DESIGNING EFFICIENT ARCHITECTURES FOR MODELING TEMPORAL FEATURES WITH CONVOLUTIONAL NEURAL NETWORKS

Jordi Pons and Xavier Serra

Music Technology Group, Universitat Pompeu Fabra, Barcelona

ABSTRACT

Many researchers use convolutional neural networks with small rectangular filters for music (spectrograms) classification. First, we discuss why there is no reason to use this filters setup by default and second, we point that more efficient architectures could be implemented if the characteristics of the music features are considered during the design process. Specifically, we propose a novel design strategy that might promote more expressive and intuitive deep learning architectures by efficiently exploiting the representational capacity of the first layer – using different filter shapes adapted to fit musical concepts within the first layer. The proposed architectures are assessed by measuring their accuracy in predicting the classes of the Ballroom dataset. We also make available[1] the used code (together with the audio-data) so that this research is fully reproducible.

Index Terms— Convolutional neural networks, deep learning, music, classification, information retrieval.

1. INTRODUCTION

Due to the convolutional neural networks (CNNs) success in image classification, its literature significantly influenced the music informatics research (MIR) community that adopted standard computer vision CNNs for music classification [1, 2, 3]. As long as these CNNs were designed for computer vision tasks, it is reasonable that some researchers assume that audio events can be recognized by *seeing* spectrograms [1, 2] (that are image-like time-frequency audio representations) and indeed, most MIR deep learning practitioners tend to use spectrograms as input to their CNNs [1, 2, 3, 4, 5]. The wide use of small rectangular filters in music classification [1, 3, 4, 5] (a standard filter shape in computer vision: $m \ll M$ and $n \ll N$)[2] states how straight-forward MIR researchers adopted computer vision CNNs, because note that image processing filter dimensions have spatial meaning while CNN-spectrogram filters dimensions correspond to time and frequency. Therefore, *wider* filters may be capable of learning longer temporal dependencies in the audio domain while *higher* filters may be capable of learning more spread timbral features. Hence, there is no grounded motivation for using by default such small rectangular filters for MIR since some relevant musical features (*ie.* rhythm, tempo or timbre) have long temporal dependencies or are spread in frequency. This observation motivates the hereby study, where we consider the characteristics of music for proposing filter shapes more suitable for music spectrograms. We hypothesize that MIR CNNs can benefit from a design oriented towards learning musical features rather than *seeing* spectrograms.

Moreover, a recent publication [4] points that small rectangular filters can limit the representational power of the first layer since these can only represent sub-band characteristics (with a small frequency context: $m \ll M$)[2] for a short period of time (with a small time context: $n \ll N$)[2]. Hence, the network needs to combine many filters (in the same layer and/or in deeper layers) in order to model larger time/frequency contexts, what adds an extra cost to the network (wider layers and/or deeper networks). Therefore, if an individual filter can model a larger time/frequency context in the first layer: *(i)* this might allow achieving a similar behavior without paying the cost of going wider and/or deeper; *(ii)* deeper layers are set free to model context – what may allow more expressive CNNs at a similar cost since depth can then be employed for learning other features; *(iii)* filters might be more interpretable since the whole desired context would be modeled within one single filter on top of a spectrogram (with clear dimensions: time and frequency); and *(iv)* interpretable filters allows taking intuitive decisions when designing CNNs in order to make an effective use of a reduced number of parameters. From previous remarks one can observe that very efficient[3] CNNs can be conceived by enabling the first layer to model larger contexts. Finally, note that given that some relevant musical features have long temporal dependencies or are spread in frequency, wide or high filters in the first layer (modeling larger time/frequency contexts) might be able to efficiently represent these features. Therefore, the here proposed strategy ties very well with the previously described need of proposing musically motivated filter shapes.

[1] https://github.com/jordipons/ICASSP2017

[2] Throughout this study we assume to use CNNs, with the input set to be music spectrograms of dimensions *M-by-N* and the CNN filter dimensions to be *m-by-n*. *M* and *m* standing for the number of frequency bins and *N* and *n* for the number of time frames.

