Introduction to audio ML with TorchAudio

Moto Hira, Zhaoheng Ni

PyTorch, Meta.

Agenda

1. Introduction
2. Fundamentals
3. Audio I/O
4. Feature Extraction
5. Examples

1. Introduction

1. TorchAudio Project
2. The PyTorch Audio Team

WHAT IS TORCHAUDIO? — A QUICK LIBRARY WALKTHROUGH

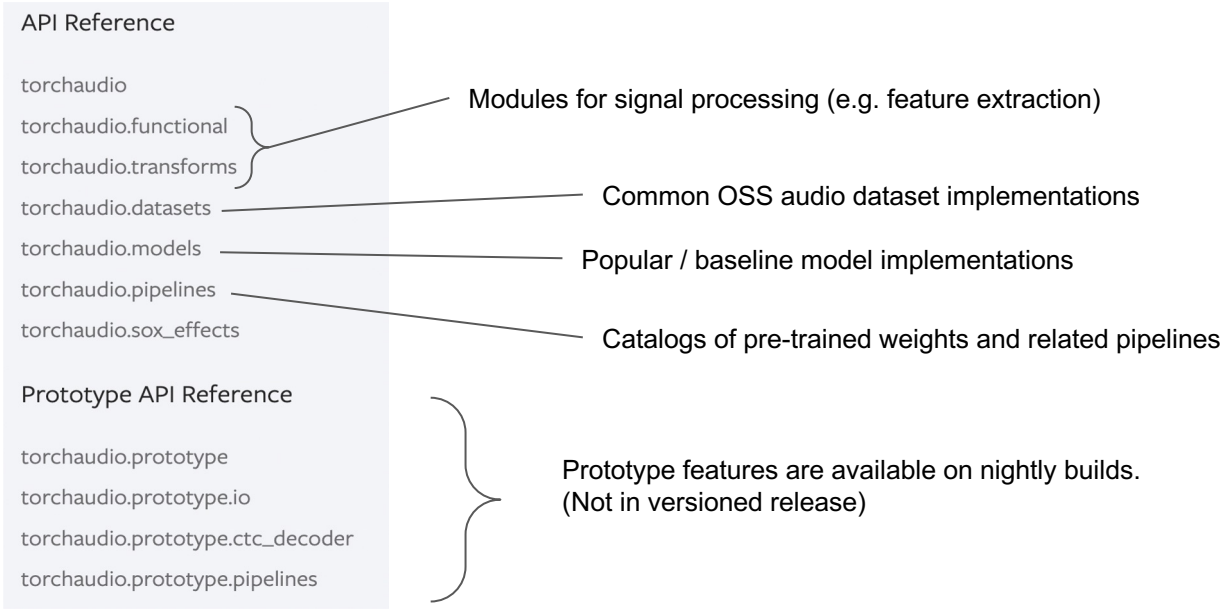
TorchAudio

Source Code: https://github.com/pytorch/audio

Documentation (dev): https://pytorch.org/audio/main

Documentation (stable): https://pytorch.org/audio/stable

WHAT IS TORCHAUDIO? — A QUICK LIBRARY WALKTHROUGH



OUR TEAM

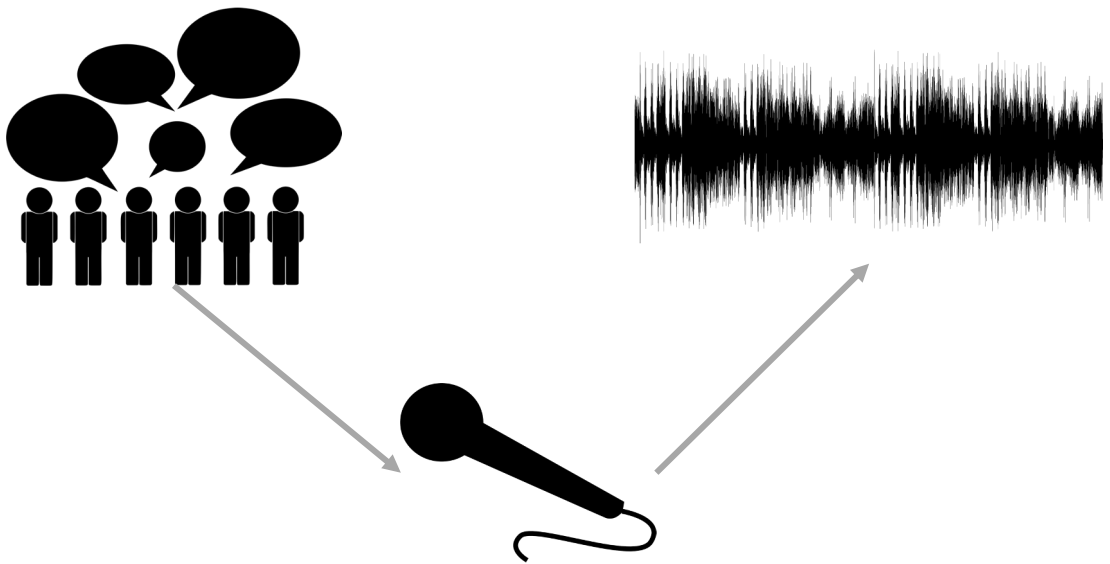
- **Moto Hira**, Software Engineer
- **Jeff Hwang**, Software Engineer
- **Zhaoheng Ni**, Research Scientist
