Optimization for Data Science
ETH Zürich, FS 2023 261-5110-00L

Lecture 6: Nonconvex Functions

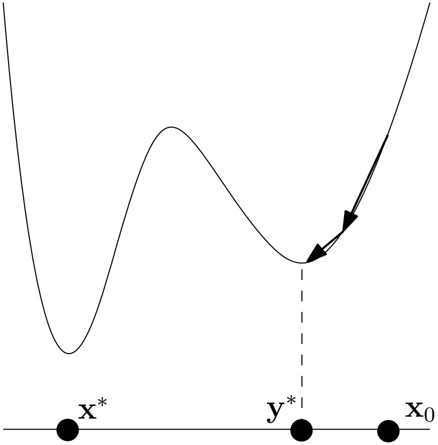
Bernd Gärtner
Niao He

`https://www.ti.inf.ethz.ch/ew/courses/ODS23/index.html`

March 20, 2023

Gradient Descent in the nonconvex world

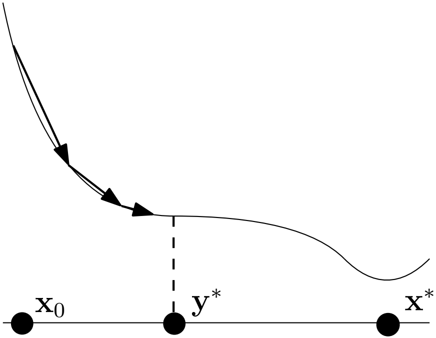
- may get stuck in a local minimum and miss the global minimum.

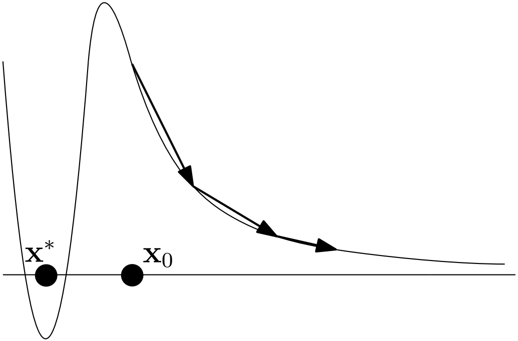


Gradient Descent in the nonconvex world II

Even if there is a unique local minimum (equal to the global minimum), we

- may get stuck in a saddle point;
- run off to infinity;
- possibly encounter other bad behaviors.





Gradient Descent in the nonconvex world III

Often, we observe good behavior in practice.

Theoretical explanations are mostly missing.

This lecture:

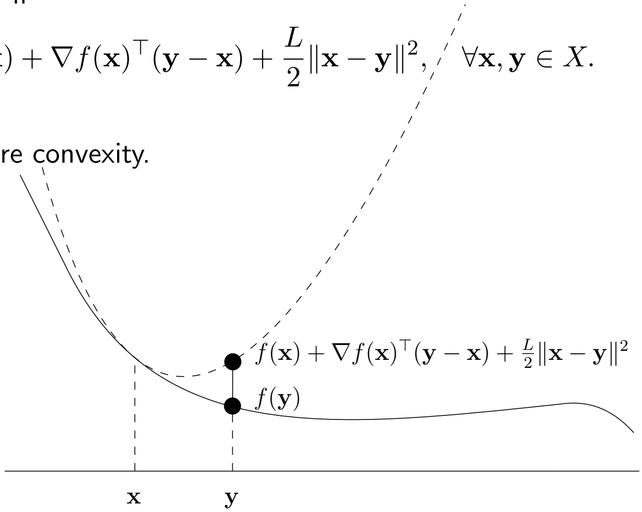
- Good news: Under favorable conditions, we sometimes can say something useful about the behavior of gradient descent, even on nonconvex functions.
- Bad news: It is computationally hard to decide whether a critical point (reached through gradient descent or any other method) is a local minimum.

