Linear Convergence of Adaptive Stochastic Gradient Descent

Yuege Xie* **Xiaoxia Wu†** **Rachel Ward*†**
*Oden Institute, UT Austin †Department of Mathematics, UT Austin

Abstract

We prove that the norm version of the adaptive stochastic gradient method (AdaGrad-Norm) achieves a linear convergence rate for a subset of either strongly convex functions or non-convex functions that satisfy the Polyak-Łojasiewicz (PL) inequality. The paper introduces the notion of Restricted Uniform Inequality of Gradients (RUIG)—which is a measure of the balanced-ness of the stochastic gradient norms—to depict the landscape of a function. RUIG plays a key role in proving the robustness of AdaGrad-Norm to its hyper-parameter tuning in the stochastic setting. On top of RUIG, we develop a two-stage framework to prove the linear convergence of AdaGrad-Norm without knowing the parameters of the objective functions. This framework can likely be extended to other adaptive stepsize algorithms. The numerical experiments validate the theory and suggest future directions for improvement.

1 Introduction

Consider the optimization problem of minimizing the empirical risk:

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} F(\boldsymbol{x}) := \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{x})$$

where $f_i(\boldsymbol{x}) = f(\boldsymbol{x}, \boldsymbol{Z}_i) : \mathbb{R}^d \to \mathbb{R}, i = 1, 2, \dots$ and $\{\boldsymbol{Z}_1, \dots, \boldsymbol{Z}_n\}$ are empirical samples drawn uniformly from an unknown underlying distribution $\mathcal{S}$. In this paper, we focus on smooth functions $F(\boldsymbol{x})$ that are either strongly convex, or non-convex with Polyak-Łojasiewicz inequality (Lojasiewicz, 1963; Polyak, 1963), which are fundamental to a variety of machine learning problems (Bottou and Cun, 2004; Bottou et al., 2018).

Linear convergence results using stochastic gradient descent (SGD) or accelerated SGD (Bottou, 1991; Nash and Nocedal, 1991; Bertsekas, 1999; Nesterov, 2005; Haykin et al., 2005; Bubeck et al., 2015) to solve the above problem have been established for this class of functions: SGD with fixed stepsize guarantees linear convergence to global minima (Allen-Zhu et al., 2018; Zou et al., 2018a) or up to a radius around the optimal solution (Moulines and Bach, 2011; Needell et al., 2016); Improved algorithms—like SAG (Schmidt et al., 2017), SVRG (Johnson and Zhang, 2013) and SAGA (Defazio et al., 2014)—allow faster linear convergence to the global minimizer. However, since the above convergence requires that fixed stepsizes must meet a certain threshold determined by unknown parameters such as the level of stochastic noise, Lipschitz smoothness constants, and strong convexity parameters, SGD and variance reduced SGD are highly sensitive to stepsize tuning in practice. Thus, seeking an algorithm that is robust to the choice of hyper-parameters is as crucial as designing an algorithm that gives faster convergence. The paper focuses on the robustness of the linear convergence of adaptive stochastic gradient descent to unknown hyperparameters.

Adaptive gradient descent methods introduced in Duchi et al. (2011) and McMahan and Streeter (2010) update the stepsize on the fly: They either adapt a vector of per-coefficient stepsizes (Kingma and Ba, 2014; Lafond et al., 2017; Reddi et al., 2018a; Shah et al., 2018; Zou et al., 2018b; Staib et al., 2019) or a single stepsize depending on the norm of the gradient (Levy, 2017; Ward et al., 2018; Wu et al., 2018). The latter one, AdaGrad-Norm (Ward et al., 2018) has the following updates:

$$b_{j+1}^2 = b_j^2 + \|\nabla f_{\xi_j}(\boldsymbol{x}_j)\|^2;$$
$$\boldsymbol{x}_{j+1} = \boldsymbol{x}_j - \frac{\eta}{b_{j+1}}\nabla f_{\xi_j}(\boldsymbol{x}_j)$$

where $\xi_j \sim \text{Unif}\{1, 2, \dots\}$ such that $\mathbb{E}_{\xi_j}\left[\nabla f_{\xi_j}(\boldsymbol{x})\right] = \nabla F(\boldsymbol{x}), \forall \boldsymbol{x}$. AdaGrad-Norm has been shown to be ex-

Proceedings of the 23rdInternational Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

tremely resilient to the functions' parameters being unknown (Levy, 2017; Levy et al., 2018; Ward et al., 2018). In addition to this robustness, AdaGrad-Norm enjoys $\mathcal{O}(1/\sqrt{T})$ convergence rate for smooth non-convex functions under the metric $\min_{j\in[T]} \|\nabla F(\boldsymbol{x}_j)\|^2$ (Ward et al., 2018; Li and Orabona, 2018). This asymptotic convergence rate has also been proved for general convex functions (Levy et al., 2018). A linear convergence rate $\mathcal{O}\left(\exp(-\kappa T)\right)$ [1] is possible for strongly convex smooth functions using variants of AdaGrad-Norm in which the final update uses a harmonic sum of the queried gradients (Levy, 2017). Yet, the analysis in Levy (2017) and Levy et al. (2018) requires a priori information: a convex set with a known diameter in which the global minimizer resides. The analysis in Ward et al. (2018) considers the smooth function under an assumption of a bounded stochastic gradient norm that rules out the strongly convex cases, while Li and Orabona (2018) only assumes bounded variance but requires prior knowledge of smoothness. Therefore, obtaining a robust linear convergence guarantee without prior knowledge of a convex set or the smoothness parameters, remains an open question for AdaGrad-Norm with strongly convex objectives.

In this paper, we establish robust linear convergence guarantees for AdaGrad-Norm for strongly convex functions without requiring knowledge of smoothness or strong convexity parameters, nor the knowledge of a convex set containing the minimizer, and we also extend our analysis to non-convex functions that satisfy the Polyak-Łojasiewicz (PL) inequality.[2] Our analysis does not follow the standard analysis—which assumes the bounded variance $\hat{\sigma}$ for $\mathbb{E}_{\xi_j}[\|\nabla f_{\xi_j}(\boldsymbol{x}) - \nabla F(\boldsymbol{x})\|^2] \leq \hat{\sigma}^2, \forall j, \forall \boldsymbol{x}$ in Levy et al. (2018); Levy (2017); Ward et al. (2018); Li and Orabona (2018)—and avoids likely sub-linear convergence results. The set of functions for which we guarantee a robust linear convergence rate using AdaGrad-Norm includes certain classes of neural networks. Among these many applications, one function class of particular interest is the over-parameterized neural network (Vaswani et al., 2018; Zhang et al., 2016; Du et al., 2019; Zhou et al., 2019; Bassily et al., 2018). Our contributions are not only significant for the algorithm in its own right, but because of the generality of our two-stage framework for the linear convergence proof, we believe it is easily applicable to other adaptive algorithms such as Adam (Kingma and Ba, 2014) and AMSGrad (Reddi et al., 2018a).

[1] κ is the condition number

[2] Note that our results are for the norm version of AdaGrad (AdaGrad-Norm), which differs from the convergence of the diagonal version of AdaGrad and its variants (with momentum) (Balles and Hennig, 2018; Bernstein et al., 2018; Mukkamala and Hein, 2017; Chen et al., 2018).

Notations $\|.\|$ denotes the ℓ_2-norm. μ is either the $\mu-$strongly convex parameter in Assumption (A1a) or the $\mu-$PL Inequality parameter in Assumption (A1b). In the batch setting, L is the smallest Lipschitz constant of $\nabla F(\boldsymbol{x})$; in the stochastic setting, $L \triangleq \sup_i L_i$, where L_i is the Lipschitz constant of $\nabla f_i(\boldsymbol{x})$. $\mathbb{P}_i(\cdot)$ is the probability w.r.t. the i-th sample point.

