Beat Tracking using Recurrent Neural Network: a Transfer Learning Approach

Davide Fiocchi*, Michele Buccoli+, Massimiliano Zanoni+, Fabio Antonacci+, Augusto Sarti+
Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano,
piazza Leonardo da Vinci 32 - 20133 Milano, Italy
Email: * davide1.fiocchi@mail.polimi.it; + {name.surname}@polimi.it

***Abstract*—Deep learning networks have been successfully applied to solve a large number of tasks. The effectiveness of deep learning networks is limited by the amount and the variety of data used for the training. For this reason, deep-learning networks can be applied in scenarios where a huge amount of data are available. In music information retrieval, this is the case of popular genres due to the wider availability of annotated music pieces. Instead, to find sufficient and useful data is a hard task for non widespread genres, like, for instance, traditional and folk music. To address this issue, Transfer Learning has been proposed, i.e., to train a network using a large available dataset and then *transfer* the learned knowledge (the hierarchical representation) to another task. In this work, we propose an approach to apply transfer learning for beat tracking. We use a deep BLSTM-based RNN as the starting network trained on popular music, and we transfer it to track beats of Greek folk music. In order to evaluate the effectiveness of our approach, we collect a dataset of Greek folk music, and we manually annotate the pieces.**

I. Introduction

In the last decade, deep learning networks have been successfully applied in a number of research fields, including image and video processing, speech recognition and forensics, greatly outperforming traditional machine learning techniques. Deep learning techniques have reached high performance thanks to their ability to decompose problems into subproblems using several hierarchical layers of representation and finding useful semantic patterns within data.

In Music Information Retrieval (MIR), deep learning has been applied to several tasks, including automatic music classification [1], music structural analysis [2] and beat tracking. With regard to the latter, several architectures have been proposed [3], [4], [5], [6].

Deep networks require a huge amount of data to be effectively trained, which might not be available. In the context of MIR, this drawback is exacerbated by issues related to music copyright and lack of annotated dataset. With regard to beat tracking, some popular datasets include Beatles [7] and Robbie Williams [8], which contain hours of annotated musical excerpts mainly drawn from Pop Western genres.

The effectiveness of deep learning networks is also limited by the variety of the data used to train them: networks trained with a high variety of data is more prone to be effective also on data that are dissimilar by those used in the training [9]. Dataset composed by music of different genres are RWC Dataset [10] and Ballroom dataset [11].

Nevertheless, acquiring large datasets with a high variability is not always feasible. In MIR, this the case of non-popular or traditional and folk music. Recently, the MIR community has started addressing the MIR tasks with non-Western music. With regard to beat tracking and rhythm analysis, in [12] the authors investigate the occurrence of pulsations in free-rhythm Turkish music, while in [13] they employ a Gaussian Mixture Model to track beats from Cretan folk music.

However, adapting deep learning approaches to non-Western music is still a complex process, due to the lack of available datasets of proper size. A recent approach based on miming the ability of human brain to address novel problems by applying methods and knowledge acquired to solve similar problems in different contexts [14] is effective for overcoming the need of large datasets.

In deep learning, this means to use parameters of networks designed and trained for a source task, in a network devoted to a different task [14]. Thanks to this process, the network can benefit from the large available datasets to build a hierarchical representation of the input data, and then *transfer* the learned knowledge to another task. This procedure, known in literature as *Transfer Learning*, has been effectively used for different tasks, including those related to MIR [15].

In this study, we propose an approach to apply transfer learning for music beat tracking. We use the deep recurrent bi-directional neural network based on Long Short-Term Memory cells proposed in [4] as the starting network. We then employ the learned parameters on a network designed for folk music beat tracking. In order to evaluate the effectiveness of our approach, we collect a dataset of Greek folk music, in which we manually annotate the beat for each piece.

II. Recurrent Neural Networks: an overview

Recurrent Neural Networks (RNNs) are a family of neural networks specialized in modeling sequential data, which makes them suitable to model the time-varying evolution of musical properties. In this Section, we offer a brief overview of the basic (*Baseline*) RNNs and the Long Short-Term Memory (LSTM) cells we used in our study. For a more in-depth discussion, please refer to [16].

