A Qualitative Study of the Dynamic Behavior of Adaptive Gradient Algorithms

Chao Ma ∗†, Lei Wu ‡, and Weinan E § ¶

1Department of Mathematics, Stanford University
2Program in Applied and Computational Mathematics, Princeton University
3Department of Mathematics, Princeton University

September 15, 2020

Abstract

The dynamic behavior of RMSprop and Adam algorithms is studied through a combination of careful numerical experiments and theoretical explanations. Three types of qualitative features are observed in the training loss curve: fast initial convergence, oscillations and large spikes. The sign gradient descent (signGD) algorithm, which is the limit of Adam when taking the learning rate to 0 while keeping the momentum parameters fixed, is used to explain the fast initial convergence. For the late phase of Adam, three different types of qualitative patterns are observed depending on the choice of the hyper-parameters: oscillations, spikes and divergence. In particular, Adam converges faster and smoother when the values of the two momentum factors are close to each other.

1 Introduction

Adaptive gradient algorithms [4, 11, 6, 9], especially RMSprop [11] and Adam [6], have demonstrated superior performance in training modern machine learning models, e.g. deep neural networks. Distinguished from the vanilla gradient descent (GD) or stochastic gradient descent (SGD), adaptive gradient algorithms use a coordinate-wise scaling of the update direction. The scaling factors are adaptively determined by using the past gradients [4], which makes the analysis of these algorithms much more challenging.

Recent theoretical efforts [9, 14, 13, 7, 2] have focused on establishing the convergence of adaptive gradient algorithms. However, these results are still unsatisfactory, since they cannot explain any of the particular features of these adaptive gradient algorithms. Moreover, all these results usually require taking the limit that the step size \( \eta_t \) goes to zero, e.g. \( \eta_t = 1/\sqrt{t} \). However, in practice, one usually starts with a large step size and only decays the step size

∗chaoma@stanford.edu
†The first two authors contribute equally.
‡leiwu@princeton.edu
§weinan@math.princeton.edu
¶Also at Beijing Institute of Big Data Research.
for several times during the training process. For the most of iterations, the step size is actually fixed. So it is interesting to see what happens for not only the convergence but also the behavior of whole training curve with a fixed step size. Figure 1 shows an example of full-batch Adam with a fixed step size. One can see that even with full batch, the loss curve does not decrease monotonically: Small oscillations and large spikes appear shortly after initialization. This complicated loss pattern, especially the large spikes, makes it difficult to pick a good stopping time.

![Figure 1](image)

Figure 1: The training loss of full-batch Adam for a multi-layer neural network model on CIFAR-10. The network has 3 hidden-layers with widths 256-256-128. The step size is fixed to be 0.001 and \((\beta_1, \beta_2) = (0.9, 0.999)\), the default values in PyTorch and TensorFlow. 2 classes are picked from CIFAR-10 with 1000 images in each class. Square loss function is used.

In addition to the learning rate, adaptive gradient algorithms also use extra hyper-parameters such as the second-order momentum factor for RMSprop and the first and second-order momentum factor for Adam. Though default values are provided in mainstream packages (e.g. \(\beta_1=0.9, \beta_2=0.999\) for Adam in PyTorch and TensorFlow), these default parameter are not necessarily optimal and changing these hyper-parameter values can drastically change the behavior and performance of the algorithms. One objective of this paper is to carry out a comprehensive study of the influence of these hyper-parameters.

**Contributions**  In this paper, we provide well-designed experiments to demystify the dynamic behavior of adaptive gradient algorithms. Specifically, our contributions are summarized as follows.

1. We identify three types of typical phenomena in the training process of these adaptive algorithms: initial fast convergence (sometimes even super-linear), small oscillations, and large spikes.

2. For RMSprop and Adam, if the learning rate decreases to zero while the momentum parameters are fixed, we show that the algorithms tends to signGD. For signGD, we prove finite-time convergence for strongly convex objective functions. These arguments together provide a partial explanation of the fast initial convergence of RMSprop and Adam, which could be the one of reasons behind the popularity of these algorithms.
3. We show that the spikes are caused by some instabilities of the algorithm at stationary points. For RMSprop on simple objective functions, we explicitly write down the limiting oscillating solution. For Adam, we classify the behavior into three different patterns in the space of the two momentum factors: the spike regime, the oscillation regime, and the divergence regime. Empirical results show that training is most stable in the “oscillation regime”, especially when \( \beta_1 \approx \beta_2 \).

