DSP Final Project

Music Synthesizer and Generation

with Autoencoder

Yu-Ting Wang (Principles), Yun-Da Tsai (Implementation)
ID: B04902040, B04902103
National Taiwan University
January 14, 2018

Abstract

In this report, we provide a brief overview of music synthesizer and offer some introduction about manifold autoencoder from the mostly used LSTM-based recurrent neural network model to the lately developed WaveNet autoencoder. The principle of Neural Audio Synthesis and NSynth(Neural Synthesizer) is discussed to support the recognition of WaveNet model. Also, our implementation of Polyphonic RNN, an LSTM-based model designed to model polyphonic music, including the result and the construction is shown.

The structure of this report is as follows: Section 1 provides an overview of music synthesizer; Section 2 introduces the autoencoder models of monotonic music generation described in [2], [3]; Section 3 gives the concept of neural audio synthesis, and the referenced papers are [1], [2]; Section 4 shows a brief description, the structure and result of our implementation of Polyphonic RNN using LSTM-based model; Section 5 concludes the report.

1. Overview

1.1 What is Music Synthesizer?

Here we refer Music Synthesizer to Algorithmic composition, which is the technique of using algorithms to create music. Algorithmic composition is the partial or total automation of the process of music composition by using computers.

1.2 Goal

The goal of automated music generation is to propose the model that could reproduce complex temporal and melodic patterns that would correspond to the style of the training input. Here we describe two areas in the field algorithmic composition.

Monotonic Music Generation:

Using a number of model set-ups starting from a prediction of a next note based on one or several previous notes, predicting a phrase or a chord based of a longer time-window or sampling the note or a melody from a previously learned distribution.

Audio Synthesis:

Like a synthesizer(an electronic music instrument) that generates electronic signals that are converted to sound through instrument amplifiers. Here we talk about audio synthesizer as a tool that synthesizes signals into one using computing algorithms.

These two topics are further discussed respectively in section 2 and section 3.

2. Monotonic Music Generation with Autoencoders

2.1 Language Model(LM)

Long short-term memory (LSTM) neural networks, being a particular type of recurrent neural networks(RNNs) is one of the model with outstanding results in comparison to other RNN models. A crucial feature of LSTM network that makes it extremely attractive in this context is that LSTM shows significantly better results when dealing with time lags of unknown size between important events. This comparable insensitivity to gap length gives a unique advantage to LSTMs over hidden Markov models, alternative recurrent neural networks and other sequence learning methods when algorithm works with music.

