Learning step sizes for unfolded sparse coding

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

Sparse coding is typically solved by iterative optimization techniques, such as the Iterative Shrinkage-Thresholding Algorithm (ISTA). Unfolding and learning weights of ISTA using neural networks is a practical way to accelerate estimation. In this paper, we study the selection of adapted step sizes for ISTA. We show that a simple step size strategy can improve the convergence rate of ISTA by leveraging the sparsity of the iterates. However, it is impractical in most large-scale applications. Therefore, we propose a network architecture where only the step sizes of ISTA are learned. We demonstrate that for a large class of unfolded algorithms, if the algorithm converges to the solution of the Lasso, its last layers correspond to ISTA with learned step sizes. Experiments show that our method is competitive with state-of-the-art networks when the solutions are sparse enough.

1 Introduction

The resolution of convex optimization problems by iterative algorithms has become a key part of machine learning and signal processing pipelines. Amongst these problems, special attention has been devoted to the Lasso (Tibshirani, 1996), due to the attractive sparsity properties of its solution (see Hastie et al. 2015 for an extensive review). For a given input $x \in \mathbb{R}^n$, a dictionary $D \in \mathbb{R}^{n \times m}$ and a regularization parameter $\lambda > 0$, the Lasso problem is

$$z^*(x) \in \operatorname*{arg\,min}_{z \in \mathbb{R}^m} F_x(z) \quad \text{with} \quad F_x(z) \triangleq \frac{1}{2}\|x - Dz\|^2 + \lambda \|z\|_1 \ . \tag{1}$$

A variety of algorithms exist to solve Problem (1), *e.g.* proximal coordinate descent (Tseng, 2001; Friedman et al., 2007), Least Angle Regression (Efron et al., 2004) or proximal splitting methods (Combettes and Bauschke, 2011). The focus of this paper is on the Iterative Shrinkage-Thresholding Algorithm (ISTA, Daubechies et al. 2004), which is a proximal-gradient method applied to Problem (1). ISTA starts from $z^{(0)} = 0$ and iterates

$$z^{(t+1)} = \mathrm{ST}\left(z^{(t)} - \frac{1}{L} D^\top (Dz^{(t)} - x), \frac{\lambda}{L}\right) \ , \tag{2}$$

where ST is the soft-thresholding operator defined as $\mathrm{ST}(x, u) \triangleq \mathrm{sign}(x) \max(|x| - u, 0)$, and L is the greatest eigenvalue of $D^\top D$. In the general case, ISTA converges at rate $1/t$, which can be improved to the *optimal* rate $1/t^2$ (Nesterov, 1983). However, this optimality stands in the worst possible case, and linear rates are achievable in practice (Liang et al., 2014).

A popular line of research to improve the speed of Lasso solvers is to try to identify the support of z^* , in order to diminish the size of the optimization problem (El Ghaoui et al., 2012; Ndiaye et al., 2017; Johnson and Guestrin, 2015; Massias et al., 2018). Once the support is identified, larger steps can also be taken, leading to improved rates for first order algorithms (Liang et al., 2014; Poon et al., 2018; Sun et al., 2019).

Submitted to 33rd Conference on Neural Information Processing Systems (NeurIPS 2019). Do not distribute.

However, these techniques only consider the case where a single Lasso problem is solved. When one wants to solve the Lasso for many samples $\{x^i\}_{i=1}^N$ – *e.g.* in dictionary learning (Olshausen and Field, 1997) – it is proposed by Gregor and Le Cun (2010) to *learn* a T-layers neural network of parameters Θ , $\Phi_\Theta : \mathbb{R}^n \to \mathbb{R}^m$ such that $\Phi_\Theta(x) \simeq z^*(x)$. This Learned-ISTA (LISTA) algorithm yields better solution estimates than ISTA on new samples for the same number of iterations/layers. This idea has led to a profusion of literature (summarized in Table A.1 in appendix). Recently, it has been hinted by Zhang and Ghanem (2018); Ito et al. (2018); Liu et al. (2019) that only a few well-chosen parameters can be learned while retaining the performances of LISTA.

In this article, we study strategies for LISTA where only step sizes are learned. In Section 3, we propose Oracle-ISTA, an analytic strategy to obtain larger step sizes in ISTA. We show that the proposed algorithm's convergence rate can be much better than that of ISTA. However, it requires computing a large number of Lipschitz constants which is a burden in high dimension. This motivates the introduction of Step-LISTA (SLISTA) networks in Section 4, where only a step size parameter is learned per layer. As a theoretical justification, we show in Theorem 4.4 that the last layers of *any* deep LISTA network converging on the Lasso *must* correspond to ISTA iterations with learned step sizes. We validate the soundness of this approach with numerical experiments in Section 5.

2 Notation and Framework

Notation The ℓ_2 norm on $\mathbb{R}^n$ is $\|\cdot\|$. For $p \in [1, \infty]$, $\|\cdot\|_p$ is the ℓ_p norm. The Frobenius matrix norm is $\|M\|_F$. The identity matrix of size m is Id_m . ST is the soft-thresholding operator. Iterations are denoted $z^{(t)}$. $\lambda > 0$ is the regularization parameter. The Lasso cost function is F_x . $\psi_\alpha(z, x)$ is one iteration of ISTA with step α: $\psi_\alpha(z, x) = \mathrm{ST}(z - \alpha D^\top(Dz - x), \alpha\lambda)$. $\phi_\theta(z, x)$ is one iteration of a LISTA layer with parameters $\theta = (W, \alpha, \beta)$: $\phi_\theta(z, x) = \mathrm{ST}(z - \alpha W^\top(Dz - x), \beta\lambda)$.

The set of integers between 1 and m is $[\![1, m]\!]$. Given $z \in \mathbb{R}^m$, the support is $\mathrm{supp}(z) = \{j \in [\![1, m]\!] : z_j \neq 0\} \subset [\![1, m]\!]$. For $S \subset [\![0, m]\!]$, $D_S \in \mathbb{R}^{n\times m}$ is the matrix containing the columns of D indexed by S. We denote L_S, the greatest eigenvalue of $D_S^\top D_S$. The equicorrelation set is $E = \{j \in [\![1, m]\!] : |D_j^\top(Dz^* - x)| = \lambda\}$. The equiregularization set is $\mathcal{B}_\infty = \{x \in \mathbb{R}^n : \|D^\top x\|_\infty = 1\}$. Neural networks parameters are between brackets, e.g. $\Theta = \{\alpha^{(t)}, \beta^{(t)}\}_{t=0}^{T-1}$. The sign function is $\mathrm{sign}(x) = 1$ if $x > 0$, -1 if $x < 0$ and 0 is $x = 0$.

Framework This paragraph recalls some properties of the Lasso. Lemma 2.1 gives the first-order optimality conditions for the Lasso.

Lemma 2.1 (Optimality for the Lasso). *The Karush-Kuhn-Tucker (KKT) conditions read*

$$z^* \in \arg\min F_x \Leftrightarrow \forall j \in [\![1, m]\!], D_j^\top(x - Dz^*) \in \lambda\partial|z_j^*| = \begin{cases} \{\lambda \,\mathrm{sign}\, z_j^*\}, & \text{if } z_j^* \neq 0 \ , \\ [-\lambda, \lambda], & \text{if } z_j^* = 0 \ . \end{cases} \tag{3}$$

Defining $\lambda_{\max} \triangleq \|D^\top x\|_\infty$, it holds $\arg\min F_x = \{0\} \Leftrightarrow \lambda \geq \lambda_{\max}$. For *some* results in Section 3, we will need the following assumption on the dictionary D:

Assumption 2.2 (Uniqueness assumption). *D is such that the solution of Problem* (1) *is unique for all λ and x i.e.* $\arg\min F_x = \{z^*\}$.

Assumption 2.2 may seem stringent since whenever $m > n$, F_x is not strictly convex. However, it was shown in Tibshirani (2013, Lemma 4) – with earlier results from Rosset et al. 2004 – that if D is sampled from a continuous distribution, Assumption 2.2 holds for D with probability one.

Definition 2.3 (Equicorrelation set). *The KKT conditions motivate the introduction of the* equicorrelation set $E \triangleq \{j \in [\![1, m]\!] : |D_j^\top(Dz^* - x)| = \lambda\}$ *, since $j \notin E \implies z_j^* = 0$* , i.e. *$E$ contains the support of any solution z^** .

