CS230

An Attention Based Model For Musical Rhythmical Structure

Mustafa Bayramov*
Department of Computer Science
Stanford University
mbayramo@stanford.edu

Abstract

In a natural process, language tries to create a model that describes a relation of words in a sentence by leveraging Long Short Memory(LTSM) and and attention based model. The project will exploit ideas that have already shown the practical result in the Natural language processing field.

1 Introduction

Throughout human history, Many philosophers and mathematicians explored the idea of deep connections between music and mathematics. Musical harmony and rhythm consist of deep and hidden mathematical models. Ancient Greek and philosopher studied these theories. Pythagoras, Plato, and Aristotle pioneered described a relation between music and mathematics. This work will explore and apply deep learning and attention based model in the practical aspects of music creation process.

There are many areas in computational music creation where deep learning and generative models can be applied. The project tries to focus on two particular aspects.

- One exploits the notion of repetition and rhythm in modern music.
- A generative factor.
- Secondly, create a deep neural network model that can understand and predict the harmony and relationship between musical notes.

The focus of the project is a practical application of computer-assisted music generation. Our objective doesn't eliminate the artistic and creative aspects from the musical composting process nor replaces the entire creation process. The main goal facilitates and supplements artists. That way, our primary goal is not to re-create the complete musical composition but instead focus on musical phrases. The fundamental goal is to create a system that tries to facilitate a computer-assisted music generation process. That will leverage deep learning computation models to learn a hidden structure from music composition patterns rather than compose an entire musical idea. That will enable an artist to leverage a model as another musical instrument.

*Use footnote for providing further information about author (webpage, alternative address)—*not* for acknowledging funding agencies.

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

2 Related work

The Transformer (Vaswaniet al., 2017), A sequence model based on self-attention has achieved compelling results in many generation tasks requiring long-range memory. Intuitively we know that the relationship between musical phrases closely connected and hidden model to be discovered.

Shaw et al. (2018) Self-Attention with Relative Position Representations presented an extension to the self-attention model that incorporates relative position information for a sequence that, in our opinion, is an essential factor for the music generation process.

Eck et al. [3] use two different LSTM networks – one to learn chord structure and local note structure and one to learn longer term dependencies in order to try to learn a melody and retain it throughout the piece. This allows the authors to generate music that never diverges far from the original chord progression melody. However, this architecture trains on a set number of chords and is not able to create a more diverse combination of notes.

3 Dataset and Features

- Our goal is to learn hidden structures and relationships between notes. The central focus of this work is a rhythmical structure that forms a baseline for modern music. As a primary source for input sequence, we decided to leverage a MIDI representation. The main factor in the our decision was a relatively less computation effort required. We consider an alternative option to analyze sound spectrum and relationship in the frequency domain that we believe is another fascinating research area. The initial plan was to use online MIDI databases, but late during the discovery process, it was clear that there several critical aspects in MIDI representation important to the entire system to work.

  Rhyme Net trained on professional-grade MIDI loops alternatively the entire system must implement a very sophisticated heuristic on the prepossessing phase.

  - The factor number one is the quality of resolution and correct serialization note on, note off, and delta time. MIDI protocol very particular way of how it encodes MIDI events. Of course, describing the entire protocol is beyond this study, but it is essential to define a couple. A MIDI event is a piece of data sent to a MIDI device to prompt it to do something at a specific time. Each event should contain two pieces of information: The number of MIDI Ticks, also known as the MIDI delta time, specifies what time something should be done in relative terms in ticks. Secondly, the MIDI message, which specifies what must be done. NOTE On, Pitch, Velocity. For example, one MIDI event could specify that at 1 second in the MIDI events sequence, the MIDI device should play a particular note on some channel with a specific velocity. Other MIDI events carry additional information. For example, MIDI can indicate a METADATA when tempo change in a particular track. It is essential to understand that the delta time indicates to MIDI devices will interpret this to mean that this event should occur N ticks after the previous event. Since delta time is expressed in relative terms, it is crucially important to have good data representation in the original MIDI source, alternatively model will train on incorrect data.
  - MIDI can indicate in several different way when device should stop playing particular note or set of notes. There are many variations of how specific software can serialize MIDI information, and many online databases contains only monophonic representation. An essential aspect of a MIDI representation and protocol must contain polyphonic information, i.e., multi-track re-representation. The choice of polyphonic is crucial since an attention-based model should learn not just how a single instrument forms a rhythmical structure but rather a relationship between different musical instruments.

