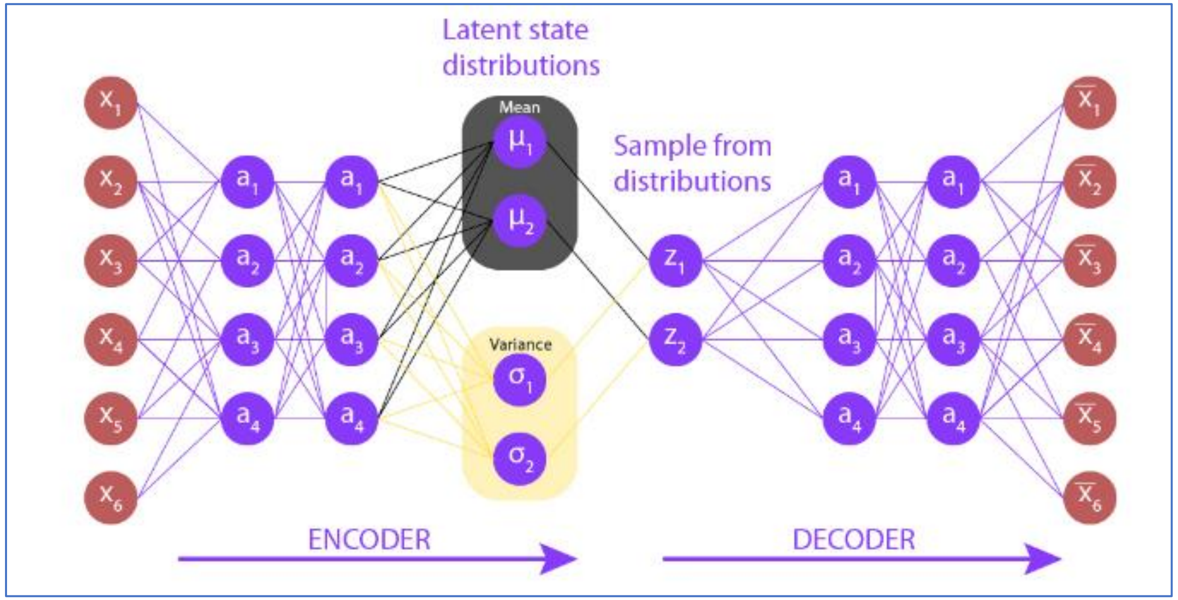
Variational Autoencoder

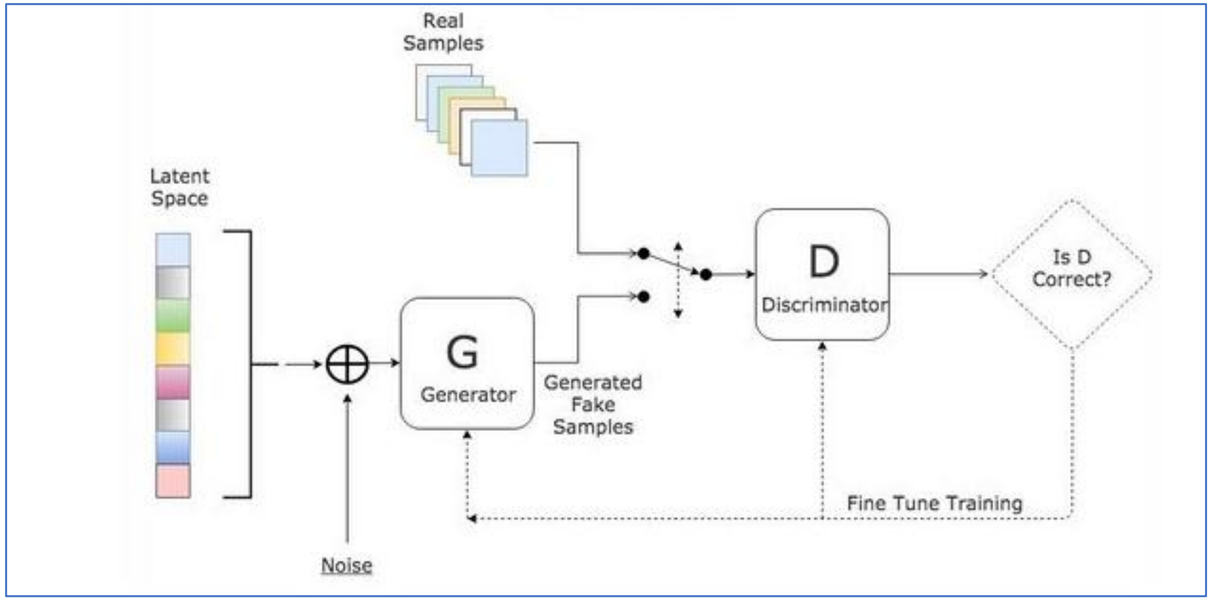
박종혁

Online Seminar

Background

Generation model