Smooth (but not necessarily convex) functions

Recall: A differentiable $f : \mathbf{dom}(f) \to \mathbb{R}$ is smooth with parameter $L \in \mathbb{R}_+$ over a convex set $X \subseteq \mathbf{dom}(f)$ if

$$f(\mathbf{y}) \leq f(\mathbf{x}) + \nabla f(\mathbf{x})^\top (\mathbf{y} - \mathbf{x}) + \frac{L}{2}\|\mathbf{x} - \mathbf{y}\|^2, \quad \forall \mathbf{x}, \mathbf{y} \in X. \tag{1}$$

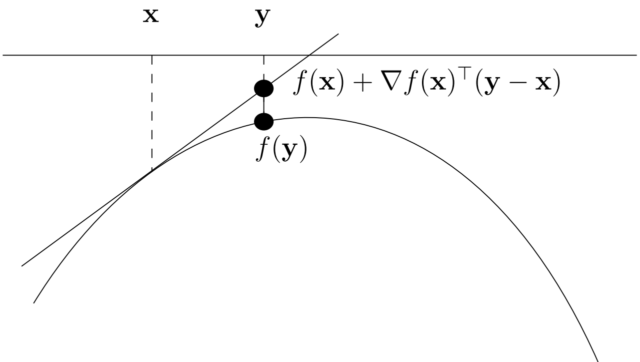
Definition does not require convexity.



Concave functions

f is called **concave** if $-f$ is convex.

For all $\mathbf{x}$, the graph of a differentiable concave function is below the tangent hyperplane at $\mathbf{x}$.



$\Rightarrow$ concave functions are smooth with $L = 0\ldots$ but boring from an optimization point of view (no global minimum), gradient descent runs off to infinity

Bounded Hessians $\Rightarrow$ smooth

A class of interesting smooth functions:

Lemma 6.1

Let $f : \mathbf{dom}(f) \rightarrow \mathbb{R}$ be twice differentiable, with $X \subseteq \mathbf{dom}(f)$ a convex set, and $\left\|\nabla^2 f(\mathbf{x})\right\| \leq L$ for all $\mathbf{x} \in X$, where $\|\cdot\|$ is spectral norm. Then f is smooth with parameter L over X.

Examples:

- all quadratic functions $f(\mathbf{x}) = \mathbf{x}^\top A\mathbf{x} + \mathbf{b}^\top \mathbf{x} + c$
- $f(x) = \sin(x)$ (many global minima)

Bounded Hessians $\Rightarrow$ smooth II

Proof.

By Theorem 2.10 (applied to the gradient function ∇f), bounded Hessians imply Lipschitz continuity of the gradient,

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L \|\mathbf{x} - \mathbf{y}\|, \quad \mathbf{x}, \mathbf{y} \in X.$$

To show that this implies smoothness, we use the fundamental theorem of calculus $h(1) - h(0) = \int_0^1 h'(t)dt$ with

$$h(t) := f\big(\mathbf{x} + t(\mathbf{y} - \mathbf{x})\big), \quad t \in [0, 1].$$

Chain rule:

$$h'(t) = \nabla f\big(\mathbf{x} + t(\mathbf{y} - \mathbf{x})\big)^\top (\mathbf{y} - \mathbf{x}).$$

Note that h' is continuous since f is twice differentiable.

Bounded Hessians $\Rightarrow$ smooth III

Proof.

For $\mathbf{x}, \mathbf{y} \in X$:

$$
\begin{aligned}
& f(\mathbf{y}) - f(\mathbf{x}) - \nabla f(\mathbf{x})^\top (\mathbf{y} - \mathbf{x}) \\
= \; & h(1) - h(0) - \nabla f(\mathbf{x})^\top (\mathbf{y} - \mathbf{x}) \quad \text{(definition of } h\text{)} \\
= \; & \int_0^1 h'(t)dt - \nabla f(\mathbf{x})^\top (\mathbf{y} - \mathbf{x}) \quad \text{(fundamental theorem)} \\
= \; & \int_0^1 \nabla f(\mathbf{x} + t(\mathbf{y} - \mathbf{x}))^\top (\mathbf{y} - \mathbf{x}) dt - \nabla f(\mathbf{x})^\top (\mathbf{y} - \mathbf{x}) \\
= \; & \int_0^1 \left(\nabla f(\mathbf{x} + t(\mathbf{y} - \mathbf{x}))^\top (\mathbf{y} - \mathbf{x}) - \nabla f(\mathbf{x})^\top (\mathbf{y} - \mathbf{x}) \right) dt \\
= \; & \int_0^1 \left(\nabla f(\mathbf{x} + t(\mathbf{y} - \mathbf{x})) - \nabla f(\mathbf{x}) \right)^\top (\mathbf{y} - \mathbf{x}) dt
\end{aligned}
$$

Bounded Hessians $\Rightarrow$ smooth IV

Proof.

