The Step Decay Schedule: A Near Optimal, Geometrically Decaying Learning Rate Procedure For Least Squares*

Rong Ge[1], Sham M. Kakade[2], Rahul Kidambi[3], and Praneeth Netrapalli[4]

[1]Duke University, Durham, NC, USA, rongge@cs.duke.edu
[2]University of Washington, Seattle, WA, USA, sham@cs.washington.edu
[3]Cornell University, Ithaca, NY, USA, rkidambi@cornell.edu
[4]Microsoft Research, Bangalore, KA, India, praneeth@microsoft.com.

Abstract

Minimax optimal convergence rates for numerous classes of stochastic convex optimization problems are well characterized, where the majority of results utilize iterate averaged stochastic gradient descent (SGD) with polynomially decaying step sizes. In contrast, the behavior of SGDs final iterate has received much less attention despite the widespread use in practice. Motivated by this observation, this work provides a detailed study of the following question: what rate is achievable using the final iterate of SGD for the streaming least squares regression problem with and without strong convexity?

First, this work shows that even if the time horizon T (i.e. the number of iterations that SGD is run for) is known in advance, the behavior of SGDs final iterate with *any* polynomially decaying learning rate scheme is highly sub-optimal compared to the statistical minimax rate (by a condition number factor in the strongly convex case and a factor of $\sqrt{T}$ in the non-strongly convex case). In contrast, this paper shows that *Step Decay* schedules, which cut the learning rate by a constant factor every constant number of epochs (i.e., the learning rate decays geometrically) offer significant improvements over *any* polynomially decaying step size schedule. In particular, the behavior of the final iterate with step decay schedules is off from the statistical minimax rate by only *log* factors (in the condition number for the strongly convex case, and in T in the non-strongly convex case). Finally, in stark contrast to the known horizon case, this paper shows that the anytime (i.e. the limiting) behavior of SGDs final iterate is poor (in that it queries iterates with highly sub-optimal function value infinitely often, i.e. in a limsup sense) irrespective of the stepsize scheme employed. These results demonstrate the subtlety in establishing optimal learning rate schedules (for the final iterate) for stochastic gradient procedures in fixed time horizon settings.

1 Introduction

Large scale machine learning relies almost exclusively on stochastic optimization methods (Bottou and Bousquet, 2007), which include stochastic gradient descent (SGD) (Robbins and Monro, 1951) and its variants Duchi et al. (2011); Johnson and Zhang (2013). In this work, we restrict our attention to the SGD algorithm where we are concerned with the behavior of the final iterate (i.e. the last point when we terminate the algorithm). A majority of (minimax optimal) theoretical results for SGD focus on polynomially decaying stepsizes (Dekel et al., 2012; Rakhlin et al., 2012; Lacoste-Julien et al., 2012; Bubeck, 2014) (or constant stepsizes (Bach and Moulines, 2013; Défossez and Bach, 2015; Jain et al., 2016) for the case of least squares regression) coupled with iterate averaging (Ruppert, 1988; Polyak and Juditsky, 1992) to achieve minimax optimal rates of convergence. However, practical SGD implementations typically return the final iterate of a stochastic gradient procedure. This line of work in theory (based on iterate averaging) and its discrepancy with regards to practice leads to the question with regards to the behavior of

*This paper appears in the proceedings of the conference in Neural Information Processing Systems (NeurIPS), 2019, held in Vancouver Canada.

Algorithm 1: Step Decay scheme

Input: Initial vector $\mathbf{w}$, starting learning rate η_0, number of iterations T

Output: $\mathbf{w}$

for $\ell \leftarrow 1$ **to** $\log T$ **do**

  $\eta_\ell \leftarrow \eta_0/2^\ell$

  for $t \leftarrow 1$ **to** $T/\log T$ **do**

    $\mathbf{w} \leftarrow \mathbf{w} - \eta_\ell \widehat{\nabla} f(\mathbf{w})$

  end

end

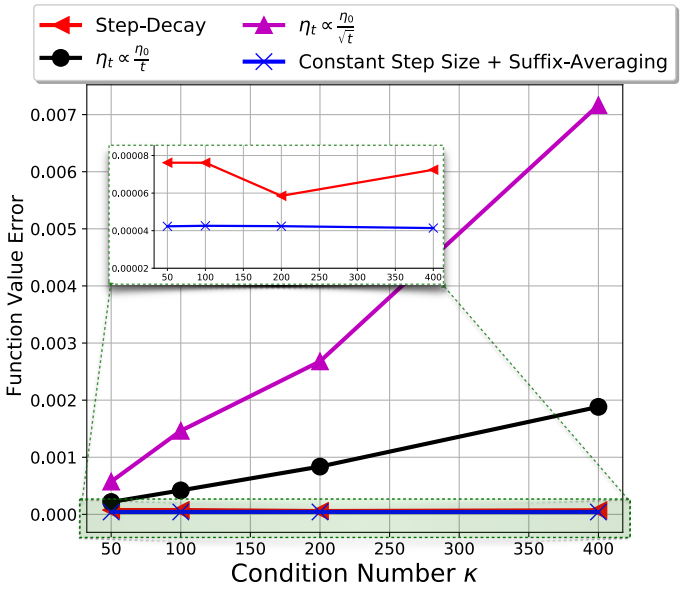

Figure 1: (Left) The Step Decay scheme for stochastic gradient descent. Note that the algorithm requires just two parameters - the starting learning rate η_0 and number of iterations T.
(Right) Plot of function value error vs. condition number for the final iterate of polynomially decaying stepsizes i.e., equation(5,6), step-decay schedule (Algorithm 1) compared against the minimax optimal suffix averaged iterate with a constant stepsize (Jain et al., 2016) for a synthetic two-dimensional least squares regression problem(1). The condition number κ is varied as $\{50, 100, 200, 400\}$. Exhaustive grid search is performed on starting stepsize and decay parameters. Initial excess risk is $d\sigma^2$ and the algorithm is run for $T = \kappa^2_{\max} = 400^2$ steps (for all experiments); results are averaged over 5 random seeds. Observe that the final iterate's error grows linearly as a function of the condition number κ for the polynomially decaying stepsize schemes, whereas, the error does not grow as a function of κ for the geometric "step-decay" stepsize scheme. See section E.1 in the appendix for details.

SGD's final iterate. Indeed, this question has motivated several efforts in stochastic convex optimization literature as elaborated below.

Non-Smooth Stochastic Optimization: The work of Shamir (2012) raised the question with regards to the behavior of SGD's final iterate for non-smooth stochastic optimization (with/without strong convexity). The work of Shamir and Zhang (2012) answered this question, indicating that SGD's final iterate with polynomially decaying stepsizes achieves near minimax rates (up to log factors) in an anytime (i.e. in a limiting) sense (when number of iterations SGD is run for is not known in advance). Under specific choices of step size sequences, Shamir and Zhang (2012)'s result on SGD's final iterate is tight owing to the recent work of Harvey et al. (2018). More recently Jain et al. (2019) presented an approach indicating that a more nuanced stepsize sequence serves to achieve minimax rates (up to constant factors) for the non-smooth stochastic optimization setting when the end time T is known in advance.

Least Squares Regression (LSR): In contrast to the non-smooth setting, the state of our understanding of SGD's final iterate for smooth stochastic convex optimization, or, say, the streaming least squares regression setting is far less mature – this gap motivates our paper's contributions. In particular, this paper studies SGD's final iterate behavior under various stepsize choices for least squares regression (with and without strong convexity). The use of SGD's final iterate for the least mean squares objective has featured in several efforts (Widrow and Hoff, 1960; Proakis, 1974; Widrow and Stearns, 1985; Roy and Shynk, 1990), but these results *do not* achieve minimax rates of convergence, which leads to the following question:

" Can polynomially decaying stepsizes (known to achieve minimax rates when coupled with iterate averaging (Ruppert, 1988; Polyak and Juditsky, 1992)) offer minimax optimal rates on SGD's *final* iterate when optimizing the streaming least squares regression objective? If not, is there *any* other family of stepsizes that can guarantee minimax rates on the final iterate of stochastic gradient descent? "

	Assumptions	Minimax rate	Rate w/ Final iterate using best poly-decay	Rate w/ Final iterate using Step Decay
General convex functions	$\mathbb{E}\left[\left\|\widehat{\nabla} f\right\|^2\right] \leq G^2$ $\text{Diam}\left(\text{ConstraintSet}\right) \leq D$	$\frac{GD}{\sqrt{T}}$	$\Theta\left(\frac{GD}{\sqrt{T}} \cdot \log T\right)$ (Shamir and Zhang, 2012; Harvey et al., 2018)	–
Non-strongly convex least squares regression	Eq. (3)	$\frac{\sigma^2 d}{T}$	$\Omega\left(\frac{\sigma^2 d}{T} \cdot \frac{\sqrt{T}}{\log T}\right)$ (This work - Theorem 1)	$\mathcal{O}\left(\frac{\sigma^2 d}{T} \cdot \log T\right)$ (This work - Theorem 2)
General strongly convex functions	$\mathbb{E}\left[\left\|\widehat{\nabla} f\right\|^2\right] \leq G^2$ $\nabla^2 f \succeq \mu \mathbf{I}$	$\frac{G^2}{\mu T}$	$\Theta\left(\frac{G^2}{\mu T} \cdot \log T\right)$ (Shamir and Zhang, 2012; Harvey et al., 2018)	–
Strongly convex least squares regression	Eq. (3) $\nabla^2 f \succeq \mu \mathbf{I}$	$\frac{\sigma^2 d}{T}$	$\Omega\left(\frac{\sigma^2 d}{T} \cdot \kappa\right)$ (This work - Theorem 1)	$\mathcal{O}\left(\frac{\sigma^2 d}{T} \cdot \log T\right)$ (This work - Theorem 2)

Table 1: Comparison of sub-optimality for *final* iterate of SGD (i.e., $\mathbb{E}\left[f(\mathbf{w}_T)\right] - f(\mathbf{w}^*)$) for stochastic convex optimization problems. This paper's focus is on SGD's final iterate for streaming least squares regression. The minimax rate refers to the best possible worst case rate with access to stochastic gradients (typically achieved with iterate averaging methods (Polyak and Juditsky, 1992; Dekel et al., 2012; Rakhlin et al., 2012)); the red shows the multiplicative factor increase (over the minimax rate) using the final iterate, under two different learning rate schedules - the polynomial decay and the step decay (refer to Algorithm 1). Polynomial decay schedules are of the form $\eta_t \propto 1/t^\alpha$ (for appropriate $\alpha \in [0.5, 1]$). For the general convex cases below, the final iterate with a polynomial decay scheme is off minimax rates by a $\log T$ factor (in an anytime/limiting sense) (Shamir and Zhang, 2012). Here $\widehat{\nabla} f, \nabla f = \mathbb{E}\left[\widehat{\nabla} f\right], \nabla^2 f$ denotes the stochastic gradient, gradient and the Hessian of the function f. With regards to least squares, we assume equation (3), following recent efforts Bach and Moulines (2013); Défossez and Bach (2015); Jain et al. (2016). While polynomially decaying stepsizes are nearly minimax optimal for general (strongly) convex functions, this paper indicates they are highly suboptimal on the final iterate for least squares. The geometrically decaying Step Decay schedule (Algorithm 1) provides marked improvements over any polynomial decay scheme on the final iterate for least squares. For simplicity of presentation, the results for least squares regression do not show dependence on initial error. See Theorems 1 and 2 for precise statements (and Nemirovsky and Yudin (1983); Shamir and Zhang (2012); Harvey et al. (2018) for precise statements of the general case).

This paper presents progress on answering the above question – refer to contributions below for more details. Note that the oracle model employed by this work (to quantify SGD's final iterate behavior) has featured in a string of recent results that present a non-asymptotic understanding of SGD for least squares regression, with the caveat being that these results crucially rely on *iterate averaging* with constant stepsize sequences (Bach and Moulines, 2013; Défossez and Bach, 2015; Jain et al., 2016, 2017b,a; Neu and Rosasco, 2018).

Our contributions: This work establishes upper and lower bounds on the behavior of SGD's final iterate, as run with polynomially decaying stepsizes as well as *step decay* schedules which tends to cut the stepsize by a constant factor after every constant number of epochs (see algorithm 1), by considering the streaming least squares regression problem (with and without strong convexity). Our main result indicates that step decay schedules offer significant improvements in achieving near minimax rates over polynomially decaying stepsizes in the known horizon case (when the end time T is known in advance). Figure 1 illustrates that this difference is evident (empirically) even when optimizing a two-dimensional synthetic least squares objective. Table 1 provides a summary. Finally, we present results that indicate the subtle (yet significant) differences between the known time horizon case and the anytime (i.e. the limiting) behavior of SGD's final iterate (see below). Note that proofs of our main claims can be found in the appendix.

Our main contributions are as follows:

- *Sub-optimality of polynomially decaying stepsizes:* For the strongly convex least squares case, this work shows that the final iterate of a polynomially decaying stepsize scheme (i.e. with $\eta_t \propto 1/t^\alpha$, with $\alpha \in [0.5, 1]$) is off the minimax rate $d\sigma^2/T$ by a factor of the *condition number* of the problem. For the non-strongly convex case of least squares, we show that *any* polynomially decaying stepsize can achieve a rate no better than $d\sigma^2/\sqrt{T}$ (up to $\log$ factors), while the minimax rate is $d\sigma^2/T$.

- *Near-optimality of the step-decay scheme:* Given a fixed end time T, the step-decay scheme (algorithm 1) presents a final iterate that is off the statistical minimax rate by just a $\log(T)$ factor when optimizing the strongly convex and non-strongly convex least squares regression [1], thus indicating vast improvements over polynomially decaying stepsize schedules. We note here that our Theorem 2 for the non-strongly case offers a rate on the initial error (i.e., the bias term) that is off the best known rate (Bach and Moulines, 2013) (that employs iterate averaging) by a dimension factor. That said, Algorithm 1 is rather straightforward and employs the knowledge of just an initial learning rate and number of iterations for its implementation.

- *SGD has to query bad iterates infinitely often:* For the case of optimizing strongly convex least squares regression, this work shows that any stochastic gradient procedure (in a $\limsup$ sense) *must* query sub-optimal iterates (off by nearly a condition number) infinitely often.

- Complementary to our theoretical results for the stochastic linear regression, we evaluate the empirical performance of different learning rate schemes when training a residual network on the cifar-10 dataset and observe that the continuous variant of step decay schemes (i.e. an exponential decay) indeed compares favorably to polynomially decaying stepsizes.

While the upper bounds established in this paper (section 3.2) merit extensions towards broader smooth convex functions (with/without strong convexity), the lower bounds established in sections 3.1, 3.3 present implications towards classes of smooth stochastic convex optimization. Even in terms of upper bounds, note that there are fewer results on non-asymptotic behavior of SGD (beyond least squares) when working in the oracle model considered in this work (see below). Bach and Moulines (2011, 2013); Bach (2014); Needell et al. (2016) are exceptions, yet they do not achieve minimax rates on appropriate problem classes; Frostig et al. (2015) does not work in standard stochastic first order oracle model (Nemirovsky and Yudin, 1983; Agarwal et al., 2012), so their work is not directly comparable to examine extensions towards broader function classes.

As a final note, this paper's result on the sub-optimality of standard polynomially decaying stepsizes for classes of smooth and strongly convex optimization doesn't contradict the (minimax) optimality results in stochastic approximation (Polyak and Juditsky, 1992). Iterate averaging coupled with polynomially decaying learning rates (or constant learning rates for least squares (Bach and Moulines, 2013; Défossez and Bach, 2015; Jain et al., 2016)) does achieve minimax rates (Ruppert, 1988; Polyak and Juditsky, 1992). However, as mentioned previously, this work deals with SGD's final iterate behavior (i.e. without iterate averaging), since this bears more relevance towards practice.

