Learning controls and interactions for DDSP

Antoine Caillon[1], Adrien Bitton[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, bitton, esling}@ircam.fr

November 2019

Abstract

Most generative models of audio directly synthesize samples in one of two domains: waveform or time-frequency. While sufficient to express any signals, these representations are inefficient as they do not utilize prior knowledge of how sound is produced and perceived. The Differentiable Digital Signal Processing (DDSP) model generates audio based on an additive sinusoidal synthesizer summed with a subtractive noise synthesizer (SMS decomposition), which are jointly controlled by a neural network. Conditioning signals of f0 and loudness are extracted from an audio at a 100Hz frame rate, then processed and upsampled by the DDSP decoder which infers time-varying control parameters for the output synthesizers. It achieves high audio quality with an unprecedented efficiency and has been used for creative purposes such as timbre transfer. However, its use for sampling and composition is limited since it requires fine-grained and realistic input envelopes. In this project, we will study several approaches aiming at generating expressive conditioning signals based on a quantized input (event classes, MIDI score) or in an unconditional fashion. Given a pretrained DDSP generator, the experiments will focus on learning an upstream control model and proposing new user interactions.

1 Introduction

The Differentiable Digital Signal Processing (DDSP) [1] library proposes a neural waveform generation model with a decoder output based on an harmonic plus noise synthesizer. As shown in figure 1 the model predicts the control parameters of both synthesis components which are summed and optionally fed into a learned reverberation module.

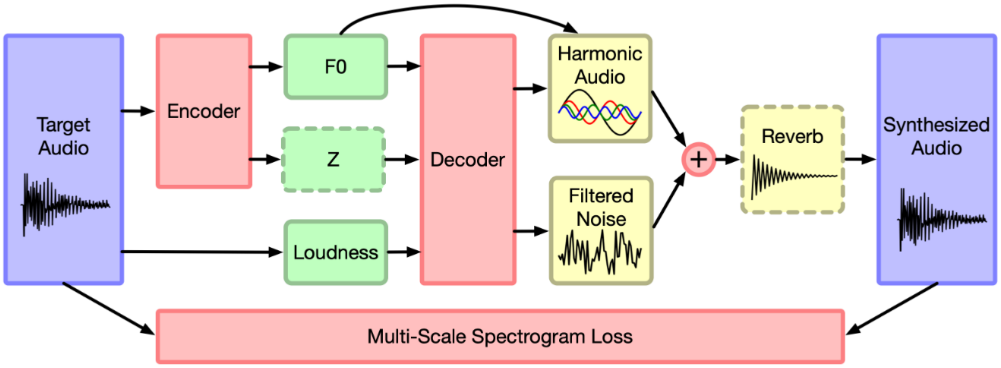

Figure 1: Overall architecture of the DDSP model

For the scope of this project, we consider the supervised decoder model that is trained as follow:

- analysis of an input audio by extraction of the loudness and fundamental frequency curves
- synthesis of the audio by processing and upsampling of the analysis features into control parameters of the output synthesizer
- optimization by audio reconstruction in the spectrogram domain (multi-scale magnitudes)

This model achieves high audio quality in a light-weight architecture that trains and synthesizes fast. When learning the supervised model on a single instrument, it is possible to perform timbre transfer by analyzing features from an audio recording of another instrument and resynthesizing its loudness and pitch envelopes with the training instrument timbre. This is an interesting creative application of the model which allows converting an instrument solo into the corresponding performance of another instrument. Yet, this model has not found concrete application for music composition. As shown in this demonstration video, playing DDSP with a keyboard requires additional user modulations to create expressivity and cannot effectively mimic the effect of an instrument performance. This information is encoded in the detailed loudness and pitch curves extracted from real audio, whereas keyboard and piano-roll representations only specify quantized pitch and velocity.

The goal of this project is to learn a higher-level control model for DDSP which would convert a quantized score into realistic pitch and loudness envelopes (e.g. see figure 2). A pretrained DDSP model will then generate the corresponding audio and create a plausible performance of the input melody. In the first place the control can be learned on individual notes which have specific envelopes such as attack/decay/sustain/release or pitch vibrato. This sampler model could be used to generate notes that are concatenated together to make melodies. A real performance has an articulation which not only affects notes individually but as well across the whole phrase and according to relationships between notes. In the second place, experiments will attempt to learn a control model for variable-length note sequences. In both cases, the evaluation should assess three main qualities:

- accuracy: the generated frequency and loudness envelopes should preserve the melody input, in average
- expressivity: the envelopes should fluctuate around the average target in a natural sounding way (e.g. modulations such as vibrato and tremolo, percussive envelopes such as pizzicato)
- diversity: for a given set of pitch and velocity combinations, the model should output several possible performances which could eventually be controlled (e.g. choosing a playing style)

