2P-07

Generating Homophonic Music with LSTMs Dedicated to Melody and Harmony

Clément Saint-Marc*, Katunobu Itou*

1 Introduction

The goal of this research is to create a Neural Network model capable of generating music without any user input.
The generated music should be "meaningful", that is to say, it should sound like it has a purpose. This is similar to how a human composer would write music. Throughout the years, many attempts have been made at creating music procedurally. However, very few of those attempts were concerned with the actual "meaning" of the music that was generated.
This idea of "meaning" makes a composition tell a story or express feelings. This is the reason humans write music, as well as create other forms of arts. As the goal of artificial creativity is to approach human creativity as close as possible, Artificial Intelligence should try to imitate humans as close as possible. Therefore, it is important for a music generating AI to understand "meaning" in music.

2 Approach to composition

To define "meaning" in music, it is first necessary to identify its key components. In modern western music, those components are melody, harmony, and rhythm [1].
It appears as evident that the most important musical component in giving a musical piece its identity and its "meaning" is the melody, as the other components revolve around it.
Therefore, it becomes necessary to understand what makes a melody. As we have already defined, a melody consists of motifs, which are short successions of notes. Those motifs can have variations, which for the purpose of this research can be defined as small changes to those motifs (either in pitch or in rhythm). By using motifs and variations, a melody can be given purpose. It can tell a story, and is therefore given meaning.
In order to approach composition procedurally, it is necessary to separate the composition process into several sub-processes.
For the purpose of this research, this can be summarized into two elements: melody generation and harmony generation.
Melody generation should be done using a randomly generated seed, which means that no initial input is required. It should also be set to a fixed number of measures, so that it can be repeated, and used for the harmony generation.
The harmony generation process should generate a chord progression that effectively supports a generated melody, using that melody as input data.
It now becomes necessary to determine what kind of music should be used as training data for the Artificial Intelligence. Since the most important feature is meaning, it should be music that uses motifs in the most effective way possible.
Such use of motifs is usually found in video game music, or movie scores. This is due to the necessity in such genres to represent ideas, characters, or locations. By using motifs to compose a meaningful melody, a composer can represent the story of a video game or a movie as music.
It is also necessary to have harmony (in the form of a chord progression) that supports the melody, to give it context, which helps in telling that story.
Therefore, the training data should consist of such genres of music, and have both melody and harmony. Furthermore, it should be possible to separate the melody and the harmony to train two separate AI models, one specializing in melody generation, and the other in harmony generation.

3 Method

To represent the music data, intervals relative to the key of the musical piece were chosen, instead of absolute pitch representation. This is akin to *movable do solfège*, where each degree in the musical scale is expressed as a different syllable (i.e. *Do, Di, Ra, Re,* etc.).

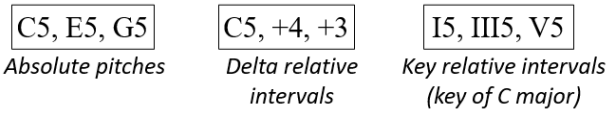

Figure 1. Kinds of music data representation.

For the actual melody and harmony generating models implemented in this research, bidirectional LSTMs are used, as they are able to access a more global context [2] when training, which is necessary when composing music since a given note depends on the notes both preceding and following it.
Due to the small amount of training data available, an algorithm that extracts the outline of melodies has been implemented. It is used to extract the outlines of the melodies used for training the melody generator. The extracted outlines are used as additional training data.
The following figure shows the topology for the melody generating model.

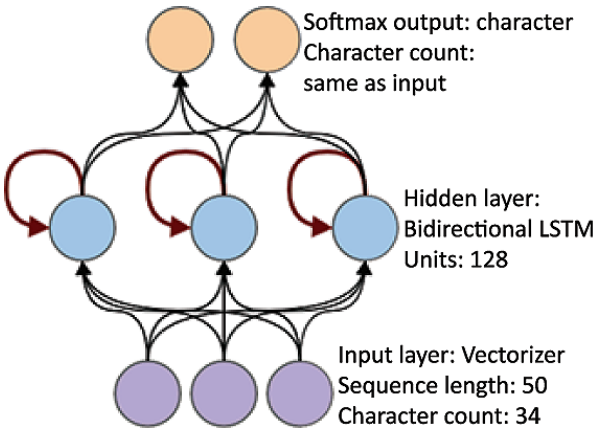

Figure 2. Topology of the melody generator.

