Machine Intelligence Lab Hackathon: Learning to Recognize Musical Genre

Matthias Büchi

January 12, 2018

Abstract

In the course of a final project in the *ZHAW Machine Intelligence Lab* the challenge *WWW 2018 Challenge: Learning to Recognize Musical Genre* was tackled. With the increasing popularity of music streaming services and large music databases, an automatic system for managing the data is essential. The challenge exactly targets this topic, specifically classifying musical audio into genres (e.g. rock, pop, etc.). During the 3 days of work in this challenge the main focus was on implementing an approach using convolutional neural networks (CNN) on raw audio signals.

1 Introduction

The task of the challenge was to recognize the genre from a piece of music. Given a piece of audio, with a length of 30 seconds, one of 16 genres should be predicted. For this purpose a dataset of musical audio was provided in form of the *FMA Dataset* ([1]). But only the *medium* subset must be used for training, consisting of 25000 tracks. Furthermore a test set with 35000 tracks without labels was given, which had to be predicted. The performance of submitted results was evaluated using *Mean Log Loss* and *Mean F1 Score* as a second metric.

First the baseline system in form of a SVM, available in the *starter-kit* ([2]) of the challenge, was reproduced and achieved a *Mean Log Loss* of ∼0.985 and a *Mean F1 Score* of 0.6922. It used features also provided for the challenge, where every track has one feature vector with a dimension of 518. For different feature types, like *MFCC*, the values and their statistics are averaged over the full track and then concatenated to form a single feature vector.

In a next step the data was explored and prepared for training a neural network. From overlapping windows of the raw audio signal a convolutional neural network (CNN) with subsequent fully-connected layers was trained to predict the genre of a given window.

2 Findings

2.1 Data preparation

The training data consisted of 25000 music tracks, but unbalanced with respect to the genres. While the biggest genre had 7097 samples, the smallest one only had 21 samples. For training, the data was further split into a training and a validation set, where the training part contained 80% of the tracks for every genre.

2.2 Recognition System

As input to the recognition system 1-second windows shifted by 0.25 seconds from the raw audio signal were used. Samples smaller than a second were padded with zeroes.

To extract features from the signal three convolutional layers with average pooling were used. The first two layers use small filter sizes, whereas the third uses a bigger filter size, intended to model longer temporal patterns. Subsequent fully-connected layers, with softmax as final activation, were used to predict the probabilities for 16 musical genres. Except for the last layer, batch normalization and ReLU activation were used.

Layer	Size	Stride	Activation
conv-32	5	1	ReLU
avgpool	4	4	-
conv-64	5	1	ReLU
avgpool	4	4	-
conv-128	100	20	ReLU
avgpool	40	30	-
fc	70	-	ReLU
fc	30	-	ReLU
fc	16	-	Softmax

Table 1: Layers and their properties used in the recognition system.

2.2.1 Training

The system was trained using the *Adam* optimizer with a learning rate of 0.001 for two iterations over the training data. It was trained to optimize the *Binary Cross Entropy Loss*.

2.2.2 Prediction

For prediction, non-overlapping 1 second windows were used. The output of all windows for a single track were averaged to represent the final prediction. The submitted result achieved a *Mean Log Loss* of 1.098 and a *Mean F1 Score* of 0.6672.

2.3 Methods

The system was implemented in Python, using PyTorch for training the neural network. Experiments were performed on a server using a single GPU.

3 Outlook

The system implemented seems to be a good starting point for further development. Its performance is not far from the baseline that uses a lot of specific features like *MFCC*, *Chroma*, etc and the full *medium* subset. Before putting a lot of effort on implementing some further system, a bit more insight on the topic should be worked out, especially on what properties make the difference between the genres. Together with knowledge of error distribution between the genres the further steps can be determined.

One problem that should be addressed anyway is the data, since it is very unbalanced. A test with additional data from the *FMA dataset* could be helpful to see if the given system works better with balanced data. If so, the given training data could be extended for example by using shorter window shifts for the smaller genres.

For improving the model, the learned feature maps from the convolutional layers have to be analyzed. This could be helpful to improve the layout and properties of the model. In addition it should also be investigated how the model behaves when training for a longer time.

Music also contains patterns over a longer time. Therefore the system could be improved to model these long-term properties, for example using a recurrent neural network (RNN).

Another approach that could be interesting is the usage of extracted features from the audio signal (e.g. *MFCC*, *Chroma*, etc). As the baseline proves they work pretty good and they just average these features and their statistics over a full track. With classification on frames the temporal patterns could probably be modelled better.

References

[1] Michaël Defferrard, Kirell Benzi, Pierre Vandergheynst, and Xavier Bresson. Fma: A dataset for music analysis. In *18th International Society for Music Information Retrieval Conference*, 2017.

[2] S.P. Mohanty and Michaël Defferrard. crowdai-musical-genre-recognition-starter-kit. `https://github.com/crowdAI/crowdai-musical-genre-recognition-starter-kit`, 2017. [Online; accessed 10-January-2018].