ADMM-DAD NET: A DEEP UNFOLDING NETWORK
FOR ANALYSIS COMPRESSED SENSING

Vasiliki Kouni[⋆]   *Georgios Paraskevopoulos*[‡,∗]   *Holger Rauhut*[†]   *George C. Alexandropoulos*[⋆]

[⋆] Dep. of Informatics and Telecommunications, National & Kapodistrian University of Athens, Greece
[†] Chair for Mathematics of Information Processing, RWTH Aachen University, Germany
[‡] School of Electrical & Computer Engineering, National Technical University of Athens, Greece
[∗] Institute for Language & Speech Processing, Athena Research Center, Athens, Greece

ABSTRACT

In this paper, we propose a new deep unfolding neural network based on the ADMM algorithm for analysis Compressed Sensing. The proposed network jointly learns a redundant analysis operator for sparsification and reconstructs the signal of interest. We compare our proposed network with a state-of-the-art unfolded ISTA decoder, that also learns an orthogonal sparsifier. Moreover, we consider not only image, but also speech datasets as test examples. Computational experiments demonstrate that our proposed network outperforms the state-of-the-art deep unfolding network, consistently for both real-world image and speech datasets.

Index Terms— Analysis Compressed Sensing, ADMM, deep neural network, deep unfolding.

1. INTRODUCTION

Compressed Sensing (CS) [1] is a modern technique to recover signals of interest $x \in \mathbb{R}^n$ from few linear and possibly corrupted measurements $y = Ax + e \in \mathbb{R}^m$, $m < n$. Iterative optimization algorithms applied on CS are by now widely used [2], [3], [4]. Recently, approaches based on deep learning were introduced [5], [6]. It seems promising to merge these two areas by considering what is called *deep unfolding*. The latter pertains to unfolding the iterations of well-known optimization algorithms into layers of a deep neural network (DNN), which reconstructs the signal of interest.

Related work: Deep unfolding networks have gained much attention in the last few years [7], [8], [9], because of some advantages they have compared to traditional DNNs: they are interpretable, integrate prior knowledge about the signal structure [10], and have a relatively small number of trainable parameters [11]. Especially in the case of CS, many unfolding networks have proven to work particularly well. The authors in [12], [13], [14], [15], [16] propose deep unfolding networks that learn a decoder, which aims at reconstructing x from y. Additionally, most of these networks jointly learn a dictionary that sparsely represents x, along with thresholds used by the original optimization algorithms.

Motivation: Our work is inspired by the articles [14] and [16], which propose unfolded versions of the iterative soft thresholding algorithm (ISTA), with learnable parameters being the sparsifying (orthogonal) basis and/or the thresholds involved in ISTA. The authors then test their frameworks on synthetic data and/or real-world image datasets. In a similar spirit, we derive a decoder by interpreting the iterations of the alternating direction method of multipliers algorithm [17] (ADMM) as a DNN and call it ADMM Deep Analysis Decoding (ADMM-DAD) network. We differentiate our approach by learning a redundant analysis operator as a sparsifier for x, i.e. we employ *analysis sparsity* in CS. The reason for choosing analysis sparsity over its synthesis counterpart is due to some advantages the former has. For example, analysis sparsity provides flexibility in modeling sparse signals, since it leverages the redundancy of the involved analysis operators. We choose to unfold ADMM into a DNN, since most of the optimization-based CS algorithms cannot treat analysis sparsity, while ADMM solves the generalized LASSO problem [18] which resembles analysis CS. Moreover, we test our decoder on speech datasets, not only on image ones. To the best of our knowledge, an unfolded CS decoder has not yet been used on speech datasets. We compare numerically our proposed network[1] to the state-of-the-art learnable ISTA of [16], on real-world image and speech data. In all datasets, our proposed neural architecture outperforms the baseline, in terms of both test and generalization error.

Key results: Our novelty is twofold: a) we introduce a new ADMM-based deep unfolding network that solves the analysis CS problem, namely ADMM-DAD net, that jointly learns an analysis sparsifying operator b) we test ADMM-DAD net on image and speech datasets (while state-of-the-art deep unfolding networks are only tested on synthetic data and images so far). Experimental results demonstrate that ADMM-DAD outperforms the baseline ISTA-net on speech and images, indicating that the redundancy of the learned analysis operator leads to a smaller test MSE and generalization error as well.

