Deep Unfolded Proximal Interior Point Algorithm for Image Restoration

C. Bertocchi[1], **E. Chouzenoux**[2], M.-C. Corbineau[2], J.-C. Pesquet[2], M. Prato[1]

[1]Università di Modena e Reggio Emilia, Modena, Italy
[2]CVN, CentraleSupélec, Université Paris-Saclay, France

5 February 2019
Mathematics of Imaging, IHP, Paris

Motivation

Inverse problem in imaging

$$y = \mathcal{D}(H\overline{x})$$

where $y \in \mathbb{R}^m$ observed image, $\mathcal{D}$ degradation model, $H \in \mathbb{R}^{m\times n}$ linear observation model, $\overline{x} \in \mathbb{R}^n$ original image

Motivation

Inverse problem in imaging

$$y = \mathcal{D}(H\overline{x})$$

where $y \in \mathbb{R}^m$ observed image, $\mathcal{D}$ degradation model, $H \in \mathbb{R}^{m \times n}$ linear observation model, $\overline{x} \in \mathbb{R}^n$ original image

Variational methods

$$\underset{x \in \mathcal{C}}{\text{minimize}} \quad f(Hx, y) + \lambda \mathcal{R}(x)$$

where $f : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ data-fitting term, $\mathcal{R} : \mathbb{R}^n \to \mathbb{R}$ regularization function, $\lambda > 0$ regularization weight

- ✓ Incorporate prior knowledge about solution and enforce desirable constraints
- ✗ No closed-form solution $\rightarrow$ advanced algorithms
- ✗ Estimation of λ and tuning of algorithm parameters $\rightarrow$ time-consuming

Motivation

Inverse problem in imaging

$$y = \mathcal{D}(H\overline{x})$$

where $y \in \mathbb{R}^m$ observed image, $\mathcal{D}$ degradation model, $H \in \mathbb{R}^{m \times n}$ linear observation model, $\overline{x} \in \mathbb{R}^n$ original image

Variational methods

$$\underset{x \in \mathcal{C}}{\text{minimize}} \quad f(Hx, y) + \lambda \mathcal{R}(x)$$

where $f : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ data-fitting term, $\mathcal{R} : \mathbb{R}^n \to \mathbb{R}$ regularization function, $\lambda > 0$ regularization weight

- ✓ Incorporate prior knowledge about solution and enforce desirable constraints
- ✗ No closed-form solution $\rightarrow$ advanced algorithms
- ✗ Estimation of λ and tuning of algorithm parameters $\rightarrow$ time-consuming

Deep-learning methods

- ✓ Generic and very efficient architectures
- ✗ Post-processing step : solve optimization problem $\rightarrow$ estimate regularization parameter
- ✗ Black-box, no theoretical guarantees

Motivation

Inverse problem in imaging

$$y = \mathcal{D}(H\overline{x})$$

where $y \in \mathbb{R}^m$ observed image, $\mathcal{D}$ degradation model, $H \in \mathbb{R}^{m \times n}$ linear observation model, $\overline{x} \in \mathbb{R}^n$ original image

Variational methods

$$\underset{x \in \mathcal{C}}{\text{minimize}} \quad f(Hx, y) + \lambda \mathcal{R}(x)$$

where $f : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ data-fitting term, $\mathcal{R} : \mathbb{R}^n \to \mathbb{R}$ regularization function, $\lambda > 0$ regularization weight

- ✓ Incorporate prior knowledge about solution and enforce desirable constraints
- ✗ No closed-form solution $\to$ advanced algorithms
- ✗ Estimation of λ and tuning of algorithm parameters $\to$ time-consuming

Deep-learning methods

- ✓ Generic and very efficient architectures
- ✗ Post-processing step : solve optimization problem $\to$ estimate regularization parameter
- ✗ Black-box, no theoretical guarantees

$\to$ Combine benefits of both approaches : unfold proximal interior point algorithm

Notation and Assumptions

Proximity operator

Let $\Gamma_0(\mathbb{R}^n)$ be the set of proper lsc convex functions from $\mathbb{R}^n$ to $\mathbb{R} \cup \{+\infty\}$. The **proximal operator** [http://proximity-operator.net/] of $g \in \Gamma_0(\mathbb{R}^n)$ at $x \in \mathbb{R}^n$ is uniquely defined as

$$\mathrm{prox}_g(x) = \underset{z \in \mathbb{R}^n}{\operatorname{argmin}} \left(g(z) + \frac{1}{2}\|z - x\|^2 \right).$$

Assumptions

$$\mathcal{P}_0 : \quad \underset{x \in \mathcal{C}}{\text{minimize}} \quad f(Hx, y) + \lambda \mathcal{R}(x)$$

We assume that $f(\cdot, y)$ and $\mathcal{R}$ are twice-differentiable, $f(H\cdot, y) + \lambda\mathcal{R} \in \Gamma_0(\mathbb{R}^n)$ is either coercive or $\mathcal{C}$ is bounded. The feasible set is defined as

$$\mathcal{C} = \{x \in \mathbb{R}^n \mid (\forall i \in \{1, \dots, p\}) \;\; c_i(x) \geq 0\}$$

where $(\forall i \in \{1, \dots, p\})$, $-c_i \in \Gamma_0(\mathbb{R}^n)$. The strict interior of the feasible set is nonempty.

