Optimization

Sibylle Hess

Technische Universiteit Eindhoven University of Technology

Optimization

Unconstrained Optimization Problem

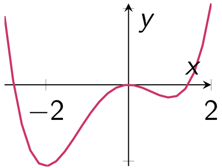
Given an objective function $f : \mathbb{R}^n \to \mathbb{R}$, the objective of an unconstrained optimization problem is:

$$\min_{x \in \mathbb{R}^n} f(\mathrm{x})$$

We say that:

- $\mathrm{x}^* \in \arg\min_{x \in \mathbb{R}^n} f(\mathrm{x})$ is a minimizer
- $\min_{\mathrm{x} \in \mathbb{R}^n} f(\mathrm{x})$ is the minimum

Local and Global Minimizers



global minimizer: $x^* = -2$
local minimizer: $x_3 = 1$

A global minimizer is a vector x^* satisfying

$$f(x^*) \leq f(x) \text{ for all } x \in \mathbb{R}^n$$

A local minimizer is a vector x_0 satisfying

$$f(x_0) \leq f(x) \text{ for all } x \in \mathcal{N}_\epsilon(x_0),$$

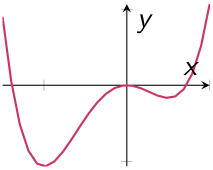
where $\mathcal{N}_\epsilon(x_0) = \{x \in \mathbb{R}^n | \|x - x_0\| \leq \epsilon\}$

How can we solve an unconstrained optimization problem?

With FONC and SONC.

Finding Stationary Points: our Minimizer Candidates

Every local minimizer x_0 is a stationary point: $\frac{d}{dx}f(x_0) = 0$
(a.k.a. 1st order necessary condition)



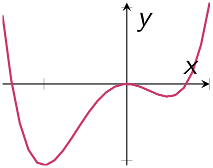
$$f(x) = \frac{1}{4}x^4 + \frac{1}{3}x^3 - x^2$$
$$\frac{d}{dx}f(x) = x^3 + x^2 - 2x$$
$$\frac{d^2}{dx^2}f(x) = 3x^2 + 2x - 2$$

$$\frac{d}{dx}f(x) = 0 \quad \Leftrightarrow \quad x_1 = -2, x_2 = 0, x_3 = 1$$

Possible local minimizers: $x_1 = -2, x_2 = 0, x_3 = 1$

Identifying Minimizers by the Curvature

Every stationary point x_0 with increasing function values around it is a local minimizer: $\frac{d}{dx}f(x_0) = 0$ & $\frac{d^2}{dx^2}f(x_0) \geq 0$
(a.k.a 2nd order sufficient condition)



$$f(x) = \frac{1}{4}x^4 + \frac{1}{3}x^3 - x^2$$
$$\frac{d}{dx}f(x) = x^3 + x^2 - 2x$$
$$\frac{d^2}{dx^2}f(x) = 3x^2 + 2x - 2$$

$$\frac{d^2}{dx^2}f(-2) = 6 \geq 0, \quad \frac{d^2}{dx^2}f(0) = -2 < 0, \quad \frac{d^2}{dx^2}f(1) = 3 \geq 0$$

We identify the local minimizers $x_1 = -2$ and $x_2 = 3$.

What Happens in Higher Dimensions?

The derivative of a function $f : \mathbb{R}^d \to \mathbb{R}$ is given by its partial derivatives:

$$\frac{\partial f(x)}{\partial x} = \begin{pmatrix} \frac{\partial f(x)}{\partial x_1} & \dots & \frac{\partial f(x)}{\partial x_d} \end{pmatrix} \in \mathbb{R}^{1 \times d} \qquad \text{(Jacobian)}$$

$$\nabla_x f(x) = \begin{pmatrix} \frac{\partial f(x)}{\partial x_1} \\ \vdots \\ \frac{\partial f(x)}{\partial x_d} \end{pmatrix} \in \mathbb{R}^d \qquad \text{(Gradient)}$$

First Order Necessary Condition

FONC

If x is a local minimizer of $f : \mathbb{R}^d \to \mathbb{R}$ and f is continuously differentiable in an open neighborhood of x, then

$$\nabla f(\mathrm{x}) = 0$$

A vector x is called stationary point if $\nabla f(\mathrm{x}) = 0$.

