Machine Learning:
Music Genre Classification

Author

Francesco Ranieri

Introduction

Music Experts have been trying for a long time to understand sound and what differentiates one song from another, how to visualize sound and what makes a tone different from another. Music classification (MC) tasks could be the answer of this long time question. The MC aim is to understand the music semantics over various different features. In this study case several solutions are explored in order to solve a specific problem of MC called Music Genre Classification which try to classify music based on its genre. The result is an ensemble learning approach that seeks better predictive performance by combining the predictions from multiple models: Random forest algorithm and Recurrent Neural Network.

Music Classification

Music classification is a music information retrieval (MIR) task whose objective is the computational comprehension of music structure. For a given song, the classifier predicts relevant musical attributes.[7] Based on the task definition, there are a nearly infinite number of classification tasks – from genres, moods, and instruments to broader concepts including music similarity and musical preferences. The retrieved information can be further utilized in many applications. Easy examples nowadays could be services such as Spotify and YouTube that quickly create playlists and suggest the next song based on our listening patterns. These patterns are influenced by the tone, mood, or genre of songs we listen to. Several variables influence the decision, but most have to do with how the song "sounds." For example, a pop song might feel faster and "funkier" than, say, a romantic song; based on specific parameters that define these features, music streaming services group similar sounding or feeling songs into the same genre. Thus, classifying music based on genres can help suggest the next songs to a listener, curate playlists of new recommendations, or filter undesirable content.[1] This paper analyzes the single label genre classification which means that the single music track belongs to a specific class.

Understanding Audio

In order to better clarify the study case is crucial to understand what is a sound and how can it is represented. Sound is defined as vibrations that travel through the air or another medium as an audible mechanical wave (figure 2). It is produced from a vibrating body. The vibrating body causes the medium around it to vibrate thus producing sound.

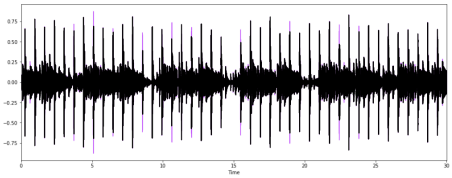

Figure 1: *Sound wave*

Fourier Transform

A mathematical representation of sound is the Fourier Transform. It is a function that gets a signal in the time domain as input, and outputs its decomposition into temporal frequencies. The scale of color represents a decibels scale.

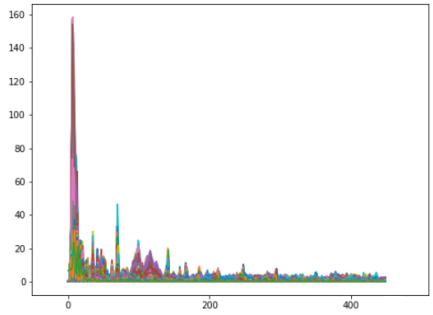

Figure 2: *Fourier Transformation*

Spectrogram

Then we have the spectrogram which is a visual representation of the spectrum of frequencies of a signal

as it varies with time where the Frequency spectrum of a signal is the range of frequencies contained by that signal [2].

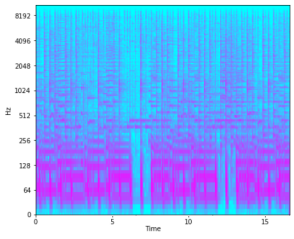

Figure 3: *Spectrogram*

Mel spectrogram

The Mel Scale, mathematically speaking, is the result of some non-linear transformation of the frequency scale. The Mel Spectrogram is a normal Spectrogram, but with a Mel Scale on the y axis [4].

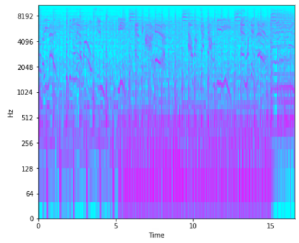

Figure 4: *Mel spectrogram*

The Mel frequency cepstral coefficients

The Mel frequency cepstral coefficients (MFCCs) of a signal are a small set of features (usually about 10–20) which concisely describe the overall shape of a spectral envelope. In other words, the MFCCs try to reduce the spectrogram feature set by selecting only a sub set that can represent its shapes. It is used to model the characteristics of the human voice.

