Polyphonic Music Modelling with LSTM-RTRBM

Qi Lyu
qilyu.pub@gmail.com

Zhiyong Wu
zywu@se.cuhk.edu.hk

Jun Zhu
dcszj@tsinghua.edu.cn

Center for Bio-inspired Computing Research
State Key Lab of Intelligent Technology & Systems
Tsinghua National Lab for Information Science & Technology
Department of Computer Science & Technology, Tsinghua Univ., Beijing 100084, China

ABSTRACT

Recent interest in music information retrieval and related technologies is exploding. However, very few of the existing techniques take advantage of the recent advancements in neural networks. The challenges of developing effective browsing, searching and organization techniques for the growing bodies of music collections call for more powerful statistical models. In this paper, we present LSTM-RTRBM, a new neural network model for the problem of creating accurate yet flexible models of polyphonic music. Our model integrates the ability of Long Short-Term Memory (LSTM) in memorizing and retrieving useful history information, together with the advantage of Restricted Boltzmann Machine (RBM) in high dimensional data modelling. Our approach greatly improves the performance of polyphonic music sequence modelling, achieving the state-of-the-art results on multiple datasets.

Categories and Subject Descriptors

H.5.5 [**Sound and Music Computing**]: Modelling

General Terms

Algorithm

1. INTRODUCTION

Music is among the most widely consumed types of signal streams. Models for finding, extracting and reproducing musical temporal structure are of considerable interest. In particular, generative models for composing (good) music might have not only artistic value but also commercial potential. Besides that, in the same way that natural language models tremendously improve the performance of speech recognition systems, *musical language model* can also improve audio music recognition, i.e., the transcription of raw audio into symbolic notations [13]. Moreover, ad hoc music retrieval could be possible, e.g., by building a model for every piece of music in the collection and then ranking each piece according to the probability that model produces when a query is provided.

A generative theory of music can be constructed by explicitly coding music rules in some logic or formal grammar. This approach is sometimes called an expert or knowledge engineering system. Although these methods could achieve impressive results in some cases, they require extensive exploitation of musical knowledge, often specific to each composer or style. Another approach relies on statistical learning or empirical induction, such as hidden Markov models (HMMs) [1], Markov random fields (MRFs) [10], etc. These Markov models are statistical models of random sequences with the typical assumption that the probability for generating the next symbol depends only on a limited past. It is often hard to choose the order of Markov models because a small fixed order will certainly limit the representation ability, while a large one will result in a large number of parameters that are difficult to estimate, with no guarantee of better performance.

Non-symbolic approaches such as Recurrent Neural Networks (RNNs), which become popular with the recent breakthrough in deep learning, can also capture the knowledge of music. For example, the system in [5] could learn entire songs given a melody and the associated chord sequence, and the RNNs combined with restricted Boltzmann machines (RBMs) for feature representation [3] have showed the best results in music generation recently. However, the music composed by RNN-type models often suffers from a lack of global structure. Although networks can learn note-by-note transition probabilities and even reproduce phrases, the attempts to learn an entire musical form and to use that knowledge to guide composition have been unsuccessful. The reason for this failure seems to be that RNNs cannot keep track of temporally distant events that indicate global music structure. On the other hand, Long Short-term Memory (LSTM) [8] has succeeded in similar domains where other RNNs have failed, with state-of-the-art results in various sequence processing tasks, including speech recognition [7], handwriting recognition [6] and machine translation [17].

In this work, we present LSTM-RTRBM, a hybrid model for the notoriously tricky problem of capturing the long-term structure in polyphonic music. Our model embeds long-term memory into the

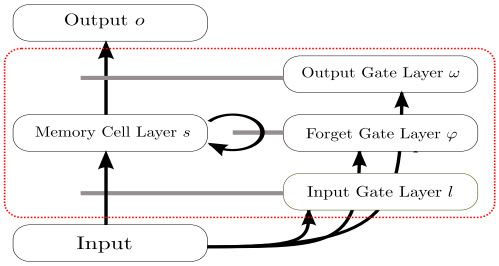

Figure 1: Layer-wise architecture of a LSTM block (peephole connections omitted) picture adapted from [11].

MM'15, October 26–30, 2015, Brisbane, Australia.
© 2015 ACM. ISBN 978-1-4503-3459-4/15/10 ...$15.00.
DOI: http://dx.doi.org/10.1145/2733373.2806383 .

