Optimization for Machine Learning

Lecture 5: Newton's Method & Adaptive Gradient Methods

Sebastian Stich

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

May 14, 2024

Quiz Week 5

Recall the coordinate-wise smoothness condition:

$$\left\|\nabla_i f(\mathbf{x}) - \nabla_i f(\mathbf{y})\right\|^2 \leq L_i \left\|\mathbf{x} - \mathbf{y}\right\|^2 \qquad \text{vs.} \qquad \left\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\right\|^2 \leq L \left\|\mathbf{x} - \mathbf{y}\right\|^2$$

1. It holds $L \leq L_i$.
2. It holds $L = \max_i L_i$.
3. It holds $L = \sum_{i=1}^n L_i$.
4. It holds $L \geq \frac{1}{n} \sum_{i=1}^n L_i$.

Quiz Week 5 (II)

Consider

$$f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x})$$

where each $f_i \colon \mathbb{R}^d \to \mathbb{R}$ is L_i-smooth, and let L denote the smoothness constant of f.

1. Then $L \geq \max_i L_i$.
2. Then $L = \sum_{i=1}^{n} L_i$.
3. Then $L \geq \frac{1}{n}\sum_{i=1}^{n} L_i$.

Theory-Practice Gap

- In theory, without imposing additional assumption or structure, it is impossible to achieve an (asymptotically!) better rate than SGD.
- In practice, acceleration techniques such as momentum, adaptive pre-conditioning are heavily used.
  - difficult to analyze!
- this lecture:
  - Newton's method (part I)
  - overview of some adaptive methods used in practice (part II)
  - (appendix: a method that adapts the stepsize)

Chapter 8

Newton's Method

1-dimensional case: Newton-Raphson method

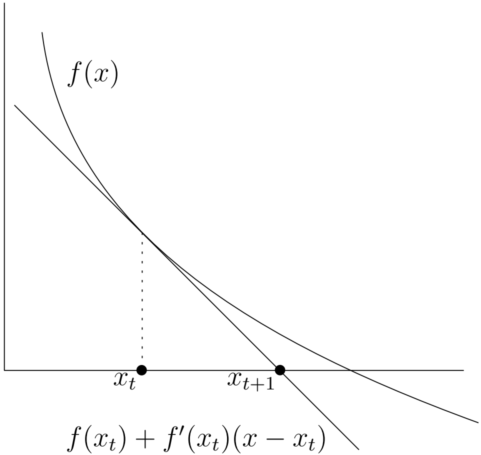
Goal: find a zero of differentiable $f : \mathbb{R} \to \mathbb{R}$.

Method:

$$x_{t+1} := x_t - \frac{f(x_t)}{f'(x_t)}, \quad t \geq 0.$$

x_{t+1} solves

$$f(x_t) + f'(x_t)(x - x_t) = 0,$$



The Babylonian method

Computing square roots: find a zero of $f(x) = x^2 - R, R \in \mathbb{R}_+$.

Newton-Raphson step:

$$x_{t+1} = x_t - \frac{f(x_t)}{f'(x_t)} = x_t - \frac{x_t^2 - R}{2x_t} = \frac{1}{2}\left(x_t + \frac{R}{x_t}\right).$$

Starting far (large $x_0 > 0$), we move slowly:

$$x_{t+1} = \frac{1}{2}\left(x_t + \frac{R}{x_t}\right) \geq \frac{x_t}{2}.$$

E.g., from $x_0 = R \geq 1$, it takes $\mathcal{O}(\log R)$ steps to get $x_t - \sqrt{R} < 1/2$ (Exercise 38).

The Babylonian method - Takeoff

Starting close, $x_0 - \sqrt{R} < 1/2$ (achievable after $\mathcal{O}(\log R)$ steps), things will speed up:

$$x_{t+1} - \sqrt{R} = \frac{1}{2}\left(x_t + \frac{R}{x_t}\right) - \sqrt{R} = \frac{x_t}{2} + \frac{R}{2x_t} - \sqrt{R} = \frac{1}{2x_t}\left(x_t - \sqrt{R}\right)^2.$$

Assume $R \geq 1/4$.Then all iterates have value at least $\sqrt{R} \geq 1/2$. Hence we get

$$x_{t+1} - \sqrt{R} \leq \left(x_t - \sqrt{R}\right)^2.$$

$$x_T - \sqrt{R} \leq \left(x_0 - \sqrt{R}\right)^{2^T} < \left(\frac{1}{2}\right)^{2^T}, \quad T \geq 0.$$

To get $x_T - \sqrt{R} < \varepsilon$, we only need $T = \log\log(\frac{1}{\varepsilon})$ steps!