When Assumption 2.2 holds, we have $E = \mathrm{supp}(z^*)$ (Tibshirani, 2013, Lemma 16).

We consider samples x in the *equiregularization* set

$$\mathcal{B}_\infty \triangleq \{x \in \mathbb{R}^n : \|D^\top x\|_\infty = 1\} \ , \tag{4}$$

which is the set of x such that $\lambda_{\max}(x) = 1$. Therefore, when $\lambda \geq 1$, the solution is $z^*(x) = 0$ for all $x \in \mathcal{B}_\infty$, and when $\lambda < 1$, $z^*(x) \neq 0$ for all $x \in \mathcal{B}_\infty$. For this reason, we assume $0 < \lambda < 1$ in the following.

3 Better step sizes for ISTA

The Lasso objective is the sum of a L-smooth function, $\frac{1}{2}\|x - D\cdot\|^2$, and a function with an explicit proximal operator, $\lambda\|\cdot\|_1$. Proximal gradient descent for this problem, with the sequence of step sizes $(\alpha^{(t)})$ consists in iterating

$$z^{(t+1)} = \mathrm{ST}\left(z^{(t)} - \alpha^{(t)} D^\top (D z^{(t)} - x), \lambda\alpha^{(t)}\right) \ . \tag{5}$$

ISTA follows these iterations with a constant step size $\alpha^{(t)} = 1/L$. In the following, denote $\psi_\alpha(z, x) \triangleq \mathrm{ST}(z - \alpha D^\top (D z^{(t)} - x), \alpha\lambda)$. One iteration of ISTA can be cast as a majorization-minimization step (Beck and Teboulle, 2009). Indeed, for all $z \in \mathbb{R}^m$,

$$\begin{aligned} F_x(z) &= \tfrac{1}{2}\|x - Dz^{(t)}\|^2 + (z - z^{(t)})^\top D^\top (Dz^{(t)} - x) + \tfrac{1}{2}\|D(z - z^{(t)})\|^2 + \lambda\|z\|_1 & (6)\\ &\leq \underbrace{\tfrac{1}{2}\|x - Dz^{(t)}\|^2 + (z - z^{(t)})^\top D^\top (Dz^{(t)} - x) + \tfrac{L}{2}\|z - z^{(t)}\|^2 + \lambda\|z\|_1}_{\triangleq Q_{x,L}(z, z^{(t)})} \ , & (7)\end{aligned}$$

where we have used the inequality $(z - z^{(t)})^\top D^\top D (z - z^{(t)}) \leq L\|z - z^{(t)}\|^2$. The minimizer of $Q_{x,L}(\cdot, z^{(t)})$ is $\psi_{1/L}(z^{(t)}, x)$, which is the next ISTA step.

Oracle-ISTA: an accelerated ISTA with larger step sizes Since the iterates are sparse, this approach can be refined. For $S \subset [\![1, m]\!]$, let us define the S-smoothness of D as

$$L_S \triangleq \max_z z^\top D^\top D z, \quad \text{s.t. } \|z\| = 1 \text{ and } \mathrm{supp}(z) \subset S \ , \tag{8}$$

with the convention $L_\emptyset = L$. Note that L_S is the greatest eigenvalue of $D_S^\top D_S$ where $D_S \in \mathbb{R}^{n \times |S|}$ is the columns of D indexed by S . For all S , $L_S \leq L$, since L is the solution of Equation (8) without support constraint. Assume $\mathrm{supp}(z^{(t)}) \subset S$. Combining Equations (6) and (8), we have

$$\forall z \ \text{ s.t. } \ \mathrm{supp}(z) \subset S, \ \ F_x(z) \leq Q_{x,L_S}(z, z^{(t)}) \ . \tag{9}$$

The minimizer of the r.h.s is $z = \psi_{1/L_S}(z^{(t)}, x)$. Furthermore, the r.h.s. is a tighter upper bound than the one given in Equation (7) (see illustration in Figure 1). Therefore, using $z^{(t+1)} = \psi_{1/L_S}(z^{(t)}, x)$ minimizes a tighter upper bound, provided that the following condition holds

$$\mathrm{supp}(z^{(t+1)}) \subset S \ . \tag{$\star$}$$

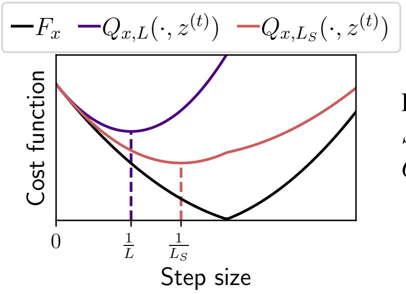


Figure 1: Majorization illustration. If $z^{(t)}$ has support S , $Q_{x,L_S}(\cdot, z^{(t)})$ is a tighter upper bound of F_x than $Q_{x,L}(\cdot, z^{(t)})$ on the set of points of support S .

Oracle-ISTA (OISTA) is an accelerated version of ISTA which leverages the sparsity of the iterates in order to use larger step sizes. The method is summarized in Algorithm 1. OISTA computes $y^{(t+1)} = \psi_{1/L_s}(z^{(t)}, x)$, using the larger step size $1/L_S$, and checks if it satisfies the support Condition $\star$. When the condition is satisfied, the step can be safely accepted. In particular Equation (9) yields $F_x(y^{(t+1)}) \leq F_x(z^{(t)})$. Otherwise, the algorithm falls back to the regular ISTA iteration with the smaller step size. Hence, each iteration of the algorithm is guaranteed to decrease F_x . The following proposition shows that OISTA converges in iterates, achieves finite support identification, and eventually reaches a safe regime where Condition $\star$ is always true.

Algorithm 1: Oracle-ISTA (OISTA) with larger step sizes

```
Input: Dictionary $D$ , target $x$ , number of iterations $T$
$z^{(0)} = 0$
for $t = 0, \dots, T-1$ do
    Compute $S = \mathrm{supp}(z^{(t)})$ and $L_S$ using an oracle ;
    Set $y^{(t+1)} = \psi_{1/L_S}(z^{(t)}, x)$ ;
    if Condition ⋆ : $\mathrm{supp}(y^{(t+1)}) \subset S$ then Set $z^{(t+1)} = y^{(t+1)}$ ;
    else Set $z^{(t+1)} = \psi_{1/L}(z^{(t)}, x)$ ;
Output: Sparse code $z^{(T)}$
```

Proposition 3.1 (Convergence, finite-time support identification and safe regime)**.** *When Assumption 2.2 holds, the sequence* $(z^{(t)})$ *generated by the algorithm converges to* $z^* = \arg\min F_x$.

Further, there exists an iteration T^* *such that for* $t \geq T^*$, $\mathrm{supp}(z^{(t)}) = \mathrm{supp}(z^*) \triangleq S^*$ *and Condition* $\star$ *is always statisfied.*

Sketch of proof (full proof in Subsection B.1). Using Zangwill's global convergence theorem (Zangwill, 1969), we show that all accumulation points of $(z^{(t)})$ are solutions of Lasso. Since the solution is assumed unique, $(z^{(t)})$ converges to z^* . Then, we show that the algorithm achieves finite-support identification with a technique inspired by Hale et al. (2008). The algorithm gets arbitrary close to z^* , eventually with the same support. We finally show that in a neighborhood of z^* , the set of points of support S^* is stable by $\psi_{1/L_S}(\cdot, x)$. The algorithm eventually reaches this region, and then Condition $\star$ is true. ☐

It follows that the algorithm enjoys the usual ISTA convergence results replacing L with L_{S^*} .

Proposition 3.2 (Rates of convergence)**.** *For* $t > T^*$, $F_x(z^{(t)}) - F_x(z^*) \leq L_{S^*} \frac{\|z^* - z^{(T^*)}\|^2}{2(t - T^*)}$.
If additionally $\inf_{\|z\|=1} \|D_{S^*} z\|^2 = \mu^* > 0$, *then the convergence rate for* $t \geq T^*$ *is*
$F_x(z^{(t)}) - F_x(z^*) \leq (1 - \frac{\mu^*}{L_{S^*}})^{t - T^*} (F_x(z^{(T^*)}) - F_x(z^*))$.

