Tacotron

분류 survey

```
Wang, Yuxuan, et al. "Tacotron: Towards end-to-end speech synthesis." arXiv preprint arXiv:1703.10135 (2017).
```

01. Abstract

- 문자에서 음성을 직접 합성하는 end-to-end 생성 TTS 모델
- [텍스트, 오디오] 쌍이 주어지면 무작위 초기화를 통해 모델을 처음부터 완전히 훈련
- 프레임 레벨에서 음성을 생성하므로 sample-level autogressive methods보다 빠름.

02. Introduction

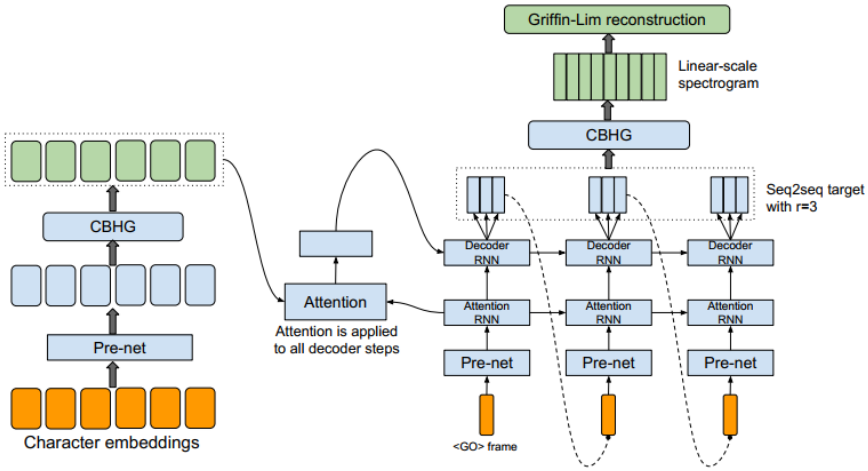
- statistical parametric TTS
  - 다양한 언어적 특징을 추출하는 텍스트 frontend, duration model, 음향 특징 예측 모델, 신호 처리 기반 vocoder를 가짐.
  - 광범위한 도메인 전문 지식을 기반으로 하며 설계가 어려움.
  - 또한, 독립적으로 학습되어 각 구성 요소의 오류가 복합될 수 있음.
- many advantages of an integrated end-to-end TTS system
  - [텍스트, 오디오] 쌍에 대해 학습
  - 휴리스틱 및 깨지기 쉬운 디자인 선택을 포함할 수 있는 엔지니어링 필요성 완화
  - 화자나 언어와 같은 다양한 속성 또는 감정과 같은 높은 수준의 기능에 대한 풍부한 조건을 더 쉽게 허용
  - 새로운 데이터에 대한 적응이 쉬울 수 있음.
  - 단일 모델은 각 구성 요소 오류가 복합될 수 있는 다단계 모델보다 더 강력할 수 있음.
    - 노이즈가 많은 데이터를 엄청나게 많이 훈련할 수 있음.
- TTS problems
  - 고도로 압축된 텍스트가 오디오로 압축 해제
  - 동일한 텍스트가 다른 발음이나 말하기 스타일에 해당할 수 있기 때문에 어려움 존재
    - 주어진 입력에 대한 신호 수준의 큰 변화에 대처해야 함.
  - end-to-end TTS 또는 기계 번역과 달리 TTS의 출력은 연속적
  - 출력 시퀀스는 일반적으로 입력 시퀀스보다 훨씬 길다.
    - 이러한 특성으로 예측 오류가 빠르게 누적
- 본 논문에서는 attention 패러다임과 함께 seq2seq 기반의 end-to-end 생성 TTS 모델 제안
  - 문자를 입력으로 사용하고 vanilla seq2seq 모델의 기능을 개선하기 위해 여러 기술을 사용하여 원시 스펙트로그램 출력
  - [텍스트, 오디오] 쌍이 주어지면 무작위 초기화를 통해 처음부터 완전히 학습
  - phoneme(음소)-level 정렬이 필요하지 않으므로 대본과 함께 대량의 음향 데이터를 사용
- produce
  - MOS(Mean Opinion Score) : 3.82 in US English eval set

