




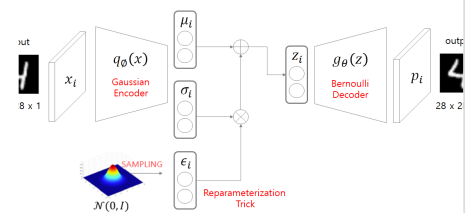
VAE

<https://lilianweng.github.io/lil-log/2018/10/13/flow-based-deep-generative-models.html>

[논문] VAE(Auto-Encoding Variational Bayes) 직관적 이해


최근 딥러닝 스터디 중 한 군데에서 케라스 창시자에게 배우는 딥러닝 교재로 공부하고 있다. 마지막 8장 부분은 일반적인 딥러닝의 여러 부분을 커버하고 있다. 8.1 Text Generation with LSTM 8.2 Implementing

 <https://taeu.github.io/paper/deeplearning-paper-vae/>



딥러닝 Ch3.3 VAE

숙명여자대학교 기계시스템학부 딥러닝 VAE 강의입니다.

 <https://www.youtube.com/watch?v=GbCAwVVKaHY>

딥러닝과 설계

3.3. Variational AutoEncoder (VAE)

강남우

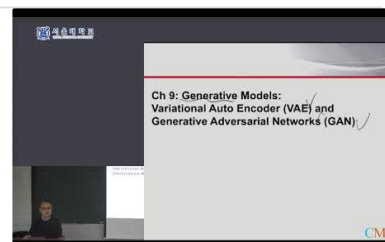
기계시스템학부
숙명여자대학교



[딥러닝의 기초] 201201 Chapter9(1): Generative Models (VAE, GAN)

Introduction to Deep Learning - Chapter9(1): Generative Models (VAE, GAN) Lecturer : Jungwoo Lee (Professor, ECE SNU) 강의자 : 이정우 교수 (서울대학교 전기정보공학부) Communication...

 <https://www.youtube.com/watch?v=bSHPL5xOze8>



도입

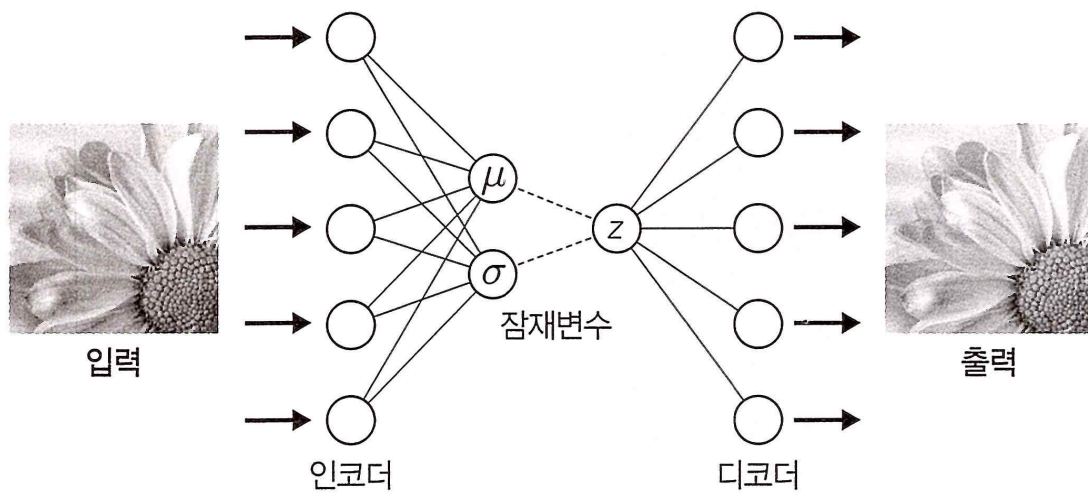
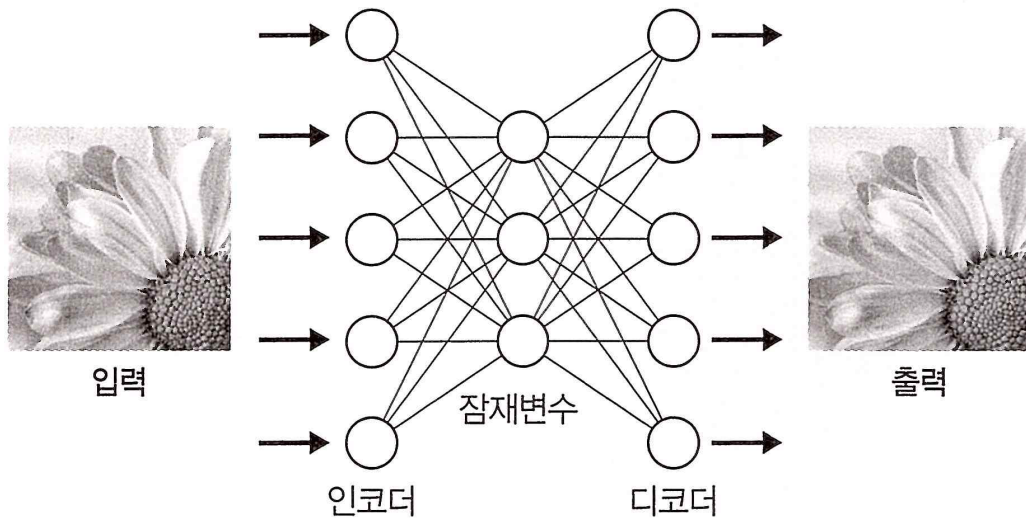
- Auto Encoder 의 목적을 생각하고 Manifold 에서 새로운 Data를 생성해보자

AE 의 목적

- Manifold Learning 을 통해 Feature의 차원을 줄이는 게 목적이다.
- Manifold 위에 찍힌 점을 찍는 다면 어떻게 될까?
 - 데이터를 새로 생성할수 있지 않을까?

