

Graph Oracle Models, Lower Bounds, and Gaps for Parallel Stochastic Optimization



Blake Woodworth
(TTIC)



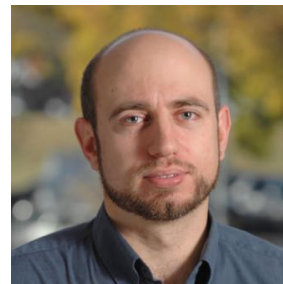
Jialei Wang
(UChicago \rightarrow 2σ Investments)



Adam Smith
(Boston University)



H. Brendan McMahan
(Google)



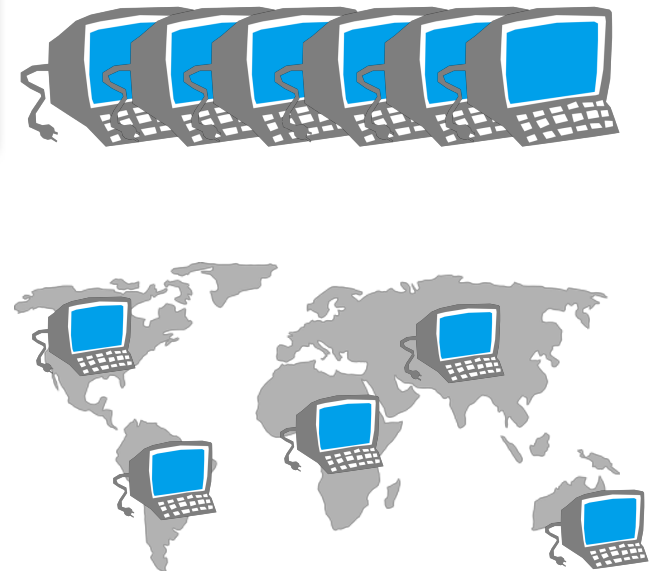
Nati Srebro
(TTIC)

Parallel Stochastic Optimization/Learning

$$\min_x F(x) := \mathbb{E}_{z \sim \mathcal{D}} [f(x; z)]$$

Many parallelization scenarios:

- Synchronous parallelism
- Asynchronous parallelism
- Delayed updates
- Few/many workers
- Infrequent communication
- Federated learning
- ...



Distributed Delayed Stochastic Optimization

Alekh Agarwal, John C. Duchi
Department of Electrical Engineering and Computer Sciences, University of California, Berkeley (alekh, jduchi)

We analyze the convergence of whose updates depend on delayed information. Our results in this paper show that for smooth and strongly convex functions, the convergence rate is known to be optimal even in the presence of delayed updates.

1 Introduction

We focus on stochastic convex optimization problems of the form

$$\min_{x \in \mathcal{X}} F(x) \text{ for } F(x) = \mathbb{E}_{z \sim \mathcal{D}} f(x; z)$$

where $\mathcal{X} \subseteq \mathbb{R}^d$ is a closed convex set, $f(x; z)$ is convex for all $z \in \mathcal{D}$, so that F is convex.

HOGWILD!: A Lock-Free Approach to Parallel Stochastic Gradient Descent

Feng Niu, Benjamin Recht, Christopher Re, Stephen J. Wright
leoni@cs.wisc.edu, brecht@cs.wisc.edu, chrisre@cs.wisc.edu, stephenjw@cs.wisc.edu

Stochastic Gradient Descent (SGD) is a popular algorithm that can achieve the best performance on a variety of machine learning tasks. Several variants have been proposed to parallelize SGD, but all require expensive synchronization. This work aims to design a novel theoretical analysis, algorithm, and implementation that SGD is implemented without any locking. We present an update scheme called HOGWILD! which allows processors to access shared memory with the possibility of interfering with each other's work. We show that when the associated optimization is *asynchronous*, meaning most gradient updates only modify small parts of the parameter space, then HOGWILD! achieves a nearly optimal rate of convergence. We demonstrate experimentally that HOGWILD! outperforms alternative schemes that use locking by an order of magnitude.

1 Introduction

With its small memory footprint, robustness against noise, and rapid learning rates, Stochastic Gradient Descent (SGD) has proved to be well suited to data-intensive machine learning tasks [3, 5, 24]. However, SGD's scalability is limited by its inherently sequential nature; it is difficult to parallelize. Nevertheless, the recent emergence of inexpensive multicore processors and mammoth, web-scale data sets has motivated researchers to develop several clever parallelization schemes for SGD [4, 10, 12, 16, 27]. As many large data sets are currently pre-processed in a MapReduce-like

3740 IEEE TRANSACTIONS ON AUTOMATIC CONTROL, VOL. 61, NO. 12, DECEMBER 2016

An Asynchronous Mini-Batch Algorithm for Regularized Stochastic Optimization

Hamid Reza Feyzmahdavian, Arda Aytekin, and Mikael Johansson

Abstract—Mini-batch optimization has proven to be a powerful paradigm for large-scale learning. However, the state-of-the-art parallel mini-batch algorithms assume synchronous operation or cyclic update orders. When worker nodes are heterogeneous (due to communication delays), these algorithms suffer from inefficient idle waiting times. In this paper, we propose an asynchronous mini-batch algorithm that allows workers to update their local models at their own pace. We show that the proposed algorithm achieves a near-linear speedup in terms of the number of iterations. Theoretical analysis shows that the algorithm converges to a distributed solution with a near-linear speedup in terms of the number of iterations. Theoretical analysis shows that the algorithm converges to a distributed solution with a near-linear speedup in terms of the number of iterations.

