Adaptive Batch sizes for SGD

Harshvardhan

January 24, 2021

1 Introduction

SGD(Robbins and Monro, 1951) has become the optimizer of choice for almost all large scale convex and even non-convex problems. Due to increased efficiency from parallelized compute resources and the remarkable efficiency of using batches and instead of single datapoints, most prefer mini-batch SGD. Inspite of its widespread usage, the presence of noise in SGD prevents the constant step size and constant batch size variant from reaching the optimal solution, and the final iterate error in this case is a sum of an exponentially decreasing bias term and a constant noise term(Needell et al., 2014). This has encouraged different step length strategies like average-SGD(Bach and Moulines, 2013), which guarantees optimal $O(\frac{\sigma^2}{n})$ convergence, where σ^2 is the noise variance and n is the number of iterations. Constant step size strategies which detect this convergence via some test are present in the literature like (Chee and Toulis, 2018) which utilizes Pflug's statistic, or (Pesme et al., 2020) which defines a new distance based diagnostic, which is the main attention of the second half of this paper.

While step length strategies for improving convergence are abundant, batch size control has not been investigated as thoroughly. The equivalence between step lengths and batch sizes have been discussed in some settings for deep networks, like for sharpest descent directions in (Jastrzebski et al., 2018), practitioners almost always have utilize their inverse relationship,i.e., higher batch sizes and smaller step lengths have similar effects. Establishing this equivalence for convergence-detection strategies is one of the main contributions of this paper. Optimal batch size strategies, like exponential increase covered in (Friedlander and Schmidt, 2012) and (Yu and Jin, 2019), are shown to be optimal for SGD as they attain $O(\frac{1}{n})$ convergence. However, hyperparameter selection still remains a key area of investigation for these strategies. (Gower et al., 2019) provides us with a theoretically-optimal batch size ensuring minimum possible total computational cost for the finite-sum setting of SGD. (Alfarra et al., 2020) explores an adaptive algorithm which learns this constant optimal batch size for the problem. Most other works on batch size strategies heavily incorporate heuristics with scarce theoretical grounding, like (Zhao et al., 2020).

Our work from the previous semester(Harshvardhan, 2020) dealt with the extension of (Gower et al., 2019) to yield theoretically optimal batch size per iteration. We use this work as the starting point. This work includes extension of previous work to special cases of constant step size and constant batch size. We show that we are able to obtain exponentially increasing batch size and convergence detection in these strategies, which have been used in existing literature to obtain optimal convergence results. These serve as motivation for the analysis of convergence-diagnostic tests of (Pesme et al., 2020), its extension to the batch size control, and its analysis in low noise cases where it performs poorly. We also define another convergence diagnostic, gradNorm test, which performs as good as the (Pesme et al., 2020) for most cases, and especially better in very low noise settings. Finally, we conduct experiments to analyze better support our theoretical observations.

In the next sections, we define the problem settings and assumptions required to prove our theoretical results. Then, we discuss the notable results from (*Goweret al.*, 2019) and the previous semester's work(Harshvardhan, 2020) which have been extended here. All the subsequent sections contain work carried out in this semester.

2 Problem Setting

The optimization problem in the finite-sum settings –

$$\mathbf{x}^\star = \operatorname*{arg\,min}_{\mathbf{x}\in\mathcal{X}} \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) \tag{1}$$

where $f_i\colon \mathbb{R}^n \to \mathbb{R}$ is the value of the objective function evaluated at the i^{th} data point and the optimizers $\mathbf{x} \in \mathcal{X}$. The whole dataset contains n datapoints.

We will now define the sampling operations to be used for SGD wrt a sampling vector $\mathbf{v}$ sampled from a distribution $\mathcal{D}$.

Definition 1. *A random vector* $\mathbf{v}$ *sampled from a distribution* $\mathcal{D}$ *is called a sampling vector if* $\mathbb{E}_{\mathcal{D}}\,\mathbf{v}_i = 1, \forall i \in [n]$.

Incorporating the sampling of datapoints in the optimization problem results in the following form –

$$\mathbf{x}^\star = \operatorname*{arg\,min}_{\mathbf{x}\in\mathcal{X}} \mathbb{E}_{\mathcal{D}}\, f_{\mathbf{v}}(\mathbf{x}) \tag{2}$$

$$\text{where } f_{\mathbf{v}}(\mathbf{x}) \coloneqq \frac{1}{n}\sum_{i=`1}^{n} \mathbf{v}_i f_i(\mathbf{x}) \tag{3}$$

Therefore,

$$\begin{aligned} \mathbb{E}_{\mathcal{D}}\, f_{\mathbf{v}}(\mathbf{x}) &= \frac{1}{n}\sum_{i=`1}^{n} \mathbb{E}_{\mathcal{D}}\,\mathbf{v}_i f_i(\mathbf{x}) &(4)\\ &= \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) &(5)\\ &\coloneqq f(\mathbf{x}) &(6)\end{aligned}$$

Thus, Definition 1 ensures that solving (2) solves (1) on expectation. This allows us to specify multiple sampling distributions for our algorithms by specifying distributions for v_i. Further, v_i values are used later for computing smoothness and noise gradient wrt the sampling scheme.

Similarly, we define the stochastic version of the gradient as

$$\begin{aligned} \nabla f_{\mathbf{v}}(\mathbf{x}) &= \frac{1}{n}\sum_{i=1}^{n} \mathbf{v}_i \nabla f_i(\mathbf{x}) &(7)\\ \mathbb{E}_{\mathcal{D}}\,\nabla f_{\mathbf{v}}(\mathbf{x}) &= \nabla f(\mathbf{x}) &(8)\end{aligned}$$

The algorithm used for optimization is SGD whose update equations are given below – where $\mathbf{x}_t$ is the iterate, γ_t is the step length and $\mathbf{v}_t \overset{i.i.d}{\sim} \mathcal{D}$ is the sampling vector at time instant t. In all our analysis, we take $\mathcal{X} = \mathbb{R}^d$

3 Assumptions

These assumptions are required for the convergence analysis for SGD.

Algorithm 1 SGD

```
Initialize $\mathbf{x}_0$
for t = 1 to T do
    Sample $\mathbf{v}_t \sim \mathcal{D}$
    $\mathbf{x}_{t+1} = \mathbf{x}_t - \gamma_t \nabla f_{\mathbf{v}_t}(\mathbf{x}_t)$
end for
```

Assumption 1. *f has a unique global minimizer $\mathbf{x}^\star \in \mathbb{R}^d$.*

Assumption 2. *f is μ-strongly quasi-convex, i.e.,*

$$f(\mathbf{x}^\star) \geq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{x}^\star - \mathbf{x} \rangle + \frac{\mu}{2} \|\mathbf{x} - \mathbf{x}^\star\|^2 \tag{9}$$

for $\mathbf{x} \in \mathbb{R}^d$ and $\mathbf{x}^\star$ being the global minimizer for f.

Assumption 3. *f is $\mathcal{L}$-smooth in expectation with respect to the distribution $\mathcal{D}$, i.e.,*

$$\mathbb{E}_{\mathcal{D}}\left[\|\nabla f_{\mathbf{v}}(\mathbf{x}) - \nabla f_{\mathbf{v}}(\mathbf{x}^\star)\|^2\right] \leq 2\mathcal{L}(f(\mathbf{x}) - f(\mathbf{x}^\star)), \forall \mathbf{x} \in \mathbb{R}^d \tag{10}$$

This is concisely represented as $(f, \mathcal{D}) \sim ES(\mathcal{L})$.

Assumption 4. *f has finite gradient($\sigma^2(f, \mathcal{D})$) noise wrt the sampling distribution $\mathcal{D}$*

$$\sigma^2 := \mathbb{E}_{\mathcal{D}}\left[\|\nabla f_{\mathbf{v}}(\mathbf{x}^\star)\|^2\right] < \infty \tag{11}$$

As a consequence of these assumptions, we state the following lemma which is directly used in the convergence analysis.

Lemma 1. *For f satisfying Assumptions* (3) *and* (4)*,*

$$\mathbb{E}_{\mathcal{D}}\Big[\|\nabla f_{\mathbf{v}}(\mathbf{x})\|^2\Big] \leq 4\mathcal{L}(f(\mathbf{x}) - f(\mathbf{x}^\star)) + 2\sigma^2 \tag{12}$$

We will use more assumptions when we discuss various forms of the distribution $\mathcal{D}$ and their corresponding $\mathcal{L}$ and σ values. Note that the smoothness and noise variance definitions are now closely dependent on the sampling scheme. We state another assumption for the smoothness of the individual functions f_i.

Assumption 5. *Each f_i is convex and $\mathbf{M}_i$-smooth, where each $\mathbf{M}_i \in \mathbb{R}^{d \times d}$ is a symmetric positive definite matrix. This can be stated as –*

$$f_i(\mathbf{x} + \mathbf{h}) \leq f_i(\mathbf{x}) + \langle \nabla f_i(\mathbf{x}), h \rangle + \frac{1}{2} \|\mathbf{h}\|^2_{\mathbf{M}_i} \tag{13}$$

for all $\mathbf{x}, \mathbf{h} \in \mathbb{R}^d$ and $i \in [n]$, where $\|\mathbf{h}\|^2_{\mathbf{M}_i} = \langle \mathbf{M}_i \mathbf{h}, \mathbf{h} \rangle$

We define the terms $L := \frac{1}{n}\lambda_{max}\big(\sum_{i=1}^n \mathbf{M}_i\big), L_{\max} := \max_i \lambda_{max}(\mathbf{M}_i), L_C = \frac{1}{|C|}\lambda_{max}\big(\sum_{i \in C} \mathbf{M}_i\big)$, where $C \subseteq [n]$

4 Optimal batch size and optimal batch size per iteration

In this section, we state, without proof, some results about the optimal batch sizes per iteration, which consist of the first part of our work. We first establish a bound on iterate error for SGD consisting of the bias and variance terms.

Theorem 2. *For f satisfying Assumptions* $(1), (2), (3)$ *and* (4)*, with a constant step length* $\gamma_t = \gamma \in (0, \frac{1}{2\mathcal{L}}]$ *,with* $\mathbf{x}_t$ *being the iterates obtained from SGD, the following inequality holds for all* $t \geq 0$ *–*

$$\mathbb{E}_{\mathcal{D}}\Big[\|\mathbf{x}_{t+1} - \mathbf{x}^\star\|^2\Big] \leq (1 - \gamma\mu)\,\mathbb{E}_{\mathcal{D}}\Big[\|\mathbf{x}_t - \mathbf{x}^\star\|^2\Big] + 2\gamma^2\sigma^2 \tag{14}$$

Proof. Let $\mathcal{F}_t$ be the normal filtration defined until iteration t.

$$\begin{aligned}\mathbb{E}_{\mathcal{D}}\Big[\|\mathbf{x}_{t+1} - \mathbf{x}^\star\|^2 \,|\mathcal{F}_t\Big] &\leq \|\mathbf{x}_t - \mathbf{x}^\star\|^2 + \gamma^2\,\mathbb{E}_{\mathcal{D}}\Big[\|\nabla f_{\mathbf{v}_t}(\mathbf{x}_t)\|^2 \,|\mathcal{F}_t\Big] - 2\gamma\,\mathbb{E}_{\mathcal{D}}[\langle \nabla f_{\mathbf{v}_t}(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}^\star\rangle \,|\mathcal{F}_t] \qquad (15)\\ &\leq \|\mathbf{x}_t - \mathbf{x}^\star\|^2 + \gamma^2\,\mathbb{E}_{\mathcal{D}}\Big[\|\nabla f_{\mathbf{v}_t}(\mathbf{x}_t)\|^2 \,|\mathcal{F}_t\Big] - 2\gamma\,\langle \nabla f(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}^\star\rangle \qquad (16)\end{aligned}$$