[3] This is the notion of efficiency we assume all through this publication.

Our aim is to discover novel deep learning architectures that can efficiently model music, what is a very challenging undertaking. This is why we first focus on studying how CNNs can model temporal cues, one of the most relevant music dimensions. By considering the introductory discussion, in Section 2 we put emphasis on the design of a single-layer CNN designed to efficiently model temporal features, in Section 3 the proposed architectures are assessed by measuring their accuracy in predicting the genres of the Ballroom [6] dataset and Section 4 concludes.

2. ARCHITECTURES

It is common in deep learning to model temporal dependencies in sequential data with recurrent neural networks (RNNs) [7, 8]. Two successful methods that used RNNs for modeling temporal features from music audio are: Böck *et al.* [7] for tempo estimation and Krebs *et al.* [8] for downbeat tracking. However, note that we aim to study the capacity of CNNs for modeling temporal features. Before moving forward, we want to discuss why RNNs are not integrated within that study: first, CNNs are suitable for modeling short time-scale temporal features since the available context is limited by the size of the input spectrogram[4] and second, RNNs are suitable for modeling short and long time-scale temporal features since the available context can be the whole music recording. By modeling short time-scale features in the first layer, deeper layers are set free for modeling other features. Consequently, if a RNN layer is stacked on top of a CNN that is modeling short time-scale features (*ie.* rhythm, tempo or onsets), such RNN can focus on learning long time-scale temporal features (*ie.* structure). This is why we focus on the efficient modeling of short time-scale temporal features with single-layer CNNs and we leave for future work modeling long time-scale temporal features with RNNs stacked on top of CNNs.

Some existing research has focused on using CNNs for modeling temporal features, proposing innovative architectures: Durand *et al.* [9] used three parallel CNNs for modeling different music dimensions, Pons *et al.* [4] proposed a light CNN for learning temporal cues with wide filters (*1-by-n*) and max-pool frequency summarization, and Phan *et al.* [2] proposed using filters representing different time-scales (setting n differently for every filter) with a max-pool layer that spans all over *time* (operation that enables time-invariance). Together with the above introduction, these works [2, 4, 9] conform the basis for designing efficient CNN architectures.

Short time-scale temporal features in music audio are fundamental for describing several musically relevant concepts: *onsets* (*ie.* attack-sustain-release signatures define many instruments, and these are a relevant cue for predicting genre), *rhythm* (*ie.* can define a genre like waltz) or *tempo* (*ie.* some genres have faster tempos than others). Note that different time-scales are required for modeling these musical concepts. For example, for modeling onsets one requires a shorter time-context than for modeling tempo or rhythm. If a long filter is used for modeling onsets, most of the weights would be set to zero: wasting part of the representational power of the filter. Therefore, and similarly as in Phan *et al.* [2], we propose setting different n's for the filters in the first layer for being able to efficiently represent several time-scale contexts.

We propose two complementary architectures meant to validate the foundations of our novel *design strategy*, that promotes an *efficient use of the representational capacity of the first layer by using different musically motivated filter shapes that model several (time-scale[5]) contexts*:

I) ***O-net*** is designed to efficiently model *onsets*, a short time-scale temporal feature. Different (short) filters of *1-by-n* followed by a max-pool layer of *4-by-N'* [6] might be capable of capturing in which frequency band a short time signature is occurring, with $n \in [6, 11, 16, 21, 26, 31, 36, 41]$. The *O-net* consists on 5 filters for each different filter length n. In total, there are 40 filters in the same (first) layer.

II) ***P-net*** is designed to efficiently model short time-scale *patterns*, *ie.* rhythm or tempo. Different (longer) filters of *1-by-n* followed by a max-pool layer of *4-by-N'* [6] might be capable of capturing in which frequency band a time pattern is occurring, with $n \in 46+5 \cdot f$ where $f \in \mathbb{Z} \mid 0 \leq f \leq 34$ stands for the filter number. The *P-net* consists on 35 filters of different length in the first layer ranging from $46 \leq n \leq 216$.