03. Related Work

- WaveNet(2016)
  - 오디오 생성 모델
  - TTS에는 잘 작동하지만 sample-level autogressive 특성으로 인해 속도 저하
  - 기존 TTS frontend의 언어 특징에 대한 conditioning이 필요하므로 end-to-end가 아님.
  - vocoder와 음향 모델만 대체
- DeepVoice(2017)
  - 일반적인 TTS 파이프라인의 모든 구성 요소를 해당 신경망으로 대체
    - 그러나 각 구성 요소는 독립적으로 훈련
  - Wang et al.(2016)은 seq2seq를 사용하여 end-to-end TTS를 다룬 최초의 모델
    - 그러나 정렬을 학습하는 데 도움이 되도록 pre-trained hidden Markov model (HMM) 요구
  - 모델을 훈련시키기 위해 몇 가지 트릭이 사용되고 이는 운율(prosody) 손상 야기
  - vocoder 매개변수를 예측하므로 vocoder 필요
- Char2Wav(2017)
  - 문자에 대해 학습할 수 있는 독립적으로 개발된 end-to-end 모델
  - SampleRNN(2016)의 neural vocoder를 사용하기 전에 vocoder 매개변수를 예측하지만, 본 논문은 원시 스펙트로그램을 직접 예측
  - seq2seq 및 SampleRNN 모델은 별도록 pretrained 필요
  - vanilla seq2seq 모델은 char-level 입력에 대해 잘 작동하지 않음.

04. Model Architecture

- backbone : seq2seq model with attention (2014)
- High-level에서 문자를 입력으로 사용하고 스펙트로그램 프레임을 생성한 다음 파형으로 변환

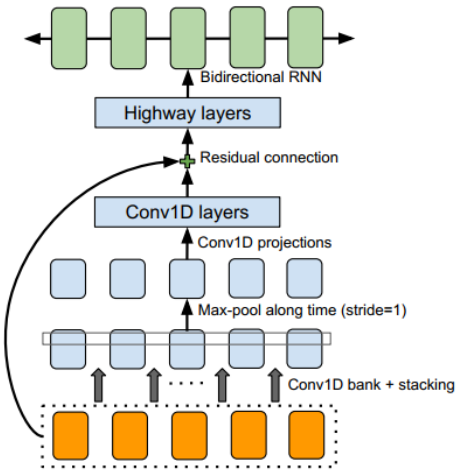


Encoder, Attention-based Decorder and Post-processing net

04-01. CBHG Module

- 1D conv filter, highway networks, Bi-GRU로 구성
- sequence에서 표현을 추출
- 입력 시퀀스는 K 세트의 1D conv filter를 거치게 된다.

    - `k` 크기의 필터가 여러개 포함
    - 로컬 및 컨텍스트 정보 모델링(akin to unigrams, bigrams, K-grams)
- conv 출력은 쌓이고 시간에 따라 추가로 풀링되어 로컬 불변성 증가
- 시간 해상도 유지를 위해 `stride=1`
- 처리된 시퀀스를 몇 개의 고정된 크기의 1D CNN으로 전달
- 출력은 residual-conn
- BN은 모든 CNN 레이어에 사용



The CBHG (1-D convolution bank + highway network + bidirectional GRU) module

- CNN 출력은 다음 네트워크의 입력으로 사용되어 high-level 특징 추출
- 마지막으로 Bi GRU RNN에 통과시켜 정방향 및 역방향 컨텍스트에서 순차적으로 특징 추출
- Non-causal CNN, BN, Residual-conn, stride=1, max-pool 사용 포함

04-02. Encoder

- 텍스트로부터 robust sequential 표현 추출
- 인코더의 입력은 각 문자가 one-hot 벡터로 표현되고 연속 벡터에 포함되는 문자 시퀀스
- **called pre-net**, 비선형 변환을 각 임베딩에 적용
    - dropout이 있는 bottlenet 구조를 pre-net으로 사용하여 수렴에 도움이 되고 일반화 능력 향상
- 다음 층인 CBHG 모듈은 출력을 attention 모듈의 입력인 최종 인코더 특징으로 변환
- 실험을 통해 CBHG 기반 인코더가 과적합을 줄이고, 다층 RNN 보다 발음 오류가 적었음.