- 실제 결과는 제대로 된 데이터가 생성되진 않는다.
- Latent Space 는 입력 데이터들에 대한 특징 정보들을 함축하고 있는데, 그렇다면 데이터를 생성할순 없을까?

<그림??>



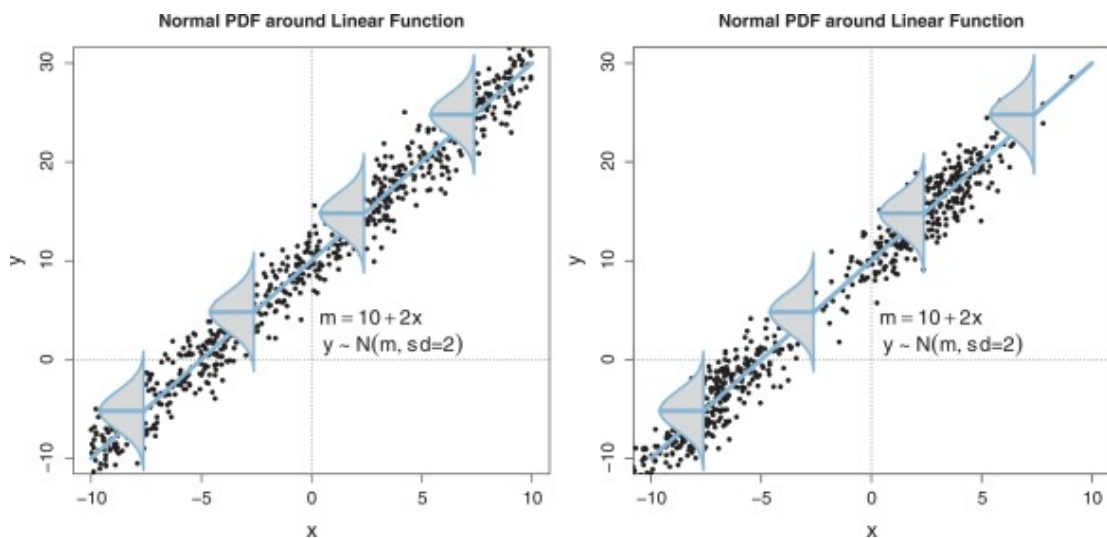
Generative Model

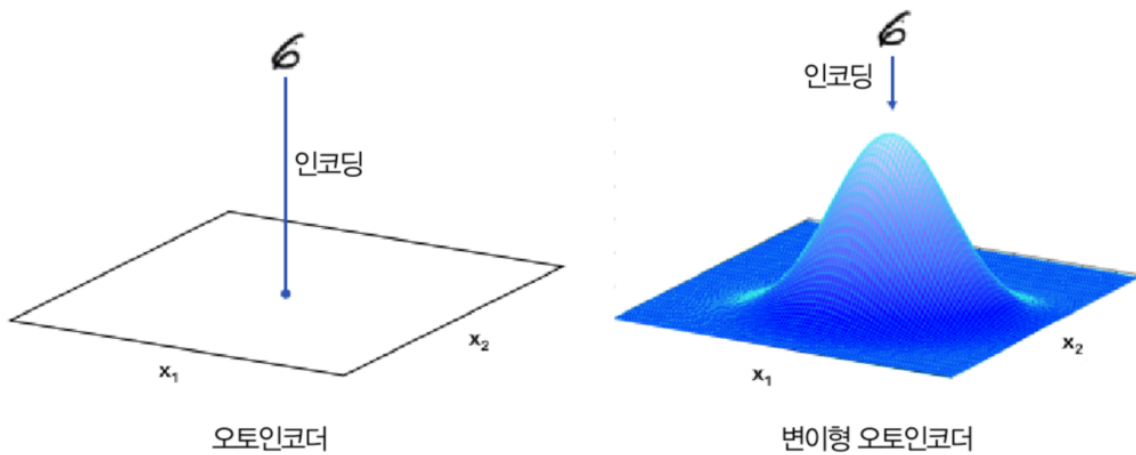
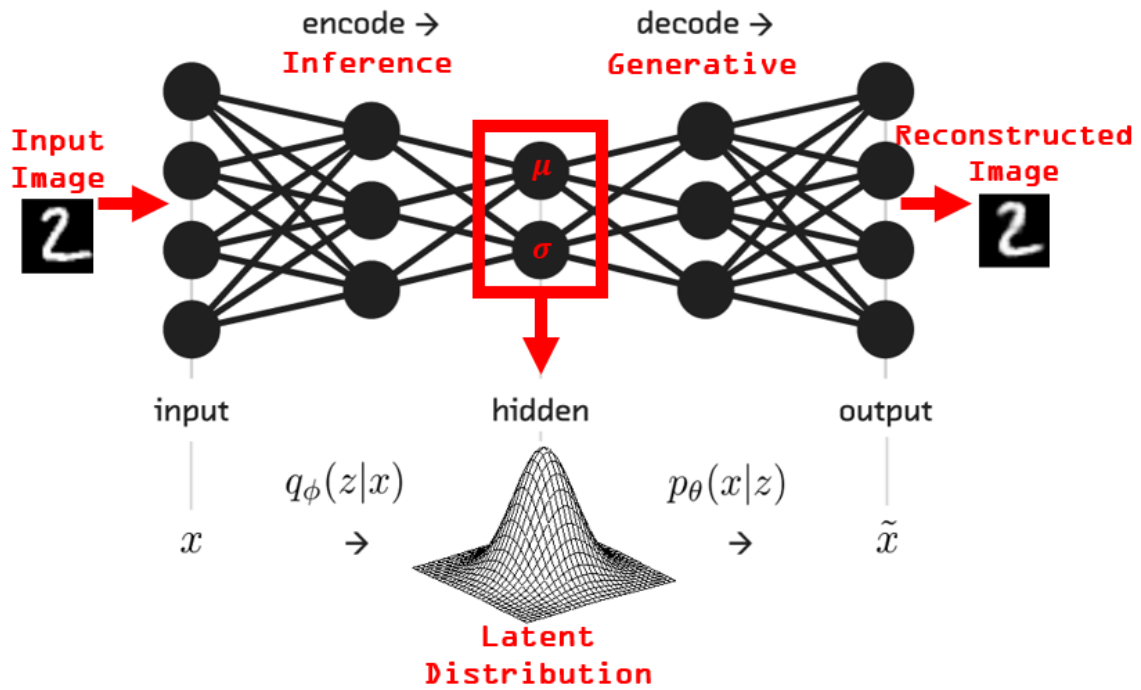
- Goal : Data Set에 존재할만한 Data 를 생성해 낸다.
- 종류
 - Variation AutoEncoder

