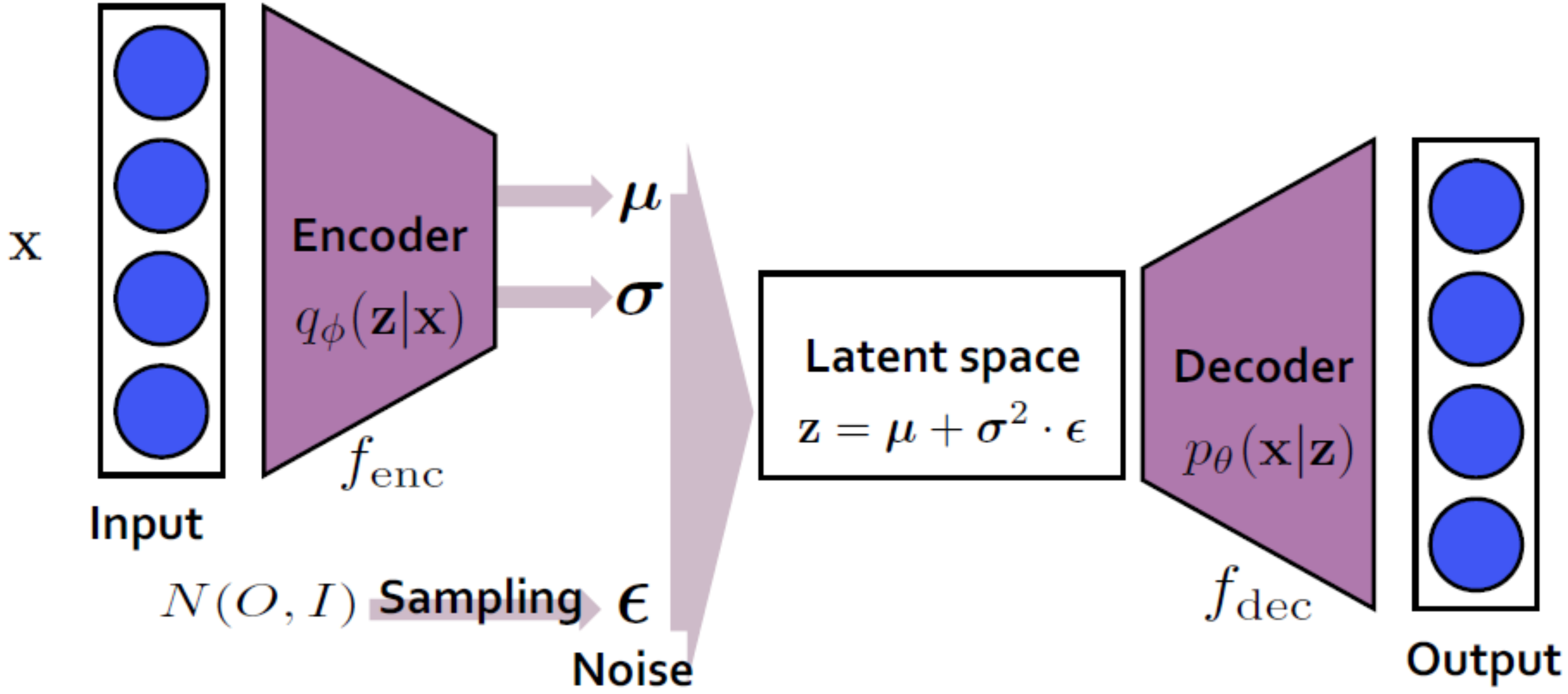
Outta 학술부 백강현

Auto-Encoding Variational Bayes

Abstract, introduction

- How can we perform efficient inference and learning in directed probabilistic models, in the presence of continuous latent variables with intractable posterior distributions, and large datasets?
- 확률 모델을 통해 데이터 생성(p(x))
- 데이터 생성에 잠재 변수 z를 도입(p(x|z))
- 잠재 변수의 posterior distribution(p(z|x)) 구하기 ->intractable하므로 q(z|x)로 근사

Abstract, Introduction



2.1 Problem scenario

- $X = \{x^{(i)}\}_{i=1}^{N}$: dataset consisting of N i.i.d. samples of some variable x.
  학습에 사용되는 데이터이기도 하면서 최종적으로 만들어야 할 데이터
- z: latent continuous variables
  데이터를 만드는데 조건이 되는 변수(잠재 변수)

- generating process

1) prior distribution $p_{\theta^*}(z)$를 통해 z를 generate
2) conditional distribution $p_{\theta^*}(x|z)$를 통해 x를 generate

