LEARNING THE WEIGHT MATRIX FOR SPARSITY AVERAGING IN COMPRESSIVE IMAGING

Dimitris Perdios*, Adrien Besson*, Philippe Rossinelli†, and Jean-Philippe Thiran*†

*Signal Processing Laboratory (LTS5), Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland
†Department of Radiology, University Hospital Center (CHUV), Switzerland

ABSTRACT

We propose to map the fast iterative soft thresholding algorithm to a deep neural network (DNN), with a sparsity prior in a concatenation of wavelet bases, in the context of compressive imaging. We exploit the DNN architecture to learn the optimal weight matrix of the corresponding reweighted $\ell_1$-minimization problem. We later use the learned weight matrix for the image reconstruction process, which is recast as a simple $\ell_1$-minimization problem. The approach, denoted as Learned-Extended-FISTA, shows promising results in terms of image quality, compared to state-of-the-art algorithms, and significantly reduces the reconstruction time required to solve the reweighted $\ell_1$-minimization problem.

Index Terms— Compressed sensing, deep learning, fast iterative soft thresholding

1. INTRODUCTION

Compressed sensing (CS) extends the principle of Nyquist sampling to non-bandlimited signals exploiting the idea that most signals have concise representations, expressed in terms of sparsity, in well-chosen models [1, 2]. Let us denote as $x \in \mathbb{R}^N$ the signal under scrutiny, measured with a linear operator $A \in \mathbb{R}^{M \times N}$ such that $y = Ax + n$, where $n \in \mathbb{R}^M$ is the observation noise. Let us assume that $x$ obeys a sparse representation in a basis $\Psi \in \mathbb{R}^{N \times N}$, i.e. that $x = \Psi c$ with $||c||_0 = K < N$. For a sufficient number of measurements $M$, CS demonstrates that one can perfectly recover $x$ from $y$, with high probability, by solving the following analysis problem:

$$\min_{x \in \mathbb{R}^N} \lambda ||\Psi^T x||_1 + \frac{1}{2} ||Ax - y||_2^2,$$

(1)

where $\Psi^T$ designates the adjoint of $\Psi$ and $||.||_0$ denotes the $\ell_0$-norm calculated as $||x||_0 = \sqrt{\sum_{i=1}^{N} |x_i|^2}$. It has been demonstrated that Gaussian and Bernoulli random matrices are particularly suited to the CS framework. In this specific case, sampling rates requirements scale logarithmically with the size of the signal [3].

Candès et al. have later extended the CS framework to coherent and redundant dictionaries $\Psi$ [4] and have shown that solving a reweighted $\ell_1$-minimization problem may improve the signal reconstruction [5]. Carrillo et al. have used this framework for compressive imaging by exploiting sparsity of images in multiple mutually coherent wavelet bases [6]. Their technique, coined as sparsity averaging for reweighted analysis (SARA) aims at solving the following problem:

$$\min_{x \in \mathbb{R}^N} ||W\Psi^T x||_1 + \frac{1}{2} ||Ax - y||_2^2,$$

(2)

where $\Psi \in \mathbb{R}^{N \times L}$, $L = qN$ is a concatenation of $q$ bases $\Psi_q$ and $W \in \mathbb{R}^{L \times L}$ is a block-diagonal matrix made of $q$ blocks of size $N \times N$ with positive entries. Carrillo et al. have shown that SARA outperforms many state-of-the-art image reconstruction algorithms.

The tremendous success of deep learning (DL) approaches for many image processing tasks has lead researchers to study in what extent DL may perform signal and image reconstruction. Gregor and LeCun [7] have bridged the gap between DL and CS by introducing the learned iterative soft thresholding algorithm (LISTA), in which the classical iterative soft thresholding algorithm (ISTA) is mapped onto a deep neural network (DNN). By learning the optimal parameters of the algorithm on a training set, LISTA outperforms classical ISTA and fast ISTA (FISTA). Kamilov and Mansour have recently extended the work of Gregor and LeCun by learning the non-linearity of LISTA, based on a B-spline decomposition [8]. Yang et al. have applied the same procedure to map the alternating direction method of multipliers (ADMM) to a DNN and have shown promising results in terms of reconstruction time and image quality for CS in magnetic resonance imaging [9].

