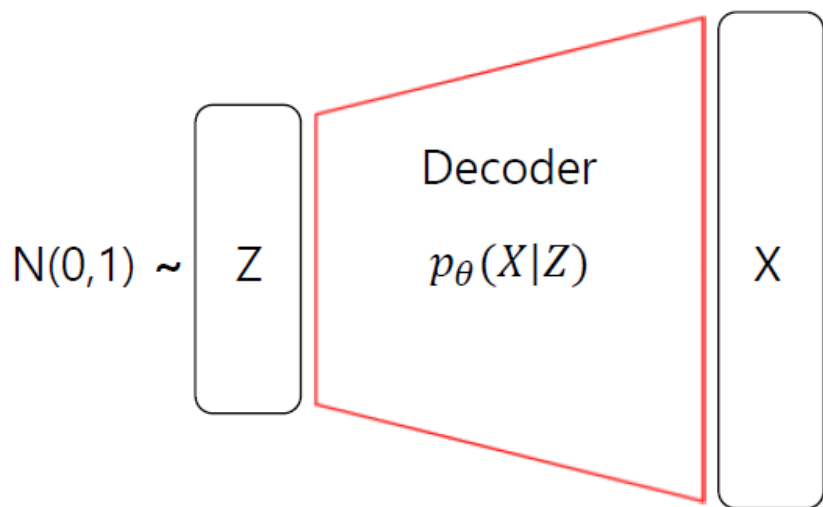


Variational Auto Encoder

이봉석

01. VAE란?

우리의 목표는 Normal distribution $N(0,1)$ 에서 Z 를 샘플링을 한 후 이를 이용하여 X 라는 데이터를 생성하는 것이다.
즉 그림으로 표현을 하면 아래와 같다.



$p_{\theta}(X|Z)$ 를 Gaussian distribution으로 가정하게 되면

이 문제는 $p(x) = \int p(x|g_{\theta}(z))p(z)dz$ 를 최대화 해주는 MLE문제가 된다.

하지만,

01. VAE란?

straightforward to compute $P(X)$ approximately: we first sample a large number of z values $\{z_1, \dots, z_n\}$, and compute $P(X) \approx \frac{1}{n} \sum_i P(X|z_i)$. The problem here is that in high dimensional spaces, n might need to be extremely large before we have an accurate estimate of $P(X)$. To see why, consider our example of handwritten digits. Say that our digit datapoints are stored in pixel space, in 28×28 images as shown in Figure 3. Since $P(X|z)$ is an isotropic Gaussian, the negative log probability of X is proportional squared Euclidean distance between $f(z)$ and X . Say that Figure 3(a) is the target (X) for which we are trying to find $P(X)$. A model which produces the image shown in Figure 3(b) is probably a bad model, since this digit is not much like a 2. Hence, we should set the σ hyperparameter of our Gaussian distribution such that this kind of erroneous digit does not contribute to $P(X)$. On the other hand, a model which produces Figure 3(c) (identical to X but shifted down and to the right by half a pixel) might be a good model.

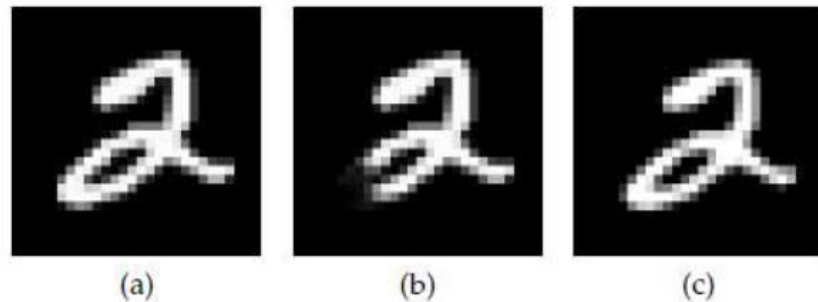


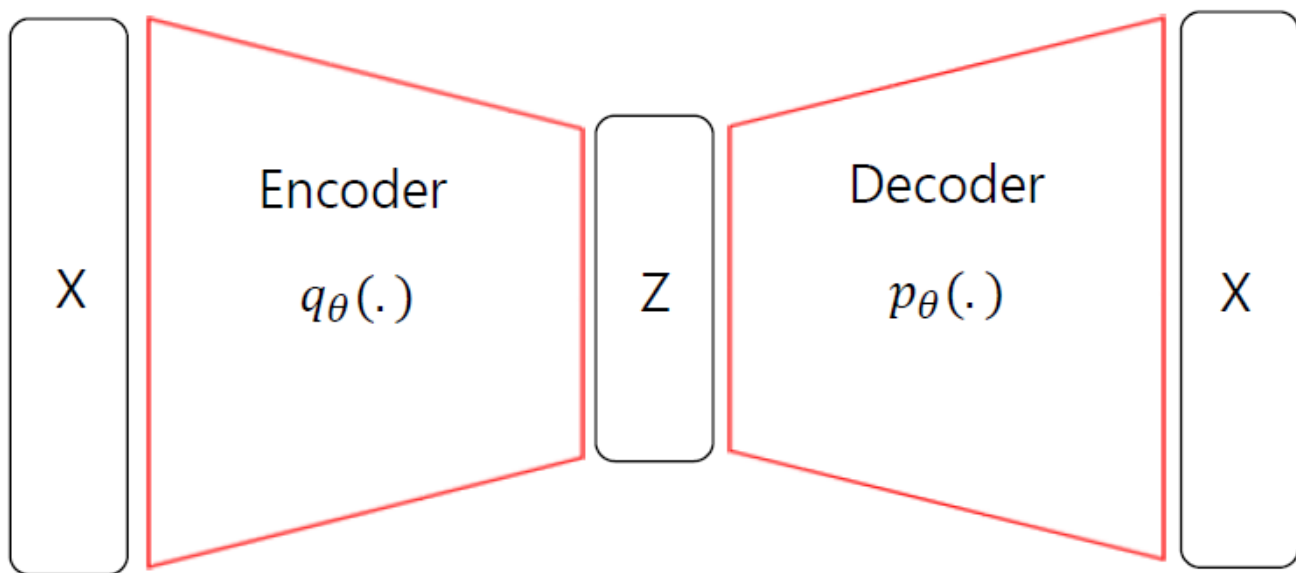
Figure 3: It's hard to measure the likelihood of images under a model using only sampling. Given an image X (a), the middle sample (b) is much closer in Euclidean distance than the one on the right (c). Because pixel distance is so different from perceptual distance, a sample needs to be extremely close in pixel distance to a datapoint X before it can be considered evidence that X is likely under the model.

