Music generation review

Aciditeam

September 17, 2015

0.1 HMM

HMM-based models have been widely used for harmonic structure analysis [**?**, **?**, **?**] or melodic composition [**?**], but not for polyphonic composition. Indeed, HMM are not well-suited for multi-dimensional data and, thus, remained targeted at monophonic or chord sequences. In harmonization models ([**?**]), an interesting "root relative" representation is used. Each note is represented by its distance to the root (given the key), while a chord label is added to indicate the harmonic context of the chord (its degree).

Hidden Markov Model (HMM) are a particular case of Markov Random Field (MRF), and Lavrenko proposed to use a less constrained lattice of binary units directly derived from the piano-roll representation[**?**]. Here, the time structure is eluded by only considering the notes onset, while removing their duration. The efficiency of such an under-defined representation can be doubted, but it offers a first easy-to-handle representation. Intuitively, this representation might perform well on music strictly aligned on a temporal grid such as Bach's chorales. Besides, pitches are reduced to 12 through an octave-equivalent transformation. This transformation (named *pitch-class transformation* here) is interesting as it can help the system focusing on the harmonic structure of music. The field is built on *directed* temporal connection, which is a reasonable assumption that only notes played in the past influence notes in the future. A more surprising and restrictive assumption is that a certain pitch is only influenced by lower pitches at the same time. The distribution of the field is defined through a set of feature functions and parameters, representing rules learned on a training dataset.

Several works are based on Restricted Boltzmann Machine (RBM), such as systems that automatically creates Jazz melody over a sequence of chords [**?**]. Here, a data representation well fitted for their problem is to perform a separate representation of chords and melodies. However the poor temporal structure (simple concatenation of successive frames) of their system appears as an immediate limitation. Very recently, two models aimed at automatic music generation specifically targeted the temporal dimension and were able to generate interesting music sequences.

0.2 RNN-RBM

0.2.1 Data representation

Pianoroll

0.2.2 Model

A first paper [**?**] introduces two models with a temporal component used to produce polyphonic music sequences : the Recurrent Temporal Restricted Boltzmann Machine (RTRBM) and the Recurrent Neural Network Restricted Boltzmann Machine (RNN-RBM) (see figure 1 on page 2). Those two models have been inspired by the aforementioned text-generative model [**?**]. Beside, an interesting evaluation framework is proposed in this article and will be used to evaluate the performances of our model (section **??**). The general definition of a Temporal Restricted Boltzmann Machine (TRBM) introduced by [**?**] is a sequence of RBM where each RBM is conditioned by the previous one. It can be seen as a HMM with an exponentially large state space (increases with N^t) but a compact parametrization (only 2 transition matrices and biases). In fact, in the simplest TRBM, only the bias of the hidden units is affected by the previous hidden unit.

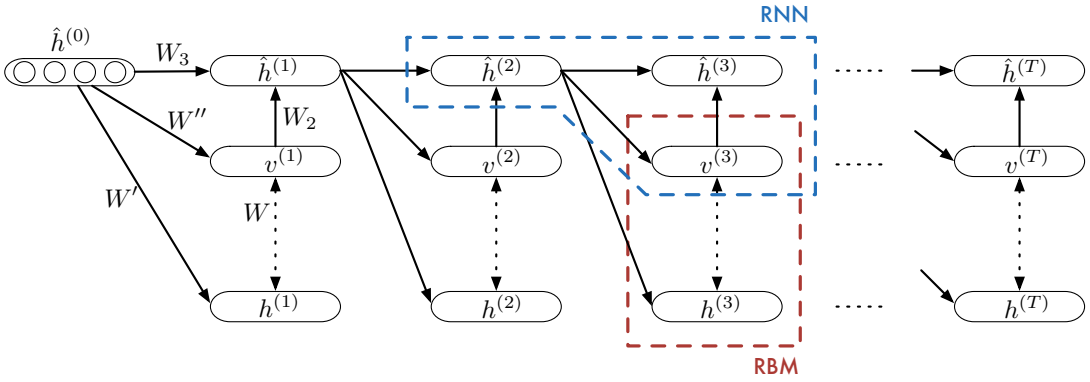

Figure 1: Graphical structure of the RNN-RBM. We can distinguish the two parts of the model, RNN in blue and RBM in red.

The probability of a sequence of observation is given by

$$P(v_1^T, h_1^T) = P_0(v_1, h_1).\prod_{t=2}^{T} P(v_t, h_t|h_{t-1}) \tag{1}$$

which is the equation of an HMM, and where the conditional distribution $P(V_t, H_t|H_{t-1})$ is the distribution of an RBM whose hidden bias H_t is a function of h_{t-1} (named *dynamic bias*).

$$P(v_t, h_t|h_{t-1}) = exp(v_t^\top b_V + v_t^\top W h_t + h_t^\top (b_H + W' h_{t-1}) \tag{2}$$

The main difference between the RNN-RBM model and the RTRBM is that the recurrent relation is based on the mean-field values of the hidden units instead of their binary values. Hence, those mean-field values units are separated from their binary alter ego and define another set of weights that models the recurrent relations. A RNN-RBM can be trained using the Back-propagation though time (BPTT) algorithm [**?**].

