Variational Autoencoder

Instructor: Seunghoon Hong

Course logistics

- The next assignment will be released this week
- No lecture on the next Wednesday (11/15)

Recap: objective of deep generative models

- Learning a model that its outputs follow the true data distribution

$$G_\theta \sim P(X)$$

Generated images from G

True Images X

Recap: objective of deep generative models

- Maximum Likelihood Estimation (MLE)

$$\hat{\theta} = \arg\max_{\theta \in \Theta} \sum_{x_i \in \mathcal{X}} \log p_\theta(x_i)$$

$$\leftrightarrow \arg\min_{\theta \in \Theta} \sum_{x_i \in \mathcal{X}} -\log p_\theta(x_i)$$

Find model parameters that minimize the Negative Log-Likelihood (NLL) of data

Recap: autoregressive model

- Factorized objective function

$$\hat{\theta} = \arg\min_{\theta \in \Theta} \sum_{x_i \in \mathcal{X}} -\log p_\theta(x_i)$$

$$= \arg\min_{\theta \in \Theta} \sum_{x_i \in \mathcal{X}} \sum_{1 \leq t \leq d} -\log p_\theta(x_i^t | x_i^1, \ldots, x_i^{t-1})$$

Today's agenda

- Latent variable model
- Autoencoder
- Variational Autoencoder

Rethinking generative process of data

- So far, we have learned to understand various data

Image

“The agreement on the European Economic Area was signed in August 1992.”

Language (e.g. sentence)

Rethinking generative process of data

- To represent data, we treat it as a fixed-dimensional vector (or matrix)

$\boxed{x \in \mathbb{R}^d}$ super large dimension!

d = width x height x 3

Image

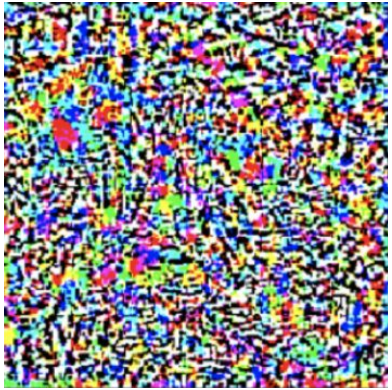
d = (sentence length) x (# of words)

"The agreement on the European Economic Area was signed in August 1992."

Language (e.g. sentence)

Rethinking generative process of data

- Q: Are all elements in an observation space valid?
  - If we sample random point in a data space,
    would there be a high chance that it is a valid data?
- A: certainly not (in almost all cases)!



Image

“Adlsladjf a;lekjfa qoweiruqewfr aaaa 123 bsd.”

Language (e.g. sentence)

Rethinking generative process of data

- Only a small portion of data in data space is valid
- It means that there could be much more compact representation of data!

Data $\quad\quad\quad\quad$ **Compact representation**

$$x \in \mathbb{R}^{\boxed{d}} \Rightarrow z \in \mathbb{R}^{\boxed{m}}$$

$$\boxed{d >> m}$$

Rethinking generative process of data

- Only a small portion of data in data space is valid
- It means that there could be much more compact representation of data!

Data		Compact representation	
$x \in \mathbb{R}^d$	$\Rightarrow$	$z \in \mathbb{R}^m$	$d >> m$
Observable variable we can directly observe this from data		**Latent variable** it is hidden; we cannot observe it it should be *inferred* from data	

How do we learn a representation?

- Supervised learning of representation
  - Given a set of paired data (x,y),
    learn parameters to associate x and y

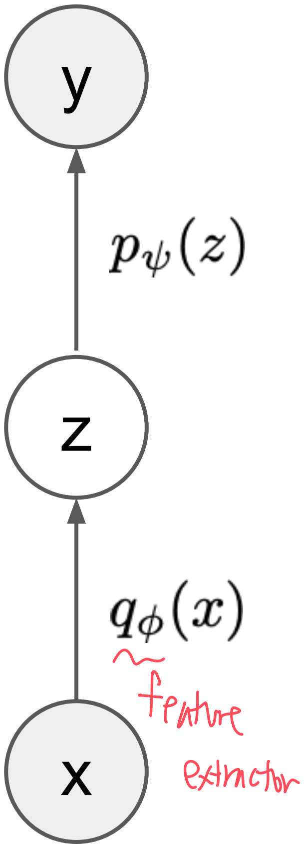
Output label이 있기 때문에 Loss function을 정의하여 feature extractor를 학습할 수 있다!!

supervised learning의 경우 [illegible]

- Learning objective

$$\arg\min_{\varphi,\phi} \sum_{(x,y)\in D} \mathcal{L}(p_\varphi(g_\phi(x)), y)$$

prediction for **x** known output **y**

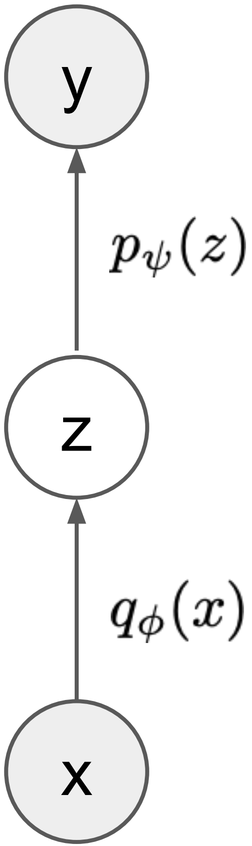


How do we learn a representation?

- Supervised learning of representation
  - Given a set of paired data (x,y),
    learn parameters to associate x and y

- Learning objective

$$\arg\min_{\varphi,\phi} \sum_{(x,y)\in D} \underbrace{\mathcal{L}(p_\varphi(g_\phi(x)), y)}_{\substack{\text{Comparison of the model output for x} \\ \text{with known output y}}}$$



How do we learn a representation?

- Unsupervised learning of representation
  - Given a set of data x,
    learn to model the probability distribution p(x)

- Learning objective (maximum likelihood estimation)

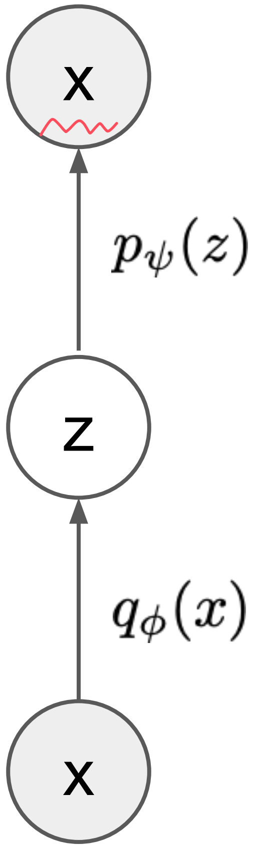
$$\arg\max_{\theta} \sum_{x \in D} \underbrace{p_\theta(x)}$$

Maximize the estimated probability
of observing data x

$P_\theta(x = \hat{x})$.

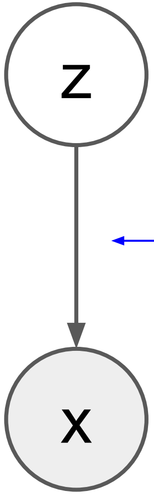
architecture 잘 고려해

$\|\hat{x} - x\|^2$를 minimize.



Latent variable model

- Introduce a latent variable to represent/generate a data



Latent variable:
compact representation of data

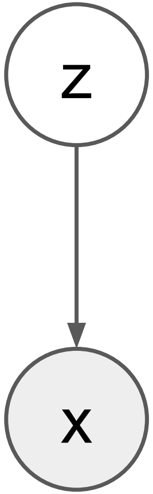
Generation of data depends on latent variable