1.1 Main Contributions

We propose Restricted Uniform Inequality of Gradients (RUIG) to measure the uniform lower bound of stochastic gradients according to $\|\boldsymbol{x} - \boldsymbol{x}^*\|$ in a restricted region. On top of RUIG, we show that the evolution of the error can be divided into the following two stages:

- **Stage I** If $b_t < \eta\mu \leq \eta L$, $\|\boldsymbol{x}_t - \boldsymbol{x}^*\|^2$ increases first (but remains smaller than a certain upper bound), and contracts after $b_t \geq \eta\mu$, while b_t continues growing until it exceeds ηL;
- **Stage II** $b_t > \eta L$, AdaGrad-Norm converges linearly. b_t increases during the optimization process but it is always bounded by $b_{\max}$.

We illustrate these stages in Figure 1 with $\eta = 1$.

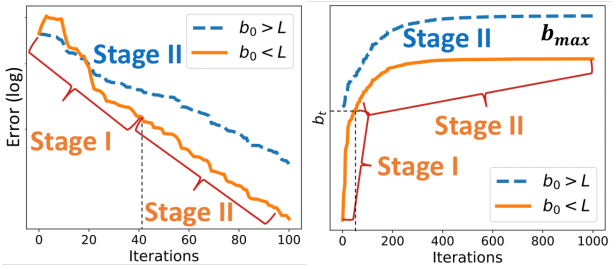

Figure 1: Two-Stage Convergence of AdaGrad-Norm with different initial stepsizes: $b_0 < L$ versus $b_0 > L$. Left: Error $\|\boldsymbol{x}_t - \boldsymbol{x}^*\|^2$ in logarithmic scale. Right: Growth of b_t to corresponding upper bounds ($\leq b_{\max}$).

We prove the non-asymptotic linear convergence of AdaGrad-Norm in the strongly convex setting for stochastic and batch updates; furthermore, we also extend our results for non-convex functions satisfying PL inequality. Our main results are as follows (informal):

1. In the stochastic setting, Theorem 1 shows that AdaGrad-Norm attains $\min_i \|\boldsymbol{x}_i - \boldsymbol{x}^*\|^2 \leq \epsilon$ with high probability after $T = \mathcal{O}(\log \frac{1}{\epsilon})$ iterations for $b_0 > \eta L$; and after $T = \mathcal{O}(\frac{1}{\epsilon} + \log \frac{1}{\epsilon})$ iterations for $b_0 \leq \eta L$, assuming that $F(\boldsymbol{x})$ is $\mu-$strong convex, $L-$smooth, almost stationary and with $(\epsilon, \alpha, \gamma)$-RUIG ($\forall \epsilon > 0$, for any fixed $\boldsymbol{x} \in \mathbb{R}^d$, if $\|\boldsymbol{x} -$

$\boldsymbol{x}^*\|^2 > \epsilon$, then $\exists(\alpha,\gamma)$ s.t. $\mathbb{P}_{\xi_j}(\|\nabla f_{\xi_j}(\boldsymbol{x})\|^2 \geq \alpha\|\boldsymbol{x}-\boldsymbol{x}^*\|^2) \geq \gamma, \xi_j = 1,2,\dots)$.

2. In the batch setting, by using the full gradient, the above probability γ degrades to 1 and $\alpha = \mu^2$. Theorem 2 shows that $\min_i \|\boldsymbol{x}_i - \boldsymbol{x}^*\|^2 \leq \epsilon$ after $T = \mathcal{O}(\log\frac{1}{\epsilon})$ iterations for $b_0 > \eta\frac{\mu+L}{2}$ and after $T = \mathcal{O}(\frac{1}{\log(1+C\epsilon)} + \log\frac{1}{\epsilon})$ iterations for $b_0 \leq \eta\frac{\mu+L}{2}$, if $F(x)$ is strongly convex and smooth.

3. For non-convex functions with the PL inequality, we alternatively consider the convergence rate of $\min_i F(\boldsymbol{x}_i) - F^*$. Theorem 3 illustrates that $\min_i F(\boldsymbol{x}_i) - F^* \leq \epsilon$ after $T = \mathcal{O}(\log\frac{1}{\epsilon})$ iterations for $b_0 > \eta L$; and after $T = \mathcal{O}(\frac{1}{\log(1+C\epsilon)} + \log\frac{1}{\epsilon})$ iterations for $b_0 \leq \eta L$.

We show that the convergence is robust starting from any initialization of b_0, without knowing the Lipschitz constant or strong convexity parameter a priori. The robustness is shown in Table 1: when starting from different initial stepsizes, the convergence rates of AdaGrad-Norm are only changed according to the slope in Stage II and negligible gain from the added-on sublinear part in Stage I. However, changing the initial stepsize for SGD causes divergence.

2 Problem Setup

Consider the empirical risk $F(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^n f_i(\boldsymbol{x})$, where $f_i(\boldsymbol{x}) = f(\boldsymbol{x}, \boldsymbol{Z}_i) : \mathbb{R}^d \to \mathbb{R}, i = 1,2,\dots,n$ with possibly infinite n. In contrast to (stochastic) Gradient Descent implemented with *fixed* stepsize, the update rules of AdaGrad-Norm (see Algorithm 1) *dynamically* incorporates the information from previous gradients into the reciprocal of the learning rates.

Algorithm 1 AdaGrad-Norm

Input: Initialize $\epsilon > 0, \eta > 0, T > 0, \boldsymbol{x}_0 \in \mathbb{R}^d, b_0 \in \mathbb{R}, j \leftarrow 0$
while $j < T$ **do**
  Generate random variable ξ_j and compute G_j (stochastic: $G_j = \nabla f_{\xi_j}(\boldsymbol{x}_j)$; batch: $G_j = \nabla F(\boldsymbol{x}_j)$)
  $b_{j+1}^2 \leftarrow b_j^2 + \|G_j\|^2$
  $\boldsymbol{x}_{j+1} \leftarrow \boldsymbol{x}_j - \frac{\eta}{b_{j+1}} G_j$
  $j \leftarrow j+1$

The algorithm follows the standard assumptions from Bottou et al. (2018): for each $j \geq 0$, the random vectors ξ_j, $j = 0,1,2,\dots$, are mutually independent, independent of $\boldsymbol{x}_j$, and satisfy $\mathbb{E}_{\xi_j}[\nabla f_{\xi_j}(\boldsymbol{x}_j)] = \nabla F(\boldsymbol{x}_j)$. In the stochastic setting, it draws one sample at a time and uses unbiased estimators ($G_j = \nabla f_{\xi_j}(\boldsymbol{x}_j)$) of the full gradients of $F(\boldsymbol{x}_j)$ to update. In the batch setting, it uses full gradients ($G_j = \nabla F(\boldsymbol{x}_j)$) instead.

In the convergence analysis, we consider the following two equivalent updates of AdaGrad-Norm:

$$\textbf{Square Form: } b_{j+1} = \sqrt{b_j^2 + \|\nabla f_{\xi_j}(\boldsymbol{x}_j)\|^2}$$

$$\textbf{Solution Form: } b_{j+1} = b_j + \frac{\|\nabla f_{\xi_j}(\boldsymbol{x}_j)\|^2}{b_j + b_{j+1}}$$

Assumptions Throughout the paper, we use different combinations of the following assumptions to analyze the convergence rates in both the stochastic (with Assumptions (A1a), (A2), (A3) and (A4)) and batch (with Assumptions (A1a)/(A1b) and (A2)) settings.

(A1a) $\mu-$**strongly convex:** $F(\boldsymbol{x})$ is differentiable and $\langle\nabla F(\boldsymbol{x}) - \nabla F(\boldsymbol{y}), \boldsymbol{x}-\boldsymbol{y}\rangle \geq \mu\|\boldsymbol{x}-\boldsymbol{y}\|^2, \forall \boldsymbol{x}, \boldsymbol{y}$.

