INSTITUT NATIONAL DES SCIENCES APPLIQUÉES TOULOUSE

Gradient Descent: A Quick Introduction

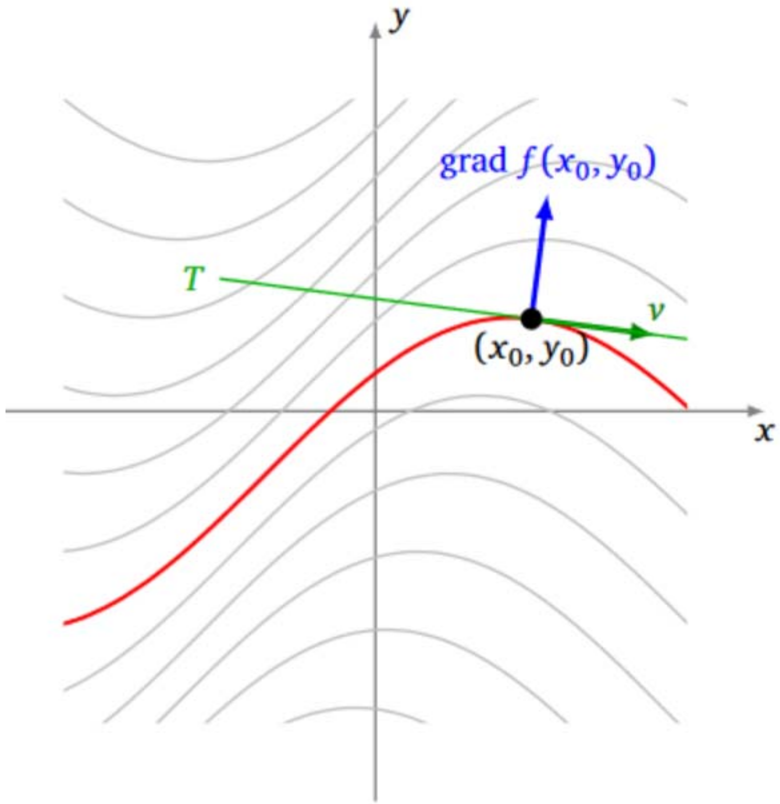
Université de Toulouse

Prerequisite: gradient

Let $f: \mathbb{R}^n \to \mathbb{R}$ a differentiable function. **The gradient** in $x^* = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$, denoted or $f_x(x^*)$ or grad $f(x^*)$ or $\Delta f(x^*)$, is the vector

$$\boldsymbol{f_x(x^*) = \begin{pmatrix} \frac{\partial f}{\partial x_1}(x^*) \\ \vdots \\ \frac{\partial f}{\partial x_n}(x^*) \end{pmatrix}}$$

Application: $f(x_1, x_2) = (x_1 + 2)^2 - 1 + x_2$. Compute the gradient in $x^* = (2,5)$



