Music Transformer

Paper Reference: Music Transformer: Generating music with long-term structure

Why Transformer is a suitable model for extended length music generation?

- **RNN**:Learn to proactively store elemnts to be referenced in a fixed size state or memory

- **Self-Attention Transformer**: autoregressive model to access any part of the previously generated output at every step of generation
- Idiomatic piano gestures such as scales, aipeggios, and other motifs all exhibit a certain grammar and recur periodically. Thus capturing relative positional distances makes it easier to model this regularity

What is the the chanllenge of transferring NLP Transformer model to music generation

- To capture timing and pitch collaboratively, the pairwise distance distance between dimensions need to be cached
- The naive $O(L^2D)$ memory requirement is not practical for 1028 dimension music representation
- The Paper effectively reduced the requirement to $O(LD)$ to enable efficient learning and seq2seq-based generation

Data Representation

- Represent polyphonic musical into a sequence of discrete tokens

Transformer Model Formulation

> L : length of the sequence (for example, the number of characters/words in a sentence)
>
> D : The dimension to represent features at each step of the sequence
>
> X : $D \times L$ matrix that represent the data
>
> W^Q, W^K, and W^V are $D \times D$ square matrices for the weight to generate $L \times D$ query, key and value matrix

$$Q = XW^Q$$
$$K = XW^K$$
$$V = XW^V$$

Q, K, and V are then split into H $L \times D_h$ attention heads, indexed by h, and with dimension $D_h = \frac{D}{H}$, which allows model to focus on different parts/dimensions of the data.

A scaled dot-productattemtion computes a sequence of vector outputs for each attention head as

$$Z^h = Attention(Q^h, K^h, V^h) = Softmax(\frac{Q^h(K^h)^T}{\sqrt{D_h}})V^h$$

Z^h are then concatenated to Z a $L \times D$ matrix. A causal mask is applied during the query step. Namely an upper triangular mask ensure queries not to get to keys later in the sequence. The feedforward sub-layer then takes the output Z from attention sub-layer, and learned W_1, W_2, b_1 and b_2 with

$$FF(Z) = ReLU(ZW_1 + b_1)W_2 + b_2$$

Variation: Relative positional self-attention

The original transformer architecture relies solely on positional sinusoids to represent timing information. Shaw et al.(2018) introduces relative position representations to inform how far 2 positions are apart in a sequence

With reduced memory requirement by "skewing process", training on long sequences is pragmatic.

Training

A Sequence to Sequence setup was applied. The melody was provided and the model is asked to predict the full harmonic elements.

Funtime: Playing around with the model

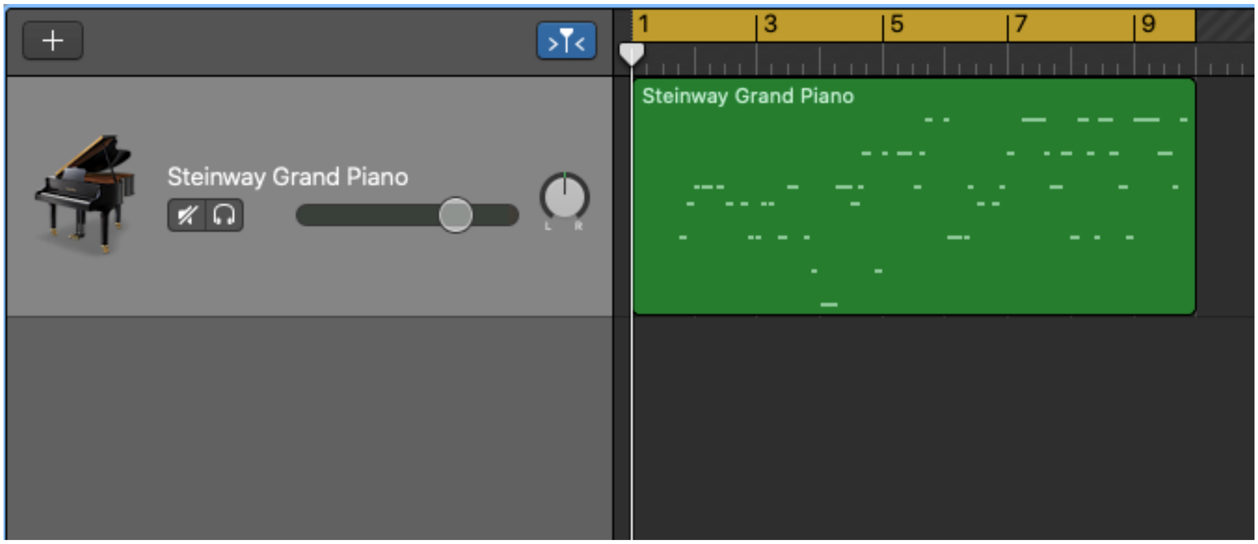
> I found colab notebooks to generate music on magenta project repo : https://magenta.github.io/listen-to-transformer/#a1_66734.mid
>
> Below I will attempt to evalute the model performance based on some trail samples

