WaveNet , auto encoder

Our method is based on a WaveNet [1] autoencoder. These autoencoders

were used to model single musical instruments [2]

and extended to perform translation between musical domains

in [3] by employing a single encoder and multiple decoders.

→ Autoencoder는 싱글 악기를 모형화하는데 사용되었고 single encode와 multiple decoders를 사용하며 음악 도메인간 번역에도 확장되었다.

Auto encoder

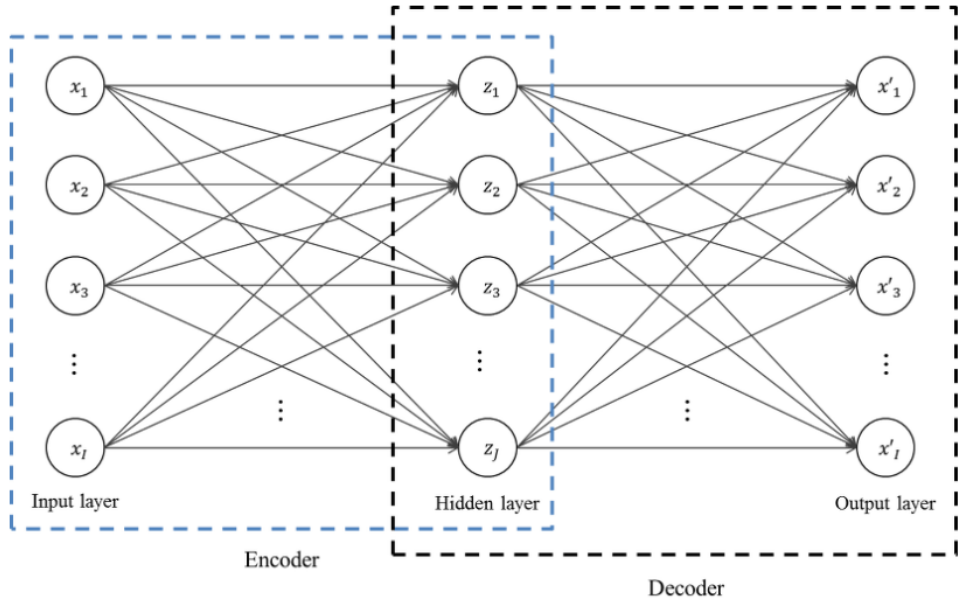
간단 요약) FNN의 구조와 매우 유사하며, 다른 점은 입력층과 출력층의 크기가 항상 같다.

NN의 unsupervised learning 버전

사용 목적) 데이터의 특징을 찾아내기 위하여(손실함수는 입력과 출력의 차이로 정의), 가중치 학습으로 데이터 특징 파악 가능

Auto encoder의 중요한 동작은 입력벡터의 차원을 축소하는 것

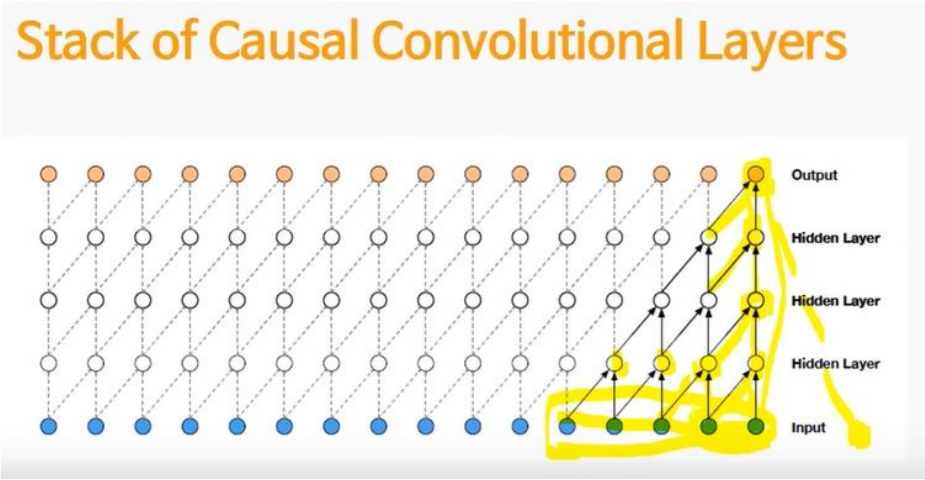
(은닉층이 1개인 auto encoder)



