ATIAM 2017 - ML Project
Disentangling variation factors in audio samples

Adrien Bitton, Philippe Esling[*]
Institut de Recherche et Coordination Acoustique Musique (IRCAM)
UPMC - CNRS UMR 9912 - 1, Place Igor Stravinsky, F-75004 Paris
{adrien.bitton, esling}@ircam.fr

Abstract

Observations of complex data in the real world might be caused by several underlying generative factors that account for specific perceptual features. However, these factors are usually entangled and cannot be underlined directly from the data. Modeling such factors could generalize the learning of complex concepts through compositions of simpler abstractions. This enables us to understand the inner structure of the data, to process it efficiently and to control meaningful generative processes which may eventually open up on artificial creativity. An extensive body of research has been carried in the field of computer vision through the *Variational Auto-Encoders*. The goal of this project is to extend these recent approaches to sound and music data, by defining a procedural toy dataset of sound synthesis and then applying the recent β*-VAE* and *SCAN* approaches to these data.

1 Introduction

Unsupervised learning methods model data and shape representations without explicit targets or direct assumptions on the generative processes. Amongst them, a key model is the *Variational Auto-Encoder* (*VAE*) Kingma and Welling [2013] and its variants. In these, an encoded space is learned under certain constraints and regularizations. Encoder and decoder networks are jointly trained in order to fit such objectives while being able to retrieve the original data. As an example, we may refer to dimensionality reduction or data compression: a reduced code that explains the data and naturally highlights its salient properties. Recently, the β*-VAE* Higgins et al. [2016a] was developed to target disentangling these factors by learning independent latent variables accounting for distinct generative processes. The subsequent factorized latent variables are then evaluated and referred as primitives. An hyper-parameter balances these desired properties with the quality of data reconstruction. Second, the *SCAN* method Higgins et al. [2017] pairs a second *VAE* in order to build a parallel symbolic model that would be grounded into the real world observations. The technique proposed is to decompose complex symbolic concepts into a dictionary of elementary symbols while constraining this representation to match the one of the grounded primitives (to be directly extracted from audio samples). Hence a second divergence imposed on the *SCAN* objective between the two latent spaces put in correspondence.

The original variational autoencoder has led to several implementations Kingma and Welling [2013], Higgins et al. [2016a,b], Burda et al. [2015]. These different techniques relate to representation learning derived from variational inference and may support the association process to symbolic modeling as developped in *SCAN* Higgins et al. [2017]. In this project, we propose to extend these approaches by applying them to audio data generated procedurally.

[*] https://esling.github.io/

As a result, a bi-directional inference may be processed through the *SCAN* network: traducing a sound into a symbolic concept and generating sound from symbolic instructions. The semi-supervised learning benefits from the robustness of the primitives built on the unlabeled data, the authors report that in the case of a procedural synthetic image dataset only few examples needed to be explicitly labeled with a concept in order to successfully train the *SCAN* network.

From a generative point of view, this enables a rich diversity as while setting the relevant factors to the required values of the concept, all other parameters may be randomly sampled to produce a great variety of examples relating to that given concept. By learning compositional symbolic operators, the authors even induce a form of artificial creativity through the network: modeling new concepts that were not comprised in the training observations.

In the case of audio and music data, sequential modeling is another key topic in order to account for the multiple temporal correlations shaping the perception of sound. Several models have been built on recurrent neural networks and may support a *SCAN* technique tailored for audio samples.

2 Toy dataset

Your first assignment, in order to gain an understanding of the problem, is to create toy datasets that will be used to test different models. The disentangling properties of the *β-VAE* were assessed on an image dataset that was procedurally synthesized according to `https://github.com/deepmind/dsprites-dataset`. Five generative factors were sampled and combined, explicitly defining a set of training data from which the network should autonomously recover the corresponding disentangled latent variables. Hence, we will try to mimic this type of *procedural generative dataset* by adapting this reasoning to audio samples. This toy dataset will then be used to analyze the models in simplified setups in order to understand their behavior.

Exercice 1 - Designing a toy dataset An equivalent of *dSprites* should be built as a set of audio samples defined by pre-sampled sound parameters (such as pitch, level, spectral shapes, harmonics-to-noise ratio). A procedural generator must be developed using DSP libraries (such as *Librosa* `https://librosa.github.io/librosa/` and *Pyo* `https://github.com/belangeo/pyo`) and techniques such as the *Sinusoidal Modeling Synthesis*. An important aspect to keep in mind is the independence and the complementarity of such factors so that the network might be evaluated in its efficiency at disentangling them. Therefore, defining a toy dataset is an open question and your proposed solution will be evaluated on the design and justification of your choices. Hence, we highly recommend that you use both your personal knowledge, signal processing books and eventually to propose some innovative approaches to the question. The resulting work should be a procedural synthesizer of audio samples and a corresponding toy dataset from which a network could be trained at disentangling generative sound factors.