- Generative Adversarial Network

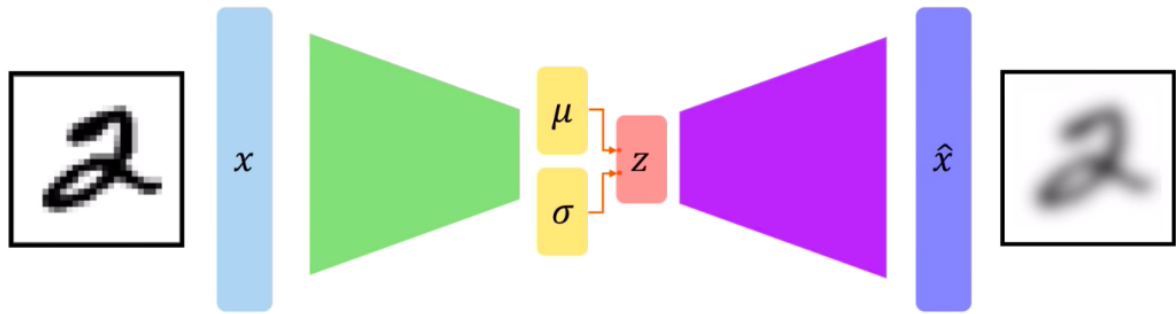
Density Estimation

- 데이터를 제대로 생성해내지 못한 이유를 생각해 보자.
 - 데이터 자체의 Density(밀도)가 많이 떨어진다.
 - 해결법 : 데이터의 밀도를 추정하면 됨!
- UnsuperVised Learning : DataSet이 존재하면 DataSet의 밀도를 추정하여 확률 밀도 함수를 만든다. 물론, **함수 자체를 알수 없다.**
- 확률 밀도 함수(pdf) 생성 → Data Set의 분포를 알수 있다





Variational Autoencoder

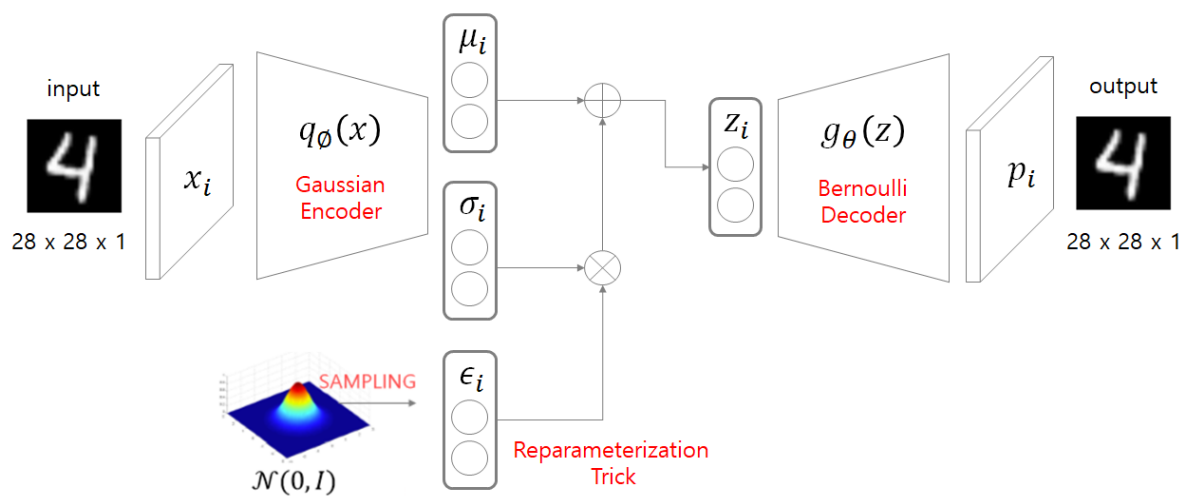


- 학습 데이터의 분포를 따르는 새로운 데이터를 만드는 AutoEncoder 기반의 생성 모델
- AE 는 Encoder 를 이용해 차원을 축소하기 위한 목적
- VAE 는 Decoder를 이용해 새로운 Data 를 생성하기 위한 Model

