Use of LSTM Recurrent Neural Networks for Generating Music

Aditi Awasthi
Department of Computer Science And Engineering
BMS College Of Engineering
Basavanagudi, Bengaluru
aditi.awas1@gmail.com

***Abstract* - Neural networks are traditionally used to tackle well-defined problems. More recently, researchers have started looking for applications regarding pursuits of a more creative nature – like art, literature and music – which have required the innovation traditionally characteristic to the human mind. In the domain of algorithmic music composition, machine learning-driven systems eliminate the need for carefully hand-crafting rules for composition.**

Index Terms – Music Composition, RNNs, LSTMs, MIDI

I. Introduction

A. Goal

The goal is to build a generative model from a deep neural network architecture which creates music with rhythm and complex structure, utilizing all types of notes including dotted notes, longer chords, and rests. This can be achieved by creating a model capable of learning long-term structure and possessing the ability to build off a melody and return to it throughout the piece.

B. Abbreviations and Acronyms

RNN – Recurrent Neural Networks
LSTM - Long Short Term Memory
MIDI - Musical Instrument Digital Interface

C. Foundational Knowledge

The capability of RNNs to learn complex temporal patterns lends itself well to the musical domain.

In an RNN, the output of each hidden layer is fed back to itself as an additional input. Each node of the hidden layer receives both the list of inputs from the previous layer and the list of outputs of the current layer in the last time step.

The power of this is that it enables the network to have a simple version of memory, with very minimal overhead.

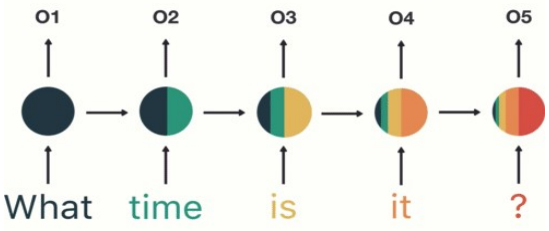

Fig. 1 Intuitive concept of the working of an RNN

One problem with RNNs is that the memory is very short-term. If a sequence is long enough, they falter at carrying information from earlier time steps to later ones.

To solve this, we can use an LSTM node instead of a normal node. This introduces:

1) Adding a forgetting mechanism. When new inputs come in, it needs to know which beliefs to keep or throw away.

2) Adding a saving mechanism. When the model sees a new input, it needs to learn whether any information about the input is worth using and saving.

3) Focusing long-term memory into working memory. Finally, the model needs to learn which parts of its long-term memory are immediately useful.

Whereas an RNN can overwrite its memory at each time step in a fairly uncontrolled fashion, an LSTM transforms its memory in a very precise way: by using specific learning mechanisms for which pieces of information to remember, which to update, and which to pay attention to. This helps it keep track of information over longer periods of time.

II. Related Work

1) Markov chains can be used to generate new musical compositions, by producing sub-sequences that also exist in the original data.

2) Early studies in this field used a multi-layer perceptron [1] to utilize gradients in order to produce music.

3) RNNs [2] attempt to extrapolate beyond the exact sub-sequences in the training data.

4) LSTMs are better than vanilla-RNNs [3] for learning longer temporal dependencies, as proved by the work of The Magenta team [4] at Google Brain.

5) Some practitioners explore alternative approaches like hierarchical temporal memory or principal components analysis.

III. Approach

A. Major challenges

1) Representation of music: Unlike text, a single moment in music can contain more than one symbol: it can be a chord, or it can have a combination of qualities that is best described by its components. There can also be long durations of silence, or states can have varying lengths. These differences may be resolved by carefully crafting the representation, or by heavily

augmenting the dataset and designing the architecture with the capacity to learn all the invariance.

2) The kind of data to use: When any automated creative system needs to be trained on a large number of cultural artifacts, it can only perpetuate the dominance of what is already well-documented.

3) The lack of global coherence or structure: While LSTMs and Transformers manage to maintain long-term consistency, there is still a gap between generating shorter phrases and generating an entire composition [5]; something that has not yet been bridged without lots of tricks and hand-tuning.

B. Training data set

Bach's chorales are highly structured and follow specific rules in their construction. Other forms of music are not always so organized. Bach's chorales always have four voices (soprano, alto, tenor and bass) that create a rich harmonic progression when played together. The nature of this structure made for good training data for the model.

The dataset [6] is constructed by extracting 8 measures long parts from the MIDI file. The input representation used is piano roll, with the pitch represented as the MIDI note number.

```
2,  96, Note_on,  0, 60, 90
2, 192, Note_off, 0, 60,  0
2, 192, Note_on,  0, 62, 90
2, 288, Note_off, 0, 62,  0
2, 288, Note_on,  0, 64, 90
2, 384, Note_off, 0, 64,  0
```

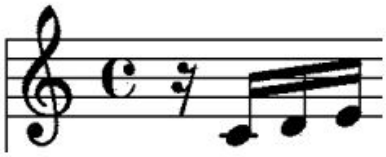

Fig. 2 MIDI file and its equivalent scales

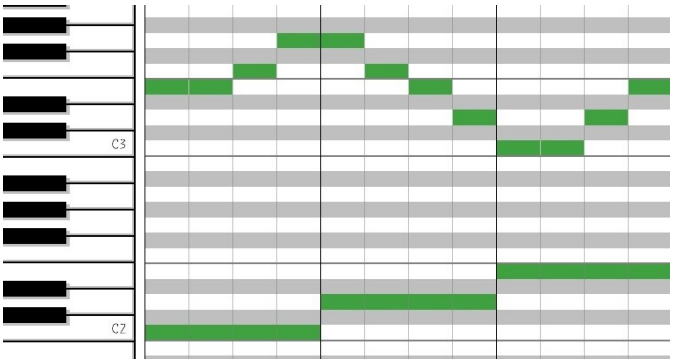

Fig. 3 Piano Roll representation

C. Properties of network design

1) Have some understanding of time signature: The neural network should know its current time in reference to a time signature, since most music is composed with a fixed time signature.

