ML Project ATIAM

Ninon Devis
Axel Chemla–Romeu-Santos
Philippe Esling

Rhythms Generation

Control over semantic rhythms attributes

2020-11-24

ninon.devis@ircam.fr
chemla@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 find generative *rhythms synthesis space*, where each point of the space corresponds to a new rhythmical score that comes from the *high-level* understanding of the input data. Then, the implementation of a recent adversarial approach is aimed to provide a certain *control* over high-level *semantic musical attributes* of rhythm. The main goal is to develop a new creative tool that strives to break the common process of rhythms composition.

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 [2]. The second is the *Generative Adversarial Network* (GAN) (see [1]), 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 rhythms synthesis, 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. *Controlling the synthesis of rhythms*. It means analyzing the organization of the latent space regarding some musical features in order to develop a regularization approach empowering rhythm generation with understandable semantic control on these attributes.

3. Rhythm 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 MIDI processing: Pretty MIDI[1] for handling data and Music21[2] for music descriptors.

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 [2] on the MNIST dataset.
3. Implement warm-up [3] 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://craffel.github.io/pretty-midi/
[2] http://web.mit.edu/music21/

4. Model

We will now aim to design a real-time rhythms synthesis 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 MIDI dataset are available for the training, such as DrummerNet[3] or this archive.

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, especially in the case of symbolic MIDI data.

1. Draw a list of possible rhythms representations that we could give to the VAE and chose the one that seems convincing in term of rhythm expressiveness and learning performance.

   ⋆ *Expose us your ideas before the next step* ⋆

2. 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 rhythm data.

- You should have one module for your model and one module for the training procedure.
- Think about the nature of rhythms to define your networks.
- Play with the hyper-parameters in order to find the best configuration for the learning.

5. Control

Exercise 5 : Latent space study As rhythms synthesis is achieved by the exploration of the VAE latent space, its topology is thus very important in terms of expressiveness and interaction. The main objective is to achieve similar results with rhythm than this paper on melodies. We will thus go deeper into the understanding of the latent space construction.

- List all the semantic attributes that you find interesting to play with for rhythm composition.
  ⋆ *Tell us your genius ideas before the next step* ⋆
- Implement a module for analyzing this semantic attributes thanks to the aforementioned reference paper, producing your own code.
- Look at the evolution of the chosen descriptors along some dimensions of latent space.

More than tendencies, we would like to have a real control -as a virtual knob- on these semantic attributes for the rhythm generation. The idea is to combine adversarial learning with your VAE as explained in this article. This procedure forces the latent space to abstract from the selected features and then reintroduce them as conditioning to the generative function in order to obtain a continuous control.

- Read carefully this gorgeous article and implement the adversarial consequently in your code.

⋆ *If you succeed, congratulations! Let's write a scientific paper together about your results* ⋆

[3] https://github.com/keunwoochoi/DrummerNet

References

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

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

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