Sparse (Linear) Optimization

Academy of Mathematics and Systems Sciences, CAS

May 11, 2023

Outline

- Compressive Sensing and Sparse Optimization

  —— Basics

- AI for Sparse Optimization

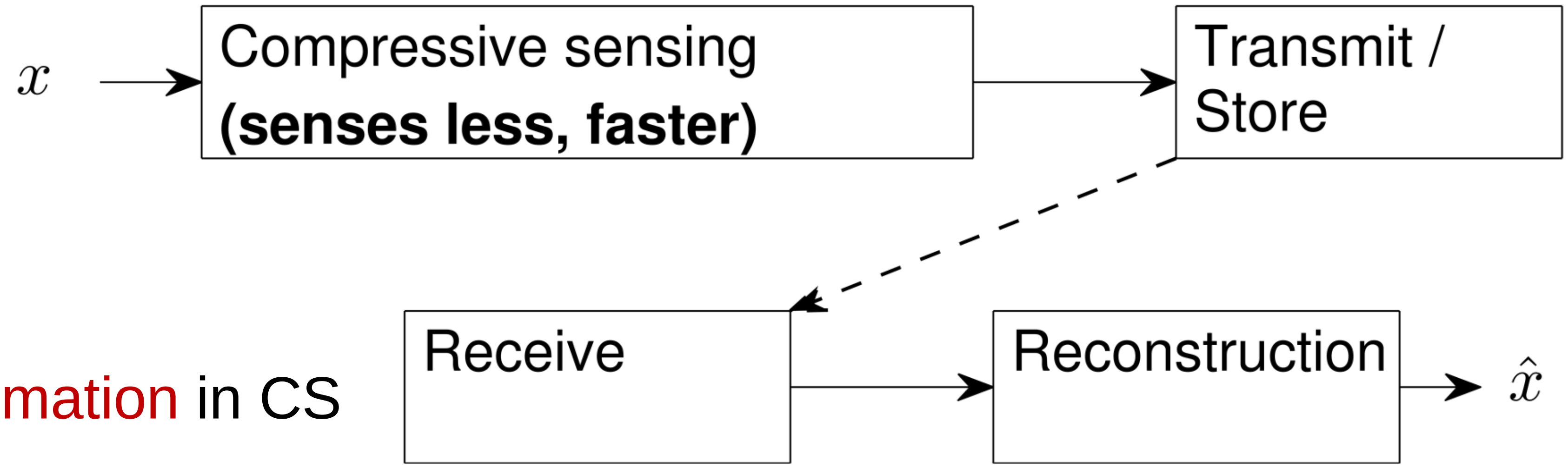
  —— End to End Fully Learned Approach

  —— Optimization Algorithm Inspired Network Design:

  Model-based Deep Learning

Background

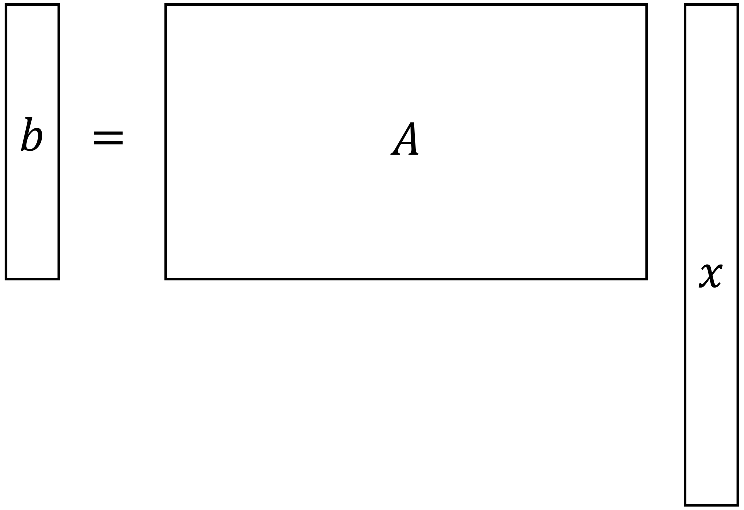
- Sparsity is a common concept that is widely used in various disciplines of science and engineering.
- Sparse signals can be efficiently compressed.
- Compressed sensing (CS) greatly enhances the acquisition and information processing capabilities by utilizing the sparsity of signals [Candes and Tao (2006), Donoho (2006)].
- Sparse optimization plays a key role in these works. Also, it is used in the field of linear inverse problems.

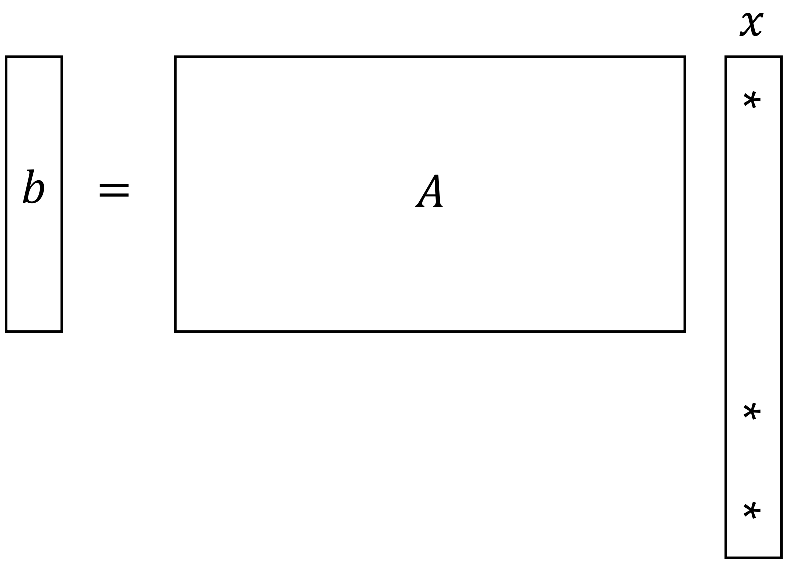


Information Transformation in CS

Underdetermined Systems of Linear Equation

$$x \in \mathbb{R}^n, A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$$



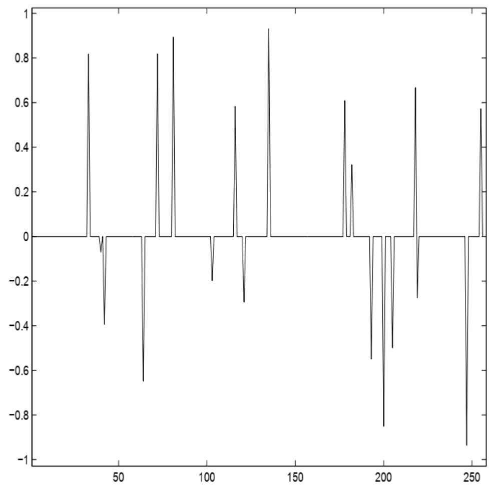


- When fewer equations than unknowns:
  - Fundamental theorem of algebra says that we cannot find unique $\boldsymbol{x}$.
- If unknown is assumed to be sparse, then one can often find unique solutions.
- Questions: How to find it?