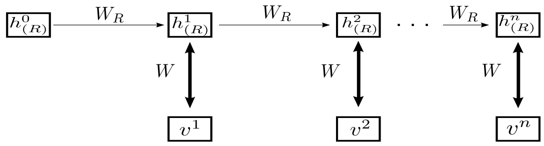

Figure 2: Structure of RTRBM.

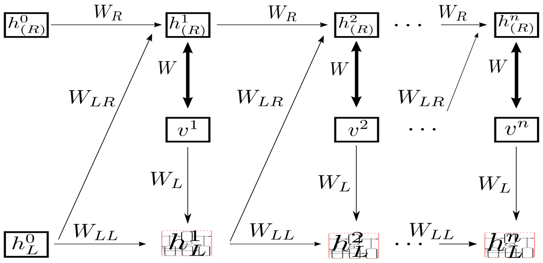

Figure 3: Structure of LSTM-RTRBM.

Recurrent Temporal RBM (RTRBM) [16], by increasing a bypassing channel from data source filtered by a recurrent LSTM layer. We show that our LSTM-RTRBM model increases performance and advances the state-of-the-art results even without additional optimizations used in the previous models.

The rest of the paper is organized as follows: In Section 2 we briefly introduce LSTM, RBM and RTRBM as background. In Section 3 we present our new model. We then validate our model on benchmark datasets in Section 4, and present our results on musical sequences in Section 5.

2. BACKGROUND

In this section, we briefly introduce the building blocks of our model. More details can be found in supplementary materials.

LSTM is a special kind of RNN that can store and retrieve information from input streams and do complicated tasks like classifying, processing and most importantly, predicting time series when there are very long time lags of an unknown size between important events. As illustrated in Fig. 1, LSTM is enabled to do so by utilizing memory cells that use logistic and linear units with multiplicative interactions with input and output gates. The information gets into (out of) the memory cell whenever a logistic input (output) gate is turned on. The memory cell state is kept from irrelevant information by keeping the input gate off, and its information stored can be later probed by turning the output gate on. Forget gates can reset the memory cell state for storage of new information, so that in continuous prediction the memory cells can keep the information when useful and forget the information when useless. With the extension of peephole connections, which are the direct connections from memory cell states to gates, the performance of LSTM is improved. The behavior of all these gates can be learned, rendering LSTM's ability in long term memory compared with simple recurrent networks without such kind of architecture.

The RBM is a typical kind of product-of-experts (PoE) model, which defines a probability distribution over the visible vector v (inputs) and the hidden vector h as:

$$P(v,h) = \exp(v^\top b_v + h^\top b_h + v^\top W h)/Z \tag{1}$$

where b_v, b_h are vectors of biases for the visible and hidden vectors respectively, W is the matrix of connection weights, Z is the usually intractable partition function that ensures $P(v,h)$ is a well-normalized probability distribution and $x^\top$ is the transpose of x while $v^T(h^T)$ is the entire sequence of $v^t(h^t)$ from start to time T.

The Recurrent Temporal RBM (RTRBM) [16] (See Fig. 2) is a sequence of conditional RBMs (one at each time step) whose parameters are time-dependent on the sequence history. Given previous hidden units h^{t-1} $(t > 1)$, the conditional distributions for current hidden units are factorized and takes the form:

$$P(h^t|v^t, h^{t-1}) = \sigma(W^\top v^t + W_R^\top h^{t-1} + b_h) \tag{2}$$

where $\sigma(x) = (1 + e^{-x})^{-1}$ is the element-wise logistic sigmoid function. To sample v^T from RTRBM, we can use the following:

$$\begin{aligned} P(v^t, h^t|h^{t-1}) = \exp((v^t)^\top b_v + (v^t)^\top W h^t + \\ (h^t)^\top (b_h + W_R h_R^{t-1}))/Z(h^{t-1}) \end{aligned} \tag{3}$$

The RTRBM is defined by its joint probability distribution:

$$P(v^T, h^T) = \prod_{t=2}^{T} P(v^t|h^{t-1}) P(h^t|v^t, h^{t-1}) \cdot P(v^1) P(h^1|v^1)$$

3. LSTM-RTRBM

Though previous models have merits in modelling different aspects of data, they cannot meet all the requirements on modelling music sequences. In this section, we present LSTM-RTRBM, a new model that conjoins the strengths of LSTM and RTRBM.

