# A Simple Convergence Proof of Adam and Adagrad

**Anonymous authors**
**Paper under double-blind review**

## Abstract

We provide a simple proof of convergence covering both the Adam and Adagrad adaptive optimization algorithms when applied to smooth (possibly non-convex) objective functions with bounded gradients. We show that in expectation, the squared norm of the objective gradient averaged over the trajectory has an upper-bound which is explicit in the constants of the problem, parameters of the optimizer, the dimension $d$, and the total number of iterations $N$. This bound can be made arbitrarily small, and with the right hyper-parameters, Adam can be shown to converge with the same rate of convergence $O(d\ln(N)/\sqrt{N})$. When used with the default parameters, Adam doesn't converge, however, and just like constant step-size SGD, it moves away from the initialization point faster than Adagrad, which might explain its practical success. Finally, we obtain the tightest dependency on the heavy ball momentum decay rate $\beta_1$ among all previous convergence bounds for non-convex Adam and Adagrad, improving from $O((1-\beta_1)^{-3})$ to $O((1-\beta_1)^{-1})$.

## 1 Introduction

First-order methods with adaptive step sizes have proved useful in many fields of machine learning, be it for sparse optimization (Duchi et al., 2013), tensor factorization (Lacroix et al., 2018) or deep learning (Goodfellow et al., 2016). Duchi et al. (2011) introduced Adagrad, which rescales each coordinate by a sum of squared past gradient values. While Adagrad proved effective for sparse optimization (Duchi et al., 2013), experiments showed that it under-performed when applied to deep learning (Wilson et al., 2017). RMSProp (Tieleman & Hinton, 2012) proposed an exponential moving average instead of a cumulative sum to solve this. Kingma & Ba (2015) developed Adam, one of the most popular adaptive methods in deep learning, built upon RMSProp and added corrective terms at the beginning of training, together with heavy-ball style momentum.

In the online convex optimization setting, Duchi et al. (2011) showed that Adagrad achieves optimal regret for online convex optimization. Kingma & Ba (2015) provided a similar proof for Adam when using a decreasing overall step size, although this proof was later shown to be incorrect by Reddi et al. (2018), who introduced AMSGrad as a convergent alternative. Ward et al. (2019) proved that Adagrad also converges to a critical point for non convex objectives with a rate $O(\ln(N)/\sqrt{N})$ when using a scalar adaptive step-size, instead of diagonal. Zou et al. (2019b) extended this proof to the vector case, while Zou et al. (2019a) displayed a bound for Adam, showing convergence when the decay of the exponential moving average scales as $1-1/N$ and the learning rate as $1/\sqrt{N}$.

In this paper, we present a simplified and unified proof of convergence to a critical point for Adagrad and Adam for stochastic non-convex smooth optimization. We assume that the objective function is lower bounded, smooth and the stochastic gradients are almost surely bounded. We recover the standard $O(\ln(N)/\sqrt{N})$ convergence rate for Adagrad for all step sizes, and the same rate with Adam with an appropriate choice of the step sizes and decay parameters, in particular, Adam can converge without using the AMSGrad variant. Compared to previous work, our bound significantly improves the dependency on the momentum parameter $\beta_1$. The best known bounds for Adagrad and Adam are respectively in $O((1-\beta_1)^{-3})$ and $O((1-\beta_1)^{-5})$ (see Section 3), while our result is in $O((1-\beta_1)^{-1})$ for both algorithms. This improvement is a step toward understanding the practical efficiency of heavy-ball momentum.

**Outline.** The precise setting and assumptions are stated in the next section, and previous work is then described in Section 3. The main theorems are presented in Section 4, followed by a full proof for the case without momentum in Section 5. The proof of the convergence with momentum is deferred to the supplementary material, Section A. Finally we compare our bounds with experimental results, both on toy and real life problems in Section 6.

## 2 Setup

### 2.1 Notation

Let $d \in \mathbb{N}$ be the dimension of the problem (i.e. the number of parameters of the function to optimize) and take $[d] = \{1, 2, \dots, d\}$. Given a function $h : \mathbb{R}^d \to \mathbb{R}$, we denote by $\nabla h$ its gradient and $\nabla_i h$ the $i$-th component of the gradient. We use a small constant $\epsilon$, e.g. $10^{-8}$, for numerical stability. Given a sequence $(u_n)_{n\in\mathbb{N}}$ with $\forall n \in \mathbb{N}, u_n \in \mathbb{R}^d$, we denote $u_{n,i}$ for $n \in \mathbb{N}$ and $i \in [d]$ the $i$-th component of the $n$-th element of the sequence.

We want to optimize a function $F : \mathbb{R}^d \to \mathbb{R}$. We assume there exists a random function $f : \mathbb{R}^d \to \mathbb{R}$ such that $\mathbb{E}\left[\nabla f(x)\right] = \nabla F(x)$ for all $x \in \mathbb{R}^d$, and that we have access to an oracle providing i.i.d. samples $(f_n)_{n\in\mathbb{N}^*}$. We note $\mathbb{E}_{n-1}\left[\cdot\right]$ the conditional expectation knowing $f_1, \dots, f_{n-1}$. In machine learning, $x$ typically represents the weights of a linear or deep model, $f$ represents the loss from individual training examples or minibatches, and $F$ is the full training objective function. The goal is to find a critical point of $F$.

### 2.2 Adaptive methods

We study both Adagrad (Duchi et al., 2011) and Adam (Kingma & Ba, 2015) using a unified formulation. We assume we have $0 < \beta_2 \leq 1$, $0 \leq \beta_1 < \beta_2$, and a non negative sequence $(\alpha_n)_{n\in\mathbb{N}^*}$. We define three vectors $m_n, v_n, x_n \in \mathbb{R}^d$ iteratively. Given $x_0 \in \mathbb{R}^d$ our starting point, $m_0 = 0$, and $v_0 = 0$, we define for all iterations $n \in \mathbb{N}^*$,

$$m_{n,i} = \beta_1 m_{n-1,i} + \nabla_i f_n(x_{n-1}) \tag{1}$$

$$v_{n,i} = \beta_2 v_{n-1,i} + \left(\nabla_i f_n(x_{n-1})\right)^2 \tag{2}$$

$$x_{n,i} = x_{n-1,i} - \alpha_n \frac{m_{n,i}}{\sqrt{\epsilon + v_{n,i}}}. \tag{3}$$

The parameter $\beta_1$ is a heavy-ball style momentum parameter (Polyak, 1964), while $\beta_2$ controls the decay rate of the per-coordinate exponential moving average of the squared gradients. Taking $\beta_1 = 0$, $\beta_2 = 1$ and $\alpha_n = \alpha$ gives Adagrad. While the original Adagrad algorithm did not include a heavy-ball-like momentum, our analysis also applies to the case $\beta_1 > 0$.