Sketch of proof (full proof in Subsection B.2). After iteration T^* , OISTA is equivalent to ISTA applied on $F_x(z)$ restricted to $z \in S^*$. This function is L_{S^*}-smooth, and μ^*-strongly convex if $\mu^* > 0$. Therefore, the classical ISTA rates apply with improved condition number. ☐

These two rates are tighter than the usual ISTA rates – in the convex case $L\frac{\|z^*\|^2}{2t}$ and in the μ-strongly convex case $(1 - \frac{\mu^*}{L})^t (F_x(0) - F_x(z^*))$ (Beck and Teboulle, 2009). Finally, the same way ISTA converges in one iteration when D is orthogonal ($D^\top D = \mathrm{Id}_m$), OISTA converges in one iteration if S^* is identified and D_{S^*} is orthogonal.

Proposition 3.3. *Assume* $D_{S^*}^\top D_{S^*} = L_{S^*} \mathrm{Id}_{|S^*|}$. *Then,* $z^{(T^*+1)} = z^*$.

Proof. For z *s.t.* $\mathrm{supp}(z) = S^*$, $F_x(z) = Q_{x, L_S}(z, z^{(T^*)})$. Hence, the OISTA step minimizes F_x . ☐

Quantification of the rates improvement in a Gaussian setting The following proposition gives an asymptotic value for $\frac{L_S}{L}$ in a simple setting.

Proposition 3.4. *Assume that the entries of* $D \in \mathbb{R}^{n \times m}$ *are i.i.d centered Gaussian variables with variance* 1 . *Assume that* S *consists of* k *integers chosen uniformly at random in* $[\![1, m]\!]$. *Assume that* $k, m, n \to +\infty$ *with linear ratios* $m/n \to \gamma, \quad k/m \to \zeta$. *Then*

$$\frac{L_S}{L} \to \left(\frac{1 + \sqrt{\zeta\gamma}}{1 + \sqrt{\gamma}} \right)^2 \quad . \tag{10}$$

This is a direct application of the Marchenko-Pastur law (Marchenko and Pastur, 1967). The law is illustrated on a toy dataset in Figure D.1. In Proposition 3.4, γ is the ratio between the number of atoms and number of dimensions, and the average size of S is described by $\zeta \leq 1$. In an overcomplete setting, we have $\gamma \gg 1$, yielding the approximation of Equation (10): $L_S \simeq \zeta L$. Therefore, if z^* is very sparse ($\zeta \ll 1$), the convergence rates of Proposition 3.2 are much better than those of ISTA.

Example Figure 2 compares the OISTA, ISTA, and FISTA on a toy problem. The improved rate of convergence of OISTA is illustrated. Further comparisons are displayed in Figure D.2 for different regularization parameters λ . While this demonstrates a much faster rate of convergence, it requires computing several Lipschitz constants L_S , which is cumbersome in high dimension. This motivates the next section, where we propose to *learn* those steps.

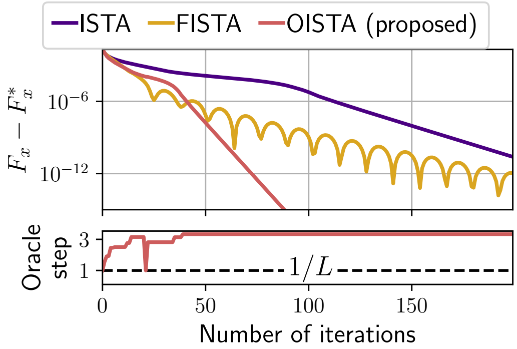
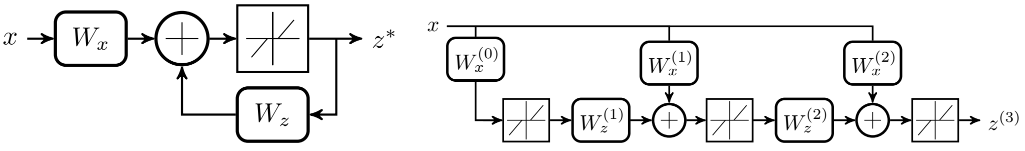


Figure 2: Convergence curves of OISTA, ISTA, and FISTA on a toy problem with $n = 10$, $m = 50$, $\lambda = 0.5$. The bottom figure displays the (normalized) steps taken by OISTA at each iteration. Full experimental setup described in Appendix D.

4 Learning unfolded algorithms

Network architectures At each step, ISTA performs a linear operation to compute an update in the direction of the gradient $D^\top(Dz^{(t)} - x)$ and then an element-wise non linearity with the soft-thresholding operator ST . The whole algorithm can be summarized as a recurrent neural network (RNN), presented in Figure 3a. Gregor and Le Cun (2010) introduced Learned-ISTA (LISTA), a neural network constructed by unfolding this RNN T times and learning the weights associated to each layer. The unfolded network, presented in Figure 3b, iterates $z^{(t+1)} = \mathrm{ST}(W_x^{(t)}x + W_z^{(t)}z^{(t)}, \lambda\beta^{(t)})$. It outputs exactly the same vector as T iterations of ISTA when $W_x^{(t)} = \frac{D^\top}{L}$, $W_z^{(t)} = \mathrm{Id}_m - \frac{D^\top D}{L}$ and $\beta^{(t)} = \frac{1}{L}$. Empirically, this network is able to output a better estimate of the sparse code solution with fewer operations.



(a) **ISTA** - Recurrent Neural Network

(b) **LISTA** - Unfolded network with $T = 3$

Figure 3: Network architecture for ISTA (*left*) and LISTA (*right*).

Due to the expression of the gradient, Chen et al. (2018) proposed to consider only a subclass of the previous networks, where the weights W_x and W_z are coupled via $W_z = \mathrm{Id}_m - W_x^\top D$. This is the architecture we consider in the following. A layer of LISTA is a function $\phi_\theta : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^m$ parametrized by $\theta = (W, \alpha, \beta) \in \mathbb{R}^{n\times m} \times \mathbb{R}_*^+ \times \mathbb{R}_*^+$ such that

$$\phi_\theta(z, x) = \mathrm{ST}(z - \alpha W^\top(Dz - x), \beta\lambda) \ . \tag{11}$$

Given a set of T layer parameters $\Theta^{(T)} = \{\theta^{(t)}\}_{t=0}^{T-1}$, the LISTA network $\Phi_{\Theta^{(T)}} : \mathbb{R}^n \to \mathbb{R}^m$ is $\Phi_{\Theta^{(T)}}(x) = z^{(T)}(x)$ where $z^{(t)}(x)$ is defined by recursion

$$z^{(0)}(x) = 0, \quad \text{and} \quad z^{(t+1)}(x) = \phi_{\theta^{(t)}}(z^{(t)}(x), x) \quad \text{for } t \in [\![0, T-1]\!] \ . \tag{12}$$

Taking $W = D$, $\alpha = \beta = \frac{1}{L}$ yields the same outputs as T iterations of ISTA.

To alleviate the need to learn the large matrices $W^{(t)}$, Liu et al. (2019) proposed to use a shared analytic matrix W_{ALISTA} for all layers. The matrix is computed in a preprocessing stage by

$$W_{\text{ALISTA}} = \arg\min_{W} \|W^\top D\|_F^2 \quad s.t. \quad \operatorname{diag}(W^\top D) = \mathbf{1}_m \ . \tag{13}$$

Then, only the parameters $(\alpha^{(t)}, \beta^{(t)})$ are learned. This effectively reduces the number of parameters from $(nm + 2) \times T$ to $2 \times T$. However, we will see that ALISTA fails in our setup.

Step-LISTA With regards to the study on step sizes for ISTA in Section 3, we propose to *learn* approximation of ISTA step sizes for the input distribution using the LISTA framework. The resulting network, dubbed Step-LISTA (SLISTA), has T parameters $\Theta_{\text{SLISTA}} = \{\alpha^{(t)}\}_{t=0}^{T-1}$, and follows the iterations:

$$z^{(t+1)}(x) = \operatorname{ST}(z^{(t)}(x) - \alpha^{(t)} D^\top (Dz^{(t)}(x) - x), \alpha^{(t)}\lambda) \ . \tag{14}$$

This is equivalent to a coupling in the LISTA parameters: a LISTA layer $\theta = (W, \alpha, \beta)$ corresponds to a SLISTA layer if and only if $\frac{\alpha}{\beta} W = D$. This network aims at learning good step sizes, like the ones used in OISTA, without the computational burden of computing Lipschitz constants. The number of parameters compared to the classical LISTA architecture Θ_{LISTA} is greatly diminished, making the network easier to train. Learning curves are shown in Figure **??** in appendix. Figure 4 displays the learned steps of a SLISTA network on a toy example. The network learns larger step-sizes as the $1/L_S$'s increase.

