Tacotron(1, 2)

research at google (2016, 2017) 발표

TTS(Text to Speech)

사전적 의미: 디지털 텍스트(Input)를 음성(Output)으로 변환하는 기술
통상적 의미: 음성합성 시스템

활용분야

- 음성 안내 시스템(물체 방법 설명): AI 스피커, 네비게이션, Siri, 빅스비
- 읽어주기 서비스(텍스트를 음성으로): 오디오북, 뉴스 읽어주기 기능
- 도네이크(후원 문구 변환): 트위치, 유튜버 음성 도네이션 기능

TTS가 어려웠던 이유

1. 문자를 음성으로 바꾸는데 복잡한 작업
   ex) 문자열 토큰화 → 운율 분석 → 음성 조각 검색, 선택 → 음성 합성 → 음성 출력

2. 각 작업의 난이도가 높아 전문가의 지식이 필요
   ex) 음성과 관련된 전반적인 지식, 음성 전처리 과정(Fourier Transform, MFCC)

3. 여러 단계로 분리된 작업 단계를 합쳤을 때 품질 보장이 안됨

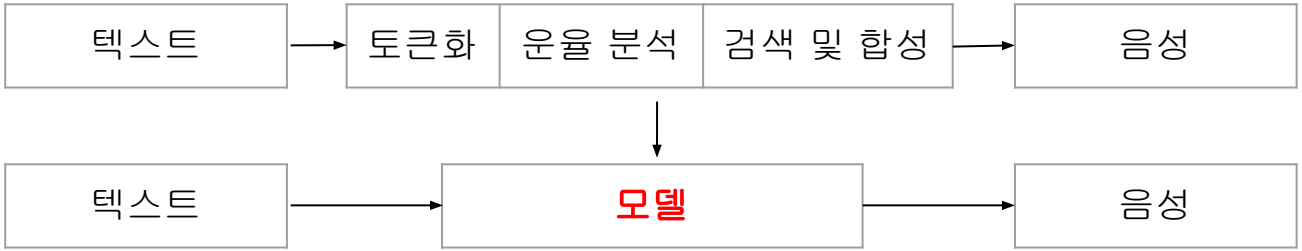
TTS 쉬워진 이유

1. 딥러닝 아키텍처를 이용, (텍스트, 음성) 쌍의 데이터로 개발이 가능
2. **전문가의 개입 없이** 데이터와 모델만으로 개발이 가능
3. E2E 모델로 구조를 변경하여 좋은 품질의 음성을 생성

음성 합성 방법

텍스트를 이용해서 음성을 만들어낼 때 **모델만 거치면** 나올 수 있도록

토큰화, 운율분석, 음성 검색 및 합성 과정을 **하나의 모델**로 통합



Tacotron의 구조

세부적으로 4개의 단계로 구성(Seq2Seq와 유사한 구조)

1. 인코더(Encoder)
2. 디코더(Decoder)
3. 어텐션(Attention)

음성 생성 단계

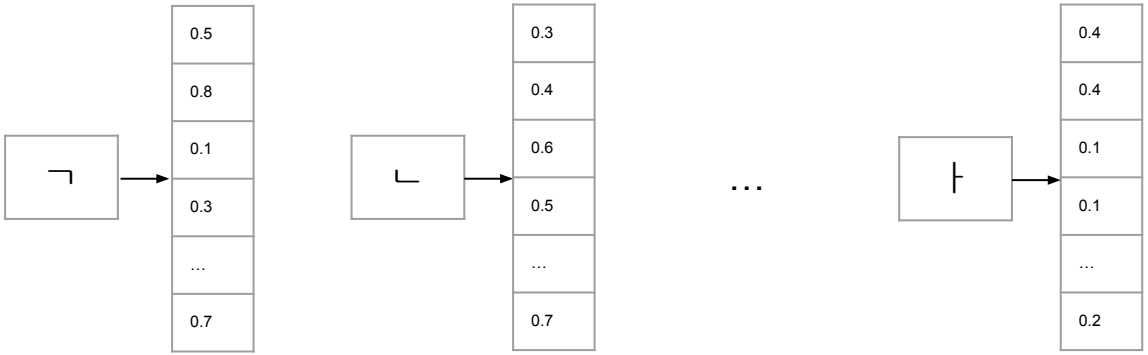
4. 보코더(Vocoder)

