Deep unfolded proximal algorithms

Nelly Pustelnik

TraDE-OPT – July 6th 2022

Collaborations and references

- M Jiu, N Pustelnik, A deep primal-dual proximal network for image restoration IEEE Journal of Selected Topics in Signal Processing 15 (2), 190-203, 2021.
- M. Jiu, N. Pustelnik, Alternative Design of DeepPDNet in the Context of Image Restoration, IEEE Signal Processing Letters, vol. 29, pp. 932 - 936, 2022.
- H. Le, N. Pustelnik, M. Foare, The faster proximal algorithm, the better unfolded deep learning architecture? The study case of image denoising. EUSIPCO, 2022.

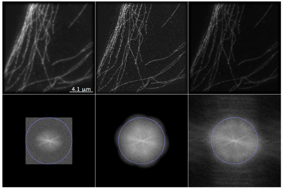
Introduction to inverse problems

Context: Image restoration

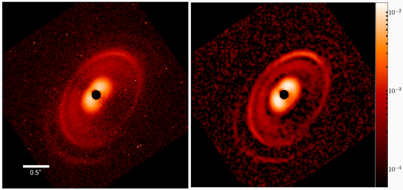
➡ **Data**: $\mathbf{z} \in \mathbb{R}^M$ degraded version of an original image $\overline{\mathbf{x}} \in \mathbb{R}^N$:

$$\mathbf{z} = \mathrm{A}\overline{\mathbf{x}} + \varepsilon$$

- $\mathrm{A} : \mathbb{R}^{M \times N}$: linear degradation (e.g. a blur)
- ε : noise (e.g. Gaussian noise)



SIM



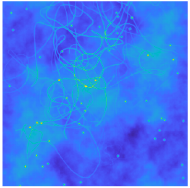
SPHERE-IRDIS

Context: Image restoration

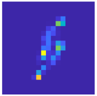
$$\mathbf{z} = \mathrm{A}\bar{\mathbf{x}} \quad \Leftrightarrow \quad \mathbf{z} = \phi * \bar{\mathbf{x}}$$

Context: Image restoration

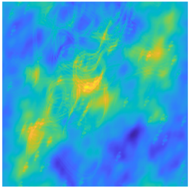
$$\mathbf{z} = \mathrm{A}\bar{\mathbf{x}} \quad \Leftrightarrow \quad \mathbf{z} = \phi * \bar{\mathbf{x}}$$



$*$



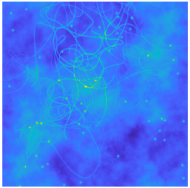
$=$



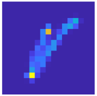
$\bar{\mathbf{x}}$ $\qquad$ ϕ $\qquad$ $\mathbf{z}$

Context: Image restoration

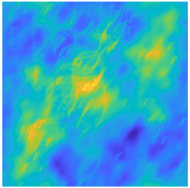
$$\mathbf{z} = \mathrm{A}\bar{\mathbf{x}} \quad \Leftrightarrow \quad \mathbf{z} = \phi * \bar{\mathbf{x}}$$



$*$



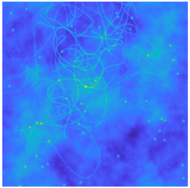
$=$



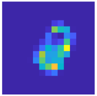
$\bar{\mathbf{x}}$ $\qquad$ ϕ $\qquad$ $\mathbf{z}$

Context: Image restoration

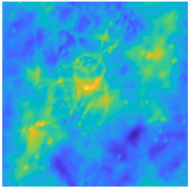
$$\mathbf{z} = \mathrm{A}\bar{\mathbf{x}} \quad \Leftrightarrow \quad \mathbf{z} = \phi * \bar{\mathbf{x}}$$



$*$



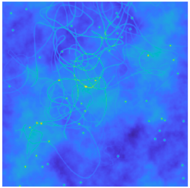
$=$



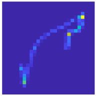
$\bar{\mathbf{x}}$ ϕ $\mathbf{z}$

Context: Image restoration

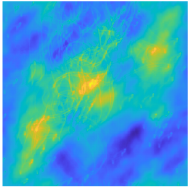
$$\mathbf{z} = \mathrm{A}\bar{\mathbf{x}} \quad \Leftrightarrow \quad \mathbf{z} = \phi * \bar{\mathbf{x}}$$



$*$



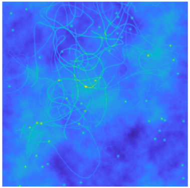
$=$



$\bar{\mathbf{x}}$ $\quad$ ϕ $\quad$ $\mathbf{z}$

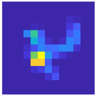
Context: Image restoration

$$\mathbf{z} = \mathrm{A}\bar{\mathbf{x}} \quad \Leftrightarrow \quad \mathbf{z} = \phi * \bar{\mathbf{x}}$$



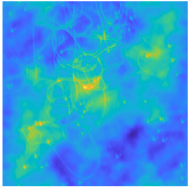
$\bar{\mathbf{x}}$

$*$



ϕ

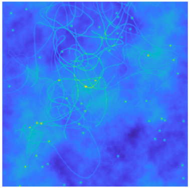
$=$



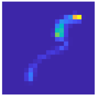
$\mathbf{z}$

Context: Image restoration

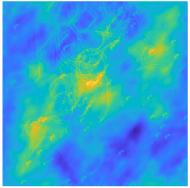
$$\mathbf{z} = \mathrm{A}\bar{\mathbf{x}} \quad \Leftrightarrow \quad \mathbf{z} = \phi * \bar{\mathbf{x}}$$



$*$



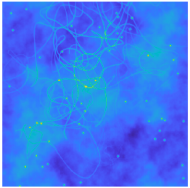
$=$



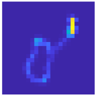
$\bar{\mathbf{x}}$ $\qquad$ ϕ $\qquad$ $\mathbf{z}$

Context: Image restoration

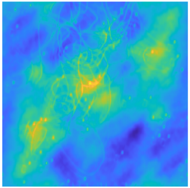
$$\mathbf{z} = \mathrm{A}\overline{\mathbf{x}} \quad \Leftrightarrow \quad \mathbf{z} = \phi * \overline{\mathbf{x}}$$



$*$



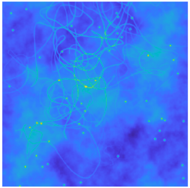
$=$



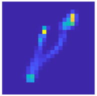
$\overline{\mathbf{x}}$ ϕ $\mathbf{z}$

Context: Image restoration

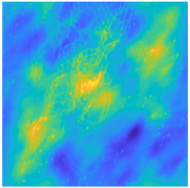
$$\mathbf{z} = \mathrm{A}\bar{\mathbf{x}} \quad \Leftrightarrow \quad \mathbf{z} = \phi * \bar{\mathbf{x}}$$



$*$



$=$



$\bar{\mathbf{x}}$ ϕ $\mathbf{z}$

Context: Image restoration

➡ **Data**: $\mathbf{z} \in \mathbb{R}^M$ degraded version of an original image $\bar{\mathbf{x}} \in \mathbb{R}^N$:

$$\mathbf{z} = A\bar{\mathbf{x}} + \varepsilon$$

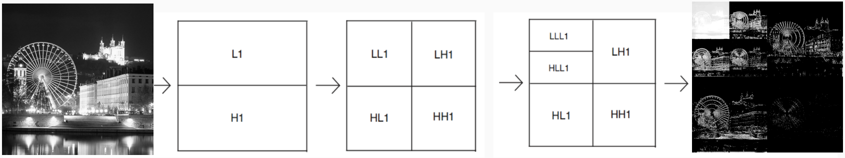
➡ **Goal**: Restore the degraded image $\mathbf{z}$ i.e., find $\widehat{\mathbf{x}}$ close to $\bar{\mathbf{x}}$:

$$\widehat{\mathbf{x}} \in \underset{\mathbf{x} \in \mathbb{R}^N}{\operatorname{Argmin}} \underbrace{\frac{1}{2}\|A\mathbf{x} - \mathbf{z}\|_2^2}_{\text{Data-term}} + \underbrace{\lambda R(\mathbf{x})}_{\text{Penalization}}$$

(a) Degraded (b) Inverse filtering Quadratic regularisation (e) Total variation

Wavelet denoising: $\mathbf{z} = \bar{\mathbf{x}} + \varepsilon$ with $\varepsilon = \mathcal{N}(0, \sigma^2\,\mathbf{I})$

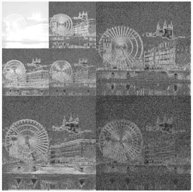
- Wavelets: sparse representation of most natural signals.
- Filterbank implementation of a dyadic wavelet transform: $\Psi \in \mathbb{R}^{N \times N}$.



$\mathbf{z} \in \mathbb{R}^N$ $\zeta = \Psi \mathbf{z}$

Wavelet denoising: $\mathbf{z} = \bar{\mathbf{x}} + \varepsilon$ with $\varepsilon = \mathcal{N}(0, \sigma^2 \mathbf{I})$

$\mathbf{z}$



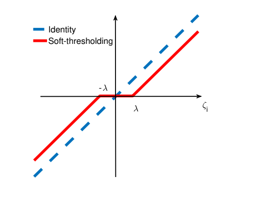
$\boldsymbol{\zeta} = \Psi \mathbf{z}$

$\mathrm{soft}_\lambda(\Psi \mathbf{z})$

$\widehat{\mathbf{x}} = \Psi^* \mathrm{soft}_\lambda(\Psi \mathbf{z})$

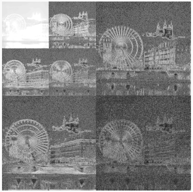
$$\mathrm{soft}_\lambda(\boldsymbol{\zeta}) = \big(\max\{|\zeta_{\underline{i}}| - \lambda, 0\} \, \mathrm{sign}(\zeta_{\underline{i}}) \big)_{\underline{i} \in \Omega}$$

$$= \arg\min_{\boldsymbol{\nu}} \frac{1}{2} \|\boldsymbol{\nu} - \boldsymbol{\zeta}\|_2^2 + \lambda \|\boldsymbol{\nu}\|_1$$



Wavelet denoising: $\mathbf{z} = \bar{\mathbf{x}} + \varepsilon$ with $\varepsilon = \mathcal{N}(0, \sigma^2 \mathbf{I})$

$\mathbf{z}$



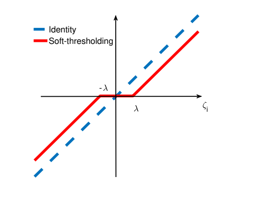
$\boldsymbol{\zeta} = \Psi \mathbf{z}$

$\mathrm{soft}_\lambda(\Psi \mathbf{z})$

$\widehat{\mathbf{x}} = \Psi^* \mathrm{soft}_\lambda(\Psi \mathbf{z})$

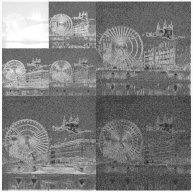
$$\begin{aligned}
\mathrm{soft}_\lambda(\boldsymbol{\zeta}) &= \big(\max\{|\zeta_{\underline{i}}| - \lambda, 0\}\,\mathrm{sign}(\zeta_{\underline{i}})\big)_{\underline{i}\in\Omega} \\
&= \arg\min_{\boldsymbol{\nu}} \frac{1}{2}\|\boldsymbol{\nu} - \boldsymbol{\zeta}\|_2^2 + \lambda\|\boldsymbol{\nu}\|_1 \\
&= \mathrm{prox}_{\lambda\|\cdot\|_1}(\boldsymbol{\zeta})
\end{aligned}$$

$\rightarrow$ **proximity operator**



Wavelet denoising: $\mathbf{z} = \bar{\mathbf{x}} + \varepsilon$ with $\varepsilon = \mathcal{N}(0, \sigma^2 \mathbf{I})$