From unknown structure to sparse structure

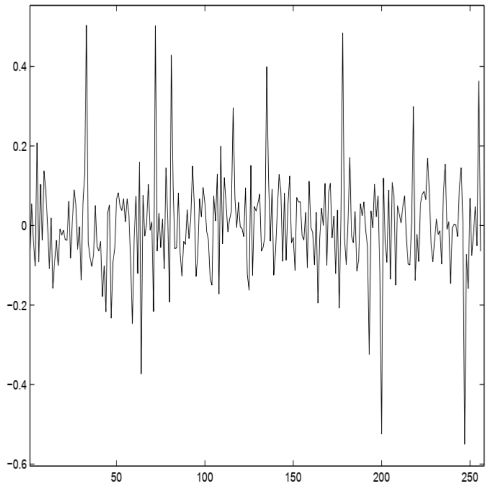
A Demo

- Given $n = 256$, $m = 128$.
- Generate a Gaussian random matrix A of $m \times n$; a n dimensional random sparse matrix with approximately $0.1 * n$ uniformly distributed non zero elements; $b = Au$;



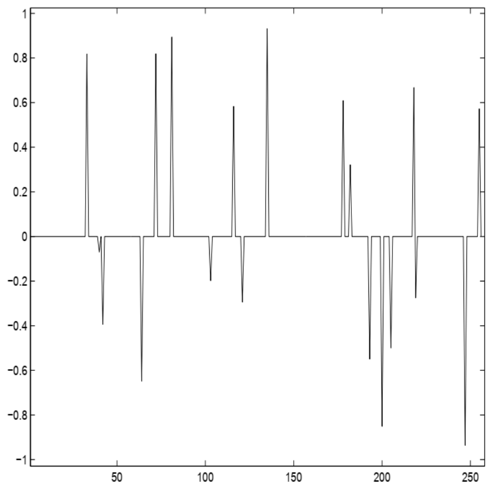
$$\begin{cases} \min_x \|x\|_0 \\ \text{s.t. } Ax = b \end{cases}$$

(a) ℓ_0-minimization



$$\begin{cases} \min_x \|x\|_2 \\ \text{s.t. } Ax = b \end{cases}$$

(b) ℓ_2-minimization



$$\begin{cases} \min_x \|x\|_1 \\ \text{s.t. } Ax = b \end{cases}$$

(c) ℓ_1-minimization

The left and right sides look the same!

Equivalent Transformation of Sparse Constraints

ℓ_0 minimization:

$$\begin{aligned} \min \quad & \|x\|_0 \\ \text{s.t.} \quad & Ax = b \end{aligned} \tag{1}$$

NP-hard
Natarajan (1995)]

Transform $\longrightarrow$

ℓ_1 minimization:

$$\begin{aligned} \min \quad & \|x\|_1 \\ \text{s.t.} \quad & Ax = b \end{aligned} \tag{2}$$

Linear Program
[Donoho (2004)]

Question:

- When do problem (1) and problem (2) have the same solution?
- If the original signal x^o is sufficiently sparse, then under certain conditions, x^o is the only solution to (2).

A Brief Note

- Sparse approximation (also known as sparse representation) theory deals with sparse solutions for systems of linear equations.

- Optimization problem with ℓ_1 norm regularization on the solution

$$\min_{x \in \mathbb{R}^n} \|x\|_1, \quad s.t. \quad y = Ax.$$

is equivalent to the linear programming

$$\min_{x,z \in \mathbb{R}^n} \sum_{i=1}^{n} z_i, \quad s.t. \quad y = Ax, \quad -z \leq x \leq z$$

- Some researchers also refer to the sparse optimization problem with ℓ_1 norm as sparse linear programming problem.

The Null Space Property of A

- Naturally, a necessary and sufficient condition for x^o to be the unique solution of (2) is $\|x^o + h\|_1 > \|x^o\|_1 , \quad \forall h \in \mathrm{Null}(A)\backslash\{0\}$

- Suppose that $\mathcal{S} := \{i \mid x_i^o \neq 0\} \quad \mathcal{S}^c := \{i \mid x_i^o = 0\}$

- Through simple deduction, we have

$$\begin{aligned} \|x^o + h\|_1 &= \|x^o_{\mathcal{S}} + h_{\mathcal{S}}\|_1 + \|0 + h_{\mathcal{S}^c}\|_1 \\ &= \|x^o\|_1 + \boxed{(\|h_{\mathcal{S}^c}\|_1 - \|h_{\mathcal{S}}\|_1)} + \underbrace{(\|x^o_{\mathcal{S}} + h_{\mathcal{S}}\|_1 - \|x^o_{\mathcal{S}}\|_1 + \|h_{\mathcal{S}}\|_1)}_{\text{Triangle inequality, } \geq 0} \end{aligned}$$

- So, the condition for $\|x^o + h\|_1 > \|x^o\|_1$ to hold true is that $\|h_{\mathcal{S}^c}\|_1 > \|h_{\mathcal{S}}\|_1$ is true.

The Null Space Property of A

- **Definition** (*k*-order null space property) $\forall h \in \mathrm{Null}(A)\backslash\{0\}$ satisfies $\|h_{S^c}\|_1 > \|h_S\|_1$ for all index sets S with $|S| \leqslant k$.

- **Theorem**[Donoho (2001)] $\min \|x\|_1$, $s.t. Ax = b$ uniquely recovers all *k*-sparse vectors x^0 from measurements $b = Ax^0$ if and only if A satisfies *k*-order null space property.

- (A more intuitive conditions) $\min \|x\|_1$, $s.t.$ $Ax = b$ recovers x uniquely if

$$\|x\|_0 < \min\left\{\frac{1}{4}\left(\frac{\|h\|_1}{\|h\|_2}\right)^2, \quad h \in \mathcal{N}(A)\backslash\{0\}\right\}$$

Requirements are placed on the sparsity of the signal!

Restricted Isometry Property (RIP)

- Definition **(Restricted isometry constants)** [Candes and Tao (2005)]

  For each $k = 1, 2, \ldots, \delta_k$ is the smallest scalar such that

  $$(1 - \delta_k)\|x\|_2^2 \leq \|Ax\|_2^2 \leq (1 + \delta_k)\|x\|_2^2$$

  for all k-sparse x.

- When δ_k is not too large, condition says that all $m \times k$ submatrices are well conditioned (sparse subsets of columns are not too far from orthonormal)

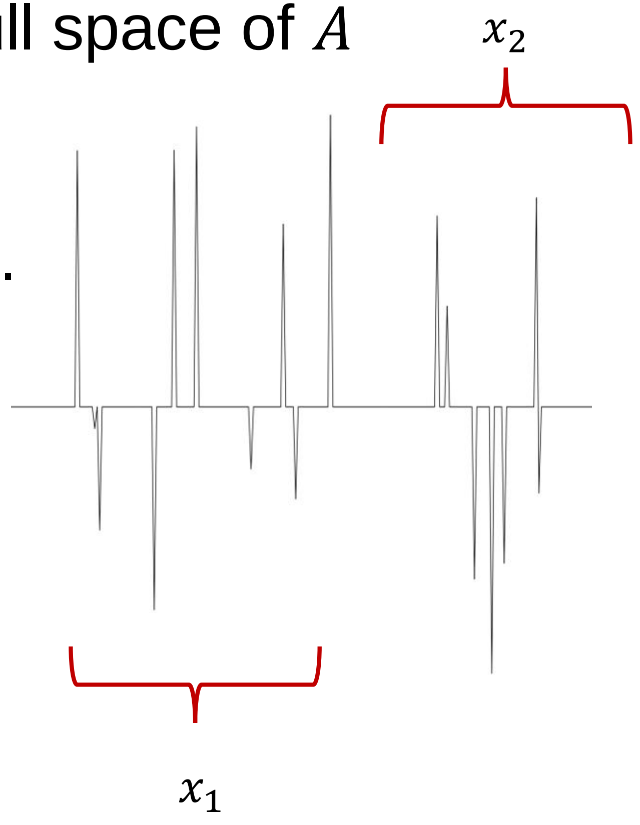
Restricted Isometry Property (RIP)

- x is k-sparse: $\|x\| \leq k$, can we recover all k-sparse vectors x from measurements $b = Ax^0$?