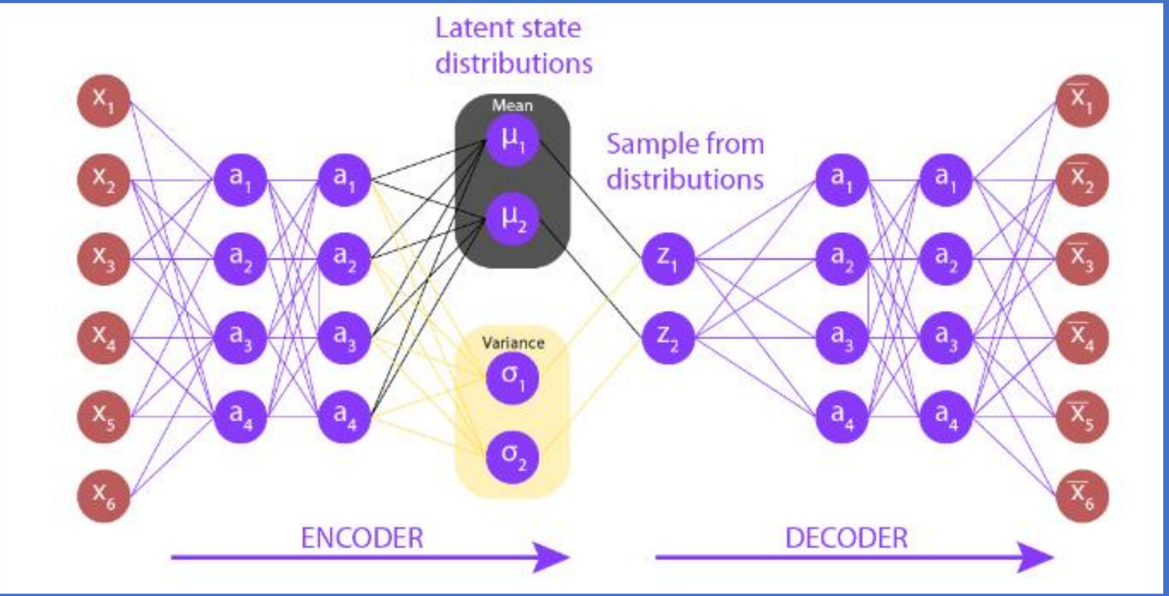


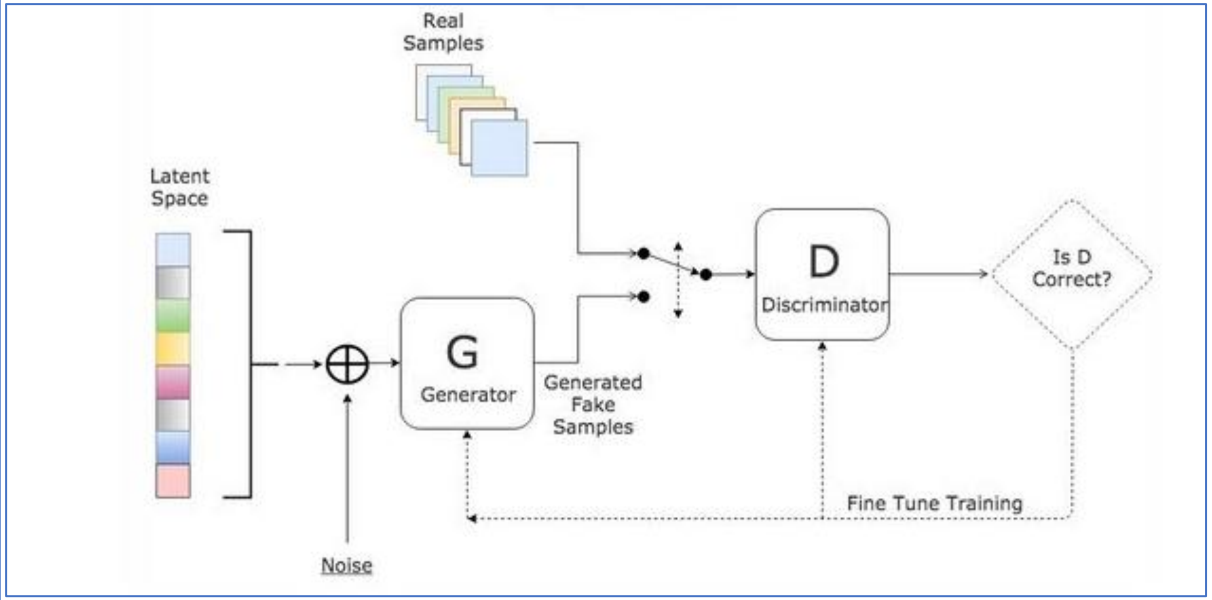


How does the model generate data?

Background

Generation model





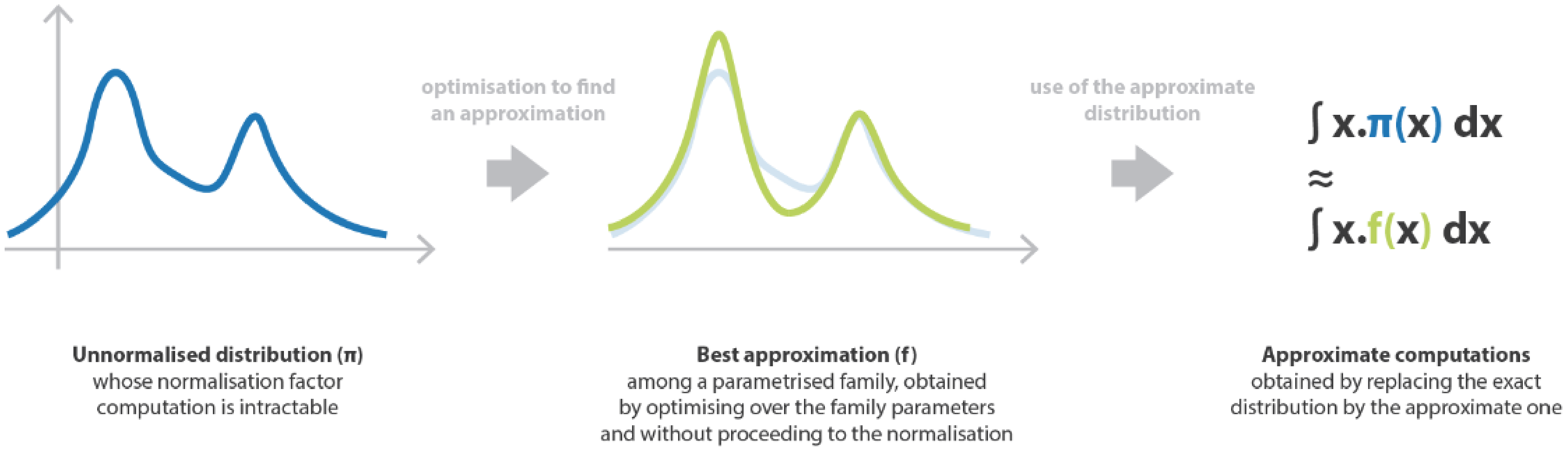
How does the model generate data?

Background

Variational Inference

What is Variational Inference?

사후확률(Posterior) 분포 $p(z|x)$ 를 우리가 다루기 쉬운 확률분포 $q(z)$로 근사하는 방법.