- **Xiaohui Zhang**, Research Scientist
- **Yumeng Tao**, Engineering Manager

2. Fundamentals

1. Waveform
2. Spectrogram

WHAT IS WAVEFORM?

An audio waveform represents pressure vibrations of sound recorded by microphone.

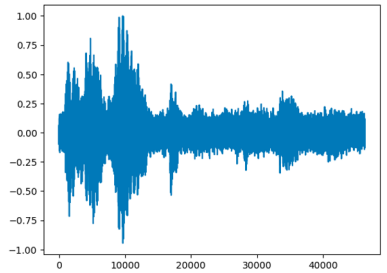


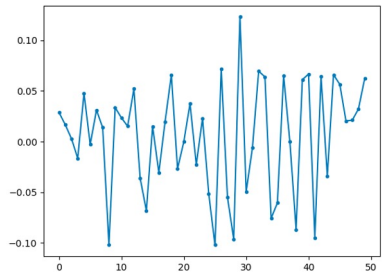
WHAT IS WAVEFORM?

Waveforms are most likely discrete.

16 kHz means sampling one point every 1/16000 second.

Each point represents the energy of sound vibration at the moment.





WHAT IS WAVEFORM?

The sample frequency must be at least twice as the sound frequency, according to Nyquist–Shannon sampling theorem

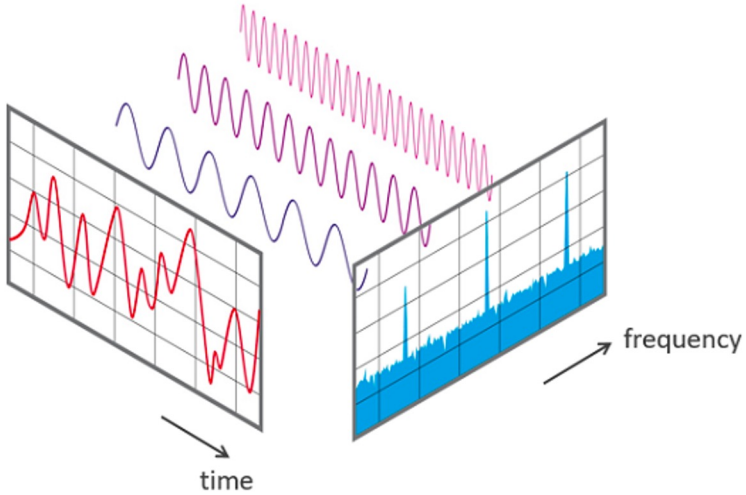
WHAT IS SPECTROGRAM?

