3-3
The architecture of the encoder, decoder, and confusion network mostly reuse the successful WaveNet autoencoder architecture

이게 핵심이다.

encoder, decoder, confusion network가
Wavenet autoencoder를 재사용한다는 점!

Wavenet autoencoder 개요

- Google Brain/DeepMind 가 만든 오디오 신호 합성 방법

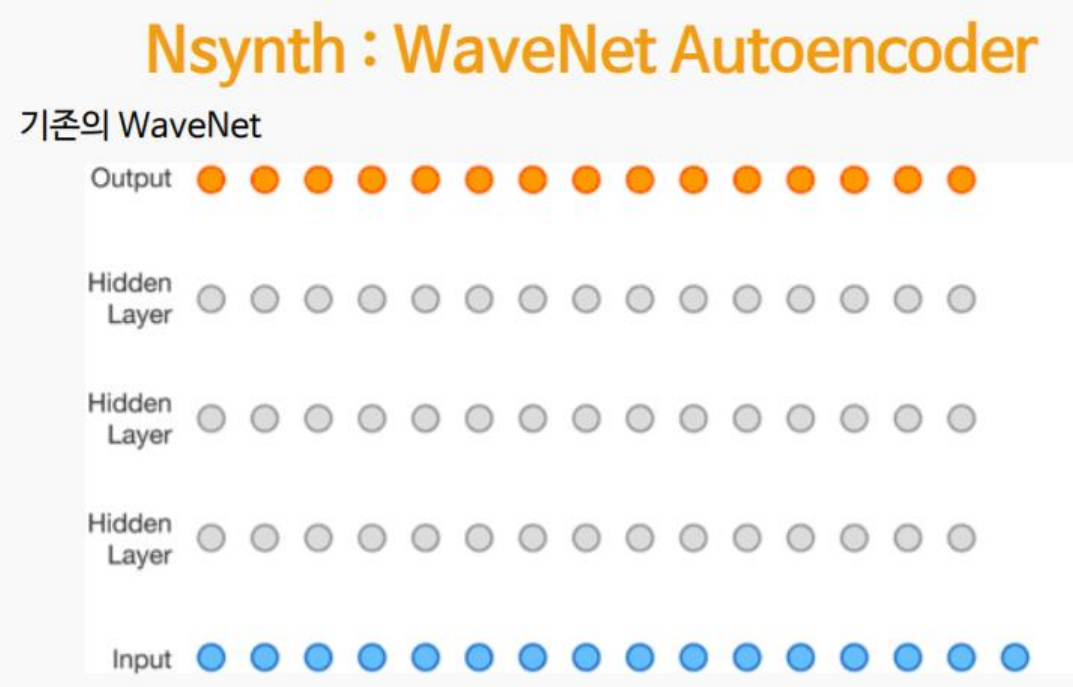
