Recurrent Neural Network-based Music Language Models for Improving Automatic Music Transcription

Abstract

In this paper we investigate using Music Language Models (MLMs) to improve Automatic Music Transcription (AMT). The MLMs are trained on sequences of symbolic polyphonic music. We train RNN-based models as they can capture complex temporal dependencies present in the data. Similar to the function of language models in ASRs, we use the MLMs to generate a prior probability for the occurrence of a sequence. We incorporate the priors into the transcription framework using Dirichlet priors. We test our hybrid models on the BACH-10 dataset and report a 2.5% improvement in accuracy which is a significant result.

1. Introduction

Automatic Music Transcription (AMT) involves automatically generating a symbolic transcription of an acoustic musical signal. The transcription can be thought of as the digitized version of the musical score corresponding to the music signal. Typically, the output of an AMT systemis a *pianoroll* representation, which is a two-dimensional matrix representation of a musical piece where the X-axis represents time quantized into regular intervals, and the Y-axis represents the 88 keys of a piano in increasing pitch. A cell in this matrix is 1 if the key represented by its X-coordinate is sounded at the time instant represented by its Y-coordinate.

NOTE: Automatic music transcription literature review, PLCA based AMT research and state-of-the-art techniques could go here. Consider citing work by Benetos, Nam, etc.

There is no doubt that a reliable acoustic model is important for generating accurate symbolic transcriptions of a given music signal. However, since music exhibits a fair amount of structural regularity much like language, it is natural for one to think of the possibility of improving transcription accuracy using a *music language model* (MLM) in a manner akin to the use of a language model to improve the performance of a speech recognizer (Rabiner & Juang, 1993). In (Boulanger-Lewandowski et al., 2012), the predictions of a polyphonic MLM were used to this end. More generally, *score informed* approaches have been found to benefit the performance of purely acoustic models in music research tasks such as source separation (Ewert & Müller, 2012), voice separation (Ewert & Müller, 2011) and tonic identification (Sentürk et al., 2013).

In the present work, we make use of the predictions made by a Recurrent Neural Network-Neural Autoregressive Distribution Estimator (RNN-NADE) based polyphonic MLM proposed in (Boulanger-Lewandowski et al., 2012) to refine the transcriptions of a PLCA based AMT system (Benetos & Dixon, 2012; Benetos et al., 2013). *NOTE: Summary of the combination strategy using Dirichlet priors, etc. could go here*. It was observed that combining the two models in this way boosts transcription accuracy to 100.00% on the Bach-10 dataset, where the existing state-of-the-art accuracy is 99.00%.

2. Automatic Music Transcription System

For combining acoustic and music language information in an automatic transcription context, we employ the transcription model of (Benetos & Dixon, 2012), which supports the transcription of multiple-instrument polyphonic music and also supports pitch deviations or frequency modulations. The model of (Benetos & Dixon, 2012) is based on probabilistic latent component analysis (PLCA), which is a latent variable analysis method which has been used for decomposing spectrograms (Shashanka et al., 2008) and can be viewed as a probabilistic version of non-negative matrix factorization (Li & Seung, 1999). For computational efficiency purposes, we employ the fast implementation from (Benetos et al., 2013), which utilized pre-extracted note templates that are also pre-shifted across log-frequency, in order to account for frequency modulations or tuning changes. In addition, as was shown in (Smaragdis & Mysore, 2009), PLCA-based models can utilise priors for estimating unknown model parameters, which will be useful in this paper for informing the acoustic transcription system with symbolic information.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

The transcription model takes as input a normalised log-frequency spectrogram $V_{\omega,t}$ (ω is the log-frequency index and t is the time index) and approximates it as a bivariate probability distribution $P(\omega,t)$. $P(\omega,t)$ is decomposed into a series of log-frequency spectral templates per pitch, instrument, and log-frequency shifting (which indicates deviation with respect to the ideal tuning), as well as probability distributions for pitch, instrument, and tuning.

The model is formulated as:

$$P(\omega,t) = P(t)\sum_{p,f,s} P(\omega|s,p,f)P_t(f|p)P_t(s|p)P_t(p) \quad (1)$$

where p denotes pitch, s denotes the musical instrument source, and f denotes log-frequency shifting (which indicates tuning/pitch deviations). $P(t)$ is the energy of the log-spectrogram, which is a known quantity. $P(\omega|s,p,f)$ denote pre-extracted log-spectral templates per pitch p and instrument s, which are also pre-shifted across log-frequency. The pre-shifting operation is made in order to account for pitch deviations, without needing to formulate a convolutive model across log-frequency. $P_t(f|p)$ is the time-varying log-frequency shifting distribution per pitch, $P_t(s|p)$ is the time-varying source contribution per pitch, and finally, $P_t(p)$ is the pitch activation, which essentially is the resulting music transcription. As a time-frequency representation in the log-frequency domain we use the constant-Q transform (CQT) with a log-spectral resolution of 60 bins/octave (Schörkhuber & Klapuri, 2010).

