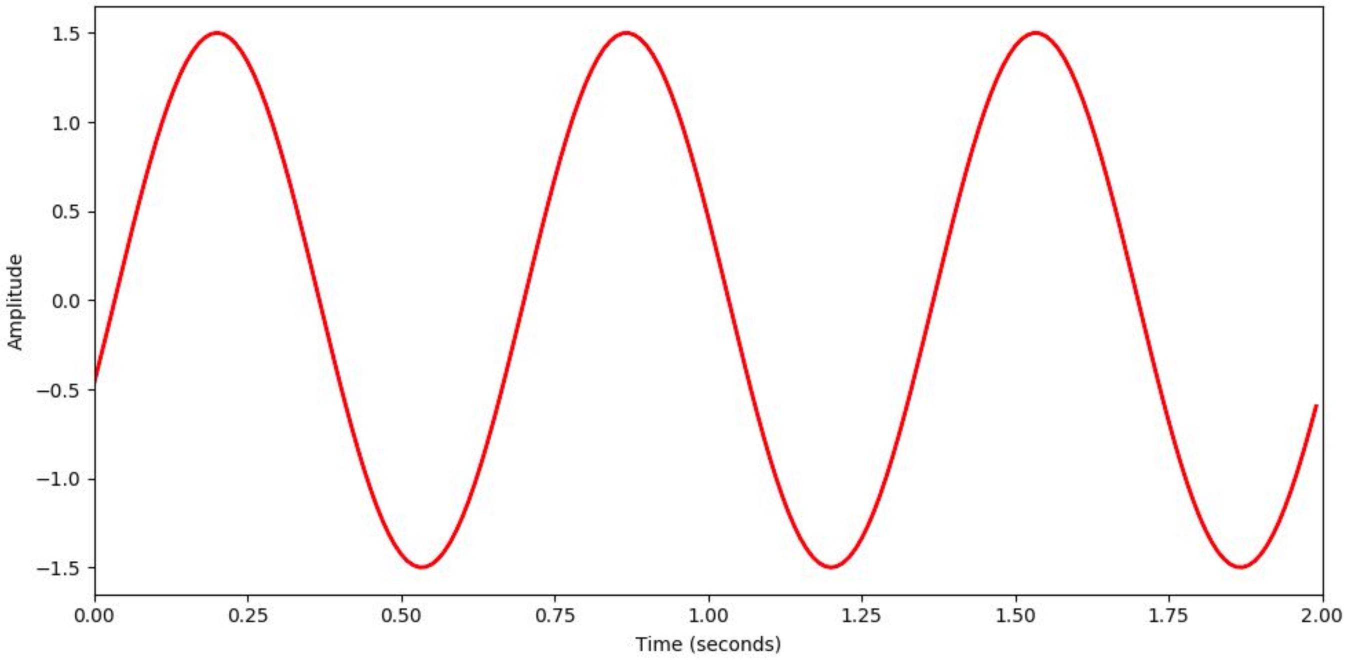
Understanding audio data for deep learning

Valerio Velardo

Sound

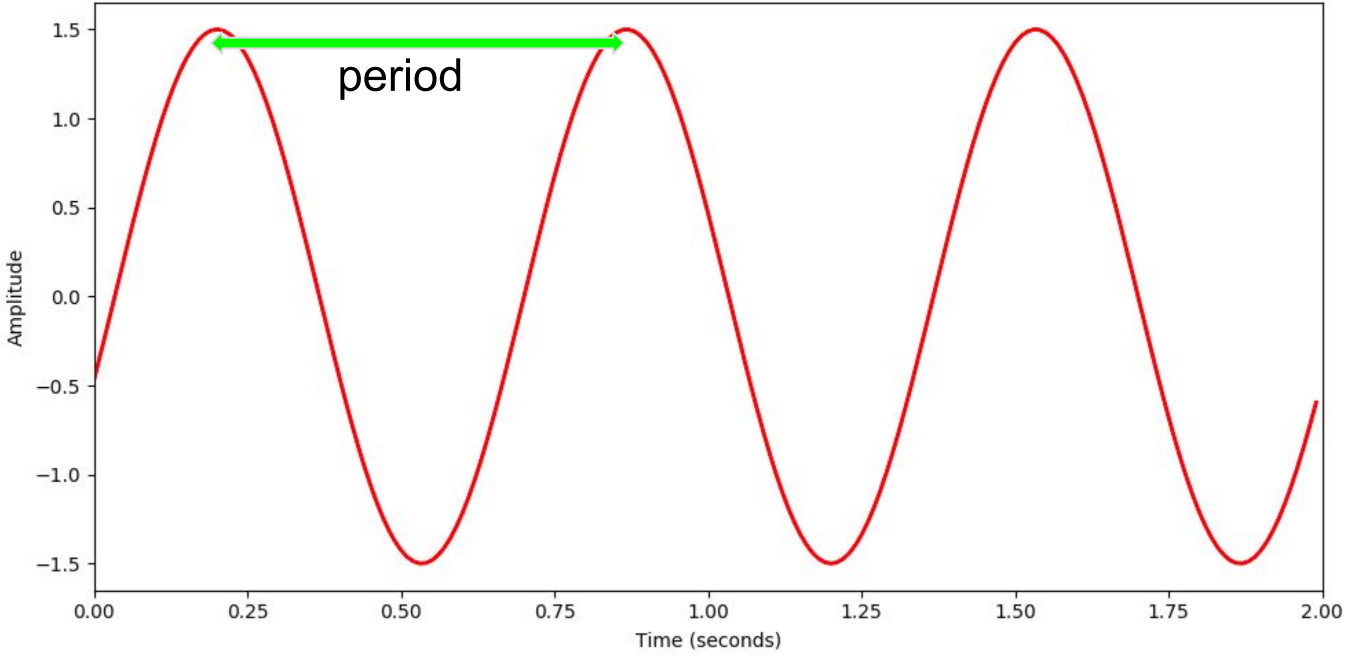
- Produced by the vibration of an object
- Vibrations determine oscillation of air molecules
- Alternation of air pressure causes a wave

Waveform



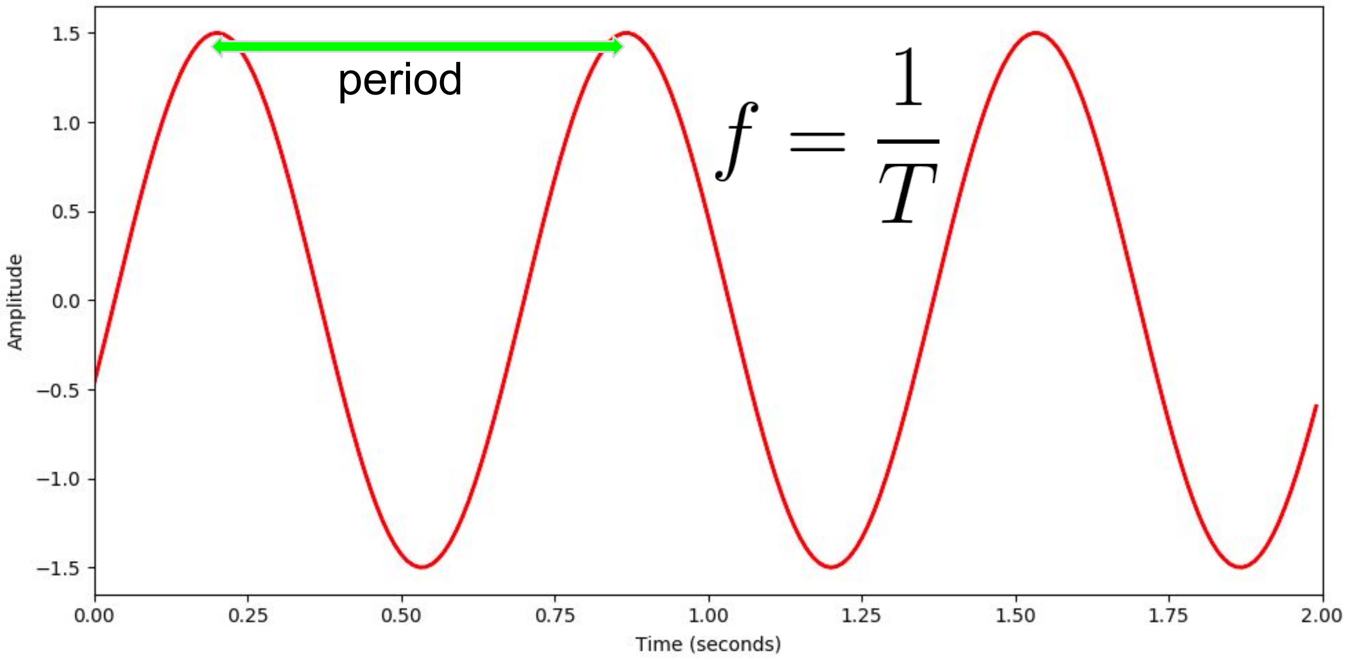
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
Amplitude
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
Time (seconds)