Background

Variational Inference

Why approximate the posterior probability?

→ Hard to Compute⋯

$$P(Z|X) = \frac{P(Z,X)}{\int_t P(Z_t|X)}$$

일반적으로 분모의 부분을 계산하기란 매우 어렵다.

Approximate

How can we approximate the posterior distribution?

Method 1 : MCMC(Markov chain Monte Carlo)

Method 2 : Variational Inference

Method 3 : Laplace's Method

Approximate

How can we approximate the posterior distribution?

Method 1 : MCMC(Markov chain Monte Carlo)

Method 2 : Variational Inference

Method 3 : Laplace's Method

Variational Inference

Use KL divergence between two distributions : $\mathrm{p}\ and\ q$

$$D_{KL}(q||p) = E_q[log\,\frac{q(Z)}{p(Z|x)}]$$

We actually can't minimize the KL divergence exactly, but we can minimize a function that is equal to it up to a constant.

This is the evidence lower bound(ELBO)

Variational Inference

Compute evidence lower bound

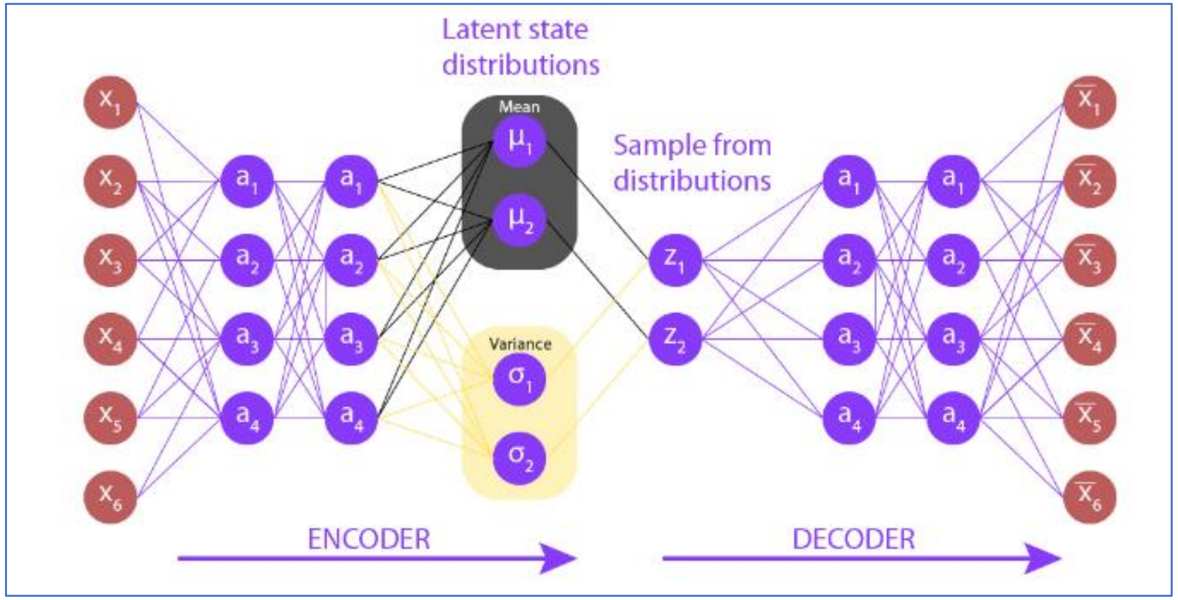
$$
\begin{aligned}
\log p(x) &= \log \int_z p(x,z) \\
&= \log \int_z p(x,z) \frac{q(z)}{q(z)} \\
&= \log(E_q \left[\frac{p(x,Z)}{q(z)} \right]) \\
&\geq E_q[\log p(x,Z)] - E_q[\log q(Z)]
\end{aligned}
$$

