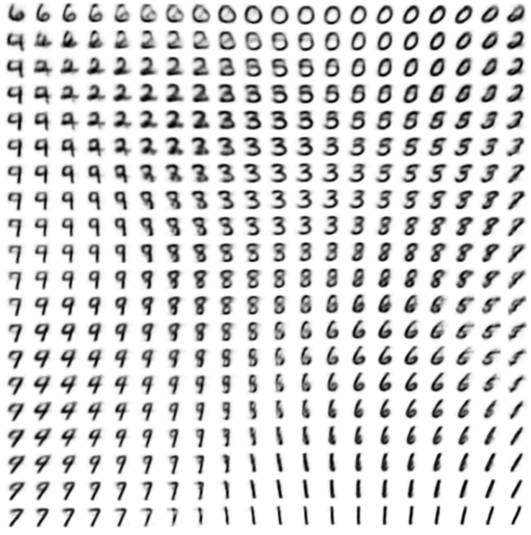
# Understanding Variational Autoencoders

- ICLR 2014
- Unsupervised generative model
- Generating samples ✗

(a) Learned Frey Face manifold



(b) Learned MNIST manifold

## Auto-Encoding Variational Bayes

**Diederik P. Kingma**
Machine Learning Group
Universiteit van Amsterdam
dpkingma@gmail.com

**Max Welling**
Machine Learning Group
Universiteit van Amsterdam
welling.max@gmail.com

Warm - up

- Supervised learning and unsupervised learning
- Notations of Deep learning

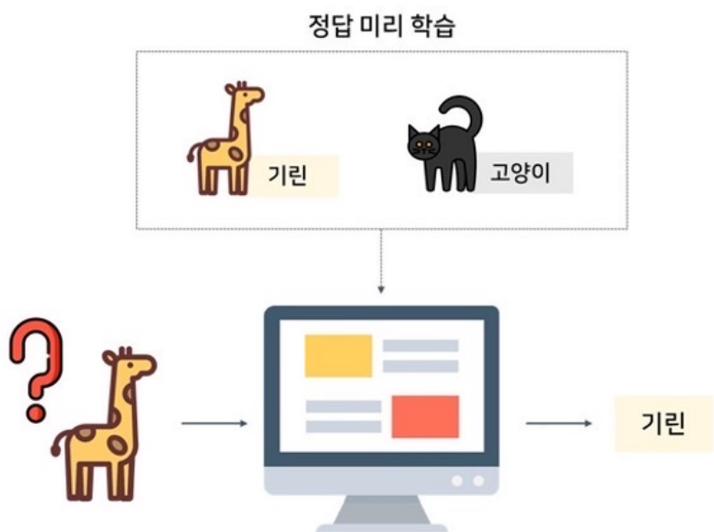
$x$ input (image or tabular data)

$y(x)$ or $\hat{y}(x)$ output of model

$t$ or $\#$ target value

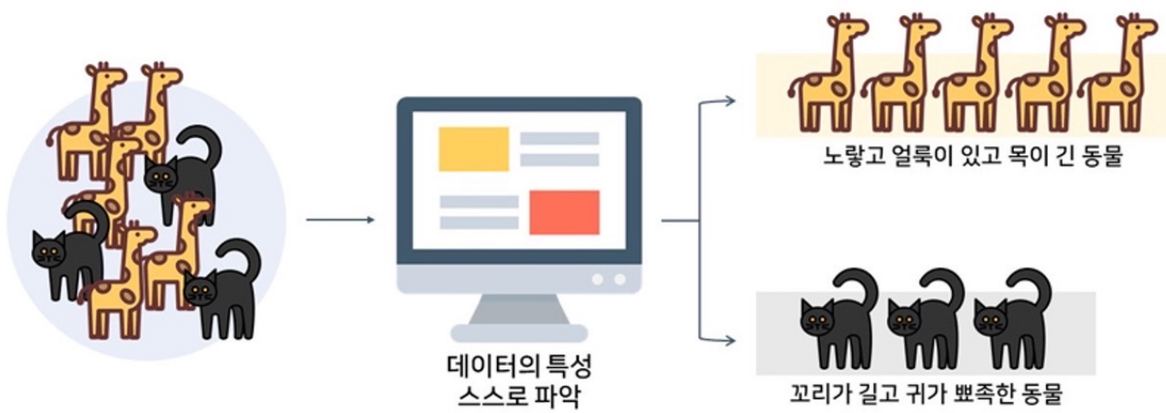
# Supervised learning

- 학습 데이터에 정답 or 라벨 $y$가 주어진 경우
- $x$ 와 $y$ 설명하는 함수 찾기
- goal : $y(x) \sim y$ 인 $y(x)$ 찾기
- e.g: regression, classification



