Convolutional dictionary learning based auto-encoders for natural exponential-family distributions

Bahareh Tolooshams*¹, Andrew H. Song*², Simona Temereanca³, and Demba Ba¹

¹Harvard University ²Massachusetts Institute of Technology ³Brown University

*Equal contributions

CRISP Group: https://crisp.seas.harvard.edu

ICML 2020
1 Motivation

2 Introduction

3 Deep Convolutional Exponential Auto-encoder (DCEA)

4 Experiments

5 Conclusion
Motivation

Deep Learning

• Fast and scalable ✓
• Not interpretable ×
• Memory and computationally expensive ×

Signal Processing (SP)

Generative models
e.g., sparse coding model
\[ p(y | x) = Hx + \epsilon, \quad x \text{ is sparse} \]

• Slow and not scalable ×
• Interpretable ✓
• Memory efficient ✓

• Benefit from scalability of deep learning for traditional SP tasks.
• Guide to design interpretable and memory efficient networks.
1 Motivation

2 Introduction

3 Deep Convolutional Exponential Auto-encoder (DCEA)

4 Experiments

5 Conclusion
Generative model for each data $j$

$$y^j = \sum_{c=1}^{C} h_c \ast x^j_c + \epsilon^j = Hx^j + \epsilon^j, \quad \epsilon^j \sim \mathcal{N}(0, \sigma^2 I)$$

where $x^j_c$ is sparse.

**Goal:** Learn $H$ that maps sparse representation $x^j$ to data $y^j$.

$$\min_{\{h_c\}_{c=1}^{C}, \{x^j\}_{j=1}^{J}} \frac{1}{2} \sum_{j=1}^{J} \|y^j - Hx^j\|_2^2 + \lambda \|x^j\|_1$$

- min w.r.t. $x^j \rightarrow$ Convolutional Sparse Coding (CSC).
- min w.r.t. $H$ and $x^j \rightarrow$ Convolutional Dictionary Learning (CDL).
Unfolding Networks

Solve CSC and CDL by iterative proximal gradient algorithm.

ISTA [1]:

\[ y \rightarrow \tilde{y}_t \rightarrow \alpha H^T \rightarrow x_t \rightarrow x_T \]

LISTA [2]:

\[ y \rightarrow W^e \rightarrow x_t \rightarrow x_T \]

CSCNet [3]:

\[ y \rightarrow \tilde{y}_t \rightarrow W^e \rightarrow x_t \rightarrow x_T \rightarrow H \]
Unfolding Networks

Solve CSC and CDL by iterative proximal gradient algorithm.

ISTA [1]:

\[ y \rightarrow \tilde{y}_t \rightarrow \alpha H^T \rightarrow x_t \rightarrow x_T \]

LISTA [2]:

\[ y \rightarrow W^e \rightarrow x_t \rightarrow x_T \]

CSCNet [3]:

\[ y \rightarrow \tilde{y}_t \rightarrow W^e \rightarrow x_t \rightarrow x_T \rightarrow H \]
Unfolding Networks

Solve CSC and CDL by iterative proximal gradient algorithm.

ISTA [1]:

\[
\begin{align*}
  y & \rightarrow \tilde{y}_t \rightarrow \alpha H^T \rightarrow x_t \rightarrow x_T \\
  \text{ISTA} \ [1] & \\
\end{align*}
\]

LISTA [2]:

\[
\begin{align*}
  y & \rightarrow W^e \rightarrow \tilde{x}_t \rightarrow x_T \\
  \text{LISTA} \ [2] & \\
\end{align*}
\]

CSCNet [3]:

\[
\begin{align*}
  y & \rightarrow \tilde{y}_t \rightarrow W^e \rightarrow x_t \rightarrow x_T \rightarrow H \\
  \text{CSCNet} \ [3] & \\
\end{align*}
\]
What if the observations are no longer Gaussian?

