MelGAN

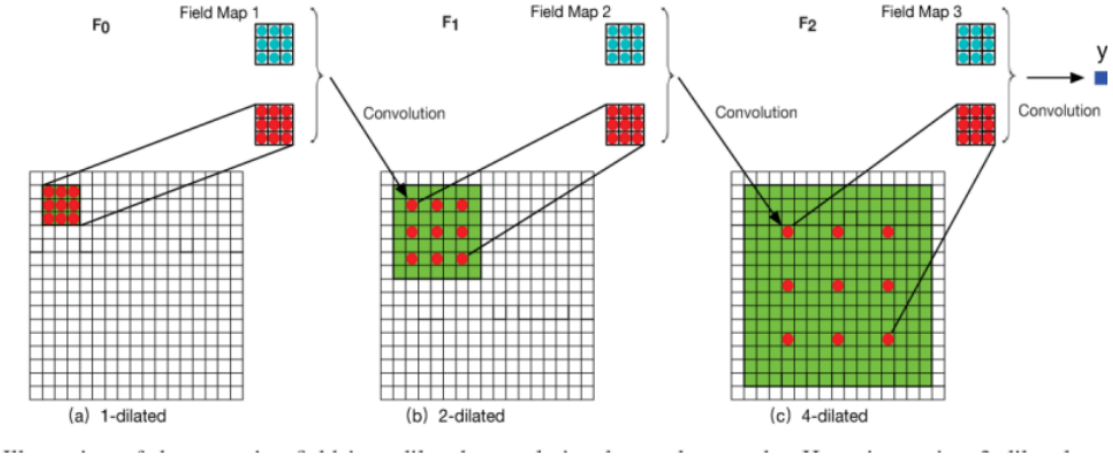
질문 리스트

1. 논문 p.3 “Induced receptive field”에 보면 아래와 같은 표현이 등장합니다. Receptive field의 의미를 정의하고 dilated convolution layers는 어떻게 receptive field를 efficient 하게 증가시키는지 설명해 주세요.

> Receptive field of a stack of dilated convolution layers increases
> exponentially with the number of layers. Similar to Van Den Oord et al. (2016), incorporating these
> in our generator allows us to
> **efficiently** increase the induced receptive fields of each output time-step...

답변 - receptive field는 입력 데이터에서 영향을 받는 부분, 픽셀에 대한 것임.
kernel_size가 커질수록 넓은 범위에서 영향을 받게 됨.

dilation을 한 마디로 말하자면 convolution 커널의 간격을 의미한다.



출처: https://www.semanticscholar.org/paper/Deep-Dilated-Convolution-on-Multimodality-Time-for-Xi-Hou/afadf82529110fadcbbe82671d35a83f334ca242

따라서 레이어가 늘어날수록 receptive field가 넓어지며 더 넓은 영역을 확인하게 됨. 따라서 동일 자원을 가지고 더 넓은 범위를 볼 수 있으며, residual을 통해서 더 긴 sequential 데이터를 볼 수 있음

2. MelGAN은 Variable length Mel Spectrogram input을 있는 그대로 처리할 수 있을까요? AST에서 Padding, Trimming 등을 이용해야 했던 것처럼 MelGAN도 그런 전처리를 거쳐야 할까요?

답변 - 길이에 대한 전처리가 필요할 것이라고 생각됨. convolution 레이어, 모양의 크기는 고정되어 있는데, 길이에 따라 stride, dilation rate를 조정하면 길이에 따라 데이터가 적당히 사용되지 않을 수 있다고 생각됨. 따라서 padding, trimming을 통한 데이터의 전처리가 필요하다고 생각함.

3. 논문 p.4 Discriminator 부분에 등장하는 아래 단락을 이해하고, 설명해 주세요. (Optional)

> This structure has an inductive bias that each discriminator learns features for different frequency range of the audio. For example, the discriminator operating on downsampled audio, does not have access to high frequency component, hence, it is biased to learn discriminative features based on low frequency components only.

답변 - 3종류의 분류기를 각자 다른 해상도를 가진 오디오 데이터를 가지고 분류하게 만든다는 의미. 원 해상도(D1)는 고주파수까지 선명하기에 고주파수, 2배(D2)는 그 절반까지, 4배(D3)는 그 절반까지만의 데이터를 가지고 주파수에 대한 데이터를 학습하게 됨

개요

- 일반적인 오디오를 생성하는것은 수많은 갯수의 일시적 데이터의 집합임
    - 따라서 raw temporal audio의 직접 모델링 대신 raw temporal signal에서 계산되는 lower-resolution representaion을 활용함