Waveform

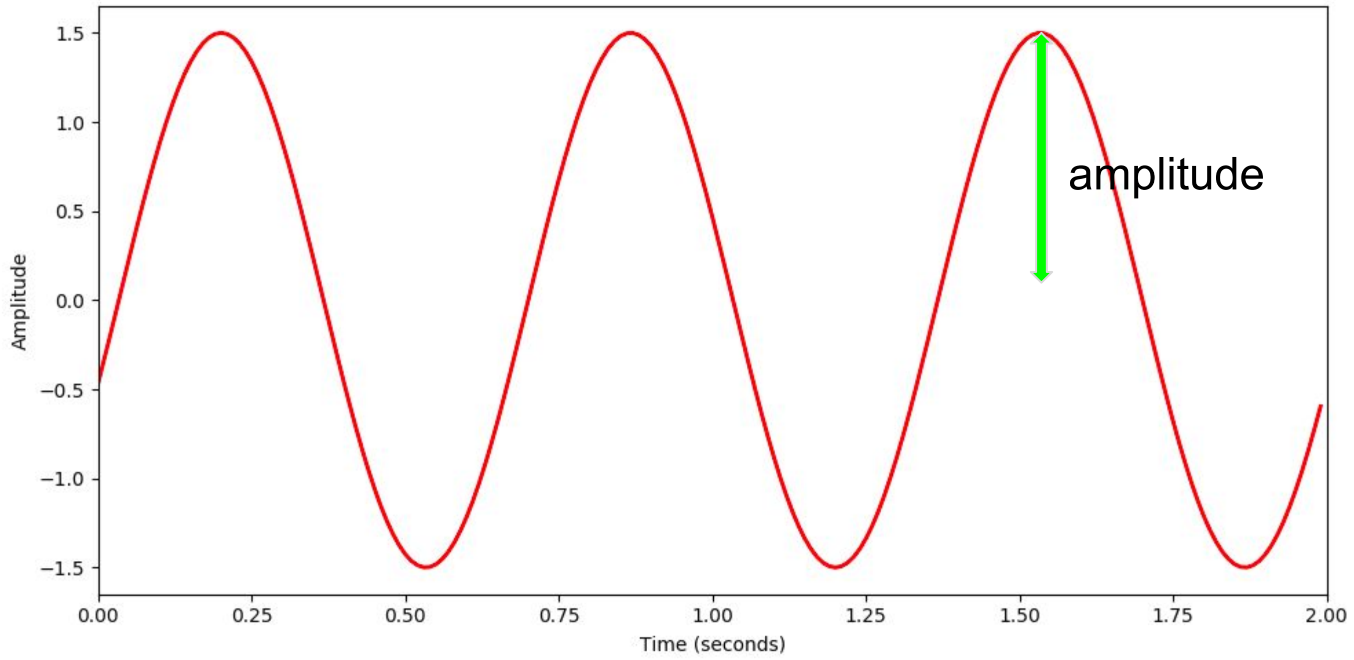


period
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
Amplitude
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
Time (seconds)

Waveform

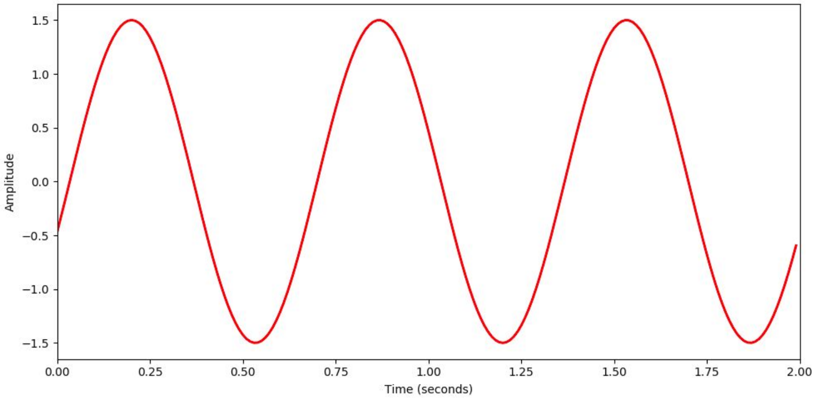

period
$f = \frac{1}{T}$
Amplitude
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
Time (seconds)

Waveform



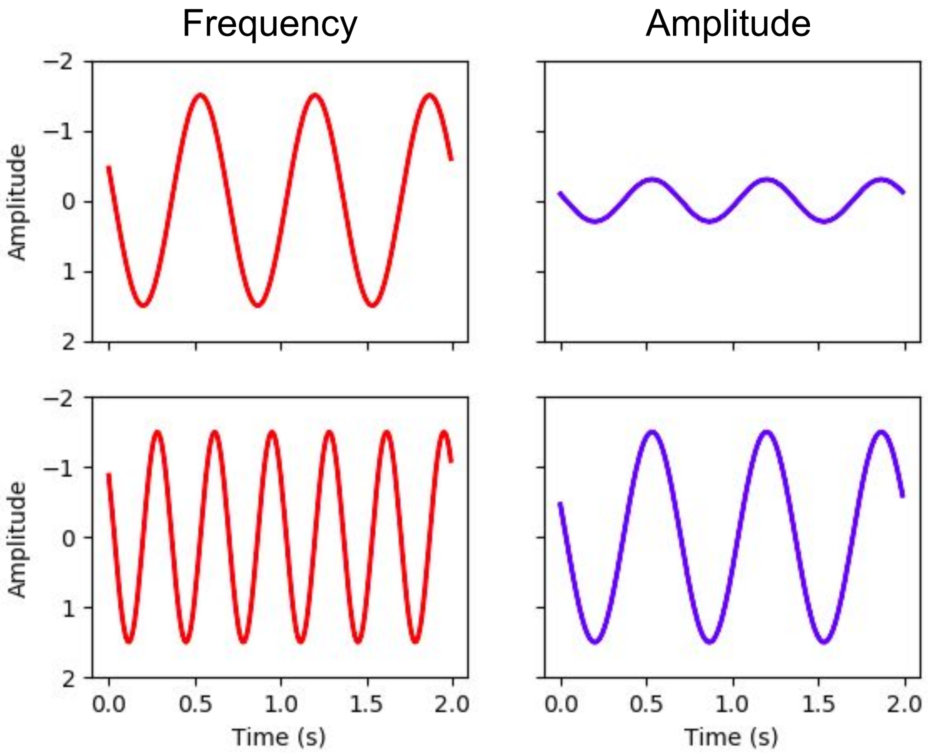
amplitude
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
Amplitude
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
Time (seconds)

