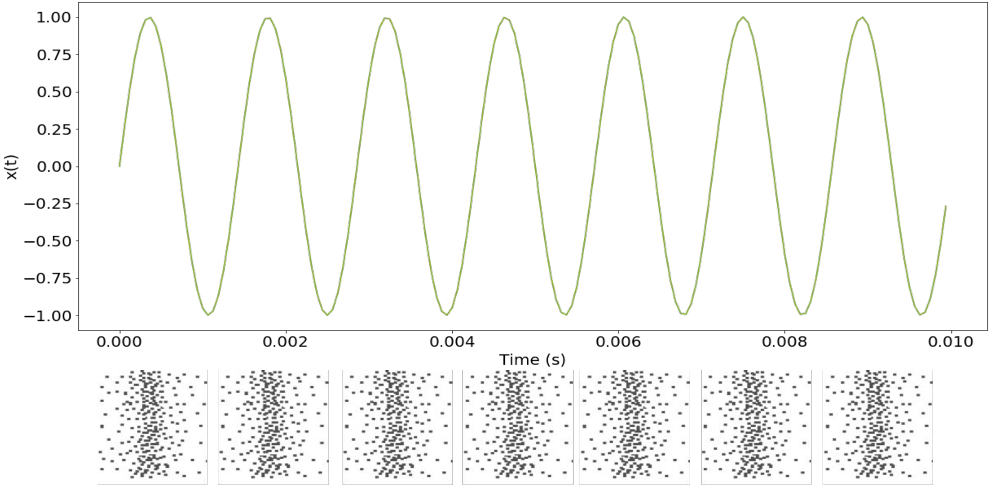
Introduction to DSP

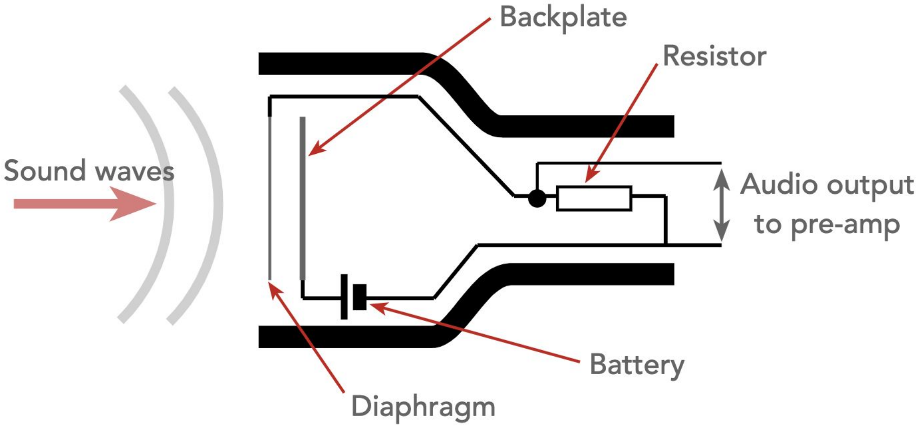
Елена Кантонистова

По мотивам материалов Александра Марковича

What is sound?

- **Sound wave** is the pattern of **oscillations** caused by the movement of energy traveling through the air
- **Microphone** picks up these air **oscillations** and converts them into electrical vibrations
- These **oscillations** are converted into an **analog** signal and then a **digital** signal

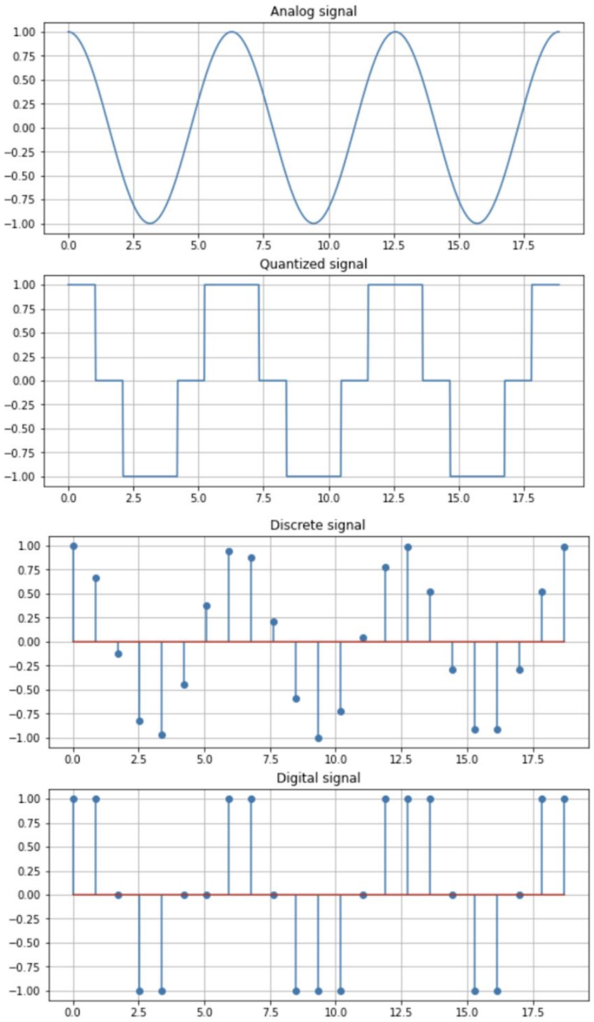




https://pudding.cool/2018/02/waveforms/
https://blog.accusonus.com/science-of-sound/how-microphones-work

How is sound stored in the computer?

- The **analog** signal is discretized, quantized and encoded
- An analog signal is **discretized** in that the signal is represented as a sequence of values taken at discrete points in time **t** with step **d**
- **Quantisation** of a signal consists in splitting the range of signal values into **N** levels in increments of **d** and selecting for each reference the level that corresponds to it
- Signal **encoding** is just a way of presenting the signal in a more compact form

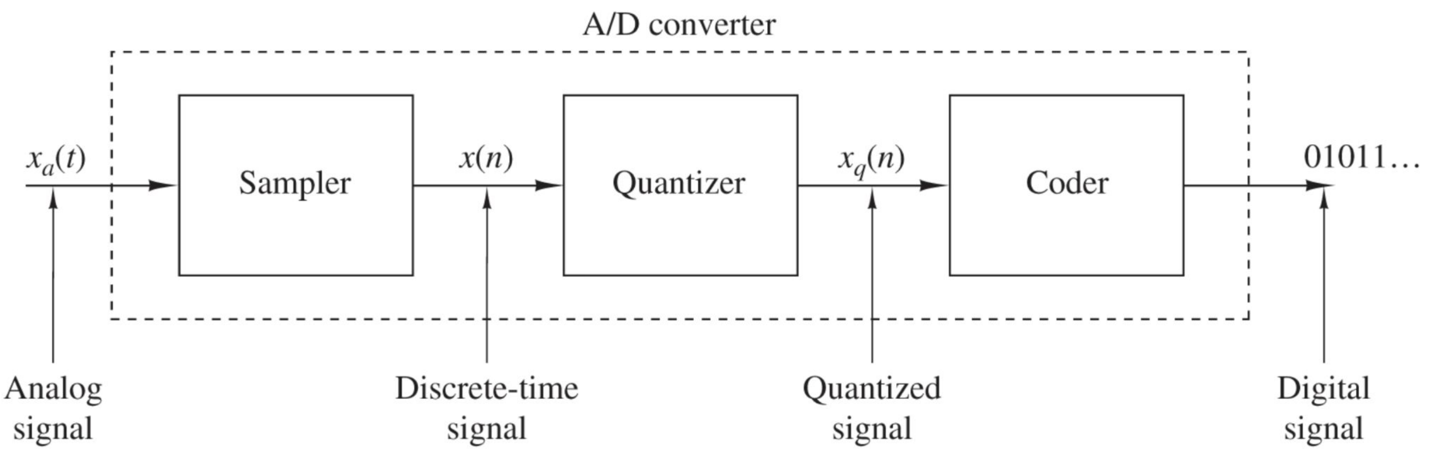


https://github.com/markovka17/dla/blob/master/week01/dsp.ipynb
https://web.sonoma.edu/esee/courses/ee442/archives/sp2019/lectures/lecture09_pcm.pdf

Analog-to-Digital Conversion

- Converting analog signals to a sequence of numbers having finite precision

- Corresponding devices are called A/D converters (ADCs)



https://is.muni.cz/el/1433/jaro2012/PA190/um/Slides_02.pdf

What other characteristics are there?

- **Sample rate (SR)** - number of audio samples per one second (e.g. 8 kHz, 22.05 kHz, 44.1 kHz)
- **Sample size** - number of bits per one sample (e.g. 8, 16, 25, 32 bits)
- **Number of channels** -- how many signals we record in parallel (e.g. mono(1), stereo(2))

8000 Hz

The international G.711 standard for audio used in telephony uses a sample rate of 8000 Hz (8 kHz). This is enough for human speech to be comprehensible.