**Notations** To make the notations more consistent, from now on we use \( \alpha \) to denote the second-order momentum factor in both Adam and RMSprop, and use \( \beta \) to denote the first-order momentum in Adam. The conventional nations \( \beta_1 \) and \( \beta_2 \) above for Adam will become \( \beta \) and \( \alpha \), respectively. For vectors \( u \) and \( v \), operations such as \( u^2 \), \( \sqrt{u} \) and \( u/v \) are understood to be element-wise.

## 2 Preliminaries

### 2.1 Adaptive gradient algorithms

Adaptive gradient algorithms are a family of optimization algorithms that use a coordinate-wise scaling of the update direction (gradient or gradient with momentum) according to the history of gradients. Many adaptive algorithms can be cast to the following form [3],

\[
\begin{align*}
\mathbf{m}_{t+1} &= h_t \nabla f(x_t) + r_t \mathbf{m}_t \\
\mathbf{v}_{t+1} &= p_t (\nabla f(x_t))^2 + q_t \mathbf{v}_t \\
x_{t+1} &= x_t - \eta \frac{\mathbf{m}_{t+1}}{\sqrt{\mathbf{v}_{t+1}} + \epsilon}
\end{align*}
\]

(1)

with different choice of \( h, r, p, q \). In [1], \( h, r, p, q \) are scalar functions of \( t \). For example, Adagrad [4] is recovered when \( h, p, q = 1 \) and \( r = 0 \), and RMSprop corresponds to the case when \( h = 1, r = 0, p = 1 - \alpha \) and \( q = \alpha \) for some constant \( \alpha \in (0, 1) \). Viewed from the dynamics of \( x \) alone, adaptive gradient algorithms usually have a “memory effect” due to the momentum terms. The strength of the memory depends on the momentum factors \( (h, r, p, q) \) and the learning rate. Because of their efficiency in training neural network models, these algorithms are extensively used. Readers are referred to [10] for a more thorough review of existing adaptive algorithms.

In this paper, we focus on RMSprop and Adam — the two algorithms that are most widely used by practitioners. The discrete update rules of these algorithms are

- **RMSprop:**

\[
\begin{align*}
\mathbf{v}_{t+1} &= \alpha \mathbf{v}_t + (1 - \alpha) (\nabla f(x_t))^2 \\
x_{t+1} &= x_t - \eta \frac{\nabla f(x_t)}{\sqrt{\mathbf{v}_{t+1}} + \epsilon}
\end{align*}
\]

(2)

- **Adam:**

\[
\begin{align*}
\mathbf{v}_{t+1} &= \alpha \mathbf{v}_t + (1 - \alpha) (\nabla f(x_t))^2 \\
\mathbf{m}_{t+1} &= \beta \mathbf{m}_t + (1 - \beta) \nabla f(x_t) \\
x_{t+1} &= x_t - \eta \frac{\mathbf{m}_{t+1}/(1 - \beta^{t+1})}{\sqrt{\mathbf{v}_{t+1}/(1 - \alpha^{t+1})} + \epsilon}
\end{align*}
\]

(3)
In (2) and (3), $\epsilon$ is a small constant used to avoid division by 0. It is usually taken to be $10^{-8}$.

In this paper, we always consider the full batch setting, i.e. $\nabla f(x_t)$ is the gradient instead of some stochastic approximation.

### 2.2 Continuous-time limits

RMSProp and Adam can be studied by considering the limiting ordinary differential equations (ODE) obtained by taking the learning rate $\eta$ to 0. However, different limiting ODEs are obtained when the hyper-parameters are scaled differently. If the hyper-parameters are kept fixed, then as $\eta \to 0$, the memory effect diminishes, because in each discrete iteration we lose the same amount of memory but one iteration occupies shorter and shorter amount of time. In this case, the continuous-time limit for both RMSprop and Adam are the following dynamics

$$
\dot{x} = -\frac{\nabla f(x)}{|\nabla f(x)| + \epsilon}.
$$

Since $\epsilon$ is a small value, this dynamics is close to the continuous-time signGD:

$$
\dot{x} = -\text{sign}(\nabla f(x)).
$$

**Proposition 1.** Assume that $\nabla f$ is bounded and Lipschitz continuous, i.e. there exists constants $M$ and $L$ such that $\|\nabla f(x_1)\| \leq M$ and $\|\nabla f(x_1) - \nabla f(x_2)\| \leq L\|x_1 - x_2\|$ hold for any $x_1$ and $x_2$. Let $\{x^k\}$, $k = 0, 1, 2, \cdots$ be the solution given by algorithm (2) or (3) starting from $x_0$, $m_0$ and $v_0 \geq 0$, with learning rate $\eta$ and some fixed $\alpha, \beta \in (0, 1)$ and $\epsilon > 0$. Let $X^\eta(\cdot)$ be a piece-wise constant function of $t \in [0, \infty)$ that satisfies

$$
X^\eta(t) = x^k, \text{ for } t \in [k\eta, (k+1)\eta).
$$

In addition, let $x(\cdot)$ be the solution of (4) initialized from $x_0$. Then, for any $T > 0$, we have

$$
\lim_{\eta \to 0} \sup_{t \in [0,T]} \|X^\eta(t) - x(t)\| = 0.
$$

The proof of the Proposition is given in the appendix. Figure 2 provides numerical evidences that RMSprop and Adam are close to signGD in a finite time interval when $\eta$ is small while $\alpha$ and $\beta$ are fixed. The closeness between signGD and RMSprop is also shown in Figure 5 for a synthetic objective function.

On the other hand, if we want to keep the strength of the memory effect fixed, we have to let $\alpha$ and $\beta$ go to 1 when $\eta$ tends to 0. Specifically, let $\alpha = 1 - a\eta$ and $\beta = 1 - b\eta$, with $a$ and $b$ being positive constants. Then, it is easy to show that the trajectories of (2) and (3) converge to the following ODEs (7) and (8), respectively.

- **ODE RMSprop:**
  $$
  \dot{v} = a(\nabla f(x)^2 - v) \\
  \dot{x} = -\frac{\nabla f(x)}{\sqrt{v} + \epsilon}
  $$

- **ODE Adam:**

4
Figure 2: Early stage training loss curve of signGD and Adam/RMSprop with different learning rates. The x-axis is the time (learning rate×number of iterations) and y-axis denotes the training loss. For Adam, $\beta = 0.9$ and $\alpha = 0.999$; for RMSprop $\alpha = 0.99$. Learning rate of signGD is $10^{-5}$. Experiments conducted on a fully-connected neural network with three hidden layers, with width 256, 128, and 64, respectively. The training data is taken from 2 classes of CIFAR10 with 1000 data per class.

$$
\dot{v} = a(\nabla f(x)^2 - v) \\
\dot{m} = b(\nabla f(x) - m) \\
\dot{x} = -\frac{(1-e^{-bt})^{-1}m}{\sqrt{(1-e^{-at})^{-1}v + \epsilon}} \\
$$

(8)

The following proposition is a simplification of Theorem 3.2 in [1]. More general results including the stochastic case for Adam are given in [1].

**Proposition 2.** Under the same condition of $f$ in Proposition 4, let $\{x^\eta_k\}, k = 0, 1, 2, \cdots$ be the solution given by algorithm (2) starting from $x_0$ and $v_0 = 0$, with learning rate $\eta$ and $\alpha = 1 - a\eta$ for a fixed constant $a > 0$. Let $X^\eta(\cdot)$ be a piece-wise constant vector function of $t \in [0, \infty)$ that satisfies

$$X^\eta(t) = x^\eta_k, \quad \text{for } t \in [k\eta, (k+1)\eta).$$

In addition, let $x(\cdot)$ be the solution of (7) initialized from $x_0$ and $v_0 \geq 0$. Then, for any $T > 0$, we have

$$\lim_{\eta \to 0} \sup_{t \in [0,T]} \|X^\eta(t) - x(t)\| = 0.$$ 

(9)

Similarly, if $\alpha = 1 - a\eta$ and $\beta = 1 - b\eta$ for some constants $a, b > 0$, then the same convergence statements hold for the solutions of (3) and (8).

In [1] a Lyapunov function is found for the continuous version of Adam in the state space of $z = (x, m, v)$:

$$V(t, z) := f(x) + \frac{1}{2} \|m\|^2_U(t, v)^{-1},$$

(10)

where

$$U(t, v) = b(1-e^{-bt}) \left( \sqrt{\frac{v}{1-e^{-at}}} + \epsilon \right),$$

(11)
and $\|\mathbf{m}\|_u^2$ is defined as $\sum_{i=1}^{d} u_i \mathbf{m}_i^2$.

