ABOUT VAE

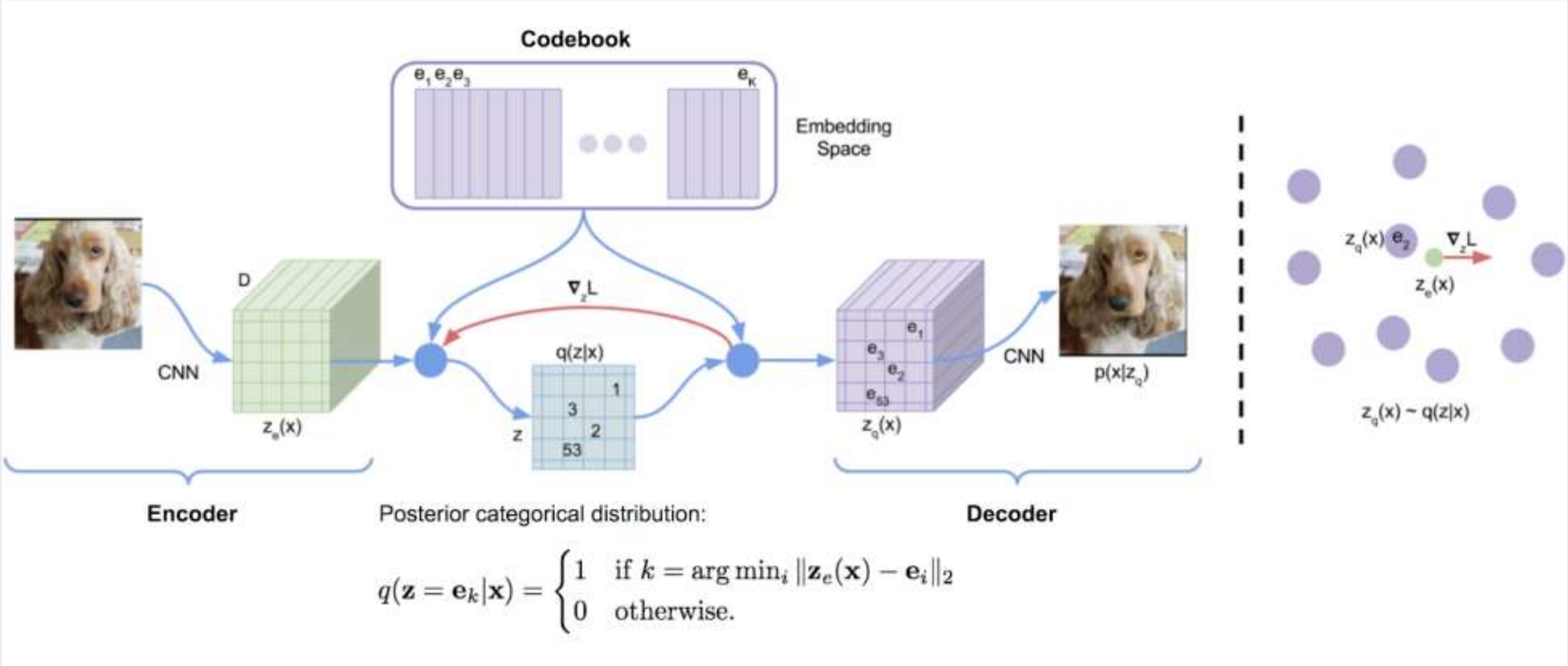
Sun-Hyuk, Choi

DALL-E 2

A photo of an astronaut riding a horse

