Unrolled deep networks for sparse signal restoration

Mouna Gharbi, Silvia Villa, Emilie Chouzenoux, Jean-Christophe Pesquet

UNROLLED DEEP NETWORKS FOR SPARSE SIGNAL RESTORATION

Mouna GHARBI[†]*, Silvia VILLA*[*]*, Emilie CHOUZENOUX*[†]*, Jean-Christophe PESQUET*[†]

[†] CVN, CentraleSupélec, Inria Saclay, University Paris Saclay, France.
[*] MaLGa, DIMA, Università degli Studi di Genova, Via Dodecaneso 35, 16146 Genova, Italy.

ABSTRACT

This paper addresses the problem of sparse signal recovery by deep unrolling approaches. The task of sparse restoration from linearly transformed and noisy observations occurs in many applied fields, including geoscience, biochemistry, remote sensing, and biomedical data processing, and was thoroughly studied in literature. Recently, novel approaches based on 'deep unrolling' or 'deep unfolding', have been introduced. They consist in creating deep neural networks inspired from iterative algorithms initially built for penalized loss minimization. The iterations of the algorithm are recast as neural network layers. The use of deep learning frameworks ensures an efficient implementation and the possibility to learn the algorithm native hyperparameters, through the minimization of a task-oriented loss. However, for a given application, choosing an adequate iterative scheme to unroll and fine-tuning the architecture parameters remains a challenging task. In this work, our goal is to present a comprehensive comparative study of deep unrolled approaches deployed for sparse signal reconstruction. Three architectures are introduced and compared through a motivating application, arising in analytical chemistry. A reproducible Github code is provided.

1. INTRODUCTION

Sparse signal restoration amounts to retrieving original data from corrupted observations, under the assumption that the sought solution is a sparse vector characterised by a small number of nonzero entries. Measurements $z \in \mathbb{R}^m$ are related to groundtruth spike signal $\bar{x} \in \mathbb{R}^n$ through the model

$$z = H\bar{x} + \varepsilon, \tag{1}$$

where $H \in \mathbb{R}^{m\times n}$ is a linear operator simulating the acquisition model, for instance a convolution, and $\varepsilon \in \mathbb{R}^m$ is the realization of a noise, here assumed additive zero-mean i.i.d. Gaussian. The signal recovery problem consisting of estimating $\bar{x}$ from z is an inverse problem that was tackled thoroughly in literature. A known approach relies on minimizing a designed objective using iterative algorithms. For instance, in explicit regularization [1], a cost function composed of a data fidelity term and a prior term balanced by a regularization hyperparameter is minimized. Regularization is incorporated into the problem to mend the ill-posedness of the problem, by enforcing sparsity in the recovered estimate. For instance, ℓ_1 norm, ℓ_0 quasi-norm or their smooth approximations can be used. Another type of regularization consists in iterative or implicit regularization [2] in which the implicit bias of first-order algorithms is exploited to favor desirable properties of the solution. In this case, the number of iterations plays the role of the regularization parameter. Solving inverse problems like (1) has also been addressed by supervised learning methods. Common deep networks (such as CNNs) have been used for deblurring [3], denoising [4], and signal reconstruction [5]. Such models are trained on large annotated databases, to learn a mapping between inputs (degraded data) and outputs (original data), to be used on test data. Lately, a new approach called unrolling (or unfolding), combining model-based and data-driven approaches has emerged [6]. The basic idea consists in creating deep networks inspired from iterative algorithms, such that each layer mirrors one iteration of the corresponding algorithm. The benefits are (i) interpretability which contrasts with black-box mainstream models, (ii) computation speed provided by deep learning frameworks such as Pytorch and Tensorflow, (iii) efficiency through supervised and task-oriented tuning of the algorithm native hyperparameters. Many works have explored unrolling of various minimiziation algorithms, such as gradient descent [7], ISTA [8], and proximal interior methods [9]. In [10], learnt denoisers were involved in a half-quadratic splitting scheme. Bayesian approaches have also recently been combined with unrolling in [11] and [12] unrolled proximal splitting algorithm for tomographic reconstruction. The deployment of the unrolling principle involves several prerequisites that might be application dependent such as the optimization problem formulation, the algorithm to unroll, the parameters to untie, etc. Several recent benchmark studies have been made in the context of image processing, including image restoration [13] and MRI reconstruction [14], but 1D signal restoration remains scarcely studied in that respect. In this work, we aim to provide a comprehensive comparative benchmarking of deep unrolled methods for solving Problem (1). We introduce three unrolled architectures, re-

