AUDIO SIGNAL PREPARATION PROCESS FOR DEEP LEARNING APPLICATION USING PYTHON

Mladen Radaković*

Faculty of Electrical Engineering and Computing,
Singidunum University,
Belgrade, Serbia

Abstract:

This paper was created as a part of the project "Development of software to improve communication, academic and social skills of children with disabilities [1]. "
Artificial Intelligence today represents a wide area of different computer algorithms and systems with one main goal – to mimic and eventually replace human thinking and logic.
One of the most important parts of the mentioned software in development is correct and meaningful collection and preparation of relevant data. Deep learning model structure, model training and results heavily depend on thoughtful identification and processing of relevant data.
As a part of a wider project, this paper is representing a short overview of sound record digitalization and recommended steps required in data preparation for use in artificial intelligence applications.

Keywords:

Audio Signals and Processing, Deep Learning, Python.

INTRODUCTION

Artificial intelligence (AI) is the future happening today. Already a part of our lives, in last several years, AI literally exploded worldwide. Significant increase of computer processing power and especially power of graphical processing units that were made available to AI algorithms, applications of these systems increased exponentially.

Artificial Intelligence is a broad definition of computer machines and algorithms trying to mimic and perform human thinking and behaviour. At first, it was applied to development of systems that would simulate intellectual processes as characteristics of humans, like reasoning. Today, we find countless of uses and applications in everyday life, like in web search, internet portal recommendations, chat bots, smart home appliances (vacuum robots, heating, cooling systems, air filters, …), shape recognition systems (cars without drivers, drones, airplanes, …), medical diagnostics, vaccine creation, agricultural applications, solving mathematical problems, autonomous playing of computer games, sound recognition, sound generation, ...

Correspondence:
Mladen Radaković

e-mail:
mladen.radakovic@gmail.com

DOI: 10.15308/Sinteza-2021-146-152

Machine Learning is a part of Artificial Intelligence that is based on statistical models and techniques that are helping machines in decision making. Most often, designer of machine learning systems is deciding which inputs and outputs are going to be used in the models created.

As a separate part of Machine Learning, Deep Learning is often using extremely large datasets and algorithms that can decide on inputs and outputs with no human help. Deep Learning systems are relying only on the model created and "experience" gained thru training large dataset models. One of main characteristics that is differing them from simpler Machine Learning systems is the use of multi-layered neuron networks.

Figure 1 – Artificial Intelligence and fundamental concepts

Using traditional Machine Learning, we would select multiple sound characteristics we prepared in advance and use them as a base to manually create, use, and test the machine learning model algorithm. In this case, results are strongly depending on many human-related decisions. In case of using machine learning systems, developer is selecting sound features to be used by himself.

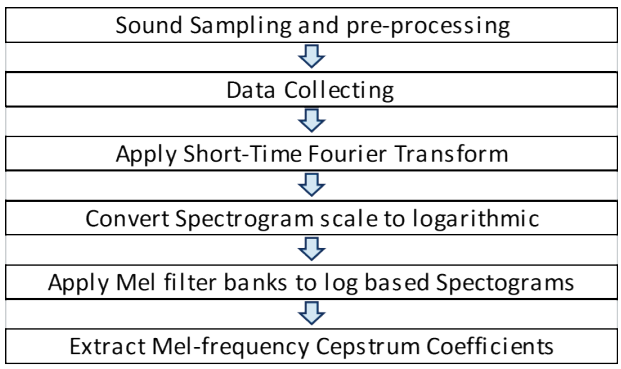

Figure 2 – Audio Signal Preparation Process

Deep Learning (DL) requires large datasets as a base to be able to produce good results. Having large database with data provided to the system, DL algorithms are providing the opportunity to developers not to extract certain coefficients from the sound themselves, but to use complex data as direct inputs. In case of working with sounds, this can be a raw unstructured audio signal, calculated spectrogram, or any other form of sound data record. Deep Learning algorithms can extract relevant audio features automatically, with no additional help or assistance from a human.

