Language Modelling for Concatenative Synthesis

Creative audio generation tool

Computing Final Project Proposal

Wojciech Kacper Werkowicz
Student ref. 33655491

Research problem

The basis for the inquiry is the potential of enhancing concatenative synthesis with deep learning-based language modelling. This would enable the design of a creative tool for audio generation, capable of coherent and context aware sequence prediction based on arbitrary sound sources, where a flexible range of scales could taken into account - from micro to macro, be it grains or whole sections of input. A crucial part of it would be optimization for inference and training performance, especially focusing on widely accessible systems.

Background

Concatenative audio synthesis has a long history of using Machine Learning methods as they prove especially capable in processing features extracted from sound corpora, therefore being heavily used in recognized toolboxes such as CataRT[1] and FluCoMa[2]. More recently, learned latent representations have been used to replace calculated grain descriptors as IRCAM's Neural Granular Sound Synthesis[3] was enabled by variational auto-encoder (VAE) architecture and was soon followed by its state-of-art real-time implementation called RAVE[4]. Around the same time before OpenAI has released JukeBox[5] - a transformer-based network that applies generative language modelling

to sequences of MIDI notes extracted from source. Fitting a model of either architectures requires system specifications ranging from these of significant computational power (single, high-end GPU), to those beyond the reach of an individual (specialized clusters of processing units).

Proposed approach

Input segmentation and grain classification can be implemented in Python using non-supervised clustering algorithms (such as DBSCAN or Affinity Propagation).
Labelled grains in their original order of occurrence can be then used as a 1-D sequence to train a generative transformed-based model designed in Keras or PyTorch, as to predict possible next labels depending on the current state using a suitable sampling method.
Since the original audio corpus is still accessible, predicted labels making up the generated sequence can be easily mapped back to corresponding grains without any extensive cost caused by calculating audio samples from scratch. These are to be re-synthesised into output signal using a method of choice. In case the time resources permit, trained model could be wrapped using TorchScript so that it can be used for inference (prediction) inside of a more user-friendly environment, comparably to the nn~ Max MSP external[6] and RAVE.js[7].

---

1. Schwarz, D. 'Corpus-Based Concatenative Synthesis'. *IEEE Signal Processing Magazine* 24, no. 2 (March 2007): 92–104. https://doi.org/10.1109/MSP.2007.323274.↩
2. Tremblay, Pierre Alexandre, Gerard Roma, and Owen Green. 'Enabling Programmatic Data Mining as Musicking: The Fluid Corpus Manipulation Toolkit'. *Computer Music Journal* 45, no. 2 (1 June 2021): 9–23. https://doi.org/10.1162/comj_a_00600.↩
3. Bitton, Adrien, Philippe Esling, and Tatsuya Harada. 'Neural Granular Sound Synthesis'. arXiv, 3 July 2021. http://arxiv.org/abs/2008.01393.↩
4. Caillon, Antoine, and Philippe Esling. 'RAVE: A Variational Autoencoder for Fast and High-Quality Neural Audio Synthesis'. arXiv, 15 December 2021. http://arxiv.org/abs/2111.05011.↩

5. Dhariwal, Prafulla, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. ‘Jukebox: A Generative Model for Music’. arXiv, 30 April 2020. http://arxiv.org/abs/2005.00341.↩
6. Caillon, Antoine, and Axel Chemla-Romeu-Santos. ‘nn~’. ACIDS Ircam. Accessed 16 January 2023. https://github.com/acids-ircam/nn_tilde.↩
7. Caillon, Antoine. ‘RAVE.js’. Accessed 16 January 2023. https://caillonantoine.github.io/ravejs/.↩