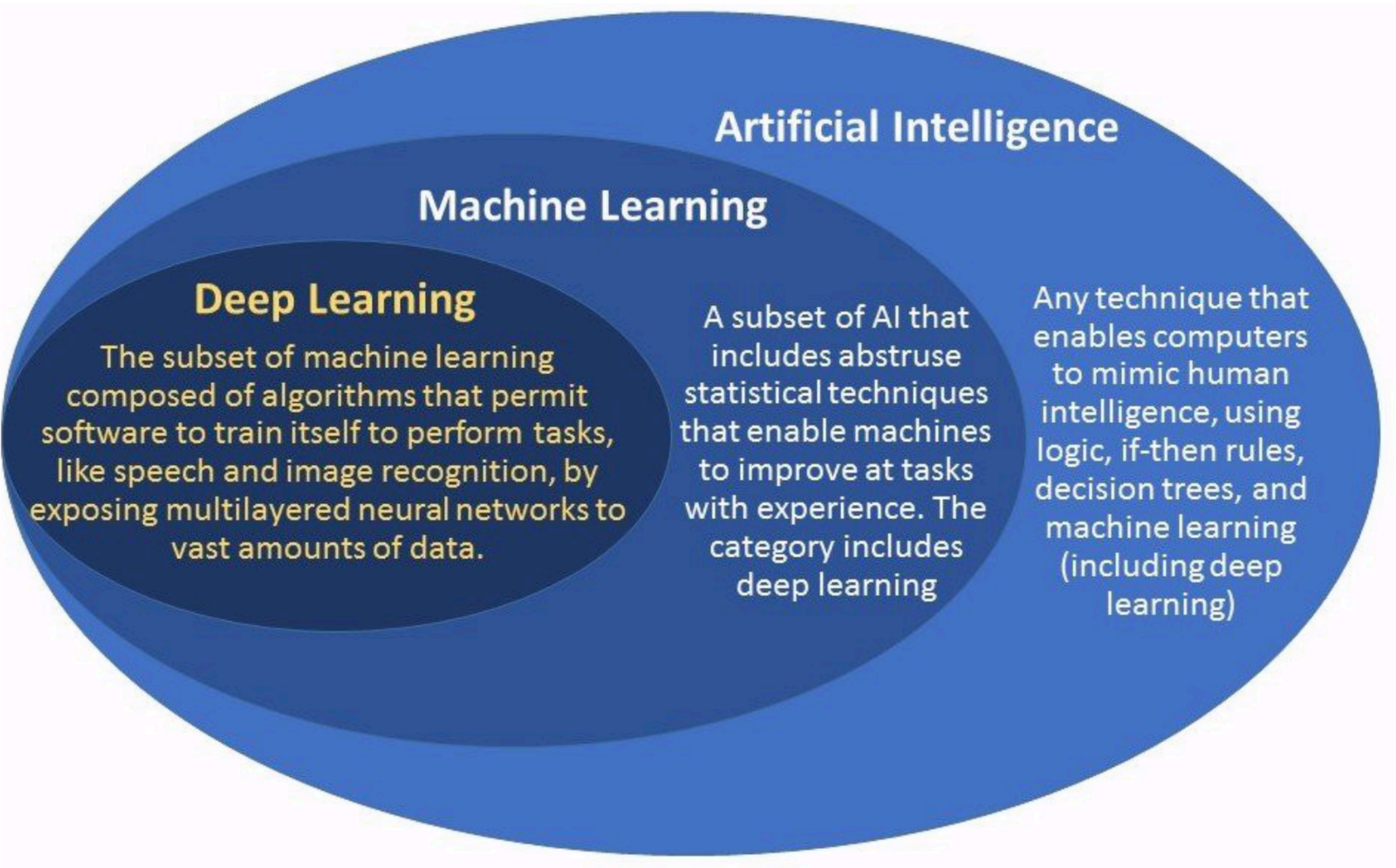
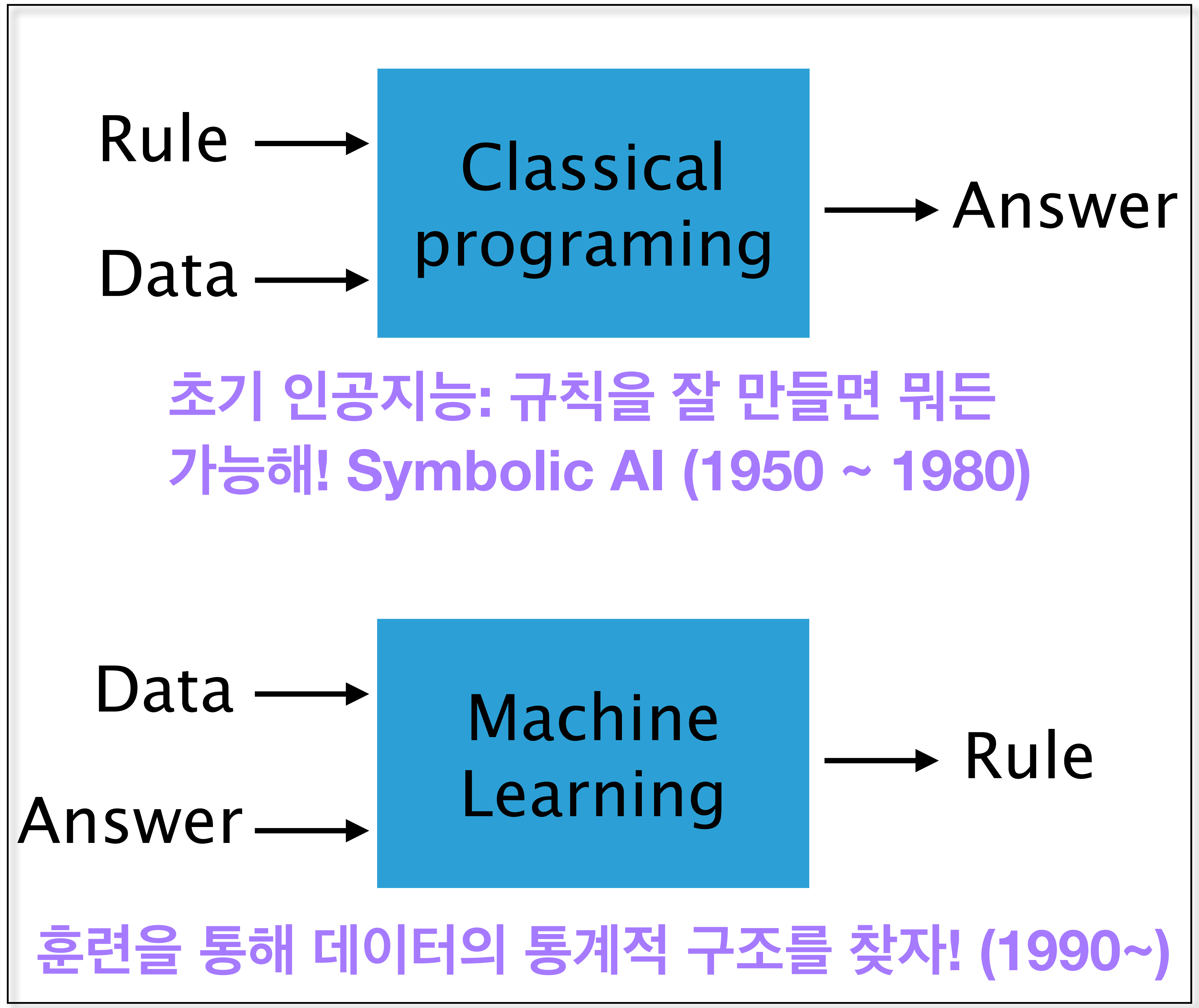
Machine Learning
Lecture 1

Machine Learning School
인하대학교 권민정

AI, Machine Learning, Deep Learning



Maching learning:
new programing paradigm



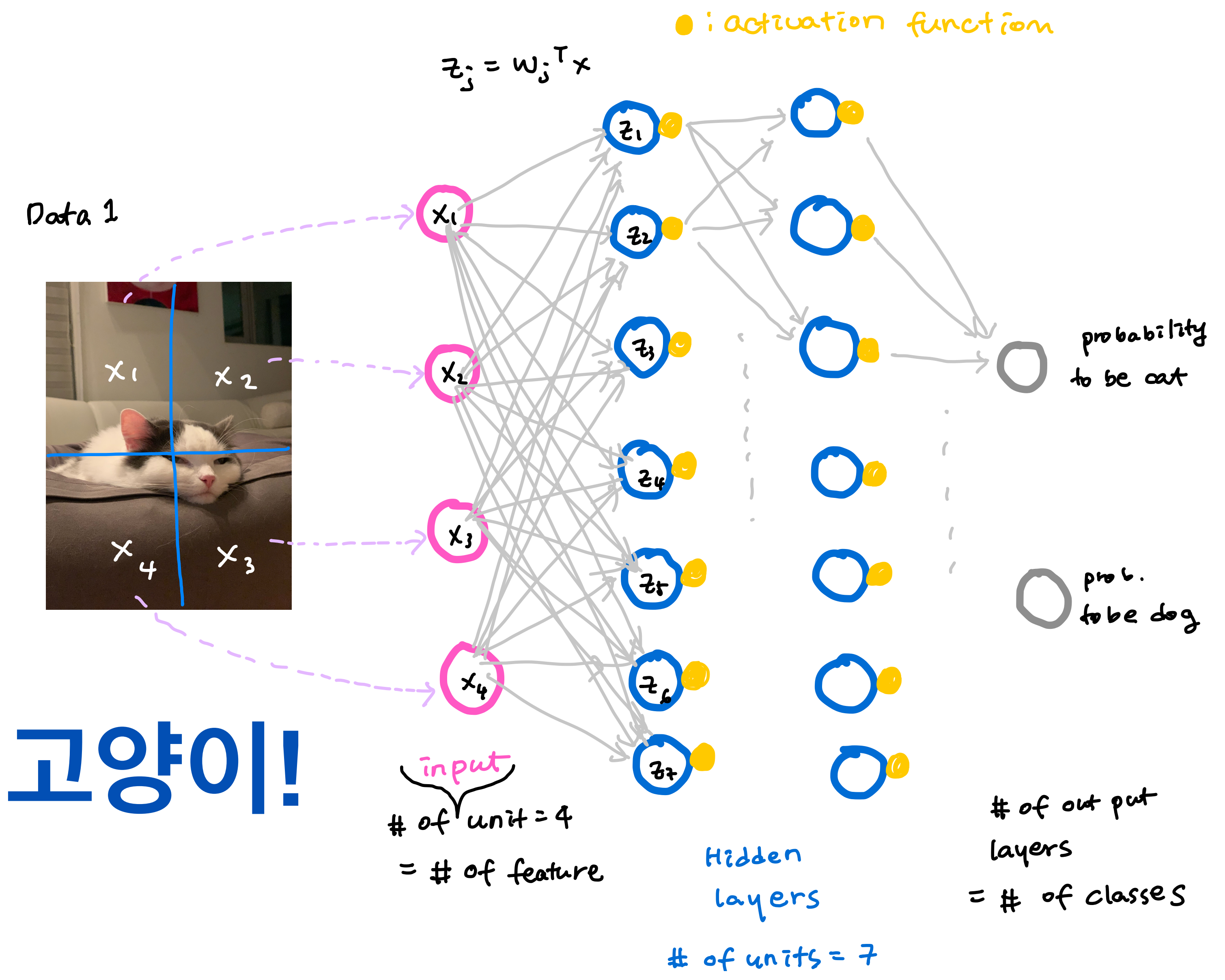
School의 목표는

다양한 물리학 분야에서 (특히나 HEP 분야에서도) 머신러닝, 딥러닝을 사용하고 있는데,

- 기초를 알면 더 정확히 깊이 있게 이해하고
- 응용 폭도 넓어진다.

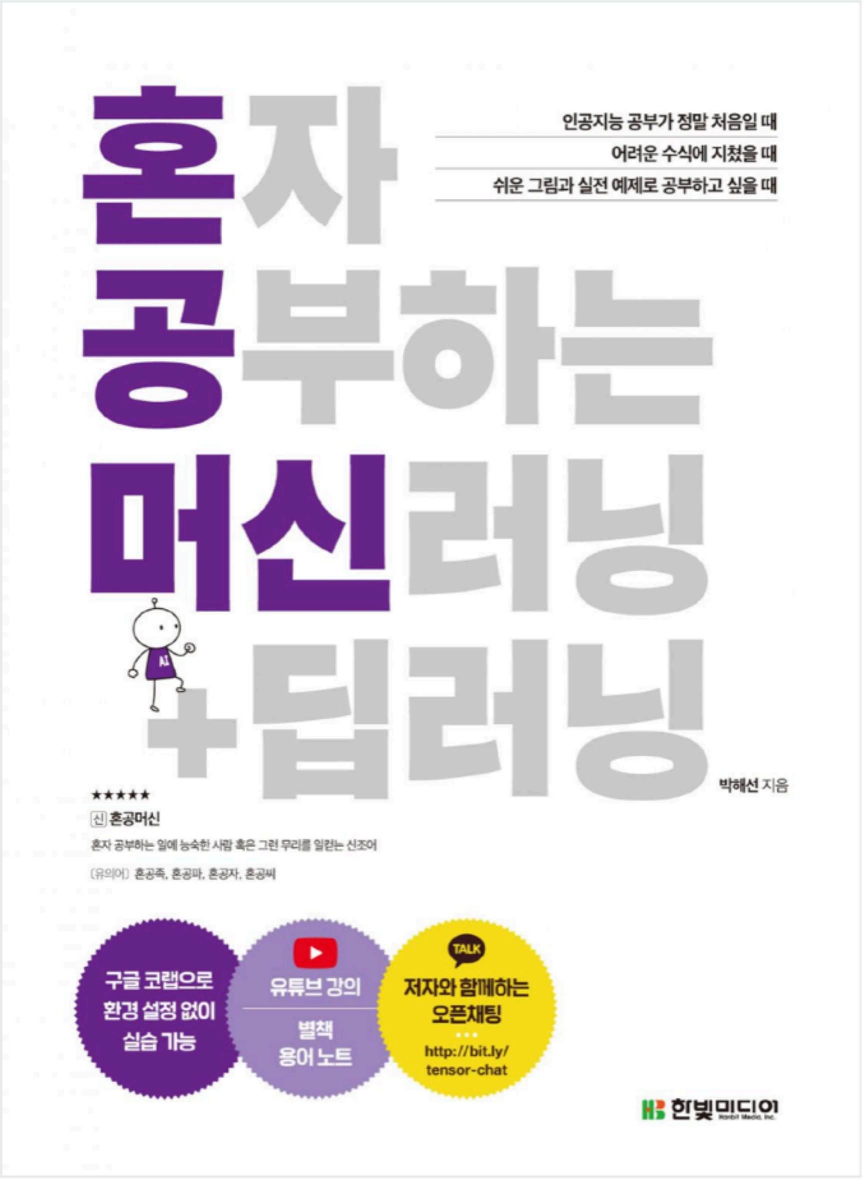
딥러닝계의 “hello world” :)

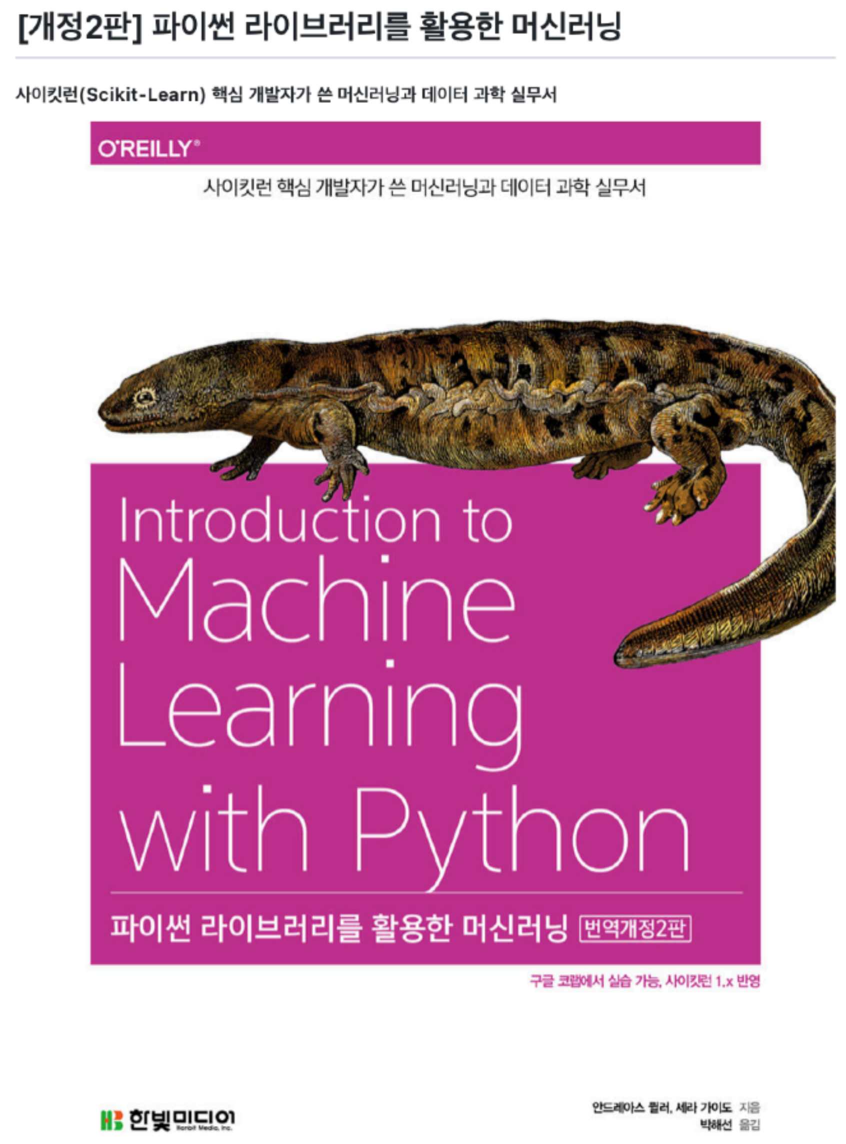
- 머신러닝/딥러닝 알고리즘의 기초를 이해한 후 간단한 문제에 응용해보자.
- 이후의 프로그레스는, up to your interest!

