# DL Seminar

한양대학교
AI Lab
유재창

한양대학교

Contents

Introduction

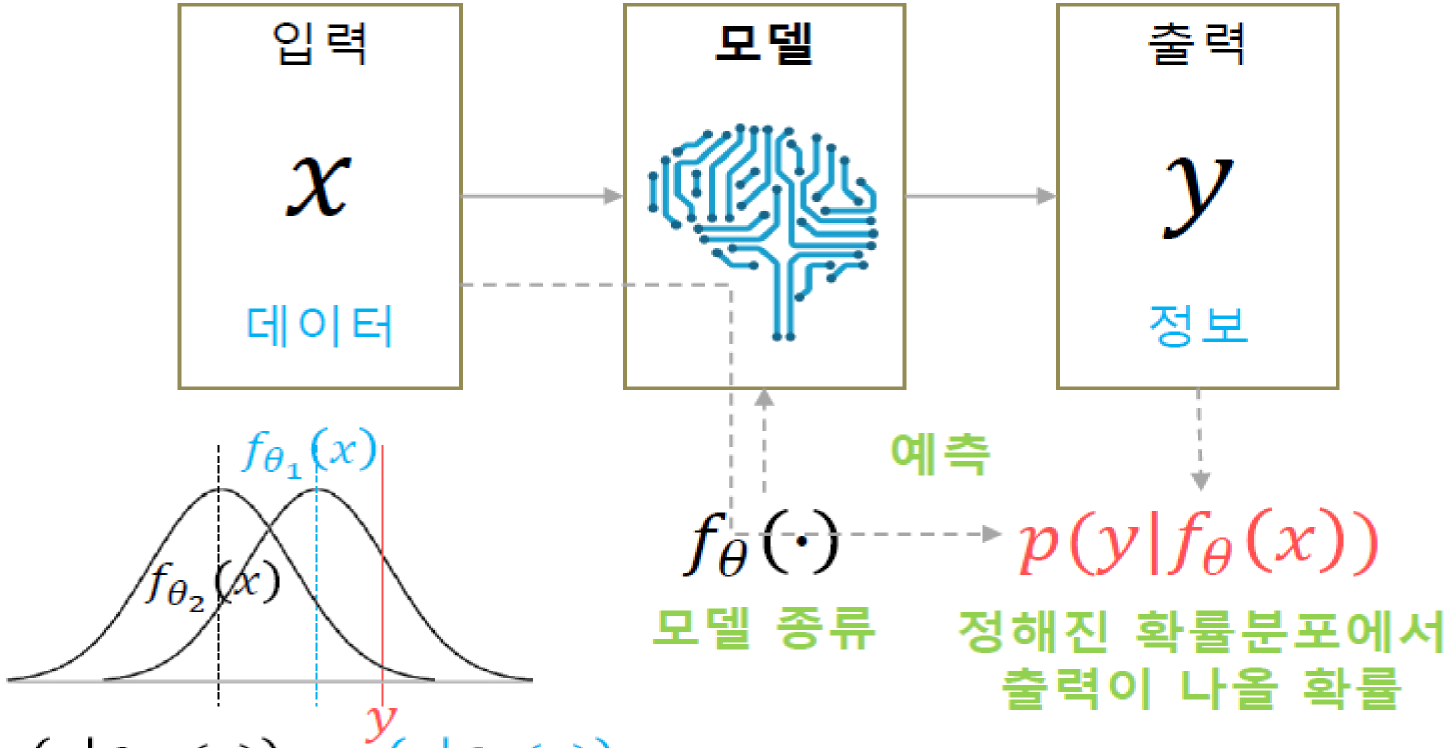
Experiment

Application

# Contents

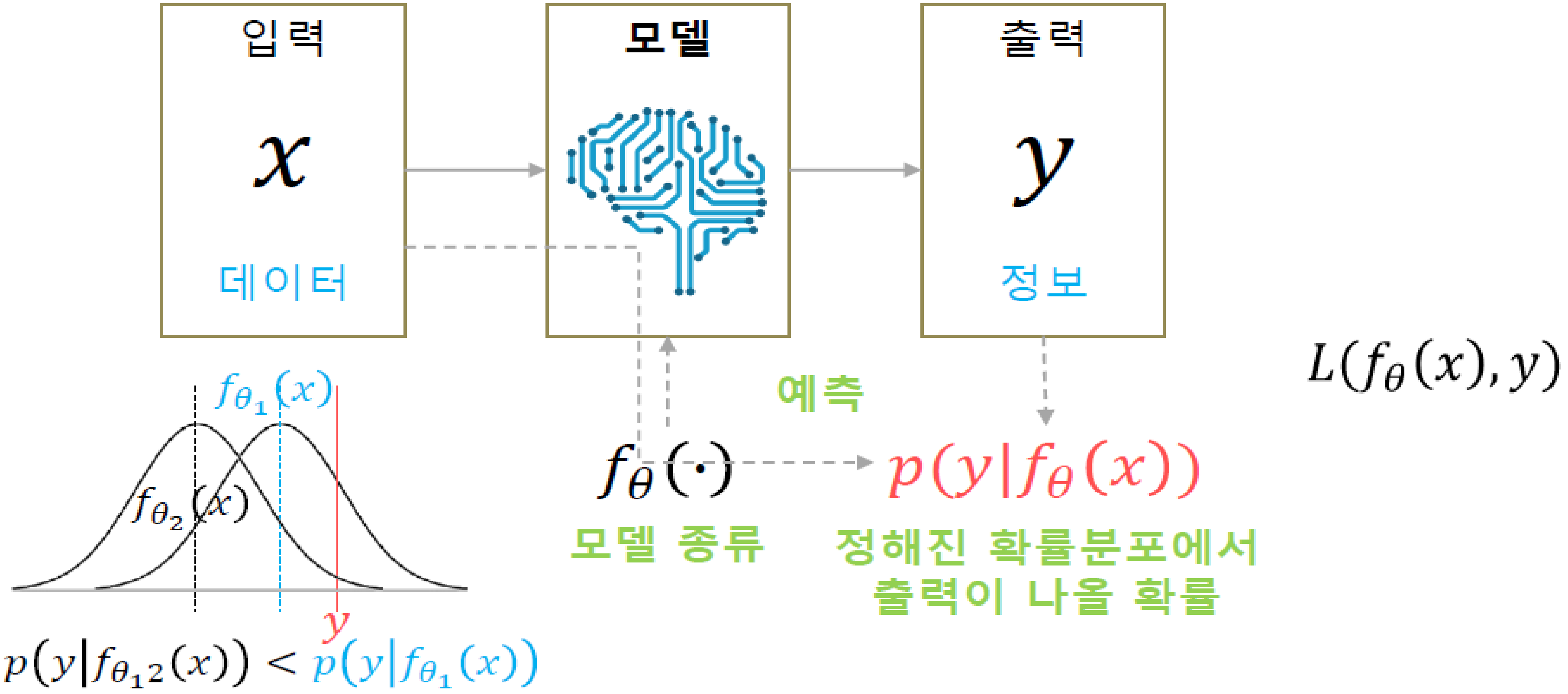
- Introduction
- Experiment
- Application

