Unfolding Proximal Algorithms

Emilie Chouzenoux[1]
in collaboration with
C. Bertocchi[2], M.-C. Corbineau[1], J.C. Pesquet[1], M. Prato[2]

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

4 February 2021
Statistics/Machine Learning at Paris Saclay

Applicative Motivation

Inverse problem in imaging

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

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

Applicative Motivation

Inverse problem in imaging

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

where $y \in \mathbb{R}^m$ observed data, $\mathcal{D}$ noise perturbation, $H \in \mathbb{R}^{m \times n}$ 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 factor, $\mathcal{C} \subset \mathbb{R}^n$

- ✓ Incorporate prior knowledge about solution and enforce desirable constraints
- ✓ Grounded on clear mathematical concepts
- ✗ No closed-form solution $\to$ iterative algorithms
- ✗ Objective function not always reflecting perceived quality
- ✗ Estimation of λ and tuning of algorithm parameters $\to$ time-consuming

Applicative Motivation

Inverse problem in imaging

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

where $y \in \mathbb{R}^m$ observed data, $\mathcal{D}$ noise perturbation, $H \in \mathbb{R}^{m \times n}$ 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 factor, $\mathcal{C} \subset \mathbb{R}^n$

- ✓ Incorporate prior knowledge about solution and enforce desirable constraints
- ✓ Grounded on clear mathematical concepts
- ✗ No closed-form solution $\rightarrow$ iterative algorithms
- ✗ Objective function not always reflecting perceived quality
- ✗ Estimation of λ and tuning of algorithm parameters $\rightarrow$ time-consuming

Deep-learning methods

- ✓ Generic methods for nonlinear approximation **[Cybenko, 1989]**
- ✓ Efficient for incorporating prior knowledge from big databases
- ✗ Make it difficult to account for physical models
- ✗ Black-box, empirical approaches

Applicative Motivation

Inverse problem in imaging

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

where $y \in \mathbb{R}^m$ observed data, $\mathcal{D}$ noise perturbation, $H \in \mathbb{R}^{m \times n}$ 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 factor, $\mathcal{C} \subset \mathbb{R}^n$

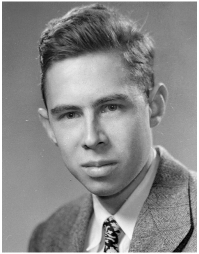
- ✓ Incorporate prior knowledge about solution and enforce desirable constraints
- ✓ Grounded on clear mathematical concepts
- ✗ No closed-form solution $\to$ iterative algorithms
- ✗ Objective function not always reflecting perceived quality
- ✗ Estimation of λ and tuning of algorithm parameters $\to$ time-consuming

Deep-learning methods

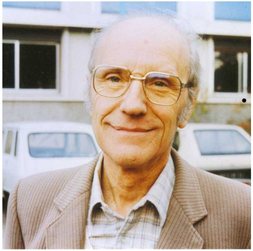
- ✓ Generic methods for nonlinear approximation **[Cybenko, 1989]**
- ✓ Efficient for incorporating prior knowledge from big databases
- ✗ Make it difficult to account for physical models
- ✗ Black-box, empirical approaches

$\to$ Combine benefits of both approaches : unfold optimization algorithms **[Gregor and LeCun, 2010]**

Theoretical Motivation



Frank Rosenblatt
(1928–1971)



Jean-Jacques Moreau
(1923–2014)

Projected Gradient Descent

Basic optimization problem

$$\underset{x \in \mathcal{C}}{\text{minimize}} \quad \frac{1}{2}\|Hx - y\|^2$$

where $\mathcal{C}$ nonempty closed convex subset of $\mathbb{R}^n$, $y \in \mathbb{R}^m$, and $H \in \mathbb{R}^{m \times n}$.

