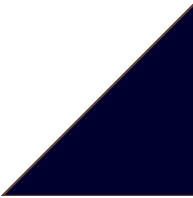




TACOTRON

: towards end-to-end speech synthesis

15 이규석





CONTENTS

01

Tacotron이란?

02

End to End

03

Encoder-Decoder

04

Tacotron
구조 및 평가



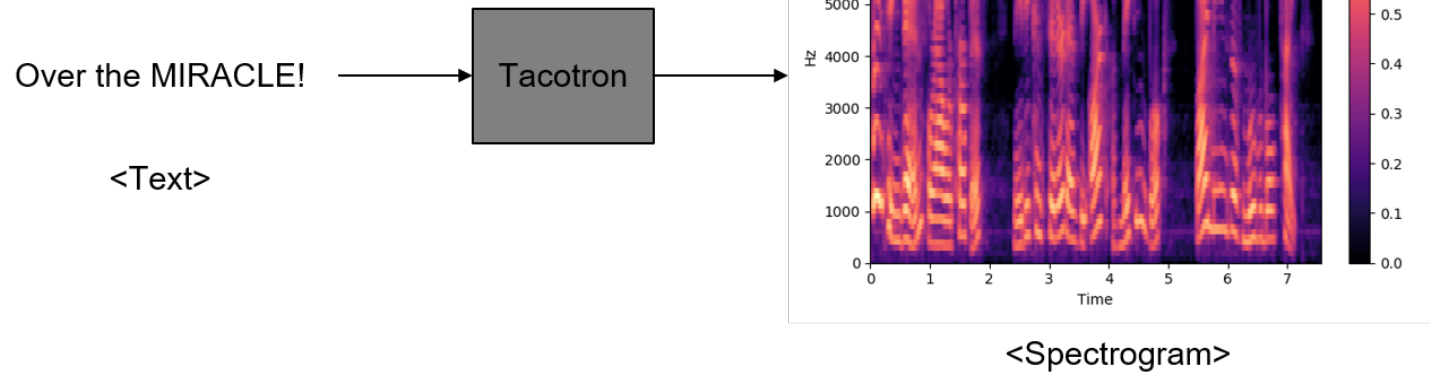


01

Tacotron이란?



01. Tacotron이란?



- 2017년 구글에서 만든 TTS 모델
- 텍스트를 입력 받아서, raw spectrogram을 바로 생성 가능
- <text, audio> pair를 이루는 End to End model

<https://google.github.io/tacotron/index.html>



02

End to End Model



02. End to End model 이란?

end-to-end 딥러닝은 자료처리 시스템 / 학습시스템에서 여러 단계의 필요한 처리과정을 한번에 처리한다.

즉, 데이터만 입력하고 원하는 목적을 학습한다.

	장점	단점
1	Let the data speak	May need large amount of data
2	Less hand-designing of components needed	Excludes potentially useful hand-designed components

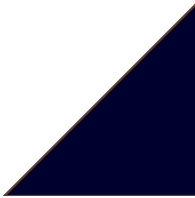


02. End to End model

TTS가 End to End가 아닌 경우,

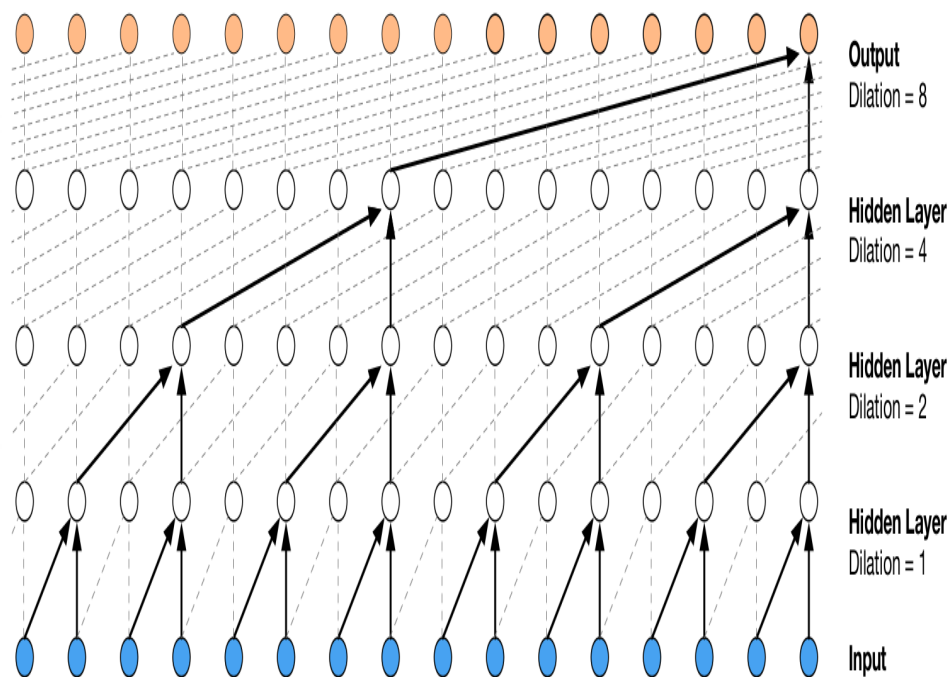
1. 방대한 Domain 지식이 요구 된다.
2. 디자인에 어려움이 있다.
3. 트레이닝 파이프라인 별로 에러 누적, 복잡

