Parameters estimation via unrolling for the PNPD algorithm

Giuseppe Scarlato[1]

Joint work with
Marco Donatelli[1]

July 2, 2026

[1]Università degli Studi dell'Insubria

MMIPU 2026 workshop
ICCSA 2026, University of Minho, Braga, Portugal

Outline

PNPD algorithm

Linear inverse problem

Consider a linear inverse problem of the form

$$b^\delta = Au + \eta_\delta = b + \eta_\delta, \tag{1}$$

where

- $b^\delta \in \mathbb{R}^s$ is the observed data corrupted by noise $\eta_\delta \in \mathbb{R}^s$
- $A \in \mathbb{R}^{s \times d}$ is a linear operator
- $u \in \mathbb{R}^d$ is the unknown signal we want to recover

Optimization problem

A common variational approach to solve eq. (1) is to consider the following optimization problem

$$\begin{aligned} &\underset{u\in\mathbb{R}^d}{\operatorname{argmin}} \ \frac{1}{2}\|Au - b^\delta\|^2 + h(Wu) \\ =\ &\underset{u\in\mathbb{R}^d}{\operatorname{argmin}} \ f(u) + h(Wu), \end{aligned} \tag{2}$$

where

- $\|.\|$ denotes the Euclidean norm
- $f(u)$ is the data fidelity term
- $h(Wu)$ is the regularization term
- $W \in \mathbb{R}^{d' \times d}$ is a linear operator
- $h : \mathbb{R}^{d'} \to \mathbb{R} \cup \{\infty\}$ is a convex and possibly non-smooth function.

Proximal gradient

Proximal gradient methods are first-order methods that alternate a gradient step on f and a proximal evaluation on $h \circ W$:

Proximal gradient

Proximal gradient methods are first-order methods that alternate a gradient step on f and a proximal evaluation on $h \circ W$:

$$\begin{cases} u_{n+\frac{1}{2}} = u - \alpha_n \nabla f(u) & \leftarrow \text{gradient descent} \\ \end{cases}$$

Proximal gradient

Proximal gradient methods are first-order methods that alternate a gradient step on f and a proximal evaluation on $h \circ W$:

$$\begin{cases} u_{n+\frac{1}{2}} = u - \alpha_n \nabla f(u) & \leftarrow \text{gradient descent} \\ u_{n+1} = \operatorname{prox}_{\alpha_n h \circ W}(u_{n+\frac{1}{2}}) & \leftarrow \text{proximal evaluation} \end{cases} \tag{3}$$

Proximal gradient

Proximal gradient methods are first-order methods that alternate a gradient step on f and a proximal evaluation on $h \circ W$:

$$\begin{cases} u_{n+\frac{1}{2}} = u - \alpha_n \nabla f(u) & \leftarrow \text{gradient descent} \\ u_{n+1} = \operatorname{prox}_{\alpha_n h \circ W}(u_{n+\frac{1}{2}}) & \leftarrow \text{proximal evaluation} \end{cases} \tag{3}$$

where

- $\alpha_n > 0$ is the step size,
- $\operatorname{prox}_g(v) = \operatorname{argmin}_u \ \frac{1}{2}\|u - v\|^2 + g(u)$ is the proximal operator of g.

Proximal gradient

Proximal gradient methods are first-order methods that alternate a gradient step on f and a proximal evaluation on $h \circ W$:

$$\begin{cases} u_{n+\frac{1}{2}} = u - \alpha_n \nabla f(u) & \leftarrow \text{gradient descent} \\ u_{n+1} = \operatorname{prox}_{\alpha_n h \circ W}(u_{n+\frac{1}{2}}) & \leftarrow \text{proximal evaluation} \end{cases} \tag{3}$$

where

- $\alpha_n > 0$ is the step size,
- $\operatorname{prox}_g(v) = \operatorname{argmin}_u \ \frac{1}{2}\|u - v\|^2 + g(u)$ is the proximal operator of g.

A Nesterov-like extrapolation, like in FISTA, can be added to eq. (3) to accelerate convergence.

Proximal gradient

Proximal gradient methods are first-order methods that alternate a gradient step on f and a proximal evaluation on $h \circ W$:

$$\begin{cases} u_{n+\frac{1}{2}} = u - \alpha_n \nabla f(u) & \leftarrow \text{gradient descent} \\ u_{n+1} = \operatorname{prox}_{\alpha_n h \circ W}(u_{n+\frac{1}{2}}) & \leftarrow \text{proximal evaluation} \end{cases} \tag{3}$$

where

- $\alpha_n > 0$ is the step size,
- $\operatorname{prox}_g(v) = \operatorname{argmin}_u \ \frac{1}{2}\|u - v\|^2 + g(u)$ is the proximal operator of g.