# Maximum Likelihood

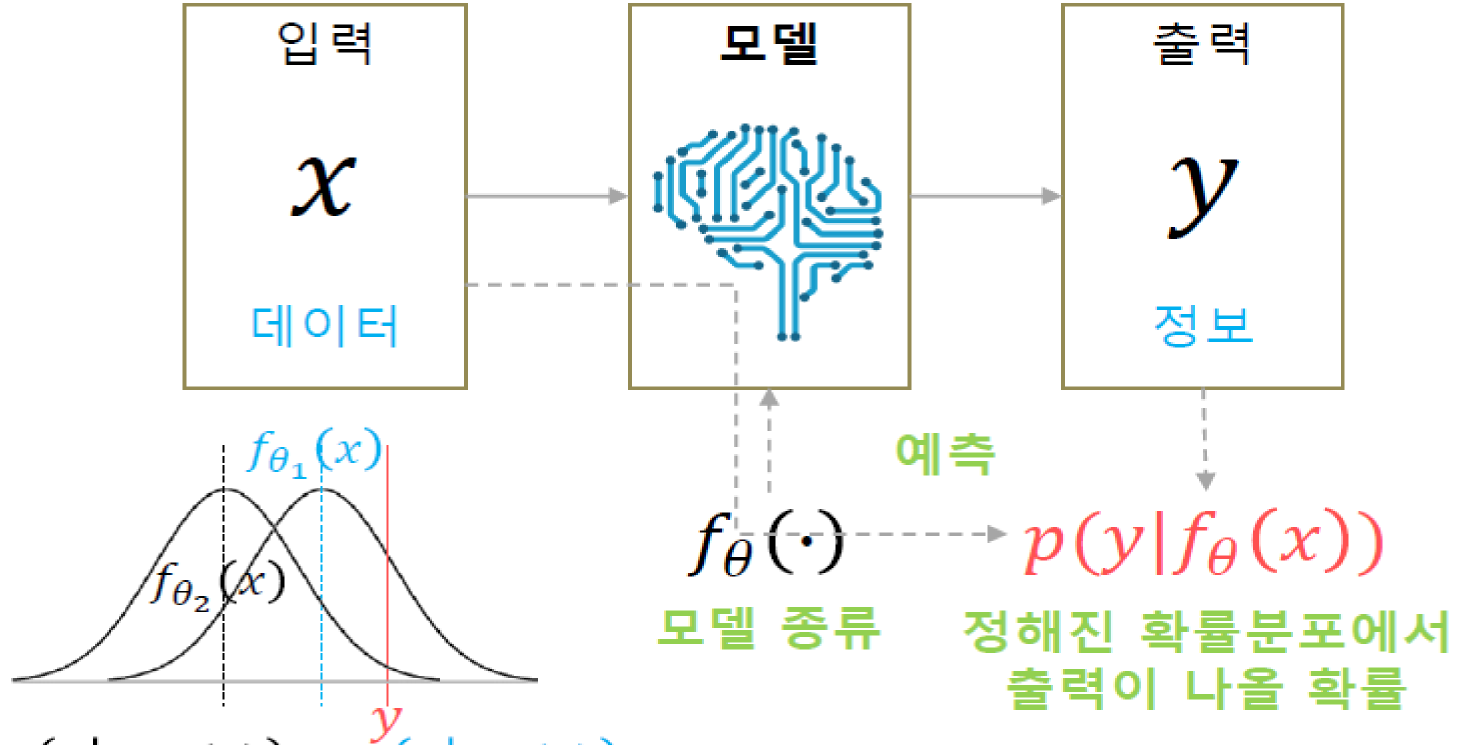



$$p(y|f_{\theta_12}(x)) < p(y|f_{\theta_1}(x))$$

# Maximum Likelihood



# Maximum Likelihood

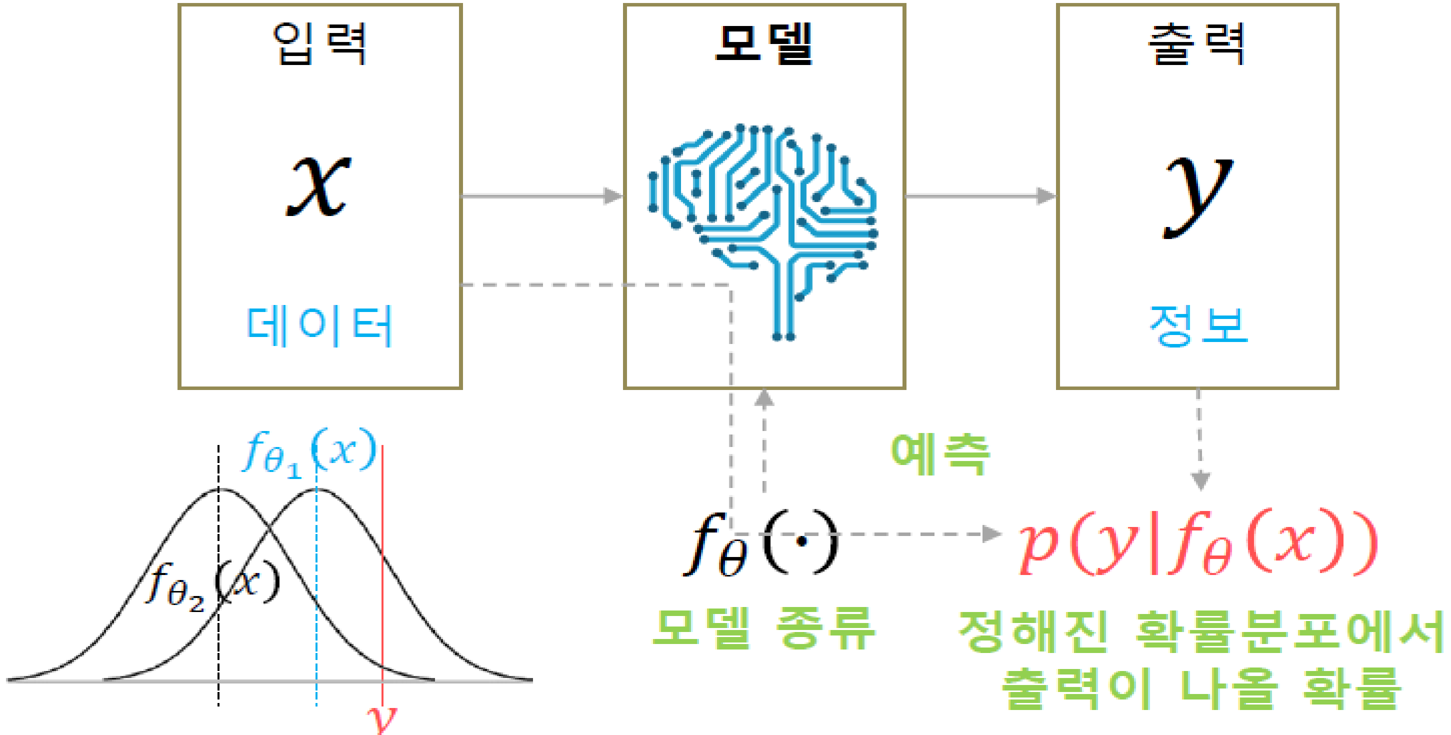



$$p(y|f_{\theta_1 2}(x)) < p(y|f_{\theta_1}(x))$$

Output : $f_\theta(x)$

Loss : $-\log(p(y|f_\theta(x)))$

# Maximum Likelihood




$$p(y|f_{\theta_1 2}(x)) < p(y|f_{\theta_1}(x))$$

$$L(f_\theta(x), y)$$

$$\theta^* = \underset{\theta}{\operatorname{argmin}} L(f_\theta(x), y) \qquad \theta^* = \underset{\theta}{\operatorname{argmin}}[-\log(p(y|f_\theta(x)))]$$

$$y_{new} = f_{\theta^*}(x_{new}) \qquad y_{new} \sim p(y|f_{\theta^*}(x_{new}))$$

# Maximum Likelihood

## i.i.d Condition on $p(y|f_\theta(x))$

### Assumption 1 : Independence

All of our data is independent of each other

$$p(y|f_\theta(x)) = \prod_i p_{D_i}(y|f_\theta(x_i))$$

### Assumption 2: Identical Distribution

Our data is identically distributed

$$p(y|f_\theta(x)) = \prod_i p(y|f_\theta(x_i))$$

# Maximum Likelihood



## i.i.d Condition on $p(y|f_\theta(x))$

### Assumption 1 : Independence

All of our data is independent of each other

$$p(y|f_\theta(x)) = \prod_i p_{D_i}(y|f_\theta(x_i))$$

### Assumption 2: Identical Distribution

Our data is identically distributed

$$p(y|f_\theta(x)) = \prod_i p(y|f_\theta(x_i))$$

$$-\log(p(y|f_\theta(x))) = -\sum_i \log(p(y_i|f_\theta(x_i)))$$

# Maximum Likelihood

$$-\log(p(y_i|f_\theta(x_i)))$$

## Gaussian distribution

$$f_\theta(x_i) = \mu_i, \sigma_i = 1$$

$$p(y_i|\mu_i,\sigma_i) = \frac{1}{\sqrt{2\pi}\sigma_i}\exp\left(-\frac{(y_i-\mu_i)^2}{2\sigma_i^2}\right)$$

$$\log(p(y_i|\mu_i,\sigma_i)) = \log\frac{1}{\sqrt{2\pi}\sigma_i} - \frac{(y_i-\mu_i)^2}{2\sigma_i^2}$$

$$-\log(p(y_i|\mu_i)) = -\log\frac{1}{\sqrt{2\pi}} + \frac{(y_i-\mu_i)^2}{2}$$

$$-\log(p(y_i|\mu_i)) \propto \frac{(y_i-\mu_i)^2}{2} = \frac{(y_i-f_\theta(x_i))^2}{2}$$

Mean Squared Error

## Bernoulli distribution

$$f_\theta(x_i) = p_i$$

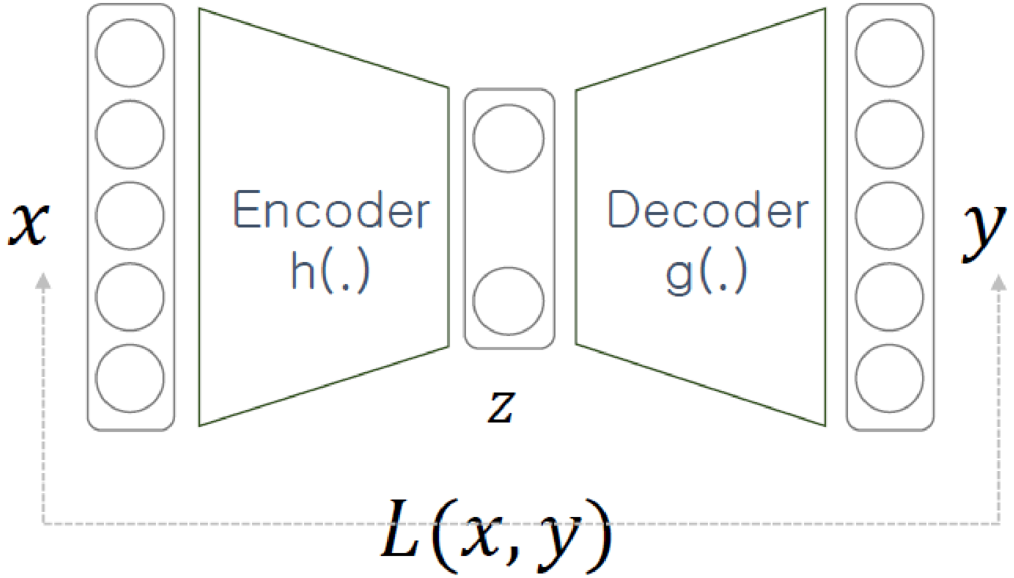
$$p(y_i|p_i) = p_i^{y_i}(1-p_i)^{1-y_i}$$

$$\log(p(y_i|p_i)) = y_i\log p_i + (1-y_i)\log(1-p_i)$$

$$-\log(p(y_i|p_i)) = -[y_i\log p_i + (1-y_i)\log(1-p_i)]$$

Cross-entropy

# AutoEncoder




$x, y \in \mathbb{R}^d$

입출력이 동일한 네트워크

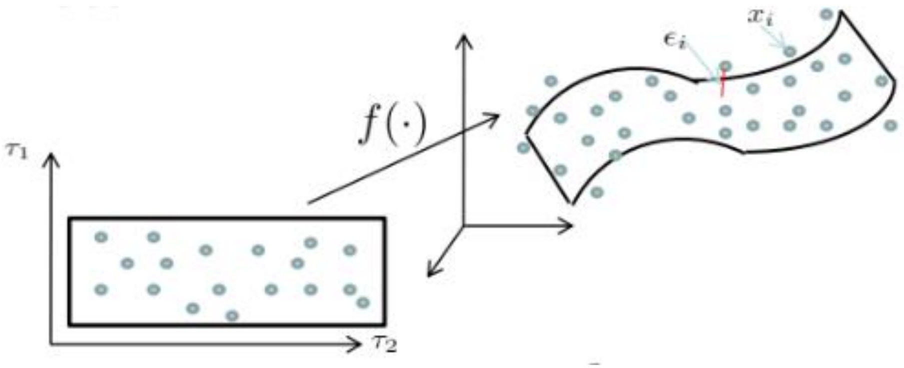
Unsupervised Learning을 Supervised Learning으로 바꾸어 해결

Decoder가 최소한 학습 데이터는 생성해 낼 수 있게 된다.
->생성된 데이터가 학습 데이터 좀 닮았다.