# Unsupervised learning

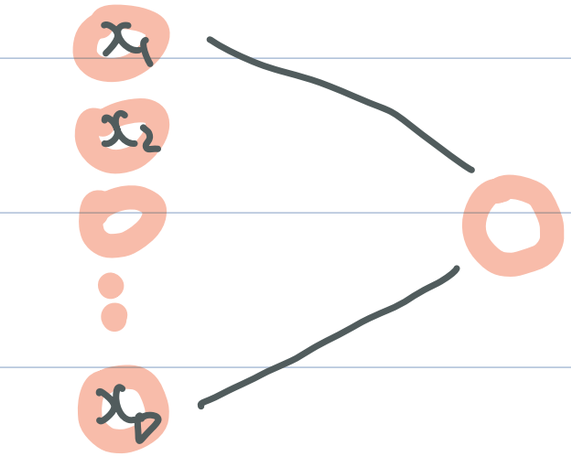
- 데이터의 특징이나 구성 찾는 문제
- 학습데이터에 x 만 주어진 경우
- e.g: clustering, feature extraction, dimensionality reduction



# Notations of Deep learning

- $X$ input (image or tabular data)
- $y(X)$ or $\hat{y}(X)$ output of model
- $t$ or $\hat{t}$ target value
- Perceptron, activation function, weight, bias
- Input, hidden, output layer, rule of activation functions
- cost (loss) function, optimization algorithm (method)

  backpropagation algorithm




$x \in \mathbb{R}^D$

$w \in \mathbb{R}^D$ , $b \in \mathbb{R}$

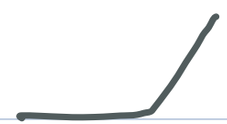
weight bias

$w^T x + b$ perceptron

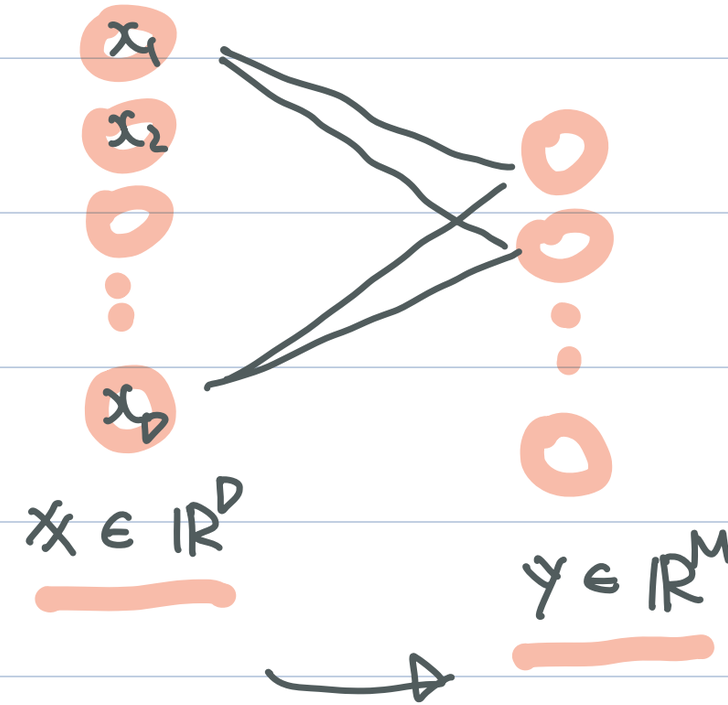
$f$ : non-linear function

is called activation function

sigmoid, ReLU, tanh, .....