- Existence of a solution to $\mathcal{P}_0$
- Twice-differentiability : training using gradient descent

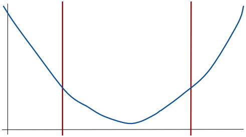
$\mathcal{B}$: logarithmic barrier

$$(\forall x \in \mathbb{R}^n) \quad \mathcal{B}(x) = \begin{cases} -\sum_{i=1}^{p} \ln(c_i(x)) & \text{if } x \in \mathrm{int}\mathcal{C} \\ +\infty & \text{otherwise.} \end{cases}$$

Logarithmic barrier method

Constrained Problem

$$\mathcal{P}_0 : \quad \underset{x \in \mathcal{C}}{\text{minimize}} \quad f(Hx, y) + \lambda \mathcal{R}(x)$$



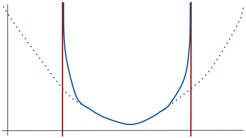
Logarithmic barrier method

Constrained Problem $\qquad \mathcal{P}_0 : \quad \underset{x \in \mathcal{C}}{\text{minimize}} \quad f(Hx, y) + \lambda \mathcal{R}(x)$

$$\Downarrow$$

Unconstrained Subproblem $\qquad \mathcal{P}_\mu : \quad \underset{x \in \mathbb{R}^n}{\text{minimize}} \ f(Hx, y) + \lambda \mathcal{R}(x) + \boldsymbol{\mu \mathcal{B}(x)}$

where $\mu > 0$ is the barrier parameter.



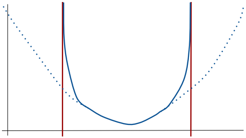
Logarithmic barrier method

Constrained Problem $\qquad \mathcal{P}_0 : \quad \underset{x \in \mathcal{C}}{\text{minimize}} \quad f(Hx, y) + \lambda \mathcal{R}(x)$

$$\Downarrow$$

Unconstrained Subproblem $\qquad \mathcal{P}_\mu : \quad \underset{x \in \mathbb{R}^n}{\text{minimize}} \; f(Hx, y) + \lambda \mathcal{R}(x) + \boldsymbol{\mu \mathcal{B}(x)}$

where $\mu > 0$ is the barrier parameter.



$\mathcal{P}_0$ is replaced by a sequence of subproblems $(\mathcal{P}_{\mu_j})_{j \in \mathbb{N}}$.

- Subproblems solved approximately for a sequence $\mu_j \to 0$
- Main advantages : feasible iterates, superlinear convergence for NLP
- ✗ Inversion of an $n \times n$ matrix at each step

Proximal interior point strategy

→ Combine interior point method with proximity operator

Exact version of the proximal IPM in **[Kaplan and Tichatschke, 1998]**.

Let $x_0 \in \operatorname{int}\mathcal{C}$, $\underline{\gamma} > 0$, $(\forall k \in \mathbb{N})$ $\underline{\gamma} \leq \gamma_k$ and $\mu_k \to 0$;
for $k = 0, 1, \ldots$ **do**
$x_{k+1} = \operatorname{prox}_{\gamma_k(f(H\cdot,y)+\lambda\mathcal{R}+\mu_k\mathcal{B})}(x_k)$
end for

✗ No closed-form solution for $\operatorname{prox}_{\gamma_k(f(H\cdot,y)+\lambda\mathcal{R}+\mu_k\mathcal{B})}$

Proximal interior point strategy

→ Combine interior point method with proximity operator

Exact version of the proximal IPM in **[Kaplan and Tichatschke, 1998]**.

Let $x_0 \in \operatorname{int}\mathcal{C}$, $\underline{\gamma} > 0$, $(\forall k \in \mathbb{N})$ $\underline{\gamma} \leq \gamma_k$ and $\mu_k \to 0$;
for $k = 0, 1, \ldots$ **do**
$\quad x_{k+1} = \operatorname{prox}_{\gamma_k (f(H\cdot, y) + \lambda\mathcal{R} + \mu_k \mathcal{B})}(x_k)$
end for

✗ No closed-form solution for $\operatorname{prox}_{\gamma_k (f(H\cdot, y) + \lambda\mathcal{R} + \mu_k \mathcal{B})}$

Proposed forward–backward proximal IPM.

Let $x_0 \in \operatorname{int}\mathcal{C}$, $\underline{\gamma} > 0$, $(\forall k \in \mathbb{N})$ $\underline{\gamma} \leq \gamma_k$ and $\mu_k \to 0$;
for $k = 0, 1, \ldots$ **do**
$\quad x_{k+1} = \operatorname{prox}_{\gamma_k \mu_k \mathcal{B}}\left(x_k - \gamma_k \left(H^\top \nabla_1 f(Hx_k, y) + \lambda \nabla \mathcal{R}(x_k)\right)\right)$
end for

✓ Only requires $\operatorname{prox}_{\gamma_k \mu_k \mathcal{B}}$

Proximity operator of the barrier

Affine constraints $\mathcal{C} = \left\{ x \in \mathbb{R}^n \mid a^\top x \leq b \right\}$

Proposition 1

Let $\varphi : (x, \alpha) \mapsto \operatorname{prox}_{\alpha \mathcal{B}}(\mathrm{x})$. Then, for every $(x, \alpha) \in \mathbb{R}^n \times \mathbb{R}_+^*$,

$$\varphi(x, \alpha) = x + \frac{b - a^\top x - \sqrt{(b - a^\top x)^2 + 4\alpha \|a\|^2}}{2\|a\|^2} a.$$