Observation variable
original representation of data

Latent variable model

- Generative process in a latent variable model

$$p(x, z) = p(x|z)p(z)$$



1. Sample the latent variable

$$z \sim p(z)$$

Assumption : we know $p(z)$

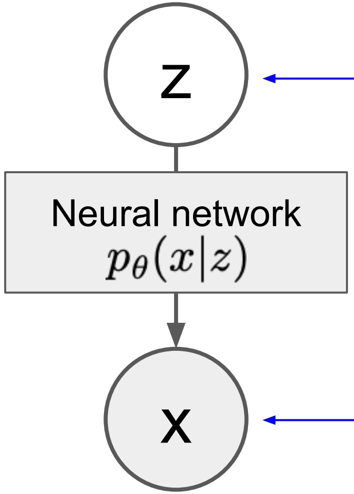
2. Generate data ***conditioned on*** the latent variable

$$x \sim p(x|z)$$

Latent variable model

- Generative process in a latent variable model

$$p(x, z) = p(x|z)p(z)$$



1. Sample the latent variable

$$z \sim p(z)$$

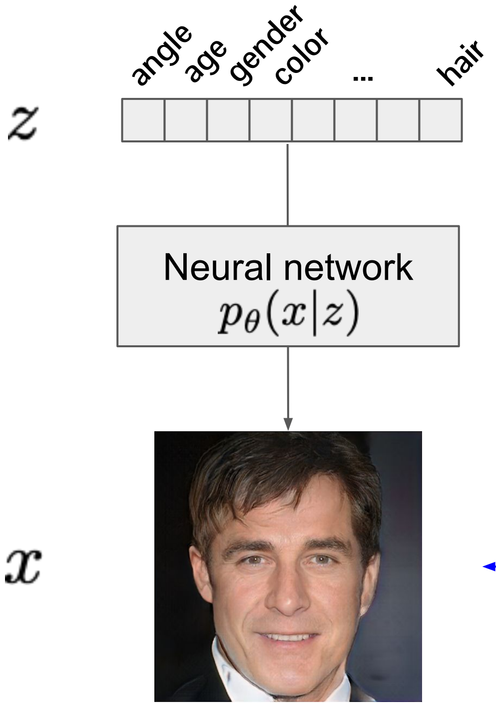
2. Generate data ***conditioned on*** the latent variable

$$x \sim p_{\boxed{\theta}}(x|z)$$

We use neural network to model this part

Example

- **Special case:** if latent variable is an attribute vector

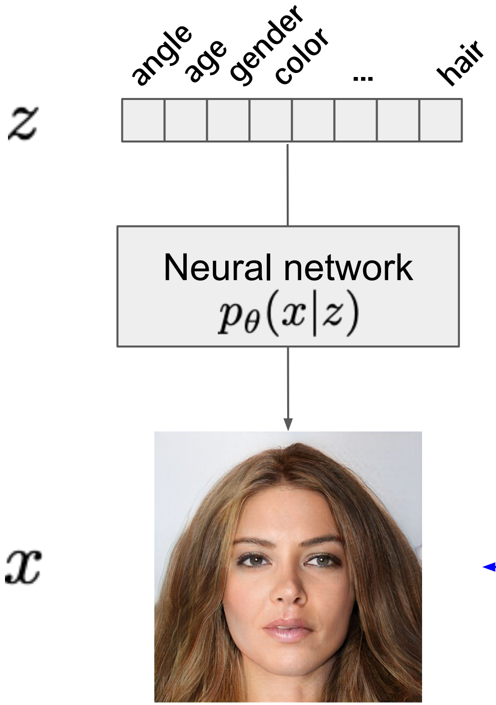


1. Sample some face attribute vector

2. Generate face image ***conditioned on*** the attribute

Example

- **Special case:** if latent variable is an attribute vector



1. Sample some face attribute vector

2. Generate face image ***conditioned on*** the attribute

Why do we care about latent variable?

① efficiency of constructing model.

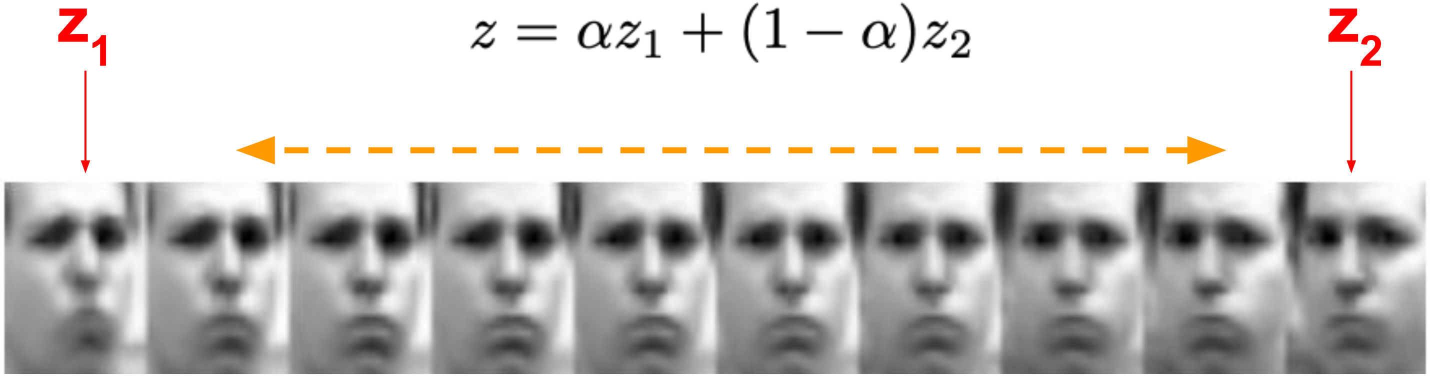
- ② It allows us to interpret and manipulate data much more easily ∵ compact representation.
- **Example:** latent interpolation

Why do we care about latent variable?

- It allows us to interpret and manipulate data much more easily
- **Example:** latent interpolation

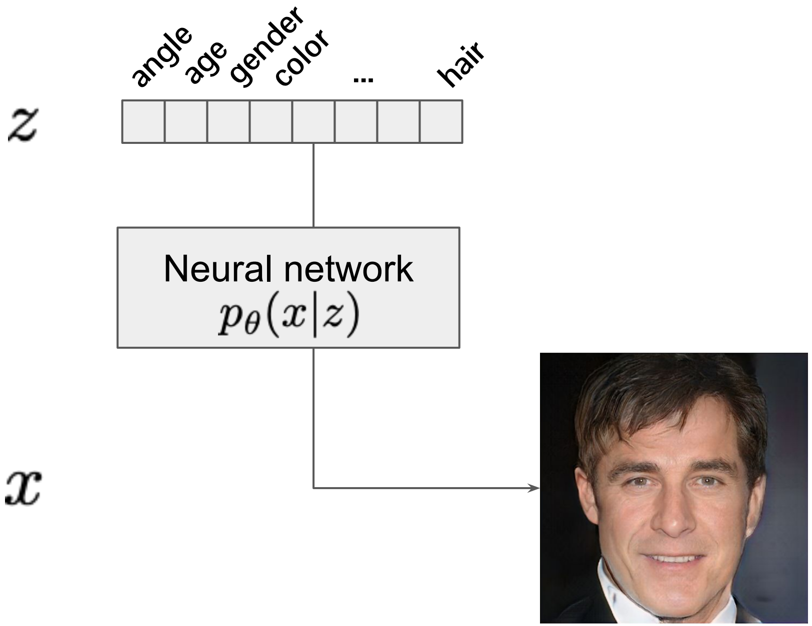
└ x (observed val)로 하면
very blurred

$$z = \alpha z_1 + (1 - \alpha) z_2$$



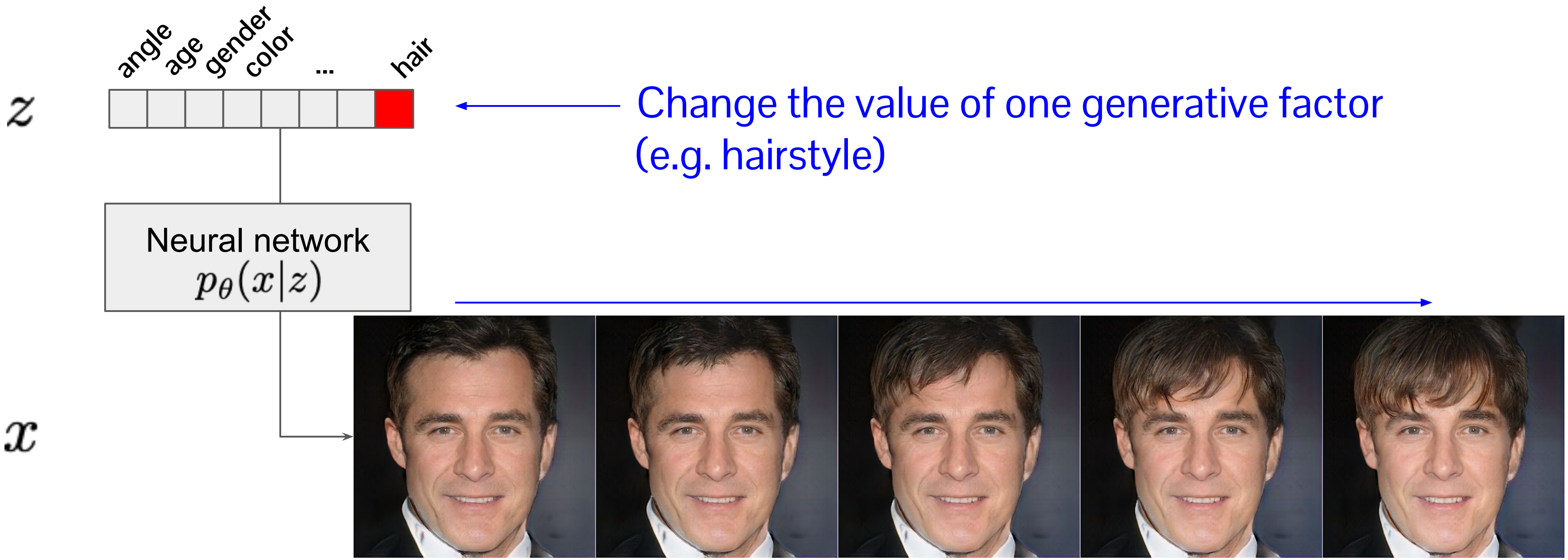
Why do we care about latent variable?

