A neural network for composer classification

Gianluca Micchi

Late-Breaking / Demos Session Abstract for the
2018 International Society for Music Information Retrieval Society Conference

A NEURAL NETWORK FOR COMPOSER CLASSIFICATION

Gianluca Micchi
CRIStAL, UMR 9189, CNRS, Université de Lille, France

ABSTRACT

I present a neural network approach to automatically extract musical features from 20-second audio clips in order to predict their composer. The network is composed of three convolutional layers followed by a long short-term memory recurrent layer. The model reaches an accuracy of 70% on the validation set when classifying amongst 6 composers. The work represents the early stage of a project devoted to automatic feature detection and visualization.

1. INTRODUCTION

The field of computational musicology often demands elaborated features to be analysed in order to answer conceptually simple questions like "Who composed this piece?" As an example, melodic lines, rhythmic pattern, chords and chord progessions, tonality, and cadenzas are used. The creation of algorithms detecting these features on symbolic data is possible but difficult and requires lots of knowledge from experts [1, 7]. On audio recordings, such an algorithm is even harder to implement.

However, in the last ten years, research in machine learning has allowed for several attempts at recreating meaningful audio features in an automatic way, for example with Deep Belief Networks (DBN) [5, 6] and variants of Convolutional Neural Networks (CNN) [2, 3]. The first achieve good results but are hard to train and to optimize since they are completely unsupervised. Classification through CNN has even better results but suffers from the opposite: the structure is typically built with a certain number of composers in mind and it is hard to adapt the results to new composers.

In this work, I give my contribution to the second approach and, similar to what Choi et al. did [3], I train a convolutional neural network followed by a recurrent neural network. However, in this case, I use a different database and I train specifically for the task of composer recognition in Western classical music.

2. THE DATA

The data is taken from audio recordings of piano music. The corpus gather 320 pieces from 21 albums featuring 6 Western classical composers: Bach, Beethoven, Schubert, Schumann, Chopin, Fauré. 20-seconds clips have been randomly extracted from each recording. The total length of these clips is about the length of each piece but, as the extraction is random, some clips may be overlapping.

On each of the audio clips, a short-time Fourier transform (STFT) analysis is performed. To do that, a Hamming window of size n_w = 2 001 has been chosen (main lobe of size n_l = 4) that, together with the sampling rate of the signal being fixed at f_s = 44 100 Hz, implies a frequency resolution $\Delta f = n_l * f_s/n_w \approx 88$ Hz. Conversely, the time resolution is $\Delta t = n_w/f_s \approx 45$ ms. The STFT has been zero-padded to 2048 to take advantage of numeric optimization. Only the magnitude of the signal for frequencies between 0 and 5 000 Hz is kept and the result plotted as a gray-scale image. This yields images with dimensions 881 (time) $\times$ 233 (frequency).

I decided to keep the magnitude in the linear scale and not in the logarithmic scale in order to have input data that facilitates the identification of the fundamental frequencies of the notes played. In such a way, the data represents the written notation of the music more faithfully, being less influenced by the acoustic characteristics of the musical instruments used to record the song.

To reduce overfitting, the data is divided at the song level in a training set (90% of the songs) and a validation set (10% of the songs). When the model is applied to the validation set, it thus predicts the composer of songs it has never listened to before. More effort should be done on this subject, in particular to overcome the *album effect* for which a classifier classifies the songs not based on the composer but rather on audio features of the album [4].

3. THE MODEL

Figure 1 shows the architecture of the model. The first part of the neural network is made of three convolutional layers, each followed by a pooling layer and a batch normalization. The shape of both the convolutional filters and the pool in the pooling layers is rectangular to better adapt to the high aspect ratio of the data images.

At the end of the convolutional neural network, the data is reduced to dimension (14, 30, 16), where the last dimension is given by the number of filters. We interpret it as a succession of 14 features of dimension (30, 16) and feed it to a standard long short-term memory recurrent neural network (LSTM-RNN) with 6 units, each representing the probability that the piece is by a certain composer.

As a loss function, we choose a standard cross entropy with softmax for a multiclass classification.

	Ba	Be	Sb	Sm	Ch	Fa
Bach (Ba)	**25**	0	1	6	4	0
Beethoven (Be)	2	**28**	12	1	12	0
Schubert (Sb)	0	0	**45**	5	7	0
Schumann (Sm)	4	0	0	**19**	20	1
Chopin (Ch)	0	13	8	14	**55**	5
Fauré (Fa)	2	2	0	2	5	**6**

Table 1. Confusion matrix. The first row means that Bach was correctly recognized as Bach 25 times, never mistaken for Beethoven or Fauré, and misclassified once as Schubert, 6 times as Schumann, and 4 times as Chopin.