The Babylonian method - Example

$R = 1000$, IEEE 754 double arithmetic

- 7 steps to get $x_7 - \sqrt{1000} < 1/2$
- 3 more steps to get x_{10} equal to $\sqrt{1000}$ up to machine precision (53 binary digits).
- First phase: $\approx$ one more correct digit per iteration
- Last phase, $\approx$ double the number of correct digits in each iteration!

Once you're close, you're there...

Newton's method for optimization

1-dimensional case: Find a global minimum $x^\star$ of a differentiable convex function $f : \mathbb{R} \to \mathbb{R}$.

Can equivalently search for a zero of the derivative f': Apply the Newton-Raphson method to f'.

Update step:

$$x_{t+1} := x_t - \frac{f'(x_t)}{f''(x_t)} = x_t - f''(x_t)^{-1} f'(x_t)$$

(needs f twice differentiable).

d-dimensional case: Newton's method for minimizing a convex function $f : \mathbb{R}^d \to \mathbb{R}$:

$$\mathbf{x}_{t+1} := \mathbf{x}_t - \nabla^2 f(\mathbf{x}_t)^{-1} \nabla f(\mathbf{x}_t)$$

Newton's method = adaptive gradient descent

General update scheme:

$$\mathbf{x}_{t+1} = \mathbf{x}_t - H(\mathbf{x}_t)\nabla f(\mathbf{x}_t),$$

where $H(\mathbf{x}) \in \mathbb{R}^{d\times d}$ is some matrix.

Newton's method: $H = \nabla^2 f(\mathbf{x}_t)^{-1}$.

Gradient descent: $H = \gamma I$.

Newton's method: "adaptive gradient descent", adaptation is w.r.t. the local geometry of the function at $\mathbf{x}_t$.

Convergence in one step on quadratic functions

A nondegenerate quadratic function is a function of the form

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top M\mathbf{x} - \mathbf{q}^\top \mathbf{x} + c,$$

where $M \in \mathbb{R}^{d\times d}$ is an invertible symmetric matrix, $\mathbf{q} \in \mathbb{R}^d, c \in R$. Let $\mathbf{x}^\star = M^{-1}\mathbf{q}$ be the unique solution of $\nabla f(\mathbf{x}) = \mathbf{0}$.

- $\mathbf{x}^\star$ is the unique global minimum if f is convex.

Lemma (Lecture-5).1

On nondegenerate quadratic functions, with any starting point $\mathbf{x}_0 \in \mathbb{R}^d$, Newton's method yields $\mathbf{x}_1 = \mathbf{x}^\star$.

Proof.

We have $\nabla f(\mathbf{x}) = M\mathbf{x} - \mathbf{q}$ (this implies $\mathbf{x}^\star = M^{-1}\mathbf{q}$) and $\nabla^2 f(\mathbf{x}) = M$. Hence,

$$\mathbf{x}_1 = \mathbf{x}_0 - \nabla^2 f(\mathbf{x}_0)^{-1}\nabla f(\mathbf{x}_0) = \mathbf{x}_0 - M^{-1}(M\mathbf{x}_0 - \mathbf{q}) = M^{-1}\mathbf{q} = \mathbf{x}^\star.$$

□

Minimizing the second-order Taylor approximation

Alternative interpretation of Newton's method:

Each step minimizes the local second-order Taylor approximation.

Lemma (Lecture-5).2 (Exercise 42)

Let f be convex and twice differentiable at $\mathbf{x}_t \in \mathbf{dom}(f)$, with $\nabla^2 f(\mathbf{x}_t) \succ 0$ being invertible. The vector $\mathbf{x}_{t+1}$ resulting from the Netwon step satisfies

$$\mathbf{x}_{t+1} = \underset{\mathbf{x} \in \mathbb{R}^d}{\operatorname{argmin}} \; f(\mathbf{x}_t) + \nabla f(\mathbf{x}_t)^\top (\mathbf{x} - \mathbf{x}_t) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_t)^\top \nabla^2 f(\mathbf{x}_t)(\mathbf{x} - \mathbf{x}_t).$$

Downside of Newton's method

Computational bottleneck in each step:

- compute and invert the Hessian matrix
- or solve the linear system $\nabla^2 f(\mathbf{x}_t)\Delta\mathbf{x} = -\nabla f(\mathbf{x}_t)$ for the next step $\Delta\mathbf{x}$.

Matrix / system has size $d \times d$, taking up to $\mathcal{O}(d^3)$ time to invert / solve.

In many applications, d is large...

