

Nostalgic Adam: Weighting More of the Past Gradients When Designing the Adaptive Learning Rate

Haiwen Huang¹, Chang Wang¹ and Bin Dong^{2,3,4}

¹School of Mathematical Sciences, Peking University

²Beijing International Center for Mathematical Research, Peking University

³Center for Data Science, Peking University

⁴Laboratory for Biomedical Image Analysis, Beijing Institute of Big Data Research, Beijing
 {smshhw, 1500010660}@pku.edu.cn, dongbin@math.pku.edu.cn

Abstract

First-order optimization algorithms have been proven prominent in deep learning. In particular, algorithms such as RMSProp and Adam are extremely popular. However, recent works have pointed out the lack of “long-term memory” in Adam-like algorithms, which could hamper their performance and lead to divergence. In our study, we observe that there are benefits of weighting more of the past gradients when designing the adaptive learning rate. We therefore propose an algorithm called the Nostalgic Adam (NosAdam) with theoretically guaranteed convergence at the best known convergence rate. NosAdam can be regarded as a fix to the non-convergence issue of Adam in alternative to the recent work of [Reddi *et al.*, 2018]. Our preliminary numerical experiments show that NosAdam is a promising alternative algorithm to Adam. The proofs, code, and other supplementary materials are already released¹.

1 Introduction

Along with the rise of deep learning, various first-order stochastic optimization methods emerged. Among them, the most fundamental one is the stochastic gradient descent, and the Nesterov’s Accelerated Gradient method [Nesterov, 1983] is also a well-known acceleration algorithm. Recently, many adaptive stochastic optimization methods have been proposed, such as AdaGrad [Duchi *et al.*, 2010], RMSProp [Tieleman and Hinton, 2012], AdaDelta [Zeiler, 2012] and Adam [Kingma and Ba, 2014]. These algorithms can be written in the following general form:

$$x_{t+1} = x_t - \frac{\alpha_t}{\psi(g_1, \dots, g_t)} \phi(g_1, \dots, g_t), \quad (1)$$

where g_i is the gradient obtained in the i -th time step, $\alpha_t/\psi(g_1, \dots, g_t)$ the adaptive learning rate, and $\phi(g_1, \dots, g_t)$ the gradient estimation. There have been extensive studies on the design of gradient estimations which can be traced back to

classical momentum methods [Polyak, 1964] and NAG [Nesterov, 1983]. In this paper, however, we focus more on how to understand and improve the adaptive learning rate.

Adam [Kingma and Ba, 2014] is perhaps the most widely used adaptive stochastic optimization method which uses an exponential moving average (EMA) to estimate the square of the gradient scale, so that the learning rate can be adjusted adaptively. More specifically, Adam takes the form of (1) with

$$\begin{aligned} \psi(g_1, \dots, g_t) &= \sqrt{V_t}, V_t = \text{diag}(v_t) \\ v_t &= \beta_2 v_{t-1} + (1 - \beta_2) g_t^2. \end{aligned} \quad (2)$$

Here in g_t^2 , and other multiplication operations in this paper, coordinate-wise multiplication is performed. We shall call v_t the re-scaling term of the Adam and its variants, since it serves as a coordinate-wise re-scaling of the gradients. Despite its fast convergence and easiness in implementation, Adam is also known for its non-convergence and poor generalization in some cases [Reddi *et al.*, 2018][Wilson *et al.*, 2017]. More recently, [Balles and Hennig, 2018] both theoretically and empirically pointed out that generalization is mainly determined by the sign effect rather than the adaptive learning rate, and the sign effect is problem-dependent. In this paper, we are mainly dealing with the non-convergence issue and will only empirically compare generalization ability among different Adam variants.

As for the non-convergence issue, [Reddi *et al.*, 2018] suggested that the EMA of v_t of Adam is the cause. The main problem lies in the following quantity:

$$\Gamma_{t+1} = \frac{\sqrt{V_{t+1}}}{\alpha_{t+1}} - \frac{\sqrt{V_t}}{\alpha_t},$$

which essentially measures the change in the inverse of learning rate with respect to time. Algorithms that use EMA to estimate the scale of the gradients cannot guarantee the positive semi-definiteness of Γ_t , and that causes the non-convergence of Adam. To fix this issue, [Reddi *et al.*, 2018] proposed *AMSGrad*, which added one more step $\hat{v}_t = \max\{\hat{v}_{t-1}, v_t\}$ in (2). *AMSGrad* is claimed by its authors to have a “long-term memory” of past gradients.

