

# Learning to Optimize: Algorithm Unrolling

Wotao Yin

Decision Intelligence Lab, DAMO Academy  
Alibaba Group US

CVPR Tutorial — June 26, 2022

Acknowledgments of my collaborators:

- ALISTA (ICLR'19) and HyperLISTA (NeurIPS'21) by Xiaohan Chen (UT Austin), Jialin Liu (Alibaba US), Zhangyang Wang (UT Austin)
- Survey paper: *Learning to Optimize: A Primer and A Benchmark*, to appear in JMLR, by Tianlong Chen, Xiaohan Chen, Wuyang Chen, Zhangyang Wang (UT Austin), Jialin Liu (Alibaba US), Howard Heaton (UCLA).
- Earlier survey: *Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing*, IEEE SPM'21, by V. Monga, Y. Li, and Y. Eldar
- GitHub: <https://github.com/VITA-Group/Open-L20>
- This tutorial was created with the help of **Jialin Liu** and **Xiaohan Chen**

## ML vs OPT

Machine learning (ML) is *induction*

- (problems, answers) are given for training
- ML learns to give answers in the future

Optimization (OPT) is *prescription*

- (problems, evaluations) are given, not answers
- OPT finds answers with best evaluations

Learning to optimize (L2O) combines ML and OPT to obtain “better” solutions “faster”, by learning from records of optimization.

## Classic vs Learned

Classic OPT:

- Experts hand-built algorithms based on theory and experience  
For example, Simplex Method and Nesterov Accelerated Gradient Method
- Algorithms are written as iterations in a few lines
- Practitioners pick an algorithm to use

L2O:

- Experts propose L2O templates and training procedures
- Practitioners
  - pick an L2O template
  - prepare training data
  - apply a training procedure

→ obtain a trained algorithm for future problems
- Practitioners are more involved in the design process

## L2O and Neural Networks (NNs)

Many optimization algorithms are similar in form to NNs

$$x^{k+1} \leftarrow \text{nonlinear}(\text{linear}(x^k) + \text{offset}), \quad k = 0, 1, \dots$$

Example: projected gradient iteration for constrained least squares

$$x^{k+1} = \text{Proj}_C(x^k - A^T(Ax^k - b))$$

Difference: in NNs,  $\text{nonlinear}_k$ ,  $\text{linear}_k$ , and  $\text{offset}_k$  vary in  $k$

Question: how to design an NN and use deep learning techniques to improve optimization algorithms?

## NN architecture for L2O

**Model-free:** *fully data driven*, train an input-to-solution NN.

- fast inference: fewer layers than classic optimization iterations
- slow training: too many parameters
- inaccurate solutions: poor generalization, not popular

**Model-based:** *modify* existing optimization algorithms.

Examples:

- Algorithms unrolling (this tutorial)
- Plug-n-play
- Deep equilibrium or fixed-point network

**Survey:** *Learning to Optimize: A Primer and A Benchmark* arXiv:2103.12828, to appear in JMLR.

## **Remaining of this Tutorial**

- Definition and examples
- Milestones of the LISTA series of work
- Theory and safeguard
- Conclusions

## Algorithm Unrolling (AU)

AU consists of two steps

- Pick a classic iteration and unroll it to an NN
- Select a set of NN parameters to learn

LASSO example: assume  $b = Ax^{\text{true}} + \text{noise}$ ; recover  $x^{\text{true}}$  by optimization

$$x^{\text{lasso}} \leftarrow \underset{x}{\text{minimize}} \frac{1}{2} \|Ax - b\|_2^2 + \lambda \|x\|_1$$

also known as  $\ell_1$ -regularized least-squares and compressed sensing

Iterative soft-thresholding algorithm (ISTA):

$$x^{k+1} = \eta_{\lambda\alpha} \left( x^k - \alpha A^T (Ax^k - b) \right)$$

- convergence requires a proper stepsize  $\alpha$  or line search
- the gradient-descent step reduces  $\frac{1}{2} \|Ax - b\|^2$
- the soft-thresholding step  $\eta_{\lambda\alpha}(\cdot)$  reduces  $\lambda \|x\|_1$

Introduce scalar  $\theta = \lambda\alpha$  and matrices  $W_1 = \alpha A^T$  and  $W_2 = I - \alpha A^T A$ .

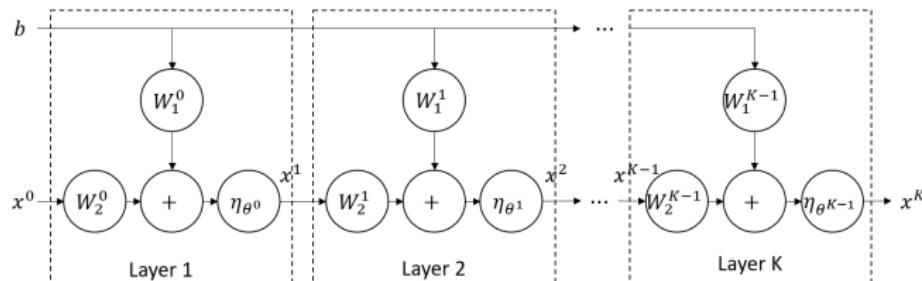
Rewrite ISTA as

$$x^{k+1} = \eta_\theta(W_1 b + W_2 x^k).$$

Unrolling: introduce  $\theta^k, W_1^k, W_2^k, k = 0, 1, \dots$ , as free parameters and re-define

$$x^{k+1} = \eta_{\theta^k}(W_1^k b + W_2^k x^k)$$

which resembles a DNN:



Once  $\theta^k, W_1^k, W_2^k$  are chosen, the algorithm is defined.