For $\mathbf{x}, \mathbf{y} \in X$:

$$
\begin{aligned}
& f(\mathbf{y}) - f(\mathbf{x}) - \nabla f(\mathbf{x})^\top(\mathbf{y}-\mathbf{x}) \\
= & \int_0^1 \big(\nabla f(\mathbf{x}+t(\mathbf{y}-\mathbf{x})) - \nabla f(\mathbf{x})\big)^\top(\mathbf{y}-\mathbf{x})dt \\
\leq & \int_0^1 \left|\big(\nabla f(\mathbf{x}+t(\mathbf{y}-\mathbf{x})) - \nabla f(\mathbf{x})\big)^\top(\mathbf{y}-\mathbf{x})\right|dt \\
\leq & \int_0^1 \left\|\big(\nabla f(\mathbf{x}+t(\mathbf{y}-\mathbf{x})) - \nabla f(\mathbf{x})\big)\right\| \|(\mathbf{y}-\mathbf{x})\| \, dt \quad \text{(Cauchy-Schwarz)} \\
\leq & \int_0^1 L \|t(\mathbf{y}-\mathbf{x})\| \|(\mathbf{y}-\mathbf{x})\| \, dt \quad \text{(Lipschitz continuous gradients)} \\
= & \int_0^1 Lt \|\mathbf{x}-\mathbf{y}\|^2 \, dt = \frac{L}{2} \|\mathbf{x}-\mathbf{y}\|^2 . \quad \text{This is smoothness!}
\end{aligned}
$$

$\square$

Smooth $\Rightarrow$ bounded Hessians?

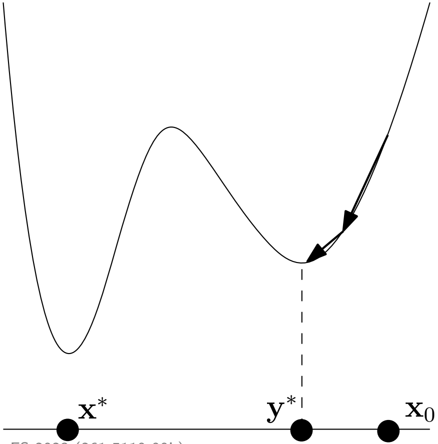
Yes, over any open convex set X (Exercise 40).

Gradient descent on smooth functions

Will prove: $\|\nabla f(\mathbf{x}_t)\|^2 \to 0$ for $t \to \infty \ldots$

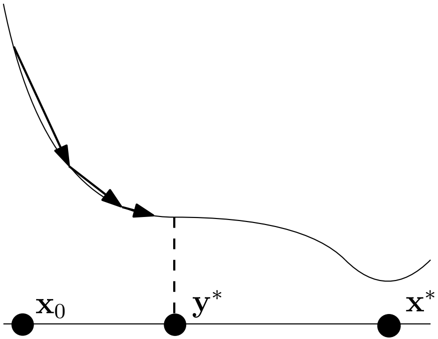
$\ldots$ at the same rate as $f(\mathbf{x}_t) - f(\mathbf{x}^\star) \to 0$ in the convex case.

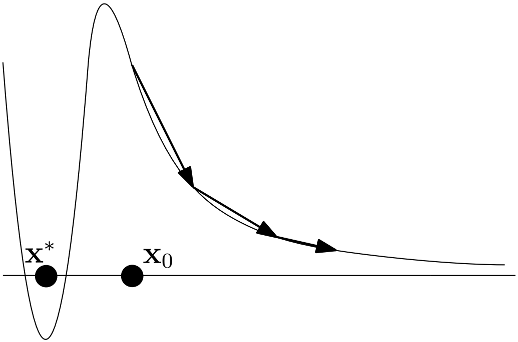
$f(\mathbf{x}_t) - f(\mathbf{x}^\star)$ itself may not converge to 0 in the nonconvex case:



What does $\|\nabla f(\mathbf{x}_t)\|^2 \to 0$ mean?

It may or may not mean that we converge to a **critical point** ($\nabla f(\mathbf{y}^\star) = \mathbf{0}$)





Gradient descent on smooth (not necessarily convex) functions

Theorem 6.2

Let $f : \mathbb{R}^d \to \mathbb{R}$ be differentiable with a global minimum $\mathbf{x}^\star$; furthermore, suppose that f is smooth with parameter L according to Definition 3.2. Choosing stepsize

$$\gamma := \frac{1}{L},$$

gradient descent yields

$$\frac{1}{T}\sum_{t=0}^{T-1} \|\nabla f(\mathbf{x}_t)\|^2 \leq \frac{2L}{T}\big(f(\mathbf{x}_0) - f(\mathbf{x}^\star)\big), \quad T > 0.$$

In particular, $\|\nabla f(\mathbf{x}_t)\|^2 \leq \frac{2L}{T}\big(f(\mathbf{x}_0) - f(\mathbf{x}^\star)\big)$ for some $t \in \{0, \ldots, T-1\}$.
And also, $\lim_{t\to\infty} \|\nabla f(\mathbf{x}_t)\|^2 = 0$ (Exercise 41).