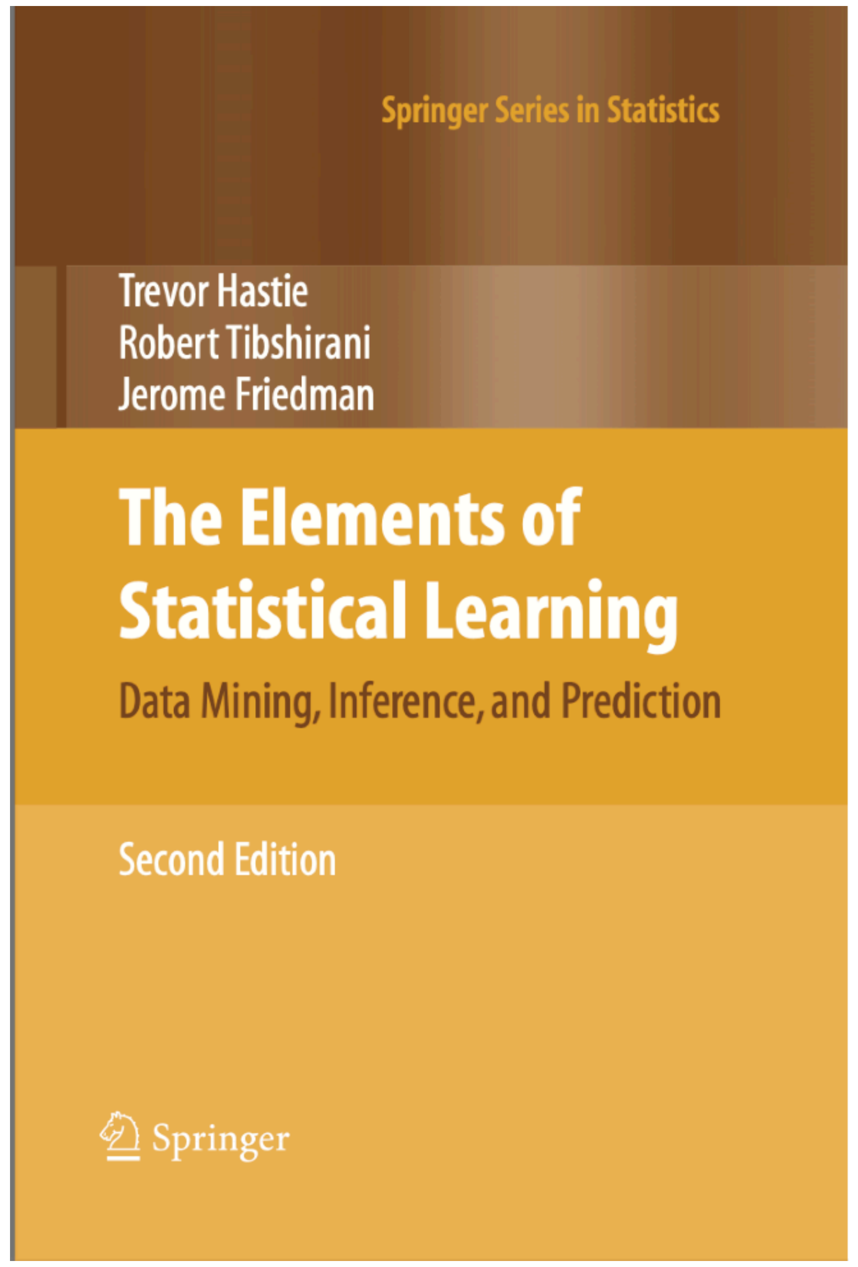


. . .

References







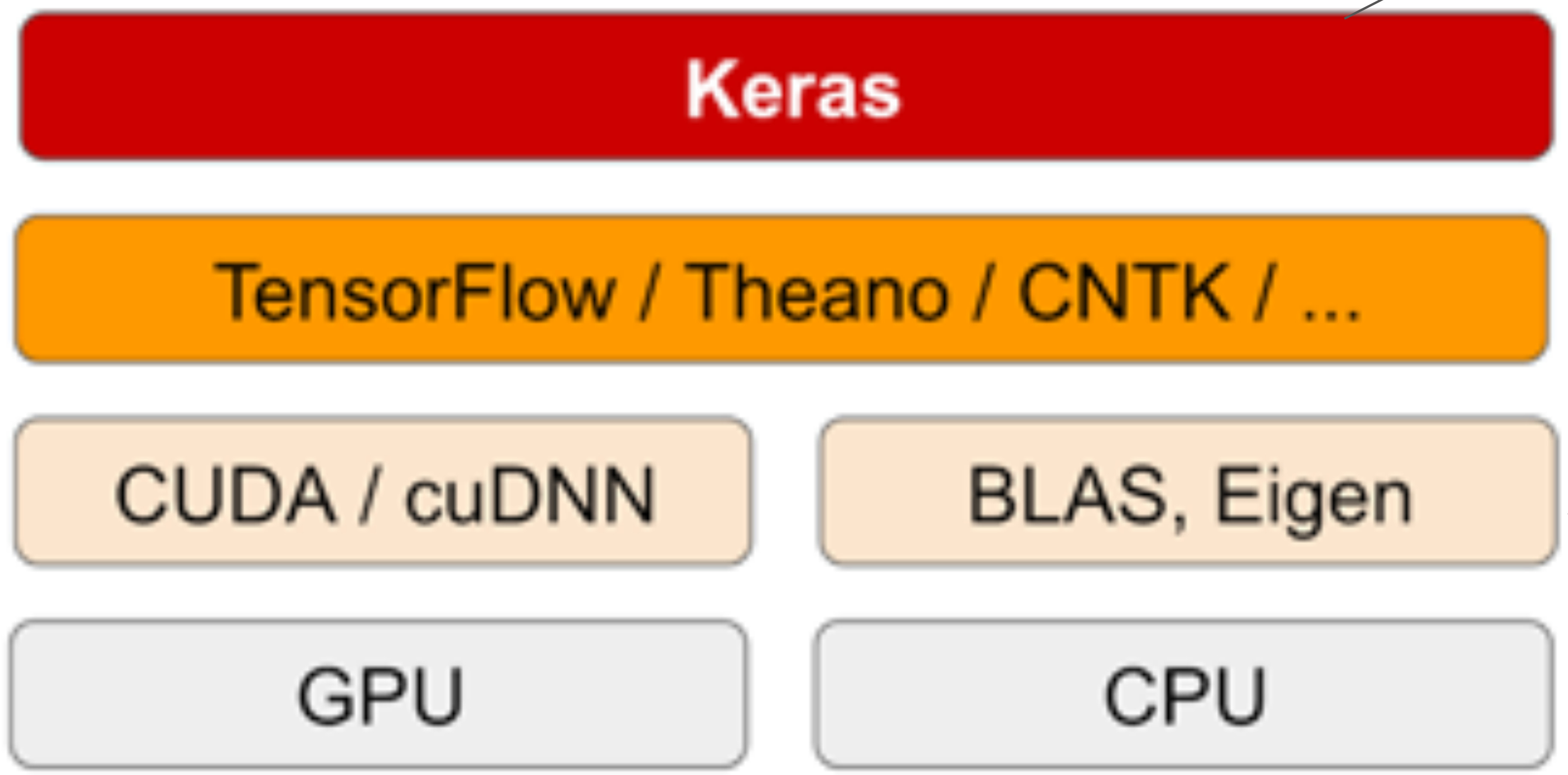
+ many… + YouTube lectures

Tools

파이썬을 위한 딥러닝 프레임워크
- 사용하기 쉬운 API, 어떤 딥러닝 모델에도 적합

API. Model-level library providing high-level building block for developing deep-learning model

고수준의 구성요소 제공



Handle low-level operations such as tensor library to do so (serving as backend engine of keras)

텐서 연산, 미분과 같은 저수준 연산에 최적화된 텐서 라이브러리 (딥러닝을 위한 주요 플랫폼)

Low-level library of well-optimized deep-learning operation

저수준 딥러닝 연산 라이브러리

Google Colab

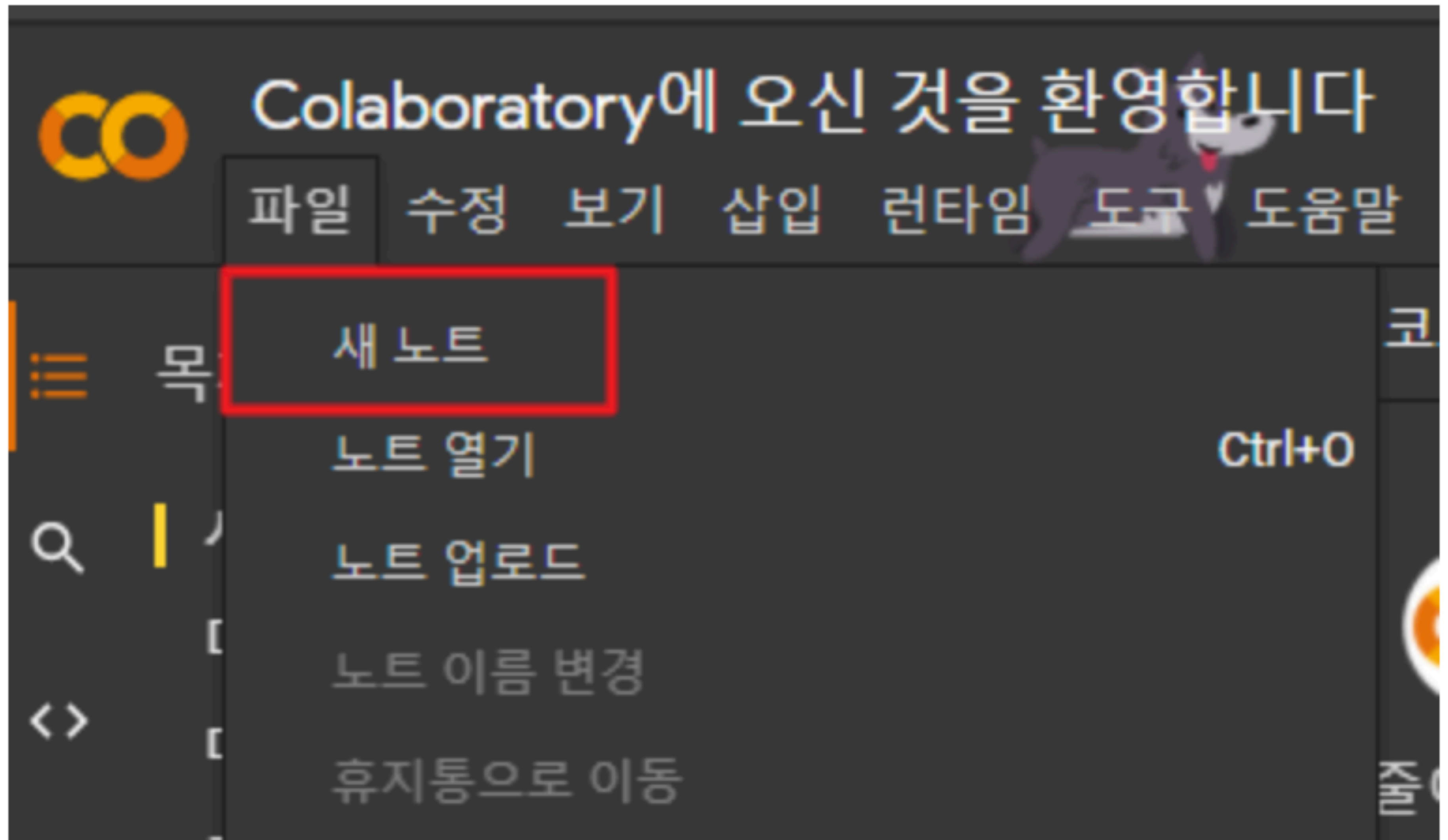
Google Colaboratory

- 브라우저에서 python 작성, 실행 가능
- 클라우드 기반의 주피터 노트북 개발 환경
- 구글 드라이브, 도커, 리눅스, 구글 클라우드 등의 기술

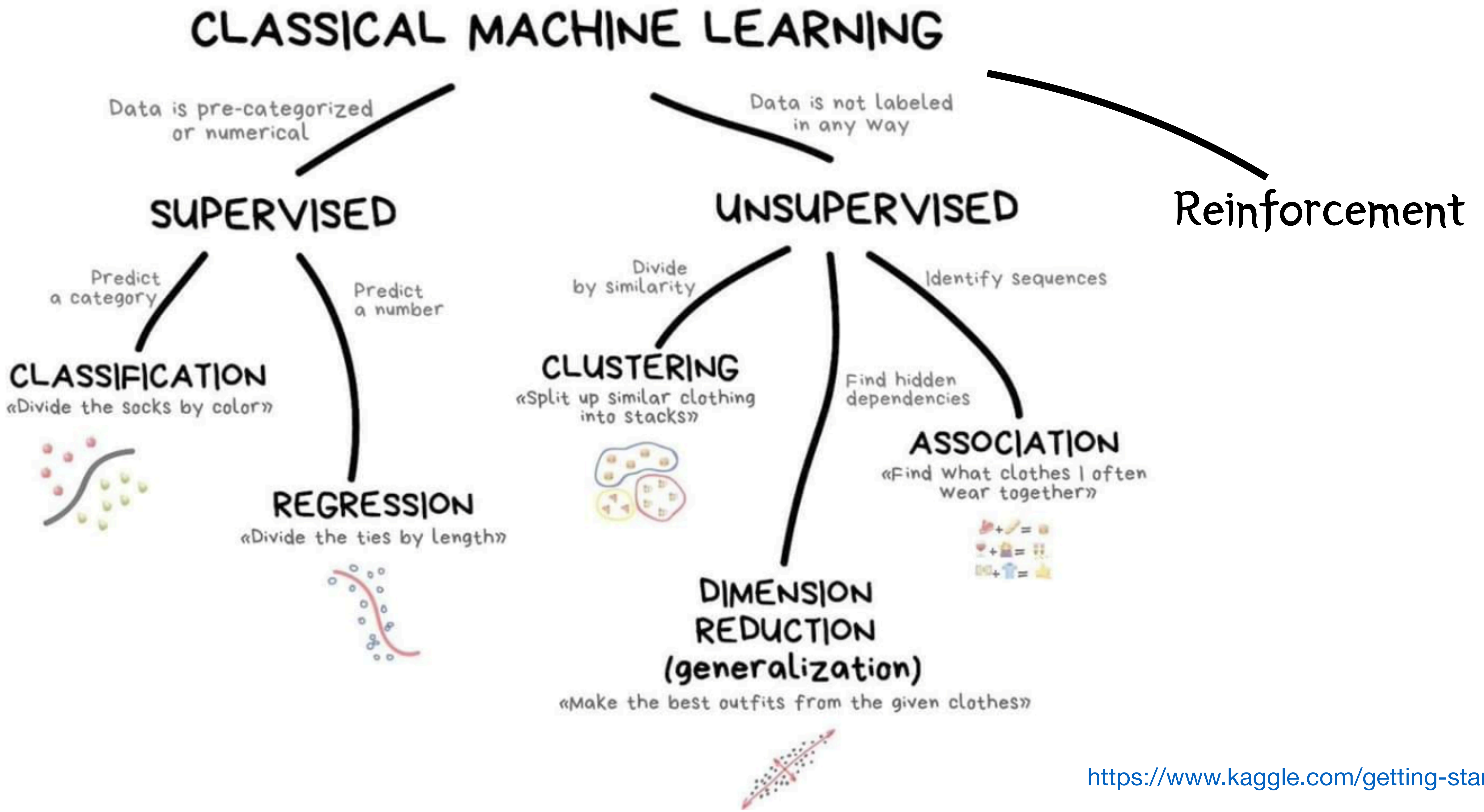
- 별도로 파이썬을 설치할 필요가 없음
- Tensor Flow, Keras, metaplotlib, scikit-learn, pandas와 같은 패키지가 이미 설치되어 있음
- GPU 무료 사용 가능
- 주피터 노트북과 비슷하지만 더 좋은 기능 제공
- Git과 연동 가능

구글 로그인을 한 뒤 colab.research.google.com/notebooks/intro.ipynb

에 접속합니다.



