From Sparse Coding to Deep Learning

Shihua Zhang

November 14, 2024

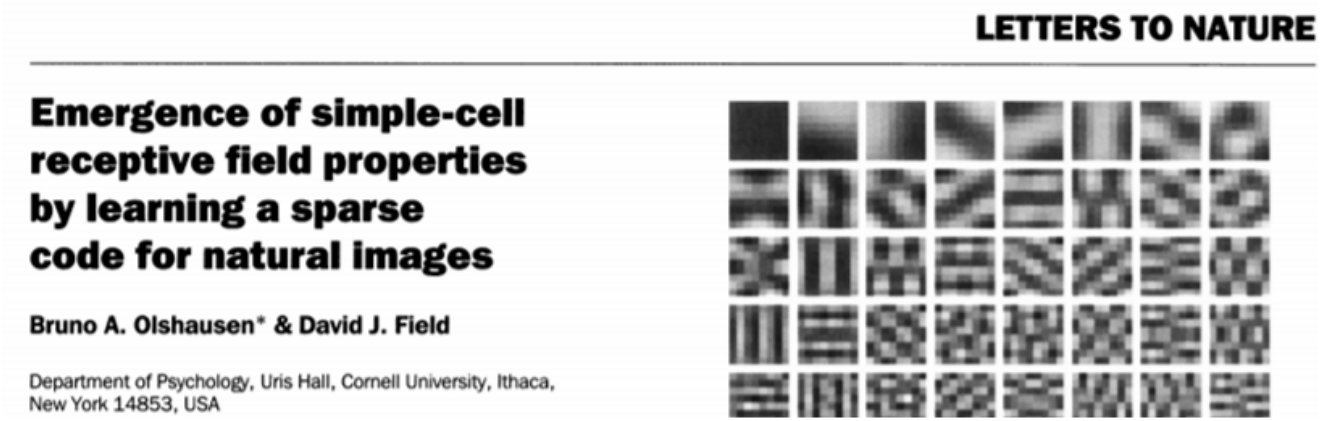
Outline

Sparse Coding: Birth

- Inspired by signal transform and visual cortex studies, sparse coding of natural images was developed [Olshausen and Field, 1996].

Bruno Olshausen David Field

Department of Psychology at Cornell University



LETTERS TO NATURE

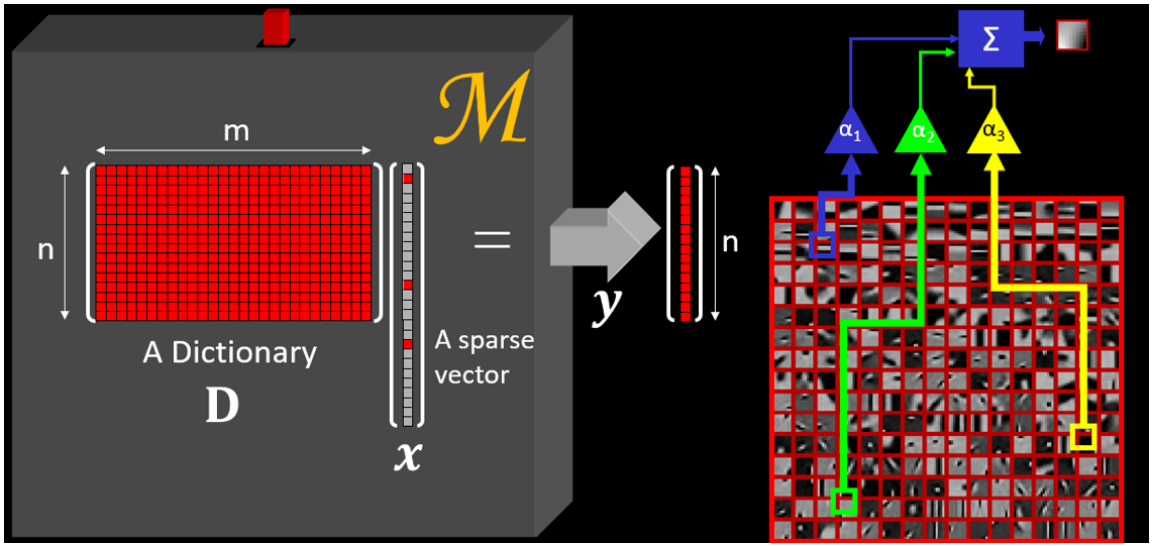
Emergence of simple-cell receptive field properties by learning a sparse code for natural images

Bruno A. Olshausen* & David J. Field

Department of Psychology, Uris Hall, Cornell University, Ithaca, New York 14853, USA

Sparse Coding: Model

- Task: model image patches
- Assumption: every patch can be described as a linear combination of a few atoms, where the atoms are learned from data.



- Assume $D \in \mathbb{R}^{n \times m}$ is an overcomplete dictionary $(m \gg n)$, $Y \in \mathbb{R}^n$ is an input signal, $X \in \mathbb{R}^m$ is a sparse representation of Y based on D:

$$Y = DX$$

Sparse Coding

- Let $P(\cdot)$ be a regularization term to ensure sparseness, then the problem can be rewritten as follows:

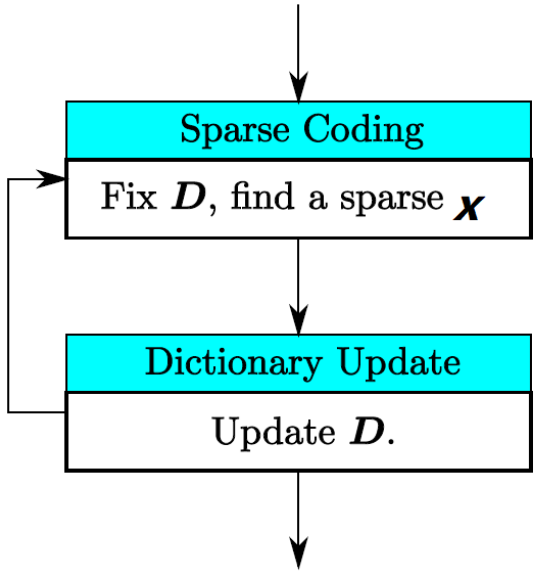
$$\min_{X,D} \frac{1}{2}\|Y - DX\|_2^2 + \lambda P(X)$$

Sparse Coding

- Let $P(\cdot)$ be a regularization term to ensure sparseness, then the problem can be rewritten as follows:

$$\min_{X,D} \frac{1}{2}\|Y - DX\|_2^2 + \lambda P(X)$$

- It can be splitted to two subproblems:
  - Sparse coding: Given X, fix D, find a sparse X
  - Dictionary learning: Given a family of Y, find a suitable dictionary D.



Iterative Shrinkage Thresholding Algorithm (ISTA)

- The origin problem can be rewritten as:

$$\min_X \frac{1}{2}\|Y - DX\|_2^2 + \lambda\|X\|_1 \qquad (P_1)$$

It is a traditional problem called Basis Pursuit (BP).

Iterative Shrinkage Thresholding Algorithm (ISTA)

- The origin problem can be rewritten as:

$$\min_X \frac{1}{2}\|Y - DX\|_2^2 + \lambda\|X\|_1 \qquad (P_1)$$

  It is a traditional problem called Basis Pursuit (BP).

- ISTA updates:

$$X^{l+1} = S_{\frac{\lambda}{L}}\left(X^l - \frac{1}{L}D^T(DX^l - Y)\right)$$

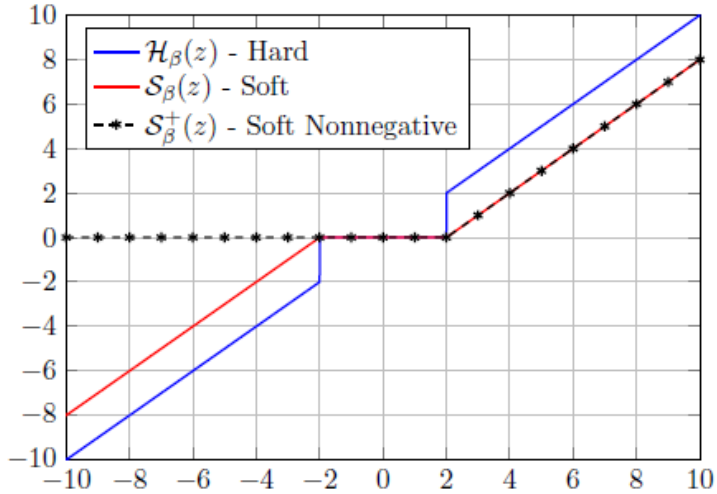


Figure 3: The thresholding operators for a constant $\beta = 2$.

Theoretical Guarantee

- Can ISTA find the unique solution?
  ——The answer is YES under certain circumstances

Definition 1

Assume d_i is the column vector of D, $\hat{d}_i = \frac{\hat{d}_i}{\|\hat{d}_i\|_2}$, the mutual coherence $\mu(D)$ of dictionary D is defined as: $\mu(D) = \max_{i \neq j} |\hat{d}_i^T \hat{d}_j|$

Theorem 2

The convex relaxation approaches above can recover the true solution X^* if $\|X^*\|_0 < \frac{1}{2}(1 + \frac{1}{\mu(D)})$ [Donoho et al., 2005]

Approximation Algorithm

- There is a simplest approximation algorithm [Papyan et al., 2017a]:
  - Compute the inner products between signal Y and all atoms in D.
  - Choose the atoms corresponding to the highest responses.

Approximation Algorithm

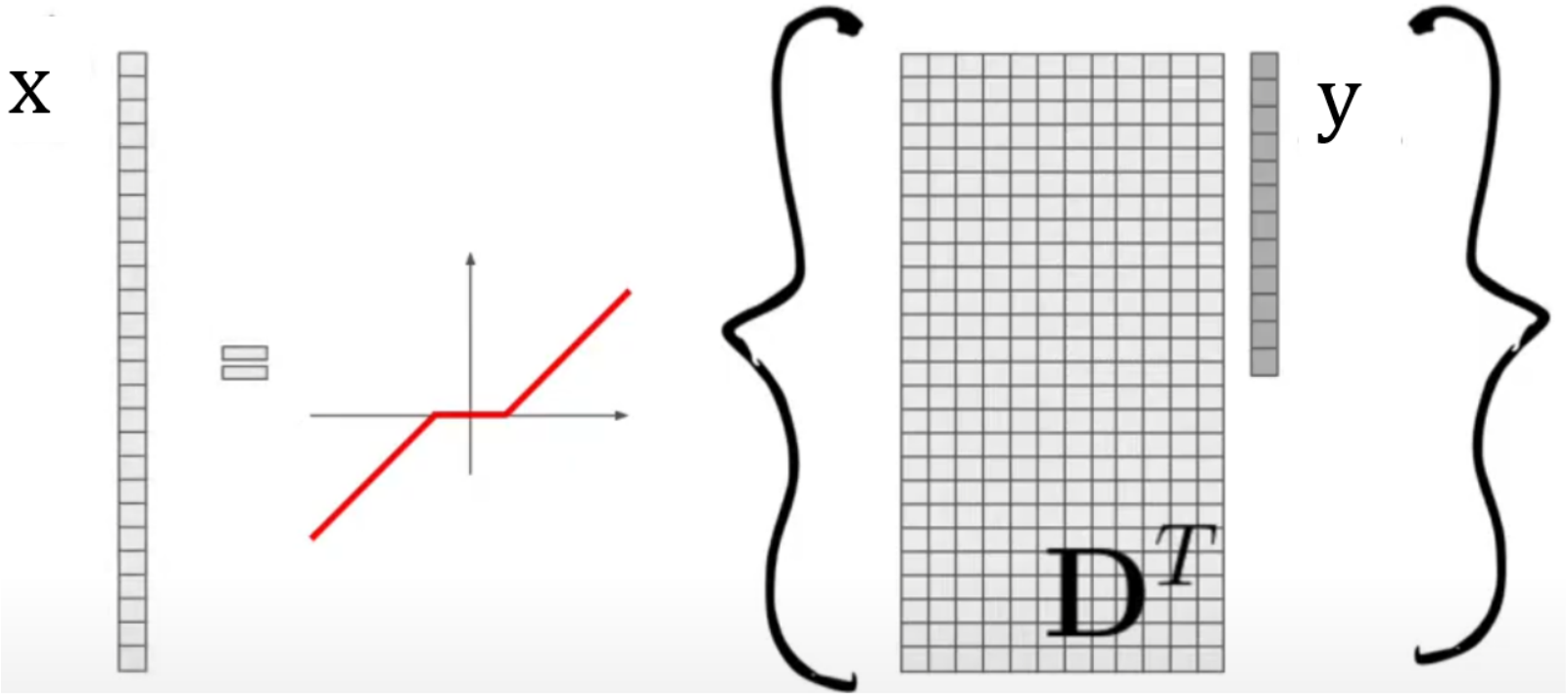
- There is a simplest approximation algorithm [Papyan et al., 2017a]:
  - Compute the inner products between signal Y and all atoms in D.
  - Choose the atoms corresponding to the highest responses.

