NEURALLY AUGMENTED ALISTA

Anonymous authors
Paper under double-blind review

ABSTRACT

It is well-established that many iterative sparse reconstruction algorithms can be unrolled to yield a learnable neural network for improved empirical performance. A prime example is learned ISTA (LISTA) where weights, step sizes and thresholds are learned from training data. Recently, Analytic LISTA (ALISTA) has been introduced, combining the strong empirical performance of a fully learned approach like LISTA, while retaining theoretical guarantees of classical compressed sensing algorithms and significantly reducing the number of parameters to learn. However, these parameters are trained to work in expectation, often leading to suboptimal reconstruction of individual targets. In this work we therefore introduce Neurally Augmented ALISTA, in which an LSTM network is used to compute step sizes and thresholds individually for each target vector during reconstruction. This adaptive approach is theoretically motivated by revisiting the recovery guarantees of ALISTA. We show that our approach further improves empirical performance in sparse reconstruction, in particular outperforming existing algorithms by an increasing margin as the compression ratio becomes more challenging.

1 INTRODUCTION AND RELATED WORK

Compressed sensing deals with the problem of recovering a sparse vector from very few compressive linear observations, far less than its ambient dimension. Fundamental works of Candes, Romberg, Tao and Donoho (Candès et al., 2006; Donoho, 2006) show that this can be achieved in a robust and stable manner with computationally tractable algorithms given that the observation matrix fulfills certain conditions, for an overview see Foucart & Rauhut (2017). Formally, consider the set of s-sparse vectors in $\mathbb{R}^N$, i.e. $\Sigma_s^N := \left\{x \in \mathbb{R}^N \middle| \|x\|_0 \leq s\right\}$ where the size of the support of x is denoted by $\|x\|_0 := |\text{supp}(x)| = |\{i : x_i \neq 0\}|$. Furthermore, let $\Phi \in \mathbb{R}^{M \times N}$ be the measurement matrix, with typically $M \ll N$. For a given noiseless observation $y = \Phi x^*$ of an unknown but s-sparse $x^* \in \Sigma_s^N$ we therefore wish to solve:

$$\underset{x}{\text{argmin}} \, \|x\|_0 \quad \text{s.t.} \quad y = \Phi x \tag{1}$$

In (Candès et al., 2006) it has been shown, that under certain assumptions on Φ, the solution to the combinatorial problem in (1) can be also obtained by a convex relaxation where one instead minimizes the ℓ_1–norm of x. The Lagrangian formalism yields then an unconstrained optimization problem also known as LASSO (Tibshirani, 1996), which penalizes the ℓ_1-norm via the hyperparameter $\lambda \in \mathbb{R}$:

$$\hat{x} = \underset{x}{\text{argmin}} \, \frac{1}{2}\|y - \Phi x\|_2^2 + \lambda \|x\|_1 \tag{2}$$

A very popular approach for solving this problem is the iterative shrinkage thresholding algorithm (ISTA) (Daubechies et al., 2004), in which a reconstruction $x^{(k)}$ is obtained after k iterations from initial $x^{(0)} = 0$ via the iteration:

$$x^{(k+1)} = \eta_{\lambda/L}\Big(x^{(k)} + \frac{1}{L}\Phi^T(y - \Phi x^{(k)})\Big) \tag{3}$$

where η_θ is the soft thresholding function given by $\eta_\theta(x) = \text{sign}(x)\max(0, |x| - \theta)$ (applied coordinate-wise) and L is the Lipschitz constant (i.e. the largest eigenvalue) of $\Phi^T\Phi$. Famously, the computational graph of ISTA with K iterations can be unrolled to yield Learned ISTA (LISTA) (Gregor & LeCun, 2010), a K-layer neural network in which all parameters involved can be trained (each layer k has an individual threshold parameter and individual or shared matrix weights) using

backpropagation and gradient descent. LISTA achieves impressive empirical reconstruction performance for many sparse datasets but loses the theoretical guarantees of ISTA. Bridging the gap between LISTA's strong reconstruction quality and the theoretical guarantees for ISTA, ALISTA (Liu et al., 2019) was introduced. ALISTA, introduces a matrix W^T, related to the measurement matrix Φ^T in (3), which is computed by optimizing the generalized coherence:

$$\mu(W, \Phi) = \inf_{W \in \mathbb{R}^{M \times N}} \max_{i \neq j} W_{:,i}^T \Phi_{:,j} \text{ s.t. } \forall i \in \{1, \dots, N\} : W_{:,i}^T \Phi_{:,i} = 1 \tag{4}$$

Then, contrary to LISTA, all matrices are excluded from learning in order to retain desirable properties such as low coherence. For each layer of ALISTA, only a scalar step size parameter $\gamma^{(k)}$ and a scalar threshold $\theta^{(k)}$ is learned from the data, yielding the iteration:

$$x^{(k+1)} = \eta_{\theta^{(k)}}\left(x^{(k)} - \gamma^{(k)} W^T (\Phi x^{(k)} - y)\right) \tag{5}$$

As in LISTA, the parameters for ALISTA are learned end-to-end using backpropagation and stochastic gradient descent by empirically minimizing the reconstruction error:

$$\min_{\theta^{(1)}, \dots, \theta^{(K)}, \gamma^{(1)}, \dots, \gamma^{(K)}} \mathbb{E}_{x^*}\left[\|x^{(K)} - x^*\|_2^2\right] \tag{6}$$

The authors rigorously upper-bound the reconstruction error of ALISTA in the noiseless case and demonstrate strong empirical reconstruction quality even in the noisy case. The empirical performance similar to LISTA, the retained theoretical guarantees, and the reduction of number of parameters to train from either $O(KM^2 + NM)$ in vanilla LISTA or $O(MNK)$ in the variant of LISTA-CPSS (Chen et al., 2018) to just $O(K)$, make ALISTA an appealing algorithm to study and extend.

