International Journal of Computing and Artificial Intelligence

E-ISSN: 2707-658X
P-ISSN: 2707-6571
IJCAI 2023; 4(2): 01-05
Received: 03-05-2023
Accepted: 02-06-2023

Sidhdharth H Pandya
Computer Science Department
California State University,
East Bay, Hayward, USA

Moayed D Daneshyari
Computer Science Department
California State University,
East Bay, Hayward, USA

Corresponding Author:
Sidhdharth H Pandya
Computer Science Department
California State University,
East Bay, Hayward, USA

Content based music genre classification using temporal and spectral features

Sidhdharth H Pandya and Moayed D Daneshyari

DOI: https://doi.org/10.33545/27076571.2023.v4.i2a.65

Abstract
Music is heard by everyone. It spreads emotions from one person to another. The music corpus of today is quite diverse. Each person has their own music evaluations. Mostly due to the diversity of the composers and musicians. However, many companies that offer music streaming services, like Pandora, Spotify, Google, Apple Music and Prime music utilize state-of-the- art (SOTA) algorithms to identify similarities and patterns between tracks. As a result, recordings are classified into several groups known as "Genres" of tunes. It makes similar audio tracks and music recommendations based on the genres a user listen to.

Keywords: Music, genre, classification, hierarchical LSTM, GTZAN

Introduction
We have access to an enormous quantity of data, and this number is growing fast every day. As a result, manual duration is becoming impractical, and automated techniques must be used to classify data. The music business is no exception. Automating the music tagging process would result in better data organization, enabling future development with this data easier, such as building themed playlists or recommending songs to users. Machine learning may be used to detect subtle patterns in data that would be difficult to explicitly build methods for. One such use case is deciding what genre a song belongs to, which is covered in this study. Finding patterns in audio is valuable for more than just musical analysis.
Music genre classification, while on the other hand, is, as previously said, an unclear and subjective process. It is worth noting that there is no precise description of what a genre should sound like, as this is a fairly conventional way of looking at music. Saying a song should be categorized a specific way is not right or incorrect; rather, it is a matter of personal opinion based on how the listener is affected by the music and how they identify with it [1]. It is also a challenging field of research, either because of low classification accuracy or because some argue that one cannot categorize genres that do not even have clear guidelines [2]. Though identifying music is not a random process for humans, there must be some consistency when it comes to what a genre sounds like [3].

Literature Review
Besides the content-based music genre classification, other techniques exist as well such as:
A. Collaborative filtering: From [4], this approach makes a prediction about the taste of a user with the help of part of the community that shares the same (or similar) taste (thus called collaborative). An important assumption is made that if a user, say a, listens to the same songs of a genre as the users B, C, D and E, then A would also prefer the songs of other genres listened to by B, C, D and E. The drawback of this technique would be, for the collaborative filtering approach to work, there must be a large set of users (i.e., the community) and user data.

B. Knowledge-based: This approach (from) [5] draws in user interests and feedback at regular periods. This technique is used mainly when the other two (content-based and collaborative filtering) cannot be applied. This approach depends on user feedback (an example is the like and dislike option provided by Spotify for each song). Thus, inadequate feedback or unable to obtain feedback regularly hinders the working of the approach.

This can be considered as a drawback.

Deep learning-based methods are fairly popular when it comes to problems such as music genre classification. However, music (audio) data has a nice sequential structure. The order of data is important, and data across the timestamps should not be treated as independent. Recurrent neural networks address this issue as they are networks with loops in them, allowing past information to persist. LSTM networks are a special kind of RNN, capable of learning long-term dependencies. In the project we have proposed a hierarchical LSTM-based model for the multi-class classification problem. We adopted the hierarchical LSTM architecture from [6] with some modification mentioned below:

- We proposed two different kind of approaches hard prediction and soft predictions.
- We are using sequence length = 256.
- We added Chroma-STFT, Spectral Centroid and Spectral Contrast features.

Methodology

A. Dataset

The dataset used in the project is the GTZAN dataset [7] available at the MARYSAS website [8]. Review of over 467 published works in music genre classification done by B. L. Sturm [9] proves that the GTZAN dataset is most used public dataset which is appearing in more than 100 works.