Machine Learning Algorithms



https://www.kaggle.com/getting-started/169622

Types of Machine Learning Algorithms

지도학습 (Supervised)

- 레이블된 데이터
- 직접 피드백
- 출력 및 예측

비지도학습 (Unsupervised)

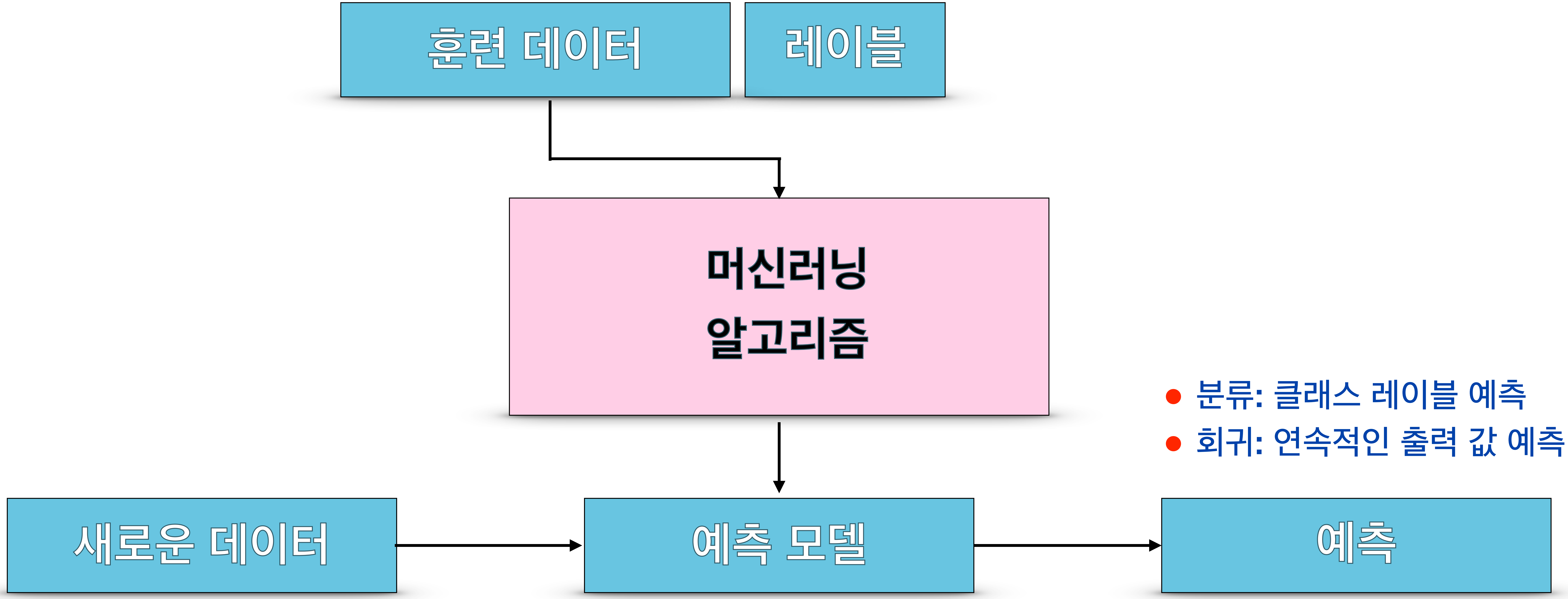
- 레이블 및 타깃 없음
- 피드백 없음
- 데이터에서 숨겨진 구조 찾기

강화학습 (Reinforcement)

- 결정 과정
- 보상 시스템
- 연속된 행동에서 학습

지도학습 기반의 예측

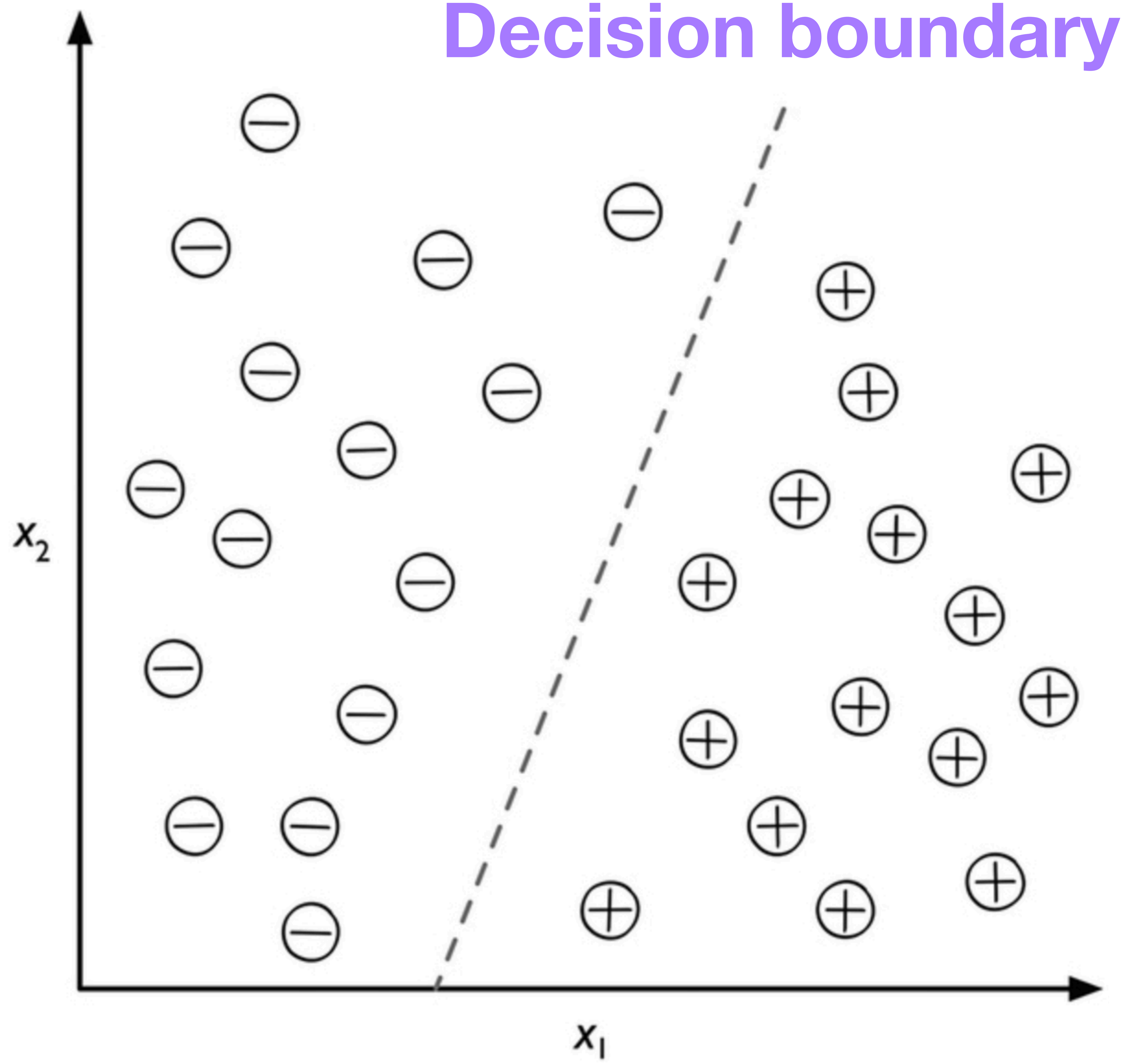
지도학습 (Supervised)



- 분류: 클래스 레이블 예측
- 회귀: 연속적인 출력 값 예측

지도학습: 분류

Decision boundary (규칙학습으로 결정)



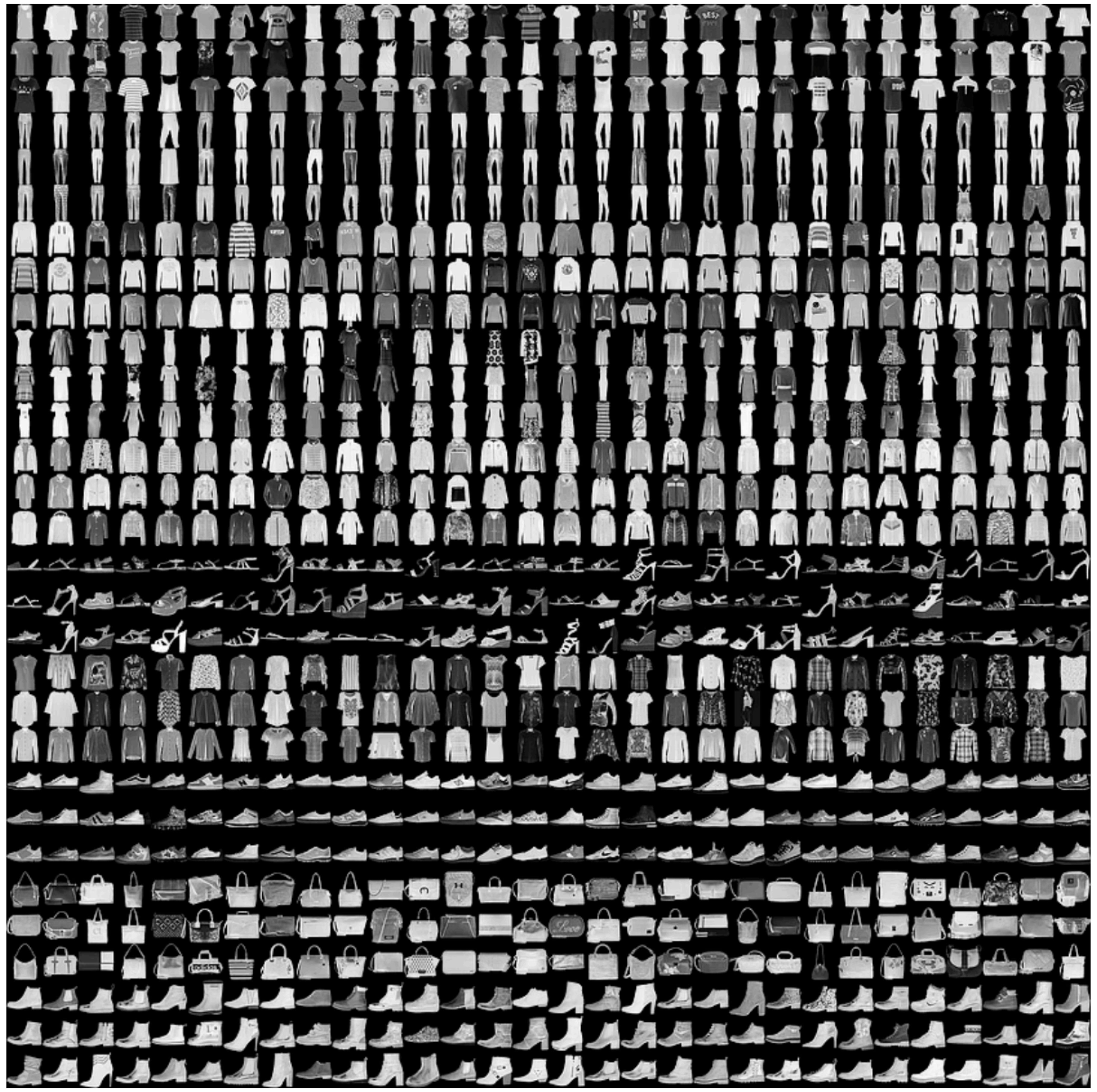
- 데이터를 기반으로, 새로운 데이터(샘플) 분류 (클레스 레이블 예측)

- x_1 = 10, x_2 = 15

?

지도학습: 분류 – 예제

Figure 2.1 MNIST sample digits



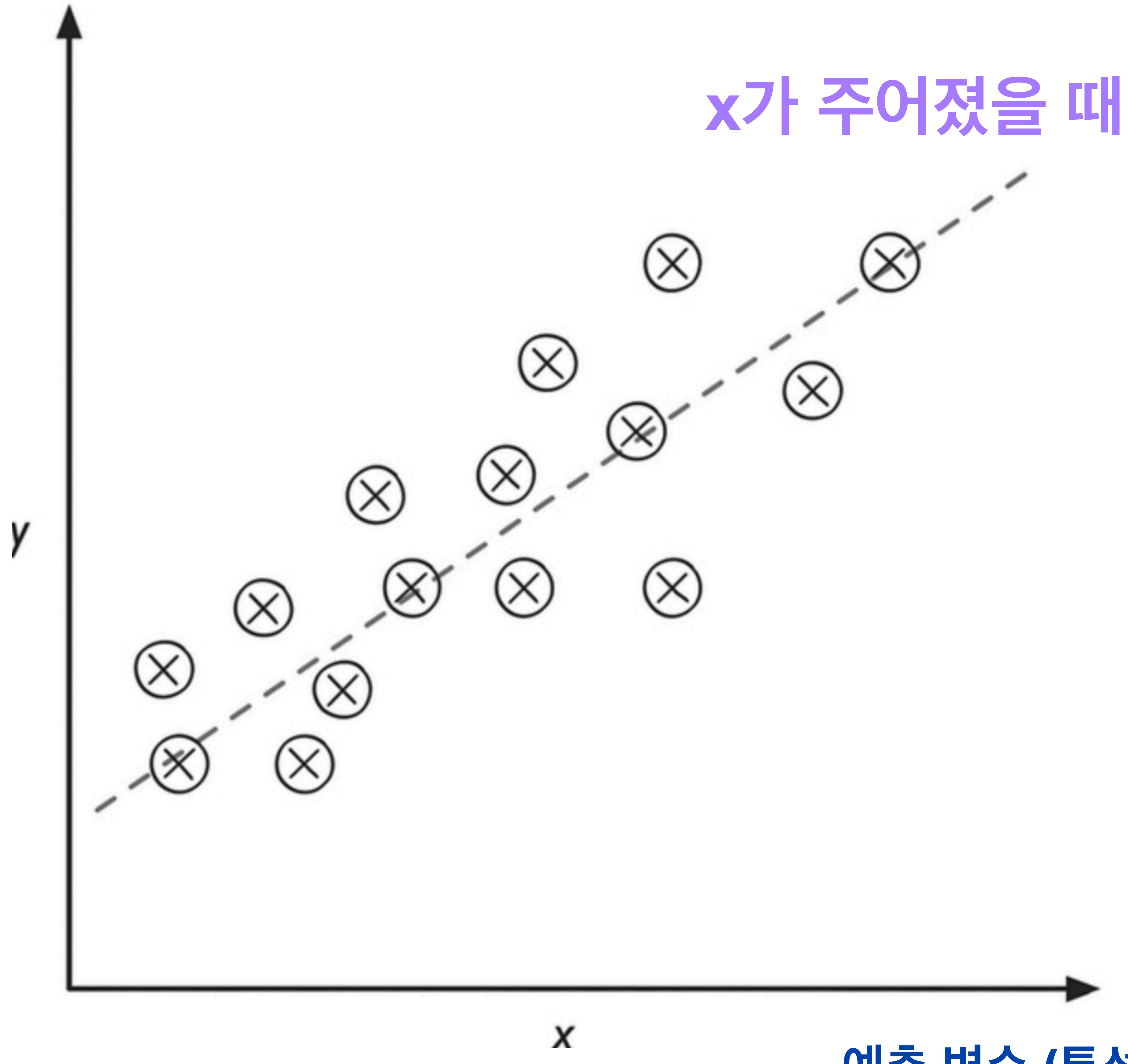
패션 MNIST

지도학습: 회귀

지도학습 (Supervised)