Using Lemma 1 and μ quasi convexity

$$\leq \|\mathbf{x}_t - \mathbf{x}^\star\|^2 - 2\gamma\left(f(\mathbf{x}_t) - f(\mathbf{x}^\star) + \frac{\mu}{2}\|\mathbf{x}_t - \mathbf{x}^\star\|^2\right) + \gamma^2\left(4\mathcal{L}(f(\mathbf{x}_t) - f(\mathbf{x}^\star)) + 2\sigma^2\right) \tag{17}$$

Taking expectation on both sides

$$\mathbb{E}_{\mathcal{D}}\Big[\|\mathbf{x}_{t+1} - \mathbf{x}^\star\|^2\Big] \leq (1 - \gamma\mu)\,\mathbb{E}_{\mathcal{D}}\Big[\|\mathbf{x}_t - \mathbf{x}^\star\|^2\Big] + 2\gamma(2\gamma\mathcal{L} - 1)\,\mathbb{E}_{\mathcal{D}}[f(\mathbf{x}) - f(\mathbf{x}^\star)] + 2\gamma^2\sigma^2 \tag{18}$$

Assuming $\gamma \leq \frac{1}{2\mathcal{L}}$

$$\mathbb{E}_{\mathcal{D}}\Big[\|\mathbf{x}_{t+1} - \mathbf{x}^\star\|^2\Big] \leq (1 - \gamma\mu)\,\mathbb{E}_{\mathcal{D}}\Big[\|\mathbf{x}_t - \mathbf{x}^\star\|^2\Big] + 2\gamma^2\sigma^2 \tag{19}$$

□

Since we have a per iteration optimal batch size, we define two new metrics, which take into account the batch size. These are the total complexity and progress per computation.

Definition 2. *For an SGD optimization*(1) *running for* k *iterations with batch size* τ_i *in* i^{th} *iteration to achieve final iterate error* ϵ*, we define total complexity* $T^\star(\epsilon)$ *as –*

$$T^\star(\epsilon) = \sum_{i=1}^{k} \tau_i \tag{20}$$

Definition 3. *We define average reduction per computation* $\mathcal{E}^\star(\tau)$ *for an SGD step with initial and final iterate errors* r_i *and* r_{i+1} *and batch size* τ *as*

$$\mathcal{E}^\star(\tau) = \frac{r_i - r_{i+1}}{\tau} \tag{21}$$

The constant optimal batch size ($\tau^\star$), the corresponding number of iterations to achieve error ϵ ($k^\star$) and the total complexity ($T^\star(\epsilon)$)for (Gower et al., 2019).

$$\tau^\star = n\frac{A_h - L_{\max}}{A_h - L_{\max} + nL} \tag{22}$$

$$k^\star = \frac{2A_hL}{(A_h - L_{max})\mu}\log\left(\frac{2\|\mathbf{x}_0 - \mathbf{x}^\star\|^2}{\epsilon}\right) \tag{23}$$

$$T^\star(\epsilon) = \frac{2A_hnL}{\mu(A_h - L_{\max} + nL)}\log\left(\frac{2\|\mathbf{x}_0 - \mathbf{x}^\star\|^2}{\epsilon}\right) \tag{24}$$

where $A_h = \frac{2}{\mu\epsilon}\bar{h}$. The first important observation is that an optimal batch size exists only when $A_h \geq L_{max}$ or

$$\epsilon \leq \frac{2\bar{h}}{\mu L_{max}} \tag{25}$$

When this condition is not satisfied, or when the above optimal batch size is less than 1, the batch size is taken to be 1.

4.1 Optimal batch size per iteration

For the optimal batch size per iteration, we minimize the upper bound for a single step of SGD.

$$\min_{\gamma \in \mathbb{R}_+} (1 - \gamma_1 \mu) r_0 + 2\gamma_1^2 \sigma^2 \tag{26}$$

This inequality is obtained from (19) which forces $\gamma \leq \frac{1}{2\mathcal{L}}$.

We define r_k as an upper bound on the iterate error after k iterations. We obtain the following value for optimal batch size for the first iteration by minimizing $\mathcal{E}^\star(\tau)$ for the first iteration.

$$\tau_0^\star = n \frac{\frac{2\bar{h}}{\mu r_0} - L_{\max}}{\frac{2\bar{h}}{\mu r_0} - L_{\max} + nL} \tag{27}$$

For this batch size choice, the upper bound on the next iterate error (r_1) is set as the minima of the above optimization problem. The expressions for r_1 and the optimal step size is –

$$r_1 = (1 - \frac{\mu}{4L}) r_0 + \frac{L_{max}\mu^2}{8\bar{h}L} r_0^2 \tag{28}$$

$$\gamma_1^\star = \frac{2\bar{h} - L_{max} r_0 \mu}{4\bar{h}L} \tag{29}$$

By using this batch size scheme, we end up with the following theorem bounding

Theorem 3. *For f satisfying Assumptions* (1), (2), (3) *and* (4)*, with batch size in each iteration defined by* (27) *and the corresponding step length defined by $\gamma_i = \frac{1}{2\mathcal{L}_i}$, where $\mathcal{L}_i$ is the expected smoothness constant for the batch size τ_i, with the initial error r_0 satisfying the condition in* (25)*, final iterate error of ϵ is achieved in $k^\star$ iterations*

$$k^\star \geq \frac{8\bar{h}L}{2\bar{h}\mu - L_{max}\mu^2 r_0} \log\left(\frac{r_0}{\epsilon}\right) \tag{30}$$

$$T^\star(\epsilon) = \sum_{i=0}^{k^\star - 1} n \frac{\frac{2\bar{h}}{\mu r_i} - L_{\max}}{\frac{2\bar{h}}{\mu r_i} - L_{\max} + nL} \tag{31}$$

where $A_{r_k} = \frac{2\bar{h}}{\mu r_k}$

Comparing the per iteration optimal batch size and the constant optimal batch size, we obtain distinct regions where different methods would be suitable.

Let T_1, T_2 be the the total complexity respectively, for obtaining final error ϵ using constant optimal batch size and per iteration optimal respectively.

Theorem 4. *The per iteration optimal batch size selection scheme is better than the constant optimal batch size for all iterations conditions Lemma* **??** *is satisfied and –*

- *If $nL \geq L_{max}$ –*

$$\frac{1}{\epsilon} \leq \frac{L(nL - L_{max})(1 - D)}{D\bar{h}\left(\frac{L_{max}}{A_{r_0}} + \frac{nL}{A_{r_0} - L_{\max} + nL}\right)} + \frac{\mu(L_{max} - nL)}{2\bar{h}} \tag{32}$$

- *If $nL \leq L_{max}$ –*

$$\frac{1}{\epsilon} \leq \frac{L(L_{max} - nL)(1 - D)}{DL_{max} r_0} \left(\frac{Dn}{(1 - D)(L_{max} - nL)} + \frac{2}{\mu}\right) + \frac{\mu(L_{max} - nL)}{2\bar{h}} \tag{33}$$

where $D = \frac{2\bar{h}L(4L - \mu) + L_{max}\mu^2 r_0}{8\bar{h}L}$

4.2 Implementation Details for Optimal Batch size per iteration

Note that the step length(28) and optimal batch(27) size per iteration depend on the iterate errors in each step, which are not so readily available during the execution of the algorithm. For this purpose, we will use a recursion between the step lengths, batch sizes and errors. Consider $\gamma_i, \tau_i, r_{i-1}$ as the step length, batch size and the error at the start of the i^{th} iteration. Then, from equations (28) and (27),

$$r_{i-1} = \frac{2\bar{h}(1-2\gamma_i L)}{\mu L_{max}} \tag{34}$$

$$\tau_i = \frac{2\gamma_i n L_{max}}{2\gamma_i L_{max} + n(1-2\gamma_i L)} \tag{35}$$

Using (28)

$$r_i = \frac{\bar{h}(1-2\gamma_i L)}{L_{max}}\left[\frac{2}{\mu} - \gamma_i\right] \tag{36}$$

$$\implies \gamma_{i+1} = \gamma_i\left[\frac{1}{4L} + \frac{1}{\mu} - \frac{\gamma_i \mu}{2L}\right] \tag{37}$$

This gives us an update scheme in terms of only the step length which is much easier to compute if we know a valid initial step length and batch size. Note that the initial iterate error is $r_0 \leq \frac{2\bar{h}}{\mu L_{max}}$. Before stepping into this regime, our algorithm advocates using batch size 1. Thus, we can run SGD with batch size 1 until we get to a sufficiently small iterate error(r_0). Given the starting point of the algorithm, we can compute the number of iterations required to achieve this. After achieving r_0 convergence, we can compute the step length and batch size for the first iteration of the variable batch size scheme using equations (28) and (27). For the subsequent iterations, we use the recursive relation between step lengths and batch sizes. We keep doing this until we reach the ϵ cutoff defined in Theorem 4. If the final iterate error requirement is better than this cutoff, we choose the cutoff as the initial error and run mini-batch SGD with constant optimal batch size. Additionally, if we assume that the optimal solution lies in a ball of radius R, our initial error is bounded by 2R. However, there is one caveat in our implementation. We need the value of r_0 to define the upper bound in Theorem 4, which we do not know. Assuming $r_0 = \frac{2\bar{h}}{\mu L_{max}} - \delta$, for a small $\delta > 0$ can be used as a very rough approximation.

5 Corner Cases

Our implementation for the optimal batch size per iteration is not of much practical value without the knowledge of several theoretical properties of the problem like the Lipschitz, convexity constants and noise variance, which are in general, not available for most problems. To alleviate this problem, we use the optimal batch per iteration techniques for the special cases – constant batch size and constant step size.

5.1 Constant Step Size

Assume that the step size γ is constant for all the iterations and we have control over only the batch size τ for each iteration. For such a setting the average reduction per computation, for the i^{th} iteration, becomes –

$$\mathcal{E}^{\star}(\tau) = \frac{\gamma\mu r_i}{\tau} - \frac{2\gamma^2\sigma^2}{\tau} \tag{38}$$

$$= \frac{1}{\tau}\left(\gamma\mu r_i + \frac{2\gamma^2\bar{h}}{n}\right) - \frac{2\gamma^2\bar{h}}{\tau^2} \tag{39}$$

Note that to maximize $\mathcal{E}^\star(\tau)$ wrt τ, we apply first order optimality conditions

$$\frac{d\mathcal{E}^\star(\tau)}{d\tau} = \frac{-1}{\tau^2}\left(\gamma\mu r_i + \frac{2\gamma^2\bar{h}}{n}\right) - \frac{4\gamma^2\bar{h}}{\tau^3} = 0 \tag{40}$$

$$\tau^\star = \left(\frac{\mu r_i}{4\bar{h}\gamma} + \frac{1}{2n}\right)^{-1} \tag{41}$$

$$\frac{d^2\mathcal{E}^\star(\tau^\star)}{d\tau^2} = -\gamma\left(\mu r_i + \frac{2\gamma\bar{h}}{n}\right)\left(\frac{\mu r_i}{4\bar{h}\gamma} + \frac{1}{2n}\right)^3 < 0 \tag{42}$$

5.1.1 Feasbility

We will now investigate the conditions for this solution to be feasible. The first restriction is on the step length γ. We need $\gamma \leq \frac{1}{2\mathcal{L}}$ for all values of τ obtained by our scheme. Thus, $\gamma \leq \frac{1}{2\mathcal{L}_{max}}$. $\mathcal{L}$ is a decreasing function of τ and thus attains its maxima at $\tau = 1$, assume the expected smoothness constant corresponding to this is $\mathcal{L}_1$.

