Neural Networks for Audio Multi-Label Classification

Victor Pascuo Celva pascuocelva.1841242@studenti.uniroma1.it

Abstract

This work searches to reproduce the idea of Neural Networks for Audio Classification from Adams (2019) repository adapted to an audio multi-genre classification problem, using as dataset the whole library of a music production company that produces album for TV services.

1. Motivation

The idea for the present work came due to a part time job that I do as data curator for a label that offers music basically for TV services. As daily work we need to classify genres, moods, categories and instrumentations of many audio tracks.

It's common in this segment that companies from different countries create partnerships among them, i.e. a company from Italy can stablish a partnership with a company in UK, so the British company starts to represent the music library affairs from the Italian company in UK. It is common that companies use different procedures to classify their music and offer to clients, so basically companies that "export" their music have to adapt the metadata from their library in order to fit in the procedure of the company that will represent them.

These procedures of audio classification can represent a lot of time spent and nowadays are normally mechanically done by a person that has to listen all tracks many times, besides being a process that can cause hearing fatigue, it's also hard sometimes to keep a good classification standard when evaluating many tracks.

As a first step trying to make the whole process of audio classification easier and faster, this work tries to use a model of Convolutional Neural Networks to make multi-label classification of audio tracks, using as dataset the whole library available from the company.

2. Audio Genre Classification

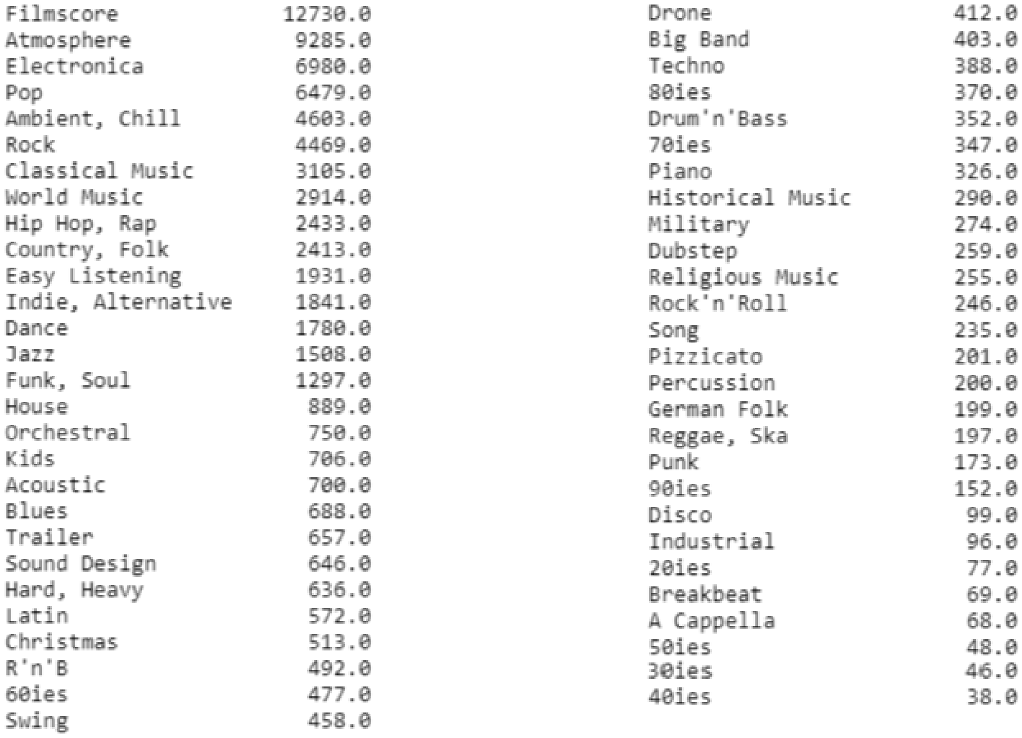
Audio genre classification is being widely studied field along the last 20 years. Tzanetakis and Cook (2002) approached this problem using supervised machine learning as Gaussian Mixture model and k-nearest neighbour classifier, using three sets of features for different tasks categorized as timbral structure. Mel-frequency cepstral coefficients (MFCCs), spectral contrast and spectral roll-off were some of the features also used by Tzanetakis and Cook (2002).

