Character-Level Based for Music Modeling

Dat Nguyen, Tiong On Saint
{dnuyen52, otiong}@cse.unl.edu

***Abstract* —** Deep learning method recently achieved great success in summarization, machine translation, and other sequence generation task. Realistic music generation is a challenging task. The preprocessing process typically involves high representation of MIDI format note or scores. In music theory, chords and duration are important for the representation of music that sound reasonable to human ear. Pure note and chord combination representations are a problematic because they have a high number of features representation. In this work, we explore ASCII character base music modeling as a means to enable lower representation of music notes, chords, and durations meanwhile still maintaining dynamic information like chords and duration in a tune. We utilize LSTM network to reproduce a music piece and ask numbers of participants to grade the music quality.

Keywords—LSTM, Music Generation, abc Notation

I. INTRODUCTION

Music has always been around us for centuries. It has provided people a way to express themselves. However, music composition is a challenging task and new ideas are hard to come by without inspiration. Recently, advanced algorithm in the field of deep learning has allowed us to explore music composition in building probabilistic models.

Models such as LSTM could be trained with several musical styles and genres and this has been done in the past. In this paper, we will introduce abcRnn, a model which can assist musicians in producing new music in a specific genre. A lot of work done with music composition have used some other form to represent the musical notes. Our main contributions in this work is to explore ASCII character representation to reduce the features (representation of musical notes), which will reduce the time spent during training.

A. ABC Notation

Abc notation are simple and powerful ASCII notation format. Abc is widely used to distribute tunes, particularly on the internet. A tune notated in abc can be directly read using multiple open source software like EasyAbc which can be converted to mp3 or music sheet.

Abc file contains a header section and tunes section. The header section is where the author can describe the music or make comments on the tunes and the tunes section is where music notes are placed. A complete example of a simple midi song in abc notation is shown in Figure I.1. Example in Figure I.1 shows an example including 4 header, where 'X' indicate the idx of the tunes, 'T' describe the title, 'M' describes the time signature of the tunes, and most importantly 'K' describes the key signature, the arrangement of sharp or flat signs on particular lines and space of musical notes.

```
X: 1
T: A Cup Of Tea
R: reel
M: 4/4
L: 1/8
K: Amix
|:eA (3AAA g2 fg|eA (3AAA BGGf|eA (3AAA g2 fg|1afge d2 gf:
|2afge d2 cd|| |:eaag efgf|eaag edBd|eaag efge|afge dgfg:|
```

Figure I.1 Example of abc notation tunes. Contains header and tune sections.

B. PROBLEM DEFINITION

Artists are having a hard time composing original music, music generation using deep learning techniques will be able to give artists inspiration on generating new music pieces. The type of problem we are solving can be stated as a sequence classification problem, where there is a sequence of inputs over time and the model will predict a note or chord for the sequence. This problem is challenging because sequences' length varies. Problem comprised of a very large vocabulary (chords) and models would have to learn their temporal information between the input sequence. Many artists will be inspired by the music generated; hence productivity will be high. Moreover, other problems that are related to sequence classification can benefit from this too. We believe that this can bring value to everyone interested.

The preprocessing method that most used, which is to convert midi files into music21 objects, is considered inefficient because each chord will be considered a feature. In this paper we will use abc notation for modeling. Our hypothesis is that it will make the model more efficient, hence the training and sampling time will be lower.

The input to our algorithm will be pure ascii character or a series of ascii from an abc file. We use LSTM, a variation of Recurrent Neural Network, to generate a news sequence of ascii char of abc notation then compile to music with the aim of making good music.

Success of the model during training time is judged based on the prediction of the model on the next note against the actual character as the cross-entropy score. For evaluating the effectiveness of the model on making music, we randomly sample tunes from different epochs then ask participants to evaluate the music piece. Midi format typically utilize to convert to piano roll. One main problem with symbolic notes representation is the amount of notes configuration is intractable. As unique notes are model as combination of different chords, notes and time signature.

II. Related work

Similar to languages, music can be represented in many readable abstract forms. the most direct representation is from raw audio signal: the waveform. Abstract music representation concern with concepts like notes, duration and chord. MIDI is one of the most popular formats that can represent all of this information. Melody can also be encoded as text in many supportive formats such as markup language and abc.[8]

Probabilistic base music generation has been existing for long time. [2] First, attempt made use of markov chain and transition matrices to model properties of each notes. In the field of recurrent symbolic generation, Andrej have successfully generated recognition latex using LSTM network. many monotone music generation algorithms based on LSTM trained on melody text representation also exist like folkrnn.com [4]. [5] Generative Adversarial Network so far show most promising result. Wavenet is a famous successful model to generate human like music.

