A Deep Learning Approach to Rhythm Modelling with Applications

Aggelos Pikrakis

Department of Informatics, University of Piraeus, Greece
e-mail: pikrakis@unipi.gr, web: www.cs.unipi.gr/pikrakis

Abstract. This paper presents a deep-learning architecture which is capable of modelling signatures that represent the rhythm of music recordings. The proposed architecture consists of a stack of Restricted Boltzmann Machines on top of which lies an associative memory. In the current study, we operate this architecture as a classifier which can discriminate among genres of rhythm. As a proof of concept, our method is applied on a standard corpus of ballroom dances. The results indicate that the proposed architecture exhibits promising learning capabilities and satisfactory generalization performance.

1 Introduction

Over the past decade, numerous studies in the field of Music Information Retrieval (MIR) have focused on the development of computational tools for tempo induction, beat tracking and related problems. This paper is making an attempt to approach the task of rhythm modelling from a *deep-learning perspective*. Deep architectures have been gaining popularity over the past few years in various machine learning disciplines but their penetration in the area of MIR is still limited [1], [2], [3].

The proposed method operates on *rhythmic signatures (patterns)*, i.e., fixed-length representations that are able to capture inherent signal periodicities (key components of the rhythmic structure), especially in the case of popular music that exhibits repetition [4], [5], [6]. In this study, we extend previous work by the authors [4] and propose a new type of rhythmic signature that has a probabilistic interpretation. Furthermore, we focus on the design of a deep architecture that can operate as a *classifier* of such signatures. During the training stage of the classifier, the patterns at the output of the feature extraction stage (Section 2) are fed as input to a deep-learning network (Section 3) that consists of a stack of Restricted Boltzmann Machines (RBMs), on top of which lies an associative memory. The training stage is based on training each RBM individually using Contrastive Divergence and then the network as a whole using back-propagation. As a proof of concept (Section 4), we have experimented with different network sizes on a 10-class task stemming from the frequently referenced dataset of Ballroom Dances [7].

2 FEATURE EXTRACTION

Each music recording is divided into highly overlapping long-term windows (10 s long, 9 s overlap). At each long-term window, a short-term feature extraction process extracts a sequence, $\mathbf{F} = \{\underline{F}(i), i = 1, \ldots, N\}$, of chroma-based MFCCs [4]. Recommended values for the length and step of the short-term moving window are 100 ms and 5 ms, respectively. At a next step, the averaged Euclidean distance, $D(k)$, between $\mathbf{F}$ and its shifted versions is computed over a range of lags, k, $k = 1, \ldots, L$:

$$D(k) = \frac{1}{N-k} \sum_{i=k}^{N} || \ \underline{F}(i) - \underline{F}(i-k) \ ||, \ k = 1, \ldots, L \tag{1}$$

In MML 2013: International Workshop on Machine Learning and Music, ECML/PKDD, Prague, 2013.

Assuming that in popular music slow music meters can be up to 4 s long, the maximum lag, L, for computing $D(k)$ is $L = \frac{4}{0.005} = 800$. Figure 1(a) presents the extracted sequence $\mathbf{D}$ from an excerpt of a Samba recording. Strong local minima (valleys) indicate dominant signal periodicities in the music recording.

At a next stage, we quantify the sharpness (depth) of the local minima. As a measure of sharpness, we have adopted the estimate of the second-order derivative, $\mathbf{D}_2$, of $\mathbf{D}$. To compute this estimate, we first approximate the first-order derivative, $\mathbf{D}_1$, and use the result to estimate the second-order derivative:

$$D_1(k) = \sum_{l=-3}^{3} D(k),\ k = 1, \ldots, L \ \ and \ \ D_2(k) = \sum_{l=-3}^{3} D_1(k),\ k = 1, \ldots, L \qquad (2)$$

Finally, $\mathbf{D}_2$ is *softmax* normalized to yield $\mathbf{D}_n$, i.e., $D_n(k) = \frac{1}{1+\exp(-\hat{D}_2(k))}$, where $\hat{D}_2(k) = \frac{D_2(k)-\mu}{\sigma}$ and μ and σ are the mean value and standard deviation of $\mathbf{D}_2$, respectively. The

