UNIVERSITÄT BONN

Optimization for Machine Learning in Python

Dr. Moritz Wolter

July 14, 2022

High Performance Computing and Analytics Lab, Uni Bonn

Overview

Introduction

The derivative

Optimization in a single dimension

Optimization in many dimensions

Second order optimization

Optimization for Machine Learning in Python

Overview

TODO

Introduction

Optimization

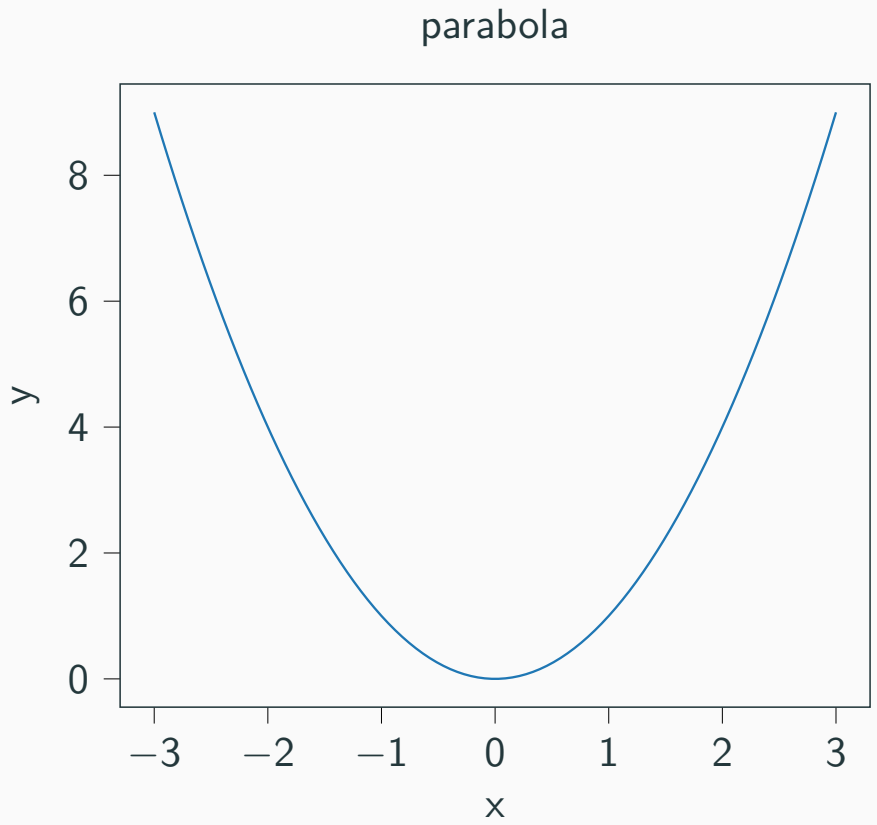
Traditionally, optimization means minimizing using a cost function $f(x)$. Given the cost, we must find the cheapest point x^* on the function, or in other words,