44100 Hz

The 44.1 kHz sample rate is used for compact disc (CD) audio. CDs provide uncompressed 16-bit stereo sound at 44.1 kHz. Computer audio also frequently uses this frequency by default.

48000 Hz

The audio on DVD is recorded at 48 kHz. This is also often used for computer audio.

96000 Hz

High-resolution audio.

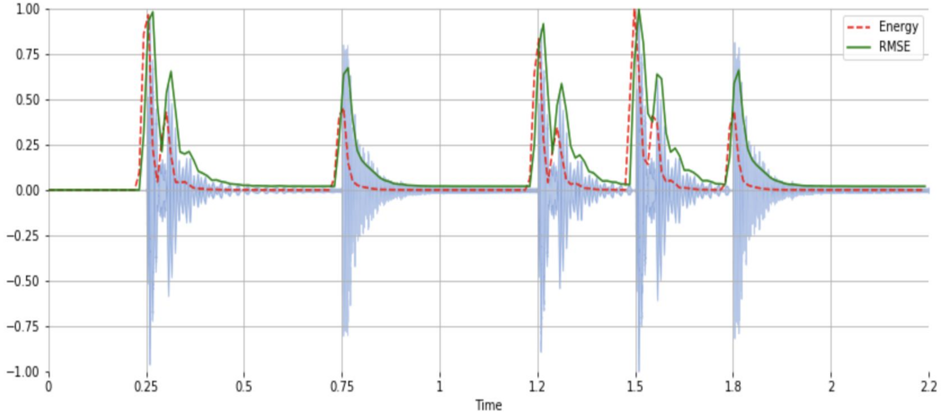
192000 Hz

Ultra-high resolution audio. Not commonly used yet, but this will change over time.

https://developer.mozilla.org/en-US/docs/Web/Media/Formats/Audio_concepts

What other characteristics are there?

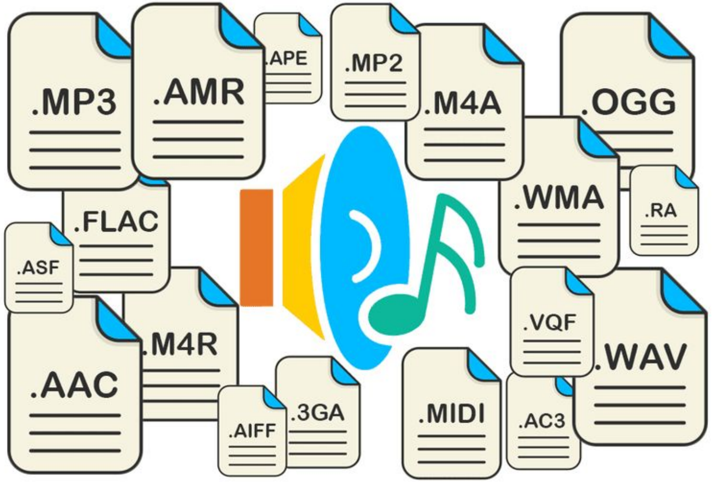
- Assume **f(n)** is our signal where **n** is time
- Power of signal is $f^2(n)$
- Energy of signal is $\sum f^2(n)$
- In practice estimated by some **window**
- Energy in **decibels:** $10 \log_{10} E$
- $\mathrm{SNR}_{dB} = 10 \log_{10} \frac{E_{\text{signal}}}{E_{\text{noise}}}$



https://musicinformationretrieval.com/energy.html

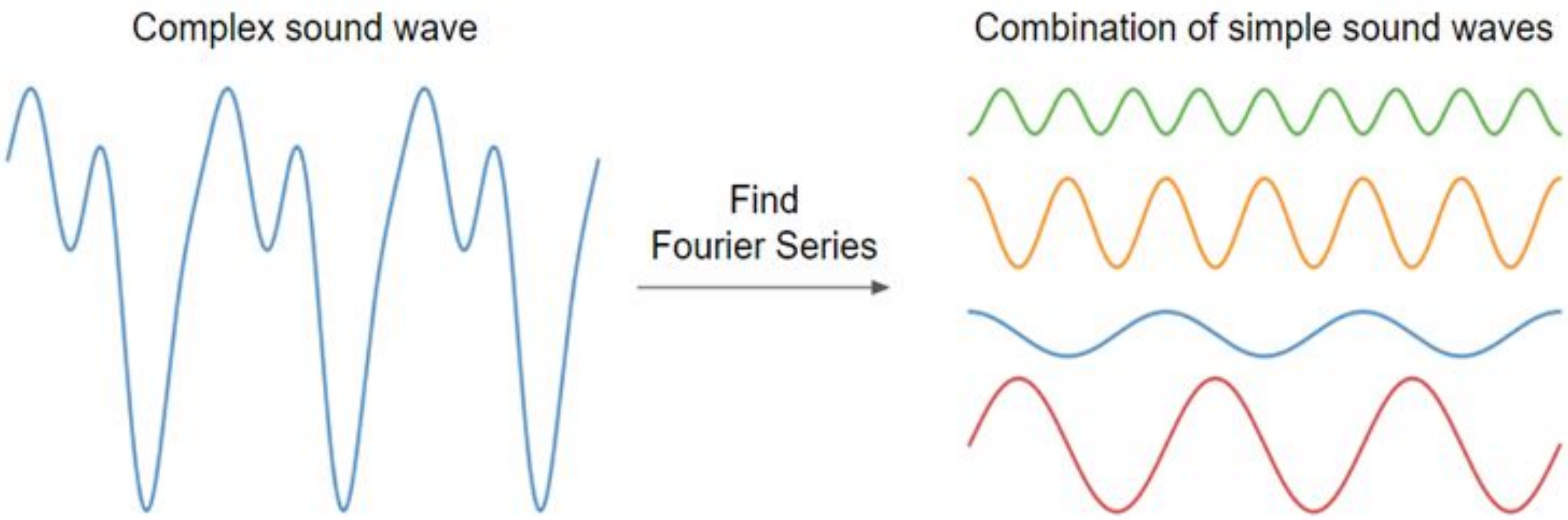
What about audio formats?

- Non-compressed formats: **WAV, AIFF, etc.**
- Lossless compression(2:1) : **FLAC, ALAC, etc.**
- Lossy compression(10:1) : **MP3, Opus, etc**
- **Bit rate** measure a degree of compression. Number of bit that are conveyed or processed per **unit of time.**