In addition, the Jacobian matrix of φ wrt x and the gradient of φ wrt α are given by

$$J_\varphi^{(x)}(x, \alpha) = \mathbb{I}_n - \frac{1}{2\|a\|^2} \left(1 + \frac{a^\top x - b}{\sqrt{(b - a^\top x)^2 + 4\alpha \|a\|^2}} \right) a a^\top$$

and

$$\nabla_\varphi^{(\alpha)}(x, \alpha) = \frac{-1}{\sqrt{(b - a^\top x)^2 + 4\alpha \|a\|^2}} a$$

Proof : **[Chaux *et al.*,2007]** and **[Bauschke and Combettes,2017]**

Proximity operator of the barrier

Hyperslab constraints $\quad \mathcal{C} = \left\{ x \in \mathbb{R}^n \mid b_m \leq a^\top x \leq b_M \right\}$

Proposition 2

Let $\varphi : (x, \alpha) \mapsto \mathrm{prox}_{\alpha\mathcal{B}}(\mathrm{x})$. Then, for every $(x, \alpha) \in \mathbb{R}^n \times \mathbb{R}_+^*$,

$$\varphi(x, \alpha) = x + \frac{\kappa(x, \alpha) - a^\top x}{\|a\|^2} a,$$

where $\kappa(x, \alpha)$ is the unique solution in $]b_m, b_M[$, of the following cubic equation,

$$0 = z^3 - (b_m + b_M + a^\top x) z^2 + (b_m b_M + a^\top x(b_m + b_M) - 2\alpha\|a\|^2) z - b_m b_M a^\top x + \alpha(b_m + b_M)\|a\|^2.$$

In addition, the Jacobian matrix of φ wrt x and the gradient of φ wrt α are given by

$$J_\varphi^{(x)}(x, \alpha) = \mathbb{I}_n - \frac{1}{\|a\|^2} \left(\frac{(b_M - \kappa(x, \alpha))(b_m - \kappa(x, \alpha))}{\eta(x, \alpha)} - 1 \right) a a^\top$$

and

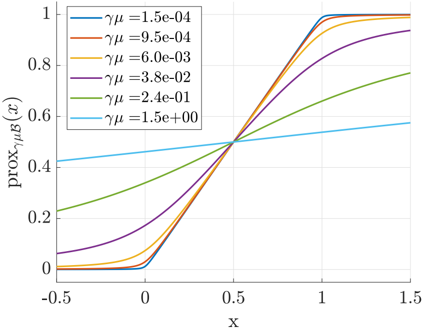
$$\nabla_\varphi^{(\alpha)}(x, \alpha) = \frac{2\kappa(x, \alpha) - b_m - b_M}{\eta(x, \alpha)} a,$$

where $\eta(x, \alpha) = (b_M - \kappa(x, \alpha))(b_m - \kappa(x, \alpha)) - (b_m + b_M - 2\kappa(x, \alpha))(\kappa(x, \alpha) - a^\top x) - 2\alpha\|a\|^2$.

Proof : **[Chaux *et al.*,2007]**, **[Bauschke and Combettes,2017]** and implicit function theorem

Proximity operator of the barrier

Bound constraints $\mathcal{C} = \{x \in \mathbb{R} \mid 0 \leq x \leq 1\}$



Proximity operator of the barrier

Bounded ℓ_2-norm $\quad \mathcal{C} = \left\{ x \in \mathbb{R}^n \mid \|x - c\|^2 \leq \rho \right\}$

Proposition 3

Let $\varphi : (x, \alpha) \mapsto \mathrm{prox}_{\alpha\mathcal{B}}(\mathrm{x})$. Then, for every $(x, \alpha) \in \mathbb{R}^n \times \mathbb{R}_+^*$,

$$\varphi(x, \alpha) = c + \frac{\rho - \kappa(x, \alpha)^2}{\rho - \kappa(x, \alpha)^2 + 2\alpha}(x - c),$$

where $\kappa(x, \alpha)$ is the unique solution in $]0, \sqrt{\rho}[$, of the following cubic equation,

$$0 = z^3 - \|x - c\|z^2 - (\rho + 2\alpha)z + \rho\|x - c\|.$$

In addition, the Jacobian matrix of φ wrt x and the gradient of φ wrt α are given by

$$J_\varphi^{(x)}(x, \alpha) = \frac{\rho - \|\varphi(x, \alpha) - c\|^2}{\rho - \|\varphi(x, \alpha) - c\|^2 + 2\alpha} M(x, \alpha)$$

and

$$\nabla_\varphi^{(\alpha)}(x, \alpha) = \frac{-2}{\rho - \|\varphi(x, \alpha) - c\|^2 + 2\alpha} M(x, \alpha)(\varphi(x, \alpha) - c),$$

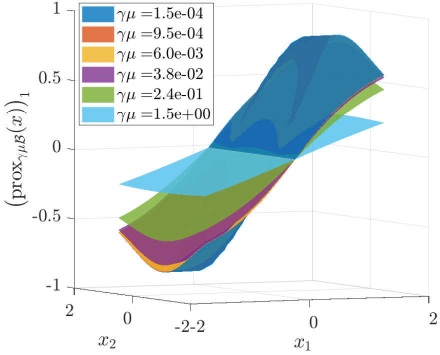
where

$$M(x, \alpha) = \mathbb{I}_n - \frac{2(x - \varphi(x, \alpha))(\varphi(x, \alpha) - c)^\top}{\rho - 3\|\varphi(x, \alpha) - c\|^2 + 2\alpha + 2(\varphi(x, \alpha) - c)^\top (x - c)}.$$

Proof : **[Bauschke and Combettes,2017]**, Sherma-Morrison lemma and implicit function theorem

Proximity operator of the barrier

Bounded ℓ_2-norm $\mathcal{C} = \left\{ x \in \mathbb{R}^2 \mid \|x\|^2 \leq 0.7 \right\}$



Proposed strategy

Forward–backward proximal IPM.