(A1b) $\mu-$**Polyak-Łojasiewicz (PL) Inequality:** $\|\nabla F(\boldsymbol{x})\|^2 \geq 2\mu(F(\boldsymbol{x}) - F(\boldsymbol{x}^*)), \forall \boldsymbol{x}$.

(A2) $L-$**smooth:** $f_i(\boldsymbol{x})$ is L_i-smooth, $\forall i$: $\|\nabla f_i(\boldsymbol{x}) - \nabla f_i(\boldsymbol{y})\| \leq L_i\|\boldsymbol{x}-\boldsymbol{y}\|, \forall \boldsymbol{x}, \boldsymbol{y}$. Let $L \triangleq \sup_i L_i$, $F(\boldsymbol{x})$ and $\{f_i(\boldsymbol{x})\}$ are all $L-$smooth.

(A3) $(\epsilon,\alpha,\gamma)-$**Restricted Uniform Inequality of Gradients (RUIG):** $\forall\epsilon > 0$, for any fixed $\boldsymbol{x} \in \mathcal{D}_\epsilon \triangleq \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}-\boldsymbol{x}^*\|^2 > \epsilon\}$, $\exists(\alpha,\gamma)$ s.t. $\alpha > 0$, $\gamma > 0$, and $\mathbb{P}_i(\|\nabla f_i(\boldsymbol{x})\|^2 \geq \alpha\|\boldsymbol{x}-\boldsymbol{x}^*\|^2) \geq \gamma$, $\forall i = 1,2,\dots$.

(A4) **convex and almost stationary:** (Moulines and Bach, 2011; Needell et al., 2016; Vaswani et al., 2018) $f_i(\boldsymbol{x})$ is convex, $\forall i$. Let $\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} F(\boldsymbol{x})$, then $\mathbb{P}_i(\nabla f_i(\boldsymbol{x}^*) = \boldsymbol{0}) = 1, \forall i$, i.e. the probability of $\boldsymbol{x}^*$ being a stationary point is almost surely over all sample data points.

Assumption (A3) is a sufficient condition to guarantee the linear convergence for AdaGrad-Norm with any initialization of stepsize, but it is not necessary when the initial stepsize is smaller than the unknown critical values, i.e. $\frac{1}{\eta L}$ or $\frac{2}{\eta(\mu+L)}$. Examples of systems with this property are in Section 3.

Assumption (A4) is the key condition for linear convergence of $\|\boldsymbol{x}-\boldsymbol{x}^*\|^2$ in the stochastic approximation algorithms (Roux et al., 2012; Wu et al., 2018) as it imposes a strong condition on each component function at the point x^*. However, this assumption is much weaker than (Strong or Weak[3]) Growth Condition in Schmidt and Roux (2013); Vaswani et al. (2018) where it is assumed that $\forall \boldsymbol{x} \in \mathbb{R}^d$, $\max_i \|\nabla f_i(\boldsymbol{x})\|^2 \leq B\|\nabla F(\boldsymbol{x})\|^2$ or $\mathbb{E}_i\|\nabla f_i(\boldsymbol{x})\|^2 < B(F(\boldsymbol{x}) - F^*)$, for some constant B. We use the weaker assumption and

[3]The weak growth condition in Cevher and Vũ (2019) is weaker than that in Vaswani et al. (2018).

Table 1: Summary of convergence rates of (stochastic) GD in strongly convex setting

Setting	Algorithm	Initial stepsize	Iterations to achieve $\|\boldsymbol{x}_{best} - \boldsymbol{x}^*\|^2 \leq \epsilon$ [1]
Stochastic GD	fixed stepsize	$\eta_0 = \frac{1}{2\sup_i L_i}$	$\mathcal{O}(\frac{\sup_i L_i}{\mu} \log \frac{\Delta_0}{\epsilon})$ (Needell et al., 2016)
	AdaGrad-Norm[2]	$\eta_0 = \frac{1}{b_0} < \frac{1}{\sup_i L_i}$	$\mathcal{O}(\frac{\sup_i L \Delta_0}{\mu} \log \frac{\Delta_0}{\epsilon})$
	AdaGrad-Norm	arbitrary	$\mathcal{O}(\frac{1}{\epsilon} + \frac{\sup_i L_i \Delta_0}{\mu} \log \frac{\Delta_0}{\epsilon})$
Deterministic GD	fixed stepsize	$\eta_0 = \frac{2}{\mu+L}$	$\mathcal{O}(\frac{(\mu+L)^2}{4\mu L} \log \frac{\Delta_0}{\epsilon})$ (Bubeck et al., 2015)
	AdaGrad-Norm	$\eta_0 = \frac{1}{b_0} < \frac{2}{\mu+L}$	$\mathcal{O}(\frac{L\Delta_0}{\mu} \log \frac{\Delta_0}{\epsilon})$
	AdaGrad-Norm	arbitrary	$\mathcal{O}(\frac{1}{\log(1+C\epsilon)} + \frac{L\Delta_0}{\mu} \log \frac{\Delta_0}{\epsilon})$

[1] $\Delta_0 = \|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2$ is the initial distance to the minimizer $\boldsymbol{x}^*$.
[2] AdaGrad-Norm with $\eta = 1$.

characterize a better convergence rate for AdaGrad-Norm over many optimization problems that satisfy Assumption (A4). One particularly relevant application is the over-parameterized neural network. Note that Assumption (A4) implies that there exists almost no noise at the solution, which may not be appropriate for certain applications.

3 Restricted Uniform Inequality of Gradients

In this section, we concretely explain our Assumption (A3) in Section 2 and restate it as follows:

Assumption. $(\epsilon, \alpha, \gamma)-$***Restricted Uniform Inequality of Gradients (RUIG):*** *$\forall \epsilon > 0$, for any fixed $\boldsymbol{x} \in \mathcal{D}_\epsilon \triangleq \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x} - \boldsymbol{x}^*\|^2 > \epsilon\}$, $\exists(\alpha, \gamma)$ s.t. $\alpha > 0$, $\gamma > 0$, and*

$$\mathbb{P}_i(\|\nabla f_i(\boldsymbol{x})\|^2 \geq \alpha \|\boldsymbol{x} - \boldsymbol{x}^*\|^2) \geq \gamma, \forall i = 1, 2, \ldots$$

The RUIG gives a lower bound on the probability γ, with which the norm of any unbiased gradient estimator $\|\nabla f_i(\boldsymbol{x})\|$ is larger than the distance between $\boldsymbol{x}$ and $\boldsymbol{x}^*$ by a constant factor α, if $\boldsymbol{x}$ is in a restricted region $\mathcal{D}_\epsilon$. This inequality depicts a set of functions $\{F(\boldsymbol{x})\}$ that preserve a flat landscape around $\boldsymbol{x}^*$ for each component loss function $f_i(\boldsymbol{x})$ and characterize the relatively sharper curvature beyond the region.

The constant tuple $(\epsilon, \alpha, \gamma)$ is determined by the distribution of the dataset. In general, α and γ are negatively correlated, i.e., $\alpha \to 0$, $\gamma \to 1$. The error ϵ could be independent of α and γ. However, with large ϵ, the product $\alpha\gamma$ is more likely far away from zero. In addition, if $\epsilon_2 \geq \epsilon_1 \geq 0$, then $\mathcal{D}_{\epsilon_2} \subseteq \mathcal{D}_{\epsilon_1} \subseteq \mathcal{D}_0 = \mathbb{R}^d$.

We provide some examples where we can directly compute the lower bounds on α and γ for a restricted region $\mathcal{D}_\epsilon$. (See Appendix E.2 for an empirical example.)

Note that these bounds depend on the dataset $\{\boldsymbol{a}_i\}_{i=1}^\infty$, hence they are data-dependent.

