Optimization for Machine Learning

Lecture 10: Non-Convex Optimization

Sebastian Stich

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

June 25, 2024

Intro Week 10

Recall the convergence proof of GD on smooth functions:

- $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L \|\mathbf{x} - \mathbf{y}\|, \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$
- We proved sufficient decrease/one-step progress for $\mathbf{x}_{t+1} = \mathbf{x}_t - \gamma \nabla f(\mathbf{x}_t)$:

$$f(\mathbf{x}_{t+1}) \leq f(\mathbf{x}_t) - \frac{1}{2L} \|\nabla f(\mathbf{x}_t)\|^2$$

- Without further knowledge on the function class, we get a rate by telescoping:

$$\sum_{t=0}^{T-1} \|\nabla f(\mathbf{x}_t)\|^2 \leq 2L \sum_{t=0}^{T-1} (f(\mathbf{x}_t) - f(\mathbf{x}_{t+1})) \leq 2L(f(\mathbf{x}_0) - f^\star)$$

Q: Which of these steps can possibly be relaxed?

Lecture Outline

Classes of non-convex functions

Trajectory Analysis

Polyak-Łojasiewicz (PŁ) inequality

Strongly convex: x

$y = x^*$

A function satisfies the PŁ inequality if the following holds for a $\mu > 0$:

$$\frac{1}{2}\|\nabla f(\mathbf{x})\|^2 \geq \mu(f(\mathbf{x}) - f^\star) \qquad \forall \mathbf{x} \in \mathbb{R}^d$$

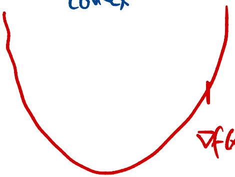
with $f^\star := \min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x})$.

Note: μ-strongly convex functions are μ-PŁ.

Illustration

$f(x) = \frac{1}{2}x^2$

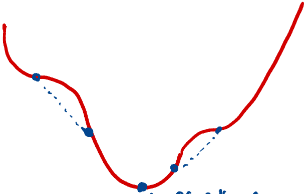
convex



$\nabla f(x) = x$

$\frac{1}{2}x^2 = f(x) - f^* \leq \frac{1}{2}\|\nabla f(x)\|^2 = \frac{1}{2}x^2$

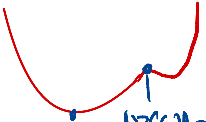
PL function



$\uparrow \|\nabla f(x)\| = 0$

- non convex
- $\|\nabla f(x)\| \sim f(x) - f^*$

not PL !



$\|\nabla f(x)\| = 0$

Linear Convergence with the PŁ condition

Theorem (L-10).1

Let $f : \mathbb{R}^d \to \mathbb{R}$ be differentiable, L-smooth and μ-PŁ for $\mu > 0$. Let $f^\star := \operatorname{argmin}_{\mathbf{x} \in \mathbb{R}^d} f(\mathbf{x})$ and assume $f^\star > -\infty$. Choosing $\gamma = \frac{1}{L}$, gradient descent satisfies the following two properties:

(i) *The function suboptimality is geometrically decreasing:*

$$f(\mathbf{x}_{t+1}) - f^\star \leq \left(1 - \frac{\mu}{L}\right)(f(\mathbf{x}_t) - f^\star) \quad t \geq 0.$$

(ii) *The absolute error after T iterations is exponentially small in T:*

$$f(\mathbf{x}_T) - f^\star \leq \left(1 - \frac{\mu}{L}\right)^T (f(\mathbf{x}_0) - f^\star) \; .$$

Proof

Proof.