Let $x_0 \in \mathrm{int}\mathcal{C}$, $\underline{\gamma} > 0$, $(\forall k \in \mathbb{N})$ $\underline{\gamma} \leq \gamma_k$ and $\mu_k \to 0$;
for $k = 0, 1, \dots$ **do**

$$x_{k+1} = \mathrm{prox}_{\gamma_k \mu_k \mathcal{B}} \left(x_k - \gamma_k \left(H^\top \nabla_1 f(Hx_k, y) + \lambda \nabla \mathcal{R}(x_k) \right) \right)$$

end for

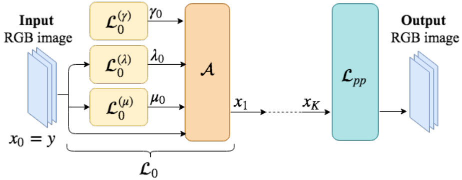
- ✓ Efficient algorithm for constrained optimization
- ✗ Setting of the parameters $(\mu_k, \gamma_k)_{k \in \mathbb{N}}$?
- ✗ Finding the regularization parameter λ so as to optimize the visual quality of the solution ?

→ Unfold proximal IP algorithm over K iterations, untie γ, μ and λ across network

$$\mathcal{A}(x_k, \mu_k, \gamma_k, \lambda_k) = \mathrm{prox}_{\gamma_k \mu_k \mathcal{B}} \left(x_k - \gamma_k \left(H^\top \nabla_1 f(Hx_k, y) + \lambda_k \nabla \mathcal{R}(x_k) \right) \right)$$

iRestNet architecture

$\longrightarrow$ **Unfold proximal IP algorithm over K iterations, untie γ, μ and λ across network**

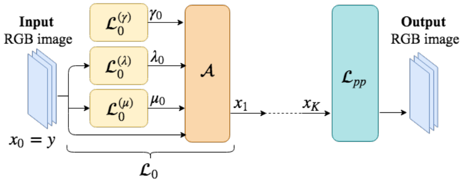


Input : $x_0 = y$ blurred image

Hidden structures

iRestNet architecture

$\longrightarrow$ **Unfold proximal IP algorithm over K iterations, untie γ, μ and λ across network**



Input : $x_0 = y$ blurred image

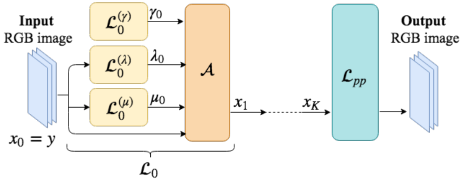
Hidden structures

- $(\mathcal{L}_k^{(\gamma)})_{0 \leq k \leq K-1}$: estimate stepsize, positive $\rightarrow$ Softplus (smooth approx ReLU)

$$\gamma_k = \mathcal{L}_k^{(\gamma)} = \mathrm{Softplus}(a_k)$$

iRestNet architecture

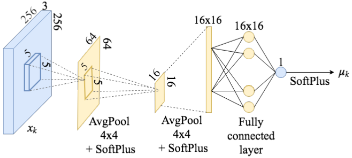
$\longrightarrow$ **Unfold proximal IP algorithm over K iterations, untie γ, μ and λ across network**



Input : $x_0 = y$ blurred image

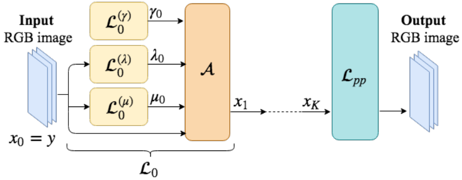
Hidden structures

- $(\mathcal{L}_k^{(\gamma)})_{0 \leq k \leq K-1}$: estimate stepsize
- $(\mathcal{L}_k^{(\mu)})_{0 \leq k \leq K-1}$: estimate barrier parameter



iRestNet architecture

$\longrightarrow$ **Unfold proximal IP algorithm over K iterations, untie γ, μ and λ across network**



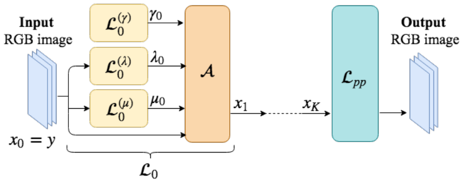
Input : $x_0 = y$ blurred image

Hidden structures

- $(\mathcal{L}_k^{(\gamma)})_{0 \leq k \leq K-1}$: estimate stepsize
- $(\mathcal{L}_k^{(\mu)})_{0 \leq k \leq K-1}$: estimate barrier parameter
- $(\mathcal{L}_k^{(\lambda)})_{0 \leq k \leq K-1}$: estimate regularization parameter $\rightarrow$ image statistics, noise level

iRestNet architecture

$\longrightarrow$ **Unfold proximal IP algorithm over K iterations, untie γ, μ and λ across network**