- It allows us to interpret and manipulate data much more easily
- **Example:** manipulating generative factor

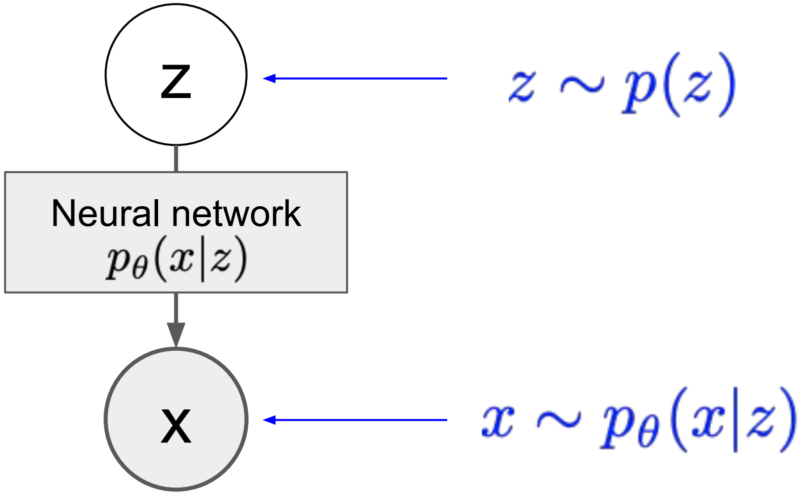


Why do we care about latent variable?

- It allows us to interpret and manipulate data much more easily
- **Example:** manipulating generative factor

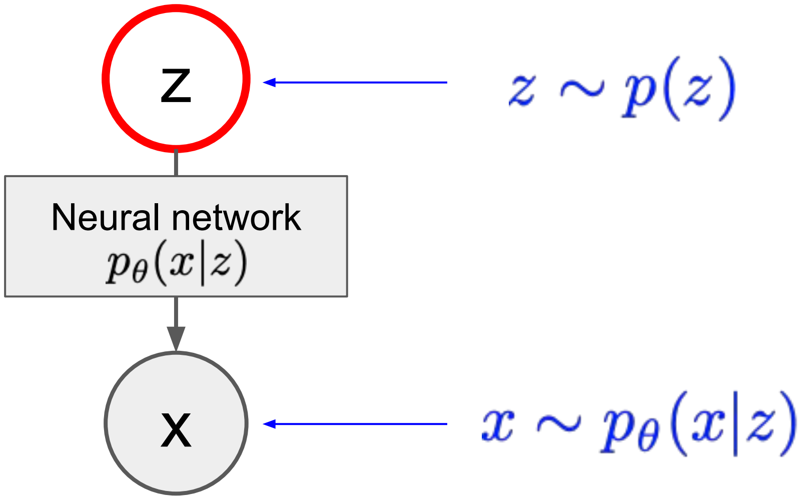


Remaining questions



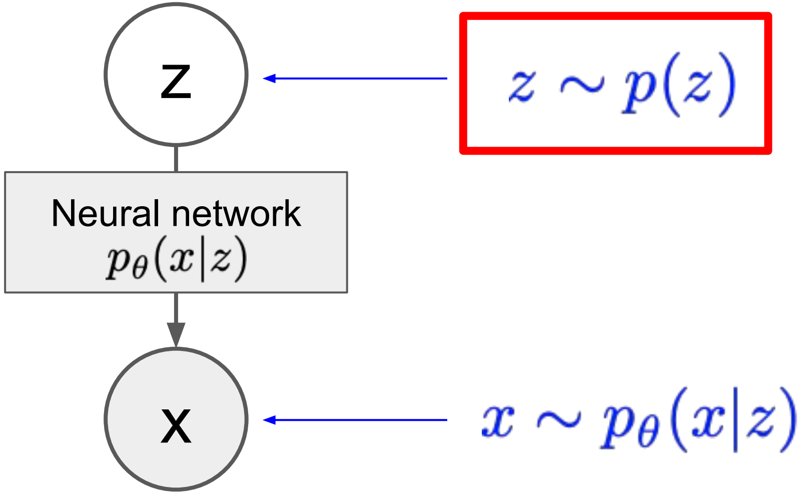
Remaining questions

- How do we learn latent variable model? There is no supervision (cause it is latent)



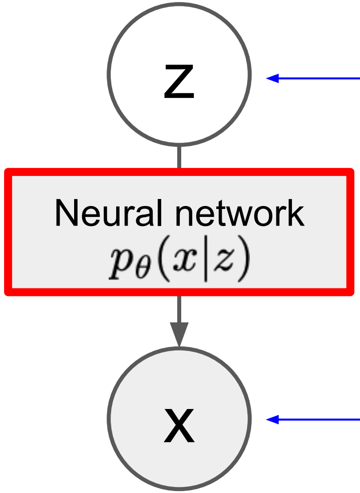
Remaining questions

- How do we learn latent variable model? There is no supervision (cause it is latent)
- How do we sample from the latent variable? We do not know $p(z)$



Remaining questions

- How do we learn latent variable model? There is no supervision (cause it is latent)
- How do we sample from the latent variable? We do not know $p(z)$
- How do we design the generator? ← decoder 이용함.



$z \sim p(z)$

It is decoder [illegible]

$x \sim p_\theta(x|z)$

Today's agenda

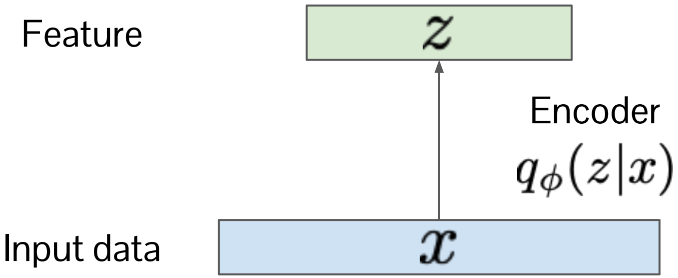
- Latent variable model
- **Autoencoder**
- Variational Autoencoder

Learning compact representation of data

We want to learn an encoder that maps an input data to a compact representation

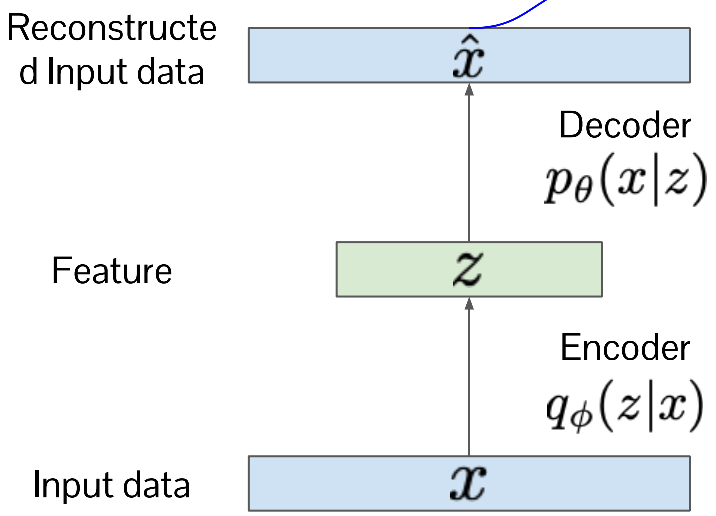
z is much smaller than **x** but encoding most useful information of **x**

How do we learn such z?



Autoencoder: unsupervised learning of latent representation

Learn to reconstruct its original input



즉, some degree of details 잃어버림.
→ identity mapping은 학습하지 않는다.