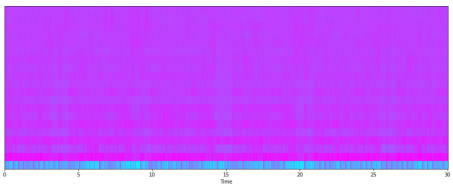

Figure 5: *MFCCs*

Dataset

The first Dataset used in this case of study is the GTZAN Genre Dataset. The GTZAN genre collection dataset consists of 1000 audio files each being 30 seconds in duration. The dataset contains 10 classes that represent 10 music genres. The music genres include blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae, and rock. Each class contains 100 audio tracks that are in .wav format. In order to have more entries in the dataset, each song is split in 10 songs of 3 seconds each. In this way, the dataset has been expanded to 10,000 songs, 1000 for each genre. This GTZAN dataset contains 60 features for each song: 58 numerical features, the labels and the file name. The second dataset is inspired by the GTZAN dataset and it tries to represent a song by using only information present in Mel Frequency Cepstral Coefficients (MFCCs). It is created by analyzed the 3-seconds songs and thanks to librosa python package it was possible to extract 13 set of features (MFCCs) per song in order to have exactly 10000 entries, one for each song in the GTZAN dataset.

Finally both the datasets are divided into training set and test set by using the 67% for training and 33% for testing. Furthermore the training sets are divided using the same percentage into the a smallest training sets and a validation sets. This later division is useful for fine-tuning the model and to avoid human contamination in the evaluation phase on the test sets. In this way, a model is trained on the smallest training set and the fine-tuning phase aim is to perform better on validation set. When a good result is achieved, then the test set is shown to the model in order to obtain the final score.

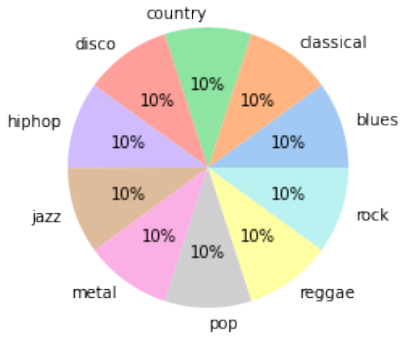

Figure 6: *GTZAN & MFCC: both have the same distribution*

It is really noticeable that the two datasets have a really good balance among the 10 classes, which is unusual in real scenarios but this situation could help to have better result and a more accurate model, because there is no need to give equal priority to each class.

Experiments

Environment

The environment used for the development and testing of models is the Google Colab Pro. A GPU environment is chosen because some of the models can leverage on this type of hardware to improve training and prediction time. The Machine Learning implementations are made up whit the library Scikit-Learn and Tensorflow (Keras). The music manipulations are possible thanks to the librosa python package.

Experiments Overview

During the study case various models and hyperparameters are tried for the Music Genre Classification task.

On the GTZAN dataset many machine learning and deep learning approaches have been applied, ranging from probabilistic algorithms, ensemble learning algorithms to arrive at shallow neural networks, such as:

- Naive Bayes
- Nearest Neighbors
- Logistic Regression
- Random Forest
- Ada Boost
- Gradient Boosting
- Decision Tree
- Support Vector Machines (SVC)
- Multi Layer Perceptron (MLP)
- Feed Forward Neural Network (FFNN)

While on the MFCCs dataset are applied only deep learning approaches that are able to work with time series. The most common architectures have been employed to solve this task: Recurrent Neural Networks and Convolutional Neural Networks.

The main idea is to train two different models, one per dataset and combine the results of this two classifiers as the final model. The most common approach is to use voting, where the predicted probabilities represent the vote made by each model for each class. Votes are then summed and the label with the largest mean probability is selecting. [6]
Best performing models are selected by accuracy comparison on the test set.
The result of each experiments for the GTZAN dataset is attached in the appendix section.