End to End인 경우,

1. <text, audio> 만으로도 학습 가능
 2. Feature engineering이 간단하다.
 3. 새로운 데이터에 Adaptable하고, 노이즈에 강함
- 

02. End to End model

Tacotron 이전의 모델들



Wavenet (2016)

- Tacotron의 vocoder로 사용(Tacotron2부터)
- 샘플 수준의 autoregressive model이라 너무 느림
- 바로 TTS로 사용할 수 없음



Deep Voice 3 Pytorch

DeepVoice (2017)

- TTS 파이프라인을 Neural Net으로 대체
- 학습이 End to End로 되지 않는다.



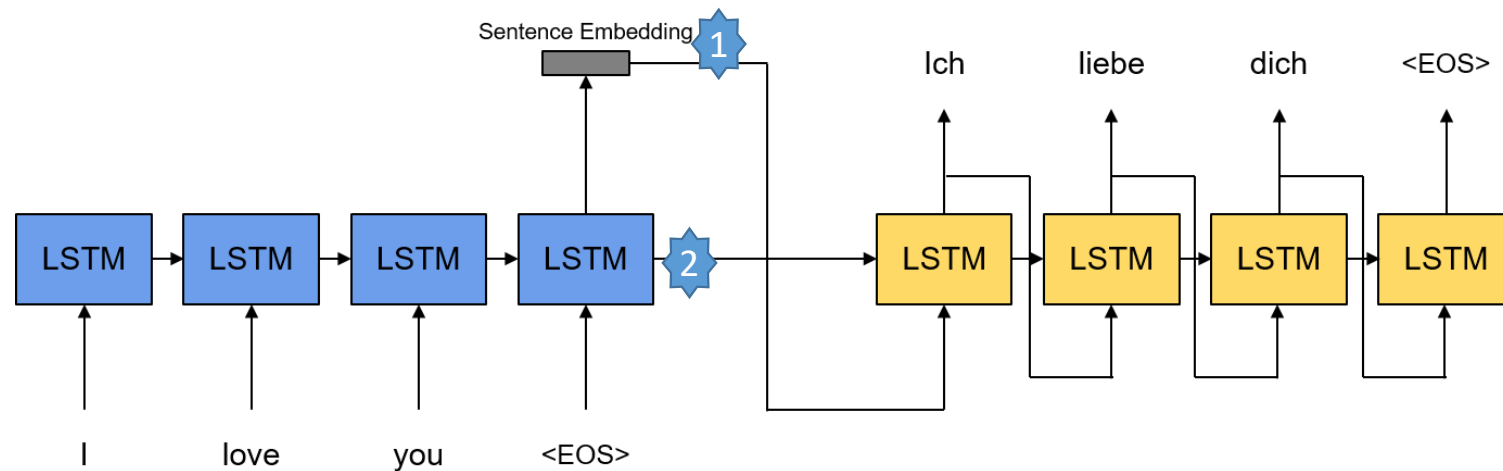
03

Encoder - Decoder



03. Encoder – Decoder

기본적인 모델



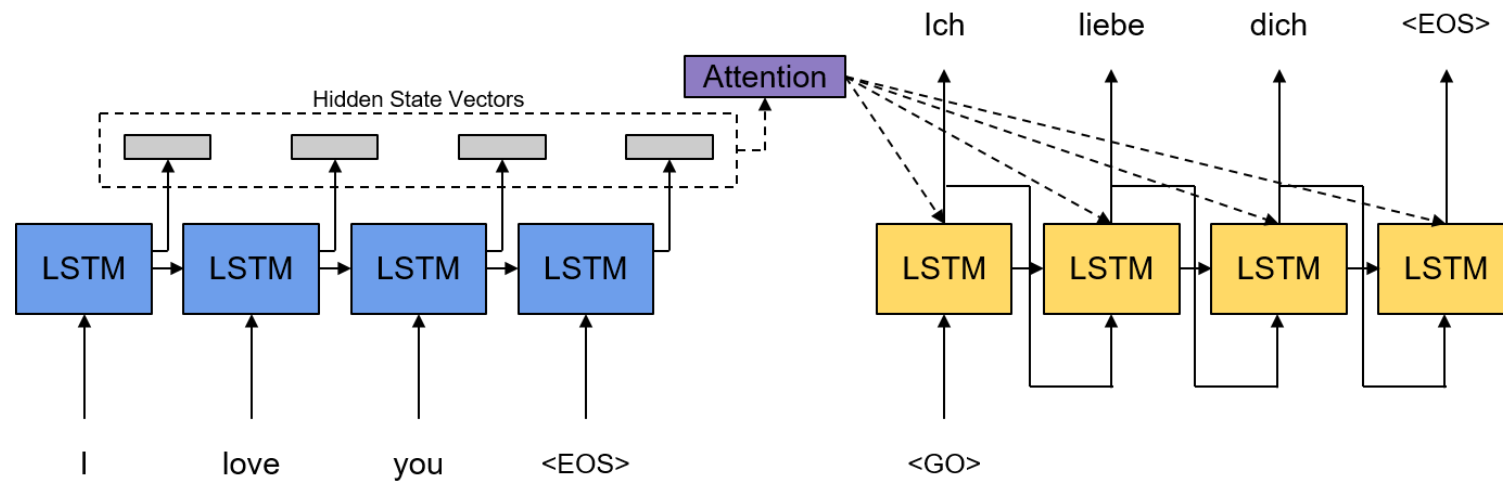
- Encoder와 Decoder를 RNN으로 구성
- 1번: 고정된 크기의 Sentence Embedding
- 1번 → 디코더의 첫번째 타임 스텝의 Input
- 2번 → 디코더의 첫번째 타임 스텝의 Hidden state vector

