Advanced Optimization
Lecture 1: Introduction

October 13, 2021

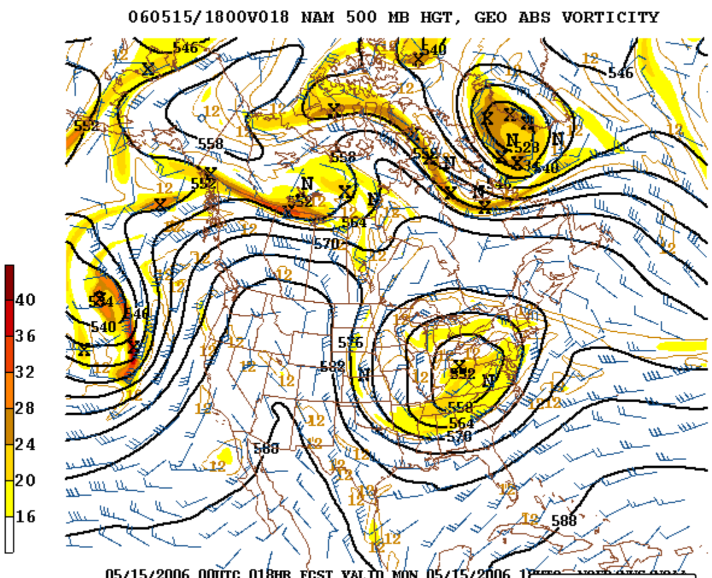
CentraleSupélec / ESSEC Business School

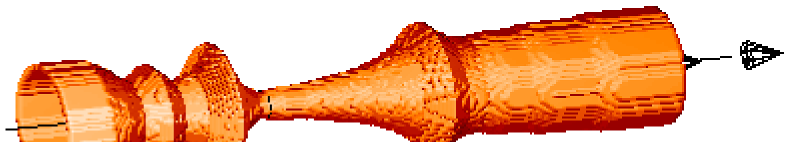
dimo.brockhoff@inria.fr

Dimo Brockhoff

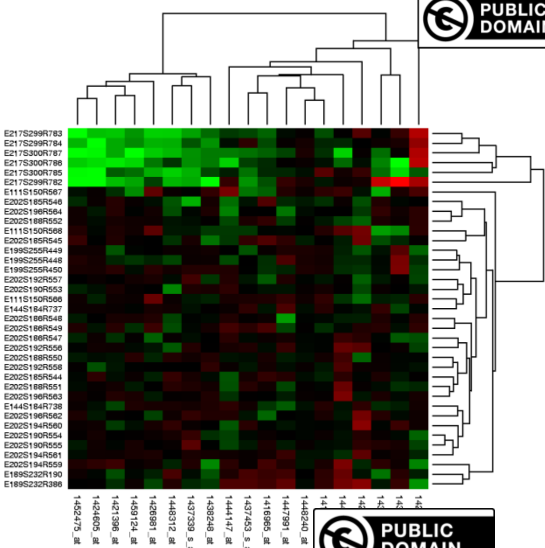
Inria Saclay – Ile-de-France

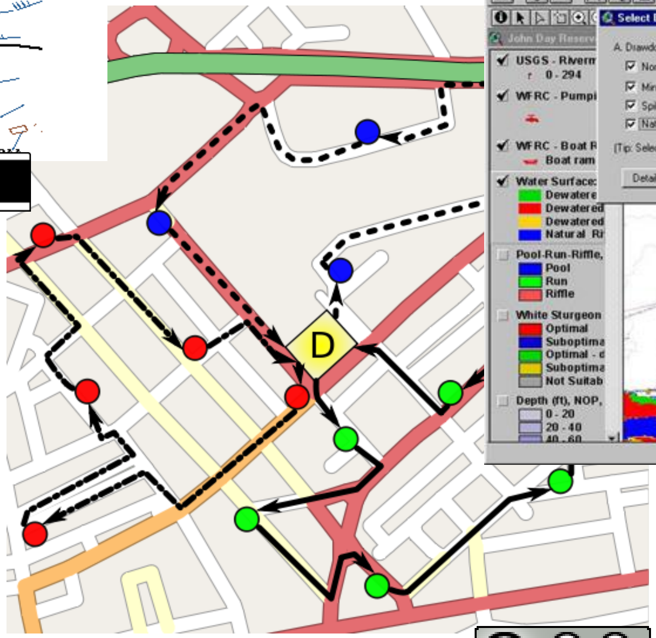
What is Optimization?



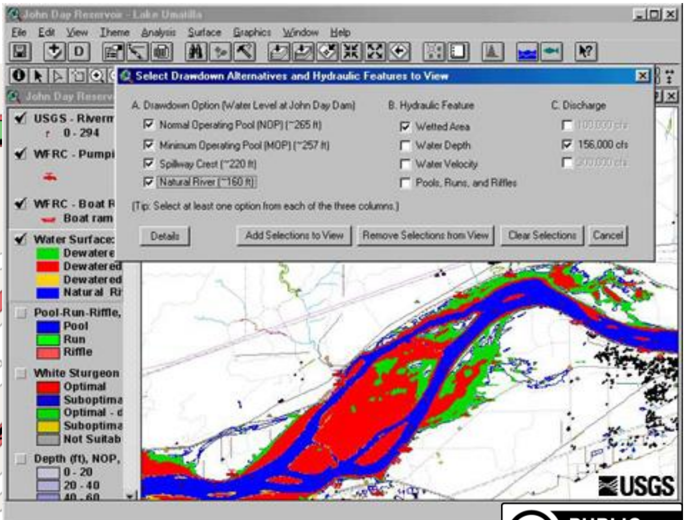


© H.-P. Schwefel





Maly LOLek



Examples of Optimization in Machine Learning

Support Vector Machines (SVMs)

- need to find/optimize weights

Hyperparameter Tuning

= optimization of internal algorithm parameters before training

Learning of (Deep) Neural Networks

- stochastic gradient descent is basis for the most advanced algorithms like Adam*

Reinforcement Learning

- …is itself like a (dynamic) optimization task

What is Optimization?

$$\min_{x \in \Omega} f(x)$$
$$s.t.\, g(x) \leq 0$$
$$h(x) = 0$$

Typically, we aim at

- finding solutions x which minimize f(x) in the shortest time possible

  (maximization is reformulated as minimization)

- or finding solutions x with as small f(x) in the shortest time possible
  (if finding the exact optimum is not possible)

Be Aware

“Optimization” is a very wide topic, maybe as wide as “vegetables“ ☺

even in “advanced optimization”, we can only touch the surface

- I am here to guide you
- and to give some hints of what might be useful later in your job
- we’ll see a range of algorithms on a range of problems

What we plan to do in the AO lecture

Learning Goals:

❶ Know basics of optimization theory

gradients, Hessian, optimality conditions, …

❷ Know basic design principles behind good optimizers

gradient descent, stochastic gradient descent, …

❸ Be able to use and understand existing algorithms

benchmarking, contributions to open source projects

What we plan to do in the AO lecture

How are we going to do that?

- look at a lot of examples of algorithms
- mixture of lectures and small exercises
- practice and theory
- additionally 2 graded mini-exams throughout the course
- 1 contribution to open source project

Please ask questions
if things are unclear throughout the course!