-> Audio용 Autoencoder를 제안

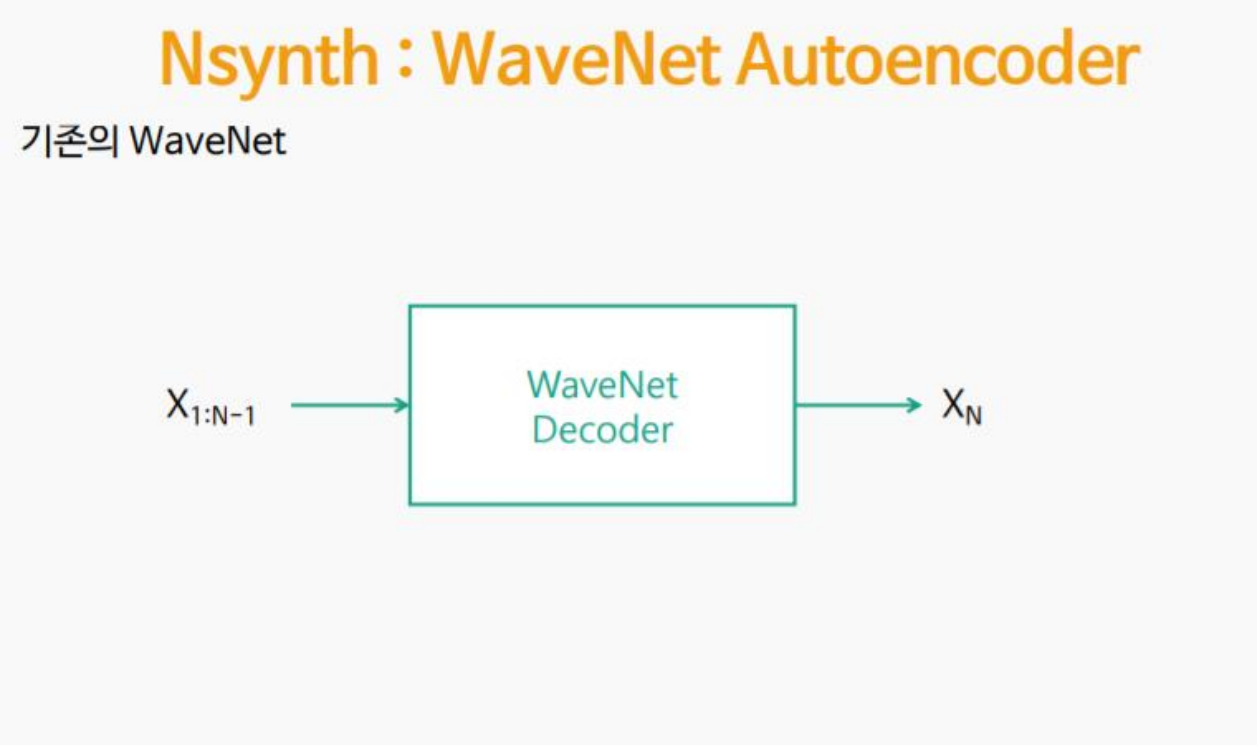
해결하고자 하는 문제 : WaveNet은 음악을 unconditional하게generation하면, babbling 과 같은 문제가 생김. (잡음 문제!)

1. External conditio을 주는 대신, 입력 x 로부터 condition(latent vector, embedding vector)을 만들어주면 어 떨까?
2. 이것을 학습하기 위해 encodin된 embedding으로부터 다시 입력 x를 만들어내게 해보자.