Projected gradient algorithm

$$(\forall k \in \mathbb{N}) \quad x_{k+1} = \mathrm{proj}_{\mathcal{C}}\left(x_k - \gamma_k H^\top (Hx_k - y)\right)$$

where $\gamma_k > 0$ is the step-size

Projected Gradient Descent

Basic optimization problem

$$\underset{x \in \mathcal{C}}{\text{minimize}} \quad \frac{1}{2}\|Hx - y\|^2$$

where $\mathcal{C}$ nonempty closed convex subset of $\mathbb{R}^n$, $y \in \mathbb{R}^m$, and $H \in \mathbb{R}^{m \times n}$.

Projected gradient algorithm

$$\begin{aligned}(\forall k \in \mathbb{N}) \quad x_{k+1} &= \mathrm{proj}_{\mathcal{C}}\Big(x_k - \gamma_k H^\top (Hx_k - y)\Big)\\ &= \mathrm{proj}_{\mathcal{C}}(W_k x_k + \gamma_k H^\top y)\end{aligned}$$

where $\gamma_k > 0$ is the step-size and $W_k = I_n - \gamma_k H^\top H$.

Projected Gradient Descent

Basic optimization problem

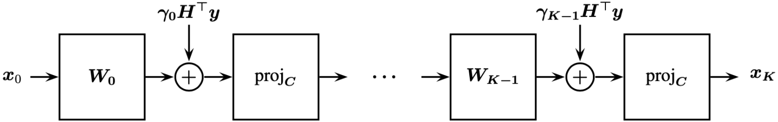
$$\underset{x \in \mathcal{C}}{\text{minimize}} \quad \frac{1}{2}\|Hx - y\|^2$$

where $\mathcal{C}$ nonempty closed convex subset of $\mathbb{R}^n$, $y \in \mathbb{R}^m$, and $H \in \mathbb{R}^{m \times n}$.

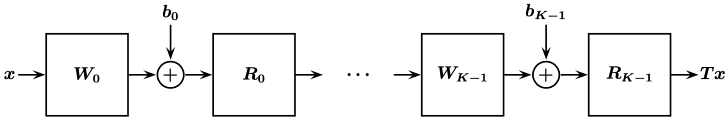
Projected gradient algorithm

$$\begin{aligned}(\forall k \in \mathbb{N}) \quad x_{k+1} &= \mathrm{proj}_{\mathcal{C}}\Big(x_k - \gamma_k H^\top (Hx_k - y)\Big) \\ &= \mathrm{proj}_{\mathcal{C}}(W_k x_k + \gamma_k H^\top y)\end{aligned}$$

where $\gamma_k > 0$ is the step-size and $W_k = I_n - \gamma_k H^\top H$.



Feedforward NNs



Neural network model

$$T = T_{K-1} \circ \cdots \circ T_0$$

where

$$(\forall k \in \{0, \dots, K-1\}) \quad T_k \colon \mathbb{R}^{n_k} \to \mathbb{R}^{n_{k+1}} \colon x \mapsto R_k(W_k x + b_k)$$

- $W_k \in \mathbb{R}^{n_{k+1} \times n_k}$ is a weight matrix
- b_k is a bias vector in $\mathbb{R}^{n_{k+1}}$
- $R_k \colon \mathbb{R}^{n_{k+1}} \to \mathbb{R}^{n_{k+1}}$ is an activation operator.

Remark $(W_k)_{0 \le k \le K-1}$ can be convolutive operators

Link

Proximity operator [Moreau, 1962]

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

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

Special case

If f is the indicator function of $\mathcal{C}$, then $\mathrm{prox}_f = \mathrm{proj}_{\mathcal{C}}$.
projected gradient algorithm $\rightsquigarrow$ proximal-gradient algorithm $\rightsquigarrow$ forward-backward algorithm

Link

Proximity operator [Moreau, 1962]

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

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

Special case

If f is the indicator function of $\mathcal{C}$, then $\operatorname{prox}_f = \operatorname{proj}_{\mathcal{C}}$.
projected gradient algorithm $\rightsquigarrow$ proximal-gradient algorithm $\rightsquigarrow$ forward-backward algorithm

Most of the activation operators are proximity operators

Example of proximal activation operators

ReLU

$$\varrho\colon \mathbb{R} \to \mathbb{R}\colon \xi \mapsto \begin{cases} \xi, & \text{if } \xi > 0; \\ 0, & \text{if } \xi \leq 0. \end{cases}$$

Then, $\varrho = \mathrm{proj}_{[0,+\infty[}$.