An alternative to the methods mentioned above consists in training classical DNN architectures, usually used to perform classification, for image reconstruction. Based on the success of autoencoders for non-linear image classification, Mousavi et al. [10] have applied a 3-layers stacked denoising autoencoder for CS and have shown that such a DNN architecture is competitive against state-of-the-art techniques. Kulkarni et al. have adapted a convolutional neural network architecture to image reconstruction and have demonstrated a significant increase of the image quality compared to classical techniques [11].

In this work we propose an alternative to SARA proposed by Carrillo et al.. The underlying idea is that one way to accelerate the algorithm may consist in learning the matrix $W$ from a training step, using a DNN, and later use it for the image reconstruction, instead of alternating between an update step and an optimization step. To this aim, we suggest to use FISTA as a DNN allowing us to use backpropagation algorithms to learn the optimal weight matrix.

The remainder of the paper is organized as follows. The proposed approach is described in Section 2 and compared against SARA and other state-of-the-art algorithms in Section 3. Concluding remarks are given in Section 4.

2. THE PROPOSED APPROACH

2.1. Mapping the fast iterative soft thresholding algorithm to a deep neural network

In order to achieve the mapping, we need to unfold FISTA, described in Algorithm 1, in a similar manner to Gregor and LeCun [7].
The objective of the learning procedure consists in finding $W$. In order to learn the weight matrix, we consider a training set made of $2L$ pairs $(y_p, x_p^*)_{p=1}^{2L}$ where $x_p^*$ denotes the reference image and $y_p$ the corresponding measurement vector. Let us denote by $\hat{x}_p = f(y_p, W)$ the estimate of $x_p^*$ obtained by applying FISTA. The objective of the learning procedure consists in finding $W$ which minimizes the following $\ell_2$-loss function:

$$
\mathcal{L} \left( (y_p, x_p^*)_{p=1}^{2L}, W \right) = \frac{1}{P} \sum_{p=1}^{2L} ||\hat{x}_p - x_p^*||^2.
$$

To achieve such a minimization, the backpropagation-through-time (BPTT) algorithm is used, where the gradient of $\mathcal{L}$ over each diagonal element of $W$ is calculated using the chain rule. Using the same notations as the one described in the paper of Gregor and LeCun, i.e. by defining as $\delta x$ the gradient $\nabla \mathcal{L}$, as $\delta W$ the gradient of $\mathcal{L}$ over the diagonal elements of $W$ and as $\text{soft}(x; \tau)$ the Jacobian of the shrinkage operator with respect to its inputs (diagonal matrix), Algorithm 3 can be derived.

**Algorithm 1** FISTA used to solve Problem (2)

Require: $A, W, \Psi, y, x_0, i = 2, L \geq 2\lambda_{max} (A^T A), t_1 = 1$

input: $c_1 = x_0, x_1 = \text{prox}_{||.||_1} (c_1 + \frac{1}{\tau} A^T (y - Ac_1); \tau)$

repeat

$t_i \leftarrow \frac{1+\sqrt{1+4t_{i-1}^2}}{2}
\alpha_i \leftarrow \frac{t_i-1}{t_i}, \beta_i \leftarrow 1 + \alpha_i$
$c_i \leftarrow x_{i-1} + \left( \frac{t_{i-1}-1}{t_{i-1}} \right) (x_{i-1} - x_{i-2})$
$x_i \leftarrow x_{i-1} + \frac{\alpha_i}{\beta_i} A^T (y - Ac_i); \tau$
$i \leftarrow i + 1$

until stopping criterion

return $x_i$

In Algorithm 1, $\text{prox}_{||.||_1}$ accounts for the proximity operator of the function $x \rightarrow ||W^T x||_1$. Such a proximity operator can be rewritten as follows [12]:

$$
\text{prox}_{||.||_1} (x; \tau) = \sum_{k=1}^{q} \frac{\psi_k}{\sqrt{q}} \text{soft} \left( \frac{\psi_k}{\tau} x: \tau W_k \right),
$$

where $W_k$ accounts for the $k^{th}$ block of $W$ and $\text{soft}(x; \tau W)$, usually recalled as the shrinkage operator, is defined for each element as:

$$
\text{soft}(x_i; \tau W_i) = \text{sign} (x_i) (|x_i| - \tau W_i) + .
$$

We introduce the matrices $S = (1 - \frac{1}{\tau} A^T A)$ and $G = \frac{1}{\tau} A^T$ and use Equation (3) to unfold FISTA, resulting in Algorithm (2).