- The approximation problem can be written as:

$$\min_X \frac{1}{2}\|X - D^T Y\|_2^2 + \beta\|X\|_1$$

- The solution to the above form is simple: $X = S_\beta(D^T Y)$.
- The theoretical guarantee of this method is weaker than ISTA.

Approximation Algorithm

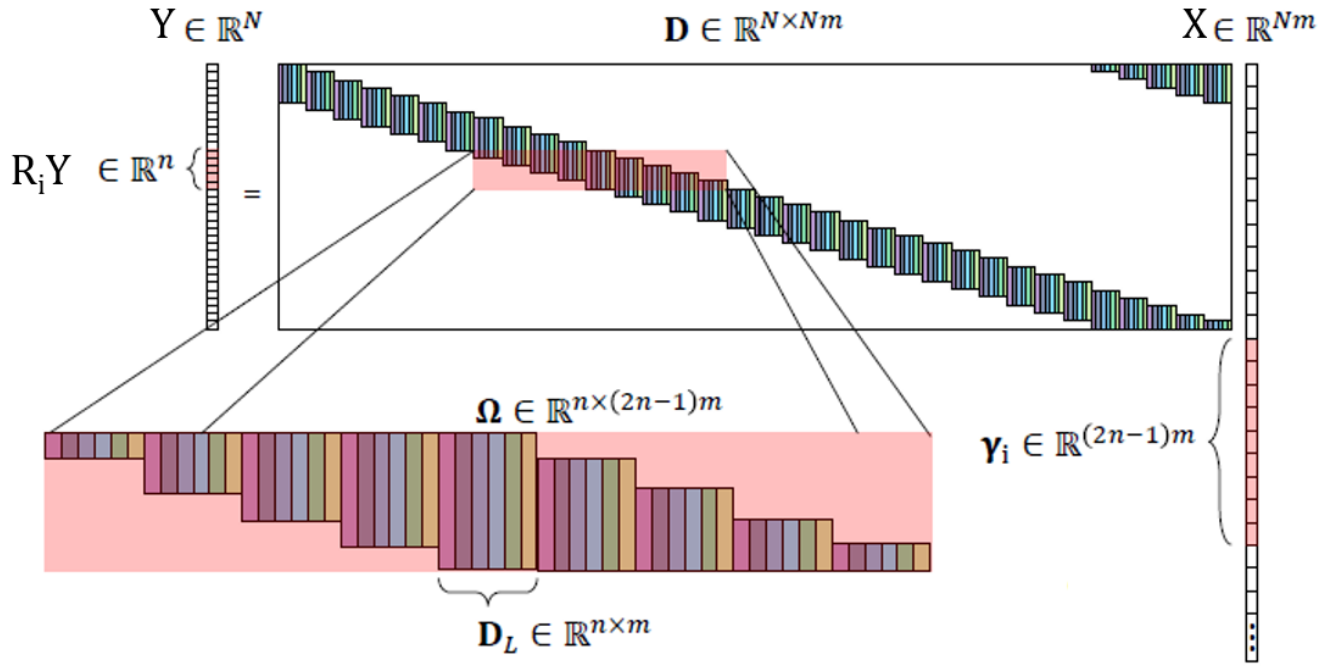


- This is very similar with a one layer hidden neural network!!

Outline

Convolutional Sparse Coding (CSC)

- Sparse coding suffers from the curse of dimensionality.
- Solution 1: train a local model for patches extracted from Y and process them independently.
- Solution 2: adopt convolutional dictionary built from shifted versions of a local matrix D_L [Sulam and Elad, 2015].



Convolution Sparse Coding (CSC)

- Why convolutional dictionary?

- Convolutional model can train the local patches naturally.
  • Assume the patch size is n, R_i is a extract operator, $a_i = R_i Y \in \mathbb{R}^n$ is a local patch extracted from Y and begin at the i-th entry of Y.
  • For convolution model, $a_i = R_i Y = R_i DX = \Omega\gamma_i$, γ_i is the corresponding patches in X, where $R_i Y \in \mathbb{R}^n, \Omega \in \mathbb{R}^{n\times(2n-1)m}, \gamma_i \in \mathbb{R}^{(2n-1)m}$.
  • Convolutional dictionary decrease the parameters significantly.

Convolution Sparse Coding (CSC)

- Why convolutional dictionary?

- Convolutional model can train the local patches naturally.
  • Assume the patch size is n, R_i is a extract operator, $a_i = R_i Y \in \mathbb{R}^n$ is a local patch extracted from Y and begin at the i-th entry of Y.
  • For convolution model, $a_i = R_i Y = R_i DX = \Omega\gamma_i$, γ_i is the corresponding patches in X, where $R_i Y \in \mathbb{R}^n, \Omega \in \mathbb{R}^{n\times(2n-1)m}, \gamma_i \in \mathbb{R}^{(2n-1)m}$.
  • Convolutional dictionary decrease the parameters significantly.

- Advantage: For a large value of m, $\mu(D) \approx \frac{1}{\sqrt{2n}}$. Classical sparse coding results would allow merely $O(\sqrt{n})$ non-zeros in all X while convolution model allow $O(\sqrt{n})$ non-zeros in n-length patches.

Convolution Sparse Coding (CSC)

- Convolutional model has a better theoretical guarantee.

Definition 3

Define the pseudo-norm $\mathcal{L}_{0,\infty}$ of a global sparse vector X as:

$$\|X\|_{0,\infty} = \max_i \|\gamma_i\|_0$$

Theorem 4

Given the system of linear equations $Y = DX$, if a solution X exists satisfying

$$\|X\|_{0,\infty} < \frac{1}{2}\left(1 + \frac{1}{\mu(D)}\right)$$

then BP and OMP is guaranteed to recover it [Papyan et al., 2017b].

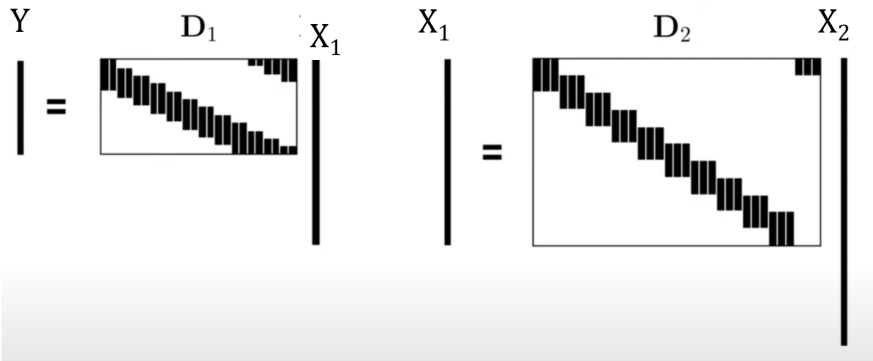
Multi-Layer CSC

- Double sparsity attempts to benefit from both the computational efficiency of analytically defined matrices and the adaptability of data driven dictionaries [Rubinstein et al., 2009].

$$Y = D_1 D_2 X_2$$

Here D_1 is an analytic dictionary and D_2 is a trained sparse one.

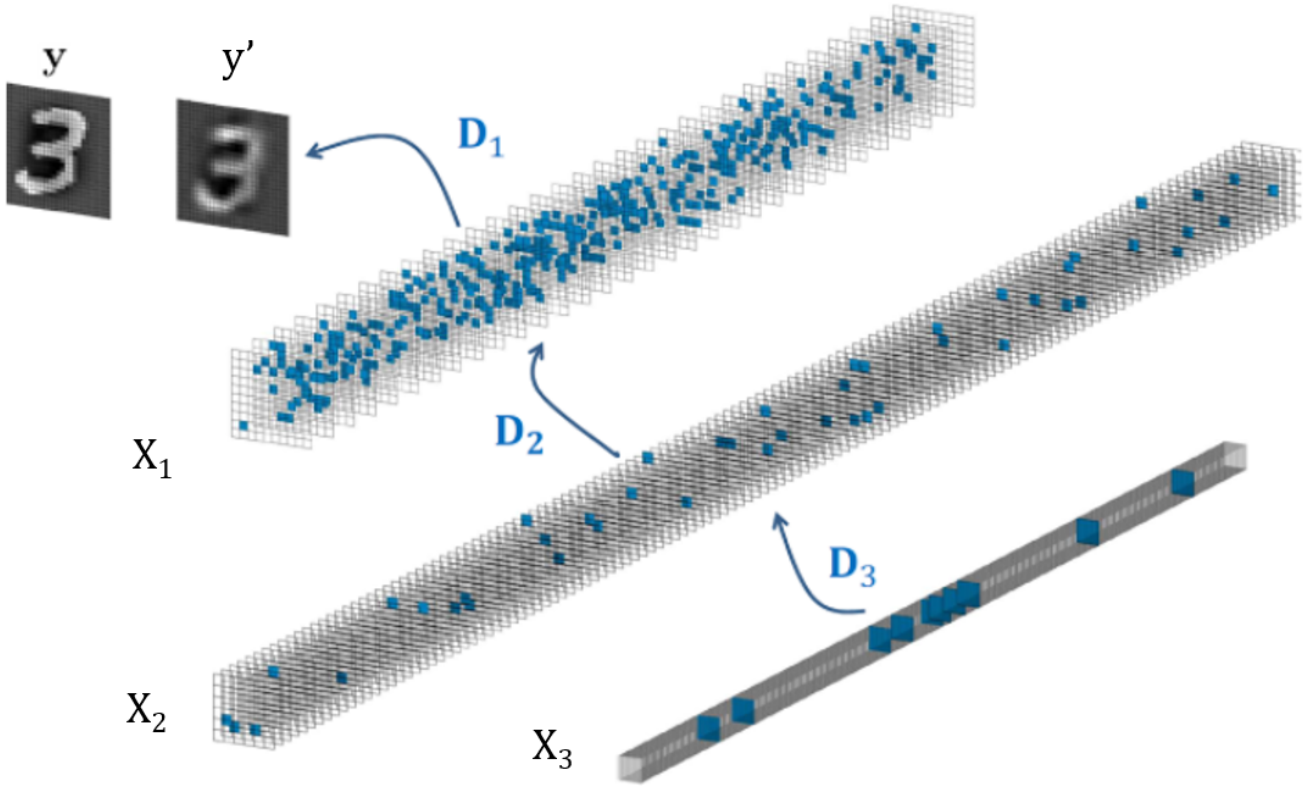
- Since both D_2 and X_2 are sparse, we expect $X_1 = D_2 X_2$ is sparse.



- In CSC, further regard the representation X_1 as a signal and learn its sparse representation X_2 [Papyan et al., 2017a].

$$Y = D_1 X_1, \quad X_1 = D_2 X_2$$

Multi-Layer CSC



- Intuitively, $Y = D_1 X_1$ assumes that the signal Y is a superposition of atoms taken from D_1. While $Y = D_1 D_2 X_2$ views the signal as a superposition of more complex entities (molecules) taken from $D_1 D_2$.

Multi-Layer CSC

- Clearly, the construction can be extended to more than two layers.

Definition 5

For a global signal Y, a set of convolutional dictionaries $\{D_i\}_{i=1}^K$, and a vector λ, define the deep coding problem DCP_λ as:

$$(DCP_\lambda): \quad \text{find} \quad \{X_i\}_{i=1}^K \quad \text{s.t.} \quad \begin{aligned} Y &= D_1 X_1, \quad \|X_1\|_{0,\infty} \leqslant \lambda_1 \\ X_1 &= D_2 X_2, \quad \|X_2\|_{0,\infty} \leqslant \lambda_2 \\ &\vdots \\ X_{K-1} &= D_K X_K, \quad \|X_K\|_{0,\infty} \leqslant \lambda_K \end{aligned}$$

where the scalar λ_i is the i-th entry of λ.

Multi-Layer CSC

- The DCP_λ problem can be extended to a noisy regime.