A Nesterov-like extrapolation, like in FISTA, can be added to eq. (3) to accelerate convergence.

Equation (3) assumes that the proximal operator of $h \circ W$ can be computed in closed form. This is not the case for some common regularization terms, such as the Total Variation (TV).

Nested Primal-Dual (NPD)

The Nested Primal-Dual (NPD)[1][2] algorithm is an inexact proximal gradient method summarized as

$$\begin{cases} \bar{u}_n = u_n + \gamma_n(u_n - u_{n-1}) & \leftarrow \text{Nesterov extrap.} \\ u_{n+\frac{1}{2}} = \bar{u}_n - \alpha_n \nabla f(\bar{u}_n) & \leftarrow \text{gradient descent} \\ u_{n+1} \approx \operatorname{prox}_{\alpha_n h \circ W}(u_{n+\frac{1}{2}}) & \leftarrow \text{inexact proximal step} \end{cases}$$

where $\gamma_n \geq 0$ is the extrapolation parameter like in FISTA.

[1] Chen and Loris 2019, "On starting and stopping criteria for nested primal-dual iterations".

[2] Bonettini et al. 2023, "A nested primal–dual FISTA-like scheme for composite convex optimization problems".

Nested Primal-Dual (NPD)

The Nested Primal-Dual (NPD)[1][2] algorithm is an inexact proximal gradient method summarized as

$$\begin{cases} \bar{u}_n = u_n + \gamma_n(u_n - u_{n-1}) & \leftarrow \text{Nesterov extrap.} \\ u_{n+\frac{1}{2}} = \bar{u}_n - \alpha_n \nabla f(\bar{u}_n) & \leftarrow \text{gradient descent} \\ u_{n+1} \approx \operatorname{prox}_{\alpha_n h \circ W}(u_{n+\frac{1}{2}}) & \leftarrow \text{inexact proximal step} \end{cases}$$

where $\gamma_n \geq 0$ is the extrapolation parameter like in FISTA.

The inexact proximal step is computed by $k_{max} \in \mathbb{N}$ steps of a dual sequence v^k:
Let $u \in \mathbb{R}^d$, $v^0 \in \mathbb{R}^{d'}$, $\alpha > 0$, $0 < \beta < 2/\|W\|^2$, and define the sequence

$$v^{k+1} = \operatorname{prox}_{\beta\alpha^{-1} h^*}(v^k + \beta\alpha^{-1} W(u - \alpha W^T v^k)),$$

having limit $\lim_{k\to\infty} v^k = \hat{v}$. Then we have

$$\operatorname{prox}_{\alpha h \circ W}(u) = u - \alpha W^T \hat{v},$$

[1] Chen and Loris 2019, "On starting and stopping criteria for nested primal-dual iterations".
[2] Bonettini et al. 2023, "A nested primal–dual FISTA-like scheme for composite convex optimization problems".

Preconditioned NPD

Applying left preconditioning to the least squares regularized problem, the norm of the data fidelity term changes. Given S positive definite, we have

$$\operatorname*{argmin}_{u \in \mathbb{R}^d} \frac{1}{2} \|Au - b^\delta\|^2_{S^{-1}} + h(Wu), \tag{4}$$

where

$$f_S(u) = \|Au - b^\delta\|^2_{S^{-1}} = \|S^{-\frac{1}{2}}(Au - b^\delta)\|^2.$$

Preconditioned NPD

Applying left preconditioning to the least squares regularized problem, the norm of the data fidelity term changes. Given S positive definite, we have

$$\underset{u\in\mathbb{R}^d}{\operatorname{argmin}} \frac{1}{2}\|Au - b^\delta\|^2_{S^{-1}} + h(Wu), \tag{4}$$

where

$$f_S(u) = \|Au - b^\delta\|^2_{S^{-1}} = \|S^{-\frac{1}{2}}(Au - b^\delta)\|^2.$$

Assuming that there exist a positive definite matrix P such that

$$P^{-1}A^T = A^T S^{-1}, \tag{5}$$

Preconditioned NPD

Applying left preconditioning to the least squares regularized problem, the norm of the data fidelity term changes. Given S positive definite, we have

$$\operatorname*{argmin}_{u\in\mathbb{R}^d} \frac{1}{2}\|Au - b^\delta\|^2_{S^{-1}} + h(Wu), \tag{4}$$

where

$$f_S(u) = \|Au - b^\delta\|^2_{S^{-1}} = \|S^{-\frac{1}{2}}(Au - b^\delta)\|^2.$$

