Ringmaster ASGD: The First Asynchronous SGD with Optimal Time Complexity

Artavazd Maranjyan

King Abdullah University of Science and Technology (KAUST)

arto.maranjyan@gmail.com

Instituto de Matemática Pura e Aplicada

Abstract

Asynchronous Stochastic Gradient Descent (Asynchronous SGD) is a cornerstone method for parallelizing learning in distributed machine learning. However, its performance suffers under arbitrarily heterogeneous computation times across workers, leading to suboptimal time complexity and inefficiency as the number of workers scales. While several Asynchronous SGD variants have been proposed, recent findings by [6] reveal that none achieve optimal time complexity, leaving a significant gap in the literature. In this paper, we propose Ringmaster ASGD, a novel Asynchronous SGD method designed to address these limitations and tame the inherent challenges of Asynchronous SGD. We establish, through rigorous theoretical analysis, that Ringmaster ASGD achieves optimal time complexity under arbitrarily heterogeneous and dynamically fluctuating worker computation times. This makes it the first Asynchronous SGD method to meet the theoretical lower bounds for time complexity in such scenarios.

Introduction

We consider stochastic nonconvex optimization problems of the form

$$\min_{x\in\mathbb{R}^d}\left\{f(x):=\mathbb{E}_{\xi\sim\mathcal{D}}\left[f(x;\xi)\right]\right\},$$

where $f:\mathbb{R}^d\times\mathbb{S}_\xi\to\mathbb{R}$, $\mathbb{R}^d$ is a linear space, and $\mathbb{S}_\xi$ is a sample space. In machine learning, $f(x;\xi)$ denotes the loss of a model parameterized by x on a data sample ξ, and $\mathcal{D}$ denotes the distribution of the training dataset. In nonconvex optimization, our goal is to find an ε–stationary point, i.e., a (random) vector $\bar{x}\in\mathbb{R}^d$ such that $\mathbb{E}[\|\nabla f(\bar{x})\|^2]\leq\varepsilon$.

We consider a setup involving n workers (e.g., CPUs, GPUs, servers), each with access to the same distribution $\mathcal{D}$. Each worker is capable of computing independent, unbiased stochastic gradients with bounded variance (Assumption 3). We consider a setup with asynchronous, heterogeneous, and varying computation speeds. We aim to account for all potential scenarios, such as random outages, varying computational performance over time, and the presence of slow or straggling workers [4].

This setup is common in both datacenter environments and federated learning [3] for distributed training. Although parallelism facilitates rapid convergence, variations in worker speeds make effective coordination more challenging.

Asynchronous Stochastic Gradient Descent (Asynchronous SGD) is a popular approach for parallelization in such distributed settings. Despite the variety of Asynchronous SGD algorithms proposed over the years, a fundamental question remained unresolved: *What is the optimal strategy for parallelization in this setting?*

In this work, we answer this question affirmatively. We reestablish the prominence of Asynchronous SGD by proposing a novel asynchronous optimization method that attains optimal time complexity.

Problem Setup

To compare methods, we consider the *fixed computation model* [5]. In this model, it is assumed that

$$\text{worker } i \text{ takes no more than } \tau_i \text{ seconds to compute a single stochastic gradient.} \tag{1}$$

and

$$0<\tau_1\leq\tau_2\leq\cdots\leq\tau_n, \tag{2}$$

without loss of generality. Let $1\leq m\leq n$, and denote the harmonic mean of the first m values of τ_i by

$$\tau_{\mathrm{h}}^m:=\left(\frac{1}{m}\sum_{i=1}^m\frac{1}{\tau_i}\right)^{-1}.$$

Assumptions

We consider the standard assumptions from the nonconvex world.

Assumption 1. *Function f is differentiable, and its gradient is L–Lipschitz continuous, i.e.,*

$$\|\nabla f(x)-\nabla f(y)\|\leq L\|x-y\|,\ \forall x,y\in\mathbb{R}^d.$$

Assumption 2. *There exist $f^{\inf}\in\mathbb{R}$ such that $f(x)\geq f^{\inf}$ for all $x\in\mathbb{R}^d$. We define $\Delta:=f(x^0)-f^{\inf}$, where x^0 is the starting point of optimization methods.*

Assumption 3. *The stochastic gradients $\nabla f(x;\xi)$ are unbiased and have bounded variance $\sigma^2\geq 0$. Specifically,*

$$\mathbb{E}_\xi\left[\nabla f(x;\xi)\right]=\nabla f(x),\ \forall x\in\mathbb{R}^d,$$
$$\mathbb{E}_\xi\left[\|\nabla f(x;\xi)-\nabla f(x)\|^2\right]\leq\sigma^2,\ \forall x\in\mathbb{R}^d.$$

Ringmaster ASGD

We are now ready to present our new versions of Asynchronous SGD, called Ringmaster ASGD (Algorithm 1), which guarantee the *optimal time complexity* without knowing the computation times a priori.

