Sparse Coding, Artificial Neural Networks, and the Brain: Toward "Substantive Intelligence"

Demba Ba[1,2]
Assoc. Prof. of Electrical Engineering and Bioengineering

[1]Harvard University
School of Engineering and Applied Sciences (SEAS)
[2]Institute for AI and Fundamental Interactions (IAIFI)

MIT, April 15^{th} 2021

HARVARD
JOHN A. PAULSON
SCHOOL OF ENGINEERING
AND APPLIED SCIENCES

Outline

The many faces of sparse coding

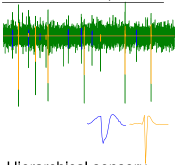
Interpretable neural networks for source separation/sparse dictionary learning

Deep sparse coding and hierarchical sensory processing

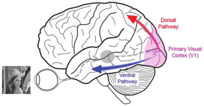
Neuroscience, ReLU nets and sparse coding

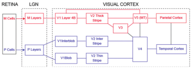
Unsupervised learning to decipher neural code

Blind source separation



Hierarchical sensory processing principles

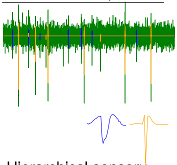




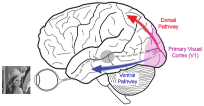
Neuroscience, ReLU nets and sparse coding

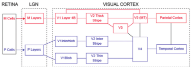
Unsupervised learning to decipher neural code

Blind source separation

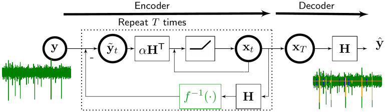


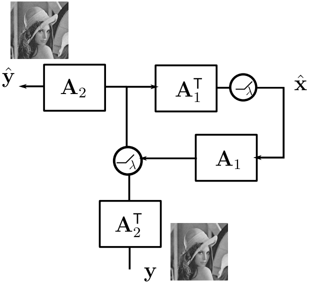
Hierarchical sensory processing principles



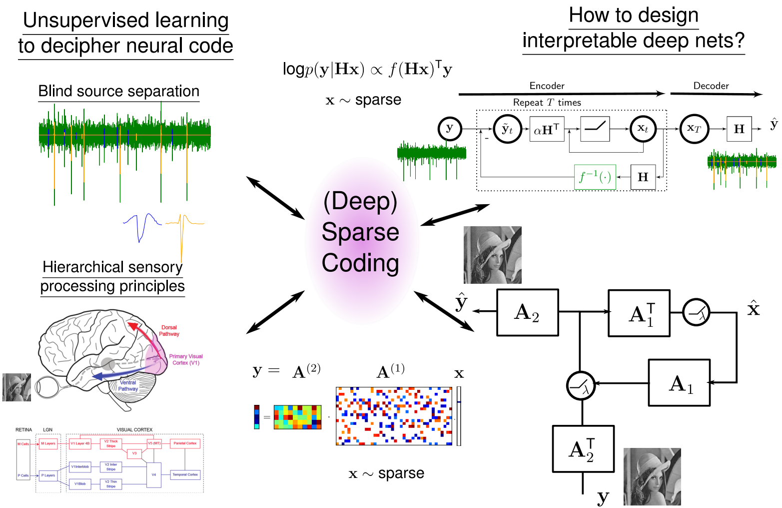


How to design interpretable deep nets?

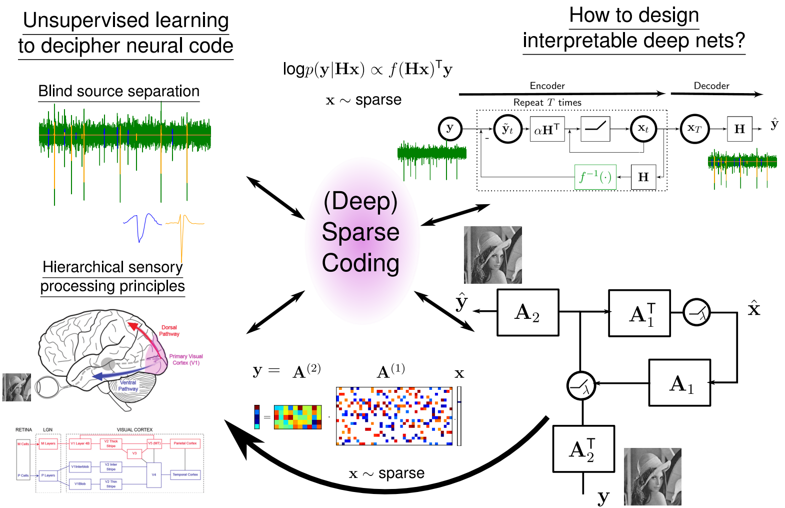




Neuroscience, ReLU nets and sparse coding



Neuroscience, ReLU nets and sparse coding



Unique opportunty to use deep learning, *in a principled fashion*.