Deep Neural Networks research emerged more recently, Abdel-Hamid et al. (2014) and Gemmeke et al. (2017) are examples of research at this field. Neural Networks normally do not use audio in the time domain as input, due to the high sampling rate of audio signals. A common alternative is to take spectrograms which represent time and frequency information of audio signals, they can be considered images and are used to train Convolutional Neural Networks as in Wyse (2017). Lidy and Schindler (2016) used a constant Q-transform (CQT) spectrogram as input to the CNN instead.

As in this work, Li et. al (2010) used the raw MFCCs matrix as input to CNN to predict music genre. This work was based on Adams (2019) repository that used MFCCs matrix as input, in Adams work it was compared a Recurrent and a Convolutional Neural Networks, since the CNN had better performance for audio classification, this was the Neural Network used in the present work. This project contains some singularities which allows it to be closer to a practical reality, first because a library from a production music company was used as dataset. Another point was the multi-label approach, Adams (2019) and most of the research done in audio classification field, consider single label classification, this dataset has most of audio classified in two genres, so it was necessary to adapt the model in order to classify two labels per track.

3. Dataset

It was used as dataset the whole library from a music production company which initially contained tracks classified in 55 different genres as shown on figure 1. The whole metadata was downloaded in CSV format with three columns, one with corresponding names of audio files and more two columns with respective genres of each track (each track could be classified with one or two genres).



```
Filmscore             12730.0
Atmosphere             9285.0
Electronica            6980.0
Pop                    6479.0
Ambient, Chill         4603.0
Rock                   4469.0
Classical Music        3105.0
World Music            2914.0
Hip Hop, Rap           2433.0
Country, Folk          2413.0
Easy Listening         1931.0
Indie, Alternative     1841.0
Dance                  1780.0
Jazz                   1508.0
Funk, Soul             1297.0
House                   889.0
Orchestral              750.0
Kids                    706.0
Acoustic                700.0
Blues                   688.0
Trailer                 657.0
Sound Design            646.0
Hard, Heavy             636.0
Latin                   572.0
Christmas               513.0
R'n'B                   492.0
60ies                   477.0
Swing                   458.0
Drone                   412.0
Big Band                403.0
Techno                  388.0
80ies                   370.0
Drum'n'Bass             352.0
70ies                   347.0
Piano                   326.0
Historical Music        290.0
Military                274.0
Dubstep                 259.0
Religious Music         255.0
Rock'n'Roll             246.0
Song                    235.0
Pizzicato               201.0
Percussion              200.0
German Folk             199.0
Reggae, Ska             197.0
Punk                    173.0
90ies                   152.0
Disco                    99.0
Industrial               96.0
20ies                    77.0
Breakbeat                69.0
A Cappella               68.0
50ies                    48.0
30ies                    46.0
40ies                    38.0
```

Figure 1: List of genres from the initial dataset and their respective number of occurrences.

All audio files were downloaded in mp3 format with sampling rate of 44,1 kHz, in total there were 48411 tracks, which were stored together with their metadata in an external hard drive. According to Termens (2009), the number of possible genres to classify audio files is inversely proportional to the accuracy obtained by the model, so based in the occurrences of each genre in the dataset the 29 genres with highest occurrences were selected, from *Filmscore* till *Drone* as shown on figure 1. The genres with lowest occurrences were all renamed as another genre category called *Others* and kept in the dataset, totalizing 30 genres in the dataset.

Another problem was the high data imbalance of this dataset, in order to reduce a bit this problem, part of the tracks that contained genres with highest occurrences were deleted from dataset, since we have a multi-label dataset, it was done another kind of analysis, counting genre 1 and genre 2 combined in one cell and generating another column to all the tracks, then it was possible to avoid deleting tracks that contain genres of higher incidence combined with those genres of lower incidence.

The audio files were all converted from mp3 to wav format, then were processed in order to trim a section of 40 seconds exactly in the middle of each track and save them in a new folder. This process was done also to avoid more data imbalance and to reduce the total size of audio tracks dataset, consequently reducing the processing time to obtain the MFCCs later.

4. Model Implementation

This section provides the information for each step of the Convolutional Neural Networks model implementation.

4.1 MFCCs and Hot-Encode Matrix

It was calculated the probability distribution for each label, and they were inserted in a list. Then a function to pick up samples randomly in the dataset was constructed and the probability distribution was added to this function in order to have random sampling based on this distribution.

