Optimization for Machine Learning

Lecture 6: Distributed Optimization I

Sebastian Stich

CISPA – `https://cms.cispa.saarland/optml24/`

May 21, 2024

Midterm Exam

- Date: June 4, 4pm, CISPA building
- Duration: 1 hour
- Closed book. you can bring one sheet (A4) paper with notes, handwritten or fontsize > 10pt.
- Content: Lecture 1–7, Exercise Sheets 1–6.

Some questions will be inspired by the "quizzes" and questions asked during the lecture. At least one question will be similar to one of the distributed exercises.

You find old practice exam on the course website. Note that this are final exams, designed for a duration of 2.5hrs.

Quiz Week 6

Assumption (Variance) Assume $\mathbf{g}\colon \mathbb{R}^d \to \mathbb{R}^d$ is a unbiased gradient oracle, with

$$\mathbb{E}[\mathbf{g}(\mathbf{x})] = \nabla f(\mathbf{x}) \qquad \mathbb{E}\left[\|\mathbf{g}(\mathbf{x}) - \nabla f(\mathbf{x})\|^2\right] \leq M \|\nabla f(\mathbf{x})\|^2 + \sigma^2$$

1. (Suppose $M = 0$). For a mini-batch of size B, the variance is exactly $\frac{\sigma^2}{B}$.

2. (Suppose $\sigma^2 = 0$). Then SGD converges with a constant stepsize on smooth functions.

Quiz Week 6 (II)

Large Scale Optimization

$$\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x})$$

with

$$f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) \qquad \text{or} \qquad f(\mathbf{x}) = \mathbb{E}_\xi[F(\mathbf{x},\xi)]$$

where $n \gg 1$.

- How can we utilize the compute power of multiple machines/parallel threads?

Mini-Batch SGD

Mini-Batch SGD

Input: $\mathbf{x}_0 \in \mathbb{R}^d$, stepsize γ, batch size $b \geq 1$

Goal: $\min_{\mathbf{x} \in \mathbb{R}^d} f(\mathbf{x})$

At iteration t:

query (in parallel) b independent stochastic gradients:

$$\mathbf{g}_t^i = \mathbf{g}(\mathbf{x}_t), i = 1, \ldots, b$$

$$\mathbf{x}_{t+1} := \mathbf{x}_t - \gamma \sum_{i=1}^{b} \mathbf{g}_t^i$$

- Note: we could also write the update as $\mathbf{x}_{t+1} := \mathbf{x}_t - \gamma' \frac{1}{b} \sum_{i=1}^{b} \mathbf{g}_t^i$ with b times larger stepsize $\gamma' = b\gamma$.
- We will use the terms ‘worker’, ‘node’, ‘process’, etc., synonymous, assuming that each compute unit computes one single stochastic gradient each (and b units compute b gradients in parallel).

Convergence of Mini-Batch SGD

Theorem (Lecture-6).1 ([SMJ21])

Let $f\colon \mathbb{R}^d \to \mathbb{R}$ be L-smooth and assume $f(\mathbf{x}) \geq f^\star$. Define $F_0 = f(\mathbf{x}_0) - f^\star$. Then there exists a stepsize $\gamma \leq \gamma_{\text{crit}} := \frac{1}{10L(M+b)}$ such that after T steps of mini-batch SGD:

$$\min_{t \leq T} \mathbb{E} \left\| \nabla f(\mathbf{x}_t) \right\|^2 = \mathcal{O}\left(\frac{LF_0(M+b)}{bT} + \frac{\sqrt{LF_0\sigma^2}}{\sqrt{bT}} \right) .$$

- The stochastic term (with σ^2) is optimal, as (bT) is equal to the total number of stochastic gradients computed.
- Note: for $b = 1$ this is equivalent Theorem 1 of Lecture 4.

Discussion

- As long as $b \ll M$, we see a linear speedup in the convergence rate:

$$\mathcal{O}\left(\frac{M+b}{b\epsilon} + \frac{\sigma^2}{b\epsilon^2}\right) \cdot LF_0 = \mathcal{O}\left(\frac{M}{b\epsilon} + \frac{\sigma^2}{b\epsilon^2}\right) \cdot LF_0$$

- If $b \gg M$, there is no speedup in the deterministic term, only in the stochastic term:

$$\mathcal{O}\left(\frac{M+b}{b\epsilon} + \frac{\sigma^2}{b\epsilon^2}\right) \cdot LF_0 = \mathcal{O}\left(\frac{1}{\epsilon} + \frac{\sigma^2}{b\epsilon^2}\right) \cdot LF_0$$

Discussion II

Learning Rate Scaling

A a consequence, we can deduce the learning rate scaling rule:
If my algorithm works well with batch size b, which learning rate should be used when doubling the batch size to $2b$?