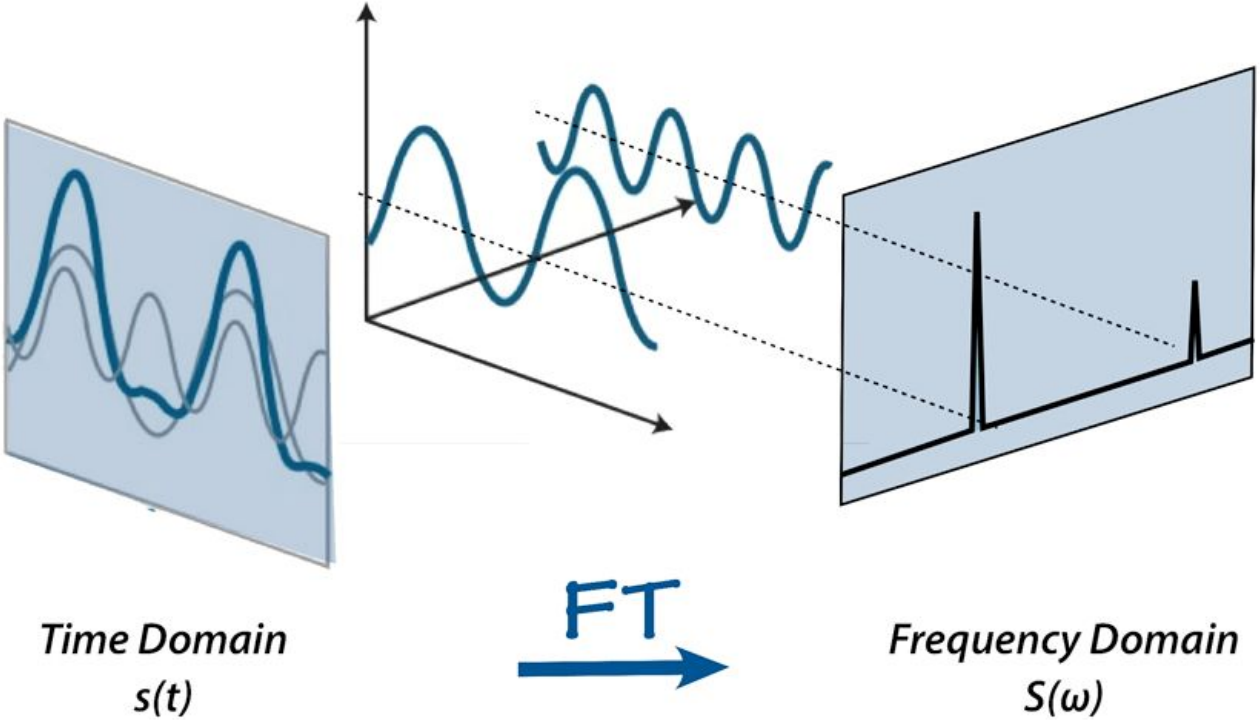


https://en.wikipedia.org/wiki/Audio_file_format

Fourier Transform



Fourier Transform



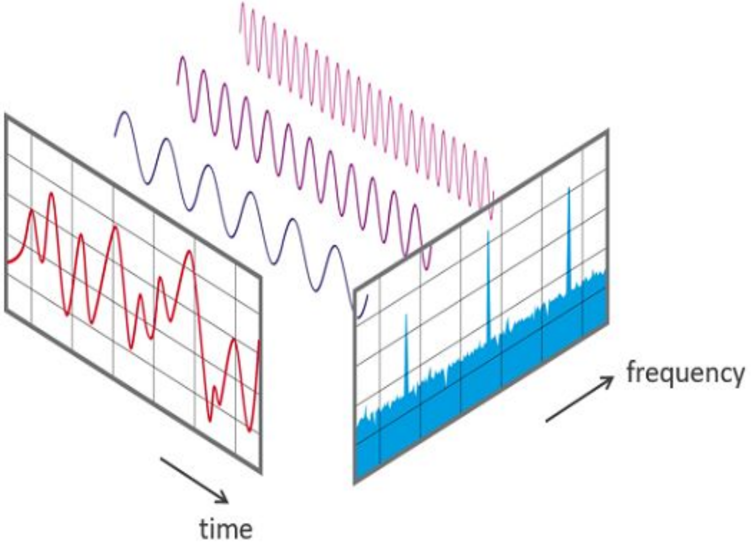
Fourier Transform

- The **Fourier transform(FT)** is a mathematical formula that allows us to decompose a signal into its individual **frequencies** and the frequency's **amplitude**

- FT transfer a signal from the **time domain** to the **frequency domain**

- $F(y) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x y} dx$

$\text{time} \rightarrow \text{frequency}$

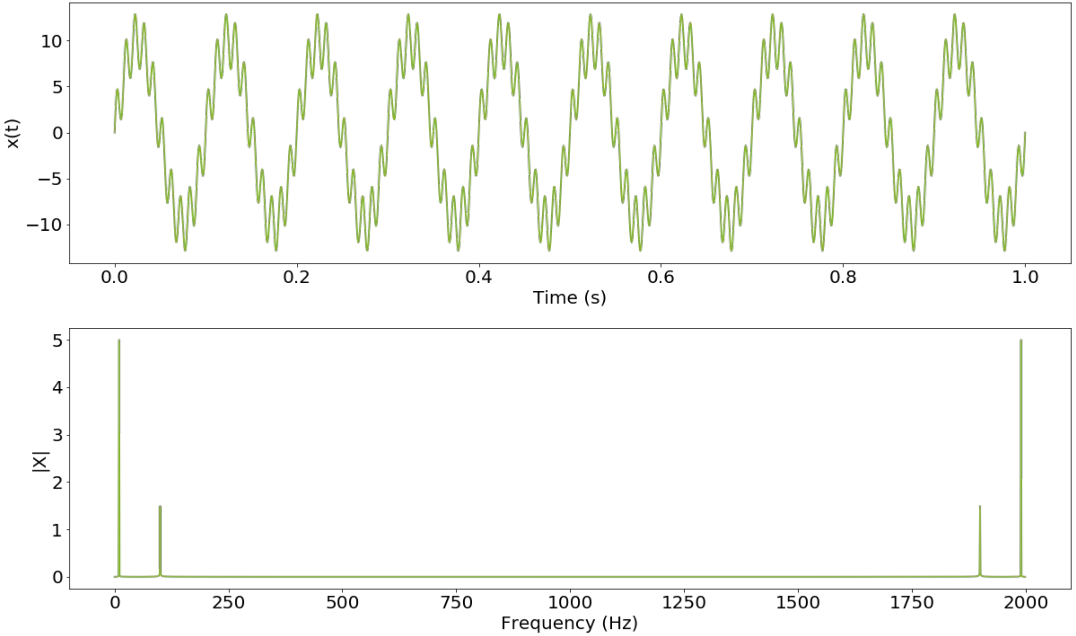


https://youtu.be/spUNpyF58BY

Example of DFT

$$F = 2kHz$$

$$f(t) = 10\sin(2\pi 10t) + 3\sin(2\pi 100t)$$

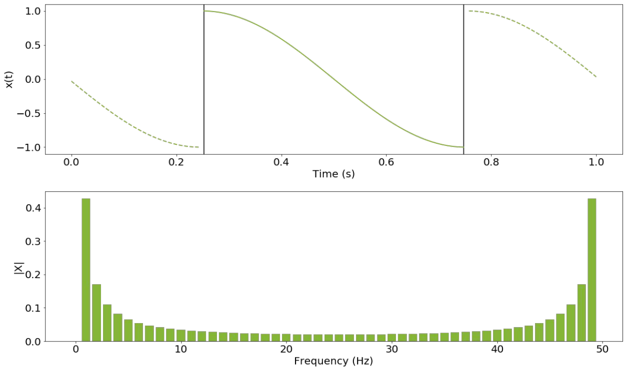


https://www.gaussianwaves.com/2015/11/interpreting-fft-results-obtaining-magnitude-and-phase-information/

