Music Generation Using Neural Networks

Qibin Lou
qibinlou@stanford.edu

Abstract

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. Early this year, Google Brain team open sourced a research project named Magenta, 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 which proposes a deep generative model of raw audio waveforms and achieves astonishing state of the art performance gain.

In this paper, we are trying to explore potential solutions to marry the merits of these two projects and create a better model for music generation. We also conduct experiments to compare the performances and advantages of Magenta's model, DeepMind's model and another model named Biaxial-RNN.

1. Introduction

Sequence learning based on LSTM has been widely explored on different fields, e.g. language modeling, natural language understanding, translation, stylized image generation and recently audio synthesis. It's usually accomplished by training a LSTM network to predict the next node in a musical sequence(e.g. Eck & Schmidhuber(2002)). Similar to a Character RNN(Mikov et al., 2010), these RNNs can be used to generate music melodies by training them on a set of short sequence of notes and then repeatedly sampling from the model's output distribution generated to obtain the next note. However, such kind of models often fall into generating over repeating notes or random sequences that lack a consistent theme or structure(tempo, chord progressions, phrasings, melodies).

Another difficulty or interesting field of music generation is the evaluation of music quality since it is largely subjective and can vary widely for any particular piece of music. Without a definitive standard for qualitatively or quantitatively evaluating a piece of music, it is difficult to judge and quality of generated music.

To address theses issues, we experiment on three different models: Melody-RNN, Biaxial-RNN, WaveNet. These three models have their own unique advantages, which result in interesting comparison of the music they generate.

To evaluate the quality of music, there are several metrics and methods to support that. Based on music theory rules, we could define metrics like notes repeating ratio, notes not in key ration, notes in motif ratio, notes in repeated motif ratio, leaps resolved ratio, composition starting with tonic ration, etc. From a statistic perspective, we could measure the quality of generated music by calculating the similarity between the generated music and the music set used to train the model. To do that, we could apply a quantitative measure based off of kernel density estimation called Indirect Sampling Likelihood(ISL). With this approach, the probability of a held-out test set is computed under the probability distribution generated by the model, returning the log-likelihood of the test set.

2. Experiment

2.1. Models

2.1.1 Melody-RNN

Melody-RNN comes from Google's open source project Magenta. One of project Magenta's main goal is to advance the state of the art in machine intelligence for music and art generation. Machine learning has already been used extensively to understand content,

as in speech recognition or translation. Project Magenta explores the other side developing algorithms that can learn how to generate art and music, potentially creating compelling and artistic content on their own. The Melody-RNN is designed as a simple dual-layer LSTM network.

Currently there are three types of Melody-RNN models. One is basic dual-layer LSTM model, which uses basic one-hot encoding to represent extracted melodies as input to the LSTM; one is Lookback RNN, which introduces custom inputs and labels to allow the model to easily recognize patterns that occur across 1 and 2 bars; the other one is Attention RNN, which introduces the use of attention to allow the model to more easily access past information without having to store that information in the RNN cell's state.

There is also a new update from Magenta team that proves that DQN network can also be applied in the Magenta generating process to work as a reward function to teach the neural network to follow certain music theories. The basic idea is as follow:

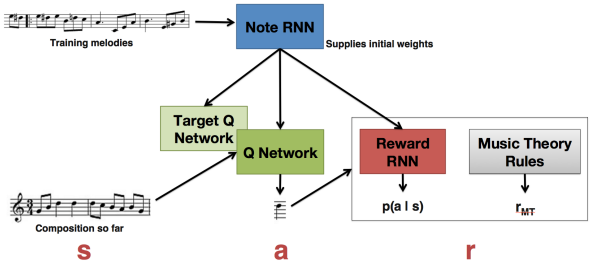

Figure 1. WaveNet Network Architecture

We choose the Attention RNN model since it gives the best and most sophisticated generated music among the three types. The open sourced model can be found at `https://github.com/tensorflow/magenta/blob/master/magenta/models/melody_rnn/melody_rnn_model.py#L170`. We set the batch size to 32, set rnn layer size to 128*128, dropout rate to 0.5, initial learning rate to 0.01, others to model defaults.

2.1.2 Biaxial-RNN

Biaxial-RNN comes from Daniel Johnson's impressing RNN music composing project. It's well designed to have the following properties:

- Understand time signature: being able to compose mostly strict to fixed time signature.
- Time-invariant: being able to compose indefinitely and being identical for each time step.
- Note-invariant: being able to transpose up and down music with identical structure of network.
- Allow multiple notes to be played simultaneously, and allow selection of coherent chords.
- Allow the same note to be repeated: playing C twice should be different than holding a single C for two beats.

It's worth mentioning that most of existing RNN-based music composition approaches are invariant in time but variant in note. Let's say we transpose one octave up and we should expect the model to generate an almost identical piece of music rather than something totally different. The Biaxial-RNN model thus designs two axes(time axis and note axis) to pass down history information along both time axis and note axis.

The input of the Biaxial-RNN model consists of note value, pitch class, previous vicinity, previous context, beat information; the output of the model consists of play probability which is the probability of a particulate note to be played and articulate probability which is the probability of a particulate note to be articulated when it's played.

The model is implemented with Theano; the first two layers are set to 300*300 and the following two layers are set to 100*50; the dropout rate is default to 0.5.