AdaDelay: Delay Adaptive Distributed Stochastic Convex Optimization

Suvrit Sra, Adams Wei Yu, Mu Li, Alexander J. Smola
suvrit@mit.edu, weiyu@cs.cmu.edu, muli@cs.cmu.edu, alexander.j.smola@cmu.edu

We study distributed stochastic convex optimization under a server nodes perform parameter updates, while the worker nodes discuss, analyze, and experiment with a setup motivated by their computation networks, where the machines are differently slow. The parameter updates to be sensitive to the actual delays experienced by the workers. This sensitivity leads to larger initial convergence without having to wait too long for slower or asymptotic complexity. We obtain encouraging improvements in experiments on real datasets with up to billions of examples and

Better Mini-Batch Algorithms via Accelerated Gradient Methods

Andrew Cotter, Ohad Shamir, Nathan Srebro, Karthik Sridharan
cotta@ttic.edu, ohadsh@microsoft.com, nsrebro@ttic.edu, karthik@ttic.edu

Abstract

Mini-batch algorithms have been proposed as a way to speed-up stochastic convex optimization problems. We study how such algorithms can be improved using accelerated gradient methods. We provide a novel analysis, which shows how standard gradient methods may sometimes be insufficient to obtain a significant speed-up and propose a novel accelerated gradient algorithm, which deals with this deficiency, enjoys a uniformly superior guarantee and works well in practice.

1 Introduction

We consider a stochastic convex optimization problem of the form $\min_{w \in \mathbb{R}^d} L(w)$, where $L(w) = \mathbb{E}_z [f(w; z)]$, based on an empirical sample of instances z_1, \dots, z_n . We assume that \mathcal{W} is a convex subset of some Hilbert space (which in this paper, we will take to be Euclidean space), and f is non-negative, convex and smooth in its first argument (i.e. has a Lipschitz-continuous gradient). The classical learning application is when $z = (x, y)$ and $f(w; (x, y))$

Distributed Stochastic Variance Reduced Gradient Methods by Sampling Extra Data with Replacement

Jason D. Lee, Qihang Lin, Tianshao Yang
jasonlee@marshall.usc.edu, qihanglin@uiowa.edu, tianshao-yang@uiowa.edu

Abstract

Stochastic gradient descent is popular for large scale optimization but has slow convergence asymptotically due to the inherent variance. To remedy this problem, we introduce an explicit variance reduction method for stochastic gradient descent which we call stochastic variance reduced gradient (SVRG). For smooth and strongly convex functions, we prove that this method enjoys the same fast convergence rate as those of stochastic dual coordinate ascent (SDCA) and Stochastic Average Gradient (SAG). However, our analysis is significantly simpler and more intuitive. Moreover, unlike SDCA or SAG, our method does not require the storage of gradients, and thus is more easily applicable to complex problems such as some structured prediction problems and neural network learning.

Accelerating Stochastic Gradient Descent using Predictive Variance Reduction

Rie Johnson, Tong Zhang
RJ Research Consulting, Tarrytown NY, USA, baids@cs.rutgers.edu, Rutgers University, New Jersey, USA

Abstract

Stochastic gradient descent is popular for large scale optimization but has slow convergence asymptotically due to the inherent variance. To remedy this problem, we introduce an explicit variance reduction method for stochastic gradient descent which we call stochastic variance reduced gradient (SVRG). For smooth and strongly convex functions, we prove that this method enjoys the same fast convergence rate as those of stochastic dual coordinate ascent (SDCA) and Stochastic Average Gradient (SAG). However, our analysis is significantly simpler and more intuitive. Moreover, unlike SDCA or SAG, our method does not require the storage of gradients, and thus is more easily applicable to complex problems such as some structured prediction problems and neural network learning.

1 Introduction

In machine learning, we often encounter the following optimization problem. Let ϕ_1, \dots, ϕ_n be a sequence of vector functions from \mathbb{R}^d to \mathbb{R} . Our goal is to find an approximate solution of the following optimization problem

$$\min_w P(w), \quad P(w) = \frac{1}{n} \sum_{i=1}^n \phi_i(w). \quad (1)$$

For example, given a sequence of training examples $(x_1, y_1), \dots, (x_n, y_n)$, where $x_i \in \mathbb{R}^d$ and

Communication-Efficient Learning of Deep Networks from Decentralized Data

H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hanson, Blake Agliera, Arca

Abstract

Modern mobile devices have access to a wealth of data suitable for learning models, which in turn can greatly improve the user experience on the device. For example, language models can improve speech recognition and text entry, and image models can automatically

FEDERATED LEARNING: STRATEGIES FOR IMPROVING COMMUNICATION EFFICIENCY

Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Ananda Theertha Suresh & Dave Bacon
jkonecny@mcmahan.com, felixyu, theertha, dabacon@google.com

Abstract

Federated Learning is a machine learning setting where the goal is to train a high-quality centralized model while training data remains distributed over a large number of clients each with unreliable and relatively slow network connections. We consider learning algorithms for this setting where on each round, each client independently computes an update to the current model based on its local data, and communicates this update to a central server, where the client-side updates are aggregated to compute a new global model. The typical clients in this setting are mobile phones, and communication efficiency is of the utmost importance.