This dataset was originally used in [7]. The files were collected in 2000-2001 from a variety of bases. The audio files were gathered using different sources like CD's, radio, microphone recordings etc. to represent a variety of recording conditions [8].

For a long time, experts have been attempting to comprehend sound and what distinguishes one music from another. Sound visualization. What separate one tone from another. This data, ideally, will provide the possibility to do so.

The dataset is made up of about 1000 audio tracks each of which is 30 seconds long. There are about 10 genres, each covering up 100 tracks for each genre. All tracks have 22050 Hz sampling rate, mono channel with 16-bit sample audio files in.wav format.

B. Preprocessing

In computers, audio track is represented as digital signal. Digital signal is discretized representation of the Analog signal, and it is sampled at some sample rate. Audio track is either recorded using microphone or synthesized within computer itself. Digital audio signal feature extraction is done emphasizing these fine-grained features like frequency, amplitude, phase, pitch, timbre, amplitude envelope, power of signal and tonal features.

Audio Signals are represented into two domains namely (1) Time Domain and (2) Frequency Domain. Both empirically and theoretically it is known that complex signals are easy to decompose in the frequency domain.

The time domain to frequency domain conversion can be done using algorithms like Fast Fourier Transform (FFT). In Discrete Fourier Transform (DFT), we can use the dot product to find the similarity between our signal and the complex signal with frequency f. And we can do this for all frequencies between fmin to fmax to find the transformed signal.

Fourier transform of the signal gives the spectrum for whole signal at once. So, it will give global features. However, we are interested to find the non-stationarities in the signals as we are using recurrent models like LSTM.

To achieve this, we can use something like windowing method and perform (Short Time Fourier Transform) STFT on each window which convolves over our signal with stride equal to hop-length. This will give something like spectrum. Frequency perception in human is non-linear in nature and spectrum captures the frequencies in the linear scale. Mel-Spectogram is variant of spectrogram which captures frequency on Mel-Scale which is similar to logarithmic scale.

Mel-Spectogram facilitates almost good features, but it is high dimensional. Some methods [10] use CNN (Convolutional Neural Network) based approaches and it use this as feature.

Since we are using LSTM based approach, we can use denser features like MFCC, Chroma- STFT, Spectral Centroid and Spectral Contrast which is described in the next section. We use 22050 Hz sampled, 30-second-long audio file.

We are taking window size = 2048 samples and hop-length = 512 samples. So, we will have approx. 1290 timestamps. Here, each window acts as a timestamp for our LSTM network.

Since LSTM has restriction on the sequence length, we fix the sequence length = 256. To do this we split the tracks into 5 segments such that each segment has 256 timestamps. Note that here number of timestamps is the same as number of different windows on which we are finding the features.

Our dataset had initially 1000 songs but after enforcing the sequence length = 256, we are kind of augmenting the dataset such that it will have total of 5000 segments each of sequence length = 256.

The MFCC Features captures the details about the envelope in IDFT (inverse Discrete Fourier Transform) of the Log Power Spectrum which is extracts similar features from Mel- Spectogram but represented in dense way.

We are using 20, Chroma-STFT features finds the Intensity for different tones of the current window. There are 12 tones namely (C, C#, D, D#, E, F, F#, G, G#, A, A#, B) So, per window its dimensionality is 12.

Spectral Centroid captures the average frequency weighted by intensity at that frequency. Per window its dimensionality is 1.

And Spectral Contrast is an alternative to MFCC features which works well to do genre classification task as described in paper [5]. Per window its dimensionality is 7.

So, combining all above features, total we have 20 (MFCCs) + 12 (Chroma-STFT) + 1 (Spectral Centroid) + 7 (Spectral Contrast) = 40 features. So, our dataset has total 5000 examples (after augmentation), each example has sequence length of 256 timestamps and each timestamp has dimensionality = 40.