Notation: For matrices $A_1, A_2 \in \mathbb{R}^{N \times N}$, we denote by

---

[1] code available at www.github.com/vicky-k-19/ADMM-DAD

$[A_1; A_2] \in \mathbb{R}^{2N \times N}$ their concatenation with respect to the first dimension, while we denote by $[A_1 \mid A_2] \in \mathbb{R}^{N \times 2N}$ their concatenation with respect to the second dimension. We denote by $O_{N \times N}$ a square matrix filled with zeros. We write $I_{N \times N}$ for the real $N \times N$ identity matrix. For $x \in \mathbb{R}$, $\tau > 0$, the soft thresholding operator $\mathcal{S}_\tau : \mathbb{R} \mapsto \mathbb{R}$ is defined in closed form as $\mathcal{S}_\tau(x) = \text{sign}(x) \max(0, |x| - \tau)$. For $x \in \mathbb{R}^n$, the soft thresholding operator acts componentwise, i.e. $(\mathcal{S}_\tau(x))_i = \mathcal{S}_\tau(x_i)$. For two functions $f, g : \mathbb{R}^n \mapsto \mathbb{R}^n$, we write their composition as $f \circ g : \mathbb{R}^n \mapsto \mathbb{R}^n$.

2. MAIN RESULTS

Optimization-based analysis CS: As we mentioned in Section 1, the main idea of CS is to reconstruct a vector $x \in \mathbb{R}^n$ from $y = Ax + e \in \mathbb{R}^m$, $m < n$, where A is the so-called measurement matrix and $e \in \mathbb{R}^m$, with $\|e\|_2 \leq \eta$, corresponds to noise. To do so, we assume there exists a redundant sparsifying transform $\Phi \in \mathbb{R}^{N \times n}$ ($N > n$) called the analysis operator, such that Φx is (approximately) sparse. Using analysis sparsity in CS, we wish to recover x from y. A common approach is the *analysis l_1-minimization* problem

$$\min_{x \in \mathbb{R}^n} \|\Phi x\|_1 \quad \text{subject to} \quad \|Ax - y\|_2 \leq \eta, \quad (1)$$

A well-known algorithm that solves (1) is ADMM, which considers an equivalent generalized LASSO form of (1), i.e.,

$$\min_{x \in \mathbb{R}^n} \frac{1}{2}\|Ax - y\|_2^2 + \lambda\|\Phi x\|_1, \quad (2)$$

with $\lambda > 0$ being a scalar regularization parameter. ADMM introduces the dual variables $z, u \in \mathbb{R}^N$, so that (2) is equivalent to

$$\min_{x \in \mathbb{R}^n} \frac{1}{2}\|Ax - y\|_2^2 + \lambda\|z\|_1 \quad \text{subject to} \quad \Phi x - z = 0. \quad (3)$$

Now, for $\rho > 0$ (penalty parameter), $k \in \mathbb{N}$ and initial points $(x^0, z^0, u^0) = (0, 0, 0)$, the optimization problem in (3) can be solved by the iterative scheme of ADMM:

$$x^{k+1} = (A^T A + \rho \Phi^T \Phi)^{-1}(A^T y + \rho \Phi^T(z^k - u^k)) \quad (4)$$

$$z^{k+1} = \mathcal{S}_{\lambda/\rho}(\Phi x^{k+1} - u^k) \quad (5)$$

$$u^{k+1} = u^k + \Phi x^{k+1} - z^{k+1}. \quad (6)$$

The iterates (4) – (6) are known [17] to converge to a solution $p^\star$ of (3), i.e., $\|Ax^k - y\|_2^2 + \|z^k\|_1 \to p^\star$ and $\Phi x^k - z^k \to 0$ as $k \to \infty$.