음성 합성 단계

Tacotron의 구조

인코더(Encoder): 문자로부터 특징을 추출

- 텍스트를 입력으로 받아 텍스트의 정보를 숫자로 변환하여 표현
- **문자단위**로 임베딩, 문자가 등장할 때마다 학습
  단, 초성의 자음과 종성의 자음은 반드시 구분!
- 학습 데이터에 없는 데이터를 표현 가능



Tacotron의 구조

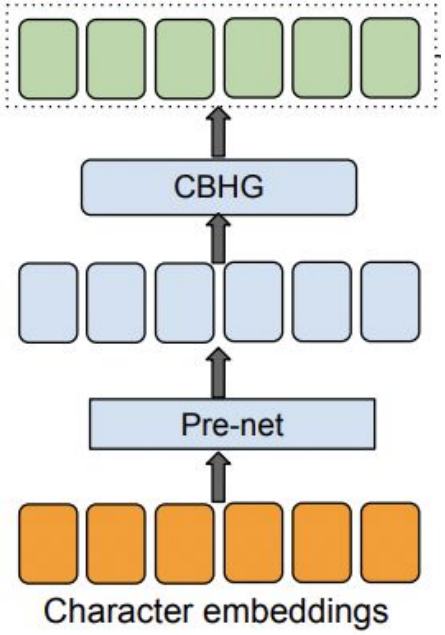
인코더(Encoder): pre-net, CBHG의 구조

pre-net

- (FC-Relu-Dropout) x 2

CBHG

- 1D-Conv-Bank,
- Highway Network,
- Bidirectional gated recurrent unit (GRU)



Tacotron의 구조

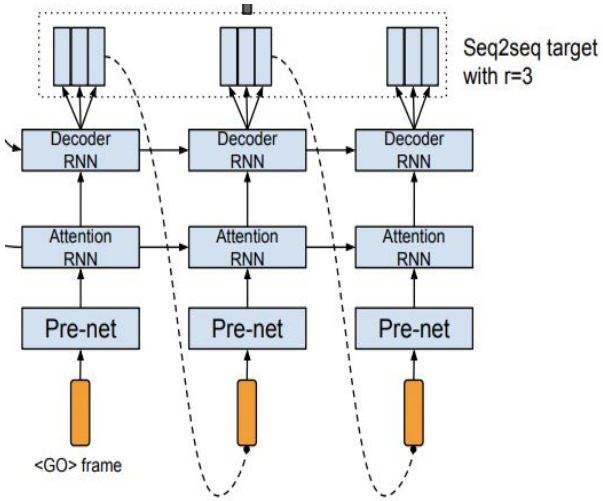
디코더(Decoder)

- RNN이 반복되는 구조
- 최종적으로 스펙트로그램을 생성

(sec2sec 디코더와 매우 유사)

Tacotron2에서는

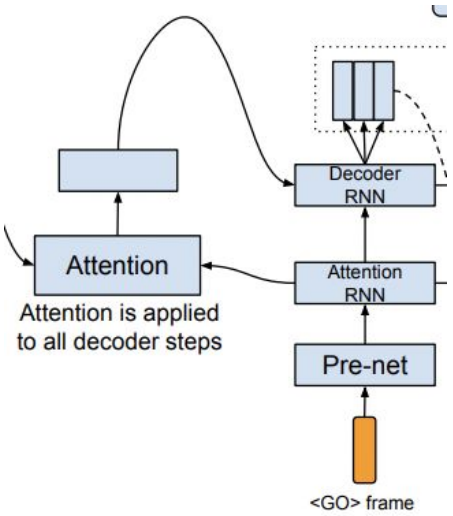
- 현재 시점의 멜스펙트로그램 생성
+ 현재 시점의 종료확률을 계산
+ 멜스펙트로그램의 품질 향상



Tacotron의 구조

어텐션(Attention): 매 시점 디코더에 인코더에서 정보를 추출 전달

- 문장 내 **어느 곳에 집중**할 것인지를 결정
- 첫 번째 음성은 몇 번째 단어에 집중?
  → 첫 번째 단어에 집중
- 공백을 만났을 때 얼마나 쉴 것인가?
  → (인공지능 데브 코스) vs (명수가 때렸다!)



