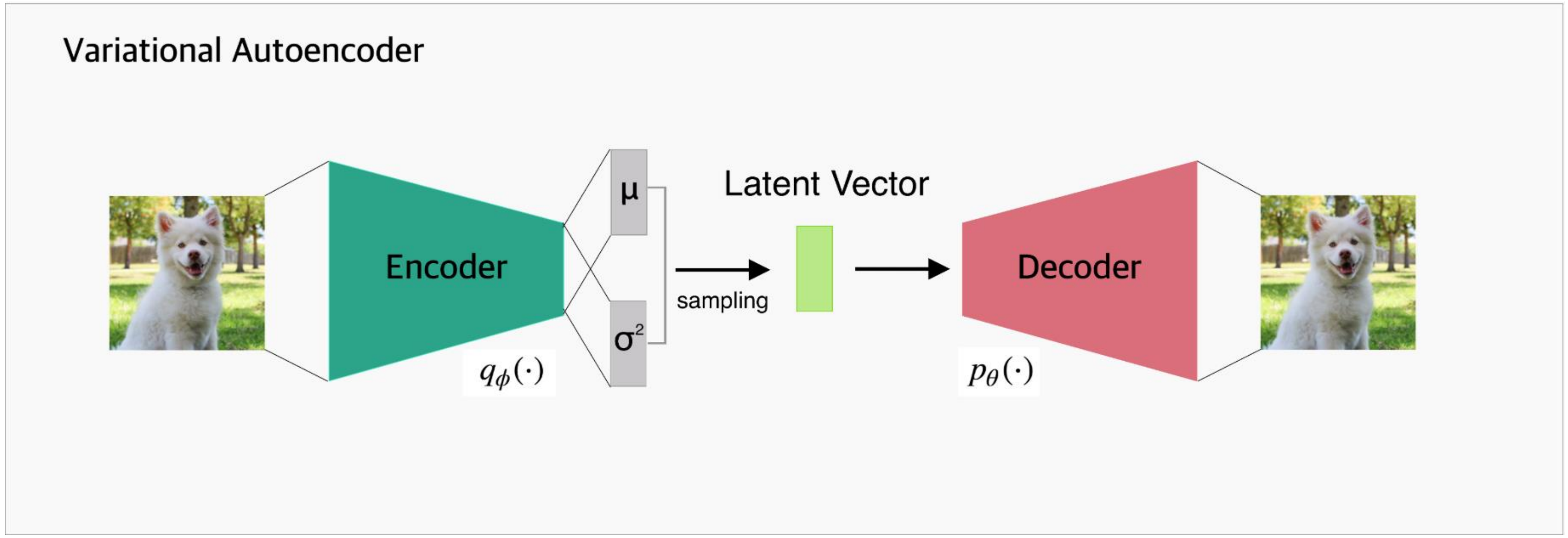
딥러닝 세미나 Season #6

VAE

한양대학교 인공지능연구실
조충현

VAE?



- VAE는 생성 모델(Generative Model)중 하나로, 확률분포 P(x)를 학습함으로써, 데이터를 생성하는 것이 목표
- Encoder에서는 학습용 데이터(이하 x)를 입력으로 받고 잠재변수(이하 z)의 확률분포에 대한 파라미터 출력
- Decoder는 잠재변수에 대한 확률 분포 $p(z)$에서 샘플링한 벡터를 입력 받아 원본 이미지 복원

VAE 목표

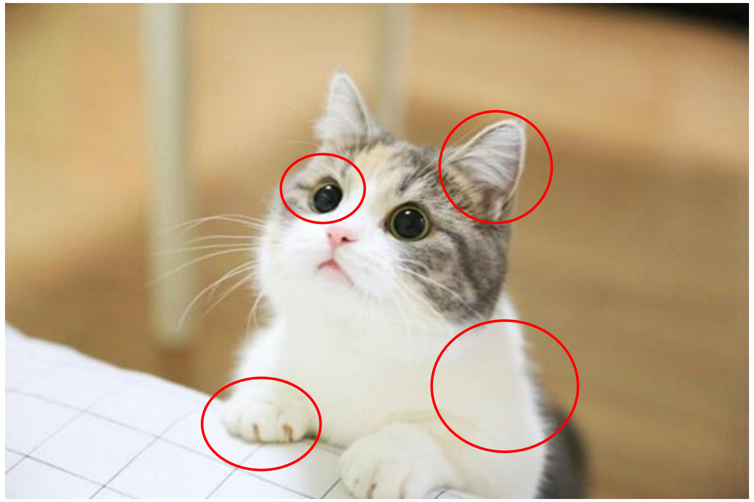
1. 주어진 데이터를 잘 설명하는 잠재변수의 분포를 찾는 것(Encoder의 역할)

2. 잠재변수로 부터 원본 이미지와 같은 이미지를 잘 복원하는 것(Decoder의 역할)

잠재변수?