As reported by Kang [3] in the past year, there has been many successive interests in machine learning based music composition. Ranging from jukedeck[9], industry base purposes to Magenta, an open source base research.

Many works [11, 12, 13] investigate on modeling temporal dependencies in high dimension using piano roll music format on generating music. Piano roll use symbolic note sequence as represented in music sheet.

III. Dataset and Features

We use 2 main method to acquire our dataset. First methods, we went on the web to collect abc file. Second, we collect classical and jazz midi file then convert it to abc notation using EasyABC api then extract out the header. In total, we collected 3 dataset each of ~130k characters in abc notation format

A. Data preprocessing

Data processing in our work mainly involves filtering out special and comment symbols used in abc notation. These symbols will not bring any value to the music and training data. It is removed to prevent syntax issues. We use the standard fraction of 80 – 10 – 10 % for train, validation and test set respectively

B. Features

Dynamics in music is defined as the relative speed of each note played. abc notation encapsulates dynamics in music tunes by placing escape symbol to represent duration of chords or notes. Our model is training based on ASCII character representation. We keep track of the characters in a matrices of N x W, where element in the matrix are numbers mapped to ASCII character. We chose N, batch size to be 16 and W, Sequence size to be 64 at training times for the algorithm to capture the long memory sequence as possible without being too computationally expensive.

$$\text{Input} = \begin{pmatrix} x_{0,0} & x_{0,1} & \cdots & x_{0,W} \\ x_{1,0} & x_{1,1} & \ddots & x_{1,W} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ x_{N,0} & x_{N,1} & \cdots & x_{N,W} \end{pmatrix}$$

The target is encoded as a one hot coded vector represented in a matrices of N x W x V. Text ASCII representation hold between 35 - 67 unique V vocab size depending on the variety of abc music encoded sequence. Note that this representation does not make any prediction on time signature.

C. Metrics

Given a training batch containing N input-output sequence pairs, we use the standard methods of Estimate

$$\theta_{MLE} = \sum_{i=1}^{N} y log\, \hat{y} \quad (1)$$

To qualitatively evaluate the music composed by the model, we ask participants to listen then rate to the generated tunes on the scale of 0 to 10. The survey was set up with 6 tunes including 2 human composed songs from the training dataset and 4 algorithm composed tunes. The placement of tunes is arranged randomly. We also collect the feeling of each participant base on scale of 0 to 5 as suggested by Gary McPherson [10] when evaluating music. Tunes are sampled out randomly from different training epochs ranging from 10, 100 and 300 epochs. Each tune is then reported with its average score and average feeling score range from 0 to 5 by participants.

IV. Methods

A. Hidden Markov Model

Hidden Markov Model makes the Markovian assumption, which assumes the future state depends only on the current state, not on the events that occur before it. To solve the sequence classification problem like music generation, we need a model that can find long term dependency, which is what Recurrent Neural Network can provide. Since we are using a complex and large dataset, Recurrent Neural Network will be the better choice [7].

B. Recurrent Neural Network

Recurrent neural network is similar to artificial neural networks with a feedback loop instead of a feedforward neural networks, which only flows in only one direction from input to output. The output of the recurrent neural network is added to the next input and fed back into the same layer.

Moreover, vanilla neural networks are too constrained. They accept a fixed size input and produces a fixed size output. On the other hand, recurrent neural network does not have strict requirements for its input and output, hence allowing flexibility for the model.

Similar to artificial neural network, recurrent neural network uses backpropagation and it is prone to the vanishing gradient problem which will lead to exponentially small gradients. This problem has led us to implement gating - a technique that makes a decision for the network to forget or remember the current input for the future.

C. LSTM

Vanilla recurrent network is often faced with problems such as vanishing gradient and lack of memory. In our case, the model needs to successfully learn the syntax and generate a well formatted sequence of characters to generate a successful sequence to compile abc notation to music notes. The main difference between vanilla RNN vs LSTM is the hidden cell distributed at each time step. LSTM is designed to fight with exploding and vanishing gradient. Apart from hidden vectors, each layer saves a hidden vector. A cell can write, forget based on a gating mechanism.

$$i_t = \sigma(w_i \cdot [h_{t-1}, x(t)])$$

$$f_t = \sigma(w_f \cdot [h_{t-1}, x(t)])$$

$$o_t = \sigma(w_o \cdot [h_{t-1}, x(t)])$$

$$g_t = tanh(w_g \cdot [h_{t-1}, x(t)])$$

Gated recurrent network allows to regulate the information the network remembers overtime. If memory unit c closer to one more memory is retaining. if closer to 0, then the memory vanished or forgot (applied by the forget gate). A visualization of a single lstm cell is shown in Figure IV.1.