$\mathbf{z}$



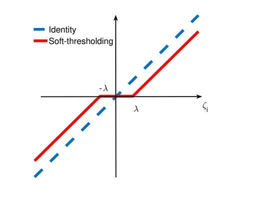
$\boldsymbol{\zeta} = \Psi \mathbf{z}$

$\mathrm{soft}_\lambda(\Psi \mathbf{z})$

$\widehat{\mathbf{x}} = \Psi^* \mathrm{soft}_\lambda(\Psi \mathbf{z})$

$$\begin{aligned}
\mathrm{soft}_\lambda(\boldsymbol{\zeta}) &= \left(\max\{|\zeta_{\underline{i}}| - \lambda, 0\} \, \mathrm{sign}(\zeta_{\underline{i}}) \right)_{\underline{i} \in \Omega} \\
&= \arg\min_{\boldsymbol{\nu}} \frac{1}{2} \|\boldsymbol{\nu} - \boldsymbol{\zeta}\|_2^2 + \lambda \|\boldsymbol{\nu}\|_1 \\
&= \mathrm{prox}_{\lambda \|\cdot\|_1}(\boldsymbol{\zeta})
\end{aligned}$$

$\rightarrow$ **proximity operator**



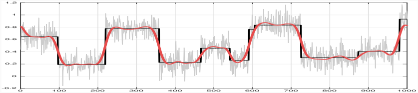
$$\begin{aligned}
\widehat{\mathbf{x}} &= \Psi^* \mathrm{soft}_\lambda(\Psi \mathbf{z}) \\
&= \arg\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{x} - \mathbf{z}\|_2^2 + \lambda \|\Psi \mathbf{x}\|_1 \\
&= \mathrm{prox}_{\lambda \|\Psi \cdot\|_1}(\mathbf{z})
\end{aligned}$$

Piecewise constant denoising: $\mathbf{z} = \bar{\mathbf{x}} + \varepsilon$ with $\varepsilon = \mathcal{N}(0, \sigma^2 \mathbf{I})$

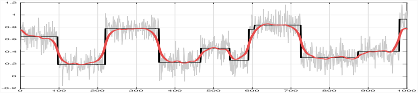
- **Minimization problem**:

$$\widehat{\mathbf{x}}(\mathbf{z}; \widehat{\lambda}) = \arg \min_{\mathbf{x} \in \mathbb{R}^N} \frac{1}{2} \|\mathbf{x} - \mathbf{z}\|_2^2 + \lambda \|\Psi \mathbf{x}\|_\bullet \quad \text{where} \quad \begin{cases} \Psi \mathbf{x} = \psi * \mathbf{x} \\ \lambda > 0 \end{cases}$$

- **Linear denoising**



$\psi = \begin{bmatrix} 1 & -1 \end{bmatrix}$; $\| \cdot \|_\bullet = \| \cdot \|_2^2$; Large λ



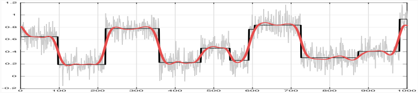
$\psi = \begin{bmatrix} 1 & -1 \end{bmatrix}$; $\| \cdot \|_\bullet = \| \cdot \|_2^2$; Small λ

Piecewise constant denoising: $\mathbf{z} = \bar{\mathbf{x}} + \varepsilon$ with $\varepsilon = \mathcal{N}(0, \sigma^2 \mathbf{I})$

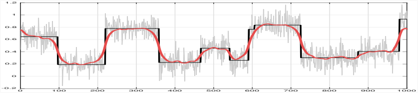
- **Minimization problem**:

$$\widehat{\mathbf{x}}(\mathbf{z}; \widehat{\lambda}) = \arg\min_{\mathbf{x} \in \mathbb{R}^N} \frac{1}{2}\|\mathbf{x} - \mathbf{z}\|_2^2 + \lambda \|\Psi \mathbf{x}\|_\bullet \quad \text{where} \quad \begin{cases} \Psi \mathbf{x} = \psi * \mathbf{x} \\ \lambda > 0 \end{cases}$$

- **Linear denoising**

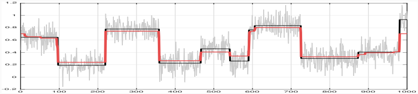


$\psi = \begin{bmatrix} 1 & -1 \end{bmatrix}$; $\|\cdot\|_\bullet = \|\cdot\|_2^2$; Large λ

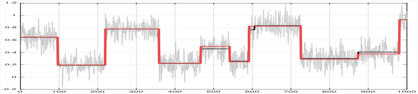


$\psi = \begin{bmatrix} 1 & -1 \end{bmatrix}$; $\|\cdot\|_\bullet = \|\cdot\|_2^2$; Small λ

- **Non-linear denoising**.



$\psi = \begin{bmatrix} 1 & -1 \end{bmatrix}$ and $\|\cdot\|_\bullet = \|\cdot\|_1$



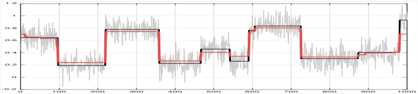
$\psi = \begin{bmatrix} 1 & -1 \end{bmatrix}$ and $\|\cdot\|_\bullet = \|\cdot\|_0$

Piecewise linear denoising: $\mathbf{z} = \bar{\mathbf{x}} + \varepsilon$ with $\varepsilon = \mathcal{N}(0, \sigma^2 \mathbf{I})$

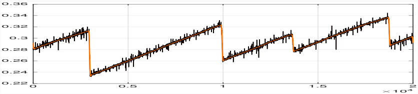
- **Minimization problem**:

$$\widehat{\mathbf{x}}(\mathbf{z}; \widehat{\lambda}) = \arg\min_{\mathbf{x} \in \mathbb{R}^N} \frac{1}{2}\|\mathbf{x} - \mathbf{z}\|_2^2 + \lambda \|\Psi \mathbf{x}\|_\bullet \quad \text{where} \quad \begin{cases} \Psi \mathbf{x} = \psi * \mathbf{x} \\ \lambda > 0 \end{cases}$$

- **Non-linear denoising: piecewise constant/linear**



$\psi = \begin{bmatrix} 1 & -1 \end{bmatrix}$ and $\|\cdot\|_\bullet = \|\cdot\|_1$



$\psi = \begin{bmatrix} 1 & -2 & 1 \end{bmatrix}$ and $\|\cdot\|_\bullet = \|\cdot\|_1$

Context: image restoration

Synthesis formulation

$\widehat{\mathbf{x}} = \Psi^* \widehat{\alpha}$ with $\Psi \in \mathbb{R}^{P \times N}$

$$\widehat{\alpha} \in \underset{\alpha}{\mathrm{Argmin}} \frac{1}{2} \| \mathrm{A} \Psi^* \alpha - \mathbf{z} \|_2^2 + \lambda \| \alpha \|_{\bullet}$$

Analysis formulation

$$\widehat{\mathbf{x}} \in \underset{\mathbf{x}}{\mathrm{Argmin}} \frac{1}{2} \| \mathrm{A} \mathbf{x} - \mathbf{z} \|_2^2 + \lambda \| \Psi \mathbf{x} \|_{\bullet}$$

$\Rightarrow$ **Equivalence for Ψ orthonormal basis.**

(Elad, Milanfar, Ron, 2007) (Chaari, Pustelnik, Chaux, Pesquet, 2009)
(Selesnick, Figueiredo, 2009), (Carlavan, Weiss, Blanc-Féraud, 2010)

(Pustelnik, Benazza-Benhayia, Zheng, Pesquet, 2010)

Context: image restoration

Synthesis formulation

$\widehat{\mathbf{x}} = \Psi^* \widehat{\alpha}$ with $\Psi \in \mathbb{R}^{P \times N}$

$$\widehat{\alpha} \in \underset{\alpha}{\operatorname{Argmin}} \frac{1}{2} \|A\Psi^* \alpha - \mathbf{z}\|_2^2 + \lambda \|\alpha\|_\bullet$$

Analysis formulation

$$\widehat{\mathbf{x}} \in \underset{\mathbf{x}}{\operatorname{Argmin}} \frac{1}{2} \|A\mathbf{x} - \mathbf{z}\|_2^2 + \lambda \|\Psi \mathbf{x}\|_\bullet$$

$\Rightarrow$ **Equivalence for Ψ orthonormal basis.**

- X-lets
- Sparse coding

- Horizontal/vertical gradients: TV
- Hessian operator
- Nonlocal total variation: weighted nonlocal gradients: NLTV
- Local dictionaries of patches

(webpage L. Duval)(Aharon, Elad, Bruckstein, 2006) (Mairal, Sapiro, Elad, 2007)(Gilboa, Osher, 2008)(K Bredies, K Kunisch, T Pock, 2010)(Jacques, Duval, Chaux, Peyré, 2011) (S Lefkimmiatis, A Bourquard, M Unser,2011) (Zoran, Weiss, 2011) (G Kutyniok, D Labate, 2012)(Chierchia et al.,2014)(Boulanger et al., 2018). . .

Proximal algorithms

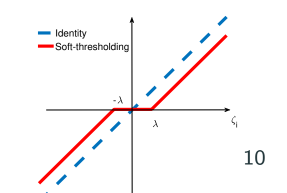
Proximity operator

<u>Definition</u> [Moreau,1965] Let $f : \mathbb{R}^N \to]-\infty, +\infty]$ be a convex, l.s.c., and proper function. The proximity operator of f at point $x \in \mathbb{R}^N$ is the **unique point** denoted by $\operatorname{prox}_f x$ such that

$$(\forall x \in \mathbb{R}^N) \qquad \operatorname{prox}_f x = \arg\min_{v \in \mathbb{R}^N} \frac{1}{2}\|x - v\|^2 + f(v)$$

➡ **Existing many closed form expressions**

- $\operatorname{prox}_{\lambda\|\cdot\|_1}$: **soft-thresholding** with a fixed threshold $\lambda > 0$.
- exhaustive list: **PROX Repository**



Proximity operator

Definition [Moreau,1965] Let $f\colon \mathbb{R}^N \to \,]-\infty,+\infty]$ be a convex, l.s.c., and proper function. The proximity operator of f at point $\mathrm{x} \in \mathbb{R}^N$ is the **unique point** denoted by $\operatorname{prox}_f \mathrm{x}$ such that

$$(\forall \mathrm{x} \in \mathbb{R}^N) \qquad \operatorname{prox}_f \mathrm{x} = \arg\min_{\mathrm{v}\in\mathbb{R}^N} \frac{1}{2}\|\mathrm{x}-\mathrm{v}\|^2 + f(\mathrm{v})$$

➡ **Existing many closed form expressions**

- $\operatorname{prox}_{\lambda\|\cdot\|_1}$: **soft-thresholding** with a fixed threshold $\lambda > 0$.
- exhaustive list: **PROX Repository**

➡ **More complicated task**: $\operatorname{prox}_{f_1+f_2}$, $\operatorname{prox}_{f\circ\Psi}$.

Iterative scheme

➡ **Minimization problem** :

$$\widehat{\mathrm{x}} \in \underset{\mathrm{x}}{\operatorname{Argmin}}\ f_1(\mathrm{x}) + f_2(\mathrm{x})$$

➡ **Design of a recursive sequence of the form**

$$(\forall k \in \mathbb{N}) \qquad \mathrm{x}^{[k+1]} = \mathbf{\Phi}\mathrm{x}^{[k]},$$

Gradient descent	$\mathbf{\Phi} = \mathbf{I} - \tau(\nabla f_1 + \nabla f_2)$
Proximal point algorithm	$\mathbf{\Phi} = \operatorname{prox}_{\tau(f_1+f_2)}$
Forward-Backward	$\mathbf{\Phi} = \operatorname{prox}_{\tau f_2}(\mathbf{I} - \tau\nabla f_1)$
Peaceman-Rachford	$\mathbf{\Phi} = (2\operatorname{prox}_{\tau f_2} - \mathbf{I}) \circ (2\operatorname{prox}_{\tau f_1} - \mathbf{I})$
Douglas-Rachford	$\mathbf{\Phi} = \operatorname{prox}_{\tau f_2}(2\operatorname{prox}_{\tau f_1} - \mathbf{I}) + \mathbf{I} - \operatorname{prox}_{\tau f_1}$

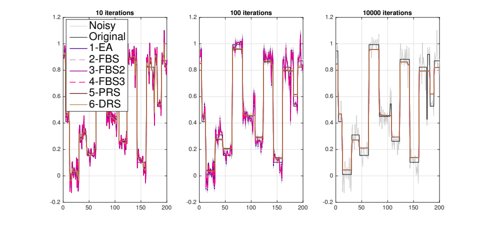
Iterative schemes: prox versus grad

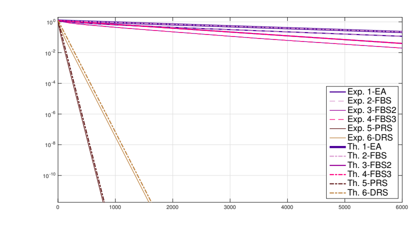
➡ **Minimization problem** :

$$\widehat{\mathrm{x}} \in \underset{\mathrm{x}}{\mathrm{Argmin}}\ f_1(\mathrm{x}) + f_2(\mathrm{x})$$

➡ **Convergence of the sequence** $\left(\mathrm{x}^{[k+1]}\right)_{k\in\mathbb{N}}$ already derived in the literature under specific assumptions for each algorithms.

