A Unified Analysis of Stochastic Momentum Methods for Deep Learning

Yan Yan[1,2], **Tianbao Yang**[3], **Zhe Li**[3], **Qihang Lin**[4], **Yi Yang**[1,2]
[1] SUSTech-UTS Joint Centre of CIS, Southern University of Science and Technology
[2] Centre for Artificial Intelligence, University of Technology Sydney
[3] Department of Computer Science, The University of Iowa
[4] Tippie College of Business, The University of Iowa
yan.yan-3@student.uts.edu.au, {tianbao-yang, zhe-li-1, qihang-lin}@uiowa.edu, yi.yang@uts.edu.au

Abstract

Stochastic momentum methods have been widely adopted in training deep neural networks. However, their theoretical analysis of convergence of the training objective and the generalization error for prediction is still under-explored. This paper aims to bridge the gap between practice and theory by analyzing the stochastic gradient (SG) method, and the stochastic momentum methods including two famous variants, *i.e.*, the stochastic heavy-ball (SHB) method and the stochastic variant of Nesterov's accelerated gradient (SNAG) method. We propose a framework that unifies the *three* variants. We then derive the convergence rates of the norm of gradient for the non-convex optimization problem, and analyze the generalization performance through the uniform stability approach. Particularly, the convergence analysis of the training objective exhibits that SHB and SNAG have no advantage over SG. However, the stability analysis shows that the momentum term can improve the stability of the learned model and hence improve the generalization performance. These theoretical insights verify the common wisdom and are also corroborated by our empirical analysis on deep learning.

1 Introduction

Momentum methods have a long history dating back to 1960's. Polyak (1964) proposed a heavy-ball (HB) method that uses the previous two iterates when computing the next one. The original motivation of momentum methods is to speed up the convergence for convex optimization. For a twice continuously differential strongly convex and smooth objective function, Polyak's analysis yields an accelerated linear convergence rate over the standard gradient method. In 1983, Nesterov (1983) proposed an accelerated gradient (NAG) method, which is also deemed as a momentum method and achieves the optimal $O(1/t^2)$ convergence rate for convex smooth optimization [1], which has a clear advantage over standard gradient method with $O(1/t)$ convergence for the same problem. It was later on shown to have an accelerated linear convergence rate for smooth and strongly convex optimization problems (Nesterov, 2004). Both the HB method and the NAG method use a *momentum* term in updating the solution, i.e., the difference between current iterate and the previous iterate. Therefore, both methods have been referred to as *momentum* methods in literature.

Due to recently increasing interests in deep learning, the stochastic variants of HB and NAG methods have been employed broadly in optimizing deep neural networks (Krizhevsky *et al.*, 2012; Hinton *et al.*, 2012). Sutskever *et al.* (2013) are probably the first to study SNAG and compare it with SHB for optimizing deep neural networks. Although they have some interesting findings of these two methods in deep learning (e.g., a distinct improvement in performance of SNAG is usually observed in their experiments), their analysis and argument are mostly restricted to convex problems. Moreover, some questions remained unanswered, e.g., (i) Do SHB and SNAG enjoy faster convergence rates than SG for deep learning (non-convex optimization problems) as in the convex and deterministic setting? (ii) If not, what is the advantage of these two methods over SG? (iii) Why does SNAG often yield improvement over SHB?

In this paper, we propose and analyze a unified framework for stochastic momentum methods and stochastic gradient method aiming at bringing answers and more insights to above questions. We summarize our results and contributions as follows:

- We propose a unified stochastic momentum framework parameterized by a free scalar parameter. The framework reduces to SHB, SNAG and SG by setting three different values for the free parameter.
- We present a unified convergence analysis of the gradient's norm of the training objective of these stochastic methods for non-convex optimization, revealing the same rate of convergence for three variants.
- We analyze the generalization error of the unified framework through the uniform stability approach. The result exhibits a clear advantage of stochastic momentum methods, i.e., adding a momentum helps generalization.
- Our empirical results for learning deep neural networks complete the unified view and analysis by showing that (i) there is no clear advantage of SHB and SNAG over

[1] t is the number of iterations.

SG in convergence speed of the training error; (ii) the advantage of SHB and SNAG lies at better generalization due to more stability; (iii) SNAG usually achieves the best tradeoff between speed of convergence in training error and stability of testing error among the three stochastic methods.

2 More Related Work

There is much analysis on the momentum methods for deterministic optimization. Nesterov pioneered the work of accelerated gradient methods for smooth convex optimization (Nesterov, 2004). The convergence analysis of HB has been recently extended to smooth functions for both convex (Ghadimi *et al.*, 2014; Ochs *et al.*, 2015) and non-convex deterministic optimization (Ochs *et al.*, 2014; Ochs, 2016). As the rising popularity of deep neural networks, the stochastic variants of HB and NAG have been employed widely for training neural networks and leading to tremendous success for many problems in computer vision and speech recognition (Krizhevsky *et al.*, 2012; Hinton *et al.*, 2012; Sutskever *et al.*, 2013). However, their *stochastic* variants in non-convex optimization are under-explored.

It is worth mentioning that two recent works have established the convergence results of the SG method (Ghadimi and Lan, 2013) and the stochastic version of a different variant of accelerated gradient method for non-convex optimization (Ghadimi and Lan, 2016). However, the variant of accelerated gradient method in (Ghadimi and Lan, 2016) is hard to be explained in the framework of momentum methods and is not widely employed for optimizing deep neural networks. Moreover, their analysis is not applicable to the SHB method. Hence, from a theoretical standpoint, it is still interesting to analyze the stochastic variants of the Nesterov's accelerated gradient method and the HB method for stochastic non-convex optimization, which are extensively employed for learning deep neural networks. Our unified analysis shows that they enjoy the same order of convergence rate as the SG method, which conincides with the results in (Ghadimi and Lan, 2013, 2016).

On the other hand, there exist few studies on analyzing the statistical properties (e.g., the generalization error) of the model learned by the SG method or stochastic momentum methods for minimizing the empirical risk. Conventional studies on the SG method in terms of statistical property focus on one pass learning, i.e., the training examples are passed once (Cesa-Bianchi *et al.*, 2004). Recently, there emerge several works that aim to establish the statistical properties of the multiple pass SG methods in machine learning (Lin and Rosasco, 2016; Hardt *et al.*, 2016). The latter work is closely related to the present work, which established the generalization error of the SG method with multiple pass for both convex and non-convex learning problems by analyzing the uniform stability. Nevertheless, it remains an open problem from a theoretical standpoint how the momentum term helps improve the generalization, though it has been observed to yield better performance in practice for deep learning (Sutskever *et al.*, 2013). Our unified analysis of the uniform stability of the SG method and stochastic momentum methods explicitly exhibit the advantage of the stochastic momentum methods in terms of the generalization error, hence providing the theoretical support for the common wisdom.

In the remainder of the paper, we first review the HB and NAG method, and present their stochastic variants. Then we present a unified view of these momentum methods. Next, we present the convergence and generalization analysis for stochastic momentum methods. In addition, we present empirical results for comparing different methods for optimizing deep neural networks. Finally, we conclude this work.

3 Momentum Methods And Their Stochastic Variants

3.1 Notations and Setup

Let us consider a general setting of learning with deep learning as a special case. Given a set of training examples $\mathcal{S} = (\mathbf{q}_1, \ldots, \mathbf{q}_n)$ sampled from an unknown distribution $\mathcal{D}$, the goal of learning is to find a model $\mathbf{x}$ that minimizes the population risk, i.e.,

$$\min_{\mathbf{x}\in\Omega} F(\mathbf{x}) \triangleq \mathrm{E}_{\mathbf{q}\sim\mathcal{D}}[\ell(\mathbf{x}, \mathbf{q})], \tag{1}$$

where ℓ is a loss function, $\ell(\mathbf{x}, \mathbf{q})$ denotes the loss of the model $\mathbf{x}$ on the example $\mathbf{q}$ and Ω denotes the hypothesis class of the model. Since we cannot compute $F(\mathbf{x})$ due to unknown distribution $\mathcal{D}$, one usually learns a model by minimizing the empirical risk, i.e.,

$$\min_{\mathbf{x}\in\Omega} f(\mathbf{x}) \triangleq \frac{1}{n}\sum_{i=1}^{n} \ell(\mathbf{x}, \mathbf{q}_i). \tag{2}$$

Two concerns usually present in the above empirical risk minimization approach. First, how fast the optimization algorithm solves Problem (2). This is usually measured by the speed of convergence to the optimal solution. However, it is NP-hard to find the global optimal solution for a general non-convex optimization problem (Hillar and Lim, 2013). As with many previous works (Ghadimi and Lan, 2013, 2016; Reddi *et al.*, 2016; Zhu and Hazan, 2016), we study the convergence rate of an iterative algorithm to the critical point, i.e., a point $\mathbf{x}_*$ such that $\nabla f(\mathbf{x}_*) = 0$.