Example 1. Least Square Problem Suppose that

$$F(\boldsymbol{x}) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{2} (\langle \boldsymbol{a}_i, \boldsymbol{x} \rangle - y_i)^2 \tag{1}$$

where each data point $\boldsymbol{a}_i$ consists of d features and $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}^*$. Suppose the entries of all the vectors $\boldsymbol{a}_i$ are i.i.d. standard Gaussian random variables. In this case, for any fixed $\boldsymbol{x} \in \mathbb{R}^d, \|\nabla f_i(\boldsymbol{x})\|^2 = \|\boldsymbol{a}_i\|^2 \langle \boldsymbol{a}_i, \boldsymbol{x} - \boldsymbol{x}^* \rangle^2$. Let $\bar{\boldsymbol{x}} \triangleq \frac{\boldsymbol{x} - \boldsymbol{x}^*}{\|\boldsymbol{x} - \boldsymbol{x}^*\|}$ and $Y \triangleq \langle \boldsymbol{a}_i, \bar{\boldsymbol{x}} \rangle, \forall i$. Using the fact that a linear combination of independent normal distributions is $\mathcal{N}(\sum_j c_j \mu_j, \sum_j c_j^2 \sigma_j^2)$, $Y \sim \mathcal{N}(0, \|\bar{\boldsymbol{x}}\|^2)$, i.e. $Y \sim \mathcal{N}(0, 1)$, then $Y^2 \sim \chi^2(1)$. For example, from the distribution table of $\chi^2(1)$, $\forall i = 1, 2, \ldots, n$,

$$\mathbb{P}_i \left(\|\nabla f_i(\boldsymbol{x})\|^2 \geq 0.45 \min_j \{\|\boldsymbol{a}_j\|^2\} \|\boldsymbol{x} - \boldsymbol{x}^*\|^2 \right) \geq 0.5$$

In the above case, $\alpha \geq 0.45 \min_j \{\|\boldsymbol{a}_j\|^2\}$ and $\gamma \geq 0.5$ in RUIG, where $\|\boldsymbol{a}_j\|^2 \sim \chi^2(d)$. Then, from the tail bound of $\chi^2(d)$, we have $\mathbb{P}_j(\|\boldsymbol{a}_j\|^2 \geq (1-t)d) \geq 1 - e^{-dt^2/8}, \forall t \in (0, 1)$. In general, α is not small—especially when the data is fairly dense. Furthermore, from the chi-squared distribution, other possible tuples $\left(\frac{\alpha}{\min_j \{\|\boldsymbol{a}_j\|^2\}}, \gamma\right)$ are $\{(0.015, 0.9), (0.1, 0.75), (1.3, 0.25), (2.7, 0.1)\}$. The inequality is for any fixed x, so $\mathcal{D}_\epsilon$ is extended to $\mathcal{D}_0$.

Example 2. $\mu-$**Strongly Convex Function**

(i) Consider $\{f_i(\boldsymbol{x})\}$ μ_i-strongly convex (Defazio et al., 2014) and $\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} F(\boldsymbol{x})$ such that $\nabla f_i(\boldsymbol{x}^*) = \boldsymbol{0}$. By strong convexity, $\|\nabla f_i(\boldsymbol{x})\|^2 \geq (\min_j \mu_j)^2 \|\boldsymbol{x} - \boldsymbol{x}^*\|^2, \forall \boldsymbol{x}, \forall i = 1, 2, \ldots$. In this case, the uniform probability γ degenerates to 1, $\alpha = (\min_j \mu_j)^2$, and $\boldsymbol{x}$ is not restricted to $\mathcal{D}_\epsilon$.

This class includes sum of convex functions such as squared and logistic loss with ℓ_2-regularization.

(ii) A more general function class: $f_i(\boldsymbol{x}) \in \mathcal{H}_1 \cup \mathcal{H}_2$, where $\mathcal{H}_1 := \{g(\boldsymbol{x}) : g(\boldsymbol{x})$ is μ_i-strongly convex and $\nabla g(\boldsymbol{x}^*) = \boldsymbol{0}\}$ and $\mathcal{H}_2 := \{h(\boldsymbol{x}) : h(\boldsymbol{x})$ is not strongly convex$\}$. $f_i(\boldsymbol{x})$ draws from $\mathcal{H}_1$ with probability γ and from $\mathcal{H}_2$ with probability $1-\gamma$, where $0 < \gamma < 1$ and $F(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^n f_i(\boldsymbol{x})$.

Convergence Under RUIG Assumption Under the RUIG assumption, the reciprocal of the step-size (i.e. b_t) in AdaGrad-Norm increases quickly with high probability in Stage I until it exceeds a threshold—for example, ηL—to reach Stage II.

Lemma 1. ***(Two-case high-probability lower bound for b_N in the stochastic setting)*** *For AdaGrad-Norm (Algorithm 1), $\forall \epsilon > 0$, suppose $F(\boldsymbol{x})$ satisfies $(\epsilon, \alpha, \gamma)-$RUIG. For any fixed C, after $N = \left\lceil \frac{C^2 - b_0^2}{\alpha\gamma\epsilon} + \frac{\delta}{\gamma} \right\rceil + 1$ steps, either $b_N > C$ or $\min_j \|\boldsymbol{x}_j - \boldsymbol{x}^*\|^2 \le \epsilon$, with high probability $1 - \exp(-\frac{\delta^2}{2(N\gamma(1-\gamma)+\delta)})$.*

Letting $\delta_1 \triangleq \exp(-\frac{\delta^2}{2(N\gamma(1-\gamma)+\delta)})$, the high probability $1-\delta_1$ is derived by applying the standard Bernstein Inequality (Wainwright, 2019) for the Bernoulli distribution (see Appendix C). When $\gamma N/\log N \to \infty$, let $\delta = \sqrt{4c\gamma(1-\gamma)N\log N}$, then $\delta_1 \le N^{-c}$; On the other hand, if $\gamma N \sim \log N$, let $\delta \sim (\log N)^{t+0.5}$, then $\delta_1 \sim \exp(-(c'\log N)^{2t}) \to 0$ as $N \to \infty$. As long as $\gamma \gg (\log N)^{t+0.5}/N$, the number of iterations $\frac{\delta}{\gamma} \ll N$ in Lemma 1. In the two examples, γ can be chosen to be at least 0.5, which leads to the high probability. In Example 2(i), $\gamma = 1$, every step is deterministic, the probability degenerates to 1.

4 Linear Convergence Rates

Throughout this section, we mainly focus on the linear convergence of AdaGrad-Norm (Algorithm 1) in both stochastic and batch settings. We highlight the robustness of the convergence rates to hyper-parameter tuning by applying our general two-stage framework. Proofs of all theorems and lemmas are in Appendix.

Theorem 1. ***(Convergence in strongly convex and stochastic setting)*** *Consider the AdaGrad-Norm Algorithm in the stochastic setting, suppose that $F(\boldsymbol{x})$ is strongly convex, smooth, almost stationary with $\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} F(\boldsymbol{x})$, and satisfies Restricted Uniform Inequality of Gradients (i.e. with Assumptions (A1a), (A2), (A3), (A4)), then*

Case 1: *If $b_0 > \eta L$, then $\|\boldsymbol{x}_T - \boldsymbol{x}^*\|^2 \le \epsilon$ with high probability $1-\delta_h$ after*

$$T = \left\lceil \frac{b_0 + L\Delta_0/\eta}{\mu} \log \frac{\Delta_0}{\epsilon\delta_h} \right\rceil + 1$$

iterations, where $\Delta_0 = \|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2$;

Case 2: *If $b_0 \le \eta L$, then $\min_i \|\boldsymbol{x}_i - \boldsymbol{x}^*\|^2 \le \epsilon$ with high probability $1 - \delta_h - \exp(-\frac{\delta^2}{2(N\gamma(1-\gamma)+\delta)})$ after*

$$T = \left\lceil \frac{\eta^2 L^2 - b_0^2}{\alpha\gamma\epsilon} + \frac{\delta}{\gamma} + \frac{L(\eta + \Delta/\eta)}{\mu} \log \frac{\Delta}{\epsilon\delta_h} \right\rceil + 1$$

iterations, where $\Delta = \|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2 + \eta^2(\log \frac{\eta^2 L^2}{b_0^2} + 1)$ and $N = \left\lceil \frac{\eta^2 L^2 - b_0^2}{\alpha\gamma\epsilon} + \frac{\delta}{\gamma} \right\rceil$.