Gregor & LeCun'10: find  $\theta^k, W_1^k, W_2^k$ ,  $k = 0, 1, \dots$ , such that the algorithm converges very fast for a set of LASSO instances with the same  $A$ .

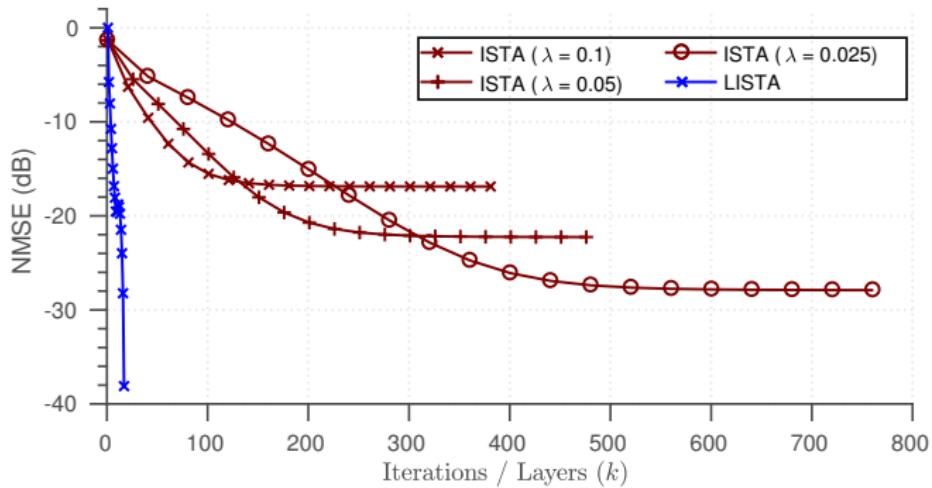
Fix random matrix  $A$ , generate a set of sparse  $x_i^{\text{true}}$ , with varying supports, and  $b_i = Ax_i^{\text{true}} + \text{noise}_i$ . Form the training set  $D = \{(x_i^{\text{true}}, b_i)\}$ .

Fix a small  $K > 0$ , and train the parameters by applying SGD to

$$\underset{\{\theta^k, W_1^k, W_2^k\}_{k=0}^K}{\text{minimize}} \sum_{(x^*, b) \in D} \|x^K(b) - x^*\|_2^2,$$

where  $x^K(b)$  is the  $K$ -layer output of the NN.

After the NN is trained with  $K = 16$ , the test performance is shockingly good:



The trained NN is called Learned ISTA (LISTA).

LISTA works much better than ISTA at any  $\lambda$  and using a theoretical stepsize.

The idea was quickly applied to other algorithms (ADMM, PDHG, etc.) and many applications:

- Image denoising/deblurring/super-resolution/segmentation Zhang and Ghanem [2018], Li et al. [2020], Wang et al. [2015], Zheng et al. [2015]
- Medical imaging Sun et al. [2016], Adler and Öktem [2018]
- Remote sensing Lohit et al. [2019]
- Wireless Communication Sun et al. [2017], Balatsoukas-Stimming and Studer [2019], He et al. [2020]

and beyond.

## Application: Super-Resolution

**Problem:** generate a high-resolution image from a low-resolution image.

**Classic:** Sparse coding. Yang et al. [2010] (compute a dictionary pair  $(D_x, D_y)$  by bi-level optimization.  $D_x$  is low-resolution dictionary,  $D_y$  is high-resolution. Recovery: image  $\rightarrow$  sparse coding  $\rightarrow$  recover with  $D_y$ )

**Unrolling:** Wang et al. [2015] (unroll sparse coding, train end-to-end)



(a) Classic (PSNR<sup>1</sup>: 30.29 dB)



(b) CNN Dong et al. [2014] (PSNR: 30.49 dB)



(c) Unrolling (PSNR: 30.86 dB)

Figure: The “butterfly” image upscaled by  $\times 4$  times using different methods.

<sup>1</sup>The PSNR is obtained on “Set 5” in BSD100 data set. The “butterfly” is in Set 5.

## Application: CT Reconstruction

**Problem:** Recover  $x$  from the observation  $b$ :

$$b = Ax + \text{noise},$$

where  $A$  is the Radon transform and the noise is Gaussian.

**Classic:** Total Variation (TV).

**Unrolling:** Adler and Öktem [2018]



(a) Classic (TV)



(b) CNN Jin et al. [2017]



(c) Unrolling

**Figure:** The “phantom” image recovered by different methods.

## Application: Image deblurring

**Problem:** recover image  $x$  from its blurry observation  $b$ :

$$b = k * x + \text{noise},$$

where  $k$  is an unknown blurring kernel and the noise is Gaussian.



(a) Total variation



(b) CNN Nah et al. [2017]



(c) Unrolling Li et al. [2020]

Figure: An image from BSD500 recovered by different methods.

## Challenges to address

- Too many parameters to train. Also how to choose  $K$ ?