Second, how the model learned by solving Problem (2) generalizes to different data. It is usually measured by the population risk $F(\widehat{\mathbf{x}})$ defined in (1). Since the model $\widehat{\mathbf{x}}$ is learned from the random samples $\mathbf{q}_1, \ldots, \mathbf{q}_n$ with randomness in the optimization algorithm itself, the expected population risk $\mathrm{E}[F(\widehat{\mathbf{x}})]$ is also used for the analysis with the expectation taking over the randomness in the samples and the algorithm itself. One way to assess the expected population risk is the generalization error, i.e., the difference between the population risk and the empirical risk,

$$\epsilon_{\text{gen}} \triangleq \mathrm{E}[F(\widehat{\mathbf{x}}) - f(\widehat{\mathbf{x}})]. \tag{3}$$

We use $\nabla h(\mathbf{x})$ to denote the gradient of a smooth function. A function is smooth iff there exists $L > 0$ such that

$$\|\nabla h(\mathbf{y}) - \nabla h(\mathbf{x})\| \leq L\|\mathbf{y} - \mathbf{x}\|, \quad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d, \tag{4}$$

where $\|\cdot\|$ denotes the Euclidean norm. Note that the above inequality does not imply convexity. Through the paper, we assume that $\ell(\mathbf{x}, \mathbf{q})$ a G-Lipschitz continuous and L-smooth non-convex function in $\mathbf{x}$, and assume that $\Omega = \mathbb{R}^d$. It follows that $f(\mathbf{x})$ is G-Lipschitz continuous and L-smoothness.

3.2 Stochastic Momentum Methods

We denote by $\mathcal{G}_k = \mathcal{G}(\mathbf{x}_k; \xi_k)$ a stochastic gradient of $f(\mathbf{x})$ at $\mathbf{x}_k$ depending on a random variable ξ_k such that $\mathrm{E}[\mathcal{G}(\mathbf{x}_k; \xi_k)] = \nabla f(\mathbf{x}_k)$. In the context of the empirical risk minimization (2), $\mathcal{G}(\mathbf{x}_k; \xi_k) = \nabla \ell(\mathbf{x}_k; \mathbf{q}_{i_k})$, where i_k is a random index sampled from $\{1, \ldots, n\}$.

There are two variants of momentum methods for solving (2), i.e., HB and NAG. HB was originally proposed for optimizing a smooth and strongly convex objective function. Based on HB, the update of stochastic HB (SHB) is given below for $k = 0, \ldots,$

$$\text{SHB:} \quad \mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \mathcal{G}(\mathbf{x}_k; \xi_k) + \beta(\mathbf{x}_k - \mathbf{x}_{k-1}), \quad (5)$$

with $\mathbf{x}_{-1} = \mathbf{x}_0$, where $\beta \in [0, 1)$ is the momentum constant and α is the step size. Equivalently, the above update can be implemented by the following two steps for $k = 0, \ldots,$:

$$\text{SHB:} \quad \begin{cases} \mathbf{v}_{k+1} = \beta \mathbf{v}_k - \alpha \mathcal{G}(\mathbf{x}_k; \xi_k) \\ \mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{v}_{k+1}. \end{cases} \quad (6)$$

Based on NAG (Nesterov, 2004), the update of stochastic NAG (SNAG) consists of the two steps below for $k = 0, \ldots,$:

$$\text{SNAG:} \quad \begin{cases} \mathbf{y}_{k+1} = \mathbf{x}_k - \alpha \mathcal{G}(\mathbf{x}_k; \xi_k) \\ \mathbf{x}_{k+1} = \mathbf{y}_{k+1} + \beta(\mathbf{y}_{k+1} - \mathbf{y}_k), \end{cases} \quad (7)$$

with $\mathbf{y}_0 = \mathbf{x}_0$. By introducing $\mathbf{v}_k = \mathbf{y}_k - \mathbf{y}_{k-1}$ with $\mathbf{v}_0 = 0$, the above update can be equivalently written as

$$\text{SNAG:} \quad \begin{cases} \mathbf{v}_{k+1} = \beta \mathbf{v}_k - \alpha \mathcal{G}(\mathbf{y}_k + \beta \mathbf{v}_k; \xi_k) \\ \mathbf{y}_{k+1} = \mathbf{y}_k + \mathbf{v}_{k+1}. \end{cases} \quad (8)$$

Finally, the traditional view of SG can be written as

$$\text{SG:} \quad \mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \mathcal{G}(\mathbf{x}_k; \xi_k). \quad (9)$$

By comparing (8) to (6), one might argue that the difference between HB and NAG lies at the point for evaluating the gradient (Sutskever *et al.*, 2013). We will present a different unified view of the three methods that allows us to analyze them in a unified framework. The convergence of HB and NAG has been established for convex optimization (Polyak, 1964; Nesterov, 1983, 2004; Ghadimi *et al.*, 2014; Ochs *et al.*, 2015).

4 A Unified View of Stochastic Momentum Methods

In this section, we present a unified view of the two (stochastic) momentum methods and (stochastic) gradient methods. We first present the unified framework and then show that HB, NAG and the gradient method are special cases of the unified framework. Denote by $\mathcal{G}(\mathbf{x}_k)$ either a gradient or a stochastic gradient of $f(\mathbf{x})$ at $\mathbf{x}_k$.

Let $\alpha > 0$, $\beta \in [0, 1)$, and $s \geq 0$. The updates of the stochastic unified momentum (SUM) method are given by

$$\text{SUM :} \quad \begin{cases} \mathbf{y}_{k+1} = \mathbf{x}_k - \alpha \mathcal{G}(\mathbf{x}_k) \\ \mathbf{y}^s_{k+1} = \mathbf{x}_k - s\alpha \mathcal{G}(\mathbf{x}_k) \\ \mathbf{x}_{k+1} = \mathbf{y}_{k+1} + \beta(\mathbf{y}^s_{k+1} - \mathbf{y}^s_k), \end{cases} \quad (10)$$

for $k \geq 0$ with $\mathbf{y}^s_0 = \mathbf{x}_0$. It is notable that in the update of $\mathbf{x}_{k+1}$, a momentum term is constructed based on the auxiliary sequence $\{\mathbf{y}^s_k\}$, whose update is parameterized by s. The following proposition indicates that SUM reduces to the concerned three special cases by setting different values to s.

Proposition 1. *SUM (10) reduces to the three variants SG (9), SHB (5) and SNAG (7) by setting $s = \frac{1}{1-\beta}$, $s = 0$ and $s = 1$, respectively. Particularly, the update of SG is $\mathbf{x}_{k+1} = \mathbf{x}_k - \frac{\alpha}{1-\beta}\mathcal{G}(\mathbf{x}_k, \xi_k)$, where the step size is $\frac{\alpha}{1-\beta}$.*

From the above result, we can see that SHB, SNAG and SG are three variants of SUM. Moreover, the SUM view of SG implies that SG can have a larger "effective" step size (i.e., $\alpha/(1-\beta)$) before the gradient $\mathcal{G}(\mathbf{x}_k)$ than that of SHB and SNAG. We note that this is a very important observation about SG since setting a smaller effective step size for SG (e.g., the same as that in SNAG) will yield much worse convergence of training error as observed in experiments.

To facilitate the unified analysis of the stochastic momentum methods, we note that (10) implies the recursions in (13) and (14) given in the following lemma.

Lemma 1. *Let $\mathbf{p}_k$ be*

$$\mathbf{p}_k = \begin{cases} \dfrac{\beta}{1-\beta}(\mathbf{x}_k - \mathbf{x}_{k-1} + s\alpha \mathcal{G}(\mathbf{x}_{k-1})), \ k \geq 1 \\ 0, \quad k = 0 \end{cases} \quad (11)$$

and

$$\mathbf{v}_k = \frac{(1-\beta)}{\beta}\mathbf{p}_k. \quad (12)$$

Then for any $k \geq 0$, we have

$$\mathbf{x}_{k+1} + \mathbf{p}_{k+1} = \mathbf{x}_k + \mathbf{p}_k - \frac{\alpha}{1-\beta}\mathcal{G}(\mathbf{x}_k), \quad (13)$$

$$\mathbf{v}_{k+1} = \beta \mathbf{v}_k + ((1-\beta)s - 1)\alpha \mathcal{G}(\mathbf{x}_k). \quad (14)$$

Remark: We note that a similar recursion in (13) with $s = 0$ and $s = 1$ has been observed and employed to (Ghadimi *et al.*, 2014) for deterministic convex optimization. However, the recursion in (14) for $\mathbf{v}_k$ (i.e., $\mathbf{p}_k$) is a key to our convergence analysis for non-convex optimization and importantly the generalization to any s allows us to analyze SHB, SNAG and SG in a unified framework.

Finally, we present a lemma stating the cumulative effect of updates for each iterate, which will be useful for our generalization error analysis.

Lemma 2. *Given the update in (10), for any $k \geq 0$ we have*

$$\mathbf{x}_{k+1} = \mathbf{x}_0 - \sum_{\tau=0}^{k} \left\{ \frac{1}{1-\beta} - \beta^{k-\tau+1} \frac{1 - s(1-\beta)}{1-\beta} \right\} \alpha \mathcal{G}(\mathbf{x}_\tau). \quad (15)$$

Remark: The above cumulative update reduce to the following three cases for SHB ($s = 0$), SNAG ($s = 1$) and SG ($s = 1/(1-\beta)$).

$$\mathbf{x}_{k+1} = \begin{cases} \mathbf{x}_0 - \sum_{t=0}^{k} \left\{ \frac{1}{1-\beta} - \frac{\beta^{k-\tau+1}}{1-\beta} \right\} \alpha \mathcal{G}(\mathbf{x}_\tau) & s = 0 \\ \mathbf{x}_0 - \sum_{t=0}^{k} \left\{ \frac{1}{1-\beta} - \frac{\beta^{k-\tau+2}}{1-\beta} \right\} \alpha \mathcal{G}(\mathbf{x}_\tau) & s = 1 \\ \mathbf{x}_0 - \sum_{t=0}^{k} \frac{1}{1-\beta} \alpha \mathcal{G}(\mathbf{x}_\tau) & s = \frac{1}{1-\beta} \end{cases}$$

From the above cumulative update, we can see that SHB and SNAG have smaller step size for each stochastic gradient. This is the main reason that SHB and SNAG are more stable than SG, and hence yield a solution with better generalization performance. We will present a more formal analysis of generalization error later.