In (Ablin et al., 2019), instead of directly focusing on the reconstruction problem, where λ is not known a priori, analytical conditions for optimal step sizes in ISTA are derived for LASSO, yielding Stepsize-ISTA. Stepsize-ISTA is a variant of LISTA in which the measurement matrices are exempt from training like in ALISTA, outperforming existing approaches to directly solving LASSO.

Thresholds that are adaptive to the current target vector have been explored in ALISTA-AT (Kim & Park, 2020). Following the majorization-minimization method, component-wise thresholds are computed from previous iterations. In a particular case this yields $\theta_i^{(k)} = 1/(1 + |x_i^{(k-1)}|/\epsilon)$ for some $\epsilon > 0$, known as iterative reweighted ℓ_1-minimization. By unrolling this algorithm, the authors demonstrate superior recovery over ALISTA for a specific setting of M, N and s.

In a related approach (Wu et al., 2020) identify undershooting, meaning that reconstructed components are smaller than target components, as a shortcoming of LISTA and propose Gated-LISTA to address these issues. The authors introduce gain and overshoot gates to LISTA, which can amplify the reconstruction after each iteration before and after thresholding, yielding an architecture resembling GRU cells (Cho et al., 2014). The authors demonstrate better sparse reconstruction than previous LISTA-variants and also show that adding their proposed gates to ALISTA, named AGLISTA, it is possible to improve its performance in the same setting of M, N and s as ALISTA-AT.

In this paper, motivated by essential proof steps of ALISTA's recovery guarantee, we propose an alternative method for adaptively choosing thresholds and step sizes during reconstruction. Our method directly extends ALISTA by using a recurrent neural network to predict thresholds and step sizes depending on an estimate of the ℓ_1-error between the reconstruction and the unknown target vector after each iteration. We refer to our method as Neurally Augmented ALISTA (NA-ALISTA), as the method falls into the general framework of neural augmentation of unrolled algorithms (Welling, 2020; Monga et al., 2019; Diamond et al., 2017). The rest of the paper is structured as follows: we provide theoretical motivation for NA-ALISTA in Section 2, before describing our method in detail in Section 3. In Section 4, we demonstrate experimentally that NA-ALISTA achieves state-of-the-art performance in all evaluated settings. To summarize, our main contributions are:

1. We introduce Neurally Augmented ALISTA (NA-ALISTA), an algorithm which learns to adaptively compute thresholds and step-sizes for individual target vectors during recovery. The number of parameters added does not scale with the problem size.
2. We provide theoretical motivation inspired by guarantees for sparse reconstruction which show that NA-ALISTA can achieve arrive tighter error bounds depending on the target x^*.
3. We find that NA-ALISTA empirically outperforms ALISTA and other state-of-the-art algorithms in all evaluated settings and that the gains increase with decreasing M/N.

2 THEORETICAL MOTIVATION

The thresholds $\theta^{(k)}$ in (5) play an important role in the analysis of ALISTA. While the authors of (Liu et al., 2019) prove that $\theta^{(k)}$ must be larger than a certain value in order to guarantee no false positives in the support of the reconstruction $x^{(k)}$, the thresholds $\theta^{(k)}$ also appear as an additive term in the reconstruction error upper bound. Thus, to guarantee good reconstruction $\theta^{(k)}$ should be just slightly larger than the value it must surpass in order to both minimize the error and verify the assumption. In this section, we repeat key insights from ALISTA and motivate the choice of adaptive thresholds - the key improvement in our proposed NA-ALISTA. More specifically, we repeat the conditions under which ALISTA guarantees no false positives and highlight an intermediate step in the error bound from (Liu et al., 2019), which tightens when the thresholds can adapt to specific instances of x^*.

Assumption (adapted from Assumption 1 from (Liu et al., 2019))
Let $x^* \in \Sigma_s^N$ be a fixed s–sparse target vector. Let W be such that it attains the infimum of the generalized coherence with Φ (as in (4)) and denote this generalized coherence as $\tilde{\mu} = \mu(W, \Phi)$. Let $s < (1 + 1/\tilde{\mu})/2$. Let $\gamma^{(1)}, \dots, \gamma^{(K)}$ be any sequence of scalars taking values in $(0, \frac{2}{2\tilde{\mu}s - \tilde{\mu} + 1})$ and $\theta^{(1)}, \dots, \theta^{(K)}$ with:

$$\theta^{(k)} \geq \gamma^{(k)} \tilde{\mu} \|x^{(k)} - x^*\|_1 \tag{7}$$

Because in ALISTA, the thresholds $\gamma^{(1)}, \dots, \gamma^{(K)}$ and stepsizes $\theta^{(1)}, \dots, \theta^{(K)}$ are optimized in expectation over the training data, the inequality in (7) holds only in the general case if the thresholds are larger than the worst case ℓ_1-error committed by the algorithm over all training vectors x^* i.e.:

$$\theta^{(k)} \geq \gamma^{(k)} \tilde{\mu} \sup_{x^*} \|x^{(k)} - x^*\|_1 \tag{8}$$