**Adam and its corrective terms** The original Adam algorithm (Kingma & Ba, 2015) uses a weighed average, rather than a weighted sum for (1) and (2), i.e. it uses

$$\tilde{m}_{n,i} = (1-\beta_1)\sum_{k=1}^{n} \beta_1^{n-k} \nabla_i f_n(x_{k-1}) = (1-\beta_1) m_{n,i},$$

We can achieve the same definition by taking $\alpha_{\text{adam}} = \alpha \cdot \frac{1-\beta_1}{\sqrt{1-\beta_2}}$. The original Adam algorithm further includes two corrective terms to account for the fact that $m_n$ and $v_n$ are biased towards 0 for the first few iterations. Those corrective terms are equivalent to taking a step-size $\alpha_n$ of the form

$$\alpha_{n,\text{adam}} = \alpha \cdot \frac{1-\beta_1}{\sqrt{1-\beta_2}} \cdot \underbrace{\frac{1}{1-\beta_1^n}}_{\substack{\text{corrective}\\ \text{term for } m_n}} \cdot \underbrace{\sqrt{1-\beta_2^n}}_{\substack{\text{corrective}\\ \text{term for } v_n}}. \tag{4}$$

Those corrective terms can be seen as the normalization factors for the weighted sum given by (1) and (2) Note that each term goes to its limit value within a few times $1/(1-\beta)$ updates (with $\beta \in \{\beta_1, \beta_2\}$). which

explains the $(1-\beta_1)$ term in (4). In the present work, we propose to drop the corrective term for $m_n$, and to keep only the one for $v_n$, thus using the alternative step size

$$\alpha_n = \alpha(1-\beta_1)\sqrt{\frac{1-\beta_2^n}{1-\beta_2}}. \tag{5}$$

This simplification motivated by several observations:

- By dropping either corrective terms, $\alpha_n$ becomes monotonic, which simplifies the proof.
- For typical values of $\beta_1$ and $\beta_2$ (e.g. 0.9 and 0.999), the corrective term for $m_n$ converges to its limit value much faster than the one for $v_n$.
- Removing the corrective term for $m_n$ is equivalent to a learning-rate warmup, which is popular in deep learning, while removing the one for $v_n$ would lead to an increased step size during early training. For values of $\beta_2$ close to 1, this can lead to divergence in practice.

We experimentally verify in Section 6.3 that dropping the corrective term for $m_n$ has no observable effect on the training process, while dropping the corrective term for $v_n$ leads to observable perturbations. In the following, we thus consider the variation of Adam obtained by taking $\alpha_n$ provided by (5).

### 2.3 Assumptions

We make three assumptions. We first assume $F$ is bounded below by $F_*$, that is,

$$\forall x \in \mathbb{R}^d,\ F(x) \geq F_*. \tag{6}$$

We then assume *the $\ell_\infty$ norm of the stochastic gradients is uniformly almost surely bounded*, i.e. there is $R \geq \sqrt{\epsilon}$ ($\sqrt{\epsilon}$ is used here to simplify the final bounds) so that

$$\forall x \in \mathbb{R}^d, \quad \|\nabla f(x)\|_\infty \leq R - \sqrt{\epsilon} \quad \text{a.s.}, \tag{7}$$

and finally, the *smoothness of the objective function*, e.g., its gradient is $L$-Liptchitz-continuous with respect to the $\ell_2$-norm:

$$\forall x, y \in \mathbb{R}^d, \quad \|\nabla F(x) - \nabla F(y)\|_2 \leq L \|x-y\|_2 . \tag{8}$$

We discuss the use of assumption (7) in Section 4.2.

## 3 Related work

Early work on adaptive methods (McMahan & Streeter, 2010; Duchi et al., 2011) showed that Adagrad achieves an optimal rate of convergence of $O(1/\sqrt{N})$ for convex optimization (Agarwal et al., 2009). Later, RMSProp (Tieleman & Hinton, 2012) and Adam (Kingma & Ba, 2015) were developed for training deep neural networks, using an exponential moving average of the past squared gradients.

Kingma & Ba (2015) offered a proof that Adam with a decreasing step size converges for convex objectives. However, the proof contained a mistake spotted by Reddi et al. (2018), who also gave examples of convex problems where Adam does not converge to an optimal solution. They proposed AMSGrad as a convergent variant, which consisted in retaining the maximum value of the exponential moving average. When $\alpha$ goes to zero, AMSGrad is shown to converge in the convex and non-convex setting (Fang & Klabjan, 2019; Zhou et al., 2018). Despite this apparent flaw in the Adam algorithm, it remains a widely popular optimizer, raising the question as to whether it converges. When $\beta_2$ goes to 1 and $\alpha$ to 0, our results and previous work (Zou et al., 2019a) show that Adam does converge with the same rate as Adagrad. This is coherent with the counter examples of Reddi et al. (2018), because they uses a small exponential decay parameter $\beta_2 < 1/5$.

The convergence of Adagrad for non-convex objectives was first tackled by Li & Orabona (2019), who proved its convergence, but under restrictive conditions (e.g., $\alpha \leq \sqrt{\epsilon}/L$). The proof technique was improved by Ward et al. (2019), who showed the convergence of "scalar" Adagrad, i.e., with a single learning rate, for any

value of $\alpha$ with a rate of $O(\ln(N)/\sqrt{N})$. Our approach builds on this work but we extend it to both Adagrad and Adam, in their coordinate-wise version, as used in practice, while also supporting heavy-ball momentum.

The coordinate-wise version of Adagrad was also tackled by Zou et al. (2019b), offering a convergence result for Adagrad with either heavy-ball or Nesterov style momentum. We obtain the same rate for heavy-ball momentum with respect to $N$ (i.e., $O(\ln(N)/\sqrt{N})$), but we improve the dependence on the momentum parameter $\beta_1$ from $O((1-\beta_1)^{-3})$ to $O((1-\beta_1)^{-1})$. Chen et al. (2019) also provided a bound for Adagrad and Adam, but without convergence guarantees for Adam for any hyper-parameter choice, and with a worse dependency on $\beta_1$. Zhou et al. (2018) also cover Adagrad in the stochastic setting, however their proof technique leads to a $\sqrt{1/\epsilon}$ term in their bound, typically with $\epsilon$=$10^{-8}$. Finally, a convergence bound for Adam was introduced by Zou et al. (2019a). We recover the same scaling of the bound with respect to $\alpha$ and $\beta_2$. However their bound has a dependency of $O((1-\beta_1)^{-5})$ with respect to $\beta_1$, while we get $O((1-\beta_1)^{-1})$, a significant improvement. Shi et al. (2020) obtain similar convergence results for RMSProp and Adam when considering the random shuffling setup. They use an affine growth condition (i.e. norm of the stochastic gradient is bounded by an affine function of the norm of the deterministic gradient) instead of the boundness of the gradient, but their bound decays with the number of total epochs, not stochastic updates leading to an overall $\sqrt{s}$ extra term with $s$ the size of the dataset. Finally, Faw et al. (2022) use the same affine growth assumption to derive high probability bounds for scalar Adagrad.

Non adaptive methods like SGD are also well studied in the non convex setting (Ghadimi & Lan, 2013), with a convergence rate of $O(1/\sqrt{N})$ for a smooth objective with bounded variance of the gradients. Unlike adaptive methods, SGD requires knowing the smoothness constant. When adding heavy-ball momentum, Yang et al. (2016) showed that the convergence bound degrades as $O((1-\beta_1)^{-2})$, assuming that the gradients are bounded. We apply our proof technique for momentum to SGD in the Appendix, Section B and improve this dependency to $O((1-\beta_1)^{-1})$. Recent work by Liu et al. (2020) achieves the same dependency with weaker assumptions. Defazio (2020) provided an in-depth analysis of SGD-M with a tight Liapunov analysis.

## 4 Main results

For a number of iterations $N \in \mathbb{N}^*$, we note $\tau_N$ a random index with value in $\{0, \dots, N-1\}$, so that

$$\forall j \in \mathbb{N}, j < N, \mathbb{P}\left[\tau = j\right] \propto 1 - \beta_1^{N-j}. \tag{9}$$

If $\beta_1 = 0$, this is equivalent to sampling $\tau$ uniformly in $\{0, \dots, N-1\}$. If $\beta_1 > 0$, the last few $\frac{1}{1-\beta_1}$ iterations are sampled rarely, and iterations older than a few times that number are sampled almost uniformly. Our results bound the expected squared norm of the gradient at iteration $\tau$, which is standard for non convex stochastic optimization (Ghadimi & Lan, 2013).

