saddle points (strict saddle property).

The strict saddle property is shared by several popular nonconvex optimization problems arising in machine learning and signal processing, including streaming principle component analysis (PCA), matrix regression/completion/sensing, independent component analysis, partial least square multiview learning, and phase retrieval (Ge et al. 2016, Li et al. 2016, Sun et al. 2016). Moreover, because there is a significant lack of understanding of the optimization landscape of general nonconvex problems, many researchers suggest that analyzing strict saddle optimization problems should be considered as the first important step toward understanding the algorithmic behaviors in general nonconvex optimization. We also want to remark that our analysis can be extended to connected local optima cases. Doing so requires more technical machinery instead of fundamental insights. Therefore, we present the analysis for the isolated optima case for readability and simplicity, and it has already conveyed our core ideas on the effect of momentum.

By making use of the diffusion approximation of stochastic optimization, we provide global and local analysis of MSGD. Specifically, to study the global dynamics, we transfer the discrete time trajectory to a continuous time one by interpolation and prove that asymptotically this continuous time solution trajectory of MSGD converges weakly to the solution of an appropriately constructed ordinary differential equation (ODE). This ODE approximation shows that the momentum helps traverse among stationary points in the nonstationary region, where the variance of the stochastic gradient can be neglected compared with the large magnitude of the gradient. Intuitively, with the help of the momentum, the algorithm makes more progress along the descent direction. Thus, the momentum can accelerate the algorithm in this region by a factor of $1/(1-\mu)$.

ODE approximation, however, cannot justify how momentum works in the stationary area where the variance of the stochastic gradient dominates the update. To highlight the effect of the variance, we consider the asymptotic behavior of the normalized estimation error obtained by MSGD around the stationary points. We show that the continuous time interpolation of the normalized error sequence converges weakly to a solution of an approximately constructed stochastic differential equation (SDE). By analyzing this SDE solution, we find that the momentum can play different but important roles around saddle points and local optima.

- The momentum helps escape from the neighborhood of saddle points. In this region, because the gradient diminishes, the variance of the stochastic gradient dominates the algorithmic behavior. Our analysis indicates that the momentum greatly increases the variance and perturbs the algorithm more aggressively. Thus, it becomes harder for the algorithm to stay around saddle points. In addition, the momentum also encourages more aggressive exploitation, and in each iteration, the algorithm makes more progress along the descent direction by a factor of $1/(1-\mu)$, where μ is the momentum parameter, compared with the VSGD.
- The momentum hurts the convergence within the neighborhood of local optima. Similar to the neighborhood of saddle points, the gradient dies out, and the variance of the stochastic gradient dominates. Because the momentum increases the variance, it is harder for the algorithm to enter the small neighborhood. To this respect, the momentum hurts in this region. We suggest to apply a step size annealing scheme to neutralize the large variance introduced by momentum within the neighborhood of local optima.

Our ODE/SDE approximation analysis justifies the role of momentum in both stationary and nonstationary areas. However, given the complicated nonconvex landscape, our diffusion approximation analysis cannot establish the second-order convergence guarantee and the asymptotic convergence rate of MSGD. Therefore, we further provide a simple but highly nontrival example, streaming PCA, to illustrate our characterization of the effect of momentum and also establish the asymptotic convergence results.

Streaming PCA is a nonconvex problem with only one global optimum and $d-1$ strict saddle points up to sign change, where d is the dimension. Its optimization landscape contains the following three regions:

- $\mathcal{R}_1$: the region containing the neighborhood of strict saddle points with negative curvatures;
- $\mathcal{R}_2$: the region including the set of points whose gradient has sufficiently large magnitude;
- $\mathcal{R}_3$: the region containing the neighborhood of all local optima with positive curvatures along all directions.

By studying the corresponding mean ODE and SDE, we show that with arbitrary initialization, MSGD can converge to the global optimum. We provide asymptotic convergence rates of MSGD, which precisely quantify the acceleration by momentum in $\mathcal{R}_1$ and $\mathcal{R}_2$. Meanwhile, we also show that with the proper step size annealing scheme implemented, MSGD can achieve the same convergence rate as VSGD in $\mathcal{R}_3$.

Our characterization helps explain some phenomena observed when training deep neural networks. There have been some empirical observations and theoretical results (Choromanska et al. 2015) showing that saddle points are the major computation bottleneck, and VSGD usually spends most of the time traveling along saddle and nonstationary regions. Because the momentum helps in both regions, we can find in practice that MSGD performs better than VSGD. In addition, from our analysis, the momentum hurts convergence within the neighborhood of the optima. However, we can address this problem by decreasing the step size or the momentum parameter.

We further verify our theoretical findings through numerical experiments on training a residual network (He et al. 2016), using both Canadian Institute For Advanced Research (CIFAR)-10 and CIFAR-100 data sets. The experimental results show that the algorithmic behavior of MSGD is consistent with our analysis. Moreover, we observe that with a proper initial step size and a proper step size annealing process, MSGD eventually achieves better generalization accuracy than that of VSGD in training neural networks.

To the best of our knowledge, our proposed theory is the first attempt toward understanding the role of momentum in nonconvex stochastic optimization beyond the convergence to stationary solutions. Taking our results as an initial start, we expect more sophisticated and stronger follow-up work for analyzing MSGD (e.g., extending our asymptotic theory to its nonasymptotic counterpart). Please refer to Section 7 for more detailed discussions.

The rest of the paper is organized as follows. Section 2 introduces our nonconvex optimization problem settings and MSGD for solving the problem. Sections 3 and 4 analyze the global and local dynamics of MSGD based on diffusion approximations, respectively. Section 5 studies streaming PCA and provides an asymptotic convergence rate analysis. Section 6 presents the numerical experiments on both streaming PCA and training deep neural networks to demonstrate our theoretical results. Section 7 makes some further discussions on the related literature, our theoretical and experimental results, and future work. The online companion includes all technical details.

2. MSGD

Recall that we study MSGD for a general nonconvex optimization problem as follows:

$$\min_{x\in\mathbb{R}^d} \mathcal{F}(x) = \mathbb{E}[f(x,\xi)], \tag{3}$$

where ξ is a random variable representing the noise and $f(x,\xi)$ is nonconvex in x given any realization of ξ. We assume that there is a stochastic gradient oracle taking $x' \in \mathbb{R}^d$ as input and outputting $\nabla f(x',\xi')$, where ξ' is a realization of the noise ξ, such that

$$\mathbb{E}[\nabla f(x',\xi')] = \nabla\mathcal{F}(x'), \quad \mathrm{Cov}[\nabla f(x',\xi')] = \Sigma.$$

Given a vector $v = (v^{(1)},\ldots,v^{(d)})^\top \in \mathbb{R}^d$, we define the vector norm: $\|v\|_2^2 = \sum_j (v^{(j)})^2$. We impose the following standard assumptions on the objective $\mathcal{F}(x)$ and $f(x,\xi)$.

Assumption 1.

- *Uniform boundedness. There exists a constant C such that* $\|\nabla f(x,\xi)\|_2 \le C, \quad \forall x,\xi.$
- *Lipschitz continuous. There exists a constant L such that*

$$\|\nabla\mathcal{F}(x_1) - \nabla\mathcal{F}(x_2)\|_2 \le L\|x_1 - x_2\|_2, \qquad \forall x_1, x_2 \in \mathbb{R}^d.$$

In general, the optimization landscape of Problem (3) can be very complicated with numerous local optima and saddle points. Here, we consider the case where all the saddle points satisfy the strict saddle property and every local optimum is isolated as stated in Assumption 2.

Assumption 2 (Isolated Optima and Strict Saddle Points).

Denote $S = \{x \in \mathbb{R}^d \mid \nabla\mathcal{F}(x) = 0\}$ as the set of all stationary points. For $x' \in S$, x' must be one of the following:

- *A strict saddle point such that $\lambda_{\min}(\nabla^2\mathcal{F}(x')) < 0$.*
- *An isolated local optimum such that $\lambda_{\min}(\nabla^2\mathcal{F}(x')) > 0$.*

We want to remark that our analysis can also be extended to study the case of connected local optima. However, the proof will be much more involved. Please refer to Section 7 for a detailed discussion.

We apply SGD with Polyak's momentum (Polyak 1964) (MSGD for short) to solve Problem (3). At the kth iteration, MSGD takes the following update:

$$x_{k+1} = x_k - \eta\nabla f(x_k, \xi_k) + \mu(x_k - x_{k-1}), \tag{4}$$

where $\eta > 0$ is the step size and $\mu(x_k - x_{k-1})$ is the momentum with the momentum parameter $\mu \in [0, 1)$. When $\mu = 0$, iterate (4) is reduced to SGD. We remark that although we focus on Polyak's momentum, extending our theoretical analysis to Nesterov's momentum (Nesterov 1983) is straightforward.

3. Analyzing Global Dynamics by ODE

We first analyze the global dynamics of MSGD by taking a diffusion approximation approach. Roughly speaking, by taking the step size $\eta \to 0$, the continuous time interpolation of the iterations $\{x_k\}_{k=0}^{\infty}$, which can be treated as a stochastic process with Càdlàg paths (right continuous with left-hand limits), becomes a continuous stochastic process. For MSGD, this continuous process follows an ODE with a unique solution. This ODE helps us understand how the momentum affects the global dynamics. We remark that the momentum parameter μ is a *fixed constant* in our analysis.

More precisely, we define the continuous time interpolation $X^\eta(\cdot)$ of the solution trajectory of the algorithm as follows. For $t \geq 0$, set $X^\eta(t) = x_k^\eta$ on the time interval $[k\eta, k\eta + \eta)$. Throughout our analysis, similar notations are applied to other interpolations (e.g., $H^\eta(t)$, $U^\eta(t)$). We then answer the following questions.