This is needed in order to fulfill the Assumption. Under these conditions it is guaranteed that no false positives are in the support of the reconstruction:

No false positives (Lemma 1 from (Liu et al., 2019))
Under the settings of the Assumption, it holds that:

$$\text{supp}(x^{(k)}) \subseteq \text{supp}(x^*) \tag{9}$$

However, the threshold $\theta^{(k)}$ also reappears in the error upper bound. Here we employ an intermediate step of the error upper bound from (Liu et al., 2019):

Reconstruction error upper bound (Theorem 1 from (Liu et al., 2019))
Under the settings of the Assumption, it holds that:

$$\|x^{(k+1)} - x^*\|_2 \leq \|x^{(k+1)} - x^*\|_1 \leq \tilde{\mu}\gamma^{(k)}(s-1)\|x^{(k)} - x^*\|_1 + \theta^{(k)}s + |1 - \gamma^{(k)}|\|x^{(k)} - x^*\|_1 \tag{10}$$

Where the first inequality holds for all real vectors and the second inequality is derived in detail in Appendix A of (Liu et al., 2019). It is therefore desirable that $\theta^{(k)}$ is as small as possible, but such that it still satisfies (7). This means that ALISTA has to learn thresholds at least proportional to the largest possible committed ℓ_1-error over all possible x^* in order to guarantee good reconstruction, for which it is in turn penalized in the error bound.

However, the thresholds that make the error bound tighter vary depending on the x^* that is to be recovered. In fact, if an algorithm would have access to $\|x^{(k)} - x^*\|_1$ and were allowed to choose thresholds adaptively, depending on this quantity, the more relaxed inequality (7) could be employed directly, without taking the supremum. An algorithm which approximates such thresholds, resulting in a tighter error bound, is the aim of this paper.

3 NEURALLY AUGMENTED ALISTA

In order to tighten the error upper bound in (10), we introduce Neurally Augmented ALISTA (NA-ALISTA), in which we adaptively predict thresholds $\theta^{(k,x^*)}$ depending on the current estimate for the ℓ_1-error between $x^{(k)}$ and the unknown x^*. As can be observed from (7), such $\theta^{(k,x^*)}$ must be proportional to $\|x^{(k)} - x^*\|_1$.

In theory, this true ℓ_1-error could be recovered exactly. This is because there are no false positives in $x^{(k)}$, making it s-sparse and for a $\tilde{\mu} < 1/(2s-1)$ the column-normalized $W^T\Phi$ is restricted-invertible for any $2s$-sparse input (Foucart & Rauhut, 2017) [Corollary 5.4, p.113]. However, it is infeasible to solve such an inverse problem at every iteration k. Furthermore, in practice the sparsity is often much larger than what is admissible via the coherence bound. For example, in the experiments of (Gregor & LeCun, 2010; Liu et al., 2019; Wu et al., 2020; Kim & Park, 2020), a sparsity of 50 is used with M=250, N=500. This sparsity already exceeds a maximum admitted sparsity of 11 derived from the minimum theoretical coherence of 0.0447 by the Welch Bound (Welch, 1974), implying that such an exact recovery is not possible in practice anyways.

NA-ALISTA is thus largely concerned with learning for each iteration k a good approximation of $\|x^{(k)} - x^*\|_1$. For this, consider the ℓ_1-norms of the residual:

$$r^{(k)} := \|\Phi x^{(k)} - y\|_1 = \|\Phi(x^{(k)} - x^*)\|_1 \tag{11}$$

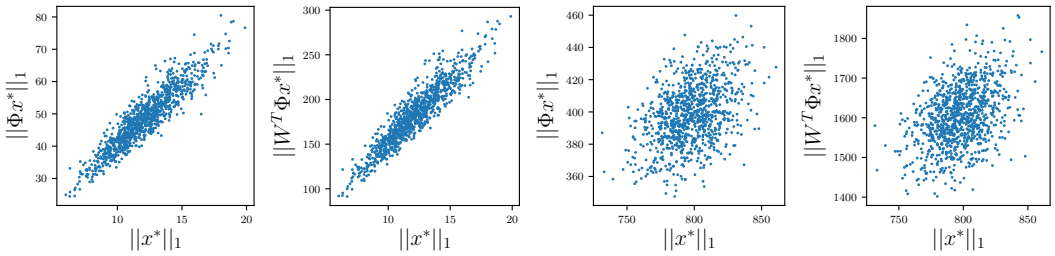
and the iterative update quantity in (5):

$$u^{(k)} := \|W^T(\Phi x^{(k)} - y)\|_1 = \|(W^T\Phi)(x^{(k)} - x^*)\|_1 \tag{12}$$

