Sparse Coding, ReLU Auto-encoders, and Pattern Discovery in Neural Data

Demba Ba[1]
Associate Professor of Electrical Engineering
and Bioengineering

[1]Harvard University
School of Engineering and Applied Sciences (SEAS)

Neuroacademy

HARVARD
JOHN A. PAULSON
SCHOOL OF ENGINEERING
AND APPLIED SCIENCES

Outline

Learning patterns from neural data

Auto-encoders for spike sorting

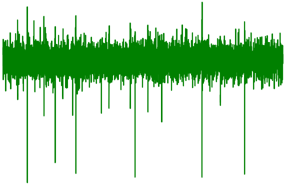
Outline

Learning patterns from neural data

Auto-encoders for spike sorting

Auto-encoders for spike sorting

Electrophysiology



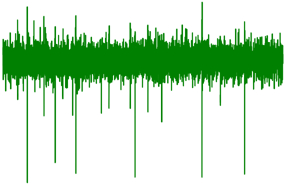
Sparse Coding

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

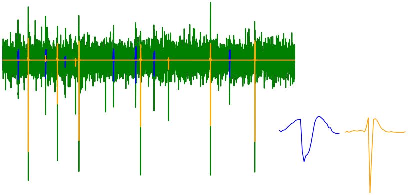
$\mathbf{x} \sim$ sparse

Auto-encoders for spike sorting

Electrophysiology



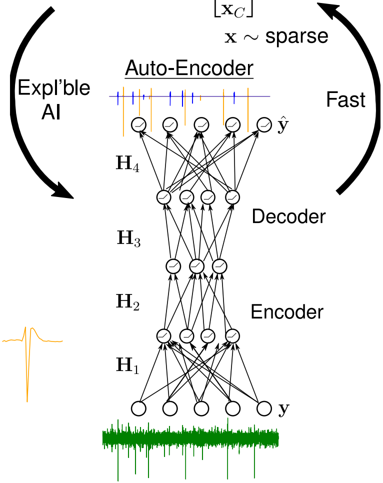
Spike Sorting



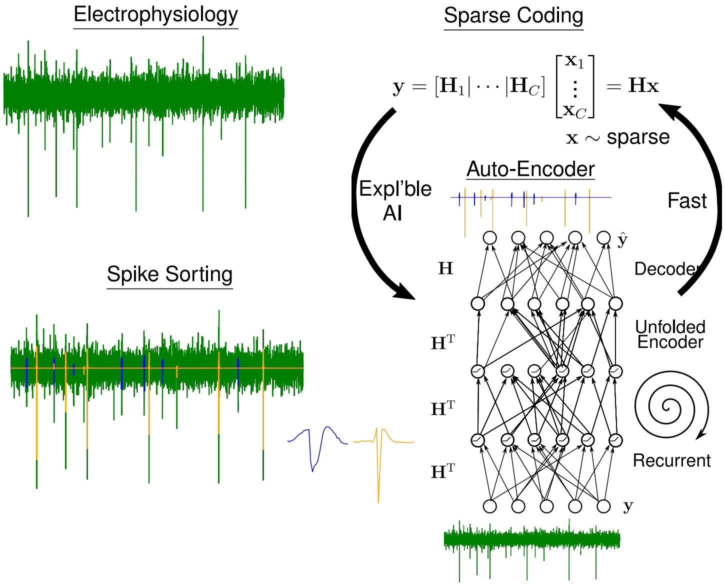
Sparse Coding

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

$\mathbf{x} \sim$ sparse



Auto-encoders for spike sorting

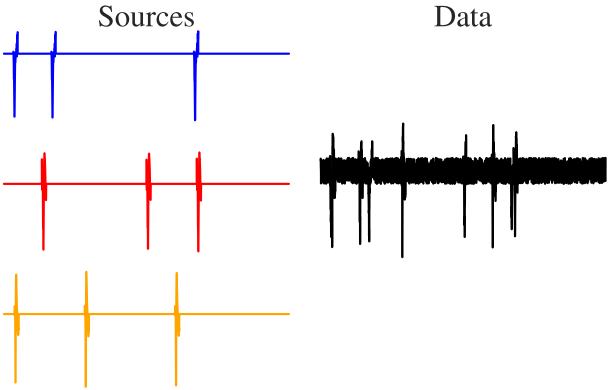


Outline

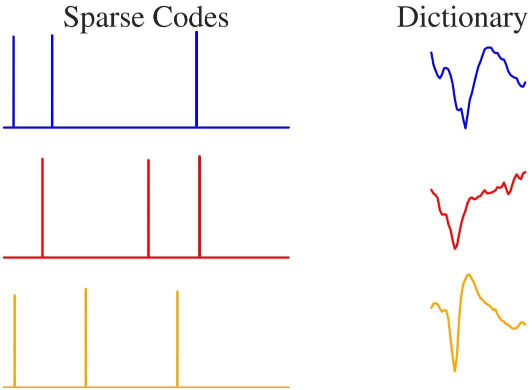
Learning patterns from neural data

Auto-encoders for spike sorting

Blind source separation by dictionary learning