Variational(변분법) Method

- 어떤 함수 $p(x)$ 의 극점을 찾는 문제에서 해당 함수를 직접 다루는 것이 쉽지 않을 때, 쉽게 다룰 수 있는 다른 함수 $q(x)$ 로 대체해 이를 최적화 하여, $p(x)$ 에 대한 근사적인 해를 구하는 방법

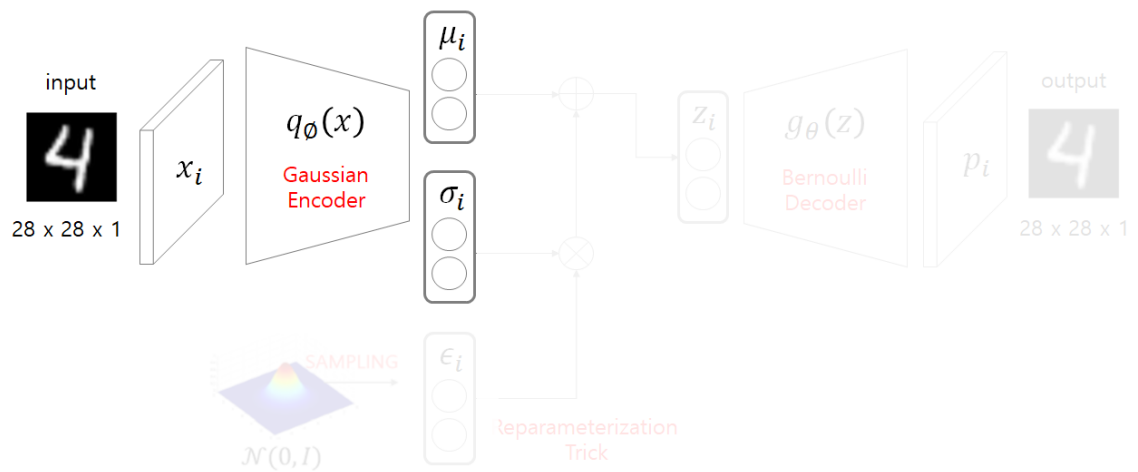
How?



1. $input(x_i) \rightarrow q_\phi(x) \rightarrow \mu_i, \sigma_i$
2. $\mu_i, \sigma_i, \epsilon_i \rightarrow z_i$
3. $z_i \rightarrow g_\theta(z_i) \rightarrow output(p_i)$

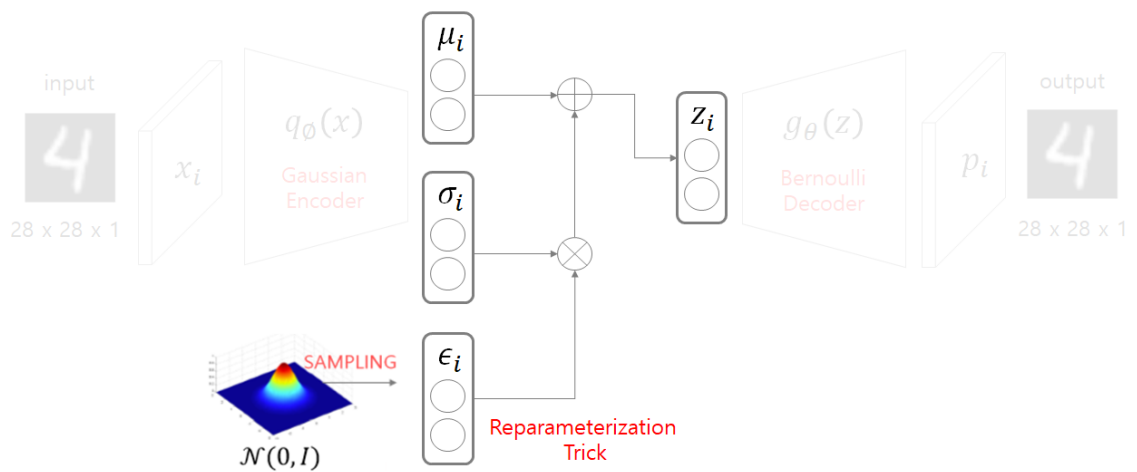
단계1. Encoder

- $input(x_i) \rightarrow q_\phi(x) \rightarrow \mu_i, \sigma_i$



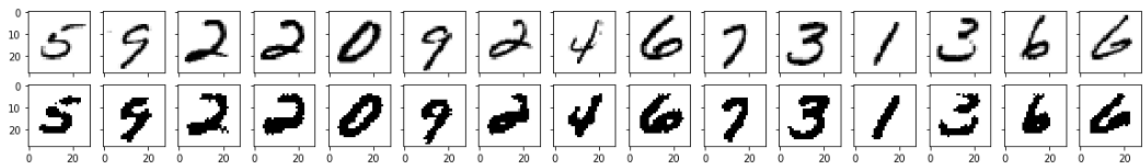
단계 2. Reparameterization Trick (Sampling)