그렇담 waveNet 은 뭐지?

$$p(\mathbf{x}) = \prod_{t=1}^{T} p(x_t \mid x_1, \ldots, x_{t-1})$$

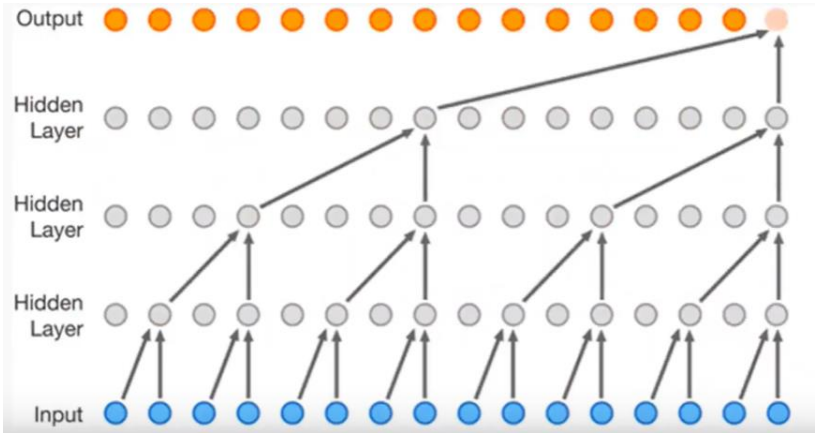
p(x1)
p(x1,x2)=p(x1)p(x2|x1)
p(x1,x2,x3)=p(x1,x2)p(x3|x1,x2)
=p(x1)p(x2|x1)p(x3|x1,x2)
...