$$x^* = \min_{x \in \mathbb{R}} f(x) \tag{1}$$

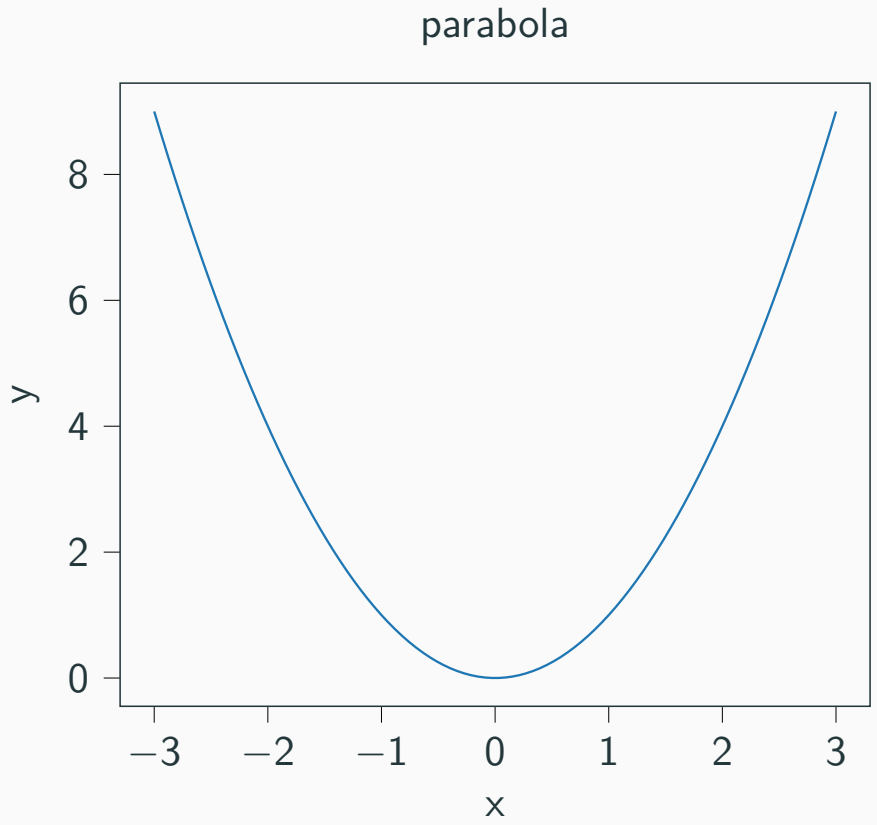
Functions

Functions are mathematical mappings. Consider for example the quadratic funtion, $f(x) : \mathbb{R} \to \mathbb{R}$:

$$f(x) = x^2 \tag{2}$$



Where is the minimum?



In this case we immediately see it's at zero. Finding it algorithmically requires derivate information.

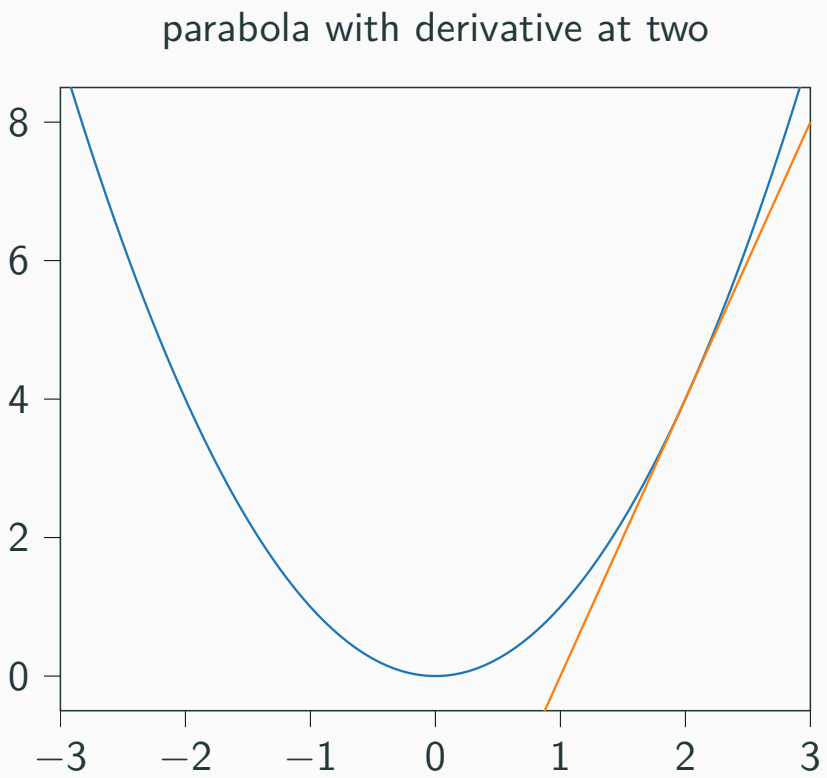
Summary

TODO

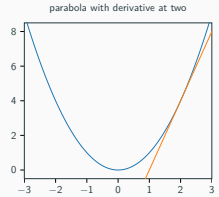
The derivative

The derivative

$$\frac{df(x)}{dx} = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h} \tag{3}$$



$$\frac{df(x)}{dx} = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h} \tag{3}$$



TODO

Derivation of the parabola derivative

$$\lim_{h \to 0} \frac{(x+h)^2 - x^2}{h} = \lim_{h \to 0} \frac{x^2 + 2xh + h^2 - x^2}{h} \tag{4}$$

$$= \lim_{h \to 0} \frac{2xh + h^2}{h} \tag{5}$$

$$= \lim_{h \to 0} \frac{h(2x + h)}{h} \tag{6}$$

$$= \lim_{h \to 0} 2x + h \tag{7}$$

$$= 2x \tag{8}$$

Derivation of the parabola derivative

$$\lim_{h\to 0}\frac{(x+h)^2-x^2}{h} = \lim_{h\to 0}\frac{x^2+2xh+h^2-x^2}{h} \quad (4)$$
$$= \lim_{h\to 0}\frac{2xh+h^2}{h} \quad (5)$$
$$= \lim_{h\to 0}\frac{h(2x+h)}{h} \quad (6)$$
$$= \lim_{h\to 0} 2x+h \quad (7)$$
$$= 2x \quad (8)$$