lying on distinct optimization problem formulation and resolution schemes. The first one relies on a primal-dual algorithm [15] using an ℓ_1-based constrained formulation inspired from the implicit regularization framework. Up to our knowledge, this is the first instance of unrolling such implicit formulation. The next two are inspired from explicit regularization schemes, as more classically done in unrolling: iterative soft thesholding (ISTA) [16] and the half-quadratic (HQ) algorithm we recently introduced in [17], that addresses a smoothed penalized least-squares formulation. We compare the three iterative (i.e., unsupervised) and unrolled (i.e., supervised) implementations on several datasets from an applicative sparse signal restoration problem arising in mass spectrometry (MS) [18]. The paper is organized as follows: in Section 2, we introduce the mathematical notation and definitions useful for the rest of the work. Section 3 describes our construction for the three deep unrolled methods, devising the algorithms, designing the architectures, and detailing our strategy to learn the desired hyperparameters for each approach. Section 4 summarizes our experimental settings, and presents our comparisons and analysis. Lastly, Section 5 draws conclusions of the work.

2. MATHEMATICAL NOTATION

We adopt convex analysis notations from [19]. Let $\mathscr{H}$ be a Hilbert space, the subdifferential of a function $f : \mathscr{H} \to \mathbb{R}$ at point x is the set valued operator $\partial f : \mathscr{H} \to 2^{\mathscr{H}}$ such that $\partial f(x) = \{u \in \mathscr{H} | (\forall y \in \mathscr{H}) \langle y - x | u \rangle + f(x) \leq f(y)\}$. The identity operator on $\mathscr{H}$ is denoted by Id. The set of proper, convex and lower semi continuous (l.s.c) functions is denoted by $\Gamma_0(\mathscr{H})$ and the proximity operator is defined for every $x \in \mathscr{H}$ by $\mathrm{prox}_f(x) = \mathrm{argmin}_{u \in \mathscr{H}} f(u) + \frac{1}{2}\|x-u\|_2^2$. Finally, the convex conjugate of a function is denoted, for every $x \in \mathscr{H}$, by $f^*(x) = \sup_{u \in \mathscr{H}} \langle x | u \rangle - f(u)$. For a given convex closed non empty set C, ι_C is the indicator function defined, for every $x \in \mathscr{H}$, by $\iota_C(x) = \begin{cases} 0, & \text{if } \quad x \in C \\ +\infty & \text{otherwise} \end{cases}$. The support function $\sigma_C = \iota_C^*$ is defined for all $x \in \mathscr{H}$ by $\sigma_C(x) = \sup_{x' \in C} \langle x' | x \rangle$. We finally denote by $\mathscr{B}(z, \rho)$ the ℓ_2 ball of center z and radius ρ whose expression is $\mathscr{B}(z, \rho) = \{z' \in \mathscr{H} \mid \|z' - z\|_2 \leq \rho\}$.

3. PROPOSED METHOD

In this section, we present three unrolled architectures, built upon primal-dual, ISTA and half-quadratic iterative schemes to solve (1). For each approach, (i) we formulate the optimization problem to be solved, (ii) we derive the iterative algorithm to solve it, and (iii) we propose the corresponding unrolled neural network by reinterpreting the algorithm iterations into layers, so allowing us to learn certain hyperparameters.

3.1. Unrolled primal-dual architecture

3.1.1. Problem statement

Let us define $\widehat{x} \in \mathbb{R}^n$ an estimate of $\bar{x}$ obtained by solving the constrained minimization problem

$$\widehat{x} \in \underset{x \in \mathbb{R}^n}{\mathrm{argmin}} \|x\|_1 \quad \text{s.t.} \quad \|Hx - z\|_2 \leq \rho. \tag{2}$$

Hereabove, $\|\cdot\|_1$ is ℓ_1 norm which aims to enforce a sparsity prior on the estimated solution and $\rho > 0$ is a hyperparameter related to the noise level. In the case of an i.i.d. Gaussian noise with standard deviation σ, a common setting arising from the law of large numbers is $\rho = \sqrt{n}\sigma$. The minimization problem (2) can be solved using a proximal splitting primal-dual algorithm [15] which we describe in the next section. The optimal solution is retrieved by combining the explicit regularization induced by ρ with the implicit regularization given by the early stopping of the iterations [20].