- 입력으로 텍스트를 받고 중간과정을 거친후, 오디오로 변경
    - 중간 과정으로 멜스펙트로그램으로 표현
        - 순수 신호 프로세싱
            - Griffin-Lim을 사용하여 STFT 시퀀스를 통해 temporal signal로 디코딩
            - world 보코더는 mel-spectrogram-like feature를 사용해 음성 모델링
            - 문제는 중간 요소에서 오디오를 만드는 과정에서 noticable artifact 발생
        - 자기회귀 형태의 뉴럴 네트워크
            - wavenet은 자기회귀 형태의 음성 요소를 활용해 음성합성을 하는 fully-convolution autoregressive sequence모델
            - sample RNN은 multi-scale recurrent network모델을 사용하여 unconditional waveform을 합성
            - 문제는 autoregressive 방식은 추론 때 audio sample을 sequential하게 생성하므로 느리고 비효율적
        - 비자기회귀 형태의 뉴럴 네트워크
            - 비자기회귀 방식은 병렬적이라 자기회귀방법보다 빠름
                - parallel wavenet, clarinet은 trained autoregressive decoder를 student model로 distil함
                - waveglow는 flow-based generation 도입
        - GANs for Audio
            - GAN을 활용해 오디오 모델링 수행
                - 하지만 지금까지의 GAN기반 오디오 모델링은 성능이 충분하지 않음

MelGAN

- GAN을 통한 웨이브폼 생성을 위해 non-autoregressive feed-forward convolution 구조 도입
- 추가적인 지식 증류를 통해 perceptual loss없이 음성 합성

    - percptual loss - 인지 손실함수
- 음악, tts등의 다양한 방면으로 melgan 확장

전체 구조

- mel spectrogram inversion을 위해 GAN에 대한 비자기회귀 convolution 구조 도입
- 간단한 training technique 도입
    - 결과적으로 더 적은 파라미터 수
    - 빠른 추론 속도 가짐

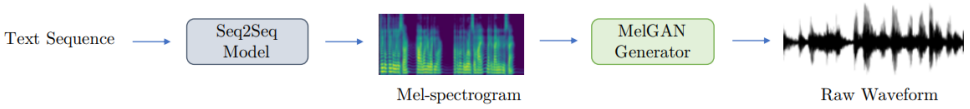
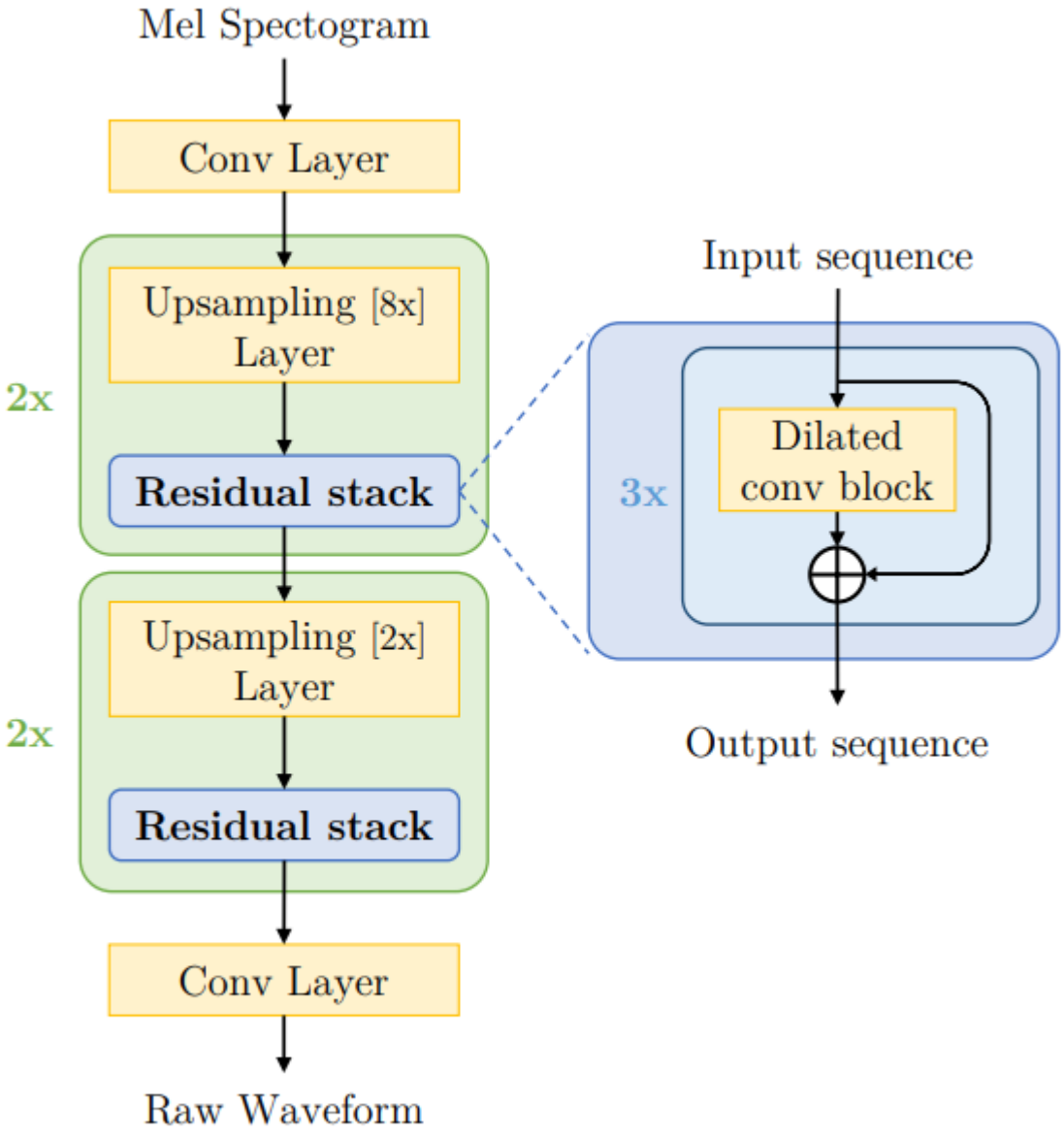