$A \in \mathbb{R}^{m \times n}$  means  $\mathcal{O}(n^2K + mnK)$  parameters, not scalable to large  $m, n, K$

- Interpretability

Applications such as medical imaging and operations decisions require the algorithms to be explainable and reliable

- Safeguard for out-of-distribution problems

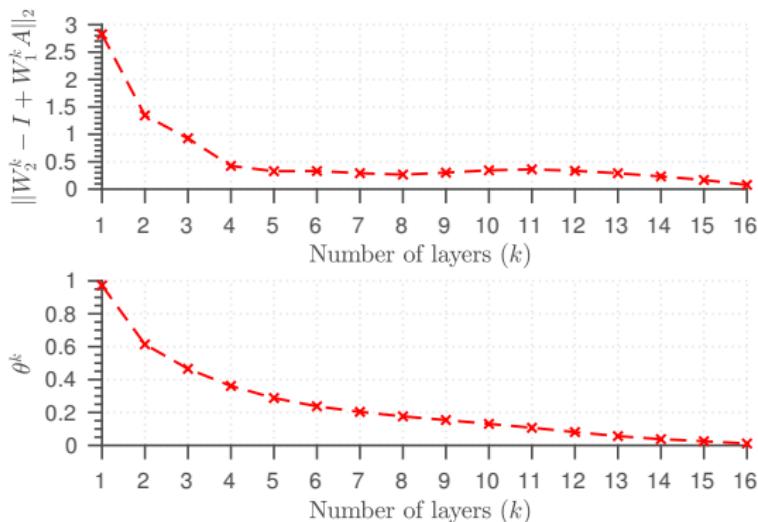
When applied to unseen data, the performance should be comparable to classic algorithms

## Reparameter reduction: coupling $W_1, W_2$

Assume no noise. If we need  $x^k \rightarrow x^{\text{true}}$  uniformly for all sparse signals, then simple calculation shows<sup>1</sup>:

- $W_2^k + W_1^k A \rightarrow I$ ,
- $\theta^k \rightarrow 0$ .

Indeed, training confirms the claims:



---

<sup>1</sup>Chen et al. [2018]

Therefore, we enforce

$$W_2^k = I_n - W_1^k A,$$

for all  $k$ , yielding the iteration:

$$x^{k+1} = \eta_{\theta^k} (x^k + W_1^k (b - Ax^k)).$$

We call it *weight coupling (CP)*.

Parameters

$$\mathcal{O}(n^2 K + mnK) \xrightarrow{\text{reduce}} \mathcal{O}(mnK),$$

significant reduction if  $m < n$  (which is often the case).

After this reduction, training also appears to be more stable.

## Support selection (SS)

Inspired by FPC (Hale, Y., Zhang'08) and Iterative Support Detection (Wang-Y.'09), at each iteration, let the largest few components *bypass soft-thresholding*.

If all bypassed nonzeros are true nonzeros, *soft-threshold induced bias* is reduced.

Control the number of bypassing components by *fraction*, a training parameter.

## Empirical results

We compare

- LISTA — original
- LISTA-CP — weight coupling
- LISTA-SS — support selection
- LISTA-CPSS — weight coupling & support detection

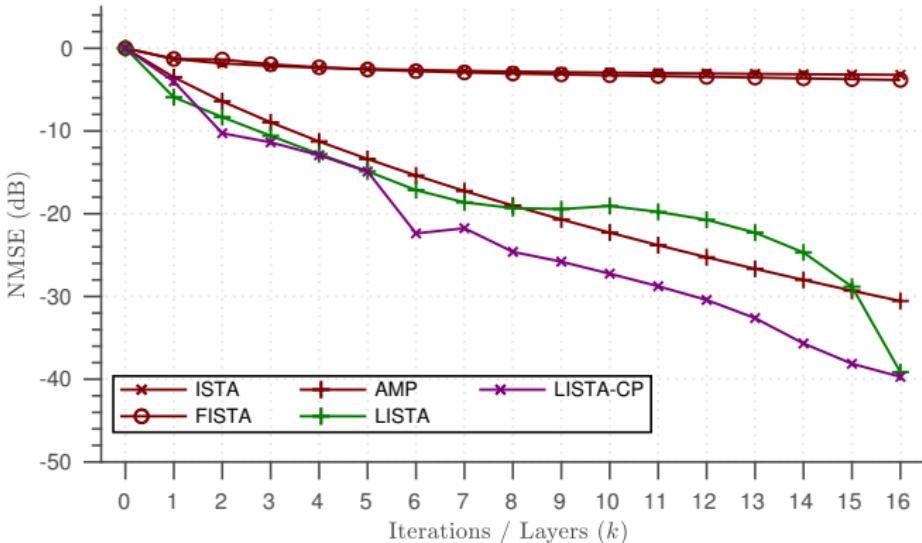
on normalized MSE (NMSE) in dB:

$$\text{NMSE}(\hat{x}, x^*) = 20 \log_{10} (\|\hat{x} - x^*\|_2 / \|x^*\|_2)$$

Tests:

- $m = 250$ ,  $n = 500$ , sparsity  $s \approx 50$ .
- $A_{ij} \sim \mathcal{N}(0, 1/\sqrt{m})$ , iid.  $A$  is column-normalized.
- Magnitudes were sampled from standard Gaussian.
- Measurement noise levels were measured by *signal-to-noise ratio*.