A. Baseline Recurrent Neural Network

Given a sequence $\mathbf{x}_t \in \mathbb{R}^N$ over a time index t, a RNN aims at modeling its time evolution and generate hidden

ISBN 978-90-827970-1-5 © EURASIP 2018

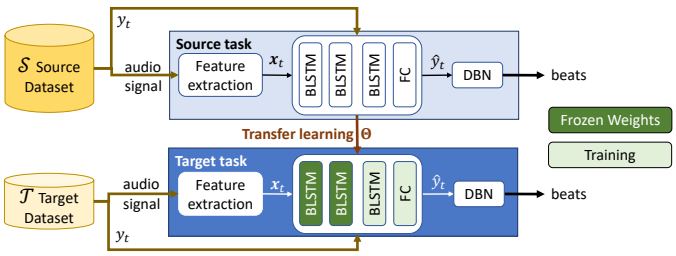


Fig. 1: General scheme of the algorithm.

units $\mathbf{h}_t = \sigma_h(\mathbf{h}_{t-1}, \mathbf{x}_t;) \in \mathbb{R}^H$ as a function σ_h of the current sample $\mathbf{x}_t$ and of the previous-sample $\mathbf{h}_{t-1}$, with parameters $\boldsymbol{\theta}_h$. Doing this, the function recursively accesses to past samples and hidden units, allowing the RNN to model the evolution of the sequence. See Fig. 2a for a simple graphical representation of a RNN unit.

RNNs are trained with the goal of learning the most representative hidden units, given a desired output $y_t \in \mathbb{R}$, in order to minimize a loss function $\mathcal{L}(y_t, o(\mathbf{h}_t))$, where o is an output function that maps the hidden units into the predicted output $\hat{y}_t$.

We can stack RNNs together in a deep architecture (shown in Fig. 2c) to learn a hierarchical representation of the sequence that is effective to bridge the gap between low-level input and high-level desired output. Given a network with L layers, we compute the generic l-th layer's units $\mathbf{h}_t^{(l)}$ as

$$\mathbf{h}_t^{(l)} = \sigma_h^{(l)}\left(\mathbf{h}_{t-1}^{(l)}, \mathbf{h}_t^{(l-1)}\right) \in \mathbb{R}^{H^{(l)}}, \text{ with } \mathbf{h}_t^{(0)} = \mathbf{x}_t.$$

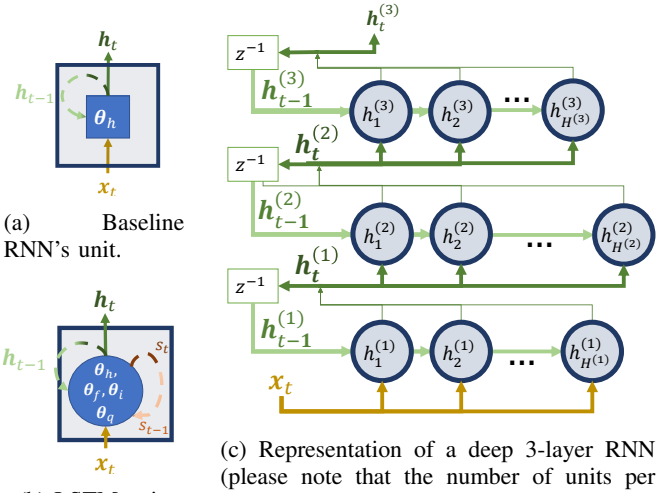
Each layer l learns a different parametrization $\boldsymbol{\theta}_h^{(l)}$. The predicted output $\hat{y}_t$ is computed from the top (and most semantic) hidden layer's output $\mathbf{h}_t^{(L)}$ or from the stacking of several layers' output, depending on the task.

A variation of RNN, named Bidirectional RNNs (BRNNs) [17], exploits not only the past hidden values but also future ones. In a BRNN, two sub-layers compose each layer, one that learns $\mathbf{h}_t$ given $\mathbf{h}_{t-1}$, and one that learns $\mathbf{h}_t$ given $\mathbf{h}_{t+1}$. Exploiting future information makes the BRNNs more effective than RNN to predict a desired output, but limits their applicability to offline (non-causal) tasks.

