Convergence Analysis of a Momentum Algorithm with Adaptive Step Size for Non Convex Optimization

Anas Barakat, Pascal Bianchi

Proceedings of Machine Learning Research 129:225–240, 2020 ACML 2020

Convergence Rates of a Momentum Algorithm with Bounded Adaptive Step Size for Nonconvex Optimization

Anas Barakat ANAS.BARAKAT@TELECOM-PARIS.FR
Pascal Bianchi PASCAL.BIANCHI@TELECOM-PARIS.FR
LTCI, Télécom Paris, Institut Polytechnique de Paris, France

Editors: Sinno Jialin Pan and Masashi Sugiyama

Abstract

Although ADAM is a very popular algorithm for optimizing the weights of neural networks, it has been recently shown that it can diverge even in simple convex optimization examples. Several variants of ADAM have been proposed to circumvent this convergence issue. In this work, we study the ADAM algorithm for smooth nonconvex optimization under a boundedness assumption on the adaptive learning rate. The bound on the adaptive step size depends on the Lipschitz constant of the gradient of the objective function and provides safe theoretical adaptive step sizes. Under this boundedness assumption, we show a novel first order convergence rate result in both deterministic and stochastic contexts. Furthermore, we establish convergence rates of the function value sequence using the Kurdyka-Łojasiewicz property.

Keywords: Nonconvex optimization, Adaptive gradient methods, Kurdyka-Łojasiewicz inequality.

1. Introduction

Consider the unconstrained optimization problem $\min_{x\in\mathbb{R}^d} f(x)$, where $f : \mathbb{R}^d \to \mathbb{R}$ is a differentiable map and d is an integer. Gradient descent is one of the most classical algorithms to solve this problem. Since the seminal work Robbins and Monro (1951), its stochastic counterpart became one of the most popular algorithms to solve machine learning problems (see Bottou et al. (2018) for a recent survey). Recently, a class of algorithms called adaptive algorithms which are variants of stochastic gradient descent became very popular in machine learning applications (Duchi et al., 2011). Using a coordinate-wise step size computed using past gradient information, the step size is adapted to the function to optimize and does not follow a predetermined step size schedule. Among these adaptive algorithms, ADAM (Kingma and Ba, 2015) is very popular for optimizing the weights of neural networks. However, recently, Reddi et al. (2018) exhibited a simple convex stochastic optimization problem over a compact set where ADAM fails to converge because of its short-term gradient memory. Moreover, they proposed an algorithm called AMSGRAD to fix the convergence issue of ADAM. This work opened the way to the emergence of other variants of ADAM to overcome its convergence issues (see Section 3 for a detailed review). In this work, under a bounded step size assumption, we propose a theoretical analysis of ADAM for nonconvex

optimization.

Contributions.

- We establish a convergence rate for Adam in the deterministic case for nonconvex optimization under a bounded step size. This algorithm can be seen as a deterministic clipped version of Adam which guarantees safe theoretical step sizes. More precisely, if n is the number of iterations of the algorithm, we show a $O(1/n)$ convergence rate of the minimum of the squared gradients norms by introducing a suitable Lyapunov function.

- We show a similar convergence result for nonconvex stochastic optimization up to the limit of the variance of stochastic gradients under an almost surely bounded step size. In comparison to the literature, the hypothesis of the boundedness of the gradients is relaxed and the convergence result is independent of the dimension d of the parameters.

- We propose a convergence rate analysis of the objective function of the algorithm using the Kurdyka-Łojasiewicz (KŁ) property. To the best of our knowledge, this is the first time such a result is established for an adaptive optimization algorithm.

The rest of the paper is organized as follows. Section 2 introduces the algorithm we analyze. Section 3 considers some related works. Section 4 establishes first order convergence rates in terms of the minimum of the gradients norms in both deterministic and stochastic settings. Finally, Section 5 derives function value convergence rates under the KŁ property. All the proofs are deferred to the Appendix in the supplementary material.

2. A Momentum Algorithm with Adaptive Step Size

Notations. All operations between vectors of $\mathbb{R}^d$ are to read coordinatewise. In particular, for two vectors x, y in $\mathbb{R}^d$ and $\alpha \in \mathbb{Z}$, we denote by xy, x/y, x^α the vectors on $\mathbb{R}^d$ whose k-th coordinates are respectively given by $x_k y_k$, x_k/y_k, x_k^α. The vector of ones of $\mathbb{R}^d$ is denoted by $\mathbf{1}$. When a scalar is added to a vector, it is added to each one of its coordinates. Inequalities are also to be read coordinatewise. If $x \in \mathbb{R}^d, x \leq \lambda \in \mathbb{R}$ means that each coordinate of x is smaller than λ.

We investigate the following algorithm defined by two sequences (x_n) and (p_n) in $\mathbb{R}^d$:

$$\begin{cases} x_{n+1} = x_n - a_{n+1} p_{n+1} \\ p_{n+1} = p_n + b\left(\nabla f(x_n) - p_n\right) \end{cases} \tag{1}$$

where $\nabla f(x)$ is the gradient of f at point x, (a_n) is a sequence of vectors in $\mathbb{R}^d$ with positive coordinates, b is a positive real constant and $x_0, p_0 \in \mathbb{R}^d$.

Algorithm (1) includes the classical Heavy-ball method as a special case, but is much more general. Indeed, we allow the sequence of step sizes (a_n) to be adaptive : $a_n \in \mathbb{R}^d$ may depend on the past gradients $g_k := \nabla f(x_k)$ and the iterates x_k for $k \leq n$. We stress that the step size a_n is a vector of $\mathbb{R}^d$ and that the product $a_{n+1} p_{n+1}$ in (1) is read componentwise (this is equivalent to the formulation with a diagonal matrix preconditioner applied to the

Table 1: Some famous algorithms.

Algorithm	Effective step size a_{n+1}	Momentum
SGD (Robbins and Monro, 1951)	$a_{n+1} \equiv a$	$b = 1$ (no momentum)
Adagrad (Duchi et al., 2011)	$a_{n+1} = a \left(\sum_{i=0}^{n} g_i^2\right)^{-1/2}$	$b = 1$
Rmsprop (Tieleman and Hinton, 2012)	$a_{n+1} = a \left[\epsilon + \left(c \sum_{i=0}^{n} (1-c)^{n-i} g_i^2\right)^{1/2}\right]^{-1}$	$b = 1$
Adam (Kingma and Ba, 2015)	$a_{n+1} = a \left[\epsilon + \left(c \sum_{i=0}^{n} (1-c)^{n-i} g_i^2\right)^{1/2}\right]^{-1}$	$0 \leq b \leq 1$ (close to 0)

gradient (McMahan and Streeter, 2010; Gupta et al., 2017; Agarwal et al., 2019; Staib et al., 2019)).