Train both encoder and decoder
to reduce the reconstruction loss

$$\arg\min_{\theta,\phi} \|x - \hat{x}\|_2^2$$

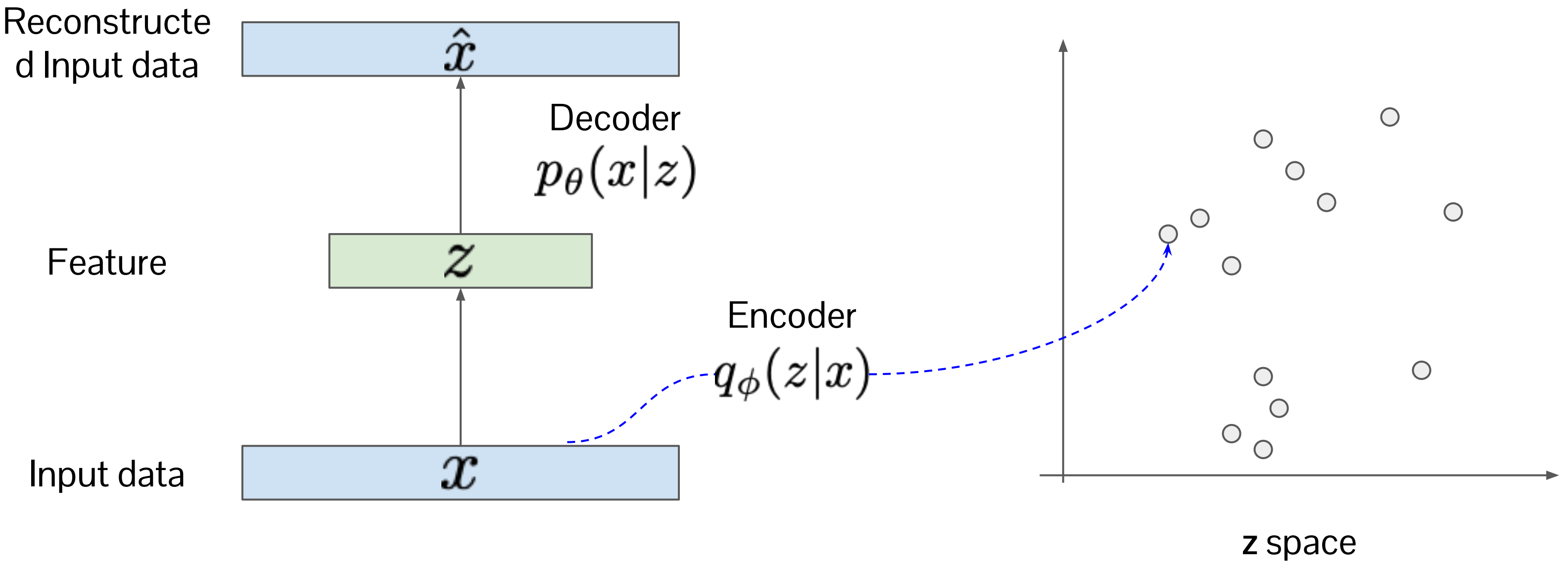
- Ideally, it should learn identity mapping
  → 하지만 dim z << dim x 로 하여 info. loss를 일으킬수있다.
- Still it learns useful representation. **Why?**

  → It should squash useful information of **x** in a small dimensional vector **z** for reconstruction
- Learning is purely unsupervised!

Autoencoder: unsupervised learning of latent representation

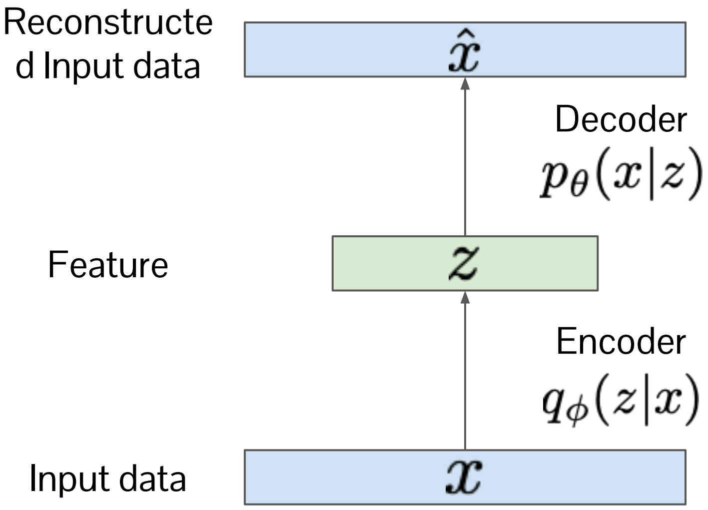
Learn to reconstruct its original input

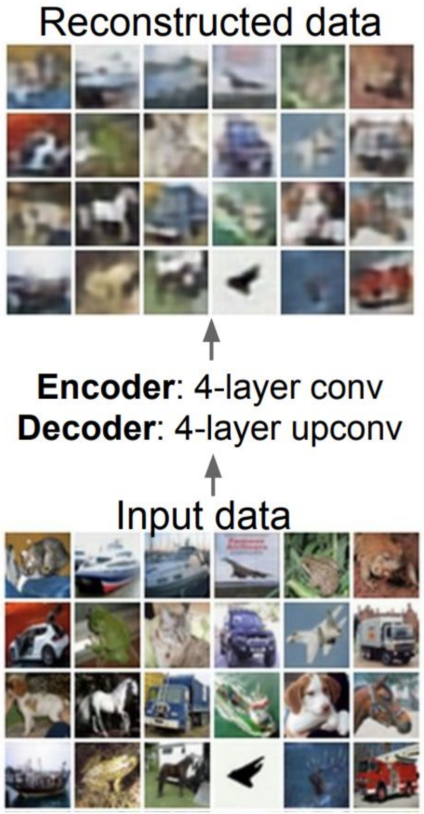
The model maps every input data to latent feature by encoder, and reconstruct it back to data space by decoder



Autoencoder: unsupervised learning of latent representation

Learn to reconstruct its original input

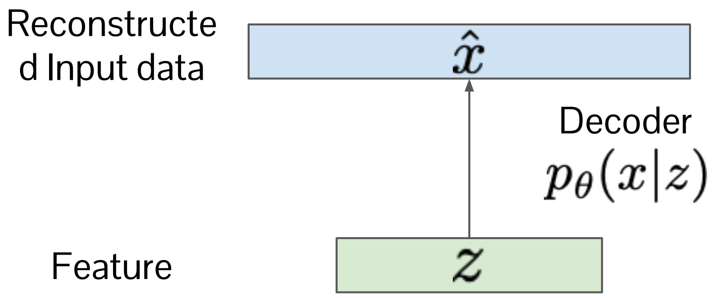




Autoencoder for generation

- Can we generate a new data using autoencoder?

Maybe we can throw the encoder after training,
and generate samples using decoder



2. Generate data by decoder $x \sim p_\theta(x|z)$

1. Sample a feature $z \sim \boxed{p(z)}$

Q: Any issues?

A: We do not know the prior

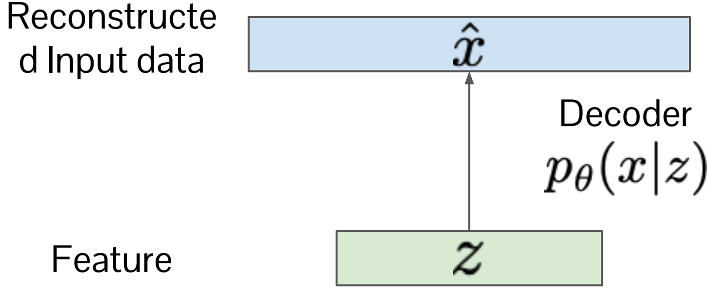
(don't know how the data is distributed in the latent space)

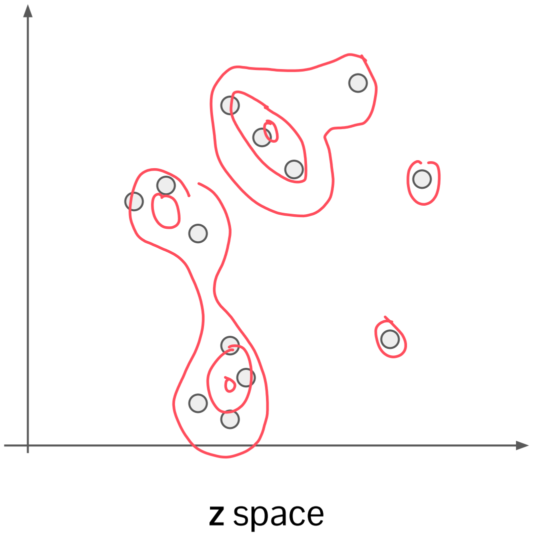
Autoencoder for generation

- Can we generate a new data using autoencoder?

sol ① training data로 만든 거로 z 학습.
② 뒤에 설명

이렇게 density를 알아야함.