Related work: Robbins and Monro (1951) introduced the stochastic approximation problem and Stochastic Gradient Descent (SGD). They present conditions on stepsize schemes satisfied by asymptotically convergent algorithms: these schemes are referred to as "convergent" stepsize sequences. Ruppert (1988); Polyak and Juditsky (1992) proved the asymptotic optimality of iterate averaged SGD with larger stepsize sequences. In terms of oracle models and notions of optimality, there exists two lines of thought (see also Jain et al. (2017b)):

Towards statistically optimal estimation procedures: The goal of this line of thought is to match the excess risk of the statistically optimal estimator (Anbar, 1971; Kushner and Clark, 1978; Polyak and Juditsky, 1992; Lehmann and Casella, 1998) on every problem instance. Several efforts consider SGD in this oracle (Bach and Moulines, 2011; Bach, 2014; Dieuleveut and Bach, 2015; Frostig et al., 2015; Needell et al., 2016) presenting non-asymptotic results, often with iterate averaging. With regards to least squares, Bach and Moulines (2013); Défossez and Bach (2015); Frostig et al. (2015); Jain et al. (2016, 2017b); Neu and Rosasco (2018) use constant step-size SGD with iterate averaging to achieve minimax rates (on a per-problem basis) in this oracle model. SGD's final iterate behavior for least squares has featured in several efforts in the signal processing/controls literature (Widrow and Hoff, 1960; Nagumo and Noda, 1967; Proakis, 1974; Widrow and Stearns, 1985; Roy and Shynk, 1990; Sharma et al., 1998), without achieving minimax rates. This paper works in this oracle model and analyzes SGD's final iterate behavior with various stepsize choices.

Towards optimality under bounded noise assumptions: The other line of thought presents algorithms with access to stochastic gradients satisfying bounded noise assumptions, aiming to match lower bounds provided in Nemirovsky and Yudin (1983); Raginsky and Rakhlin (2011); Agarwal et al. (2012). Asymptotic properties of "convergent" stepsize

[1] This dependence can be improved to log of the condition number of the problem (for the strongly convex case) using a more refined stepsize decay scheme.

schemes have been studied in great detail (Kushner and Clark, 1978; Benveniste et al., 1990; Ljung et al., 1992; Bharath and Borkar, 1999; Kushner and Yin, 2003; Lai, 2003; Borkar, 2008). Dekel et al. (2012); Lacoste-Julien et al. (2012); Rakhlin et al. (2012); Ghadimi and Lan (2012, 2013b); Hazan and Kale (2014); Bubeck (2014); Dieuleveut et al. (2016) use iterate averaged SGD to achieve minimax rates for various problem classes non-asymptotically. Allen-Zhu (2018) present an alternative approach towards minimizing the gradient norm with access to stochastic gradients. As noted, Shamir and Zhang (2012) achieves anytime optimal rates (upto a $\log T$ factor) with the final iterate of an SGD procedure, and this is shown to be tight with the recent work of Harvey et al. (2018). Jain et al. (2019) achieve minimax rates on the final iterate using a nuanced stepsize scheme when the number of iterations is fixed in advance.

Geometrically Decaying Stepsize Schedules date to Goffin (1977). Davis and Drusvyatskiy (2019) employ the stepdecay schedule to prove high-probability guarantees for SGD with strongly convex objectives. In stochastic optimization, several other works, including Ghadimi and Lan (2013a); Hazan and Kale (2014); Aybat et al. (2019); Kulunchakov and Mairal (2019) consider doubling argument based approaches, where the epoch length is doubled everytime the stepsizes are halved. The step decay schedule is employed to yield faster rates of convergence under certain growth (and related) conditions both in convex (Xu et al., 2016) and non-convex settings (Yang et al., 2018; Davis et al., 2019).

Paper organization: Section 2 describes notation and problem setup. Section 3 presents our results on the suboptimality of polynomial decay schemes and the near optimality of the step decay scheme. Section 3.3 presents results on the anytime behavior of SGD (i.e. the asymptotic/infinite horizon case). Section 4 presents experimental results and Section 5 presents conclusions.

2 Problem Setup

Notation: We present the setup and associated notation in this section. We represent scalars with normal font a, b, L etc., vectors with boldface lowercase characters $\mathbf{a}, \mathbf{b}$ etc. and matrices with boldface uppercase characters $\mathbf{A}, \mathbf{B}$ etc. We represent positive semidefinite (PSD) ordering between two matrices using $\succeq$. The symbol $\gtrsim$ represents that the inequality holds for some universal constant.

We consider here the minimization of the following expected square loss objective:

$$\min_{\mathbf{w}} f(\mathbf{w}) \text{ where } f(\mathbf{w}) \stackrel{\text{def}}{=} \tfrac{1}{2}\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[(y - \langle \mathbf{w}, \mathbf{x}\rangle)^2]. \tag{1}$$

Note that the hessian of the objective $\mathbf{H} \stackrel{\text{def}}{=} \nabla^2 f(\mathbf{w}) = \mathbb{E}\left[\mathbf{x}\mathbf{x}^\top\right]$. We are provided access to stochastic gradients obtained by sampling a new example $(\mathbf{x}_t, y_t) \sim \mathcal{D}$. These examples satisfy:

$$y = \langle \mathbf{w}^*, \mathbf{x}\rangle + \epsilon,$$

where, ϵ is the noise on the example pair $(\mathbf{x}, y) \sim \mathcal{D}$ and $\mathbf{w}^*$ is a minimizer of the objective $f(\mathbf{w})$. Given an initial iterate $\mathbf{w}_0$ and stepsize sequence $\{\eta_t\}$, our stochastic gradient update is:

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \eta_t \widehat{\nabla} f(\mathbf{w}_{t-1}); \quad \widehat{\nabla} f(\mathbf{w}_t) = -(y_t - \langle \mathbf{w}_t, \mathbf{x}_t\rangle) \cdot \mathbf{x}_t. \tag{2}$$

We assume that the noise $\epsilon = y - \langle \mathbf{w}^*, \mathbf{x}\rangle \ \forall \ (\mathbf{x}, y) \sim \mathcal{D}$ satisfies the following condition:

$$\Sigma \stackrel{\text{def}}{=} \mathbb{E}\left[\widehat{\nabla} f(\mathbf{w}^*)\widehat{\nabla} f(\mathbf{w}^*)^\top\right] = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[(y - \langle \mathbf{w}^*, \mathbf{x}\rangle)^2 \mathbf{x}\mathbf{x}^\top] \preceq \sigma^2 \mathbf{H}. \tag{3}$$

Next, assume that covariates $\mathbf{x}$ satisfy the following fourth moment inequality:

$$\mathbb{E}\left[\|\mathbf{x}\|^2 \mathbf{x}\mathbf{x}^\top\right] \preceq R^2 \, \mathbf{H} \tag{4}$$

This assumption is satisfied, say, when the norm of the covariates $\sup \|\mathbf{x}\|^2 < R^2$, but is true more generally. Finally, note that both 3 and 4 are general and are used in recent works (Bach and Moulines, 2013; Jain et al., 2016) that present a sharp analysis of SGD for streaming least squares problem. Next, we denote by

$$\mu \stackrel{\text{def}}{=} \lambda_{\min}(\mathbf{H}), \quad L \stackrel{\text{def}}{=} \lambda_{\max}(\mathbf{H}), \text{ and }, \kappa \stackrel{\text{def}}{=} R^2/\mu$$

the smallest eigenvalue, largest eigenvalue and condition number of $\mathbf{H}$ respectively. $\mu > 0$ in the strongly convex case but not necessarily so in the non-strongly convex case (in section 3 and beyond, the non-strongly case is referred to as the "smooth" case). Let $\mathbf{w}^* \in \arg\min_{\mathbf{w}\in\mathbb{R}^d} f(\mathbf{w})$. The excess risk of an estimator $\mathbf{w}$ is $f(\mathbf{w}) - f(\mathbf{w}^*)$. Given t accesses to the stochastic gradient oracle in equation 2, any algorithm that uses these stochastic gradients and outputs $\widehat{\mathbf{w}}_t$ has sub-optimality that is lower bounded by $\frac{\sigma^2 d}{t}$. More concretely, we have that (Van der Vaart, 2000)

$$\lim_{t\to\infty} \frac{\mathbb{E}\left[f(\widehat{\mathbf{w}}_t)\right] - f(\mathbf{w}^*)}{\sigma^2 d/t} \geq 1\,.$$

The rate of $(1 + o(1)) \cdot \sigma^2 d/t$ is achieved using iterate averaged SGD (Ruppert, 1988; Polyak and Juditsky, 1992) with constant stepsizes (Bach and Moulines, 2013; Défossez and Bach, 2015; Jain et al., 2016). This rate of $\sigma^2 d/t$ is called the statistical minimax rate.

3 Main results

Sections 3.1, 3.2 consider the fixed time horizon setting; the former presents the significant sub-optimality of polynomially decaying stepsizes on SGD's final iterate behavior, the latter section presenting the near-optimality of SGD's final iterate. Section 3.3 presents negative results on SGD's final iterate behavior (irrespective of stepsizes employed), in the anytime (i.e. limiting) sense.

3.1 Suboptimality of polynomial decay schemes

This section begins by showing that there exist problem instances where polynomially decaying stepsizes considered stochastic approximation theory (Robbins and Monro, 1951; Polyak and Juditsky, 1992) i.e., those of the form $\frac{a}{b+t^\alpha}$, for any choice of $a, b > 0$ and $\alpha \in [0.5, 1]$ are significantly suboptimal (by a factor of the condition number of the problem, or by $\sqrt{T}$ in the smooth case) compared to the statistical minimax rate (Kushner and Clark, 1978).

Theorem 1. *Under assumptions 3, 4, there exists a class of problem instances where the following lower bounds on excess risk hold on SGD's final iterate with polynomially decaying stepsizes when given access to the oracle as written in equation 2.*
***Strongly convex case**: Suppose $\mu > 0$. For any condition number κ, there exists a least squares problem instance with initial suboptimality $f(\mathbf{w}_0) - f(\mathbf{w}^*) \leq \sigma^2 d$ such that, for any $T \geq \kappa^{\frac{4}{3}}$, and for all $a, b \geq 0$ and $0.5 \leq \alpha \leq 1$, and for the learning rate scheme $\eta_t = \frac{a}{b+t^\alpha}$, we have*

$$\mathbb{E}\left[f(\mathbf{w}_T)\right] - f(\mathbf{w}^*) \geq \exp\left(-\frac{T}{\kappa \log T}\right)\left(f(\mathbf{w}_0) - f(\mathbf{w}^*)\right) + \frac{\sigma^2 d}{64} \cdot \frac{\kappa}{T}.$$

***Smooth case**: For any fixed $T > 1$, there exists a least squares problem instance such that, for all $a, b \geq 0$ and $0.5 \leq \alpha \leq 1$, and for the learning rate scheme $\eta_t = \frac{a}{b+t^\alpha}$, we have*

$$\mathbb{E}\left[f(\mathbf{w}_T)\right] - f(\mathbf{w}^*) \geq \left(L \cdot \|\mathbf{w}_0 - \mathbf{w}^*\|^2 + \sigma^2 d\right) \cdot \frac{1}{\sqrt{T}\log T}.$$

For both cases (with/without strong convexity), the minimax rate is $\sigma^2 d/T$. In the strongly convex case, SGD's final iterate with polynomially decaying stepsizes pays a suboptimality factor of $\Omega(\kappa)$, whereas, in the smooth case, SGD's final iterate pays a suboptimality factor of $\Omega\left(\frac{\sqrt{T}}{\log T}\right)$.

3.2 Near optimality of Step Decay schemes

Given the knowledge of an end time T when the algorithm is terminated, this section presents the step decay schedule (Algorithm 1), which offers significant improvements over standard polynomially decaying stepsize schemes, and obtains near minimax rates (off by only a $\log(T)$ factor).

Theorem 2. *Suppose we are given access to the stochastic gradient oracle 2 satisfying Assumptions 3 and 4. Running Algorithm 1 with an initial stepsize of $\eta_1 = 1/(2R^2)$ allows the algorithm to achieve the following excess risk guarantees.*

- ***Strongly convex case**: Suppose $\mu > 0$. We have:*

$$\mathbb{E}\left[f(\mathbf{w}_T)\right] - f(\mathbf{w}^*) \leq 2 \cdot \exp\left(-\frac{T}{2\kappa \log T \log \kappa}\right)\left(f(\mathbf{w}_0) - f(\mathbf{w}^*)\right) + 4\sigma^2 d \cdot \frac{\log T}{T}\,.$$

- ***Smooth case**: We have:*

$$\mathbb{E}\left[f(\mathbf{w}_T)\right] - f(\mathbf{w}^*) \leq 2 \cdot \left(R^2 d \cdot \|\mathbf{w}_0 - \mathbf{w}^*\|^2 + 2\sigma^2 d\right) \cdot \frac{\log T}{T}$$

While theorem 2 presents significant improvements over polynomial decay schemes, as mentioned in the contributions, the above result presents a worse rate on the initial error (by a dimension factor) in the smooth case (i.e. non-strongly convex case), compared to the best known result (Bach and Moulines, 2013), which relies heavily on iterate averaging to remove this factor. It is an open question with regards to whether this factor can actually be improved or not. Furthermore, comparing the initial error dependence between the lower bound for the smooth case (Theorem 1) with the upper bound for the step decay scheme, we believe that the dependence on the smoothness L should be improved to one on the R^2.

In terms of the variance, however, note that the polynomial decay schemes, are plagued by a polynomial dependence on the condition number κ (for the strongly convex case), and are off the minimax rate by a $\sqrt{T}$ factor (for the smooth case). The step decay schedule, on the other hand, is off the minimax rate (Ruppert, 1988; Polyak and Juditsky, 1992; Van der Vaart, 2000) by only a $\log(T)$ factor. It is worth noting that Algorithm 1 admits an efficient implementation in that it requires the knowledge only of R^2 (similar to iterate averaging results (Bach and Moulines, 2013; Jain et al., 2016)) and the end time T. Finally, note that this $\log T$ factor can be improved to a $\log \kappa$ factor for the strongly convex case by using an additional polynomial decay scheme before switching to the Step Decay scheme.

Proposition 3. *Suppose we have access to the stochastic gradient oracle 2 satisfying Assumptions 3 and 4. Let $\mu > 0$ and $\kappa \geq 2$. For any problem and fixed time horizon $T/\log T > 5\kappa$, there exists a learning rate scheme that achieves*

$$\mathbb{E}\left[f(\mathbf{w}_T)\right] - f(\mathbf{w}^*) \leq 2\exp(-T/(6\kappa \log \kappa)) \cdot \left(f(\mathbf{w}_0) - f(\mathbf{w}^*)\right) + 100 \log_2 \kappa \cdot \frac{\sigma^2 d}{T}.$$

In order to have improved the dependence on the variance from $\log(T)$ (in theorem 2) to $\log(\kappa)$ (in proposition 3), we require access to the strong convexity parameter $\mu = \lambda_{\min}(\mathbf{H})$ in addition to R^2 and knowledge of the end time T. This parallels results known for general strongly convex setting (Rakhlin et al., 2012; Lacoste-Julien et al., 2012; Shamir and Zhang, 2012; Bubeck, 2014; Jain et al., 2019).

As a final remark, note that this section's results (on step decay schemes) assumed the knowledge of a fixed time horizon T. In contrast, most results SGD's averaged iterate obtain anytime (i.e., limiting/infinite horizon) guarantees. Can we hope to achieve such guarantees with the final iterate?

3.3 SGD queries bad points infinitely often

This section shows that obtaining near minimax rates with the *final* iterate is not possible without knowledge of the time horizon T. Concretely, we show that irrespective of the learning rates employed (be it polynomially decaying or step-decay), SGD *requires* to query a point with sub-optimality $\Omega(\kappa/\log\kappa) \cdot \sigma^2 d/T$ for infinitely many time steps T.

Theorem 4. *Suppose we have access to a stochastic gradient oracle 2 satisfying Assumption 3, 4. There exists a universal constant $C > 0$, and a problem instance such that SGD algorithm with any $\eta_t \leq 1/2R^2$ for all t[2], we have*

$$\limsup_{T\to\infty} \frac{\mathbb{E}\left[f(\mathbf{w}_T)\right] - f(\mathbf{w}^*)}{(\sigma^2 d/T)} \geq C\frac{\kappa}{\log(\kappa+1)}\,.$$

The bad points guaranteed to exist by Theorem 4 are not rare. We show that such points occur at least once in $\mathcal{O}\left(\frac{\kappa}{\log\kappa}\right)$ iterations. Refer to Theorem 16 in appendix D in the appendix.