어텐션(Attention)이 중요한 이유

학습하지 않았던 문장도 얼마나 유창하게 말할 수 있는가?

일반화(Generalization)

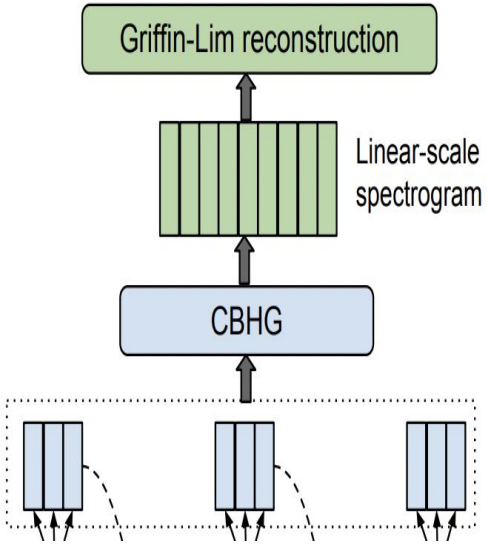
Tacotron의 구조

보코더(Vocoder)

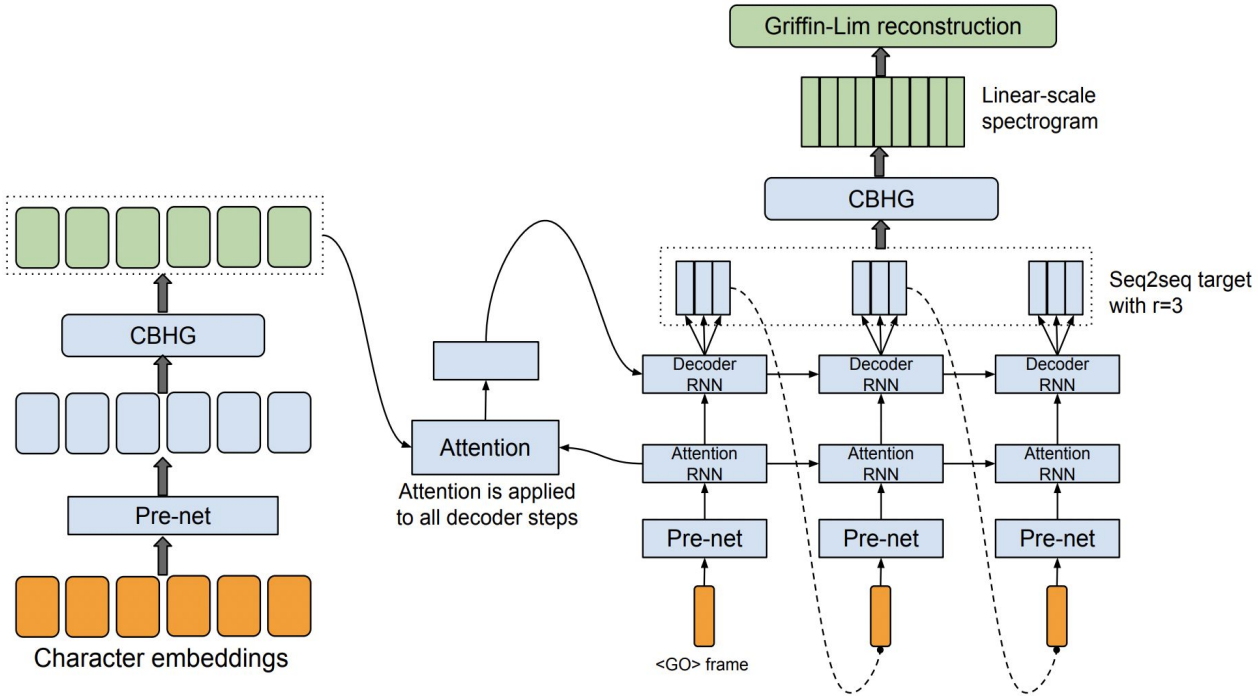
- 디코더에서 발생한 스펙트로그램을 이용
  음성으로 만들어주는 단계

- CBHG를 거쳐 Griffin-Lim을 거침

- Griffin-Lim은 스펙트로그램을
  음성으로 만들어주는 한 가지 알고리즘



Tacotron의 전체 구조



Tacotron2의 구조

Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions(2017)