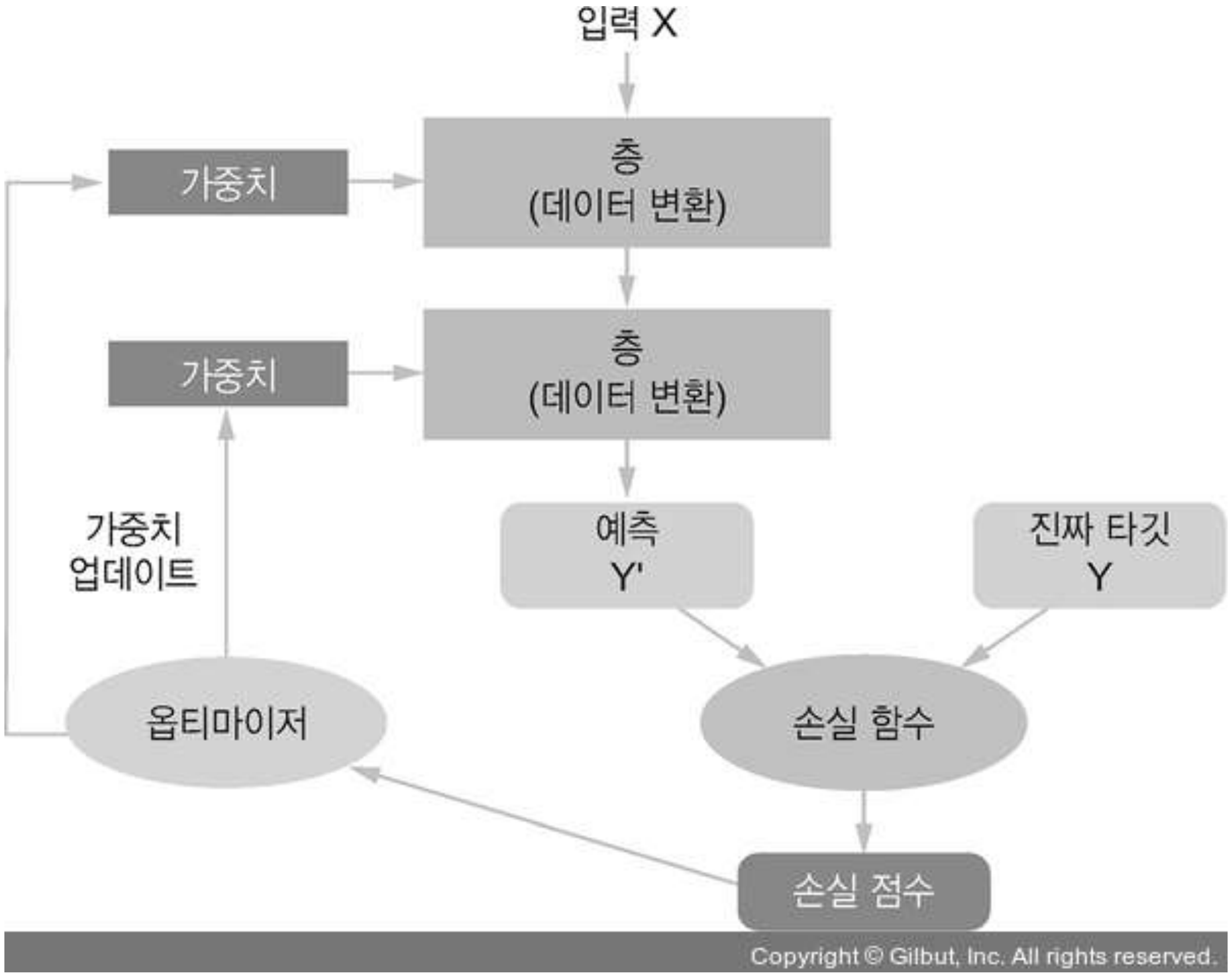
DALL-E

: Hot Text-to-Image Generative Model



DALL-E's Structure

How does Deep Learning work ?



