음성학회
학제간 워크숍

한국전자통신연구원

초지능창의연구소, 복합지능연구실

박기영

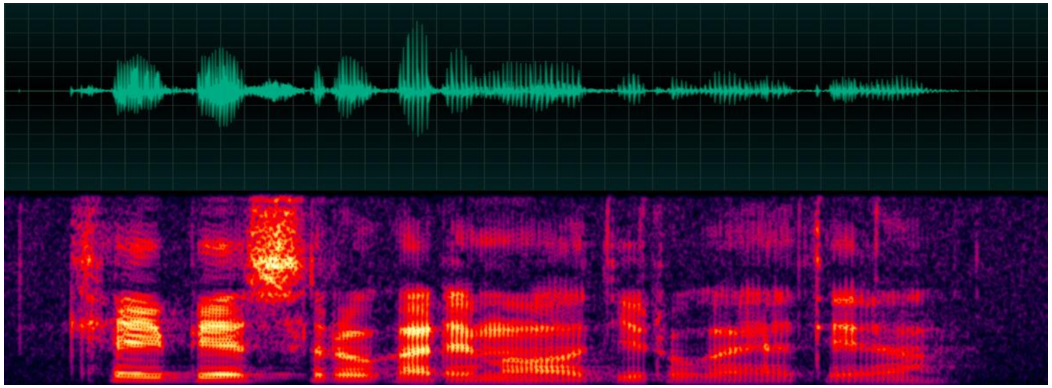
강의내용

- https://github.com/pkyoung/a1003
  - doc/psss2507.pdf
- PART I
  - 음성 인식 개요
  - 음성 인식의 성능 측정
- PART II
  - 특징 추출 방법
- PART III
  - 트랜스포머 기반 종단형 음성인식 기술

PART I: 음성인식 개요

What is speech recognition

- ASR(Automatic Speech Recognition), STT(Speech-to-text)



  - Isolated, Connected, Continuous, Keyword Spotting
  - Speaker Dependent/Independent
- Difference with Image/Video Classification
  - Sequence Generation Problem

Recent Landmarks in ASR

1986,
Error Backpropagation
Algorithm

Geoffrey E. Hinton

2016

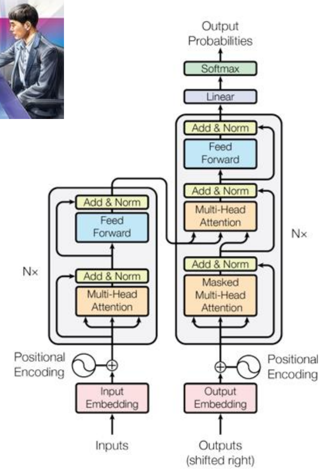
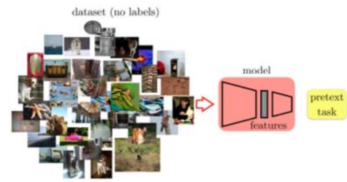
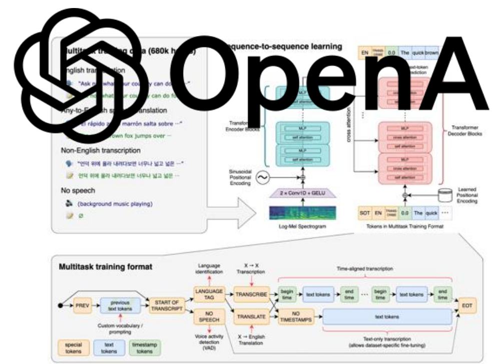


Figure 1: The Transformer - model architecture.





2018, BERT, GPT

2017, Attention is all you need

2022, whisper

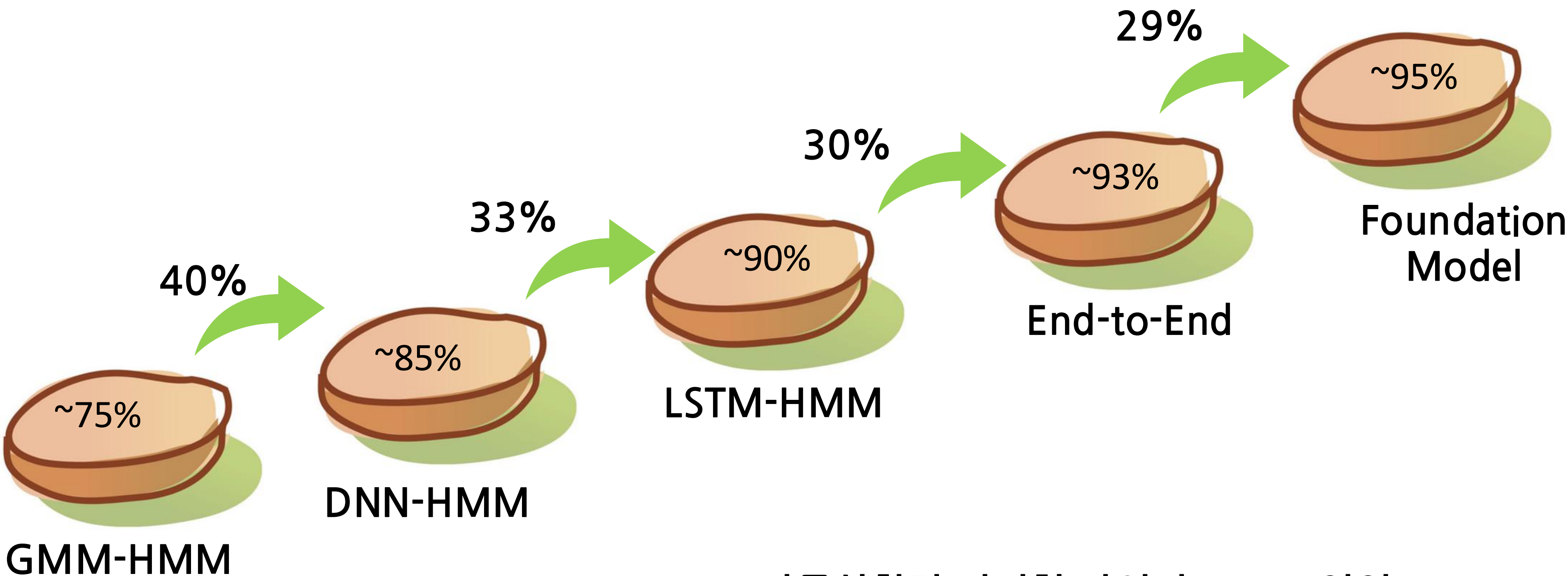
2023, whisper large-v3

2012, ImageNet Classification with Deep Convolutional Neural Networks

2009, An Efficient Learning Procedure for Deep Boltzmann Machines

2006, A Fast Learning Algorithm for Deep Belief Nets

음성인식 기술의 현재 성능



* 사무실환경 비정형 자연어 LVCSR영역
* 상대적 체감 성능

음성인식의 어려움

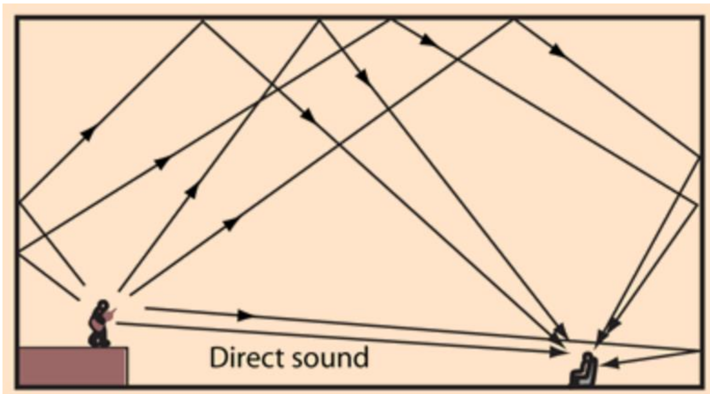
- 화자별 변이
  - Age, gender, accent, dialect
- 배경잡음 및 발화 환경
  - Noise, echo
  - Far-field (reverberation)
- 비정형 발화 & 유창성
  - 간투사(Filler words: “음”, “그러니까”, “uh”, “um”)
  - hesitations, self-corrections, repetition
  - Code-switching
- Homophones & Ambiguities
  - Words with similar sounds (e.g., “to, two, too”)
  - Context-dependent disambiguation

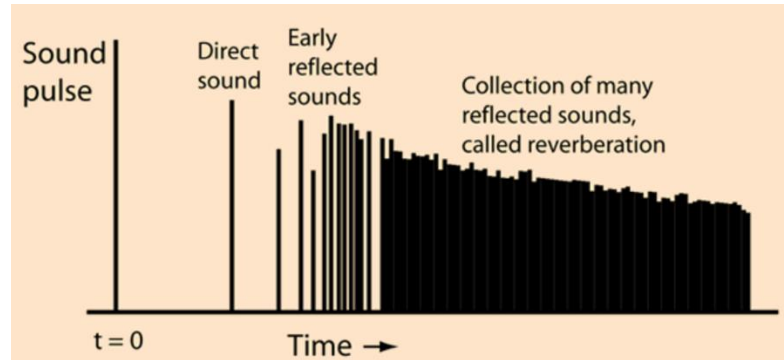
배경 잡음

- Stationary vs Non-stationary Noise
  - 자동차 엔진소리, 공조기 소리
  - 음악 잡음, 주변 사람 말소리
- 측정 Metric
  - Signal-to-noise (SNR)
  - 10*log(SignalPower/NoisePower)
  - 0dB, 10dB, -10dB
  - SDR: Signal-to-Distortion Ratio
  - SIR: Signal-to-Inteferance Ratio

발화 환경

- Closed-talk vs Distant-talk
- 반사된 소리: Reververation(반향, 잔향)



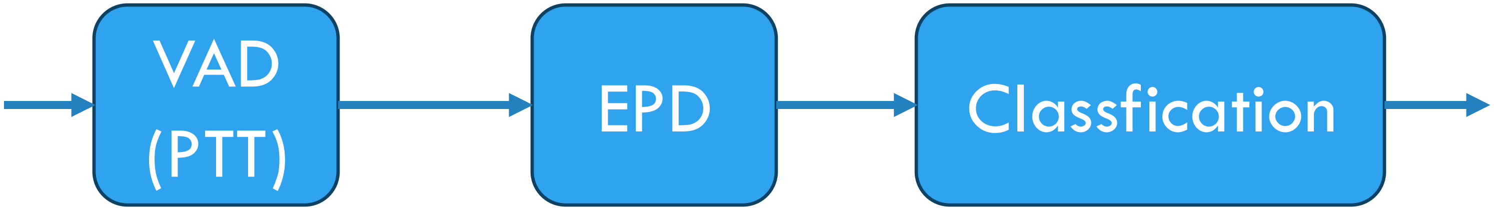


- Metric
  - rt60 (강당, 공연장: 1.5~2.5초)

http://hyperphysics.phy-astr.gsu.edu/hbase/Acoustic/reverb.html

고립어 인식 및 Keyword Spotting

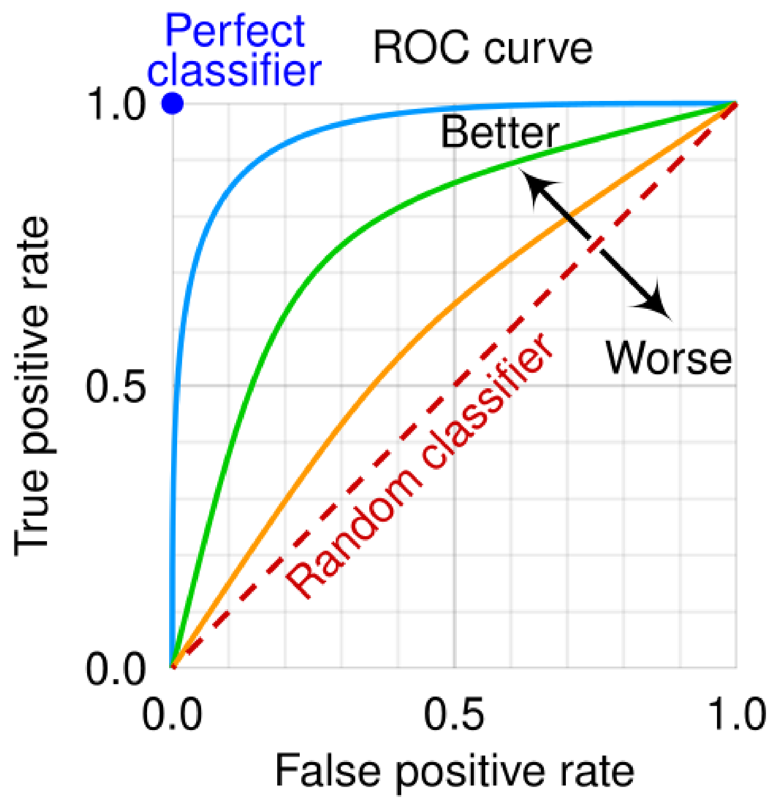
- Push-to-talk (PTT)
- Endpoint-Detection (EPD)
- VAD: Voice Activity Detection



<고립어 인식 절차>

Evaluation Metric: Classification

- keyword spotting 의 경우
  - 2class classfication problem
- Type of errors
  - False positive (false alarm)
  - False negative (false rejection)
- AUROC (Area under ROC)
- EER (Equal-error-rate)



Evaluation Metric

Accuracy $\frac{TP + TN}{TP + FP + FN + TN}$		Actual	
		Positive	Negative
Predicted	Positive	TP	FP
	Negative	FN	TN

$$\text{Precisoin} = \frac{\text{TP}}{\text{TP+FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}}$$

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$\text{Sensitivity} = \text{Recall}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN+FP}}$$

연속음성인식(LVCSR)

- Large vocabulary continuous speech recognition
- Find mapping between speech and text
  - X: speech/spectrogram → Y: sequence of words/tokens
  - X: 100Hz (100 frames / sec)
  - Y: 150 wpm = 2.5 words / sec
- How to handle length variation
  - HMM: state transition + self-transition
  - CTC: blank symbol
  - Sequence-to-sequence model: token generation

Evaluation Metric

- Types of Error
  - Substitution
  - Deletion
  - Insertion
- Error Rate (cab be > 1)
  - (S + D + I)/N
- Accuracy (can be < 0)
  - 1-(Error Rate)
- WER/CER/SER:
  - Word/Character/Sentence Error Rate

```
REF    : how is the weather today
REC/HYP: how was the better  to day

In Words: WER = 100%, Acc=0%
 - N= 5: how, is, the, weather, today
 - S = 2
 - D = 1
 - I = 2
```

```
how is  the weather today
how was the better  to day
```

```
In Chars: CER = 25%, Acc=75%
 - N= 20:
h,o,w,i,s,t,h,e,w,e,a,t,h,e,r,t,o,d,a,y
 - S = 3
 - D = 1
 - I = 1
```

```
how is  the weather today
how was the better  to day
```

```
In Sentence: SER = 100%, Acc=0%
 - N = 1
 - S = 1
```

한국어

- REF: 오늘 서울의 날씨가 어때
- REC: 음 오눌의 날씨 가 어때
- WER=4/4 = 1.0 Acc=0.0
  - N=4, 오늘, 날씨가, 어때요
  - S = 2, D = 1, I = 2, WER=5/4
  - S = 3, I = 1, WER=4/4
- CER= 4/10 = 0.4, Acc=0.6
  - N = 10
  - S = 1
  - D = 2
  - I = 1

	오늘	서울의	날씨가		어때
음	오눌의		날씨	가	어때

오늘	서울의	날씨가		어때
음	오눌의	날씨	가	어때

	오	늘	서	울	의	날	씨	가	어	때
음	오	눌			의	날	씨	가	어	때

측정방법

- Edit distance 측정
  - 최소편집거리
  - https://en.wikipedia.org/wiki/Edit_distance
- 사용도구
  - HResults (HTK)
  - compute-wer (kaldi)
  - sclite (NIST, ESPnet)
- 예)
  - compute-wer ark:ref.txt ark:rec.txt