Both are known to the algorithm even though x^* is unknown. That $r^{(k)}$ and $u^{(k)}$ are useful quantities for approximating the true ℓ_1-error stems from the fact that $W^T\Phi$ has low mutual coherence, thus being a restricted isometry for sparse vectors. This is visualized in Figure 1. Other useful quantities to approximate the true ℓ_1-error are given by $\|x^{(0)} - x^*\|_1, \dots, \|x^{(k-1)} - x^*\|_1$. This is highlighted by Figure 2 and suggests the use of a recurrent neural network in NA-ALISTA. We therefore propose to use an LSTM (Hochreiter & Schmidhuber, 1997) which has two input neurons, receiving $u^{(k)}$ and $r^{(k)}$ at each iteration k. This is used to update the internal state and produce the outputs $\theta^{(k,x^*)}$ and $\gamma^{(k,x^*)}$, which are used to compute the next iteration, producing the update rule:

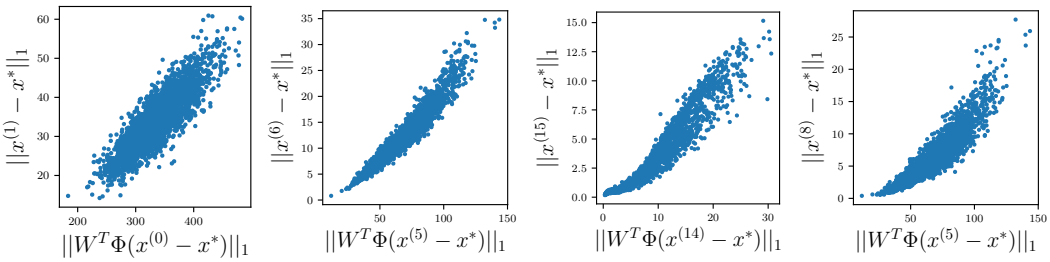
$$x^{(k+1)} = \eta_{\theta^{(k,x^*)}}\left(x^{(k)} - \gamma^{(k,x^*)} W^T(\Phi x^{(k)} - y)\right) \tag{13}$$

A computational expression for NA-ALISTA is given in Algorithm 1. Note that the introduction of LSTM-cells in NA-ALISTA does not significantly increase the required computing power in practice. In fact, in Section 4, we show that small LSTM-cells suffice for best empirical performance, independently of the problem size. Let H be the size of the hidden layer of the LSTM-cells, then



(a) r, $\|x^*\|_0 = 15$ (b) u, $\|x^*\|_0 = 15$ (c) r, $\|x^*\|_0 = 1000$ (d) u, $\|x^*\|_0 = 1000$

Figure 1: Correlation between $\|x^*\|_1$ and $r = \|\Phi x^*\|_1$ and $u = \|W^T\Phi x^*\|_1$ for sparse vectors with $\|x^*\|_0 = 15$ (a) and (b) and non-sparse vectors $\|x^*\|_0 = N$ (b) and (c). Nonzero components of x^* are drawn i.i.d. from $\mathcal{N}(0,1)$ with $N = 1000$. One can see that for sparse x^*, r and u are correlated with $\|x^*\|_1$, whereas there is no obvious correlation for non-sparse vectors.



(a) (b) (c) (d)

Figure 2: Correlation between $u^{(i)}$ and $\|x^{(j)} - x^*\|_1$ in a trained instance of NA-ALISTA for $(i, j) = (0, 1)$ (a), $(5, 6)$ (b), $(14, 15)$ (c), $(5, 8)$ (d). There is a clear correlation, that is even preserved across multiple iterations (d), suggesting the use of a recurrent neural network to predict $\theta^{(k,x^*)}$. Training was performed with the settings described in Section 4, with $N = 1000$, $H = 128$ and $K = 16$.

Algorithm 1: Neurally Augmented ALISTA

Learnable Parameters: initial cell state $c_0 \in \mathbb{R}^H$, initial hidden state $h_0 \in \mathbb{R}^H$, cell state to output matrix $U \in \mathbb{R}^{2\times H}$ and parameters of LSTM cell.
Input: y
$x \leftarrow 0$; $h \leftarrow h_0$; $c \leftarrow c_0$
for $\{1, \dots, K\}$ **do**
  $r \leftarrow \|\Phi x - y\|_1$
  $u \leftarrow \|W^T(\Phi x - y)\|_1$
  $c, h \leftarrow \texttt{LSTM}(c, h, [r, u])$
  $\theta, \gamma \leftarrow \texttt{Softsign}(Uc)$
  $x \leftarrow \eta_\theta\Big(x - \gamma W^T(\Phi x - y)\Big)$
end
Return x;

the computation for a single forward computation of the cell takes $O(H^2)$ computations. As a regular iteration of ALISTA takes $O(MN)$ operations and computing the ℓ_1-norm of the update quantity $W^T(\Phi x^{(k)} - y)$ takes an additional $O(N)$ operations, an iteration of NA-ALISTA requires $O(MN + N + H^2)$ operations. For example, when $M = 250, N = 2000, H = 64$ as in one of the experimental settings in Figure 5, then $H^2/MN = 4096/500000 = 0.008192$, showing that the added computation is negligible in practice.

4 Experiments

In this section, we evaluate NA-ALISTA in a sparse reconstruction task and compare it against ALISTA (Liu et al., 2019), ALISTA-AT (Kim & Park, 2020), AGLISTA (Wu et al., 2020), as well as the classical ISTA (Daubechies et al., 2004) and FISTA (Beck & Teboulle, 2009). To emphasize a fair and reproducible comparison between the models, the code for all experiments listed will be available on GitHub [1].