### 4.1 Convergence bounds

For simplicity, we first give convergence results for $\beta_1 = 0$, along with a complete proof in Section 5. We then provide the results with momentum, with their proofs in the Appendix, Section A.6. We also provide a bound on the convergence of SGD with a $O(1/(1-\beta_1)$ dependency in the Appendix, Section B.2, along with its proof in Section B.4.

**No heavy-ball momentum**

**Theorem 1** (Convergence of Adagrad without momentum)**.** *Given the assumptions from Section 2.3, the iterates $x_n$ defined in Section 2.2 with hyper-parameters verifying $\beta_2 = 1$, $\alpha_n = \alpha$ with $\alpha > 0$ and $\beta_1 = 0$, and $\tau$ defined by (9), we have for any $N \in \mathbb{N}^*$,*

$$\mathbb{E}\left[\|\nabla F(x_\tau)\|^2\right] \leq 2R\frac{F(x_0) - F_*}{\alpha\sqrt{N}} + \frac{1}{\sqrt{N}}\left(4dR^2 + \alpha dRL\right)\ln\left(1 + \frac{NR^2}{\epsilon}\right). \tag{10}$$

**Theorem 2** (Convergence of Adam without momentum)**.** *Given the assumptions from Section 2.3, the iterates $x_n$ defined in Section 2.2 with hyper-parameters verifying $0 < \beta_2 < 1$, $\alpha_n = \alpha\sqrt{\frac{1-\beta_2^n}{1-\beta_2}}$ with $\alpha > 0$*

*and* $\beta_1 = 0$*, and* $\tau$ *defined by* (9)*, we have for any* $N \in \mathbb{N}^*$*,*

$$\mathbb{E}\left[\|\nabla F(x_\tau)\|^2\right] \leq 2R\frac{F(x_0) - F_*}{\alpha N} + E\left(\frac{1}{N}\ln\left(1 + \frac{R^2}{(1-\beta_2)\epsilon}\right) - \ln(\beta_2)\right), \tag{11}$$

*with*

$$E = \frac{4dR^2}{\sqrt{1-\beta_2}} + \frac{\alpha dRL}{1-\beta_2}.$$

**With heavy-ball momentum**

**Theorem 3** (Convergence of Adagrad with momentum)**.** *Given the assumptions from Section 2.3, the iterates* $x_n$ *defined in Section 2.2 with hyper-parameters verifying* $\beta_2 = 1$*,* $\alpha_n = \alpha$ *with* $\alpha > 0$ *and* $0 \leq \beta_1 < 1$*, and* $\tau$ *defined by* (9)*, we have for any* $N \in \mathbb{N}^*$ *such that* $N > \frac{\beta_1}{1-\beta_1}$*,*

$$\mathbb{E}\left[\|\nabla F(x_\tau)\|^2\right] \leq 2R\sqrt{N}\frac{F(x_0) - F_*}{\alpha\tilde{N}} + \frac{\sqrt{N}}{\tilde{N}}E\ln\left(1 + \frac{NR^2}{\epsilon}\right), \tag{12}$$

*with* $\tilde{N} = N - \frac{\beta_1}{1-\beta_1}$*, and,*

$$E = \alpha dRL + \frac{12dR^2}{1-\beta_1} + \frac{2\alpha^2 dL^2\beta_1}{1-\beta_1}.$$

**Theorem 4** (Convergence of Adam with momentum)**.** *Given the assumptions from Section 2.3, the iterates* $x_n$ *defined in Section 2.2 with hyper-parameters verifying* $0 < \beta_2 < 1$*,* $0 \leq \beta_1 < \beta_2$*, and,* $\alpha_n = \alpha(1-\beta_1)\sqrt{\frac{1-\beta_2^n}{1-\beta_2}}$ *with* $\alpha > 0$*, and* $\tau$ *defined by* (9)*, we have for any* $N \in \mathbb{N}^*$ *such that* $N > \frac{\beta_1}{1-\beta_1}$*,*

$$\mathbb{E}\left[\|\nabla F(x_\tau)\|^2\right] \leq 2R\frac{F(x_0) - F_*}{\alpha\tilde{N}} + E\left(\frac{1}{\tilde{N}}\ln\left(1 + \frac{R^2}{(1-\beta_2)\epsilon}\right) - \frac{N}{\tilde{N}}\ln(\beta_2)\right), \tag{13}$$

*with* $\tilde{N} = N - \frac{\beta_1}{1-\beta_1}$*, and*

$$E = \frac{\alpha dRL(1-\beta_1)}{(1-\beta_1/\beta_2)(1-\beta_2)} + \frac{12dR^2\sqrt{1-\beta_1}}{(1-\beta_1/\beta_2)^{3/2}\sqrt{1-\beta_2}} + \frac{2\alpha^2 dL^2\beta_1}{(1-\beta_1/\beta_2)(1-\beta_2)^{3/2}}.$$

### 4.2 Analysis of the bounds

**Dependency on** $d$**.** The dependency in $d$ is present in previous works on coordinate wise adaptive methods (Zou et al., 2019a;b). Note however that $R$ is defined as the $\ell_\infty$ bound on the on the stochastic gradient, so that in the case where the gradient has a similar scale along all dimensions, $dR^2$ would be a reasonable bound for $\|\nabla f(x)\|_2^2$. However, if many dimensions contribute little to the norm of the gradient, this would still lead to a worse dependency in $d$ that e.g. scalar Adagrad Ward et al. (2019) or SGD.

Diving into the technicalities of the proof to come, we will see in Section 5 that we apply Lemma 5.2 once per dimension. The contribution from each coordinate is mostly independent of the actual scale of its gradients (as it only appears in the log), so that the right hand side of the convergence bound will grow as $d$. In contrast, the scalar version of Adagrad (Ward et al., 2019) has a single learning rate, so that Lemma 5.2 is only applied once, removing the dependency on $d$. However, this variant is rarely used in practice.

**Almost sure bound on the gradient.** We chose to assume the existence of an almost sure uniform $\ell_\infty$-bound on the gradients given by (7). This is a strong assumption, although it is weaker than the one used by Duchi et al. (2011) for Adagrad in the convex case, where the iterates were assumed to be almost surely bounded. There exist a few real life problems that verifies this assumption, for instance logistic regression without weight penalty, and with bounded inputs. It is possible instead to assume only a uniform bound on the expected gradient $\nabla F(x)$, as done by Ward et al. (2019) and Zou et al. (2019b). This however lead to a bound on $\mathbb{E}\left[\|\nabla F(x_\tau)\|_2^{4/3}\right]^{2/3}$ instead of a bound on $\mathbb{E}\left[\|\nabla F(x_\tau)\|_2^2\right]$, all the other terms staying the same. We provide the sketch of the proof using Hölder inequality in the Appendix, Section A.7.

It is also possible to replace the bound on the gradient with an affine growth condition, i.e. the norm of the stochastic gradient is bounded by an affine function of the norm of the expected gradient. A proof for scalar Adagrad is provided by Faw et al. (2022). Shi et al. (2020) do the same for RMSProp, however their convergence bound is decays as $O(\log(T)/\sqrt{T})$ with $T$ the number of epoch, not the number of updates, leading to a significantly less tight bound for large datasets.