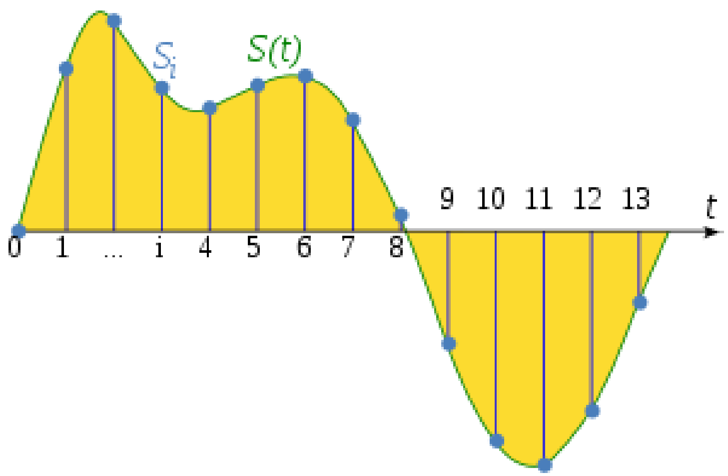
실습

- 음성인식 오유율 측정 실습
- https://colab.research.google.com/github/pkyoung/a1003/blob/main/local/wer.ipynb

PART II: 특징 추출

Sampling

- Analog to Digital Conversion (ADC)
  - Sampling



8kHz: Narrowband, 전화망
16kHz: Wideband
44.1/48kHz: High quality audio

  - Quantization
    - 16bit = 2-byte short integer -32768 < s <+32767
    - 24bit, 32bit

https://en.wikipedia.org/wiki/Sampling_(signal_processing)

Fourier Series / Transform

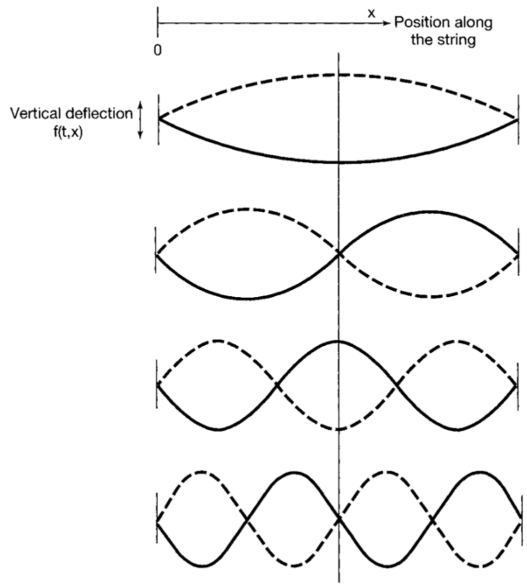
- Jean Baptiste Joseph Fourier(1768-1830)
- Claimed "any" periodic signal can be represented by series of harmonically related sinusoids

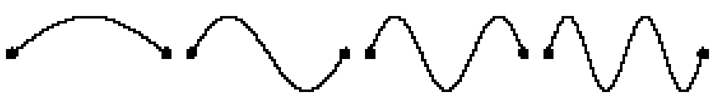
$$x[n] = \sum_{k=\langle N\rangle} a_k e^{jk\omega_0 n} = \sum_{k=\langle N\rangle} a_k e^{jk(2\pi/N)n},$$

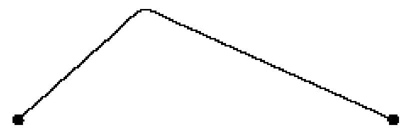
$$a_k = \frac{1}{N}\sum_{n=\langle N\rangle} x[n]e^{-jk\omega_0 n} = \frac{1}{N}\sum_{n=\langle N\rangle} x[n]e^{-jk(2\pi/N)n}.$$

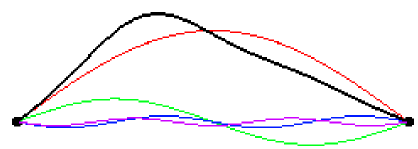
Vibrating Strings

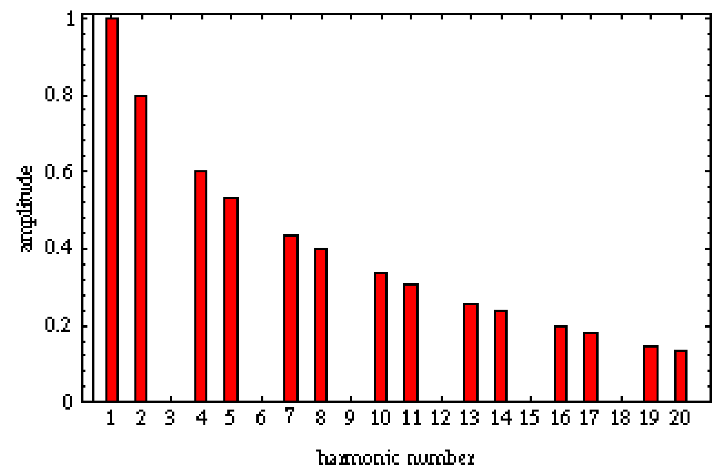
- Vibration of fixed-fixed string





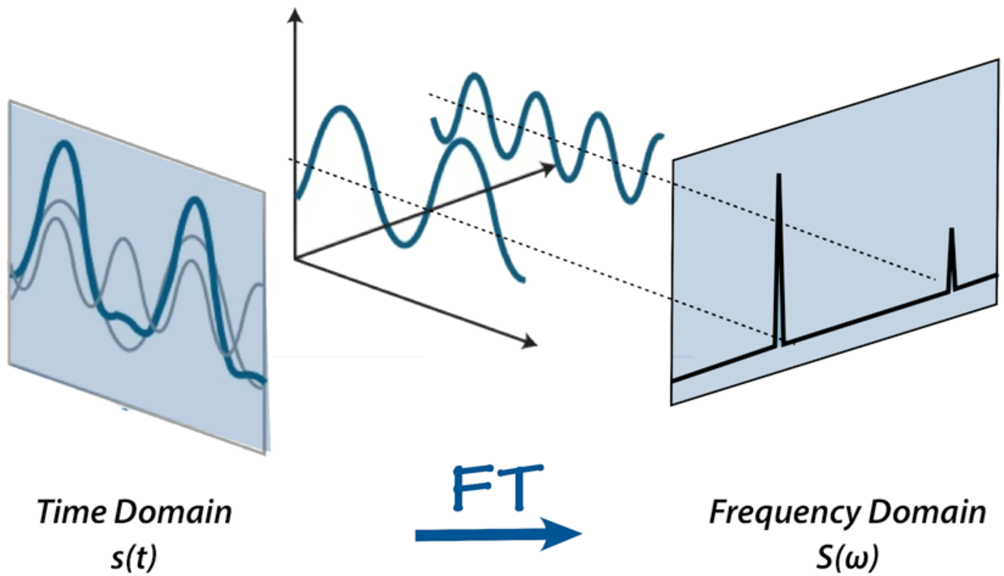






https://www.acs.psu.edu/drussell/demos/string/fixed.html

Spectrum



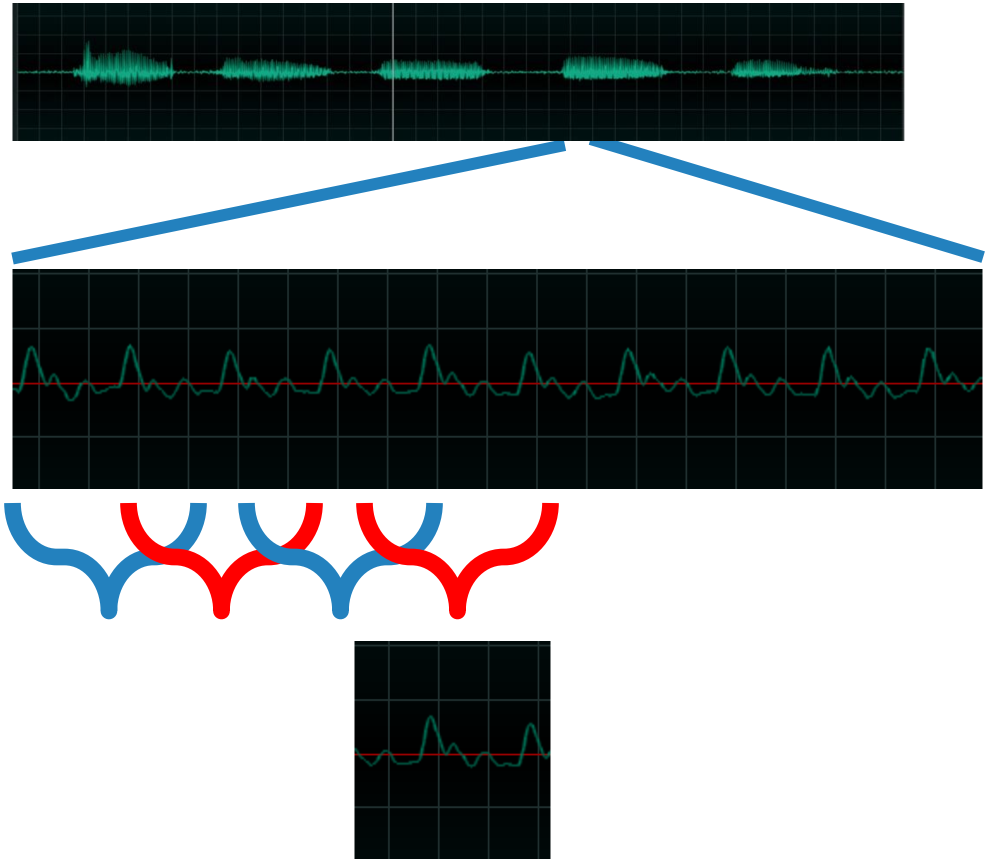
https://towardsdatascience.com/understanding-audio-data-fourier-transform-fft-spectrogram-and-speech-recognition-a4072d228520

Frame-wise Processing

4 sec

0.1 sec

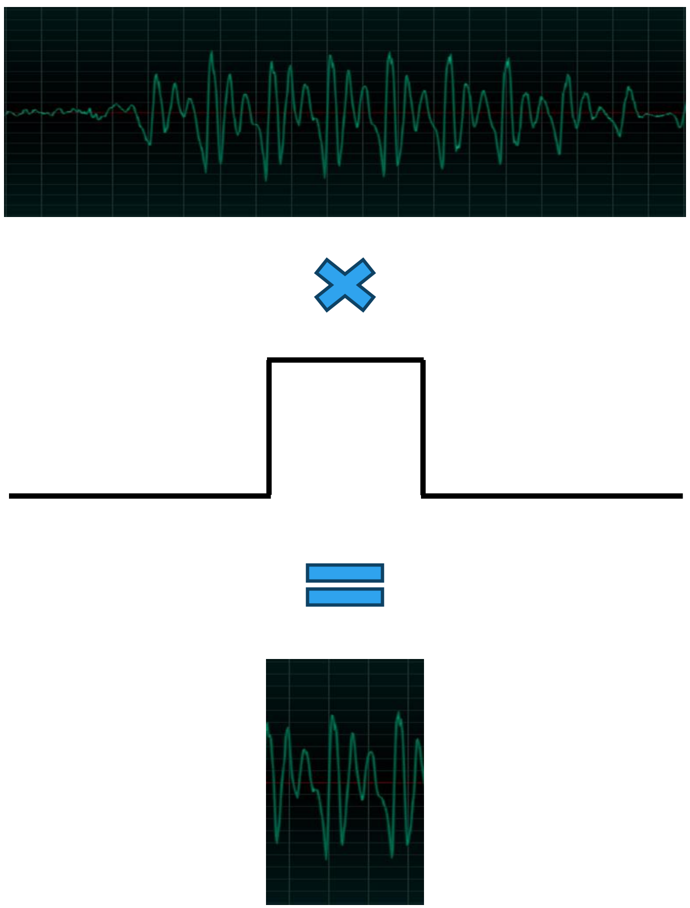
0.02 sec



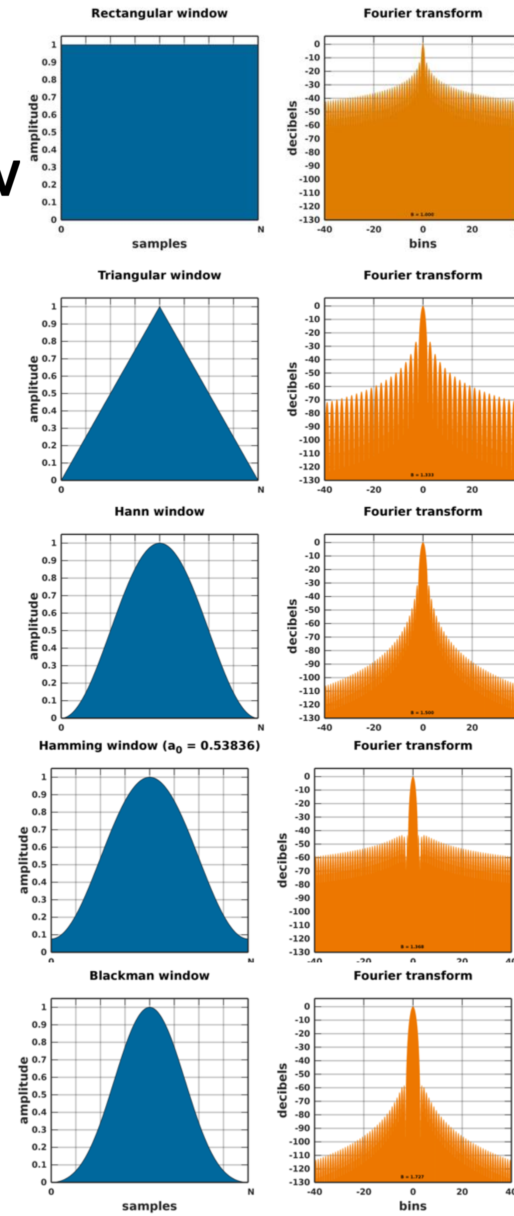
Window length,
Hop size

Windowing

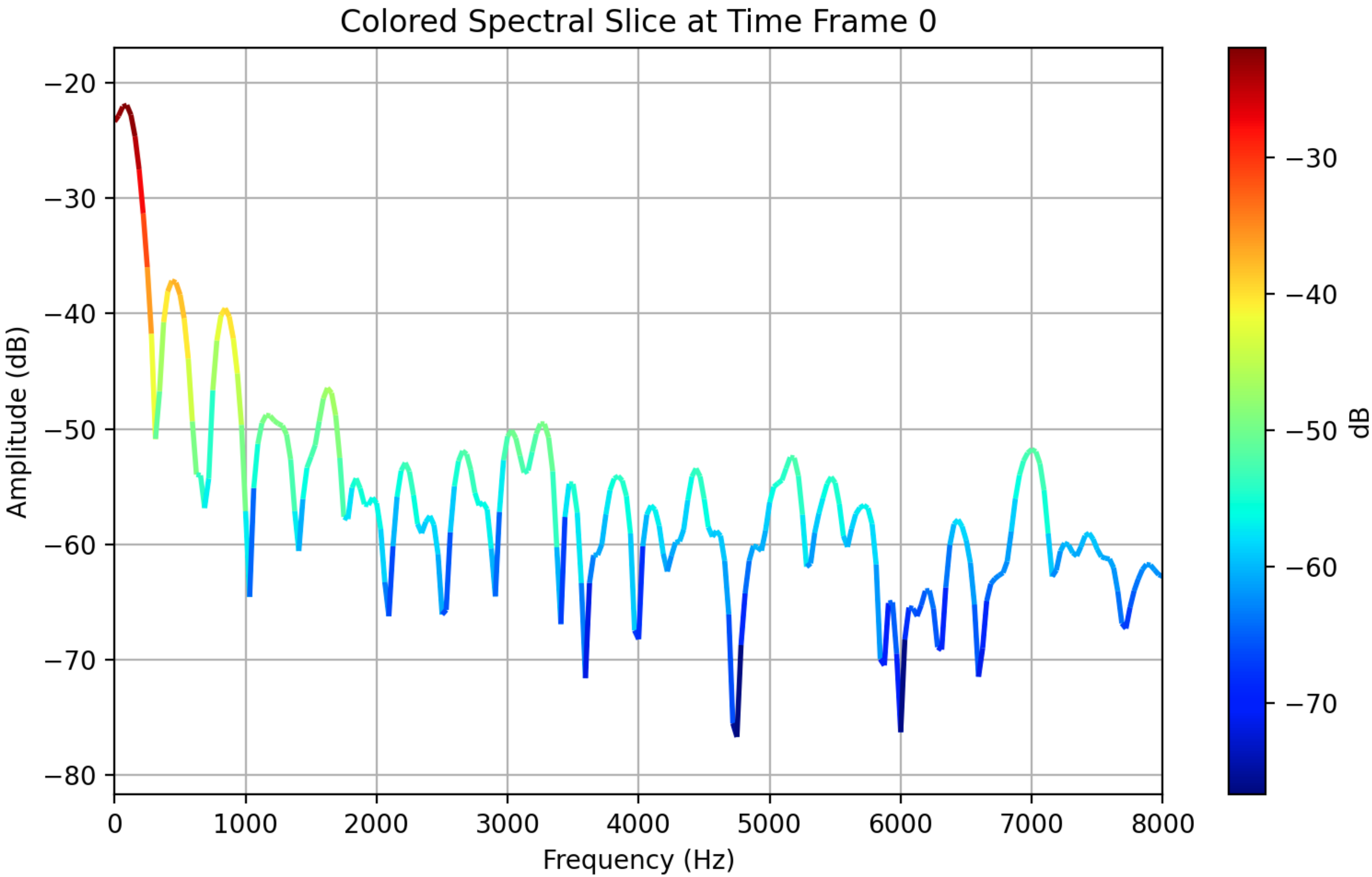
- Windowing = Framing
  - Element-wise multiplication with window