[2]Learning rate more than $2/R^2$ will make the algorithm diverge.

4 Experimental Results

We present experimental validation on the suitability of the Step-decay schedule (or more precisely, its continuous counterpart, which is the exponentially decaying schedule), and compare its with the polynomially decaying stepsize schedules. In particular, we consider the use of:

$$\eta_t = \frac{\eta_0}{1 + b \cdot t} \qquad (5) \qquad\qquad \eta_t = \frac{\eta_0}{1 + b\sqrt{t}} \qquad (6) \qquad\qquad \eta_t = \eta_0 \cdot \exp\left(-b \cdot t\right). \qquad (7)$$

Where, we perform a systematic grid search on the parameters η_0 and b. In the section below, we consider a real world non-convex optimization problem of training a residual network on the cifar-10 dataset, with an aim to illustrate the practical implications of the results described in the paper. Refer to Appendix E for more details.

4.1 Non-Convex Optimization: Training a Residual Net on cifar-10

We consider training a $44-$layer deep residual network (He et al., 2016b) with pre-activation blocks (He et al., 2016a) (dubbed preresnet-44) on cifar-10 dataset. The code for implementing the network can be found here [3]. For all experiments, we use Nesterov's momentum (Nesterov, 1983) implemented in pytorch [4] with a momentum of 0.9, batchsize 128, 100 training epochs, ℓ_2 regularization of 0.0005.

Our experiments are based on grid searching for the best learning rate decay scheme on the parametric family of learning rate schemes described above (5),(6),(7); all grid searches are performed on a separate validation set (obtained by setting aside one-tenth of the training dataset) and with models trained on the remaining 45000 samples. For presenting the final numbers in the plots/tables, we employ the best hyperparameters from the validation stage and train it on the entire $50,000$ samples and average results run with 10 different random seeds. The parameters for grid searches and other details are presented in Appendix E. Furthermore, we always extend the grid so that the best performing grid search parameter lies in the interior of our grid search.

How does the step decay scheme compare with the polynomially decaying stepsizes? Figure 2 and Table 2 present a comparison of the performance of the three schemes (5)-(7). These results demonstrate that the exponential scheme convicingly outperforms the polynomial step-size schemes.

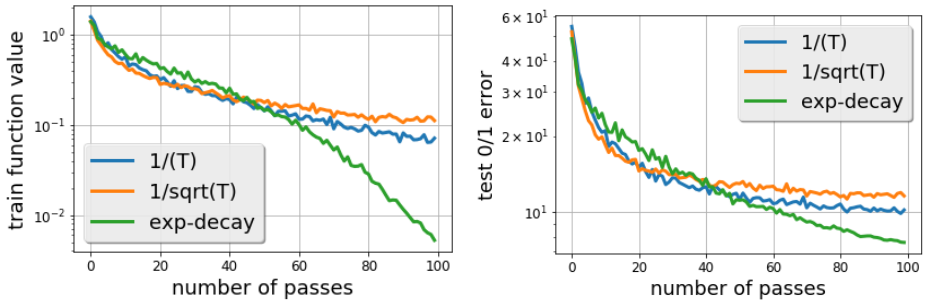

Figure 2: Plot of the training function value (left) and test $0/1-$ error (right) comparing the three decay schemes (two polynomial) 5, 6, (and one exponential) 7 scheme.

Decay Scheme	Train Function Value	Test $0/1$ error
$O(1/t)$ (equation (5))	0.0713 ± 0.015	$10.20 \pm 0.7\%$
$O(1/\sqrt{t})$ (equation (6))	0.1119 ± 0.036	$11.6 \pm 0.67\%$
$\exp(-t)$ (equation (7))	$\mathbf{0.0053 \pm 0.0015}$	$\mathbf{7.58 \pm 0.21\%}$

Table 2: Comparing Train Cross-Entropy and Test $0/1$ Error of various learning rate decay schemes for the classification task on cifar-10 using a $44-$layer residual net with pre-activations.

[3] https://github.com/D-X-Y/ResNeXt-DenseNet
[4] https://github.com/pytorch

Does suffix iterate averaging improve over final iterate's behavior for polynomially decaying stepsizes? Towards answering this question, firstly, we consider the best performing values of equation 5 and 6, and then, average iterates of the algorithm starting from $5, 10, 20, 40, 80, 85, 90, 95, 99$ epochs when training the model for a total of 100 epochs. While such iterate averaging (and their suffix) variants have strong theoretical support for (stochastic) convex optimization (Ruppert, 1988; Polyak and Juditsky, 1992; Rakhlin et al., 2012; Bubeck, 2014; Jain et al., 2016), their impact on non-convex optimization is largely debatable. Nevertheless, this experiments's results (figure 3) indicates that suffix averaging tends to hurt the algorithm's generalization behavior (which is unsurprising given the non-convex nature of the objective). Note that, figure 3 serves to indicate that averaging the final few (≤ 5) epochs tends to offer nearly the same result as the final iterate's behavior, indicating that the gains of using suffix iterate averaging are relatively limited for several such settings.

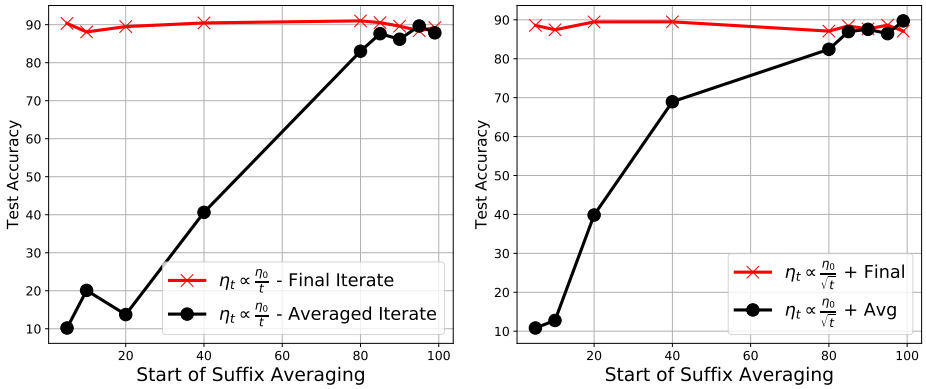

Figure 3: Performance of the suffix averaged iterate compared to the final iterate when varying the iteration when iterate averaging is begun from $\{5, 10, 20, 40, 80, 85, 90, 95, 99\}$ epochs for the $1/T$ learning rate 5 (left) and the $1/\sqrt{T}$ learning rate 6 (right).

Does our result on "knowing" the time horizon (for step-decay schedule) present implications towards hyper-parameter search methods that work based on results from truncated runs? Towards answering this question, consider the figure 4 and Tables 3 and 4, which present a comparison of the performance of three exponential decay schemes each of which is tuned to achieve the best performance at 33, 66 and 100 epochs respectively. The key point to note is that best performing hyperparameters at 33 and 66 epochs are not the best performing at 100 epochs (which is made stark from the perspective of the validation error - refer to table 4). This demonstrates that hyper parameter selection methods that tend to discard hyper-parameters which don't perform well at earlier stages of the optimization (i.e. based on comparing results on truncated runs), which, for e.g., is indeed the case with hyperband (Li et al., 2017), will benefit from a round of rethinking.

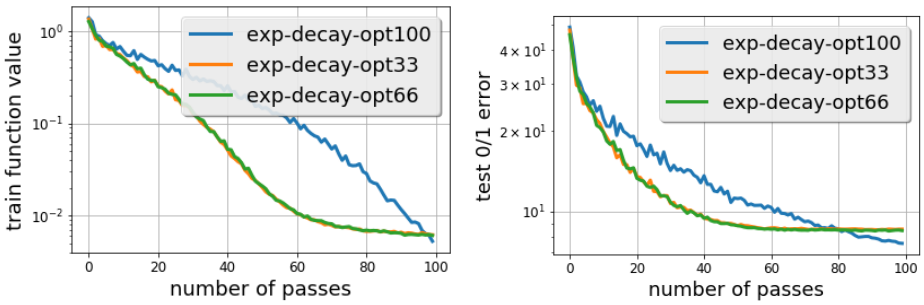

Figure 4: Plot of the training function value (left) and test $0/1-$ error (right) comparing exponential decay scheme (equation 7), with parameters optimized for 33, 66 and 100 epochs.

Decay Scheme	Train FVal @33	Train FVal @66	Train FVal @100
$\exp(-t)$ [optimized for 33 epochs] (eqn (7))	$\mathbf{0.098 \pm 0.006}$	$\mathbf{0.0086 \pm 0.002}$	$\mathbf{0.0062 \pm 0.0015}$
$\exp(-t)$ [optimized for 66 epochs] (eqn (7))	$\mathbf{0.107 \pm 0.012}$	$\mathbf{0.0088 \pm 0.0014}$	$\mathbf{0.0061 \pm 0.0011}$
$\exp(-t)$ [optimized for 100 epochs] (eqn (7))	0.3 ± 0.06	0.071 ± 0.017	$\mathbf{0.0053 \pm 0.0016}$

Table 3: Comparing training (softmax) function value by optimizing the exponential decay scheme with end times of $33/66/100$ epochs on cifar-10 dataset using a $44-$layer residual net.

Decay Scheme	Test $0/1$ @33	Test $0/1$ @66	Test $0/1$ @100
$\exp(-t)$ [optimized for 33 epochs] (eqn (7))	$\mathbf{10.36 \pm 0.235}\%$	$\mathbf{8.6 \pm 0.26}\%$	$8.57 \pm 0.25\%$
$\exp(-t)$ [optimized for 66 epochs] (eqn (7))	$\mathbf{10.51 \pm 0.45}\%$	$\mathbf{8.51 \pm 0.13}\%$	$8.46 \pm 0.19\%$
$\exp(-t)$ [optimized for 100 epochs] (eqn (7))	$14.42 \pm 1.47\%$	$9.8 \pm 0.66\%$	$\mathbf{7.58 \pm 0.21}\%$

Table 4: Comparing test $0/1$ error by optimizing the exponential decay scheme with end times of $33/66/100$ epochs for the classification task on cifar-10 dataset using a $44-$layer residual net.

5 Conclusions and Discussion

The main contribution of this work shows that the behavior of SGD's final iterate for least squares regression is much more nuanced than what has been indicated by prior efforts that have primarily considered non-smooth stochastic convex optimization. The results of this paper point out the striking limitations of polynomially decaying stepsizes on SGD's final iterate, as well as sheds light on the effectiveness of starkly different schemes based on a Step Decay schedule. Somewhat coincidentally, practical implementations for certain classes of stochastic optimization do return the final iterate of SGD with step decay schedule $-$ this connection does merit an understanding through future work.

Acknowledgments: Rong Ge acknowledges funding from NSF CCF-1704656, NSF CCF-1845171 (CAREER), Sloan Fellowship and Google Faculty Research Award. Sham Kakade acknowledges funding from the Washington Research Foundation for Innovation in Data-intensive Discovery, NSF Award 1740551, and ONR award N00014-18-1-2247. Rahul Kidambi acknowledges funding from NSF Award 1740822.

References

A. Agarwal, P. L. Bartlett, P. Ravikumar, and M. J. Wainwright. Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization. *IEEE Transactions on Information Theory*, 2012.

Z. Allen-Zhu. How to make the gradients small stochastically. *CoRR*, abs/1801.02982, 2018.

D. Anbar. *On Optimal Estimation Methods Using Stochastic Approximation Procedures*. University of California, 1971. URL http://books.google.com/books?id=MmpHJwAACAAJ.

N. S. Aybat, A. Fallah, M. Gürbüzbalaban, and A. E. Ozdaglar. A universally optimal multistage accelerated stochastic gradient method. *CoRR*, abs/1901.08022, 2019.

F. R. Bach. Adaptivity of averaged stochastic gradient descent to local strong convexity for logistic regression. *Journal of Machine Learning Research (JMLR)*, volume 15, 2014.

F. R. Bach and E. Moulines. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In *NIPS 24*, 2011.

F. R. Bach and E. Moulines. Non-strongly-convex smooth stochastic approximation with convergence rate O(1/n). In *NIPS 26*, 2013.

A. Benveniste, M. Metivier, and P. Priouret. *Adaptive Algorithms and Stochastic Approximations*. Springer texts in Stochastic Modelling and Applied Probability, 1990.

B. Bharath and V. S. Borkar. Stochastic approximation algorithms: overview and recent trends. *Sādhanā*, 1999.

V. Borkar. *Stochastic approximation*. Cambridge Books, 2008.

L. Bottou and O. Bousquet. The tradeoffs of large scale learning. In *NIPS 20*, 2007.

S. Bubeck. Theory of convex optimization for machine learning. *CoRR*, abs/1405.4980, 2014.

D. Davis and D. Drusvyatskiy. Robust stochastic optimization with the proximal point method. *CoRR*, abs/1907.13307, 2019.

D. Davis, D. Drusvyatskiy, and V. Charisopoulos. Stochastic algorithms with geometric step decay converge linearly on sharp functions. *CoRR*, abs/1907.09547, 2019.

A. Défossez and F. R. Bach. Averaged least-mean-squares: Bias-variance trade-offs and optimal sampling distributions. In *Artifical Intelligence and Statistics (AISTATS)*, 2015.

O. Dekel, R. Gilad-Bachrach, O. Shamir, and L. Xiao. Optimal distributed online prediction using mini-batches. *Journal of Machine Learning Research (JMLR)*, volume 13, 2012.

A. Dieuleveut and F. R. Bach. Non-parametric stochastic approximation with large step sizes. *The Annals of Statistics*, 2015.

A. Dieuleveut, N. Flammarion, and F. R. Bach. Harder, better, faster, stronger convergence rates for least-squares regression. *CoRR*, abs/1602.05419, 2016.

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

R. Frostig, R. Ge, S. M. Kakade, and A. Sidford. Competing with the empirical risk minimizer in a single pass. In *COLT*, 2015.

S. Ghadimi and G. Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization i: A generic algorithmic framework. *SIAM Journal on Optimization*, 2012.

S. Ghadimi and G. Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization, ii: Shrinking procedures and optimal algorithms. *SIAM Journal on Optimization*, 23(4), 2013a.

S. Ghadimi and G. Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization, ii: shrinking procedures and optimal algorithms. *SIAM Journal on Optimization*, 2013b.

J. L. Goffin. On the convergence rates of subgradient optimization methods. *Mathematical Programming*, 13:329–347, 1977.

N. J. A. Harvey, C. Liaw, Y. Plan, and S. Randhawa. Tight analyses for non-smooth stochastic gradient descent. *CoRR*, 2018. URL http://arxiv.org/abs/1812.05217.

E. Hazan and S. Kale. Beyond the regret minimization barrier: optimal algorithms for stochastic strongly-convex optimization. *Journal of Machine Learning Research (JMLR)*, volume 15, 2014.

K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *ECCV (4)*, Lecture Notes in Computer Science, pages 630–645. Springer, 2016a.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016b.

P. Jain, S. M. Kakade, R. Kidambi, P. Netrapalli, and A. Sidford. Parallelizing stochastic approximation through mini-batching and tail-averaging. *arXiv preprint arXiv:1610.03774*, 2016.

P. Jain, S. M. Kakade, R. Kidambi, P. Netrapalli, V. K. Pillutla, and A. Sidford. A markov chain theory approach to characterizing the minimax optimality of stochastic gradient descent (for least squares). *CoRR*, 2017a. URL http://arxiv.org/abs/1710.09430.

P. Jain, S. M. Kakade, R. Kidambi, P. Netrapalli, and A. Sidford. Accelerating stochastic gradient descent. *arXiv preprint arXiv:1704.08227*, 2017b.

P. Jain, D. Nagaraj, and P. Netrapalli. Making the last iterate of sgd information theoretically optimal. *CoRR*, 2019. URL http://arxiv.org/abs/1904.12443.

R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *NIPS 26*, 2013.

A. Kulunchakov and J. Mairal. A generic acceleration framework for stochastic composite optimization. *CoRR*, abs/1906.01164, 2019.