Waveform

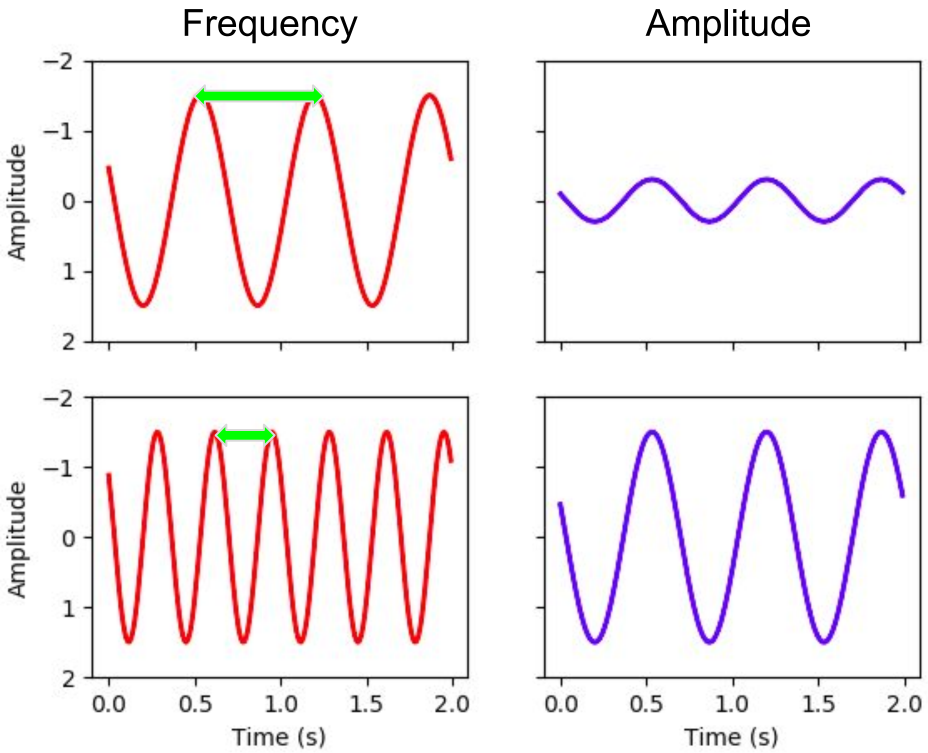


$$y(t) = A\sin(2\pi f t + \varphi)$$

Frequency/pitch and amplitude/loudness

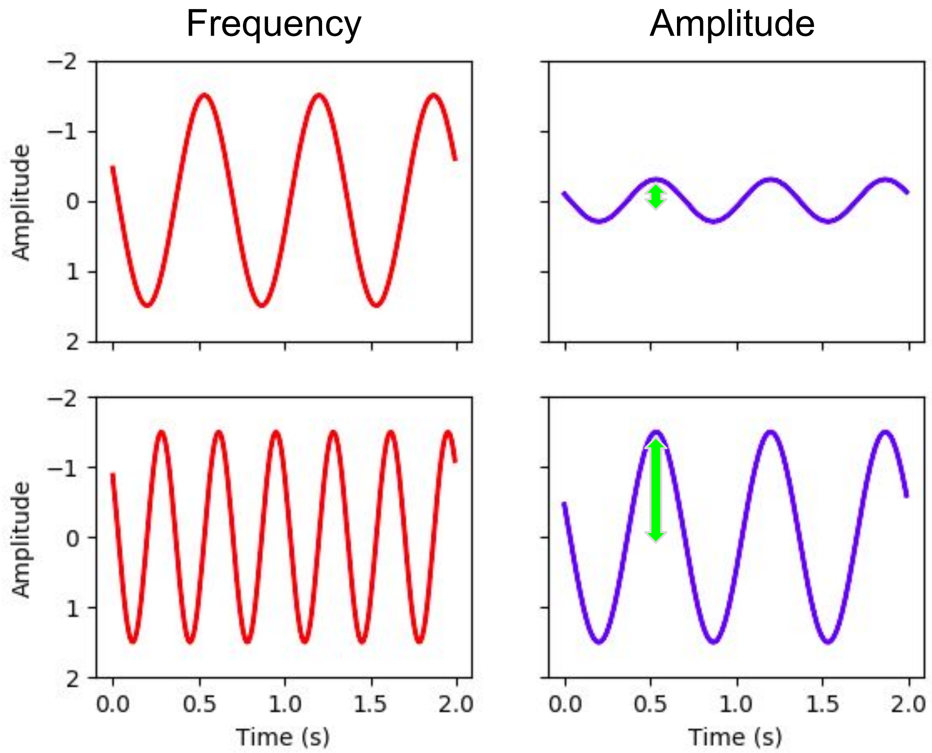


Frequency/pitch and amplitude/loudness



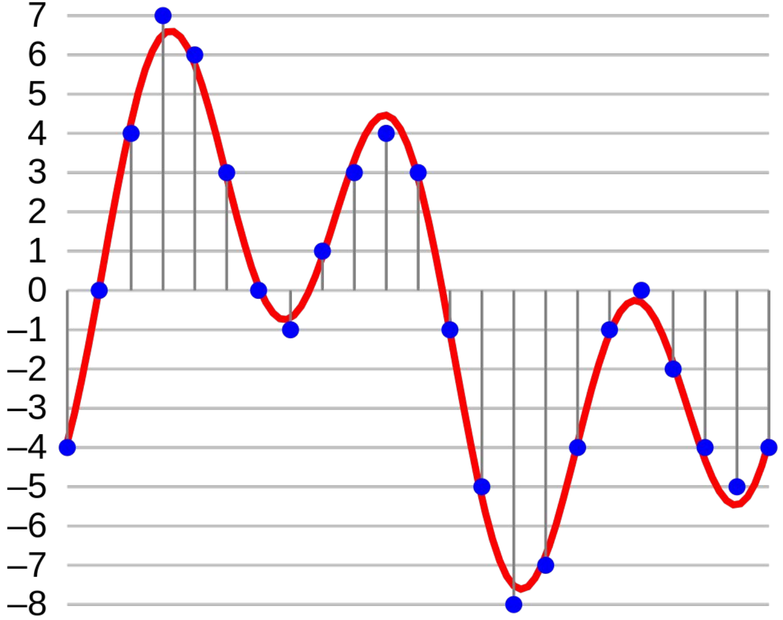
higher frequency -> higher pitch

Frequency/pitch and amplitude/loudness



larger amplitude -> louder

Analog digital conversion (ADC)



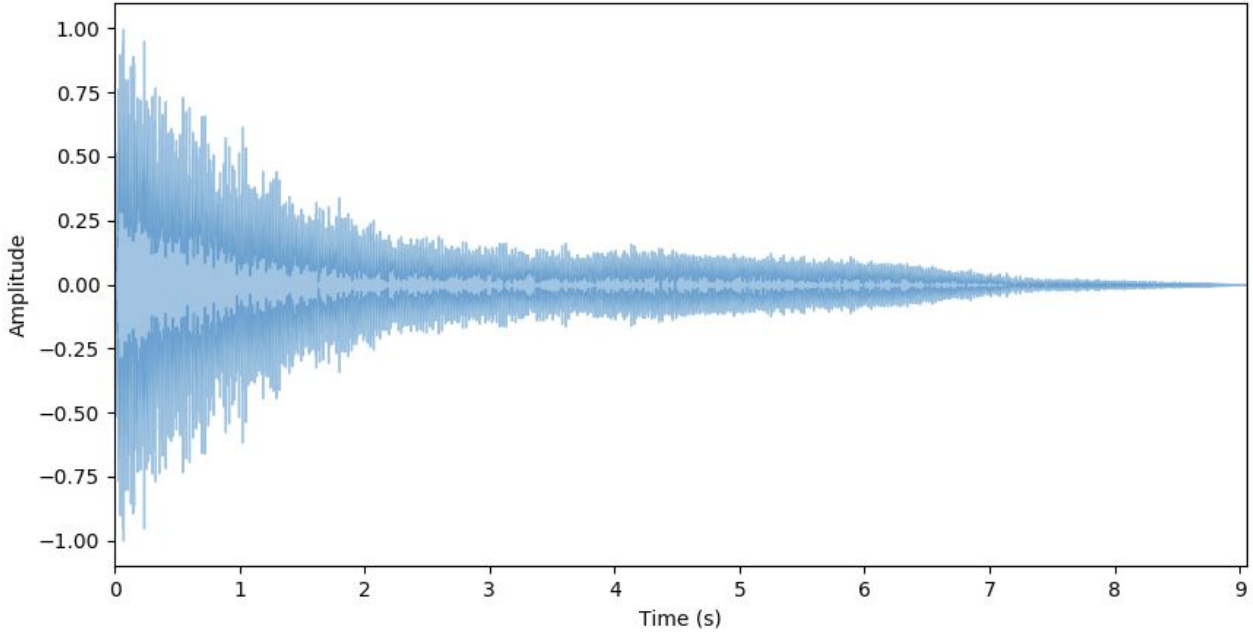
- Signal sampled at uniform time intervals
- Amplitude quantised with limited number of bits

Analog digital conversion (ADC)

Sample rate = 44,100 Hz

Bit depth = 16 bits/channel

A real-world sound wave (piano key)

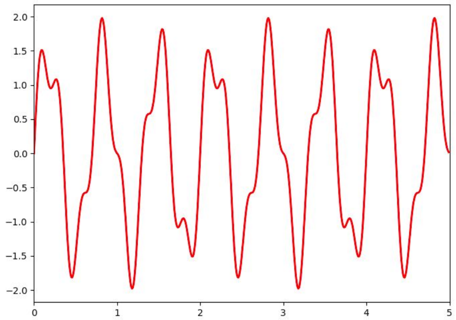


Fourier transform

- Decompose complex periodic sound into sum of sine waves oscillating at different frequencies

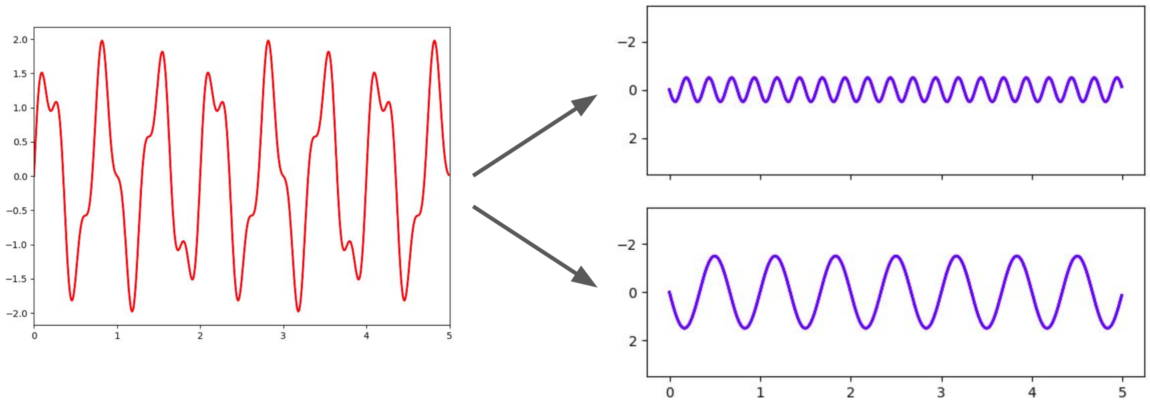
Fourier transform

- Decompose complex periodic sound into sum of sine waves oscillating at different frequencies



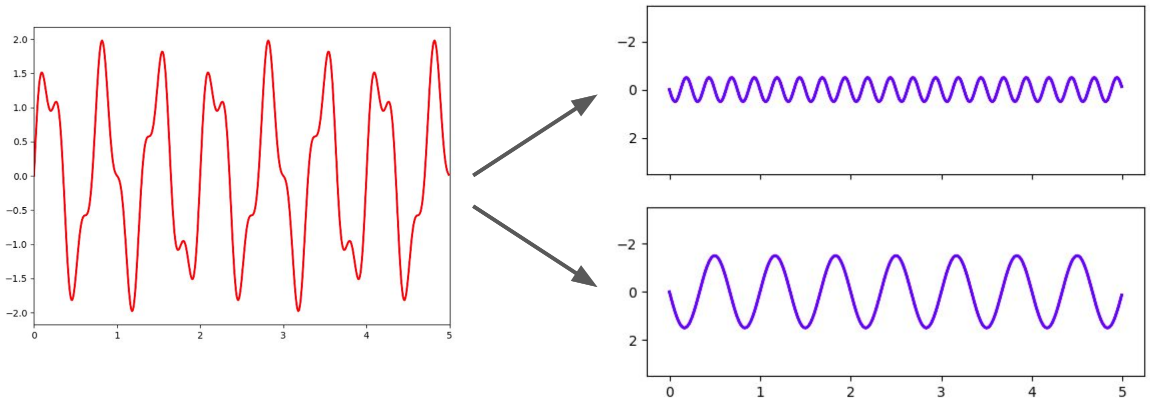
Fourier transform

- Decompose complex periodic sound into sum of sine waves oscillating at different frequencies