위의 논문에도 나와있듯 $P(X|Z)$ 를 Gaussian distribution으로 가정하고, $-\log(P(X))$ 를 Minimize 할 시 데이터 사이의 Euclidian distance가 작아지도록 학습이 일어나 학습이 정확한 방향으로 되지 않는다.

Figure 3의 (a)라는 데이터를 이용해 학습을 했다고 하면 (c)라는 결과가 나와야 좋은데, Euclidian distance가 가까운 (b)가 생성된다.

01. VAE란?

이를 해결하기 위해 Z 를 무작정 $N(0,1)$ 에서 샘플링 하는 것이 아닌, 내가 가지고 있는 Input data X 를 이용하여 Sampling을 한 것이 Variational AutoEncoder 이다.



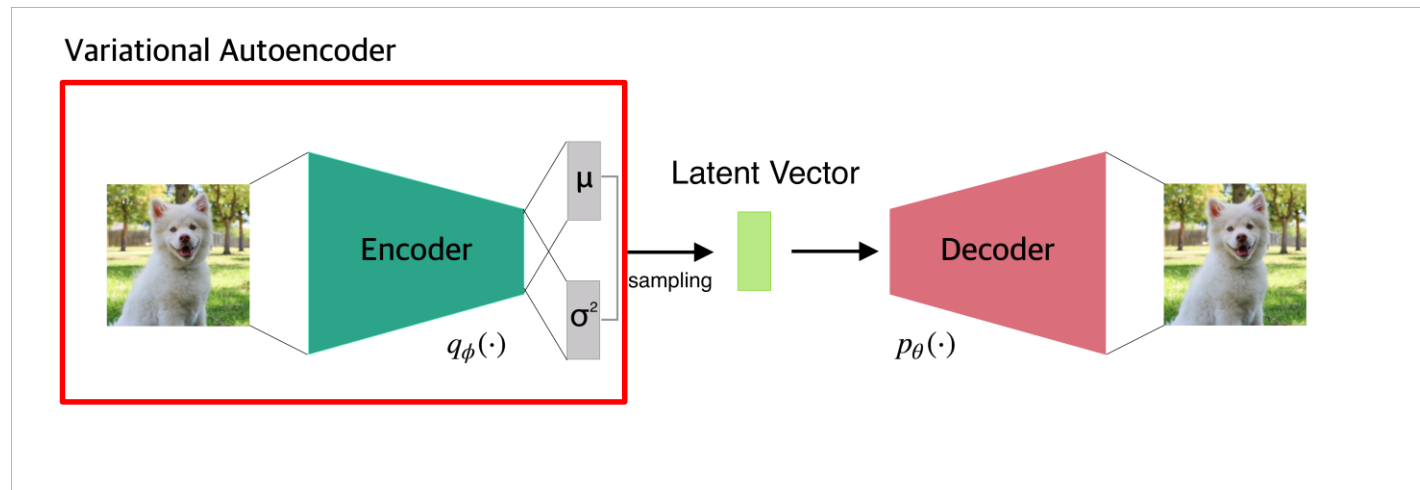
02. VAE 목표

1. 주어진 데이터를 잘 설명하는 잠재변수의 분포를 찾는 것(Encoder의 역할)
2. 잠재변수로부터 원본 이미지와 같은 이미지를 잘 복원하는 것(Decoder의 역할)

03. Encoder

• Encoder의 역할

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



But!!

- 어떤 것이 이상적인 확률분포 인지를 알 수 없다.