Loss function

When doing multi-class classification, categorical cross entropy loss is used a lot. It compares the predicted label and true label and calculates the loss. In Keras, the TensorFlow back-end supports both Categorical Cross-entropy and its variant: Sparse Categorical Cross-entropy. For all models involved in this study case it is used the **Sparse Categorical Cross entropy**. The only difference between sparse categorical cross entropy and categorical cross entropy is the format of true labels. Sparse Categorical Cross entropy is used, in fact, when the labels are mutually exclusive for each data, meaning each data entry can only belong to one class. This cross entropy variant is really suitable for this context because in GTZAN dataset a song must belong to a single class so the ten genres (blues, classical, country, etc..) are mutually exclusive. The formula of the Sparse Cross Entropy is the same as the Categorical Cross Entropy and the difference is both variants covers a subset of use cases and the implementation can be different to speed up the calculation.

$$J(w) = \sum_{i=1}^{size_output} y_i * \log \hat{y}_i \tag{1}$$

where

- y_i is the true label
- $\hat{y}_i$ is the predicted label

Final solution

Lots of models have been validated during this study case on the GTZAN dataset but most of them are under the accuracy threshold of 50%.
There are only a few exceptions which are the Random Forest classifier, the Gradient Boosting approach and the Feed Forward Neural Network. It turned out empirically that the best model for the first dataset, in terms of accuracy and training time, is the Random forest model that achieves 86% of accuracy on the validation set and 85% on the test set.
For this task Derek A. Huang [5] proposed a complex feed forward neural network and the result was a 96% accuracy on the training data and 54% on test set. A very simple neural structure has been adopted in order to avoid over fitting on training data and to obtain very high performance on the validation and test set. In fact, with a shallow design the accuracy final result is 86% on the test set.
For the MFCCs dataset which is composed only by time series element, we decided to employ one of the best architecture for time series manipulation, the Recurrent Neural Network. Then a simple Convolutional Neural Network structure is used. This is an uncommon architecture for time series but according to Felix Andika paper titled "Convolutional Neural Network for time series classification" [3], this kind of networks could be applied to time series predictions and shows good results. Finally, the vain attempt of applying a Feed Forward Neural Network produced poor results.

The models used to produce the final output is the one which resulted the best in the test phase on both datasets, namely the result of fine-tuning of Random Forest(RF) for GTZAN dataset and RNN model for MFCCs. The result is an ensemble learning approach by using the chosen models. The prediction of the ensemble model is the genre class with higher mean probability of the two models. More specifically the input

Model	Structure	% Accuracy
FNN	Dense128, Dropout 0.4, Dense 64, Dense 32, Dense 16, Dense 10	42
CNN	Conv 2D 8, Maxpooling (2,2), Conv 2D 32, Maxpooling (2,2), Flatten, Dense 128, Dense 64,Dense 10	63
RNN	LSTM 64, LSTM 64, Dense 64, Dropout 0.3, Dense 10	89

Table 1: *Optimal configurations for model trained on MFCCs dataset*

of this model is a 3 seconds song and its corresponding MFCCs matrix; the 3 second song is analyzed by the RF model and the MFCCs matrix is analyzed by the RNN model. Then, both the models calculate a vector of probability of belonging to each specific musical genre. Finally an arithmetic average is carried out between the probability vectors and the final prediction is the higher mean probability. The ensemble model scores on test set an accuracy of 87%.

