GENERATING AND DISCRIMINATING SYMBOLIC MUSIC CONTINUATIONS WITH BACHPROP

Florian Colombo
Brain-Mind Institute, School of Life Sciences
École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland

ABSTRACT

The algorithm described here was submitted to the 2018 MIREX "Patterns for Prediction" task. The same algorithm was applied to generate a possible continuation of monophonic and polyphonic music primes and to discriminate between possible continuations. To address these tasks, we used BachProp, an algorithm based on a recurrent neural network model, which aim is to model the note transition probabilities.

1. INTRODUCTION

To address the problem of pattern discovery in musical sequence, we trained our algorithm entitled *BachProp* to predict as good as possible each upcoming note in music examples provided in a training corpus. Therefore, rather than explicitly finding patterns in musical sequences, we trained a deep artificial neural network to implicitly model these patterns in the form of note transition probabilities. The first task for which we adapted BachProp consists in suggesting a possible continuation to an original truncated music piece. To address this task, BachProp is set to its generative mode in order to suggest possible continuations. The second task requires BachProp to discriminate between two possible continuations, i.e. retrieve the original continuation among the two possibilities. To achieve that, we evaluate the two possible continuations using our model and classify as original the continuation that is most probable for BachProp.

2. METHODS

2.1 Symbolic Music Representation

In written music, the n^{th} note $\mathbf{note}[n]$ of a piece of music $\mathbf{song} = (\mathbf{note}[1], \ldots, \mathbf{note}[N])$ can be characterized by its pitch $P[n]$, duration $T[n]$ and the time-shift $dT[n]$ of its onset relative to the previous note, i.e. $\mathbf{note}[n] = (dT[n], T[n], P[n])$. The time-shift $dT[n]$ is zero for notes played at the same time as the previous note. In contrast to most other approaches that discretize the rhythm into multiples of a base unit, we round all durations into a set of common musical durations which allows a more faithful representation of timing that is limited only by the number of possible values considered for $T[n]$ and $dT[n]$. For example, our representation allows to easily and without any distortion represent 32nd notes, triplets and dotted notes in the same dataset. As well as any other more complex note durations that can be needed for specific corpora.

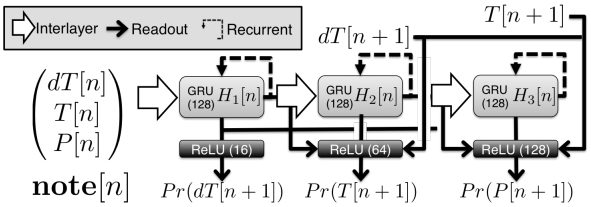

Figure 1. **BachProp neural architecture.** See text for details.

Our approach is to approximate probability distributions over note sequences in music scores $\mathbf{song}_1, \ldots, \mathbf{song}_S$ with distributions parameterized by recurrent neural networks and move its weights θ towards the maximum likelihood estimate

$$\theta^* = \arg\max_{\theta} Pr(\mathbf{song}_1, \ldots, \mathbf{song}_S|\theta)\,, \qquad (1)$$

Since each note in each song consists of the triplet $(dT[n], T[n], P[n])$ we can parametrize the distributions by further conditioning the note features on each others (e.g. condition the pitch prediction on the actual note timing and duration). Importantly, our model takes into account that pitch and duration of a note are generally not independent. For example in classical music, the fundamental, e.g. the note C in a piece written in C major, tends to be longer than other notes.

2.2 The BachProp neural network

We used a deep GRU [1] network with three consecutive layers as schematized in Figure 1. The network's task is to infer the probability distribution over the next possible notes from the representation of the current note and the network's internal state (the network representation of the history of notes).

The probability of a sequence of N notes $\mathbf{note}[1 :$

$N] = (\mathbf{note}[1], \ldots, \mathbf{note}[N])$ is given by

$$Pr(\mathbf{note}[1:N]) = Pr(\mathbf{note}[1]) \prod_{n=1}^{N-1} Pr(\mathbf{note}[n+1]|\mathbf{note}[1:n]) . \quad (2)$$

Each term on the right hand side can be further split into

$$\begin{aligned} Pr(\mathbf{note}[n+1]|\mathbf{note}[1:n]) = \\ Pr(dT[n+1]|\mathbf{note}[1:n])\times \\ Pr(T[n+1]|\mathbf{note}[1:n], dT[n+1])\times \\ Pr(P[n+1]|\mathbf{note}[1:n], dT[n+1], T[n+1]) . \end{aligned} \quad (3)$$

The goal of training the Bachprop network with parameters θ is to approximate the conditional probability distributions on the right hand side of 3.