Fourier transform

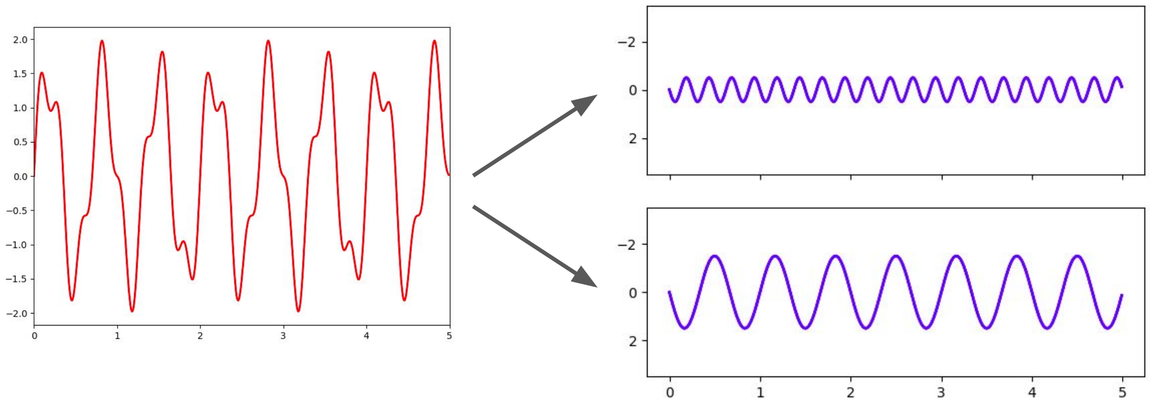
- Decompose complex periodic sound into sum of sine waves oscillating at different frequencies



$$\boxed{s = A_1 \sin(2\pi f_1 t + \varphi_1) + A_2 \sin(2\pi f_2 t + \varphi_2)}$$

Fourier transform

- Decompose complex periodic sound into sum of sine waves oscillating at different frequencies



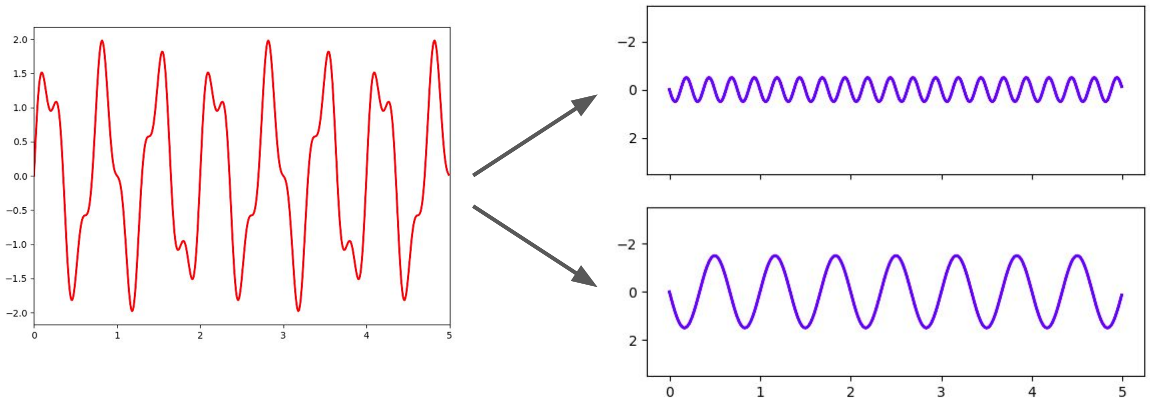
$$\boxed{A_1 = 0.5, f_1 = 4, \varphi_1 = 0}$$

$$\boxed{A_2 = 1.5, f_2 = 1.5, \varphi_2 = 0}$$

$$s = A_1 \sin(2\pi f_1 t + \varphi_1) + A_2 \sin(2\pi f_2 t + \varphi_2)$$

Fourier transform

- Decompose complex periodic sound into sum of sine waves oscillating at different frequencies

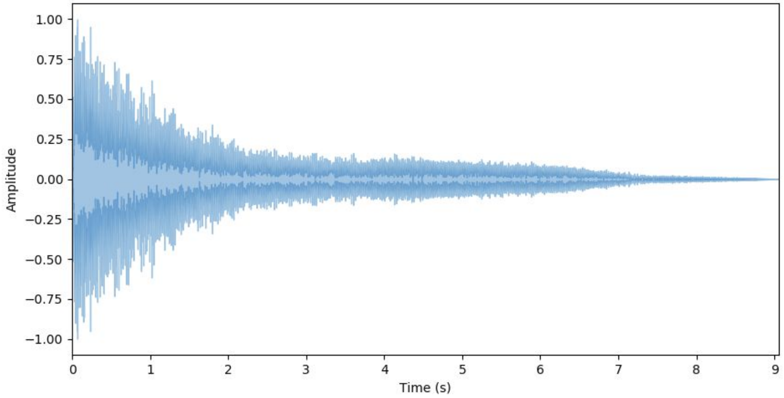


$\boxed{A_1 = 0.5}, f_1 = 4, \varphi_1 = 0$

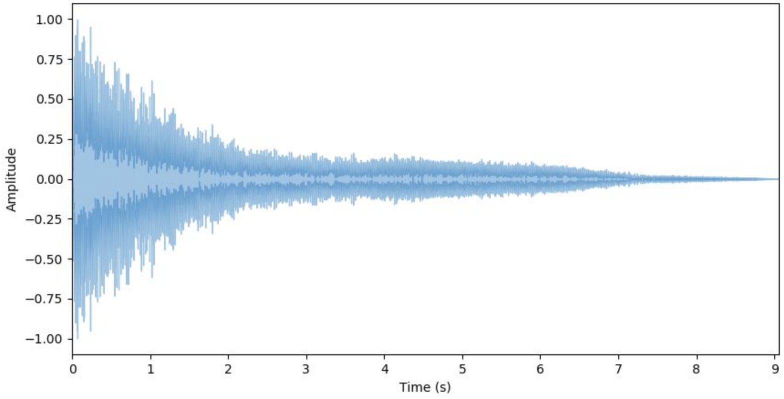
$\boxed{A_2 = 1.5}, f_2 = 1.5, \varphi_2 = 0$

