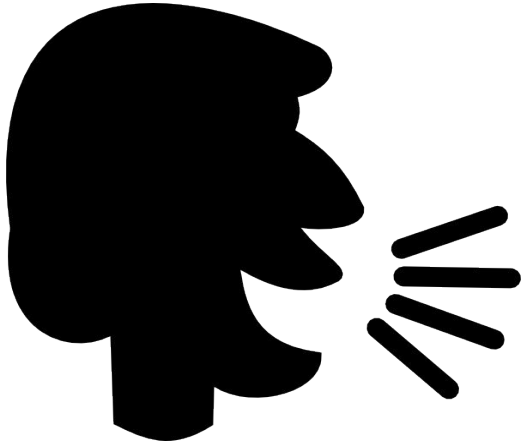


2019 음성인식 해커톤

Chapter 1. Feature for Deep Learning

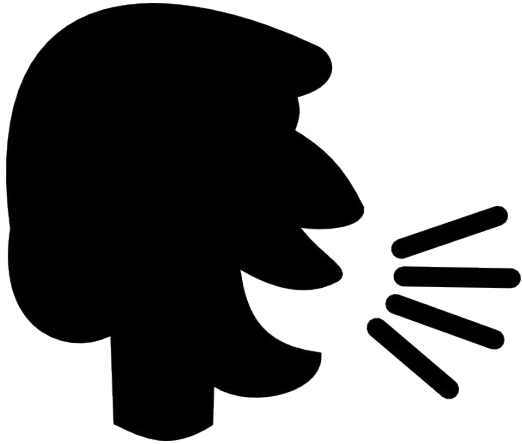
음성 인식?



안녕하세요, 오늘 날씨는 어떨까요?

사람이 발성하는 언어를 문자로 변환

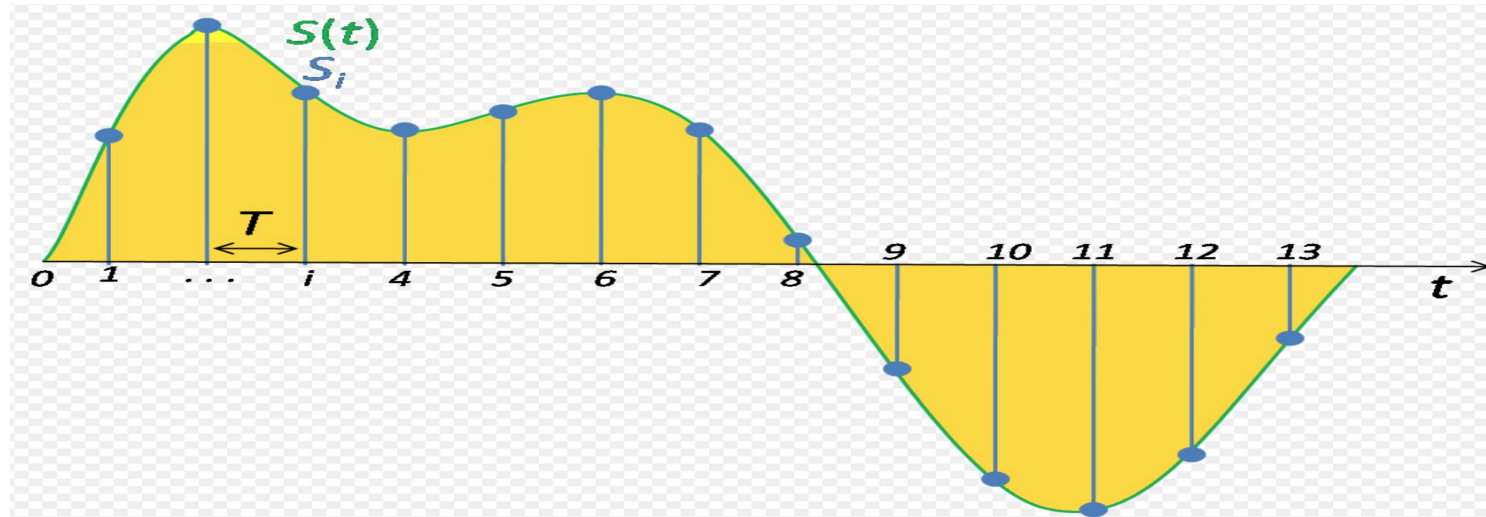
음성
신호란?



Speech Signal Waveform

어떻게 음성인식에 사용할 수 있을까요?

