AUTOMATIC GENERATION OF MONOPHONIC MELODIES WITH COMPLEXITY CONSTRAINTS USING DEEP NEURAL NETWORKS

Hany Tawfik, Michael Taenzer, Estefanía Cano and Jakob Abeßer
Fraunhofer IDMT, Germany
jakob.abesser@idmt.fraunhofer.de

ABSTRACT

Music students and amateur musicians benefit from having sheet music and practice content that matches their own level of expertise. This work explores the use of deep neural networks (DNNs) to automatically generate monophonic melodies of different difficulties. By introducing complexity constraints in the training data, we aim at generating practice material that matches the expertise of musicians in training. Two generative deep neural network architectures were trained on five subsets of monophonic melodies classified based on a rhythmic complexity measure. Results show a statistically significant correlation between the complexity of the melodies in the training data and the complexity of the generated melodies.

1. INTRODUCTION

Musicians with different levels of expertise benefit from sheet music exercises that help them improve their reading and playing skills. These exercises should match and gradually increase the level of proficiency in the performer. The goal of this study is to assert whether state-of-the-art systems for automatic melody generation can be effectively used to generate practice content following certain complexity constraints.

Early research in automatic music generation focused on generating musical sequences using Markov chains limited to short-term dependencies in the training data [3, 5]. More recently, Recurrent Neural Networks (RNNs) [1, 4], such as the one used in the DeepBach system, have become the main approach for automatic melody generation [2].

We focus on two generative network architectures for monophonic melody generation: The CharRNN model, which incorporates three LSTM layers, and processes melodies in ABC notation [1] [6], and the MelodyRNN model proposed by the Magenta research project [2], which generates melodies in MIDI format.

[1] http://abcnotation.com/wiki/abc:standard:v2.1
[2] https://github.com/tensorflow/magenta

© Hany Tawfik, Michael Taenzer, Estefanía Cano, Jakob Abeßer. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Hany Tawfik, Michael Taenzer, Estefanía Cano, Jakob Abeßer. "Automatic Generation of Monophonic Melodies with Complexity Constraints using Deep Neural Networks", 19th International Society for Music Information Retrieval Conference, Paris, France, 2018.

CharRNN is a character-based model consisting of three hidden layers with 512 LSTM blocks each. We use the RMSprop algorithm with batches of 500 samples to reduce training time. The model is trained for 20 epochs. The MelodyRNN architecture uses two LSTM-layers with 128 nodes each. We use its *Attention*-method, which gives weights to events happening in previous time steps. The model thereby decides the degree of attention it pays to past events, enabling it to learn longer-term structures.

We focus on generating monophonic melodies in the style of traditional Irish folk music. Training data stems from the dataset used in [6], originally collected from a weekly repository [3] . Moreover, we use an academic standard from the Associated Board of the Royal Schools of Music (ABRSM) to determine musical complexity [4] .

2. PROPOSED METHOD

The main processing steps of the proposed method are summarized in Fig. 1. First, features are extracted from a set of melodies, which are used to classify the data set into five subsets of gradually increasing levels of complexity. Each of the neural networks is then trained on each of these subsets. A total of 1000 melodies per complexity level were generated with each model. The generated melodies were then compared to the set of melodies used for training. Our main hypothesis is that by training a model with melodies of higher complexity, the model will also generate melodies with higher complexity.

Initially, the melodies of the input data set were classified according to the ABRSM standard. However, by strictly enforcing ABRSM rules, melodies that contained a single 16th note, for example, were immediately graded in a higher level (regardless of the complexity of the remaining content in the melody). As a solution, we introduce a penalty strategy for each note: Time attributes in the MIDI file are represented by incrementing scalars, from long duration to short ones (see Table 1). This allows us to compute an average rhythmic complexity value Γ_r of a melody, which in turn results in smoother classification and a more equal distribution over the complexity levels. As can be seen in Fig. 2-bottom, we use the cumulative sum over the histogram over Γ_r to ensure the five subsets have approx.

[3] https://thesession.org
[4] We used the outlines of the violin standard. Find details at https://de.abrsm.org/en/our-exams/bowed-strings-exams/violin-exams.