H. J. Kushner and D. S. Clark. *Stochastic Approximation Methods for Constrained and Unconstrained Systems.* Springer-Verlag, 1978.

H. J. Kushner and G. Yin. Stochastic approximation and recursive algorithms and applications. *Springer-Verlag*, 2003.

S. Lacoste-Julien, M. W. Schmidt, and F. R. Bach. A simpler approach to obtaining an o(1/t) convergence rate for the projected stochastic subgradient method. *CoRR*, 2012. URL http://arxiv.org/abs/1212.2002.

T. L. Lai. Stochastic approximation: invited paper, 2003.

E. L. Lehmann and G. Casella. *Theory of Point Estimation*. Springer Texts in Statistics. Springer, 1998. ISBN 9780387985022.

L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *The Journal of Machine Learning Research*, 18(1):6765–6816, 2017.

L. Ljung, G. Pflug, and H. Walk. *Stochastic Approximation and Optimization of Random Systems*. Birkhauser Verlag, Basel, Switzerland, Switzerland, 1992. ISBN 3-7643-2733-2.

J.-I. Nagumo and A. Noda. A learning method for system identification. *IEEE Transactions on Automatic Control*, 1967.

D. Needell, N. Srebro, and R. Ward. Stochastic gradient descent, weighted sampling, and the randomized kaczmarz algorithm. *Mathematical Programming*, 2016.

A. S. Nemirovsky and D. B. Yudin. *Problem Complexity and Method Efficiency in Optimization*. John Wiley, 1983.

Y. E. Nesterov. A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$. *Doklady AN SSSR*, 269, 1983.

G. Neu and L. Rosasco. Iterate averaging as regularization for stochastic gradient descent. *CoRR*, 2018. URL http://arxiv.org/abs/1802.08009.

B. T. Polyak and A. B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, volume 30, 1992.

J. G. Proakis. Channel identification for high speed digital communications. *IEEE Transactions on Automatic Control*, 1974.

M. Raginsky and A. Rakhlin. Information-based complexity, feedback and dynamics in convex programming. *IEEE Transactions on Information Theory*, 2011.

A. Rakhlin, O. Shamir, and K. Sridharan. Making gradient descent optimal for strongly convex stochastic optimization. In *ICML*, 2012.

H. Robbins and S. Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, vol. 22, 1951.

S. Roy and J. J. Shynk. Analysis of the momentum lms algorithm. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 1990.

D. Ruppert. Efficient estimations from a slowly convergent robbins-monro process. *Tech. Report, ORIE, Cornell University*, 1988.

O. Shamir. Open problem: Is averaging needed for strongly convex stochastic gradient descent? In *COLT*, 2012.

O. Shamir and T. Zhang. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. *CoRR*, abs/1212.1824, 2012.

R. Sharma, W. A. Sethares, and J. A. Bucklew. Analysis of momentum adaptive filtering algorithms. *IEEE Transactions on Signal Processing*, 1998.

A. W. Van der Vaart. *Asymptotic statistics*, volume 3. Cambridge university press, 2000.

B. Widrow and M. E. Hoff. *Adaptive switching circuits*. Defense Technical Information Center, 1960.

B. Widrow and S. D. Stearns. *Adaptive Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1985.

Y. Xu, Q. Lin, and T. Yang. Accelerate stochastic subgradient method by leveraging local error bound. *CoRR*, abs/1607.01027, 2016.

T. Yang, Y. Y. 0006, Z. Yuan, and R. Jin. Why does stagewise training accelerate convergence of testing error over sgd? *CoRR*, abs/1812.03934, 2018.

A Preliminaries

Before presenting the lemmas establishing the behavior of SGD under various learning rate schemes, we introduce some notation. We recount that the SGD update rule denoted through:

$$\mathbf{w}_t = \mathbf{w}_{t-1} - \eta_t \widehat{\nabla f}(\mathbf{w}_{t-1})$$

We then write out the expression for the stochastic gradient $\widehat{\nabla f}(\mathbf{w}_{t-1})$.

$$\widehat{\nabla f}(\mathbf{w}_{t-1}) = \mathbf{x}_t \mathbf{x}_t^\top (\mathbf{w}_{t-1} - \mathbf{w}^*) - \epsilon_t \mathbf{x}_t,$$

where, given the stochastic gradient corresponding to an example $(\mathbf{x}_t, y_t) \sim \mathcal{D}$, with $y_t = \langle \mathbf{w}^*, \mathbf{x}_t \rangle + \epsilon_t$, the above stochastic gradient expression naturally follows. Now, in order to analyze the contraction properties of the SGD update rule, we require the following notation:

$$P_t = \mathbf{I} - \eta_t \mathbf{x}_t \mathbf{x}_t^\top.$$

Lemma 5. *[For e.g. Appendix A.2.2 from Jain et al. (2016)]* **Bias-Variance tradeoff:** *Running SGD for $T-$steps starting from $\mathbf{w}_0$ and a stepsize sequence $\{\eta_t\}_{t=1}^T$ presents a final iterate $\mathbf{w}_T$ whose excess risk is upper-bounded as:*

$$\langle \mathbf{H}, \mathbb{E}\left[(\mathbf{w}_T - \mathbf{w}^*) \otimes (\mathbf{w}_T - \mathbf{w}^*)\right] \rangle \leq 2 \cdot \Bigg(\langle \mathbf{H}, \mathbb{E}\left[P_T \cdots P_1 (\mathbf{w}_0 - \mathbf{w}^*) \otimes (\mathbf{w}_0 - \mathbf{w}^*) P_1 \cdots P_T\right] \rangle + \left\langle \mathbf{H}, \sum_{\tau=1}^T \eta_\tau^2 \cdot \mathbb{E}\left[P_T \cdots P_{\tau+1} n_\tau \otimes n_\tau P_{\tau+1} \cdots P_T\right] \right\rangle \Bigg),$$

where, $P_t = \mathbf{I} - \eta_t \cdot \mathbf{x}_t \mathbf{x}_t^\top$ and $n_t = \epsilon_t x_t$. Note that $\mathbb{E}\left[n_t | \mathcal{F}_{t-1}\right] = 0$ and $\mathbb{E}\left[n_t \otimes n_t | \mathcal{F}_{t-1}\right] \preceq \sigma^2 \mathbf{H}$, where, $\mathcal{F}_{t-1}$ is the filtration formed by all samples $(\mathbf{x}_1, y_1) \cdots (\mathbf{x}_{t-1}, y_{t-1})$ until time t.

Proof. One can view the contribution of the above two terms as ones stemming from SGD's updates, which can be written as:

$$\begin{aligned}
\mathbf{w}_t &= \mathbf{w}_{t-1} - \eta_t \widehat{\nabla f}(w_{t-1}) \\
\mathbf{w}_t - \mathbf{w}^* &= (\mathbf{I} - \eta_t \mathbf{x}_t \mathbf{x}_t)(\mathbf{w}_{t-1} - \mathbf{w}^*) + \eta_t n_t \\
\mathbf{w}_t - \mathbf{w}^* &= P_t \cdots P_1 (\mathbf{w}_0 - \mathbf{w}^*) + \sum_{\tau=1}^T P_t \cdots P_{\tau+1} \eta_\tau n_\tau
\end{aligned}$$

From the above equation, the result of the lemma follows straightforwardly. Now, clearly, if the noise ϵ and the inputs $\mathbf{x}$ are indepdent of each other, and if the noise is zero mean i.e. $\mathbb{E}\left[\epsilon\right] = 0$, the above inequality holds with equality (without the factor of two). This is true more generally iff

$$\mathbb{E}\left[\epsilon \mathbf{x}^{(i)} \mathbf{x}^{(j)} \mathbf{x}^{(k)}\right] = 0.$$

For more details, refer to Défossez and Bach (2015).

Now, in order to bound the total error, note that the original stochastic process associated with SGD's updates can be decoupled into two (simpler) processes, one being the noiseless process (which corresponds to reducing the dependence on the initial error, and is termed "bias"), i.e., where, the recurrence evolves as:

$$\mathbf{w}_t^{\text{bias}} - \mathbf{w}^* = P_t (\mathbf{w}_{t-1}^{\text{bias}} - \mathbf{w}^*) \tag{8}$$

The second recursion corresponds to the dependence on the noise (termed as variance), wherein, the process is initiated at the solution, i.e. $\mathbf{w}_0^{\text{var}} = \mathbf{w}^*$ and is driven by the noise n_t. The update for this process corresponds to:

$$\mathbf{w}_t^{\text{var}} - \mathbf{w}^* = P_t (\mathbf{w}_{t-1}^{\text{var}} - \mathbf{w}^*) + \eta_t n_t, \quad \text{with} \quad \mathbf{w}_0^{\text{var}} = \mathbf{w}^* \tag{9}$$

$$= \sum_{\tau=1}^{t} P_t \cdots P_{\tau+1} \cdot (\eta_\tau n_\tau).$$

We represent by B_t the covariance of the t^{th} iterate of the bias process, i.e.,

$$\begin{aligned} B_t &= \mathbb{E}\left[\left(\mathbf{w}_t^{\text{bias}} - \mathbf{w}^*\right)\left(\mathbf{w}_t^{\text{bias}} - \mathbf{w}^*\right)^\top\right] \\ &= \mathbb{E}\left[P_t B_{t-1} P_t^\top\right] = \mathbb{E}\left[P_t \cdots P_1 B_0 P_1 \cdots P_t\right] \end{aligned}$$

The quantity that routinely shows up when bounding SGD's convergence behavior is the covariance of the variance error, i.e. $V_t := \mathbb{E}\left[(\mathbf{w}_t^{\text{var}} - \mathbf{w}^*) \otimes \mathbf{w}_t^{\text{var}} - \mathbf{w}^*)\right]$. This implies the following (simplified) expression for V_t:

$$\begin{aligned} V_t &= \mathbb{E}\left[(\mathbf{w}_t^{\text{var}} - \mathbf{w}^*) \otimes (\mathbf{w}_t^{\text{var}} - \mathbf{w}^*)\right] \\ &= \mathbb{E}\left[\left(\sum_{\tau=1}^{t} P_t \cdots P_{\tau+1} \cdot (\eta_\tau n_\tau)\right) \otimes \left(\sum_{\tau'=1}^{t} P_t \cdots P_{\tau'+1} \cdot (\eta'_\tau n'_\tau)\right)\right] \\ &= \sum_{\tau,\tau'} \mathbb{E}\left[P_T \cdots P_{\tau+1}(\eta_\tau n_\tau) \otimes (\eta_{\tau'} n_{\tau'}) P_{\tau'+1} \cdots P_T\right] \\ &= \sum_{\tau=1}^{T} \eta_\tau^2 \mathbb{E}\left[P_T \cdots P_{\tau+1} n_\tau \otimes n_\tau P_{\tau+1} \cdots P_T\right] \end{aligned}$$

Firstly, note that this naturally implies that the sequence of covariances V_τ, $\tau = 1, \cdots, T$ initialized at (say), the solution, i.e., $\mathbf{V}_0 = 0$ naturally grows to its steady state covariance, i.e.,

$$V_1 \preceq V_2 \preceq \cdots \preceq V_\infty.$$

See lemma 3 of Jain et al. (2017a) for more details. Furthermore, what naturally follows in relating V_t to V_{t-1} is:

$$V_t \preceq \mathbb{E}\left[P_t V_{t-1} P_t^\top\right] + \eta_t^2 \sigma^2 \mathbf{H}. \tag{10}$$

□

Lemma 6 (Lemma 5 of Jain et al. (2017a))**.** *Running SGD with a (constant) stepsize sequence $\eta < 1/R^2$ achieves the following steady-state covariance:*

$$V_\infty \preceq \frac{\eta\sigma^2}{1 - \eta R^2}\mathbf{I}.$$

Lemma 7. *Suppose $\eta = 1/2R^2$, and $V_0 = \frac{\eta\sigma^2}{1-\eta R^2}\mathbf{I} = 2\eta\sigma^2\mathbf{I}$. For any sequence of learning rates $\eta_t \leq \eta = 1/2R^2 \; \forall \; t \in \{1, \cdots, t\}$, then,*

$$V_t \preceq 2\eta\sigma^2\mathbf{I} \quad \forall \quad t.$$

Proof. We will prove the lemma using an inductive argument. The base case, i.e. $t = 0$ follows from the problem statement. Note also that for SGD, $V_0 = 0$ implying the statement naturally follows. If, say, V_t satisfies the equation above, from equation 10, we have the following covariance for V_{t+1}:

$$\begin{aligned} V_{t+1} &\preceq \mathbb{E}\left[P_t V_t P_t^\top\right] + \eta_t^2 \sigma^2 \mathbf{H} \\ &= \mathbb{E}\left[(\mathbf{I} - \eta_t \mathbf{x}_t \mathbf{x}_t^\top) V_t (\mathbf{I} - \eta_t \mathbf{x}_t \mathbf{x}_t^\top)\right] + \eta_t^2 \sigma^2 \mathbf{H} \\ &\preceq 2\eta\sigma^2 \mathbb{E}\left[(\mathbf{I} - \eta_t \mathbf{x}_t \mathbf{x}_t^\top)(\mathbf{I} - \eta_t \mathbf{x}_t \mathbf{x}_t^\top)\right] + \eta_t^2 \sigma^2 \mathbf{H} \\ &\preceq 2\eta\sigma^2 \mathbf{I} - 4\eta_t \eta \sigma^2 \mathbf{H} + 2\eta_t^2 \eta \sigma^2 R^2 \mathbf{H} + \eta_t^2 \sigma^2 \mathbf{H} \\ &\preceq 2\eta\sigma^2 \mathbf{I} - 2\eta_t \eta \sigma^2 \mathbf{H} + \eta_t^2 \sigma^2 \mathbf{H} \end{aligned}$$

$$
\begin{aligned}
&= 2\eta\sigma^2\mathbf{I} + \eta_t \cdot (\eta_t - 2\eta)\sigma^2\mathbf{H} \\
&\preceq 2\eta\sigma^2\mathbf{I},
\end{aligned}
$$

from which the lemma follows. □

Lemma 8. (Reduction from Multiplicative noise oracle) *Let V_t be the (expected) covariance of the variance error. Then, the recursion that connects V_{t+1} to V_t can be expressed as:*

$$
V_{t+1} \preceq (\mathbf{I} - \eta_t\mathbf{H})V_t(\mathbf{I} - \eta_t\mathbf{H}) + 2\eta_t^2\sigma^2\mathbf{H}
$$

Proof. From equation 10, we already know that the evolution of the co-variance of the variance error follows:

$$
\begin{aligned}
V_{t+1} &\preceq \mathbb{E}\left[P_tV_tP_t^\top\right] + \eta_t^2\sigma^2\mathbf{H} \\
&\preceq \mathbb{E}\left[(\mathbf{I} - \eta_t\mathbf{H})V_t(\mathbf{I} - \eta_t\mathbf{H})\right] + \eta_t^2\mathbb{E}\left[\mathbf{x}_t\mathbf{x}_t^\top V_t\mathbf{x}_t\mathbf{x}_t^\top\right] + \eta_t^2\sigma^2\mathbf{H} \\
&\preceq (\mathbf{I} - \eta_t\mathbf{H})V_t(\mathbf{I} - \eta_t\mathbf{H}) + \eta_t^2\left\|V_t\right\|_2 R^2\mathbf{H} + \eta_t^2\sigma^2\mathbf{H} \\
&= (\mathbf{I} - \eta_t\mathbf{H})V_t(\mathbf{I} - \eta_t\mathbf{H}) + \eta_t^2 \cdot 2\eta\sigma^2R^2\mathbf{H} + \eta_t^2\sigma^2\mathbf{H} \\
&\preceq (\mathbf{I} - \eta_t\mathbf{H})V_t(\mathbf{I} - \eta_t\mathbf{H}) + 2\eta_t^2\sigma^2\mathbf{H}.
\end{aligned}
$$

Where the steps follow from lemma 7, and owing from the fact that $\eta_t \leq \eta = 1/2R^2 \ \forall \ t$. □

Note: Basically, one could analyze an auxiliary process driven by noise with variance off by a factor of two and convert the analysis into one involving exact (deterministic) gradients.

