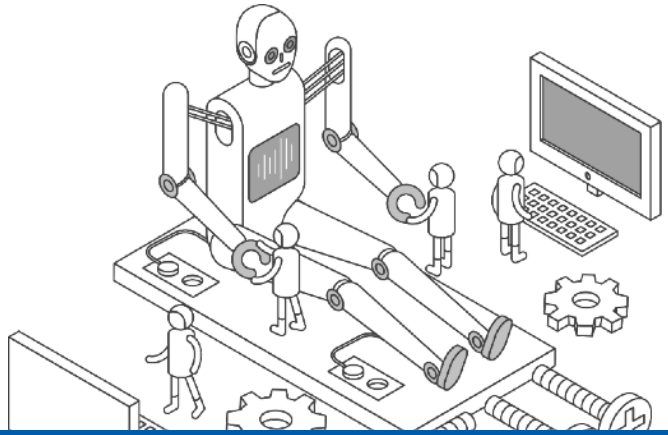
2022 디지털 전환을 위한 AI 전문가 과정

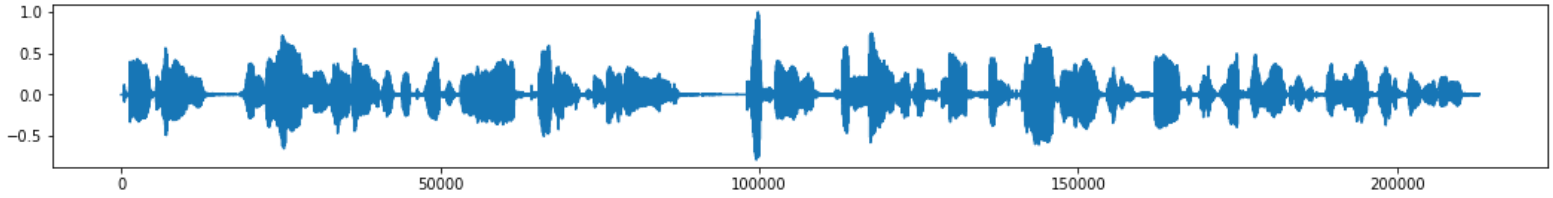
ARTIFICIAL INTELLIGENCE
BIG DATA
SMART FACTORY

AI·빅데이터 심화과정

"딥러닝으로 만드는 음성인식/음성합성"

