Problem.

Approach.

Methodology.

Results.

Demo.

References.

5-10 min. o más.

Problem:

Few/None of the current models create violin music only, most are for piano or multiple instruments.

Creating a model that can compose/create violin music, starting from a pretrained model trained for piano.

Approach:

String instruments like violin have different characteristics than key instruments like piano, main characteristics are:

Monophonic--G Key --> Violin

Polyphonic--F Key --> Piano

Piano has a larger range than violin, so there are some notes that can be played on the piano but can't be played on the violin.

Original dataset used for training the pre-trained model is not available, however, the dataset is for piano according to the creator of the model.

Methodology:

After searching for documentation and finding diverse models like PerformanceRNN and Music Transformer, this models had implementation problems and/or the

code was not available for modifications. Then, after searching on Github, a model was found using transformers, and this model allowed to modify the code

and implement new functions.

This Github repository contained several .ipynb files, however, only 2 were used: Train.ipynb and Generate.ipynb.

The first file allows to train a model from scratch on new data, and the second file allows to create piano music using a pretrained model.

This repository also contained implementation problems, which were solved after reviewing the code: some of the libraries had to be added, installed and imported manually.

To be able to fine tune the model, a combination of both files were implemented to create a .ipynb which allows to train a model starting from the

pre trained one.

Another problem was the violin dataset, which is hard to find on the internet. To solve this problem, it was decided to create a dataset manually, by

searching violin midi files. These files were easier to find, and after founding a group of midi files which contained Sonatas and Partitas from J.S. Bach for solo violin, these files were joined within the same folder and used to fine-tune the model. The final dataset consists on 30 midi files with approximately

3 minutes of music each, for a total of 90 minutes of music (1.5 h).

Specifics of the model:

The model selected uses the Transformers architecture and implements Beam Search for the predictions, were the value of the "N" predictions is decided

by the user.

The baseline of the model is exactly like fastai's TransformerXL, but it contains a more aggressive attention mask, with size 1 to look at each specific

part of the input simultaneously.

Transformers XL, according to the original article: "Transformers have a potential of learning longer-term dependency, but are limited by a fixed-length

context in the setting of language modeling. We propose a novel neural architecture Transformer-XL that enables learning dependency beyond a fixed

length without disrupting temporal coherence. It consists of a segment-level recurrence mechanism and a novel positional encoding scheme". (https://arxiv.org/abs/1901.02860)

Results:

To obtain the results, both models were used to make a comparison.

The results of both models can be obtained using a seed file from any music composition as a seed, which the models used as a baseline to create new

music. In this case, the output can be either piano or violin music depending on the seed given to the model.

The most important results are obtained with the predictions from scratch, in which the model is not given any seed, and it has to predict/create music

using only the data it was trained on. For this case, the results of the models were clearly different: the pretrained model created piano music

(music with piano characteristics: polyphonic-multiple notes (chords), F key), while the fine-tuned model created violin music (monophonic-only one

note at a time, G key). These results were verified by running each model 3 times and saving the results, and in most of the cases, the pretrained model created piano music and the fine-tuned model created violin music, which was the final objective of this project: creating violin music. The pretrained model sometimes created violin music, but this is because in piano there are two pentagrams in a music sheet, and one of them is similar to the violin (one is in F key, and the other is in G key), however, this violin music created by the model still had polyphony problems.

Demos:

Important to note that the synthesizer is for piano in both cases, but this is not a problem since violin music can be played by a piano, and the problem occurs only when trying to play piano music on a violin. Music can be converted to violin if a different synthesizer is used, but this is not the objective of this project.

References:

Original code by:

https://github.com/bearpelican/musicautobot/tree/15bc523548f8ae737a594ee92564538d02e0dc94

Dataset from:

http://www.jsbach.net/midi/midi\_solo\_violin.html

https://bitmidi.com/eine-kleine-nachtmusik1-violin-mid

Transformers XL Paper:

https://arxiv.org/abs/1901.02860