Encoder

Decoder

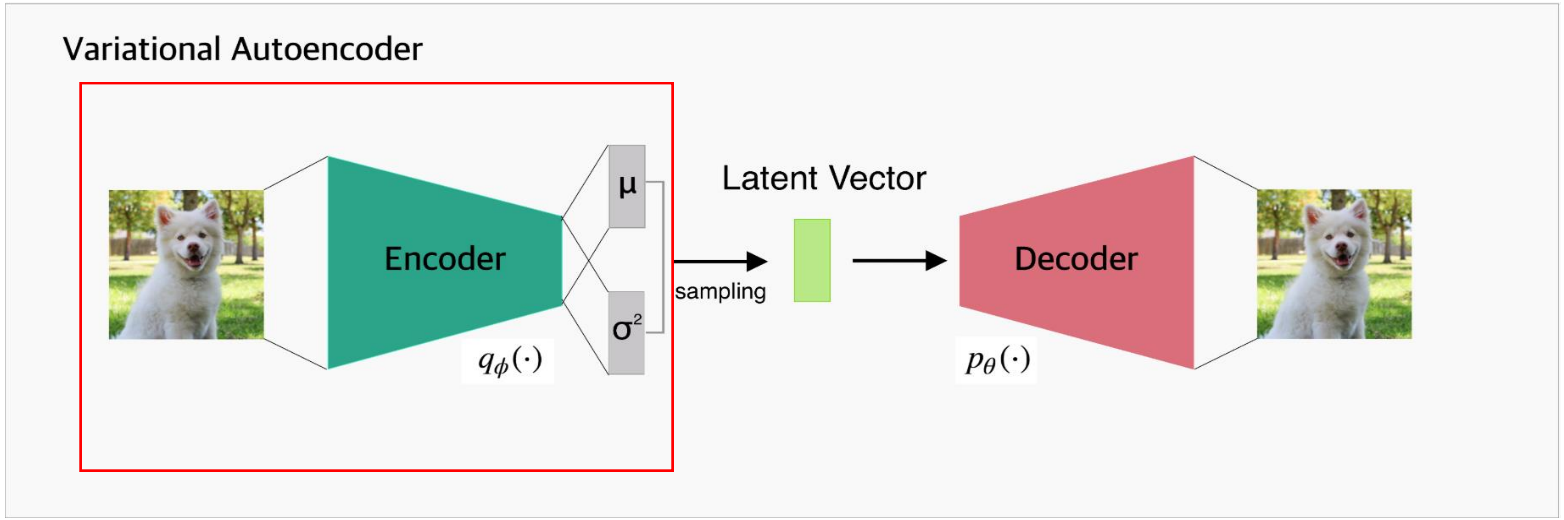


- 잠재변수는 encoder에서 입력 데이터를 추상화하여 잠재적인 특징을 추출해서 나온 것
- 또한 decoder에서는 이러한 잠재적 특징을 바탕으로 원 데이터로 복원하는 역할

Encoder

Encoder의 역할

- 데이터가 주어졌을 때 Decoder가 원래의 데이터로 잘 복원할 수 있는 z를 샘플링 할 수 있는 이상적인 확률분포 $p(z|x)$를 찾는 것



But!!

- 어떤 것이 이상적인 확률분포 인지는 아무도 모른다!!!

