FOLK MUSIC STYLE MODELLING BY RECURRENT NEURAL NETWORKS WITH LONG SHORT TERM MEMORY UNITS

Bob L. Sturm
Centre for Digital Music, QMUL
b.sturm@qmul.ac.uk

João Felipe Santos
INRS-EMT, Montreal Canada
jfsantos@emt.inrs.ca

Iryna Korshunova
Ghent University, Belgium
iryna.korshunova@ugent.be

ABSTRACT

We demonstrate two generative models created by training a recurrent neural network (RNN) with three hidden layers of long short-term memory (LSTM) units. This extends past work in numerous directions, including training deeper models with nearly 24,000 high-level transcriptions of folk tunes. We discuss our on-going work.

1. INTRODUCTION

The application of artificial neural networks to modelling and generating music is well-studied, e.g., [1–7]. Todd et al. [7] train an RNN with one hidden layer of 8-15 units to reproduce melodies of length 34 notes with quantised time steps. Mozer [6] builds a similar model for melody, but uses an encoding of pitch and time that is pyschoacoustically motivated. Eck and Schmidhuber [3] use LSTM to model larger structures than possible with an RNN, and train them on chord progressions and melodies exhibiting 12-bar blues conventions. Eck and Lapamle [2] expands upon this work to model folk tunes, and uses one-hot encoded input vectors to represent pitches at uniform time delays. Franklin [4] uses a coding scheme similar to [6] to build an RNN with LSTM for pitch and duration, but modelling "jazz-related tasks."

All of the work above entails network architectures that are shallow (e.g., one hidden layer of a number of units of the order 10), and training with a number of music transcriptions that are of the orders 10-100, most with fixed length. We new look at deeper recurrent architectures (3 hidden layers of 512 LSTM units each), and training on tens of thousands of textual transcriptions of music.

2. DATA

We use data retrieved from The Session, [1] an online community of folk music enthusiasts discussing relevant topics, and contributing transcriptions of tunes in ABC format. An example transcription is below:

```
X: 1
T: Scottish Horse, The
Z: Weejie
S: https://thesession.org/tunes/12696#setting21449
R: jig
M: 6/8
L: 1/8
K: Amix
A|c3 ecA|d>ef a2 f|edc cAe|B3 B<BA|
c3 ecA|d>ef afd|ecA B2e|A3 A<A:|
|:e|a2 e A<Ae|d2 f a2 f|e>dc cAe|B3 B<Be/d/|
[1 ceA eaA|faA dfA|ceA B2 e|A3 A<A:|
[2 c3 ecA|d>ef afd|ecA B2 e|A3 A<A||
```

We retrieve 23,958 tunes from this resource from their weekly compilations. [2] Eck and Lapamle [2] also use transcriptions from this resource in 2008, but only 56 reels converted to MIDI format. We create two different datasets: A) removes from the original data all the following ABC fields: `X:`, `Z:`, `S:`, and `R:`; B) as A, but further removes the ABC fields `T:` and `L:`, removes tunes with different meters and/or keys and multiple voices, removes ornaments and gracenotes, transposes all tunes to have the tonic C, (thus giving four modes: major, mixolydian, dorian, and minor), expresses each transcription as a sequence of tokens (separating pitches, duration, measure bars), and removes entries that are not complete transcriptions but instead are comments and suggestions, e.g., alternative endings. The number of transcriptions in B is 23,636.

3. MODELLING AND GENERATION

We use two approaches to modelling and generating ABC. The architecture of our models involves 3 hidden layers with 512 LSTM units each, and a softmax output layer given the distribution over the vocabulary conditioned on the one-hot encoded input. We train each model by backpropogation with one-hot encoded vectors, a mini-batch approach (batch size 50), and with drop out of 0.5. We train our first model using dataset A, with sequences of 50 characters. [3] In the above ABC, examples of characters are "`M`", ">" and "`:`". The size of the vocabulary is 134. Training makes 100 full passes through the dataset. As an example, a good model should always predict "`:`" given an input of "`M`" or "`K`", but never given ">". We train our second model using dataset B with a vocabulary of tokens. Examples of tokens are, "`d>`", "`K:Amix`", and "`|[1`", but not "`/`" or "`:`". Unlike the model above, we train this model using complete tunes: variable length sequences ranging

[1] https://thesession.org/

[2] https://github.com/adactio/TheSession-data

[3] https://github.com/karpathy/char-rnn

Attribution: Bob L. Sturm, João Felipe Santos, Iryna Korshunova. "Folk Music Style Modelling by Recurrent Neural Networks with Long Short Term Memory Units", Extended abstracts for the Late-Breaking Demo Session of the 16th International Society for Music Information Retrieval Conference, 2015.

```
T: Bornity Horse
M: 4/4
L: 1/8
K: Dmaj
|:FG A2 BGBd|AFGE DEdB|AF  A2  BGBd|(3cBA cd eAFG|
FA  B2 ABde|fedB AF F2|BG G2 ABdf|1 afea fdd2:|2 afea ~d3z||
|:d3e fd d2|Bd d2 Adfd|effe dff2|afeg fd d2 |
defd ed B2|ABde faaf|a2 fd Bd d2|AFde fdeg||
```