Parametric rectified linear unit activation function

$$\varrho\colon \mathbb{R} \to \mathbb{R}\colon \xi \mapsto \begin{cases} \xi, & \text{if } \xi > 0; \\ \alpha\xi, & \text{if } \xi \leq 0 \end{cases}, \qquad \alpha \in]0,1].$$

Then $\varrho = \mathrm{prox}_\phi$ where

$$\phi\colon \mathbb{R} \to \mathbb{R}\colon \xi \mapsto \begin{cases} 0, & \text{if } \xi > 0; \\ (1/\alpha - 1)\xi^2/2, & \text{if } \xi \leq 0. \end{cases}$$

Example of proximal activation operators

Unimodal sigmoid activation function

$$\varrho\colon \mathbb{R} \to \mathbb{R}\colon \xi \mapsto \frac{1}{1+e^{-\xi}} - \frac{1}{2}$$

Then $\varrho = \operatorname{prox}_\phi$ where

$$\phi\colon \xi \mapsto \begin{cases} (\xi+1/2)\ln(\xi+1/2) + (1/2-\xi)\ln(1/2-\xi) - \dfrac{1}{2}(\xi^2+1/4) & \text{if } |\xi| < 1/2; \\ -1/4, & \text{if } |\xi| = 1/2; \\ +\infty, & \text{if } |\xi| > 1/2. \end{cases}$$

Elliot activation function

$$\varrho\colon \mathbb{R} \to \mathbb{R}\colon \xi \mapsto \frac{\xi}{1+|\xi|}.$$

We have $\varrho = \operatorname{prox}_\phi$, where

$$\phi\colon \mathbb{R} \to]-\infty,+\infty]\colon \xi \mapsto \begin{cases} -|\xi| - \ln(1-|\xi|) - \frac{\xi^2}{2}, & \text{if } |\xi| < 1; \\ +\infty, & \text{if } |\xi| \geq 1. \end{cases}$$

Example of proximal activation operators

Softmax

$$R\colon \mathbb{R}^n \to \mathbb{R}^n\colon (\xi_i)_{1\le i\le n} \mapsto \left(\exp(\xi_i) \Big/ \sum_{j=1}^{N} \exp(\xi_j)\right)_{1\le i\le n} - u,$$

where $u = (1,\ldots,1)/n \in \mathbb{R}^n$.
Then $R = \operatorname{prox}_\varphi$ where $\varphi = \psi(\cdot + u) + \langle \cdot \mid u\rangle$ and

$$\psi\colon \mathbb{R}^n \to]-\infty,+\infty]$$

$$(\xi_i)_{1\le i\le n} \mapsto \begin{cases} \displaystyle\sum_{i=1}^{n} \left(\xi_i \ln \xi_i - \frac{\xi_i^2}{2}\right), & \text{if } (\xi_i)_{1\le i\le n} \in [0,1]^n \text{ and } \displaystyle\sum_{i=1}^{n} \xi_i = 1; \\ +\infty, & \text{otherwise.} \end{cases}$$

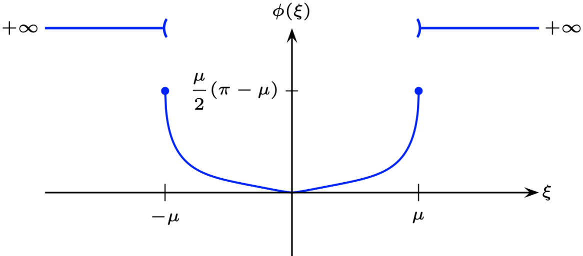
Example of proximal activation operators