Figure 2: Text-to-speech pipeline.

생성기, 분류기 구조

- 생성기



(a) Generator

- 입력값 - melspectrogram s
- 출력값 - raw waveform x
- 생성기
  - mel-spectrogram이 256배 더 낮은 temporal resolution을 가지기에 입력값을 업샘플링할 필요가 있음
  - 따라서 transposed convolution의 stack을 사용함

    - 각 transposed convolution layer 이후, dilated convolution이 있는 residual block사용
    - 원본 생성기와 다르게 global noise 벡터는 없음
    - 추가적인 노이즈가 존재해도 인지 차이가 크지는 않음
    - 입력값→출력값의 관계에는 loss값이 이미 포함되어 있음
        - one to many 관계의 매핑이 되어있음
        - 따라서 직관에 반하는 결과(?)
- induced receptive field
    - 이미지 대상의 컨볼루션 네트워크
        - 공간적으로 가까운 픽셀들은 서로 상관관계가 존재한다는 개념
        - 컨볼루션을 통해 줄어드는 과정에서, 하나의 픽셀의 값이 여러종류의 계산에 사용되여 연관성을 가지게 된다는 것
    - 따라서 오디오의 경우, 장거리 연관성이 있다는 내용을 활용할 수 있게 생성기 구조 설계
        - 따라서 업샘플링 이후에 residual을 활용해 연관성을 가지게 해야함
- checkerboard artifacts(패턴 일정x, 격자무늬 형태)
    - 컨볼루션 네트워크의 kernel_size, stride를 잘 선택해야 함
    - 히싱 노이즈(고주파수 쉬 소리) 발생 가능
- normalization technique
    - 샘플 품질을 향상하는데 중요함
    - 이미지 생성에서는 instance normalization 사용
    - 하지만 오디오는 pitch 정보를 지워내 metalic 사운드 생성
        - weight normalization을 사용하여 분류기의 용량을 제한, activation의 normalization을 막음
            - weight vector의 scale을 분리해서 가중치 행렬로 reparameterize 함
            - 모든 생성기에서 weight normalization 수행

분류기

- 멀티 스케일 구조
  - 동일한 네트워크 구조
  - 서로 다른 오디오 스케일에서 동작하는 D1, D2, D3를 가지는 멀티 스케일 구조
    - kernel_size=4, stride는 average pooling 적용
  - D1 - 원본
  - D2 - 2배 다운샘플링 된 샘플
  - D3 - 4배 다운샘플링 된 샘플
  - 각 분류기별로 다양한 주파수 대역을 학습할 수 있게 됨
    - 다운샘플링을 시행하면 고주파수 대역 특징 추출이 어려움
      - 따라서 저주파수 대역만 가지고 분류 특징 추출
- Window based objective
  - 각 분류기는 큰 kernel_size를 가지는 makrovian window based 분류기
    - 파라미터 갯수를 작게 유지하면서 더 큰 kernel_size를 사용할 수 있게 만듦
  - 일반적으로는 전체 audio sequence 분포의 classify 학습
    - 하지만 window based 분류기는 audio chunk간의 분포를 학습
    - 따라서 전체 시퀀스의 일관성 유지
  - 따라서 해당 방법은 유효 고주파수를 빠르게 추적함
    - 빠르고 적은 수의 파라미터로 다양한 길이의 오디오에 적용가능함

training objective

- GAN의 학습을 위해 GAN objective에 대한 hinge loss활용
  - hinge loss - 이진 분류 문제, 데이터의 올바른 분류여부 loss 값
    - 실제 클레스와 떨어진 정도에 따라 패널티 부여