$$s = A_1 \sin(2\pi f_1 t + \varphi_1) + A_2 \sin(2\pi f_2 t + \varphi_2)$$

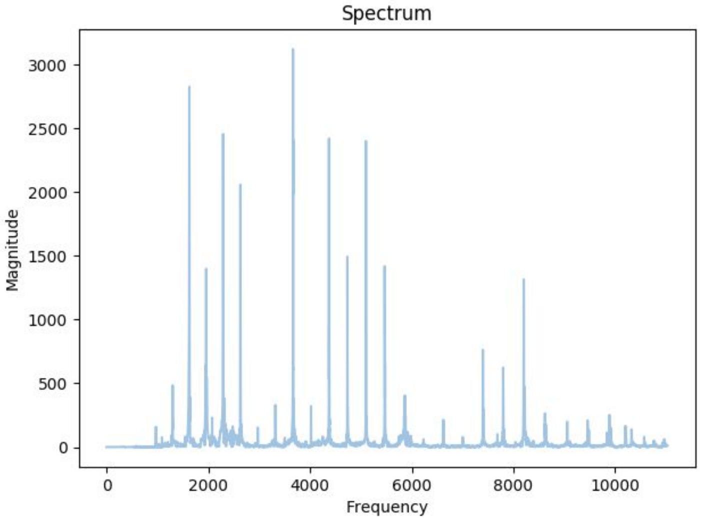
Fourier transform



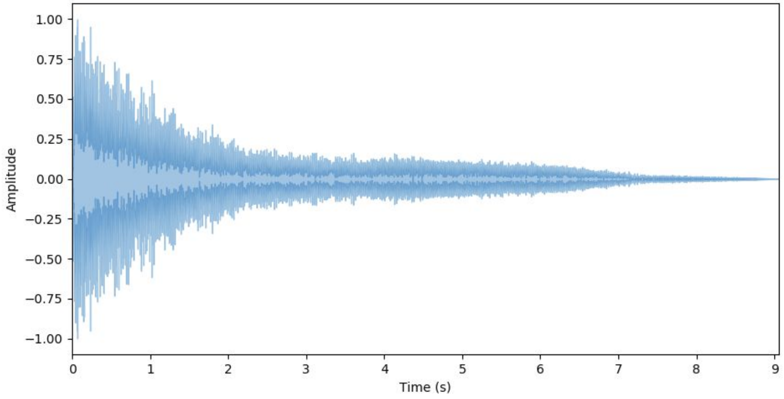
Fourier transform



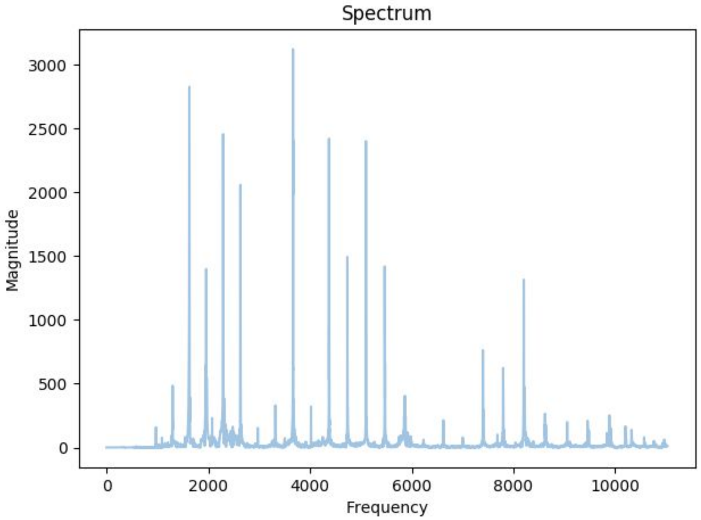
FFT



Fourier transform



FFT

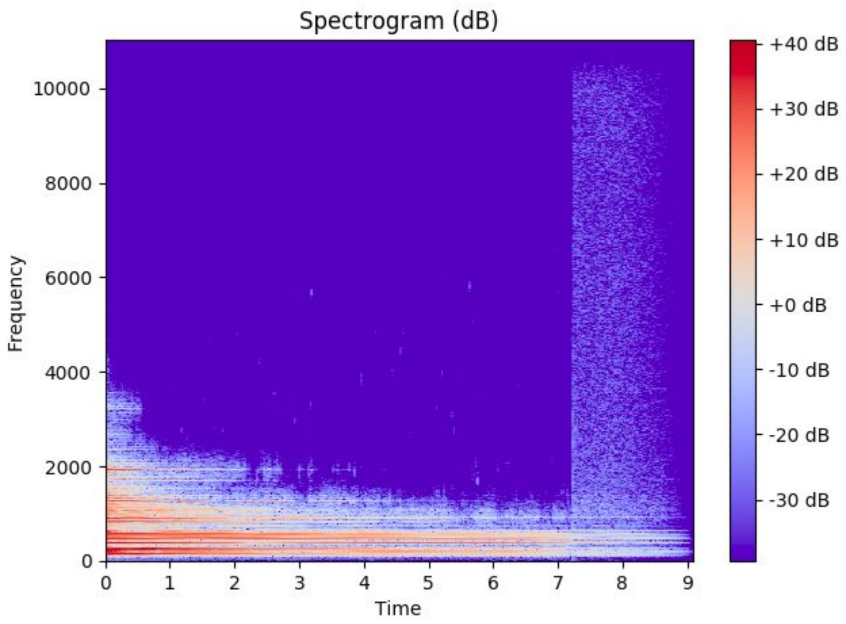


- From *time domain* to *frequency domain*
- No time information

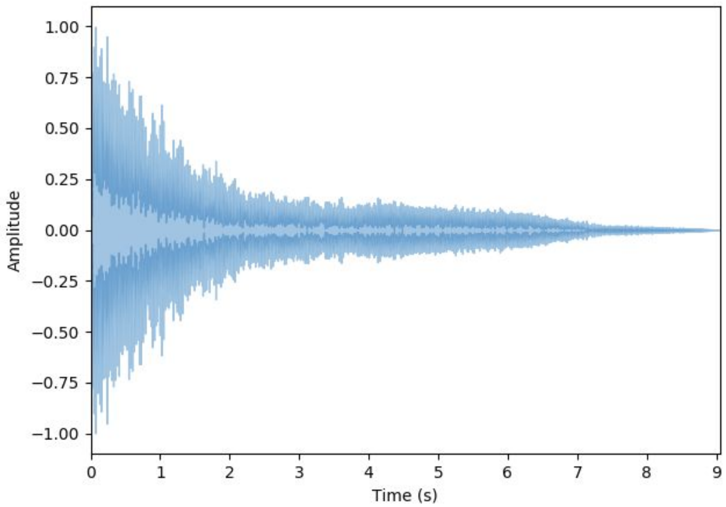
Short Time Fourier Transform (STFT)

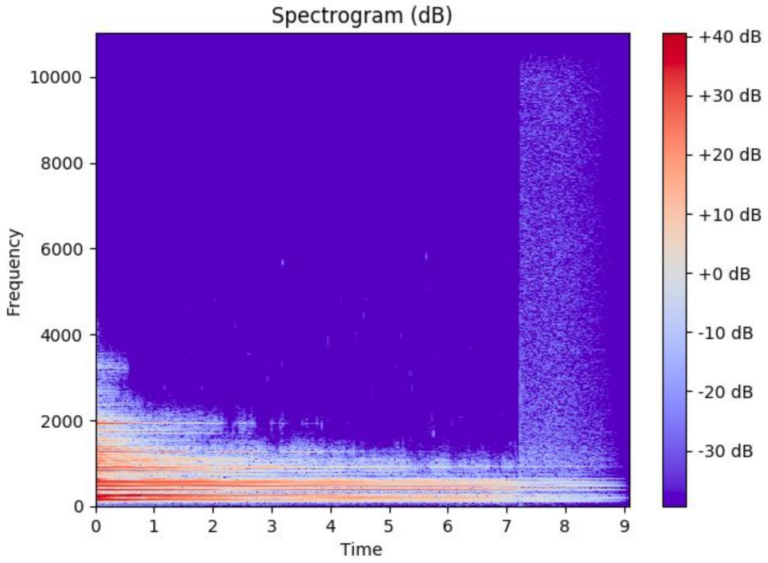
- Computes several FFT at different intervals
- Preserves time information
- Fixed frame size (e.g., 2048 samples)
- Gives a *spectrogram* (time + frequency + magnitude)

Short Time Fourier Transform (STFT)

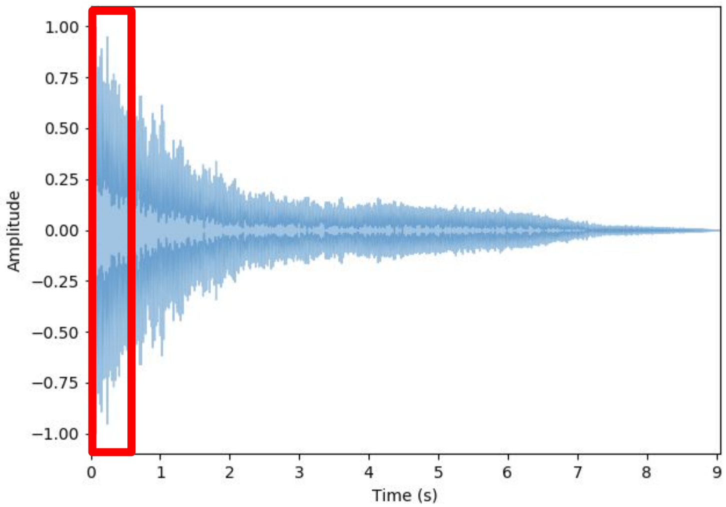


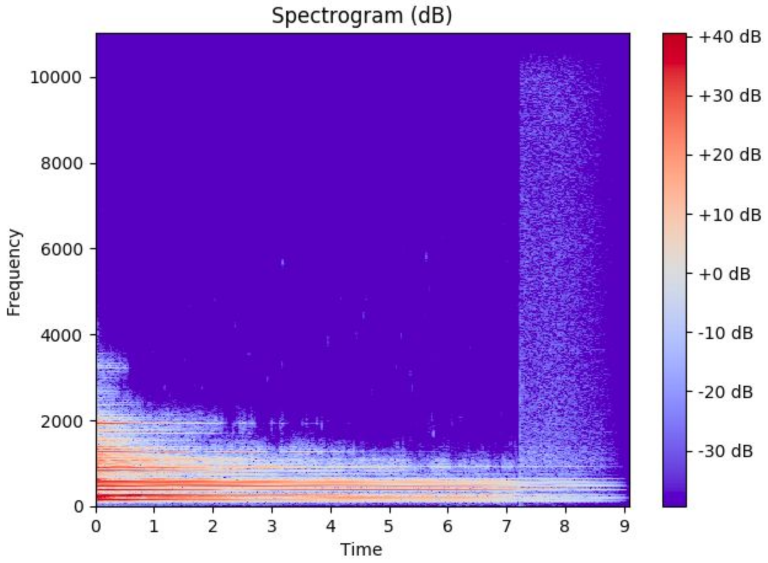
Short Time Fourier Transform (STFT)



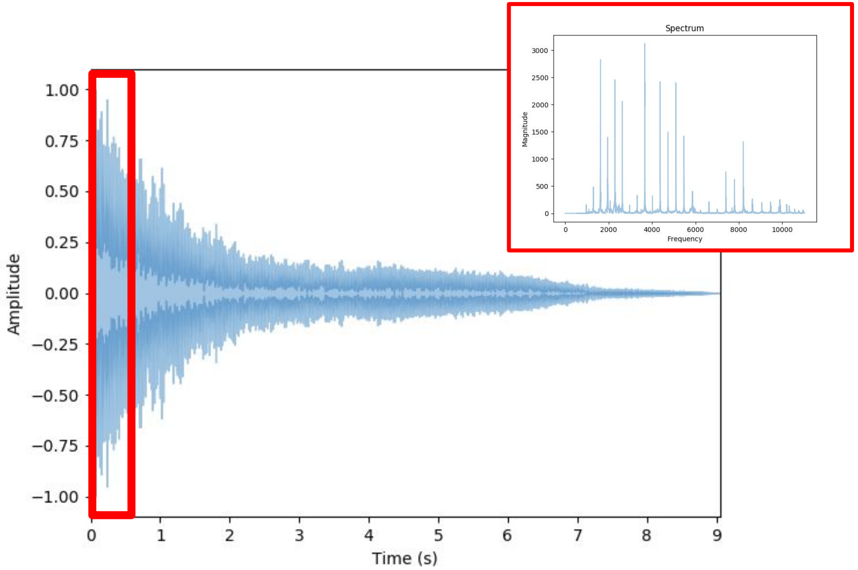


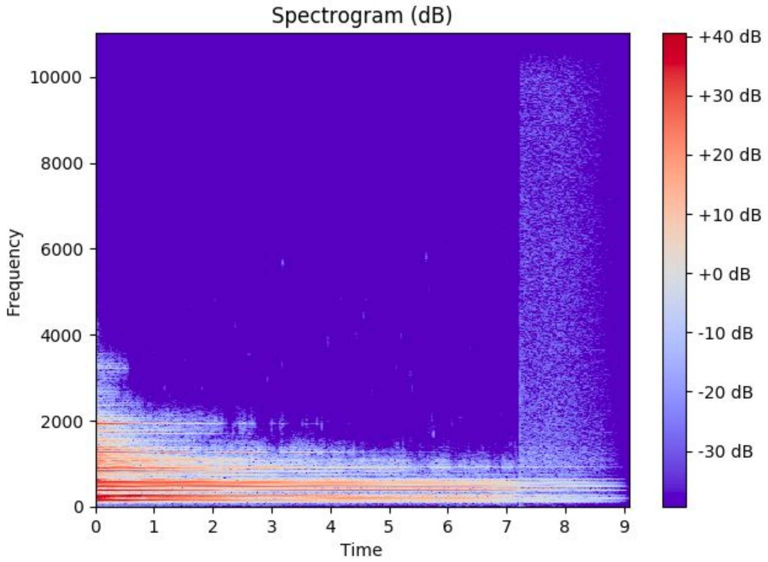
Short Time Fourier Transform (STFT)