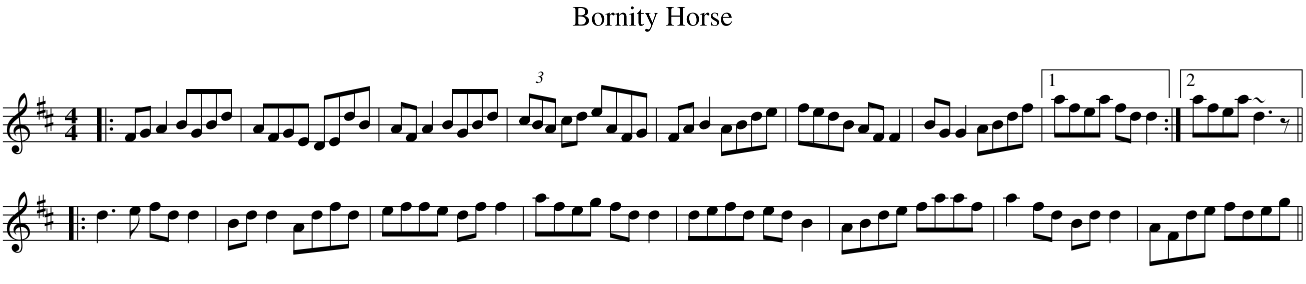

Figure 1. Verbatim output of the model trained on dataset A, and its typeset notation. The model has also titled the work.

```
M:4/4
K:Cmaj
|: G 2 E > C G > C E > C | G 2 E > G c > G E > C | D 2 D > F (3 D E D [ B, C ] > E | C 2 (3 E C C B > F D > A |
G 2 E > C E > C E > C | G 2 E 2 G > C E > C | D 2 d 2 G > B d > B |1 c 4 c 2 (3 G A B :| |2 c 2 B > A C 3 G /2 A /2
|: B > c d > e f > d B > c | d > e d > e c > A (3 G A B | c 2 e > c c 4 | e > a (3 e e e c 2 (3 G A B |
c 2 c > e f > e d > c | _B 2 B 2 A < B d < c | B > G F > D D > _B B < d |1 c 2 B 2 c > A G < B :| |2 c 2 B 2 c 2 c 2 |
```

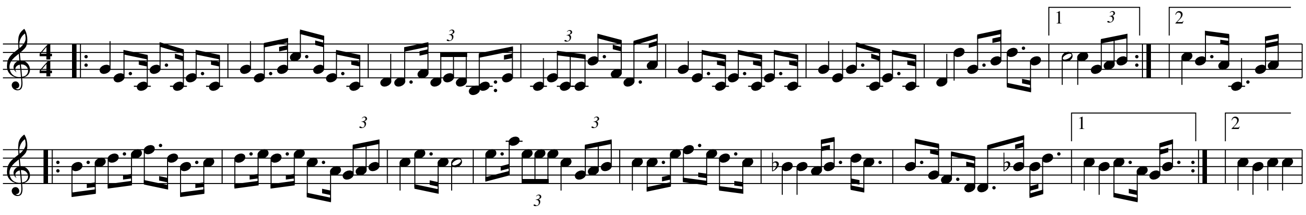

Figure 2. Verbatim output of the model trained on dataset B, and its typeset notation. Tokens are visible.

from about 50 to 2000 tokens depending on the tune. [4] We do not split the tunes into unrelated parts, and do not mix parts from different tunes. This preserves the structure of the data. Training makes 100 passes through the dataset. The size of the vocabulary is 137. (with <s> and </s> to mark the start and the end of each tune).

To generate ABC from the first model we merely prime it with input text, e.g., "M: 4/4", and sample any number of times from the distribution at the softmax output. Each generated character produces the next input to the model. For the second model, we prime it with a start symbol <s> and sequentially sample tokens from the softmax outputs until we encounter the end symbol </s>.

4. RESULTS AND FUTURE WORK

Figures 1 and 2 show example output of the two models. Both models have learned some of the structural rules of the transcriptions in the dataset, as well as stylistic conventions: 1) each measure in these examples is correctly counted (but this is not always the case); 2) a transcription can involve two repeated 8-measure sections; 3) a section can have a second ending that varies the first; 4) a section can consist of 4-measure subsections delimited by a varied introduction figure (see m. 1 and 4, 9 and 13 in Fig. 1); 5) a section ends on the tonic; 6) the dominant appears a measure before a section ends. Regardless, these transcriptions are not immediately "session-ready" tunes. Our current work explores refining such models by incorporating corrections made by a domain expert to the output.

5. REFERENCES

[1] D. Correa, J. Saito, and S. Abib. Composing music with BPTT and LSTM networks: Comparing learning and generalization aspects. In *Proc. Int. Conf. Comp. Sci. Eng. Workshops*, pages 95–100, 2008.

[2] D. Eck and J. Lapamle. Learning musical structure directly from sequences of music. Technical report, University of Montreal, 2008.

[3] D. Eck and J. Schmidhuber. Learning the long-term structure of the blues. In *Proc. Int. Conf. on Artificial Neural Networks*, 2002.

[4] Judy A. Franklin. Recurrent neural networks for music computation. *ORSA journal on computing*, 18(3):321–338, 2006.

[5] Niall Griffith and Peter M Todd. *Musical networks: Parallel distributed perception and performance*. MIT Press, 1999. In *Proc. Artificial Intell. Interactive Digital Entertainment Conf.*, 2014.

[6] M. C. Mozer. Neural network composition by prediction: Exploring the benefits of psychophysical constraints and multiscale processing. *Cog. Science*, 6:(247-280), 1994.

[7] Peter M. Todd. A connectionist approach to algorithmic composition. *Computer Music Journal*, 13(4):pp. 27–43, 1989.

[4] https://github.com/IraKorshunova/folk-rnn