Our theorem establishes not only the robustness to hyper-parameters of the AdaGrad-Norm algorithm but also, more importantly, the strong linear convergence in the stochastic setting. To put the theorem in context, we compare with the sub-linear convergence rate of AdaGrad-Norm (i.e., $T = \mathcal{O}\left(1/\epsilon^2\right)$) in Levy et al. (2018); Levy (2017); Ward et al. (2018); Li and Orabona (2018). The key breakthrough in our theorem is that we use a novel assumption in high dimensional probability (c.f. RUIG) and utilize the nice landscape property at the solution (c.f. Assumption (A4)), instead of following the standard analysis of SGD where it is often assumed that there is noise at the solution, $\mathbb{E}_{\xi_j}[\|\nabla f_{\xi_j}(\boldsymbol{x}) - \nabla F(\boldsymbol{x})\|^2] \le \hat{\sigma}^2, \forall \boldsymbol{x}$.

The high probability guarantee can be verified in both stages: In Stage I, the high probability $\delta_1 \triangleq \exp(-\frac{\delta^2}{2(N\gamma(1-\gamma)+\delta)})$ is guaranteed by the high probability explanation in Lemma 1; In Stage II, δ_h is derived from removing expectation in $\mathbb{E}\|\boldsymbol{x} - \boldsymbol{x}^*\|^2$ with high probability $1-\delta_h$ by Markov Inequality. In general, δ_h is appropriately chosen to be a small term.

Remark 1. *The classic result (Needell et al., 2016) for SGD in the strongly convex setting with $\sigma^2 \triangleq \mathbb{E}\|\nabla f_i(\boldsymbol{x}^*)\|^2 = 0$ is: with stepsize $\eta_t = \frac{1}{2\sup_i L_i}$, after $T = \frac{2\sup_i L_i}{\mu}\log\frac{2\Delta_0}{\epsilon}$ iterations, $\mathbb{E}\|\boldsymbol{x}_T - \boldsymbol{x}^*\|^2 \le \epsilon$. Theorem 1 recovers the convergence rate up to a factor difference of Δ_0 in multiplier and δ_h in the log term with high probability, if $b_0 > \sup_i L_i$. Hence, if the initialization of $\boldsymbol{x}_0$ is extremely bad, the convergence is relatively slow. However, with tuning $\eta = \Theta(\Delta_0)$, the convergence rate is $(c_1\frac{L}{\mu} + c_2)\log\frac{\Delta_0}{\epsilon\delta_h}$ as expected. See the numerical experiments of extreme initialization of $\boldsymbol{x}_0$ and corresponding tuning η in Appendix E.1.*

In the batch setting, the full gradient at each step is available. Now, the moving direction becomes noiseless (i.e. $G_j = \nabla F(\boldsymbol{x}_j)$), and the uniform probability γ in $\mathbb{P}_i(\|\nabla f_i(\boldsymbol{x})\|^2 \ge \alpha\|\boldsymbol{x} - \boldsymbol{x}^*\|^2) \ge \gamma$ degenerates to 1. Hence, the linear convergence rate is guaranteed in Stage II instead of with high probability.

Theorem 2. ***(Convergence in strongly convex and batch setting)*** *Consider the AdaGrad-Norm Algorithm in the batch setting, suppose that $F(\boldsymbol{x})$ is*

$L-$smooth and μ-strongly convex (i.e. with Assumptions (A1a) and (A2)), and $\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} F(\boldsymbol{x})$. Then $\min_{0\le i\le T-1} \|\boldsymbol{x}_i - \boldsymbol{x}^*\|^2 \le \epsilon$ after

Case 1: *If $b_0 > \eta\frac{\mu+L}{2}$,*

$$T = 1 + \left\lceil \max\left\{\frac{L(1+\Delta_0/\eta^2)}{\mu}, \frac{\mu+L}{2\mu}\right\} \log\frac{\Delta_0}{\epsilon} \right\rceil$$

iterations, where $\Delta_0 = \|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2$;

Case 2: *If $b_0 \le \eta\frac{\mu+L}{2}$,*

$$T = 1 + \left\lceil \max\left\{\frac{L(1+\Delta/\eta^2)}{\mu}, \frac{\mu+L}{2\mu}\right\} \log\frac{\Delta}{\epsilon} + \frac{\log(\eta^2(\mu+L)^2/4b_0^2)}{\log(1+4\mu^2\epsilon/(\mu+L)^2)} \right\rceil$$

iterations, where $\Delta = \|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2 + \eta^2(\log\frac{(\mu+L)^2}{4b_0^2} + 1)$.

Remark 2. *Let $b_0 > \frac{\mu+L}{2}$ and $\eta = \Theta(\sqrt{\Delta_0})$, then $T = (c_1\frac{L}{\mu} + c_2)\log\frac{\Delta_0}{\epsilon}$. Theorem 2 recovers the classic result of GD with constant stepsize—whose $T = \frac{(\mu+L)^2}{4\mu L}\log\frac{\Delta_0}{\epsilon}$—up to a constant factor difference. Note that the order of η w.r.t Δ_0 is different from $\eta = \mathcal{O}(\Delta_0)$ in Remark 1, but the effect of tuning η in both settings for extreme case is similar.*

For non-convex functions that satisfy the $\mu-$PL inequality, we extend the proof of linear convergence by bounding $F(\boldsymbol{x}_j) - F^*$ at each step in Theorem 3.

Theorem 3. ***(Convergence in non-convex batch setting)*** *Consider the AdaGrad-Norm Algorithm in the batch setting, suppose that $F(\boldsymbol{x})$ is $L-$smooth and satisfies the $\mu-$PL inequality (i.e. with Assumptions (A1b) and (A2)), and $F^* = \inf_{\boldsymbol{x}} F(\boldsymbol{x}) > -\infty$, then*

Case 1: *If $b_0 > \eta L$, $\min_{0\le i\le T-1} F(\boldsymbol{x}_i) - F^* \le \epsilon$ after*

$$T = \left\lceil \frac{b_0 + \frac{2}{\eta}(F(\boldsymbol{x}_0) - F^*)}{\mu\eta} \log\frac{F(\boldsymbol{x}_0) - F^*}{\epsilon} \right\rceil + 1$$

iterations;

Case 2: *If $b_0 \le \eta L$, $\min_{0\le i\le T-1} F(\boldsymbol{x}_i) - F^* \le \epsilon$ after*

$$T = \left\lceil \frac{\log(\eta^2L^2/b_0^2)}{\log(1+2\mu\epsilon/(\eta L)^2)} + \frac{\eta L + (2/\eta)\Delta}{\mu\eta}\log\frac{\Delta}{\epsilon} \right\rceil + 1$$

iterations, where $\Delta = \frac{\eta^2 L}{2}(1 + 2\log\frac{\eta L}{b_0}) + F(\boldsymbol{x}_0) - F^*$.

Compared with the result in Ward et al. (2018), our theory—using additional Assumption (A1b)—significantly improves from sublinear convergence rate to linear convergence in Stage II. The Assumption (A1b) is a generally well-known condition satisfied by a wide range of non-convex optimization problems including over-parameterized neural networks (Soltanolkotabi et al., 2019; Kleinberg et al., 2018; Li and Yuan, 2017; Vaswani et al., 2018; Wu et al., 2019). For the convergence of AdaGrad-Norm in the over-parameterized problem, Wu et al. (2019) proved the same convergence rate as ours. The convergence rate in Wu et al. (2019) was tailored to a multi-layer network with two fully connected layers. Our theorem is for general functions, however, with some additional assumptions such as $\mu-$PL inequality.

