Variational Autoencoders (VAE)

Namwoo Kang

Smart Design Lab
CCS Graduate School of Green Transportation
KAIST

Reference

❑ 강의 슬라이드 및 실습코드는 아래의 링크에서 받으실 수 있습니다

- http://www.smartdesignlab.org/dl_aischool_2021.html
- Contributors: 김성신, 유소영, 이성희, 김은지

❑ 강의 소스

- Andrew Ng의 ML Class (www.holehouse.org/mlclass/)
- Fei-Fei Li & Justin Johnson & Serena Yeung, CS231n: Convolutional Neural Networks for Visual Recognition, Stanford (http://cs231n.stanford.edu/)
- Stefano Ermon & Aditya Grover, CS 236: Deep Generative Models , Stanford (https://deepgenerativemodels.github.io/)
- 모두를 위한 딥러닝 (https://hunkim.github.io/ml/)
- 모두를 위한 딥러닝 시즌 2 (https://deeplearningzerotoall.github.io/season2/lec_tensorflow.html)
- 이활석, Autoencoders (https://www.slideshare.net/NaverEngineering/ss-96581209)
- 최윤제, 1시간만에 GAN(Generative Adversarial Network) 완전 정복하기 (https://www.slideshare.net/NaverEngineering/1-gangenerative-adversarial-network?qid=c53ce33f-6643-4437-8e93-88776c9cebb1&v=&b=&from_search=5)
- 김성범, [핵심 머신러닝] Principal Component Analysis (PCA, 주성분 분석) (https://youtu.be/FhQm2Tc8Kic)

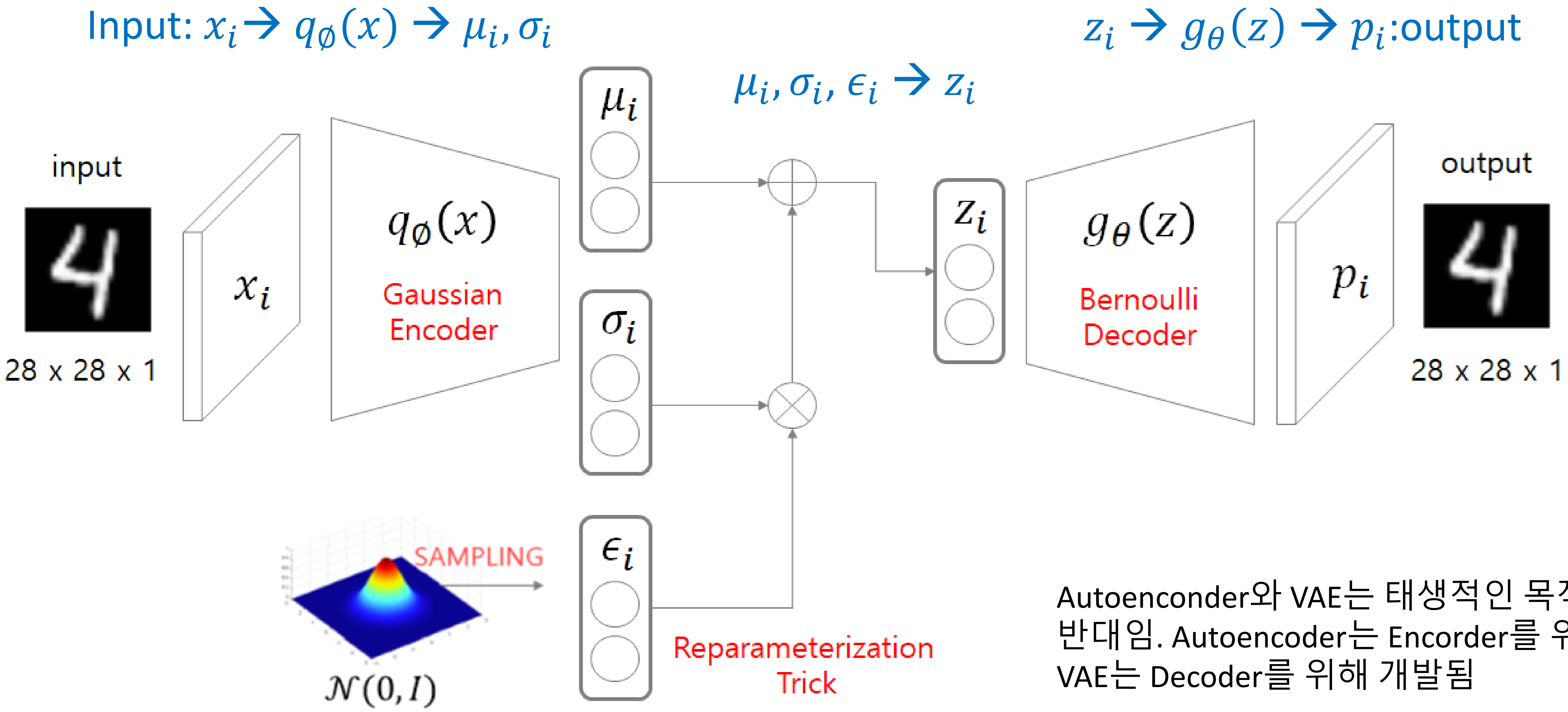
Contents

- **Ch1: Introduction to Unsupervised Learning Part I** → Probability & Maximum Likelihood
- **Ch2: Introduction to Unsupervised Learning Part II** → Generative Model & Dimensionality Reduction
- **Ch3: Principal Component Analysis (PCA)** → Machine Learning Model
- **Ch4: Autoencoder & Anomaly Detection** + 실습
- **Ch5: Variational AutoEncoder (VAE)** + 실습
- **Ch6: Generative Adversarial Network (GAN)** + 실습

→ Deep Learning Models

- **Ch7: Application: Mechanical Design + AI** → CAD/CAM/CAE/Design Optimization + AI

