Rocco Clemente Bonjour

Josep Torné Chertó

Music genre detection using deep learning

Motivation

The recent years the music industry have suffered a huge transformation due to the appearance of audio streaming platforms such as Spotify or even YouTube. Millions of users are registered in these applications, so it is widely known the huge amount of money that has been invested in this area. These companies offer recommendations to their clients based on the music they listen to and create lists with similar music.

Music genre detection is probably the first step for a recommendation system or an automatic list creator, so we have thought that the implementation of an algorithm that satisfies this task will be interesting. The use of deep learning algorithms has been proved as a good solution for classifying problems so we have decided that it would be interesting to merge this approach with this problem.

Finally, the parametrization of audio signal and the processing with neural networks will be useful to settle some of the knowledge acquired during the course.

Objective

Basically, the main goal of this project is to obtain a music classification system that uses Deep Learning algorithms with a pretty good accuracy. Ideally, the desired accuracy is about 90%.

With the use of a database in order to train and test the system, two approaches have been developed in order to achieve the proposed objective. The approaches goes from low to high complexity and from low to great results too. In the first one, a simple CNN has been implemented. In the second one, a RNN has been added to the previous CNN in order to improve the results.

Apart from that, the features extracted from database that act as input data for both approaches have been carefully chosen, taking into account previous work from other researchers on this field.

State of the art

The first significant work related to music genre classification was done by George Tzanetakis [1] in 2002. In this paper he extracted different features related to timbral texture, rhythm and pitch content. After getting them, a statistical pattern recognition (SPR) was used in order to make the classification.

Since then, different features inside the groups mentioned above and different algorithms based on machine learning have been used offering better performance with the same database.

The recent years, the increasing popularity of deep learning has produced that the neural networks have been used for this purpose offering better result than the previous approaches and without needing to select the features.

Selection of the database

Obtaining big and well databases in order to solve this problem has been troublesome due to the intellectual property rights the music have.

In previous works, the database GTZAN created by George Tzanetakis has been the most used so we have decided to use the same one in order to compare the results.

This database contains 1000 songs of 30 seconds equally divided in 10 music genres. The format is a 16-bit wav.

The database can be downloaded from the following link:

http://marsyas.info/downloads/datasets.html

Selection of the input of the neural network

Each song is composed for around 30 seconds sampled with 16 bits at a rate of a 22050 Hz. This is a lot of information for being introduced into a neural network so is needed to use another representation of this signal.

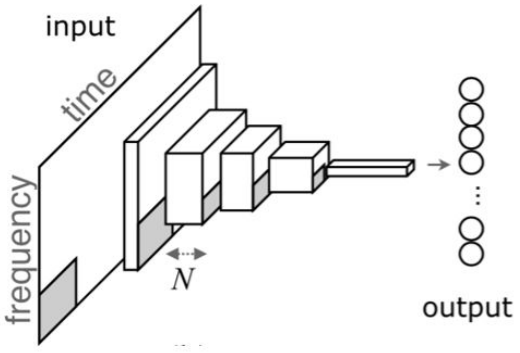
The Mel-Spectrogram consists in compute the spectrogram of the signal and summarize the power in the Mel bands that are similar to the acoustic human system.

This reduces significantly the data and represents the data in 2D that is useful to treat the problem as an image classification problem that has been widely studied in image processing.

Finally, for computing the Mel-Spectrogram, 29,12 seconds of the song had been used resulting in a 96x1366 matrix that will be as the input of the network.

Selection of the algorithm for the neural network and its parametrization

As a first approach we have selected the design defined by Choi et al. [2] consisting in a 5 layer CNN neural network:



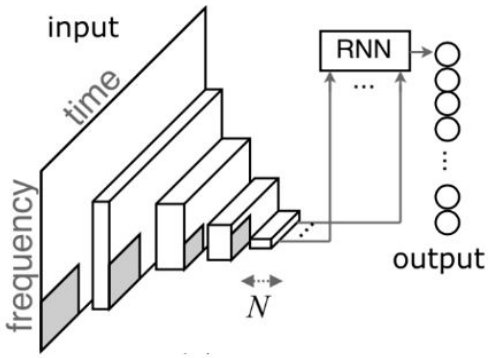
All the 2D convolutional layers have a kernel of size 3 and RELU activation. The hidden size of the nodes in each of this layers is as following 32,64,64,64,32.

Between these layer is a max pooling layer ((2×4)-(2×4)-(2×4)-(3×5)-(4×4)) that ends reducing the size of feature map to 1x1.

Finally the network has flatten layer that transform the output to one dimension and a dense layer with a softmax activation of size 10 (genres we analyze).

The first results we observed denote that the network had some overfitting so drope layers were added after each pooling layer with a rate of 0,1.

Another approach was also implemented consisting in a CRNN also mentioned in Choi et. al [3] paper:



The structure of this network consists in a CNN similar to the previous one but with 4 convolutional layers instead of 5 and 4 max pooling layers ((2×2)(3×3)-(4×4)-(4×4)).

The output is connected to a RNN consisting in two GRU layers of size 32 and a final dropout layer of rate 0.3. We didn't manage to implement correctly this scheme due that the validation and test accuracy didn't improve in function of epochs. We thought that this could be a local minimum so we used an SGD optimizer with different learning rates but the result was similar.

Results analysis

In the following table we can see the results obtained for the different architectures used and the results obtained by previous works with the same databaze:

Tzanetakis	61.00%
Holzapfel	74.00%
Benetos	75.00%
Lidy	76.80%

Bergstra	82.50%
CNN	**52.00%**
CNN with dropping layers	**62.00%**

It is observed that using a CNN without dropout the results were pretty bad compared to the previous work done with only machine learning. That is caused due to an overfitting of the network produced by using an small database.

In order to solve this problem 4 dropout layers with 0,1 after each max pooling layer were added and an improvement appeared.

There are no results for the CRNN due to its implementation wasn't done correctly and its accuracy was 10%, equivalent to a random selection accuracy.

As conclusion can be said that the results are worse than the expected and we think that the main reason is due to the use of a small database. This could be solved by using a trained model and retrain only the last layers.

References

[1]: A. Tzanetakis, G. and Cook, P. "Musical genre classification of audio signal", IEEE Transactions on Speech and Audio Processing,Vol. 10, No. 3, pp. 293-302, July 2002.

[2]: Keunwoo Choi, George Fazekas, Mark Sandler, P. "Automatic Tagging using Deep Convolutional Neural Networks", 17th International Society for Music Information Retrieval Conference, New York, USA, 2016

[3]: Keunwoo Choi, György Fazekas, Mark Sandler, and Kyunghyun Cho. "Convolutional recurrent neural networks for music classification". In 2017 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2017.