THE ROLE OF MOMENTUM PARAMETERS IN THE OPTIMAL CONVERGENCE OF ADAPTIVE POLYAK'S HEAVY-BALL METHODS

Anonymous authors
Paper under double-blind review

ABSTRACT

The adaptive SGD with momentum has been widely adopted in deep learning as well as convex optimization. In practice, the last iterate is commonly used as the final solution to make decisions. However, the available regret analysis and the setting of constant momentum parameters only guarantee the optimal convergence of the averaged solution. In this paper, we fill this theory-practice gap by investigating the convergence of the last iterate (referred to as *individual convergence*), which is a more difficult task than convergence analysis of the averaged solution. Specifically, in the constrained convex cases, we prove that the adaptive Polyak's Heavy-ball (HB) method, in which only the step size is updated using the exponential moving average strategy, attains an optimal individual convergence rate of $O(\frac{1}{\sqrt{t}})$, as opposed to the optimality of $O(\frac{\log t}{\sqrt{t}})$ of SGD, where t is the number of iteration. Our new analysis not only shows how the HB momentum and its time-varying weight $\beta_{1t} = \frac{t}{t+2}$ help us to achieve the acceleration in convex optimization but also gives valuable hints how the momentum parameters should be scheduled in deep learning. Empirical results on optimizing convex functions and training deep networks validate the correctness of our convergence analysis and demonstrate the improved performance of the adaptive HB methods.

1 INTRODUCTION

One of the most popular optimization algorithms in deep learning is the momentum method (Krizhevsky et al., 2012). The first momentum can be traced back to the pioneering work of Polyak's heavy-ball (HB) method (Polyak, 1964), which helps accelerate SGD in the relevant direction and dampens oscillations (Ruder, 2016). Recent studies also find that the HB momentum has the potential to escape from the local minimum and saddle points (Ochs et al., 2014; Sun et al., 2019). From the perspective of theoretical analysis, HB enjoys a smaller convergence factor than SGD when the objective function is twice continuously differentiable and strongly convex (Ghadimi et al., 2015). In nonsmooth convex cases, with suitably chosen step size, HB attains an optimal convergence rate of $O(\frac{1}{\sqrt{t}})$ in terms of the averaged output (Yang et al., 2016), where t is the number of iteration.

To overcome the data-independent limitation of predetermined step size rules, some adaptive gradient methods have been proposed to exploit the geometry of historical data. The first algorithm in this line is Adagrad (Duchi et al., 2011). The intuition behind Adagrad is that the seldom-updated weights should be updated with a larger step size than the frequently-updated weights. Typically, Adagrad rescales each coordinate and estimates the predetermined step size by a sum of squared past gradient values. As a result, Adagrad has the same convergence rate as vanilla SGD but enjoying a smaller factor especially in sparse learning problems. The detailed analysis of Adagrad (Mukkamala & Hein, 2017) implies that one can derive similar convergence rates for the adaptive variants of the predetermined step size methods without additional difficulties.

Unfortunately, experimental results illustrate that Adagrad under-performed when applied to training deep neural newtworks (Wilson et al., 2017). Practical experience has led to the development of adaptive methods that is able to emphasize the more recent gradients. Specifically, an exponential moving average (EMA) strategy was proposed in RMSProp to replace the cumulative sum operation (Tieleman & Hinton, 2012). Adam (Kingma & Ba, 2014), which remains one of the most popular

optimization algorithms in deep learning till today, built upon RMSProp together with updating the search directions via the HB momentum. Generally speaking, the gradient-based momentum algorithms that simultaneously update the search directions and learning rates using the past gradients are referred to as the Adam-type method (Chen et al., 2019). This kinds of methods have achieved several state-of-the-art results on various learning tasks (Sutskever et al., 2013).

Compared with HB and Adagrad, the main novelty of Adam lies in applying EMA to gradient estimate (first-order) and to element-wise square-of-gradients (second-order), with parameters β_{1t} and β_{2t} (Alacaoglu et al., 2020). However, the use of EMA causes a lot of complexities to the convergence analysis. For example, in the online setting, (Kingma & Ba, 2014) offered a proof that Adam would converge to the optimum. Despite its remarkable practicality, Adam suffers the non-convergence issue . To overcome its advantages, several variants such as AMSgrad and AdamNC were proposed (Reddi et al., 2018). Unfortunately, the best regret bound of AMSgrad is only $O(\sqrt{\log t}\sqrt{t})$ for nonsmooth convex problems (Reddi et al., 2018), as opposed to the optimality of $O(\sqrt{t})$ of SGD. On the other hand, EMA uses only the current gradient in updating the search direction while the original HB can use the past gradients information (Zou et al., 2018). This will lead the update to stagnate when β_{1t} is very close to 1. Fortunately, such a dilemma will not appear in Polyak's HB method.

In this paper, we will focus on the adaptive Polyak's HB method, in which only the step size is updated using EMA. Despite various practical performance reported for the Adam-type methods, there still exists gaps between theoretical guarantees and empirical success.

