Homework 3

Deep Learning
EE 298/CoE 197/EE 197/ECE 197
University of the Philippines Diliman
2022

Dataset: SpeechCommands

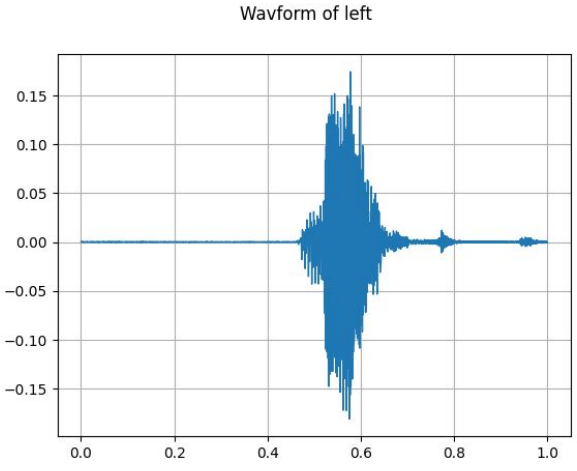
- **SPEECHCOMMANDS()** dataset returns the ff format:
  - (waveform, sample_rate, label, speaker_id, utterance_number)

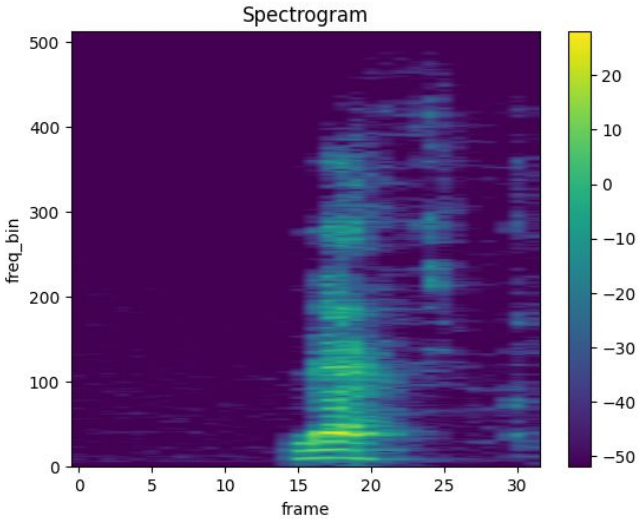
```
>>> val = torchaudio.datasets.SPEECHCOMMANDS('/home/izza/Work/Grad_Work/kws/data', download=False, subset='validation')
>>> val[0]
(tensor([[-0.0004, -0.0007, -0.0009,  ...,  0.0062,  0.0058,  0.0057]]), 16000, 'right', 'a69b9b3e', 0)
```

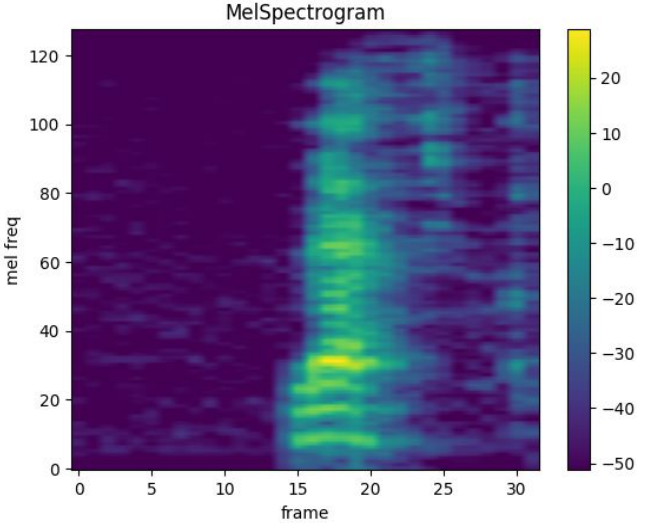
    - waveform - audio waveform (torch.float32)
    - sample_rate - sampling rate (fs)
    - label - label of the audio (string)
- **MelSpectrogram()** transform returns the mel spectrogram of an audio
  - Output is of the ff format: (channel, num_mels, time)
    - channel - dependent on num_channel of waveform
    - num_mels - number of mel features
    - time - number of frames

What are Mel Spectrograms?