Smoothness implies sufficient decrease:

$$f(\mathbf{x}_{t+1}) - f^\star \leq f(\mathbf{x}_t) - f^\star - \frac{1}{2L}\left\|\nabla f(\mathbf{x}_t)\right\|^2$$

Apply PŁ:

$$f(\mathbf{x}_{t+1}) - f^\star \leq \underbrace{f(\mathbf{x}_t) - f^\star - \frac{\mu}{L}\left(f(\mathbf{x}_t) - f^\star\right)}_{(1-\frac{\mu}{L})\cdot[f(x_t)-f^*]}$$

and the first (and second) claim follows. □

Star Convexity

A differentiable function is quasi convex (or star convex) with respect to $\mathbf{x}^\star$ if the following holds for a $\mu > 0$:

$$f(\mathbf{x}) - f(\mathbf{x}^\star) + \frac{\mu}{2}\|\mathbf{x} - \mathbf{x}^\star\|^2 \leq \nabla f(\mathbf{x})^\top (\mathbf{x} - \mathbf{x}^\star) \qquad \forall \mathbf{x} \in \mathbb{R}^d$$

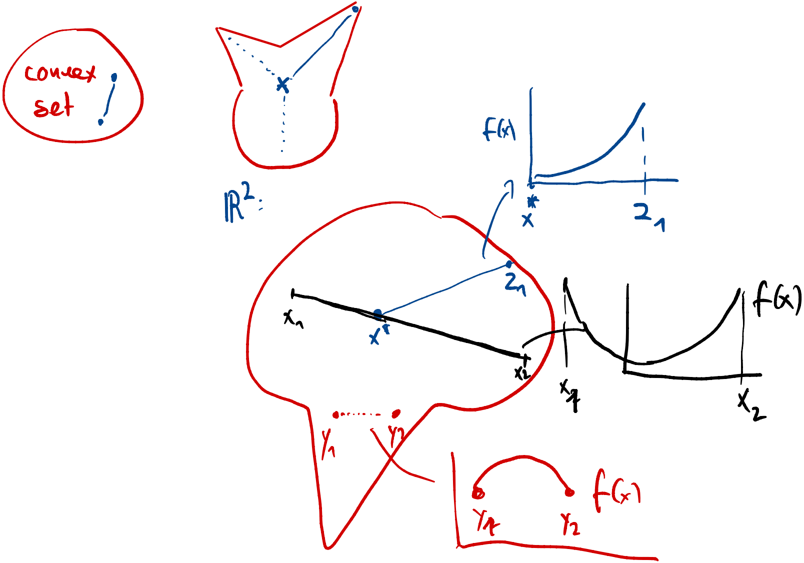
strongly convex functions for any x, x^* pair, here: x^* is fixed!

Note I: $\mathbf{x}^\star = \operatorname{argmin}_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x})$.

Note II: μ-strongly convex functions are μ-quasi convex wrt. to the minimizer $\mathbf{x}^\star$.

Note III: μ-quasi convex functions are μ-PŁ.

Illustration



Linear Convergence for quasi-convex functions

Theorem (L-10).2

Let $f : \mathbb{R}^d \to \mathbb{R}$ be differentiable, L-smooth and μ-quasi convex with respect to a point $\mathbf{x}^\star$ for $\mu > 0$. Choosing $\gamma = \frac{1}{L}$, gradient descent satisfies the following two properties:

(i) *The square distance to the minimizer is geometrically decreasing:*

$$\|\mathbf{x}_{t+1} - \mathbf{x}^\star\| \leq \left(1 - \frac{\mu}{L}\right) \|\mathbf{x}_t - \mathbf{x}^\star\|^2 \quad t \geq 0.$$

(ii) *The absolute error after T iterations is exponentially small in T:*

$$\|\mathbf{x}_T - \mathbf{x}^\star\| \leq \left(1 - \frac{\mu}{L}\right)^T \|\mathbf{x}_0 - \mathbf{x}^\star\|^2 \ .$$

Proof.

Expand:

$$\|\mathbf{x}_{t+1}-\mathbf{x}^\star\|^2 = \|\mathbf{x}_t-\mathbf{x}^\star\|^2 - 2\gamma\nabla f(\mathbf{x}_t)^\top(\mathbf{x}_t-\mathbf{x}^\star) + \gamma^2\|\nabla f(\mathbf{x}_t)\|^2$$

↗ here we used convexity!, now: μ-star convexity!