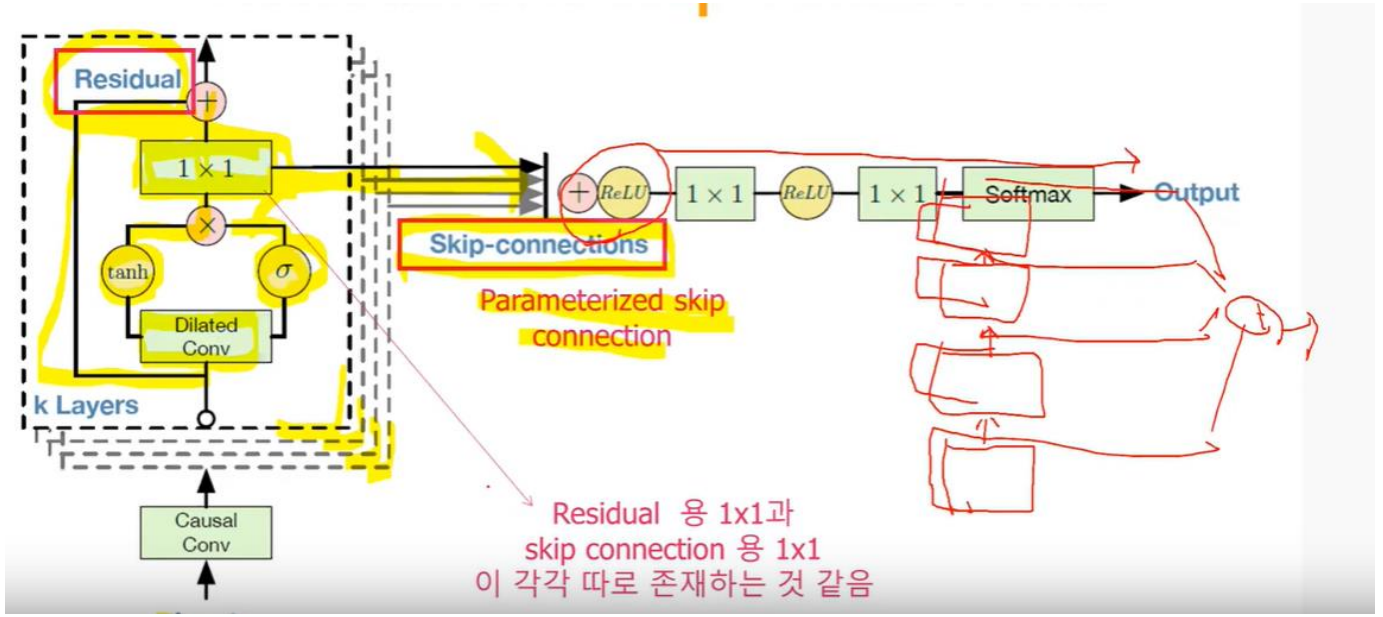


시간에 따른 음성 데이터를 stack cnn으로 쌓는다.

waveNet

- 마지막에 나온 아웃풋이 다음의 input으로 들어감
- Dilated (mask와 같은 느낌, 어느 곳에 집중할 수 있게) 중간중간 선택한다.





Weight gate를 거쳐서 1x1 conv한 뒤, 활성화 함수와 softmax를 거쳐 output 생성

WaveNet auto encoder

- 기존 waveNet 경우 짧은 오디오 신호에 대한 모델링에는 훌륭히 작동하지만 긴 오디오를 생성하는데 있어서는 외부로부터의 condition에 많은 영향을 받는 것으로 확인 (대충 한계가 있다는 듯)
- 하지만 우리 모델은 내부에 시간에 따른 hidden embedding 벡터를 가지고 오디오를 생성하기 때문에 외부로부터 condition될 필요가 없다.

- Long term structure를 유지하도록 해보자 !

WaveNet auto encoder

- 기존 wavenet 에서 external feature(언어학 변수? 부수적인 데이터에 대한 정보라고 이해했음) 의 필요를 Autoencoder가 대체

-> 입력으로 부터 latent vector, embedding vecto를 만들자

-> encoding된 embedding으로 다시 입력 x를 만들어내게 해보자

→ 입력에서 많은 정보를 뽑아내게 개선해보자

WaveNet auto encoder

- 기존 wavenet 에서 external feature(언어학 변수? 부수적인 데이터에 대한 정보라고 이해했음) 의 필요를 Autoencoder가 대체

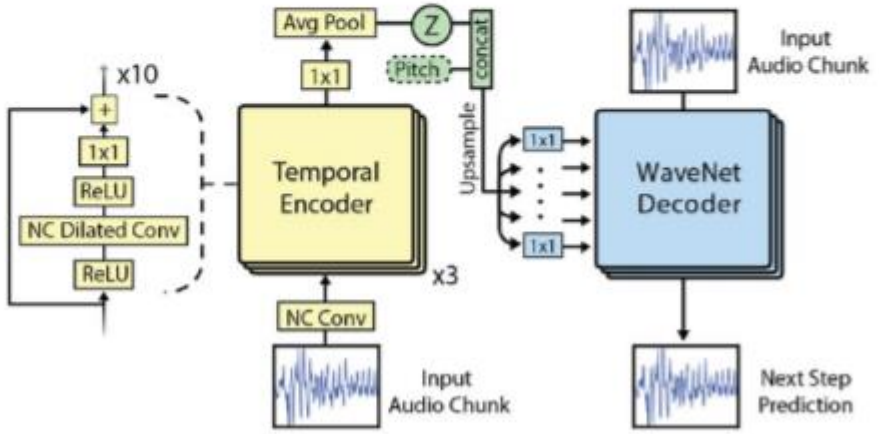
-> 입력으로 부터 latent vector, embedding vecto를 만들자