- The second source musical instruments with existing presets will feed into a computer via the MIDI input channel and be saved as a multi-track MIDI file. That model also enables to train of a model from a live input signal. This option might be desirable for a musician, specifically for live performance. For example, a Human plays guitar solo, and a computer generates drums.

- Machines that will generate Euclidean rhythm input sequences.

Data Representation and post processing.

- Input MIDI notes represented in one hot encoded vector $X = (x_1, x_2, ..., x_L)$.
- A pause between notes will be represented as pseudo note.
- A single input vector X will represent a single instrument for a given composition.
- Each MIDI channel that re-present an instrument from a separate vector from the same music composition and all midi combined midi channels will form N X N matrix. For example, if MIDI contains six instruments, it six separate vectors.
- Each Bar quantized in time signature for a given quantization 1/16.
- Even though MIDI protocol contains on and off command for a given note, not all MIDI re-presentation use it, many re-presentation contains only velocity information. One hot vector will use that information.

4 Encoder

Since the model uses MIDI embedding, it essential to construct an encoder that provides a very efficient representation; otherwise, it computationally infeasible. In the original formulation of relative attention (Shaw et al., 2018) requires O(L2D) memory where L is the sequence length and D is the dimension of the model's hidden state

MIDI protocol sends a message containing a series of concurrent tracks, Each containing a list of MIDI events, note on and note off, meta-messages, and many others. Encoder extracts the MIDI event related to pitch, tempo change. The vocabulary of the encoder pitch range is 0-128 note-on events and 0-128 note-off-event. Same or different notes can and usually played at different velocities. If all MIDI events played the same velocity, perceptually for a human, that sounds very harsh. Therefore, velocity ranges must efficiently embed. The choice is to create a binns bucket for each velocity range and reduce the vocabulary size. Each token in vocabulary re-presents a MIDI event that attached as embedding.

5 Methods

Entire system consist of several important block. Of the most important aspect is efficient date representation. Most neural sequence models have an encoder-decoder that produce input vector [127, 128, 129] and encoder maps an input sequence of symbol representations $(x1, ..., x_n)$ to a sequence of continuous representations $z = (z_1, ..., z_n)$. Given input z, the decoder then generates an output sequence $(y1, ..., y_m)$ of symbols one element at a time. At each step the model is auto-regressive [10], consuming the previously generated symbols as additional input when generating the next. The Transformer and our model follows the architecture using stacked self-attention and point-wise, fully connected layers for both the encoder and decoder.

Attention mechanisms are shown more significant results in many sequence-based tasks. In attention network, each layer consists of a self-attention sub-layer followed by a feed forward sub-layer.

For example in Transformer defines a attention layer that first transforms a sequence of L D-dimensional vectors $X = (x_1, x_2, ..., x_L)$ into queries $Q = XW^Q$, keys $K = XW^K$, values $V = XW^V$, where W^Q, W^K, and W^V are each D × D square matrices. Each respected matrix is then split into H LD_h parts and indexed by **h**, and with dimension $D_h = \frac{H}{D}$, that allow the model to learn and focus on different parts of the history.

The scaled dot-product attention computes a sequence of vector outputs for each head as

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

The output of Z forward propagated to a next layers and at each sub-layer of network.

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

In our work, we use a similar idea. A model will try to construct a pairwise distance between notes for a given input. At the core, the algorithm needs to find a minimum number of transpositions that take one note o͡ another note b̲ased on distance.

Rhyme Net evaluated both relative and sinusoids to represent timing information between a notes. The relative position described Shaw et al. (2018), and it introduced relative position representations. It allows attention to be informed by how far two positions are apart in a sequence. That might be specifically important if we decide not to use fixed quantization for a given input sequence.