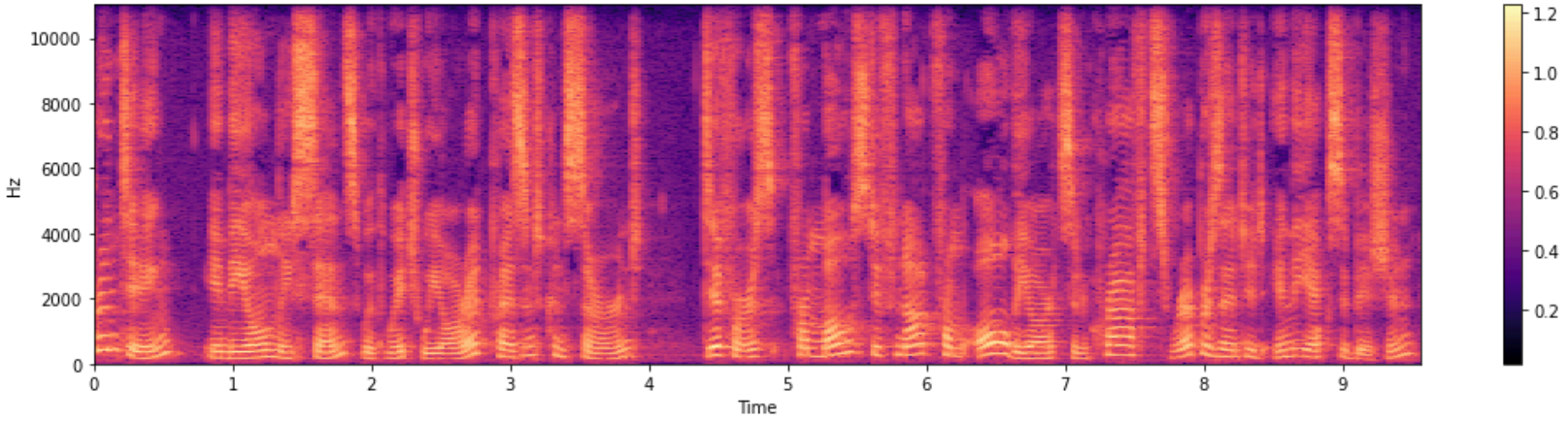


Wave



STFT

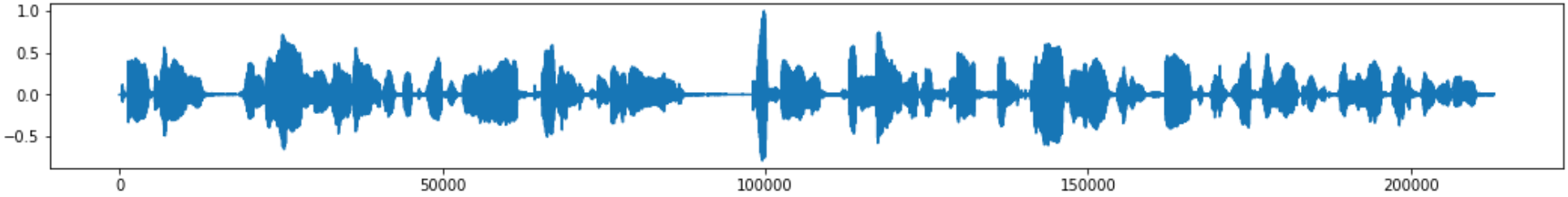
Spectrogram
(magnitude)



Spectrogram Librosa 실습

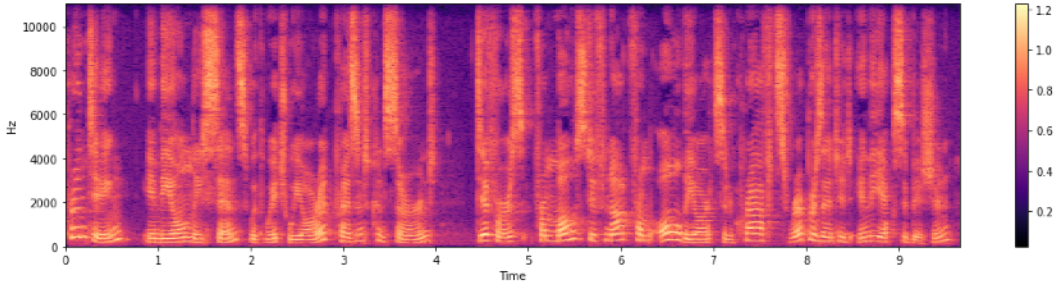
```
# 첫번째 파일 로드
wav, _ = librosa.core.load('LJ001-0001.wav')
# normalizing
wav /= max(abs(wav))
```

librosa.core.load함수로 wav파일을 로드합니다. sample들 중 가장 크기가 큰 값으로 나누어 normalization을 할 수 있습니다.




```
# spectrogram 구하기
spec = librosa.core.stft(wav, n_fft=2048, hop_length=512)
spec = np.abs(spec)
```

librosa.core.stft함수로 wav를 spectrogram으로 변환합니다. n_fft와 hop_length argument로 FFT size와 hop size를 설정할 수 있습니다. Magnitude값을 얻기 위해 abs function을 취합니다.