Apply the μ-quasi convex inequality for the middle term (and smoothness to bound $\|\nabla f(\mathbf{x})\| \leq 2L(f(\mathbf{x}_t)-f(\mathbf{x}^\star))$:

$$\begin{aligned}\|\mathbf{x}_{t+1}-\mathbf{x}^\star\|^2 &\leq \|\mathbf{x}_t-\mathbf{x}^\star\|^2 - \mu\gamma\|\mathbf{x}_t-\mathbf{x}^\star\|^2 - 2\gamma(f(\mathbf{x}_t)-f^\star) + 2\gamma^2L(f(\mathbf{x}_t)-f^\star)\\ &= (1-\mu\gamma)\|\mathbf{x}_t-\mathbf{x}^\star\|^2 + 2\underbrace{(\gamma^2L-\gamma)}_{=0}(f(\mathbf{x}_t)-f^\star)\\ &= \left(1-\frac{\mu}{L}\right)\|\mathbf{x}_t-\mathbf{x}^\star\|^2\end{aligned}$$

□

Graded non-convex functions

Assume $f\colon \mathbb{R}^d \to \mathbb{R}$ is twice differentiable. (, and bounded.) For an integer $\tau \geq 1$:

$$f \text{ is non-convex of grade } \tau \leftrightarrow \underbrace{\nabla^2_\tau f(\mathbf{x})}_{\tau \text{ largest eigenvalues are } \geq 0} \succeq 0$$

where

$$\nabla^2_\tau f(\mathbf{x}) = \sum_{i=1}^{\tau} \lambda_i(\mathbf{x}) \cdot \mathbf{u}_i(\mathbf{x})\mathbf{u}_i(\mathbf{x})^\top$$

for eigenvalues/eigenvector pairs $(\lambda_i, \mathbf{u}_i)$, $\lambda_1 \geq \cdots \geq \lambda_n$.

Observation:

$$\boxed{\underset{\textbf{all smooth functions}}{\mathcal{F}_0} \supset \mathcal{F}_1 \supset \cdots \supset \mathcal{F}_{n-1} \supset \underset{\textbf{convex functions}}{\mathcal{F}_n}}$$

$$\underset{\text{all eigenvalues positive}}{\parallel}$$

Illustration

Properties

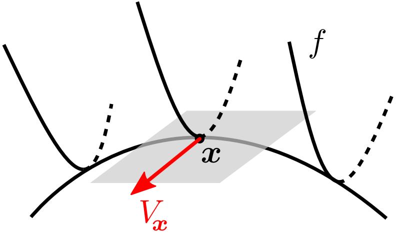
- A function $f \in \mathcal{F}_\tau$ for $\tau \geq 1$ cannot have a local maximum. For any compact C:

$$\max_{\mathbf{x}\in C} f(\mathbf{x}) = \max_{\mathbf{x}\in \partial C} f(\mathbf{x})$$

- Suppose that for every $\mathbf{x}$ there exists a subspace $V_\mathbf{x} \subset \mathbb{R}^n$ with $\dim(V_\mathbf{x}) \geq \tau$, such that

$$f(\mathbf{x}+\mathbf{h}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^\top \mathbf{h} \qquad \forall \mathbf{h} \in V_\mathbf{x} \,.$$

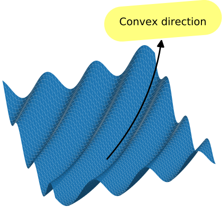
Then $f \in \mathcal{F}_\tau$.



Examples

- Low rank vector fields, $f(\mathbf{x}) = \psi(\mathbf{u}^\top \mathbf{x})$, for $\mathbf{u} \in \mathbb{R}^n$. Then $f \in \mathcal{F}_{n-1}$.

loss: $\| y - \underbrace{g(x)}_{\text{first layers / NN}} \|^2$



$\sin(x+y) + q(x,y)$

- **Convex Loss Functions.** Consider $f(\mathbf{x}, \mathbf{y})$, $\mathbf{x} \in \mathbb{R}^m$, $\mathbf{y} \in \mathbb{R}^n$. Suppose that for any fixed $\mathbf{y}$, $f(\cdot, \mathbf{y})$ is convex. Then $f \in \mathcal{F}_n$.
- **Matrix Factorization.** Consider $f(\mathbf{X}_1, \cdots, \mathbf{X}_d) = \frac{1}{2} \| \mathbf{X}_1 \cdots \mathbf{X}_d - \mathbf{C} \|_F^2$ with $\mathbf{X}_i \in \mathbb{R}^{n_i \times m_i}$, is non-convex with grade $\tau \geq \max_i (n_i \cdot m_i)$.