## Weight coupling (CP)

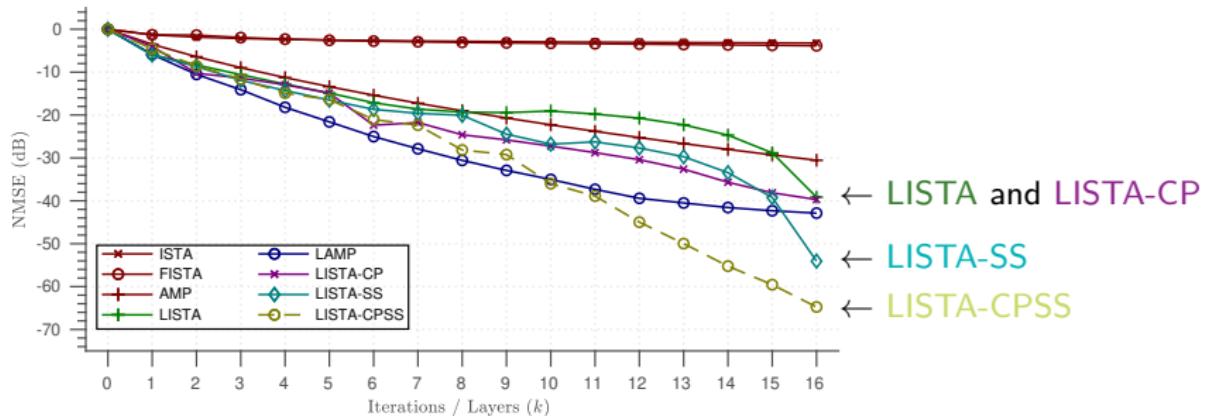


CP stabilizes intermediate results.

Same final recovery quality.

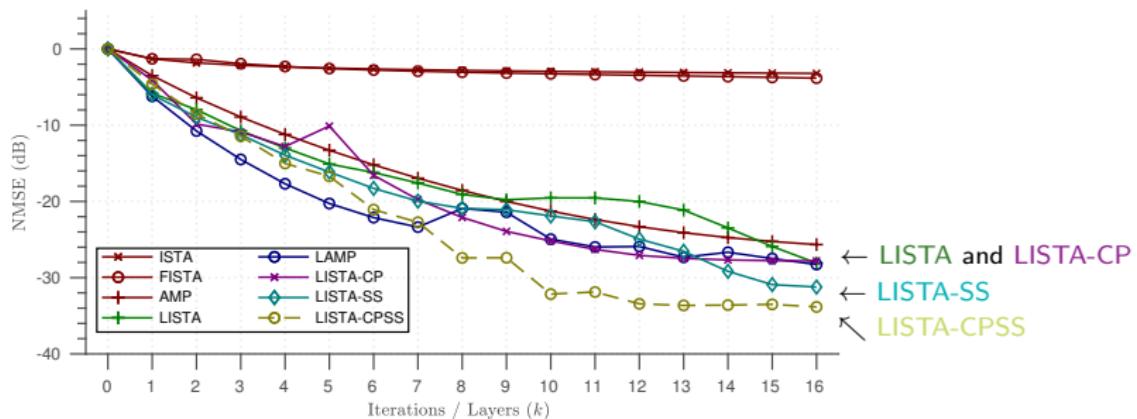
# Support selection (SS)

Noiseless case (SNR= $\infty$ )



# Support selection (SS)

Noisy case (SNR=30)



## Parameter reduction: tie $W_1$ across iterations

Inspired by analysis, let us try using the same  $W_1^k$  for all  $k$ . Write it as  $W$ .

→ Tied LISTA (TiLISTA) iteration:

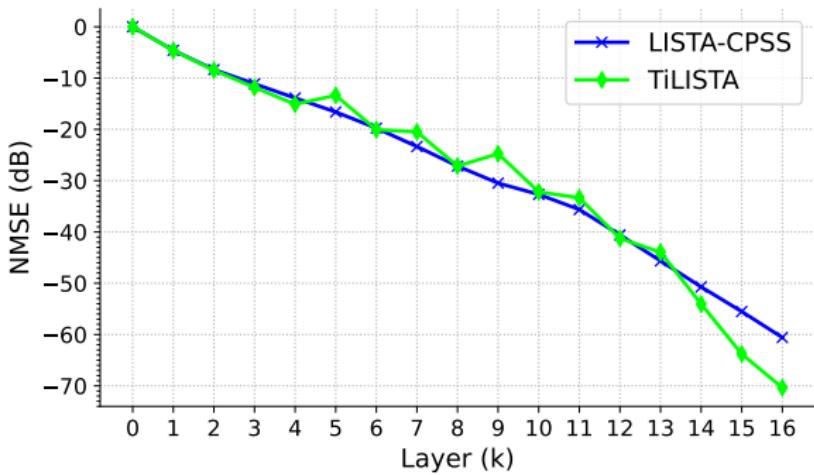
$$x^{k+1} = \eta_{\theta^k}(x^k - \gamma^k W^T(Ax^k - b)).$$

Parameters:

$$\mathcal{O}(mnK) \xrightarrow{\text{reduce}} \mathcal{O}(mn + K),$$

We learn only step sizes  $\{\gamma^k\}_k$  and thresholds  $\{\theta^k\}_k$ .