Distributed Delayed Stochastic Optimization

Alekh Agarwal, John C. Duchi
Department of Electrical Engineering and Computer Sciences, University of California, Berkeley (alekh, jduchi)

We analyze the convergence of whose updates depend on delayed information. Our results in this paper show that for smooth and strongly convex functions, the convergence rate is known to be optimal even in the presence of delayed updates.

1 Introduction

We focus on stochastic convex optimization problems of the form

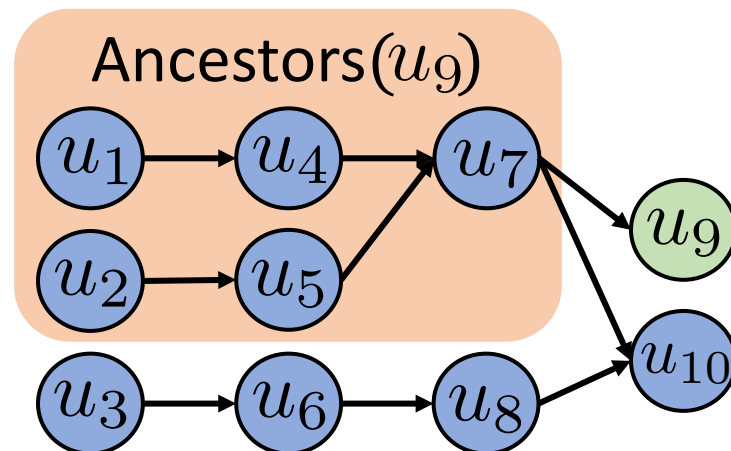
$$\min_{x \in \mathcal{X}} F(x) \text{ for } F(x) = \mathbb{E}_{z \sim \mathcal{D}} f(x; z)$$

where $\mathcal{X} \subseteq \mathbb{R}^d$ is a closed convex set, $f(x; z)$ is convex for all $z \in \mathcal{D}$, so that F is convex.

What is the best we can hope for in a given parallelism scenario?

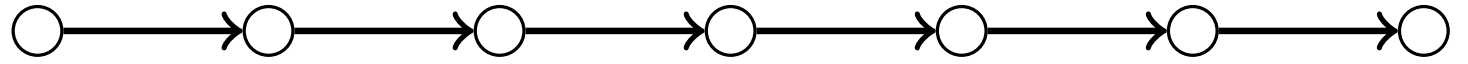
What is the best we can hope for in a given parallelism scenario?

- We formalize the parallelism in terms of a dependency graph:

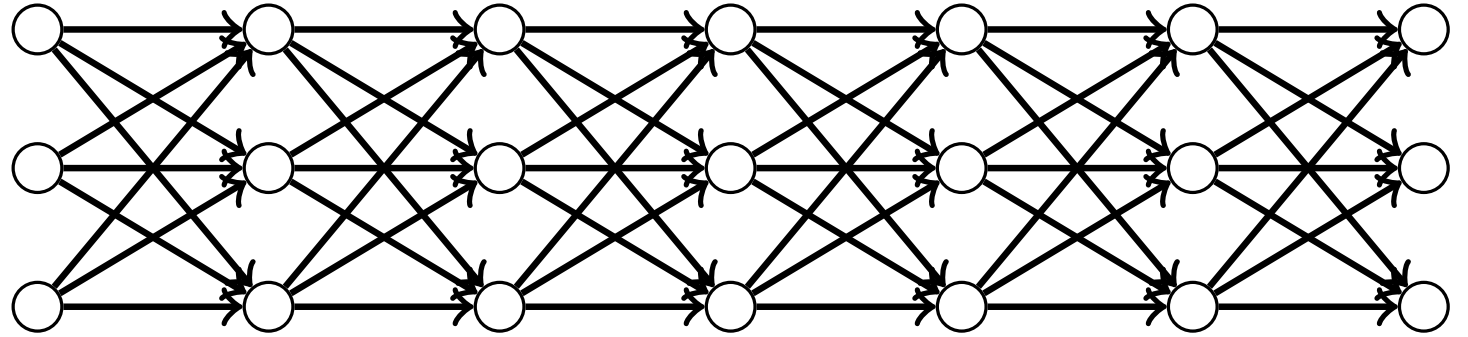


- At each node u , make a query based only on knowledge of *ancestors'* oracle interaction (plus shared randomness)
- Graph defines class of optimization algorithms $\mathcal{A}(\mathcal{G})$
- Come to our poster for details

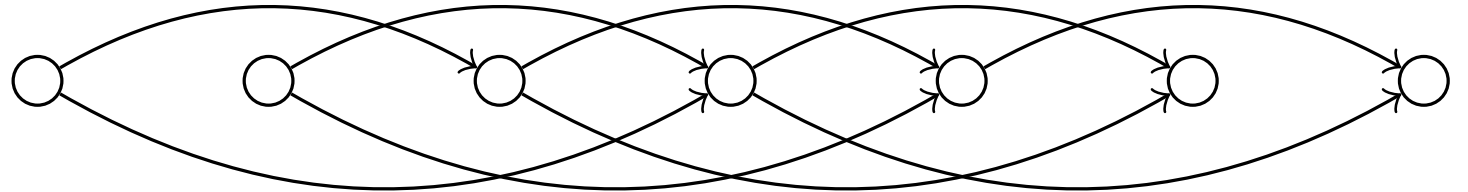
• Sequential:



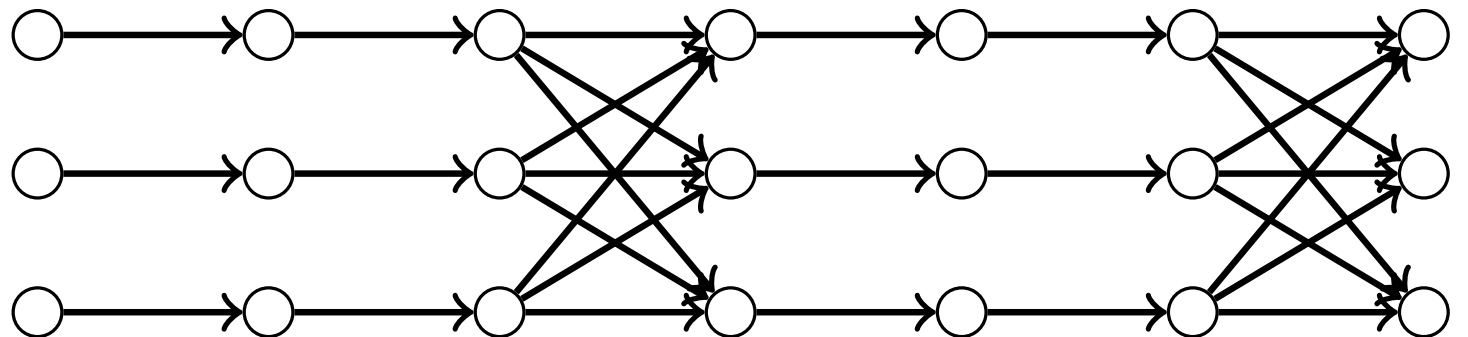
• Layer:



• Delays:



• Intermittent Communication:



Generic Lower Bounds

Theorem: For any dependency graph \mathcal{G} with N nodes and depth D , no algorithm for optimizing convex, L -Lipschitz, H -smooth $f(x; z)$ on a bounded domain in high dimensions can guarantee error less than:

With stochastic gradient oracle:

$$\Omega \left(\min \left\{ \frac{L}{\sqrt{D}}, \frac{H}{D^2} \right\} + \frac{L}{\sqrt{N}} \right)$$

Generic Lower Bounds

Theorem: For any dependency graph \mathcal{G} with N nodes and depth D , no algorithm for optimizing convex, L -Lipschitz, H -smooth $f(x; z)$ on a bounded domain in high dimensions can guarantee error less than:

With stochastic gradient oracle:

$$\Omega \left(\min \left\{ \frac{L}{\sqrt{D}}, \frac{H}{D^2} \right\} + \frac{L}{\sqrt{N}} \right)$$

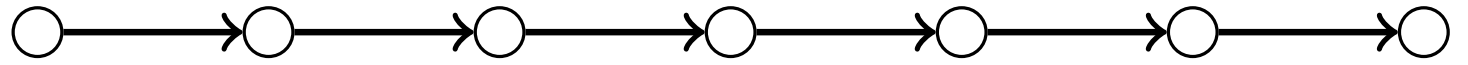
With stochastic prox oracle:

$$\Omega \left(\min \left\{ \frac{L}{D}, \frac{H}{D^2} \right\} + \frac{L}{\sqrt{N}} \right)$$

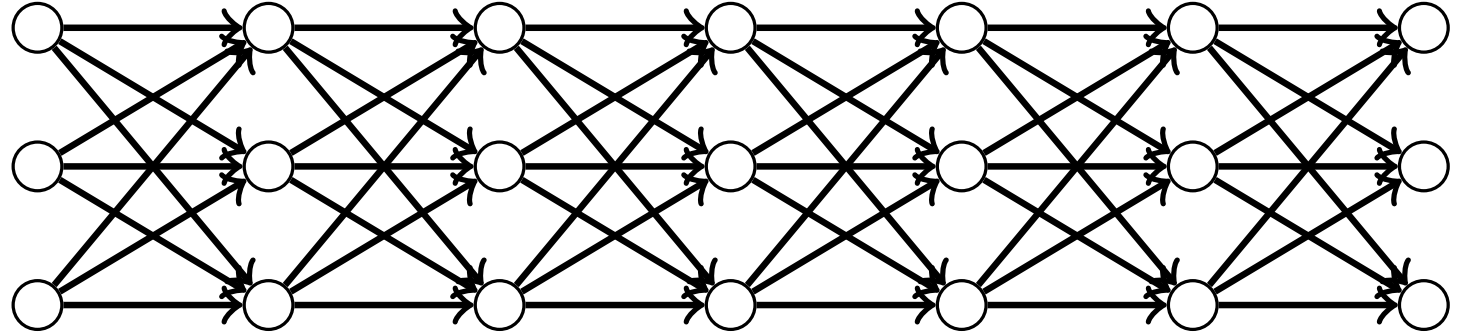
$$\text{Prox oracle: } x, \beta, z \mapsto \arg \min_y f(y; z) + \frac{\beta}{2} \|y - x\|^2$$

i.e. exactly optimize subproblem in each node (ADMM, DANE, etc.)