Squashing function used in capsnets

$$(\forall x \in \mathbb{R}^n) \quad Rx = \frac{\mu\|x\|}{1+\|x\|^2}x = \mathrm{prox}_{\phi\circ\|\cdot\|}x, \quad \mu = \frac{8}{3\sqrt{3}},$$

where

$$\phi\colon \xi \mapsto \begin{cases} \mu \arctan\sqrt{\dfrac{|\xi|}{\mu-|\xi|}} - \sqrt{|\xi|(\mu-|\xi|)} - \dfrac{\xi^2}{2}, & \text{if } |\xi| < \mu; \\ \dfrac{\mu(\pi-\mu)}{2}, & \text{if } |\xi| = \mu; \\ +\infty, & \text{otherwise.} \end{cases}$$



Problem

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

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

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

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

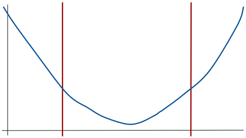
$\mathcal{B}$: logarithmic barrier

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

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

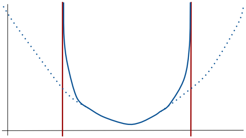
$$\mathcal{P}_0 : \quad \underset{x \in \mathcal{C}}{\text{minimize}} \quad f(Hx, y) + \lambda \mathcal{R}(x)$$

$$\Downarrow$$

Unconstrained Subproblem

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

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



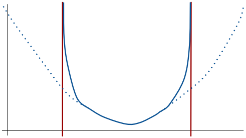
Logarithmic barrier method

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

$$\Downarrow$$

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

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

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

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 expression 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 $\varphi : (x, \alpha) \mapsto \operatorname{prox}_{\alpha\mathcal{B}}(x)$. 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

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

and

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

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 $\varphi : (x, \alpha) \mapsto \operatorname{prox}_{\alpha\mathcal{B}}(x)$. 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

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

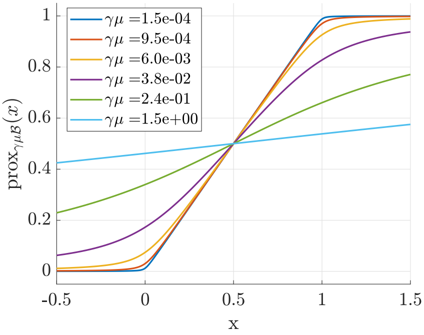
and

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

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

Proximity operator of the barrier

Bound constraints $\mathcal{C} = [0, 1]$



Proximity operator of the barrier

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

Proposition 3

Let $\varphi : (x, \alpha) \mapsto \mathrm{prox}_{\alpha\mathcal{B}}(x)$. 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

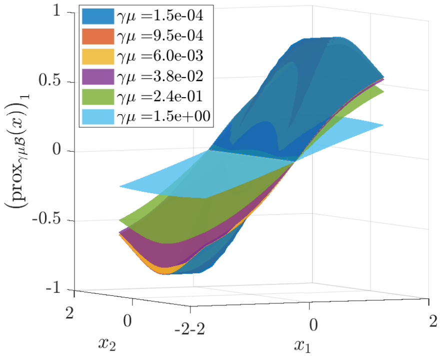
$$\nabla_\varphi^{(\alpha)}(x, \alpha) = \frac{-2}{\rho - \|\varphi(x, \alpha) - c\|^2 + 2\alpha} M(x, \alpha)(\varphi(x, \alpha) - c),$$

where

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

Proximity operator of the barrier

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



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 $k = 0, 1, \ldots$ **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

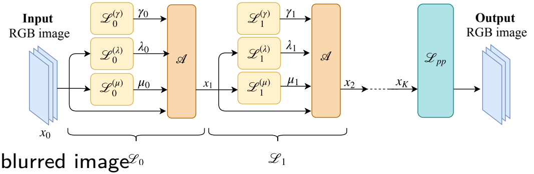
- ✓ Efficient algorithm for constrained optimization
- ✗ Setting of the parameters $(\mu_k, \gamma_k)_{k\in\mathbb{N}}$?
- ✗ How to finding the regularization parameter λ leading to the best visual quality of the solution ?