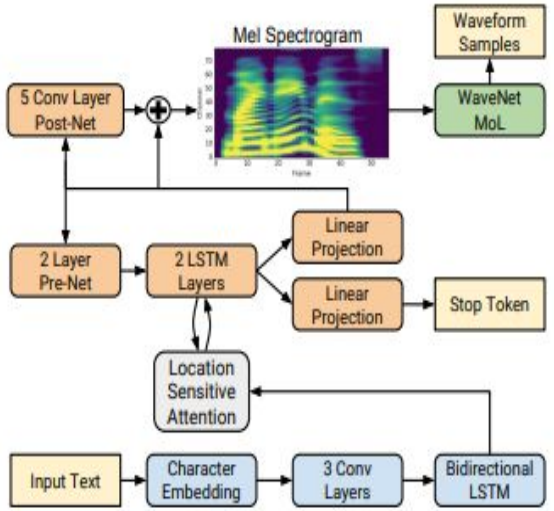
Tacotron1과의 차이점으로는

1. CBHG 사라짐
   → 3 Conv Layers와 양방향 LSTM로 대체
   또한 모델 내 모든 LSTM은 Zoneout LSTM

2. 디코더 부분에서
   두 개의 Linear Projection을 통해
   각각 현재 시점의 mel-vector와
   종료 확률(Stop Token)을 계산

3. Post-net(5 Conv Layers)를 통한 보정
   → 디코더에서 전체 멜스펙트로그램을 생성
   전체적인 멜스펙트로그램을 보고 스무싱

참고) Conv Layer는 1D Conv(5 x 1) + Batch Norm + ReLU



multi-speaker의 구현

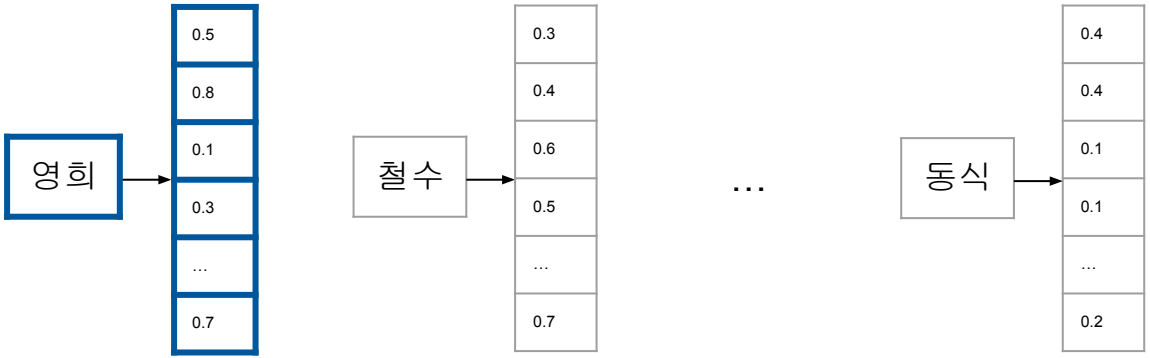
Deep Voice 2:
Multi-Speaker Neural Text-to-Speech