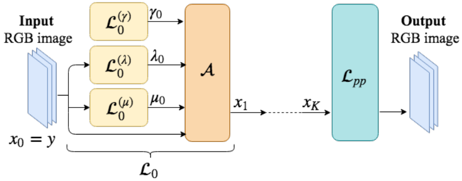
Input : $x_0 = y$ blurred image

Hidden structures

- $(\mathcal{L}_k^{(\gamma)})_{0\leq k\leq K-1}$: estimate stepsize
- $(\mathcal{L}_k^{(\mu)})_{0\leq k\leq K-1}$: estimate barrier parameter
- $(\mathcal{L}_k^{(\lambda)})_{0\leq k\leq K-1}$: estimate regularization parameter
- $\mathcal{A}(x_k, \mu_k, \gamma_k, \lambda_k) = \mathrm{prox}_{\gamma_k \mu_k \mathcal{B}}\left(x_k - \gamma_k \left(H^\top \nabla_1 f(Hx_k, y) + \lambda_k \nabla \mathcal{R}(x_k)\right)\right)$

iRestNet architecture

$\longrightarrow$ **Unfold proximal IP algorithm over K iterations, untie γ, μ and λ across network**



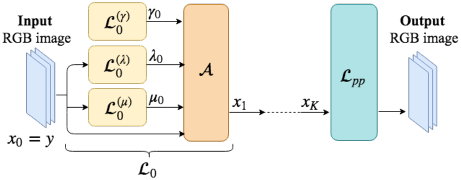
Input : $x_0 = y$ blurred image

Hidden structures

- $(\mathcal{L}_k^{(\gamma)})_{0 \le k \le K-1}$: estimate stepsize
- $(\mathcal{L}_k^{(\mu)})_{0 \le k \le K-1}$: estimate barrier parameter
- $(\mathcal{L}_k^{(\lambda)})_{0 \le k \le K-1}$: estimate regularization parameter
- $\mathcal{A}(x_k, \mu_k, \gamma_k, \lambda_k) = \mathrm{prox}_{\gamma_k \mu_k \mathcal{B}} \left(x_k - \gamma_k \left(H^\top \nabla_1 f(Hx_k, y) + \lambda_k \nabla \mathcal{R}(x_k) \right) \right)$
- $\mathcal{L}_{\mathrm{pp}}$: post-processing layer $\rightarrow$ e.g. removes small artifacts

iRestNet architecture

$\longrightarrow$ **Unfold proximal IP algorithm over K iterations, untie γ, μ and λ across network**



Input : $x_0 = y$ blurred image

Hidden structures

- $(\mathcal{L}_k^{(\gamma)})_{0\leq k\leq K-1}$: estimate stepsize
- $(\mathcal{L}_k^{(\mu)})_{0\leq k\leq K-1}$: estimate barrier parameter
- $(\mathcal{L}_k^{(\lambda)})_{0\leq k\leq K-1}$: estimate regularization parameter
- $\mathcal{A}(x_k,\mu_k,\gamma_k,\lambda_k) = \mathrm{prox}_{\gamma_k\mu_k\mathcal{B}}\left(x_k - \gamma_k\left(H^\top\nabla_1 f(Hx_k,y) + \lambda_k\nabla\mathcal{R}(x_k)\right)\right)$
- $\mathcal{L}_{\mathrm{pp}}$: post-processing layer $\rightarrow$ e.g. removes small artifacts

Training Gradient descent and backpropagation ($\nabla\mathcal{A}$ with Propositions 1-3)

Network stability

What about the network performance when the input is perturbed ?

Network stability

What about the network performance when the input is perturbed ?

- Deep learning : lack of theoretical guarantees, e.g. AlexNet **[Szegedy *et al.*, 2013]**
- Applications with high risk and legal responsibility (medical image processing, defense, etc...) $\rightarrow$ need guarantees
- Recent work of **[Combettes and Pesquet, 2018]**
- Robustness addressed with the framework of **averaged operators**

Averaged operators

Definition – Nonexpansiveness

Let $T : \mathbb{R}^n \to \mathbb{R}^n$. Then, T is nonexpansive if it is 1-Lipschitz continuous, i.e.,

$$(\forall x \in \mathbb{R}^n)(\forall y \in \mathbb{R}^n) \quad \|T(x) - T(y)\| \leq \|x - y\|.$$

Definition – α-averaged operator

Let $T : \mathbb{R}^n \to \mathbb{R}^n$ be nonexpansive, and let $\alpha \in [0, 1]$. Then, T is α-averaged if there exists a nonexpansive operator $R : \mathbb{R}^n \to \mathbb{R}^n$ such that $T = (1 - \alpha)I_n + \alpha R$.

Averaged operators

Definition – Nonexpansiveness

Let $T : \mathbb{R}^n \to \mathbb{R}^n$. Then, T is nonexpansive if it is 1-Lipschitz continuous, i.e.,

$$(\forall x \in \mathbb{R}^n)(\forall y \in \mathbb{R}^n) \quad \|T(x) - T(y)\| \leq \|x - y\|.$$

Definition – α-averaged operator

Let $T : \mathbb{R}^n \to \mathbb{R}^n$ be nonexpansive, and let $\alpha \in [0,1]$. Then, T is α-averaged if there exists a nonexpansive operator $R : \mathbb{R}^n \to \mathbb{R}^n$ such that $T = (1-\alpha)I_n + \alpha R$.

- If T is averaged, then it is nonexpansive.
- Let $\alpha \in]0,1]$. T is α-averaged if and only if for every $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^n$,

$$\|T(x) - T(y)\|^2 \leq \|x - y\|^2 - \frac{1-\alpha}{\alpha}\|(I_n - T)(x) - (I_n - T)(y)\|^2.$$

$\Longrightarrow$ Bound on the output variation when input is perturbed.