$\rightarrow$ 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**

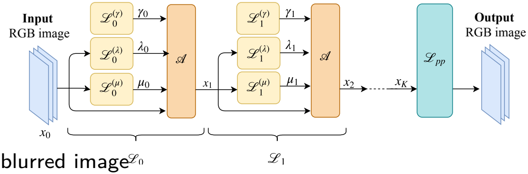


Input : $x_0 = y$ blurred image

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

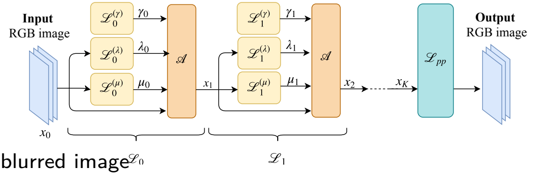
Hidden structures

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

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

iRestNet architecture

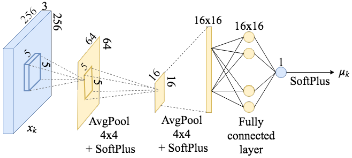
$\longrightarrow$ **Unfold proximal IP algorithm over K iterations, untie γ, μ and λ across network**



Input : $x_0 = y$ blurred image

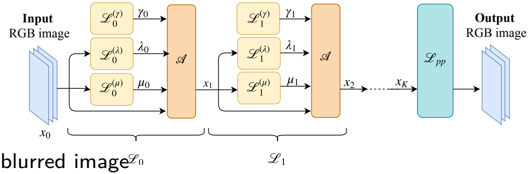
Hidden structures

- $(\mathcal{L}_k^{(\gamma)})_{0 \leq k \leq K-1}$: 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**



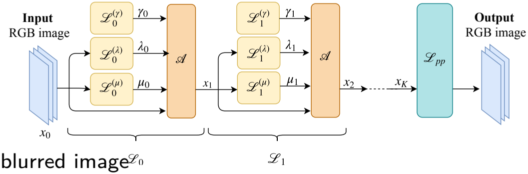
Input : $x_0 = y$ blurred image

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**



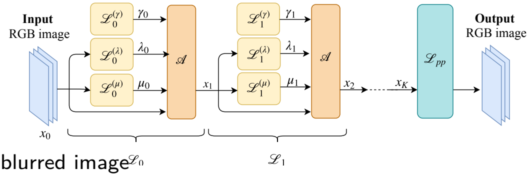
Input : $x_0 = y$ blurred image

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**



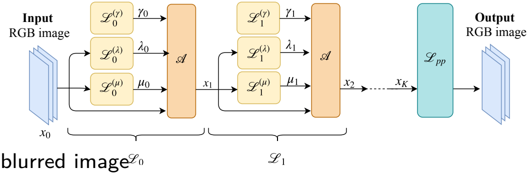
Input : $x_0 = y$ blurred image

Hidden structures

- $(\mathcal{L}_k^{(\gamma)})_{0\leq k\leq K-1}$: 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

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
- $\mathcal{A}(x_k, \mu_k, \gamma_k, \lambda_k) = \operatorname{prox}_{\gamma_k \mu_k \mathcal{B}} \left(x_k - \gamma_k \left(H^\top \nabla_1 f(Hx_k, y) + \lambda_k \nabla \mathcal{R}(x_k) \right) \right)$
- $\mathcal{L}_{\mathrm{pp}}$: post-processing layer $\rightarrow$ removes remaining artifacts

Training Stochastic gradient descent and backpropagation ($\nabla \mathcal{A}$ thanks to Propositions 1-3)

Network stability

What about the network stability ?