➡ **New result: Convergence rate and comparisons**: requires **strong convexity** (f-$\frac{\rho}{2}\|\cdot\|^2$ convex). [Briceño-Arias, P., 2021]





Iterative scheme

➡ **Minimization problem** :

$$\widehat{x} \in \underset{x}{\operatorname{Argmin}} \; f_1(x) + h_2(\Psi x)$$

- ☛ **Require the computation of** $\operatorname{prox}_{h_2(\Psi \cdot)}$. **Few closed form.**
- ☛ **Reformulation in the dual**: $\min_{w \in \mathcal{G}} f_1^*(-\Psi^* w) + h_2^*(w)$,
- ☛ **Primal-dual algorithms**: $\min_{x} f_1(x) + f_2(x) + h_2(\Psi x)$,

  [Condat,2013][Vũ,2013] [Chambolle-Pock,2011]

  $\rightarrow$ with f_1 ν-gadient Lipschitz.

Hyperparameters setting: $\tau > 0$, $\gamma > 0$, such that $\frac{1}{\tau} - \gamma \|\Psi\|^2 > \frac{\nu}{2}$

For $k = 0, 1, \ldots$

$$\left\lfloor \begin{array}{l} w^{[k+1]} = \operatorname{prox}_{\tau f_2}\left(w^{[k]} - \tau \nabla f_1(w^{[k]}) - \tau \Psi^* x^{[k]}\right) \\ x^{[k+1]} = \operatorname{prox}_{\gamma h_2^*}\left(x^{[k]} + \gamma \Psi(2w^{[k+1]} - w^{[k]})\right) \end{array} \right.$$

Iterative scheme

➡ **Minimization problem** :

$$\widehat{x} \in \underset{x}{\mathrm{Argmin}}\ f_1(x) + h_2(\Psi x)$$

- **Require the computation of** $\mathrm{prox}_{h_2(\Psi \cdot)}$. **Few closed form.**
- **Reformulation in the dual**: $\min_{w \in \mathcal{G}} f_1^*(-\Psi^* w) + h_2^*(w)$,
- **Primal-dual algorithms**: $\min_{x} f_1(x) + f_2(x) + h_2(\Psi x)$,

  [Condat,2013][Vũ,2013] [Chambolle-Pock,2011]

  $\rightarrow$ Acceleration with f_2 **strongly convex**.

Hyperparameters setting: $\tau > 0$, $\gamma > 0$, such that $\frac{1}{\tau} - \gamma \|\Psi\|^2 > \frac{\nu}{2}$

For $k = 0, 1, \ldots$

$$\left\lfloor \begin{array}{l} w^{[k+1]} = \mathrm{prox}_{\tau f_2}\left(w^{[k]} - \tau \nabla f_1(w^{[k]}) - \tau \Psi^* x^{[k]}\right) \\ x^{[k+1]} = \mathrm{prox}_{\gamma h_2^*}\left(x^{[k]} + \gamma \Psi (2 w^{[k+1]} - w^{[k]})\right) \end{array} \right.$$

Context

Data: $\mathbf{z}$

↓

Direct/acquisition model

↓

Design a cost function

↓

Minimization algorithm

↓

Hyperparameter tuning

↓

Estimated parameters: $\widehat{\mathbf{x}}(\mathbf{z};\widehat{\lambda})$

$$\mathbf{z} = \mathcal{D}(\mathrm{A}\overline{\mathbf{x}})$$

↓

$$\widehat{\mathbf{x}}(\mathbf{z};\lambda) \in \underset{\mathrm{x}}{\mathrm{Argmin}}\; L(\mathbf{x};\mathbf{z}) + \lambda R(\mathbf{x})$$

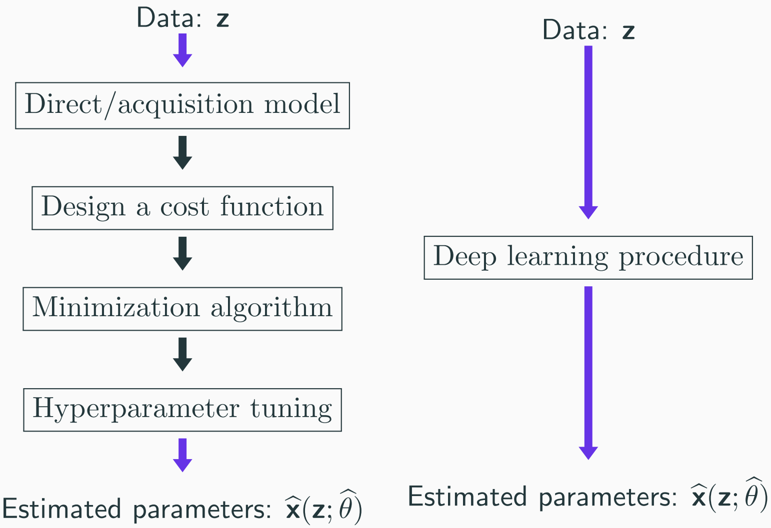
↓

Sequence $\mathbf{x}^{[k+1]} = \boldsymbol{\Phi}\mathbf{x}^{[k]}$

↓

$$\widehat{\lambda} \in \underset{\lambda}{\mathrm{Argmin}} \|\overline{\mathbf{x}} - \widehat{\mathbf{x}}(\mathbf{z};\lambda)\|_2^2$$

Standard learning and deep learning



Design Deep-NN architectures with proximal algorithms

Generalities about deep learning

- **Database**: $\mathcal{S} = \left\{(\mathbf{u}_i, \mathbf{c}_i) \in \mathcal{H} \times \mathcal{G} \;\middle|\; i \in \{1, \ldots, \mathbb{L}\}\right\}$
- **Goal**: Learn a prediction function $d_{\boldsymbol{\theta}} \colon \mathcal{H} \to \mathcal{G}$
- **Deep NN predictor**:

$$d_{\boldsymbol{\theta}}(\mathbf{u}) = \eta^{[K]}\big(\mathrm{W}^{[K]} \ldots \eta^{[1]}(\mathrm{W}^{[1]}\mathbf{u} + b^{[1]}) \ldots + b^{[K]}$$

  - ⊙ Linear operators: $\mathrm{W}^{[1]}, \mathrm{W}^{[2]}, \ldots, \mathrm{W}^{[K]}$
  - ⊙ Activation functions: $\eta^{[1]}, \eta^{[2]}, \ldots, \eta^{[K]}$
  - ⊙ Biais vectors: $b^{[1]}, b^{[2]}, \ldots, b^{[K]}$

$\Rightarrow$ $\boldsymbol{\theta} = \{\mathrm{W}^{[1]}, \ldots, \mathrm{W}^{[K]}, b^{[1]}, \ldots, b^{[K]}\}$

- A NN is characterise by a set of parameters $\boldsymbol{\theta} \in \mathbb{R}^S$ that are learned during a **training process**:

$$\underset{\boldsymbol{\theta}}{\text{minimise}} \;\; \frac{1}{\#\mathbb{L}} \sum_{i=1}^{\mathbb{L}} F(\mathbf{c}_i, d_{\boldsymbol{\theta}}(\mathbf{u}_i)) \;+\; \lambda R(\boldsymbol{\theta})$$

Training a NN for inverse problem task

☛ **Database**: $\mathcal{S} = \left\{ (\mathbf{u}_i, \mathbf{c}_i) \in \mathcal{H} \times \mathcal{G} \;\middle|\; i \in \{1, \dots, \mathbb{L}\} \right\}$

Training a NN for inverse problem task

- **Database**: $\mathcal{S} = \left\{(\mathbf{z}_i, \overline{\mathbf{x}}_i) \in \mathbb{R}^M \times \mathbb{R}^N \;\middle|\; i \in \{1, \ldots, \mathbb{L}\}\right\}$
  - We consider two sets of images: the *training set* $(\mathbf{z}_i, \overline{\mathbf{x}}_i)_{i\in\mathbb{I}}$ of size $\sharp\,\mathbb{I}$ and the *testing set* $(\mathbf{z}_j, \overline{\mathbf{x}}_j)_{j\in\mathbb{J}}$ of size $\sharp\,\mathbb{J}$ where

$$(\forall i \in \mathbb{I} \cup \mathbb{J}) \quad \mathbf{z}_i = \mathrm{A}\overline{\mathbf{x}}_i + \varepsilon_i$$

Training a NN for inverse problem task

- **Database**: $\mathcal{S} = \left\{ (\mathbf{z}_i, \overline{\mathbf{x}}_i) \in \mathbb{R}^M \times \mathbb{R}^N \;\middle|\; i \in \{1, \ldots, \mathbb{L}\} \right\}$
  - We consider two sets of images: the *training set* $(\mathbf{z}_i, \overline{\mathbf{x}}_i)_{i \in \mathbb{I}}$ of size $\sharp\,\mathbb{I}$ and the *testing set* $(\mathbf{z}_j, \overline{\mathbf{x}}_j)_{j \in \mathbb{J}}$ of size $\sharp\,\mathbb{J}$ where

$$(\forall i \in \mathbb{I} \cup \mathbb{J}) \quad \mathbf{z}_i = \mathrm{A}\overline{\mathbf{x}}_i + \varepsilon_i$$

- **Training**:
  - The NN is trained using the *training set* to estimate:

$$\widehat{\boldsymbol{\theta}} \in \underset{\boldsymbol{\theta} \in \mathbb{R}^S}{\mathrm{Argmin}} \; \frac{1}{\sharp\,\mathbb{I}} \sum_{i \in \mathbb{I}} \| \overline{\mathbf{x}}_i - d_{\boldsymbol{\theta}}(\mathbf{z}_i) \|^2$$

- **Testing**:
  - The learned NN $d_{\widehat{\boldsymbol{\theta}}}$ is then validated on the testing set. A properly trained network should satisfy

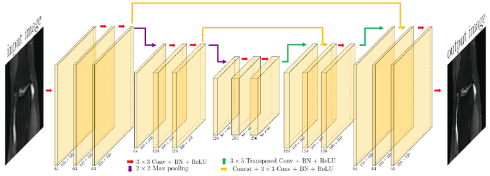
$$(\forall j \in \mathbb{J}) \quad \overline{\mathbf{x}}_j \approx d_{\widehat{\boldsymbol{\theta}}}(\mathbf{z}_j) \,.$$

Design $d_{\boldsymbol{\theta}}$

- **End-to-end approaches**
  - Start from an estimate $\widetilde{\mathbf{x}}$ (e.g. backprojected image in MRI: $\widetilde{\mathbf{x}} = (\mathrm{A}^*\mathrm{A})^{1}\mathrm{A}^*\mathbf{z}$ and apply the NN as a *post-processing* :

$$(\forall j \in \mathbb{J}) \quad \overline{\mathbf{x}}_j \approx d_{\widehat{\boldsymbol{\theta}}}((\mathrm{A}^*\mathrm{A})^{1}\mathrm{A}^*\mathbf{z}_j) \, .$$