- $\mu_i, \sigma_i, \epsilon_i \rightarrow z_i$

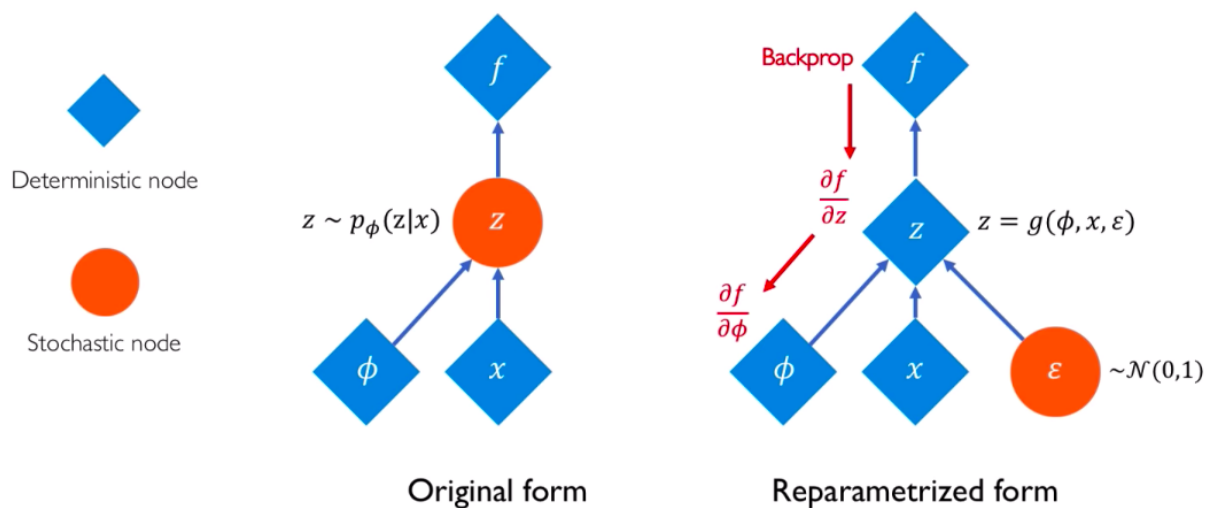


- 새로운 Data를 생성하기 위해 Sampling 을통해 Decoding 해준다.
- Sampling 을 위해 Data의 평균과 분산을 학습한다.

- Sampling 없이 추론한 경우



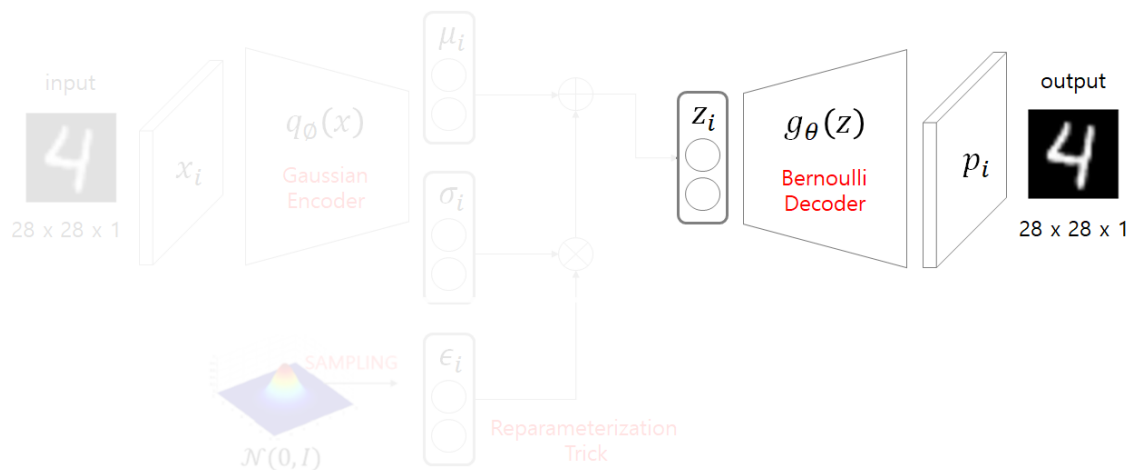
- Reparameterized form



- 우리가 추정하고 싶은 확률 분포 : $z^{i,j} \sim N(\mu_i, \sigma_i^2 I)$
 - Sampling 결과를 backpropagation 할수는 없다.
- reparameterization trick을 통해 추정할 식 : $z^{i,j} = \mu_i + \sigma_i^2 \odot \epsilon$
 - 단, ($\epsilon \sim N(0, I)$)
 - $z^{i,j} \sim N(\mu_i, \sigma_i^2 I)$ 와 같은 분포를 갖고, Backpropagation 이 가능한 식
 - 정규분포에서 z_i 를 샘플링 하는것과 ϵ 을 정규분포에서 샘플링하고 그 값을 분산과 곱하고 평균을 더하는것은 같은 분포를 갖는다.