0.3 Long Short-Term Memory (LSTM)

0.3.1 Data representation

Pianoroll

0.3.2 Model

The paper [**?**] addresses blues improvisation by proposing a model based on a slightly modified Long Short-Term Memory (LSTM) network [**?**]. More precisely, their model is composed of an input layer, an output layer, and a hidden layer made of a standard feed-forward layer in parallel with several LSTM blocks, and use the same aforementioned evaluation framework [**?**]. A *LSTM* block in a network plays the same role as any other computational unit (they compute a single output value from weighted inputs), but have a more complex structure. Since the generation process is a random selection, very unlikely musical event can still happen. Even if we want to keep a certain amount of randomness in our generative process, we still don't want those very

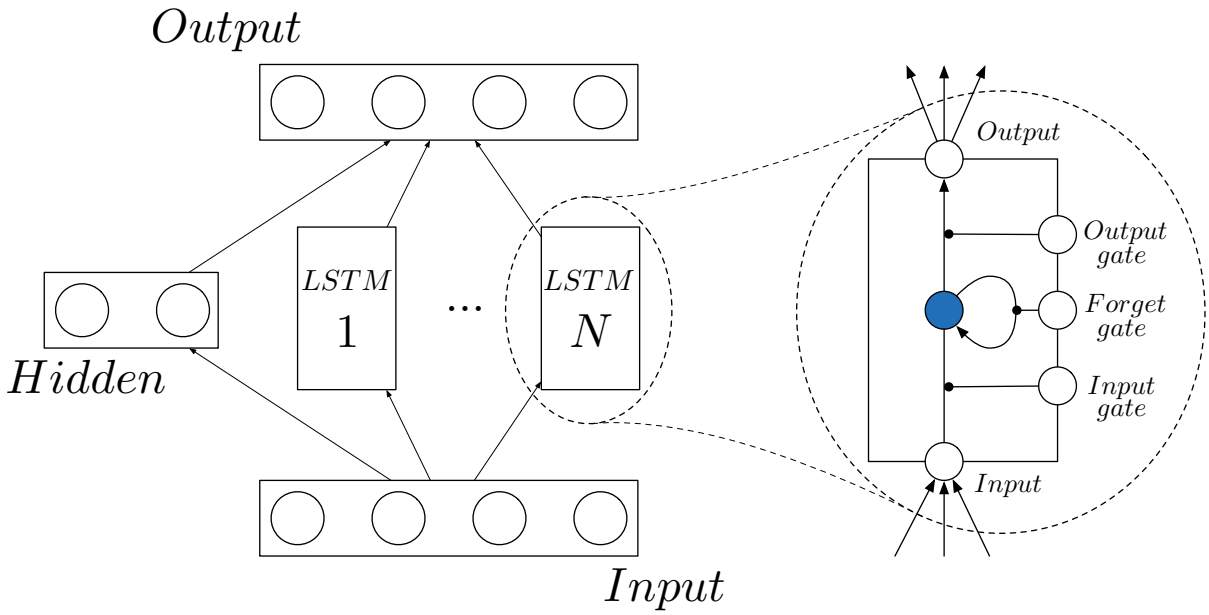

Figure 2: The modified LSTM model used in [**?**]. The hidden layer is a standard feed-forward layer in parallel with

unlikely event to happen. To do so, a threshold is applied before sampling from the output distribution in order to avoid very low probability notes to be chosen.

0.4 Hexadria (RNN-based architecture)

0.4.1 Data representation

0.4.2 Model

Proposed by Daniel Johnson (no paper, only a well documented website http://www.hexahedria.com/2015/08/03/com music-with-recurrent-neural-networks/, 2015). "The solution I decided to go with is something I am calling a biaxial RNN. The idea is that we have two axes (and one pseudo-axis): there is the time axis and the note axis (and the direction-of-computation pseudo-axis). Each recurrent layer transforms inputs to outputs, and also sends recurrent connections along one of these axes. But there is no reason why they all have to send connections along the same axis!" "Notice that the first two layers have connections across time steps, but are independent across notes. The last two layers, on the other hand, have connections between notes, but are independent between time steps. Together, this allows us to have patterns both in time and in note-space without sacrificing invariance!" Note that the two last layers on the figure don't have temporal connections.

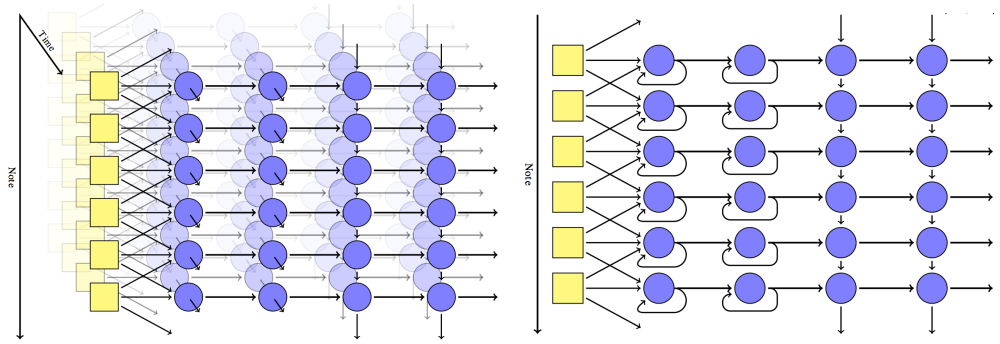

Figure 3: Convolution RNN. The same little network is associated to each note (convolutive network). In upper layers, thos network have *lateral* connection.