Index

1	Auto-Encoder
2	Generative Model
3	Variational Inference
4	Object Function / Loss Function
5	Experiment / Result

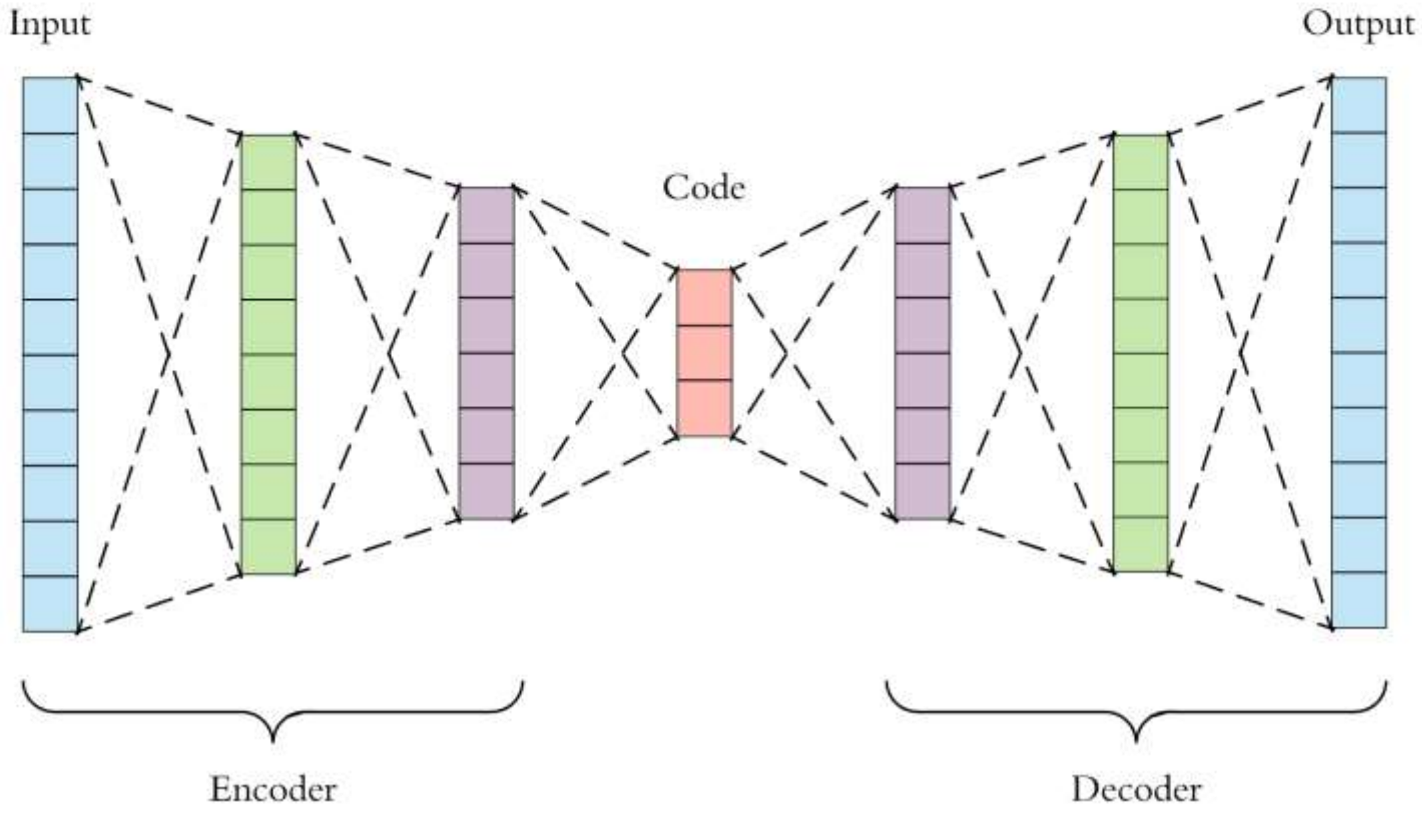
Part 1

Auto-Encoder

Part 1

Auto-Encoder

: Model used to compress/extract the information of sample



Point 1 : Structure

Point 2 : Encoder

Point 3 : Decoder

Point 4 : Object

Part 2

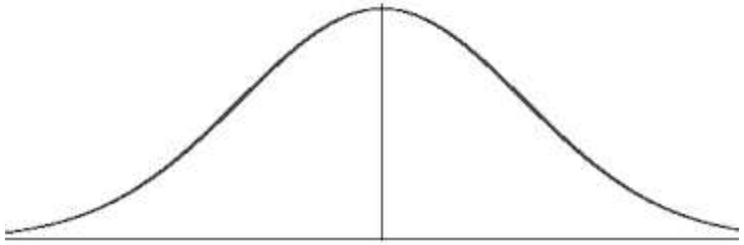
Generative Model

Part 2

Generative Model

: model that estimate the distribution of given sample

Generate data similar to the given sample = Sample data from the distribution of given sample = Estimate the distribution of given sample

Given data →  → New data

$N(\mu, \sigma)$

Part 2

Generative Model

Likelihood : the joint probability of the observed data viewed as a function of the parameters of the chosen statistical model

$$L(\theta|X) = L(\theta|x_1) \times L(\theta|x_2) \times L(\theta|x_3) \times L(\theta|x_4) \times \cdots L(\theta|x_n)$$
$$= p(x_1|\theta) \times p(x_1|\theta) \times p(x_1|\theta) \times p(x_1|\theta) \cdots p(x_1|\theta)$$

$$logL(\theta|X) = \sum_i logP(x_i|\theta)$$

Generative Model's object : Maximize the loglikelihood with given data.

