MUSIC GENRE RECOGNITION WITH DEEP NEURAL NETWORKS

Albert Jiménez, Ferran José

Universitat Politècnica de Catalunya

ABSTRACT

We discuss the application of convolutional neural networks and convolutional recurrent neural networks for the task of music genre classification. We focus in the case of a low-computational and data budget where we cannot afford to train with a large dataset. We start using a well-known architecture in the field and we use transfer learning techniques to adapt it to our task. Different strategies for fine-tuning, initializations and optimizers will be discussed to see how to obtain the model that fits better in the music genre classification. Moreover, we introduce a multiframe approach with an average stage in order to analyze in detail almost the full song. It is used at training time to generate more samples and at test time to achieve an overview of the whole song. Finally, we evaluate its performance both in a handmade dataset and in the GTZAN dataset, used in a lot of works, in order to compare the performance of our approach with the state of the art.

Index Terms— Music genre classification, recurrent neural networks, convolutional neural networks

1. INTRODUCTION

Music genres are a set of descriptive keywords that convey high-level information about a music clip (jazz, classical, rock...). Genre classification is a task that aims to predict music genre using the audio signal. Being able to automatize the task of detecting musical tags allow to create interesting content for the user like music discovery and playlist creations, and for the content provider like music labeling and ordering.

Building this system requires extracting acoustic features that are good estimators of the type of genres we are interested, followed by a single or multi-label classification or in some cases, regression stage. Conventionally, feature extraction relies on a signal processing front-end in order to compute relevant features from time or frequency domain audio representation. The features are then used as input to the machine learning stage. However, it is difficult to know which features are be the most relevant to perform each task. The recent approaches using Deep Neural Networks (DNNs), unify feature extraction and decision taking. Thus allow learning the relevant features for each task at the same time that the system is learning to classify them.

Several DNN-related algorithms have been proposed for automatic music tagging. In [1] and [2], spherical k- means and multi-layer perceptrons are used as feature ex- tractor and classifier respectively. Multi-resolution spectrograms are used in [1] to leverage the information in the audio signal on different time scales. In [2], pretrained weights of multilayer perceptrons are transferred in order to predict tags for other datasets. A two-layer convolutional network is used in [3] with mel-spectrograms as well as raw audio signals as input features.

Our paper is organized as it follows: in Section 1 we introduce the topic of music genre classification, in Section 2 we review two state-of-the-art network architectures, in Section 3 an explanation on how we train them to adapt them for the new task is provided, and in Sections 4 and 5 we explain the experiments done and the final conclusions.

2. CNNS AND CRNN FOR MUSIC CLASSIFICATION

As most of the works, we are using the mel-spectograms of the music signals as an input to our system. For this reason, we focus on neural networks that have been designed to cope with images.

2.1. Convolutional Neural Networks

Convolutional neural networks (CNNs) have been actively used for various music classification tasks such as music tagging [3] [4], genre classification [5] [6], and user-item latent feature prediction for recommendation [7]. CNNs assume features that are in different levels of hierarchy and can be extracted by convolutional kernels. The hierarchical features are learned to achieve a given task during supervised training. For example, learned features from a CNN that is trained for genre classification exhibit low-level features (e.g., onset) to high-level features (e.g., percussive instrument patterns) [8].

2.2. Recurrent Convolutional Neural Networks

Recently, CNNs have been combined with recurrent neural networks (RNNs) which are often used to model sequential data such as audio signals or word sequences. This hybrid model is called a convolutional recurrent neural network

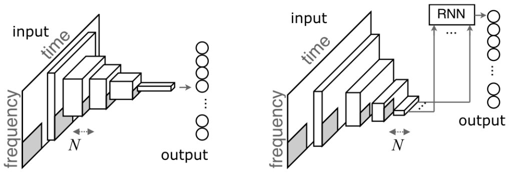
(CRNN). A CRNN can be described as a modified CNN by replacing the last convolutional layers with a RNN. In CRNNs, CNNs and RNNs play the roles of feature extractor and temporal summariser, respectively. Adopting an RNN for aggregating the features enables the networks to take the global structure into account while local features are extracted by the remaining convolutional layers. This structure was first proposed in [9] for document classification and later applied to image classification [10] and music transcription [11]. CRNNs fit the music tagging task well. RNNs are more flexible in selecting how to summarise the local features than CNNs which are rather static by using weighted average (convolution) and subsampling.