4. RESULTS

The program is written in TensorFlow (`tensorflow.org`). The classification accuracy is up to 70% on the validation set (Figure 2), far better than random (17%). A different split of the validation and training set yields slightly different results, with the worst run still achieving 60% accuracy. In all cases, the loss on the validation set decreases almost monotonically, indicating that probably no overfitting takes place (data not shown).

A confusion matrix can help understand the results a little better. Table 1 shows it in the worst accuracy case. Errors were made especially between Chopin and Schumann, two composers from the romantic period, stylistically closer to each other than to the others. It is also interesting to notice that Beethoven was often mistaken as Schubert but never the opposite.

5. CONCLUSIONS AND PERSPECTIVES

I have shown a neural network structure made of a CNN connected to a RNN for automatic composer classification. The network shows a promising 70% accuracy on the validation set when tested with 6 composers with little hyperparameter optimization.

A first perspective for future work is to use a triplet loss function to recognize if two songs are composed by the same person. This would allow one to remove the constraint of the fixed number of composers. Composer

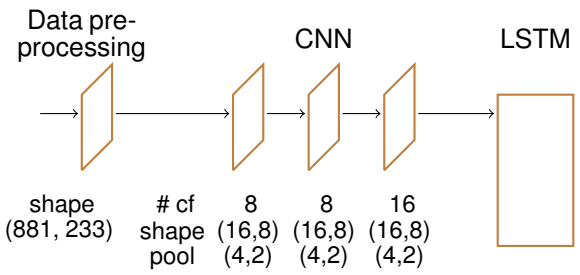

Figure 1. The model. For the CNN, the three rows of data represent the number of convolutional filters, their shape, and the shape and stride of the pooling layers.

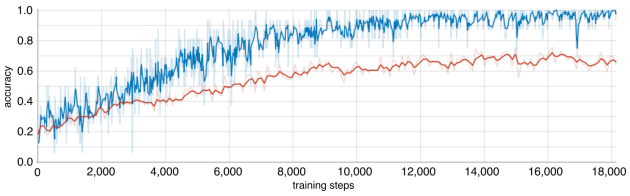

Figure 2. Evolution of the accuracy during the training, as displayed with TensorBoard. After about 14 000 steps, which correspond to 150 epochs, accuracy on the validation set reaches a plateau at about 70%.

recognition, rather than classification, could still extract useful musical features from the input data while retaining a larger flexibility. A second possible direction is to recreate an evolutionary tree of music based on the learned fetures and to compare it to the known results from musicology.

I acknowledge Mathieu Giraud for useful discussions.

6. REFERENCES

[1] Louis Bigo, Mathieu Giraud, Richard Groult, Nicolas Guiomard-Kagan, and Florence Levé. Sketching sonata form structure in selected classical string quartets. In *International Society for Music Information Retrieval Conference (ISMIR 2017)*, 2017.

[2] Keunwoo Choi, George Fazekas, and Mark Sandler. Automatic tagging using deep convolutional neural networks. *International Society for Music Information Retrieval Conference*, pages 805–811, jun 2016.

[3] Keunwoo Choi, Gyorgy Fazekas, Mark Sandler, and Kyunghyun Cho. Convolutional Recurrent Neural Networks for Music Classification. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2392–2396. IEEE, 2017.

[4] Arthur Flexer and Dominik Schnitzer. Effects of Album and Artist Filters in Audio Similarity Computed for Very Large Music Databases. *Computer Music Journal*, 34(3):20–28, sep 2010.

[5] Philippe Hamel and Douglas Eck. Learning Features from Music Audio with Deep Belief Networks. *International Society for Music Information Retrieval Conference (ISMIR)*, (Ismir):339–344, 2010.

[6] Honglak Lee, Pt Pham, Yan Largman, and Ay Ng. Unsupervised feature learning for audio classification using convolutional deep belief networks. *Nips*, pages 1–9, 2009.

[7] Cory McKay, Julie Cumming, and Ichiro Fujinaga. JSymbolic 2.2: Extracting features from symbolic music for use in musicological and MIR research. In *International Society for Music Information Retrieval Conference (ISMIR 2018)*, 2018.