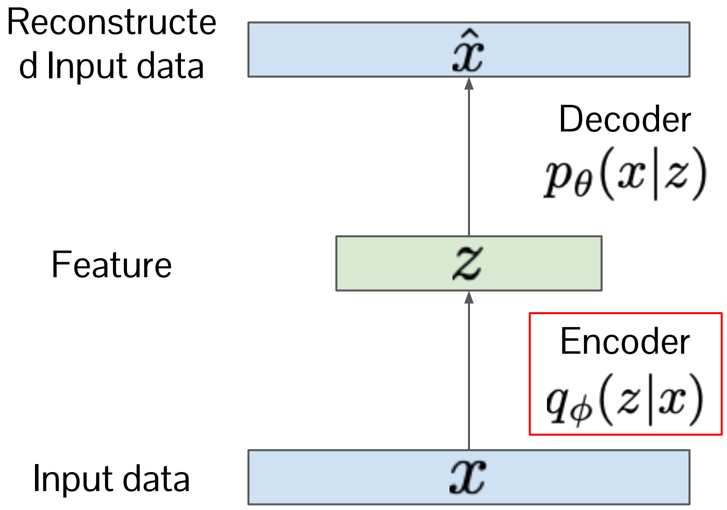
What should we sample here?

Q: Any issues?

A: We do not know the prior $p(z)$
(don't know how the data is distributed in the latent space)

Autoencoder for generation

What if we constrain the distribution of latent features?



Force the encoder distribution follows a simple distribution (e.g. standard gaussian)

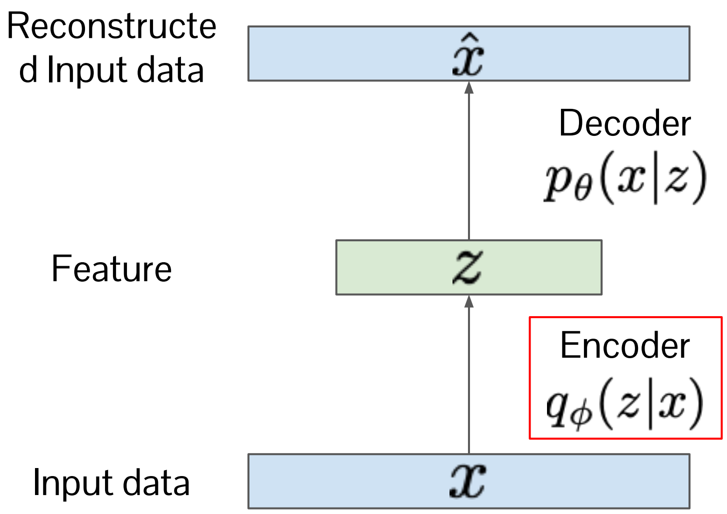
$$\min_\phi D_{KL}(q_\phi(z|x)||p(z))$$

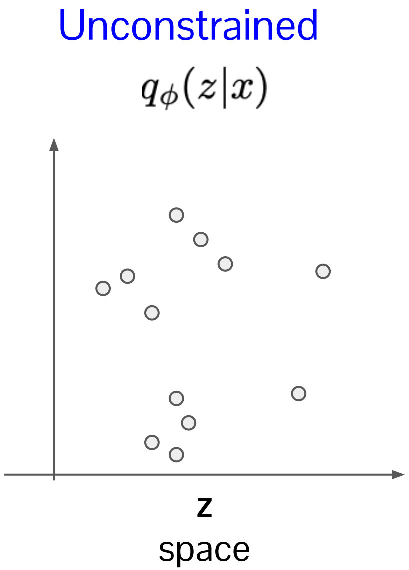
Encoder output

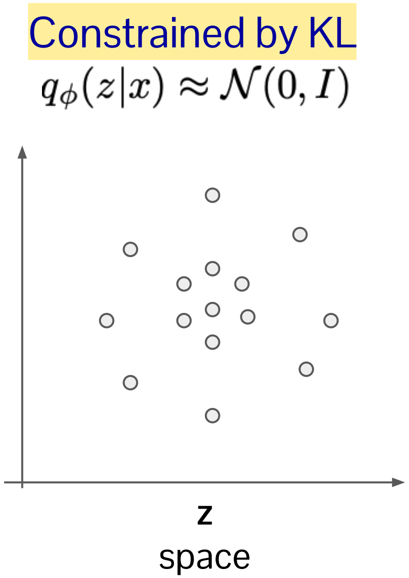
Simple prior
$p(z) = \mathcal{N}(0, I)$

Autoencoder for generation

What if we constrain the distribution of latent features?





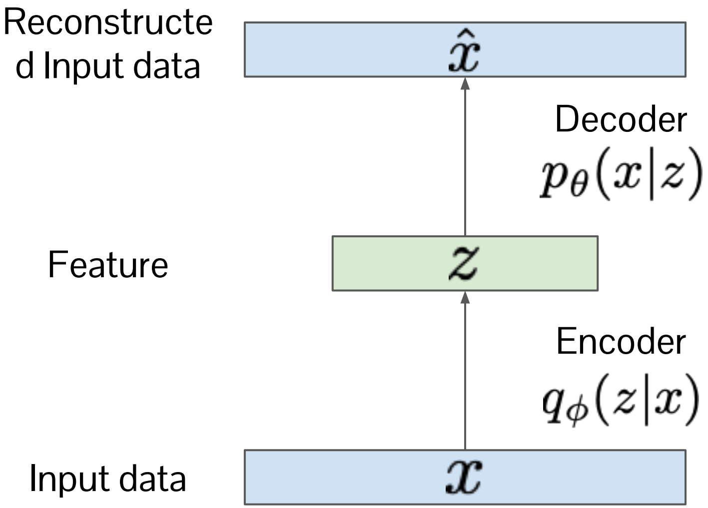


We can now sample z by

$$z \sim \mathcal{N}(0, I)$$

Autoencoder for generation

Training of autoencoder as a generative model



Overall objective

$$\arg\min_{\theta,\phi} \sum_i \|x^{(i)} - \hat{x}^{(i)}\|_2^2 + D_{KL}(q_\phi(z|x^{(i)})||p(z))$$

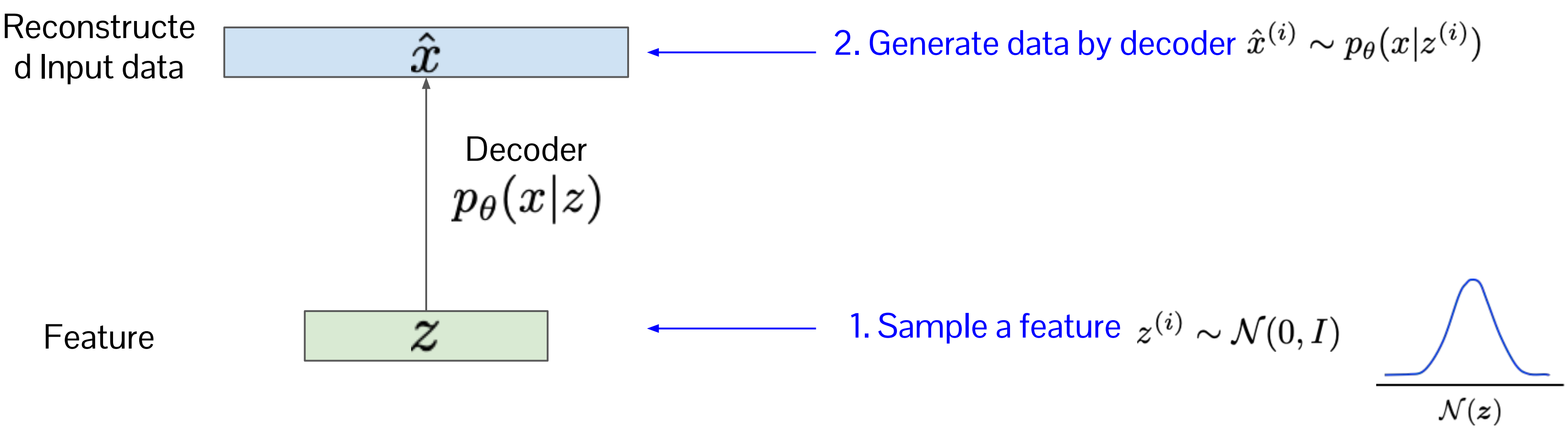
$\hat{x}$

if $p(z) = N(0, I)$,

↳ x가 가지면 learn each dimension independently.