The second feasibility condition is on the batch size. Note that the batch size $1 \leq \tau^\star \leq n$. Thus,

$$\frac{1}{n} \leq \frac{1}{\tau^\star} \leq 1 \tag{43}$$

$$\frac{1}{n} \leq \left(\frac{\mu r_i}{4\bar{h}\gamma} + \frac{1}{2n}\right) \leq 1 \tag{44}$$

$$\frac{2\bar{h}\gamma}{\mu n} \leq r_i \leq \frac{4\bar{h}\gamma}{\mu}\left(1 - \frac{1}{2n}\right) \tag{45}$$

Thus, we can use this constant step length and batch size strategy when the iterate error lies in the given region. For $r_i \geq \frac{4\bar{h}\gamma}{\mu}\left(1 - \frac{1}{2n}\right)$, batch size is 1 and for $r_i \leq \frac{2\bar{h}\gamma}{\mu n}$, batch size is n.

5.1.2 Convergence rates

We will now analyze convergence of mini-batch *SGD* for this batch size strategy. Further, we assume that the initial and final iterate errors (r_0 and ϵ respectively), lie in the feasible region described above.

Consider the i^{th} step in SGD,

$$r_{i+1} \leq (1 - \gamma\mu)r_i + 2\gamma^2\sigma^2 \tag{46}$$

$$r_{i+1} \leq (1 - \gamma\mu)r_i + 2\gamma^2\bar{h}\left(\frac{1}{\tau^\star} - \frac{1}{n}\right) \tag{47}$$

Substituting the value of $\tau^\star$

$$r_{i+1} \leq (1 - \gamma\mu)r_i + 2\gamma^2\bar{h}\left(\frac{\mu r_i}{4\gamma\bar{h}} - \frac{1}{2n}\right) \tag{48}$$

$$r_{i+1} \leq (1 - \frac{\gamma\mu}{2})r_i - \frac{\gamma^2\bar{h}}{n} \tag{49}$$

Summing from $i = 0$ to $k - 1$

$$r_k \leq (1 - \frac{\gamma\mu}{2})^k r_0 - \frac{\gamma^2\bar{h}}{n}\sum_{i=0}^{k-1}(1 - \frac{\gamma\mu}{2})^i \tag{50}$$

$$r_k \leq (1 - \frac{\gamma\mu}{2})^k r_0 \tag{51}$$

To achieve ϵ final error, the number of iterations

$$k \geq \frac{2}{\gamma\mu} \log\left(\frac{r_0}{\epsilon}\right) \tag{52}$$

$$\tag{53}$$

Thus, the optimal batch size that we obtain is of the form –

$$\tau_k^\star = \left(\frac{\mu r_0 (1 - \frac{\gamma\mu}{2})^k}{4\bar{h}\gamma} + \frac{1}{2n}\right)^{-1} \tag{54}$$

$$\tag{55}$$

This is an exponentially increasing batch size. (Yu and Jin, 2019) use a similar strategy for increasing batch size exponentially in the distributed settings although they do not show an optimal value for the rate of this increase. While the terms of L and L_{max} no longer appear in our batch size updates, we still require the values of $\bar{h}, \mu, r_0$ for this scheme, making it difficult for practical implementations.

5.2 Constant Batch Size

In this section, we assume that the batch size is constant and we have control over only the step length. Thus, both $\mathcal{L}$ and σ^2 are fixed. Then, maximizing the average reduction per computation is same as minimizing the single step iterate error. Then, for the i^{th} SGD step

$$\min_{\gamma_i \in \left(0, \frac{1}{2\mathcal{L}}\right)} \quad (1 - \gamma_i\mu) r_i + 2\gamma_i^2\sigma^2 \tag{56}$$

This is a quadratic in γ and is minimized at $\gamma_i^\star = \min\left\{\frac{1}{2\mathcal{L}}, \frac{\mu r_i}{4\sigma^2}\right\}$. This is a minima of a constant term and a term dependent on the iterate error. The iterate error when the two terms inside the minima is exactly equal to the iterate error when the optimal batch size per iteration is the constant batch size τ. The step length strategy consists of 2 regions–

1. Stage 1 : $r_i \geq \frac{2\bar{h}(\tau-1)}{\mu((\tau-1)L_{max}+nL\tau)}$. We use a constant step length $\gamma_i = \frac{1}{2\mathcal{L}}$ for this stage. For a constant step length, the convergence is linear in this region.

$$r_{i+1} \leq \left(1 - \frac{\mu}{2\mathcal{L}}\right) r_i + \frac{\sigma^2}{2\mathcal{L}^2} \tag{57}$$

   Summing over $i = 0$ to $k - 1$

$$r_k \leq \left(1 - \frac{\mu}{2\mathcal{L}}\right)^k r_0 + \frac{\sigma^2}{\mu\mathcal{L}} \tag{58}$$

2. Stage 2 : $r_i \leq \frac{2\bar{h}(\tau-1)}{\mu((\tau-1)L_{max}+nL\tau)}$ In this region, $\gamma_i = \frac{\mu r_i}{4\sigma^2}$, which is a decreasing step length. The i^{th} iteration in this region then becomes –

$$r_{i+1} \leq \left(1 - \frac{\mu r_i}{8\sigma^2}\right) r_i \tag{59}$$

This quadratic does not have a simple closed form solution for r_k in terms of r_0, however, we can use the worst case approximation to obtain an upper bound.

$$r_{i+1} \leq \left(1 - \frac{\mu\epsilon}{8\sigma^2}\right) r_i \tag{60}$$

Iterating from $i = 0$ to $k - 1$

$$r_k \leq \left(1 - \frac{\mu\epsilon}{8\sigma^2}\right)^k r_0 \tag{61}$$

An important outcome of this analysis is the switching point between stages 1 and 2. This switch happens when the bias and variance terms are equal. As SGD with constant step size gets stuck when this requirement is fulfilled, our algorithm, like others in literature notably (Pesme et al., 2020) and (Chee and Toulis, 2018), where diagnostic tests are used to detect this convergence. With a constant step length, no further progress is made in the iterate error, after this convergence is reached, so the step length is decreased. We utilize this result as motivation for the investigation of (Pesme et al., 2020).

6 Using convergence-diagnostics for updating batch sizes

The SGD algorithm(1) can be divided into two phases, the transient phase when the error decreases exponentially because of decrease in the bias($(1 - \gamma\mu)^k r_0$), and the saturation phase, when the iterate error is of the same magnitude as the noise variance term ($\frac{2\gamma\sigma^2}{\mu}$). Upon reaching the saturation phase, the iterates keep oscillating about the minima, due to large noise in the gradient steps. To alleviate this problem, (Chee and Toulis, 2018) proposes the use of Pflug's statistic (Ermoliev and Wets, 1988) to detect convergence. (Pesme et al., 2020) shows the large variance in Pflug's statistic and proposes a new diagnostic based on distance from the first iterate. After detecting convergence, (Pesme et al., 2020) proposes decreasing step size exponentially. This strategy shown to work theoretically for quadratic objectives and experimentally for most common objective functions. These strategies can be divided into 2 steps –

- Detect Convergence : (Pesme et al., 2020) uses distance-based diagnostic for this purpose. This distance based diagnostic analyzes the rate of increase of $\mathbb{E}\left[\|\mathbf{x}_0 - \mathbf{x}_t\|^2\right]$, whose behaviour can be shown to be exactly opposite to that of $\mathbb{E}\left[\|\mathbf{x}_t - \mathbf{x}^\star\|^2\right]$ for quadratic objectives. We later propose another statistic for this purpose based on the gradient norm, which shows exactly same behaviour as $\mathbb{E}\left[\|\mathbf{x}_t - \mathbf{x}^\star\|^2\right]$ for all strongly convex and smooth objectives.

- Decrease Variance term after detection: Since the bottleneck in iterate error is introduced due to the constant variance term $\frac{\sigma^2\gamma}{\mu}$, decreasing this term exponentially after every detection allows us to decrease the iterate error even further. Additionally, exponential decrease ensures that any algorithm which detects convergence correctly, achieves $O(\frac{\sigma^2}{n})$ optimal error, where n is the effective number of computations. This is best possible error attainable for SGD(Bach and Moulines, 2013). (Pesme et al., 2020) decreases step sizes exponentially, however, increasing batch sizes should have the same effect on the variance term. We first prove this theoretically and later verify this via experiments on common problems.

We first define our diagnostic-based SGD algorithm which increases batch size exponentially.

We now prove that Algorithm 2 achieves $O(\frac{\sigma^2}{n})$ error for the final iterate, where n is the total number of computations. A result of this form has been established for exponentially decreasing step length algorithm in (Pesme et al., 2020).

Algorithm 2 Convergence-Diagnostic SGD : ExpBatch

```
Require: Starting point $\mathbf{x}_0$, step size $\gamma$, batch size increase $r > 1$, initial batch size $\tau_0$, Num of datapoints n
  $\tau \leftarrow \tau_0$
  for t = 1 to T do
    Sample $\mathbf{v}_t \sim \mathcal{D}_\tau$
    {$\mathcal{D}_\tau$ is the independent sampling distribution with expected batch size $\tau$}
    $\mathbf{x}_{t+1} = \mathbf{x}_t - \gamma_t \nabla f_{\mathbf{v}_t}(\mathbf{x}_t)$
    if Convergence Diagnostic then
      $\tau \leftarrow \left( \frac{1}{r\tau} + \frac{1}{n}\left(1 - \frac{1}{r}\right) \right)^{-1}$
    end if
  end for
```

Algorithm 3 Convergence-Diagnostic Oracle

```
Require: $\gamma, \mu, \sigma_\tau^2, T, r_0$
  Bias $\leftarrow (1 - \mu\gamma)^T r_0$
  Variance $\leftarrow \frac{2\gamma\sigma_\tau^2}{\mu}$
  return {$Bias < Variance$}
```

Proposition 1. *Under Assumptions (1) to (5), algorithm 2 instantiated with algorithm 3, with $r > 1$ and $\gamma \in (0, \frac{1}{2\mathcal{L}_{\tau_0}})$, where $\mathcal{L}_{\tau_0}$ is the expected smoothness constant for batch size τ_0, $r_0 = \|\mathbf{x}_0 - \mathbf{x}^\star\|^2$ and $\Delta t_1 = \frac{1}{\gamma\mu} \log\left(\frac{\mu r_0}{2\gamma\sigma_{\tau_0}^2}\right)$, we have for all $t \leq \Delta t_1$ –*

$$\mathbb{E}\left[\|\mathbf{x}_t - \mathbf{x}^\star\|^2\right] \leq (1 - \gamma\mu)^t r_0 + \frac{2\gamma\sigma_{\tau_0}^2}{\mu} \tag{62}$$

and for all $t > \Delta t_1$–

$$\mathbb{E}\left[\|\mathbf{x}_t - \mathbf{x}^\star\|^2\right] \leq \frac{8\sigma_{\tau_0}^2\gamma}{r^{\frac{(t-\Delta t_1)\mu\gamma}{\log(2r)}}\mu} \tag{63}$$

Further, to achieve an error ϵ, when $\epsilon \leq \frac{4\sigma_{\tau_0}^2\gamma}{\mu}$, the total gradient evaluations (T), required are –

$$\begin{aligned} T \geq& \frac{\log(2r)\tau_0 r}{\log(r)\gamma\mu(r-1)}\left(\frac{8\sigma_{\tau_0}^2\gamma}{\mu\epsilon} - 1\right) + \frac{\tau_0}{\gamma\mu}\log\frac{2r_0}{\epsilon} \\ T \geq& O(\frac{\sigma_{\tau_0}^2}{\epsilon}) + O(\log\frac{2r_0}{\epsilon}) \end{aligned} \tag{64}$$