Outline

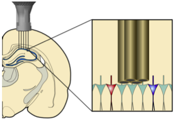
The many faces of sparse coding

Interpretable neural networks for source separation/sparse dictionary learning

Deep sparse coding and hierarchical sensory processing

Electrophysiological recordings of neural spiking

Tetrodes: Four electrodes “listen” to ≈ 10 neighboring neurons (circa ≈ 2000)



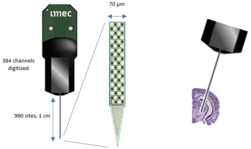
Data rate

32-bit time series

10 kHz per second

≈ 1 Gb per 6 hours!

Neuropixels: Hundreds of electrodes “listen” to ≈ 100 neighboring neurons (circa ≈ 2020)



Data rate

14-bit time series

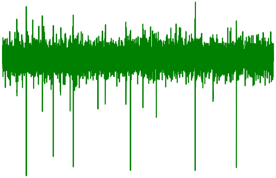
10 kHz per second

≈ 400 Gb per hour!

Goal: identify neural sources (spike sorting).

Auto-encoders for spike sorting

Electrophysiology



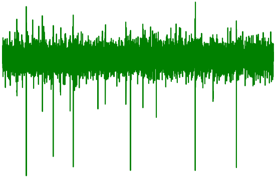
Sparse Coding

$$\mathbf{y} = [\mathbf{H}_1 | \cdots | \mathbf{H}_C] \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_C \end{bmatrix} + \mathbf{v} = \mathbf{H}\mathbf{x} + \mathbf{v}$$