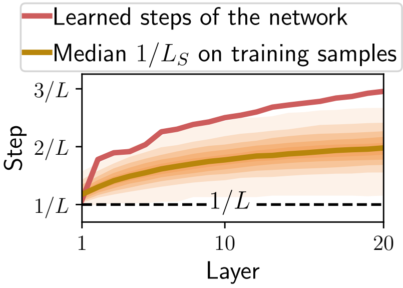


Figure 4: Steps learned with a 20 layers SLISTA network on a 10×20 problem. For each layer t and each training sample x, we compute the support $S(x, t)$ of $z^{(t)}(x)$. The brown curves display the quantiles of the distribution of $1/L_{S(x,t)}$ for each layer t . Full experimental setup described in Appendix D.

Training the network We consider the framework where the network learns to solve the Lasso on $\mathcal{B}_\infty$ in an *unsupervised* way. Given a distribution p on $\mathcal{B}_\infty$, the network is trained by solving

$$\tilde{\Theta}^{(T)} \in \arg\min_{\Theta^{(T)}} \mathcal{L}(\Theta^{(T)}) \triangleq \mathbb{E}_{x\sim p}[F_x(\Phi_{\Theta^{(T)}}(x))] \ . \tag{15}$$

Most of the literature on learned optimization train the network with a different *supervised* objective (Gregor and Le Cun, 2010; Xin et al., 2016; Chen et al., 2018; Liu et al., 2019). Given a set of pairs (x^i, z^i) , the supervised approach tries to learn the parameters of the network such that $\Phi_\Theta(x^i) \simeq z^i$ *e.g.* by minimizing $\|\Phi_\Theta(x^i) - z^i\|^2$. This training procedure differs critically from ours. For instance, ISTA does not converge for the supervised problem in general while it does for the unsupervised one. As Proposition 4.1 shows, the unsupervised approach allows to *learn to minimize* the Lasso cost function F_x .

Proposition 4.1 (Pointwise convergence)**.** *Let* $\tilde{\Theta}^{(T)}$ *found by solving Problem* (15).
For $x \in \mathcal{B}_\infty$ *such that* $p(x) > 0$, $F_x(\Phi_{\tilde{\Theta}^{(T)}}(x)) \xrightarrow[T\to+\infty]{} F_x^*$ *almost everywhere.*

Proof. Let $\Theta^{(T)}_{\text{ISTA}}$ the parameters corresponding to ISTA *i.e.* $\theta^{(t)}_{\text{ISTA}} = (D, 1/L, 1/L)$. For all T , we have $\mathbb{E}_{x\sim p}[F_x^*] \leq \mathbb{E}_{x\sim p}[F_x(\Phi_{\tilde{\Theta}^{(T)}}(x))] \leq \mathbb{E}_{x\sim p}[F_x(\Phi_{\Theta^{(T)}_{\text{ISTA}}}(x))]$. Since ISTA converges uniformly on any compact, the right hand term goes to $\mathbb{E}_{x\sim p}[F_x^*]$. Therefore, by the squeeze theorem, $\mathbb{E}_{x\sim p}[F_x(\Phi_{\tilde{\Theta}^{(T)}}(x)) - F_x^*] \to 0$. This implies almost sure convergence of $F_x(\Phi_{\tilde{\Theta}^{(T)}}(x)) - F_x^*$ to 0 since it is non-negative. □

Asymptotical weight coupling theorem In this paragraph, we show the main result of this paper: any LISTA network minimizing F_x on $\mathcal{B}_\infty$ reduces to SLISTA in its deep layers (Theorem 4.4). It relies on the following Lemmas.

Lemma 4.2 (Stability of solutions around D_j)**.** *Let $D \in \mathbb{R}^{n\times m}$ be a dictionary with non-duplicated unit-normed columns. Let $c \triangleq \max_{l\neq j} |D_l^\top D_j| < 1$. Then for all $j \in [\![1, m]\!]$ and $\varepsilon \in \mathbb{R}^m$ such that $\|\varepsilon\| < \lambda(1-c)$ and $D_j^\top \varepsilon = 0$, the vector $(1-\lambda)e_j$ minimizes F_x for $x = D_j + \varepsilon$.*

It can be proven by verifying the KKT conditions (3) for $(1-\lambda)e_j$, detailed in Subsection C.1.

Lemma 4.3 (Weight coupling)**.** *Let $D \in \mathbb{R}^{n\times m}$ be a dictionary with non-duplicated unit-normed columns. Let $\theta = (W, \alpha, \beta)$ a set of parameters. Assume that all the couples $(z^*(x), x) \in \mathbb{R}^m \times \mathcal{B}_\infty$ such that $z^*(x) \in \arg\min F_x(z)$ verify $\phi_\theta(z^*(x), x) = z^*(x)$. Then, $\frac{\alpha}{\beta}W = D$.*

Sketch of proof (full proof in Subsection C.2). For $j \in [\![1, m]\!]$, consider $x = D_j + \varepsilon$, with $\varepsilon^\top D_j = 0$. For $\|\varepsilon\|$ small enough, $x \in \mathcal{B}_\infty$ and ε verifies the hypothesis of Lemma 4.2, therefore $z^* = (1-\lambda)e_j \in \arg\min F_x$. Writing $\phi_\theta(z^*, x) = z^*$ for the j-th coordinate yields $\alpha W_j^\top(\lambda D_j + \varepsilon) = \lambda\beta$. We can then verify that $(\alpha W_j^\top - \beta D_j^\top)(\lambda D_j + \varepsilon) = 0$. This stands for any ε orthogonal to D_j and of norm small enough. Simple linear algebra shows that this implies $\alpha W_j - \beta D_j = 0$. ☐

Lemma 4.3 states that the Lasso solutions are fixed points of a LISTA layer only if this layer corresponds to a step size for ISTA. The following theorem extends the lemma by continuity, and shows that the deep layers of any converging LISTA network must tend toward a SLISTA layer.

Theorem 4.4. *Let $D \in \mathbb{R}^{n\times m}$ be a dictionary with non-duplicated unit-normed columns. Let $\Theta^{(T)} = \{\theta^{(t)}\}_{t=0}^T$ be the parameters of a sequence of LISTA networks such that the transfer function of the layer t is $z^{(t+1)} = \phi_{\theta^{(t)}}(z^{(t)}, x)$. Assume that*

(i) the sequence of parameters converges i.e. $\theta^{(t)} \xrightarrow[t\to\infty]{} \theta^* = (W^*, \alpha^*, \beta^*)$,

(ii) the output of the network converges toward a solution $z^*(x)$ of the Lasso (1) uniformly over the equiregularization set $\mathcal{B}_\infty$, i.e. $\sup_{x\in\mathcal{B}_\infty} \|\Phi_{\Theta^{(T)}}(x) - z^*(x)\| \xrightarrow[T\to\infty]{} 0$.

Then $\frac{\alpha^*}{\beta^*}W^* = D$.

Sketch of proof (full proof in Subsection C.3). Let $\varepsilon > 0$, and $x \in \mathcal{B}_\infty$. Using the triangular inequality, we have

$$\|\phi_{\theta^*}(z^*, x) - z^*\| \quad \leq \quad \|\phi_{\theta^*}(z^*, x) - \phi_{\theta^{(t)}}(z^{(t)}, x)\| + \|\phi_{\theta^{(t)}}(z^{(t)}, x) - z^*\| \tag{16}$$

Since the $z^{(t)}$ and $\theta^{(t)}$ converge, they are valued over a compact set K. The function $f : (z, x, \theta) \mapsto \phi_\theta(z, x)$ is continuous, piecewise-linear. It is therefore Lipschitz on K. Hence, we have $\|\phi_{\theta^*}(z^*, x) - \phi_{\theta^{(t)}}(z^{(t)}, x)\| \leq \varepsilon$ for t large enough. Since $\phi_{\theta^{(t)}}(z^{(t)}, x) = z^{(t+1)}$ and $z^{(t)} \to z^*$, $\|\phi_{\theta^{(t)}}(z^{(t)}, x) - z^*\| \leq \varepsilon$ for t large enough. Finally, $\phi_{\theta^*}(z^*, x) = z^*$. Lemma 4.3 allows to conclude. ☐

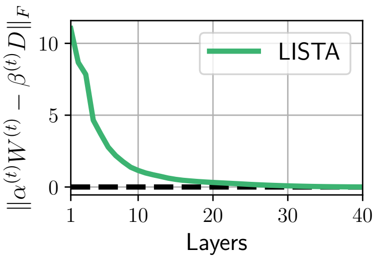


Figure 5: Illustration of Theorem 4.4: for deep layers of LISTA, we have $\|\alpha^{(t)}W^{(t)} - \beta^{(t)}D\|_F \to 0$, indicating that the network ultimately only learns a step size. Full experimental setup described in Appendix D.