5 Two-Stage Framework

We develop the following two-stage proof framework to analyze the convergence rate starting from any point x_0 and any initial stepsize parameter b_0 in both the stochastic and batch settings. See the demonstration of the two-stage behavior in Figure 1.

Stage I If we initialize with small b_0—i.e. our initial step size is large—we can get a better convergence in Stage I than SGD with constant stepsize. In Stage I, b_0 grows to some given level, such as L and $\frac{\mu+L}{2}$, which depends on different settings, with deterministic iterations unless the function achieves a global minimal with tolerance ϵ, i.e. $\|\boldsymbol{x} - \boldsymbol{x}^*\|^2 \le \epsilon$. Details are in two-case lemmas: Lemma 1 and 2. By Lemma 3, $\|\boldsymbol{x} - \boldsymbol{x}^*\|$ is bounded by radius $\Delta = \mathcal{R}(b_0, \|\boldsymbol{x}_0 - \boldsymbol{x}^*\|, C)$ before b_t grows up to C, instead of blowing up.

Stage II After Stage I, b_t exceeds a certain threshold deterministically in the batch setting and with high probability in the stochastic setting. Conditioned on this, the update is a contraction in the strongly convex setting, i.e. $\|\boldsymbol{x}_{j+1} - \boldsymbol{x}^*\|^2 \le (1 - \mathcal{P}(b_{\max}, \mu, L))\|\boldsymbol{x}_j - \boldsymbol{x}^*\|^2$, where $\mathcal{P}$ is a function s.t. $0 < \mathcal{P}(b_{\max}, \mu, L) < 1$. $b_{\max}$ is bounded by Lemma 4.

5.1 Growth of b_t in Stage I

We introduce some lemmas that are critical in the proof of the growth of b_t in Stage I in the section. Note that in the stochastic setting, RUIG is a sufficient condition for b_t's growth in Stage I to achieve a certain threshold with high probability, so the corresponding two-case growth of b_t is provided in Lemma 1. Detailed proofs are provided in Appendix C.

Lemma 2. ***(Two-case lower bound for b_N in the batch setting)*** *For fixed $\epsilon \in (0,1)$ and C, consider AdaGrad-Norm in the batch setting to minimize the objective function $F(\boldsymbol{x})$, then*

(a) If $F(\boldsymbol{x})$ is $\mu-$strongly convex, then after $N = \left\lceil \frac{\log(C^2/b_0^2)}{\log(1+\mu^2\epsilon/C^2)} \right\rceil + 1$ iterations, either $b_N > C$ or $\min_{0\le i\le N-1} \|\boldsymbol{x}_i - \boldsymbol{x}^*\|^2 \le \epsilon$, where $\boldsymbol{x}^* = \arg\min F(\boldsymbol{x})$;

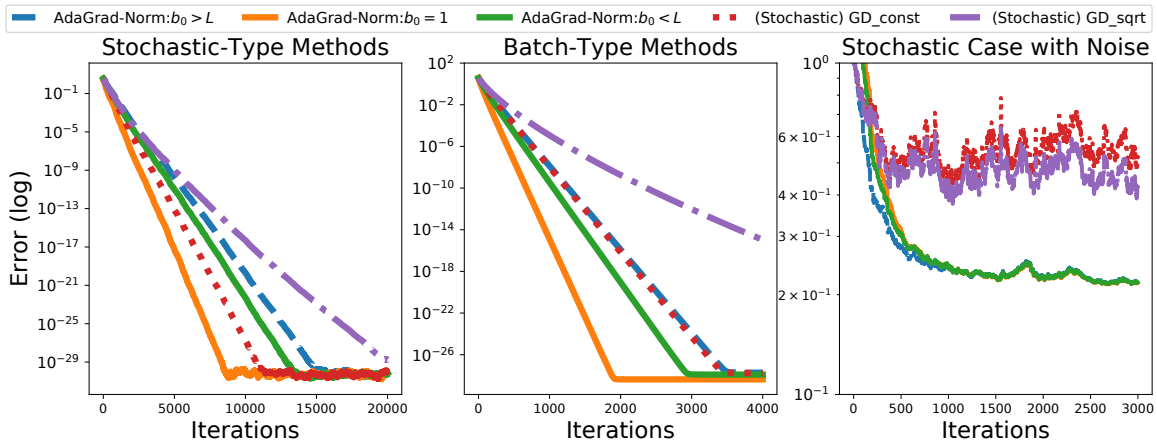

Figure 2: Error in log scale of least square problem: $F(\boldsymbol{x}) = \frac{1}{2n}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|^2$. The left and central figures show error $\|\boldsymbol{x}_t - \boldsymbol{x}^*\|^2$ in the noiseless case. The right figure shows the loss $F(\boldsymbol{x}_t)$ in the noisy case.

(b) If $F(\boldsymbol{x})$ is a non-convex function satisfying $\mu-$PL inequality, then after $N = \left\lceil \frac{\log(C^2/b_0^2)}{\log(1+2\mu\epsilon/C^2)} \right\rceil$ iterations, either $b_N > C$ or $\min_{0\le i\le N-1} F(\boldsymbol{x}_i) - F^* \le \epsilon$, where $F^* = \inf_{\boldsymbol{x}} F(\boldsymbol{x}) > -\infty$.

Lemma 1 and 2 depict the two-stage growth of b_t, which is less and over certain thresholds, in stochastic and batch settings, respectively.

Remark 3. *In Lemma 1 and 2, we provide the worst cases for the growth of b_t. However, b_t actually grows very quickly in practice, especially in the stochastic setting. For Lemma 2, since $\log(1+x) \sim x$, for $x = \mu^2\epsilon/C^2$ small, $N \sim \frac{C^2}{\mu^2\epsilon}\log\frac{C^2}{b_0}$.*

Lemma 3. ***(Upper bound for $\|\boldsymbol{x}_{J-1} - \boldsymbol{x}^*\|^2$)*** *For any fixed C and η, consider AdaGrad-Norm in either stochastic or batch setting with $G_j(\boldsymbol{x}^*) = 0, \forall j$ (stochastic: $\nabla f_j(\boldsymbol{x}^*) = \boldsymbol{0}$; batch: $\nabla F(\boldsymbol{x}^*) = \boldsymbol{0}$) using update rule $b_{j+1}^2 = b_j^2 + \|G_j\|^2$. Suppose that J is the first index s.t. $b_J > C$, then*

$$\|\boldsymbol{x}_{J-1} - \boldsymbol{x}^*\|^2 \le \|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2 + \eta^2(\log(C^2/b_0^2) + 1)$$

Lemma 3 gives an upper bound on the distance between the snapshot before contraction and the optimal solution $\boldsymbol{x}^*$ i.e. $\|\boldsymbol{x}_{J-1} - \boldsymbol{x}^*\|$. It guarantees that the extreme distance to $\boldsymbol{x}^*$ is always bounded during AdaGrad-Norm updates, even without projection, or additional assumption, for example, $\forall t, \|\boldsymbol{x}_t - \boldsymbol{x}^*\|_2 \le D$, for some constant D, in Adam (Kingma and Ba, 2014) and AMSGrad (Reddi et al., 2018b).

5.2 Upper Bounds on b_t in Stage II

In Stage II, we focus on the maximum value that b_t can obtain during the optimization process.

