Published in Towards Data Science

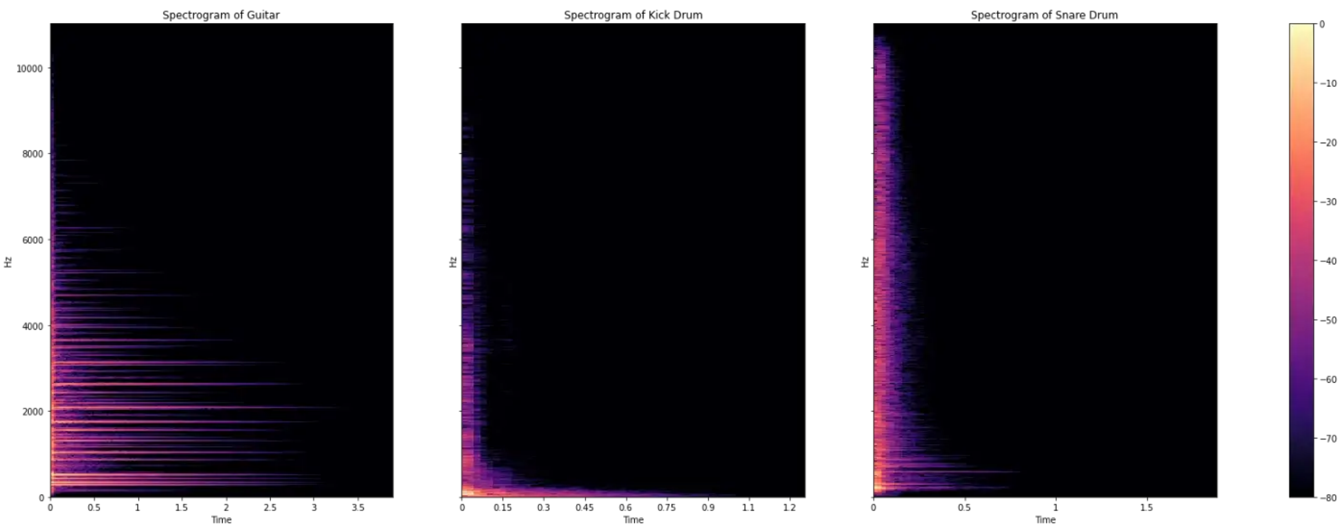
mlearnere Follow

Jan 26, 2021 · 4 min read · · Listen



# Learning from Audio: Spectrograms

Visualizing the structure of audio down to the millisecond.




By the end of this article, you will be able to create figures like these. Image by Author.

## Introduction:

When it comes to Machine Learning, or even Deep Learning, how the data is processed is fundamental to the model's training and testing performance. When working in the domain of audio, there are a few steps to understand before getting to this stage, but once you get there, learning from audio becomes a quite easy task. Make sure to understand the concepts linked below before going through this article.

**Related Articles:**

- Learning from Audio: Wav

- Learning from Audio: Time Domain Features
- Learning from Audio: Fourier Transformation
- Learning from Audio: The Mel Scale, Mel Spectrograms, and Mel Frequency Cepstral Coefficients
- Learning from Audio: Pitch and Chromagrams

In this article I aim to break down what exactly a spectrogram is, how it is used in the field Machine Learning, and how you can use them for whatever problem you are attempting to solve.

**As always, if you would like to view the code, as well as the files needed to follow along, you can find everything on my GitHub.**

## What is a Spectrogram?

You can think of spectrograms as pictures of audio. Kind of weird, I know, but you should strengthen this intuition as much as you can. The *spec* portion in spectrogram comes from spectrum and the color-bar you see on the right of the figure is just that. What is the spectrum of? The frequencies in which the audio has.

With all of this information in mind, let me formalize the definition.

> A spectrogram is a figure which represents the spectrum of frequencies of a recorded audio over time.

This means that as we get brighter in color in the figure, the sound is heavily concentrated around those specific frequencies, and as we get darker in color, the sound is close to empty/dead sound. This allows us to get a good understanding of the shape and structure of the audio without even listening to it! This is where the power of spectrograms come into play for various ML/DL models.

## How to Create Spectrograms:

Now a question arises, how do we calculate the spectrograms? The answer to this question is much simpler than expected.