## TiLISTA Performance



TiLISTA works even slightly better than LISTA-CPSS

## Mutual Coherence

Coherence or mutual coherence [Donoho and Huo, 2001] of matrix  $A \in \mathbb{R}^{m \times n}$ , where columns  $a_i^\top a_i = 1$ , is

$$\max_{1 \leq i \neq j \leq n} |a_i^\top a_j|,$$

which is the max cross-correlation between pairs of columns.

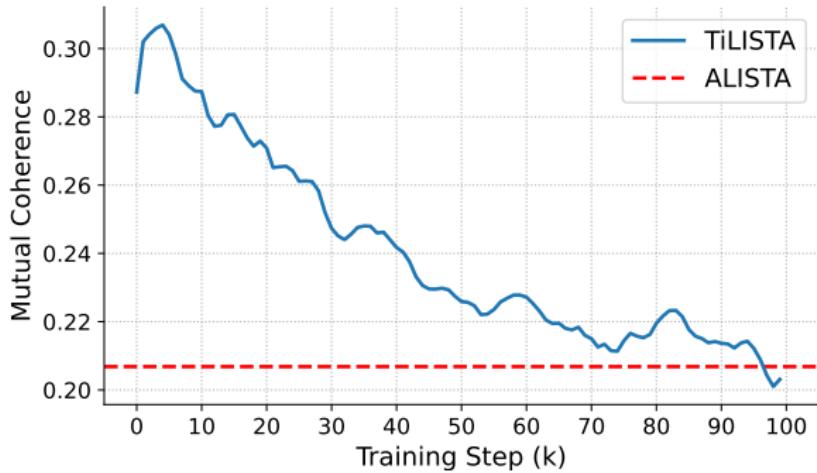
Smaller coherence of  $A$  tends to make sparse-signal recovery [Donoho and Elad, 2003].

Given  $A$  with columns  $a_i^\top a_i = 1$ , mutual coherence between matrices  $W$  and  $D$  is

$$\max_{1 \leq i \neq j \leq n} |w_i^\top a_j|$$

## Observation

We scale  $W$  such that  $w_i^\top a_i = 1$  for  $i = 1, \dots, n$  and then measure  $\max_{1 \leq i \neq j \leq n} |w_i^\top a_j|$  in TiLISTA.



Good  $W$  needs to have small mutual coherence to  $A$ .

## Analytic LISTA (ALISTA)

We use this principle to determine  $W$  *without training* [Liu and Chen, 2019] .

Two steps:

1. Compute approximately optimal  $\tilde{W}$ :

$$\tilde{W} \in \underset{W \in \mathbb{R}^{m \times n}}{\operatorname{argmin}} \|W^T A\|_F^2, \text{ s.t. } (W_{:,j})^T A_{:,j} = 1, \forall j = 1, 2, \dots, n,$$

which is a convex quadratic program (QP).

2. With  $\tilde{W}$  fixed, learn  $\{\gamma^k, \theta^k\}_k$  from data

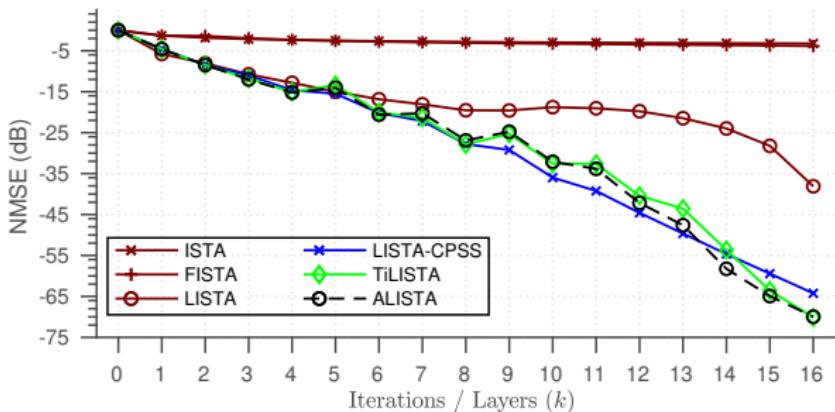
Parameters:

$$\mathcal{O}(mn + K) \xrightarrow{\text{reduce}} \mathcal{O}(K).$$

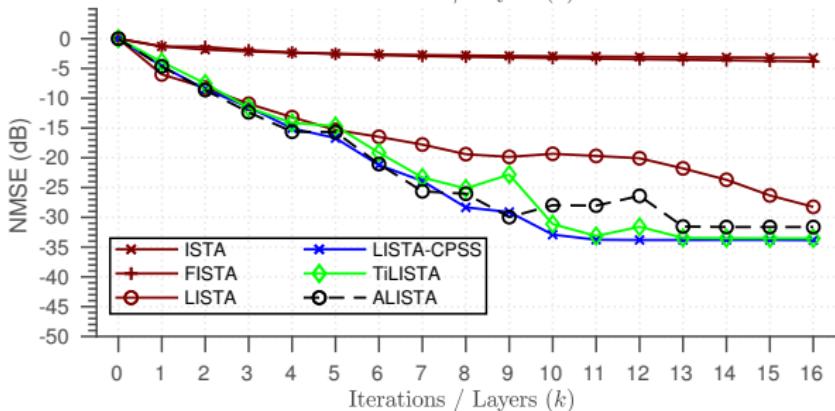
Training takes only minutes.

