MASTER THESIS
Major in Information Technologies

STRUCTURED AUTO-ENCODER
WITH APPLICATION TO
MUSIC GENRE RECOGNITION

Student
Michaël Defferrard

Professor
Pierre Vandergheynst

Supervisors
Xavier Bresson
Johan Paratte

EPFL LTS2 Laboratory
July 3, 2015
Objective: unsupervised representation learning toward the goal of automatic features extraction.

Model: we introduce the *structured auto-encoder*, an hybrid auto-encoder variant, which preserves the structure of the data while transforming it in a sparse representation.

Ideas: borrowed from sparse coding and manifold learning.

Application: the proposed model shall be evaluated through a classification task. We propose an application in Music Information Retrieval (MIR).
Overview

Introduction

Algorithm
  Background
  Model
  Related works
  Optimization

Application
  Music genre recognition
  System
  Implementation
  Results

Conclusion
Auto-encoders
A kind of feed-forward neural network

\[ x \xrightarrow{z = h_1(Ex)} x^* = h_2(Dz) \]
Assumptions

1. **Sparse representation**: we make the hypothesis that a set of sample signals drawn from the same distribution can be sparsely represented in some frame.

2. **Manifold assumption, i.e. structured data**: we assume that the data is drawn from sampling a probability distribution that has support on or near to a submanifold embedded in the ambient space.

3. **Encoder**: we further make the assumption that a simple encoder can be learned to avoid the need of an optimization process that extracts the features during testing, i.e. when the model is trained.
Definitions

- A set $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N \in \mathbb{R}^{n \times N}$ of $N$ signals of dimensionality $n$.

- The set $\mathbf{Z} = \{\mathbf{z}_i\}_{i=1}^N \in \mathbb{R}^{m \times N}$ of their associated representations of dimensionality $m$.

- A dictionary (frame) $\mathbf{D} \in \mathbb{R}^{n \times m}$ of learning capacity $m$.

- A trainable direct encoder $\mathbf{E} \in \mathbb{R}^{m \times n}$. 
Linear regression
Find a representation

A signal $\mathbf{x} \in \mathcal{X} = \text{span} \mathbf{X} \subset \mathbb{R}^n$, where $\mathcal{X}$ is the subspace spanned by the input data, is represented by $\mathbf{z} \in \mathbb{R}^m$ with a reconstruction error $\epsilon \in \mathbb{R}^n$.

Model:

$$\mathbf{x} = \mathbf{Dz} + \epsilon.$$  

Ordinary least squares:

$$\mathbf{z}^* = \arg \min_{\mathbf{z}} \|\mathbf{x} - \mathbf{Dz}\|_2^2 = (\mathbf{D}^T \mathbf{D})^{-1} \mathbf{D}^T \mathbf{x}.$$
Sparse coding
Regularize the ill-posed linear regression model

Motivations:
▶ Succinct representation of the signal, explanatory.
▶ Easier linear separability due to higher dimensionality ($m > n$).

Sparse coding:
\[
\mathbf{z}^* = \arg\min_{\mathbf{z}} \frac{\lambda_d}{2} \| \mathbf{x} - \mathbf{Dz} \|_2^2 + \lambda_z \| \mathbf{z} \|_0.
\]

Basis Pursuit approximation:
\[
\mathbf{z}^* = \arg\min_{\mathbf{z}} \frac{\lambda_d}{2} \| \mathbf{x} - \mathbf{Dz} \|_2^2 + \lambda_z \| \mathbf{z} \|_1.
\]
Dictionary learning

Learn adaptive features

Motivations:

- Hand-crafted features are hard to design.
- Adaptive dictionary leads to more compact representation and discovery of previously unknown discriminative features.
- A strategy employed in the cortex for visual and auditory processing.

\[
\begin{align*}
\text{minimize} & \quad \frac{\lambda_d}{2} \|X - DZ\|_F^2 + \lambda_z \|Z\|_1 \\
\text{s.t.} & \quad \|d_i\|_2 \leq 1, \quad i = 1, \ldots, m.
\end{align*}
\]
Manifold learning
Structured representation

Motivation: exploit the geometrical structure of the data space.

Similarity graph:

\[ w_{ij} = \exp \left( -\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \in [0, 1] \quad \text{and} \quad a_{ii} = \sum_{j=1}^{N} w_{ij}. \]

Combinatorial graph Laplacian:

\[ L = A - W, \quad \text{with} \quad W = (w_{ij}) \in \mathbb{R}^{N \times N} \quad \text{and} \quad A = (a_{ij}). \]
The Laplacian as a difference operator on the graph signal $y \in \mathbb{R}^N$:

$$(Ly)_i = \sum_{j=1}^{N} w_{ij} (y_i - y_j).$$

Promote smoothness on the data manifold by minimizing the Dirichlet energy:

$$\text{tr}(ZLZ^T) = \sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij} \|z_i - z_j\|_2^2 \geq 0.$$
Auto-encoder

Train an explicit encoder

Objective function as an energy formulation:

\[
\frac{\lambda_d}{2} \|X - DZ\|_F^2 + \lambda_z \|Z\|_1 + \frac{\lambda_g}{2} \text{tr}(ZLZ^T) + \frac{\lambda_e}{2} \|Z - EX\|_F^2.
\]

Auto-encoder model.