Theorem 4.4 means that the deep layers of any LISTA network that converges to solutions of the Lasso correspond to SLISTA iterations: $W^{(t)}$ aligns with D , and $\alpha^{(t)}, \beta^{(t)}$ get coupled. This is illustrated in Figure 5, where a 40-layers LISTA network is trained on a 10×20 problem with

$\lambda = 0.1$. As predicted by the theorem, $\frac{\alpha^{(t)}}{\beta^{(t)}} W^{(t)} \to D$. The last layers only learn a step size. This is consistent with the observation of Moreau and Bruna (2017) which shows that the deep layers of LISTA stay close to ISTA. Further, Theorem 4.4 also shows that it is hopeless to optimize the unsupervised objective (15) with W_{ALISTA} (13), since this matrix is not aligned with D .

5 Numerical Experiments

This section provides numerical arguments to compare SLISTA to LISTA and ISTA. All the experiments were run using `Python` (Python Software Foundation, 2017) and `pytorch` (Paszke et al., 2017). The code to reproduce the figures is available online[1].

Network comparisons We compare the proposed approach SLISTA to state-of-the-art learned methods LISTA (Chen et al., 2018) and ALISTA (Liu et al., 2019) on synthetic and semi-real cases.

In the synthetic case, a dictionary $D \in \mathbb{R}^{n \times m}$ of Gaussian i.i.d. entries is generated. Each column is then normalized to one. A set of Gaussian i.i.d. samples $(\tilde{x}^i)_{i=1}^N \in \mathbb{R}^n$ is drawn. The input samples are obtained as $x^i = \tilde{x}^i / \|D^\top \tilde{x}^i\|_\infty \in \mathcal{B}_\infty$, so that for all i , $x^i \in \mathcal{B}_\infty$. We set $m = 256$ and $n = 64$.

For the semi-real case, we used the digits dataset from `scikit-learn` (Pedregosa et al., 2011) which consists of 8×8 images of handwritten digits from 0 to 9 . We sample $m = 256$ samples at random from this dataset and normalize it do generate our dictionary D . Compared to the simulated Gaussian dictionary, this dictionary has a much richer correlation structure, which is known to imper the performances of learned algorithms (Moreau and Bruna, 2017). The input distribution is generated as in the simulated case.

The networks are trained by minimizing the empirical loss $\mathcal{L}$ (15) on a training set of size $N_{\text{train}} = 10,000$ and we report the loss on a test set of size $N_{\text{test}} = 10,000$. Further details on training are in Appendix D.

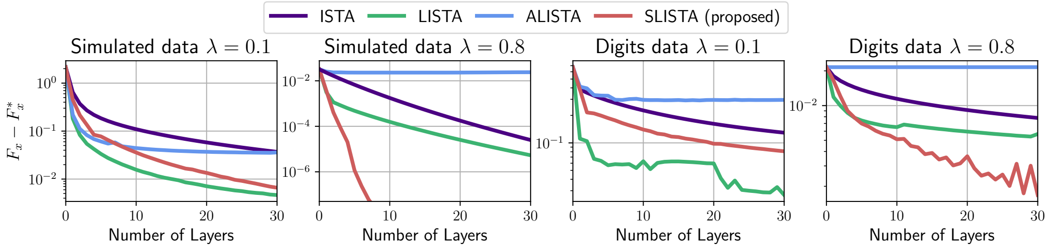


Figure 6: Test loss of ISTA, ALISTA, LISTA and SLISTA on simulated and semi-real data for different regularization parameters.

Figure 6 shows the test curves for different levels of regularization $\lambda = 0.1$ and 0.8. SLISTA performs best for high λ, even for challenging semi-real dictionary D . In a low regularization setting, LISTA performs best as SLISTA cannot learn larger steps due to the low sparsity of the solution. In this unsupervised setting, ALISTA does not converge in accordance with Theorem 4.4.

6 Conclusion

We showed that using larger step sizes is an efficient strategy to accelerate ISTA for sparse solution of the Lasso. In order to make this approach practical, we proposed SLISTA, a neural network architecture which learns such step sizes. Theorem 4.4 shows that the deepest layers of any converging LISTA architecture must converge to a SLISTA layer. Numerical experiments show that SLISTA outperforms LISTA in a high sparsity setting. An major benefit of our approach is that it preserves the dictionary. We plan on leveraging this property to apply SLISTA in convolutional or wavelet cases, where the structure of the dictionary allows for fast multiplications.

---

[1] The code can be found in supplementary materials.

References

Jonas Adler, Axel Ringh, Ozan Öktem, and Johan Karlsson. Learning to solve inverse problems using Wasserstein loss. *preprint ArXiv*, 1710.10898, 2017.

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

Mark Borgerding, Philip Schniter, and Sundeep Rangan. AMP-inspired deep networks for sparse linear inverse problems. *IEEE Transactions on Signal Processing*, 65(16):4293–4308, 2017.

Xiaohan Chen, Jialin Liu, Zhangyang Wang, and Wotao Yin. Theoretical linear convergence of unfolded ISTA and its practical weights and thresholds. In *Advances in Neural Information Processing Systems (NIPS)*, pages 9061–9071, 2018.

Patrick L Combettes and Heinz H. Bauschke. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer, 2011. ISBN 9788578110796. doi: 10.1017/CBO9781107415324.004.

Ingrid Daubechies, Michel Defrise, and Christine De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57(11):1413–1457, 2004.

Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *Ann. Statist.*, 32(2):407–499, 2004.

Laurent El Ghaoui, Vivian Viallon, and Tarek Rabbani. Safe feature elimination in sparse supervised learning. *J. Pacific Optim.*, 8(4):667–698, 2012.

Jerome Friedman, Trevor Hastie, Holger Höfling, and Robert Tibshirani. Pathwise coordinate optimization. *The Annals of Applied Statistics*, 1(2):302–332, 2007.

Raja Giryes, Yonina C. Eldar, Alex M. Bronstein, and Guillermo Sapiro. Tradeoffs between convergence speed and reconstruction accuracy in inverse problems. *IEEE Transaction on Signal Processing*, 66(7):1676–1690, 2018.

Karol Gregor and Yann Le Cun. Learning Fast Approximations of Sparse Coding. In *International Conference on Machine Learning (ICML)*, pages 399–406, 2010.

Elaine Hale, Wotao Yin, and Yin Zhang. Fixed-point continuation for ℓ_1-minimization: Methodology and convergence. *SIAM J. Optim.*, 19(3):1107–1130, 2008.

Trevor Hastie, Robert Tibshirani, and Martin Wainwright. *Statistical Learning with Sparsity: The Lasso and Generalizations*. CRC Press, 2015.

John R. Hershey, Jonathan Le Roux, and Felix Weninger. Deep unfolding: Model-based inspiration of novel deep architectures. *preprint ArXiv*, 1409.2574, 2014.

Daisuke Ito, Satoshi Takabe, and Tadashi Wadayama. Trainable ISTA for sparse signal recovery. In *IEEE International Conference on Communications Workshops*, pages 1–6, 2018.

Tyler Johnson and Carlos Guestrin. Blitz: A principled meta-algorithm for scaling sparse optimization. In *International Conference on Machine Learning (ICML)*, pages 1171–1179, 2015.

Jingwei Liang, Jalal Fadili, and Gabriel Peyré. Local linear convergence of forward–backward under partial smoothness. In *Advances in Neural Information Processing Systems*, pages 1970–1978, 2014.

Jialin Liu, Xiaohan Chen, Zhangyang Wang, and Wotao Yin. ALISTA: Analytic weights are as good as learned weigths in LISTA. In *International Conference on Learning Representation (ICLR)*, 2019.

Vladimir A Marchenko and Leonid Andreevich Pastur. Distribution of eigenvalues for some sets of random matrices. *Mathematics of the USSR-Sbornik*, 1(4):457, 1967.