Details on Mini-Exams

- expected to be done via Evalmee
- before the first mini-exam, we will do a technical test
- multiple-choice questions, similar to final exam
- embedded into the lecture
  - 3rd lecture on October 27
  - 6th lecture on November 17
- both mini-exams together will count for 1/6 of overall grade

Contributions to Open Source Project I

- Practical project
- Group project of up to 5 students (large groups encouraged ☺)
- Goal: Contribution to an existing open source project, e.g.
  - Scikit-optimize https://github.com/scikit-optimize/scikit-optimize
  - CMA-ES solver https://github.com/CMA-ES/pycma
  - COmparing Continuous Optimizers (benchmarking platform) https://github.com/numbbo/coco
  - Nevergrad https://facebookresearch.github.io/nevergrad/contributing.html
  - optimization-related issues in https://github.com/scikit-learn/scikit-learn
  - Projects of AutoML group https://github.com/automl
  - any other open source project related to optimization of your own choice
- Will count 1/6 of overall grade

Contributions to Open Source Project II

What I expect from you:

❶ Contributions in any way count, e.g.

- solving an existing issue
- addition of a new component (implement feature, algorithm, …)
- comparison of a new algorithm in COCO or Nevergrad
- …

❷ Write a report about your contributions (5-10 pages)

- Including details on the software and your contribution
- Detail the contributions of each team member
- Deadline: November 30, 2021 at 23h59 Paris time
- Submission of PDF by email to me

Considerations on Open Source Project

How to Pick a Project?

- Small vs. large: both have advantages, but smaller projects seem to be more suited here due to the limited time
- Prefer active repositories over inactive

Start early enough!

- Decide on groups & project early, if possible this week
- Both writing code and writing the report takes time

Versioning system

- it cannot hurt to also use a versioning system like github for your report
- https://gitlab-student.centralesupelec.fr

The Exam

- Wednesday, 8th December 2021 in the afternoon (3 hours)
- (most likely) multiple-choice with 20-30 questions
- (most likely) on-site + online via Evalmee
- open book: use as much material as you want
- accounts for 2/3 of overall grade

all information (incl. the slides) will be available at EDUNAO

Course Overview

		Topic
Wed, 13.10.2021	PM	Introduction, examples of problems, problem types
Wed, 20.10.2021	PM	Continuous (unconstrained) optimization: convexity, gradients, Hessian, … [technical test Evalmee]
Wed, 27.10.2021	PM	Continous optimization II: gradient descent, Newton direction, quasi-Newton (BFGS) [1st mini-exam] Linear programming: duality, maxflow/mincut, simplex algo
Wed, 03.11.2021	PM	Constrained optimization: Lagrangian, optimality conditions
Wed, 10.11.2021	PM	Gradient-based and derivative-free stochastic algorithms: SGD and CMA-ES
Wed, 17.11.2021	PM	Other blackbox optimizers: Nelder-Mead, Bayesian optimization [2nd mini-exam]
Wed, 24.11.2021	PM	Benchmarking solvers: runtime distributions, performance profiles
Tue, 30.11.2021	23:59	Deadline open source project (PDF sent by email)
Wed, 01.12.2021	PM	Discrete optimization: branch and bound, branch and cut, k-means clustering
Wed, 8.12.2021	PM	Exam

Overview of Today's Lecture

- More examples of optimization problems
  - introduce some basic concepts of optimization problems such as domain, constraint, ...
- Beginning of continuous optimization part
  - typical difficulties in continuous optimization
  - differentiability
  - … [we'll see how far we get]

General Context Optimization

Given:

set of possible solutions *Search space*

quality criterion *Objective function*

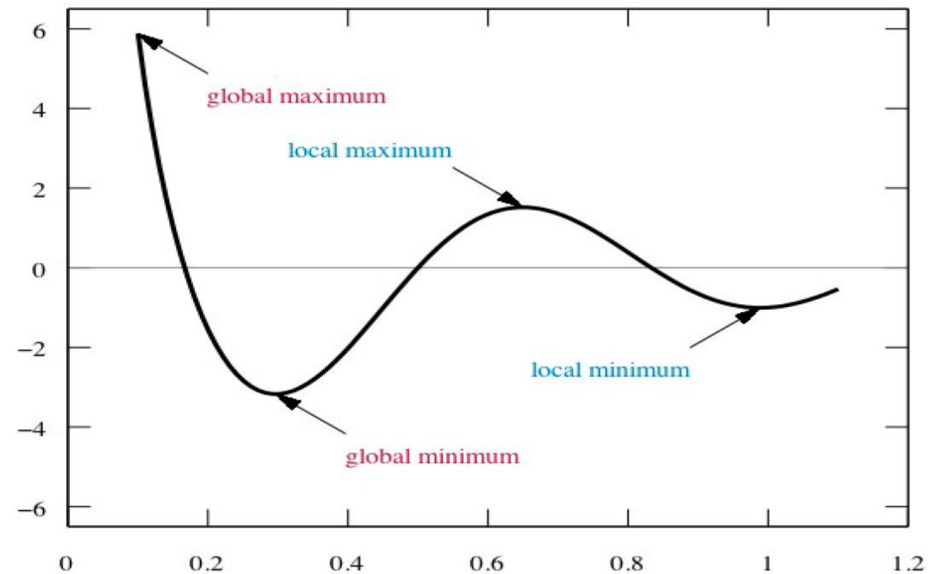
Objective:

Find the best possible solution for the given criterion

Formally:

Maximize or minimize

$$\mathcal{F}: \Omega \longmapsto \mathbb{R},$$
$$x \longmapsto \mathcal{F}(x)$$



Constraints

Maximize or minimize

$$\mathcal{F}: \Omega \longmapsto \mathbb{R},$$
$$x \longmapsto \mathcal{F}(x)$$

unconstrained

Ω

Maximize or minimize

$$\mathcal{F}: \Omega \longmapsto \mathbb{R},$$
$$x \longmapsto \mathcal{F}(x)$$

where $g_i(x) \leq 0$

$$h_i(x) = 0$$

example of a

constrained Ω

Constraints explicitly or implicitly define the feasible solution set

[e.g. ||x|| - 7 ≤ 0 vs. every solution should have at least 5 zero entries]

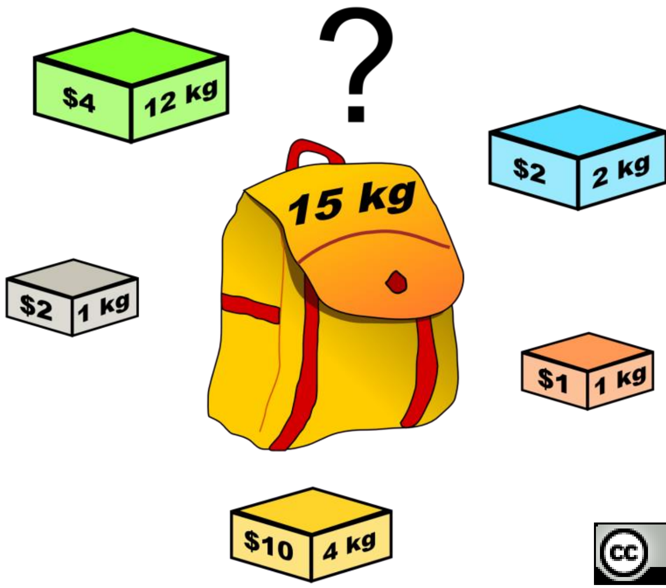
Hard constraints *must* be satisfied while soft constraints are preferred to hold but are not required to be satisfied