We propose combining these architectures in parallel [9] and as a result of that, the resulting model is shallow: a single layer with many different filters. On top of this parallel combination of CNNs (no matter which combination) we stuck a softmax layer as output – Table 1 outlines the studied models. Note that these models fulfill the specifications of our design strategy (filter shapes are intuitively designed to represent different relevant musical contexts using a reduced number of parameters in the first layer) and therefore, it serves as test-bed to validate the proposed design strategy.

2.1. Filter and max-pool shapes discussion

We aim to investigate how CNNs can efficiently model short time-scale temporal features in music audio spectrograms. For doing so we propose using temporal filters (*1-by-n*) [4], that are a cheap filter expression to model temporal features where the temporal context can be easily adjusted by setting n. For example, faster patterns can be better represented by shorter filters than slower patterns, what allows minimizing the number of parameters used for these filters. But also note that shorter filter lengths can facilitate modeling faster

[4] This is the definition of short time-scale temporal features that we assume: a short time-scale temporal feature can be described within the available context limited by the size of the input spectrogram.

[5] Within the context of a spectrogram: $n \in [1, ..., N]$.

[6] *N'* and *M'* denote, in general, the dimensions of any feature map. Therefore, although the filter map dimensions will be different depending on the filter size, we will refer to their dimensions by the same name: *N'* and *M'*.

patterns since shorter filters may better fit these patterns. Therefore, in order to efficiently model different time-scales, different filter lengths (n) are set. However, how to set the n's appropriately? We dimensioned them by defining n_O and n_P, that stand for the longer n in *O-net* and *P-net*, respectively. n_O is set to be the slowest (*longest*) onset in the dataset and n_P the slowest (*longest*) pattern. We assume 6 beats to be enough to represent a temporal pattern and therefore, the length of a *P-net* filter is determined by: $n = 1 + 5 \cdot \Delta Fr$ where $\Delta Fr \in \mathbb{Z}$ stands for the number of frames between beats, a frame-based inter-beat interval depending on the tempo (bpm). Note that ΔFr approximates the onset length for a given tempo. Given that the slowest tempo in the dataset is of 60 bpm's [6] and the STFT-spectrogram is computed with a window of 2048 samples (50% overlap) at 44.1 kHz:

$$n_O \equiv \Delta Fr|_{bpm=60} = \frac{44100 \times 60_{(sec)}}{60_{(bpm)} \times 2048 \times 0.5} = 43$$

$$n_P \equiv 1 + 5 \cdot \Delta Fr|_{bpm=60} = 216$$

Note that this result corresponds with the filter lengths proposed for *O-net*: $1 \leq \Delta Fr \leq 8 \equiv 6 \leq n \leq 41 \leq n_O$ and *P-net*: $9 \leq \Delta Fr \leq 43 \equiv n_O \leq 46 \leq n \leq 216 \leq n_P$. As seen, the way we define n_O is arbitrary and depends on the dataset characteristics. Therefore, it could be that for datasets with faster tempos some patterns are learned by *O-net*. However this is not a capacity problem for the model since five filters of equal length are available in *O-net*, what enables learning onsets and patterns simultaneously (if necessary). An alternative way of seeing the design process, that would cope with the issue of *O-net* learning patterns, is to remove the distinction between onsets and patterns. However, we argue that it is interesting to define separately *O-net* and *P-net* since shorter filters are cheaper, what allows adding extra learning capacity (filters) to *O-net* at a low cost. This is why *O-net* (but not *P-net*!) includes 5 filters for each different filter length.