Specifically, the creation of music is a complicated mental activity and we would like our neural network model to mimic this process, especially from the information memorizing and retrieving perspective. Music feature extraction is a sort of perceptual categorization and grouping of music data. Music notes processed at this stage can activate those parts of long-term memory evoked by similar notes in the past. Activated long-term memory at this point forms a context for current awareness. Long-term memory of conceptual categories that reaches higher states of activation can then persist as current short-term memory while other *semi-activated* remains as context or degrades with time, and the information being circulated in short-term memory will cause modifications in permanent long-term memory [15]. For example, in order to complete a melody line, the beginning of the music sequence needs to be held in mind while the rest is played, a task which is carried out by the short-term memory. And the long-term memory will serve as the theme and emotion that will help maintain the global coherence of music. The existence of both the short-term and the long-term memory is vital for generating melodic and coherent music sequences.

Moreover, polyphonic music sequences are of high dimensions where simply predicting the expected value at the next time step for generating music sequences is not satisfying, since the conditional distribution is very often muti-modal. For example, it is obvious that the occurrence of a particular note at a particular time modifies considerably the probability with which other notes may occur at the same time. In other words, notes appear together in correlated patterns cannot be conveniently described by a normal RNN architecture designed for multi-class classification task, because enumerating all configurations of the variable to predict would be very expensive. This difficulty motivates energy-based models which allow us to express the log-likelihood of a given configuration by an arbitrary energy function, such as the RBM.

In this context, we wish to combine the ability of RBM to represent a complicated distribution for each time step, together with a temporal model in sequence. We consider both long-term memory and short-term memory in our design of guide and learning modules, by integrating LSTM with normal recurrent models, RTRBM in our case. Adding LSTM units to RTRBM is not trivial, considering RTRBM's hidden units and visible units are intertwined in inference and learning. The simplest way to circumvent this difficulty is to use bypass connections from LSTM units to the hidden units besides the existing recurrent connections of hidden units, as in LSTM-RTRBM (Fig. 3).

The recurrent hidden units ($h_{(R)}$s in Fig. 3) serves as the short-term memory and the LSTM units ($h_{(L)}$s) serves as the long-term memory. To sample v^T, we can simply use the following:

$$P(v^t, h^t|h^{t-1}) = \exp((v^t)^\top b_v + (v^t)^\top W h_t + (h^t)^\top (b_h + W_R h_R^{t-1} + W_L h_L^{t-1}))/Z(h^{t-1}) \quad (4)$$

By this means, there are two channels for temporal information flow, the direct connection (W_R) between the conditional RBM at each time step, and the connection (W_{LR}) from the recurrent LSTM units of previous time steps. The main computation complexity comes from the repeated sampling procedure of RBM learning and replacing some hidden units in RBM with LSTM units actually boosts learning speed. The inference and sampling procedure is roughly the same as in RTRBM while the Backpropagation Through Time (BPTT) procedure is a bit complex. Due to space limit, we defer the details to supplementary materials.

4. BASELINE EXPERIMENTS

Before presenting the music modelling experiments, we first compare the performance of our model with RTRBM on the human motion capture data experiment, a classic baseline experiment for evaluating sequence models. The human motion capture dataset[1] represents human motion by sequences of joint angles, translations, and rotations of the base of spine. This dataset is recorded on real persons and has more repeated patterns than normal music sequences, such as the half circular motion of arm and the alternative moving and static status of feet. Since the data consists of 49 real values per time step, we use the Gaussian RBM variant (see supplementary materials for details) for this task. The mean squared prediction error per dimension per time step is 0.32 for the RTRBM with 200 logistic hidden units and reduced to 0.13 by LSTM-RTRBM with 100 logistic hidden units and 100 LSTM units, and to 0.09 with twice as many units. It can be seen that with the inclusion of long-term information in LSTM-RTRBM, the prediction is significantly more accurate and stable.

5. MODELLING SEQUENCES OF POLYPHONIC MUSIC

In this section, we show results with the main application of interest: probabilistic modelling of sequences of polyphonic music. We conduct experiments on two music datasets of different styles: MuseData, an electronic library of orchestral and piano classical music from CCARH 4[2] and JSB chorales, the entire corpus of 382 four-part harmonized chorales by J. S. Bach with the split of [1].