Discussion

- Newton's Method
  - fast local convergence, $\mathcal{O}(\log\log\frac{1}{\epsilon})$
  - slow (or might even diverge) when initialized far-away from the optimal solution
- a method with global convergence guarantees:
  Cubic Regularized Newton's Method [NP06]
- computationally more efficient versions based on the secant-equation:
  quasi-Newton methods (see also [JJM24])

A First Adaptive Method (without proof)

Stochastic Gradient Descent

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

$$\mathbf{x}_{t+1} := \mathbf{x}_t - \gamma_t \mathbf{g}_t$$
$$\text{with } \mathbb{E}[\mathbf{g}_t] = \nabla f(\mathbf{x}_t)$$

Recall Lecture 3:

- Under the assumptions of convexity & $\mathbb{E}[\|\mathbf{g}_t\|^2] \leq B^2$, $\forall t$
- $\mathcal{O}\left(\frac{1}{\sqrt{T}}\right)$ convergence
- for the constant stepsize $\gamma_t = \gamma = \mathcal{O}\left(\frac{1}{B\sqrt{T}}\right)$

Estimating $\gamma = \frac{c}{B\sqrt{T}}$

- in practice we do not know B (or T)
- if we set $\gamma_t = \frac{c}{B\sqrt{t}}$ (for a constant c), we only need to estimate B
- empirical estimate:

$$B^2 \approx \frac{1}{t} \sum_{i=0}^{t} \|\mathbf{g}_i\|^2$$

- this leads to

$$\gamma_t = \frac{c}{\sqrt{\sum_{i=0}^{t} \|\mathbf{g}_i\|^2}}$$

The resulting method is quite tricky to analyze, as γ_t depends on $\mathbf{g}_t$.

Main Theorem

Theorem (Lecture-5).3 ([LO19, Cut22])

Let $f: \mathbb{R}^d \to \mathbb{R}$ *be* L*-smooth,* B*-Lipschitz and let* $\Delta = f(\mathbf{x}_0) - f^\star$*. Suppose* $\mathbb{E}[\max_{t \le T} \|\mathbf{g}_t\|] \le B$ *and* $\mathbb{E}[\|\mathbf{g}_t - \nabla f(\mathbf{x}_t)\|^2] \le \sigma^2$ *for all* t*. Then Adaptive SGD guarantees:*

$$\frac{1}{T+1}\mathbb{E}\left[\sqrt{\sum_{t=0}^{T} \|\nabla f(\mathbf{x}_t)\|^2}\right]^2 \le \tilde{\mathcal{O}}\left(\frac{\sigma}{\sqrt{T}}\right).$$

See appendix for more details.

Adaptive Methods in Practice

Adaptive Stochastic Gradient Methods

- Some limitations of SGD:
  - learning rate tuning
  - uniform learning rate for all coordinates

- Adaptive stepsizes are widely used in practice to improve the performance of SGD:
  - AdaGrad [DHS11]
  - RMSProp [TH12]
  - ADAM [KB14]
  - AMSGrad [RKK19]
  - ....

Popular Variants

- Momentum SGD

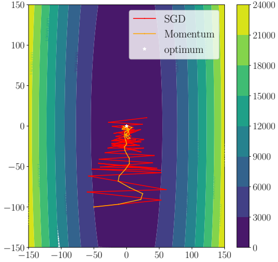
$$\begin{cases} \mathbf{m}_t &= \alpha \mathbf{m}_{t-1} + (1-\alpha)\nabla f(\mathbf{x}_t, \boldsymbol{\xi}_t) \\ \mathbf{x}_{t+1} &= \mathbf{x}_t - \gamma_t \mathbf{m}_t \end{cases}$$

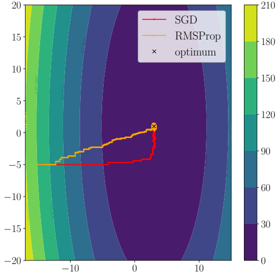
- AdaGrad

$$\begin{cases} \mathbf{v}_t &= \mathbf{v}_{t-1} + \nabla f(\mathbf{x}_t, \boldsymbol{\xi}_t)^{\odot 2} \\ \mathbf{x}_{t+1} &= \mathbf{x}_t - \frac{\gamma_0}{\epsilon + \sqrt{\mathbf{v}_t}} \odot \nabla f(\mathbf{x}_t, \boldsymbol{\xi}_t) \end{cases}$$

- RMSProp

$$\begin{cases} \mathbf{v}_t &= \beta \mathbf{v}_{t-1} + (1-\beta)\nabla f(\mathbf{x}_t, \boldsymbol{\xi}_t)^{\odot 2} \\ \mathbf{x}_{t+1} &= \mathbf{x}_t - \frac{\gamma_0}{\varepsilon + \sqrt{\mathbf{v}_t}} \odot \nabla f(\mathbf{x}_t, \boldsymbol{\xi}_t) \end{cases}$$