5 Convergence Analysis of SUM

In this section, we present the convergence results for the empirical risk minimization (2) of the SUM methods. As mentioned before, for deep learning problems the loss function $\ell(\mathbf{x}, \mathbf{q})$ is a non-convex function, which makes finding the global optimal solution an NP-hard problem. Instead, as in many previous works we will present the convergence rates of SUM in terms of the norm of the gradient. We will present the main results first and then sketch the analysis. Detailed proofs are deferred to the supplement due to limit of space.

5.1 Main results

Theorem 1. *(Convergence of SUM) Suppose $f(\mathbf{x})$ is a non-convex and L-smooth function, $\mathrm{E}[\|\mathcal{G}(\mathbf{x};\xi) - \nabla f(\mathbf{x})\|^2] \leq \sigma^2$ and $\|\nabla f(\mathbf{x})\| \leq G$ for any $\mathbf{x}$. Let update (10) run for t iterations with $\mathcal{G}(\mathbf{x}_k;\xi_k)$. By setting $\alpha = \min\{\frac{1-\beta}{2L}, \frac{C}{\sqrt{t+1}}\}$ we have*

$$\begin{aligned} &\min_{k=0,\ldots,t} \mathrm{E}[\|\nabla f(\mathbf{x}_k)\|^2] \\ &\leq \frac{2(f(\mathbf{x}_0) - f_*)(1-\beta)}{t+1} \max\left\{\frac{2L}{1-\beta}, \frac{\sqrt{t+1}}{C}\right\} \\ &+ \frac{C}{\sqrt{t+1}} \frac{L\beta^2((1-\beta)s-1)^2(G^2+\sigma^2) + L\sigma^2(1-\beta)^2}{(1-\beta)^3} \end{aligned}$$

Remark: We would like to make several remarks. (i) The assumption on the magnitude of the gradient and the variance of stochastic gradient can be simply replaced by the magnitude of the stochastic gradient, which are standard assumptions in the previous analysis of stochastic gradient method (Ghadimi and Lan, 2013). (ii) This is the first time that the convergence rate of SHB for non-convex optimization is established. A similar convergence rate of SG and a different stochastic variant of accelerated gradient method has been established in (Ghadimi and Lan, 2013) and (Ghadimi and Lan, 2016), respectively under similar assumptions. (iii) The unified convergence makes it clear that the difference of the convergence bounds for different variants of SUM lies at the term $((1-\beta)s-1)^2$, which is equal to β^2, β^4 and 0 for SHB, SNAG and SG, respectively. (iv) The step size α of different variants of SUM used in the analysis of Theorem 1 is the same value.

The above result shows that the convergence upper bound of the three methods are of the same order, i.e., $O(1/\sqrt{t})$ for the gradient's square norm. In addition, when the momentum term β is large, the effect of different values of s in the term would $L\beta^2((1-\beta)s-1)^2(G^2+\sigma^2)$ becomes marginal in contrast to the term $L\sigma^2(1-\beta)^2$ in the convergence bound. This reveals that SHB and SNAG have no advantage over SG in terms of empirical risk minimization.

Below, we present a result with different step sizes α for different variants of SUM in the analysis, which sheds more insights of different methods.

Theorem 2. *(Convergence of SUM) Suppose $f(\mathbf{x})$ is a non-convex and L-smooth function, $\mathrm{E}[\|\mathcal{G}(\mathbf{x};\xi) - \nabla f(\mathbf{x})\|^2] \leq \sigma^2$ and $\|\nabla f(\mathbf{x})\| \leq G$ for any $\mathbf{x}$. Let update (10) run for t iterations with $\mathcal{G}(\mathbf{x}_k;\xi_k)$. By setting $\alpha = \min\{\frac{1-\beta}{2L[1+((1-\beta)s-1)^2]}, \frac{C}{\sqrt{t+1}}\}$ we have*

$$\begin{aligned} &\min_{k=0,\ldots,t} \mathrm{E}[\|\nabla f(\mathbf{x}_k)\|^2] \leq \frac{2(f(\mathbf{x}_0) - f_*)(1-\beta)}{t+1}\Lambda \\ &+ \frac{C}{\sqrt{t+1}} \frac{L\beta^2(G^2+\sigma^2) + L\sigma^2(1-\beta)^2}{(1-\beta)^3} \end{aligned}$$

where $\Lambda = \max\left\{\frac{2L[1+((1-\beta)s-1)^2]}{1-\beta}, \frac{\sqrt{t+1}}{C}\right\}$.

Remark: The above result allows us to possibly set a larger initial value of α for SG (where $s = 1/(1-\beta)$) and SNAG (where $s = 1$) than that for SHB (where $s = 0$). Our empirical studies for deep learning also confirms this point.

5.2 Generalization Error Analysis of SUM

In this section, we provide a unified analysis for the generalization error of the solution returned by SUM after a finite number of iterations. By employing the unified analysis, we are able to analyze the effect of the scalar s on the generalization error. Our analysis is inspired by (Hardt *et al.*, 2016), which leverages the uniform stability of a randomized algorithm (Bousquet and Elisseeff, 2002) to bound the generalization error of multiple pass SG method. To this end, we first introduce the uniform stability and its connection with generalization error.

Let $\mathcal{A} : \mathcal{S} \to \mathbb{R}^d$ denote a randomized algorithm that generates a model $\mathcal{A}(\mathcal{S})$ from the set of training samples $\mathcal{S}$ of size n. The uniform stability measures that how likely the prediction of the learned model on any sample $\mathbf{q}$ would change if one example in $\mathcal{S}$ is changed to a different data. In particular, let $\mathcal{S}'$ denote a set of training examples that differ from $\mathcal{S}$ in one example. The algorithm $\mathcal{A}$ is said to be ϵ-uniform stable, if

$$\epsilon(\mathcal{A}, n) \triangleq \sup_{\mathcal{S},\mathcal{S}'} \sup_{\mathbf{q}} \mathrm{E}_{\mathcal{A}}[\ell(\mathcal{A}(\mathcal{S}), \mathbf{q}) - \ell(\mathcal{A}(\mathcal{S}'), \mathbf{q})] \leq \epsilon$$

The following proposition states that the generalization error of $\mathcal{A}(\mathcal{S})$ is bounded by the uniform stability of $\mathcal{A}$.

Proposition 2. *(Theorem 2.2 in Hardt* et al. *(2016)) For a randomized algorithm $A : \mathcal{S} \to \mathbb{R}$,*

$$\mathrm{E}[F(\mathcal{A}(\mathcal{S})) - f(\mathcal{A}(\mathcal{S}))] \leq \epsilon(\mathcal{A}, n) \tag{16}$$

The above proposition allows us to use the uniform stability of a randomized algorithm as a proxy of the generalization error. Below, we will show that SHB and SNAG are more uniform stable than SG, which exhibits that SHB and SNAG has potentially smaller generalization error than SG.

To proceed, we assume that loss function is G-Lipschitz continuous, then $|\ell(\mathcal{A}(\mathcal{S}), \mathbf{q}) - \ell(\mathcal{A}(\mathcal{S}'), \mathbf{q})| \leq G\|\mathcal{A}(\mathcal{S}) - \mathcal{A}(\mathcal{S}')\|$ and $\epsilon(\mathcal{A}, n) \leq \sup_{\mathcal{S},\mathcal{S}'} \mathrm{E}[\|\mathcal{A}(\mathcal{S}) - \mathcal{A}(\mathcal{S}')\|]$. To analyze the uniform stability of SUM, we will assume that there are two instances of SUM starting from the same initial solution, with one running on $\mathcal{S}$ and the other one running on $\mathcal{S}'$, where $\mathcal{S}$ and $\mathcal{S}'$ differs only at one example. Let $\mathbf{x}_t = \mathbf{x}_t(\mathcal{S})$ denote the t-th iterate of the first instance and $\mathbf{x}'_t = \mathbf{x}_t(\mathcal{S}')$ denote the t-th iterate of the second instance. Below, we establish a result showing how $\Delta_t = \mathrm{E}[\|\mathbf{x}_t - \mathbf{x}'_t\|]$ grows based on the unified framework in Lemma 2.

Proposition 3. *Assume that $\|\nabla \ell(\mathbf{x}, \mathbf{q})\|_2 \leq G$ for any $\mathbf{x}$ and $\mathbf{q}$ and $\ell(\mathbf{x}, \mathbf{q})$ is L-smooth w.r.t $\mathbf{x}$. For two data sets $\mathcal{S}, \mathcal{S}'$ that differs at one example, let $\mathbf{x}_t$ and $\mathbf{x}'_t$ denote the t-th iterates of running SUM for the empirical risk minimization on S and S', we have*

$$\Delta_{t+1} \leq \sum_{k=0}^{t} \frac{2\alpha G}{n} \eta_k^t + \left(1 - \frac{1}{n}\right) \sum_{k=0}^{t} \alpha L \eta_k^t \Delta_k$$

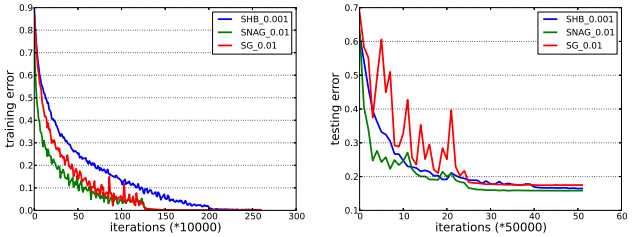
with $\Delta_0 = 0$, where $\eta_k^t = \frac{1}{1-\beta} - \beta^{t-k+1} \frac{1-s(1-\beta)}{1-\beta}$.