As can be seen from (7) and (8), the smaller the value of $a$ and $b$, the slower the dynamics of $\mathbf{v}$ (and $\mathbf{m}$), and consequently the slower the whole dynamics. Numerical results in Figure 3 confirm this. However, it is worth mentioning that this difference in convergence speed does not manifest at the very beginning of the training process. To understand this, consider the dynamics of Adam [8]. Assume that at the beginning $\nabla f(\mathbf{x}) = \nabla f(\mathbf{x}_0)$ is unchanged. Further assume that $\mathbf{v}_0 = \mathbf{m}_0 = 0$ and $\epsilon = 0$. Then,

$$\mathbf{v}_t = (1 - e^{-at})\nabla f(\mathbf{x})^2,$$
$$\mathbf{m}_t = (1 - e^{-bt})\nabla f(\mathbf{x}).$$

Hence, we have

$$\dot{\mathbf{x}} = -\frac{(1 - e^{-bt})^{-1}(1 - e^{-bt})\nabla f(\mathbf{x})}{\sqrt{(1 - e^{-at})^{-1}(1 - e^{-at})\nabla f(\mathbf{x})^2}} = -1,$$

which shows that the initial speed of $\mathbf{x}$ does not depend on $a$ and $b$.

Figure 3: How the values of $a$ and $b$ affect the speed of dynamics. **Left:** Adam; **Right:** RMSprop. The learning rate is 0.001 for all the experiments. The model and training data are the same as Figure 2. One can see that at the early stage of the training (after a very short period from initialization), optimizers with larger $a$ and $b$ converge faster.

## 3 RMSprop and signGD: Fast convergence and oscillation

In this section we focus on RMSprop. Figure 4 shows the loss curves and trajectories of RMSprop on a typical multi-layer neural network model. There are three obvious features:

1. **Fast initial convergence:** the loss curve decreases very fast, sometimes even super-linearly, at the early stage of the training.

2. **Small oscillations:** The fast initial convergence is followed by oscillations around the minimum.

3. **Large spikes:** spikes are sudden increase of the value of the loss. They are followed by an oscillating recovery. Different from small oscillations, spikes make the loss much larger and the interval between two spikes is longer.
Figure 4: The loss curves and trajectories of the RMSprop on a neural network model and CIFAR-10 data. Model and data the same as Figure 1. The learning rate is $1e^{-3}$, and $\alpha = 0.99$. 2000 iterations are run. **Left:** The whole training loss curves **Right:** The training loss of the last 500 iterations.

**Fast initial convergence.** As discussed in the last section, when $\eta$ tends to 0 while $\alpha$ stays fixed, RMSprop tends to signGD. So the loss curve of RMSprop and signGD align well during initial phase as shown in Figure 2. Figure 5 shows the loss curves of both signGD and RMSprop on a quadratic objective function. Their behaviors are similar — they both experience fast initial convergence and then the loss stops decreasing. For this reason, we will study the fast initial convergence of RMSprop with the help of signGD. In the (strongly) convex setting, the following proposition shows that continuous-time signGD can reach the global minimum in finite time.

**Proposition 3.** Assume that the objective function satisfies the Polyak-Lojasiewicz (PL) condition: $\|\nabla f(x)\|_2^2 \geq \mu f(x)$ for any $x$. Assume that $x(\cdot)$ is given by the continuous-time signGD dynamics (5), then we have

$$f(x(t)) \leq \left( \sqrt{f(x_0)} - \frac{\sqrt{\mu}}{2} t \right)^2.$$
Proof. We have
\[
\frac{d}{dt} f(x(t)) = -\langle \text{sign}(\nabla f(x(t))), \nabla f(x(t)) \rangle = -\|\nabla f(x(t))\|_1 \\
\leq -\|\nabla f(x(t))\|_2 \leq -\sqrt{\mu f(x(t))}.
\]
Hence, we have
\[
\frac{d}{dt} \sqrt{f(x(t))} \leq -\frac{\sqrt{\mu}}{2},
\]
which implies
\[
f(x(t)) \leq \left( \sqrt{f(x_0)} - \frac{\sqrt{\mu}}{2} t \right)^2.
\]
\[
\square
\]

Small oscillations For standard dynamical systems, small oscillations can be analyzed via linearization. Oscillations occur when the Jacobian has purely imaginary eigenvalues. However, in our case standard linearization cannot be applied easily. If we set \( \epsilon = 0 \), then the dynamics is singular at the stationary point where \( \mathbf{v} = 0 \). If \( \epsilon \) is positive, all the eigenvalues of the Jacobian are negative real numbers, which means the linearized dynamics is strictly attractive and no oscillation will happen. However, since \( \epsilon \) is usually small, the linearization approximates the original dynamics well only in a very small neighborhood of the stationary point, smaller than the range of the oscillations, hence cannot explain the oscillation.