1. Describe what (independent) factors of variations may exist in sounds and that should minimally be understood by any system trying to tackle sound generation
2. Describe different levels of complexity and how to combine them
3. Describe situations in which these could require multiple scales of time
4. Code different *procedural functions* that could generate large sets of examples following all of your observations in the previous questions
5. Generate your sets of data in a structured way
6. Explain how these sets can be labeled and analyzed

3 Models and expected work

In order to understand the models and to be able to code these by yourself, we will start by understanding the basic VAE, and perform their implementation by relying on the *Pytorch* framework for Python. Tutorials are available at `http://pytorch.org/tutorials/`

Exercice 2 - Learning Pytorch and understanding VAEs As you might see on the web, multiple tutorials propose some pre-developed layers for VAEs. However, this might not help you to understand their inner functioning. Therefore, we will try to better understand those architectures and to code their simpler version by ourselves. Here, we will rely on the tutorial `https://wiseodd.github.io/techblog/2016/12/10/variational-autoencoder/` (This tutorial is intended for Keras, but the same type can be found for Pytorch, however this tutorial gives you a very good understanding of the mathematical intuition behind the VAEs).

1. Based on the description of the tutorial, try to develop your own VAE (using the Pytorch layer definition `http://pytorch.org/tutorials/beginner/examples_nn/two_layer_net_module.html`). You can also use other sources of information.
2. Compare your models to the VAE results from Kingma and Welling [2013] on MNIST
3. Train your model on your toy datasets and compare performance
4. Analyze and explain the behavior of the models for different properties
5. Find interesting visualizations to understand this behavior

3.1 Analyzing the performances of β-VAE

Exercice 3 - Adapting the reference paper As you can see in the original paper Higgins et al. [2016a] (available at `http://www.matthey.me/pdf/betavae_iclr_2017.pdf`), the modifications needed for a β-VAE compared to a VAE is quite minimal. However, the interesting aspects of these papers lie in their analysis of the disentangling of factors of variation. Therefore, we will focus here on this disentanglement analysis but for audio data.

1. Based on the paper, try to re-implement the model.
2. Train all models on the toy datasets and evaluate their results
3. Extend the visualization techniques proposed in the papers to audio data
4. Evaluate its disentangling performance on this toy dataset.
5. Propose new visualization techniques to assess disentanglement
6. Analyze and explain the behavior of the models for the different factors

3.2 Improving the β-VAE for temporal analysis

One of the key aspect in all VAE models lies in the actual choice of the encoder and decoder. Usually, we start with a traditional Multi-Layer Perceptron (MLP) for both the encoder and decoder. However, these can be easily replaced by any model of any complexity, depending on the task at hand. As we are assessing here a set of sounds that can typically have a temporal component, we should rely on Recurrent networks to take this into account.

Exercice 4 - Extending to recurrent encoder/decoder To understand the different approaches to attention and sequential temporal processing in recurrent neural networks, you can read `http://colah.github.io/posts/2015-08-Understanding-LSTMs/` and `https://distill.pub/2016/augmented-rnns/` for a high-level insight. Then, we will rely specifically on the tutorial given at `https://medium.com/datalogue/attention-in-keras-1892773a4f22`

1. Based on the tutorial, implement the attention recurrent encoder/decoder in Pytorch
2. Replace the encoder and decoder in the β-VAE with your new model
3. Train all models on the toy datasets and evaluate their results
4. Perform the same disentangling performance evaluation as Exercice 3
5. Apply the same set of visualizations as Exercice 3
6. Analyze and explain the behavior of the models for the different factors

3.3 Assessing the SCAN model for audio data

As explained in the introduction, even though the β-VAE shows interesting properties of disentangling factors of variations, if those are not known in advance, the *semantics* behind the dimension cannot be directly uncovered (ie. the dimension are not labeled by their meaning). Recently, the *SCAN* model was proposed, to directly target this aspect Higgins et al. [2017]. The *SCAN* model is also developed intuitively in the following blog post `https://deepmind.com/blog/imagine-creating-new-visual-concepts-recombining-familiar-ones/` that we will use as a baseline for this exercise

Exercice 5 - Adapting SCAN to audio data After reading the blog post and corresponding article, you might see that the models need a dataset that provides *semantic concepts* but most importantly *compositionality* between these concepts. However, the toy dataset developed earlier does not provide these properties. Therefore, we will think on how to define a dataset for compositional audio features (e.g. effects) corresponding to a hierarchical symbolic structure that maps to these sound primitives. In a second step, we will work towards adapting SCAN to audio data.

1. Define what semantic properties could be delineated for sound samples
2. List the compositional operators that could exist in sound
3. Define a hierarchical symbolic structure that maps to these sound primitives
4. Extend your previous functional generators to create a new toy dataset matching these
5. Explain how the *SCAN* framework could be extended for audio samples
6. (Bonus) Implement the *SCAN* framework for audio samples
7. (Bonus) Evaluate the performance of your model

References

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016a.

Irina Higgins, Nicolas Sonnerat, Loic Matthey, Arka Pal, Christopher P Burgess, Matthew Botvinick, Demis Hassabis, and Alexander Lerchner. Scan: Learning abstract hierarchical compositional visual concepts. *arXiv preprint arXiv:1707.03389*, 2017.

Irina Higgins, Loic Matthey, Xavier Glorot, Arka Pal, Benigno Uria, Charles Blundell, Shakir Mohamed, and Alexander Lerchner. Early visual concept learning with unsupervised deep learning. *arXiv preprint arXiv:1606.05579*, 2016b.

Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015.