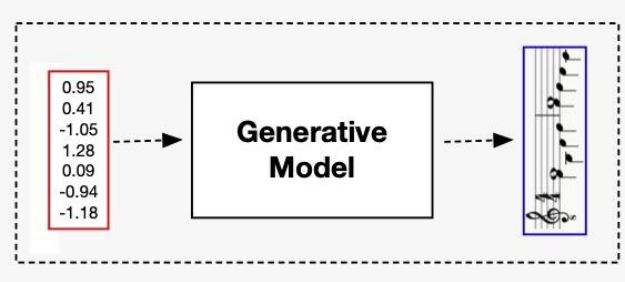
Prescribe Your Music

I. Koz H. Uzar A. Balci

Abstract

The music composed by machines can be huge landmark for the musicians as a helper tool to give them inspiration as well as be used in movies. There are many researches about music generation and with the development of deep learning algorithms, the programs that are created for music composition became really powerful and useful. Even if there are some music composers that leveraging machine learning algorithms now, these composers can be improved. So, we decided to examine the traditional way of achieving it besides the State of Art technique called WaveNet architecture. We explained why we choose it for our application with the justifications and the statistical data in which it and its competitors are dissected. Finally, we talked about how to make enhancements in performance.



1. Introduction

The idea of making machines to compose music has been in existence from a long time. We hope that the music and film industry can leverage that state of art.

Music generation is the process of composing a short piece of music with minimum human intervention. The first music generated by a computer appeared in 1957 and it was generated by the software for sound synthesis named Music I, developed by Mathews at Bell Laboratories.

As the time passed, supervised machine learning models started to be used. These models can be divided into two categories: discriminative models and generative models. Discriminative models identify a decision boundary and produce a corresponding classification. Generative models create new instances of a class. The generative models are used to compose music.

The newest of these models is WaveNet which is a deep learning-based generative model for raw audio developed by Google DeepMind (Google, 2021b). The main objective of WaveNet is to generate new samples from the original distribution of the data. Hence, it is known as a Generative Model. In addition, WaveNet is a type of feedforward neural network known as a deep convolutional neural network. In WaveNet, the CNN takes a raw signal as an input and synthesises an output one sample at a time.

We know that the WaveNet outperforms the older models significantly according to the experiments, so we decided to work on this model to generate our music by giving some prescriptions. We hope that the present work will contribute to making the generation of music accessible to a wider audience, from non-musicians to professional musicians.

2. Related Work

Sequence learning is attracting more and more attention both in industry and academic world with the wide usage of RNN and LSTM neural network architecture. Many research groups and companies started make researches about the sequence learning from the musical side. Google Brain team open sourced a research project named Magenta (Google, 2021a), which tries to provide a platform for musicians, artists and programmers to create their music and art works using machine intelligence. Several months later, DeepMind published their WaveNet paper (Google, 2021b) which proposes a deep generative model of raw audio waveforms and achieves astonishing state of the art performance gain. Also a research group at Spotify named CRTL(Creator Technology Research Lab) worked on music generation and led by professor and musician François Pachet (Spotify, 2017). Along with the big companies, lots

of research groups are wprking on music generation and publish papers (res).

The application of machine learning to music data aims to create machines that are beneficial to working with music. The uses of such systems span from the analytic, e.g., description and recommendation via music information retrieval, to the synthetic, e.g., creative transformation and generation via algorithmic composition. The latter continues to be a very active research area, especially with deep learning methodologies, and has growing commercial interest.

In fact the music generation process has started to be commercial. In 2014, Amper Music (AmperMusic) is founded with the mission to enable anyone to express themselves creatively through music regardless of their background or expertise and became the world's first end-to-end AI music composition platform for enterprise content creators . Using the Amper Score™ platform, content teams can create and edit music to accompany videos, podcasts, and many other types of content(Newswire, 2019). Similar to Amper Score, AIVA (AIVA) is another music composer. It was created in 2016 and specializes in classical and symphonic music composition. It became the world's first virtual composer to be recognized by a music society (SACEM). The algorithm AIVA is based on deep learning and reinforcement learning architectures. Sony also has a composer called Flow Machines which aims to help the artists (Sony).

3. The Approach

We have a problem subsumed into the seq2seq problems. The seq2seq problems are usually handled by either RNN's or its derivatives. Lately a paper has proposed a technique called WaveNet. This approach adopts a technique involving Convolution. Since the audio files are sequential data and there exists high dependency between the audio samples, we can apply convolutional layers confidently as with the image data. WaveNet employs a 1D causal dilated convolutional neural network. Audios are sequential data, so we must keep the order in which they are processed. Causal refers to that feature. But there lacks some important feature in naïve CNN that we cannot dump all the data to CNN at once for it to tackle the entire dependency between audio samples.