Second Order Necessary Condition

SONC

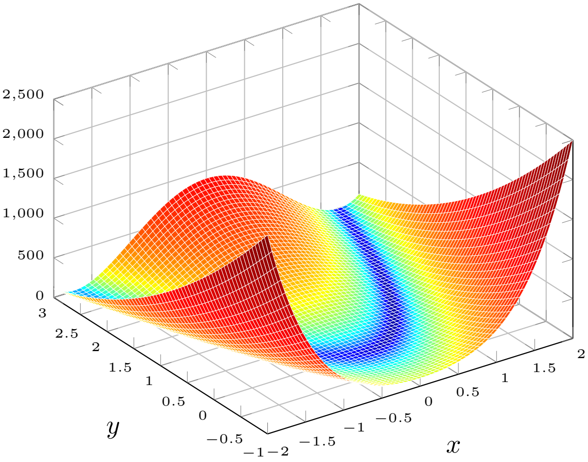
If x is a local minimizer of $f : \mathbb{R}^d \to \mathbb{R}$ and $\nabla^2 f$ is continuous in an open neighborhood of x, then

$$\nabla f(\mathrm{x}) = 0 \text{ and } \nabla^2 f(\mathrm{x}) \text{ is positive semidefinite}$$

A matrix $A \in \mathbb{R}^{d \times d}$ is positive semidefinite if

$$\mathrm{x}^\top A\mathrm{x} \geq 0 \text{ for all } \mathrm{x} \in \mathbb{R}^d$$

Example: the Rosenbrock Function



Candidate Minimizers of the Rosenbrock Function

The Rosenbrock function is given by

$$f(\mathrm{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

We compute the gradient and set it to zero:

$$\nabla f(\mathrm{x}) = \begin{pmatrix} 400x_1(x_1^2 - x_2) + 2(x_1 - 1) \\ 200(x_2 - x_1^2) \end{pmatrix} = 0,$$

$$\Leftrightarrow \mathrm{x} = (1, 1)$$

According to FONC we have one stationary point, i.e., one local minimizer candidate at $\mathrm{x}_0 = (1, 1)$.

Evaluating the Curvature at the Candidate Minimizer

We compute the Hessian function of f at $\mathrm{x}_0 = (1, 1)$:

$$\nabla^2 f(\mathrm{x}) = 200 \begin{pmatrix} 1 & -2x_1 \\ -2x_1 & 6x_1^2 - 2x_2 + 0.01 \end{pmatrix}$$

$$\nabla^2 f(\mathrm{x}_0) = 200 \begin{pmatrix} 1 & -2 \\ -2 & 4.01 \end{pmatrix}$$

We check now if the Hessian is positive semi-definite at the stationary point. Let $\mathrm{x} \in \mathbb{R}^2$, then

$$\mathrm{x}^\top \nabla^2 f(\mathrm{x}_0)\mathrm{x} = (x_1 - 2x_2)^2 + 0.01x_2^2 \geq 0$$

Hence, $\nabla^2 f(\mathrm{x}_0)$ is p.s.d. and $\mathrm{x}_0 = (1, 1)$ satisfies the SONC for a local minimizer of f.

Nice, so finding local minimizers is not a big deal

IF we have an unconstrained objective with an objective function which is twice continuously differentiable.

Let's consider a more complex setting:

Introducing Constraints

Constrained Optimization Problem

Given

- an objective function $f : \mathbb{R}^d \to \mathbb{R}$ and
- constraint functions $c_i, g_k : \mathbb{R}^d \to \mathbb{R}$,

then the objective of an constrained optimization problem is

$$
\begin{aligned}
\min_{x \in \mathbb{R}^n} \; & f(\mathrm{x}) & \\
\text{s.t. } \; & c_i(\mathrm{x}) = 0 & \text{for } 1 \leq i \leq m, \\
& g_k(\mathrm{x}) \geq 0 & \text{for } 1 \leq k \leq l
\end{aligned}
$$

We call the set of vectors satisfying the constraints the feasible set:

$$\mathcal{C} = \{\mathrm{x} \mid c_i(\mathrm{x}) = 0, g_k(\mathrm{x}) \geq 0 \text{ for } 1 \leq i \leq m, 1 \leq k \leq m\}.$$

How can we solve a constrained optimization tasks?