Table 1: Log-likelihood (LL) and expected frame-level accuracy (ACC%) for various musical models in the generation task. The double line separates frame-level models (above) and models with a temporal component (below).

MODEL	MUSEDATA		JSB CHORALES	
	LL	ACC%	LL	ACC%
RANDOM	-61.00	3.74	-61.00	4.42
GMM	-12.20	7.37	-11.90	15.84
NADE	-10.06	7.65	-7.19	17.88
RBM	-9.56	8.19	-7.43	4.47
PREVIOUS + GAUSSIAN	-12.90	25.93	-19.00	18.36
GMM+HMM	-11.17	13.93	-11.89	19.24
MLP	-7.94	25.68	-8.70	30.41
RNN	-8.13	23.25	-8.71	28.46
RNN(HF)	-7.19	30.49	-8.58	29.41
LSTM	-6.88	30.15	-7.92	30.07
RNN-NADE	-6.74	24.91	-5.83	32.11
RTRBM	-6.35	30.85	-6.35	30.17
RNN-RBM	-6.01	**34.02**	-6.27	33.12
RNN-NADE(HF)	**-5.60**	32.60	-5.56	32.50
LSTM-RTRBM	**-5.54**	**33.89**	**-4.72**	**35.22**

Each dataset contains at least 7 hours of polyphonic music and the total duration is approximately 29 hours. The polyphony (number of simultaneous notes) varies from 0 to 15 and the average polyphony is 4.2. We use a completely general piano-roll representation with an input of 88 binary visible units that span the whole range of piano from A0 to C8 and temporally aligned on an integer fraction of the beat (quarter note). Consequently, pieces with different time signatures will not have their measures start at the same interval. Although it is not strictly necessary, learning is facilitated if the sequences are transposed in a common tonality (e.g. C major/minor) as preprocessing.

In addition to the models previously described, the results (taken from [3]) of the following commonly used methods are also listed for comprehensive comparison:

- The simplest baseline model is the Gaussian density model (PREVIOUS + GAUSSIAN) with the previous frame as mean value $u = v^{(t-1)}$ and learned covariance Σ.
- The neural autoregressive distribution estimator (NADE) [9] is a model inspired by the RBM, which decomposes the joint distribution of observations into tractable conditional distributions, and modelling each conditional using a non-linear function similar to a conditional of an RBM. Thus it is a tractable model and can be further optimized with its exact gradient.
- Other common methods include Gaussian mixture models (GMM), hidden Markov models (HMM) using GMM indices as their state, and multilayer perceptron (MLP) with the last n time steps as input.

We adopt the classic momentum training regime, with learning rate 0.01 and momentum 0.9. The learning starts with CD_{10} (10 steps of Contrastive Divergence) for the first 1000 weight updates, which then switches to CD_{25}. We use 88 hidden units and 88 LSTM units, the same number as the input and the output dimension, which is trained faster than using hundreds of hidden units in the RTRBM, for the main computation takes place in the CD steps (not needed if changed to LSTM units) . The partition function of conditional RBM at each time step is calculated with 100 run of annealed importance sampling [14] on GPU. The Log-likelihood and expected frame-level accuracy [2] are presented in Table 1. There are some key observations:

- It generally improves the performance of the model, to predict the parameters of the distribution of the data, i.e., the hid-

[1] people.csail.mit.edu/ehsu/work/sig05stf

[2] www.musedata.org

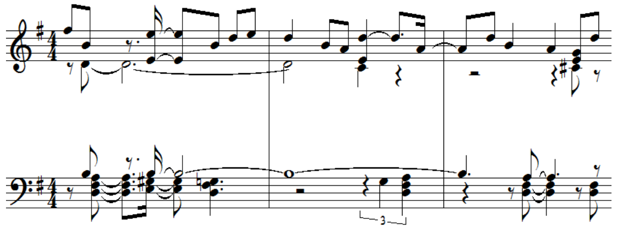

Figure 4: Slice of sample music generated by LSTM-RTRBM.

den units of RBM, rather than the raw data itself. However, LSTM or RBM alone only gives moderate performance; it is the combination of both the best recurrent model and the best static feature extraction model that gives the best performance.

