Pôle Projet

DDSP Preject Report

Peizhou ZHANG
Molin LIU
Xinjian OUYANG
Haoyu YU
Supervisor : Simon LEGLAIVE

CentraleSupélec - 2A , Campus de RENNES

Juin 2021

1 Introduction

1.1 Motivation

Nowadays, the concept of deep learning is one of the most powerful tools in almost every domain, including audio signal processing. In this article we present the work focusing more specifically on the audio synthesis.

Although a normal synthesizer has already been benefit from the traditional tools of audio generation and perception without deep learning, yet it still faces problems and has room for improvement : the controllability of sound effect and the limited expressivity, etc.

Under such kind of circumstance, many researches have been carried out to take the advantage of deep learning methods in audio synthesis. Indeed, the newly developed audio synthesizers (with deep learning methods implemented) overcome previous drawbacks but they still have their own shortcomings. For example, some deep learning models are limited by bias and not incompatible with perceptual losses, some others require an enormous network (more than 50M parameters for instance).

So a more efficient, modular and controllable method which could combine all the strengths of precedent strategies is demanded, and Google MAGENTA provided a satisfying solution in 2020 : DDSP

1.2 DDSP

DDSP refers to Differentiable Digital Signal Processing, which is based on differentiable strategy of common DSP functions like synthesizers and filters. As is discussed above, it is an advanced method which takes into the advantages of precedent ways :

- It makes it possible to realize the manipulation of a given sound input ;
- It helps to train high-quality audio synthesis models with a small scale of data ;
- It controls the pitch and loudness independently during the synthesis.
- It has smaller network size than precedent neural synthesizers ;

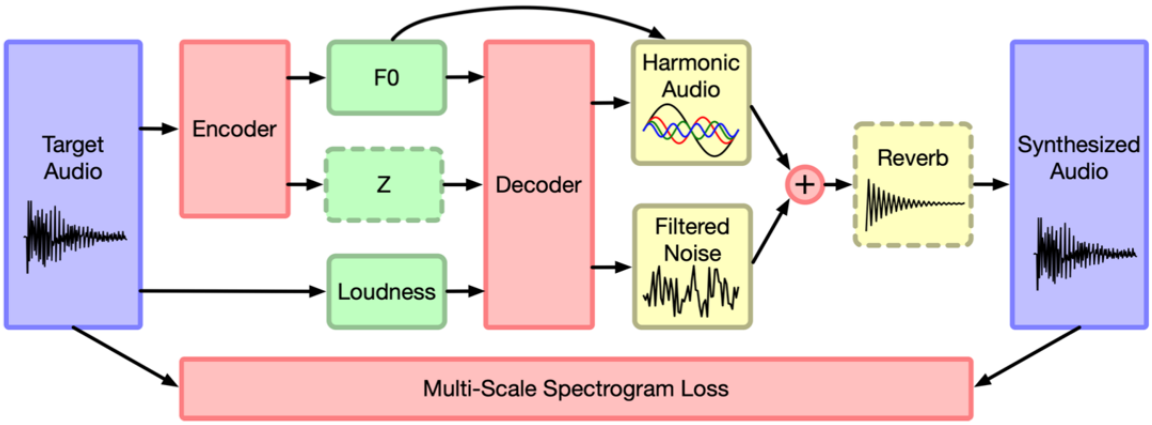

FIGURE 1 – DDSP Architecture

A closer look into the DDSP architecture is shown in Figure 1. The red components represent the neural network architecture ; green components represent the latent functions or variables, among which f_0 refers to the pitch of sound and Z describes others features like timbre ; yellow components are the deterministic synthesizers and effects, which is similar with the traditional DSP elements.

1.3 Objectives

Since the DDSP is already a complete library,our first goal of the project Semester_8 is to study available references of DDSP and have a deep understanding of it. We are supposed to get familiar with the DDSP implementation. We choose Tensorflow in this project Since DDSP library is coded by Tensorflow.

While after that, we need to gather sound resources which are used as training set, and train our own synthesizer models, which is also a process to verify the official DDSP library.

Most importantly, it should be noticed that the current DDSP is not a truly generative model, which means that it still needs an input audio to generate an audio. What we want to achieve in the end is to make DDSP truly generative, given only pitch (f_0), loudness and Z information (without audios as input). To make it easier to understand, it can be imagined that, in Figure 1, the Target Audio and Encoder are removed, while the residuals are remained, with f_0, Z and Loudness being inputs.

2 Methodology

2.1 DDSP Training

As Figure 1 shows, DDSP autoencoder is an end-to-end training process. The target audio is processed by encoder, decoder and synthesiser sequentially, then a synthesized audio is the output.

For more technical details about DDSP training process, please refer to the official DDSP paper [1]. The most significant part of this report is to introduce how we make DDSP generative.