If we have constraints, then FONC and SONC do not help much anymore..

Can we transform the constrained problem into an unconstrained one?

Yes, maybe, kind of, with the Lagrangian..

The Lagrangian Function

Given a constrained optimization task:

$$\begin{aligned} \min_{\mathrm{x}\in\mathbb{R}^n} \; & f(\mathrm{x}) \\ \text{s.t.} \;\; & c_i(\mathrm{x}) = 0 && \text{for } 1 \le i \le m, \\ & g_k(\mathrm{x}) \ge 0 && \text{for } 1 \le k \le l \end{aligned}$$

The Lagrangian function is defined as

$$\mathcal{L}(\mathrm{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\mathrm{x}) - \sum_{i=1}^{m} \boldsymbol{\lambda}_i c_i(\mathrm{x}) - \sum_{k=1}^{l} \boldsymbol{\mu}_k g_k(\mathrm{x}).$$

The parameters $\boldsymbol{\lambda}_i \in \mathbb{R}$ and $\boldsymbol{\mu}_i \ge 0$ are called Lagrange multipliers.

The Lagrangian Forms a Lower Bound of the Objective

For feasible $x \in \mathcal{C}$ and $\boldsymbol{\lambda} \in \mathbb{R}^m, \boldsymbol{\mu} \in \mathbb{R}^l$ we have

$$\mathcal{L}(x, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(x) - \sum_{i=1}^{m} \boldsymbol{\lambda}_i \underbrace{c_i(x)}_{=0} - \sum_{k=1}^{l} \underbrace{\boldsymbol{\mu}_k}_{\geq 0} \underbrace{g_k(x)}_{\geq 0} \leq f(x)$$

This introduces the dual objective function $\mathcal{L}_{dual}$:

$$\min_{x \in \mathcal{C}} f(x) \geq \inf_{x \in \mathcal{C}} \mathcal{L}(x, \boldsymbol{\lambda}, \boldsymbol{\mu}) \geq \inf_{x \in \mathbb{R}^d} \mathcal{L}(x, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \mathcal{L}_{dual}(\boldsymbol{\lambda}, \boldsymbol{\mu})$$

Primal and Dual Problem

Primal Problem

$$\min_{x \in \mathbb{R}^n} f(\mathrm{x})$$
$$\text{s.t. } c_i(\mathrm{x}) = 0 \quad 1 \le i \le m,$$
$$g_k(\mathrm{x}) \ge 0 \quad 1 \le k \le l$$

Dual Problem

$$\max_{\boldsymbol{\lambda}, \boldsymbol{\mu}} \mathcal{L}_{dual}(\boldsymbol{\lambda}, \boldsymbol{\mu})$$
$$\text{s.t. } \boldsymbol{\lambda} \in \mathbb{R}^m, \boldsymbol{\mu} \in \mathbb{R}^l_+$$

The solution to the primal problem is always bounded below by the solution to the dual problem $f^* \ge \mathcal{L}^*_{dual}$.

Some conditions yield that $f^* = \mathcal{L}^*_{dual}$, then solving the dual is equivalent to solving the primal.

Okay, so if I have an unconstrained optimization problem then I try FONC and SONC...

... and if I have a constrained optimization problem then I can try to solve it over the dual problem.

What if I can't compute the minimizers by these approaches?

Do Numerical Optimization

Approximating a Minimizer

If the minimizers can not be computed directly/analytically, then Numerical Optimization can come to the rescue.

The general scheme of numerical optimization methods is:

function $\textsc{Optimizer}(f)$
    $x_0 \leftarrow \textsc{Initialize}(x_0)$
    for $t \in \{1, \ldots, t_{max} - 1\}$ **do**
        $x_{t+1} \leftarrow \textsc{Update}(x_t, f)$
    end for
    return $x_{t_{max}}$
end function

Coordinate Descent

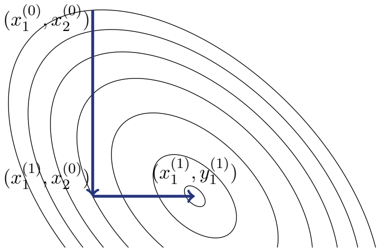
Sometimes, we can not determine the minimum analytically, but the minimum in a coordinate direction.