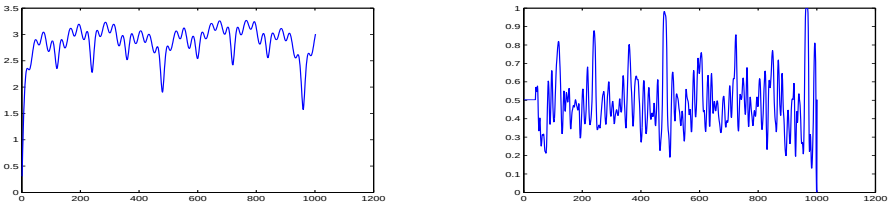

Fig. 1. Averaged Euclidean Distance (left) and Rhythmic Signature (right).

values of $\mathbf{D}_n$ (right part of Figure 1) lie in the range $[0, 1]$ and $D_n(k), k = 1, \ldots, L$, can be interpreted as the probability that k corresponds to an inherent signal periodicity. $\mathbf{D}_n$ is the *rhythmic signature (pattern)* of the long-term window. The demand for a probabilistic interpretation raises from the fact that the patterns will be eventually used to feed the layer of visible nodes of a Restricted Boltzmann Machine (RBM) [8]. Note that although some of the extracted signatures may turn out to be outliers, it is beyond the scope of this study to filter out any such cases, even though they introduce noisy data to our datasets.

3 Deep-learning architecture

The architecture that we propose is based on the guidelines that have been set in [8] and [9]. Specifically, we have experimented with a stack of Restricted Boltzmann Machines (RBMs), $RBM_i, i = 1, \ldots, R$, where the output of the hidden layer of the i-th RBM is clamped to the visible (input) layer of the next RBM. Note that the patterns of the training set feed the visible nodes of the first RBM. On top of the stack of RBMs lies an associative memory, whose role is to produce a vector of soft outputs by means of a softmax operation. *Each soft output is interpreted as the posterior probability that the rhythmic signature belongs to the respective class of the problem at hand.* Before the actual training procedure begins, the patterns are shuffled and form batches. We now provide an outline of the training steps:

- Each RBM is trained separately using the CD_1 algorithm [8]. CD_1 stands for a version of the Contrastive Divergence (CD) algorithm, according to which a single iteration of alternate Gibbs sampling is used to compute the contribution of each training vector in the equation that updates the weights of the RBM. During a training epoch, all data

batches are processed and in the end of each batch an update of the weights takes place. The training of the i-th RBM starts after the training of the $(i-1)$-th RBM has been completed.
- After all RBMs have been trained, an iterative back propagation algorithm that minimizes the cross-entropy error takes place for the deep network as a whole. During the back-propagation training stage, the output nodes of the associative memory are clamped to the binary representation of the pattern labels. In other words, the weights of the previously trained RBMs only serve for the initialization of the back-propagation training procedure. Several studies have shown that this approach outperforms the direct application of back-propagation on a randomly initialized deep architecture [9].

After the deep architecture has been trained, it is used for the classification of signatures. Specifically, each recording of the test set goes through the feature extraction stage. Each extracted pattern is clamped to the nodes of the visible layer of the first RBM and alternate Gibbs sampling is employed to determine the states of the hidden layer of the first RBM. The results propagate in the stack of RBMs until the softmax nodes of the associative memory are activated. For each signature, the maximum soft output is selected and the class which corresponds to this soft output is the class to which the signature is assigned.

4 Experiments

Our experimental setup evolves around the Ballroom Dataset (BD) [7], which consists of 10 dance styles: Cha Cha Cha, Jive, Quickstep, three Rumba styles, Samba, Tango, Viennese Waltz and Waltz. Each genre consists in turn of a varying number of 30 s excerpts. Overall the dataset is class-imbalanced which poses an extra challenge for any classification algorithm. For example, the Cha Cha Cha style consists of 111 excerpts, whereas a compilation of 60 segments forms the Jive genre.

A repeated hold-out scheme is applied on the BD and at each hold-out iteration 80% of the recordings of each genre are randomly selected for training and the rest for testing. The feature extraction stage produces, on the average, 22 patterns per file and over 12000 patterns for the training set as a whole. Each signature is associated with a 10-bit class label, e.g., [0 1 0 0 0 0 0 0 0 0] for the Jive genre. The generated signatures are randomly shuffled and grouped to batches (100 signatures per batch). Each RBM is individually trained for 100 epochs and the back-propagation procedure lasts 200 epochs. We have experimented with several architectures, with 1 up to 10 RBMs (layers) and with a varying size of hidden layers ($50-2000$ nodes).