Definition 6

For a global signal Y, a set of convolutional dictionaries $\{D_i\}_{i=1}^K$, and a vector λ and ϵ, define the deep coding problem DCP_λ^ϵ as:

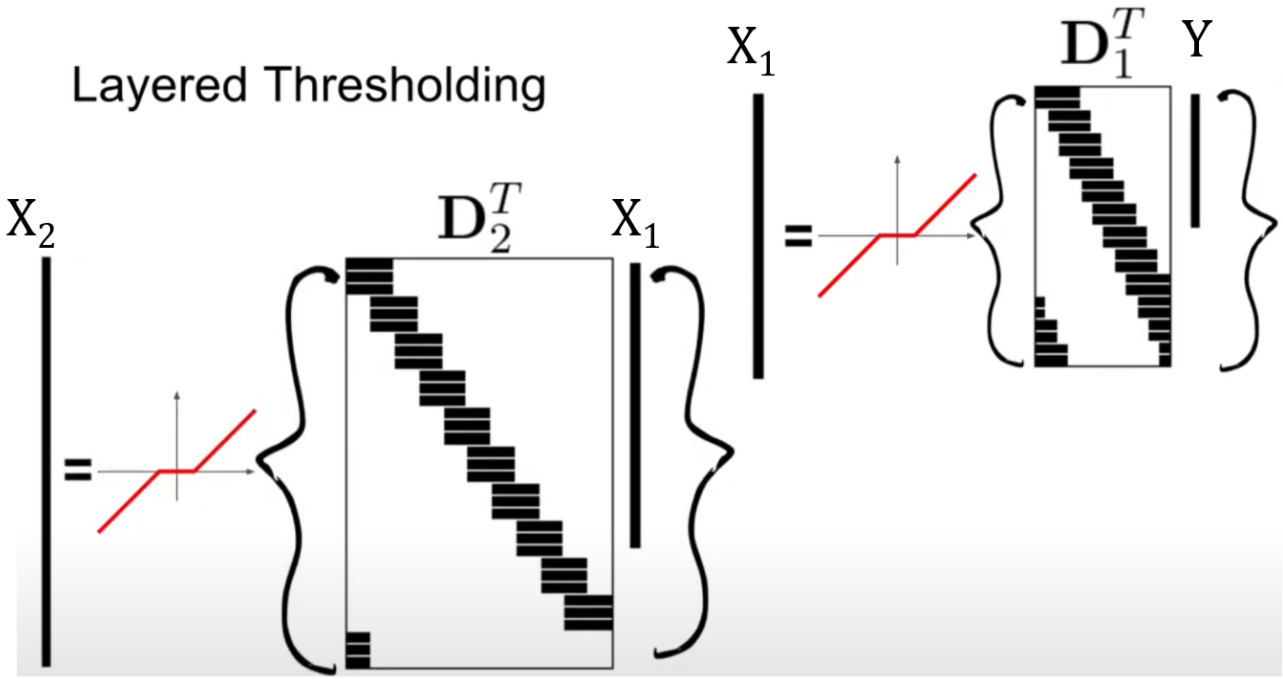
$$
\begin{aligned}
\left(DCP_\lambda^\epsilon\right): \text{find} \quad \{X_i\}_{i=1}^K \quad \text{s.t.} \quad & \|Y - D_1 X_1\|_2 \leqslant \epsilon_0, \quad \|X_1\|_{0,\infty} \leqslant \lambda_1 \\
& \|X_1 - D_2 X_2\|_2 \leqslant \epsilon_1, \quad \|X_2\|_{0,\infty} \leqslant \lambda_2 \\
& \qquad \vdots \\
& \|X_{K-1} - D_K X_K\|_2 \leqslant \epsilon_{K-1}, \quad \|X_K\|_{0,\infty} \leqslant \lambda_K
\end{aligned}
$$

where the scalar λ_i and ϵ_i is the i-th entry of λ and ϵ.

Multi-Layer CSC

- For DCP_λ problem, we can use the layered thresholding method

$$X_i = S_{\beta_i}(D_i^T X_{i-1})$$



Theoretical Guarantee

Theorem 7

Suppose a signal Z has a decomposition $Z = D_1 X_1, ..., X_{K-1} = D_K X_K$ and that it is contaminated with noise E to create the signal $Y = Z + E$, such that $\|E\|_{0,\infty} \leqslant \epsilon_0$. Denote by $|X_i^{min}|$ and $|X_i^{max}|$ the lowest and highest entries in absolute value in the vector X_i, respectively. Let $\{X_i'\}_{i=1}^K$ be the set of solutions obtained by running the layered soft thresholding algorithm with thresholds $\{\beta_i\}_{i=1}^K$,i.e.$X_i' = S_{\beta_i}(D_i^T X_{i-1}')$ where $X_0' = Y$. Assuming that $\forall 1 \leqslant i \leqslant K$

a. $\|X_i\|_{0,\infty} < \frac{1}{2}\left(1 + \frac{1}{\mu(D_i)}\frac{|X_i^{\min}|}{|X_i^{\max}|}\right) - \frac{1}{\mu(D_i)}\frac{\epsilon_{i-1}}{|X_i^{\max}|}$

b. The threshold β_i is chosen according to

$$|X_i^{\min}| - (\|X_i\|_{0,\infty} - 1)\,\mu(D_i)\,|X_i^{\max}| - \epsilon_{i-1} > \beta_i > \|X_i\|_{0,\infty}\,\mu(D_i)\,|X_i^{\max}| + \epsilon_{i-1}$$

then 1. The support of the solution X_i' is equal to that of X_i;
2. $\|X_i' - X_i\|_{2,\infty} \leqslant \epsilon_i$,
where $\epsilon_i = \sqrt{\|X_i\|_{0,\infty}^{\mathrm{P}}}\left(\epsilon_{i-1} + \mu(D_i)\left(\|X_i\|_{0,\infty} - 1\right)|X_i^{\max}| + \beta_i\right)$

Layered ISTA

- For DCP_λ problem, we can also use the layered ISTA

$$X_i^{l+1} = S_{\frac{\lambda}{L}}\left(X_i^l - \frac{1}{L}(D_i)^T(D_i X_i^l - X_{i-1})\right)$$

Theorem 8

For DCP_λ problem, the layered ISTA is guaranteed to recover the true representation $\{X_i\}$, if $\forall 1 \leqslant i \leqslant K$

$$\|X_i\|_{0,\infty} < \frac{1}{2}\left(1 + \frac{1}{\mu(D_i)}\right)$$

More General Settings

- Li *et al.* extended the models and also established the similar theoretical guarantees [Li et al., 2024].

Definition 9

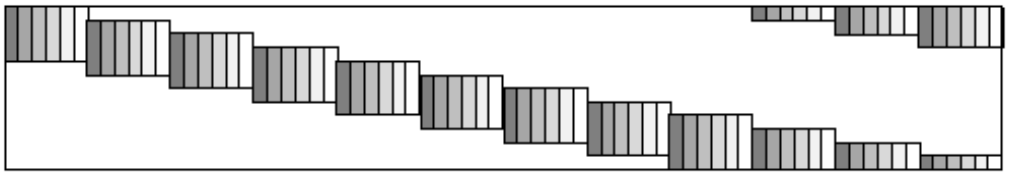
For a global noised signal Y, a set of dictionaries $\{D_i\}_{i=1}^K$, a vector $\lambda \in \mathbb{R}_+^K$ and a tolerance vector $\epsilon \in \mathbb{R}_+^K$, we call $\{X_i\}_{i=1}^K$ a set of sparse codings of $DCP_{0,\lambda}^{\epsilon}$ if it satisfires

$$DCP_{0,\lambda}^{\epsilon}(Y, \{D_i\}_{i=1}^K) \text{ find } \{X_i\}_{i=1}^K \text{s.t.} \quad \begin{aligned} \|Y - D_1X_1\|_2 &\leqslant \epsilon_0, \quad \|X_1\|_0 \leqslant \lambda_1 \\ \|X_1 - D_2X_2\|_2 &\leqslant \epsilon_1, \quad \|X_2\|_0 \leqslant \lambda_2 \\ &\vdots \\ \|X_{K-1} - D_KX_K\|_2 &\leqslant \epsilon_{K-1}, \quad \|X_K\|_0 \leqslant \lambda_K \end{aligned}$$

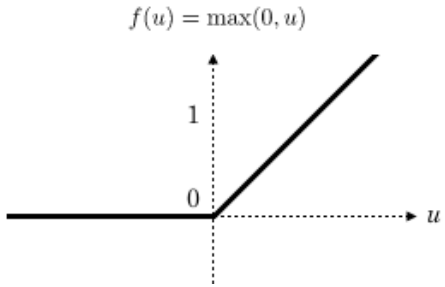
Outline

Convolution Neural Network (CNN)

- Convolution operator can be expressed as a convolutional matrix multipier



- ReLU is the commonly used nonlinear activation:



- The output of the i-th layer is

$$X_i = \text{ReLU}(W_i X_{i-1} + b_i)$$

Connections of ML-CSC and NN

- Convolutional Sparse Coding (CSC) [Zeiler et al., 2011]
  - Why Convolutional? Local interactions!
  - Dictionary can be learned via local processing
- Multi-Layered CSC (ML-CSC) [Papyan et al., 2017a]
  - Why Deep? Learn more complex filters!
  - Related closely with CNN
  - Sparse dictionaries assumption

Connections of ML-CSC and NN

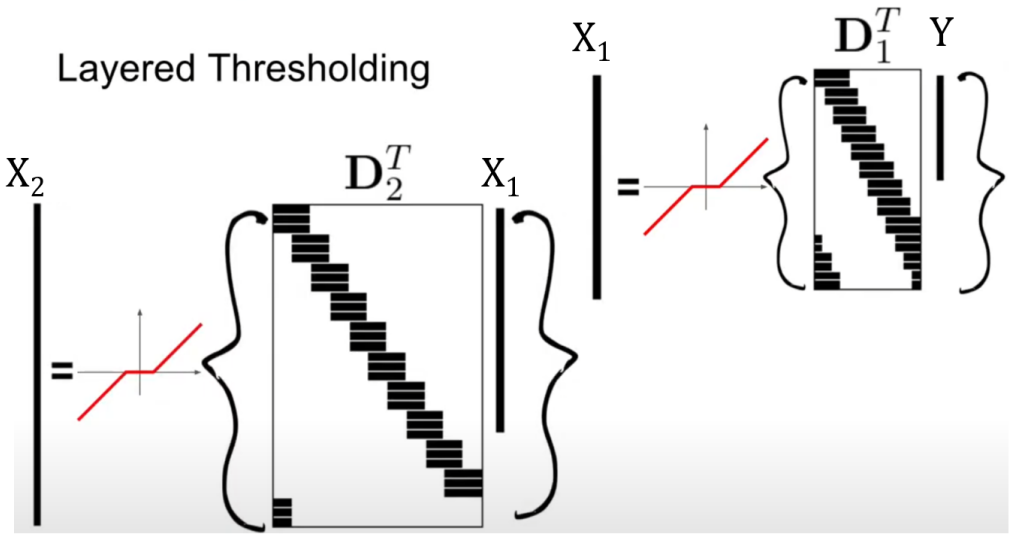
- Convolutional Sparse Coding (CSC) [Zeiler et al., 2011]
  - Why Convolutional? Local interactions!
  - Dictionary can be learned via local processing
- Multi-Layered CSC (ML-CSC) [Papyan et al., 2017a]
  - Why Deep? Learn more complex filters!
  - Related closely with CNN
  - Sparse dictionaries assumption
- In CSC, using layered threshold algorithm, the update of X_i is:

$$X_i = S_{\beta_i}(D_i^T X_{i-1})$$

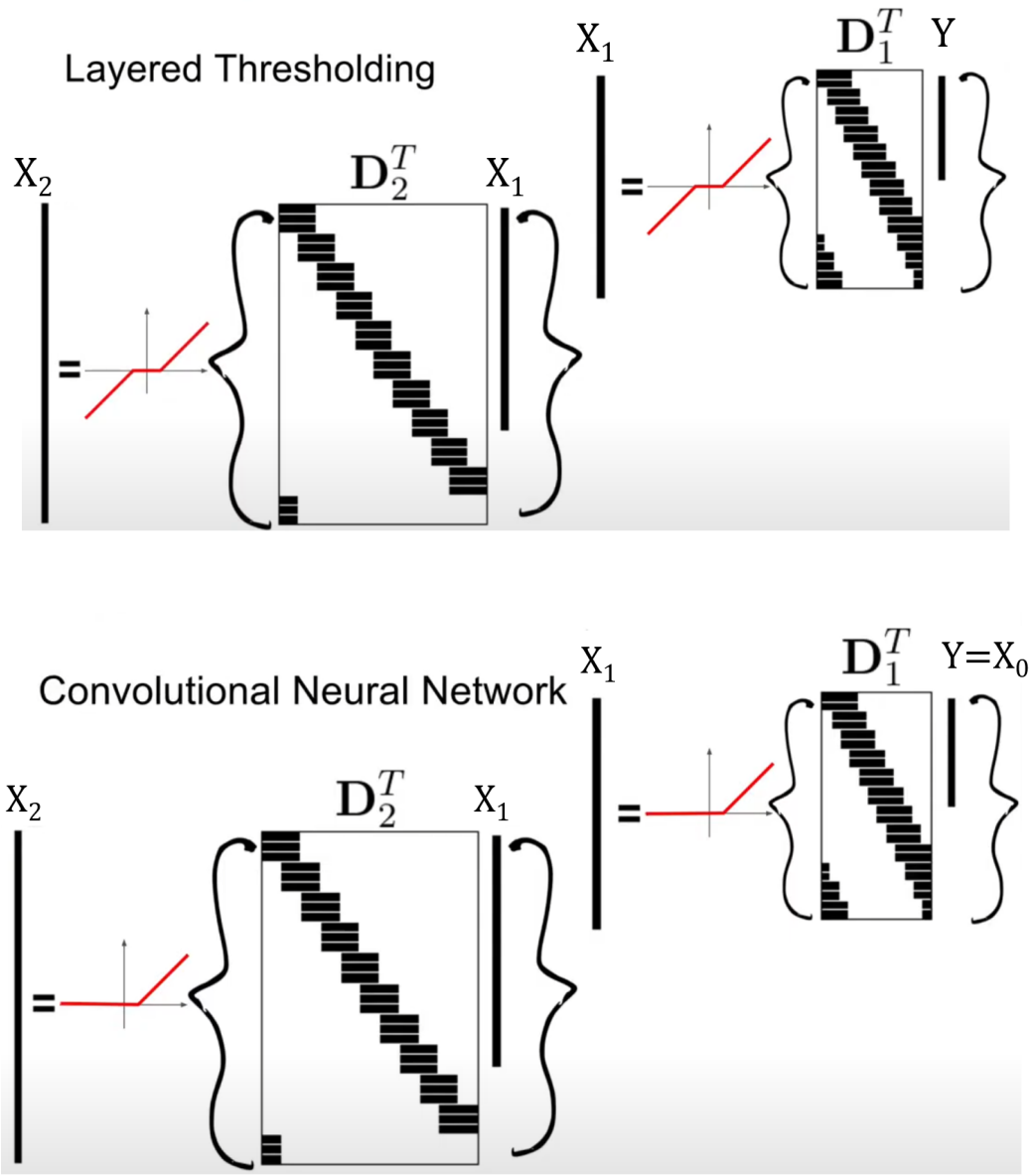
  where the thresholding operator $S_{\beta_i}(\cdot)$ is very similar to ReLU$(\cdot)$
- It is trival that the update form of X_i is same to the the update of features X in the forward propagation of CNN

$$X_i = \text{ReLU}(W_i X_{i-1} + b_i)$$

Connection of ML-CSC and CNN



Connection of ML-CSC and CNN



Matrix-vector Multiplication Form

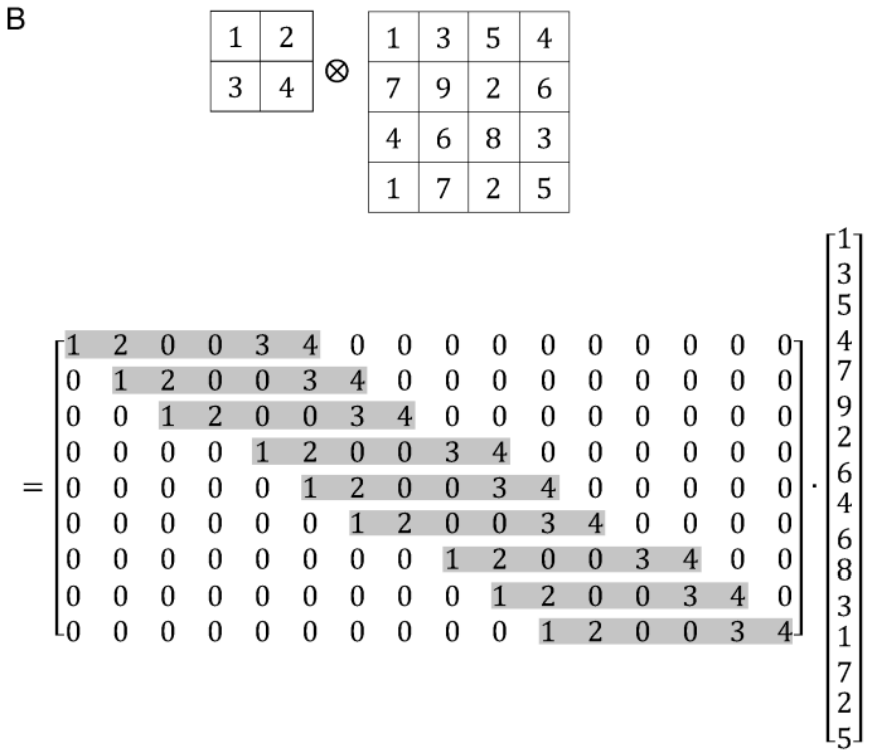


Figure: Dilation convolution ($s = 1$)

Matrix-vector Multiplication Form

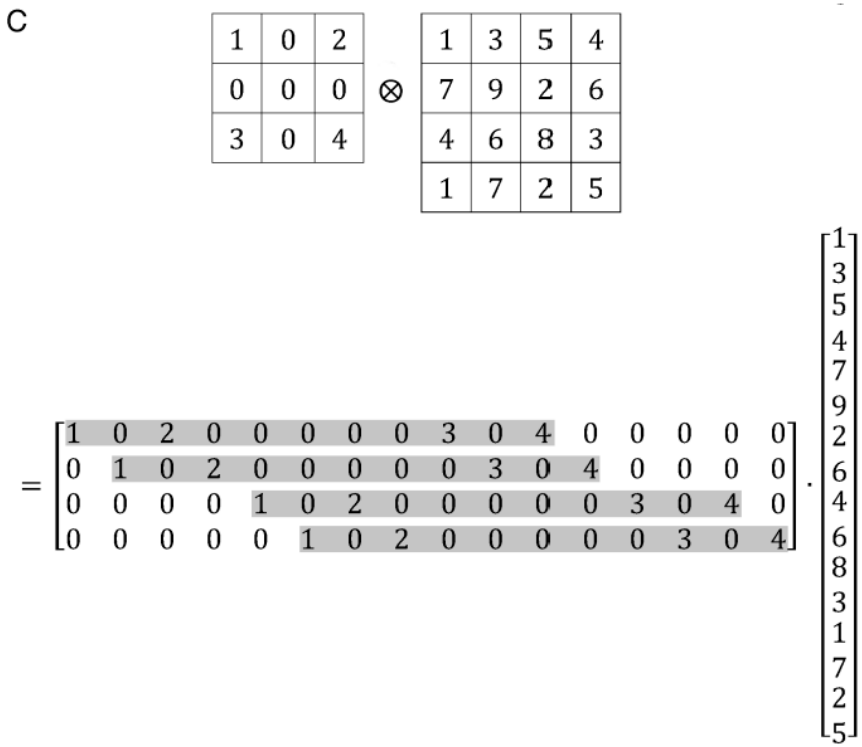
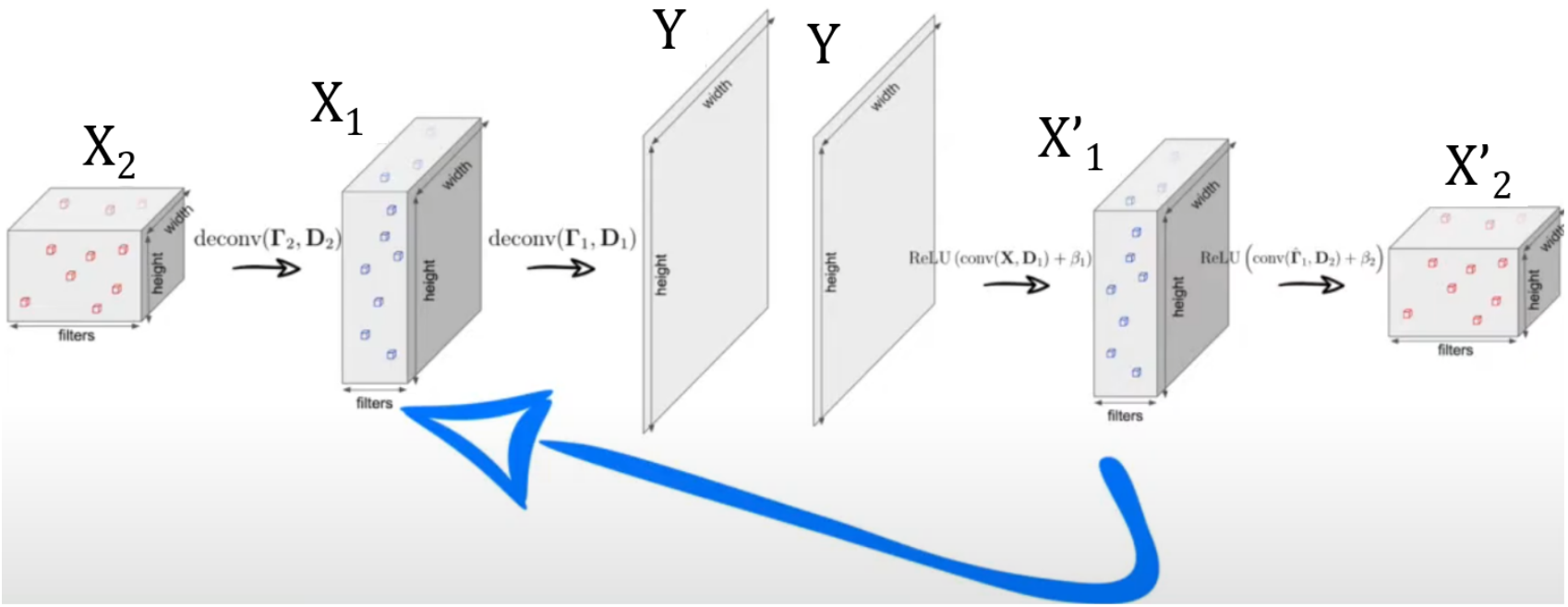


Figure: Dilation convolution ($s = 2$)

Theories of Deep Learning



Success of Forward Pass

- If $\|X_i\|_{0,\infty} < \frac{1}{2}\left(1 + \frac{1}{\mu(D_i)}\frac{|X_i^{\min}|}{|X_i^{\max}|}\right) - \frac{1}{\mu(D_i)}\frac{\epsilon_{i-1}}{|X_i^{\max}|}$

  Layered thresholding guarantees:
  - Find correct places of nonzeros.
  - $\|X_i' - X_i\|_{2,\infty} \leqslant \epsilon_i$,

  where $\epsilon_i = \sqrt{\|X_i\|_{0,\infty}^{\mathrm{P}}}\left(\epsilon_{i-1} + \mu(D_i)\left(\|X_i\|_{0,\infty} - 1\right)\left|X_i^{\max}\right| + \beta_i\right)$

- Limits:
  - ⋆ Forward pass always fail at recovering representations exactly.
  - ⋆ Success depends on ratio.
  - ⋆ Distance increases with layer.

Another view of connection

- In ISTA, the code is updated as follows:

$$X^{l+1} = S_{\frac{\lambda}{L}}\left(X^l - \frac{1}{L}D^T(DX^l - y)\right)$$

- Let the initial code $X^0 = 0$. Then we have

$$X^1 = S_{\frac{\lambda}{L}}\left(\frac{1}{L}D^T y\right)$$

- Multi-Layer CSC with initial code $X_i^0 = 0$

$$X_i^1 = S_{\frac{\lambda}{L}}\left(\frac{1}{L}(D_i)^T X_{i-1}^1\right)$$

Deep CNN:

$$X_i = \mathrm{ReLU}((W_i)^T X_{i-1} + b_i)$$

Success of Layered ISTA

- If $\|X_i\|_{0,\infty} < \frac{1}{3}\left(1 + \frac{1}{\mu(D_i)}\right)$
  Layered ISTA guarantees:
  - Find only correct places of nonzeros.
  - Find all coefficients that are big enough.
  - $\|X_i' - X_i\|_{2,\infty} \leqslant \epsilon_i$,

  where $\epsilon_i = \|E\|_{2,\infty}^{\mathrm{P}} 7.5^i \prod_{j=1}^{i} \sqrt{\|X_j\|_{0,\infty}^{\mathrm{P}}}$

- Limits:
  $\star$ Distance increases with layer.

Exponential convergence

- Li *et al.* has conducted an analysis of feature approximation in CNNs for deep sparse coding problems [Li et al., 2024].