Gradient descent on smooth (not necessarily convex) functions II

Proof.

Sufficient decrease (Lemma 3.7), does not require convexity:

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

Rewriting:

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

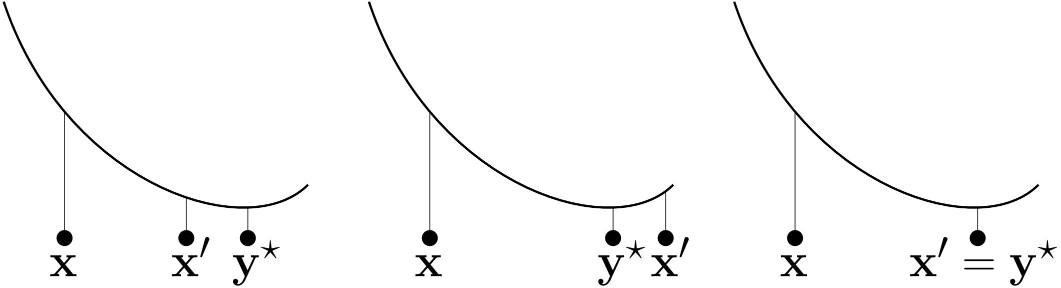
Telescoping sum:

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

The statement follows (divide by T). $\square$

No overshooting

In the smooth setting, and with stepsize $1/L$, gradient descent cannot overshoot, i.e. pass a critical point (Lemma 6.3, Exercise 42).



$\mathbf{x}' = \mathbf{x} - \gamma \nabla f(\mathbf{x}), \gamma < 1/L$ | overshooting | may happen with $\gamma = 1/L$

Be critical with critical points...

Suppose that we have found a critical point $\tilde{\mathbf{x}}$ of a (nonconvex) f, i.e. $\nabla f(\tilde{\mathbf{x}}) = \mathbf{0}$.

What can we say about $\tilde{\mathbf{x}}$, and how?

- Is $\tilde{\mathbf{x}}$ a global minimum? Probably hard to tell from local information.
- Is $\tilde{\mathbf{x}}$ a local minimum? We will see: this is coNP-complete already for a rather simple class of functions (with derivatives of all orders).
- Any optimization method might reach a critical point where it is computationally hard to distinguish between a local minimum and a saddle point.

$\Rightarrow$ Be skeptical when a method "guarantees" convergence to a local minimum!

Typical documentations...

TensorFlow

- `objective_value` : A tensor containing the value of the objective function at the `position` . If the search converged, then this is the (local) minimum of the objective function.

MathWorks®

Documentation

CONTENTS Schließen

« Documentation Home

« Optimization Toolbox

« Getting Started with Optimization Toolbox

« Optimization Toolbox

Local vs. Global Optima

Why Didn't the Solver Find the Smallest Minimum?

In general, solvers return a local minimum. The result might be a global
This section describes why solvers behave this way, and gives suggesti
if needed.

Local optimality is hard

Let $\mathcal{F}$ be a class of functions from $\mathbb{R}^n$ to $\mathbb{R}$.

The problem $\textsc{LocMin}(\mathcal{F})$ is to decide whether $\mathbf{0}$ is a local minimum of a given function $\phi \in \mathcal{F}$.

Theorem (first proved by Murty and Kabadi [MK87])

The problem $\textsc{LocMin}(\mathcal{F})$ *is coNP-complete for the class* $\mathcal{F} := \{\phi_M : M \text{ symmetric}\}$*, where the function* ϕ_M *is defined by*

$$\phi_M(\mathbf{x}) = (\mathbf{x}^2)^\top M(\mathbf{x}^2),$$

with $\mathbf{x}^2 = (x_1^2, x_2^2, \ldots, x_n^2)$*.*

Proof outline:

- $\mathbf{0}$ is a local minimum if and only if the matrrix M is copositive.
- Deciding whether M is copositive is coNP-complete.

Copositive matrices

Lemma

$\mathbf{0}$ is a local minimum of $(\mathbf{x}^2)^\top M(\mathbf{x}^2)$ if and only if $\mathbf{x}^\top M\mathbf{x} \geq 0$ for all $\mathbf{x} \geq \mathbf{0}$.