Variational Autoencoders (VAE) – How to work



Autoenconder와 VAE는 태생적인 목적이 반대임. Autoencoder는 Encorder를 위해 VAE는 Decoder를 위해 개발됨

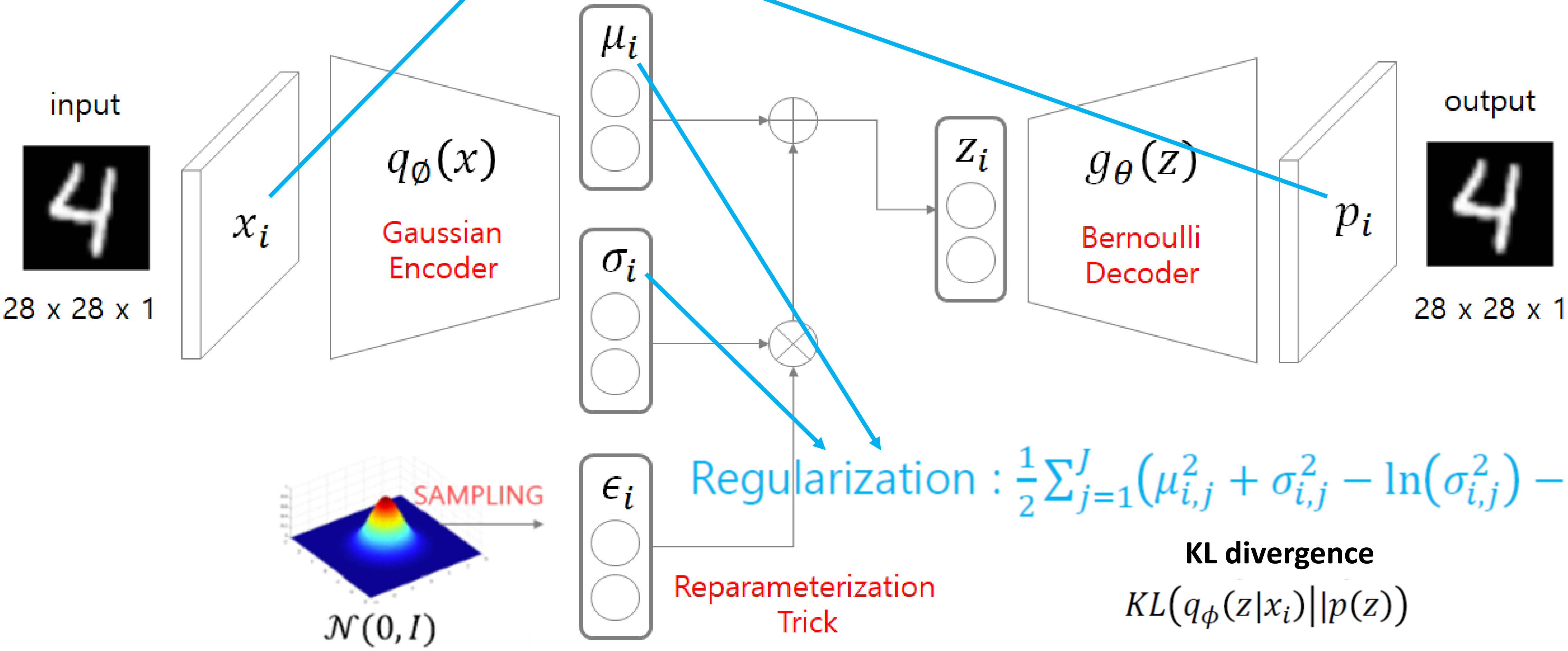
https://taeu.github.io/paper/deeplearning-paper-vae/

VAE – How to work

$$-\mathbb{E}_{q_\phi(z|x_i)}\left[\log\left(p(x_i|g_\theta(z))\right)\right]$$

Cross entropy

Reconstruction Error: $-\sum_{j=1}^{D} x_{i,j}\log p_{i,j} + (1-x_{i,j})\log\left(1-p_{i,j}\right)$



Regularization : $\frac{1}{2}\sum_{j=1}^{J}\left(\mu_{i,j}^2 + \sigma_{i,j}^2 - \ln\left(\sigma_{i,j}^2\right) - 1\right)$

KL divergence

$$KL\left(q_\phi(z|x_i)\,||\,p(z)\right)$$

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

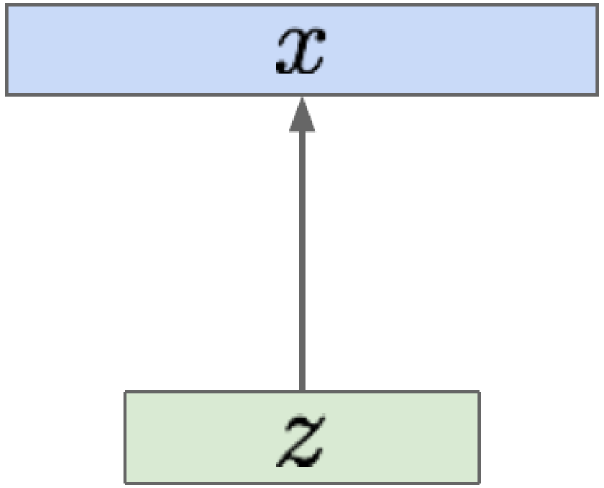
VAE - Loss Function

Probabilistic spin on autoencoders - will let us sample from the model to generate data!

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

Sample from true conditional $p_{\theta^*}(x \mid z^{(i)})$

Sample from true prior $p_{\theta^*}(z)$



Intuition (remember from autoencoders!): **x** is an image, **z** is latent factors used to generate **x:** attributes, orientation, etc.

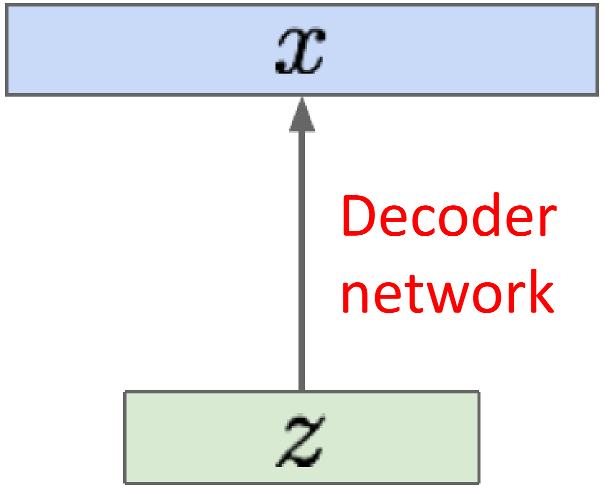
VAE - Loss Function

We want to estimate the true parameters θ^* of this generative model.