Mathurin Massias, Alexandre Gramfort, and Joseph Salmon. Celer: a Fast Solver for the Lasso with Dual Extrapolation. In *International Conference on Machine Learning (ICML)*, 2018.

Thomas Moreau and Joan Bruna. Understanding neural sparse coding with matrix factorization. In *International Conference on Learning Representation (ICLR)*, 2017.

Eugene Ndiaye, Olivier Fercoq, Alexandre Gramfort, and Joseph Salmon. Gap safe screening rules for sparsity enforcing penalties. *J. Mach. Learn. Res.*, 18(128):1–33, 2017.

Yurii Nesterov. A method for solving a convex programming problem with rate of convergence $O(1/k^2)$. *Soviet Math. Doklady*, 269(3):543–547, 1983.

Bruno A. Olshausen and David J Field. Sparse coding with an incomplete basis set: a strategy employed by V1, 1997.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*, 2017.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Clarice Poon, Jingwei Liang, and Carola-Bibiane Schönlieb. Local convergence properties of SAGA and prox-SVRG and acceleration. In *International Conference on Machine Learning (ICML)*, 2018.

Python Software Foundation. Python Language Reference, version 3.6. *http://python.org/*, 2017.

Saharon Rosset, Ji Zhu, and Trevor Hastie. Boosting as a regularized path to a maximum margin classifier. *J. Mach. Learn. Res.*, 5:941–973, 2004.

Pablo Sprechmann, Alex M. Bronstein, and Guillermo Sapiro. Learning efficient structured sparse models. In *International Conference on Machine Learning (ICML)*, pages 615–622, 2012.

Pablo Sprechmann, Roee Litman, and TB Yakar. Efficient supervised sparse analysis and synthesis operators. In *Advances in Neural Information Processing Systems (NIPS)*, pages 908–916, 2013.

Yifan Sun, Halyun Jeong, Julie Nutini, and Mark Schmidt. Are we there yet? manifold identification of gradient-related proximal methods. In *Proceedings of Machine Learning Research*, volume 89 of *Proceedings of Machine Learning Research*, pages 1110–1119. PMLR, 2019.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

Ryan Tibshirani. The lasso problem and uniqueness. *Electron. J. Stat.*, 7:1456–1490, 2013.

Paul Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *J. Optim. Theory Appl.*, 109(3):475–494, 2001.

Zhangyang Wang, Qing Ling, and Thomas S. Huang. Learning deep ℓ_0 encoders. In *AAAI Conference on Artificial Intelligence*, pages 2194–2200, 2015.

Bo Xin, Yizhou Wang, Wen Gao, and David Wipf. Maximal sparsity with deep networks? In *Advances in Neural Information Processing Systems (NIPS)*, pages 4340–4348, 2016.

Yan Yang, Jian Sun, Huibin Li, and Zongben Xu. Deep ADMM-Net for compressive censing MRI. In *Advances in Neural Information Processing Systems (NIPS)*, pages 10–18, 2017.

Willard I Zangwill. Convergence conditions for nonlinear programming algorithms. *Management Science*, 16(1):1–13, 1969.

Jian Zhang and Bernard Ghanem. ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1828–1837, 2018.

A Unfolded optimization algorithms literature summary

In Table A.1, we summarize the prolific literature on learned unfolded optimization procedures for sparse recovery. A particular focus is set on the chosen training loss training which is either supervised, with a regression of z^i from the input x^i for a given training set (x^i, z^i), or unsupervised, where the objective is to minimize the Lasso cost function F_x for each training point x.

Table A.1: Neural network for sparse coding

Reference	Base Algo	Train Loss	Coupled weights	Remarks
Gregor and Le Cun (2010)	ISTA / CD	supervised	×	–
Sprechmann et al. (2012)	Block CD	unsupervised	×	Group ℓ_1
Sprechmann et al. (2013)	ADMM	supervised	N/A	–
Hershey et al. (2014)	NMF	supervised	×	NMF
Wang et al. (2015)	IHT	supervised	×	Hard-thresholding
Xin et al. (2016)	IHT	supervised	×/✓	Hard-thresholding
Giryes et al. (2018)	PGD/IHT	supervised	N/A	Group ℓ_1
Yang et al. (2017)	ADMM	supervised	N/A	–
Adler et al. (2017)	ADMM	supervised	N/A	Wasserstein distance with z^*
Borgerding et al. (2017)	AMP	supervised	×	–
Moreau and Bruna (2017)	ISTA	unsupervised	×	–
Chen et al. (2018)	ISTA	supervised	✓	Linear convergence rate
Ito et al. (2018)	ISTA	supervised	✓	MMSE shrinkage non-linearity
Zhang and Ghanem (2018)	PGD	supervised	✓	Sparsity of Wavelet coefficients
Liu et al. (2019)	ISTA	supervised	✓	Analytic weight W_{ALISTA}
Proposed	ISTA	unsupervised	✓	–

B Proofs of Section 3's results

B.1 Proof of Proposition 3.1

We consider that the solution of the Lasso is unique, following the result of Tibshirani (2013)[Lemmas 4 and 16] when the entries of D and x come from a continuous distribution.

Proposition 3.1 (Convergence, finite-time support identification and safe regime)**.** *When Assumption 2.2 holds, the sequence* $(z^{(t)})$ *generated by the algorithm converges to* $z^* = \arg\min F_x$.

Further, there exists an iteration T^* *such that for* $t \geq T^*$, $\operatorname{supp}(z^{(t)}) = \operatorname{supp}(z^*) \triangleq S^*$ *and Condition* $\star$ *is always statisfied.*

Proof. Let $z^{(t)}$ be the sequence of iterates produced by Algorithm 1. We have a *descent function*

$$F_x(z^{(t+1)}) - F_x(z^{(t)}) \leq -\frac{\gamma}{2}\|z^{(t+1)} - z^{(t)}\|^2 \leq -\frac{\min \|D_j\|}{2}\|z^{(t+1)} - z^{(t)}\|^2 \ , \qquad (17)$$

where $\gamma = L_S$ if Condition $\star$ is met, and L otherwise. Additionally, the iterates are bounded because $F_x(z^{(t)})$ decreases at each iteration and F_x is coercive. Hence we can apply Zangwill's Global Convergence Theorem (Zangwill, 1969). Any z^* accumulation point of $(z^{(t)})_{t\in\mathbb{N}}$ is a minimizer of F_x .

Since we only consider the case where the minimizer is unique, the bounded sequence $(z^{(t)})_{t\in\mathbb{N}}$ has a unique accumulation point, thus converges to z^* .

The support identification is a simplification of a result of Hale et al. (2008), we include it here for completeness.

Lemma B.1 (Approximation of the soft-thresholding)**.** *Let* $z \in \mathbb{R}, \nu > 0$. *For* ϵ *small enough, we have*

$$\mathrm{ST}(z+\epsilon, \nu) = \begin{cases} 0 \ , & \text{if } |z| < \nu \ , \\ \max(0, \epsilon)\,\mathrm{sign}(z) \ , & \text{if } |z| = \nu \ , \\ z + \epsilon - \nu\,\mathrm{sign}\, z \ , & \text{if } |z| > \nu \ . \end{cases} \tag{18}$$

Let $\rho > 0$ be such that Equation (18) holds for $\nu = \lambda/L$, every $\epsilon < \rho$, and every $z = z_j^* - \frac{1}{L} D_j^\top (Dz^* - x)$.

Let $t \in \mathbb{N}$ such that $z^{(t)} = z^* + \epsilon$, with $\|\epsilon\| \le \rho$. With $\epsilon' \triangleq (\mathrm{Id} - \frac{1}{L} D^\top D)\epsilon$, we also have $\|\epsilon'\| \le \rho$. Let $j \in [\![1, m]\!]$.

If $j \notin E$, $|z_j^* - \frac{1}{L} D_j^\top (Dz^* - x)| = |\frac{1}{L} D_j^\top (Dz^* - x)| < \lambda/L$ hence $\mathrm{ST}(z_j^* - \frac{1}{L} D_j^\top (Dz^* - x) + \epsilon'_j, \lambda/L) = 0$.

If $j \in E$, $|z_j^* - \frac{1}{L} D_j^\top (Dz^* - x)| = |z_j^* + \frac{\lambda}{L}\,\mathrm{sign}\, z_j^*| > \lambda/L$, and $\mathrm{sign}\,\mathrm{ST}(z_j^* - \frac{1}{L} D_j^\top (Dz^* - x) + \epsilon'_j, \lambda/L) = \mathrm{sign}\, z_j^*$.