Lemma 9. ***[Bias decay - strongly convex case]*** *Let the minimal eigenvalue of the Hessian $\mu = \lambda_{min}(\mathbf{H}) > 0$. Consider the bias recursion as in equation 8 with the stepsize set as $\eta = 1/(2R^2)$. Then,*

$$
\mathbb{E}\left[\left\|\mathbf{w}_t^{bias} - \mathbf{w}^*\right\|_2^2\right] \leq (1 - 1/(2\kappa))\mathbb{E}\left[\left\|\mathbf{w}_{t-1}^{bias} - \mathbf{w}^*\right\|_2^2\right]
$$

Proof. The proof follows through straight forward computations:

$$
\begin{aligned}
\mathbb{E}\left[\left\|\mathbf{w}_t^{\text{bias}} - \mathbf{w}^*\right\|_2^2\right] &\leq \mathbb{E}\left[\left\|\mathbf{w}_{t-1}^{\text{bias}} - \mathbf{w}^*\right\|_2^2\right] - 2\eta\mathbb{E}\left[\left\|\mathbf{w}_{t-1}^{\text{bias}} - \mathbf{w}^*\right\|_{\mathbf{H}}^2\right] + \eta^2R^2\mathbb{E}\left[\left\|\mathbf{w}_{t-1}^{\text{bias}} - \mathbf{w}^*\right\|_{\mathbf{H}}^2\right] \\
&= \mathbb{E}\left[\left\|\mathbf{w}_{t-1}^{\text{bias}} - \mathbf{w}^*\right\|_2^2\right] - \eta\mathbb{E}\left[\left\|\mathbf{w}_{t-1}^{\text{bias}} - \mathbf{w}^*\right\|_{\mathbf{H}}^2\right] \\
&\leq (1 - \eta\mu)\mathbb{E}\left[\left\|\mathbf{w}_{t-1}^{\text{bias}} - \mathbf{w}^*\right\|_2^2\right],
\end{aligned}
$$

where, the first line follows from the fact that $\mathbb{E}\left[\left\|\mathbf{x}_t\right\|_2^2\mathbf{x}_t\mathbf{x}_t^\top\right] \preceq R^2\mathbf{H}$ and the result follows through the definition of κ. □

Lemma 10. ***[Reduction of the bias recursion with multiplicative noise to one resembling the variance recursion]*** *Consider the bias recursion that evolves as*

$$
B_t = \mathbb{E}\left[(\mathbf{w}_t - \mathbf{w}^*)(\mathbf{w}_t - \mathbf{w}^*)^\top\right] = \mathbb{E}\left[(\mathbf{I} - \gamma_t\mathbf{x}_t\mathbf{x}_t^\top)B_{t-1}(\mathbf{I} - \gamma_t\mathbf{x}_t\mathbf{x}_t^\top)\right] \quad \textit{with} \ \ B_0 = (\mathbf{w}_0 - \mathbf{w}^*)(\mathbf{w}_0 - \mathbf{w}^*)^\top.
$$

Then, the following recursion holds $\forall \gamma_t \leq 1/R^2$:

$$
B_t \preceq (\mathbf{I} - \gamma_t\mathbf{H})B_{t-1}(\mathbf{I} - \gamma_t\mathbf{H}) + \gamma_t^2R^2\left\|\mathbf{w}_0 - \mathbf{w}^*\right\|^2\mathbf{H}.
$$

Proof. The result follows owing to the following computations:

$$
\begin{aligned}
B_t &= \mathbb{E}\left[(\mathbf{w}_t - \mathbf{w}^*)(\mathbf{w}_t - \mathbf{w}^*)^\top\right] \\
&= \mathbb{E}\left[(\mathbf{I} - \gamma_t\mathbf{x}_t\mathbf{x}_t^\top)B_{t-1}(\mathbf{I} - \gamma_t\mathbf{x}_t\mathbf{x}_t^\top)\right]
\end{aligned}
$$

$$
\begin{aligned}
&\preceq (\mathbf{I}-\gamma_t\mathbf{H})B_{t-1}(\mathbf{I}-\gamma_t\mathbf{H}) + \gamma_t^2\mathbb{E}\left[(\mathbf{x}_t^\top B_{t-1}\mathbf{x}_t)\mathbf{x}_t\mathbf{x}_t^\top\right]\\
&\preceq (\mathbf{I}-\gamma_t\mathbf{H})B_{t-1}(\mathbf{I}-\gamma_t\mathbf{H}) + \gamma_t^2\mathbb{E}\left[\|B_{t-1}\|_2\right]R^2\mathbf{H}\\
&\preceq (\mathbf{I}-\gamma_t\mathbf{H})B_{t-1}(\mathbf{I}-\gamma_t\mathbf{H}) + \gamma_t^2\mathbb{E}\left[\|\mathbf{w}_{t-1}-\mathbf{w}^*\|_2^2\right]R^2\mathbf{H}\\
&\preceq (\mathbf{I}-\gamma_t\mathbf{H})B_{t-1}(\mathbf{I}-\gamma_t\mathbf{H}) + \gamma_t^2\mathbb{E}\left[\|\mathbf{w}_{0}-\mathbf{w}^*\|_2^2\right]R^2\mathbf{H},
\end{aligned}
$$

with the last inequality holding true if the squared distance to the optimum doesn't grow as a part of the recursion. We prove that this indeed is the case below:

$$
\begin{aligned}
\mathbb{E}\left[\|\mathbf{w}_{t-1}-\mathbf{w}^*\|_2^2\right] &= \mathbb{E}\left[\left\|\mathbf{w}_{t-2}-\gamma_{t-1}\mathbf{x}_{t-1}\mathbf{x}_{t-1}^\top-\mathbf{w}^*\right\|_2^2\right]\\
&\leq \mathbb{E}\left[\|\mathbf{w}_{t-2}-\mathbf{w}^*\|_2^2\right] - 2\gamma_{t-1}\mathbb{E}\left[\|\mathbf{w}_{t-2}-\mathbf{w}^*\|_{\mathbf{H}}^2\right] + \gamma_{t-1}^2R^2\mathbb{E}\left[\|\mathbf{w}_{t-2}-\mathbf{w}^*\|_{\mathbf{H}}^2\right]\\
&\leq \mathbb{E}\left[\|\mathbf{w}_{t-2}-\mathbf{w}^*\|_2^2\right] - \gamma_{t-1}\mathbb{E}\left[\|\mathbf{w}_{t-2}-\mathbf{w}^*\|_{\mathbf{H}}^2\right]\\
&\leq \mathbb{E}\left[\|\mathbf{w}_{t-2}-\mathbf{w}^*\|_2^2\right].
\end{aligned}
$$

Recursively applying the above argument yields the desired result. □

Note: This result implies that the bias error (in the smooth non-strongly convex case of the least squares regression with multiplicative noise) can be bounded by employing a similar lemma as that of the variance, where one can look at the quantity $R^2\cdot\|\mathbf{w}_0-\mathbf{w}^*\|_2^2$ as the analog of the variance σ^2 that drives the process.

Lemma 11. ***[Lower bounds on the additive noise oracle imply ones for the multiplicative noise oracle]*** *Under the assumption that the covariance of noise $\Sigma=\sigma^2\mathbf{H}$, the following statement holds. Let V_t be the (expected) covariance of the variance error. Then, the recursion that connects V_{t+1} to V_t can be expressed as:*

$$
V_{t+1} = \mathbb{E}\left[(\mathbf{I}-\eta_t\mathbf{x}_t\mathbf{x}_t^\top)V_t(\mathbf{I}-\eta_t\mathbf{x}_t\mathbf{x}_t^\top)\right] + \eta_t^2\sigma^2\mathbf{H}
$$

Then,

$$
V_{t+1} \succeq (\mathbf{I}-\eta_t\mathbf{H})V_t(\mathbf{I}-\eta_t\mathbf{H}) + \eta_t^2\sigma^2\mathbf{H}
$$

Proof. Let us consider firstly, the setting of (bounded) additive noise. Here, we have:

$$
\hat{\nabla}f(\mathbf{w}_t) = \mathbf{H}(\mathbf{w}_t-\mathbf{w}^*)+\zeta_t, \text{ with } \mathbb{E}\left[\zeta_t|\mathbf{w}_t\right]=0, \text{ and } \mathbb{E}\left[\zeta_t\zeta_t^\top|\mathbf{w}_t\right]=\sigma^2\mathbf{H}.
$$

Then, updates leading upto time $t+1$ can be written as:

$$
\mathbf{w}_{t+1}-\mathbf{w}^* = \prod_{\tau=1}^{t+1}(\mathbf{I}-\eta_\tau\mathbf{H})(\mathbf{w}_0-\mathbf{w}^*) + \sum_{\tau'=1}^{t+1}\eta_{\tau'}\prod_{\tau=\tau'+1}^{t+1}(\mathbf{I}-\eta_\tau\mathbf{H})\zeta_{\tau'}
$$

This implies the covariance of the variance error is:

$$
\begin{aligned}
\tilde{V}_{t+1} &= \mathbb{E}\left[\left(\sum_{\tau'=1}^{t+1}\eta_{\tau'}\prod_{\tau=\tau'+1}^{t+1}(\mathbf{I}-\eta_\tau\mathbf{H})\zeta_{\tau'}\right)\otimes\left(\sum_{\tau''=1}^{t+1}\eta_{\tau''}\prod_{\tau=\tau''+1}^{t+1}(\mathbf{I}-\eta_\tau\mathbf{H})\zeta_{\tau''}\right)\right]\\
&= \sum_{\tau'=1}^{t+1}\eta_{\tau'}^2\mathbb{E}\left[\prod_{\tau=\tau'+1}^{t+1}(\mathbf{I}-\eta_\tau\mathbf{H})\zeta_{\tau'}\otimes\zeta_{\tau'}\prod_{\tau=t+1}^{\tau'+1}(\mathbf{I}-\eta_\tau\mathbf{H})\right]\\
&= (\mathbf{I}-\eta_{t+1}\mathbf{H})V_t(\mathbf{I}-\eta_{t+1}\mathbf{H}) + \eta_{t+1}^2\sigma^2\mathbf{H}.
\end{aligned}
$$

Now, let us consider the statement of the lemma:

$$
V_{t+1} = \mathbb{E}\left[(\mathbf{I}-\eta_{t+1}\mathbf{x}_{t+1}\mathbf{x}_{t+1}^\top)V_t(\mathbf{I}-\eta_{t+1}\mathbf{x}_{t+1}\mathbf{x}_{t+1}^\top)\right] + \eta_{t+1}^2\sigma^2\mathbf{H}
$$

$$= (\mathbf{I} - \eta_{t+1}\mathbf{H})V_t(\mathbf{I} - \eta_{t+1}\mathbf{H}) + \eta_{t+1}^2\mathbb{E}\left[(\mathbf{x}_{t+1}\mathbf{x}_{t+1}^\top - \mathbf{H})V_t(\mathbf{x}_{t+1}\mathbf{x}_{t+1}^\top - \mathbf{H})\right] + \eta_{t+1}^2\sigma^2\mathbf{H}$$
$$\succeq (\mathbf{I} - \eta_{t+1}\mathbf{H})V_t(\mathbf{I} - \eta_{t+1}\mathbf{H}) + \eta_t^2\sigma^2\mathbf{H}.$$

Unrolling the above argument and straightforward induction, we see that $V_{t+1} \succeq \tilde{V}_{t+1}$, implying that the process driven by the multiplicative noise oracle can be lower bounded (in a PSD sense) by one that employs deterministic gradients with additive noise. □

B Proofs of results in Section 3.1

Theorem 12. *Consider the additive noise oracle setting, where, we have access to stochastic gradients satisfying:*

$$\widehat{\nabla f}(\mathbf{w}) = \nabla f(\mathbf{w}) + \zeta = \mathbf{H}(\mathbf{w} - \mathbf{w}^*) + \zeta,$$

where,

$$\mathbb{E}\left[\zeta|\mathbf{w}\right] = 0, \text{ and, } \mathbb{E}\left[\zeta\zeta^\top|\mathbf{w}\right] = \sigma^2\mathbf{H}$$

The following lower bounds hold on the final iterate of a Stochastic Gradient procedure with access to the above stochastic gradients when using polynomially decaying stepsizes.

***Strongly convex case**: Suppose $\mu > 0$. For any condition number κ, there exists a problem instance with initial suboptimality $f(\mathbf{w}_0) - f(\mathbf{w}^*) \leq \sigma^2 d$ such that, for any $T \geq \kappa^{\frac{4}{3}}$, and for all $a, b \geq 0$ and $0.5 \leq \alpha \leq 1$, and for the learning rate scheme $\eta_t = \frac{a}{b+t^\alpha}$, we have*

$$\mathbb{E}\left[f(\mathbf{w}_T)\right] - f(\mathbf{w}^*) \geq \exp\left(-\frac{T}{\kappa\log T}\right)\left(f(\mathbf{w}_0) - f(\mathbf{w}^*)\right) + \frac{\sigma^2 d}{64}\cdot\frac{\kappa}{T}.$$

***Smooth case**: For any fixed $T > 1$, there exists a problem instance such that, for all $a, b \geq 0$ and $0.5 \leq \alpha \leq 1$, and for the learning rate scheme $\eta_t = \frac{a}{b+t^\alpha}$, we have*

$$\mathbb{E}\left[f(\mathbf{w}_T)\right] - f(\mathbf{w}^*) \geq \left(L\cdot\|\mathbf{w}_0 - \mathbf{w}^*\|^2 + \sigma^2 d\right)\cdot\frac{1}{\sqrt{T}\log T}.$$

Proof. **Strongly convex case**: The problem instance is simple. Consider the case where the inputs are such that in every example $\mathbf{x}$, there is only one co-ordinate that is non-zero. Furthermore, let each co-ordinate be Gaussian with mean zero and variance for the first $d/2$ co-ordinates be $d\kappa/3$ whereas the rest be 1. This implies $\mathbf{H} = \begin{bmatrix} d\kappa/3 & & & \\ & \ddots & & \\ & & 1 & \\ & & & \ddots \end{bmatrix}$, where the first $\frac{d}{2}$ diagonal entries are equal to $\kappa/3$ and the remaining $\frac{d}{2}$ diagonal entries are equal to 1 and all the off diagonal entries are equal to zero. Furthermore, consider the noise to be additive (and independent of $\mathbf{x}$) with mean zero. Finally, let us denote by $v_t^{(i)} \stackrel{\text{def}}{=} \mathbb{E}\left[\left(\mathbf{w}_t^{(i)} - (\mathbf{w}^*)^{(i)}\right)^2\right]$ the variance in the i^{th} direction at time step t. Let the initialization be such that $v_0^{(i)} = 3\sigma^2/\kappa$ for $i = 1, 2, ..., d/2$ and $v_0^{(i)} = \sigma^2$ for $i = d/2+1, ..., d$. This means that the variances for all directions with eigenvalue κ remain equal as t progresses and similarly for all directions with eigenvalue 1. We have

$$v_T^{(1)} \stackrel{\text{def}}{=} \mathbb{E}\left[\left(\mathbf{w}_T^{(1)} - (\mathbf{w}^*)^{(1)}\right)^2\right] = \prod_{j=1}^{T}\left(1 - \eta_j\kappa/3\right)^2 v_0^{(1)} + \kappa\sigma^2/3\sum_{j=1}^{T}\eta_j^2\prod_{i=j+1}^{T}\left(1 - \eta_i\kappa/3\right)^2 \text{ and}$$

$$v_T^{(d)} \stackrel{\text{def}}{=} \mathbb{E}\left[\left(\mathbf{w}_T^{(d)} - (\mathbf{w}^*)^{(d)}\right)^2\right] = \prod_{j=1}^{T}\left(1 - \eta_j\right)^2 v_0^{(d)} + \sigma^2\sum_{j=1}^{T}\eta_j^2\prod_{i=j+1}^{T}\left(1 - \eta_i\right)^2.$$

We consider a recursion for $v_t^{(i)}$ with eigenvalue λ_i (κ or 1). By the design of the algorithm, we know

$$v_{t+1}^{(i)} = (1-\eta_t\lambda_i)^2 v_t^{(i)} + \lambda_i\sigma^2\eta_t^2.$$

Let $s(\eta,\lambda) = \frac{\lambda\sigma^2\eta^2}{1-(1-\eta\lambda)^2}$ be the solution to the stationary point equation $x = (1-\eta\lambda)^2 + \lambda\sigma^2\eta^2$. Intuitively if we keep using the same learning rate η, then $v_t^{(i)}$ is going to converge to $s(\eta,\lambda_i)$. Also note that $s(\eta,\lambda) \approx \sigma^2\eta/2$ when $\eta\lambda \ll 1$.