Network stability

What about the network stability ?

- Deep learning : lack of robustness, e.g. AlexNet **[Szegedy *et al.*, 2013]**
- Applications with high risk and legal responsibility (medical image processing, driving, security, etc...) $\rightarrow$ need for theoretical guarantees
- Asymptotic and robustness analyses addressed within the framework of **averaged operators [Combettes and Pesquet, 2020]**

Averaged operators

Definition – α-averaged operator

Let $T : \mathbb{R}^n \to \mathbb{R}^n$ 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 – α-averaged operator

Let $T : \mathbb{R}^n \to \mathbb{R}^n$ 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 $R_{K-1} \circ (W_{K-1} \cdot + b_{K-1}) \circ \cdots \circ R_0 \circ (W_0 \cdot + b_0)$

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

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

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

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

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

Averageness result

Theorem 1 [Combettes and Pesquet, 2020]

Let $\alpha \in [1/2, 1]$. Let $K = 2$. Let $\rho = \inf_{x \in \mathbb{R}^n,\, \|x\|=1} \langle W_1 W_0 x \mid x \rangle$, and let

$$\theta_1 = \|W_1 W_0\| + \|W_1\| \|W_0\|$$

If one of the following conditions is satisfied :

(i) $W_0 = 0$ or $W_1 = 0$;

(ii) $\|W_1 W_0 - 4(1-\alpha) I_n\| - \|W_1 W_0\| + 2\theta_1 \leq 4\alpha$;

(iii) $\alpha \neq 1$, $W_0 \neq 0$, $W_1 \neq 0$, and there exists $\eta \in [0, \alpha/((1-\alpha)\theta_1)]$ such that

$$\begin{cases} \theta_1 \leq 2\alpha \\ \alpha\theta_1 + (1-\alpha)(\|\mathbb{I}_n - \eta W_1 W_0\| - \eta\|W_1 W_0\|)(\theta_1 - \|W_1 W_0\|) \leq 2\alpha - 1 + (1-\alpha)\rho, \end{cases}$$

then $T = R_1 \circ (W_1 \cdot + b_1) \circ R_0 \circ (W_0 \cdot + b_0)$ is α-**averaged**.

Avarageness result

Theorem 1 [Combettes and Pesquet, 2020]

Let $\alpha \in [1/2, 1]$. Let $K = 3$. Let $W = W_2 \circ W_1 \circ W_0$ Let $\rho = \inf_{x \in \mathbb{R}^n, \|x\|=1} \langle Wx \mid x \rangle$, and let

$$\theta_2 = \|W\| + \|W_2\|\|W_1 W_0\| + \|W_2 W_1\|\|W_0\| + \|W_2\|\|W_1\|\|W_0\|$$

If one of the following conditions is satisfied :

(i) $W_0 = 0$ or $W_1 = 0$ or $W_2 = 0$;

(ii) $\|W - 8(1-\alpha)I_n\| - \|W\| + 2\theta_2 \leq 8\alpha$;

(iii) $\alpha \neq 1$, $W_0 \neq 0$, $W_1 \neq 0$, $W_2 \neq 0$, and there exists $\eta \in [0, \alpha/((1-\alpha)\theta_2)]$ such that

$$\begin{cases} \theta_2 \leq 4\alpha \\ \alpha\theta_2 + (1-\alpha)(\|\mathbb{I}_n - \eta W\| - \eta\|W\|)(\theta_2 - \|W\|) \leq 2(2\alpha - 1) + (1-\alpha)\rho, \end{cases}$$

then $T = R_2 \circ (W_2 \cdot + b_2) \circ R_2 \circ (W_3 \cdot + b_3) \circ R_0 \circ (W_0 \cdot + b_0)$ is α-**averaged**.