$$\min_{D_k} \mathbb{E}_x \Big[\min(0, 1 - D_k(x)) \Big] + \mathbb{E}_{s,z} \Big[\min(0, 1 + D_k(G(s,z))) \Big], \ \forall k = 1,2,3 \tag{1}$$

$$\min_{G} \mathbb{E}_{s,z} \left[\sum_{k=1,2,3} -D_k(G(s,z)) \right] \tag{2}$$

    - 첫번째 식
        - 진짜 데이터 x에 대해 높은 값 부여, 가짜 데이터 G(s, z)에 대하여 낮은 값
    - 두번째 식
        - 생성자는 분류기가 가짜 데이터를 진짜로 판단하도록 학습됨
- Feature Matching Loss
    - 분류기 signal 이외에도 feature matching objective 적용
    - 실제와 합성 audio간의 discriminator feature map 간의 L1 거리(맨해튼 거리) 최소화
    - 원본에는 어떠한 LOSS도 사용하지 않음
        - L1을 추가하면 오디오 품질을 저하하는 audible noise(들을 수 있는 노이즈) 발생

$$\mathcal{L}_{\text{FM}}(G, D_k) = \mathbb{E}_{x,s\sim p_{\text{data}}}\left[\sum_{i=1}^{T}\frac{1}{N_i}||D_k^{(i)}(x) - D_k^{(i)}(G(s))||_1\right]$$

    - Dk(i) - k번째 분류기의 i번째 feature map
    - N은 레이어 갯수
    - L1거리를 활용해 실제값, 생성값 사이의 차이 계산
- 최종식은 아래와 같음
    - λ는 10으로 지정

$$\min_G\left(\mathbb{E}_{s,z}\left[\sum_{k=1,2,3}-D_k(G(s,z))\right] + \lambda\sum_{k=1}^{3}\mathcal{L}_{\text{FM}}(G, D_k)\right)$$

기존 모델들보다 파라미터 수로 압도적으로 효율적임

Table 1: Comparison of the number of parameters and the inference speed. Speed of n kHz means that the model can generate $n \times 1000$ raw audio samples per second. All models are benchmarked using the same hardware [3].

Model	Number of parameters (in millions)	Speed on CPU (in kHz)	Speed on GPU (in kHz)
Wavenet (Shen et al., 2018)	24.7	0.0627	0.0787
Clarinet (Ping et al., 2018)	10.0	1.96	221
WaveGlow (Prenger et al., 2019)	87.9	1.58	223
MelGAN (ours)	**4.26**	**51.9**	**2500**

학습 결과

- abalation study(절제 학습)
  - 생성기에서 dilated convolution 또는 weight normalization을 제거하는 경우, highfrequency artifact가 발생
    - 특히 multi-scale discriminator 없애면 metalic audio 생성됨
      - dilated convolution - 장거리 연관성 없어짐
      - weight normalization - 안정적 수렴 불가, 고주파수 성능 저하
  - spectral normalization 사용, window-based discriminator loss를 제거하는 경우,

    sharp high frequency pattern 학습 불가
      - 노이즈 생성
        - spectral normalization, window based discriminator loss
          - 고주파수 성분을 못 잡아냄

Model	MOS	95% CI
w/ Spectral Normalization	1.33	±0.07
w/ L1 loss (audio space)	2.59	±0.11
w/o Window-based Discriminator	2.29	±0.10
w/o Dilated Convolutions	2.60	±0.10
w/o Multi-scale Discriminator	2.93	±0.11
w/o Weight Normalization	3.03	±0.10
Baseline (MelGAN)	**3.09**	± **0.11**

benchmarking competing models

- 다른 모델과 비교했을때, melgan이 우수한 성능을 보임

Table 3: Mean Opinion Scores

Model	MOS	95% CI
Griffin Lim	1.57	±0.04
WaveGlow	4.11	±0.05
WaveNet	4.05	±0.05
MelGAN	3.61	±0.06
Original	**4.52**	± **0.04**

generalization to unseen speakers

- unseen speakers에 대해서도 melgan이 우수했음

Table 4: Mean Opinion Scores on the VCTK dataset (Veaux et al., 2017).

Model	MOS	95% CI
Griffin Lim	1.72	±0.07
MelGAN	3.49	±0.09
Original	**4.19**	± **0.08**

end to end speech synthesis

- end to end 모델과 melgan 결합
- tacotron2, text2mel과 보코더 합쳐서 진행

Model	MOS	95% CI
Tacotron2 + WaveGlow	3.52	±0.04
Text2mel + WaveGlow	4.10	±0.03
Text2mel + MelGAN	3.72	±0.04
Text2mel + Griffin-Lim	1.43	±0.04
Original	4.46	± 0.04