We present in Table 1 how to recover some of the famous algorithms with a vector step size formulation. In particular, adam (Kingma and Ba, 2015) defined by the iterates :

$$\begin{cases} x_{n+1} = x_n - \frac{a}{\epsilon + \sqrt{v_{n+1}}} p_{n+1} \\ p_{n+1} = p_n + b\left(\nabla f(x_n) - p_n\right) \\ v_{n+1} = v_n + c\left(\nabla f(x_n)^2 - v_n\right) \end{cases} \tag{2}$$

for constants $a \in \mathbb{R}_+$, $b, c \in [0,1]$, can be seen as an instance of this algorithm by setting $a_n = \frac{a}{\epsilon + \sqrt{v_n}}$ where the vector v_n, as defined above, is an exponential moving average of the gradient squared. For simplification, we omit bias correction steps for p_{n+1} and v_{n+1}. Their effect vanishes quickly along the iterations.

We introduce the main assumption on the objective function which is standard in gradient-based algorithms analysis.

Assumption 1 *The mapping $f : \mathbb{R}^d \to \mathbb{R}$ is:*
(i) continuously differentiable and its gradient ∇f is $L-$Lipschitz continuous,
(ii) bounded from below, i.e., $\inf_{x \in \mathbb{R}^d} f(x) > -\infty$.

3. Related Works

3.1. The Heavy-Ball Algorithm.

Adaptive algorithms as Heavy Ball. Thanks to its small per-iteration cost and its acceleration properties (at least in the strongly convex case), the Heavy-ball method, also called gradient descent with momentum, recently regained popularity in large-scale optimization (Sutskever et al., 2013). This speeding up idea dates back to the sixties with the seminal work of Polyak (1964). In order to tackle nonconvex optimization problems, Ochs et al. (2014) proposed iPiano, a generalization of the well known heavy-ball in the form of a forward-backward splitting algorithm with an inertial force for the sum of a smooth possibly nonconvex and a convex function. In the particular case of the Heavy-ball method, this algorithm writes for two sequences of reals (α_n) and (β_n):

$$x_{n+1} = x_n - \alpha_n \nabla f(x_n) + \beta_n (x_n - x_{n-1}) \,. \tag{3}$$

We remark that Algorithm (1) can be written in a similar fashion by choosing step sizes $\alpha_n = b a_{n+1}$ and inertial parameters $\beta_n = (1-b) a_{n+1}/a_n$. Ochs et al. (2014) only

consider the case where α_n and β_n are real-valued. Moreover, the latter does not consider adaptive step sizes, i.e step sizes depending on past gradient information. We can show some improvement with respect to Ochs et al. (2014) with weaker convergence conditions in terms of the step size of the algorithm (see Appendix A.6) while allowing adaptive vector-valued step sizes a_n (see Proposition 14).

It is shown in Ochs et al. (2014) that the sequence of function values converges and that every limit point is a critical point of the objective function. Moreover, supposing that the Lyapunov function has the KŁ property at a cluster point, they show the finite length of the sequence of iterates and its global convergence to a critical point of the objective function. Similar results are shown in Wu and Li (2019) for a more general version than iPiano (Ochs et al., 2014) computing gradients at an extrapolated iterate like in Nesterov's acceleration.

Convergence rate. Ochs et al. (2014) determines a $O(1/n)$ convergence rate (where n is the number of iterations of the algorithm) with respect to the proximal residual which boils down to the gradient for noncomposite optimization. Furthermore, a recent work introduces a generalization of the Heavy-ball method (and Nesterov's acceleration) to constrained convex optimization in Banach spaces and provides a non-asymptotic hamiltonian based analysis with $O(1/n)$ convergence rate (Diakonikolas and Jordan, 2019). In the same vein, in Section 4, we establish a similar convergence result for an adaptive step size instead of a fixed predetermined step size policy like in the Heavy-ball algorithm (see Theorem 2).

Convergence rates under the KŁ property. The KŁ property is a powerful tool to analyze gradient-like methods. We elaborate on this property in Section 5. Assuming that the objective function satisfies this geometric property, it is possible to derive convergence rates. Indeed, some recent progress has been made to study convergence rates of the Heavy-ball algorithm in the nonconvex setting. Ochs (2018) establishes local convergence rates for the iterates and the function values sequences under the KŁ property. The convergence proof follows a general method that is often used in non-convex optimization convergence theory. This framework was used for gradient descent (Absil et al., 2005), for proximal gradient descent (see Attouch and Bolte (2009) for an analysis with the Łojasiewicz inequality) and further generalized to a class of descent methods called *gradient-like descent* algorithms.

KŁ-based asymptotic convergence rates were established for constant Heavy-ball parameters (Ochs, 2018). Asymptotic convergence rates based on the KŁ property were also shown (Johnstone and Moulin, 2017) for a general algorithm solving nonconvex nonsmooth optimization problems called Multi-step Inertial Forward-Backward splitting (Liang et al., 2016) which has iPiano and Heavy-ball methods as special cases. In this work, step sizes and momentum parameter vary along the algorithm run and are not supposed constant. However, specific values are chosen and consequently, their analysis does not encompass adaptive step sizes i.e. stepsizes that can possibly depend on past gradient information. In the present work, we establish similar convergence rates for methods such as ADAM under a bounded step size assumption (see Theorem 10). We also mention Li et al. (2017) which analyzes the accelerated proximal gradient method for nonconvex programming (APGnc) and establishes convergence rates of the function value sequence by exploiting the KŁ property. This algorithm is a descent method i.e. the function value sequence is shown to decrease over time. In the present work, we analyze adaptive algorithms which are not descent methods.

Note that even Heavy-ball is not a descent method. Hence, our analysis requires additional treatments to exploit the KŁ property : we introduce a suitable Lyapunov function which is not the objective function. We also point out the recent work Xie et al. (2019) which analyzes the AdaGrad-Norm algorithm under the global Polyak-Łojasiewicz condition. This condition is a particular case of the KŁ property (see Section 5).