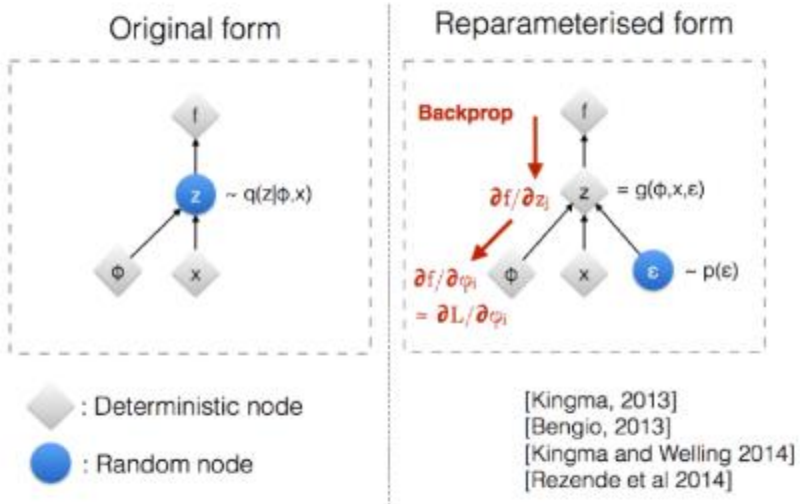
Variational inference

- 변분추론이란, 계산이 어려운 확률분포를 추정하기 위해서 다루기 쉬운 분포(approximation class, $q_\emptyset$)를 가정
이 확률분포의 모수를 바꿔가며, 이상적인 확률분포에 근사하게 만들어 그 확률분포를 대신 사용

- Encoder는 ∅라는 파라미터들을 바꾸어가며, $q_\emptyset(z|x)$ 확률 분포를 이상적인 확률분포 $p(z|x)$에 근사시키는 역할을 수행
- 보통 $q_\emptyset$은 Gaussian 정규 분포라 가정한다. 그리고 이 때 z의 marginal distribution은 평균이 0이고 분산이 1인 표준 정규분포로 가정

Reparameterization trick

$$z = \mu(x) + \sigma(x) \times \epsilon, \quad \epsilon \sim N(0,1)$$



$$z^{i,l} \sim N(\mu_i, \sigma_i^2 I)$$

$$z^{i,l} = \mu_i + \sigma_i^2 \odot \epsilon$$
$$\epsilon \sim N(0, I)$$

잠재변수 z를 샘플링 하기 위하 사용하는 트릭

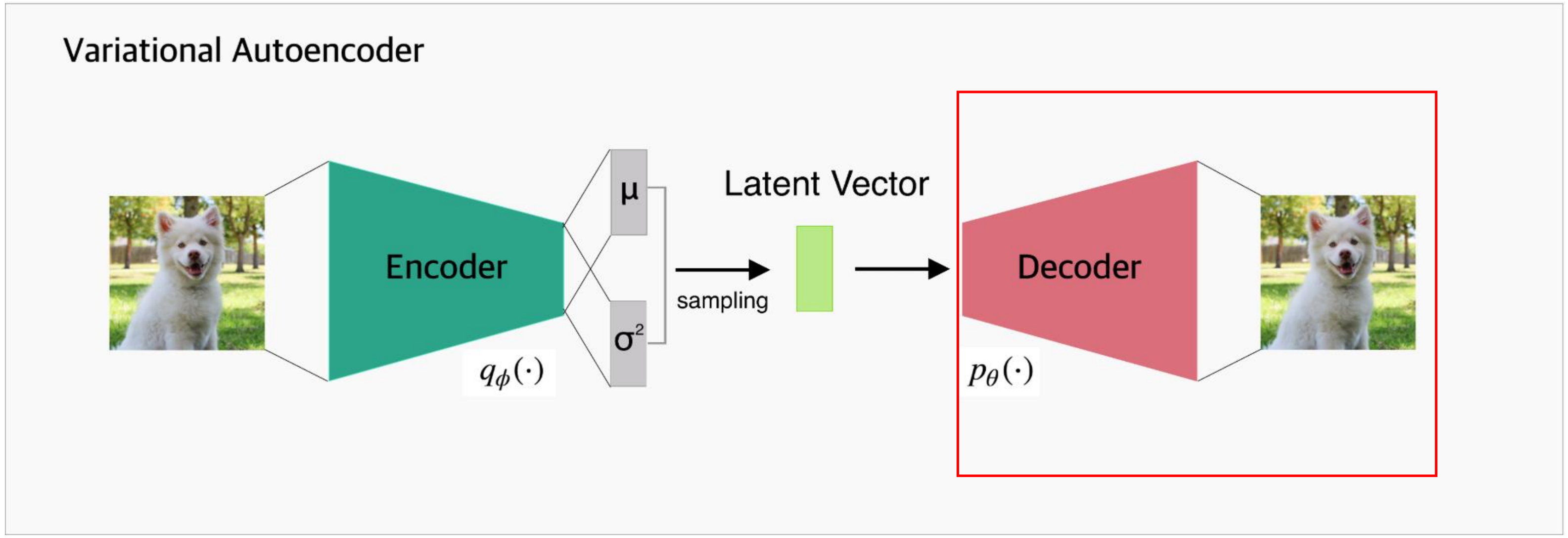
왜?