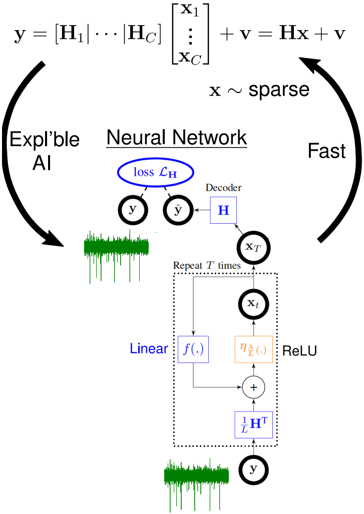
$\mathbf{x} \sim$ sparse

Auto-encoders for spike sorting

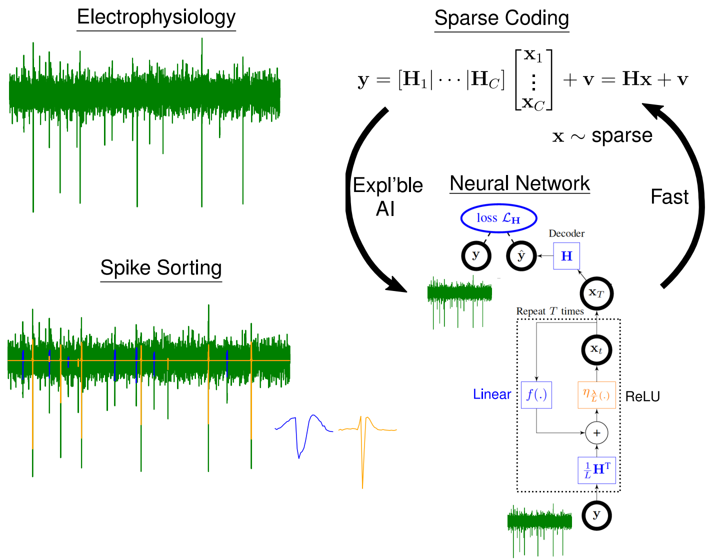
Electrophysiology



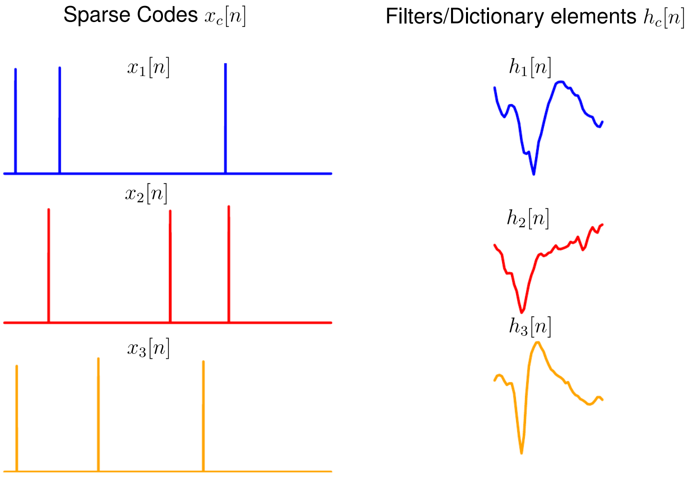
Sparse Coding



Auto-encoders for spike sorting



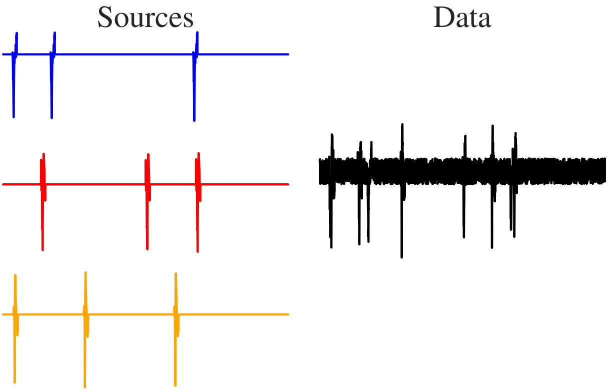
Blind source separation by dictionary learning



Convolutional generative model

The generative model for shift-invariant sparse representation:

$$y_n = \sum_{c=1}^{C} h_c[n] * x_c[n] + v_n$$



Convolutional generative model

The generative model for shift-invariant sparse representation:

$$y_n = \sum_{c=1}^{C} h_c[n] * x_c[n] + v_n$$

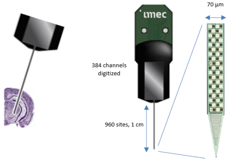
(Linear-algebraic form) $\quad \mathbf{y} = \left[\mathbf{H}_1 | \cdots | \mathbf{H}_C\right] \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_C \end{bmatrix} + \mathbf{v} = \mathbf{H}\mathbf{x} + \mathbf{v}$

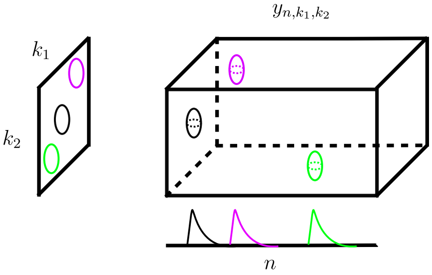
Given only the data $\mathbf{y}$, how to solve for $\mathbf{H}$ and $\mathbf{x}$?

Spatio-temporal generalizations of convolutional model

$$y_{n,k_1,k_2} = \sum_{c=1}^{C} \sum_{i=1}^{N_c} x_c[n, k_1, k_2] * h_c[n] g_c[k_1, k_2] + v_{n,k_1,k_2}$$

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{v}$$





Pervasiveness of $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{v}$ model

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{v}$$

Discipline	$\mathbf{y}$	$\mathbf{H}$	$\mathbf{x}$	Conv?
Neuro	Electrode array	Neurons	event times/amplitudes	Y
Acoustics	Mic. array	Sound source	event times/amplitudes	Y
Astro	Tel. array	Celest. objects	location/intensity	Y
Part. physics	Detector observ. (mixture)	Particle “topics”	Proportions	N
Optics	Intensity	Scattering matrix	Object	N/Y
Genomics	Cell gene expr.	Gene expr. modules	Gene expr. levels	N
Text	Word hist.	Topic hist.	Topic proportions	N
Radar	Antenna array	Sources	Locations	Y
...				

Optimization perspective

Given J examples of data $\{\mathbf{y}^j\}_{j=1}^J$,

CDL solves:

$$\min_{(\mathbf{x}^j)_{j=1}^J,(\mathbf{h}_c)_{c=1}^C} \sum_{j=1}^J \frac{1}{2} \left|\left|\mathbf{y}^j - \mathbf{H}\mathbf{x}^j\right|\right|_2^2 + \lambda \left|\left|\mathbf{x}^j\right|\right|_1$$

$$\text{s.t. } \left|\left|\mathbf{h}_c\right|\right|_2 \leq 1 \quad \text{for } c = 1, \cdots, C,$$

where $\lambda > 0$ (regularization parameter enforcing sparsity).

Convolutional sparse coding step

Given the filters,

CSC is separable over J examples

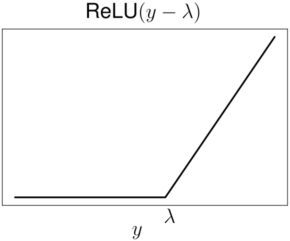
$$\min_{\mathbf{x}^j} \frac{1}{2} \left|\left|\mathbf{y}^j - \mathbf{H}\mathbf{x}^j\right|\right|_2^2 + \lambda \left|\left|\mathbf{x}^j\right|\right|_1$$

This step is

- Embarrassingly parallelizable.
- Amenable to GPU processing (long recordings).