3. PROPOSAL

3.1. Network architectures

In this work we will take as a starting point the architectures of a CNN and the CRNN proposed by Choi et al. in [4], [12].

The first one is a fully convolutional neural network with 5 convolutional layers of 33 kernels and max-pooling layers ((2 × 4)-(2 × 4)-(2 × 4)-(3 × 5)-(4 × 4)) as illustrated in Figure 1a. The network reduces the size of feature maps to 1 × 1 at the final layer, where each feature covers the whole input. This model allows time and frequency invariances in different scale by gradual 2D sub-samplings. Being fully-convolutional reduces considerably the number of parameters.



(a) CNN Architecture (b) CRNN Architecture

Fig. 1: Network Architectures from [4] and [12]

The second architecture uses a 2-layer RNN with gated recurrent units (GRU) [13] to summarize temporal patterns on the top of two- dimensional 4-layer CNNs as shown in Figure 1b. The assumption underlying this model is that the temporal pattern can be aggregated better with RNNs then CNNs, while relying on CNNs on input side for local feature extraction. In CRNN, RNNs are used to aggregate the temporal patterns instead of, for instance, averaging the results from shorter segments as in [3] or convolution and sub-sampling as in other CNNs. In its CNN sub-structure, the sizes of convolutional layers and max-pooling layers are 33 and (2 × 2)-(3 × 3)-(4 × 4)-(4 × 4). This sub-sampling results in a feature map size of $N \times 1 \times 15$ (number of feature maps x frequency x time). They are then fed into a 2-layer RNN, of which the last hidden state is connected to the output of the network.

3.2. Transfer Learning

Transfer learning [14] has proven to be very effective in the image processing scene, it studies and provides techniques on how adapt a model trained in large-scale database [15] to perform well in other tasks different from the one that was trained for as in [16], [17].

We aim at learning from a source data distribution (multi-class tags) a well performing model on a different target data distribution (single class genres). Inside the transfer learning paradigm this is known as domain adaptation. The two most common practices and the ones that we will apply are:

Using the network as feature extractor. That is removing the last fully-connected layer and treat the network as a feature extractor. Once we have extracted all the features at the top we can include a classifier like SVM or a Softmax classifier for the new dataset.

Fine-tuning the network. This strategy is based on not only replace the classifier layer of the network, but also retrain part or the whole network. Through backpropagation we can modify the weights of the pre-trained model to adapt the model to the new data distribution. Sometimes its preferable to keep the first layers of the network fixed (or freezed) to avoid overfitting, and only fine-tune the deeper part. This is motivated because the lower layers of the networks capture generic features, that are similar to many tasks while the higher layers contain features that are task and dataset oriented as demonstrated in [18].

3.3. Multiframe

We propose to use a multiframe strategy that allows us to extract more than one frame (or mel-spectogram image) per song. For each song we discard the first and last N seconds and then, we divide the rest into frames of equal time-length t. The final parameters are stated in the experiments. This approach has two advantages:

At training time: we are able to generate more data to train the network than in the approach of [4], as they are only extracting the central part of the song. We are very limited by the number of songs, this is a very useful tool to provide data augmentation.

At test time: we can average or perform a KNN with the scores of every frame to infer the genre tag for the complete song with more confidence.

4. EXPERIMENTS

4.1. Frames Acquisition and Evaluation

In [4] only one frame of 29.12s per song is extracted. Specifically, this frame contains the central part of the song as it should be the most representative. Then, a log-amplitude mel-spectogram is extracted using 96 mel-bins and a hop-size of 256 samples, resulting in an input shape of 96 × 1366. In reference to our multiframe approach, we divide the song in a set of frames of also 29.12s. Therefore, at each frame a mel-spectrogram can be extracted using the same parameters, and thus the same resolution that in [4].

In order to select the 29.12s that compound each frame, two different steps are carried out. Firstly, a short period of the song is removed both in the beginning and in the end. These parts of the songs are hardly ever representative. Therefore, their inclusion to the classification procedure would lead to weaker results. The following step consists in dividing the remaining part of the song in frames of 29.12s. They are not overlapped and the last frame is also removed if its duration is lower than 29.12s.