3.1.2. Primal-dual algorithm

Problem (2) can be recast into an unconstrained form which is the primal problem

$$\min_{x \in \mathbb{R}^n} \quad \Big(\iota_{\mathscr{B}(z,\rho)}(Hx) + \|x\|_1 \Big), \tag{3}$$

The associated dual problem [15] is

$$\max_{y \in \mathbb{R}^m} \quad -\Big(\sigma_{\mathscr{B}(z,\rho)}(y) + \| - H^\top y\|_1 \Big). \tag{4}$$

The general saddle point problem is

$$\min_{x \in \mathbb{R}^n} \max_{y \in \mathbb{R}^m} \quad \langle Hx, y \rangle + \|x\|_1 - \sigma_{\mathscr{B}(z,\rho)}(y). \tag{5}$$

To solve (5), the primal-dual approach initially proposed in [15] can be used, leading to Algorithm 1. Convergence of the sequence $(x_k)_{k \in \mathbb{N}}$ to a solution of (2) is established, for γ, τ positive stepsizes verifying $\gamma\tau \leq \frac{1}{\|H\|_2^2}$.

Algorithm 1 Primal-dual algorithm

Init: Choose $\tau, \gamma > 0$, $(x_0, y_0) \in \mathbb{R}^n \times \mathbb{R}^m$ and $y_0 = y_{-1}$.
for $k = 0, 1, \ldots$ **do**
  $\tilde{y}_k = 2y_k - y_{k-1}$
  $x_{k+1} = \mathrm{prox}_{\tau\|\cdot\|_1}(x_k - \tau H^\top \tilde{y}_k)$
  $y_{k+1} = \mathrm{prox}_{\gamma\sigma_{\mathscr{B}(z,\rho)}(\cdot)}(y_k + \gamma H x_{k+1})$
end for

We now give explicitly the expressions of the proximity operators involved in Algorithm 1. First,

$$(\forall x = (x_i)_{1 \leq i \leq n} \in \mathbb{R}^n) \quad \mathrm{prox}_{\tau\|\cdot\|_1}(x) = (\mathrm{sign}(x_i) \max(|x_i| - \tau, 0))_{1 \leq i \leq n}. \tag{6}$$

Second, using Moreau's decomposition in [19, Theorem 14.3 (ii)], for every $\gamma > 0$,

$$(\forall y \in \mathbb{R}^m) \quad \operatorname{prox}_{\gamma \sigma_{\mathscr{B}(z,\rho)}}(y) = y - \gamma P_{\mathscr{B}(z,\rho)}\left(\gamma^{-1} y\right). \quad (7)$$

Hereabove, $P_{\mathscr{B}(z,\rho)}$ is the projection onto the ball $\mathscr{B}(z,\rho)$ of center z and radius ρ,

$$(\forall y \in \mathbb{R}^m)\ P_{\mathscr{B}(z,\rho)}(y) = \begin{cases} y & \text{if} \quad y \in \mathscr{B}(z,\rho) \\ z + \frac{y-z}{\|y-z\|_2}\rho & \text{if} \quad y \notin \mathscr{B}(z,\rho) \end{cases}. \quad (8)$$

Finally, this leads, for every $y \in \mathbb{R}^m$, and for every $\gamma > 0$, to

$$\operatorname{prox}_{\gamma \sigma_{\mathscr{B}(z,\rho)}}(y) = \begin{cases} 0, & \text{if} \quad \gamma^{-1} y \in \mathscr{B}(z,\rho) \\ y - \gamma(z + \frac{y-\gamma z}{\|y-\gamma z\|_2}\rho) & \text{if} \quad \gamma^{-1} y \notin \mathscr{B}(z,\rho) \end{cases}. \quad (9)$$

3.1.3. Unrolled architecture

To create a deep neural network architecture, unrolling starts by converting, for a fixed number of layers K, the concerned algorithm iterations into layers. In our case, the primal and dual iterations in Algorithm 1 reveal building blocks of feed-forward network structures. Therefore, we propose the following multi-branch architecture: for $k \in \{0,\dots,K-1\}$, the primal $\mathscr{L}_{p_k}$ and dual $\mathscr{L}^z_{d_k}$ branches process respectively the sequences $(x_k)_{0\le k\le K}$ and $(y_k)_{0\le k\le K}$ as follows:

$$\Big(x_{k+1}, y_{k+1}\Big) = \Big(\mathscr{L}_{p_k}(x_k, y_k, y_{k-1}), \mathscr{L}^z_{d_k}(\mathscr{L}_{p_k}(x_k, y_k, y_{k-1}), y_k)\Big). \quad (10)$$