ADAM

ADAM $\approx$ RMSProp + Momentum (>100K citations)

$$\begin{cases} \mathbf{v}_t & = \beta \mathbf{v}_{t-1} + (1-\beta)\nabla f(\mathbf{x}_t, \boldsymbol{\xi}_t)^{\odot 2} \\ \mathbf{m}_t & = \alpha \mathbf{m}_{t-1} + (1-\alpha)\nabla f(\mathbf{x}_t, \boldsymbol{\xi}_t) \\ \mathbf{x}_{t+1} & = \mathbf{x}_t - \frac{\gamma_0}{\varepsilon + \sqrt{\tilde{\mathbf{v}}_t}} \odot \tilde{\mathbf{m}}_t \end{cases}$$

- Exponential decay of previous information $\mathbf{m}_t, \mathbf{v}_t$.
- Note $\tilde{\mathbf{v}}_t = \frac{\mathbf{v}_t}{1-\beta^t}$ and $\tilde{\mathbf{m}}_t = \frac{\mathbf{m}_t}{1-\alpha^t}$ are bias-corrected estimates.
- In practice, α and β are chosen to be close to 1.

Numerical Illustration

for an animation: CS231n (`https://cs231n.github.io/neural-networks-3/`)

Generic Adaptive Scheme

The following scheme encapsulates these popular adaptive methods in a unified framework. [RKK19]

$$
\begin{aligned}
\mathbf{g}_t &= \nabla f(\mathbf{x}_t, \boldsymbol{\xi}_t) \\
\mathbf{m}_t &= \phi_t(\mathbf{g}_1, \ldots, \mathbf{g}_t) \\
V_t &= \psi_t(\mathbf{g}_1, \ldots, \mathbf{g}_t) \\
\hat{\mathbf{x}}_t &= \mathbf{x}_t - \alpha_t V_t^{-1/2} \mathbf{m}_t \\
\mathbf{x}_{t+1} &= \operatorname*{argmin}_{\mathbf{x} \in X} \{ (\mathbf{x} - \hat{\mathbf{x}}_t)^T V_t^{1/2} (\mathbf{x} - \hat{\mathbf{x}}_t) \}
\end{aligned}
$$

Popular Examples

- **SGD**

$$\phi_t(\mathbf{g}_1, \ldots, \mathbf{g}_t) = \mathbf{g}_t, \quad \psi_t(\mathbf{g}_1, \ldots, \mathbf{g}_t) = \mathbb{I}$$

- **AdaGrad**

$$\phi_t(\mathbf{g}_1, \ldots, \mathbf{g}_t) = \mathbf{g}_t, \quad \psi_t(\mathbf{g}_1, \ldots, \mathbf{g}_t) = \frac{\mathrm{diag}(\sum_{\tau=1}^{t} \mathbf{g}_\tau^2)}{t}$$

- **Adam**

$$\phi_t(\mathbf{g}_1, \ldots, \mathbf{g}_t) = (1-\beta_1)\sum_{\tau=1}^{t} \beta_1^{t-\tau}\mathbf{g}_\tau, \quad \psi_t(\mathbf{g}_1, \ldots, \mathbf{g}_t) = (1-\beta_2)\mathrm{diag}(\sum_{\tau=1}^{t} \beta_2^{t-\tau}\mathbf{g}_\tau^2)$$

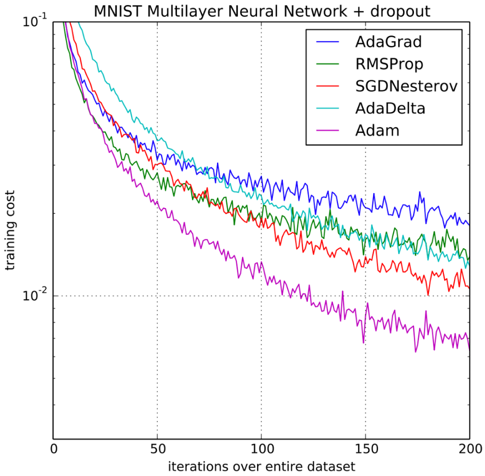
In other words, $\mathbf{m}_t = \beta_1 \mathbf{m}_{t-1} + (1-\beta_1)\mathbf{g}_t$, $V_t = \beta_2 V_{t-1} + (1-\beta_2)\mathrm{diag}(\mathbf{g}_t^2)$.

What do we know in practice?