Neural network formulation: Our goal is to formulate the previous iterative scheme as a neural network. We substitute first (4) into the update rules (5) and (6) and second (5) into (6), yielding

$$u^{k+1} = (I - W)u^k + Wz^k + b$$
$$\quad - \mathcal{S}_{\lambda/\rho}((-I - W)u^k + Wz^k + b) \quad (7)$$
$$z^{k+1} = \mathcal{S}_{\lambda/\rho}((-I - W)u^k + Wz^k + b),$$

where

$$W = \rho\Phi(A^T A + \rho\Phi^T\Phi)^{-1}\Phi^T \in \mathbb{R}^{N \times N} \quad (8)$$
$$b = b(y) = \Phi(A^T A + \rho\Phi^T\Phi)^{-1}A^T y \in \mathbb{R}^{N \times 1}. \quad (9)$$

We introduce $v_k = [u^k; z^k] \in \mathbb{R}^{2N \times 1}$ and set $\Theta = (-I - W \mid W) \in \mathbb{R}^{N \times 2N}$, $\Lambda = (I - W \mid W) \in \mathbb{R}^{N \times 2N}$ to obtain

$$v_{k+1} = \begin{pmatrix} \Lambda \\ O_{N \times 2N} \end{pmatrix} v_k + \begin{pmatrix} b \\ 0 \end{pmatrix} + \begin{pmatrix} -\mathcal{S}_{\lambda/\rho}(\Theta v_k + b) \\ \mathcal{S}_{\lambda/\rho}(\Theta v_k + b) \end{pmatrix}. \quad (10)$$

Now, we set $\tilde{\Theta} = [\Lambda; O_{N \times 2N}] \in \mathbb{R}^{2N \times 2N}$ and $I_1 = [I_{N \times N}; O_{N \times N}] \in \mathbb{R}^{2N \times N}$, $I_2 = [-I_{N \times N}; I_{N \times N}] \in \mathbb{R}^{2N \times N}$, so that (10) is transformed into

$$v_{k+1} = \tilde{\Theta}v_k + I_1 b + I_2 \mathcal{S}_{\lambda/\rho}(\Theta v_k + b). \quad (11)$$

Based on (11), we formulate ADMM as a neural network with L layers/iterations, defined as

$$f_1(y) = I_1 b(y) + I_2 \mathcal{S}_{\lambda/\rho}(b(y)),$$
$$f_k(v) = \tilde{\Theta}v + I_1 b + I_2 \mathcal{S}_{\lambda/\rho}(\Theta v + b), \quad k = 2, \ldots, L.$$

The trainable parameters are the entries of Φ (or more generally, the parameters in a parameterization of Φ). We denote the concatenation of L such layers (all having the same Φ) as

$$f_\Phi^L(y) = f_L \circ \cdots \circ f_1(y). \quad (12)$$

The final output $\hat{x}$ is obtained after applying an affine map T motivated by (4) to the final layer L, so that

$$\hat{x} = T(f_\Phi^L(y))$$
$$= (A^T A + \rho\Phi^T\Phi)^{-1}(A^T y + \rho\Phi^T(z^L - u^L)), \quad (13)$$

where $[u^L; z^L] = v_L$. In order to clip the output in case its norm falls out of a reasonable range, we add an extra function $\sigma : \mathbb{R}^n \to \mathbb{R}^n$ defined as $\sigma(x) = x$ if $\|x\|_2 \leq B_{\text{out}}$ and $\sigma(x) = B_{\text{out}}x/\|x\|_2$ otherwise, for some fixed constant $B_{\text{out}} > 0$. We introduce the hypothesis class

$$\mathcal{H}^L = \{\sigma \circ h : \mathbb{R}^m \mapsto \mathbb{R}^n : h(y) = T(f_\Phi^L(y)),$$
$$\Phi \in \mathbb{R}^{N \times n}, N > n\} \quad (14)$$

consisting of all the functions that ADMM-DAD can implement. Then, given the aforementioned class and a set $\mathcal{S} = \{(y_i, x_i)\}_{i=1}^s$ of s training samples, ADMM-DAD yields a function/decoder $h_\mathcal{S} \in \mathcal{H}^L$ that aims at reconstructing x from $y = Ax$. In order to measure the difference between x_i and $\hat{x}_i = h_\mathcal{S}(y_i)$, $i = 1, \ldots, s$, we choose the training mean squared error (train MSE)

$$\mathcal{L}_{train} = \frac{1}{s} \sum_{i=1}^s \|h(y_i) - x_i\|_2^2 \quad (15)$$

