Computational imaging & learning: optimisation & data-driven learning I

Luca Calatroni
MaLGa center, DIBRIS, UniGe
MMS, Istituto Italiano di Tecnologia
Genoa, Italy

Mini-Corso Data Science @ UniPD

Università degli studi di Padova

May 11-14 2026

Table of contents

Introduction

Goal

For $\mathbf{y} \in \mathbb{R}^M$, $\mathbf{A} \in \mathbb{R}^{M \times N}$ and $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_M)$:

$$\bar{\mathbf{u}} \quad \text{s.t.} \quad \mathbf{y} = \mathbf{A}\bar{\mathbf{u}} + \mathbf{n}$$

Through Bayesian approaches/MAP. Seek $\mathbf{u}^* \approx \bar{\mathbf{u}}$:

$$\mathbf{u}^* \in \underset{\mathbf{u} \in \mathbb{R}^N}{\arg\min} \; F(\mathbf{u})$$

with F **model** for problem at hand ($F = f_{\mathbf{y}} + g$). Consider **algorithm** Algo:

$$\mathbf{u}^* \approx \mathbf{u}^K = \text{Algo}^K(\mathbf{u}^0; \texttt{params}) \quad .$$

Goal

For $\mathbf{y} \in \mathbb{R}^M$, $\mathbf{A} \in \mathbb{R}^{M \times N}$ and $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_M)$:

$$\bar{\mathbf{u}} \quad \text{s.t.} \quad \mathbf{y} = \mathbf{A}\bar{\mathbf{u}} + \mathbf{n}$$

Through Bayesian approaches/MAP. Seek $\mathbf{u}^* \approx \bar{\mathbf{u}}$:

$$\mathbf{u}^* \in \underset{\mathbf{u} \in \mathbb{R}^N}{\arg\min} \; F(\mathbf{u})$$

with F **model** for problem at hand ($F = f_{\mathbf{y}} + g$). Consider **algorithm** Algo:

$\mathbf{u}^* \approx \mathbf{u}^K = \text{Algo}^K(\mathbf{u}^0; \texttt{params}) \quad \rightarrow \quad \mathbf{u}^* \approx \mathbf{u}^K = \text{Algo}^K_{\text{NN}_\theta}(\mathbf{u}^0; \texttt{params})$, NN_θ is a NN.

Why?

Goal

For $\mathbf{y} \in \mathbb{R}^M$, $\mathbf{A} \in \mathbb{R}^{M \times N}$ and $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_M)$:

$$\bar{\mathbf{u}} \quad \text{s.t.} \quad \mathbf{y} = \mathbf{A}\bar{\mathbf{u}} + \mathbf{n}$$

Through Bayesian approaches/MAP. Seek $\mathbf{u}^* \approx \bar{\mathbf{u}}$:

$$\mathbf{u}^* \in \underset{\mathbf{u} \in \mathbb{R}^N}{\arg\min} \; F(\mathbf{u})$$

with F **model** for problem at hand ($F = f_{\mathbf{y}} + g$). Consider **algorithm** Algo:

$\mathbf{u}^* \approx \mathbf{u}^K = \text{Algo}^K(\mathbf{u}^0; \texttt{params}) \quad \rightarrow \quad \mathbf{u}^* \approx \mathbf{u}^K = \text{Algo}^K_{\text{NN}_\theta}(\mathbf{u}^0; \texttt{params})$, NN_θ is a NN.

* **Physics-informed learning**:

$$\texttt{params} \leftrightarrow (\texttt{physical parameters}, \texttt{algorithmic parameters})$$

* **Convergence guarantees**:

$$\exists \text{ function } F_\theta \quad \text{s.t.} \quad F_\theta(\mathbf{u}^{k+1}) \leq F_\theta(\mathbf{u}^k)? \qquad \mathbf{u}^K \rightarrow ??$$

- **Reconstruction guarantees**:

$$\|\mathbf{u}^* - \bar{\mathbf{u}}\| \leq C(\sigma), \quad \text{where } \sigma \text{ is the noise level.}$$

Optimisation for imaging problems

$$\mathbf{u}^* \in \arg\min_{\mathbf{u}\in\mathbb{R}^N} \; F(\mathbf{u}) := f_{\mathbf{y}}(\mathbf{u}) + g(\mathbf{u})$$

- $f_{\mathbf{y}}(\mathbf{u}) = f(\mathbf{u}; \mathbf{y}, \mathbf{A})$: fidelity term, "discrepancy" function describing fit with the linear model + noise statistics, e.g. $f_{\mathbf{y}}(\mathbf{u}) = \frac{1}{2}\|\mathbf{A}\mathbf{u} - \mathbf{y}\|^2, \mathrm{KL}(\mathbf{y}; \mathbf{A}\mathbf{u}+\epsilon)$
- $g(\mathbf{u})$: regularisation term, it encodes *a priori* information expected on the desired solution, $g(\mathbf{u}) = \lambda\|\mathbf{u}\|_1, \lambda\|\mathbf{D}\mathbf{u}\|_{2,1}, \ldots$

Optimisation for imaging problems

$$\mathbf{u}^* \in \arg\min_{\mathbf{u}\in\mathbb{R}^N} \; F(\mathbf{u}) := f_{\mathbf{y}}(\mathbf{u}) + g(\mathbf{u})$$

- $f_{\mathbf{y}}(\mathbf{u}) = f(\mathbf{u}; \mathbf{y}, \mathbf{A})$: fidelity term, "discrepancy" function describing fit with the linear model + noise statistics, e.g. $f_{\mathbf{y}}(\mathbf{u}) = \frac{1}{2}\|\mathbf{Au} - \mathbf{y}\|^2, \mathrm{KL}(\mathbf{y}; \mathbf{Au}+\epsilon)$
- $g(\mathbf{u})$: regularisation term, it encodes *a priori* information expected on the desired solution, $g(\mathbf{u}) = \lambda\|\mathbf{u}\|_1, \lambda\|\mathbf{Du}\|_{2,1}, \ldots$

$f\_\mathbf{y}$ is often 'easy' to choose, while g is not.
Learn it through optimisation?

Gold references in optimisation (and imaging)

- R. Tyrrell Rockafeller, *Convex Analysis*, Princeton University Press, 1970.
- S. Boyd, L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.
- N. Parikh, S. Boyd, *Proximal Algorithms*, Foundations and Trends in Optimization, 2013.
- A. Beck, *First-order methods in optimization*, Volume 25, MOS-SIAM series on Optimization, 2017.
- A. Chambolle, T. Pock, *An introduction to continuous optimization for imaging*, Acta Numerica, 2016

Crash-course on convex optimisation

Basic notions

- $(\mathbb{R}^d, \langle \mathbf{v}, \mathbf{w} \rangle = \mathbf{v}^T \mathbf{w})$ with Euclidean norm $\|\cdot\|$.
  Extensions to general (infinite-dimensional) Hilbert setting often straightforward.
- $\overline{\mathbb{R}} := \mathbb{R} \cup \{+\infty\}$, $\mathbb{R}_+ := \{\alpha \in \mathbb{R} : \alpha \geq 0\}$, $\mathbb{R}_{++} := \{\alpha \in \mathbb{R} : \alpha > 0\}$
- Closed ball of radius $\delta > 0$ centered at $\mathbf{v} \in \mathbb{R}^d$:

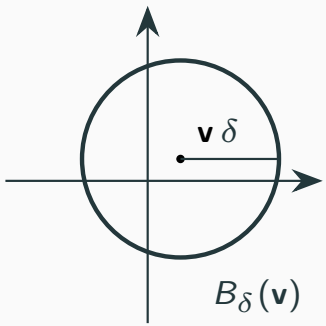
$$B_\delta(\mathbf{v}) = \left\{ \mathbf{w} \in \mathbb{R}^d : \|\mathbf{v} - \mathbf{w}\| \leq \delta \right\}$$

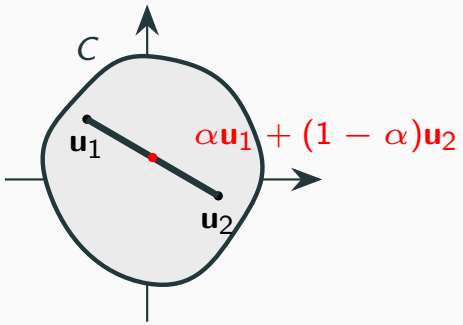
- Convex set $C \subset \mathbb{R}^d$:

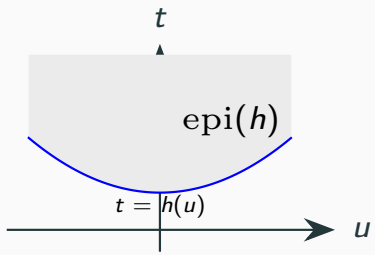
$$(\forall \mathbf{u}_1, \mathbf{u}_2 \in C)\ \forall \alpha \in [0,1] \quad \alpha \mathbf{u}_1 + (1-\alpha)\mathbf{u}_2 \in C$$

- Epigraph of $h : \mathbb{R}^d \to \overline{\mathbb{R}}$:

$$\operatorname{epi}(h) = \left\{ (\mathbf{u}, t) \in \mathbb{R}^d \times \mathbb{R} :\ t \geq h(\mathbf{u}) \right\}$$





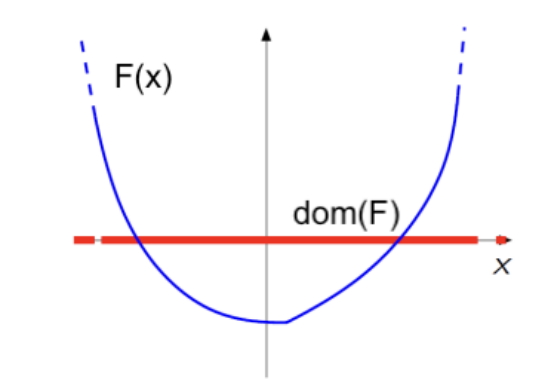


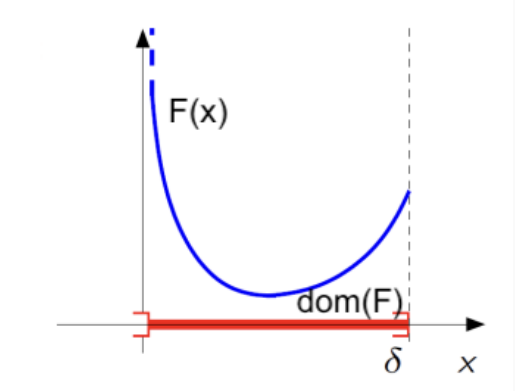
Basic notions for well-posedness

$F : \mathbb{R}^N \to \overline{\mathbb{R}}$:

- F is proper if

$$\mathrm{dom}(F) := \left\{ \mathbf{u} \in \mathbb{R}^N : F(\mathbf{u}) < +\infty \right\} \neq \emptyset.$$

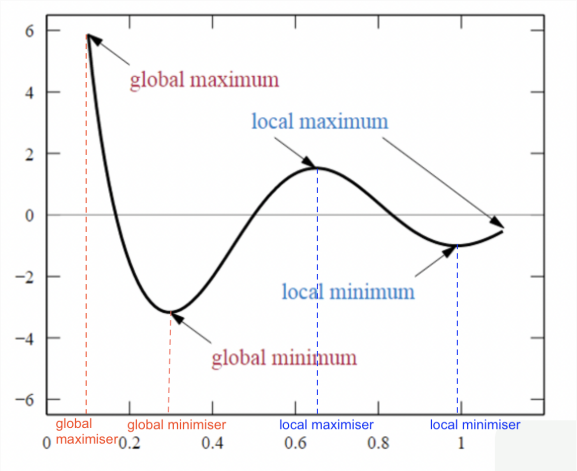




Basic notions for well-posedness

$F : \mathbb{R}^N \to \overline{\mathbb{R}}$:

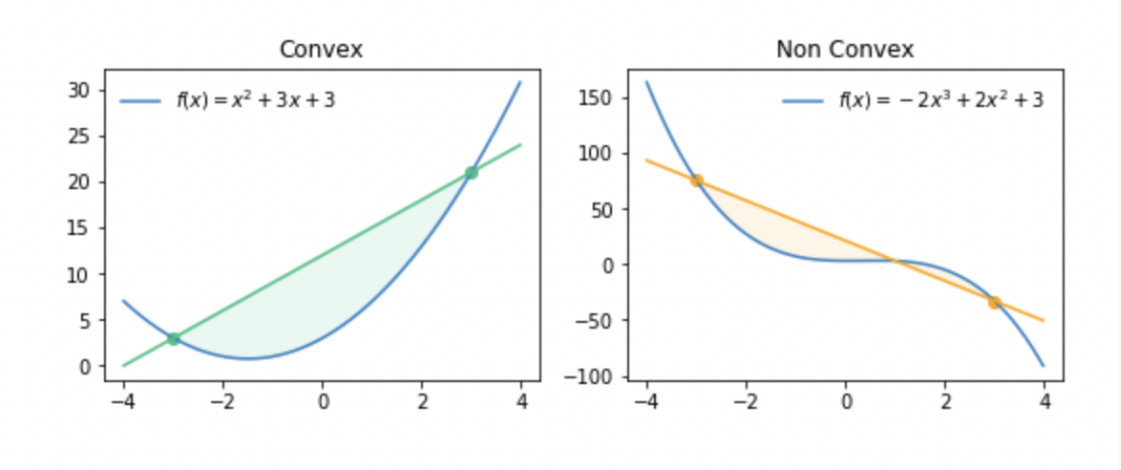
- F is proper if
$$\mathrm{dom}(F) := \left\{ \mathbf{u} \in \mathbb{R}^N : F(\mathbf{u}) < +\infty \right\} \neq \emptyset.$$
- $\mathbf{u}^* \in \mathbb{R}^N$ is **global minimiser**: $F(\mathbf{u}^*) \leq F(\mathbf{u})$ for every $\mathbf{u} \in \mathbb{R}^N$.
$$\arg\min F = \left\{ \mathbf{u}^* \in \mathbb{R}^N : \mathbf{u}^* \text{ is a global minimiser of } F \right\} \subset \mathbb{R}^N$$
- $\mathbf{u}^* \in \mathbb{R}^N$ is **local minimiser**: there exists $\delta > 0$ and $B_\delta(\mathbf{u}^*)$ such that $F(\mathbf{u}^*) \leq F(\mathbf{u})$ for every $\mathbf{u} \in B_\delta(\mathbf{u}^*)$.



Basic notions for well-posedness

$F : \mathbb{R}^N \to \overline{\mathbb{R}}$:

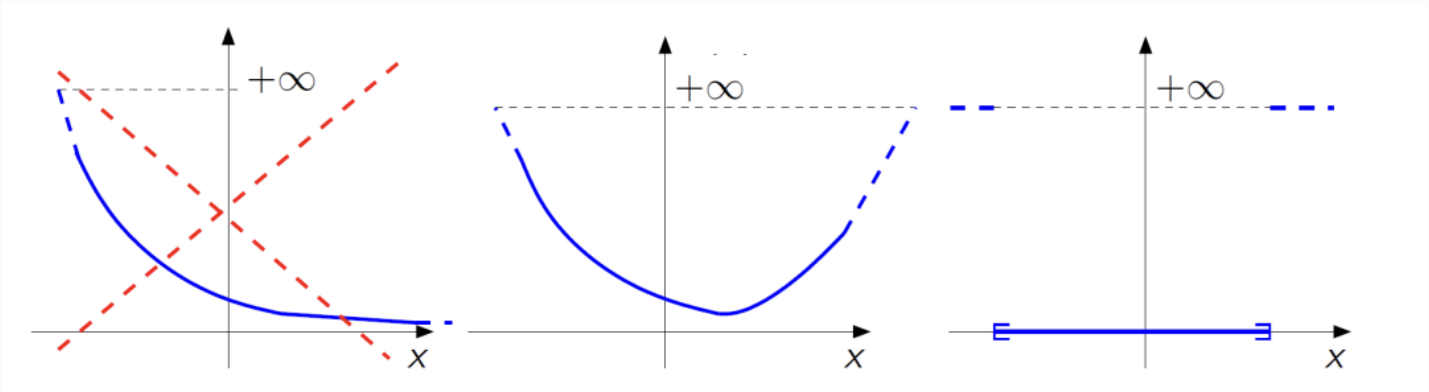
- F is proper if
$$\mathrm{dom}(F) := \left\{ \mathbf{u} \in \mathbb{R}^N : F(\mathbf{u}) < +\infty \right\} \neq \emptyset.$$
- $\mathbf{u}^* \in \mathbb{R}^N$ is **global minimiser**: $F(\mathbf{u}^*) \leq F(\mathbf{u})$ for every $\mathbf{u} \in \mathbb{R}^N$.
$$\arg\min F = \left\{ \mathbf{u}^* \in \mathbb{R}^N : \mathbf{u}^* \text{ is a global minimiser of } F \right\} \subset \mathbb{R}^N$$
- $\mathbf{u}^* \in \mathbb{R}^N$ is **local minimiser**: there exists $\delta > 0$ and $B_\delta(\mathbf{u}^*)$ such that $F(\mathbf{u}^*) \leq F(\mathbf{u})$ for every $\mathbf{u} \in B_\delta(\mathbf{u}^*)$.
- F is convex if $\forall \mathbf{u}_1, \mathbf{u}_2 \in \mathbb{R}^N$, $\forall \alpha \in [0, 1]$:
$$F(\alpha \mathbf{u}_1 + (1-\alpha)\mathbf{u}_2) \leq \alpha F(\mathbf{u}_1) + (1-\alpha) F(\mathbf{u}_2). \quad \Leftrightarrow \quad \mathrm{epi}(F) \text{ is convex.}$$



Basic notions for well-posedness

$F : \mathbb{R}^N \to \overline{\mathbb{R}}$:

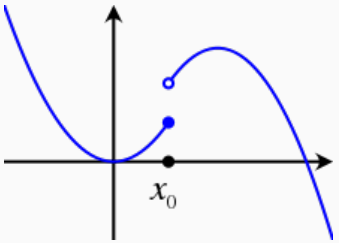
- F is proper if
$$\mathrm{dom}(F) := \left\{ \mathbf{u} \in \mathbb{R}^N : F(\mathbf{u}) < +\infty \right\} \neq \emptyset.$$
- $\mathbf{u}^* \in \mathbb{R}^N$ is **global minimiser**: $F(\mathbf{u}^*) \leq F(\mathbf{u})$ for every $\mathbf{u} \in \mathbb{R}^N$.
$$\arg\min F = \left\{ \mathbf{u}^* \in \mathbb{R}^N : \mathbf{u}^* \text{ is a global minimiser of } F \right\} \subset \mathbb{R}^N$$
- $\mathbf{u}^* \in \mathbb{R}^N$ is **local minimiser**: there exists $\delta > 0$ and $B_\delta(\mathbf{u}^*)$ such that $F(\mathbf{u}^*) \leq F(\mathbf{u})$ for every $\mathbf{u} \in B_\delta(\mathbf{u}^*)$.
- F is convex if $\forall \mathbf{u}_1, \mathbf{u}_2 \in \mathbb{R}^N$, $\forall \alpha \in [0,1]$:
$$F(\alpha \mathbf{u}_1 + (1-\alpha)\mathbf{u}_2) \leq \alpha F(\mathbf{u}_1) + (1-\alpha) F(\mathbf{u}_2). \quad \Leftrightarrow \quad \mathrm{epi}(F) \text{ is convex.}$$
- F is coercive if:
$$\lim_{\|\mathbf{u}\| \to +\infty} F(\mathbf{u}) = +\infty.$$



Basic notions for well-posedness

$F : \mathbb{R}^N \to \overline{\mathbb{R}}$:

- F is proper if
$$\mathrm{dom}(F) := \left\{ \mathbf{u} \in \mathbb{R}^N : F(\mathbf{u}) < +\infty \right\} \neq \emptyset.$$
- $\mathbf{u}^* \in \mathbb{R}^N$ is **global minimiser**: $F(\mathbf{u}^*) \leq F(\mathbf{u})$ for every $\mathbf{u} \in \mathbb{R}^N$.
$$\arg\min F = \left\{ \mathbf{u}^* \in \mathbb{R}^N : \mathbf{u}^* \text{ is a global minimiser of } F \right\} \subset \mathbb{R}^N$$
- $\mathbf{u}^* \in \mathbb{R}^N$ is **local minimiser**: there exists $\delta > 0$ and $B_\delta(\mathbf{u}^*)$ such that $F(\mathbf{u}^*) \leq F(\mathbf{u})$ for every $\mathbf{u} \in B_\delta(\mathbf{u}^*)$.
- F is convex if $\forall \mathbf{u}_1, \mathbf{u}_2 \in \mathbb{R}^N$, $\forall \alpha \in [0,1]$:
$$F(\alpha \mathbf{u}_1 + (1-\alpha)\mathbf{u}_2) \leq \alpha F(\mathbf{u}_1) + (1-\alpha) F(\mathbf{u}_2). \quad \Leftrightarrow \quad \mathrm{epi}(F) \text{ is convex.}$$
- F is coercive if:
$$\lim_{\|\mathbf{u}\| \to +\infty} F(\mathbf{u}) = +\infty.$$
- F is lower semi-continuous, $\forall (\mathbf{u}^k) \to \mathbf{u}$:
$$F(\mathbf{u}) \leq \liminf_{k \to +\infty} F(\mathbf{u}_k) = \lim_{k \to +\infty} \inf \left\{ F(\mathbf{u}_j) : j \geq k \right\}.$$



Existence of minimisers

Theorem (existence of minimisers)

If $F : \mathbb{R}^N \to \overline{\mathbb{R}}$ is proper, l.s.c. and coercive, then $\arg\min F \neq \emptyset$.

Note: generalises the Bolzano-Weirestrass theorem for *continuous* F:

$$\min_{\mathbf{u} \in C} F(\mathbf{u})$$

for compact $C \subset \mathbb{R}^N$ s.t. $C \cap \mathrm{dom}(F) \neq \emptyset$.

Existence of minimisers

Theorem (existence of minimisers)

If $F : \mathbb{R}^N \to \overline{\mathbb{R}}$ is proper, l.s.c. and coercive, then $\arg\min F \neq \emptyset$.

Note: generalises the Bolzano-Weirestrass theorem for *continuous* F:

$$\min_{\mathbf{u} \in C} F(\mathbf{u})$$

for compact $C \subset \mathbb{R}^N$ s.t. $C \cap \mathrm{dom}(F) \neq \emptyset$.

Theorem (convex case)

If $F : \mathbb{R}^N \to \overline{\mathbb{R}}$ is proper, l.s.c., coercive and convex, then every local minimiser is a global minimiser.

Theorem (existence+uniqueness of minimisers)

If $F : \mathbb{R}^N \to \overline{\mathbb{R}}$ is proper, l.s.c., coercive and strictly convex. Then, F admits a unique minimiser, hence:

$$\arg\min F = \{\mathbf{u}^*\}.$$

Crash-course on convex optimisation

Smooth optimisation

Gâteaux differentiability

Suitable notion of "∇f"?

Definition (Gâteaux differentiability)

Let $f : \mathbb{R}^N \to \overline{\mathbb{R}}$ be proper and let $\mathbf{u} \in \mathrm{dom}(f)$. For $\mathbf{v} \in \mathbb{R}^N$, we denote the *directional derivative* in $\mathbf{u}$ along the direction $\mathbf{v}$ as the limit

$$f'(\mathbf{u})[\mathbf{v}] := \lim_{t \to 0^+} \frac{f(\mathbf{u} + t\mathbf{v}) - f(\mathbf{u})}{t},$$

when it exists. If there exists $\mathbf{w} \in \mathbb{R}^N$ such that:

$$(\forall \mathbf{v} \in \mathbb{R}^N) \quad f'(\mathbf{u})[\mathbf{v}] = \langle \mathbf{w}, \mathbf{v} \rangle,$$

we say that f is *Gâteaux differentiable* at $\mathbf{u}$ and denote by $\nabla f(\mathbf{u}) \equiv \mathbf{w}$ the *Gâteaux gradient* of f at $\mathbf{u}$. f is differentiable if it is at all points $\mathbf{u} \in \mathrm{dom}(f)$.

Gâteaux differentiability

Suitable notion of "∇f"?

Definition (Gâteaux differentiability)

Let $f : \mathbb{R}^N \to \overline{\mathbb{R}}$ be proper and let $\mathbf{u} \in \mathrm{dom}(f)$. For $\mathbf{v} \in \mathbb{R}^N$, we denote the *directional derivative* in $\mathbf{u}$ along the direction $\mathbf{v}$ as the limit

$$f'(\mathbf{u})[\mathbf{v}] := \lim_{t \to 0^+} \frac{f(\mathbf{u} + t\mathbf{v}) - f(\mathbf{u})}{t},$$

when it exists. If there exists $\mathbf{w} \in \mathbb{R}^N$ such that:

$$(\forall \mathbf{v} \in \mathbb{R}^N) \quad f'(\mathbf{u})[\mathbf{v}] = \langle \mathbf{w}, \mathbf{v} \rangle,$$

we say that f is *Gâteaux differentiable* at $\mathbf{u}$ and denote by $\nabla f(\mathbf{u}) \equiv \mathbf{w}$ the *Gâteaux gradient* of f at $\mathbf{u}$. f is differentiable if it is at all points $\mathbf{u} \in \mathrm{dom}(f)$.

Example: $f_{\mathbf{y}}(\mathbf{u}) = \frac{1}{2}\|\mathbf{A}\mathbf{u} - \mathbf{y}\|^2$.

$$f'_{\mathbf{y}}(\mathbf{u})[\mathbf{v}] = \lim_{t \to 0} \frac{\frac{1}{2}\|\mathbf{A}(\mathbf{u} + t\mathbf{v}) - \mathbf{y}\|^2 - \frac{1}{2}\|\mathbf{A}\mathbf{u} - \mathbf{y}\|^2}{t} = \lim_{t \to 0} \frac{\frac{1}{2}\|\mathbf{A}\mathbf{u} - \mathbf{y} + t\mathbf{A}\mathbf{v}\|^2 - \frac{1}{2}\|\mathbf{A}\mathbf{u} - \mathbf{y}\|^2}{t}$$

$$= \lim_{t \to 0} \frac{\cancel{\frac{1}{2}\|\mathbf{A}\mathbf{u} - \mathbf{y}\|^2} + t^2\|\mathbf{A}\mathbf{v}\|^2 + t\langle \mathbf{A}\mathbf{u} - \mathbf{y}, \mathbf{A}\mathbf{v} \rangle - \cancel{\frac{1}{2}\|\mathbf{A}\mathbf{u} - \mathbf{y}\|^2}}{t}$$

$$= \langle \mathbf{A}\mathbf{u} - \mathbf{y}, \mathbf{A}\mathbf{v} \rangle = \langle \mathbf{A}^\top(\mathbf{A}\mathbf{u} - \mathbf{y}), \mathbf{v} \rangle = \langle \nabla f_{\mathbf{y}}(\mathbf{u}), \mathbf{v} \rangle$$

Optimality conditions and relations with convexity

Theorem (Fermat's rule)

Let $f : \mathbb{R}^N \to \overline{\mathbb{R}}$ be proper, convex and differentiable at point $\mathbf{u}^* \in \mathrm{dom}(f)$. Then:

$$\mathbf{u}^* \in \arg\min \ f \quad \Longleftrightarrow \quad \nabla f(\mathbf{u}^*) = 0.$$

Optimality conditions and relations with convexity

Theorem (Fermat's rule)

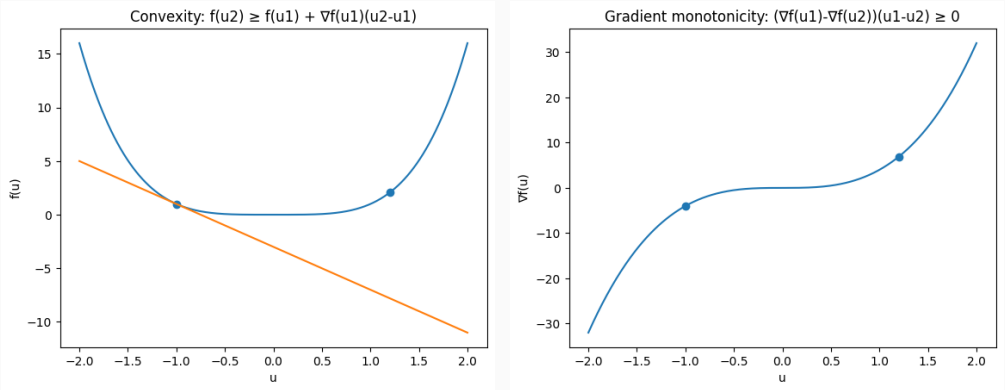
Let $f : \mathbb{R}^N \to \overline{\mathbb{R}}$ be proper, convex and differentiable at point $\mathbf{u}^* \in \mathrm{dom}(f)$. Then:

$$\mathbf{u}^* \in \arg\min\ f \quad \Longleftrightarrow \quad \nabla f(\mathbf{u}^*) = 0.$$

Proposition (Differentiability and convexity)

Let $f : \mathbb{R}^N \to \overline{\mathbb{R}}$ be proper, convex and differentiable everywhere on $\mathrm{dom}(f)$. Then, the following statements are equivalent:

1. f is convex;
2. (*above the tangent*) $\forall \mathbf{u}_1, \mathbf{u}_2 \in \mathrm{dom}(f)$, $f(\mathbf{u}_2) \geq f(\mathbf{u}_1) + \langle \nabla f(\mathbf{u}_1), \mathbf{u}_2 - \mathbf{u}_1 \rangle$;
3. (*monotonicity*) $\forall \mathbf{u}_1, \mathbf{u}_2 \in \mathrm{dom}(f)$, $\langle \nabla f(\mathbf{u}_1) - \nabla f(\mathbf{u}_2), \mathbf{u}_1 - \mathbf{u}_2 \rangle \geq 0$.



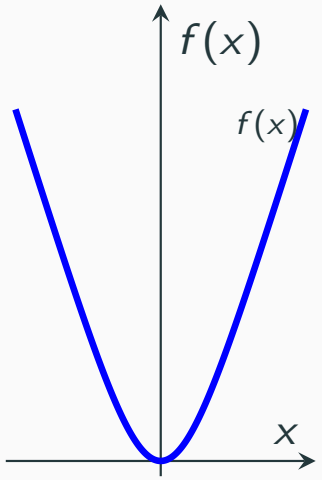
Gradient Lipschitz smoothness (L-smoothness)

Definition (L-smoothness)

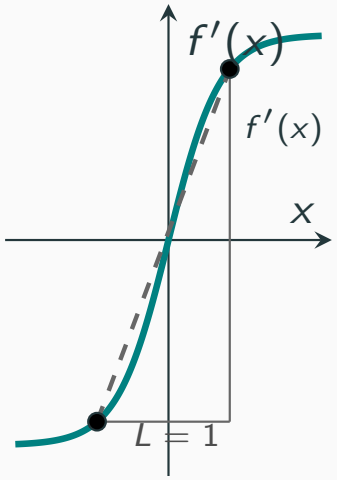
Let $f : \mathbb{R}^N \to \overline{\mathbb{R}}$ be proper, convex and differentiable. f is *L-smooth* with $L > 0$ iff:

$$\exists L > 0 : \forall \mathbf{u}_1, \mathbf{u}_2 \in \mathbb{R}^N \quad \|\nabla f(\mathbf{u}_1) - \nabla f(\mathbf{u}_2)\| \leq L\|\mathbf{u}_1 - \mathbf{u}_2\|.$$

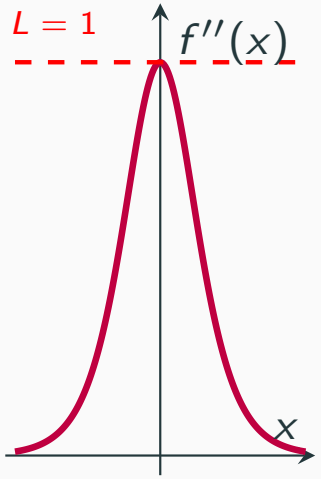
Example: $f(x) = \log(\cosh x)$ (convex, 1-smooth).



Function



Gradient growth



Curvature bound

$$\frac{|f'(x_1) - f'(x_2)|}{|x_1 - x_2|} \leq 1 \quad \Rightarrow \quad |f''(x)| \leq 1$$

whenever f'' exists.

Gradient Lipschitz smoothness (L-smoothness)

Definition (L-smoothness)

Let $f : \mathbb{R}^N \to \overline{\mathbb{R}}$ be proper, convex and differentiable. f is *L-smooth* with $L > 0$ iff:

$$\exists L > 0 : \forall \mathbf{u}_1, \mathbf{u}_2 \in \mathbb{R}^N \quad \|\nabla f(\mathbf{u}_1) - \nabla f(\mathbf{u}_2)\| \leq L\|\mathbf{u}_1 - \mathbf{u}_2\|.$$

Example: For $f_{\mathbf{y}}(\mathbf{u}) = \frac{1}{2}\|\mathbf{A}\mathbf{u} - \mathbf{y}\|^2$, we have $\nabla f_{\mathbf{y}}(\mathbf{u}) = \mathbf{A}^\top(\mathbf{A}\mathbf{u} - \mathbf{y})$.

$$\|\nabla f_{\mathbf{y}}(\mathbf{u}_1) - \nabla f_{\mathbf{y}}(\mathbf{u}_2)\| = \|\mathbf{A}^\top\mathbf{A}(\mathbf{u}_1 - \mathbf{u}_2)\| \leq \|\mathbf{A}^\top\mathbf{A}\|_2 \, \|\mathbf{u}_1 - \mathbf{u}_2\|, \quad L = \|\mathbf{A}^\top\mathbf{A}\|_2 = \sigma_{\max}(\mathbf{A})^2.$$

Gradient Lipschitz smoothness (L-smoothness)

Definition (L-smoothness)

Let $f : \mathbb{R}^N \to \overline{\mathbb{R}}$ be proper, convex and differentiable. f is *L-smooth* with $L > 0$ iff:

$$\exists L > 0 : \forall \mathbf{u}_1, \mathbf{u}_2 \in \mathbb{R}^N \quad \|\nabla f(\mathbf{u}_1) - \nabla f(\mathbf{u}_2)\| \leq L\|\mathbf{u}_1 - \mathbf{u}_2\|.$$

Example: For $f_{\mathbf{y}}(\mathbf{u}) = \frac{1}{2}\|\mathbf{A}\mathbf{u} - \mathbf{y}\|^2$, we have $\nabla f_{\mathbf{y}}(\mathbf{u}) = \mathbf{A}^\top(\mathbf{A}\mathbf{u} - \mathbf{y})$.

$$\|\nabla f_{\mathbf{y}}(\mathbf{u}_1) - \nabla f_{\mathbf{y}}(\mathbf{u}_2)\| = \|\mathbf{A}^\top\mathbf{A}(\mathbf{u}_1 - \mathbf{u}_2)\| \leq \|\mathbf{A}^\top\mathbf{A}\|_2 \, \|\mathbf{u}_1 - \mathbf{u}_2\|, \quad L = \|\mathbf{A}^\top\mathbf{A}\|_2 = \sigma_{\max}(\mathbf{A})^2.$$

Theorem (L-smoothness and convexity)

Let $f : \mathbb{R}^N \to \mathbb{R}$ a convex differentiable function and let $L > 0$. The following statements are equivalent:

1. f is L-smooth
2. $(\forall \mathbf{u}_1, \mathbf{u}_2 \in \mathbb{R}^N) \quad f(\mathbf{u}_2) \leq f(\mathbf{u}_1) + \langle \nabla f(\mathbf{u}_1), \mathbf{u}_2 - \mathbf{u}_1 \rangle + \frac{L}{2}\|\mathbf{u}_1 - \mathbf{u}_2\|^2$
3. $\frac{L}{2}\|\cdot\|^2 - f(\cdot)$ is convex.