How should we represent this model?

Sample from true conditional

$p_{\theta^*}(x \mid z^{(i)})$



Sample from true prior

$p_{\theta^*}(z)$

Choose prior $p(z)$ to be simple, e.g. Gaussian. Reasonable for latent attributes, e.g. pose, how much smile.

Conditional $p(x|z)$ is complex (generates image) => represent with neural network

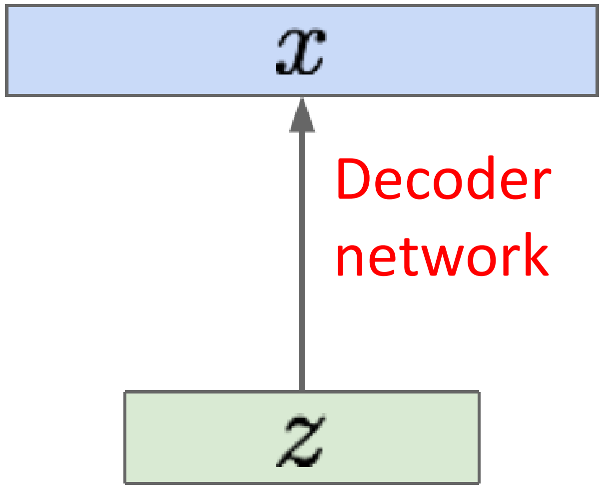
VAE - Loss Function

We want to estimate the true parameters θ^* of this generative model.

How to train the model?

Sample from true conditional

$p_{\theta^*}(x \mid z^{(i)})$



Sample from true prior

$p_{\theta^*}(z)$

Learn model parameters to maximize likelihood of training data

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

Now with latent z

Q: What is the problem with this?

Intractable!

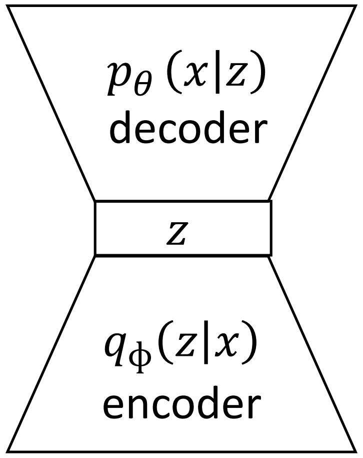
VAE - Loss Function

Intractible to compute $p(x|z)$ for every z!

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

Simple Gaussian prior

Decoder neural network



Posterior density also intractable: $p_\theta(z|x) = p_\theta(x|z) p_\theta(z) / p_\theta(x)$

Intractable data likelihood

Solution: In addition to decoder network modeling $p_\theta(x|z)$, define additional encoder network $q_\phi(z|x)$ that approximates $p_\theta(z|x)$

Will see that this allows us to derive a lower bound on the data likelihood that is tractable, which we can optimize

VAE - Loss Function

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

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

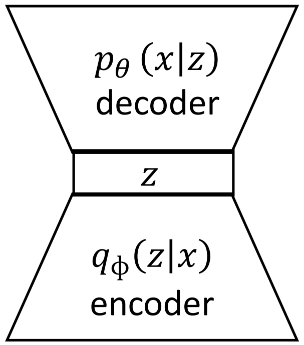
Taking expectation wrt. z (using encoder network) will come in handy later

$$= \mathbf{E}_z\left[log\frac{p_\theta\left(x^{(i)}|z\right)p_\theta(z)}{p_\theta\left(z|x^{(i)}\right)}\right]$$

(Bayes' Rule)

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)}$$

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



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

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

참고: $E_{z\sim \boxed{q_\phi(z|x^{(i)})}}\left[\boxed{log\frac{q_\phi(z|x^{(i)})}{p_\theta(z)}}\right] = \int_z \boxed{log\frac{q_\phi(z|x^{(i)})}{p_\theta(z)}}\boxed{q_\phi(z|x^{(i)})}dz$

$$KL(P||Q) = \sum_x P(x)\log\frac{P(x)}{Q(x)}$$

The expectation wrt. z (using encoder network) let us write nice KL terms

VAE - Loss Function

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

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

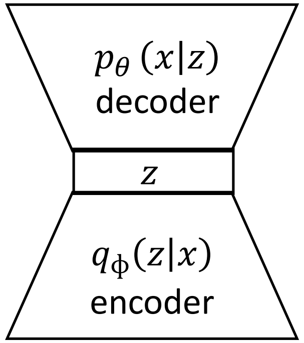
We want to maximize the data likelihood

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

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

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

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



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.

VAE - Loss Function

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

We want to maximize the data likelihood

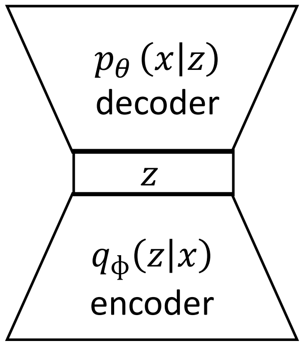
$$\log p_\theta(x^{(i)}) = \mathbf{E}_{z\sim q_\phi(z|x^{(i)})}\left[log p_\theta(x^{(i)})\right] \qquad (p_\theta(x^{(i)})\ Does\ not\ depend\ on\ z)$$

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

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

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

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



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

SDL SMART DESIGN LAB KAIST

VAE - Loss Function

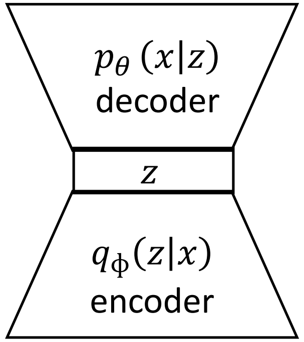
Now equipped with our encoder and decoder networks, let’s work out the (log) data likelihood:

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

We want to maximize the data likelihood

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

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



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

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

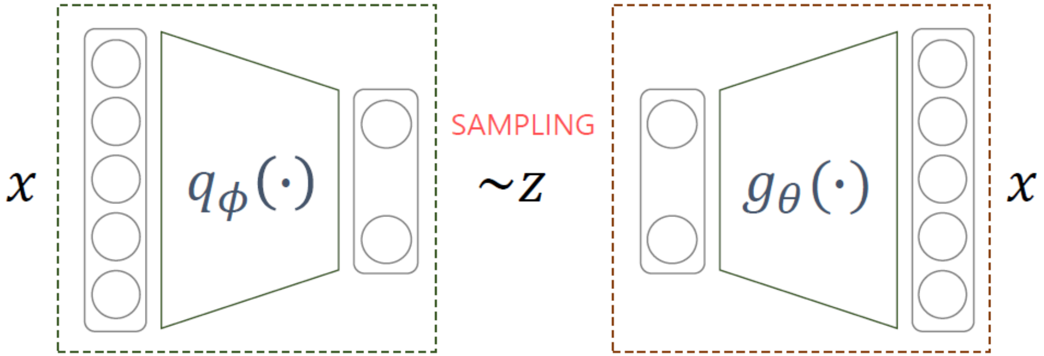
$$log\ p_\theta(x^{(i)}) \geq \mathcal{L}(x^{(i)},\theta,\phi)$$

Variational lower bound (“ELBO”)

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

Training: Maximize lower bound

VAE - Loss Function



$$\arg\min_{\theta,\phi} \sum_i -\mathbb{E}_{q_\phi(z|x_i)}\left[log\left(p(x_i|g_\theta(z))\right)\right] + KL(q_\phi(z|x_i)||p(z))$$

Reconstruction Error

- 현재 샘플링용 함수에 대한 negative log likelihood
- x_i에 대한 복원 오차 (Autoencoder 관점)

Regularization

- 현재 샘플링용 함수에 대한 추가 조건
- 샘플링의 용의성/생성 데이터에 대한 통제성을 위한 조건을 prior에 부여하고 이와 유사해야 한다는 조건을 부여

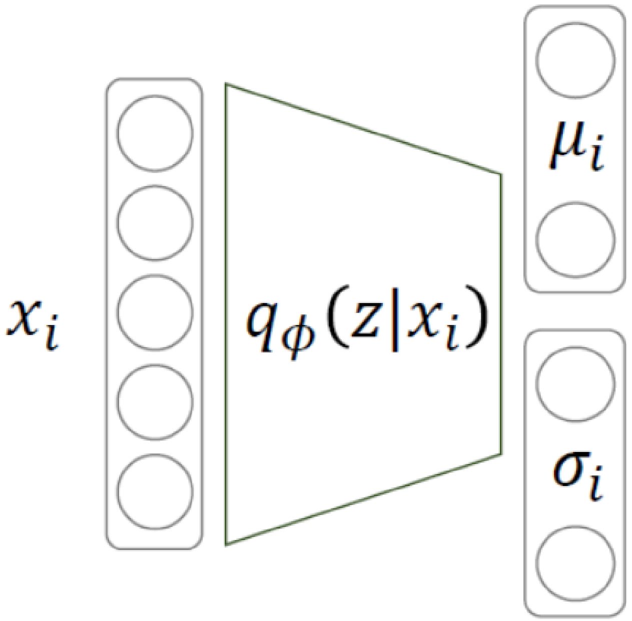
참고: $p(x|g_\theta(z)) = p_\theta(x|z)$

VAE - Optimization

Assumptions

$$\arg\min_{\theta,\phi} \sum_i -\mathbb{E}_{q_\phi(z|x_i)}\left[log\left(p(x_i|g_\theta(z))\right)\right] + KL(q_\phi(z|x_i)||p(z))$$

Regularization



Assumption 1

[Encoder : approximation class]

multivariate gaussian distribution with a diagonal covariance

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

Assumption 2

[prior] multivariate normal distribution

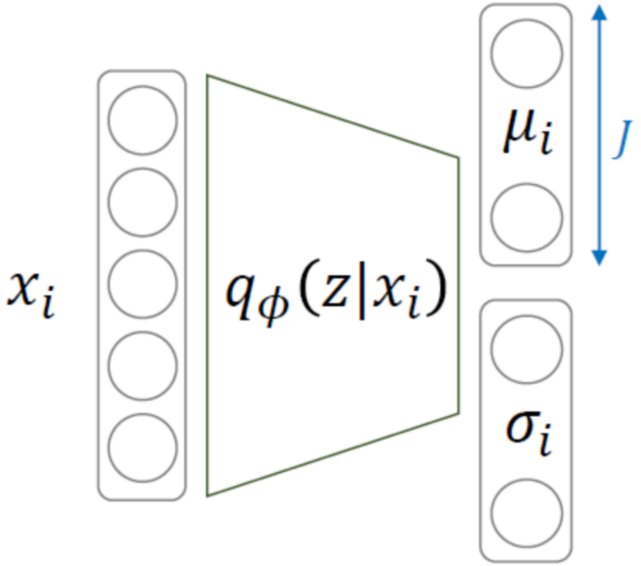
$$p(z) \sim N(0, I)$$

VAE - Optimization

KLD

$$\arg\min_{\theta,\phi} \sum_i -\mathbb{E}_{q_\phi(z|x_i)}\left[log\left(p(x_i|g_\theta(z))\right)\right] + KL(q_\phi(z|x_i)||p(z))$$

Regularization



$$KL(q_\phi(z|x_i)||p(z)) = \frac{1}{2}\left\{tr\left(\sigma_i^2 I\right) + \mu_i^T\mu_i - J + ln\frac{1}{\prod_{j=1}^{J}\sigma_{i,j}^2}\right\}$$

$$= \frac{1}{2}\left\{\sum_{j=1}^{J}\sigma_{i,j}^2 + \sum_{j=1}^{J}\mu_{i,j}^2 - J - \sum_{j=1}^{J} ln(\sigma_{i,j}^2)\right\}$$

$$= \frac{1}{2}\sum_{j=1}^{J}(\mu_{i,j}^2 + \sigma_{i,j}^2 - ln(\sigma_{i,j}^2) - 1)$$

KLD for multivariate normal distributions

$$D_{\mathrm{KL}}(\mathcal{N}_0 \parallel \mathcal{N}_1) = \frac{1}{2}\left(\mathrm{tr}\left(\Sigma_1^{-1}\Sigma_0\right) + (\mu_1 - \mu_0)^{\mathsf{T}}\Sigma_1^{-1}(\mu_1 - \mu_0) - k + \ln\left(\frac{\det\Sigma_1}{\det\Sigma_0}\right)\right)$$