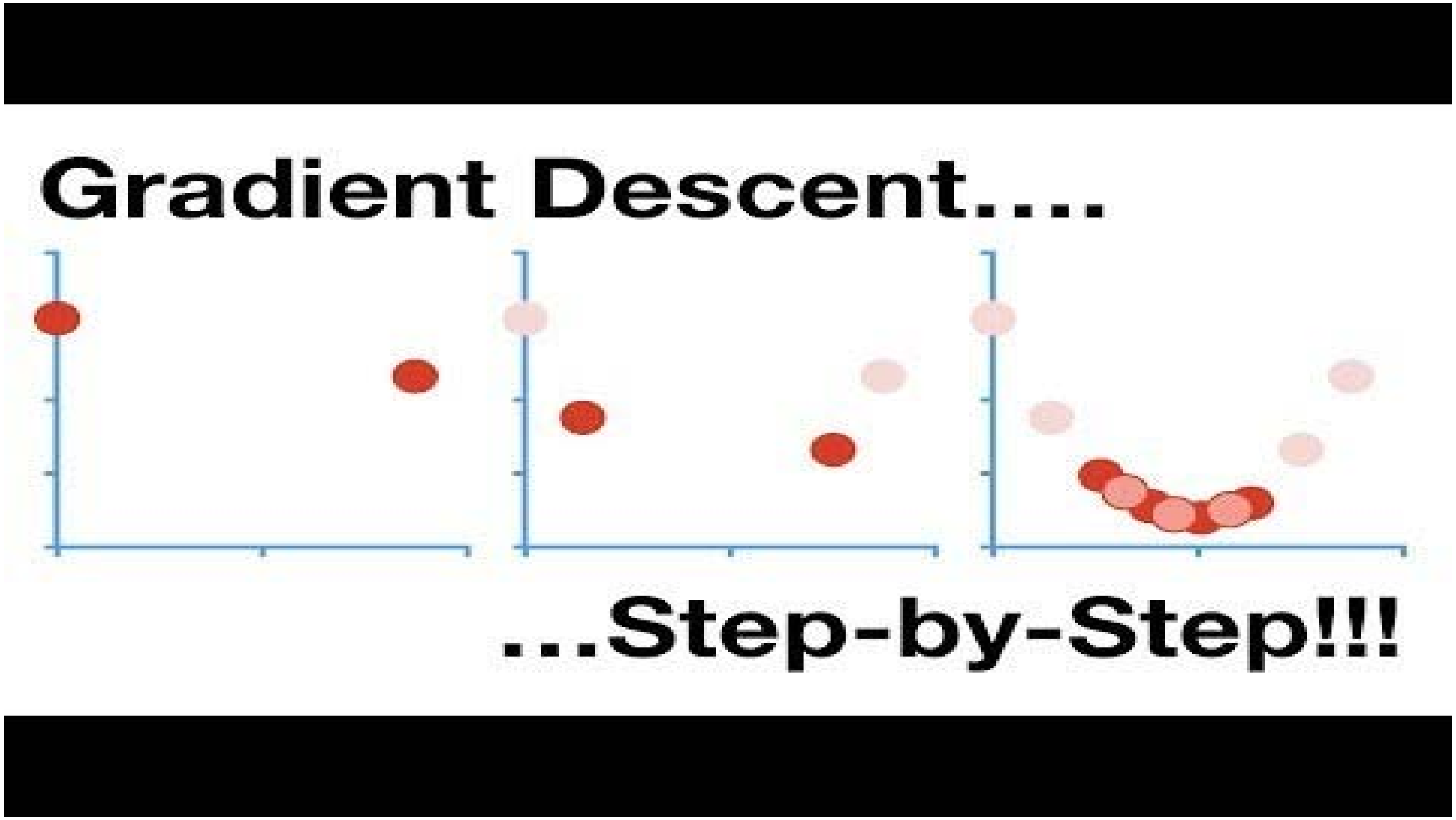
At each point of the plane a gradient vector starts. This gradient vector is **orthogonal** to the level line passing through this point.

Prerequisite: Hessian

F_{XX} the **Hessian matrix,** square, symmetric matrix, also denoted H_f is defined as:

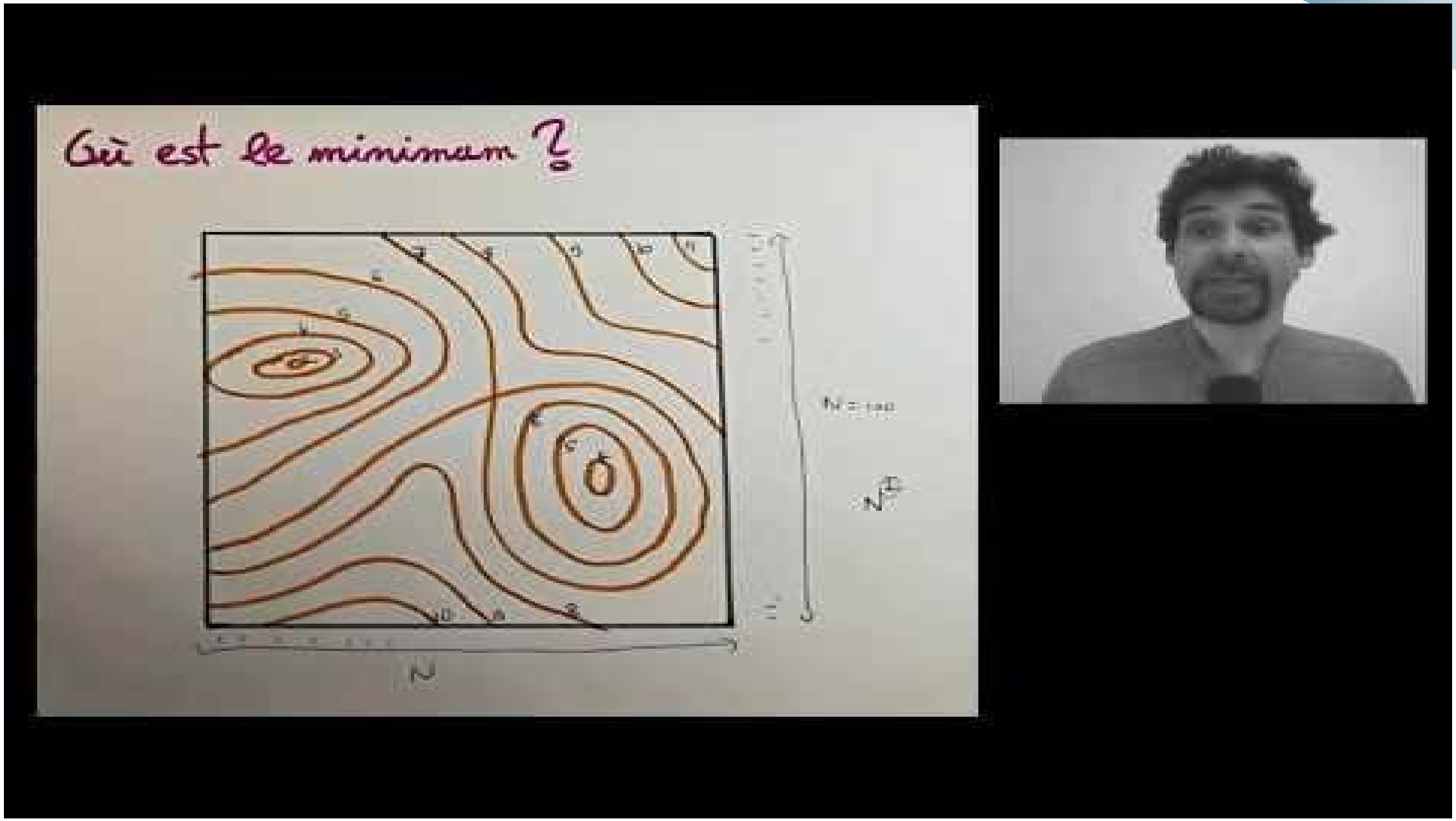
$$F_{XX} = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}$$

In English



https://www.youtube.com/watch?v=sDv4f4s2SB8

In French

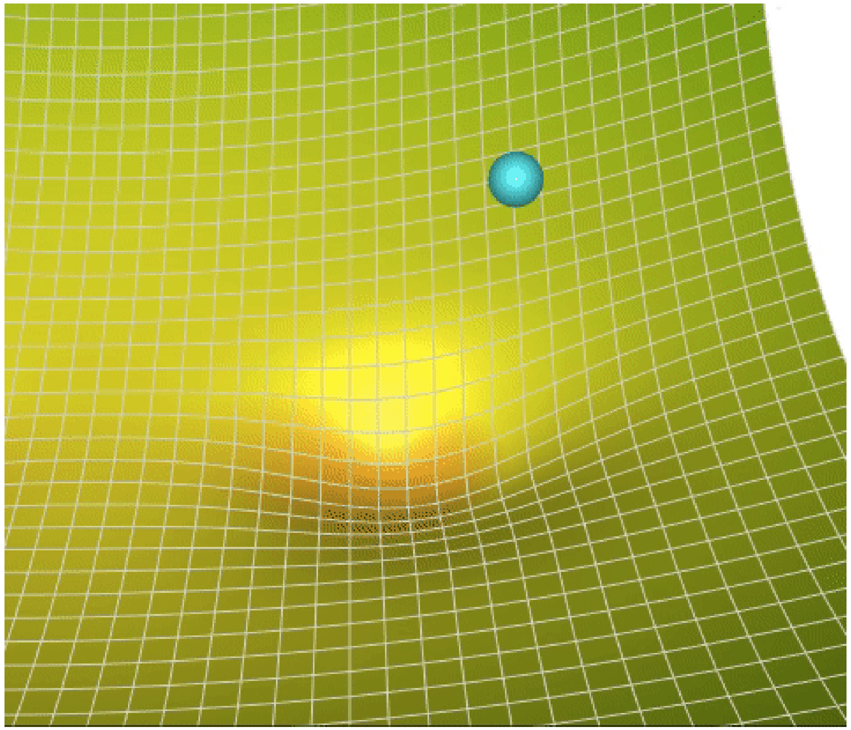


https://www.youtube.com/watch?v=GE4UwT-JV4A

The goal of **Gradient Descent** is to **minimize an objective function f using iterations**