A severe issue prevents RNNs from learning an effective multi-layer representation, namely the *vanishing gradient* [4], i.e., the gradient of $\mathcal{L}$ decreases so much that the weights' update stops. The LSTM cells [18] address this issue by using an internal memory and using *gates* to weigh the contribution of input and previous output to the memory, and of the output of the network.

B. Long Short-Term Memory units

LSTM units introduce a cell state s_t that acts as a memory for the unit, and three *gates* f, i and q, named *forget*, *input* and *output* gates respectively, as scalars ranging between 0 and 1 to weigh the contribution from $\mathbf{x_t}$, $\mathbf{h}_{t-1}$ and s_{t-1}. The forget gate f_t is computed as $f_t = \sigma_f(\mathbf{x}_t, \mathbf{h}_{t-1})$, using



(a) Baseline RNN's unit.

(b) LSTM unit.

(c) Representation of a deep 3-layer RNN (please note that the number of units per layer may vary).

Fig. 2: A representation of two units of a deep RNN.

specific parameters $\boldsymbol{\theta}_f$; the gates i_t and q_t are computed accordingly with parameters $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_q$, respectively. The vector $\tilde{\mathbf{h}}_t = \sigma_h(\mathbf{x}_t, \mathbf{h}_{t-1})$, with parameters $\boldsymbol{\theta}_h$, is used to update the state cell as $s_t = f_t s_{t-1} + i_t \tilde{\mathbf{h}}_t$ by weighing the contribution of the previous state (first addend) and of the input (second addend). Lastly, the hidden units are computed by weighing the contribution of the cell state as $\mathbf{h}_t = q_t \tanh(s_t)$.

The training of deep bi-directional LSTMs (BLSTM) is analogous to that of BRNNs, with the space of parameters also including $\boldsymbol{\theta}_f$, $\boldsymbol{\theta}_i$, $\boldsymbol{\theta}_q$ (Fig. 2b), hence increasing the requirements in terms of training data.

III. Proposed Approach

In this work, we use a beat tracking algorithm similar to the one proposed in [3] and [5], and we use transfer learning to use the learned network on a different beat tracking task. In Figure 1 we show the global schema of our approach.

Given an audio signal, we extract a perceptually-meaningful time-frequency representation (Section III-A), which is the input of the deep network (Section III-B). The network outputs a likelihood that a beat occurs at a certain instant of the audio signal. Since in music the sequence of beats follows some rules given by the composition and the music theory, we add a Dynamic Bayesian Network (DBN) (Section III-C) at the top of the deep network in order to transform likelihoods to a sequence of beat instants.

We train tracking algorithm, such as described in [3] and [5] and we apply transfer learning such as described in Section III-D.

A. Feature extraction

Given a generic audio signal x, we extract a sequence of vectors $\mathbf{x}_t$ with $t = 1, ..., T$. Following the approach proposed in [19], we compute the log-magnitude STFT of the signal with three different window sizes W_1, W_2 and W_3, to exploit the trade-off between frequency and time resolution. In order for the frames to be comparable, we use a common

ISBN 978-90-827970-1-5 © EURASIP 2018

hopsize. We filter the spectrograms using a bank of triangular filters with a constant resolution per octave, to reduce the number of components of the vector and extract a frequency representation closer to human perception. We stack together the three versions of the spectrogram and their first-order half-rectified difference to compose the final sequence $\mathbf{x}_t \in \mathbb{R}^N$. We then use it as the input of the network.

B. Network

We use a three-layer BLSTM-based RNN network followed by a fully-connected (FC) layer to compute the output *beat activation* $\hat{y}_t = \sigma_o\left(\mathbf{h}_t^{(3)}\right)$ as the probability that a beat occurs at time t. We train the network using labeled annotation $y_t = \{0, 1\}$ that indicates whether ($y_t = 1$) or not ($y_t = 0$) a beat occurs at time t. The sequence $\hat{y}_t \in [0, 1]$ ranges in the continuous domain and is discretized into a set of beat instants $\mathbf{b} = \{b_1, ..., b_B\}$ by the DBN.