Coordinate descent update:

$$x_i^{(t+1)} \leftarrow \arg\min_{x_i} f(x_1^{(t)}, \ldots, x_i, \ldots x_d^{(t)}), \quad 1 \leq i \leq d$$

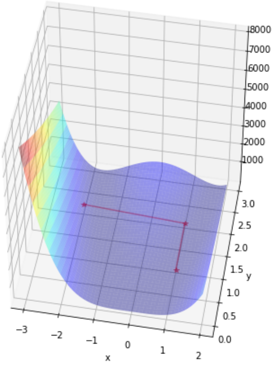
Coordinate descent minimizes in every step, hence

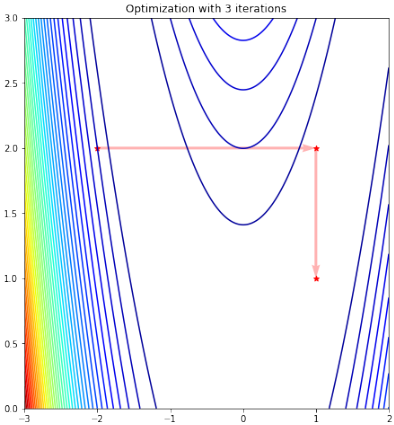
$$f(\mathrm{x}^{(0)}) \geq f(\mathrm{x}^{(1)}) \geq f(\mathrm{x}^{(2)}) \geq \ldots$$



Example: Coordinate Descent on the Rosenbrock Function

$$\arg\min_{x_1 \in \mathbb{R}} f(x_1, x_2) = 1 \qquad \arg\min_{x_2 \in \mathbb{R}} f(x_1, x_2) = x_1^2$$





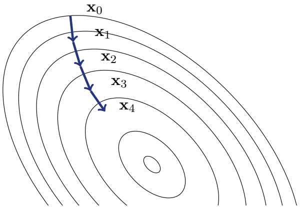
Gradient Descent

If we do not know much but a gradient, we can apply gradient descent.

Gradient descent update:

$$x_{t+1} \leftarrow x_t - \eta \nabla f(x_t)$$

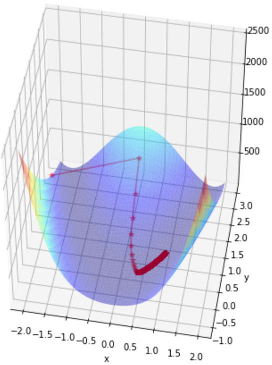
where η is the step size.

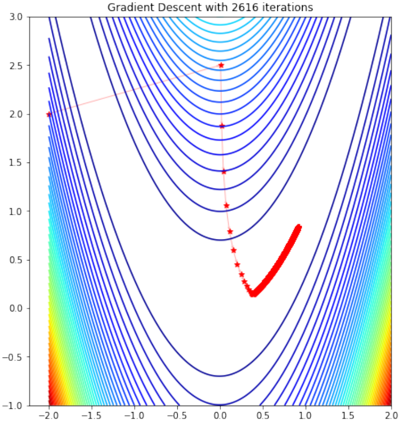


The negative gradient points into the direction of steepest descent. Hence, for a small enough step size we obtain a sequence

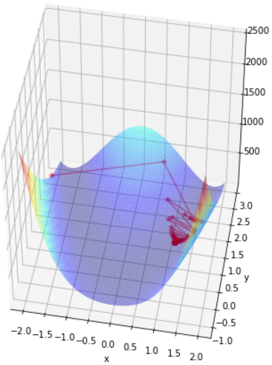
$$f(x_0) \geq f(x_1) \geq f(x_2) \geq \dots$$