Part 3

Variational Inference

Part 3

Variational Inference

Why don't we use maximum likelihood estimation directly?

$$logLikelihood, logP(X|\theta) = \sum_i logP(x_i|\theta) = \sum_i logP(x_i|z_i)P(z_i|\theta), \boxed{z\sim P(z|x)}$$

Problem 1. The generated distribution is hard to be treated numerically

Problem 2. The generated distribution don't describe the given data well.

So, we approximate the given data to the distribution we familiar with.

Variational Inference : $P(Z|X) \approx Q(Z), Q(Z): variational\ distribution$

Example) $z\sim N(\mu,\sigma)$, $x\sim Bernoulli(p)$

Part 4

Object Function

Object Function / Loss Function

VARIATIONAL INFERENCE | ELBO : Evidence LowerBOund

Relationship among $p(x), p(z|x), q_\phi(z|x)$: Derivation 2

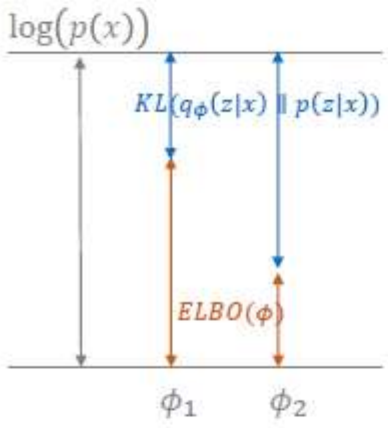
$$\log(p(x)) = \int \log(p(x))q_\phi(z|x)dz \qquad \leftarrow \int q_\phi(z|x)dz = 1$$

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

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

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

두 확률분포 간의 거리≥0



KL을 최소화하는 $q_\phi(z|x)$의 ϕ값을 찾으면 되는데 $p(z|x)$를 모르기 때문에,
KL최소화 대신에 ELBO를 최대화하는 ϕ값을 찾는다.

Object Function / Loss Function

1st term : can't be computed

$$\log p_\theta(x^{(i)}) = \boxed{D_{KL}(q_\varphi(z|x^{(i)})||p_\theta(z|x^{(i)}))} - D_{KL}(q_\varphi(z|x^{(i)}||p_\theta(z)) + E_{q_\varphi(z|x^i)}[logp_\theta\left(x^{(i)}\middle|z\right)]$$

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

$$logp_\theta\left(x^{(i)}\right) \geq L\left(\theta,\varphi;x^{(i)}\right) = E_{q_\varphi(z|x)}\left[-logq_\varphi(z|x) + logp_\theta(x,z)\right]$$

$$L(\theta,\varphi;x^{(i)}) = \boxed{-D_{KL}(q_\varphi(z|x^{(i)}||p_\theta(z))} + \boxed{E_{q_\varphi(z|x^i)}[logp_\theta\left(x^{(i)}\middle|z\right)]}$$

Regularization Term:
The distance Between
prior distribution
and posterior distribution

Reconstruction term :
How the generated distribution
describes the given data
(log likelihood)

Object Function / Loss Function

$$L(\theta, \varphi; x^{(i)}) = -D_{KL}(q_{\varphi}(z|x^{(i)}||p_{\theta}(z)) + E_{q_{\varphi}(z|x^{i})}[logp_{\theta}(x^{(i)}|z)]$$