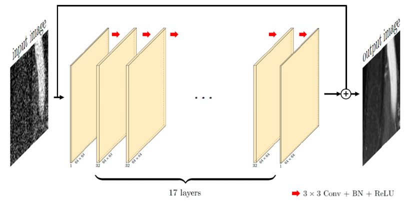
  - Example: Unet architectures



Design d_θ

☛ Plug&Play approaches

- ☛ NNs can be incorporated into optimisation algorithms (e.g. FB, HQS)
- ☛ Usually simple NN architectures are used in PnP.
- ☛ Example: DnCNN architecture



Design $d_{\boldsymbol{\theta}}$

Find an estimate $\widehat{\mathbf{x}}$ of $\overline{\mathbf{x}}$ from the observed measurements $\mathbf{z} = \boldsymbol{\Phi}\overline{\mathbf{x}} + \varepsilon$.

$$\text{Find } \widehat{\mathbf{x}} \in \operatorname{Argmin} \frac{1}{2}\|\mathrm{A}\mathbf{x} - \mathbf{z}\|^2 + \lambda R(\mathbf{x})$$

- NNs incorporated in forward-backward scheme:

$$\mathbf{x}^{[k+1]} = \operatorname{prox}_{\lambda R}(\mathbf{x}^{[k]} - \mathrm{A}^*(\mathrm{A}\mathbf{x}^{[k]} - \mathbf{z}))$$

- $\operatorname{prox}_{\lambda R}$ acts as a denoiser and can be replaced by standard denoiser such as BM3D, NLmeans,... or NN architectures.

$$\mathbf{x}^{[k+1]} = d_{\boldsymbol{\theta}}(\mathbf{x}^{[k]} - \mathrm{A}^*(\mathrm{A}\mathbf{x}^{[k]} - \mathbf{z}))$$

- Convergence of $(\mathbf{x}^{[k]})_{k\in\mathbb{N}}$ ensured if denoiser firmly nonexpansive.
  - ✘ Most of the existing denoisers used in PnP do not satisfy this condition
  - ✔ Recent works propose denoisers that can be built to satisfy this condition

(Hasannasab *et al.* 2020, Terris *et al.* 2020, Terris *et al.* 2021, Repetti *et al.* 2022)

Design $d_{\boldsymbol{\theta}}$

Find an estimate $\widehat{\mathbf{x}}$ of $\overline{\mathbf{x}}$ from the observed measurements $\mathbf{z} = \mathbf{\Phi}\overline{\mathbf{x}} + \varepsilon$.

$$\text{Find } \widehat{\mathbf{x}} \in \operatorname*{Argmin}_{\mathbf{x}} \frac{1}{2}\|\mathrm{A}\mathbf{x} - \mathbf{z}\|^2 + \lambda R(\mathbf{x})$$

Half Quadratic Splitting (HQS) Algorithm:

$$\widehat{\mathbf{x}} \in \operatorname*{Argmin}_{\mathbf{x}} \frac{1}{2}\|\mathrm{A}\mathbf{x} - \mathbf{z}\|^2 + \lambda R(\mathbf{u}) \quad \text{s.t.} \quad \mathbf{u} = \mathbf{x}$$

Design $d_{\boldsymbol{\theta}}$

Find an estimate $\widehat{\mathbf{x}}$ of $\overline{\mathbf{x}}$ from the observed measurements $\mathbf{z} = \boldsymbol{\Phi}\overline{\mathbf{x}} + \varepsilon$.

$$\text{Find } \widehat{\mathbf{x}} \in \operatorname*{Argmin}_{\mathbf{x}} \frac{1}{2}\|\mathrm{A}\mathbf{x} - \mathbf{z}\|^2 + \lambda R(\mathbf{x})$$

Half Quadratic Splitting (HQS) Algorithm:

$$\widehat{\mathbf{x}} \in \operatorname*{Argmin}_{\mathbf{x}} \frac{1}{2}\|\mathrm{A}\mathbf{x} - \mathbf{z}\|^2 + \lambda R(\mathbf{u}) \quad \text{s.t.} \quad \mathbf{u} = \mathbf{x}$$

reformulated as: $\widehat{\mathbf{x}} \in \operatorname*{Argmin}_{\mathbf{x}} \frac{1}{2}\|\mathrm{A}\mathbf{x} - \mathbf{z}\|^2 + \lambda R(\mathbf{u}) + \frac{\mu}{2}\|\mathbf{u} - \mathbf{x}\|^2$ where $\mu > 0$

Design d_{θ}

Find an estimate $\widehat{\mathbf{x}}$ of $\overline{\mathbf{x}}$ from the observed measurements $\mathbf{z} = \mathbf{\Phi}\overline{\mathbf{x}} + \varepsilon$.

$$\text{Find } \widehat{\mathbf{x}} \in \underset{\mathbf{x}}{\text{Argmin}} \frac{1}{2}\|\mathrm{A}\mathbf{x} - \mathbf{z}\|^2 + \lambda R(\mathbf{x})$$

Half Quadratic Splitting (HQS) Algorithm:

$$\widehat{\mathbf{x}} \in \underset{\mathbf{x}}{\text{Argmin}} \frac{1}{2}\|\mathrm{A}\mathbf{x} - \mathbf{z}\|^2 + \lambda R(\mathbf{u}) \quad \text{s.t.} \quad \mathbf{u} = \mathbf{x}$$

reformulated as: $\widehat{\mathbf{x}} \in \underset{\mathbf{x}}{\text{Argmin}} \frac{1}{2}\|\mathrm{A}\mathbf{x} - \mathbf{z}\|^2 + \lambda R(\mathbf{u}) + \frac{\mu}{2}\|\mathbf{u} - \mathbf{x}\|^2$ where $\mu > 0$

and solved by alternating minimization:

$$\begin{cases} \mathbf{x}^{[k]} = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathrm{A}\mathbf{x} - \mathbf{z}\|^2 + \mu\|\mathbf{u}^{[k-1]} - \mathbf{x}\|^2 \\ \mathbf{u}^{[k]} = \arg\min_{\mathbf{u}} \frac{1}{2}\|\mathbf{u} - \mathbf{x}^{[k]}\|^2 + \frac{\mu\lambda}{2} R(\mathbf{u}) \end{cases}$$

Design $d_{\boldsymbol{\theta}}$

Find an estimate $\widehat{\mathbf{x}}$ of $\overline{\mathbf{x}}$ from the observed measurements $\mathbf{z} = \boldsymbol{\Phi}\overline{\mathbf{x}} + \varepsilon$.

$$\text{Find } \widehat{\mathbf{x}} \in \underset{\mathbf{x}}{\mathrm{Argmin}} \frac{1}{2}\|\mathrm{A}\mathbf{x} - \mathbf{z}\|^2 + \lambda R(\mathbf{x})$$

Half Quadratic Splitting (HQS) Algorithm:

$$\widehat{\mathbf{x}} \in \underset{\mathbf{x}}{\mathrm{Argmin}} \frac{1}{2}\|\mathrm{A}\mathbf{x} - \mathbf{z}\|^2 + \lambda R(\mathbf{u}) \quad \text{s.t.} \quad \mathbf{u} = \mathbf{x}$$

reformulated as: $\widehat{\mathbf{x}} \in \underset{\mathbf{x}}{\mathrm{Argmin}} \frac{1}{2}\|\mathrm{A}\mathbf{x} - \mathbf{z}\|^2 + \lambda R(\mathbf{u}) + \frac{\mu}{2}\|\mathbf{u} - \mathbf{x}\|^2$ where $\mu > 0$

and solved by alternating minimization:

$$\begin{cases} \mathbf{x}^{[k]} = \underset{\mathbf{x}}{\arg\min} \frac{1}{2}\|\mathrm{A}\mathbf{x} - \mathbf{z}\|^2 + \mu\|\mathbf{u}^{[k-1]} - \mathbf{x}\|^2 \\ \mathbf{u}^{[k]} = \mathrm{prox}_{\frac{\mu\lambda}{2}R}(\mathbf{x}^{[k]}) \end{cases}$$

Design $d_{\boldsymbol{\theta}}$

Find an estimate $\widehat{\mathbf{x}}$ of $\overline{\mathbf{x}}$ from the observed measurements $\mathbf{z} = \boldsymbol{\Phi}\overline{\mathbf{x}} + \varepsilon$.

$$\text{Find } \widehat{\mathbf{x}} \in \operatorname*{Argmin}_{\mathbf{x}} \frac{1}{2}\|\mathrm{A}\mathbf{x} - \mathbf{z}\|^2 + \lambda R(\mathbf{x})$$

Half Quadratic Splitting (HQS) Algorithm:

$$\widehat{\mathbf{x}} \in \operatorname*{Argmin}_{\mathbf{x}} \frac{1}{2}\|\mathrm{A}\mathbf{x} - \mathbf{z}\|^2 + \lambda R(\mathbf{u}) \quad \text{s.t.} \quad \mathbf{u} = \mathbf{x}$$

reformulated as: $\widehat{\mathbf{x}} \in \operatorname*{Argmin}_{\mathbf{x}} \frac{1}{2}\|\mathrm{A}\mathbf{x} - \mathbf{z}\|^2 + \lambda R(\mathbf{u}) + \frac{\mu}{2}\|\mathbf{u} - \mathbf{x}\|^2$ where $\mu > 0$

and solved by alternating minimization:

$$\begin{cases} \mathbf{x}^{[k]} = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathrm{A}\mathbf{x} - \mathbf{z}\|^2 + \mu\|\mathbf{u}^{[k-1]} - \mathbf{x}\|^2 \\ \mathbf{u}^{[k]} = d_{\boldsymbol{\theta}}(\mathbf{x}^{[k]}) \end{cases}$$

Design $d_{\boldsymbol{\theta}}$

- ☛ **End-to-end approaches**
- ☛ **Plug&Play approaches**
- ☛ **Unfolded/unrolled approaches**

Unfolded networks

Synthesis formulation & proximal gradient descent: LISTA

- **Synthesis formulation:**

$$\min_{\mathbf{x}} \frac{1}{2}\|A\Psi^*\mathbf{x} - \mathbf{z}\|_2^2 + \lambda\|\mathbf{x}\|_1$$

where $H = A\Psi^* \in \mathbb{R}^{\overline{N}\times N}$

- **Forward-backward iterations:**

$$\mathbf{x}^{[k+1]} = \mathrm{prox}_{\tau\lambda\|\cdot\|_1}(\mathbf{x}^{[k]} - \tau H^*(H\mathbf{x}^{[k]} - \mathbf{z}))$$

- **Reformulation:**

$$\mathbf{x}^{[k+1]} = \mathrm{prox}_{\tau\lambda\|\cdot\|_1}((\mathbf{I} - \tau H^*H)\mathbf{x}^{[k]} + \tau H^*\mathbf{z}))$$

- **Layer network:** [Gregor, LeCun, 2010]

$$\mathbf{x}^{[k+1]} = \underbrace{\mathrm{prox}_{\tau\lambda\|\cdot\|_1}}_{\eta^{[k]}}\Big(\underbrace{\mathrm{Id} - \tau H^*H}_{W^{[k]}}\ \mathbf{x}^{[k]} + \underbrace{\tau H^*\mathbf{z}}_{b^{[k]}}\Big)$$

Preliminary remarks

[Combettes, Pesquet, 2020]

- **Most of activation functions are proximity operator** : ReLU, Unimodal sigmoid, Softmax ...

- Let $\mathrm{W}^{[k]}$ be a bounded linear operators, b_k a vector, η_k proximity operators (1/2-averaged operator), $d_{\boldsymbol{\theta}} = \mathrm{T}_K \circ \cdots \circ \mathrm{T}_1$ with $\mathsf{T}_k = \eta_k(\mathrm{W}_k \cdot + b_k)$ model allows to derive tight Lipschitz bounds for feedforward neural networks in order to evaluate their **robustness** i.e.

$$\|d_{\boldsymbol{\theta}}(x+\epsilon) - d_{\boldsymbol{\theta}}(x)\| \leqslant \chi \|\epsilon\|$$

.