Additionally, note that even though *1-by-n* filters themselves can not learn frequency features, upper layers may be capable of learning frequency cues since the frequency interpretation still holds for the resulting feature map because the convolution operation is done bin-wise (*m=1*). Actually, this observation motivates the sub-band analysis interpretation for the max-pool layer (*4-by-N'*), where the most prominent activations of the 40 bins feature map are summarized in a 10 bands feature map – note that it is common in the MIR literature to do sub-bands analysis for modeling temporal features [10, 9]. One can also note that the max-pool operation picks only the most prominent activation all over the x-axis (*N'*) of the feature map. This has two main advantages: *(i)* although the dimensionality of the feature maps varies depending on the length of the filters, after pooling over *N'* all the feature maps have the same x-axis size – one; and *(ii)* the learnt features are time-invariant [2].

3. EXPERIMENTAL RESULTS

Experiments are realized using the Ballroom dataset that consist on 698 tracks of $\approx$ 30 sec long, divided into 8 music genres [6]. Two main shortcomings are regularly issued against this dataset: *(i)* its small size and *(ii)* the fact that its classes are highly correlated with tempo – although being proposed for evaluating rhythmic descriptors. And precisely, the previously described shortcomings motivate our study. Deep learning approaches rely on the assumption that *large* amounts of training data are available to train the *large* amount of parameters of a network, and the data assumption do not holds for most MIR problems. We want to study if a CNN architecture designed to efficiently represent musical concepts can achieve competitive results in a context where an *small* amount of parameters is trained from a *small* dataset. The Ballroom dataset provides an excellent opportunity for studying so, due to its reduced size and because its classes are highly correlated with short time-scale temporal features (tempo and rhythm). We exploit this *prior* knowledge to propose and assess some *small* efficient musically motivated architectures that might be capable of learning these temporal features.

The audio is fed to the network through fixed-length mel spectrograms, N = 250 frames wide. It is set to 250 in order to fit the longest filter in *P-net*: $n = 216$. Throughout this work we use 40 bands mel-spectrograms derived from a STFT-spectrogram computed with a Blackman Harris window of 2048 samples (50% overlap) at 44.1 kHz. Phases are discarded. A dynamic range compression is applied to the input spectrograms element-wise: $log(1 + C \cdot x)$ where $C = 10.000$ [11]. The resulting spectrograms are normalized so that the whole dataset spectrograms (together) have zero mean and variance one. The activation functions are linear rectifiers (ReLUs) with a final 8-way softmax, where each output unit corresponds to a Ballroom class. 50% dropout is applied to the output layer. The output unit having the highest output activation is selected to be the model's class prediction. Each network is trained using gradient descent with a minibatch size of 50, minimizing the categorical cross-entropy. Networks are trained from random initialization [12] (with the same random seed) using an initial learning rate of 0'01. A learning schedule is programmed: the learning rate is divided by ten every time the training loss gets stacked until there is no more improvement. The best model in the validation set is kept for testing. Mean accuracies are computed using 10-fold cross validation with the same randomly generated train-validation-test split of 80%-10%-10%. Since the input spectrograms are shorter than the total length of the song spectrogram, several estimations for each song can be done – we cut the input spectrograms with overlapping, and the hop-size is set differently depending on the experiment[7]. A simple

[7] Overlapping input spectrograms can be seen as a data augmentation technique. But also note that the smaller the hop size, the more estimations per song are done at test time – what can be useful for the majority vote stage.

Model:	hop	# params	accuracy	Model:	hop	# params	accuracy
O-net	250/80	4,188	76.66/85.24 %	*4x O-net + 4x P-net*	250/80	46,408	88.82/91.55 %
P-net	250/80	7,428	83.95/89.26 %	*8x O-net + 8x P-net*	250/80	92,808	88.68/**92.27** %
2x O-net	250/80	8,368	81.53/86.54 %	*Marchand et al.* [10]	-	-	**96** %
O-net + P-net	250/80	11,608	87.25/89.68 %	*Time* [4]	80	7,336	81.79 %
2x P-net	250/80	14,848	85.67/89.11 %	*Time-freq* [4]	80	196,816	87.68 %
2x O-net + 2x P-net	250/80	23,208	87.25/91.27 %	*Black-box* [4]	80	3,275,312	87.25 %

Table 1. Mean accuracy results comparing different approaches predicting the Ballroom dataset classes. *# params* stands for the number of parameters of the model and *hop* for the hop-size when cutting the input spectrograms with overlapping.

majority vote approach serves to decide the estimated class for each song.