VAE - Optimization

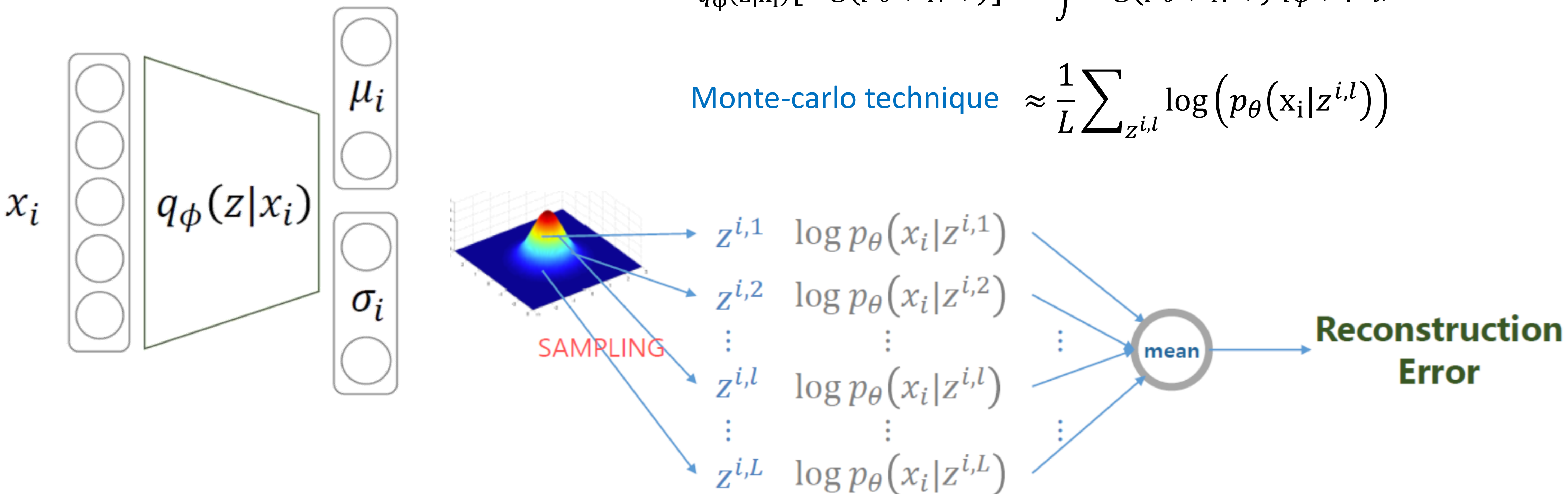
Sampling

$$\arg\min_{\theta,\phi} \sum_i -\mathbb{E}_{q_\phi(z|x_i)}\left[log\left(p(x_i|g_\theta(z))\right)\right] + KL(q_\phi(z|x_i)||p(z))$$

Reconstruction Error

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

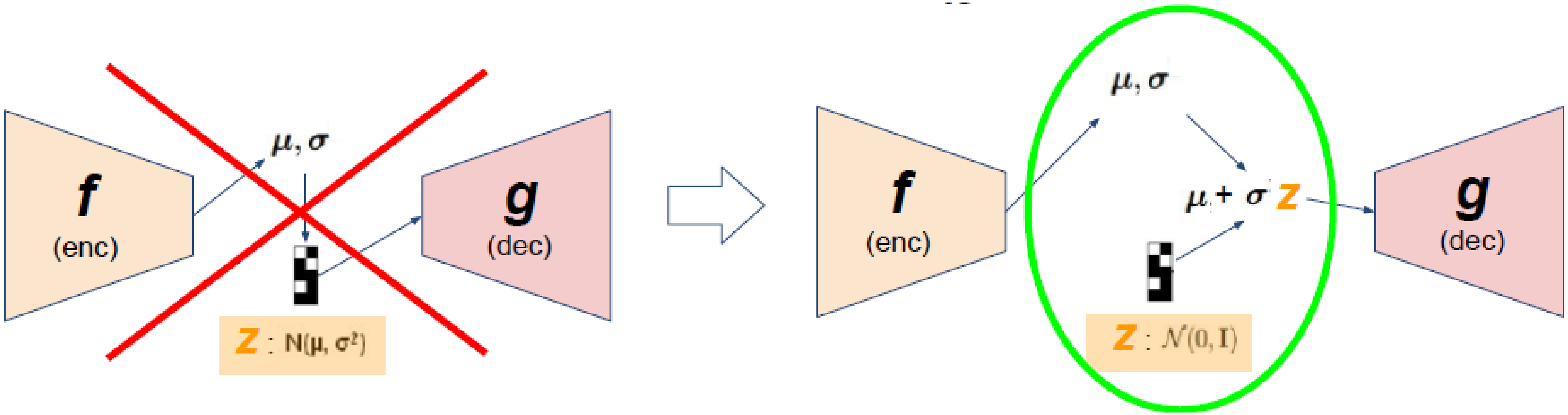
Monte-carlo technique $\approx \frac{1}{L}\sum_{z^{i,l}} \log\left(p_\theta\left(\mathrm{x_i}|z^{i,l}\right)\right)$



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

VAE - Optimization

Reparameterization Trick



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

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

Same distribution!
But it makes backpropagation possible!