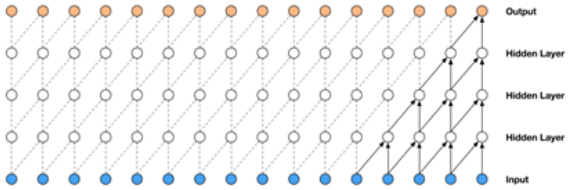


Figure 1. A naive convolutional layer

The reason why we use dilation in our convolutional layer is arising from the intention of that we fancy to use all the data in hand in each evaluation. By comparison with RNN which is a common technique to handle seq data, it does not suffer from having short term memory by virtue of the dilation. Dilation paves the way for taking all data as input.

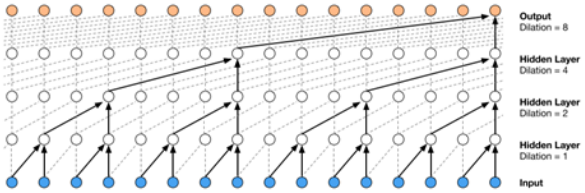


Figure 2. A dilated convolutional layer

Dilation factor for our experiment is doubling at each timestep. The motivation behind picking it as 2 is two-fold. Since it is a convolutional layer, we have a kernel sliding through the audio samples in order. We start our composition with a random audio samples picked up from the training set. Then, we calculate the probability distribution for each audio sample in the training set to decide what audio sample will come after the group we picked up. The sample whose value is the maximum will be chosen to add to the queue to be used with later our random sample training data. But to keep the sample in the same as the initial state, we need to chop off it because a new predicted audio sample is added to the sample data. This stripping operation is performed from the beginning of the data. This process is repeated until we are done. This technique which is the state of art outperforms its competitors as seen below:

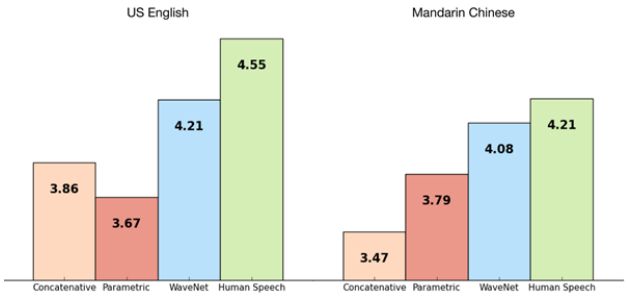


Figure 3. Difference between US English and Mandarin Chinese

As mentioned above, we use a probabilistic model and it has such a form given below:

$$p(\mathbf{x}) = \prod_{t=1}^{T} p(x_t \mid x_1, \ldots, x_{t-1})$$

Figure 4. Conditional Distribution Formula

Other thing that is worth of mention is because we are using a conditional probabilistic model we applied to use Softmax Layer as an activation layer.

Because raw audio is typically stored as a sequence of 16-bit integer values (one per timestep), a softmax layer would need to output 65,536 probabilities per timestep to model all possible values. To make this more tractable, we first apply a μ-law companding transformation (ITU-T, 1988) to the data, and then quantize it to 256 possible values:

$$f(x_t) = \operatorname{sign}(x_t) \frac{\ln(1 + \mu |x_t|)}{\ln(1 + \mu)},$$

Figure 5. Softmax Activation Function

We applied the following gated activation unit for our model.

$$\mathbf{z} = \tanh(W_{f,k} * \mathbf{x}) \odot \sigma(W_{g,k} * \mathbf{x}),$$

Figure 6. Tanh Activation Function

We are convinced that non-linearity worked significantly better than the rectified linear activation function by the former research. For the sake of performance enhancements, we applied an architecture as shown below:

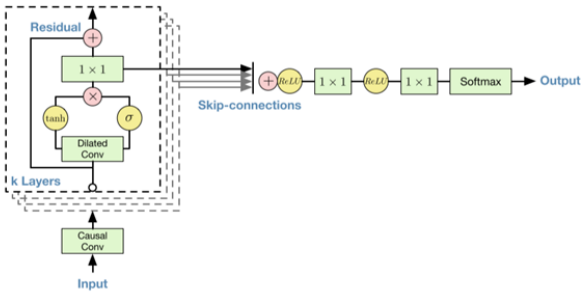


Figure 7. Architecture of the system

4. Experimental Evaluation

We saw that there are two ways of using data as training data in musical affairs. These are using raw audio files and MIDI files. MIDI files have an edge over raw audio files because they are composed only of the instructions rather than the actual audio. Hence, they are smaller. This is what makes them eligible for transferring files. We've thought that we can use these MIDI files as our dataset, and we can leverage the sites where these MIDI files are available to be downloaded (MIDIWorld). We applied a library called Music 21 that is developed by MIT to convert MIDI files to a form that Python can decipher. We've used a dataset that is of 1000 samples, consisting of varying genres. We split 800 samples as train and 200 samples as validation dataset.