C. Dynamic Bayesian Network

As done in [19], we use DBNs to track and model the beat sequences. In the DBN, states $\mathbf{z}_t$ are identified by two variables: a measure of the tempo Φ_t and the position of the beat within a bar $\dot{\Phi}_t$. States can be imagined as in a grid spanned by these two components. At each time instant t, the *transition model* estimates the probability of a transition between states $P(\mathbf{z}_t|\mathbf{z}_{t-1})$. Given the sequence of states $\mathbf{z}_{1:T}$, the probability sequence of transitions among states is:

$$P(\mathbf{z}_{1:T}|\hat{y}_{1:T}) \propto P(\mathbf{z}_1)\prod_{t=2}^{T} P(\mathbf{z}_t|\mathbf{z}_{t-1})P(\hat{y}_t|\mathbf{z}_k), \qquad (1)$$

where: $P(\mathbf{z}_1)$ is the *initial state distribution*, usually initialized as a uniform (equiprobable) distribution, and $P(\hat{y}_t|\mathbf{z}_k)$ is the *observation model*, which employs the beat activation as the observable variable to guide the transitions.

Using the Viterbi algorithm [19], the most likely sequence of transitions among states $\mathbf{z}^*_{1:T}$ is computed as:

$$\mathbf{z}^*_{1:T} = \{\mathbf{z}^*_1, \ldots, \mathbf{z}^*_T\} = \arg\max_{\mathbf{z}_{1:T}} P(\mathbf{z}_{1:T}|\hat{y}_{1:T}). \qquad (2)$$

By looking at the transition of states occurring through $\mathbf{z}^*_{1:T}$, we obtain an indicator of the ranges where a beat is most likely to occur. Within this ranges, beats are finally found as t where $\hat{y}_t$ is a local maximum.

D. Transfer Learning

Transfer learning aims to exploit the knowledge acquired on a source task to perform a target task. This is consistent with what happens in real life, where pre-existing knowledge is used to process and understand new informations. We apply *inductive transfer learning* to use a model previously trained for a source task on a source domain, to perform a new target task on the same source domain [14]. Therefore, we use a *parameter* transfer learning, which assumes that network coefficients are composed of two parts: one shared among tasks and one task-specific [14].

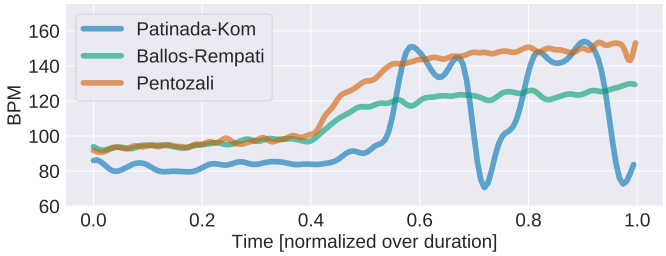


Fig. 3: Evolution of BPM for three pieces in the *Greek folk dataset*.

Due to the nature of the network, the lower layers of RNN learn a basic representation of the music rhythm, which can be effectively used for the new task. The higher layers, instead, tend to be more specialized and, for this reason, need to be re-trained, as in [15].

In Figure 1, we detail the proposed approach. We first train a RNN for the source task over a large database $\mathcal{S}$, learning the parameters $\mathbf{\Theta} = \{\boldsymbol{\theta}_h^{(l)}, \boldsymbol{\theta}_i^{(l)}, \boldsymbol{\theta}_f^{(l)}, \boldsymbol{\theta}_q^{(l)} \forall l = 1, 2, 3\} \cup \boldsymbol{\theta}_o$. We then copy found parameters into a target network with the same architecture. Then, we use a smaller database $\mathcal{T}$ for the target task, for the training of the top BLSTM layer ($l = 3$) and of the fully-connected layer, whereas the lower two layers do not update the weights (i.e., their weights are *frozen*).