We first prove the following claim showing that eventually the variance in direction i is going to be at least $s(\eta_T,\lambda_i)$.

Claim 1. *Suppose $s(\eta_t,\lambda_i) \le v_0^{(i)}$, then $v_t^{(i)} \ge s(\eta_t,\lambda_i)$.*

Proof. We can rewrite the recursion as

$$v_{t+1}^{(i)} - s(\eta_t,\lambda_i) = (1-\eta_t\lambda_i)^2(v_t^{(i)} - s(\eta_t,\lambda_i)).$$

In this form, it is easy to see that the iteration is a contraction towards $s(\eta_t,\lambda_i)$. Further, $v_{t+1}^{(i)} - s(\eta_t,\lambda_i)$ and $v_t^{(i)} - s(\eta_t,\lambda_i)$ have the same sign. In particular, let t_0 be the first time such that $s(\eta_t,\lambda_i) \le v_0^{(i)}$ (note that η_t is monotone and so is $s(\eta_t,\lambda_i)$), it is easy to see that $v_t^{(i)} \ge v_0^{(i)}$ when $t \le t_0$. Therefore we know $v_{t_0}^{(i)} \ge s(\eta_{t_0},\lambda_i)$, by the recursion this implies $v_{t_0+1}^{(i)} \ge s(\eta_{t_0},\lambda_i) \ge s(\eta_{t_0+1},\lambda_i)$. The claim then follows from a simple induction. □

If $s(\eta_T,\lambda_i) \ge v_0^{(i)}$ for $i=1$ or $i=d$ then the error is at least $\sigma^2 d/2 \ge \kappa\sigma^2 d/T$ and we are done. Therefore we must have $s(\eta_T,\kappa) \le v_0^{(1)} = 3\sigma^2/\kappa$, and by Claim 1 we know $v_T^{(1)} \ge s(\eta_T,\kappa) \ge \sigma^2\eta_T/2$. The function value is at least

$$\mathbb{E}\left[f(\mathbf{w}_T)\right] - f(\mathbf{w}^*) \ge \frac{d}{2}\cdot\kappa\cdot v_T^{(1)} \ge \frac{d\kappa\sigma^2\eta_T}{12}.$$

To make sure $\mathbb{E}\left[f(\mathbf{w}_T)\right] - f(\mathbf{w}^*) \le \frac{d\kappa\sigma^2}{64T}$ we must have $\eta_T \le \frac{1}{6T}$. Next we will show that when this happens, $v_T^{(d)}$ must be large so the function value is still large.

We will consider two cases, in the first case, $b \ge T^\alpha$. Since $\frac{1}{16T} \ge \eta_T = \frac{a}{b+T^\alpha} \ge \frac{a}{2b}$, we have $\frac{a}{b} \le \frac{1}{8T}$. Therefore $v_T^{(d)} \ge (1-\frac{a}{b})^{2T} v_0^{(d)} \ge \sigma^2/2$, so the function value is at least $\mathbb{E}\left[f(\mathbf{w}_t)\right] \ge \frac{d}{2}\cdot v_T^{(d)} \ge \frac{d\sigma^2}{4} \ge \frac{\kappa d\sigma^2}{T}$, and we are done.

In the second case, $b < T^\alpha$. Since $\frac{1}{16T} \ge \eta_T = \frac{a}{b+T^\alpha} \ge \frac{a}{2T^\alpha}$, we have $a \le \frac{1}{8}T^{\alpha-1}$. The sum of learning rates satisfy

$$\sum_{i=1}^T \eta_i \le \sum_{i=1}^T \frac{a}{i^\alpha} \le \sum_{i=1}^T \frac{1}{8} i^{-1} \le 0.125\log T.$$

Here the second inequality uses the fact that $T^{\alpha-1}i^{-\alpha} \le i^{-1}$ when $i \le T$. Similarly, we also know $\sum_{i=1}^T \eta_i^2 \le \sum_{i=1}^T (0.125)^2 i^{-2} \le \pi^2/384$. Using the approximation $(1-u)^2 \ge \exp(-2u-4u^2)$ for $u < 1/4$, we get $v_T^{(d)} \ge \exp(-2\sum_{i=1}^T \eta_i - 4\sum_{i=1}^T \eta_i^2) v_0^{(d)} \ge \sigma^2/5T^{\frac{1}{4}}$, so the function value is at least $\mathbb{E}\left[f(\mathbf{w}_t)\right] \ge \frac{d}{2}\cdot v_T^{(d)} \ge \frac{d\sigma^2}{10T^{\frac{1}{4}}} \ge \frac{\kappa d\sigma^2}{32T}$. This concludes the second case and proves the strongly convex part of the theorem.

Smooth case: The proof of this part is quite similar to that of the strongly convex case above but with a subtle change in the initialization. In order to make this clear, we will do the proof from scratch with out borrowing anything from the previous argument. Let $\mathbf{H} = \begin{bmatrix} 1 & & & \\ & \ddots & & \\ & & \frac{d}{\kappa} & \\ & & & \ddots \end{bmatrix}$, where the first $\frac{d}{2}$ diagonal entries are equal to 1 and the remaining $\frac{d}{2}$ diagonal entries are equal to $\frac{d}{\kappa}$ and all the off diagonal entries are equal to zero. We will use $\kappa = \frac{1}{\sqrt{T}}$. Let us denote by $v_t^{(i)} \stackrel{\text{def}}{=} \mathbb{E}\left[\left(\mathbf{w}_t^{(i)} - (\mathbf{w}^*)^{(i)}\right)^2\right]$ the variance in the i^{th} direction at time step t. Let the initialization be

such that $v_0^{(i)} = \sigma^2/\kappa$ for $i = 1, 2, ..., d/2$ and $v_0^{(i)} = \sigma^2$ for $i = d/2+1, ..., d$. This means that the variances for all directions with eigenvalue κ remain equal as t progresses and similarly for all directions with eigenvalue 1. We have

$$v_T^{(1)} \stackrel{\text{def}}{=} \mathbb{E}\left[\left(\mathbf{w}_T^{(1)} - (\mathbf{w}^*)^{(1)}\right)^2\right] = \prod_{j=1}^{T}(1-\eta_j\kappa/3)^2\, v_0^{(1)} + \kappa\sigma^2/3\sum_{j=1}^{T}\eta_j^2\prod_{i=j+1}^{T}(1-\eta_i\kappa/3)^2 \text{ and}$$

$$v_T^{(d)} \stackrel{\text{def}}{=} \mathbb{E}\left[\left(\mathbf{w}_T^{(d)} - (\mathbf{w}^*)^{(d)}\right)^2\right] = \prod_{j=1}^{T}(1-\eta_j)^2\, v_0^{(d)} + \sigma^2\sum_{j=1}^{T}\eta_j^2\prod_{i=j+1}^{T}(1-\eta_i)^2.$$

We consider a recursion for $v_t^{(i)}$ with eigenvalue λ_i (1 or $\frac{1}{\kappa}$). By the design of the algorithm, we know

$$v_{t+1}^{(i)} = (1-\eta_t\lambda_i)^2 v_t^{(i)} + \lambda_i\sigma^2\eta_t^2.$$

Let $s(\eta,\lambda) = \frac{\lambda\sigma^2\eta^2}{1-(1-\eta\lambda)^2}$ be the solution to the stationary point equation $x = (1-\eta\lambda)^2 + \lambda\sigma^2\eta^2$. Intuitively if we keep using the same learning rate η, then $v_t^{(i)}$ is going to converge to $s(\eta,\lambda_i)$. Also note that $s(\eta,\lambda) \approx \sigma^2\eta/2$ when $\eta\lambda \ll 1$.

If $s(\eta_T,\lambda_i) \geq v_0^{(i)}$ for $i = 1$ or $i = d$ then the error is at least $\sigma^2 d/2\kappa \geq \kappa\sigma^2 d/T$ and we are done. Therefore we must have $s(\eta_T,\kappa) \leq v_0^{(1)} = 3\sigma^2/\kappa$, and by Claim 1 we know $v_T^{(1)} \geq s(\eta_T,\kappa) \geq \sigma^2\eta_T/2$. The function value is at least

$$\mathbb{E}\left[f(\mathbf{w}_T)\right] - f(\mathbf{w}^*) \geq \frac{d}{2}\cdot v_T^{(1)} \geq \frac{d\sigma^2\eta_T}{4}.$$

To make sure $\mathbb{E}\left[f(\mathbf{w}_T)\right] - f(\mathbf{w}^*) \leq \frac{d\kappa\sigma^2}{64T\log T}$ we must have $\eta_T \leq \frac{\kappa}{16T\log T}$. Next we will show that when this happens, $v_T^{(d)}$ must be large so the function value is still large.

We will consider two cases, in the first case, $b \geq T^\alpha$. Since $\frac{\kappa}{16T\log T} \geq \eta_T = \frac{a}{b+T^\alpha} \geq \frac{a}{2b}$, we have $\frac{a}{b} \leq \frac{\kappa}{8T\log T}$. Therefore $v_T^{(d)} \geq (1-\frac{a}{b})^{2T} v_0^{(d)} \geq \sigma^2/2$, so the function value is at least $\mathbb{E}\left[f(\mathbf{w}_t)\right] - f(\mathbf{w}^*) \geq \frac{d}{2}\cdot\frac{1}{\kappa}\cdot v_T^{(d)} \geq \frac{d\sigma^2}{4\kappa} \geq \frac{\kappa d\sigma^2}{T}$, and we are done.

In the second case, $b < T^\alpha$. Since $\frac{\kappa}{16T\log T} \geq \eta_T = \frac{a}{b+T^\alpha} \geq \frac{a}{2T^\alpha}$, we have $a \leq \frac{1}{8\log T}\kappa T^{\alpha-1}$. The sum of learning rates satisfy

$$\sum_{i=1}^{T}\eta_i \leq \sum_{i=1}^{T}\frac{a}{i^\alpha} \leq \sum_{i=1}^{T}\frac{1}{8\log T}\kappa i^{-1} \leq 0.125\kappa.$$

Here the second inequality uses the fact that $T^{\alpha-1}i^{-\alpha} \leq i^{-1}$. Similarly, we also know

$$\sum_{i=1}^{T}\eta_i^2 \leq \sum_{i=1}^{T}(0.125\kappa/\log T)^2 i^{-2} \leq \pi^2\kappa^2/384.$$

Using the approximation $(1-u)^2 \geq \exp(-2u-4u^2)$ for $u < 1/4$, we get $v_T^{(d)} \geq \exp(-2\sum_{i=1}^{T}\frac{\eta_i}{\kappa} - 4\sum_{i=1}^{T}\frac{\eta_i^2}{\kappa^2})v_0^{(d)} \geq \sigma^2/5$, so the function value is at least $\mathbb{E}\left[f(\mathbf{w}_t)\right] \geq \frac{d}{2}\cdot\frac{1}{\kappa}\cdot v_T^{(d)} \geq \frac{d\sigma^2}{10\kappa} \geq \frac{d\sigma^2}{10\sqrt{T}}$. This concludes the second case and proves the strongly convex part of the theorem. Since $\|\mathbf{H}\|\cdot\|\mathbf{w}_0 - \mathbf{w}^*\|^2 = d\sigma^2$, we have

$$\mathbb{E}\left[f(\mathbf{w}_T)\right] - f(\mathbf{w}^*) \geq \sigma^2 d\cdot\min\left(\frac{\kappa}{T\log T}, \frac{1}{10\sqrt{T}}\right) \geq \left(L\cdot\|\mathbf{w}_0 - \mathbf{w}^*\|^2 + \sigma^2 d\right)\cdot\frac{1}{\sqrt{T}\log T}.$$

This proves the theorem. □

Proof of Theorem 1. The proof of theorem 1 follows straightforwardly when combining the result of lemma 11 and theorem 12. □

C Proofs of results in Section 3.2

Theorem 13. *Consider the additive noise oracle setting, where, we have access to stochastic gradients satisfying:*

$$\widehat{\nabla f}(\mathbf{w}) = \nabla f(\mathbf{w}) + \zeta = \mathbf{H}(\mathbf{w} - \mathbf{w}^*) + \zeta,$$

where,

$$\mathbb{E}\left[\zeta|\mathbf{w}\right] = 0, \text{ and, } \mathbb{E}\left[\zeta\zeta^\top|\mathbf{w}\right] \preceq \hat{\sigma}^2\mathbf{H}$$

Running Algorithm 1 with an initial stepsize of $\eta_1 = 1/R^2$, starting from the solution, i.e. $\mathbf{w}_0 = \mathbf{w}^*$ allows the algorithm to obtain the following dependence on the variance error:

$$\mathbb{E}\left[f(\mathbf{w}_T^{var})\right] - f(\mathbf{w}^*) \leq 2\frac{d\hat{\sigma}^2\log T}{T}$$

Proof. The learning rate scheme is as follows. Divide the total time horizon T into $\log T$ phases, each of length $\frac{T}{\log T}$. In the ℓ^{th} phase, the learning rate is set to be $\frac{1}{2^\ell R^2}$. The variance in the k^{th} coordinate can be bounded as

$$\begin{aligned}
v_T^{(k)} &\leq \prod_{j=1}^{T}\left(1-\eta_j\lambda^{(k)}\right)^2 v_0^{(k)} + \lambda^{(k)}\hat{\sigma}^2\sum_{j=1}^{T}\eta_j^2\prod_{i=j+1}^{T}\left(1-\eta_i\lambda^{(k)}\right)^2 \\
&\leq \exp\left(-2\sum_{j=1}^{T}\eta_j\lambda^{(k)}\right)v_0^{(k)} \\
&\quad + \lambda^{(k)}\hat{\sigma}^2\sum_{\ell=1}^{\log T}\frac{1}{2^{2\ell}(R^2)^2}\sum_{j=1}^{T/\log T}\left(1-\frac{\lambda^{(k)}}{2^\ell(R^2)}\right)^{2j}\cdot\prod_{u=\ell+1}^{\log T}\left(1-\frac{\lambda^{(k)}}{2^u R^2}\right)^{T/\log T} \\
&\leq \exp\left(-\frac{2\lambda^{(k)}}{R^2}\cdot\frac{T}{\log T}\right)v_0^{(k)} + \lambda^{(k)}\hat{\sigma}^2\sum_{\ell=1}^{\log T}\frac{1}{2^{2\ell}(R^2)^2}\cdot\frac{2^\ell R^2}{\lambda^{(k)}}\cdot\prod_{u=\ell+1}^{\log T}\exp\left(-\frac{\lambda^{(k)}T}{2^u R^2\log T}\right) \\
&\leq \exp\left(-\frac{2\lambda^{(k)}}{R^2}\cdot\frac{T}{\log T}\right)v_0^{(k)} + \sum_{\ell=1}^{\log T}\frac{\hat{\sigma}^2}{2^\ell R^2}\prod_{u=\ell+1}^{\log T}\exp\left(-\frac{\lambda^{(k)}T}{2^u R^2\log T}\right). \qquad (11)
\end{aligned}$$

Let $\ell^* \stackrel{\text{def}}{=} \max\left(0, \lfloor\log\left(\frac{\lambda^{(k)}}{R^2}\cdot\frac{T}{\log T}\right)\rfloor\right)$. We now split the summation in the second term in (11) into two parts and bound each of them below.