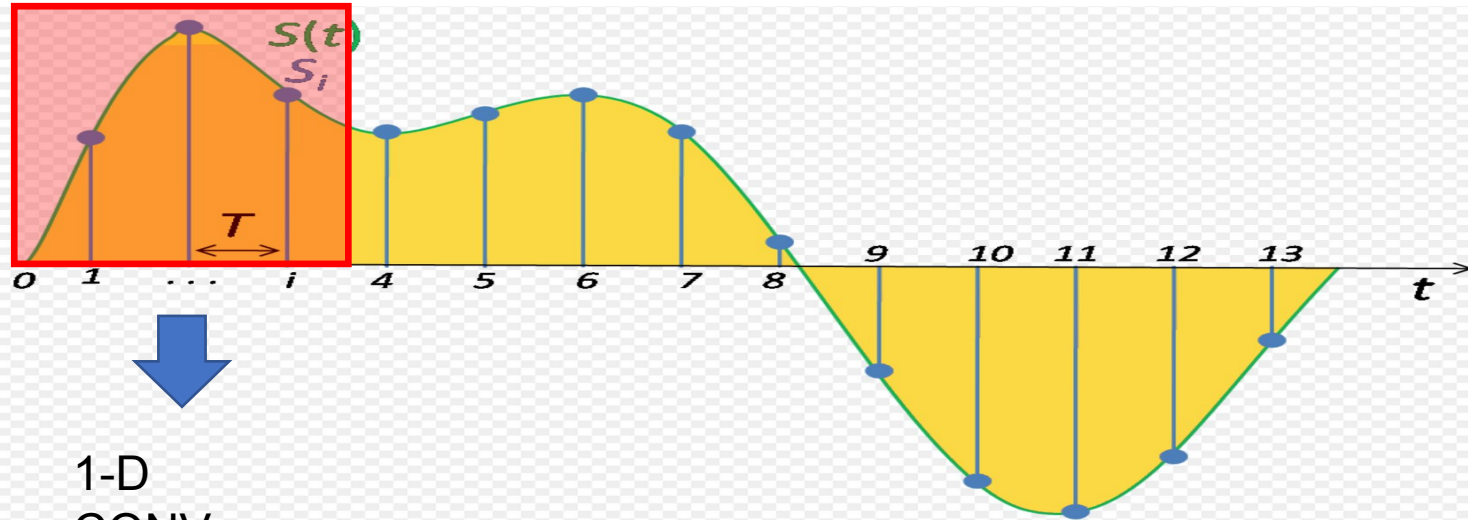
Analog to Digital (Sampling)



[29 18 11 1 -19 -29 -12 ... -10 -9 -12 -11 -12 -12 -13]

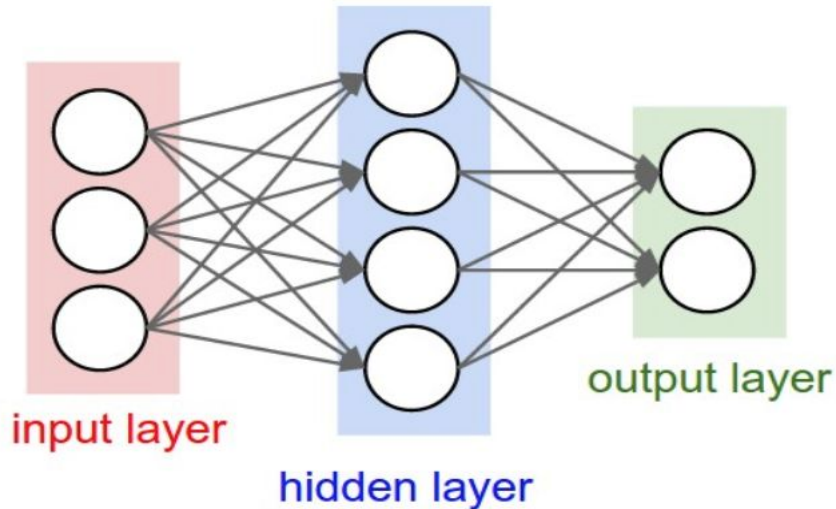
- sampling-rate N = 초당 N 개의 short array
- 일반적으로 음성인식에서는 16kHz sampling(또는 8kHz)을 사용

Deep Learning with **Raw** Waveform



1-D
CONV.

0.0001
0.0002
0.0003
...
0.0005
0.0001
0.0005



“안녕하세요”

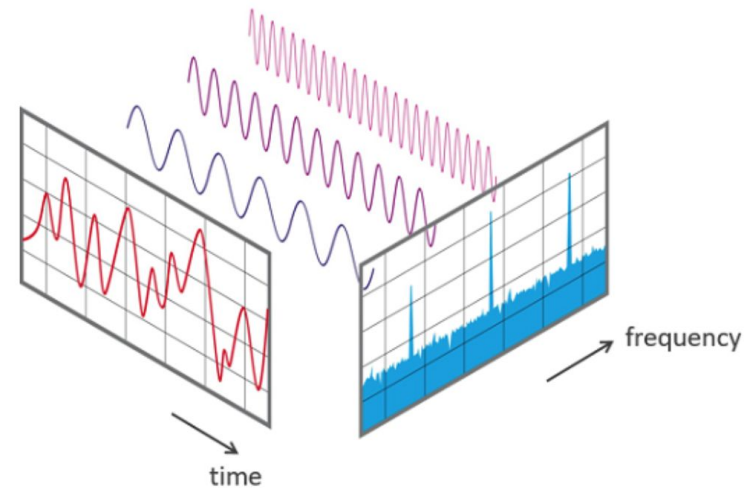
Time-Frequency Analysis (Fourier Transform)

