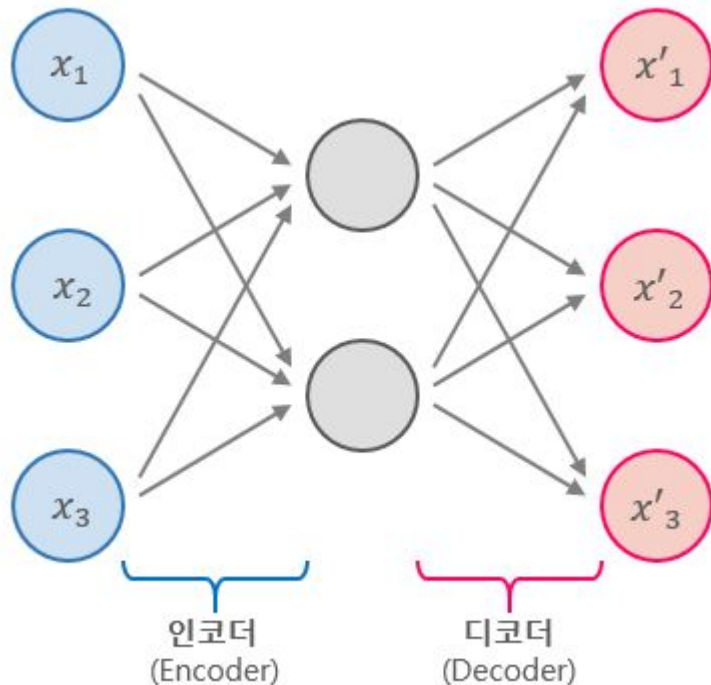


# Auto Encoder & VAE

김민규

# Auto Encoder란?



- 인코더(encoder) : 인지 네트워크라고도 하며, 입력을 내부 표현으로 변환.
- 디코더(decoder) : 생성 네트워크라고도 하며, 내부 표현을 출력으로 변환한다.
- 입력과 출력이 같은 구조
- **Bottleneck Hiddenlayer**
  - Latent Variable
  - Feature
  - Hidden representation

# AutoEncoder 수식과 학습 방법

- 수식

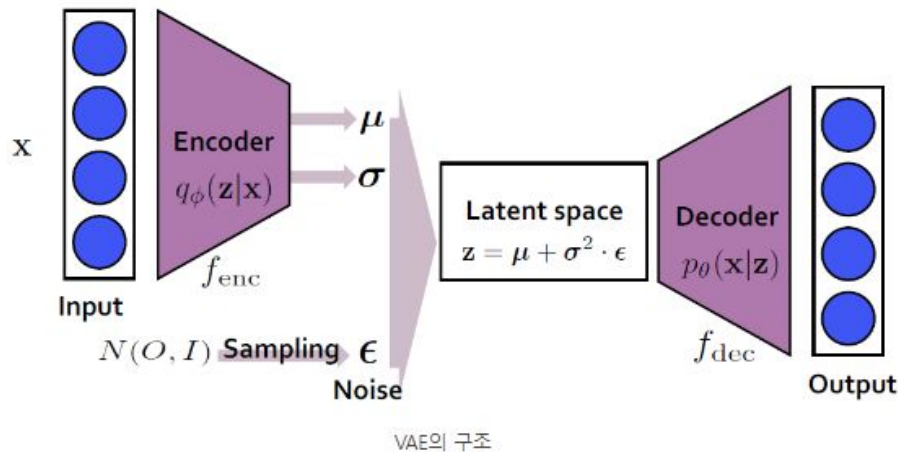
- Input Data를 Encoder Network에 통과시켜 압축된 Z값을 얻습니다.
- 압축된 Z vector로부터 Input Data와 같은 크기의 출력 값을 생성합니다.
- 이때 Loss값은 입력값  $x$ 와 Decoder를 통과한  $y$ 값의 차이입니다.