**Count-valued data**

Fingerprint

Photon-based imaging

**Classical CDL approach:** Alternating minimization with a Poisson generative model [4, 5].

- Unsupervised ✓
- Follows a generative model ⇒ interpretable ✓
- Not scalable (can take minutes ~ hours to denoise single image) ✗
Our Contributions

- Auto-encoder inspired by CDL, termed **Deep Convolutional Exponential Auto-encoder (DCEA)**, for non real-valued data
- Demonstration of the flexibility of DCEA for both
  - *unsupervised* task, e.g., CDL
  - *supervised* task, e.g., Poisson denoising problem
- Gradient dynamics of shallow exponential auto-encoder (SEA)
  - Prove that SEA recovers parameters of the generative model.
1 Motivation

2 Introduction

3 Deep Convolutional Exponential Auto-encoder (DCEA)

4 Experiments

5 Conclusion
Deep Convolutional Exponential Auto-encoder

Problem description

Natural exponential family with convolutional generative model:

$$\log p(y|\mu) = f(\mu)^T y + g(y) - B(\mu), \text{ where } f(\mu) = Hx, \text{ } x \text{ is sparse.}$$

<table>
<thead>
<tr>
<th></th>
<th>$y$</th>
<th>$B(z)$</th>
<th>Inverse link: $f^{-1}(\cdot)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian</td>
<td>$\mathbb{R}$</td>
<td>$z^Tz$</td>
<td>$I(\cdot)$</td>
</tr>
<tr>
<td>Binomial</td>
<td>$[0..M]$</td>
<td>$-1^T \log(1 - z)$</td>
<td>sigmoid(\cdot)</td>
</tr>
<tr>
<td>Poisson</td>
<td>$[0..\infty)$</td>
<td>$1^Tz$</td>
<td>exp(\cdot)</td>
</tr>
</tbody>
</table>

Exponential Convolutional Dictionary Learning (ECDL):

$$\min_{H,x} -\log p(y|\mu) + \lambda \|x\|_1$$
Deep Convolutional Exponential Auto-encoder

Network architecture

Components for different distributions

<table>
<thead>
<tr>
<th></th>
<th>$y$</th>
<th>$f^{-1}(\cdot)$</th>
<th>Encoder Unfolding ($x_t$)</th>
<th>Decoder ($f(\hat{\mu})$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian</td>
<td>$\mathbb{R}$</td>
<td>$I(\cdot)$</td>
<td>$S_b \left( x_{t-1} + \alpha H^T \tilde{y}_t \right)$</td>
<td>$Hx_T$</td>
</tr>
<tr>
<td>Binomial</td>
<td>$[0..M]$</td>
<td>sigmoid(\cdot)</td>
<td>$S_b \left( x_{t-1} + \alpha H^T \left( \frac{1}{M} \tilde{y}_t \right) \right)$</td>
<td>$Hx_T$</td>
</tr>
<tr>
<td>Poisson</td>
<td>$[0..\infty)$</td>
<td>exp(\cdot)</td>
<td>$S_b \left( x_{t-1} + \alpha H^T \left( \text{Elu}(\tilde{y}_t) \right) \right)$</td>
<td>$Hx_T$</td>
</tr>
</tbody>
</table>
Deep Convolutional Exponential Auto-encoder

Training & inference

Forward pass

Repeat $T$ times

$y \rightarrow \tilde{y}_t \rightarrow \alpha H^T \rightarrow x_t \rightarrow x_T \rightarrow H \rightarrow \mathcal{L} \rightarrow y$

Backward pass

Training

- **Forward pass**: Estimate code $x_T$ & compute loss function.
- **Backward pass** (back-propagation): Estimate dictionary $H$.
- Equivalent to *alternating minimization* in CDL.

Inference: Once trained, the inference (forward pass) is fast.
Unsupervised $\rightarrow$ Supervised

Repurpose DCEA for supervised tasks with two modifications

1. **Loss function**: Any supervised loss function, e.g., reconstruction MSE loss or perceptual loss.

2. **Architecture**: Relax the constraints $\rightarrow$ Untie the weights of encoder and decoder, learn the bias $b$.

<table>
<thead>
<tr>
<th></th>
<th>Encoder</th>
<th>Decoder</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>$x_t = S_b(x_{t-1} + \alpha H^T(y - f^{-1}(Hx_{t-1})))$</td>
<td>$Hx_T$</td>
</tr>
<tr>
<td>Relaxed</td>
<td>$x_t = S_b(x_{t-1} + \alpha (W^c)^T(y - f^{-1}(W^d x_{t-1})))$</td>
<td>$Hx_T$</td>
</tr>
</tbody>
</table>

- Further relaxations possible, i.e., deep & non-linear decoder.
Motivation

Introduction

Deep Convolutional Exponential Auto-encoder (DCEA)