[e.g. constraints related to manufacturing precisions vs. cost constraints]

Example 1: Combinatorial Optimization

Knapsack Problem

- Given a set of objects with a given weight and value (profit)
- Find a subset of objects whose overall mass is below a certain limit and maximizing the total value of the objects



[Problem of ressource allocation with financial constraints]

$$\max \sum_{j=1}^{n} p_j x_j \quad \text{with } x_j \in \{0,1\}$$

$$\text{s.t.} \sum_{j=1}^{n} w_j x_j \leq W$$

$$\Omega = \{0,1\}^n$$

Traveling Salesperson Problem (TSP)

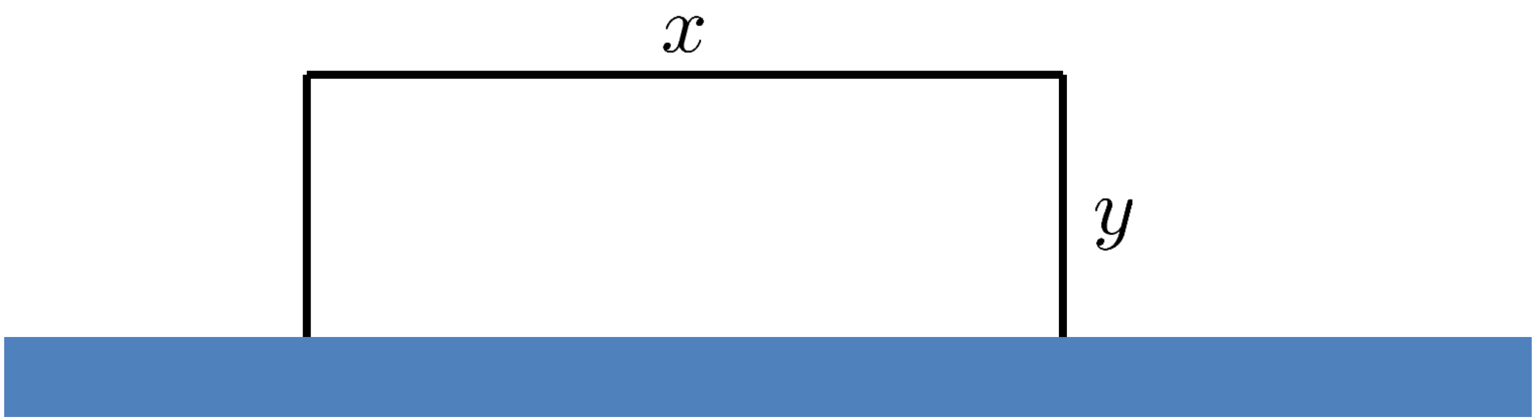
- Given a set of cities and their distances
- Find the shortest path going through all cities

PUBLIC DOMAIN

$$\Omega = S_n \text{ (set of all permutations)}$$

Example 3: Continuous Optimization

A farmer has 500m of fence to fence off a rectangular field that is adjacent to a river. What is the maximal area he can fence off?



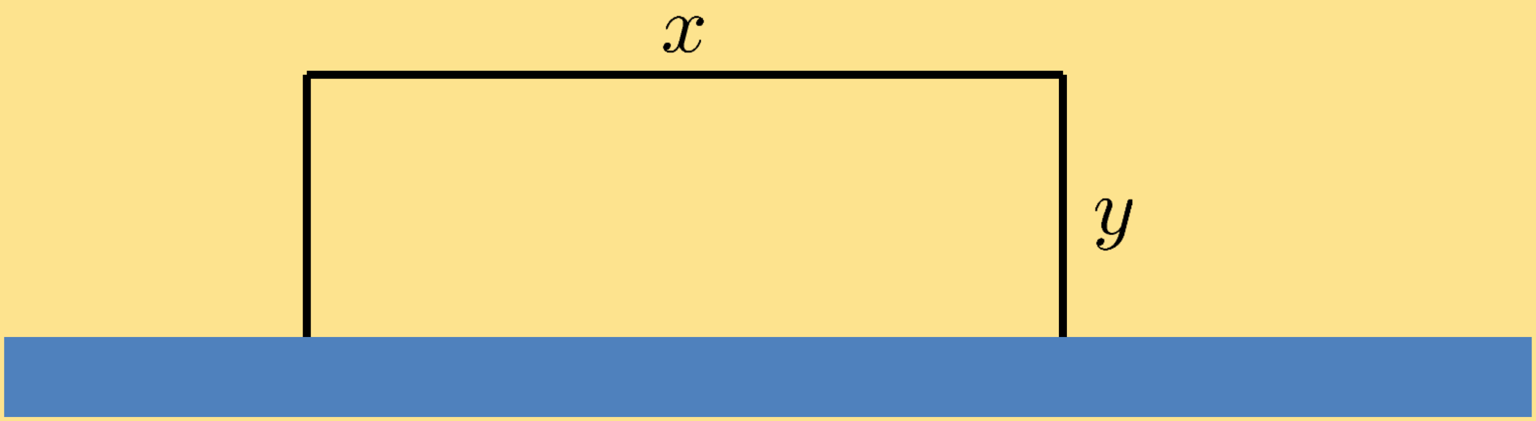
Exercise:

a) what is the search space?

b) what is the objective function?

Example 3: Continuous Optimization

A farmer has 500m of fence to fence off a rectangular field that is adjacent to a river. What is the maximal area he can fence off?