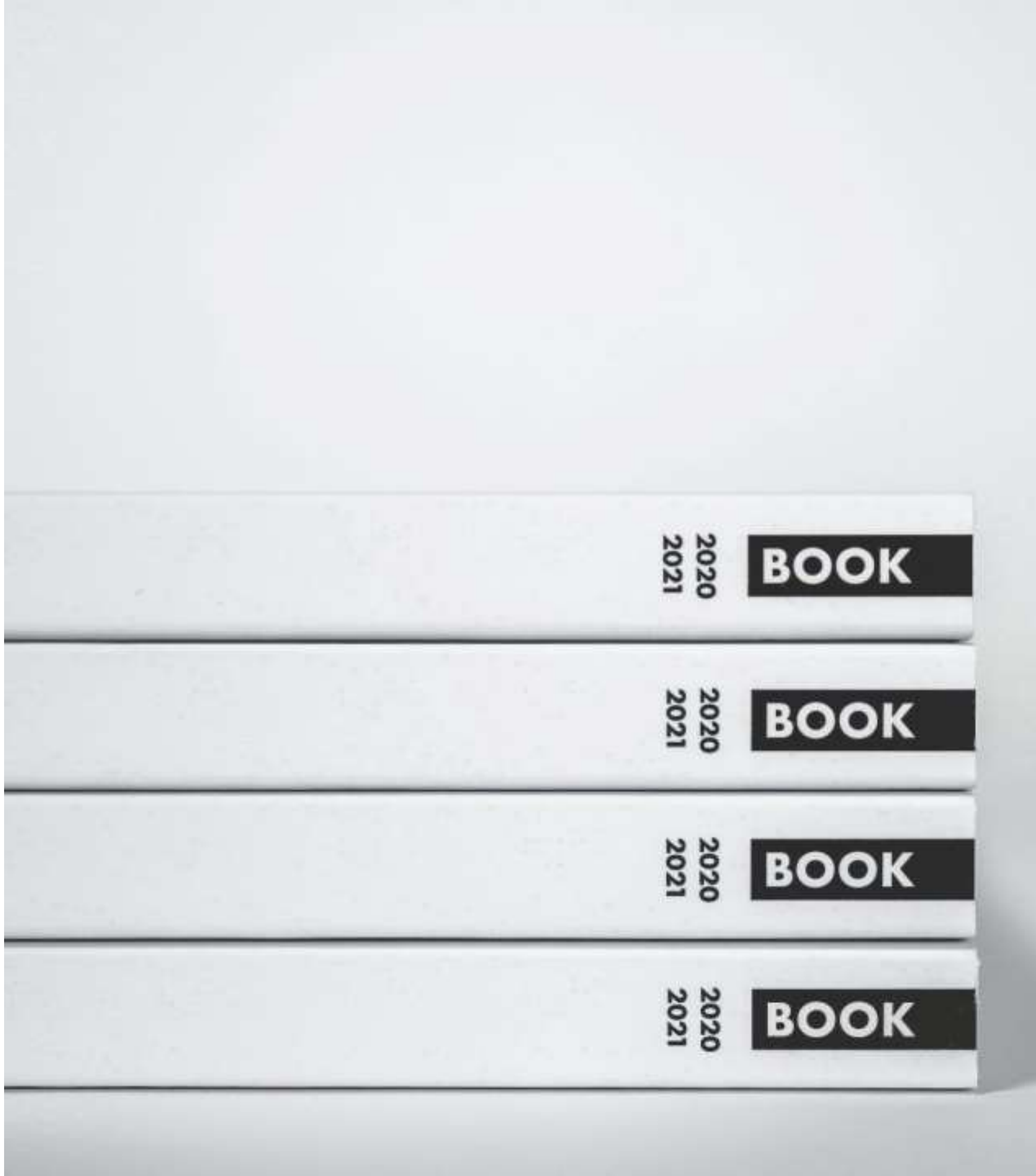
$$\boxed{argmin_{\theta,\varphi} \sum_{i} +D_{KL}(q_{\varphi}(z|x^{(i)}||p_{\theta}(z)) - E_{q_{\varphi}(z|x^{i})}[logp_{\theta}(x^{(i)}|z)]}$$

Point 1. Generate data as similar to the input data as possible.

Point 2. Make the prior distribution as close as possible to the posterior distribution of latent variable z.