**Impact of heavy-ball momentum.** Looking at Theorems 3 and 4, we see that increasing $\beta_1$ always deteriorates the bounds. Taking $\beta_1 = 0$ in those theorems gives us almost exactly the bound without heavy-ball momentum from Theorems 1 and 2, up to a factor 3 in the terms of the form $dR^2$.

As discussed in Section 3, previous bounds for Adagrad in the non-convex setting deteriorates as $O((1-\beta_1)^{-3})$ (Zou et al., 2019b), while bounds for Adam deteriorates as $O((1-\beta_1)^{-5})$ (Zou et al., 2019a). Our unified proof for Adam and Adagrad achieves a dependency of $O((1-\beta_1)^{-1})$, a significant improvement. We refer the reader to the Appendix, Section A.3, for a detailed analysis. While our dependency still contradicts the benefits of using momentum observed in practice, see Section 6, our tighter analysis is a step in the right direction.

**On sampling of $\tau$** Note that in (9), we sample with a lower probability the latest iterations. This can be explained by the fact that the proof technique for stochastic optimization in the non-convex case is based on the idea that for every iteration $n$, either $\nabla F(x_n)$ is small, or $F(x_{n+1})$ will decrease by some amount. However, when introducing momentum, and especially when taking the limit $\beta_1 \to 1$, the latest gradient $\nabla F(x_n)$ has almost no influence over $x_{n+1}$, as the momentum term updates slowly. Momentum *spreads* the influence of the gradients over time, and thus, it will take a few updates for a gradient to have fully influenced the iterate $x_n$ and thus the value of the function $F(x_n)$. From a formal point of view, the sampling weights given by (9) naturally appear as part of the proof which is presented in Section A.6.

### 4.3 Optimal finite horizon Adam is Adagrad

Let us take a closer look at the result from Theorem 2. It could seem like some quantities can explode but actually not for any reasonable values of $\alpha$, $\beta_2$ and $N$. Let us try to find the best possible rate of convergence for Adam for a finite horizon $N$, i.e. $q \in \mathbb{R}_+$ such that $\mathbb{E}\left[\|\nabla F(x_\tau)\|^2\right] = O(\ln(N)N^{-q})$ for some choice of the hyper-parameters $\alpha(N)$ and $\beta_2(N)$. Given that the upper bound in (11) is a sum of non-negative terms, we need each term to be of the order of $\ln(N)N^{-q}$ or negligible. Let us assume that this rate is achieved for $\alpha(N)$ and $\beta_2(N)$. The bound tells us that convergence can only be achieved if $\lim \alpha(N) = 0$ and $\lim \beta_2(N) = 1$, with the limits taken for $N \to \infty$. This motivates us to assume that there exists an asymptotic development of $\alpha(N) \propto N^{-a} + o(N^{-a})$, and of $1 - \beta_2(N) \propto N^{-b} + o(N^{-b})$ for $a$ and $b$ positive. Thus, let us consider only the leading term in those developments, ignoring the leading constant (which is assumed to be non-zero). Let us further assume that $\epsilon \ll R^2$, we have

$$\mathbb{E}\left[\|\nabla F(x_\tau)\|^2\right] \leq 2R\frac{F(x_0) - F_*}{N^{1-a}} + E\left(\frac{1}{N}\ln\left(\frac{R^2N^b}{\epsilon}\right) + \frac{N^{-b}}{1-N^{-b}}\right), \tag{14}$$

with $E = 4dR^2N^{b/2} + dRLN^{b-a}$. Let us ignore the log terms for now, and use $\frac{N^{-b}}{1-N^{-b}} \sim N^{-b}$ for $N \to \infty$ , to get

$$\mathbb{E}\left[\|\nabla F(x_\tau)\|^2\right] \lessapprox 2R\frac{F(x_0) - F_*}{N^{1-a}} + 4dR^2N^{b/2-1} + 4dR^2N^{-b/2} + dRLN^{b-a-1} + dRLN^{-a}.$$

Adding back the logarithmic term, the best rate we can obtain is $O(\ln(N)/\sqrt{N})$, and it is only achieved for $a = 1/2$ and $b = 1$, i.e., $\alpha = \alpha_1/\sqrt{N}$ and $\beta_2 = 1 - 1/N$. We can see the resemblance between Adagrad on one side and Adam with a finite horizon and such parameters on the other. Indeed, an exponential moving average with a parameter $\beta_2 = 1-1/N$ as a typical averaging window length of size $N$, while Adagrad would be an exact average of the past $N$ terms. In particular, the bound for Adam now becomes

$$\mathbb{E}\left[\|\nabla F(x_\tau)\|^2\right] \leq \frac{F(x_0) - F_*}{\alpha_1\sqrt{N}} + \frac{1}{\sqrt{N}}\left(4dR^2 + \alpha_1 dRL\right)\left(\ln\left(1 + \frac{RN}{\epsilon}\right) + \frac{N}{N-1}\right), \tag{15}$$

which differ from (10) only by a $+N/(N-1)$ next to the log term.

**Adam and Adagrad are twins.** Our analysis highlights an important fact: *Adam is to Adagrad like constant step size SGD is to decaying step size SGD.* While Adagrad is asymptotically optimal, it also leads to a slower decrease of the term proportional to $F(x_0) - F_*$, as $1/\sqrt{N}$ instead of $1/N$ for Adam. During the initial phase of training, it is likely that this term dominates the loss, which could explain the popularity of Adam for training deep neural networks rather than Adagrad. With its default parameters, Adam will not converge. It is however possible to choose $\alpha$ and $\beta_2$ to achieve an $\epsilon$ critical point for $\epsilon$ arbitrarily small and, for a known time horizon, they can be chosen to obtain the exact same bound as Adagrad.

## 5 Proofs for $\beta_1 = 0$ (no momentum)

We assume here for simplicity that $\beta_1 = 0$, i.e., there is no heavy-ball style momentum. Taking $n \in \mathbb{N}^*$, the recursions introduced in Section 2.2 can be simplified into

$$\begin{cases} v_{n,i} &= \beta_2 v_{n-1,i} + \left(\nabla_i f_n(x_{n-1})\right)^2, \\ x_{n,i} &= x_{n-1,i} - \alpha_n \frac{\nabla_i f_n(x_{n-1})}{\sqrt{\epsilon + v_{n,i}}}. \end{cases} \tag{16}$$

Remember that we recover Adagrad when $\alpha_n = \alpha$ for $\alpha > 0$ and $\beta_2 = 1$, while Adam can be obtained taking $0 < \beta_2 < 1$, $\alpha > 0$,

$$\alpha_n = \alpha \sqrt{\frac{1-\beta_2^n}{1-\beta_2}}, \tag{17}$$

Throughout the proof we denote by $\mathbb{E}_{n-1}[\cdot]$ the conditional expectation with respect to $f_1, \dots, f_{n-1}$. In particular, $x_{n-1}$ and $v_{n-1}$ are deterministic knowing $f_1, \dots, f_{n-1}$. For all $n \in \mathbb{N}^*$, we also define $\tilde{v}_n \in \mathbb{R}^d$ so that for all $i \in [d]$,

$$\tilde{v}_{n,i} = \beta_2 v_{n-1,i} + \mathbb{E}_{n-1}\left[\left(\nabla_i f_n(x_{n-1})\right)^2\right], \tag{18}$$

i.e., we replace the last gradient contribution by its expected value conditioned on $f_1, \dots, f_{n-1}$.

### 5.1 Technical lemmas

A problem posed by the update (16) is the correlation between the numerator and denominator. This prevents us from easily computing the conditional expectation and as noted by Reddi et al. (2018), the expected direction of update can have a positive dot product with the objective gradient. It is however possible to control the deviation from the descent direction, following Ward et al. (2019) with this first lemma.