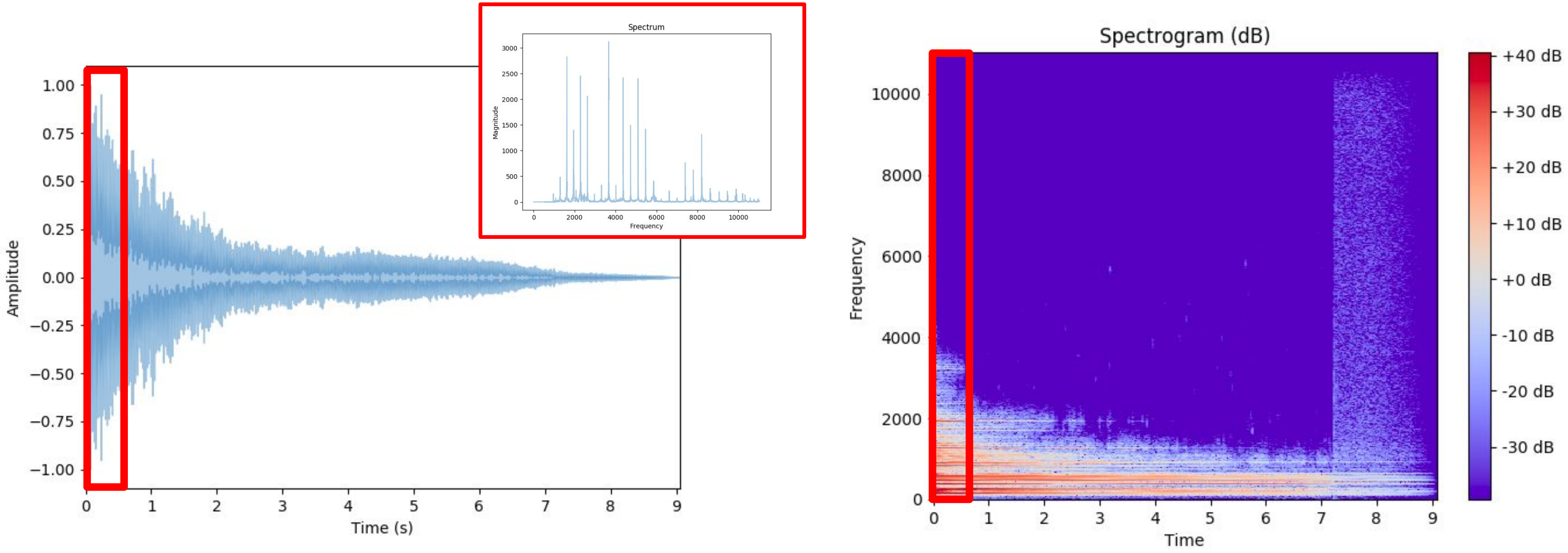


Short Time Fourier Transform (STFT)

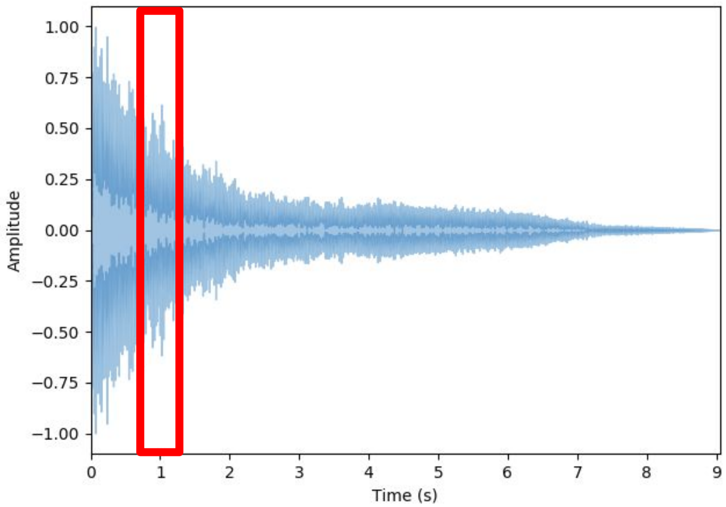


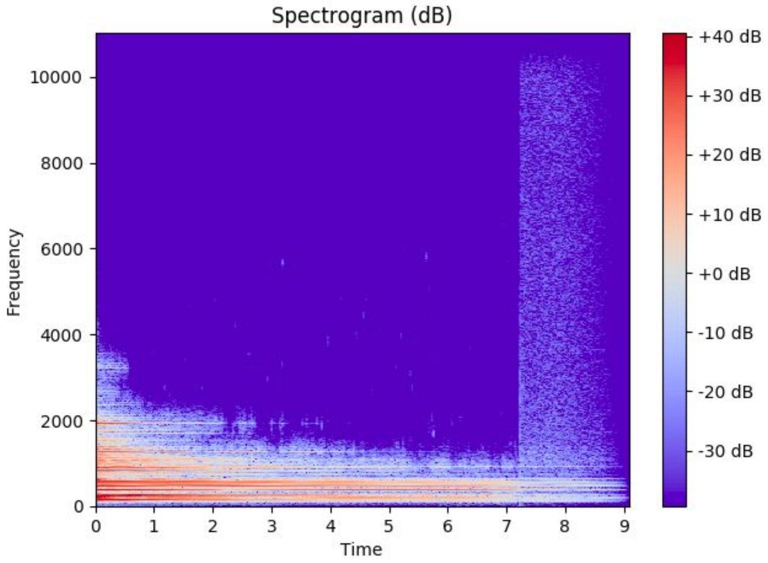


Short Time Fourier Transform (STFT)

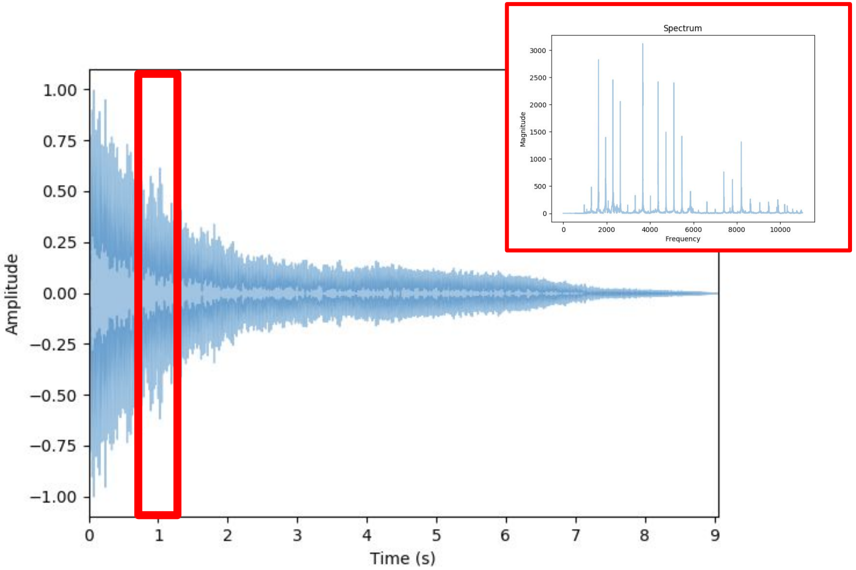


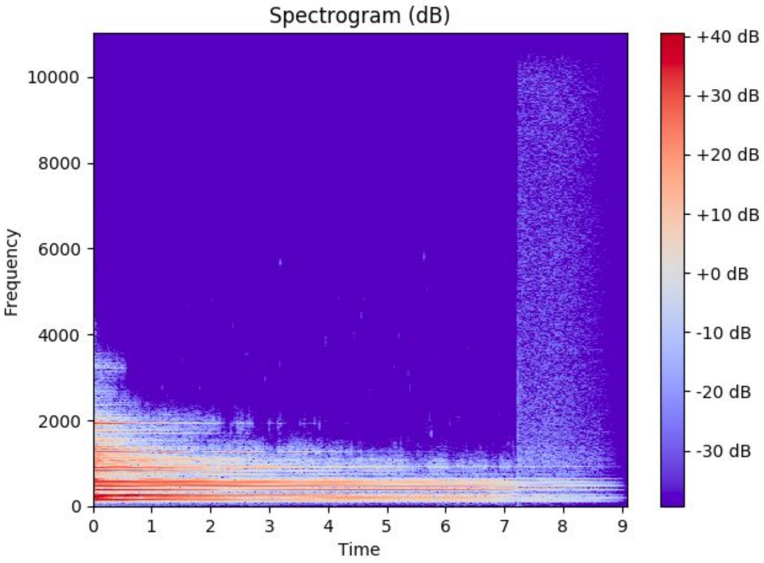
Short Time Fourier Transform (STFT)