- The combined RNN-NADE model with Hessian-free (HF) optimization [12] is a robust distribution estimator because the RNN part and the NADE part are optimized jointly, though frame-level NADEs are less powerful than RBMs. The use of HF optimization significantly helps the density estimation and prediction performance, although considerably increases the training time. On the other hand, the approximated gradient and iterative CD estimation process alienate models with RBM from optimization methods like HF.
- LSTM performs better than standard RNN with strong optimization method like HF generally, and the inclusion of LSTM units significantly improves the performance of the model (LSTM-RTRBM compared with RTRBM), achieving state-of-the-art result in these datasets. It can be conjectured that NADE combined with LSTM might give better result with the aid of HF optimization.

We also evaluate our models qualitatively by generating sample sequences (see Fig. 4 for a glimpse). The model has learned the chords (such as sequential D major triads in Fig. 4), local and global temporal coherence, melody lines and generate music that is harmonic and coherent. With the same configuration, LSTM-RTRBM could learn melody lines from both datasets while RTRBM generates inconsistent and unpleasant sample sequences. However, all the recurrent temporal model forms a closed loop that have no new incitations from outside, making the long piece of music dull. This can be solved with the technique of side-slipping [4], by playing out-of-key to produce a short sensation of surprise in a context deemed too predictable.

6. CONCLUSIONS

We investigate the problem of modelling long-term dependencies in high-dimensional polyphonic music sequences. We present a new neural network model for the problem of creating accurate yet flexible statistical models of polyphonic music, and conduct extensive experiments to evaluate the proposed model. Our model greatly improves the performance of polyphonic music modelling, achieving the state-of-the-art results on various datasets. For future work, we are interested in optimizing LSTM with hessian-free techniques for better results, and integrating side-slipping mechanism for more variable music generation.

7. ACKNOWLEDGEMENTS

Q.L and Z.W are supported by the National High Technology Research and Development Program of China (2015AA016305), the National NSF of China (61375027 and 61433018) and the Major Program for the National Social Science Foundation of China (13& ZD189). J.Z is supported by NSF of China Projects (61322308, 61332007) and Tsinghua Initiative Scientific Research Program (20121088071, 20141080934).

8. REFERENCES

[1] M. Allan and C. K. Williams. Harmonising chorales by probabilistic inference. *Advances in neural information processing systems*, 17:25–32, 2005.

[2] M. Bay, A. F. Ehmann, and J. S. Downie. Evaluation of multiple-f0 estimation and tracking systems. In *ISMIR*, pages 315–320, 2009.

[3] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent. Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. In *Proceedings of the 29th International Conference on Machine Learning*, pages 1159–1166, 2012.

[4] J. Coker. *The complete method for improvisation*. Studio P/R, 1980.

[5] J. A. Franklin. Recurrent neural networks for music computation. *INFORMS Journal on Computing*, 18(3):321–338, 2006.

[6] A. Graves. *Supervised sequence labelling with recurrent neural networks*, volume 385. Springer, 2012.

[7] A. Graves, A.-r. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *International Conference on Acoustics, Speech and Signal Processing*, pages 6645–6649. IEEE, 2013.

[8] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[9] H. Larochelle and I. Murray. The neural autoregressive distribution estimator. *Journal of Machine Learning Research*, 15:29–37, 2011.

[10] V. Lavrenko and J. Pickens. Polyphonic music modeling with random fields. In *Proceedings of the eleventh ACM international conference on Multimedia*, pages 120–129. ACM, 2003.

[11] Q. Lyu and J. Zhu. Revisit long short-term memory: An optimization perspective. *Advances in neural information processing systems workshop on deep Learning and representation Learning*, 2014.

[12] J. Martens. Deep learning via hessian-free optimization. In *Proceedings of the 27th International Conference on Machine Learning*, pages 735–742, 2010.

[13] J. Nam, J. Ngiam, H. Lee, and M. Slaney. A classification-based polyphonic piano transcription approach using learned feature representations. In *ISMIR*, pages 175–180, 2011.

[14] R. Salakhutdinov and I. Murray. On the quantitative analysis of deep belief networks. In *Proceedings of the 25th International Conference on Machine learning*, pages 872–879. ACM, 2008.

[15] B. Snyder. *Music and memory: An introduction*. MIT press, 2000.

[16] I. Sutskever, G. E. Hinton, and G. W. Taylor. The recurrent temporal restricted boltzmann machine. In *Advances in Neural Information Processing Systems*, pages 1601–1608, 2008.

[17] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112, 2014.