Melody Generation:

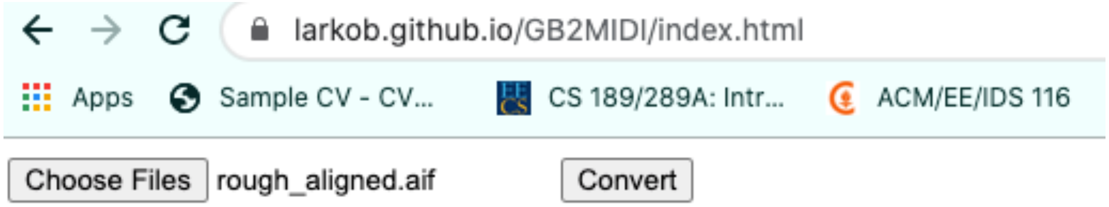
A motif of music is required to generate the both new music and piano accompaniment for music-transfomer model. I therefore generate single-line melody in GarageBand and converted the loop.aif file into .midi file use the script by https://larkob.github.io/GB2MIDI/index.html.

1. **Generate music pitch and timing in Garageband**

   > I typed in the melody lines for GFriend - Rough, GFriend - Me Gustas Tu, and Eason Chen

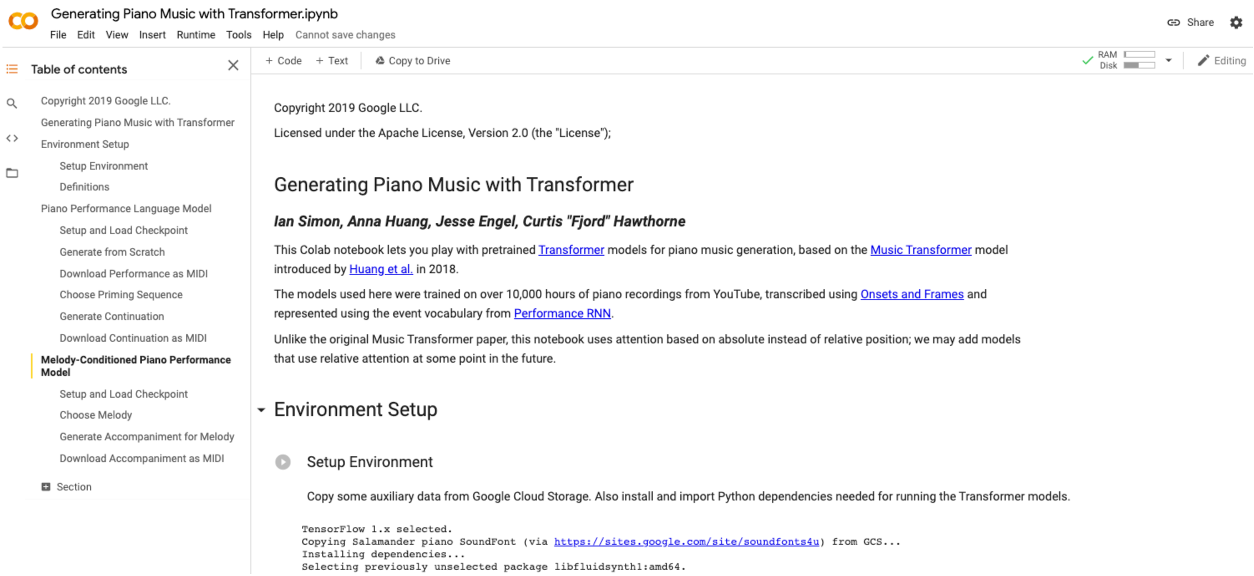


2. **Convert stored .aif GarageBand loop file into midi file using the perl script from https://larkob.github.io/GB2MIDI/index.html.**


larkob.github.io/GB2MIDI/index.html

Choose Files rough_aligned.aif

3. **Use the generate midi to run through the collab notebook provided by the google magenta team**


Generating Piano Music with Transformer.ipynb
File Edit View Insert Runtime Tools Help Cannot save changes
Share
+ Code + Text
Copy to Drive
RAM
Disk
Editing
Table of contents
Copyright 2019 Google LLC.
Generating Piano Music with Transformer
Environment Setup
Setup Environment
Definitions
Piano Performance Language Model
Setup and Load Checkpoint
Generate from Scratch
Download Performance as MIDI
Choose Priming Sequence
Generate Continuation
Download Continuation as MIDI
Melody-Conditioned Piano Performance Model
Setup and Load Checkpoint
Choose Melody
Generate Accompaniment for Melody
Download Accompaniment as MIDI
Section

Copyright 2019 Google LLC.

Licensed under the Apache License, Version 2.0 (the "License");

Generating Piano Music with Transformer

Ian Simon, Anna Huang, Jesse Engel, Curtis "Fjord" Hawthorne

This Colab notebook lets you play with pretrained Transformer models for piano music generation, based on the Music Transformer model introduced by Huang et al. in 2018.