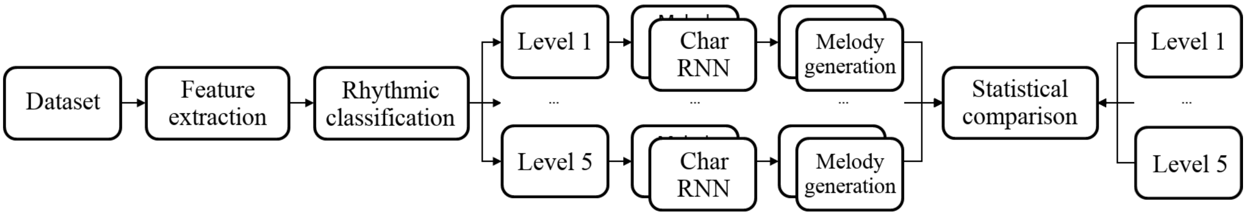

Figure 1. Block diagram of our system

Complexity	*Initial indicators*
Level 1	half (1), quarter (1), 8th (1)
Level 2	whole (2), dotted half (2)
Level 3	dotted whole (3), dotted quarter (3), dotted 8th (3), 16th (3)
Level 4	song is in 6/8th time signature
Level 5	dotted 16th (10), dotted 32nd (10), dotted 64th (10), 32nd (10), 64th (10), triplets (6)

Table 1. Complexity levels as specified by ABRSM. Each note duration is assigned a scalar penalty (in brackets).

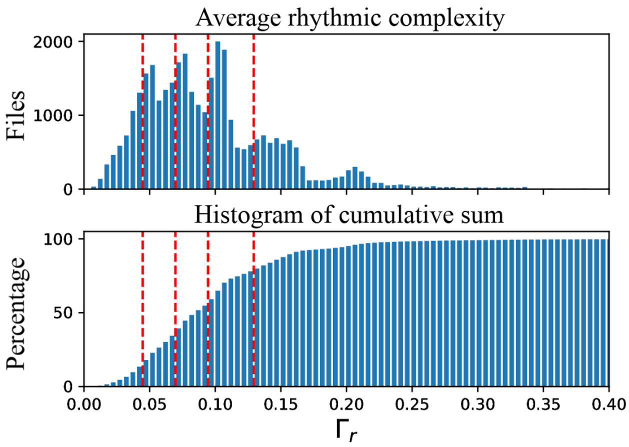

Figure 2. Average rhythmic complexity Γ_r for all training files (top) and cumulative sum (bottom). The red dashed lines separate the data from each level of complexity.

equal size. In this study, the feature extraction step is limited to rhythmic features.

3. EVALUATION & RESULTS

To test our hypothesis, we train CharRNN and MelodyRNN on each of the data subsets. Each subset is then represented by the average rhythmic complexity Γ_r over its melodies. A correlation analysis between the average complexity of the training data subsets and the subsets of the generated melodies shows a significant correlation for the CharRNN model (see Table 2). For MelodyRNN, a high (0.8035), but not significant correlation was found. These results suggest that by constraining the complexity of the training data, existing generative neural networks for melody generation can potentially be used to generate practice content of different difficulties. Other constraints related to harmonic properties and pitch intervals still need to be evaluated.

	Pearson correlation coeff. r	p
CharRNN	0.9844	0.0023
MelodyRNN	0.8035	0.1013

Table 2. Performance comparison of both models between original and generated datasets. A significant correlation ($p < 0.05$) was found for the CharRNN architecture.

4. ACKNOWLEDGEMENTS

This work has been supported by the German Research Foundation (AB 675/2-1).

5. REFERENCES

[1] J.-P. Briot, G. Hadjeres, and F. Pachet. Deep learning techniques for music generation - a survey. *arXiv preprint arXiv:1709.01620*, 2017.

[2] G. Hadjeres, F. Pachet, and F. Nielsen. Deepbach: A steerable model for bach chorales generation. In *34th International Conference on Machine Learning*, Sydney, Australia, 2017.

[3] L. A. Hiller Jr. and L. M. Isaacson. *Musical Composition with a High Speed Digital Computer*. University of Illinois, Urbana, IL, USA, 1957.

[4] N. Jaques, S. Gu, R. E. Turner, and D. Eck. Tuning recurrent neural networks with reinforcement learning. In *5th International Conference on Learning Representations*, Toulon, France, 2017.

[5] B. Manaris, D. Hughes, and Y. Vassilandonakis. Monterey mirror: Combining markov models, genetic algorithms, and power laws. an experiment in interactive evolutionary music performance. In *Proceedings of IEEE Congress on Evolutionary Computation*, pages 33–40, New Orleans, Louisiana, USA, 2011.

[6] B. L. Sturm, J. F. Santos, O. Ben-Tal, and I. Korshunova. Music transcription modeling and composition using deep learning. In *1st Conference on Computer Simulation of Musical Creativity*, University of Huddersfield, UK, 2016.