Analysis formulation and the proposed DeepPDNet

➡ **Analysis formulation:** $\min_{\mathbf{x}} \frac{1}{2}\|\mathrm{A}\mathbf{x} - \mathbf{z}\|_2^2 + \|\mathrm{H}\mathbf{x}\|_1 \quad \text{where } \mathrm{H} = \lambda\Psi$

➡ **Condat-Vũ iterations:**

$$\begin{cases} \mathbf{x}^{[k+1]} = \mathbf{x}^{[k]} - \tau \mathrm{A}^*(\mathrm{A}\mathbf{x}^{[k]} - \mathbf{z}) - \tau \mathrm{H}^* \mathrm{y}^{[k]} \\ \mathrm{y}^{[k+1]} = \mathrm{prox}_{\gamma\|\cdot\|_1^*}\left(\mathrm{y}^{[k]} + \gamma \mathrm{H}(2\mathbf{x}^{[k+1]} - \mathbf{x}^{[k]})\right) \end{cases}$$

➡ **Reformulation:**

$$\begin{cases} \mathbf{x}^{[k+1]} = (\mathrm{Id} - \tau \mathrm{A}^*\mathrm{A})\mathbf{x}^{[k]} - \tau \mathrm{H}^* \mathrm{y}^{[k]} + \tau \mathrm{A}^* \mathrm{z} \\ \mathrm{y}^{[k+1]} = \mathrm{prox}_{\gamma\|\cdot\|_1^*}\left(\gamma \mathrm{H}(\mathrm{Id} - 2\tau \mathrm{A}^*\mathrm{A})\mathbf{x}^{[k]} + (\mathrm{Id} - 2\tau\gamma \mathrm{H}\mathrm{H}^*)\mathrm{y}^{[k]} + 2\tau\gamma \mathrm{H}\mathrm{A}^*\mathrm{z}\right). \end{cases}$$

➡ **Layer network:**

$$\begin{bmatrix} \mathbf{x}^{[k+1]} \\ \mathrm{y}^{[k+1]} \end{bmatrix} = \underbrace{\begin{matrix} \mathbf{I} \\ \mathrm{prox}_{\gamma\|\cdot\|_1^*} \end{matrix}}_{\eta^{[k]}} \left(\underbrace{\begin{bmatrix} \mathrm{Id} - \tau \mathrm{A}^*\mathrm{A} & -\tau \mathrm{H}^* \\ \gamma \mathrm{H}(\mathrm{Id} - 2\tau \mathrm{A}^*\mathrm{A}) & \mathrm{Id} - 2\tau\gamma \mathrm{H}\mathrm{H}^* \end{bmatrix}}_{\mathrm{W}^{[k]}} \begin{bmatrix} \mathbf{x}^{[k]} \\ \mathrm{y}^{[k]} \end{bmatrix} + \underbrace{\begin{bmatrix} \tau \mathrm{A}^*\mathrm{z} \\ 2\tau\gamma \mathrm{H}\mathrm{A}^*\mathrm{z} \end{bmatrix}}_{\mathrm{b}^{[k]}} \right)$$

Analysis formulation and the proposed DeepPDNet

$$\mathrm{d}_\Theta(\mathbf{x}) = \eta^{[K]}\big(W^{[K]}\ldots\eta^{[1]}(W^{[1]}\mathbf{x}+b^{[1]})\ldots+b^{[K]}\big)$$

➡ **Network with fixed layer:** $\boldsymbol{\theta} = \{\mathrm{H}, \tau, \gamma\}$

$$\begin{bmatrix}\mathbf{x}^{[k+1]}\\ \mathrm{y}^{[k+1]}\end{bmatrix} = \underbrace{\begin{matrix}\mathbf{I}\\ \mathrm{prox}_{\gamma\|\cdot\|_1^*}\end{matrix}}_{\eta^{[k]}}\left(\underbrace{\begin{bmatrix}\mathrm{Id}-\tau\mathrm{A}^*\mathrm{A} & -\tau\mathrm{H}^*\\ \gamma\mathrm{H}(\mathrm{Id}-2\tau\mathrm{A}^*\mathrm{A}) & \mathrm{Id}-2\tau\gamma\mathrm{H}\mathrm{H}^*\end{bmatrix}}_{W^{[k]}}\begin{bmatrix}\mathbf{x}^{[k]}\\ \mathrm{y}^{[k]}\end{bmatrix} + \underbrace{\begin{bmatrix}\tau\mathrm{A}^*\mathrm{z}\\ 2\tau\gamma\mathrm{H}\mathrm{A}^*\mathrm{z}\end{bmatrix}}_{b^{[k]}}\right)$$

➡ **Network with variable layers:** $\boldsymbol{\theta} = \{\mathrm{H}^{[k]}, \tau^{[k]}, \gamma^{[k]}, \}_{1\leqslant k\leqslant K}$

$$\begin{bmatrix}\mathbf{x}^{[k+1]}\\ \mathrm{y}^{[k+1]}\end{bmatrix} = \underbrace{\begin{matrix}\mathbf{I}\\ \mathrm{prox}_{\gamma_k\|\cdot\|_1^*}\end{matrix}}_{\eta^{[k]}}\left(\underbrace{\begin{bmatrix}\mathrm{Id}-\tau_k\mathrm{A}^*\mathrm{A} & -\tau_k\mathrm{H}_k^*\\ \gamma_k\mathrm{H}_k(\mathrm{Id}-2\tau_k\mathrm{A}^*\mathrm{A}) & \mathrm{Id}-2\tau\gamma_k\mathrm{H}_k\mathrm{H}_k^*\end{bmatrix}}_{W^{[k]}}\begin{bmatrix}\mathbf{x}^{[k]}\\ \mathrm{y}^{[k]}\end{bmatrix} + \underbrace{\begin{bmatrix}\tau_k\mathrm{A}_k^*\mathrm{z}\\ 2\tau_k\gamma_k\mathrm{H}_k\mathrm{A}^*\mathrm{z}\end{bmatrix}}_{b^{[k]}}\right)$$

+ specificities for the first and last layers.

Analysis formulation and the proposed DeepPDNet

➡ **Learn a prediction function $d_{\boldsymbol{\theta}}$:**

$$\widehat{\boldsymbol{\theta}} \in \underset{\boldsymbol{\theta}}{\mathrm{Argmin}}\; \mathrm{E}(\boldsymbol{\theta}) := \frac{1}{\sharp\,\mathbb{I}} \sum_{i\in\mathbb{I}} \|\overline{\mathbf{x}}_i - d_{\boldsymbol{\theta}}(\mathbf{z}_i)\|^2$$

➡ **Gradient based strategy**

$$\boldsymbol{\theta}_{\ell+1}^{[k]} = \boldsymbol{\theta}_{\ell}^{[k]} - \gamma_{\boldsymbol{\theta}} \frac{\partial E(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^{[k]}}$$

Analysis formulation and the proposed DeepPDNet

➡ **Learn a prediction function $d_{\boldsymbol{\theta}}$:**

$$\widehat{\boldsymbol{\theta}} \in \underset{\boldsymbol{\theta}}{\operatorname{Argmin}}\ \mathrm{E}(\boldsymbol{\theta}) := \frac{1}{\sharp\,\mathbb{I}} \sum_{i\in\mathbb{I}} \|\overline{\mathbf{x}}_i - d_{\boldsymbol{\theta}}(\mathbf{z}_i)\|^2$$

➡ **Gradient based strategy**

$$\boldsymbol{\theta}^{[k]}_{\ell+1} = \boldsymbol{\theta}^{[k]}_{\ell} - \gamma_{\boldsymbol{\theta}} \frac{\partial E}{\partial \mathrm{u}^{[K]}} \frac{\partial \mathrm{u}^{[K]}}{\partial \mathrm{u}^{[K-1]}} \cdots \frac{\partial \mathrm{u}^{[k+1]}}{\partial \mathrm{u}^{[k]}} \frac{\partial \mathrm{u}^{[k]}}{\partial \boldsymbol{\theta}^{[k]}}$$

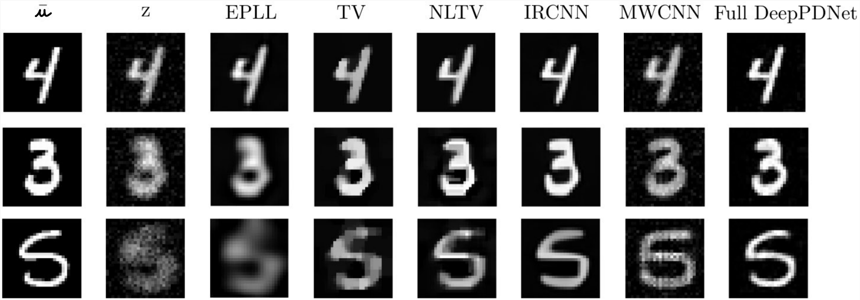
where

$$\frac{\partial \mathrm{u}^{[k]}}{\partial \mathrm{u}^{[k-1]}} = \frac{d\eta^{[k]}(\mathrm{v}^{[k]})}{d\mathrm{v}^{[k]}} \mathrm{W}^{[k]}$$

$$\frac{\partial \mathrm{u}^{[k]}}{\partial \boldsymbol{\theta}^{[k]}} = \left(\frac{\partial \eta^{[k]}(\mathrm{v}^{[k]})}{\partial \mathrm{v}^{[k]}} \left(\frac{\partial \mathrm{W}^{[k]}}{\partial \boldsymbol{\theta}^{[k]}} \mathrm{u}^{[k-1]} + \frac{\partial b^{[k]}}{\partial \boldsymbol{\theta}^{[k]}} \right) + \frac{\partial \eta^{[k]}(\mathrm{v}^{[k]})}{\partial \boldsymbol{\theta}^{[k]}} \right)$$

with $\mathrm{v}^{[k]} = \mathrm{W}^{[k]}\mathrm{u}^{[k-1]} + b^{[k]}$ and $\mathrm{u}^{[k]} = \eta^{[k]}(\mathrm{v}^{[k]})$

Analysis formulation and proposed DeepPDNet



Design of H_k: global vs local structured H_k

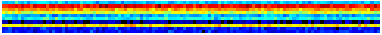
➡ **Proposed DeepPDNet**: linear transform $\mathrm{H}_k \in \mathbb{R}^{P \times N}$, where each row corresponds to a learned pattern of the image.

➡ **Studied architectures for H_k**:

- dense matrix,
- block-sparse matrix inspired by the local patch dictionary,
- combination of dense and block-sparse matrices.

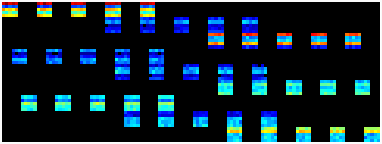
➡ **Examples of H_k**:

Dense matrix



10×121 and $\|\mathrm{H}_k\|_0 = 1210$

Block-sparse matrix



45×121 and $\|\mathrm{H}_k\|_0 = 1125$

Design of H_k: global vs local structured H_k

➡ **Fully connected layer or convolutional layer** are implemented in $(\mathrm{W}^{[k]})_{1\leqslant k\leqslant K}$ as being either a dense matrix or a block-sparse matrix.

➡ **DeepPDNet**: $\mathrm{W}^{[k]}$ has a special structure coming from Condat-Vũ proximal iterations, whose expression is:

$$\mathrm{W}^{[k]} = \begin{pmatrix} \mathrm{Id} - \tau_k \mathrm{A}^*\mathrm{A} & -\tau_k (\mathrm{H}_k)^* \\ \sigma_k \mathrm{H}_k(\mathrm{Id} - 2\tau_k \mathrm{A}^*\mathrm{A}) & \mathrm{Id} - 2\tau_k\sigma_k \mathrm{H}_k(\mathrm{H}_k)^* \end{pmatrix}.$$

In this work, dense matrix or block-sparse matrix at the level of the analysis operator H_k.

Design of H_k: global vs local structured H_k

➡ **Value of P and sparsity rate** for different choices of local sparse H_k.

Setting	"f28s28n10"	"f14s7n10"	f7s7n10'	"f7s3n10"	"f5s2n10"
P	10	90	160	640	1210
Sparsity rate	0%	75%	93.75%	93.75%	96.81%

➡ **Comparison results between global and local H_k on the validation set of MNIST dataset** from data degraded by a uniform 3×3 blur and a Gaussian noise with $\alpha = 20$.