Assuming that there exist a positive definite matrix P such that

$$P^{-1}A^T = A^T S^{-1}, \tag{5}$$

then NPD applied to eq. (4) gives the Preconditioned NPD (PNPD) algorithm:

$$\begin{cases} \bar{u}_n = u_n + \gamma_n(u_n - u_{n-1}), \\ u_{n+\frac{1}{2}} = \bar{u}_n - \alpha_n P^{-1}\nabla f(\bar{u}_n), \\ u_{n+1} \approx \operatorname{prox}_{\alpha_n h\circ W}(u_{n+\frac{1}{2}}), \end{cases}$$

Preconditioner choice

To satisfy the condition in eq. (5), we can choose S and P as follows:

$$P = A^T A + \nu I, \qquad S = AA^T + \nu I,$$

with $\nu > 0$. With this choice, the gradient step is equivalent to the Levenberg-Marquardt method applied to linear inverse problems, or equivalently, to the Iterated Tikhonov method[3].

[3] Hanke and Groetsch 1998, “Nonstationary Iterated Tikhonov Regularization”.

Preconditioner choice

To satisfy the condition in eq. (5), we can choose S and P as follows:

$$P = A^T A + \nu I, \qquad S = AA^T + \nu I,$$

with $\nu > 0$. With this choice, the gradient step is equivalent to the Levenberg-Marquardt method applied to linear inverse problems, or equivalently, to the Iterated Tikhonov method[3].

In general, the identity in eq. (5) is satisfied whenever P is a polynomial of $A^T A$ and S is a corresponding polynomial of AA^T.

[3] Hanke and Groetsch 1998, "Nonstationary Iterated Tikhonov Regularization".

Unrolling architecture

Unrolling (single step)

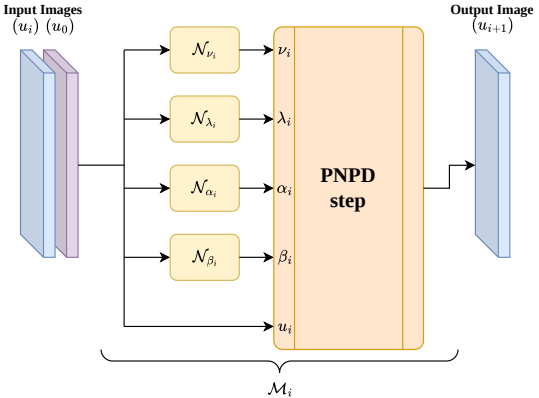


Figure 1: Single iteration of the unrolling architecture $\mathcal{M}_i$ composed by the predictors for each parameter $\mathcal{N}_{\nu_i}, \mathcal{N}_{\lambda_i}, \mathcal{N}_{\alpha_i}, \mathcal{N}_{\beta_i}$ and by one step of the PNPD algorithm.

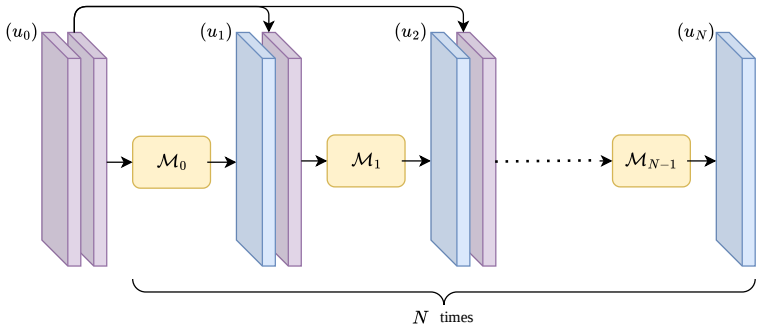


Figure 2: Global unrolling architecture obtained by composing the base network $\mathcal{M}_i$ N times.