Variational Autoencoder

Architecture

VAE

Variational AutoEncoder has two part : Encoder & Decoder



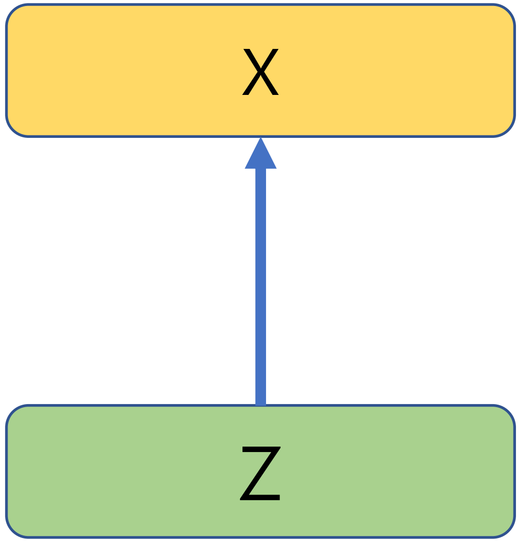
Decoder Network

VAE

Assume training data $\{x^{(i)}\}_{i=1}^{N}$ is generated from underlying unobserved(latent) representation z

Sample from true condition

$p_\theta(x|z)$



Sample from true prior

$p_\theta(z)$

We want to estimate the true parameter θ of this generative model

Q : How to train the model?

Maximize likelihood of training data

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

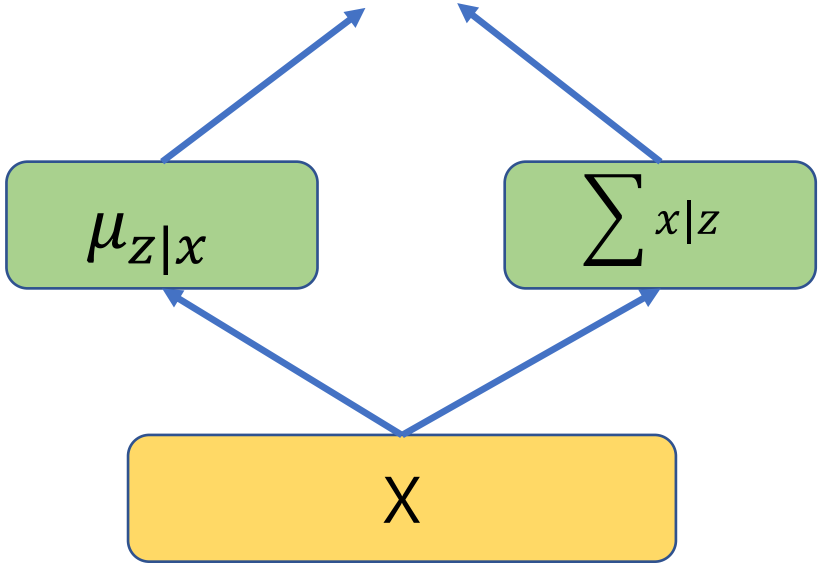
However, it is intractable

Encoder Network

VAE

Since we're modeling probabilistic generation of data, encoder and decoder network are probabilistic

$$Sample\ z\ from\ \ z|x \sim N(\mu_{z|x}, \Sigma_{z|x})$$



$$Approximate:$$

$$encoder\ network\ q_{\phi}(z|x)\ \rightarrow p_{\theta}(z|x)$$

Optimization

VAE

Now equipped with our encoder and decoder networks, let's work out the (log) data likelihood:

$$\log p_\theta(x^{(i)}) = \mathbf{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right] \qquad (p_\theta(x^{(i)}) \text{ Does not depend on } z)$$

$$= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\right] \qquad \text{(Bayes' Rule)}$$

$$= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\frac{q_\phi(z \mid x^{(i)})}{q_\phi(z \mid x^{(i)})}\right] \qquad \text{(Multiply by constant)}$$

$$= \mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z)}\right] + \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z \mid x^{(i)})}\right] \qquad \text{(Logarithms)}$$

$$= \mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z)) + D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z \mid x^{(i)}))$$

Decoder network gives $p_\theta(x|z)$, can compute estimate of this term through sampling. (Sampling differentiable through reparam. trick, see paper.)

This KL term (between Gaussians for encoder and z prior) has nice closed-form solution!

$p_\theta(z|x)$ intractable (saw earlier), can't compute this KL term :(But we know KL divergence always >= 0.