Upper/lower quadratic bounds

- f is L-smooth if and only if:

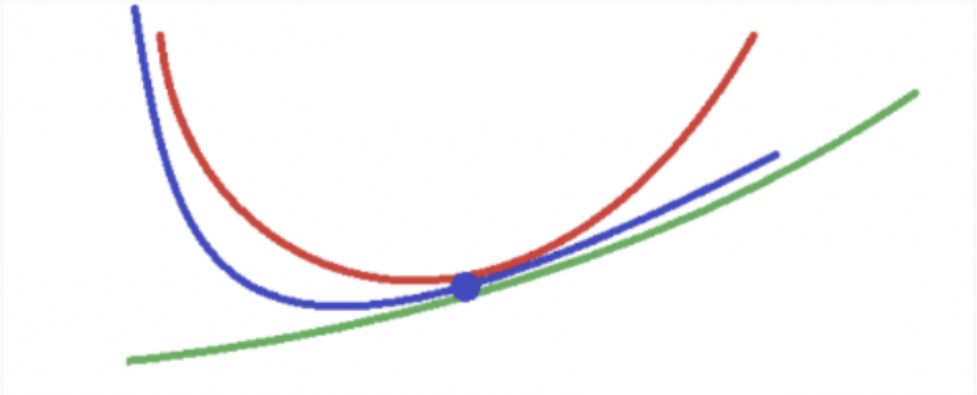
$$(\forall \mathbf{u}_1, \mathbf{u}_2 \in \mathbb{R}^N) \quad f(\mathbf{u}_2) \leq f(\mathbf{u}_1) + \langle \nabla f(\mathbf{u}_1), \mathbf{u}_2 - \mathbf{u}_1 \rangle + \frac{L}{2} \|\mathbf{u}_1 - \mathbf{u}_2\|^2$$

- f is μ-strongly convex if and only if:

$$(\forall \mathbf{u}_1, \mathbf{u}_2 \in \mathbb{R}^N) \quad f(\mathbf{u}_2) \geq f(\mathbf{u}_1) + \langle \nabla f(\mathbf{u}_1), \mathbf{u}_2 - \mathbf{u}_1 \rangle + \frac{\mu}{2} \|\mathbf{u}_1 - \mathbf{u}_2\|^2$$

If f is C^2, then:

$$\mu \mathbf{I} \preceq \nabla^2 f(\mathbf{u}) \preceq L \mathbf{I}, \qquad \forall \mathbf{u} \in \mathbb{R}^N$$



Crash-course on convex optimisation

Gradient descent

Gradient descent algorithm

Ubiquitous for minimising (non-)convex, differentiable functions $f : \mathbb{R}^N \to \overline{\mathbb{R}}$

Algorithm: Gradient Descent (GD)

Input: $\tau > 0, \mathbf{u}^0 \in \mathbb{R}^N$.

for $k \geq 0$ **till** convergence

$$\mathbf{u}^{k+1} = \mathbf{u}^k - \tau \nabla f(\mathbf{u}^k)$$

end

Gradient descent algorithm

Ubiquitous for minimising (non-)convex, differentiable functions $f : \mathbb{R}^N \to \overline{\mathbb{R}}$

Algorithm: Gradient Descent (GD)

Input: $\tau > 0, \mathbf{u}^0 \in \mathbb{R}^N$.

for $k \geq 0$ **till** convergence

$$\mathbf{u}^{k+1} = \mathbf{u}^k - \tau \nabla f(\mathbf{u}^k)$$

end

Theorem (convergence of GD)

Let $f : \mathbb{R}^N \to \overline{\mathbb{R}}$ be proper, convex, L-smooth, $\mathbf{u}^* \in \arg\min f$ and let $(\mathbf{u}^k)_k$ be the sequence of iterates generated by GD. Then, if $\tau \in (0, 2/L)$:

$$f(\mathbf{u}^k) - f(\mathbf{u}^*) \leq \frac{\|\mathbf{u}^0 - \mathbf{u}^*\|^2}{2\tau k} = O\left(\frac{1}{k}\right)$$

Note: the constant is unknown, as $\mathbf{u}^*$ is.

Gradient descent algorithm

Ubiquitous for minimising (non-)convex, differentiable functions $f : \mathbb{R}^N \to \overline{\mathbb{R}}$

Algorithm: Gradient Descent (GD)

Input: $\tau > 0, \mathbf{u}^0 \in \mathbb{R}^N$.

for $k \geq 0$ **till** convergence

$$\mathbf{u}^{k+1} = \mathbf{u}^k - \tau \nabla f(\mathbf{u}^k)$$

end

Theorem (convergence of GD)

Let $f : \mathbb{R}^N \to \overline{\mathbb{R}}$ be proper, convex, L-smooth, $\mathbf{u}^* \in \arg\min f$ and let $(\mathbf{u}^k)_k$ be the sequence of iterates generated by GD. Then, if $\tau \in (0, 2/L)$:

$$f(\mathbf{u}^k) - f(\mathbf{u}^*) \leq \frac{\|\mathbf{u}^0 - \mathbf{u}^*\|^2}{2\tau k} = O\left(\frac{1}{k}\right)$$

Note: the constant is unknown, as $\mathbf{u}^*$ is.

Stopping criteria:

$$\frac{\|\mathbf{u}^{k+1} - \mathbf{u}^k\|}{\|\mathbf{u}^k\| + \delta} \leq \epsilon, \quad \frac{|f(\mathbf{u}^{k+1}) - f(\mathbf{u}^k)|}{f(\mathbf{u}^k) + \delta} \leq \epsilon, \quad \|\nabla f(\mathbf{u}^k)\| \leq \epsilon.$$

Understanding the step-size upper bound in GD

Lemma

For all $k \geq 0$, there holds:

$$\tau \left(1 - \frac{\tau L}{2}\right) \|\nabla f(\mathbf{u}^k)\|^2 \leq f(\mathbf{u}^k) - f(\mathbf{u}^{k+1}).$$

Thus, if $0 < \tau < \frac{2}{L}$, then $f(\mathbf{u}^{k+1}) < f(\mathbf{u}^k)$, the GD algorithm is descending.

Proof. Since $\mathbf{u}^{k+1} - \mathbf{u}^k = -\tau \nabla f(\mathbf{u}^k)$, then by the characterisation of L-smoothness for convex functions we have the descent lemma.

$$f(\mathbf{u}^{k+1}) \leq f(\mathbf{u}^k) - \tau \langle \nabla f(\mathbf{u}^k), \nabla f(\mathbf{u}^k) \rangle + \frac{L}{2} \tau^2 \|\nabla f(\mathbf{u}^k)\|^2.$$

Understanding the step-size upper bound in GD

Lemma

For all $k \geq 0$, there holds:

$$\tau \left(1 - \frac{\tau L}{2}\right) \|\nabla f(\mathbf{u}^k)\|^2 \leq f(\mathbf{u}^k) - f(\mathbf{u}^{k+1}).$$

Thus, if $0 < \tau < \frac{2}{L}$, then $f(\mathbf{u}^{k+1}) < f(\mathbf{u}^k)$, the GD algorithm is descending.

Proof. Since $\mathbf{u}^{k+1} - \mathbf{u}^k = -\tau \nabla f(\mathbf{u}^k)$, then by the characterisation of L-smoothness for convex functions we have the descent lemma.

$$f(\mathbf{u}^{k+1}) \leq f(\mathbf{u}^k) - \tau \langle \nabla f(\mathbf{u}^k), \nabla f(\mathbf{u}^k) \rangle + \frac{L}{2} \tau^2 \|\nabla f(\mathbf{u}^k)\|^2.$$

Beyond slow convergence:

- $O(1/k^2)$ acceleration possible (Nesterov, '83). For $\mathbf{u}^0 = \mathbf{u}^{-1}$, $t_0 = 1$:

$$t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}, \qquad \mathbf{w}^k = \mathbf{u}^k + \frac{t_k - 1}{t_{k+1}} (\mathbf{u}^k - \mathbf{u}^{k-1}) \qquad \mathbf{u}^{k+1} = \mathbf{w}^k - \tau \nabla f(\mathbf{w}^k)$$

- Linear convergence $(1 - \frac{\mu}{L})^k$ for μ-strongly convex functions.

Towards non-smooth optimisation

An implicit perspective: for $\tau > 0$

$$\mathbf{u}^{k+1} = \mathbf{u}^k - \tau \nabla f(\mathbf{u}^{k+1}) \quad \Leftrightarrow \quad \nabla f(\mathbf{u}^{k+1}) + \frac{\mathbf{u}^{k+1} - \mathbf{u}^k}{\tau} = 0,$$

So if $\mathbf{u}^{k+1}$ exists, since f is convex, it is the **unique** minimiser of the strongly convex function:

$$\mathbf{u} \mapsto f(\mathbf{u}) + \frac{\|\mathbf{u}^k - \mathbf{u}\|^2}{2\tau}.$$

A global minimiser of this function may exist also when f is non-smooth!

Towards non-smooth optimisation

An implicit perspective: for $\tau > 0$

$$\mathbf{u}^{k+1} = \mathbf{u}^k - \tau \nabla f(\mathbf{u}^{k+1}) \quad \Leftrightarrow \quad \nabla f(\mathbf{u}^{k+1}) + \frac{\mathbf{u}^{k+1} - \mathbf{u}^k}{\tau} = 0,$$

So if $\mathbf{u}^{k+1}$ exists, since f is convex, it is the **unique** minimiser of the strongly convex function:

$$\mathbf{u} \mapsto f(\mathbf{u}) + \frac{\|\mathbf{u}^k - \mathbf{u}\|^2}{2\tau}.$$

A global minimiser of this function may exist also when f is non-smooth!

. . . non-smoothness encoded via "implicit" gradient updates?

Crash-course on convex optimisation

Non-smooth optimisation

Subgradients and subdifferential

For smooth f, recall that f is convex if and only if:

$$(\textit{above the tangent}) \quad (\forall \mathbf{u}_1, \mathbf{u}_2 \in \mathrm{dom}(f)) \quad f(\mathbf{u}_2) \geq f(\mathbf{u}_1) + \langle \nabla f(\mathbf{u}_1), \mathbf{u}_2 - \mathbf{u}_1 \rangle$$

Subgradients and subdifferential

For smooth f, recall that f is convex if and only if:

$$(\textit{above the tangent}) \quad (\forall \mathbf{u}_1, \mathbf{u}_2 \in \mathrm{dom}(f)) \quad f(\mathbf{u}_2) \geq f(\mathbf{u}_1) + \langle \nabla f(\mathbf{u}_1), \mathbf{u}_2 - \mathbf{u}_1 \rangle$$

Definition (Subgradients and subdifferential)

Let $g : \mathbb{R}^N \to \overline{\mathbb{R}}$ be proper, convex, l.s.c.. Then, a vector $\mathbf{p} \in \mathbb{R}^N$ is a *subgradient* of g at $\mathbf{u}_1 \in \mathrm{dom}(g)$ iff:

$$g(\mathbf{u}_2) \geq g(\mathbf{u}_1) + \langle \mathbf{p}, \mathbf{u}_2 - \mathbf{u}_1 \rangle, \qquad \forall \mathbf{u}_2 \in \mathbb{R}^N.$$

If $\mathbf{u}_1 \notin \mathrm{dom}(g)$, we set $\partial g(\mathbf{u}_1) = \emptyset$. The set of all subgradients at $\mathbf{u}_1 \in \mathbb{R}^N$ is called the *subdifferential* of g in $\mathbf{u}_1$, and it is the denoted by:

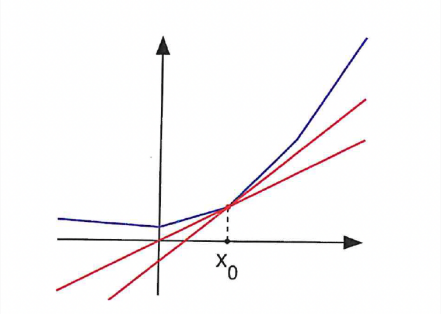
$$\partial g(\mathbf{u}_1) = \left\{ \mathbf{p} \in \mathbb{R}^N : \mathbf{p} \text{ is a subgradient of } g \text{ at } \mathbf{u}_1 \right\}$$

Interpretation:

- $\mathbf{p} \in \partial g(\mathbf{u}_1)$ iff $\phi(\mathbf{u}_2; \mathbf{u}_1) = g(\mathbf{u}_1) + \mathbf{p}^T(\mathbf{u}_2 - \mathbf{u}_1)$ is a lower affine bound for g.
- $\partial g(\mathbf{u}_1)$ collects all the **slopes** of the tangent lines to g through $\mathbf{u}_1$.

Examples

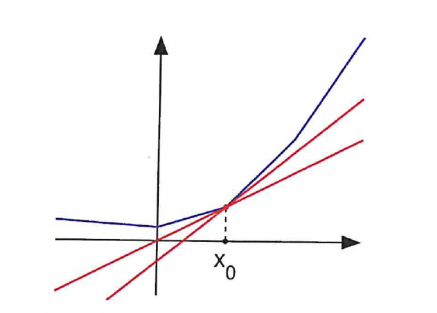
In general, $\partial g(\cdot) : \mathbb{R}^N \to 2^{\mathbb{R}^N}$ is not a singleton



Multiple subgradients at a non-differentiable point x_0.

Examples

In general, $\partial g(\cdot) : \mathbb{R}^N \to 2^{\mathbb{R}^N}$ is not a singleton



Multiple subgradients at a non-differentiable point x_0.

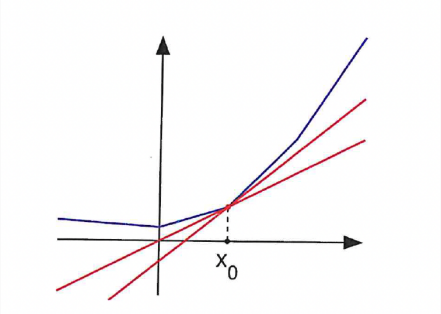
Proposition (subdifferential at differentiable points)

If $g : \mathbb{R}^N \to \overline{\mathbb{R}}$ is proper, convex and differentiable in $\mathbf{u} \in \mathrm{dom}(g)$, then:

$$\partial g(\mathbf{u}) = \{\nabla g(\mathbf{u})\}.$$

Examples

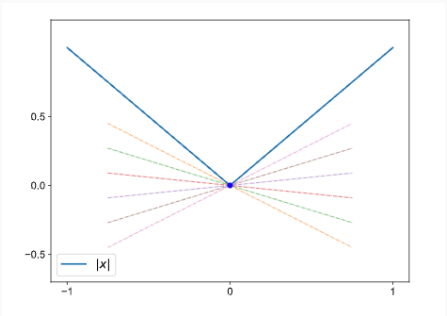
In general, $\partial g(\cdot) : \mathbb{R}^N \to 2^{\mathbb{R}^N}$ is not a singleton

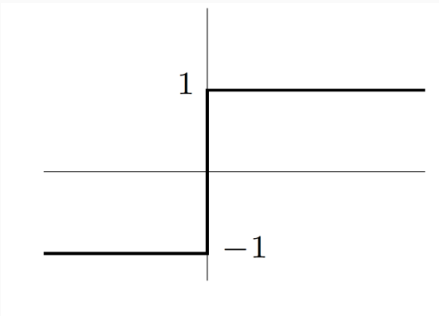


Multiple subgradients at a non-differentiable point x_0.

Example: $g : \mathbb{R} \to \overline{\mathbb{R}}$, $g(x) = |x|$.

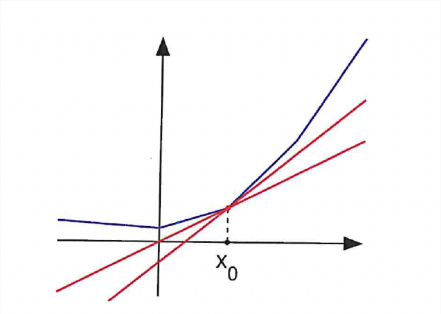
$$\partial g(x) = \begin{cases} \{1\} & \text{if} \quad x > 0 \\ \{-1\} & \text{if} \quad x < 0 \\ [-1, 1] & \text{if} \quad x = 0. \end{cases}$$





Examples

In general, $\partial g(\cdot) : \mathbb{R}^N \to 2^{\mathbb{R}^N}$ is not a singleton



Multiple subgradients at a non-differentiable point x_0.