Proof. Let t_k be the number of iterations until the k^{th} restart and $\Delta t_k = t_k - t_{k-1}$. Then, $n_k = \sum_{k'=1}^{k} \Delta n_{k'}$ and let $r_t = \mathbb{E}\left[\|\mathbf{x}_t - \mathbf{x}^\star\|^2\right]$ and let $\sigma_{\tau_k}^2, \tau_k$ be the noise variance and the batch sizes used from the k^{th} to $(k+1)^{th}$ restart. Before the first restart, i.e., $t \leq t_1$

$$r_t \leq (1 - \gamma\mu)^t r_0 + \frac{2\gamma\sigma_{\tau_0}^2}{\mu} \tag{65}$$

The first restart is achieved when both the terms in RHS of above equation become equal, thus, $\Delta t_1 =$

$\frac{1}{\gamma\mu}\log\left(\frac{\mu r_0}{2\gamma\sigma^2_{\tau_0}}\right)$. Further, $r_{t_1} \leq \frac{4\gamma\sigma_0^2}{\mu}$ After the first restart, i.e., $t \geq \Delta t_1$, assume $t_k \leq t \leq t_{k+1}$, then-

$$r_t \leq (1-\gamma\mu)^{t-t_k} r_{t_k} + \frac{2\gamma\sigma^2_{\tau_k}}{\mu} \tag{66}$$

Oracle restarts when the two terms on RHS are equal, thus

$$\Delta t_{k+1} = \frac{1}{\gamma\mu}\log\left(\frac{\mu r_{t_k}}{2\gamma\sigma^2_{\tau_k}}\right) \tag{67}$$

Using $r_{t_k} \leq \frac{4\sigma^2_{\tau_{k-1}}\gamma}{\mu}$

$$\leq \frac{1}{\gamma\mu}\log\left(\frac{\sigma^2_{\tau_{k-1}}}{\sigma^2_{\tau_k}}\right) \tag{68}$$

From the update equation of τ, and the definition of σ^2 in (4), we obtain $\sigma^2_{\tau_k} = \frac{\sigma^2_{\tau_{k-1}}}{r}$

$$\Delta t_{k+1} \leq \frac{1}{\gamma\mu}\log(2r) \tag{69}$$

As $t_k = \Delta_{t_1} + \sum_{k'=2}^{k} \Delta t_{k'}$

$$t_k - \Delta t_1 = \sum_{k'=2}^{k} \Delta t_{k'} \leq \sum_{k'=2}^{k} \frac{1}{\gamma\mu}\log(2r) \tag{70}$$

$$\leq \frac{k-1}{\gamma\mu}\log(2r) \tag{71}$$

$$k-1 \geq \frac{(t_k - \Delta t_1)\gamma\mu}{\log(2r)} \tag{72}$$

Since, $r_{t_k} \leq \frac{4\sigma^2_{\tau_{k-1}}\gamma}{\mu}$

$$r_{t_k} \leq \frac{4\gamma\sigma^2_{\tau_0}}{r^{\frac{(t_k-\Delta t_1)\mu\gamma}{\log(2r)}}\mu} \tag{73}$$

We only need to extend this result from t_k to a general $t \geq t_k$. Let $A : \mathbb{R} \to \mathbb{R}$, such that $A(t) = \frac{4\gamma\sigma^2_{\tau_0}}{r^{\frac{t\mu\gamma}{\log(2r)}}\mu}$. A is an exponentially decreasing function in t. Consider two functions – $G(t) = (1-\gamma\mu)^{t-t_k} A(t_k - \Delta t_1) + \frac{2\gamma\sigma^2_{\tau_k}}{\mu}$ and $H(t) = A(t - \Delta t_1) + \frac{2\gamma\sigma^2_{\tau_k}}{\mu}$. Then, $H(t_k) = G(t_k)$. As both functions are exponentially decreasing, we need to check the sign of their gradients at t_k.

$$G'(t_k) = \log(1-\gamma\mu)A(t-t_k) \geq -\gamma\mu \tag{74}$$

$$H'(t_k) = \frac{-\log(r)\gamma\mu}{\log(2r)} \tag{75}$$

Thus. $H'(t_k) \geq G'(t_k)$, and $H(t) \geq G(t), \forall t \geq t_k$. Thus, for $t_k \leq t \leq t_{k+1}$,

$$r_t \leq G(t_k) \leq H(t_k) \tag{76}$$

$$\leq A(t - \Delta t_1) + \frac{2\gamma\sigma^2_{\tau_k}}{\mu} \tag{77}$$

$$\leq A(t - \Delta t_1) + \frac{4\gamma\sigma^2_{\tau_k}}{\mu} \tag{78}$$

By construction, $\frac{4\gamma\sigma^2_{\tau_k}}{\mu} \leq A(t_{k+1} - \Delta t_1)$, but $A(t_{k+1} - \Delta t_1) < A(t - \Delta t_1)$, for $t \leq t_{k+1}$. Thus, $\frac{4\gamma\sigma^2_{\tau_k}}{\mu} \leq A(t - \Delta t_1)$. Therefore, for any general t –

$$r_t \leq 2A(t - \Delta t_1) \tag{79}$$

Now, we compute the number of gradient computations required for reaching ϵ error. We consider the case when atleast one restart has occurred, i.e., $\epsilon \leq \frac{4\sigma^2_{\tau_0}\gamma}{\mu}$. Let t be the number of iterations to reach error ϵ and let $t_k \leq t_{k+1}$,

$$t - \Delta t_1 \geq \frac{\log(2r)}{\log(r)\gamma\mu} \left(\log\left(\frac{8\sigma^2_{\tau_0}\gamma}{\mu\epsilon}\right) \right. \tag{80}$$

$$k \geq \left\lfloor \frac{1}{\log(r)} \log\left(\frac{8\sigma^2_{\tau_0}\gamma}{\mu\epsilon}\right) \right\rfloor \tag{81}$$

Thus, the total number of computations(N) required to reach ϵ error

$$T \geq \frac{\log(2r)\tau_0}{\log(r)\gamma\mu} \sum_{k'=2}^{k+1} r^{k'-1} + \tau_0 \Delta t_1 \tag{82}$$

$$\geq \frac{\log(2r)\tau_0 r}{\log(r)\gamma\mu(r-1)}(r^{k+1} - 1) + \tau_0 \Delta t_1 \tag{83}$$

Substituting the values of k and Δt_1

$$T \geq \frac{\log(2r)\tau_0 r}{\log(r)\gamma\mu(r-1)}(\exp\left\lceil \log\left(\frac{8\sigma^2_{\tau_0}\gamma}{\mu\epsilon}\right) \right\rceil - 1) + \frac{\tau_0}{\gamma\mu} \log \frac{\mu r_0}{2\sigma^2_{\tau_0}\gamma} \tag{84}$$

$$T \geq \frac{\log(2r)\tau_0 r}{\log(r)\gamma\mu(r-1)} \left(\frac{8\sigma^2_{\tau_0}\gamma}{\mu\epsilon} - 1 \right) + \frac{\tau_0}{\gamma\mu} \log \frac{2r_0}{\epsilon} \tag{85}$$

□

6.1 Further analysis of Convergence Detection for (Pesme et al., 2020)

In this section, we analyze the distance based metric in (*Pesmeet al.*, 2020). This metric detects convergence for quadratic objectives. To illustrate this point and investigate certain cases where this may fail, we consider $f(x) = \frac{1}{2}x^2$.We set initial point x_0 and noise level σ^2 and step length γ, borrowing results from Corollary 15 of (Pesme et al., 2020), we obtain–

Algorithm 4 Distance Based metric(Pesme et al., 2020)

```
Require: $\mathbf{x}_0, \mathbf{x}_t, \mathbf{x}_{t/q}, q > 1, thresh \in (0, 2]$
  if $n == q^k$ for k in $\mathbb{N}^\star$ then
    $S \leftarrow \frac{\log\|\mathbf{x}_t - \mathbf{x}_0\|^2 - \log\left\|\mathbf{x}_{t/q} - \mathbf{x}_0\right\|^2}{\log(t) - \log(t/q)}$
    return $S < thresh$
  else
    return False
  end if
```

$$|x_t - x_0|^2 = (1 - (1-\gamma)^t)^2 x_0^2 + \frac{\gamma\sigma^2}{2-\gamma}(1 - (1-\gamma)^{2t}) \tag{86}$$

For $t \to \infty$

$$|x_\infty - x_0|^2 = x_0^2 + \frac{\gamma\sigma^2}{2-\gamma} \tag{87}$$

For very small γt, using Taylor's expansion

$$|x_t - x_0|^2 = x_0^2\gamma^2 t^2 + \frac{\gamma^2\sigma^2}{2-\gamma}t + o((t\gamma)^2) \tag{88}$$

Distance-based diagnostic measures the changeover from large t which is of the form t^2 to constant $|x_\infty - x_0|^2$, by measuring when the slope of $|x_\infty - x_0|^2$ v/s t, becomes smaller than a threshold less than 1, in loglog scale. This test functions properly when the two terms in (86) to be comparable. This depends on the values of σ^2 and r_0. After the first restart, r_0^2 for the next tests becomes $O(\gamma)$ and thus, even the remaining tests perform well. The main problem with this test are the cases when the first positive test is obtained too far from the optimal. This can be observed when $\sigma^2 <<< r_0^2$. If the first test occurs at x_{t_1} then, if $r_{t_1}^2 = o(\gamma)$, the successive tests become erroneous and we decrease the step length too early. To better illustrate this, we plot (86) for different σ^2.

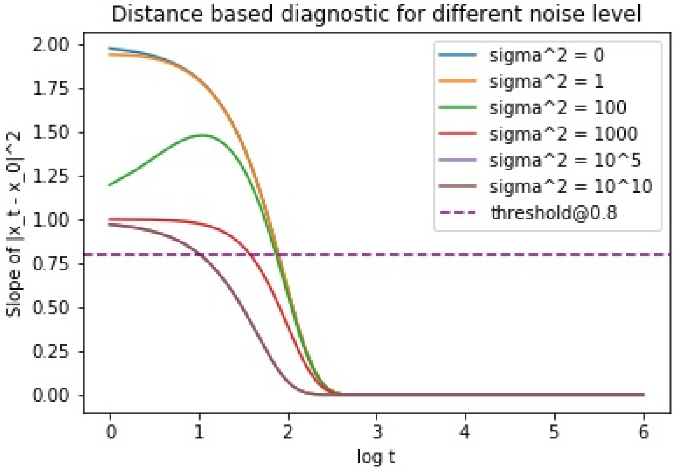


Figure 1: Slope of $|x_t - x_0|^2$ for different noise levels

For $\sigma^2 = 0, 1$, the exponential behaviour of $(1-(1-\gamma)^t)^2$ dominates so the slope decreases from 2 to 0. For $\sigma^2 = 100$, the two terms in (86) are comparable so the slope is not strictly decreasing, but the noise term still does not dominate. For $\sigma^2 = 1000, 10^5, 10^10$, the noise term dominates and the slope decreases from 1 to 0. Thus, for the same threshold, we should expect large separation between the time required to reach this threshold, however, this separation is very small. In the ideal scenario, if Scott's test were to measure bias and variance, then increasing σ^2 by 100, would imply that we would need to increase the number of iterations by 100 times. According to the experiments, if only one of the 2 terms (noise or bias) starts to dominate in (86), which is mostly observed except for few special cases, the number of iterations for achieving the same threshold does not change.