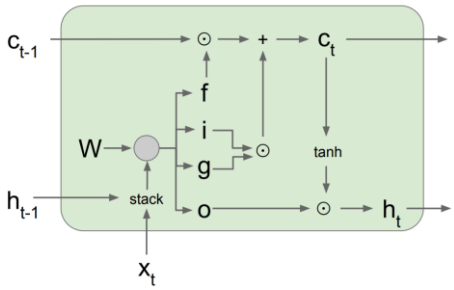

Figure IV.2: describe the direction flow of a single lstm cell unit(Image source: Fei Fei li et at. [17])

As reported by Chung[14, 15], Base on amount of data LSTM gated RNN outperform GRU and RNN in long run. Intuitively, the performance of LSTM make sense in the long run because of the complex structure that LSTM cell holds. Mainly the forget(g) gate allow network to gain longer temporal information

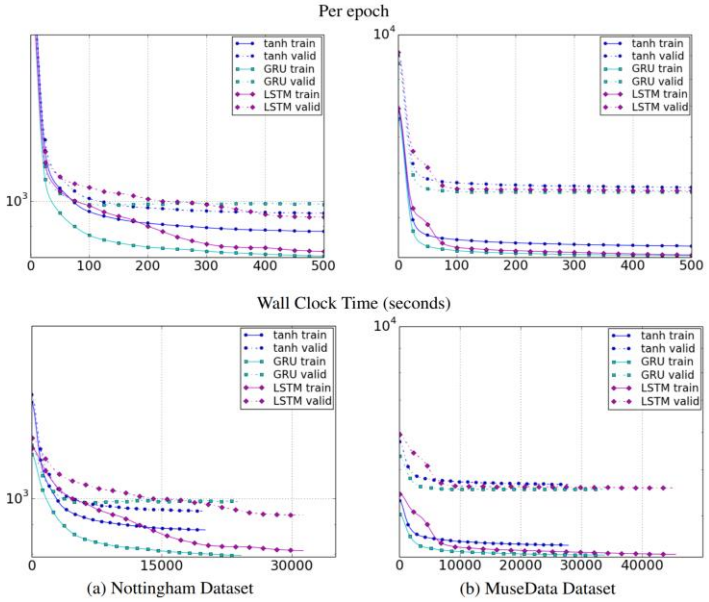

Figure IV. 3: Learning curve for training and validation set of different types of unit with respect to(top) number of iterations and bottom the wall clock times. y-axis corresponding to the negative log likelihood of the model shown in log-scale (Figure source: Chung[14, 15])

V. Experiments/Results

A. Hidden Markov chain

Our HMM model was implemented in python utilizing hmmlearn library. We used first order Discrete version of hmm provided by hmmlearn under 'MultinomialHMM' object. we chose number of hidden states to be equal to the number of the dataset vocab size. Optimizing EM problem is challenging. since EM algorithm is a gradient-based optimization method, algorithm generally get stuck in local minima. we trained 5 hmm model and achieved a lowest negative log score of -121000. Model output abc notated ASCII would fail to compile in music21. We had to dive in and fix model output abc text. Figure V.1 show an example of music sheet converted from hmm model output.