Does the solution trajectory sequence $\{X^\eta(\cdot)\}_\eta$ converge weakly as η goes to zero? If so, what is the limit?

This question has been studied for VSGD in the existing literature for special nonconvex optimization problems, such as streaming PCA in Chen et al. (2017). The infinitesimal perturbation analysis (IPA) technique is widely used to show that under some regularity conditions, $X^\eta(\cdot)$ converges weakly to a solution of the following ODE:

$$\dot{X}(t) = -\nabla\mathcal{F}(X(t)). \tag{5}$$

This method, however, cannot be applied to analyze MSGD because of the additional momentum term. Here, we explain why this method fails. Rewrite Algorithm (4) as

$$\delta_{k+1} = \mu\delta_k - \eta\nabla f(x_k, \xi_k), \quad x_{k+1} = x_k + \delta_{k+1}.$$

One can easily check that (δ_k, x_k) is Markovian. To apply IPA, the infinitesimal conditional expectation (ICE) must converge to a constant. However, the ICE for MSGD, which can be calculated as

$$\frac{\mathbb{E}[\delta_{k+1} - \delta_k \mid \delta_k, x_k]}{\eta} = \frac{(\mu - 1)\delta_k}{\eta} - \nabla\mathcal{F}(x_k),$$

goes to infinity (blows up). Thus, IPA is not applicable here.

To address this challenge, we provide a new technique to prove the weak convergence and find the desired ODE. In a nutshell, we first prove rigorously the weak convergence of the trajectory sequence and then, use martingale theory to find the ODE. To be self-contained, we provide a summary on the prerequisite weak convergence theory in Online Companion Section 1.

Under Assumption 1, we characterize the global behavior of MSGD as follows.

Theorem 1. *Let $D^d[0, \infty)$ be the space of $\mathbb{R}^d$-valued operators, which are right continuous and have left-hand limits for each dimension. Suppose $x_0 = x_1 \in \mathbb{R}^d$. Then, for each subsequence of $\{X^\eta(\cdot)\}_{\eta>0}$, there exists a further subsequence and a process $X(\cdot)$ such that $X^\eta(\cdot) \Rightarrow X(\cdot)$ in the weak sense as $\eta \to 0$ through the convergent subsequence in the space $D^d[0, \infty)$, where $X(\cdot)$ satisfies the following ODE:*

$$\dot{X} = -\frac{1}{1-\mu}\nabla\mathcal{F}(X), \quad X(0) = x_0. \tag{6}$$

Moreover, for any $\delta > 0$, there exists a sequence $T^{\eta,\delta} \to \infty$ such that

$$\limsup_{\eta\to 0} \mathbb{P}(\exists t \leq T^{\eta,\delta},\ X^\eta(t) \neq N_\delta(S)) = 0,$$

where $N_\delta(S)$ is the δ neighborhood of the stationary points.

Proof Sketch. To prove this theorem, we first show that the trajectory sequence $\{X^\eta(\cdot)\}_\eta$ converges weakly. By Prokhorov's theorem 1 (in Online Companion Section 1), we need to prove tightness, which means that $\{X^\eta(\cdot)\}_\eta$ is bounded in probability in space $D^d[0,\infty)$. This can be proved by Theorem 4 (in Online Companion Section 1), which requires the following two conditions. (1) x_k must be bounded in probability for any k uniformly in step size η. (2) The maximal discontinuity (the largest difference between two iterations; i.e., $\max_k\{x_{k+1}-x_k\}$) must go to zero as η goes to zero. This can be shown by using the bounded gradient assumption.

We next compute the weak limit. For simplicity, we define

$$\beta_k^\eta = -\sum_{i=0}^{k-1}\mu^{k-i}[\nabla f(x_i^\eta,\xi_i)-\nabla\mathcal{F}(x_i^\eta)] \quad \text{and} \quad \epsilon_k^\eta = -(\nabla f(x_k^\eta,\xi_k)-\nabla\mathcal{F}(x_k^\eta)).$$

We then rewrite the algorithm as follows:

$$m_{k+1}^\eta = m_k^\eta + (1-\mu)[-m_k^\eta + \widetilde{M}(x_k^\eta)], \quad x_{k+1}^\eta = x_k^\eta + \eta(m_{k+1}^\eta + \beta_k^\eta + \epsilon_k^\eta), \tag{7}$$

where $\widetilde{M}(x) = -(1-\mu)^{-1}\nabla\mathcal{F}(x)$. The basic idea of the proof is to view iterate (7) as a two-timescale algorithm (Borkar 1997, 2009), where m_k is updated with a larger step size $(1-\mu)$ and thus, under a faster time scale and v_k is under a slower one. Then, we can treat the slower timescale iterate v as static and replace the faster timescale iterate m by its stable point in terms of this fixed v in iterate (7). This stable point can be shown to be $\widetilde{M}(x)$.

We then show that the continuous time interpolation of the error $[m_{k+1}^\eta - \widetilde{M}(x_k^\eta)] + \beta_k^\eta + \epsilon_k^\eta$ converges weakly to a Lipschitz continuous martingale with zero initialization. From the martingale theory, we know that such kinds of martingales must be a constant. Thus, the error sequence converges weakly to zero, and what is left is actually the discretization of ODE (6). Please refer to Online Companion Section 2 for the detailed proof. □

Note that for any solution $X(t)$ to ODE (5) (i.e., the mean ODE of SGD), $X(t/(1-\mu))$ is a solution to ODE (6). This implies that asymptotically, MSGD is $1/(1-\mu)$ faster than SGD to converge to the neighborhood of a stationary point given the same initialization. Intuitively, with the help of the momentum, the algorithm makes more progress along the descent direction, and therefore, momentum can accelerate the algorithm asymptotically.

However, because the noise of the stochastic gradient diminishes as $\eta \to 0$, such a deterministic ODE-based approach is insufficient to analyze the local behavior of MSGD around stationary points where the noise plays a dominant role over the vanishing gradient. Thus, we resort to the following SDE-based approach for a more precise characterization.

4. Analyzing Local Dynamics by SDE

To characterize the local algorithmic behavior, we need to rescale the influence of the noise. For this purpose, we consider the *normalized error* $(x_k - x^*)/\sqrt{\eta}$ under the diffusion approximation framework, where $x^* \in S$ is a stationary point. Different from the previous ODE-based approach, we obtain an SDE approximation here. Intuitively, the previous ODE-based approach is analogous to the Law of Large Numbers for random variables, whereas the SDE-based approach serves the same role as the Central Limit Theorem.

Recall that under Assumption 2, the optimization problem (3) has strict saddle points and isolated local optima. We remark that the assumption on isolated local optima helps avoid the cases where the normalization error explodes when the iterate wanders along the connected local optima. Our analysis can be further extended to handle connected local optima. However, the analysis will be much more complicated. Please refer to Section 7 for a detailed discussion. For consistency, we first study the algorithmic behavior around the local optimum.

Remark 1. The $\sqrt{\eta}$ normalization actually normalizes the error by its standard deviation. Specifically, consider the $\lfloor 1/\eta \rfloor$ th iterate of SGD initialized at the stationary point x^*,

$$x_{\lfloor 1/\eta\rfloor} = x^* - \eta\sum_{i=0}^{\lfloor 1/\eta\rfloor -1}\nabla f(x_i) - \eta\sum_{i=0}^{\lfloor 1/\eta\rfloor -1}\xi_i,$$

where $f(x)$ is the objective to be maximized and $\{\xi_i\}_i$ are the noises in the stochastic gradient independently and identically distributed (i.i.d.) sampled from some unknown distribution with mean zero and bounded variance. Given the continuity of the gradient, $\nabla f(x_i)$ is approximately zero, and noise will dominate around the stationary point x^*. Therefore, $x_{\lfloor 1/\eta\rfloor}$ can be further approximated as follows:

$$x_{\lfloor 1/\eta\rfloor} \approx x^* - \eta\sum_{i=0}^{\lfloor 1/\eta\rfloor -1}\xi_i.$$

Thus, the variance of the error $x_{\lfloor 1/\eta \rfloor} - x^*$ is of order $O(\eta)$:

$$\mathrm{Var}(x_{\lfloor 1/\eta \rfloor} - x^*) = \eta^2 \mathrm{Var}\left(\sum_{i=0}^{\lfloor 1/\eta \rfloor - 1} \xi_i \right) = O(\eta).$$

Therefore, we actually normalize the error by its standard deviation $O(\sqrt{\eta})$, which is analogous to rescaling the sample sum by $\sqrt{N}$ in Central Limit Theorem.

4.1. Local Dynamics Around Local Optima

We first consider the algorithmic behavior of MSGD when it is around a local optimum x^*. Define the normalized process $u_k^\eta = (x_k^\eta - x^*)/\sqrt{\eta}$, where $\lambda_{\min}(\nabla^2 \mathcal{F}(x^*)) > 0$. Accordingly, $U^\eta(t) = (X^\eta(t) - x^*)/\sqrt{\eta}$. The next theorem characterizes the limiting process of $U^\eta(t)$.

Theorem 2. *As $\eta \to 0$, $\{U^\eta(\cdot)\}$ converges weakly to the unique stationary solution of*

$$dU = -\frac{1}{1-\mu} \nabla^2 \mathcal{F}(x^*) U dt + \frac{1}{1-\mu} dW_t, \tag{8}$$

where $\{W_t\}$ is a Wiener process with covariance matrix $\Sigma = \mathbb{E}[\nabla f(x^*, \xi) \nabla f(x^*, \xi)^\top]$.