Results are compared with the state-of-the-art of the Ballroom dataset (Marchand *et al.* [10]) and with three deep learning approaches applied to this dataset (*time*, *time-freq* and *black-box* [4]). Marchand *et al.* [10] is based on a scale and shift invariant time/frequency representation that uses auditory statistics, not deep learning. The *time* architecture has a single CNN layer with *1-by-60* filters (one filter shape in a single-layer CNN). *Time-freq* and *black-box* [4] have two layers: CNN + feed-forward and they differ in the filter shape setup of the CNN layer. *Time-freq* uses *1-by-60* and *32-by-1* filters (two different filter shapes in the CNN layer) and *black-box* uses small rectangular filters (one filter shape in the CNN layer). For fair comparison *wrt.* these models, two hop sizes are used: $hop = 250, 80$. When setting $hop = N = 250$, no input spectrograms overlap is used – equally as in [4]. However, the input spectrogram is set smaller ($N = 80$) for the three deep learning approaches [4] and therefore, more training examples are available. In order to have as many training examples as in [4], we also compare our results when $hop = 80$, although overlapping data is used. These two setups ($hop = 250, 80$) provide a fair test-bed to compare our results with the state-of-the-art.

Results are presented in Table 1. First, observe that for $hop = 80$ most of the here presented models (very small and shallow) can achieve better performance than *time*, *time-freq* and *black-box* (having many repeated filter shapes in the first layer). But also observe that *O-net + P-net* architectures, having the most diverse combination of filter shapes, are the best among the presented models. Therefore, these results validate the here proposed design strategy of adding musically motivated filters with different shapes (instead of having the same filter repeated many times) in the first layer. Second, observe that this novel design strategy is specially useful in circumstances when not many training examples are available – observe that for $hop = 250$ bigger accuracy gains are achieved when more different filter shapes are available (*ie.* compare *O-net + P-net* and *2x P-net*). Since adding different filter shapes is cheaper than doubling the capacity of the network, the here proposed design strategy allows increasing the representational power of the first layer at a very low cost. Efficiently using a reduced number of parameters ($92{,}808 \ll 196{,}816 \ll 3{,}275{,}312$) for modeling the main dimensions of a problem is a straight-forward way of fighting overfitting in scenarios where *small* datasets and little computational resources are available – and note that the short time-scale temporal features are the most relevant dimension in the Ballroom dataset. Third, note that models containing *P-net* are the most successful ones. We speculate that this is because the most relevant features in this dataset (rhythm and tempo) can be better encoded in *P-net* than in *O-net*, as we hypothesized during the design process. And fourth, none of the proposed models overcome the result from Marchand *et al.* [10]. However, note the limitations of the here proposed models (basically designed to model short time-scale temporal features and to validate the proposed design strategy): only a limited amount of *1-by-n* filters are used, within a single layer and with a limited amount of data. In future work we plan to increase the representational capacity of the first layer (*ie.* by also using filter shapes designed to model timbre), we want to stack more layers on top of an efficient CNN (*aka.* go deeper, *ie.* with RNNs), and we want to use more data to train the model (*ie.* by setting *hop=5*).

4. CONCLUSIONS

We have presented a novel CNNs design strategy that consists on modeling different (time-scale) contexts within the first layer with different (musically motivated) filter shapes that are intuitively designed to represent (musical) concepts. Our results show that this design strategy is useful for fully exploiting the representational power of the first CNN layer for modeling music, but note that similar reasonings could also be useful for speech, audio or other MIR tasks. These results provide an advance in: *(i)* gaining intuition towards what CNNs learn, *(ii)* efficiently adapting deep learning for MIR and *(iii)* designing networks at a lower cost.