Goal: Find $\hat{x} \in \mathbb{R}^n$, the absolute or relative minimum of $f(x)$

$$\hat{x} = \min f(x)$$



https://towardsdatascience.com/a-visual-explanation-of-gradient-descent-methods-momentum-adagrad-rmsprop-adam-f898b102325c

Mathematical conditions : 1st order condition

Assume that $f: \mathbb{R}^n \to \mathbb{R}$ is **continuous and continuously differentiable** to the 1st order

1st order Taylor approximation: for δx sufficiently small, we can develop $f(x)$ in Taylor series to the 1st order: $f(x+\delta x) \approx f(x) + f_x^T \delta x + O(\|\delta x\|^2)$ (1)

If $\hat{x}$ is a relative minimum of f, then it exists a neighborhood of $\hat{x}$ where:

$$f(\hat{x}+\delta x) \geq f(\hat{x}) \quad \forall \delta x \text{ small}, \delta x \in \mathbb{R}^n \qquad (2)$$

(1)+(2) → $f_x^T \delta x \geq 0 \; \forall \delta x \in \mathbb{R}^n$ (the term $O(\|\delta x\|^2$ is neglected)

This relationship must hold for any small δx , and in particular when we change δx to -δx,

so $\boxed{f_x(\hat{x}) = 0 \text{ and } f(\hat{x}+\delta x) \approx f(\hat{x}) = \text{cst}}$

Necessary condition of local extremum: Let f a function *continuous and continuously differentiable to the 1st order*, **if $\hat{x}$ is a relative minimum of f** then

$$\boldsymbol{f_x(\hat{x}) = 0}$$ **The gradient is zero**

Mathematical conditions: 2nd order conditions

Assume that $f: \mathbb{R}^n \rightarrow \mathbb{R}$ is C^2, then (2nd order development)

$$f(x+\delta x) = f(x) + f_x^T \delta x + {}^1\!/_2\, \delta x^T F_{XX} \delta x + (o(\|\delta x\|^3))$$

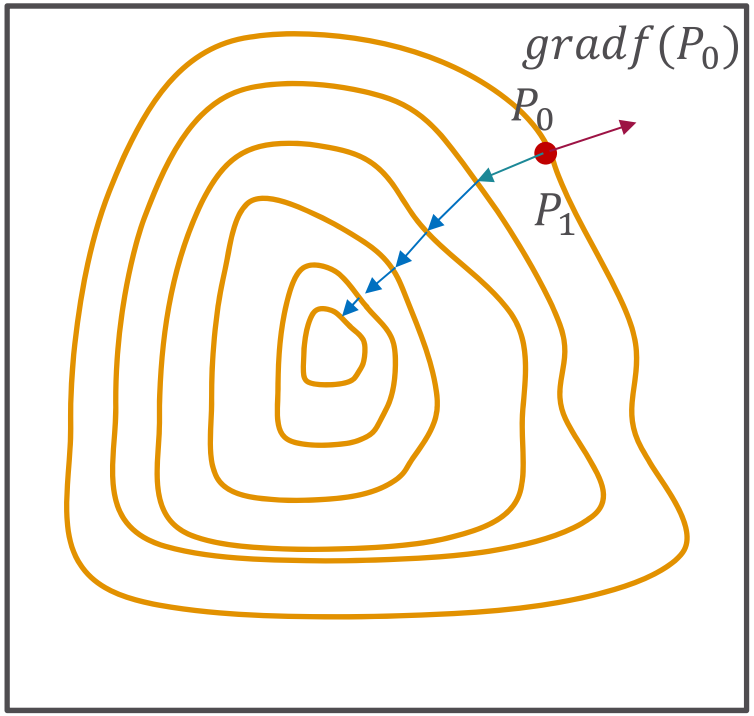
If $\hat{x}$ is a relative minimum of f then **$F_{XX}(\widehat{x})$ is a positive definite matrix**