Indicator of $[a, b] \subset \mathbb{R}$: $g(x) = \iota_{[a,b]}(x)$

$$\partial g(x) = \begin{cases} \{0\} & \text{if } a < x < b, \\ (-\infty, 0] & \text{if } x = a, \\ [0, +\infty) & \text{if } x = b, \\ \emptyset & \text{otherwise.} \end{cases}$$

Normal cone to $[a, b]$

ReLU: $g(x) = \max(0, x)$.

$$\partial g(x) = \begin{cases} \{0\} & \text{if } x < 0, \\ [0, 1] & \text{if } x = 0, \\ \{1\} & \text{if } x > 0. \end{cases}$$

Subdifferential calculus rules: separable functions

Often, n-dimensional functions can be nicely expressed as the sum of 1D components:

- **norms** $p \geq 1$, $\|\mathbf{u}\|_p^p, = \sum_{i=1}^N |u_i|^p \ldots$
- **sum of norms**, e.g. $g(\mathbf{u}) = \|\mathbf{u}\|_1 + \frac{\lambda}{2}\|\mathbf{u}\|_2^2 = \sum_{i=1}^N \left(|u_i| + \lambda |u_i|^2\right)$.
- $\ldots$

Subdifferential calculus rules: separable functions

Often, n-dimensional functions can be nicely expressed as the sum of 1D components:

- **norms** $p \geq 1$, $\|\mathbf{u}\|_p^p, = \sum_{i=1}^N |u_i|^p \ldots$
- **sum of norms**, e.g. $g(\mathbf{u}) = \|\mathbf{u}\|_1 + \frac{\lambda}{2}\|\mathbf{u}\|_2^2 = \sum_{i=1}^N \left(|u_i| + \lambda |u_i|^2\right)$.
- $\ldots$

Definition (separable function)

Let $g : \mathbb{R}^N \to \overline{\mathbb{R}}$ be proper. We say that g is *separable* if there exist proper functions $g_i : \mathbb{R} \to \overline{\mathbb{R}}$ such that

$$g(\mathbf{u}) = \sum_{i=1}^N g_i(u_i), \qquad \forall \mathbf{u} \in \mathbb{R}^N.$$

Proposition (subdifferential of separable functions)

Let $g : \mathbb{R}^N \to \overline{\mathbb{R}}$ be proper ,convex, l.s.c. and separable in term of proper, convex, ls.c. functions $g_i : \mathbb{R} \to \overline{\mathbb{R}}$. Then, for all $\mathbf{u} \in \mathrm{dom}(g)$:

$$\partial g(\mathbf{u}) = (\partial g_i(u_i))_{i=1}^N = (\partial g_1(u_1)) \times \ldots \times (\partial g_N(u_N)).$$

Optimality conditions

Analogous to Fermat's rule in non-smooth case.

Theorem (optimality conditions in non-smooth, convex case)

Let $g : \mathbb{R}^N \to \overline{\mathbb{R}}$ be proper, convex and l.s.c. Then:

$$\mathbf{u}^* \in \operatorname*{arg\,min}_{\mathbf{u} \in \mathbb{R}^N} g(\mathbf{u}) \qquad \Longleftrightarrow \qquad \mathbf{0} \in \partial g(\mathbf{u}^*).$$

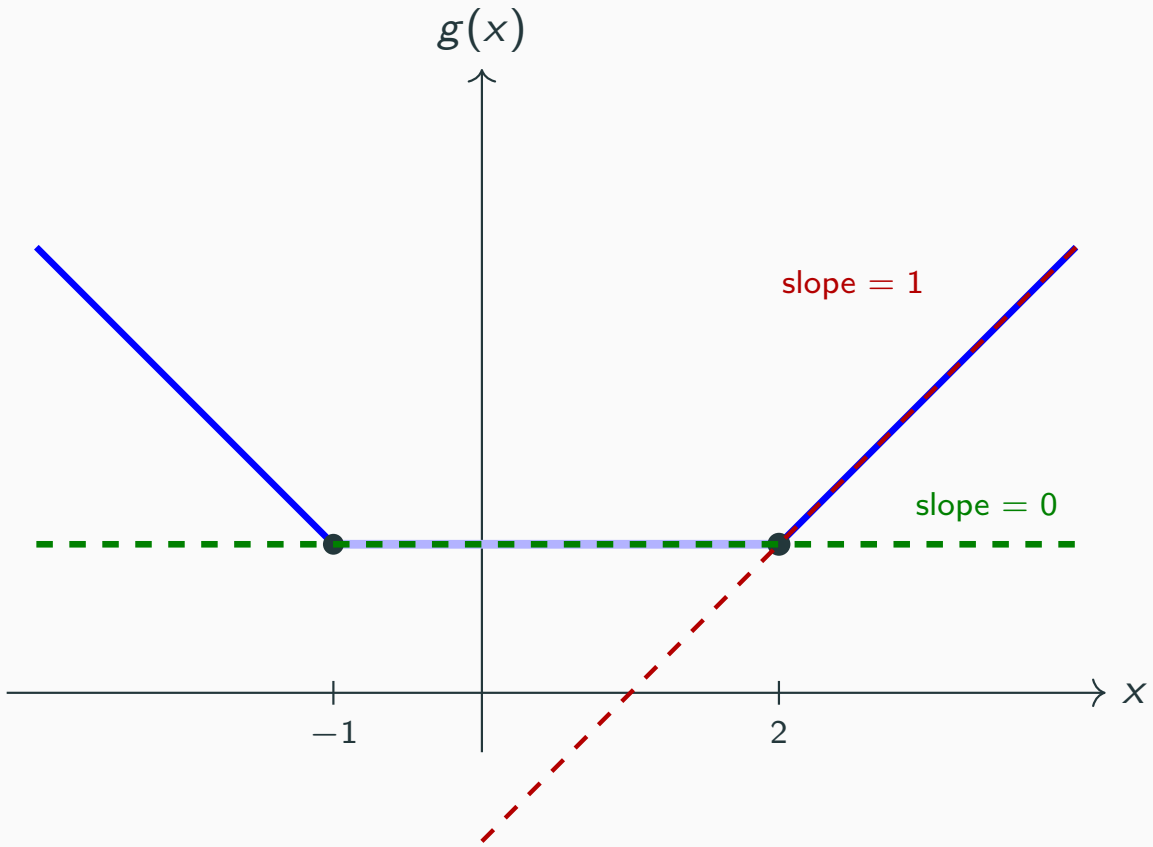
Interpretation:

- If the vector $\mathbf{0} \in \mathbb{R}^N$ belongs to $\partial g(\mathbf{u}^*)$ ("flat tangent"), then $\mathbf{u}^*$ is a minimiser.
- If g is differentiable, the result reads $\mathbf{0} = \nabla g(\mathbf{u}^*)$ (Fermat's rule).

Graphical representation

Example: $g(x) = 1 + \max\{0,\ x - 2,\ -1 - x\}$.

$$\partial g(2) = [0, 1] \ni 0 \quad \Rightarrow \quad x^* = 2 \in \arg\min g = [-1, 2]$$



Crash-course on convex optimisation

Proximal operator

The proximal operator: definition

Crucial tool for the development of non-smooth optimisation algorithms. Relations with activation functions in the context of deep networks (Combettes, Pesquet, '20).

Definition

Let $g : \mathbb{R}^N \to \overline{\mathbb{R}}$ be proper, and l.s.c. Then, the *proximal operator* of g with parameter $\gamma > 0$ is defined as the **multi-valued map** $\text{prox}_{\gamma g} : \mathbb{R}^N \to 2^{\mathbb{R}^N}$:

$$\text{prox}_{\gamma g}(\mathbf{u}) := \underset{\mathbf{z} \in \mathbb{R}^N}{\arg\min} \underbrace{g(\mathbf{z}) + \frac{1}{2\gamma}\|\mathbf{z} - \mathbf{u}\|^2}_{=:h(\mathbf{z};\mathbf{u})}$$

The proximal operator: definition

Crucial tool for the development of non-smooth optimisation algorithms. Relations with activation functions in the context of deep networks (Combettes, Pesquet, '20).

Definition

Let $g : \mathbb{R}^N \to \overline{\mathbb{R}}$ be proper, and l.s.c. Then, the *proximal operator* of g with parameter $\gamma > 0$ is defined as the **multi-valued map** $\mathrm{prox}_{\gamma g} : \mathbb{R}^N \to 2^{\mathbb{R}^N}$:

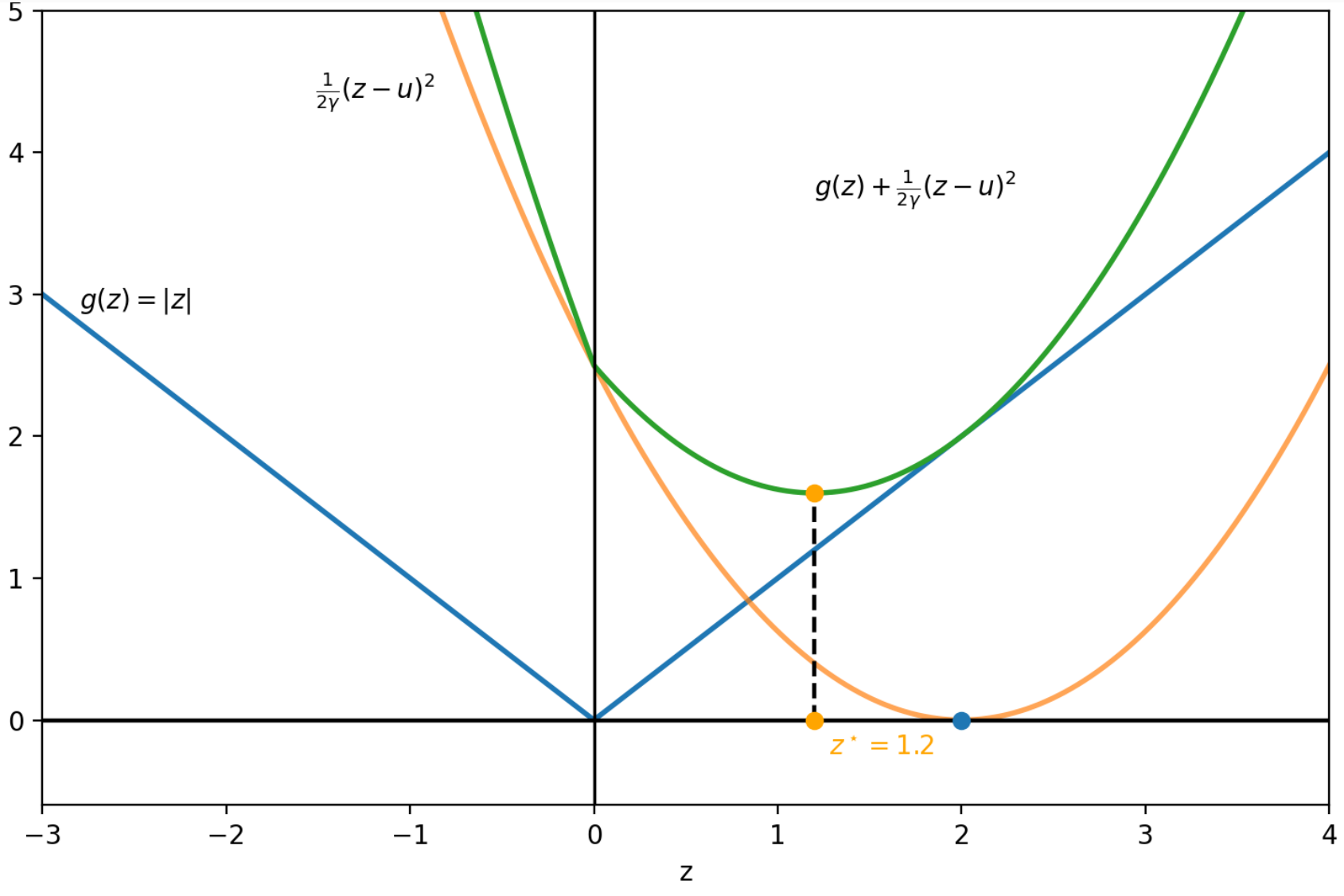
$$\mathrm{prox}_{\gamma g}(\mathbf{u}) := \underset{\mathbf{z}\in\mathbb{R}^N}{\arg\min} \underbrace{g(\mathbf{z}) + \frac{1}{2\gamma}\|\mathbf{z}-\mathbf{u}\|^2}_{=:h(\mathbf{z};\mathbf{u})}$$

Proposition (uniqueness for convex functions)

If $g : \mathbb{R}^N \to \overline{\mathbb{R}}$ is proper, l.s.c., convex, then $\mathrm{prox}_{\gamma g}(\mathbf{u})$ exists and is unique $\forall \mathbf{u} \in \mathbb{R}^N$.

"*Proof*": For all $\mathbf{u} \in \mathbb{R}^N$, the function $h(\cdot;\mathbf{u})$ is $\frac{1}{\gamma}$-strongly (hence strictly) convex, hence it admits a unique minimiser.

Graphical interpretation



Geometrical interpretation of $\text{prox}_{\gamma|\cdot|}(u)$ for $u = 2, \gamma = 0.8$.

Examples: indicator function

Example: Let $C \subset \mathbb{R}^N$ be a closed and convex set. Recall:

$$\iota_C(\mathbf{u}) := \begin{cases} 0 & \text{if } \mathbf{u} \in C \\ +\infty & \text{if } \mathbf{u} \notin C \end{cases}$$

The function $\iota_C(\mathbf{u})$ is proper, convex and l.s.c.

$$\text{prox}_{\gamma\iota_C}(\mathbf{u}) = \underset{\mathbf{y}\in\mathbb{R}^N}{\arg\min}\ \iota_C(\mathbf{y}) + \frac{1}{2\gamma}\|\mathbf{y}-\mathbf{u}\|^2 = \underset{\mathbf{y}\in C}{\arg\min}\ \frac{1}{2\gamma}\|\mathbf{y}-\mathbf{u}\|^2 = \Pi_C(\mathbf{u}),$$

i.e. the (orthogonal) **projection** of $\mathbf{u}$ onto C (the closest point $\mathbf{y}^* \in C$ to $\mathbf{u}$).

Remark: The prox operator is often referred to as *generalised projection*.

Examples: prox of the ℓ_1 norm

Example: Let $g(x) = |u|$ and $\gamma > 0$:

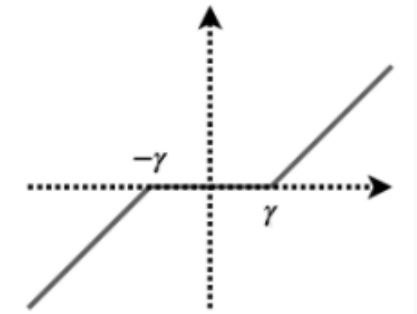
$$w = \text{prox}_{\gamma g}(u) = \underset{y \in \mathbb{R}}{\arg\min} \ |y| + \frac{1}{2\gamma}(y-u)^2$$

By optimality:

$$\gamma p + w - u = 0, \quad p \in \partial|w| \quad \Leftrightarrow \quad w = u - \gamma p, \quad p \in \partial|w|$$

Recalling the expression of $\partial|\cdot|$, one finds the *soft-thresholding* function

$$w = \text{prox}_{\gamma g}(u) = \begin{cases} u-\gamma & \text{if} \quad u > \gamma \\ u+\gamma & \text{if} \quad u < -\gamma \\ 0 & \text{if} \quad -\gamma \leq u \leq \gamma \end{cases} = \mathcal{T}_\gamma(u) := \text{sign}(u)\max\{|u|-\gamma, 0\}$$



Computation of proximal points: properties

Proposition (prox of separable functions)

Let $g : \mathbb{R}^N \to \overline{\mathbb{R}}$ be proper, l.s.c., convex and separable, i.e. $g(\mathbf{u}) = \sum_{i=1}^N g_i(u_i)$ for uni-variate proper, l.s.c. and convex functions $g_i : \mathbb{R} \to \overline{\mathbb{R}}$. Then for $\gamma > 0$

$$\text{prox}_{\gamma g}(\mathbf{u}) = \left(\text{prox}_{\gamma g_1}(u_1), \ldots, \text{prox}_{\gamma g_N}(u_N)\right),$$

Example: $g(\mathbf{u}) = \lambda\|\mathbf{u}\|_1$, then $\text{prox}_{\lambda\|\cdot\|_1}(\mathbf{u}) = (\mathcal{T}_\lambda(u_1), \ldots, \mathcal{T}_\lambda(u_N))$.

Proposition (prox of rescaled and perturbed functions)

Let $g : \mathbb{R}^N \to \overline{\mathbb{R}}$ be proper, l.s.c. and convex, $\lambda \neq 0$ and $\gamma > 0$:

- $h_1(\mathbf{u}) := \lambda g(\mathbf{u}/\lambda)$.
$$\text{prox}_{\gamma h_1}(\mathbf{u}) = \lambda \text{prox}_{\frac{\gamma}{\lambda} g}(\mathbf{u}/\lambda).$$
- $h_2(\mathbf{u}) := \alpha g(\mathbf{u}) + \frac{\beta}{2}\|\mathbf{u}\|^2$, for $\alpha, \beta \in \mathbb{R}_{++}$.
$$\text{prox}_{\gamma h_2}(\mathbf{u}) = \text{prox}_{\frac{\alpha\gamma}{1+\beta\gamma} g}\left(\frac{\mathbf{u}}{1+\beta\gamma}\right).$$
- $h_3(\mathbf{u}) := g(\mathbf{W}\mathbf{u})$ where $\mathbf{W} \in \mathbb{R}^{d\times N}$ is s.t. $\mathbf{W}^T\mathbf{W} = \mathbf{W}\mathbf{W}^\top = \mathbf{I} \in \mathbb{R}^{N\times N}$.
$$\text{prox}_{\gamma h_3}(\mathbf{u}) = \mathbf{W}^\top \text{prox}_{\gamma g}(\mathbf{W}\mathbf{u}).$$

Computation of proximal points

Why?

- Model-based regularisation function are typically invoked in terms of their proximal operators:

$$\text{prox}_{\eta g}(\cdot) = \arg\min \; g(\cdot) + \ldots$$

- Often, no closed-form expression: computationally expensive.

Examples of easily-proximable function, see, e.g.:

- Beck, *First-order methods in optimization* 2006 (Chapter 6)
- Parikh, Boyd, *Proximal algorithms*, 2013
- `http://proximity-operator.net/index.html`
  $\rightarrow$ plenty of examples in MATLAB/Python.