Time-domain signal can be expanded to a series of sines.

Each sine can be represented as

$$x(t) = A \cdot \cos(2\pi f t + \varphi)$$

Where A is the energy, f is the frequency, φ is the initial phase

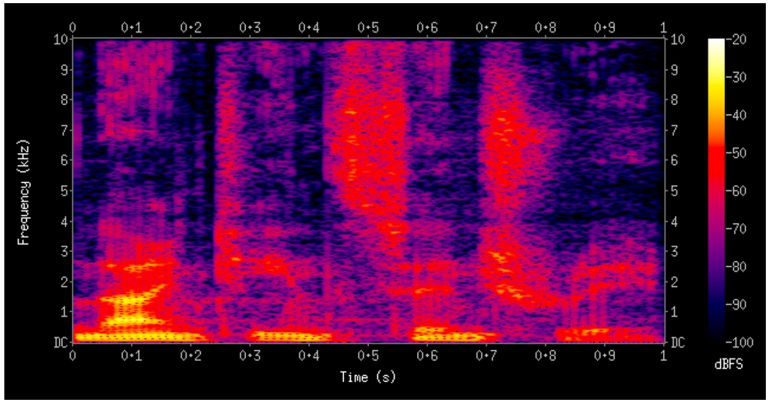


Reference: https://www.nti-audio.com/en/support/know-how/fast-fourier-transform-fft

WHAT IS SPECTROGRAM?

A spectrogram is a visual representation of the spectrum of frequencies of a signal as it varies with time.

Waveforms can be transformed into spectrograms by Short-time Fourier Transform (STFT).



A spectrogram visualizing the results of a STFT of the words "nineteenth century"
Reference: https://en.wikipedia.org/wiki/Short-time_Fourier_transform

TorchAudio

3. Audio I/O

AUDIO I/O

```
import torchaudio

# Load audio data
waveform, sample_rate = torchaudio.load('original.flac')
```

AUDIO I/O

```
# Resample to 8000 Hz

new_sample_rate = 8000
waveform = torchaudio.functional.resample(
    waveform, sample_rate, new_sample_rate)
```

AUDIO I/O

```
# Save the audio
torchaudio.save(
    'resampled.flac', waveform, new_sample_rate)
```

4. Feature Extraction

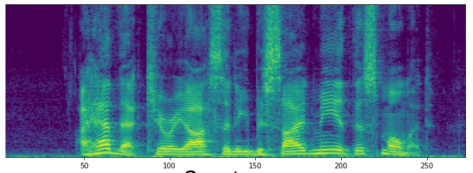
FEATURE EXTRACTION

Feature extraction and augmentation

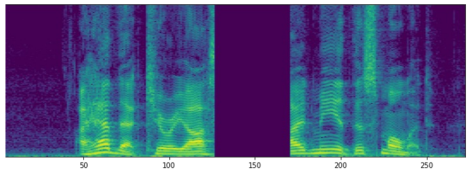
```
import torchaudio.transforms as T

# Get spectrogram
trans = T.Spectrogram(...)
spectrogram = trans(waveform)

# Mask along time axis a.k.a SpecAugment
time_masking = T.TimeMasking(...)
time_masked = time_masking(spectrogram)
```



Spectrogram



Masked along time axis

5. Examples

Examples

- Streaming ASR
  https://pytorch.org/audio/stable/tutorials/online_asr_tutorial.html
- Text-to-Speech
  https://pytorch.org/audio/stable/tutorials/tacotron2_pipeline_tutorial.html
- Speech Enhancement
  https://pytorch.org/audio/stable/tutorials/mvdr_tutorial.html
- Music Separation
  https://pytorch.org/audio/stable/tutorials/hybrid_demucs_tutorial.html

THANK YOU