- Perhaps possible if sparse vectors lie away from null space of A

- Yes if any $2k$ columns of A are linearly independent.

- If x_1, x_2 k-sparse such that $Ax_1 = Ax_2 = b$
  $A(x_1 - x_2) = 0 \Rightarrow x_1 - x_2 = 0 \Leftrightarrow x_1 = x_2$



Restricted Isometry Property (RIP)

δ_{2k} is the smallest scalar such that

$$(1-\delta_{2k})\|x_1 - x_2\|_2^2 \leq \|Ax_1 - Ax_2\|_2^2 \leq (1+\delta_{2k})\|x_1 - x_2\|_2^2$$

for all k-sparse vectors x_1, x_2.

- Mo and Li (2011) prove that $\delta_{2k} < 0.493$ is sufficient to recover all k-sparse vectors x.
- Gaussian random matrices or other random matrices can satisfy the Restricted Isometry Property (RIP) with high probability when

$$m > O(k * \log(n/k))$$ [Zhang (2008)]

ℓ_1-regularized Least Square Problem

- Consider $\min \ \psi_\mu(x) := \mu\|x\|_1 + \frac{1}{2}\|Ax - b\|_2^2$

- Approaches:
  - Interior point methos: l1_ls
  - Spectral gradient method: GPSR
  - Fixed-point continuation method: FPC
  - Active set method: FPC_AS
  - Alternating direction augmented Lagrangian method: ADMM
  - Nesterov’s optimal first-order method
  - Iterative greedy algorithms

- Among the traditional sparse recovery algorithms, the ones that are greedy and iterative perform faster [Donoho (2009)].
- Each iteration in these greedy or iterative algorithms includes a matrix-vector multiplication which has the computational cost of $O(m * n)$.

Conclusion

$$\begin{array}{ll} \min & \|x\|_0 \\ \text{s.t.} & Ax = b \end{array} \qquad (1)$$

NP-hard

$\xrightarrow{\text{Transform}}$

ℓ_1 minimization:

$$\begin{array}{ll} \min & \|x\|_1 \\ \text{s.t.} & Ax = b \end{array} \qquad (2)$$

Linear Program

- Established the equivalent conditions for the mutual transformation of two problems (Null space property, RIP, etc.)
- Some classical convex optimization methods can be used to solve (2).

Outline

- Compressive Sensing and Sparse Optimization

  —— Basics

- AI for Sparse Optimization

  —— End to End Fully Learned Approach

  —— Optimization Algorithm Inspired Network Design:

  Model-based Deep Learning

A Deep Learning Approach to Compressed sensing

- Replace ℓ_0-norm in problem(1) with its convex relaxation ℓ_1-norm to convert (1) to a tractable and stable linear programming problem.

- **Question:** Can problem (1) be solved directly using neural networks? Will it be computationally faster?

- One important property of measurement matrix A that guarantees successful sparse signal recovery with very high probability is restricted isometry property (RIP).
- The main drawback of random measurements is that they are not optimally designed according to the signal under acquisition.

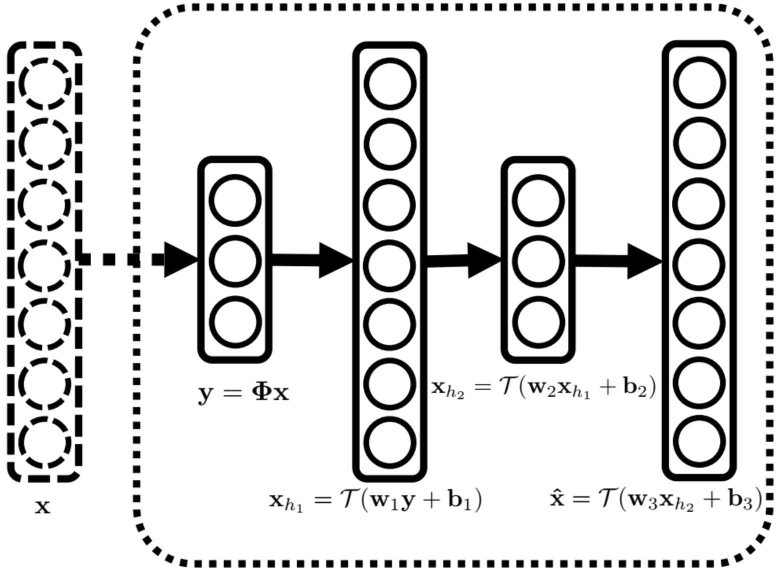
- **Question:** Can deep neural networks help us to adapt the measurements to the signal being under acquisition instead of taking random measurements and hence enhance the performance of the overall system?

SDA (Stacked Denoising Autoencoders) [Mousavi et al. (2015)]

- Consider the supervised learning framework: training set $\mathcal{D}_{\text{train}}$ has l pairs consisting of original signals and their corresponding measurements, i.e.,

$$\mathcal{D}_{\text{train}} = \{(\mathbf{y}^{(1)}, \mathbf{x}^{(1)}), (\mathbf{y}^{(2)}, \mathbf{x}^{(2)}), \ldots, (\mathbf{y}^{(l)}, \mathbf{x}^{(l)})\}$$

- Each layer of the SDA used for sparse recovery:
  - an input size of n (the ambient dimension of the original signal)
  - an output size of m (the dimension of the measurement vector)
  - or vice versa.



$$\mathbf{x}_{h_1} = \mathcal{T}(\mathbf{W}_1\mathbf{y} + \mathbf{b}_1)$$

$$\mathbf{x}_{h_2} = \mathcal{T}(\mathbf{W}_2\mathbf{x}_{h_1} + \mathbf{b}_2)$$

$$\hat{\mathbf{x}} = \mathcal{T}(\mathbf{W}_3\mathbf{x}_{h_2} + \mathbf{b}_3)$$

Loss Function:

$$\mathcal{L}(\Omega_{\mathrm{L}}) = \frac{1}{l}\sum_{i=1}^{l} \|\mathcal{M}_{\mathrm{L}}(\mathbf{y}^{(i)}, \Omega_{\mathrm{L}}) - \mathbf{x}^{(i)}\|_2^2.$$

SDA + Nonlinear Measurement

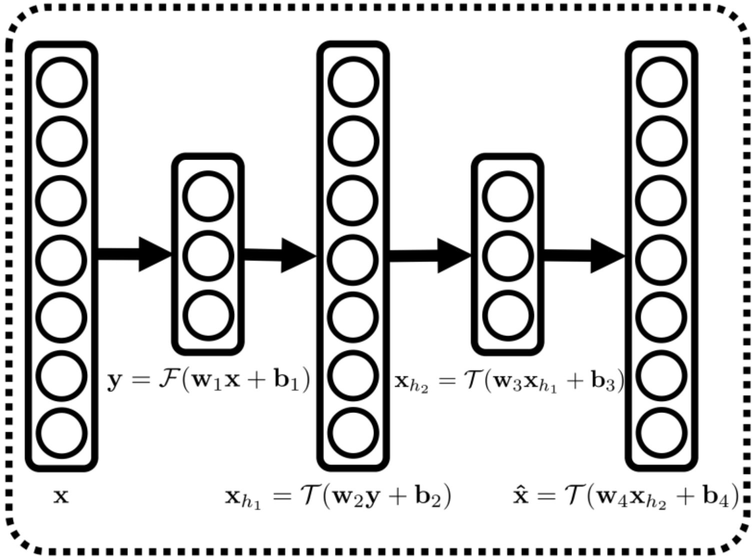
- The structure of SDA for nonlinear measurement paradigm is almost the same as the one before.
- The only difference: consider the mapping from original signal to its measurement vector as one layer of the SDA.
- This extra layer will let SDA adapt its structure to the training set $\mathcal{D}_{\text{train}}$