Relation to generic deep neural networks

Feedforward architecture $R_{K-1} \circ (W_{K-1} \cdot + b_{K-1}) \circ \cdots \circ R_0 \circ (W_0 \cdot + b_0)$

- $(R_k)_{0 \le k \le K-1}$ non linear activation functions
- $(W_k)_{0 \le k \le K-1}$ weight operators
- $(b_k)_{0 \le k \le K-1}$ bias parameters

$\rightarrow$ **iRestNet shares same structure**

Relation to generic deep neural networks

Feedforward architecture $R_{K-1} \circ (W_{K-1} \cdot + b_{K-1}) \circ \cdots \circ R_0 \circ (W_0 \cdot + b_0)$

Quadratic problem $\underset{x \in \mathcal{C}}{\text{minimize}} \ \frac{1}{2}\|Hx - y\|^2 + \frac{\lambda}{2}\|Dx\|^2$

$$
\begin{aligned}
x_{k+1} &= \mathrm{prox}_{\gamma_k \mu_k \mathcal{B}}(x_k - \gamma_k(H^\top(Hx_k - y) + \lambda_k D^\top D x_k)) \\
&= \mathrm{prox}_{\gamma_k \mu_k \mathcal{B}}\left([\mathbb{I}_n - \gamma_k(H^\top H + \lambda_k D^\top D)]x_k + \gamma_k H^\top y\right) \\
&= R_k(W_k x_k + b_k)
\end{aligned}
$$

- $W_k = \mathbb{I}_n - \gamma_k(H^\top H + \lambda D^\top D)$ weight operator
- $b_k = \gamma_k H^\top y$ bias parameter
- $R_k = \mathrm{prox}_{\gamma_k \mu_k \mathcal{B}}$

Standard activation functions (ReLU, Sigmoid, etc...) are derived from a proximity operator **[Combettes and Pesquet, 2018]**.

$\rightarrow$ R_k **specific activation function**

Network stability result

Assumptions

Consider the quadratic problem, assume that $H^\top H$ and $D^\top D$ are **diagonalizable in the same basis** $\mathcal{P}$. For every $p \in \{1, \dots, n\}$ let $\beta_H^{(p)}$ and $\beta_D^{(p)}$ denote the p^{th} eigenvalue of $H^\top H$ and $D^\top D$ in $\mathcal{P}$, resp. Let β_+ and β_- be defined by

$$\beta_+ = \max_{1 \le p \le n} \prod_{k=0}^{K-1} \left(1 - \gamma_k \left(\beta_H^{(p)} + \lambda_k \beta_D^{(p)}\right)\right) \text{ and } \beta_- = \min_{1 \le p \le n} \prod_{k=0}^{K-1} \left(1 - \gamma_k \left(\beta_H^{(p)} + \lambda_k \beta_D^{(p)}\right)\right).$$

Let $\theta_{-1} = 1$ and for all $k \in \{0, \dots, K-1\}$,

$$\theta_k = \sum_{l=0}^{k} \theta_{l-1} \max_{1 \le q_l \le n} \left| \left(1 - \gamma_k \left(\beta_H^{(q_l)} + \lambda_k \beta_D^{(q_l)}\right)\right) \dots \left(1 - \gamma_l \left(\beta_H^{(q_l)} + \lambda_l \beta_D^{(q_l)}\right)\right) \right|.$$

Network stability result

Assumptions

Consider the quadratic problem, assume that $H^\top H$ and $D^\top D$ are **diagonalizable in the same basis** $\mathcal{P}$. For every $p \in \{1, \dots, n\}$ let $\beta_H^{(p)}$ and $\beta_D^{(p)}$ denote the p^{th} eigenvalue of $H^\top H$ and $D^\top D$ in $\mathcal{P}$, resp. Let β_+ and β_- be defined by

$$\beta_+ = \max_{1 \le p \le n} \prod_{k=0}^{K-1} \left(1 - \gamma_k \left(\beta_H^{(p)} + \lambda_k \beta_D^{(p)}\right)\right) \text{ and } \beta_- = \min_{1 \le p \le n} \prod_{k=0}^{K-1} \left(1 - \gamma_k \left(\beta_H^{(p)} + \lambda_k \beta_D^{(p)}\right)\right).$$

Let $\theta_{-1} = 1$ and for all $k \in \{0, \dots, K-1\}$,

$$\theta_k = \sum_{l=0}^{k} \theta_{l-1} \max_{1 \le q_l \le n} \left| \left(1 - \gamma_k \left(\beta_H^{(q_l)} + \lambda_k \beta_D^{(q_l)}\right)\right) \dots \left(1 - \gamma_l \left(\beta_H^{(q_l)} + \lambda_l \beta_D^{(q_l)}\right)\right) \right|.$$

Theorem 1 – α-averaged operator

Let $\alpha \in [1/2, 1]$. If one of the following conditions is satisfied

(i) $\beta_+ + \beta_- \le 0$ and $\theta_{K-1} \le 2^{K-1}(2\alpha - 1)$;

(ii) $0 \le \beta_+ + \beta_- \le 2^{K+1}(1-\alpha)$ and $2\theta_{K-1} \le \beta_+ + \beta_- + 2^K(2\alpha - 1)$;

(iii) $2^{K+1}(1-\alpha) \le \beta_+ + \beta_-$ and $\theta_{K-1} \le 2^{K-1}$,

then the operator $R_{K-1} \circ (W_{K-1} \cdot + b_{K-1}) \circ \dots \circ R_0 \circ (W_0 \cdot + b_0)$ is α-**averaged**.

$\Longrightarrow$ Bound on the output variation when input is perturbed.