**Lemma 5.1** (adaptive update approximately follow a descent direction)**.** *For all $n \in \mathbb{N}^*$ and $i \in [d]$, we have:*

$$\mathbb{E}_{n-1}\left[\nabla_i F(x_{n-1}) \frac{\nabla_i f_n(x_{n-1})}{\sqrt{\epsilon + v_{n,i}}}\right] \geq \frac{\left(\nabla_i F(x_{n-1})\right)^2}{2\sqrt{\epsilon + \tilde{v}_{n,i}}} - 2R\mathbb{E}_{n-1}\left[\frac{\left(\nabla_i f_n(x_{n-1})\right)^2}{\epsilon + v_{n,i}}\right]. \tag{19}$$

*Proof.* We take $i \in [d]$ and note $G = \nabla_i F(x_{n-1})$, $g = \nabla_i f_n(x_{n-1})$, $v = v_{n,i}$ and $\tilde{v} = \tilde{v}_{n,i}$.

$$\mathbb{E}_{n-1}\left[\frac{Gg}{\sqrt{\epsilon + v}}\right] = \mathbb{E}_{n-1}\left[\frac{Gg}{\sqrt{\epsilon + \tilde{v}}}\right] + \mathbb{E}_{n-1}\Bigg[\underbrace{Gg\left(\frac{1}{\sqrt{\epsilon + v}} - \frac{1}{\sqrt{\epsilon + \tilde{v}}}\right)}_{A}\Bigg]. \tag{20}$$

Given that $g$ and $\tilde{v}$ are independent knowing $f_1, \dots, f_{n-1}$, we immediately have

$$\mathbb{E}_{n-1}\left[\frac{Gg}{\sqrt{\epsilon + \tilde{v}}}\right] = \frac{G^2}{\sqrt{\epsilon + \tilde{v}}}. \tag{21}$$

Now we need to control the size of the second term $A$,

$$\begin{aligned} A &= Gg\frac{\tilde{v}-v}{\sqrt{\epsilon+v}\sqrt{\epsilon+\tilde{v}}(\sqrt{\epsilon+v}+\sqrt{\epsilon+\tilde{v}})} \\ &= Gg\frac{\mathbb{E}_{n-1}\left[g^2\right]-g^2}{\sqrt{\epsilon+v}\sqrt{\epsilon+\tilde{v}}(\sqrt{\epsilon+v}+\sqrt{\epsilon+\tilde{v}})} \\ |A| &\leq \underbrace{|Gg|\frac{\mathbb{E}_{n-1}\left[g^2\right]}{\sqrt{\epsilon+v}(\epsilon+\tilde{v})}}_{\kappa}+\underbrace{|Gg|\frac{g^2}{(\epsilon+v)\sqrt{\epsilon+\tilde{v}}}}_{\rho}. \end{aligned}$$

The last inequality comes from the fact that $\sqrt{\epsilon+v}+\sqrt{\epsilon+\tilde{v}} \geq \max(\sqrt{\epsilon+v}, \sqrt{\epsilon+\tilde{v}})$ and $\left|\mathbb{E}_{n-1}\left[g^2\right]-g^2\right| \leq \mathbb{E}_{n-1}\left[g^2\right]+g^2$. Following Ward et al. (2019), we can use the following inequality to bound $\kappa$ and $\rho$,

$$\forall\lambda>0,\ x,y\in\mathbb{R}, xy \leq \frac{\lambda}{2}x^2+\frac{y^2}{2\lambda}. \tag{22}$$

First applying (22) to $\kappa$ with

$$\lambda=\frac{\sqrt{\epsilon+\tilde{v}}}{2},\ x=\frac{|G|}{\sqrt{\epsilon+\tilde{v}}},\ y=\frac{|g|\,\mathbb{E}_{n-1}\left[g^2\right]}{\sqrt{\epsilon+\tilde{v}}\sqrt{\epsilon+v}},$$

we obtain

$$\kappa \leq \frac{G^2}{4\sqrt{\epsilon+\tilde{v}}}+\frac{g^2\mathbb{E}_{n-1}\left[g^2\right]^2}{(\epsilon+\tilde{v})^{3/2}(\epsilon+v)}.$$

Given that $\epsilon+\tilde{v} \geq \mathbb{E}_{n-1}\left[g^2\right]$ and taking the conditional expectation, we can simplify as

$$\mathbb{E}_{n-1}\left[\kappa\right] \leq \frac{G^2}{4\sqrt{\epsilon+\tilde{v}}}+\frac{\mathbb{E}_{n-1}\left[g^2\right]}{\sqrt{\epsilon+\tilde{v}}}\mathbb{E}_{n-1}\left[\frac{g^2}{\epsilon+v}\right]. \tag{23}$$

Given that $\sqrt{\mathbb{E}_{n-1}\left[g^2\right]} \leq \sqrt{\epsilon+\tilde{v}}$ and $\sqrt{\mathbb{E}_{n-1}\left[g^2\right]} \leq R$, we can simplify (23) as

$$\mathbb{E}_{n-1}\left[\kappa\right] \leq \frac{G^2}{4\sqrt{\epsilon+\tilde{v}}}+R\mathbb{E}_{n-1}\left[\frac{g^2}{\epsilon+v}\right]. \tag{24}$$

Now turning to $\rho$, we use (22) with

$$\lambda=\frac{\sqrt{\epsilon+\tilde{v}}}{2\mathbb{E}_{n-1}\left[g^2\right]},\ x=\frac{|Gg|}{\sqrt{\epsilon+\tilde{v}}},\ y=\frac{g^2}{\epsilon+v},$$

we obtain

$$\rho \leq \frac{G^2}{4\sqrt{\epsilon+\tilde{v}}}\frac{g^2}{\mathbb{E}_{n-1}\left[g^2\right]}+\frac{\mathbb{E}_{n-1}\left[g^2\right]}{\sqrt{\epsilon+\tilde{v}}}\frac{g^4}{(\epsilon+v)^2}, \tag{25}$$

Given that $\epsilon+v \geq g^2$ and taking the conditional expectation we obtain

$$\mathbb{E}_{n-1}\left[\rho\right] \leq \frac{G^2}{4\sqrt{\epsilon+\tilde{v}}}+\frac{\mathbb{E}_{n-1}\left[g^2\right]}{\sqrt{\epsilon+\tilde{v}}}\mathbb{E}_{n-1}\left[\frac{g^2}{\epsilon+v}\right], \tag{26}$$

which we simplify using the same argument as for (24) into

$$\mathbb{E}_{n-1}\left[\rho\right] \leq \frac{G^2}{4\sqrt{\epsilon+\tilde{v}}}+R\mathbb{E}_{n-1}\left[\frac{g^2}{\epsilon+v}\right]. \tag{27}$$

Notice that in (25), we possibly divide by zero. It suffice to notice that if $\mathbb{E}_{n-1}\left[g^2\right] = 0$ then $g^2 = 0$ a.s. so that $\rho = 0$ and (27) is still verified. Summing (24) and (27) we can bound

$$\mathbb{E}_{n-1}\left[|A|\right] \leq \frac{G^2}{2\sqrt{\epsilon + \tilde{v}}} + 2R\mathbb{E}_{n-1}\left[\frac{g^2}{\epsilon + v}\right]. \tag{28}$$

Injecting (28) and (21) into (20) finishes the proof. □

Anticipating on Section 5.2, the previous Lemma gives us a bound on the deviation from a descent direction. While for a specific iteration, this deviation can take us away from a descent direction, the next lemma tells us that the sum of those deviations cannot grow larger than a logarithmic term. This key insight introduced in Ward et al. (2019) is what makes the proof work.