Theoretical guarantees for Adam-like algorithms. The recent literature on adaptive optimization algorithms is vast. For instance, for AdaGrad-like algorithms, several works cover the nonconvex setting (Wu et al., 2018; Ward et al., 2019; Xie et al., 2019; Li and Orabona, 2019). In the following, we almost exclusively focus on Adam-like algorithms which are different because of the momentum. The first type of convergence results uses the online optimization framework which controls the convergence rate of the average regret. This framework was adopted for AmsGrad, AdamNC (Reddi et al., 2018), AdaBound and AmsBound (Luo et al., 2019). In this setting, it is assumed that the feasible set containing the iterates is bounded by adding a projection step to the algorithm if needed. We do not make such an assumption in our analysis. (Reddi et al., 2018) establishes a regret bound in the convex setting.

The second type of theoretical results is based on the control of the norm of the (stochastic) gradients. We remark that some of these results depend on the dimension of the parameters. Zhou et al. (2018) improves this dependency in comparison to Chen et al. (2019). The convergence result in De et al. (2018) is established under quite specific values of a_{n+1}, b_n and ϵ. Zaheer et al. (2018) show a $O(1/n)$ convergence rate for an increasing mini-batch size. However, the proof is provided for RMSprop and seems difficult to adapt to Adam which involves a momentum term. Indeed, unlike RMSProp, Adam does not admit the objective function as a Lyapunov function.

We also remark that all the available theoretical results assume boundedness of the (stochastic) gradients. We do not make such an assumption. Furthermore, we do not add any decreasing $1/\sqrt{n}$ factor in front of the adaptive step size as it is considered in Reddi et al. (2018); Luo et al. (2019) and Chen et al. (2019). Although constant hyperparameters b and c are used in practice, theoretical results are often established for non constant b_n and c_n (Reddi et al., 2018; Luo et al., 2019). We also mention that most of the theoretical bounds depend on the dimension of the parameter (Reddi et al., 2018; Zhou et al., 2018; Chen et al., 2018; Zou et al., 2019; Chen et al., 2019; Luo et al., 2019).

Other variants of Adam . Recently, several other algorithms were proposed in the literature to enhance Adam . Although these algorithms lack theoretical guarantees, they present interesting ideas and show good practical performance. For instance, AdaShift (Zhou et al., 2019) argues that the convergence issue of Adam is due to its unbalanced step sizes. To solve this issue, they propose to use temporally shifted gradients to compute the second moment estimate in order to decorrelate it from the first moment estimate. Nadam (Dozat, 2016) incorporates Nesterov's acceleration into Adam in order to improve its speed of convergence. Moreover, originally motivated by variance reduction, QHAdam (Ma and Yarats, 2019) replaces both Adam's moment estimates by quasi-hyperbolic terms and recovers Adam , Rmsprop and Nadam as particular cases (modulo the bias correction). Guided by the same variance reduction principle, Radam (Liu et al., 2019) estimates the variance of the effective

step size of the algorithm and proposes a multiplicative variance correction to the update rule.

Step size bound. Perhaps, the closest idea to our algorithm is the recent AdaBound (Luo et al., 2019) which considers a dynamic learning rate bound. Luo et al. (2019) show that extremely small and large learning rates can cause convergence issues to ADAM and exhibit empirical situations where such an issue shows up. Inspired by the gradient clipping strategy proposed in Pascanu et al. (2013) to tackle the problem of vanishing and exploding gradients in training recurrent neural networks (see Zhang et al. (2019) for recent progress), Luo et al. (2019) apply clipping to the effective step size of the algorithm in order to circumvent step size instability. More precisely, authors propose dynamic bounds on the learning rate of adaptive methods such as Adam or AmsGrad to solve the problem of extreme learning rates which can lead to poor performance. Initialized respectively at 0 and ∞, lower and upper bounds both converge smoothly to a constant final step size following a predetermined formula defined by the user. Consequently, the algorithm resembles an adaptive algorithm in the first iterations and becomes progressively similar to a standard SGD algorithm. Our approach is different : we propose a static bound on the adaptive learning rate which depends on the Lipschitz constant of the objective function. This bound stems naturally from our theoretical derivations.

4. First Order Convergence Rate

4.1. Deterministic setting

Let $(H_n)_{n\geq 0}$ be a sequence defined for all $n \in \mathbb{N}$ by $H_n := f(x_n) + \frac{1}{2b}\langle a_n, p_n^2\rangle$.

We further assume the following step size growth condition.

Assumption 2 *There exists $\alpha > 0$ s.t. $a_{n+1} \leq \frac{a_n}{\alpha}$.*

Note that this assumption is satisfied for ADAM with $\alpha = \sqrt{1-c}$ where c is the parameter in (2). Unlike in AmsGrad (Reddi et al., 2018), the step size a_n is not necessarily nonincreasing. Indeed, α can be strictly smaller than 1 in Assumption 2 as it is the case for Adam .

We provide a proof of the following key lemma in Appendix A.2.

Lemma 1 *Let Assumptions 1 and 2 hold true. Then, for all $n \in \mathbb{N}$, for all $u \in \mathbb{R}_+$,*

$$H_{n+1} \leq H_n - \langle a_{n+1}p_{n+1}^2, A_{n+1}\rangle - \frac{b}{2}\langle a_{n+1}(\nabla f(x_n) - p_n)^2, B\mathbf{1}\rangle \,, \tag{4}$$

where $A_{n+1} := 1 - \dfrac{a_{n+1}L}{2} - \dfrac{|b-(1-\alpha)|}{2u} - \dfrac{1-\alpha}{2b}$, *and* $B := 1 - \dfrac{|b-(1-\alpha)|u}{b} - (1-\alpha)$.

We now state one of the principal convergence results about Algorithm 1. In particular, we establish a sublinear convergence rate for the minimum of the gradients norms until time n.