The same reasoning can be applied with ρ' such that Equation (18) holds for $\nu = \lambda/L_{S^*}$, every $\epsilon < \rho'$, and every $z = z_j^* - \frac{1}{L_S^*} D_j^\top (Dz^* - x)$. If we introduce $\eta > 0$ such that $\|\epsilon\| \le \eta \implies \|(\mathrm{Id} - \frac{1}{L_{S^*}} D \top D)\epsilon\| \le \rho'$, in the ball of center z^* and radius η , the iteration with step size L_{S^*} identifies the support.

Additionnally, since $\mathrm{Id} - \frac{1}{L_{S^*}} D_{S^*}^\top D_{S^*}$ is non expansive on vectors which support is S^* , the iterations with the step L_{S^*} never leave this ball once they have entered it.

Therefore, once the iterates enter $\mathcal{B}(z^*, \min(\eta, \rho))$, Condition $\star$ is always satisfied.

$\square$

B.2 Proof of Proposition 3.2

Proposition 3.2 (Rates of convergence)**.** *For* $t > T^*$, $F_x(z^{(t)}) - F_x(z^*) \le L_{S^*} \frac{\|z^* - z^{(T^*)}\|^2}{2(t - T^*)}$. *If additionally* $\inf_{\|z\|=1} \|D_{S^*} z\|^2 = \mu^* > 0$, *then the convergence rate for* $t \ge T^*$ *is* $F_x(z^{(t)}) - F_x(z^*) \le (1 - \frac{\mu^*}{L_{S^*}})^{t - T^*} (F_x(z^{(T^*)}) - F_x(z^*))$.

Proof. For $t \ge T^*$, the iterates support is S^* and the objective function is L_{S^*}-smooth instead of L-smooth. It is also μ^* strongly convex if $\mu^* > 0$. The obtained rates are a classical result of the proximal gradient descent method in these cases. $\square$

C Proof of Section 4's Lemmas

C.1 Proof of Lemma 4.2

Lemma 4.2 (Stability of solutions around D_j)**.** *Let* $D \in \mathbb{R}^{n \times m}$ *be a dictionary with non-duplicated unit-normed columns. Let* $c \triangleq \max_{l \neq j} |D_l^\top D_j| < 1$. *Then for all* $j \in [\![1, m]\!]$ *and* $\varepsilon \in \mathbb{R}^m$ *such that* $\|\varepsilon\| < \lambda(1 - c)$ *and* $D_j^\top \varepsilon = 0$, *the vector* $(1 - \lambda)e_j$ *minimizes* F_x *for* $x = D_j + \varepsilon$.

Proof. Let $j \in [\![1, m]\!]$ and let $\varepsilon \in \mathbb{R}^m \cap D_j^\perp$ be a vector such that $\|\varepsilon\| < \lambda(1 - c)$. For notation simplicity, we denote $z^* = z^*(D_j - \varepsilon)$.

$$D_j^\top (Dz^* - D_j - \varepsilon) = D_j^\top (-\lambda D_j - \varepsilon) = -\lambda = -\lambda\,\mathrm{sign}\, z_j^* \ , \tag{19}$$

since $1-\lambda>0$. For the other coefficients $l \in [\![1,m]\!] \setminus \{j\}$, we have

$$
\begin{aligned}
|D_l^\top(Dz^* - D_j - \varepsilon)| &= |D_l^\top(-\lambda D_j - \varepsilon)| \;, && (20)\\
&= |\lambda D_l^\top D_j + D_l^\top \varepsilon)| \;, && (21)\\
&\leq \lambda |D_l^\top D_j| + |D_l^\top \varepsilon| \;, && (22)\\
&\leq \lambda c + \|D_l\| \|\varepsilon\| \;, && (23)\\
&\leq \lambda c + \|\varepsilon\| < \lambda \;, && (24)\\
& && (25)
\end{aligned}
$$

Therefore, $(1-\lambda)e_j$ verifies the KKT conditions (3) and $z^*(D_j+\varepsilon) = (1-\lambda)e_j$. ☐

C.2 Proof of Lemma 4.3

Lemma 4.3 (Weight coupling)**.** *Let $D \in \mathbb{R}^{n\times m}$ be a dictionary with non-duplicated unit-normed columns. Let $\theta = (W, \alpha, \beta)$ a set of parameters. Assume that all the couples $(z^*(x), x) \in \mathbb{R}^m \times \mathcal{B}_\infty$ such that $z^*(x) \in \arg\min F_x(z)$ verify $\phi_\theta(z^*(x), x) = z^*(x)$. Then, $\frac{\alpha}{\beta} W = D$.*

Proof. Let $x \in \mathcal{B}_\infty$ be an input vector and $z^*(x) \in \mathbb{R}^m$ be a solution for the Lasso at level $\lambda > 0$. Let $j \in [\![1,m]\!]$ be such that $z_j^* > 0$. The KKT conditions (3) gives

$$D_j^\top (Dz^*(x) - x) = -\lambda \;\; . \tag{26}$$

Suppose that $z^*(x)$ is a fixed point of the layer, then we have

$$\mathrm{ST}(z_j^*(x) - \alpha W_j^\top (Dz^*(x) - x), \lambda\beta) = z_j^*(x) > 0 \;\; . \tag{27}$$

By definition, $\mathrm{ST}(a,b) > 0$ implies that $a > b$ and $\mathrm{ST}(a,b) = a - b$. Thus,

$$
\begin{aligned}
& z_j^*(x) - \alpha W_j^\top (Dz^*(x) - x) - \lambda\beta = z_j^*(x) && (28)\\
\Leftrightarrow \quad & \alpha W_j^\top (Dz^*(x) - x) + \lambda\beta = 0 && (29)\\
\Leftrightarrow \quad & \alpha W_j^\top (Dz^*(x) - x) - \beta D_j^\top (Dz^*(x) - x) = 0 \qquad \text{by (26)} && (30)\\
\Leftrightarrow \quad & (\alpha W_j - \beta D_j)^\top (Dz^*(x) - x) = 0 \;\; . && (31)
\end{aligned}
$$

As the relation (31) must hold for all $x \in \mathcal{B}_\infty$, it is true for all $D_j + \varepsilon$ for all $\varepsilon \in \mathcal{B}(0, \lambda(1-c)) \cap D_j^\perp$. Indeed, in this case, $\|D^\top(D_j+\varepsilon)\|_\infty = 1$. D verifies the conditions of Lemma 4.2, and thus $z^* = (1-\lambda)e_j$, *i.e.*

$$
\begin{aligned}
(\alpha W_j - \beta D_j)^\top (D(1-\lambda)e_j - (D_j + \varepsilon)) = 0 && (32)\\
(\alpha W_j - \beta D_j)^\top (-\lambda D_j - \varepsilon) = 0 && (33)
\end{aligned}
$$

Taking $\varepsilon = 0$ yields $(\alpha W_j - \beta D_j)^\top D_j = 0$, and therefore Eq. (33) becomes $(\alpha W_j - \beta D_j)^\top \varepsilon = 0$ for all ε small enough and orthogonal to D_j , which implies $\alpha W_j - \beta D_j = 0$ and concludes our proof. ☐

C.3 Proof of Theorem 4.4

Theorem 4.4. *Let $D \in \mathbb{R}^{n\times m}$ be a dictionary with non-duplicated unit-normed columns. Let $\Theta^{(T)} = \{\theta^{(t)}\}_{t=0}^T$ be the parameters of a sequence of LISTA networks such that the transfer function of the layer t is $z^{(t+1)} = \phi_{\theta^{(t)}}(z^{(t)}, x)$. Assume that*

(i) the sequence of parameters converges i.e. $\theta^{(t)} \xrightarrow[t\to\infty]{} \theta^* = (W^*, \alpha^*, \beta^*)$,

(ii) the output of the network converges toward a solution $z^*(x)$ of the Lasso (1) uniformly over the equiregularization set $\mathcal{B}_\infty$, i.e. $\sup_{x\in\mathcal{B}_\infty} \|\Phi_{\Theta^{(T)}}(x) - z^*(x)\| \xrightarrow[T\to\infty]{} 0$.

Then $\frac{\alpha^*}{\beta^*} W^* = D$.