Experiments

Conclusion
### Baseline frameworks

<table>
<thead>
<tr>
<th>Supervised?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPDA [5]</td>
<td>✓</td>
</tr>
<tr>
<td>CA [6]</td>
<td>✓</td>
</tr>
<tr>
<td>DCEA-C (ours)</td>
<td>✓</td>
</tr>
<tr>
<td>DCEA-UC (ours)</td>
<td>✓</td>
</tr>
</tbody>
</table>

**PSNR performance on test dataset**

<table>
<thead>
<tr>
<th></th>
<th>SPDA</th>
<th>CA</th>
<th>DCEA-C</th>
<th>DCEA-UC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peak 1</td>
<td>20.39</td>
<td>21.78</td>
<td>20.72</td>
<td>21.84</td>
</tr>
<tr>
<td>BSD68</td>
<td>21.51</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Peak 2</td>
<td>21.70</td>
<td>22.90</td>
<td>22.02</td>
<td>22.79</td>
</tr>
<tr>
<td>BSD68</td>
<td>22.97</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Peak 4</td>
<td>22.56</td>
<td>23.98</td>
<td>23.51</td>
<td>24.10</td>
</tr>
<tr>
<td>BSD68</td>
<td>24.40</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td># of Params</td>
<td>160K</td>
<td>655K</td>
<td>20K</td>
<td>61K</td>
</tr>
</tbody>
</table>
Experiments
Poisson image denoising

<table>
<thead>
<tr>
<th>Original</th>
<th>Noisy peak= 4</th>
<th>DCEA-C</th>
<th>DCEA-UC</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Original Image" /></td>
<td><img src="image2" alt="Noisy Image" /></td>
<td><img src="image3" alt="DCEA-C Image" /></td>
<td><img src="image4" alt="DCEA-UC Image" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Original</th>
<th>Noisy peak= 2</th>
<th>DCEA-C</th>
<th>DCEA-UC</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image5" alt="Original Image" /></td>
<td><img src="image6" alt="Noisy Image" /></td>
<td><img src="image7" alt="DCEA-C Image" /></td>
<td><img src="image8" alt="DCEA-UC Image" /></td>
</tr>
</tbody>
</table>
Experiments

Poisson image denoising

- **Classical ECDL**: SPDA vs. **DCEA-C**
  - ⇒ better denoising + *much more efficient*
  - ⇒ classical inference task leveraging *scalability of NN*

- **Denoising NN**: CA vs. **DCEA-UC**
  - ⇒ competitive denoising + *much less parameters*
  - ⇒ NN architecture leveraging *generative model paradigm*
Experiments
CDL for simulated binomial data

Figure: Example of simulated neural spikes and the rate (truth)

Figure: Random initialized (Blue), true (Orange), and learned templates (Green)
Experiments

CDL for simulated binomial data

• If we untie the weights, i.e., relax generative model constraints

![Graph showing time in milliseconds vs. output for different values of C](image1)

• If we treat binomial data as Gaussian obs., i.e., model mismatch

![Graph showing number of trials per group vs. output for different filters](image2)
1 Motivation

2 Introduction

3 Deep Convolutional Exponential Auto-encoder (DCEA)

4 Experiments

5 Conclusion
In conclusion, **Deep Convolutional Exponential Auto-encoder (DCEA)**

- is a class of NN based on a generative model for CDL, using data from natural exponential family.
- shows competitive performance in Poisson denoising tasks against SOTA frameworks, *with an order of magnitude fewer* trainable parameters (**supervised task**).
- is able to learn accurate convolutional patterns in ECDL task with simulated binomial and real neural spiking observations (**unsupervised task**).
Reference

I. Daubechies, M. Defrise, and C. De Mol.
An iterative thresholding algorithm for linear inverse problems with a sparsity constraint.

Karol Gregor and Yann Lecun.
Learning fast approximations of sparse coding.

D. Simon and M. Elad.
Rethinking the CSC model for natural images.

Joseph Salmon, Zachary Harman, Charles-Alban Deledalle, and Rebecca Willett.
Poisson noise reduction with non-local pca.

Raja Giryes and Michael Elad.
Sparsity-based poisson denoising with dictionary learning.

Tal Remez, Or Litany, Raja Giryes, and Alexander M. Bronstein.
Class-aware fully-convolutional gaussian and poisson denoising.