For low dimensional strongly convex objective functions, RMSprop can converge to a 2-periodic solution instead of the global minimum. For example, if the objective function is \( f(x) = \frac{1}{2}x^2 \), then the 2-periodic solution is an oscillation between \( \eta \) and \( -\eta \), where \( \eta \) is the learning rate. Figure 6 shows the convergence to this 2-periodic solution.

Figure 6: The trajectory of RMSprop for the 1-dimensional quadratic function \( f(x) = \frac{x^2}{2} \) for different values of \( \alpha \). \( \eta = 0.01 \). One sees that all the trajectories eventually converge to the 2-periodic solution at \( \frac{\eta}{2} \) and \( -\frac{\eta}{2} \).

For more complicated objective functions, such as high-dimensional quadratic function, or the loss function of neural network models, the RMSprop trajectories show more complicated
oscillations patterns, such as the spikes. As we will see in the next section, Adam is more vulnerable to large spikes. We will take a closer look of the large spikes in the next section.

4 Adam: performances for different values of $a$ and $b$

The dynamic behavior of Adam is more complicated than RMSprop since it is influenced by 2 hyper-parameters. Different combinations of $\alpha$ and $\beta$ (or $a$ and $b$) can lead to different dynamic patterns. To rule out the influence of the learning rate, we will consider $a$ and $b$ instead of $\alpha$ and $\beta$. As is mentioned before, $\alpha$ and $\beta$ are given by $a$ and $b$ through $\alpha = 1 - a\eta$ and $\beta = 1 - b\eta$. As we have seen in Proposition 1, when $a$ and $b$ are sufficiently large compared to $\eta$, Adam behaves like signGD. For relatively small $a$ and $b$, through extensive numerical experiments, we have found that there are roughly three different regimes of qualitative patterns in the parameter space (see Figure 7):

1. **The spike regime** happens when $b$ is sufficiently larger than $a$. In this regime large spikes appear in the loss curve, which makes the optimization process unstable.

2. **The oscillation regime** happens when $a$ and $b$ have similar magnitude (or in the same order). In this regime the loss curve exhibits fast and small oscillations. Small loss and stable loss curve can be achieved.

3. **The divergence regime** happens when $a$ is sufficiently larger than $b$. In this regime the loss curve is unstable and usually diverges after a period of training. This regime should be avoided in practice since the training loss stays large.

In Figure 7 we show one typical loss curve for each regime for a typical neural network model. We also show typical trajectories in the state space of $(\|x\|, \|m\|, \|\sqrt{v}\|)$ for the three regimes. These trajectories are also qualitatively different for different regimes.

Next we study the transition between the different regimes and the training loss behavior in different regimes. To this end, we carried out experiments for a multi-layer neural network model on the Fashion-MNIST dataset, with different values of $a$ and $b$ until the behavior of the training loss curve stabilizes. The left panel of Figure 8 shows the heatmap of the average loss value of the last 1000 iterations. The right panel of Figure 8 shows the classification of the behavior of the training curve into three different categories (oscillations, spikes and divergence).

From these figures we see that in the divergence regime the training loss does not perform well (actually in some cases it may even blow up). Hence this regime should be avoided in practice. In the oscillation regime the loss values are small and quite robust with respect to the change of hyper-parameters. Therefore this is the regime that should be preferred in practice. This is the regime when $a \approx b$.