Baidu Research에서 2017년에 제시

스피커 임베딩(speaker embedding)을 통해 구현

화자별 특징을 숫자로 표현

Tacotron 중간 중간 임베딩 값이 들어가서 각기 다른 연산



multi-speaker TTS의 장점

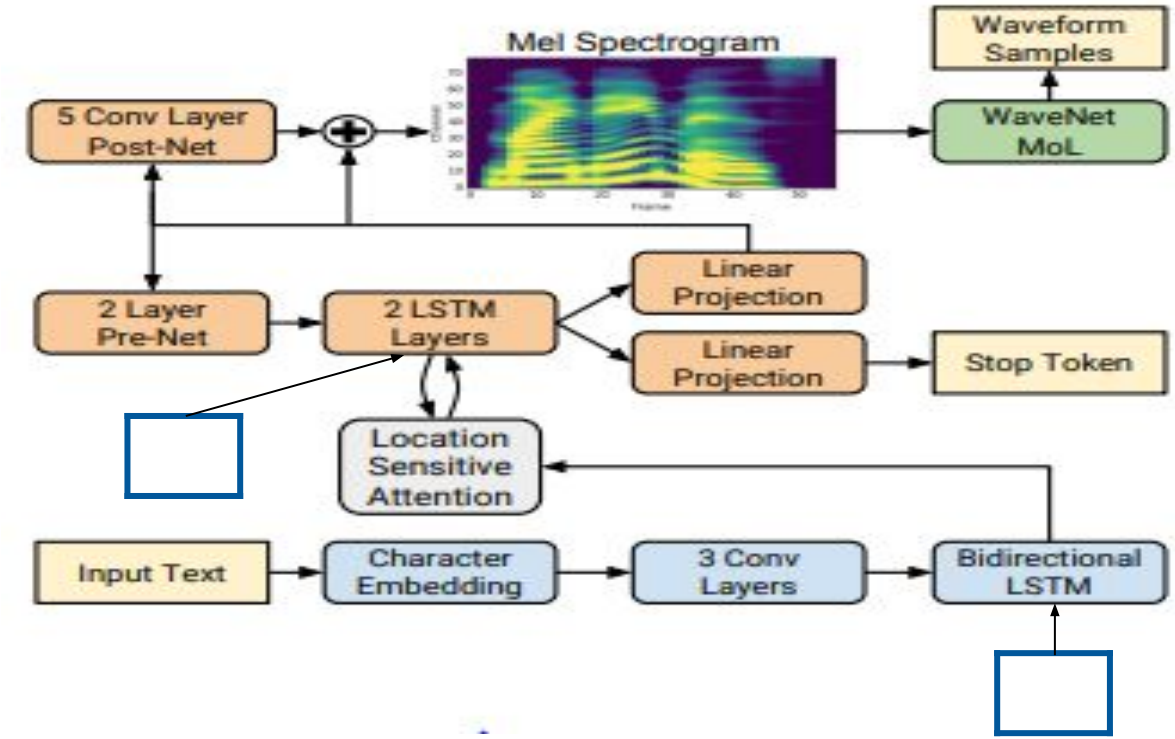
1. GPU 용량 효율성
   1개의 Tacotron이 3(GB), 20명의 목소리를 학습?
   → 20개의 Tacotron모델, 60(GB) 필요

   그러나 Speaker Embedding을 이용하게되면
   → 3(GB)에 근접하지만 조금 더 많은 용량으로 학습

2. 학습 효과 개선
   완벽하지 못한 데이터들이 어텐션을 생성하는데
   완벽한 데이터들이 도움을 줌

   충분하지 못한 데이터들도 원활하게 학습, 어텐션을 생성

multi-Speaker Tacotron2의 구조

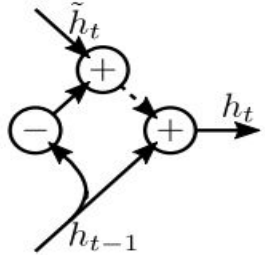


Zoneout LSTM

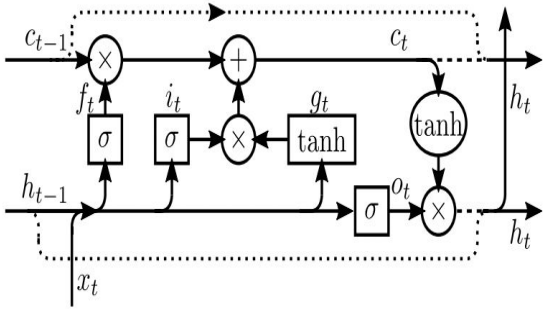
ZONEOUT: REGULARIZING RNNS BY RANDOMLY PRESERVING HIDDEN ACTIVATIONS(2016)

Dropout: 현재 값의 일부를 0으로

Zoneout: 현재 값의 일부를 이전 값으로



- 그레디언트 정보와 상태 정보가 시간에 지남에 따라 잘 유지
- 그레디언트 소멸문제를 해결하는데 도움이 된다고 함



$$c_t = d_t^c \odot c_{t-1} + (1 - d_t^c) \odot \left(f_t \odot c_{t-1} + i_t \odot g_t \right)$$

$$h_t = d_t^h \odot h_{t-1} + (1 - d_t^h) \odot \left(o_t \odot \tanh \left(f_t \odot c_{t-1} + i_t \odot g_t \right) \right)$$