ReLU nonlinearity from sparse approximation in 1D

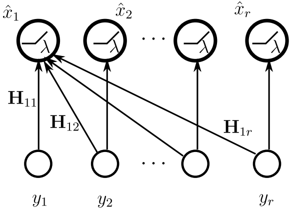
$$\begin{aligned}\hat{x} &= \arg\min_{x \in \mathbb{R}^+} \frac{1}{2}(y-x)^2 + \lambda |x| \\ &= \mathsf{max}(y-\lambda, 0) \\ &= \mathsf{ReLU}(y-\lambda).\end{aligned}$$



Neural net with ReLU from sparse approximation

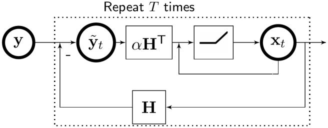
Case 1: $\mathbf{H}^\mathsf{T}\mathbf{H} = \mathbf{I}$

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathbb{R}^r} \frac{1}{2}\,||\mathbf{y} - \mathbf{H}\mathbf{x}||_2^2 + \lambda\,||\mathbf{x}||_1 = \mathsf{ReLU}\left(\mathbf{H}^\mathsf{T}\mathbf{y} - \lambda\right).$$



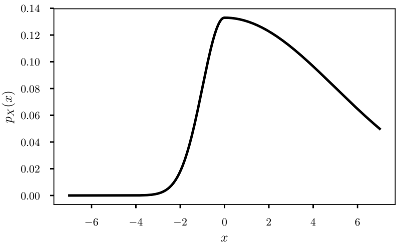
Case 2: $\mathbf{H}^\mathsf{T}\mathbf{H} \neq \mathbf{I}$, $\sigma_{\mathsf{max}}(\mathbf{H}^\mathsf{T}\mathbf{H}) \leq L$

$$\mathbf{x}_t = \mathsf{ReLU}\left(\mathbf{x}_{t-1} + \alpha\mathbf{H}^\mathsf{T}(\mathbf{y} - \mathbf{H}\mathbf{x}_{t-1}) - \frac{\lambda}{L}\right)$$

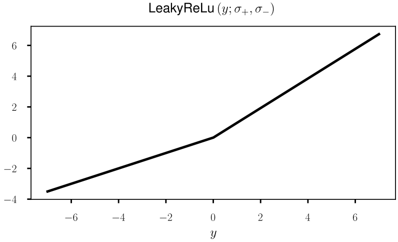


Other nonlinearities?

$$\log p_X(x) \propto -\frac{x^2}{2\sigma_-^2}\mathbf{1}_{\{x\le 0\}} - \frac{x^2}{2\sigma_+^2}\mathbf{1}_{\{x\ge 0\}}$$



$$\begin{aligned}\hat{x} &= \arg\min_{x\in\mathbb{R}} \frac{1}{2}(y-x)^2 - \log p_X(x) = \\ &= \frac{\sigma_-^2}{1+\sigma_-^2} y \mathbf{1}_{\{y\le 0\}} + \frac{\sigma_+^2}{1+\sigma_+^2} y \mathbf{1}_{\{y\ge 0\}} \\ &= \mathrm{LeakyReLu}\,(y; \sigma_+, \sigma_-)\,.\end{aligned}$$



Convolutional dictionary update Step

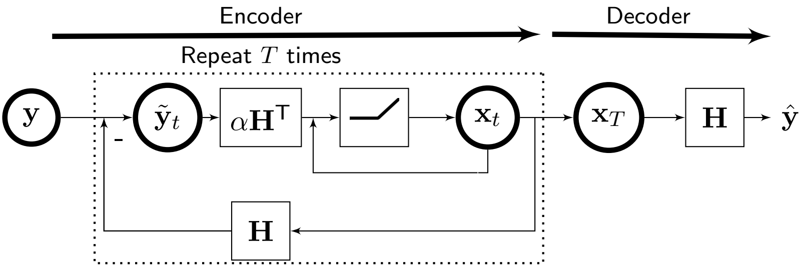
$$\min_{(\mathbf{h}_c)_{c=1}^{C}} \sum_{j=1}^{J} \frac{1}{2} \left|\left|\mathbf{y}^j - \mathbf{H}\mathbf{x}^j\right|\right|_2^2 \text{ s.t. } ||\mathbf{h}_c||_2 \leq 1 \quad \text{for } c = 1, \cdots, C.$$

This step is

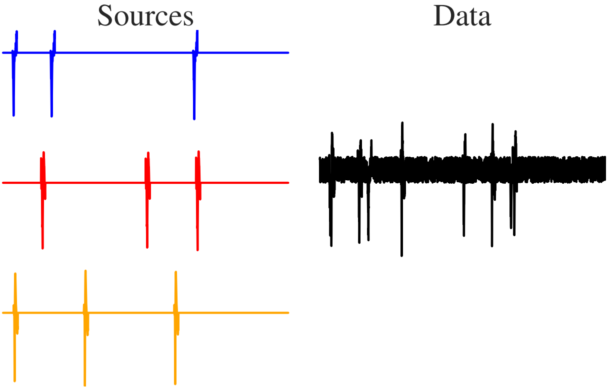
- Computationally Expensive.
- Not parallelizable over J examples.