- 이 트릭을 사용하면 역전파를 통해 Encoder가 산출하면 평균과 분산을 업데이트 할 수 있다.

Decoder

Decoder의 역할

- 추출한 샘플을 입력으로 받아, 다시 원본으로 재구축하는 역할을 수행



Decoder

$p(x)$ 실제 데이터의 분포

$$\log(p(x)) = \log\left(\int p(x,z)dz\right) = \log\left(\int p(x|z)p(z)dz\right) \xrightarrow{q_\emptyset(z|x)} \log(p(x)) = \log\left(\int p(x|z)\frac{p(z)}{q_\phi(z|x)}q_\phi(z|x)dz\right)$$

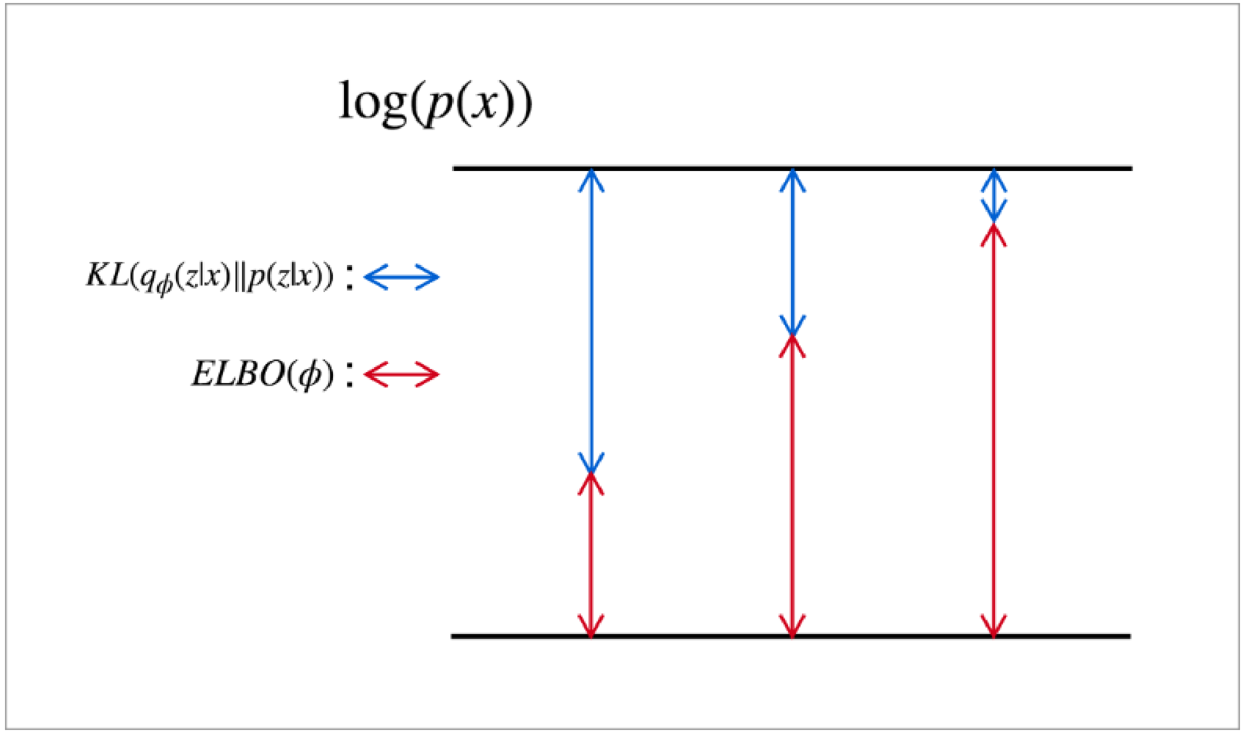
Jensen's Inequality

$$\log(p(x)) \geq \int \log\left(p(x|z)\frac{p(z)}{q_\phi(z|x)}\right)q_\phi(z|x)dz \xrightarrow{\text{최종}} \log(p(x)) \geq \boxed{\int \log(p(x|z))q_\phi(z|x)dz - \int \log\left(\frac{q_\phi(z|x)}{p(z)}\right)q_\phi(z|x)dz}$$

최종 식의 우변식이 Evidence LowerBOund 줄어서 ELBO라고 부른다.