2. SOUND SAMPLING AND DATA COLLECTING

Sound is a mechanical signal, accumulation of waves, transmitted via air molecules that are oscillating. As a result of change in air pressure, sound is caused by a vibration of an object. By its nature and type, sound is analogue physical signal.

As computers are working with digital data exclusively - before any action is performed with a sound, we need to convert that analogue signal into a digital one. To use this type of data in any of artificial intelligence systems, we need to have it ready in digital form.

Audio signal waveform is a computerized visual graph, representing a sound. It contains all the details, characteristics, and specifics of a sound. With this type of record, we can reproduce and generate the original sound anytime (Figure 6).

To be able to work with any signal using a computer, we must digitize it and prepare it for computer processing. After digitalization is completed, with use of algorithms and mathematical methods we can extract all required characteristics and specifics for further signal processing.

2.1. SOUND SAMPLING

Digitalization process, depending on sampling quality and rate, can provide larger or smaller datasets.

Analogue sound signals are continuous in frequency and in time. Depending on digitalization (sampling) process, certain quantity of data describing the signal will be lost. Level of loosing signal data is directly proportional to digitalized quality of the signal.

Inside analogue signal, between two values, we have countless number of values describing it. After we convert it to digital form, we are creating a list of finite number of elements.

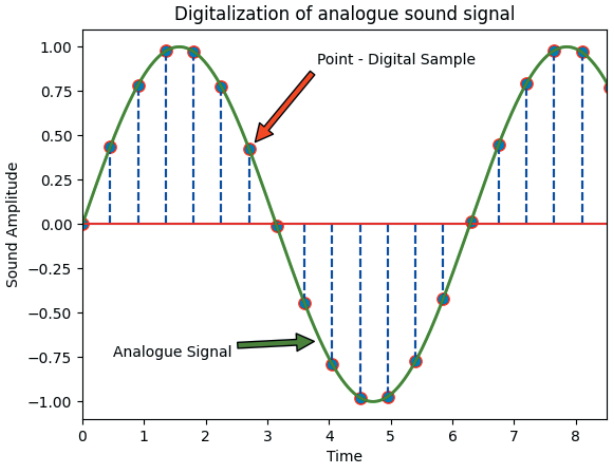

Figure 3 – Sampling: conversion from analogue to digital.

One of the most important tasks for a developer is to use a proper method of sampling generating quantity of information that would be sufficient for meaningful further mathematical processing.

Digitalization is based on defining fixed time periods for signal sampling, but also on defining fixed steps in frequency for quantization of signals. Number of computer bits used for defining quantization levels are called "resolution" or often "bit depth".

After we convert analogue signals to a digital form, we can save them permanently on a computer storage device. Keeping this data safe sometimes can become a problem in case we did not consider the size of it. This is one of especially important facts we need to consider when we decide to start sampling sound records. We can decide keeping the data without any loss or compressing it to save some space.

Based on the approximate dataset size, we should always take into consideration required storage space before we start creating a database. Calculations are based on several important factors:

```
S – Required digital space for preserving sound
data.
P – Sampling period (Standard CD quality
frequency rate is 44.100 Hz).
B – Number of quantization levels in bits
(for 16 levels, we are required to use 4
bits).
T – Time recording period in seconds.
Example:
S = P x B x T
S = 44.100 Hz x 4 bits x 60 s
S = 10.584.000 bit = 1.26 MB
```

Equation 1 – Required space based on sampling details.

Quantization of analogue signal is also resulting with a data loss, that needs to be considered during sampling process. Depending on future data use and its application, it needs to be decided on the number of sampling frequency levels, meaning, quality of the future digital signal.