# Numerical evaluation

Noiseless case  
(SNR=∞)



Noisy case  
(SNR=30dB)



## Numbers of parameters to train

$K$ : number of layers.  $A$  has  $m$  rows and  $n$  columns.

	Parameters	Training Time	Performance
LISTA	$\mathcal{O}(Km^2 + Kmn)$	1.5 hours	LISTA
LISTA-CPSS	$\mathcal{O}(Kmn + K)$	50 minutes	$\ll$ LISTA-CPSS
TiLISTA	$\mathcal{O}(mn + K)$	20 minutes	$\approx$ TiLISTA
ALISTA	$\mathcal{O}(K)$	6 minutes	$\approx$ ALISTA

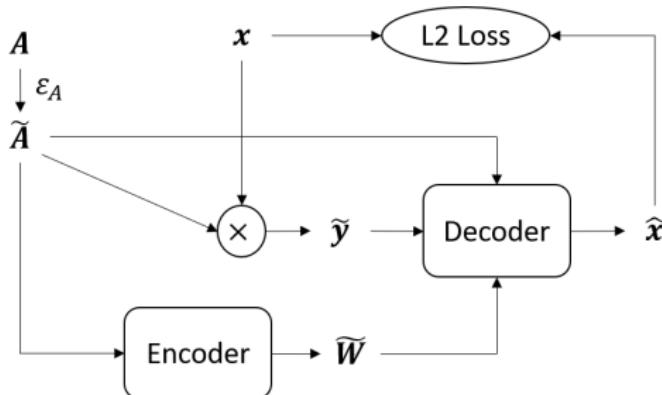
## Robust ALISTA

Consider  $\tilde{y} = \tilde{A}x + \varepsilon$  with  $\tilde{A} = A + \varepsilon_A$ . Given  $\tilde{A}$  and  $\tilde{y}$ , recover  $x$ . Must handle varying  $\tilde{A}$ .

Unroll an algorithm into an NN to generate  $\tilde{W}$  for  $\tilde{A}$ .

Method:

- train an NN (called *encoder*) with many pairs of  $(\tilde{A}, \tilde{W})$
- train an ALISTA (called *decoder*) with many  $(\tilde{A}, \tilde{y}, \tilde{W}, x)$
- jointly train them with many  $(\tilde{A}, \tilde{y}, \tilde{W}, x)$

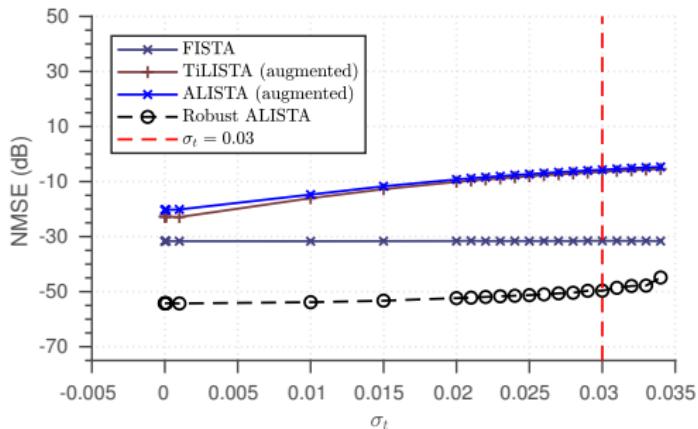


## Numerical results

Fix an  $A$ . Training:

- Non-robust LISTA methods used their  $W$  matrices obtained with  $A$ .
- Robust ALISTA trained with perturbed  $A$  (Gaussian  $\sigma = 0.03$ ).

Testing: All methods tested with perturbed  $A$ 's (Gaussian  $\sigma_1, \sigma_2, \dots \leq 0.03$ ).



Robust ALISTA is significantly more robust.

## Ada-LISTA [Aberdam et al., 2021]

Instead learning  $W$  and using it in

$$x^{k+1} = \eta_{\theta^k}(x^k - \gamma^k W^T(Ax^k - b)),$$

Ada-LISTA learns a *symmetric positive semidefinite*  $U$  and use it in

$$x^{k+1} = \eta_{\theta^k}(x^k - \gamma^k A^T U(Ax^k - b)).$$

This makes  $A^T U(Ax^k - b)$  a descent direction of  $\frac{1}{2} \|Ax - b\|_U^2$ , so we can use the latter as a loss function, train without the ground truth.

Motivated by FISTA, Ada-LISTA also adds momentum.

## LISTA Capacity Theory

ALISTA [Liu and Chen, 2019] proves: given low mutual coherence ( $A, W$ ) and any sparse, significant signal  $x$ ,  $\exists$  parameters such that ALISTA converges linearly.

The paper also proves a negative result: for any  $(W_1^k, W_2^k, \theta^k)$ , for sparse  $x$  with uniform-random supports and values, linear convergence is the best rate w.h.p.

Ada-LISTA [Aberdam et al., 2021] proves [robust linear convergence.]

Step-LISTA<sup>2</sup> provides the necessary condition that the model converges to the solution of LASSO.

**Generalization:** [Schnoor et al., 2021, Kouni, 2022, Joukovsky et al., 2021] analyzed the Rademacher complexity of LISTA and variants.