Remark: From the above result, we can easily analyze how the value of s affects the growth of Δ_t that implies the growth of the generalization error of $\mathbf{x}_t$. The values of η_k^t for the three variants (i.e., SHB, SNAG and SG) are given by $\eta_k^t(\text{SHB}) = \frac{1}{1-\beta} - \frac{\beta^{t-k+1}}{1-\beta}$, $\eta_k^t(\text{SNAG}) = \frac{1}{1-\beta} - \frac{\beta^{t-k+2}}{1-\beta}$ and $\eta_k^t(\text{SG}) = \frac{1}{1-\beta}$, respectively. It is obvious that $\eta_k^t(\text{SHB}) < \eta_k^t(\text{SNAG}) < \eta_k^t(\text{SG})$. As a result, Δ_t of SG grows faster than that of SNAG, and then followed by Δ_t of SHB. Since the generalization error of $\mathbf{x}_t$ is bounded by Δ_t up to a constant, we can conclude that by running the same number of iterations, the generalization error of the model returned by SHB and SNAG is potentially smaller than that of SG.

More Discussion. So far, we have analyzed the convergence rate for optimizing the empirical risk and the generalization error of the learned models of different variants of SUM, which provide answers to the questions raised at the beginning except the last one (why is SNAG observed to yield improvement on the prediction performance over SHB by some studies (Sutskever *et al.*, 2013)). Next, we show that how our analysis can shed lights on this question. In fact, the population risk of $\mathbf{x}_k$ that is usually assessed in practice by the testing error can be decomposed into three parts, consisting of the optimization error, the generalization error, and an optimization independent term, i.e.,

$$\mathrm{E}[F(\mathbf{x}_t)] = \mathrm{E}[f(\mathbf{x}_*)] + \underbrace{\mathrm{E}[F(\mathbf{x}_t) - f(\mathbf{x}_t)]}_{\text{gen}} + \underbrace{\mathrm{E}[f(\mathbf{x}_t) - f(\mathbf{x}_*)]}_{\text{opt}}$$

where $\mathbf{x}_*$ is the optimal solution to the empirical risk minimization problem. An informal analysis follows: our Theorem 2 implies that SNAG converges potentially faster than SHB in terms of the optimization error, while Proposition 3 implies that SHB has potentially smaller generalization error. If the optimization error of SNAG decreases faster than the generalization error increases comparing with SHB, then SNAG could yield a solution with a smaller population risk. However, a rigorous analysis of the optimization error is complicated by the non-convexity of the problem. In next section, we will present empirical results to corroborate and complete our theoretical analysis.



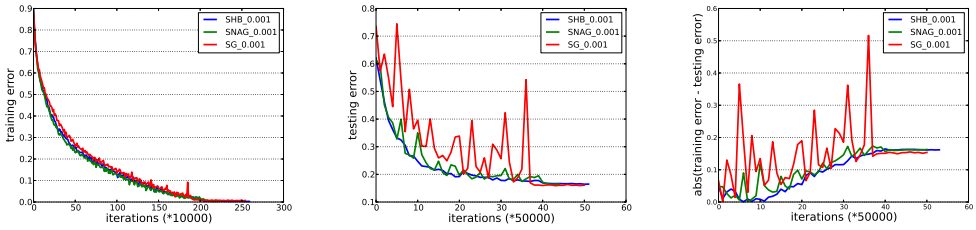
(a) training (b) testing

Figure 1: Training and testing error of different methods on CIFAR-10 with the best initial step size α. The result is consistent with our convergence result in Theorem 2.

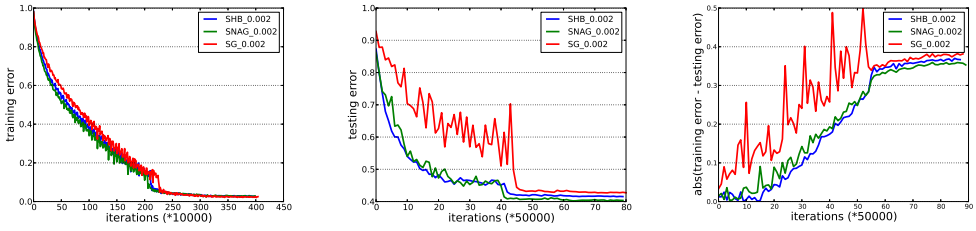
6 Empirical Studies

In this section, we present empirical results on the non-convex optimization of deep neural networks. We train a deep convolutional neural network (CNN) for classification on two benchmark datasets, i.e., CIFAR-10 and CIFAR-100. Both datasets contain $50,000$ training images of size $32 * 32$ from 10 classes (CIFAR-10) or 100 classes (CIFAR-100) and $10,000$ testing images of the same size. The employed CNN consists of 3 convolutional layers and 2 fully-connected layers. Each convolutional layer is followed by a max pooling layer. The output of the last fully-connected layer is fed into a 10-class or 100-class softmax loss function. We emphasize that we do not intend to obtain the state-of-the-art prediction performance by trying different network structures and different engineering tricks, but instead focus our attention on verifying the theoretical analysis. We compare the three variants of SUM, i.e., SHB, SNAG, and SG, which corresponds to $s = 0$, $s = 1$ and $s = 1/(1-\beta)$ in (10). We fix the momentum constant $\beta = 0.9$ and the regularization parameter of weights to 0.0005. We use a mini-batch of size 128 to compute a stochastic gradient at each iteration. All three methods use the same initialization. We follow the procedure in (Krizhevsky *et al.*, 2012) to set the step size α, i.e., initially giving a relatively large step size and and decreasing the step size by 10 times after certain number of iterations when observing the performance on testing data saturates.

Results on CIFAR-10. We first present the convergence results of different methods with the best initial step size. In particular, for the initial step size, we search in a range ($\{0.001, 0.002, 0.005, 0.01, 0.02, 0.05\}$) for different methods and select the best one that yields the fastest convergence in training error. In particular, for SHB the best initial step size is 0.001 and that for SNAG and SG is 0.01. In fact, a

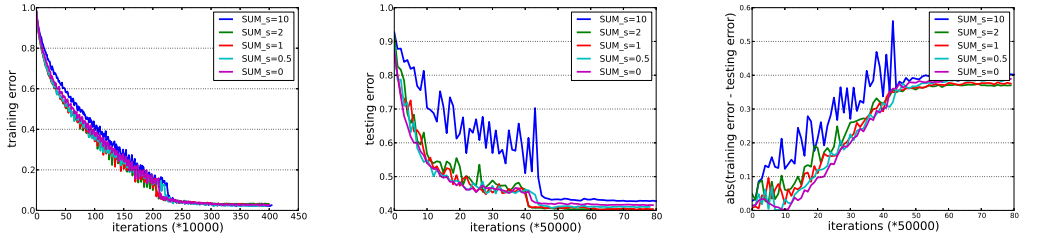


(a) training error on CIFA-10 (b) testing error on CIFA-10 (c) abs difference on CIFAR-10



(d) training error on CIFA-100 (e) testing error on CIFA-100 (f) abs difference on CIFAR-100

Figure 2: Training error, testing error and their absolute difference (i.e., $|\text{training error} - \text{testing error}|$) on CIFAR-10 and CIFAR-100 of three SUM variants. The numbers in legends indicate the initial step size α.



(a) training error (b) testing error (c) abs(training error - testing error)

Figure 3: Training and testing error and their absolute difference on CIFAR-100 of SUM with different s.

larger initial step size (e.g, 0.002) for SHB gives a divergent result. The training and testing error of different methods versus the number of iterations is plotted in Figure 1. This result is consistent with our convergence result in Theorem 2.

Next, we plot the performance of different methods with the same initial step size 0.001 in Figure 2. We report the training error, the testing error and their absolute difference in Figure 2(a), 2(b) and 2(c), respectively. We use the absolute difference between the training and testing error as an estimate of the generalization error. We can see that the convergence of training error of the three methods are very close, which is consistent with our theoretical result in Theorem 1. Moreover, the behavior of the absolute difference between the training and testing error is also consistent with the theoretical result in Proposition 3, i.e, SG has a larger generalization error than SNAG and SHB.

Results on CIFAR-100. We plot the training and testing error and their absolute difference of the three methods with the same initial step sizes (0.002) in Figure 2(d), 2(e) and 2(f), respectively. We observe similar trends in the training error and the generalization error, i.e., the three methods have similar performance (convergence speed) in training error but exhibit different degree of generalization error. The testing error curve shows that SNAG achieves the best prediction performance on the testing data.

Finally, we present a comparison of SUM with different values of s including variants besides SHB, SNAG and SG. In particular, we compare SUM with $s \in \{0, 0.5, 1, 2, 10\}$ and the same initial step size 0.002. Note that $s = 0$ corresponds to SHB, $s = 1$ corresponds to SNAG, $s = 10$ corresponds to SG since $\beta = 0.9$ and $s = 2$ corresponds to a new variant. The results on the CIFAR-100 data are shown in Figure 3. From the results, we can observe that the convergence of training error for different variants perform similarly. For the generalization error, we observe a clear trend from $s = 10$ to $s = 0$ in that the generalization error decreases.