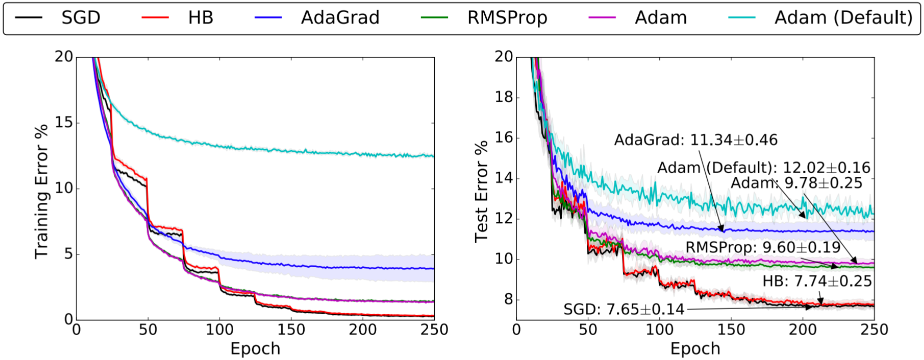
Adaptive methods

- Less sensitive to parameter tuning and adapt to sparse gradients.
- Outperform SGD for NLP tasks, training generative adversarial networks (GANs), deep reinforcement learning, etc., but are less effective in computer vision tasks.
- Tend to overfit and generalize worse than their non-adaptive counterparts [WRS+17].
- Often display faster initial progress on the training set, but their performance quickly plateaus on the testing set [WRS+17].

Some Good Stories



Some Bad Stories



(a) CIFAR-10 (Train) (b) CIFAR-10 (Test)

What do we know in theory?

- SGD with momentum has no acceleration even for some convex quadratic functions.
- For convex problems, Adagrad does converge, but RMSProp and Adam may not converge when $\beta_1 < \sqrt{\beta_2}$ (same for decreasing β_1 over time).

The Non-Convergence of Adam

Counterexample: consider a one-dimensional problem:

$$X = [-1, 1], \quad f(x, \xi) = \begin{cases} Cx, & \text{if } \xi = 1 \\ -x, & \text{if } \xi = 0 \end{cases}, \text{ where } P(\xi = 1) = p = \frac{1+\delta}{C+1}.$$

- Here $F(x) = \mathbb{E}[f(x, \xi)] = \delta x$ and $x^\star = -1$.
- The Adam step is $x_{t+1} = x_t - \gamma_0 \Delta_t$ with $\Delta_t = \frac{\alpha m_t + (1-\alpha) g_t}{\sqrt{\beta v_t + (1-\beta) g_t^2}}$
- For large enough $C > 0$, one can show that $\mathbb{E}[\Delta_t] \leq 0$.
- The Adam steps keep drifting away from the optimal solution $x^\star = -1$.

A Convergent Adam-type Algorithm

AMSGrad [RKK19]

Algorithm 2 AMSGRAD

Input: $x_1 \in \mathcal{F}$, step size $\{\alpha_t\}_{t=1}^T$, $\{\beta_{1t}\}_{t=1}^T$, β_2
Set $m_0 = 0$, $v_0 = 0$ and $\hat{v}_0 = 0$
for $t = 1$ **to** T **do**
  $g_t = \nabla f_t(x_t)$
  $m_t = \beta_{1t} m_{t-1} + (1 - \beta_{1t}) g_t$
  $v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$
  $\hat{v}_t = \max(\hat{v}_{t-1}, v_t)$ and $\hat{V}_t = \mathrm{diag}(\hat{v}_t)$
  $x_{t+1} = \Pi_{\mathcal{F}, \sqrt{\hat{V}_t}}(x_t - \alpha_t m_t / \sqrt{\hat{v}_t})$
end for

- Use maximum value for normalizing the running average of the gradient.
- Ensure non-increasing stepsize and avoid pitfalls of Adam and RMSProp.
- Allow long-term memory of past gradients.

Lecture 5 Recap

- introduction to Newton's method

- overview of adaptive methods used in practice

Bibliography I

A. Cutkosky.
Lecture notes for ec500: Optimization for machine learning, 2022.

John Duchi, Elad Hazan, and Yoram Singer.
Adaptive subgradient methods for online learning and stochastic optimization.
Journal of Machine Learning Research, 12(61):2121–2159, 2011.

Qiujiang Jin, Ruichen Jiang, and Aryan Mokhtari.
Non-asymptotic global convergence rates of bfgs with exact line search.
arXiv preprint arXiv:2404.01267, 2024.

Diederik P Kingma and Jimmy Ba.
Adam: A method for stochastic optimization.
arXiv preprint arXiv:1412.6980, 2014.

Bibliography II

Xiaoyu Li and Francesco Orabona.
On the convergence of stochastic gradient descent with adaptive stepsizes.
In Kamalika Chaudhuri and Masashi Sugiyama, editors, *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 983–992. PMLR, 16–18 Apr 2019.

Yurii Nesterov and B.T. Polyak.
Cubic regularization of newton method and its global performance.
Math. Program., Ser. A, 2006.

Sashank J Reddi, Satyen Kale, and Sanjiv Kumar.
On the convergence of adam and beyond.
arXiv preprint arXiv:1904.09237, 2019.

T. Tieleman and G. Hinton.
Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude.
COURSERA: Neural Networks for Machine Learning, pages 26–31, 2012.