- Before we go to mel, we need to understand that a **waveform** consists of varying sound waves with different frequencies and amplitudes.
  - There's a tendency for these frequencies to vary over time
  - A **spectrogram** will tell us how these frequencies vary over time, and their corresponding signal strength (this is done through the use of **Short Time Fourier Transform**)



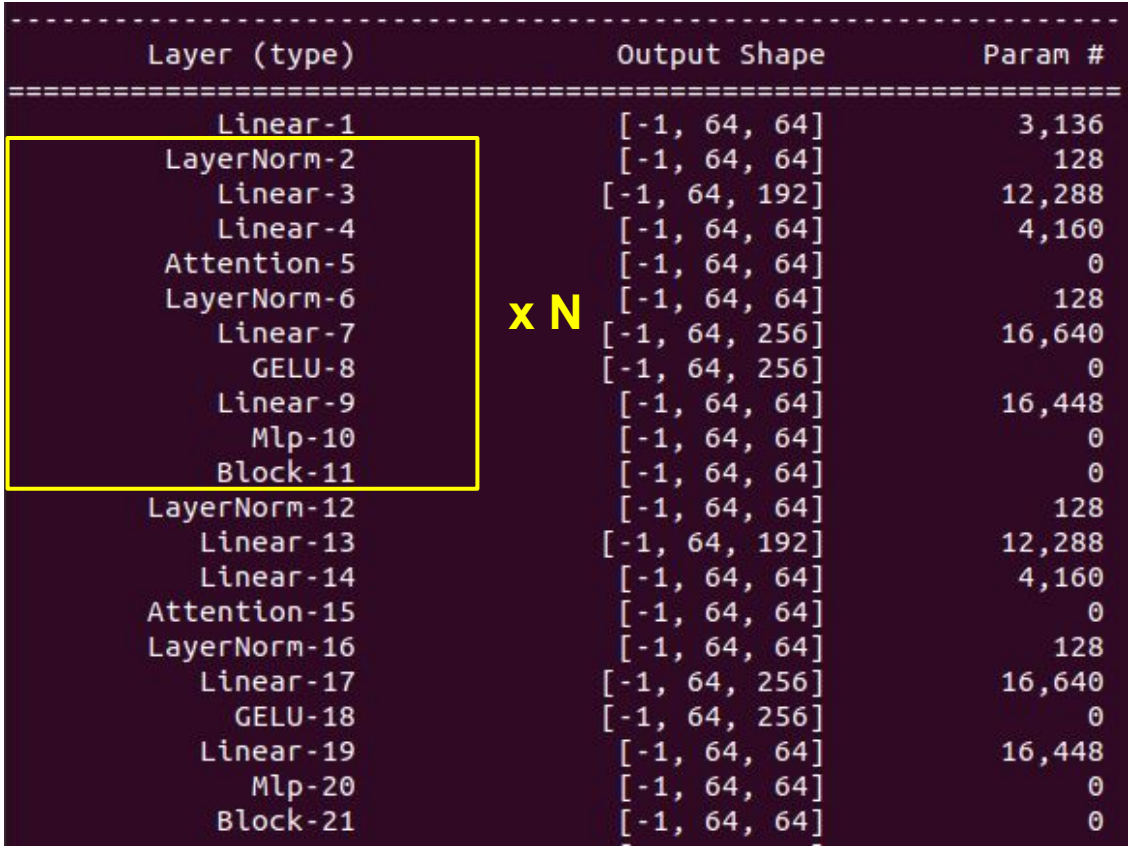




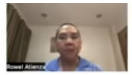
Model: Transformer

- Linear layer
- Repeating block *N* times
  - Each block:
    - LayerNorm
    - Attention
    - LayerNorm
    - MLP
  - *N* determined by **depth**
- Linear layer

Note: Model here uses Cifar10 data.

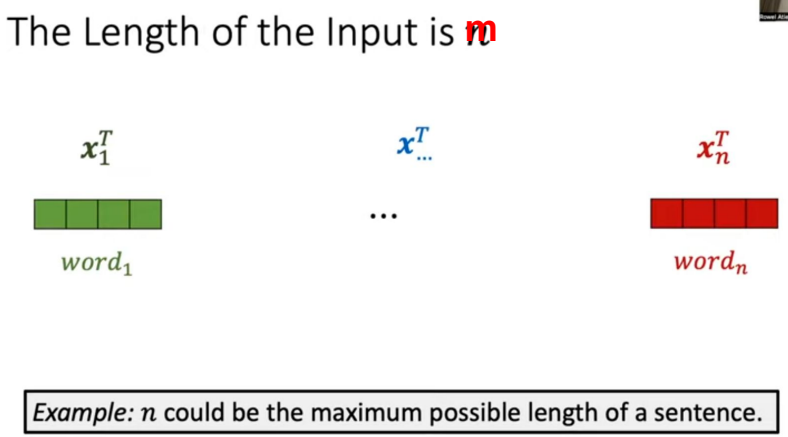


Model: Transformer



- **Question**: How do transformers process data?
  - Remember sir's lecture!
  - Transformers can process sequences of length *m* (seqlen), where each word (data) consists of *n-dim* vectors (features)
  - **Embedding Layer** is in charge of converting raw data into *n-dim* vectors
    - Images first tokenized to *p_w x p_h* patches
    - nn.Linear in previous example
  - **Q:** seqlen:??::word:??

Note: Model here uses Cifar10 data.



Layer (type)	Output Shape	Param #
Linear-1	[-1, 64, 64]	3,136