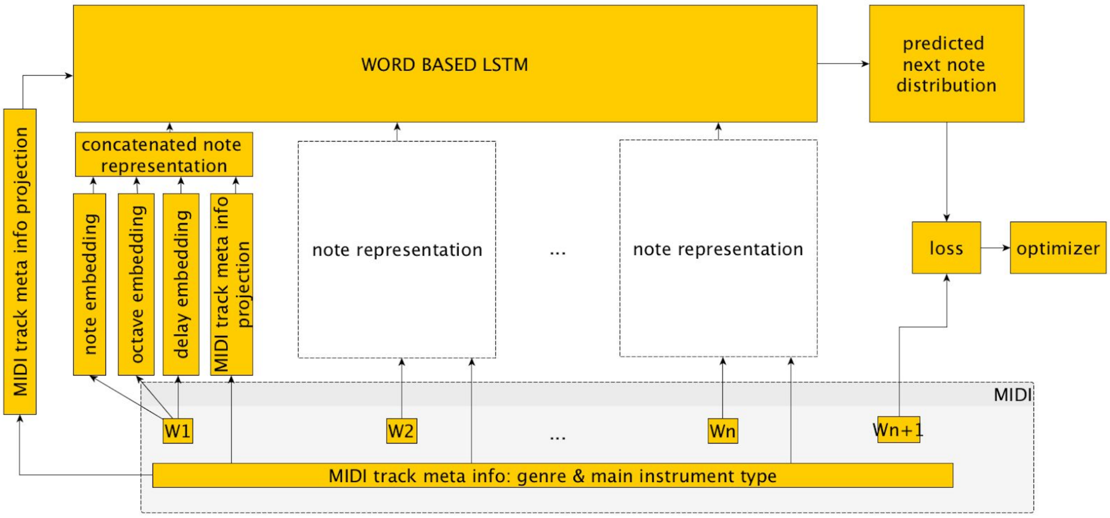

WORD BASED LSTM
predicted next note distribution
MIDI track meta info projection
concatenated note representation
note embedding
octave embedding
delay embedding
MIDI track meta info projection
note representation
...
note representation
loss
optimizer
MIDI
W1
W2
...
Wn
Wn+1
MIDI track meta info: genre & main instrument type

Fig.1. LSTM language model in the context of music generation

A weakness of LM is that it does not capture global features in an interpretable way . There are a number of approaches to the music generation that would deal with this problem paying more attention to the macro structure of the track, for example, VAE.

2.2 Variational Autoencoder(VAE)

VAE is a variational approach for latent representation learning based on several assumptions on the distribution of latent variables. VAE-based generative models can generate realistic examples as if they are drawn from the input data distribution. So there is the architecture shown in Figure 2, a version of VAE called variational recurrent autoencoder. Here the first network (encoder) compresses the given track into a latent vector that works as a bottleneck. The second network (decoder) learns to reconstruct the melody out of a latent representation. This approach stimulates the network to work with a macrostructure of the track due to the low dimensionality of the latent vector.

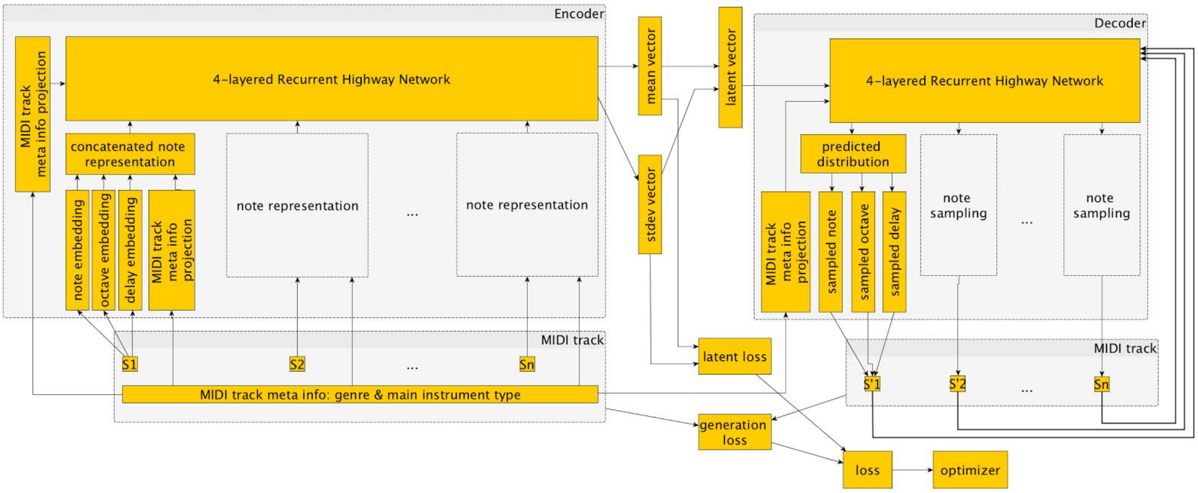


Fig.2. Variational autoencoder scheme for music generation

Naturally, there is a trade-off between the potential of the network to capture the macro-structure and its possibility to generate locally diverse melodies.

2.3 Variational Recurrent Autoencoder Supported by History(VRASH)

VRASH combines a language model and variational recurrent autoencoder in order to increase the performance on the data with varying input length. VRASH architecture is principally described in Figure 3. Here analogously to the scheme on Figure 2 the decoder tries to reconstruct the track out of the latent vector, but this vector is distorted with a variational bayesian noise. The decoder also uses the previous outputs as additional inputs. It ”listens” to the notes that it has composed already and uses them as additional ”historic” inputs.