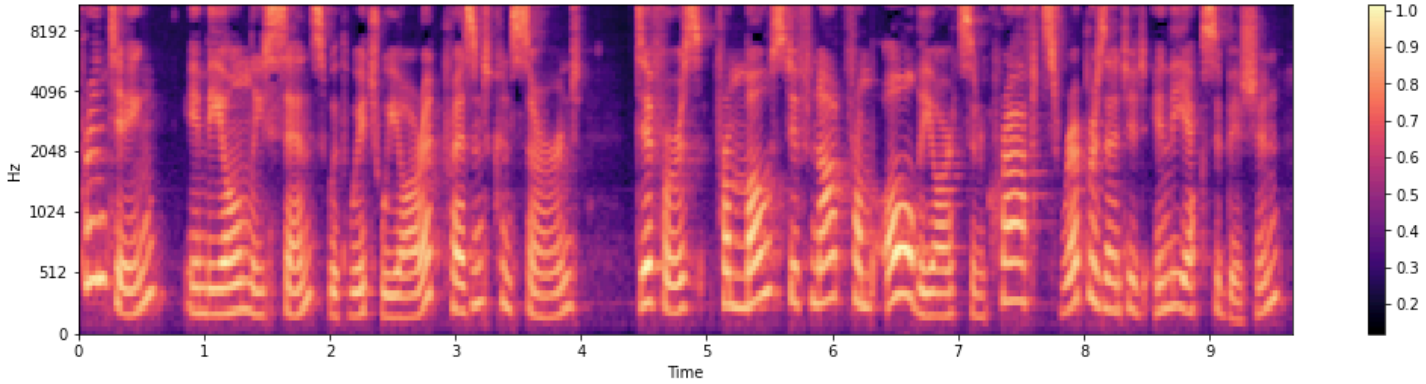


Mel-spectrogram Librosa 실습

```
spec = librosa.core.stft(wav, n_fft=2048, hop_length=512)
spec = np.abs(spec)

# spectrogram에 mel-matrix 적용
mel = mel_matrix @ spec
```

```
mel-spectrogram shape: (80, 416)
```



- Spectrogram에 mel matrix를 곱하여 mel-spectrogram을 구합니다.
- Matrix multiplication은 python의 내장 operator인 @로 수행할 수 있습니다.
- Spectrogram의 shape이 (n_fft/2+1, time)이고, mel matrix의 shape이 (n_mels, n_fft/2+1)이므로, mel-spectrogram의 shape은 (n_mels, time)이 됩니다.