04-03. Decoder

- content-based tanh attention decoder[Vinyals et al. (2015)] 사용
- stateful recurrent layer는 각 디코더 시간 단계에서 어텐션 쿼리 생성
- 컨텍스트 벡터와 attention RNN 셀 출력을 연결하여 디코더 RNN에 대한 입력 형성
- vertical residual connection이 있는 GRU 스택 사용
    - 수렴 속도 향상
- 원시 스펙트럼을 직접 예측할 수 있지만 음성 신호와 텍스트 사이의 정렬을 학습하기 위한 것으로 매우 중복된 특징
    - 이러한 중복성으로 인해 seq2seq 디코딩 및 파형 합성에 다른 대상 사용
    - seq2seq는 충분한 intelligibility(명료도) 및 prosody(운율) 정보를 제공하면 고도로 압축 가능

- 80-band mel-scale spectrogram 사용
- post-processing network를 사용하여 seq2seq 타겟에서 파형으로 변환
- 디코더 대상을 예측하기 위해 FC 사용
- 각 디코더 단계에서 겹치지 않는 여러 출력 프레임 예측
  - 한번에 `r` 개의 프레임을 예측하면 총 디코더 단계의 수를 `r` 로 나누게 됨
  - 모델의 크기, 학습 시간, 추론 시간 전부 감소
  - attention에서 훨씬 더 빠르고 안정적으로 수렴
  - 저자는 인접한 음성 프레임이 상관되어 있고 각 문자가 일반적으로 여러 프레임에 해당되기 때문이라고 설명
- 프로세스
  - `r` 시점에서 `r` 예측의 마지막 프레임이 단계 `t+1` 에서 디코더에 입력으로 전파
  - 훈련하는 동안 항상 모든 `r` 번째 GT 프레임을 디코더에 전달
  - 입력 프레임은 pre-net으로 전달

04-04. Post-processing Net

- seq2seq 타겟을 파형으로 합성할 수 있는 타겟으로 변환하는 것
- Griffin-Lim을 synthesizer로 사용
  - 선형 주파수 스케일에서 샘플링된 스펙트럼 크기를 예측하는 방법 학습
- 디코딩된 전체 시퀀스를 볼 수 있음.
  - 항상 왼쪽에서 오른쪽으로 실행되는 seq2seq와 달리 각 개별 프레임에 대한 예측 오류를 수정하기 위해 전방 및 후방 정보를 모두 가지고 있음.

04. Details

- log magnitude spectrogram
  - Hann windowing,
  - 50 ms frame length,
  - 12.5 ms frame shift
  - 2048-point Fourier transform.
  - 24 kHz sampling rate
- `r=2` (output layer reduction factor)
- LR decay
- L1 loss

Spectral analysis	*pre-emphasis*: 0.97; *frame length*: 50 ms; *frame shift*: 12.5 ms; *window type*: Hann
Character embedding	256-D
Encoder CBHG	*Conv1D bank*: K=16, conv-k-128-ReLU *Max pooling*: stride=1, width=2 *Conv1D projections*: conv-3-128-ReLU $\rightarrow$ conv-3-128-Linear *Highway net*: 4 layers of FC-128-ReLU *Bidirectional GRU*: 128 cells
Encoder pre-net	FC-256-ReLU $\rightarrow$ Dropout(0.5) $\rightarrow$ FC-128-ReLU $\rightarrow$ Dropout(0.5)
Decoder pre-net	FC-256-ReLU $\rightarrow$ Dropout(0.5)$\rightarrow$ FC-128-ReLU $\rightarrow$ Dropout(0.5)
Decoder RNN	2-layer residual GRU (256 cells)
Attention RNN	1-layer GRU (256 cells)
Post-processing net CBHG	*Conv1D bank*: K=8, conv-k-128-ReLU *Max pooling*: stride=1, width=2 *Conv1D projections*: conv-3-256-ReLU $\rightarrow$ conv-3-80-Linear *Highway net*: 4 layers of FC-128-ReLU *Bidirectional GRU*: 128 cells
Reduction factor (r)	2

Hyper-parameters and network architectures