4.1 Experimental Setup

Following the same experimental setup as (Liu et al., 2019; Wu et al., 2020; Chen et al., 2018; Kim & Park, 2020), the support of $x^* \in \mathbb{R}^N$ is determined via i.i.d. Bernoulli random variables with parameter S/N, leading to an expected sparsity of S. The non-zero components of x^* are then sampled according to $\mathcal{N}(0, 1)$. The entries of Φ are also sampled from $\mathcal{N}(0, 1)$, before each column is normalized to unit ℓ_2-norm. W is then computed by minimizing the generalized coherence in (4)

[1] For purposes of double-blind peer-review, the code is provided in the supplementary material

between W and Φ via the Frobenius-Norm approximation using projected gradient descent. This procedure is identical to (Liu et al., 2019; Wu et al., 2020; Kim & Park, 2020). The Adam optimizer (Kingma & Ba, 2015) is used to minimize the ℓ_2-error from (6) for all algorithms. A test set of 10000 samples is fixed before training and recovery performance is measured with the normalized mean squared error (NMSE):

$$\text{NMSE} = 10 \log_{10} \left(\frac{\mathbb{E}_{x^*}[\|x^{(K)} - x^*\|^2]}{\mathbb{E}_{x^*}[\|x^*\|^2]} \right)$$

A support selection trick was introduced in (Chen et al., 2018) to speed up convergence and stabilize training and has been subsequently used extensively in variants LISTA and ALISTA (Liu et al., 2019; Kim & Park, 2020; Wu et al., 2020). When support selection is used, a hyperparameter $p = (p^{(1)}, \ldots, p^{(K)})$ is set such that for each layer, a certain percentage of the largest absolute values are exempt from thresholding, i.e.:

$$\eta_{(\theta, p^{(k)})}(x)_i = \begin{cases} x_i, & \text{if} \quad |x_i| \geq \lfloor p^{(k)}/N \rfloor\text{-largest value of } |x| \\ \text{sign}(x_i) \max(0, |x_i| - \theta) & \text{else} \end{cases}$$

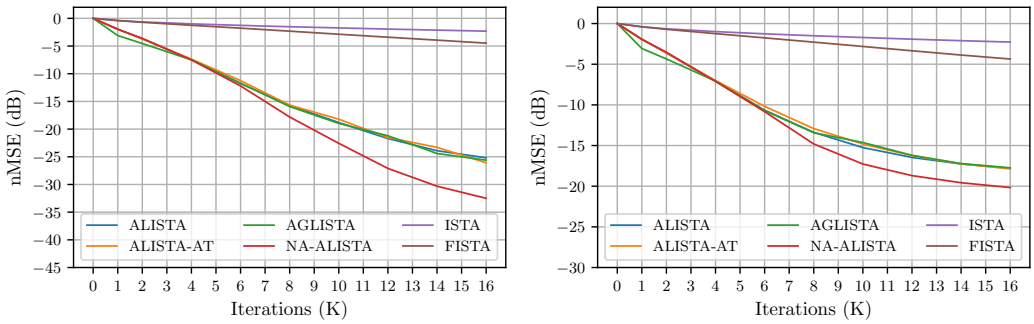
For a fair comparison, we employ support selection in all learned models compared in this paper similarly to the literature (Liu et al., 2019; Chen et al., 2018; Wu et al., 2020; Kim & Park, 2020). Our AGLISTA implementation follows the description in the paper (Wu et al., 2020): we use exponential gain gates and inverse-proportional-based overshoot gains. The λ parameter in ISTA and FISTA was tuned by hand, we found that $\lambda = 0.4$ led to the best performance in our tasks. NA-ALISTA by default uses both $r^{(k)}$ and $u^{(k)}$ as inputs to the LSTM in iteration k.

When not otherwise indicated we use the following settings for experiments and algorithms: M=250, N=1000, S=50, K=16, H=128, and $y = \Phi x^* + z$ with additive white Gaussian noise z with a signal to noise ratio SNR:$= \mathbb{E}(\|\Phi x^*\|_2^2)/\mathbb{E}(\|z\|_2^2) = 40$dB. We train all algorithms for 400 epochs, with each epoch containing 50,000 sparse vectors with a batch size of 512.

4.2 Comparison with Competitors

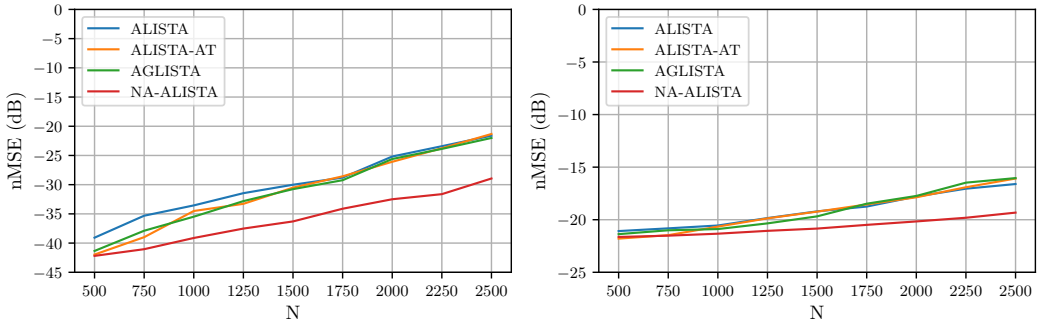
As an established experimental setting to compare the performance of of ISTA-based methods the compressed sensing, previous work (Liu et al., 2019; Kim & Park, 2020; Wu et al., 2020) has focused on a compression level of $M/N = 0.5$ with sparsity S=50 following (Chen et al., 2018). However, practical applications in communication and imaging favor even lower compression rates like $10 \ldots 20\%$, which is why we extend our analysis to more challenging rates. To achieve different compression rates we keep the sparsity S and measurements M constant while increasing N.