Averageness result

Theorem 1 [Combettes and Pesquet, 2020]

Let $\alpha \in [1/2, 1]$. Let $K \geq 1$ be an integer. Let $W = W_{K-1} \circ \cdots \circ W_0$, let $\rho = \inf_{x \in \mathbb{R}^n,\, \|x\|=1} \langle Wx \mid x \rangle$, and let

$$\theta_{K-1} = \|W\| + \sum_{\ell=0}^{K-2} \sum_{0 \leq j_0 < \cdots < j_\ell \leq K-2} \|W_{K-1} \circ \cdots \circ W_{j_\ell+1}\| \|W_{j_\ell} \circ \cdots \circ W_{j_{\ell-1}+1}\| \cdots \|W_{j_0} \circ \cdots \circ W_0\|.$$

If one of the following conditions is satisfied :

(i) There exists $k \in \{0, \ldots, K-1\}$ such that $W_k = 0$;

(ii) $\|W - 2^K(1-\alpha)I_n\| - \|W\| + 2\theta_{K-1} \leq 2^K \alpha$;

(iii) $\alpha \neq 1$, for every $k \in \{0, \ldots, K-1\}$ $W_k \neq 0$, and there exists $\eta \in [0, \alpha/((1-\alpha)\theta_{K-1})]$ such that

$$\begin{cases} \theta_{K-1} \leq 2^{K-1}\alpha \\ \alpha\theta_{K-1} + (1-\alpha)(\|\mathbb{I}_n - \eta W\| - \eta\|W\|)(\theta_{K-1} - \|W\|) \leq 2^{K-2}(2\alpha - 1) + (1-\alpha)\rho, \end{cases}$$

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

Take-home message : the stability a neural network depends on its weight operators

Network stability result

Assumption

Consider the quadratic problem, assume that $H^\top H$ and $D^\top D$ are **diagonalizable in the same basis** $\mathcal{P}$.

Network stability result

Assumption

Consider the quadratic problem, assume that $H^\top H$ and $D^\top D$ are **diagonalizable in the same basis** $\mathcal{P}$.

Notation

For every $p \in \{1, \ldots, n\}$ 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

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

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

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

Network stability result

Assumption

Consider the quadratic problem, assume that $H^\top H$ and $D^\top D$ are **diagonalizable in the same basis** $\mathcal{P}$.

Notation

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

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

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

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

Theorem 2

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**.

Numerical experiments

Image deblurring

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

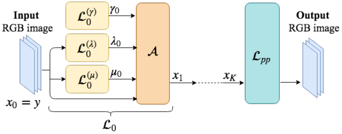
- $H \in \mathbb{R}^{n\times n}$: circular convolution with known blur
- $\omega \in \mathbb{R}^{n}$: additive white Gaussian noise with standard deviation σ
- $y \in \mathbb{R}^{n}$, $\overline{x} \in \mathbb{R}^{n}$: RGB images

Variational formulation

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

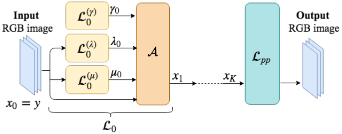
- δ : smoothing parameter, $\delta = 0.01$ for iRestNet
- $D_{\mathrm{h}} \in \mathbb{R}^{n\times n}$, $D_{\mathrm{v}} \in \mathbb{R}^{n\times n}$: horizontal and vertical spatial gradient operators

Network characteristics



- Number of layers : $K = 40$

Network characteristics



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

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

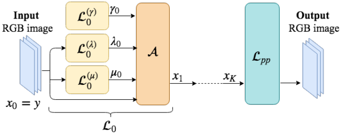
where $\eta(x_k)$ is the standard deviation of $[(D_{\mathrm{h}}x_k)^\top (D_{\mathrm{v}}x_k)^\top]^\top$ and $\widehat{\sigma}(y)$ is an estimation of noise level **[Ramadhan *et al.*,2017]**,