- Denote this extra layer that is the first layer of the SDA by

$$\mathbf{y} = \mathcal{F}(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1)$$

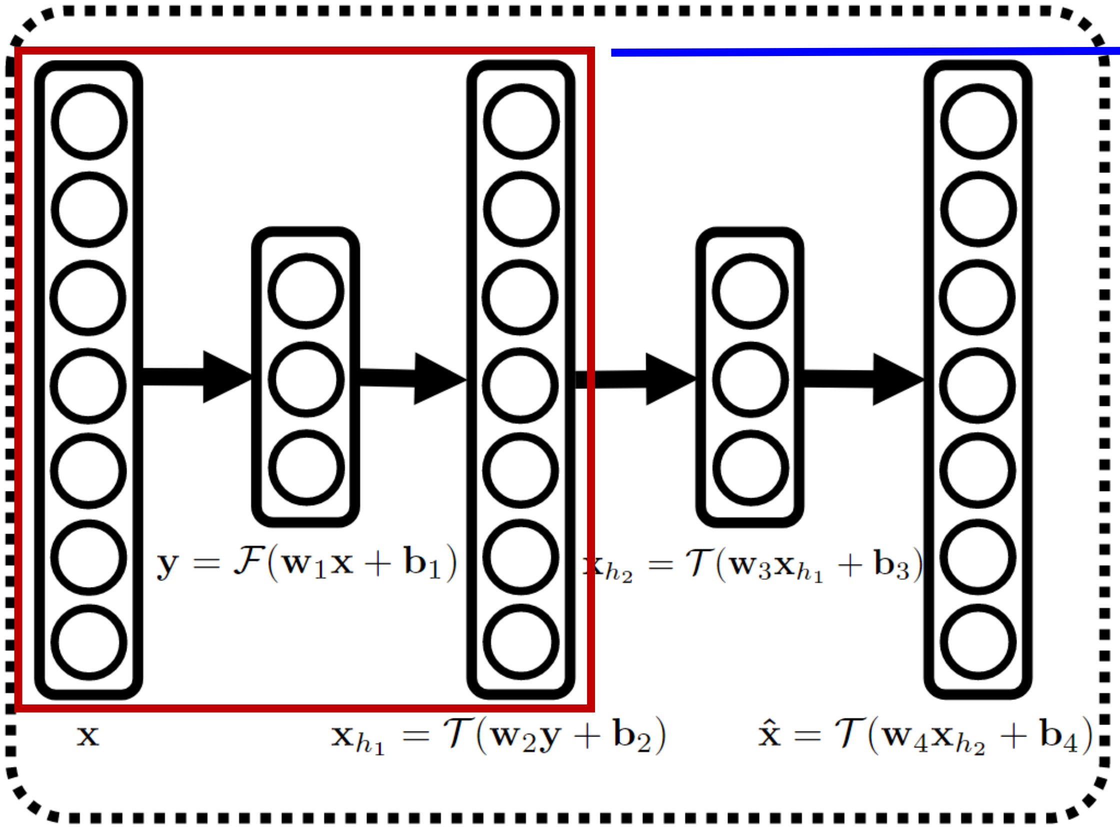
- The Loss function is also with some minor changes

$$\mathcal{L}(\Omega_{\text{NL}}) = \frac{1}{l}\sum_{i=1}^{l} \|\mathcal{M}_{\text{NL}}(\mathbf{x}^{(i)}, \Omega_{\text{NL}}) - \mathbf{x}^{(i)}\|_2^2.$$



Unsupervised Pre-training of SDA

- In the stacked version of denoising autoencoders, the unsupervised pre-training phase is done one layer at a time.



- Minimizing the error in reconstructing its input
- Compute the corresponding latent representation of the first t-layers and use it as an input in order to train the $t+1$-th layer.

SDA (Stacked Denoising Autoencoders) [Mousavi et al. (2015)]

- Traditional optimization problem vs Deep learning approach

- Similarity: We have the measurement vector (compressed data), we know the original signal model (*k*-sparse), and the goal is to retrieve the original signal from the compressed measurements.

- Difference:
  - For traditional optimization problems, we need an optimization algorithm to retrieve the signal from its measurements.
  - In deep networks, we pass the compressed data into a trained feedforward network without any need to solve an optimization problem.
  - Deep neural networks help us to adapt the measurements to the signal being under acquisition.

ReconNet [Kulkarni et al. (2016)]

- From FCN to CNN

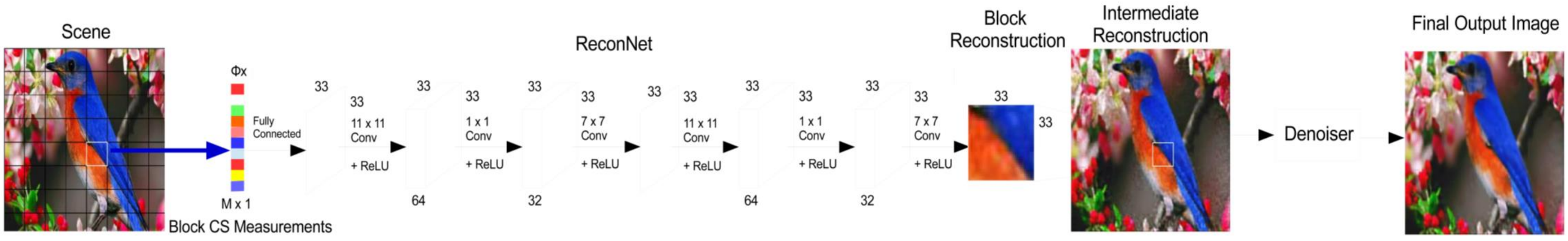


Figure 2: Overview of our non-iterative block CS image recovery algorithm.

- Note:
  - The input is an m-dimensional vector of compressive measurements.
  - The first layer is a fully connected layer that takes compressive measurements as input and outputs a feature map of size 33 × 33.

Outline

- Compressive Sensing and Sparse Optimization
  —— Basics

- AI for Sparse Optimization
  —— End to End Fully Learned Approach
  —— Optimization Algorithm Inspired Network Design:
  Model-based Deep Learning

Network Design based on Optimization Methods: An Example

- Compressive Sensing (CS) is an effective approach for fast Magnetic Resonance Imaging (MRI).
- Yang et al. (2019) proposed a novel deep architecture—ADMM-Net.
- ADMM-Net is defined over a data flow graph, which is derived from the iterative procedures in Alternating Direction Method of Multipliers (ADMM) algorithm for optimizing a CS-based MRI model.
- ADMM-Net significantly improves the baseline ADMM algorithm and achieves high reconstruction accuracies with fast computational speed.

ADMM-Net [Yang et al. (2019)]

- Assume $x \in \mathbb{C}^N$ is an MRI image to be reconstructed, $y \in \mathbb{C}^{N'} (N' < N)$ is the under-sampled *k*-space data, according to the CS theory.

- The reconstructed image can be estimated by solving the optimization problem

$$\hat{x} = \arg\min_x \left\{ \frac{1}{2} \|Ax - y\|_2^2 + \sum_{l=1}^{L} \lambda_l g(D_l x) \right\}$$

- where $A = PF \in \mathbb{R}^{N' \times N}$ is a measurement matrix, $P \in \mathbb{R}^{N' \times N}$ is under-sampling matrix, and F is a Fourier transform, D_l denotes a transform matrix for a filtering operation, λ_l is a regularization parameter.