IV. Experimental Setup and Evaluation

A. Data Collection

The source dataset $\mathcal{S}$ we used in this study is the one proposed in [4], here named *Böck dataset*. It is composed of 120 songs from the Ballroom set [11], some training and bonus excerpts from the MIREX 2006 beat tracking contest[1] and six files from the set used in [20]. This dataset is used for training the source network and learn the first set of parameters $\mathbf{\Theta}$.

The smaller target dataset $\mathcal{T}$, named the *Greek folk dataset*, is composed of 56 music pieces typically used in traditional Greek folk dance. We asked traditional Greek music experts to annotate them with their beat instants. The dataset is publicly available[2]. $\mathcal{T}$ contains a wide variety of binary, ternary and odd meters, which may also change throughout the same piece. Moreover, musicians rarely use the metronome during recordings and tempo fluctuation are extremely common. In Fig. 3 we show some examples of BPM evolution over time for three excerpts from the dataset; whereas in Fig. 4 we show the distribution of song-level average BPM in the dataset.

Tempo fluctuations in $\mathcal{T}$ make the target task more challenging for a beat tracking approach that we trained over $\mathcal{S}$, due to the lack of non-steady examples to learn from. We highlight the difference of tempo fluctuations between the two datasets by computing the *Median Absolute Deviation* (MAD) over inter-beat intervals $\Delta_b = \{b_2 - b_1, b_3 - b_2, ..., b_B - b_{B-1}\}$, where b_β is the generic beat instant and B is the total number

[1] http://www.music-ir.org/mirex/wiki/2006:Audio_Beat_Tracking
[2] http://home.deib.polimi.it/buccoli/dataset/BeatGreek.zip

ISBN 978-90-827970-1-5 © EURASIP 2018

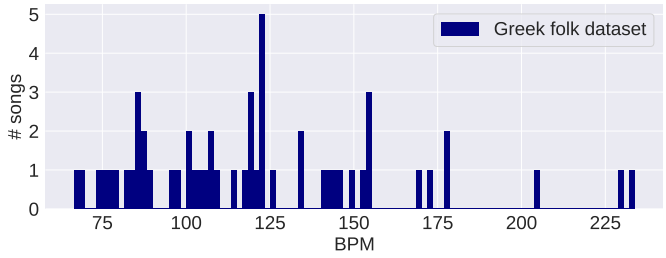


Fig. 4: Distribution of average BPM in the *Greek folk dataset*.

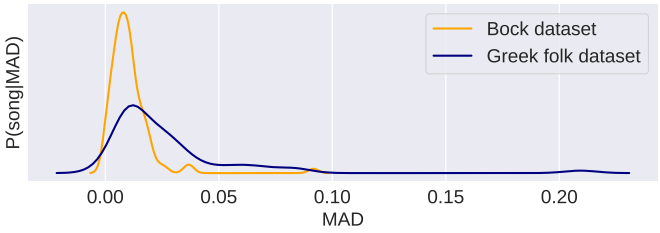


Fig. 5: Comparison of the distributions of MAD in Böck and Greek Folk datasets.

of beats. MAD is a robust measure of the variability of a univariate distribution and it is computed as:

$$MAD = \text{median}\left(|\Delta_b - \text{median}(\Delta_b)|\right).$$

The higher the MAD, the higher the variability of the inter-beat intervals and, hence, the degree of tempo fluctuations. In Fig. 5 we show a comparison between the normalized distributions of the annotations-based MADs in *Böck* and *Greek folk* datasets. While the former exhibits a high and narrow peak over low values of MAD, the latter is more skewed toward high values.

B. Setup

Given a generic monauralized PCM audio signal, re-sampled at 44.1 KHz, we use a filterbank with six bands per octave to filter the three STFTs with common hopsize of 10 ms and three different windows sized $W_1 = 1024$, $W_2 = 2048$ and $W_3 = 4096$ samples, obtaining 39, 45 and 49 components per frame, respectively. Considering also the first order difference of the spectrograms, we obtain $N = 266$ input units.

We follow the RNN setup described in [4]. In order to augment the number of training examples we divided the training set into 10-second batches, and we use an early stop approach with 20 epochs. We implemented the deep neural network using Keras [21].