The number of samples was chosen in order to have one sample per second when calculating MFCCs, which were stored in an array on variable X. The correspondent genres of each audio file sample, that had their respective MFCCs calculated, were encoded in a hot-encode tensor that was stored on variable y. After completed the MFCCs calculus and hot-encode procedure, X and y were saved in a pickle file, in order to avoid the long-time calculation of MFCCs each time that model being run.

The Cepstrum of an input signal is defined as the inverse Fourier transform of the logarithm of the signal spectrum. The use of Mel scale is justified as a mapping of the perceived frequency of a tone onto a linear scale, as a rough approximation to estimate the bandwidths of a human auditory filters. As a result of the MFCC computation we obtain a compact representation of the spectrum of the input signal that can be isolated from the original pitch. For the

MFCCs calculated on this project, the sampling rate was 44,1 kHz, the frame/window size (nfft) was 2048, the number of features was 13 and the number of filters in the filterbank was 26.

4.2 Convolutional Neural Networks

Using the MFCCs stored on X and the hot-encode matrix stored on y, the 2D Convolutional Neural Netoworks sequential model was built as shown on figure 2.

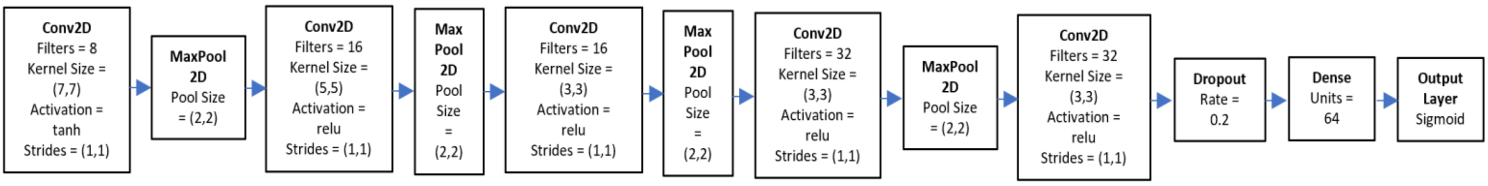


Figure 2: Sequence of layers, with their respective parameters, of the Convolutional Neural Networks model used.

This Convolutional Neural Networks was based in models built to classify single-label problems, so some modifications were needed, as the sigmoid output layer which is the one recommended for classification of multi-label problems, instead of softmax used in single-label classification. In the output layer, 30 nodes were used, since it corresponds to the number of labels in this case.

Another modification needed was in the loss function, Adams (2019) used categorical crossentropy, in this situation was used binary crossentropy. The adam optimization was used, and binary accuracy metrics in order to approximate the output predictions to 0 or 1, depending if it is under or over the settled threshold which was 0.5 in this case.

To train the model it was defined a batch size of 96 samples with 50 epochs, 15% of the dataset was separated for validation. The occurrences of each label inside of tensor y were calculated in order to determine the class weights, which were obtained concerning that each label supposed to have the same percentage of frequency. Checkpoint was also done to monitor loss values and save the best one as callbacks.

5. Results

Training the model was clear that the model reached the best results almost directly from the fist epoch with validation loss around 0.21 and validation binary accuracy of 0.93, without shown significant improvement of both variables during the other epochs as shown on figure 3.