- The optimization problem can be solved efficiently using ADMM algorithm [Boyd et al. (2011)]

ADMM-Net [Yang et al. (2019)]

- By introducing auxiliary variables $z = \{z_1, z_2, \cdots, z_L\}$, the equation is equivalent to:

$$\min_{x,z} \frac{1}{2}\|Ax - y\|_2^2 + \sum_{l=1}^{L} \lambda_l g(z_l) \qquad s.t. \;\; z_l = D_l x, \;\; \forall\, l \in [1, 2, \cdots, L]$$

- Its augmented Lagrangian function is:

$$\mathfrak{L}_\rho(x, z, \alpha) = \frac{1}{2}\|Ax - y\|_2^2 + \sum_{l=1}^{L} \lambda_l g(z_l) - \sum_{l=1}^{L} \langle \alpha_l, z_l - D_l x \rangle + \sum_{l=1}^{L} \frac{\rho_l}{2} \|z_l - D_l x\|_2^2,$$

- $\alpha = \{\alpha_l\}$ are Lagrangian multipliers and $\rho = \{\rho_l\}$ are penalty parameters. ADMM alternatively optimizes $\{x, z, \alpha\}$ by solving the following three subproblems:

$$\begin{cases} x^{(n+1)} = \arg\min\limits_{x} \frac{1}{2}\|Ax - y\|_2^2 - \sum_{l=1}^{L} \langle \alpha_l^{(n)}, z_l^{(n)} - D_l x \rangle + \sum_{l=1}^{L} \frac{\rho_l}{2} \|z_l^{(n)} - D_l x\|_2^2, \\ z^{(n+1)} = \arg\min\limits_{z} \sum_{l=1}^{L} \lambda_l g(z_l) - \sum_{l=1}^{L} \langle \alpha_l^{(n)}, z_l - D_l x^{(n+1)} \rangle + \sum_{l=1}^{L} \frac{\rho_l}{2} \|z_l - D_l x^{(n+1)}\|_2^2, \\ \alpha^{(n+1)} = \arg\min \sum_{l=1}^{L} \langle \alpha_l, D_l x^{(n+1)} - z_l^{(n+1)} \rangle, \end{cases}$$

- For simplicity, let $\beta_l = \frac{\alpha_l}{\rho_l} \; (l \in [1, 2, \cdots, L])$ and substitute $A = PF$

ADMM-Net [Yang *et al.* (2019)]

- Then the three subproblems have the following solutions:

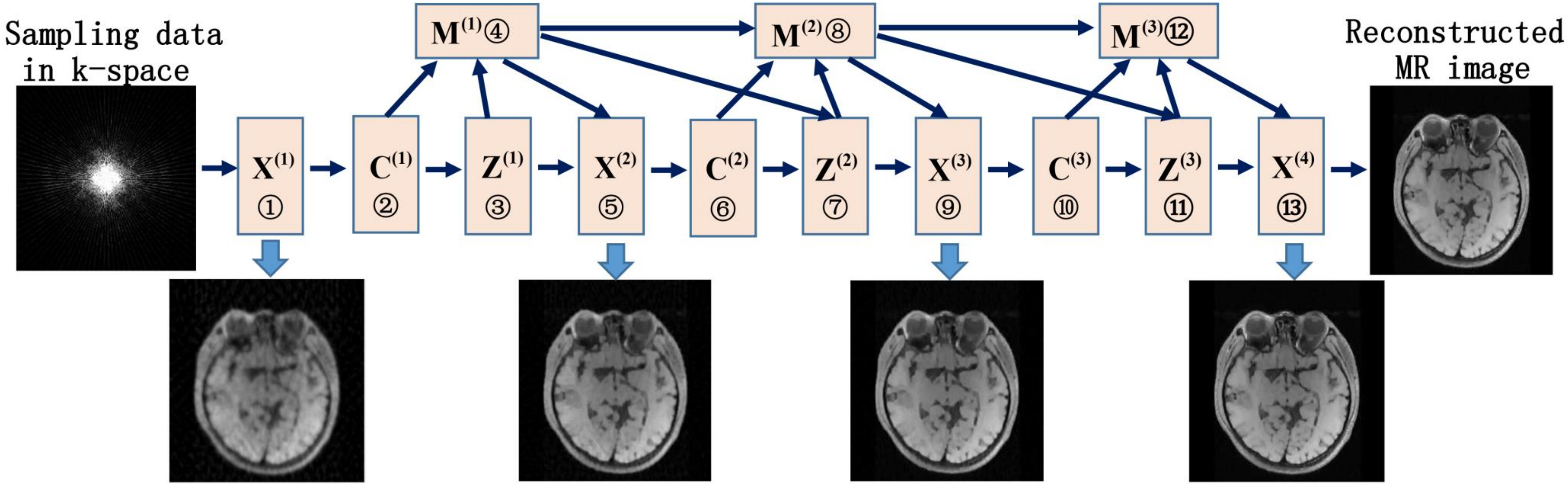
$$\begin{cases} \mathbf{X^{(n)}} : x^{(n)} = F^T[P^TP + \sum_{l=1}^{L} \rho_l F D_l^T D_l F^T]^{-1}[P^Ty + \sum_{l=1}^{L} \rho_l F D_l^T (z_l^{(n-1)} - \beta_l^{(n-1)})], \\ \mathbf{Z^{(n)}} : z_l^{(n)} = S(D_l x^{(n)} + \beta_l^{(n-1)}; \lambda_l/\rho_l), \\ \mathbf{M^{(n)}} : \beta_l^{(n)} = \beta_l^{(n-1)} + \eta_l(D_l x^{(n)} - z_l^{(n)}), \end{cases}$$

- **Note**: S(·) is a nonlinear shrinkage function [Bach *et al.* (2011)].

- **Problem:**
  - It is challenging to choose the transform D_l and shrinkage function S(·) for general regularization function g(·).
  - It is not trivial to tune the parameters ρ_l and η_l for k-space data with different sampling ratios.

ADMM-Net [Yang et al. (2019)]

- First map the ADMM iterative procedures to a data flow graph [Kavi et al. (1986)].



- There are four types of nodes mapped from four types of operations in ADMM-Net:
  - Reconstruction operation $(\mathbf{X}^{(\mathbf{n})})$
  - Convolution operation $(\mathbf{C}^{(\mathbf{n})})$
  - Nonlinear transform operation $(\mathbf{Z}^{(\mathbf{n})})$
  - Multiplier update operation $(\mathbf{M}^{(\mathbf{n})})$

Loss Function:

$$E(\Theta) = \frac{1}{|\Gamma|} \sum_{(y, x^{gt}) \in \Gamma} \frac{\sqrt{\|\hat{x}(y, \Theta) - x^{gt}\|_2^2}}{\sqrt{\|x^{gt}\|_2^2}},$$

ADMM-Net [Yang et al. (2019)]

- In the deep architecture, we aim to learn the following parameters:
  - $H_l^{(n)}$ and $\rho_l^{(n)}$ in reconstruction layer
  - filters $D_l^{(n)}$ in convolution layer
  - $\{q_{l,i}^{(n)}\}_{i=1}^{N_c}$ in nonlinear transform layer
  - $\eta_l^{(n)}$ in multiplier update layer
- Compared with traditional methods (ADMM solver), it tunes preset or tuned parameters become learnable parameters.
- It is a novel deep architecture defined over a data flow graph determined by an ADMM algorithm.
- Due to its flexibility in parameter learning, this deep net achieved high reconstruction accuracy while keeping the computational efficiency of the ADMM algorithm.