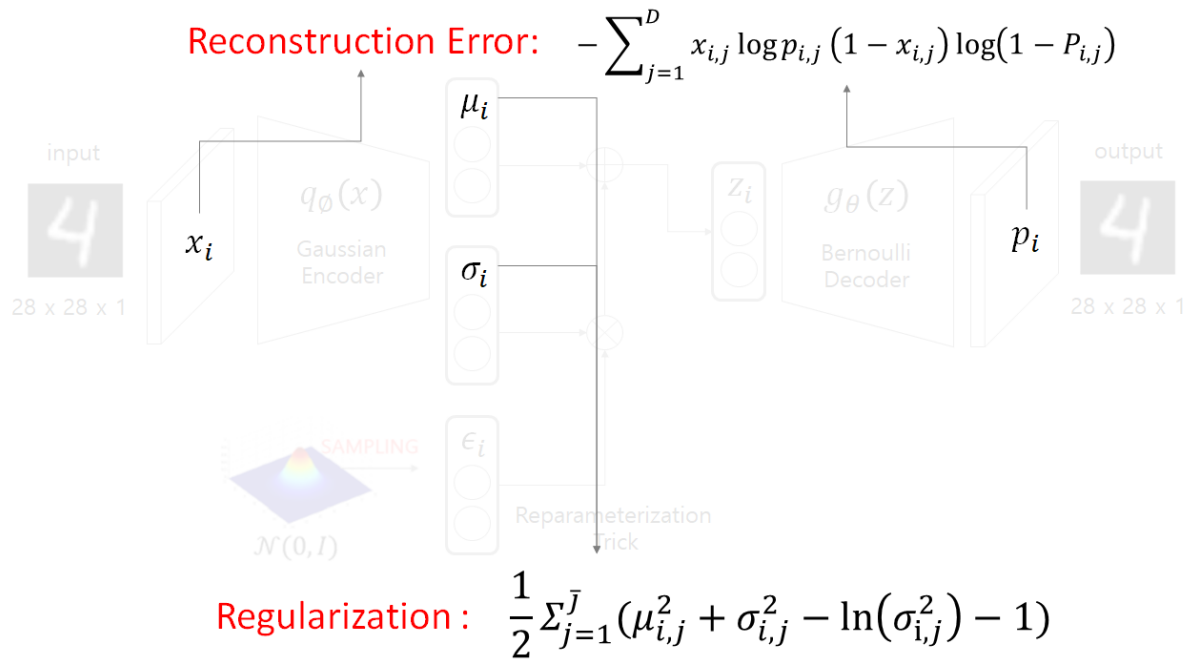
단계3. Decoder

- $z_i \rightarrow g_\theta(z_i) \rightarrow \text{output}(p_i)$



- p_i 의 값을 Bernoulli 시행의 결과로 가정했기 때문에 activation function 은 sigmoid 로 설정 준다.

Loss Function



- Loss function 은 Reconstruction Error 와 Regularization 두가지를 고려해야 한다.
 - Reconstruction Error : input data와 output data 에 대한 손실함수 이다.
 - Decoder의 결과가 Bernoulli 분포를 따른다고 가정하였으므로 확률값으로 손실함수를 계산할수 있는 cross entropy 를 사용
 - Regularization : 원래 Data가 갖고있는 True 분포에 근사하기 위한 loss function
 - True pdf 와 Approximated pdf 두 확률 분포간의 차이를 계산하기 위해 위해 **Kullback-Leibler divergence** 방식을 사용한다.

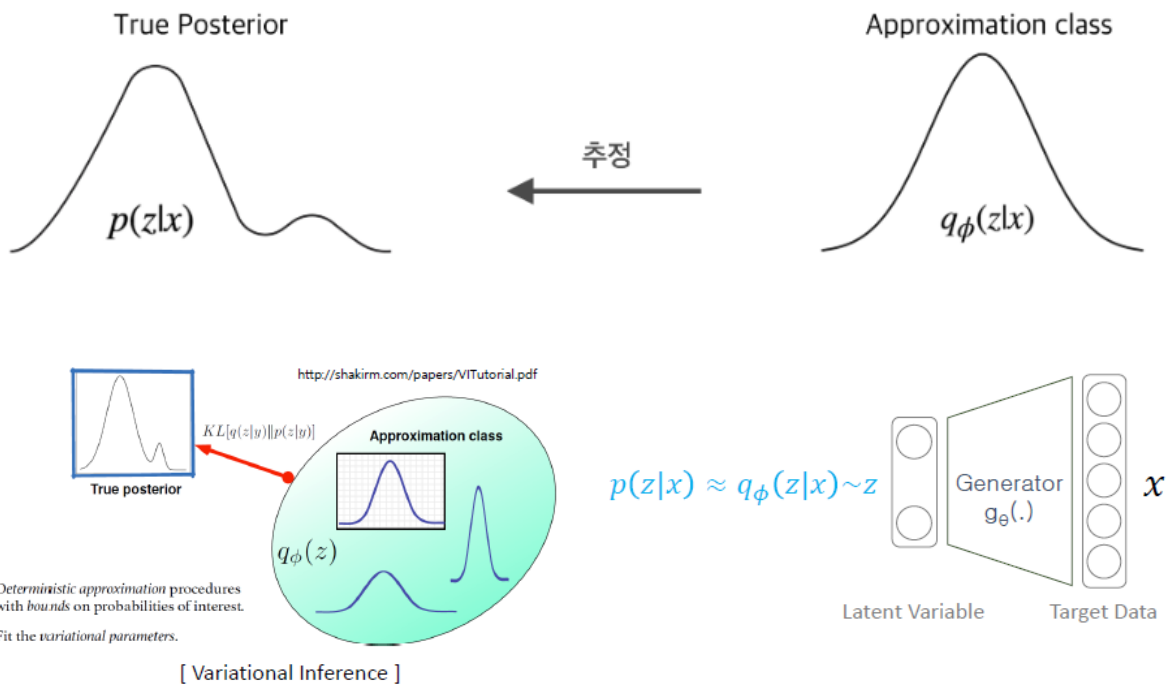
- pdf(probability density function) 에 근사하도록 Data를 Sampling 한다.
- $p_\theta(x) = \int p_\theta(z)p_\theta(x|z)dz$
- $p_\theta(z)$: z 의 분포는 Data의 Sampling 을 통해 추정한다.
- $p_\theta(x)$ 의 값을 직접적으로 구하기 어렵기 때문에 $p_\theta(x)$ 의 log likelihood를 다른 확률 분포인 가우시안 분포를 이용해 추정한다.