Note that our analysis is very different from that in Chen et al. (2017) because of the failure of IPA because of the similar blowup issue. We remark that our technique mainly relies on theorem 5 (in Online Companion Section 1) from Kushner and Yin (2003). Because the proof is much more sophisticated and involved than IPA, we introduce the key technique, fixed-state chain, in a high level.

Proof Sketch. Note that the algorithm can be rewritten as

$$x_{k+1}^{\eta,i} = x_k^{\eta,i} - \eta \left[\sum_{j=1}^{k-1} \mu^{k-j} \nabla f\left(x_j^\eta, \xi_j\right) + \nabla \mathcal{F}(x_k) \right] - \eta \left[\nabla f\left(x_k^\eta, \xi_k\right) - \nabla \mathcal{F}\left(x_k^\eta\right) \right].$$

Here, for a vector $x \in \mathbb{R}^d$ and an integer $i \le d$, $x^{(i)}$ represents the ith dimension of x. We define

$$\zeta_k^\eta = -\left[\sum_{j=1}^{k-1} \mu^{k-j} \nabla f(x_j^\eta, \xi_j) \right], \quad Z_k^\eta = g\left(\zeta_k^\eta, x_k^\eta\right) + \gamma_k^\eta,$$

$$\gamma_k^\eta = \nabla \mathcal{F}\left(x_k^\eta\right) - \nabla f\left(x_k^\eta, \xi_k\right), \quad \text{and} \quad g\left(\zeta_k^\eta, x_k^\eta\right) = \zeta_k^\eta - \nabla \mathcal{F}\left(x_k^\eta\right).$$

Here, g is the accelerated gradient flow, and γ_k^η is the noise. Then, the algorithm becomes

$$x_{k+1}^\eta = x_k^\eta + \eta Z_k^\eta = x_k^\eta + \eta g(\zeta_k^\eta, x_k^\eta) + \eta \gamma_k^\eta,$$

and thus, we have $u_{k+1}^\eta = u_k^\eta + \sqrt{\eta}[g(\zeta_k^\eta, x_k^\eta) + \gamma_k^\eta]$. Note that $g(\zeta_k^\eta, x_k^\eta) \in \mathcal{F}_k^\eta$ and $\mathbb{E}[\gamma_k^\eta \mid \mathcal{F}_k^\eta] = 0$ imply that the noise $\{\gamma_k^\eta\}$ is a martingale difference sequence.

We then manipulate the algorithm to extract the Markov structure of the algorithm in an explicit form. To make it clear, given X, there exists a transition function $P(\cdot, \cdot \mid X)$ such that

$$P\{\zeta_{k+1}^\eta \in \cdot \mid \mathcal{F}_k^\eta\} = P(\zeta_k^\eta, \cdot \mid X = x_k^\eta).$$

This comes from the observation $\zeta_{k+1}^\eta = \mu \zeta_k^\eta - \mu \nabla f(x_k^\eta, \xi_k^\eta)$, where the randomness only comes from ξ_k when state x_k is given. Then, the fixed-state chain refers to the Markov chain with transition function $P(\cdot, \cdot \mid X)$ for a fixed X. The state of this Markov chain is denoted by $\{\zeta_k(X)\}$. For notational simplicity, let $\widetilde{M}(x) = -1/(1-\mu) \nabla \mathcal{F}(x)$. We then decompose $x_{k+1}^\eta - x_k^\eta$ as follows:

$$\begin{aligned} x_{k+1}^\eta - x_k^\eta = {} & \eta \widetilde{M}(x_k^\eta) + \eta \gamma_k^\eta + \eta[g(\zeta_k(x_k^\eta), x_k^\eta) - \widetilde{M}(x_k^\eta)] \\ & + \eta[g(\zeta_k^\eta, x_k^\eta) - g(\zeta_k(x_k^\eta), x_k^\eta)] = \eta \widetilde{M}(x_k^\eta) + \eta W_k^\eta. \end{aligned} \tag{9}$$

The error term W_k^η in (9) comes from three sources: (1) difference between the fixed-state chain and the limiting process: $g(\zeta_k(x_k^\eta), x_k^\eta) - \widetilde{M}(x_k^\eta)$; (2) difference between the accelerated gradient flow and the fixed-state chain: $g(\zeta_k^\eta, x_k^\eta) - g(\zeta_k(x_k^\eta), x_k^\eta)$; and (3) the noise γ_k^η.

We handle them separately and combine the results together to get the variance of $W_k^{\eta,i}$. Note that $\{u_k^\eta\}$ satisfies the following update:

$$u_{k+1}^\eta - u_k^\eta = \sqrt{\eta}\widetilde{M}(x_k^\eta) + \sqrt{\eta}W_k^\eta.$$

Together with the fact that around the optimum x^*, $\widetilde{M}(x) = -1/(1-\mu)\nabla^2\mathcal{F}(x^*)(x-x^*) + o(\|(x-x^*)\|_2)$, we further obtain

$$\frac{u_{k+1}^\eta - u_k^\eta}{\eta} = -\frac{1}{1-\mu}\nabla^2\mathcal{F}(x^*)u_k^\eta + \frac{W_k^\eta}{\sqrt{\eta}} + o\left(|\ u_k^{\eta,1}\ |\right). \tag{10}$$

After calculating the variance of W, we see that essentially (10) is the discretization of SDE (8). For the detailed proof, please refer to Online Companion Section 3.1. □

Note that SDE (8) admits an explicit solution, which is known as an Ornstein–Uhlenbeck (O-U) process (Øksendal 2003), having the following expression:

$$U(t) = \exp\left(-\frac{1}{1-\mu}\nabla^2\mathcal{F}(x^*)t\right)U(0) + \int_0^t \exp\left(\frac{1}{1-\mu}\nabla^2\mathcal{F}(x^*)(t-s)\right)\frac{\Sigma^{\frac{1}{2}}}{1-\mu}dB_t.$$

Given $U(0)$, the formula shows that $U(t)$ is Gaussian for all $t > 0$. Therefore, we can identify the limiting density of $U(t)$ as $t \to \infty$ by figuring out the limiting mean and covariance matrix. In fact, the mean $m(t) = \mathbb{E}(U(t))$ and covariance matrix $\rho_t = \mathbb{E}[U(t)U(t)^\top]$ satisfy the following ODEs, respectively:

$$\begin{aligned} dm(t) &= -\frac{\nabla^2\mathcal{F}(x^*)}{1-\mu}m(t)dt,\\ d\rho(t) &= -\frac{1}{1-\mu}\left(\nabla^2\mathcal{F}(x^*)\rho + \rho\nabla^2\mathcal{F}(x^*)\right) + \frac{1}{(1-\mu)^2}\Sigma. \end{aligned}$$

Because $\nabla^2\mathcal{F}(x^*)$ is positive definite, we have $m(t) \to 0$ and

$$\rho_\mu = \lim_{t\to\infty}\rho(t) = \int_0^\infty \exp\left(-\frac{1}{1-\mu}\nabla^2\mathcal{F}(x^*)s\right)\frac{1}{(1-\mu)^2}\Sigma\exp\left(-\frac{1}{1-\mu}\nabla^2\mathcal{F}(x^*)s\right)ds < \infty.$$

Therefore, when MSGD enters the neighborhood of a local optimum, it will stay near the local optimum and behave like a Brownian motion. Moreover, by a change of variables, we can rewrite ρ_μ as follows:

$$\begin{aligned} \rho_\mu &= \int_0^\infty \exp\left(-\frac{1}{1-\mu}\nabla^2\mathcal{F}(x^*)s\right)\frac{1}{(1-\mu)^2}\Sigma\exp\left(-\frac{1}{1-\mu}\nabla^2\mathcal{F}(x^*)s\right)ds\\ &= \frac{1}{(1-\mu)}\int_0^\infty \exp\left(-\nabla^2\mathcal{F}(x^*)s\right)\Sigma\exp\left(-\nabla^2\mathcal{F}(x^*)s\right)ds\\ &= \frac{1}{(1-\mu)}\rho_0. \end{aligned}$$

We see clearly that the momentum essentially increases the variance of the normalized error by a factor of $1/(1-\mu)$ around the local optimum compared with VSGD. Thus, it becomes harder for the algorithm to converge. The next theorem provides a more precise characterization of such a phenomenon.

Theorem 3. *Let (λ_i, e_i) be the eigenvalue, eigenvector pairs of $\nabla^2\mathcal{F}(x^*)$ such that $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d > 0$. Given a sufficiently small $\epsilon > 0$ and $\phi = \sum_{i=1}^d \Sigma_{i,i} < \infty$, we need the step size η satisfying*

$$\eta < (1-\mu)\lambda_d\epsilon/(4\phi) \tag{11}$$

such that $X^\eta(t)$ enters the ϵ neighborhood of the local optimum with probability at least 3/4 at some time T_3 after restarting the counter of time: that is, $\|X^\eta(T_3) - x^*\ \|_2^2 \leq \epsilon$, where

$$T_3 \asymp \frac{(1-\mu)}{2\lambda_d}\cdot\log\left(\frac{8\lambda_d\delta^2}{\lambda_d\epsilon - 4\eta\phi}\right),$$

given $\|X^\eta(0) - x^*\ \|_2^2 \leq \delta^2$.