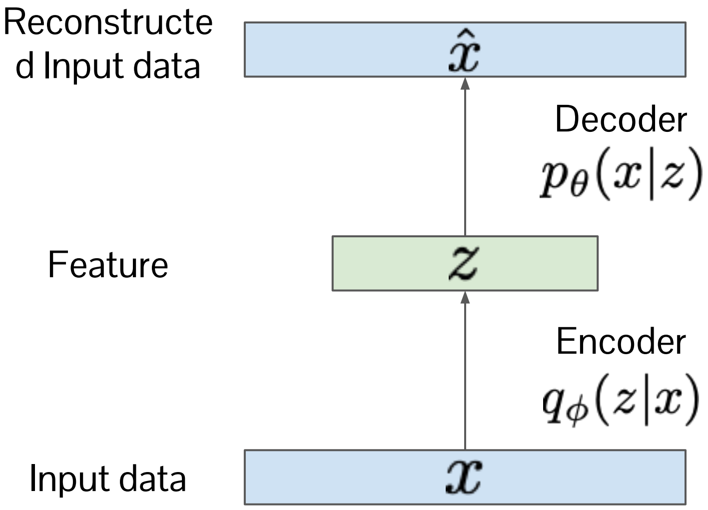
Autoencoder for generation

Inference of autoencoder as a generative model



Autoencoder for generation

What are we doing in this objective **exactly**?



Overall objective

$$\arg\min_{\theta,\phi} \sum_i \|x^{(i)} - \hat{x}^{(i)}\|_2^2 + D_{KL}(q_\phi(z|x^{(i)})||p(z))$$

We will derive how we can arrive this equation in perspective of generative modeling!

(i.e. maximizing observation likelihood)

Today's agenda

- Latent variable model
- Autoencoder
- **Variational Autoencoder**

Recap: objective of deep generative models

Maximum Likelihood Estimation (MLE)

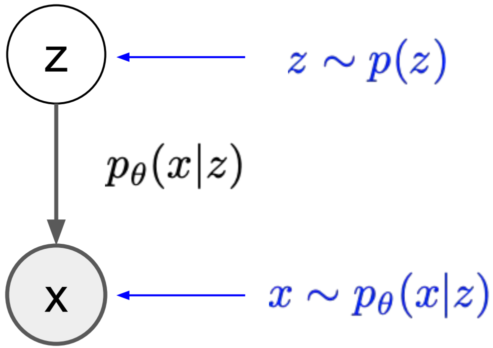
$$\arg\max_{\theta \in \Theta} \sum_i \log p_\theta(x^{(i)})$$

Recap: objective of deep generative models

Maximum Likelihood Estimation (MLE)

$$\arg\max_{\theta \in \Theta} \sum_i \log p_\theta(x^{(i)})$$

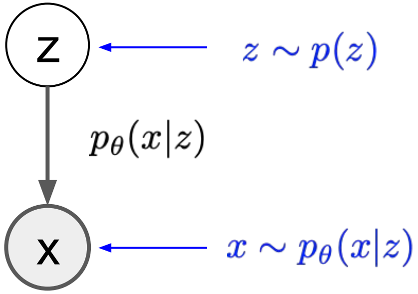
We want to optimize this with latent variable model



Likelihood estimation with latent variable

For notational brevity, let $x = x^{(i)}$. Then,

$$\begin{aligned}\log p(x) &= \log \int p(x, z) dz \\ &= \log \int p_\theta(x|z) p(z) dz\end{aligned}$$



Likelihood estimation with latent variable

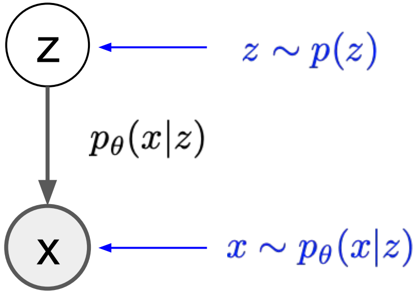
For notational brevity, let $x = x^{(i)}$. Then,

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

다루기 힘든

It is intractable. why?

→ integration over all latent variables
So we are going to approximate this term



Likelihood estimation with latent variable

For notational brevity, let $x = x^{(i)}$. Then,

$$\begin{aligned}
\log p(x) &= \log \int p(x, z) dz \\
&= \log \int p_\theta(x|z) p(z) dz \\
&= \log \int p_\theta(x|z) p(z) \frac{q_\phi(z|x)}{q_\phi(z|x)} dz
\end{aligned}$$

(Multiply by one)

Likelihood estimation with latent variable

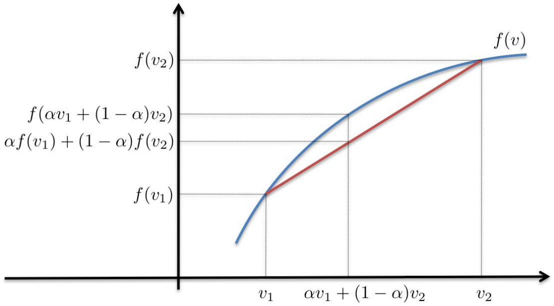
For notational brevity, let $x = x^{(i)}$. Then,

$$
\begin{aligned}
\log p(x) &= \log \int p(x, z) dz \\
&= \log \int p_\theta(x|z) p(z) dz \\
&= \log \int p_\theta(x|z) p(z) \frac{q_\phi(z|x)}{q_\phi(z|x)} dz && \text{(Multiply by one)} \\
&= \log \mathbb{E}_{q_\phi(z|x)} \left[\frac{p_\theta(x|z) p(z)}{q_\phi(z|x)} \right] && \text{(Rewrite in expectation form)}
\end{aligned}
$$

Likelihood estimation with latent variable

For notational brevity, let $x = x^{(i)}$. Then,

$$\begin{aligned}
\log p(x) &= \log \int p(x,z)dz \\
&= \log \int p_\theta(x|z)p(z)dz \\
&= \log \int p_\theta(x|z)p(z)\frac{q_\phi(z|x)}{q_\phi(z|x)}dz && \text{(Multiply by one)} \\
&= \log \mathbb{E}_{q_\phi(z|x)}\left[\frac{p_\theta(x|z)p(z)}{q_\phi(z|x)}\right] && \text{(Rewrite in expectation form)} \\
&\geq \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p_\theta(x|z)p(z)}{q_\phi(z|x)}\right] && \text{(Jensen's inequality)}
\end{aligned}$$



$$f(\alpha v_1 + (1-\alpha)v_2) \geq \alpha f(v_1) + (1-\alpha)f(v_2)$$

It holds for any concave function.
We can use it here since log is concave.

Likelihood estimation with latent variable

For notational brevity, let $x = x^{(i)}$. Then,

$$
\begin{aligned}
\log p(x) &= \log \int p(x, z) dz \\
&= \log \int p_\theta(x|z) p(z) dz \\
&= \log \int p_\theta(x|z) p(z) \frac{q_\phi(z|x)}{q_\phi(z|x)} dz && \text{(Multiply by one)} \\
&= \log \mathbb{E}_{q_\phi(z|x)} \left[\frac{p_\theta(x|z) p(z)}{q_\phi(z|x)} \right] && \text{(Rewrite in expectation form)} \\
&\geq \mathbb{E}_{q_\phi(z|x)} \left[\log \frac{p_\theta(x|z) p(z)}{q_\phi(z|x)} \right] && \text{(Jensen's inequality)} \\
&= \mathbb{E}_{q_\phi(z|x)} \left[\log p_\theta(x|z) + \log \frac{p(z)}{q_\phi(z|x)} \right] \\
&= \mathbb{E}_{q_\phi(z|x)} [\log p_\theta(x|z)] + \int q_\phi(z|x) \log \frac{p(z)}{q_\phi(z|x)} dz && \text{(Distribute expectation)} \\
&= \mathbb{E}_{q_\phi(z|x)} [\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x) || p(z)) && \text{(Rewrite in KL divergence)}
\end{aligned}
$$

Likelihood estimation with latent variable

To summarize, we just derived:

$$\underbrace{\log p(x^{(i)})}_{\text{intractable}} \geq \underbrace{\mathbb{E}_{q_\phi(z|x^{(i)})}[\log p_\theta(x^{(i)}|z)] - D_{KL}(q_\phi(z|x^{(i)})||p(z))}_{\text{tractable}}$$

This is the likelihood we want to maximize

This is referred as a **variational** lower bound of likelihood

By maximizing this, we are indirectly maximizing the likelihood

The objective of latent variable generative model

$$\arg\max_{\theta,\phi} \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{q_\phi(z|x^{(i)})}[\log p_\theta(x^{(i)}|z)] - D_{KL}(q_\phi(z|x^{(i)})||p(z))$$

Variational AutoEncoder (VAE)

The objective of latent variable generative model

$$\arg\max_{\theta,\phi} \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{q_\phi(z|x^{(i)})}[\log p_\theta(x^{(i)}|z)] - D_{KL}(q_\phi(z|x^{(i)})||p(z))$$

Architecture for latent variable model

Input Data x

Variational AutoEncoder (VAE)