Another explanation on the cause of non-convergence was recently proposed by [Zhou *et al.*, 2018]. The authors observed that Adam may diverge because a small gradient may

¹<https://github.com/andrehuang/NostalgicAdam-NosAdam>

have a large step size which leads to a large update. Therefore, if the small g_t with large step size is often in the wrong direction, it could lead to divergence. Thus, they proposed a modification to Adam called *AdaShift* by replacing g_t^2 with g_{t-n}^2 for some manually chosen n when calculating v_t .

Both *AdaShift* and *AMSGrad* suggest that we should not fully trust the gradient information we acquire at current step, and the past gradients are useful when g_t is not reliable. In this paper, we take this idea one step further by suggesting that we may weight *more* of the past gradients than the present ones. We call our algorithm *Nostalgic Adam* (*NosAdam*). We will show that the design of the algorithm is inspired by our mathematical analysis on the convergence, and *NosAdam* has the fastest known convergence rate. Furthermore, we will discuss why “nostalgia” is important, and empirically investigate how different designs of v_t can lead to different performances from a loss landscape perspective. Finally, we examine the empirical performance of *NosAdam* on some common machine learning tasks. The experiments show us that *NosAdam* is a promising alternative to Adam and its variants.

2 Related Work

Adam is widely used in both academia and industry. However, it is also one of the least well-understood algorithms. In recent years, some remarkable works provided us with better understanding of the algorithm, and proposed different variants of it. Most of works focused on how to interpret or modify the re-scaling term v_t of (2).

As mentioned above, [Reddi *et al.*, 2018], [Zhou *et al.*, 2018] focused on the non-convergence issue of Adam, and proposed their own modified algorithms. More recently, [Luo *et al.*, 2019] proposed another modification. [Wilson *et al.*, 2017] pointed out the generalization issue of adaptive optimization algorithms. Based on the assumption that v_t is the estimate of the second moment estimate of g_t , [Balles and Hennig, 2018] dissected Adam into sign-based direction and variance adaption magnitude. They also pointed out that the sign-based direction part is the decisive factor of generalization performance, and that is problem-dependent. This in a way addressed the generalization issue raised in [Wilson *et al.*, 2017].

However, the interpretation of v_t as an estimate of the second moment assumption may not be correct, since [Chen and Gu, 2019] showed that $v_t^{1/2}$ in the Adam update (2) can be replaced by v_t^p for any $p \in (0, \frac{1}{2}]$. The modified algorithm is called *Padam*. In our supplementary material, we also proved that a convergence theorem of a “p-norm” form of *NosAdam*, where the re-scaling term v_t can be essentially viewed as a “p-moment” of g_t . These discoveries cast doubts on the second moment assumption, since both the convergence analysis and empirical performance seemed not so dependent on this assumption.

The true role of v_t , however, remains a mystery. In *AdaGrad* [Duchi *et al.*, 2010], which is a special case of *NosAdam*, the authors mentioned an metaphor that “the adaptation allows us to find needles in haystacks in the form of very predictive but rarely seen features.” They were suggesting that v_t is to some extent balancing the update speeds of

different features according to their abundance in the data set. This understanding might be supported by a previous work called *SWATS* (Switching from Adam to SGD) [Keskar and Socher, 2017], which uses Adam for earlier epochs and then fix the re-scaling term v_t for later epochs. This suggests that there may be some sort of optimal re-scaling term, and we can keep using it after we obtain a good enough estimate.

Despite all the previous efforts, our understanding of the re-scaling term v_t is still very limited. In this paper, we investigate the issue from a loss landscape approach, and this provides us with some deeper understanding of when and how different Adam-like algorithms can perform well or poorly.

3 Nostalgic Adam Algorithm

In this section, we introduce the *Nostalgic Adam* (*NosAdam*) algorithm, followed by a discussion on its convergence. Let us first consider a general situation where we allow the parameter β_2 in Equation (2) change in time t . Without loss of generality, we may let $\beta_{2,t} = \frac{B_{t-1}}{B_t}$ and $B_0 = 0$. Considering the constraint that $0 \leq \beta_{2,t} \leq 1$, $\{B_t\}$ is assumed to be a non-decreasing positive sequence. We then define $b_t := B_t - B_{t-1}$, then $B_t = \sum_{k=1}^t b_k$ and $\{b_k\}$ is a non-negative sequence.

Like Adam and its variants, the condition $\Gamma_t \succeq 0$ is crucial in ensuring convergence. We will now see that to ensure positive semi-definiteness of Γ_t , the algorithm naturally requires to weight more of the past gradients than the more recent ones when calculating v_t . To see this, we first present the following lemma.