Proof.

$$\begin{aligned}
& \mathbf{0} \text{ is a local minimum} \\
\Leftrightarrow\quad & (\mathbf{x}^2)^\top M(\mathbf{x}^2) \geq 0 \text{ for all } \mathbf{x} \text{ in some neighborhood of } \mathbf{0} \\
\Leftrightarrow\quad & \mathbf{x}^\top M\mathbf{x} \geq 0 \text{ for all } \mathbf{x} \geq \mathbf{0} \text{ in some neighborhood of } \mathbf{0} \\
\Leftrightarrow\quad & \mathbf{x}^\top M\mathbf{x} \geq 0 \text{ for all } \mathbf{x} \geq \mathbf{0}
\end{aligned}$$

□

A matrix M satisfying $\mathbf{x}^\top M\mathbf{x} \geq 0$ for all $\mathbf{x} \geq \mathbf{0}$ is called copositive.

Observation

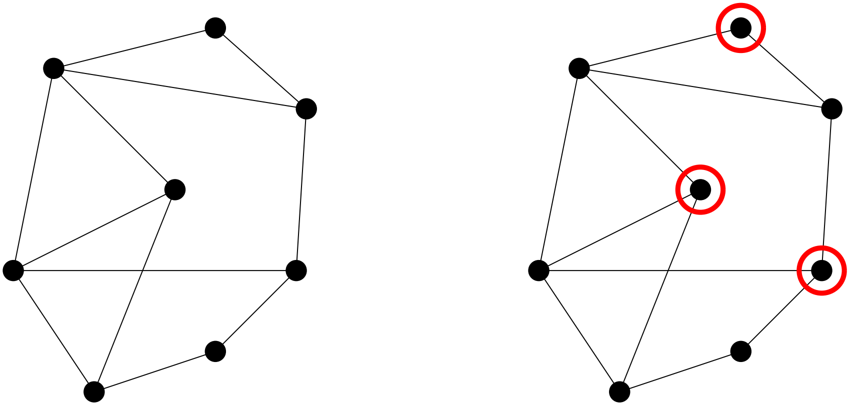
If M is positive semidefinite ($\mathbf{x}^\top M\mathbf{x} \geq 0$ for all $\mathbf{x}$), then M is copositive. The converse is false:

$$M = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

is copositive but not positive semidefinite ($\mathbf{x} = (1, -1) \Rightarrow \mathbf{x}^T M\mathbf{x} = -2$).

Independent Set: a classical NP-complete problem

Given a graph $G = (V, E)$ and a natural number k, decide whether G contains an independent set of size larger than k.



Right: an independent set (set of vertices with no edges between them) of size 3.
In this example: is there an independent set of size larger than 3?

Solving Independent Set by optimization

Let $G = (V, E)$ be a graph with n vertices $V = \{1, 2, \ldots, n\}$.

$\alpha(G)$: the independence number of G, the size of the maximum independent set in G.

A_G: the adjacency matrix of G,

$$(A_G)_{ij} = \begin{cases} 1, & \text{if } \{i, j\} \in E \\ 0, & \text{otherwise.} \end{cases}$$

$\mathbb{I}_n$: the $(n \times n)$ identity matrix.

Theorem (Motzkin–Straus [MS65])

$$\frac{1}{\alpha(G)} = \min\{\mathbf{x}^T(A_G + \mathbb{I}_n)\mathbf{x} : \mathbf{x} \geq \mathbf{0}, \sum_{i=1}^{n} x_i = 1\}.$$

The independence number can be computed by minimizing a quadratic function over the standard simplex! Doesn't sound like a difficult task, but Motzkin-Straus says it is.

Copositivity (and hence LocMin) is hard

Theorem

Given a symmetric matrix M, it is coNP-complete to decide whether M is copositive.

Proof.

coNP membership: any $\mathbf{x} > \mathbf{0}$ such that $\mathbf{x}^T M\mathbf{x} < 0$ proves that M is not copositive. (Exercise: there is a such an $\mathbf{x}$ of encoding size polynomial in the encoding size of M).

coNP-completeness: reduction from the independent set problem: given a graph G and an integer k, does G have an independent set of size larger than k?

Construct matrix $M(G, k) = kA_G + k\mathbb{I}_n - \mathbb{J}_n$ ($\mathbb{J}_n$ is the $(n \times n)$ all-1 matrix).