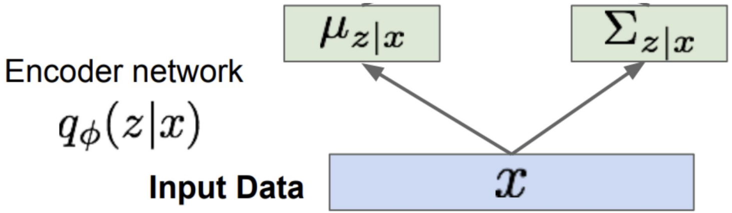
The objective of latent variable generative model

$$\arg\max_{\theta,\phi} \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{q_\phi(z|x^{(i)})}[\log p_\theta(x^{(i)}|z)] - D_{KL}(q_\phi(z|x^{(i)})||p(z))$$

We design the encoder network
to model Gaussian distribution

$$q_\phi(z|x) = \mathcal{N}(\mu_{z|x}, \Sigma_{z|x})$$

Architecture for latent variable model



Variational AutoEncoder (VAE)

The objective of latent variable generative model

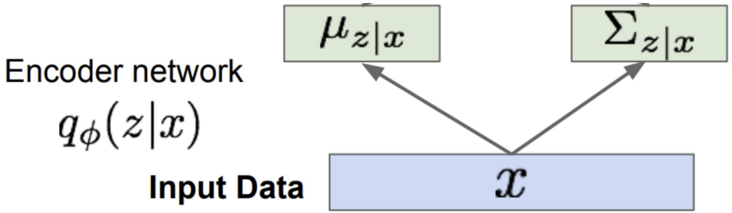
$$\arg\max_{\theta,\phi} \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{q_\phi(z|x^{(i)})}[\log p_\theta(x^{(i)}|z)] - D_{KL}(q_\phi(z|x^{(i)})||p(z))$$

We design the encoder network
to model Gaussian distribution

$$q_\phi(z|x) = \mathcal{N}(\mu_{z|x}, \sigma^2_{z|x} I)$$

For simplicity, we usually choose the
Isotropic Gaussian

Architecture for latent variable model



Variational AutoEncoder (VAE)

The objective of latent variable generative model

$$\arg\max_{\theta,\phi} \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{q_\phi(z|x^{(i)})}[\log p_\theta(x^{(i)}|z)] - D_{KL}(q_\phi(z|x^{(i)})||p(z))$$

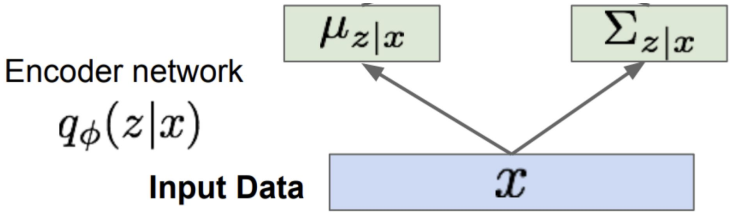
We design the encoder network
to model Gaussian distribution

$$q_\phi(z|x) = \mathcal{N}(\mu_{z|x}, \sigma^2_{z|x} I)$$

We choose simple prior
such as standard Normal

$$p(z) = \mathcal{N}(0, I)$$

Architecture for latent variable model



Variational AutoEncoder (VAE)

The objective of latent variable generative model

$$\arg\max_{\theta,\phi} \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{q_\phi(z|x^{(i)})}[\log p_\theta(x^{(i)}|z)] - D_{KL}(q_\phi(z|x^{(i)})||p(z))$$

This term constrain the encoder outputs to follow the prior distribution

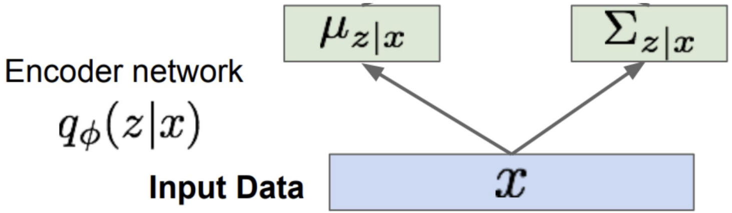
We design the encoder network to model Gaussian distribution

$$q_\phi(z|x) = \mathcal{N}(\mu_{z|x}, \sigma^2_{z|x} I)$$

We choose simple prior such as standard Normal

$$p(z) = \mathcal{N}(0, I)$$

Architecture for latent variable model



Variational AutoEncoder (VAE)

The objective of latent variable generative model

$$\arg\max_{\theta,\phi} \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{q_\phi(z|x^{(i)})}[\log p_\theta(x^{(i)}|z)] - D_{KL}(q_\phi(z|x^{(i)})||p(z))$$

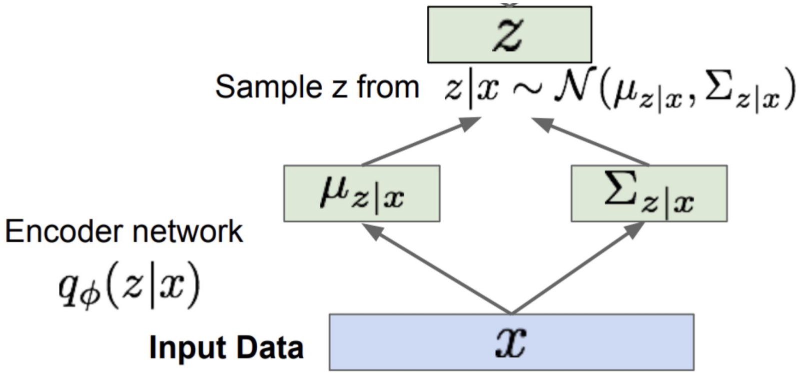
We design the encoder network to model Gaussian distribution

$$q_\phi(z|x) = \mathcal{N}(\mu_{z|x}, \sigma^2_{z|x} I)$$

The latent variable is then sampled from q

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

Architecture for latent variable model



Variational AutoEncoder (VAE)

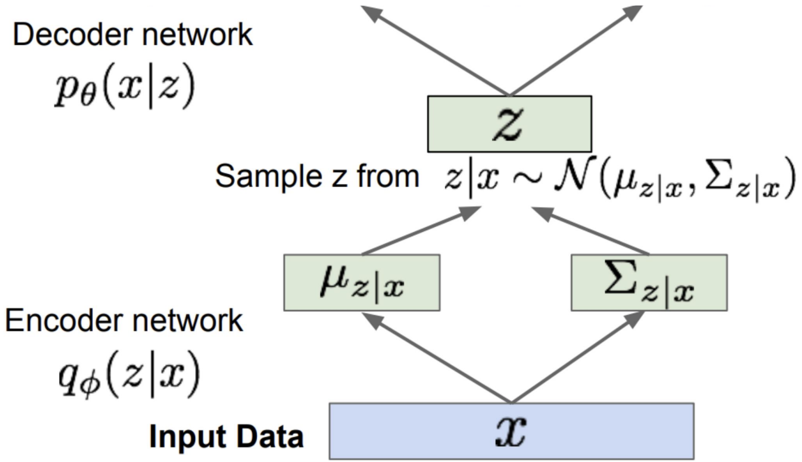
The objective of latent variable generative model

$$\arg\max_{\theta,\phi} \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{q_\phi(z|x^{(i)})}[\log p_\theta(x^{(i)}|z)] - D_{KL}(q_\phi(z|x^{(i)})||p(z))$$

The sampled latent is fed to decoder

$$p_\theta(x|z)$$

Architecture for latent variable model



Variational AutoEncoder (VAE)

The objective of latent variable generative model

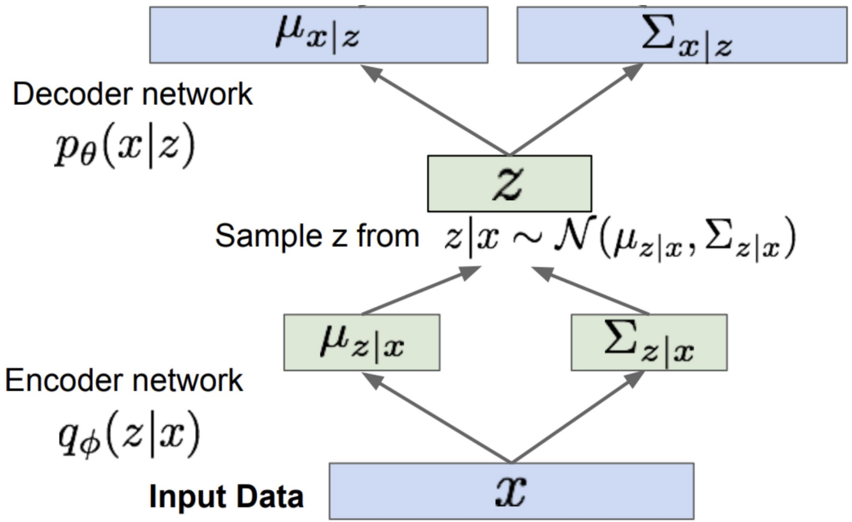
$$\arg\max_{\theta,\phi} \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{q_\phi(z|x^{(i)})}[\log p_\theta(x^{(i)}|z)] - D_{KL}(q_\phi(z|x^{(i)})||p(z))$$