Proof. For simplicity of the notation, we will drop the x variable whenever possible, *i.e.* $z^* = z^*(x)$ and $\phi_\theta(z) = \phi_\theta(z, x)$. We denote $z^{(t)} = \Phi_{\Theta^{(t)}}(x)$ the output of the network with t layers. Let $\epsilon > 0$. By hypothesis (i), there exists T_0 such that for all $t \geq T_0$,

$$\|W^{(t)} - W^*\| \leq \epsilon \quad |\alpha^{(t)} - \alpha^*| \leq \epsilon \quad |\beta^{(t)} - \beta^*| \leq \epsilon \, . \tag{34}$$

By hypothesis (ii), , there exists T_1 such that for all $t \geq T_1$ and all $x \in \mathcal{B}_\infty$,

$$\|z^{(t)} - z^*\| \leq \epsilon \, . \tag{35}$$

Let $x \in \mathcal{B}_\infty$ be an input vector and $t \geq \max(T_0, T_1)$. Using (35), we have

$$\|z^{(t+1)} - z^{(t)}\| \quad \leq \quad \|z^{(t+1)} - z^*\| + \|z^{(t)} - z^*\| \leq 2\epsilon \tag{36}$$

By (i), there exist a compact set $\mathcal{K}_1 \subset \mathbb{R}^{n \times m} \times \mathbb{R}_*^+ \times \mathbb{R}_*^+$ *s.t.* $\theta^{(t)} \in \mathcal{K}_1$ for all $t \in \mathbb{N}$ and $\theta^* \in \mathcal{K}$. The input x is taken in a compact set $\mathcal{B}_\infty$ and as $z^* = \arg\min_z F_x(z)$, we have $\lambda\|z\|_1 \leq F_x(z^*) \leq F_x(0) = \|x\|$ thus z^* is also in a compact set $\mathcal{K}_2$. We consider the function $f(z, x, \theta) = \mathrm{ST}(z - \alpha W^\top(Dz - x), \beta)$ on the compact set $\mathcal{K}_2 \times \mathcal{B}_\infty \times \mathcal{K}_1$. This function is continuous and piece-wise linear on a compact set. It is thus L-Lipschitz and thus

$$\|\phi_{\theta^{(t)}}(z^{(t)}) - \phi_{\theta^{(t)}}(z^*)\| \quad \leq \quad L\|z^{(t)} - z^*\| \leq L\epsilon \tag{37}$$

$$\|\phi_{\theta^*}(z^*) - \phi_{\theta^{(t)}}(z^*)\| \quad \leq \quad L\|\theta^{(t)} - \theta^*\| \leq L\epsilon \tag{38}$$

Using these inequalities, we get

$$\begin{aligned}
\|\phi_{\theta^*}(z^*, x) - z^*\| \quad \leq \quad & \underbrace{\|\phi_{\theta^*}(z^*) - \phi_{\theta^{(t)}}(z^*)\|}_{< L\epsilon \text{ by } (38)} + \underbrace{\|\phi_{\theta^{(t)}}(z^*) - \phi_{\theta^{(t)}}(z^{(t)})\|}_{< L\epsilon \text{ by } (37)} && (39) \\
& + \underbrace{\|\phi_{\theta^{(t)}}(z^{(t)}) - z^{(t)}\|}_{< 2\epsilon \text{ by } (36)} + \underbrace{\|z^{(t)} - z^*\|}_{< \epsilon \text{ by } (35)} \\
\leq \quad & (2L + 3)\epsilon \, . && (40)
\end{aligned}$$

As this result holds for all $\epsilon > 0$ and all $x \in \mathcal{B}_\infty$, we have $\phi_{\theta^*}(z^*) = z^*$ for all $x \in \mathcal{B}_\infty$. We can apply the Lemma 4.3 to conclude this proof. □

D Experimental setups and supplementary figures

Dictionary generation: Unless specified otherwise, to generate synthetic dictionaries, we first draw a random i.i.d. Gaussian matrix $\hat{D} \in \mathbb{R}^{n \times m}$. The dictionary is obtained by normalizing the columns: $D_{ij} = \frac{1}{\|\hat{D}_{i:}\|}\hat{D}_{ij}$.

Samples generation: The samples x are generated as follows: Random i.i.d. Gaussian samples $\hat{x} \in \mathbb{R}^n$ are generated. We then normalize them: $x = \frac{1}{\|D^\top \hat{x}\|_\infty}\hat{x}$, so that $x \in \mathcal{B}_\infty$.

Training the networks Since the loss function and the network are continuous but non-differentiable, we use sub-gradient descent for training. The sub-gradient of the cost function with respect to the parameters of the network is computed by automatic differentiation. We use full-batch sub-gradient descent with a backtracking procedure to find a suitable learning rate. To verify that we do not overfit the training set, we always check that the test loss and train loss are comparable.

Main text figures setup

- Figure 2: We generate a random dictionary of size 10×50. We take $\lambda = 0.5$, and a random sample $x \in \mathcal{B}_\infty$. F_x^* is computed by iterating ISTA for 10000 iterations.
- Figure 4: We generate a random dictionary of size 10×20. We take $\lambda = 0.2$. We generate a training set of $N = 1000$ samples $(x^i)_{i=1}^{1000} \in \mathcal{B}_\infty$. A 20 layers SLISTA network is trained by gradient descent on these data. We report the learned step sizes. For each layer t of the network and each training sample x, we compute the support at the output of the t-th layer, $S(x, t) = \mathrm{supp}(z^{(t)}(x))$. For each t, we display the quantiles of the distribution of the $(1/L_{S(x^i,t)})_{i=1}^{1000}$.
- Figure 5: A random 10×20 dictionary is generated. We take 1000 training samples, and $\lambda = 0.05$. A 40 layers LISTA network is trained by gradient descent on those samples. We report the quantity $\|\alpha^{(t)}W^{(t)} - \beta^{(t)}D\|_F$ for each layer t.

Supplementary experiments

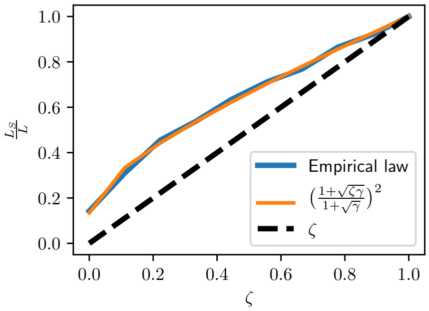


Figure D.1: Illustration of Proposition 3.4. A toy Gaussian dictionary is generated with $n = 200$, $m = 600$ so that $\gamma = 3$. We compute its Lipschitz constant L . For ζ between 0 and 1 , we extract $\lfloor \zeta m \rfloor$ columns at random and compute the corresponding Lipschitz constant L_S . The plot shows an almost perfect fit between the empirical law and the theoretical limit (10).

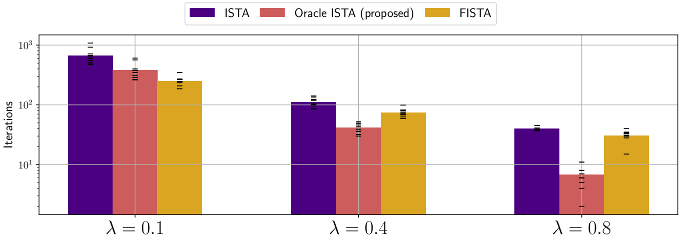


Figure D.2: Comparison between ISTA, FISTA and Oracle-ISTA for different levels of regularization on a Gaussian dictionnary, with $n = 100$ and $m = 200$. We report the average number of iterations taken to reach a point z such that $F_x(z) < F_x^* + 10^{-13}$. The experiment is repeated 10 times, starting from random points in $\mathcal{B}_\infty$. OISTA is always faster than ISTA, and is faster than FISTA for high regularization.

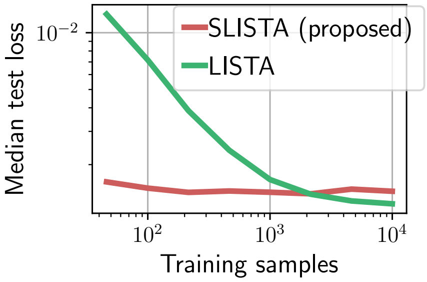


Figure D.3: Learning curves of SLISTA and LISTA. Random Gaussian dictionaries with $n = 10$ and $m = 20$ are generated. We take $\lambda = 0.3$. Networks with 10 layers are fit on those dictionaries, and their test loss is reported for different number of training samples. The process is repeated 100 times; the curves shown display the median of the test-loss.