5 layers	25% CS ratio							
Decoder ＼ Dataset	SpeechCommands		TIMIT		MNIST		CIFAR10	
	test MSE	gen. error	test MSE	gen. error	test MSE	gen. error	test MSE	gen. error
ISTA-net	$0.58 \cdot 10^{-2}$	$0.13 \cdot 10^{-2}$	$0.22 \cdot 10^{-3}$	$0.24 \cdot 10^{-4}$	$0.67 \cdot 10^{-1}$	$0.17 \cdot 10^{-1}$	$0.22 \cdot 10^{-1}$	$0.12 \cdot 10^{-1}$
ADMM-DAD	$\mathbf{0.25 \cdot 10^{-2}}$	$\mathbf{0.16 \cdot 10^{-3}}$	$\mathbf{0.79 \cdot 10^{-4}}$	$\mathbf{0.90 \cdot 10^{-5}}$	$\mathbf{0.23 \cdot 10^{-1}}$	$\mathbf{0.16 \cdot 10^{-3}}$	$\mathbf{0.15 \cdot 10^{-1}}$	$\mathbf{0.11 \cdot 10^{-3}}$

10 layers	40% CS ratio				50% CS ratio			
Decoder ＼ Dataset	SpeechCommands		TIMIT		SpeechCommands		TIMIT	
	test MSE	gen. error	test MSE	gen. error	test MSE	gen. error	test MSE	gen. error
ISTA-net	$0.46 \cdot 10^{-2}$	$0.18 \cdot 10^{-2}$	$0.20 \cdot 10^{-3}$	$0.25 \cdot 10^{-4}$	$0.45 \cdot 10^{-2}$	$0.20 \cdot 10^{-2}$	$0.20 \cdot 10^{-3}$	$0.25 \cdot 10^{-4}$
ADMM-DAD	$\mathbf{0.13 \cdot 10^{-2}}$	$\mathbf{0.58 \cdot 10^{-4}}$	$\mathbf{0.42 \cdot 10^{-4}}$	$\mathbf{0.47 \cdot 10^{-5}}$	$\mathbf{0.87 \cdot 10^{-3}}$	$\mathbf{0.10 \cdot 10^{-4}}$	$\mathbf{0.29 \cdot 10^{-4}}$	$\mathbf{0.30 \cdot 10^{-5}}$

Table 1: Average test MSE and generalization error for 5-layer decoders (all datasets) and 10-layer decoders (speech datasets). Bold letters indicate the best performance between the two decoders.

as loss function. The test mean squared error (test MSE) is defined as

$$\mathcal{L}_{test} = \frac{1}{d} \sum_{i=1}^{d} \|h(\tilde{y}_i) - \tilde{x}_i\|_2^2, \qquad (16)$$

where $\mathcal{D} = \{(\tilde{y}_i, \tilde{x}_i)\}_{i=1}^{d}$ is a set of d test data, not used in the training phase. We examine the generalization ability of the network by considering the difference between the average train MSE and the average test MSE, i.e.,

$$\mathcal{L}_{gen} = |\mathcal{L}_{test} - \mathcal{L}_{train}|. \qquad (17)$$

3. EXPERIMENTAL SETUP

Datasets and pre-processing: We train and test the proposed ADMM-DAD network on two speech datasets, i.e., SpeechCommands [19] (85511 training and 4890 test speech examples, sampled at 16kHz) and TIMIT [20] (phonemes sampled at 16kHz; we take 70% of the dataset for training and the 30% for testing) and two image datasets, i.e. MNIST [21] (60000 training and 10000 test 28×28 image examples) and CIFAR10 [22] (50000 training and 10000 test 32×32 coloured image examples). For the CIFAR10 dataset, we transform the images into grayscale ones. We preprocess the raw speech data, before feeding them to both our ADMM-DAD and ISTA-net: we downsample each .wav file from 16000 to 8000 samples and segment each downsampled .wav into 10 segments.