Derive on the board:

$$\lim_{h\to 0}\frac{(x+h)^2-x^2}{h} = \lim_{h\to 0}\frac{x^2+2xh+h^2-x^2}{h} \quad (9)$$
$$= \lim_{h\to 0}\frac{2xh+h^2}{h} \quad (10)$$
$$= \lim_{h\to 0}\frac{h(2x+h)}{h} \quad (11)$$
$$= \lim_{h\to 0} 2x+h \quad (12)$$
$$= 2x \quad (13)$$

Summary

TODO

Optimization in a single dimension

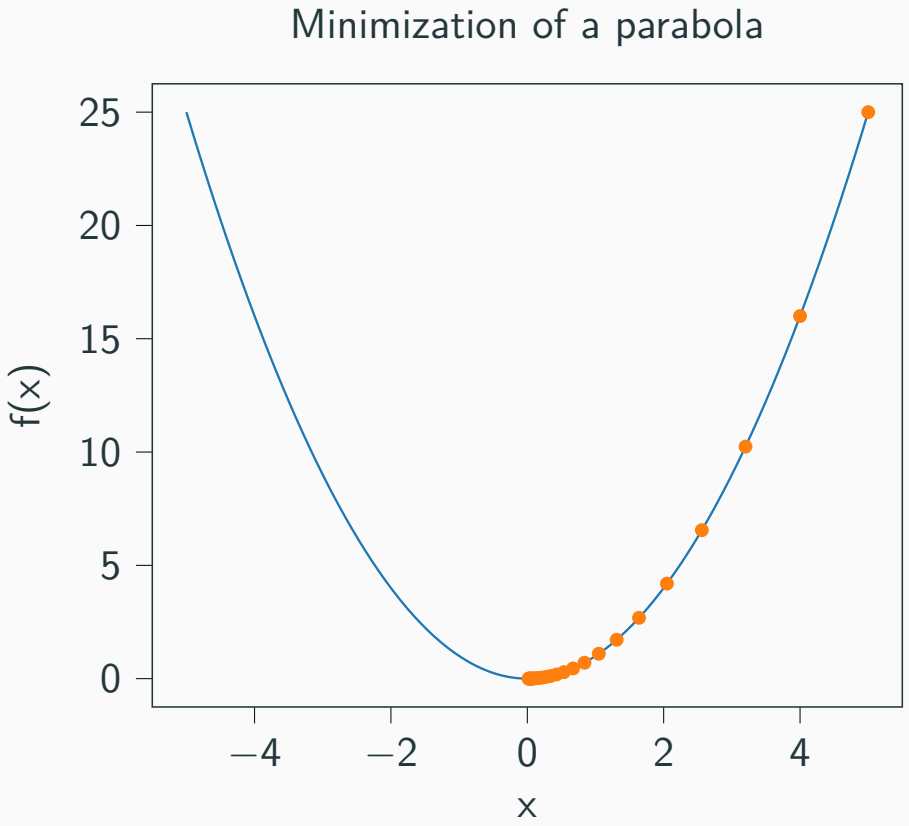
Steepest descent

To find a minumum we descent along the gradient, with n denoting the step number, $\alpha \in \mathbb{R}$ the step size and $\frac{df}{dx}$ the derivate of f along $x \in \mathbb{R}$:

$$x_n = x_{n-1} - \alpha \cdot \frac{df}{dx}. \tag{14}$$

Steepest descent on the parabola

Working with the initial position $x_0 = 5$ and a step size of $\alpha = 0.1$ for 25 steps leads to:



Summary

TODO

Optimization in many dimensions

Multidimensional problems

The Rosenbrock test function:

$$f(x, y) = (a - x)^2 + b(y - x^2)^2 \tag{15}$$

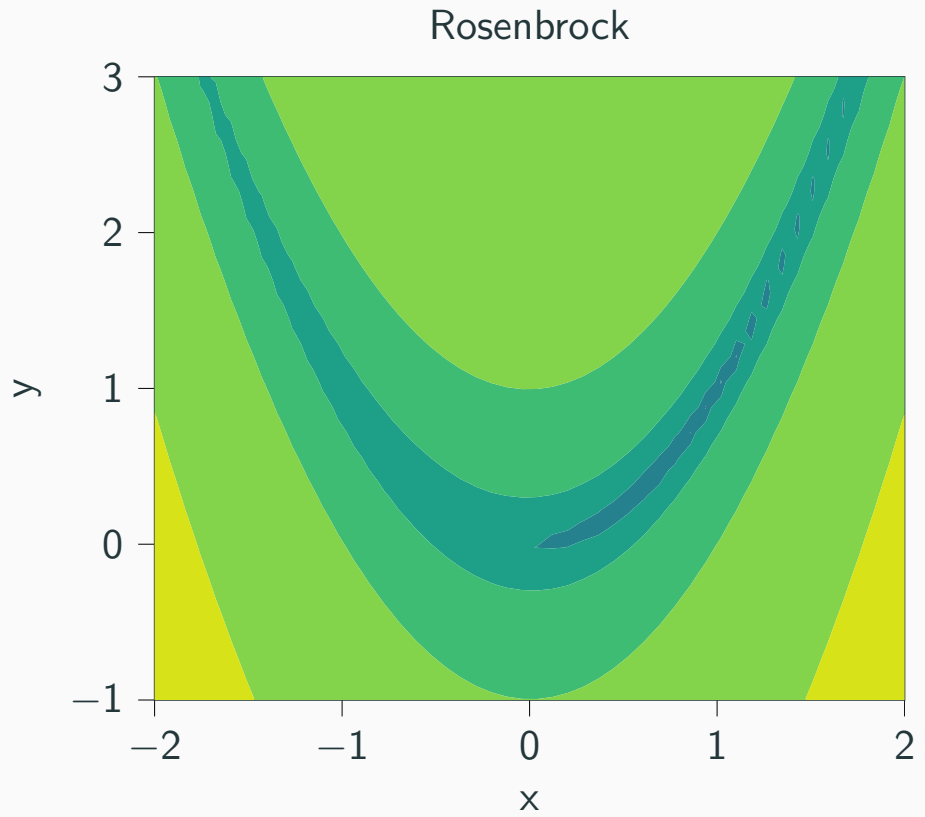


Figure: Rosenbrock function with a=1 and b=100 .

The gradient

$$\nabla f = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix} \tag{16}$$

TODO

$$\nabla f = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_d} \end{pmatrix} \tag{16}$$

Rosenbrock gradient

Recall the Rosenbrock function:

$$f(x, y) = (a - x)^2 + b(y - x^2)^2 \tag{17}$$

$$\nabla f(x, y) = \begin{pmatrix} -2a + 2x - 4byx + 4bx^3 \\ 2by - 2bx^2 \end{pmatrix} \tag{18}$$

Rosenbrock gradient

Recall the Rosenbrock function:

$$f(x,y) = (a-x)^2 + b(y-x^2)^2 \tag{17}$$

$$\nabla f(x,y) = \begin{pmatrix} -2a + 2x - 4byx + 4bx^3 \\ 2by - 2bx^2 \end{pmatrix} \tag{18}$$

On the board derive:

$$f(x,y) = (a-x)^2 + b(y-x^2)^2 \tag{19}$$

$$= a^2 - 2ax + x^2 + b(y^2 - 2yx^2 + x^4) \tag{20}$$

$$= a^2 - 2ax + x^2 + by^2 - 2byx^2 + bx^4 \tag{21}$$

$$\Rightarrow \frac{\partial f(x,y)}{\partial x} = -2a + 2x - 4byx + 4bx^3 \tag{22}$$

$$\Rightarrow \frac{\partial f(x,y)}{\partial y} = 2by - 2bx^2 \tag{23}$$

Gradient descent

Initial position: $x_0 = [0.1, 3.]$,
Gradient step size: $\alpha = 0.01$

$$x_n = x_{n-1} - \alpha \cdot \nabla f(\mathbf{x}) \tag{24}$$

n denotes the step number, ∇ the gradient operator, and $f(\mathbf{x})$ a vector valued function.

Gradient descent on the Rosenbrock function

Rosenbrock Optimization

Gradient descent with momentum

Rosenbrock Optimization

summary

TODO

Second order optimization

Second order optimization

TODO