2.2 GMM over Z datasets

To make DDSP generative, Z information besides pitch and loudness is needed. Thanks to GMM (Gaussian Mixture Model), a probability distribution over "Z" can be generated by fitting Z datasets in GMM. Z datasets are extracted from audios using pretrained DDSP model.

GMM is an unsupervised learning algorithm as well as a clustering algorithm. As its name implies, each cluster is modeled according to a different Gaussian distribution.

The objective function of GMM is to maximize the log likelihood L for the data X, $p(X)$. By assuming a mixture of K gaussians to have generated the data, $p(X)$ can be written as marginalized probability, summed over all K clusters for all data points [2].

$$p(x_i) = \sum_{k=1}^{K} p(x_i|c_k)p(c_k) \tag{1}$$

$$p(X) = \prod_{i=1}^{N} p(x_i) = \prod_{i=1}^{N} \sum_{I=1}^{K} p(x_i|c_k)p(c_k) \tag{2}$$

$$L = log(p(X)) = \sum_{i=1}^{N} log(\sum_{k=1}^{K} p(x_i|c_k)p(c_k)) \tag{3}$$

Notice that it's difficult to find an analytical solution of L, and there is an elegant solution to this problem, which is Expectation Maximization(EM) algorithm. However, we don't detail it here.

After getting a probability distribution of Z using GMM method, we will implement some sampling methods to get Z data, which will be used as the input in the DDSP generative model.

2.3 Sampling Methods over Z Distribution

2.3.1 Direct Sampling

Scikit-learn Python library provides a Gaussian Mixture class which allows estimation of the parameters of a Gaussian mixture distribution, and also, sample directly from fitted GMM.

This method samples Z data points randomly from all the clusters of GMM, which results in some problems in the effects of DDSP output sounds, such as noise and the inconsistency of timbre. These problems are already observed in our experiments.

2.3.2 Inverse Transform Sampling

To avoid the drawbacks of direct sampling, inverse transform sampling is implemented to sample Z from fitted GMM.

Inverse transform sampling is a basic method for pseudo-random number sampling, $i.e.$, for generating sample numbers at random from any probability distribution given its cumulative distribution function.

More precisely, inverse transform sampling takes uniform samples of a number u between 0 and 1, interpreted as a probability, and then returns the largest number x from the domain of the distribution $P(X)$ such that $P(-\infty < X < x) \leq u$. [3]

Inverse transform sampling method is implemented for a certain gaussian cluster to sample Z data points. We can get K synthesized audios in total, thanks to the implement of this sampling method for all the clusters respectively (K is the number of clusters in GMM).

In our experiments, the noise is largely reduced and the timbre of sounds is much more consistent and "smooth".

3 Results

First of all, a DDSP model with Z encoder included is trained. Using this model, The Z information is extracted from input piano audios as a Z dataset. After fitting the Z dataset in GMM, a probability distribution of Z are generated. With a define distribution of Z, we can implement the sample methods mentioned in section 3.3 to get Z predictions.

Loudness and pitch are defined as fixed value in order to better observe the results.

All of the codes and results are shared in Github [4]. The main results of our work $i.e.$ making DDSP truly generative are presented as followed.

- Configuration :
  The training datasets are piano sounds. And the Z dataset for GMM fitting is also extracted from piano sounds.

  The number of GMM clusters is 10 and the weight of each is [0.08038034, 0.12804873, 0.03591031, 0.07024887, 0.13935333, 0.25402857, 0.06130376, 0.08094855, 0.08358932, 0.06618821].

  Loudness and pitch are fixed-value sequences, which are 0.39189076 and 0.4487863 repectivelyly.