	PSNR		SSIM	
P	Global	Local sparse	Global	Local sparse
10	21.64	21.61	0.7846	0.7831
90	22.35	23.06	0.8052	0.8287
160	22.35	23.06	0.8052	0.8370
640	22.49	24.48	0.8076	0.9122
1210	22.49	24.80	0.8112	0.9278

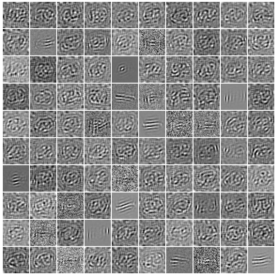
Design of H_k: global vs local structured H_k

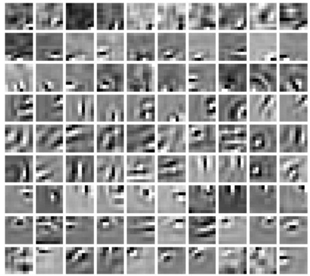
➡ **Performance of combination of multiple local sparse filters** on the validation MNIST dataset with uniform blur filter 3×3 and Gaussian noise $\alpha = 20$.

Fusion	P	PSNR/SSIM
"f5s2n10"	1210	24.80/0.9278
"f5s2n10" + "f7s3n10"	1700	25.04/0.9317
"f5s2n10" + "f7s3n10" + "f14s7n10"	1790	25.06/0.9301
"f5s2n10" + "f7s3n10" + "f14s7n10" + "f28s28n10"	1800	25.33/0.9335

Design of H_k: global vs local structured H_k

➡ Visualization of the rows of H_k for the layer $k = 6$ on the validation MNIST dataset with uniform blur filter 3×3 and Gaussian noise $\alpha = 20$





Partial versus Full DeepPDNet

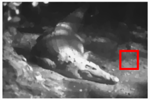
- Partial: $\gamma_k = 0.99 \frac{(1/\tau_k - \|\mathrm{A}\|^2/2)}{\|\mathrm{H}_k\|^2}$
- Full: All parameters are learned.

Method	3×3 Blur			5×5 Blur	7×7 Blur
	$\alpha = 10$	$\alpha = 20$	$\alpha = 30$	$\alpha = 20$	$\alpha = 20$
	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
EPLL	24.02/0.8564	20.99/0.7628	19.05/0.6871	16.42/0.5629	13.97/0.3265
TV	25.07/0.8583	19.58/0.7004	18.86/0.6681	18.86/0.6681	16.31/0.5665
NLTV	25.49/0.8697	21.98/0.7738	20.73/0.7353	20.73/0.7353	16.79/0.6228
MWCNN	19.16/0.7219	18.53/0.6782	17.78/0.6499	15.83/0.5343	13.04/0.3175
IRCNN	**28.52**/0.8904	25.00/0.8193	22.63/0.7723	21.46/0.7698	18.29/0.6546
P-DeepPDNet	23.67/0.8366	22.03/0.7983	20.93/0.7750	17.96/0.6534	16.21/0.5505
F-DeepPDNet	27.40/**0.9410**	**25.09**/**0.9254**	**23.61**/**0.9097**	**22.43**/**0.8738**	**20.43**/**0.8157**

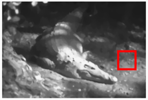
BSD dataset

$\bar{u}$

z



TV



NLTV

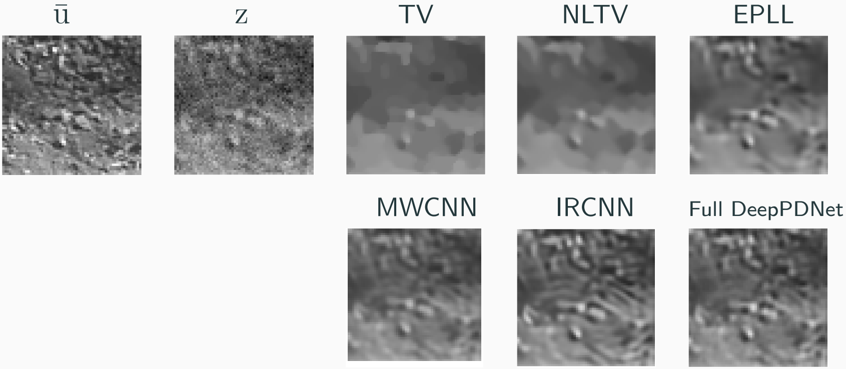
EPLL

MWCNN

IRCNN

Full DeepPDNet

BSD dataset



BSD dataset

	Blur filter 3×3			Blur filter 5×5		
	$\alpha = 15$	$\alpha = 25$	$\alpha = 50$	$\alpha = 15$	$\alpha = 25$	$\alpha = 50$
Method	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
TV	25.52/0.6746	25.16/0.6634	23.27/0.5836	24.04/0.6141	23.83/0.6047	22.77/0.5622
NLTV	25.86/0.6875	25.49/0.6780	23.52/0.5932	24.22/0.6238	24.02/0.6165	22.88/0.5711
EPLL	27.01/0.7450	25.60/0.6785	23.72/0.6137	25.32/0.6674	24.38/0.6198	22.99/0.5715
MWCNN	26.56/0.7537	25.76/0.7136	**23.88**/**0.6265**	24.39/0.6533	24.03/0.6313	22.88/**0.5759**
IRCNN	26.78/**0.7840**	**26.13**/**0.7203**	23.63/0.5981	24.66/**0.6947**	24.64/**0.6555**	22.96/0.5651
DeepPDNet (Q=28, K=6)	25.83/0.6628	24.63/0.6042	23.37/0.5789	24.44/0.6086	23.62/0.5612	22.27/0.5026
DeepPDNet (Q=10, K=20)	**27.33**/0.7637	25.95/0.7055	23.69/0.6052	**25.48**/0.6819	**24.66**/0.6430	**23.04**/0.5717

Unfolded networks: faster algorithm better DNN architecture ?

Focus on denoising problem

➡ **Minimization problem** : $\widehat{\mathbf{x}} = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{x}-\mathbf{z}\|^2 + \|\Psi\mathbf{x}\|_1$

➡ **Dual reformulation**: $\widehat{\mathbf{w}} \in \operatorname{Argmin}_{\mathrm{w}\in\mathcal{G}} \frac{1}{2}\|\mathbf{z}-\Psi^\top\mathbf{w}\|^2 + \iota_{\|\cdot\|_\infty\leqslant 1}(\mathbf{w})$

- Primal solution: $\widehat{\mathbf{x}} = \mathbf{z} - \Psi^\top\widehat{\mathbf{w}}$.
- Solution obtained with proximal gradient based procedure.
- Accelerated schemes (e.g., FISTA).

➡ **Primal-dual algorithms**:

- Resolution with Chambolle-Pock iterations.
- Acceleration when the data-term is **strongly convex**.

Hyperparameters setting: $\tau > 0$, $\gamma > 0$, such that $\tau\gamma\|\Psi\|^2 < 1$

For $k = 0, 1, \ldots$

$$\left\lfloor \begin{array}{l} \mathrm{w}^{[k+1]} = \operatorname{prox}_{\tau\|\cdot-\mathbf{z}\|_2^2}\left(\mathrm{w}^{[k]} - \tau\Psi^*\mathrm{x}^{[k]}\right) \\ \mathrm{x}^{[k+1]} = \operatorname{prox}_{\gamma\|\cdot\|_1^*}\left(\mathrm{x}^{[k]} + \gamma\Psi(2\mathrm{w}^{[k+1]} - \mathrm{w}^{[k]})\right) \end{array} \right.$$

(F)ISTA in the dual

➡ **Minimization problem**: $\widehat{\mathbf{x}} = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{x}-\mathbf{z}\|^2 + \|\Psi\mathbf{x}\|_1$

➡ **Dual reformulation**: $\widehat{\mathbf{w}} \in \operatorname*{Argmin}_{\mathrm{w}\in\mathcal{G}} \frac{1}{2}\|\mathbf{z}-\Psi^\top\mathbf{w}\|^2 + \iota_{\|\cdot\|_\infty\leqslant 1}(\mathbf{w})$

➡ **(F)ISTA to solve dual reformulation**:

Set $\mathrm{w}_1 \in \mathbb{R}^{|\mathbb{F}|}$, and $\mathrm{y}_1 \in \mathbb{R}^{|\mathbb{F}|}$. For every iteration k,

$$\left\lfloor \begin{array}{ll} \mathrm{w}_{k+1} & = \operatorname{prox}_{\iota_{\|\cdot\|_\infty\leqslant 1}}\left((\mathbf{I}-\tau_k\Psi\Psi^\top)\mathrm{y}_k + \tau_k\Psi\mathbf{z}\right) \\ \mathrm{y}_{k+1} & = (1+\alpha_k)\mathrm{w}_{k+1} - \alpha_k\mathrm{w}_k \end{array}\right.$$

➡ **Preliminary remarks**:

• FISTA: $(\mathrm{w}_k)_{k\in\mathbb{N}}$ converges to $\widehat{\mathrm{w}}$ when $\alpha_k = \frac{t_k-1}{t_{k+1}}$ and $t_{k+1} = \frac{k+a-1}{a}$, $a>2$, $\tau < \frac{1}{\|\Psi\|^2}$ and $\widetilde{F}(\mathrm{w}_k) - \widetilde{F}(\widehat{\mathrm{w}}) \leqslant \frac{\zeta}{k^2}$.

• ISTA: When $\alpha_k \equiv 0$, $(\mathrm{w}_k)_{k\in\mathbb{N}}$ converges to $\widehat{\mathrm{w}}$ when $\tau < \frac{2}{\|\Psi\|^2}$ for this limit case, and $\widetilde{F}(\mathrm{w}_k) - \widetilde{F}(\widehat{\mathrm{w}}) \leqslant \frac{\zeta}{k}$.

• (F)ISTA: $\widehat{\mathbf{x}} = \mathbf{z} - \Psi^\top\widehat{\mathrm{w}}$

(F)ISTA in the dual

➡ **Minimization problem**: $\widehat{\mathbf{x}} = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{x} - \mathbf{z}\|^2 + \|\Psi\mathbf{x}\|_1$

➡ **(F)ISTA to solve dual reformulation**:

Set $\mathrm{w}_1 \in \mathbb{R}^{|\mathbb{F}|}$, and $\mathrm{y}_1 \in \mathbb{R}^{|\mathbb{F}|}$. For every iteration k,

$$
\left\lfloor
\begin{array}{ll}
\mathrm{w}_{k+1} & = \mathrm{prox}_{\iota_{\|\cdot\|_\infty \leqslant 1}}\left((\mathbf{I} - \tau_k \Psi\Psi^\top)\mathrm{y}_k + \tau_k \Psi \mathbf{z} \right) \\
\mathrm{y}_{k+1} & = (1+\alpha_k)\mathrm{w}_{k+1} - \alpha_k \mathrm{w}_k
\end{array}
\right.
$$

➡ **Proposition** : The proximity operator of the conjugate of the ℓ_1-norm scaled by parameter $\lambda > 0$ fits the HardTanh activation function, i.e., for every $\mathbf{x} = (\mathbf{x}_i)_{1\leqslant i\leqslant N}$:

$$\mathrm{P}_{\|\cdot\|_\infty \leqslant \lambda}(\mathbf{x}) = \mathrm{HardTanh}_\lambda(\mathbf{x}) = (\mathrm{p}_i)_{1\leqslant i\leqslant N}$$

where

$$
\mathrm{p}_i = \begin{cases} -\lambda & \text{if} \quad \mathrm{p}_i < -\lambda, \\ \lambda & \text{if} \quad \mathrm{p}_i > \lambda, \\ \mathrm{p}_i & \text{otherwise.} \end{cases}
$$

(F)ISTA in the dual

➡ **Minimization problem**: $\widehat{\mathbf{x}} = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{x} - \mathbf{z}\|^2 + \|\Psi \mathbf{x}\|_1$