- The Rhyme Net, consists of a stacked Encoder that inputs Embedding, positional Encoding for each encoder layer. In the experimental result for midi sequence length 256 tokens, our Network produced better perceptual quality with 12 layers. The input is put through an embedding which is summed with the positional Encoding. The output of this summation is the input to the encoder layers. The output of the encoder is the input to the decoder and follows the original attention model. Similarly, a decoder layer consists of 12 decoder Layer.
- In original transformer architecture the result from each head are concatenated to form the sub layer's output. Each attention head operates on an input sequence, $x = (x1, ..., x_n)$ of n elements. and it restricts to maximum number of tokens and it set to 512). Rhyme Net uses similar approach. x_i in dimension R^{d_x}

  $$z_i = \sum_{1}^{n} a_{ij}(x_j W^v)$$

  where a_{ij} is weight computed by softmax and where e_{ij} is the attention weight.
- The model uses input vector that holds MIDI Event token of fixed size and attach embedding and it learn to convert the input midi sequence and sequence tokens to vectors of model dimension d_model. The softmax output probability distribution and it used predicts a next midi event for sequence. And similarly [6] model share the same weight matrix between the two embedding layers.
- Attention based models as it describe in [6] requires positional encoding and it must be embedded in input embedding. The positional encoding have the same dimension d_model as the embedding. I order for the model to make use of the order of the sequence, we must inject some information about the relative or absolute position.

  In original proposal [6] use sine and cosine functions of different frequencies:

  $$PE_(position, 2i) = sim(pos/10000^{2i/d})$$

  $$PE_(position, 2i + 1) = cos(pos/10000^{2i/d})$$

  Where i is the dimension. Recently, Shaw et al. (2018) demonstrated the importance of relative position representations. They presented an efficient way of incorporating relative position representations into the transformer self-attention layer.

  Relation-aware Self-Attention. Proposed an extension to self-attention to consider the pairwise relationships between input elements. In this sense, we model the input labeled, directed, fully-connected graph and the edge between input elements represented by vectors.

  $$z_i = \sum_{1}^{n} a_{ij}(x_j W^v a_{ij}^V)$$

  Our model evaluated both and relation aware is our current choice and we planing to evaluate other proposal in the future.
- We are currently experimenting with a high dropout [33] rate to simulate many missing MIDI events in the input sequence. Our model currently produces outstanding perceptual quality for the input sequence, but we evaluate the model in a smaller input sequence.
- Rhymnet uses Adam optimizer [9]

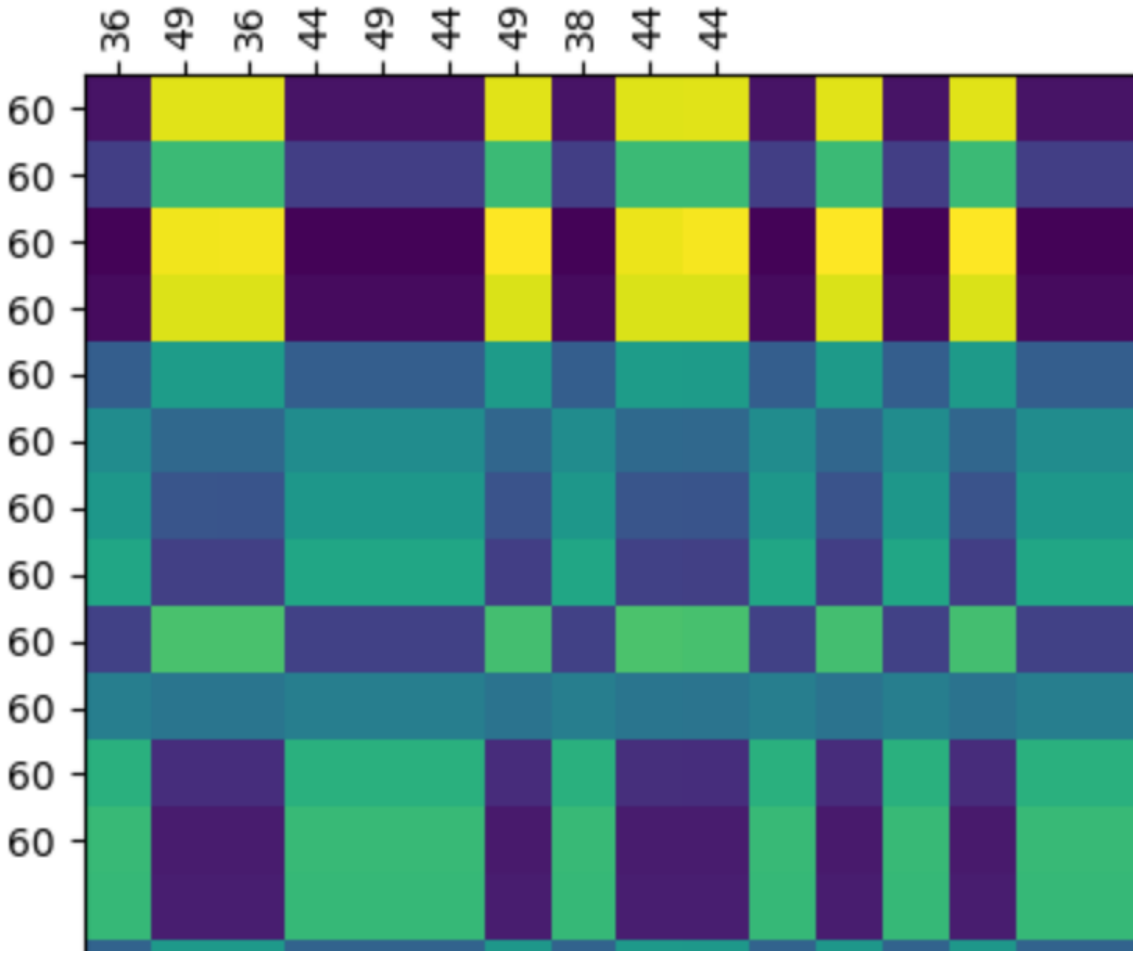
  $$lrate = d0.5min(step_num0.5, step_nwarmup_steps1.5)$$