Bibliography III

Ashia C Wilson, Rebecca Roelofs, Mitchell Stern, Nati Srebro, and Benjamin Recht.
The marginal value of adaptive gradient methods in machine learning.
Advances in neural information processing systems, 30, 2017.

Discussion

Discussion

Discussion

Appendix

An Adaptive Method (with Proof)

*Not part of the course materials/exam.

Adaptive Stochastic Gradient Descent

Input: $\mathbf{x}_0$, scaling c, a small constant $\epsilon > 0$

Repeat:

sample stochastic gradient $\mathbf{g}_t$

$$\gamma_t = \frac{c}{\sqrt{\epsilon^2 + \sum_{i=0}^{t} \|\mathbf{g}_i\|^2}}$$

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \gamma_t \mathbf{g}_t$$

Remark:

- this an (almost) parameter-free method, rate depends 'mildly' on c, ϵ
- small issue: correct choice of the remaining hyper-parameters, e.g. $\epsilon \approx B^2$

Auxiliary Theorem (Lecture-5).4

Theorem (Lecture-5).4 (A)

Let $f : \mathbb{R}^d \to \mathbb{R}$ *be* B*-Lipschitz,* L*-smooth and for every* t *let* $\mathbf{g}_t$ *denote a stochastic gradient* $\mathbb{E}_t[\mathbf{g}_t] = \nabla f(\mathbf{x}_t)$*, with* $\mathbb{E}[\max_{t \leq T} \|\mathbf{g}_t\|] \leq B$*. Let* $\gamma_0, \ldots, \gamma_T$ *be any sequence of learning rates such that (1)* $\gamma_t \geq 0$*, (2)* $\gamma_0 \geq \gamma_1 \geq \cdots \geq \gamma_T$*, and (3) the sequence is 'causal' in the sense that* γ_t *is not allowed to depend on* $\mathbf{g}_{t+1}, \ldots, \mathbf{g}_T$*. Let* γ_{-1} *be a deterministic quantity such that* $\gamma_{-1} \geq \gamma_0$*. Consider the SGD update:* $\mathbf{x}_{t+1} = \mathbf{x}_t - \gamma_t \mathbf{g}_t$*. Then we have*

$$\mathbb{E}\left[f(\mathbf{x}_{T+1})\right] \leq \mathbb{E}\left[f(\mathbf{x}_0) - \sum_{t=0}^{T} \gamma_{t-1} \left\|\nabla f(\mathbf{x}_t)\right\|^2 + \frac{L}{2} \sum_{t=0}^{T} \gamma_t^2 \left\|\mathbf{g}_t\right\|^2\right] + \gamma_{-1} B^2$$

Main Theorem

Theorem (Lecture-5).5 ([LO19, Cut22])

Let $f : \mathbb{R}^d \to \mathbb{R}$ *be* L*-smooth,* B*-Lipschitz and let* $\Delta = f(\mathbf{x}_0) - f^\star$*. Suppose* $\mathbb{E}[\max_{t \le T} \|\mathbf{g}_t\|] \le B$ *and* $\mathbb{E}[\|\mathbf{g}_t - \nabla f(\mathbf{x}_t)\|^2] \le \sigma^2$ *for all* t*. Define*

$$K = \frac{\Delta}{c} + \frac{Lc\log\left(1 + \frac{(T+1)(B^2+\sigma^2)}{\epsilon^2}\right)}{2} + \frac{B^2}{\epsilon} = \mathcal{O}(\log(T)) .$$

Then Adaptive SGD guarantees:

$$\frac{1}{T+1}\mathbb{E}\left[\sqrt{\sum_{t=0}^{T} \|\nabla f(\mathbf{x}_t)\|^2}\right]^2 \le \frac{8K^2 + 4K\epsilon}{T+1} + \frac{4K\sigma}{\sqrt{T+1}} = \tilde{\mathcal{O}}\left(\frac{\sigma}{\sqrt{T}}\right) .$$

Proof I

Applying Theorem A with $\gamma_{-1} = \frac{c}{\epsilon}$ gives

$$\mathbb{E}\left[f(\mathbf{x}_{T+1})\right] \leq \mathbb{E}\left[f(\mathbf{x}_0) - \sum_{t=0}^{T} \gamma_{t-1} \left\|\nabla f(\mathbf{x}_t)\right\|^2 + \frac{L}{2} \sum_{t=0}^{T} \gamma_t^2 \left\|\mathbf{g}_t\right\|^2\right] + \gamma_{-1} B^2$$

With the definition of γ_t:

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=0}^{T} \gamma_t^2 \left\|\mathbf{g}_t\right\|^2\right] &= \mathbb{E}\left[c^2 \sum_{t=0}^{T} \frac{\left\|\mathbf{g}_t\right\|^2}{\epsilon^2 + \sum_{i=0}^{t} \left\|\mathbf{g}_t\right\|^2}\right] \\
&\overset{\text{Lemma (Lecture-5).6}}{\leq} \mathbb{E}\left[c^2 \log\left(1 + \frac{\sum_{t=0}^{T} \left\|\mathbf{g}_t\right\|^2}{\epsilon^2}\right)\right] \\
&\overset{\text{Jensen ineq.}}{\leq} c^2 \log\left(1 + \frac{\sum_{t=0}^{T} \mathbb{E}\left\|\mathbf{g}_t\right\|^2}{\epsilon^2}\right) \\
&\leq c^2 \log\left(1 + \frac{(T+1)(B^2 + \sigma^2)}{\epsilon^2}\right)
\end{aligned}$$

Proof II

Thus, we have

$$\mathbb{E}\left[f(\mathbf{x}_{T+1})\right] \leq \mathbb{E}\left[f(\mathbf{x}_0) - \sum_{t=0}^{T} \gamma_{t-1} \left\|\nabla f(\mathbf{x}_t)\right\|^2\right] + \frac{Lc^2 \log\left(1 + \frac{(T+1)(B^2+\sigma^2)}{\epsilon^2}\right)}{2} + \frac{cB^2}{\epsilon}$$

rearranging:

$$\mathbb{E}\left[\sum_{t=0}^{T} \gamma_{t-1} \left\|\nabla f(\mathbf{x}_t)\right\|^2\right] \leq \mathbb{E}\left[f(\mathbf{x}_0) - f(\mathbf{x}_T)\right] + \frac{Lc^2 \log\left(1 + \frac{(T+1)(B^2+\sigma^2)}{\epsilon^2}\right)}{2} + \frac{cB^2}{\epsilon}$$

and using $\gamma_T \leq \gamma_t$:

$$\mathbb{E}\left[\sum_{t=0}^{T} \gamma_T \left\|\nabla f(\mathbf{x}_t)\right\|^2\right] \leq \Delta + \frac{Lc^2 \log\left(1 + \frac{(T+1)(B^2+\sigma^2)}{\epsilon^2}\right)}{2} + \frac{cB^2}{\epsilon} = cK$$

A technical Lemma

Lemma (Lecture-5).6 (Exercise)

Suppose $x_0, \ldots, x_T$ *are arbitrary non-negative values. And let* $f\colon \mathbb{R} \to \mathbb{R}$ *be an arbitrary decreasing function. Then*

$$\sum_{t=1}^{T} x_t f\left(\sum_{i=0}^{t} x_i\right) \leq \int_{x_0}^{\sum_{i=0}^{T} x_i} f(x)dx\,.$$

As a corollary:

$$\sum_{t=0}^{T} \frac{\|\mathbf{g}_t\|^2}{\epsilon^2 + \sum_{i=0}^{t} \|\mathbf{g}_t\|^2} \leq \int_{\epsilon^2}^{\epsilon^2+\sum_{t=0}^{T}\|\mathbf{g}_t\|^2} \frac{dx}{x} = \log\left(1 + \frac{\sum_{t=0}^{T} \|\mathbf{g}_t\|^2}{\epsilon^2}\right)$$

Proof cont. III

Define random variables

$$A^2 = \sum_{t=0}^{T} \gamma_T \left\|\nabla f(\mathbf{x}_t)\right\|^2 \qquad\qquad B^2 = \frac{1}{\gamma_T}$$

Then by Cauchy-Schwarz for random variables (Exercise)

$$\mathbb{E}[AB] \le \sqrt{\mathbb{E}[A^2]\mathbb{E}[B^2]}$$

$$\frac{\mathbb{E}[AB]^2}{\mathbb{E}[B^2]} \le \mathbb{E}[A^2]$$

$$\frac{\mathbb{E}\left[\sqrt{\sum_{t=0}^{T} \left\|\nabla f(\mathbf{x}_t)\right\|^2}\right]^2}{\mathbb{E}[\gamma_T^{-1}]} \le \mathbb{E}\left[\sum_{t=0}^{T} \gamma_T \left\|\nabla f(\mathbf{x}_t)\right\|^2\right]$$

Proof IV

With this, it now follows

$$\mathbb{E}\left[\sqrt{\sum_{t=0}^{T}\|\nabla f(\mathbf{x}_t)\|^2}\right]^2 \leq cK\mathbb{E}\left[\gamma_T^{-1}\right] = K\mathbb{E}\left[\sqrt{\epsilon^2 + \sum_{t=0}^{T}\|\mathbf{g}_t\|^2}\right]$$

Define $X = \mathbb{E}\left[\sqrt{\sum_{t=0}^{T}\|\nabla f(\mathbf{x}_t)\|^2}\right]$ and note

