Session 13: Stochastic gradient descent

Optimization and Computational Linear Algebra for Data Science

Léo Miolane

Final exam

- Scope: everything except today's lecture and this week's video.
- Of course, it will be a bit more focused on what we did after the midterm (PCA, linear regression, convex functions, optimization...)
- Same format as for the midterm
- "24 hours window" on Thursday December 17th.
- 1 hour 40 minutes to work + 20 minutes to scan + upload on Gradescope.
- In case you have any issue when uploading: **email me your work**.

Contents

Introduction

Supervised learning

Supervised learning

Supervised learning

Why not using gradient descent ?

$$f(\theta) = \frac{1}{N}\sum_{i=1}^{N} f_i(\theta).$$

Gradient descent iterations:

$$\begin{aligned}\theta_{t+1} &= \theta_t - \alpha_t \nabla f(\theta_t)\\ &= \theta_t - \frac{\alpha_t}{N}\sum_{i=1}^{N}\nabla f_i(\theta_t).\end{aligned}$$

Stochastic gradient descent

Stochastic gradient descent

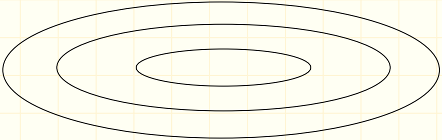
$$f(\theta) = \frac{1}{N} \sum_{i=1}^{N} f_i(\theta).$$

Starting at some $\theta_0 \in \mathbb{R}^n$, perform the updates:

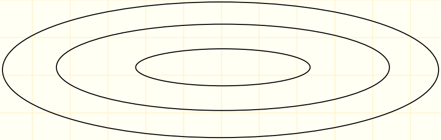
$$\begin{array}{lll} \text{Pick} & i & \text{uniformly at random in} \quad \{1, \ldots, N\}, \\ \text{Update} & \multicolumn{2}{l}{\theta_{t+1} = \theta_t - \alpha_t \nabla f_i(\theta_t),} \end{array}$$

Tradeoffs in SGD

Rapidly decaying step sizes



Slowly decaying step sizes



SGD in practice

Mini-batch stochastic gradient descent:

Pick a mini-batch $i_1, \ldots, i_k$ in $\{1, \ldots, N\}$,

Update $\theta_{t+1} = \theta_t - \frac{\alpha_t}{k} \sum_{m=1}^{k} \nabla f_{i_m}(\theta_t)$,

- Decrease the step size after a fixed number of epochs.
- Use momentum + “adaptive gradient”: Adagrad, RMSprop, Adedelta, Adam, Adamax, Nadam...

Excellent reference:

`https://arxiv.org/pdf/1609.04747.pdf`

Convergence analysis

Convergence rates

- if the f_i are convex and L-smooth: SGD with $\alpha_t = 1/\sqrt{t}$ achieves an error $\leq C/\sqrt{t}$.
- if the f_i are μ-strongly convex and L-smooth: SGD with $\alpha_t = 1/(\mu t)$ achieves an error $\leq C/t$.

GD vs SGD

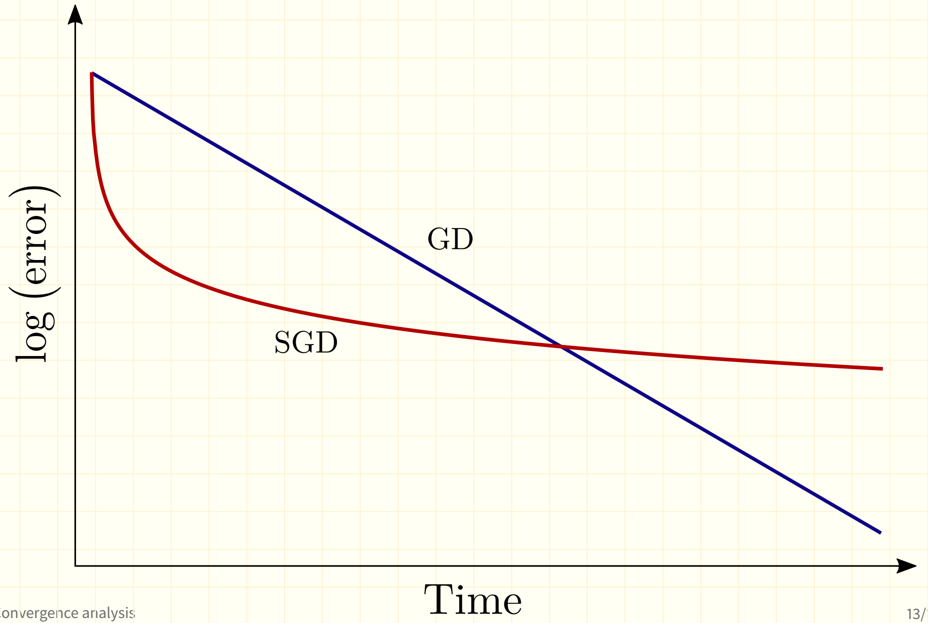
Gradient descent

- Time per step
- Error after t steps
- Log-error after τ units of time

Stochastic gradient descent

- Time per step
- Error after t steps
- Log-error after τ units of time

GD vs SGD



GD vs SGD: who wins ?

- If one is looking for a very small optimization error $f(\theta_t) - \min f$, then gradient descent wins.
- If one has a limited time budget and does not need a very small $f(\theta_t) - \min f$, then stochastic gradient descent wins.

Questions?

Questions?