Now for each timestamp, we calculate following features

Table 1: Dimensionality

Features	Dimensionality
MFCC	20
Chroma STFT	12
Spectral Centroid	1
Spectral Contrast	7

LSTM Networks as multi-class classifier

To achieve multi-task classification, we have used a LSTM-based model. LSTM [11] is good technique to use for music genre classification as it remembers the past result of the cell in the recurrent layer and classify music in a better and efficient way.

Instead of using a single LSTM network to perform a 10-class classification task, we use a divide and conquer approach, by using a hierarchical tree-based 7 LSTM network architecture.

The following figure shows the proposed hierarchical LSTM architecture.

The functionality of each of the LSTM networks is given as follows:

LSTM 1: It classifies between strong (hip-hop, metal, pop, rock, reggae) and mild (jazz, disco, country, classic, and blues) genres of music.

LSTM 2a: It classifies between sub-strong 1 (hip-hop, metal, and rock) and sub-strong 2 (pop and reggae).

LSTM 2b: It classifies between sub-mild 1 (disco and country) and sub-mild 2 (jazz, classic, and blues)

LSTM 3a: It classifies between hip-hop, metal and rock.

LSTM 3b: It classifies between pop and reggae.

LSTM 3c: It classifies between disco and country.

LSTM 3d: It classifies between jazz, classic, and blues.

This hierarchical architecture, helps to tackle the multi-class classification problem, by a divide and conquer based approach, where each LSTM in the tree, is trained using samples of the relevant classes. The idea for the hierarchical LSTM model was adopted from [6].

The architecture of the individual LSTMs is given in Fig 1:

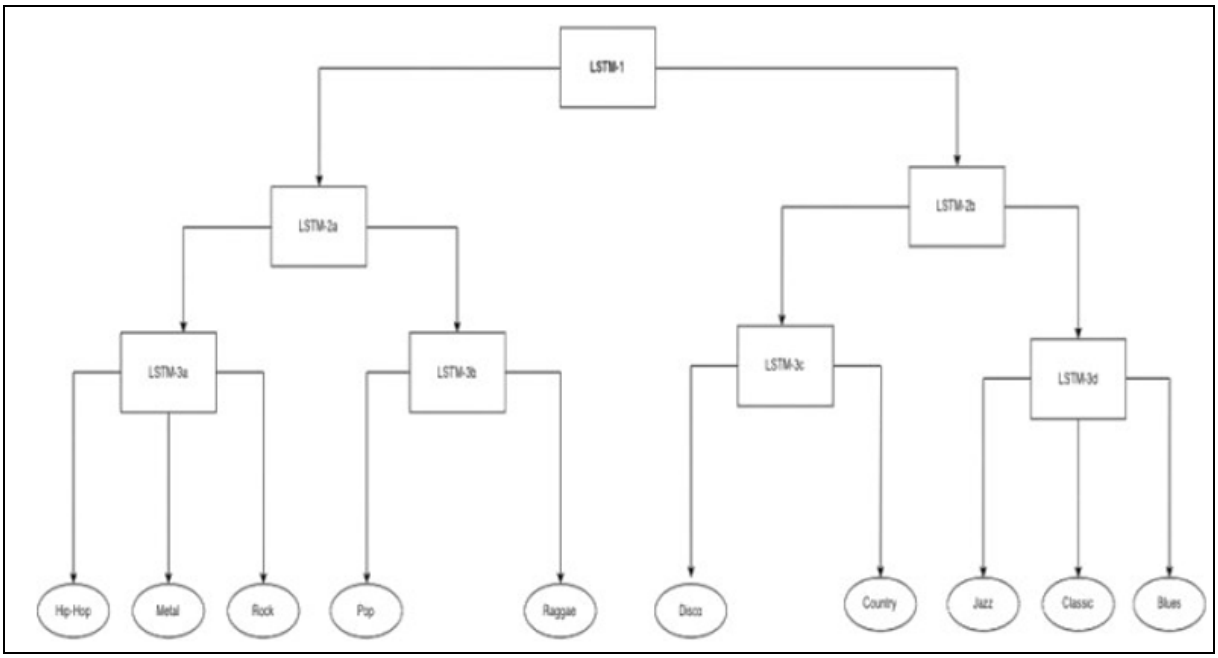

Fig 1: The Hierarchical LSTM architecture

Table 2: Hierarchical LSTM Models

Input Layer	Total 40 features (MFCC, Chroma STFT, Spectral Centroid, Spectral Contrast)
Hidden Layer I	64 LSTM units
Hidden Layer II	32 LSTM units
Output Layer	Number of SoftMax units depends on which the number of fan-out results.