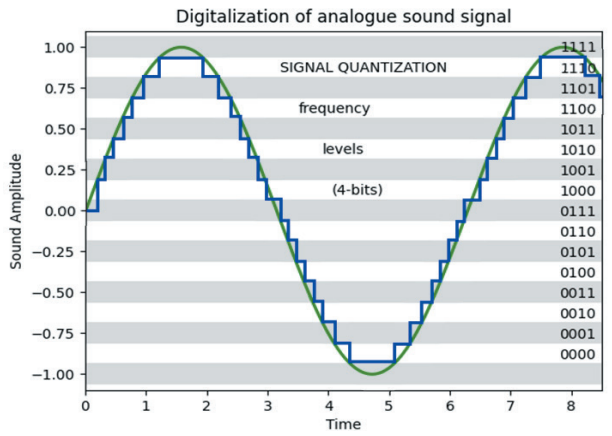

Figure 4 – Quantization resolution in this example is 4.

In the figure 4, we showed quantization with 16 different frequency levels, which means that future frequency values will be only one of pre-defined ones – defined by 4 digital bits. Like in figure 3, where the same rule applies for time segment of the sound sample.

2.2. DATA COLLECTING

Digitized sounds on computers can be presented visually as waveform graphs. Basically, sounds are made of multiple simple and complex sound characteristics, that are describing it. Some of the most common parameters, describing sounds, used in sound processing are frequency, period, amplitude, phase, pitch, cents, intensity, timbre, loudness, …

Some additional, most often used, relevant more complex sound features we can use for sound data processing are amplitude envelope, zero crossing rate, root-mean-square energy, …

One of the crucial decisions to be made by a machine learning model developer is related to data collection. Using Deep Learning algorithms, we need to provide largest database possible. The more data we have, we will be able to create a better and more suitable model for a future system. As Machine Learning depends heavily on data, without data, it is impossible for the "AI" system to "learn".

In most cases, we are not building separate databases, but rather finding sources and pointing our algorithms to them [2].

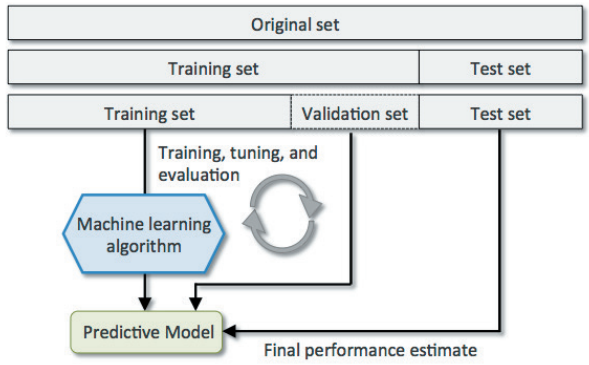

Figure 5 – Dataset use in Machine Learning.

Most crucial aspect of AI is dataset as it makes algorithm training possible. After creation of ML model, we train it, validate it and finally, test the set we created.

3. EXTRACTING AUDIO FEATURES FROM SOUND FOR MACHINE LEARNING

Following digitalization of analogue audio signal, best step to go forward with it is to extract most important characteristics from the sound and prepare them for further processing.

Basic characteristics defining audio signals - like tempo, frequency, noise, intensity etc. we extract using specific mathematical methods. Many characteristics of sound waves are defining them and making them unique. Today, there are many tools available to developers, starting from programming languages like C or Python, all the way to commercial packages for building Machine Learning based systems.

Tools used in developing software in the project "Development of software to improve communication, academic and social skills of children with disabilities. [1]" are based on programming language „Python" and available modules like „Librosa" [3].

In majority of cases, AI manipulation with sound data is happening using systems and algorithms for comparing photos. Sound data with its characteristics is being translated into coefficient-based diagrams, and after that, processed using ML models.

Like in other applications, using artificial intelligence systems (machine learning or deep learning) on sound signals can replace humans almost completely. In many applications, computers are working with this kind of data far more efficient than people.