Claim: $M(G, k)$ is copositive if and only if

$$\underbrace{\min\{\mathbf{x}^T(A_G + \mathbb{I}_n)\mathbf{x} : \mathbf{x} \geq \mathbf{0}, \sum_{i=1}^n x_i = 1\}}_{=1/\alpha(G) \text{ by Motzkin-Straus}} \geq \frac{1}{k}.$$

Hence, $\alpha(G) > k$ if and only if $M(G, k)$ is not copositive.

Copositivity is hard: Proof of the claim

Claim: $M(G,k) = kA_G + k\mathbb{I}_n - \mathbb{J}_n$ is copositive if and only if

$$\min\{\mathbf{x}^T(A_G + \mathbb{I}_n)\mathbf{x} : \mathbf{x} \geq \mathbf{0}, \sum_{i=1}^n x_i = 1\} \geq \frac{1}{k}.$$

Proof.

M copositive $\Leftrightarrow$ $\mathbf{x}^\top M\mathbf{x} \geq 0$ for all $\mathbf{x} \geq \mathbf{0}$ such that $\sum_{i=1}^n x_i = 1$.

For $\mathbf{x} \geq \mathbf{0}$ such that $\sum_{i=1}^n x_i = 1$, we have

$$\mathbf{x}^\top M(G,k)\mathbf{x} = \mathbf{x}^\top(kA_G + k\mathbb{I}_n - \mathbb{J}_n)\mathbf{x} = k \cdot \mathbf{x}^\top(A_G + \mathbb{I}_n)\mathbf{x} - 1.$$

Hence,

$$\mathbf{x}^\top M(G,k)\mathbf{x} \geq 0 \quad \Leftrightarrow \quad \mathbf{x}^\top(A_G + \mathbb{I}_n)\mathbf{x} \geq \frac{1}{k}.$$

Applying this for all $\mathbf{x} \geq \mathbf{0}$ such that $\sum_{i=1}^n x_i = 1$:

$$M(G,k) \text{ is copositive} \quad \Leftrightarrow \quad \min\{\mathbf{x}^T(A_G + \mathbb{I}_n)\mathbf{x} : \mathbf{x} \geq \mathbf{0}, \sum_{i=1}^n x_i = 1\} \geq \frac{1}{k}.$$

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 neural networks, under suitable conditions [ACGH18].

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

Linear models with several outputs

Learning linear models (see for example the Master's Admission in Section 2.6.2 of the notes):

- n inputs $\mathbf{x}_1, \ldots, \mathbf{x}_n$, where each input $\mathbf{x}_i \in \mathbb{R}^d$
- n output values $y_1, \ldots, y_n \in \mathbb{R}$
- Hypothesis: outputs depend (approximately) linearly on the inputs, i.e.

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

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

With several output values per input:

- n output vectors $\mathbf{y}_1, \ldots, \mathbf{y}_n$, 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)

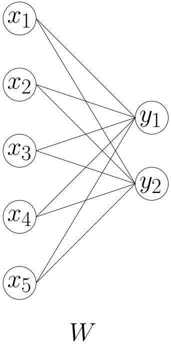
Minimizing the least squares error II

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

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

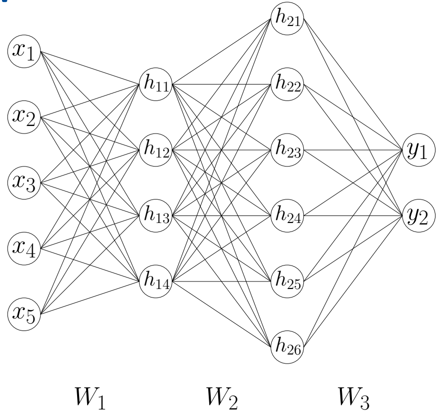
To find $W^\star$, solve $\nabla f(W) = \mathbf{0}$ (system of linear equations).

$\Leftrightarrow$ 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:

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

But "overparameterization" can help in practice for "real" (nonlinear) deep neural networks.