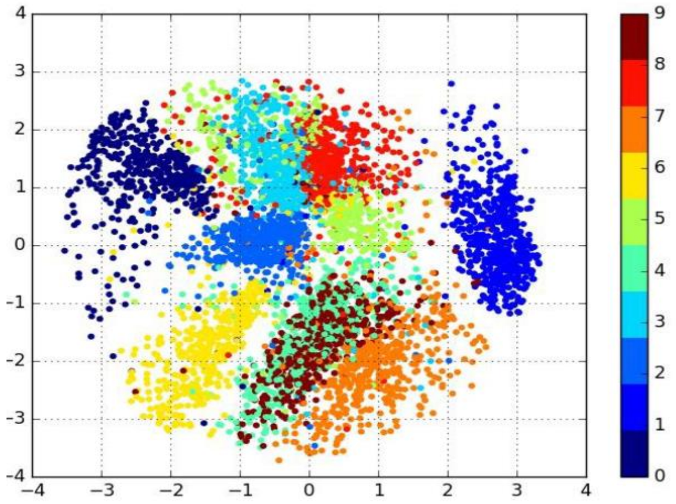
Encoder가 최소한 학습 데이터는 잘 latent vector로 표현 할 수 있게 된다.
->데이터의 추상화를 위해 많이 사용됨.

# Manifold Learning





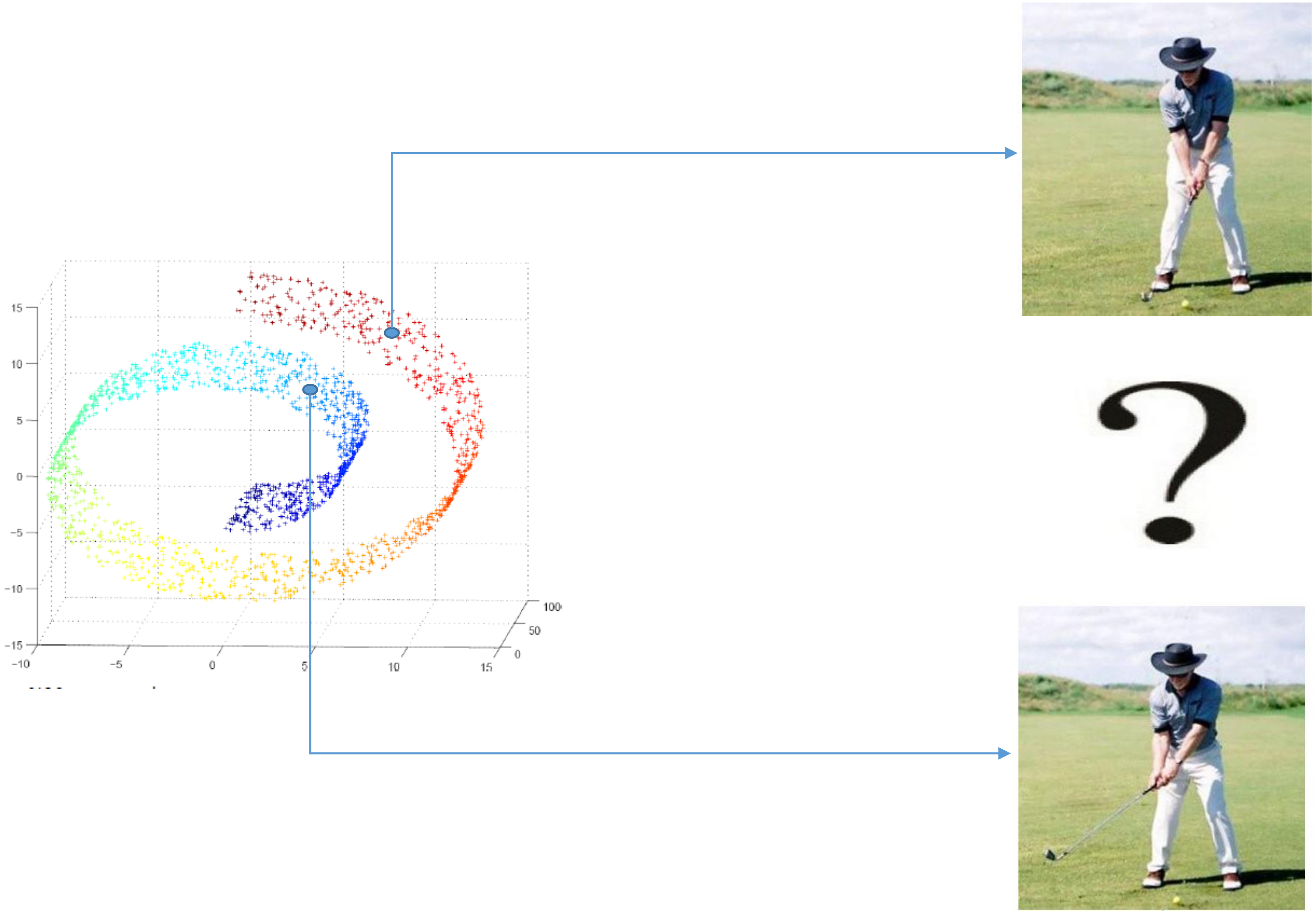

- 고차원의 데이터 공간에 데이터들이 존재할 때, 이들을 잘 포함할 수 있는 저차원의 서브스페이스인 Manifold가 존재한다.



