ML Project ATIAM
Ninon Devis
Philippe Esling

Effect Modeling

Distortion simplification for Eurorack

2021-11-18

ninon.devis@ircam.fr
philippe.esling@ircam.fr

Abstract

Generative systems are machine-learning models whose training is based on two simultaneous optimization tasks. The first consists in building a latent space, which provides a low-dimensional representation of the data, eventually subject to various regularization and constraints. The second is the reconstruction of the original data through the sampling of this latent space. These systems are very promising as their space is a high-level, *over-compressed* representation that can be used as an intermediate space for several tasks, such as visualization, measurement, or classification. The steps of this project are first to develop variational models to understand generative *effects space*, where each point of the space corresponds to an audio sample. In our case, we will focus on modeling the effect of *different types of distortion* on samples. Then, the implementation of specific topological constraints between pairs of distorted and non-distorted sounds aims to provide a high-level *semantic understanding* of distortion. This model could then be embedded inside a Eurorack synthesizer with simple controls.

1. Overall presentation

Global instruction

Generalities

Deadline	January 2021
Organization	Group from 4 to 5 students
Deposit	`Github`

Project folders

`code/` with your code following the PEP8 coding style and organized in modules	
`report/` in PDF format	

Report

Redaction	LaTeX with a **given format style** that you will receive
Language	English
Maximum number of pages	8 pages following the scientific papers

Evaluation grid

Report - Including content, results and style	**7 pts**
Code - Accuracy, evaluation and coding style	**13 pts**

For more information on the project: ATIAM machine learning project webpage

2. Introduction

Among recent generative systems found in the literature, two have had a large success in the machine learning community. First, the *variational auto-encoder* (VAE) is based on a two-stage inference/generation procedure that showed great generalization properties and good reconstruction abilities despite of its light structure [5]. The second is the *Generative Adversarial Network* (GAN) (see [3]), that showed impressive reconstruction abilities but rather poor latent expressivity.
This project aims to study the use of VAEs to learn an *expressive* latent space for effect modeling, and to build a real-time synthesizer that can be used for artistic purposes. This study involves two main objectives:

1. *Obtaining the highest quality of reconstruction*. This implies to find an appropriate representation of the input data and design an appropriate model specifically designed for this learning task.

2. *Simplifying the use of a range of distortions*. It means analyzing the organization of the latent space regarding the application of distortion in order to develop a regularization approach allowing semantic control on these attributes.

3. Mel-based VAE

The first step with variational auto-encoders will be to implement them on simple data in order to understand their inner workings. This will allow you to decide the implementation details that you will need to use for the future.

Exercise 1: Learn Pytorch and useful libraries. This exercise requires learning basic programmation skills that you will need for the following work.

1. Install and learn PyTorch through basic tutorials `http://pytorch.org/tutorials/`

2. Install and read about the support libraries for audio processing: Librosa[1] for handling data, PySox[2] and Pedalboard[3] for audio effects.

Exercise 2 : code the VAE. You will now code your very own variational auto-encoder. As the main article is quite cryptic on how to code this, you can rely on the tutorial `https://wiseodd.github.io/techblog/2016/12/10/variational-autoencoder/` that is in Keras but will give you a good intuition for the mathematics behind the system.

1. Based on the tutorial or another source you will find, develop your VAE in PyTorch.

2. Compare your models to the original VAE results from Kingma & Welling [5] on the MNIST dataset.

3. Implement warm-up [6] by yourself and make a qualitative analysis when varying the β parameter between 0 and 4.

4. Play with parameters, and make quick assumptions on the results.

⋆ *Show your code and results to your supervisors* ⋆

[1] `https://librosa.org/doc/latest/index.html`
[2] `https://pysox.readthedocs.io/en/latest/`
[3] `https://github.com/spotify/pedalboard`

4. Model

We will now aim to design a real-time effects modeling system based on the variational auto-encoder framework. The next step will be the in-depth study of the generative process to ameliorate the model in terms of interaction and expressiveness. Several audio dataset are available for the training, such as NSynth[4] or StudioOnLine (IRCAM orchestral instrument dataset)

Exercise 3: Data-processing. A mandatory first step for adequate learning is to pre-process the data into known ranges and properties. Indeed, contrary to images where we directly give raw information to the model, we usually rely on specific representations, here we will focus on the Mel spectrum (with different factors of compression). In the specific case of this project, we need to develop an *efficient and modular* way to apply audio effects on different samples.