반응 변수 (타깃)

x가 주어졌을 때 y를 예측하기 위한 두 변수사이의 관계 학습



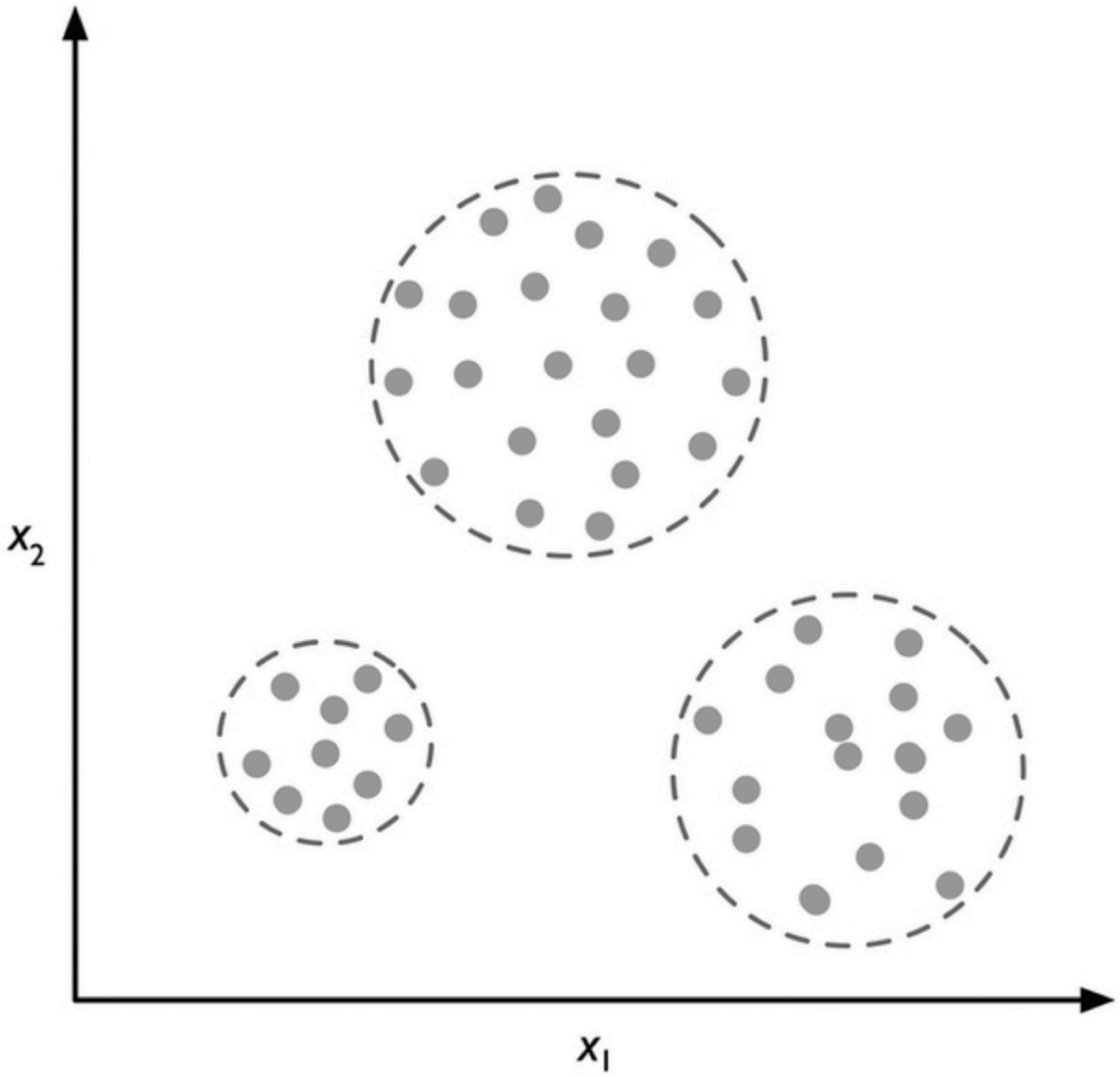
예측 변수 (특성)

지도학습: 회귀 – 예제

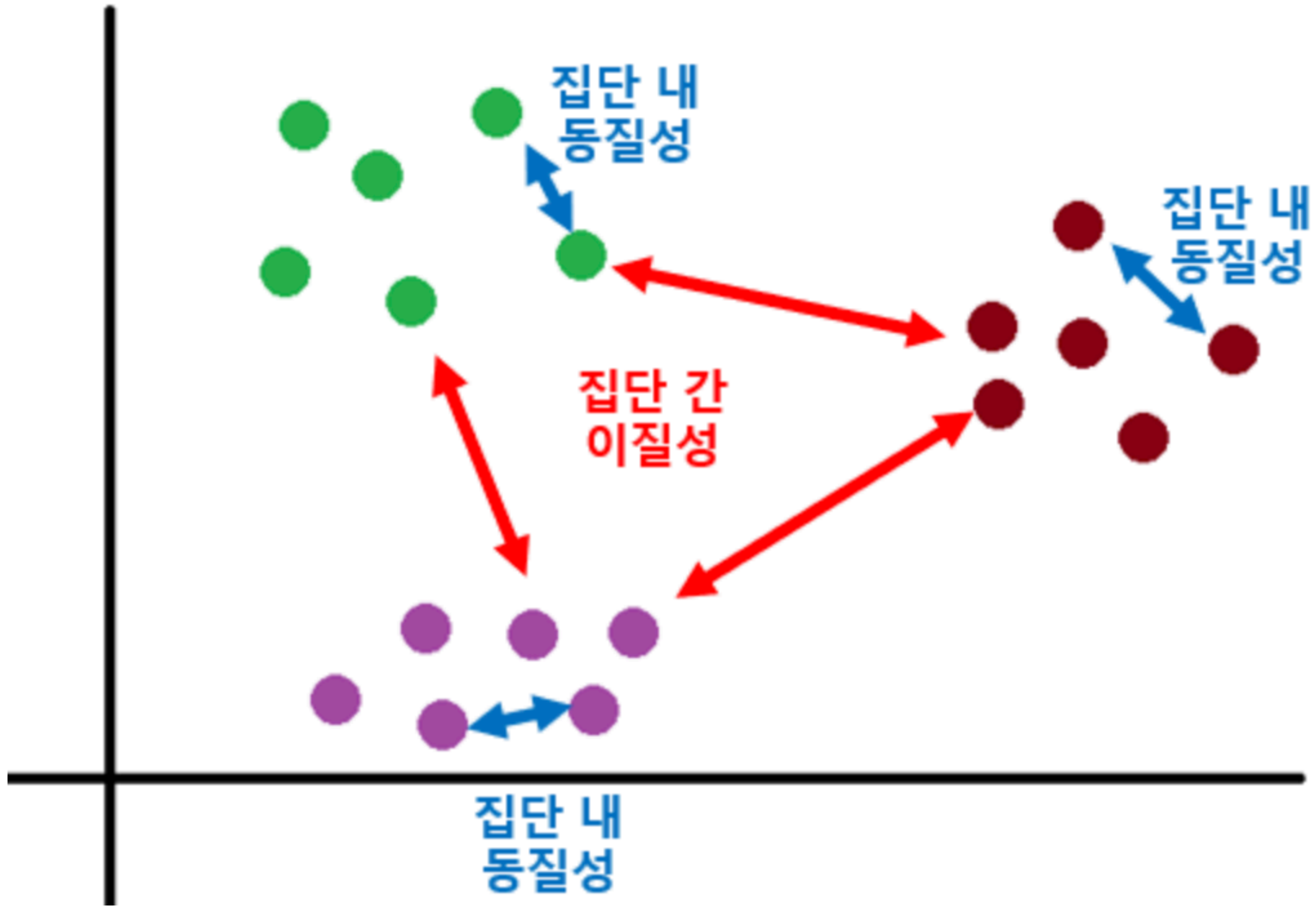

회귀분석 예측
원래 데이터
300
200
100
0
-100
-2
-1
0
x
y
Degree 1
MSE = 4.08e-01(+/- 4.25e-01)
Degree 4
MSE = 4.32e-02(+/- 7.08e-02)
Degree 15
MSE = 1.82e+08(+/- 5.45e+08)
Model
True function
Samples
x
y

비지도학습 기반의 구조 발견: Clustering (군집)

비지도학습 (Unsupervised)

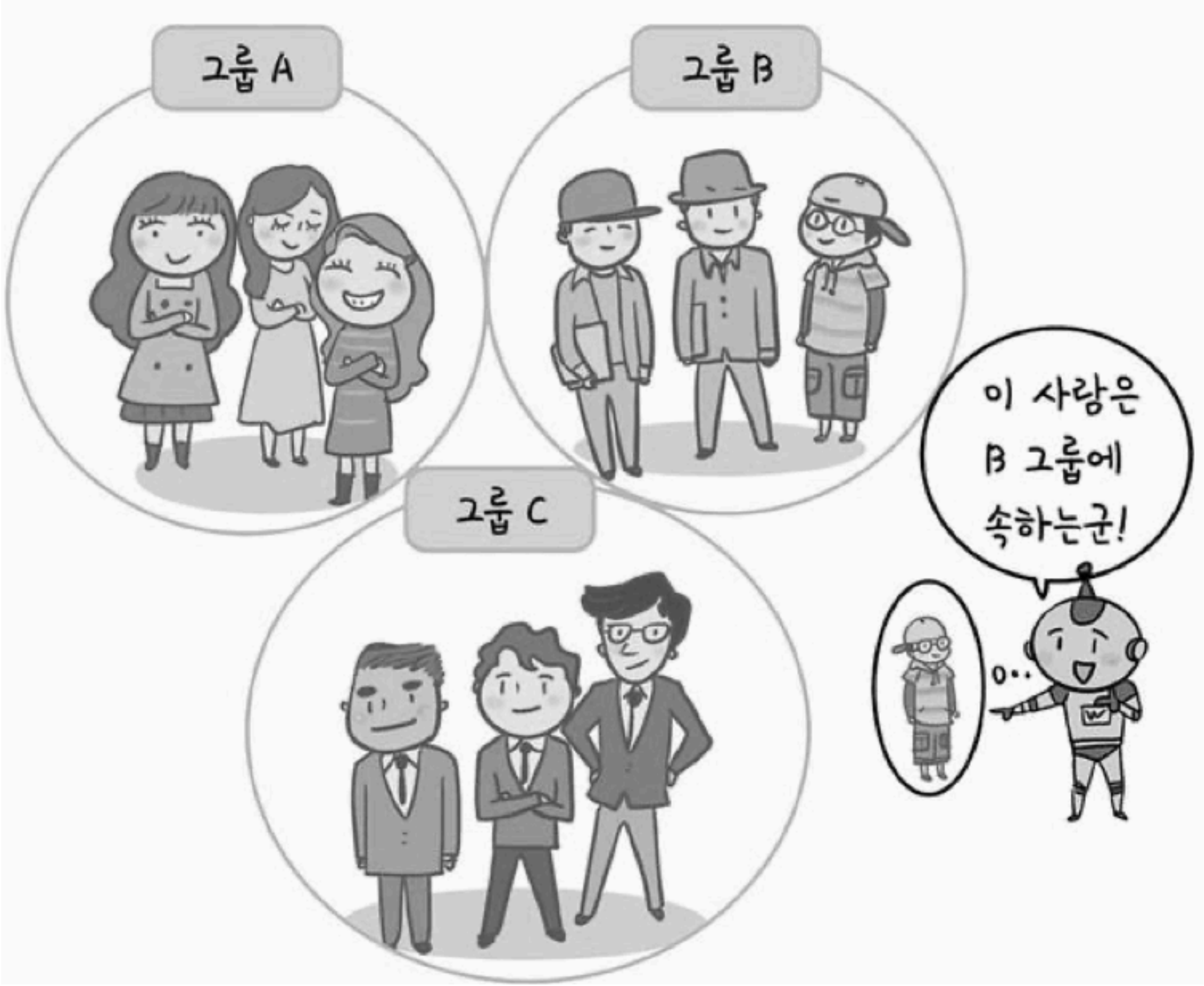


- 특성 x_1, x_2의 유사도를 기반으로 샘플 그룹을 형성

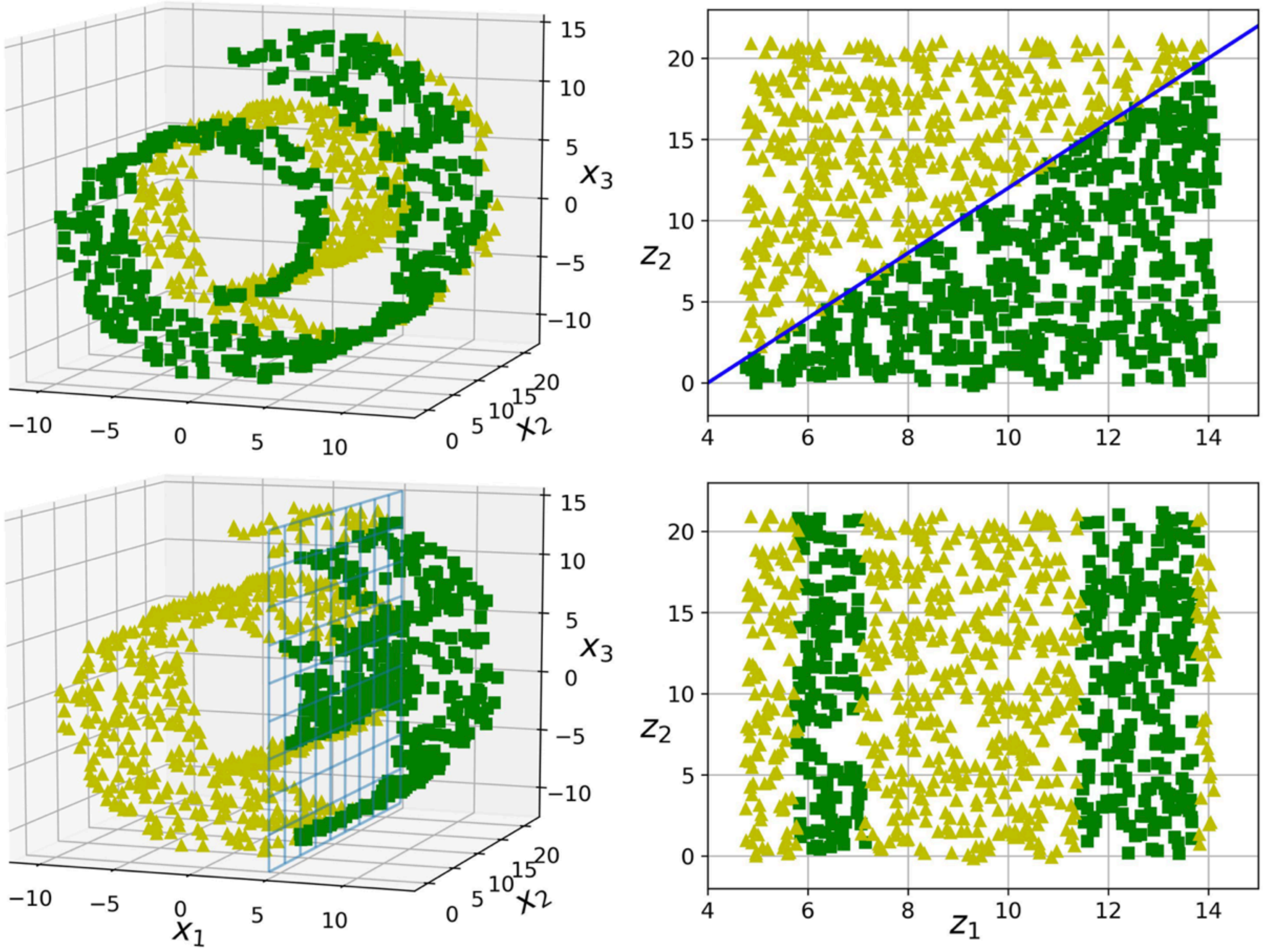


비지도학습 기반의 구조 발견: Clustering (군집)

