Deep Learning Workshop - Project Report
Music Generation via S4

Gal Bezalel[†]

[†]Tel Aviv University, galbezalel {at} mail.tau.ac.il

1 Overview

This project is aimed at leveraging the recent advances in sequence modeling made by structured state-space models, and explore how well those architectures can generate music, *preferably using a small number of parameters (up to* 10^9*)*. The base model to be investigated and trained is SaShiMi [1].

2 End product (system from user's perspective)

The end product is straightforward: a user will input a prompt in the form of a music waveform. The system will generate a continuation based on this prompt, akin to a companion musician or a band mate. Naturally - considerations concerning usability, functionality and performance will come into play:

- ◇ From a product perspective, user's prompt need not a be strictly pre-known `.wav` file. Rather, the user should be able to dynamically choose the music source (e.g. from YouTube) and the system will take care of required conversions.
- ◇ The backend model might be constrained by number of parameters, usable training data, etc. - and therefore may not generalize well for diverse pieces of music (e.g. different instruments, or genres). In order to compensate for such limitations, the frontend will guide the user towards reasonable prompts.

3 Training and inference schemes

Model checkpoints, specifically for the YouTubeMix dataset, were made available.
In terms of training, we consider the following schemas:

- ◇ **Naive finetuning** - According to the article, training on the YouTubeMix dataset was done for 600K steps; So simply continuing training from the checkpoint could be done - first, for 150K steps in the hope of showing significant improvement in loss value, and continuing if so. Mixtures of the original pretraining dataset and new FT datasets should be used.
- ◇ **Reparameterization** - The article mentions that the only parameter tying was done for Λ, for simplicity and stability. However, parameter tying for Δ in S4 is considered an important differentiator of this architecture from other Seq2Seq models. We will explore ways to fine-tune the model with additional parameters to optimize.

- ◇ **Complete Pretraining** - Training the model from scratch could be feasible (see Compute and storage requirements and options). We will test this on current maximal configuration (8 S4 blocks) and depending on available resources, consider adding expressive power, e.g. more blocks.

Inference in the base model can be done conditionally (given a long enough prompt) or unconditionally (generate from scratch from the distribution captured in the parameters). Both of these options are interesting to investigate, and can be incorporated in the end product. Evaluation of the model post-training will be done similarly to the article - using negative log likelihood (NLL) on stratified test set.

4 Datasets

Suitable music datasets are common:

- ◇ MIDI datasets (e.g. The Lakh MIDI Dataset [3]) can offer flexibility in isolating a specific track (e.g. piano), and some unified structure after sampling into waveform.
- ◇ As mentioned in the original article [1], long audio tracks from YouTube can be downloaded and used. Further inspection of copyrights is required.

5 Compute and storage requirements and options

As noted in the original article [1], the autoregressive versions of SaShiMi were trained on a single V100 GPU machine*. This makes training on Google Colab (Pro subscription) feasible. However, if more advanced training schemas will be applied, a request for using TAU's GPU cluster might be made.

6 Third party tools and models to be used

The base model to be trained is SaShiMi [1], which is an adaptation of the original S4 model [2] into the audio domain. All S4 descendants are open source, and a reproduction of the original article (claiming to have achieved better NLL) is also available. Training will be done on Google Colab (with Pro subscription).

References

[1] K. Goel, A. Gu, C. Donahue, and C. Ré, "It's raw! audio generation with state-space models," 2022. [Online]. Available: https://arxiv.org/abs/2202.09729

[2] A. Gu, K. Goel, and C. Ré, "Efficiently modeling long sequences with structured state spaces," 2022. [Online]. Available: https://arxiv.org/abs/2111.00396

[3] C. Raffel, "Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching," 2016. [Online]. Available: https://academiccommons.columbia.edu/doi/10.7916/D8N58MHV

*It was mentioned that in such setting, training took up to a week.