As shown in Figure 3, we first fix $N = 2000$ and observe the reconstruction error for a varying amount of iterations. In Figure 4 we then decrease the compression ratio while keeping the sparsity constant. We observe that NA-ALISTA outperforms state-of-the-art adaptive methods in all evaluated scenarios. Whereas for the more established setting from the literature of N=500, the improvement



(a) N=2000, SNR=40dB (b) N=2000, SNR=20dB

Figure 3: The reconstruction error for ALISTA, AGLISTA, ALISTA-AT and NA-ALISTA over the number of iterations K for SNR=40dB (3a) and SNR=20dB (3b). NA-ALISTA outperforms all competitors. Results for settings with smaller N can be found in Appendix A.



(a) SNR=40dB (b) SNR=20dB

Figure 4: Reconstruction error over different compression ratios. For a constant expected sparsity of S=50 and M=250 measurements and K=16 iterations, the input size N varies. Both under a SNR of 40dB and 20dB NA-ALISTA increases its reconstruction margin to competitors as N increases and the compression ratio becomes more challenging.

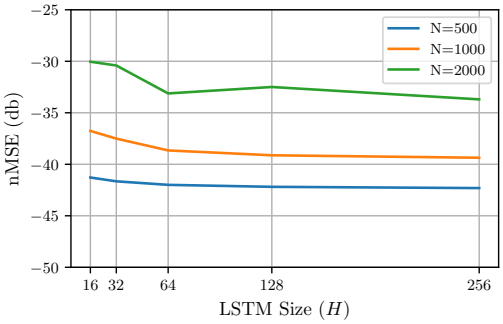


Figure 5: Reconstruction error for varying settings of the LSTM size in NA-ALISTA. Larger N profit more from larger H, but in all settings an exponential increase of the LSTM size only yields a marginal improvement once H=64 is surpassed.

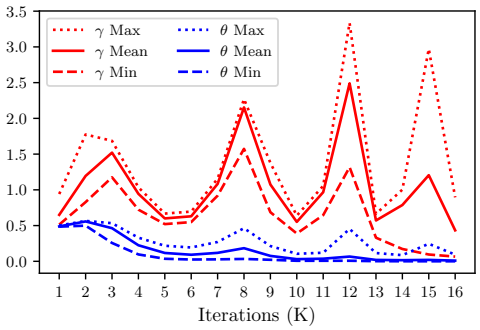


Figure 6: Predicted step sizes and thresholds from a trained instance of NA-ALISTA ($N = 1000, M = 250, S = 50, K = 16$), highlighting the adpativity of NA-ALISTA. Inference to obtain these values is performed on the test set.

of NA-ALISTA is small, this margin increases as the compression ratio becomes more challenging. In Figure 4a the reconstruction error achieved by ALISTA-AT and AGLISTA deteriorates to the performance of ALISTA, while our NA-ALISTA can sustain its advantage over ALISTA even for compression rates up to 0.1 when $N = 2500$. This suggests that our method is interesting to a wider range of practical applications.

To verify that the added computation, determined by the size H of the LSTM, is negligible in practice, we test different settings of H. In Figure 5 we show that an exponential increase in hidden neurons yields only a small error reduction for different N, suggesting that the size H=128 is a sufficient default value for several settings of N. This implies that neural augmentation only marginally affects the runtime. We tested NA-ALISTA using different inputs $r^{(k)}, u^{(k)}$ for the LSTM in Table 1 and conclude that all approximations perform similarly and a single approximation of the ℓ_1-error is

Model	N=500	N=1000	N=2000
NA-ALISTA $r^{(k)}$	-42.00	-39.15	**-32.50**
NA-ALISTA $\{r^{(k)}, u^{(k)}\}$	**-42.18**	-39.12	-32.49
NA-ALISTA $u^{(k)}$	-42.03	**-39.24**	-29.36

Table 1: Reconstruction error in dB for NA-ALISTA with different inputs $r^{(k)}$ and/or $u^{(k)}$ to the LSTM (see (11) and (12)) with K=16, SNR=40. It does not matter which quantities we use to estimate the ℓ_1-error, since all perform equally well as input to the LSTM.

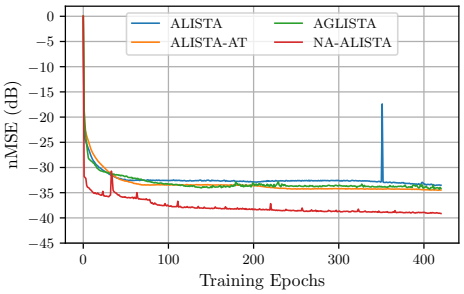


Figure 7: Training curves for N=1000, K=16, SNR=40 from the learned algorithms we compare in this paper, showing that NA-ALISTA outperforms the competitors after only a few epochs of training. Each epoch consists of 50,000 randomly drawn sparse vectors.