Proximal algorithms

Proximal algorithms

Proximal-gradient algorithm

Projected gradient descent

For L-smooth and convex $f : \mathbb{R}^N \to \overline{\mathbb{R}}$, closed and convex $C \subset \mathbb{R}^N$:

$$\arg\min_{\mathbf{u}\in C} \; f(\mathbf{u}) = \arg\min_{\mathbf{u}\in\mathbb{R}^N} \; f(\mathbf{u}) + \iota_C(\mathbf{u})$$

Algorithm: Projected gradient descent algorithm

Input: $\tau \in \left(0, \frac{2}{L}\right), \mathbf{u}^0 \in \mathbb{R}^N$.

for $k \geq 0$ **till** convergence

$$\mathbf{u}^{k+\frac{1}{2}} = \mathbf{u}^k - \tau \nabla f(\mathbf{u}^k)$$

$$\mathbf{u}^{k+1} = P_C(\mathbf{u}^{k+\frac{1}{2}}) = \arg\min_{\mathbf{y}\in C} \; \frac{1}{2}\|\mathbf{y} - \mathbf{u}^{k+\frac{1}{2}}\|^2$$

$$= \arg\min_{\mathbf{y}\in\mathbb{R}^N} \; \iota_C(\mathbf{y}) + \frac{1}{2}\|\mathbf{y} - \mathbf{u}^{k+\frac{1}{2}}\|^2 = \mathrm{prox}_{\iota_C}(\mathbf{u}^{k+\frac{1}{2}})$$

end

Idea: deal explicitly with smooth part and implicitly with convex constraints.

Proximal gradient algorithm/forward backward splitting

$$\underset{\mathbf{u}\in\mathbb{R}^N}{\arg\min} \quad \{F(\mathbf{u}) := f(\mathbf{u}) + g(\mathbf{u})\},$$

- $f : \mathbb{R}^N \to \overline{\mathbb{R}}$ is proper, convex and L-smooth
- $g : \mathbb{R}^N \to \overline{\mathbb{R}}$ is proper, l.s.c., convex and (possibly) non-smooth

---

Algorithm: Proximal gradient/forward-backward splitting algorithm[1]

Input: $\mathbf{u}^0 \in \mathbb{R}^N$, $\tau \in \left(0, \frac{2}{L}\right)$.
for $k \geq 0$ **do**

$$\mathbf{u}^{k+1} = \text{prox}_{\tau g}\left(\mathbf{u}^k - \tau \nabla f(\mathbf{u}^k)\right)$$

end for

---

[1] Combettes, Wajs, 2005, Combettes, Pesquet, 2007

Use in inverse problems

$$\arg\min_{\mathbf{u}\in\mathbb{R}^n} \; f_{\mathbf{y}}(\mathbf{u}) + g(\mathbf{u})$$

- If $g \equiv 0$: smooth-optimisation problem $\rightarrow$ GD.
- If $g(\mathbf{u}) = \iota_C(\mathbf{u})$ for closed and convex $C \rightarrow$ Proj-GD.
- If $g(\mathbf{u}) = \lambda\|\mathbf{W}\mathbf{u}\|_1$ for $\lambda > 0$ and semi-orthogonal $\mathbf{W} \in \mathbb{R}^{d\times N}$ (e.g., Wavelet . . .)

$$\min_{\mathbf{u}\in\mathbb{R}^N} \; \frac{1}{2}\|\mathbf{A}\mathbf{u} - \mathbf{y}\|^2 + \lambda\|\mathbf{W}\mathbf{u}\|_1,$$

Iterative Soft-Thresholding Algorithm (ISTA) (Daubechies, Defrise, De Mol, '04).

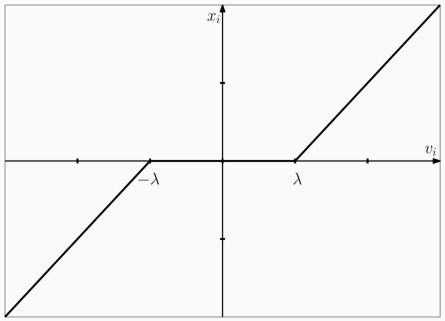
Iterative Soft Thresholding Algorithm (ISTA)

Proximal gradient iterations take the form:

$$\mathbf{u}^{k+1} = \mathbf{W}^\top \mathcal{T}_{\tau\lambda}(\mathbf{W}\mathbf{u}^k - \tau\mathbf{W}\mathbf{A}^\top(\mathbf{A}\mathbf{u}^k - \mathbf{y})),$$

where $\mathcal{T}_{\tau\lambda}(\cdot)$ is the *soft-thresholding* operator:

$$\mathcal{T}_{\tau\lambda}(\mathbf{z}) = \Big([|z_j| - \lambda\tau]_+ \operatorname{sign}(z_j)\Big)_{j=1,\dots,d}$$



$\tau = 1$

Convergence of FB iterations

Theorem (convergence of Proximal gradient descent)

Let $\mathbf{u}^* \in \arg\min(F = f + g)$ and $(\mathbf{u}^k)_k$ be the sequence of proximal gradient iterates. Then, if $\tau \in (0, 2/L)$, there holds:

$$F(\mathbf{u}^k) - F(\mathbf{u}^*) \leq \frac{\|\mathbf{u}^0 - \mathbf{u}^*\|^2}{2\tau k}.$$

- Same convergence speed as GD!
- The regularisation function g is invoked only in terms of its prox operator
- Acceleration (FISTA, Nesterov, '83, Beck, Teboulle, '05), $\mathbf{u}^0 = \mathbf{u}^{-1}$, $t_0 = 1$:

$$t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}, \qquad \mathbf{w}^k = \mathbf{u}^k + \frac{t_k - 1}{t_{k+1}}(\mathbf{u}^k - \mathbf{u}^{k-1}) \qquad \mathbf{y}^{k+1} = \mathrm{prox}_{\tau g}(\mathbf{w}^k - \tau \nabla f(\mathbf{w}^k))$$

to improve convergence speed in function values:

$$F(\mathbf{u}^k) - F(\mathbf{u}^*) \leq \frac{C(\mathbf{u}^0, \tau, \mathbf{u}^*)}{(k+1)^2}$$

with unknown constant $C > 0$.

Proximal algorithms

Primal-dual algorithm

Convex conjugation and Fenchel-Rockafellar formulation

Consider problems written in primal form as:

$$\min_{\mathbf{u}\in\mathbb{R}^n} \quad f_{\mathbf{y}}(\mathbf{u}) + g\big(\mathbf{K}\mathbf{u}\big)$$

where $\mathbf{K} \in \mathbb{R}^{d\times N}$ is linear (examples: $\lambda\|\mathbf{D}\mathbf{u}\|_{2,1}$ for TV, $\lambda\|\mathbf{W}\mathbf{u}\|_1$ for wavelets...): **proximal operator may be not explicit** ($\mathbf{K}$ not semi-orthogonal)!

Alternatives to PGD?

Convex conjugation and Fenchel-Rockafellar formulation

Consider problems written in primal form as:

$$\min_{\mathbf{u}\in\mathbb{R}^n} \quad f_{\mathbf{y}}(\mathbf{u}) + g\big(\mathbf{K}\mathbf{u}\big)$$

where $\mathbf{K} \in \mathbb{R}^{d\times N}$ is linear (examples: $\lambda\|\mathbf{D}\mathbf{u}\|_{2,1}$ for TV, $\lambda\|\mathbf{W}\mathbf{u}\|_1$ for wavelets...): **proximal operator may be not explicit** ($\mathbf{K}$ not semi-orthogonal)!

Alternatives to PGD?

- Dual formulation, $g^* : \mathbb{R}^d \to \overline{\mathbb{R}}$:

$$g^*(\mathbf{q}) = \sup_{\mathbf{v}\in\mathbb{R}^d} \ \mathbf{q}^\top\mathbf{v} - g(\mathbf{v}) \qquad \text{(often simpler)}$$

- Inexact solvers: solve $\mathrm{prox}_{\tau g}(\mathbf{z}) = \arg\min_{\mathbf{w}} \ldots$ for some iterations/up to some tolerance (Villa, Salzo, Baldassarre, Verri, '13).
- Smoothing, e.g.:

$$\|\mathbf{D}\mathbf{u}\|_{2,1,\xi} = \sum_i \|(\mathbf{D}\mathbf{u})_i\|_{2,\xi} = \sum_i \sqrt{(\mathbf{D}_h\mathbf{u})_i^2 + (\mathbf{D}_v\mathbf{u})_i^2 + \xi}.$$

Chambolle–Pock algorithm (no extrapolation)

Choose step-sizes $\tau > 0, \eta > 0$ and extrapolation parameter $\theta \in [0, 1]$.

Initialize $\mathbf{u}^0, \mathbf{v}^0$. Iterate (Chambolle, Pock, '11):

$$\boxed{\begin{aligned} \mathbf{v}^{k+1} &= \mathrm{prox}_{\eta g^*}(\mathbf{v}^k + \eta \mathbf{K}\mathbf{u}^k), \\ \mathbf{u}^{k+1} &= \mathrm{prox}_{\tau f_{\mathbf{y}}}\left(\mathbf{u}^k - \tau \mathbf{K}^\top \mathbf{v}^{k+1}\right). \end{aligned}}$$

Sufficient convergence condition:

$$\tau \, \eta \, \|\mathbf{K}\|^2 < 1.$$

(In practice, $\|\mathbf{K}\|$ can be estimated by a few power iterations.)

Chambolle–Pock algorithm (no extrapolation)

Choose step-sizes $\tau > 0, \eta > 0$ and extrapolation parameter $\theta \in [0, 1]$.

Initialize $\mathbf{u}^0, \mathbf{v}^0$. Iterate (Chambolle, Pock, '11):

$$\boxed{\begin{aligned} \mathbf{v}^{k+1} &= \text{prox}_{\eta g^*}(\mathbf{v}^k + \eta \mathbf{K}\mathbf{u}^k), \\ \mathbf{u}^{k+1} &= \text{prox}_{\tau f_{\mathbf{y}}}\left(\mathbf{u}^k - \tau \mathbf{K}^\top \mathbf{v}^{k+1}\right). \end{aligned}}$$

Sufficient convergence condition:

$$\tau \, \eta \, \|\mathbf{K}\|^2 < 1.$$

(In practice, $\|\mathbf{K}\|$ can be estimated by a few power iterations.)

- **TV example:**

$$\boxed{\lambda \|\mathbf{D}\mathbf{u}\|_{2,1} \;\rightarrow\; g^*(\mathbf{v}) = \iota_{\{\|\mathbf{v}\|_{2,\infty} \le \lambda\}} \;\Rightarrow\; \mathbf{v}^{k+1} = \text{proj}_{\{\|\cdot\|_{2,\infty} \le \lambda\}}\left(\mathbf{v}^k + \eta \mathbf{D}\mathbf{u}^k\right).}$$

- **Connections:** CP is effective when primal and dual prox are both cheap.

Chambolle–Pock algorithm (no extrapolation)

Choose step-sizes $\tau > 0, \eta > 0$ and extrapolation parameter $\theta \in [0, 1]$.

Initialize $\mathbf{u}^0, \mathbf{v}^0$. Iterate (Chambolle, Pock, '11):

$$\mathbf{v}^{k+1} = \operatorname{prox}_{\eta g^*}(\mathbf{v}^k + \eta \mathbf{K}\mathbf{u}^k) = \mathbf{v}^k + \eta \mathbf{K}\mathbf{u}^k - \eta \operatorname{prox}_{g/\eta}\left(\frac{\mathbf{v}^k}{\eta} + \mathbf{K}\mathbf{u}^k\right),$$

$$\mathbf{u}^{k+1} = \operatorname{prox}_{\tau f_{\mathbf{y}}}\left(\mathbf{u}^k - \tau \mathbf{K}^\top \mathbf{v}^{k+1}\right).$$

Sufficient convergence condition:

$$\tau \, \eta \, \|\mathbf{K}\|^2 < 1.$$

(In practice, $\|\mathbf{K}\|$ can be estimated by a few power iterations.)

- **TV example:**

$$\lambda \|\mathbf{D}\mathbf{u}\|_{2,1} \;\rightarrow\; g^*(\mathbf{v}) = \iota_{\{\|\mathbf{v}\|_{2,\infty} \le \lambda\}} \;\Rightarrow\; \mathbf{v}^{k+1} = \operatorname{proj}_{\{\|\cdot\|_{2,\infty} \le \lambda\}}\left(\mathbf{v}^k + \eta \mathbf{D}\mathbf{u}^k\right).$$

- **Connections:** CP is effective when primal and dual prox are both cheap.
- **Dual prox from primal prox (Moreau identity).** For any convex h and $\sigma > 0$:

$$\operatorname{prox}_{\eta h^*}(\mathbf{w}) \;=\; \mathbf{w} - \eta \, \operatorname{prox}_{h/\eta}\left(\frac{\mathbf{w}}{\eta}\right).$$

Short worked example (TV denoising)

TV image denoising (Rudin, Osher, Fatemi, '92)

$$\min_{\mathbf{u}} \ \tfrac{1}{2}\|\mathbf{u}-\mathbf{y}\|^2 + \lambda\|\mathbf{D}\mathbf{u}\|_{2,1}$$

Then CP updates become

$$\mathbf{v}^{k+1} = \operatorname{proj}_{\{\|\cdot\|_{2,\infty}\leq\lambda\}}\left(\mathbf{v}^k + \eta\mathbf{D}\mathbf{u}^k\right) \quad \text{(pointwise vector clipping)},$$

$$\mathbf{u}^{k+1} = \frac{\mathbf{u}^k - \tau\mathbf{D}^\top\mathbf{v}^{k+1} + \tau\mathbf{y}}{1+\tau} \quad \text{(explicit prox of data term)}$$

Both steps are inexpensive. Explains CP's popularity for TV-type problems against running iterative solvers/smoothing strategies.

Extensive use in the imaging community: see Chambolle, Pock, '16 for a review, Bredies, Kunisch, Pock, '10 for TGV, Chambolle Erhardt, Richtarik, Schönlieb, '18 for stochastic updates...

Which algorithm should we use?

Framework: Smooth/convex f, non-smooth/convex g, closed/convex C.

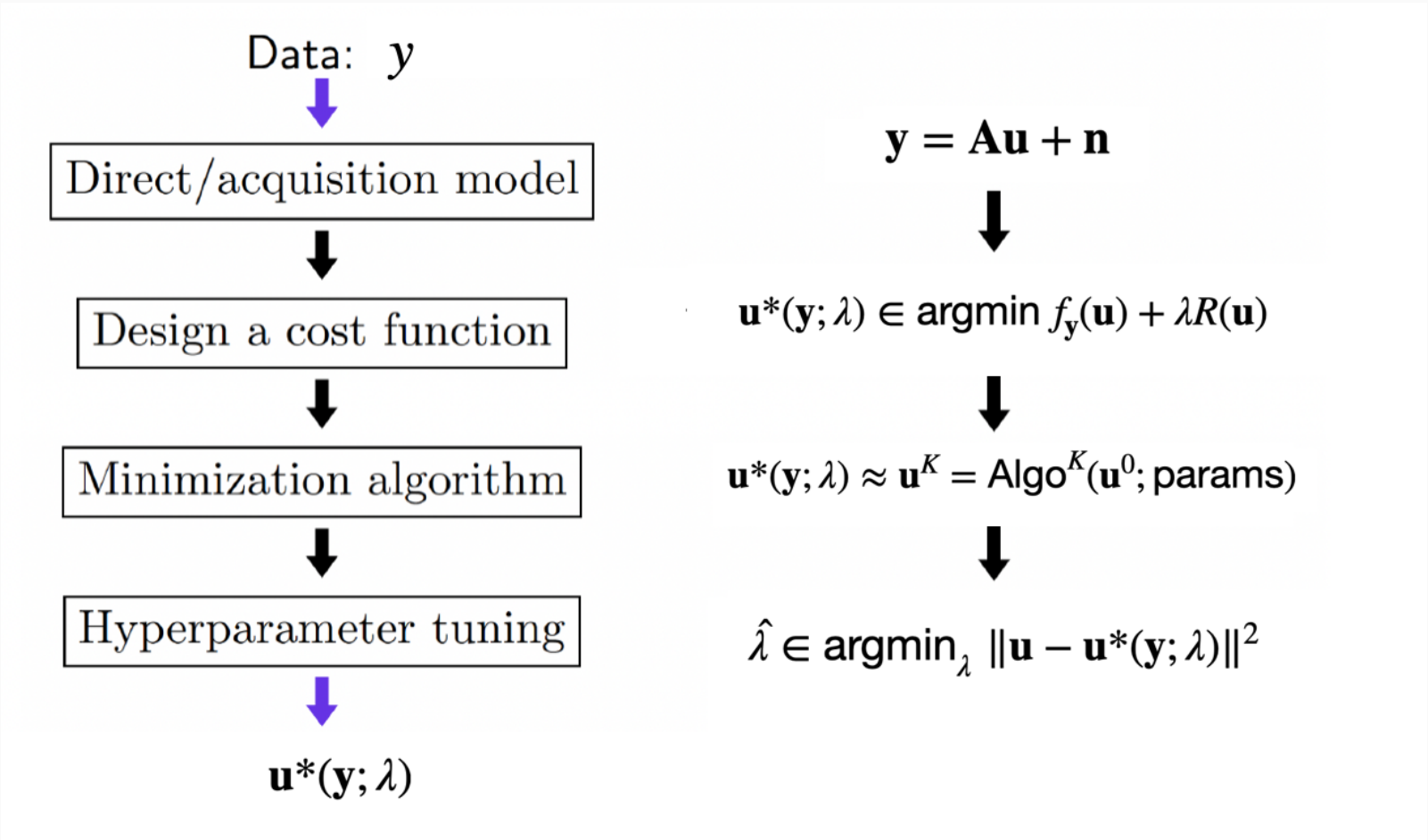
$$\min_{\mathbf{u}\in C} \; f(\mathbf{u}) + g(\mathbf{u})$$

Algorithm	**Smooth f**	**Constraints**	**prox**	**Handles $g(\mathbf{Ku})$**
GD	✓	× ($C = \mathbb{R}^N$)	× ($g \equiv 0$)	×
Proj. GD	✓	✓	Π_C (orth. proj.)	×
PGD	✓	✓(add ι_C)	✓(need $\text{prox}_{\iota_C + g}$)*	only if easy
Primal-Dual	✓(not required)	✓(add ι_C)	$\text{prox}_{g^*} + \text{prox}_f$	✓