Auto-encoder for CDL by deep unfolding and weight tying

1. Nonlinear encoder:
   $\mathbf{x}_t = \text{ReLU}\left(\mathbf{x}_{t-1} + \alpha \mathbf{H}^\mathsf{T}(\mathbf{y} - \mathbf{H}\mathbf{x}_{t-1}) - \frac{\lambda}{L}\right)$.
2. Linear decoder: $\hat{\mathbf{y}} = \mathbf{H}\mathbf{x}_T$.
3. Training: backprop with MSE loss.



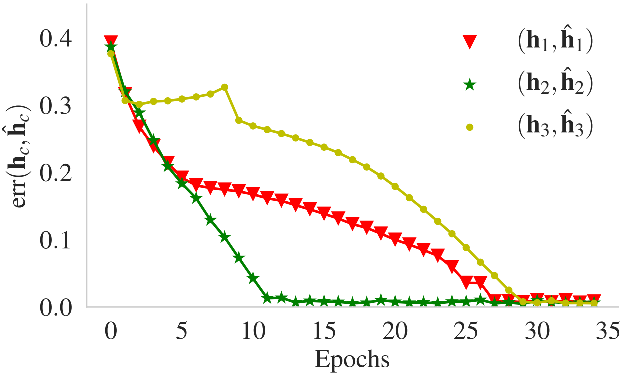
Simulated data



Data: 17 minutes of electrical activity from 3 neurons!
Sampling rate: $f_s = 10$ kHz.
Firing rate: 30 Hz.

AE performs dictionary learning

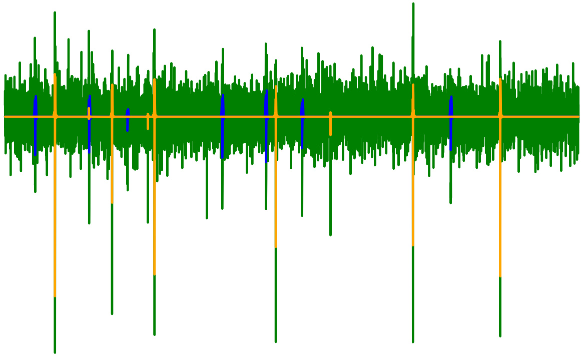
$$\mathrm{err}(\mathbf{h}_c, \hat{\mathbf{h}}_c) = \sqrt{1 - \frac{\langle \mathbf{h}_c, \hat{\mathbf{h}}_c \rangle^2}{\|\mathbf{h}_c\|_2^2 \|\hat{\mathbf{h}}_c\|_2^2}}$$



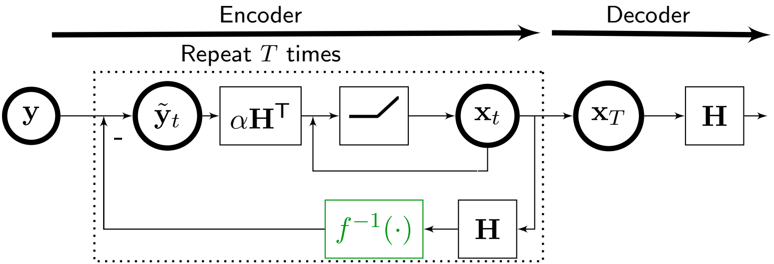
Training of AE is fast

		CRsAE	Sporco	CBP
Learning Spike Shapes	runtime	**69.27 s**	319.52 s	
	iterations	**10**	89	
Spike Sorting	runtime	**0.93 s**		17 hours

Harris dataset: spike sorting and denoising



Unfolded auto-encoders for exponential family data



Components for different distributions
($\mathcal{S}_b(\cdot) = \text{ReLU}(\cdot - b)$, b depends on λ)

	$\mathbf{y}$	$f^{-1}(\cdot)$	Encoder Unfolding ($\mathbf{x}_t$)	Decoder ($f(\hat{\boldsymbol{\mu}})$)
Gaussian	$\mathbb{R}$	$I(\cdot)$	$\mathcal{S}_b\left(\mathbf{x}_{t-1} + \alpha\mathbf{H}^{\mathsf{T}}\widetilde{\mathbf{y}}_t\right)$	$\mathbf{H}\mathbf{x}_T$
Binomial	$[0..M]$	sigmoid$(\cdot)$	$\mathcal{S}_b\left(\mathbf{x}_{t-1} + \alpha\mathbf{H}^{\mathsf{T}}(\frac{1}{M}\widetilde{\mathbf{y}}_t)\right)$	$\mathbf{H}\mathbf{x}_T$
Poisson	$[0..\infty)$	$\exp(\cdot)$	$\mathcal{S}_b\left(\mathbf{x}_{t-1} + \alpha\mathbf{H}^{\mathsf{T}}(\text{Elu}(\widetilde{\mathbf{y}}_t))\right)$	$\mathbf{H}\mathbf{x}_T$