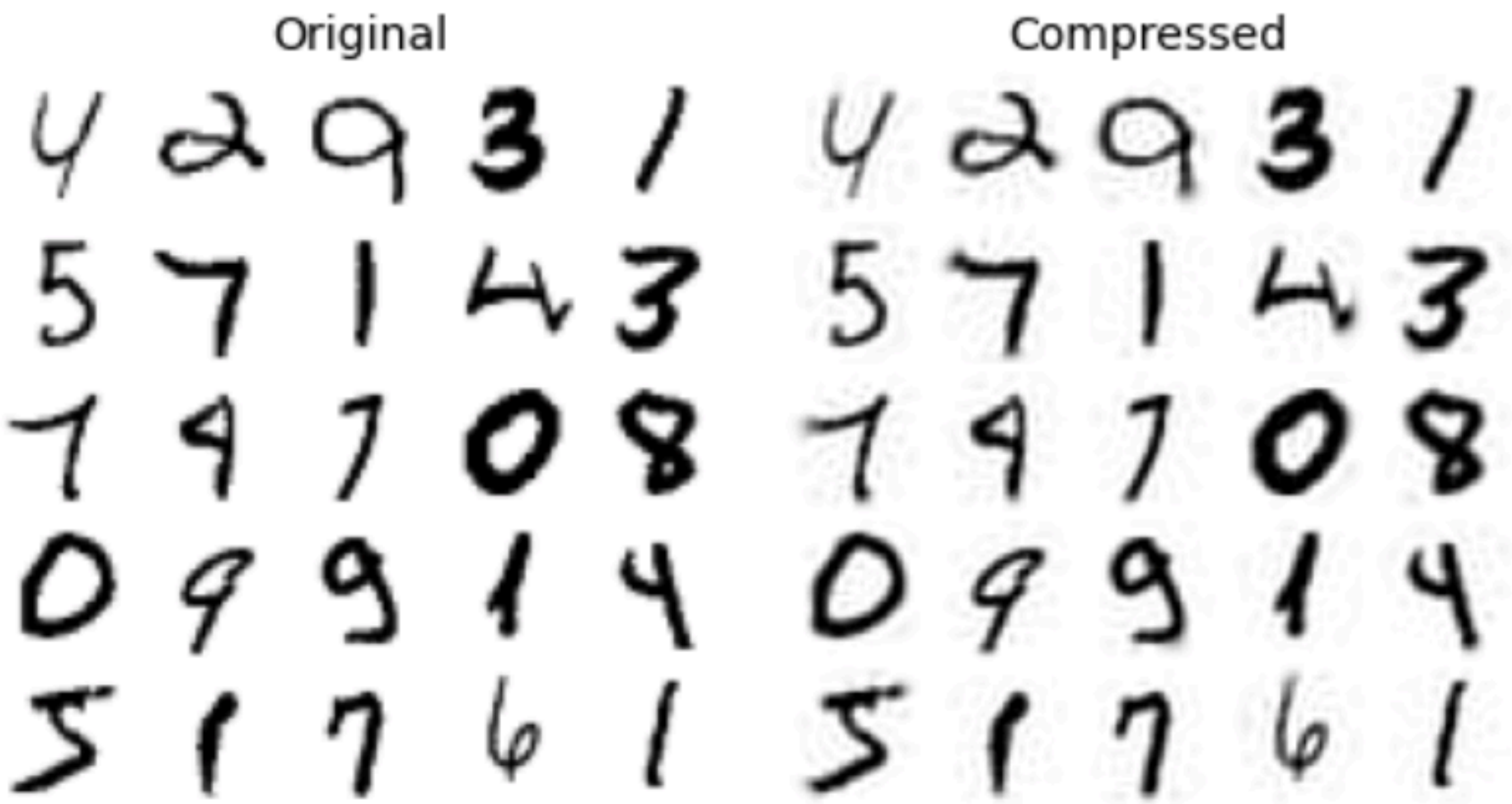
비지도학습 (Unsupervised)



비지도학습 기반의 구조 발견: 차원 축소

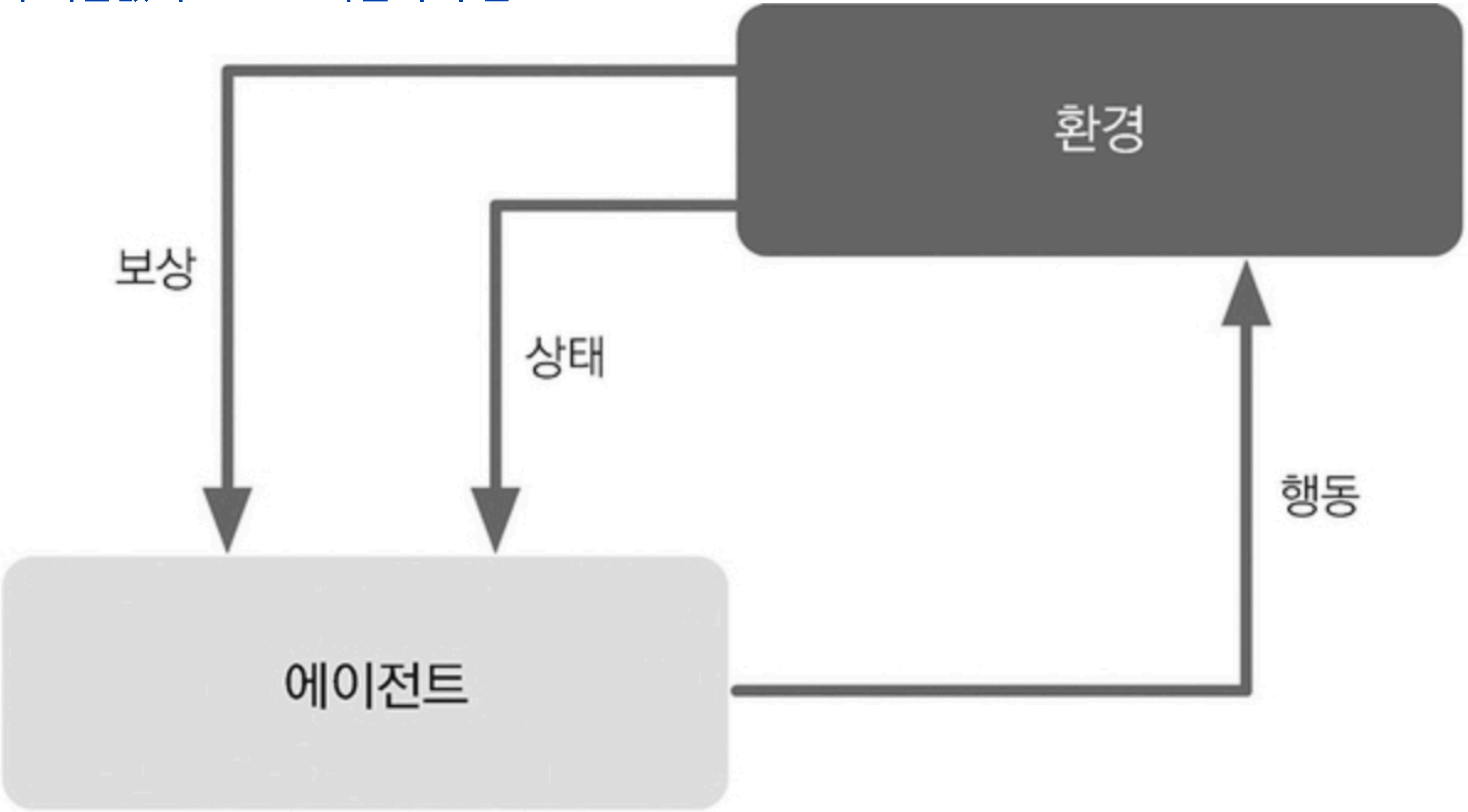


- 관련 있는 정보를 대부분 유지하면서 더 작은 차원을 가진 부분 공간으로 데이터를 압축

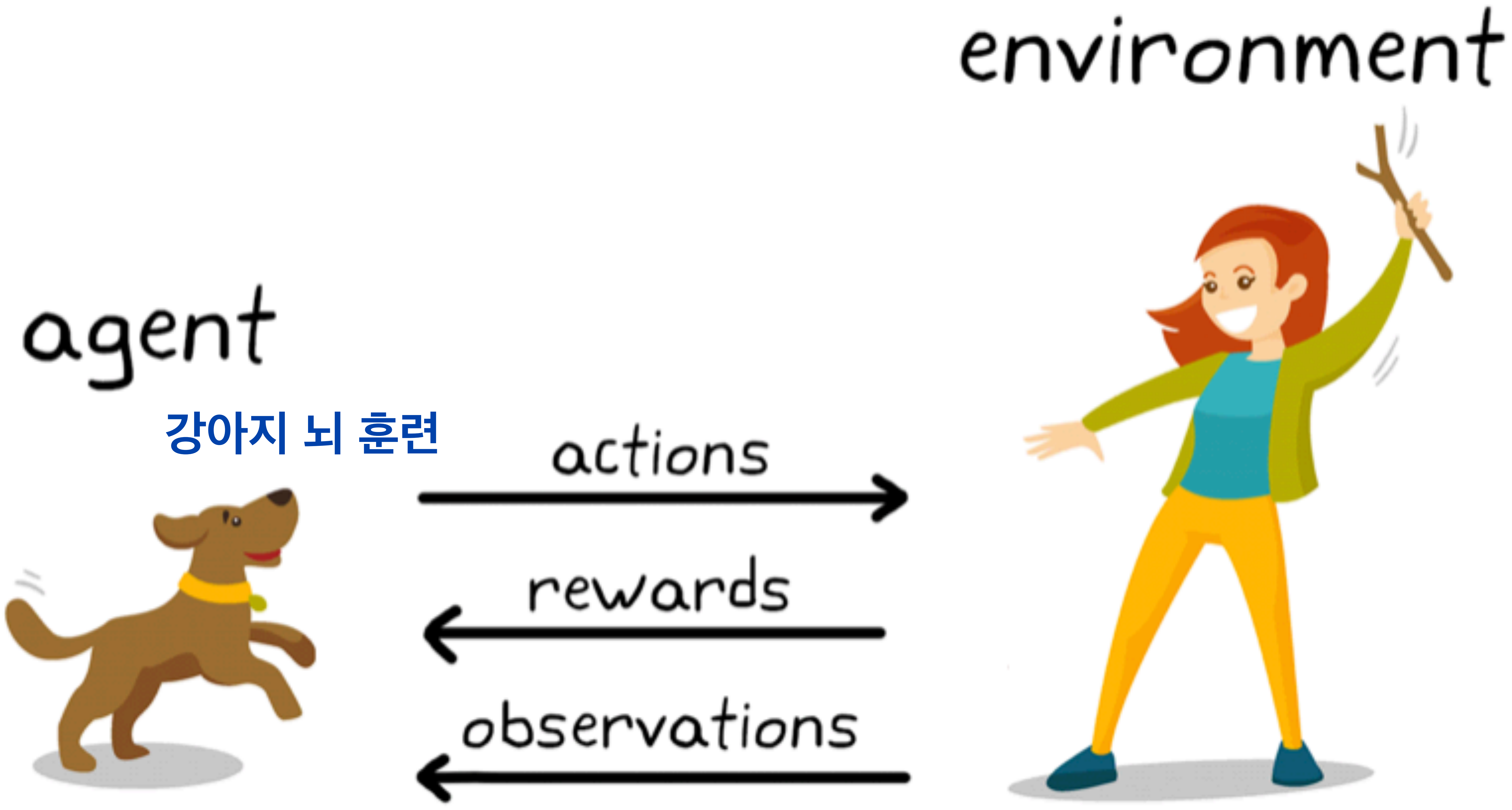


강화학습

- 반복적인 시행착오 상호작용을 통해 작업 수행 방법을 학습
- 적절한 보상을 통해 인간의 개입없이 스스로 학습하게 함

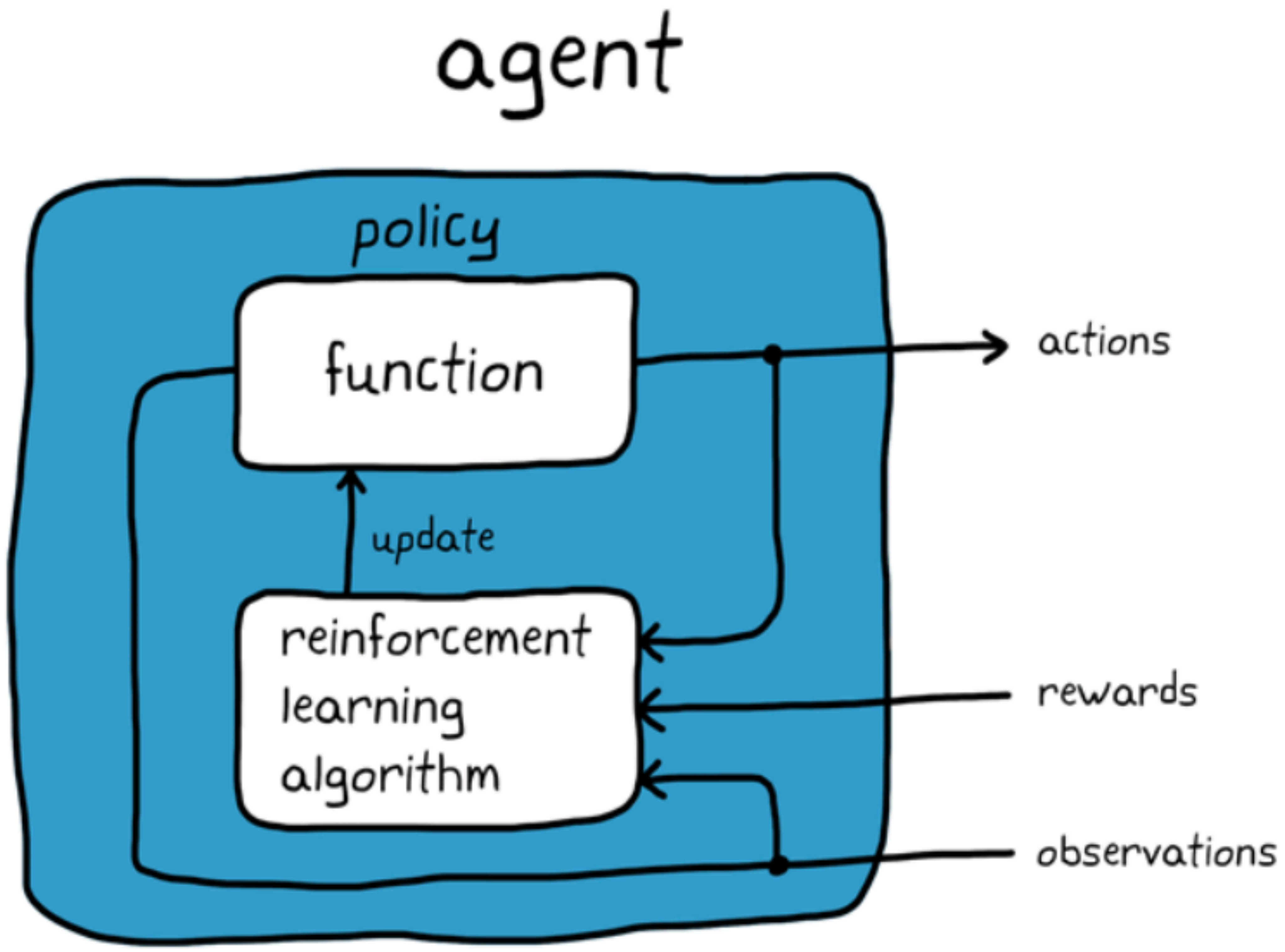


강화학습 – 예제



- 보상 - 간식, 간식, 간식.....

강화학습 – 예제



자율주행 훈련 알고리즘

- 보상 - 주차 OK!