Note that when $\mu = 0,$ we can choose the step size of VSGD as $\eta_0 \asymp \lambda_d \epsilon/(4\phi)$, which does not satisfy (11) for μ close to one. This means that when using the same step size of VSGD, MSGD fails to converge because the variance increased by the momentum becomes too large. To handle this issue, we have to decrease the step size by a factor $1 - \mu$, also known as the step size annealing: that is,

$$\eta \asymp (1-\mu)\epsilon\lambda_d/(4\phi) \asymp (1-\mu)\eta_0. \tag{12}$$

We also want to remark that here the probability $3/4$ can be any constant in $(0,1)$. Theorem 3 implies the algorithm needs asymptotically at most

$$N_3 \asymp \frac{T_3}{\eta} \asymp \frac{\phi}{\epsilon\lambda_d^2} \cdot \log\left(\frac{8\lambda_d\delta^2}{\lambda_d\epsilon - 4\eta_0\phi}\right)$$

iterations to converge to an ϵ-optimal solution. Note that the N_3 does not depend on μ. Therefore, MSGD does not have an advantage over VSGD around local optima.

4.2. Local Dynamics Around Saddle Points

We then study the algorithmic behavior around strict saddle points. Define the normalized process $u_k^\eta = (x_k^\eta - \hat{x})/\sqrt{\eta}$, where $\hat{x} \in S, \lambda_{\min}(\nabla^2\mathcal{F}(\hat{x})) < 0$. Accordingly, $U^\eta(t) = (X^\eta(t) - \hat{x})/\sqrt{\eta}$. By the same SDE approximation technique used in Section 4.1, we obtain Theorem 4.

Theorem 4. *For any $C > 0,$ there exist $\delta > 0$ and $\eta' > 0$ such that*

$$\sup_{\eta<\eta'} \mathbb{P}(\sup_{\tau>0} \|U^\eta(\tau)\|_2 \leq C) \leq 1 - \delta. \tag{13}$$

Proof Sketch. We prove (13) by contradiction. Assume that the conclusion does not hold: that is, there exists a constant $C > 0,$ such that for any $\eta' > 0,$ we have

$$\sup_{\eta\leq\eta'} \mathbb{P}\left(\sup_{\tau>0} |\, U^\eta(\tau)\|_2 \leq C\right) = 1.$$

That implies there exists a sequence $\{\eta_n\}_{n=1}^\infty$ converging to zero such that

$$\lim_{n\to\infty} \mathbb{P}\left(\sup_{\tau>0} |\, U^{\eta_n}(\tau)\|_2 \leq C\right) = 1. \tag{14}$$

We next show that this subsequence $\{U^{\eta_n}(\cdot)\}_n$ is tight. To do so, we need to verify two conditions of theorem 3 in Online Companion Section 1: (i) for large enough n, U^{η_n} is bounded in probability; (2) the modulus of continuous converges to 0 in probability as $n\to\infty$. By (14), we know that condition (i) in theorem 3 holds. We next check condition (ii) in theorem 3. When $\sup_{\tau>0} |\, U^{\eta_n,i}(\tau)\,| \leq C$ holds, Assumption 1 yields that $\|u_{k+1}^{\eta_n} - u_k^{\eta_n}\|_2 \leq C'\eta_n$, where C' is some constant. Thus, for any $t, \epsilon > 0,$ we have

$$\|_2 U^{\eta_n}(t) - U^{\eta_n}(t+\epsilon)\|_2 \leq \epsilon/\eta C'\eta = C'\epsilon,$$

or equivalently,

$$\varpi_T'(U^{\eta_n}, \epsilon) \leq C'\epsilon,\ \forall T > 0,$$

where ϖ is the modulus of continuous defined in definition 3. Thus, condition (ii) in theorem 3 holds. Then, we have that $\{U^{\eta_n}(\cdot)\}_n$ is tight and thus, converges weakly. Following similar lines to Theorem 2, we can verify C.5–C.8 in theorem 5 in Online Companion Section 1, and show that $\{U^{\eta_n}(\cdot)\}_n$ converges weakly to a solution of

$$dU = -\frac{1}{1-\mu}\nabla^2\mathcal{F}(\hat{x})Udt + \frac{1}{1-\mu}dW_t. \tag{15}$$

The process defined by SDE (15) is an unstable O-U process. When initialized at $\hat{x},$ it has mean zero and exploding variance. When not initialized at $\hat{x},$ it has exploding mean and variance. Thus, for any $\delta,$ there exists a time τ', such that

$$\mathbb{P}(\|U(\tau')\|_2 \geq C) \geq 2\delta.$$

Because $\{U^{\eta_n}\}_n$ converges weakly to U, $\{U^{\eta_n}(\tau')\}_n$ converges in distribution to $U(\tau')$. This implies that there exists $n > 0$, such that for any $n > N,$

$$|\,\mathbb{P}(\|U(T)\|_2 \geq C) - \mathbb{P}(\|U^{\eta_n}(T)\|_2 \geq C)\,| \leq \delta.$$

Then, we find a $\tau' > 0$ such that

$$\mathbb{P}(\|U^{\eta_n}(\tau')\|_2 \geq C) \geq \delta, \quad \forall n > N,$$

or equivalently,

$$\mathbb{P}(\|U^{\eta_n}(\tau')\|_2 \leq C) < 1 - \delta, \quad \forall n > N.$$

Because $\{\omega \mid \sup_\tau \|U^{\eta_n}(\tau)(\omega)\|_2 \leq C\} \subset \{\omega \| \mid U^{\eta_n}(\tau')(\omega)\|_2 < C\}$, we have

$$\mathbb{P}(\sup_\tau \|U^{\eta_n}(\tau)\|_2 \leq C) \leq 1 - \delta, \quad \forall n > N,$$

which leads to a contradiction with (14). Our assumption does not hold. We prove Theorem 4. □

Theorem 4 implies that with a constant probability δ, MSGD escapes from the saddle points at some time T_1 (i.e., $\|X^\eta(T_1) - \hat{x}\|_2^2$ is greater than δ^2; $\delta = \mathcal{O}(\sqrt{\eta})$). Note that from the proof of Theorem 4, when the step size η is small, the process defined by SDE (15) characterizes the local behavior of X^η around saddle points. For any fixed μ, let U_μ be the solution to (15). Then, we can verify that

$$\mathbb{E}(U_\mu(t)) = \frac{1}{1-\mu}\mathbb{E}(U_0(t)), \quad \mathrm{Var}(U_\mu(t)) = \frac{1}{1-\mu}\mathrm{Var}(U_0(t)).$$

More precisely, we can obtain the following proposition on the asymptotic escaping rate of MSGD.

Theorem 5. *Let $\nabla^2\mathcal{F}(\hat{x}) = P\Lambda P^\top$ be the eigenvalue decomposition of $\nabla^2\mathcal{F}(\hat{x})$, where $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_d)$ and $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d$ and $\lambda_d < 0$. Denote $\Sigma = \mathbb{E}[\nabla f(\hat{x},\xi)\nabla f(\hat{x},\xi)^\top]$. Given a prespecified $\nu \in (0,1)$, $\eta \asymp \eta_0$, and $\delta = \mathcal{O}(\sqrt{\eta})$, then the following result holds. We need at most*

$$T_1 \asymp \frac{(1-\mu)}{2\,|\,\lambda_d\,|}\log\left(\frac{2\eta^{-1}\delta^2(1-\mu)\,|\,\lambda_d\,|}{\Phi^{-1}\left(\frac{9}{16}\right)^2(P^\top\Sigma P)_{d,d}} + 1\right), \tag{16}$$

such that $\|X_\eta(T_1) - \hat{x}\|_2^2 \geq \delta^2$ with probability at least $\frac{3}{4}$, where $\Phi(x)$ is the cumulative distribution function (CDF) of the standard normal distribution.

Theorem 5 suggests that we need asymptotically

$$N_1 \asymp \frac{(1-\mu)\phi}{|\,\lambda_d|^2\epsilon}\log\left(\frac{2(1-\mu)\eta^{-1}\delta^2\,|\,\lambda_d\,|}{\Phi^{-1}\left(\frac{1+\nu/2}{2}\right)^2(P^\top\Sigma P)_{d,d}} + 1\right)$$

iterations to escape from saddle points. Thus, when using the same step size, MSGD can escape from saddle points in fewer iterations than VSGD by a factor of $1-\mu$. This is because of the fact that the momentum can greatly increase the variance and perturb the algorithm more aggressively. Thus, it becomes harder to stay around saddle points. Moreover, the momentum also encourages more aggressive exploitation, and in each iteration, the algorithm makes more progress along the descent direction by a factor of $1/(1-\mu)$.

In summary, compared with VSGD ($\mu = 0$), momentum accelerates escaping from saddle points by a factor of $1-\mu$. However, momentum can also hurt the final convergence around the local optimum because of the increased variance. Therefore, we suggest to decrease the step size by a factor $1-\mu$ in the later stage, MSGD can then achieve the similar convergence rate as VSGD. Note that we can also decrease the momentum parameter μ instead of the step size η. We will show in Section 6 that momentum annealing and step size annealing can both ensure the convergence of MSGD.

5. Example: Streaming PCA

In this section, we apply our convergence analysis to study the algorithmic behavior of MSGD and provide explicit convergence result for the streaming PCA problem formulated as follows:

$$\max_v \; v^\top\mathbb{E}_{X\sim\mathcal{D}}[XX^\top]v \quad \text{subject to} \quad v \in \mathbb{S} = \{v \in \mathbb{R}^d \mid \|v\|_2 = 1\}. \tag{17}$$

For notational simplicity, we denote the covariance matrix as $\Sigma = \mathbb{E}[XX^\top]$. Before we proceed, we impose the following assumption on Σ.