한계

- 1) 고정된 크기의 벡터에 모든 정보 압축
→ 정보 손실
- 2) Vanishing Gradient

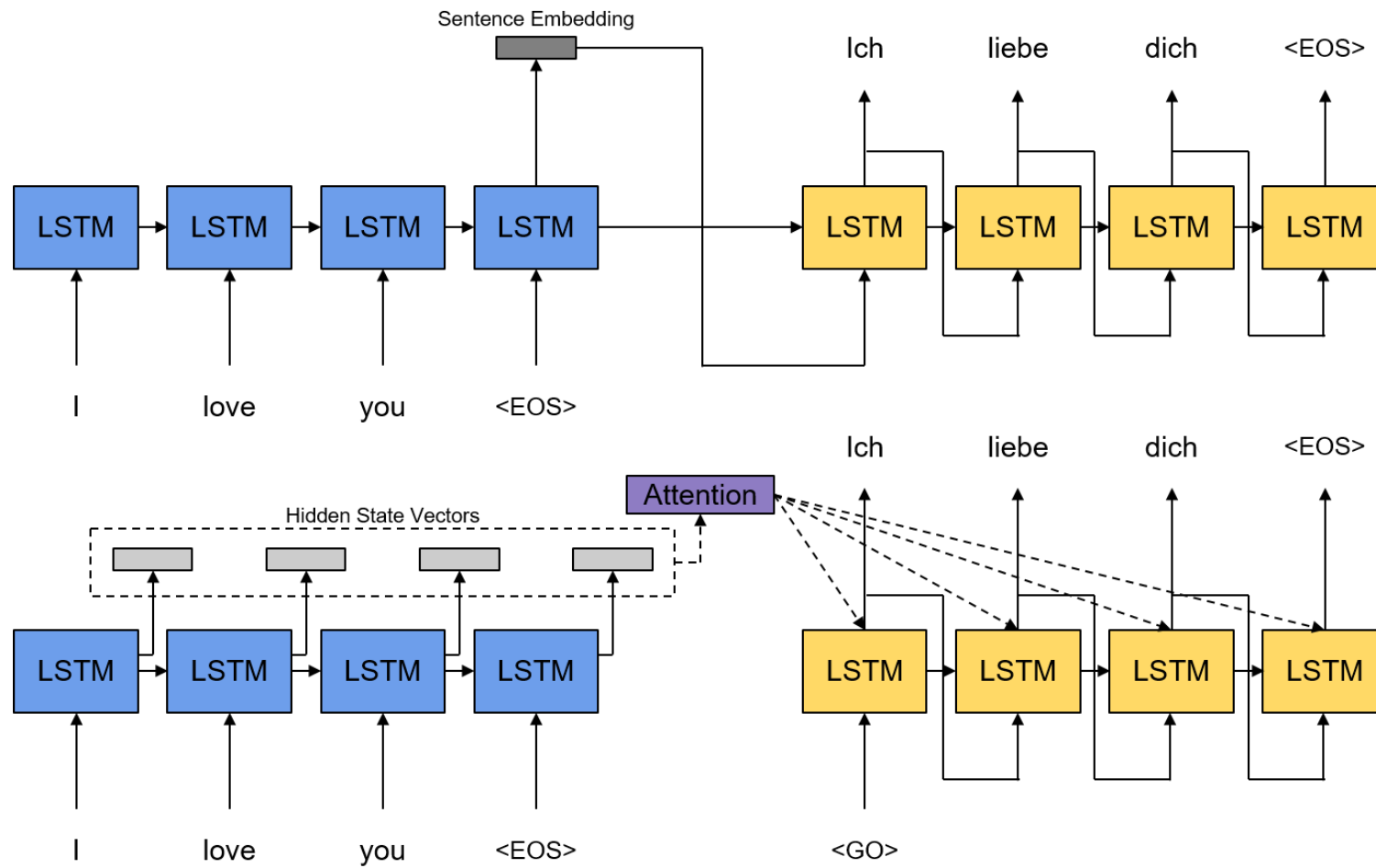
03. Encoder – Decoder

Attention 모델



- 인코더: 각각의 단어들을 임베딩한다.
- 디코더: 출력 단어를 예측하는 시점마다, 인코더에서의 전체 입력 문장을 다시 한번 참고
- 디코더: 같은 비율이 아닌, 연관성 있는 것을 중점으로 예측