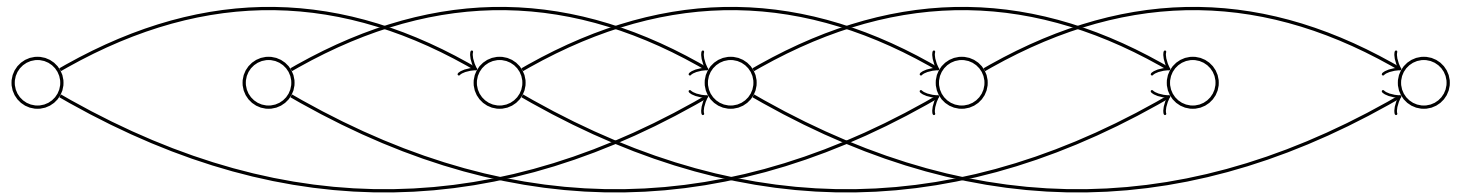
- Sequential:
 - SGD is optimal



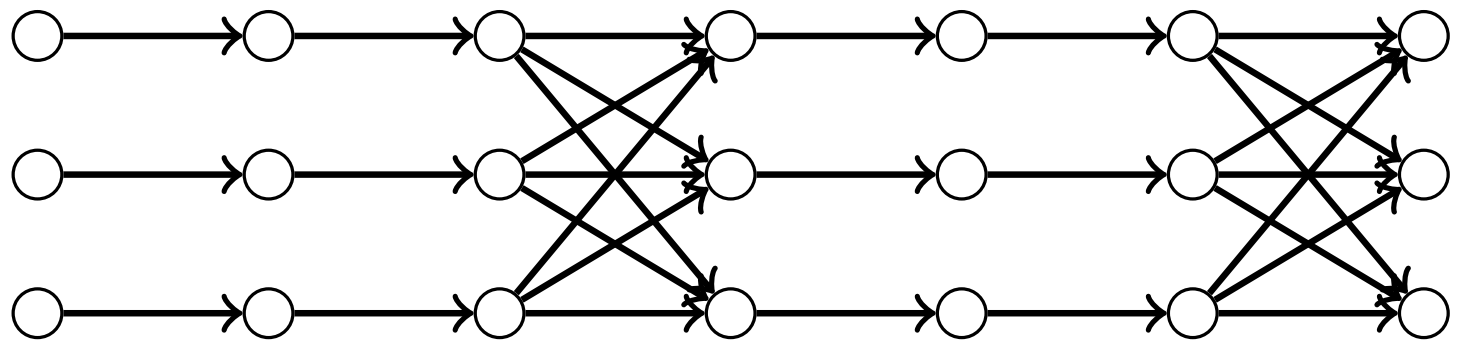
- Layers:
 - Accelerated minibatch SGD is optimal



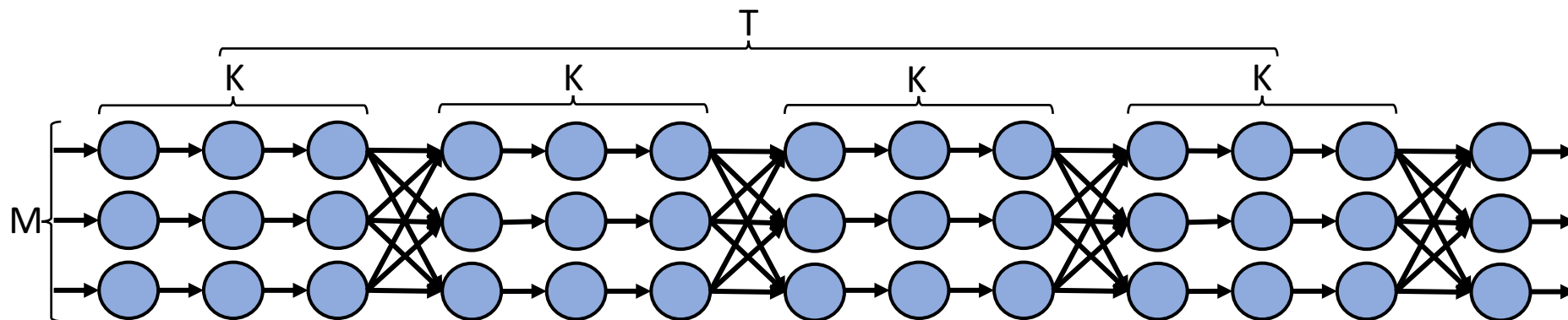
- Delays:
 - Delayed-update SGD is *not* optimal
 - “Wait-and-Collect” minibatch is optimal