Numerical experiments

Image deblurring $y = H\overline{x} + \omega$

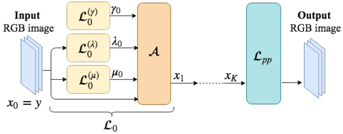
- $H \in \mathbb{R}^n \times \mathbb{R}^n$: circular convolution with known blur
- $\omega \in \mathbb{R}^n$: additive white Gaussian noise with standard deviation σ
- $y \in \mathbb{R}^n$, $\overline{x} \in \mathbb{R}^n$: RGB images

Variational formulation

$$\underset{x\in[0,x_{\max}]^n}{\text{minimize}} \quad \frac{1}{2}\|Hx - y\|^2 + \lambda \sum_{i=1}^{n} \sqrt{\frac{(D_{\mathrm{h}}x)_i^2 + (D_{\mathrm{v}}x)_i^2}{\delta^2} + 1}$$

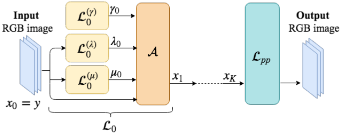
- δ : smoothing parameter, $\delta = 0.01$ for iRestNet
- $D_{\mathrm{h}} \in \mathbb{R}^{n\times n}$, $D_{\mathrm{v}} \in \mathbb{R}^{n\times n}$: horizontal and vertical spatial gradient operators

Network characteristics



- Number of layers : $K = 40$

Network characteristics



- Number of layers : $K = 40$
- **Estimation of regularization parameter**

$$\lambda_k = \mathcal{L}_k^{(\lambda)}(x_k) = \frac{\widehat{\sigma}(y) \times \mathrm{Softplus}(\mathrm{b_k})}{\eta(x_k) + \mathrm{Softplus}(\mathrm{c_k})}$$

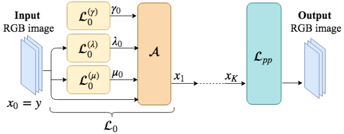
where $\eta(x_k)$ is the standard deviation of $[(D_\mathrm{h} x_k)^\top (D_\mathrm{v} x_k)^\top]^\top$ and $\widehat{\sigma}(y)$ is an estimation of noise level **[Ramadhan *et al.*,2017]**,

$$\widehat{\sigma}(y) = \mathrm{median}(|\mathrm{W_H y}|)/0.6745,$$

where $|W_\mathrm{H} y|$ is the vector gathering the absolute value of the diagonal coefficients of the first level Haar wavelet decomposition of the blurred image.

$\rightarrow$ iRestNet does not require knowledge of noise level

Network characteristics



- Number of layers : $K = 40$
- **Estimation of regularization parameter**

$$\lambda_k = \mathcal{L}_k^{(\lambda)}(x_k) = \frac{\widehat{\sigma}(y) \times \mathrm{Softplus}(\mathrm{b_k})}{\eta(x_k) + \mathrm{Softplus}(\mathrm{c_k})}$$

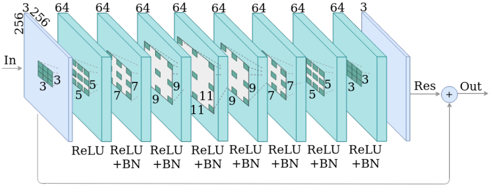
where $\eta(x_k)$ is the standard deviation of $[(D_{\mathrm{h}} x_k)^\top (D_{\mathrm{v}} x_k)^\top]^\top$ and $\widehat{\sigma}(y)$ is an estimation of noise level **[Ramadhan *et al.*,2017]**,

$$\widehat{\sigma}(y) = \mathrm{median}(|\mathrm{W_H y}|)/0.6745,$$

where $|W_{\mathrm{H}} y|$ is the vector gathering the absolute value of the diagonal coefficients of the first level Haar wavelet decomposition of the blurred image.

$\rightarrow$ iRestNet does not require knowledge of noise level

- Post-processing $\mathcal{L}_{\mathrm{pp}}$ **[Zhang *et al.*,2017]**



Numerical experiments

Dataset

- Training set : 200 RGB images from BSD500 + 1000 images from COCO
- Validation set : 100 validation images from BSD500
- Test set : 200 test images from BSD500

Numerical experiments

Dataset

- Training set : 200 RGB images from BSD500 + 1000 images from COCO
- Validation set : 100 validation images from BSD500
- Test set : 200 test images from BSD500

Test configurations

- GaussA : Gaussian kernel with std=1.6, $\sigma = 0.008$
- GaussB : Gaussian kernel with std=1.6, $\sigma \in [0.01, 0.05]$
- GaussC : Gaussian kernel with std=3, $\sigma = 0.04$
- Motion : motion kernel from **[Levin *et al.*,2009]** $\sigma = 0.01$
- Square : 7×7 square kernel, $\sigma = 0.01$

Numerical experiments

Dataset

- Training set : 200 RGB images from BSD500 + 1000 images from COCO
- Validation set : 100 validation images from BSD500
- Test set : 200 test images from BSD500

Test configurations

- GaussA : Gaussian kernel with std=1.6, $\sigma = 0.008$
- GaussB : Gaussian kernel with std=1.6, $\sigma \in [0.01, 0.05]$
- GaussC : Gaussian kernel with std=3, $\sigma = 0.04$
- Motion : motion kernel from **[Levin *et al.*,2009]** $\sigma = 0.01$
- Square : 7×7 square kernel, $\sigma = 0.01$