For the sake of short notation we denote by $\mathscr{L}^z_k$, for $k \in \{0,\dots,K-1\}$, a global layer of our proposed architecture. It depends on the noisy observation given the dependence of the dual branch on z. It takes as input the couple (x_y, y_k) and returns as output the couple (x_{k+1}, y_{k+1}) except the last layer which only returns x_K. In that case, (10) becomes

$$\Big(x_{k+1}, y_{k+1}\Big) = \mathscr{L}^z_k(x_k, y_k) \quad (11)$$

such that,

$$\begin{cases} x_{k+1} = \mathscr{A}_{p_k}(W_{p_k} x_k + b_{p_k}) \\ y_{k+1} = \mathscr{A}_{d_k}(W_{d_k} y_k + b_{d_k}) \end{cases}. \quad (12)$$

Here, $\mathscr{A}_{p_k}$ and $\mathscr{A}_{d_k}$ are primal and dual activation functions:

$$(\forall x \in \mathbb{R}^n) \quad \mathscr{A}_{p_k}(x) = \operatorname{prox}_{\tau_k \|\cdot\|_1}(x), \quad (13)$$

$$(\forall y \in \mathbb{R}^m) \quad \mathscr{A}_{d_k}(y) = \operatorname{prox}_{\gamma_k \sigma_{\mathscr{B}(z,\rho)}(\cdot)}(y). \quad (14)$$

So as to retrieve equivalence with our Algorithm 1, we set the weight matrices W_{p_k} and W_{d_k} equal to the identity of their respective ambient space and bias terms are given by

$$b_{p_k} = -\tau_k H^\top \tilde{y}_k \in \mathbb{R}^n, \quad b_{d_k} = \gamma_k H x_{k+1} \in \mathbb{R}^m. \quad (15)$$

As can be seen in equations (13)-(15), we propose to untie sequences $(\tau_k)_{0\le k\le K-1}$, $(\gamma_k)_{0\le k\le K-1}$ and $(\rho_k)_{0\le k\le K-1}$, allow them to vary along layers and to learn them automatically. To this end, we design network architectures which will be reattached to each layer $\mathscr{L}^z_k$. We enforce positivity on learnable parameters through the ReLU activation function

$$(\forall z = (z_\ell)_{1\le\ell\le m} \in \mathbb{R}^m) \quad \operatorname{ReLU}(z) = (\max(0, z_\ell))_{1\le\ell\le m}. \quad (16)$$

Specifically, let $\theta = \{t_k, s_k, r_k\}_{0\le k\le K-1}$ be the vector of parameters to be learned then, for every $k \in \{0,\dots,K-1\}$,

$$\tau_k = \operatorname{ReLU}(t_k), \quad \gamma_k = \operatorname{ReLU}(g_k), \quad \rho_k = \operatorname{ReLU}(r_k). \quad (17)$$

The global layer architecture is showcased on Fig.1. Once the multi-layer architecture is defined, updating the learnable weights is a straightforward task. Let $\mathscr{S} = \{(\bar{x}_s, z_s) \mid s \in \{1,\dots,S\}\}$ be a training set comprised of S pairs of groundtruth and degraded data samples. Let the prediction function be

$$g^z_\theta(x_{s,0}, y_{s,0}) = \mathscr{L}^z_{K-1} \circ \dots \circ \mathscr{L}^z_k \circ \dots \circ \mathscr{L}^z_0(x_{s,0}, y_{s,0}), \quad (18)$$

$(x_{s,0}, y_{s,0})$ denoting the primal and dual inputs for the first layer of the network for sample s. Then, for a predefined loss ℓ, the optimal hyperparameter setting $\widehat{\theta}$ is the solution to the optimization problem

$$\min_\theta \quad E(\theta) = \frac{1}{S}\sum_{s=1}^{S} \ell(g^{z_s}_\theta(x_{s,0}, y_{s,0}), \bar{x}_s). \quad (19)$$

A common choice for the loss is the mean squared error (MSE) loss obtained by setting $\ell(x,\bar{x}) = \frac{1}{n}\|x-\bar{x}\|^2$. Standard neural network training algorithms such as stochastic gradient or Adam can be used to solve the above problem.