In this work, we investigate whether a source network trained for beat tracking of a general-purpose dataset is able to *transfer* its knowledge to a target network for beat tracking on a specific dataset. For this reason, we compare the performance of three models derived from the source network: 1) the **source network**, as described [4], trained over the *Böck dataset*; 2) the target network **TL Freeze**, obtained using transfer learning only on the higher two levels (one BLSTM and one fully-connected), as described in Sec. III-D, and trained on the *Greek folk dataset*; 3) a network trained only with the Greek folk dataset, named **No TL**.

We trained all models but **No TL** with all the pieces in $\mathcal{S}$; then, we randomly picked 60% of $\mathcal{T}$ to train all models but the **source network**, and we left the remaining 40% for the evaluation.

C. Evaluation

The MIR community agrees that, given the complexity of the problem to address, it does not exist a unique metric able to capture all the aspect needed to evaluate effectiveness of beat-tracking methods. For this reason we evaluate the performance of the three aforementioned networks using a set of well-known metrics for beat-tracking [7]: **F-measure**, **P-score**, **Cemgil** measure, all the continuity-based metrics (Correct Metrical Level **CML** and Allowed Metrical Level **AML**, in both variants: continuous and total) and average **Information Gain** (info gain).

In Table I we report the result of the evaluation on the Greek folk dataset. The basic case of our experiments is the one we called the **source network** where the network obtained good results even if no Greek folk music was part of the training set. We obtained similar scores in the case where the network was only trained using Greek folk music (**no TL**). This is predictable since in the former case the network had no idea about Greek folk music, in the latter case, instead, the network tends to overfit on the training set, which in our case is very small, and is not able to generalize. Let the reader notice that in **no TL**, continuity-based CMLc and AMLc metrics obtained a very low value. We assume that, due to the limited training dataset, **no TL** is not able to correctly track beats for wide temporal regions in the musical piece.

Using the proposed method based on transfer learning, as expected, we obtained a significant improvement. In the **TL Freeze** case, indeed, for all the metrics we obtained an average improvement of about 6%.

In order to prove the effectiveness of our method we also performed preliminary tests on the SMC dataset, which is well-known to be extremely challenging for beat tracking algorithms [9]. In this tests we kept the three networks (TL Freeze, source network, no TL) unchanged, using the SMC only as the test set. In Table II we report the result on SMC dataset. For comparison, we also add the results achieved by [19] with a similar network trained over the SMC dataset.

Here it is possible to notice the positive effect of using transfer learning, while the improvement is not as relevant as in the previous scenario. However, the performance is still far from those obtained by a network trained over the specific dataset, as in [19].

Despite to perform beat tracking on the Greek dataset is an hard task, the SMC seems to be more challenging, as expected, and the result are much lower with respect to the previous cases.

ISBN 978-90-827970-1-5 © EURASIP 2018

TABLE I: Evaluation result on Greek dataset

	F-measure	P-score	cemgil	CMLc	CMLt	AMLc	AMLt	info gain
TL Freeze	0.640	0.645	0.574	0.495	0.509	0.752	0.780	0.555
Source Network	0.572	0.586	0.511	0.421	0.435	0.706	0.744	0.529
No TL	0.584	0.620	0.516	0.278	0.445	0.475	0.685	0.383

TABLE II: Evaluation result on SMC dataset

	F-measure	P-score	cemgil	CMLc	CMLt	AMLc	AMLt	info gain
TL Freeze	0.388	0.506	0.309	0.268	0.286	0.437	0.472	0.416
Source Network	0.374	0.488	0.298	0.221	0.234	0.367	0.399	0.369
No TL	0.368	0.506	0.298	0.187	0.253	0.263	0.378	0.183
Krebs [19]	0.543	-	0.431	-	0.458	-	0.613	1.578

V. Conclusions

In this work, we presented an application of transfer learning for beat tracking task, from a general-purpose to a narrow specific case. For the sake of evaluation, we collected a dataset of 56 Greek folk pieces annotated with the corresponding beats instants by experts. Performance achieved by the networks shows that the transfer learning is effective in re-tuning a RNN's parameters to track rhythmically-challenging music pieces.