$y(x) = \max(0, x)$




$w \in \mathbb{R}^D$ , $b \in \mathbb{R}$ $\quad w^T x + b$

weight bias

weight

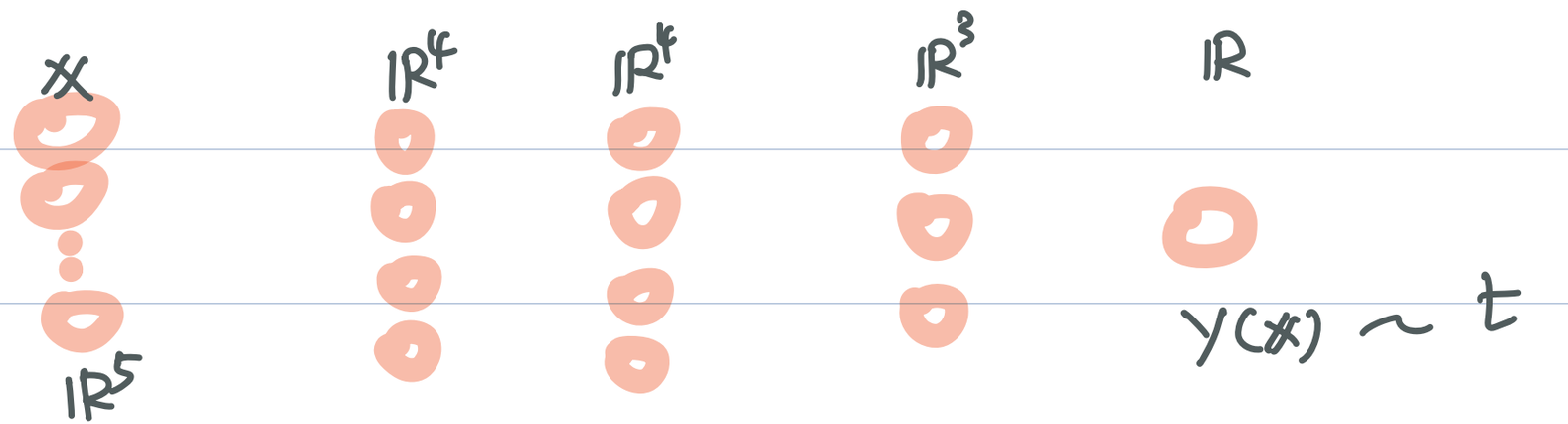
$W$ : $D \times M$ matrix

$b \in \mathbb{R}^M$

$f(x^T W + b)$

$\frac{1}{N} \sum (y(x) - t)^2$

예) $x$ input $t$ 값을 예측




1차원

# Autoencoder

- $y(x) \sim x$ 가 되게 끔 cost 정의

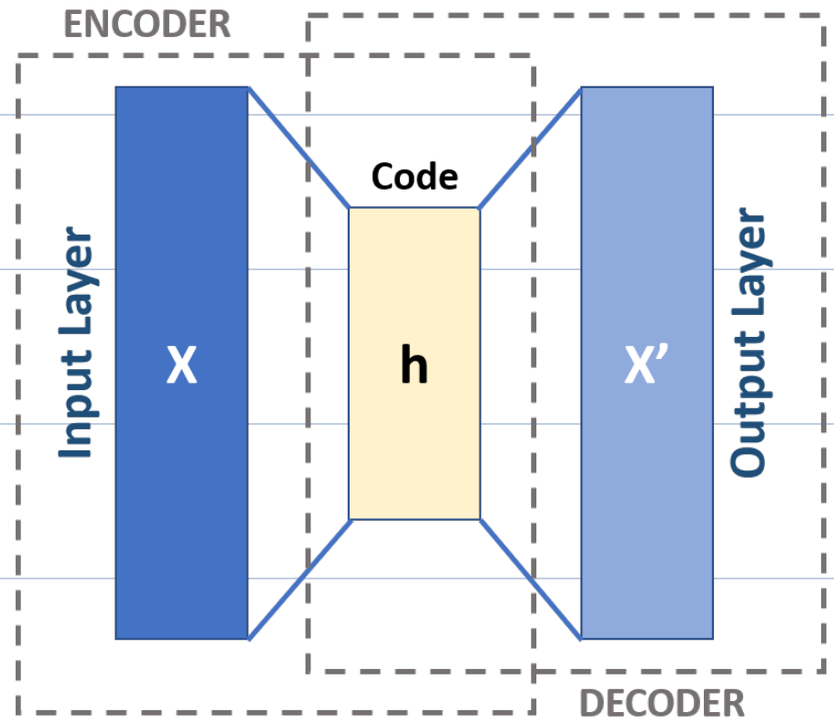
  e.g. $\sum_n \{y(x_n) - x_n\}^2$

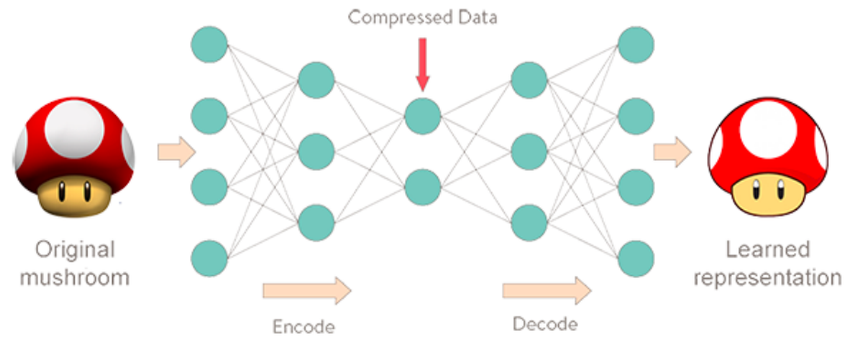