Rectagular
Triangular
Hann
Hamming
Blackman

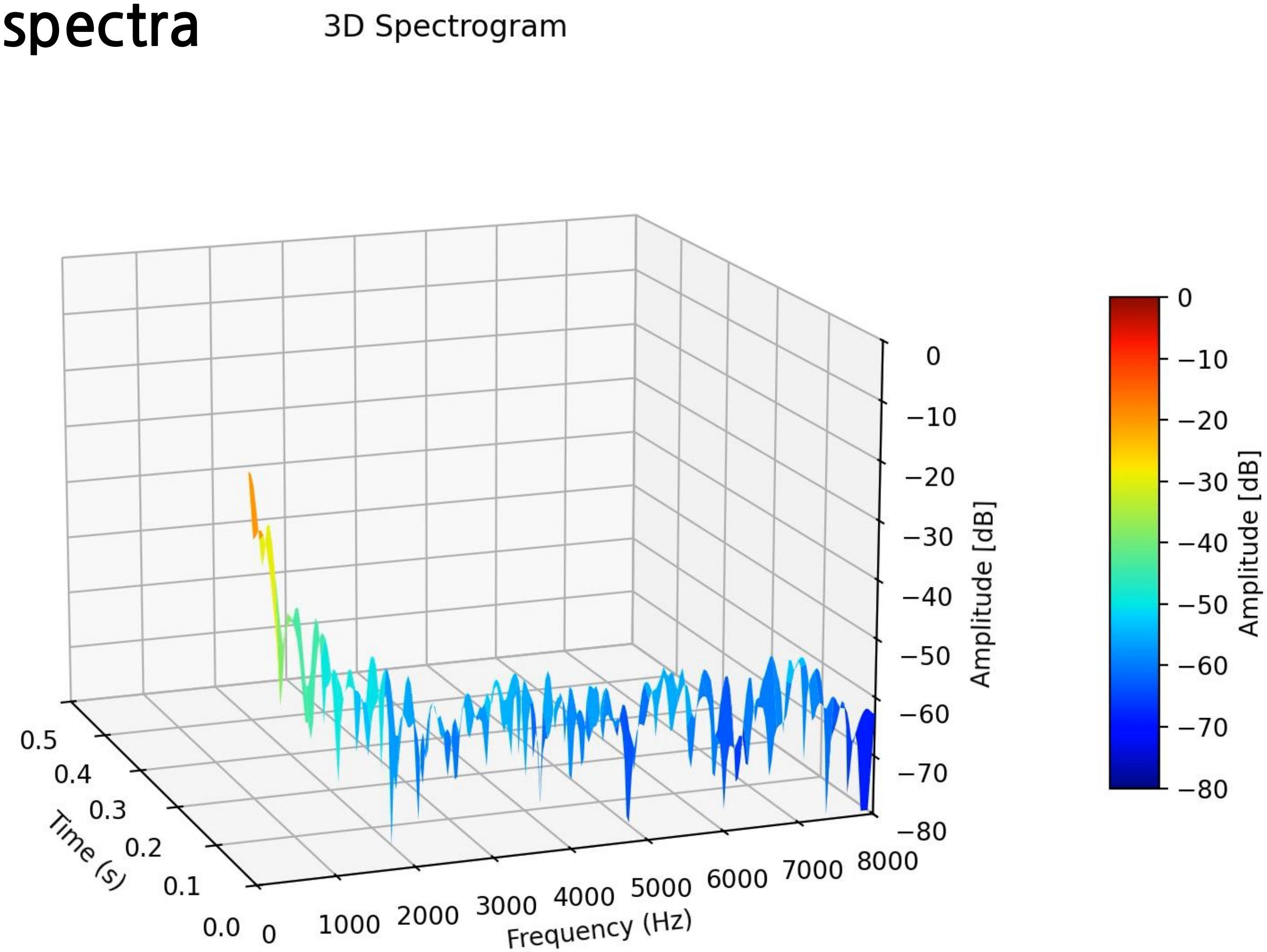


Spectrum

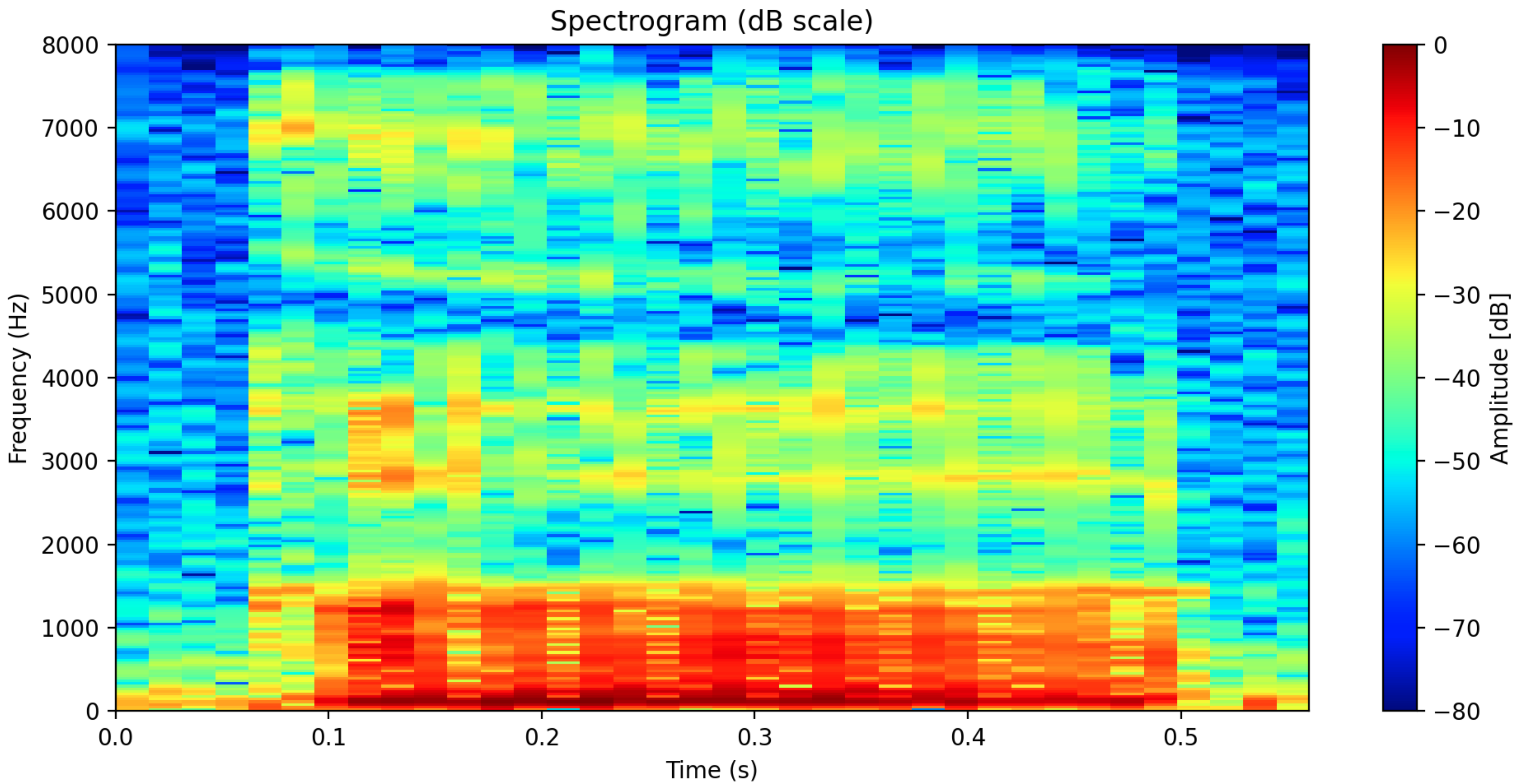

Colored Spectral Slice at Time Frame 0
Amplitude (dB)
−20
−30
−40
−50
−60
−70
−80
0
1000
2000
3000
4000
5000
6000
7000
8000
Frequency (Hz)
−30
−40
−50
−60
−70
dB

Spectrogram: 3D

- 3D plot of spectra

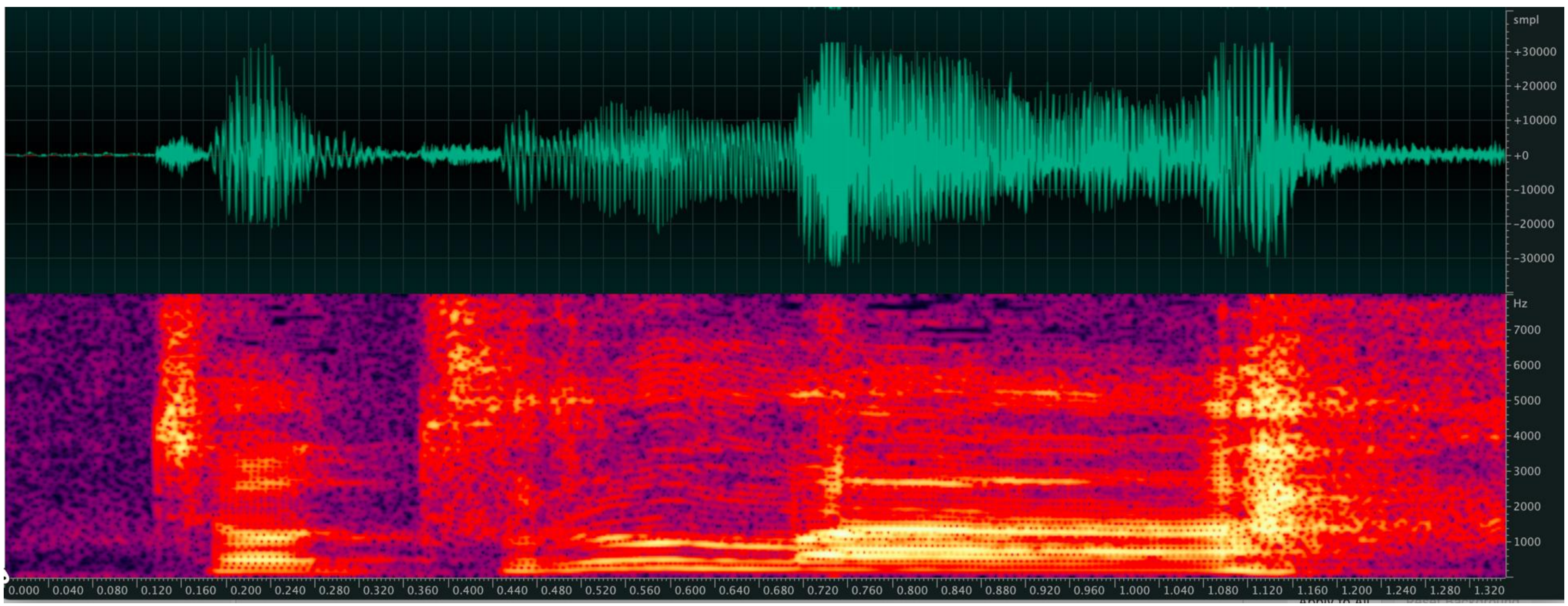


Spectrogram: 2D

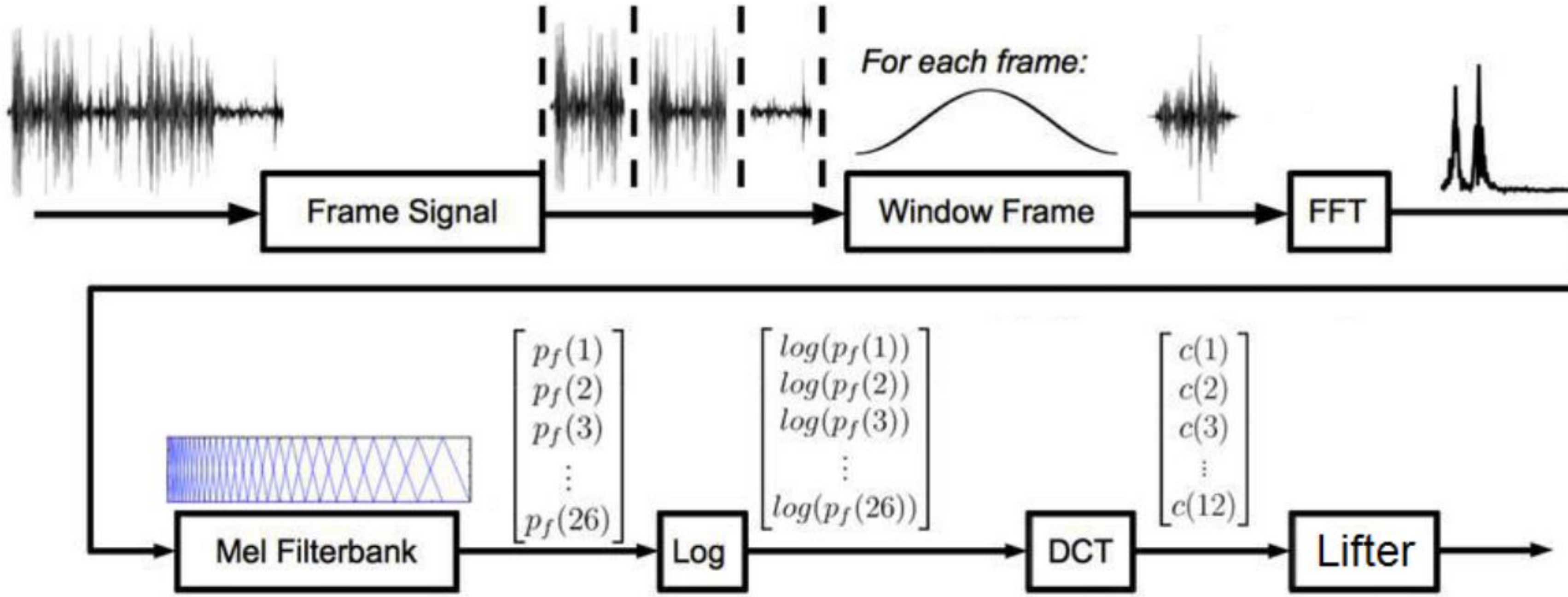


Spectrogram

- Matlab, Python, Adobe Audition, Audacity, …
- Frame Shift, Overlap, Window Length, Windowing, FFT points

