Realtime instrument interpolation using Differentiable Digital Signal Processing
ATIAM Machine Learning Project

Antoine Caillon[1], Théis Bazin[1], Philippe Esling[1]
[1] Institut de Recherche et Coordination Acoustique Musique (IRCAM)
UMPC - CNRS UMR 9912 - 1, Place Igor Stravinsky, F-75004 Paris
{caillon, bazin, esling}@ircam.fr

November 2019

Abstract

Most generative models of audio directly generate samples in one of two domains: time or frequency. While sufficient to express any signal, these representations are inefficient, as they do not utilize existing knowledge of how sound is generated and perceived. A third approach (vocoders/synthesizers) successfully incorporates strong domain knowledge of signal processing and perception, but has been less actively researched due to limited expressivity and difficulty integrating with modern auto differentiation based machine learning methods. The use and potential of such approaches are to be evaluated during this Machine Learning project.

1 Introduction

The Differentiable Digital Signal Processing (DDSP) [1] model comes from a pretty recent article (september 2019) proposing a harmonic + noise synth based architecture to generate raw waveform (see figure 1).

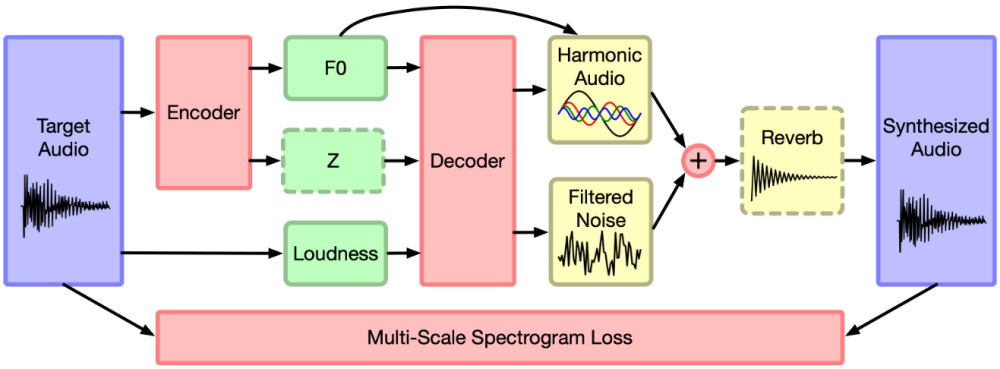

Figure 1: Overall architecture of the DDSP model

While this model has proven itself to be very powerful at reconstructing a single instrument, we still don't know if the model is able to properly use a latent space in order to interpolate between different instrument, hence you will have to reimplement it, reproduce the original article's results, and try to make the model go further by adding a latent space and training it on a multi-instrument dataset.

2 Datasets

One of the main advantages of the DDSP model is its ability to train on a very little amount of data ($\approx$ 15mn of audio is more than enough), thus we leave you the pleasure to collect and process audio from an instrument of your choice (preferably monophonic and harmonic)[1]. When building the multi instrument dataset, keep in mind that it is important to keep the dataset *balanced*.

3 Expected work

Preliminary work

You will be using PyTorch during this project, and we encourage you to try out this tutorial. When ready to begin, make sure your dataset is ready and write a DataLoader class using pytorch base class (pytorch.docs.DataLoader) that you will use during your training. For parameters extraction such as loudness or fundamental frequency you can use libraries like *librosa, crepe...*

Exercise 1 - Benchmark of the DDSP synthesizer

1. Validate data pre/post processing (f0 and loudness estimation)
2. Implementation of multiple variations of the DDSP synthesizer (harmonic only, harmonic + noise, reverberation)
3. Evaluate reconstruction quality and training stability for those variations

Exercise 2 - Multiple instrument synth

1. Add a probabilistic encoder to the model [2] (KL regularized latent space)
2. Monitor latent space during generation
3. Evaluate the model interpolation abilities

Exercise 3 - Exploring context

The DDSP's base conditioning is fundamental frequency f_0 and loudness l. It might be interesting to test other types of conditioning, such as spectral centroid, periodicity, mfcc, roughness, ...

1. Train multiple DDSP using different conditioning signals
2. Evaluate the performances of each variation for several tasks: reconstruction, conversion, control...

Exercise 4 - BONUS

Make it a real-time synth using the wrapper of your choice ! You can either use the pythonic way (with sounddevice blocking mode and some OSC / midi libraries), or wrap it inside a PD / MAX external... The choice is yours !

References

[1] Anonymous. DDSP: Differentiable Digital Signal Processing. September 2019.

[2] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. *arXiv:1312.6114 [cs, stat]*, May 2014. arXiv: 1312.6114.

[1] Remember that lots of things can be found on youtube, and that *youtube-dl* is your friend.