Optimization

VAE

Now equipped with our encoder and decoder networks, let's work out the (log) data likelihood:

$$\log p_\theta(x^{(i)}) = \mathbf{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right] \qquad (p_\theta(x^{(i)}) \text{ Does not depend on } z)$$

$$= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\right] \qquad \text{(Bayes' Rule)}$$

$$= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\frac{q_\phi(z \mid x^{(i)})}{q_\phi(z \mid x^{(i)})}\right] \qquad \text{(Multiply by constant)}$$

$$= \mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z)}\right] + \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z \mid x^{(i)})}\right] \qquad \text{(Logarithms)}$$

$$= \boxed{\underbrace{\mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z))}_{\mathcal{L}(x^{(i)},\theta,\phi)}} + \underbrace{D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z \mid x^{(i)}))}_{\boxed{\geq 0}}$$

Tractable lower bound which we can take gradient of and optimize! ($p_\theta(x|z)$ differentiable, KL term differentiable)

Optimization

VAE

Now equipped with our encoder and decoder networks, let's work out the (log) data likelihood:

$$
\begin{aligned}
\log p_\theta(x^{(i)}) &= \mathbf{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right] \qquad (p_\theta(x^{(i)}) \text{ Does not depend on } z) \\
&= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\right] \qquad (\text{Bayes' Rule}) \\
&= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\frac{q_\phi(z \mid x^{(i)})}{q_\phi(z \mid x^{(i)})}\right] \qquad (\text{Multiply by constant}) \\
&= \mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z)}\right] + \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z \mid x^{(i)})}\right] \qquad (\text{Logarithms}) \\
&= \underbrace{\mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z))}_{\mathcal{L}(x^{(i)},\theta,\phi)} + \underbrace{D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z \mid x^{(i)}))}_{\geq 0}
\end{aligned}
$$

$$\boxed{\log p_\theta(x^{(i)}) \geq \mathcal{L}(x^{(i)},\theta,\phi)}$$

Variational lower bound ("ELBO")

$$\boxed{\theta^*,\phi^* = \arg\max_{\theta,\phi}\sum_{i=1}^{N}\mathcal{L}(x^{(i)},\theta,\phi)}$$

Training: Maximize lower bound

Optimization

VAE

$$Loss = \boxed{-E_z\left[\log p_\theta\left(x^{(i)}\middle|z\right)\right]} - \boxed{D_{KL}\left(q_\phi\left(z\middle|x^{(i)}\right)||P_\theta(z)\right)}$$

<Reconstruction Error> <Regularization>

$$= D_{KL}(N(\mu, \Sigma)||N(0, I))$$
$$= -\frac{1}{2}\sum_{j=1}^{J}\left(1 + \log\left(\sigma_j^2\right) - \mu_j^2 + \sigma_j^2\right)$$

Reconstruction Error :

[1] 현재 샘플링된 z에 대한 negative loglikelihood

[2] x에 대한 복원 오차

Regularization :

[1] 현재 샘플된 z에 대한 추가 조건

[2] 샘플링되는 z들에 대한 통제성을 prior를 통해 부여, vaiational distribution $q(z|x)$ 가 $p(z)$와 유사해야 한다는 조건을 부여