-> encoding된 embedding으로 다시 입력 x를 만들어내게 해보자

→ 입력에서 많은 정보를 뽑아내게 개선해보자

WaveNet auto encoder

- Goal: 오토인코더 구조를 사용해서 external condition 없이 시그널 데이터를 학습 및 생성
  → embedding layer가 역할을 대신함
- (WaveNet-like) Encoder: infers hidden embeddings distributed in time
- (WaveNet) Decoder: use the embeddings to effectively reconstruct the original audio



Decoder는 입력 embedding을 다시 만들어내는 역할을 함

WaveNet auto encoder

The translation is done without parallel data in a method that is similar to our first phase of training, except that we employ a single, singer conditioned, WaveNet decoder and a different data augmentation procedure. Most previous work that employ a WaveNet decoder that is conditioned on the embedding of the speaker [4, 5, 6], employ supervised learning, while we employ unsupervised learning.

-> 그래서 우리는 wave net decode가 unsupervised learning이게 사용했다.
→ Encoder 단에 wave net을 써서 waveNet auto encoder가 된 것 같음

VQ-VAE

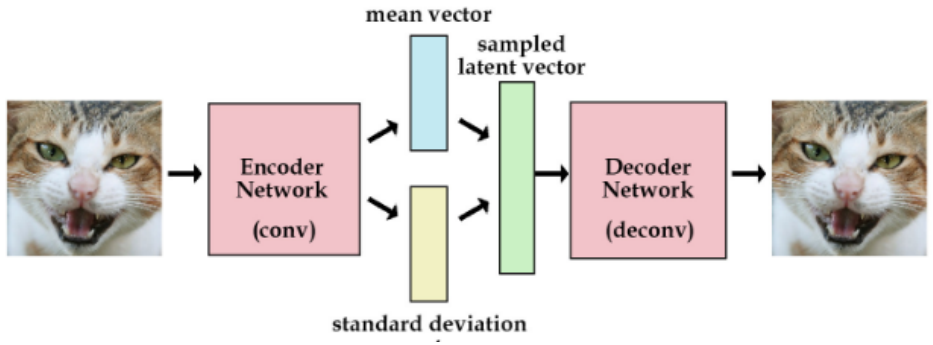
In the unsupervised VQ-VAE method [7], voice conversion
was obtained by employing a WaveNet autoencoder that produces
a quantized latent space. The decoder is conditioned on
the target speaker's identity, using a one-hot encoding.

→ Decoder단에 목적 speakers의 특징을 조건부로 두어서 decode를 하는구나..
→ Encoder로 input data 넣어주고 p(노래|target) target이 불렀을 땐 어떨까..이런 느낌

VQ-VAE(Vector Quantized Variational Autoencoders)

concept

VAE는 데이터가 생성되는 과정, 즉 데이터의 확률분포를 학습하기 위한 두 개의 뉴럴네트워크로 구성되어 있습니다. VAE는 잠재변수(latent variable) z를 가정하고 있는데요. 우선 *encoder*라 불리는 뉴럴네트워크는 관측된 데이터 x를 받아서 잠재변수 z를 만들어 냅니다. *decoder*라 불리는 뉴럴네트워크는 *encoder*가 만든 z를 활용해 x를 복원해내는 역할을 합니다. VAE 아키텍처는 다음 그림과 같습니다.



이를 잠재변수 z와 VAE 아키텍처 관점에서 이해해 보자면, *encoder*는 입력 데이터를 추상화하여 잠재적인 특징을 추출하는 역할, *decoder*는 이러한 잠재적인 특징을 바탕으로 원 데이터로 복원하는 역할을 한다고 해석해볼 수 있겠습니다. 실제로 잘 학습된 VAE는 임의의 z값을 *decoder*에 넣으면 다양한 데이터를 생성할 수 있다고 합니다.