5. ACKNOWLEDGMENTS

We are grateful for the GPUs donated by NVidia. This work is partially supported by the Maria de Maeztu Units of Excellence Programme (MDM-2015-0502).

6. REFERENCES

[1] Keunwoo Choi, George Fazekas, and Mark Sandler, "Automatic tagging using deep convolutional neural networks," in *17th International Society for Music Information Retrieval Conference (ISMIR)*, 2016.

[2] Huy Phan, Lars Hertel, Marco Maass, and Alfred Mertins, "Robust audio event recognition with 1-max pooling convolutional neural networks," *arXiv preprint arXiv:1604.06338*, 2016.

[3] Yoonchang Han, Jaehun Kim, and Kyogu Lee, "Deep convolutional neural networks for predominant instrument recognition in polyphonic music," *arXiv preprint arXiv:1605.09507*, 2016.

[4] Jordi Pons, Thomas Lidy, and Xavier Serra, "Experimenting with musically motivated convolutional neural networks," in *14th International Workshop on Content-Based Multimedia Indexing (CBMI)*. IEEE, 2016.

[5] Jan Schlüter and Sebastian Böck, "Improved musical onset detection with convolutional neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014.

[6] Fabien Gouyon, Simon Dixon, Elias Pampalk, and Gerhard Widmer, "Evaluating rhythmic descriptors for musical genre classification," in *Proceedings of the 25th AES International Conference*, 2004, pp. 196–204.

[7] Sebastian Böck, Florian Krebs, and Gerhard Widmer, "Accurate tempo estimation based on recurrent neural networks and resonating comb filters," in *16th International Society for Music Information Retrieval Conference (ISMIR)*, 2015.

[8] Florian Krebs, Sebastian Böck, Matthias Dorfer, and Gerhard Widmer, "Downbeat tracking using beat-synchronous features and recurrent neural networks," in *17th International Society for Music Information Retrieval Conference (ISMIR)*, 2016.

[9] Simon Durand, Juan P Bello, Bertrand David, and Gaël Richard, "Feature adapted convolutional neural networks for downbeat tracking," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016.

[10] Ugo Marchand and Geoffroy Peeters, "The modulation scale spectrum and its application to rhythm-content description," in *IEEE International Workshop on Machine Learning for Signal Processing*, 2016.

[11] Sander Dieleman and Benjamin Schrauwen, "End-to-end learning for music audio," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *The IEEE International Conference on Computer Vision (ICCV)*, 2015.

7. APPENDIX

This appendix is not included in the proceedings of the ICASSP 2017 conference due to space constraints.

Fig. 1 depicts *O-net* + *P-net* architecture. Remember that for every different filter shape in *O-net*, *5x* filters are available. However, *P-net* filters have only one filter available for each filter shape - due to its higher cost: more parameters are needed to represent a longer temporal context. Also note that not all *P-net* filters are represented in Fig. 1. However, the three dots in *P-net* are depicting that the length of filters is increasing following the same logic.

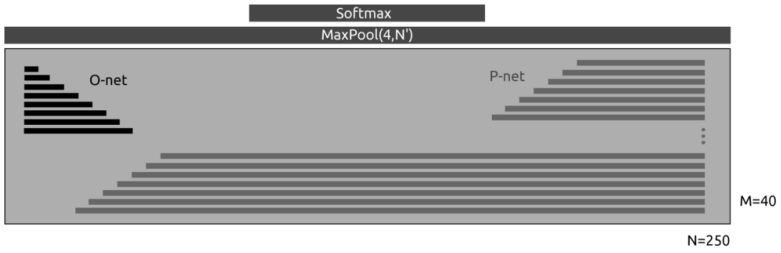

Fig. 1. *O-net* + *P-net* architecture.