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

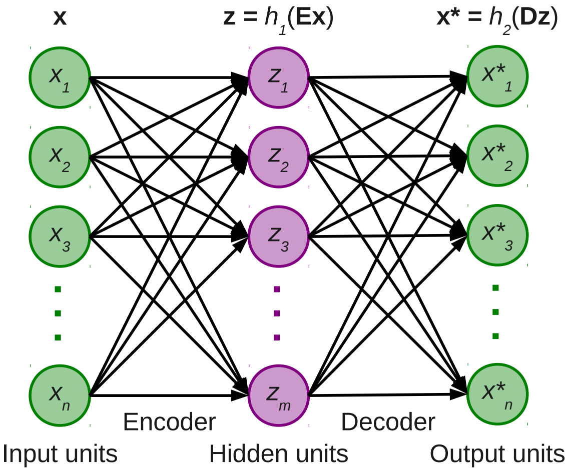
Introduction

- 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

Auto-encoders

A kind of feed-forward neural network



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 $\mathbf{x} \in \mathcal{X} = \operatorname{span} \mathbf{X} \subset \mathbb{R}^n$, 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:

$$\mathbf{x} = \mathbf{D}\mathbf{z} + \epsilon.$$

Ordinary least squares:

$$\mathbf{z}^* = \arg\min_{\mathbf{z}} \|\mathbf{x} - \mathbf{D}\mathbf{z}\|_2^2 = (\mathbf{D}^T\mathbf{D})^{-1}\mathbf{D}^T\mathbf{x}.$$

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:

$$\mathbf{z}^* = \arg\min_{\mathbf{z}} \frac{\lambda_d}{2} \|\mathbf{x} - \mathbf{D}\mathbf{z}\|_2^2 + \lambda_z \|\mathbf{z}\|_0.$$

Basis Pursuit approximation:

$$\mathbf{z}^* = \arg\min_{\mathbf{z}} \frac{\lambda_d}{2} \|\mathbf{x} - \mathbf{D}\mathbf{z}\|_2^2 + \lambda_z \|\mathbf{z}\|_1.$$

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.

$$\underset{\mathbf{Z},\mathbf{D}}{\text{minimize}} \quad \frac{\lambda_d}{2}\|\mathbf{X}-\mathbf{DZ}\|_F^2 + \lambda_z\|\mathbf{Z}\|_1$$

$$\text{s.t. } \|\mathbf{d}_i\|_2 \leq 1, \; i = 1, \dots, m.$$

Manifold learning

Structured representation

Motivation: exploit the geometrical structure of the data space.

Similarity graph:

$$w_{ij} = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma^2}\right) \in [0,1] \quad \text{and} \quad a_{ii} = \sum_{j=1}^{N} w_{ij}.$$

Combinatorial graph Laplacian:

$$\mathbf{L} = \mathbf{A} - \mathbf{W}, \quad \text{with} \quad \mathbf{W} = (w_{ij}) \in \mathbb{R}^{N\times N} \quad \text{and} \quad \mathbf{A} = (a_{ij}).$$

Manifold learning

Structured representation

The Laplacian as a difference operator on the graph signal $\mathbf{y} \in \mathbb{R}^N$:

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

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

$$\operatorname{tr}(\mathbf{Z}\mathbf{L}\mathbf{Z}^T) = \sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}\|\mathbf{z}_i - \mathbf{z}_j\|_2^2 \geq 0.$$

Auto-encoder

Train an explicit encoder

Objective function as an energy formulation:

$$\underbrace{\frac{\lambda_d}{2}\|\mathbf{X}-\mathbf{DZ}\|_\text{F}^2}_{f_d(\mathbf{Z},\mathbf{D})} + \underbrace{\lambda_z\|\mathbf{Z}\|_1}_{f_z(\mathbf{Z})} + \underbrace{\frac{\lambda_g}{2}\operatorname{tr}(\mathbf{ZLZ}^T)}_{f_g(\mathbf{Z})} + \underbrace{\frac{\lambda_e}{2}\|\mathbf{Z}-\mathbf{EX}\|_\text{F}^2}_{f_e(\mathbf{Z},\mathbf{E})}.$$