Lemma 3.1. *The positive semi-definiteness of $\frac{V_t}{\alpha_t^2} - \frac{V_{t-1}}{\alpha_{t-1}^2}$ is satisfied if and only if $\frac{B_t}{t}$ is non-increasing.*

Proof.

$$\begin{aligned} \frac{V_t}{\alpha_t^2} &= \frac{t}{\alpha^2} \sum_{j=1}^t \prod_{k=1}^{t-j} \beta_{2,t-k+1} (1 - \beta_{2,j}) g_j^2 \\ &= \frac{t}{\alpha^2} \sum_{j=1}^t \frac{B_{t-1}}{B_t} \cdots \frac{B_j}{B_{j+1}} \frac{B_j - B_{j-1}}{B_j} g_j^2 \\ &= \frac{t}{B_t \alpha^2} \sum_{j=1}^t b_j g_j^2 \geq \frac{t-1}{B_{t-1} \alpha^2} \sum_{j=1}^{t-1} b_j g_j^2 \\ &= \frac{V_{t-1}}{\alpha_{t-1}^2} \end{aligned}$$

□

Here the “if and only if” in the lemma is due to that if $\frac{B_t}{t}$ is non-increasing, then because $\sum_{j=1}^t b_j g_j^2 \geq \sum_{j=1}^{t-1} b_j g_j^2$, we can derive the positive semi-definiteness of $\frac{V_t}{\alpha_t^2} - \frac{V_{t-1}}{\alpha_{t-1}^2}$; and if $\frac{V_t}{\alpha_t^2} - \frac{V_{t-1}}{\alpha_{t-1}^2} \succeq 0$ for every $\{g_j\}$, then for $\{g_j\}$ whose limit is 0, we can derive that $\frac{B_t}{t}$ is non-increasing from the above inequality.

Then, if we assume that $\{b_k\}$ is an monotonous sequence (note in Adam, $\{b_k\}$ is a decreasing sequence), it is not hard

Algorithm 1 Nostalgic Adam Algorithm

Input: $x \in F$, $m_0 = 0$, $V_0 = 0$

- 1: **for** $t = 1$ **to** T **do**
- 2: $g_t = \nabla f_t(x_t)$
- 3: $\beta_{2,t} = B_{t-1}/B_t$, where $B_t = \sum_{k=1}^t b_k$ for $t \geq 1$, $b_k \geq 0$ and $B_0 = 0$
- 4: $m_t = \beta_1 m_{t-1} + (1 - \beta_1)g_t$
- 5: $V_t = \beta_{2,t}V_{t-1} + (1 - \beta_{2,t})g_t^2$
- 6: $\hat{x}_{t+1} = x_t - \alpha_t m_t / \sqrt{V_t}$
- 7: $x_{t+1} = \mathcal{P}_{\mathcal{F}, \sqrt{V_t}}(\hat{x}_{t+1})$
- 8: **end for**

to see that $\frac{B_t}{t}$ is non-increasing if and only if b_j is non-increasing. Noting that $v_t = \sum_{k=1}^t g_k^2 \frac{b_k}{B_t}$, we can see that the sufficient condition for positive semi-definiteness of Γ_t is that in the weighted average v_t , the weights of gradients should be non-increasing w.r.t. t . In other words, we should weight more of the past gradients than the more recent ones.

The NosAdam algorithm reads as in Algorithm 1. We now introduce the necessary notations and definitions in our presentation of the algorithm and convergence theorem. We have adopted the notation of online optimization introduced in [Zinkevich, 2003]. At each time step t , the optimization algorithm picks a point x_t in its feasible set $\mathcal{F} \in \mathbb{R}^d$. Let f_t be the loss function corresponding to the underlying mini-batch, and the algorithm incurs loss $f_t(x_t)$. We evaluate our algorithm using the regret that is defined as the sum of all the previous differences between the online prediction $f_t(x_t)$ and loss incurred by the fixed parameter point in \mathcal{F} for all the previous steps, i.e.

$$R_T = \sum_{t=1}^T f_t(x_t) - \min_{x \in \mathcal{F}} \sum_{t=1}^T f_t(x). \quad (3)$$

Denote \mathcal{S}_d^+ the set of all positive definite $d \times d$ matrices. The projection operator $\mathcal{P}_{\mathcal{F}, A}(y)$ for $A \in \mathcal{S}_d^+$ is defined as $\operatorname{argmin}_{x \in \mathcal{F}} \|A^{1/2}(x - y)\|$ for $y \in \mathbb{R}^d$. Finally, we say \mathcal{F} has bounded diameter D_∞ if $\|x - y\|_\infty \leq D_\infty$ for all $x, y \in \mathcal{F}$.