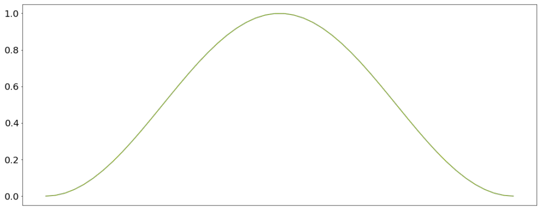
Short-time Fourier transform

FFT + Windowing

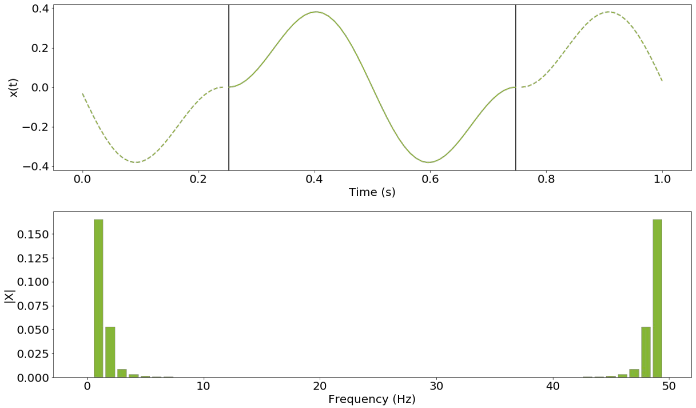
Sliced signal



Window

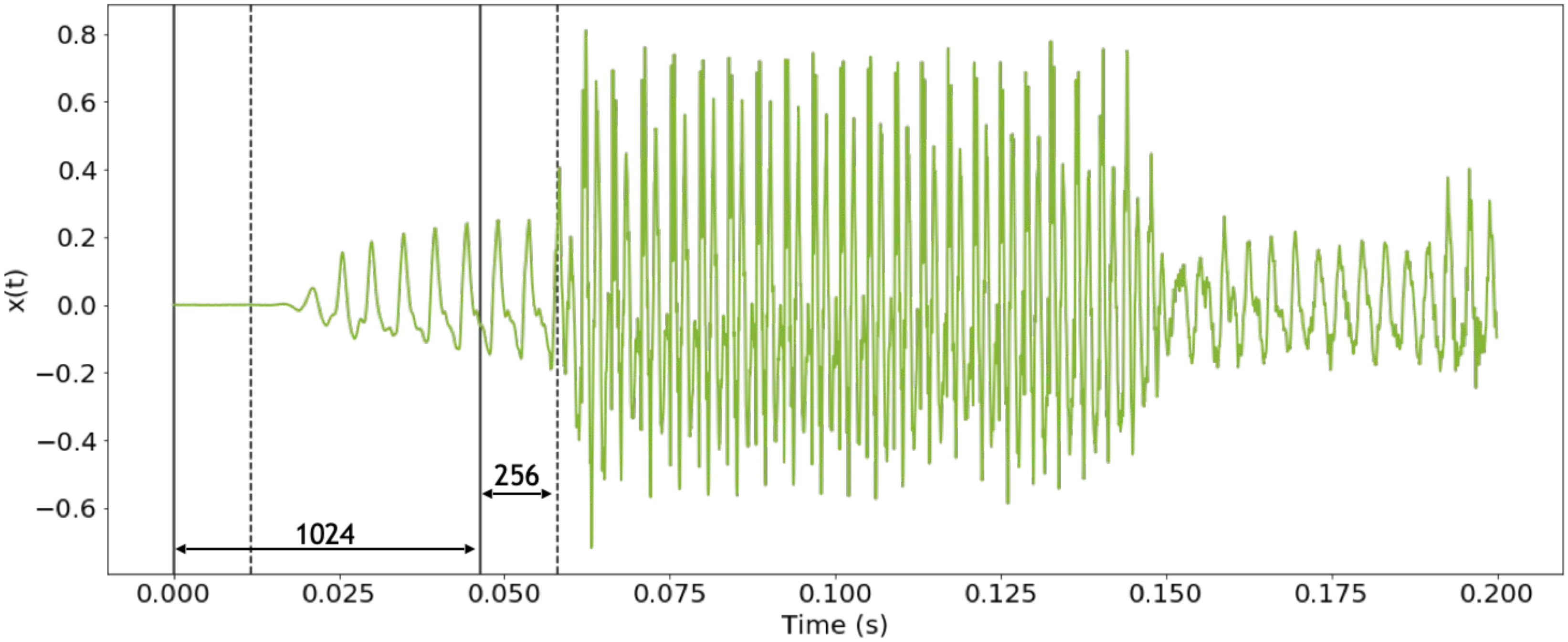


Windowed signal

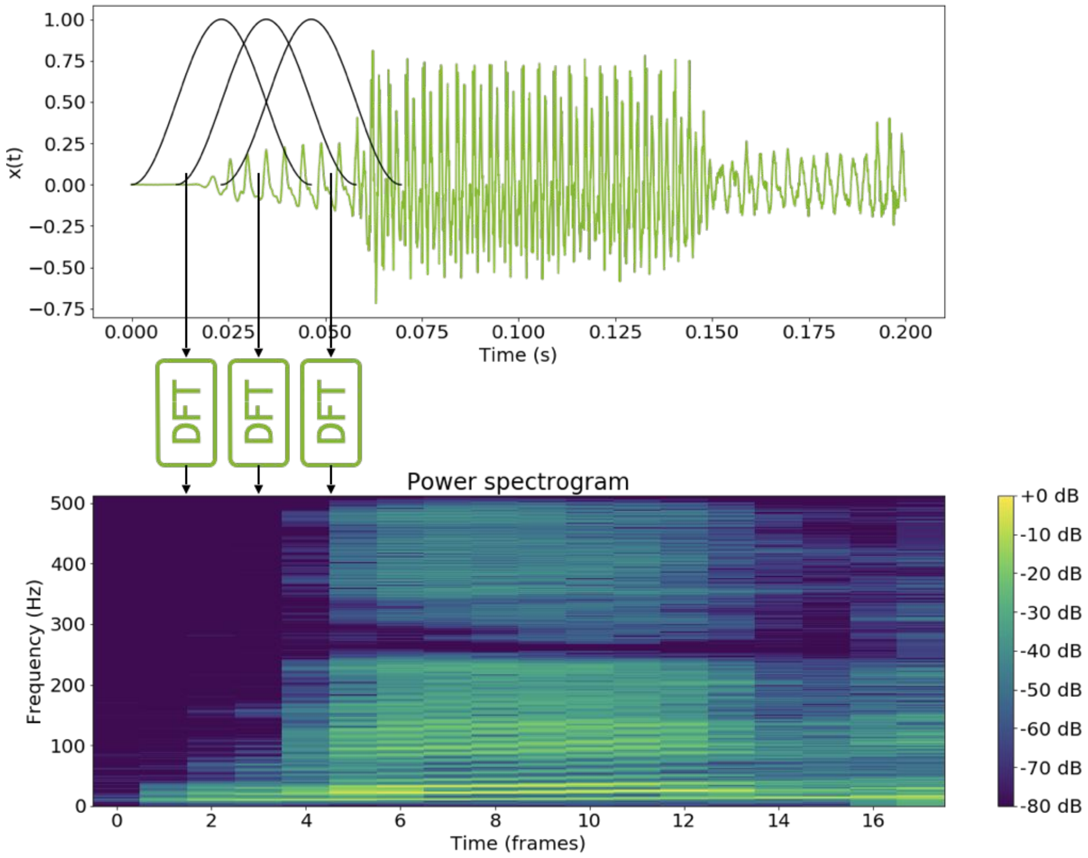


Short-time Fourier transform

FFT + Windowing



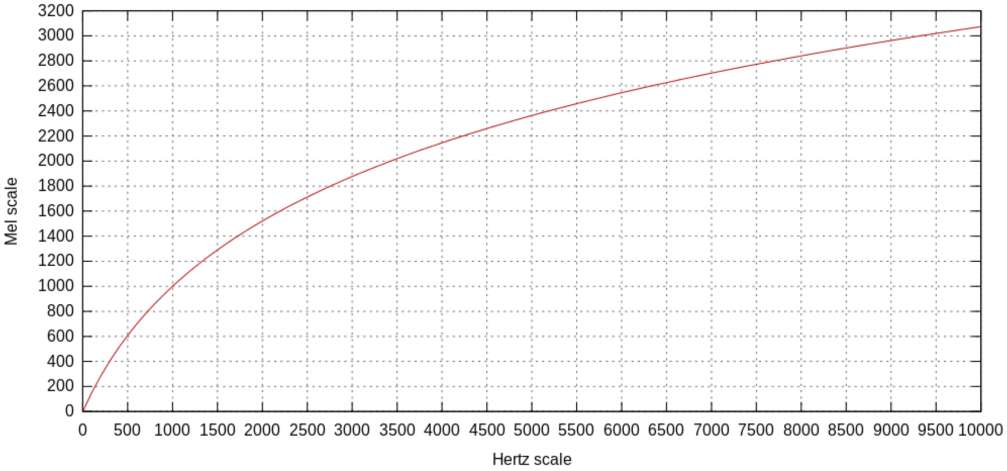
Spectrograms

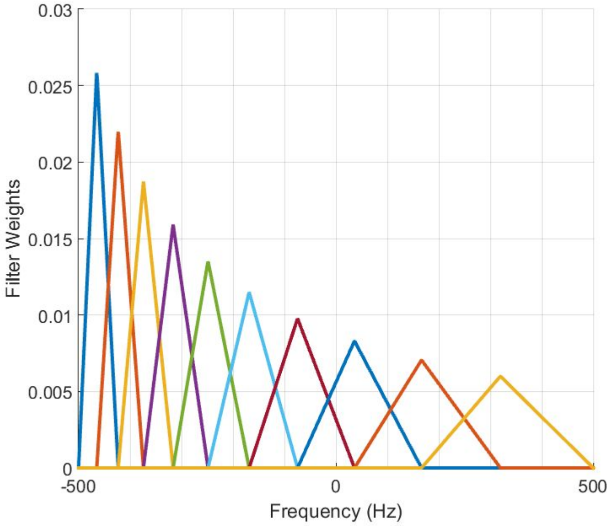


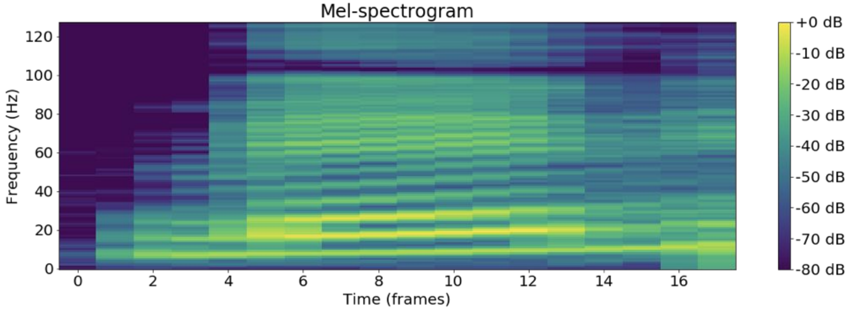
Mel Scale

$$m = 2595 \log_{10}\left(1 + \frac{f}{700}\right) = 1127 \ln\left(1 + \frac{f}{700}\right)$$