03. Encoder

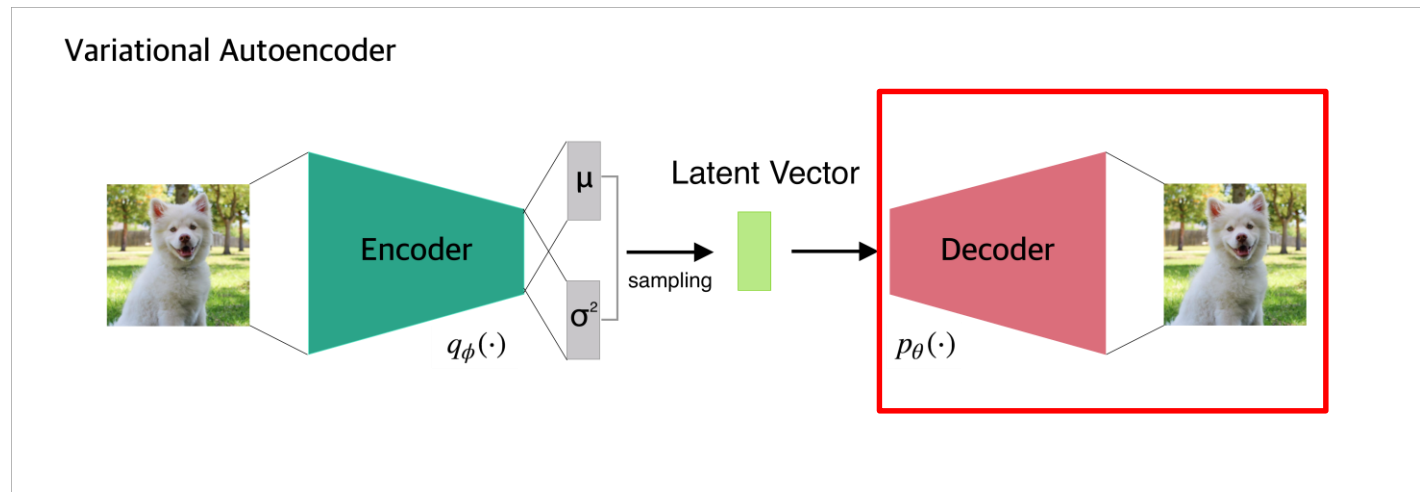


- 변분추론이란, 계산이 어려운 확률분포를 추정하기 위해서 다루기 쉬운 분포(approximation class, q_{ϕ})를 가정
- Encoder는 ϕ 라는 파라미터들을 바꾸어가며, $q_{\phi}(z|x)$ 확률 분포를 이상적인 확률분포 $p(z|x)$ 에 근사시키는 역할을 수행
- 보통 q_{ϕ} 은 Gaussian 분포라 가정한다. 그리고 이 때 z 의 marginal distribution은 평균이 0이고 분산이 1인 표준정규분포로 가정

04. Decoder

- Decoder의 역할

- 추출한 샘플을 입력으로 받아, 다시 원본으로 재구축하는 역할을 수행



04. Decoder

$P(x)$ 는 실제 데이터의 분포

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

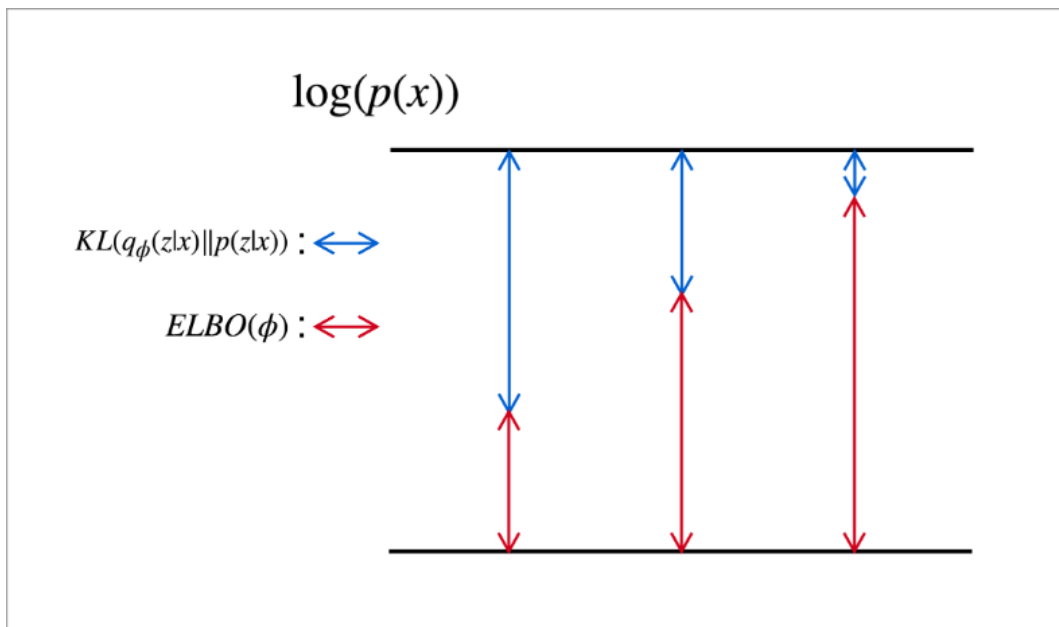
Jensen's Inequality

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

ELBO

- 최종 식의 우변식이 ELBO(Evidence LowerBOund)라고 한다.
- 이 $\text{ELBO}(\phi)$ 값을 최대화하는 ϕ 를 찾으면 최종식의 우변과 좌변을 같게 된다.