- Split the audio into overlapping chunks, or windows.
- Perform the Short Time Fourier Transformation on each window. Remember to take its absolute value!
- Each resulting window has a vertical line representing the magnitude vs frequency.
- Take the resulting window and convert to decibels. This gives us a rich image of the sound's structure.
- Finally, we lay out these windows back into the length of the original song and display the output.

Now that we have a decent understanding of what spectrograms are exactly, let's learn how to retrieve them from sounds in Python! Using functions in `librosa` , we can have this be done for us with little to no effort.

First, let's import the needed packages and load in the audio.

Second, I am going to define two functions; one that will perform all the necessary steps and output the processed signal, and another that will plot the spectrogram. Make sure to read through the comments and lines to understand the process in which this is done.

Now, with our functions defined, we can simply use `plot_spec` to graph the results!

```
plot_spec(to_decibles(guitar), sr, 'Guitar')
```

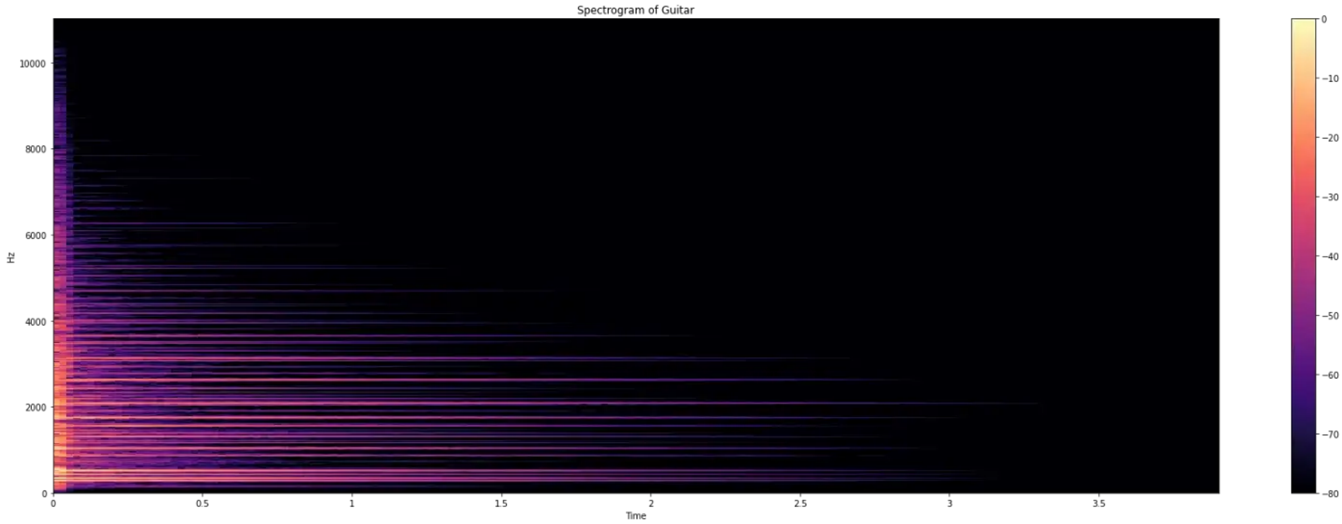



Figure by Author.

```
plot_spec(to_decibles(kick), sr, 'Kick')
```

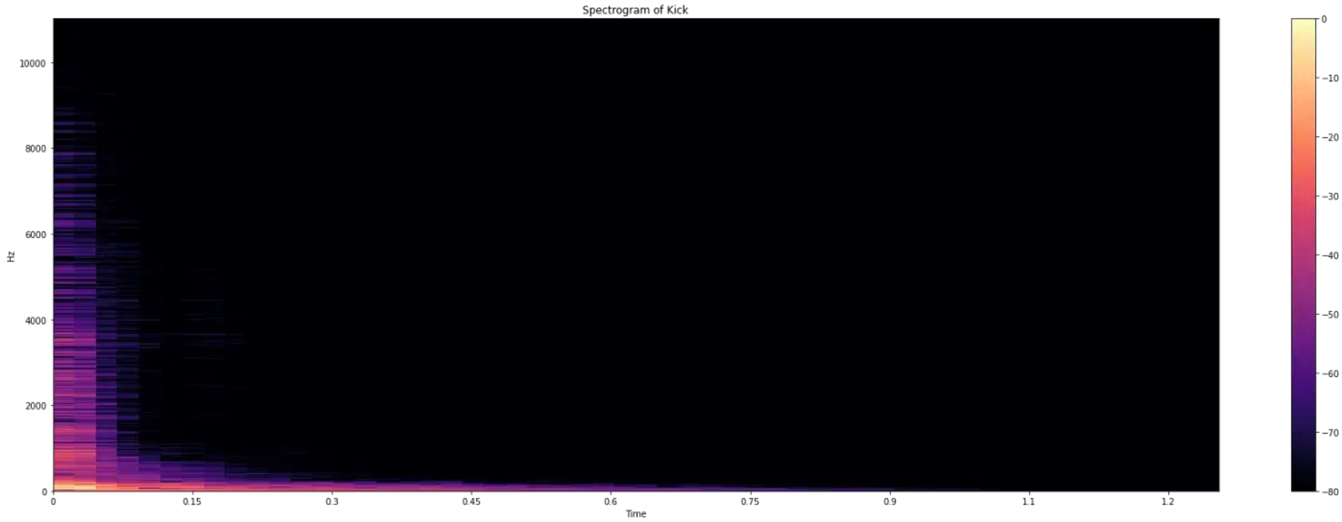



Figure by Author.

```
plot_spec(to_decibles(snare), sr, 'Snare')
```

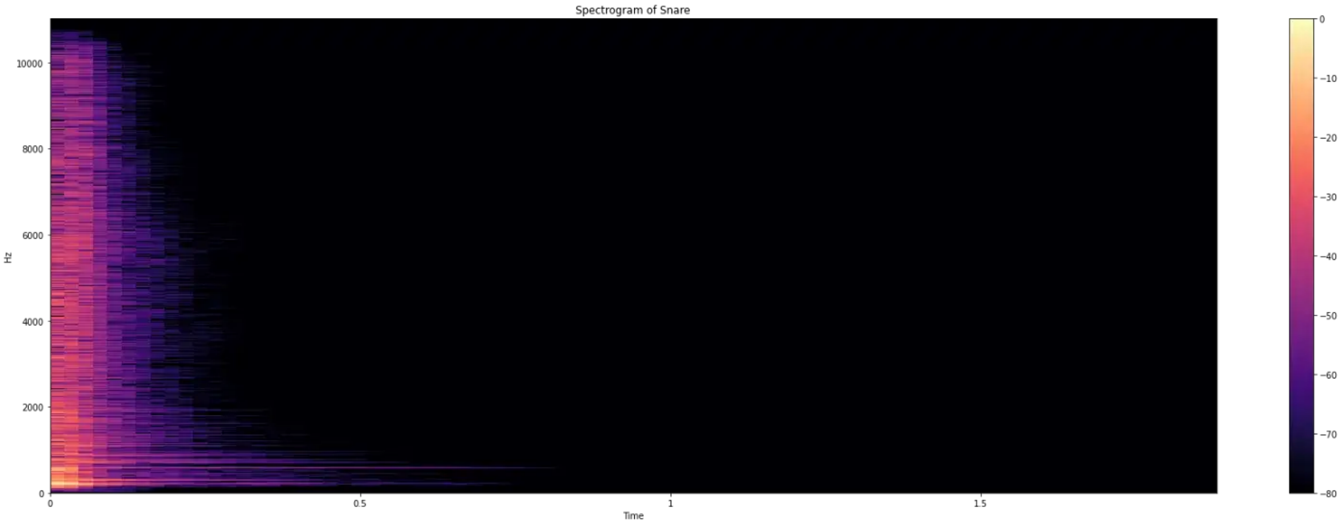



Figure by Author.

## Conclusion:

By now, you should be able to understand how spectrograms are created using the Short Time Fourier Transformation, as well as how to create them in Python. These representations of audio allow for various Deep Learning architectures to extract features much easier than wave form and even Fourier representations.

Stay tuned for even more spectrogram representations of audio.

Thank you for reading.

Signal Processing Deep Learning Data Science Machine Learning Audio