Using AI systems with sound can enable automatic song classification, speech recognition, diagnostics and analysing words and sentences (in education, medical applications, or rehabilitation), song identification, instruments recognition, emotional analysis based on speech, …

3.1. DISCRETE FOURIER TRANSFORM

Before we apply any sound processing algorithm, it is required for the sound wave to be pre-processed. This means that the sound was trimmed, normalized and noise was filtered out as much as it was possible.

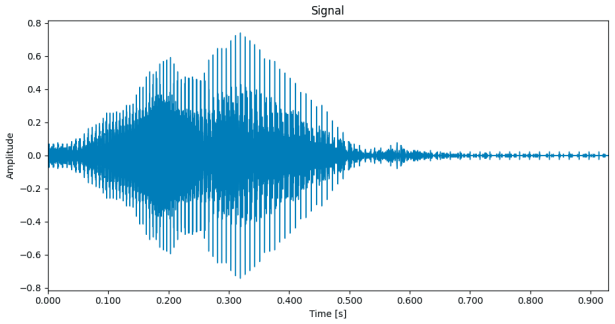

Figure 6 – Sound Waveform

By Fourier, every audio signal is made from several single frequency sound waves. On figure 5, we can see a characteristic sound waveform [4], where on x-axis we have time, and on y-axis is amplitude of the signal (Hz).

Discrete Fourier Transform (DFT) is a mathematical formula that, when applied, is moving a soundwave from the time domain into the frequency domain [4]. Fourier proved that every sound signal can be decomposed into simple sine and cosine waves that when added up and then again creating the original signal.

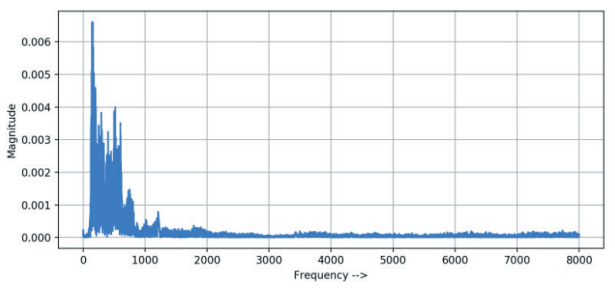

Figure 7 – Fourier Transformation applied to a waveform.

Fast Fourier transform (FFT) is an algorithm that is computing Fourier transform fast and efficiently. As many other features, for ease of use, it is incorporated into Librosa and many other Python modules.

After we apply this transformation to a sound wave, we call the result "spectrum" (as in figure 6). This spectrum is representing frequency representations in the signal, but it does not represent them in time.

3.2. SHORT-TIME FOURIER TRANSFORM

To add a time component to that diagram, we need to apply Short-Time Fourier Transform algorithm onto data. This algorithm is taking segments of the signal, computes FFT on overlapping windowed segments and creates a separate spectrogram for every processed segment. These segments are being joined into one large dataset. Result of this operation is a matrix of data that represents complex future spectrogram data with time component included.

Having Short-Time Fourier Transform (STFT) data in place, having both time and frequency data in it, we can visually present them in a human-acceptable manner in a form of a "Spectrogram".

Having Python as a programming language of choice, for our project, we were using one of the best existing libraries for sound management and manipulation - "Librosa" [3]. Using that library, we can easily create spectrograms, extract relevant audio data, and get other important sound characteristics.

3.3. SPECTROGRAM

Matrix of data created by applying Fast Fourier Transform formula on windowed parts of sound signal is a base for creating a visual representation named "spectrogram [5]".

Spectrograms are visually representing sound in linear frequency Hertz scale in time. This representation is mathematically correct, but it is not corresponding to human perception of sound. Humans can hear sounds between 20 Hz and 20.000 Hz, but if we compare human perception of difference between frequencies in lower and higher range – it is not the same.

Same difference in frequency (Hz) in lower range humans will not be able to compare to the same frequency difference in the higher range. We tend to be better at detecting sound differences in lower frequencies than higher.

As an example, we can easily detect a difference between 3.000 Hz and 4.000 Hz but will hardly tell the difference between 13.000 Hz and 14.000 Hz.