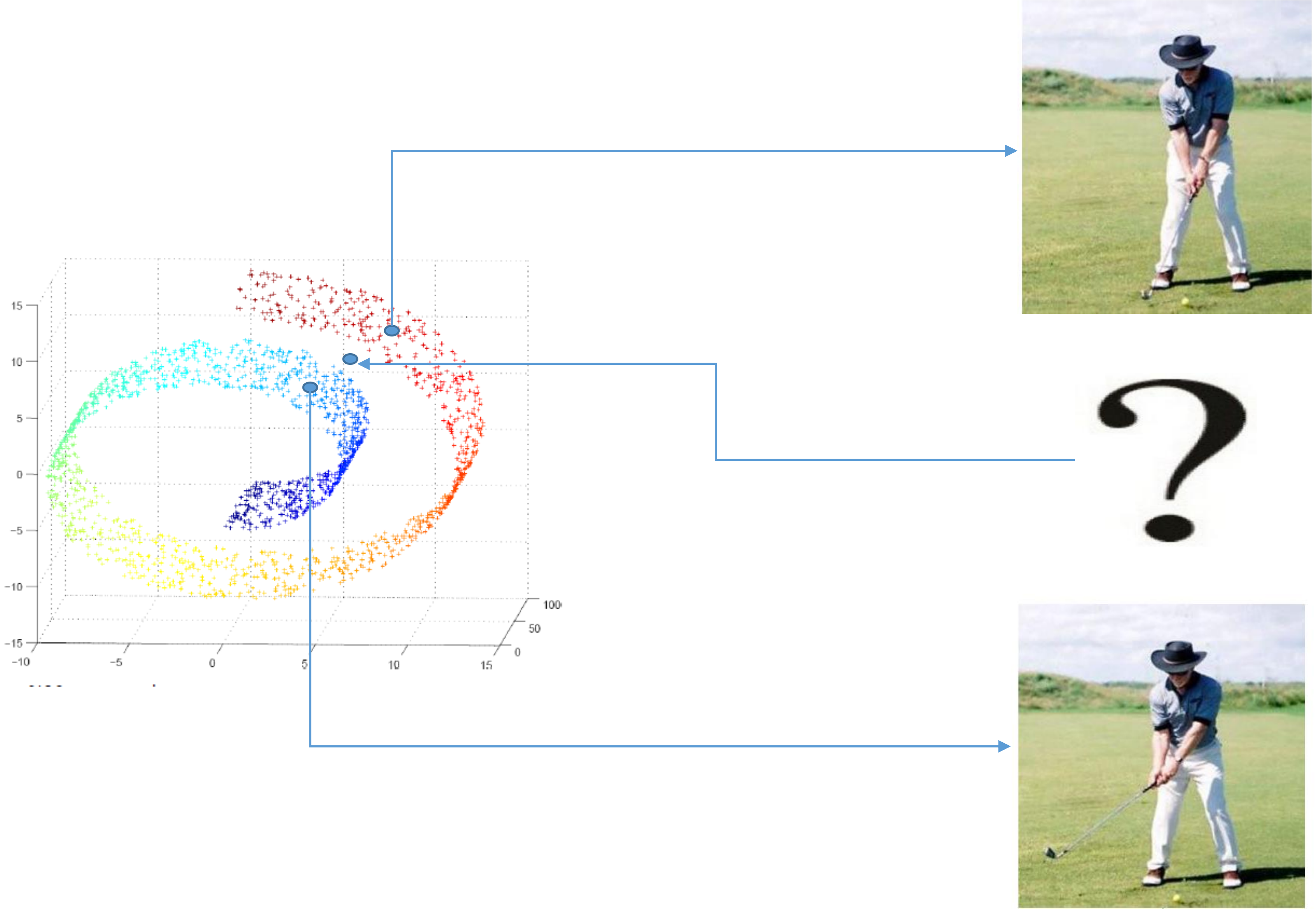
MNIST Data → 2D manifold

- 목적
- 데이터 압축
- 데이터 시각화
- 차원의 저주의 문제
- 중요한 Feature

# Manifold Learning

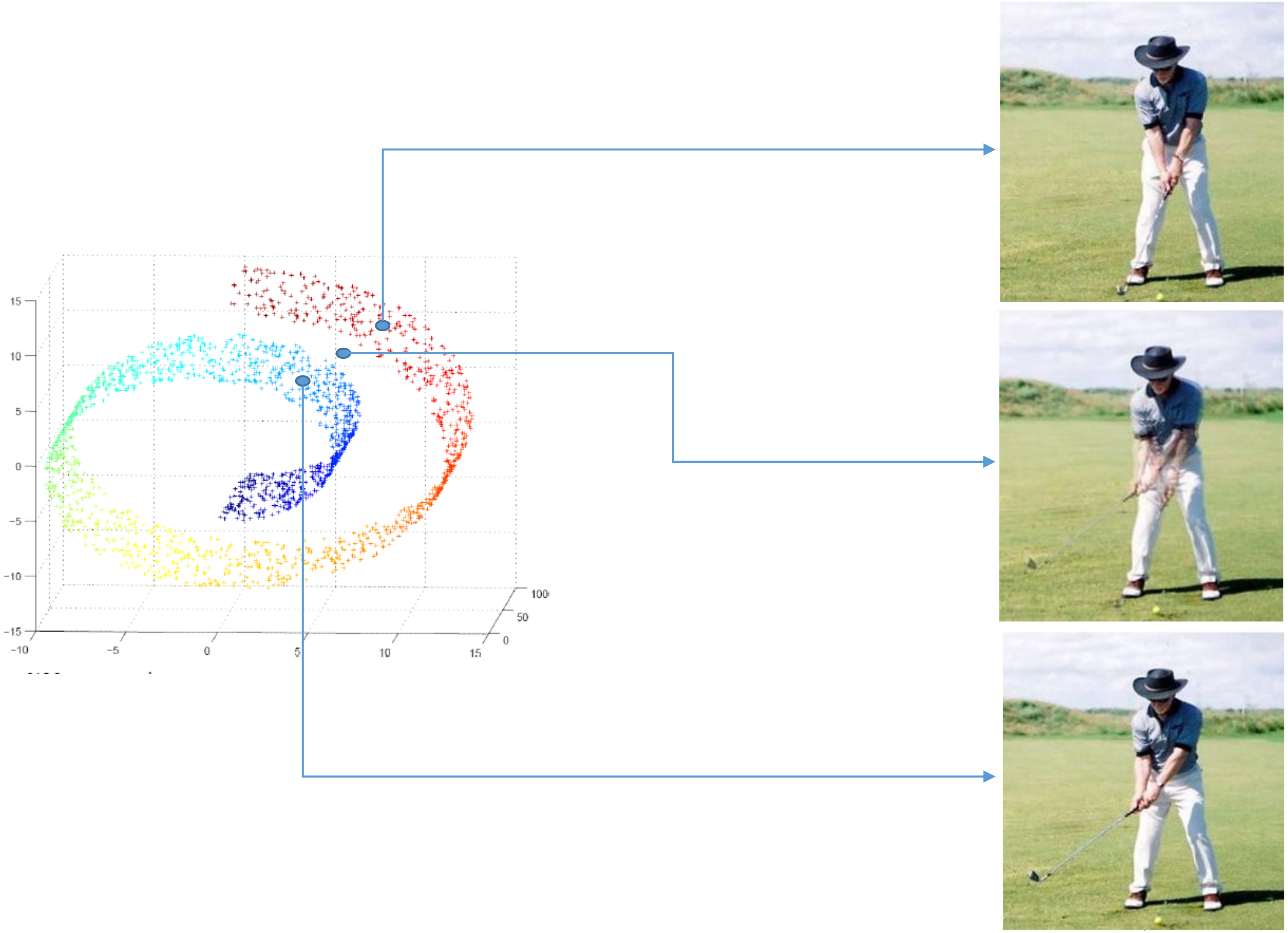


# Manifold Learning



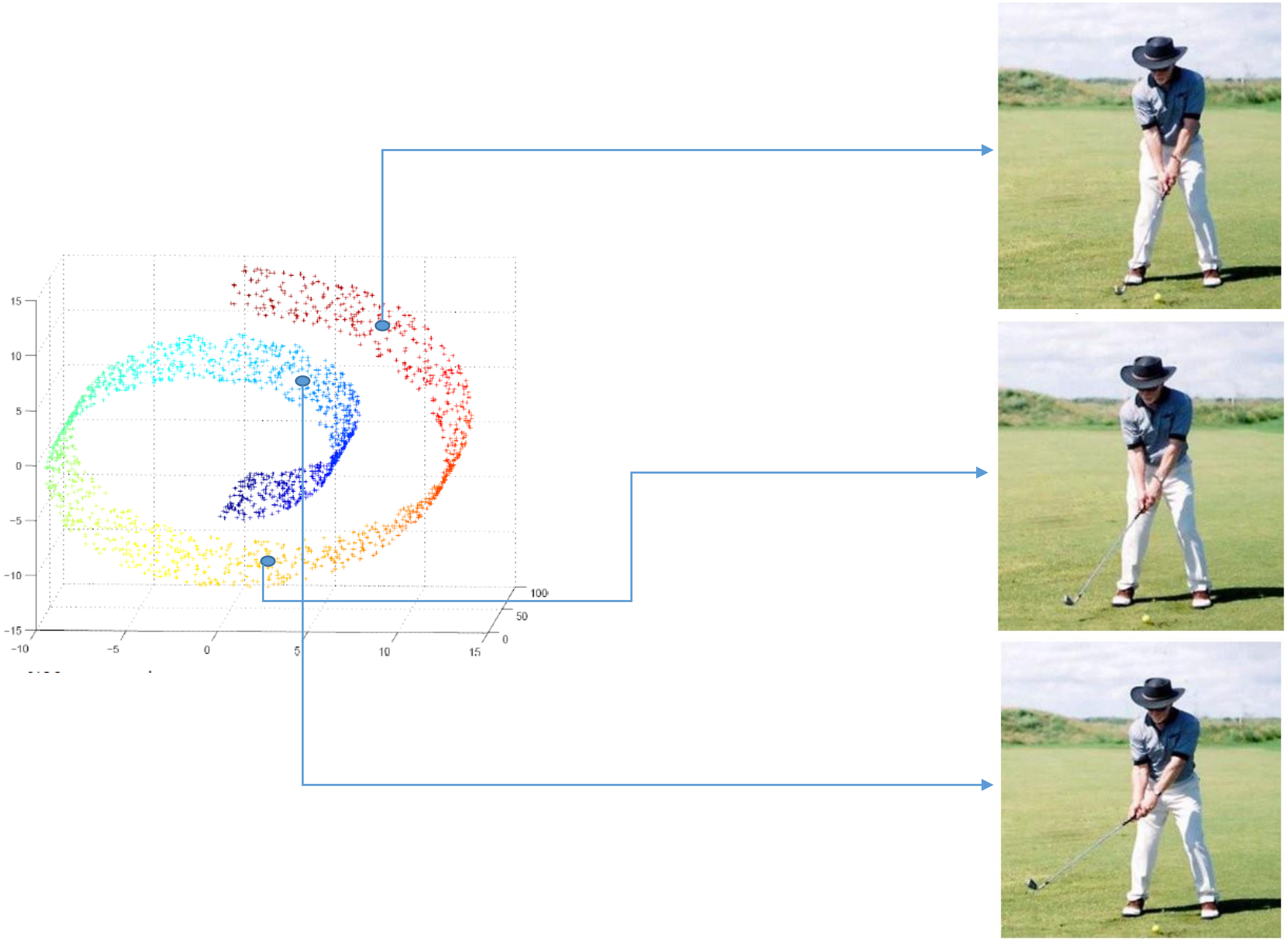
# Manifold Learning





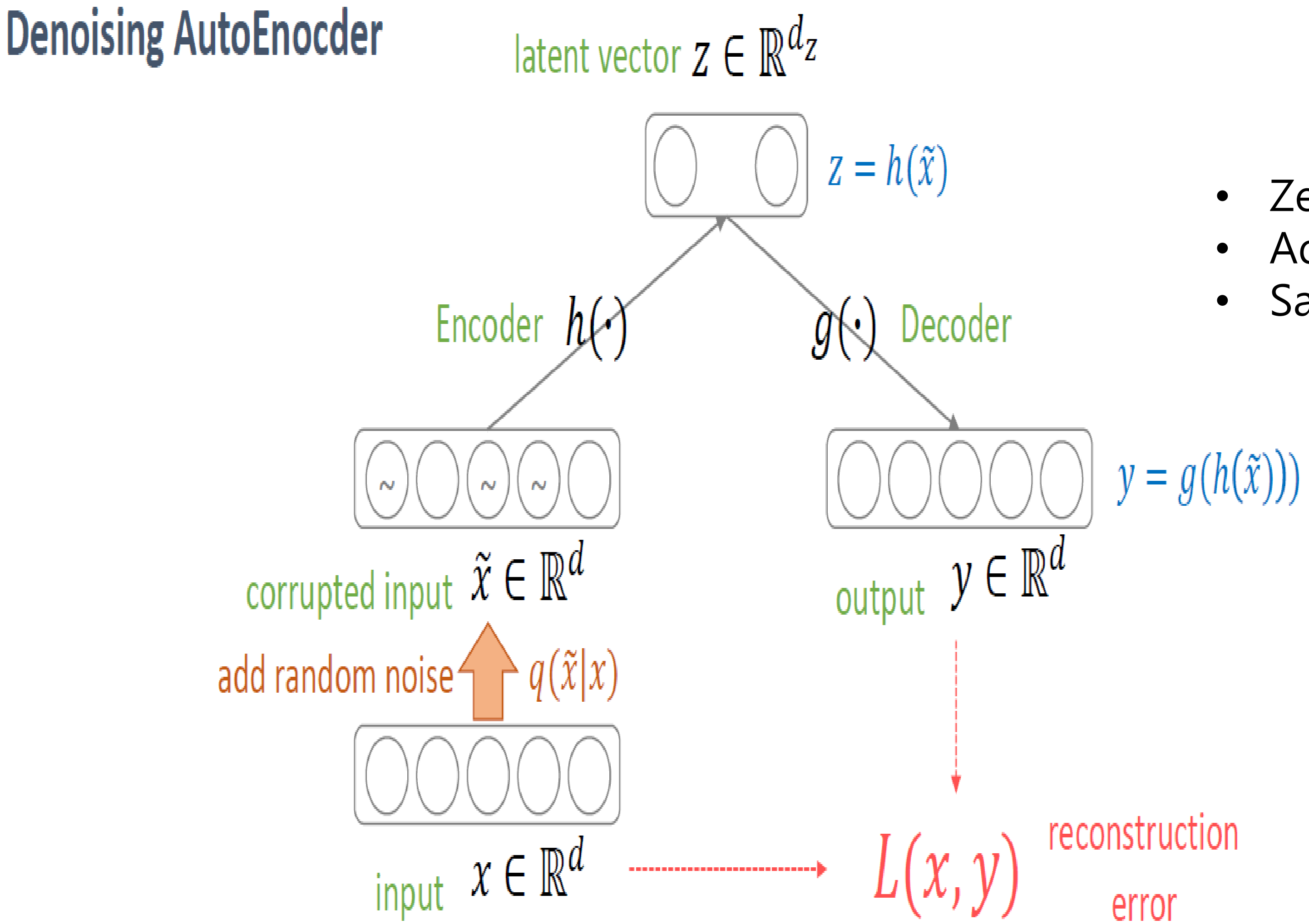