— Direct Sampling from all the clusters

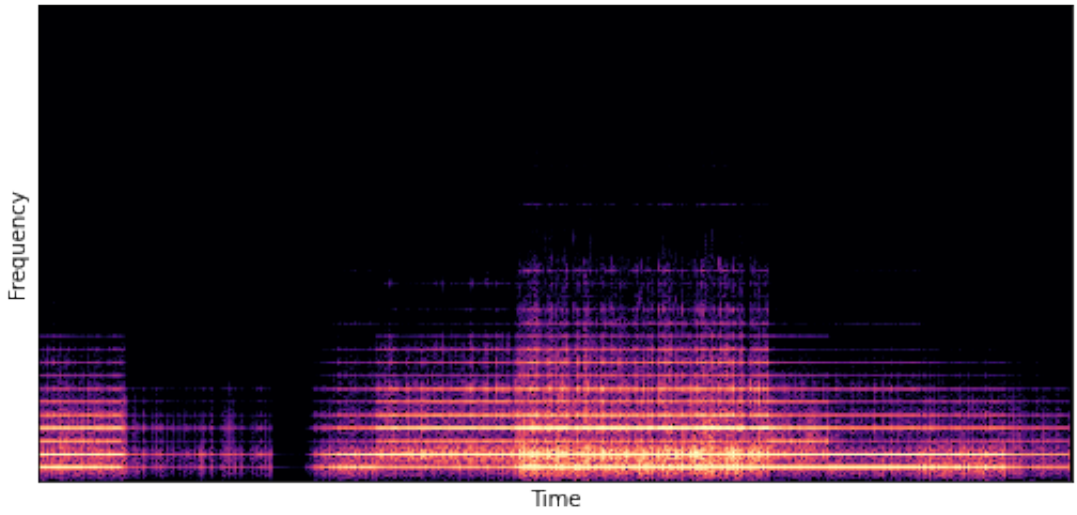

Figure 2 – Spectrogram of audio 0

It can be seen from the spectrogram of audio 0 that, in Figure 2, there are lots of noisy points and the spectrogram is not smooth.

— Inverse Transform Sampling from each cluster

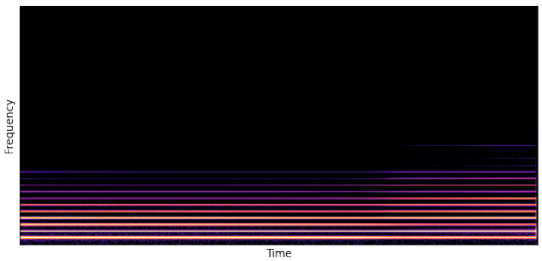

Figure 3 – Spectrogram of audio 1(cluster 1)

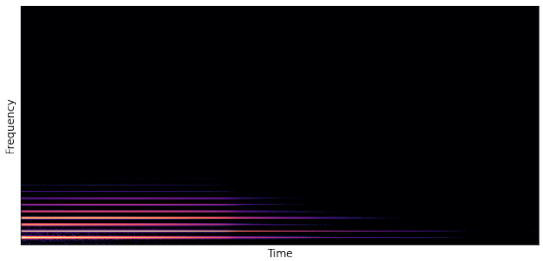

Figure 4 – Spectrogram of audio 2(cluster 2)

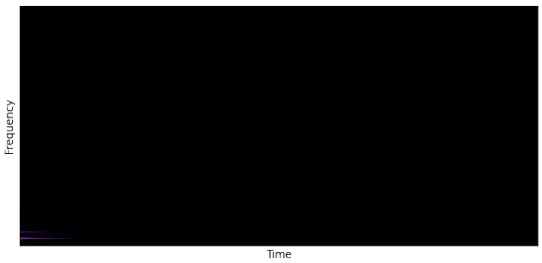

Figure 5 – Spectrogram of audio 3(cluster 3)

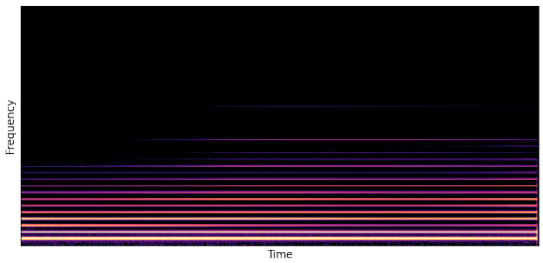

Figure 6 – Spectrogram of audio 4(cluster 4)

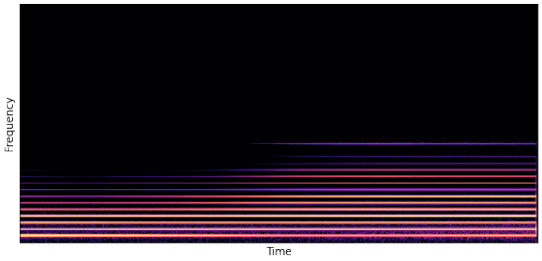

FIGURE 7 – Spectrogram of audio 5(cluster 5)

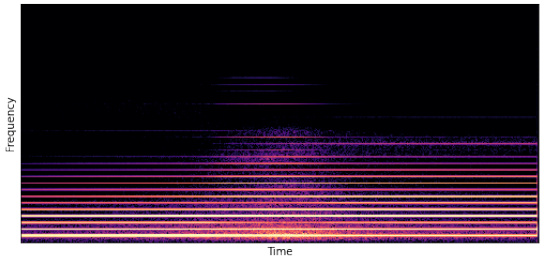

FIGURE 8 – Spectrogram of audio 6(cluster 6)