Assumption 3. *The covariance matrix Σ is positive definite with eigenvalues $\lambda_1 > \lambda_2 \geq \ldots \geq \lambda_d > 0$ and associated normalized eigenvectors $v^1, v^2, \ldots, v^d$. Moreover, there exists an orthogonal matrix Q such that $\Sigma = Q\Lambda Q^\top$, where $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_d)$.*

Under this assumption, the optimization landscape of Problem (17) has been well studied. Chen et al. (2017) have shown that the eigenvectors $\pm v^1, \pm v^2, \ldots, \pm v^d$ are all the stationary points for Problem (17) on the unit sphere $\mathbb{S}$. Moreover, the eigen-gap assumption ($\lambda_1 > \lambda_2$) guarantees that the global optimum v^1 is identifiable up to sign change. Meanwhile, $v^2, \ldots, v^{d-1}$ are $d-2$ strict saddle points, and v^d is the global minimum.

Given the optimization landscape of Problem (17), we have already understood well the behavior of VSGD algorithms, including Oja's rule and stochastic generalized Hebbian algorithms (SGHAs) for streaming PCA (Chen et al. 2017). We consider a variant of SGHA with Polyak's momentum (Polyak 1964). Recall that we are given a streaming data set $\{X_k\}_{k=1}^\infty$ drawn independently from some zero-mean distribution $\mathcal{D}$. At the kth iteration, the algorithm takes

$$v_{k+1} = v_k + \eta(I - v_k v_k^\top)\Sigma_k v_k + \mu(v_k - v_{k-1}), \tag{18}$$

where $\Sigma_k = X_k X_k^\top$ and $\mu(v_k - v_{k-1})$ is the momentum with a parameter $\mu \in [0,1)$. When $\mu = 0$, iterate (18) is reduced to SGHA. A detailed derivation of iterate (18) is provided in Online Companion Section 4.

Remark 2. The constraint in Problem (17) restricts the solution space to be a unit sphere $\mathbb{S}$, which is a manifold. In order to match our Algorithm (2), we consider Problem (17) to be an unconstraint optimization problem on the manifold by using the manifold gradient $(I - xx^\top)\Sigma x$. For general manifold optimization problems, additional projection may be required to ensure that the solution trajectory stays on the manifold. However, for the sphere constraint as in Problem (17), when η is small, moving along the direction of the manifold gradient, the solution trajectory can stay close to $\mathbb{S}$, as shown in lemma 1 in Online Companion Section 4.

Before we proceed, we impose the following assumption on the problem.

Assumption 4. *The data points $\{X_k\}_{k=1}^\infty$ are drawn independently from a distribution $\mathcal{D}$ in R^d, such that*

$$\mathbb{E}[X] = 0,\ \mathbb{E}[XX^\top] = \Sigma,\ \|X\| \leq C_d,$$

where C_d is a constant (possibly dependent on d).

This uniformly boundedness assumption can actually be relaxed to the boundedness of the $(4+\delta)$ th-order moment ($\delta > 0$) with a careful truncation argument. The proof, however, will be much more involved and beyond the scope of this paper. Thus, we use the uniformly boundedness assumption for convenience.

Under Assumptions 3 and 4, we first apply Theorem 1 and provide an ODE approximation for Algorithm (18) in Corollary 1.

Corollary 1. *Suppose $v_0 = v_1 \in \mathbb{S}$. Then, $V^\eta(\cdot) \Rightarrow V(\cdot)$ in the weak sense as $\eta \to 0$ in the space $D^d[0,\infty)$, where $V(\cdot)$ is the unique solution to the following ODE,*

$$\dot{V} = \frac{1}{1-\mu}(\Sigma V - V^\top \Sigma V V),\ \ V(0) = v_0, \tag{19}$$

and has the following explicit form $V(t) = QH(t)$, where

$$H^{(i)}(t) = \left(\sum_{i=1}^d \left[H^{(i)}(0)\exp\left(\frac{\lambda_i t}{1-\mu}\right)\right]^2\right)^{-\frac{1}{2}} H^{(i)}(0)\exp\left(\frac{\lambda_i t}{1-\mu}\right),\quad i = 1, \ldots, d,$$

where $H(0) = Q^\top v_0$. Moreover, suppose $v_0 \neq \pm v^i,\ \forall i = 2, \ldots, d$, as $t \to \infty$, $V(t)$ converges to v^1, which is the global maximum to Problem (17).

Please refer to Online Companion Section 4.2 for the detailed proof. Different from the general ODE (6), ODE (19) has an explicit form solution, which implies that whenever MSGD escapes from strict saddle points v^i, $i \geq 2$, it will directly converge to the global optimum v^1. Therefore, we can provide a more precise characterization of the algorithmic behavior in the nonstationary area for streaming PCA than general nonconvex problems. Moreover, because streaming PCA has one isolated global optimum and strict saddle points, our SDE analysis for the stationary area can be directly applied. We have Corollary 2 to characterize the asymptotic convergence rate of MSGD.

Corollary 2. *Let η be the step size of MSGD and $\eta_0 \asymp (\lambda_1 - \lambda_2)\epsilon/\phi$ be the step size of VSGD as chosen in Chen et al.* (2017).

- *Phase I: Escape from saddle points. Suppose $v_0^\eta = v^2$, the strict saddle point corresponding to λ_2. Given $\eta \asymp \eta_0$ and $\delta = \mathcal{O}(\sqrt{\eta})$, we need asymptotically at most*

$$N_1 \asymp \frac{(1-\mu)\phi}{(\lambda_1-\lambda_2)^2\epsilon}\log\left(\frac{2(1-\mu)\eta^{-1}\delta^2(\lambda_1-\lambda_2)}{\Phi^{-1}\left(\frac{9}{16}\right)^2\alpha_{12}^2}+1\right), \tag{20}$$

iterations such that $\|v_{N_1}^\eta - v_2\|_2^2 \geq \delta^2$ with probability at least $3/4$, where $\Phi(x)$ is the CDF of the standard normal distribution.

- *Phase II: Traverse from saddle points to the global optimum. Suppose $\|v_0^\eta - v^i\|_2^2 \geq \delta^2, \quad \forall i \geq 2$. For sufficiently small η, $\delta = \mathcal{O}(\sqrt{\eta})$, we need*

$$N_2 \asymp \frac{(1-\mu)\phi}{2\epsilon(\lambda_1-\lambda_2)^2}\log\left(\frac{2-\delta^2}{\delta^2}\right) \tag{21}$$

iterations such that $\|v_{N_2}^\eta - v^1\|_2^2 \leq \delta^2$ with probability at least $3/4$.

- *Phase III: Converge to the global optimum. For $\epsilon > 0$ and $\eta \asymp (1-\mu)\eta_0$, there exists some constant $\delta = \mathcal{O}(\sqrt{\eta})$, such that $\|v_0^\eta - v^1\|^2 \leq \delta^2$, we need*

$$N_3 \asymp \frac{\phi}{\epsilon(\lambda_1-\lambda_2)^2}\cdot\log\left(\frac{8(\lambda_1-\lambda_2)\delta^2}{(\lambda_1-\lambda_2)\epsilon - 4\eta_0\phi}\right) \tag{22}$$

iterations to ensure $\|v_{N_3}^\eta - v^1\|_2^2 \leq \epsilon$ with probability at least $3/4$.

Please refer to Online Companion Section 4.3 for the detailed proof. From Corollary 2, we can see clearly that momentum accelerates escaping from saddle points and traversal to the global optimum by a factor of $1-\mu$. If we further decrease the step size in Phase III, MSGD can achieve the same convergence rate as VSGD.

6. Numerical Experiments

We present numerical experiments for both streaming PCA and training deep neural networks. The experiments on streaming PCA verify our theory in Section 5, and the experiments on training deep neural networks support our theoretical results for the general problem and also verify some of our discussions in Section 7.