03. Encoder – Decoder



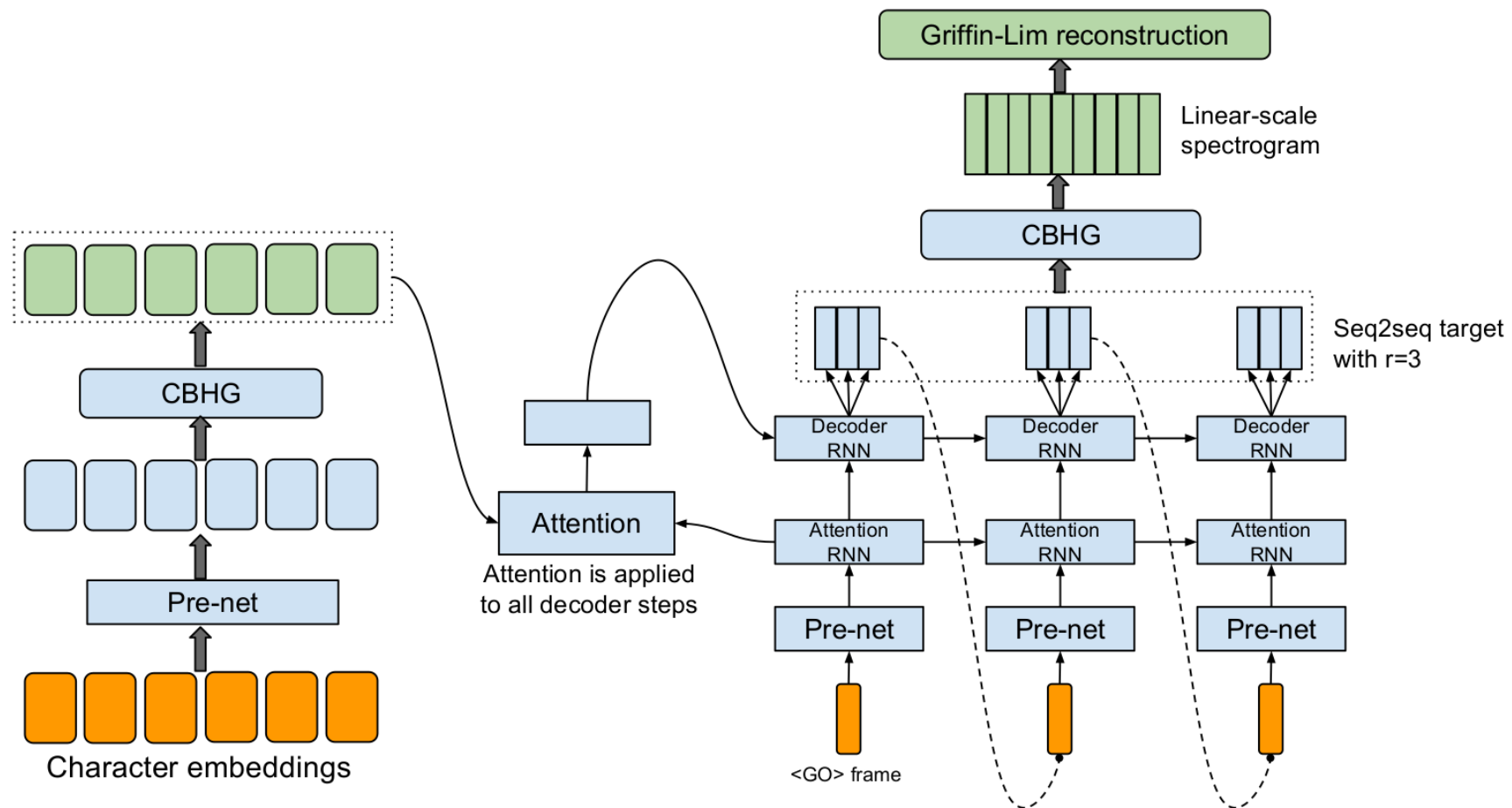


04

Tacotron 구조 및 평가

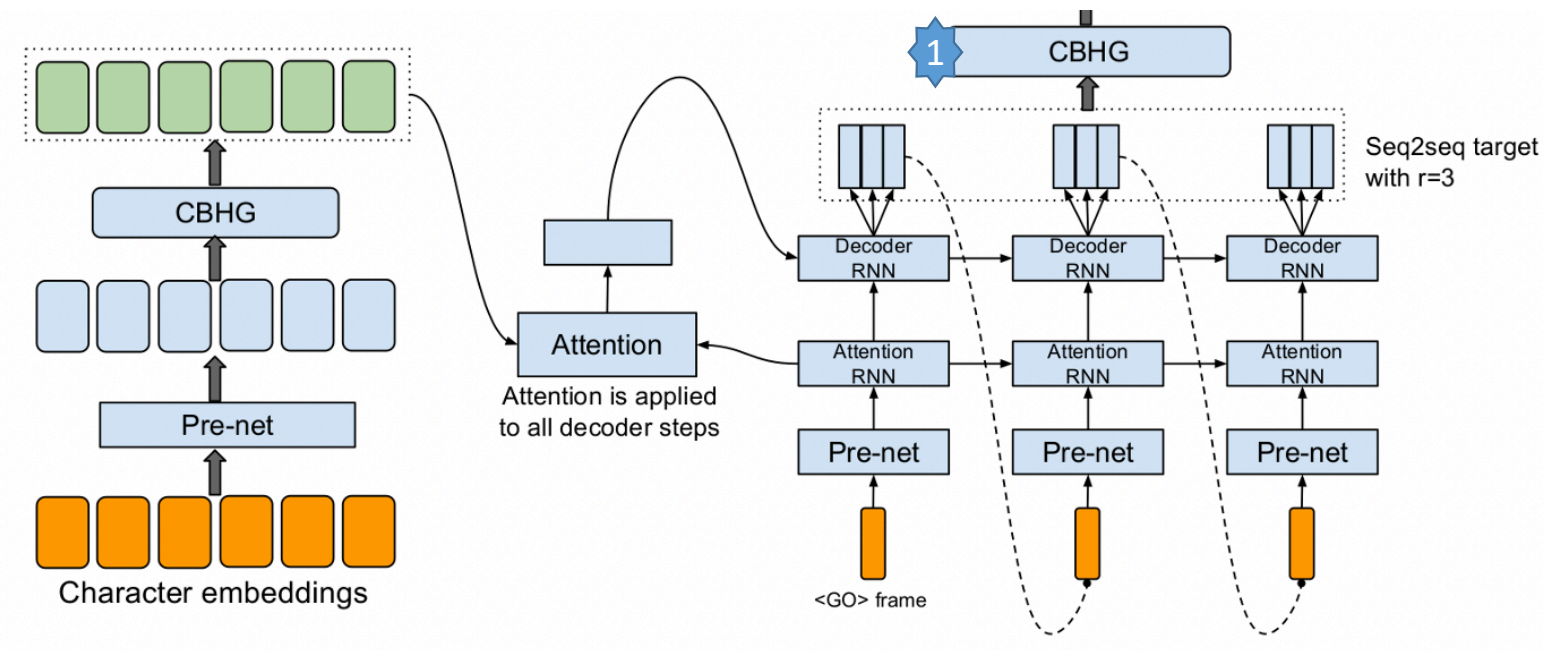


04. Tacotron 모델 구조



04. Tacotron 모델 구조

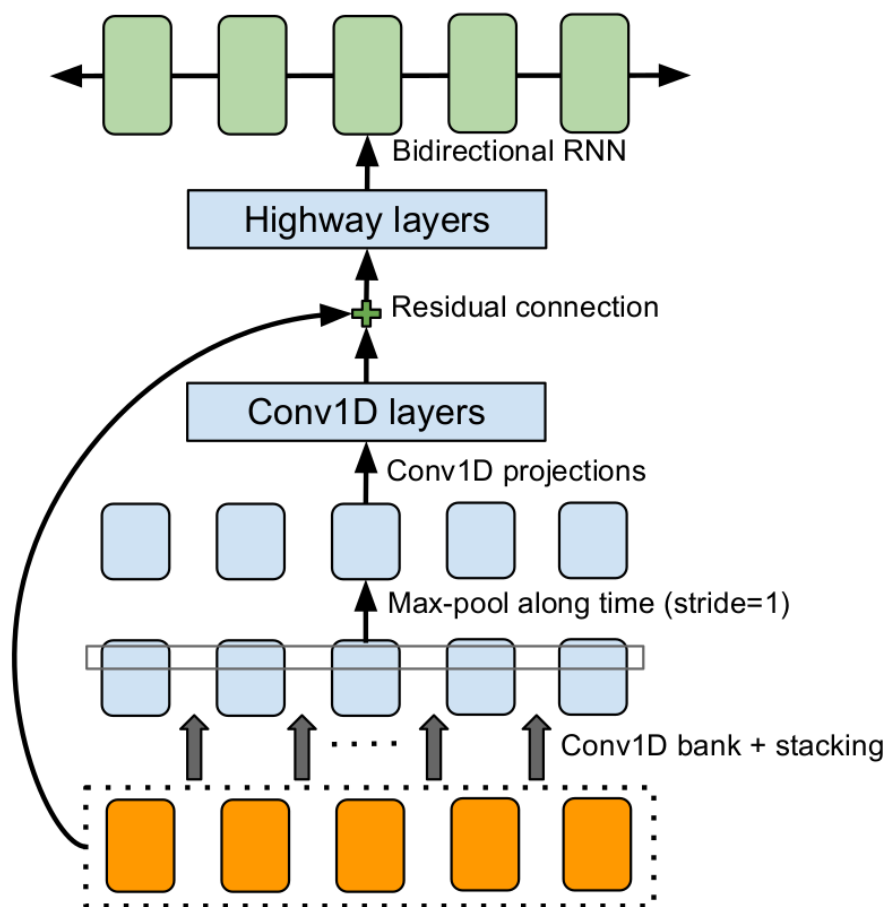
Encoder - Decoder



- Network(LSTM & CNN layers): character sequences → Mel Spectrogram
- Encoder: character sequences → Internal feature representation
- Decoder: Internal feature representation → Mel Spectrogram
- 1 CBHG: Mel Spectrogram → Linear Spectrogram

04. Tacotron 모델 구조

CBHG



CBHG: Convolution Bank + Highway + GRU

흐름:

Sequence time step을 따라서
1D Convolution Kernel이
여러 사이즈를 가진 여러 종류가 사용
→ Max pooling
→ 1D Convolution
→ Residual connection
→ Highway
→ Bidirection GRU

참고:

- 각각의 1D Convolution을 거친 이후 Batch Norm
- Highway: Gating 구조를 사용하는 residual 네트워크
- GRU: Gated recurrent unit

04. Tacotron 모델 구조

Training

1. Data 준비

입력은 Text, 출력은 Mel & Linear Spectrogram이므로
→ Target이 되는 Mel & Linear Spectrogram을 준비

2. Loss function

→ L1 Distance 이용

```
mel_loss = tf.reduce_mean(tf.abs(mel_outs - mel_targets))
```

```
lin_loss = tf.reduce_mean(tf.abs(lin_outs - lin_targets))
```

```
loss = mel_loss + lin_loss
```