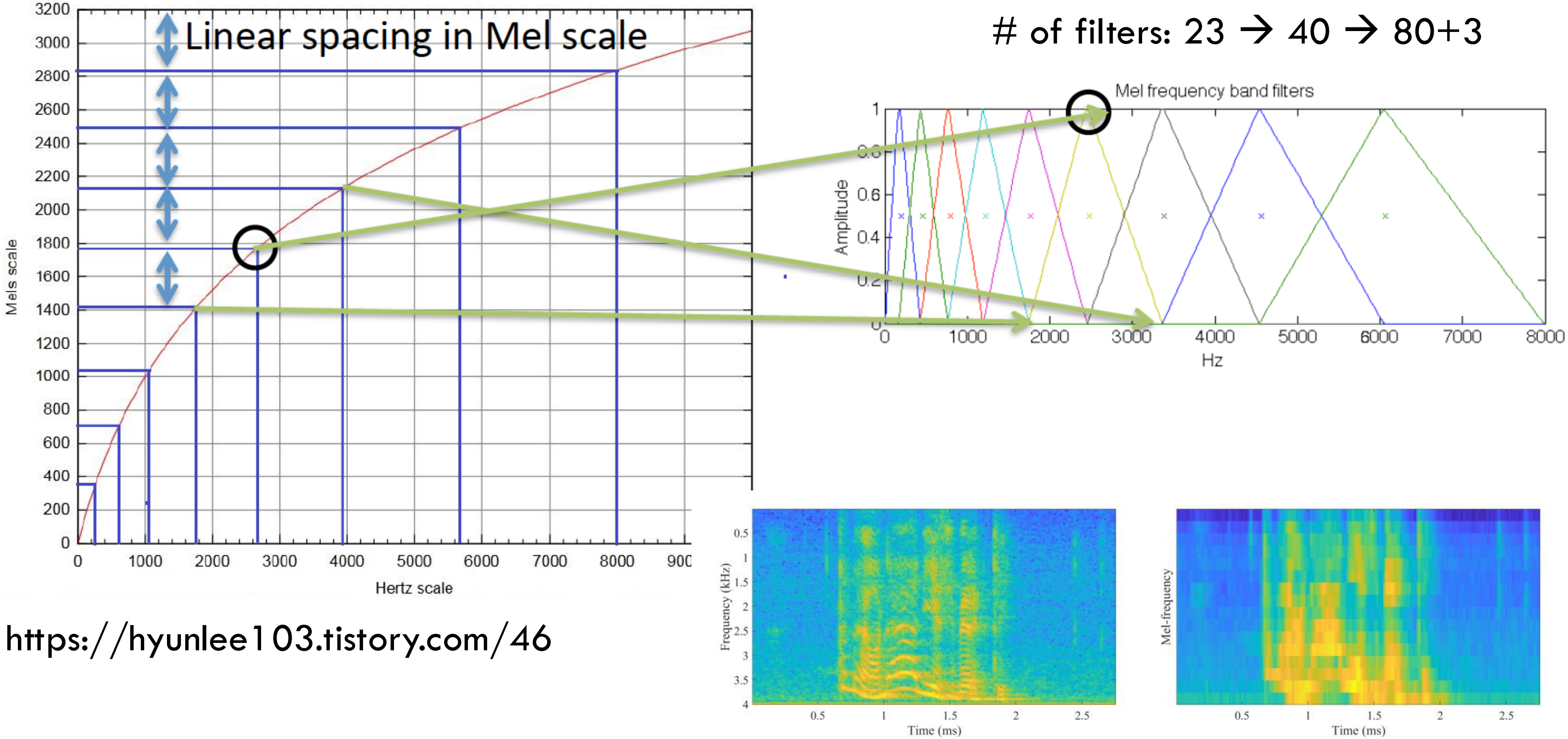


Feature extraction: MFCC



https://hyunlee103.tistory.com/46

Mel Filterbank



https://hyunlee103.tistory.com/46

https://wiki.aalto.fi/display/ITSP/Cepstrum+and+MFCC

Feature Normalization

- CMS
  - Cepstral Mean Substraction

Log-mel

t=0: [3, 2, 3, 4, 4, 3]
t=1: [4, 1, 3, 3, 3, 1]
t=2: [2, 3, 3, 5, 8, 2]

↓

mean vector = [3, 2, 3, 4, 5, 2]

⇒

Normalized log-mel

t=0: [0, 0, 0, 0, -1, 1]
t=1: [1, -2, 0, -1, -2, -1]
t=2: [-1, 1, 0, 1, 3, 0]

CMVN

- Cepstral Mean Variance Normalization
  - Zero-mean Unit Variance
- CMS: Cepstral Mean Substraction
  - Per Utterance
  - 채널/화자 효과를 제거하고 발성의 특성만 남김
- For Deep Learning
  - Global CMVN
  - For better convergence

실습

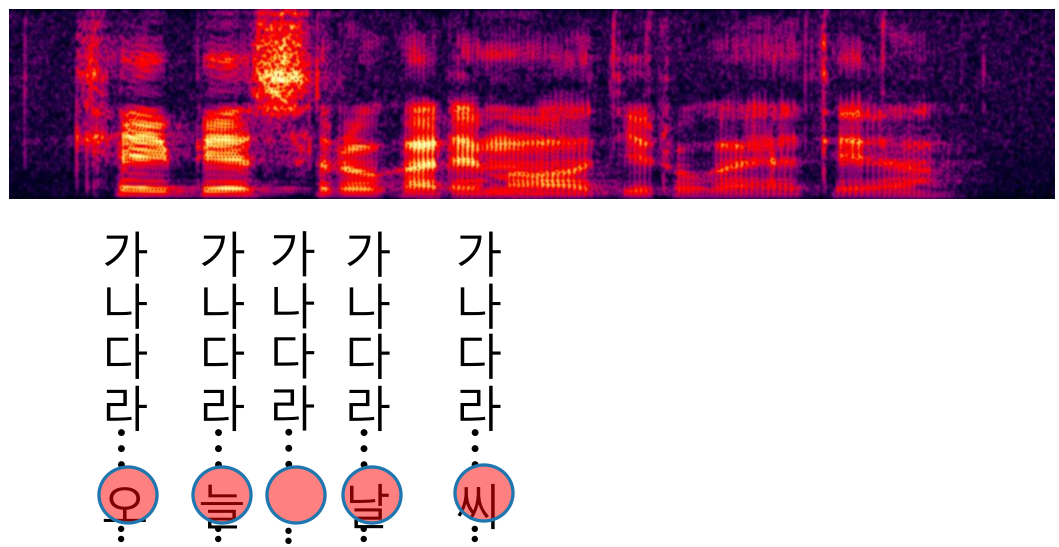
- 특징 추출 실습

- https://colab.research.google.com/github/pkyoung/a1003/blob/main/local/fx.ipynb

PART III:
트랜스포머 기반 종단형 음성인식 기술

확률모델 기반 음성인식

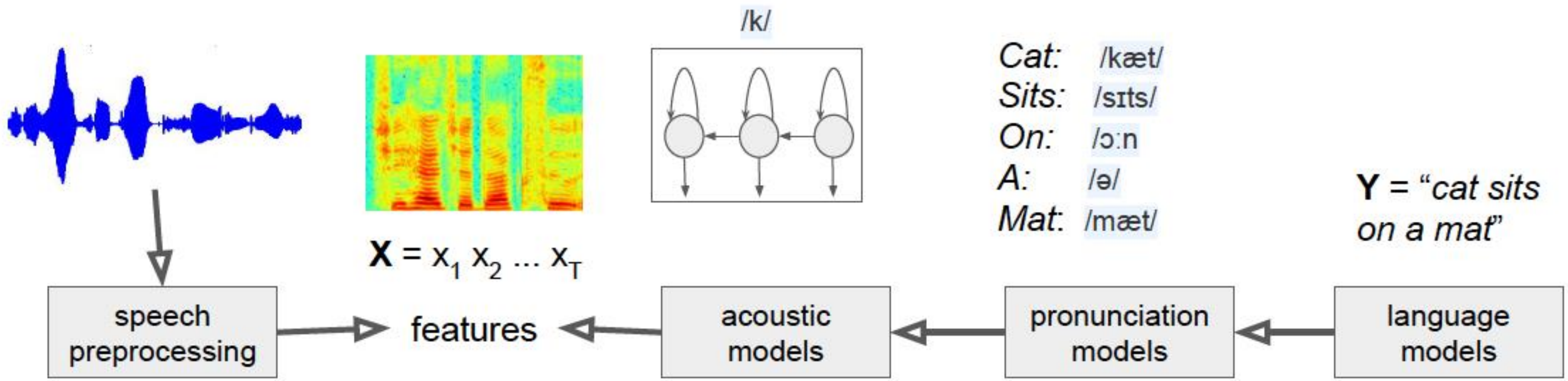
- $W* = \text{argmax } P(W|X)$
  - To Find Most Probable Word Sequence Given Input Signal/Feature



- Considerations
  - Boundary? Segmentation?
  - Output Units? Words, Characters, Phoneme, …
  - Classification Accuarcy? Unit Accuracy vs. Sentence Accuracy

How It really works

- $W* = \text{argmax} \log P(W|X) \rightarrow P(X|W)P(W)$
- $\quad = \text{argmax} \log P(X|Q)P(Q|W)P(W)$
- To Find Most Probable Sequence Among Plausible Words Sequences



https://heartbeat.fritz.ai/the-3-deep-learning-frameworks-for-end-to-end-speech-recognition-that-power-your-devices-37b891ddc380

Guess who?

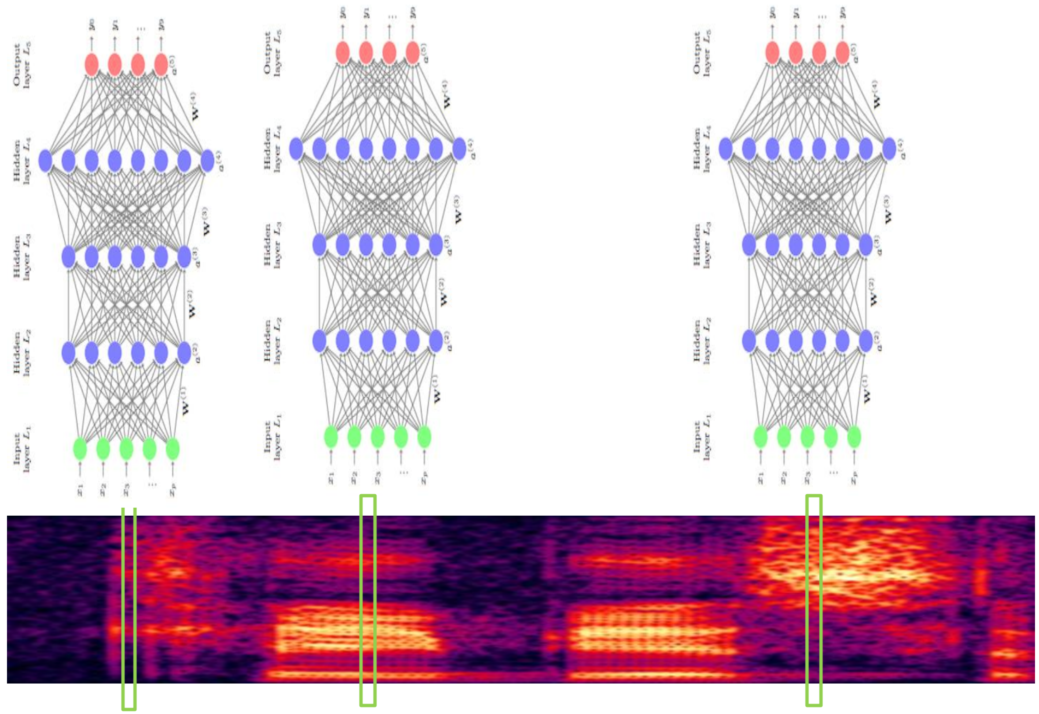
- Find A Criminal Among Suspects Given Evidence
- Criminal = argmax P(Suspect|Evidence)
- Criminal = argmax P(Evidence|Suspect)

  = argmax

  P(Evidence|Behavior)P(Behavior|Suspect)P(Suspect)

LVCSR with DNN: Concept

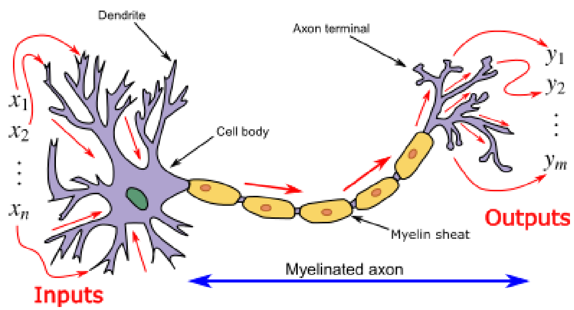
hhhheeellllllllllllllllllllooooo_____wwwooorrrrllllldddddd → hello world



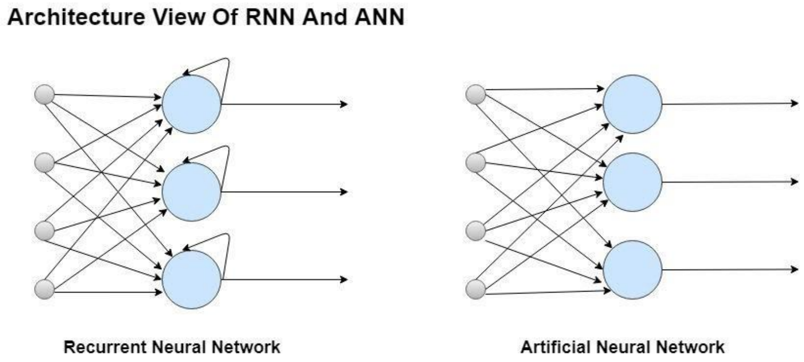
CTC: Connectionist Temporal Classification

RNN: Recurrent neural network

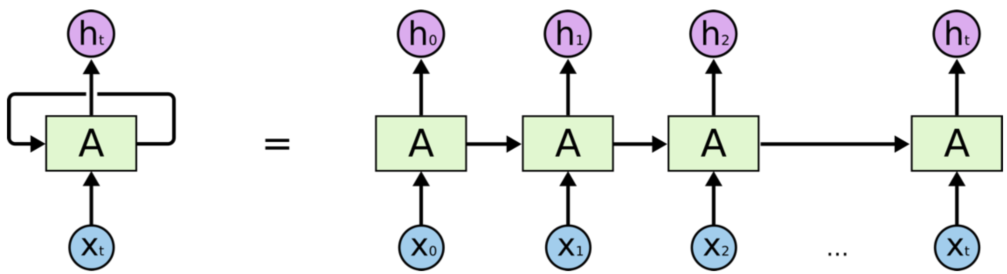
- Neural Networks
  - Mimic human brain: Neuron, Synapse