Algorithm 1 Ringmaster ASGD

Input: point $x^0\in\mathbb{R}^d$, stepsize $\gamma>0$, delay threshold $R\in\mathbb{N}$
Set $k=0$
Workers start computing stochastic gradients at x^0
while True **do**
  Stop calculating stochastic gradients with delays $\geq R$, and start computing new ones at x^k instead
  Gradient $\nabla f(x^{k-\delta^k};\xi_i^{k-\delta^k})$ arrives from worker i
  Update: $x^{k+1}=x^k-\gamma\nabla f(x^{k-\delta^k};\xi_i^{k-\delta^k})$
  Worker i begins calculating $\nabla f(x^{k+1};\xi_i^{k+1})$
  Update the iteration number $k=k+1$
end while

Note that we have a parameter R called the *delay threshold*. When $R=1$, the algorithm reduces to the classical SGD method, i.e.,

$$x^{k+1}=x^k-\gamma\nabla f(x^k;\xi_i^k),$$

since $\delta^k=0$ for all $k\geq 0$. In this case, the algorithm becomes highly conservative, ignoring all stochastic gradients computed at earlier points $x^{k-1},\ldots,x^0$. Conversely, if $R=\infty$, the method incorporates stochastic gradients with arbitrarily large delays, and becomes classical Asynchronous SGD. Intuitively, there should be a balance – a "proper" value of R that would i) prevent the method from being overly conservative, while ii) ensuring stability by making sure that only informative stochastic gradients are used to update the model. We formalize these intuitions by proposing an optimal R in Theorem 2. Interestingly, the value of R does *not* depend on the computation times.

Why Do We Ignore the Old Gradients?

Ignoring old gradients allows us to establish tighter convergence guarantees. Intuitively, old gradients not only fail to provide additional useful information about the function f, but they can also negatively impact the algorithm's performance.

Table 1 below shows our method achieves optimal time complexity.

Method	Time Complexity
Asynchronous SGD [2] [5]	$\tau_{\mathrm{h}}^n\left(\frac{1}{\varepsilon}+\frac{\sigma^2}{m\varepsilon^2}\right)$
Ringmaster ASGD	$\min_{m\in[n]}\left\{\tau_{\mathrm{h}}^m\left(\frac{1}{\varepsilon}+\frac{\sigma^2}{m\varepsilon^2}\right)\right\}$
Lower Bound [6]	$\min_{m\in[n]}\left\{\tau_{\mathrm{h}}^m\left(\frac{1}{\varepsilon}+\frac{\sigma^2}{m\varepsilon^2}\right)\right\}$

Table 1: The time complexities of asynchronous stochastic gradient methods, which preform the step $x^{k+1}=x^k-\gamma_k\nabla f(x^{k-\delta^k};\xi_i^{k-\delta^k})$, to get an ε-stationary point in the nonconvex setting.

Theoretical Results

Here is the theoretical analysis of Ringmaster ASGD. We start with an *iteration complexity* bound:

Theorem 1. *Under Assumptions 1, 2, and 3, let the stepsize in Ringmaster ASGD (Algorithm 1) be*

$$\gamma=\min\left\{\frac{1}{2RL},\frac{\varepsilon}{4L\sigma^2}\right\}.$$

Then

$$\frac{1}{K+1}\sum_{k=0}^K\mathbb{E}\left[\left\|\nabla f\left(x^k\right)\right\|^2\right]\leq\varepsilon,$$

as long as

$$K\geq\frac{8RL\Delta}{\epsilon}+\frac{16\sigma^2L\Delta}{\epsilon^2}, \tag{3}$$

where $R\in\{1,2,\ldots,\}$.

The classical analysis of Asynchronous SGD achieves the same convergence rate, with R defined as $R\equiv\max_{k\in[K]}\delta^k$ [1]. This outcome is expected, as setting $R=\max_{k\in[K]}\delta^k$ in Ringmaster ASGD makes it equivalent to classical Asynchronous SGD, since no gradients are ignored.

It is important to recognize that the *iteration complexity* (3) does *not* capture the actual "runtime" performance of the algorithm. In order to find the *time complexity*, we need the following lemma.