# Manifold Learning





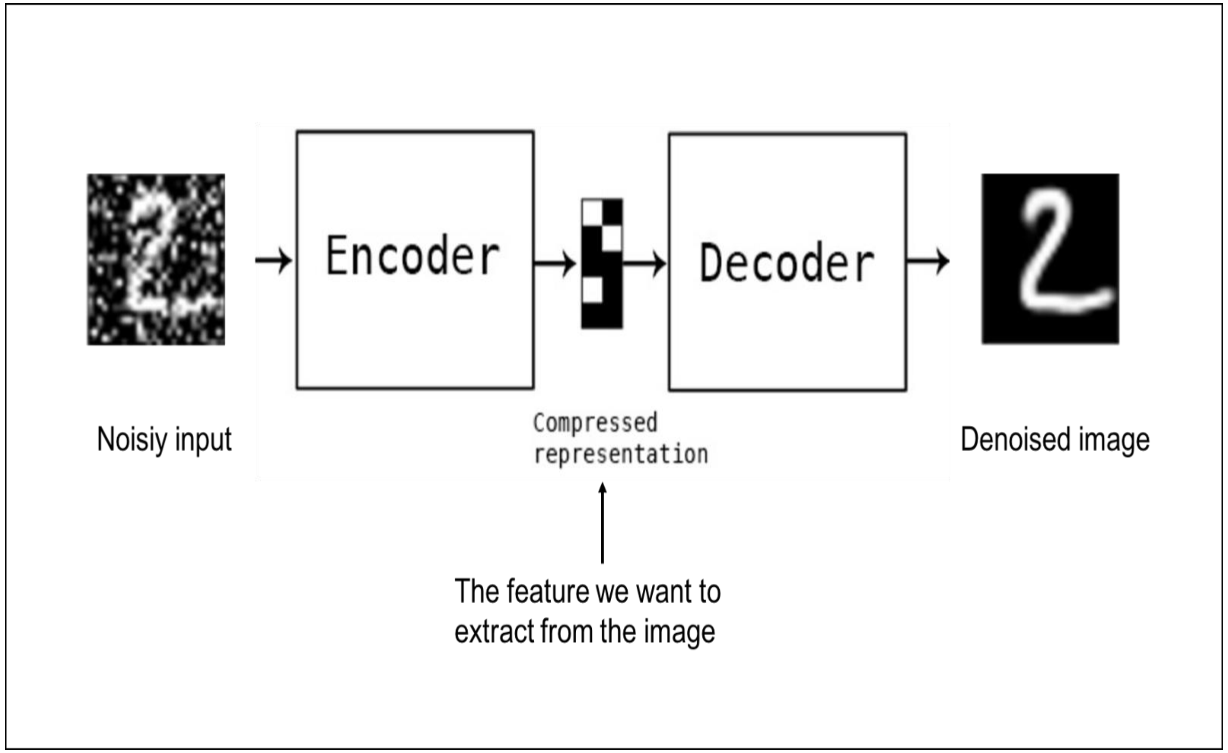
# Denoising AutoEncoder






- Zeroing at random point
- Adding Gaussian noise
- Salt and pepper noise

# Denoising AutoEncoder




$$L_{DAE} = \sum_{x \in D} E_{q(\tilde{x}|x)} \left[ L(x, g(h(\tilde{x}))) \right] \approx \sum_{x \in D} \frac{1}{L} \sum_{i=1}^{L} L(x, g(h(\tilde{x}_i)))$$