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

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**

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

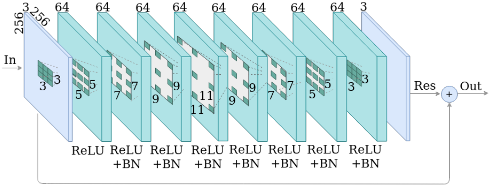
where $\eta(x_k)$ is the standard deviation of $[(D_\mathrm{h} x_k)^\top (D_\mathrm{v} x_k)^\top]^\top$ and $\widehat{\sigma}(y)$ is an estimation of noise level **[Ramadhan *et al.*,2017]**,

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

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 uniform 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 uniform 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, $\mathcal{L}_{\rm pp} \circ \mathcal{L}_{39} \circ \cdots \circ \mathcal{L}_{30}$ trained end-to-end $\rightarrow$ low memory
- Implemented with Pytorch using a GPU, ∼3-4 days per training (one iRestNet for each degradation model)

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 uniform 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**

FIGURE – SSIM results on the test set.

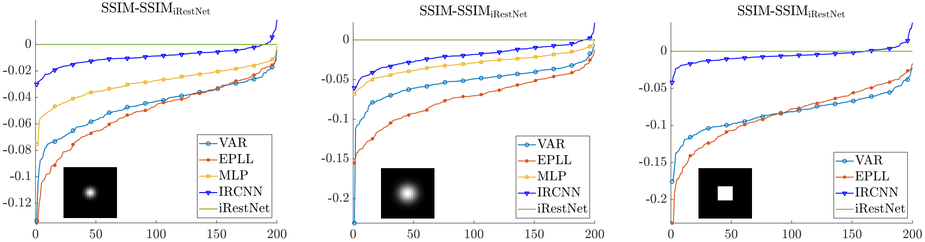


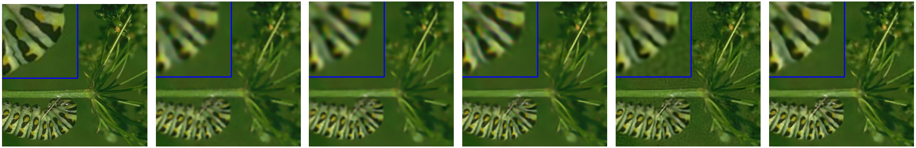
FIGURE – From left to right : GaussianA, GaussianC, Square.

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

- Neural network architecture built in an explainable manner
- Practically efficient methods developed by mixing ideas from iterative optimization algorithms and NN techniques
- Expressions of the proximity operator of some barrier functions and their gradients
- Requirement of better nonconvex optimization methods
- Optimization concepts are not only useful to train NNs, but also to analyze them

Related publications

iRestNet

C. Bertocchi, E. Chouzenoux, M.-C. Corbineau,J.-C. Pesquet, and M. Prato
Deep unfolding of a proximal interior point method for image restoration
Inverse Problems, vol. 36, no 3, pp. 034005, Feb. 2020.

M. Galinier, M. Prato, C. Bertocchi, E. Chouzenoux, and J.-C. Pesquet
A hybrid interior point - deep learning appproach for Poisson image deblurring
IEEE International Workshop on Machine Learning for Signal Processing, 2020.

Variational analysis of neural networks

P. L. Combettes and J.-C. Pesquet
Deep neural network structures solving variational inequalities
Set-Valued and Variational Analysis, vol. 28, pp. 491–518, Sept. 2020.

P. L. Combettes and J.-C. Pesquet
Lipschitz certificates for layered network structures driven by averaged activation operators
SIAM Journal on Mathematics of Data Science, vol. 2, no. 2, pp. 529–557, June 2020.

Proximal interior point methods

M.-C. Corbineau, E. Chouzenoux, and J.-C. Pesquet
A Proximal Interior Point Algorithm with applications to image processing
Journal of Mathematical Imaging and Vision, vol. 62, no. 6, pp. 919-940, 2020

Thank you !