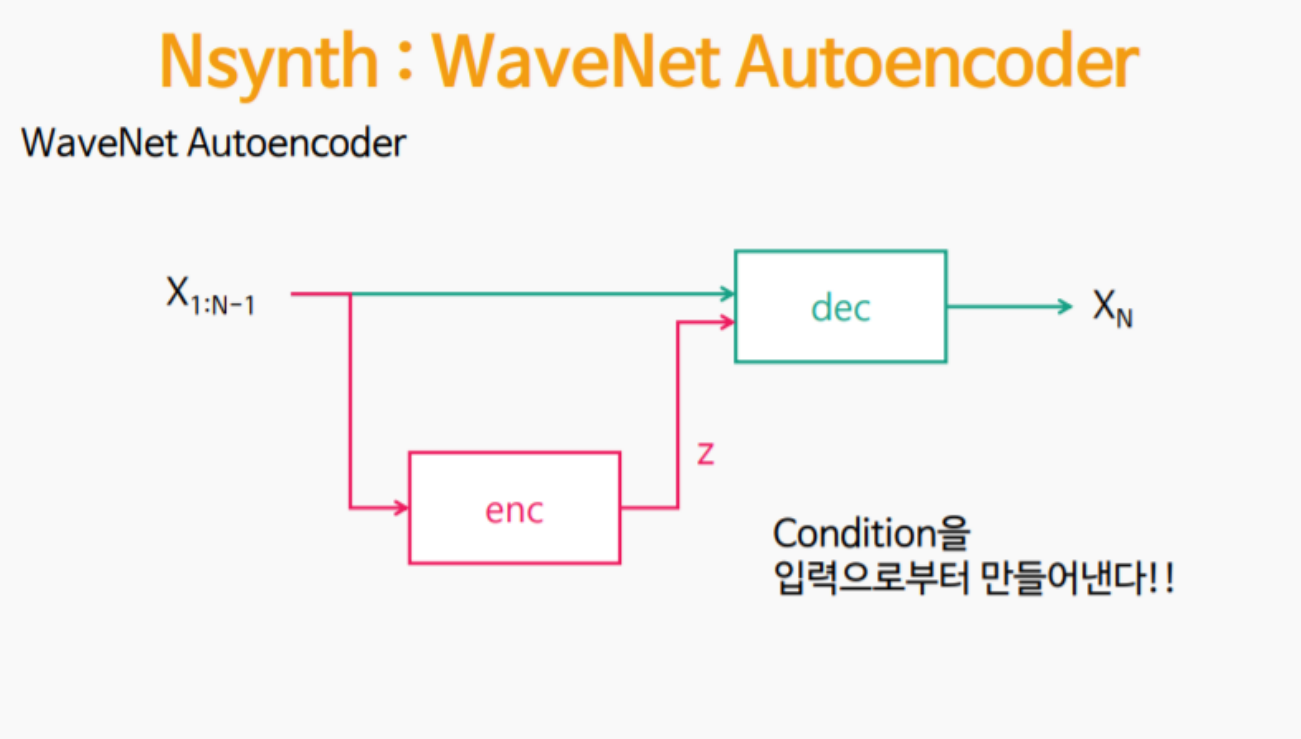
➔ WaveNet Autoencoder

Wavenet autoencoder 개요





Wavenet autoencoder 개요



Encoder