L개 샘플에 대한 평균으로 대체

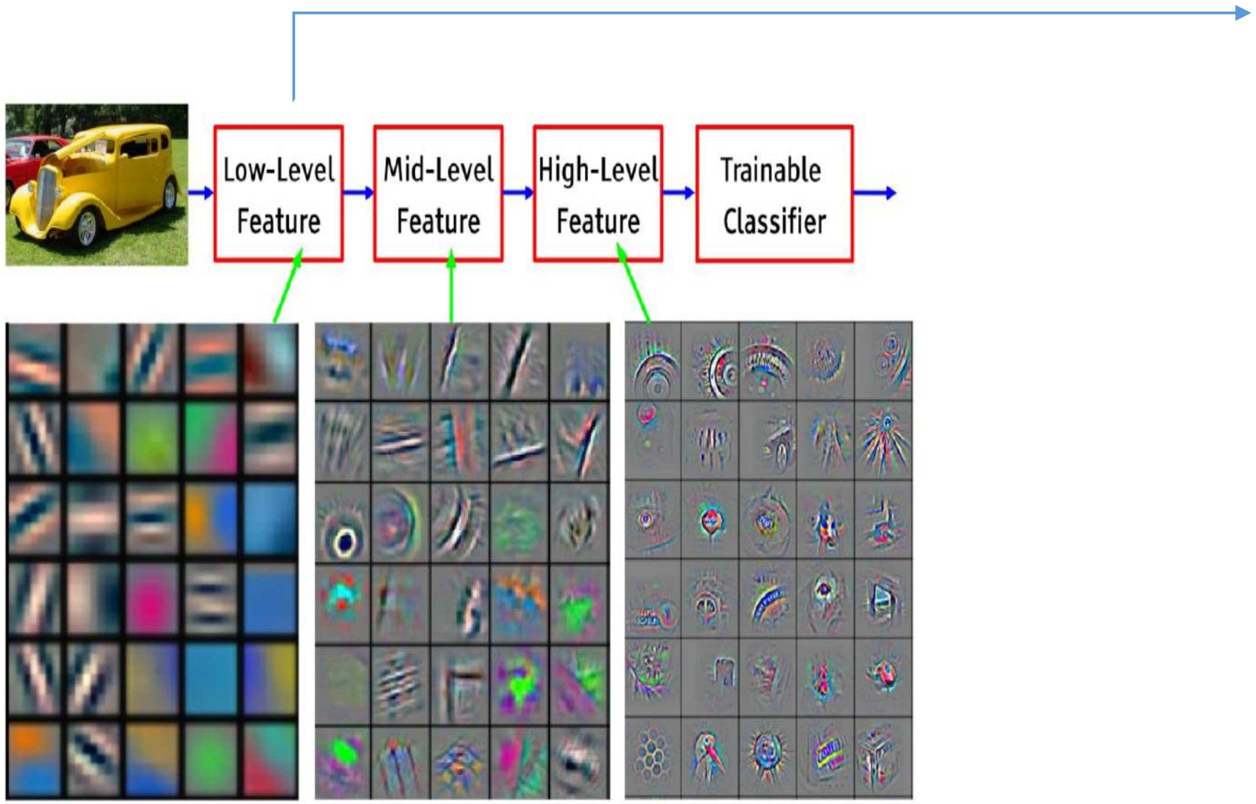
# Denoising AutoEncoder

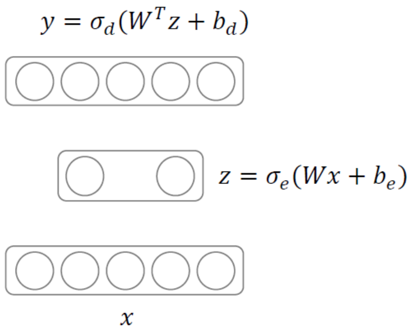

Contents

Introduction

Experiment

Application

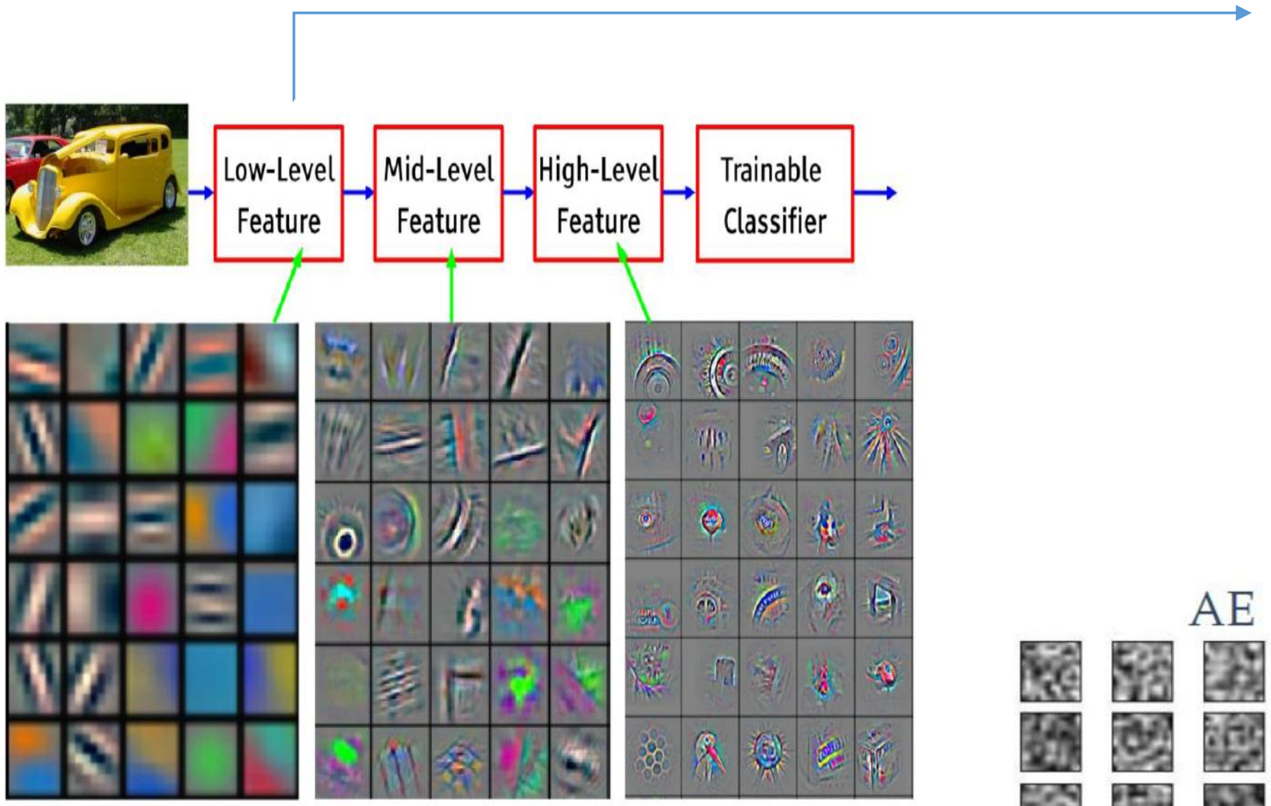



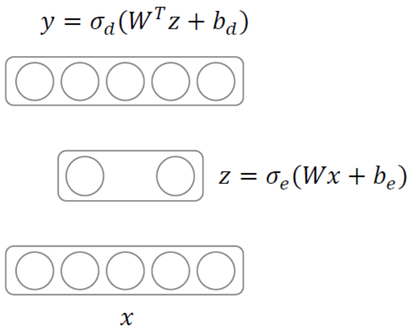




AutoEncoder
with 1hidden layer

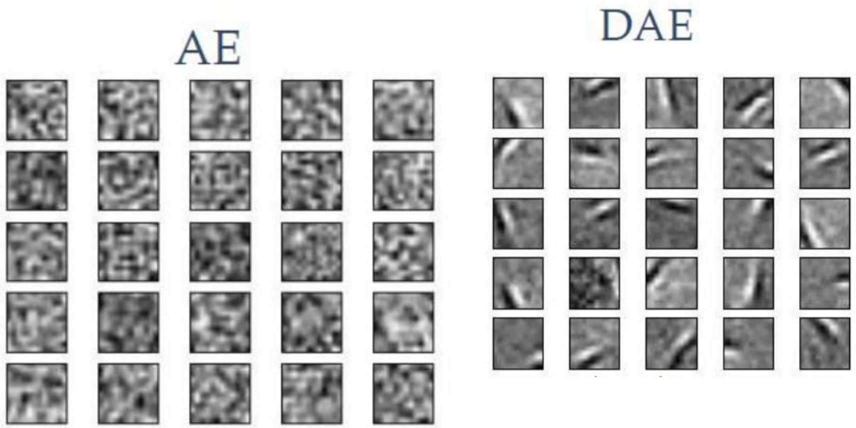
# Denoising AutoEncoder



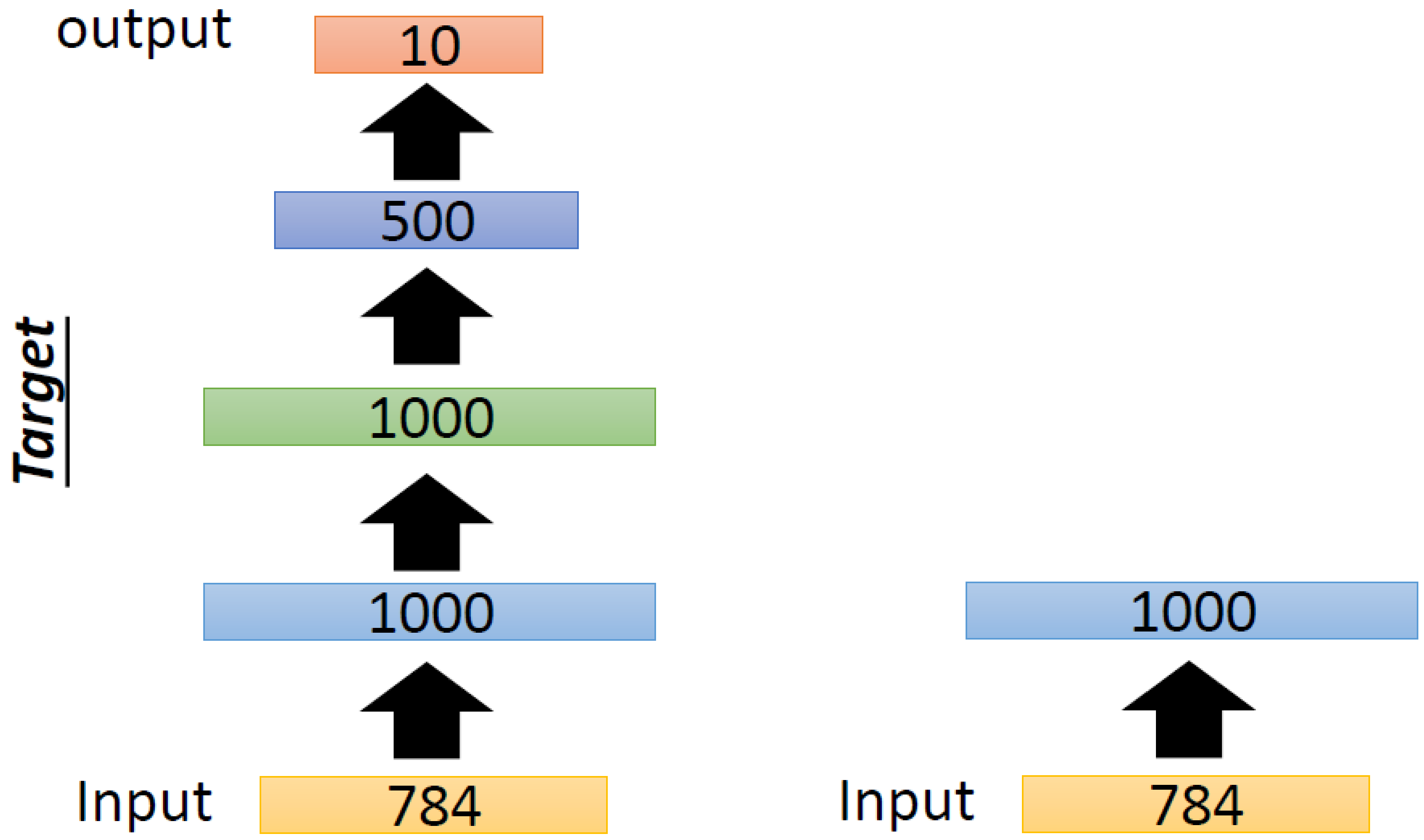


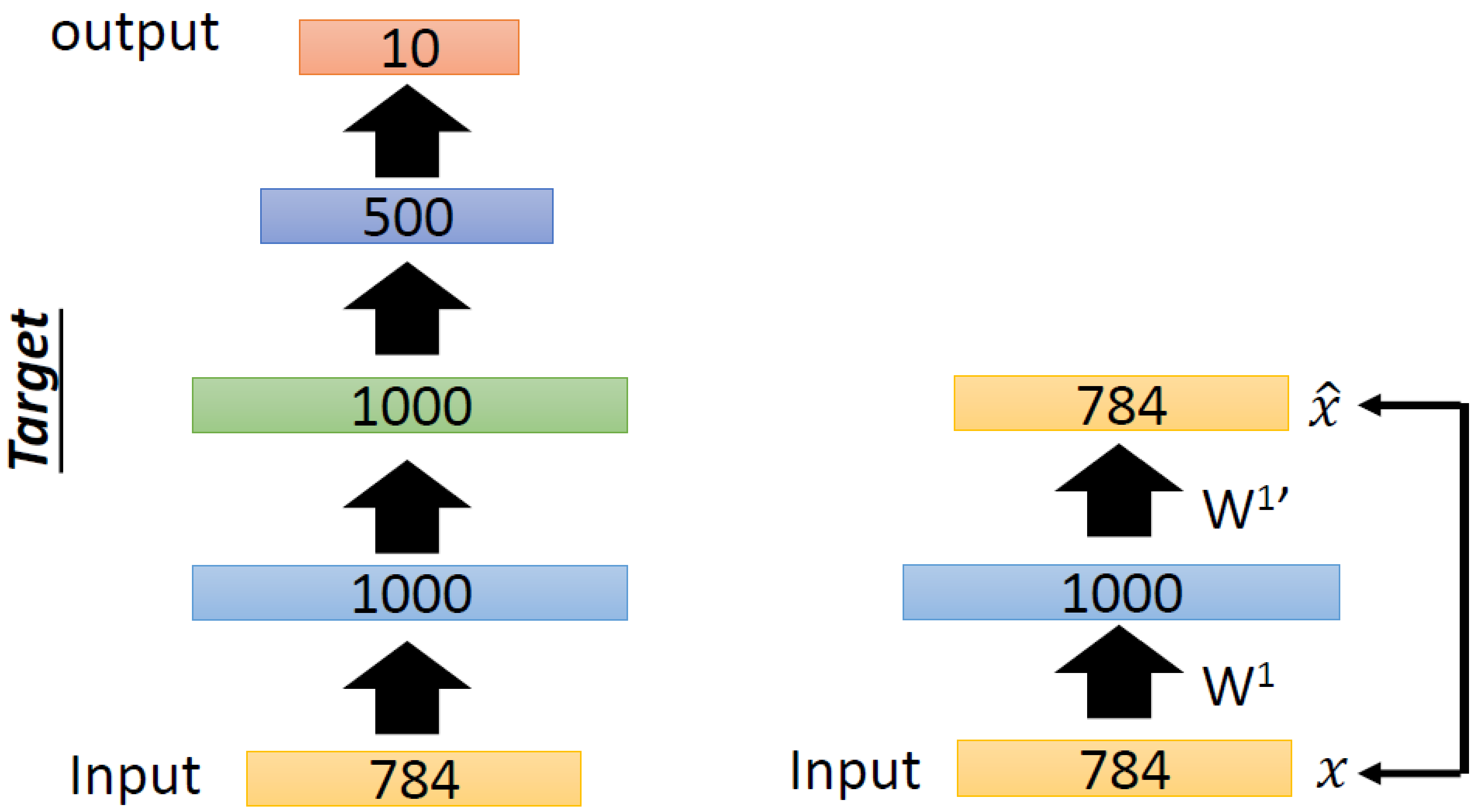

AutoEncoder
with 1hidden layer