Lemma 1. *Let the workers' computation times satisfy the* fixed computation model *((1) and (2)). Let R be the delay threshold of Algorithm 1. The time required to complete any R consecutive iterate updates of Algorithm 1 is at most*

$$t(R):=2\min_{m\in[n]}\left[\left(\frac{1}{m}\sum_{i=1}^m\frac{1}{\tau_i}\right)^{-1}\left(1+\frac{R}{m}\right)\right].$$

Combining Theorem 1 and Lemma 1, we have:

Theorem 2 (Optimality of Ringmaster ASGD)**.** *Let Assumptions 1, 2, and 3 hold. Let the stepsize in Ringmaster ASGD (Algorithm 1) be $\gamma=\min\left\{\frac{1}{2RL},\frac{\varepsilon}{4L\sigma^2}\right\}$. Then, under the* fixed computation model *((1) and (2)), Ringmaster ASGD achieves the optimal time complexity*

$$\mathcal{O}\left(\min_{m\in[n]}\left[\left(\frac{1}{m}\sum_{i=1}^m\frac{1}{\tau_i}\right)^{-1}\left(\frac{L\Delta}{\varepsilon}+\frac{\sigma^2L\Delta}{m\varepsilon^2}\right)\right]\right)$$

with the delay threshold

$$R=\max\left\{1,\left\lceil\frac{\sigma^2}{\varepsilon}\right\rceil\right\}. \tag{4}$$

Note that the value of R does not in any way depend on the computation times $\{\tau_1,\ldots,\tau_n\}$.

Experiments

The optimization task is based on the convex quadratic function $f:\mathbb{R}^d\to\mathbb{R}$ such that

$$f(x)=\frac{1}{2}x^\top Ax-b^\top x\qquad\forall x\in\mathbb{R}^d.$$

The computation times for each worker are simulated as $\tau_i=i+|\eta_i|$ for all $i\in[n]$, where $\eta_i\sim\mathcal{N}(0,i)$. We tuned the stepsize from the set $\{5^p:p\in[-5,5]\}$. Both the batch size for Rennala SGD and the delay threshold for Ringmaster ASGD were tuned from the set $\{\lceil n/4^p\rceil:p\in\mathbb{N}_0\}$. The experimental results are shown in Figure 1.

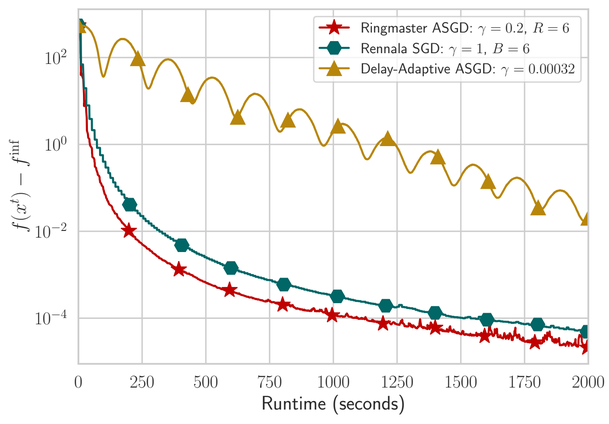

Figure 1: Experiment with $n=6174$ and $d=1729$ showing the convergence of Ringmaster ASGD, Delay-Adaptive ASGD, and Rennala SGD.

Conclusion

In this work, we developed the first Asynchronous SGD method, named Ringmaster ASGD, that achieves optimal time complexity. By selecting an appropriate delay threshold R in Algorithm 1, the method attains the theoretical lower bounds established by [6].

References

[1] Yossi Arjevani, Ohad Shamir, and Nathan Srebro. A tight convergence analysis for stochastic gradient descent with delayed updates. In *Algorithmic Learning Theory*, pages 111–132. PMLR, 2020.

[2] Anastasiia Koloskova, Sebastian U Stich, and Martin Jaggi. Sharper convergence guarantees for Asynchronous SGD for distributed and federated learning. *Advances in Neural Information Processing Systems*, 35:17202–17215, 2022.

[3] Jakub Konečný, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[4] Artavazd Maranjyan, Omar Shaikh Omar, and Peter Richtárik. MindFlayer SGD: Efficient parallel SGD in the presence of heterogeneous and random worker compute times. In *Conference on Uncertainty in Artificial Intelligence*, 2025.

[5] Konstantin Mishchenko, Francis Bach, Mathieu Even, and Blake Woodworth. Asynchronous SGD beats minibatch SGD under arbitrary delays. *arXiv preprint arXiv:2206.07638*, 2022.

[6] Alexander Tyurin and Peter Richtárik. Optimal time complexities of parallel stochastic optimization methods under a fixed computation model. *Advances in Neural Information Processing Systems*, 36, 2023.