1. Draw a list of possible audio effects and how to best process chains of effects.
2. In the specific case of *distortion*, list all types of distortions and their respective parameters.

   ⋆ *Expose us your ideas before the next step* ⋆
3. Implement a versatile and modular approach to apply different types of audio effects with various parameters to an input dataset.
4. Code the dataloader to compute your data along with all the data-processing you find relevant to simplify the learning.

Exercise 4: Reconstruction. Design your own VAE model, which should be able to give good reconstructions of your audio data. In this first step, we will simply provide *all samples* (with or without effects) as input to the training.

- You should have one module for your model and one module for the training procedure.
- Think about the nature of audio and effects to define your networks.
- Play with the hyper-parameters in order to find the best configuration for the learning.
- Plot the latent space with t-SNE and analyze the distribution of sounds.
- Analyze potential relations based on effects parameters.

5. Topology and control

Starting from here, the project has a rather "open-ended" nature, as no perfect solution exists yet for this type of modeling. Hence, we propose three different directions of studies, which you should analyze *prior to implementation*. This implies to ponder on the properties (advantages and flaws) of the different directions and select the one that you find more appropriate.

Direction 1 : Timbre transfer approach. A simple approach to effects modeling is to consider it as a sub-case of the more generic *timbre transfer* approach [1]. Hence, in this direction we give the original sound to the VAE and ask it to reconstruct the modified sound.

Direction 2 : Latent space topological constraints As we have both original and modified sounds, an interesting direction would be to constrain our learning in order to include specific organization of our latent space. In a spirit similar to this paper [4], we can impose either a *specific dimension* (or group of dimensions) to reflect the effect. In that case we compute differences between vectors linking samples.

[4] https://magenta.tensorflow.org/datasets/nsynth

Direction 3 : Space matching (world model) approach In a spirit similar to the *FlowSynth* model [2], we can consider the relation that exists between the latent space and the corresponding parameters of the audio effect. In that case, we only give modified sounds and try to regress to our effect parameters.

Exercice 5 : Research direction

- List all the attributes that you find interesting or problematic with the different proposed directions. ⋆ *Tell us your genius ideas before the next step* ⋆
- Implement a module for implementing your favorite solution.
- Evaluate the success of your solution.

Exercice 6 : Eurorack embedding More than tendencies, we would like to have a real control -as a virtual knob- on these approaches for effects modeling. Ideally, the model should also be light enough to run on an embedded platform. If times permit, we will embed your proposed solution inside Eurorack.
⋆ *If you succeed, congratulations! Let's write a scientific paper together about your results* ⋆

References

[1] Adrien Bitton, Philippe Esling, and Axel Chemla-Romeu-Santos. "Modulated variational auto-encoders for many-to-many musical timbre transfer". In: *arXiv preprint arXiv:1810.00222* (2018).

[2] Philippe Esling et al. "Flow synthesizer: Universal audio synthesizer control with normalizing flows". In: *Applied Sciences* 10.1 (2020), p. 302.

[3] Ian Goodfellow et al. "Generative adversarial nets". In: *Advances in neural information processing systems*. 2014, pp. 2672–2680.

[4] Hyunjik Kim and Andriy Mnih. "Disentangling by factorising". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 2649–2658.

[5] Diederik P Kingma and Max Welling. "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114* (2013).

[6] Casper Kaae Sønderby et al. "How to train deep variational autoencoders and probabilistic ladder networks". In: *arXiv preprint arXiv:1602.02282* (2016).