- First of all, some important regret bounds have been established to guarantee the performance of online Adam-type algorithms. Nevertheless, the online-to-batch conversion can inevitably lead the solution of the induced stochastic algorithm to take the form of averaging of all the past iterates. In practice, the last iterate is popularly used as the final solution, which has the advantage of readily enforcing the learning structure (Chen et al., 2012). For SGD, the convergence of the last iterate, which is referred to as *individual convergence* in (Tao et al., 2020b), was posed as an open problem (Shamir, 2012). Only recently, its optimal individual convergence rate is proved to be $O(\frac{\log t}{\sqrt{t}})$ and $O(\frac{\log t}{t})$ for general and strongly convex problems respectively (Harvey et al., 2019; Jain et al., 2019). Despite enjoying the optimal averaging convergence (Yang et al., 2016), as far as know, the individual convergence has not been discussed for the adaptive HB.
- Secondly, the momentum technique is often claimed as an accelerated strategy in machine learning community. However so far, almost all the theoretical analysis is only limited to the Nesterov's accelerated gradient (NAG) (Nesterov, 1983) method especially in smooth cases (Hu et al., 2009; Liu & Belkin, 2020), which accelerates the rate of SGD from $O(\frac{1}{t})$ to $O(\frac{1}{t^2})$. Only recently, it was discovered that NAG is capable of accelerating the rate of individual convergence of SGD from $O(\frac{\log t}{\sqrt{t}})$ to $O(\frac{1}{\sqrt{t}})$ (Tao et al., 2020a) in nonsmooth convex cases. Nevertheless, as far as we know, there is still a lack of the report about the acceleration of the adaptive HB.
- Finally, in practice, almost all the momentum and Adam-type algorithms are often used with a constant momentum parameter β_{1t} (typically between 0.9 and 0.99). In theory, regret guarantees in the online Adam require a rapidly decaying $\beta_{1t} \to 1$ schedule. This gap is recently bridged by getting the same regret bounds as that in (Reddi et al., 2018) with a constant β_{1t}. In each state-of-the-art deep learning library (e.g. TensorFlow, PyTorch and Keras), HB is named as SGD with momentum and β_{1t} is empirically set to 0.9 (Ruder, 2016). Despite its intuition in controlling the number of forgotten past gradients and its guarantee in optimal averaging convergence (Yang et al., 2016), as far as we know, how β_{1t} affects the optimal individual convergence has not been discussed (Gitman et al., 2019).

The goal of this paper is to close a theory-practice gap when using HB methods to train the deep neural networks as well as optimize the convex objective functions. Specifically,

- By setting $\beta_{1t} = \frac{t}{t+2}$, we prove that the adaptive HB attains an optimal individual convergence rate of $O(\frac{1}{\sqrt{t}})$ (Theorem 4), as opposed to the optimality of $O(\frac{\log t}{\sqrt{t}})$ of SGD. Our proof is different from all the existing analysis of averaging convergence. It not only

provides a theoretical guarantee for the acceleration of HB but also clarifies how the momentum and its parameter β_{1t} help us to achieve the optimal individual convergence.

- If $0 \leq \beta_t \equiv \beta < 1$, we prove that the optimal averaging convergence of the adaptive HB is guaranteed (Theorem 5). However, in order to get the optimal individual convergence, β_{1t} has to be time-varying (Theorem 4). Note $\beta_{1t} = \frac{t}{t+2} \to 1$, thus our new analysis not only offers an interesting explanation why we usually restrict $\beta_{1t} \to 1$ but also gives valuable hints how the momentum parameters should be scheduled in deep learning.

We mainly focus on the proof of optimal individual convergence of HB (Theorem 2, Appendix A). The analysis of averaging convergence (Theorem 3) is simpler. Their extensions to include adaptive strategy are slightly more complex (Theorem 4 and 5), but it is similar to the proof of Adagrad (Mukkamala & Hein, 2017) and the details can be found in supplementary material.

2 Problem Statement and Related Work

Consider the following optimization problem,

$$\min f(\mathbf{w}),\ s.t.\ \mathbf{w} \in \mathbf{Q}. \tag{1}$$

where $\mathbf{Q} \subseteq \mathbb{R}^d$ is a closed convex set and $f(\mathbf{w})$ is a convex function. Denote that $\mathbf{w}^*$ is an optimal solution and P is the projection operator on $\mathbf{Q}$. Generally, the averaging convergence is defined as

$$f(\bar{\mathbf{w}}_t) - f(\mathbf{w}^*) \leq \epsilon(t), \tag{2}$$

where $\bar{\mathbf{w}}_t = \sum_{i=1}^{t} \mathbf{w}_i$ and $\epsilon(t)$ is the convergence bound about t. In contrast, the *individual convergence* is described as

$$f(\mathbf{w}_t) - f(\mathbf{w}^*) \leq \epsilon(t). \tag{3}$$

Throughout this paper, we use $\mathbf{g}(\mathbf{w}_t)$ to denote the subgradient of f at $\mathbf{w}_t$. Projected subgradient descent (PSG) is one of the most fundamental algorithms for solving problem (1) (Dimitri P. et al., 2003), and the iteration of which is

$$\mathbf{w}_{t+1} = P[\mathbf{w}_t - \alpha_t \mathbf{g}(\mathbf{w}_t)],$$

where $\alpha_t > 0$ is the step size. To analyze the convergence, we need the following assumption.

Assumption 1. *Assume that there exists a number $M > 0$ such that*

$$\|\mathbf{g}(\mathbf{w})\| \leq M,\ \forall \mathbf{w} \in \mathbf{Q}.$$

It is known that the optimal bound for the nonsmooth convex problem (1) is $O(\frac{1}{\sqrt{t}})$ (Nemirovsky & Yudin, 1983). PSG can attain this optimal convergence rate but in terms of the averaged output while its optimal individual rate is only $O(\frac{\log t}{\sqrt{t}})$ (Harvey et al., 2019; Jain et al., 2019).