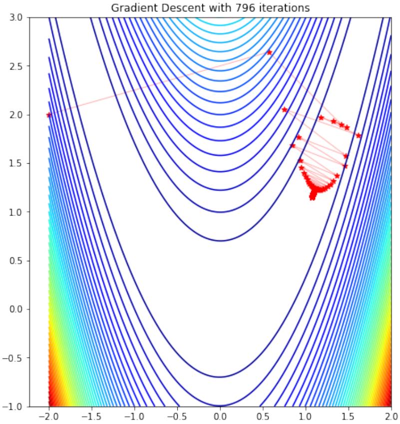
Example: Gradient Descent with $\eta = 0.00125$ on the Rosenbrock Function



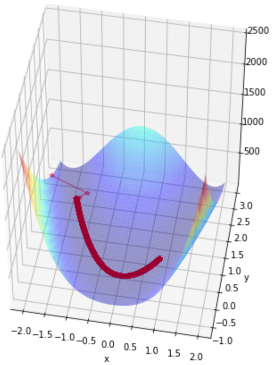


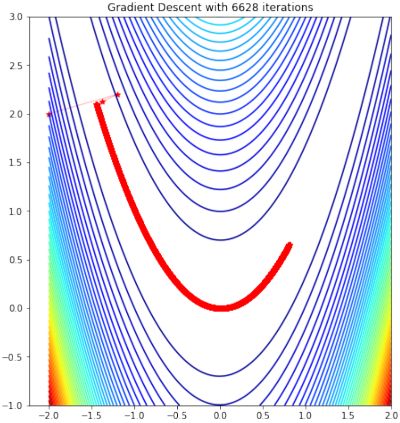
Example: Gradient Descent with $\eta = 0.0016$ on the Rosenbrock Function





Example: Gradient Descent with $\eta = 0.0005$ on the Rosenbrock Function





With every run of numerical optimization I get one minimizer candidate. How do I know if I can do bettter?

Analyze the optimization problem!

When every local minimizer is a global minimizer:

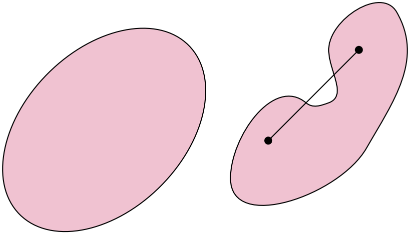
Convex Optimization

Convex Sets

A set $\mathcal{X} \subseteq \mathbb{R}^d$ is convex if and only if the line segment between every pair of points in the set is in the set.

That is, for all $x, y \in \mathcal{X}$ and $\alpha \in [0, 1]$

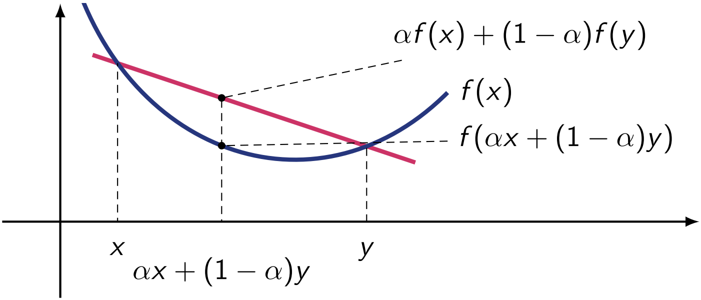
$$\alpha \mathsf{x} + (1 - \alpha)\mathsf{y} \in \mathcal{X}.$$



Convex Functions

A function $f : \mathbb{R}^d \to \mathbb{R}$ is convex if and only if for every $\alpha \in [0, 1]$, and $\mathrm{x}, \mathrm{y} \in \mathbb{R}^d$:

$$f(\alpha \mathrm{x} + (1 - \alpha)\mathrm{y}) \leq \alpha f(\mathrm{x}) + (1 - \alpha) f(\mathrm{y})$$



Convex Optimization Problem

Given

- a convex objective function $f : \mathbb{R}^d \to \mathbb{R}$ and
- a convex feasible set $\mathcal{C} \subseteq \mathbb{R}^d$

then the objective of a convex optimization problem is

$$\min_{x \in \mathbb{R}^n} \; f(\mathrm{x})$$

$$\text{s.t. } \mathrm{x} \in \mathcal{C}$$

Properties of Convex Functions

Theorem

If f is convex, then every local minimizer x^* is a global minimizer.

Note: not every function with one global and local minimum is convex (cf. Rosenbrock function).