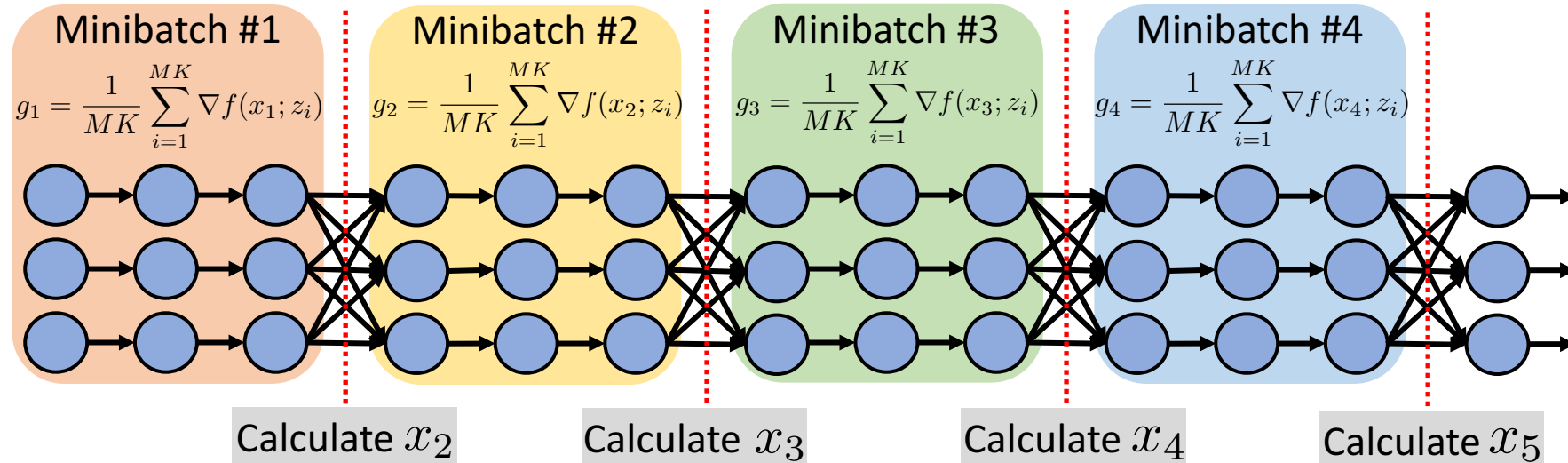
- Intermittent Communication:
 - Gaps between existing algorithms and lower bound



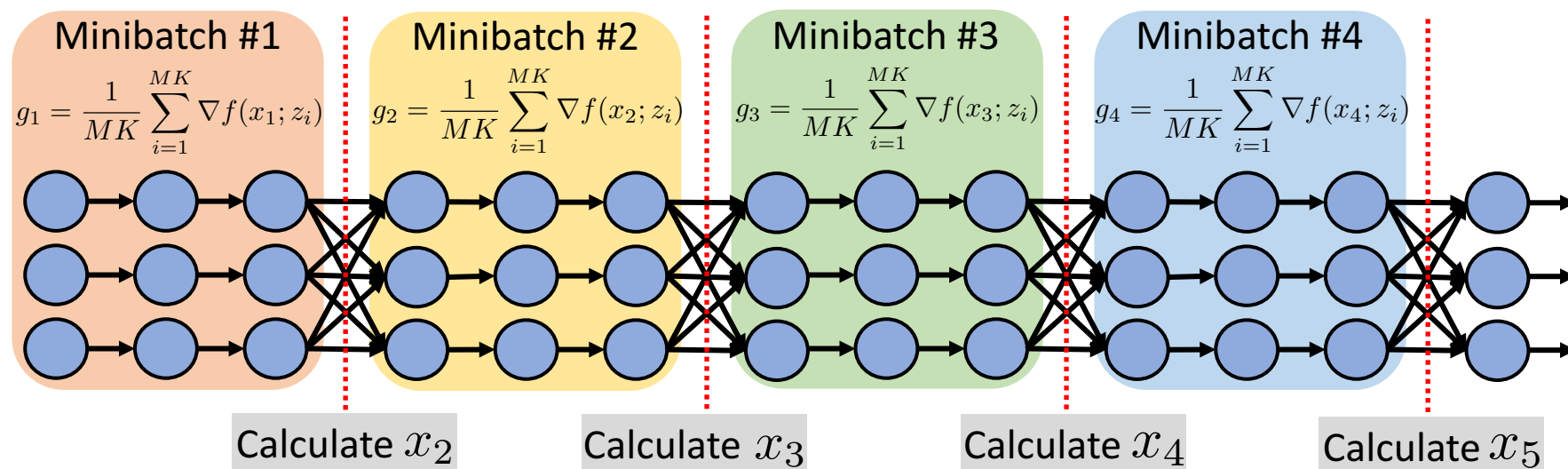
- Lower bound: $\Omega \left(\min \left\{ \frac{L}{\sqrt{TK}}, \frac{H}{T^2 K^2} \right\} + \frac{L}{\sqrt{TKM}} \right)$



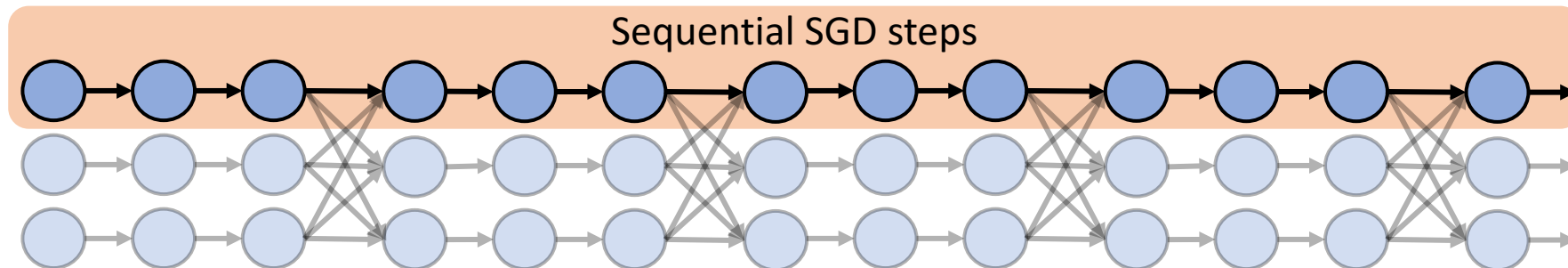
- Lower bound: $\Omega \left(\min \left\{ \frac{L}{\sqrt{TK}}, \frac{H}{T^2 K^2} \right\} + \frac{L}{\sqrt{TKM}} \right)$
- Option 1: Accelerated Minibatch SGD $O \left(\frac{H}{T^2} + \frac{L}{\sqrt{TKM}} \right)$