음성의 특징 – 시간, 주파수 영역 분석 가능함

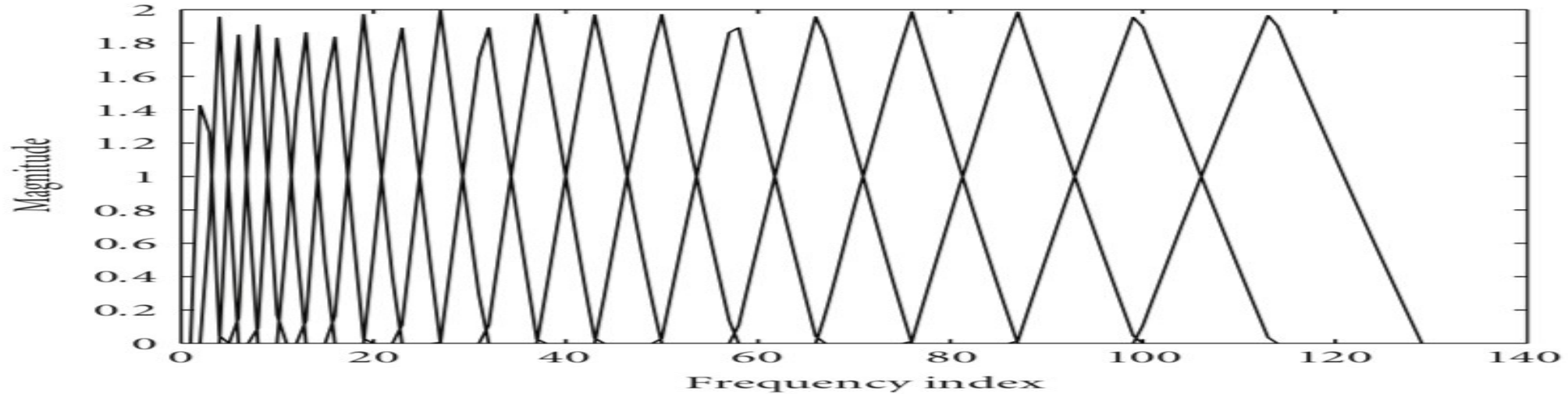
- Feature Embedding to T-F domain

Short-Time Fourier Transform (STFT)