Proof (Sketch): Assume that a convex function f has a local minimizer x_{loc} which is not a global minimizer: $f(x_{loc}) > f(x^*)$. Then going towards x^* from x_{loc} minimizes the function value, hence x_{loc} is not a local minimizer.

Properties of Convex Functions

- Nonnegative weighted sums of convex functions are convex: for all $\lambda_1, \ldots, \lambda_k \geq 0$ and $f_1, \ldots, f_k$ convex, then the function

$$f(\mathrm{x}) = \lambda_1 f_1(\mathrm{x}) + \ldots + \lambda_k f_k(\mathrm{x})$$

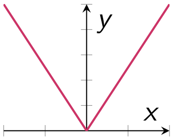
  is convex.
- If $g : \mathbb{R}^d \to \mathbb{R}^k$, $g(\mathrm{x}) = A\mathrm{x} + \mathrm{b}$ is an affine map, and $f : \mathbb{R}^k \to \mathbb{R}$ is a convex function, then the composition

$$f(g(\mathrm{x})) = f(A\mathrm{x} + \mathrm{b})$$

  is a convex function.

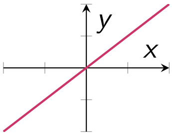
Proof: Exercise

Examples of Convex Functions



Every norm is a convex function: for any $x, y \in \mathbb{R}^d$ and $\alpha \in [0, 1]$ we have:

$$\begin{aligned} \|\alpha x + (1-\alpha)y\| &\leq \|\alpha x\| + \|(1-\alpha)y\| \\ &\leq |\alpha|\|x\| + |1-\alpha|\|y\| \\ &= \alpha\|x\| + (1-\alpha)\|y\| \end{aligned}$$



Every linear function f is convex and concave ($-f$ is convex): for any $x, y \in \mathbb{R}^d$ and $\alpha \in [0, 1]$ we have:

$$f(\alpha x + (1-\alpha)y) = \alpha f(x) + (1-\alpha)f(y)$$

Okay nice, so if my optimization problem is convex then I *only* need to find a local minimium (for example by gradient descent).

How do I compute the gradient? Do I always have to compute the partial derivatives?

No, use the chain rule whenever you can!

Gradient Descent needs a Gradient

There are two ways to define the derivative of a function

$$f : \mathbb{R}^{n\times d} \to \mathbb{R}.$$

$$\frac{\partial f(X)}{\partial X} = \begin{pmatrix} \frac{\partial f(X)}{\partial X_{11}} & \cdots & \frac{\partial f(X)}{\partial X_{n1}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f(X)}{\partial X_{1d}} & \cdots & \frac{\partial f(X)}{\partial X_{nd}} \end{pmatrix} \in \mathbb{R}^{d\times n} \qquad \text{(Jacobian)}$$

$$\nabla f(X) = \begin{pmatrix} \frac{\partial f(X)}{\partial X_{11}} & \cdots & \frac{\partial f(X)}{\partial X_{1d}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f(X)}{\partial X_{n1}} & \cdots & \frac{\partial f(X)}{\partial X_{nd}} \end{pmatrix} \in \mathbb{R}^{n\times d} \qquad \text{(Gradient)}$$

Be careful!

This notation is not used by all authors!

The Jacobian of f

$$f : \mathbb{R}^d \to \mathbb{R} \qquad \frac{\partial f(\mathrm{x})}{\partial \mathrm{x}} = \begin{pmatrix} \frac{\partial f(\mathrm{x})}{\partial x_1} & \cdots & \frac{\partial f(\mathrm{x})}{\partial x_d} \end{pmatrix} \in \mathbb{R}^{1\times d}$$

$$f : \mathbb{R}^{n\times d} \to \mathbb{R} \qquad \frac{\partial f(X)}{\partial X} = \begin{pmatrix} \frac{\partial f(X)}{\partial X_{11}} & \cdots & \frac{\partial f(X)}{\partial X_{n1}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f(X)}{\partial X_{1d}} & \cdots & \frac{\partial f(X)}{\partial X_{nd}} \end{pmatrix} \in \mathbb{R}^{d\times n}$$

