Composing Music using Neural Networks

Machine Learning (CS 6140) Project Proposal

Manthan Thakar Rashmi Dwaraka Tirthraj Parmar

1 Problem description

Recurrent Neural Network have been used to train the character level and word level language models to generate texts for variety of tasks, ranging from Shakespeare to Latex code[1]. For this project, we're interested in using similar models to learn character level language models of music sheets. This can be used to generate new music compositions.

The primary goal of the project is to train a character-level language model over the dataset containing notes for various music compositions, which can be formally described as:

$$P(c_1, ..., c_n) = \prod_{i=1}^{n} P(c_i|c_1, ..., c_{i-1})$$

Here, $P(c_i|c_1, ..., c_{i-1})$ is the probability of observing a character at location i in the given character sequence, when we have already observed characters starting from 1^{st} location to $(i-1)^{th}$ location.

For our purposes, the input is the sequence of characters in the dataset and the output is the probability distribution for observing the characters in the vocabulary V, given character sequence I is observed. Since this is a generative model, we can use the language model to generate new text with the estimated probabilities. We think that it would be interesting to assess the efficacy of such mechanisms to model music composition.

2 Data

We plan to use Irish folk music dataset[2, 3, 4] cumulatively consisting of 50000 tunes. All the data is textual and is encoded using ABC notation[5]. The ABC notation is very well supported through tools that can be used to convert the ABC format text files to audio format easily, which will allow us to listen to the new compositions generated by trained model. Moreover, similar datasets have been used in the past to generate interesting results for music transcription modeling[6].

3 Algorithms

Since we are dealing with musical notes, we used models that can express a sequence of directly observable events as a probability distribution. We plan to implement three models; N-gram Language Model and two flavors Recurrent Neural Network (LSTM and GRU). The language model derives probability distribution with Markov assumption and MLE based on character frequencies whereas RNN based models model the language model by means of hidden states and loss function.

3.1 N-gram Language Model

The intuition behind this model is to assign probabilities to a sequence of events in order to predict the next possible sequence of events based on a given sequence. Formally, models that assign probabilities to sequence of words are called language models or LMs[7]. However, instead of a sequence of words we use a sequence of musical notes encoded using ABC notation[5]. Once we get trained probabilities, we can use them to generate a new sequence of notes. Using N-gram language model, we can formulate our problem as

$$P(c_1, ..., c_n) \equiv exp\left(\sum_{k=1}^{n}\left(P(c_k|c_{k-N+1}^{k-1})\right)\right)$$

Trochidis, et el. have created an N-gram language model to generate Carnatic style percussion[11].

3.2 Recurrent Neural Network

Although vanilla neural networks are used for variety of regression and classification tasks they are known to work poorly on sequential data. Recurrent neural networks are considered more effective for sequential data due to their ability to capture the calculations done for previous inputs in the hidden layer. Since we are dealing with sequential data, we believe that recurrent neural network is a reasonable algorithm to use.

We can use the recurrent structure of RNNs to model the probability distribution of next likely characters given an input character sequence. An example of a small RNN depicting this idea can be seen in figure 1. Here, we take a training example *hello* and feed it to RNN. Notice that the target (y) for each character in the sequence is the next character in that training example (e.g. e is the target for h). Therefore, the network tries to maximize the probability of e when the input character is h. Here, W_{hy} is responsible for capturing the previous calculations (weights). Moreover, we can use cross-entropy [10] loss function to back-propagate errors in the network.

Character-level RNNs are used in problems like generating C code [1] and more recently for generating music transcriptions[6] as well. We'd like to extend this idea and apply it to ABC format files.

Since the simplest form of RNN is known to suffer from vanishing gradients, we plan to implement RNN variants based on **LSTM** [8] and **GRU** [9] cells.

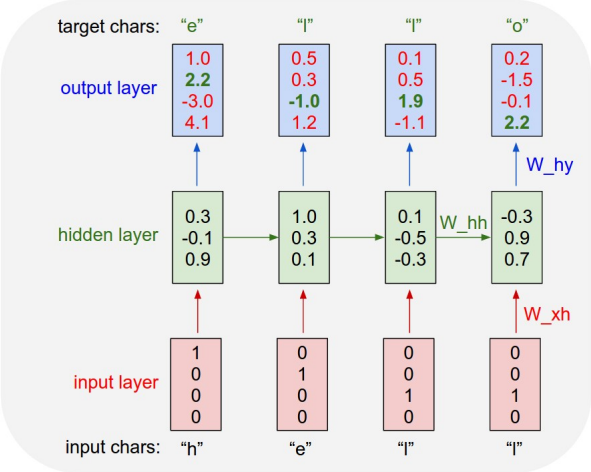

Figure 1: Character Sequence `credits: http://karpathy.github.io/`

4 Results

Applying the methods described above, we can obtain a generative model. Using the probability distributions assigned by this model, we can generate new sequences that represent new music compositions. Since the generated sequences will be in ABC notation, we can convert them to mp3 format for manual assessment of the results.

We plan to conduct a user survey to evaluate generated music based on user perception. The survey will include generated compositions by the model as well as some human created tunes. We expect the users to identify some of the model generated tunes as human created tunes.

References

[1] *The Unreasonable Effectiveness of Recurrent Neural Networks*
http://karpathy.github.io/2015/05/21/rnn-effectiveness/

[2] *O'Neill's Music of Ireland*
http://trillian.mit.edu/~jc/music/book/oneills/1850/X/

[3] *ABC version of the Nottingham Music Database*
http://abc.sourceforge.net/NMD/

[4] *Folk music style modelling using LSTMs*
https://github.com/IraKorshunova/folk-rnn/tree/master/data

[5] *ABC notation*
https://en.wikipedia.org/wiki/ABC_notation

[6] *Music transcription modelling and composition using deep learning*
Bob L. Sturm, Joao Felipe Santos, Oded Ben-Tal and Iryna Korshunova

[7] *Daniel Jurafsky and James H. Martin. 2000. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition (1st ed.). Prentice Hall PTR, Upper Saddle River, NJ, USA.*
https://web.stanford.edu/~jurafsky/slp3/4.pdf

[8] http://colah.github.io/posts/2015-08-Understanding-LSTMs

[9] http://en.wikipedia.org/wiki/Gated_recurrent_unit

[10] https://en.wikipedia.org/wiki/Cross_entropy

[11] *Trochidis, Konstantinos and Guedes, Carlos and Klaric, Akshay Anantapadmanabhan Andrija. CAMEL: Carnatic Percussion Music generation using N-gram models*
http://smcnetwork.org/system/files/SMC2016_submission_75.pdf