- 일반적으로 encoder 와 decoder 모양은 대칭으로 구성

- 중: latent variable, bottleneck layer

  feature or hidden representation.

- Nonlinear identity function



# Applications of Autoencoder

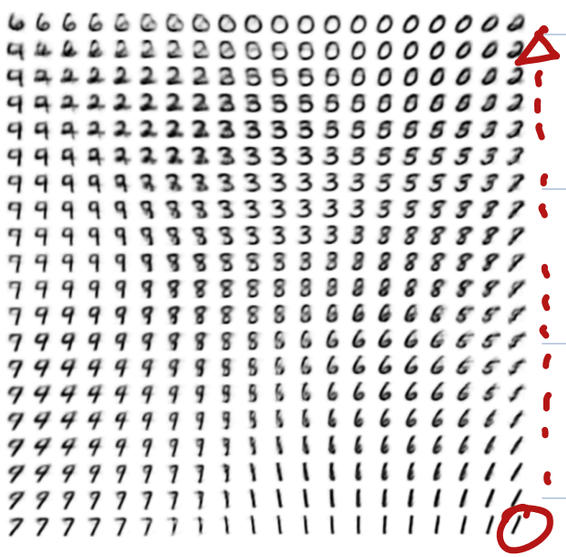
- Feature extract, dimension reduction
- Anomaly detection (reconstruction based)
- Fine tuning, denoising

# Variational Autoencoder (VAE)

- Generative modeling: deal with models of distribution $p(x)$.
- Latent variable $z$, $p(z)$ is defined over $z$