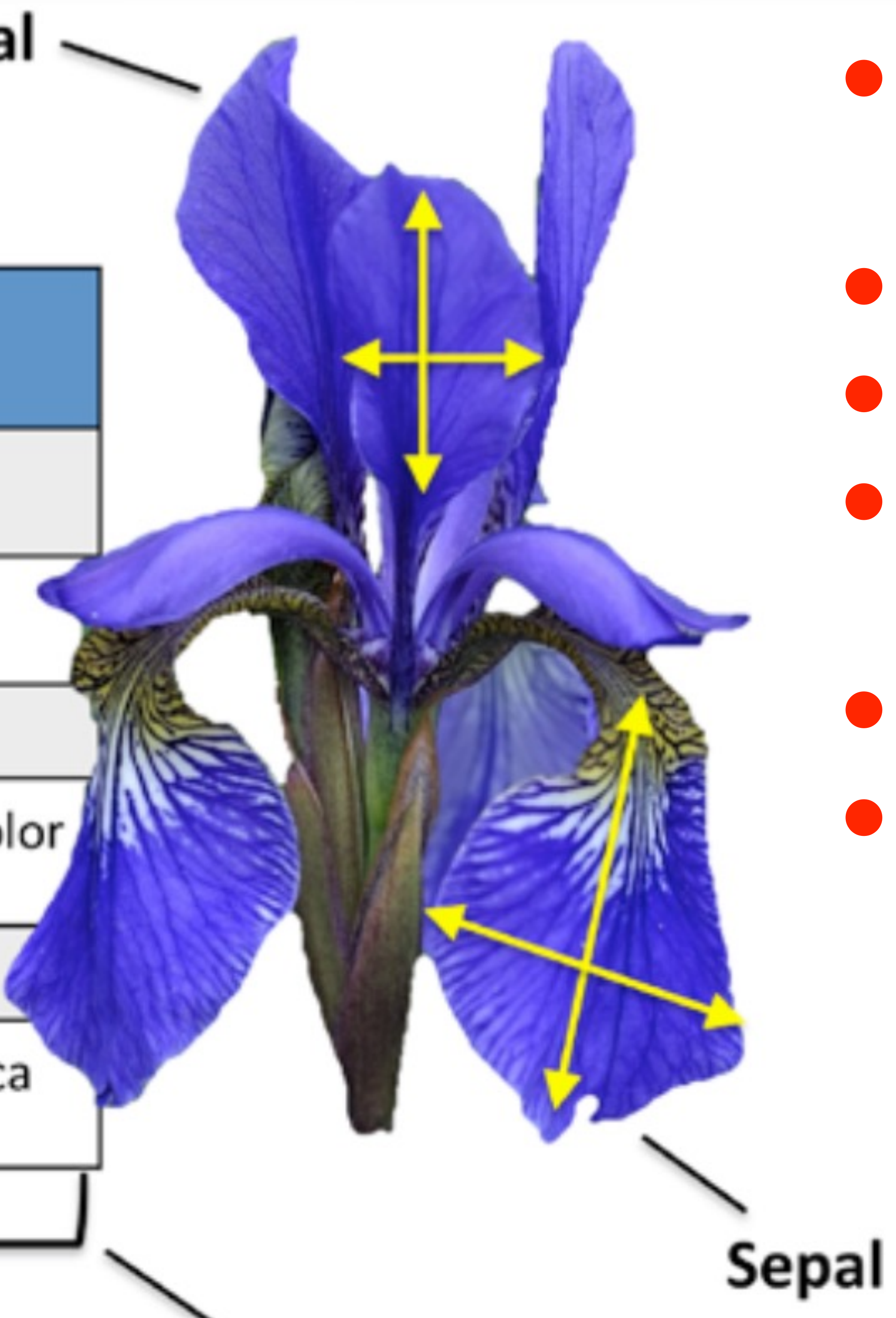
표기법과 규칙 (책마다 다름)

Samples
(instances, observations)

	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
			...		
50	6.4	3.5	4.5	1.2	Versicolor
			...		
150	5.9	3.0	5.0	1.8	Virginica

Features
(attributes, measurements, dimensions)

Class labels
(targets)

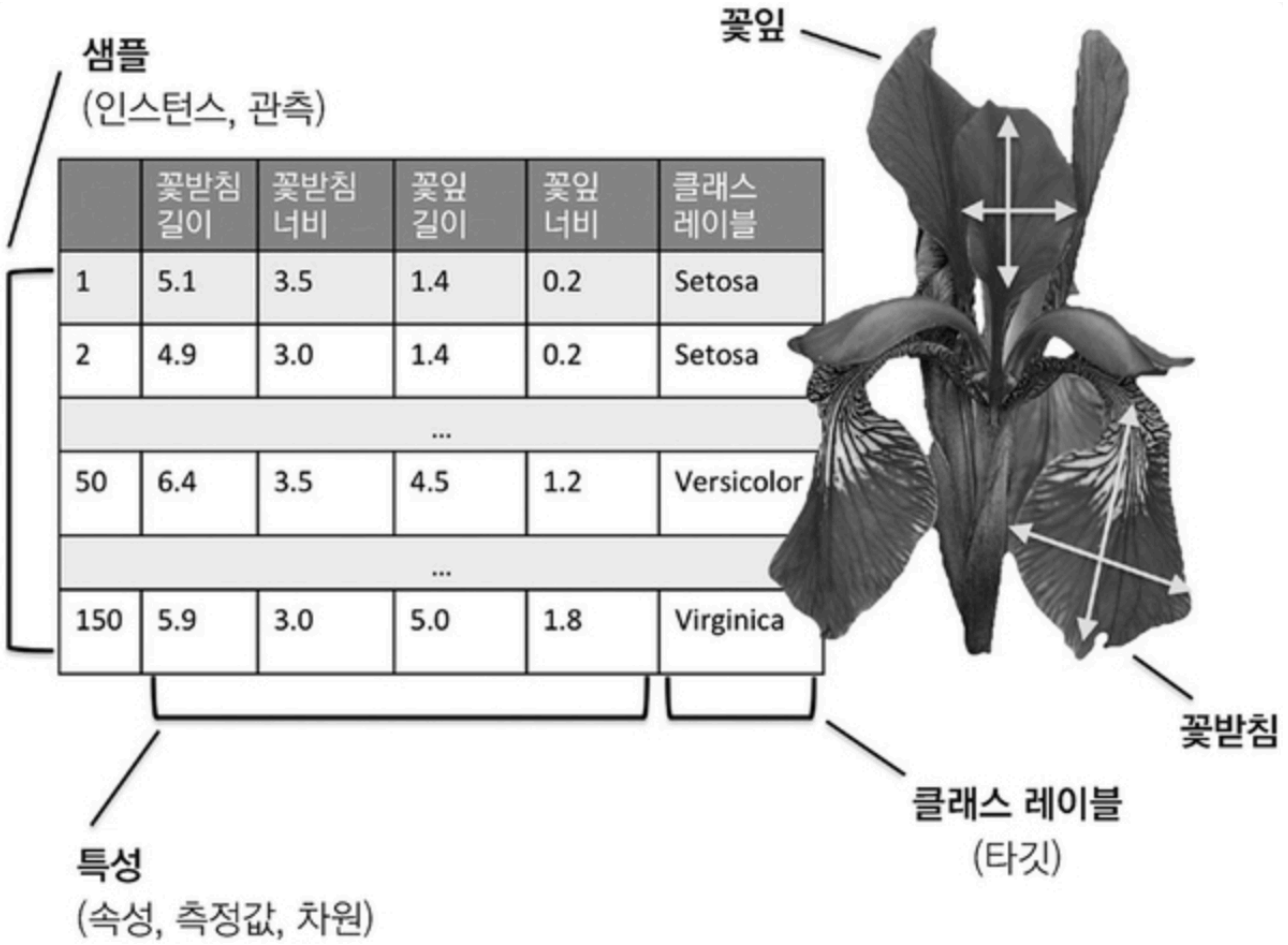


- **Training samples (sets)** : Observations, Records, Instances
- **Training** : Model fitting, Parameter estimation,...
- **Feature (x)** : Descriptor, Variable, Input, Attribute,...
- **Target (y)**: Outcome, Output, Class label, Ground truth,...
- Loss function: Cost function
- Slope (w) : weight,...

$$\begin{bmatrix} x_1^{(1)} & x_2^{(1)} & x_3^{(1)} & x_4^{(1)} \\ x_1^{(2)} & x_2^{(2)} & x_3^{(2)} & x_4^{(2)} \\ \vdots & \vdots & \vdots & \vdots \\ x_1^{(150)} & x_2^{(150)} & x_3^{(150)} & x_4^{(150)} \end{bmatrix}$$

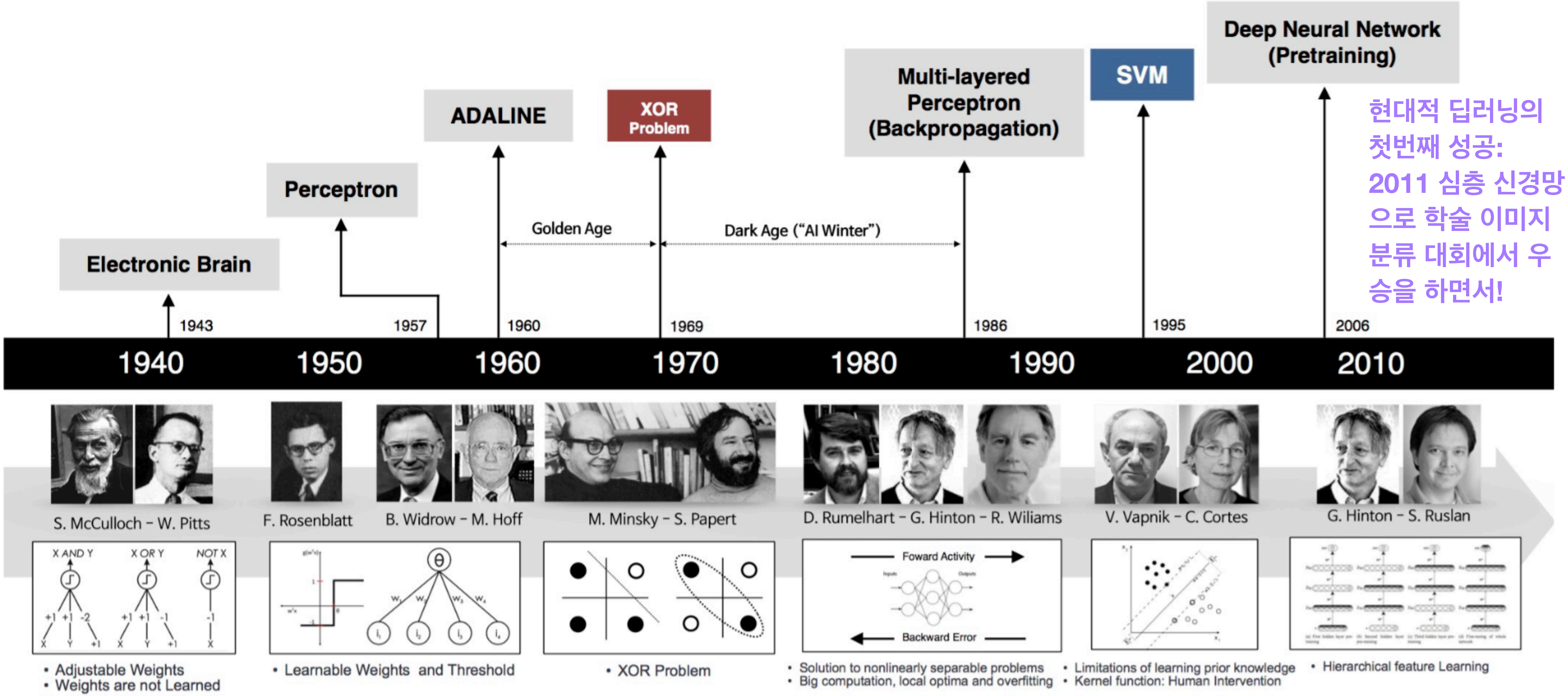
$$\boldsymbol{y} = \begin{bmatrix} y^{(1)} \\ \dots \\ y^{(150)} \end{bmatrix} (y \in \{\text{Setosa}, \text{Versicolor}, \text{Virginica}\})$$

표기법과 규칙 (책마다 다름): 한국어로는…



	꽃받침 길이	꽃받침 너비	꽃잎 길이	꽃잎 너비	클래스 레이블
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

Milestons in the development of neural networks

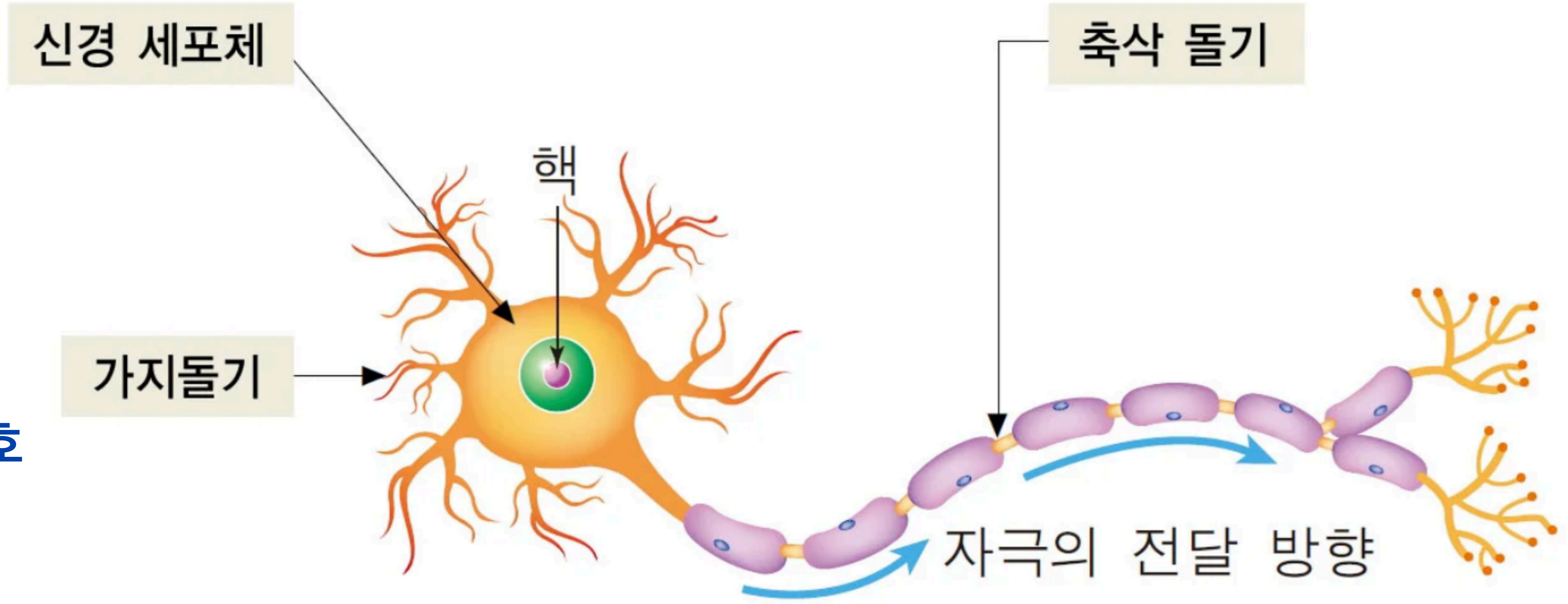


현대적 딥러닝의 첫번째 성공: 2011 심층 신경망으로 학술 이미지 분류 대회에서 우승을 하면서!