---

<sup>2</sup>[Ablin et al., 2019]

## HyperLISTA [Chen et al., 2021]

### Introduce

- a hybrid-thresholding operator to bypass  $p^k$  largest entries
- analytic formulas for the parameters
- three hyper-parameters subject to grid search

### Significance:

- allow the parameters to be “instance optimal”
- proves  $\exists$  parameters to obtain *superlinear* error reduction

HyperLISTA learns  $c_1, c_2, c_3 > 0$  and use them to set

$$\begin{aligned}\theta^k &= c_1 \mu \|A^\dagger(Ax^k - b)\|_1, && \text{soft threshold} \\ \beta^k &= c_2 \mu \|x^k\|_0, && \text{momentum stepsize} \\ p^k &= c_3 \min \left( \log \left( \frac{\|A^\dagger b\|_1}{\|A^\dagger(Ax^k - b)\|_1} \right), n \right), && \text{pass-through count}\end{aligned}$$

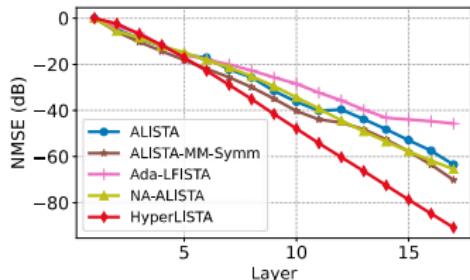
The formulas are motivated by the analysis but use  $x^k$  instead of  $x^{\text{true}}$ .

Parameters:

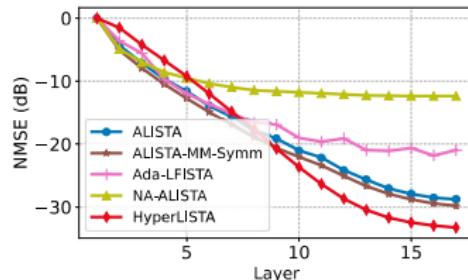
$$\mathcal{O}(K) \xrightarrow{\text{reduce}} 3.$$

Training can be done by grid search or a global optimization method.

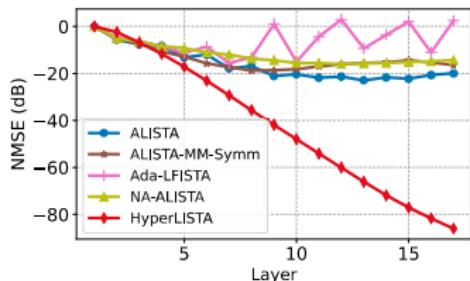
# HyperLISTA is fast and robust



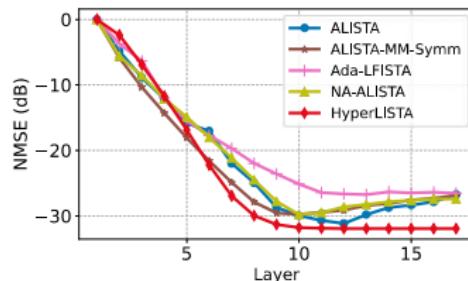
(a) Noiseless. No train/test mismatch.



(b) Sparsity ratio  $p$  changed to 0.15.



(c) Variance  $\sigma$  of non-zero elements changed to 2.



(d) Noise level changed to SNR=30dB.

Good analytic rules have better generalization perf.

## Uncovered LISTA topics

- [Moreau and Bruna, 2017] proposed to understand LISTA by the similarity between LISTA and a matrix-factorization method.
- [Xin et al., 2016] proposed learned iterative hard-thresholding-CP.
- [Wu et al., 2019] proposed gated mechanisms to improve LISTA.
- [Ito et al., 2019] proposed a minimum mean squared error (MMSE) estimator-based shrinkage function in LISTA.
- [Yang et al., 2020] proposed to use nonconvex-function-induced regularizers in LISTA.
- [Heaton et al., 2020] introduced a safeguard wrapper for LISTA methods applied to structured convex problems.
- When  $K$  is large or  $K = \infty$ , LISTA cannot be trained. Instead, we can use deep equilibrium[Bai et al., 2019, Winston and Kolter, 2020] and fixed-point network [Fung et al., 2022]. [Gilton et al., 2021] demonstrated better image recovery.

## Summary

There is still huge room for optimization speed to improve. Integrating optimization and ML is a viable approach.

AU integrates data-driven (slow/fast, adaptive) and analytic (fast/slow, universal) approaches to obtain **fast/fast** and **adaptive** algorithms.

Despite the success in sparse coding, much still needs to be advanced and understood for other AU applications.

Thank you!

## References:

- Aviad Aberdam, Alona Golts, and Michael Elad. Ada-LISTA: Learned solvers adaptive to varying models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.
- Pierre Ablin, Thomas Moreau, Mathurin Massias, and Alexandre Gramfort. Learning step sizes for unfolded sparse coding. *Advances in Neural Information Processing Systems*, 32, 2019.
- Jonas Adler and Ozan Öktem. Learned primal-dual reconstruction. *IEEE transactions on medical imaging*, 37(6):1322–1332, 2018.
- Shaojie Bai, J Zico Kolter, and Vladlen Koltun. Deep equilibrium models. *Advances in Neural Information Processing Systems*, 32, 2019.
- Alexios Balatsoukas-Stimming and Christoph Studer. Deep unfolding for communications systems: A survey and some new directions. In *2019 IEEE International Workshop on Signal Processing Systems (SiPS)*, pages 266–271. IEEE, 2019.
- Xiaohan Chen, Jialin Liu, Zhangyang Wang, and Wotao Yin. Theoretical linear convergence of unfolded ista and its practical weights and thresholds. *Advances in Neural Information Processing Systems*, 31, 2018.
- Xiaohan Chen, Jialin Liu, Zhangyang Wang, and Wotao Yin. Hyperparameter tuning is all you need for lista. *Advances in Neural Information Processing Systems*, 34: 11678–11689, 2021.

- Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *European conference on computer vision*, pages 184–199. Springer, 2014.
- David L Donoho and Michael Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via  $\ell_1$  minimization. *Proceedings of the National Academy of Sciences*, 100(5):2197–2202, 2003.
- David L Donoho and Xiaoming Huo. Uncertainty principles and ideal atomic decomposition. *IEEE transactions on information theory*, 47(7):2845–2862, 2001.
- Samy Wu Fung, Howard Heaton, Qiuwei Li, Daniel McKenzie, Stanley Osher, and Wotao Yin. Jfb: Jacobian-free backpropagation for implicit networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.
- Davis Gilton, Gregory Ongie, and Rebecca Willett. Deep equilibrium architectures for inverse problems in imaging. *IEEE Transactions on Computational Imaging*, 7: 1123–1133, 2021.
- Hengtao He, Chao-Kai Wen, Shi Jin, and Geoffrey Ye Li. Model-driven deep learning for mimo detection. *IEEE Transactions on Signal Processing*, 68:1702–1715, 2020.
- Howard Heaton, Xiaohan Chen, Zhangyang Wang, and Wotao Yin. Safeguarded learned convex optimization. *arXiv preprint arXiv:2003.01880*, 2020.
- Daisuke Ito, Satoshi Takabe, and Tadashi Wadayama. Trainable ista for sparse signal recovery. *IEEE Transactions on Signal Processing*, 67(12):3113–3125, 2019.

Kyong Hwan Jin, Michael T McCann, Emmanuel Froustey, and Michael Unser. Deep convolutional neural network for inverse problems in imaging. *IEEE Transactions on Image Processing*, 26(9):4509–4522, 2017.

Boris Joukovsky, Tanmoy Mukherjee, Huynh Van Luong, and Nikos Deligiannis. Generalization error bounds for deep unfolding rnns. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pages 1515–1524. PMLR, 2021.

Vasiliki Kouni. Generalization error bounds for deconet: a deep unfolded network for analysis compressive sensing. *arXiv preprint arXiv:2205.07050*, 2022.

Yuelong Li, Mohammad Tofghi, Junyi Geng, Vishal Monga, and Yonina C Eldar. Efficient and interpretable deep blind image deblurring via algorithm unrolling. *IEEE Transactions on Computational Imaging*, 6:666–681, 2020.

Jialin Liu and Xiaohan Chen. Alista: Analytic weights are as good as learned weights in lista. In *International Conference on Learning Representations (ICLR)*, 2019.

Suhas Lohit, Dehong Liu, Hassan Mansour, and Petros T Boufounos. Unrolled projected gradient descent for multi-spectral image fusion. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7725–7729. IEEE, 2019.

Thomas Moreau and Joan Bruna. Understanding trainable sparse coding via matrix factorization. In *5th International Conference on Learning Representations, ICLR 2017*, 2017.

Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3883–3891, 2017.

Ekkehard Schnoor, Arash Behboodi, and Holger Rauhut. Generalization error bounds for iterative recovery algorithms unfolded as neural networks. *arXiv preprint arXiv:2112.04364*, 2021.

Haoran Sun, Xiangyi Chen, Qingjiang Shi, Mingyi Hong, Xiao Fu, and Nikos D Sidiropoulos. Learning to optimize: Training deep neural networks for wireless resource management. In *2017 IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pages 1–6. IEEE, 2017.

Jian Sun, Huibin Li, Zongben Xu, et al. Deep admm-net for compressive sensing mri. *Advances in neural information processing systems*, 29, 2016.

Zhaowen Wang, Ding Liu, Jianchao Yang, Wei Han, and Thomas Huang. Deep networks for image super-resolution with sparse prior. In *Proceedings of the IEEE international conference on computer vision*, pages 370–378, 2015.

Ezra Winston and J Zico Kolter. Monotone operator equilibrium networks. *Advances in neural information processing systems*, 33:10718–10728, 2020.

Kailun Wu, Yiwen Guo, Ziang Li, and Changshui Zhang. Sparse coding with gated learned ista. In *International Conference on Learning Representations*, 2019.

Bo Xin, Yizhou Wang, Wen Gao, David Wipf, and Baoyuan Wang. Maximal sparsity with deep networks? *Advances in Neural Information Processing Systems*, 29, 2016.

- Chengzhu Yang, Yuantao Gu, Badong Chen, Hongbing Ma, and Hing Cheung So. Learning proximal operator methods for nonconvex sparse recovery with theoretical guarantee. *IEEE Transactions on Signal Processing*, 68:5244–5259, 2020.
- Jianchao Yang, John Wright, Thomas S Huang, and Yi Ma. Image super-resolution via sparse representation. *IEEE transactions on image processing*, 19(11):2861–2873, 2010.
- 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*, pages 1828–1837, 2018.
- Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE international conference on computer vision*, pages 1529–1537, 2015.