$f(x_1, \dots x_d) = \frac{1}{2} \| x_1 \cdot \dots \cdot x_d - c \|^2 \qquad x_i \in \mathbb{R}$

Lecture Outline

Classes of non-convex functions

Trajectory Analysis

Trajectory Analysis

Even if the “landscape” (graph) of a nonconvex function has local minima, saddle points, and flat parts, gradient descent may avoid them and still converge to a global minimum.

For this, one needs a good starting point and some theoretical understanding of what happens when we start there—this is **trajectory analysis**.

2018: trajectory analysis for training deep linear linear neural networks, under suitable conditions [ACGH18].

Here: vastly simplified setting that allows us to show the main ideas (and limitations).

> Disclaimer: We will not be able to cover all details in this lecture; we will only go over the high-level concepts. Please refer to Chapter 5 in the lecture notes if you are interested in this topic.

Linear models with several outputs

Recall: Learning linear models

- n inputs $\mathbf{x}_1, \ldots, \mathbf{x}_n$, where each input $\mathbf{x}_i \in \mathbb{R}^d$
- n outputs $y_1, \ldots, y_n \in \mathbb{R}$
- Hypothesis (after centering):

$$y_i \approx \mathbf{w}^\top \mathbf{x}_i,$$

  for a weight vector $\mathbf{w} = (w_1, \ldots, w_d) \in \mathbb{R}^d$ to be learned.

Now more than one output value:

- n outputs $\mathbf{y}_1, \ldots, \mathbf{y}_n$, where each output $\mathbf{y}_i \in \mathbb{R}^m$
- Hypothesis:

$$\mathbf{y}_i \approx W \mathbf{x}_i,$$

  for a weight matrix $W \in \mathbb{R}^{m \times d}$ to be learned.

Minimizing the least squares error

Compute

$$W^\star = \operatorname*{argmin}_{W \in \mathbb{R}^{m \times d}} \sum_{i=1}^{n} \|W\mathbf{x}_i - \mathbf{y}_i\|^2 .$$

- $X \in \mathbb{R}^{d \times n}$: matrix whose columns are the $\mathbf{x}_i$
- $Y \in \mathbb{R}^{m \times n}$: matrix whose columns are the $\mathbf{y}_i$

Then

$$W^\star = \operatorname*{argmin}_{W \in \mathbb{R}^{m \times d}} \|WX - Y\|_F^2,$$

where $\|A\|_F = \sqrt{\sum_{i,j} a_{ij}^2}$ is the Frobenius norm of a matrix A.

Frobenius norm of A = Euclidean norm of $\mathrm{vec}(A)$ ("flattening" of A)

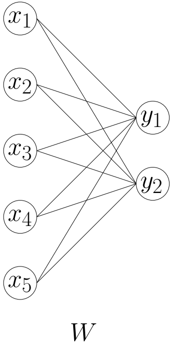
Minimizing the least squares error II

$$W^{\star} = \underset{W \in \mathbb{R}^{m \times d}}{\operatorname{argmin}} \|WX - Y\|_F^2$$

is the global minimum of a convex quadratic function $f(W)$.

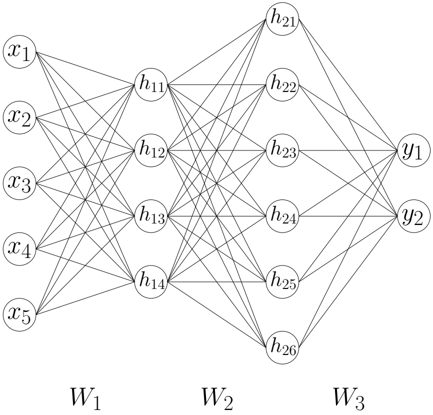
To find $W^{\star}$, solve $\nabla f(W) = \mathbf{0}$ (system of linear equations).