**Training ResNets on CIFAR10** The above investigation suggests that Adam performs better when $\alpha \approx \beta$. Here we provide further support by considering a more realistic problem: training a ResNet18 [5] on CIFAR10 using stochastic Adam with large batch size. The results are shown in Figure 9. We see that with the default parameters ($\beta = 0.9, \alpha = 0.999$), there are large spikes during the late phase of training. In contrast, when $a \approx b$, Adam converges very smoothly and is also faster than using the default parameters.
Figure 7: The three typical behavior patterns for Adam and the trajectories in the state space of ($\|x\|$, $\|m\|$, $\|\sqrt{v}\|$). $\eta = 0.001$. The model and the training data are the same as Figure 2. The first row shows the loss curve of totally 1000 iterations, the second row shows part of the loss curve (the last 200 iterations for oscillation and divergence regimes, and 400 – 800 iterations for the spike regime), the bottom row shows the state space trajectory in the same period shown in the second row. **Left:** $a = 1$, $b = 100$, large spikes appear in the loss curve; **Middle:** $a = 10$, $b = 10$, the loss is small and oscillates very fast, and the amplitude of the oscillation is also small; **Right:** $a = 100$, $b = 1$, the loss is large and blows up.
Figure 8: **Left:** Heatmap of average training loss of Adam on a multi-layer neural network model. The loss is averaged over the last 1000 iterations and is shown in logarithmic scale. \( a \) and \( b \) range from 0.1 to 100 and are also shown in logarithmic scale. **Right:** The classification of the different training behavior. 500 data samples are taken from each class of Fashion-MNIST. The neural network model is fully connected with 6 hidden layers, with 500 neurons per layer. The learning rate of is 1e-3.

Figure 9: Loss curves of stochastic Adam on a ResNet18 model and CIFAR-10 dataset. The learning rate is 1e-3. The red line shows the results of using the default hyper-parameters setting \((\beta = 0.9, \alpha = 0.999)\). 1000 images are taken from each class to form the training dataset. The network is a ResNet18. The number of channels is half of the ones from the typical setting in [5]. The batch size is 1000.

5 Discussion

In this paper we reported the results of some systematical investigation on the dynamic behavior of adaptive gradient algorithms, particularly RMSprop and Adam. Three typical phenomena—fast initial convergence, small oscillation and spikes—are observed and analyzed. The influence of the choice of the hyper-parameters on the dominant training behavior is studied.
It is worth noting that the investigation in this paper focuses on the full-batch setting. However, the result in Figure[9] provides some evidence to show that the phenomena revealed here should also be of relevance for the stochastic setting when the batch size is large enough. The systematical study of the influence of batch size, especially in the small-batch regime, is left to future work.

There are still many other important open questions. For example, learning rate decay is a common practice used in training large neural networks. When performing learning rate decay, usually one does not change the values of $\alpha$ and $\beta$. This makes the effective $a$ and $b$ larger, pushing the optimizer to the signGD-like regime. Another choice is to adaptively tune $\alpha, \beta$ such that $a$ and $b$ are kept fixed. It is interesting to see the comparison of the two strategies.

This paper focuses on optimization. For machine learning problems, another important consideration when implementing optimization algorithms is the generalization performance. It has been reported that the solutions found by adaptive gradient algorithms usually perform a bit worse than those found by SGD in terms of generalization (see [12]). The study of generalization performance of adaptive gradient algorithms is left for future work.

References

[1] Anas Barakat and Pascal Bianchi. Convergence and dynamical behavior of the ADAM algorithm for non convex stochastic optimization. arXiv preprint arXiv:1810.02263, 2018.

[2] 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, 2018.

[3] André Belotto da Silva and Maxime Gazeau. A general system of differential equations to model first order adaptive algorithms. arXiv preprint arXiv:1810.13108, 2018.

[4] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. Journal of machine learning research, 12(7), 2011.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[6] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

[7] Xiaoou Li and Francesco Orabona. On the convergence of stochastic gradient descent with adaptive stepsizes. In The 22nd International Conference on Artificial Intelligence and Statistics, pages 983–992, 2019.

[8] Boris Teodorovich Polyak. Gradient methods for minimizing functionals. Zhurnal Vychislitel’noi Matematiki i Matematicheskoi Fiziki, 3(4):643–653, 1963.

[9] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of Adam and beyond. In International Conference on Learning Representations, 2018.

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

[11] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural networks for machine learning, 4(2):26–31, 2012.
A Proof of Proposition 1

Here we prove Proposition 1. For that purpose, we show that for any $T > 0$ and $\tau > 0$, there exists an $\eta_{T, \tau}$, such that as long as $\eta < \eta_{T, \tau}$ we have

$$\sup_{t \in [0, T]} \|X^n(t) - x(t)\| < \tau. \tag{12}$$

In the following we focus on RMSprop. The proof for Adam is similar.

First, let $K$ be a positive integer whose value will be specified later, and let $\tilde{x}^n_k = x(k\eta)$.