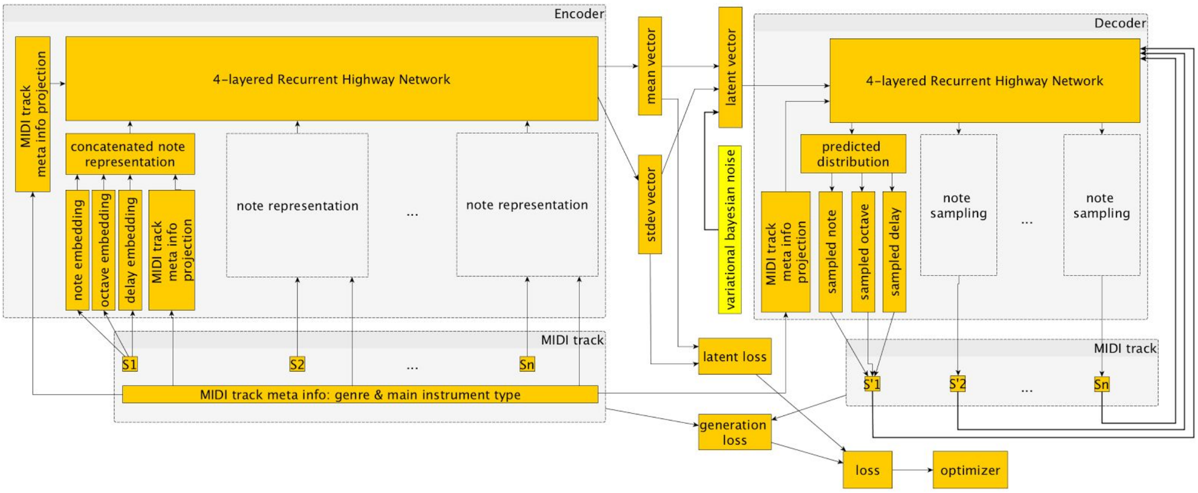


Fig.4. Cross-entropy of the proposed architectures near the saturation point

2.4 Comparison

It is still not clear how one could compare the results of generative algorithms that work in the area of arts. Since music, literature, cinema etc. are intrinsically subjective, it is rather hard to compare them according to a certain rigorous metric.

Here we compare the proposed architectures with respect to the cross-entropy that is commonly used as a loss-term in such tasks and take a look at the output produced by different architectures shared in [3]. In Figure 3 one can see cross-entropy of the proposed architectures near the saturation point. The first untrained random network is used as a reference baseline.