When $\mathbf{Q} = \mathbb{R}^N$, the regular HB for solving the unconstrained problem (1) is

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha_t \mathbf{g}(\mathbf{w}_t) + \beta_t(\mathbf{w}_t - \mathbf{w}_{t-1}). \tag{4}$$

If $0 \leq \beta_t \equiv \beta < 1$, the key property of HB is that it can be reformulated as (Ghadimi et al., 2015)

$$\mathbf{w}_{t+1} + \mathbf{p}_{t+1} = \mathbf{w}_t + \mathbf{p}_t - \frac{\alpha_t}{1-\beta}\mathbf{g}(\mathbf{w}_t), \text{where } \mathbf{p}_t = \frac{\beta}{1-\beta}(\mathbf{w}_t - \mathbf{w}_{t-1}). \tag{5}$$

Thus its convergence analysis makes almost no difference to that of PSG. Especially, if $\alpha_t \equiv \frac{\alpha}{\sqrt{T}}$, its averaging convergence rate is $O(\frac{1}{\sqrt{T}})$ (Yang et al., 2016), where T is the total number of iteration.

Simply speaking, the regular Adam (Kingma & Ba, 2014) takes the form of

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \frac{\alpha}{\sqrt{t}} V_t^{-\frac{1}{2}} \hat{\mathbf{g}}_t,$$

where $\hat{\mathbf{g}}(\mathbf{w}_t)$ is a unbiased estimation of $\mathbf{g}(\mathbf{w}_t)$, and

$$\hat{\mathbf{g}}_t = \beta_{1t}\hat{\mathbf{g}}_{t-1} + (1-\beta_{1t})\hat{\mathbf{g}}(\mathbf{w}_t),\ V_t = \beta_{2t}V_{t-1} + (1-\beta_{2t})\text{diag}(\hat{\mathbf{g}}(\mathbf{w}_t)\hat{\mathbf{g}}(\mathbf{w}_t)^\top).$$

3 INDIVIDUAL CONVERGENCE OF HB

To solve the constrained problem (1), HB can be naturally reformulated as

$$\mathbf{w}_{t+1} = P_{\mathbf{Q}}[\mathbf{w}_t - \alpha_t \mathbf{g}(\mathbf{w}_t) + \beta_t(\mathbf{w}_t - \mathbf{w}_{t-1})]. \tag{6}$$

We first prove a key lemma, which extends (5) to the constrained and time-varying cases.

Lemma 1. *Let $\{\mathbf{w}_t\}_{t=1}^{\infty}$ be generated by HB (6). Let*

$$\mathbf{p}_t = t(\mathbf{w}_t - \mathbf{w}_{t-1}),\ \beta_t = \frac{t}{t+2},\ \alpha_t = \frac{\alpha}{(t+2)\sqrt{t}}.$$

Then HB (6) can be reformulated as

$$\mathbf{w}_{t+1} + \mathbf{p}_{t+1} = P_{\mathbf{Q}}[\mathbf{w}_t + \mathbf{p}_t - \frac{\alpha}{\sqrt{t}}\mathbf{g}(\mathbf{w}_t)]. \tag{7}$$

Due to the non-expansive property of $P_{\mathbf{Q}}$ (Dimitri P. et al., 2003), Lemma 1 implies that the convergence analysis for unconstrained problems can be applied to analyze the constrained problems.

Theorem 2. *Assume that $\mathbf{Q}$ is bounded. Let $\{\mathbf{w}_t\}_{t=1}^{\infty}$ be generated by HB (6). Set*

$$\beta_t = \frac{t}{t+2} \text{ and } \alpha_t = \frac{\alpha}{(t+2)\sqrt{t}}.$$

Then

$$f(\mathbf{w}_t) - f(\mathbf{w}^*) \leq O(\frac{1}{\sqrt{t}}).$$

It is necessary to give some remarks about Theorem 2.

- In nonsmooth convex cases, Theorem 2 shows that the individual convergence rate of SGD can be accelerated from $O(\frac{\log t}{\sqrt{t}})$ to $O(\frac{1}{\sqrt{t}})$ via the HB momentum. The proof here clarifies how the HB-type momentum $\mathbf{w}_t - \mathbf{w}_{t-1}$ and its time-varying weight β_t help us to derive the optimal individual convergence.
- The convergence analysis in Theorem 2 is obviously different from the regret analysis in all the available papers, this is because the connection between $f(\mathbf{w}_t) - f(\mathbf{w}^*)$ and $f(\mathbf{w}_{t-1}) - f(\mathbf{w}^*)$ should be established here. It can be seen that seeking an optimal individual convergence is more difficult than the analysis of averaging convergence in many papers such as (Zinkevich, 2003) and (Yang et al., 2016).
- We can get a stochastic HB by replacing the subgradient $\mathbf{g}(\mathbf{w}_t)$ in (6) with its unbiased estimation $\hat{\mathbf{g}}(\mathbf{w}_t)$. Such substitution will not influence our convergence analysis. This means that we can get $\mathbb{E}[f(\mathbf{w}_t) - f(\mathbf{w}^*)] \leq O(\frac{1}{\sqrt{t}})$ under the same assumptions.

If β_t remains a constant, we can get the averaging convergence rate, in which the proof of the first part is similar to Lemma 1 and that of the second part is similar to online PSG (Zinkevich, 2003).