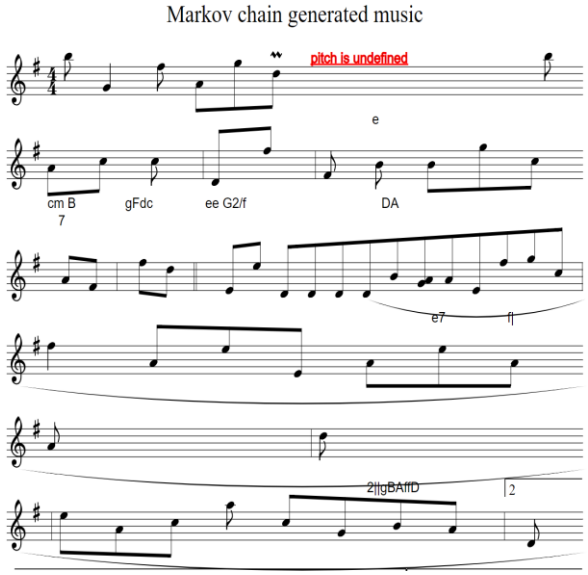

Figure V.1: Example of music sheet converted from hmm abc notation output

B. LSTM

Our LSTM model was implemented in Tensorflow keras resembling an implementation from Gaurav Sharma [5]. Our model consists of 1-layer stacked LSTM with 256 and 512 neurons, respectively. Input of LSTM is from an embedding layer with embedding size of 256. The algorithm was optimized using Adam algorithm with a .01 learning rate, $\beta_1 = 0.9, \beta_2 = 0.99$. High number of model parameter and model complex cause model to highly overfit. We added l1 + l2 regularization of 0.001 and 0.0001 respectively. We found that naively stacking deeper network cause the network to have less confidence in prediction. The validation accuracy curve would stay constant while validation cross entropy loss increasing.

C. LSTM variation (Efficient LSTM with skip connection)

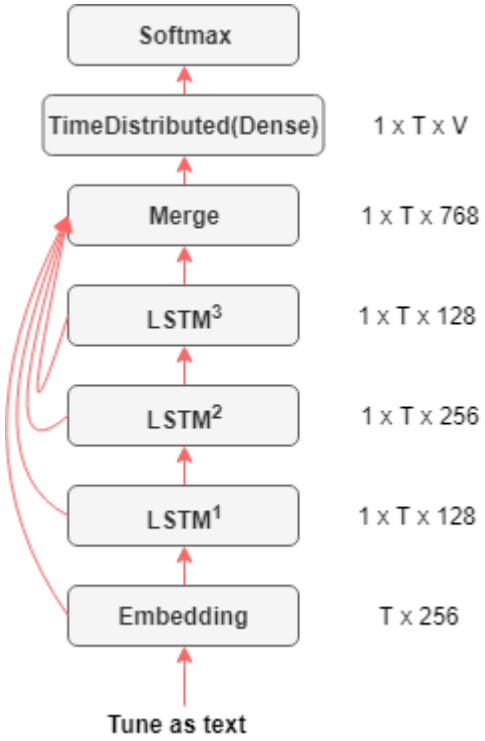

Figure V.2 Illustrate network archetecture we used in this work with T being the text length and V being the vocab size.

Deep stacked LSTM often gives better accuracy over shallow stacked LSTM. Naively, stacking LSTM layers often cause network to become too difficult to understand and computationally expensive. We investigated a novel LSTM architecture inspired by Deepmoji[16] without using attention. Input goes through an embedding layer with embedded size of 256. The network depth construct of 128, 256, 128 lstm layers neuron size. A low number of neurons on each layer allows us to stack lstm layers deeper with low computation cost. This network holds a smaller number of parameter and result in similar cross entropy loss. The lowest validation accuracy achieved by the model is 0.74. This model is then evaluated on the test set with accuracy of 0.72. This is expected as shown in [11, 12, 13] work on similar dataset using piano roll music representation. The loss curve for training and validation set is shown in figure V.3

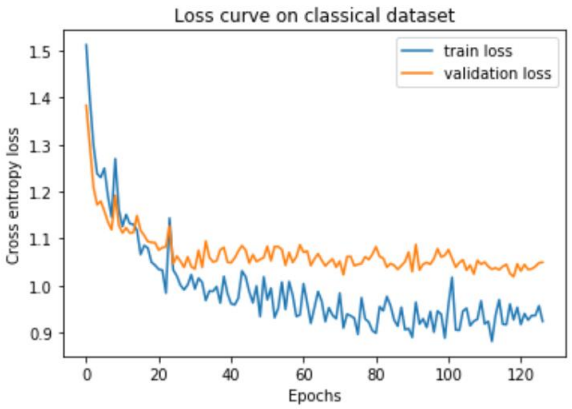

Figure V.3 Loss and validation entropy loss of stacked Residual LSTM network

The skip connection is added to increase model complexity without being too computationally expensive. Skip connection from all sandwiched layers also help information from all layer's flow to main output dense. stacked LSTM with skip connection output neat and compilable abc notated text. an example music sheet converted from network abc text output is shown in figure V.4.