We have a hyperparameter called no_of_timesteps that is used to decide how broad the subsequences will be. According to this parameter choice, we'll have a batch of sequences as training input when it comes to building the model that will be used to predict to successor note in the sequence.

This is the last state of the X and Y, respectively:

[['F4' 'F3' 'A4' ... 'B3' 'B2' 'C4']
['F3' 'A4' 'A3' ... 'B2' 'C4' 'C#4']
['A4' 'A3' 'G4' ... 'C4' 'C#4' 'C3']
...
['E3' 'C4' 'A4' ... 'C5' 'E5' 'A5']
['C4' 'A4' 'B4' ... 'E5' 'A5' 'C6']
['A4' 'B4' 'C5' ... 'A5' 'C6' 'B4']]

Each row is of the length of no_of_timesteps.

['C#4' 'C3' 'C4' ... 'C6' 'B4' 'E5']

y vector is of the length of the number of rows in the X. Of course, in the implementation, these notes are replaced with the label encoded representations of them, but for the sake of clarity, we've kept them in such a form.

We used Google Colab to run our code. We evaluated our model using Log Loss Function and our enjoyment, it thrives by the time as seen below:

```
Epoch 00005: val_loss improved from 3.04337 to 2.99770, saving model to best_model.h5
Epoch 6/50
19/19 [==============================] - 1s 48ms/step - loss: 2.9037 - val_loss: 2.9428

Epoch 00006: val_loss improved from 2.99770 to 2.94280, saving model to best_model.h5
Epoch 7/50
19/19 [==============================] - 1s 39ms/step - loss: 2.8750 - val_loss: 2.9364

Epoch 00007: val_loss improved from 2.94280 to 2.93640, saving model to best_model.h5
Epoch 8/50
19/19 [==============================] - 1s 40ms/step - loss: 2.8386 - val_loss: 2.9063

Epoch 00008: val_loss improved from 2.93640 to 2.90632, saving model to best_model.h5
Epoch 9/50
19/19 [==============================] - 1s 38ms/step - loss: 2.7833 - val_loss: 2.8675

Epoch 00009: val_loss improved from 2.90632 to 2.86755, saving model to best_model.h5
Epoch 10/50
19/19 [==============================] - 1s 38ms/step - loss: 2.7017 - val_loss: 2.8347

Epoch 00010: val_loss improved from 2.86755 to 2.83466, saving model to best_model.h5
Epoch 11/50
19/19 [==============================] - 1s 44ms/step - loss: 2.6591 - val_loss: 2.8105

Epoch 00011: val_loss improved from 2.83466 to 2.81052, saving model to best_model.h5
Epoch 12/50
19/19 [==============================] - 1s 40ms/step - loss: 2.6118 - val_loss: 2.7932

Epoch 00012: val_loss improved from 2.81052 to 2.79324, saving model to best_model.h5
Epoch 13/50
19/19 [==============================] - 1s 41ms/step - loss: 2.5667 - val_loss: 2.7656

Epoch 00013: val_loss improved from 2.79324 to 2.76561, saving model to best_model.h5
Epoch 14/50
19/19 [==============================] - 1s 40ms/step - loss: 2.5063 - val_loss: 2.6990
```

References

Automatic music generation. URL https://www.academia.edu/Documents/in/Automatic_Music_Generation.

AIVA. Aiva. URL ,https://www.aiva.ai/.

AmperMusic. Amper music. URL ,https://www.ampermusic.com/.

Google. Google magenta, 2021a. URL https://magenta.tensorflow.org/.

Google. Deepmind WaveNet, 2021b. URL https://deepmind.com/blog/article/wavenet-generative-model-raw-audio.

MIDIWorld. Midi files. URL ,https://www.midiworld.com/.

Newswire, G. Amper music, 2019. URL ,https://www.globenewswire.com/news-release/2019/01/23/1704214/0/en/Amper-Music-Launches-First-AI-Music-Composition-P
html#:~:text=Amper%20was%20founded%20in%202014,create%20and%20customize%20original%20music.

Sony. Flow machines. URL https://www.flow-machines.com/.

Spotify. Spotify creator technology research lab, 2017. URL https://artists.spotify.com/blog/innovating-for-writers-and-artists.

van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A. W., and Kavukcuoglu, K. Wavenet: A generative model for raw audio. *CoRR*, abs/1609.03499, 2016. URL http://arxiv.org/abs/1609.03499.