Auto-encoder model.
Given a training set $\mathbf{X}$, fix the hyper-parameters $\lambda_d, \lambda_z, \lambda_g, \lambda_e \geq 0$, construct the graph Laplacian $\mathbf{L}$ and

$$\underset{\mathbf{Z},\mathbf{D},\mathbf{E}}{\text{minimize}} \;\; f_d(\mathbf{Z},\mathbf{D}) + f_z(\mathbf{Z}) + f_g(\mathbf{Z}) + f_e(\mathbf{Z},\mathbf{E})$$
$$\text{s.t.} \;\; \|\mathbf{d}_i\|_2 \leq 1,\; \|\mathbf{e}_k\|_2 \leq 1,\; i = 1,\dots,m,\; k = 1,\dots,n$$

to learn the model parameters $\mathbf{D}$ and $\mathbf{E}$.

Approximation schemes

Encoder: find the representation $\mathbf{z}$ of an unseen sample $\mathbf{x}$.

$$\mathbf{z}^* = \arg\min_{\mathbf{z}} \frac{\lambda_d}{2}\|\mathbf{x} - \mathbf{D}\mathbf{z}\|_2^2 + \lambda_z\|\mathbf{z}\|_1 + \frac{\lambda_g}{2}\langle\mathbf{z}, \mathbf{L}\mathbf{z}\rangle + \frac{\lambda_e}{2}\|\mathbf{z} - \mathbf{E}\mathbf{x}\|_2^2$$

Direct: $\tilde{\mathbf{z}} = \arg\min_{\mathbf{z}} \frac{\lambda_e}{2}\|\mathbf{z} - \mathbf{E}\mathbf{x}\|_2^2 + \lambda_z\|\mathbf{z}\|_1 = h_{\lambda_z/\lambda_e}(\mathbf{E}\mathbf{x}) \approx \mathbf{z}^*$
where h_λ is a shrinkage function.

Decoder: find the reciprocal sample $\mathbf{x}$ of a representation $\mathbf{z}$.

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \frac{\lambda_d}{2}\|\mathbf{x} - \mathbf{D}\mathbf{z}\|_2^2 + \frac{\lambda_e}{2}\|\mathbf{z} - \mathbf{E}\mathbf{x}\|_2^2$$

Direct: $\tilde{\mathbf{x}} = \arg\min_{\mathbf{x}} \frac{\lambda_d}{2}\|\mathbf{x} - \mathbf{D}\mathbf{z}\|_2^2 = \mathbf{D}\mathbf{z} \approx \mathbf{x}^*$.

Related works

Standard auto-encoders: learn **D** and **E** with an ℓ_2 fidelity term (and non-linear activation functions), without any explicit regularization on **Z**.

Sparse auto-encoders: learn **D** with an ℓ_2 fidelity term and an ℓ_1 regularization on **Z**.

Predictive sparse decomposition: add an explicit encoder **E** (ℓ_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:

$$\begin{aligned}
&\underset{\mathbf{Z}}{\text{minimize}}\ f_d(\mathbf{Z},\mathbf{D}) + f_z(\mathbf{Z}) + f_g(\mathbf{Z}) + f_e(\mathbf{Z},\mathbf{E}),\\
&\underset{\mathbf{D}}{\text{minimize}}\ f_d(\mathbf{Z},\mathbf{D}) \text{ s.t. } \|\mathbf{d}_i\|_2 \leq 1,\ i = 1,\dots,m,\\
&\underset{\mathbf{E}}{\text{minimize}}\ f_e(\mathbf{Z},\mathbf{E}) \text{ s.t. } \|\mathbf{e}_k\|_2 \leq 1,\ k = 1,\dots,n.
\end{aligned}$$

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

Proximal splitting

Solve $\underset{\mathbf{x}}{\text{minimize}} \; f_1(\mathbf{x}) + f_2(\mathbf{x})$ where f_1 is non-smooth and f_2 is differentiable with a β-Lipschitz continuous gradient ∇f_2.