Then, for $x^n_K$ and $\tilde{x}^n_K$ we have

$$\|x^n_K - x_0\| \leq \eta \sum_{i=0}^{K-1} \left\| \frac{\nabla f(x_i)}{\sqrt{v_{i+1} + \epsilon}} \right\| \leq \eta \sum_{i=0}^{K-1} \frac{M}{\epsilon} = \frac{\eta KM}{\epsilon},$$

and

$$\|\tilde{x}^n_K - x_0\| \leq \int_{0}^{K\eta} \left\| \frac{\nabla f(x(t))}{|\nabla f(x(t))| + \epsilon} \right\| dt \leq \frac{\eta KM}{\epsilon}.$$ 

Therefore,

$$\|x^n_K - \tilde{x}^n_K\| \leq \frac{2\eta KM}{\epsilon}. \tag{13}$$

Next, for $k \geq K$, we have

$$x^n_{k+1} - \tilde{x}^n_{k+1} = (x^n_k - \tilde{x}^n_k) + \left( \int_{k\eta}^{(k+1)\eta} \frac{\nabla f(x(t))}{|\nabla f(x(t))| + \epsilon} dt - \eta \frac{\nabla f(x^n_k)}{\sqrt{v_{k+1} + \epsilon}} \right).$$

Let

$$\Delta = \int_{k\eta}^{(k+1)\eta} \frac{\nabla f(x(t))}{|\nabla f(x(t))| + \epsilon} dt - \eta \frac{\nabla f(x^n_k)}{\sqrt{v_{k+1} + \epsilon}},$$

then

$$\|x^n_{k+1} - \tilde{x}^n_{k+1}\| \leq ||x^n_k - \tilde{x}^n_k|| + ||\Delta||. \tag{14}$$

Next we estimate $||\Delta||$. First we have

$$\Delta = \int_{k\eta}^{(k+1)\eta} \frac{\nabla f(x(t))}{|\nabla f(x(t))| + \epsilon} dt - \eta \frac{\nabla f(x^n_k)}{\sqrt{v_{k+1} + \epsilon}} \tag{15}$$

$$= \int_{k\eta}^{(k+1)\eta} \frac{\nabla f(x(t))}{|\nabla f(x(t))| + \epsilon} dt + \int_{k\eta}^{(k+1)\eta} \frac{\nabla f(x(t)) - \nabla f(x^n_k)}{\sqrt{v_{k+1} + \epsilon}} dt$$

$$:= I + J.$$
For $J$, we have

$$
\|J\| \leq \frac{1}{\epsilon} \int_{k\eta}^{(k+1)\eta} \|\nabla f(x(t)) - \nabla f(x_k^{\eta})\| dt
\leq \frac{L}{\epsilon} \int_{k\eta}^{(k+1)\eta} \|x(t) - x_k^{\eta}\| dt
\leq \frac{L}{\epsilon} \int_{k\eta}^{(k+1)\eta} (\|x(t) - \tilde{x}_k^{\eta}\| + \|\tilde{x}_k^{\eta} - x_k^{\eta}\|) dt
\leq \frac{L}{\epsilon} \left( \eta^2 M + \eta \|\tilde{x}_k^{\eta} - x_k^{\eta}\| \right)
= \eta L \|\tilde{x}_k^{\eta} - x_k^{\eta}\| + \frac{\eta^2 LM}{\epsilon^2}.
$$

(15)

For $I$, we have

$$
\|I\| \leq \int_{k\eta}^{(k+1)\eta} \left\| \frac{|\nabla f(x(t))|}{|\nabla f(x(t))| + \epsilon} \frac{\sqrt{\sqrt{k+1} - |\nabla f(x(t))|}}{\sqrt{\sqrt{k+1} + \epsilon}} \right\| dt
\leq \frac{1}{\epsilon} \int_{k\eta}^{(k+1)\eta} \|\sqrt{\sqrt{k+1} - |\nabla f(x_k^{\eta})|}\| dt + \frac{1}{\epsilon} \int_{k\eta}^{(k+1)\eta} \|\|\nabla f(x_k^{\eta})\| - |\nabla f(x(t))|\| dt
$$

(16)

The second term in (16) can be estimated in a similar way as $\|J\|$, and it can be bounded by

$$
\frac{\eta L}{\epsilon} \|\tilde{x}_k^{\eta} - x_k^{\eta}\| + \frac{\eta^2 LM}{\epsilon^2}.
$$