**Lemma 5.2** (sum of ratios with the denominator being the sum of past numerators)**.** *We assume we have* $0 < \beta_2 \leq 1$ *and a non-negative sequence* $(a_n)_{n\in\mathbb{N}^*}$*. We define for all* $n \in \mathbb{N}^*$*,* $b_n = \sum_{j=1}^{n} \beta_2^{n-j} a_j$*. We have*

$$\sum_{j=1}^{N} \frac{a_j}{\epsilon + b_j} \leq \ln\left(1 + \frac{b_N}{\epsilon}\right) - N\ln(\beta_2). \tag{29}$$

*Proof.* Given that ln is increasing, and the fact that $b_j > a_j \geq 0$, we have for all $j \in \mathbb{N}^*$,

$$\begin{aligned}\frac{a_j}{\epsilon + b_j} &\leq \ln(\epsilon + b_j) - \ln(\epsilon + b_j - a_j)\\ &= \ln(\epsilon + b_j) - \ln(\epsilon + \beta_2 b_{j-1})\\ &= \ln\left(\frac{\epsilon + b_j}{\epsilon + b_{j-1}}\right) + \ln\left(\frac{\epsilon + b_{j-1}}{\epsilon + \beta_2 b_{j-1}}\right).\end{aligned}$$

The first term forms a telescoping series, while the second one is bounded by $-\ln(\beta_2)$. Summing over all $j \in [N]$ gives the desired result. □

### 5.2 Proof of Adam and Adagrad without momentum

Let us take an iteration $n \in \mathbb{N}^*$, we define the update $u_n \in \mathbb{R}^d$:

$$\forall i \in [d], u_{n,i} = \frac{\nabla_i f_n(x_{n-1})}{\sqrt{\epsilon + v_{n,i}}}. \tag{30}$$

**Adagrad.** As explained in Section 2.2, we have $\alpha_n = \alpha$ for $\alpha > 0$. Using the smoothness of $F$ (8), we have

$$F(x_{n+1}) \leq F(x_n) - \alpha\nabla F(x_n)^T u_n + \frac{\alpha^2 L}{2}\left\|u_n\right\|_2^2. \tag{31}$$

Taking the conditional expectation with respect to $f_0, \ldots, f_{n-1}$ we can apply the descent Lemma 5.1. Notice that due to the a.s. $\ell_\infty$ bound on the gradients (7), we have for any $i \in [d]$, $\sqrt{\epsilon + \tilde{v}_{n,i}} \leq R\sqrt{n}$, so that,

$$\frac{\alpha\left(\nabla_i F(x_{n-1})\right)^2}{2\sqrt{\epsilon + \tilde{v}_{n,i}}} \geq \frac{\alpha\left(\nabla_i F(x_{n-1})\right)^2}{2R\sqrt{n}}. \tag{32}$$

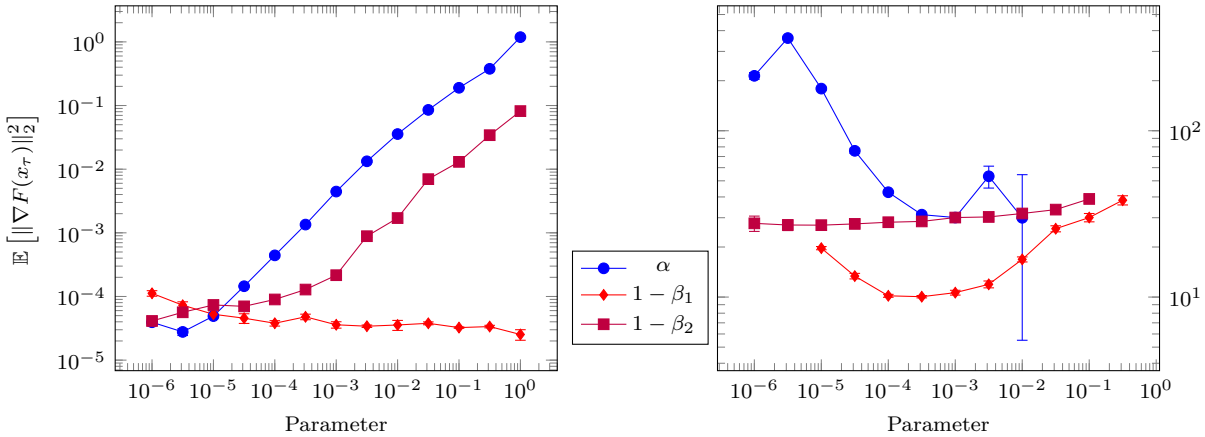
This gives us

$$\mathbb{E}_{n-1}\left[F(x_n)\right] \leq F(x_{n-1}) - \frac{\alpha}{2R\sqrt{n}}\left\|\nabla F(x_{n-1})\right\|_2^2 + \left(2\alpha R + \frac{\alpha^2 L}{2}\right)\mathbb{E}_{n-1}\left[\left\|u_n\right\|_2^2\right].$$

Summing the previous inequality for all $n \in [N]$, taking the complete expectation, and using that $\sqrt{n} \leq \sqrt{N}$ gives us,

$$\mathbb{E}\left[F(x_N)\right] \leq F(x_0) - \frac{\alpha}{2R\sqrt{N}}\sum_{n=0}^{N-1}\mathbb{E}\left[\left\|\nabla F(x_n)\right\|_2^2\right] + \left(2\alpha R + \frac{\alpha^2 L}{2}\right)\sum_{n=0}^{N-1}\mathbb{E}\left[\left\|u_n\right\|_2^2\right].$$

From there, we can bound the last sum on the right hand side using Lemma 5.2 once for each dimension. Rearranging the terms, we obtain the result of Theorem 1.




(a) Average squared norm of the gradient on a toy task, see Section 6, for more details. For the $\alpha$ and $1-\beta_2$ curves, we initialize close to the optimum to make the $F_0 - F_*$ term negligible.

(b) Average squared norm of the gradient of a small convolutional model Gitman & Ginsburg (2017) trained on CIFAR-10, with a random initialization. The full gradient is evaluated every epoch.

Figure 1: Observed average squared norm of the objective gradients after a fixed number of iterations when varying a single parameter out of $\alpha$, $1-\beta_1$ and $1-\beta_2$, on a toy task (left, $10^6$ iterations) and on CIFAR-10 (right, 600 epochs with a batch size 128). All curves are averaged over 3 runs, error bars are negligible except for small values of $\alpha$ on CIFAR-10. See Section 6 for details.