Part 5

Experiment & Result



Part 5

Experiment & Result

Goal : Let make a face

Part 5

Experiment & Result

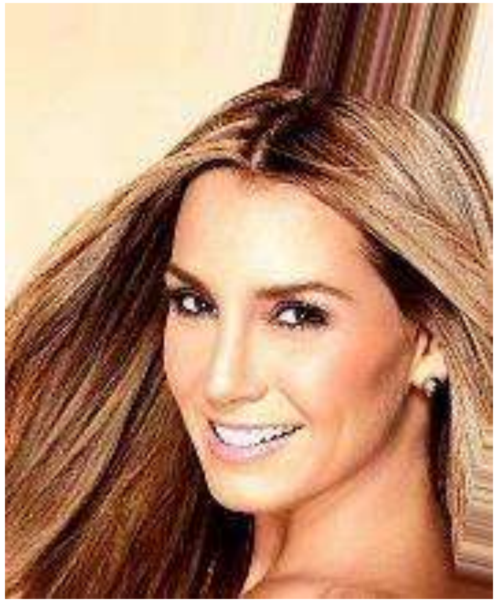
Celeba : Large face dataset with over 200,000 celebrity images each with 40 attribute annotations

Detail :

202,599 images
178x218 RGB
Scale : 0~255

Preprocessing :

Train images : 141,819 images
Test images : 60,780 images
Size = 64x64x3
Scale : 0~1



Experiment & Result

Assumption :

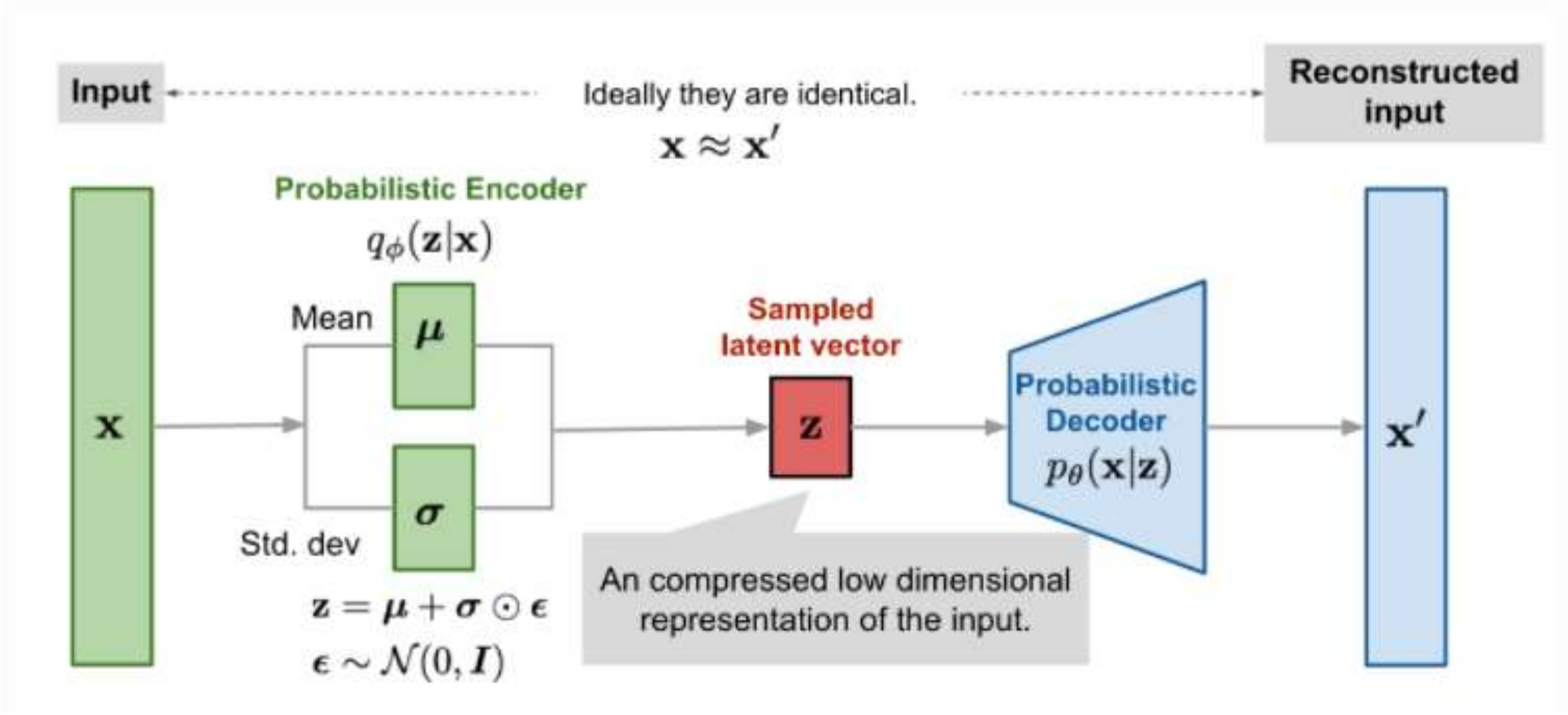
$$Image\ data\ x \sim N(\mu, \sigma),$$
$$Latent\ variable\ z \sim N(\mu', \sigma')$$