Theorem 10

Given a set of dictionaries $\{D_i \in \mathbb{R}^{d_{i-1} \times d_i}\}_{i=1}^K$ *with each dictionary* D_i *being column-normalized with respective to* ℓ_2 *norm. We assume that each* X_i *satisfies* $\|X_i\|_0 \leqslant \lambda_i$, *and* $\|X_i\|_\infty \leqslant B_i, \|\epsilon\|_1 \leqslant \delta\}$. *Then there exists a CNN with kernel size* s, *depth* $O(M \log_s \prod_{i=1}^K (d_{i-1} + d_i))$ *and number of weights* $O(M \prod_{i=1}^K (d_{i-1} + d_i)^2)$ *such that the output, denoted by* $\{\tilde{X}_i\}_{i=1}^K$ *satisfies*

$$\|\tilde{X}_i - X_i\|_2 \leqslant C_{D,B,\lambda} e^{-c_{D,\lambda} M} + C_D \sum_{i=1}^K \delta_i.$$

where $c_{D,\lambda}, C_D, C_{D,B,\lambda} > 0$ *only depend on* $\{D_i\}_{i=1}^K$, λ *and* $B := \{B_i\}_{i=1}^K$

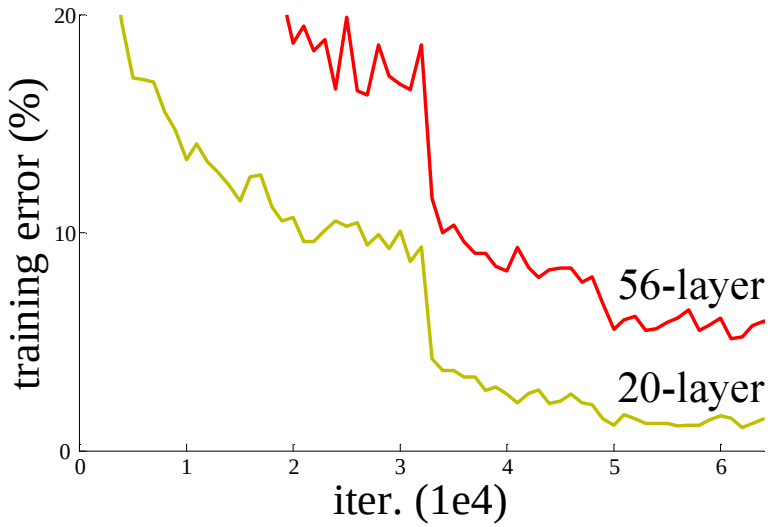
Conclusion

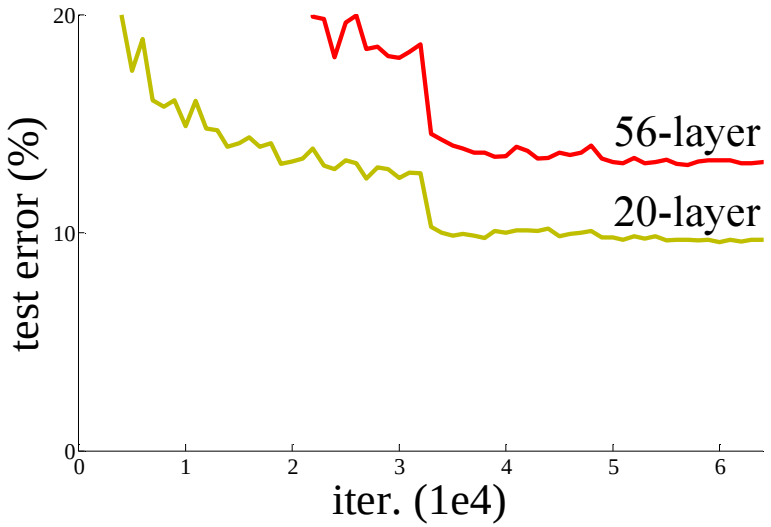
- Sparsity was well established theoretically.
- Sparsity is covertly exploited in practice: ReLU, dropout, stride, dilation...
- Sparsity is the secret sauce behind CNN.
- Need to bring sparsity to the surface to better understand CNNs.

Outline

Residual Neural Network (ResNet)

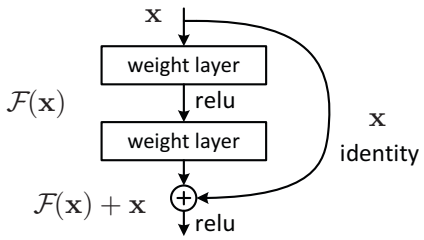
- For plain networks, there are two serious problems [He et al., 2016]:
  - Vanishing gradients
  - Degradation problem: with the network depth increasing, accuracy gets saturated and then degrades rapidly.

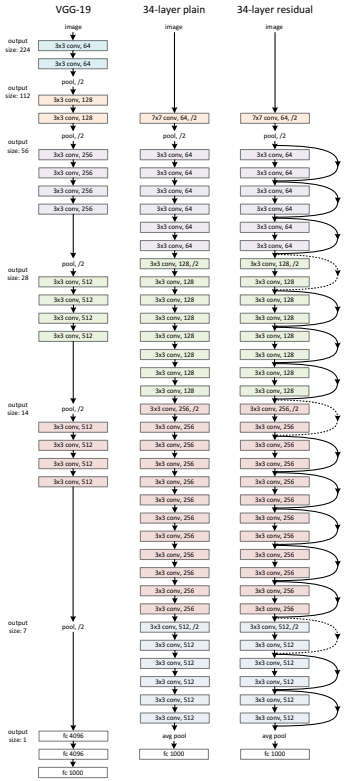




Residual Neural Network (ResNet)

- To avoid these problems, skip connections were introduced.



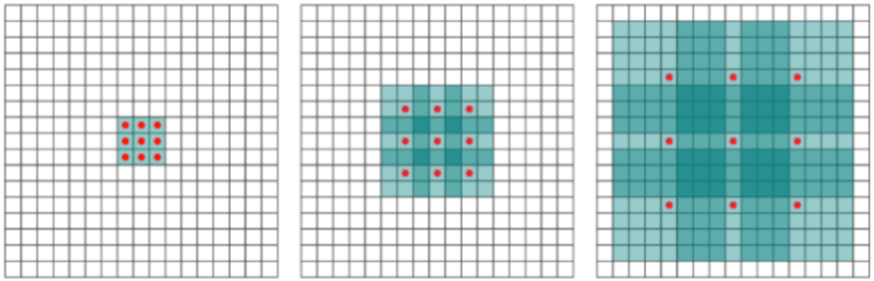


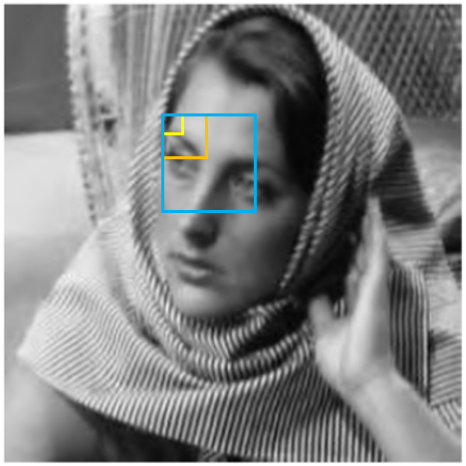
- The output of the i-th layer is

$$X_i = \sigma(W_{i-1,i}X_{i-1} + b_i + W_{i-2,i}X_{i-2})$$

Mixed-Scale Dense CNN (MSDNet)

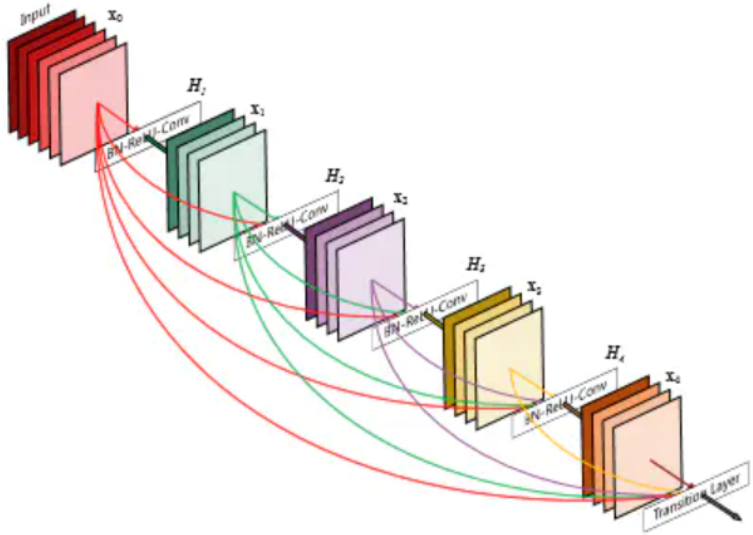
- MSDNet is another model of deep learning.
- Two special structures: dialated convolution and dense connection.
- Dialated convolution: it can capture the features in different scale with the same amount of parameters [Pelt and Sethian, 2018].





Mixed-Scale Dense CNN (MSDNet)

- Dense connection: the iuput of current layer is the concatenation of the output of all the previous layers.



- Dense connection can maximize the utilization of data and features captured by the shallow layers.

Towards to Understand Skip-Connection DNN

Can we generalize ML-CSC for those advanced NNs?

ResNet, DenseNet, MSDNet, ...

Towards to Understand Skip-Connection DNN

Can we generalize ML-CSC for those advanced NNs?
ResNet, DenseNet, MSDNet, ...

Three factors in each layer of ML-CSC
affect their performance [Zhang and Zhang, 2021]

- The initialization (Res-CSC)
- The dictionary design (MSD-CSC)
- The number of iterations (Optimization)

Res-CSC

Here we denote X as the signal and Γ as the sparse code. ISTA update:

$$\Gamma^{k+1} = S_{\frac{\beta}{L}}\left(\Gamma^k - \frac{1}{L}\left(-D^T X + D^T D\Gamma^k\right)\right) \tag{1}$$

Its first step when set $\Gamma^0 = 0$

$$\Gamma^1 = S_{\frac{\beta}{L}}\left(\frac{1}{L}\left(D^T X\right)\right) \tag{2}$$

Res-CSC

Here we denote X as the signal and Γ as the sparse code. ISTA update:

$$\Gamma^{k+1} = S_{\frac{\beta}{L}}\left(\Gamma^k - \frac{1}{L}\left(-D^T X + D^T D\Gamma^k\right)\right) \tag{1}$$

Its first step when set $\Gamma^0 = 0$

$$\Gamma^1 = S_{\frac{\beta}{L}}\left(\frac{1}{L}\left(D^T X\right)\right) \tag{2}$$

Layer-Initialization is the key

$$\Gamma^1 = S_{\frac{\beta}{L}}\left(\frac{1}{L}D^T X + X_{-1} - \frac{1}{L}D^T D X_{-1}\right) \tag{3}$$

Res-CSC

Layer-Initialization is the key!

$$\Gamma^1 = S_{\frac{\beta}{L}}\left(\frac{1}{L}D^T X + X_{-1} - \frac{1}{L}D^T D X_{-1}\right) \tag{4}$$

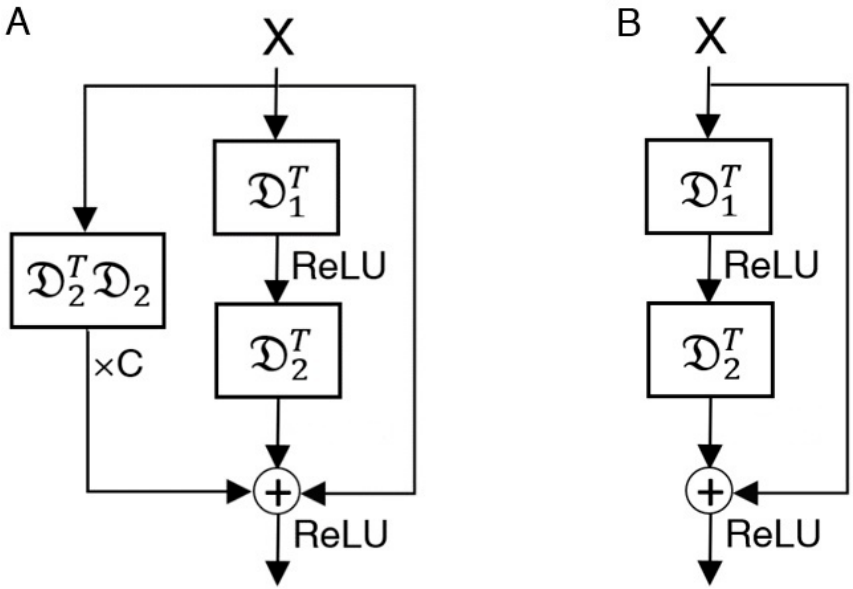


Figure: Res-CSC (A) and its variant (B)