➡ **(F)ISTA to solve dual reformulation**:

Set $\mathrm{w}_1 \in \mathbb{R}^{|\mathbb{F}|}$, and $\mathrm{y}_1 \in \mathbb{R}^{|\mathbb{F}|}$. For every iteration k,

$$\left\lfloor \begin{array}{ll} \mathrm{w}_{k+1} & = \mathrm{HardTanh}_1\Big((\mathbf{I} - \tau_k \Psi\Psi^\top)\mathrm{y}_k + \tau_k \Psi \mathbf{z}\Big) \\ \mathrm{y}_{k+1} & = (1+\alpha_k)\mathrm{w}_{k+1} - \alpha_k \mathrm{w}_k \end{array} \right.$$

➡ **Proposition** : The proximity operator of the conjugate of the ℓ_1-norm scaled by parameter $\lambda > 0$ fits the HardTanh activation function, i.e., for every $\mathbf{x} = (\mathbf{x}_i)_{1 \leqslant i \leqslant N}$:

$$\mathrm{P}_{\|\cdot\|_\infty \leqslant \lambda}(\mathbf{x}) = \mathrm{HardTanh}_\lambda(\mathbf{x}) = (\mathrm{p}_i)_{1 \leqslant i \leqslant N}$$

where

$$\mathrm{p}_i = \begin{cases} -\lambda & \text{if} \quad \mathrm{p}_i < -\lambda, \\ \lambda & \text{if} \quad \mathrm{p}_i > \lambda, \\ \mathrm{p}_i & \text{otherwise.} \end{cases}$$

(F)ISTA in the dual

- **Minimization problem**: $\widehat{\mathbf{x}} = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{x}-\mathbf{z}\|^2 + \|\Psi\mathbf{x}\|_1$
- **(F)ISTA to solve dual reformulation**:

Set $\mathrm{w}_1 \in \mathbb{R}^{|\mathbb{F}|}$, and $\mathrm{y}_1 \in \mathbb{R}^{|\mathbb{F}|}$. For every iteration k,

$$\left\lfloor \begin{aligned} \mathrm{w}_{k+1} &= \mathrm{HardTanh}_1\Big((\mathbf{I}-\tau_k\Psi\Psi^\top)\mathrm{y}_k + \tau_k\Psi\mathbf{z}\Big) \\ \mathrm{y}_{k+1} &= (1+\alpha_k)\mathrm{w}_{k+1} - \alpha_k\mathrm{w}_k \end{aligned}\right.$$

- **Unfolded (F)ISTA**: $\theta = \{\Psi_1^{[k]}, \Psi_2^{[k]}, \alpha_k, \}_{1\leqslant k\leqslant K}$

$$\begin{bmatrix}\mathrm{w}^{[k]}\\ \mathrm{w}^{[k+1]}\end{bmatrix} = \underbrace{\begin{Bmatrix}\mathbf{I}_{|\mathbb{F}|}\\ \mathrm{HardTanh}_1\end{Bmatrix}}_{\eta^{[k]}}\Bigg(\underbrace{\begin{bmatrix}0 & \mathbf{I}_{|\mathbb{F}|}\\ -\alpha_{k-1}(\mathbf{I}_{|\mathbb{F}|}-\Psi_1^{[k]}\Psi_2^{[k]}) & (1+\alpha_{k-1})(\mathbf{I}_{|\mathbb{F}|}-\Psi_1^{[k]}\Psi_2^{[k]})\end{bmatrix}}_{\mathrm{W}^{[k]}}\begin{bmatrix}\mathrm{w}^{[k-1]}\\ \mathrm{w}^{[k]}\end{bmatrix} + \underbrace{\begin{bmatrix}0\\ \Psi_1^{[k]}\mathbf{z}_l\end{bmatrix}}_{\mathrm{b}^{[k]}}$$

Network Deep-(F)ISTA-GD

➡ **Network**: For every layer $k \in \{2, \ldots, K-1\}$:

$$\begin{cases} W^{[1]} = \begin{bmatrix} \Psi_1^{[1]} \\ (\mathbf{I}_{|\mathbb{F}|} - \Psi_1^{[1]}\Psi_2^{[1]})\Psi_1^{[1]} \end{bmatrix}, \\ b^{[1]} = \begin{bmatrix} 0 \\ \Psi_1^{[1]}\mathbf{z}_l \end{bmatrix}, \eta^{[1]} = \begin{Bmatrix} \mathbf{I}_{|\mathbb{F}|} \\ \mathrm{HardTanh}_\lambda \end{Bmatrix}, \\ W^{[k]} = \begin{bmatrix} 0 & \mathbf{I}_{|\mathbb{F}|} \\ -\alpha_{k-1}(\mathbf{I}_{|\mathbb{F}|} - \Psi_1^{[k]}\Psi_2^{[k]}) & (1+\alpha_{k-1})(\mathbf{I}_{|\mathbb{F}|} - \Psi_1^{[k]}\Psi_2^{[k]}) \end{bmatrix}, \\ b^{[k]} = \begin{bmatrix} 0 \\ \Psi_1^{[k]}\mathbf{z}_l \end{bmatrix}, \eta^{[k]} = \begin{Bmatrix} \mathbf{I}_{|\mathbb{F}|} \\ \mathrm{HardTanh}_\lambda \end{Bmatrix}, \\ W^{[K]} = \begin{bmatrix} 0 & -\Psi_2^{[k]} \end{bmatrix}, b^{[K]} = \mathbf{z}_l, \eta^{[K]} = \mathbf{I}_N \,. \end{cases}$$

➡ **Proposition**: If $\Psi_1^{[k]} = \tau_k \Psi$ and $\Psi_2^{[k]} = \Psi^\top$, then `Deep-(F)ISTA-GD` network fits the generic `(F)ISTA` scheme.

(Sc)CP

➡ **Minimization problem**: $\widehat{\mathbf{x}} = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{x} - \mathbf{z}\|^2 + \|\Psi\mathbf{x}\|_1$

➡ **(Sc)CP to solve the minimization problem**:

Set $\mathrm{w}_1 \in \mathbb{R}^{|\mathbb{F}|}$, and $\mathbf{x}_1 = \mathbf{x}_0 \in \mathbb{R}^{|\mathbb{F}|}$. For every iteration k,

$$\left\lfloor \begin{array}{ll} \mathrm{w}_{k+1} & = \mathrm{prox}_{\iota_{\|\cdot\|_\infty \leqslant 1}}\Big(\mathrm{w}_k + \tau_k \Psi\Big((1+\alpha_k)\,\mathbf{x}_k - \alpha_k \mathrm{x}_{k-1}\Big)\Big) \\ \mathbf{x}_{k+1} & = \mathrm{prox}_{\frac{\sigma_k}{2}\|\cdot - \mathbf{z}\|_2^2}\Big(\mathbf{x}_k - \sigma_k \Psi^\top \mathrm{w}_{k+1}\Big) \end{array} \right.$$

➡ **Remarks** :

- ScCP: $\alpha_k = \frac{1}{\sqrt{1+2\gamma\sigma_k}}$, $\sigma_{k+1} = \alpha_k \sigma_k$, $\tau_{k+1} = \frac{\tau_k}{\alpha_k}$.
- CP: $\gamma = 0$, $\sigma_k \equiv \sigma$, $\tau_k \equiv \tau$ and assuming $\sigma\tau\|D\|^2 < 1$.
- $(\mathbf{x}_k)_{k\in\mathbb{N}}$ converges to $\widehat{\mathbf{x}}$.
- Convergence rate $O(1/k)$ for CP and $O(1/k^2)$ for ScCP.

(Sc)CP

➡ **Minimization problem**: $\widehat{\mathbf{x}} = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{x} - \mathbf{z}\|^2 + \|\Psi \mathbf{x}\|_1$

➡ **(Sc)CP to solve the minimization problem**:

Set $\mathrm{w}_1 \in \mathbb{R}^{|\mathbb{F}|}$, and $\mathbf{x}_1 = \mathbf{x}_0 \in \mathbb{R}^{|\mathbb{F}|}$. For every iteration k,

$$\left\lfloor \begin{array}{ll} \mathrm{w}_{k+1} & = \mathrm{HardTanh}_1\Big(\mathrm{w}_k + \tau_k \Psi\Big(\left(1+\alpha_k\right)\mathbf{x}_k - \alpha_k \mathrm{x}_{k-1}\Big)\Big) \\ \mathbf{x}_{k+1} & = \frac{\sigma_k}{1+\sigma_k}\mathbf{z} + \frac{1}{1+\sigma_k}\mathbf{x}_k - \frac{\sigma_k}{1+\sigma_k}\Psi^\top \mathrm{w}_{k+1} \end{array} \right.$$

➡ **Remarks** :

- ScCP: $\alpha_k = \frac{1}{\sqrt{1+2\gamma\sigma_k}}$, $\sigma_{k+1} = \alpha_k \sigma_k$, $\tau_{k+1} = \frac{\tau_k}{\alpha_k}$.
- CP: $\gamma = 0$, $\sigma_k \equiv \sigma$, $\tau_k \equiv \tau$ and assuming $\sigma\tau\|D\|^2 < 1$.
- $(\mathbf{x}_k)_{k\in\mathbb{N}}$ converges to $\widehat{\mathbf{x}}$.
- Convergence rate $O(1/k)$ for CP and $O(1/k^2)$ for ScCP.

(Sc)CP

➡ **Minimization problem**: $\widehat{\mathbf{x}} = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{x}-\mathbf{z}\|^2 + \|\Psi\mathbf{x}\|_1$

➡ **(Sc)CP to solve the minimization problem**:

Set $\mathrm{w}_1 \in \mathbb{R}^{|\mathbb{F}|}$, and $\mathbf{x}_1 = \mathbf{x}_0 \in \mathbb{R}^{|\mathbb{F}|}$. For every iteration k,

$$\left\lfloor \begin{aligned} \mathrm{w}_{k+1} &= \mathrm{HardTanh}_1\Big(\mathrm{w}_k + \tau_k \Psi\big((1+\alpha_k)\mathbf{x}_k - \alpha_k \mathrm{x}_{k-1}\big)\Big) \\ \mathbf{x}_{k+1} &= \tfrac{\sigma_k}{1+\sigma_k}\mathbf{z} + \tfrac{1}{1+\sigma_k}\mathbf{x}_k - \tfrac{\sigma_k}{1+\sigma_k}\Psi^\top \mathrm{w}_{k+1} \end{aligned}\right.$$

➡ **Unfolded (Sc)CP**: $\boldsymbol{\theta} = \{\Psi_1^{[k]}, \Psi_2^{[k]}, \sigma_k, \alpha_k, \}_{1\leqslant k\leqslant K}$

$$\begin{bmatrix}\mathbf{x}_k \\ \mathrm{w}_{k+1}\end{bmatrix} = \underbrace{\begin{Bmatrix}\mathbf{I}_N \\ \mathrm{HTanh}_1\end{Bmatrix}}_{\eta^{[k]}}\Bigg(\underbrace{\begin{bmatrix}\frac{1}{1+\sigma_{k-1}} & -\frac{\sigma_{k-1}}{1+\sigma_{k-1}}\Psi_2^{[k-1]} \\ (\frac{1+\alpha_k}{1+\sigma_{k-1}} - \alpha_k)\Psi_1^{[k]} & \mathbf{I}_{|\mathbb{F}|} - \frac{(1+\alpha_k)\sigma_{k-1}}{1+\sigma_{k-1}}\Psi_1^{[k]}\Psi_2^{[k-1]}\end{bmatrix}}_{\mathrm{W}^{[k]}}\begin{bmatrix}\mathbf{x}_{k-1} \\ \mathrm{w}_k\end{bmatrix} + \underbrace{\begin{bmatrix}\frac{\sigma_{k-1}}{1+\sigma_{k-1}}\mathbf{z} \\ \frac{(1+\alpha_k)\sigma_{k-1}}{1+\sigma_{k-1}}\Psi_1^{[k]}\mathbf{z}\end{bmatrix}}_{\mathrm{b}^{[k]}}\Bigg)$$