- VAE에서는 $q_\theta(x|x)$ 의 분포를 가우시안 분포로 가정하고 평균 벡터 μ 와 표준편차 벡터 Σ 를 학습한다.

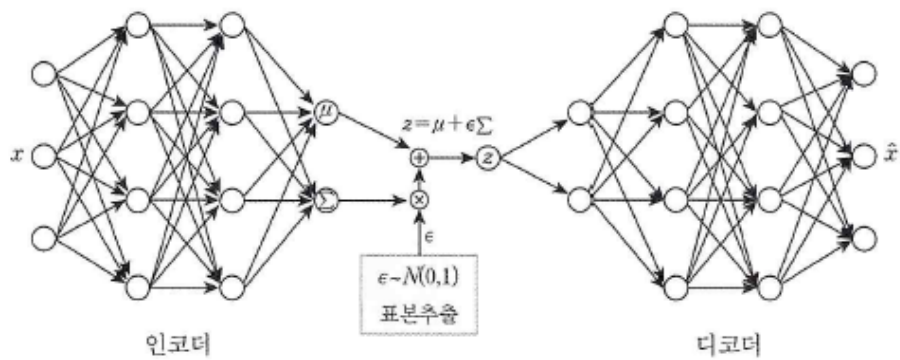
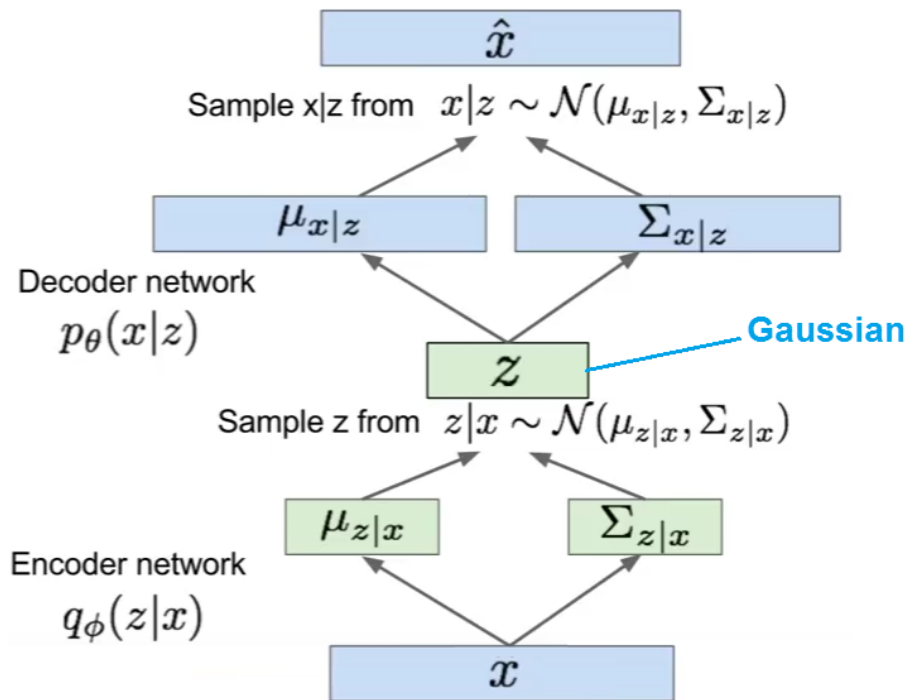
$$\log p_\theta(x) = \sum_z q_\phi(z|x) \log p_\theta(x) = \sum_z q_\phi(z|x) \log \frac{p_\theta(x, z)}{p_\theta(z|x)}$$