Lemma 4. ***(Upper bound for $b_{\max}$)*** *Consider AdaGrad-Norm in either stochastic or batch setting with $G_j(\boldsymbol{x}^*) = \boldsymbol{0}, \forall j$ (stochastic: $\nabla f_j(\boldsymbol{x}^*) = \boldsymbol{0}$; batch: $\nabla F(\boldsymbol{x}^*) = \boldsymbol{0}$), for any fixed $C \ge \eta L$, if J is the first index s.t. $b_J > C$, then $b_{\max} \triangleq \max_{l\ge 0} b_{J+l}$ is upper bounded by*

$$b_{\max} \le C + (L/\eta)(\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2 + \eta^2(\log(C^2/b_0^2) + 1))$$

Lemma 4 indicates that even though b_t^2 increases due to adding $\|G_t\|^2$ to b_t^2 at each iteration, it is always upper bounded by $b_{\max}$. The asymptotic behavior of the stepsize (i.e. $\frac{\eta}{b_t}$) is $\mathcal{O}(\frac{1}{\sqrt{t}})$ at first, and it approaches to a constant in the end as $\boldsymbol{x}_t \to \boldsymbol{x}^*$, which also explains the auto-tuning nature of AdaGrad-Norm.

After b_t exceeds certain thresholds like $\eta L/2$, the following Lemma 5 shows that AdaGrad-Norm is indeed a descent algorithm, i.e. $\|\boldsymbol{x}_t - \boldsymbol{x}^*\|^2$ will not increase subsequently, so we can take $\boldsymbol{x}_T$ as $\boldsymbol{x}_{best}$ in Stage II.

Lemma 5. ***(Descent lemma for $\|\boldsymbol{x}_t - \boldsymbol{x}^*\|^2$)*** *Once $b_j > \eta L/2$, Algorithm 1 is a descent algorithm for the error $\|\boldsymbol{x}_t - \boldsymbol{x}^*\|^2$. Furthermore, if $\|\boldsymbol{x}_{j-1} - \boldsymbol{x}^*\|^2 \le \Delta$, then $\forall l \ge 0$, x_{j-1+l} will stay in the ball centering at $\boldsymbol{x}^*$ with radius $\sqrt{\Delta}$, i.e. $\|\boldsymbol{x}_{j-1+l} - \boldsymbol{x}^*\|^2 \le \Delta$.*

6 Numerical Experiments

In this section, we present numerical results to compare AdaGrad-Norm and (stochastic) Gradient Descent methods with fixed stepsize $\eta_j = \frac{1}{b_0}$ (GD_const or SGD_const) or square-root decaying stepsize $\eta_j = \frac{1}{b_0+0.2\sqrt{j}}$ (GD_sqrt or SGD_sqrt), in the stochastic and batch settings, respectively.

Consider the least square problem from (1). The Lipschitz constants are $L_i = \|\boldsymbol{a}_i\|^2$ and $\bar{L} =$

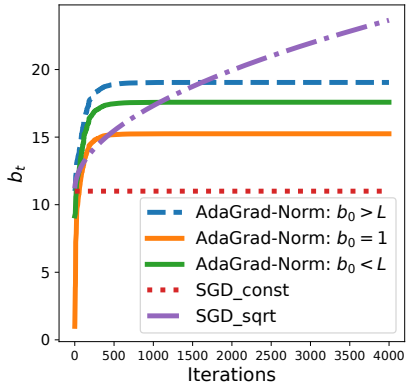

Figure 3: Growth of b_t in stochastic setting using different algorithms: AdaGrad-Norm and SGD.

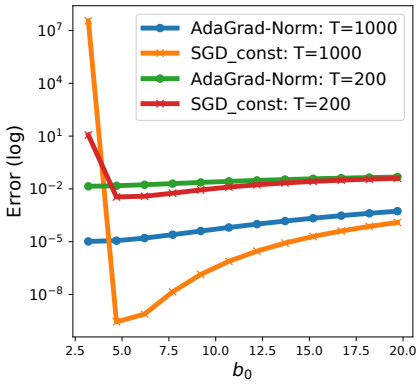

Figure 4: Robustness of AdaGrad-Norm: $\|\boldsymbol{x}_T - \boldsymbol{x}^*\|^2$ in log scale with different choices of b_0.

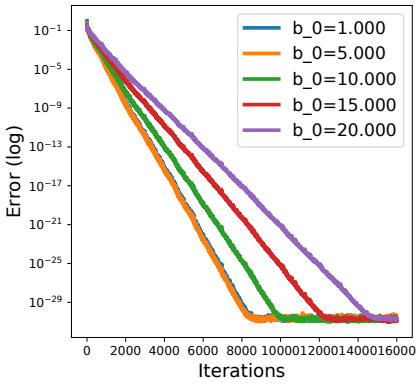

Figure 5: Comparison of different initial stepsizes ($\eta_0 = \frac{\eta}{b_0}$) of AdaGrad-Norm with $\|\boldsymbol{x}_t - \boldsymbol{x}^*\|^2$ in log scale.

$\sum_{i=1}^n \frac{1}{n}\|\boldsymbol{a}_i\|^2 = \frac{1}{n}\|\boldsymbol{A}\|_F^2$, respectively. In the experiments, we setup the noiseless problem with a 1000×20 random matrix $\boldsymbol{A}$ and a vector $\boldsymbol{x}^*$ with $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}^*$.

We first illustrate the linear convergence and robustness in the noiseless cases. Figure 2 verifies the expected linear convergence of AdaGrad-Norm in the stochastic and batch settings. In order to compare the convergence rates of AdaGrad-Norm with vanilla (S)GD, we choose $\eta = 1$ and a $b_0 > \sup_i L_i \triangleq L$ to prevent (S)GD from blowing up. AdaGrad-Norm with $b_0 = 1$, $b_0 < L$ and $b_0 > L$ have similar linear convergence as (S)GD_const, up to a constant difference, while (S)GD_sqrt converges more slowly.

Figure 2 shows that even simply setting $b_0 = 1$, AdaGrad-Norm has a better convergence rate than that of non-adaptive (S)GD, since AdaGrad-Norm takes a big stepsize when $\boldsymbol{x}$ is far away from $\boldsymbol{x}^*$, and then very small stepsize around $\boldsymbol{x}^*$ when b_j grows to a value $b_{\max}$. Eventually, b_j converges to a constant value since $\|G_j\| \to 0$ or $\|\nabla F(\boldsymbol{x}_j)\| \to 0$ as $\boldsymbol{x}_j \to \boldsymbol{x}^*$. In the noisy case (Figure 2 Right), AdaGrad-Norm has a similar convergence rate up to a constant factor and achieves a better approximation of $\boldsymbol{x}^*$, with less vibrations compared to SGD_const or SGD_sqrt.

Figure 3 shows that the growth of b_t in AdaGrad-Norm is similar to SGD_sqrt at first, but after exceeding the threshold and approximately reaching $b_{\max}$, b_t's growth is similar to SGD_const. Figure 4 and 5 show that the linear convergence rates of AdaGrad-Norm are more robust to the choice of initial stepsize $1/b_0$ compared to SGD_const. The error $\|\boldsymbol{x}_T - \boldsymbol{x}^*\|^2$ of AdaGrad-Norm after T iterations remains stable for a relatively arbitrary range of b_0 while the error of SGD_const blows up at first and then decreases significantly when b_0 approaches to L since SGD_const is sensitive to the choice of stepsize.

The result of experiment on one hidden layer over-parameterized neural net and experimental details are in Appendix E.2. Figure 12 shows that (1) AdaGrad-Norm converges faster than GD_const and almost linearly; (2) The gradients of the first few iterations are often big enough to accumulate to exceed ηL, which empirically verifies Assumption (A3).