For the first term of (16), use the fact that $(a - b)^2 \leq a^2 - b^2$ for any $a \geq b \geq 0$, we have

$$
\frac{1}{\epsilon} \int_{k\eta}^{(k+1)\eta} \|\sqrt{\sqrt{k+1} - |\nabla f(x_k^{\eta})|}\| dt
\leq \frac{\eta}{\epsilon} \|\sqrt{k+1} - |\nabla f(x_k^{\eta})|\|^2
= \frac{\eta}{\epsilon} \left( (1 - \alpha) \nabla f^2(x_k^{\eta}) + \alpha (1 - \alpha) \nabla f^2(x_{k-1}^{\eta}) + \cdots + \alpha^k (1 - \alpha) \nabla f^2(x_0^{\eta}) - \nabla f^2(x_k^{\eta}) \right)^2
\leq \frac{\eta}{\epsilon} \left( (1 - \alpha) \sum_{i=0}^{K-1} \alpha^i (\nabla f^2(x_k^{\eta} - x_i^{\eta}) - \nabla f^2(x_k^{\eta})) \right)^2
\leq \frac{2\eta^3/2 MK^{1/2}}{\epsilon^{3/2}} + \frac{2\eta M \alpha K/2}{\epsilon}.
$$

(17)

Hence we have

$$
\|I\| \leq \frac{\eta L}{\epsilon} \|\tilde{x}_k^{\eta} - x_k^{\eta}\| + \frac{\eta^2 LM}{\epsilon^2} + \frac{2\eta^3/2 MK^{1/2}}{\epsilon^{3/2}} + \frac{2\eta M \alpha K/2}{\epsilon}.
$$

(18)
Combining (18) with (15) we get the estimate of $\Delta$:

$$
\|\Delta\| \leq \frac{2\eta L}{\epsilon} \|\tilde{x}_k - x_k\| + \frac{2\eta^2 LM}{\epsilon^2} + \frac{2\eta^3/2 MK^{1/2}}{\epsilon^{3/2}} + \frac{2\eta M \alpha^{K/2}}{\epsilon}.
$$

(19)

Hence

$$
\|x_{k+1} - \tilde{x}_{k+1}\| \leq (1 + \frac{2\eta L}{\epsilon}) \|x_k - \tilde{x}_k\| + \frac{2\eta^2 LM}{\epsilon^2} + \frac{2\eta^3/2 MK^{1/2}}{\epsilon^{3/2}} + \frac{2\eta M \alpha^{K/2}}{\epsilon}.
$$

(20)

Finally, by Gronwall’s inequality, we have

$$
\|x_k - \tilde{x}_k\| \leq \left(1 + \frac{2\eta L}{\epsilon}\right)^k \|x_K - \tilde{x}_K\| + \left(1 + \frac{2\eta L}{\epsilon}\right)^{k-K} \left(\frac{\eta M}{\epsilon} + \frac{\eta^{1/2} MK^{1/2}}{\epsilon^{1/2} L} + \frac{M \alpha^{K/2}}{L}\right)
$$

\leq \left(1 + \frac{2\eta L}{\epsilon}\right)^k \left(\frac{2\eta K M}{\epsilon} + \frac{\eta^{1/2} MK^{1/2}}{\epsilon^{1/2} L} + \frac{M \alpha^{K/2}}{L}\right).

(21)

We want (21) to hold for $t \leq T$, which means for all $k \leq \frac{T}{\eta}$. For these values of $k$, we have

$$
\|x_k - \tilde{x}_k\| \leq e^{\frac{2\eta L}{\epsilon}} \left(\frac{2\eta K M}{\epsilon} + \frac{\eta^{1/2} MK^{1/2}}{\epsilon^{1/2} L} + \frac{M \alpha^{K/2}}{L}\right).
$$

(22)

Therefore, for any fixed small value $\tau > 0$, by taking sufficiently large $K$ and sufficiently small $\eta$, we can achieve

$$
\|x_k - \tilde{x}_k\| \leq \frac{\tau}{2},
$$

(23)

for any $0 \leq k \leq \frac{T}{\eta} + 1$. Then, if we further let

$$
\eta < \frac{\tau \epsilon}{4M}
$$

for any $t \in [0, T]$, let $k$ satisfy $t \in [k\eta, (k + 1)\eta)$, we have

$$
\|X^\eta(t) - x(t)\| \leq \|x_k - \tilde{x}_k\| + \|X^\eta(t) - \tilde{x}_k\| + \|x(t) - x_k\|
$$

\leq \frac{\tau}{2} + \frac{2\eta M}{\epsilon}
$$

\leq \tau.

(24)

This completes the proof.