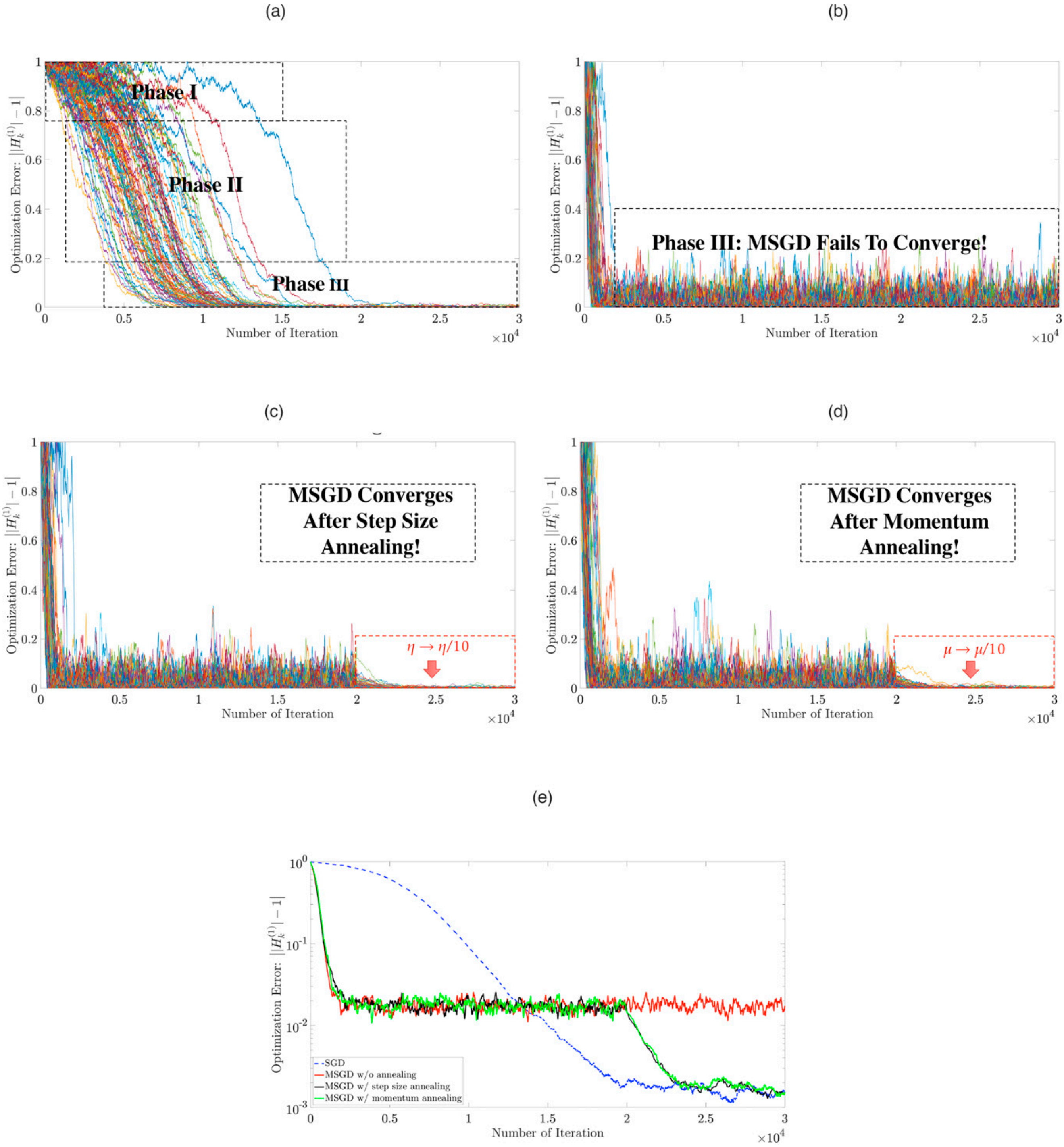
6.1. Streaming PCA

We first provide a numerical experiment to verify our theory for streaming PCA. We set $d = 4$ and the covariance matrix $\Lambda = \text{diag}\{4,3,2,1\}$. The optimum is $(1,0,0,0)$. Figure 1 compares the performance of VSGD and MSGD (with and without the step size annealing and momentum annealing in Phase III). The initial solution is the saddle point $(0,1,0,0)$. We choose $\mu = 0.9$ and $\eta = 5\times10^{-4}$, decrease the step size of MSGD by a factor $1-\mu$ after 2×10^4 iterations in Figure 1(b), and decrease the momentum by a factor $1/10$ after 2×10^4 iterations in Figure 1(c). Figure 1 plots the results of 100 simulations, and the vertical axis corresponds to $\|H_k^{(1)}|-1|$. We can clearly differentiate the three phases of VSGD in Figure 1(a). For MSGD in Figure 1, (b)–(d), we hardly recognize Phases I and II because they last for a much shorter time. This is because the momentum significantly helps escape from saddle points and evolve toward the global optimum. Moreover, we also observe in Figure 1(b) that MSGD without the step size annealing and the momentum annealing does not converge well, but the step size annealing or the momentum annealing resolves this issue. All these observations are consistent with our analysis. Figure 1(e) plots the optimization errors of these three algorithms averaged over all 100 simulations, and we observe similar results.

6.2. Deep Neural Networks

MSGD and its variants have been widely applied in training deep neural networks (Sutskever et al. 2013, Kingma and Ba 2014, Goodfellow et al. 2016, He et al. 2016) and have been implemented in popular deep learning libraries, such as Tensorflow (Abadi et al. 2016) and PyTorch (Paszke et al. 2019). In this section, we present several experiments to compare MSGD with VSGD in training a nine-layer Residual Net (ResNet-9; Page 2018) over CIFAR-10 and CIFAR-100 data sets for 10- and 100-class image classification tasks, respectively. Both data sets contain 60,000 images, in which 50,000 images are used for training, and the other 10,000 are used for testing. The network architecture of ResNet-9 is shown in Figure 2 and summarized in Table 1. All experiments are done in PyTorch with one NVIDIA RTX 2080-Ti graphics processing unit (GPU). For each experiment, we repeat 20 times with different random seeds and report the average and standard deviation.

Figure 1. Comparison Between SGD and MSGD (with and Without the Step Size Annealing (SSA) and Momentum Annealing (MA) in Phase III)

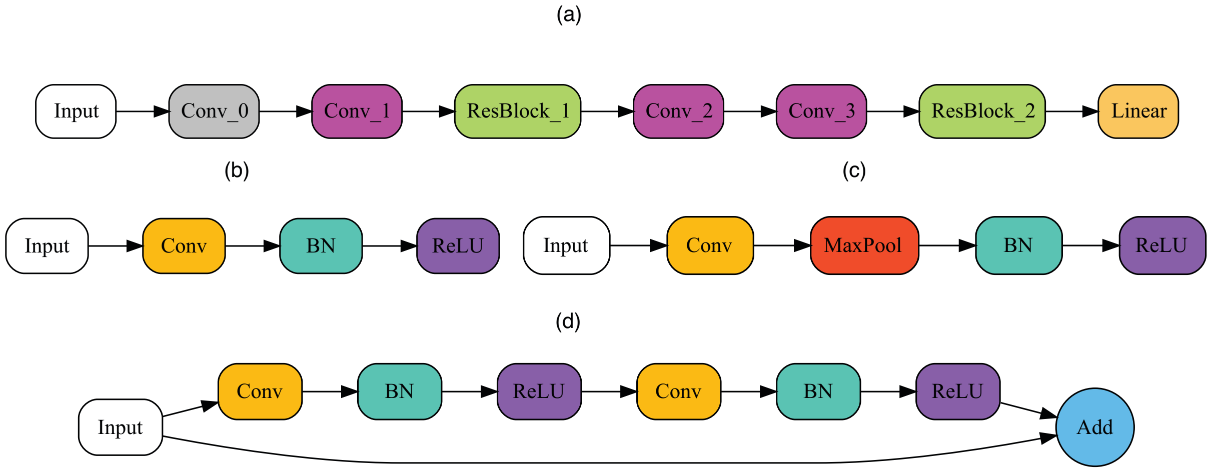


Notes. (a) Three phases in SGD. (b) MSGD does not converge. (c) MSGD with SSA converges. (d) MSGD with MA converges. (e) Comparison among SGD and different MSGDs.

We adopt the training configure from Page (2018), which uses the label smooth loss function (Szegedy et al. 2016). Specifically, for a K classification problem, given a training sample x with class y, we denote its predicted probability for class i as $p_i(x)$, and then, the loss function is

$$f(x, y; \epsilon) = (1 - \epsilon)\sum_{i=1}^{K} \delta_i(y) \log p_i(x) + \frac{\epsilon}{K}\sum_{i=1}^{K} \log p_i(x),$$

Figure 2. The Network Architecture of ResNet-9 and Its Detailed Components



Notes. (a) An illustrative visualization of nine layers in ResNet-9 (Residual block contains two layers). (b) Grey convolutional layer. (c) Pink convolutional layer. (d) Residual block containing two convolutional layers. ReLU, rectified linear unit; Conv, convolutional layer; BN, batch normalization.

where ϵ denotes the smoothing parameter and $\delta_i(y) = \mathbf{1}_{\{i=y\}}$ is the indicator function. In our experiments, we set ϵ as 0.2. In addition, for each experiment, we train the network for 100 epochs and use the batch size as 512. Moreover, we use the state-of-the-art step size setting with warm-up as follows:

$$\eta_i = \begin{cases} \dfrac{i}{20}\eta, & 1 \le i \le 20, \\ \left(1 - \dfrac{i-20}{80}\right)\eta, & 21 \le i \le 100, \end{cases}$$

where η_i is the step size used in the ith epoch for $1 \le i \le 100$. The warm-up is effective to obtain a good parameter in training deep neural network.

For MSGD, we set the momentum parameter μ as 0.9 and choose the step size η_{M} as $\{0.04\ell : \ell \in \mathbb{N}, 4 \le \ell \le 15\}$. Thus, for VSGD, we use the equivalent step size of MSGD $(\eta\ \mathrm{V} = \eta\ \mathrm{M}/(1-\mu))$ chosen from $\{0.4\ell : \ell \in \mathbb{N}, 4 \le \ell \le 15\}$. We then compare the loss values achieved by MSGD and VSGD under these settings over CIFAR 10 in Figure 3 and CIFAR 100 in Figure 4. As can be seen, the validate loss of MSGD decreases faster than that of the VSGD and eventually achieves a smaller value. For more comparison results, please see Online Companion Section 5. In addition, Table 2 presents the validate accuracy of both MSGD and VSGD over CIFAR data sets. As can be seen, on average, the MSGD is better than the VSGD with the equivalent step size over CIFAR-10 and CIFAR-100 tasks. We further test the significance of the pairwise comparison between the best MSGD and the best VSGD. For CIFAR-10 ($\eta_{\mathrm{M}} = 0.36$ and $\eta_{\mathrm{V}} = 2$) and CIFAR-100 ($\eta_{\mathrm{M}} = 0.56$ and $\eta_{\mathrm{V}} = 2.4$), the corresponding p-values are $\mathbf{0.0108}$ and $\mathbf{1.023 \times 10^{-5}}$, respectively. This shows that the best MSGD significantly outperforms the best VSGD.