6.2 Gradient Norm Test

To overcome problems of (Pesme et al., 2020) for low noise settings, we consider a test based on gradient norms. We first establish the theoretical basis for our test. Consider $\mathbb{E}\left[\|\mathbf{x}_{t+1} - \mathbf{x}_t\|^2\right]$

$$\mathbb{E}\left[\|\mathbf{x}_{t+1} - \mathbf{x}_t\|^2\right] = \gamma_t^2\,\mathbb{E}\left[\|\nabla f_{\mathbf{v}}(\mathbf{x}_t)\|^2\right] \tag{89}$$

Using Lemma 1

$$\mathbb{E}\left[\|\mathbf{x}_{t+1}-\mathbf{x}_t\|^2\right] \leq 4\gamma_t^2\mathcal{L}(f(\mathbf{x}_t)-f(\mathbf{x}^\star))+2\gamma_t^2\sigma^2 \tag{90}$$

$$= 4\gamma_t^2\mathcal{L}\,\mathbb{E}\left[f_{\mathbf{v}}(\mathbf{x}_t)-f_{\mathbf{v}}(\mathbf{x}^\star)\right]+2\gamma_t^2\sigma^2 \tag{91}$$

Since f is $\mathcal{L}$-Lipschitz

$$\mathbb{E}\left[\|\mathbf{x}_{t+1}-\mathbf{x}_t\|^2\right] \leq 2\gamma_t^2\mathcal{L}^2\|\mathbf{x}_t-\mathbf{x}^\star\|^2+2\gamma_t^2\sigma^2 \tag{92}$$

$$\frac{1}{\gamma_t^2}\mathbb{E}\left[\|\mathbf{x}_{t+1}-\mathbf{x}_t\|^2\right] \leq 2\mathcal{L}^2\|\mathbf{x}_0\|^2(1-\gamma\mu)^t+2\sigma^2\left(1+\frac{\mathcal{L}^2\gamma}{\mu}\right) \tag{93}$$

Thus, $\frac{1}{\gamma_t^2}\mathbb{E}\left[\|\mathbf{x}_{t+1}-\mathbf{x}_t\|^2\right]$ is upper bounded by an exponentially decreasing term and a constant term related to the noise variance. Identifying convergence or when bias is smaller than variance, boils down to identifying when the rate of decrease of this term is no longer exponential.

The algorithm for a test based on this statistic is shown in (5)

Algorithm 5 GradNorm Based metric

```
Require: $\mathbf{x}_t, \mathbf{x}_{t+1}, \mathbf{x}_{t/q+1}, \mathbf{x}_{t/q}, q > 1, thresh, \gamma_t, \gamma_{t/q}$
  if $n == q^k$ for k in $\mathbb{N}^\star$ then
    $S \leftarrow 2\frac{(\log\|\mathbf{x}_{t/q+1}-\mathbf{x}_{t/q}\|-\log\|\mathbf{x}_{t+1}-\mathbf{x}_t\|)+(\log\gamma_t-\log\gamma_{t/q})}{\log(t)-\log(t/q)}$
    return $S < thresh$
  else
    return False
  end if
```

Note that in the cases of small noise ($\sigma^2 = 0$), gradient norm is upper bounded by an exponentially decreasing term only and thus the rate of decrease should always stay exponential and we should perform better than the distance based diagnostic.

7 Experiments

We perform experiments on synthetic data for least squares regression and logistic regression for 4 Algorithms – 3 choices each for Convergence Test (Distance based, Gradient Norm based, Gradient Norm over window) and variance term updates (step size decrease, batch size increase). The gradient norm over window corresponds to Algorithm 5 by taking average over a window of size 4 for the slope.

7.1 Least Squares

We generate $n = 10^6$ datapoints iid with dimension $d = 20$ from $\mathcal{N}(0, H)$, where H has eigenvalues $(\frac{1}{k}), k \in [20]$ and a randomly selected orthogonal matrix for its eigenvectors. The outputs y_i are sampled as $y_i =< \mathbf{x}_i, \mathbf{w}^\star > +\varepsilon_i$, where ε_i are sampled iid from $\mathcal{N}(0, 1)$ The objective function is $f(\mathbf{w}) = \frac{1}{2}\mathbb{E}\left[\|y_i- < \mathbf{w}, \mathbf{x}_i >\|^2\right]$ and the initial step length for all cases is $\gamma = \frac{1}{2\text{Tr H}}$ and initial batch size 10. Further, for Algorithm 4, $q = 2$, while $q = 2.5$ for Algorithm 5. The threshold for both tests is fixed at 1 and batch size and step size are increased and decreased respectively by a factor of $r = 4$. Further, for each algorithm, we give a burn-in time of 64 iterations before the first test.

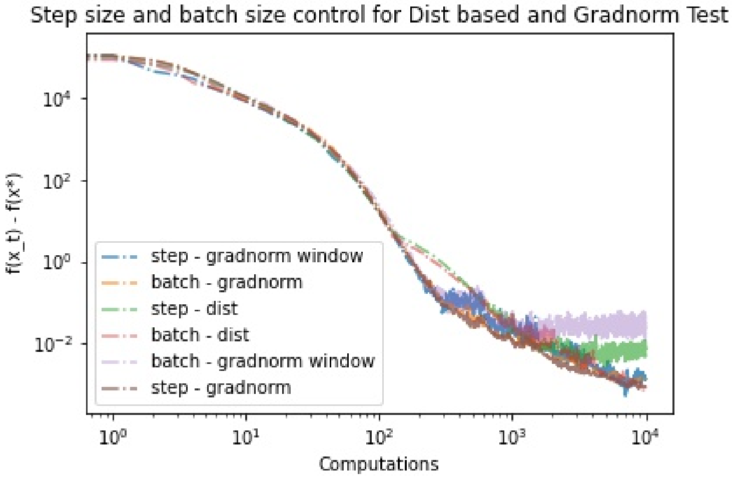


Figure 2: Error for Least squares

We find that batch size increase with gradnorm test window and distance based test with step size increase perform worse than all other cases which perform similarly.

7.2 Logistic Regression

We generate datapoints similar to previous case. The outputs are sampled from logistic model and the objective function is $f(\mathbf{w}) = \mathbb{E}\left[\log(1 + \exp(-y_i < \mathbf{w}, \mathbf{x}_i >)\right]$. Further, we set a burn-in time of 200 iterations with threshold 1 and $r = 2$. For distance based diagnostic, $q = 1.5$ while $q = 2$ for the gradnorm test and the step size used is $\gamma = \frac{4}{\text{Tr H}}$.

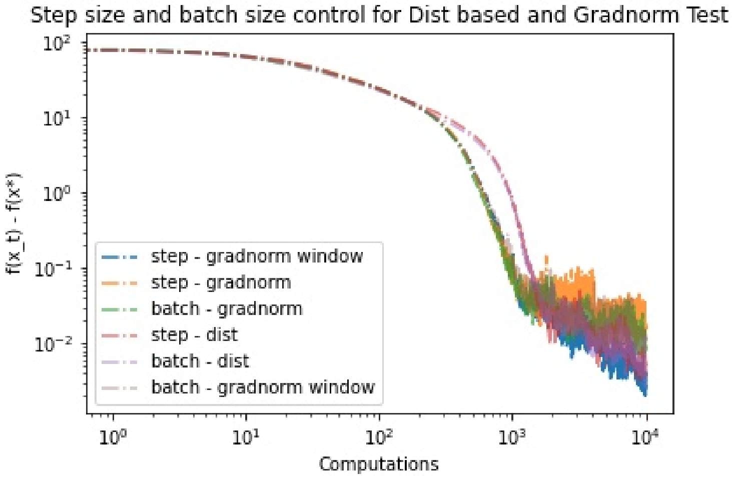


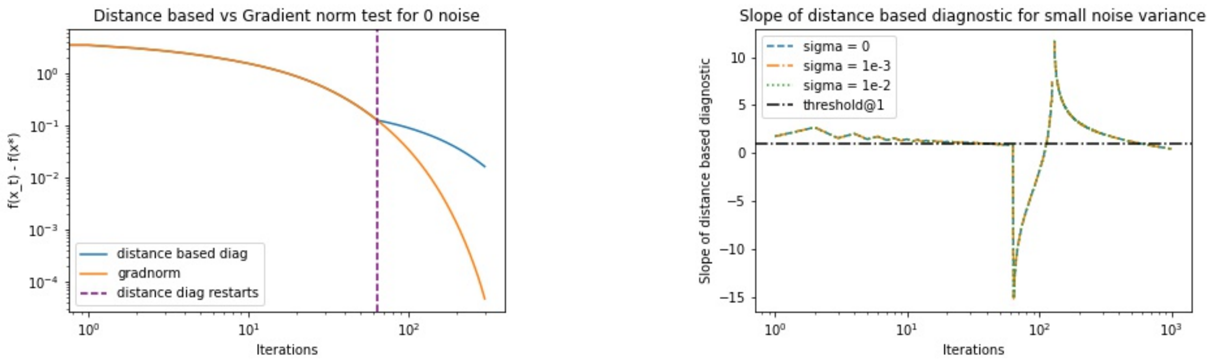
Figure 3: Error for Logistic Regression

Note that it is very difficult to gauge which algorithm performs the best from the given figure, due to large noise in the later iterations. However, taking into account the noise, we find that all cases perform equally well.

7.3 Distance based diagnostic for low noise settings

In this section, we take a small sample of datapoints generated for least squares experiment and perform SGD with complete gradient along with an additional noise term sampled iid from $\mathcal{N}(0, \sigma^2)$. In this case, we put $\sigma^2 = 0$ and compare the two diagnostics controlling only the step length. Burn-in time for both is 50 iterations in these experiments. Further, we vary $\sigma \in \{0, 10^{-2}, 10^{-3}\}$ and plot the distance based statistic and its corresponding threshold.

The distance based diagnostic restarts for the no noise case as well and thus, performs worse than our gradnorm based statistic. Further, for very small noise levels, the plots for the distance based statistic overlap which validates our hypothesis that for small noise levels, the distance based statistic is indifferent to the noise levels and all restarts are governed by only the bias term in (86).



(a) Distance based test v/s Gradient Norm Test for no noise

(b) Slope of statistic for different noise levels - Distance based diagnostic

Figure 4: Analysis of Distance based Diagnostic:Least Squares

8 Conclusion

By extending the analysis of optimal batch size per iteration from previous semester, to corner cases of constant step size and constant batch size, we obtained theoretical results corresponding to those in existing literature, like the exponentially increasing batch size for constant step length and decreasing step length after convergence for constant batch size. Further, we have tried to utilize the equivalence between step length and batch size to extend exponential decrease in step size from (Pesme et al., 2020) to exponential increase in batch size obtaining the same asymptotic convergence for the final iterate error. We analyzed the poor performance of distance based convergence diagnostic in (Pesme et al., 2020) for low noise levels and proposed a different gradient norm based convergence diagnostic. Our gradient norm based diagnostic was able to show similar if not better performance than distance based diagnostic for both step length control and batch size control for Linear and Logistic regression on synthetic data. We were, however, not able to find any significant difference between the step length and batch size control strategies, both in theoretical and experimental results, which further goes to prove their equivalence. The advantage of step length over batch size or vice versa is still highly dependent on the specifications of the training machines and it's analysis for different problem parameters is a direction which we could pursue.

Some more directions which we were not able to pursue were the theoretical proof of the distance based statistic for strongly convex strongly smooth functions, comparison of gradient norm and distance based statistic for non-convex problems, better implementation for the per iteration optimal batch size regime or its corner cases and obtaining theoretically sound hyperparameter values for the two tests.