$\Leftrightarrow$ training a **linear neural network with one layer** under least squares error.



$$\mathbf{x} \mapsto \mathbf{y} = W\mathbf{x}$$

Deep linear neural networks



$$\mathbf{x} \mapsto \mathbf{y} = W_3 W_2 W_1 \mathbf{x}$$

Not more expressive:

$$\mathbf{x} \mapsto \mathbf{y} = W_3 W_2 W_1 \mathbf{x} \quad \Leftrightarrow \quad \mathbf{x} \mapsto \mathbf{y} = W\mathbf{x},\ W := W_3 W_2 W_1.$$

Training deep linear neural networks

With ℓ layers:

$$W^\star = \underset{W_1, W_2, \ldots, W_\ell}{\operatorname{argmin}} \|W_\ell W_{\ell-1} \cdots W_1 X - Y\|_F^2,$$

Nonconvex function for $\ell > 1$.

Simple playground in which we can try to understand why training deep neural networks with gradient descent works.

Here: all matrices are 1×1, $W_i = x_i, X = 1, Y = 1, \ell = d \quad \Rightarrow f : \mathbb{R}^d \to \mathbb{R}$,

$$f(\mathbf{x}) := \frac{1}{2}\left(\prod_{k=1}^{d} x_k - 1\right)^2.$$

$x \in \mathbb{R}^d$

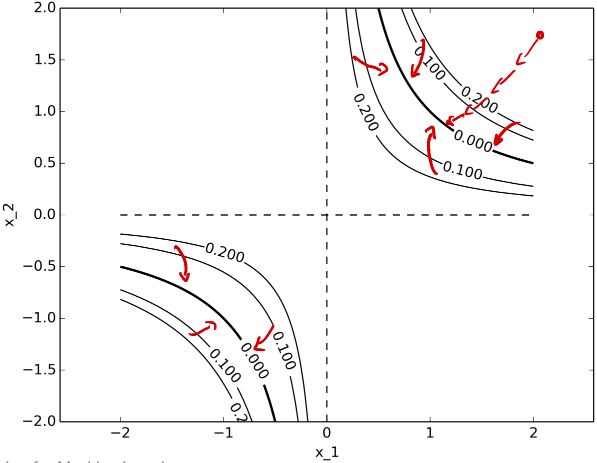
$x_i \in \mathbb{R}$

Toy example in our simple playground.

But analysis of gradient descent on f has similar ingredients as the one on general deep linear neural networks [ACGH18].

A simple nonconvex function

As d is fixed, abbreviate $\prod_{k=1}^{d} x_k$ by $\prod_k x_k$: $f(\mathbf{x}) = \frac{1}{2}\left(\prod_k x_k - 1\right)^2$