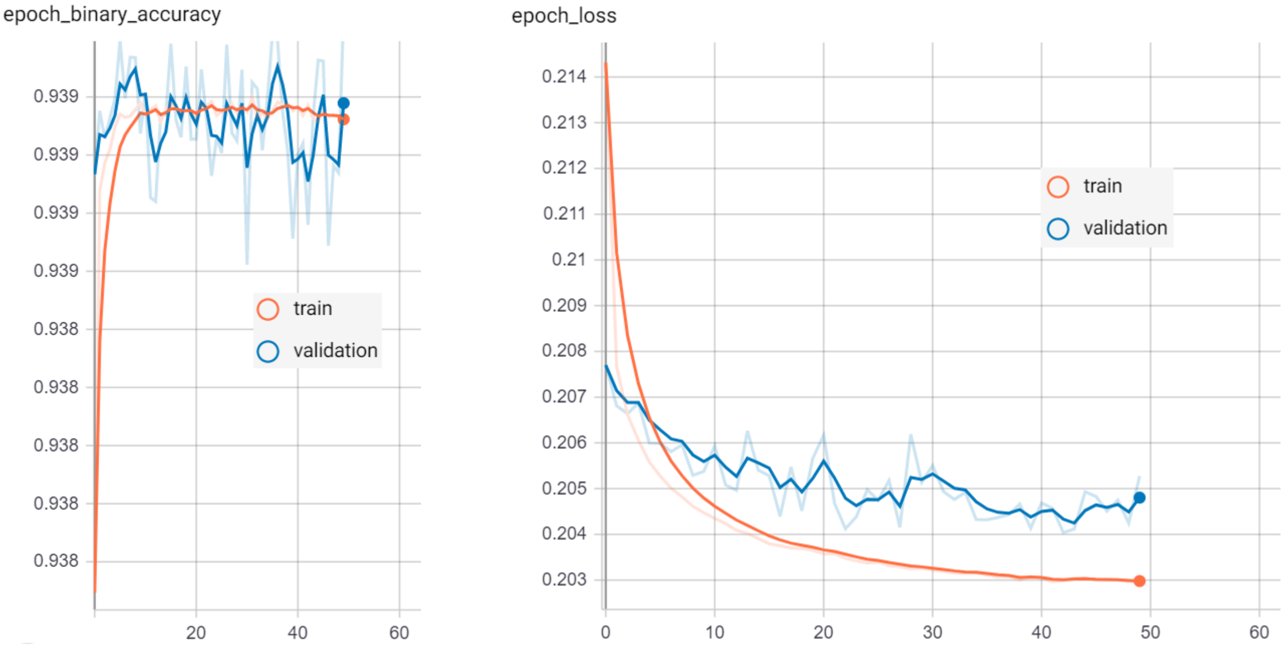


Figure 3: Graphics of binary accuracy and loss for each epoch, with the respective training and validation curves.

It was unexpected that a model had this kind of behaviour already in the first epochs, in order to discover the problem, some analysis in the predictions were done and discussed in the next section.

6. Discussion

Going further to investigate the reason for this model behaviour, it was clear from model predictions that results obtained were not satisfactory, since almost in all cases the labels were predicted with values between almost zero and 0.2, this explains the high values obtained for binary accuracy, since the whole prediction tensor was composed of zeros when binary accuracy was calculated and proportionally the number of true results with zeros is really higher than those of ones. Also trying to set lower thresholds, to turn the predictions of higher probability as one, did not help to improve the results.

Probably it was a wrong choice to use the whole library of the company as dataset, since the person responsible to classify track genres changed over the years, so the criteria of classification could be also changed. The huge size of dataset also made difficult to process some steps of the study, for example, it induced to long-time duration when MFCCs were calculated, it took almost four days to complete the whole calculus.

In continuity with the proposed theme, it will be tried to use as dataset all the tracks that had their genres classified by the person that do this task nowadays in the company. Since the dataset will be smaller, it will be also possible to set the number of samples that consists a quarter of second, which is the recommended value for MFCCs calculus. Another point that could be helpful to improve model accuracy, it would be to delete all tracks classified with genre *Others*.

7. References

Seth Adams. Audio-Classification. URL https://github.com/seth814/Audio-Classification

George Tzanetakis and Perry Cook. 2002. Musical genre classification of audio signals. IEEE Transactions on speech and audio processing 10(5):293–302.

Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu. 2014. Convolutional neural networks for speech recognition. IEEE/ACM Transactions on audio, speech, and language processing 22(10):1533–1545. 1 - https://github.com/seth814/Audio-Classification

Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. 2017. Audio set: An ontology and human-labeled dataset for audio events. In Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on. IEEE, pages 776–780.

Lonce Wyse. 2017. Audio spectrogram representations for processing with convolutional neural networks. arXiv preprint arXiv:1706.09559.

Thomas Lidy and Alexander Schindler. 2016. Parallel convolutional neural networks for music genre and mood classification. MIREX2016.

Tom LH Li, Antoni B Chan, and A Chun. 2010. Automatic musical pattern feature extraction using convolutional neural network. In Proc. Int. Conf. Data Mining and Applications.

Enric Gauss I Termens, 2009. Audio content processing for automatic music genre classification; descriptors, databases, and classifiers. URL https://www.tdx.cat/bitstream/handle/10803/7559/tegt.pdf?sequence=1&isAllowed=y

Multi-label deep learning with scikit-multilearn. URL http://scikit.ml/multilabeldnn.html

Harresh Bahuleyan, 2018. Music Genre Classification Using Machine Learning Techniques. URL https://arxiv.org/pdf/1804.01149.pdf