References

Motasem Alfarra, Slavomir Hanzely, Alyazeed Albasyoni, Bernard Ghanem, and Peter Richtarik. Adaptive learning of the optimal mini-batch size of sgd, 2020.

Francis Bach and Eric Moulines. Non-strongly-convex smooth stochastic approximation with convergence rate o(1/n), 2013.

Jerry Chee and Panos Toulis. Convergence diagnostics for stochastic gradient descent with constant learning rate. In Amos Storkey and Fernando Perez-Cruz, editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 1476–1485, Playa Blanca, Lanzarote, Canary Islands, 09–11 Apr 2018. PMLR.

Y.M. Ermoliev and R.J.-B. Wets. *Numerical Techniques for Stochastic Optimization.* Springer-Verlag, Heidelberg, 1988.

Michael P. Friedlander and Mark Schmidt. Hybrid deterministic-stochastic methods for data fitting. *SIAM Journal on Scientific Computing*, 34(3):A1380–A1405, Jan 2012. ISSN 1095-7197. doi: 10.1137/110830629.

Robert Mansel Gower, Nicolas Loizou, Xun Qian, Alibek Sailanbayev, Egor Shulgin, and Peter Richtárik. SGD: General analysis and improved rates. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5200–5209, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

Harshvardhan. Analysis of optimal batch sizes for sgd. *Semester Project EPFL*, 2020.

Stanisław Jastrzebski, Zachary Kenton, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. On the relation between the sharpest directions of dnn loss and the sgd step length, 2018.

Deanna Needell, Rachel Ward, and Nati Srebro. Stochastic gradient descent, weighted sampling, and the randomized kaczmarz algorithm. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 1017–1025. Curran Associates, Inc., 2014.

Scott Pesme, Aymeric Dieuleveut, and Nicolas Flammarion. On convergence-diagnostic based step sizes for stochastic gradient descent. 2020.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *Ann. Math. Statist.*, 22(3): 400–407, 09 1951. doi: 10.1214/aoms/1177729586.

Hao Yu and Rong Jin. On the computation and communication complexity of parallel SGD with dynamic batch sizes for stochastic non-convex optimization. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 7174–7183. PMLR, 2019.

Shen-Yi Zhao, Yin-Peng Xie, and Wu-Jun Li. Stagewise enlargement of batch size for sgd-based learning, 2020.

Optimal Batch size per iteration

Harshvardhan

24 Feb 2020

1 Problem Setting

The optimization problem in the finite-sum settings –

$$\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathcal{X}} \frac{1}{n}\sum_{i=‘1}^{n} f_i(\mathbf{x}) \tag{1}$$

where $f_i(\mathbf{x})$ is the value of the objective function evaluated at the i^{th} data point and $\mathcal{X}$ is the domain of all optimizers $\mathbf{x}$. The whole dataset contains n datapoints.

We will now define the sampling operations to be used for SGD wrt a sampling vector $\mathbf{v}$

Definition 1. *A random vector* $\mathbf{v}$ *sampled from a distribution* $\mathcal{D}$ *is called a sampling vector if* $\mathbb{E}_{\mathcal{D}}\,\mathbf{v}_i = 1, \forall i \in [n]$.

Incorporating the sampling of datapoints in the optimization problem results in the following form –

$$\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathcal{X}} \mathbb{E}_{\mathcal{D}}\, f_{\mathbf{v}}(\mathbf{x}) \tag{2}$$

$$\text{where } f_{\mathbf{v}}(\mathbf{x}) \coloneqq \frac{1}{n}\sum_{i=‘1}^{n} \mathbf{v}_i f_i(\mathbf{x}) \tag{3}$$

Definition 1 ensures that solving (2) solves (1) on expectation.

Similarly, we define the stochastic version of the gradient as

$$\nabla f_{\mathbf{v}}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{v}_i \nabla f_i(\mathbf{x}) \tag{4}$$

$$\mathbb{E}_{\mathcal{D}}\, \nabla f_{\mathbf{v}}(\mathbf{x}) = \nabla f(\mathbf{x}) \tag{5}$$

The algorithm used for optimization is SGD whose update equations are given below –

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \gamma_t \nabla f_{\mathbf{v}_t}(\mathbf{x}_t) \tag{6}$$

where $\mathbf{x}_t$ is the iterate, γ_t is the step length and $\mathbf{v}_t \overset{i.i.d}{\sim} \mathcal{D}$ is the sampling vector at time instant t. In all our analysis, we take $\mathcal{X} = \mathbb{R}^d$

2 Assumptions

These assumptions are required for the convergence analysis for SGD.

Assumption 1. *f is* μ*-strongly quasi-convex, i.e.,*

$$f(\mathbf{x}^*) \geq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{x}^* - \mathbf{x}\rangle + \frac{\mu}{2}\|\mathbf{x} - \mathbf{x}^*\|^2 \tag{7}$$

Assumption 2. *f has a unique global minimizer $\mathbf{x}^* \in \mathbb{R}^d$.*

Assumption 3. *f is $\mathcal{L}$-smooth in expectation with respect to the distribution $\mathcal{D}$, i.e.,*

$$\mathbb{E}_{\mathcal{D}}\left[\|\nabla f_{\mathbf{v}}(\mathbf{x}) - \nabla f_{\mathbf{v}}(\mathbf{x}^*)\|^2\right] \le 2\mathcal{L}(f(\mathbf{x}) - f(\mathbf{x}^*)), \forall \mathbf{x} \in \mathbb{R}^d \tag{8}$$

This is concisely represented as $(f, \mathcal{D}) \sim ES(\mathcal{L})$.

Assumption 4.

$$\sigma^2 \coloneqq \mathbb{E}_{\mathcal{D}}\left[\|\nabla f_{\mathbf{v}}(\mathbf{x}^*)\|^2\right] < \infty \tag{9}$$

The gradient noise $\sigma(f, \mathcal{D})$ is finite.

As a consequence of these assumptions, we state the following lemma which is directly used in the convergence analysis.

Lemma 1. *For f satisfying Assumptions (3) and (4),*

$$\mathbb{E}_{\mathcal{D}}\Big[\|\nabla f_{\mathbf{v}}(\mathbf{x})\|^2\Big] \le 4\mathcal{L}(f(\mathbf{x}) - f(\mathbf{x}^*)) + 2\sigma^2 \tag{10}$$

We will use more assumptions when we discuss various forms of the distribution $\mathcal{D}$ and their corresponding $\mathcal{L}$ and σ values. Note that the smoothness and noise variance definitions are now closely dependent on the sampling scheme. We state another assumption for the smoothness of the individual functions f_i.

Assumption 5. *Each f_i is convex and $\mathbf{M}_i$-smooth, where each $\mathbf{M}_i \in \mathbb{R}^{d\times d}$ is a symmetric positive definite matrix. This can be stated as –*

$$f_i(\mathbf{x} + \mathbf{h}) \le f_i(\mathbf{x}) + \langle \nabla f_i(\mathbf{x}), h\rangle + \frac{1}{2}\|\mathbf{h}\|^2_{\mathbf{M}_i} \tag{11}$$

for all $\mathbf{x}, \mathbf{h} \in \mathbb{R}^d$ and $i \in [n]$, where $\|\mathbf{h}\|^2_{\mathbf{M}_i} = \langle \mathbf{M}_i\mathbf{h}, \mathbf{h}\rangle$

We define the terms $L \coloneqq \frac{1}{n}\lambda_{max}\big(\sum_{i=1}^n \mathbf{M}_i\big), L_{\max} \coloneqq \max_i \lambda_{max}(\mathbf{M}_i), L_C = \frac{1}{|C|}\lambda_{max}\big(\sum_{i\in C}\mathbf{M}_i\big)$, where $C \subseteq [n]$

3 Results used from (Gower et al., 2019)

3.1 Lemmas and Definitions

Lemma 2. *For independent sampling with uniform probabilities with expected batch size τ–*

1. *The expected smoothness constant is*

$$\mathcal{L} \le L + \left(\frac{1}{\tau} - \frac{1}{n}\right)L_{max} = L + G_\tau L_{\max} \tag{12}$$

2. *The gradient noise is*

$$\sigma^2 = \left(\frac{1}{\tau} - \frac{1}{n}\right)\bar{h} = G_\tau \bar{h} \tag{13}$$

where $\bar{h} = \frac{1}{n}\sum_{i\in[n]}\|\nabla f_i(\mathbf{x}^*)\|^2$ and $G_\tau = \frac{1}{\tau} - \frac{1}{n}$.

Definition 2. *For an SGD optimization running for k iterations with batch size τ_i in i^{th} iteration to achieve final iterate error ϵ, we define total complexity $T^*(\epsilon)$ as –*

$$T^*(\epsilon) = \sum_{i=1}^k \tau_i \tag{14}$$

3.2 SGD single step expression

Theorem 3. *For f satisfying Assumptions* $(2),(1),(3)$ *and* (4)*, with a constant step length* $\gamma_t = \gamma \in (0, \frac{1}{2\mathcal{L}}]$ *,with* $\mathbf{x}_t$ *being the iterates obtained from SGD, the following inequality holds for all* $t \geq 0$ *–*

$$\mathbb{E}_{\mathcal{D}}\Big[\|\mathbf{x}_{t+1} - \mathbf{x}^*\|^2\Big] \leq (1-\gamma\mu)\,\mathbb{E}_{\mathcal{D}}\Big[\|\mathbf{x}_t - \mathbf{x}^*\|^2\Big] + 2\gamma^2\sigma^2 \tag{15}$$

Proof. Let $\mathcal{F}_t$ be the normal filtration defined until iteration t.

$$\begin{aligned}\mathbb{E}_{\mathcal{D}}\Big[\|\mathbf{x}_{t+1} - \mathbf{x}^*\|^2\,|\mathcal{F}_t\Big] &\leq \|\mathbf{x}_t - \mathbf{x}^*\|^2 + \gamma^2\,\mathbb{E}_{\mathcal{D}}\Big[\|\nabla f_{\mathbf{v}_t}(\mathbf{x}_t)\|^2\,|\mathcal{F}_t\Big] - 2\gamma\,\mathbb{E}_{\mathcal{D}}[\langle\nabla f_{\mathbf{v}_t}(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}^*\rangle\,|\mathcal{F}_t] &(16)\\ &\leq \|\mathbf{x}_t - \mathbf{x}^*\|^2 + \gamma^2\,\mathbb{E}_{\mathcal{D}}\Big[\|\nabla f_{\mathbf{v}_t}(\mathbf{x}_t)\|^2\,|\mathcal{F}_t\Big] - 2\gamma\,\langle\nabla f(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}^*\rangle &(17)\\ &\leq \|\mathbf{x}_t - \mathbf{x}^*\|^2 - 2\gamma\left(f(\mathbf{x}_t) - f(\mathbf{x}^*) + \frac{\mu}{2}\|\mathbf{x}_t - \mathbf{x}^*\|^2\right) + \gamma^2\left(4\mathcal{L}(f(\mathbf{x}_t) - f(\mathbf{x}^*)) + 2\sigma^2\right) &(18)\end{aligned}$$

Taking expectation on both sides

$$\mathbb{E}_{\mathcal{D}}\Big[\|\mathbf{x}_{t+1} - \mathbf{x}^*\|^2\Big] \leq (1-\gamma\mu)\,\mathbb{E}_{\mathcal{D}}\Big[\|\mathbf{x}_t - \mathbf{x}^*\|^2\Big] + 2\gamma(2\gamma\mathcal{L} - 1)\,\mathbb{E}_{\mathcal{D}}[f(\mathbf{x}) - f(\mathbf{x}^*)] + 2\gamma^2\sigma^2 \tag{19}$$

Assuming $\gamma \leq \frac{1}{2\mathcal{L}}$

$$\mathbb{E}_{\mathcal{D}}\Big[\|\mathbf{x}_{t+1} - \mathbf{x}^*\|^2\Big] \leq (1-\gamma\mu)\,\mathbb{E}_{\mathcal{D}}\Big[\|\mathbf{x}_t - \mathbf{x}^*\|^2\Big] + 2\gamma^2\sigma^2 \tag{20}$$

□

In the proof, we utilise the strong smoothness(with expected smoothness constant) and $\mu - strong$ concavity of the objective function.