- If the algorithm is written as $\mathbf{x}_{t+1} = \mathbf{x}_t - \gamma \sum_{i=1}^{b} \mathbf{g}_t^i$, then
  - if $b \ll M$, keep the same γ
  - if $b \gg M$, use $\gamma/2$
- If the algorithm is written as $\mathbf{x}_{t+1} = \mathbf{x}_t - \gamma \frac{1}{b} \sum_{i=1}^{b} \mathbf{g}_t^i$, then
  - if $b \ll M$, use 2γ
  - if $b \gg M$, keep the same γ

See [GDG$^+$17] and [SMJ21].

HogWild!
Asynchronous SGD

Motivation

- Computing b stochastic gradients might take take different (real) time on different nodes/processes.
- If we have to wait until all computations are finished,
- Can we apply gradients whenever they have been computed, in an asynchronous fashion?

Hogwild! [RRWN11]

Input: $\mathbf{x}_0 \in \mathbb{R}^d$, stepsize γ, accessible memory location to store $\mathbf{x} \in \mathbb{R}^d$

At iteration t (in parallel):

```
$\mathbf{x}_t \leftarrow \mathbf{x}$                                   (inconsistent read of the memory $\mathbf{x}$)
$\mathbf{g}_t = \mathbf{g}(\mathbf{x}_t)$                              (stochastic gradient)
for $i \in [d]$                                       (atomic coordinate write)
    $[\mathbf{x}]_i := [\mathbf{x}]_i - \gamma[\mathbf{g}_t]_i$
```

- Historically developed for shared-memory implementations, allowing coordinate-wise read/write (and overwrites).
- In distributed settings, we can often assume (but do not need) atomic vector operations.

Difficulties

- defining iterates $\mathbf{x}_t$
  - $\mathbf{x}_t$ might not exist
- Assign index t at the moment the worker reads the last entry of $\mathbf{x}$, with breaking ties arbitrarily (after-read approach).
  - $\mathbb{E}\mathbf{g}(\mathbf{x}_t) = \nabla f(\mathbf{x}_t)$
- Large delays might impact the convergence.
  - Assume (for simplicity) atomic vector operations. Then

$$\mathbf{x}_t = \mathbf{x}_0 - \gamma \sum_{k \in \mathcal{I}_t} \mathbf{g}_k$$

for an index set $\mathcal{I}_t \subseteq [t-1]$. Define $\mathcal{J}_t = [t-1] \setminus \mathcal{I}_t$.

Definition (Lecture-6).2 (Delay)

Define the level of parallelism/delay $\tau \geq 1$ as:

$$\tau = \sup_{t \geq 0} \max_{k \in \mathcal{J}_t} (t - k)$$

Theorem

Theorem (Lecture-6).3 ([SK20, SMJ21])

Let $f\colon \mathbb{R}^d \to R$ be L-smooth with $F_0 = f(\mathbf{x}_0) - f^\star$. Then there exists a stepsize $\gamma \leq \gamma_{\text{crit}} := \frac{1}{10L(M+\tau)}$ such that after T steps of delayed SGD (with atomic vector operations):

$$\min_{t \leq T} \mathbb{E}\left\|\nabla f(\mathbf{x}_t)\right\|^2 = \mathcal{O}\left(\frac{F_0 L(M+\tau)}{T} + \frac{\sqrt{LF_0\sigma^2}}{\sqrt{T}}\right) .$$

- Mini-Batch SGD can be seen as a variant of delayed SGD with $\tau = b$.
- We recover the mini-batch SGD result when considering the same number of gradient computations, i.e. $T \to Tb$ and replacing $\tau \to b$).

Proof I

The main ingredient for the proof is to define a virtual sequence $\tilde{\mathbf{x}}_t$ of iterates, $\tilde{\mathbf{x}}_0 = \mathbf{x}_0$, defined as [MPP+17, SK20]

$$\tilde{\mathbf{x}}_{t+1} = \tilde{\mathbf{x}}_t - \gamma \mathbf{g}_t .$$

Lemma (Lecture-6).4 (Decrease)

For $\gamma \leq \gamma_{\mathrm{crit}}$ it holds

$$\mathbb{E} f(\tilde{\mathbf{x}}_{t+1}) \leq \mathbb{E} f(\tilde{\mathbf{x}}_t) - \frac{\gamma}{4} \left\| \nabla f(\mathbf{x}_t) \right\|^2 + \frac{\gamma^2 L \sigma^2}{2} + \frac{\gamma L^2}{2} \mathbb{E} \left\| \mathbf{x}_t - \tilde{\mathbf{x}}_t \right\|^2$$