Theorem 2 *Let Assumptions 1 and 2 hold true. Suppose that $1-\alpha < b \leq 1$. Let $\varepsilon > 0$ s.t. $a_{\sup} := \frac{2}{L}\left(1 - \frac{(b-(1-\alpha))^2}{2b\alpha} - \frac{1-\alpha}{2b} - \varepsilon\right)$ is nonnegative. Let $\delta > 0$ s.t. for all $n \in \mathbb{N}$,*

$$\delta \leq a_{n+1} \leq \min\left(a_{\sup}, \frac{a_n}{\alpha}\right) . \tag{5}$$

Then, the sequence (H_n) *is nonincreasing and* $\sum_n \|p_n\|^2 < \infty$. *In particular,* $\lim x_{n+1} - x_n \to 0$ *and* $\lim \nabla f(x_n) \to 0$ *as* $n \to +\infty$. *Moreover, for all* $n \geq 1$,

$$\min_{0 \leq k \leq n-1} \|\nabla f(x_k)\|^2 \leq \frac{4}{nb^2} \left(\frac{H_0 - \inf f}{\delta \varepsilon} + \|p_0\|^2 \right).$$

Sketch of the proof. The key element of the proof is Lemma 1 which is a descent lemma on the function H. Indeed, the assumptions of the theorem guarantee that $A_{n+1} \geq \varepsilon$ and $B \geq 0$. Then, the result stems from summing the inequalities of Lemma 1. The proof can be found in Appendix A.4.

We provide some comments on this result.

Dimension dependence. Unlike most of the theoretical results for variants of Adam as gathered in Appendix A.1, we remark that the bound does not depend on the dimension d of the parameter x_k.

Comparison to gradient descent. A similar result holds for deterministic gradient descent (see Nesterov (2004, p.28)). If γ is a fix step size for gradient descent and there exist $\delta > 0, \varepsilon > 0$ s.t. $\gamma > \delta$ and $1 - \frac{\gamma L}{2} > \varepsilon$, then (see Appendix A.7) for all $n \geq 1$:

$$\min_{0 \leq k \leq n-1} \|\nabla f(x_k)\|^2 \leq \frac{f(x_0) - \inf f}{n\gamma(1 - \frac{\gamma L}{2})} \leq \frac{f(x_0) - \inf f}{n\delta\varepsilon}.$$

When $p_0 = 0$ (this is the case for Adam), the bound in Theorem 2 coincides with the gradient descent bound, up to the constant $4/b^2$. We mention however that ε for Algorithm (1) is defined by a slightly more restrictive condition than for gradient descent : when $b = 1$, there is no momentum and $a_{\sup} = \frac{1}{L}(1 - 2\varepsilon) < 2/L$. Hence, under the boundedness of the effective step size, the algorithm has a similar convergence guarantee to gradient descent. Remark that the step size bound almost matches the classical $2/L$ upper-bound on the step size of gradient descent (see for example Nesterov (2004, Theorem 2.1.14)).

Stepsize bound. Condition 5 should be seen as a clipping step of the algorithm. Indeed, the lower bound on the effective stepsize has not to be verified a posteriori after running the algorithm. Instead, a clipping of the learning rate would ensure that this boundedness assumption holds. Furthermore, if we drop the lower bound assumption on the effective step size a_n from Theorem 2, we still get the following result (see Theorem 14), for all $n \geq 1$,

$$\frac{1}{n} \sum_{k=0}^{n-1} \langle a_{k+1}, \nabla f(x_k)^2 \rangle \leq \frac{2(1+\alpha)}{nb^2\alpha} \left(\frac{H_0 - \inf f}{\varepsilon} + \langle a_0, p_0^2 \rangle \right).$$

Influence of ε and δ. In the specific case of Adam, we obtain $La_{\sup}/2 + \varepsilon = 0.93$ with the recommended default parameters $b = 0.1$ and $c = 0.001$. Hence, we can choose ε of the order of 0.1 without exceeding 0.93. In view of Equation (6), the smaller is ε and the larger will be the stepsizes. However, a small ε deteriorates the bounds of Theorems 2 and 3. Once b, c (and then α) are fixed, ε can be seen as a constant. The clipping parameter δ can also be seen as constant once it is chosen.

4.2. Stochastic setting

We establish a similar bound in the stochastic setting. Note that the control of the minimum of the gradients norms is also standard in nonconvex stochastic optimization literature (see for example Ghadimi and Lan (2013)). Let $(\Xi, \mathfrak{S})$ denote a measurable space and $d \in \mathbb{N}$. Consider the problem of finding a local minimizer of the expectation $F(x) := \mathbb{E}(f(x,\xi))$ w.r.t. $x \in \mathbb{R}^d$, where $f : \mathbb{R}^d \times \Xi \to \mathbb{R}$ is a measurable map and $f(\,.\,,\xi)$ is a possibly nonconvex function depending on some random variable ξ. The distribution of ξ is assumed to be unknown, but revealed online by the observation of iid copies $(\xi_n : n \geq 1)$ of the r.v. ξ. For a fixed value of ξ, the mapping $x \mapsto f(x,\xi)$ is supposed to be differentiable, and its gradient w.r.t. x is denoted by $\nabla f(x,\xi)$. We study a stochastic version of Algorithm (1) by replacing the deterministic gradient $\nabla f(x_n)$ by $\nabla f(x_n, \xi_{n+1})$.

Theorem 3 *Let Assumption 1 (for F) and Assumption 2 hold true. Assume the following bound on the variance in stochastic gradients: $\mathbb{E}\|\nabla f(x,\xi) - \nabla F(x)\|^2 \leq \sigma^2$ for all $x \in \mathbb{R}^d$. Suppose moreover that $1 - \alpha < b \leq 1$. Let $\varepsilon > 0$ s.t. $\bar{a}_{\sup} := \frac{2}{L}\left(\frac{3}{4} - \frac{(b-(1-\alpha))^2}{2b\alpha} - \frac{1-\alpha}{2b} - \varepsilon\right)$ is nonnegative. Let $\delta > 0$ s.t. for all $n \geq 1$, almost surely,*

$$\delta \leq a_{n+1} \leq \min\left(\bar{a}_{\sup}, \frac{a_n}{\alpha}\right). \tag{6}$$

Then,

$$\mathbb{E}[\|\nabla F(x_\tau)\|^2] \leq \frac{4}{nb^2}\left(\frac{H_0 - \inf f}{\delta\varepsilon} + \|p_0\|^2\right) + \frac{4\bar{a}_{\sup}}{\delta\varepsilon b^2}\sigma^2,$$

where x_τ is an iterate uniformly randomly chosen from $\{x_0, \cdots, x_{n-1}\}$.