$$= \frac{1}{2}\sum_{j}^{J}(1 + \log((\sigma_j)^2) - (\mu_j)^2 - (\sigma_j)^2)$$
$$- D_{KL}((q\varphi(z)||p\theta(z))$$
$$= \int q\theta(z)\ (log\ p\theta(z)\ -\ log\ q\theta(z))\ dz$$

$$E_{q_\varphi(z|x^i)}[logp_\theta(x^{(i)}|z)] \cong \frac{1}{L}\sum_{l=1}^{L} logp_\theta(x^{(i)}|z^{i,l})$$
$$= \frac{1}{L}\sum_{l=1}^{L} log\frac{1}{\sqrt{2\pi}\sigma^{i,l}} - \frac{(y_i - \mu_{i,l})^2}{2\sigma^{i,l}}$$
$$\propto \frac{1}{L}\sum_{l=1}^{L} -\frac{(y_i - \mu_{i,l})^2}{2}$$

Part 5

Experiment & Result



$$Loss = -(x - \hat{x})^2 + \frac{1}{2}(1 + \log(\sigma^2) - \mu^2 - \sigma^2)$$

Experiment & Result

CODES

```python
class VAE(nn.Module):
    def __init__(self):
        super().__init__()
        self.encoder = encoder()
        self.decoder = decoder()
    def forward(self,x):
        [mu,sig,z] = self.encoder(x)
        x_approx = self.decoder(z)
        return x_approx,x,mu,sig

#if __name__ == "__main__":
from torchviz import make_dot
model = VAE().to(device)
x = torch.rand(1,3,64,64).to(device)
[x_approx,x,mu,sig] = model(x)
make_dot(x_approx.mean(), params=dict(model.named_parameters()))
```

```python
def train_loop(dataloader, model, optimizer,scaling):
    size = len(dataloader.dataset)
    for batch, X in enumerate(dataloader):
        X = X.to(device)
        # 예측 오류 계산
        [appro_x,x,mu,sigma] = model(X)
        reconst_error = mse(appro_x,X)
        regularization = torch.mean(-0.5*torch.sum(1+sigma-mu**2-
                                    sigma.exp(),dim=1),dim=0)
        loss = reconst_error + 0.00024*regularization
        # 역전파
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
        if batch % 100 == 0:
            psnr = _PSNR(x,appro_x,scaling)
            loss, current = loss.item(), batch * len(X)
            print(f"loss: {loss:>7f} psnr:{psnr:>7f} [{current:>5d}/{size:>5d}
```

Experiment & Result

Real Images

Predictions

Mean of ELBO : 0.5657

EXTRA Experiment – Bernoulli Dist

Real Images

Predictions

Mean of ELBO : 0.5364

Reference

Diederik P. Kingma, Max Welling. Auto-Encoding Variational Bayes, 2014

PyTorch-VAE/vanilla_vae.py at master · AntixK/PyTorch-VAE · GitHub

이활석(NAVER), 오토인코더의 모든 것 (slideshare.net), 2017

명사 | TensorFlow Datasets

VAE(Variational AutoEncoder) - gaussian37

[정리노트] AutoEncoder의 모든것 Chap1. Deep Neural Network의 학습 방법에 대해 알아보자(딥러닝 학습방법) (tistory.com)

프랑소와 숄레, 케라스 창시자에게 배우는 딥러닝, 2018

Thank you for watching