Finally, once obtained all the frames, the evaluation criterion can be set at frame level or at song level. We use the accuracy metric to measure the performance of our system. If we evaluate at song level, we propose to use an averaging of the predicted tags for each frame of the song. In order to do that, the mean among the tags score of all the frames of the song is computed and the tag with the highest score is selected. Therefore, if a song contains a small period that can be classified as a different genre, it will not affect the final song classification. Another approach that would lead to a unique tag per song would be the nearest neighbors algorithm between the tags of all the frames of the song. Therefore, the most repeated tag among all the frames of the song would be selected as the tag of the song.

4.2. GTZAN dataset

GTZAN is a dataset created by Tzanetakis et al[[19]]. It is compounded of 1000 music excerpts of 30 seconds duration with 100 examples in each of 10 different music genres: Blues, Classical, Country, Disco, Hip Hop, Jazz, Metal, Popular, Reggae, and Rock. The files were collected in 2000-2001 from a variety of sources including personal CDs, radio, microphone recordings, in order to represent a variety of recording conditions. And they are all 22050Hz Mono 16-bit audio files in .wav format.

However, regarding our approach it has one important limitation. The duration of each music excerpt (30s) makes that only one frame per song can be extracted and without discarding anything neither at the begging nor in the end of the song. Therefore, although this dataset has been useful to compare the transfer learning performance with other works, it cannot be applied to evaluate the multiframe approach, as in this case more than one frame per song is needed.

4.3. Handmade dataset

In order to evaluate the performance of the multiframe approach, we built a dataset with the genres of the GTZAN dataset, but using longer songs. Specifically, our dataset is compounded by 300 music excerpts with 30 examples for each of 10 GTZAN genres. To be coherent, the genre-song association has been taking using Spotify lists [20].

As we use whole songs, the number of frames per song is variable, leading to 4 frames per song in the shortest songs and more than 10 in the opposite case (e.g. classical songs). Furthermore, to process this dataset we have trimmed both the begging and the end of each song. Specifically, 20s of each extreme are removed as empirically we have seen that these parts are non-representative of the song content most of the time. We have divided the dataset into train subset, 20 songs per genre or 1468 frames and test with 10 songs per genre or 747 frames.

4.4. Training

As we have stated before, we have fine-tuned two different networks, a CNN and a CRNN. We have made experiments by freezing the lowest layers and fine-tuning different top layers to see the differences. The parameters have been set as in standard fine-tuning, setting the learning rate a bit slower than in the original model. We have tried two different optimizers, the adaptative learning rate ADAM method [21] as in the original model work [4] and SGD with Nesterov Momentum [22] as it is widely used in machine learning. We set categorical cross-entropy as the loss function. Batch normalization and dropout layers are implemented as the original authors did. To perform the training we use all of our handmade dataset as training data (2215 frames) and the GTZAN dataset as testing data (1000 frames).

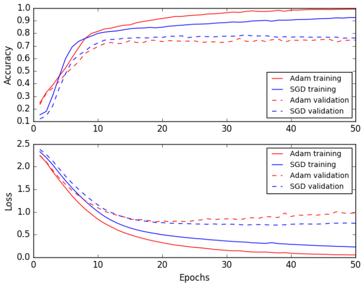
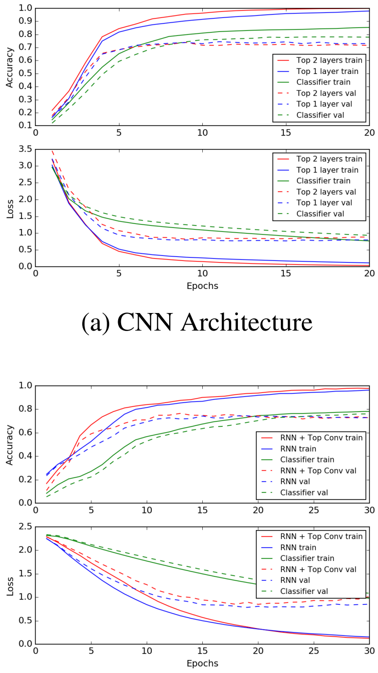


Fig. 2: Adam vs SGD

We can observe in figure 2 that ADAM method converge faster than SGD. However, the plot shows that SGD is generalizing a bit better, although both are reaching overfitting in approximately 25 epochs. We decide to use ADAM to perform the rest of experiments.