6 Results

Our main evaluation metrics are perceptual quality. In link below all in AI section sample generated by Rhyme Net .

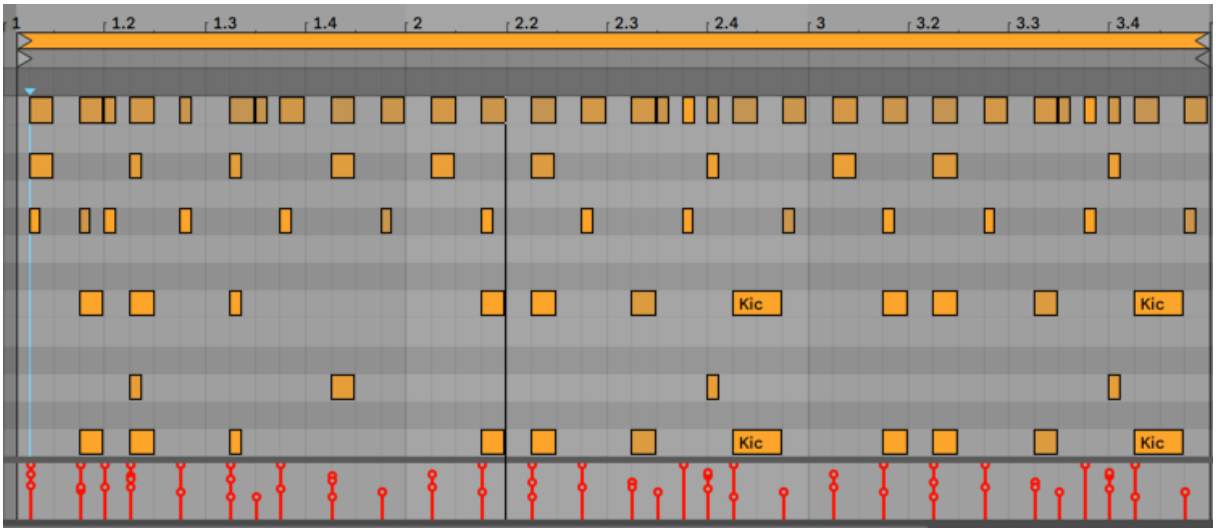
Below is example of attention weight we list of MIDI events. It seen that the model learns and influenced by set of MIDI events for a given input.

https://soundcloud.com/monkeymind



attention weight

Below is very complex MIDI structure generated by Rhyme Net. It prefect in time and quantized.



