Team 23: CS470 Final Project (Option #1)

Minsung Park, Youngjin Jin, Junghyun Lee

December 1, 2019

Repository link: `https://github.com/iJinjin/CS470_Team23`

1 Introduction

The advent of deep learning has revolutionized many different fields: from natural language processing, bioinformatics, to computer vision. Recently, deep learning has found its way in the field of music information retrieval (MIR) research with growing interest from researchers. While there were 2 deep learning articles in 2010 in ISMIR (International Society for Music Information Retrieval) conferences and 6 articles in 2015, this increased to 16 articles in 2016. This shows the current trend in using deep learning in MIR research.

MIR research is an interdisciplinary science of retrieving information from music that includes:

- Recommender systems
- Track separation and instrument recognition
- Automatic music transcription
- **Automatic categorization**
- Music generation

Here, we consider the problem of **automatic categorization**; specifically, **music genre classification** i.e. classifying the give set of music audio files into different genres such as classical, jazz, hiphop...etc.

2 Audio data representations

2.1 Audio signal

Depending on the deep learning architecture that one utilizes, different "versions" of input sound data are required. Also, an efficient representation of audio data is necessary so that we can easily train it without heavy memory usage/computation.

The "original" representation is a 1-dimensional (discrete) audio signal. It may be appropriate if one uses techniques such as RNN or LSTM, but if one wants to use techniques such as CNN, the data representation has to be processed differently. The point is, the initial input audio data representations do matter, and this section is devoted to providing the necessary theoretical background.

2.2 Spectrum/Spectrogram

Spectrum, or energy spectral density, describes how the energy of a signal (or a time series) is distributed with frequency.

Suppose we have a time signal $x(t)$. Then the energy of the signal is defined as

$$E = \int_{-\infty}^{\infty} |x(t)|^2 dt$$

By Parseval's theorem, we have

$$\int_{-\infty}^{\infty} |x(t)|^2 dt = \int_{-\infty}^{\infty} |\hat{x}(f)|^2 df$$

where $\hat{x}(t)$ represents the Fourier transform of the signal.

Therefore, we can interpret $|\hat{x}(f)|^2$ as a density function describing the energy per unit frequency contained in the signal at the frequency f. From this, the natural definition of the energy spectral density of a signal is

$$S_{xx}(f) = |\hat{x}(f)|^2$$

Note that S_{xx} is a function of frequency, not a function of time. However, the spectral density of small windows of a longer signal may be calculated, plotted versus time associated with the window. Such graph is called a spectrogram. In other words, a spectrogram is a visual representation of the spectrum of frequencies of a signal as it varies with time. A common format is a 2-D representation: one axis for time, one axis for frequency, and the intensity (or color) of each point.

The equation for deriving the spectrogram of a time signal $x(t)$ is

$$\text{spectrogram}\{x(t)\}(\tau, \omega) = |X(\tau, \omega)|^2$$

where $X(\tau, \omega)$ is the discrete-time SFTF(Short-Time Fourier Transform):

$$\mathbf{STFT}\{x[n]\}(\tau, \omega) \equiv X(\tau, \omega) = \sum_{n=-\infty}^{\infty} x[n] w[n - \tau] e^{-jwn}$$

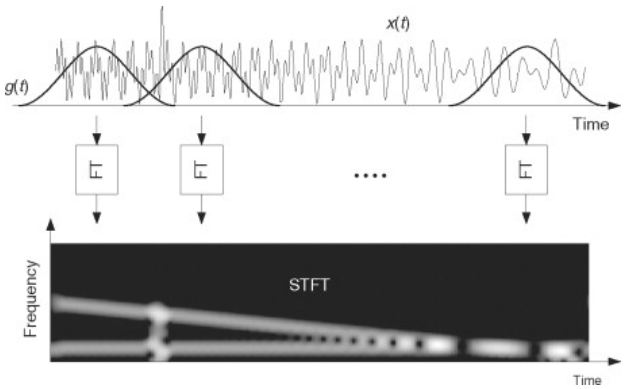

Figure 1: Computing STFT (g is the (sliding) window function)

Here, $\omega[n]$ is the window function, typically a Hann window or Gaussian window. (Window function is a function that is zero-valued outside some chosen interval, and usually symmetric.)

2.3 Mel scale/Mel spectrogram

The Mel scale, introduced by Stevens, Volkmann, and Newman in 1937, is a perceptual scale of pitches judged by listeners to be equal in distance from one another. In other words, the Mel Scale is constructed such that sounds of equal distance from each other on the Mel Scale, also "sound" to humans as they are equal in distance from one another. (Think of the Mel scale as a psychoacoustic scale)

There isn't a "single" Mel scale formula, but the popular formula from O'Shaughnessy's book is as the following:

$$m = 2595 \log_{10} \left(1 + \frac{f}{700}\right)$$

Mel spectrogram is the combination of above concepts: spectrogram with Mel scale as its y-axis.