Theorem 3. *Assume that $\mathbf{Q}$ is bounded and $0 \leq \beta_t \equiv \beta < 1$. Let $\{\mathbf{w}_t\}_{t=1}^{\infty}$ be generated by HB (6). Set*

$$\mathbf{p}_t = \frac{\beta}{1-\beta}(\mathbf{w}_t - \mathbf{w}_{t-1}) \text{ and } \alpha_t = \frac{\alpha}{\sqrt{t}}.$$

Then we have

$$\mathbf{w}_{t+1} + \mathbf{p}_{t+1} = P_{\mathbf{Q}}[\mathbf{w}_t + \mathbf{p}_t - \frac{\alpha_t}{1-\beta}\mathbf{g}(\mathbf{w}_t)],\ f(\frac{1}{t}\sum_{k=1}^{t}\mathbf{w}_k) - f(\mathbf{w}^*) \leq O(\frac{1}{\sqrt{t}}).$$

If Q is not bounded, the boundness of sequence $\|\mathbf{w}^* - (\mathbf{w}_{t+1} + \mathbf{p}_{t+1})\|$ can not be ensured, which may lead to the failure of Theorem 3. Fortunately, like that in (Yang et al., 2016), $\mathbb{E}[f(\frac{1}{T}\sum_{k=1}^{T}\mathbf{w}_k) - f(\mathbf{w}^*)] \leq O(\frac{1}{\sqrt{T}})$ still holds, but we need to set $\alpha_t \equiv \frac{\alpha}{\sqrt{T}}$, where T is the total number of iteration.

4 EXTENSION TO ADAPTIVE CASES

It is easy to find that HB (7) is in fact a gradient-based algorithm with predetermined step size $\frac{\alpha}{\sqrt{t}}$. Thus its adaptive variant with EMA can be naturally formulated as

$$\mathbf{w}_{t+1} = P_{\mathbf{Q}}[\mathbf{w}_t - \frac{\alpha\beta_{1t}}{t\sqrt{t}} V_t^{-\frac{1}{2}} \hat{\mathbf{g}}(\mathbf{w}_t) + \beta_{1t}(\mathbf{w}_t - \mathbf{w}_{t-1})]. \tag{8}$$

where

$$\beta_{1t} = \frac{t}{t+2},\ V_t = \beta_{2t} V_{t-1} + (1-\beta_{2t})\text{diag}(\hat{\mathbf{g}}(\mathbf{w}_t)\hat{\mathbf{g}}(\mathbf{w}_t)^\top).$$

The detailed steps of the adaptive HB are shown in Algorithm 1.

Algorithm 1 Adaptive HB

Input: momentum parameters β_{1t}, β_{2t}, constant $\delta > 0$, the total number of iterations T

Initialize $\mathbf{w}_0 = \mathbf{0}$, $V_0 = \mathbf{0}_{d\times d}$

repeat

  $\hat{\mathbf{g}}_t(\mathbf{w}_t) = \nabla f_t(\mathbf{w}_t)$,

  $V_t = \beta_{2t} V_{t-1} + (1-\beta_{2t}) diag(\hat{\mathbf{g}}_t(\mathbf{w}_t)\hat{\mathbf{g}}_t(\mathbf{w}_t)^\top)$,

  $\hat{V}_t = V_t^{\frac{1}{2}} + \frac{\delta}{\sqrt{t}} I_d$,

  $\mathbf{w}_{t+1} = P_{\mathbf{Q}}[\mathbf{w}_t - \frac{\alpha\beta_{1t}}{t\sqrt{t}} \hat{V}_t^{-1} \hat{\mathbf{g}}(\mathbf{w}_t) + \beta_{1t}(\mathbf{w}_t - \mathbf{w}_{t-1})]$,

until $t = T$

Output: $\mathbf{w}_T$

Theorem 4. *Assume that* $\mathbf{Q}$ *is a bounded set. Let* $\{\mathbf{w}_t\}_{t=1}^{\infty}$ *be generated by the adaptive HB (Algorithm 1). Denote* $\mathbf{p}_t = t(\mathbf{w}_t - \mathbf{w}_{t-1})$. *Suppose that* $\beta_{1t} = \frac{t}{t+2}$ *and* $1 - \frac{1}{t} \le \beta_{2t} \le 1 - \frac{\gamma}{t}$ *for some* $0<\gamma \le 1$. *Then*

$$\mathbf{w}_{t+1} + \mathbf{p}_{t+1} = P_{\mathbf{Q}}[\mathbf{w}_t + \mathbf{p}_t - \frac{\alpha}{\sqrt{t}} \hat{V}_t^{-1} \hat{\mathbf{g}}(\mathbf{w}_t)] \tag{9}$$

$$\mathbb{E}[f(\mathbf{w}_t) - f(\mathbf{w}^*)] \le O(\frac{1}{\sqrt{t}}).$$

The proof of (9) is identical to that of Lemma 1. It is easy to find that (9) is an adaptive variant of (7). This implies that the proof of the second part is similar to that of Adagrad (Mukkamala & Hein, 2017). When $0 \le \beta_{1t} \equiv \beta < 1$, the adaptive variant of HB (6) is

$$\mathbf{w}_{t+1} = P_{\mathbf{Q}}[\mathbf{w}_t - \frac{\alpha}{\sqrt{t}} V_t^{-\frac{1}{2}} \hat{\mathbf{g}}(\mathbf{w}_t) + \beta(\mathbf{w}_t - \mathbf{w}_{t-1})]. \tag{10}$$

where

$$V_t = \beta_{2t} V_{t-1} + (1-\beta_{2t})\text{diag}(\hat{\mathbf{g}}(\mathbf{w}_t)\hat{\mathbf{g}}(\mathbf{w}_t)^\top).$$

Similar to the proof of Theorem 4, we can get the following averaging convergence.