3.2. Unrolled ISTA architecture

3.2.1. Problem formulation

Let $\widehat{x}$ be the minimizer of the following regularized cost function F for estimating $\bar{x}$,

$$\widehat{x} \in \operatorname*{argmin}_{x\in\mathbb{R}^n} \Big(F(x) = \frac{1}{2}\|Hx - z\|_2^2 + \chi\|x\|_1\Big). \quad (20)$$

Hereabove, χ is a regularization hyperparameter balancing the data fidelity term and the non-differentiable ℓ_1 norm aiming to promote the sparsity prior.

3.2.2. Algorithm

To solve (20), ISTA [16], summarized in Algorithm 2 can be used, γ is a stepsize in $]0, 2/\|H\|_2^2[$.

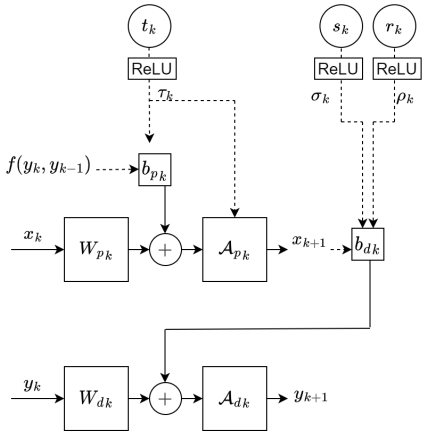

Fig. 1. Overview of one layer $\mathscr{L}_k^z$ of the proposed unrolled primal dual-algorithm. $(W_{pk})_{1\leq k\leq K}$, $(W_{dk})_{1\leq k\leq K}$, $(b_{pk})_{1\leq k\leq K}$, $(b_{dk})_{1\leq k\leq K}$, $(\mathscr{A}_{pk})_{1\leq k\leq K}$ and $(\mathscr{A}_{dk})_{1\leq k\leq K}$ are respectively primal/dual weight operators, biases and activations. $(s_k)_{1\leq k\leq K}$, $(r_k)_{1\leq k\leq K}$, and $(t_k)_{1\leq k\leq K}$ are learnable weights and ReLU is an activation enforcing positivity.

Algorithm 2 ISTA algorithm

Init: Choose $\gamma, \chi > 0$, $x_0 \in \mathbb{R}^n$.
for $k = 0, 1, \ldots$ **do**
  $x_{k+1} = \text{prox}_{\gamma\chi\|\cdot\|_1}(x_k - \gamma H^\top(Hx_k - z))$
end for

3.2.3. Unrolled architecture

As described in Section 3.1.3, after reinterpreting the algorithm initial K iterations as feedforward structures, sequences $(\gamma_k)_{0\leq k\leq K-1}$ and $(\chi_k)_{0\leq k\leq K-1}$ are untied and learnt by enforcing nonnegativity by using ReLU.

3.3. Unrolled half-quadratic architecture

3.3.1. Problem formulation

Lastly, we propose minimizing a differentiable cost function to obtain the estimate $\widehat{x}$ of $\bar{x}$. The objective function is composed of a data fidelity term and a regularization function balanced by a regularization hyperparameter λ, so yielding

$$\widehat{x} \in \underset{x\in\mathbb{R}^n}{\text{argmin}} \left(F(x) = \frac{1}{2}\|Hx - z\|_2^2 + \lambda\Psi(x) \right), \qquad (21)$$

with

$$(\forall x \in \mathbb{R}^n) \quad \Psi(x) = \sum_{i=1}^{n} \psi(x_i). \qquad (22)$$

The i-th component of a vector x in $\mathbb{R}^n$ is denoted by x_i with $i \in \{1, \ldots, n\}$ and $\psi : \mathbb{R} \to [0, +\infty)$ is defined as,

$$(\forall t \in \mathbb{R}) \quad \psi(t) = \delta\left(|t| - \delta \log\left(\frac{|t|}{\delta} + 1 \right) \right). \qquad (23)$$

The above penalty, known as the Fair potential, approximates the ℓ_1 norm, depending on a smoothing hyperparameter $\delta > 0$.