Experimental settings: We choose a random Gaussian measurement matrix $A \in \mathbb{R}^{m \times n}$ and appropriately normalize it, i.e., $\tilde{A} = A/\sqrt{m}$. We consider three CS ratios $m/n \in \{25\%, 40\%, 50\%\}$. We add zero-mean Gaussian noise with standard deviation $\text{std} = 10^{-4}$ to the measurements, set the redundancy ratio $N/n = 5$ for the trainable analysis operator $\Phi \in \mathbb{R}^{N \times n}$, perform He (normal) initialization for Φ and choose $(\lambda, \rho) = (10^{-4}, 1)$. We also examined different values for λ, ρ, as well as treating λ, ρ as trainable parameters, but both settings yielded identical performance. We evaluate ADMM-DAD for 5 and 10 layers. All networks are trained

using the *Adam* optimizer [23] and batch size 128. For the image datasets, we set the learning rate $\eta = 10^{-4}$ and train the 5- and 10-layer ADMM-DAD for 50 and 100 epochs, respectively. For the audio datasets, we set $\eta = 10^{-5}$ and train the 5- and 10-layer ADMM-DAD for 40 and 50 epochs, respectively. We compare ADMM-DAD to the ISTA-net proposed in [16]. For ISTA-net, we set the best hyper-parameters proposed by the original authors and experiment with 5 and 10 layers. All networks are implemented in PyTorch [24]. For our experiments, we report the average test MSE and generalization error as defined in (16) and (17) respectively.

4. EXPERIMENTS AND RESULTS

We compare our decoder to the baseline of the ISTA-net decoder, for 5 layers on all datasets with a fixed 25% CS ratio, and for 10 layers and both 40% and 50% CS ratios on the speech datasets and report the corresponding average test MSE and generalization error in Table 1. Both the test errors and the generalization errors are always lower for our ADMM-DAD net than for ISTA-net. Overall, the results from Table 1 indicate that the redundancy of the learned analysis operator improves the performance of ADMM-DAD net, especially when tested on the speech datasets. Furthermore, we extract the spectrograms of an example test raw audio file of TIMIT reconstructed by either of the 5-layer decoders. We use 1024 FFT points. The resulting spectrograms for 25% and 50% CS ratio are illustrated in Fig. 1. Both figures indicate that our decoder outperforms the baseline, since the former distinguishes many more frequencies than the latter. Naturally, the quality of the reconstructed raw audio file by both decoders increases, as the CS ratio also increases from 25% to 50%. However, ADMM-DAD reconstructs –even for the 25% CS ratio– a clearer version of the signal compared to ISTA-net; the latter recovers a significant part of noise, even for the 50% CS ratio. Finally, we examine the robustness of both 10-layer decoders. We consider noisy measurements in the test set of TIMIT, taken at 25% and 40% CS ratio, with varying std of the additive Gaussian noise. Fig. 2 shows how the aver-
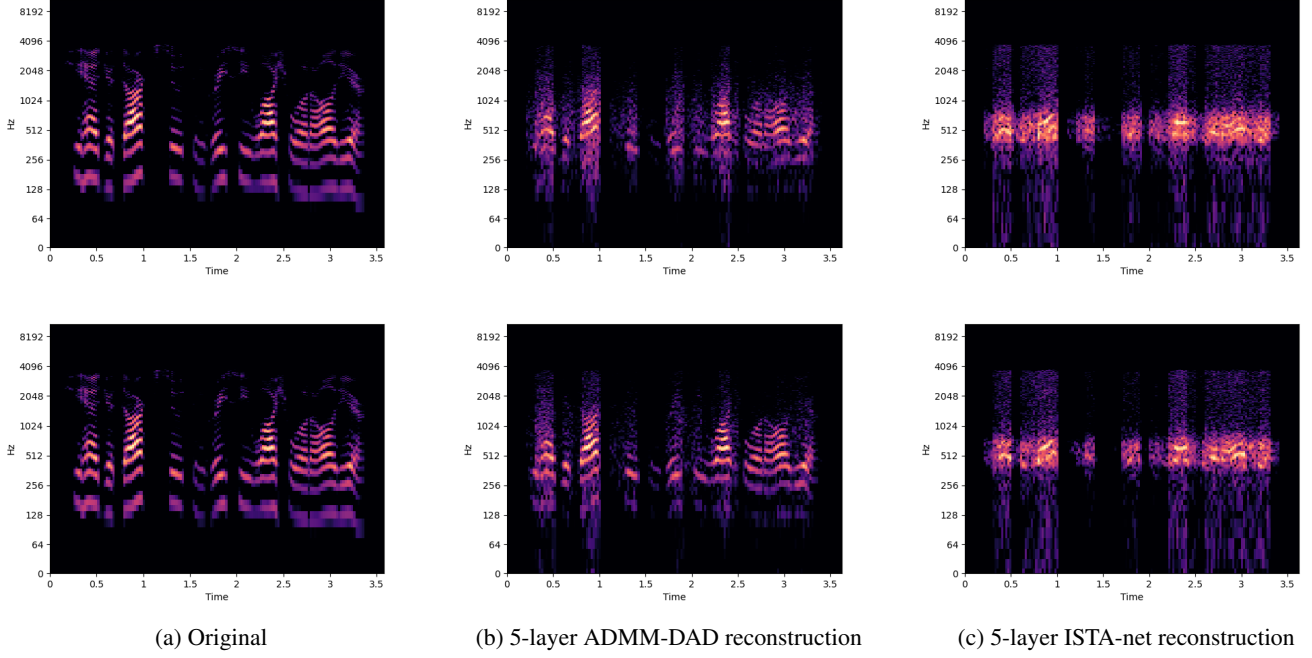
(a) Original   (b) 5-layer ADMM-DAD reconstruction   (c) 5-layer ISTA-net reconstruction