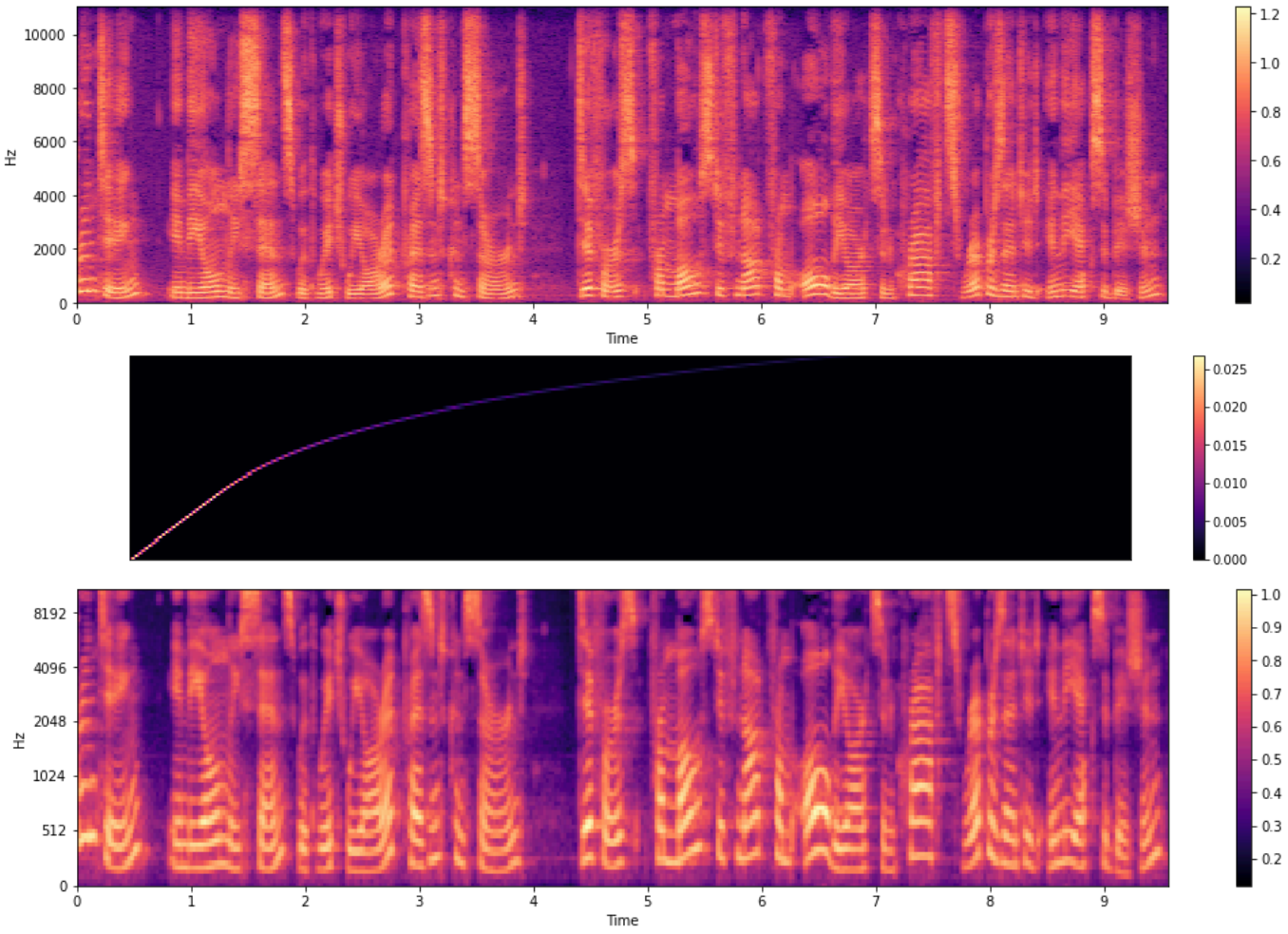
Spectrogram

•

Mel-scale matrix

=

Mel-spectrogram



- CTC는 2006년에 Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks 논문에서 제안된 알고리즘입니다.

- CTC는 segment되지 않은 sequence data에 labelling을 하기 위한 목적에서 설계되었습니다.

- ASR에 사용되는 audio sequence, label sequence 쌍과 같이 길이가 서로 다르고 alignment가 주어지지 않은 조건에 사용될 수 있습니다.

- CTC는 input sequence가 output sequence보다 길어야한다는 제약이 있습니다. (Transducer는 제약 없음) 대체로 audio sequence는 label sequence보다 길므로 이에 적합합니다.

- CTC는 loss를 구하고 트레이닝을 하도록 하는 framework이고, audio sequence를 다루기 위한 encoder는 RNN, CRNN, 또는 Transformer등 sequential data를 다룰 수 있는 모델이면 무엇이든 적용 가능합니다.

- CTC algorithm에서는 path라는 개념을 도입합니다.

- Audio sequence 와 label sequence 의 길이가 다르므로 각 label을 audio feature에 대응시키는 방법이 필요합니다.