Training

Training loss

The training loss is defined as a weighted average of a base loss function ℓ computed on each intermediate iterate

$$\mathcal{L}(u_0, \dots, u_{N-1}, u) = \frac{\sum_{i=0}^{N-1} w_i \, \ell(u_i, u)}{\sum_{i=0}^{N-1} w_i}.$$

Training loss

The training loss is defined as a weighted average of a base loss function ℓ computed on each intermediate iterate

$$\mathcal{L}(u_0, \dots, u_{N-1}, u) = \frac{\sum_{i=0}^{N-1} w_i \, \ell(u_i, u)}{\sum_{i=0}^{N-1} w_i}.$$

For the numerical experiments, we use the negative Structural Similarity Index (SSIM) as the base loss function

$$\ell(u_i, u) = -\operatorname{SSIM}(u_i, u),$$

Training loss

The training loss is defined as a weighted average of a base loss function ℓ computed on each intermediate iterate

$$\mathcal{L}(u_0, \dots, u_{N-1}, u) = \frac{\sum_{i=0}^{N-1} w_i \, \ell(u_i, u)}{\sum_{i=0}^{N-1} w_i}.$$

For the numerical experiments, we use the negative Structural Similarity Index (SSIM) as the base loss function

$$\ell(u_i, u) = -\operatorname{SSIM}(u_i, u),$$

and we choose the weights w_i as

$$w_i = \frac{1}{N - i}.$$

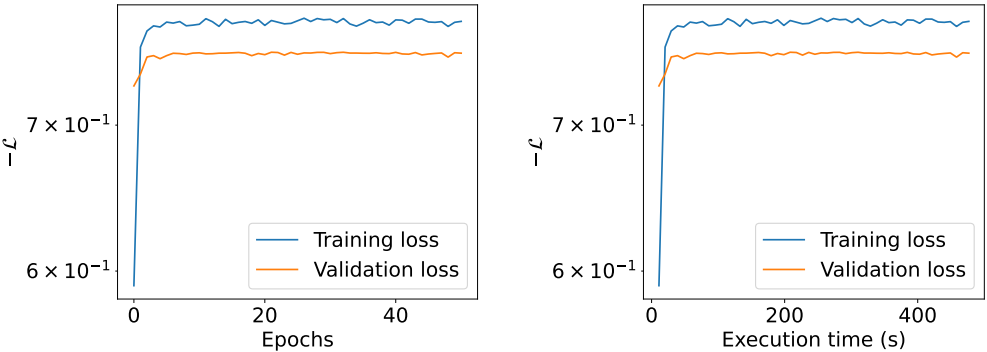


Figure 3: Evolution of the loss function as a function of epochs (left) and of time (right).

Numerical results

Image deblurring

For the numerical experiments, we consider the image deblurring problem with Total Variation regularization, which can be formulated as

$$\underset{u\in\mathbb{R}^d}{\operatorname{argmin}} \underbrace{\frac{1}{2}\|Au-b^\delta\|^2}_{\text{data fidelity}} + \underbrace{\lambda\,\mathrm{TV}(u)}_{\text{regularization}},$$

which is a specific instance of the model in eq. (2) where

- $f(u)=\frac{1}{2}\|Au-b^\delta\|^2$,
- $h(Wu)=\lambda\,\mathrm{TV}(u)$,
- $A\in\mathbb{R}^{d\times d}$ is the blurring operator,
- $\lambda>0$ is a regularization parameter
- $b^\delta\in\mathbb{R}^d$ is the observed image (blurred image with noise).

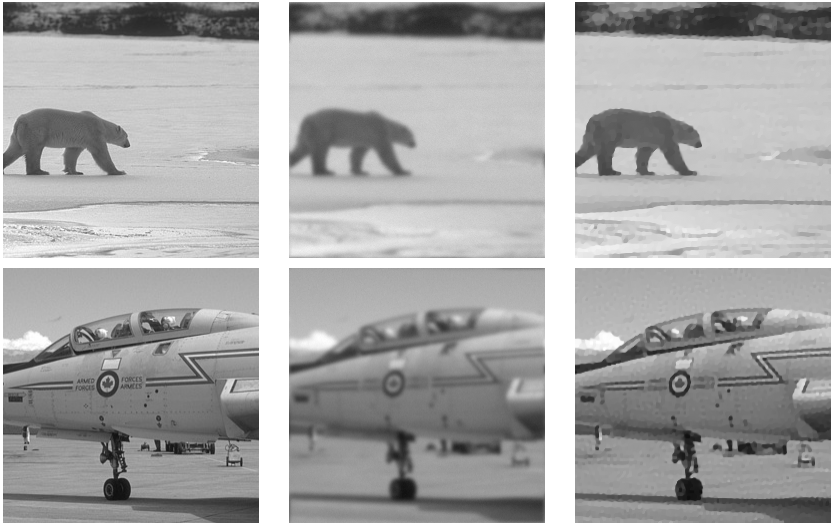
Total Variation

The TV is the Total Variation operator defined as

$$\mathrm{TV}(x) = \sum_{i=1}^{d} \|\nabla_i u\|,$$

where $W \in \mathbb{R}^{2d \times d}$ is the discretization of the gradient operator, and $h : \mathbb{R}^{2d} \to \mathbb{R}$ is the sum of the pixel-wise euclidean norms.