Training deep linear neural networks

With ℓ layers:

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

Nonconvex minimization 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, $W_i = x_i, X = 1, Y = 1, \ell = d \quad \Rightarrow f : \mathbb{R}^d \to \mathbb{R}$,

$$\frac{1}{2}\|W_\ell W_{\ell-1} \cdots W_1 X - Y\|_F^2 = f(\mathbf{x}) := \frac{1}{2}\left(\prod_{k=1}^{d} x_k - 1\right)^2.$$

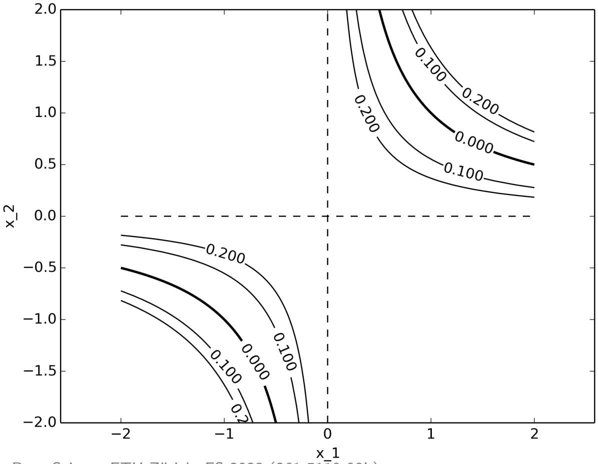
Toy example in our simple playground.

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, we abbreviate $\prod_{k=1}^{d} x_k$ by $\prod_k x_k$:

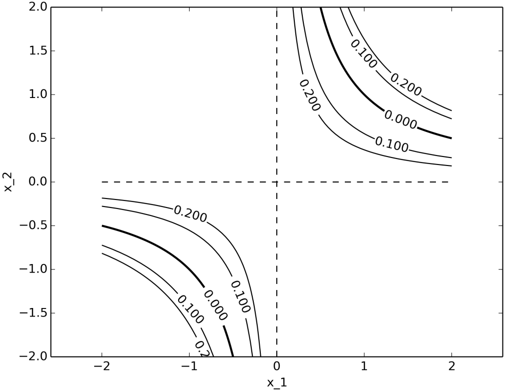
$$f(\mathbf{x}) = \frac{1}{2}\left(\prod_k x_k - 1\right)^2$$



Level set plot

The gradient of $f(\mathbf{x}) = \frac{1}{2}\left(\prod_k x_k - 1\right)^2$

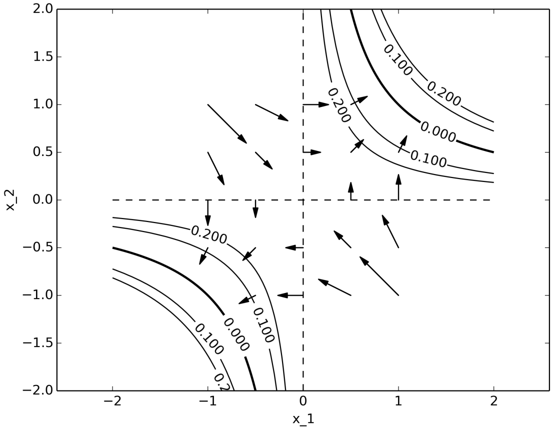
$$\nabla f(\mathbf{x}) = \left(\prod_k x_k - 1\right)\left(\prod_{k\neq 1} x_k, \ldots, \prod_{k\neq d} x_k\right).$$



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 44).

Convergence analysis on $f(\mathbf{x}) = \frac{1}{2}\left(\prod_k x_k - 1\right)^2$: 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 on $f(\mathbf{x}) = \frac{1}{2}\left(\prod_k x_k - 1\right)^2$: Overview II

For $\mathbf{x} > \mathbf{0}, \prod_k \mathbf{x}_k \geq 1$, we also get convergence (Exercise 43).

$\Rightarrow$ 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 44).

Main tool: Balanced iterates

Definition 6.4

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 6.5

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

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

□

Bounded Hessians along the trajectory (yields smoothness)

Compute $\nabla^2 f(\mathbf{x})$:

$\nabla^2 f(\mathbf{x})_{ij}$ is the j-th partial derivative of the i-th entry of $\nabla f(\mathbf{x})$.

$$(\nabla f)_i = \left(\prod_k x_k - 1\right)\prod_{k\neq i} x_k$$

$$\nabla^2 f(\mathbf{x})_{ij} = \begin{cases} \left(\prod_{k\neq i} x_k\right)^2, & j = i \\ 2\prod_{k\neq i} x_k \prod_{k\neq j} x_k - \prod_{k\neq i,j} x_k, & j \neq i \end{cases}$$

Need to bound $\prod_{k\neq i} x_k, \quad \prod_{k\neq j} x_k, \quad \prod_{k\neq i,j} x_k!$

Bounded Hessians along the trajectory II

Lemma 6.6

Suppose that $\mathbf{x} > \mathbf{0}$ is c-balanced. Then for any $I \subseteq \{1, \ldots, d\}$, we have

$$\left(\frac{1}{c}\right)^{|I|} \left(\prod_k x_k\right)^{1-|I|/d} \leq \prod_{k \notin I} x_k \leq c^{|I|} \left(\prod_k x_k\right)^{1-|I|/d}.$$

Proof.