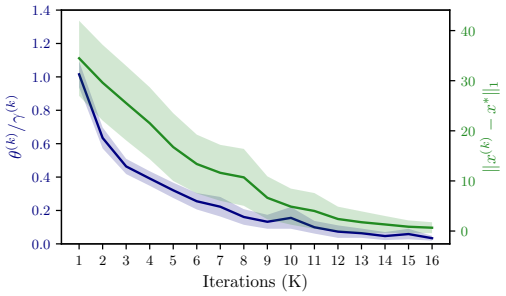


Figure 8: Comparison of the ratio $\theta^{(k)}/\gamma^{(k)}$ with the true ℓ_1-error $||x^* - x^{(k)}||_1$ at each iteration for NA-ALISTA. We report the mean for a batch of randomly drawn test data $\{x^*\}$ along with the standard deviation for each quantity. Together these terms behave as desired, see Eq. (7) and its discussion.

sufficient. However, we observe a slight increase in convergence speed and training stability when using both inputs.

We also evaluate whether the increased empirical performance of NA-ALISTA is truly due to its adaptivity or simply due to its architecture, since the LSTM architecture could in principle enable a more stable optimization of the desired parameters due to more stable gradients. This would imply that when run on a test set, the learned step sizes would not vary depending on the input. Figure 6 shows that this is not the case, since step sizes and thresholds vary within a margin on a test set of 10,000 randomly sampled inputs. Also, the decreasing threshold $\theta^{(k)}$ corresponds to "warm start" behavior for ISTA to first go through a thresholding phase and then through a fitting phase where the threshold becomes essentially zero, see exemplary (Loris, 2009). An additional strength of NA-ALISTA is that it is fast and stable to train, outperforming competitors after only a few epochs, as shown in Figure 7.

As an empirical verification of Assumption 1 in (7) we need to check for every x^*, whether the ratio $\theta^{(k,x^*)}/\gamma^{(k,x^*)}$ is proportional to the ℓ_1-error $||x^* - x^{(k)}||_1$. Since it is infeasible to check the assumption for the infinite set of sparse vectors Σ_s^N, we empirically verify (7) for a sample of inputs from the training distribution. In Figure 8 the means of both values are proportional to each other for such a test sample, suggesting that the reconstruction bound (10) holds for NA-ALISTA.

5 Conclusion and Future Work

In this paper, we propose Neurally Augmented ALISTA (NA-ALISTA), an extension of ALISTA in which the step sizes and thresholds are predicted adaptively to the target vector by a neural network. Besides a theoretical motivation for NA-ALISTA, we experimentally demonstrate that it is able to outperform state-of-the-art algorithms such as ALISTA (Liu et al., 2019), AGLISTA (Wu et al., 2020), and ALISTA-AT (Kim & Park, 2020) in sparse reconstruction in a variety of experimental settings. In particular, NA-ALISTA outperforms the existing algorithms by a wide margin in settings with a large compression.

While in this paper we restrict ourselves to the classical compressed sensing setting, in which s-sparse vectors are reconstructed, neural augmentation provides a more flexible framework for incorporating additional knowledge into classical algorithms. Therefore, an interesting line of future work is to explore how neural augmentation can incorporate notions of structured sparsity or other constraints into sparse reconstruction. There is a plethora of signal processing algorithms, going much beyond variants of ISTA, proximal gradient methods, and even beyond sparse reconstruction in general, which lend itself to an interpretation of a neural network when unfolded (Monga et al., 2019). Identifying algorithms which could benefit from neural augmentation in the way that ALISTA does is left as future work.

REFERENCES

Pierre Ablin, Thomas Moreau, Mathurin Massias, and Alexandre Gramfort. Learning step sizes for unfolded sparse coding. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 32*, pp. 13100–13110. Curran Associates, Inc., 2019. URL http://papers.nips.cc/paper/9469-learning-step-sizes-for-unfolded-sparse-coding.pdf.

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.

Emmanuel J Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on information theory*, 52(2):489–509, 2006.

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*, pp. 9061–9071, 2018.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1179. URL https://www.aclweb.org/anthology/D14-1179.

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: A Journal Issued by the Courant Institute of Mathematical Sciences*, 57(11):1413–1457, 2004.

Steven Diamond, Vincent Sitzmann, Felix Heide, and Gordon Wetzstein. Unrolled optimization with deep priors. *CoRR*, abs/1705.08041, 2017. URL http://arxiv.org/abs/1705.08041.

D L Donoho. Compressed sensing. *IEEE T. Inform. Theory.*, 52(4):1289–1306, apr 2006. doi: 10.1109/TIT.2006.871582.

Simon Foucart and Holger Rauhut. A mathematical introduction to compressive sensing. *Bull. Am. Math*, 54:151–165, 2017.

Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th international conference on international conference on machine learning*, pp. 399–406, 2010.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

D. Kim and D. Park. Element-wise adaptive thresholds for learned iterative shrinkage thresholding algorithms. *IEEE Access*, 8:45874–45886, 2020.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Jialin Liu, Xiaohan Chen, Zhangyang Wang, and Wotao Yin. Alista: Analytic weights are as good as learned weights in lista. In *International Conference on Learning Representations*, 2019.

