Music Source Separation with DDSP

February 16, 2024

Davide Gabrielli

Abstract

When listening to music, we listen to a mixture of different instruments and vocals. Music Source Separation is the task of separating the different sources which compose a music track. In this work a novel approach for MSS is proposed, based on the **Audio Spectrogram Transformer** performing regression over the parameters of the **Differentiable Digital Signal Processing** in order to reconstruct the stem track of an instrument from the mixture.

1. Introduction

Music Source Separation (MSS) is a pivotal task in the field of audio signal processing, which aims to isolate individual sound sources within a complex musical composition. The ability to disentangle the mixture of instruments and vocals present in a recording is of great interest in many applications, such as music transcription, remixing, and automatic music transcription.

The model proposed in this work **(AST-DDSP)** is an end-to-end neural network that takes as input the mixture logmel spectrogram and outputs the reconstructed audio of an instrument source by using the Differentiable Digital Signal Processing (DDSP) (Engel et al., 2020a) synthesizer.

In particular, the contributions of this work are:

- The application of the Audio Spectrogram Transformer (AST) (Gong et al., 2021) to regression tasks.
- A novel architecture for MSS based on the AST and the DDSP Sinusoidal Synthesizer.
- The implementation of the Sinusoidal Synthesizer in PyTorch and some other core components of DDSP.
- The code of the proposed model on GitHub `https://github.com/davegabe/ast-ddsp-mss`.

Email: Davide Gabrielli <gabrielli.1883616@studenti.uniroma1.it>.

Deep Learning and Applied AI 2023, Sapienza University of Rome, 2nd semester a.y. 2022/2023.

2. Related Work

2.1. MSS in Spectrogram Domain

Many approaches to Music Source Separation (MSS) revolve around generating masks for individual sources and applying them to the mixture (Manilow et al., 2020). While this method is widely adopted, it has inherent limitations. Instruments occupying the same frequency range may pose challenges for the mask, resulting in incomplete separation. Additionally, reconstructing audio from the spectrogram, whether through inverse Short-Time Fourier Transform or neural vocoders, can introduce undesirable artifacts.

2.2. MSS in Waveform Domain

An alternative approach involves using neural networks to directly estimate the waveform of the source as for Demucs (Défossez et al., 2019). Or by combining both representations as for the recent Hybrid Transformer Demucs (Rouard et al., 2023).

2.3. DDSP for Inverse Audio Synthesis

DDSP is a differentiable synthesizer that can be used to generate audio from a set of parameters. In the work of (Engel et al., 2020b) the authors propose a method to train DDSP for **inverse audio synthesis**, so that given an audio signal the model is able to extract the parameters that can be used to reconstruct the original audio.

The model is composed by a Sinusoidal Encoder, which extracts the sinusoidal frequencies, sinusoidal amplitudes and filtered noise magnitudes using a ResNet-38 on the logmel spectrogram and synthesizes the audio using a Sinusoidal Encoder and a Harmonic Encoder, which use the output of the Sinusoidal Encoder to extract the fundamental frequency, amplitude and harmonic distribution.

In fact the authors based their implementation on the Harmonic Plus Noise Model (HPNM) (Serra & Smith, 1990) which is a model that can be used to represent an audio signal as combination of sinusoids in harmonic ratios of a fundamental frequency alongside a time-varying filtered noise signal, but this model can be used **only for monophonic sources**.

3. Method

The proposed method is an attempt to combine the advantages of the approaches described above, using the **logmel spectrogram domain** to extract the relevant information about an instrument from the mixture and use them to reconstruct it directly in the source in the waveform domain leveraging a synthesizer over the DDSP parameters.

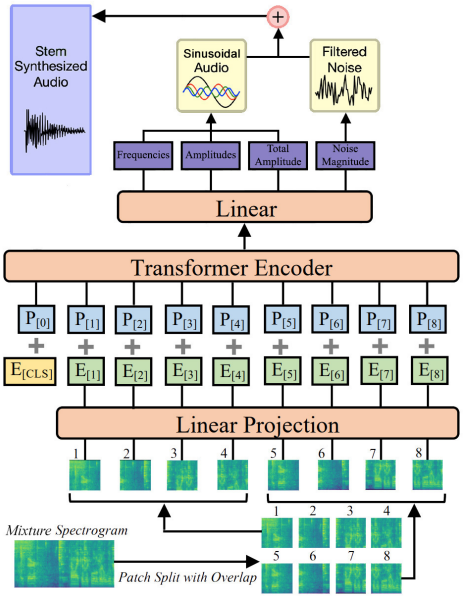

Figure 1. The proposed AST-DDSP model.