Training

- Loss : Structural SImilarity Measure (SSIM) **[Wang *et al.*, 2004]**, ADAM optimizer
- $\mathcal{L}_0, \ldots, \mathcal{L}_{29}$ trained individually, $\mathcal{L}_{\mathrm{pp}} \circ \mathcal{L}_{39} \circ \cdots \circ \mathcal{L}_{30}$ trained end-to-end $\rightarrow$ low memory
- Implemented with Pytorch using a GPU, $\sim$3-4 days per training (one iRestNet for each degradation model)

Numerical experiments

Dataset

- Training set : 200 RGB images from BSD500 + 1000 images from COCO
- Validation set : 100 validation images from BSD500
- Test set : 200 test images from BSD500

Test configurations

- GaussA : Gaussian kernel with std=1.6, $\sigma = 0.008$
- GaussB : Gaussian kernel with std=1.6, $\sigma \in [0.01, 0.05]$
- GaussC : Gaussian kernel with std=3, $\sigma = 0.04$
- Motion : motion kernel from **[Levin *et al.*,2009]** $\sigma = 0.01$
- Square : 7×7 square kernel, $\sigma = 0.01$

Training

- Loss : Structural SImilarity Measure (SSIM) **[Wang *et al.*, 2004]**, ADAM optimizer
- $\mathcal{L}_0, \ldots, \mathcal{L}_{29}$ trained individually, $\mathcal{L}_{\mathrm{pp}} \circ \mathcal{L}_{39} \circ \cdots \circ \mathcal{L}_{30}$ trained end-to-end $\rightarrow$ low memory
- Implemented with Pytorch using a GPU, ∼3-4 days per training (one iRestNet for each degradation model)

Competitors

- VAR : solution to $\mathcal{P}_0$ with projected gradient algorithm, (λ, δ) leading to best SSIM
- Deep learning methods : EPLL **[Zoran and Weiss, 2011]**, MLP **[Schuler *et al.*,2013]**, IRCNN **[Zhang *et al.*,2017]** (require noise level)

Results

- ✓ Higher average SSIM than competitors
- ✓ Higher SSIM on almost all images

	GaussA	GaussB	GaussC	Motion	Square
Blurred	0.675	0.522	0.326	0.548	0.543
VAR	0.804	0.724	0.585	0.829	0.756
EPLL	0.799	0.709	0.564	0.838	0.754
MLP	0.821	0.734	0.608	-	-
IRCNN	0.841	0.768	0.618	0.907	0.833
iRestNet	**0.850**	**0.786**	**0.638**	**0.911**	**0.839**

TABLE – SSIM results on the test set.

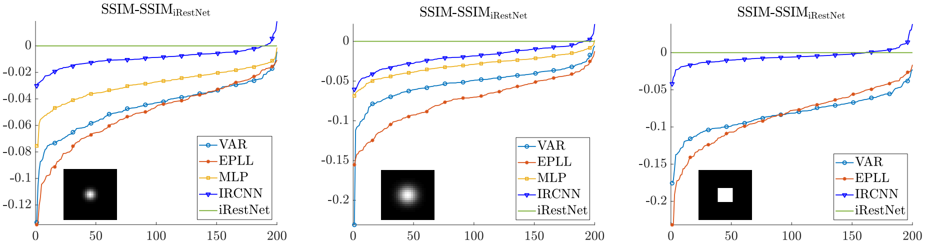


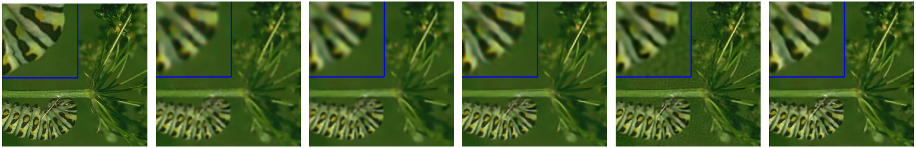
FIGURE – From left to right : GaussianA, GaussianC, Square.

Visual results

✓ Better contrast and more details

Ground-truth | VAR : 0.622 | EPLL : 0.552 | IRCNN : 0.685 | iRestNet : **0.708**

FIGURE – Visual results and SSIM obtained on one test image degraded with Square.



Ground-truth | VAR : 0.838 | EPLL : 0.842 | MLP : 0.862 | IRCNN : 0.842 | iRestNet : **0.887**

FIGURE – Visual results and SSIM obtained on one test image degraded with GaussB.

Conclusion

- Novel architecture based on an unfolded proximal interior point algorithm
- Allows to apply hard constraints on the image
- Expression and gradient of the proximity operator of the barrier

$\rightarrow$ Different application (classification, ...)

$\rightarrow$ When degradation is unkown : blind or semi-blind deconvolution

Related publications

iRestNet

C. Bertocchi, E. Chouzenoux, M.-C. Corbineau, M. Prato, J.-C. Pesquet

Deep unfolding of a proximal interior point method for image restoration

https://arxiv.org/abs/1812.04276

Network stability

P. L. Combettes and J.-C. Pesquet.

Deep neural network structures solving variational inequalities

https://arxiv.org/abs/1808.07526.

Proximal interior point methods

M.-C. Corbineau, E. Chouzenoux and J.-C. Pesquet.

PIPA : a new proximal interior point algorithm for large-scale convex optimization.

Proceedings of the 20th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2018.

M.-C. Corbineau, E. Chouzenoux and J.-C. Pesquet.

Geometry-texture decomposition/reconstruction using a proximal interior point algorithm

Proceedings of the 10th IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM), 2018.

Thank you !