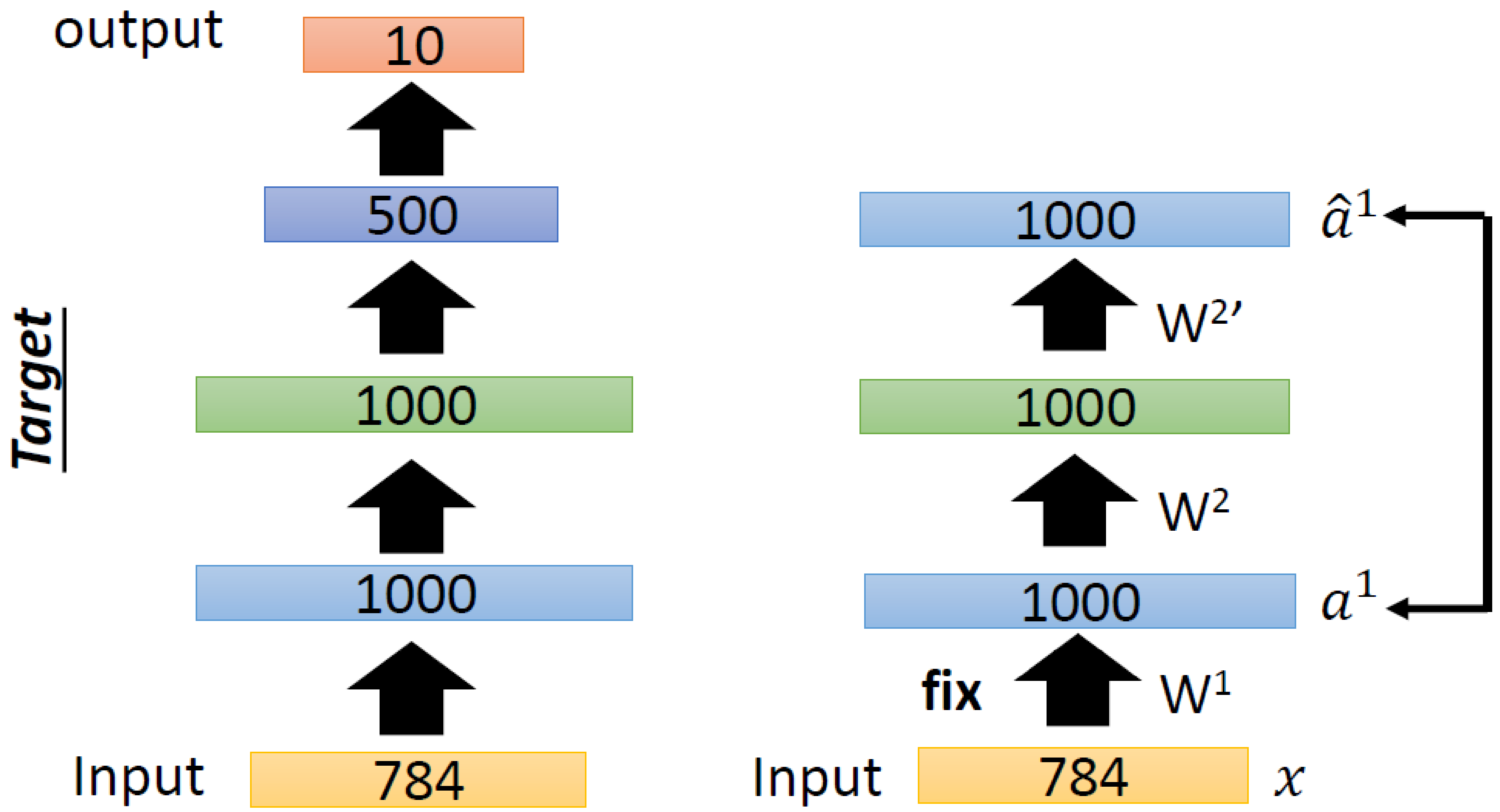
# Stacked AutoEncoder



# Stacked AutoEncoder

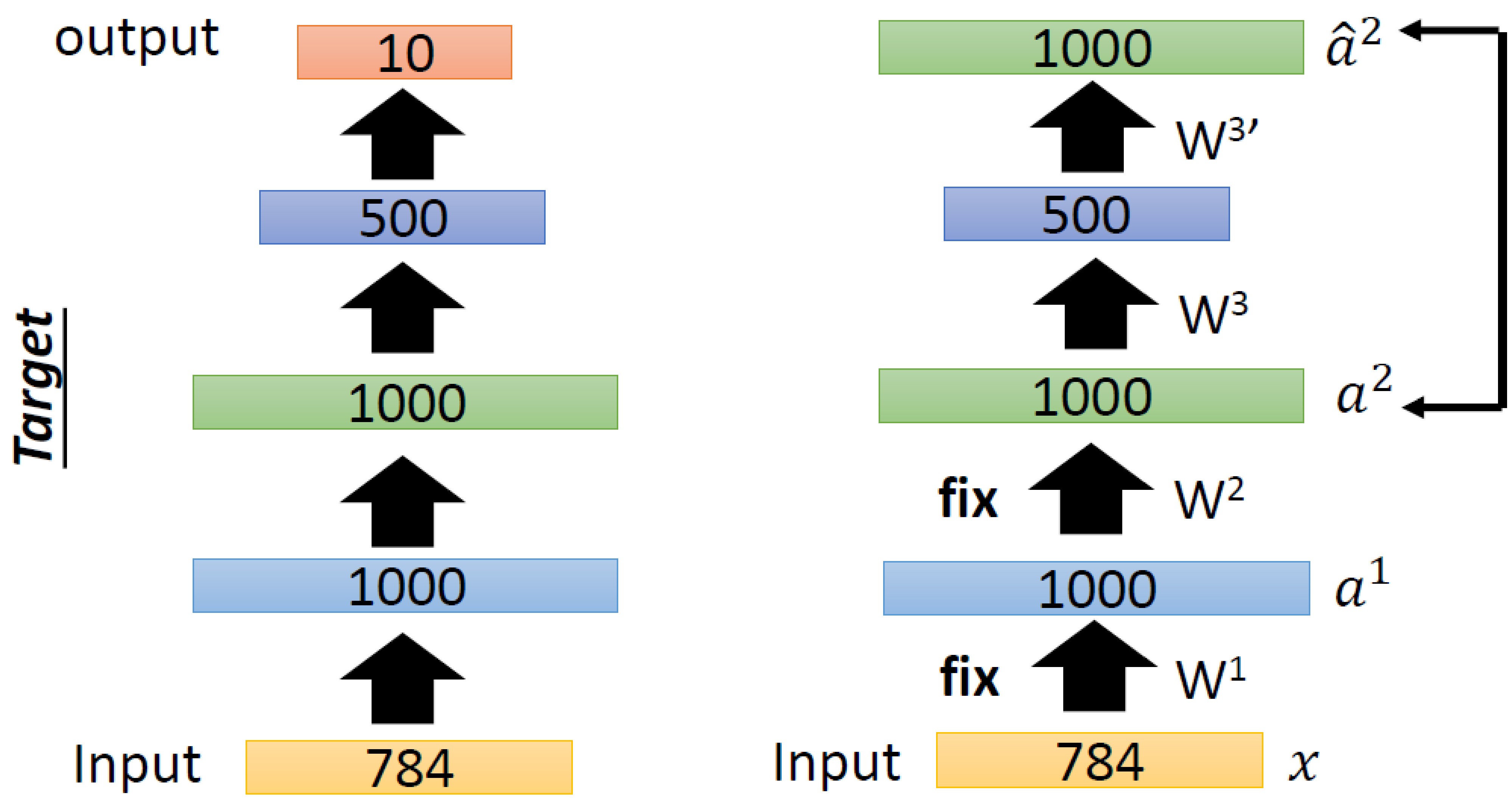


# Stacked AutoEncoder

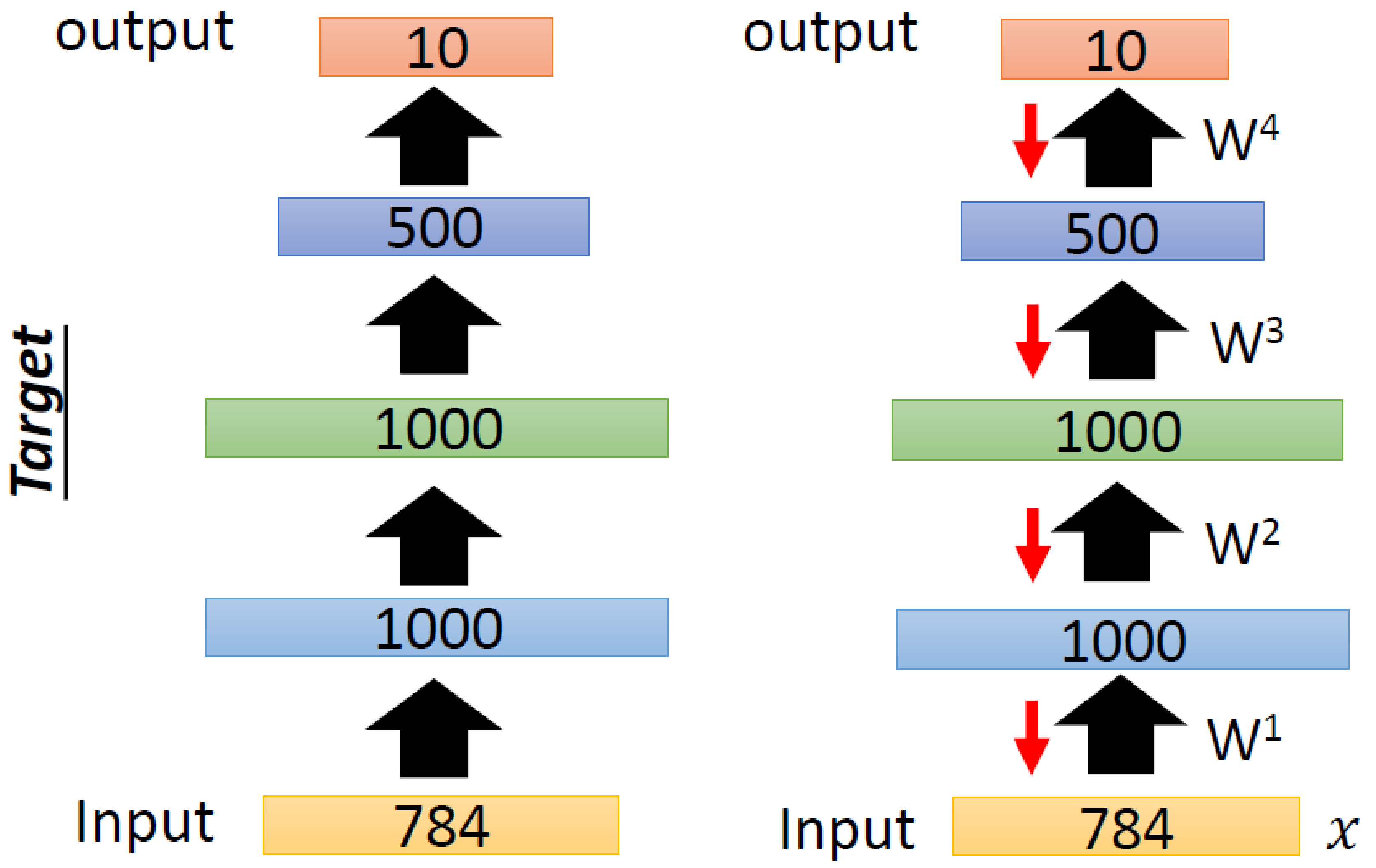


# Stacked AutoEncoder





# Stacked AutoEncoder



# SCAE & CAE

$$L_{SCAE} = \sum_{x \in D} \underbrace{L(x, g(h(x)))}_{\text{Reconstruction Error}} + \lambda \underbrace{E_{q(\tilde{x}|x)}\left[\|h(x) - h(\tilde{x})\|^2\right]}_{\text{Stochastic Regularization}}$$