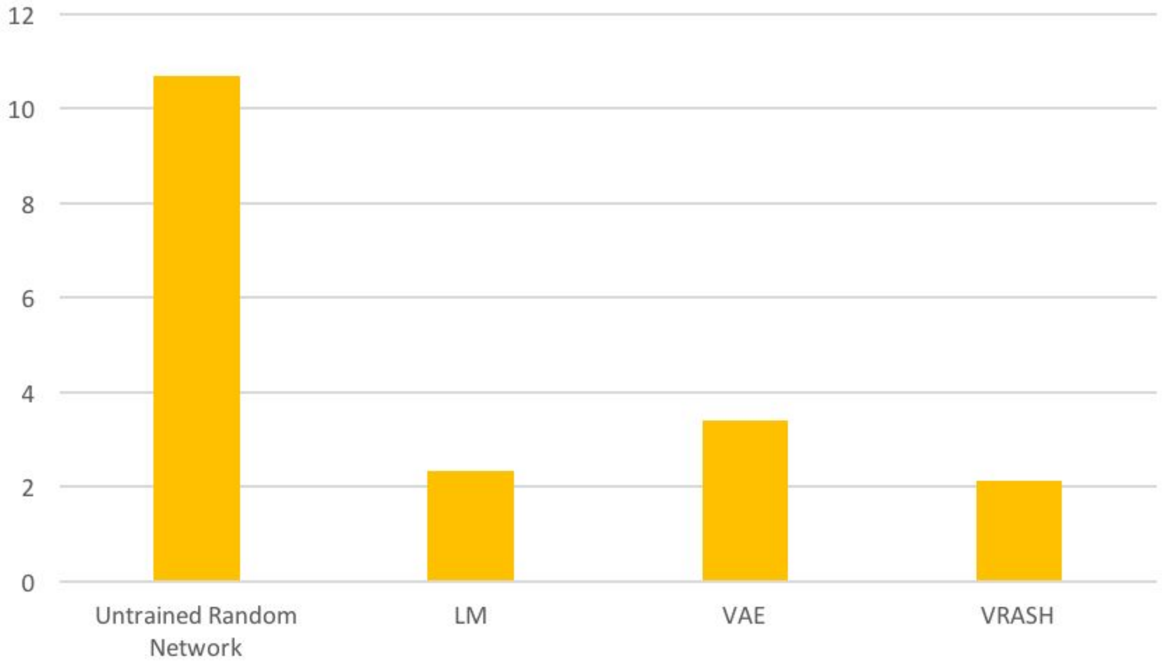


Fig.3. Cross-entropy of the proposed architectures near the saturation point.

2.5 Discussion

All three proposed architectures work relatively well and generate music that is diverse and interesting enough if the dataset for training is big and has high quality, however, they have certain important differences. The first general problem that occurs in many generative models is the tendency to repeat a certain note. This difficulty is more prominent for Language Model whereas VAE and, specifically, VRASH tend to deal with this challenge better.

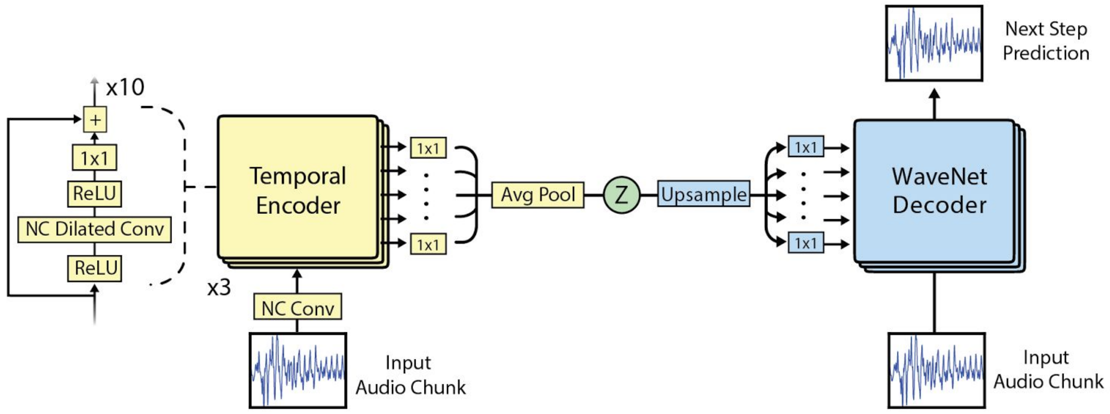
First of all, it provides a good balance between global and local structures of a track. VAE allows to focus on the macrostructure but advancing it in the way described above enables a network to generate more locally diverse and interesting patterns. Second, the proposed structure is relatively easy to implement and train. The last, but not the least, it allows to control the style of the output (through the latent representation of the input vector) and generate tracks corresponding to the given parameters.

Here we conclude our introduction of Monotonic Music Generation. Next, in Section 3 we provide the idea of Neural Audio Synthesis and its application.

3. Neural Audio Synthesis with WaveNet Autoencoders

3.1 Nsynth

NSynth is a novel approach to music synthesis designed to aid the creative process. Unlike a traditional synthesizer which generates audio from hand-designed components like oscillators and wavetables, NSynth uses deep neural networks to generate sounds at the level of individual samples. Learning directly from data, NSynth provides artists with intuitive control over timbre and dynamics and the ability to explore new sounds that would be difficult or impossible to produce with a hand-tuned synthesizer.



The temporal encoder looks very much like a WaveNet and has the same dilation block structure. However, its convolutions are not causal, so it sees the entire context of the input chunk. After thirty layers of computation, there is then a final average pooling to create a temporal embedding of 16 dimensions for every 512 samples. Consequently, the embedding can be thought of as a 32x compression of the original data.