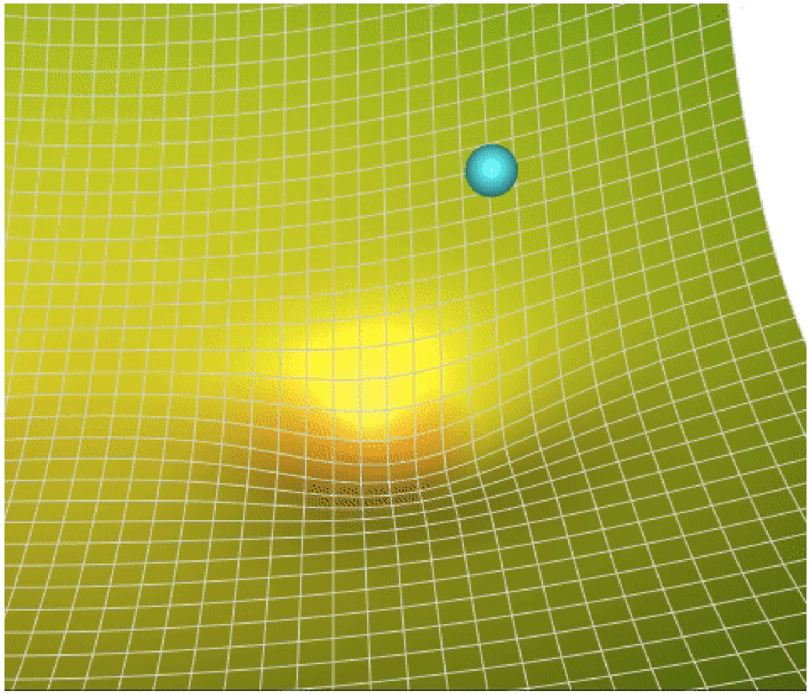
proof: $f(\hat{x}+\delta x) \geq f(\hat{x}) \quad \forall \delta x\ small, \delta x \in \mathbb{R}^n$

If $f: \mathbb{R}^n \rightarrow \mathbb{R}$ is C^2, **if $f_x(x^*) = 0$ and $F_{XX}(x^*)$ is a positive definite matrix, then x^* is a relative minimum for f.**

Descent methods

Basic idea: we do as on a topographic map, and use the level lines to go to the bottom of a valley





2 notions

$\omega_k > 0$: the step at iteration k

s_k: the descent direction

Descent direction. If $f: \mathbb{R}^n \to \mathbb{R}$, a **vector** $\boldsymbol{s} \in \mathbb{R}^{\boldsymbol{n}}$ is said to be a descent direction in x if it exists $\omega^* > 0 \; s.t \; f(x + \omega s) < f(x) \; \omega \in \;]0, \omega^*]$

→ The descent direction moves us closer towards a local minimum x^* of the objective function f

Descent algorithms: general algorithm

Start with an arbitrary initial point x_0. At each step $k \geq 0$:

- Choose a **descent direction** s_k in x_k,
- Choose a **step** $\omega_k > 0$ such that $x_k + \omega_k s_k \in \mathbb{R}^{\mathrm{n}}$ and that $\boldsymbol{f(x_k + \omega_k s_k) < f(x_k)}$
- $\boldsymbol{x_{k+1} = x_k + \omega_k s_k}$

→ The different gradient methods correspond to different choices for descent directions and for steps

We know that $\boldsymbol{f(x_k + \omega_k s_k) < f(x_k)}$

And as f is differentiable, we have:

$$\lim_{\omega \to 0} \frac{f(x_k + \omega s_k) - f(x_k)}{\omega} = \left. \frac{df(x_k + \omega s_k)}{d\omega} \right|_{\omega=0} = s_k^T f_x(x_k)$$

So

$\boldsymbol{s_k}$ **is a descent direction** in x_k if $\boldsymbol{s_k^T f_x(x_k) < 0}$

Choosing a descent direction

Property - Let M be a positive definite matrix.