- 학습 방법

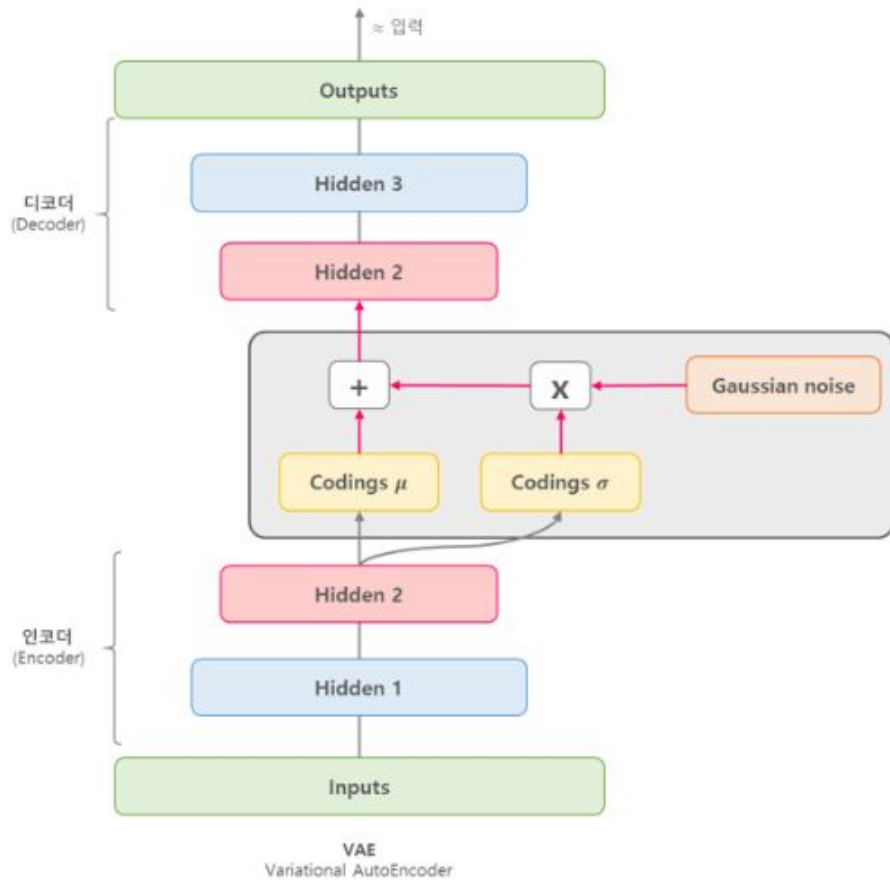
- Decoder Network를 통과한 Output layer의 출력 값은 Input 값의 크기와 같아야 합니다.
- 학습을 위한 출력 값과 입력 값이 같아져야합니다.

# VAE (Variational AutoEncoder)

- Encoder : input을 latent space로 변환하는 역할
- Decoder : latent space를 input으로 변환하는 역할
- Latent space : 어떤 숨겨진 vector, latent space가 주어져야, decoder는 이를 활용해 data를 generate 할 수 있음.

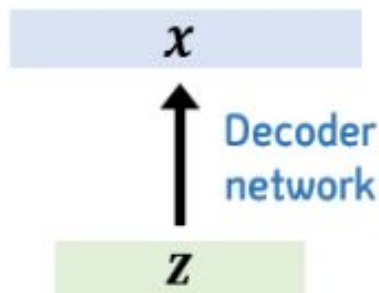


# VAE의 구조



# VAE의 수식

Decoder



$z$  latent variable의 확률분포

$$p_{\theta}(z)$$

$z$ 가 given일 때  $x$ 의 확률분포

$$p_{\theta}(x|z^{(i)})$$

어떻게 학습?

네트워크의 출력값이 있을 때  
우리가 원하는 정답  $x$ 가 나올 확률이 높길바람  
=  $x$ 의 likelihood를 최대화하는 확률분포 찾자



Maximize

$$p_{\theta}(x) = \int p_{\theta}(z)p_{\theta}(x|z)dz$$

# VAE 장단점

- 장점 : 확률 모델을 기반으로 했기 때문에, 잠재 코드를 더 유연하게 계산할 수 있음
- 단점 : **Density**를 직접적으로 구한 것이 아니기 때문에 **Pixel RNN/CNN**과 같이 직접적으로 **Density**를 구한 모델보다는 성능이 떨어집니다.

# AutoEncoder vs VAE

- AutoEncoder의 목적은 Manifold Learning입니다.
  - AE는 네트워크의 앞단을 학습하기 위해 뒷단을 붙인 것 입니다.
  - 입력 데이터의 압축을 통해 데이터의 의미있는 manifold를 학습합니다.
- Variation AutoEncoder는 Generative Model입니다.
  - Decoder(생성)을 학습시키기 위해 앞단을 붙인 것 입니다.
  - 공교롭게도, 구조를 보니 AE가 같은 것
  - VAE는 Generative Model 입니다.