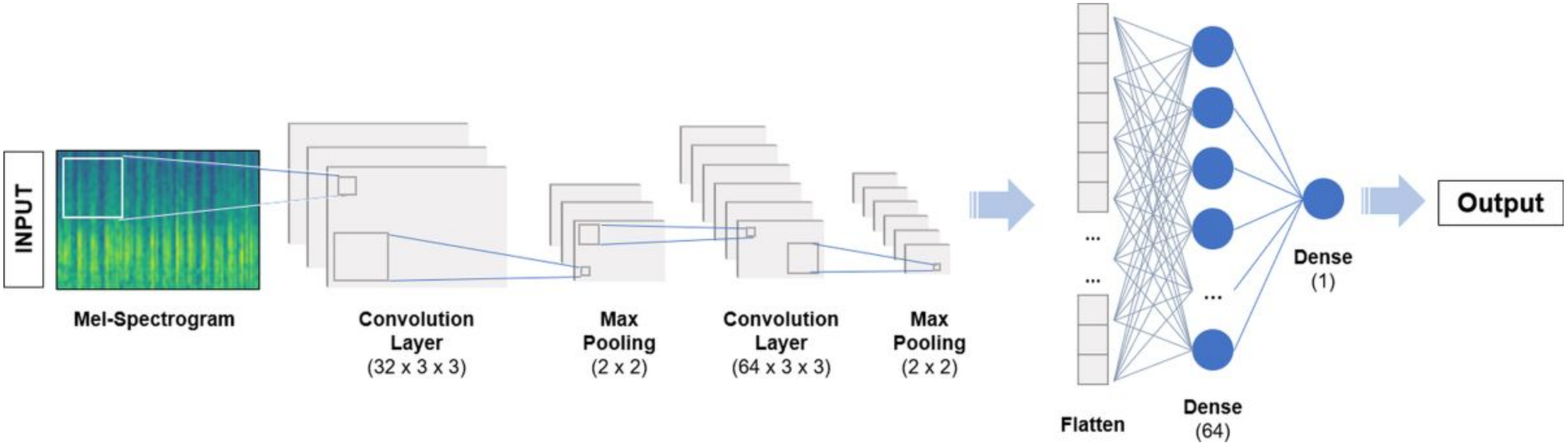
$$f = 700\left(10^{\frac{m}{2595}} - 1\right) = 700\left(e^{\frac{m}{1127}} - 1\right)$$







CNN-approach



Automatic Speech Recognition

Task:

- Transform speech (audio) to text

Also known as:

- STT - Speech To Text

WER

Word Error Rate

How to compute?

Edit path from reference to prediction

- **S** – substitution count
- **D** – deletions count
- ***I*** – insertions count
- **C** – correct count
- **N** – **S** + **D** +**C** - Total word count in refrence

```
True: quick  brown     fox jumped over   a lazy dog
Pred: quick  brow   an fox jumped over     lazy dog
```

$$WER = \frac{S+D+I}{N} = \frac{S+D+I}{S+D+C}$$

CER

Character Error Rate

The same, but on character level

Questions:

1) what is more important?
2) what is more difficult to minimize?

LibriSpeech

Domain:
- audiobooks

Parts:
- clean - low-WER speakers
- other - high-WER speakers

Features:
- 10-20s audio
- long sentences
- complex language

subset	hours	per-spk minutes	female spkrs	male spkrs	total spkrs
dev-clean	5.4	8	20	20	40
test-clean	5.4	8	20	20	40
dev-other	5.3	10	16	17	33
test-other	5.1	10	17	16	33
train-clean-100	100.6	25	125	126	251
train-clean-360	363.6	25	439	482	921
train-other-500	496.7	30	564	602	1166

[1]

Human WER:
- test-clean: **5.83**
- test-other: **12.69**

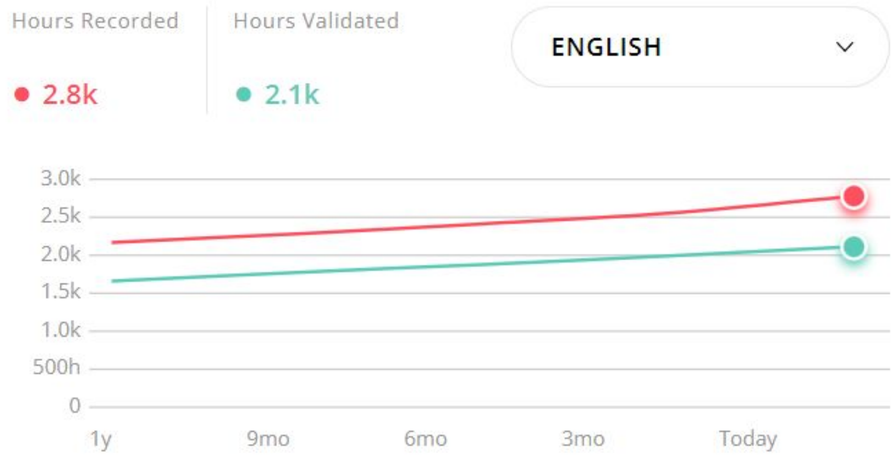
Mozilla Common Voice

Domain:

- Short random phrases
- Multiple languages

Features:

- Crowdsourced
- Simple language
- Short phrases
- Frequently updated and validated

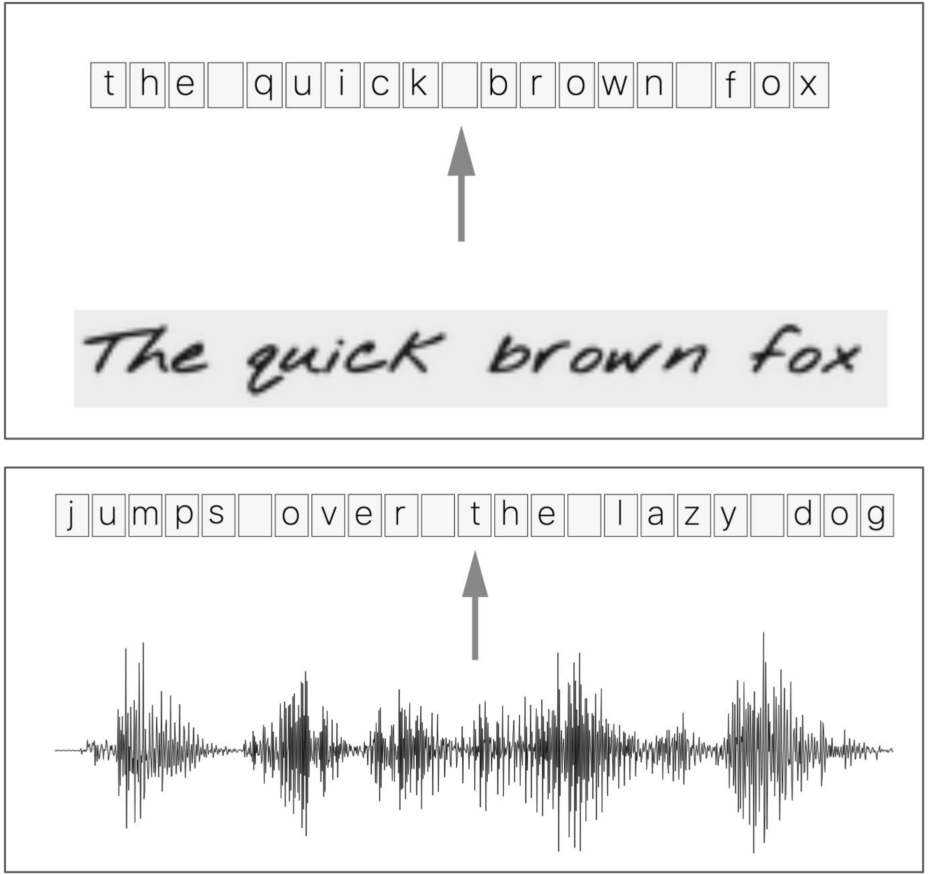


Fundamental problem

Alignment problem

- Variable length input
- Variable length output
- No alignment

How to train?



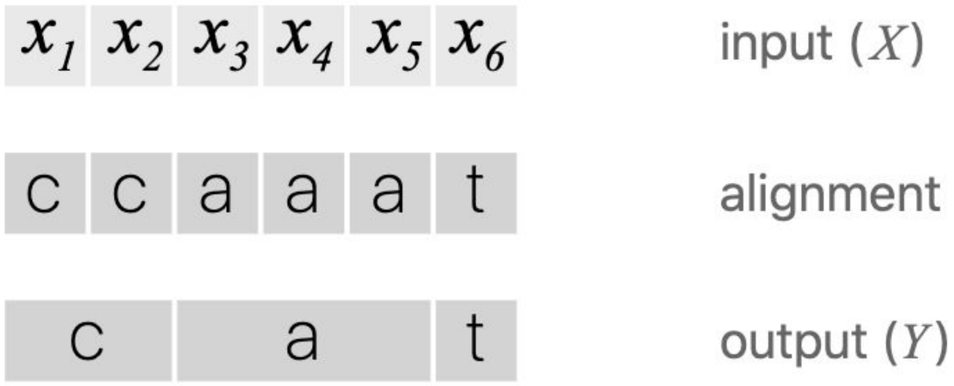
[3]

Connectionist Temporal Classification

2006
Idea:

- Split input into frames
- Classify each frame into *num_letters* classes
- Merge consecutive letters

Issues?