From Algorithm 1, we can see that v_t can either decrease or increase based on the relationship between v_{t-1} and g_t^2 , which is the reason why NosAdam circumvents the flaw of AMSGrad (Figure 4). Convergence of NosAdam is also guaranteed as stated by the following theorem.

Theorem 3.2 (Convergence of NosAdam). *Let B_t and b_k be the sequences defined in Algorithm 1, $\alpha_t = \alpha/\sqrt{t}$, $\beta_{1,1} = \beta_1$, $\beta_{1,t} \leq \beta_1$ for all t . Assume that \mathcal{F} has bounded diameter D_∞ and $\|\nabla f_t(x)\|_\infty \leq G_\infty$ for all t and $x \in \mathcal{F}$. Furthermore, let $\beta_{2,t}$ be such that the following conditions are satisfied:*

1. $\frac{B_t}{t} \leq \frac{B_{t-1}}{t-1}$
2. $\frac{B_t}{tb_t^2} \geq \frac{B_{t-1}}{(t-1)b_{t-1}^2}$

Then for $\{x_t\}$ generated using NosAdam, we have the fol-

lowing bound on the regret

$$R_T \leq \frac{D_\infty^2}{2\alpha(1-\beta_1)} \sum_{i=1}^d \sqrt{T} v_{T,i}^{\frac{1}{2}} + \frac{D_\infty^2}{2(1-\beta_1)} \sum_{t=1}^T \sum_{i=1}^d \frac{\beta_{1,t} v_{t,i}^{\frac{1}{2}}}{\alpha_t} + \frac{\alpha\beta_1}{(1-\beta_1)^3} \sum_{i=1}^d \sqrt{\frac{B_T}{T} \frac{\sum_{t=1}^T b_t g_{t,i}^2}{b_T^2}}$$

One notable characteristic of NosAdam, which makes it rather different from the analysis by [Reddi *et al.*, 2018], is that the conditions on B_t and b_t are data-independent and are very easy to check. In particular, if we choose B_t as a hyperharmonic series, i.e. $B_t = \sum_{k=1}^t k^{-\gamma}$, then the convergence criteria are automatically satisfied. We shall call this special case NosAdam-HH (HyperHarmonic), and its convergence result is summarized in the following corollary.

Corollary 3.2.1. *Suppose $\beta_{1,t} = \beta_1 \lambda^{t-1}$, $b_k = k^{-\gamma}$, $\gamma \geq 0$, thus $B_t = \sum_{k=1}^t k^{-\gamma}$, and $\beta_{2,t} = B_{t-1}/B_t < 1$ in Algorithm 1. Then B_t and b_t satisfy the constraints in Theorem 3.2, and we have*

$$R_T \leq \frac{D_\infty^2}{2\alpha(1-\beta_1)} \sum_{i=1}^d \sqrt{T} v_{T,i}^{\frac{1}{2}} + \frac{D_\infty^2 G_\infty \beta_1}{2(1-\beta_1)} \frac{1}{(1-\lambda)^2} \cdot d + \frac{2\alpha\beta_1}{(1-\beta_1)^3} G_\infty \sqrt{T}$$

To see this, we simply need to see that $B_t = O(t^{1-\gamma})$. More detailed explanation is put in our supplement. Our theory shows that the proposed NosAdam achieves convergence rate of $O(1/\sqrt{T})$, which is so far the best known convergence rate.

4 Why Nostalgic?

In this section, we investigate more about the mechanism behind Adam and AMSGrad, and analyze the pros and cons of being “nostalgic”.

As mentioned in Section 1, [Reddi *et al.*, 2018] proved that if Γ_t is positive semi-definite, Adam converges. Otherwise, it may diverge. An example of divergence made by [Reddi *et al.*, 2018] is

$$f_t(x) = \begin{cases} Cx & t \bmod 3=1 \\ -x & \text{otherwise} \end{cases}, \quad (4)$$

where C is slightly larger than 2. The correct optimization direction should be -1, while Adam would go towards 1. To fix this, they proposed AMSGrad, which ensures $\Gamma_t \succeq 0$ by updating v_t as follows

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2, \\ \hat{v}_t = \max(\hat{v}_{t-1}, v_t),$$

where \hat{v}_t is used in the update step.