7 Conclusion

We have developed a unified framework of stochastic momentum methods that subsumes SHB, SNAG and SG as spe-

cial cases, which have been widely adopted in training deep neural network. We also analyzed convergence of the training for non-convex optimization and generalization error for learning of the unified stochastic momentum methods. The unified framework and analysis bring more insights about differences between different methods and help explain experimental results for optimization deep neural networks. In particular, the momentum term helps improve the generalization performance but not helps speed up the training process.

Acknowledgements

This work is partially supported by NSF-1545995, the Data to Decisions Cooperative Research Centre and ARC DP180100106. Most work of Y. Yan was done when he was visiting the University of Iowa.

References

Olivier Bousquet and André Elisseeff. Stability and generalization. *J. Mach. Learn. Res.*, 2:499–526, 2002.

Nicolò Cesa-Bianchi, Alex Conconi, and Claudio Gentile. On the generalization ability of on-line learning algorithms. *IEEE Trans. on Inf. Theory*, 50(9), 2004.

Saeed Ghadimi and Guanghui Lan. Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

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

Euhanna Ghadimi, Hamid Reza Feyzmahdavian, and Mikael Johansson. Global convergence of the heavy-ball method for convex optimization. *coRR*, 2014.

Moritz Hardt, Benjamin Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *Proceedings of the International Conference on Machine Learning*, 2016.

Christopher J. Hillar and Lek-Heng Lim. Most tensor problems are np-hard. *J. ACM*, 60(6):45:1–45:39, 2013.

Geoffrey Hinton, Li Deng, Dong Yu, George E. Dahl, Abdelrahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N. Sainath, and Brian Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE*, 29:82–97, 2012.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

Junhong Lin and Lorenzo Rosasco. Optimal learning for multi-pass stochastic gradient methods. In *NIPS*, pages 4556–4564, 2016.

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

Yurii Nesterov. *Introductory lectures on convex optimization : a basic course*. Applied optimization. Kluwer Academic Publ., 2004.

Peter Ochs, Yunjin Chen, Thomas Brox, and Thomas Pock. ipiano: Inertial proximal algorithm for nonconvex optimization. *SIAM J. Imaging Sciences*, 7(2):1388–1419, 2014.

Peter Ochs, Thomas Brox, and Thomas Pock. ipiasco: Inertial proximal algorithm for strongly convex optimization. *Journal of Mathematical Imaging and Vision*, 53(2):171–181, 2015.

Peter Ochs. Local convergence of the heavy-ball method and ipiano for non-convex optimization. *CoRR*, abs/1606.09070, 2016.

B. T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4:791–803, 1964.

Sashank J. Reddi, Ahmed Hefny, Suvrit Sra, Barnabás Póczos, and Alexander J. Smola. Stochastic variance reduction for nonconvex optimization. *CoRR*, abs/1603.06160, 2016.

Ilya Sutskever, James Martens, George E. Dahl, and Geoffrey E. Hinton. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pages 1139–1147, 2013.

Zeyuan Allen Zhu and Elad Hazan. Variance reduction for faster non-convex optimization. *CoRR*, abs/1603.05643, 2016.

A Proof of Proposition 1

We re-present Proposition 1.

Proposition 1. *SUM (10) reduces to the three variants SG (9), SHB (5) and SNAG (7) by setting* $s = \frac{1}{1-\beta}$, $s = 0$ *and* $s = 1$, *respectively. Particularly, the update of SG is* $\mathbf{x}_{k+1} = \mathbf{x}_k - \frac{\alpha}{1-\beta}\mathcal{G}(\mathbf{x}_k, \xi_k)$, *where the step size is* $\frac{\alpha}{1-\beta}$.

Proof. We prove the result by separately discussing three different values of s, i.e., 0, 1 and $\frac{1}{1-\beta}$. We show that these three settings in fact correspond to SHB, SNAG and SG, respectively.

1. When $s = 0$, then $\mathbf{y}_{k+1}^s = \mathbf{x}_k$, the update of $\mathbf{x}_{k+1}$ becomes

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha\mathcal{G}(\mathbf{x}_k) + \beta(\mathbf{x}_k - \mathbf{x}_{k-1}), \tag{17}$$

which is exactly the update of the HB method in (5) and (6).

2. When $s = 1$, then $\mathbf{y}_{k+1}^s = \mathbf{y}_{k+1}$, then the update in (10) reduces to that in (7) or (8) of the NAG method.

3. The last special case corresponds to using $s = \frac{1}{1-\beta}$ for SG. We will show that the update of $\mathbf{x}_{k+1}$ is equivalent to

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \frac{\alpha}{1-\beta}\mathcal{G}(\mathbf{x}_k), k \geq 0, \tag{18}$$

which is exactly the update of the gradient method but with a step size $\frac{\alpha}{1-\beta}$. First, we verify (18) holds for $k=0$. From the updates in (10), we have

$$\begin{aligned}\mathbf{x}_1 &= \mathbf{y}_1 + \beta(\mathbf{y}_1^s - \mathbf{y}_0^s)\\ &= \mathbf{x}_0 - \alpha\mathcal{G}(\mathbf{x}_0) + \beta(\mathbf{x}_0 - s\alpha\mathcal{G}(\mathbf{x}_0) - \mathbf{x}_0)\\ &= \mathbf{x}_0 - \alpha\mathcal{G}(\mathbf{x}_0) - s\beta\alpha\mathcal{G}(\mathbf{x}_0)\\ &= \mathbf{x}_0 - \alpha\mathcal{G}(\mathbf{x}_0)(1+\frac{\beta}{1-\beta}) = \mathbf{x}_0 - \frac{\alpha}{1-\beta}\mathcal{G}(\mathbf{x}_0).\end{aligned}$$

Then we show (18) holds for any $k \geq 1$. From the updates in (10), we have

$$\begin{aligned}&\mathbf{x}_{k+1} - \mathbf{x}_k = -\alpha\mathcal{G}(\mathbf{x}_k) + \beta(\mathbf{y}_{k+1}^s - \mathbf{y}_k^s)\\ &= -\alpha\mathcal{G}(\mathbf{x}_k) + \beta(\mathbf{x}_k - s\alpha\mathcal{G}(\mathbf{x}_k) - \mathbf{x}_{k-1} + s\alpha\mathcal{G}(\mathbf{x}_{k-1})).\end{aligned}$$

Then

$$\begin{aligned}&\mathbf{x}_{k+1} - \mathbf{x}_k + s\alpha\mathcal{G}(\mathbf{x}_k) = \beta(\mathbf{x}_k - \mathbf{x}_{k-1} + s\alpha\mathcal{G}(\mathbf{x}_{k-1}))\\ &+ (s - 1 - \beta s)\alpha\mathcal{G}(\mathbf{x}_k).\end{aligned}$$

Since $s = \frac{1}{1-\beta}$, then $s - 1 - \beta s = 0$, thus for any $k \geq 1$

$$\mathbf{x}_{k+1} - \mathbf{x}_k + s\alpha\mathcal{G}(\mathbf{x}_k) = \beta(\mathbf{x}_k - \mathbf{x}_{k-1} + s\alpha\mathcal{G}(\mathbf{x}_{k-1})),$$

Therefore

$$\mathbf{x}_{k+1} - \mathbf{x}_k + s\alpha\mathcal{G}(\mathbf{x}_k) = \beta^k(\mathbf{x}_1 - \mathbf{x}_0 + \frac{\alpha}{1-\beta}\mathcal{G}(\mathbf{x}_0)) = 0,$$

which leads to (18).

□

B Proof of Lemma 1

We re-present Lemma 1.

Lemma 1. *Let* $\mathbf{p}_k$ *be*

$$\mathbf{p}_k = \begin{cases} \frac{\beta}{1-\beta}(\mathbf{x}_k - \mathbf{x}_{k-1} + s\alpha\mathcal{G}(\mathbf{x}_{k-1})),\ k \geq 1 \\ 0, \quad k = 0 \end{cases} \tag{11}$$

and

$$\mathbf{v}_k = \frac{(1-\beta)}{\beta}\mathbf{p}_k. \tag{12}$$

Then for any $k \geq 0$*, we have*

$$\mathbf{x}_{k+1} + \mathbf{p}_{k+1} = \mathbf{x}_k + \mathbf{p}_k - \frac{\alpha}{1-\beta}\mathcal{G}(\mathbf{x}_k), \tag{13}$$

$$\mathbf{v}_{k+1} = \beta\mathbf{v}_k + ((1-\beta)s - 1)\alpha\mathcal{G}(\mathbf{x}_k). \tag{14}$$

Proof. Let us first write down the updates:

$$\begin{aligned}\mathbf{y}_{k+1} &= \mathbf{x}_k - \alpha\mathcal{G}(\mathbf{x}_k)\\ \mathbf{y}_{k+1}^s &= \mathbf{x}_k - s\alpha\mathcal{G}(\mathbf{x}_k)\\ \mathbf{x}_{k+1} &= \mathbf{y}_{k+1} + \beta(\mathbf{y}_{k+1}^s - \mathbf{y}_k^s)\end{aligned}$$

We can see that

$$\begin{aligned}\mathbf{x}_{k+1} &= \mathbf{x}_k - \alpha\mathcal{G}(\mathbf{x}_k)\\ &+ \beta(\mathbf{x}_k - s\alpha\mathcal{G}(\mathbf{x}_k) - \mathbf{x}_{k-1} + s\alpha\mathcal{G}(\mathbf{x}_{k-1})).\end{aligned}$$

If we define $\mathbf{x}_{-1} = \mathbf{x}_0$ and $\mathcal{G}(\mathbf{x}_{-1}) = 0$, the above equation holds for any $k \geq 0$. Similarly, we can write $\mathbf{p}_k$ as

$$\mathbf{p}_k = \frac{\beta}{1-\beta}(\mathbf{x}_k - \mathbf{x}_{k-1} + s\alpha\mathcal{G}(\mathbf{x}_{k-1}))$$

for any $k \geq 0$.