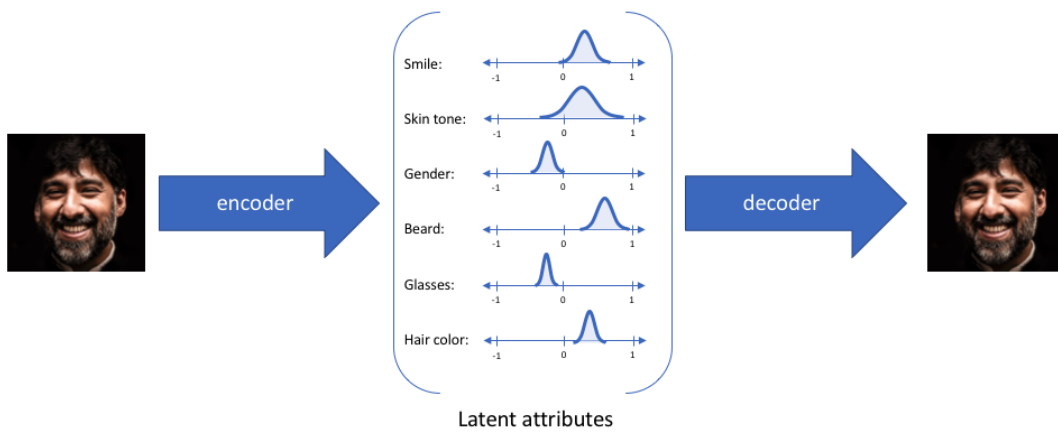
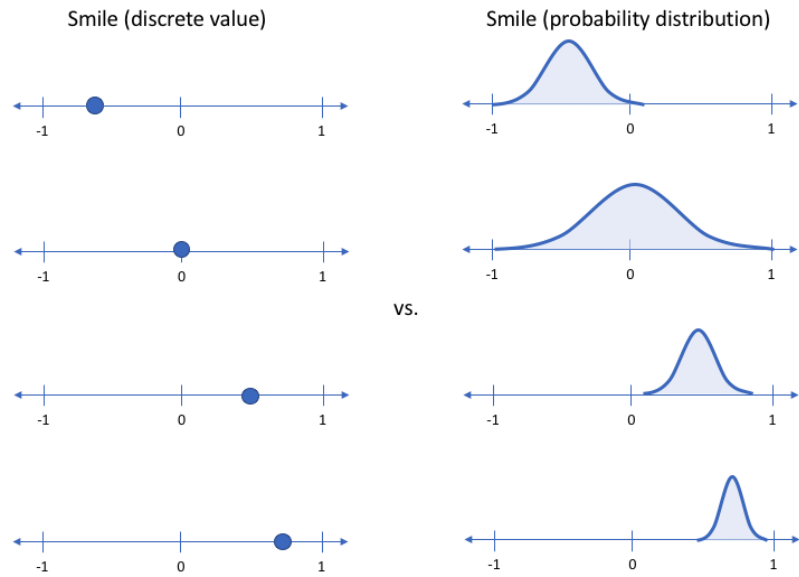
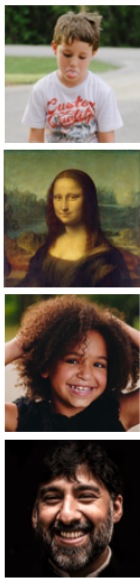
$$\begin{aligned} \log p_\theta(x) &= \sum_z q_\phi(z|x) \log \frac{p_\theta(x, z)}{p_\theta(z|x)} \frac{q_\phi(z|x)}{q_\phi(z|x)} \\ &= \sum_z q_\phi(z|x) \log \frac{q_\phi(z|x)}{p_\theta(z|x)} \frac{p_\theta(x, z)}{q_\phi(z|x)} \\ &= \sum_z q_\phi(z|x) \log \frac{q_\phi(z|x)}{p_\theta(z|x)} + \sum_z q_\phi(z|x) \log \frac{p_\theta(x, z)}{q_\phi(z|x)} \end{aligned}$$

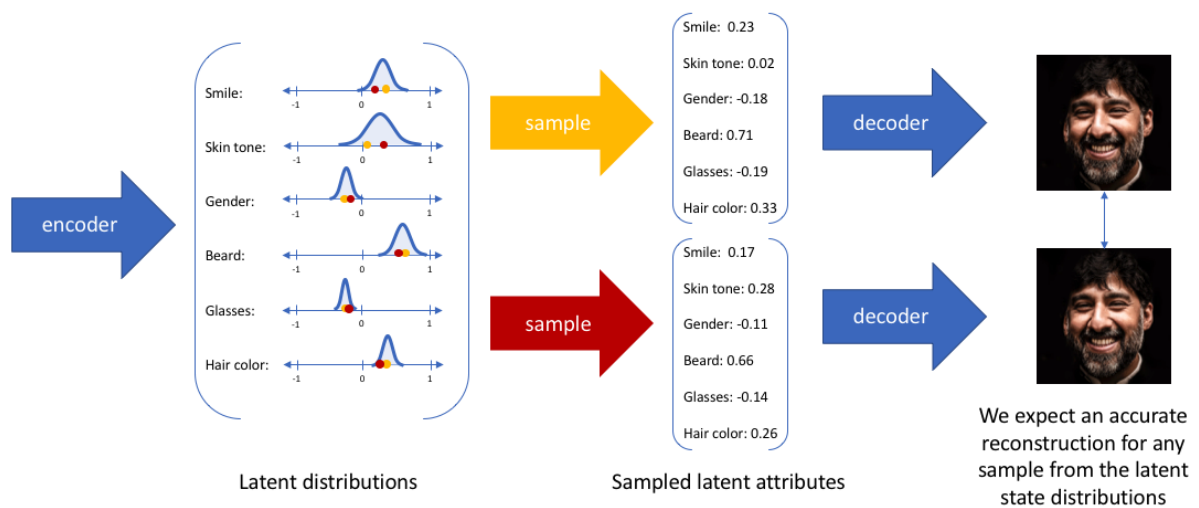
Variational Inference



- VAE의 Encoding 부분은 데이터를 이용해 $q_\theta(x|x)$ 인 확률분포를 계산하는 신경망으로 구성되고, 디코더는 $p_\theta(x|x)$ 인 확률 분포에 따라 출력 \hat{x} 을 생성하는 신경망으로 구성







<https://www.jeremyjordan.me/variational-autoencoders/>

From Autoencoder to Beta-VAE

Updated on 2019-07-18: add a section on VQ-VAE & VQ-VAE-2.]

[Updated on 2019-07-26: add a section on TD-VAE.] Autocoder is invented to reconstruct high-dimensional data using a neural network

<https://lilianweng.github.io/posts/2018-08-12-vae/>