However, this example is not representative of real situations. Also, the explanation of “long-term memory” by [Reddi *et al.*, 2018] is not very illustrative. In the remaining part of this section, we aim to discuss some more realistic

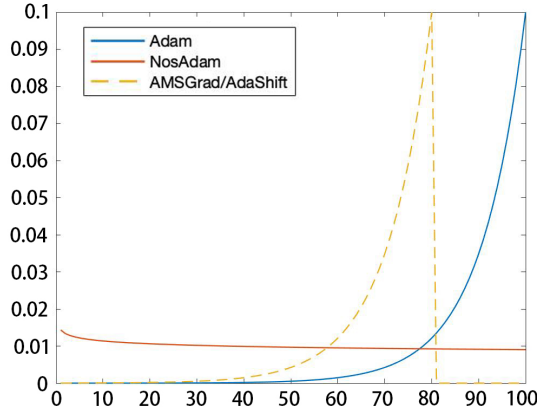


Figure 1: Weight comparison among Adam, NosAdam and AMSGrad. The y -axis shows the weight of each step, and the x -axis shows the number of steps

scenarios and try to understand the pros and cons of different algorithms.

We start from analyzing the different weighting strategies when calculating v_t . For Adam,

$$v_t^{(\text{Adam})} = \sum_{k=1}^t (1 - \beta_2) \beta_2^{t-k} g_k^2,$$

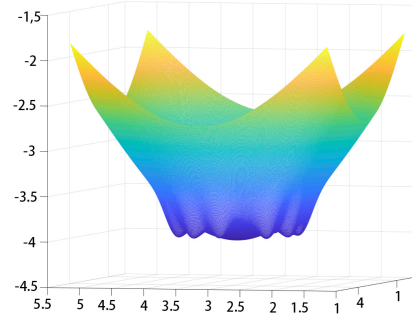
and the weight $(1 - \beta_2) \beta_2^{t-k}$ increases exponentially. For NosAdam,

$$v_t^{(\text{NosAdam})} = \sum_{k=1}^t g_k^2 \frac{b_k}{B_t},$$

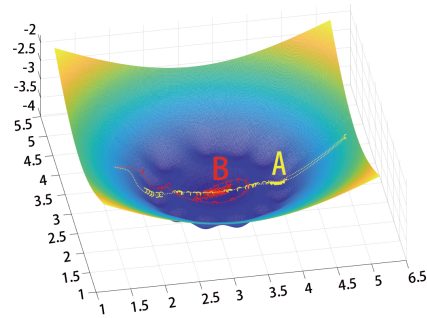
and for NosAdam-HH, $b_k = k^{-\gamma}$ is the k -th term of a hyperharmonic series. For AMSGrad, $v_t^{(\text{AMSGrad})}$ is data-dependent and therefore cannot be explicitly expressed. However, $v_t^{(\text{AMSGrad})}$ is chosen to be the largest in $\{v_s^{(\text{Adam})} : 0 \leq s \leq t\}$. Therefore, it can be seen as a shifted version of $v_t^{(\text{Adam})}$, i.e. $v_s^{(\text{Adam})} = v_{t-n}^{(\text{Adam})}$, where n depends on the data. This is similar as AdaShift, where n is instead a hyperparameter. Figure 1 plots the first 100 weights of Adam, NosAdam and AMSGrad, where β_2, γ, n , is chosen as 0.9, 0.1 and 20, respectively.

From the above analysis, we can see that v_t of Adam is mainly determined by its most current gradients. Therefore, when g_t keeps being small, the adaptive learning rate could be large, which may lead to oscillation of the sequence, and increasing chance of being trapped in local minimum. On the other hand, NosAdam adopts a more stable calculation of v_t , since it relies on all the past gradients.

We support the above discussion with an example of an objective function with a bowl-shaped landscape where the global minima is at the bottom of the bowl with lots of local minimum surrounding it. The explicit formula of the objec-



(a) Bowl-shaped Landscape



(b) Trajectories of NosAdam (red) and Adam (yellow)

Figure 2: Example of an objective function with a bowl-shaped landscape. It has a wide global minima and some sharp local minimum surrounding it. In 2b, the red trajectory is the sequence generated by NosAdam and yellow trajectory by Adam. The trajectory of Adam ends up in valley A and NosAdam in valley B. This shows that Adam would easily diverge due to unstable calculations of v_t .

tive function is

$$f(x, y, z) = -ae^{-b((x-\pi)^2+(y-\pi)^2)+(z-\pi)^2} - c \sum_i \cos(x)\cos(y)e^{-\beta((x-r\sin(\frac{i}{2})-\pi)^2+(y-r\cos(\frac{i}{2})-\pi)^2)}.$$

Figure 2a shows one slice of the function for $z = 2.34$. In the function, a and b determine the depth and width of the global minima, and c, r, β determine depth, location and width of the local minimums. In this example, a, b, c, r, β are set to 30, 0.007, 0.25, 1, 20, respectively. Figure 2b shows different trajectories of Adam and NosAdam when they are initiated at the same point on the side of the bowl. As expected, the trajectory of Adam (yellow) passes the global minima and ends up trapped in valley A, while NosAdam (red) gradually converges to the global minima, i.e. valley B.