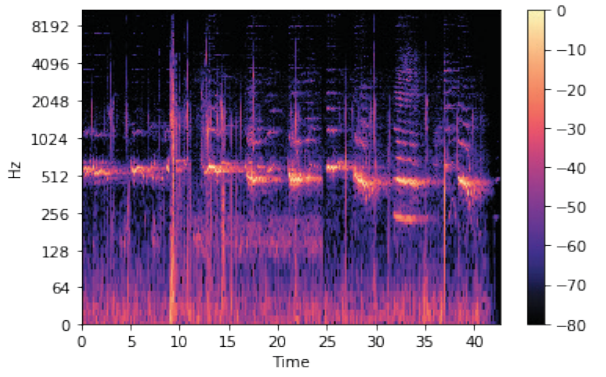

Figure 8 – Spectrogram

Spectrograms are graphically showing frequency representations of the sound taking in consideration time. They are significantly better than a simple Fourier transform algorithm applied to a data record. Anyhow, the downside of it is their frequency scale that is shown in Hertz.

Converting amplitudes from Hz values into dB values by applying logarithmic (log) function to data, we get a human perception like diagram.

3.4. MEL-SPECTROGRAMS

People perceive sound frequency logarithmically. Therefore, visual representation of spectrograms is more human readable if we apply logarithmic scale to it and present it in a different way.

Perceptual scale of pitches (judged by listeners) to have them equally distanced from one another is called Mel scale. Proposed by Stevens, Volkmann and Newmann in 1937, mathematical operation on frequencies is converting frequency in Hz to pitch in "Mels" and getting logarithmic scale as a result.

Optionally, applying Mel filter banks, we can additionally filter sound source and accommodate it for further mathematical use. Converting regular sound power spectrum into Mel Scale is fundamentally important in Machine Learning as it is mimicking human perception of sound.

A popular formula for converting f [Hertz] into m [Mels] is:

$$f_{Mel} = 2595 \log_{10} (1 + f / 700)$$

Spectrogram with amplitudes converted to dB (decibels) are a good base for creating mel-spectrograms. We do it by applying "mel" filter banks [6].

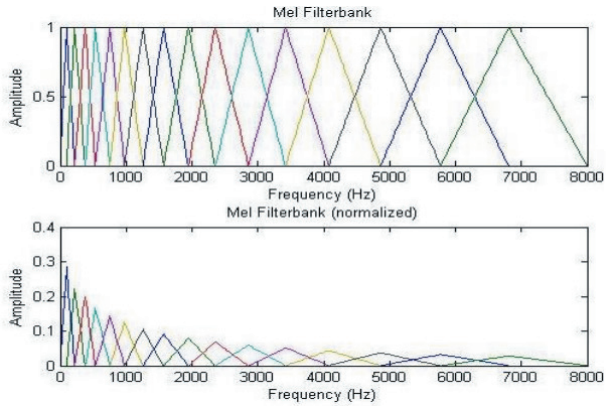

Figure 9 – Mel filter bank

Applying these filters, result is a sound representation that is mimicking human perception of sound. As the graph looks almost the same, frequencies on the y-axis are converted into mel-scale and on the x-axis, we keep the time.

3.5. MFCCS

Representation of the short-term power spectrum of sound, in linear cosine transform of a log power spectrum on a nonlinear mel-scale of frequency is called mel-frequency Cepstrum (MFC).

MFCC stands for Mel-frequency Cepstrum and Coefficients. Cepstrum is representing the information of rate of change in spectral bands.