ADMM-Net [Yang et al. (2019)]

Table 1: Performance comparisons on brain data with different sampling ratios.

Method	20%		30%		40%		50%		Test time
	NMSE	PSNR	NMSE	PSNR	NMSE	PSNR	NMSE	PSNR	
Zero-filling	0.1700	29.96	0.1247	32.59	0.0968	34.76	0.0770	36.73	0.0013s
TV [2]	0.0929	35.20	0.0673	37.99	0.0534	40.00	0.0440	41.69	0.7391s
RecPF [4]	0.0917	35.32	0.0668	38.06	0.0533	40.03	0.0440	41.71	0.3105s
SIDWT	0.0885	35.66	0.0620	38.72	0.0484	40.88	0.0393	42.67	7.8637s
PBDW [6]	0.0814	36.34	0.0627	38.64	0.0518	40.31	0.0437	41.81	35.3637s
PANO [10]	0.0800	36.52	0.0592	39.13	0.0477	41.01	0.0390	42.76	53.4776s
FDLCP [8]	0.0759	36.95	0.0592	39.13	0.0500	40.62	0.0428	42.00	52.2220s
BM3D-MRI [11]	0.0674	37.98	0.0515	40.33	0.0426	41.99	0.0359	43.47	40.9114s
Init-Net_{13}	0.1394	31.58	0.1225	32.71	0.1128	33.44	0.1066	33.95	0.6914s
ADMM-Net_{13}	0.0752	37.01	0.0553	39.70	0.0456	41.37	0.0395	42.62	0.6964s
ADMM-Net_{14}	0.0742	37.13	0.0548	39.78	0.0448	41.54	0.0380	42.99	0.7400s
ADMM-Net_{15}	0.0739	37.17	0.0544	39.84	0.0447	41.56	0.0379	43.00	0.7911s

- It is the best considering the reconstruction accuracy and running time.

FISTA-Net [Xiang et al. (2021)]

- FISTA-Net is also a deep architecture by unrolling FISTA solver [Beck et al. (2009)] into iterative steps.

- FISTA Solver:
  - Objective function: $\hat{\mathbf{x}} = \operatorname*{argmin}_{\mathbf{x}} \left\{ \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \mu \|\mathbf{x}\|_1 \right\}$

Iteration process:

Put $y^{(1)} = x$, then

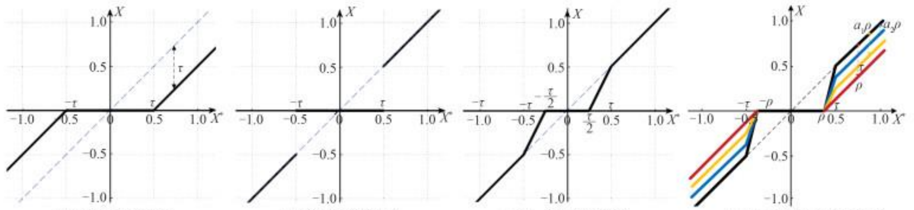
$$\mathbf{x}^{(k)} = \mathcal{T}_\alpha \left(\mathbf{y}^{(k)} - \mu \mathbf{A}^T \left(\mathbf{A}\mathbf{y}^{(k)} - \mathbf{b} \right) \right)$$

$$t^{(k+1)} = \frac{1 + \sqrt{1 + 4 \left(t^{(k)} \right)^2}}{2}$$

$$\mathbf{y}^{(k+1)} = \mathbf{x}^{(k)} + \left(\frac{t^{(k)} - 1}{t^{(k+1)}} \right) \left(\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)} \right)$$

Note:

- $\mathcal{T}_\alpha$ is the iterative shrinkage operator.



(a) 软阈值收缩算子 (b) 硬阈值收缩算子 (c) Firm阈值收缩算子 (d) 不同衰减特性的自适应阈值收缩算子

FISTA-Net [Xiang *et al.* (2021)]

- Network Mapping of FISTA:

$$\mathbf{r}^{(k)} = \mathbf{y}^{(k)} - \left(\mathbf{W}^{(k)}\right)^T \left(\mathbf{A}\mathbf{y}^{(k)} - \mathbf{b}\right)$$

$$\mathbf{x}^{(k)} = \mathcal{T}_{\theta^{(k)}}\left(\mathbf{r}^{(k)}\right)$$

$$\mathbf{y}^{(k+1)} = \mathbf{x}^{(k)} + \rho^{(k)}\left(\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\right)$$

- Gradient descent module $\mathbf{r}^{(k)}$

Liu *et al.* (2018) showed that $\mathbf{W}^{(k)}$ can be decomposed as the product as the product of a scalar $\mu^{(k)}$ and a matrix $\tilde{\mathbf{W}}$: $\mathbf{W}^{(k)} = \mu^{(k)}\tilde{\mathbf{W}}$
$\tilde{\mathbf{W}}$ has small coherence with $\mathbf{A}$.

- $\mathbf{r}^{(k)}, \mathbf{y}^{(k)}$ and $\mathbf{x}^{(k)}$ are intermediate variables;
- $\mathbf{W}^{(k)}$ is the gradient operator;
- $\mathcal{T}_{\theta^{(k)}}$ denotes the nonlinear proximal operator;
- $\rho^{(k)}$ denotes the scalar for momentum update.

- $\tilde{\mathbf{W}}$ is precomputed by solving:

$$\tilde{\mathbf{W}} \in \underset{\mathbf{W}\in\mathbb{R}^{N\times M}}{\arg\min} \left\|\mathbf{W}^T\mathbf{A}\right\|_F^2$$

$$\text{s.t.} \quad \left(\mathbf{W}_{:,m}\right)^T \mathbf{A}_{:,m} = 1, \forall m = 1, 2, \cdots, M$$

A standard convex quadratic program

FISTA-Net [Xiang *et al.* (2021)]

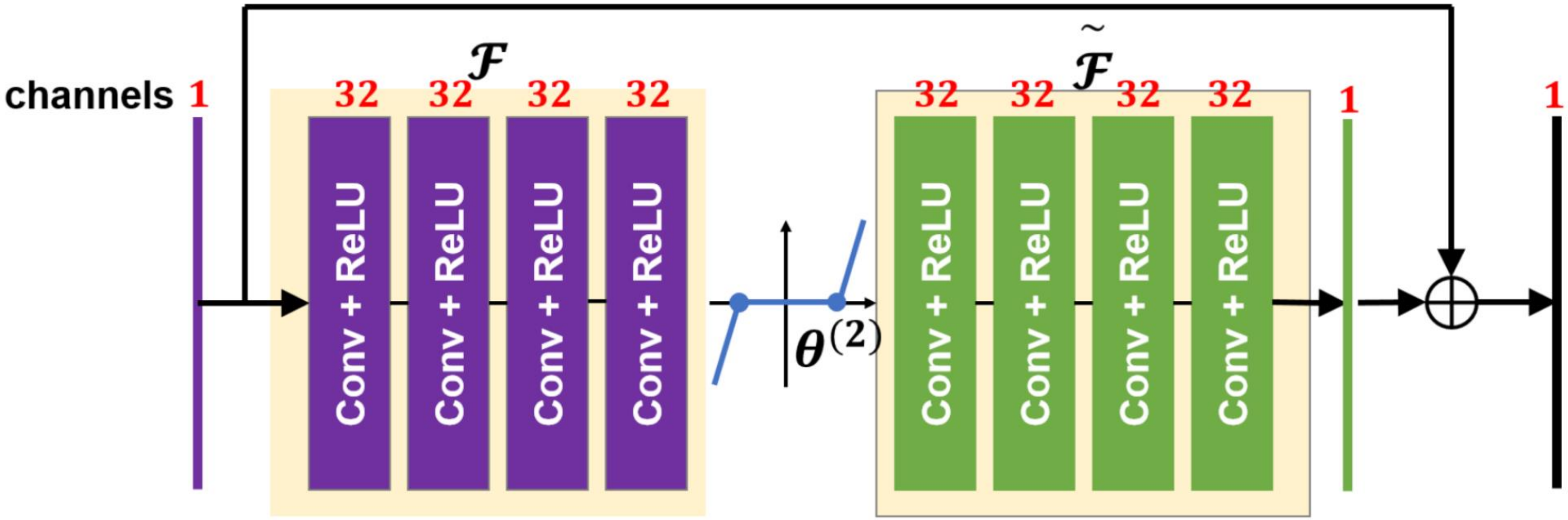
- Proximal mapping module $\mathbf{x}^{(k)}$.