In figure 3 we show the comparison of fine-tuning both architectures. For the CNN case, we can observe that it gets overfitted very fast, in less than 10 epochs. The best results are produced when we only train a classifier layer in the top of the network. However as we can see in the plot, the network is getting stalled and it cannot improve more because of the lack of data. On the other hand, with the CRNN, we can see the same behavior, if we fine-tune the recurrent layers and the last convolutional layer, the network is getting overfitted very fast. As in this case the network complexity is increased it improves more before convergence. If only a classifier is trained we observe the same as in the CNN case, it needs more data to improve the performance and not getting stalled. The best results are around 78%.



(a) CNN Architecture

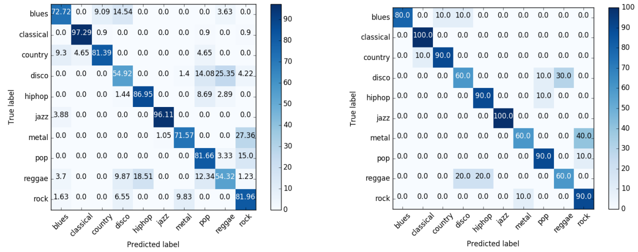
(b) CRNN Architecture

Fig. 3: Fine-tuning results for the different architectures

4.5. Results

The model to evaluate the performance of our multiframe approach has been trained with the train partition of our handmade dataset and evaluated using the test partition. The genre prediction has been made using the recurrent layers fine-tuned CRNN model with a total accuracy of 77.89%. Furthermore, the confusion matrix between the real genre and the predicted genre has been built for two different scenarios: using the predicted tag of each frame of the test database, and using the predicted tag of the average score obtained for each song of the test database. Thus allowing an evaluation of the average stage improvement.

Figure 4 shows both matrices. The results are expressed in percentage, from 0 to 100%, in such a way that at each row (true label), we have the distribution of the predicted tags for the corresponding genre. Therefore, the most diagonal is the confusion matrix, the better is the classification performance. As can be seen, both of the obtained matrices are quite diagonal. Therefore, we can conclude that the resulting model works well in almost all of the 10 genres. The weaker results have been obtained in disco, metal and reggae, which is reasonable as they are the less distinguished genres. Metal can be confused by rock in some songs, the same with reggae and hiphop, and finally disco is a genre that can be understood as a mix of other genres.



	blues	classical	country	disco	hiphop	jazz	metal	pop	reggae	rock
blues	72.72	0.0	9.09	14.54	0.0	0.0	0.0	0.0	3.63	0.0
classical	0.0	97.29	0.9	0.0	0.0	0.0	0.0	0.9	0.0	0.9
country	9.3	4.65	81.39	0.0	0.0	0.0	0.0	4.65	0.0	0.0
disco	0.0	0.0	0.0	54.92	0.0	0.0	1.4	14.08	25.35	4.22
hiphop	0.0	0.0	0.0	1.44	86.95	0.0	0.0	8.69	2.89	0.0
jazz	3.88	0.0	0.0	0.0	0.0	96.11	0.0	0.0	0.0	0.0
metal	0.0	0.0	0.0	0.0	0.0	1.05	71.57	0.0	0.0	27.36
pop	0.0	0.0	0.0	0.0	0.0	0.0	0.0	81.66	3.33	15.0
reggae	3.7	0.0	0.0	9.87	18.51	0.0	0.0	12.34	54.32	1.23
rock	1.63	0.0	0.0	6.55	0.0	0.0	9.83	0.0	0.0	81.96

(a) Using all the frames

	blues	classical	country	disco	hiphop	jazz	metal	pop	reggae	rock
blues	80.0	0.0	10.0	10.0	0.0	0.0	0.0	0.0	0.0	0.0
classical	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
country	0.0	10.0	90.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
disco	0.0	0.0	0.0	60.0	0.0	0.0	0.0	10.0	30.0	0.0
hiphop	0.0	0.0	0.0	0.0	90.0	0.0	0.0	10.0	0.0	0.0
jazz	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0
metal	0.0	0.0	0.0	0.0	0.0	0.0	60.0	0.0	0.0	40.0
pop	0.0	0.0	0.0	0.0	0.0	0.0	0.0	90.0	0.0	10.0
reggae	0.0	0.0	0.0	20.0	20.0	0.0	0.0	0.0	60.0	0.0
rock	0.0	0.0	0.0	0.0	0.0	0.0	10.0	0.0	0.0	90.0

(b) Using the mean

Fig. 4: Confusion matrices

Regarding the improvement of the implementation of the average stage, the diagonal elements after using the average stage have increased in all the genres with the exception of metal. Nevertheless, the average accuracy, computed as the mean of the diagonal elements, has been increased from 77.89% to 82%. Therefore, we can conclude that the implementation of the average stage outperforms the case where only one frame per song is selected.