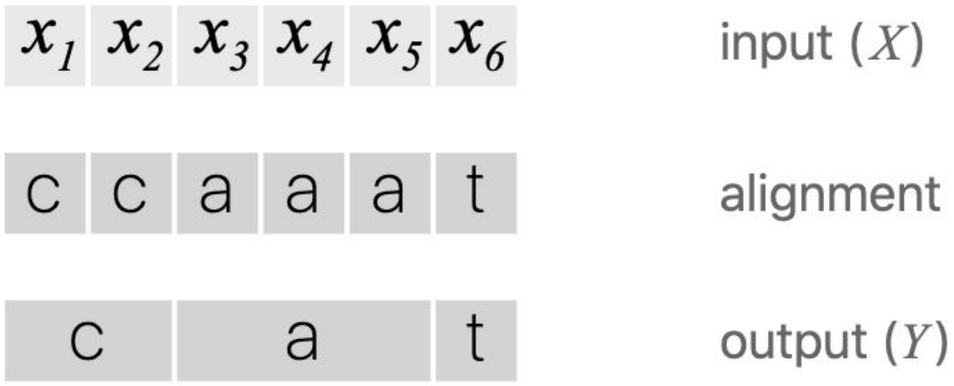


[3]

Connectionist Temporal Classification

Idea:

- Split input into frames
- Classify each frame into *num_letters* classes
- Merge consecutive letters

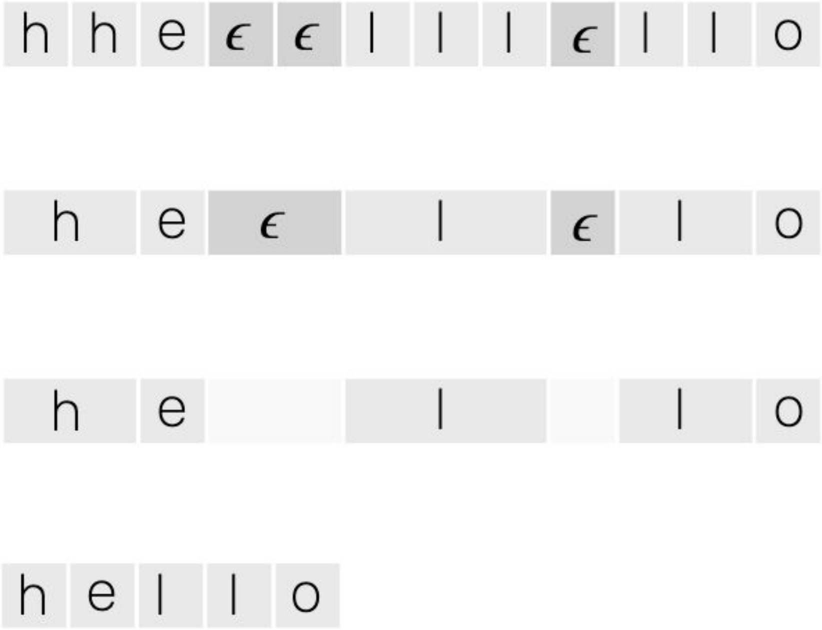


[3]

Issues?

- Multiple consecutive letters in target word (e.g.: *he**ll**o*)
- Silence between words and letters

CTC: Empty token



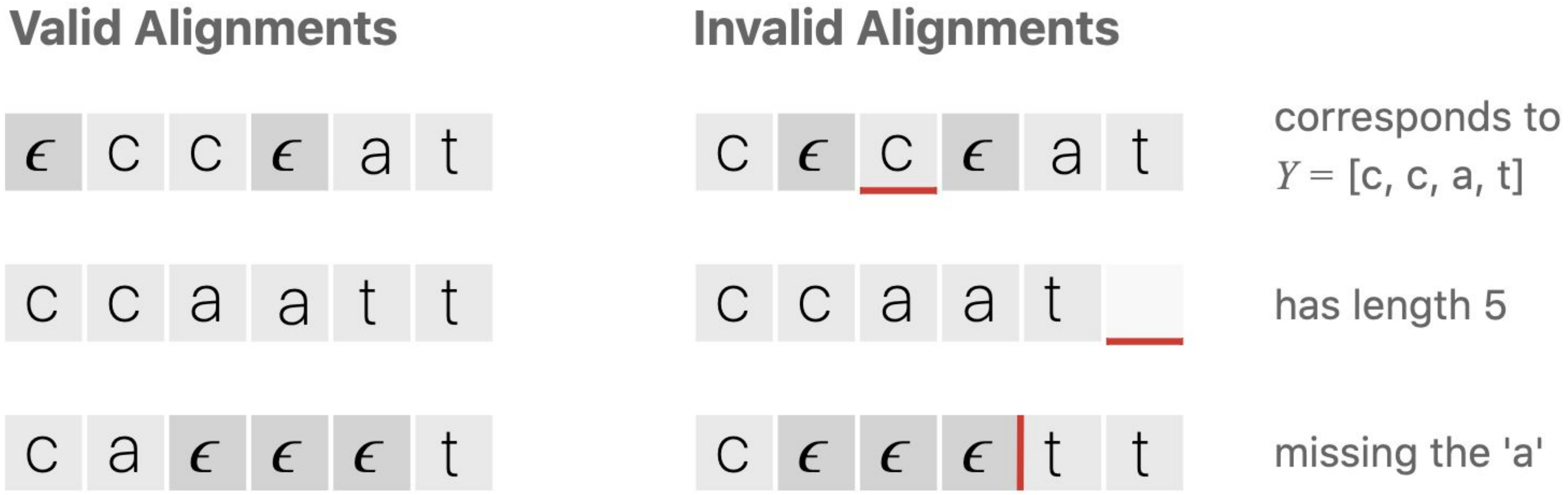
[3]

First, merge repeat characters.

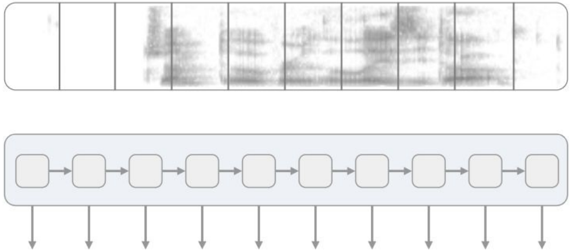
Then, remove any ϵ tokens.

The remaining characters are the output.

CTC: Empty token



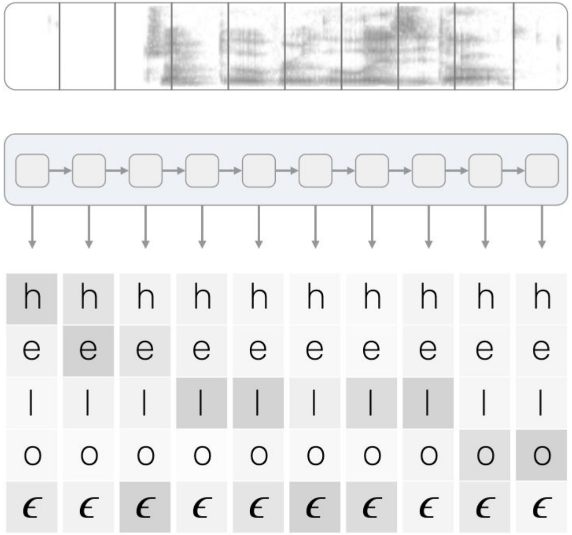
[3]



We start with an input sequence,
like a spectrogram of audio.

The input is fed into an RNN,
for example.