Numerical results

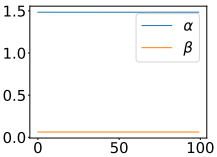


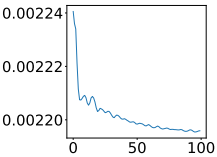
Ground truth | Observed | Reconstructed

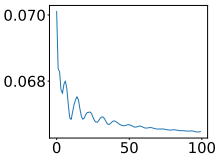
Numerical results

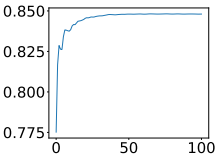
Ground truth | Observed | Reconstructed

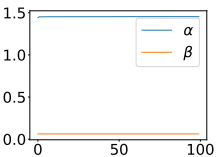
Parameters evolution

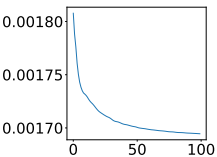


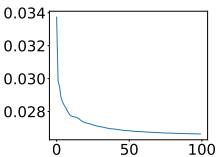


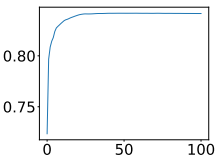






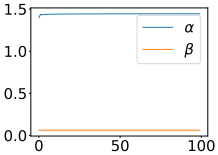


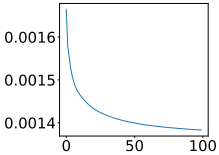


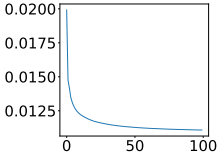


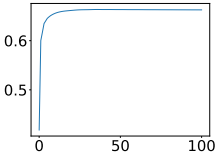
Step sizes (α, β) | Reg. param. (λ) | Prec. param. (ν) | SSIM

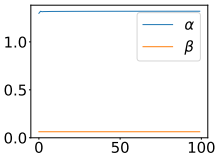
Parameters evolution

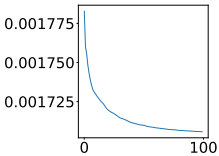


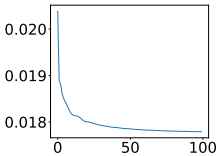


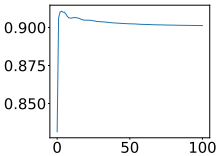












Step sizes (α, β) Reg. param. (λ) Prec. param. (ν) SSIM

Future work

- Use different designs for the predictor network $\mathcal{N}$.

Future work

- Use different designs for the predictor network $\mathcal{N}$.
- Input more information about the current iteration to the predictor, e.g. the dual variable v^0 of the inner loop or the residual.

Future work

- Use different designs for the predictor network $\mathcal{N}$.
- Input more information about the current iteration to the predictor, e.g. the dual variable v^0 of the inner loop or the residual.
- Test weighting strategies for the loss function.

Future work

- Use different designs for the predictor network $\mathcal{N}$.
- Input more information about the current iteration to the predictor, e.g. the dual variable v^0 of the inner loop or the residual.
- Test weighting strategies for the loss function.
- Refine training to avoid local minima and improve stability.

Future work

- Use different designs for the predictor network $\mathcal{N}$.
- Input more information about the current iteration to the predictor, e.g. the dual variable v^0 of the inner loop or the residual.
- Test weighting strategies for the loss function.
- Refine training to avoid local minima and improve stability.
- Test the unrolling architecture on different problems, algorithms, and regularization terms.

Future work

- Use different designs for the predictor network $\mathcal{N}$.
- Input more information about the current iteration to the predictor, e.g. the dual variable v^0 of the inner loop or the residual.
- Test weighting strategies for the loss function.
- Refine training to avoid local minima and improve stability.
- Test the unrolling architecture on different problems, algorithms, and regularization terms.

References

- S. Aleotti, M. Donatelli, R. Krause, G. Scarlato
  A Preconditioned Version of a Nested Primal-Dual Algorithm for Image Deblurring
  J. Sci. Comput. 103, 85 (2025)
- git repository for the PNPD unrolling codes
  `https://github.com/Giuseppe499/PNPD_unrolling`
- git repository for the PNPD codes
  `https://github.com/Giuseppe499/PNPD`

Thank you for your attention!