04. Decoder



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

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


 $ELBO(\phi)$


 $KL(q_\phi(z|x)||p(z|x))$

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

05. Loss

$$\begin{aligned} \int \log(p(x|z)) q_\phi(z|x) dz &= \mathbb{E}_{q_\phi(z|x)}[\log(p(x|z))] \\ \int \log\left(\frac{q_\phi(z|x)}{p(z)}\right) q_\phi(z|x) dz &= KL(q_\phi(z|x) \| p(z)) \end{aligned} \quad \longrightarrow \quad ELBO(\phi) = \mathbb{E}_{q_\phi(z|x)}[\log(p(x|z))] - KL(q_\phi(z|x) \| p(z))$$

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

따라서, 최종 VAE의 Loss 함수는 아래와 같다

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

Reconstruction Error

- 현재 샘플링 용 함수에 대한 negative log likelihood
- x_i 에 대한 복원 오차 (AutoEncoder 관점)

Regularization

- 현재 샘플링 용 함수에 대한 추가 조건
- 샘플링의 용의성/생성 데이터에 대한 통제성을 위한 조건을 prior에 부여 하고 이와 유사해야 한다는 조건을 부여

05. Loss

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

Assumption 1

- $q_{\phi} \models$ Gaussian distribution

$$q_{\phi}(z|x_i) \sim N(\mu_i, \sigma_i^2 I)$$

Assumption 2

- $p(z) \models$ standard normal distribution

$$p(z) \sim N(0, I)$$

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

Kullback–Leibler divergence [\[edit\]](#)

The Kullback–Leibler divergence from $\mathcal{N}_0(\mu_0, \Sigma_0)$ to $\mathcal{N}_1(\mu_1, \Sigma_1)$, for non-singular matrices Σ_0 and Σ_1 , is:^[8]

$$D_{\text{KL}}(\mathcal{N}_0 || \mathcal{N}_1) = \frac{1}{2} \left\{ \text{tr}(\overset{\text{posterior}}{\Sigma_1^{-1} \Sigma_0}) + (\overset{\text{prior}}{\mu_1} - \mu_0)^T \Sigma_1^{-1} (\mu_1 - \mu_0) - k + \ln \frac{|\Sigma_1|}{|\Sigma_0|} \right\},$$

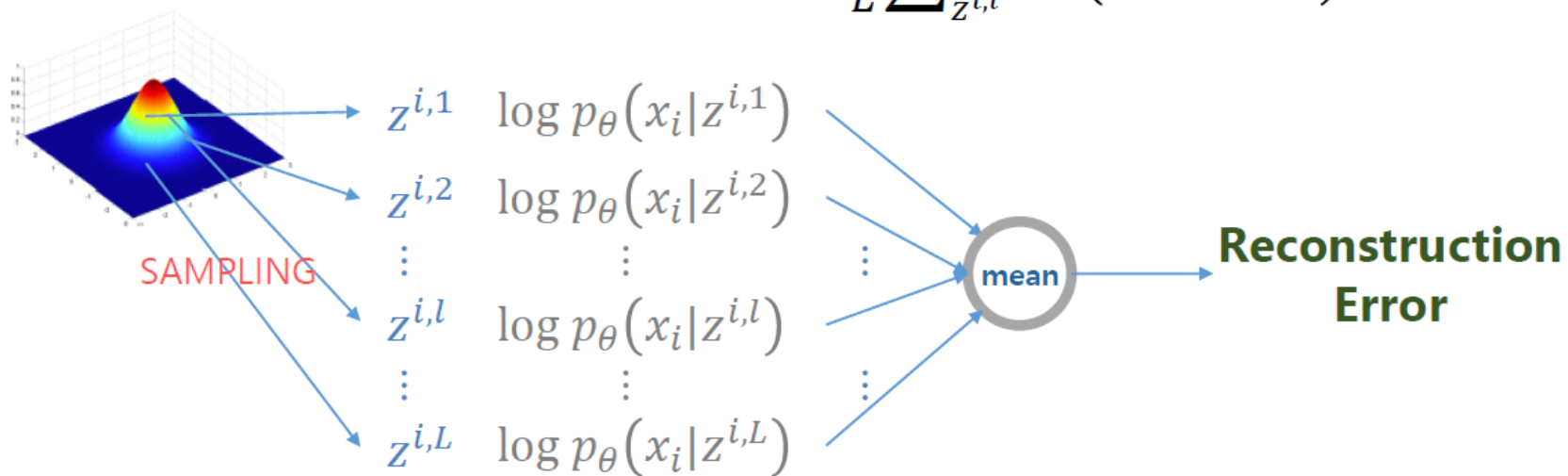
where k is the dimension of the vector space.

05. Loss

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

$$\mathbb{E}_{q_{\phi}(z|x_i)}[\log(p_{\theta}(x_i|z))] = \int \log(p_{\theta}(x_i|z)) q_{\phi}(z|x_i) dz$$