7 Discussions

In this work, we propose the notion of RUIG to measure the uniform lower bound of gradients with respect to $\|\boldsymbol{x} - \boldsymbol{x}^*\|^2$ in a restricted region. We propose a two-stage framework and use it to prove the non-asymptotic convergence rates for AdaGrad-Norm starting from any initialization and without knowing the smooth or strongly convex parameter a priori. In the stochastic setting, we prove linear convergence with high probability under strongly convex and RUIG assumptions, without requiring a uniform bound on $\mathbb{E}\|G_t\|^2$. In the batch setting, we prove deterministic linear convergence for strongly convex functions and non-convex functions with PL inequality. Both theoretical and numerical results validate the robustness of AdaGrad-Norm starting at any initial stepsize.

There are still some open problems to be solved: First, drawing on Needell et al. (2016), we may improve $L = \sup_i L_i$ in convergence rates to $\bar{L} = \frac{1}{n}\sum_i L_i$ with importance sampling. Second, extending Assumption (A4) to the weak growth condition $\mathbb{E}_{\xi_t}[\|\nabla f_{\xi_t}(\boldsymbol{x}_t)\|^2] \leq M\|\nabla F(\boldsymbol{x}_t)\|^2 + \sigma^2$ in Cevher and Vũ (2019) may lead to a more general result. Third, since AdaGrad-Norm is fundamentally related to both Adam and AMSGrad, extending our theoretical guarantees to the two algorithms is an exciting direction for future research.

Acknowledgments

We thank anonymous reviewers, Amelia Henriksen, Jiayi Wei, and Thomas Herben for helpful comments, which greatly improved the manuscript. We thank Purnamrita Sarkar for helpful discussion. This project was supported in part by AFOSR MURI Award N00014-17-S-F006.

References

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

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

Léon Bottou and Yann L Cun. Large scale online learning. In *Advances in neural information processing systems*, pages 217–224, 2004.

Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311, 2018.

Léon Bottou. *Une Approche théorique de l'Apprentissage Connexionniste: Applications à la Reconnaissance de la Parole*. PhD thesis, Université de Paris XI, Orsay, France, 1991.

Stephen G Nash and Jorge Nocedal. A numerical study of the limited memory bfgs method and the truncated-newton method for large scale optimization. *SIAM Journal on Optimization*, 1(3):358–372, 1991.

Dimitri P Bertsekas. *Nonlinear programming*. Athena Scientific, 1999.

Yu Nesterov. Smooth minimization of non-smooth functions. *Mathematical programming*, 103(1):127–152, 2005.

Simon Haykin et al. Cognitive radio: brain-empowered wireless communications. *IEEE journal on selected areas in communications*, 23(2):201–220, 2005.

Sébastien Bubeck et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. *arXiv preprint arXiv:1811.03962*, 2018.

Difan Zou, Yuan Cao, Dongruo Zhou, and Quanquan Gu. Stochastic gradient descent optimizes over-parameterized deep relu networks. *arXiv preprint arXiv:1811.08888*, 2018a.

Eric Moulines and Francis R Bach. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In *Advances in Neural Information Processing Systems*, pages 451–459, 2011.

Deanna Needell, Nathan Srebro, and Rachel Ward. Stochastic gradient descent, weighted sampling, and the randomized kaczmarz algorithm. *Mathematical Programming*, 155(1):549–573, Jan 2016. ISSN 1436-4646.

Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1-2):83–112, 2017.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, pages 315–323, 2013.

Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in neural information processing systems*, pages 1646–1654, 2014.

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

H Brendan McMahan and Matthew Streeter. Adaptive bound optimization for online convex optimization. *arXiv preprint arXiv:1002.4908*, 2010.

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

Jean Lafond, Nicolas Vasilache, and Léon Bottou. Diagonal rescaling for neural networks. Technical report, arXiV:1705.09319, 2017.

Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. *CoRR*, abs/1904.09237, 2018a.

Vatsal Shah, Anastasios Kyrillidis, and Sujay Sanghavi. Minimum norm solutions do not always generalize well for over-parameterized problems. *arXiv preprint arXiv:1811.07055*, 2018.

Fangyu Zou, Li Shen, Zequn Jie, Weizhong Zhang, and Wei Liu. A sufficient condition for convergences of adam and rmsprop. *arXiv preprint arXiv:1811.09358*, 2018b.

Matthew Staib, Sashank J Reddi, Satyen Kale, Sanjiv Kumar, and Suvrit Sra. Escaping saddle points with adaptive gradient methods. *arXiv preprint arXiv:1901.09149*, 2019.

Kfir Levy. Online to offline conversions, universality and adaptive minibatch sizes. In *Advances in Neural*

Information Processing Systems, pages 1613–1622, 2017.

Rachel Ward, Xiaoxia Wu, and Leon Bottou. Adagrad stepsizes: Sharp convergence over nonconvex landscapes, from any initialization. *arXiv preprint arXiv:1806.01811*, 2018.

Xiaoxia Wu, Rachel Ward, and Léon Bottou. Wngrad: learn the learning rate in gradient descent. *arXiv preprint arXiv:1803.02865*, 2018.

Yehuda Kfir Levy, Alp Yurtsever, and Volkan Cevher. Online adaptive methods, universality and acceleration. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 6500–6509. Curran Associates, Inc., 2018.

Xiaoyu Li and Francesco Orabona. On the convergence of stochastic gradient descent with adaptive stepsizes. *arXiv preprint arXiv:1805.08114*, 2018.

Lukas Balles and Philipp Hennig. Dissecting adam: The sign, magnitude and variance of stochastic gradients. In *International Conference on Machine Learning*, pages 413–422, 2018.

Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. signSGD: Compressed Optimisation for Non-Convex Problems. In *International Conference on Machine Learning (ICML-18)*, 2018.

Mahesh Chandra Mukkamala and Matthias Hein. Variants of rmsprop and adagrad with logarithmic regret bounds. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2545–2553. JMLR. org, 2017.

Zaiyi Chen, Yi Xu, Enhong Chen, and Tianbao Yang. SADAGRAD: Strongly adaptive stochastic gradient methods. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 913–921, Stockholmsmssan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

Sharan Vaswani, Francis Bach, and Mark Schmidt. Fast and faster convergence of sgd for over-parameterized models and an accelerated perceptron. *arXiv preprint arXiv:1810.07288*, 2018.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

Simon S. Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. In *International Conference on Learning Representations*, 2019.

Yi Zhou, Junjie Yang, Huishuai Zhang, Yingbin Liang, and Vahid Tarokh. SGD converges to global minimum in deep learning via star-convex path. In *International Conference on Learning Representations*, 2019.

Raef Bassily, Mikhail Belkin, and Siyuan Ma. On exponential convergence of sgd in non-convex over-parametrized learning. *arXiv preprint arXiv:1811.02564*, 2018.

Nicolas L. Roux, Mark Schmidt, and Francis R. Bach. A stochastic gradient method with an exponential convergence _rate for finite training sets. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 2663–2671. Curran Associates, Inc., 2012.

Volkan Cevher and Bng Công Vũ. On the linear convergence of the stochastic gradient method with constant step-size. *Optimization Letters*, 13(5):1177–1187, Jul 2019. ISSN 1862-4480.

Mark Schmidt and Nicolas Le Roux. Fast convergence of stochastic gradient descent under a strong growth condition. *arXiv preprint arXiv:1308.6370*, 2013.

Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

Mahdi Soltanolkotabi, Adel Javanmard, and Jason D Lee. Theoretical insights into the optimization landscape of over-parameterized shallow neural networks. *IEEE Transactions on Information Theory*, 65(2):742–769, 2019.

Robert Kleinberg, Yuanzhi Li, and Yang Yuan. An alternative view: When does sgd escape local minima? In *International Conference on Machine Learning*, pages 2703–2712, 2018.

Yuanzhi Li and Yang Yuan. Convergence analysis of two-layer neural networks with relu activation. In *Advances in Neural Information Processing Systems*, pages 597–607, 2017.

Xiaoxia Wu, Simon S Du, and Rachel Ward. Global convergence of adaptive gradient methods for an over-parameterized neural network. *arXiv preprint arXiv:1902.07111*, 2019.

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