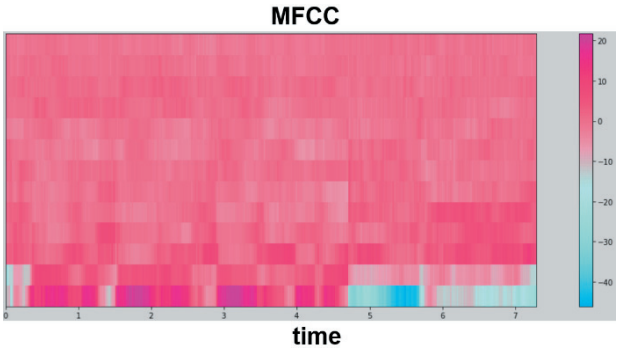

Figure 10 – MFC Coefficients

Discrete Cosine Transform (Fast Fourier Transform) is significantly simpler than Fourier Transform, providing non-complex (real value) numbers as coefficients. As for Deep Learning applications complex numbers are not acceptable, and Discrete Cosine Transform is providing acceptable results, we use it for further operations.

Mel-Frequency Cepstral Coefficients (MFCCs in short) are coefficients that collectively build up an "Mel-frequency Cepstrum" (MFC). They represent and contain sound characteristics that are very suitable for further machine or deep learning processing.

Out of total 39 MFC coefficients, mostly used are the first 12-13 coefficients as they contain and keep most information about the sound. After selecting MFCCs we are going to use in our application, we can also visualize them in coefficient/time graph.

3.6. ALTERNATIVES TO MFCCS

There are many MFCC advantages and some disadvantages. Using MFCCs we can describe large parts of spectrum, we can ignore fine spectral structures that could have negative influence on our data, and they are proved to be excellent working with music (genre classification, automatic tagging, recognition, …), and speech (speech recognition, person recognition, gender recognition, etc).

MFCCs are not good with synthesis of sound, they are overly sensitive to noise and overly complicated to use.

Apart from mostly used Mel-frequency cepstral coefficients, some of the most used algorithms for extraction of sound characteristics and analysis are:

- Linear Prediction Coefficients (LPC),
- Linear Prediction Cepstral Coefficients (LPCC),
- Line Spectral Frequencies (LSF),
- Discrete Wavelet Transform (DWT),
- Perceptual Linear Prediction (PLP), …

Depending on data source and application, artificial intelligence model designer should select most suitable algorithm model for extracting sound characteristics. Most often this is based only on experience and knowledge on the sound in AI.

4. CONCLUSION

Artificial intelligence as a scientific area is finding its way into all parts of our lives. Every day it is altering the world and we are finding new and better ways to use it in society, economy, governance, engineering, education, agriculture, etc.

As one of the most influential innovations in human history, we should all take the opportunity and find a way to apply it in our lives and generate benefits for all.

REFERENCES

[1] M. Radakovic, S. Nestorov and K. Radakovic, "Artificial intelligence and computers as assistance in school and extracurricular education of children with developmental disabilities," in *International Conference: Multidisciplinary Approaches in Education and Rehabilitation*, Sarajevo, 2021.

[2] A. Gonfalonieri, ""How to Build A Data Set For Your Machine Learning Project"," 2019. [Online]. Available: https://towardsdatascience.com/how-to-build-a-data-set-for-your-machine-learning-project-5b3b871881ac. [Accessed 09 06 2021].

[3] McFee, Brian, Colin Raffel, Dawen Liang, Daniel PW, "Librosa: Audio and music signal analysis in python.," in SciPy, 2015.

[4] K. Chaudhary, "Understanding Audio data, Fourier Transform, FFT and Spectrogram features for a Speech Recognition System," 19 Jan 2020. [Online]. Available: https://towardsdatascience.com/understanding-audio-data-fourier-transform-fft-spectrogram-and-speech-recognition-a4072d228520. [Accessed 09 06 2021].

[5] M. Pasini, "Voice Translation and Audio Style Transfer with GANs," 6 Nov 2019. [Online]. Available: https://towardsdatascience.com/voice-translation-and-audio-style-transfer-with-gans-b63d58f61854. [Accessed 09 06 2021].

[6] N. V. Parinam, C. Vootkuri and S. A. Zahorian, "Comparison of Spectral Analysis Methods for Automatic Speech Recognition," *INTERSPEECH*, p. 3357, 2013.