VQ-VAE(Vector Quantized Variational Autoencoders)

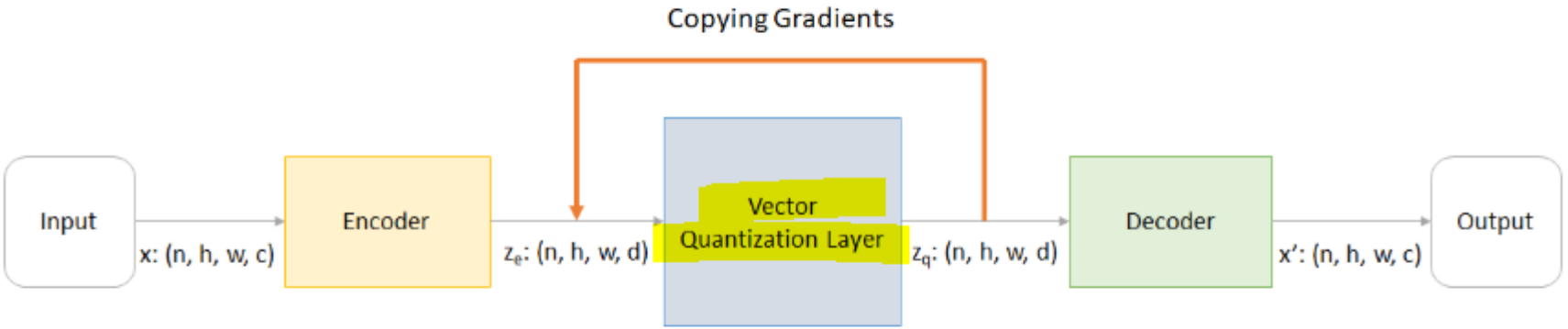


Fig 1: VQ-VAE Architecture

Vetor quantization layer가 붙어서 VQ-VAE인데, 이부분은 나중에 필요하게 되면 다시 공부하겠음..어렵다..

WGAN

Other autoencoder-based approaches in the field of voice conversion have relied on variational auto encoders [9] to generate spectral frames. In [10], the notion of a single encoder and a parameterized decoder, where the parameters represent the identity, was introduced. The method was subsequently improved [11] to include a WGAN [12] to improve the naturalness of the output (not as a domain confusion term).

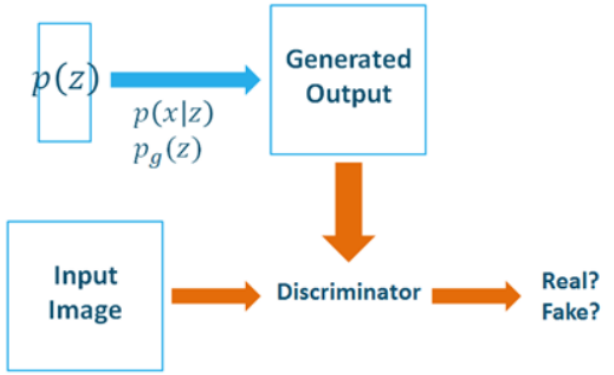
VAE-> OUPUT 을 자연스럽게 하도록 WGAN으로 개선되었다.

VAE vs GAN

VAE는?

- Variational Auto-Encoder
- 복잡한 데이터 생성 모델을 설계하고 대규모 set에 적응 할 수 있게 해줌
- input data를 잠재변수 z로 encoding 한 후, 스스로 input을 복원해 내는 방법
- VAE에서의 Loss functoion은 input x와 복원된 x'(decoding 된 x) 간의 Loss로 정의
- VAE에서는 Auto-Encoder가 input을 따라 그리는 것에만 맞게 학습 시킴
- 결론적으로 z는 의미론적이지 않음

GAN은?