Given a training set \(X\), fix the hyper-parameters \(\lambda_d, \lambda_z, \lambda_g, \lambda_e \geq 0\), construct the graph Laplacian \(L\) and

\[
\minimize_{Z,D,E} f_d(Z,D) + f_z(Z) + f_g(Z) + f_e(Z,E)
\]

s.t. \(\|d_i\|_2 \leq 1, \|e_k\|_2 \leq 1, \ i = 1, \ldots, m, \ k = 1, \ldots, n\)

to learn the model parameters \(D\) and \(E\).
Approximation schemes

Encoder: find the representation $z$ of an unseen sample $x$.

$$z^* = \arg \min_z \frac{\lambda_d}{2} \| x - Dz \|^2_2 + \lambda_z \| z \|^1_1 + \frac{\lambda_g}{2} \langle z, Lz \rangle + \frac{\lambda_e}{2} \| z - Ex \|^2_2$$

Direct: $\tilde{z} = \arg \min_z \frac{\lambda_e}{2} \| z - Ex \|^2_2 + \lambda_z \| z \|^1_1 = h_{\lambda_z/\lambda_e}(Ex) \approx z^*$

where $h_{\lambda}$ is a shrinkage function.

Decoder: find the reciprocal sample $x$ of a representation $z$.

$$x^* = \arg \min_x \frac{\lambda_d}{2} \| x - Dz \|^2_2 + \frac{\lambda_e}{2} \| z - Ex \|^2_2$$

Direct: $\tilde{x} = \arg \min_x \frac{\lambda_d}{2} \| x - Dz \|^2_2 = Dz \approx x^*$.
Related works

**Standard auto-encoders**: learn \( D \) and \( E \) with an \( \ell_2 \) fidelity term (and non-linear activation functions), without any explicit regularization on \( Z \).

**Sparse auto-encoders**: learn \( D \) with an \( \ell_2 \) fidelity term and an \( \ell_1 \) regularization on \( Z \).

**Predictive sparse decomposition**: add an explicit encoder \( E \) (\( \ell_2 \) fidelity, non-linear activation) to sparse coding.

**Denoising auto-encoders**: same model as the standard ones, but trained with stochastically corrupted data.
Convex sub-problems

Three inter-dependent but convex sub-problems:

\[
\begin{align*}
\text{minimize} & \quad f_d(Z, D) + f_z(Z) + f_g(Z) + f_e(Z, E), \\
\text{minimize} & \quad f_d(Z, D) \quad \text{s.t.} \quad \|d_i\|_2 \leq 1, \quad i = 1, \ldots, m, \\
\text{minimize} & \quad f_e(Z, E) \quad \text{s.t.} \quad \|e_k\|_2 \leq 1, \quad k = 1, \ldots, n.
\end{align*}
\]

- Iteratively solve each sub-problem.
- Several (iterative) methods to solve each of them.
Proximal splitting

Solve minimize $f_1(x) + f_2(x)$ where $f_1$ is non-smooth and $f_2$ is differentiable with a $\beta$-Lipschitz continuous gradient $\nabla f_2$.

Proximity operator: $\text{prox}_f x = \min_y f(y) + \frac{1}{2} \| x - y \|_2^2$.

Forward-backward: $x^{t+1} = \text{prox}_{\gamma^t f_1}(x^t - \gamma^t \nabla f_2(x^t))$.

FISTA is an efficient scheme which exploits variable time steps and multiple points to achieve an optimal $O(1/t^2)$ rate of convergence.
Sub-problems casting

For $Z$: minimize $\mathcal{J}(Z) = f_2(Z) + f_1(Z)$

$\triangledown f_2(Z) = \lambda_d D^T (X - DZ) + \lambda_e (Z - EX) + \lambda_g LZ$

$\beta \geq \lambda_e + \lambda_d \|D^T D\|_2 + \lambda_g \|L\|_2$

$\text{prox}_{\beta^{-1}f_1}(Z) = h_{\lambda_z/\beta}(Z)$

For $D$ (and similarly $E$): minimize $\mathcal{J}(D) = f_2(D) + f_1(D)$

$\triangledown f_2(D) = \lambda_d Z (X^T - Z^T D^T)$

$\beta \geq \lambda_d \|ZZ^T\|_2$

$\text{prox}_{\beta^{-1}f_1}(D) = \left\{ \frac{d_i}{\max(1,\|d_i\|_2)} \right\}_{i=1}^{m}$
Music genre recognition

- Problem: automatically recognize the musical genre of an unknown clip without access to any meta-data.

