Algorithmic composition of polyphonic music with the WaveCRF

Umut Güçlü*, Yağmur Güçlütürk*, Luca Ambrogioni, Eric Maris, Rob van Lier, Marcel van Gerven, Radboud University, Donders Institute for Brain, Cognition and Behaviour, Nijmegen, the Netherlands
{u.guclu, y.gucluturk}@donders.ru.nl
*Equal contribution

Abstract

Here, we propose a new approach for modeling conditional probability distributions of polyphonic music by combining WaveNET and CRF-RNN variants, and show that this approach beats LSTM and WaveNET baselines that do not take into account the statistical dependencies between simultaneous notes.

1 Introduction

The history of algorithmically generated music dates back to the musikalisches würfelspiel (musical dice game) of the eighteenth century. These early methods share with modern machine learning techniques the emphasis on stochasticity as a crucial ingredient for “machine creativity”. In recent years, artificial neural networks dominated the creative music generation literature [3]. In several of the most successful architectures, such as WaveNet, the output of the network is an array that specifies the probabilities of each note given the already produced composition [3]. This approach tends to be infeasible in the case of polyphonic music because the network would need to output a probability value for every possible simultaneous combination of notes. The naive strategy of producing each note independently produces less realistic compositions as it ignores the statistical dependencies between simultaneous notes. Earlier work has tackled this problem by using RNN-RBMs on high-dimensional piano rolls [2] or language models on one-dimensional note events (BachBot[1] Polyphony RNN[2]). In this paper, we model the dependencies between notes as a conditional random field (CRF), and we use our new WaveCRF architecture to approximate the joint probabilities using a factorized mean field approach.

2 Methods

The WaveCRF is a combination of WaveNet [9] and CRF-RNN [10] variants, which models conditional probability distributions of simultaneous notes given their history. The WaveNet component learns to output both the unary potential and the densely connected pairwise kernels of the CRF for simultaneous notes at current time points as a function of those at previous time points. The CRF-RNN component learns to output the mean field approximation of the Gibbs distribution of the CRF [7] for simultaneous notes at current time points as a function of outputs of the WaveNet component. Since the mean field approximation of the Gibbs distribution of the CRF factorizes as a product of conditional probability distributions over simultaneous notes, they can be sampled independently from one another.

[1] https://github.com/feynmanliang/bachbot

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.
The idea of combining convnets with CRF-RNNs for modeling high-dimensional conditional probability distributions has already been proposed in the context of semantic segmentation [10]. While these proposals made it possible to learn unary potentials of CRFs, they relied on either densely connected but fixed [10] or learnable but sparsely connected [4] pairwise kernels to work around the limitation of prohibitively large number of potential pairwise connections. The WaveCRF takes this idea to the context of sequence modeling and formulates it such that it relies on neither fixed nor sparsely connected pairwise kernels as well as learning all of the terms end-to-end.

3 Results

Everything was implemented with Chainer[^1] and Cupy[^2] except for preprocessing, which was implemented with Magenta[^3]. Our implementation will be made available at https://github.com/umuguc/WaveCRF[^4].

We evaluated the WaveCRF on the music21[^5] Bach corpus that comprises 404 Bach chorales in digital sheet music format. We preprocessed the corpus by splitting time changes, quantizing (to sixteenth note), transposing (to ± major third) and extracting polyphonic tracks (between five and 32 bars) and encoding as piano rolls. We randomly assigned 90% of the corpus to the training set and the remaining data to the test set.

The specific WaveCRF that was evaluated comprised a WaveNet component with nine layers and a CRF-RNN component with one layer, which made five training and 10 test iterations to update the mean field approximation of the Gibbs distribution of the CRF. The WaveCRF was trained for predicting simultaneous notes at current time points given preceding five bars by minimizing the softmax cross entropy loss function with Adam [6].

We also evaluated two baselines that did not model the statistical dependencies between simultaneous notes. The first baseline comprised only the unary potentials of the WaveNet component of the WaveCRF. The second baseline comprised an LSTM [5] with four layers. The baselines were trained and tested similarly to the WaveCRF.

Table 1 shows the quantitative results (i.e., accuracy and loss on the test set) of the WaveCRF and the baselines. In comparison to the baselines, the WaveCRF had significantly higher accuracy and lower loss on the test set (p < 0.05, Student’s t-test). The qualitative results of the WaveCRF (i.e., example compositions in both digital sheet music format and MIDI format) will be made available at https://github.com/umuguc/WaveCRF[^4].

Table 1: Quantitative results of the WaveCRF and the baselines. Accuracy is defined as expected frame-level accuracy [1].

<table>
<thead>
<tr>
<th></th>
<th>accuracy</th>
<th>loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM</td>
<td>0.361</td>
<td>0.061</td>
</tr>
<tr>
<td>WaveNet</td>
<td>0.727</td>
<td>0.028</td>
</tr>
<tr>
<td>WaveCRF</td>
<td>0.749</td>
<td>0.026</td>
</tr>
</tbody>
</table>

4 Conclusion

In summary, we proposed the WaveCRF that combines WaveNet and CRF-RNN variants for modeling conditional probability distributions of simultaneous notes given their history. The WaveCRF achieved promising results, which warrant further experiments. In the future, we plan to use additional baselines, encodings and datasets.

[^1]: https://chainer.org
[^2]: https://cupy.org
[^3]: https://github.com/tensorflow/magenta
[^4]: https://github.com/cuthbertLab/music21
[^5]: https://github.com/cuthbertLab/music21

Acknowledgments

This work has been partially supported by a VIDI grant (639.072.513) from the Netherlands Organization for Scientific Research and a GPU grant (GeForce Titan X) from the Nvidia Corporation.

References