3.3.2. Algorithm

The optimization problem (21) is minimized by a half-quadratic algorithm [21], given in Algorithm 3. For every $k \in \mathbb{N}$, $A(x_k)$ is a symmetric definite positive matrix ensuring a majorizing property on the curvature of F. Explicit expressions of $\nabla F(x_k)$ and $A(x_k)$ for Problem (1) can be found in [17].

Algorithm 3 Half-quadratic algorithm

Init: Choose $x_0 \in \mathbb{R}^n$.
for $k = 0, 1, \ldots$ **do**
  Build majorant metric $A(x_k)$,
  $x_{k+1} = x_k - A(x_k)^{-1}\nabla F(x_k)$.
end for

3.3.3. Unrolled architecture

As for unrolled HQ, we learn $(\lambda_k)_{1\leq k\leq K}$ according to the paradigm described in [17].

4. EXPERIMENTAL RESULTS

In this section we present our experimental results through an application arising from mass spectrometry (MS) [22] signal restoration. MS is a chemical tool used for detection and quantification of molecules of interest. It produces spectra with positively valued peaks distributed according to the isotopic distribution and charge of studied molecule. We consider here the problem of deblurring and denoising MS data.

4.1. Datasets

We create two datasets of realistic MS signals, generated using the Averagine model as in [18]. First, for a given number of proteins $p = 10$, a 2000 dimensional signal with p randomly uniformly placed peaks between index 50 and 1950 is created. The associated positive intensities are randomly defined as the absolute value of a Gaussian realization with mean 10 and standard deviation 100. This signal is later multiplied by the MS averagine dictionary gathering isotopic distribution of a predefined list of atomic mass. This results in the groundtruth signal $\bar{x}$. Model (1) is then used to create the associated degraded signal z. Matrix H mimicks the convolution (with circular-padding) with a centered Gaussian shape with support size 50 and standard deviation 1. For Dataset 1, the added Gaussian noise $\varepsilon \in \mathbb{R}^m$ is i.i.d., zero mean with standard deviation $\sigma = 2$. From the same set of groundtruth signals $\bar{x}$, we furthermore create Dataset 2, where z is built

from an i.i.d. zero-mean Gaussian noise with standard deviation σ randomly chosen between 0 and 2. We create respectively training, validation, and test sets of sizes 1000, 200 and 100 for both datasets.

4.2. Iterative methods

We compare unrolled methods described in Section 3 to their respective iterative optimization approaches. First, we run the primal-dual algorithm described in Algorithm 1. The stepsizes γ and τ are tuned following the strategy proposed in [20] such that $\tau\gamma = \frac{0.99}{\|H\|_2^2}$ and $\tau = \gamma$. The bound ρ is finetuned manually. Specifically, for Dataset 1, we assume the noise level to be known, and we tune ρ using a predefined grid of values including $\sigma\sqrt{n}$. For Dataset 2, as the noise level is varying, we propose to estimate it through $\hat{\sigma}(y) = \frac{\text{median}(|W_H y|)}{0.6745}$ as done in [9] for each degraded observation y with $|W_H y|$ the vector gathering the absolute value of the diagonal coefficients of the first level of the Haar wavelet decomposition of y. We then set $\rho = \hat{\sigma}(y)\tilde{\rho}$ with $\tilde{\rho}$ finetuned through gridsearch. For initialization we use $x_0 = y_0 = y_{-1}$. The set of parameters providing the lowest MSE on the training set is retained, and per-used on the test set. For ISTA, we adopt again different strategies for Dataset 1 or 2. Namely, we directly finetuned χ for the former, while we set $\chi = \tilde{\chi}\hat{\sigma}(y)$ with $\tilde{\chi}$ to be tuned for the latter. The last iterative-based method we compare with is the HQ algorithm [17], in which we employ gridsearch to determine the described parameters (λ, δ).

4.3. Training settings

The mean-squared error (MSE) loss is used for training all unrolled methods as well as to finetune parameters of optimization-based methods. In all experiments, training and validation batch sizes are set to 5. Learnable weights are tuned through backpropagation using Adam, where the learning rate (lr) is set according to the experiment to maintain a stable training. As entry for the multiple architectures, we use the null vector, namely for unrolled HQ and ISTA, the input is $x_0 = 0$ and for unrolled primal dual $x_0 = y_0 = y_{-1} = 0$. To set the number of layers, we train the unrolled architectures for several choices of K. Then we choose the number of layers returning a minimal loss on validation set while maintaining a reasonable training execution time. We summarize in the table hereafter the learning rate and the number of layers used for each experiment.