Network Deep-(Sc)CP-GD

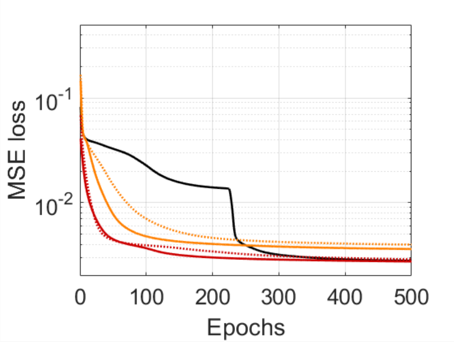
➡ **Network**: For every layer $k \in \{2, \dots, K-1\}$:

$$\begin{cases} W^{[1]} = \begin{bmatrix} \mathbf{I}_N \\ 2D_1^{[1]} \end{bmatrix}, b^{[1]} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \eta^{[1]} = \begin{Bmatrix} \mathbf{I}_N \\ \mathrm{HardTanh}_\lambda \end{Bmatrix}, \\ W^{[k]} = \begin{bmatrix} \frac{1}{1+\sigma_{k-1}} & -\frac{\sigma_{k-1}}{1+\sigma_{k-1}} \Psi_2^{[k-1]} \\ \frac{1+\alpha_k}{1+\sigma_{k-1}} \Psi_1^{[k]} - \alpha_k \Psi_1^{[k]} & \mathbf{I}_{|\mathbb{F}|} - \frac{(1+\alpha_k)\sigma_{k-1}}{1+\sigma_{k-1}} \Psi_1^{[k]} \Psi_2^{[k-1]} \end{bmatrix}, \\ b^{[k]} = \begin{bmatrix} \frac{\sigma_{k-1}}{1+\sigma_{k-1}} \mathbf{z} \\ \frac{(1+\alpha_k)\sigma_{k-1}}{1+\sigma_{k-1}} \Psi_1^{[k]} \mathbf{z} \end{bmatrix}, \eta^{[k]} = \begin{Bmatrix} \mathbf{I}_N \\ \mathrm{HardTanh}_\lambda \end{Bmatrix}, \\ W^{[K]} = \begin{bmatrix} \mathbf{I}_N & 0 \end{bmatrix}, b^{[K]} = 0, \eta^{[K]} = \mathbf{I}_N \,. \end{cases}$$

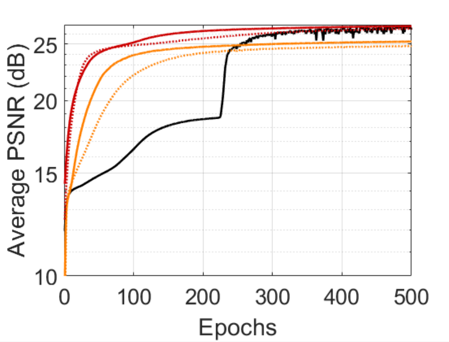
➡ **Proposition**: If $\Psi_1^{[k]} = \tau_k \Psi$ and $\Psi_2^{[k]} = \Psi^\top$, then the `Deep-(Sc)CP-GD` network fits the generic `(Sc)CP` scheme.

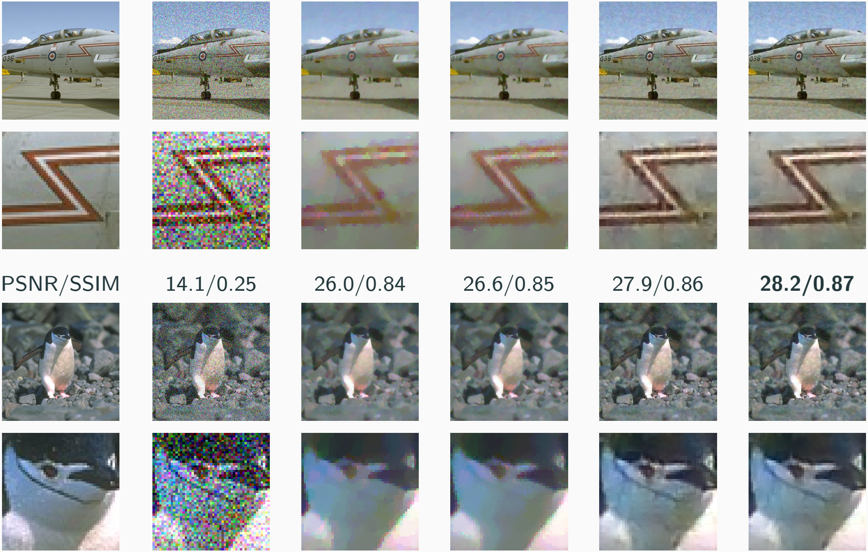
Performance Gaussian image denoising

Training loss



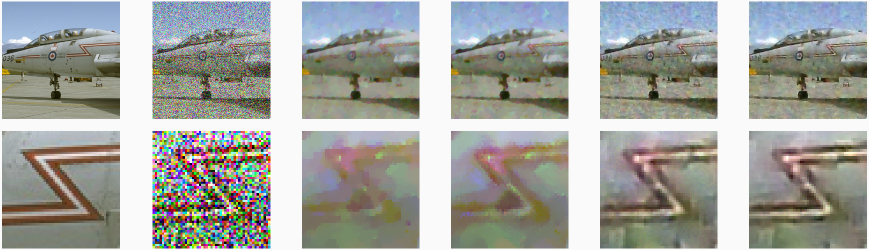
PSNR on test dataset



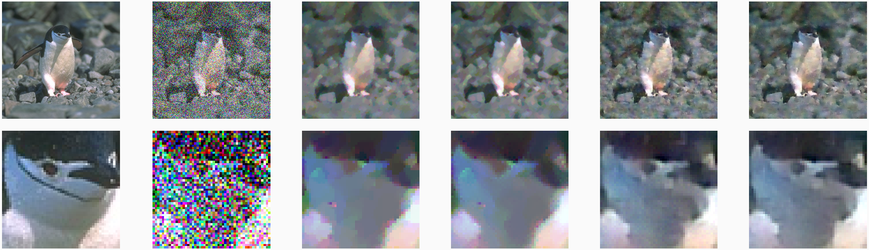


Original	Noisy	TV	NL-TV	DnCNN	Proposed
PSNR/SSIM	14.1/0.25	26.0/0.84	26.6/0.85	27.9/0.86	**28.2/0.87**
PSNR/SSIM	14.1/0.13	26.0/0.76	27.7/0.79	28.5/0.79	**28.8/0.81**

Original	Noisy	TV	NL-TV	DnCNN	Proposed



PSNR/SSIM	8.13/0.09	23.6/0.76	24.0/0.76	24.4/0.76	**25.2/0.80**

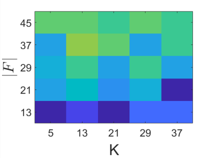


PSNR/SSIM	8.14/0.043	24.5/0.64	25.1/0.65	25.4/0.65	**25.9/0.70**

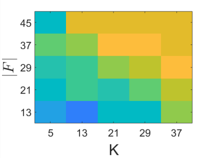
Architecture comparisons for denoising

☛ **SNR**

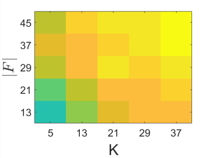
Deep-ISTA-GD



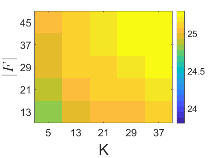
Deep-FISTA-GD



Deep-CP-GD

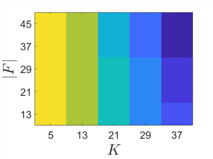


Deep-ScCP-GD

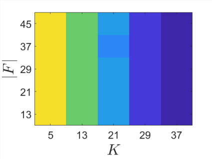


☛ **Robustness**: $\|f_\Theta(\mathbf{z}+\epsilon) - f_\Theta(\mathbf{z})\| \leqslant \chi\|\epsilon\|.$

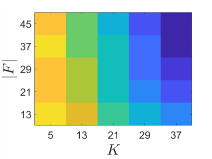
Deep-ISTA-GD



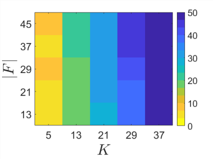
Deep-FISTA-GD



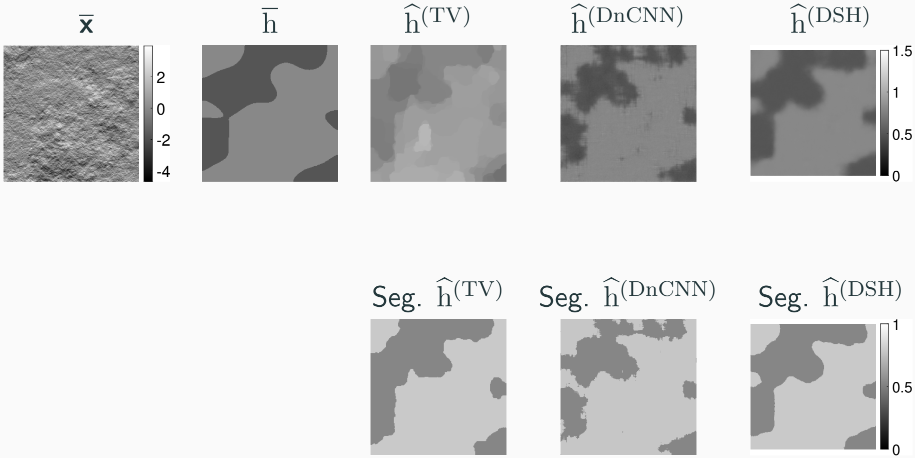
Deep-CP-GD



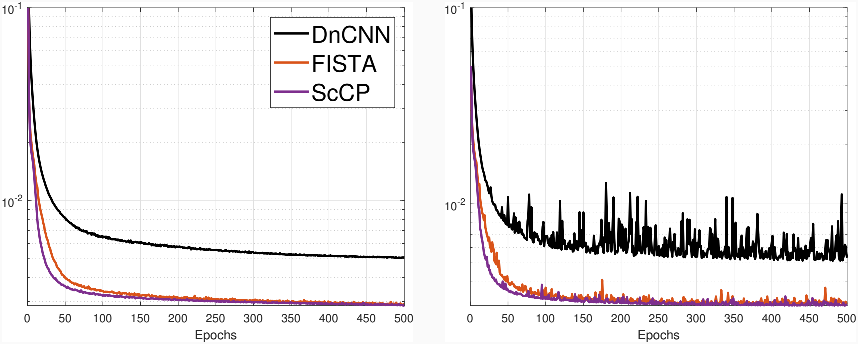
Deep-ScCP-GD



Performance texture segmentation



Performance texture segmentation



(i) (ii)

Conclusions

- Unfolded architecture allowing to go from standard variational approached to deep learning architectures depending on the learned $\boldsymbol{\theta}$:
  - $\boldsymbol{\theta} = \lambda$
  - $\boldsymbol{\theta} = (\sigma, \tau)$
  - $\boldsymbol{\theta} = (\sigma_k, \tau_k)$
  - $\boldsymbol{\theta} = (\sigma_k, \tau_k, \mathrm{H}_k)$

- Relation between activation functions and proximal operators.
- Unfolded primal-dual proximal algorithms provide comparable or better performance than state-of-the-art deep learning methods in image restoration.
- Faster schemes lead to better performance. ScCP appears to be the "best" unfolded architecture.

Collaborations and references

- M Jiu, N Pustelnik, A deep primal-dual proximal network for image restoration IEEE Journal of Selected Topics in Signal Processing 15 (2), 190-203, 2021.

- M. Jiu, N. Pustelnik, Alternative Design of DeepPDNet in the Context of Image Restoration, IEEE Signal Processing Letters, vol. 29, pp. 932 - 936, 2022.

- H. Le, N. Pustelnik, M. Foare, The faster proximal algorithm, the better unfolded deep learning architecture? The study case of image denoising. EUSIPCO, 2022.

- L. Briceño-Arias, N. Pustelnik, Theoretical and numerical comparison of first-order algorithms for cocoercive equations and smooth convex optimization, arXiv:2101.0615, 2022.