Auto-encoders for denoising Poisson images

Low photon count image

Denoised image

Poisson Convolutional Model

$$\mathbf{y} \sim \text{Poisson}(\boldsymbol{\lambda})$$

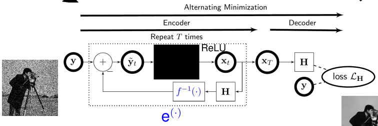
$$\log\boldsymbol{\lambda} = [\mathbf{H}_1|\cdots|\mathbf{H}_C]\begin{bmatrix}\mathbf{x}_1\\ \vdots \\ \mathbf{x}_C\end{bmatrix} = \mathbf{H}\mathbf{x}$$

$$\mathbf{x} \sim \text{sparse}$$

Expl'ble AI

Fast

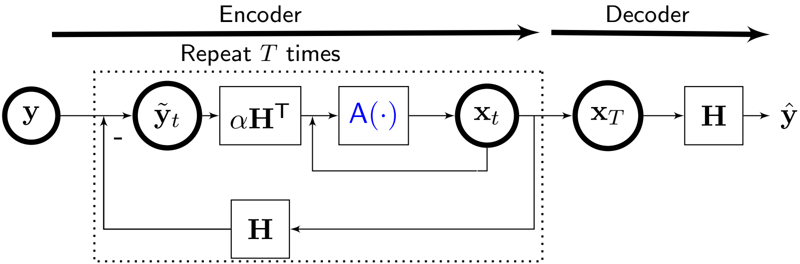
Neural Network



		SPDA	CA	DCEA-C	DCEA-UC
Peak 1	Set12	20.39	**21.51**	20.72	21.37
	BSD68	.	21.78	21.27	**21.84**
Peak 2	Set12	21.70	**22.97**	22.02	22.79
	BSD68	.	22.90	22.31	**22.92**
Peak 4	Set12	22.56	**24.40**	23.51	24.37
	BSD68	.	23.98	23.54	**24.10**
# of Params		160K	655K	20K	61K

Tolooshams et al., ICML 2020

Unfolded auto-encoders with different priors

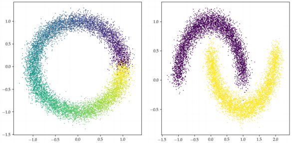


Activation functions for different priors, for (input $\tilde{\mathbf{x}}_t = \mathbf{x}_{t-1} + \alpha\mathbf{H}^\mathsf{T}\widetilde{\mathbf{y}}_t$)

Prior	Constraints	Expression	Activation
Gaussian/Ridge	None	$\frac{1}{\sigma^2}\sum x_j^2$	$\frac{\sigma^2}{1/L+\sigma^2}\tilde{\mathbf{x}}_t = c(\sigma)\tilde{\mathbf{x}}_t$
Sparsity	$\mathbf{x} \geq 0$	$\lambda \sum x_j$	$\mathsf{ReLU}(\tilde{\mathbf{x}}_t - \frac{\lambda}{L})$
2-sided Gaussian	None	$\sum \frac{x_j^2}{\sigma_+^2}\mathbf{1}_{\{x_j \geq 0\}} + \frac{x_j^2}{\sigma_-^2}\mathbf{1}_{\{x_j \leq 0\}}$	$c(\sigma_+)\tilde{\mathbf{x}}_t\mathbf{1}_{\{\tilde{\mathbf{x}}_t \geq 0\}} + c(\sigma_-)\tilde{\mathbf{x}}_t\mathbf{1}_{\{\tilde{\mathbf{x}}_t \leq 0\}}$
Group sparsity	$\mathbf{x} \geq 0$	$\sum \|x_g\|_2$	$\left(\tilde{\mathbf{x}}_{t,g} - \lambda\frac{\tilde{\mathbf{x}}_{t,g}}{\|\tilde{\mathbf{x}}_{t,g}\|_2}\right)\mathbf{1}_{\{\|\tilde{\mathbf{x}}_{t,g}\|_2 \geq \lambda\}}$
Simplex	$\mathbf{x} \geq 0, \mathbf{x}^T\mathbf{1} = 1$	None	$\mathsf{ReLU}(\tilde{\mathbf{x}}_t - b(\tilde{\mathbf{x}}_t))$
...			