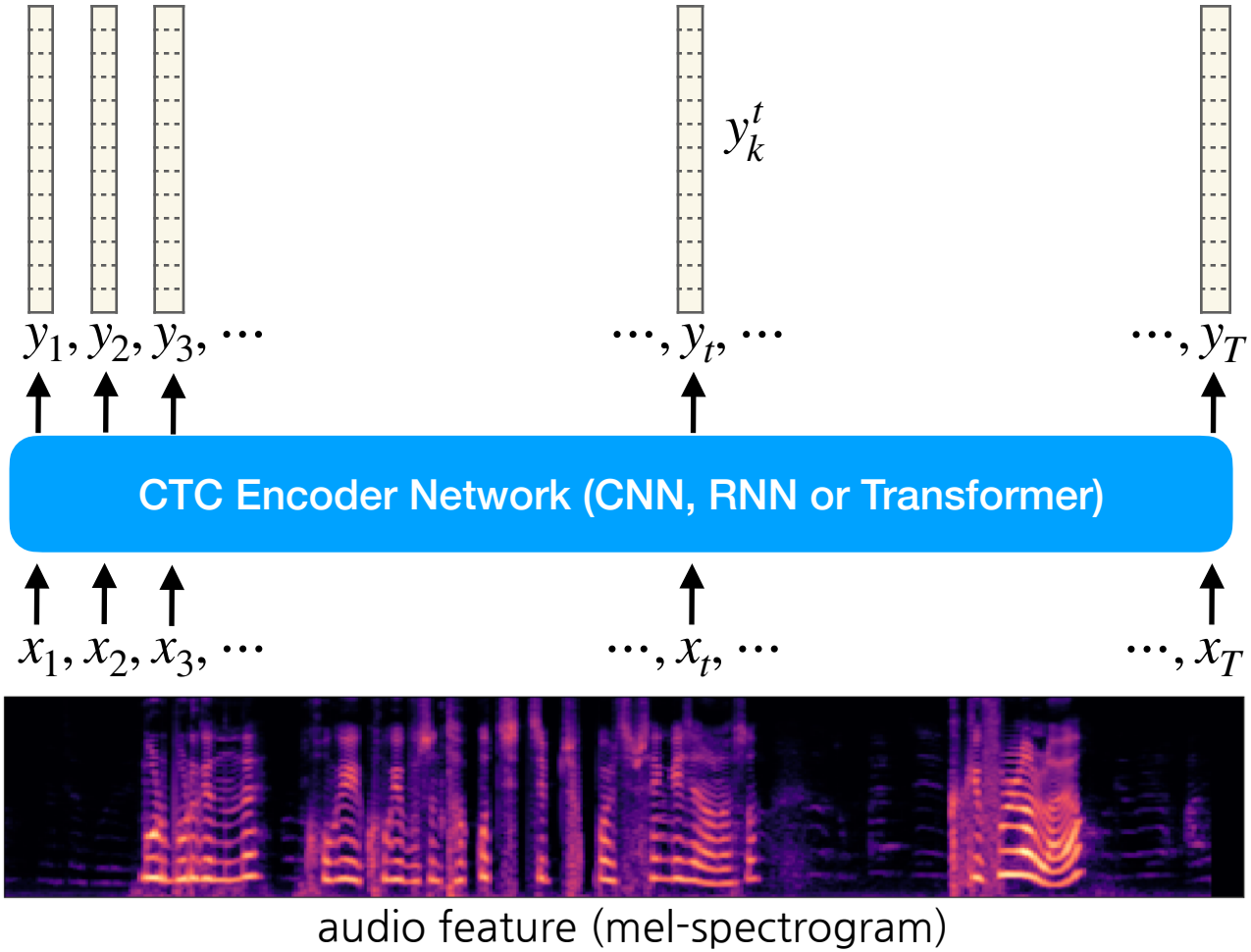
- 먼저, audio sequence에는 label sequence에 있는 label들 중 해당하지 않는 feature들도 있을 것 (말하지 않는 구간, 숨소리)이라는 가정하에 blank 을 추가합니다.

- 또한 label 하나가 여러 audio feature에 대응될 것이라는 가정하에 label이 반복될 수 있도록 합니다.

- label이 path상에서 반복된 것인지 원래 label sequence에서 연속해서 나타난 것인지 구별하기 위해 항상 구분되는 label 간에는 blank 을 넣는 것으로 정합니다. 즉, path 는 sequence 에 해당하지만, path 는 sequence 에 해당합니다.

-
  path는 항상 ‘blank’에서 시작하고 끝나는 것으로 정합니다.
- 이러한 식으로 sequence 라는 label sequence에 대해 path 또는 와 같은 path를 만들어 볼 수 있습니다.

- 이렇게 정의된 path는 audio sequence와 같은 길이로 만들 수 있습니다.