Encoder는 fully convolutional network

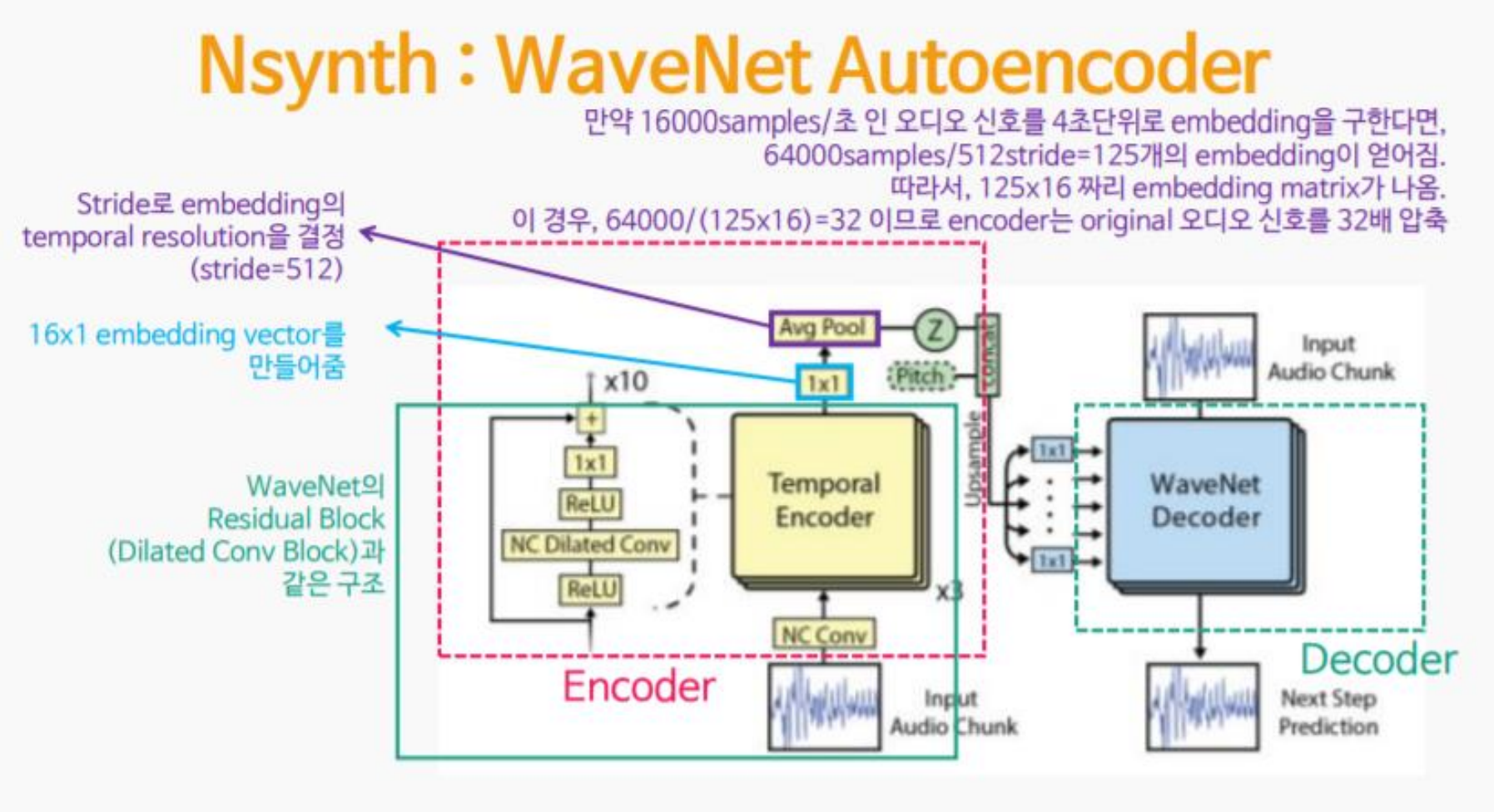
10개의 residual layer x 3 block = total 30
각각은 relu 활성화 함수 가지고 있고