The models used here were trained on over 10,000 hours of piano recordings from YouTube, transcribed using Onsets and Frames and represented using the event vocabulary from Performance RNN.

Unlike the original Music Transformer paper, this notebook uses attention based on absolute instead of relative position; we may add models that use relative attention at some point in the future.

Environment Setup

Setup Environment

Copy some auxiliary data from Google Cloud Storage. Also install and import Python dependencies needed for running the Transformer models.

```
TensorFlow 1.x selected.
Copying Salamander piano SoundFont (via https://sites.google.com/site/soundfonts4u) from GCS...
Installing dependencies...
Selecting previously unselected package libfluidsynth1:amd64.
```

Result

> Below I generated the accompaniment and extended melody for three times, and select the best version of my musical taste
>
> Let's have a look at how good transformer model performed each task:

Original Midi Audio Melody Line

- GFRIEND - Rough
    -
- GFRIEND - Me Gustas Tu
    -
- Eason Chen
    -

The Midi Audio for continuition

- GFRIEND - Rough
    -
- GFRIEND - Me Gustas Tu
    -

- Eason Chen
  -

The Midi Audio for accompaniment

- GFRIEND - Rough
  -
- GFRIEND - Me Gustas Tu
  -
- Eason Chen
  -

Brief Analysis of the produced result

Piano Continuition:

> We selected three pop songs here, and there are lots of repetitive patterns here. The music transformer model was primarily trained on piano pieces, and the melody line may be more diverse and less repetitive. As a result, the generated continuition also contains a lot of repetitive pattern and not so pleasantly formulated. To me, the music transformer model does not acheive acceptable level of production.

Piano Accompaniment:

> For music accompaniment, the model could successfully identify the tempo, measure and produced similar chord progression as the original song. The accompaniment is surprisingly good and is beyond my expectation :)