- $x_1, x_2, x_3, \ldots, x_T$: 길이 T를 갖는 audio feature sequence, encoder의 입력으로 사용

- $y_1, y_2, y_3, \ldots, y_T$: encoder의 출력 벡터들

- y_k^t: t시점에 label k가 관측될 확률, label 0은 blank (softmax activation 사용)

- Audio sequence $\mathbf{x}$가 주어졌을 때, 경로 $\pi = \{\pi_1, \pi_2, \ldots, \pi_T\}$가 관측될 확률은

$$p(\pi \mid \mathbf{x}) = \prod_{t=1}^{T} y_{\pi_t}^t$$

CLASS `torch.nn.CTCLoss(blank=0, reduction='mean', zero_infinity=False)` [SOURCE]

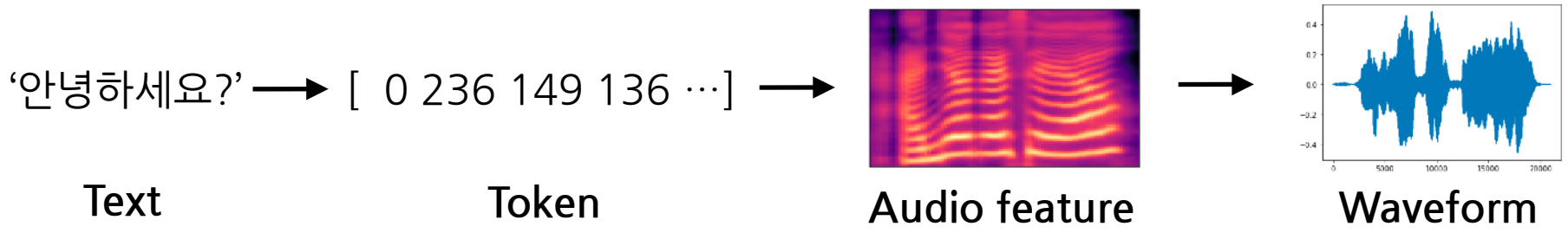
The Connectionist Temporal Classification loss.

Calculates loss between a continuous (unsegmented) time series and a target sequence. CTCLoss sums over the probability of possible alignments of input to target, producing a loss value which is differentiable with respect to each input node. The alignment of input to target is assumed to be "many-to-one", which limits the length of the target sequence such that it must be $\leq$ the input length.

Parameters

- **blank** (*int, optional*) – blank label. Default 0 .
- **reduction** (*string, optional*) – Specifies the reduction to apply to the output: `'none'` | `'mean'` | `'sum'` . `'none'` : no reduction will be applied, `'mean'` : the output losses will be divided by the target lengths and then the mean over the batch is taken. Default: `'mean'`
- **zero_infinity** (*bool, optional*) – Whether to zero infinite losses and the associated gradients. Default: `False` Infinite losses mainly occur when the inputs are too short to be aligned to the targets.

- TTS는 Text-to-Speech의 약자로 주어진 문장을 음성 데이터로 변환하는 작업을 말하며, speech synthesis라고도 말합니다.
- 어진 음성 데이터를 문장으로 변환하는 작업인 ASR(Automatic Speech Recognition)와 정확히 반대 개념이라고 생각할 수 있습니다.
- Audio feature는 현재에는 mel-spectrogram이나 waveform을 주로 사용하고 과거에는 MFCC를 사용했습니다.
- 입력이 되는 token들은 다음과 같이 integer들의 sequence로 표기할 수 있습니다.
  $\mathbf{x} = (x_1, x_2, \ldots, x_L)$, where $x_l \in \mathbf{N}$
- 출력이 되는 audio feature들은 다음과 같이 d-dimensional vector들의 sequence로 표기할 수 있습니다.
  $\mathbf{y} = (\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_T)$, where $\mathbf{y}_t \in \mathbf{R}^d$



- Tacotron2는 2018년에 Natural tts synthesis by conditioning wavenet on mel spectrogram predictions 논문에서 제안되었습니다.
- Tacotron2은 크게 text를 인코딩하는 encoder와 mel-spectrogram을 디코딩하는 decoder 모듈로 이루어져 있습니다.
- 이에 더해 attention 모듈이 encoder와 decoder 모듈간의 정보 전달을 위해 사용됩니다.
- 또한 별도의 모델인 WaveNet을 통해 mel-spectrogram을 waveform으로 변환하여 사실적인 음성을 만드는데 기여합니다.