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

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

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 $\mathbf{Z}$: $\underset{\mathbf{Z}}{\text{minimize}} \; \overbrace{f_d(\mathbf{Z},\mathbf{D}) + f_g(\mathbf{Z}) + f_e(\mathbf{Z},\mathbf{E})}^{f_2(\mathbf{Z})} + \overbrace{f_z(\mathbf{Z})}^{f_1(\mathbf{Z})}$

- $\nabla f_2(\mathbf{Z}) = \lambda_d \mathbf{D}^T(\mathbf{X} - \mathbf{D}\mathbf{Z}) + \lambda_e(\mathbf{Z} - \mathbf{E}\mathbf{X}) + \lambda_g \mathbf{L}\mathbf{Z}$
- $\beta \geq \lambda_e + \lambda_d \|\mathbf{D}^T\mathbf{D}\|_2 + \lambda_g \|\mathbf{L}\|_2$
- $\text{prox}_{\beta^{-1} f_1}(\mathbf{Z}) = h_{\lambda_z/\beta}(\mathbf{Z})$

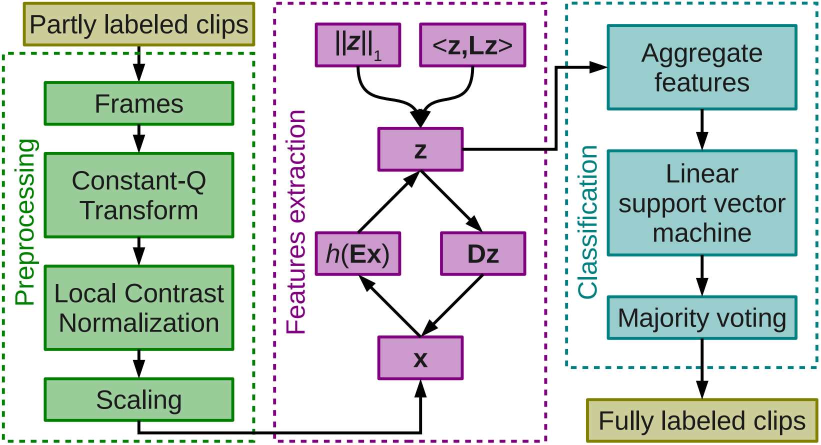
For $\mathbf{D}$ (and similarly $\mathbf{E}$): $\underset{\mathbf{D}}{\text{minimize}} \; \overbrace{\frac{\lambda_d}{2}\|\mathbf{X}^T - \mathbf{Z}^T\mathbf{D}^T\|_F^2}^{f_2(\mathbf{D})} + \overbrace{\iota_C(\mathbf{D})}^{f_1(\mathbf{D})}$

- $\nabla f_2(\mathbf{D}) = \lambda_d \mathbf{Z}(\mathbf{X}^T - \mathbf{Z}^T\mathbf{D}^T)$
- $\beta \geq \lambda_d \|\mathbf{Z}\mathbf{Z}^T\|_2$
- $\text{prox}_{\beta^{-1} f_1}(\mathbf{D}) = \left\{ \frac{\mathbf{d}_i}{\max(1, \|\mathbf{d}_i\|_2)} \right\}_{i=1}^{m}$