2.1 Problem scenario

(a) Learned Frey Face manifold

z_2

z_1

예시)

z_1: 사람이 쳐다보는 방향
z_2: 사람이 웃는 정도

*베이즈 정리

- $p(z|x) = \frac{p(x|z)p(z)}{p(x)}$
- x: 주어진 대상(관측값)
- z: 구하고자 하는 대상
- $p(z)$: prior(사전확률)
- $p(x|z)$: likelihood
- $p(z|x)$: posterior(사후확률)

* ML(maximum likelihood)와 MAP(maximum a posteriori)

- ML: likelihood ($p(x|z)$)를 최대화 하는 것
- MAP: posterior($p(z|x)$)를 최대화 하는 것

- Ex) 바닥에 떨어진 머리카락의 길이(x)를 보고 그 머리카락이 남자 것인지 여자 것인지 성별(z)를 판단하는 문제
- ML: $p(x|남)$과 $p(x|여)$중 최댓값을 선택
- MAP: $p(남|x)$과 $p(여|x)$중 최댓값을 선택(posterior inference)
- ->MAP는 z의 분포를 고려하기 때문에 더 정확한 모델을 찾을 수 있음

2.1 Problem scenario

- 구해야 하는 것: θ^*, z
- 문제점

1) Intractability: posterior density($p(z|x)$)를 구할 수 없다.

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

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

-> sampling을 통해서 분포를 근사할 수 있지 않을까?

1) Large dataset: sampling based solutions(e.g. Monte Carlo EM)은 너무 느리다.

2.1 Problem scenario

- 해결책: variational inference

Posterior ($p_\theta(z|x)$)를 잘 근사하는 $q_\phi(z|x)$를 구한다.

$q_\phi(z|x)$를 우리가 잘 아는 분포로 가정하고 p, q의 분포의 차이를 줄인다(minimize $D_{KL}(q_\phi(z|x)||p_\theta(z|x))$.

* KL Divergence

$$D_{KL}(P \| Q) = \int_{-\infty}^{\infty} p(x) \log \frac{p(x)}{q(x)} dx$$

$$= \int_{-\infty}^{\infty} p(x) \log p(x) dx - \int_{-\infty}^{\infty} p(x) \log q(x) dx$$

2.2 The variational bound

- Marginal likelihood를 최대화하기

$$\log p_\theta(x) = \log p_\theta\left(x^{(1)}, x^{(2)}, \dots, x^{(N)}\right) = \sum_{i=1}^{N} \log p_\theta(x^{(i)})$$

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) = D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})||p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}^{(i)})) + \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)})$$

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

* ELBO(Evidence Lower Bound)

$$\log p_\theta(x) = \log p_\theta(x) \int q_\phi(z|x)\,dz = \int q_\phi(z|x) \log p_\theta(x)\,dz$$

$$= \log \frac{p_\theta(x|z)\,p(z)}{p_\theta(z|x)} = \log p_\theta(x|z) + \log p_\theta(z) - \log p_\theta(z|x)$$

$$= \int q_\phi(z|x) \log p_\theta(x|z) + \int q_\phi(z|x) \log p_\theta(z)\,dz - \int q_\phi(z|x) \log p_\theta(z|x)\,dz$$

$$\pm \int q_\phi(z|x) \log q_\phi(z|x)\,dz$$

$$= E_{q_\phi(z|x)}\left[\log p_\theta(x|z)\right] - \int q_\phi(z|x) \log \frac{q_\phi(z|x)}{p_\theta(z)}\,dz + \int q_\phi(z|x) \log \frac{q_\phi(z|x)}{p_\theta(z|x)}\,dz$$

$$= E_{q_\phi(z|x)}\left[\log p_\theta(x|z)\right] - D_{KL}\left(q_\phi(z|x) \,||\, p_\theta(z)\right) + D_{KL}\left(q_\phi(z|x) \,||\, p_\theta(z|x)\right)$$

* ELBO(Evidence Lower Bound)

$$\underbrace{\log p_\theta(x)}_{\uparrow} = \underbrace{D_{KL}\left(q_\phi(z|x) \,\|\, p_\theta(z|x)\right)}_{\downarrow} + \underbrace{E_{q_\phi(z|x)}\left[\log p_\theta(x|z)\right] - D_{KL}\left(q_\phi(z|x) \,\|\, p_\theta(z)\right)}_{\text{ELBO}}$$