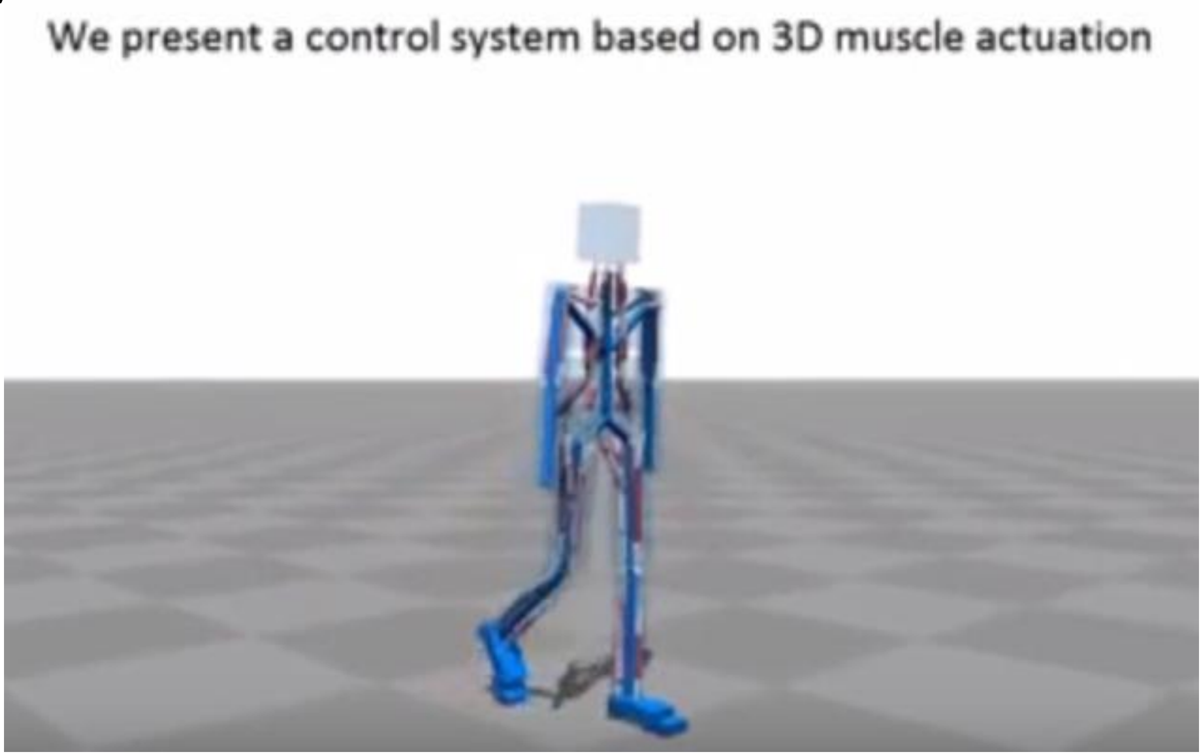
solution can be found analytically:

exercise for the weekend ;-)

$$\Omega = \mathbb{R}_+^2 :$$
$$\max xy$$
where $x + 2y \leq 500$

Example 4: Continuous Optimization Problem

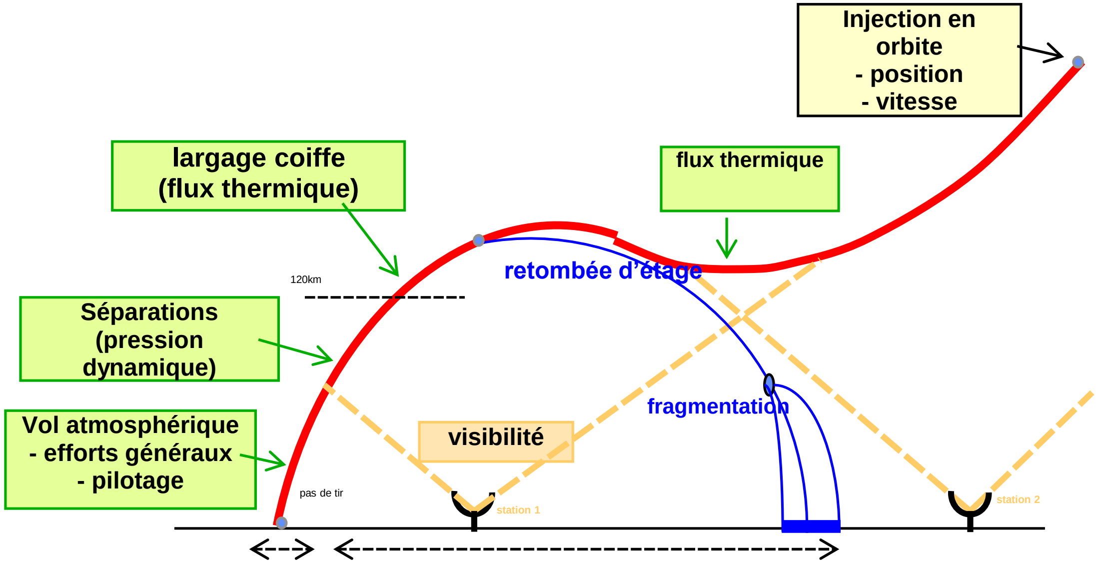
Computer simulation teaches itself to walk upright (virtual robots (of different shapes) learning to walk, through stochastic optimization (CMA-ES)), by Utrecht University:



https://www.youtube.com/watch?v=pgaEE27nsQw

T. Geitjtenbeek, M. Van de Panne, F. Van der Stappen: "Flexible Muscle-Based Locomotion for Bipedal Creatures", SIGGRAPH Asia, 2013.

Design of a Launcher



- Scenario: multi-stage launcher brings a satellite into orbit
- Minimize the overall cost of a launch
- Parameters: propellant mass of each stage / diameter of each stage / flux of each engine / parameters of the command law

23 continuous parameters to optimize

+ constraints

$$\Omega = \mathbb{R}^{23}$$

Example 6: Data Fitting – Data Calibration

Objective

- Given a sequence of data points $(\boldsymbol{x}_i, y_i) \in \mathbb{R}^p \times \mathbb{R}, i = 1, \dots, N$, find a model "$y = f(\boldsymbol{x})$" that *"explains"* the data

experimental measurements in biology, chemistry, ...

- In general, choice of a parametric model or family of functions $(f_\theta)_{\theta \in \mathbb{R}^n}$

use of expertise for choosing model

or only a simple model is affordable (e.g. linear, quadratic)

- Try to find the parameter $\theta \in \mathbb{R}^n$ fitting best to the data

Fitting best to the data

Minimize the quadratic error:

$$\min_{\theta \in \mathbb{R}^n} \sum_{i=1}^{N} |f_\theta(\boldsymbol{x}_i) - y_i|^2$$

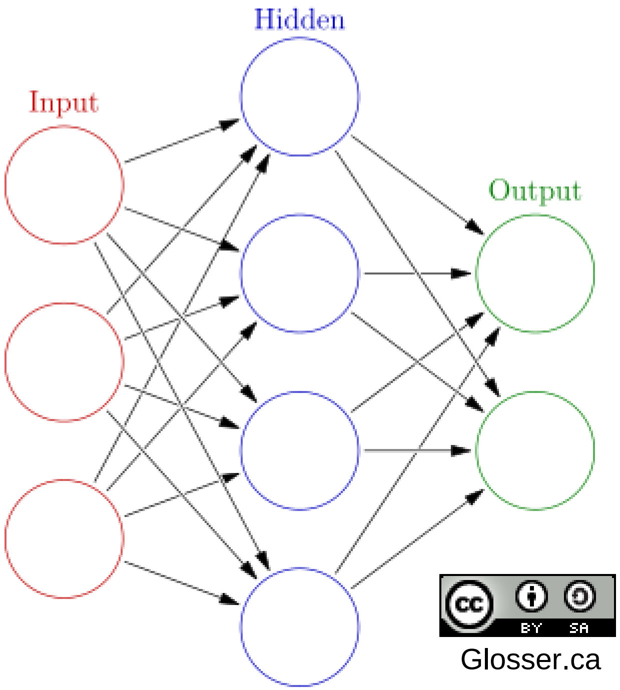
Example 7: Deep Learning

Actually the same idea:

match model best to given data

Model here:

artificial neural nets
with many hidden layers
(aka deep neural networks)



Parameters to tune:

- weights of the connections (continuous parameter)
- topology of the network (discrete)
- firing function (less common)

Specificity:

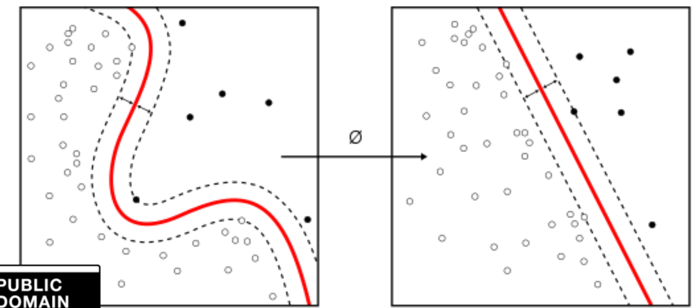
- large amount of training data, hence often batch learning

Example 8: Classification with SVMs

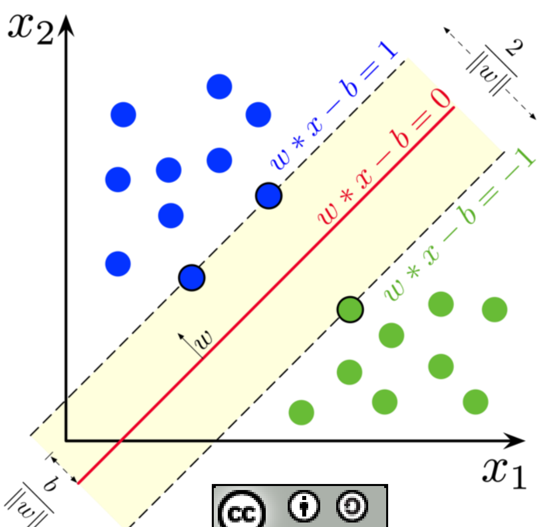
$y_i \in \{-1, +1\}$

Scenario:

- supervised learning of 2-class samples
- Support Vector Machines (SVMs):
  - decide to which class a new sample belongs
  - learns from the training data the "best linear model" (= a hyperplane separating the two classes); non-linear transformations possible via the kernel trick



PUBLIC DOMAIN



CC BY SA

Larhmam

- hard margin (when data linearly separable):

$$\min \|\boldsymbol{w}\| \ \text{s.t.}\ y_i\,(\boldsymbol{w} \cdot \boldsymbol{x}_i) - b \geq 1 \ \forall 1 \leq i \leq n$$

- soft margin (e.g. via hinge loss):

$$\min \left[\frac{1}{n} \sum_{i=1}^{n} \max(0, 1 - y_i(\boldsymbol{w} \cdot \boldsymbol{x}_i) - b) \right] + \lambda \left|\left|\boldsymbol{w}\right|\right|^2$$

with λ being a tradeoff parameter (constrained optimization)

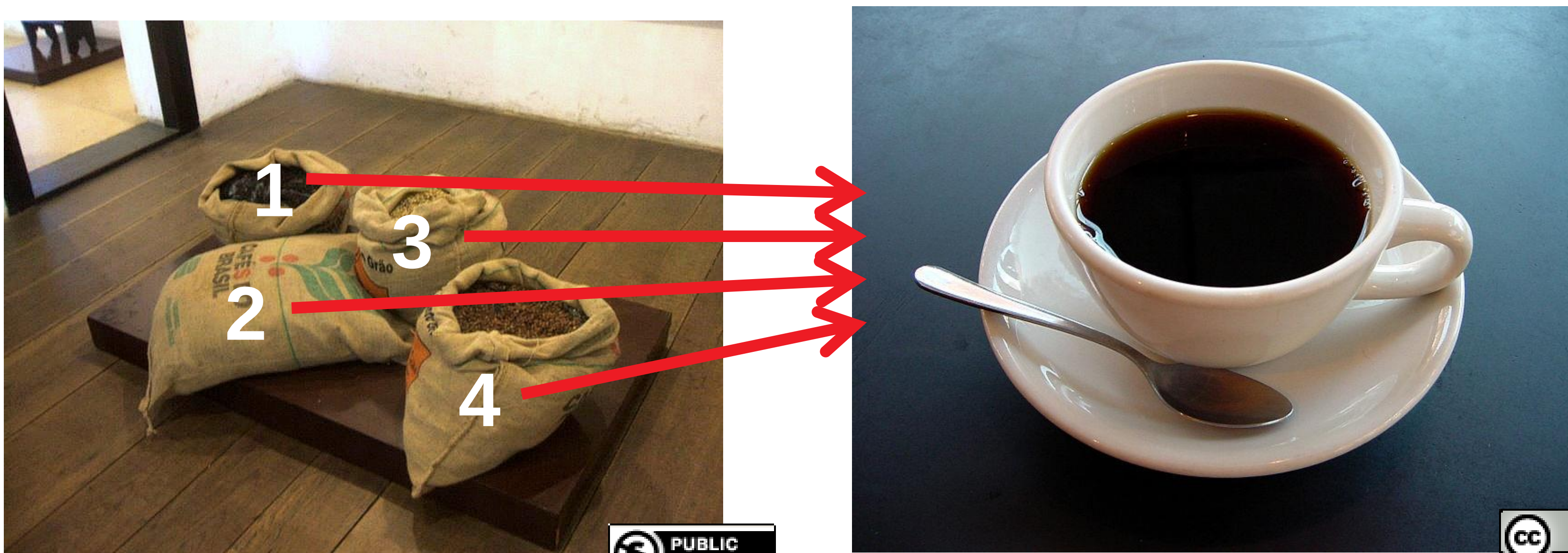
Example 9: Hyperparameter Tuning

Scenario:

- many existing algorithms (in ML and elsewhere) have internal parameters
    - “In machine learning, a hyperparameter is a parameter whose value is set before the learning process begins.” --- Wikipedia
    - can be model parameters
        - #trees in random forest
        - #nodes in neural net
        - …
    - or other generic parameters such as learning rates, …
- choice has typically a big impact and is not always obvious
- search space often mixed discrete-continuous or even categorical

Coffee Tasting Problem

- Find a mixture of coffee in order to keep the coffee taste from one year to another
- Objective function = opinion of one expert



M. Herdy: "Evolution Strategies with subjective selection", 1996

Many Problems, Many Algorithms?

Observation:

- Many problems with different properties
- For each, it seems a different algorithm?