2) Be time-invariant: The network to be able to compose indefinitely, so it needed to be identical for each time step.

3) Be (mostly) note-invariant: Music can be freely transposed up and down, and it stays fundamentally the same. Thus, the structure of the neural network should be almost identical for each note.

4) Allow multiple notes to be played simultaneously, and allow selection of coherent chords.

5) Allow the same note to be repeated: Playing C twice should be different than holding a single C for two beats.

D. Implementation

The system for polyphonic music proposed by Johnson in his Hexahedria [7] blog involves:

1) The first part made of two LSTM layers, each with 300 hidden units and recurrent over the time dimension. These layers are in charge of the temporal horizontal aspect, i.e. the relations between notes in a sequence. Each layer has connections across time steps, while being independent across notes.

2) The second part is made of two other LSTM layers, with 100 and 50 hidden units, recurrent over the note dimension. These layers are in charge of the harmony vertical aspect, i.e. the relation between simultaneous notes within the same time step. Each layer is independent between time steps but has transversal directed connections between notes.

The main originality in this approach is using recurrent networks not only on the time dimension, but also on the note dimension, more precisely on the pitch class dimension. This latter type of recurrence is used to model the occurrence of a simultaneous note based on other simultaneous notes.

The three axes represented in Fig. 4 include:

1) The flow axis represents the flow of (feedforward) computation through the architecture, from the input layer to the output layer.

2) The note axis represents the connections between units corresponding to successive notes of each of the two last (note-oriented) recurrent hidden layers [8].

3) The time axis represents the time steps and the propagation of the memory within a same unit of the two first (time-oriented) recurrent hidden layers.

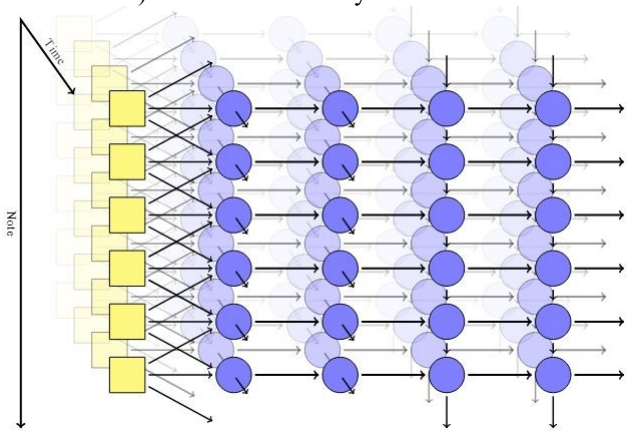

Fig. 4 Unfolded view of the three axes of the LSTM network described

The output representation is also a piano roll, in order to represent the possibility of more than one note at the same time. Generation is done in an iterative way.

IV. Applications

For companies like Google, the primary application of such audio generation is the conversion of text to speech. They could benefit immensely from how much domain-specific knowledge can be infused into the speech i.e. instead of the speech sounding like different words spoken distinctly and simply strung together awkwardly, we can embed cadence into machine-generated sound.

We can make such systems learn from the entire documented history of music with a vague goal of producing something similar, or something novel. They can construct entire compositions or improvise with us.

Few algorithms can allow the user to generate music with tunable parameters [7]. The ability to tune properties of generated music will yield more practical benefits for aiding artists, filmmakers, and composers in their creative tasks. Such innovations include methods to learn musical style and dynamics, harmonize melodies, smooth transitions between disconnected fragments of music and compose from scratch.

V. Conclusion

The aforementioned system demonstrates its ability to emulate some of the notable features of music composed by human composers such as harmony, melodic complexity, tonality, counterpoint, rhythm, and the proper use of treble and bass components as a coherent whole, thus producing music that is one more step closer to human-level composition.

There is no real expression of self, style, beauty, etc. in merely following a set of rules. Even theorists don't think rules are the be-all and end-all. That's not to say rules are not important, but a composer of common-practice tonality should have a good enough command of the style that the rules are followed instinctually, and the focus is on expressing a musical idea (that just happens to follow the rules).

Acknowledgment

The work reported in this paper is supported by the college through the TECHNICAL EDUCATION QUALITY IMPROVEMENT PROGRAMME [TEQIP-III] of the MHRD, Government of India.

References

[1] Lewis, 1988—"Creation by Refinement: A creativity paradigm for gradient descent learning networks" *in International Conference on Neural Networks*.

[2] Todd, 1989 – "A connectionist approach to algorithmic composition" in *Computer Music Journal*

[3] Eck & Schmidhuber, 2002—"Finding temporal structure in music: Blues improvisation with LSTM recurrent networks" in *IEEE Workshop on Neural Networks for Signal Processing.*

[4] Natasha Jaques et al., 2016 – "Generating Music by Fine-Tuning Recurrent Neural Networks with Reinforcement Learning" *at Deep Reinforcement Learning Workshop, NIPS*

[5] Dieleman et al., 2018—"The challenge of realistic music generation: modelling raw audio at scale" - arXiv:1806.10474v1

[6] Jean-Pierre Briot et al., 2019 - "Deep Learning Techniques for Music Generation - A Survey" - arXiv:1709.01620v3.

[7] Huanru Henry Mao et al., 2018 - "DeepJ: Style-specific music generation" - arXiv:1801.00887v1.

[8] Humphrey & Bello, 2012—"Rethinking automatic chord recognition with convolutional neural networks" *in International Conference on Machine Learning and Applications (ICMLA)*.