	Dataset1	**Dataset2**
Unrolled ISTA	$(10^{-4}, 14)$	$(10^{-3}, 14)$
Unrolled Primal Dual	$(10^{-4}, 22)$	$(10^{-4}, 16)$
Unrolled HQ	$(10^{-2}, 8)$	$(10^{-2}, 8)$

4.4. Numerical results

Table 1 summarizes all our results. Unrolled architectures outperform their iterative counterparts in terms of average restoration quality (MSE). Moreover, Dataset 2 has better results, since the noise level is lower in average. While iterative HQ yields the best MSE, its unrolled version is slightly disadvantaged. This might show that learning only the regularization parameter was not enough to reach full representation capacity for the network. In fact, as shown on Fig. 2 (upper left zoom), unrolled HQ tends to underestimate peak intensities which is assessed by lower truncated signal-to-noise ratio (TSNR, [18]) scores. We also study peak detection performance of the methods. We define the presence/absence of peaks using a minimal intensity threshold of 0.8, and compute the F1 score between the ground truth and estimated binary labels. Iterative methods appear rather good at detecting peak locations, in particular on Dataset 2. Unrolled HQ is competitive with other unrolled architectures on this metric. Inference time of unrolled methods is lower than that of iterative methods, since the number of layers is considerably lower than typical iteration numbers. Among all unrolled approaches, half-quadratic exhibits the highest execution time, due to costly matrix inversions.

5. CONCLUSIONS

This work proposes a comprehensive study of three deep unrolling approaches for solving a sparse inverse problem arising in spectroscopy. Our experiments show the competitiveness of all unrolling paradigms with respect to their respective iterative counterparts. Surprisingly, the best iterative method does not necessarily yield an optimal unrolling architecture, which confirms previous studies in the field of image processing. The final choice remains highly dataset and task dependent, and benchmarking remains necessary. To that end, we provide the readers with a reproducible code for the presented datasets and architectures, available at `https://github.com/GHARBIMouna/Deep-Unrolling.git`.

6. REFERENCES

[1] J. Honerkamp and J. Weese, "Tikhonovs regularization method for ill-posed problems: A comparison of different methods for the determination of the regularization parameter," *Continuum Mechanics and Thermodynamics*, vol. 2, pp. 17–30, 1990.

[2] B. Kaltenbacher, A. Neubauer, and O. Scherzer, "Iterative regularization methods for nonlinear ill-posed problems," in *Iterative Regularization Methods for Nonlinear Ill-Posed Problems*. de Gruyter, 2008.

[3] C.-J. Schuler, M. Hirsch, S. Harmeling, and B. Schölkopf, "Learning to deblur," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 7, pp. 1439–1451, 2015.

[4] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process*, vol. 26, no. 7, pp. 3142–3155, 2017.

	Dataset 1				Dataset 2			
Name	MSE (std)	TSNR (std)	Time (s.)	F1	MSE (std)	TSNR (std)	Time (s.)	F1
Primal-dual	3.14 (1.15)	3.88 (0.08)	5.27	0.72 (0.05)	1.77 (1.02)	7.07 (0.15)	5.60	0.85 (0.09)
ISTA	3.07 (1.16)	3.96 (1.16)	3.54	0.74 (0.05)	1.84 (1.17)	7.19 (1.17)	2.72	0.86 (0.07)
HQ	0.92 (0.28)	9.99 (0.86)	1.03	0.62 (0.05)	0.62 (0.19)	11.43 (1.12)	0.75	0.77 (0.14)
Unrolled primal-dual	0.34 (0.10)	13.41 (1.59)	0.04	0.80 (0.05)	0.19 (0.11)	16.57 (2.50)	0.03	0.84 (0.05)
Unrolled ISTA	0.32 (0.08)	13.68 (1.61)	0.008	0.81 (0.05)	0.16 (0.09)	17.37 (2.70)	0.008	0.84 (0.04)
Unrolled HQ	0.42 (0.09)	12.68 (1.47)	0.17	0.80 (0.05)	0.20 (0.12)	16.29 (2.53)	0.17	0.84 (0.05)

Table 1. Comparative performance: MSE, TSNR, F1 scores (mean and std), and averaged test time for Datasets 1 and 2.