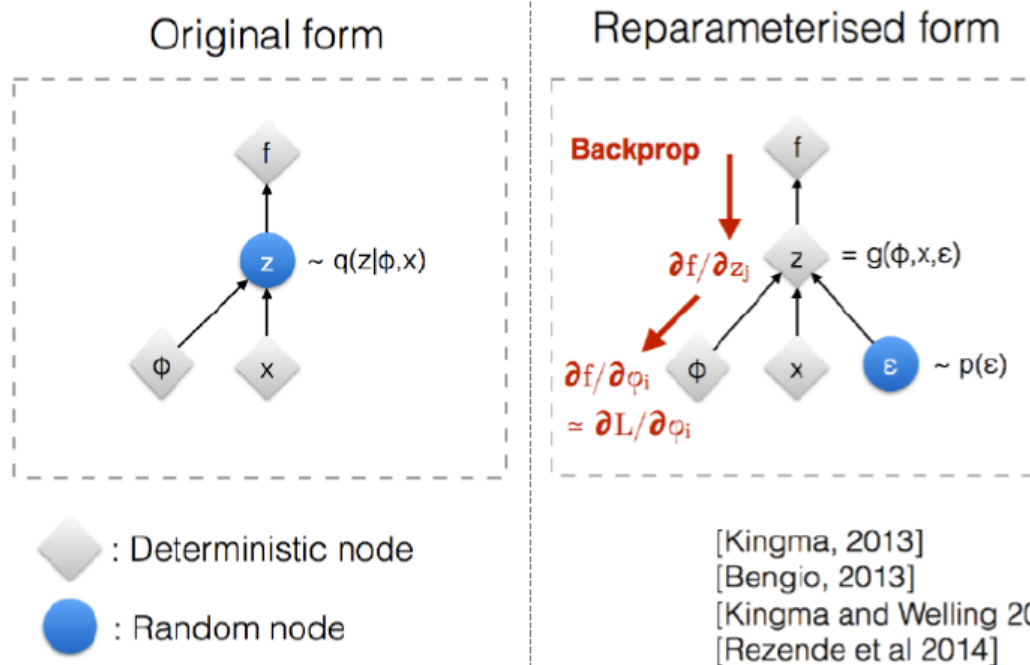
Monte-carlo technique $\rightarrow \approx \frac{1}{L} \sum_{z^{i,l}} \log(p_{\theta}(x_i|z^{i,l}))$



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

05. Loss

Reparameterization Trick



Sampling Process

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



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

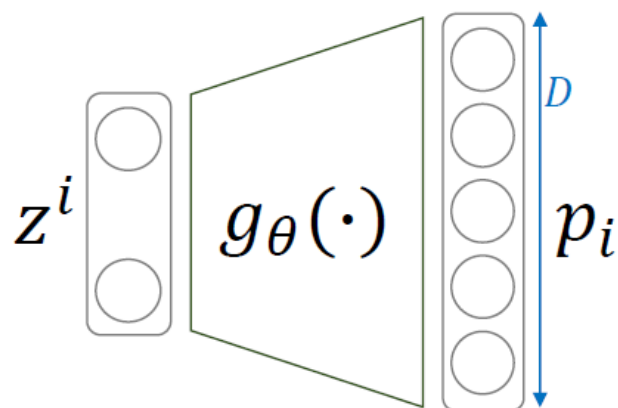
Same distribution!
But it makes backpropagation possible!!

05. Loss

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

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

↑
Monte-carlo
technique
↑
L=1



$$p_{\theta}(x_i|z^i) \sim \text{Bernoulli}(p_i)$$

Assumption 3-1

[Decoder, likelihood]

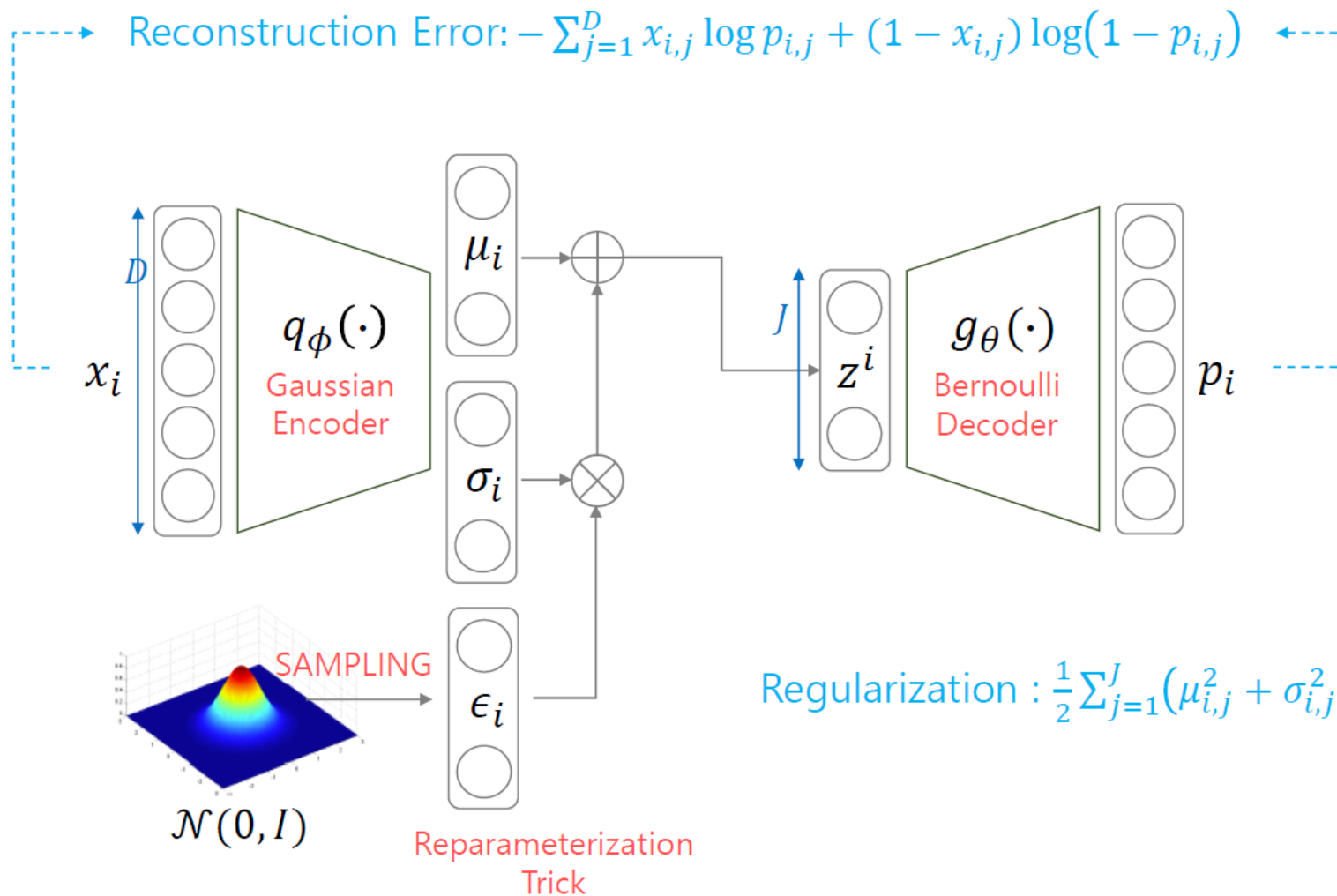
multivariate bernoulli or gaussian distribution