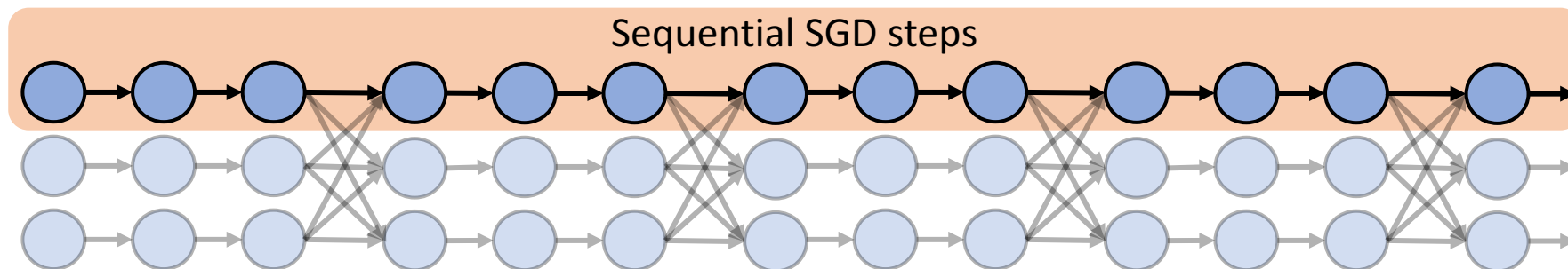
- Lower bound: $\Omega \left(\min \left\{ \frac{L}{\sqrt{TK}}, \frac{H}{T^2 K^2} \right\} + \frac{L}{\sqrt{TKM}} \right)$
- Option 1: Accelerated Minibatch SGD $O \left(\frac{H}{T^2} + \frac{L}{\sqrt{TKM}} \right)$



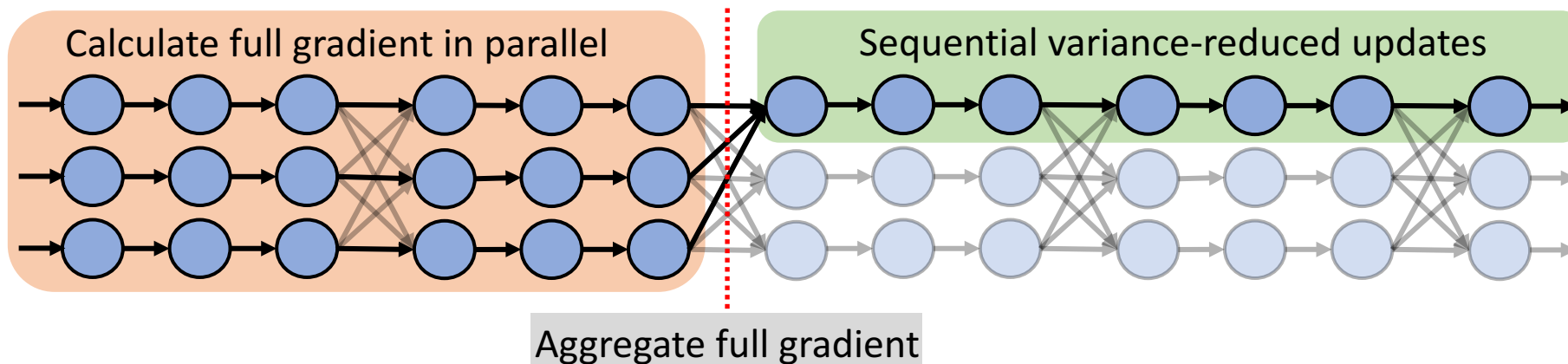
- Lower bound: $\Omega \left(\min \left\{ \frac{L}{\sqrt{TK}}, \frac{H}{T^2 K^2} \right\} + \frac{L}{\sqrt{TKM}} \right)$
- Option 1: Accelerated Minibatch SGD $O \left(\frac{H}{T^2} + \frac{L}{\sqrt{TKM}} \right)$
- Option 2: Sequential SGD $O \left(\frac{L}{\sqrt{TK}} \right)$