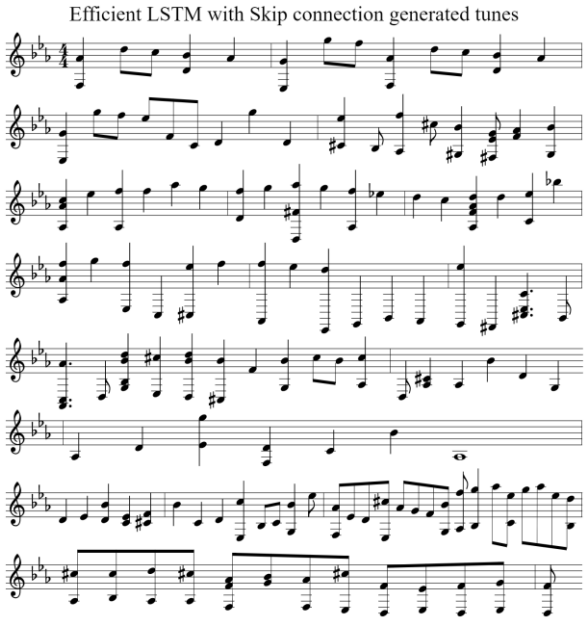

Figure V.4 example of generated music sheet converted from LSTM network with skip connection abc output

C. Overview

We had total of 14 participant in our study to evaluate our best model performance. We used the efficient skip LSTM network to generate a batch of 30 songs then 4 are selected to be in the Turing test. The Turing test include of 6 songs in total. Two human played songs is added to avoid bias. In contrast, 7 participant classified a highest bot played score songs as human played with average score(both bot and human classified) of 5.9. Human score is classified right 50% of times with average score of 6.3.

	Score(0-10)	Total classified as human played
Model performance	5.9	7 out of 14
Human play	6.3	7 out of 14

Table a. Participant evaluation on survey with randomly arranged human and bot played songs

VI. Conclusion and Future Work

From the results above. we can conclude that the usage of ASCII characters representation can be used to model music generation and the efficiency in terms of training complex datasets is useful. Human judgement is also needed to evaluate the music, as we cannot rely blindly on the evaluation metric. In the future, we are planning to make a music generation application for artists to use. An algorithm is needed to run on each model generated songs to detect plagiarism if apps is deployed for open publ

VII. Code & Music

Our code can be found in https://github.com/abcrnn/music_generation.python

Survey can be found under https://abcrnn.github.io/model-eval

A list of well generated tunes by all of our model can be found here(https://soundcloud.com/datnguyen5653/sets/abcrnn-great-sample)

VIII. References

[1] available at http://www.lesession.co.uk/abc/abc_notation.htm

[2] Qi et al. (2007. hidden markov base to produce music. Available at https://ieeexplore.ieee.org/document/4355329/.

[3] Oord et al.(2016) . Available at https://arxiv.org/abs/1609.03499

[4] Kang et al. (2018). Project milestone: Generating music with Machine Learning. Available at http://cs229.stanford.edu/proj2018/report/18.pdf

[5] Available at https://medium.com/datadriveninvestor/music-generation-using-deep-learning-85010fb982e2

[6] Wu et al, (2016). Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. arXiv:1609.08144v2

[7] Panzner, M., & Cimiano, P. (2016). Comparing Hidden Markov Models and Long Short Term Memory Neural Networks for Learning Action Representations. *Lecture Notes in Computer Science Machine Learning, Optimization, and Big Data*,94-105. doi:10.1007/978-3-319-51469-7_8

[8] Briot et al. (2017). Deep Learning Techniques for Music Generation – A Survey. Available at https://arxiv.org/abs/1709.01620

[9] https://www.jukedeck.com/

[10] Mcpherson, G. E., & Thompson, W. F. (1998). Assessing Music Performance: Issues and Influences. Research Studies in Music Education, 10(1), 12-24. doi:10.1177/1321103x9801000102

[11] advances optimizing recurrent network

[12] modeling temporal dependencies in high dim sequencese: application to polyphonic music generation and transcription

[13] Modelling Symbolic Music: Beyond the Piano Roll. Available at https://arxiv.org/abs/1606.01368

[14] Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. Available at https://arxiv.org/pdf/1606.01368.pdf

[15] Recent Trends in Deep Learning Based Natural Language Processing . Available at https://arxiv.org/pdf/1708.02709.pdf

[16] Deepmoji. Available at https://arxiv.org/pdf/1708.00524.pdf

[17] Recurrent neural network lecture. Available at https://www.youtube.com/watch?v=6niqTuYFZLQ