Ignace Loris. On the performance of algorithms for the minimization of ℓ_1-penalized functionals. *Inverse Problems*, 25(3):035008, jan 2009. doi: 10.1088/0266-5611/25/3/035008.

Vishal Monga, Yuelong Li, and Yonina C Eldar. Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. *arXiv preprint arXiv:1912.10557*, 2019.

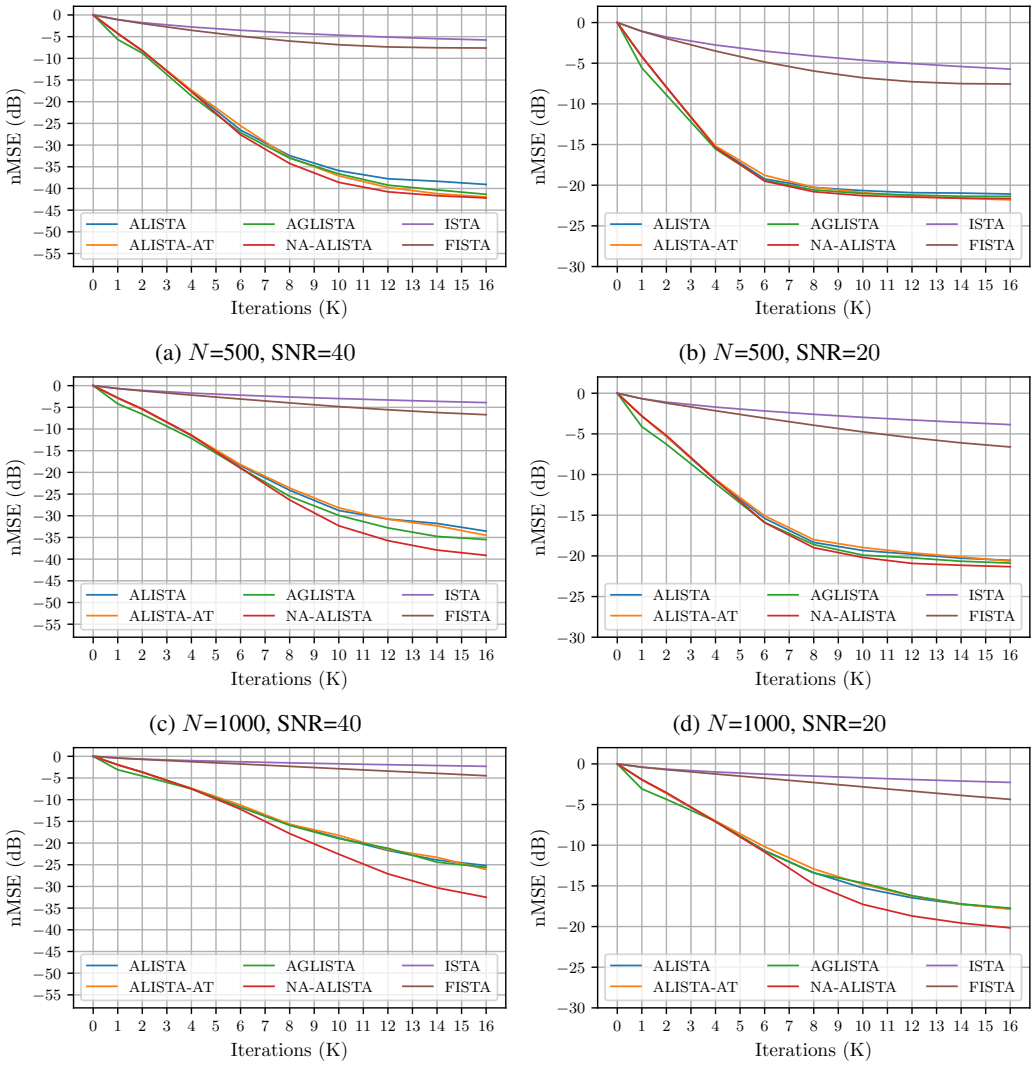
Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

Lloyd Welch. Lower bounds on the maximum cross correlation of signals (corresp.). *IEEE Transactions on Information theory*, 20(3):397–399, 1974.

Max Welling. Neural augmentation in wireless communication. *Keynote at 2020 IEEE International Symposium on Information Theory*, 2020.

Kailun Wu, Yiwen Guo, Ziang Li, and Changshui Zhang. Sparse coding with gated learned ista. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=BygPO2VKPH`.

A SUPPLEMENTARY EXPERIMENTS



(a) N=500, SNR=40

(b) N=500, SNR=20

(c) N=1000, SNR=40

(d) N=1000, SNR=20

(e) N=2000, SNR=40

(f) N=2000, SNR=20

Figure 9: The reconstruction error for ALISTA, ALISTA-AT and NA-ALISTA over the number of iterations run for different noise and N settings. In 9a, for the standard setting in the literature with N= 500 and a noise level of 40dB NA-ALISTA performs on par with competitors after 16 iterations. For an increased N=1000 under the same noise level in 9c, our algorithm outperforms the other methods clearly. For a noise level of 20dB all algorithms perform similarly for N =500 and N=1000 and NA-ALISTA outperforms the others at $N = 2000$.