Lemma (Lecture-6).5 (Difference)

For $\gamma \leq \gamma_{\mathrm{crit}}$ it holds *here* $(t-\tau)_+ = \max\{0, t-\tau\}$

$$\mathbb{E} \left\| \mathbf{x}_t - \tilde{\mathbf{x}}_t \right\|^2 \leq \frac{1}{50 L^2 \tau} \sum_{k=(t-\tau)_+}^{t-1} \mathbb{E} \left\| \nabla f(\mathbf{x}_k) \right\|^2 + \frac{\gamma}{5L} \sigma^2 .$$

Proof II

Plug (Difference) into (Decrease), re-arrange, and divide by γ:

$$\frac{1}{4}\mathbb{E}\left\|\nabla f(\mathbf{x}_t)\right\|^2 \leq \frac{1}{\gamma}\left(\mathbb{E}f(\tilde{\mathbf{x}}_t) - \mathbb{E}f(\tilde{\mathbf{x}}_{t+1})\right) + \frac{\gamma L\sigma^2}{2} + \frac{1}{100\tau}\sum_{k=(t-\tau)_+}^{t-1}\mathbb{E}\left\|\nabla f(\mathbf{x}_k)\right\|^2 + \frac{\gamma L\sigma^2}{10}$$

Now we average over T. Note that the highlighted $\left\|\nabla f(\mathbf{x}_k)\right\|^2$ terms appear at most τ times.

$$\frac{1}{4T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f(\mathbf{x}_t)\right\|^2 \leq \frac{\Delta}{\gamma T} + \gamma L\sigma^2 + \frac{1}{100T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f(\mathbf{x}_t)\right\|^2$$

Note that $\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f(\mathbf{x}_t)\right\|^2$ appears on both sides, with $\frac{1}{4T} - \frac{1}{100T} \geq \frac{1}{5T}$.

$$\frac{1}{5T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f(\mathbf{x}_t)\right\|^2 \leq \frac{F_0}{\gamma T} + \gamma L\sigma^2 \,.$$

Now the result follows by tuning γ (in the same way as before).

Discussion

Of the result:

- delay and batch sizes are levels of parallelism that are to some extend interchangeable
  - note that we assumed uniform sampling of the stochastic gradients (in the batches, or for the worker processes)
- Extension to atomic coordinate writes possible (exercise)

Of the proof:

- The virtual iterate analysis has proven to be a very useful tool (also for other applications/algorithms).
  - also for strongly-convex and convex settings

Proof of Lemma (Decrease)

This follows our standard path, with one small trick. By L-smoothness (at $\tilde{\mathbf{x}}_t$):

$$\begin{aligned}\mathbb{E}[f(\tilde{\mathbf{x}}_{t+1}) &\leq \mathbb{E}[f(\tilde{\mathbf{x}}_t)] - \gamma \nabla f(\tilde{\mathbf{x}}_t)^\top \mathbf{g}_t + \tfrac{\gamma^2 L}{2} \mathbb{E} \left\| \mathbf{g}_t \right\|^2 \\ &\leq \mathbb{E}[f(\tilde{\mathbf{x}}_t)] - \gamma \nabla f(\tilde{\mathbf{x}}_t)^\top \nabla f(\mathbf{x}_t) + \frac{\gamma^2 L}{2} \left((M+1) \left\| \nabla f(\mathbf{x}_t) \right\|^2 + \sigma^2 \right)\end{aligned}$$

Now:

$$\begin{aligned}-\nabla f(\tilde{\mathbf{x}}_t)^\top \nabla f(\mathbf{x}_t) &= - \left(\nabla f(\tilde{\mathbf{x}}_t) - \nabla f(\mathbf{x}_t) + \nabla f(\mathbf{x}_t) \right)^\top \nabla f(\mathbf{x}_t) \\ &= - \left\| \nabla f(\mathbf{x}_t) \right\|^2 - \left(\nabla f(\tilde{\mathbf{x}}_t) - \nabla f(\mathbf{x}_t) \right)^\top \nabla f(\mathbf{x}_t) \\ &\leq - \left\| \nabla f(\mathbf{x}_t) \right\|^2 + \tfrac{1}{2} \left\| \nabla f(\mathbf{x}_t) \right\|^2 + \tfrac{1}{2} \left\| \nabla f(\tilde{\mathbf{x}}_t) - \nabla f(\mathbf{x}_t) \right\|^2 \\ &\leq -\tfrac{1}{2} \left\| \nabla f(\mathbf{x}_t) \right\|^2 + \tfrac{L^2}{2} \left\| \tilde{\mathbf{x}}_t - \mathbf{x}_t \right\|^2\end{aligned}$$