7. Discussions

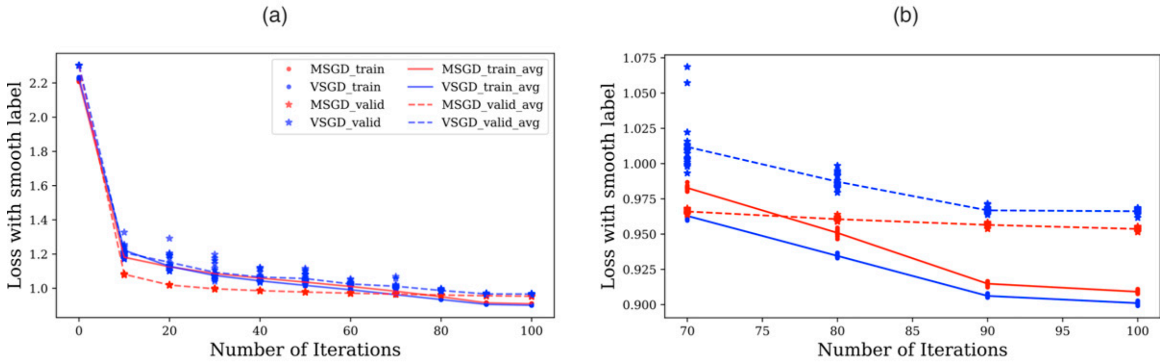
7.1. Related Literature

In the existing literature, we are only aware of Ghadimi and Lan (2016) and Jin et al. (2017) considering stochastic nonconvex optimization using momentum. Ghadimi and Lan (2016) only consider convergence to the first-order optimal solution and therefore, cannot justify the advantage of the momentum in escaping from saddle points;

Table 1. Network Architecture of ResNet-9

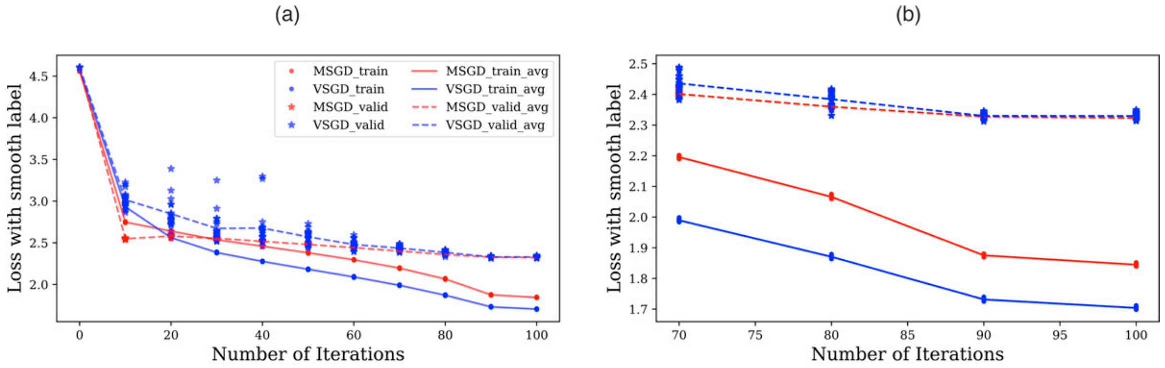
Layer	Output size	Filter, activation, and pooling
Convolutional	32 × 32	$[3 \times 3, 64] \times 1$, stride 1
Convolutional	16 × 16	$[3 \times 3, 128] \times 1$, stride 1, max pooling (2)
Residual block	16 × 16	$[3 \times 3, 128]$, stride 1
Convolutional	8 × 8	$[3 \times 3, 216] \times 1$, stride 1, max pooling (2)
Convolutional	4 × 4	$[3 \times 3, 512] \times 1$, stride 1, max pooling (2)
Residual block	4 × 4	$[3 \times 3, 512]$, stride 1
Linear	Number of classes	Max pooling (4), fully connected

Figure 3. Experimental Results of ResNet-9 on CIFAR-10 Under the Best Settings: $\eta_V = 2$, $\eta_M = 0.36$



Notes. (a) Best setting: $\eta_V = 2$ and $\eta_M = 0.36$. (b) Zoomed-in view for the best setting.

Figure 4. Experimental Results of ResNet-9 on CIFAR-100 Under the Best Settings: $\eta_V = 2.4$, $\eta_M = 0.56$



Notes. (a) Best setting: $\eta_V = 2.4$ and $\eta_M = 0.56$. (b) Zoomed-in view for the best setting.

Table 2. Results of Validation Accuracy and the Corresponding Standard Deviations (in Parentheses) for the Last Epoch Under the ResNet-9 over CIFAR-10 and CIFAR-100

$\frac{\eta}{1-\mu}$	1.6	2	2.4	2.8	3.2	3.6	4	4.4	4.8	5.2	5.6	6
CIFAR-10												
VSGD	95.31 (0.14)	**95.32** (0.14)	95.19 (0.23)	95.23 (0.19)	95.07 (0.22)	95.06 (0.20)	94.91 (0.25)	94.80 (0.34)	94.70 (0.20)	94.45 (0.30)	94.38 (0.26)	94.06 (0.51)
MSGD	95.65 (0.13)	95.71 (0.13)	95.78 (0.11)	95.81 (0.11)	95.83 (0.11)	**95.87** (0.14)	95.82 (0.08)	95.84 (0.11)	95.80 (0.14)	95.78 (0.15)	95.77 (0.12)	95.75 (0.10)
CIFAR-100												
VSGD	75.44 (0.39)	75.46 (0.42)	**75.49** (0.30)	75.21 (0.35)	75.10 (0.40)	74.81 (0.67)	74.73 (0.50)	74.45 (0.52)	74.18 (0.45)	73.83 (0.61)	73.47 (0.73)	73.12 (0.80)
MSGD	76.95 (0.21)	77.09 (0.26)	77.38 (0.25)	77.53 (0.25)	77.76 (0.25)	77.80 (0.23)	78.02 (0.17)	78.04 (0.25)	78.01 (0.26)	78.15 (0.28)	**78.17** (0.32)	78.16 (0.24)

Note. The best results among different settings are highlighted in bold.

Table 3. Comparison with Relevant Literature

	FOOS	SOOS	SA	SEA	Assumptions	A/N
Our study	√	√	√	√	Strict saddle, isolated optima	A
Ghadimi and Lan (2016)	√	×	√	×	LCG/LH/unconstrained	N
Jin et al. (2017)	√	√	×	√	LCG/LH/unconstrained	N

Note. A/N, asymptotic/nonasymptotic; FOOS, first-order optimal solution; LCG, Lipschitz continuous gradient; LH, Lipschitz continuous Hessian; SA, stochastic approximation; SEA, saddle escaping analysis; SOOS, second-order optimal solution.

Jin et al. (2017) only consider a batch algorithm, which cannot explain why the momentum hurts when MSGD converges to optima. Moreover, Jin et al. (2017) need an additional negative curvature exploitation procedure, which is not used in popular Nesterov's accelerated gradient algorithms. We summarize the comparison between our results and related works in Table 3.

Our analysis technique is closely related to several recent works using stochastic differential equations to study stochastic gradient-based methods. Li et al. (2017) adopt a numerical SDE approach to derive the so-called stochastic modified equations for VSGD. However, their analysis requires the drift term in the SDE to be bounded, which is not satisfied by MSGD. Other results consider SDE approximations of several accelerated SGD algorithms for convex smooth problems only (Krichene and Bartlett 2017, Wang 2017). In contrast, our analysis is for nonconvex problems, which are more general and more technically challenging.

In a broader sense, our work is also related to Matthews et al. (2018); Mei et al. (2018, 2019); Rotskoff and Vanden-Eijnden (2018); and Sirignano and Spiliopoulos (2018, 2019), which use weak convergence to prove the asymptotic approximation of extreme large neural networks. However, they consider that the size of the networks goes to infinity, whereas we consider the case that step size goes to zero.

7.2. Connected Local Optima

We want to remark that our analysis can be extended to handle connected global optima. As we have mentioned, the major difficulty is the unboundedness of the normalized error $(x_t - x^*)/\sqrt{\eta}$. This can be overcome by choosing a suitable metric to characterize the distance between the iterate and global optima. Take rank-r PCA as an example, where the rotation of any global optimum is also global optimal, and thus, all the global optima are connected. In this case, we can use the principal angle between column spans of a given global optimum and the iterate (Chen et al. 2018) to characterize the error. Because the principal angle is rotational invariant, the normalized error will be a unique quantity and will not blow up even when the iterate is wandering among different optima. Moreover, we can also utilize special landscape structure, such as partial dissipativity (Zhou et al. 2019), around the connected local optima to facilitate our analysis. However, the analysis will be more involved and is out of the scope of our paper.

7.3. Connection to Deep Neural Networks (DNNs)

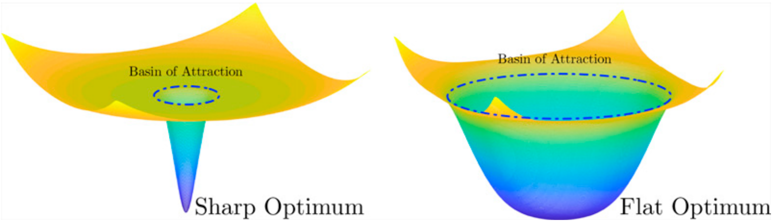
The results on training DNNs are expectable or partially expectable, given our theoretical analysis for streaming PCA. Our results show that with a good network architecture, the momentum indeed improves the training.

Our analysis implies that when η is sufficiently small, MSGD with step size η and momentum μ performs similarly to VSGD with step size $\eta/(1-\mu)$. In practice, however, people actually use a relative large step size during training, and we can still observe the advantage of MSGD over VSGD with the same equivalent step size. As we can observe in Table 2, MSGD always performs better than VSGD. Moreover, MSGD achieves the optimal generalization using $\eta_{\mathrm{M}}/(1-\mu) = 5.6$, but VSGD performs the best using a smaller equivalent step size $\eta_{\mathrm{V}} = 2.4 < 5.6$ under the ResNet over CIFAR-100. This implies MSGD can afford larger equivalent step size than VSGD. These phenomena cannot be fully explained by our theory.