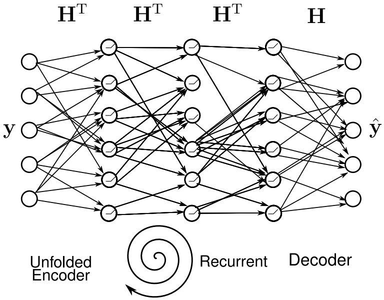


Blind source separation by dictionary learning



Goal: neural networks for dictionary learning

Auto-Encoder



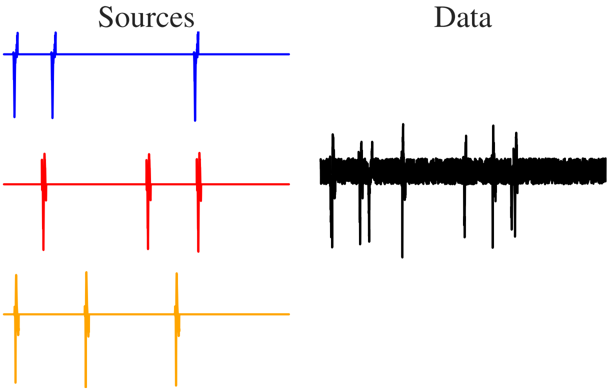
Dictionary Learning

$$\mathbf{y} = \begin{bmatrix} & & \\ & \mathbf{H} & \\ & & \end{bmatrix} \begin{bmatrix} \\ \\ \mathbf{x} \\ \\ \\ \end{bmatrix}$$

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:

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

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

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

Optimization perspective

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

<u>CDL solves</u>:

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

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).

Convolutional dictionary update Step

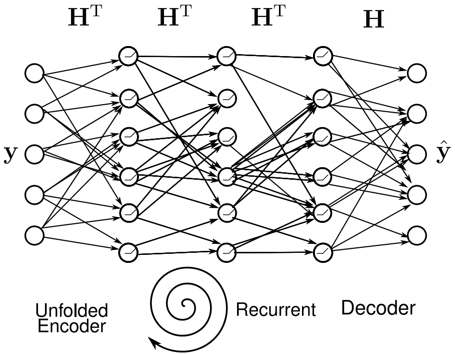
$$\min_{(\mathbf{h}_c)_{c=1}^{C}} \sum_{j=1}^{J} \frac{1}{2} \left|\left|\mathbf{y}^j - \mathbf{H}\mathbf{x}^j\right|\right|_2^2 \text{ s.t. } ||\mathbf{h}_c||_2 \leq 1 \quad \text{for } c = 1, \cdots, C.$$

This step is

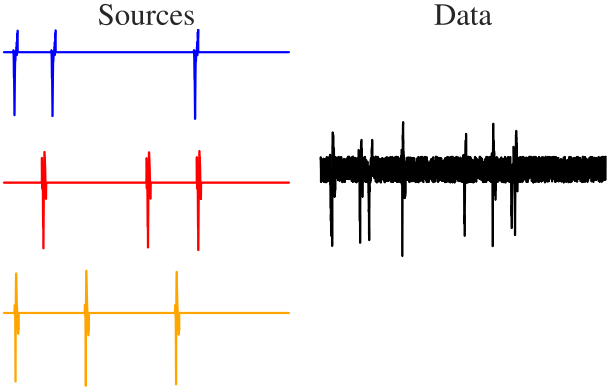
- Computationally Expensive.
- Not parallelizable over J examples.