30번 다하면 1x1 layer와 average pooling layer를 지남

kernel size는 50ms면 800 samples를 얻음

downsampling은 샘플링 주기를 낮추는 것
ex) [1, 2, 3, 4, 5, 6, 7] => [1, 4, 7]
12.5 정도로 down sampling을 함

Encoder 부분만 보면 residual block부터 avg pool까지 보면 된다.
논문에서 stride기준은 나와있지 않지만 50 x 16 = 800인것으로 보아
50ms가 encoder 결과값이고 그것을 10000samples/초를 인풋으로 넣어
12.5의 다운샘플링 계수(압축)를 얻은 듯 합니다!
10000/(50 x 16) = 12.5



decoder

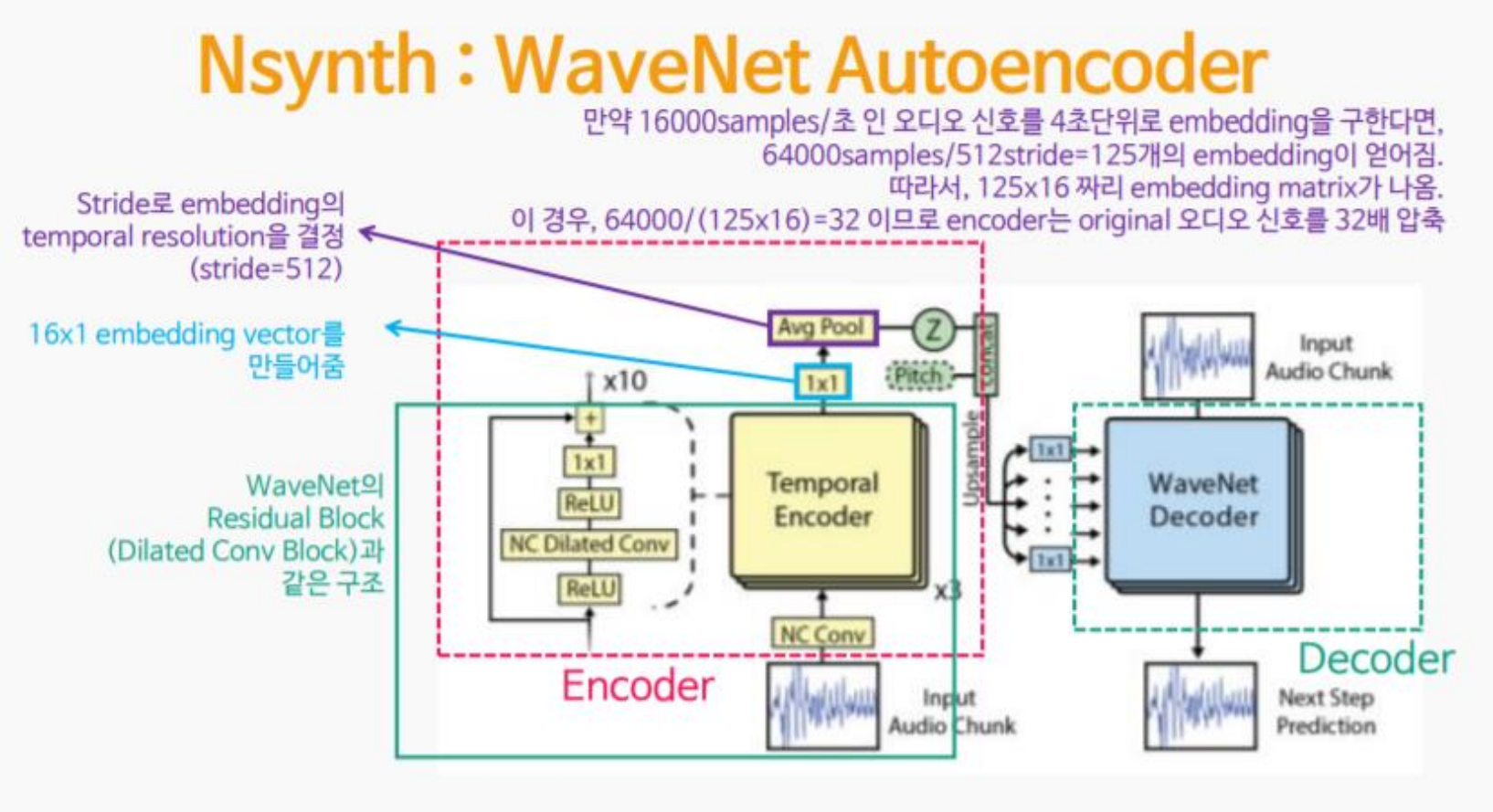
컨디션을 만들어 내기 위해

인코더로부터 주어진 벡터와 타겟 싱어의 임베딩(128차원)과 합쳐야한다.

이 벡터의 절반은 시간에 따라 변하고 반은 변하지 않는다.

인코딩은 잠시 upsampled된다.

decoder



Avg pool 다음에

Concat한 후

upsampled 되는 과정이다.