Auto-encoders for manifold learning

Nonlinear manifold learning



Sparse Coding on *simplices*

$$\min_{\mathbf{x}\in\mathbb{R}^{m\times n}} \mathcal{L}(\mathbf{A},\mathbf{y},\mathbf{x}) = \frac{1}{2}\|\mathbf{y}-\mathbf{A}\mathbf{x}\|^2 + \lambda\sum_{j=1}^{m} x_j\|\mathbf{y}-\mathbf{a}_j\|^2$$

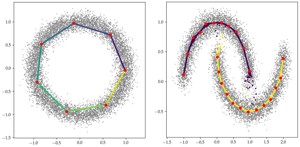
$$\mathbf{x}^\top\mathbf{1} = \mathbf{1},$$

$$x_j \geq 0, \quad \text{for all } j.$$

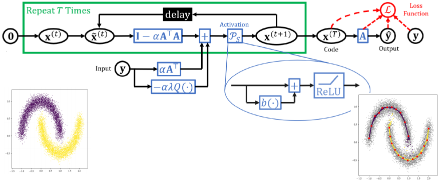
Expl'ble AI

Fast

Piecewise-Linear Manifold *Approximation*



Neural Network



Outline

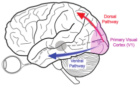
The many faces of sparse coding

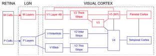
Interpretable neural networks for source separation/sparse dictionary learning

Deep sparse coding and hierarchical sensory processing

Deep sparse coding, hierarchical "virtual" brains

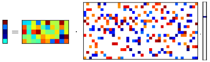
Natural images/Hierar. rep.





Deep Sparse Coding
Generative Model

$\mathbf{y} = \mathbf{A}^{(2)} \quad \mathbf{A}^{(1)} \quad \mathbf{x}$

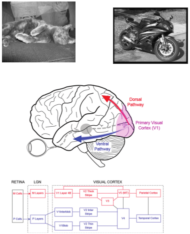


$\mathbf{x} \sim$ sparse
$\mathbf{x}_2 = \mathbf{A}^{(1)}\mathbf{x} \sim$ sparse

Ba, IEEE TSP 2020

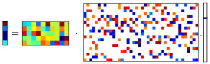
Deep sparse coding, hierarchical “virtual” brains

Natural images/Hierar. rep.



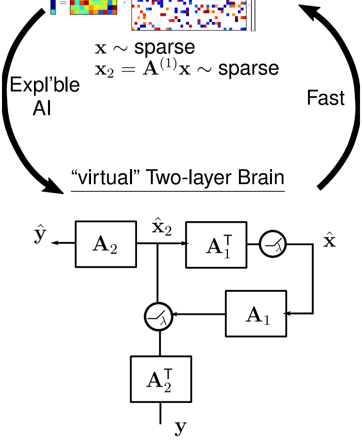
Deep Sparse Coding Generative Model

$\mathbf{y} = \mathbf{A}^{(2)} \quad \mathbf{A}^{(1)} \quad \mathbf{x}$



$\mathbf{x} \sim$ sparse

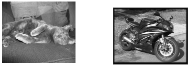
$\mathbf{x}_2 = \mathbf{A}^{(1)}\mathbf{x} \sim$ sparse

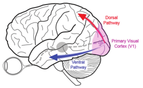


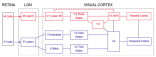
Ba, IEEE TSP 2020

Deep sparse coding, hierarchical "virtual" brains

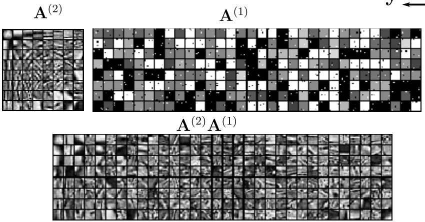
Natural images/Hierar. rep.



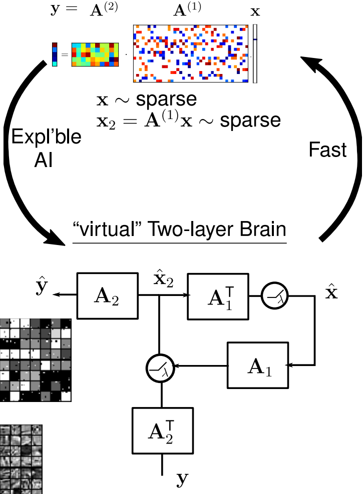




Learned filters



Deep Sparse Coding
Generative Model

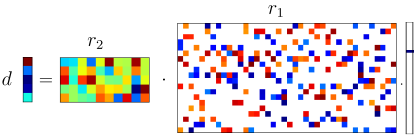


Ba, IEEE TSP 2020

Deep sparse coding theory guides architecture design

Deeply-sparse coding model

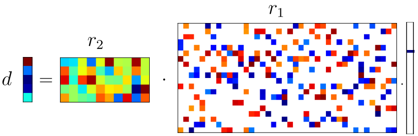
$$\mathbf{y} = \mathbf{A}^{(2)} \quad \mathbf{A}^{(1)} \quad \mathbf{x}$$