The unknown model parameters ($P_t(f|p)$, $P_t(s|p)$, $P_t(p)$) can be iteratively estimated using the expectation-maximisation (EM) algorithm (Dempster et al., 1977). For the *Expectation* step, the following posterior is computed:

$$P_t(p,f,s|\omega) = \frac{P(\omega|s,p,f)P_t(f|p)P_t(s|p)P_t(p)}{\sum_{p,f,s} P(\omega|s,p,f)P_t(f|p)P_t(s|p)P_t(p)} \quad (2)$$

For the *Maximization* step (without using any priors) unknown model parameters are updated using the posterior computed from the Expectation step:

$$P_t(f|p) = \frac{\sum_{\omega,s} P_t(p,f,s|\omega)V_{\omega,t}}{\sum_{f,\omega,s} P_t(p,f,s|\omega)V_{\omega,t}} \quad (3)$$

$$P_t(s|p) = \frac{\sum_{\omega,f} P_t(p,f,s|\omega)V_{\omega,t}}{\sum_{s,\omega,f} P_t(p,f,s|\omega)V_{\omega,t}} \quad (4)$$

$$P_t(p) = \frac{\sum_{\omega,f,s} P_t(p,f,s|\omega)V_{\omega,t}}{\sum_{p,\omega,f,s} P_t(p,f,s|\omega)V_{\omega,t}} \quad (5)$$

We consider the sound state templates to be fixed, so no update rule for $P(\omega|s,p,f)$ is applied. Using fixed templates, 20-30 iterations using the update rules presented in the present section are sufficient for convergence. The output of the system is a pitch activation which is scaled by the energy of the log-spectrogram:

$$P_t(p)\sum_{\omega} V_{\omega,t} \quad (6)$$

After performing 5-sample median filtering for note smoothing, thresholding is performed on $P(p,t)$ followed by minimum note duration pruning set to 40ms (corresponding to the length of one time frame) in order to convert $P(p,t)$ into a binary piano-roll representation, which is the output of the transcription system, and is also used for evaluation purposes.

3. Polyphonic Music Prediction System

It was demonstrated in (Boulanger-Lewandowski et al., 2012) how a music language model (MLM) can be used to improve the transcription performance of a purely acoustic model. The MLM employed there was based on the recurrent neural network-restricted Boltzmann machine (RNN-RBM). A related model — the recurrent neural network-neural autoregressive distribution estimator (RNN-NADE) was also used for the same purpose with comparable results. In the present work, we employ both the standard RNN, and the RNN-NADE as MLMs for boosting the transcription accuracy of the PLCA based model described in the previous section. In this section, we briefly describe the RNN-NADE which we used in our work as the MLM, and the necessary background for understanding this model.

3.1. Recurrent Neural Network

A recurrent neural network (RNN) is a powerful model for time-series data which is known to account for long-term temporal dependencies when trained effectively. Given a sequence of inputs $v_1, v_2, \ldots, v_T$ each in $\mathbb{R}^n$, the network computes a sequence of hidden states $\hat{h}_1, \hat{h}_2, \ldots, \hat{h}_T$ each in $\mathbb{R}^m$, and a sequence of predictions $\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_T$ each in $\mathbb{R}^k$ by iterating the equations

$$h_t = e(W_{\hat{h}x}v_t + W_{\hat{h}\hat{h}}\hat{h}_{t-1} + b_{\hat{h}}) \quad (7)$$

$$\hat{y}_t = g(W_{y\hat{h}}) \quad (8)$$

where $W_{y\hat{h}}$, $W_{\hat{h}x}$, $W_{\hat{h}\hat{h}}$ are the weight matrices and $b_{\hat{h}}$, b_y are the biases and e and g are pre-defined vector valued functions which are typically non-linear and applied element-wise. The RNN also has a special initial bias $b_{\hat{h}}^{init}$ which replaces the formally undefined expression $W_{\hat{h}\hat{h}}\hat{h}_0$ at time $t = 1$.

In theory, a recurrent neural network can be easily trained using the gradient-based Back-Propagation Through Time algorithm (Werbos, 1990) using the exactly computable error gradients in the network. However, 1^{st} order gradient methods fail to correctly train RNNs in certain cases. This difficulty has been associated with what is known as the *vanishing/exploding gradients* phenomenon (Bengio et al., 1994), where the errors exhibit exponential decay/growth as they are back-propagated through time. Several proposals have been made to overcome this difficulty while retaining the predictive power of the RNN (Hochreiter & Schmidhuber, 1997; Jaeger, 2002; Martens & Sutskever, 2011).