$$\begin{aligned} \log(p_{\theta}(x_i|z^i)) &= \log \prod_{j=1}^D p_{\theta}(x_{i,j}|z^i) = \sum_{j=1}^D \log p_{\theta}(x_{i,j}|z^i) \\ &= \sum_{j=1}^D \log p_{i,j}^{x_{i,j}} (1 - p_{i,j})^{1-x_{i,j}} \leftarrow p_{i,j} \triangleq \text{network output} \\ &= \sum_{j=1}^D x_{i,j} \log p_{i,j} + (1 - x_{i,j}) \log(1 - p_{i,j}) \leftarrow \text{Cross entropy} \end{aligned}$$

05. Loss

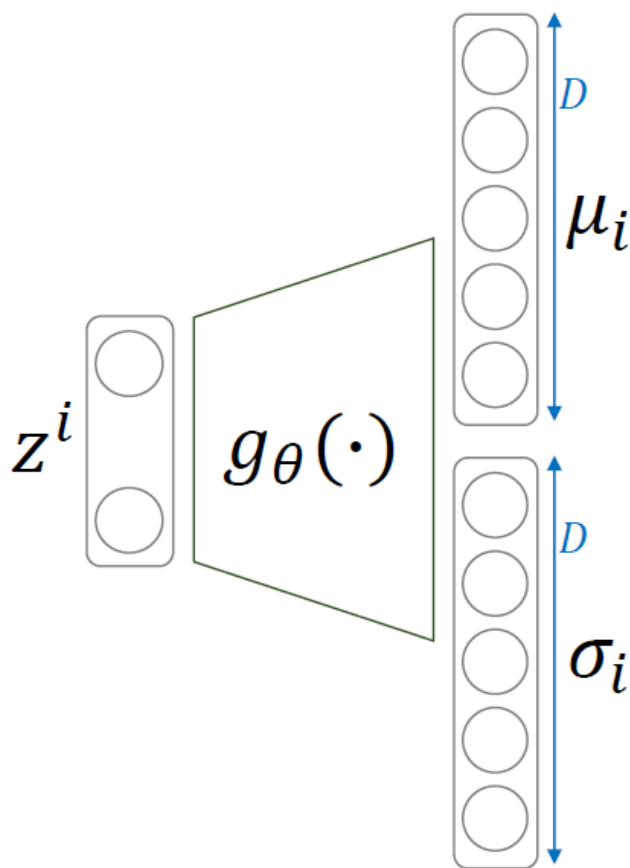
STRUCTURE

Default : Gaussian Encoder + Bernoulli Decoder



05. Loss

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



$$\mathbb{E}_{q_{\phi}(z|x_i)} [\log(p_{\theta}(x_i|z))] \approx \log(p_{\theta}(x_i|z^i))$$

Assumption 3-2

[Decoder, likelihood]

multivariate bernoulli_or gaussian distribution

$$\begin{aligned} \log(p_{\theta}(x_i|z^i)) &= \log(N(x_i; \mu_i, \sigma_i^2 I)) \\ &= - \sum_{j=1}^D \frac{1}{2} \log(\sigma_{i,j}^2) + \frac{(x_{i,j} - \mu_{i,j})^2}{2\sigma_{i,j}^2} \end{aligned}$$