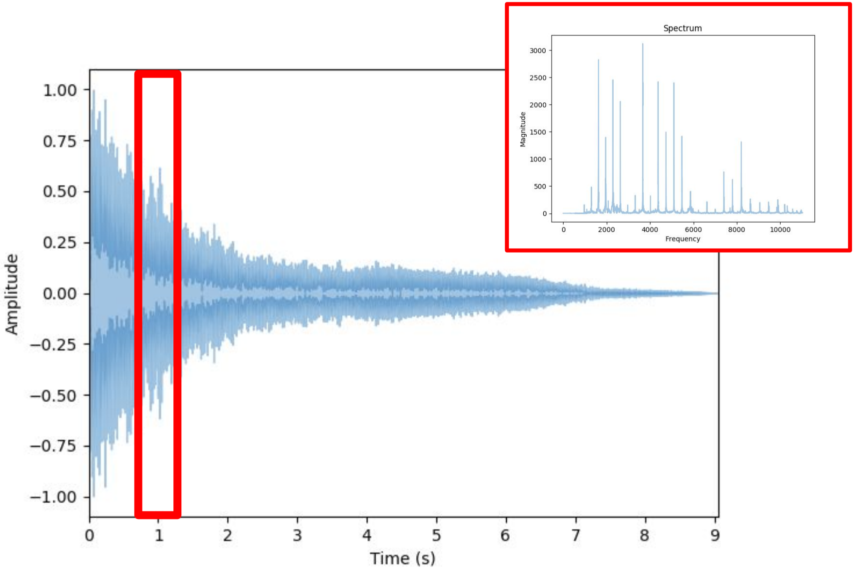


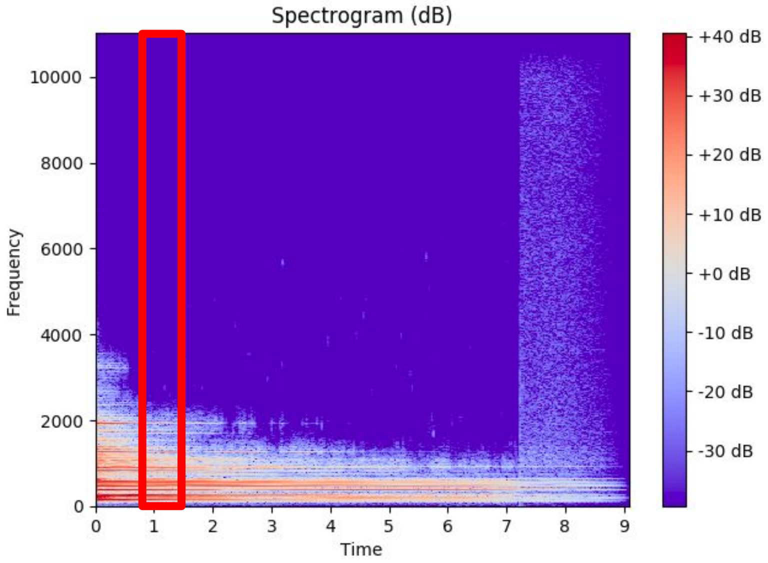
Short Time Fourier Transform (STFT)



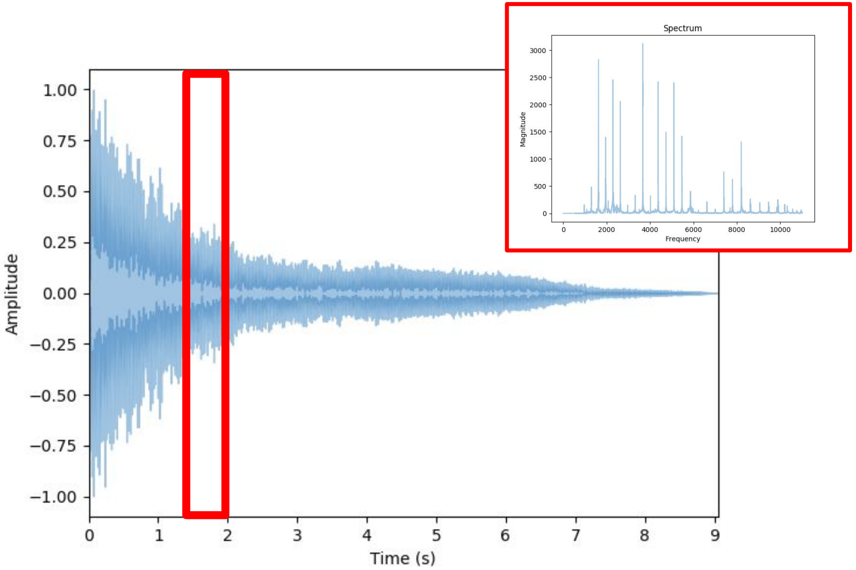


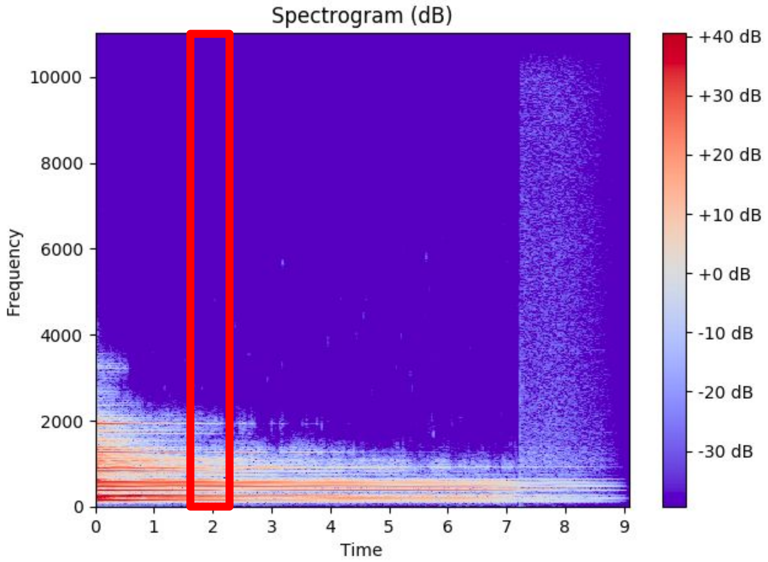
Short Time Fourier Transform (STFT)



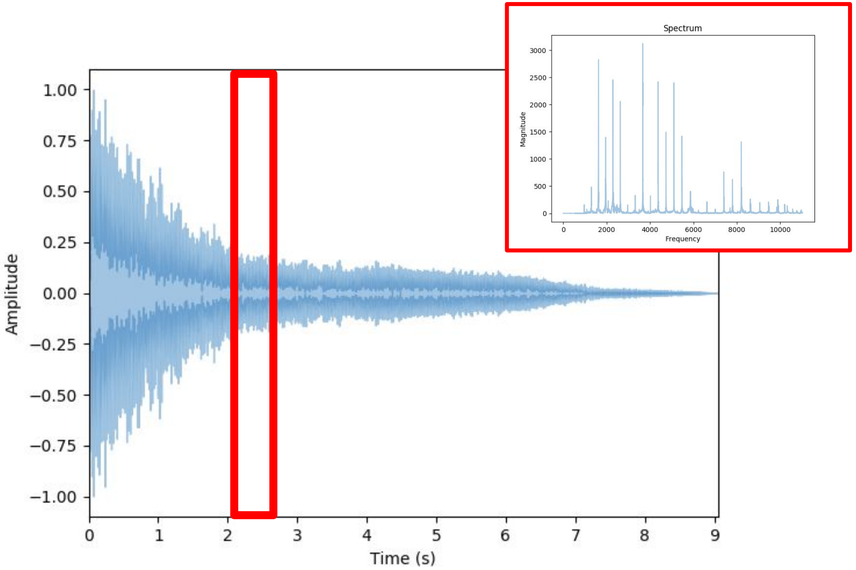


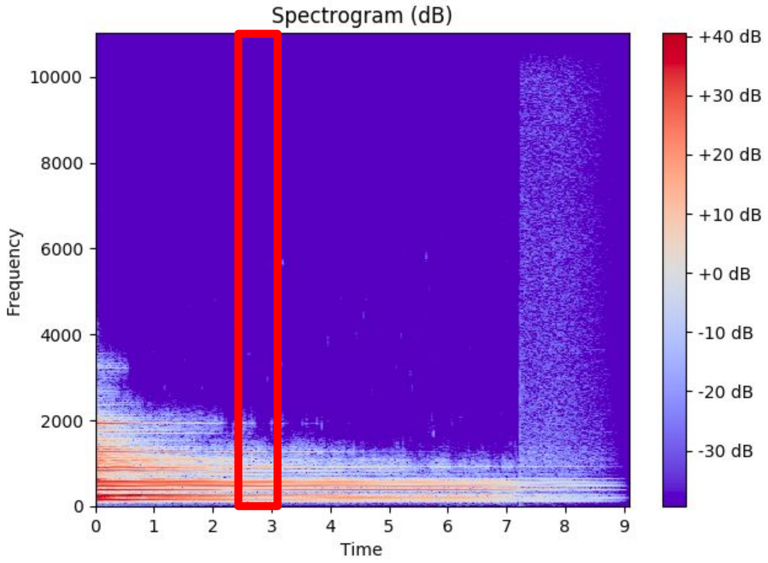
Short Time Fourier Transform (STFT)