[3]

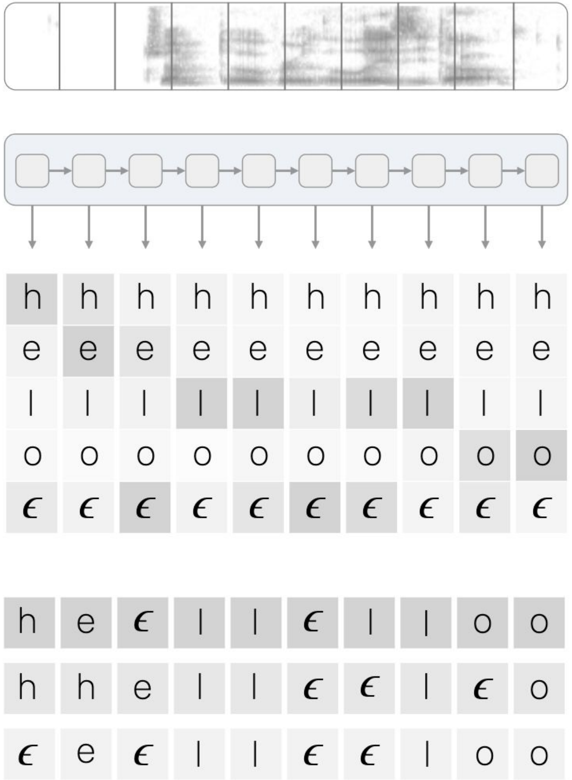


We start with an input sequence, like a spectrogram of audio.

The input is fed into an RNN, for example.

The network gives $p_t(a \mid X)$, a distribution over the outputs $\{h, e, l, o, \epsilon\}$ for each input step.

[3]



We start with an input sequence, like a spectrogram of audio.

The input is fed into an RNN, for example.

The network gives $p_t(a \mid X)$, a distribution over the outputs $\{h, e, l, o, \epsilon\}$ for each input step.

With the per time-step output distribution, we compute the probability of different sequences

[3]

We start with an input sequence, like a spectrogram of audio.

The input is fed into an RNN, for example.

The network gives $p_t(a \mid X)$, a distribution over the outputs {h, e, l, o, ϵ} for each input step.

With the per time-step output distribution, we compute the probability of different sequences

By marginalizing over alignments, we get a distribution over outputs.

[3]

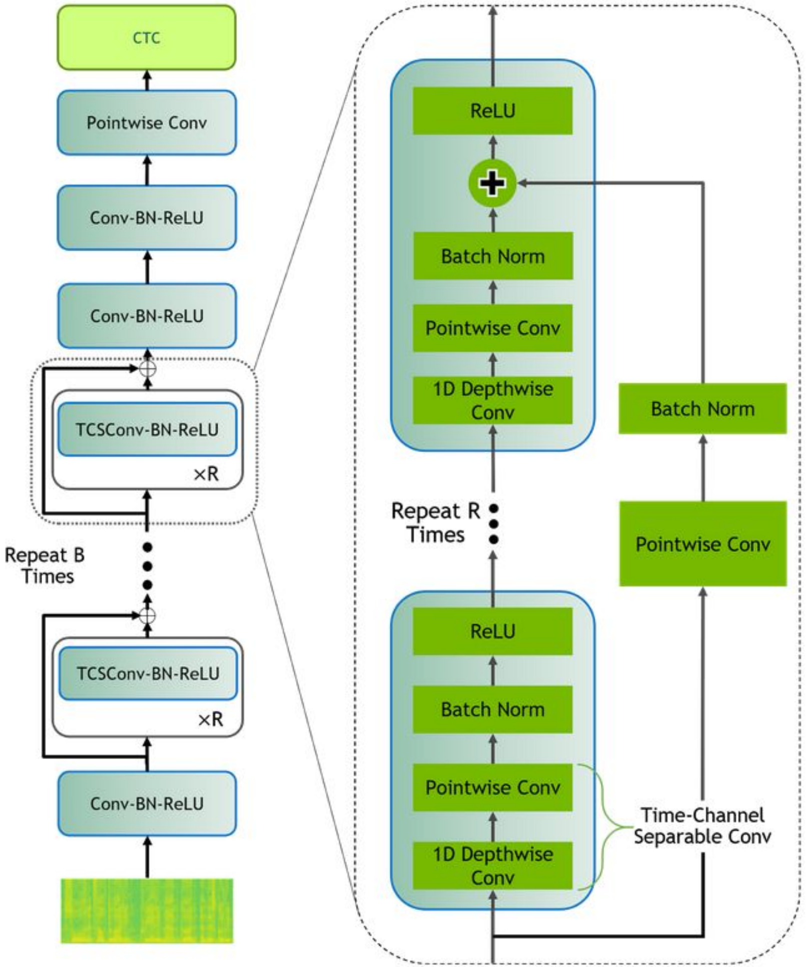
[3]

Loss Function:

$$p(Y \mid X) \quad = \quad \sum_{A \in \mathcal{A}_{X,Y}} \quad \prod_{t=1}^{T} p_t(a_t \mid X)$$

The CTC conditional **probability**	**marginalizes** over the set of valid alignments	computing the **probability** for a single alignment step-by-step.

QuartzNet



CTC
Pointwise Conv
Conv-BN-ReLU
Conv-BN-ReLU
TCSConv-BN-ReLU
×R
Repeat B Times
TCSConv-BN-ReLU
×R
Conv-BN-ReLU
ReLU
Batch Norm
Pointwise Conv
1D Depthwise Conv
Batch Norm
Repeat R Times
Pointwise Conv
ReLU
Batch Norm
Pointwise Conv
1D Depthwise Conv
Time-Channel Separable Conv

Language Models (LM): motivation

Problem:

- Spelling of a word heavily depends on its context
- We always want ***more*** data
- Labeled audio data is difficult to obtain

hypo 1: let's go ***two*** a movie (score: 0.21)
hypo 2: let's go ***to*** a movie (score: 0.19)
hypo 3: let's go ***too*** a movie (score: 0.13)

Idea:

- ASR predicts texts
- Unlabeled text data is ***very easy*** to get
- Let's make use of it!

LM: motivation

Language model - a model that estimates the probability of a text.

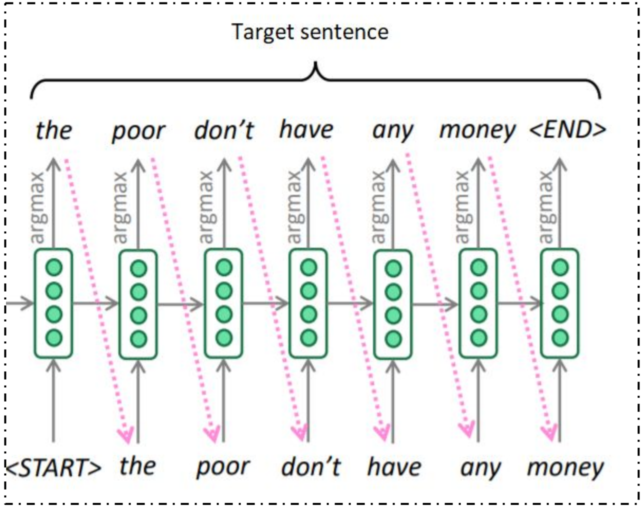
Usually requires unlabeled text corpus to train.

Examples:

- N-gramms (simple)
- Neural networks (compex) e.g: Bert, GPT-3

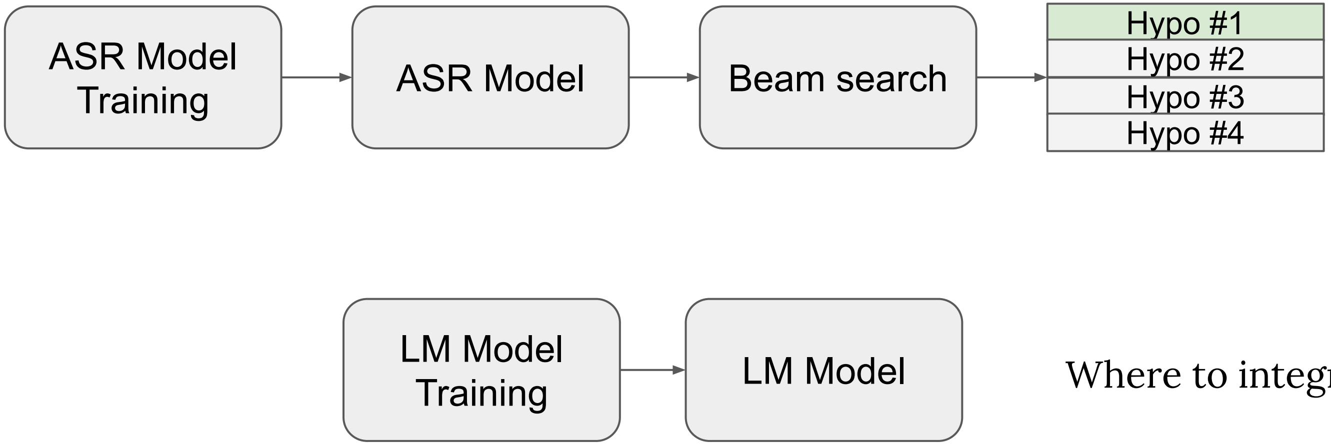
P(let's go ***two*** a movie) = 0.01

P(let's go **to** a movie) = 0.6



[1]

LM: ASR pipeline



Where to integrate LM?