2.4 Mel-frequency cepstrum (MFC)

The power cepstrum of a time signal $x[n]$, as defined by Bogert *et al.* in 1963, is

$$c[n] = \left|\mathcal{F}^{-1}\left\{\log\left(|\mathcal{F}\{x[n]\}|^2\right)\right\}\right|^2$$

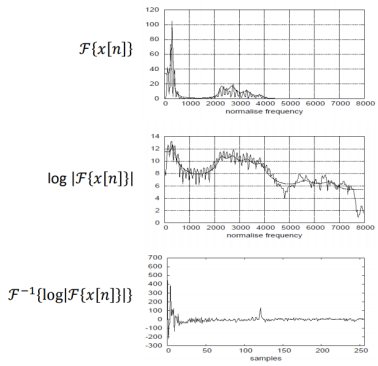

Figure 2: Computing cepstrum

Intuitively, this gives us the information about the rate of change in the different spectrum bands. This is often utilized as a feature vector for representing musical signals.

But in musical signals, the spectrum is usually first transformed using the Mel scale. The result is called the Mel-frequency cepstrum (MFC), and we call its coefficients as the Mel-frequency cepstral coefficients (MFCCs). In the MFC, the frequency bands are equally spaced on the Mel scale, which approximates the human auditory system's response more closely than the linearly-spaced frequency bands used in the normal cepstrum. This frequency warping can allow for better representation of sound, for example, in audio compression.

MFCCs of $x[n]$ are commonly derived as follows (with some variantions in the process, depending on the task):

1. Take the Fourier transform of a windows excerpt of $x[n]$
2. Map the powers of the obtained spectrum onto the Mel scale (using triangular overlapping windows)
3. Take the logs of the powers at each of the mel frequencies
4. Take the discrete cosine transform of the list of mel log powers
5. The MFCCs are the amplitudes of the resulting spectrum

Summarizing above using mathematical expressions:

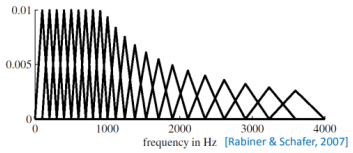

Figure 3: $V_r[k]$

where

$$MFCC[m] = \frac{1}{R}\sum_{r=1}^{R}\log(MF[r])\cos\left[\frac{2\pi}{R}\left(r+\frac{1}{2}\right)m\right]$$

$$MF[r] = \left(\sum_{k=L_r}^{U_r}|V_r[k]|^2\right)^{-1}\sum_{k=L_r}^{U_r}|V_r[k]X(n,k)|$$

($V_r[k]$ is the triangular weighting function for the r-th filter, ranging from DFT index L_r to U_r. Usually, $MFCC[m]$ is evaluated for a number of coefficients N_{MFCC} that is less than the number of mel-filters R)

2.5 Spectral statistics

2.5.1 Spectral centroid

Spectral centroid indicates where the "center of mass" of the mass of the spectrum is located *i.e.* it represents *at which frequency the energy of a spectrum is centered upon.*

$$f_c = \frac{\sum_k S(k)f(k)}{\sum_k S(k)}$$

where $S(k)$ is the spectral magnitude at frequency bin k, $f(k)$ is the frequency at bin k.

2.5.2 (Octave-based) Spectral Contrast

MFCC is a representation of spectral characteristics of music that averages the spectra in each sub-band and reflects the average spectral characteristics. (It is sometimes called the average spectral envelope) But MFCC cannot represent the relative spectral characteristics in each sub-band, which might be more important in our music genre classification problem.

Octave-based Spectral Contrast feature considers the strength of spectral peaks and spectral valleys in each sub-band separately to represent the relative spectral characteristics. For most music, the strong spectral spectral peaks roughly correspond with harmonic components; while non-harmonic components, or noises, often appear at spectral valleys. This is why the *Spectral Contract feature roughly reflects the relative distribution of harmonic and nonharmonic components in the spectrum*, while MFCC loses that information.

To make this point more clear, consider two spectra with different spectral distribution and similar average spectral characteristics. Then, MFCC won't be able to distinguish between them while Spectral Contrast may perform better.

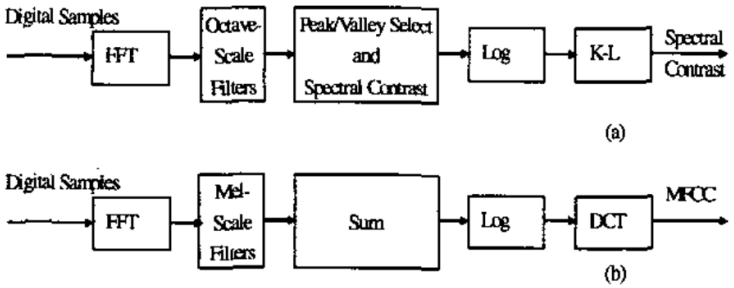

Figure 4: Comparison of (a) Octave-based Spectral Contrast and (b) MFCC

Main differences between the Spectral Contrast and MFCC are...