3.2 WaveNet

WaveNet is a powerful deep generative approach to probabilistic modeling of raw audio waveforms. In this section we describe a novel WaveNet autoencoder structure.

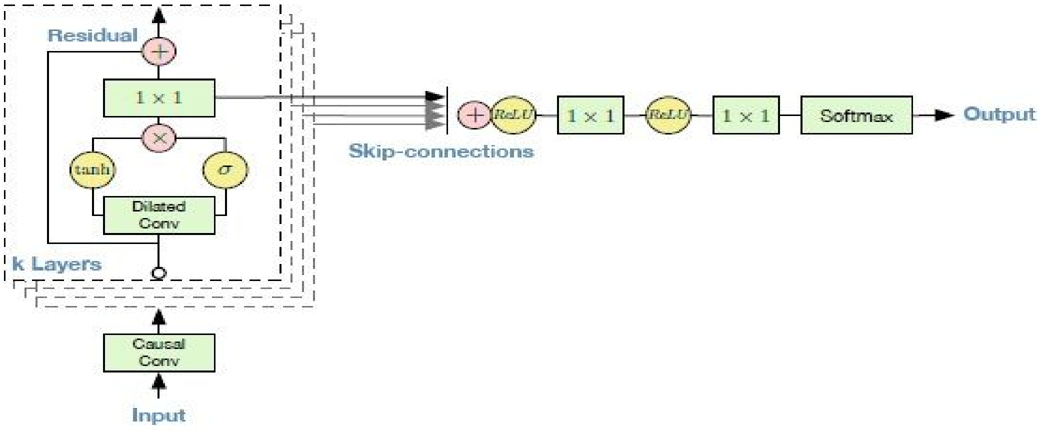


Figure 4: Overview of the residual block and the entire architecture.

3.3 Exploring a Latent Representation of Sound

Mixing the audio of two instruments creates a sound of the two instruments being played simultaneously. In contrast, mixing the latent-representations and then decoding results in what sounds more like a single hybrid or even “multigrid” instrument. Interpolation combines semantic aspects of the two original sounds to create a unique sound that is still musical. Finally, it’s worth mentioning that vector analogies are a powerful creative tool to manipulate specific aspects of media and explore spaces outside the training data. While the sounds were diverse, the temporal dimension of the embeddings generated from vector analogies led to less interpretable outcomes.

4. Implemetation (Polyphonic RNN)

4.1 Introduction

The implemented model is a relatively simple model in comparison to those models mentioned above which applies language modeling to polyphonic music generation using an LSTM. This model is an imitation and reconstruction to one of the model from the Google Brain project, Magenta. We also take advantages from the magenta python libraries and make it an ease to deal with audio data processing.

4.2 Datasets

Our model is trained on the **Bach Chorales Data Set** from the UCI machine learning repository.

Source:

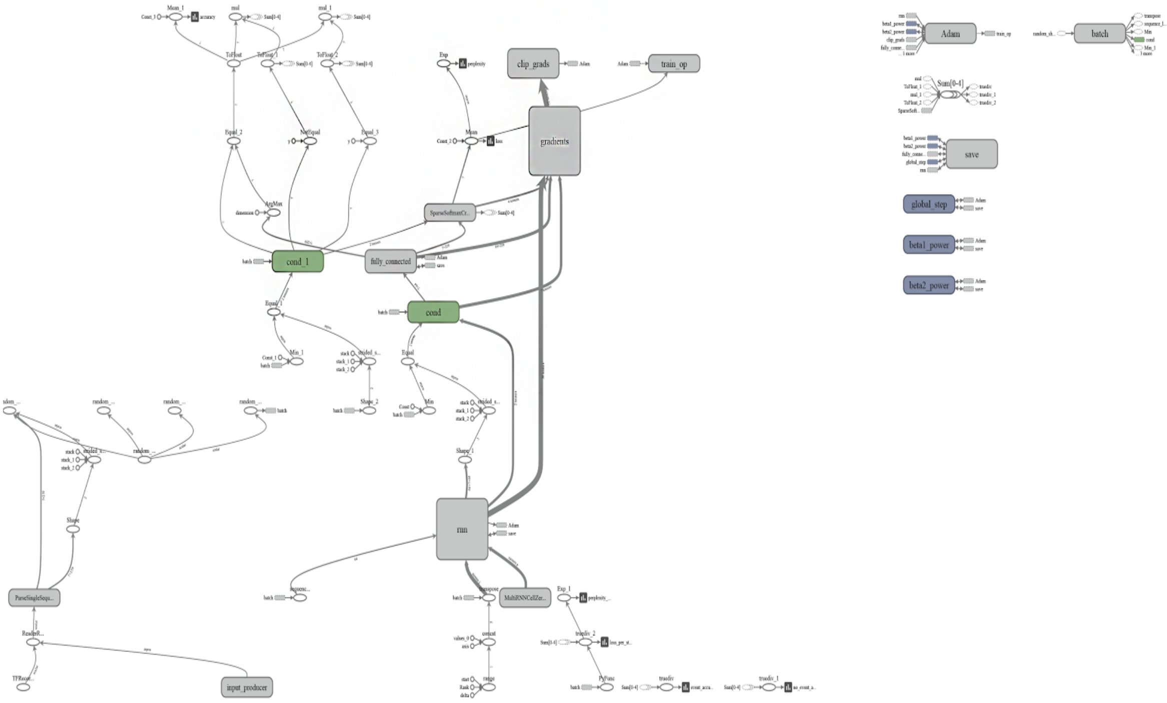
Chorales: Mainous and Ottman edition.Mainous, Frank D. and Robert W. Ottman, eds. 1966.
The 371 Bach Chorales. Holt, Rinehart and Winston, New York.

Attribute Information:

Number of Attributes: 6 (nominal) per event
(a) start-time, measured in 16th notes from chorale beginning (time 0)
(b) pitch, MIDI number (60 = C4, 61 = C#4, 72 = C5, etc.)
(c) duration, measured in 16th notes
(d) key signature, number of sharps or flats, positive if key signature has sharps, negative if key signature has flats
(e) time signature, in 16th notes per bar
(f) fermata, true or false depending on whether event is under a fermata

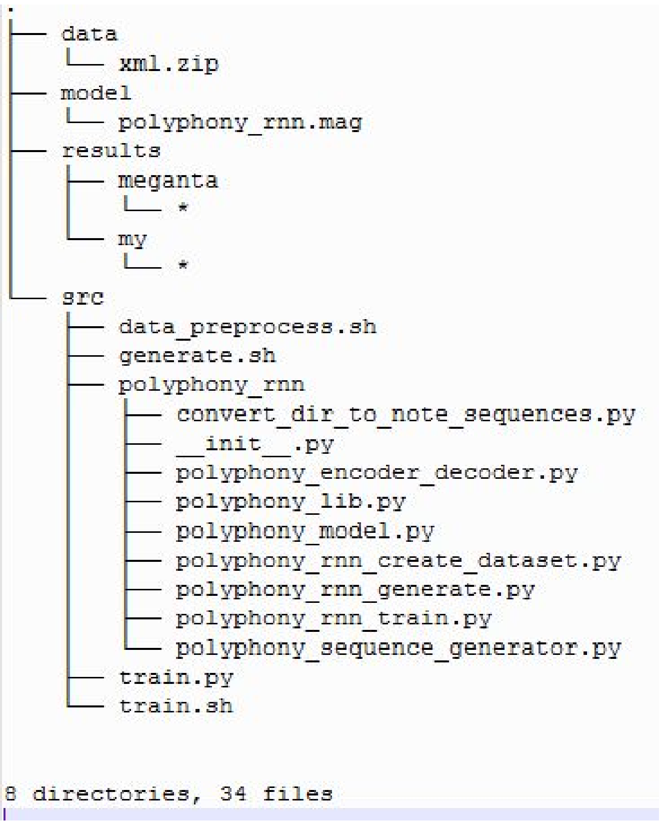
Attribute domains (all integers):
(a) {0,1,2,...}
(b) {60,...,75}
(c) {1,...,16}
(d) {-4,...,+4}
(e) {12,16}
(f) {0,1}

4.3 Model Structure



This is the model structure graph from the tensorboard.