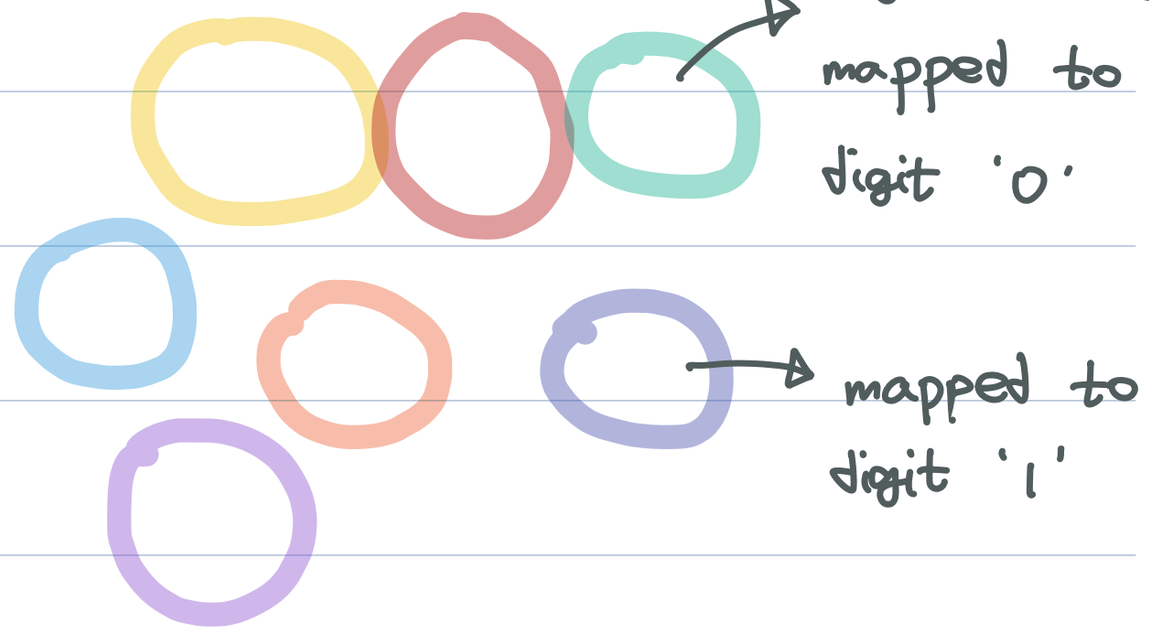
$p(x, z)$ joint distribution

(a) Learned Frey Face manifold



(b) Learned MNIST manifold

$p(z|x)$




Space of $z|x$

$p(x)$ (observations $x_1 \dots x_N$) / want posterior $p(z|x)$

$p(z)$ prior over $z$ / why?

We want to know the posterior $p(z|x)$ so we (never)

will approximate it with variational distribution $q_\theta(z|x)$

Let $q_\theta(z|x)$ be a Gaussian distribution as

$$q_\theta(z|x) = \mathcal{N}(z \mid \mu_\theta(x), \sigma_\theta(x)^2 I)$$

D-dim

$(0 \dots D)$

$\mu_\theta$ and $\sigma_\theta$ is some neural network function of $x$

parametrized by $\theta$.

Goal : Maximize Log Marginal likelihood of given data

i.e. maximize $\frac{1}{N} \sum_n \log p(x_i)$ $(x_n)_{n=1,2,..N}$

Step 1.

$$\log p(x) = \log \frac{p(z)\, p(x \mid z)}{p(z \mid x)} = \log p(z) + \log p(x \mid z) - \log p(z \mid x)$$

Step 2. (approximating posterior $q(z|x)$ over $z$.)

$$\log p(x) = \log p(x) \underbrace{\int q(z|x)\,dz}_{=1}$$

$$= \int q(z|x) \left\{ \log p(z) + \log p(x|z) - \log p(z|x) \right\} dz$$

$$= \mathbb{E}_{q(z|x)}[\log p(x|z)] + \int q(z|x) \log p(z)\,dz$$

$$- \int q(z|x) \log p(z|x)\,dz$$

$$\int f(x)\,p(x)\,dx$$

$$= \mathbb{E}_{p(x)}[f(x)]$$

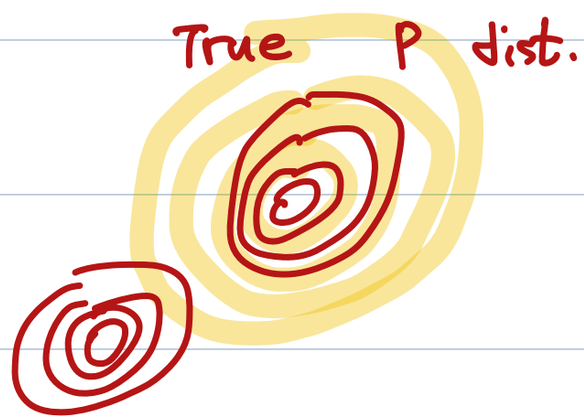
Step 3

$$\log p(x) \pm \int q(z|x) \log q(z|x)\, dz$$

$$= \mathbb{E}_{q(z|x)}[\log p(x|z)] - KL(q(z|x) \| p(z)) + KL(q(z|x) \| p(z|x))$$

???

