

# Optimization for Machine Learning

## CS-439

Lecture 6: Non-convex optimization

**Nicolas Flammarion**

EPFL – [github.com/epfml/OptML\\_course](https://github.com/epfml/OptML_course)

March 28, 2025

# 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 [ACGH19].

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

# Linear models with several outputs

Recall: Learning linear models

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

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

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

Now more than one output value:

- ▶  $n$  outputs  $\mathbf{y}_1, \dots, \mathbf{y}_n$ , 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  $\operatorname{vec}(A)$  (“flattening” of  $A$ )

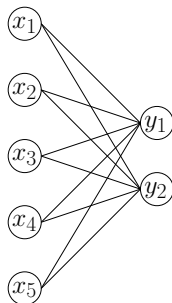
# Minimizing the least squares error II

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

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

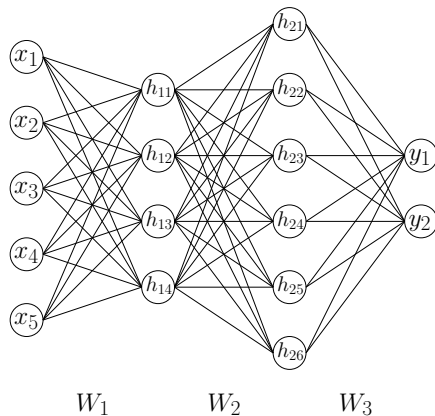
To find  $W^*$ , solve  $\nabla f(W) = \mathbf{0}$  (system of linear equations).

$\Leftrightarrow$  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:

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

# Training deep linear neural networks

With  $\ell$  layers:

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

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  $1 \times 1$ ,  $W_i = x_i$ ,  $X = 1$ ,  $Y = 1$ ,  $\ell = d \Rightarrow f : \mathbb{R}^d \rightarrow \mathbb{R}$ ,

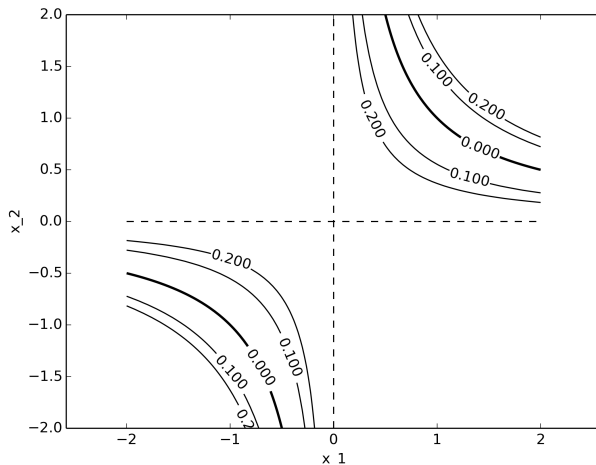
$$f(\mathbf{x}) := \frac{1}{2} \left( \prod_{k=1}^d x_k - 1 \right)^2.$$

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 [ACGH19].

## A simple nonconvex function

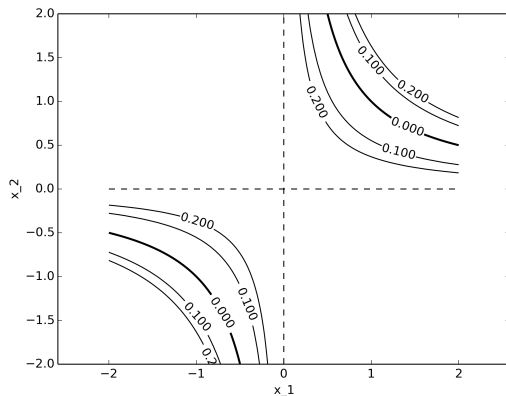
As  $d$  is fixed, abbreviate  $\prod_{k=1}^d x_k$  by  $\prod_k x_k$ :  $f(\mathbf{x}) = \frac{1}{2} \left( \prod_k x_k - 1 \right)^2$





# The gradient

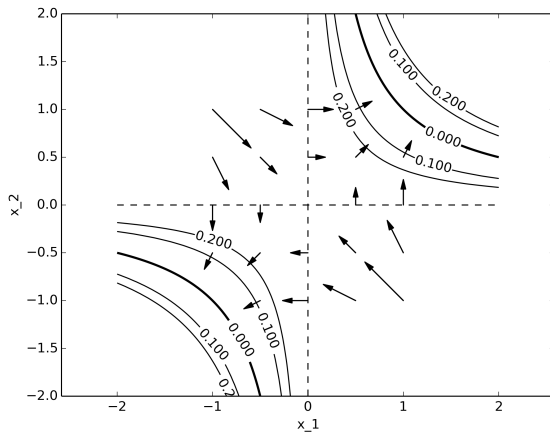
$$\nabla f(\mathbf{x}) = \left( \prod_k x_k - 1 \right) \left( \prod_{k \neq 1} x_k, \dots, \prod_{k \neq d} x_k \right).$$



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 42).

## 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 minimum? 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 41).  $\Rightarrow$  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 42).

# Main tool: Balanced iterates

## Definition

Let  $\mathbf{x} > \mathbf{0}$  (componentwise), and let  $c \geq 1$  be a real number.  $\mathbf{x}$  is called *c-balanced* if  $x_i \leq cx_j$  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

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.

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

Gradient descent step:

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

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

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



## Bounded Hessians along the trajectory

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  $\prod_{k \neq i} x_k$ ,  $\prod_{k \neq j} x_k$ ,  $\prod_{k \neq i, j} x_k$ !

## Bounded Hessians along the trajectory II

### Lemma

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

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

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

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

## Proof.

$\|A\| \leq \|A\|_F$ : Exercise 43. Now use previous lemma and  $\prod_k x_k \leq 1$ :

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

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

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



## Smoothness along the trajectory

### Lemma

Let  $\mathbf{x} > \mathbf{0}$  be  $c$ -balanced with  $\prod_k x_k < 1$ ,  $L = 3dc^2$ . Let  $\gamma := 1/L$ . Then for all  $0 \leq \nu \leq \gamma$ ,

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

is  $c$ -balanced with  $\prod_k x'_k \leq 1$ , and  $f$  is smooth with parameter  $L$  over the line segment connecting  $\mathbf{x}$  and  $\mathbf{x} - \gamma \nabla f(\mathbf{x})$ .

### Proof.

- ▶  $\mathbf{x}' \geq \mathbf{x} > \mathbf{0}$  is  $c$ -balanced by Lemma 6.5.
- ▶  $\nabla f(\mathbf{x}) \neq \mathbf{0}$  (due to  $\mathbf{x} > \mathbf{0}$ ,  $\prod_k x_k < 1$ , we can't be at a critical point).
- ▶ No overshooting: we can't reach  $\prod_k x'_k = 1$  (global minimum) for  $\nu < \gamma$ , as  $f$  is smooth with parameter  $L$  between  $\mathbf{x}$  and  $\mathbf{x}'$  (using previous bound on Hessians in Lemma 6.1).
- ▶ By continuity,  $\prod_k x'_k \leq 1$  for all  $\nu \leq \gamma$ .
- ▶  $f$  is smooth with parameter  $L$  between  $\mathbf{x}$  and  $\mathbf{x}'$  for  $\nu = \gamma$ .

# Convergence

## Theorem

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 44: 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  $\mathbf{x}_0 = (1/2, \dots, 1/2)$ .

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

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.  
*In ICLR - International Conference on Learning Representations, 2019.*