where we used $(-\mathbf{a}^\top \mathbf{b}) \leq \frac{1}{2} \left\| \mathbf{a} \right\|^2 + \frac{1}{2} \left\| \mathbf{b} \right\|^2$. Now with $\gamma \leq \frac{1}{2L(M+\tau)} \leq \frac{1}{2L(M+1)}$:

$$\mathbb{E}[f(\tilde{\mathbf{x}}_{t+1}) \leq \mathbb{E}[f(\tilde{\mathbf{x}}_t)] - \frac{\gamma}{4} \left\| \nabla f(\mathbf{x}_t) \right\|^2 + \frac{\gamma^2 L \sigma^2}{2} + \frac{\gamma L^2}{2} \left\| \tilde{\mathbf{x}}_t - \mathbf{x}_t \right\|^2$$

Proof of Lemma (Difference)

Note that $\mathbf{x}_t = \mathbf{x}_0 - \gamma \sum_{k \in \mathcal{I}_t} \mathbf{g}_k$ and $\tilde{\mathbf{x}}_t = \mathbf{x}_t - \gamma \sum_{k=0}^{t-1} \mathbf{g}_k$ and define $\xi_k = \mathbf{g}_k - \nabla f(\mathbf{x}_k)$, with $\mathbb{E}[\xi_k] = 0$.

Then

$$\begin{aligned}\mathbb{E}\left\|\tilde{\mathbf{x}}_t - \mathbf{x}_t\right\|^2 &= \gamma^2 \mathbb{E}\left\|\sum_{k \in \mathcal{J}_t} \mathbf{g}_k\right\|^2 \leq 2\gamma^2 \mathbb{E}\left\|\sum_{k \in \mathcal{J}_t} \nabla f(\mathbf{x}_k)\right\|^2 + 2\gamma^2 \mathbb{E}\left\|\sum_{k \in \mathcal{J}_t} \xi_k\right\|^2 \\ &\leq 2\gamma^2 \tau \sum_{k=(t-\tau)_+}^{t-1} \left\|\nabla f(\mathbf{x}_k)\right\|^2 + 2\gamma^2 M \sum_{k=(t-\tau)_+}^{t-1} \left\|\nabla f(\mathbf{x}_k)\right\|^2 + 2\gamma^2 \tau \sigma^2\end{aligned}$$

Where we used

- $\left\|\nabla f(\mathbf{x}_k) + \xi_k\right\|^2 \leq 2\left\|\nabla f(\mathbf{x}_k)\right\|^2 + 2\left\|\xi_k\right\|^2$
- $\left\|\sum_{k=1}^{\tau} \mathbf{a}_k\right\|^2 \leq \tau \sum_{k=1}^{\tau} \left\|\mathbf{a}_k\right\|^2$
- $\mathbb{E}\left\|\sum_{k=1}^{\tau} \xi_k\right\|^2 = \sum_{k=1}^{\tau} \mathbb{E}\left\|\xi_k\right\|^2$ (independent noise)

Lecture 6 Recap

- we have seen a strategy to parallelize computation, under the assumption that we can compute IID stochastic gradients for a given $\mathbf{x} \in \mathbb{R}^d$.
- discussion of mini-batch SGD:
  - $M > 0$ allows to explain the benefit of SGD over GD
  - $M > 0$ allows to understand the effect of the batch size on the optimization
  - $M = 0$ (bounded variance) suffices for most purposes, as this is the most difficult scenario for SGD
- delayed gradient methods
- the 'perturbed iterate' analysis as new tool

Bibliography I

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He.
Accurate, large minibatch SGD: Training imagenet in 1 hour.
arXiv preprint arXiv:1706.02677, 2017.

Horia Mania, Xinghao Pan, Dimitris Papailiopoulos, Benjamin Recht, Kannan Ramchandran, and Michael I Jordan.
Perturbed iterate analysis for asynchronous stochastic optimization.
SIAM Journal on Optimization, 27(4):2202–2229, 2017.

Benjamin Recht, Christopher Re, Stephen Wright, and Feng Niu.
Hogwild: A lock-free approach to parallelizing stochastic gradient descent.
In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 693–701. Curran Associates, Inc., 2011.

Bibliography II

Sebastian U. Stich and Sai P. Karimireddy.
The error-feedback framework: Better rates for SGD with delayed gradients and compressed communication.
Journal of Machine Learning Research (JMLR), 2020.

Sebastian Stich, Amirkeivan Mohtashami, and Martin Jaggi.
Critical parameters for scalable distributed learning with large batches and asynchronous updates.
In Arindam Banerjee and Kenji Fukumizu, editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 4042–4050. PMLR, 13–15 Apr 2021.

Discussion

Discussion

Discussion