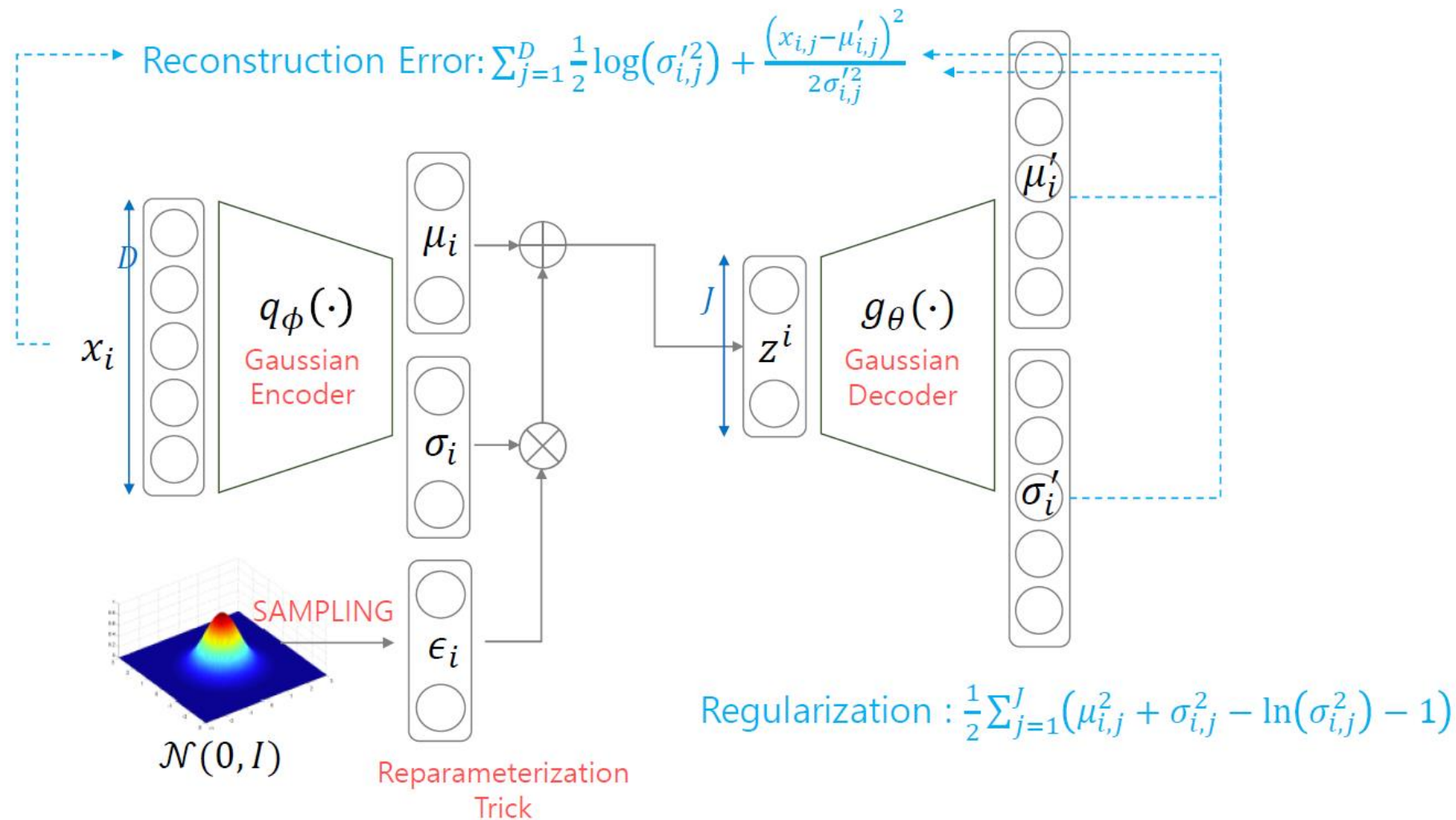
For gaussian distribution with identity covariance

$$\log(p_{\theta}(x_i|z^i)) \propto - \sum_{j=1}^D (x_{i,j} - \mu_{i,j})^2 \quad \leftarrow \text{Squared Error}$$