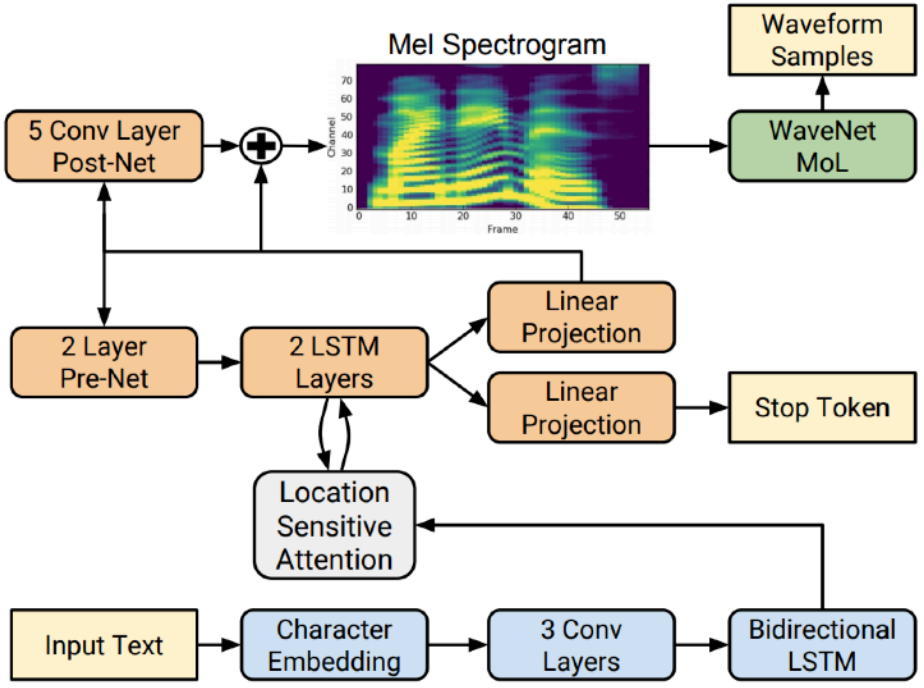


Fig. 1. Block diagram of the Tacotron 2 system architecture.

Shen, Jonathan, et al. "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions." 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018.

- Encoder는 input text를 입력받아 decoder에 넘겨줄 encoding data를 만드는 역할을 합니다.
- Character embedding : input text (tokens)를 입력받아 embedding table을 이용하여 vector sequence를 출력합니다. Embedding table은 trainable한 parameter로 이루어져 있습니다.
- 3 Conv Layers : 가까운 거리에 있는 token간의 context를 파악하는데 사용합니다.
- Bidirectional LSTM : 멀리 떨어져 있는 token간의 context를 파악하는데 사용합니다.