In Practice:

  - often most important to categorize your problem first in order to find / develop the right method
  - → problem types

Problem Types

- discrete vs. continuous
  - discrete: integer (linear) programming vs. combinatorial problems
  - continuous: linear, quadratic, smooth/nonsmooth, blackbox/DFO, ...
  - both discrete&continuous variables: mixed integer problem
  - categorical variables (“no order”)
- unconstrained vs. constrained (and then which type of constraint)

Not covered in this lecture:

- deterministic vs. stochastic outcome of objective function(s)
- one or multiple objective functions

Example: Numerical Blackbox Optimization

Typical scenario in the continuous, unconstrained case:

$$\text{Optimize } f\colon \Omega \subset \mathbb{R}^n \mapsto \mathbb{R}$$

zero order blackbox: no gradients

first order blackbox also outputs gradients slides with a blueish background have not been discussed in class and, thus, are not part of the final exam

General Concepts in Optimization

- search domain
  - discrete or continuous or mixed integer or even categorical
  - finite vs. infinite dimension
- constraints
  - bound constraints (on the variables only)
  - linear/quadratic/non-linear constraints
  - blackbox constraints
  - many more

    (see e.g. Le Digabel and Wild (2015), https://arxiv.org/abs/1505.07881)

Further important aspects (in practice):

- deterministic vs. stochastic algorithms
- exact vs. approximation algorithms vs. heuristics
- anytime algorithms
- simulation-based optimization problem / expensive problem

Details on Continuous Optimization Lectures

Introduction to Continuous Optimization

- examples and typical difficulties in optimization

Mathematical Tools to Characterize Optima

- reminders about differentiability, gradient, Hessian matrix
- unconstraint optimization
  - first and second order conditions
  - convexity
- constraint optimization
  - linear programming, dual problem
  - Lagrangian, optimality conditions

Gradient-based Algorithms

- stochastic gradient
- quasi-Newton method (BFGS)

Learning in Optimization / Optimization in Machine Learning

- Stochastic gradient descent (SGD) + Adam
- CMA-ES (adaptive algorithms / Information Geometry)
- Other derivative-free algorithms: Nelder-Mead, Bayesian opt.

Continuous Optimization

- Optimize f: $\begin{cases} \Omega \subset \mathbb{R}^n \to \mathbb{R} \\ x = (x_1, \dots, x_n) \to f(x_1, \dots, x_n) \end{cases}$

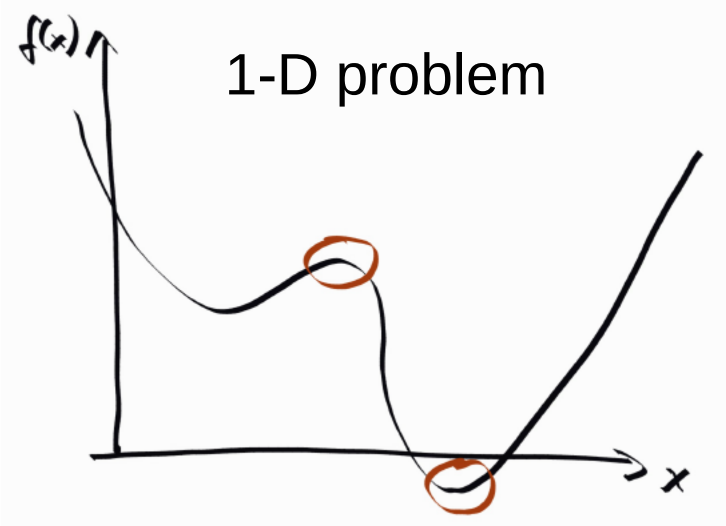
  $\in \mathbb{R}$

  unconstrained optimization

- Search space is continuous, i.e. composed of real vectors $x \in \mathbb{R}^n$

- $n =$ [dimension of the problem
  dimension of the search space $\mathbb{R}^n$ (as vector space)

1-D problem



2-D level sets

What Makes a Function Difficult to Solve?

- dimensionality

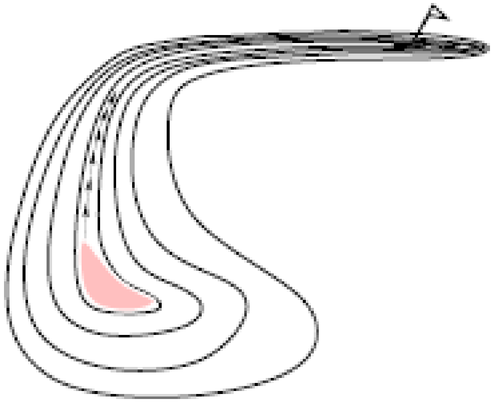
  (considerably) larger than three

- non-separability

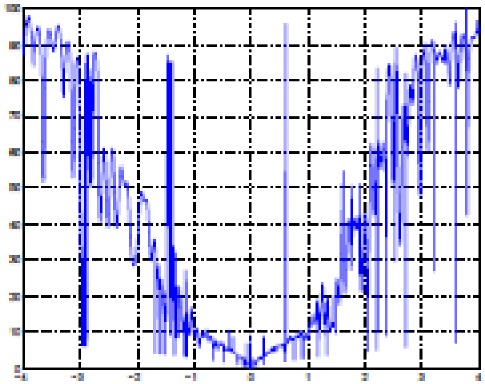
  dependencies between the objective variables

- ill-conditioning
- ruggedness

  non-smooth, discontinuous, multimodal, and/or noisy function



a narrow ridge



cut from 3D example, solvable with an evolution strategy

Curse of Dimensionality

- The term *Curse of dimensionality* (Richard Bellman) refers to problems caused by the rapid increase in volume associated with adding extra dimensions to a (mathematical) space.

Example: Consider placing 100 points onto a real interval, say $[0,1]$. How many points do you need to get a similar coverage, in terms of distance between adjacent points, in the 10-dimensional space $[0,1]^{10}$?

Curse of Dimensionality

- The term *Curse of dimensionality* (Richard Bellman) refers to problems caused by the rapid increase in volume associated with adding extra dimensions to a (mathematical) space.

Example: Consider placing 100 points onto a real interval, say $[0,1]$. How many points do you need to get a similar coverage, in terms of distance between adjacent points, in the 10-dimensional space $[0,1]^{10}$?

- Answer: This requires $100^{10} = 10^{20}$ points. The original 100 points appear now as isolated points in a vast empty space.

- Consequently, a search policy (e.g. exhaustive search) that is valuable in small dimensions might be useless in moderate or large dimensional search spaces.

Definition (Separable Problem)

A function f is separable if

$$\underset{(x_1,\ldots,x_n)}{\operatorname{argmin}} f(x_1,\ldots,x_n) = \left(\underset{x_1}{\operatorname{argmin}} f(x_1,\ldots),\ldots,\underset{x_n}{\operatorname{argmin}} f(\ldots,x_n)\right)$$

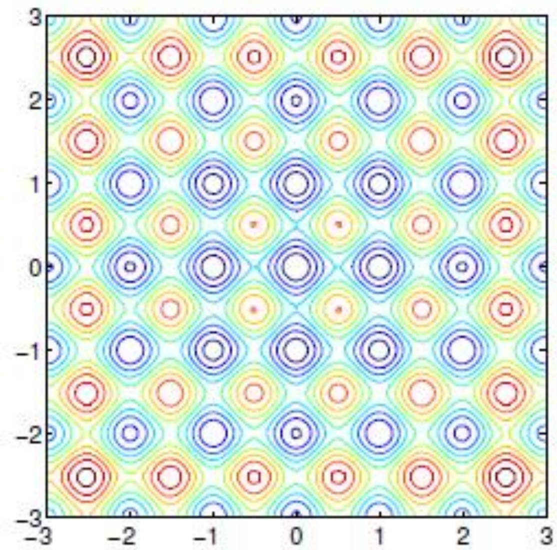
$\Longrightarrow$ it follows that f can be optimized in a sequence of n independent 1-D optimization processes

Example:

Additively decomposable functions

$$f(x_1,\ldots,x_n) = \sum_{i=1}^{n} f_i(x_i)$$

Rastrigin function

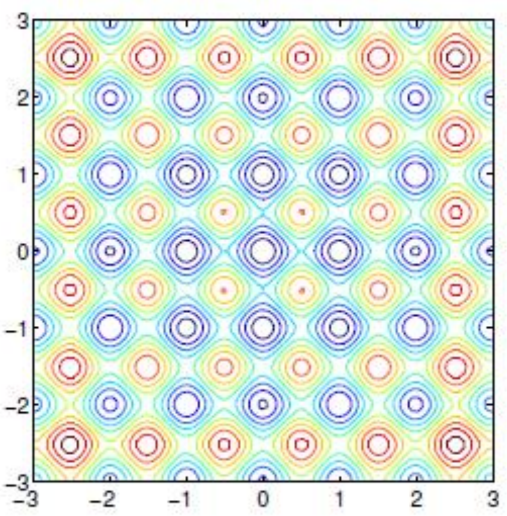


Building a non-separable problem from a separable one [1,2]

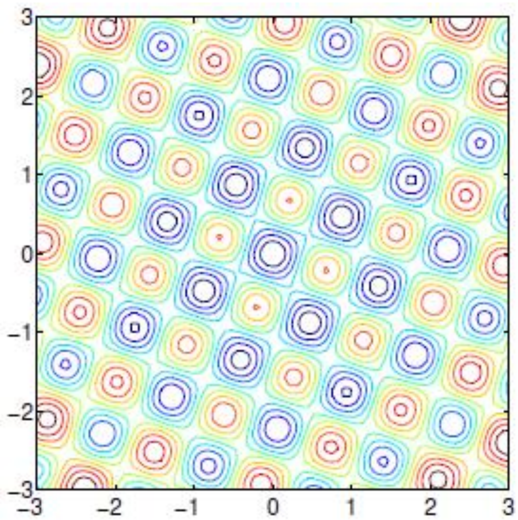
Rotating the coordinate system

- $f: \boldsymbol{x} \mapsto f(\boldsymbol{x})$ separable
- $f: \boldsymbol{x} \mapsto f(R\boldsymbol{x})$ non-separable

R rotation matrix



$\xrightarrow{R}$



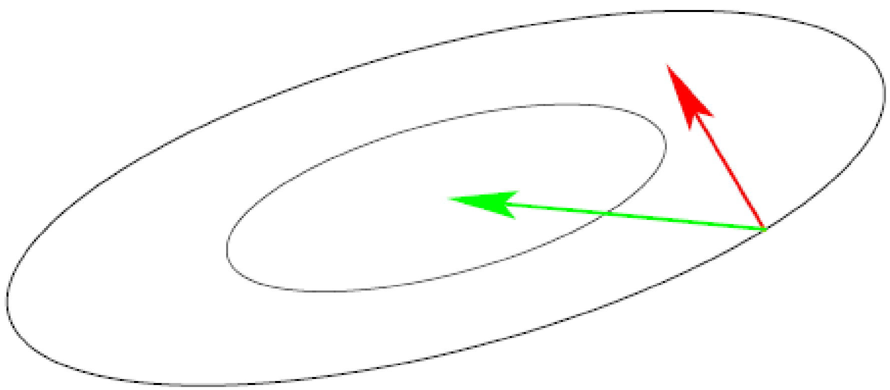
[1] N. Hansen, A. Ostermeier, A. Gawelczyk (1995). "On the adaptation of arbitrary normal mutation distributions in evolution strategies: The generating set adaptation". Sixth ICGA, pp. 57-64, Morgan Kaufmann
[2] R. Salomon (1996). "Reevaluating Genetic Algorithm Performance under Coordinate Rotation of Benchmark Functions; A survey of some theoretical and practical aspects of genetic algorithms." BioSystems, 39(3):263-278

Ill-Conditioned Problems: Curvature of Level Sets

Consider the convex-quadratic function

$$f(\boldsymbol{x}) = \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}^*)^T H(\boldsymbol{x} - \boldsymbol{x}^*) = \frac{1}{2}\sum_i h_{i,i}x_i^2 + \frac{1}{2}\sum_{i,j} h_{i,j}x_i x_j$$

H is Hessian matrix of f and symmetric positive definite



gradient direction $-f'(x)^T$

Newton direction $-H^{-1}f'(x)^T$

Ill-conditioning means squeezed level sets (high curvature). Condition number equals nine here. Condition numbers up to 10^{10} are not unusual in real-world problems.

If $H \approx I$ (small condition number of H) first order information (e.g. the gradient) is sufficient. Otherwise second order information (estimation of H^{-1}) information necessary.

Reminder: Different Notions of Optimum

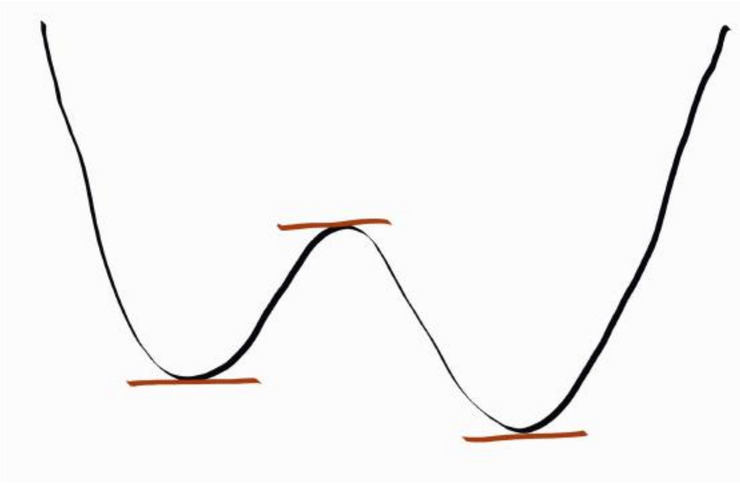
Unconstrained case

- local vs. global
  - local minimum $\boldsymbol{x}^*$: ∃ a neighborhood V of $\boldsymbol{x}^*$ such that $\forall \boldsymbol{x} \in \mathrm{V}: f(\boldsymbol{x}) \geq f(\boldsymbol{x}^*)$
  - global minimum: $\forall \boldsymbol{x} \in \Omega: f(\boldsymbol{x}) \geq f(\boldsymbol{x}^*)$
- strict local minimum if the inequality is strict

Mathematical Characterization of Optima

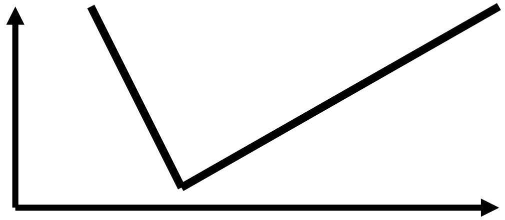
Objective: Derive general characterization of optima

Example: if $f: \mathbb{R} \to \mathbb{R}$ differentiable,
$f'(x) = 0$ at optimal points



- generalization to $f: \mathbb{R}^n \to \mathbb{R}$?
- generalization to constrained problems?