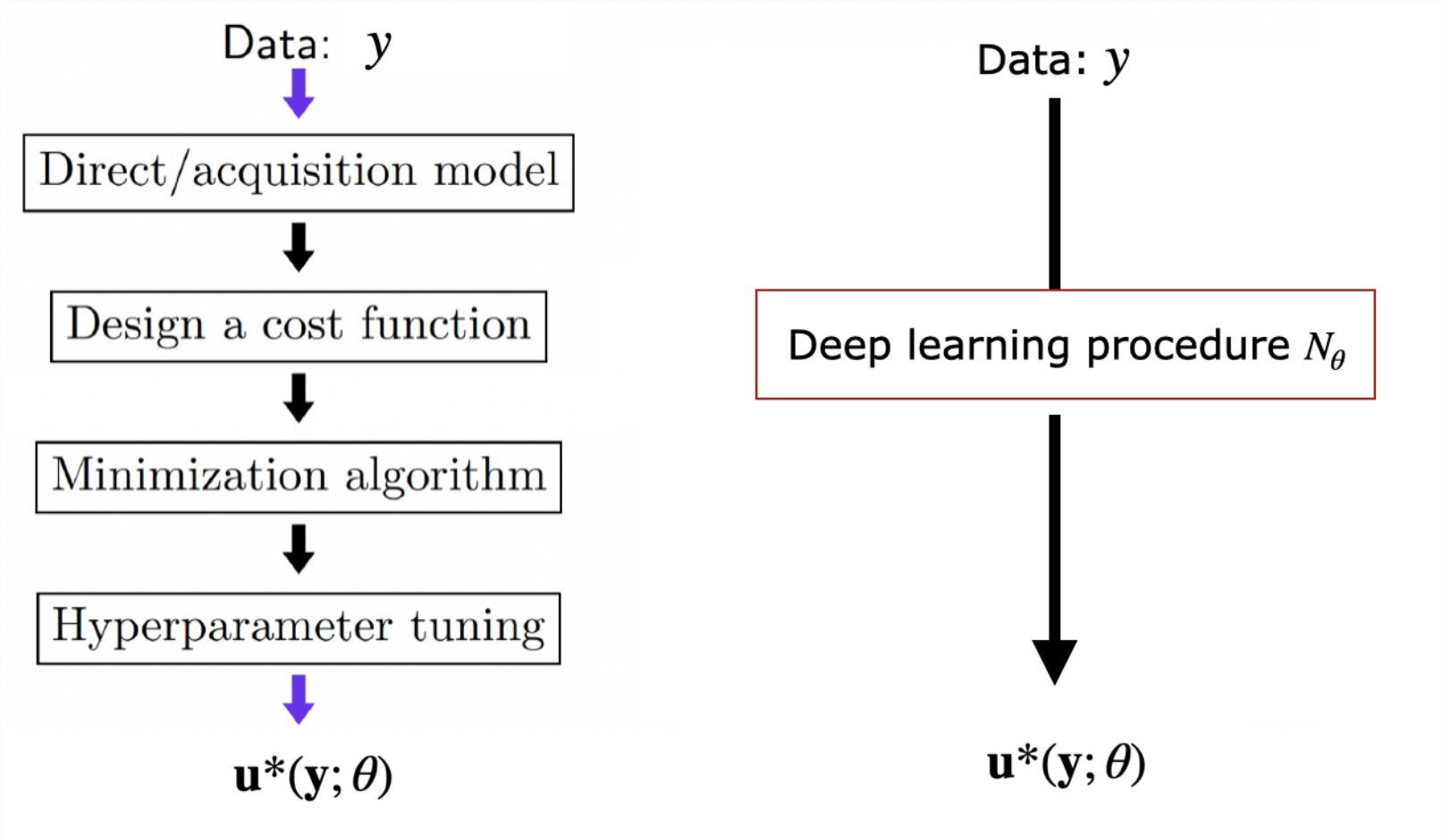
*: $\text{prox}_{f_1+f_2} \neq \text{prox}_{f_2} \circ \text{prox}_{f_2}$ in general, but true under specific assumptions.

Optimisation-driven deep learning

Towards optimisation-driven deep learning

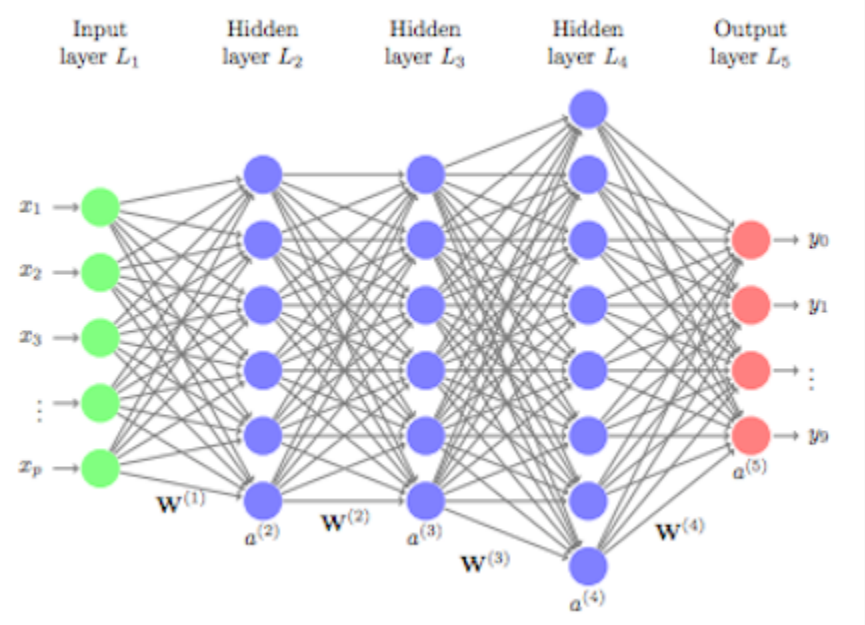


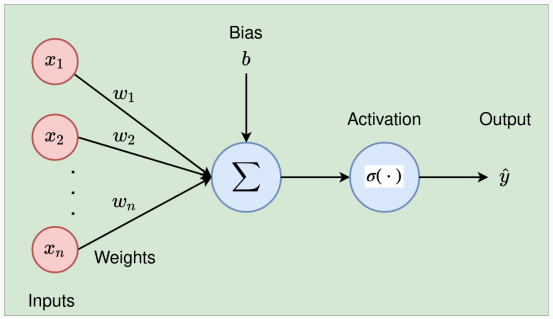
Towards optimisation-driven deep learning



Review on neural networks

Neural networks are overparametrised functions $N_{\boldsymbol{\theta}} : \mathbb{R}^d \to \mathbb{R}^s$ where $\boldsymbol{\theta} \in \Theta$ with $\Theta \subset \mathbb{R}^S$, $S \gg 1$. They are defined as composition of layers:





$$N_\theta = N^L_{\boldsymbol{\theta}_L} \circ \ldots \circ N^2_{\boldsymbol{\theta}_2} \circ N^1_{\boldsymbol{\theta}_1}, \quad \boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_L\}$$

Typical structure of a layer:

$$N^i_{\boldsymbol{\theta}_i}(\mathbf{z}) = \sigma\left(\mathbf{W_i z} + \mathbf{b_i}\right), \qquad \boldsymbol{\theta}_i = \{\mathbf{W_i}, \mathbf{b_i}\}$$

where $\mathbf{W_i} \in \mathbb{R}^{n_i^1 \times n_i^2}$, $\mathbf{b_i} \in \mathbb{R}^{n_i^1}$ and $\sigma : \mathbb{R}^{n_i^1} \to \mathbb{R}^{n_i^1}$ is a non-linear function.

Training a network

Given a dataset of clean/corrupted data $\left\{(\mathbf{u}_j, \mathbf{y}_j)\right\}_{j=1,\dots,J}$, **training** means:

find $\quad \boldsymbol{\theta} = (\mathbf{W_1}, \mathbf{b_1}, \dots, \mathbf{W_L}, \mathbf{b_L}) \quad$ s.t. $\quad N_{\boldsymbol{\theta}}(\mathbf{y}_i) = N^L_{\mathbf{W_L},\mathbf{b_L}} \circ \dots \circ N^2_{\mathbf{W_2},\mathbf{b_2}} \circ N^1_{\mathbf{W_1},\mathbf{b_1}}(\mathbf{y}_i) \approx \mathbf{u}_i$

Training a network

Given a dataset of clean/corrupted data $\{(\mathbf{u}_j, \mathbf{y}_j)\}_{j=1,\dots,J}$, **training** means:

find $\quad \boldsymbol{\theta} = (\mathbf{W_1}, \mathbf{b_1}, \dots, \mathbf{W_L}, \mathbf{b_L}) \quad$ s.t. $\quad N_{\boldsymbol{\theta}}(\mathbf{y}_i) = N^L_{\mathbf{W_L},\mathbf{b_L}} \circ \dots \circ N^2_{\mathbf{W_2},\mathbf{b_2}} \circ N^1_{\mathbf{W_1},\mathbf{b_1}}(\mathbf{y}_i) \approx \mathbf{u}_i$

To do so introduce loss function:

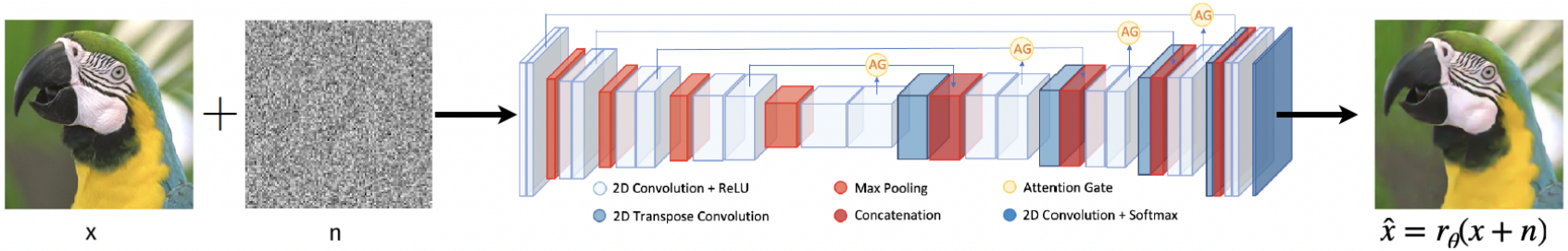
$$\min_{\boldsymbol{\theta}} \ \frac{1}{J} \sum_i \mathcal{L}(N_{\boldsymbol{\theta}}(\mathbf{y}_j), \mathbf{u}_j) + \eta h(\boldsymbol{\theta})$$

Example: $\mathcal{L}(\mathbf{w}_1, \mathbf{w}_2) = \|\mathbf{w}_1 - \mathbf{w}_2\|^2$

- Large-scale, non-convex optimization problem
- Typically, **stochastic** gradient algorithms (e.g., ADAM)
- $h : \Theta \to \mathbb{R}$ is itself a *regularisation* function on network weights, e.g., weights decay $h(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|^2$.

One instructive example: deep denoisers

In imaging, NNs performing denoising are now state-of-the-art[2].



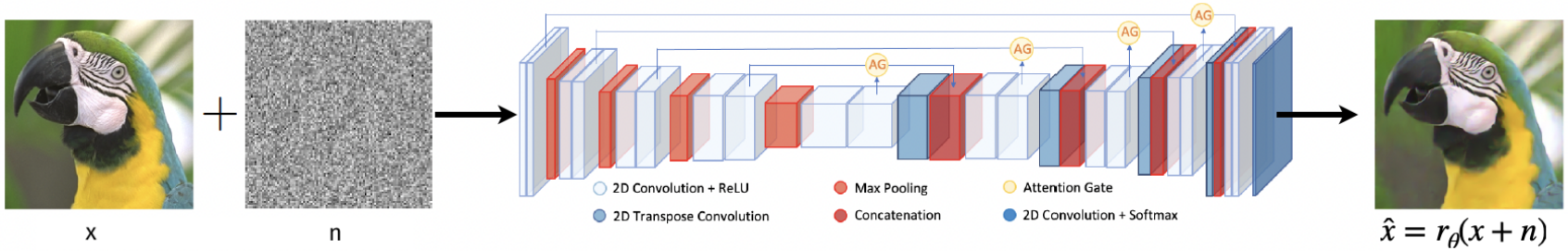
Training is performed using $\{(\mathbf{u}_j, \mathbf{y}_j)\}$ with $\mathbf{y}_j = \mathbf{u}_j + \mathbf{n}_j$, $\mathbf{n}_j \sim \mathcal{N}(\mathbf{0}, \varsigma^2 \mathbf{I})$ by solving:

$$\hat{\boldsymbol{\theta}} \in \arg\min_{\theta} \frac{1}{2J} \sum_j \|\mathbf{u}_j - r_\theta(\mathbf{u}_j)\|^2 \approx \mathbb{E}_{\mathbf{u},\mathbf{y}} \left[\|\mathbf{u} - r_\theta(\mathbf{y})\|^2\right] \qquad \text{(MMSE)}$$

[2] Zhang, Zuo, Chen, Meng, and Zhang, '17, Zhang, Li, Zuo, Zhang, Van Gool, and Timofte, '22

One instructive example: deep denoisers

In imaging, NNs performing denoising are now state-of-the-art[2].



Training is performed using $\{(\mathbf{u}_j, \mathbf{y}_j)\}$ with $\mathbf{y}_j = \mathbf{u}_j + \mathbf{n}_j$, $\mathbf{n}_j \sim \mathcal{N}(\mathbf{0}, \varsigma^2 \mathbf{I})$ by solving:

$$\hat{\boldsymbol{\theta}} \in \arg\min_{\theta} \frac{1}{2J} \sum_j \|\mathbf{u}_j - r_\theta(\mathbf{u}_j)\|^2 \approx \mathbb{E}_{\mathbf{u},\mathbf{y}} \left[\|\mathbf{u} - r_\theta(\mathbf{y})\|^2\right] \qquad \text{(MMSE)}$$

Remark: NN-denoisers approximate MMSE estimators of π_U, with $\mathbf{u}_j \sim \pi_U$.

[2] Zhang, Zuo, Chen, Meng, and Zhang, '17, Zhang, Li, Zuo, Zhang, Van Gool, and Timofte, '22

MMSE denoising and Tweedie's formula

Forward model:

$$\mathbf{y} = \mathbf{u} + \mathbf{n}, \qquad \mathbf{n} \sim \mathcal{N}(0, \varsigma^2 \mathbf{I}), \qquad \mathbf{u} \sim \pi_U$$

Observation distribution: (marginal of $Y = U + N$)

$$\pi_Y(\mathbf{y}) = (\pi_U * \mathcal{N}_{0,\varsigma^2 I})(\mathbf{y})$$

MMSE estimator: characterisation by the Tweedie's formula[3]

$$\boxed{\hat{\mathbf{u}}_{\mathrm{MMSE}}(\mathbf{y}) = \mathbb{E}[\mathbf{u} \mid \mathbf{y}] = \mathbf{y} + \varsigma^2 \nabla \log \pi_Y(\mathbf{y})}$$

Interpretation:

- The optimal MMSE denoiser depends on the *score* $\nabla \log \pi_Y$, i.e., the gradient of the Gaussian-smoothed log-density $\pi_U * \mathcal{N}$
- An MMSE-trained neural denoiser therefore implicitly learns the score of the data distribution upon smoothing.

[3] Laumont, De Bartoli et al., '22

Optimisation-driven deep learning

Plug & play approaches

MAP estimators and Plug-and-Play denoisers

$$\mathbf{u}^* \in \arg\min_{\mathbf{u}} \frac{1}{2\varsigma^2}\|\mathbf{y}-\mathbf{u}\|^2 + g(\mathbf{u}) = \text{prox}_{\varsigma^2 g}(\mathbf{y}).$$

MAP Gaussian denoising with prior $g = -\log \pi_U$ $\quad \Leftrightarrow \quad$ proximal operator of g

MAP estimators and Plug-and-Play denoisers

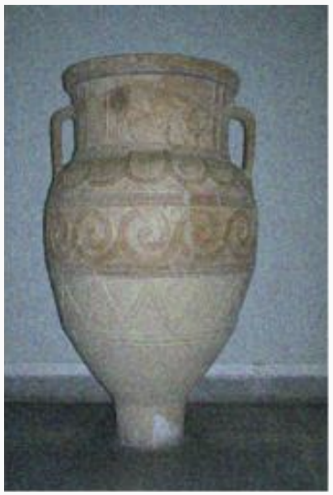
$$\mathbf{u}^* \in \arg\min_{\mathbf{u}} \frac{1}{2\varsigma^2}\|\mathbf{y}-\mathbf{u}\|^2 + g(\mathbf{u}) = \text{prox}_{\varsigma^2 g}(\mathbf{y}).$$

MAP Gaussian denoising with prior $g = -\log \pi_U$ $\quad\Leftrightarrow\quad$ proximal operator of g

Plug-and-Play idea: ~~$\text{prox}_{\varsigma^2 g}$~~ $\quad\longrightarrow\quad$ $D^\theta_\varsigma(\mathbf{y})$, Gaussian denoiser trained via MMSE

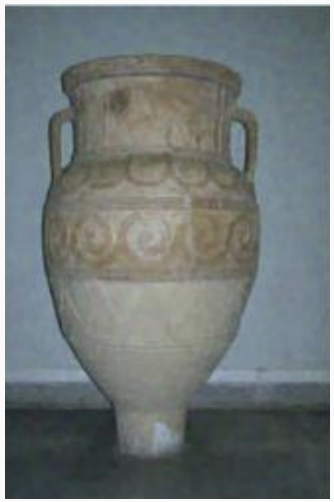
- $D^\theta_\varsigma(\mathbf{y}) \approx \mathbb{E}[\mathbf{u}|\mathbf{y}]$
- In general, D^θ_ς is *not* the prox of any explicit g

Examples of Gaussian denoisers: not only learned ones



y

($\varsigma^2 = 0.01$)



BM3D

(Dabov et al., '07)

Non-Local Means

(Buades et al., '05)

DnCNN, D^θ_ς

(Zhang et al., '17)

PnP-proximal gradient descent

Algorithm: PnP-PGD

Input: $\mathbf{u}^0 \in \mathbb{R}^N$, $\tau > 0$.

for $k \geq 0$ **till convergence**

$$\mathbf{u}^{k+1} = \mathrm{NN}_\theta \left(\mathbf{u}^k - \tau \nabla f(\mathbf{u}^k) \right)$$

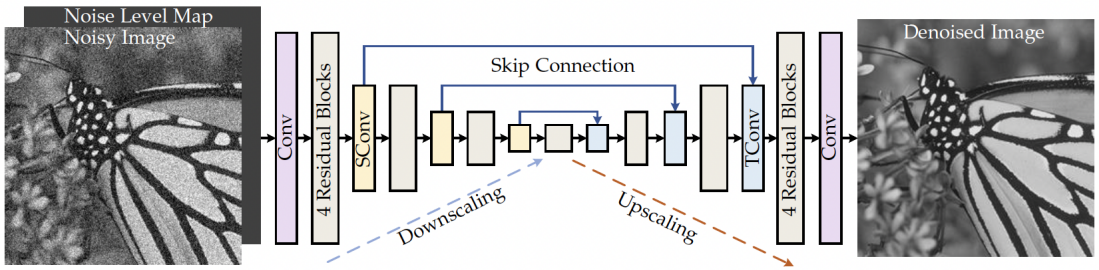
end for

NN_θ trained on denoising/image class, but used for a different task/class!