LM: final hypothesis rescoring

a.k.a.: *second-pass rescoring*

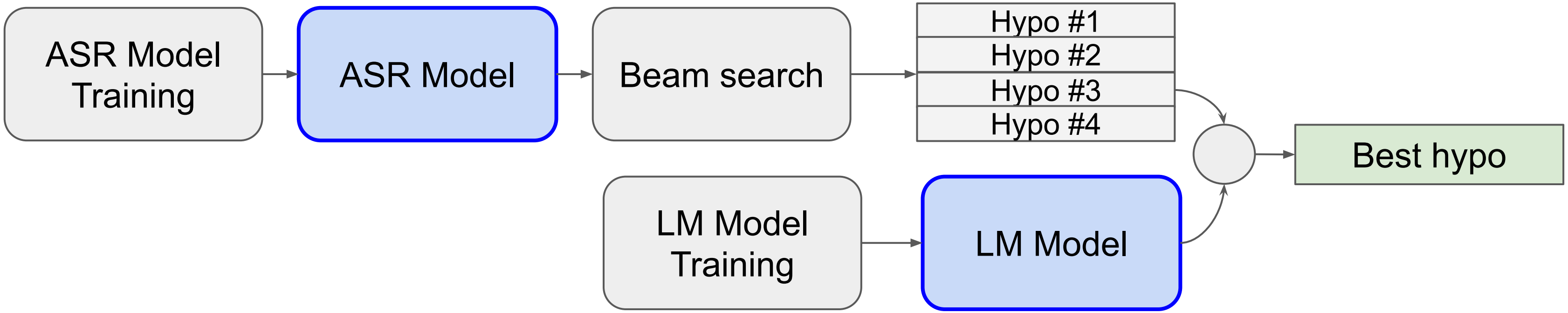
Oracle WER – lowest WER among all beam search output hypothesis

Prerequisites:

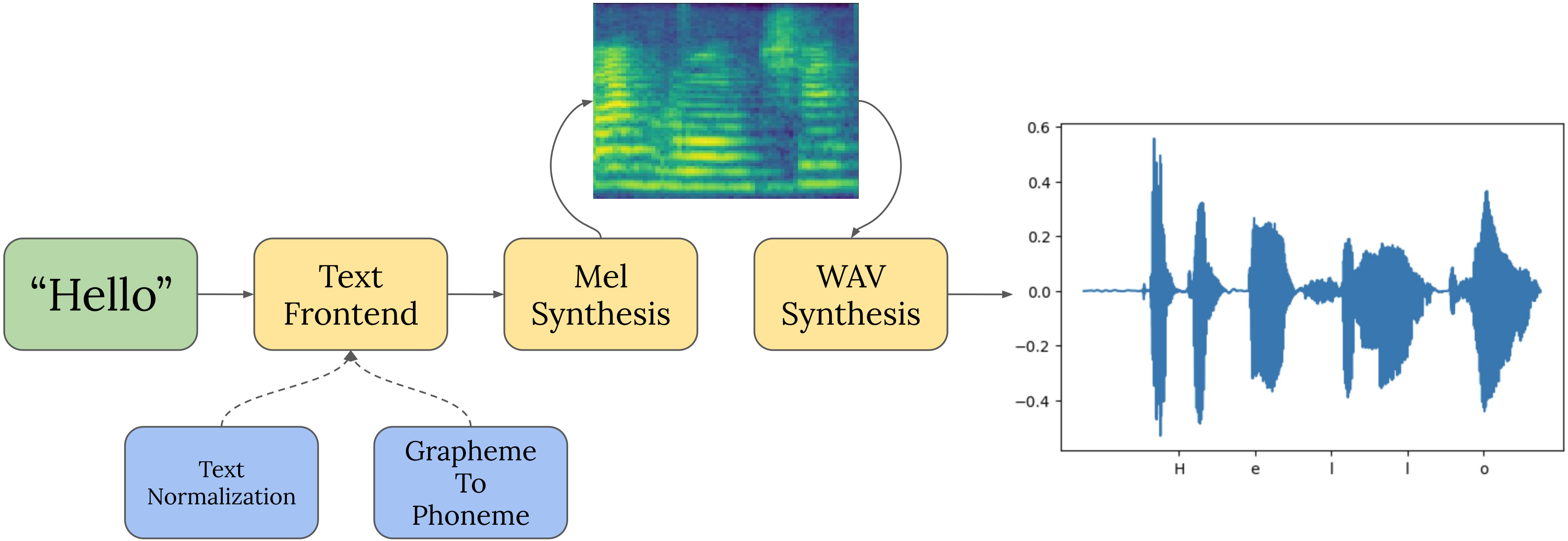
- Trained ASR
- Trained LM

Concept: Rescore beam-search output with LM scores

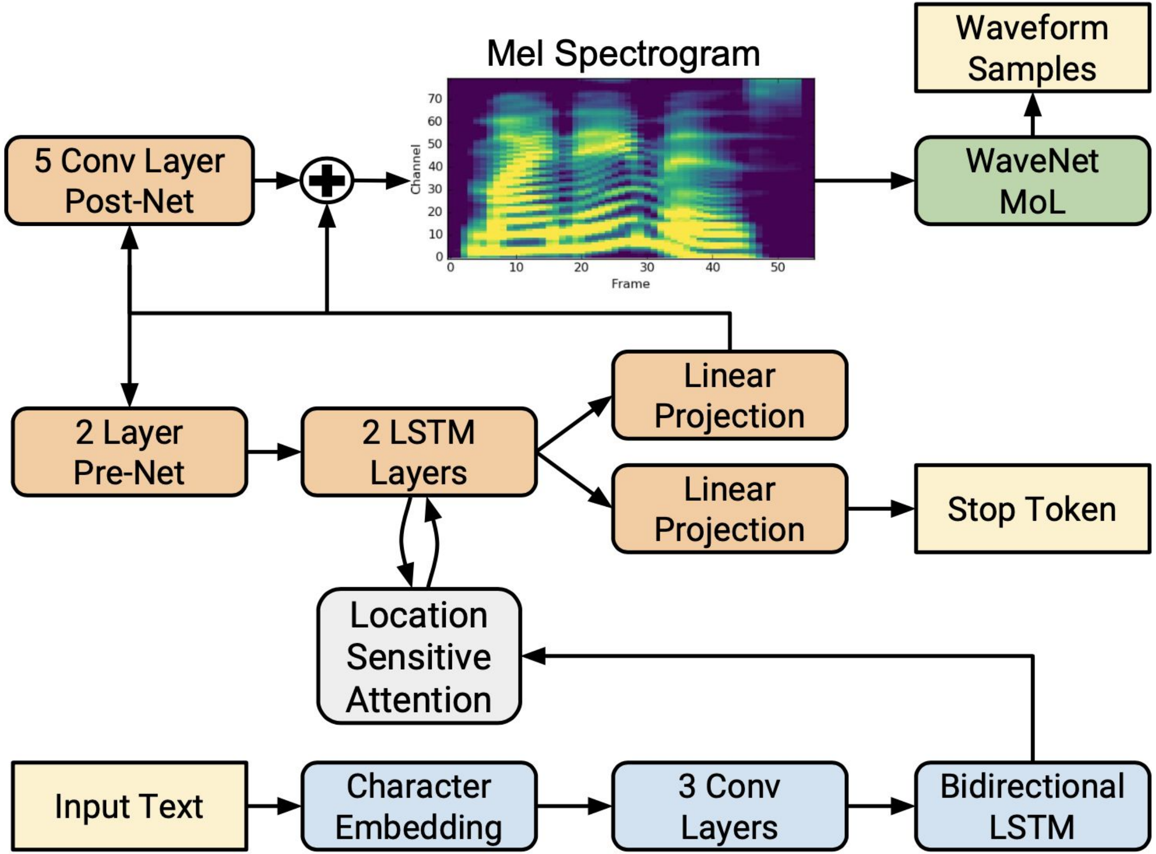
$$P_{final} = P_{ASR}(Y|X) + \alpha \cdot P_{LM}(Y) + \beta \cdot len(Y)$$



Text-to-Speech (TTS)



Tacotron 2


Waveform Samples
Mel Spectrogram
5 Conv Layer Post-Net
WaveNet MoL
Channel
Frame
0
10
20
30
40
50
60
70
2 Layer Pre-Net
2 LSTM Layers
Linear Projection
Linear Projection
Stop Token
Location Sensitive Attention
Input Text
Character Embedding
3 Conv Layers
Bidirectional LSTM