A. Experimental Results

The proposed multi-step classifier involves the 7 LSTMs shown in Fig 1. The input music is initially identified as being either strong or mild by LSTM1 during the testing phase. After that, either LSTM2a or LSTM2b is applied depending on the outcome. In accordance with the outcomes from the previous level, LSTM3a, 3b, 3c, or 3d are then utilized to categorize the music into the desired categories.

Fig 1Error! Reference source not found. Displays the results of this experiment. Our method yielded a 75.00% accuracy. It outperformed the convolutional neural network approach, which had an accuracy of 73.54%. The LSTM hierarchy in our multi-step classifier is depicted schematically in Fig 1.

The dataset was split into 75 percent for train, 15 percent for validation and 10 percent for test sets, while applying the stratified property. Using the stratified property, the dataset was divided into 75 percent train, 15 percent validation, and 10 percent test sets. The stratified attribute was used because it is preferable to divide the dataset into train and test sets while preserving the proportions of instances in each class that were seen in the original dataset, hence avoiding class imbalance. The total number of samples for LSTM1, LSTM2a, LSTM2b, LSTM3a, LSTM3b, LSTM3c, and LSTM3d are 5000, 2500, 2500, 1500, 1500, 1000, 1000, 1500 respectively. We have used two prediction methods: Hard Prediction

- In this, we perform segmentation on the original track so that each segment will have 256 timestamps after feature extraction.
- Predict the class for each segment.
- Select label with highest frequency.
- For all segments, select the path with majority predicted label.

- Repeat for all internal nodes until leaf nodes are not predicted.
- Return the predicted label. Soft Prediction
- In this, we perform segmentation on the original track so that each segment will have 256 timestamps after feature extraction.
- Predict the class for each segment.
- For each segment, choose path independently.
- Do this for all internal nodes until leaf nodes are not predicted.
- Return all predicted labels with the frequencies.