Complex MIDI Generated by Rhyme Net

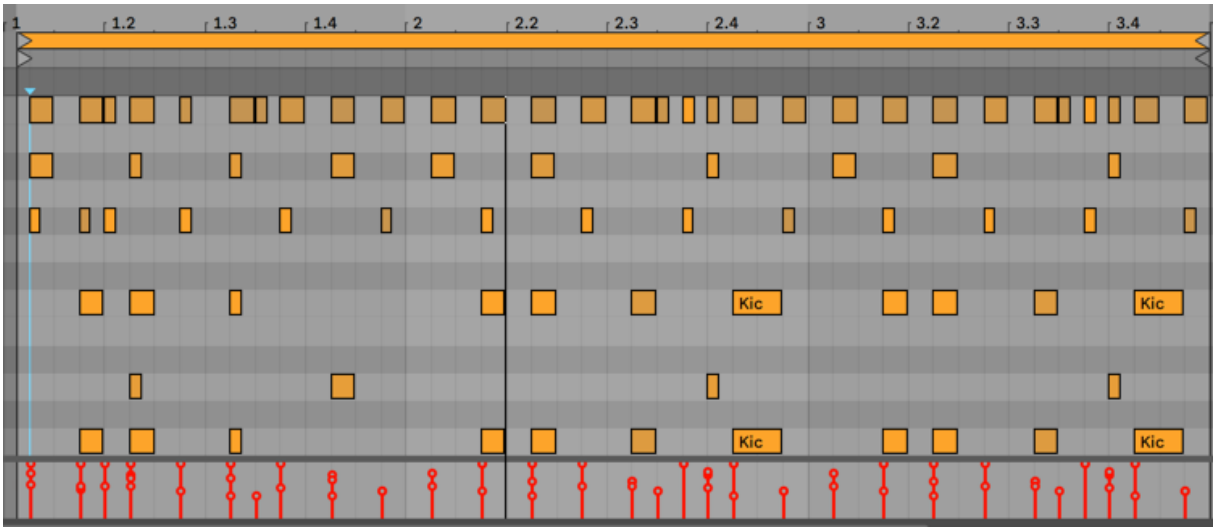
In our test model always converge to very high accuracy. Epoch 1 Loss 0.0109 Accuracy 0.9948

7 Challenges

There are a couple of challenges that are foreseen. One is a relationship between human perception and rhythm structure. For example, rhythmical repetition is an art in its own form. If the composition contains too much repetition, then human perception usually dissonance. The trained model will need to able to learn and generate a repetitive pattern with incremental changes and variation into musical phrases. For example if we take simple 4/4 time signature structure for drum each 4th bar might have small variation in rhythm that create more naturally sounded parts. Another foreseen challenge is how a model will need to learn musical pauses and rests between notes and the relationship between instruments. Our target is to be able to create a complex rhythmical structure that will consist of multiply instruments.

8 Evaluation

Link below under AI contains sample all generated by Rhyme Net
https://soundcloud.com/monkeymind



Complex MIDI composition

The primary evaluation metric is the human perception of music.

A set of chord clips.

- Model Generate sample 4 , 16, 32 bar.
- Human made sample and A and B blind test.

Evaluate model for ability to generate a continuation, repetition and basic rhythm structure for a given chosen music genre.

9 Conclusion

Evaluate different potion

References

[1] G. T. Toussaint, The Euclidean algorithm generates traditional musical rhythms, Proceedings of BRIDGES: Mathematical Connections in Art, Music, and Science, Banff, Alberta, Canada, July 31 to August 3, 2005, pp. 47–56.

[2] Sequence Transduction with Recurrent Neural Networks Alex Graves

[3] Daniel Johnson. Composing music with recurrent neural networks.

[4] Music Transformer Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Ian Simon, Curtis Hawthorne, Andrew M. Dai, Matthew D. Hoffman, Monica Dinculescu, Douglas Eck

[5] I-Ting Liu and Bhiksha Ramakrishnan. Bach in 2014: Music composition with recurrent neural network. Under review as a workshop contribution at ICLR 2015, 2015.

[6] Attention Is All You Need Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin arXiv:1706.03762 [cs.CL]

[7] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In ICLR, 2015.

[8] Vinyals Kaiser, Koo, Petrov, Sutskever, and Hinton. Grammar as a foreign language. In Advances in Neural Information Processing Systems, 2015.