Next, we prove that (13) and (14) hold for any $k \geq 0$. By the definition of $\mathbf{p}_k$, we have

$$\begin{aligned}&\mathbf{x}_{k+1} + \mathbf{p}_{k+1} = \mathbf{x}_{k+1} + \frac{\beta}{1-\beta}(\mathbf{x}_{k+1} - \mathbf{x}_k + s\alpha\mathcal{G}(\mathbf{x}_k))\\ &= \frac{1}{1-\beta}\mathbf{x}_{k+1} - \frac{\beta}{1-\beta}(\mathbf{x}_k - s\alpha\mathcal{G}(\mathbf{x}_k))\\ &\overset{(10)}{=} \frac{1}{1-\beta}[\mathbf{x}_k - \alpha\mathcal{G}(\mathbf{x}_k)\\ &+ \beta(\mathbf{x}_k - s\alpha G(\mathbf{x}_k) - \mathbf{x}_{k-1} + s\alpha G(\mathbf{x}_{k-1}))]\\ &- \frac{\beta}{1-\beta}(\mathbf{x}_k - s\alpha\mathcal{G}(\mathbf{x}_k))\\ &= \frac{1+\beta}{1-\beta}\mathbf{x}_k - \frac{1+s\beta}{1-\beta}\alpha\mathcal{G}(\mathbf{x}_k)\\ &- \frac{\beta}{1-\beta}(\mathbf{x}_{k-1} - s\alpha\mathcal{G}(\mathbf{x}_{k-1}))\\ &- \frac{\beta}{1-\beta}(\mathbf{x}_k - s\alpha\mathcal{G}(\mathbf{x}_k))\\ &= \frac{1}{1-\beta}(\mathbf{x}_k - \alpha\mathcal{G}(\mathbf{x}_k)) - \frac{\beta}{1-\beta}(\mathbf{x}_{k-1} - s\alpha\mathcal{G}(\mathbf{x}_{k-1})).\end{aligned}$$

Similarly, we have

$$\mathbf{x}_k + \mathbf{p}_k = \frac{1}{1-\beta}\mathbf{x}_k - \frac{\beta}{1-\beta}(\mathbf{x}_{k-1} - s\alpha\mathcal{G}(\mathbf{x}_{k-1})).$$

Thus

$$\mathbf{x}_{k+1} + \mathbf{p}_{k+1} = \mathbf{x}_k + \mathbf{p}_k - \frac{1}{1-\beta}\alpha\mathcal{G}(\mathbf{x}_k),$$

which verifies (13).

To verify (14), we use the definition of $\mathbf{v}_k$ and $\mathbf{p}_k$, and have

$$\begin{aligned}\mathbf{v}_{k+1} &= \frac{1-\beta}{\beta}\mathbf{p}_{k+1} = \mathbf{x}_{k+1} - \mathbf{x}_k + s\alpha\mathcal{G}(\mathbf{x}_k)\\ &\overset{(10)}{=} \mathbf{x}_k - \alpha\mathcal{G}(\mathbf{x}_k)\\ &+ \beta(\mathbf{x}_k - s\alpha\mathcal{G}(\mathbf{x}_k) - \mathbf{x}_{k-1} + s\alpha\mathcal{G}(\mathbf{x}_{k-1}))\\ &- \mathbf{x}_k + s\alpha\mathcal{G}(\mathbf{x}_k)\\ &= \beta(\mathbf{x}_k - \mathbf{x}_{k-1}) + [s(1-\beta) - 1]\alpha\mathcal{G}(\mathbf{x}_k)\\ &+ \beta s\alpha\mathcal{G}(\mathbf{x}_{k-1})\end{aligned}$$

and

$$\begin{aligned}&\beta\mathbf{v}_k + [(1-\beta)s - 1]\alpha\mathcal{G}(\mathbf{x}_k)\\ =&\beta(\mathbf{x}_k - \mathbf{x}_{k-1}) + \beta s\alpha\mathcal{G}(\mathbf{x}_{k-1})\\ &+ [(1-\beta)s - 1]\alpha\mathcal{G}(\mathbf{x}_k)\end{aligned}$$

We can see that

$$\mathbf{v}_{k+1} = \beta\mathbf{v}_k + [(1-\beta)s - 1]\alpha\mathcal{G}(\mathbf{x}_k),$$

which verifies (14).

□

C Proof of Lemma 2

We re-present Lemma 2.

Lemma 2. *Given the update in (10), for any $k \geq 0$ we have*

$$\mathbf{x}_{k+1} = \mathbf{x}_0 - \sum_{\tau=0}^{k} \left\{ \frac{1}{1-\beta} - \beta^{k-\tau+1} \frac{1-s(1-\beta)}{1-\beta} \right\} \alpha \mathcal{G}(\mathbf{x}_\tau). \tag{19}$$

Proof. Start with the update of (10). We can rewrite the update by

$$\begin{aligned}
&\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \mathcal{G}(\mathbf{x}_k) \\
&\qquad + \beta(\mathbf{x}_k - s\alpha \mathcal{G}(\mathbf{x}_k) - (\mathbf{x}_{k-1} - s\alpha \mathcal{G}(\mathbf{x}_{k-1}))) \\
&\quad \Rightarrow \\
&\mathbf{x}_{k+1} - (\mathbf{x}_k - s\alpha \mathcal{G}(\mathbf{x}_k)) \\
&\qquad = \beta(\mathbf{x}_k - (\mathbf{x}_{k-1} - s\alpha \mathcal{G}(\mathbf{x}_{k-1}))) \\
&\qquad\quad + ((1-\beta)s - 1)\alpha \mathcal{G}(\mathbf{x}_k) \\
&\quad \Rightarrow \\
&\mathbf{x}_{k+1} - (\mathbf{x}_k - s\alpha \mathcal{G}(\mathbf{x}_k)) \\
&\qquad = \sum_{\tau=0}^{k} ((1-\beta)s - 1)\beta^{k-\tau} \alpha \mathcal{G}(\mathbf{x}_\tau) \\
&\quad \Rightarrow \\
&\mathbf{x}_{k+1} = \mathbf{x}_k - s\alpha \mathcal{G}(\mathbf{x}_k) + \sum_{\tau=0}^{k} ((1-\beta)s - 1)\beta^{k-\tau} \alpha \mathcal{G}(\mathbf{x}_\tau) \\
&\quad \Rightarrow \\
&\mathbf{x}_{k+1} = \mathbf{x}_0 - \sum_{\tau=0}^{k} s\alpha \mathcal{G}(\mathbf{x}_\tau) \\
&\qquad + \sum_{\tau=0}^{k} \sum_{i=0}^{\tau} ((1-\beta)s - 1)\beta^{\tau-i} \alpha \mathcal{G}(\mathbf{x}_i) \\
&\quad \Rightarrow \\
&\mathbf{x}_{k+1} = \mathbf{x}_0 - \sum_{\tau=0}^{k} s\alpha \mathcal{G}(\mathbf{x}_\tau) \\
&\qquad + \frac{(1-\beta)s - 1}{1-\beta} \sum_{\tau=0}^{k} (1 - \beta^{k-\tau+1}) \alpha \mathcal{G}(\mathbf{x}_\tau) \\
&\quad \Rightarrow \\
&\mathbf{x}_{k+1} = \mathbf{x}_0 \\
&\qquad - \sum_{\tau=0}^{k} \left\{ \frac{1}{1-\beta} - \beta^{k-\tau+1} \frac{1-s(1-\beta)}{1-\beta} \right\} \alpha \mathcal{G}(\mathbf{x}_\tau).
\end{aligned} \tag{20}$$

□

D Proof of Theorem 1

Before proving Theorem 1, we present two key lemmas.