With the operation of taking max, AMSGrad does not have the same non-convergence issue as discussed above. However, taking max may be problematic as well since \hat{v}_t can never decrease. If a very large gradient appears at an iteration, then the adaptive learning rate for all later steps will keep being small. For example, if a large gradient (e.g. 100 times of the original gradient) appears at the 10^6 step in the example

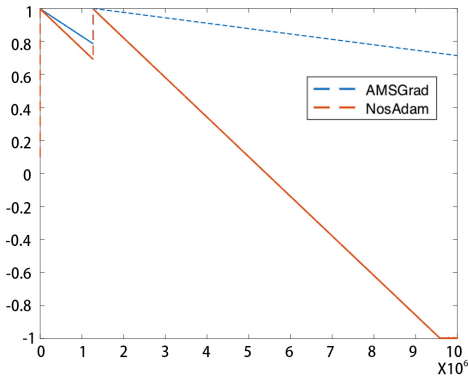


Figure 3: Appearance of a large gradient at around 10^6 step. The y -axis shows the value x , and the x -axis shows the number of iterations. The figure shows AMSGrad is greatly slowed down after encountering a large gradient.

Equation (4), we can see that AMSGrad will converge very slowly. This, however, will not be a problem for NosAdam which has the ability of both increasing and decreasing its v_t . See Figure 3 for a demonstration. Another example with sharper local minima by setting $b = 2, c = 4, r = 1.3$ is given in Figure 4; and the algorithms are initialized at location A. One can see that the sequence generated by AMSGrad is trapped in a sharp local minima, whereas NosAdam still converges to the global minimum. From these examples we can see that the operation of taking max of AMSGrad has some intrinsic flaws though it promises convergence. The way of computing v_t in NosAdam seems superior.

There are also situations in which NosAdam can work poorly. Just because NosAdam is nostalgic, it requires a relatively good initial point to achieve good performances though this is commonly required by most optimization algorithms. However, Adam can be less affected by bad initializations sometime due to its specific way of calculating v_t . This gives it a chance of jumping out of the local minimum (and a chance of jumping out of the global minima as well as shown in Figure 2). To demonstrate this, we let both Adam and NosAdam initialize in the valley A (see Figure 5). We can see that the trajectory of Adam manages to jump out of the valley, while it is more difficult for NosAdam to do so.

We note that although NosAdam requires good initialization, it does not necessarily mean initializing near the global minima. Since the algorithm is nostalgic, as long as the initial gradients are pointing towards the right direction, the algorithm may still converge to the global minima even though the initialization is far away from the global minima. As we can see from Figure 4 that NosAdam converges because all of the gradients are good ones at the beginning of the algorithm, which generates enough momentum to help the sequence dashes through the region with sharp local minimum.

Like any Adam-like algorithm, the convergence of NosAdam depends on the loss landscape and initialization. However, if the landscape is as shown in the above figures, then NosAdam has a better chance to converge than Adam and AMSGrad. In practice, it is therefore helpful to first ex-

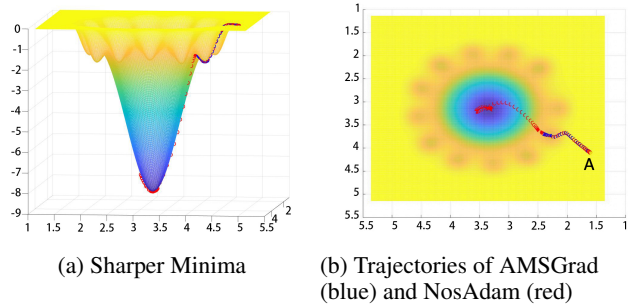


Figure 4: Figure 4a shows the landscape of the objective function. Figure 4b shows the different behaviours of AMSGrad and NosAdam, with the sequence generated by AMSGrad colored in blue and NosAdam in red, and they are initiated at location A.