계산 불가능

ELBO

계산 가능

$$= \mathcal{L}(\theta, \phi; x)$$

2.3 The SGVB estimator and AEVB algorithm (SGVB)

- 어떻게 계산할 것인가

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

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

- 평균 계산 시 몇 개의 z를 sampling해서 사용.

$$\widetilde{\mathbf{z}} \sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \quad \widetilde{\mathbf{z}} = g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \mathbf{x}) \quad \text{with} \quad \boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$$

$$\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})} \left[f(\mathbf{z}) \right] = \mathbb{E}_{p(\boldsymbol{\epsilon})} \left[f(g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \mathbf{x}^{(i)})) \right] \simeq \frac{1}{L} \sum_{l=1}^{L} f(g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}^{(l)}, \mathbf{x}^{(i)})) \quad \text{where} \quad \boldsymbol{\epsilon}^{(l)} \sim p(\boldsymbol{\epsilon})$$

2.3 The SGVB estimator and AEVB algorithm (SGVB)

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

$$\widetilde{\mathcal{L}}^A(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) = \frac{1}{L} \sum_{l=1}^{L} \log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}, \mathbf{z}^{(i,l)}) - \log q_{\boldsymbol{\phi}}(\mathbf{z}^{(i,l)} | \mathbf{x}^{(i)})$$

$$\text{where} \quad \mathbf{z}^{(i,l)} = g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}^{(i,l)}, \mathbf{x}^{(i)}) \quad \text{and} \quad \boldsymbol{\epsilon}^{(l)} \sim p(\boldsymbol{\epsilon})$$

2.3 The SGVB estimator and AEVB algorithm (SGVB)

- B)

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

$$\widetilde{\mathcal{L}}^{B}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) = -D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})||p_{\boldsymbol{\theta}}(\mathbf{z})) + \frac{1}{L}\sum_{l=1}^{L}(\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z}^{(i,l)}))$$

$$\text{where} \quad \mathbf{z}^{(i,l)} = g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}^{(i,l)}, \mathbf{x}^{(i)}) \quad \text{and} \quad \boldsymbol{\epsilon}^{(l)} \sim p(\boldsymbol{\epsilon})$$

- KL Divergence 계산
- 가정: $p_{\boldsymbol{\theta}}(\mathbf{z}) = \mathcal{N}(0, \mathbf{I})$ $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})$ are Gaussian.

$$-D_{KL}((q_{\boldsymbol{\phi}}(\mathbf{z})||p_{\boldsymbol{\theta}}(\mathbf{z})) = \int q_{\boldsymbol{\theta}}(\mathbf{z})\left(\log p_{\boldsymbol{\theta}}(\mathbf{z}) - \log q_{\boldsymbol{\theta}}(\mathbf{z})\right)\, d\mathbf{z}$$

$$= \frac{1}{2}\sum_{j=1}^{J}\left(1 + \log((\sigma_j)^2) - (\mu_j)^2 - (\sigma_j)^2\right)$$

2.3 The SGVB estimator and AEVB algorithm (AEVB)

- X가 N 개의 datapoints로 이루어질 때 , M 개의 datapoints를 뽑아 미니 배치를 만든 뒤 계산해줄 수 있다.

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{X}) \simeq \widetilde{\mathcal{L}}^M(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{X}^M) = \frac{N}{M} \sum_{i=1}^{M} \widetilde{\mathcal{L}}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)})$$