3. Optimizer

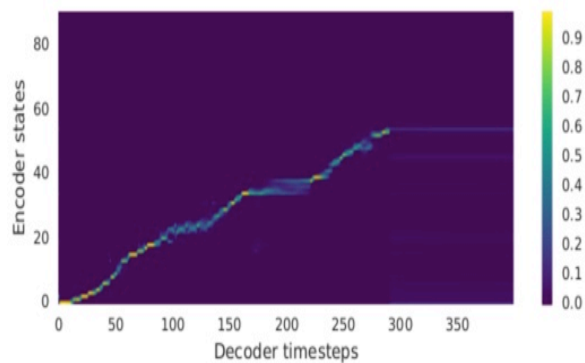
→ Adam 사용

4. learning rate

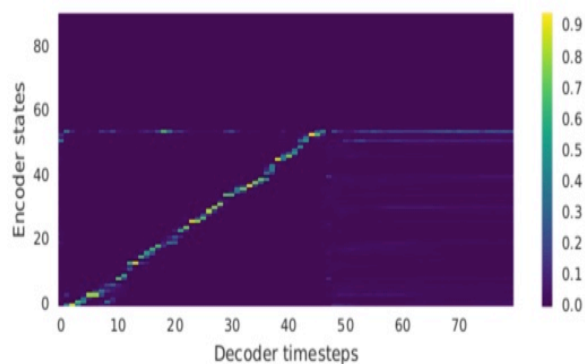
→ Step마다 순차적으로 줄임

04. Tacotron 모델 평가

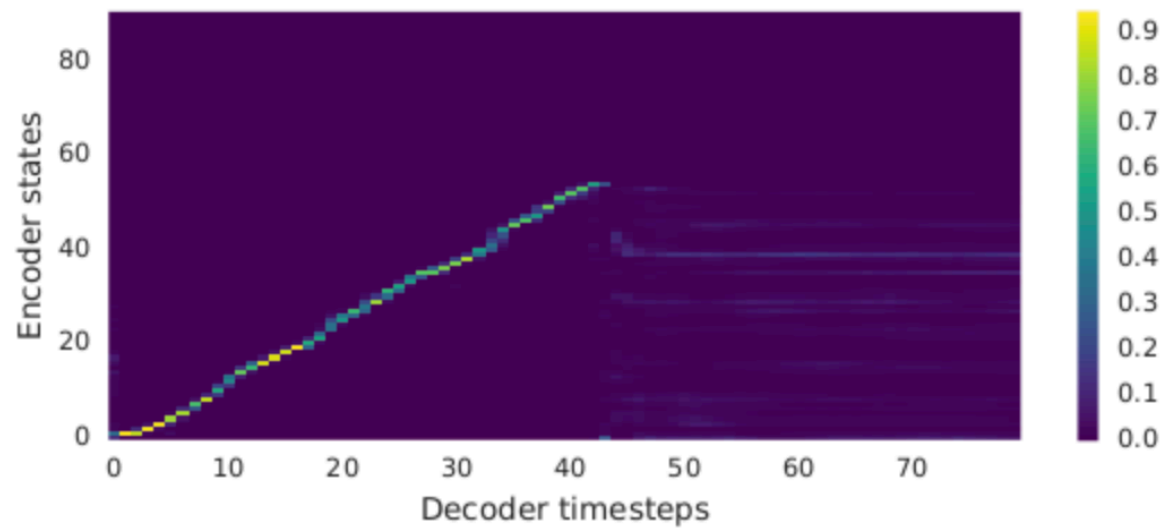
Testing- alignment



(a) Vanilla seq2seq + scheduled sampling



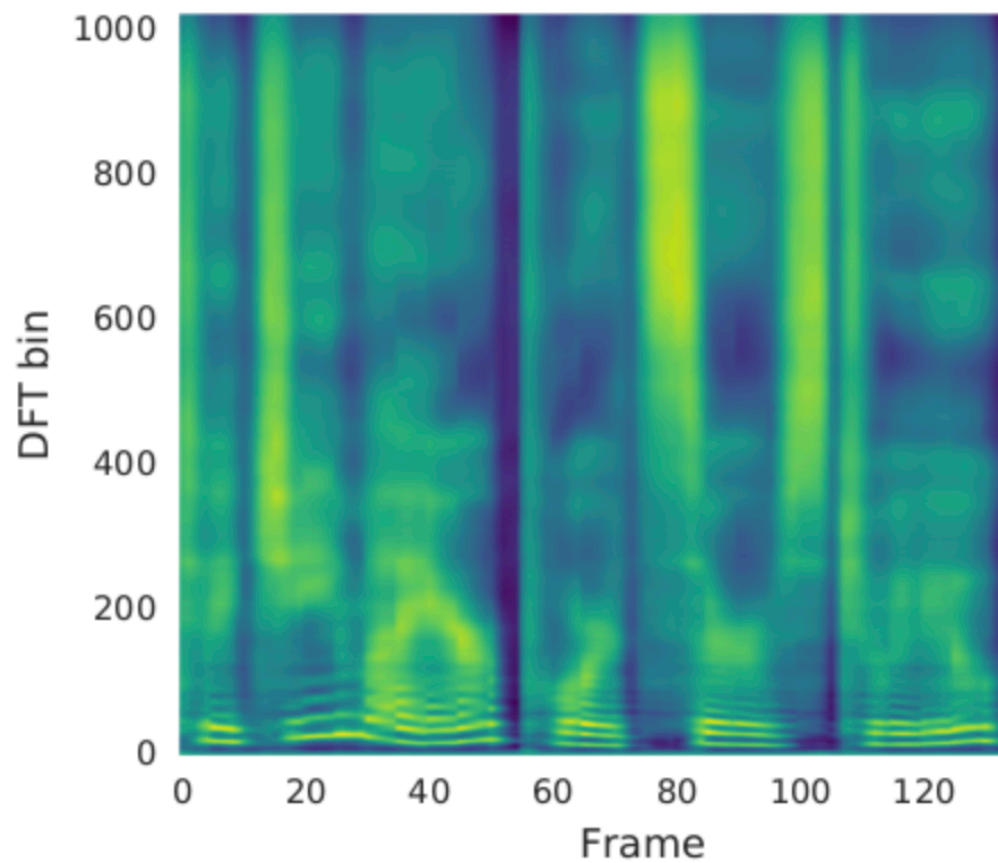
(b) GRU encoder



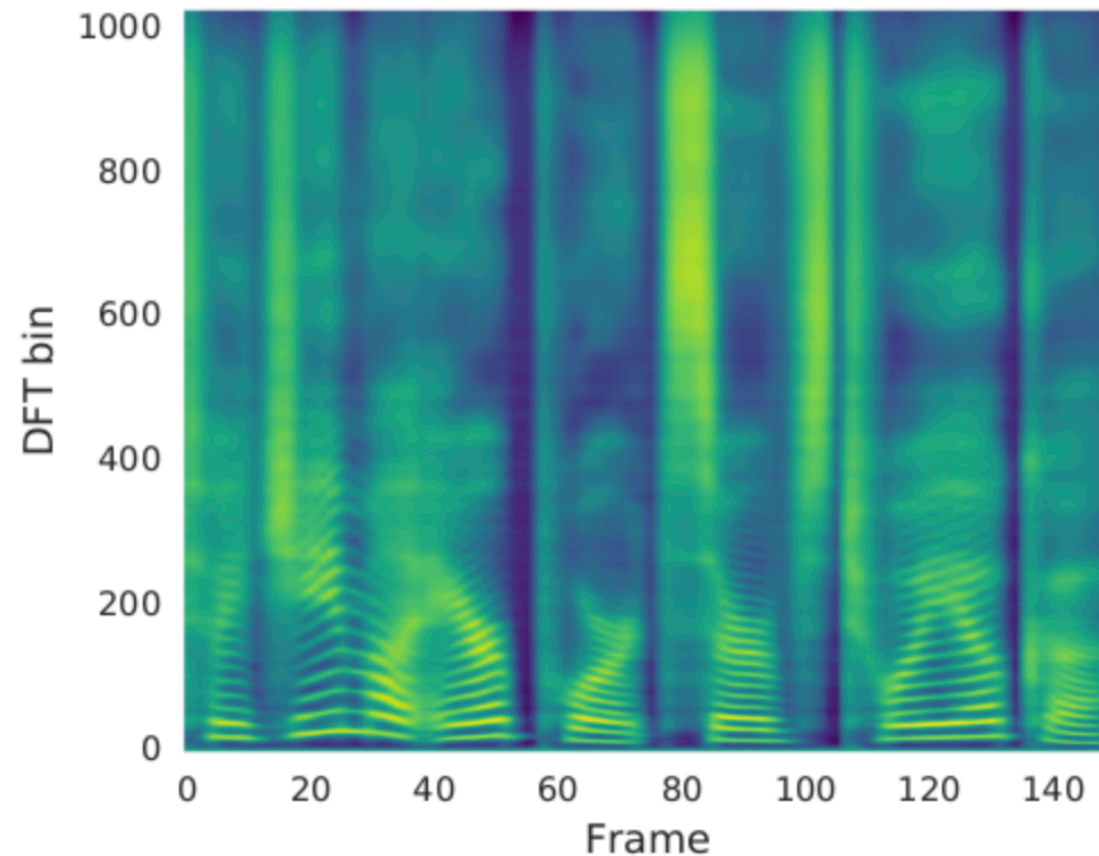
(c) Tacotron (proposed)

04. Tacotron 모델 평가

Testing-Harmonic



(a) Without post-processing net



(b) With post-processing net



05

Q & A





MelSpectrogram이 궁금하다

1. 푸리에 변환

음성 신호에 푸리에 변환을 적용하면, 음성신호에 저음과 고음을 정량적으로 구할 수 있다.

2. STFT (Short Time Fourier Transform)

음성을 작게 (0.01초 수준)으로 잘라서 작은 조각에 푸리에 변환을 적용

→ 이를 Spectrogram이라고 한다.

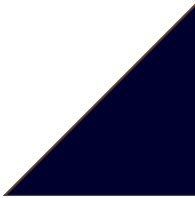
3. MelSpectrogram

Spectrogram에 mel-filter를 적용한다.

이는 사람의 청각기관이 고음보다 저음 주파수 변화에 민감한 것을 반영

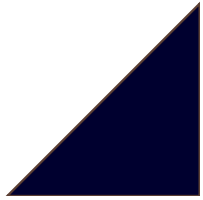
+ Faster, Parallel training of Network & WaveNet parts.

+ Emphasize low frequency signals, allowing better intelligibility when converting to speech





출처

1. <https://hcnoh.github.io/2018-12-11-tacotron>
 2. <https://www.youtube.com/watch?v=xXMtY2oVzmY>
 3. <https://chldkato.tistory.com/176>
 4. <https://joytk.tistory.com/21>
 5. <https://google.github.io/tacotron/publications/tacotron/index.html>
 6. <https://glee1228.tistory.com/3>
 7. <https://wgonnamakeit.tistory.com/25>
 8. <https://paperswithcode.com/method/griffin-lim-algorithm>
 9. <https://ahnjg.tistory.com/93>
- 



Thank you