VAE - Optimization

Assumptions

$$\arg\min_{\theta,\phi} \sum_i -\mathbb{E}_{q_\phi(z|x_i)}\left[log\left(p(x_i|g_\theta(z))\right)\right] + KL(q_\phi(z|x_i)||p(z))$$

Reconstruction Error

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

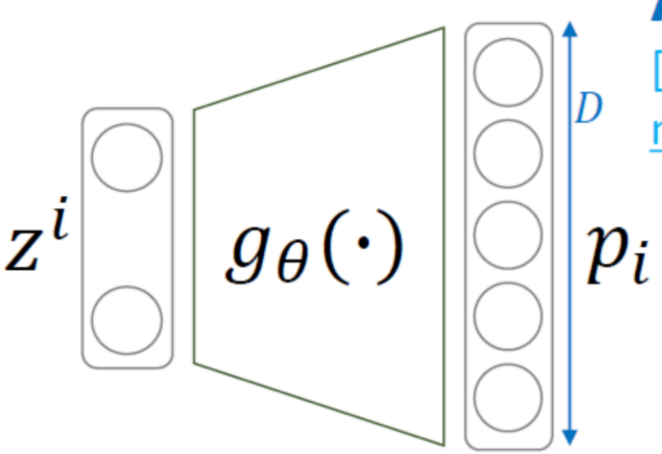
Monte-carlo technique

L=1

Assumption 3-1

[Decoder, likelihood]

multivariate bernoulli or gaussain distribution



$p_\theta\left(x_i|z^i\right) \sim Bernoulli(p_i)$

$$log\left(p_\theta\left(x_i|z^i\right)\right) = log\prod_{j=1}^{D} p_\theta\left(x_{i,j}|z^i\right) = \sum_{j=1}^{D} log\, p_\theta\left(x_{i,j}|z^i\right)$$

$$= \sum_{j=1}^{D} log\, p_{i,j}^{x_{i,j}}\left(1 - p_{i,j}\right)^{1-x_{i,j}}$$

$\longleftarrow$ $p_{i,j}$: network output

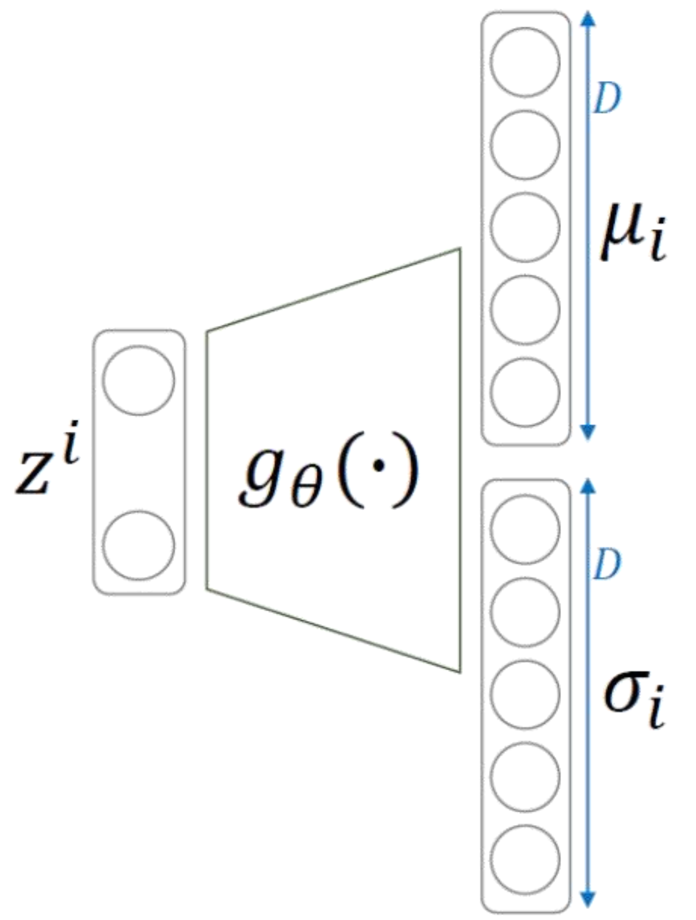
$$= \sum_{j=1}^{D} x_{i,j} log\, p_{i,j} + \left(1 - x_{i,j}\right)\log\left(1 - p_{i,j}\right)$$

Cross entropy

Assumptions

$$\arg\min_{\theta,\phi} \sum_i -\mathbb{E}_{q_\phi(z|x_i)}\left[log\left(p(x_i|g_\theta(z))\right)\right] + KL(q_\phi(z|x_i)||p(z))$$

Reconstruction Error



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

Assumption 3-2

[Decoder, likelihood]

multivariate bernoulli or gaussain distribution

$$\log\left(p_\theta\left(x_i|z^i\right)\right) = \log\left(N(x_i;\mu_i,\sigma_i^2 I)\right)$$

$$= -\sum_{j=1}^{D} \frac{1}{2}\log\left(\sigma_{i,j}^2\right) + \frac{(x_{i,j}-\mu_{i,j})^2}{2\sigma_{i,j}^2}$$

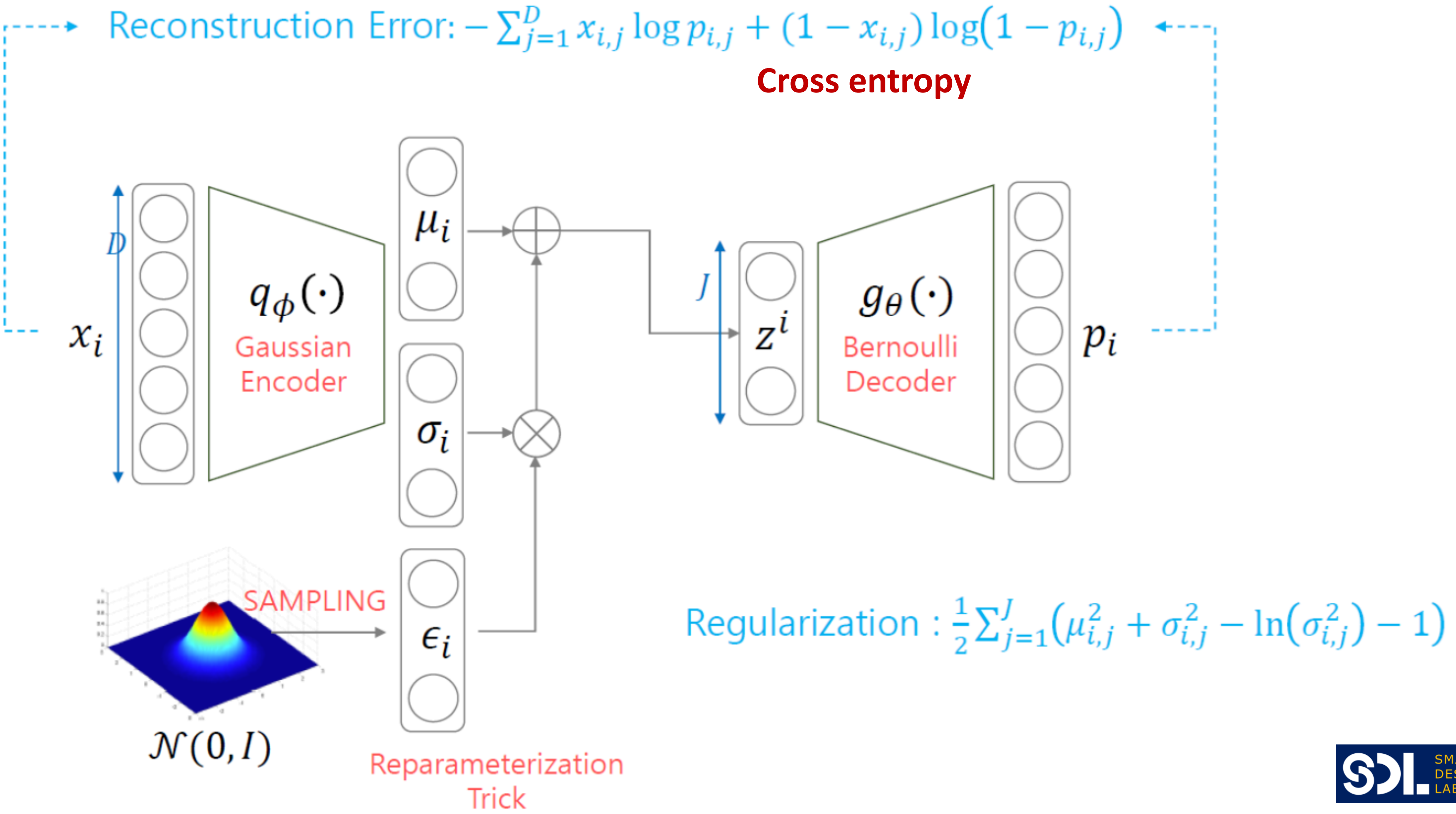
For gaussain distribution with identity covariance