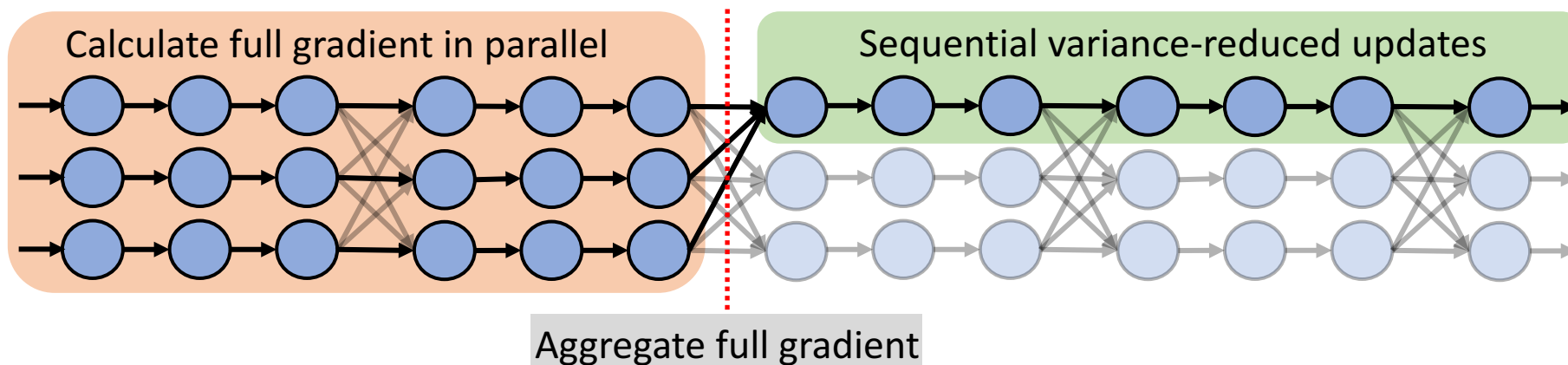
- Lower bound: $\Omega \left(\min \left\{ \frac{L}{\sqrt{TK}}, \frac{H}{T^2 K^2} \right\} + \frac{L}{\sqrt{TKM}} \right)$
- Option 1: Accelerated Minibatch SGD $O \left(\frac{H}{T^2} + \frac{L}{\sqrt{TKM}} \right)$
- Option 2: Sequential SGD $O \left(\frac{L}{\sqrt{TK}} \right)$



- Lower bound: $\Omega \left(\min \left\{ \frac{L}{\sqrt{TK}}, \frac{H}{T^2 K^2} \right\} + \frac{L}{\sqrt{TKM}} \right)$
- Option 1: Accelerated Minibatch SGD $O \left(\frac{H}{T^2} + \frac{L}{\sqrt{TKM}} \right)$
- Option 2: Sequential SGD $O \left(\frac{L}{\sqrt{TK}} \right)$
- Option 3: SVRG on empirical objective $\tilde{O} \left(\frac{H}{TK} + \frac{L}{\sqrt{TKM}} \right)$



- Lower bound: $\Omega \left(\min \left\{ \frac{L}{\sqrt{TK}}, \frac{H}{T^2 K^2} \right\} + \frac{L}{\sqrt{TKM}} \right)$
- Option 1: Accelerated Minibatch SGD $O \left(\frac{H}{T^2} + \frac{L}{\sqrt{TKM}} \right)$
- Option 2: Sequential SGD $O \left(\frac{L}{\sqrt{TK}} \right)$
- Option 3: SVRG on empirical objective $\tilde{O} \left(\frac{H}{TK} + \frac{L}{\sqrt{TKM}} \right)$



- Lower bound: $\Omega \left(\min \left\{ \frac{L}{\sqrt{TK}}, \frac{H}{T^2 K^2} \right\} + \frac{L}{\sqrt{TKM}} \right)$
- Option 1: Accelerated Minibatch SGD $O \left(\frac{H}{T^2} + \frac{L}{\sqrt{TKM}} \right)$
- Option 2: Sequential SGD $O \left(\frac{L}{\sqrt{TK}} \right)$
- Option 3: SVRG on empirical objective $\tilde{O} \left(\frac{H}{TK} + \frac{L}{\sqrt{TKM}} \right)$
- Combining 1-3:

$$\tilde{O} \left(\min \left\{ \frac{L}{\sqrt{TK}}, \frac{H}{TK}, \frac{H}{T^2} \right\} + \frac{L}{\sqrt{TKM}} \right)$$

- Lower bound: $\Omega \left(\min \left\{ \frac{L}{\sqrt{TK}}, \frac{H}{T^2 K^2} \right\} + \frac{L}{\sqrt{TKM}} \right)$
- Option 1: Accelerated Minibatch SGD $O \left(\frac{H}{T^2} + \frac{L}{\sqrt{TKM}} \right)$
- Option 2: Sequential SGD $O \left(\frac{L}{\sqrt{TK}} \right)$
- Option 3: SVRG on empirical objective $\tilde{O} \left(\frac{H}{TK} + \frac{L}{\sqrt{TKM}} \right)$

- **Combining 1-3:**

$$\tilde{O} \left(\min \left\{ \frac{L}{\sqrt{TK}}, \frac{H}{TK}, \frac{H}{T^2} \right\} + \frac{L}{\sqrt{TKM}} \right)$$

- Lower bound: $\Omega \left(\min \left\{ \frac{L}{\sqrt{TK}}, \frac{H}{T^2 K^2} \right\} + \frac{L}{\sqrt{TKM}} \right)$
- Option 1: Accelerated Minibatch SGD $O \left(\frac{H}{T^2} + \frac{L}{\sqrt{TKM}} \right)$
- Option 2: Sequential SGD $O \left(\frac{L}{\sqrt{TK}} \right)$
- Option 3: SVRG on empirical objective $\tilde{O} \left(\frac{H}{TK} + \frac{L}{\sqrt{TKM}} \right)$

- **Combining 1-3:**

$$\tilde{O} \left(\min \left\{ \frac{L}{\sqrt{TK}}, \frac{H}{TK}, \frac{H}{T^2} \right\} + \frac{L}{\sqrt{TKM}} \right)$$

- Option 4: Parallel SGD ???

Come to our poster
tonight from 5-7pm