**Algorithm 2** LEFISTA forward propagation

Require: $L \geq 2\lambda_{max} (A^T A), W, \Psi, y, T, q$

initialization: $i = 1, t_0 = 1, x_{-1} = x_0 = 0$

repeat

$t_i \leftarrow \frac{1+\sqrt{1+4t_{i-1}^2}}{2}
\alpha_i \leftarrow \frac{t_i-1}{t_i}, \beta_i \leftarrow 1 + \alpha_i$
$z_i \leftarrow \beta_i x_{i-1} - \alpha_i x_{i-2} + Gy$
$\delta z_i \leftarrow \delta z_i + \delta W_k$
$\delta W_k \leftarrow \delta W_k - \text{soft} \left( \frac{\psi_k}{\sqrt{q}} z_i; \tau W_k \right)$
$\delta x_i \leftarrow \tau^{-1} \delta z_i$
$z_i \leftarrow \beta_i x_{i-1} - \alpha_i x_{i-2} + Gy$
$\delta W_k \leftarrow \delta W_k - \text{soft} \left( \frac{\psi_k}{\sqrt{q}} z_i; \tau W_k \right)$

end for

$i \leftarrow i + 1$

until $i = T$

return $(\delta W_k)_{k=1}^{q}$

The learning procedure consists in alternating, over a fixed number of epochs, between estimating $\hat{x}_p$ for all the training samples and a fixed $W$ using Algorithm 2 and updating $W$ for a fixed set of estimates $\hat{x}_p$ using Algorithm 3. The update step consists in applying a gradient descent: $W_k \leftarrow W_k - \varepsilon \delta W_k$ with $\varepsilon$ the learning rate and $(\delta W_k)_{k=1}^{q}$ computed with the BPTT algorithm.

**Algorithm 3** LEFISTA Backpropagation algorithm

Require: $i = T - 1, (x_i)_{i=1}^{T}, (z_i)_{i=1}^{T}, (\alpha_i)_{i=1}^{T}, W, G, S, \Psi, q$

repeat

for $k = 1$ to $q$

$\delta h_k \leftarrow \frac{\psi_k}{\sqrt{q}} \text{soft} \left( \frac{\psi_k}{\sqrt{q}} z_i; \tau W_k \right)$

end for

if $i = T$

$\delta x_i \leftarrow S \delta z_i$

else

$\delta z_i \leftarrow (1 + \alpha_i) S \delta z_i - \alpha_i S \delta z_{i+1}$

end if

$i \leftarrow i - 1$

until $i = 0$

return $(\delta W_k)_{k=1}^{q}$

2.2. Learning the weight matrix

In order to learn the weight matrix, we consider a training set made of $P$ pairs $(y_p, x_p^*)_{p=1}^{P}$ where $x_p^*$ denotes the reference image and $y_p$ the corresponding measurement vector. Let us denote by $\hat{x}_p = f(y_p, W)$ the estimate of $x_p^*$ obtained by applying FISTA. The objective of the learning procedure consists in finding $W$ which

minimizes the following $\ell_2$-loss function:

$$
\mathcal{L} \left( (y_p, x_p^*)_{p=1}^{P}, W \right) = \frac{1}{P} \sum_{p=1}^{P} ||\hat{x}_p - x_p^*||^2.
$$

DNN-based approaches require to work on image patches, which usually exploit the block-compressed-sensing (BCS) framework, introduced by Gan [13] which permits to use CS techniques in a block-based manner.

The idea consists in training LEFISTA with a given measurement matrix $A \in \mathbb{R}^{M \times N_B}$ on patches of size $N_B \times N_B$ pixels. Once the network is trained, we divide each image of the test set into non-overlapping patches of size $N_B \times N_B$ pixels. We compressed each patch with the matrix $A$ used in the training phase and we reconstruct each patch independently by applying Algorithm 2 with $W$ learned in the training phase. The whole image is obtained by placing each reconstructed patch at its initial location in the canvas.

One remarkable property of such a technique is that it reduces the resolution of Problem (2) to the resolution of Problem (1), much faster to solve.
3. PERFORMANCE EVALUATION

3.1. Experimental settings

The proposed LEFISTA is tested with different numbers of layers $T = 30, 40, 50$. The patch size is set to $64 \times 64$ pixels and the measurement matrix $A \in \mathbb{R}^{M \times N}$ is chosen with Gaussian random entries, with the measurement rate $M/N$ varying between 0.1 and 0.5. In order to fit with the framework described by Carrillo et al., $W$ is composed of a concatenation of $S$ wavelet bases with Daubechies 1 to Daubechies 8 as mother function. Two decomposition levels are considered.