이 ELBO(ø) 값을 최대화하는 ø를 찾으면 최종식의 우변과 좌변은 같게 된다.

ELBO 와 KL의 관계



$$\log(p(x)) = \int \log(p(x))\, q_\emptyset(z|x)dz$$

$$\log(p(x))$$
$$= \int \log\left(\frac{p(x,z)}{q_\emptyset(z|x)}\right) q_\emptyset(z|x)dz + \int \log\left(\frac{q_\emptyset(z|x)}{p(z|x)}\right) q_\emptyset(z|x)dz$$

ELBO(∅) $KL(q_\emptyset(z|x)||p(z|x))$

따라서 위의 그림처럼 ELBO(∅)를 최대화하는 것이 곧, $KL(q_\emptyset(z|x)||p(z|x))$를 최소화 하는 것

ELBO

$$\int \log(p(x|z))q_\phi(z|x)dz = \mathbb{E}_{q_\phi(z|x)}[\log(p(x|z))]$$
$$\int \log\left(\frac{q_\phi(z|x)}{p(z)}\right)q_\phi(z|x)dz = KL(q_\phi(z|x)\|p(z))$$

$\rightarrow$

$$ELBO(\phi) = \mathbb{E}_{q_\phi(z|x)}[\log(p(x|z))] - KL(q_\phi(z|x)\|p(z))$$

$\log(p(x|z)) \rightarrow \log(p_\theta(x|z))$ 표현하면 θ를 조정하여 최대하는 것

따라서, 최종 VAE의 Loss 함수는 아래와 같다

$$\mathcal{L}_{(\theta,\phi;x^i)} = -\mathbb{E}_{q_\phi(z|x^i)}[\log(p_\theta(x^i|z))] + KL(q_\phi(z|x^i)\|p(z))$$

Loss

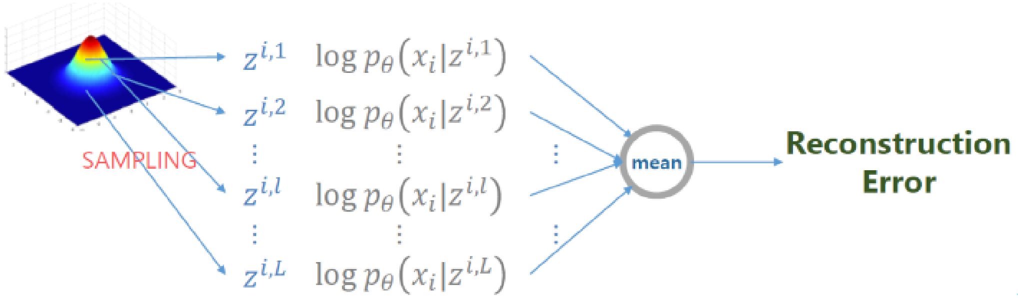
1. Expectation

$$\mathbb{E}_{q_\phi(z|x^i)}[\log(p_\theta(x^i|z))] \approx \frac{1}{L}\sum_{z^{i,l}} \log(p_\theta(x^i|z^{i,l}))$$

딥러닝의 특성상 Mini-batch로 학습하기 때문에 $q_\emptyset$로 부터 L개의 데이터를 샘플링하여, Monte-carlo 방식으로 위와 같이 구한다.

이 때, $p(x)$를 Bernoulli 분포로 가정하면, $\log(p_\theta(x^i|z^{i,l}))$ 값은 Cross Entropy 식이 된다.

$$\begin{aligned}
\log(p_\theta(x^i|z^i)) &= \log\prod_{j=1}^{D} p_\theta(x_{i,j}|z^i) \\
&= \sum_{j=1}^{D} \log p_\theta(x_{i,j}|z^i) \\
&= \sum_{j=1}^{D} \log p_{i,j}^{x_{i,j}}(1-p_{i,j})^{x_{i,j}} \\
&= \sum_{j=1}^{D} x_{i,j}\log p_{i,j} + (1-x_{i,j})\log(1-p_{i,j})
\end{aligned}$$