$$\begin{aligned}
&\sum_{\ell=1}^{\ell^*}\frac{\hat{\sigma}^2}{2^\ell R^2}\prod_{u=\ell+1}^{\log T}\exp\left(-\frac{\lambda^{(k)}T}{2^u R^2\log T}\right) \leq \sum_{\ell=1}^{\ell^*}\frac{\hat{\sigma}^2}{2^\ell R^2}\prod_{u=\ell+1}^{\ell^*}\exp\left(-\frac{\lambda^{(k)}T}{2^u R^2\log T}\right) \\
&\leq \sum_{\ell=1}^{\ell^*}\frac{\hat{\sigma}^2}{2^\ell R^2}\prod_{u=\ell+1}^{\ell^*}\exp\left(-2^{\ell^*-u}\right) \leq \sum_{\ell=1}^{\ell^*}\frac{\hat{\sigma}^2}{2^\ell R^2}\exp\left(-2^{\ell^*-\ell}\right) \\
&\leq \frac{\hat{\sigma}^2}{2^{\ell^*}R^2}\sum_{\ell=1}^{\ell^*}2^{\ell^*-\ell}\exp\left(-2^{\ell^*-\ell}\right) \leq \frac{\hat{\sigma}^2}{2^{\ell^*}R^2} \leq \frac{\hat{\sigma}^2}{\lambda^{(k)}}\cdot\frac{\log T}{T}. \qquad (12)
\end{aligned}$$

For the second part, we have

$$\sum_{\ell=\ell^*+1}^{\log T}\frac{\hat{\sigma}^2}{2^\ell R^2}\prod_{u=\ell+1}^{\log T}\exp\left(-\frac{\lambda^{(k)}T}{2^u R^2\log T}\right) \leq \sum_{\ell=\ell^*+1}^{\log T}\frac{\hat{\sigma}^2}{2^\ell R^2} \leq \sum_{\ell=\ell^*+1}^{\log T}\frac{\hat{\sigma}^2}{2^{\ell^*}R^2} \leq \frac{\hat{\sigma}^2}{\lambda^{(k)}}\cdot\frac{\log T}{T}. \qquad (13)$$

Plugging (12) and (13) into (11), we obtain

$$v_T^{(k)} \leq \exp\left(-\frac{2\lambda^{(k)}}{R^2}\cdot\frac{T}{\log T}\right)v_0^{(k)} + \frac{2\hat{\sigma}^2}{\lambda^{(k)}}\cdot\frac{\log T}{T}.$$

The function suboptimality can now be bounded as

$$\begin{aligned}\mathbb{E}\left[f(\mathbf{w}_T^{\text{var}})\right] - f(\mathbf{w}^*) &= \sum_{k=1}^{d}\lambda^{(k)}\cdot v_T^{(k)} \\ &\leq \sum_{k=1}^{d}\lambda^{(k)}\left(\exp\left(-\frac{2\lambda^{(k)}}{R^2}\cdot\frac{T}{\log T}\right)v_0^{(k)} + \frac{2\hat{\sigma}^2}{\lambda^{(k)}}\cdot\frac{\log T}{T}\right).\end{aligned}$$

$$\mathbb{E}\left[f(\mathbf{w}_T^{\text{var}})\right] - f(\mathbf{w}^*) \leq \sum_{k=1}^{d}\left(\frac{L\log T}{T}v_0^{(k)} + 2\hat{\sigma}^2\cdot\frac{\log T}{T}\right) = 2\left(\hat{\sigma}^2 d\right)\frac{\log T}{T}.$$

□

Proof of Theorem 2. **Smooth case:** The result follows by instantiating $\hat{\sigma}^2$ in theorem 13 with $2\sigma^2$ (lemma 8) and $R^2\left\|\mathbf{w}_0 - \mathbf{w}^*\right\|_2^2$ (lemma 10) and using the lemma 5 to obtain the result.

Strongly convex case: As with the smooth case, the result relies on instantiating theorem 13 with $2\sigma^2$ (lemma 8) and using lemma 9 and then appealing to lemma 5. □

Proposition 14. *Consider the additive noise oracle setting, where, we have access to stochastic gradients satisfying:*

$$\widehat{\nabla f}(\mathbf{w}) = \nabla f(\mathbf{w}) + \zeta = \mathbf{H}(\mathbf{w} - \mathbf{w}^*) + \zeta,$$

where,

$$\mathbb{E}\left[\zeta|\mathbf{w}\right] = 0, \textit{ and, } \mathbb{E}\left[\zeta\zeta^\top|\mathbf{w}\right] \leq \sigma^2\mathbf{H}$$

There exists a stepsize scheme with which, by starting at the solution (i.e. $\mathbf{w}_0 = \mathbf{w}^*$) the algorithm obtains the following dependence on the variance error, under the assumption that $\mu > 0$ and $\kappa \geq 2$.

$$\mathbb{E}\left[f(\mathbf{w}_T^{var})\right] - f(\mathbf{w}^*) \leq 50\log_2\kappa\cdot\frac{\sigma^2 d}{T}.$$

Proof. The learning rate scheme is as follows.

We first break T into three equal sized parts. Let $A = T/3$ and $B = 2T/3$. In the first $T/3$ steps, we use a constant learning rate of $1/R^2$. Note that at the end of this phase, (since $T > \kappa$) the dependence on the initial error decays geometrically. In the second $T/3$ steps, we use a polynomial decay learning rate $\eta_{A+t} = \frac{1}{\mu(\kappa+t/2)}$. In the third $T/3$ steps, we break the steps into $\log_2(\kappa)$ equal sized phases. In the ℓ^{th} phase, the learning rate to be used is $\frac{5\log_2\kappa}{2^\ell\cdot\mu\cdot T}$. Note that the learning rate in the first phase depends on strong convexity and that in the last phase depends on smoothness (since the last phase has $\ell = \log\kappa$).

Recall the variance in the k^{th} coordinate can be upper bounded by

$$\begin{aligned}v_T^{(k)} \overset{\text{def}}{=} \mathbb{E}\left[\left(\mathbf{w}_T^{(k)} - (\mathbf{w}^*)^{(1)}\right)^2\right] &\leq \prod_{j=1}^{T}\left(1-\eta_j\lambda^{(k)}\right)^2 v_0^{(1)} + \lambda^{(k)}\sigma^2\sum_{j=1}^{T}\eta_j^2\prod_{i=j+1}^{T}\left(1-\eta_i\lambda^{(k)}\right)^2 \\ &\leq \exp\left(-2\sum_{j=1}^{T}\eta_j\lambda^{(k)}\right)v_0^{(1)} + \lambda^{(k)}\sigma^2\sum_{j=1}^{T}\eta_j^2\exp\left(-2\sum_{i=j+1}^{T}\eta_i\lambda^{(k)}\right).\end{aligned}$$

We will show that for every k, we have

$$v_T^{(k)} \le \frac{v_0^{(k)}}{T^3} + \frac{50\log_2\kappa}{\lambda^{(k)}T}\cdot\sigma^2., \tag{14}$$

which directly implies the theorem.

We will consider the first $T/3$ steps. The guarantee that we will prove for these iterations is: for any $t \le A$, $v_t^{(k)} \le (1-\lambda^{(k)}/R^2)^{2t}v_0^{(k)} + \frac{\sigma^2}{R^2}$.

This can be proved easily by induction. Clearly this is true when $t=0$. Suppose it is true for $t-1$, let's consider step t. By recursion of $v_t^{(k)}$ we know

$$\begin{aligned} v_t^{(k)} &= (1-\lambda^{(k)}/R^2)^2 v_{t-1}^{(k)} + \lambda^{(k)}\sigma^2/(R^2)^2 \\ &\le (1-\lambda^{(k)}/R^2)^{2t}v_0^{(k)} + \frac{\sigma^2}{R^2}\left((1-\lambda^{(k)}/R^2)^2 + \lambda^{(k)}/R^2\right) \\ &\le (1-\lambda^{(k)}/R^2)^{2t}v_0^{(k)} + \frac{\sigma^2}{R^2}. \end{aligned}$$

Here the second step uses induction hypothesis and the third step uses the fact that $(1-x)^2 + x \le 1$ when $x \in [0,1]$. In particular, since $(1-\lambda^{(k)}/R^2)^{2T/3} \le (1-1/\kappa)^{2T/3} \le (1-1/\kappa)^{3\kappa\log T} = 1/T^3$, we know at the end of the first phase, $v_A^{(k)} \le v_0^{(k)}/T^3 + \frac{\sigma^2}{R^2}$.

In the second $T/3$ steps, the guarantee would be: for any $t \le T/3$, $v_{A+t}^{(k)} \le v_0^{(k)}/T^3 + 2\eta_{A+t}\sigma^2$.

We will again prove this by induction. The base case ($t=0$) follows immediately from the guarantee for the first part. Suppose this is true for $A+t-1$, let us consider $A+t$, again by recursion we know

$$\begin{aligned} v_{A+t}^{(k)} &= (1-\lambda^{(k)}\eta_{A+t-1})^2 v_{A+t-1}^{(k)} + \lambda^{(k)}\sigma^2\eta_{A+t-1}^2 \\ &\le v_0^{(k)}/T^3 + 2\eta_{A+t-1}\sigma^2\left((1-\lambda^{(k)}\eta_{A+t-1})^2 + \frac{1}{2}\lambda^{(k)}\eta_{A+t-1}\right) \\ &\le v_0^{(k)}/T^3 + 2\eta_{A+t-1}\sigma^2(1-\frac{1}{2}\mu\eta_{A+t-1}) \le v_0^{(k)}/T^3 + 2\eta_{A+t}\sigma^2. \end{aligned}$$

Here the last line uses the fact that $2\eta_{A+t-1}(1-\frac{1}{2}\mu\eta_{A+t-1}) \le 2\eta_{A+t}\sigma^2$, which is easy to verify by our choice of η. Therefore, at the end of the second part, we have $v_B^{(k)} \le v_0^{(k)}/T^3 + \frac{2\sigma^2}{\mu(\kappa+T/6)}$.

Finally we will analyze the third part. Let $\hat{T} = T/3\log_2\kappa$, we will consider the variance $v_{B+\ell\hat{T}}^{(k)}$ at the end of each phase. We will make the following claim by induction:

Claim 2. *Suppose $2^\ell\cdot\mu \le \lambda^{(k)}$, then*

$$v_{B+\ell\hat{T}}^{(k)} \le v_B^{(k)}\exp(-3\ell) + 2\hat{T}\eta_\ell^2\lambda^{(k)}\sigma^2.$$

Proof. We will prove this by induction. When $\ell=0$, clearly we have $v_B^{(k)} \le v_B^{(k)}$ so the claim is true. Suppose the claim is true for $\ell-1$, we will consider what happens after the algorithm uses η_ℓ for $\hat{T}$ steps. By the recursion of the variance we have

$$v_{\ell\hat{T}}^{(k)} \le v_{(\ell-1)\hat{T}}^{(k)}\cdot\exp(-2\eta_\ell\cdot\lambda^{(k)}\hat{T}) + \hat{T}\eta_\ell^2\lambda^{(k)}\sigma^2.$$

Since $2^\ell\cdot\mu \le \lambda^{(k)}$, we know $\exp(-2\eta_\ell\cdot\lambda^{(k)}\hat{T}) \le \exp(-3)$. Therefore by induction hypothesis we have

$$v_{B+\ell\hat{T}}^{(k)} \le v_B^{(k)}\exp(-3\ell) + \exp(-3)\cdot 2\hat{T}\eta_{\ell-1}^2\lambda^{(k)} + \hat{T}\eta_\ell^2\lambda^{(k)} \le v_B^{(k)}\exp(-3\ell) + 2\hat{T}\eta_\ell^2\lambda^{(k)}.$$

This finishes the induction. □

By Claim 2, Let ℓ^* denote the number satisfying $2^{\ell^*} \cdot \mu \leq \lambda^{(k)} < 2^{\ell^*+1} \cdot \mu$, by this choice we know $\mu/\lambda^{(k)} \geq \frac{1}{2}\exp(-3\ell^\star)$ we have

$$\begin{aligned} v_T^{(k)} \leq v_{B+\ell^*\hat{T}}^{(k)} &\leq v_B^{(k)} \exp(-3\ell^*) + 2\hat{T}\eta_{\ell^*}^2 \lambda^{(k)}\sigma^2 \\ &\leq \frac{v_0^{(k)}}{T^3} + \frac{24\sigma^2}{\lambda^{(k)}T} + \frac{50\log_2\kappa}{3\lambda^{(k)}T} \cdot \sigma^2. \\ &\leq \frac{v_0^{(k)}}{T^3} + \frac{50\log_2\kappa}{\lambda^{(k)}T} \cdot \sigma^2. \end{aligned}$$

Therefore, the function value is bounded by $\mathbb{E}\left[f(\mathbf{w}_T^{\text{var}})\right] - f(\mathbf{w}^*) = \sum_{i=1}^d \lambda^{(k)} v_T^{(k)} \leq \frac{50\log_2\kappa}{T} \cdot \sigma^2 d$. □

Proof of proposition 3. The proof of the proposition works similar to the proof of the strongly convex case of theorem 2, wherein, we combine the result of proposition 14 with lemma 9 and lemma 5 to obtain the result. □

D Proofs of results in Section 3.3

All of our counter-examples in this section are going to be the same simple function. Let the inputs x be such that only a single co-ordinate be active on each example. We refer to this case as the "discrete" case. Furthermore, let each co-ordinate be a Gaussian with mean 0 and variance for the first $d/2$ directions being $d\kappa/3$ and the final $d/2$ directions being 1. Furthermore, consider the noise to be additive (and independent of $\mathbf{x}$) with mean zero. This indicates that $R^2 = \kappa$ for this problem.

Intuitively, we will show that in order to have a small error in the first eigendirection (with eigenvalue κ), one need to set a small learning rate η_t which would be too small to achieve a small error in the second eigendirection (with eigenvalue 1). As a useful tool, we will decompose the variance in the two directions corresponding to κ eigenvalue and 1 eigenvalue respectively as follows:

$$\begin{aligned} v_T^{(1)} \stackrel{\text{def}}{=} \mathbb{E}\left[\left(\mathbf{w}_T^{(1)} - (\mathbf{w}^*)^{(1)}\right)^2\right] &= \prod_{j=1}^T (1-\eta_j\kappa)^2 v_0^{(1)} + \kappa\sigma^2 \sum_{j=1}^T \eta_j^2 \prod_{i=j+1}^T (1-\eta_i\kappa)^2 \\ &\geq \exp\left(-2\sum_{j=1}^T \eta_j\kappa\right) v_0^{(1)} + \kappa\sigma^2 \sum_{j=1}^T \eta_j^2 \exp\left(-2\sum_{i=j+1}^T \eta_i\kappa\right) \text{ and} \qquad (15) \\ v_T^{(2)} \stackrel{\text{def}}{=} \mathbb{E}\left[\left(\mathbf{w}_T^{(2)} - (\mathbf{w}^*)^{(2)}\right)^2\right] &= \prod_{j=1}^T (1-\eta_j)^2 v_0^{(2)} + \sigma^2 \sum_{j=1}^T \eta_j^2 \prod_{i=j+1}^T (1-\eta_i)^2 \\ &\geq \exp\left(-2\sum_{j=1}^T \eta_j\right) v_0^{(2)} + \sigma^2 \sum_{j=1}^T \eta_j^2 \exp\left(-2\sum_{i=j+1}^T \eta_i\right). \qquad (16) \end{aligned}$$

Theorem 15. *Consider the additive noise oracle setting, where, we have access to stochastic gradients satisfying:*

$$\widehat{\nabla f}(\mathbf{w}) = \nabla f(\mathbf{w}) + \zeta = \mathbf{H}(\mathbf{w} - \mathbf{w}^*) + \zeta,$$

where,

$$\mathbb{E}\left[\zeta|\mathbf{w}\right] = 0, \text{ and, } \mathbb{E}\left[\zeta\zeta^\top|\mathbf{w}\right] = \sigma^2\mathbf{H}$$

There exists a universal constant $C > 0$, and a problem instance, such that for SGD algorithm with any $\eta_t \leq 1/2\kappa$ for all t[5], we have

$$\limsup_{T\to\infty} \frac{\mathbb{E}\left[f(\mathbf{w}_T)\right] - f(\mathbf{w}^*)}{(\sigma^2 d/T)} \geq C\frac{\kappa}{\log(\kappa+1)}.$$

[5]Learning rate more than $2/\kappa$ will make the algorithm diverge.

Proof. Fix $\tau = \kappa/C\log(\kappa+1)$ where C is a universal constant that we choose later. We need to exhibit that the $\limsup$ is larger than τ. For simplicity we will also round κ up to the nearest integer.