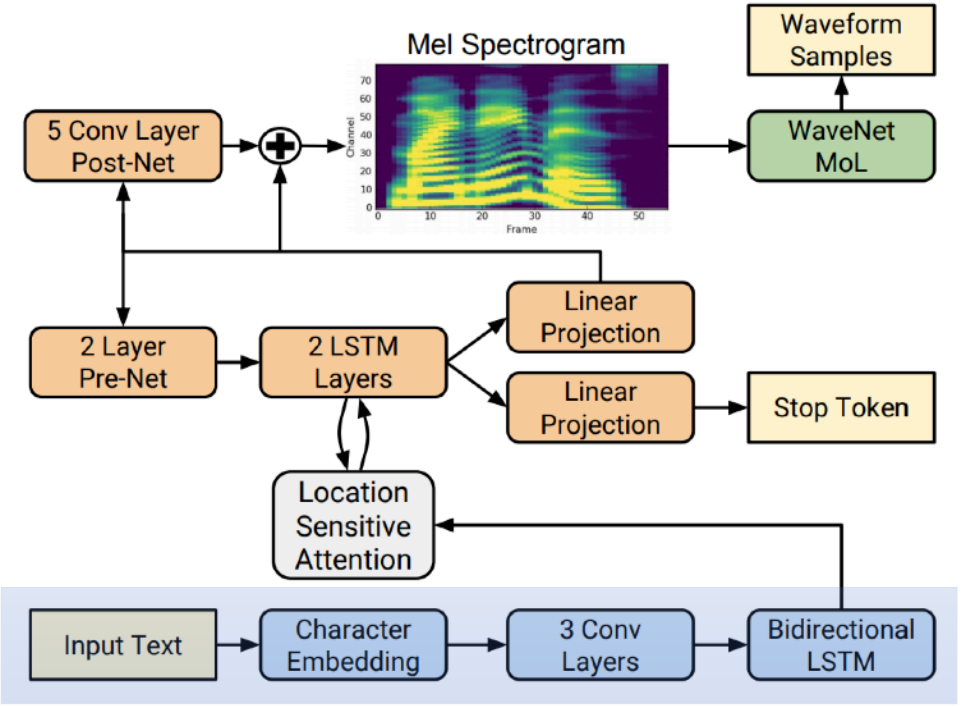


Fig. 1. Block diagram of the Tacotron 2 system architecture.

Shen, Jonathan, et al. "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions." 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018.

- Decoder는 auto-regressive하게 작동합니다. 즉, 이전의 mel frame들을 받아 현재 시점의 mel frame을 생성합니다. 이 때, attention 모듈에 의해 encoding data를 참조합니다.

- 2 Layer Pre-Net : 이전 step의 mel frame을 받아 두 개의 linear layer를 적용한 결과를 출력합니다.

- 2 LSTM Layers : 이전의 mel frame들의 context를 분석하는데 사용합니다. 두 LSTM 사이에 attention 모듈에 의해 가져온 context를 concat으로 더합니다.

- Linear Projection : mel frame과 stop token을 얻는데 사용합니다.

- 5 Conv Layer Post-Net : Auto-regressive하게 한 frame씩 생성한 mel-spectrogram의 품질을 향상시키기 위해 사용합니다.

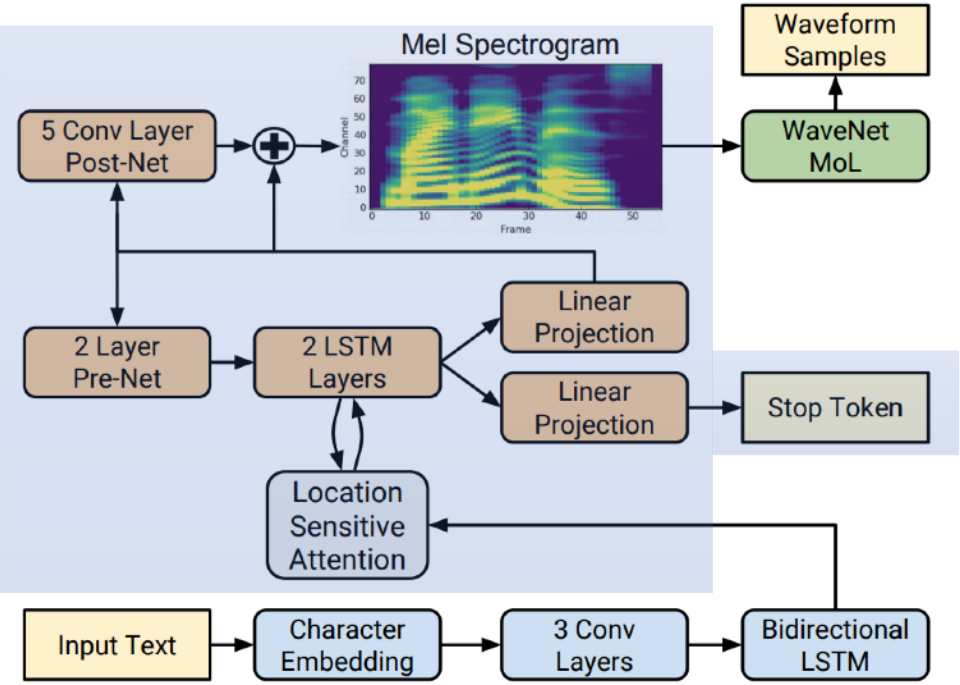


Fig. 1. Block diagram of the Tacotron 2 system architecture.

Shen, Jonathan, et al. "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions." 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018.


똑똑
파이팅!!