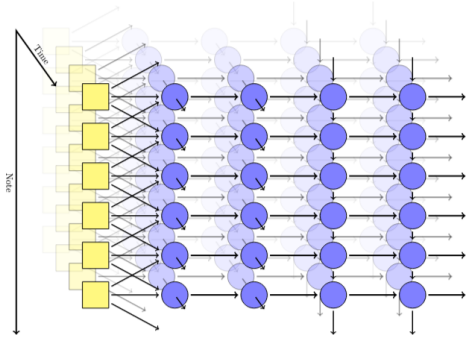

Figure 2. Biaxial-RNN Network Architecture

2.1.3 WaveNet

For WaveNet model, it's inspired by an earlier model named Pixel-RNN also developed by DeepMind. Usually, it's really tricky to model raw audio input because every second of audio usually contains 16000 samples and predicting one future sample conditioned on all previous samples is a really challenging task. While WaveNet is designed as a fully convolutional neural network, where the convolutional layers have various dilation factors that allow its receptive field to grow exponentially with depth and cover thousands of timesteps. The output audio sounds more natural than the best existing TTS systems, reducing the gap with human performance by over 50%.

At training time, the input sequences are real waveforms. After training, we can sample the network to generate music sound. At each step during sampling a value is drawn from the probability distribution computed by the network. This value is then fed back into the input and a new prediction for the next step is made. Building up samples one step at a time like this is computationally expensive, but it's essential for generating complex, realistic-sounding audio and music.

It's also worth mentioning that this model requires to have abundant training data to be able to generate a listenable music piece, otherwise the generated piece contains lots of background noises since we are training on raw audios. Thus, this model is the most computationally expensive one among three models we conduct experiments. We use the open sourced implementation of Wavenet `https://github.com/ibab/tensorflow-wavenet` and use the default model parameters.

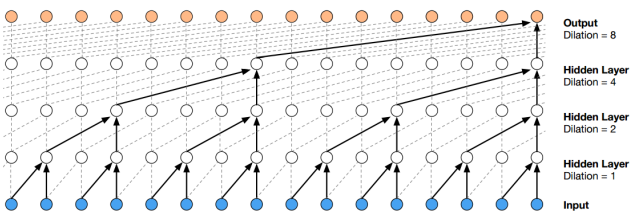

Figure 3. Visualization of a stack of dilated causal convolutional layers

2.2. Dataset

For our experiment, we choose VCTK corpus(around 10.4 GB) for first round validation on WaveNet. Since it's human voice dataset, we later apply another similar raw classical piao music files with our MIDI file set. And for Melody-RNN and Biaxial-RNN, since they are designed to work on MIDI note sequences, we collect our own training dataset which consists of lots of classical piano music categorized by composers downloaded from midiworld.com(e.g. 21 pieces of Chopin nocturnes). Those MIDI piano compositions range in length from 30 seconds to several minutes. For these two models, we sample fixed length note sequences from each MIDI file We spent most of our time applying these dataset to our models, comparing their generated sample music pieces manually.

2.3. Result

We trained our models on a workstation with 32G memory, 12 Cores CPU(without Cuda support). With the dataset we collect, we achieved pretty good result on the Melody-RNN model and Biaxial-RNN model. For Wavenet, it appears that it needs to consume large amount of raw music audios to generate a listenable piece, for our case, 200 piano music files are still not enough to train a workable model. The best output sample of our Wavenet mode can be listened at `https://github.com/qibinlou/Mozart/tree/master/output/wavenet`. We finally gave up on feeding more data into our model because it takes a long time to train a model and evaluate its quality. Our Melody-RNN model produces quite listenable monophonic piano music piece given a prime cue. However, it will fall into the over repeating rabbit hole as most RNN based models when generating a music piece longer than 16 seconds. Some sample outputs of Melody-RNN can be found at `https://github.com/qibinlou/Mozart/tree/master/output/magenta`. For Biaxial-RNN, it gives the best rhythmic music composition since we restrict on time-invariant and note-invariant properties. It also takes long to train without Cuda GPU support(5 days in our case to finish one training). Some sample outputs of Biaxial-RNN can be found `https://github.com/qibinlou/Mozart/tree/master/output/biaxial-rnn`.

3. Summery and future work

We have conducted intensive experiments comparing three different music composition neural network models. We produced pretty listenable music pieces with Melody-RNN and Biaxial-RNN while failed to take advantage of Wavenet due to lack of computation power and time.

As to future work, there are multiple places we can improve: 1) acquire more training data to enable the model to learn a better note probability distribution 2) migrate our model training to cloud platform like AWS high performance computing instances with GPU support 3) Change Wavenet's input to MIDI note sequences to give a consistency over training data input 4) Use a well-defined evaluation method like cross-entropy to better measure the quality of outputs of three models. 5) enable multi-channels note generation, which is our ultimate vision to have a smart neural network that can produce music as beautifully as songs from the Silk Road Ensemble(http://www.silkroadproject.org).

References

[1] Goolge Magenta, *https://magenta.tensorflow.org*

[2] van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A. and Kavukcuoglu, K., 2016. WaveNet: A generative model for raw audio. arXiv preprint arXiv:1609.03499.

[3] Daniel Johnson, Composing Music With Recurrent Neural Networks,

http://www.hexahedria.com/2015/08/03/composing-music-with-recurrent-neural-networks

[4] Gatys, L.A., Ecker, A.S. and Bethge, M., 2015. A neural algorithm of artistic style. arXiv preprint arXiv:1508.06576.

[5] Theis, L., Oord, A.V.D. and Bethge, M., 2015. A note on the evaluation of generative models. arXiv preprint arXiv:1511.01844.

[6] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., 2014. Generative adversarial nets. In Advances in Neural Information Processing Systems (pp. 2672-2680).

[7] Eck, D., Schmidhuber, J. (2002) A First Look at Music Composition using LSTM Recurrent Neural Networks. Technical Report No. IDSIA-07-02

[8] Breuleux, O., Bengio, Y., and Vincent, P. (2011). Quickly generating representative samples from an RBM-derived process. Neural Computation, 23 (8), 20532073.

[9] Karpathy, A., The unreasonable effectiveness of recurrent neural networks, 2015.