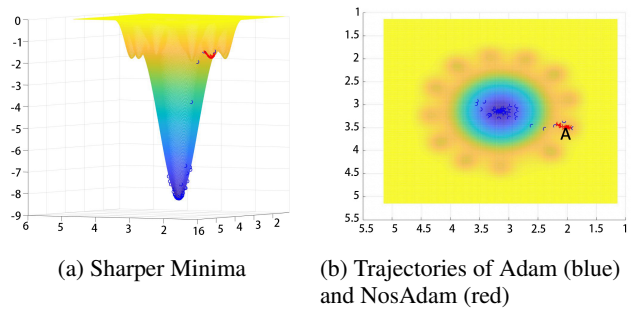
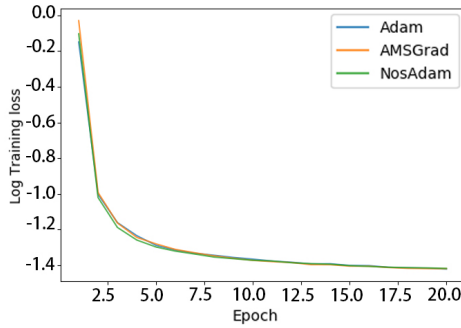


Figure 5: Figure 5a shows the landscape of the objective function. Figure 5b shows the different behaviours of Adam and NosAdam, with the sequence generated by Adam colored in blue and NosAdam in red, and they are initiated in the valley A.

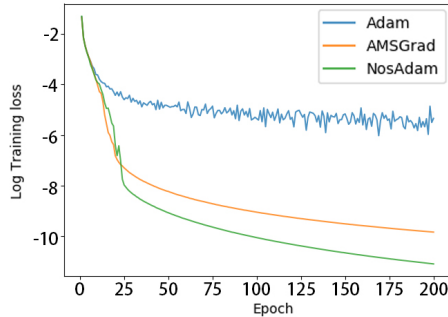
amine the loss landscape before selecting an algorithm. However, it is time consuming to do in general. Nonetheless, earlier studies showed that neural networks with skip connections like ResNet and DenseNet lead to coercive loss functions similar to the one shown in the above figures [Li *et al.*, 2018].

5 Experiments

In this section, we conduct some experiments to compare NosAdam with Adam and its variant AMSGrad. We consider the task of multi-class classification using logistic regression, multi-layer fully connected neural networks and deep convolutional neural networks on MNIST [Lecun,] and CIFAR-10 [Krizhevsky *et al.*,]. The results generally indicate that NosAdam is a promising algorithm that works well in practice. Throughout our experiments, we fixed β_1 to be 0.9, β_2 to be 0.999 for Adam and AMSGrad, and search γ in $\{1e-1, 1e-2, 1e-3, 1e-4\}$ for NosAdam. The initial learning rate is chosen from $\{1e-3, 2e-3, \dots, 9e-3, 1e-2, 2e-2, \dots, 9e-2, 1e-1, 2e-1, \dots, 9e-1\}$ and the results are reported using the best set of hyperparameters. All the experiments are done using Pytorch0.4.



(a) Logistic Regression



(b) Multi-layer Fully Connected Neural Network

Figure 6: Experiments of logistic regression and multi-layer fully connected neural network on MNIST.

Logistic Regression

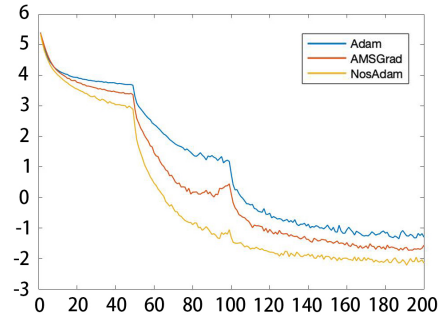
To investigate the performance of the algorithms on convex problems, we evaluate Adam, AMSGrad and NosAdam on multi-class logistic regression problem using the MNIST dataset. To be consistent with the theory, we set the step size $\alpha_t = \alpha/\sqrt{t}$. We set the minibatch size to be 128. According to Figure 6a, the three algorithms have very similar performance.

Multilayer Fully Connected Neural Networks

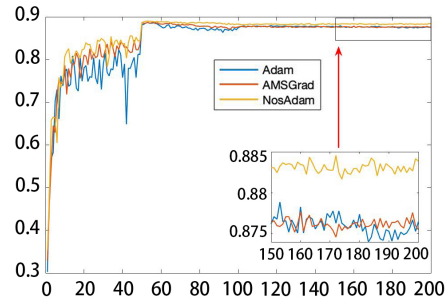
We first train a simple fully connected neural network with 1 hidden layer (with 100 neurons and ReLU as the activation function) for the multi-class classification problem on MNIST. We use constant step size $\alpha_t = \alpha$ and minibatch=128 throughout the experiments for this set of experiments. The results are shown in Figure 6b. We can see that NosAdam slightly outperforms AMSGrad, while Adam is much worse than both NosAdam and AMSGrad and oscillates a lot. This is due to the difference of the definition of v_t for each algorithm: v_t in AMSGrad and NosAdam gradually becomes stationary and stays at a good re-scaling value; while v_t in Adam does not have such property.