7.4. Flat/Sharp Local Optima

Keskar et al. (2016), Neyshabur et al. (2017), and Zhang et al. (2017) suggest that the landscape of these spurious/bad local optima is usually sharp (i.e., their basins of attractions are small and steep). From this aspect, using a larger equivalent step size can help MSGD escape from spurious/bad local optima and stay in "flat/good local optima" because the higher variance of the noise introduced by the momentum encourages more exploration outside the small basin of attraction of sharp local optima (Figure 5).

Figure 5. Two Illustrative Examples of the Flat and Sharp Local Optima



Note. MSGD tends to avoid the sharp local optimum because its high variance encourages exploration.

7.5. Extension

Our theoretical analysis can be applied to study other problems related to momentum. For example, Liu et al. (2018) use the main technique of this paper to study an asynchronous MSGD with the focus on the trade-off between momentum and asynchrony. For another example, by analyzing the SDE around different local optima, we can theoretically characterize how momentum helps select flat optima.

Acknowledgments

The authors thank Prof. Shuangchi He and Prof. Le Chen for helpful suggestions. The follow-up work (Liu et al. 2018) has been published in the *Proceedings of the Thirty-Second Conference on Neural Information Processing Systems*. Liu et al. (2018) adopt analysis from this paper but study a different problem. Please refer to Section 7 for detailed discussion.

References

Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado GS, et al. (2016) Tensorflow: Large-scale machine learning on heterogeneous distributed systems. Preprint, submitted March 14, https://arxiv.org/abs/1603.04467v2.

Borkar VS (1997) Stochastic approximation with two time scales. *Systems Control Lett.* 29(5):291–294.

Borkar VS (2009) *Stochastic Approximation: A Dynamical Systems Viewpoint*, vol. 48 (Springer, Berlin).

Borkar VS, Meyn SP (2000) The ODE method for convergence of stochastic approximation and reinforcement learning. *SIAM J. Control Optim.* 38(2):447–469.

Chen Z, Yang FL, Li CJ, Zhao T (2017) Online multiview representation learning: Dropping convexity for better efficiency. Preprint, submitted February 27, https://arxiv.org/abs/1702.08134v1.

Chen M, Yang L, Wang M, Zhao T (2018) Dimensionality reduction for stationary time series via stochastic nonconvex optimization. Bengio S, Wallach H, Larochelle H, Grauman K, Cesa-Bianchi N, Garnett R, eds. *Adv. Neural Inform. Processing Systems* (Curran Associates, Inc., Montreal, Quebec, Canada), 3496–3506.

Choromanska A, Henaff M, Mathieu M, Arous GB, LeCun Y (2015) The loss surfaces of multilayer networks. Guy L, Vishwanathan SVN, eds. *Artificial Intelligence Statist.* (PMLR, California), 192–204.

Fu MC, ed. (2015) *Handbook of Simulation Optimization*, vol. 216 (Springer, Berlin).

Ge R, Lee JD, Ma T (2016) Matrix completion has no spurious local minimum. Lee D, Sugiyama M, Luxburg U, Guyon I, Garnett R, eds. *Adv. Neural Inform. Processing Systems* (Curran Associates, Inc., Barcelona, Spain), 2973–2981.

Ghadimi S, Lan G (2013) Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM J. Optim.* 23(4):2341–2368.

Ghadimi S, Lan G (2016) Accelerated gradient methods for nonconvex nonlinear and stochastic programming. *Math. Programming* 156(1-2): 59–99.

Goodfellow I, Bengio Y, Courville A, Bengio Y (2016) *Deep Learning*, vol. 1 (MIT Press, Cambridge, MA).

He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. *Proc. IEEE Conf. Comput. Vision Pattern Recognition* (IEEE, San Juan, PR), 770–778.

Jin C, Netrapalli P, Jordan MI (2017) Accelerated gradient descent escapes saddle points faster than gradient descent. Preprint, submitted November 28, https://arxiv.org/abs/1711.10456.

Keskar NS, Mudigere D, Nocedal J, Smelyanskiy M, Tang PTP (2016) On large-batch training for deep learning: Generalization gap and sharp minima. Preprint, submitted September 15, https://arxiv.org/abs/1609.04836.

Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. Preprint, submitted December 22, https://arxiv.org/abs/1412.6980.

Krichene W, Bartlett PL (2017) Acceleration and averaging in stochastic mirror descent dynamics. Preprint, submitted July 19, https://arxiv.org/abs/1707.06219.

Kushner HJ, Yin GG (2003) *Stochastic Approximation and Recursive Algorithms and Applications*, vol. 35 (Springer-Verlag, New York).

Li Q, Tai C, Weinan E (2017) Stochastic modified equations and adaptive stochastic gradient algorithms. Precup D, Whye Teh Y, eds. *Internat. Conf. Machine Learn.* (PMLR, Sydney, Australia), 2101–2110.

Li X, Wang Z, Lu J, Arora R, Haupt J, Liu H, Zhao T (2016) Symmetry, saddle points, and global geometry of nonconvex matrix factorization. Preprint, submitted December 29, https://arxiv.org/abs/1612.09296.

Liu T, Li S, Shi J, Zhou E, Zhao T (2018) Toward understanding acceleration tradeoff between momentum and asynchrony in distributed nonconvex stochastic optimization. Preprint, submitted June 4, https://arxiv.org/abs/1806.01660.

Matthews AGdG, Rowland M, Hron J, Turner RE, Ghahramani Z (2018) Gaussian process behaviour in wide deep neural networks. Preprint, submitted April 30, https://arxiv.org/abs/1804.11271.

Mei S, Misiakiewicz T, Montanari A (2019) Mean-field theory of two-layers neural networks: Dimension-free bounds and kernel limit. Preprint, submitted February 16, https://arxiv.org/abs/1902.06015.

Mei S, Montanari A, Nguyen PM (2018) A mean field view of the landscape of two-layer neural networks. *Proc. Natl. Acad. Sci. USA* 115(33): E7665–E7671.

Nesterov Y (1983) A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$. *Proc. USSR Acad. Sci.* 269: 543–547.

Newton D, Pasupathy R, Yousefian F (2018) Recent trends in stochastic gradient descent for machine learning and big data. Rabe M, Juan AA, Mustafee N, Skoogh A, Jain S, Johansson B, eds. *Proc. 2018 Winter Simulation Conf.* (IEEE Press, Gothenburg, Sweden), 366–380.

Neyshabur B, Bhojanapalli S, McAllester D, Srebro N (2017) Exploring generalization in deep learning. Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R, eds. *Adv. Neural Inform. Processing Systems* (Curran Associates, Inc., California), 5949–5958.

Øksendal B (2003) *Stochastic Differential Equations* (Springer, Berlin).

Page D (2018) How to train your ResNet. Accessed September 24, 2018, https://myrtle.ai/learn/how-to-train-your-resnet/.

Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, et al. (2019) Pytorch: An imperative style, high-performance deep learning library. Wallach H, Larochelle H, Beygelzimer A, d'Alché-Buc F, Fox E, Garnett R, eds. *Adv. Neural Inform. Processing Systems* (Curran Associates, Inc., Vancouver, BC, Canada), 8026–8037.

Polyak BT (1964) Some methods of speeding up the convergence of iteration methods. *USSR Comput. Math. Math. Phys.* 4(5):1–17.

Robbins H, Monro S (1951) A stochastic approximation method. *Ann. Math. Statist.* 22(3):400–407.

Rotskoff GM, Vanden-Eijnden E (2018) Neural networks as interacting particle systems: Asymptotic convexity of the loss landscape and universal scaling of the approximation error. Preprint, submitted May 2, https://arxiv.org/pdf/1805.00915v1.pdf.

Sirignano J, Spiliopoulos K (2018) Mean field analysis of neural networks. Preprint, submitted May 2, https://arxiv.org/abs/1805.01053.

Sirignano J, Spiliopoulos K (2019) Mean field analysis of deep neural networks. Preprint, submitted March 11, https://arxiv.org/abs/1903.04440.

Sun J, Qu Q, Wright J (2016) A geometric analysis of phase retrieval. *2016 IEEE Internat. Sympos. Inform. Theory (ISIT)* (IEEE), 2379–2383.

Sutskever I, Martens J, Dahl G, Hinton G (2013) On the importance of initialization and momentum in deep learning. Dasgupta S, McAllester D, eds. *Internat. Conf. Machine Learn.* (PMLR, Atlanta), 1139–1147.

Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. *Proc. IEEE Conf. Comput. Vision Pattern Recognition* (IEEE, San Juan, PR), 2818–2826.

Wang Y (2017) Asymptotic analysis via stochastic differential equations of gradient descent algorithms in statistical and computational paradigms. Preprint, submitted November 27, https://arxiv.org/abs/1711.09514.

Zhang C, Liao Q, Rakhlin A, Sridharan K, Miranda B, Golowich N, Poggio T (2017) Theory of deep learning III: Generalization properties of SGD. Technical report, Center for Brains, Minds and Machines, Massachusetts Institute of Technology, Cambridge, MA.

Zhou M, Liu T, Li Y, Lin D, Zhou E, Zhao T (2019) Toward understanding the importance of noise in training neural networks. Chaudhuri K, Salakhutdinov R, eds. *Internat. Conf. Machine Learn.* (PMLR, California), 7594–7602.