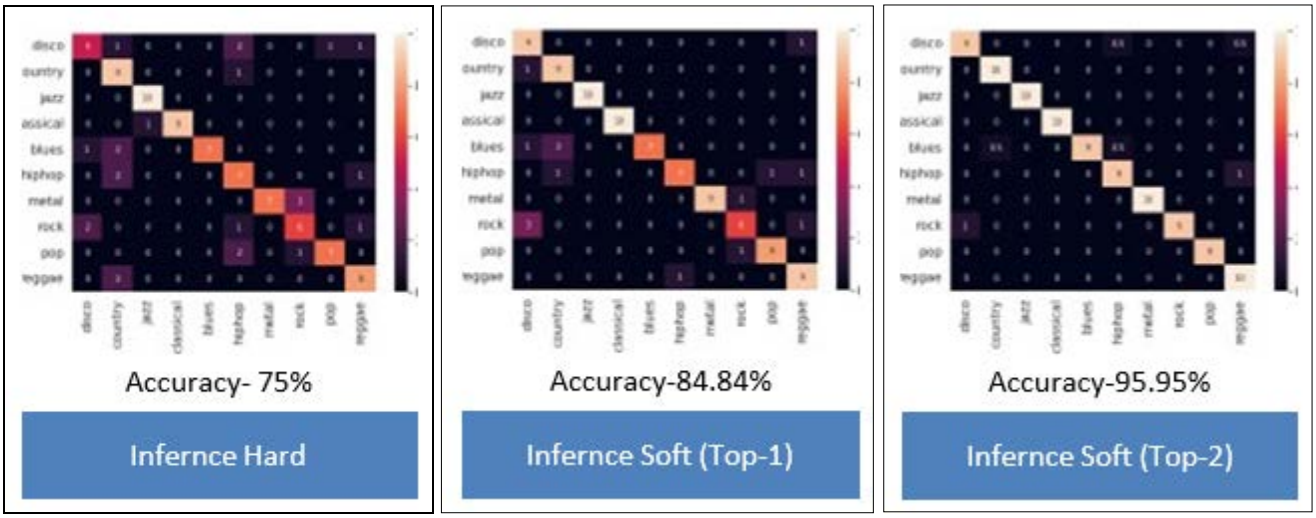

Fig 2: Result of hierarchical LSTM Model

Conclusion

The content-based music classification approach is being used in the industry widely. As there are numerous machine learning techniques that work well on extracting pattern, trends, or other useful information from a large dataset, thus these techniques are suitable for performing music analysis. Several companies are using music classification to segment their database according to genre and are either using it to recommend songs to their users like done by Spotify, YouTube Music etc. or even purely as a product like Shazam.

In conclusion, the experimental results show that our multi-step classifier based on Long Short-Term Memory (LSTM) model is effective in recognizing music genres. I employed a divide-and-conquer strategy to classify ten different genres. We reached an accuracy of 75.00%, which was higher than one of the SOTA approaches, which had an accuracy of 73%.

Even after dealing with over fitting difficulties, CNN with the Max-Pooling function beat both RNN with the LSTM model and the conventional multi- layer model with three hidden layers. Our experiment is carried out on a MacBook Pro with M1 chip running on a mac operating system. Compared to the multilayer design, CNN has the disadvantage of being more costly. As a result, the suggested multilayer model is faster than the CNN model. We run RNN with the LSTM model across 50 epochs, and each epoch takes substantially longer to build than any prior model.

Table 3: Performance comparison of different models

Evaluation Metric	Multi- Layer Perception	CNN with Max-Pooling	RNN with Long Short-Term Memory	Hierarchical LSTM
Training Accuracy	73.12	86.58	77.57	Inference Hard- 75% Inference Soft-Top-1 – 84% Top-2 – 95%
Testing Accuracy	58.85	73.54	69.23	

References

1. Lee JH, Downie JS. Survey of music information needs, uses, and seeking behaviours: Preliminary findings. Proceedings of the international conference on music, information retrieval; c2004.
2. North AC. Liking for musical styles. Music Scientae. 1997;1(1):109-128.
3. Tekman HG, Hortacsu N. Aspects of stylistic knowledge: What are different styles like and why do we listen to them? Psychol. Music. 2002;30(1):28-47.
4. J Ben Schafer, Dan Frankowski, Jon Herlocker, Shilad Sen. collaborative filtering recommender systems. Springer Berlin Heidelberg, Berlin, Heidelberg; c2007. p. 291-324.
5. Charu C. Aggarwal. Knowledge-Based Recommender Systems. Springer International Publishing, Cham; c2016. p. 167-197.
6. Chun Pui Tang, Ka Long Chui, Ying Kin Yu, Zhiliang Zeng, Kin HongWong. Music genre classification using a hierarchical long short term memory (lstm) model. In Third International Workshop on Pattern Recognition, Volume 10828, Page 108281B. International Society for Optics and Photonics; c2018.
7. George Tzanetakis. Perry Cook. Musical genre classification of audio signals. IEEE Transactions on Speech and Audio Processing. 2002;10:293-302.
8. GTZAN. Dataset for Music genre Classification- http://marsyas.info/download/data_sets or
9. https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music- genre-classification

10. BL Sturm. A survey of evaluation in music genre recognition, in Proc. Adaptive Multimedia Retrieval, Copenhagen, Denmark; c2012 Oct.
11. Qiuqiang Kong, Xiaohui Feng, and Yanxiong Li. Music genre classification using convolutionalneural network. In Proc. of Int. Society for Music Information Retrieval Conference (ISMIR); c2014.
12. Douglas Eck, Juergen Schmidhuber. Finding temporal structure in music: Blues improvisation with lstm recurrent networks. In Proceedings of the 12th IEEE workshop on neural networks for signal processing. IEEE; c2002. p. 747-756.