Deep sparse coding theory guides architecture design

Deeply-sparse coding model

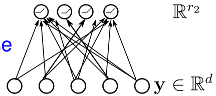
$\mathbf{y} = \quad \mathbf{A}^{(2)} \qquad \mathbf{A}^{(1)} \qquad \mathbf{x}$



1. Pick r_2, # units in 1st hidden layer.

Associated AE

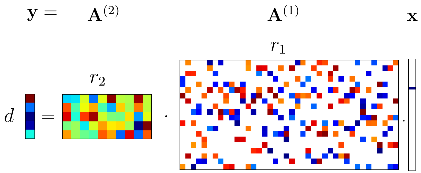
$\mathbf{A}^{(2)\mathsf{T}} \sim$ Dense



Deep sparse coding theory guides architecture design

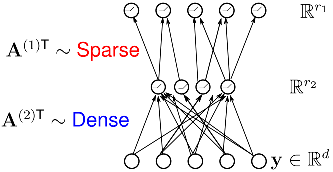
Deeply-sparse coding model

Associated AE



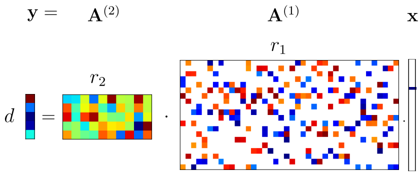
1. Pick r_2, # units in 1st hidden layer.

2. $r_1 = \mathcal{O}\left(\max(r_2^2, r_2 s_{\mathbf{A}^{(1)}})\right)$, i.e. expansion.



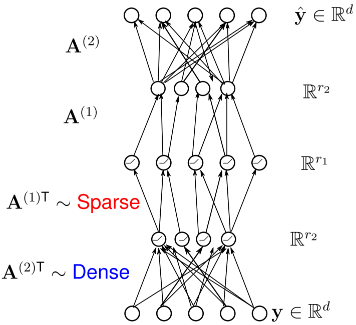
Deep sparse coding theory guides architecture design

<u>Deeply-sparse coding model</u>



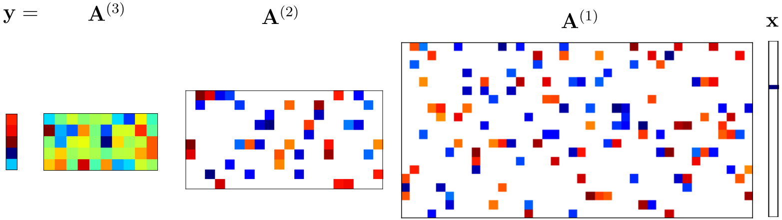
1. Pick r_2, # units in 1st hidden layer.
2. $r_1 = \mathcal{O}\left(\max(r_2^2, r_2 s_{\mathbf{A}^{(1)}})\right)$, i.e. expansion.
3. Reconstruct.

<u>Associated AE</u>



<u>Theorem (Ba 2020)</u>: need $n = \mathcal{O}\left(\max(r_1^2, r_1 s_{\mathbf{x}})\right)$ examples to learn $\mathbf{A}^{(1)}$ and $\mathbf{A}^{(2)}$.

Predictions of deep sparse coding theory



- Predictions for hierarchical sensory processing
  - Sparse activation of neurons at every level
  - *Increasingly sparse* activations as one goes up the hierarchy: Barlow 1961, (cat neuron, dog neuron?).
  - Sparse receptive fields at higher levels of hierarchy.
- Can we design experiments to test/verify/modify?

Implications/Concluding thoughts

1. A principled way to design neural networks. Choices of
   - Nonlinearities (e.g. ReLU) $\iff$ *prior* on hidden activations.
   - Training loss $\iff$ assumptions on data *likelihood*.
2. Fast learning and inference using GPUs.
3. Interpretable architectures; orders of mag. fewer params. than standard neural nets.
4. A call for utilizing neural networks in a principled way to
   - 4.1 Solve pattern discovery problems in neuroscience.
   - 4.2 Elucidate principles of hierarchical sensory processing, in conjunction with experiments.

Thank you

Demba Ba
demba@seas.harvard.edu
https://crip.seas.harvard.edu/

https://github.com/demba/
https://bitbucket.org/demba/

Tankala, Tassissa, Murphy and **Ba**. "K-deep simplex: deep manifold learning via local dictionaries". Submitted to ICML 2021.

Ba. "Deeply-sparse signal representations". IEEE Transactions on Signal Processing, 2020.

Tolooshams, Dey and **Ba**. "Constrained recurrent sparse auto-encoders: Expectation-maximization based architectures for convolutional dictionary learning". IEEE Transcations on Neural Networks, 2020.

Tolooshams, Song, Temereanca and **Ba**. "Convolutional dictionary learning based auto-encoders for natural exponential-family distributions". ICML 2020.

Zazo, Tolooshams and **Ba**. "Convolutional dictionary learning in hierarchical networks". in Proceedings of IEEE CAMSAP 2019.