Remark 4 *We recover the deterministic bound of Theorem 2 when the gradients are noiseless ($\sigma = 0$). The complete proof is deferred to Appendix A.5.*

Before proceeding, a few remarks are in order.

SGD as a particular case. By setting $b = 1$ (no momentum) and $a_{n+1} = a_n$ for all n which implies $\alpha = 1$, we recover a known rate for nonconvex SGD (Ghadimi and Lan, 2013) with a maximal stepsize here of $\bar{a}_{\sup} = \frac{1}{2L}(1 - 2\varepsilon)$ and note that the proof can be slightly modified to make $\bar{a}_{\sup}$ as close as possible to $1/L$. We highlight though that the Lyapunov function H was especially tailored to handle a momentum algorithm and an analysis with f as a Lyapunov function is largely satisfying for SGD.

RMSProp. In the particular case where there is no momentum in the algorithm (i.e. RMSProp) and assuming that the gradients are bounded, a similar convergence rate is obtained in Zaheer et al. (2018, Thm. 1) (see Appendix A.1). Furthermore, although we assume boundedness of the step size by Condition (6), we do not suppose that $a_1 \leq \frac{\epsilon}{2L}$ (see table in Appendix A.1). The latter assumption imposes a very small step size ($\epsilon = 10^{-8}$ in Kingma and Ba (2015)) which may result in a slow convergence.

Stepsize lower bound. In the case of Adam ($a_n = \frac{a}{\epsilon + \sqrt{v_n}}$), the uniform lower bound $a_{n+1} \geq \delta$ prevents the exponential moving average v_n of the squared gradients from exploding.

This can be guaranteed on the fly by a clipping of a_n. If we drop the uniform lower bound on the effective step size, we still obtain the following result (see Appendix. Theorem 15)

$$\mathbb{E}\left[\sum_{k=0}^{n-1}\langle a_{k+1}, \nabla f(x_k, \xi_{k+1})^2\rangle\right] \leq \frac{2(1+\alpha)}{b^2\alpha}\left(\frac{H_0 - \inf f}{\varepsilon} + \langle a_0, p_0^2\rangle + \frac{n\bar{a}_{\sup}\sigma^2}{\varepsilon}\right).$$

Influence of the momentum parameter. Note that ε depends on the momentum parameter b and consequently the bound does not decrease with b. The influence of this parameter is more complex.

5. Convergence Analysis under the KŁ Property

Historically introduced by the fundamental works of Łojasiewicz (1963) and Kurdyka (1998), the KŁ inequality is the key tool of our analysis. We refer to Bolte et al. (2010) for an in-depth presentation of this property. The KŁ inequality is satisfied by a broad class of functions including most nonsmooth deep neural networks. More precisely, as exposed in Davis et al. (2019, Section 5.2, Corollary 5.11) and Castera et al. (2019, Section 2.2), feedforward neural networks with arbitrary number of layers of arbitrary dimensions, with activations such as sigmoid, ReLU, leaky ReLU, tanh, softplus (and many others), with a loss function such as l_p norm, hinge loss, logistic loss or cross entropy (and many others), belong to this class of so-called *definable* functions in an *o-minimal structure* (Kurdyka, 1998; Attouch et al., 2010; Davis et al., 2019). We refer the interested reader to Zeng et al. (2019, Section 3, Section C) for general conditions for which KŁ inequality holds in the context of deep neural networks training models. The class of *definable* functions is stable under all the typical functional operations in optimization (e.g. sums, compositions, inf-projections) and generalizes the class of semialgebraic functions including objective functions such as $\|\cdot\|_p$ for p rational, real polynomials, rank, etc. (see Bolte et al. (2014, Appendix)).

The KŁ inequality has been used to show the convergence of several first-order optimization methods towards critical points (Attouch and Bolte, 2009; Attouch et al., 2010; Bolte et al., 2014; Li et al., 2017). In this section, we use a methodology exposed in Bolte et al. (2018, Appendix) to show convergence rates based on the KŁ property. Recently developed in Bolte et al. (2014), this abstract convergence mechanism can be used for any *descent* type algorithm. We modify it to encompass momentum methods. Note that although this modification was initiated in Ochs et al. (2014); Ochs (2018), we use a different separable Lyapunov function. The first part of the proof follows these approaches and the second part follows the proof of Johnstone and Moulin (2017, Theorem 2).

Consider the function $H : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ defined for all $z = (x, y) \in \mathbb{R}^d \times \mathbb{R}^d$ by

$$H(z) = H(x, y) = f(x) + \frac{1}{2b}\|y\|^2. \tag{7}$$

Notice that $H_n = f(x_n) + \frac{1}{2b}\langle a_n, p_n^2\rangle = H(x_n, y_n)$ where $(y_n)_{n\in\mathbb{N}}$ is defined for all $n \in \mathbb{N}$ by $y_n = \sqrt{a_n}p_n$.

Notations and definitions. If (E, d) is a metric space, $z \in E$ and A is a non-empty subset of E, we use the notation $\mathsf{d}(z, A) := \inf\{\mathsf{d}(z, z') : z' \in A\}$. The set of critical points of the function H is defined by $\operatorname{crit} H := \{z \in \mathbb{R}^{2d} \text{ s.t. } \nabla H(z) = 0\}$.

Assumption 3 *f is coercive, that is $f(x) \to +\infty$ as $\|x\| \to +\infty$.*

Assumption 3 will be particularly useful to ensure that the sequence of the iterates $(z_k)_{k\geq 0}$ of Algorithm (1) is bounded. Indeed, a coercive function has compact level sets and Lemma 1 will guarantee that the iterates lie in a level set of the function H.

We now introduce the limit point set of the sequence $(z_k)_{k\geq 0}$ and exhibit some of its properties.

Definition 5 ***(Limit point set)*** *The set of all limit points of $(z_k)_{k\in\mathbb{N}}$ initialized at z_0 is defined by*

$$\omega(z_0) := \{\bar{z} \in \mathbb{R}^{2d} : \exists \text{ an increasing sequence of integers } (k_j)_{j\in\mathbb{N}} \text{ s.t } z_{k_j} \to \bar{z} \text{ as } j \to \infty\}.$$

Lemma 6 ***(Properties of the limit point set)*** *Let $(z_k)_{k\in\mathbb{N}}$ be the sequence defined for all $k \in \mathbb{N}$ by $z_k = (x_k, y_k)$ where $y_k = \sqrt{a_k}p_k$ and (x_k, p_k) is generated by Algorithm (1) from a starting point z_0. Let Assumptions 1 to 3 hold true. Assume that Condition (5) holds. Then,*

(i) $\omega(z_0)$ *is a nonempty compact set.*
(ii) $\omega(z_0) \subset critH = critf \times \{0\}$.
(iii) $\lim_{k\to+\infty} \mathsf{d}(z_k, \omega(z_0)) = 0$.
(iv) H *is finite and constant on* $\omega(z_0)$.