05. Loss

STRUCTURE

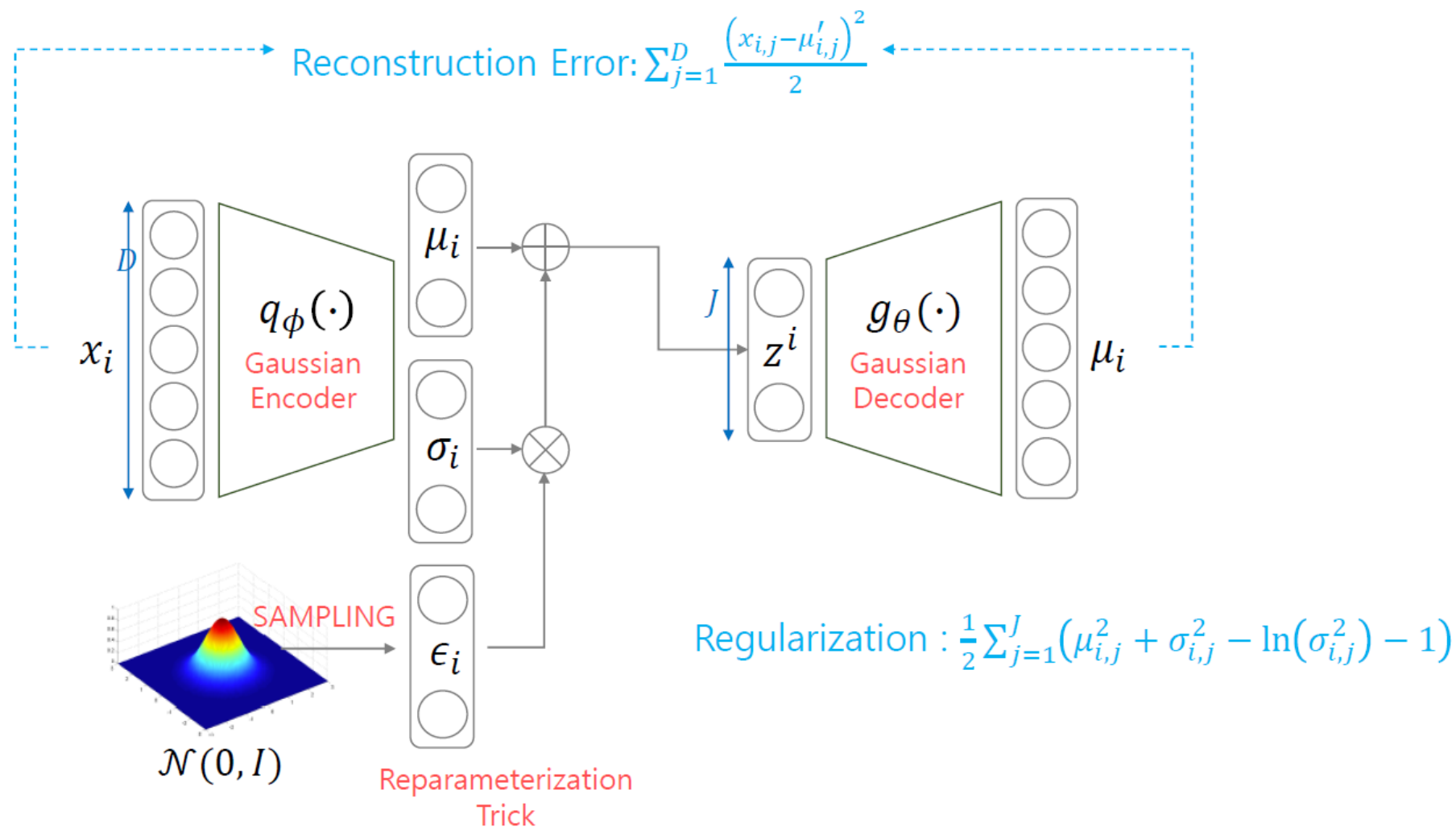
Gaussian Encoder + Gaussian Decoder



05. Loss

STRUCTURE

Gaussian Encoder + Gaussian Decoder with Identity Covariance



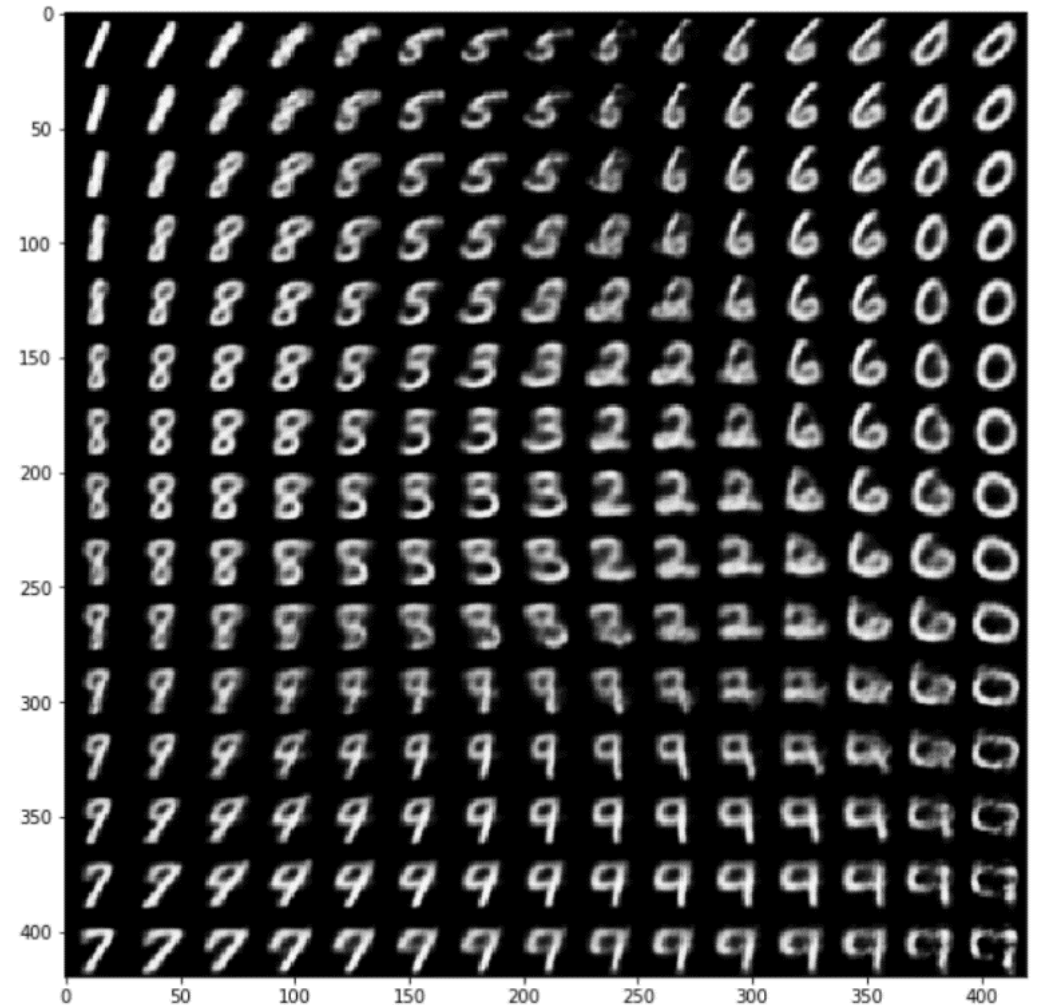
06. VAE의 장단점

장점

- VAE는 GAN에 비해 학습이 안정적인 편
- 내재한 잠재 변수 z 도 함께 학습 할 수 있다.(feature learning)

단점

- 오른쪽 결과처럼 출력이 선명하지 않고 평균값 형태로 출력되는 문제



감사합니다.