The sampled latent is fed to decoder

$$p_\theta(x|z) = \mathcal{N}(\mu_{x|z}, \Sigma_{x|z})$$

which also models data distribution using
Gaussian distribution (usually unit variance)

Architecture for latent variable model



Variational AutoEncoder (VAE)

The objective of latent variable generative model

$$\arg\max_{\theta,\phi} \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{q_\phi(z|x^{(i)})}[\log p_\theta(x^{(i)}|z)] - D_{KL}(q_\phi(z|x^{(i)})||p(z))$$

The sampled latent is fed to decoder

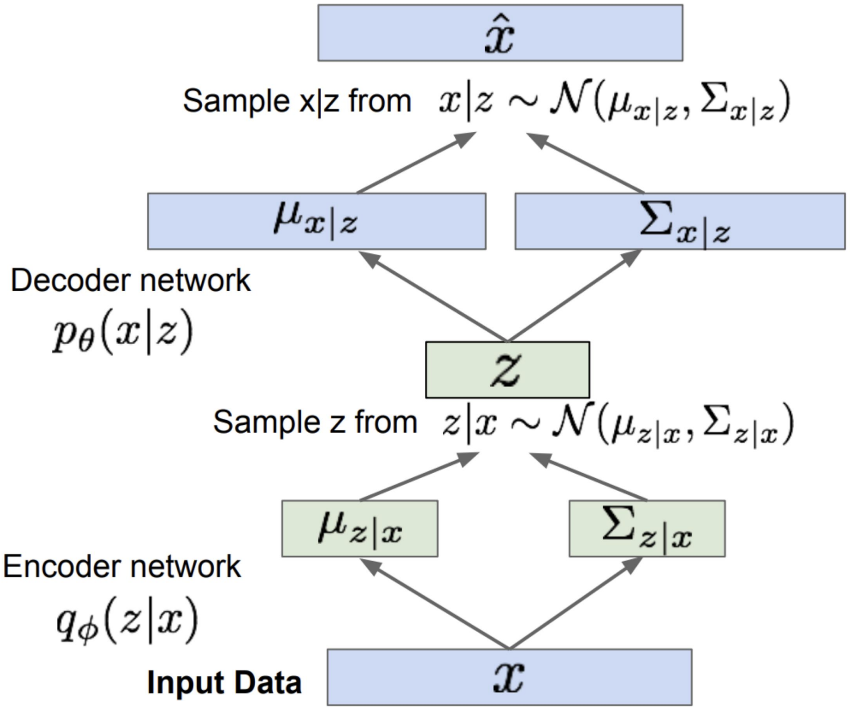
$$p_\theta(x|z) = \mathcal{N}(\mu_{x|z}, \Sigma_{x|z})$$

which also models data distribution using Gaussian distribution (usually unit variance)

The final output of the model is then generated by sampling from decoder

$$\hat{x} \sim \mathcal{N}(\mu_{x|z}, \Sigma_{x|z})$$

Architecture for latent variable model



P: sampling이 있기 때문에
not differentiable

Variational AutoEncoder (VAE)

The objective of latent variable generative model

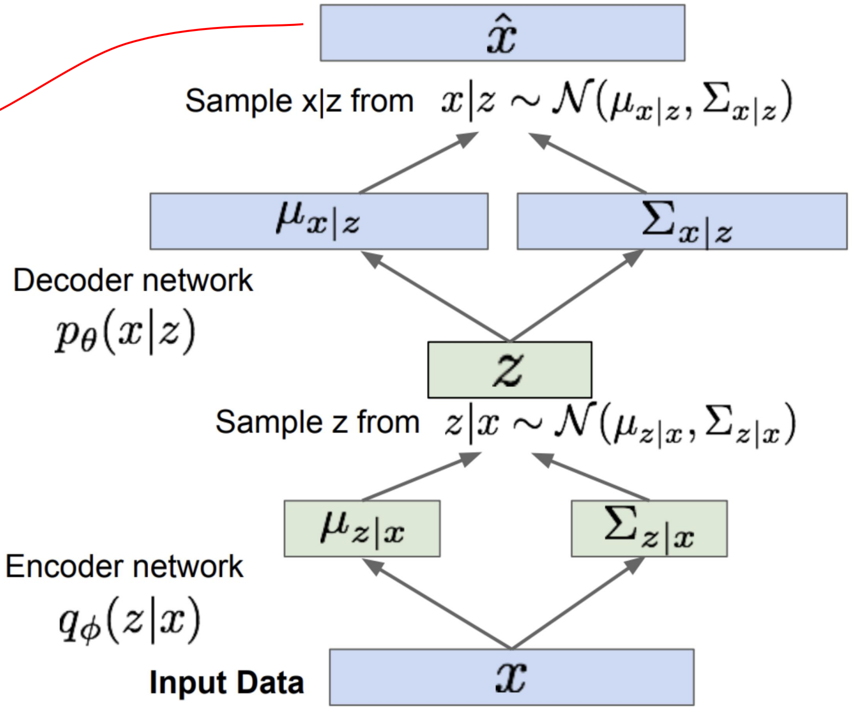
$$\arg\max_{\theta,\phi} \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{q_\phi(z|x^{(i)})}[\log p_\theta(x^{(i)}|z)] - D_{KL}(q_\phi(z|x^{(i)})||p(z))$$

Then we compute the reconstruct loss
between the input and generated output

With Gaussian decoder with unit variance,
this is same as computing the L2 distance

$$\mathbb{E}_{q_\phi(z|x^{(i)})}[\log p_\theta(x^{(i)}|z)] = -\|\hat{x} - x\|_2^2$$

Architecture for latent variable model



Variational AutoEncoder (VAE)

- How do we backprop through sampling?

The sampling is a discrete operation, hence it is not differentiable!

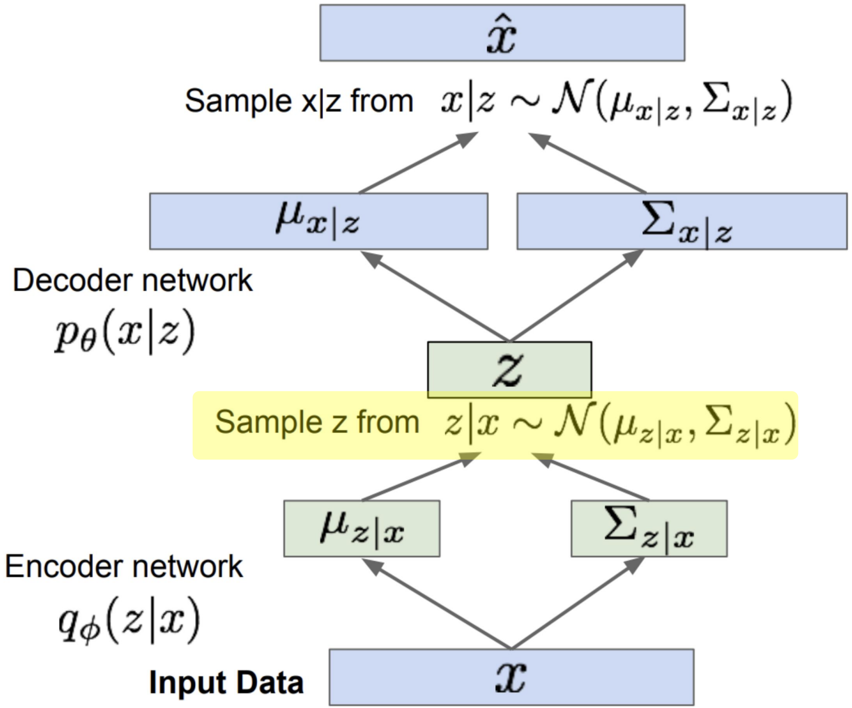
- **Reparameterization trick**

$$q_\phi(z|x) = \mathcal{N}(\mu_{z|x}, \sigma^2_{z|x} I)$$
$$= \mu_{z|x} + \sigma_{z|x} \cdot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$$

The equation is differentiable w.r.t. parameters of the Gaussian distribution, but not w.r.t. sampled noise

This is sufficient since we want to learn the parameters!

Architecture for latent variable model



Summary: VAE

- Latent variable model
- Optimizing a variational lower-bound of likelihood
- Nice probabilistic interpretation