3.3 Batch size results

The optimal batch size (τ^*), the corresponding number of iterations to achieve error ϵ (k^*) and the total complexity ($T^*(\epsilon)$)for (Gower et al., 2019).

$$\tau^* = n\frac{A_h - L_{\max}}{A_h - L_{\max} + nL} \tag{21}$$

$$k^* = \frac{2A_hL}{(A_h - L_{max})\mu}\log\left(\frac{2\|\mathbf{x}_0 - \mathbf{x}^*\|^2}{\epsilon}\right) \tag{22}$$

$$T^*(\epsilon) = \frac{2A_hnL}{\mu(A_h - L_{\max} + nL)}\log\left(\frac{2\|\mathbf{x}_0 - \mathbf{x}^*\|^2}{\epsilon}\right) \tag{23}$$

where $A_h = \frac{2}{\mu\epsilon}\bar{h}$. The first important observation is that an optimal batch size exists only when $A_h \geq L_{max}$ or

$$\epsilon \leq \frac{2\bar{h}}{\mu L_{max}} \tag{24}$$

When this condition is not satisfied, or when the above optimal batch size is less than 1, the batch size is taken to be 1.

Additionally, differentiating the above terms wrt A_h, we get –

$$\frac{\partial\tau^*}{\partial A_h} = \frac{nL}{(A_h - L_{\max} + nL)^2} \tag{25}$$

$$k \tag{26}$$

$$\frac{\partial T^*(\epsilon)}{\partial A_h} = \frac{n^2L(nL - L_{\max})}{\mu(A_h - L_{\max} + nL)^2} \tag{27}$$

Thus, the optimal batch size increases with decrease in final iterate error. This provides motivation for the optimal batch size per iteration result.

4 Optimal Batch size per iteration

For the optimal batch size per iteration, we consider only a single step of SGD as our optimization problem.

$$\min_{\gamma \in \mathbb{R}_+} (1 - \gamma_1 \mu) r_0 + 2\gamma_1^2 \sigma^2 \tag{28}$$

This inequality is obtained from (20) which forces $\gamma \leq \frac{1}{2\mathcal{L}}$. Assume that r_k is an upper bound on the iterate error after k iterations. In the above optimization problem, we try to minimize r_1 to obtain a recursive relation for r_k in terms of r_{k-1}. Solving this recursion would yield the final iterate error r_k in terms of the initial error r_0.

We minimize the optimization problem (28) directly in terms of γ_1, as it is a quadratic in γ_1. We then impose conditions for selection of a valid step length γ_1 to obtain bounds for r_1, in terms of r_0 and the batch size. Since the objective is a quadratic in γ_1, applying first order conditions gives us the optimal γ_1

Differentiating wrt γ_1,

$$-\mu r_0 + 4\gamma_1 \sigma^2 = 0 \tag{29}$$

$$\implies \gamma_1 = \frac{\mu r_0}{4\sigma^2} \tag{30}$$

Since, we already have the condition that $\gamma_1 \leq \frac{1}{2\mathcal{L}}$, and the objective is a quadratic in γ, the minima is attained at the optimal, if it is in the domain or the end point. Thus, the optimal step length which minimizes the objective is given by –

$$\gamma_1^* = \min\left\{\frac{1}{2\mathcal{L}}, \frac{\mu r_0}{4\sigma^2}\right\} \tag{31}$$

Since this is a single step optimization, we need a metric to compare the quality of the two solutions for γ. Note that total complexity was able to compare the efficiency of two methods achieving the same error for different batch size regimes. In our case, since we consider only a single iteration, the total complexity would be just the batch size. But, this might be erroneous, as for a large batch size, we should ideally always get larger number of computations as well as a larger reduction in error in each step. We will now define a metric which takes into account this tradeoff.

Definition 3. *We define average reduction per computation $\mathcal{E}^*(\tau)$ for an SGD step with initial and final iterate errors r_i and r_{i+1} and batch size τ as*

$$\mathcal{E}^*(\tau) = \frac{r_i - r_{i+1}}{\tau} \tag{32}$$

The batch size choice which maximizes this metric should be the optimal batch size for the given iteration as it effectively does the largest amount of work per computation.

For the two cases of step lengths, average reduction per computation takes the form –

$$\mathcal{E}^*(\tau) = \min\left\{\frac{\mu r_0}{2\mathcal{L}\tau} - \frac{\sigma^2}{\tau \mathcal{L}^2}, \frac{\mu^2 r_0^2}{8\sigma^2 \tau}\right\} \tag{33}$$

The function containing $\mathcal{L}$ is an increasing function of τ while the function containing σ^2 is a decreasing function of τ for $r_0 \leq \frac{2\bar{h}}{\mu L_{max}}$. To maximize $\mathcal{E}^*$, both the terms inside the minima should be equal, which implies that both the terms in minima of (31) are equal. Thus, the optimal batch size to achieve this condition is

$$\tau^* = n \frac{\frac{2\bar{h}}{\mu r_0} - L_{\max}}{\frac{2\bar{h}}{\mu r_0} - L_{\max} + nL} \tag{34}$$

This is the same as the constant optimal batch size for all iterations, with $\epsilon = r_0$. We will use this observation later, when we compare the two methods. Again, we need the batch size to be positive so the following condition holds

$$r_0 \leq \frac{2\bar{h}}{\mu L_{\max}} \tag{35}$$

The increasing and decreasing nature of functions that we had discussed earlier also hold in this range.

For this batch size choice, the next iterate error and the optimal step size is –

$$r_1 = (1 - \frac{\mu}{4L})r_0 + \frac{L_{max}\mu^2}{8\bar{h}L}r_0^2 \tag{36}$$

$$\gamma_1^* = \frac{2\bar{h} - L_{max}r_0\mu}{4\bar{h}L} \tag{37}$$

We will now analyze the recursion for r_k.

Lemma 4. *If* $0 \leq r_0 \leq \frac{2\bar{h}}{\mu L_{\max}}$*, the sequence of positive* r_i*'s defined by the following recursion are strictly decreasing and converge to 0.*

$$r_{i+1} \geq (1 - \frac{\mu}{4L})r_i + \frac{L_{max}\mu^2}{8\bar{h}L}r_i^2 \tag{38}$$

Proof. If r_i satisfies the above condition,

$$r_{i+1} \leq r_i - \frac{\mu}{4L}\frac{2\bar{h}}{L_{max}} + \frac{L_{max}\mu^2}{8\bar{h}L}\frac{4\bar{h}^2}{L_{max}^2\mu^2} \leq r_i \tag{39}$$

Thus, if $r_0 \leq \frac{2\bar{h}}{L_{max}\mu}$, $r_1 \leq r_0 \leq \frac{2\bar{h}}{L_{max}\mu}$ and $r_i \leq \frac{2\bar{h}}{L_{max}\mu}, \forall i$. □

Thus if we operate in the above region, our optimal batch size per iteration procedure is decreasing and converges to zero. This is also the same region, where we can select a non-negative batch size.

We will now try to analyse multiple iterations of the above procedure until we reach the final iterate error ϵ.

Theorem 5. *For* f *satisfying Assumptions* (2), (1), (3) *and* (4)*, with batch size in each iteration defined by* (34) *and the corresponding step length defined by* $\gamma_i = \frac{1}{2\mathcal{L}_i}$*, with the initial error* r_0 *satisfying the condition in Lemma* 4*, final iterate error of* ϵ *is achieved in* k^* *iterations*

$$k^* \geq \frac{8\bar{h}L}{2\bar{h}\mu - L_{max}\mu^2 r_0}\log\left(\frac{r_0}{\epsilon}\right) \tag{40}$$

$$T^*(\epsilon) = \sum_{i=0}^{k^*-1} n\frac{\frac{2\bar{h}}{\mu r_i} - L_{\max}}{\frac{2\bar{h}}{\mu r_i} - L_{\max} + nL} \tag{41}$$

Proof. From Lemma 4

$$r_{i+1} \geq (1 - \frac{\mu}{4L})r_i + \frac{L_{max}\mu^2}{8\bar{h}L}r_i^2 \tag{42}$$

Since $r_i \leq r_0$, we can set r_{i+1} to following $\forall i$

$$r_{i+1} = \left(1 - \frac{\mu}{4L} + \frac{L_{max}\mu^2 r_0}{8\bar{h}L}\right)r_i \tag{43}$$

Iterating from $i = 0$ to $k^* - 1$

$$r_{k^*} = (1 - \frac{\mu}{4L} + \frac{L_{max}\mu^2 r_0}{8\bar{h}L}\Bigg)^k r_0 \tag{44}$$

$$\implies k^* \log\left(\frac{1}{1 - \frac{\mu}{4L} + \frac{L_{max}\mu^2 r_0}{8\bar{h}L}}\right) \geq \log\left(\frac{r_0}{\epsilon}\right) \tag{45}$$

$$k^* \geq \frac{8\bar{h}L}{2\bar{h}\mu - L_{max}\mu^2 r_0} \log\left(\frac{r_0}{\epsilon}\right) \tag{46}$$

$$\tag{47}$$

In the last inequality, we use the identity $\log\left(\frac{1}{\rho}\right) \geq 1 - \rho$ for $0 < \rho \leq 1$. $T^*(\epsilon)$ is simply sum of batch sizes till k^* iterations. □

The form of the batch size and number of iterations to reach ϵ error are very similar for (Gower et al., 2019) and the per iteration optimal, however, the total complexity, which is the main metric used for comparison, differs a lot.

5 Constant optimal v/s Per iteration optimal

Let τ_ϵ, k_1, T_1 be the optimal batch size, total number of iterations and the total complexity respectively, for obtaining ϵ in (Gower et al., 2019). Then, the per iteration optimal has batch sizes $\tau_{r_0}, \tau_{r_1}, \ldots, \tau_\epsilon$. Let k_2 and T_2 be its corresponding number of iterations and total complexity for obtaining ϵ final iterate error.

Then one can easily observe the following relationships –

$$\tau_{r_i} \leq \tau_\epsilon \forall r_i \geq \epsilon \tag{48}$$

$$k_2 \geq k_1 \tag{49}$$

The batch size is a decreasing function of iterate error as shown in previous sections. Thus, the per optimal method uses smaller batch sizes for each of its computations but uses approximately larger number of iterations to achieve the same error. Thus, in the per iteration optimal, with more iterations, the error decreases and the batch size for the iteration increases, and after achieving ϵ error, the optimal batch size for the next iteration is same as the constant optimal batch size used to get to that error. These inequalities, however, do not differentiate one of the two methods to be better than the other, so we still need to analyze the total complexity.

But the total complexity for iteration optimal does not have a closed form which we can use. In the remainder of this section, we try to find the conditions where choosing the batch size every iteration is more computationally efficient. For this, we will find an upper bound for T_2 and and a lower bound for T_1.

Lemma 6.