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

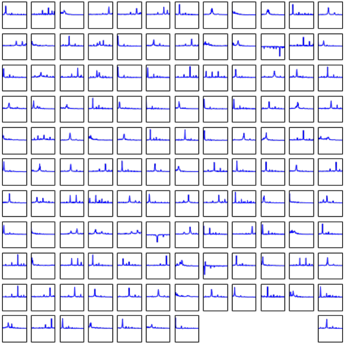

Partly labeled clips
Preprocessing
Frames
Constant-Q Transform
Local Contrast Normalization
Scaling
Features extraction
||z||1
<z,Lz>
z
h(Ex)
Dz
x
Classification
Aggregate features
Linear support vector machine
Majority voting
Fully labeled clips

Implementation[2]

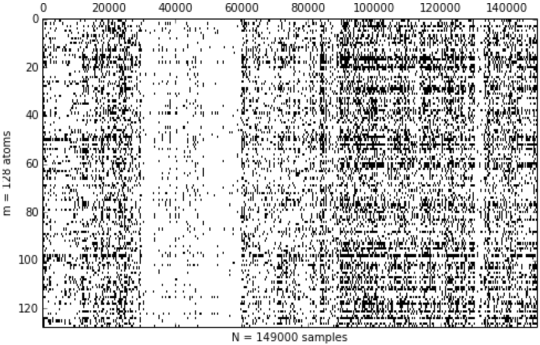
1. Tools: numpy, scipy, matplotlib, scikit-learn, h5py, librosa, PyUNLocBoX[1], IPython notebook, OpenStack lab cluster.
2. Notebooks: model construction, test on images, dataset conversion to HDF5, pre-processing, graph construction, auto-encoder model, features extraction, classification and test, 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



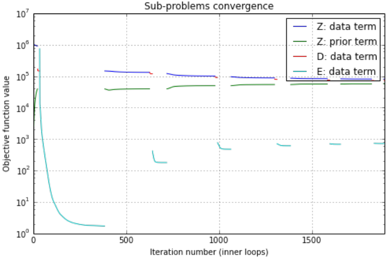
(a) $m = 128$ atoms of a learned dictionary.

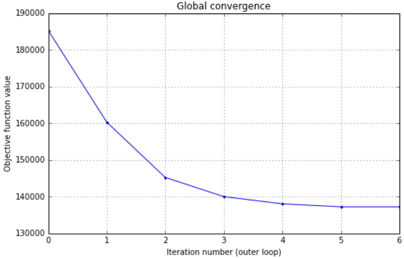


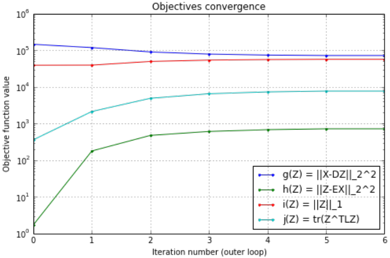
(b) A learned sparse (20% of non-zero coefficients) representation.

Figure: Learned dictionary **D** and representation **Z** of spectrograms.

Typical convergence







- Sub-problem objectives: $f_2(\mathbf{Z})$, $f_1(\mathbf{Z})$, $f_2(\mathbf{D})$ and $f_2(\mathbf{E})$.
- Sub-objectives: $f_d(\mathbf{Z}, \mathbf{D})$, $f_e(\mathbf{Z}, \mathbf{E})$, $f_z(\mathbf{Z})$ and $f_g(\mathbf{Z})$.
- Global objective $f_d(\mathbf{Z}, \mathbf{D}) + f_e(\mathbf{Z}, \mathbf{E}) + f_z(\mathbf{Z}) + f_g(\mathbf{Z})$.

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 λ_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[3][4].

[3] http://nbviewer.ipython.org/github/mdeff/dlaudio_results
[4] https://lts2.epfl.ch/blog/mdeff

Classification accuracy

Noise level (standard deviation)	0.0	0.1	0.2
Accuracy using CQT spectrograms [%]	69.7	58.7	46.9
Accuracy with $\lambda_g = 0$ [%]	75.9	57.1	42.6
Accuracy with $\lambda_g = 100$ [%]	78.0	65.9	51.6

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

Conclusion

- 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 ?