Lemma 3. *Let $\mathbf{z}_k = \mathbf{x}_k + \mathbf{p}_k$. For SUM, we have for any $k \geq 0$,*

$$\begin{aligned}
&\mathrm{E}[f(\mathbf{z}_{k+1}) - f(\mathbf{z}_k)] \leq \frac{1}{2L} \mathrm{E}[\|\nabla f(\mathbf{z}_k) - \nabla f(\mathbf{x}_k)\|^2] \\
&\quad + \left(\frac{L\alpha^2}{(1-\beta)^2} - \frac{\alpha}{1-\beta} \right) \mathrm{E}[\|\nabla f(\mathbf{x}_k)\|^2] + \frac{L\alpha^2\sigma^2}{2(1-\beta)^2}.
\end{aligned}$$

Proof. Let $\delta_k = \mathcal{G}_k - \nabla f(\mathbf{x}_k)$. Then $\mathrm{E}[\delta_k] = 0$. Beginning by exploring the smoothness of $f(\mathbf{x})$ we have

$$\begin{aligned}
&f(\mathbf{z}_{k+1}) \leq f(\mathbf{z}_k) + \nabla f(\mathbf{z}_k)^\top (\mathbf{z}_{k+1} - \mathbf{z}_k) \\
&\quad + \frac{L\|\mathbf{z}_{k+1} - \mathbf{z}_k\|^2}{2} \\
&\overset{(13)}{=} f(\mathbf{z}_k) - \frac{\alpha}{1-\beta} \nabla f(\mathbf{z}_k)^\top \mathcal{G}_k + \frac{L}{2} \frac{\alpha^2}{(1-\beta)^2} \|\mathcal{G}_k\|^2 \\
&= f(\mathbf{z}_k) - \frac{\alpha}{1-\beta} \nabla f(\mathbf{z}_k)^\top (\delta_k + \nabla f(\mathbf{x}_k)) \\
&\quad + \frac{L}{2} \frac{\alpha^2}{(1-\beta)^2} \|\mathcal{G}_k\|^2 \\
&= f(\mathbf{z}_k) - \frac{\alpha}{1-\beta} \nabla f(\mathbf{z}_k)^\top \delta_k - \frac{\alpha}{1-\beta} \nabla f(\mathbf{z}_k)^\top \nabla f(\mathbf{x}_k) \\
&\quad + \frac{L}{2} \frac{\alpha^2}{(1-\beta)^2} \|\mathcal{G}_k\|^2 \\
&= f(\mathbf{z}_k) - \frac{\alpha}{1-\beta} \nabla f(\mathbf{z}_k)^\top \delta_k \\
&\quad - \frac{\alpha}{1-\beta} (\nabla f(\mathbf{z}_k) - \nabla f(\mathbf{x}_k))^\top \nabla f(\mathbf{x}_k) \\
&\quad - \frac{\alpha}{1-\beta} \|\nabla f(\mathbf{x}_k)\|^2 + \frac{L}{2} \frac{\alpha^2}{(1-\beta)^2} \|\delta_k + \nabla f(\mathbf{x}_k)\|^2.
\end{aligned}$$

Taking expectation on both sides

$$\begin{aligned}
&\mathrm{E}[f(\mathbf{z}_{k+1}) - f(\mathbf{z}_k)] \\
&\leq \mathrm{E}\left[-\frac{\alpha}{1-\beta} (\nabla f(\mathbf{z}_k) - \nabla f(\mathbf{x}_k))^\top \nabla f(\mathbf{x}_k) \right. \\
&\left. - \frac{\alpha}{1-\beta} \|\nabla f(\mathbf{x}_k)\|^2 + \frac{L}{2} \frac{\alpha^2}{(1-\beta)^2} \|\nabla f(\mathbf{x}_k)\|^2 \right] \\
&\quad + \frac{L}{2} \frac{\alpha^2}{(1-\beta)^2} \mathrm{E}[\|\delta_k\|^2] \\
&= \mathrm{E}\left[-\frac{\alpha}{1-\beta} (\nabla f(\mathbf{z}_k) - \nabla f(\mathbf{x}_k))^\top \nabla f(\mathbf{x}_k) \right] + \\
&\left(\frac{L}{2} \frac{\alpha^2}{(1-\beta)^2} - \frac{\alpha}{1-\beta} \right) \mathrm{E}[\|\nabla f(\mathbf{x}_k)\|^2] + \frac{L\alpha^2}{2(1-\beta)^2} \sigma^2 \\
&\leq \mathrm{E}\left[\frac{1}{2L} \|\nabla f(\mathbf{z}_k) - \nabla f(\mathbf{x}_k)\|^2 + \frac{L\alpha^2}{2(1-\beta)^2} \|\nabla f(\mathbf{x}_k)\|^2 \right] \\
&\quad + \left(\frac{L}{2} \frac{\alpha^2}{(1-\beta)^2} - \frac{\alpha}{1-\beta} \right) \mathrm{E}[\|\nabla f(\mathbf{x}_k)\|^2] + \frac{L\alpha^2}{2(1-\beta)^2} \sigma^2,
\end{aligned}$$

where the last inequality uses the Cauchy-Schwarz inequality. □

Lemma 4. *For SUM, we have for any $k \geq 0$,*

$$\mathrm{E}[\|\nabla f(\mathbf{z}_k) - \nabla f(\mathbf{x}_k)\|^2] \leq \frac{L^2\beta^2((1-\beta)s-1)^2\alpha^2(G^2+\sigma^2)}{(1-\beta)^4}.$$

Proof.

$$\|\nabla f(\mathbf{z}_k) - \nabla f(\mathbf{x}_k)\|^2 \leq L^2\|\mathbf{z}_k - \mathbf{x}_k\|^2 = L^2\|\mathbf{p}_k\|^2 \overset{(12)}{=} \frac{L^2\beta^2}{(1-\beta)^2}\mathrm{E}[\|\mathbf{v}_k\|^2].$$

Recall the recursion in (14):

$$\mathbf{v}_{k+1} = \beta\mathbf{v}_k + ((1-\beta)s-1)\alpha\mathcal{G}_k.$$

Note that $\mathbf{v}_0 = 0$. Denote by $\hat{\alpha} = \alpha((1-\beta)s-1)$. By induction, we can show that

$$\mathbf{v}_k = \hat{\alpha}\sum_{i=0}^{k-1}\beta^{k-1-i}\mathcal{G}_i = \hat{\alpha}\sum_{i=0}^{k-1}\beta^i\mathcal{G}_{k-1-i}$$

Let $\Gamma_{k-1} = \sum_{i=0}^{k-1}\beta^i = \frac{1-\beta^k}{1-\beta}$. Then

$$\begin{aligned}\|\mathbf{v}_k\|^2 &= \left\|\sum_{i=0}^{k-1}\frac{\beta^i}{\Gamma_{k-1}}\hat{\alpha}\mathcal{G}_{k-1-i}\right\|^2\Gamma_{k-1}^2 \\ &\leq \Gamma_{k-1}^2\sum_{i=0}^{k-1}\frac{\beta^i}{\Gamma_{k-1}}\hat{\alpha}^2\|\mathcal{G}_{k-1-i}\|^2 \\ &= \Gamma_{k-1}\sum_{i=0}^{k-1}\beta^i\hat{\alpha}^2\|\mathcal{G}_{k-1-i}\|^2.\end{aligned}$$

Then

$$\mathrm{E}[\|\mathbf{v}_k\|^2] \leq \Gamma_{k-1}\sum_{i=0}^{k-1}\beta^i\hat{\alpha}^2(G^2+\sigma^2)$$

$$= \Gamma_{k-1}^2\hat{\alpha}^2(G^2+\sigma^2) \leq \frac{\alpha^2((1-\beta)s-1)^2(G^2+\sigma^2)}{(1-\beta)^2}.$$

Then

$$\begin{aligned}\|\nabla f(\mathbf{z}_k) - \nabla f(\mathbf{x}_k)\|^2 &\leq \frac{L^2\beta^2}{(1-\beta)^2}\mathrm{E}[\|\mathbf{v}_k\|^2] \\ &\leq \frac{L^2\beta^2((1-\beta)s-1)^2\alpha^2(G^2+\sigma^2)}{(1-\beta)^4}.\end{aligned}$$

□

Here we re-present Theorem 1.

Theorem 1. *(Convergence of SUM) Suppose $f(\mathbf{x})$ is a non-convex and L-smooth function, $\mathrm{E}[\|\mathcal{G}(\mathbf{x};\xi) - \nabla f(\mathbf{x})\|^2] \leq \sigma^2$ and $\|\nabla f(\mathbf{x})\| \leq G$ for any $\mathbf{x}$. Let update (10) run for t iterations with $\mathcal{G}(\mathbf{x}_k;\xi_k)$. By setting $\alpha = \min\{\frac{1-\beta}{2L}, \frac{C}{\sqrt{t+1}}\}$ we have*

$$\begin{aligned}&\min_{k=0,\ldots,t}\mathrm{E}[\|\nabla f(\mathbf{x}_k)\|^2] \\ &\leq \frac{2(f(\mathbf{x}_0)-f_*)(1-\beta)}{t+1}\max\left\{\frac{2L}{1-\beta}, \frac{\sqrt{t+1}}{C}\right\} \\ &+ \frac{C}{\sqrt{t+1}}\frac{L\beta^2((1-\beta)s-1)^2(G^2+\sigma^2)+L\sigma^2(1-\beta)^2}{(1-\beta)^3}.\end{aligned}$$

Proof. Let B, B' be defined as

$$B = \frac{\alpha}{(1-\beta)} - \frac{L\alpha^2}{(1-\beta)^2} > 0$$

$$B' = \frac{L\beta^2((1-\beta)s-1)^2\alpha^2(G^2+\sigma^2)}{2(1-\beta)^4} + \frac{L\alpha^2\sigma^2}{2(1-\beta)^2}.$$

Lemma 3 and Lemma 4 imply that

$$\mathrm{E}[f(\mathbf{z}_{k+1}) - f(\mathbf{z}_k)] \leq -B\mathrm{E}[\|\nabla f(\mathbf{x}_k)\|^2] + B'.$$

By summing the above inequalities for $k = 0, \ldots, t$ and noting that $\alpha < \frac{1-\beta}{L}$,

$$\begin{aligned}&B\sum_{k=0}^{t}\mathrm{E}[\|\nabla f(\mathbf{x}_k)\|^2] \leq \mathrm{E}[f(\mathbf{z}_0) - f(\mathbf{z}_{t+1})] + (t+1)B' \\ &\leq \mathrm{E}[f(\mathbf{z}_0) - f_*] + (t+1)B'.\end{aligned}$$