For any i, we have $x_i^d \geq (1/c)^d \prod_k x_k$ by balancedness, hence $x_i \geq (1/c)(\prod_k x_k)^{1/d}$. It follows that

$$\prod_{k \notin I} x_k = \frac{\prod_k x_k}{\prod_{i \in I} x_i} \leq \frac{\prod_k x_k}{(1/c)^{|I|}(\prod_k x_k)^{|I|/d}} = c^{|I|} \left(\prod_k x_k\right)^{1-|I|/d}.$$

The lower bound follows in the same way from $x_i^d \leq c^d \prod_k x_k$. □

Bounded Hessians along the trajectory III

Lemma 6.7

Let $\mathbf{x} > \mathbf{0}$ be c-balanced with $\prod_k x_k \leq 1$. Then

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

where $\|\cdot\|_F$ is the Frobenius norm and $\|\cdot\|$ the spectral norm.

Proof.

$\|A\| \leq \|A\|_F$ for every matrix: Exercise 45. Now use previous lemma and $\prod_k x_k \leq 1$:

$$\left|\nabla^2 f(\mathbf{x})_{ii}\right| = |(\prod_{k \neq i} x_k)^2| \leq c^2$$

$$\left|\nabla^2 f(\mathbf{x})_{ij}\right| \leq |2 \prod_{k \neq i} x_k \prod_{k \neq j} x_k| + |\prod_{k \neq i,j} x_k| \leq 3c^2.$$

Hence, $\left\|\nabla^2 f(\mathbf{x})\right\|_F^2 \leq 9d^2c^4$. Taking square roots, the statement follows. □

Smoothness along the trajectory

Lemma 6.8

Let $\mathbf{x} > \mathbf{0}$ be c-balanced with $\prod_k x_k < 1$, $L = 3dc^2$. Let $\gamma := 1/L$. We already know from Lemma 6.5 that

$$\mathbf{x}' := \mathbf{x} - \gamma \nabla f(\mathbf{x}) \geq \mathbf{x}$$

is c-balanced. Furthermore, f **is smooth with parameter** L **over the line segment connecting** $\mathbf{x}$ **and** $\mathbf{x}'$. Lemma 6.3 (no overshooting) then also yields $\prod_k x'_k \leq 1$.

Proof.

- Imagine traveling from $\mathbf{x}$ to $\mathbf{x}'$ along the line segment. Call the current point $\mathbf{y}$.
- As long as $\prod_k y_k \leq 1$, Hessians remain bounded (previous lemma) and f is smooth over the part traveled so far (Lemma 6.1).
- Smoothness over the whole segment can only fail if we reach $\prod_k y_k = 1$ before $\mathbf{x}'$.
- We have $\nabla f(\mathbf{x}) \neq \mathbf{0}$ (due to $\mathbf{x} > \mathbf{0}, \prod_k x_k < 1$), so $\mathbf{x}$ is not a critical point.
- $\mathbf{y} \neq \mathbf{x}'$ results from $\mathbf{x}$ by a gradient descent step with stepsize $< 1/L$ and is also not a critical point by Lemma 6.3 (no overshooting). Contradiction to $\prod_k y_k = 1$!

Convergence

Theorem 6.9

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

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

gradient descent satisfies

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

- Error converges to 0 exponentially fast.
- Exercise 46: 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:

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

For every c-balanced $\mathbf{x}$ with $\delta \leq \prod_k x_k \leq 1$, $\left\|\nabla f(\mathbf{x})\right\|^2$ equals (using Lemma 6.6)

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

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

□

Discussion

Fast convergence as for strongly convex functions!

But there is a catch...

Consider starting solution $\mathbf{x}_0 = (1/2, \ldots, 1/2)$ (this is 1-balanced, very nice).

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

With $\delta = 2^{-d}$ and $c = 1$, the function value is guaranteed to decrease 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.

K. G. Murty and S. K. Kabadi.
Some NP-complete problems in quadratic and nonlinear programming.
Math. Programming, 39:117–129, 1987.

T. .S. Motzkin and E. G. Straus.
Maxima for graphs and a new proof of a theorem of Turán.
Canadian Journal of Mathematics, 17:533–540, 1965.