Deep Convolutional Neural Networks

Finally, we train a deep convolutional neural network on CIFAR-10. Wide Residual Network [Zagoruyko and Komodakis, 2016] is known to be able to achieve high accuracy with much less layers than ResNet [He *et al.*, 2015]. In our



(a) Log Training Loss



(b) Test Accuracy

Figure 7: Experiments of Wide ResNet on CIFAR-10.

experiment, we choose Wide ResNet28. The model is trained on 4 GPUs with the minibatch size 100. The initial learning rate is decayed at epoch 50 and epoch 100 by multiplying 0.1. In our experiments, the optimal performances are usually achieved when the learning rate is around 0.02 for all the three algorithms. For reproducibility, an anonymous link of code will be provided in the supplementary material.

Our results are shown in Figure 7. We observe that NosAdam works slightly better than AMSGrad and Adam in terms of both convergence speed and generalization. This indicates that NosAdam is a promising alternative to Adam and its variants.

6 Conclusion

In this paper, we suggested that we should weight more of the past gradients when designing the adaptive learning rate. Our original intuition came from the convergence analysis of Adam-like algorithms. Based on such observation, we then proposed Nostalgic Adam (NosAdam), and provided convergence analysis. We also discussed the pros and cons of NosAdam comparing to Adam and AMSGrad, which gave us a better idea when NosAdam could be effective.

Acknowledgments

This work would not have existed without the support of BICMR and School of Mathematical Sciences, Peking University. Bin Dong is supported in part by Beijing Natural Science Foundation (Z180001).

References

- [Balles and Hennig, 2018] Lukas Balles and Philipp Hennig. Dissecting adam: The sign, magnitude and variance of stochastic gradients. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pages 413–422, 2018.
- [Chen and Gu, 2019] Jinghui Chen and Quanquan Gu. Padam: Closing the generalization gap of adaptive gradient methods in training deep neural networks, 2019.
- [Duchi *et al.*, 2010] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. Technical Report UCB/EECS-2010-24, EECS Department, University of California, Berkeley, Mar 2010.
- [He *et al.*, 2015] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.
- [Keskar and Socher, 2017] Nitish Shirish Keskar and Richard Socher. Improving generalization performance by switching from adam to SGD. *CoRR*, abs/1712.07628, 2017.
- [Kingma and Ba, 2014] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.
- [Krizhevsky *et al.*,] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research).
- [Lecun,] Y. Lecun. The mnist database of handwritten digits. <http://yann.lecun.com/exdb/mnist/>.
- [Li *et al.*, 2018] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In *Neural Information Processing Systems*, 2018.
- [Luo *et al.*, 2019] Liangchen Luo, Yuanhao Xiong, Yan Liu, and Xu Sun. Adaptive gradient methods with dynamic bound of learning rate. In *Proceedings of the 7th International Conference on Learning Representations*, New Orleans, Louisiana, May 2019.
- [Nesterov, 1983] Y. Nesterov. A method for unconstrained convex minimization problem with the rate of convergence $o(1/k^2)$. *Doklady AN USSR*, 269:543–547, 1983.
- [Polyak, 1964] Boris Polyak. Some methods of speeding up the convergence of iteration methods. 4:1–17, 12 1964.
- [Reddi *et al.*, 2018] Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. In *International Conference on Learning Representations*, 2018.
- [Tieleman and Hinton, 2012] T. Tieleman and G. Hinton. Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 2012.
- [Wilson *et al.*, 2017] Ashia C Wilson, Rebecca Roelofs, Mitchell Stern, Nati Srebro, and Benjamin Recht. The marginal value of adaptive gradient methods in machine learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4148–4158. Curran Associates, Inc., 2017.
- [Zagoruyko and Komodakis, 2016] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *CoRR*, abs/1605.07146, 2016.
- [Zeiler, 2012] Matthew D. Zeiler. ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701, 2012.
- [Zhou *et al.*, 2018] Zhiming Zhou, Qingru Zhang, Guan-song Lu, Hongwei Wang, Weinan Zhang, and Yong Yu. Adashift: Decorrelation and convergence of adaptive learning rate methods. *CoRR*, abs/1810.00143, 2018.
- [Zinkevich, 2003] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. *ICML’03*, pages 928–935. AAAI Press, 2003.