Sparse Coding Scheme of MSD-CSC

Dictionary design $D_i^{s_i} = \begin{bmatrix} \mathrm{I} & \left(F_i^{s_i}\right)^T \end{bmatrix}$

$$\begin{bmatrix} \Gamma_{i-1}^{(1)} \\ \Gamma_{i-1}^{(2)} \\ \Gamma_{i-1}^{(3)} \\ \vdots \\ \Gamma_{i-1}^{(j)} \\ \vdots \\ \Gamma_{i-1}^{(n)} \end{bmatrix} \approx \left[\begin{array}{ccccccc|} 1 & 0 & 0 & \cdots & 0 & 0 & 0 & \\ 0 & 1 & 0 & \cdots & 0 & 0 & 0 & \\ 0 & 0 & 1 & \cdots & 0 & 0 & 0 & \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \\ 0 & 0 & 0 & \cdots & 1 & 0 & 0 & \\ 0 & 0 & 0 & \cdots & 0 & 1 & 0 & \\ 0 & 0 & 0 & \cdots & 0 & 0 & 1 & \end{array}\right. \left(F_i^{s_i}\right)^T \left.\right] \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ \Gamma_{i-1}^{(j)} - \xi_j \\ 0 \\ \vdots \\ 0 \\ - - - - \\ \Gamma_i^{(1)} \\ \Gamma_i^{(2)} \\ \Gamma_i^{(3)} \\ \vdots \\ \Gamma_i^{(j)} \\ \vdots \\ \Gamma_i^{(n)} \end{bmatrix}$$

Sparse Coding Scheme of MSD-CSC

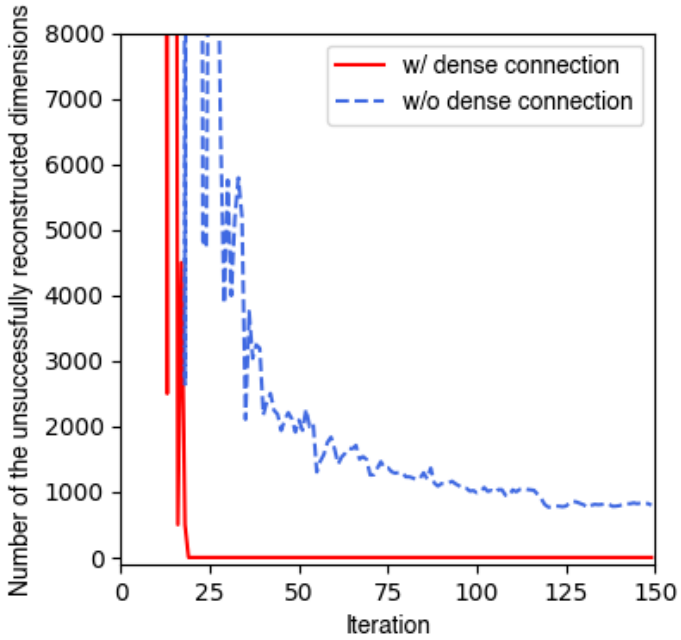
Dictionary design $D_i^{s_i} = \begin{bmatrix} \mathrm{I} & \left(F_i^{s_i}\right)^T \end{bmatrix}$

$$\begin{bmatrix} \Gamma_{i-1}^{(1)} \\ \Gamma_{i-1}^{(2)} \\ \Gamma_{i-1}^{(3)} \\ \vdots \\ \Gamma_{i-1}^{(j)} \\ \vdots \\ \Gamma_{i-1}^{(n)} \end{bmatrix} \approx \left[\begin{array}{ccccccc|} 1 & 0 & 0 & \cdots & 0 & 0 & 0 & \\ 0 & 1 & 0 & \cdots & 0 & 0 & 0 & \\ 0 & 0 & 1 & \cdots & 0 & 0 & 0 & \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \\ 0 & 0 & 0 & \cdots & 1 & 0 & 0 & \\ 0 & 0 & 0 & \cdots & 0 & 1 & 0 & \\ 0 & 0 & 0 & \cdots & 0 & 0 & 1 & \end{array}\right. \left(F_i^{s_i}\right)^T \left.\right] \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ \Gamma_{i-1}^{(j)} - \xi_j \\ 0 \\ \vdots \\ 0 \\ \hline \Gamma_i^{(1)} \\ \Gamma_i^{(2)} \\ \Gamma_i^{(3)} \\ \vdots \\ \Gamma_i^{(j)} \\ \vdots \\ \Gamma_i^{(n)} \end{bmatrix}$$

Proposition 1: For a given MSDNet, there exists a MSD-CSC model, which is equivalent to MSDNet when propagates with the layered thresholding algorithm.

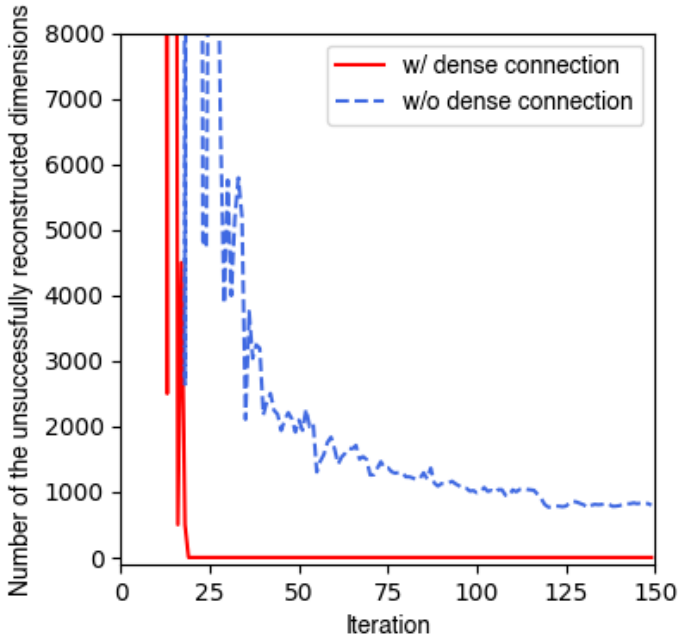
MSD-CSC and ML-CSC

Simulation study demonstrates that MSD-CSC
shows better reconstruction ability than ML-CSC



MSD-CSC and ML-CSC

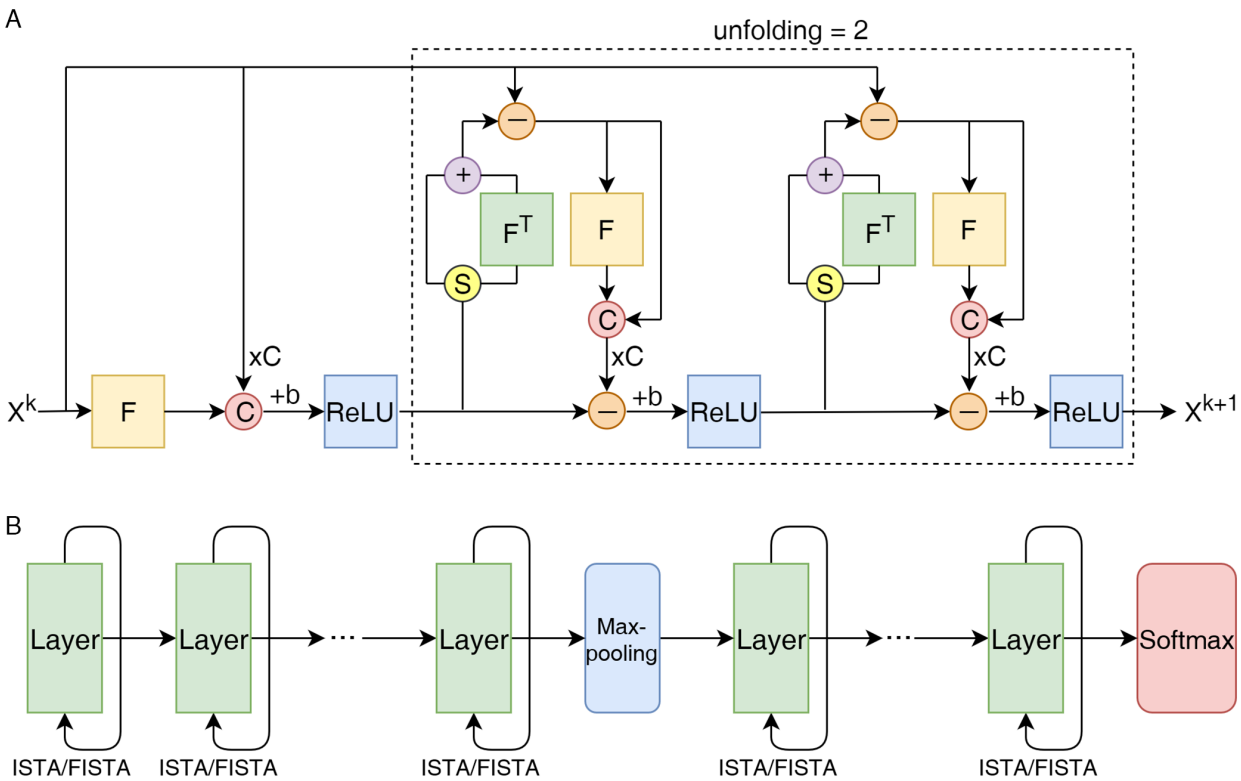
Simulation study demonstrates that MSD-CSC
shows better reconstruction ability than ML-CSC



Theorem 1: For the Lasso problem in each layer, the performance of MSD-CSC is better than that of ML-CSC.

ISTA for MSD-CSC

Unfold the iteration for MSD-CSC



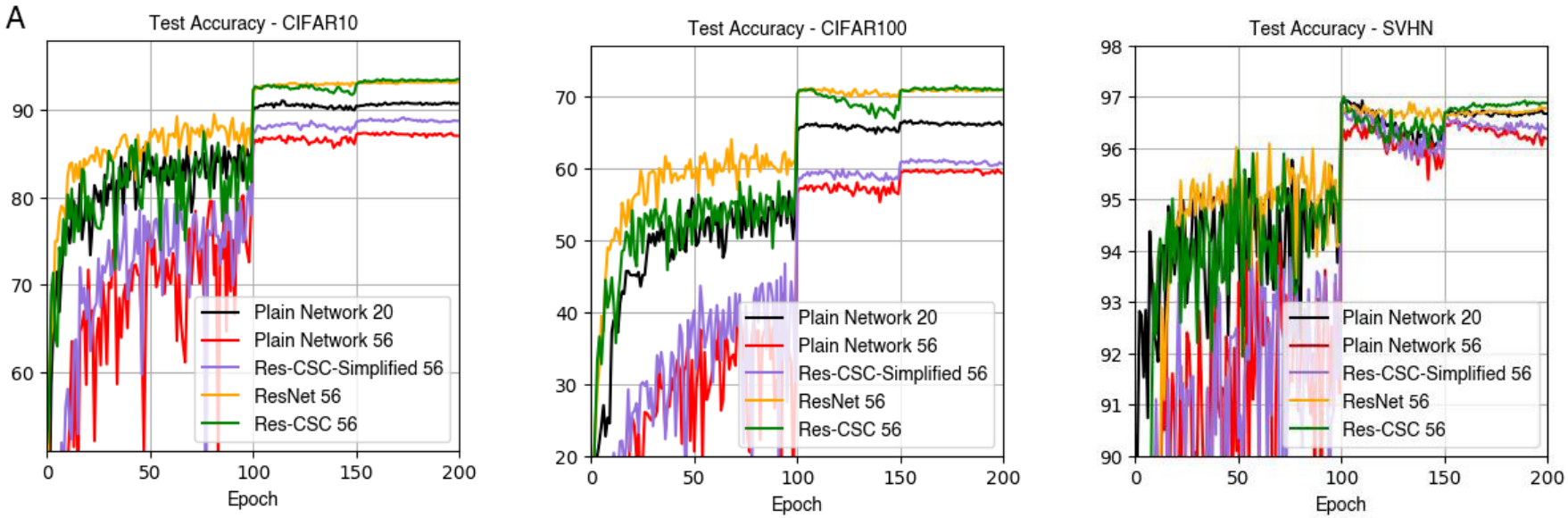
Comparison of CNN vs CSC

Table 1. The relationship between the generalized CNN and generalized CSC model.