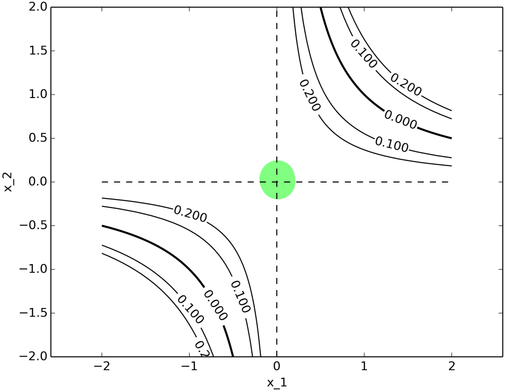


Level set plot

The gradient

$$\frac{d}{dx_1} x_1 x_2 x_3 x_4 = x_2 x_3 x_4$$

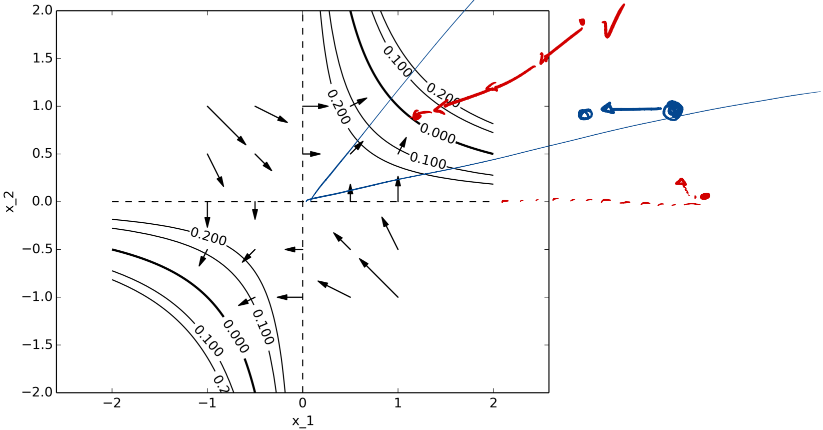
$$\nabla f(\mathbf{x}) = \left(\prod_k x_k - 1\right)\left(\prod_{k\neq 1} x_k, \dots, \prod_{k\neq d} x_k\right).$$



Critical points ($\nabla f(\mathbf{x}) = \mathbf{0}$):

- $\prod_k x_k = 1$ (global minima)
  - $d = 2$: the hyperbola $\{(x_1, x_2) : x_1 x_2 = 1\}$
- at least two of the x_k are zero (saddle points)
  - $d = 2$: the origin $(x_1, x_2) = (0, 0)$

Negative gradient directions (followed by gradient descent)



Difficult to avoid convergence to a global minimum, but it is possible (Exercise 35).

Convergence analysis: Overview

Want to show that for any $d > 1$, and from anywhere in $X = \{\mathbf{x} : \mathbf{x} > \mathbf{0}, \prod_k \mathbf{x}_k \leq 1\}$, gradient descent will converge to a global minimum.

f is not smooth over X. We show that f is smooth along the trajectory of gradient descent for suitable L, so that we get sufficient decrease

$$f(\mathbf{x}_{t+1}) \leq f(\mathbf{x}_t) - \frac{1}{2L}\|\nabla f(\mathbf{x}_t)\|^2, \quad t \geq 0.$$

Then, we cannot converge to a saddle point: all these have (at least two) zero entries and therefore function value $1/2$. But for starting point $\mathbf{x}_0 \in X$, we have $f(\mathbf{x}_0) < 1/2$, so we can never reach a saddle while decreasing f.

Doesn't this imply converge to a global mimimum? No!

- Sublevel sets are unbounded, so we could in principle run off to infinity.
- Other bad things might happen (we haven't characterized what can go wrong).

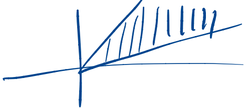
Convergence analysis: Overview II

For $\mathbf{x} > \mathbf{0}, \prod_k \mathbf{x}_k \geq 1$, we also get convergence (Exercise 34).

$\Rightarrow$ convergence from anywhere in the interior of the positive orthant $\{\mathbf{x} : \mathbf{x} > \mathbf{0}\}$.

But there are also starting points from which gradient descent will not converge to a global minimum (Exercise 35).

Main tool: Balanced iterates



Definition (L-10).3

Let $\mathbf{x} > \mathbf{0}$ (componentwise), and let $c \geq 1$ be a real number. $\mathbf{x}$ is called c-balanced if $x_i \leq c x_j$ for all $1 \leq i, j \leq d$.

Any initial iterate $\mathbf{x}_0 > \mathbf{0}$ is c-balanced for some (possibly large) c.

Lemma (L-10).4

Let $\mathbf{x} > \mathbf{0}$ be c-balanced with $\prod_k x_k \leq 1$. Then for any stepsize $\gamma > 0$, $\mathbf{x}' := \mathbf{x} - \gamma \nabla f(\mathbf{x})$ satisfies $\mathbf{x}' \geq \mathbf{x}$ (componentwise) and is also c-balanced.

Proof.