Let T be a given number. Our goal is to exhibit a $\tilde{T} > T$ such that $\frac{f(\mathbf{w}_{\tilde{T}})-f(\mathbf{w}^*)}{(\sigma^2/\tilde{T})} \geq \tau$. Given the step size sequence η_t, consider the sequence of numbers $T_0 = T, T_1, \cdots, T_\kappa$ such that T_i is the first number that

$$\frac{1}{\kappa} \leq \sum_{t=T_{i-1}+1}^{T_i} \eta_t \leq \frac{3}{\kappa}.$$

Note that such a number always exists because all the step sizes are at most $2/\kappa$. We will also let Δ_i be $T_i - T_{i-1}$. Firstly, from (15) and (16), we see that $\sum_t \eta_t = \infty$. Otherwise, the bias will never decay to zero. If $f(\mathbf{w}_{T_{i-1}+\Delta_i}) - f(\mathbf{w}^*) > \frac{\tau\sigma^2 d}{T_{i-1}+\Delta_i}$ for some $i = 1, \cdots, \kappa$, we are done. If not, we obtain the following relations:

$$\begin{aligned}
&\frac{\sigma^2}{\Delta_1} \leq \sigma^2 \sum_{t=1}^{\Delta_1} \eta_{T_0+t}^2 \leq \frac{\exp(3)}{\kappa} \cdot \mathbb{E}\left[\left(\mathbf{w}_{T_0+\Delta_1}^{(1)} - (\mathbf{w}^*)^{(1)}\right)^2\right] \\
&\leq \exp(3)(f(\mathbf{w}_{T_0+\Delta_1}) - f(\mathbf{w}^*)) \leq \frac{\exp(3)\tau\sigma^2}{T_0+\Delta_1} \\
&\Rightarrow T_0 \leq (\exp(3)\tau - 1)\,\Delta_1.
\end{aligned}$$

Here the second inequality is based on (15). We will use C_1 to denote $\exp(3)$. Similarly, we have

$$\begin{aligned}
&\frac{\sigma^2}{\Delta_2} \leq \sigma^2 \sum_{t=1}^{\Delta_2} \eta_{T_1+t}^2 \leq \frac{C_1}{\kappa}\mathbb{E}\left[\left(\mathbf{w}_{T_1+\Delta_2}^{(1)} - (\mathbf{w}^*)^{(1)}\right)^2\right] \leq C_1(f(\mathbf{w}_{T_1+\Delta_2}) - f(\mathbf{w}^*)) \leq \frac{C_1\tau\sigma^2}{T_1+\Delta_2} \\
&\Rightarrow T_1 \leq (C_1\tau - 1)\,\Delta_2 \quad \Rightarrow \quad T_0 \leq \frac{(C_1\tau-1)^2}{C_1\tau}\Delta_2.
\end{aligned}$$

Repeating this argument, we can show that

$$T = T_0 \leq \frac{(C_1\tau-1)^i}{(C_1\tau)^{i-1}}\Delta_i \quad \text{and} \quad T_i \leq \frac{(C_1\tau-1)^{j-i}}{(C_1\tau)^{j-i-1}}\Delta_j \quad \forall\, i < j.$$

We will use $i = 1$ in particular, which specializes to

$$T_1 \leq \frac{(C_1\tau-1)^{j-1}}{(C_1\tau)^{j-2}}\Delta_j \quad \forall\, j \geq 2.$$

Using the above inequality, we can lower bound the sum of Δ_j as

$$\begin{aligned}
\sum_{j=2}^{\kappa}\Delta_j &\geq T_1 \cdot \sum_{j=2}^{\kappa} \frac{(C_1\tau)^{j-2}}{(C_1\tau-1)^{j-1}} \geq T_1 \cdot \frac{1}{C_1\tau} \cdot \sum_{j=2}^{\kappa}\left(1+\frac{1}{C_1\tau}\right)^{j-2} \\
&\geq T_1 \cdot \frac{1}{C_1\tau} \cdot \exp\left(\kappa/(C_1\tau)\right). \qquad (17)
\end{aligned}$$

This means that

$$\begin{aligned}
\mathbb{E}\left[f(\mathbf{w}_{T_i})\right] - f(\mathbf{w}^*) &\geq \frac{d}{2} \cdot \mathbb{E}\left[\left(\mathbf{w}_{T_i}^{(2)} - (\mathbf{w}^*)^{(2)}\right)^2\right] \geq \exp(-6)\sigma^2 d \cdot \sum_{i=1}^{\Delta_1} \eta_{T+i}^2 \\
&\geq \frac{\exp(-6)\sigma^2 d}{\Delta_1} \geq \frac{\exp(-6)\sigma^2 d}{T_1} \geq \frac{\exp\left(\kappa/(C_1\tau)-3\right)}{C_1\tau} \cdot \frac{\sigma^2 d}{\sum_{j=2}^{\kappa}\Delta_j},
\end{aligned}$$

where we used (17) in the last step. Rearranging, we obtain

$$\frac{\mathbb{E}\left[f(\mathbf{w}_{T_\kappa})\right] - f(\mathbf{w}^*)}{(\sigma^2 d/T_\kappa)} \geq \frac{\exp\left(\kappa/(C_1\tau) - 3\right)}{C_1\tau}.$$

If we choose a large enough C (e.g., $3C_1$), the right hand side is at least $\frac{\exp((C/C_1)\log(\kappa+1)-3)}{\kappa} \geq \kappa$.

□

Proof of theorem 4. Theorem 4 follows as a straightforward consequence of Theorem 15 and lemma 11. □

Theorem 16. *There exists universal constants $C_1, C_2 > 0$ such that for any $\tau \leq \frac{\kappa}{CC_1\log(\kappa+1)}$ where C is the constant in Theorem 4, for any SGD algorithm and any number of iteration $T > 0$ there exists a $T' \geq T$ such that for any $\tilde{T} \in [T', (1 + 1/C_2\tau)T']$ we have $\frac{\mathbb{E}\left[f(\mathbf{w}_{\tilde{T}})\right] - f(\mathbf{w}^*)}{(\sigma^2 d/\tilde{T})} \geq \tau$.*

Theorem 17. *Consider the additive noise oracle setting, where, we have access to stochastic gradients satisfying:*

$$\widehat{\nabla f}(\mathbf{w}) = \nabla f(\mathbf{w}) + \zeta = \mathbf{H}(\mathbf{w} - \mathbf{w}^*) + \zeta,$$

where,

$$\mathbb{E}\left[\zeta|\mathbf{w}\right] = 0, \text{ and, } \mathbb{E}\left[\zeta\zeta^\top|\mathbf{w}\right] = \sigma^2\mathbf{H}$$

There exists universal constants $C_1, C_2 > 0$ such that for any $\tau \leq \frac{\kappa}{CC_1\log(\kappa+1)}$ where C is the constant in Theorem 4, for any SGD algorithm and any number of iteration $T > 0$ there exists a $T' \geq T$ such that for any $\tilde{T} \in [T', (1 + 1/C_2\tau)T']$ we have $\frac{\mathbb{E}\left[f(\mathbf{w}_{\tilde{T}})\right] - f(\mathbf{w}^*)}{(\sigma^2 d/\tilde{T})} \geq \tau$.

To prove Theorem 17, we rely on the following key lemma, which says if a query point $\mathbf{w}_T$ is bad (in the sense that it has expected value more than $10\tau\sigma^2 d/T$), then it takes at least $\Omega(T/\tau)$ steps to bring the error back down.

Lemma 18. *There exists universal constants $C_1, C_2 > 0$ such that for any $\tau \leq \frac{\kappa}{CC_1\log(\kappa+1)}$ where C is the constant in Theorem 4, suppose at step T, the query point $\mathbf{w}_T$ satisfies $f(\mathbf{w}_T) - f(\mathbf{w}^*) \geq C_1\tau\sigma^2 d/T$, then for all $\tilde{T} \in [T, (1 + \frac{1}{C_2\tau})T]$ we have $\mathbb{E}\left[f(\mathbf{w}_{\tilde{T}})\right] - f(\mathbf{w}^*) \geq \tau\sigma^2 d/T \geq \tau\sigma^2 d/\tilde{T}$.*

Proof of Lemma 18. Since $f(\mathbf{w}_T) - f(\mathbf{w}^*) \geq C_1\tau\sigma^2 d/T$ and $f(\mathbf{w}_T) = \frac{d}{2}\left(\kappa\left(\mathbf{w}_T^{(1)} - (\mathbf{w}^*)^{(1)}\right)^2 + \left(\mathbf{w}_T^{(2)} - (\mathbf{w}^*)^{(2)}\right)^2\right)$, we know either $\left(\mathbf{w}_T^{(1)} - (\mathbf{w}^*)^{(1)}\right)^2 \geq C_1\tau\sigma^2/2\kappa T$ or $\left(\mathbf{w}_T^{(2)} - (\mathbf{w}^*)^{(2)}\right)^2 \geq C_1\tau\sigma^2/2T$. Either way, we have a coordinate i with eigenvalue λ_i (κ or 1) such that $\left(\mathbf{w}_T^{(i)} - (\mathbf{w}^*)^{(i)}\right)^2 \geq C_1\tau\sigma^2/(2T\lambda_i)$.

Similar as before, choose Δ to be the first point such that

$$\eta_{T+1} + \eta_{T+2} + \cdots + \eta_{T+\Delta} \in [1/\lambda_i, 3/\lambda_i].$$

First, by (15) or (16), we know for any $T \leq \tilde{T} \leq T + \Delta$, $\mathbb{E}\left[\left(\mathbf{w}_{\tilde{T}}^{(i)} - (\mathbf{w}^*)^{(i)}\right)^2\right] \geq \exp(-6)C_1\tau\sigma^2/(2\lambda_i T)$ just by the first term. When we choose C_1 to be large enough the contribution to function value by this direction alone is larger than $\tau\sigma^2/T$. Therefore every query in $[T, T + \Delta]$ is still bad.

We will consider two cases based on the value of $S^2 := \sum_{\tilde{T}=T+1}^{T+\Delta}\eta_{\tilde{T}}^2$.

If $S^2 \leq C_2\tau/(\lambda_i^2 T)$ (where C_2 is a large enough universal constant chosen later), then by Cauchy-Schwartz we know

$$S^2 \cdot \Delta \geq (\sum_{\tilde{T}=T+1}^{T+\Delta}\eta_{\tilde{T}})^2 \geq 1/\lambda_i^2.$$

Therefore $\Delta \geq T/C_2\tau$, and we are done.

If $S^2 > C_2\tau/(\lambda_i^2 T)$, by Equation (15) and (16) we know

$$\mathbb{E}\left[\left(\mathbf{w}_{T+\Delta}^{(i)} - (\mathbf{w}^*)^{(i)}\right)^2\right] \geq \sigma^2 \sum_{\tilde{T}=T+1}^{T+\Delta} \eta_{\tilde{T}}^2 \exp\left(-2\lambda_i \sum_{j=\tilde{T}+1}^{T+\Delta} \eta_j\right)$$
$$\geq \exp(-6)\sigma^2 \sum_{\tilde{T}=T+1}^{T+\Delta} \eta_{\tilde{T}}^2 \geq \exp(-6)\cdot C_2\tau\sigma^2/(\lambda_i^2 T).$$

Here the first inequality just uses the second term in Equation (15) or (16), the second inequality is because $\sum_{j=\tilde{T}+1}^{T+\Delta} \eta_j \leq \sum_{j=T+1}^{T+\Delta} \eta_j \leq 3/\lambda_i$ and the last inequality is just based on the value of S^2. In this case as we can see as long as C_2 is large enough, $T+\Delta$ is also a point with $\mathbb{E}\left[f(\mathbf{w}_{T+\Delta})\right] - f(\mathbf{w}^*) \geq \lambda_i \mathbb{E}\left[\left(\mathbf{w}_{T+\Delta}^{(i)} - (\mathbf{w}^*)^{(i)}\right)^2\right] \geq C_1\tau\sigma^2/(T+\Delta)$, so we can repeat the argument there. Eventually we either stop because we hit case 1: $S^2 \leq C_2\tau/\lambda_i^2 T$ or the case 2 $S^2 > C_2\tau/\lambda_i^2 T$ happened more than $T/C_2\tau$ times. In either case we know for any $\tilde{T} \in [T, (1+1/C_2)T]$ $\mathbb{E}\left[f(\mathbf{w}_{\tilde{T}})\right] - f(\mathbf{w}^*) \geq \tau\sigma^2/T \geq \tau\sigma^2/\tilde{T}$ as the lemma claimed. □

Proof of Theorem 17. Theorem 17 is an immediate corollary of Theorem 15 and Lemma 18. □

Proof of Theorem 16. Theorem 16 is an immediate corollary of Theorem 17 and lemma 11 □

E Details of experimental setup

E.1 Synthetic 2-d Streaming Least Squares Experiments

As mentioned in the main paper, we consider four condition numbers namely $\kappa \in \{50, 100, 200, 400\}$. We run all experiments for a total of $\kappa_{\max}^2 = 400^2 = 160000$ iterations. The two eigenvalues of the Hessian are 1 and $1/\kappa$ respectively and the noise level $\sigma^2 = 1$ and we average our results with five random seeds. All our grid search results for the polynomially decaying learning rates are conducted on a 8×8 grid of learning rates $\times$ decay factor and whenever a best run lands at the edge of the grid, the grid is extended so that we have the best run in the interior of the grid search. For the step decay schedules, note that we fix the learning rate (details below), and vary only the decay factor.

For the $O(1/t)$ learning rate, we search for decay parameter over $8-$points log-spaced between $\{1/(200\kappa), 5000/\kappa\}$. The starting learning rate is searched over 8 points logarithmically spaced between $\{1/\kappa, 5\}$.

For the $O(1/\sqrt{t})$ learning rate, the decay parameter is searched over 8 logarithmically spaced points between $\{1/(2500\kappa), 100/\kappa\}$. The starting learning rate is searched between $\{1/(10\kappa), 5\}$ with 8 logarithmically spaced points.

For the step decay schedule experiments, we kept the initial learning rate to be 0.1 and swept over when to decay in multiples of $T/\log T$, i.e., vary some parameter $c \in \{0.25, 0.5, 0.75, 1.0, 1.25, 1.5, 2, 4\}$ where the learning rate decays by a factor of 2 every $c \cdot T/logT$ steps. We found that the values chosen in most experiments were very close to 1, i.e., they were either 1 or 1.25 or some very rare cases, 1.5.

With regards to the suffix iterate averaging, we used a constant stepsize of 0.1 and averaged iterates over the final half of the iterations.

E.2 Non-Convex experiments on cifar-10 dataset with a 44-layer residual net

As mentioned in the main paper, for all the experiments, we use the Nesterov's Accelerated gradient method (Nesterov, 1983) implemented in pytorch [6] with a momentum set to 0.9 and batchsize set to 128, total number of training epochs set to 100, ℓ_2 regularization set to 0.0005.

[6] https://github.com/pytorch

With regards to learning rates, we consider $10-$values geometrically spaced as $\{1, 0.6, \cdots, 0.01\}$. To set the decay factor for any of the schemes such as 5,6, and 7, we use the following rule. Suppose we have a desired learning rate that we wish to use towards the end of the optimization (say, something that is 100 times lower than the starting learning rate, which is a reasonable estimate of what is typically employed in practice), this can be used to obtain a decay factor for the corresponding decay scheme. In our case, we found it advantageous to use an additively spaced grid for the learning rate γ_t, i.e., one which is searched over a range $\{0.0001, 0.0002, \cdots, 0.0009, 0.001, \cdots, 0.009\}$ at the 80^{th} epoch, and cap off the minimum possible learning rate to be used to be 0.0001 to ensure that there is progress made by the optimization routine. For any of the experiments that yield the best performing gridsearch parameter that falls at the edge of the grid, we extend the grid to ensure that the finally chosen hyperparameter lies in the interior of the grid. All our gridsearches are run such that we separate a tenth of the training dataset as a validation set and train on the remaining $9/10^{th}$ dataset. Once the best grid search parameter is chosen, we train on the entire training dataset and evaluate on the test dataset and present the result of the final model (instead of choosing the best possible model found during the course of optimization).