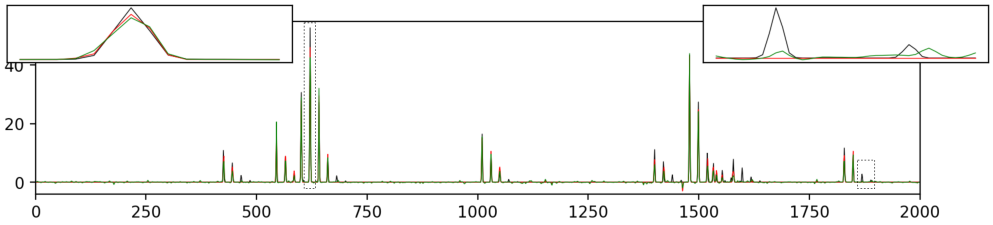

Fig. 2. Superposed groundtruth data $\bar{x}$ (black), estimate $\hat{x}$ using unrolled ISTA (red) and estimate $\hat{x}$ using unrolled HQ (green).

[5] M.-T. McCann, K.-H. Jin, and M. Unser, "Convolutional neural networks for inverse problems in imaging: A review," *IEEE Signal Process Mag*, vol. 34, no. 6, pp. 85–95, 2017.

[6] V. Monga, Y. Li, and Y.-C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," *IEEE Signal Process Mag*, vol. 38, no. 2, pp. 18–44, 2021.

[7] D. Ren, W. Zuo, D. Zhang, L. Zhang, and M.-H. Yang, "Simultaneous fidelity and regularization learning for image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 284–299, 2019.

[8] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proceedings of the 27th International Conference on International Conference on Machine Learning (ICML 2010)*, 2010, pp. 399–406.

[9] 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.

[10] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep CNN denoiser prior for image restoration," in *Proceedings of the 32nd Conference on Computer Vision and Pattern Recognition (CVPR 2017)*, 2017, pp. 3929–3938.

[11] Y. Huang, E. Chouzenoux, and J.-C. Pesquet, "Unrolled variational Bayesian algorithm for image blind deconvolution," *IEEE Trans. Image Process*, vol. 32, pp. 430–445, 2022.

[12] M. Savanier, E. Chouzenoux, J.-C. Pesquet, and C. Riddell, "Deep unfolding of the DBFB algorithm with application to ROI CT imaging with limited angular density," *arXiv preprint arXiv:2209.13264*, 2022.

[13] H.T.-Vy. Le, N. Pustelnik, and M. Foare, "The faster proximal algorithm, the better unfolded deep learning architecture ? the study case of image denoising," in *Proceedings of the 30th European Signal Processing Conference (EUSIPCO 2022)*, 2022, pp. 947–951.

[14] Z. Ramzi, P. Ciuciu, and J.-L. Starck, "Density compensated unrolled networks for non-Cartesian MRI reconstruction," in *Proceedings of 18th IEEE International Symposium on Biomedical Imaging (ISBI 2021)*, 2021, pp. 1443–1447.

[15] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *J. Math. Imaging Vision*, vol. 40, no. 1, pp. 120–145, 2011.

[16] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm with application to wavelet-based image deblurring," in *Proceedings of 34th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2009)*, 2009, pp. 693–696.

[17] M. Gharbi, E. Chouzenoux, J.-C. Pesquet, and L. Duval, "GPU-based implementations of MM algorithms. application to spectroscopy signal restoration," in *Proceedings of the 29th European Signal Processing Conference (EUSIPCO 2021)*, 2021, pp. 2094–2098.

[18] A. Cherni, E. Chouzenoux, L. Duval, and J.-C. Pesquet, "SPOQ ℓ_p-over-ℓ_q regularization for sparse signal recovery applied to mass spectrometry," *IEEE Trans. Signal Process*, vol. 68, pp. 6070–6084, 2020.

[19] H.-H. Bauschke and Patrick L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, Springer Publishing Company, Incorporated, 2nd edition, 2017.

[20] C. Molinari, M. Massias, L. Rosasco, and S. Villa, "Iterative regularization for low complexity regularizers," *arXiv preprint arXiv:2202.00420*, 2022.

[21] M. Allain, J. Idier, and Y. Goussard, "On global and local convergence of half-quadratic algorithms," *IEEE Trans. Image Process*, vol. 15, no. 5, pp. 1130–1142, 2006.

[22] L.-M Heaney, D.-JL Jones, and T. Suzuki, "Mass spectrometry in medicine: a technology for the future?," 2017.