$$\mathbf{r}^{(k)} = \mathbf{y}^{(k)} - \left(\mathbf{W}^{(k)}\right)^{T}\left(\mathbf{A}\mathbf{y}^{(k)} - \mathbf{b}\right)$$

$$\mathbf{x}^{(k)} = \mathcal{T}_{\theta^{(k)}}\left(\mathbf{r}^{(k)}\right)$$

$$\mathbf{y}^{(k+1)} = \mathbf{x}^{(k)} + \rho^{(k)}\left(\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\right)$$

- FISTA-Net aims to learn a more flexible representation $\mathcal{T}$



FISTA-Net [Xiang *et al.* (2021)]

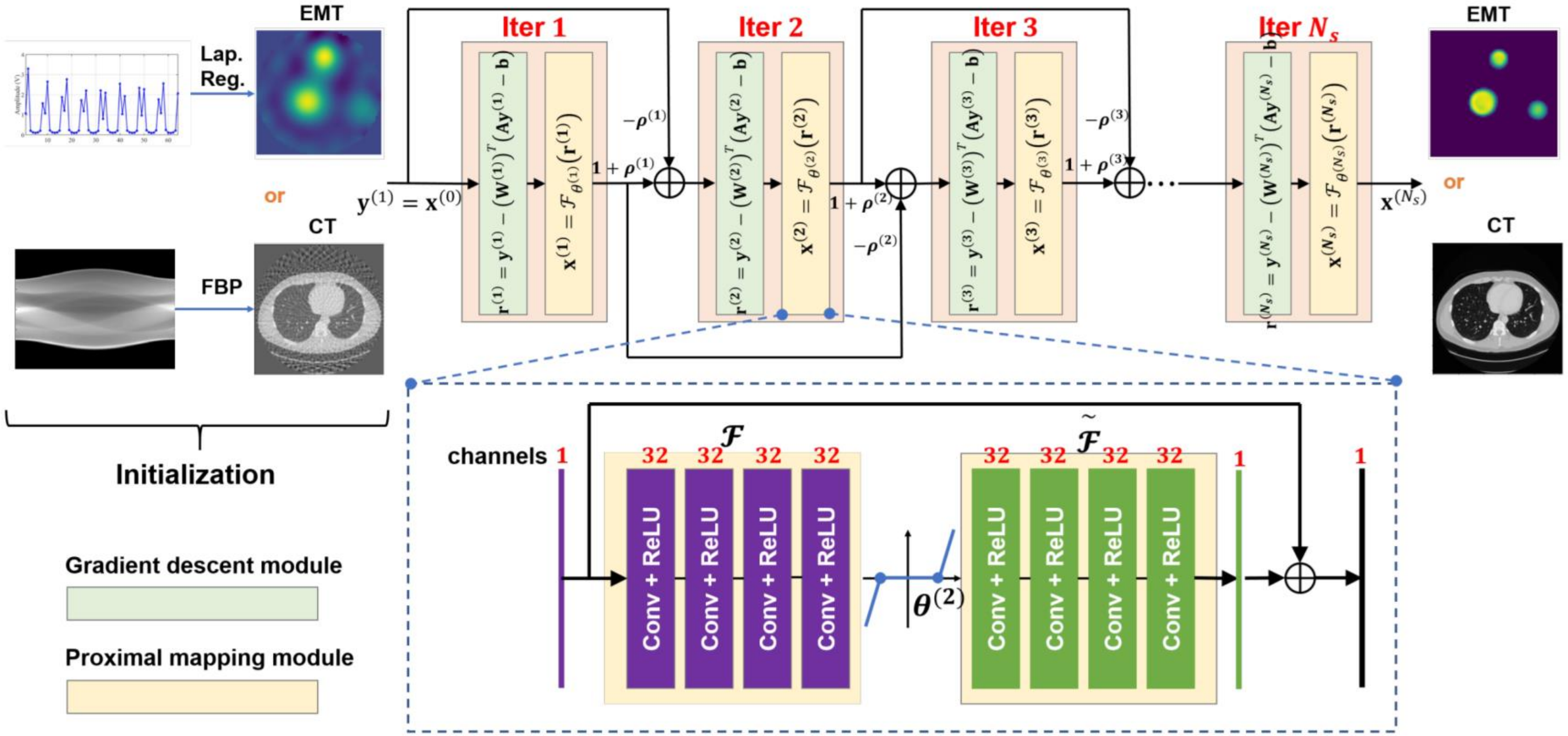


Fig. 2. The overall architecture of the proposed FISTA-Net with N_s iterations. In specific, FISTA-Net consists of three main modules, i.e. gradient descent, proximal mapping and two-step update.

Key:

1. The smooth differentiable part using the gradient information.

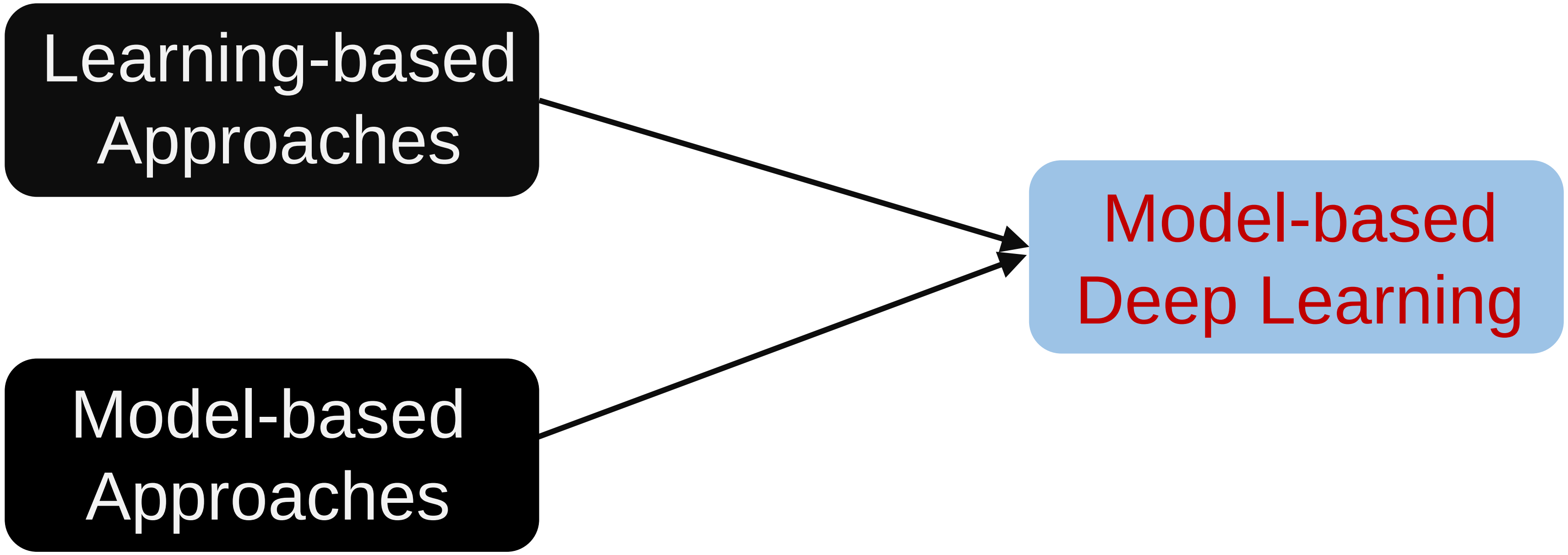
2. The non-differentiable part using a operator represented by a learned network.