3.2. Neural Autoregressive Distribution Estimator

The neural autoregressive distribution estimator (NADE) (Larochelle & Murray, 2011) is a graphical model inspired by the Restricted Boltzmann Machine (Smolensky, 1986; Hinton, 2002). It shares the structural properties of the RBM in that it has a visible layer v (with biases b_v), a hidden layer h (with biases b_h), with these two layers connected by a weight-matrix W. It facilitates the exact inference $p(v)$ given an input vector v, which is not possible in RBMs since there one has to compute the intractable *partition function* (Larochelle & Murray, 2011). This wass made possible by thinking of the RBM as a *fully visible sigmoid belief network* (FVSBN) (Neal, 1992). The FVSBN is a special case of a family of models known as fully visible Bayesian networks (Frey, 1998) with the property

$$p(v) = \prod_{i=1}^{D} p(v_i | v_{parents(i)}) \quad (9)$$

where all observation variables v_i are arranged into a directed acyclic graph and $v_{parents(i)}$ corresponds to all the variables in v that are parents of v_i. In an FVSBN, the acyclic graph is obtained by defining the parents of v_i as all variables that are to its left, or $v_{parents(i)} = v_{<i}$ where $v_{<i}$ refers to the subvector containing all variables v_j such that $j < i$. In the case of the NADE, $p(v_i|v_{parents(i)})$ can be computed as follows

$$\begin{aligned} p(v_i = 1 | v_{parents(i)}) &= \sigma(b_v^{(i)}) + (W^T)_{i,\cdot} h_i) \quad (10) \\ h_i &= \sigma(b_h + W_{\cdot,<i} v_{<i}) \quad (11) \end{aligned}$$

Untying the weights W and W^T results in a more powerful model. In the NADE, the cost of computing $p(v)$ is $O(HD)$, where H is the number of hidden units and D is the dimensionality of the vector v.

3.3. Recurrent Neural Network-Neural Autoregressive Distribution Estimator

Putting together the models described in Sections 3.1 and 3.2, we obtain the RNN-NADE, which is a model proposed for high-dimensional time-series.

4. Combining Transcription and Prediction

NOTE: Could describe how the predictions made by the MLM influence the transcription of the AMT system in this section.

5. Evaluation

6. Conclusions

References

Benetos, Emmanouil and Dixon, Simon. A shift-invariant latent variable model for automatic music transcription. *Computer Music Journal*, 36(4):81–94, 2012.

Benetos, Emmanouil, Cherla, Srikanth, and Weyde, Tillman. An effcient shiftinvariant model for polyphonic music transcription. In *6th International Workshop on Machine Learning and Music*, 2013.

Bengio, Yoshua, Simard, Patrice, and Frasconi, Paolo. Learning long-term dependencies with gradient descent is difficult. *Neural Networks, IEEE Transactions on*, 5 (2):157–166, 1994.

Boulanger-Lewandowski, N., Bengio, Y., and Vincent, P. Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. In *29th International Conference on Machine Learning*, Edinburgh, Scotland, UK, 2012.

Dempster, A. P., Laird, N. M., and Rubin, D. B. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1): 1–38, 1977.

Ewert, Sebastian and Müller, Meinard. Score-informed voice separation for piano recordings. In *ISMIR*, pp. 245–250, 2011.

Ewert, Sebastian and Müller, Meinard. Score-informed source separation for music signals. In *Multimodal Music Processing*, pp. 73–94, 2012.

Frey, Brendam J. *Graphical models for machine learning and digital communication*. The MIT press, 1998.

Hinton, Geoffrey E. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14 (8):1771–1800, 2002.

Hochreiter, Sepp and Schmidhuber, Jürgen. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Jaeger, Herbert. Adaptive nonlinear system identification with echo state networks. In *Advances in neural information processing systems*, pp. 593–600, 2002.

Larochelle, Hugo and Murray, Iain. The neural autoregressive distribution estimator. *Journal of Machine Learning Research*, 15:29–37, 2011.

Li, D. D. and Seung, H. S. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, October 1999.

Martens, James and Sutskever, Ilya. Learning recurrent neural networks with hessian-free optimization. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 1033–1040, 2011.

Neal, Radford M. Connectionist learning of belief networks. *Artificial intelligence*, 56(1):71–113, 1992.

Rabiner, Lawrence and Juang, Biing-Hwang. Fundamentals of speech recognition. 1993.

Schörkhuber, C. and Klapuri, A. Constant-Q transform toolbox for music processing. In *7th Sound and Music Computing Conf.*, Barcelona, Spain, July 2010.

Sentürk, Sertan, Gulati, Sankalp, and Serra, Xavier. Score informed tonic identification for makam music of turkey. In *14th Int. Soc. for Music Info. Retrieval Conf. Curitiba, Brazil.(to appear)*, volume 195, pp. 206, 2013.

Shashanka, M., Raj, B., and Smaragdis, P. Probabilistic latent variable models as nonnegative factorizations. *Computational Intelligence and Neuroscience*, 2008. Article ID 947438.

Smaragdis, P. and Mysore, G. Separation by "humming": user-guided sound extraction from monophonic mixtures. pp. 69–72, October 2009.

Smolensky, Paul. Parallel distributed processing: explorations in the microstructure of cognition, vol. 1. chapter Information processing in dynamical systems: foundations of harmony theory, pp. 194–281. MIT Press, Cambridge, MA, USA, 1986. ISBN 0-262-68053-X.

Werbos, Paul J. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10): 1550–1560, 1990.