Fig. 1: Spectrograms of reconstructed test raw audio file from TIMIT for 25% CS ratio (top), as well as 50% CS ratio (bottom).

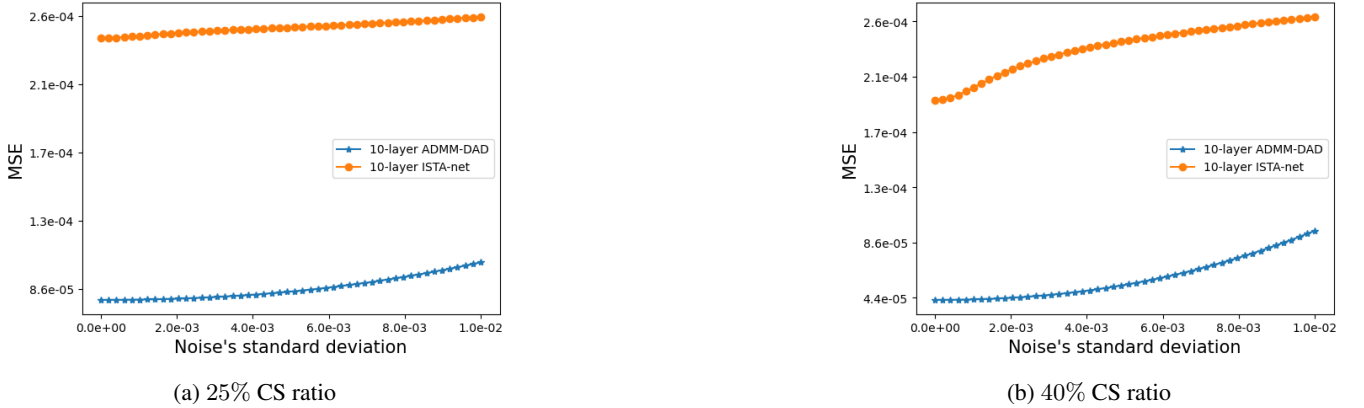(a) 25% CS ratio   (b) 40% CS ratio

Fig. 2: Average test MSE for increasing std levels of additive Gaussian noise. Blue: 10-layer ADMM-DAD, orange: 10-layer ISTA-net.

age test MSE scales as the noise's std increases. Our decoder outperforms the baseline by an order of magnitude and is robust to increasing levels of noise. This behaviour confirms improved robustness when learning a redundant sparsifying dictionary instead of an orthogonal one.

5. CONCLUSION AND FUTURE DIRECTIONS

In this paper we derived ADMM-DAD, a new deep unfolding network for solving the analysis Compressed Sensing problem, by interpreting the iterations of the ADMM algorithm as layers of the network. Our decoder jointly reconstructs the signal of interest and learns a redundant analysis operator, serving as sparsifier for the signal. We compared our framework with a state-of-the-art ISTA-based unfolded network on speech and image datasets. Our experiments confirm improved performance: the redundancy provided by the learned analysis operator yields a lower average test MSE and generalization error of our method compared to the ISTA-net. Future work will include the derivation of generalization bounds for the hypothesis class defined in (14) similar to [16]. Additionally, it would be interesting to examine the performance of ADMM-DAD, when constraining Φ to a particular class of operators, e.g., for Φ being a tight frame.