$$\log\left(p_\theta\left(x_i|z^i\right)\right) \propto -\sum_{j=1}^{D} (x_{i,j}-\mu_{i,j})^2$$

Squared Error

VAE - Structure

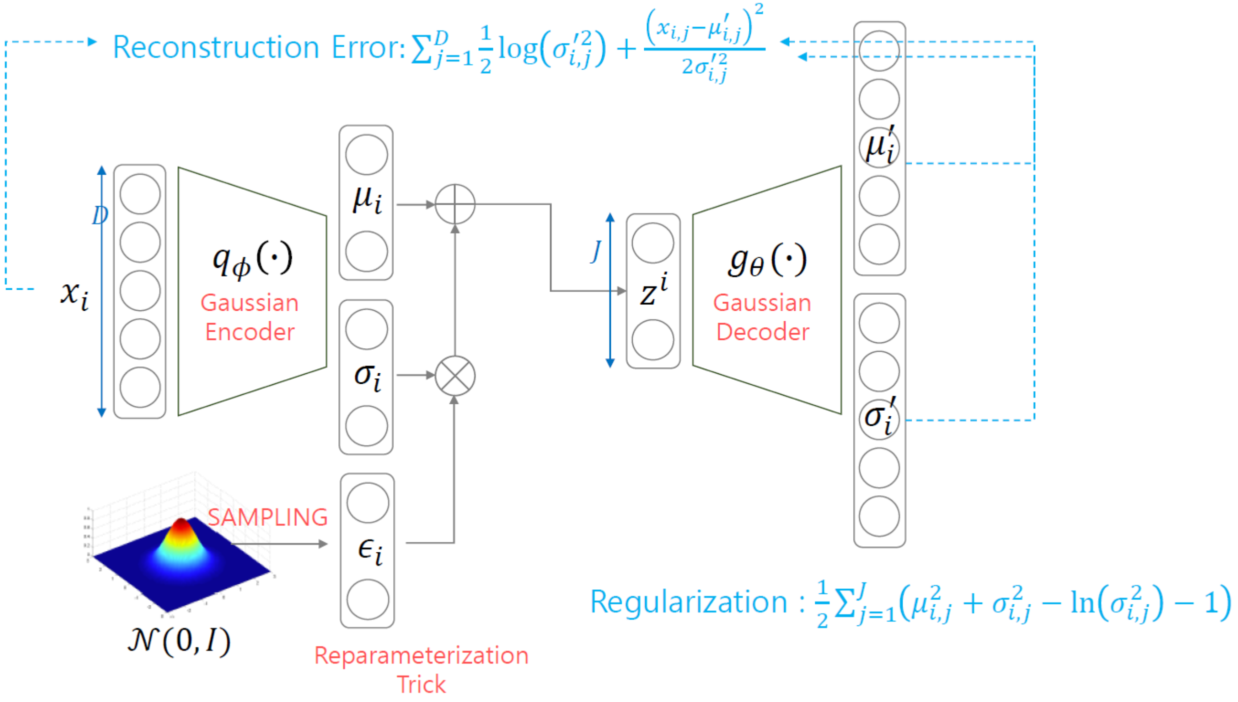
Default : Gaussian Encoder + Bernoulli Decoder