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



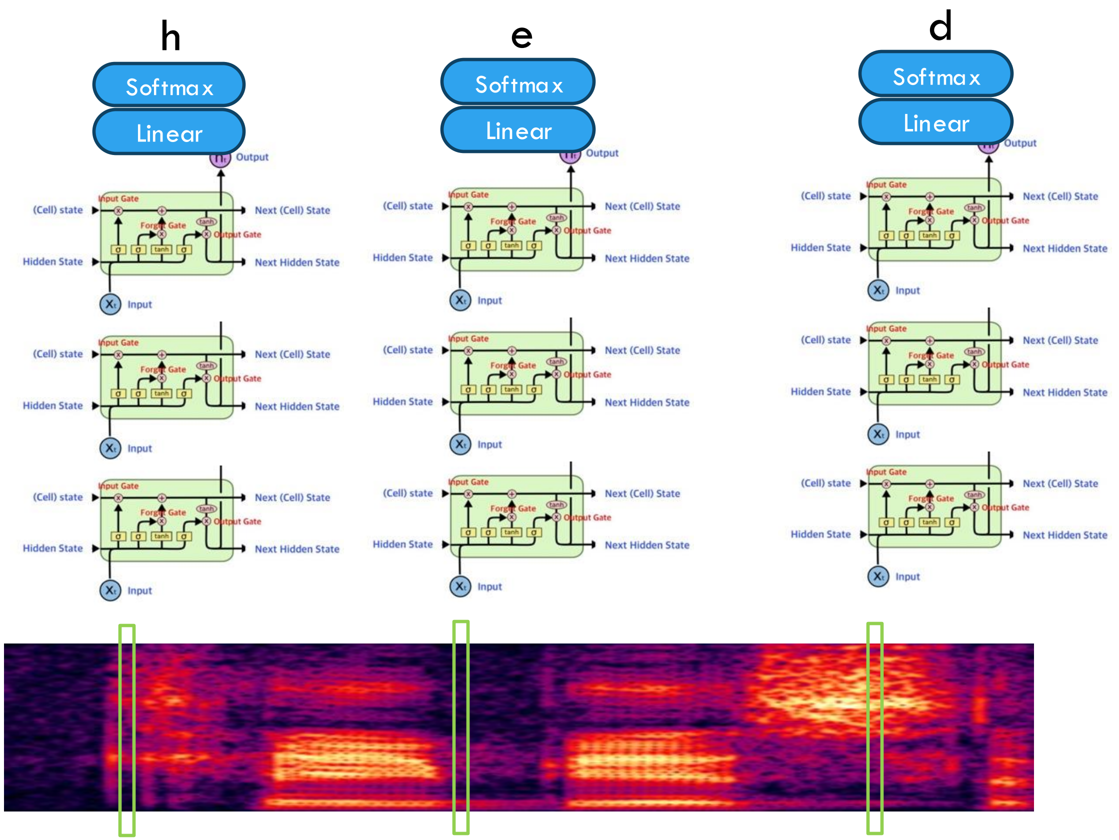
https://medium.com/datadriveninvestor/recurrent-neural-networks-in-deep-learning-part-1-df3c8c9198ba



http://colah.github.io/posts/2015-08-Understanding-LSTMs/

LVCSR with RNN: Concept

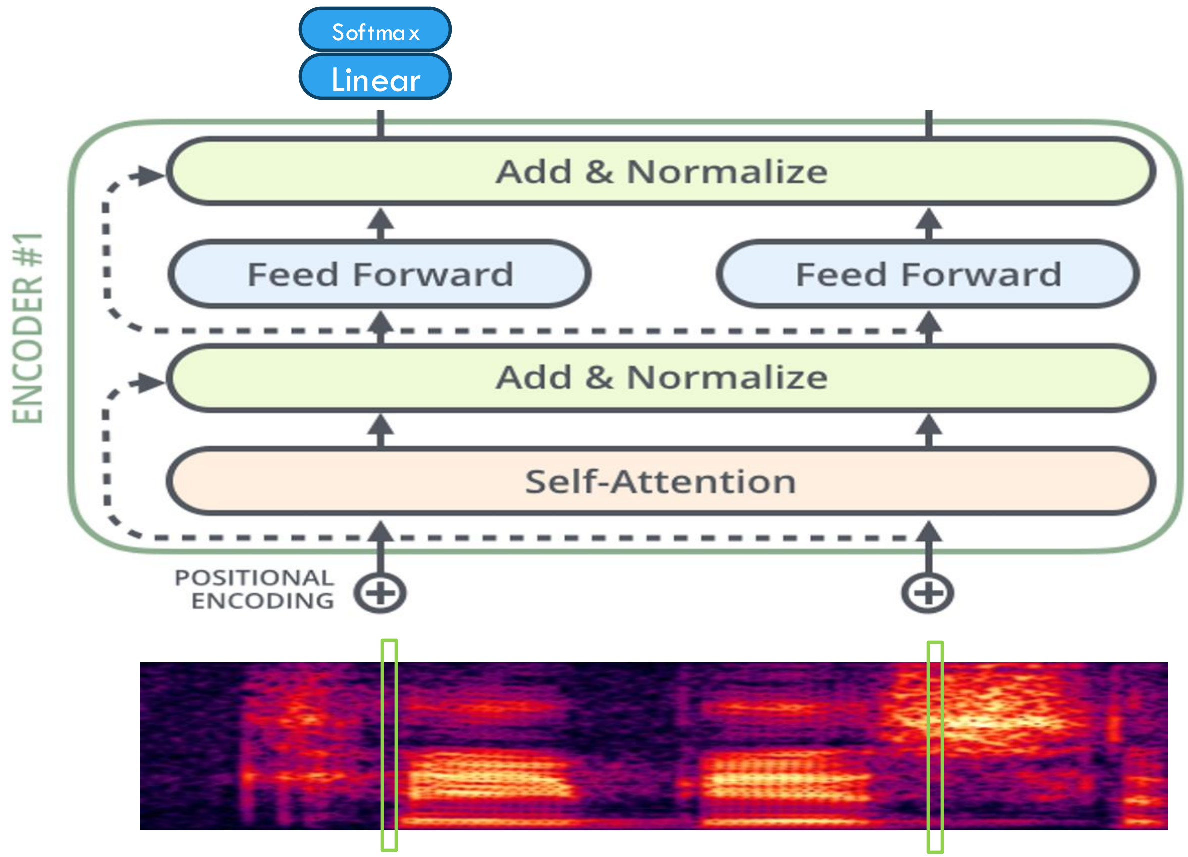
hhhheeellllllllllllllllllooooo____wwwooorrrrlllllddddd



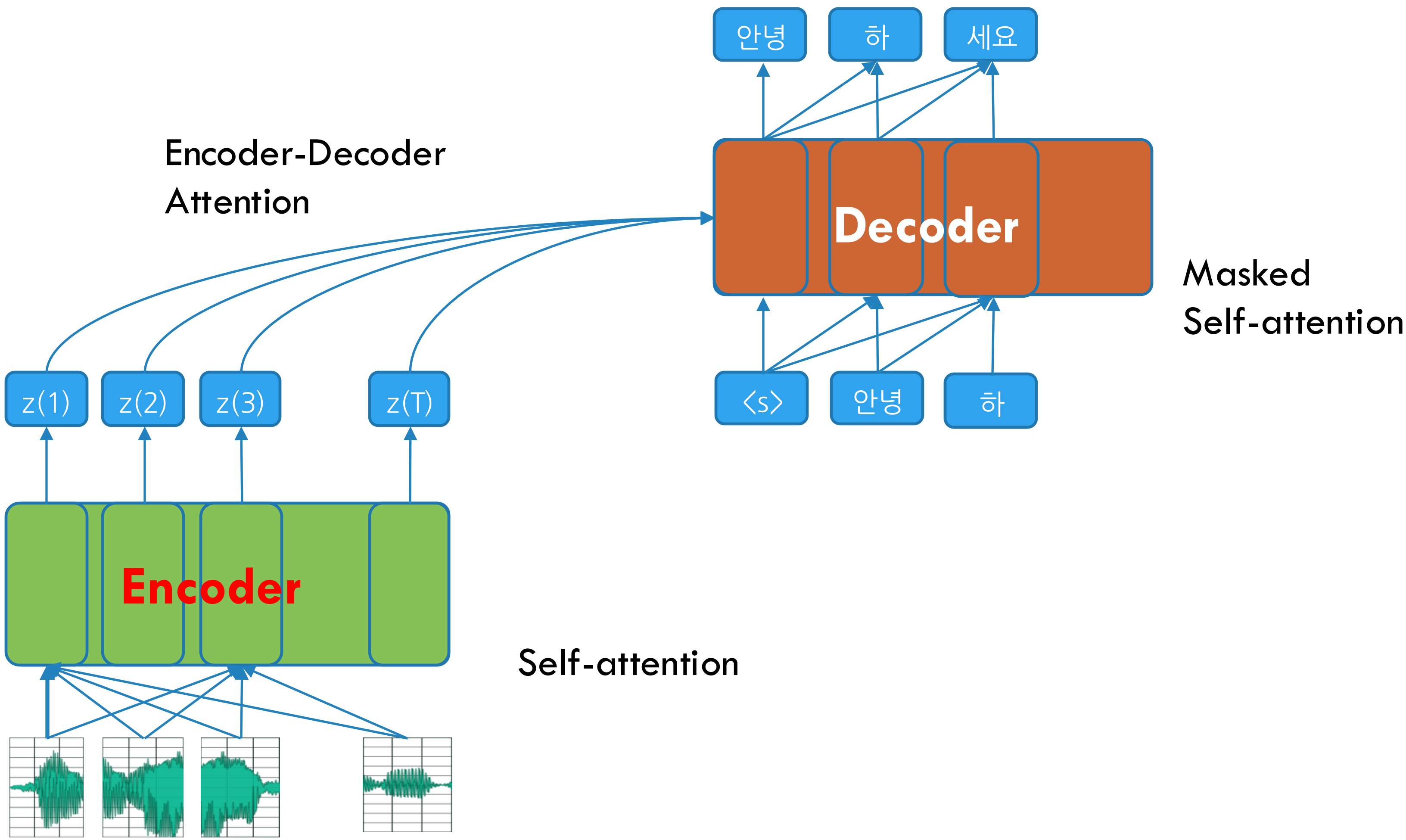
hello world

Transformer: Parallelize RNN

hhhheeelllllllllllllllllooooo____wwwooorrrrlllllddddd → hello world

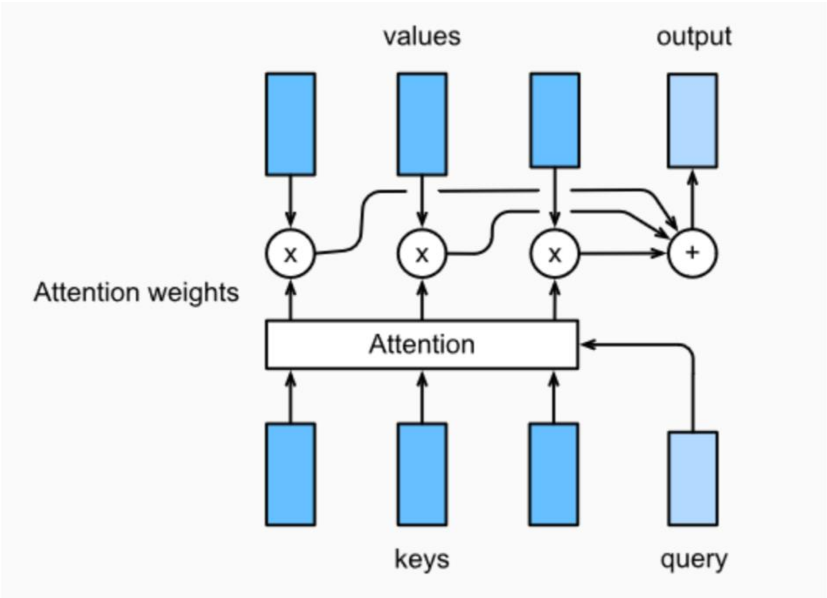


Encoder-Decoder



Attention

- Query, Key, Value
- Memory = Dictionary(Key,Value)
- Output = Weighted Sum of Value
- Weight = Similarity Between Query and Key



https://programming.vip/docs/5e4cadd75dc1d.html

Word Encoding

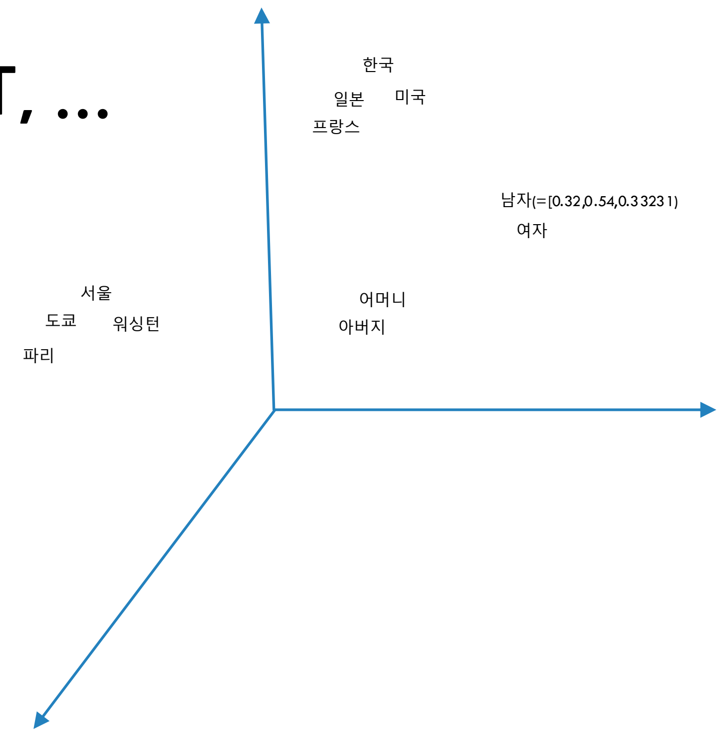
- Representation of a word as integer or vector
- Integer encoding vs one-hot vector encoding
- Sparse Representation vs. Dense Representation

raw_text="The sky turned red"
vocab = ["the", "sky", "turned", "red"]

Word	integer encoding	one-hot encoding
<pad>	0	[0, 0, 0, 0, 0]
<unk>	1	[1, 0, 0, 0, 0]
the	2	[0, 1, 0, 0, 0]
sky	3	[0, 0, 1, 0, 0]
turned	4	[0, 0, 0, 1, 0]
red	5	[0, 0, 0, 0, 1]

Word Embedding

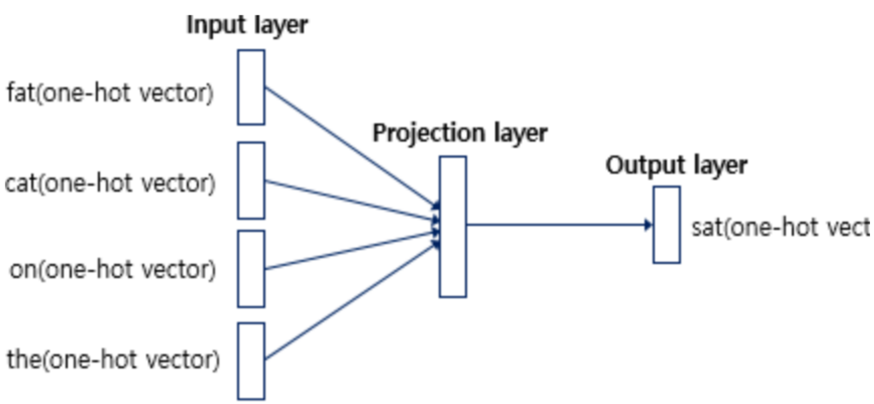
- Dense representation
  - One-hot: Sparse representation
  - Word2vec, GloVe, FastText, BERT, …
  - Dimension reduction
- Capture semantic relationship
  - 한국 - 서울 + 파리 = 프랑스
  - 어머니 - 아버지 + 여자 = 남자
  - 아버지 + 여자 = 어머니
  - Ehance generalization performance

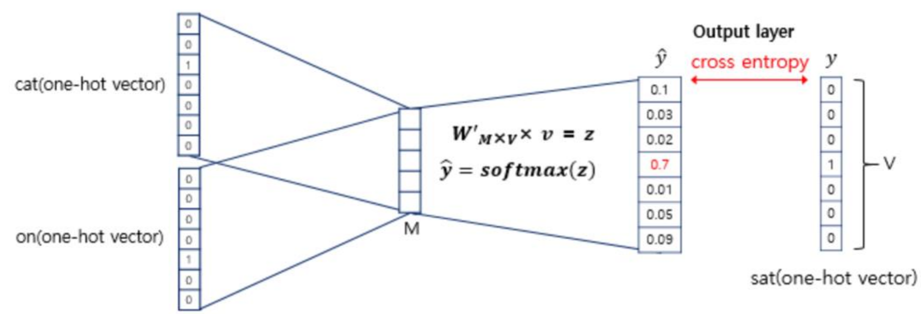


Word2Vec: CBOW

- Efficient Estimation of Word Representations in Vector Space, 2013, Tomas Mikolov, et al.
- CBOW: Continous Bag of Words
- No-nonlinearity in projection layer

예문 : "The fat cat sat on the mat"