- L is the number of samples for latent vector
- Usually L is set to 1 for convenience

Loss

2. KL

$$
\begin{aligned}
KL(q_\phi(z|x^i)\|p(z)) &= \frac{1}{2}\left\{ tr(\sigma_i^2) + \mu_i^T\mu_i - J + \ln\frac{1}{\prod_{j=1}^{J}\sigma_{j,j}^2} \right\} \\
&= \frac{1}{2}\left\{ \sum_{j=1}^{J}\sigma_{i,j}^2 + \sum_{j=1}^{J}\mu_{i,j}^2 - J - \sum_{j=1}^{J}\ln(\sigma_{i,j}^2) \right\} \\
&= \frac{1}{2}\sum_{j=1}^{j}(\mu_{i,j}^2 + \sigma_{i,j}^2 - \ln(\sigma_{i,j}^2) - 1)
\end{aligned}
$$

여기서 KL값 계산을 쉽게 하기 위해서 $q_\emptyset(z|x^i)$와 $p(z)$를 정규분포로 가정했다.

Kullback–Leibler divergence [edit]

The Kullback–Leibler divergence from $\mathcal{N}_0(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ to $\mathcal{N}_1(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$, for non-singular matrices Σ_0 and Σ_1, is:[8]

posterior prior

$$D_{\text{KL}}(\mathcal{N}_0\|\mathcal{N}_1) = \frac{1}{2}\left\{ \text{tr}\left(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_0\right) + (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^{\text{T}}\boldsymbol{\Sigma}_1^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) - k + \ln\frac{|\boldsymbol{\Sigma}_1|}{|\boldsymbol{\Sigma}_0|} \right\},$$

where k is the dimension of the vector space.

Loss

Variational inference를 위한
approximation class 중 선택

원 데이터에 대한 likelihood 선택

다루기 쉬운 확률 분포 중 선택

$$L_i(\phi, \theta, x_i) = -\mathbb{E}_{q_\phi(z|x_i)}\left[\log\left(p_\theta(x_i|z)\right)\right] + KL\left(q_\phi(z|x_i)||p(z)\right)$$

Reconstruction Error

- 현재 샘플된 z에 대한 negative log likelihood
- x_i에 대한 복원 오차 (AutoEncoder 관점)

Regularization

- 현재 샘플된 z에 대한 대한 추가 조건
- 샘플링되는 z들에 대한 통제성을 prior를 통해 부여, Variational distribution q(z|x)가 p(z)와 유사해야 한다는 조건을 부여

VAE의 장단점

장점

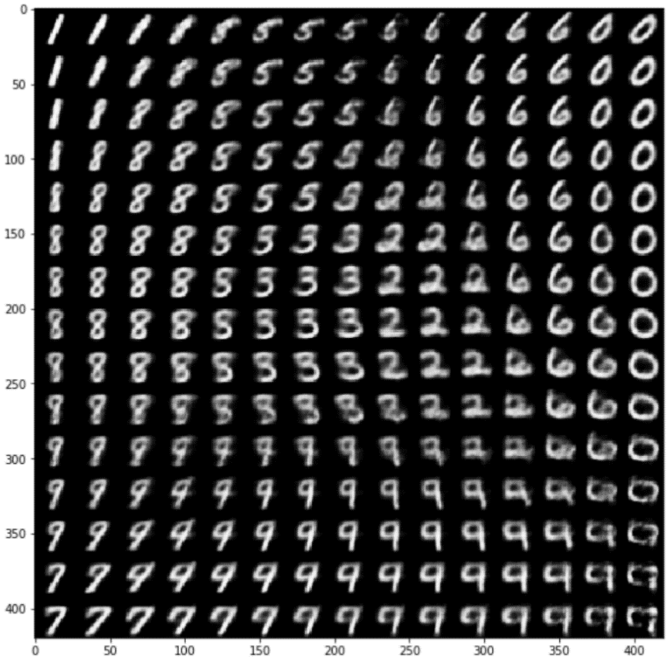
- VAE는 GAN에 비해 학습이 안정적인 편
  (왜? Reconstruction error와 같이 평가 기준이 명확

- 내재한 잠재변수 z도 함께 학습 할 수 있는 장점(feature learning)

단점

- 오른쪽 결과 처럼 출력이 선명하지 않고 평균값 형태로 표시되는 문제

- reparameterization trick이 모든 경우에 적용되지 않는 문제



Q&A