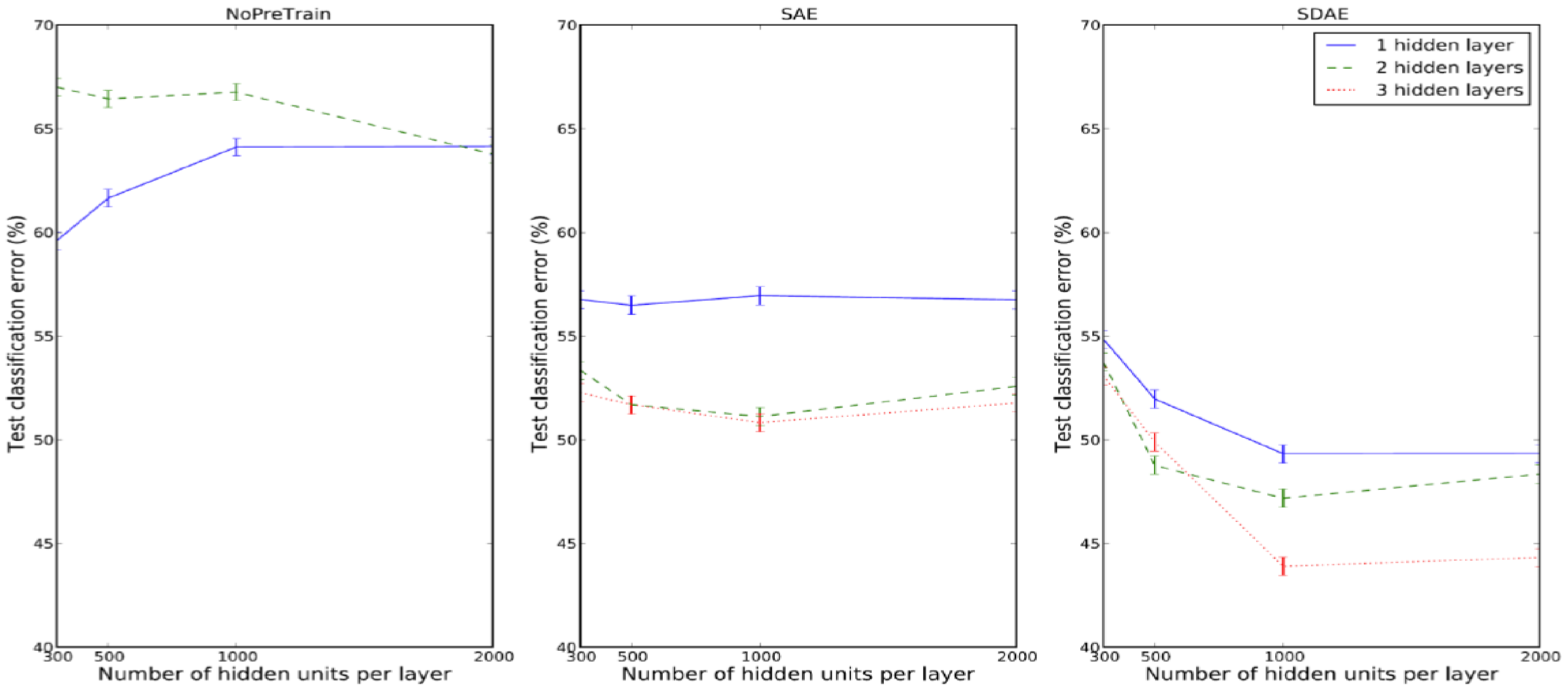
SCAE stochastic regularization term : $E_{q(\tilde{x}|x)}\left[\|h(x) - h(\tilde{x})\|^2\right]$

For small additive noise, $\tilde{x}|x = x + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma^2 I)$

Taylor series expansion yields, $h(\tilde{x}) = h(x + \epsilon) = h(x) + \frac{\partial h}{\partial x}\epsilon + \cdots$

$$\underbrace{E_{q(\tilde{x}|x)}\left[\|h(x) - h(\tilde{x})\|^2\right]}_{\text{Stochastic Regularization (SCAE)}} \approx \underbrace{\left\|\frac{\partial h}{\partial x}(x)\right\|_F^2}_{\text{Analytic Regularization (CAE)}}$$

# Result



NoPreTrain
SAE
SDAE
1 hidden layer
2 hidden layers
3 hidden layers
Test classification error (%)
Number of hidden units per layer
70
65
60
55
50
45
40
300
500
1000
2000

# Result

| | Model | Test error | Average $\|J_f(x)\|_F$ | SAT |
|---|---|---|---|---|
| MNIST | CAE | **1.14** | 0.73 $10^{-4}$ | 86.36% |
| | DAE-g | **1.18** | 0.86 $10^{-4}$ | 17.77% |
| | RBM-binary | 1.30 | 2.50 $10^{-4}$ | 78.59% |
| | DAE-b | 1.57 | 7.87 $10^{-4}$ | 68.19% |
| | AE+wd | 1.68 | 5.00 $10^{-4}$ | 12.97% |
| | AE | 1.78 | 17.5 $10^{-4}$ | 49.90% |
| CIFAR-bw | CAE | **47.86** | 2.40 $10^{-5}$ | 85,65% |
| | DAE-b | **49.03** | 4.85 $10^{-5}$ | 80,66% |
| | DAE-g | 54.81 | 4.94 $10^{-5}$ | 19,90% |
| | AE+wd | 55.03 | 34.9 $10^{-5}$ | 23,04% |
| | AE | 55.47 | 44.9 $10^{-5}$ | 22,57% |

- DAE-g : DAE with gaussian noise
- DAE-b : DAE with binary masking noise
- CIFAR-bw : gray scale version
- Training/Validation/test : 10k/2k/50k
- SAT : average fraction of saturated units per sample
- 1-hidden layer with 1000 units

한양대학교

Contents

Introduction

Experiment

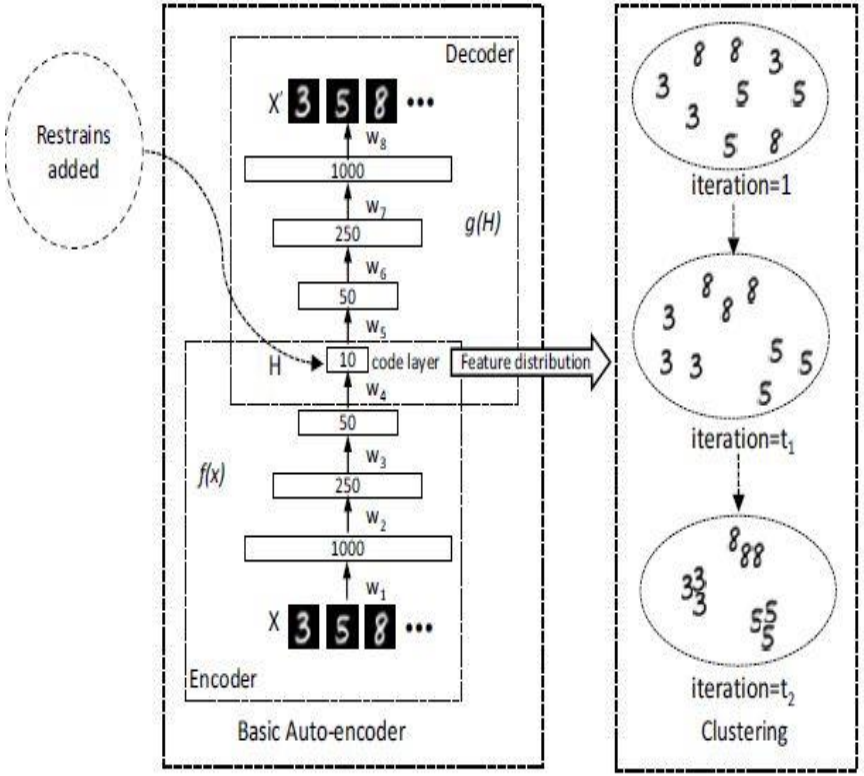
Application

# Result

| Data Set | $SVM_{rbf}$ | SAE-3 | RBM-3 | DAE-b-3 | CAE-1 | CAE-2 |
|---|---|---|---|---|---|---|
| *basic* | 3.03±0.15 | 3.46±0.16 | 3.11±0.15 | 2.84±0.15 | 2.83±0.15 | **2.48**±0.14 |
| *rot* | 11.11±0.28 | 10.30±0.27 | 10.30±0.27 | **9.53**±0.26 | 11.59±0.28 | **9.66**±0.26 |
| *bg-rand* | 14.58±0.31 | 11.28±0.28 | **6.73**±0.22 | 10.30±0.27 | 13.57±0.30 | 10.90 ±0.27 |
| *bg-img* | 22.61±0.379 | 23.00±0.37 | 16.31±0.32 | 16.68±0.33 | 16.70±0.33 | **15.50**±0.32 |
| *bg-img-rot* | 55.18±0.44 | 51.93±0.44 | 47.39±0.44 | **43.76**±0.43 | 48.10±0.44 | 45.23±0.44 |
| *rect* | 2.15±0.13 | 2.41±0.13 | 2.60±0.14 | 1.99±0.12 | 1.48±0.10 | **1.21**±0.10 |
| *rect-img* | 24.04±0.37 | 24.05±0.37 | 22.50±0.37 | **21.59**±0.36 | **21.86**±0.36 | **21.54**±0.36 |

- basic: smaller subset of MNIST
- rot: digits with added random rotation
- bg-rand: digits with random noise background
- bg-img: digits with random image background
- bg-img-rot: digits with rotation and image background
- rect: discriminate between tall and wide rectangles (white on black)
- rect-img: discriminate between tall and wide rectangular image on a different background image

# Application




- 클러스터링 알고리즘
  - Step 0: 각 f(x)를 임의의 클러스터에 할당
    각 클러스터의 c(x) 계산
  - Step 1: 목적 함수 계산
    오토 인코더 학습
  - Step 2: 변경된 f(x)를 이용한 중심점 c(x) 계산
  - Step 3: 각 f(x)를 새로 구한 c(x)에 할당
  - Step 1~3 반복
    (step 1이 없으면 그냥 k-mean과 동일)

$$\min_{W,b} \frac{1}{N} \sum_{i=1}^{N} \|x_i - z_i\|^2 + \lambda \sum_{i=1}^{N} \|f(x_i) - c(x_i)\|^2$$

$$c(x) = \underset{c_i}{argmin} \|f(x) - c_i\|^2$$

# 끝

AI Lab