2012년 ImageNet 대회. 1,400만개 이미지를 1천개 카테고리로 분류: 심층합성곱신경망사용 83.6%, 2015: 96.4%

http://beamlab.org/deeplearning/2017/02/23/deep_learning_101_part1.html

인공 뉴런: 초기 머신 러닝의 역사

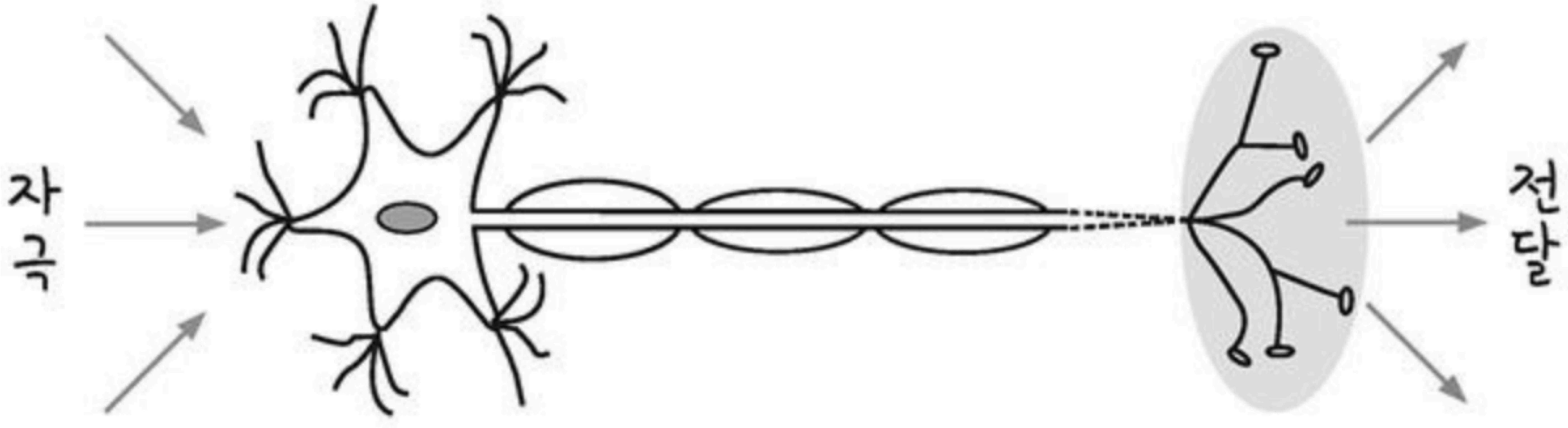


입력신호

출력신호

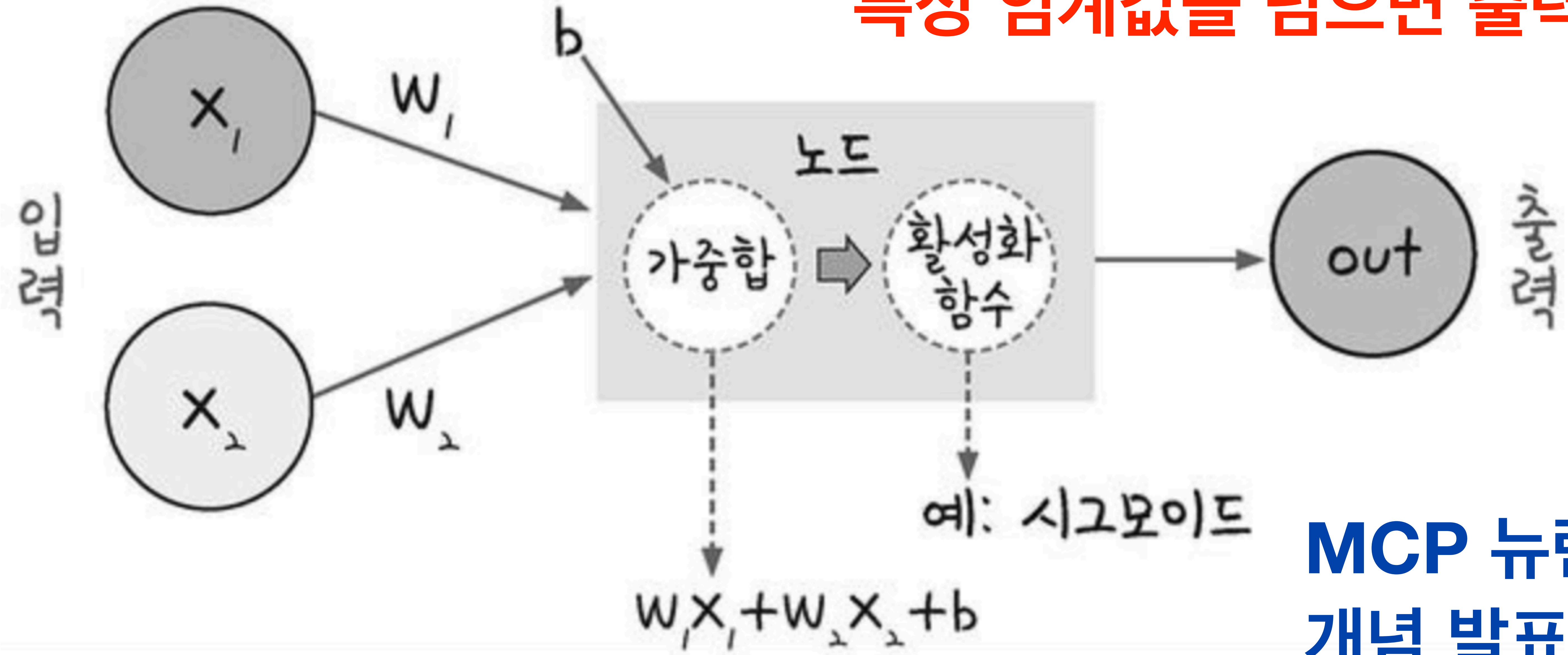
1943년 매컬록, 피츠, 신경세포를 논리 회로로 표현 (MCP 뉴런모델)

인공 뉴런: 초기 머신 러닝의 역사



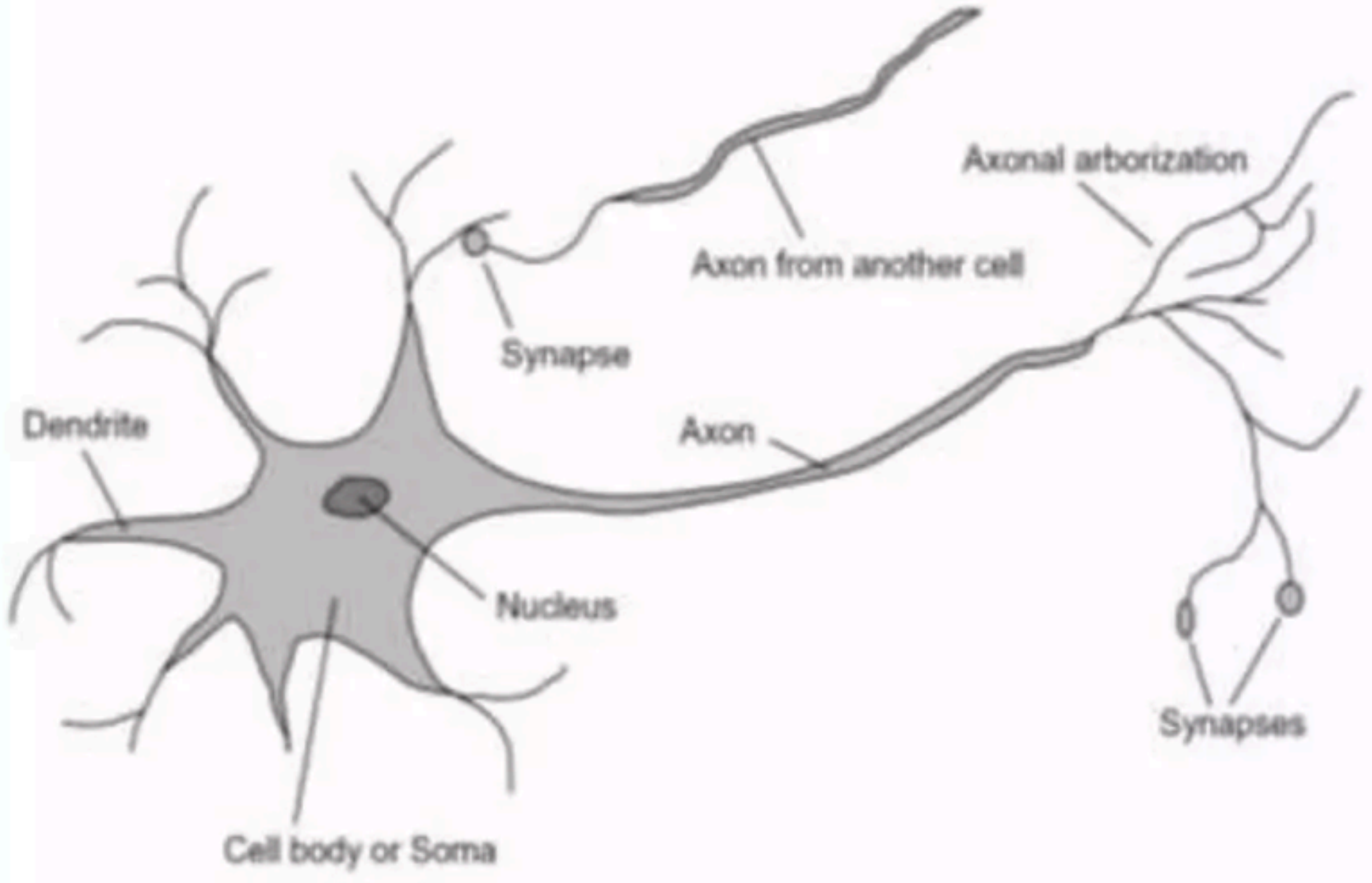
MCP 뉴런 모델

특정 임계값을 넘으면 출력 신호 생성

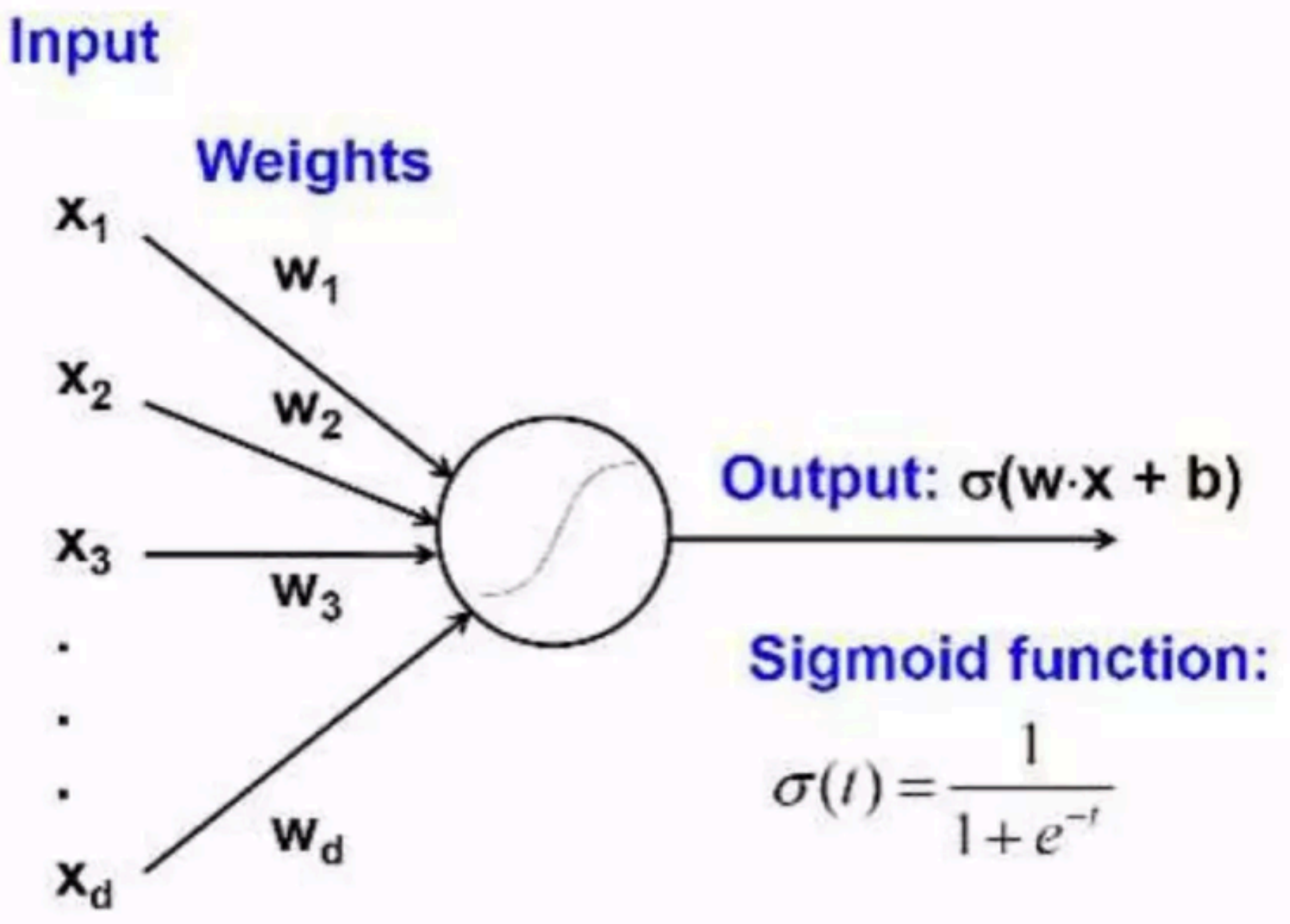


MCP 뉴런 모델을 기반으로 퍼셉트론 개념 발표 (프랑크 로젠블라트)

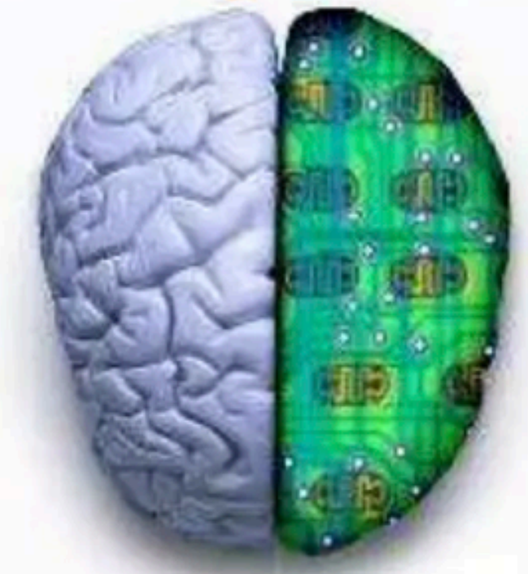
인공 뉴런: 초기 머신 러닝의 역사 – 퍼셉트론



A biological neuron



An artificial neuron (Perceptron)
- a linear classifier



자동으로 최적의 가중치를
학습하는 알고리즘 제안

가장 고전적인 선형 분류 모델

인공 뉴런: 초기 머신 러닝의 역사

input

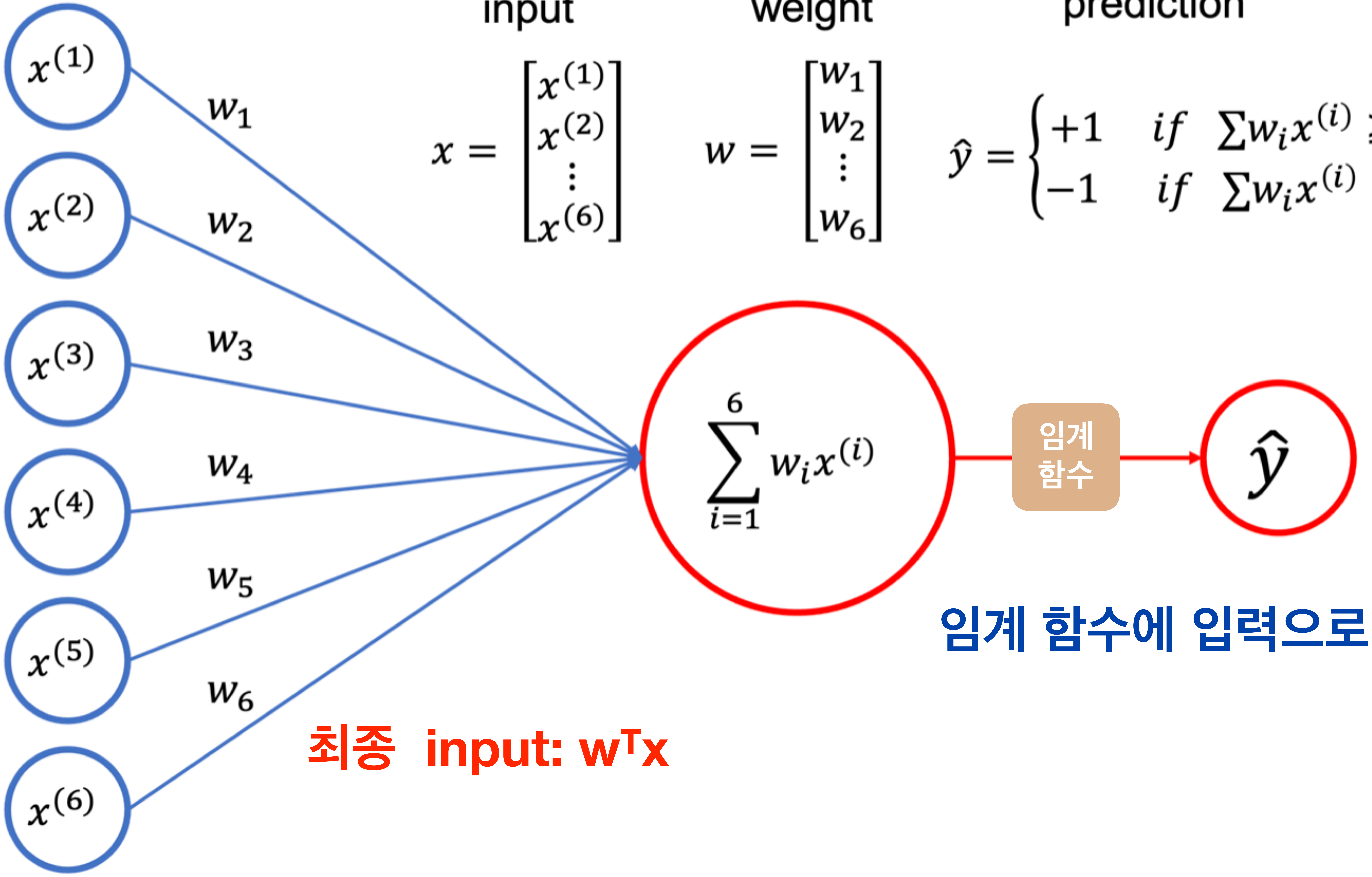
$$x = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(6)} \end{bmatrix}$$

weight

$$w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_6 \end{bmatrix}$$

prediction

$$\hat{y} = \begin{cases} +1 & if \;\; \sum w_i x^{(i)} \geq threshold \\ -1 & if \;\; \sum w_i x^{(i)} < threshold \end{cases}$$