Theorem 5. *Assume that* $\mathbf{Q}$ *is bounded and* $0 \le \beta_{1t} \equiv \beta < 1$ *in Algorithm 1. Let* $\{\mathbf{w}_t\}_{t=1}^{\infty}$ *be generated by the adaptive HB ((Algorithm 1)). Suppose that* $1-\frac{1}{t} \le \beta_{2t} \le 1-\frac{\gamma}{t}$ *for some* $0<\gamma \le 1$. *Denote* $\mathbf{p}_t = \frac{\beta}{1-\beta}(\mathbf{w}_t - \mathbf{w}_{t-1})$. *Then*

$$\mathbf{w}_{t+1} + \mathbf{p}_{t+1} = P_{\mathbf{Q}}[\mathbf{w}_t + \mathbf{p}_t - \frac{\alpha}{(1-\beta)\sqrt{t}} \hat{V}_t^{-1} \hat{\mathbf{g}}(\mathbf{w}_t)],\ \mathbb{E}[f(\frac{1}{t}\sum_{k=1}^{t} \mathbf{w}_k) - f(\mathbf{w}^*)] \le O(\frac{1}{\sqrt{t}}).$$

It is necessary to give some remarks about Theorem 4 and Theorem 5.

- The adaptive HB is usually used with a constant β_{1t} in deep learning. However, according to Theorem 5, the constant β_{1t} only guarantees the optimal data-dependent averaging convergence. The convergence property of the last iterate still remains unknown.
- In order to guarantee the optimal individual convergence, according to Theorem 4, β_{1t} has to be time-varying. $\beta_{1t} = \frac{t}{t+2}$ can explain why usually restrict $\beta_{1t} \to 1$ in practice. It also offers a new schedule about the selection of momentum parameters in deep learning.

5 EXPERIMENTS

In this section, we present some empirical results to validate the correctness of our convergence analysis and demonstrate the improved performance of the adaptive HB methods. The experiments on optimizing the constrained hinge loss problem are given in Appendix B. For fair comparison, we independently repeat the experiments five times and report the averaged the results.

5.1 TRAINING DEEP NEURAL NETWORKS

This experiment is to show the improved performance of Adaptive HB on 4-layer CNN and ResNet-18. The experiments are conducted on a sever with 2 NVIDIA 2080Ti GPUs.

Datasets: MNIST (60000 training samples, 10000 test samples), CIFAR10 (50000 training samples, 10000 test samples), and CIFAR100 (50000 training samples, 10000 test samples).

Algorithms: Adam (α, $\beta_{1t} \equiv 0.9$, $\beta_{2t} \equiv 0.999$, $\epsilon = 10^{-8}$) (Kingma & Ba, 2014), SGD ($\alpha_t \equiv \alpha$), SGD-momentum ($\alpha_t \equiv \alpha$, $\beta_t \equiv 0.9$), Adagrad ($\alpha_t \equiv \alpha$) (Duchi et al., 2011), RMSprop ($\alpha_t \equiv \alpha$, $\beta_{2t} \equiv 0.9$, $\epsilon = 10^{-8}$) (Tieleman & Hinton, 2012). For our adaptive HB, $\gamma = 0.1$ and $\delta = 10^{-8}$. Different from the existing methods, we set $\beta_{1t} = \frac{t}{t+2}$ and $\beta_{2t} = 1 - \frac{\gamma}{t}$ in Algorithm 1, where t is the number of the epoch. Within each epoch, β_{1t} and β_{2t} remain unchanged.

Note that all methods have only one adjustable parameter α, we choose α from the set of $\{0.1, 0.01, 0.001, 0.0001\}$ for all experiments. Following (Mukkamala & Hein, 2017) and (Wang et al., 2020), we design a simple 4-layer CNN architecture that consists two convolutional layers (32 filters of size 3×3), one max-pooling layer (2×2 window and 0.25 dropout) and one fully connected layer (128 hidden units and 0.5 dropout). We also use weight decay and batch normalization to reduce over-fitting. The optimal rate is always chosen for each algorithm separately so that one achieves either best training objective or best test performance after a fixed number of epochs.

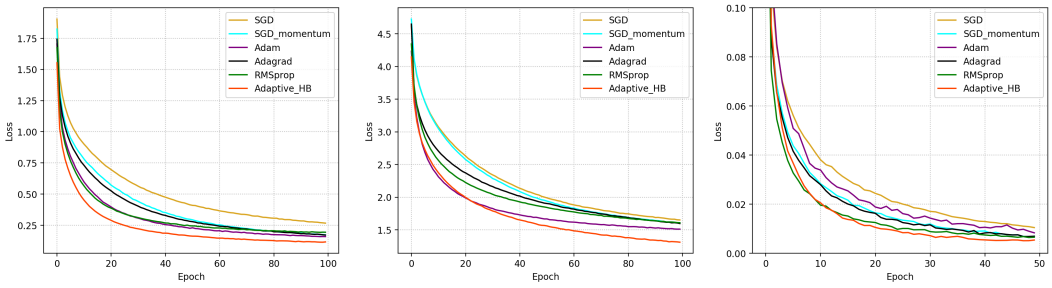

Figure 1: Training loss v.s. number of epochs on different datasets for 4-layer CNN: (left) CIFAR10, (middle) CIFAR100, (right) MNIST