$\Delta := -\gamma(\prod_k x_k - 1)(\prod_k x_k) \geq 0.$ $\quad \nabla f(\mathbf{x}) = (\prod_k x_k - 1)\left(\prod_{k \neq 1} x_k, \ldots, \prod_{k \neq d} x_k\right).$

Gradient descent step:

$$x'_k = x_k + \frac{\Delta}{x_k} \geq x_k, \quad k = 1, \ldots, d.$$

For i, j, we have $x_i \leq c x_j$ and $x_j \leq c x_i$ ($\Leftrightarrow 1/x_i \leq c/x_j$). We therefore get

$$x'_i = x_i + \frac{\Delta}{x_i} \leq c x_j + \frac{\Delta c}{x_j} = c x'_j.$$

□

How to use the main tool

c-balanced iterates allow to prove:

- the Hessian $\nabla^2 f(\mathbf{x})$ is bounded along the trajectory of gradient descent

$$\left\|\nabla^2 f(\mathbf{x}_t)\right\| \leq 3dc^2$$

- the function f is smooth with parameter $L = 3dc^2$ along the trajectory of gradient descent with stepsize $\gamma = 1/L$.

Convergence

Theorem (L-10).5

Let $c \geq 1$ and $\delta > 0$ such that $\mathbf{x}_0 > \mathbf{0}$ is c-balanced with $\delta \leq \prod_k (\mathbf{x}_0)_k < 1$. Choosing stepsize

$$\gamma = \frac{1}{3dc^2}, \text{"} = \frac{1}{L} \text{"}$$

gradient descent satisfies

$$f(\mathbf{x}_T) \leq \left(1 - \frac{\delta^2}{3c^4}\right)^T f(\mathbf{x}_0), \quad T \geq 0.$$

↑ Linear convergence ! (?)

- Error converges to 0 exponentially fast.
- Exercise 37: iterates themselves converge (to an optimal solution).

Convergence: Proof

Proof.

- For $t \geq 0$, f is smooth between $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$ with parameter $L = 3dc^2$.
- Sufficient decrease:

$$f(\mathbf{x}_{t+1}) \leq f(\mathbf{x}_t) - \frac{1}{6dc^2} \|\nabla f(\mathbf{x}_t)\|^2 .$$

For every c-balanced $\mathbf{x}$ with $\delta \leq \prod_k x_k \leq 1$, $\|\nabla f(\mathbf{x})\|^2$ equals

$$2f(\mathbf{x}) \sum_{i=1}^{d} \left(\prod_{k \neq i} x_k \right)^2 \geq 2f(\mathbf{x}) \frac{d}{c^2} \left(\prod_k x_k \right)^{2-2/d} \geq 2f(\mathbf{x}) \frac{d}{c^2} \left(\prod_k x_k \right)^2 \geq 2f(\mathbf{x}) \frac{d}{c^2} \delta^2 .$$

- Hence, $f(\mathbf{x}_{t+1}) \leq f(\mathbf{x}_t) - \frac{1}{6dc^2} 2f(\mathbf{x}_t) \frac{d}{c^2} \delta^2 = f(\mathbf{x}_t) \left(1 - \frac{\delta^2}{3c^4} \right)$.

□

Discussion

Fast convergence as for strongly convex functions!

But there is a catch...

Consider starting point $\mathbf{x}_0 = (1/2, \ldots, 1/2)$.

$\delta \leq \prod_k (\mathbf{x}_0)_k = 2^{-d}$.

Decrease in function value by a factor of

$$\left(1 - \frac{1}{3 \cdot 4^d}\right),$$

per step.

Need $T \approx 4^d$ to reduce the initial error by a constant factor not depending on d.

Problem: gradients are exponentially small in the beginning, extremely slow progress.

For polynomial runtime, must start at distance $O(1/\sqrt{d})$ from optimality.

Bibliography

Sanjeev Arora, Nadav Cohen, Noah Golowich, and Wei Hu.
A convergence analysis of gradient descent for deep linear neural networks.
CoRR, abs/1810.02281, 2018.