- Training data: a set of labeled clips.

- Classification accuracy used as a proxy to assess the discriminative power of the learned representations.

- GTZAN dataset: 1000 30-second audio clips with 100 examples in each of 10 different categories: blues, classical, country, disco, hiphop, jazz, metal, pop, reggae and rock.
System

Partly labeled clips

Frames

Constant-Q Transform

Local Contrast Normalization

Scaling

Preprocessing

Features extraction

\[ \|z\|_1, \langle z, Lz \rangle \]

Aggregate features

Linear support vector machine

Classification

h(Ex)

Dz

\[ x \]

Fully labeled clips

Features extraction

\[ \langle z, Lz \rangle \]

Preprocessing

Frames

Constant-Q Transform

Local Contrast Normalization

Scaling

Partly labeled clips

Linear support vector machine

Majority voting
Implementation

1. **Tools**: numpy, scipy, matplotlib, scikit-learn, h5py, librosa, PyUNLocBoX\(^1\), IPython notebook, OpenStack lab cluster.

2. **Notebooks**: model construction, test on images, dataset conversion to HDF5, pre-processing, graph construction, auto-encoder model, features extraction, classification and test, experiments.

3. **Performance**:
   - Optimization for space: avoid copies, modify in place, float32, store \(Z\) as a scipy sparse matrix.
   - Optimization for speed: ATLAS/OpenBLAS, float32 (memory bandwidth), efficient trace, projection in the ball (not on the sphere), approximate KNN search with FLANN.

\(^1\)https://github.com/epfl-lts2/pyunlocbox

\(^2\)https://github.com/mdeff/dlaudio
Typical learning

(a) $m = 128$ atoms of a learned dictionary.

(b) A learned sparse (20% of non-zero coefficients) representation.

Figure: Learned dictionary $D$ and representation $Z$ of spectrograms.
Typical convergence

Sub-problems convergence

- Sub-problem objectives: $f_2(Z)$, $f_1(Z)$, $f_2(D)$ and $f_2(E)$.
- Sub-objectives: $f_d(Z, D)$, $f_e(Z, E)$, $f_z(Z)$ and $f_g(Z)$.
- Global objective $f_d(Z, D) + f_e(Z, E) + f_z(Z) + f_g(Z)$. 
Experiments
Backed up by simulation reports

1. Better convergence correlates with higher performance [12i].
2. Hyper-parameters do not have a huge influence. Only the order of magnitude is important [12j, 12k, 12l, 13h, 13j].
3. Distance metric (Euclidean or cosine) is not significant [13i].
4. Decreasing accuracy with increasing noise [13d].
5. Same optimal $\lambda_g$ in the presence of 10% noise [13b].
6. Training over testing ratio: no edge [13g, …].
7. Self-connections make no difference [14a].
8. Higher performance with a normalized graph Laplacian [14b].
9. $K \in [10, 20]$ neighbors is good [14c].
10. And many others\footnote{http://nbviewer.ipython.org/github/mdeff/dlaudio_results \footnote{https://lts2.epfl.ch/blog/mdeff}}.

\footnote{http://nbviewer.ipython.org/github/mdeff/dlaudio_results} \footnote{https://lts2.epfl.ch/blog/mdeff}
Classification accuracy

<table>
<thead>
<tr>
<th>Noise level (standard deviation)</th>
<th>0.0</th>
<th>0.1</th>
<th>0.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy using CQT spectrograms [%]</td>
<td>69.7</td>
<td>58.7</td>
<td>46.9</td>
</tr>
<tr>
<td>Accuracy with $\lambda_g = 0$ [%]</td>
<td>75.9</td>
<td>57.1</td>
<td>42.6</td>
</tr>
<tr>
<td>Accuracy with $\lambda_g = 100$ [%]</td>
<td>78.0</td>
<td>65.9</td>
<td>51.6</td>
</tr>
</tbody>
</table>

Table: Classification accuracies (mean of 20 10-fold cross-validation) on a subset of GTZAN: $N_{\text{genres}} = 5$ genres, $N_{\text{clips}} = 100$ clips per genre and $N_{\text{frames}} = 149$ frames per clip.

- Extracted features increase accuracy by $\sim 7\%$ over baseline for all scenarios.
- Structure increases accuracy by $2\%$ in the absence of noise.
- Structure provides robustness to noise.
Conservation of the structure in the data via graph regularization (the manifold assumption) is able to denoise the data.

Reasonable assumptions:
1. The representation is sparse.
2. The representation preserves the structure.
3. The existence of an encoder was not tested by lack of time.

Ways to improve accuracy:
- Fine-tune the hyper-parameters.
- Add complexity to the system, e.g. LCN or individual octaves.
- Construct better graphs, e.g. no KNN approximation.
- Work on a bigger dataset.
- Multiple layers to extract hierarchical features.
Questions ?