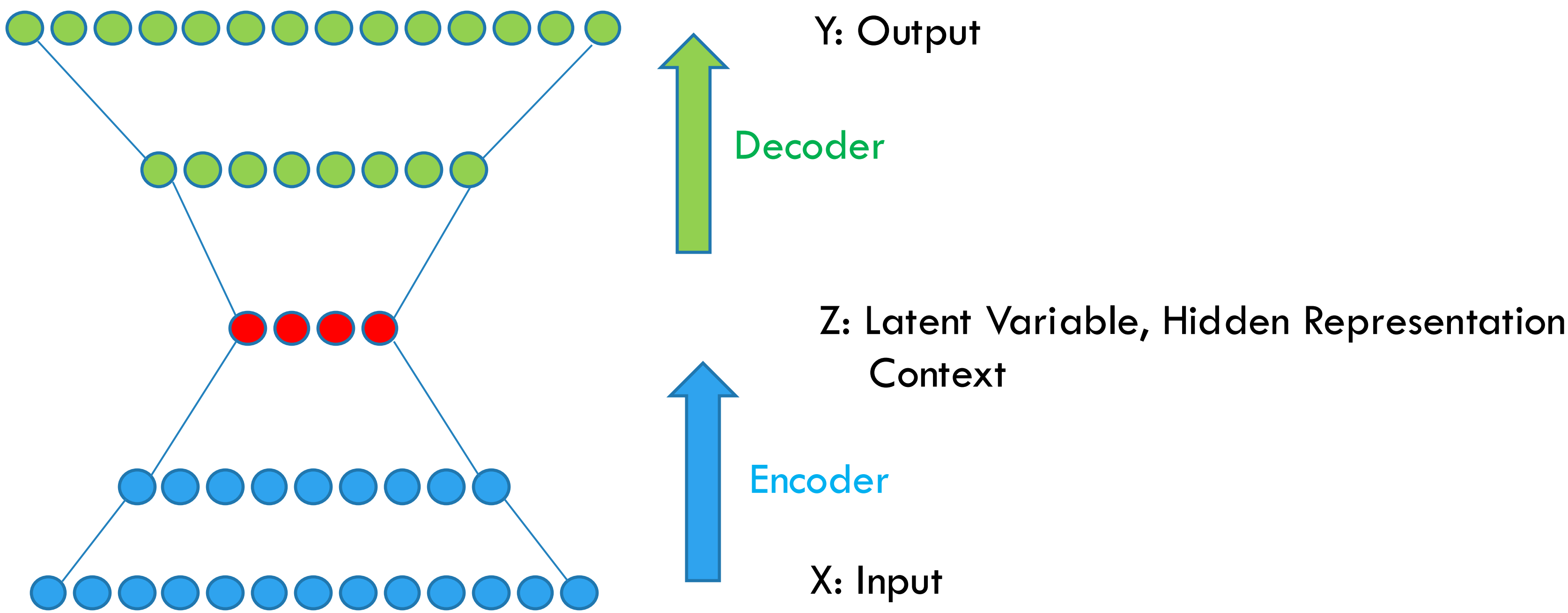
Tokenizing

- Token?
  - Basic unit of input and output
  - Char vs Word vs Subword (아버지?)
- Subword tokenizing: word → subword
  - To solve OOV problem
- Byte Pair Encoding
  - Neural Machine Translation of Rare Words with Subword Units, ACL, 2016
  - Not encoding but tokenization
  - A New Algorithm for Data Compression, 1994, C Users Journal

aaabdaaabac	→ ZabdZabac Z=aa	→ ZYdZYac Y=ab Z=aa	→ XdXac X=ZY Y=ab Z=aa

Encoder-Decoder

- Auto Encoder



Encoder-Decoder for Sequence

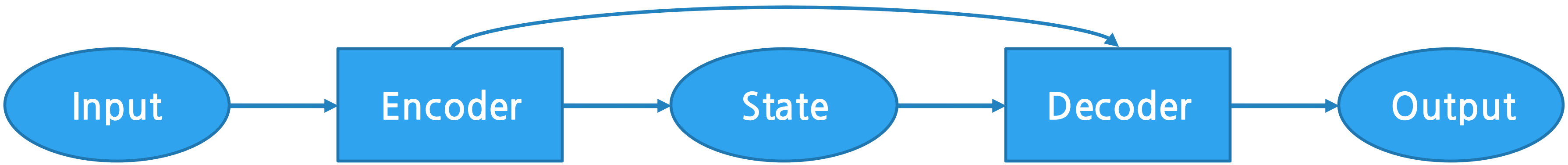


Image
Video
Sentence
Speech

Sequence Encoder
- RNN, LSTM
- Transformer

Sequence Decoder
- RNN, LSTM
- Transformer

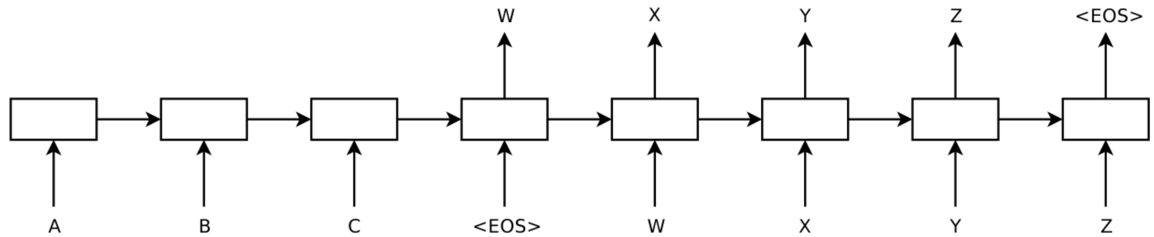
Class Index
Sentence
Image
Video
Speech

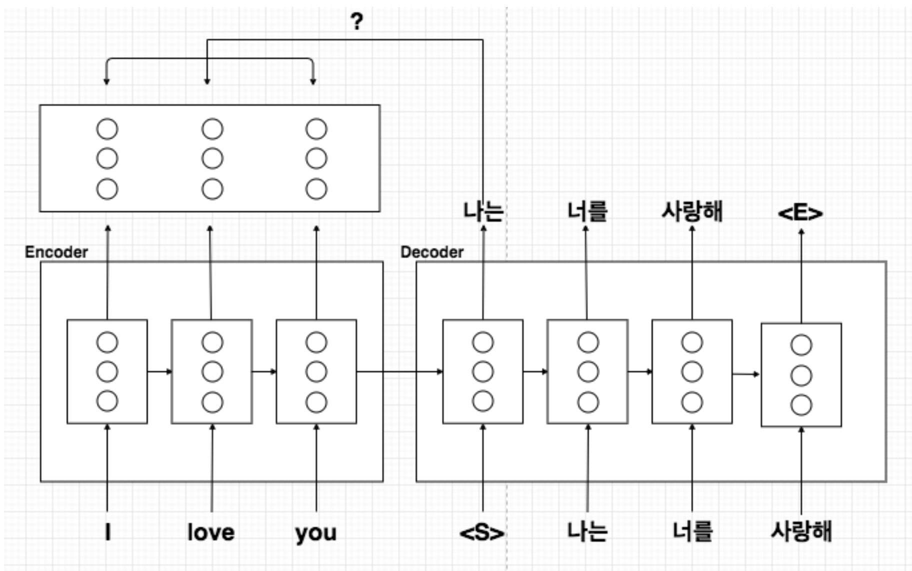
- Translation
- Image/Video Captioning
- Q&A, Document Summarization
- Speech
  - Recognition, Synthesis, Translation, Dialog System(Google Duplex, 2018)

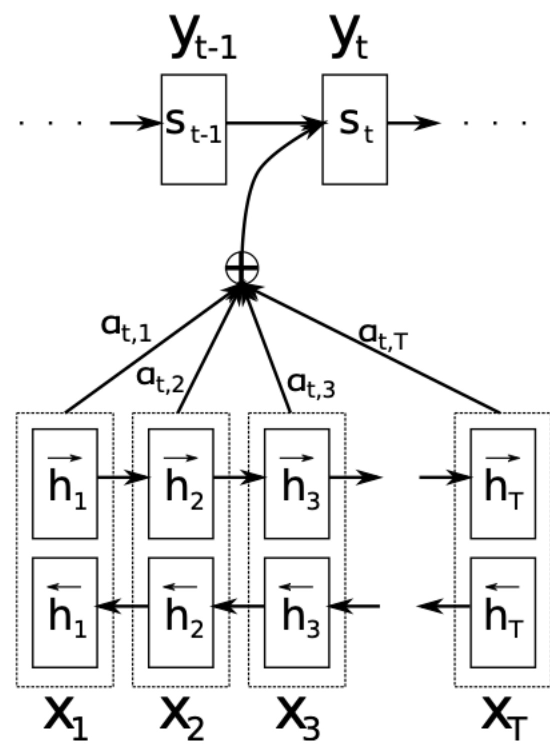
Era of Sequence-to-Sequence

- Natural Language Processing
- Sequence to Sequence Leanring with Neural Networks, NeurIPS, 2015
- Neural Machine Translation By Jointly Learning To Align And Translate, ICLR, 2016
- Attention Is All You Need, NuerIPS, 2017
- BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, ACL, 2019