In the BachProp network (Figure 1), the conditioning on the history $\mathbf{note}[1:n]$ (see Equation 3) is implemented by the values of the shared hidden states. The hidden state is composed of 3 recurrent layers with 128 gated-recurrent units (GRU). The state $H_1[n]$ of the first hidden layer is updated with input $\mathbf{note}[n]$ and previous state $H_1[n-1]$. The state of the upper layers $H_i[n]$ for $i = 2, 3$ is updated with input $H_{i-1}[n]$ and $H_i[n-1]$. To generate $\mathbf{note}[n+1]$, one third ($H_1[n]$ in 1) of the full hidden state is fed into a feedforward network with one layer of 16 Relu units and one output softmax-layer that represents $Pr(dT[n+1]|H_1[n]) \approx Pr(dT[n+1]|\mathbf{note}[1:n])$. The chosen $dT[n+1]$ together with $H_1[n]$ and $H_2[n]$ is fed into a second feedforward network with one layer of 64 Relu units and an output softmax-layer that represents $Pr(T[n+1]|H_1[n], H_2[n], dT[n+1]) \approx Pr(T[n+1]|\mathbf{note}[1:n], dT[n+1])$. In a similarly way, the pitch is sampled from $Pr(P[n+1]|H_1[n], H_2[n], H_3[n], dT[n+1], T[n+1]) \approx Pr(T[n+1]|\mathbf{note}[1:n], dT[n+1], T[n+1])$. These three small steps of sampling $dT[n+1], T[n+1]$ and $P[n+1]$ form together one big step from note n to note $n+1$.

2.3 Training

BachProp was trained with truncated back propagation through time, with a truncation at 128 notes, the Adam optimizer [2] and cross-entropy loss function. The mini-batch size is 32 scores, the validation set a 0.1 fraction of the transposed-augmented original corpus, and one training epoch consists of updating the network parameters with all training examples and evaluating the performances on the entire validation set. Training is stopped when the performances on the validation set saturates. The model leading to the highest validation accuracy is used for solving both tasks.

The training set was provided by the organizers of the 2018 MIREX "Patterns for Prediction" competition. It consists of a processed subset of the LAKH dataset [3]. To train BachProp, we presented sequentially every notes of each piece in the training data. We trained two BachProp models, one on the monophonic and one on the polyphonic version of the original dataset. Because the training corpus is very heterogeneous, we added input skip connections towards each hidden layer (see Figure 1) to improve predicting performances.

2.4 MIREX Tasks

To generate a possible continuation, the network is seeded with the sequence of note present in the prime. We then sample the output probabilities of the neural network to construct 32 possible continuations (10 beats) and finally select the one with the highest overall probability.

The second task consisting in finding the real continuation is addressed by comparing the likelihood of BachProp given the prime and the possible continuations. The continuation that is associated with the highest likelihood is then selected as true. Furthermore, we added a penalty term that is proportional to the rounding error of our representation (see Section 2.1).

3. RESULTS

For results, we refer the reader to the respective 2018 MIREX "Patterns for Prediction" result page.

4. DISCUSSION

We achieved good performances in the discrimination task with around 90% of the possible continuations classified correctly. These results are way above chance level but unfortunately cannot be compared with other approaches as no other were submitted.

The results on the continuation generation task are however not as convincing. A reason for that is that BachProp performances depend on the heterogeneity of the training data. Indeed, our model is designed to extract the temporal relationship between notes in their context. A task that can be associated with finding the invariants, or rules, in the observed musical structures. While BachProp performs well on datasets with relatively homogeneous structures (e.g. pieces from a single composer only), the provided dataset include too many different music styles so that our model struggles to find any common structures within these data. This observation opens the discussion into how can we designed sharper neural network models that can scope with high heterogeneity in the training corpus.

5. REFERENCES

[1] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[2] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[3] Colin Raffel. *Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching*. Columbia University, 2016.