* Graduate School of Computer and Information Sciences, Hosei University

Each interval is converted into an ASCII character based on its value. The first character outside the ASCII range, 'Ā', is used to represent the time (one character corresponding to a 64th note triplet).
This is done for each "interpretation" of the melody. Each melody is thus converted and added to the training set, with a special character, 'ā' (the second unicode character outside the ASCII range), being used as a delimiter between the melodies. The melodies are then split into sequences of 50 characters overlapping each other, and converted into one-hot vectors.
The network is then trained for 15 epochs.
For generation, the input data consists of a seed randomly sampled from the distribution of the training data. A separate LSTM model is specifically trained to this end. This input is processed by a bidirectional LSTM layer consisting of 128 units.
The output is a prediction of the next character. The prediction process is repeated iteratively for the desired length of the melody to be generated. The activation function for the output layer is *softmax*.
To study the relationship between melody and harmony, which is necessary to implement the harmony generator, a simple GRU-based classifier was implemented to label chord symbols to snippets of melodies.
The training data for that GRU consists of melody snippets labeled with chord symbols.

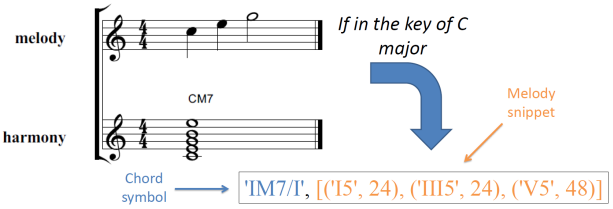

Figure 3. Example of a melody snippet over a chord represented as a sequence.

The input data consists of those sequences of notes, vectorized as sequences of a maximum length of 50 words. The vocabulary size of the vectorizer is 250 words.
This input is processed by a GRU layer consisting of 128 units. In this case, a GRU layer was used due to getting better results than with an LSTM layer, which is due to the relatively small size of the training data [3] (around 45,000 samples).
The output is a prediction of the chord symbol corresponding to the input melody snippet. There are 47 chord labels. The activation function for the output layer is *softmax*.
Once the relationship between melody and harmony had been studied, and promising results obtained, the harmony generator could be implemented.
As there is more training data as well as more chord labels than with the previously used dataset, a bidirectional LSTM layer with 128 units is used this time. Otherwise, the principle is exactly the same.
A chord symbol, which includes the chord root and the chord quality, is predicted for a given melody snippet.
Given a generated melody as input, a pre-processing algorithm divides that melody into snippets of around the same length: four quarter notes. Those snippets are then used as the inputs for the harmony generator, and a chord is predicted for each one, then added on top.

4 Evaluation of output

To evaluate the quality of the output of the melody generator, subjective evaluation is necessary. This would include both musically trained and non musically trained listeners.
For the GRU-based classifier, a validation accuracy of around 78% was reached for a training accuracy of around 83% after training for 50 epochs.
Even though this suggests overfitting, 78% is still more than acceptable for predicting something as arbitrary as a chord based on the melody it corresponds to.
Those results show promise for the harmony generator.
Unlike the GRU-based chord classifier, the harmony generator (bidirectional LSTM) does not converge during training (the accuracy remains at around 30%).
However, this is most likely due to there being a lot of chord that are similar in feeling, but have a different chord name.
Chords such as *C* and *CMaj7* are virtually the same since the latter contains all the notes of the former.
It should therefore be possible to change the target labels into chord families (*e.g.* the *C* family using the two previous examples) during training, instead of individual chords.
However, this has not been done as satisfactory results were still obtained during testing.

5 Current progress and future work

Currently, both melody and harmony generation have been implemented. The obtained results are good enough given the small amount of training data. The currently implemented model has no notion of tempo, and generates melodies of a fixed length (16 measures, assuming a tempo of 120bpm and a time signature of 4/4).
The separate LSTM for melody seed sampling is currently being tuned. Since poor results were obtained when training solely on the beginning of melodies, the model will be trained on the entirety of the melodies. The next step in this research will be to implement the notion of tempo. Work has already begun in that regard. This involves the implementation of an additional neural model that predicts tempo for a melody.
Furthermore, it will be necessary to obtain more training data, possibly through other forms of data augmentation, or through finding a suitable dataset.
Finally, once more data has been obtained, proper evaluation of the final output will be done, likely in the form of a survey.

References

[1] Thomson, Virgil (1957). "Introduction" to Robert Erickson. The Structure of Music: A Listener's Guide: A Study of Music in Terms of Melody and Counterpoint. New York: Noonday Press.

[2] T. Jiang, Q. Xiao and X. Yin, "Music Generation Using Bidirectional Recurrent Network," 2019 IEEE 2nd International Conference on Electronics Technology (ICET)

[3] Chung, Junyoung; Gulcehre, Caglar; Cho, KyungHyun; Bengio, Yoshua (2014). "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling". arXiv:1412.3555