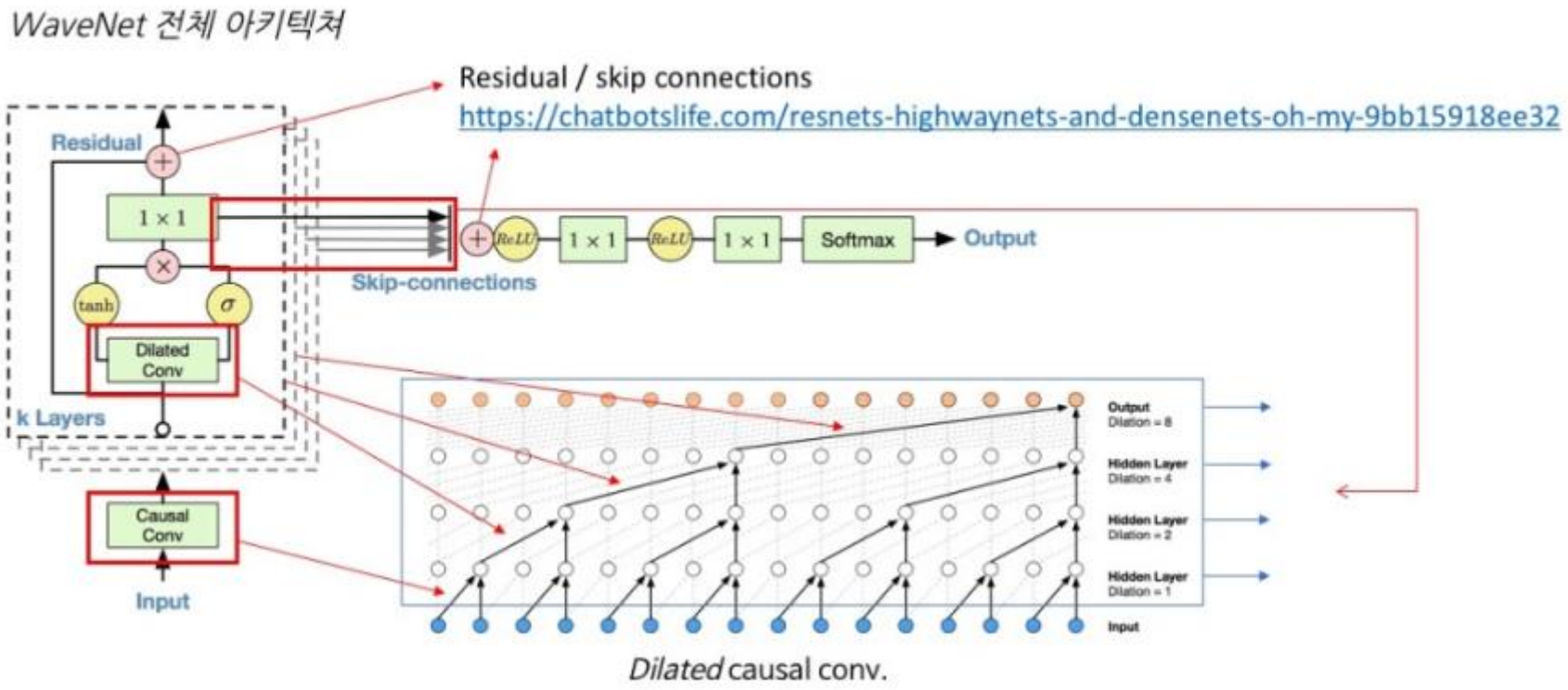
그다음 여러 번 1x1 레이어를 지나면서 wavent decoder 가 conditional signal을 받는다.