$$\mathrm{f} : \mathbb{R} \to \mathbb{R}^c \qquad \frac{\partial \mathrm{f}(x)}{\partial x} = \begin{pmatrix} \frac{\partial f_1(x)}{\partial x} \\ \vdots \\ \frac{\partial f_c(x)}{\partial x} \end{pmatrix} \in \mathbb{R}^c$$

$$\mathrm{f} : \mathbb{R}^d \to \mathbb{R}^c \qquad \frac{\partial \mathrm{f}(\mathrm{x})}{\partial \mathrm{x}} = \begin{pmatrix} \frac{\partial f_1(\mathrm{x})}{\partial x_1} & \cdots & \frac{\partial f_1(\mathrm{x})}{\partial x_d} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_c(\mathrm{x})}{\partial x_1} & \cdots & \frac{\partial f_c(\mathrm{x})}{\partial x_d} \end{pmatrix} \in \mathbb{R}^{c\times d}$$

The Gradient of f

$$f : \mathbb{R}^d \to \mathbb{R} \qquad \nabla_x f(x) = \begin{pmatrix} \frac{\partial f(x)}{\partial x_1} \\ \vdots \\ \frac{\partial f(x)}{\partial x_d} \end{pmatrix} \in \mathbb{R}^d$$

$$f : \mathbb{R}^{n\times d} \to \mathbb{R} \qquad \nabla_X f(X) = \begin{pmatrix} \frac{\partial f(X)}{\partial X_{11}} & \cdots & \frac{\partial f(X)}{\partial X_{1d}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f(X)}{\partial X_{n1}} & \cdots & \frac{\partial f(X)}{\partial X_{nd}} \end{pmatrix} \in \mathbb{R}^{n\times d}$$

$$\mathrm{f} : \mathbb{R} \to \mathbb{R}^c \qquad \nabla_x \mathrm{f}(x) = \begin{pmatrix} \frac{\partial f_1(x)}{\partial x} & \cdots & \frac{\partial f_c(x)}{\partial x} \end{pmatrix} \in \mathbb{R}^{1\times c}$$

$$\mathrm{f} : \mathbb{R}^d \to \mathbb{R}^c \qquad \nabla_x \mathrm{f}(x) = \begin{pmatrix} \frac{\partial f_1(x)}{\partial x_1} & \cdots & \frac{\partial f_c(x)}{\partial x_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_1(x)}{\partial x_d} & \cdots & \frac{\partial f_c(x)}{\partial x_d} \end{pmatrix} \in \mathbb{R}^{d\times c}$$

Most Important Derivation Rules

$$\nabla_x f(x) = \left(\frac{\partial f(x)}{\partial x}\right)^\top$$

$$\frac{\partial \alpha f(x) + g(x)}{\partial x} = \alpha \frac{\partial f(x)}{\partial x} + \frac{\partial g(x)}{\partial x} \qquad \text{(linearity)}$$

$$\frac{\partial f(g(x))}{\partial x} = \frac{\partial f(g)}{\partial g}\frac{\partial g(x)}{\partial x} \qquad \text{(chain rule)}$$

Exercise: Derive the following equations:

$$\frac{\partial \|x\|^2}{\partial x}, \frac{\partial b - ax}{\partial x}, \frac{\partial b - Ax}{\partial x}, \nabla_x \|b - Ax\|^2, \nabla_X \|D - YX^\top\|^2$$

Most Important Derivation Rules

$$\nabla_x f(x) = \left(\frac{\partial f(x)}{\partial x}\right)^\top$$

$$\frac{\partial \alpha f(x) + g(x)}{\partial x} = \alpha \frac{\partial f(x)}{\partial x} + \frac{\partial g(x)}{\partial x} \qquad \text{(linearity)}$$

$$\frac{\partial f(g(x))}{\partial x} = \frac{\partial f(g)}{\partial g}\frac{\partial g(x)}{\partial x} \qquad \text{(chain rule)}$$

Exercise: Derive the following equations:

$$\frac{\partial \|x\|^2}{\partial x}, \frac{\partial b - ax}{\partial x}, \frac{\partial b - Ax}{\partial x}, \nabla_x \|b - Ax\|^2, \nabla_X \|D - YX^\top\|^2$$