$$\left( KL(p\|q) := -\int p(z) \log \left\{ \frac{q(z)}{p(z)} \right\} dz \right)$$




reverse KL-D : $KL(q \| p)$ (true, approximation)

$KL(p\|q)$ 를 최소화하는 unimodal Gaussian

$KL(q\|p)$ " '

Clean - up

$$\log p(x)$$

$$= \mathbb{E}_{q(z|x)}[\log p(x|z)] - KL(q(z|x) \| p(z)) + \underbrace{KL(q(z|x) \| p(z|x))}_{\geq 0}$$

$$\geq \mathbb{E}_{q(z|x)}[\log p(x|z)] - KL(q(z|x) \| p(z))$$

$$\approx \frac{1}{J} \sum_{j=1}^{J} \log p(x|z_j) \qquad z_j \sim q(z|x)$$

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

1. Maximize $p(x)$

2. Introduce prior $p(z)$ and joint distribution $p(x, z)$

3. Introduce approximating posterior $q(z|x) = \mathcal{N}(z|\mu(x), \sigma(x)I)$

$p(z|x$

## Actual calculation

- Encoder (approximating posterior): $q(z|x) := \mathcal{N}(z|\mu(x), \sigma(x) I)$

  where $\mu$, $\sigma$ are neural net. (D-dim random vector)

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

- Likelihood / Decoder : $p(x|z)$ depending on input data

- Since $p(z)$ and $q(z|x)$ are Gaussian,

$$KL(q(z|x) \| p(z)) = \frac{1}{2} \sum_{j=1}^{D} \{ 1 + \log(\sigma_j^2(x)) - \mu_j^2(x) - \sigma_j^2(x) \}$$

where $\mu(x) = (\mu_1(x), \ldots \mu_D(x))^T$, $\sigma(x) = (\sigma_1(x), \ldots \sigma_D(x))^T$

# Autoencoder

encoder → $h(x)$ → decoder → $q(h(x))$ ≈ $x$

$x$

$h(x)$: D-dim multi-valued function

$x_1 = x_2 \Rightarrow h(x_1) = h(x_2)$ and $q(h(x_1)) = q(h(x_2))$

# VAE

encoder → $\mu(x), \sigma(x) I$

$x$

D-dim multi-valued functions

sample $z$ from $q(z|x) = \mathcal{N}(z | \mu(x), \sigma(x) I)$

decoder → $q(z)$

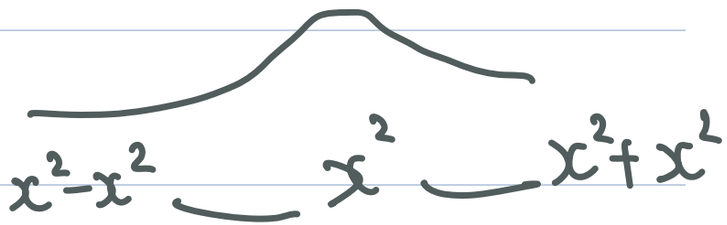
$x$ → $N(z \mid x^2, x^4)$ → $z$ sample

$z \sim N(z \mid x^2, x^4)$

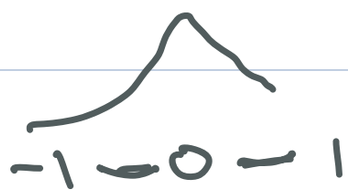
$\mu(x) = x^2$

$\sigma^2(x) = x^4$

$x$ 로 이동가능




$t \sim N(t \mid 0, 1)$




$z := x^2 + x^2 t$

$w_j^{t+1} = w_j^t - \eta \nabla_{w_j} \text{cost}$

# Reparametrization trick

If we sample $z$ from $\mathcal{N}(z \mid \mu(x), \sigma(x) I)$, then we lose the information about weights and bias of $\mu$, $\sigma$.

I.e. sampling operation is non-differentiable.