VAE - Structure

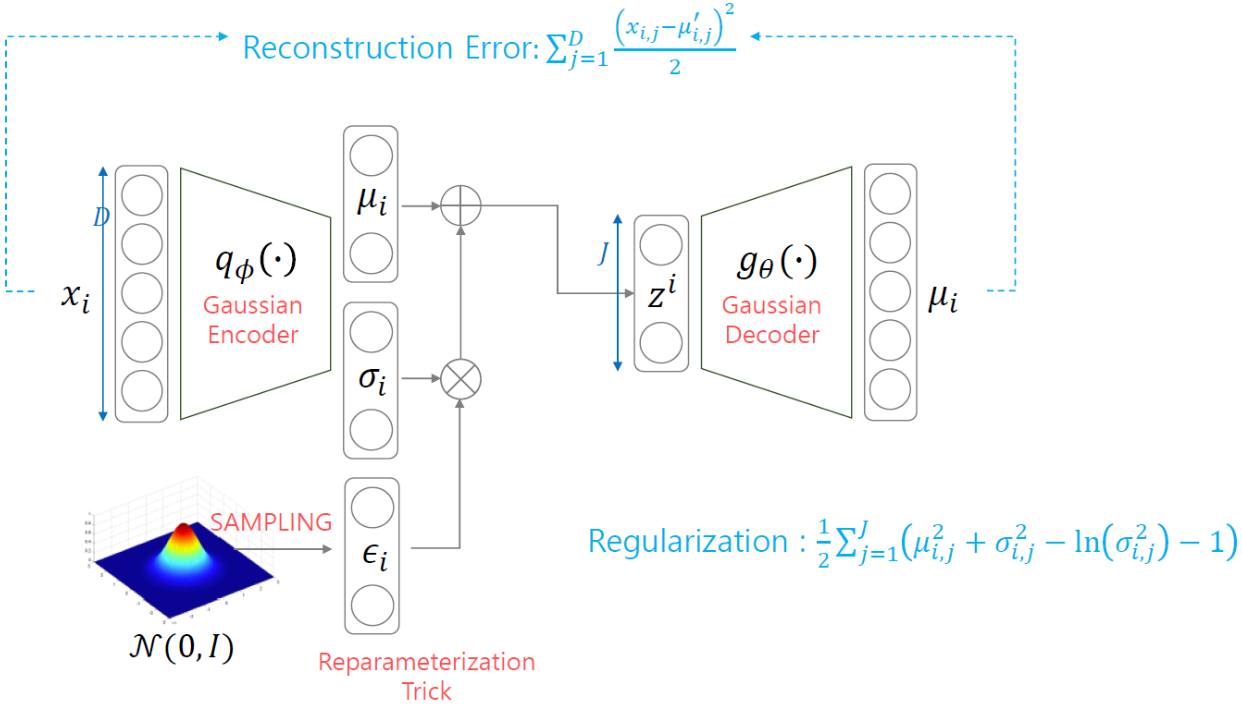
Gaussian Encoder + Gaussian Decoder

$$\sum_{j=1}^{D} \frac{1}{2}\log(\sigma_{i,j}^2) + \frac{(x_{i,j}-\mu_{i,j})^2}{2\sigma_{i,j}^2}$$



Gaussian Encoder + Gaussian Decoder with Identity Covariance

$$\sum_{j=1}^{D} (x_{i,j}-\mu_{i,j})^2$$ **Squared Error**



SMART DESIGN LAB KAIST

VAE – Characteristics

Latent variable dimensions

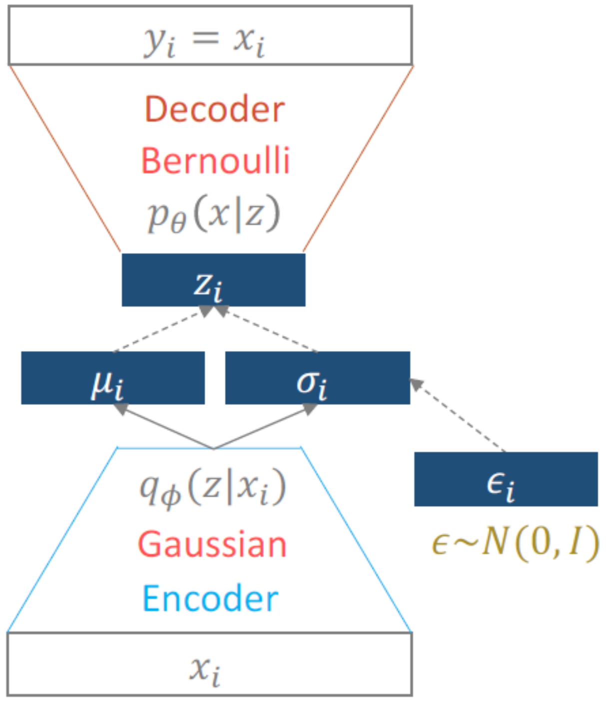
Input image

|z| =2

|z| =5

|z| =20

VAE – Characteristics



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

복원 오차 — *입력과 출력 간의 cross-entropy*

제약 조건 — *Prior 분포와의 다른 정도*

- Probabilistic spin to traditional autoencoders ➔ allows generating data
- Defines an intractable density ➔ derive and optimize a (variational) lower bound

[VAE의 특징들]

1. **Decoder**가 <u>최소한</u> 학습 데이터는 생성해 낼 수 있게 된다.
   → 생성된 데이터가 학습 데이터 좀 닮아 있다.

2. **Encoder**가 <u>최소한</u> 학습 데이터는 잘 latent vector로 표현할 수 있게 된다.
   → 데이터의 추상화를 위해 많이 사용된다.

VAE coding

$$L_i(\phi, \theta, x_i) = \underbrace{-\mathbb{E}_{q_\phi(z|x_i)}\left[\log\left(p\left(x_i|g_\theta(z)\right)\right)\right]}_{} \underbrace{+KL(q_\phi(z|x_i)||p(z))}_{} \Rightarrow \textit{argmax}\ \text{ELBO}(\phi)$$

Reconstruction Error
원데이터에 대한 Log Likelihood

Regularization
다루기 쉬운 확률 분포 중 선택해서 변이추론을 위한 근사 class중 선택하여 유사해야 한다는 조건을 부여함.

코딩에 적용된 수식

[*Regularization* : $\textit{Kullback} - \textit{leibler divergence}$]

$$KL(q_\phi(z|x_i)||p(z)) = \boxed{\frac{1}{2}\sum_{j=1}^{J} (\mu_{i,j}^2 + \sigma_{i,j}^2 - \ln(\sigma_{i,j}^2) - 1)}$$

[*Reconstruction Error*]

$$-\mathbb{E}_{q_\phi(z|x_i)}\left[\log\left(p\left(x_i|g_\theta(z)\right)\right)\right] = \int \log\left(p\left(x_i|g_\theta(z)\right)\right) \approx \underbrace{\frac{1}{L}\sum_{z^{i,l}} log(p_\theta(x_i|z^{i,l}))}_{} \approx \log(p_\theta(x_i|z^{i,l})) = \boxed{\sum_{j=1}^{D} x_{i,j} log p_{i,j} + (1 - x_{i,j}) log(1 - p_{i,j})}$$

Monte-carlo technique

For Bernoulli = cross-entropy

For Gaussian distibition
= mean square Error

What Questions Do You Have?

nwkang@kaist.ac.kr

www.smartdesignlab.org