We introduce the KŁ inequality in the following. Define $[\alpha < H < \beta] := \{z \in \mathbb{R}^{2d} : \alpha < H(z) < \beta\}$. Let $\eta > 0$ and define Φ_η as the set of continuous functions φ on $[0, \eta)$ which are also continuously differentiable on $(0, \eta)$, concave and satisfy $\varphi(0) = 0$ and $\varphi' > 0$.

Definition 7 ***(KŁ property, Bolte et al. (2018, Appendix))*** *A proper and lower semicontinuous (l.s.c) function $H : \mathbb{R}^{2d} \to (-\infty, +\infty]$ has the KŁ property locally at $\bar{z} \in dom\, H$ if there exist $\eta > 0$, $\varphi \in \Phi_\eta$ and a neighborhood $U(\bar{z})$ s.t. for all $z \in U(\bar{z}) \cap [H(\bar{z}) < H < H(\bar{z}) + \eta]$:*

$$\varphi'(H(z) - H(\bar{z}))\, \|\nabla H(z)\| \geq 1. \tag{8}$$

When $H(\bar{z}) = 0$, we can rewrite Equation (8) as : $\|\nabla(\varphi \circ H)(z)\| \geq 1$ for suitable z points. This means that H becomes sharp under a reparameterization of its values through the so-called desingularizing function φ.

The function H is said to be a KŁ function if it has the KŁ property at each point of the domain of its gradient. Note that this property can be defined for nonsmooth functions using the Clarke subdifferential in order to encompass nonsmooth neural networks. We limit ourserlves to the simpler differentiable setting. KŁ inequality holds at any non critical point (see Attouch et al. (2010, Remark 3.2 (b))). We introduce now a uniformized version of the KŁ property which will be useful for our analysis.

Lemma 8 ***(Uniformized KŁ property, Bolte et al. (2014, Lemma 6, p 478))*** *Let Ω be a compact set and let $H : \mathbb{R}^{2d} \to (-\infty, +\infty]$ be a proper l.s.c function. Assume that H is constant on Ω and satisfies the KŁ property at each point of Ω. Then, there exist $\varepsilon > 0, \eta > 0$ and $\varphi \in \Phi_\eta$ such that for all $\bar{z} \in \Omega$, for all $z \in \{z \in \mathbb{R}^d : \mathsf{d}(z, \Omega) < \varepsilon\} \cap [H(\bar{z}) < H < H(\bar{z}) + \eta]$, one has*

$$\varphi'(H(z) - H(\bar{z}))\|\nabla H(z)\| \geq 1 \tag{9}$$

Definition 9 ***(KŁ exponent)*** *If φ can be chosen as $\varphi(s) = \frac{\bar{c}}{\theta} s^\theta$ for some $\bar{c} > 0$ and $\theta \in (0, 1]$ in Theorem 7, then we say that H has the KŁ property at $\bar{z}$ with an exponent of θ* [1]*. We say that H is a KŁ function with an exponent θ if it has the same exponent θ at any $\bar{z}$.*

In the particular case when $\theta = 1/2$, we recover the Polyak-Łojasiewicz condition (see for example Karimi et al. (2016)) satisfied for strongly convex functions. Furthermore, if H is a proper closed semialgebraic function, then H is a KŁ function with a suitable exponent $\theta \in (0, 1]$. The slope of φ around the origin informs about the "flatness" of a function around a point. Hence, the KŁ exponent allows to obtain convergence rates. In the light of this remark, we state one of the main results of this work.

Theorem 10 ***(Convergence rates)*** *Let $(z_k)_{k\in\mathbb{N}}$ be the sequence defined for all $k \in \mathbb{N}$ by $z_k = (x_k, y_k)$ where $y_k = \sqrt{a_k} p_k$ and (x_k, p_k) is generated by Algorithm (1) from a starting point z_0. Let Assumptions 1 to 3 hold true. Assume that Condition (5) holds. Suppose moreover that H is a KŁ function with KŁ exponent θ. Then, the sequence $(H(z_k))_{k\in\mathbb{N}}$ converges to $f(x_*)$ where x_* is a critical point of f and the following convergence rates hold:*

(i) If $\theta = 1$, then $f(x_k)$ converges in a finite number of iterations.
(ii) If $1/2 \leq \theta < 1$, then $f(x_k)$ converges to $f(x_*)$ linearly i.e. there exist $q \in (0, 1), C > 0$ s.t. $f(x_k) - f(x_*) \leq C\, q^k$.
(iii) If $0 < \theta < 1/2$, then $f(x_k) - f(x_*) = O(k^{\frac{1}{2\theta-1}})$.

The exact same rates hold for gradient descent by supposing that f (instead of H) is KŁ with exponent θ. Assumption 2 and condition (5) are not needed in this case.

Sketch of the proof. The proof consists of two main steps. The first one is to show that the iterates enter and stay in a region where the KŁ inequality holds. This is achieved using the properties of the limit set (Lemma 6) and the uniformized KŁ property (Lemma 8). Then, the second step is to exploit this inequality to derive the sought convergence results. We defer the complete proof to Appendix B.3.

We introduce a lemma in order to make the KŁ assumption on the objective function f instead of the function H.

Lemma 11 *Let f be a continuously differentiable function satisfying the KŁ property at $\bar{x}$ with an exponent of $\theta \in (0, 1/2]$. Then the function H defined in Equation (7) has also the KŁ property at $(\bar{x}, 0)$ with an exponent of θ.*

The following result derives a convergence rate on the objective function values under a KŁ assumption on this same function instead of an assumption on the Lyapunov function H. The result is an immediate consequence of Lemma 11 and Theorem 10.