So we sample $\epsilon$ from $\mathcal{N}(\epsilon \mid 0, I)$ and let

$z := \mu(x) + \sigma(x) \# I$

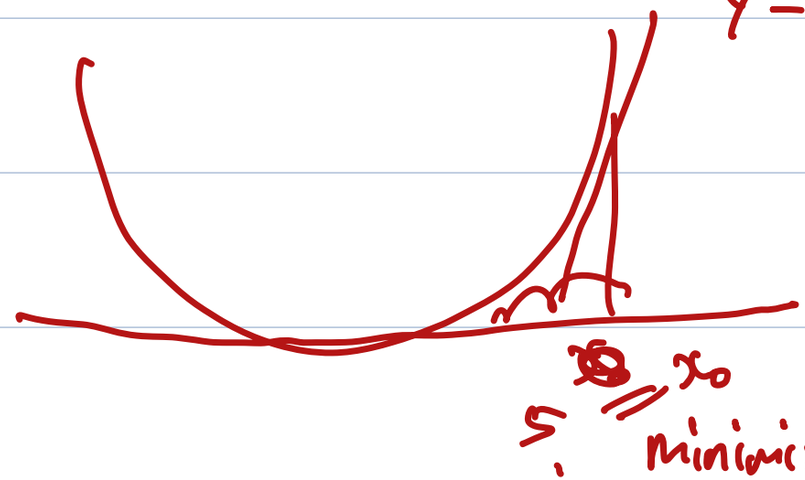
$$z \sim \mathcal{N}(z \mid x^2, x^4) \qquad \frac{z - x^2}{x^2} \sim \mathcal{N}(0, 1)$$

$$t \sim \mathcal{N}(t \mid 0, 1)$$

$$z := x^2 + t x^2$$

$$t = \frac{z - x^2}{x^2}$$

$y = f(x) = x^2$



Want $f(x)$의 최솟값 찾기

$x_0$

5

Minimize $x^2$

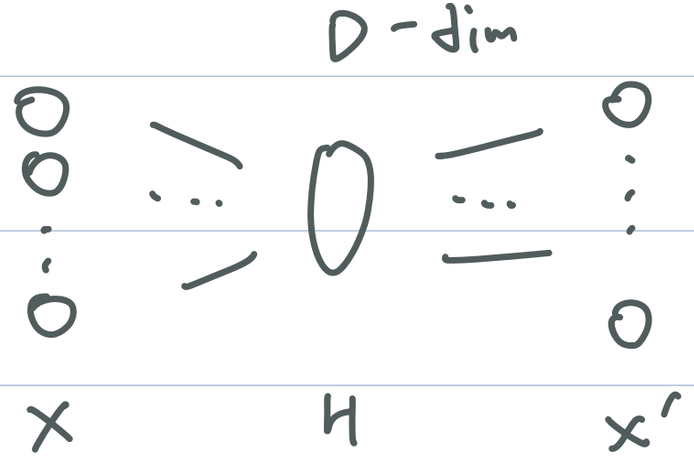
$$x^{t+1} = x^t - \eta \cdot f'(x^t)$$

$\eta$: 0.1

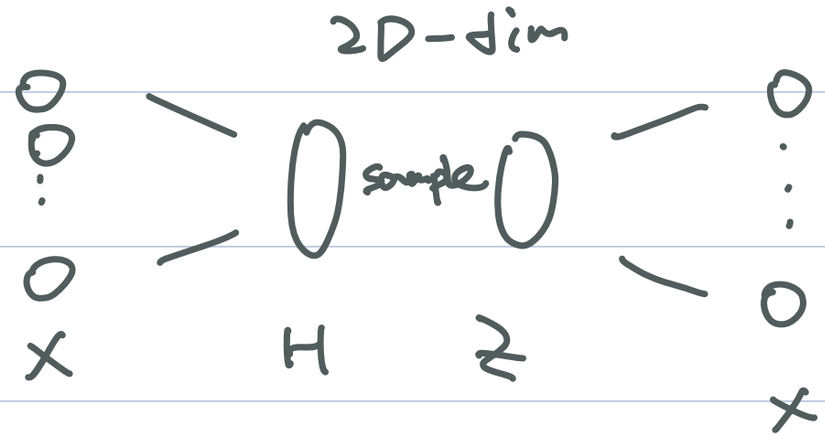
$2x$

$x_1 = 5 - 0.1 \cdot 10 = 4$

$x_2 = 4 - 0.1 \cdot 8 = 3.2$




$x \sim x'$




$x' \sim x$