5. CONCLUSIONS

We explore the application of CNN and CRNN for the task of music genre classification focusing in the case of a low-computational and data budget.

The results have shown that this kind of networks need large quantities of data to be trained from scratch. In the scenario of having a small dataset and a task to perform, transfer learning can be used to fine-tune models that have been trained on large datasets and for other different purposes.

We have shown that our multiframe approach with an average stage improves the single-frame song model. In the experiments, a homemade dataset compounded by songs longer than our frame duration has been used. These songs belong to 10 different genres and the experiments have revealed that the average stage achieves better results in 9 of these 10 genres and a higher total accuracy. Therefore, using the average stage we are able to remove the non-representative frames dependence.

As a future work, some other techniques to obtain a single genre tag per song from multiple frames can be analyzed. For instance, knn or geometric mean.

6. REFERENCES

[1] Sander Dieleman and Benjamin Schrauwen, "Multi-scale approaches to music audio feature learning," in *14th International Society for Music Information Retrieval Conference (ISMIR-2013)*. Pontifícia Universidade Católica do Paraná, 2013, pp. 116–121.

[2] Aäron Van Den Oord, Sander Dieleman, and Benjamin Schrauwen, "Transfer learning by supervised pre-training for audio-based music classification," in *Conference of the International Society for Music Information Retrieval (ISMIR 2014)*, 2014.

[3] Sander Dieleman and Benjamin Schrauwen, "End-to-end learning for music audio," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 6964–6968.

[4] Keunwoo Choi, George Fazekas, and Mark Sandler, "Automatic tagging using deep convolutional neural networks," *arXiv preprint arXiv:1606.00298*, 2016.

[5] Keunwoo Choi, George Fazekas, Mark Sandler, and Jeonghee Kim, "Auralisation of deep convolutional neural networks: Listening to learned features," in *Proceedings of the 16th International Society for Music Information Retrieval Conference, ISMIR*, 2015, pp. 26–30.

[6] Paulo Chiliguano and Gyorgy Fazekas, "Hybrid music recommender using content-based and social information," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 2618–2622.

[7] Aaron Van den Oord, Sander Dieleman, and Benjamin Schrauwen, "Deep content-based music recommendation," in *Advances in Neural Information Processing Systems*, 2013, pp. 2643–2651.

[8] Keunwoo Choi, George Fazekas, and Mark Sandler, "Explaining deep convolutional neural networks on music classification," *arXiv preprint arXiv:1607.02444*, 2016.

[9] Duyu Tang, Bing Qin, and Ting Liu, "Document modeling with gated recurrent neural network for sentiment classification," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1422–1432.

[10] Zhen Zuo, Bing Shuai, Gang Wang, Xiao Liu, Xingxing Wang, Bing Wang, and Yushi Chen, "Convolutional recurrent neural networks: Learning spatial dependencies for image representation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2015, pp. 18–26.

[11] Siddharth Sigtia, Emmanouil Benetos, and Simon Dixon, "An end-to-end neural network for polyphonic piano music transcription," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 5, pp. 927–939, 2016.

[12] Keunwoo Choi, George Fazekas, Mark Sandler, and Kyunghyun Cho, "Convolutional recurrent neural networks for music classification," *arXiv preprint arXiv:1609.04243*, 2016.

[13] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[14] Sinno Jialin Pan and Qiang Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[16] Matthew D Zeiler and Rob Fergus, "Visualizing and understanding convolutional networks," in *European Conference on Computer Vision*. Springer, 2014, pp. 818–833.

[17] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 806–813.

[18] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson, "How transferable are features in deep neural networks?," in *Advances in neural information processing systems*, 2014, pp. 3320–3328.

[19] George Tzanetakis and Perry Cook, *Manipulation, analysis and retrieval systems for audio signals*, Princeton University Princeton, NJ, USA, 2002.

[20] Spotify, "Spotify genres, the full listing," `http://news.spotify.com/us/2009/03/24/spotify-genres-the-full-listing`, 2009.

[21] Diederik Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[22] Léon Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*, pp. 177–186. Springer, 2010.