- *For $nL \geq L_{max}$ –*

$$T_2 \leq \frac{nD}{1-D} \cdot \frac{1}{nL - L_{max}} \left(L_{max} \left(\frac{A_\epsilon - A_{r_0}}{A_{r_0}} \right) + nL \left(\frac{A_\epsilon - A_{r_0}}{A_{r_0} - L_{\max} + nL} \right) \right) \tag{50}$$

- *For $L \leq L_{max}$*

$$T_2 \leq \frac{nD}{1-D} \cdot \frac{1}{L_{max} - nL} \left(L_{max} \left(\frac{A_\epsilon - A_{r_0}}{A_{r_0}} \right) - nL \left(\frac{A_\epsilon - A_{r_0}}{A_\epsilon - L_{\max} + nL} \right) \right) \tag{51}$$

where $A_r = \frac{2\bar{h}}{\mu r}$ and $D = \frac{2\bar{h}L(4L-\mu) + L_{max}\mu^2 r_0}{8\bar{h}L}$

Proof. T_2 can be expressed in terms of the function A_r.

$$T_2 = \sum_{i=0}^{k_2-1} n \frac{A_{r_i} - L_{\max}}{A_{r_i} - L_{\max} + nL} \tag{52}$$

$$= \sum_{i=0}^{k_2-1} n \frac{A_{r_i} - L_{\max}}{A_{r_i} - L_{\max} + nL} \frac{A_{r_{i+1}} - A_{r_i}}{A_{r_{i+1}} - A_{r_i}} \tag{53}$$

$$\leq \sum_{i=0}^{k_2-1} n \frac{A_{r_i} - L_{\max}}{A_{r_i} - L_{\max} + nL} \frac{A_{r_{i+1}} - A_{r_i}}{A_{r_i}(\frac{1}{D} - 1)} \tag{54}$$

where $D = \frac{2\bar{h}L(4L-\mu)+L_{max}\mu^2 r_0}{8\bar{h}L}$

$$\leq \frac{nD}{1-D}\left(\frac{1}{A_{ri}} - nL\frac{1}{A_{r_i}(A_{r_i} - L_{\max} + nL)}\right)(A_{r_{i+1}} - A_{r_i}) \tag{55}$$

$$\leq \frac{nD}{1-D}\left(\frac{1}{A_{ri}} - \frac{nL}{nL - L_{max}}\left(\frac{1}{A_{r_i}} - \frac{1}{(A_{r_i} - L_{\max} + nL)}\right)\right)(A_{r_{i+1}} - A_{r_i}) \tag{56}$$

$$\tag{57}$$

The term D depends only on r_0 and is always less than 1 so $1 - D$ is always positive Note that this is a discrete sum lower bound to the integral in terms of A_r.

$$T_2 \leq \frac{nD}{1-D}\int_{A_{r_0}}^{A_\epsilon} \frac{1}{nL - L_{max}}\left(\frac{L_{max}}{A_r} + \frac{nL}{A_r - L_{\max} + nL}\right)dA_r \tag{58}$$

$$\leq \frac{nD}{1-D} \cdot \frac{1}{nL - L_{max}}\left(L_{max}\log\left(\frac{A_\epsilon}{A_{r_0}}\right) + nL\log\left(\frac{A_\epsilon - L_{\max} + nL}{A_{r_0} - L_{\max} + nL}\right)\right) \tag{59}$$

Now, we consider the two cases –

- Case 1 : $nL \geq L_{max}$ Since $A_\epsilon > A_{r_0}$, we use the identity $\log(1 + x) \leq x$.

$$T_2 \leq \frac{nD}{1-D} \cdot \frac{1}{nL - L_{max}}\left(L_{max}\left(\frac{A_\epsilon - A_{r_0}}{A_{r_0}}\right) + nL\left(\frac{A_\epsilon - A_{r_0}}{A_{r_0} - L_{\max} + nL}\right)\right) \tag{60}$$

- Case 2 : $nL \leq L_{max}$

$$T_2 \leq \frac{nD}{1-D}\left(\frac{1}{A_{ri}} + \frac{nL}{L_{max} - nL}\left(\frac{1}{A_{r_i}} - \frac{1}{(A_{r_i} - L_{\max} + nL)}\right)\right)(A_{r_{i+1}} - A_{r_i}) \tag{61}$$

$$\leq \frac{nD}{1-D} \cdot \frac{1}{L_{max} - nL}\left(L_{max}\log\left(\frac{A_\epsilon}{A_{r_0}}\right) - nL\log\left(\frac{A_\epsilon - L_{\max} + nL}{A_{r_0} - L_{\max} + nL}\right)\right) \tag{62}$$

Here we use $\log(1 + x) \leq x$ and $\log(\frac{1}{\rho} \geq 1 - \rho, 0 < \rho \leq 1$

$$\leq \frac{nD}{1-D} \cdot \frac{1}{L_{max} - nL}\left(L_{max}\left(\frac{A_\epsilon - A_{r_0}}{A_{r_0}}\right) - nL\left(\frac{A_\epsilon - A_{r_0}}{A_\epsilon - L_{\max} + nL}\right)\right) \tag{63}$$

□

The next lemma lower bounds T_1

Lemma 7.

$$T_1 \leq \frac{2nL(A_\epsilon - A_{r_0})}{\mu(A_\epsilon - L_{max} + nL} \tag{64}$$

Proof.

$$\begin{aligned}
T_1 &\geq \frac{2A_\epsilon nL}{\mu(A_\epsilon - L_{\max} + nL)} \log\left(\frac{A_\epsilon}{A_{r_0}}\right) && (65)\\
&\geq \frac{2A_\epsilon nL}{\mu(A_\epsilon - L_{\max} + nL)} \left(\frac{A_\epsilon - A_{r_0}}{A_\epsilon}\right) && (66)\\
&\geq \frac{2nL(A_\epsilon - A_{r_0})}{\mu(A_\epsilon - L_{max} + nL} && (67)
\end{aligned}$$

The second inequality is obtained from identity $\log(\frac{1}{\rho}) \geq 1 - \rho$. □

To now obtain the region where the per iteration variant is better, we find conditions such that the lower bound of T_1 is larger than the upper bound of T_2. We use bounds that are both multiples of $(A_\epsilon - A_{r_0}$, so these terms cancel out.

Theorem 8. *The per iteration optimal batch size selection scheme is better than the constant optimal batch size for all iterations conditions Lemma 4 is satisfied and –*

- *If* $nL \geq L_{max}$ –

$$\frac{1}{\epsilon} \leq \frac{L(nL - L_{max})(1-D)}{D\bar{h}\left(\frac{L_{max}}{A_{r_0}} + \frac{nL}{A_{r_0} - L_{\max} + nL}\right)} + \frac{\mu(L_{max} - nL)}{2\bar{h}} \qquad (68)$$

- *If* $nL \leq L_{max}$ –

$$\frac{1}{\epsilon} \leq \frac{L(L_{max} - nL)(1-D)}{DL_{max}r_0}\left(\frac{Dn}{(1-D)(L_{max} - nL)} + \frac{2}{\mu}\right) + \frac{\mu(L_{max} - nL)}{2\bar{h}} \qquad (69)$$

Proof. By comparing the two bounds from Lemmas 6 and 7, we get –

Case 1 : $nL \geq L_{max}$.

$$\frac{D}{1-D} \cdot \frac{1}{nL - L_{max}}\left(\frac{L_{max}}{A_{r_0}} + \frac{nL}{A_{r_0} - L_{\max} + nL}\right) \leq \frac{2L}{\mu(A_\epsilon - L_{max} + nL} \qquad (70)$$

$$\implies \frac{1}{\epsilon} \leq \frac{L(nL - L_{max})(1-D)}{D\bar{h}\left(\frac{L_{max}}{A_{r_0}} + \frac{nL}{A_{r_0} - L_{\max} + nL}\right)} + \frac{\mu(L_{max} - nL)}{2\bar{h}} \qquad (71)$$

- Case 2 : $nL \leq L_{max}$

$$\begin{aligned}
&\frac{D}{1-D} \cdot \frac{1}{L_{max} - nL}\left(\frac{L_{max}}{A_{r_0}} - \frac{nL}{A_\epsilon - L_{\max} + nL}\right) \leq \frac{2L}{\mu(A_\epsilon - L_{max} + nL)} && (72)\\
\implies &\frac{D}{1-D} \cdot \frac{L_{max}}{(L_{max} - nL)A_{r_0}} \leq \frac{L}{\mu(A_\epsilon - L_{max} + nL)}\left(\frac{Dn}{(1-D)(L_{max} - nL)} + \frac{2}{\mu}\right) && (73)\\
\implies &\frac{1}{\epsilon} \leq \frac{L(L_{max} - nL)(1-D)}{DL_{max}r_0}\left(\frac{Dn}{(1-D)(L_{max} - nL)} + \frac{2}{\mu}\right) + \frac{\mu(L_{max} - nL)}{2\bar{h}} && (74)
\end{aligned}$$

□

These conditions show that until some small cutoff is reached the per iteration technique is better. If the desired error is still smaller than this cutoff, we use a constant batch size after the cutoff.

- Large Error($\geq \frac{2\bar{h}}{\mu L_{max}}$) : For this region, taking simple SGD steps (batch size =1) is the most optimal strategy. Being so far from the optimal, even with very noisy steps, we are able to get good reduction in the error per computation.
- Moderate Error : For this region, the optimal batch size should be chosen at every iteration. This strikes a balance between error reduction and computational cost and as the error decreases, we keep increasing the batch size to maintain this balance.
- Small Error : Here, the error is too small to maintain the balance in the previous region and we jump to a large batch size based on the final error to maximize our error reduction at the cost of computation.

6 Implementation Details for Optimal Batch size per iteration

Note that the step length(36) and optimal batch(34) size per iteration depend on the iterate errors in each step, which are not so readily available during the execution of the algorithm. For this purpose, we will use a recursion between the step lengths, batch sizes and errors. Consider $\gamma_i, \tau_i, r_{i-1}$ as the step length, batch size and the error at the start of the i^{th} iteration. Then, from equations (36) and (34),

$$r_{i-1} = \frac{2\bar{h}(1-2\gamma_i L)}{\mu L_{max}} \tag{75}$$

$$\tau_i = \frac{2\gamma_i n}{2\gamma_i + n(1-2\gamma_i L)} \tag{76}$$

Now, using the Lemma 4,

$$r_i = \frac{\bar{h}(1-2\gamma_i L)}{L_{max}}\Bigg[\frac{2}{\mu} - \gamma_i] \tag{77}$$

$$\implies \gamma_{i+1} = \gamma_i \Bigg[\frac{1}{4L} + \frac{1}{\mu} - \frac{\gamma_i \mu}{2L}] \tag{78}$$

This gives us an update scheme in terms of only the step length which is much easier to compute if we know a valid initial step length and batch size. Note that the initial iterate error is $r_0 \leq \frac{2\bar{h}}{\mu L_{max}}$. Before stepping into this regime, our algorithm advocates using batch size 1. Thus, we can run SGD with batch size 1 until we get to a sufficiently small iterate error(r_0). Given the starting point of the algorithm, we can compute the number of iterations required to achieve this. After achieving r_0 convergence, we can compute the step length and batch size for the first iteration of the variable batch size scheme using equations (36) and (34). For the subsequent iterations, we use the recursive relation between step lengths and batch sizes. We keep doing this until we reach the ϵ cutoff defined in Theorem 8. If the final iterate error requirement is better than this cutoff, we choose the cutoff as the initial error and run mini-batch SGD with constant optimal batch size. Additionally, if we assume that the optimal solution lies in a ball of radius R, our initial error is bounded by 2R.

References

Robert Mansel Gower, Nicolas Loizou, Xun Qian, Alibek Sailanbayev, Egor Shulgin, and Peter Richtárik. SGD: General analysis and improved rates. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5200–5209, Long Beach, California, USA, 09–15 Jun 2019. PMLR.