6. REFERENCES

[1] Emmanuel J. Candès, Justin Romberg, and Terence Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on information theory*, vol. 52, no. 2, pp. 489–509, 2006.

[2] Amir Beck and Marc Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM journal on imaging sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[3] Sundeep Rangan, Philip Schniter, and Alyson K. Fletcher, "Vector approximate message passing," *IEEE Transactions on Information Theory*, vol. 65, no. 10, pp. 6664–6684, 2019.

[4] Elaine T. Hale, Wotao Yin, and Yin Zhang, "Fixed-point continuation applied to compressed sensing: implementation and numerical experiments," *Journal of Computational Mathematics*, pp. 170–194, 2010.

[5] Guang Yang et al., "DAGAN: Deep de-aliasing generative adversarial networks for fast compressed sensing MRI reconstruction," *IEEE transactions on medical imaging*, vol. 37, no. 6, pp. 1310–1321, 2017.

[6] Ali Mousavi, Gautam Dasarathy, and Richard G. Baraniuk, "Deepcodec: Adaptive sensing and recovery via deep convolutional neural networks," *arXiv preprint arXiv:1707.03386*, 2017.

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

[8] Scott Wisdom, John Hershey, Jonathan Le Roux, and Shinji Watanabe, "Deep unfolding for multichannel source separation," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 121–125.

[9] Carla Bertocchi, Emilie Chouzenoux, Marie-Caroline Corbineau, Jean-Christophe Pesquet, and Marco Prato, "Deep unfolding of a proximal interior point method for image restoration," *Inverse Problems*, vol. 36, no. 3, pp. 034005, 2020.

[10] Yuqing Yang, Peng Xiao, Bin Liao, and Nikos Deligiannis, "A robust deep unfolded network for sparse signal recovery from noisy binary measurements," in *2020 28th European Signal Processing Conference (EUSIPCO)*. IEEE, 2021, pp. 2060–2064.

[11] Anand P. Sabulal and Srikrishna Bhashyam, "Joint sparse recovery using deep unfolding with application to massive random access," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 5050–5054.

[12] Zhonghao Zhang, Yipeng Liu, Jiani Liu, Fei Wen, and Ce Zhu, "AMP-Net: Denoising-based deep unfolding for compressive image sensing," *IEEE Transactions on Image Processing*, vol. 30, pp. 1487–1500, 2021.

[13] Daisuke Ito, Satoshi Takabe, and Tadashi Wadayama, "Trainable ISTA for sparse signal recovery," *IEEE Transactions on Signal Processing*, vol. 67, no. 12, pp. 3113–3125, 2019.

[14] Jian Zhang and Bernard Ghanem, "ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1828–1837.

[15] Jian Sun, Huibin Li, Zongben Xu, et al., "Deep ADMM-net for compressive sensing MRI," *Advances in neural information processing systems*, vol. 29, 2016.

[16] Arash Behboodi, Holger Rauhut, and Ekkehard Schnoor, "Compressive sensing and neural networks from a statistical learning perspective," *arXiv preprint arXiv:2010.15658*, 2020.

[17] Stephen Boyd, Neal Parikh, and Eric Chu, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Now Publishers Inc, 2011.

[18] Yunzhang Zhu, "An augmented ADMM algorithm with application to the generalized lasso problem," *Journal of Computational and Graphical Statistics*, vol. 26, no. 1, pp. 195–204, 2017.

[19] Pete Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," *arXiv preprint arXiv:1804.03209*, 2018.

[20] John S. Garofolo, "Timit acoustic phonetic continuous speech corpus," *Linguistic Data Consortium*, 1993.

[21] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[22] Alex Krizhevsky, Geoffrey Hinton, et al., "Learning multiple layers of features from tiny images," 2009.

[23] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[24] Nikhil Ketkar, "Introduction to pytorch," in *Deep learning with python*, pp. 195–208. Springer, 2017.