**Adam.** As given by (5) in Section 2.2, we have $\alpha_n = \alpha\sqrt{\frac{1-\beta_2^n}{1-\beta_2}}$ for $\alpha > 0$. Using the smoothness of $F$ defined in (8), we have

$$F(x_n) \leq F(x_{n-1}) - \alpha_n \nabla F(x_{n-1})^T u_n + \frac{\alpha_n^2 L}{2} \|u_n\|_2^2 . \tag{33}$$

We have for any $i \in [d]$, $\sqrt{\epsilon + \tilde{v}_{n,i}} \leq R\sqrt{\sum_{j=0}^{n-1} \beta_2^j} = R\sqrt{\frac{1-\beta_2^n}{1-\beta_2}}$, thanks to the a.s. $\ell_\infty$ bound on the gradients (7), so that,

$$\alpha_n \frac{(\nabla_i F(x_{n-1}))^2}{2\sqrt{\epsilon + \tilde{v}_{n,i}}} \geq \frac{\alpha\,(\nabla_i F(x_{n-1}))^2}{2R}. \tag{34}$$

Taking the conditional expectation with respect to $f_1, \ldots, f_{n-1}$ we can apply the descent Lemma 5.1 and use (34) to obtain from (33),

$$\mathbb{E}_{n-1}\left[F(x_n)\right] \leq F(x_{n-1}) - \frac{\alpha}{2R}\|\nabla F(x_{n-1})\|_2^2 + \left(2\alpha_n R + \frac{\alpha_n^2 L}{2}\right)\mathbb{E}_{n-1}\left[\|u_n\|_2^2\right].$$

Given that $\beta_2 < 1$, we have $\alpha_n \leq \frac{\alpha}{\sqrt{1-\beta_2}}$. Summing the previous inequality for all $n \in [N]$ and taking the complete expectation yields

$$\mathbb{E}\left[F(x_N)\right] \leq F(x_0) - \frac{\alpha}{2R}\sum_{n=0}^{N-1}\mathbb{E}\left[\|\nabla F(x_n)\|_2^2\right] + \left(\frac{2\alpha R}{\sqrt{1-\beta_2}} + \frac{\alpha^2 L}{2(1-\beta_2)}\right)\sum_{n=0}^{N-1}\mathbb{E}\left[\|u_n\|_2^2\right].$$

Applying Lemma 5.2 for each dimension and rearranging the terms finishes the proof of Theorem 2.

## 6 Experiments

On Figure 1, we compare the effective dependency of the average squared norm of the gradient in the parameters $\alpha$, $\beta_1$ and $\beta_2$ for Adam, when used on a toy task and CIFAR-10.

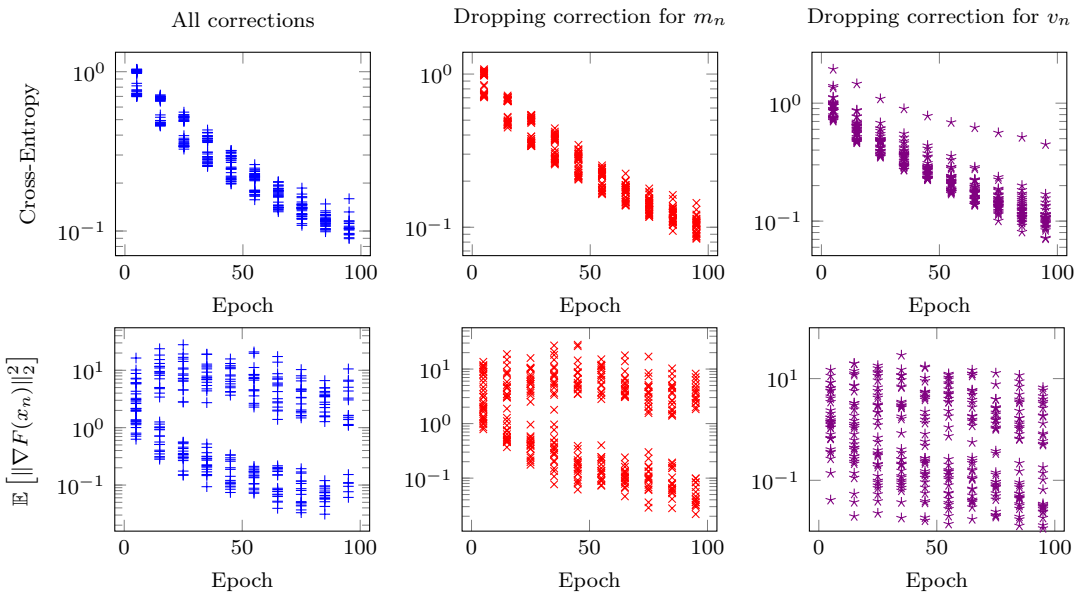



Figure 2: Training trajectories for varying values of $\alpha \in \{10^{-4}, 10^{-3}\}$, $\beta_1 \in \{0., 0.5, 0.8, 0.9, 0.99\}$ and $\beta_2 \in \{0.9, 0.99, 0.999, 0.9999\}$. The top row (resp. bottom) gives the training loss (resp. squared norm of the expected gradient). The left column uses all corrective terms in the original Adam algorithm, the middle column drops the corrective term on $m_n$ (equivalent to our proof setup), and the right column drops the corrective term on $v_n$. We notice a limited impact when dropping the corrective term on $m_n$, but dropping the corrective term on $v_n$ has a much stronger impact.

### 6.1 Setup

**Toy problem.** In order to support the bounds presented in Section 4, in particular the dependency in $\beta_2$, we test Adam on a specifically crafted toy problem. We take $x \in \mathbb{R}^6$ and define for all $i \in [6]$, $p_i = 10^{-i}$. We take $(Q_i)_{i\in[6]}$, Bernoulli variables with $\mathbb{P}[Q_i = 1] = p_i$. We then define $f$ for all $x \in \mathbb{R}^d$ as

$$f(x) = \sum_{i\in[6]} (1 - Q_i)\,\mathrm{Huber}(x_i - 1) + \frac{Q_i}{\sqrt{p_i}}\,\mathrm{Huber}(x_i + 1), \tag{35}$$

with for all $y \in \mathbb{R}$,

$$\mathrm{Huber}(y) = \begin{cases} \frac{y^2}{2} & \text{when } |y| \leq 1 \\ |y| - \frac{1}{2} & \text{otherwise.} \end{cases}$$

Intuitively, each coordinate is pointing most of the time towards 1, but exceptionally towards -1 with a weight of $1/\sqrt{p_i}$. Those rare events happens less and less often as $i$ increase, but with an increasing weight. Those weights are chosen so that all the coordinates of the gradient have the same variance[1]. It is necessary to take different probabilities for each coordinate. If we use the same $p$ for all, we observe a phase transition when $1 - \beta_2 \approx p$, but not the continuous improvement we obtain on Figure 1a.

We plot the variation of $\mathbb{E}\left[\|F(x_\tau)\|_2^2\right]$ after $10^6$ iterations with batch size 1 when varying either $\alpha$, $1 - \beta_1$ or $1 - \beta_2$ through a range of 13 values uniformly spaced in log-scale between $10^{-6}$ and 1. When varying $\alpha$, we take $\beta_1 = 0$ and $\beta_2 = 1 - 10^{-6}$. When varying $\beta_1$, we take $\alpha = 10^{-5}$ and $\beta_2 = 1 - 10^{-6}$ (i.e. $\beta_2$ is so that we are in the Adagrad-like regime). Finally, when varying $\beta_2$, we take $\beta_1 = 0$ and $\alpha = 10^{-6}$. We start from $x_0$ close to the optimum by running first $10^6$ iterations with $\alpha = 10^{-4}$, then $10^6$ iterations with

[1]We deviate from the a.s. bounded gradient assumption for this experiment, see Section 4.2 for a discussion on a.s. bound vs bound in expectation.

$\alpha = 10^{-5}$, always with $\beta_2 = 1 - 10^{-6}$. This allows to have $F(x_0) - F_* \approx 0$ in (11) and (13) and focus on the second part of both bounds. All curves are averaged over three runs. Error bars are plotted but not visible in log-log scale.