CNN	CSC
The ith convolution with dilation scale s_i	The convolutional matrix $D_i^{s_i}$
Bias term	The balance coefficient β and $\lambda_{\max}(D^\top D)$
ReLU	Soft non-negative thresholding operator $S_\beta^+(\cdot)$
Feed-forward algorithm	$\Gamma^0 = 0$ in the update formula and iterate once
ResNet	$\Gamma^0 = X_{-1}$ in the update formula and iterate Equations (3) and (5) alternately
Dense connection	The identity matrix in $D_i^{s_i}$

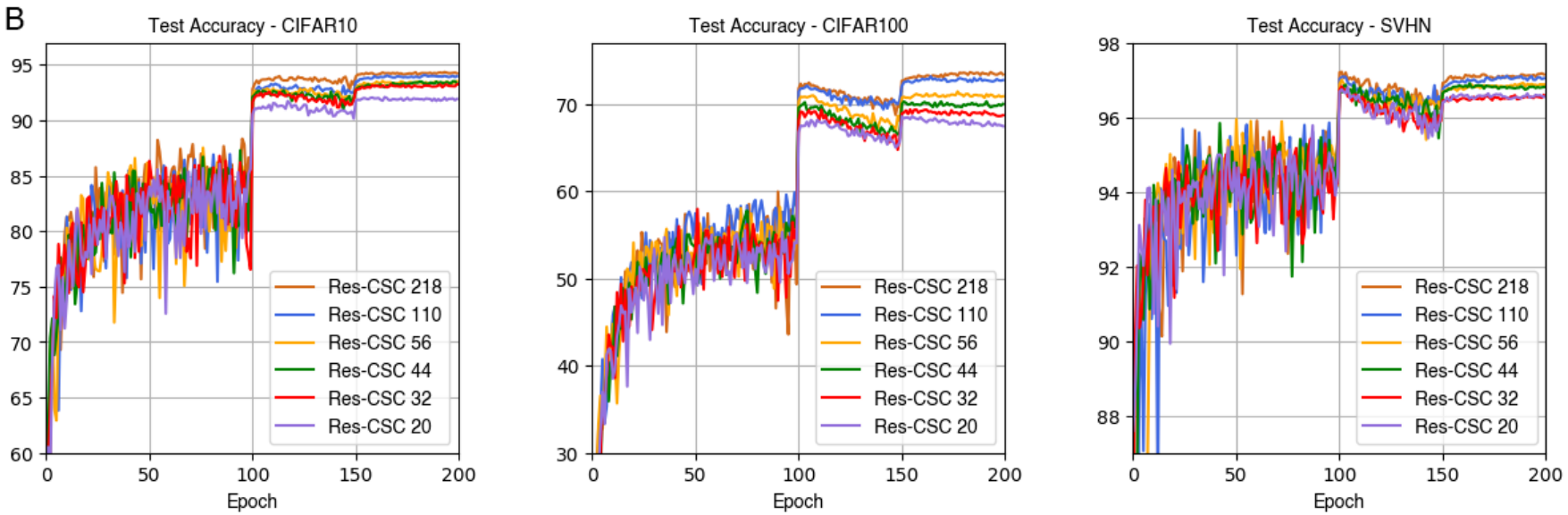
Performance of Res-CSC

Res-CSC indeed show equivalent performance



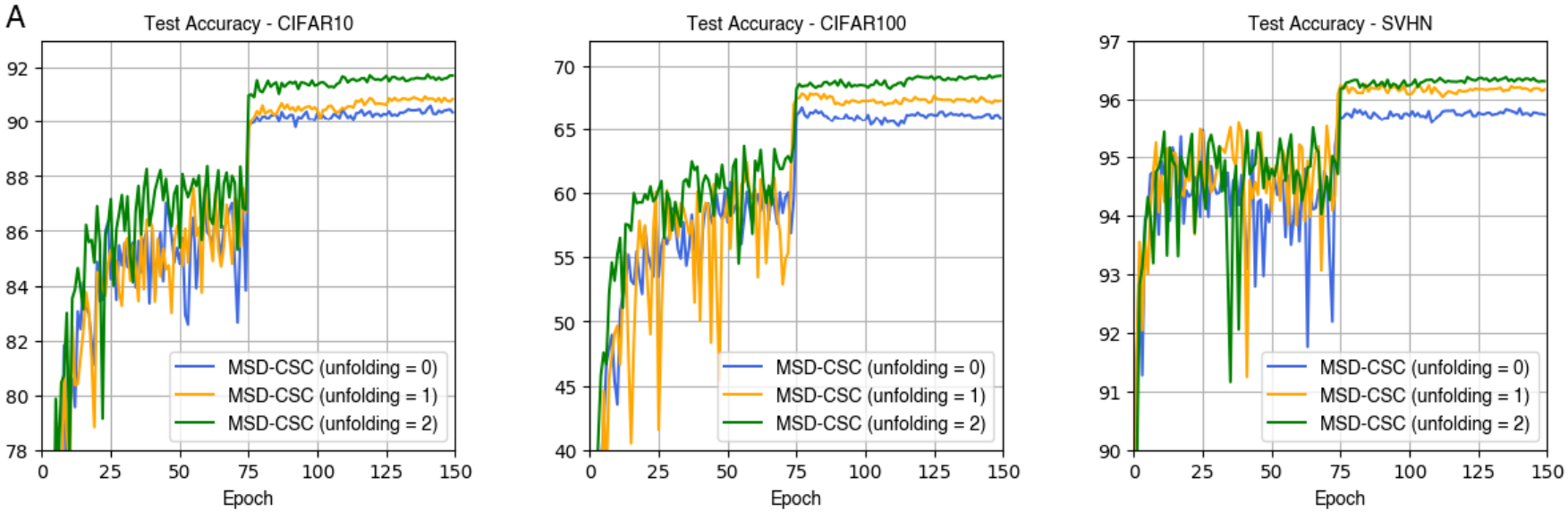
Performance of Res-CSC

Res-CSC can alleviate the degradation phenomenon



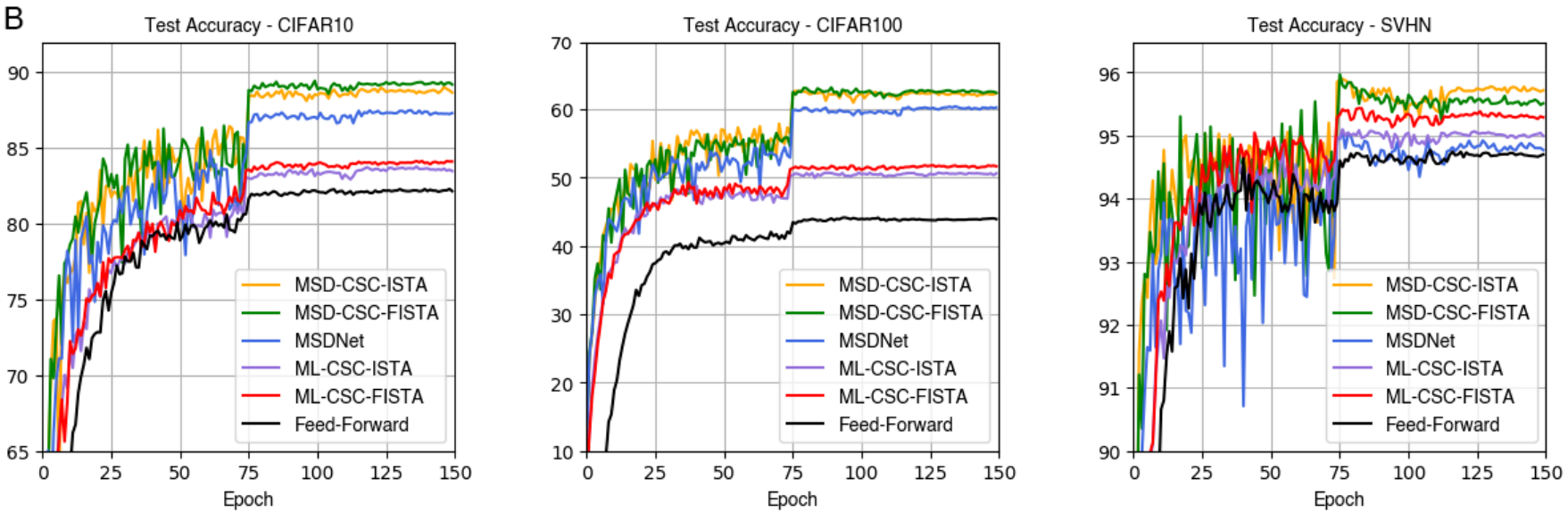
Performance of MSD-CSC

Unfolding indeed improve the performance of MSD-CSC



Performance of MSD-CSC

MSD-CSC shows better performance than MSDNet



Outline

Bridging Theoretical Insights with Practical Application

- CNNs can effectively serve as deep sparse coding solvers like the layered threshold algorithm.

$$X_i = S_{\beta_i}(D_i^\top X_{i-1}) \rightarrow X_i = \text{ReLU}(W_i X_{i-1} + b_i)$$

Bridging Theoretical Insights with Practical Application

- CNNs can effectively serve as deep sparse coding solvers like the layered threshold algorithm.

$$X_i = S_{\beta_i}(D_i^\top X_{i-1}) \rightarrow X_i = \text{ReLU}(W_i X_{i-1} + b_i)$$

- We generalize ML-CSC for advanced NNs like ResNet, MSDNet.
  - The initialization (Res-CSC $\rightarrow$ Res-Net)
  - The dictionary design (MSD-CSC$\rightarrow$ MSD-Net)

Bridging Theoretical Insights with Practical Application

- CNNs can effectively serve as deep sparse coding solvers like the layered threshold algorithm.

$$X_i = S_{\beta_i}(D_i^\top X_{i-1}) \rightarrow X_i = \text{ReLU}(W_i X_{i-1} + b_i)$$

- We generalize ML-CSC for advanced NNs like ResNet, MSDNet.
  - The initialization (Res-CSC $\rightarrow$ Res-Net)
  - The dictionary design (MSD-CSC$\rightarrow$ MSD-Net)
- Is it possible to combine sparse models with deep learning to achieve comparable performance when handling modern image datasets such as ImageNet?
  - Consider ISTA updates: $X_i^{l+1} = S_{\frac{\lambda}{L}}(X_i^l - \frac{1}{L} D_i^\top (D_i X_i^l - X_{i-1}))$.

Bridging Theoretical Insights with Practical Application

- CNNs can effectively serve as deep sparse coding solvers like the layered threshold algorithm.

$$X_i = S_{\beta_i}(D_i^\top X_{i-1}) \rightarrow X_i = \text{ReLU}(W_i X_{i-1} + b_i)$$

- We generalize ML-CSC for advanced NNs like ResNet, MSDNet.
  - The initialization (Res-CSC $\rightarrow$ Res-Net)
  - The dictionary design (MSD-CSC$\rightarrow$ MSD-Net)
- Is it possible to combine sparse models with deep learning to achieve comparable performance when handling modern image datasets such as ImageNet?
  - Consider ISTA updates: $X_i^{l+1} = S_{\frac{\lambda}{L}}(X_i^l - \frac{1}{L} D_i^\top (D_i X_i^l - X_{i-1}))$.

Principles work, only need to be implemented correctly. *–Prof. Yi Ma*

Sparse Dictionary Net (SDNet)

- Li *et al.* incorporate sparse modeling into a given existing network architecture (ResNet) by replacing first convolution layer with the CSC-layer to get Sparse Dictionary Net (SDNet) [Li et al., 2022].

Classical Conv. Layer

$$X_{i-1} \in \mathbb{R}^{M\times H\times W} \xrightarrow[D_i^\top \in \mathbb{R}^{C\times M\times k\times k}]{\text{Convolutional}} X_i \in \mathbb{R}^{C\times H\times W}$$

Convolutional Sparse Coding (CSC) Layer

$$X_{i-1} = D_i X_i$$

$$X_i = \operatorname{argmin}_X \lambda\|X\|_1 + \frac{1}{2}\|X_{i-1} - D_i X\|_2^2$$

$$X_{i-1} \in \mathbb{R}^{M\times H\times W} \xrightarrow[D_i \in \mathbb{R}^{M\times C\times k\times k}]{\text{FISTA}} X_i \in \mathbb{R}^{C\times H\times W}$$

Forward and Backward Propagation

- Forward propagation of the sparse coding layer is carried out by solving the optimization problem

$$X_i = \text{argmin}_X \lambda \|X\|_1 + \frac{1}{2}\|X_{i-1} - D_i X\|_2^2$$

Forward and Backward Propagation

- Forward propagation of the sparse coding layer is carried out by solving the optimization problem

$$X_i = \text{argmin}_X \lambda \|X\|_1 + \frac{1}{2}\|X_{i-1} - D_i X\|_2^2$$

- Carry out iteratively the following steps for $l \geqslant 1$ (FISTA Iteration) :

$$X_i^l = S_{\frac{\lambda}{L}}(Z_i^l - \frac{1}{L} D_i^\top (D_i Z_i^l - X_{i-1}))$$

$$m_i^{l+1} = \frac{1 + \sqrt{1 + 4(m_i^l)^2}}{2}$$

$$Z_i^{l+1} = X_i^l + \frac{m_i^l - 1}{m_i^{l+1}}(X_i^l - X_i^{l-1})$$

Forward and Backward Propagation

- Forward propagation of the sparse coding layer is carried out by solving the optimization problem

$$X_i = \text{argmin}_X \lambda \|X\|_1 + \frac{1}{2}\|X_{i-1} - D_i X\|_2^2$$

- Carry out iteratively the following steps for $l \geqslant 1$ (FISTA Iteration) :

$$X_i^l = S_{\frac{\lambda}{L}}(Z_i^l - \frac{1}{L} D_i^\top (D_i Z_i^l - X_{i-1}))$$

$$m_i^{l+1} = \frac{1 + \sqrt{1 + 4(m_i^l)^2}}{2}$$

$$Z_i^{l+1} = X_i^l + \frac{m_i^l - 1}{m_i^{l+1}}(X_i^l - X_i^{l-1})$$

- The FISTA iteration leads to an optimization-driven network. Then backward propagation can be carried out by auto-differentiation.