Corollary 12 *Let $(z_k)_{k\in\mathbb{N}}$ be the sequence defined for all $k \in \mathbb{N}$ by $z_k = (x_k, y_k)$ where $y_k = \sqrt{a_k} p_k$ and (x_k, p_k) is generated by Algorithm (1) from a starting point z_0. Let Assumptions 1 to 3 hold true. Assume that Condition (5) holds. Suppose moreover that f is a KŁ function with KŁ exponent $\theta \in (0, 1/2)$. Then, the sequence $(H(z_k))_{k\in\mathbb{N}}$ converges to $f(x_*)$ where x_* is a critical point of f and $f(x_k) - f(x_*) = O(k^{\frac{1}{2\theta-1}})$.*

1. $\alpha := 1 - \theta$ is also defined as the KŁ exponent in other papers (Li and Pong, 2018).

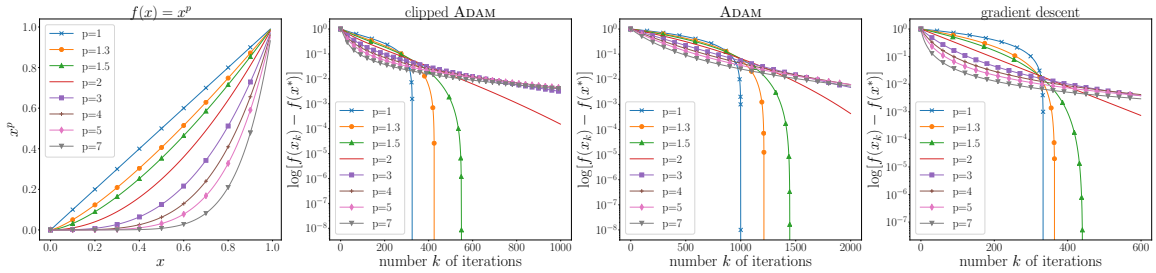

Figure 1: Illustration of KŁ rates for a simple objective function $f(x) = x^p$. From left to right : (i) curves of $f(x) = x^p$, (ii) clipped version of Adam (see Algorithm (1)), (iii) Adam ,(iv) Gradient Descent. Best seen in color.

5.1. Toy problem : KŁ rates for $f(x) = x^p$.

KŁ rates are *asymptotic* rates in the sense that the constants cannot be explicited in the convergence rates. As a consequence, the rates can be hardly observable in practice from experiments. However, we can still illustrate these convergence results (Theorem 10) in a simple toy example to give more insight. Consider the problem of minimizing the function $f(x) = x^p$ for a real $p \in [1, 7]$. One can easily show that f is a KŁ function with KŁ exponent $\theta = \frac{1}{p}$. Note that the KŁ exponent is difficult to compute in general. This justifies the choice of this toy problem. Moreover, even if the function f is indeed convex, we recall the reader that the KŁ property is a local geometric property of the function that is only interesting at its critical points (since it is automatically verified at any non critical point). Notice that the KŁ analysis is valid in the general nonconvex case. The present toy example remains relevant if we modify the objective function f to be nonconvex and still keep a x^p shape in a neighborhood of the point zero which is the unique critical point in this example.

The KŁ exponent as shown in the first plot in Figure 1 encodes information about the flatness of the function f. Indeed, as p increases, the function f gets flatter around the origin $x = 0$. We run the clipped version of Adam (see Algorithm 1), the Adam algorithm and gradient descent on the functions f corresponding to different values of the exponent θ, from the same initialization point $x = 1$. As expected from Theorem 10 for the clipped Adam , we observe in Figure 1 that $f(x_k)$ converges linearly or even in a finite number of iterations for $p \in \{1, 1.3, 1.5, 2\}$. Notice that the linear rate is clearly observable for $p = 2$ corresponding to $\theta = \frac{1}{2}$. Even if we did not establish KŁ rates for original Adam , Figure 1 shows that it presents a very similar behavior to the clipped version of Adam in terms of KŁ convergence rates in this simple problem. We also represent gradient descent iterates for comparison. Note that KŁ rates are known to hold for gradient descent. Moreover, for $p > 2$, we also observe a slower rate corresponding to the sublinear rate of the function values.

6. Conclusion

In this paper, we provided convergence rates for a clipped version of Adam which stems from a boundedness assumption on the effective stepsize of the original Adam . More precisely, similarly to gradient descent, we established a $O(1/n)$ convergence rate of the minimum of the squared gradient norms in the deterministic case. Furthermore, we showed a similar convergence result in the stochastic setting up to the variance of the noisy gradients. Finally,

we established function value convergence rates under the same boundedness assumption on the effective stepsizes together with the KŁ geometric property. This property is a powerful tool allowing to address nonconvex nonsmooth optimization and covers most deep neural networks.

Acknowledgments

We thank the anonymous reviewers for their helpful comments. A.B. was supported by the 'Futur & Ruptures' research program which is jointly funded by the IMT, the Mines-Télécom Foundation and the Carnot TSN Institute.

References

P-A. Absil, R. Mahony, and B. Andrews. Convergence of the iterates of descent methods for analytic cost functions. *SIAM Journal on Optimization*, 16(2):531–547, 2005.

N. Agarwal, B. Bullins, X. Chen, E. Hazan, K. Singh, C. Zhang, and Y. Zhang. Efficient full-matrix adaptive regularization. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 102–110, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL http://proceedings.mlr.press/v97/agarwal19b.html.

H. Attouch and J. Bolte. On the convergence of the proximal algorithm for nonsmooth functions involving analytic features. *Mathematical Programming*, 116(1-2):5–16, 2009.

H. Attouch, J. Bolte, P. Redont, and A. Soubeyran. Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the kurdyka-łojasiewicz inequality. *Mathematics of Operations Research*, 35(2):438–457, 2010.

J. Bolte, A. Daniilidis, O. Ley, and L. Mazet. Characterizations of łojasiewicz inequalities: subgradient flows, talweg, convexity. *Transactions of the American Mathematical Society*, 362(6):3319–3363, 2010.

J. Bolte, S. Sabach, and M. Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1-2):459–494, 2014.

J. Bolte, S. Sabach, M. Teboulle, and Y. Vaisbourd. First order methods beyond convexity and lipschitz gradient continuity with applications to quadratic inverse problems. *SIAM Journal on Optimization*, 28(3):2131–2151, 2018.

L. Bottou, F. Curtis, and J. Nocedal. Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311, 2018.

C. Castera, J. Bolte, C. Févotte, and E. Pauwels. An inertial newton algorithm for deep learning. *arXiv preprint arXiv:1905.12278*, 2019.