Any vector s_k defined as $\boldsymbol{s_k = -M f_x(x_k)}$ **is a descent direction**

Proof:

s_k **is a descent direction** in x_k if $s_k^T f_x(x_k) < 0$

$$s_k{}^T f_{x(x_k)} = (-M f_x(x_k))^T . f_{x(x_k)} = -f_x^T(x_k) M^T . f_x(x_k) = -f_x^T(x_k) M f_x(x_k) < 0$$

→ *Different methods depend on the choice of M*

The simplest choice: the Gradient Method

- The simplest choice is for $\boldsymbol{M} = \boldsymbol{Id}$
- The descent direction is then $\boldsymbol{s_k} = -\boldsymbol{f_x}(\boldsymbol{x_k})$
- This direction is named the **steepest descent**. It corresponds to (-) the gradient of the f function → so-called **gradient descent method**
- The computation of the descent directions only needs 1st order partial derivative computation → this is a 1st order method

Choosing the step ω_k

Non optimal step

$$\boldsymbol{f}(\boldsymbol{x_k} + \boldsymbol{\omega_k s_k}) < \boldsymbol{f}(\boldsymbol{x_k}), \text{ with } \omega_k > 0$$

Optimal step

Given a point x_k and a descent direction s_k, **a step $\boldsymbol{\omega_k}$** is said to be **optimal** if it is solution of the problem:

$$\boldsymbol{f}(\boldsymbol{x_k} + \boldsymbol{\omega_k s_k}) = \min_{\boldsymbol{\omega}} \boldsymbol{f}(\boldsymbol{x_k} + \boldsymbol{\omega s_k})$$

→ Monovariable optimisation

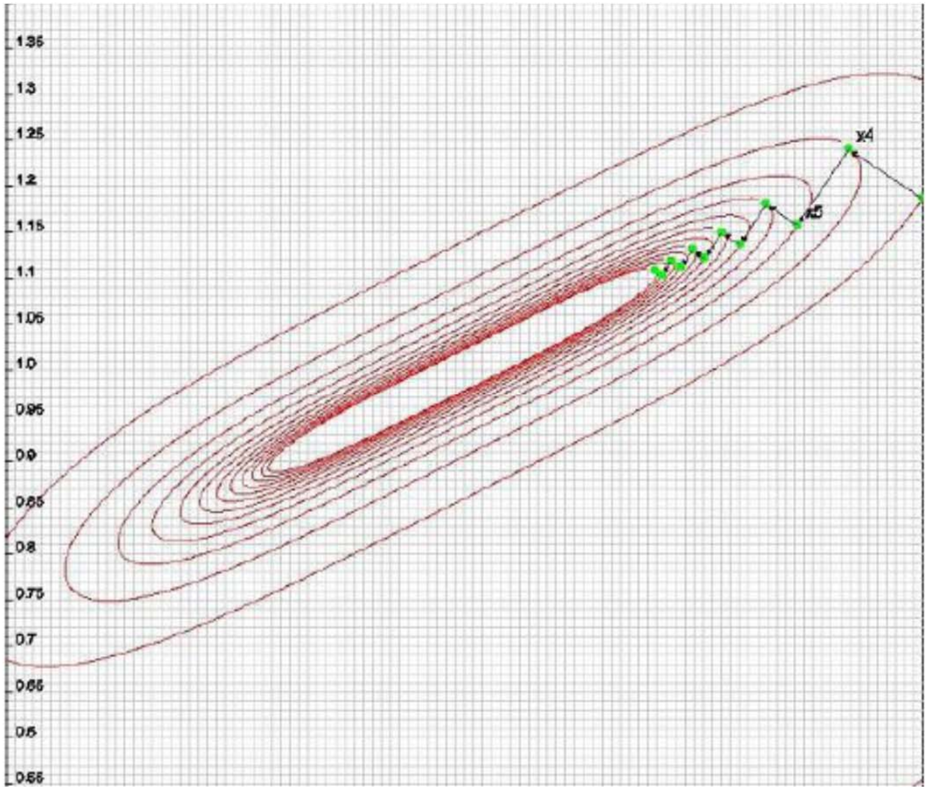
Effects of the step

https://developers.google.com/machine-learning/crash-course/fitter/graph

Example

$$f(x_1, x_2) = (x_1 - 2)^4 + (x_1 - 2x_2)^2$$

$$f_x(x) = \begin{pmatrix} 4(x_1 - 2)^3 + 2(x_1 - 2x_2) \\ -4(x_1 - 2x_2) \end{pmatrix}$$



- https://www.ceremade.dauphine.fr/~amic/enseignement/MNO2015/MNO2015.pdf
- https://mrmint.fr/gradient-descent-algorithm