We intend to continue the investigation on transfer learning for beat tracking exploring wider diversity across music genres. Moreover, together with the music pieces, we also collected a dataset of motion-capture movements of the Greek folk dataset. Future work includes an investigation of automatic approaches for tracking beats from dance motion.

Acknowledgment

This study was conducted for the WhoLoDancE project, which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 688865.

We would like to thank the project partner Amalia Markatzi and all the people who contributed to the manual annotation of the Greek folk dataset. Thanks to Sebastian Böck for his annotations and the precious informations he gave us during the development of this study.

References

[1] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Convolutional recurrent neural networks for music classification," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[2] M. Buccoli, M. Zanoni, A. Sarti, S. Tubaro, and D. Andreoletti, "Unsupervised feature learning for music structural analysis," in *Proc. of the 24th European Signal Processing Conference (EUSIPCO)*, 2016.

[3] S. Böck, F. Krebs, and G. Widmer, "Joint beat and downbeat tracking with recurrent neural networks." in *Proc. of the 17th Interational Society for Music Information Retrieval (ISMIR) conference*, 2016.

[4] S. Böck and M. Schedl, "Enhanced beat tracking with context-aware neural networks," in *Proc. of the 14th International Conference on Digital Audio Effects (DAFx)*, 2011.

[5] S. Böck, F. Krebs, and G. Widmer, "A multi-model approach to beat tracking considering heterogeneous music styles." in *Proc. of the 15th Interational Society for Music Information Retrieval (ISMIR) conference*, 2014.

[6] S. Durand, J. P. Bello, B. David, and G. Richard, "Downbeat tracking with multiple features and deep neural networks," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015.

[7] M. E. Davies, N. Degara, and M. D. Plumbley, "Evaluation methods for musical audio beat tracking algorithms," Queen Mary University of London, Centre for Digital Music, Tech. Rep., 2009.

[8] B. Di Giorgi, M. Zanoni, S. Böck, and A. Sarti, "Multipath beat tracking," *Journal of Audio Engeenering Society*, vol. 64, pp. 493–502, 2016.

[9] A. Holzapfel, M. Davies, J. Zapata, J. Lobato Oliveira, and F. Gouyon, "Selective sampling for beat tracking evaluation," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 20, pp. 2539–2548, 2012.

[10] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "Rwc music database: Popular, classical and jazz music databases." in *Proc. of the 3rd International Society for Music Information Retrieval (ISMIR) Conference*, 2002.

[11] F. Gouyon, A. Klapuri, S. Dixon, M. Alonso, G. Tzanetakis, C. Uhle, and P. Cano, "An experimental comparison of audio tempo induction algorithms," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1832–1844, 2006.

[12] A. Holzapfel, "Tempo and prosody in turkish taksim improvisation," in *Proc. of the 3rd Workshop on Folk Music Analysis (FMA*. Meertens Institute; Department of Information and Computing Sciences, Utrecht University, 2013.

[13] A. Holzapfel, F. Krebs, and A. Srinivasamurthy, "Tracking the "odd": meter inference in a culturally diverse music corpus," in *Proc. of the 15th International Society for Music Information Retrieval (ISMIR) Conference*, 2014.

[14] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, Oct 2010.

[15] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Transfer learning for music classification and regression tasks," in *Proc. of the 18th International Society of Music Information Retrieval (ISMIR) Conference 2017*. International Society of Music Information Retrieval, 2017.

[16] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[17] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[18] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[19] F. Krebs, S. Böck, and G. Widmer, "An efficient state-space model for joint tempo and meter tracking," in *Proc. of the 16th Interational Society for Music Information Retrieval (ISMIR) conference*, 2015.

[20] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. B. Sandler, "A tutorial on onset detection in music signals," *IEEE Trans. on Speech and Audio Processing*, vol. 13, no. 5, pp. 1035–1047, Sept 2005.

[21] F. Chollet *et al.*, "Keras," https://github.com/keras-team/keras, 2015.

ISBN 978-90-827970-1-5 © EURASIP 2018