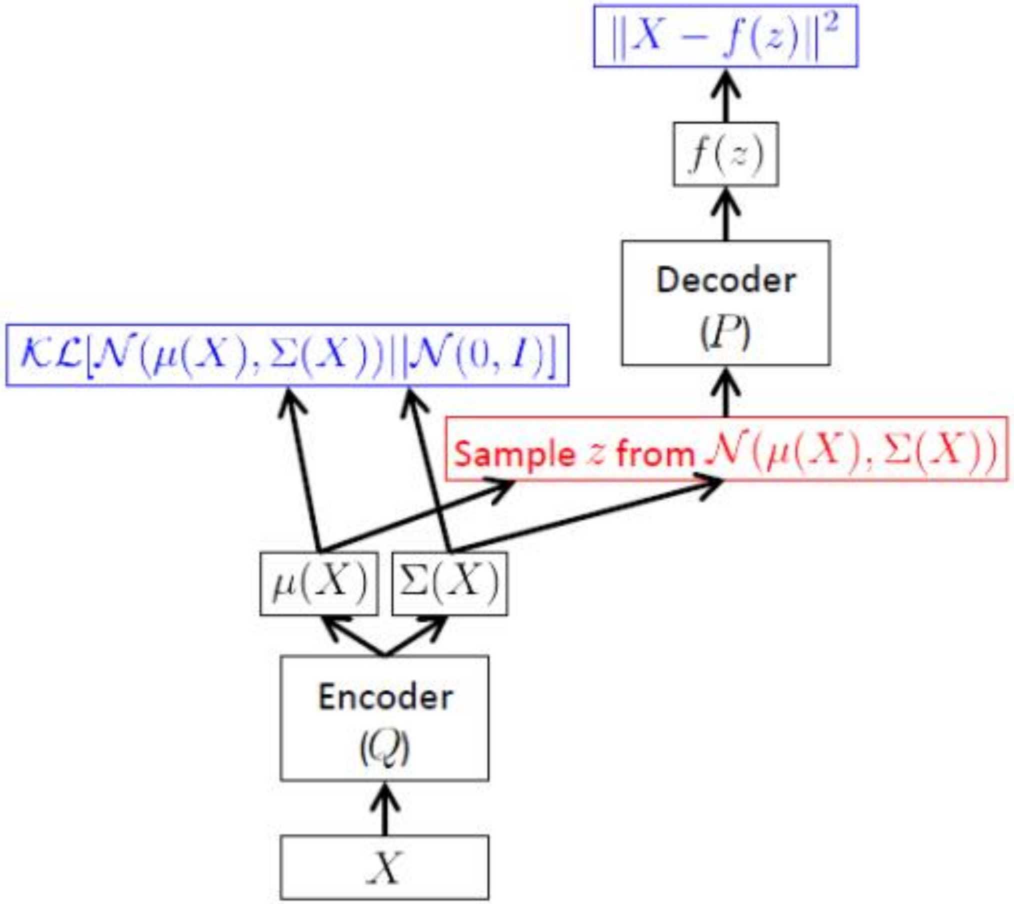
Summary

VAE

Encoder Network $q(z|x)$ **:**
- **Calculate mean and variance of data X**
- **Approximate** $p(z|x)$

Decoder Network $p(x|z)$ **:**
- **Generate a data given z**

Prior distribution $p(z)$ **:**
- Usually a N(0,1) distribution



Issue

Posterior Collapse

Posterior Collapse란?

- 요약 : LSTM과 같은 auto-regressive 모델을 decode에 사용하는 경우, latent vector를 무시해버리는 문제
- Z와는 관계없이 Reconstruction Error만 줄이는 방법으로 학습이 진행 → Encoder까지 학습 X
- 따라서 Encoder는 KL term에 의해서만 학습이 진행되며, 의미 있는 z를 생성하지 못하고 Prior와 동일한 분포를 가지게 됨 → KL term = 0
- KL term = 0 → **각기 다른 문장에 대해서 같은 latent vector를 사용하므로 VAE를 사용하는 의미가 사라짐.**

$$Loss = \boxed{-E_z\left[\log p_\theta\left(x^{(i)}|z\right)\right]} - \boxed{D_{KL}\left(q_\phi\left(z|x^{(i)}\right)||P_\theta(z)\right)}$$

<Reconstruction Error> <Regularization>

Thank you!

Reference

- https://www.sallys.space/blog/2018/02/08/variational-inference-naver-connect/
- https://www.cs.princeton.edu/courses/archive/fall11/cos597C/lectures/variational-inference-i.pdf
- https://towardsdatascience.com/bayesian-inference-problem-mcmc-and-variational-inference-25a8aa9bce29
- https://ratsgo.github.io/generative%20model/2018/01/27/VAE/
- https://redstarhong.tistory.com/77
- https://www.jeremyjordan.me/variational-autoencoders/
- https://www.geeksforgeeks.org/variational-autoencoders/
- https://www.slideshare.net/ssuser06e0c5/variational-autoencoder-76552518