Example: if $f_{\mathbf{y}}(\mathbf{u}) = \frac{1}{2} \|\mathbf{A}\mathbf{u} - \mathbf{y}\|^2$

$$\mathbf{u}^{k+1} = \mathrm{NN}_\theta \left(\mathbf{u}^k - \tau \mathbf{A}^\top (\mathbf{A}\mathbf{u}^k - \mathbf{y}) \right)$$

Choice of NN_θ? Specialised NN performing Gaussian denoising at a certain noise level ς, i.e. $\mathrm{NN}_\theta = D_\varsigma^\theta$. Interpret as a *denoising of gradient step*.



DRUNet: K. Zhang, Y. Li, W. Zuo, L. Zhang, L. Van Gool, and R. Timofte,'22.

No explicit objective is guaranteed to be minimised! $D_\varsigma^\theta = \mathrm{prox}_{??}$

Convergence of PnP-PGD

$$\mathbf{u}^{k+1} = \mathrm{NN}_\theta\left(\mathbf{u}^k - \tau \nabla f_{\mathbf{y}}(\mathbf{u}^k)\right)$$

Theorem (convergence of PnP-PGD, informal)

Assume that, NN_θ is, for all $\mathbf{u}_1, \mathbf{u}_2 \in \mathbb{R}^N$:

- (*non-expansiveness*) $\|\mathrm{NN}_\theta(\mathbf{u}_1) - \mathrm{NN}_\theta(\mathbf{u}_2)\| \leq \|\mathbf{u}_1 - \mathbf{u}_2\|$
- (*approximation of* $\mathbf{I}$) $\|(\mathrm{NN}_\theta - \mathbf{I})(\mathbf{u}_1) - (\mathrm{NN}_\theta - \mathbf{I})(\mathbf{u}_2)\| \leq \varepsilon \|\mathbf{u}_1 - \mathbf{u}_2\|$.

Then, for suitably small step-sizes τ the sequence $(\mathbf{u}^k)_k$ converges to a fixed point of

$$T(\mathbf{u}) := \mathrm{NN}_\theta\left(\mathbf{u} - \tau \nabla f_{\mathbf{y}}(\mathbf{u})\right).$$

Remarks:

- Convergence to a *fixed point*, not necessarily minimiser of an explicit energy.
- If $\mathrm{NN}_\theta = D_\varsigma^\theta = \mathrm{prox}_g$, classical proximal-gradient convergence applies.

. . . more general theory imposes less restrictions on NN_θ (to avoid lack of expressivity).

(Sreehari et al. '16, Ryu et al., '19)

PnP-Primal-Dual

Model: $\min_{\mathbf{u}} f_{\mathbf{y}}(\mathbf{u}) + g(\mathbf{K}\mathbf{u})$.

(Option 1) *Primal-PnP-dual*: modify dual update, possibly using Moreau identity

$$\operatorname{prox}_{\eta g^*}(\mathbf{w}^k) = \mathbf{v}^{k+1} = \mathbf{w}^k - \eta \operatorname{prox}_{g/\eta}(\mathbf{w}^k/\eta)$$

$\mathbf{w}^k = \mathbf{v}^k + \eta \mathbf{K}\mathbf{u}^k$, keeping primal update fixed:

$$\mathbf{u}^{k+1} = \operatorname{prox}_{\tau f_{\mathbf{y}}}(\mathbf{u}^k - \tau \mathbf{v}^{k+1}).$$

PnP-Primal-Dual

Model: $\min_{\mathbf{u}} f_{\mathbf{y}}(\mathbf{u}) + g(\mathbf{K}\mathbf{u})$.

(Option 1) *Primal-PnP-dual*: modify dual update, possibly using Moreau identity

$$D_{\varsigma}^{\theta_1}(\mathbf{w}^k) \quad \leftarrow \quad \operatorname{prox}_{\eta g^*}(\mathbf{w}^k) = \mathbf{v}^{k+1} = \mathbf{w}^k - \eta \operatorname{prox}_{g/\eta}(\mathbf{w}^k/\eta) \quad \rightarrow \quad \mathbf{w}^k - \eta\, D_{\varsigma}^{\theta_1}(\mathbf{w}^k/\eta)$$

$\mathbf{w}^k = \mathbf{v}^k + \eta \mathbf{K}\mathbf{u}^k$, keeping primal update fixed:

$$\mathbf{u}^{k+1} = \operatorname{prox}_{\tau f_{\mathbf{y}}}(\mathbf{u}^k - \tau \mathbf{v}^{k+1}).$$

PnP-Primal-Dual

Model: $\min_{\mathbf{u}} f_{\mathbf{y}}(\mathbf{u}) + g(\mathbf{K}\mathbf{u})$.

(Option 1) *Primal-PnP-dual*: modify dual update, possibly using Moreau identity

$$D_{\varsigma}^{\theta_1}(\mathbf{w}^k) \quad \leftarrow \quad \operatorname{prox}_{\eta g^*}(\mathbf{w}^k) = \mathbf{v}^{k+1} = \mathbf{w}^k - \eta \operatorname{prox}_{g/\eta}(\mathbf{w}^k/\eta) \quad \rightarrow \quad \mathbf{w}^k - \eta\, D_{\varsigma}^{\theta_1}(\mathbf{w}^k/\eta)$$

$\mathbf{w}^k = \mathbf{v}^k + \eta \mathbf{K}\mathbf{u}^k$, keeping primal update fixed:

$$\mathbf{u}^{k+1} = \operatorname{prox}_{\tau f_{\mathbf{y}}}(\mathbf{u}^k - \tau \mathbf{v}^{k+1}).$$

(Option 2) *PnP-Primal-dual*: modify primal-update

$$\mathbf{u}^{k+1} = D_{\gamma}^{\theta_2}\left(\mathbf{u}^k - \tau \mathbf{K}^\top \mathbf{v}^{k+1}\right)$$

with dual update computed by $\operatorname{prox}_{g^*}$ (if available) or via **Option 1**.

PnP-Primal-Dual

Model: $\min_{\mathbf{u}} f_{\mathbf{y}}(\mathbf{u}) + g(\mathbf{K}\mathbf{u})$.

(Option 1) *Primal-PnP-dual*: modify dual update, possibly using Moreau identity

$$D_{\varsigma}^{\theta_1}(\mathbf{w}^k) \quad \leftarrow \quad \operatorname{prox}_{\eta g^*}(\mathbf{w}^k) = \mathbf{v}^{k+1} = \mathbf{w}^k - \eta \operatorname{prox}_{g/\eta}(\mathbf{w}^k/\eta) \quad \rightarrow \quad \mathbf{w}^k - \eta\, D_{\varsigma}^{\theta_1}(\mathbf{w}^k/\eta)$$

$\mathbf{w}^k = \mathbf{v}^k + \eta \mathbf{K}\mathbf{u}^k$, keeping primal update fixed:

$$\mathbf{u}^{k+1} = \operatorname{prox}_{\tau f_{\mathbf{y}}}(\mathbf{u}^k - \tau \mathbf{v}^{k+1}).$$

(Option 2) *PnP-Primal-dual*: modify primal-update

$$\mathbf{u}^{k+1} = D_{\gamma}^{\theta_2}\left(\mathbf{u}^k - \tau \mathbf{K}^\top \mathbf{v}^{k+1}\right)$$

with dual update computed by $\operatorname{prox}_{g^*}$ (if available) or via **Option 1**.

- Plugging one (or two) denoiser(s) makes the algorithm modular, but may lack an explicit energy minimised by the iterates.
- Choose ς so that the denoiser strength matches the implicit proximal scale (tuning required).

(Adler, Öktem, '17, Ono, '17, Jiu, Pustelnik, '21, Le, Repetti, Pustelnik, '24)

Convergence of PnP-Primal–Dual

Consider Primal-Dual algorithm in the form:

$$\boxed{\begin{aligned}\mathbf{v}^{k+1} &= \mathrm{prox}_{\eta f_{\mathbf{y}}^*}\left(\mathbf{v}^k + \eta \mathbf{K}\,\mathbf{u}^k\right),\\ \mathbf{u}^{k+1} &= \mathrm{NN}_\theta\left(\mathbf{u}^k - \tau \mathbf{K}^\top \mathbf{v}^{k+1}\right),\end{aligned}}$$

Theorem (convergence PnP-Primal-dual, informal)

Assume the denoiser $\mathrm{NN}_\theta = D_\varsigma^\theta$ satisfies:

$$\mathbf{Q}_\theta := 2\mathrm{NN}_\theta - \mathbf{I} \text{ is non-expansive} \underbrace{\Leftrightarrow}_{\mathrm{NN}_\theta \text{ smooth}} \|\nabla \mathbf{Q}_\theta\|_2 \leq 1.$$

If $\tau\eta\|\mathbf{K}\|^2 < 1$, then $(\mathbf{u}^k, \mathbf{v}^k)$ converge to solutions to (suitably defined) optimality conditions of the primal-dual problem.

Important:

- Theory here relies on monotone operator theory (Bauschke, Combettes, '16)
- Conditions for convergence can be imposed during training via an appropriate extra term to the loss.

(see Pesquet, Repetti, Terris, Wiaux, '21, Suzuki, Isono, Ono, '25)

Regularisation by Denoising (RED)

Idea: use the denoiser as a proxy for the gradient of a regulariser, not as its prox.

RED objective: energy is explicit here (Romano, Elad, Milanfar, '17)

$$\boxed{F(\mathbf{u}) = f_{\mathbf{y}}(\mathbf{u}) + \frac{\lambda}{2}\,\mathbf{u}^{\top}\left(\mathbf{u} - \mathrm{NN}_{\theta}(\mathbf{u})\right)}$$

Algorithm: RED

Input: $\mathbf{u}^0$, $\tau > 0$, $\lambda > 0$.
for $k \geq 0$ **till convergence**

$$\mathbf{u}^{k+1} = \mathbf{u}^k - \tau \nabla F(\mathbf{u}^k)$$

end

Theorem (convergence of RED, informal

If NN_{θ} satisfies suitable conditions:

- **(Lipschitz continuity)**: $\|\mathrm{NN}_{\theta}(\mathbf{u}_1) - \mathrm{NN}_{\theta}(\mathbf{u}_2)\| \leq L_{\mathrm{NN}_{\theta}} \|\mathbf{u}_1 - \mathbf{u}_2\|$.
- **(Local homogeneity)**: $\mathbf{J}_{\mathrm{NN}_{\theta}}(\mathbf{u})\,\mathbf{u} = \mathrm{NN}_{\theta}(\mathbf{u})$,
- **(Jacobian symmetry)**: $\mathbf{J}_{\mathrm{NN}_{\theta}}(\mathbf{u}) = \mathbf{J}_{\mathrm{NN}_{\theta}}(\mathbf{u})^{\top}$,

then for $\tau \in (0, 2/L)$ with $L = L_{f_{\mathbf{y}}} + \lambda(1 + L_{\mathrm{NN}_{\theta}})$,
$\nabla F(\mathbf{u}) = \nabla f_{\mathbf{y}}(\mathbf{u}) + \lambda\big(\mathbf{u} - \mathrm{NN}_{\theta}(\mathbf{u})\big)$ and standard (non-convex) GD theory applies:

$$\|\nabla F(\mathbf{u}^k)\| \to 0.$$

PnP VS. classical approaches

$\mathbf{u}$

$\mathbf{y}$, PSNR=22.1

$\mathbf{u}^*$, Tik., PSNR=25.1

$\mathbf{u}^*$, TV, PSNR-25.3.

$\mathbf{u}^*$, PnP, PSNR=26.3.

Test it yourself today!

Summary & take-home messages

- **Composite optimisation problems solving imaging inverse problems:**

$$\mathbf{u}^* \in \underset{\mathbf{u}\in\mathbb{R}^n}{\arg\min}\ F(\mathbf{u}) = f_{\mathbf{y}}(\mathbf{u}) + g(\mathbf{K}\mathbf{u})$$

- **Key tools (convex case):**

$$\text{subdifferential } \partial g, \quad \text{prox}_{\gamma g}(\mathbf{u}) = \underset{\mathbf{z}}{\arg\min}\ g(\mathbf{z}) + \tfrac{1}{2\gamma}\|\mathbf{z}-\mathbf{u}\|^2$$

- **Algorithmic choices** (GD, PGD, Primal-dual. . .)
- **From models to learned priors:**

$$\text{prox}_{\gamma g} \longrightarrow D_{\varsigma}^{\theta}$$

PnP: modular, but no guaranteed correspondence to prox (see Gribonval, Nikolova, '20, Hurault et al. '23).
RED: explicit F under structural assumptions on D_{ς}^{θ}.

Take-home message, towards labs

Choose algorithm by the structure of $f_{\mathbf{y}}$ and g,$\mathbf{K}$, then decide where to plug suitable denoisers as a way to learn prior regularisation.

Bibliography on PnP methods in imaging


Y. Romano, M. Elad, P. Milanfar, *The Little Engine That Could: Regularization by Denoising (RED)*, SIAM Journal on Imaging Sciences, 10(4), 2017.


S. V. Venkatakrishnan, C. A. Bouman, B. Wohlberg, *Plug-and-Play priors for model based reconstruction*, EEE Global Conference on Signal and Information Processing, 2013.


U. S. Kamilov, C. A. Bouman, G. T. Buzzard, B. Wohlberg, *Plug-and-Play Methods for Integrating Physical and Learned Models in Computational Imaging: Theory, algorithms, and applications*, IEEE Signal Processing Magazine, 40(1), 2023.


A. Hauptmann, S. Mukherjee, C.-B. Schönlieb, F. Sherry, *Convergent regularization in inverse problems and linear plug-and-play denoisers* , Foundations of Computational Mathematics, 25, 2025.

Computational resources:

- `https://wustl-cig.github.io/spmpnp/`
- `https://deepinv.github.io/deepinv/auto_examples/plug-and-play/index.html`

Thank you!
Questions?

luca.calatroni@unige.it

Backup

Proper functions

Minimal property to have well-defined minimisation problems.