$\|\mathbf{g}_t\|^2 = \|\mathbf{g}_t - \nabla f(\mathbf{x}_t) + \nabla f(\mathbf{x}_t)\|^2 \leq 2\|\mathbf{g}_t - \nabla f(\mathbf{x}_t)\|^2 + 2\|\mathbf{g}_t\|^2$. Therefore

$$\begin{aligned}
X^2 &\leq K\mathbb{E}\left[\sqrt{\epsilon^2 + 2\sum_{t=0}^{T}\|\mathbf{g}_t - \nabla f(\mathbf{x}_t)\|^2 + 2\sum_{t=0}^{T}\|\nabla f(\mathbf{x}_t)\|^2}\right] \\
&\leq K\mathbb{E}\left[\sqrt{\epsilon^2 + 2\sum_{t=0}^{T}\|\mathbf{g}_t - \nabla f(\mathbf{x}_t)\|^2}\right] + K\sqrt{2}\mathbb{E}\left[\sqrt{\sum_{t=0}^{T}\|\nabla f(\mathbf{x}_t)\|^2}\right] \\
&= K\mathbb{E}\left[\sqrt{\epsilon^2 + 2\sum_{t=0}^{T}\|\mathbf{g}_t - \nabla f(\mathbf{x}_t)\|^2}\right] + K\sqrt{2}X
\end{aligned}$$

Proof IV

And with Jensen:

$$X^2 \le K\sqrt{\epsilon^2 + 2\sum_{t=0}^{T}\mathbb{E}[\|\mathbf{g}_t - \nabla f(\mathbf{x}_t)\|^2]} + K\sqrt{2}X \le K\sqrt{\epsilon^2 + 2(T+1)\sigma^2} + K\sqrt{2}X$$

Now, by the quadratic formula ($ax^2 + bx + c = 0$, $x \le \frac{-b+\sqrt{b^2-4ac}}{2a}$)

$$\begin{aligned} X &\le \frac{K\sqrt{2} + \sqrt{2K^2 + 4K\sqrt{\epsilon^2 + 2(T+1)\sigma^2}}}{2} \\ &\le K\sqrt{2} + \sqrt{K}(\epsilon^2 + 2(T+1)\sigma^2)^{1/4} \\ &\le K\sqrt{2} + \sqrt{K\epsilon} + \sqrt{2K\sigma}(T+1)^{1/4} \end{aligned}$$

Finally, from

$$\frac{1}{\sqrt{T+1}}X \le \frac{K\sqrt{2} + \sqrt{K\epsilon}}{\sqrt{T+1}} + \frac{\sqrt{2K\sigma}}{(T+1)^{1/4}}$$

and squaring both sides the theorem follows. (Note $\sqrt{a+b} \le \sqrt{a} + \sqrt{b}$, $(a+b)^2 \le 2a^2 + 2b^2$)

Proof of Theorem (Lecture-5).4

By smoothness:

$$
\begin{aligned}
f(\mathbf{x}_{t+1}) &\leq f(\mathbf{x}_t) - \gamma_t \nabla f(\mathbf{x}_t)^\top \mathbf{g}_t + \frac{L}{2}\gamma_t^2 \left\|\mathbf{g}_t\right\|^2 \\
&= f(\mathbf{x}_t) - \gamma_{t-1} \nabla f(\mathbf{x}_t)^\top \mathbf{g}_t + (\gamma_{t-1} - \gamma_t) \nabla f(\mathbf{x}_t)^\top \mathbf{g}_t + \frac{L}{2}\gamma_t^2 \left\|\mathbf{g}_t\right\|^2
\end{aligned}
$$

Summing up:

$$
f(\mathbf{x}_{T+1}) \leq f(\mathbf{x}_0) - \sum_{t=0}^{T} \gamma_{t-1} \left\|\nabla f(\mathbf{x}_t)\right\|^2 + \frac{L}{2} \sum_{t=0}^{T} \gamma_t^2 \left\|\mathbf{g}_t\right\|^2 + A
$$

where

$$
\begin{aligned}
A = \sum_{t=0}^{T} (\gamma_{t-1} - \gamma_t) \nabla f(\mathbf{x}_t)^\top \mathbf{g}_t &\leq \max_{t \leq T} \left|\nabla f(\mathbf{x}_t)^\top \mathbf{g}_t\right| \sum_{t=0}^{T} (\gamma_{t-1} - \gamma_t) \\
&\leq \max_{t \leq T} \left\|\nabla f(\mathbf{x}_t)\right\| \max_{t \leq T} \left\|\mathbf{g}_t\right\| \sum_{t=0}^{T} (\gamma_{t-1} - \gamma_t) \leq \max_{t \leq T} \left\|\nabla f(\mathbf{x}_t)\right\| \max_{t \leq T} \left\|\mathbf{g}_t\right\| \gamma_{-1}
\end{aligned}
$$

And the proof follows by taking expectation.