4.4 Execute



```
.
├── data
│   └── xml.zip
├── model
│   └── polyphony_rnn.mag
├── results
│   ├── meganta
│   │   └── *
│   └── my
│       └── *
└── src
    ├── data_preprocess.sh
    ├── generate.sh
    ├── polyphony_rnn
    │   ├── convert_dir_to_note_sequences.py
    │   ├── __init__.py
    │   ├── polyphony_encoder_decoder.py
    │   ├── polyphony_lib.py
    │   ├── polyphony_model.py
    │   ├── polyphony_rnn_create_dataset.py
    │   ├── polyphony_rnn_generate.py
    │   ├── polyphony_rnn_train.py
    │   └── polyphony_sequence_generator.py
    ├── train.py
    └── train.sh

8 directories, 34 files
```

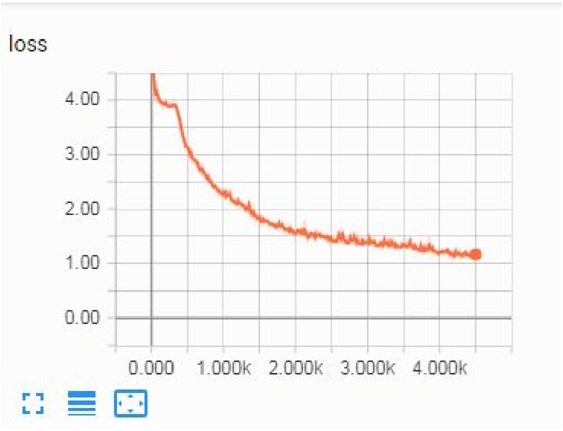
To run the code, the Magenta environment has to be set up. There are too many files that we cannot cover it all since the magenta libraries also depends on many other python packages. We have come up with 3 scripts under ***/src/*** as follow:

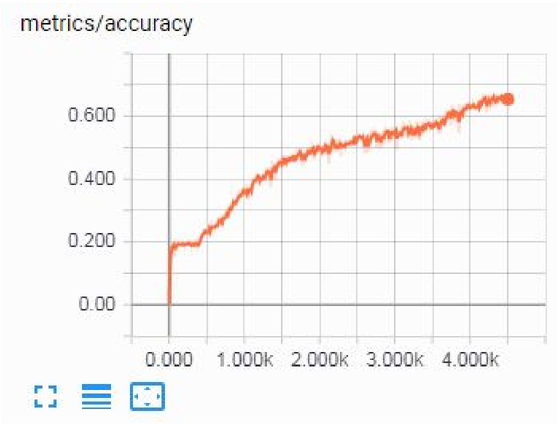
1. generate.sh : generate midi files with model under ***/model/***
2. data_preprocess : generates tfrecords then create sequence examples
3. train.sh : create and train a new model

To be noticed, the scripts has to be executed under ***/src/*** in order to function properly.

4.5 Results

10 midi files generated by out model is under ***/result/my*** and we provide some files generated by the original magenta model under ***/results/magenta***. The training preprocess is more time consuming than we thought and we only achieve an accuracy of about 0.67.





5. Conclusion

In this report, we introduce a new powerful method for monotonic music generation, Variational Recurrent Autoencoder Supported by History; and one for audio synthesis, WaveNet Autoencoder. Both of them show interesting and promising outstanding performance in their field respectively. We expect to see these methods' further development and application to advance music generation and synthesis.

As to our implementation, due to time constraints, our model hasn't been completely trained yet, since the accuracy is still increasing steadily. The results might not seem to be anything special however this is our work for understanding and performing music generation.

References

[1] Sarroff, Andy M and Casey, Michael A. Musical audio synthesis using autoencoding neural nets. In ICMC, 2014.

[2] Jesse Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Douglas Eck, Karen Simonyan, Mohammad Norouzi. Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders. CoRR, abs/1704.01279, 2017.

[3]Alexey Tikhonov and Ivan P. Yamshchikov. Music generation with variational autoencoder supported by history. CoRR, abs/1705.05458, 2017.