LEFISTA is implemented on TensorFlow and experiments have been performed on a NVIDIA Titan X GPU card. It is trained on 1600 patches randomly extracted from 23 images using mini-batch learning. The chosen optimization algorithm is Adam with a learning rate of 0.0005 and a batch size of 32. The number of epochs is fixed to 20.

The quality of the reconstruction is evaluated on a test set composed of 3 images, namely Barbara, Goldhill and Peppers. The Peak-Signal-to-Noise Ratio (PSNR) and the Structural-Similarity Index (SSIM) are used as image quality metrics.

3.2. Comparison to SARA

LEFISTA is firstly compared to a tiled version of SARA, where the algorithm is applied on the same non-overlapping patches. In this case, the matrix A used for SARA reconstruction is the same Gaussian random matrix as before, the reconstruction algorithm is Douglas-Rachford with 200 iterations and 10 updates of the weight matrix.

The results, expressed in terms of PSNR and SSIM, computed on Barbara image, and displayed in Figure 2a and Figure 2b, show that LEFISTA outperforms SARA for low measurement rates while SARA gets better PSNR when the measurement rate is above 0.4. One remarkable aspect of LEFISTA is that it takes less than a second to reconstruct an image while SARA takes around one hour, thus reducing the reconstruction time by a factor of several thousands. From Figure 2c and Figure 2d which show the images reconstructed with SARA and LEFISTA (50 layers) respectively, for a measurement rate of 0.5, one can notice that the visual quality is similar between the two methods. One may notice that the image reconstructed with LEFISTA is slightly more blurred than the one reconstructed with SARA. This can justify the difference observed in terms of PSNR.

3.3. Comparison to state-of-the-art block-compressed sensing algorithms

We also compare the proposed approach to state-of-the-art BCS reconstruction methods, namely multiscale BCS with smooth projected Landweber with the dual tree wavelet transform (BCS-SPL-DDWT) and the discrete wavelet transform (BCS-SPL-DWT) [14]. The code used for the two methods is provided by their authors.

The results, displayed on Table 1, show that LEFISTA is competitive against both BCS-SPL-DWT and BCS-SPL-DDWT. In terms of PSNR, LEFISTA is a bit lower while it has a far higher SSIM. One possible explanation is that BCS-SPL-DWT and BCS-SPL-DDWT use a Wiener filter at each iteration of the reconstruction algorithm to avoid blocking artifacts. This results in an over-smoothing of the image which has a significant impact on the SSIM.

One may notice that LEFISTA may achieve better image quality by increasing the number of layers, resulting in a higher number of parameters to be learned and in a deeper algorithm.

Regarding the computation time, BCS-SPL-DWT and BCS-SPL-DDWT take several seconds to reconstruct the image while the LEFISTA takes less than a second, resulting in an increase of one order of magnitude of the reconstruction time.

![Fig. 1: LEFISTA network architecture: (a) the 4-first layers of LEFISTA, with $\beta_i = (1 + \alpha_i)$ and (b) the non-linearity $f$ involved in LEFISTA, composed of a shrinkage operator in multiple wavelet bases.](image-url)
Fig. 2. Comparison between SARA and LEFISTA in terms of (a) PSNR and (b) SSIM with 30, 40 and 50 layers and for different measurement rates $M/N$ on the Barbara image. Image reconstructed for a measurement rate $M/N = 0.5$ (c) with SARA and (d) with LEFISTA (50 layers).

4. CONCLUSION

In this work, we propose to map the fast iterative soft thresholding algorithm to a deep neural network (DNN). We use this architecture to learn the weight matrix of a reweighted $\ell_1$-minimization problem, with a prior of sparsity in a concatenation of wavelet bases. This learned matrix is later used in the problem, which is recast as a classical $\ell_1$-minimization problem, much faster to solve than the classical reweighted $\ell_1$-minimization problem. The proposed approach leads to promising results in terms of image quality and opens the way to more advanced models, where the DNN architecture is not only used to learn the weight matrix but also to learn the non-linearities and the different matrices involved in the network.

5. REFERENCES