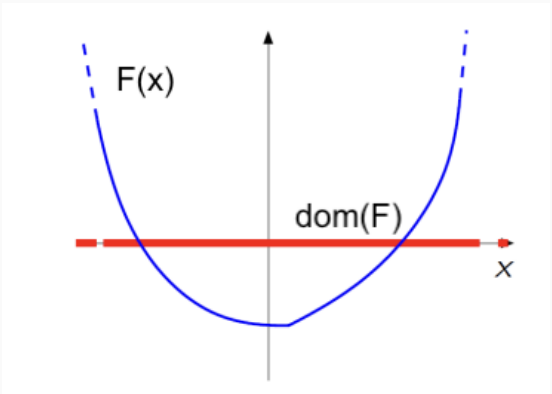
Definition (proper function)

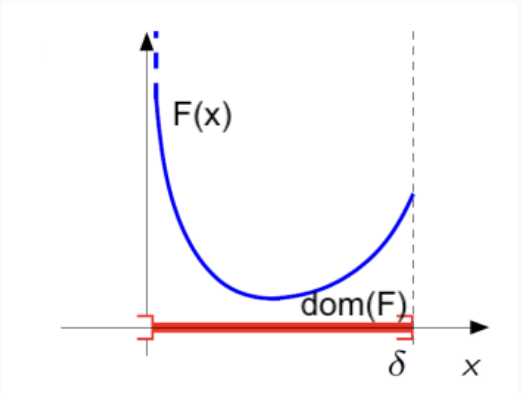
A function $F : \mathbb{R}^n \to \overline{\mathbb{R}}$ is said to be *proper* if:

$$\exists \mathbf{u} \in \mathbb{R}^n \quad \text{such that} \quad F(\mathbf{u}) \neq +\infty.$$

Equivalently, F is proper if

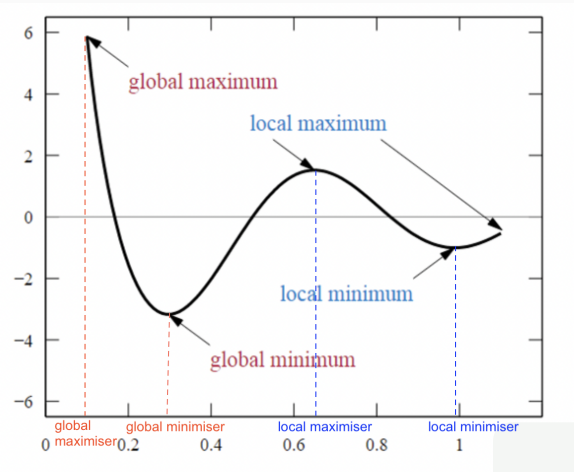
$$\text{dom}(F) := \{\mathbf{u} \in \mathbb{R}^n : F(\mathbf{u}) < +\infty\} \neq \emptyset.$$





Global/local minimisers

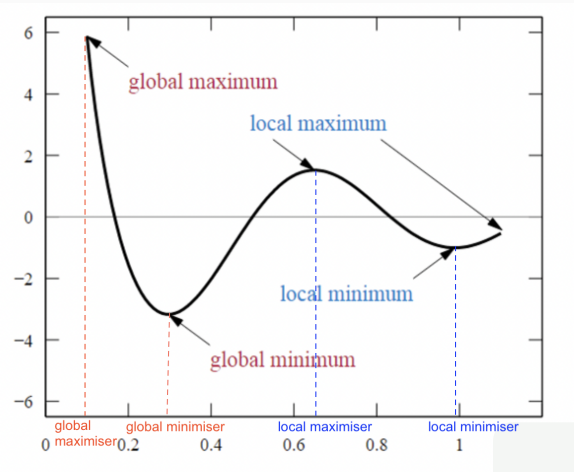
- **Global minimiser**: $\mathbf{u}^* \in \mathbb{R}^n$: $F(\mathbf{u}^*) \leq F(\mathbf{u})$ for every $\mathbf{u} \in \mathbb{R}^n$.
- **Local minimiser**: $\mathbf{u}^* \in \mathbb{R}^n$: there exists $\delta > 0$ and $B_\delta(\mathbf{u}^*)$ such that $F(\mathbf{u}^*) \leq F(\mathbf{u})$ for every $\mathbf{u} \in B_\delta(\mathbf{u}^*)$.



$$\min_{\mathbf{u}\in\mathbb{R}^n} F(\mathbf{u}) \quad \text{VS} \quad \arg\min_{\mathbf{u}\in\mathbb{R}^n} F(\mathbf{u})$$

Global/local minimisers

- **Global minimiser**: $\mathbf{u}^* \in \mathbb{R}^n$: $F(\mathbf{u}^*) \leq F(\mathbf{u})$ for every $\mathbf{u} \in \mathbb{R}^n$.
- **Local minimiser**: $\mathbf{u}^* \in \mathbb{R}^n$: there exists $\delta > 0$ and $B_\delta(\mathbf{u}^*)$ such that $F(\mathbf{u}^*) \leq F(\mathbf{u})$ for every $\mathbf{u} \in B_\delta(\mathbf{u}^*)$.



$$\min_{\mathbf{u}\in\mathbb{R}^n} F(\mathbf{u}) \quad \text{VS} \quad \arg\min_{\mathbf{u}\in\mathbb{R}^n} F(\mathbf{u})$$

Definition (argmin set)

For $F : \mathbb{R}^n \to \overline{\mathbb{R}}$ proper:

$$\arg\min F = \{\mathbf{u}^* \in \mathbb{R}^n : \mathbf{u}^* \text{ is a global minimiser of } F\} \subset \mathbb{R}^n$$

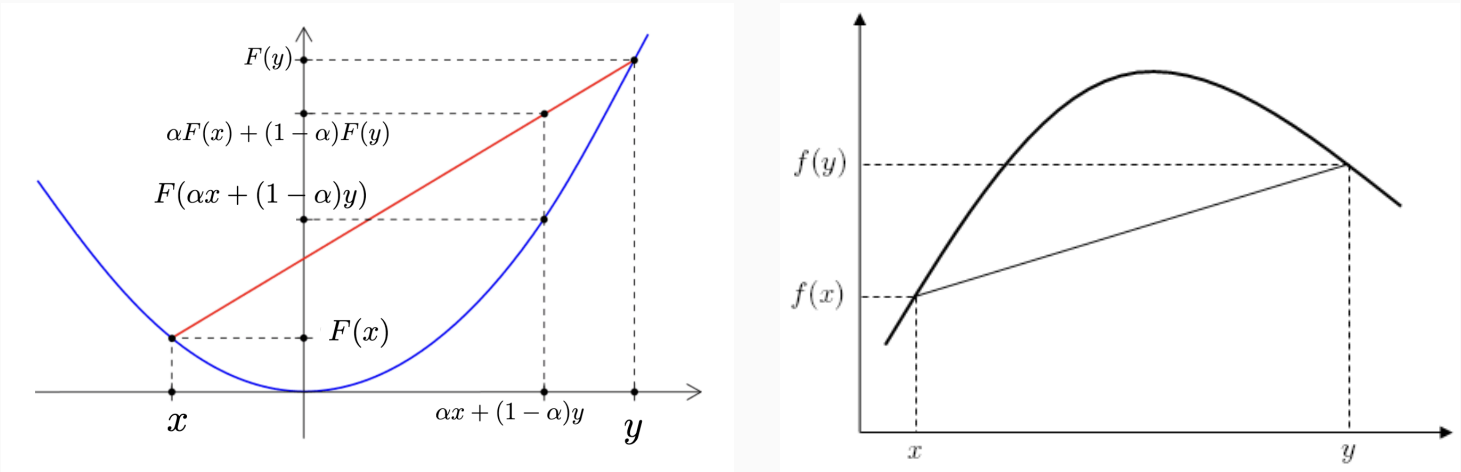
Convex functions

Definition (convex function)

A proper function $F : \mathbb{R}^n \to \overline{\mathbb{R}}$ is *convex* iff:

$$(\forall \mathbf{u}_1, \mathbf{u}_2 \in \mathbb{R}^n), \quad (\forall \alpha \in [0,1]), \quad F(\alpha \mathbf{u}_1 + (1-\alpha)\mathbf{u}_2) \leq \alpha F(\mathbf{u}_1) + (1-\alpha)F(\mathbf{u}_2).$$

Moreover, F is *strictly convex* if the inequality holds when $\mathbf{u}_1, \mathbf{u}_2 \in \text{dom}(F)$, $\mathbf{u}_1 \neq \mathbf{u}_2$ and $\alpha \in (0,1)$. $G : \mathbb{R}^n \to [-\infty, +\infty)$ is *concave* is $F = -G$ is convex.
If a function is not convex nor concave we say that is *non-convex*.



Convex/concave function

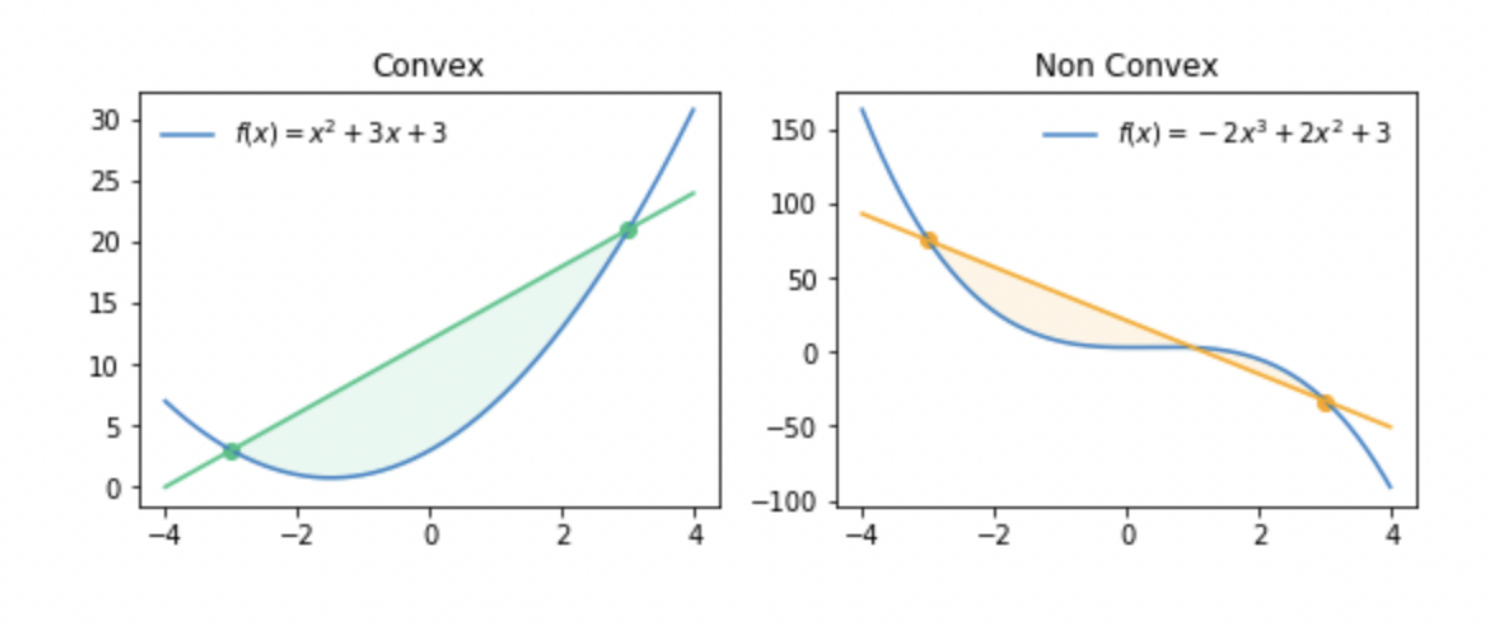
Convex functions

Definition (convex function)

A proper function $F : \mathbb{R}^n \to \overline{\mathbb{R}}$ is *convex* iff:

$$(\forall \mathbf{u}_1, \mathbf{u}_2 \in \mathbb{R}^n), \quad (\forall \alpha \in [0,1]), \quad F(\alpha \mathbf{u}_1 + (1-\alpha)\mathbf{u}_2) \leq \alpha F(\mathbf{u}_1) + (1-\alpha)F(\mathbf{u}_2).$$

Moreover, F is *strictly convex* if the inequality holds when $\mathbf{u}_1, \mathbf{u}_2 \in \text{dom}(F)$, $\mathbf{u}_1 \neq \mathbf{u}_2$ and $\alpha \in (0,1)$. $G : \mathbb{R}^n \to [-\infty, +\infty)$ is *concave* is $F = -G$ is convex.
If a function is not convex nor concave we say that is *non-convex*.



Convex VS. non-convex function

Convex functions

Definition (convex function)

A proper function $F : \mathbb{R}^n \to \overline{\mathbb{R}}$ is *convex* iff:

$$(\forall \mathbf{u}_1, \mathbf{u}_2 \in \mathbb{R}^n), \quad (\forall \alpha \in [0,1]), \quad F(\alpha \mathbf{u}_1 + (1-\alpha)\mathbf{u}_2) \leq \alpha F(\mathbf{u}_1) + (1-\alpha)F(\mathbf{u}_2).$$

Moreover, F is *strictly convex* if the inequality holds when $\mathbf{u}_1, \mathbf{u}_2 \in \text{dom}(F)$, $\mathbf{u}_1 \neq \mathbf{u}_2$ and $\alpha \in (0,1)$. $G : \mathbb{R}^n \to [-\infty, +\infty)$ is *concave* is $F = -G$ is convex.
If a function is not convex nor concave we say that is *non-convex*.

Examples:

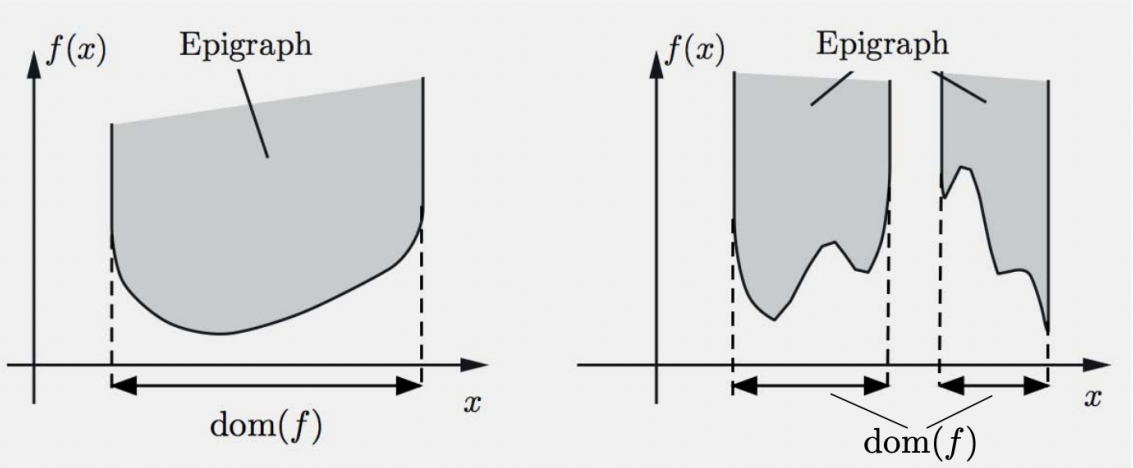
- $F(\mathbf{u}) = \|\mathbf{u}\|$ is convex

  $$\|\alpha\mathbf{u}_1 + (1-\alpha)\mathbf{u}_2\| \leq \|\alpha\mathbf{u}_1\| + \|(1-\alpha)\mathbf{u}_2\| = \alpha\|\mathbf{u}_1\| + (1-\alpha)\|\mathbf{u}_2\| \qquad \forall \mathbf{u}_1, \mathbf{u}_2 \in \mathbb{R}^n$$

- $F(\mathbf{u}) = \|\mathbf{u}\|^2$ is strictly convex
- $F(\mathbf{u}) = \|\mathbf{u}\|_p$, $p \in [1, +\infty)$ are convex

Proposition (epigraph of convex functions)

Let $F : \mathbb{R}^n \to \overline{\mathbb{R}}$ be a proper function. Then F is convex if and only if epi(F) is a convex set.



Proposition (operations with convex functions)

If F_1 and F_2 are convex, and $\alpha, \beta \in \mathbb{R}_{++}$, then $\alpha F_1 + \beta F_2$ is convex.

Lower semi-continuity

Definition (lower semi-continuity)

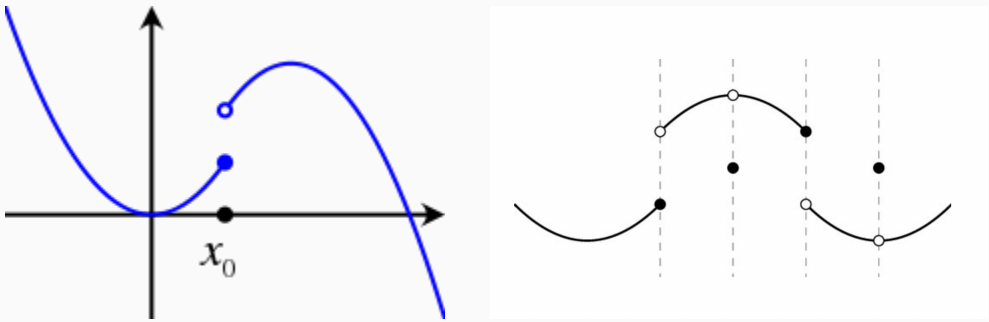
A proper function $F : \mathbb{R}^n \to \overline{\mathbb{R}}$ is *lower semi-continuous (l.s.c.) at* $\mathbf{u} \in \mathbb{R}^n$ if:

$$F(\mathbf{u}) \leq \liminf_{\mathbf{v} \to \mathbf{u}} F(\mathbf{v}).$$

Equivalently, for every sequence $(\mathbf{u}_k)_{k \in \mathbb{N}}$ with $\mathbf{u}_k \to \mathbf{u}$:

$$F(\mathbf{u}) \leq \liminf_{k \to +\infty} F(\mathbf{u}_k) = \lim_{k \to +\infty} \inf \left\{ F(\mathbf{u}_j) : j \geq k \right\}.$$

F is l.s.c. if it is at every $\mathbf{u} \in \mathbb{R}^n$.



Left: lower l.s.c. **Right**: where the function is lower l.s.c.?

Examples of l.s.c. functions

- The functions

$$F(u) = \begin{cases} 0 & \text{if } u \leq 0 \\ 1 & \text{if } u > 0 \end{cases}, \qquad F(u) = \lceil u \rceil = \min \{k \in \mathbb{Z} : u \leq k\}$$

  are l.s.c. (but not continuous).

- The indicator function $\iota_C : \mathbb{R}^n \to \overline{\mathbb{R}}$ of a closed $C \subset \mathbb{R}^n$:

$$\iota_C(\mathbf{u}) = \begin{cases} 0 & \text{if } \mathbf{u} \in C \\ +\infty & \text{if } \mathbf{u} \notin C \end{cases}$$

- All continuous functions (l.s.c + u.s.c.).

Coercivity

How to ensure that the minimum is not attained at "extreme points" of the domain?

Definition (coercivity)

Let $F : \mathbb{R}^n \to \overline{\mathbb{R}}$ be proper. We say that F is *coercive* iff

$$\lim_{\|\mathbf{u}\| \to +\infty} F(\mathbf{u}) = +\infty.$$

Coercivity

How to ensure that the minimum is not attained at "extreme points" of the domain?

Definition (coercivity)

Let $F : \mathbb{R}^n \to \overline{\mathbb{R}}$ be proper. We say that F is *coercive* iff

$$\lim_{\|\mathbf{u}\| \to +\infty} F(\mathbf{u}) = +\infty.$$

Examples:

- $F : \mathbb{R} \to \mathbb{R}_+$, $F(x) = e^x$ is **not** coercive, but $F : \mathbb{R} \to \mathbb{R}_+$, $F(x) = e^{|x|}$ is.
- $F : \mathbb{R}^2 \to \mathbb{R}_+$, $F(x, y) = x^2 + y^2$ is coercive.
- $F : \mathbb{R}^2 \to \mathbb{R}_+$, $F(x, y) = x^2 - 2xy + x^2 = (x - y)^2$ is **not** coercive.