Remark: notion of optimum independent of notion of derivability

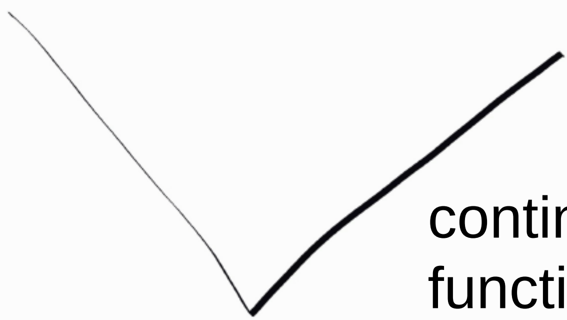


optima of such function can be easily approached by certain type of methods

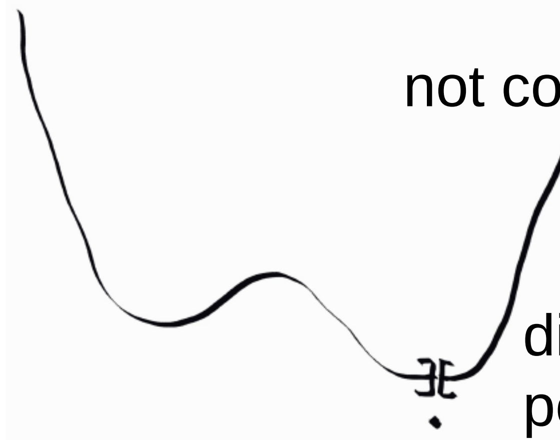
Reminder: Continuity of a Function

$f: (V, || \, ||_V) \longrightarrow (W, || \, ||_W)$ is continuous in $x \in V$ if

$\forall \epsilon > 0, \exists \eta > 0$ such that $\forall y \in V$: $||x - y||_V \leq \eta$; $||f(x) - f(y)||_W \leq \epsilon$



continuous function



not continuous

discontinuity point

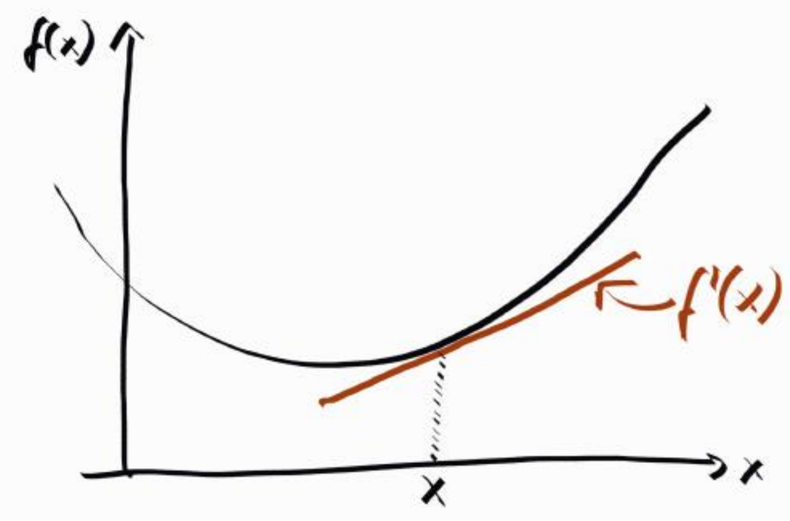
Reminder: Differentiability in 1D (n=1)

$f: \mathbb{R} \to \mathbb{R}$ is differentiable in $x \in \mathbb{R}$ if

$$\lim_{h \to 0} \frac{f(x+h) - f(x)}{h} \text{ exists, } h \in \mathbb{R}$$

Notation:

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$



The derivative corresponds to the slope of the tangent in x.

Reminder: Differentiability in 1D (n=1)

Taylor Formula (Order 1)

If f is differentiable in x then

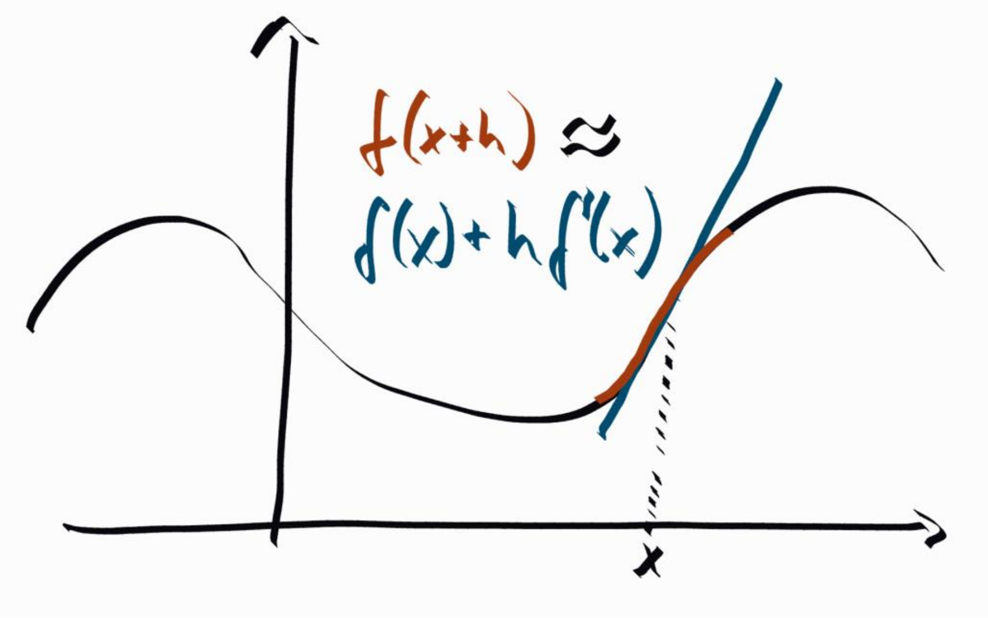
$$f(x+h) = f(x) + f'(x)h + o(||h||)$$

i.e. for h small enough, $h \longmapsto f(x+h)$ is approximated by $h \longmapsto f(x) + f'(x)h$

$h \longmapsto f(x) + f'(x)h$ is called a first order approximation of $f(x+h)$

Reminder: Differentiability in 1D (n=1)

Geometrically:



The notion of derivative of a function defined on $\mathbb{R}^n$ is generalized via this idea of a linear approximation of $f(x+h)$ for h small enough.

How to generalize this to arbitrary dimension?

Gradient Definition Via Partial Derivatives

- In $(\mathbb{R}^n, \ ||\ \ ||_2)$ where $||\boldsymbol{x}||_2 = \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle}$ is the Euclidean norm deriving from the scalar product $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \ \boldsymbol{x}^T \boldsymbol{y}$

$$\nabla f(x) = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}$$

- Reminder: partial derivative in x_0

$$\mathrm{f_i}\colon\ y \rightarrow\ f\left(x_0^1, \dots, x_0^{i-1}, y, x_0^{i+1}, \dots, x_0^n\right)$$

$$\frac{\partial f}{\partial x_i}(x_0) = {f_i}'(x_0)$$

Exercise: Gradients

Exercise:

Compute the gradients of

a) $f(x) = x_1$ with $x \in \mathbb{R}^n$

b) $f(x) = a^T x$ with $\mathrm{a}, x \in \mathbb{R}^n$

c) $f(x) = x^T x$ $(= ||\mathrm{x}||^2)$ with $x \in \mathbb{R}^n$