Then

$$\min_{k=0,\ldots,t}\mathrm{E}[\|\nabla f(\mathbf{x}_k)\|^2] \leq \frac{f(\mathbf{z}_0)-f_*}{(t+1)B} + \frac{B'}{B}.$$

Assume $\alpha \leq \frac{1-\beta}{2L}$, then $B = \frac{\alpha}{1-\beta} - \frac{\alpha^2 L}{(1-\beta)^2} \geq \frac{\alpha}{2(1-\beta)}$. Then

$$\begin{aligned}&\min_{k=0,\ldots,t}\mathrm{E}[\|\nabla f(\mathbf{x}_k)\|^2] \\ &\leq \frac{2(f(\mathbf{z}_0)-f_*)(1-\beta)}{\alpha(t+1)} + \frac{2(1-\beta)}{\alpha}B'. \qquad (21)\end{aligned}$$

Noting that $\alpha = \min\{\frac{1-\beta}{2L}, \frac{C}{\sqrt{t+1}}\}$, we can have

$$\begin{aligned}&\min_{k=0,\ldots,t}\mathrm{E}[\|\nabla f(\mathbf{x}_k)\|^2] \\ &\leq \frac{2(f(\mathbf{z}_0)-f_*)(1-\beta)}{t+1}\max\left\{\frac{2L}{1-\beta}, \frac{\sqrt{t+1}}{C}\right\} \\ &+ \frac{C}{\sqrt{t+1}}\frac{L\beta^2((1-\beta)s-1)^2(G^2+\sigma^2)+L(1-\beta)^2\sigma^2}{(1-\beta)^3}.\end{aligned}$$

We then complete the proof by noting that $\mathbf{z}_0 = \mathbf{x}_0$. □

E Proof of Theorem 2

As in Section D, with a slightly different analysis from that of Lemma 3, we can have the following lemma.

Lemma 5 *Let $\mathbf{z}_k = \mathbf{x}_k + \mathbf{p}_k$. For SUM, we have for any $k \geq 0$,*

$$\begin{aligned}&\mathrm{E}[f(\mathbf{z}_{k+1}) - f(\mathbf{z}_k)] \leq \frac{L\alpha^2\sigma^2}{2(1-\beta)^2} \\ &+ \frac{1}{2L((1-\beta)s-1)^2}\mathrm{E}[\|\nabla f(\mathbf{z}_k) - \nabla f(\mathbf{x}_k)\|^2] \\ &+ \left(\frac{[1+((1-\beta)s-1)^2]\alpha^2 L}{2(1-\beta)^2} - \frac{\alpha}{1-\beta}\right)\mathrm{E}[\|\nabla f(\mathbf{x}_k)\|^2].\end{aligned}$$

Proof. We can follow the same analysis as in the proof of

Lemma 3 and get

$$
\begin{aligned}
&\mathrm{E}[f(\mathbf{z}_{k+1}) - f(\mathbf{z}_k)] \\
&\leq \mathrm{E}\left[-\frac{\alpha}{1-\beta}(\nabla f(\mathbf{z}_k) - \nabla f(\mathbf{x}_k))^\top \nabla f(\mathbf{x}_k)\right] \\
&+ \left(\frac{L}{2}\frac{\alpha^2}{(1-\beta)^2} - \frac{\alpha}{1-\beta}\right)\mathrm{E}[\|\nabla f(\mathbf{x}_k)\|^2] + \frac{L\alpha^2\sigma^2}{2(1-\beta)^2} \\
&\leq \frac{1}{2}\mathrm{E}\left[\frac{1}{L((1-\beta)s-1)^2}\|\nabla f(\mathbf{z}_k) - \nabla f(\mathbf{x}_k)\|^2\right. \\
&\left.+\frac{L\alpha^2((1-\beta)s-1)^2}{(1-\beta)^2}\|\nabla f(\mathbf{x}_k)\|^2\right] \\
&+ \left(\frac{L}{2}\frac{\alpha^2}{(1-\beta)^2} - \frac{\alpha}{1-\beta}\right)\mathrm{E}[\|\nabla f(\mathbf{x}_k)\|^2] + \frac{L\alpha^2\sigma^2}{2(1-\beta)^2} \\
&= \frac{1}{2L((1-\beta)s-1)^2}\mathrm{E}[\|\nabla f(\mathbf{z}_k) - \nabla f(\mathbf{x}_k)\|^2] \\
&+ \left(\frac{\alpha^2 L[1+((1-\beta)s-1)^2]}{2(1-\beta)^2} - \frac{\alpha}{1-\beta}\right)\mathrm{E}[\|\nabla f(\mathbf{x}_k)\|^2] \\
&+ \frac{L\alpha^2\sigma^2}{2(1-\beta)^2}.
\end{aligned}
$$

□

With Lemma 5 and Lemma 4 and a similar analysis as that for Theorem 1, we can easily prove Theorem 2.

F Proof of Proposition 3

We re-present Proposition 3.

Proposition 3. *Assume that $\|\nabla\ell(\mathbf{x},\mathbf{q})\|_2 \leq G$ for any $\mathbf{x}$ and $\mathbf{q}$ and $\ell(\mathbf{x},\mathbf{q})$ is L-smooth w.r.t $\mathbf{x}$. For two data sets $\mathcal{S},\mathcal{S}'$ that differs at one example, let $\mathbf{x}_t$ and $\mathbf{x}'_t$ denote the t-th iterates of running SUM for the empirical risk minimization on S and S', we have*

$$\Delta_{t+1} \leq \sum_{k=0}^{t}\frac{2\alpha G}{n}\eta_k^t + \left(1-\frac{1}{n}\right)\sum_{k=0}^{t}\alpha L\eta_k^t\Delta_k$$

with $\Delta_0 = 0$, where $\eta_k^t = \frac{1}{1-\beta} - \beta^{t-k+1}\frac{1-s(1-\beta)}{1-\beta}$.

Proof. Recall that in Lemma 2, we have

$$
\begin{aligned}
\mathbf{x}_{t+1} &= \mathbf{x}_0 - \sum_{k=0}^{t}\eta_k^t\alpha\mathcal{G}(\mathbf{x}_k) \\
&= \mathbf{x}_0 - \sum_{k=0}^{t}\eta_k^t\alpha\nabla\ell(\mathbf{x}_k,\mathbf{q}_{i_k}),
\end{aligned}
$$

where $\eta_k^t = \frac{1}{1-\beta} - \beta^{t-k+1}\frac{1-s(1-\beta)}{1-\beta}$.

Then we could upper bound of Δ_{t+1} as follows

$$
\begin{aligned}
\Delta_{t+1} &= \mathrm{E}\|\mathbf{x}_{t+1} - \mathbf{x}'_{t+1}\| \\
&= \mathrm{E}\|\mathbf{x}_0 - \sum_{k=0}^{t}\eta_k^t\alpha\nabla\ell(\mathbf{x}_k,\mathbf{q}_{i_k}) - (\mathbf{x}'_0 - \sum_{k=0}^{t}\eta_k^t\alpha\nabla\ell(\mathbf{x}'_k,\mathbf{q}_{i'_k}))\| \\
&\leq \mathrm{E}\|\mathbf{x}_0 - \mathbf{x}'_0\|_2 + \sum_{k=0}^{t}\eta_k^t\alpha\mathrm{E}\|\nabla\ell(\mathbf{x}_k,\mathbf{q}_{i_k}) - \nabla\ell(\mathbf{x}'_k,\mathbf{q}_{i'_k})\| \\
&= \sum_{k=0}^{t}\eta_k^t\alpha\mathrm{E}\|\nabla\ell(\mathbf{x}_k,\mathbf{q}_{i_k}) - \nabla\ell(\mathbf{x}'_k,\mathbf{q}_{i'_k})\|
\end{aligned}
\tag{22}
$$

To bound the expectation term on the R.H.S, we can use the fact that with probability $1/n$, $i_k \neq i'_k$ and $\|\nabla\ell(\mathbf{x}_k,\mathbf{q}_{i_k}) - \nabla\ell(\mathbf{x}'_k,\mathbf{q}_{i'_k})\| \leq 2G$ due to Lipschitz continuity of $\ell(\mathbf{x},\mathbf{q})$, and with probability $1-1/n$, $i_k = i'_k$ and $\|\nabla\ell(\mathbf{x}_k,\mathbf{q}_{i_k}) - \nabla\ell(\mathbf{x}'_k,\mathbf{q}_{i'_k})\| \leq L\|\mathbf{x}_k - \mathbf{x}'_k\|$ due to the smoothness of $\ell(\mathbf{x},\mathbf{z})$. Therefore,

$$
\begin{aligned}
\Delta_{t+1} &\leq \sum_{k=0}^{t}\eta_k^t\alpha\left\{(1-\frac{1}{n})L\mathrm{E}||\mathbf{x}_k - \mathbf{x}'_k||_2 + \frac{2G}{n}\right\} \\
&= \sum_{k=0}^{t}\eta_k^t\alpha\left\{(1-\frac{1}{n})L\Delta_k + \frac{2G}{n}\right\} \\
&= \sum_{k=0}^{t}\frac{2\alpha G}{n}\eta_k^t + \left(1-\frac{1}{n}\right)\sum_{k=0}^{t}\alpha L\eta_k^t\Delta_k.
\end{aligned}
\tag{23}
$$

□