**CIFAR-10.** We train a simple convolutional network (Gitman & Ginsburg, 2017) on the CIFAR-10[2] image classification dataset. Starting from a random initialization, we train the model on a single V100 for 600 epochs with a batch size of 128, evaluating the full training gradient after each epoch. This is a proxy for $\mathbb{E}\left[\|F(x_\tau)\|_2^2\right]$, which would be to costly to evaluate exactly. All runs use the default config $\alpha = 10^{-3}$, $\beta_2 = 0.999$ and $\beta_1 = 0.9$, and we then change one of the parameter.

We take $\alpha$ from a uniform range in log-space between $10^{-6}$ and $10^{-2}$ with 9 values, for $1 - \beta_1$ the range is from $10^{-5}$ to 0.3 with 9 values, and for $1 - \beta_2$, from $10^{-6}$ to $10^{-1}$ with 11 values. Unlike for the toy problem, we do not initialize close to the optimum, as even after 600 epochs, the norm of the gradients indicates that we are not at a critical point. All curves are averaged over three runs. Error bars are plotted but not visible in log-log scale, except for large values of $\alpha$.

### 6.2 Analysis

**Toy problem.** Looking at Figure 1a, we observe a continual improvement as $\beta_2$ increases. Fitting a linear regression in log-log scale of $\mathbb{E}[\|\nabla F(x_\tau)\|_2^2]$ with respect to $1 - \beta_2$ gives a slope of 0.56 which is compatible with our bound (11), in particular the dependency in $O(1/\sqrt{1-\beta_2})$. As we initialize close to the optimum, a small step size $\alpha$ yields as expected the best performance. Doing the same regression in log-log scale, we find a slope of 0.87, which is again compatible with the $O(\alpha)$ dependency of the second term in (11). Finally, we observe a limited impact of $\beta_1$, except when $1 - \beta_1$ is small. The regression in log-log scale gives a slope of -0.16, while our bound predicts a slope of -1.

**CIFAR 10.** Let us now turn to Figure 1b. As we start from random weights for this problem, we observe that a large step size gives the best performance, although we observe a high variance for the largest $\alpha$. This indicates that training becomes unstable for large $\alpha$, which is not predicted by the theory. This is likely a consequence of the bounded gradient assumption (7) not being verified for deep neural networks. We observe a small improvement as $1 - \beta_2$ decreases, although nowhere near what we observed on our toy problem. Finally, we observe a sweet spot for the momentum $\beta_1$, not predicted by our theory. We conjecture that this is due to the variance reduction effect of momentum (averaging of the gradients over multiple mini-batches, while the weights have not moved so much as to invalidate past information).

### 6.3 Impact of the Adam corrective terms

Using the same experimental setup on CIFAR-10, we compare the impact of removing either of the corrective term of the original Adam algorithm (Kingma & Ba, 2015), as discussed in Section 2.2. We ran a cartesian product of training for 100 epochs, with $\beta_1 \in \{0, 0.5, 0.8, 0.9, 0.99\}$, $\beta_2 \in \{0.9, 0.99, 0.999m0.9999\}$, and $\alpha \in \{10^{-4}, 10^{-3}\}$. We report both the training loss and norm of the expected gradient on Figure 2. We notice a limited difference when dropping the corrective term on $m_n$, but dropping the term $v_n$ has an important impact on the training trajectories. This confirm our motivation for simplifying the proof by removing the corrective term on the momentum.

## 7 Conclusion

We provide a simple proof on the convergence of Adam and Adagrad without heavy-ball style momentum. Our analysis highlights a link between the two algorithms: with right the hyper-parameters, Adam converges like Adagrad. The extension to heavy-ball momentum is more complex, but we significantly improve the dependence on the momentum parameter for Adam, Adagrad, as well as SGD. We exhibit a toy problem where the dependency on $\alpha$ and $\beta_2$ experimentally matches our prediction. However, we do not predict the practical interest of momentum, so that improvements to the proof are needed for future work.

[2] `https://www.cs.toronto.edu/~kriz/cifar.html`

### Broader Impact Statement

The present theoretical results on the optimization of non convex losses in a stochastic settings impact our understanding of the training of deep neural network. It might allow a deeper understanding of neural network training dynamics and thus reinforce any existing deep learning applications. There would be however no direct possible negative impact to society.

## References

Alekh Agarwal, Martin J Wainwright, Peter L Bartlett, and Pradeep K Ravikumar. Information-theoretic lower bounds on the oracle complexity of convex optimization. In *Advances in Neural Information Processing Systems*, 2009.

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

Aaron Defazio. Momentum via primal averaging: Theoretical insights and learning rate schedules for non-convex optimization. *arXiv preprint arXiv:2010.00406*, 2020.

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

John Duchi, Michael I Jordan, and Brendan McMahan. Estimation, optimization, and parallelism when data is sparse. In *Advances in Neural Information Processing Systems 26*, 2013.

Biyi Fang and Diego Klabjan. Convergence analyses of online adam algorithm in convex setting and two-layer relu neural network. *arXiv preprint arXiv:1905.09356*, 2019.

Matthew Faw, Isidoros Tziotis, Constantine Caramanis, Aryan Mokhtari, Sanjay Shakkottai, and Rachel Ward. The power of adaptivity in sgd: Self-tuning step sizes with unbounded gradients and affine variance. In Po-Ling Loh and Maxim Raginsky (eds.), *Proceedings of Thirty Fifth Conference on Learning Theory*, Proceedings of Machine Learning Research. PMLR, 2022.

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

Igor Gitman and Boris Ginsburg. Comparison of batch normalization and weight normalization algorithms for the large-scale image classification. *arXiv preprint arXiv:1709.08145*, 2017.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2015.

Timothée Lacroix, Nicolas Usunier, and Guillaume Obozinski. Canonical tensor decomposition for knowledge base completion. *arXiv preprint arXiv:1806.07297*, 2018.

Xiaoyu Li and Francesco Orabona. On the convergence of stochastic gradient descent with adaptive stepsizes. In *AI Stats*, 2019.

Yanli Liu, Yuan Gao, and Wotao Yin. An improved analysis of stochastic gradient descent with momentum. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 18261–18271. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/file/d3f5d4de09ea19461dab00590df91e4f-Paper.pdf`.

H Brendan McMahan and Matthew Streeter. Adaptive bound optimization for online convex optimization. In *COLT*, 2010.

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

Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of Adam and beyond. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2018.

Naichen Shi, Dawei Li, Mingyi Hong, and Ruoyu Sun. Rmsprop converges with proper hyper-parameter. In *International Conference on Learning Representations*, 2020.

T. Tieleman and G. Hinton. Lecture 6.5 — rmsprop. COURSERA: Neural Networks for Machine Learning, 2012.

Rachel Ward, Xiaoxia Wu, and Leon Bottou. Adagrad stepsizes: Sharp convergence over nonconvex landscapes. In *International Conference on Machine Learning*, 2019.

Ashia C Wilson, Rebecca Roelofs, Mitchell Stern, Nati Srebro, and Benjamin Recht. The marginal value of adaptive gradient methods in machine learning. In *Advances in Neural Information Processing Systems*, 2017.

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

Dongruo Zhou, Yiqi Tang, Ziyan Yang, Yuan Cao, and Quanquan Gu. On the convergence of adaptive gradient methods for nonconvex optimization. *arXiv preprint arXiv:1808.05671*, 2018.

Fangyu Zou, Li Shen, Zequn Jie, Weizhong Zhang, and Wei Liu. A sufficient condition for convergences of Adam and RMSprop. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019a.

Fengyu Zou, Li Shen, Zenqun Jie, Ju Sun, and Wei Liu. Weighted Adagrad with unified momentum. *arXiv preprint arXiv:1808.03408*, 2019b.