Loss Function: $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{mse}} + \lambda_1 \mathcal{L}_{\text{sym}} + \lambda_2 \mathcal{L}_{\text{spa}}$

FISTA-Net [Xiang *et al.* (2021)]

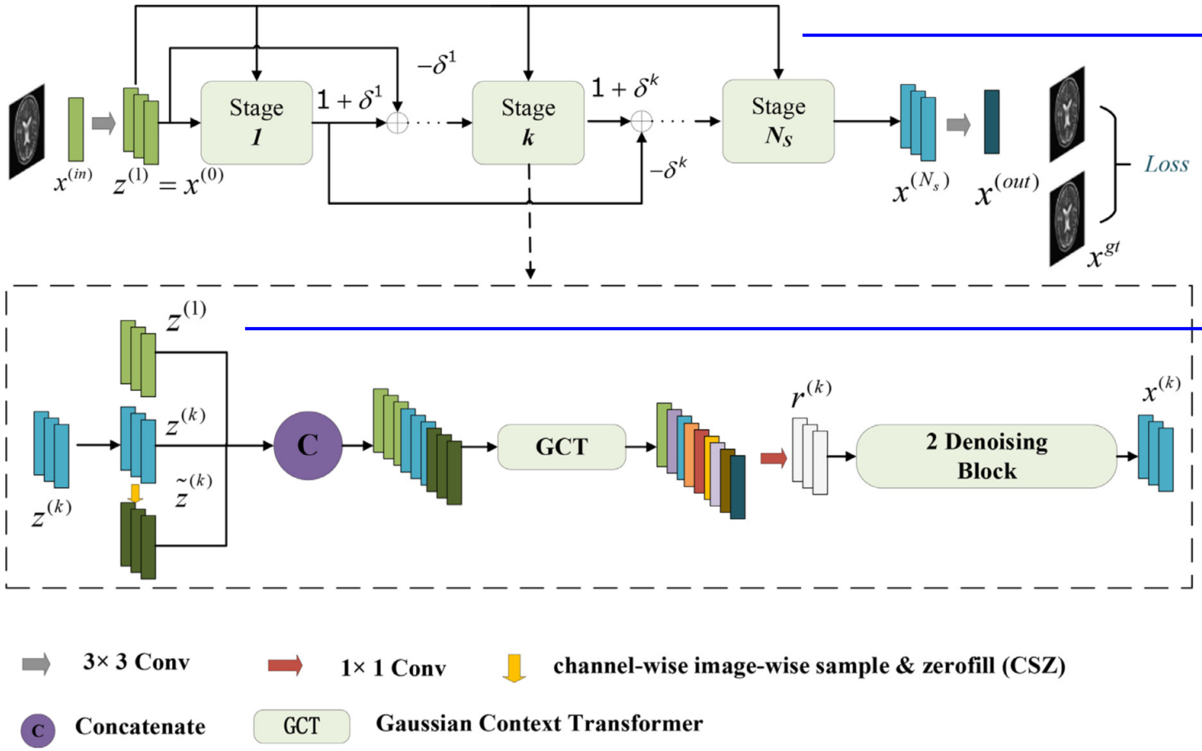
- Performance: It outperforms the state-of-the-art model-based and deep learning methods and exhibits good generalization

- Highlights: It proposed a model-based deep learning.



HFSIC-Net [Geng et al. (2023)]

- Single-channel information transmission significantly limits the learning abilities of the network → Multi-channel.
- How to assign weights based on different channels → Introducing the channel attention mechanism.



Multi-channel

Attention

Conclusion

- Sparse optimization is an important optimization and is widely used in the field of compressed sensing and linear inverse problems.

- There are some approaches for solving sparse optimization:
  - Fully learned approaches (e.g., SDA, ReconNet) which use an end-to-end learning strategy, has the advantage of being computationally efficient.

  - Another approach aims to train a predictor by unrolling the iterative algorithm into feed-forward layers (e.g., ADMM-Net, FISTA-Net, HFSIC-Net).

  - FISTA-Net incorporates traditional optimization procedures in DL training.

  - HFSIC-Net uses more deep learning techniques (Attention, Cross connection in DL).

Reference

- Candes E, Tao T. Decoding by linear programming [J]. IEEE Transactions on Information Theory, 2005, 51: 4203-4215.
- D. L. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," Proceedings of the National Academy of Sciences, vol. 106, no. 45, pp. 18 914–18 919, 2009.
- Natarajan B K. Sparse approximate solutions to linear systems [J]. SIAM Journal on Computing, 1995, 24: 227-234.
- Donoho D, Huo X. Uncertainty principles and ideal atomic decompositions [J]. IEEE Transactions on Information Theory, 2001, 47: 2845-2862.
- Mo Q, Li S. New bounds on the restricted isometry constant δ2k [J]. Applied and Computational Harmonic Analysis, 2011, 31(3): 460-468.
- Zhang Y. Theory of compressive sensing via l1-minimization: a non-RIP analysis and extensions [R]. Rice University, CAAM Technical Report TR08-11, 2008.
- Mousavi A, Patel A B, Baraniuk R G. A deep learning approach to structured signal recovery[C]//2015 53rd annual allerton conference on communication, control, and computing (Allerton). IEEE, 2015: 1336-1343.
- Kulkarni, K., Lohit, S., Turaga, P., Kerviche, R., & Ashok, A. (2016). Reconnet: Non-iterative reconstruction of images from compressively sensed measurements. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 449-458).

Reference

- Shi W, Jiang F, Liu S, et al. Image compressed sensing using convolutional neural network[J]. IEEE Transactions on Image Processing, 2019, 29: 375-388.
- Boyd, S., Parikh, N., Chu, E., Peleato, B., & Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, *3*(1), 1-122.
- Bach, F., Jenatton, R., & Mairal, J. (2011). Optimization with Sparsity-Inducing Penalties (Foundations and Trends(R) in Machine Learning).
- Kavi, Buckles, & Bhat. (1986). A Formal Definition of Data Flow Graph Models. IEEE Transactions on Computers, C–35(11), 940–948
- Xiang J, Dong Y, Yang Y. FISTA-net: Learning a fast iterative shrinkage thresholding network for inverse problems in imaging[J]. IEEE Transactions on Medical Imaging, 2021, 40(5): 1329-1339.
- Liu, J., & Chen, X. (2019, January). ALISTA: Analytic weights are as good as learned weights in LISTA. In *International Conference on Learning Representations (ICLR)*.
- Beck A, Teboulle M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems[J]. SIAM journal on imaging sciences, 2009, 2(1): 183-202.
- Geng, C., Jiang, M., Fang, X., Li, Y., Jin, G., Chen, A., & Liu, F. (2023). HFIST-Net: High-throughput fast iterative shrinkage thresholding network for accelerating MR image reconstruction. *Computer Methods and Programs in Biomedicine*, *232*, 107440.