Figure 2 presents the overall classification error for selected network sizes and for increasing decision threshold. The use of a decision threshold implies that we only trust a decision if the maximum soft output exceeds a user defined threshold, T_h. As a result, some patterns are left unclassified. The dotted curves present the percentage of patterns that have been left unclassified and the curves with squares present the classification error. Each curve has been tagged with the respective network size. Each tag is a description of the number of hidden nodes of the RBMs. For example, the tag $(500 \times 500 \times 2000)$ implies that 3 RBMs are used. The size of the first RBM is (800×500) (800 is the length of the signature) and the sizes of the two remaining RBMs are (500×500) and (500×2000), respectively. Obviously, the size of the associative memory is (2000×10).

The error is computed over the patterns whose soft output exceeds the threshold. It can be seen that for the 10-class task, the probability of error is always less than 20% irrespective of the size of the network and the confidence threshold. An example of interpretation of the presented curves is the following: if a random signature is drawn from the test set, there exists a 12% error probability that it is misclassified by the $500 \times$

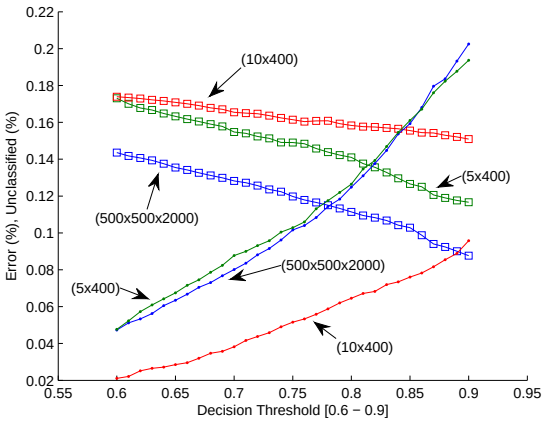

Fig. 2. Overall Accuracy and Unclassified Patterns vs decision threshold.

500×2000 network, given that the decision threshold is equal to 0.75 and that unclassified patterns have been ignored. Another interesting observation is that the (10×400) network is less sensitive to the decision threshold to the expense of a higher classification error. We have observed a similar behaviour with other networks that consist of at least 10 layers (which are however not presented here due to space restrictions).

5 Conclusions

The proposed modelling approach provides supporting evidence that deep-learning networks can be adopted for discriminating between genres based on extracted rhythmic signatures. Future research is expected to shed light on the impact of the training set on the performance measurements and on the generalization capabilities of the networks with respect to their depth and the number of classes which are involved.

References

1. Battenberg, E., Wessel, D.: Analyzing drum patterns using conditional deep belief networks. In: Proc. ISMIR. (2012) 37–42
2. Bickerman, G., Bosley, S., Swire, P., Keller, R.: Learning to create jazz melodies using deep belief nets. In: Proc. International Conference On Computational Creativity, Lisbon, Portugal. (2010)
3. Hamel, P., Eck, D.: Learning features from music audio with deep belief networks. In: Proc. ISMIR, Utrecht, The Netherlands (2010) 339–344
4. Pikrakis, A., Antonopoulos, I., Theodoridis, S.: Music meter and tempo tracking from raw polyphonic audio. In: Proc. ISMIR. (2004) 192–197
5. Holzapfel, A., Stylianou, Y.: Rhythmic similarity in traditional turkish music. In: Proc. ISMIR. (2009) 99–104
6. Gouyon, F., Dixon, S.: A review of automatic rhythm description systems. CMJ **29**(1) (2005)
7. Gouyon, F., Klapuri, A., Dixon, S., Alonso, M., Tzanetakis, G., Uhle, C., Cano, P.: An experimental comparison of audio tempo induction algorithms. Audio, Speech, and Language Processing, IEEE Transactions on **14**(5) (2006) 1832–1844
8. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural computation **18**(7) (2006) 1527–1554
9. Hinton, G.: A practical guide to training restricted boltzmann machines. Momentum **9**(1) (2010)