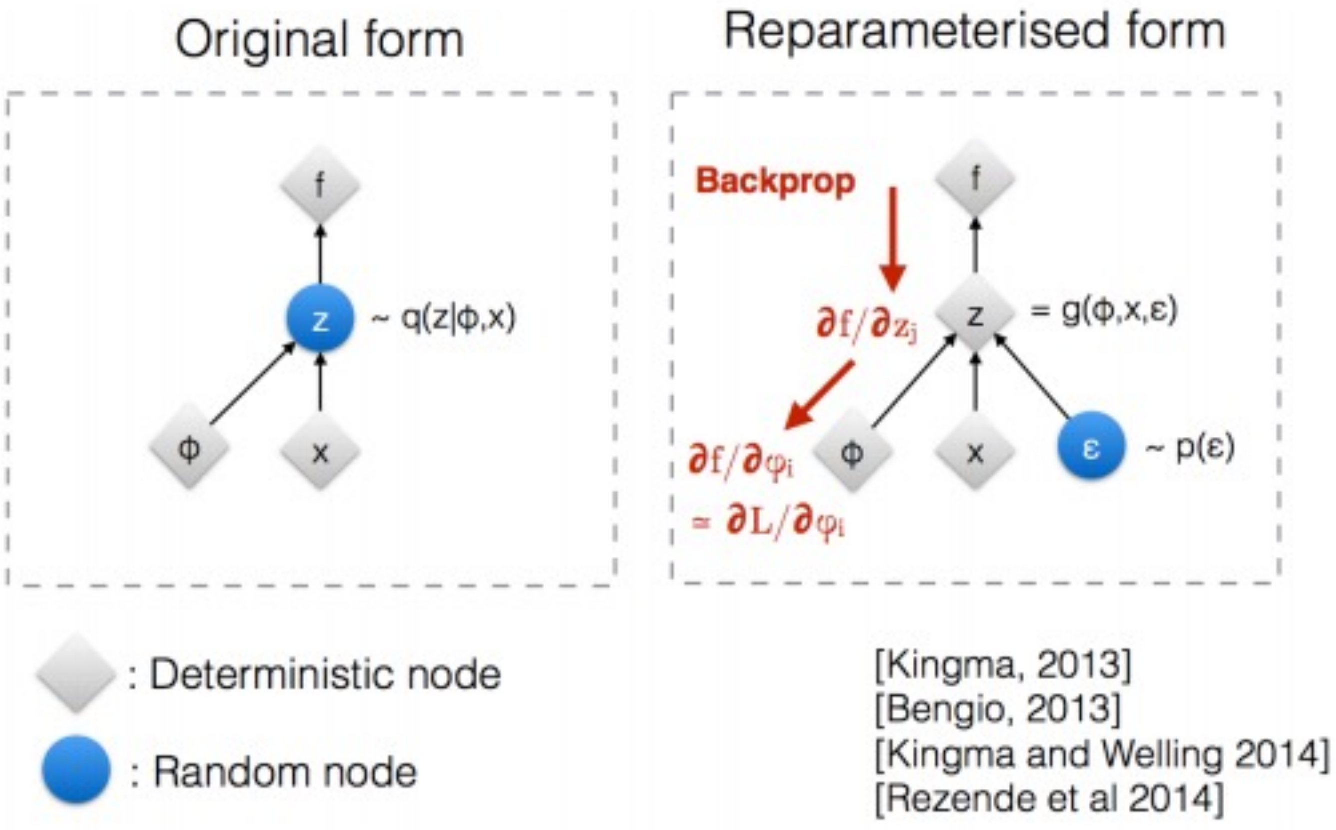
Algorithm 1 Minibatch version of the Auto-Encoding VB (AEVB) algorithm. Either of the two SGVB estimators in section 2.3 can be used. We use settings $M = 100$ and $L = 1$ in experiments.

$\boldsymbol{\theta}, \boldsymbol{\phi} \leftarrow$ Initialize parameters
repeat
  $\mathbf{X}^M \leftarrow$ Random minibatch of M datapoints (drawn from full dataset)
  $\boldsymbol{\epsilon} \leftarrow$ Random samples from noise distribution $p(\boldsymbol{\epsilon})$
  $\mathbf{g} \leftarrow \nabla_{\boldsymbol{\theta}, \boldsymbol{\phi}} \widetilde{\mathcal{L}}^M(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{X}^M, \boldsymbol{\epsilon})$ (Gradients of minibatch estimator (8))
  $\boldsymbol{\theta}, \boldsymbol{\phi} \leftarrow$ Update parameters using gradients $\mathbf{g}$ (e.g. SGD or Adagrad [DHS10])
until convergence of parameters $(\boldsymbol{\theta}, \boldsymbol{\phi})$
return $\boldsymbol{\theta}, \boldsymbol{\phi}$

2.4 The reparameterization trick

- 평균을 구하기 위한 sampling 시 $q_\phi(z|x)$에서 z를 random하게 뽑으면 backpropagation이 안되는 문제 발생
- Noise variable $\epsilon \sim p(\epsilon)$를 뽑아 $\tilde{z} = g_\phi(\epsilon, x)$를 계산해 $\tilde{z}$가 마치 $q_\phi(z|x)$에서 샘플링 된 것 처럼 만든다.
- $z \sim q_\phi(z|x) = \mathrm{N}(\mu, \sigma^2)$일 때, $\tilde{z} = \mu + \sigma\epsilon, \epsilon \sim N(0,1^2)$을 계산해 $\tilde{z}$이 $q_\phi(z|x)$에서 샘플링 된 것 같은 효과를 준다.