Auto-encoder for CDL by deep unfolding and weight tying

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



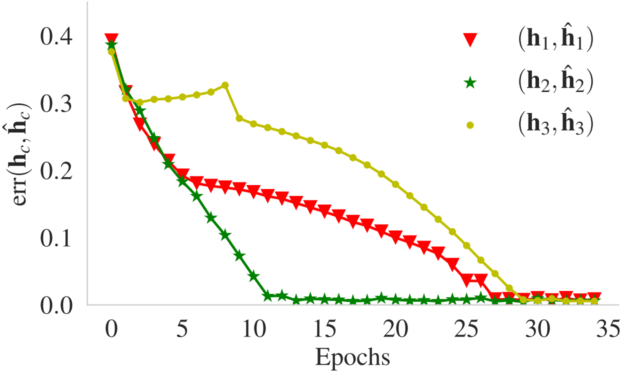
Simulated data



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

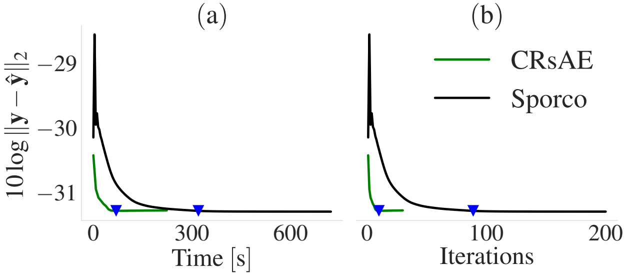
AE performs dictionary learning

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

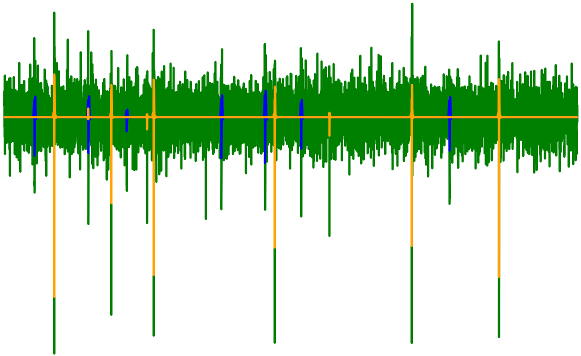


Training of AE is fast

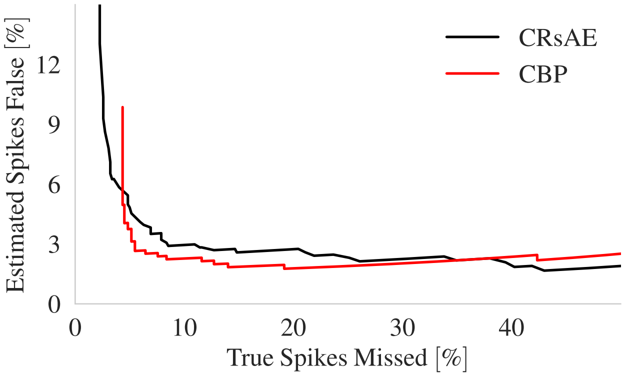
		CRsAE	Sporco	CBP
Learning Spike Shapes	runtime	**69.27 s**	319.52 s	
	iterations	**10**	89	
Spike Sorting	runtime	**0.93 s**		17 hours



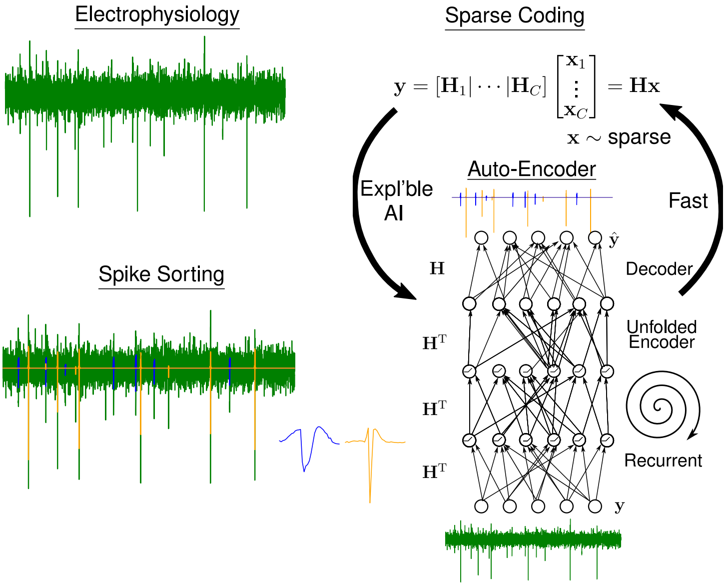
Harris dataset: spike sorting and denoising



Spike Sorting



Concluding thoughts



Thank you

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

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

Special shout out to Konrad Kording for converting the slides to the NMA format.

Song, Tolooshams, Temereanca and **Ba**. Convolutional dictionary learning of stimulus form spiking data. Cosyne 2020.

Song, Flores and **Ba**. Convolutional dictionary learning with grid refinement. IEEE Transcations on Neural Networks, 2020.

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

Tolooshams, Dey and **Ba**. Deep residual auto encoders for expectation maximization inspired dictionary learning. Submitted to IEEE Transactions on Neural Networks, 2019.