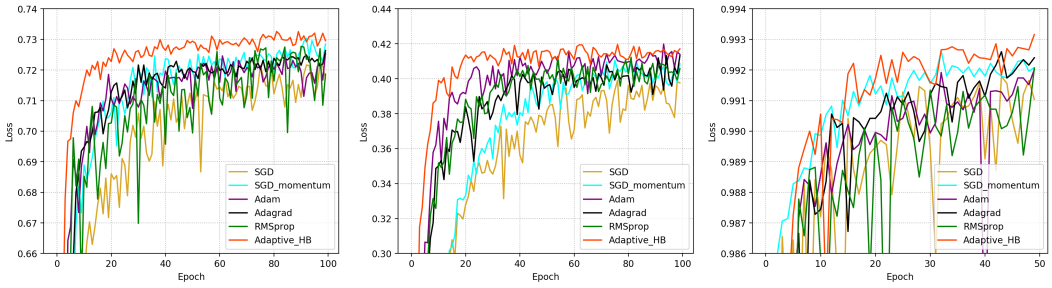

Figure 2: Test accuracy v.s. number of epochs on different datasets for 4-layer CNN: (left) CIFAR10, (middle) CIFAR100, (right) MNIST

The loss function is the cross-entropy. The training loss results are illustrated in Figure 1 and 3, and the test accuracy results are presented in Figure 2 and 4. As can be seen, the adaptive HB achieves the improved training loss. Moreover, this improvement also leads to good performance on test

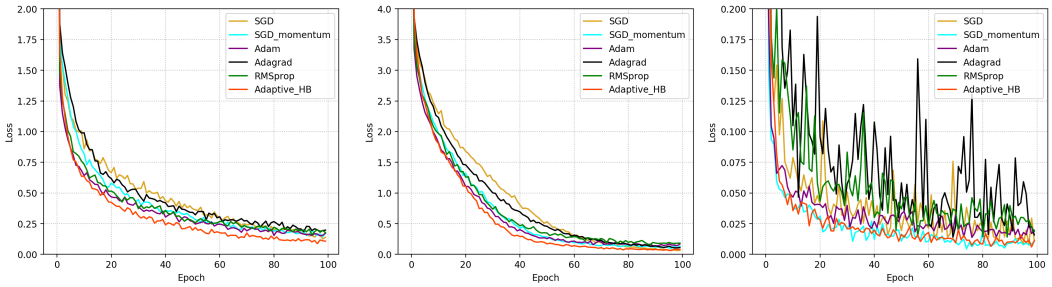

Figure 3: Training loss v.s. number of epochs on different datasets for ResNet-18: (left) CIFAR10, (middle) CIFAR100, (right) MNIST

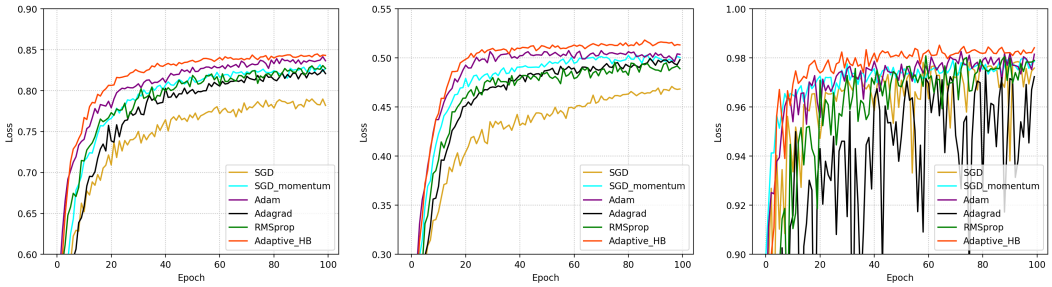

Figure 4: Test accuracy v.s. number of epochs on different datasets for ResNet-18: (left) CIFAR10, (middle) CIFAR100, (right) MNIST

accuracy. The experimental results show that although the last iterate of the adaptive HB is optimal for general convex functions, its schedule about the momentum parameters could gain improved practical performance even in deep learning tasks.

6 Conclusion

In this paper, we prove that the adaptive HB method attains an optimal data-dependent individual convergence rate in the constrained convex cases, which bridges a theory-practice gap in using momentum methods to train the deep neural networks as well as optimize the convex functions. Our new analysis not only clarifies how the HB momentum and its time-varying weight $\beta_{1t} = \frac{t}{t+2}$ help us to achieve the acceleration but also gives valuable hints how its momentum parameters should be scheduled in deep learning. Empirical results on optimizing convex functions validate the correctness of our convergence analysis and several typical deep learning experiments demonstrate the improved performance of the adaptive HB.

A Appendix

Lemma 6. *(Dimitri P. et al., 2003) For* $\mathbf{w} \in \mathbb{R}^d, \mathbf{w}_0 \in \mathbf{Q}$,

$$\langle \mathbf{w} - \mathbf{w}_0, \mathbf{u} - \mathbf{w}_0 \rangle \leq 0$$

for all $\mathbf{u} \in \mathbf{Q}$ *if and only if* $\mathbf{w}_0 = P(\mathbf{w})$.

Proof for Lemma 1. The projection operation can be rewritten as an optimization problem (Duchi, 2018), i.e., $\mathbf{w}_{t+1} = P_{\mathbf{Q}}[\mathbf{w}_t - \alpha_t \mathbf{g}(\mathbf{w}_t) + \beta_t(\mathbf{w}_t - \mathbf{w}_{t-1})]$ is equivalent to

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w} \in \mathbf{Q}} \{\alpha_t \langle \mathbf{g}(\mathbf{w}_t), \mathbf{w} \rangle + \frac{1}{2} \|\mathbf{w} - \mathbf{w}_t - \beta_t(\mathbf{w}_t - \mathbf{w}_{t-1})\|^2\}. \tag{11}$$

Then, $\forall \mathbf{u} \in \mathbf{Q}$, we have

$$\langle \mathbf{w}_{t+1} - \mathbf{w}_t - \beta_t(\mathbf{w}_t - \mathbf{w}_{t-1}) + \alpha_t \mathbf{g}(\mathbf{w}_t), \mathbf{w}_{t+1} - \mathbf{w} \rangle \leq 0.$$