- The filter bank is different; Octave-based Spectral Contrast uses octave-scale filters while MFCC uses Mel-scale filters. (Although Mel-scale is suitable for general auditory model, octave-scale filter is more suitable for music processing.)
- Spectral Contrast extracts the spectral peaks, valleys, and their differences in each sub-band while MFCC sums the FFT amplitudes
- Spectral Contrast feature uses the Karhunen-Loéve Transform (KLT) while MFCC uses a Discrete Cosine Transform (DCT).

To solve the problem of some relativity among the different dimensions of raw feature, the KLT is performed. Then by the Karhunen-Loéve Theorem, the feature vector is mapped into an orthogonal space, and the covariance matrix is diagonalized. These properties of KLT make the classifying procedure easier and lead to good classification performance, even with simple classifier with Spectral Contrast (Jiang *et al.*).

3 Method

Please refer to our github repository[1] for the code and additional documentation (in README).

3.1 CNN Classification

We created two separate image files (spectrogram and Mel spectrogram) from the GTZAN dataset, which contains 100 wav files, each 30 seconds in length. These spectrograms and Mel spectrograms were generated using the librosa library in python (found in createmelimg.py and createspecimg.py) using a 3 second window. One wav file could generate 10 spectrogram and Mel spectrogram images, bringing the total number of images to 2000 (1000 spectrogram images and 1000 Mel spectrogram images). The spectrogram image file was set to 256×256 resolution, while the Mel spectrogram image file was set to 680×480 resolution. These image datasets were split into training data and test data. Each genre (label) contains 800 training images, and 200 test images. We trained these two models with each of their CNN models. The CNN model architectures can be found in cnn.py. The trained model was then tested with the test data, and the accuracy was recorded.

[1] `https://github.com/iJinjin/CS470_Team23`

3.2 LSTM Classification

We introduced LSTM based classification model for genre prediction. The features of music file can be extracted by librosa package, which contains MFCC, spectral centroid, chroma, spectral contrast. Two-layer LSTM model with 4 features is constructed with 128 unit on first layer and 32 unit for second layer. Using same model architecture, we trained model with 5 features including melspectrogram. Also, we made 3-layer LSTM model, constructed with 128, 64, 32 units, for classification. The specific information of models can be found in .json files. The models trained on GTZAN dataset, divided into train, test and validation with random shuffling.

4 Results

Under GTZAN test data...
CNN using Spectrogram: Using the CNN architecture, the model achieved 38% accuracy.
CNN using Mel Spectrogram: Using the CNN architecture, the model achieved 36% accuracy.
2L-LSTM using MFCC, spectral centroid, chroma, spectral contrast: The model achieved 39% accuracy.
2L-LSTM using MFCC, spectral centroid, chroma, spectral contrast, Mel Spectrogram: The model achieved 34% accuracy.
3L-LSTM using MFCC, spectral centroid, chroma, spectral contrast, Mel Spectrogram: The model achieved 41% accuracy.

5 Discussion

While classification of GTZAN test data using CNN did not yield completely random results (with accuracy 10%), it certainly did not perform as well as we expected. We expected classification using Mel Spectrogram to yield a better accuracy compared to the regular Spectrogram, as the Mel Spectrogram gives a better representation of what humans hear. There are two major points that we would like to address about the subpar performance of our CNN model. One is that we may not have found the ideal architecture / hyperparameters to optimize the performance of this approach. We would have liked to use the exact architecture to measure the performance differences between Spectrogram and Mel spectrogram, but the two images had different resolutions, which resulted in the architectures having different kernel / window sizes. Moreover, the use of CNN means that we have essentially transformed an audio genre classification problem into an image classification problem. Since we are not directly utilizing the audio data itself but instead using an image, this does not directly translate to human's ability to recognize sound and differentiate between genres. From this, we may conclude that CNNs may not be the best approach when it comes to aural data analytics. In LSTM models did not show significant improvement on performance. Even though these 5 features are known to be significant features for representing musical components such as rythm and pitch, it was not enough to overcome the limitations of a small dataset.

6 Contributions

- Minsung Park: Extension of feature extraction from Youngjin's works, implementing codes in LSTMmodel directory and parameter tuning for three LSTM genre classification models.
- Youngjin Jin: Implementing the data processing code for the spectrogram and melspectrogram images, csv file generation, making the overall CNN architecture and code for training and testing CNN data, writing the README file, proof-checking the final report document and writing the results and discussion section of the report
- Junghyun Lee: Topic research, oversaw the theoretical aspects of the project and wrote the background section of the final report.