Dataset	Model type	Hyperparameters	Accuracy	Training Time
	Naive Bayes		0.42	0s
		n_neighbors=1 metric=minkowski	0.26	2s
		n_neighbors=2 metric=minkowski	0.25	2s
		n_neighbors=5 metric=minkowski	0.27	3s
	Nearest Neighbors	n_neighbors=8 metric=minkowski	0.28	3s
		hidden_layer_sizes=(128, 64, 8), random_state=1, max_iter=200, learning_rate_init=0.001	0.097	5s
		hidden_layer_sizes=(128, 64, 8), random_state=1, max_iter=250, learning_rate_init=0.001	0.09	4s
		hidden_layer_sizes=(128, 64, 8), random_state=1, max_iter=300, learning_rate_init=0.001	0.10	8s
		hidden_layer_sizes=(128, 64, 8), random_state=1, max_iter=250, learning_rate_init=0.0001	0.10	8s
		hidden_layer_sizes=(256, 128, 64, 8), random_state=1, max_iter=250, learning_rate_init=0.0001	0.11	12s
		hidden_layer_sizes=(16, 8), random_state=1, max_iter=250, learning_rate_init=0.0001	0.10	13s
		hidden_layer_sizes=(128, 64, 8), random_state=1, max_iter=250, learning_rate_init=0.00001	0.10	15s
	MLP	hidden_layer_sizes=(128, 64, 8), random_state=1, max_iter=250, learning_rate_init=0.000001	0.11	12s
		max_iterint, default=1000	0.15	20s
	SVC	max_iterint, default=10000	0.23	180s
	RANDOM FOREST		0.86	10s
	Ada Boost		0.40	8s
	Gradient Boosting		0.82	90s
	Decision Tree	max_depthint=None	0.60	15s
	Logistic Regression	multi_class = multinomial	0.21	60s
		multi_class = multinomial, solver= sag, max_iter = 1000	0.36	75s
		multi_class = multinomial, solver= sag, max_iter = 2000	0.36	81s
		multi_class = multinomial, solver= sag, max_iter = 10000	0.45	182s
		multi_class = multinomial, solver= sag, max_iter = 10000	0.34	182s
GTZAN	FFNN	Dense (128, 64, 32, 10, 10), epoch = 50 relu on dense layer, the last one with softmax	0.59	<130s
		Dense (128, 64, 32,10), epoch = 50 relu on dense layer, the last one with softmax	0.50	<120s
		Dense (64, 32, 32, 10), epoch = 50 relu on dense layer, the last one with softmax	0.74	<120s
		Dense (32, 32, 16, 10), epoch = 50 relu on dense layer, the last one with softmax	0.77	<120s
		Dense (32, 16, 10), epoch = 50 relu on dense layer, the last one with softmax	0.78	<120s
		Dense (32, 16, 10), epoch = 100 relu on dense layer, the last one with softmax	0.81	<120s
		Dense (32, 16, 10), epoch = 200 relu on dense layer, the last one with softmax	0.79	<120s
		Dense (32, 16, 10, 10), epoch = 50 relu on dense layer, the last one with softmax	0.76	<120s
		Dense (32, 16, 10, 10), epoch = 100 relu on dense layer, the last one with softmax	0.80	<120s

Table 2: GTZAN Experiments Results

References

[1] In: 2 (). URL: https://www.projectpro.io/article/music-genre-classification-project-python-code/566#mcetoc_1fsm039887.

[2] In: 4 (). URL: https://www.sciencedirect.com/topics/engineering/frequency-spectrum#:~:text=Frequency%5C%20spectrum%5C%20of%5C%20a%5C%20signal,(5f)t%5C%20%5C%2B%5C%20%5C%E2%5C%80%5C%A6).

[3] Felix Andika. In: 7 (). URL: https://machinelearningmastery.com/how-to-develop-convolutional-neural-network-models-for-time-series-forecasting/.

[4] Dayla Gartzman. "Getting to Know the Mel Spectrogram". In: 3 (). URL: https://towardsdatascience.com/getting-to-know-the-mel-spectrogram-31bca3e2d9d0.

[5] Derek A. Huang. In: 6 (). URL: https://github.com/derekahuang/Music-Classification/blob/master/CS229_Final_Report.pdf.

[6] "machinelearningmastery". In: 5 (). URL: https://machinelearningmastery.com/combine-predictions-for-ensemble-learning/.

[7] Keunwoo Choi Minz Won Janne Spijkervet. "What is Music Classification?" In: 1 (). URL: https://music-classification.github.io/tutorial/part1_intro/what-is-music-classification.html.