The implementation is inspired by the idea of inverse audio synthesis of (Engel et al., 2020b), by combining the Sinusoidal Encoder and the AST model (Figure 1). The Harmonic Encoder has been removed due to the fact that it limits the model to generate only monophonic sources while in MSS we are interested in separating **also inharmonic** (e.g. drums) **or polyphonic** (e.g. piano, guitar) **sources**.

So we have the **Sinusoidal Encoder** which outputs the sinusoidal frequencies f_k, amplitudes A_k and filtered noise magnitudes N_k every 62.5ms, which are upsampled to audio rate. The **Sinusoidal Synthesizer** as described in (Engel et al., 2020b) which reconstructs the audio as a sum of sinusoids:

$$x(n) = \sum_{k=0}^{K-1} A_k(n) \sin(\phi_k(n)) \tag{1}$$

where $\phi_k(n)$ is the phase obtained by cumulatively summing the frequency $f_k(n)$:

$$\phi_k(n) = 2\pi \sum_{m=0}^{n} f_k(m) \tag{2}$$

The **Filtered Noise** is generated uniformly at random from a set of 65 bandpass filters whose amplitude is modulated by the filtered noise magnitudes N_k.

The two signals are then combined to obtain the output of the model, which is the stem track of the instrument in waveform domain.

4. Results

For the experiments 3 different models have been trained, one for each instrument (bass, drums and guitar), for the purpose of demonstrating the effectiveness of the proposed method on monophonic (bass), inharmonic (drums) and polyphonic (guitar) sources.

Each model has been trained on the **Slakh2100 dataset** (Manilow et al., 2019) for 800 epochs (∼250K steps) using ADAM optimizer with a batch size of 128 and learning rate of 3e-4, and exponential learning rate decay 0.98 every 10,000 steps.

The loss used is **Multi-Resolution STFT Loss** (Steinmetz & Reiss, 2020) which is the sum of the STFT losses with different analysis parameters (window size, hop size, etc.) to make the model more robust to variations in frequency and temporal content, improving its ability to accurately separate different audio sources with different spectral and temporal characteristics.

Since the model is computationally expensive to train, the dataset has been downsampled to 8kHz and the network has been fed with 2 seconds of audio. The audio results can be found on the GitHub Page `https://davegabe.github.io/ast-ddsp-mss/`

5. Conclusions

As shown in the results **the model is able to extract the audio** of monophonic and inharmonic instrument sources from the mixture, but the quality is not very good. This can be due to the fact that the model has not been trained for enough epochs because of the limited computational resources available.

Moreover the model struggles to extract the guitar from the mixture probably because the training data consists of both monophonic and polyphonic sources and also has a wider frequency range than the other instruments.

For the future work I plan to extend the training for more epochs, using a higher sample rate (in order to not lose significant spectral content) and to perform a more in-depth hyperparameter search in order to improve the quality of the audio and to make the model more robust to polyphonic sources. Moreover, I plan to extend the model to be able to separate multiple sources at the same time.

References

Défossez, A., Usunier, N., Bottou, L., and Bach, F. Demucs: Deep extractor for music sources with extra unlabeled data remixed, 2019.

Engel, J., Hantrakul, L. H., Gu, C., and Roberts, A. Ddsp: Differentiable digital signal processing. In *International Conference on Learning Representations*, 2020a. URL https://openreview.net/forum?id=B1x1ma4tDr.

Engel, J., Swavely, R., Hantrakul, L. H., Roberts, A., and Hawthorne, C. Self-supervised pitch detection with inverse audio synthesis. In *International Conference on Machine Learning, Self-supervised Audio and Speech Workshop*, 2020b. URL https://openreview.net/forum?id=RlVTYWhsky7.

Gong, Y., Chung, Y.-A., and Glass, J. AST: Audio Spectrogram Transformer. In *Proc. Interspeech 2021*, pp. 571–575, 2021. doi: 10.21437/Interspeech.2021-698.

Manilow, E., Wichern, G., Seetharaman, P., and Le Roux, J. Cutting music source separation some Slakh: A dataset to study the impact of training data quality and quantity. In *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2019.

Manilow, E., Seetharman, P., and Salamon, J. *Open Source Tools & Data for Music Source Separation*. https://source-separation.github.io/tutorial, 2020. URL https://source-separation.github.io/tutorial.

Rouard, S., Massa, F., and Défossez, A. Hybrid transformers for music source separation. In *ICASSP 23*, 2023.

Serra, X. and Smith, J. Spectral modeling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decomposition. *Computer Music Journal*, 14(4):12–24, 1990. ISSN 01489267, 15315169. URL http://www.jstor.org/stable/3680788.

Steinmetz, C. J. and Reiss, J. D. auraloss: Audio focused loss functions in PyTorch. In *Digital Music Research Network One-day Workshop (DMRN+15)*, 2020.