2.4 The reparameterization trick

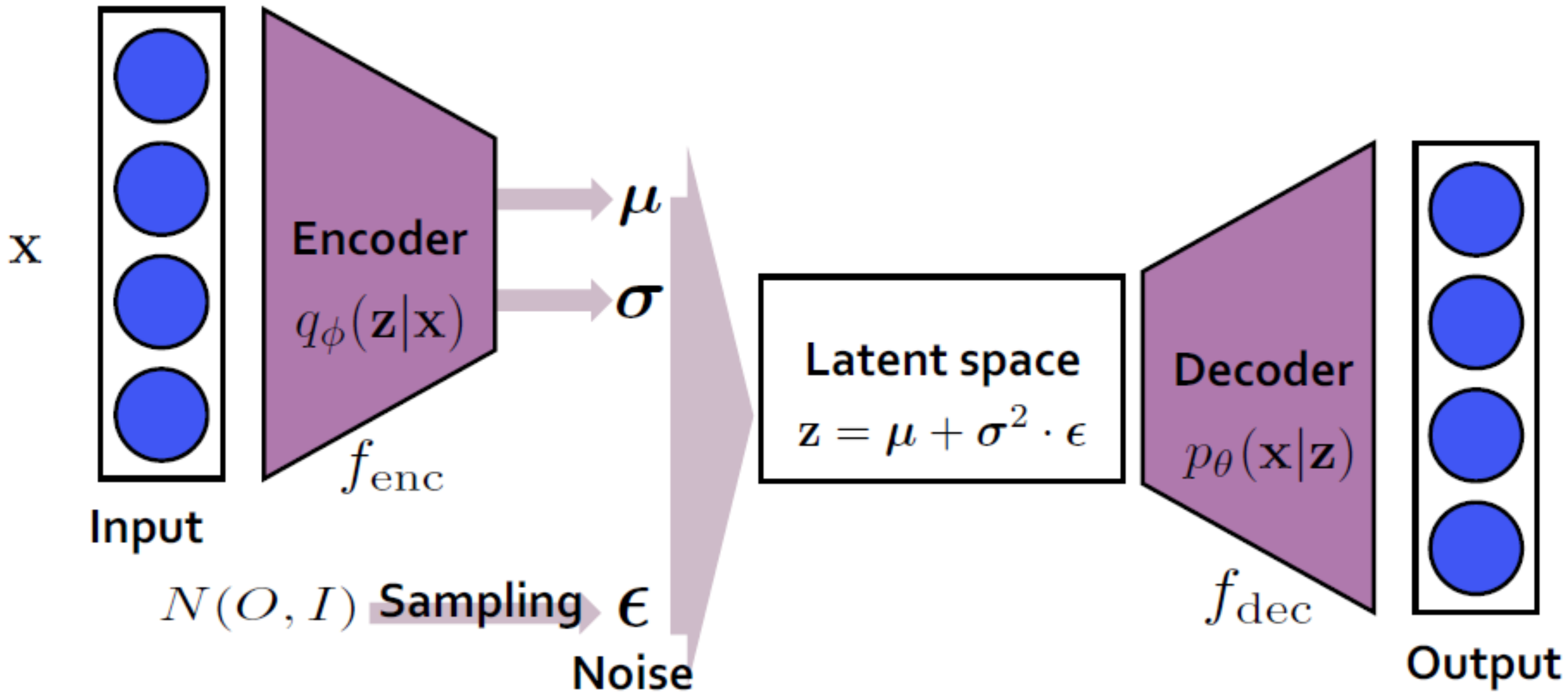


3. Example: Variational Auto-Encoder

- 뉴럴 네트워크에 적용해보기
- Assumption

1) $p_\theta(z) = N(z; 0, I)$
2) $p_\theta(x|z)$ is multivariate Gaussian(in case of real-valued data) or Bernoulli(in case of binary data)
3) $q_\phi\left(z \middle| x^{(i)}\right) = N(z;\ \mu^{(i)}, \sigma^{2(i)} I)$

3. Example: Variational Auto-Encoder



Reference

- https://arxiv.org/pdf/1312.6114.pdf
- https://hugrypiggykim.com/2018/09/07/variational-autoencoder와-elboevidence-lower-bound/
- https://jaejunyoo.blogspot.com/2017/05/auto-encoding-variational-bayes-vae-3.html
- https://process-mining.tistory.com/161
- https://velog.io/@hong_journey/VAEVariational-AutoEncoder-구현하기

감사합니다!