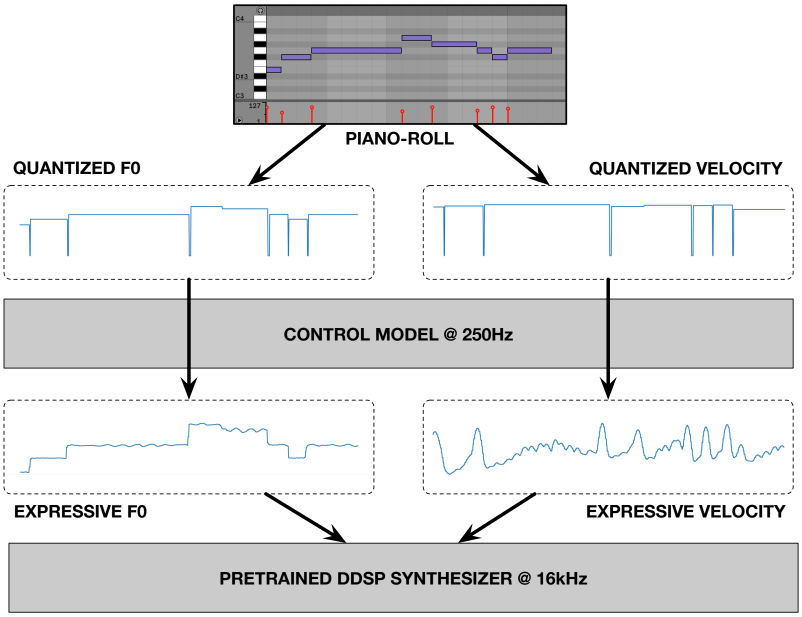

Figure 2: A possible approach to composing with the DDSP model

2 VAE MNIST

You should first get familiar with PyTorch and generative models such as the Variational Auto-Encoder [2]. You will implement and train your own VAEs on MNIST which is a rather simple collection of hand-written digits from 0 to 9. You can compare different architectures (e.g. linear layers, convolutions, batch-normalization etc.) as well as training on harder datasets (e.g. Fashion-MNIST). Once the model has achieved a reasonable reconstruction quality, you can generate new data by sampling and interpolation in its latent space.

3 VAE for fixed-length samples

As observed in the first exercise, the trained VAE decoder can be used as a generator of new data. We propose you to train a VAE model at reconstructing pairs of frequency and loudness envelopes extracted from individual notes of fixed length. You will need to:

- define your fundamental frequency and loudness analysis
- choose and prepare a dataset
- train a VAE at reconstructing pairs of control envelopes
- test it with a pretrained DDSP model for audio synthesis

4 VAE for conditional samples

In order to control the pitch and velocity, you should extend your previous experiment by training a VAE with a conditional decoder:

- collect discrete pitch and velocity labels for each note
- add conditioning to the decoder
- evaluate the accuracy and expressivity of conditional generation

As a result, you can play melodies by concatenating the envelopes of each note and synthesizing the corresponding audio with DDSP. When there is no note activations, you can assign silence in the envelopes as 0Hz frequency and -120dB loudness. The overall phrase may not sound totally natural because of this, you can possibly improve that with fade-in/out between note events.

5 Recurrent approach for variable-length samples

The expressivity of a real instrument performance is encoded in the individual note envelopes as well as in the articulation with respect to note relationships. Your previous models are not able to capture this expressivity since they trained on individual notes. To address this challenge, we propose you to train a recurrent model (RNN of your choice) on note sequences:

- choose and prepare a dataset with note-level annotations of solo instrument performances
- train a RNN model at generating frequency and loudness envelopes from input note sequences
- evaluate the accuracy and expressivity of the variable-length generation

6 OPTIONAL: Adversarial approach

Generative Adversarial Networks have recently achieved impressive results in image [3], video [4] and sound generation [5]. For this part of the project you will study the use of an adversarial approach to increase the expressivity and naturalness of the previous models, by adding an adversarial objective [6] and a noise sampling to allow multimodal outputs. Compare the results from both models, and discuss the pros and cons of the adversarial approach for this task.

References

[1] Jesse Engel, Lamtharn Hantrakul, Chenjie Gu, and Adam Roberts. Ddsp: Differentiable digital signal processing, 2020.

[2] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2014.

[3] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks, 2019.

[4] Jiawei Chen, Yuexiang Li, Kai Ma, and Yefeng Zheng. Generative adversarial networks for video-to-video domain adaptation, 2020.

[5] Chris Donahue, Julian McAuley, and Miller Puckette. Adversarial audio synthesis, 2019.

[6] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric, 2016.