Robust Inference of SDNet

- CSC-layers enables us to design a robust inference strategy that cannot be achieved by classical explicit layers by using a λ that is proportional to the norm of the perturbation.

Theorem 11

Suppose Y has a representation DX, and that it is contaminated by noise E to create the input $Y' = Y + E$. Then as long as X is sufficiently sparse, the solution X_* to $\text{argmin}_X \lambda\|X\|_1 + \frac{1}{2}\|Y' - DX\|_2^2$ with $\lambda = O(\|E\|_2)$ satifies the suuport of X_* is contained in that of X and $\|X_* - X\|_2 = O(\|E\|_2)$.

Choose the optimal λ

- The authors present a practical technique for determining a proper choice of value λ based on residual $r_c := \|Y - DX\|_2$.

Algorithm 1 Robust inference with neural networks constructed from CSC-layers

Input: A network architecture with CSC-layers $f(\cdot; \boldsymbol{\theta}, \lambda_0)$, a (clean) training data $\mathcal{T}_{\text{train}}$, a (corrupted) test data $\mathcal{T}_{\text{test}}$, corruption type $\mathtt{T}$, a set $\mathcal{C}$ of corruption levels, a set Λ of values for λ.

 # Training the network
Train the network $f(\cdot; \boldsymbol{\theta}, \lambda_0)$ on $\mathcal{T}_{\text{train}}$ as described in Sec. 3.2 to obtain learned parameters $\boldsymbol{\theta}_\star$.
 # Fitting a relationship between optimal λ and the residual from CSC-layers using $\mathcal{T}_{train}$
for each noise level $c \in \mathcal{C}$ **do**
    Generate corrupted data $\mathcal{T}^c_{\text{train}}$ by injecting random noise of type $\mathtt{T}$ with level c to $\mathcal{T}_{\text{train}}$.
    Apply $f(\cdot; \boldsymbol{\theta}_\star, \lambda_0)$ on $\mathcal{T}^c_{\text{train}}$ and compute averaged residual from all CSC-layers as r_c.
    for each parameter $\lambda \in \Lambda$ **do**
        Apply $f(\cdot; \boldsymbol{\theta}_\star, \lambda)$ on $\mathcal{T}^c_{\text{train}}$ and compute averaged accuracy as a_λ.
    end for
    Set $\lambda_c = \arg\max_{\lambda \in \Lambda} a_\lambda$.
end for
Fit a function $\lambda := \lambda(r)$ from $\{\lambda_c, r_c\}_{c \in \mathcal{C}}$ via linear least squares.
 # Computing the residual from CSC-layers on $\mathcal{T}_{test}$
Apply $f(\cdot; \boldsymbol{\theta}_\star, \lambda_0)$ on $\mathcal{T}_{\text{test}}$ and compute averaged residual from all CSC-layers as r_{test}.

Output: Predicted labels on $\mathcal{T}_{\text{test}}$ with the network $f(\cdot; \boldsymbol{\theta}_\star, \lambda(r_{\text{test}}))$.

Experiments

- Compare the method with standard network architectures ResNet-18 and ResNet-34.
- Use the network architectures with the first convolutional layer of ResNet-18 and ResNet-34 replaced by CSC-layers.
- Refer to these networks as SDNet-18 and SDNet-34.

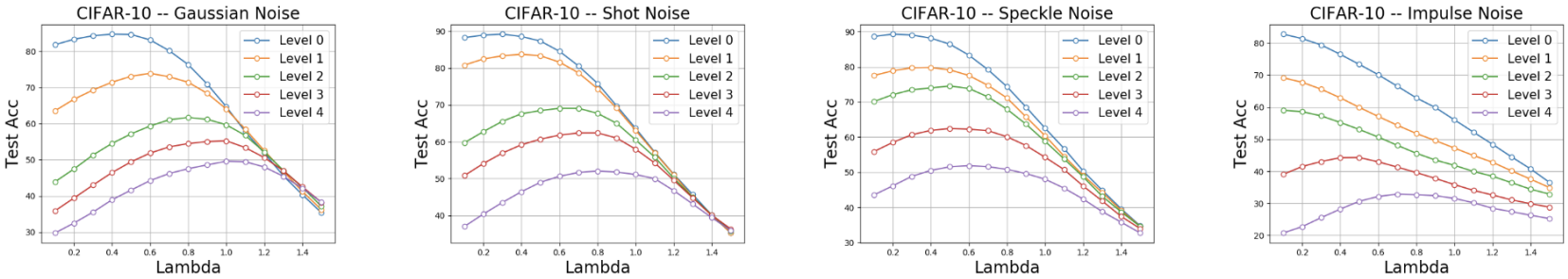
Performance for Image Classification

SDNet produces a Top-1 accuracy closely matches or surpasses ResNet while having a comparable speed.

Dataset	Architecture	Model Size	Top-1 Acc	Memory	Speed
CIFAR-10	ResNet-18 [21]	11.2M	95.54%	1.0 GB	1600 n/s
	ResNet-34 [21]	21.1M	95.57%	2.0 GB	1000 n/s
	MDEQ [27]	11.1M	93.80%	2.0 GB	90 n/s
	SCN [15]	0.7M	94.36%	10.0GB	39 n/s
	SCN-18	11.2M	95.12%	3.5 GB	158 n/s
	SDNet-18 (ours)	11.2M	95.20%	1.2 GB	1500 n/s
	SDNet-34 (ours)	21.1M	95.57%	2.4 GB	900 n/s
CIFAR-100	ResNet-18 [21]	11.2M	77.82%	1.0 GB	1600 n/s
	ResNet-34 [21]	21.1M	78.39%	2.0 GB	1000 n/s
	MDEQ [27]	11.2M	74.12%	2.0 GB	90 n/s
	SCN [15]	0.7M	80.07%	10.0GB	39 n/s
	SCN-18	11.2M	78.59%	3.5 GB	158 n/s
	SDNet-18 (ours)	11.3M	78.31%	1.2 GB	1500 n/s
	SDNet-34 (ours)	21.2M	78.48%	2.4 GB	900 n/s
ImageNet	ResNet-18 [21]	11.7M	68.98%	24.1 GB	2100 n/s
	ResNet-34 [21]	21.5M	72.83%	32.3 GB	1400 n/s
	SCN [15]	9.8M	70.42%	95.1 GB	51 n/s
	SDNet-18 (ours)	11.7M	69.47%	37.6 GB	1800 n/s
	SDNet-34 (ours)	21.5M	72.67%	46.4 GB	1200 n/s

Robustness as a function of λ

The peak performance w.r.t λ increases monotonically with the severity level of corruption.



(a) Gaussian Noise (b) Shot Noise (c) Speckle Noise (d) Impulse Noise

Figure 2: Test accuracy of SDNet-18 trained on CIFAR-10 dataset with $\lambda = 0.1$ and evaluated on 4 types of additive noise from CIFAR-10-C [53] in 5 severity levels each with varying values of λ. For each corruption type, optimal value of λ for testing increases monotonically with the severity level.

Robustness to Adversarial Perturpations

SDNet also exhibits robustness to adversarial perturbations by tuning the parameter λ.

Table 4: Robust accuracy on CIFAR-10 with adversarial perturbation using PGD attack.

Model	Robust Accuracy ($L_\infty = 8/255$)	Robust Accuracy ($L_2 = 0.5$)
ResNet-18 [21]	0.01%	29.47%
SDNet-18 w/ $\lambda = 0.1$	0.11%	29.95%
SDNet-18 (After tuning λ)	35.18%	62.80%

Outline

Summary

- CNN lead to remarkable results in many fields.
- ResNet and MSDNet have even more superior performance.
  - Clear and profound theoretical understanding is still lacking.

Summary

- CNN lead to remarkable results in many fields.
- ResNet and MSDNet have even more superior performance.
  - Clear and profound theoretical understanding is still lacking.
- Sparse coding is a powerful model
  - Enjoys from a vast theoretical study, supporting its success.
  - CSC and ML-CSC have been proposed recently.

Summary

- CNN lead to remarkable results in many fields.
- ResNet and MSDNet have even more superior performance.
  - Clear and profound theoretical understanding is still lacking.
- Sparse coding is a powerful model
  - Enjoys from a vast theoretical study, supporting its success.
  - CSC and ML-CSC have been proposed recently.
- Res-CSC and MSD-CSC have been proposed here!!
  - Res-CSC/MSD-CSC can be equivalent with ResNet/MSDNet.
  - All Residual, Dilation and Dense operations can be explained.
  - Optimization in each layer can be improved with unfolding.

Summary

- CNN lead to remarkable results in many fields.
- ResNet and MSDNet have even more superior performance.
  - Clear and profound theoretical understanding is still lacking.
- Sparse coding is a powerful model
  - Enjoys from a vast theoretical study, supporting its success.
  - CSC and ML-CSC have been proposed recently.
- Res-CSC and MSD-CSC have been proposed here!!
  - Res-CSC/MSD-CSC can be equivalent with ResNet/MSDNet.
  - All Residual, Dilation and Dense operations can be explained.
  - Optimization in each layer can be improved with unfolding.
- The sparse modeling can be combined with deep learning.
  - SDNet obtains performance on par with standard ConvNets but with better layer-wise interpretability and stability.

Discussion

- Attention mechanism is a widely used technique in deep learning.
- It is found that scaled dot-product attention with positional encoding can approximate any convolutional neural networks [Cordonnier et al., 2020].

Discussion

- Attention mechanism is a widely used technique in deep learning.
- It is found that scaled dot-product attention with positional encoding can approximate any convolutional neural networks [Cordonnier et al., 2020].

- (Question) Does deep sparse feature extraction ability widely exists in deep learning architecture?

Discussion

- Attention mechanism is a widely used technique in deep learning.
- It is found that scaled dot-product attention with positional encoding can approximate any convolutional neural networks [Cordonnier et al., 2020].

- (Question) Does deep sparse feature extraction ability widely exists in deep learning architecture?
- (Question) Can we leverage sparsity to further design more interpretable neural networks? Further Reading [Yu et al., 2023].

References I

Cordonnier, J.-B., Loukas, A., and Jaggi, M. (2020).

On the relationship between self-attention and convolutional layers.

In *International Conference on Learning Representations*.

Donoho, D. L., Elad, M., and Temlyakov, V. N. (2005).

Stable recovery of sparse overcomplete representations in the presence of noise.

IEEE Transactions on information theory, 52(1):6–18.

He, K., Zhang, X., Ren, S., and Sun, J. (2016).

Deep residual learning for image recognition.

In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Li, J., Feng, H., and Zhou, D.-X. (2024).

Convergence analysis for deep sparse coding via convolutional neural networks.

arXiv preprint arXiv:2408.05540.

Li, M., Zhai, P., Tong, S., Gao, X., Huang, S.-L., Zhu, Z., You, C., Ma, Y., et al. (2022).

Revisiting sparse convolutional model for visual recognition.

Advances in Neural Information Processing Systems, 35:10492–10504.

Olshausen, B. A. and Field, D. J. (1996).

Emergence of simple-cell receptive field properties by learning a sparse code for natural images.

Nature, 381(6583):607–609.

Papyan, V., Romano, Y., and Elad, M. (2017a).

Convolutional neural networks analyzed via convolutional sparse coding.

The Journal of Machine Learning Research, 18(1):2887–2938.

References II

Papyan, V., Sulam, J., and Elad, M. (2017b).

Working locally thinking globally: Theoretical guarantees for convolutional sparse coding.
IEEE Transactions on Signal Processing, 65(21):5687–5701.

Pelt, D. M. and Sethian, J. A. (2018).

A mixed-scale dense convolutional neural network for image analysis.
Proceedings of the National Academy of Sciences, 115(2):254–259.

Rubinstein, R., Zibulevsky, M., and Elad, M. (2009).

Double sparsity: Learning sparse dictionaries for sparse signal approximation.
IEEE Transactions on signal processing, 58(3):1553–1564.

Sulam, J. and Elad, M. (2015).

Expected patch log likelihood with a sparse prior.
In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 99–111. Springer.

Yu, Y., Buchanan, S., Pai, D., Chu, T., Wu, Z., Tong, S., Haeffele, B. D., and Ma, Y. (2023).

White-box transformers via sparse rate reduction.
In *Thirty-seventh Conference on Neural Information Processing Systems*.

Zeiler, M., Krishnan, D., Taylor, G., and Fergus, R. (2011).

Deconvolutional networks for feature learning.
In *Comput. Vis. Pattern Recognit.(CVPR), 2010 IEEE Conf*, pages 2528–2535. Citeseer.

Zhang, Z. and Zhang, S. (2021).

Towards understanding residual and dilated dense neural networks via convolutional sparse coding.
National Science Review, 8(3):nwaa159.