Sequence to Sequence with Attention





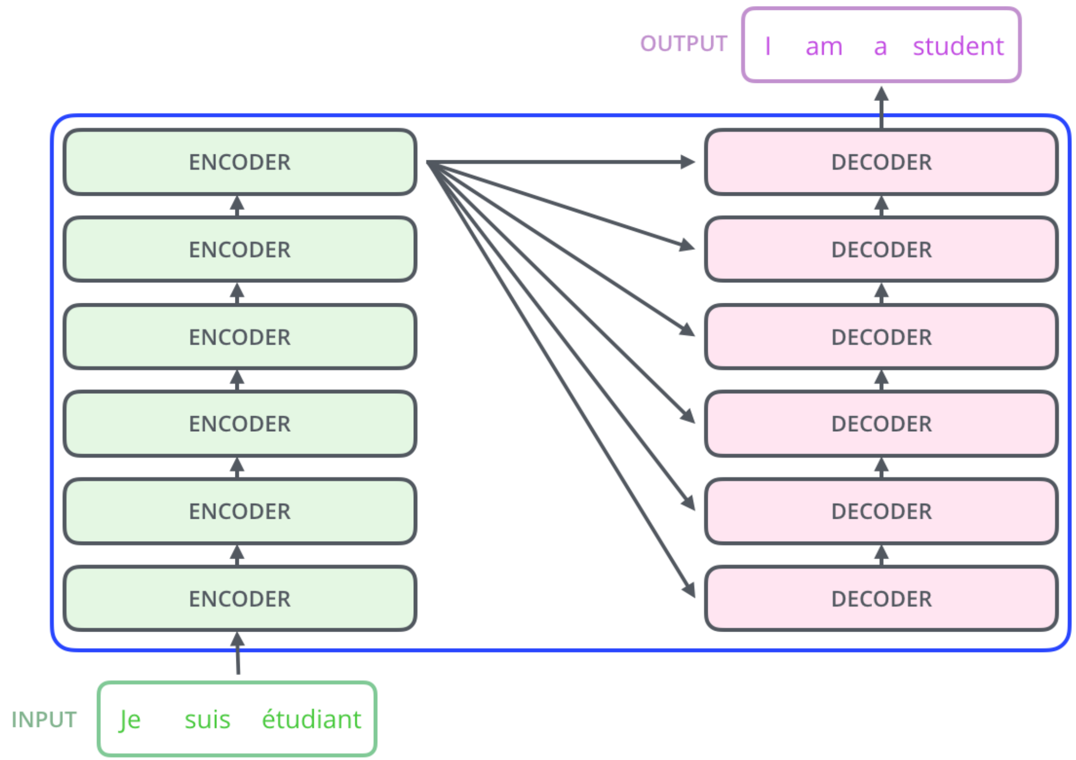


https://medium.com/platfarm어텐션-메커니즘과-transfomer-self-attention-842498fd3225

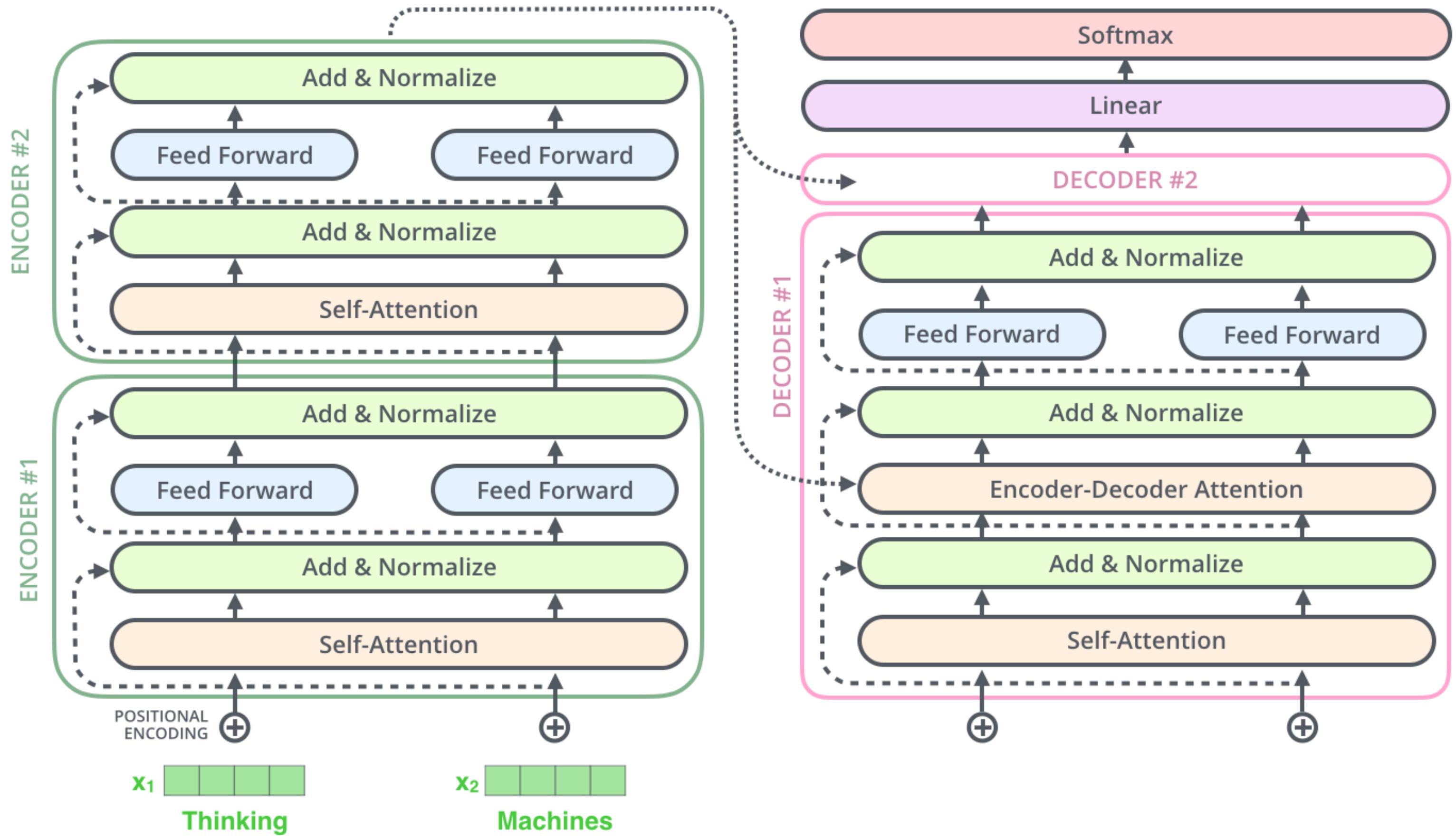
Transformer

Transformer: Overall Structure

- https://jalammar.github.io/illustrated-transformer/

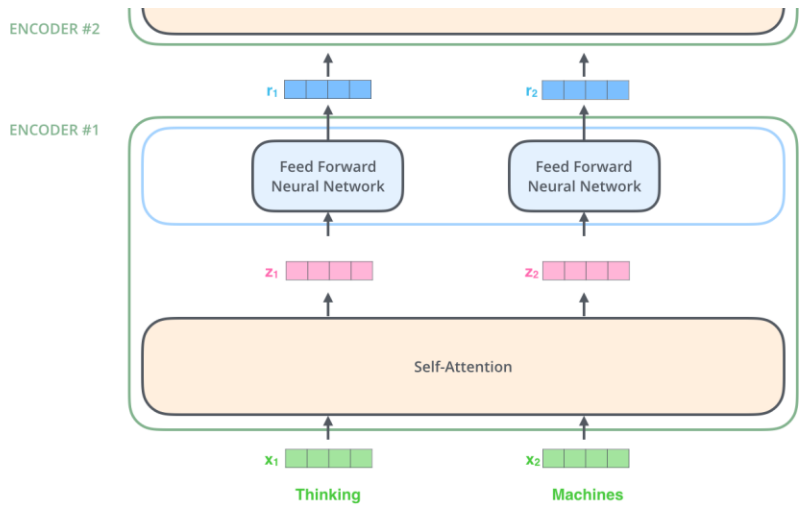


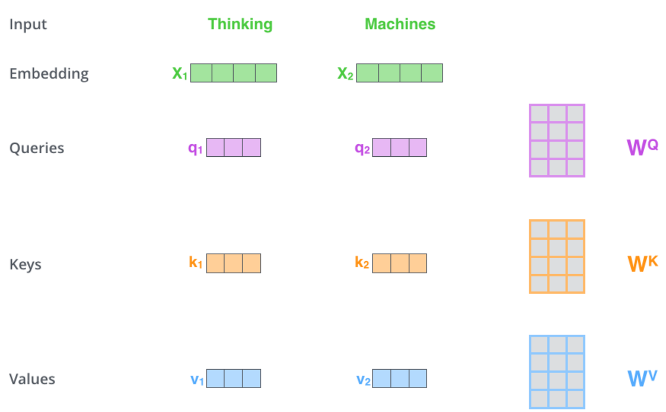
Transfomer: Detailed Structure

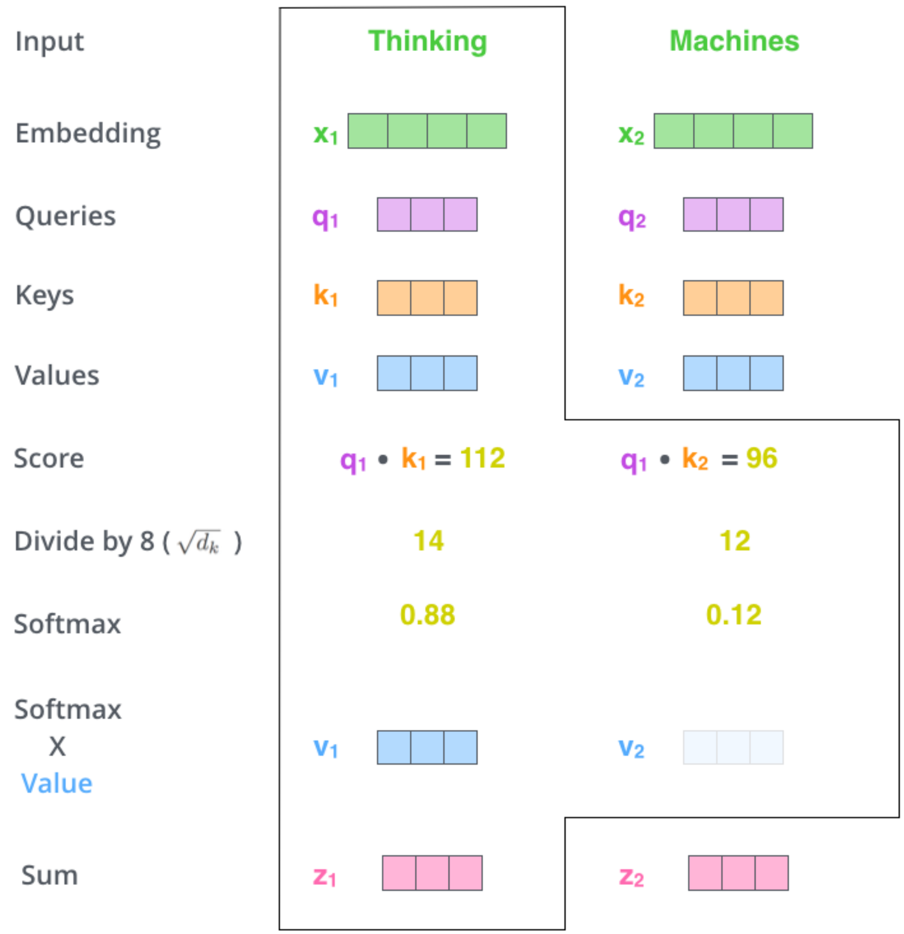


https://jalammar.github.io/illustrated-transformer/

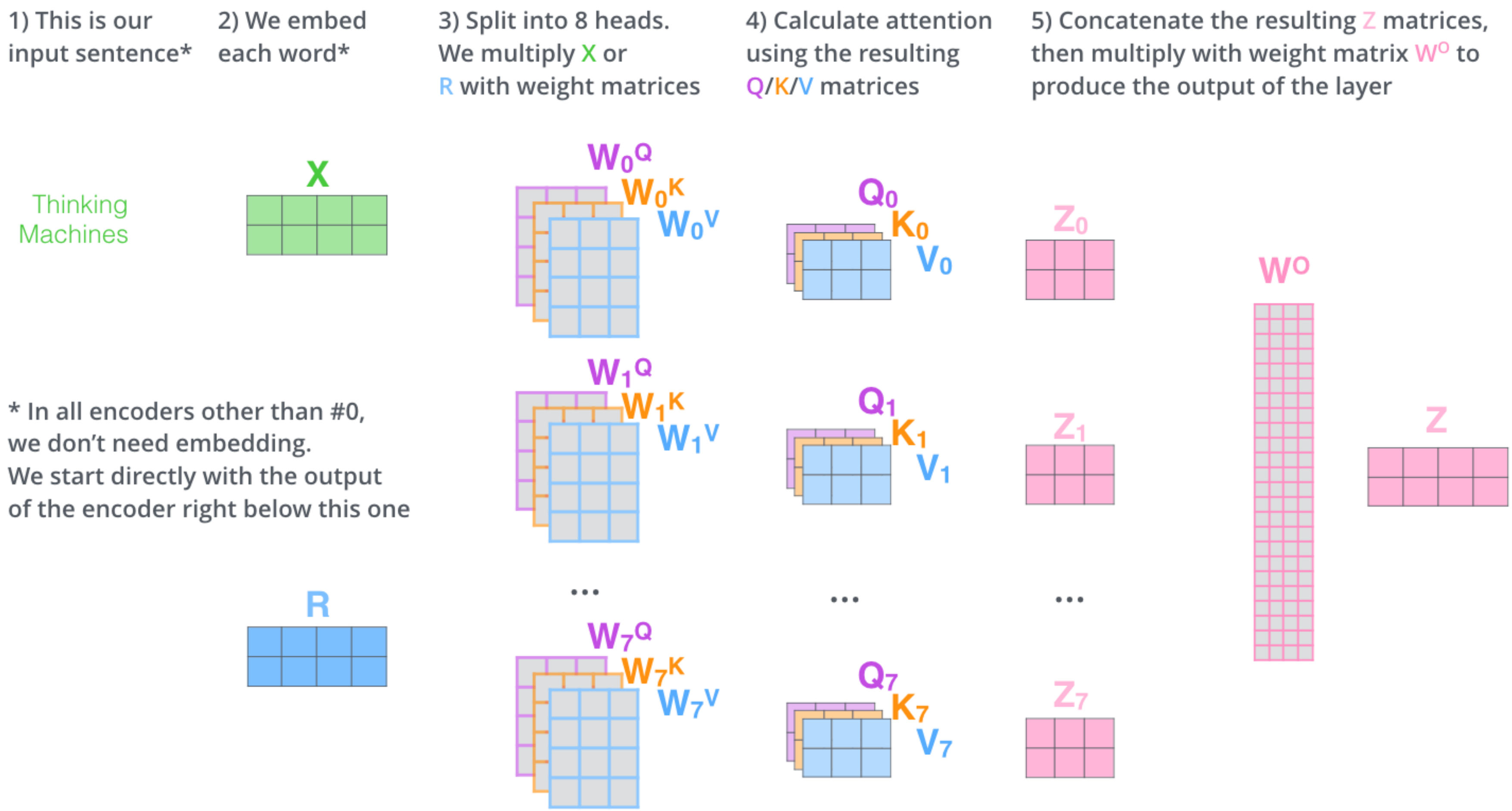
Self Attention







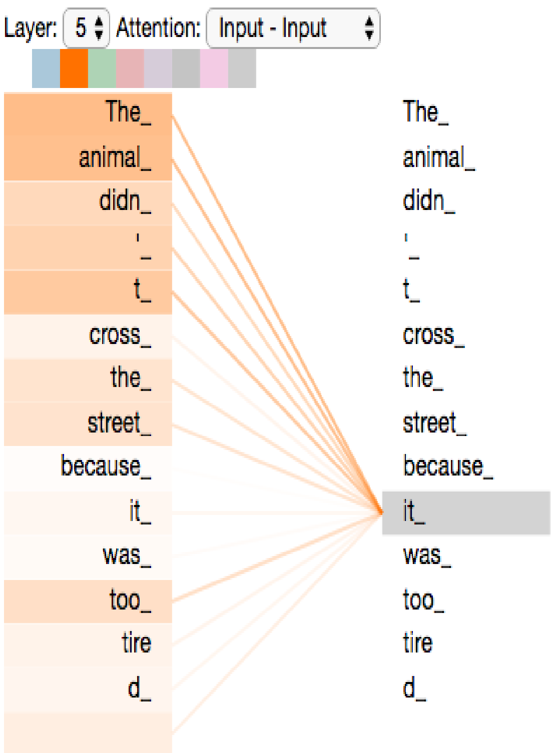
Multihead attention

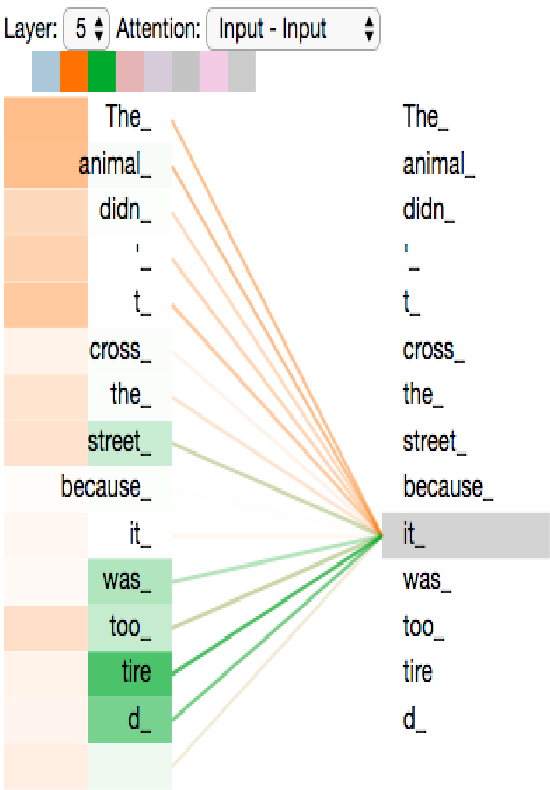


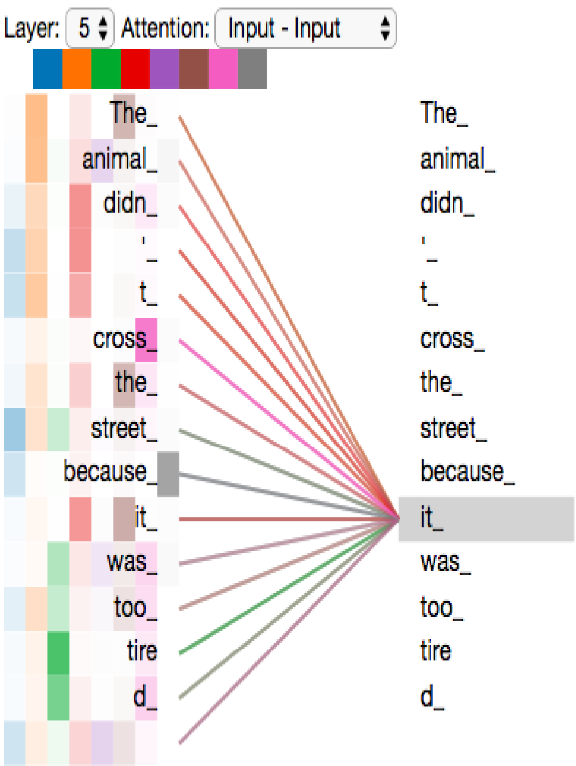
https://jalammar.github.io/illustrated-transformer/

Effect of Self Attention

The animal didn't cross the street because it was too tired

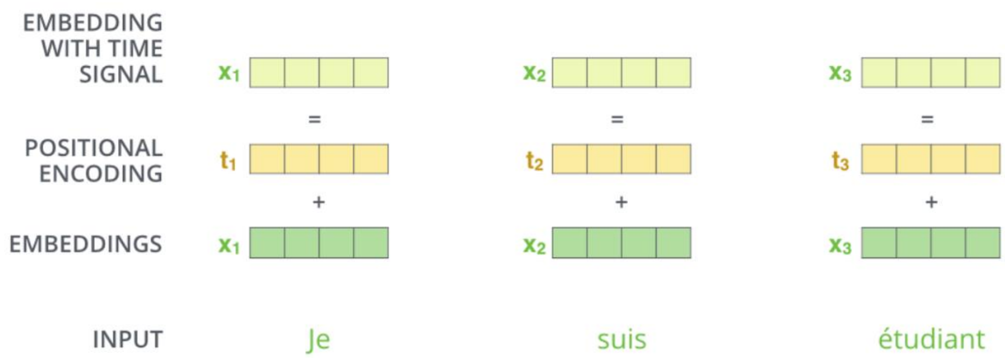






Positional Encoding

- No Position Depenent Computation in Transfomer



0 :	0	0	0	0
1 :	0	0	0	1
2 :	0	0	1	0
3 :	0	0	1	1
4 :	0	1	0	0
5 :	0	1	0	1
6 :	0	1	1	0
7 :	0	1	1	1

- Absolute/Relative Position Encoding
  - Sinusoidal Positional Encoding

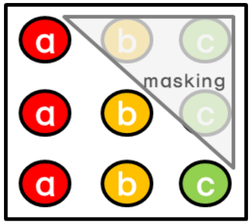
$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$

$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

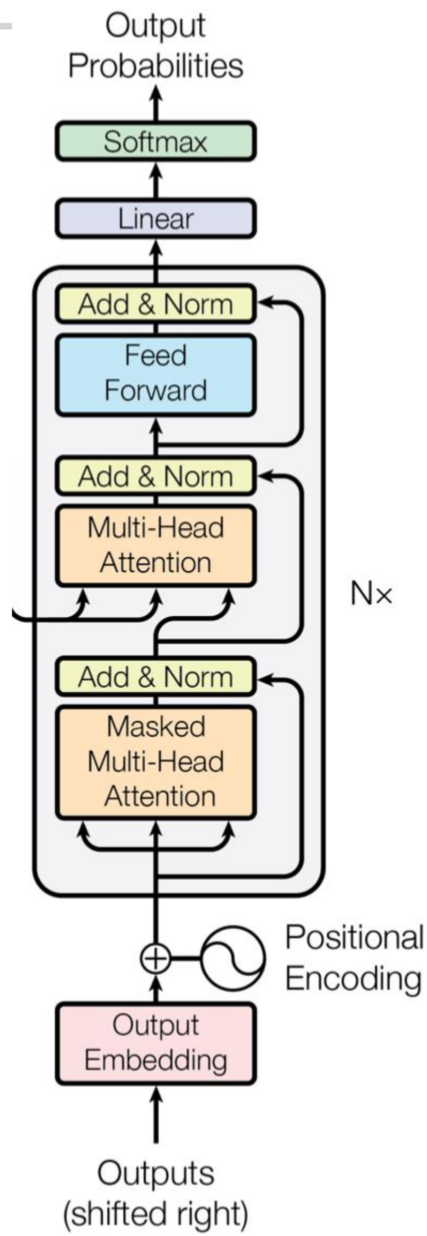
https://kazemnejad.com/blog/transformer_architecture_positional_encoding/