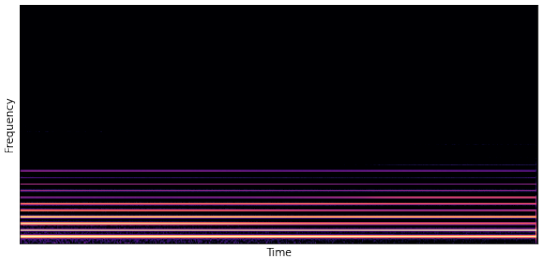

FIGURE 9 – Spectrogram of audio 7(cluster 7)

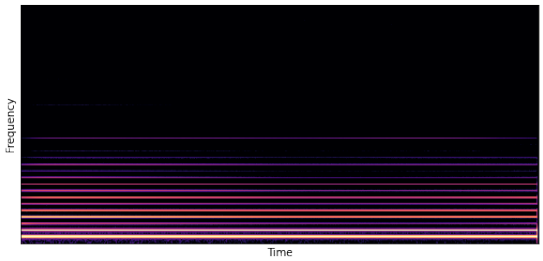

FIGURE 10 – Spectrogram of audio 8(cluster 8)

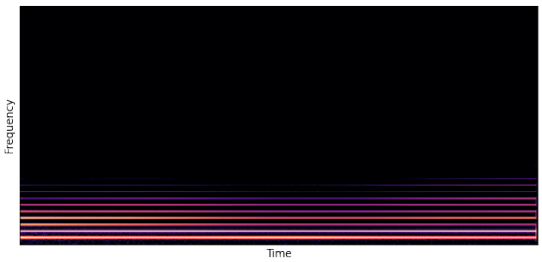

FIGURE 11 – Spectrogram of audio 9(cluster 9)

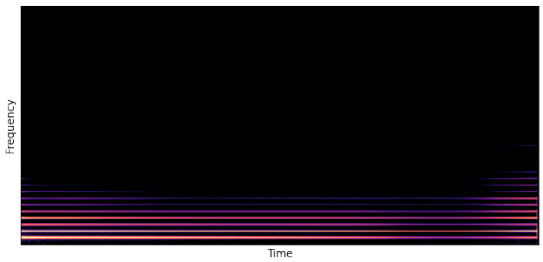

FIGURE 12 – Spectrogram of audio 10(cluster 10)

It can be seen from Figure 1-10 that there are less noisy points in the audios whose Z information is sampled by inverse transform sampling. And also, the timbre is much more consistent.

Another thing to be noticed, from Figure 5, is that audio 3 is an invalid sound since there are few useful information we can see from its spectrogram, with only noise points left. In fact, the Z information of audio 3 are sampled from cluster 3 whose weight is minimum among all the clusters. There may exist certain relationship between them.

4 Conclusion

Under a schedule of two semesters (290 HEE), we finally achieved all the targets we set in the beginning. Now we not only have a full-scale understanding of DDSP, but also make it a truly generative module.

Our work successfully realizes the prolongation of DDSP, which could be meaningful for different potential applications of audio synthesis, not to mention its obvious advantages compared with other audio synthesizing methods. For example, it could be a possible way in synthesizer music production; it could also be an ideal choice for voice disguise and artificial customer service.

On the other hand, there is still room for the improvement of this project. Firstly, a quantitative evaluation of synthesized audios can be made besides the qualitative one. Also, the generative DDSP model should be improved in the further work, in order to make its output audio more like a instrumental sound.

In a word, a truly generative DDSP has great potential in audio synthesis.

While from the whole process, we, as participants, also gained some values personally. Our knowledge about signal processing, practical skills in Deep Learning and programming have been upgraded. Besides, a team cooperation of people who speak the same language indeed makes it more efficient to execute, and we also realized the importance of a well-organized planning and regular communication with the supervisor. That explains why we could finish our project as expected in a smooth rhythm.

5 Distribution

Preliminary Work	Everyone
Model training	Xinjian OUYNAG ; Haoyu YU
GMM	Xinjian OUYNAG
Decoder	Peizhou ZHANG ; Xinjian OUYANG ; Molin LIU
"Generative" Realization	Xinjian OUYANG ; Peizhou ZHANG
Report	Molin LIU ; Xinjian OUYANG
Slides for presentation	Peizhou ZHANG ; Molin LIU

6 References

[1] J. Engel, L. Hantrakul, C. Gu, A. Roberts, "DDSP: Differentiable Digital Signal Processing", International Conference on Learning Representations (ICLR), 2020

[2] Ribhu Nirek, Gaussian Mixture Models(GMM): Understanding GMM: Idea, Maths, EM algorithm python implementation

[3] Wikipedia contributors. "Inverse transform sampling." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 22 Mar. 2021. Web. 29 May. 2021

[4] DDSP project, Github Resources

[5] Simon Leglaive, "AI x Audio (Differentiable Digital Signal Processing for Artificial Sound Generation"