- 일정 길이의 **sample**을 주파수 영역으로 변환



Deep Learning with



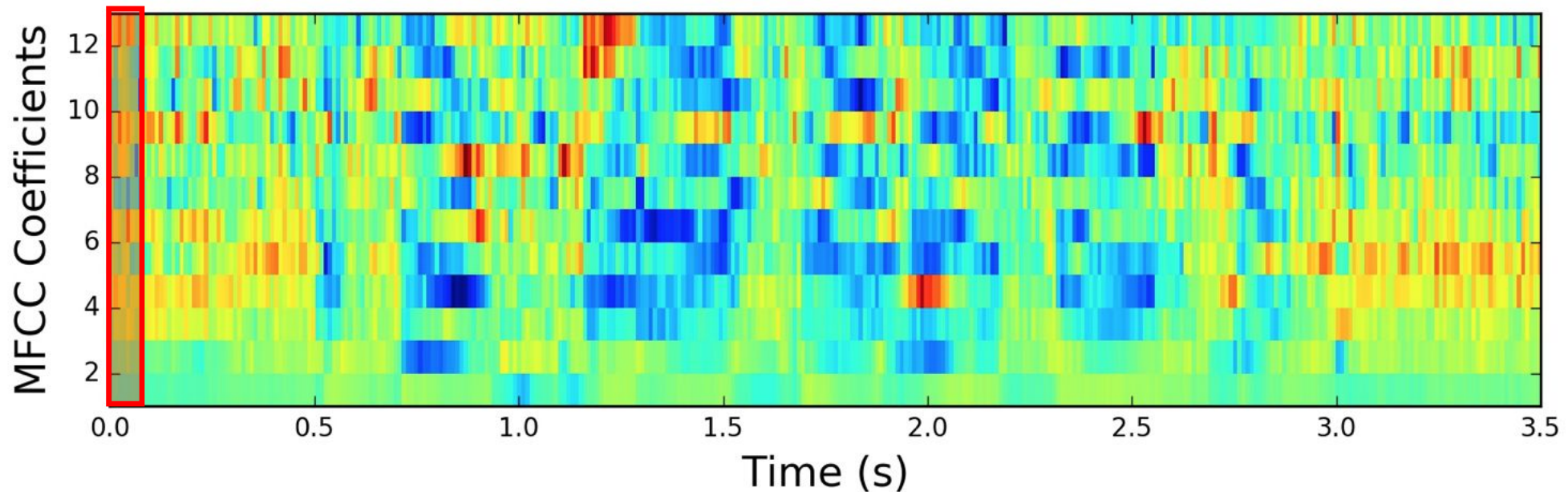
Mel-Scale : 인간의 청각 민감성에 상응하는 scale

Filter-bank : Mel-scale 상 band별 grouping

□ 주요 feature인 MFCC, mel-spectrogram에 사용됨

<https://librosa.github.io/librosa/generated/librosa.filters.mel.html>

Deep Learning with MFCC



MFCC : Mel-Frequency Cepstral Coefficients

Mel-filterbank \square Log(.) \square DCT \square MFCC

Librosa 사용가능

– 주요 Parameters : `sampling frequency`, `n_mfcc` (MFCC 차수)

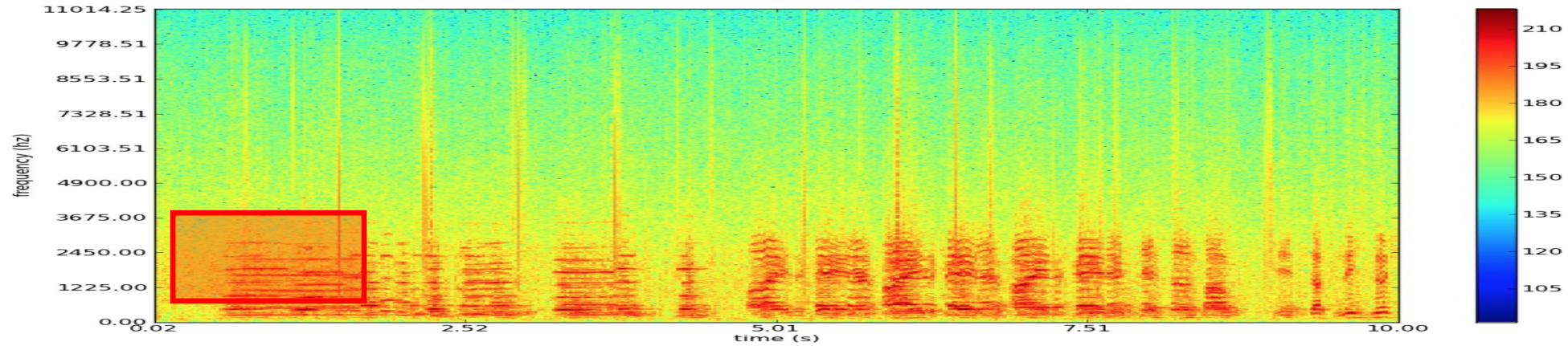
<https://librosa.github.io/librosa/generated/librosa.feature.mfcc.html>

Deep Learning with MFCC

Sample Codes

```
35 def get_feature_from_librosa(filepath, feature_size):
36     global first
37     global sig
38     global sample_rate
39
40     sample_rate = 16000
41     hop_length = 128
42
43     sig, sample_rate = librosa.core.load(filepath, sample_rate)
44
45     assert sample_rate == 16000, '%s sample rate must be 16000 but sample-rate is %d' % (filepath, rate)
46
47     mfcc_feat = librosa.feature.mfcc(y=sig, sr=sample_rate, hop_length=hop_length, n_mfcc=feature_size, n_fft=512)
48     mfcc_feat = torch.FloatTensor(mfcc_feat).transpose(0, 1)
49
50     return mfcc_feat
51
```

Deep Learning with Spectrogram



STFT 결과를 TF 2D plotting

Speech Signal을 Image처럼 취급 가능

Librosa 사용가능

<https://librosa.github.io/librosa/generated/librosa.feature.melspectrogram.html>

Deep Learning with Spectrogram

Sample Codes

```
35 def get_feature_from_librosa(filepath, feature_size):
36     global first
37     global sig
38     global sample_rate
39
40     sample_rate = 16000
41     hop_length = 128
42
43     sig, sample_rate = librosa.core.load(filepath, sample_rate)
44
45     assert sample_rate == 16000, '%s sample rate must be 16000 but sample-rate is %d' % (filepath, rate)
46
47     mel_spectrogram = librosa.feature.melspectrogram(sig, n_mels=40, n_fft=512, hp_length=128)
48
49     return mel_spectrogram
```

Thank You!