Decoder

- Masked Multi-Head Self Attention

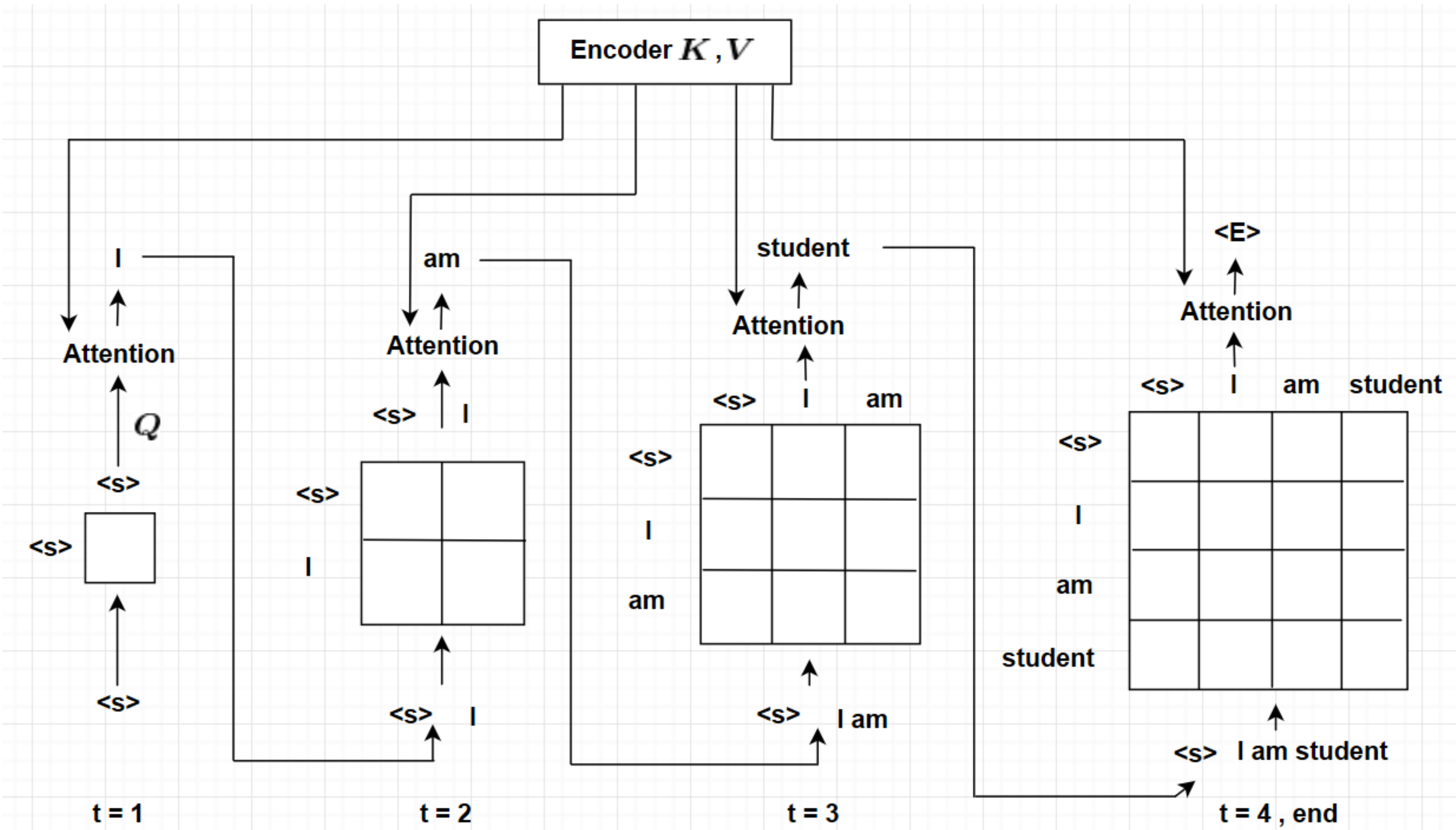


- Encoder-Decoder Attention
  - K, V from Encoder Last Layer
  - Q from Self Attention
- Beam Search



Attention Is All You Need, NuerIPS, 2017

Decoder in action



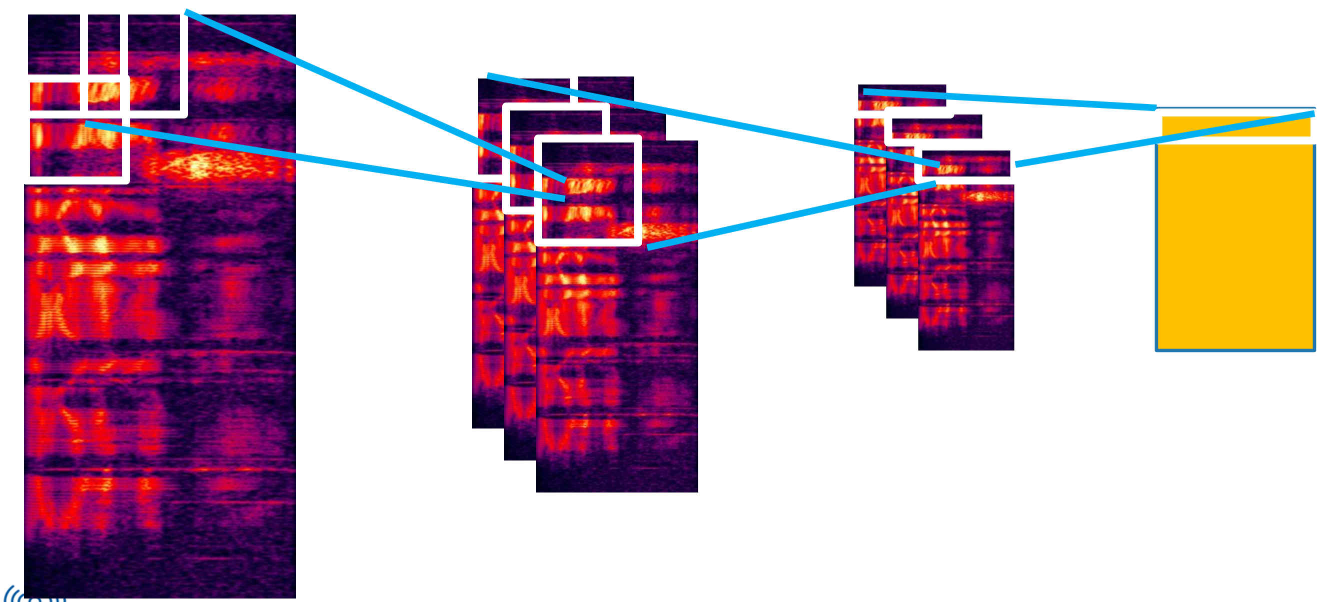
https://medium.com/platfarm/어텐션-메커니즘과-transfomer-self-attention-842498fd3225

End-to-end For ASR

- ESPNet: End-to-end Speech Processing Toolkit
  - ASR, TTS, Speech Translation
  - https://github.com/espnet/espnet
- CTC Hybrid*: Connectionist Temporal Classification
  - Multi-task training with CTC Criteiron
  - Increase Stability while Training
  - Hybrid CTC/Attention Architecture for End-to-End Speech Recognition, IEEE Journal of Selected Topics in Signal Processing, 2018
- Input Embedding

Input embedding

- TEXT: Input = Word: One-hot → Vector
- ASR: Input = MELFB: Vector → Vector
- 2x Conv2d layer, 3x3 kernel with stride=2
  - TxF → adim x T/2 x F/2 → adim x T/4 x F/4 → T/4 x adim



End-to-End ASR In Practice

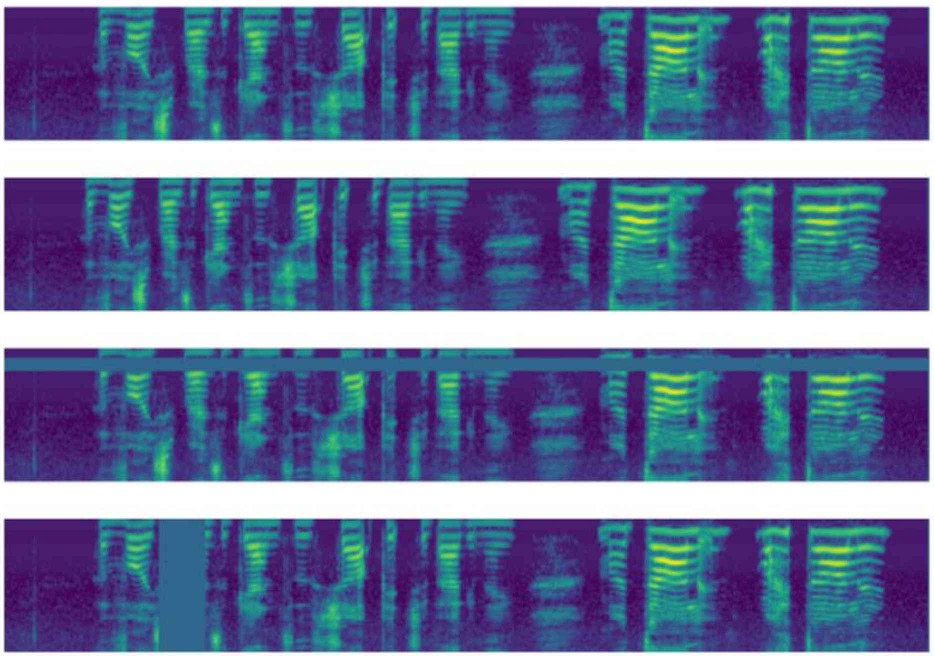
- Output Units
  - 영어: Alphabet, BPE(Byte Pair Encoding), Word
  - 한국어: Char(음절~2500), BPE(~5000), 형태소분석기
- Relative Performance
  - WER/CER
  - 25% (GMM-HMM) → 15% (DNN-HMM) → 10% (LSTM-HMM)
  - 7% Transformer
- Limitation
  - Process Whole Sentence → Streaming ASR

성능개선방안

- 데이터!
  - 실환경 데이터수집: 적응훈련/연결학습
  - 음향모델/언어모델?
- 데이터!!
  - 데이터 증강
  - SpecAug, Speed/Volume perturbation, Noise addition, Simulated data
- 모델 파라미터
  - Number of epoch
  - Number of paramerters: layers, dimension etc
  - Gradient scale: batchsize, learning rate etc
  - Robustness: dropout rate,

SpecAug

- SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition (2019)
- https://ai.googleblog.com/2019/04/specaugment-new-data-augmentation.html



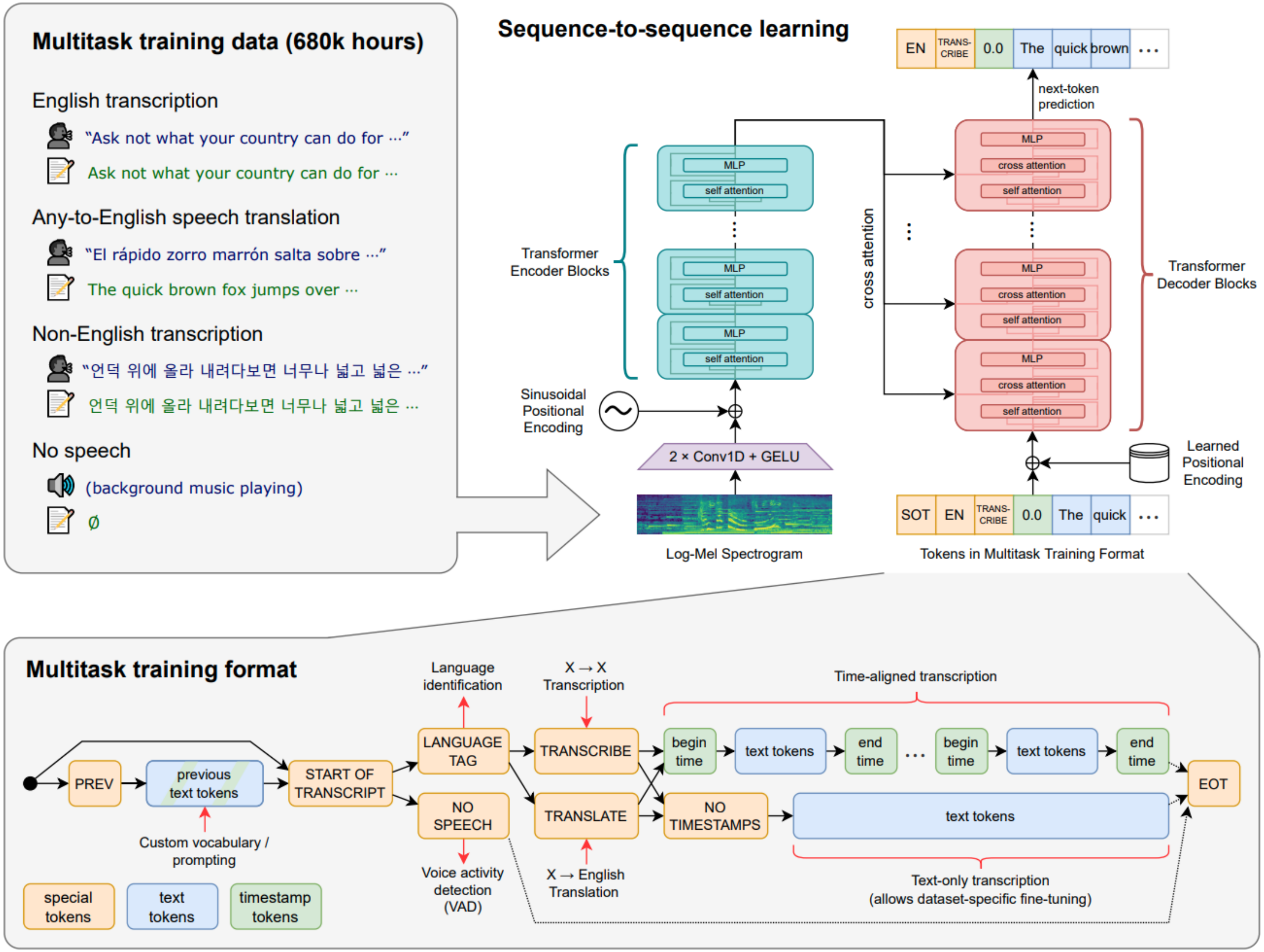
OpenAI Whisper

- Transformer-based Encoder-Decoder Model
  - No CTC
  - Multi-task Training
  - Model Card

Size	Parameters	English-only model	Multilingual model
tiny	39 M	✓	✓
base	74 M	✓	✓
small	244 M	✓	✓
medium	769 M	✓	✓
large	1550 M		✓
turbo	798 M		✓

  - Release
    - September 2022 (original series), December 2022 (large-v2), November 2023 (large-v3), September 2024 (large-v3-turbo)

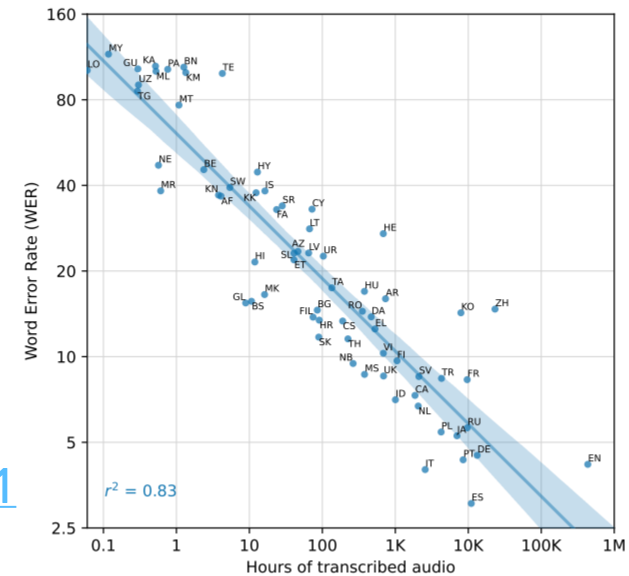
Whisper Model Archtecture



Whisper: Training Data

- large-v2(22.9)
  - 680k hours of audio
  - Weakly labelled

- large-v2 (22.12)
  - 2.5x more epochs with
    added regularization for improved performance
  - https://github.com/openai/whisper/discussions/661
- large-v3(23.11)
  - https://huggingface.co/openai/whisper-large-v3
  - 1M hours of weakly labeled audio
  - 4M hours of pseudo-labeled audio
- large-v3-turbo(24.11)
  - https://github.com/openai/whisper/discussions/2363
  - decoding layers have reduced from 32 to 4



Whisper as Foundation Model

- Finetuning Whisper
  - https://huggingface.co/blog/fine-tune-whisper
- Multimodal Encoder
  - Text: BERT
  - Video: ResNet
  - Audio:
    - MFCC, wav2vec, HuBERT
    - Whisper Encoder
- Decoder
  - Whisper Decoder, LLM

실습

- ESPnet 기반 음성인식
  - https://colab.research.google.com/github/pkyoung/a1003/blob/main/local/espnet.ipynb

- OpenAI Whisper 기반 음성인식
  - https://colab.research.google.com/github/pkyoung/a1003/blob/main/local/whisper.ipynb

Discussion and Q&A