J. Chen, D. Zhou, Y. Tang, Z. Yang, and Q. Gu. Closing the generalization gap of adaptive gradient methods in training deep neural networks. *arXiv preprint arXiv:1806.06763*, 2018.

X. Chen, S. Liu, R. Sun, and M. Hong. On the convergence of a class of adam-type algorithms for non-convex optimization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=H1x-x309tm.

D. Davis, D. Drusvyatskiy, S. Kakade, and J.D. Lee. Stochastic subgradient method converges on tame functions. *Foundations of Computational Mathematics*, pages 1–36, 2019.

S. De, A. Mukherjee, and E. Ullah. Convergence guarantees for rmsprop and adam in non-convex optimization and their comparison to nesterov acceleration on autoencoders. *arXiv preprint arXiv:1807.06766*, 2018.

J. Diakonikolas and M. I. Jordan. Generalized momentum-based methods: A hamiltonian perspective. *arXiv preprint arXiv:1906.00436*, 2019.

T. Dozat. Incorporating nesterov momentum into adam. 2016.

J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

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

V. Gupta, T. Koren, and Y. Singer. A unified approach to adaptive regularization in online and stochastic optimization. *arXiv preprint arXiv:1706.06569*, 2017.

P. R. Johnstone and P. Moulin. Convergence rates of inertial splitting schemes for nonconvex composite optimization. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4716–4720. IEEE, 2017.

Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 795–811. Springer, 2016.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

K. Kurdyka. On gradients of functions definable in o-minimal structures. In *Annales de l'institut Fourier*, volume 48, pages 769–783, 1998.

G. Li and T. K. Pong. Calculus of the exponent of kurdyka–łojasiewicz inequality and its applications to linear convergence of first-order methods. *Foundations of computational mathematics*, 18(5):1199–1232, 2018.

Q. Li, Y. Zhou, Y. Liang, and P. K. Varshney. Convergence analysis of proximal gradient with momentum for nonconvex optimization. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2111–2119. JMLR. org, 2017.

X. Li and F. Orabona. On the convergence of stochastic gradient descent with adaptive stepsizes. In *Proceedings of Machine Learning Research*, volume 89, pages 983–992. PMLR, 16–18 Apr 2019. URL http://proceedings.mlr.press/v89/li19c.html.

J. Liang, J. Fadili, and G. Peyré. A multi-step inertial forward-backward splitting method for non-convex optimization. In *Advances in Neural Information Processing Systems*, pages 4035–4043, 2016.

L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han. On the variance of the adaptive learning rate and beyond. *arXiv preprint arXiv:1908.03265*, 2019.

S. Łojasiewicz. Une propriété topologique des sous-ensembles analytiques réels. *Les équations aux dérivées partielles*, 117:87–89, 1963.

L. Luo, Y. Xiong, and Y. Liu. Adaptive gradient methods with dynamic bound of learning rate. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg3g2R9FX.

J. Ma and D. Yarats. Quasi-hyperbolic momentum and adam for deep learning. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=S1fUpoR5FQ.

H. B. McMahan and M. J. Streeter. Adaptive bound optimization for online convex optimization. In *COLT*, pages 244–256, 2010.

Y. Nesterov. *Introductory lectures on convex optimization: a basic course*. Springer: New York, NY, USA, 2004.

P. Ochs. Local convergence of the heavy-ball method and ipiano for non-convex optimization. *Journal of Optimization Theory and Applications*, 177(1):153–180, 2018.

P. Ochs, Y. Chen, T. Brox, and T. Pock. ipiano: Inertial proximal algorithm for nonconvex optimization. *SIAM Journal on Imaging Sciences*, 7(2):1388–1419, 2014. doi: 10.1137/130942954. URL https://doi.org/10.1137/130942954.

R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318, 2013.

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

S. J. Reddi, S. Kale, and S. Kumar. On the convergence of adam and beyond. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=ryQu7f-RZ.

H. Robbins and S. Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

P. Savarese. On the convergence of adabound and its connection to sgd. *arXiv preprint arXiv:1908.04457*, 2019.

M. Staib, S. Reddi, S. Kale, S. Kumar, and S. Sra. Escaping saddle points with adaptive gradient methods. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 5956–5965, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL http://proceedings.mlr.press/v97/staib19a.html.

I. Sutskever, J. Martens, G. Dahl, and G. Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147, 2013.

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*, 4(2):26–31, 2012.

R. Ward, X. Wu, and L. Bottou. Adagrad stepsizes: Sharp convergence over nonconvex landscapes. In *International Conference on Machine Learning*, pages 6677–6686, 2019.

X. Wu, R. Ward, and L. Bottou. Wngrad: Learn the learning rate in gradient descent. *arXiv preprint arXiv:1803.02865*, 2018.

Z. Wu and M. Li. General inertial proximal gradient method for a class of nonconvex nonsmooth optimization problems. *Computational Optimization and Applications*, 73(1): 129–158, 2019.

Y. Xie, X. Wu, and R. Ward. Linear convergence of adaptive stochastic gradient descent. *arXiv preprint arXiv:1908.10525*, 2019.

M. Zaheer, S. Reddi, D. Sachan, S. Kale, and S. Kumar. Adaptive methods for nonconvex optimization. In *Advances in Neural Information Processing Systems*, pages 9793–9803, 2018.

J. Zeng, T. T. Lau, S. Lin, and Y. Yao. Global convergence of block coordinate descent in deep learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 7313–7323, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL http://proceedings.mlr.press/v97/zeng19a.html.

J. Zhang, T. He, S. Sra, and A. Jadbabaie. Why gradient clipping accelerates training: A theoretical justification for adaptivity. In *International Conference on Learning Representations*, 2019.

D. Zhou, Y. Tang, Z. Yang, Y. Cao, and Q. Gu. On the convergence of adaptive gradient methods for nonconvex optimization. *arXiv preprint arXiv:1808.05671*, 2018.

Z. Zhou, Q. Zhang, G. Lu, H. Wang, W. Zhang, and Y. Yu. Adashift: Decorrelation and convergence of adaptive learning rate methods. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HkgTkhRcKQ.

F. Zou, L. Shen, Z. Jie, W. Zhang, and W. Liu. A sufficient condition for convergences of adam and rmsprop. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11127–11135, 2019.