- Generative Adversarial Nets
- Generator과 Discriminator가 서로 대립하여 서로의 성능을 점차 개선해 나가는 개념
- ex) 지폐위조범(Generator)은 경찰(Discriminator)을 최대한 열심히 속이기 위해 노력함
  경찰은 지폐위조범의 위조된 지폐를 감별하기 위해(Classify) 노력함
  이런 경쟁 속에서 두 그룹 모두 속이는 능력, 구별하는 능력이 발전하게 됨
  = 결과적으로, 진짜 지폐와 위조 지폐를 구별할 수 없을 정도에 이름

- Generative model G -> data x의 distibution을 알아내려고 노력
  (G가 data distribution을 모사할 수 있으면 sample과 data를 구별할 수 있다.)

- Discriminator model D -> sample이 training data인지, G가 만들어낸 data 인지 구별하여 각각의 확률을 estimate시킴

VAE vs GAN

결론적으로 VAE와 GAN의 차이점은?

GAN

-generator model의 목적 자체가 어떤 data의 분포를 학습하는 것이 아님
- 진짜 같은 sample을 generate하는 것이 목적

VAE
- data의 분포를 학습하고 싶은데, 이 data가 다루기 힘들기 때문에 variational inference(변화 추론)하는 방법
- VAE는 data 분포가 잘 학습되기만 하면 sampling (=data generation)이 저절로 따라옴

GAN 도 비지도 학습 GAN, 지도 학습 GAN이 있지만 VAE는 비지도 학습임
지도학습 GAN경우 generator 한 이미지와 정답으로 input준 이미지 중 discriminater가 뭐가 정답인지 판별하게 시킴으로써 모델을 최적화함
VAE는 그딴거 없음, 그냥 Z가 INPUT 학습하고 그걸 바탕으로 DECODER가 만들어냄 (데이터의 분포를 학습)

WGAN(Wasserstein GAN) / 거리를 재는 방법이 Wasserstein

GAN LOSS $\min_G \max_D E_{x\sim p(data)}[\log(D(X))] + E_{z\sim p(z)}[\log(1 - D(G(Z))]$

LOSS 학습 시키기 어렵다.

WGAN 특징

- discriminator대신 새로 정의한 critic을 사용한다. discriminator는 가짜/진짜를 판별하기 위해 sigmoid를 사용하고, output은 가짜/진짜에 대한 예측 확률 값이다.
- 반면 critic은 EM(Earth Mover) distance로부터 얻은 scalar 값을 이용한다.
- EM distance는 확률 분포 간의 거리를 측정하는 척도 중 하나인데, 그 동안 일반적으로 사용된 척도는 KL divergence이다. KL divergence는 매우 strict 하게 거리를 측정하는 방법이라서, continuous하지 않은 경우가 있고 학습시키기 어렵다.

https://kionkim.github.io/2018/06/01/WGAN_1/

WGAN(Wasserstein GAN) / 거리를 재는 방법이 Wasserstein

GAN LOSS $\min_G \max_D E_{x\sim p(data)}[\log(D(X))] + E_{z\sim p(z)}[\log(1 - D(G(Z))]$

LOSS 학습 시키기 어렵다.

WGAN 특징

- discriminator대신 새로 정의한 critic을 사용한다. discriminator는 가짜/진짜를 판별하기 위해 sigmoid를 사용하고, output은 가짜/진짜에 대한 예측 확률 값이다.
- 반면 critic은 EM(Earth Mover) distance로부터 얻은 scalar 값을 이용한다.
- EM distance는 확률 분포 간의 거리를 측정하는 척도 중 하나인데, 그 동안 일반적으로 사용된 척도는 KL divergence이다. KL divergence는 매우 strict 하게 거리를 측정하는 방법이라서, continuous하지 않은 경우가 있고 학습시키기 어렵다.

https://kionkim.github.io/2018/06/01/WGAN_1/
https://haawron.tistory.com/21

수식이 많아 다 설명하긴 어렵고 (사실 이해안되서 그런거임) WGAN 공부할 일 생기면 함께 공부합시다..