디코더는

4 blocks of 10 residual-layers

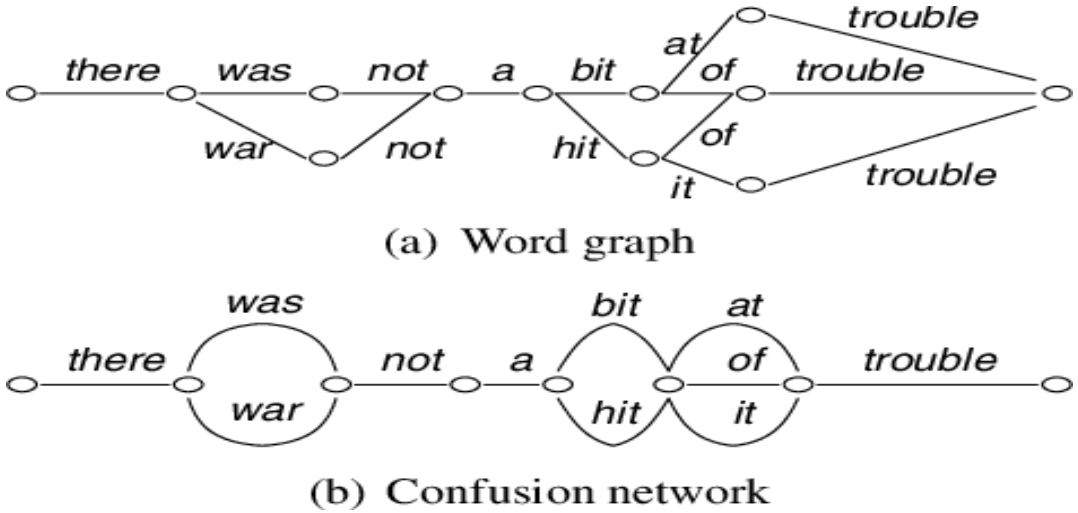
Decoder의 구조가 이거랑 같음!
Wave net 구조 보면 될 듯

WaveNet : A *Generative* Model for Raw Audio



Confusion network

generate model에서 성능측정으로 많이 쓴다는데 뭔소린지 모르겠다. 띵띵..



A Confusion Network (CN) G is a weighted directed graph with a start node, an end node, and word labels over its edges. The CN has the peculiarity that each path from the start node to the end node goes through all the other nodes. As shown in Figure 1, a CN can be represented as a matrix of words whose columns have different depths. Each word wj,k in G is identified by its column j and its position k in the column; word wj,k is associated to the weight pj,k corresponding to the posterior probability Pr(f = wj,k | o, j) of having f = wj,k at position j given o. A realization f = f1, . . . , fm of G is associated with the probability Pr(f | o), which is factorized as follows: Pr(f | o) = Ym j=1 Pr(fj | o, j) (3) The generation of a CN from an ASR word-graph [9] can also produce special empty-words in some columns. These empty-words permit to generate source sentences of different length and are treated differently from regular words only at the level of feature functions.

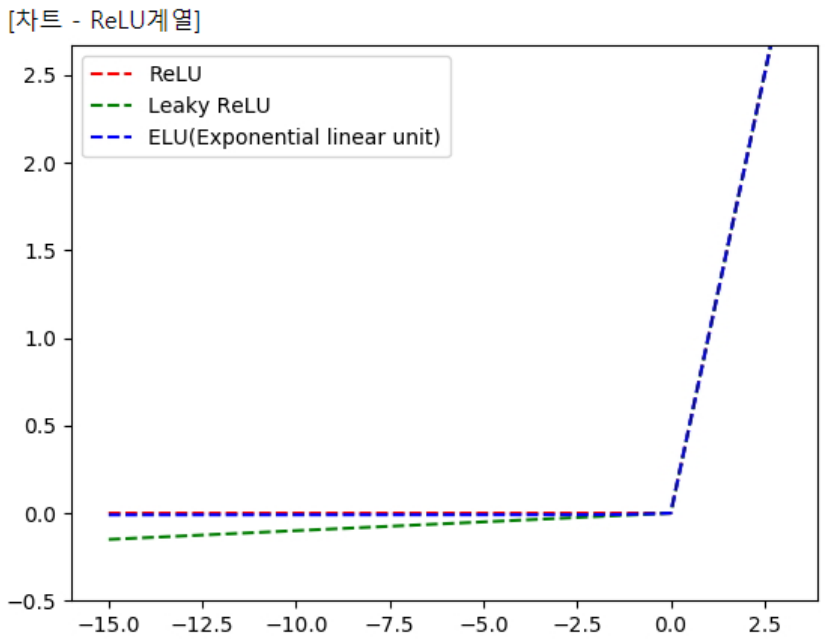
Confusion network은 참고논문과 같은데

3개의 1d convolution layer를 사용하였고

elu activate function을 사용하였다.

```
def elu_func(x): # ELU(Exponential linear unit)
    return (x>=0)*x + (x<0)*0.01*(np.exp(x)-1)
```

참고논문 봐도 confusion networ가 어디있는지...



출처

- http://www.modulabs.co.kr/?module=file&act=procFileDownload&file_srl=19972&sid=3cb67a2e845035fe1beb87ac3c4ca4d4&module_srl=17958
- https://kakalabblog.wordpress.com/2017/07/18/wavenetnsynth-deep-audio-generative-models/