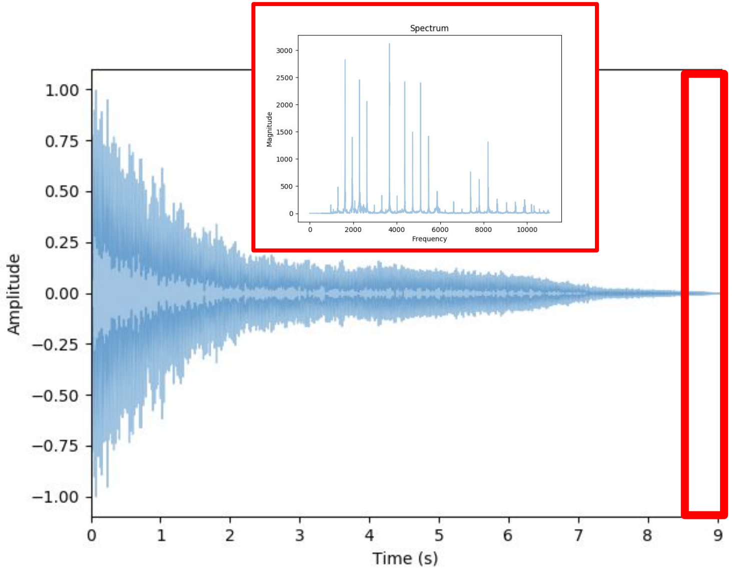


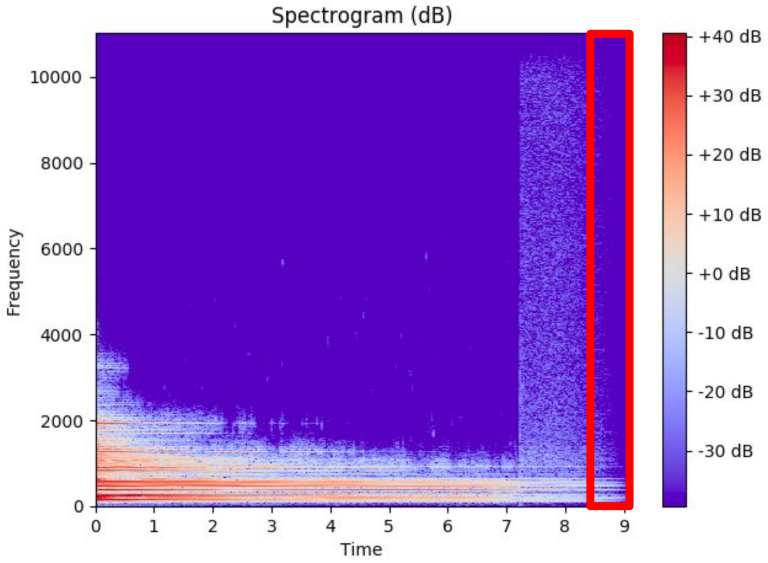
Short Time Fourier Transform (STFT)



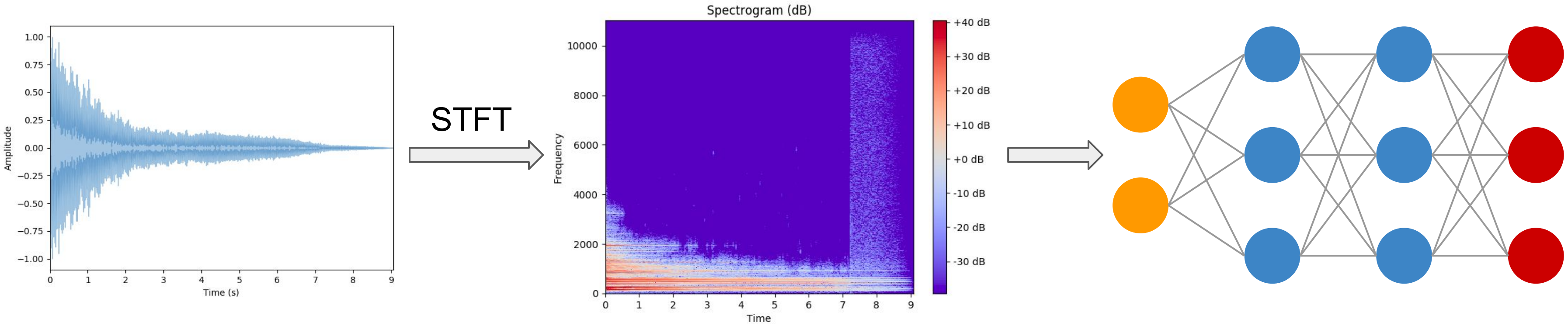


Short Time Fourier Transform (STFT)

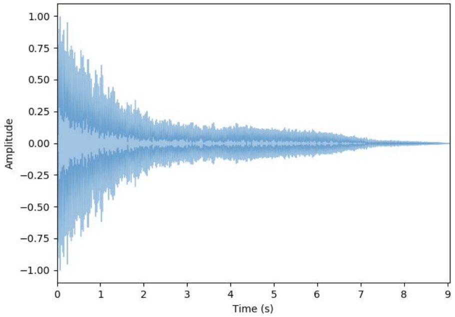




DL pre-proprocessing pipeline for audio data



Traditional ML pre-proprocessing pipeline for audio data



- Amplitude envelope
- Spectral centroid
- 0-crossing rate
- Spectral flux
- ...

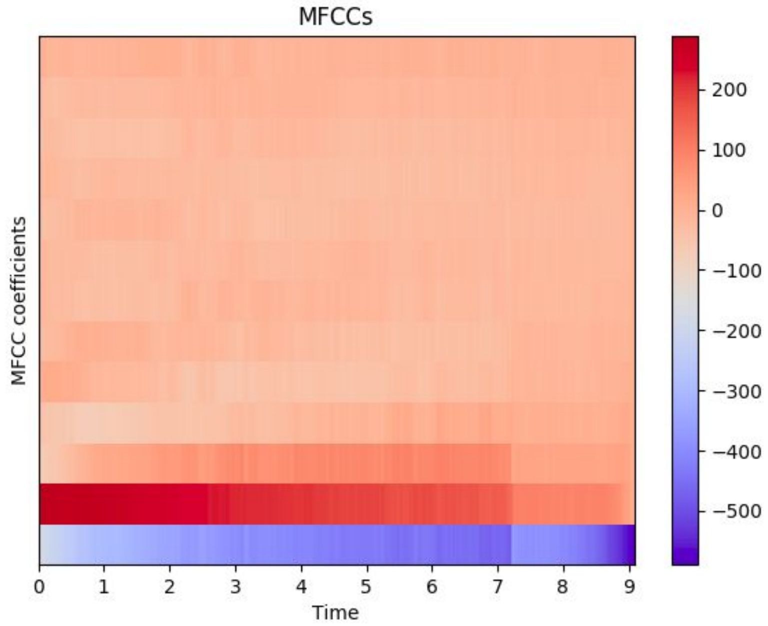
ML algorithm

- Feature engineering
- Perform STFT
- Extract time + frequency domain features

Mel Frequency Cepstral Coefficients (MFCCs)

- Capture timbral/textural aspects of sound
- Frequency domain feature
- Approximate human auditory system
- 13 to 40 coefficients
- Calculated at each frame

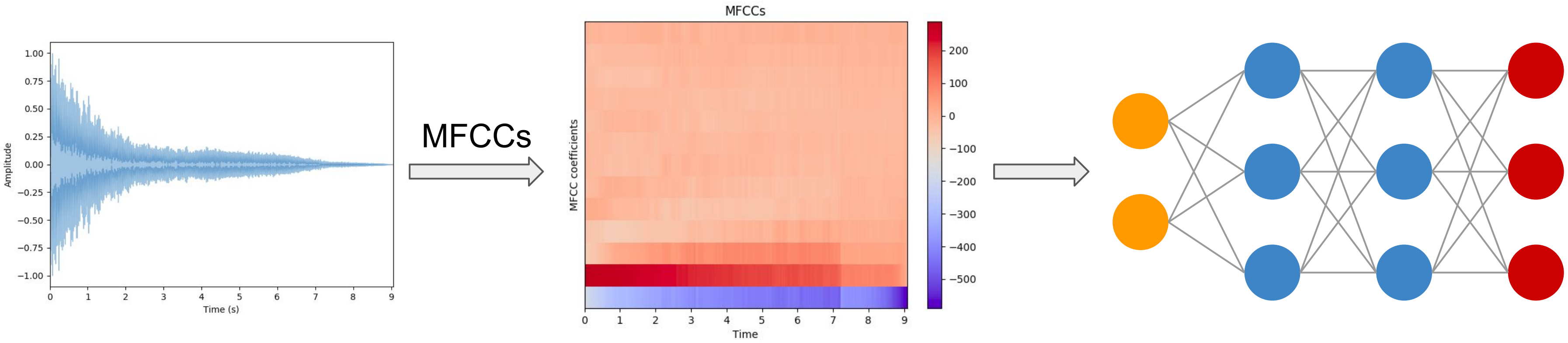
Mel Frequency Cepstral Coefficients (MFCCs)



MFCCs applications

- Speech recognition
- Music genre classification
- Music instrument classification
- ...

DL pre-proprocessing pipeline for audio data



What’s up next?

- Perform FFT and STFT with Python
- Extract MFCCs
- Get familiar with *librosa*