This is

$$\langle \mathbf{w}_{t+1} + \mathbf{p}_{t+1} - (\mathbf{w}_t + \mathbf{p}_t) + \frac{\alpha}{\sqrt{t}}\mathbf{g}(\mathbf{w}_t), \mathbf{w}_{t+1} - \mathbf{w}\rangle \leq 0 \tag{12}$$

Specifically,

$$\langle \mathbf{w}_{t+1} + \mathbf{p}_{t+1} - (\mathbf{w}_t + \mathbf{p}_t) + \frac{\alpha}{\sqrt{t}}\mathbf{g}(\mathbf{w}_t), \mathbf{w}_{t+1} - \mathbf{w}_t\rangle \leq 0. \tag{13}$$

From (12) and (13),

$$\langle \mathbf{w}_{t+1} + \mathbf{p}_{t+1} - (\mathbf{w}_t + \mathbf{p}_t) + \frac{\alpha}{\sqrt{t}}\mathbf{g}(\mathbf{w}_t), \mathbf{w}_{t+1} - \mathbf{w} + (t+1)(\mathbf{w}_{t+1} - \mathbf{w})\rangle \leq 0.$$

i.e.,

$$\langle \mathbf{w}_{t+1} + \mathbf{p}_{t+1} - (\mathbf{w}_t + \mathbf{p}_t) + \frac{\alpha}{\sqrt{t}}\mathbf{g}(\mathbf{w}_t), \mathbf{w}_{t+1} + \mathbf{p}_{t+1} - \mathbf{w}\rangle \leq 0.$$

Using Lemma 6, Lemma 1 is proved.

Proof for Theorem 2. According to Lemma 1,

$$\|\mathbf{w}^* - (\mathbf{w}_{t+1} + \mathbf{p}_{t+1})\|^2 \leq \|\mathbf{w}^* - (\mathbf{w}_t + \mathbf{p}_t) + \frac{\alpha}{\sqrt{t}}\mathbf{g}(\mathbf{w}_t)\|^2.$$

$$\|\mathbf{w}^* - (\mathbf{w}_t + \mathbf{p}_t) + \frac{\alpha}{\sqrt{t}}\mathbf{g}(\mathbf{w}_t)\|^2$$
$$= \|\mathbf{w}^* - (\mathbf{w}_t + \mathbf{p}_t)\|^2 + \|\frac{\alpha}{\sqrt{t}}\mathbf{g}(\mathbf{w}_t)\|^2 + 2\langle \frac{\alpha}{\sqrt{t}}\mathbf{g}(\mathbf{w}_t), \mathbf{w}^* - \mathbf{w}_t\rangle + 2\langle \frac{\alpha t}{\sqrt{t}}\mathbf{g}(\mathbf{w}_t), \mathbf{w}_{t-1} - \mathbf{w}_t\rangle$$

Note

$$\langle \mathbf{g}(\mathbf{w}_t), \mathbf{w}^* - \mathbf{w}_t\rangle \leq f(\mathbf{w}^*) - f(\mathbf{w}_t),\ \langle \mathbf{g}(\mathbf{w}_t), \mathbf{w}_{t-1} - \mathbf{w}_t\rangle \leq f(\mathbf{w}_{t-1}) - f(\mathbf{w}_t).$$

Then

$$(t+1)(f(\mathbf{w}_t) - f(\mathbf{w}^*))$$
$$\leq t(f(\mathbf{w}_{t-1}) - f(\mathbf{w}^*)) + \frac{\sqrt{t}}{2\alpha}\|\mathbf{w}^* - (\mathbf{w}_t + \mathbf{p}_t)\|^2 - \frac{\sqrt{t}}{2\alpha}\|\mathbf{w}^* - (\mathbf{w}_{t+1} + \mathbf{p}_{t+1})\|^2 + \frac{\alpha}{2\sqrt{t}}\|\mathbf{g}(\mathbf{w}_t)\|^2.$$

Summing this inequality from $k = 1$ to t, we obtain

$$(t+1)(f(\mathbf{w}_t) - f(\mathbf{w}^*))$$
$$\leq f(\mathbf{w}_0) - f(\mathbf{w}^*) + \sum_{k=1}^{t}\frac{\alpha}{2\sqrt{k}}\|\mathbf{g}(\mathbf{w}_k)\|^2 + \sum_{k=1}^{t}\big[\frac{\sqrt{k}}{2\alpha}(\|\mathbf{w}^* - (\mathbf{w}_k + \mathbf{p}_k)\|^2 - \|\mathbf{w}^* - (\mathbf{w}_{k+1} + \mathbf{p}_{k+1})\|^2)\big].$$

Note

$$\sum_{k=1}^{t}\frac{1}{2\sqrt{k}}\|\mathbf{g}(\mathbf{w}_k)\|^2 \leq \sqrt{t}M^2.$$

and

$$\sum_{k=1}^{t}\big[\frac{\sqrt{k}}{2}(\|\mathbf{w}^* - (\mathbf{w}_k + \mathbf{p}_k)\|^2 - \|\mathbf{w}^* - (\mathbf{w}_{k+1} + \mathbf{p}_{k+1})\|^2)\big].$$
$$\leq \frac{1}{2}\|\mathbf{w}^* - (\mathbf{w}_1 + \mathbf{p}_1)\|^2 - \frac{\sqrt{t}}{2}\|\mathbf{w}^* - (\mathbf{w}_{t+1} + \mathbf{p}_{t+1})\|^2 + \sum_{k=2}^{t}(\frac{\sqrt{k}}{2} - \frac{\sqrt{k-1}}{2})\|\mathbf{w}^* - (\mathbf{w}_k + \mathbf{p}_k)\|^2.$$

Since $\mathbf{Q}$ is a bounded set, there exists a positive number $M_0 > 0$ such that

$$\|\mathbf{w}^* - (\mathbf{w}_{t+1} + \mathbf{p}_{t+1})\|^2 \leq M_0, \forall t \geq 0.$$

Therefore

$$(t+1)[f(\mathbf{w}_t) - f(\mathbf{w}^*)] \leq f(\mathbf{w}_0) - f(\mathbf{w}^*) + \alpha\sqrt{t}M^2 + \frac{\sqrt{t}}{2\alpha}M_0.$$

This completes the proof of Theorem 2.

REFERENCES

Ahmet Alacaoglu, Yura Malitsky, Panayotis Mertikopoulos, and Volkan Cevher. A new regret analysis for adam-type algorithms. In *Proceedings of the International Conference on Machine Learning*, 2020.

Xi Chen, Qihang Lin, and Javier Pena. Optimal regularized dual averaging methods for stochastic optimization. In *Advances in Neural Information Processing Systems*, pp. 395–403, 2012.

Xiangyi Chen, Sijia Liu, Ruoyu Sun, and Mingyi Hong. On the convergence of a class of adam-type algorithms for non-convex optimization. In *International Conference on Learning Representations*, 2019.

Bertsekas Dimitri P., Nedić Angelia., and Ozdaglar Asuman E. *Convex analysis and optimization*. Athena Scientific, 2003.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(7), 2011.

John C Duchi. Introductory lectures on stochastic optimization. *The mathematics of data*, 25:99, 2018.

Euhanna Ghadimi, Hamid Reza Feyzmahdavian, and Mikael Johansson. Global convergence of the heavy-ball method for convex optimization. In *2015 European Control Conference (ECC)*, pp. 310–315. IEEE, 2015.

Igor Gitman, Hunter Lang, Pengchuan Zhang, and Lin Xiao. Understanding the role of momentum in stochastic gradient methods. In *Advances in Neural Information Processing Systems*, pp. 9630–9640, 2019.

Nicholas JA Harvey, Christopher Liaw, Yaniv Plan, and Sikander Randhawa. Tight analyses for non-smooth stochastic gradient descent. In *Annual Conference on Learning Theory*, pp. 1579–1613, 2019.

Chonghai Hu, Weike Pan, and James T Kwok. Accelerated gradient methods for stochastic optimization and online learning. In *Advances in Neural Information Processing Systems*, pp. 781–789, 2009.

Prateek Jain, Dheeraj Nagaraj, and Praneeth Netrapalli. Making the last iterate of sgd information theoretically optimal. In *Annual Conference on Learning Theory*, pp. 1752–1755, 2019.

Diederik P Kingma and Jimmy Ba. The unusual effectiveness of averaging in gan training. In *International Conference on Learning Representations*, 2014.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.

Chaoyue Liu and Mikhail Belkin. Accelerating sgd with momentum for over-parameterized learning. In *International Conference on Learning Representations*, 2020.

Mahesh Chandra Mukkamala and Matthias Hein. Variants of rmsprop and adagrad with logarithmic regret bounds. In *International Conference on Machine Learning*, 2017.

Arkadi Semenovich Nemirovsky and David Borisovich Yudin. Problem complexity and method efficiency in optimization. *John Wiley & Sons*, 1983.

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

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

Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

Alexander Rakhlin, Ohad Shamir, and Karthik Sridharan. Making gradient descent optimal for strongly convex stochastic optimization. *arXiv preprint arXiv:1109.5647*, 2011.

Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. In *International Conference on Learning Representations*, 2018.

Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

Ohad Shamir. Open problem: Is averaging needed for strongly convex stochastic gradient descent? In *Anual Conference on Learning Theory*, pp. 1–3, 2012.

Tao Sun, Dongsheng Li, Zhe Quan, Hao Jiang, Shengguo Li, and Yong Dou. Heavy-ball algorithms always escape saddle points. *arXiv preprint arXiv:1907.09697*, 2019.

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

Wei Tao, Zhisong Pan, Gaowei Wu, and Qing Tao. The strength of nesterov's extrapolation in the individual convergence of nonsmooth optimization. *IEEE Transactions on Neural Networks and Learning Systems*, 31(7):2557–2568, 2020a.

Wei Tao, Zhisong Pan, Gaowei Wu, and Qing Tao. Primal averaging: A new gradient evaluation step to attain the optimal individual convergence. *IEEE Transactions on Cybernetics*, 50(2):835–845, 2020b.

Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop, coursera: Neural networks for machine learning. *University of Toronto, Technical Report*, 2012.

Guanghui Wang, Shiyin Lu, Weiwei Tu, and Lijun Zhang. Sadam: A variant of adam for strongly convex functions. In *International Conference on Learning Representations*, 2020.

Ashia C Wilson, Rebecca Roelofs, Mitchell Stern, Nati Srebro, and Benjamin Recht. The marginal value of adaptive gradient methods in machine learning. In *Advances in neural information processing systems*, pp. 4148–4158, 2017.

Tianbao Yang, Qihang Lin, and Zhe Li. Unified convergence analysis of stochastic momentum methods for convex and non-convex optimization. *arXiv preprint arXiv:1604.03257*, 2016.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the International Conference on Machine Learning*, pp. 928–936, 2003.

Fangyu Zou, Li Shen, Zequn Jie, Ju Sun, and Wei Liu. Weighted adagrad with unified momentum. *arXiv preprint arXiv:1808.03408*, 2018.