최종 input: $w^T x$

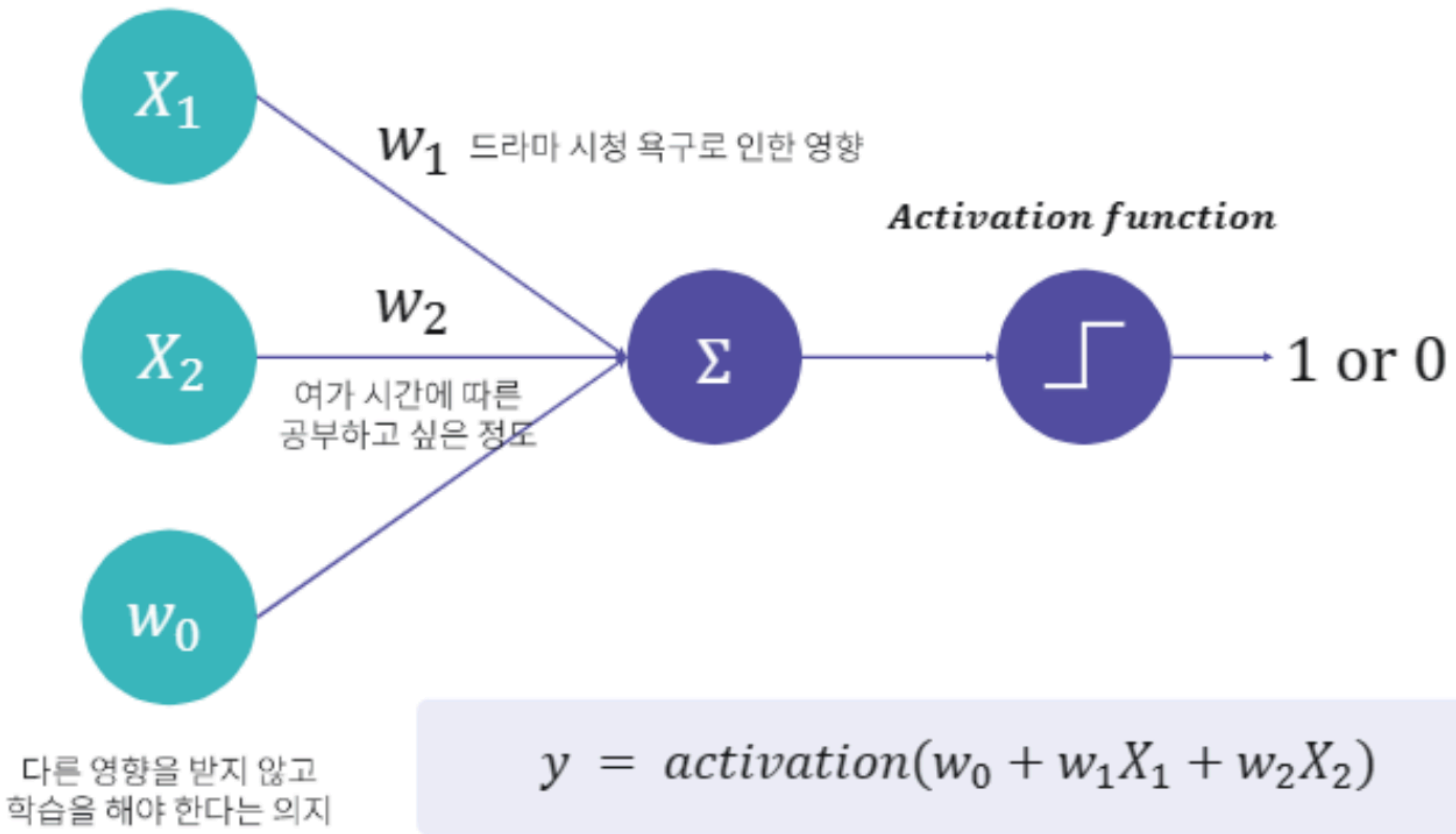
임계 함수에 입력으로 넣고 계산

퍼셉트론 동작 예시

강의 학습 여부를 예측하기 위한 데이터

Ref) https://kcy51156.tistory.com/52

오늘 나온 신작 드라마 수(x1)	확보한 여가 시간(x2)	강의 학습 여부(y)
2	4	1
5	4	1
7	1	0
3	0	0
0	2	1
4	1	0
...	...	...

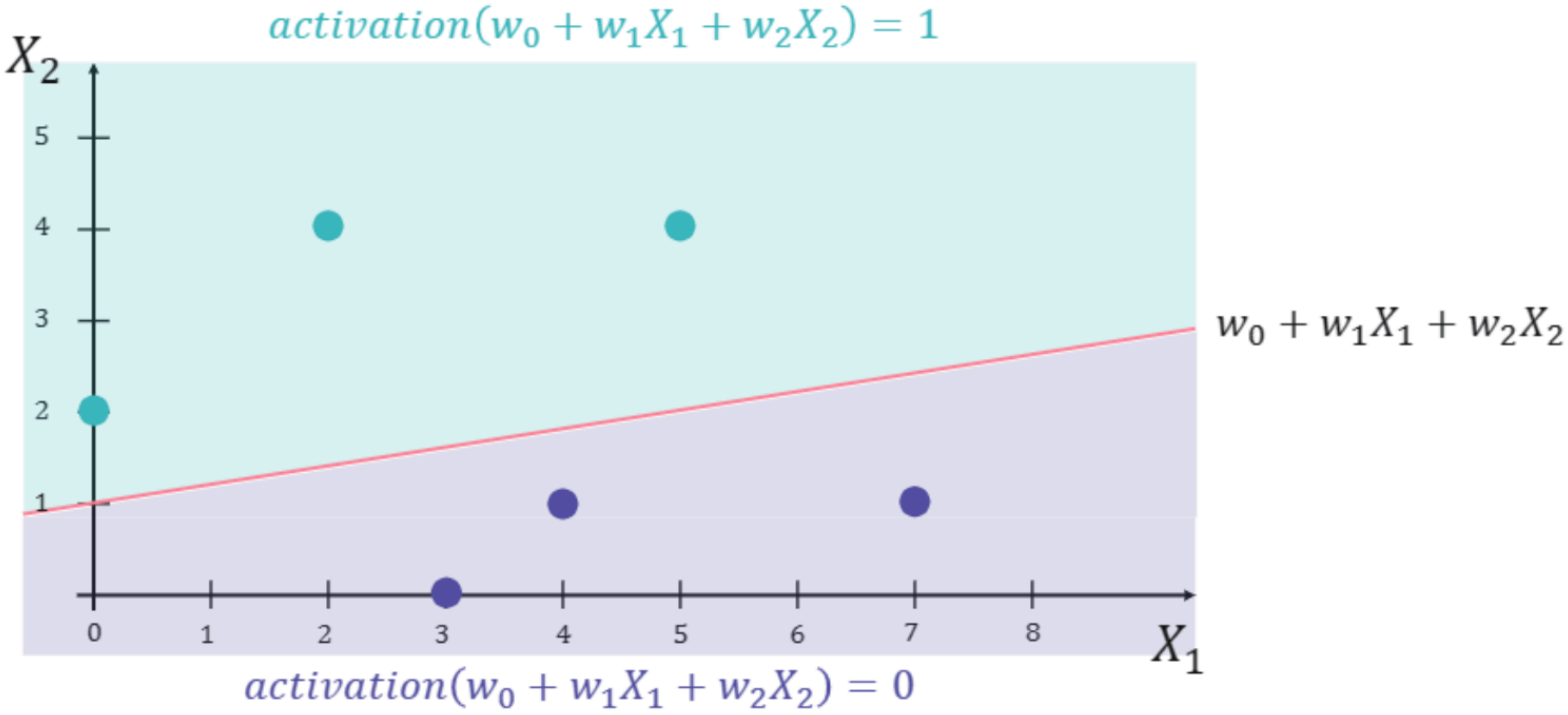


$$y = activation(w_0 + w_1X_1 + w_2X_2)$$

퍼셉트론 동작 예시

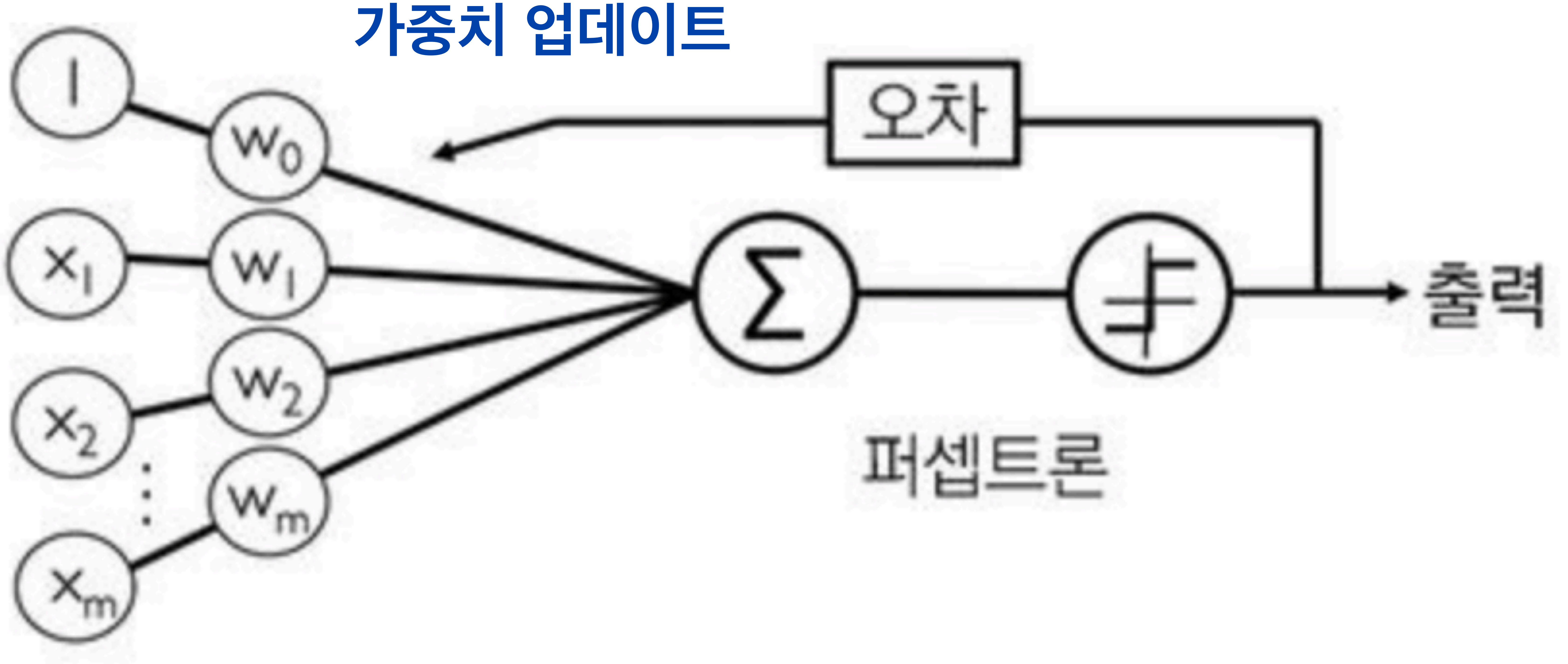
$w_0: -5, \quad w_1: -1, \quad w_2: 5$

X_1	X_2	$w_0 + w_1X_1 + w_2X_2$	예측 Y	Y
2	4	-5+(-2)+20=13	1	1
5	4	-5+(-5)+20=10	1	1
7	1	-5+(-7)+5=-7	0	0
3	0	-5+(-3)+0=-8	0	0
0	2	-5+0+10=5	1	1
4	1	-5+(-4)+5=-4	0	0
⋮	⋮	⋮	⋮	⋮



초기 퍼셉트론 규칙

가중치 업데이트



가중치를 결정하는 방법은 다양하다!

머신러닝 방법론: 데이터를 학습하며 모델이 스스로 적절한 가중치를 찾아나간다! → 학습

초기 퍼셉트론 규칙

1. 가중치를 0 또는 랜덤한 작은 값으로 초기화
2. 각 훈련 샘플 $x^{(i)}$에서 다음을 계산
   - 출력값 계산
   - 가중치 업데이트

$$w = \begin{bmatrix} w_1 \\ \vdots \\ w_m \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix}$$

$$z = w^T x$$

1. 각샘플 $x^{(i)}$ 에 대해 $\hat{y}^{(i)}$ 계산

가중치를 일괄업데이트!

2. 가중치 업데이트 $w_j := w_j + \Delta w_j$

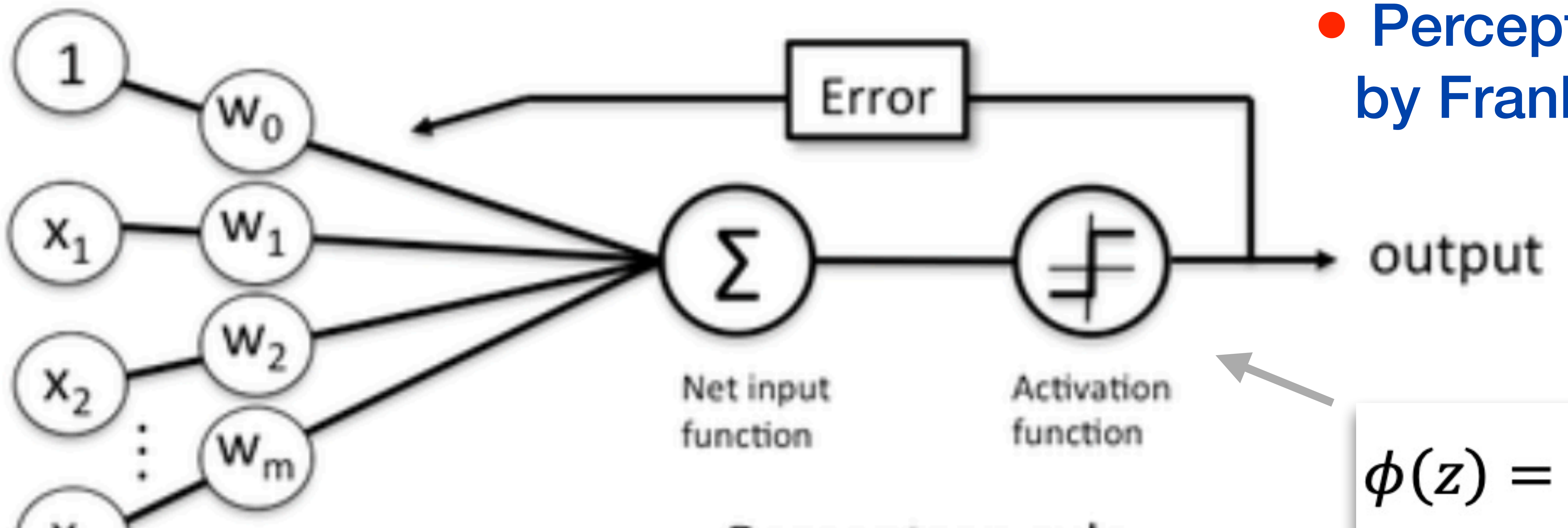
$$\Delta w_j = \eta \left(y^{(i)} - \hat{y}^{(i)} \right) x_j^{(i)}$$

학습률 0.0 ~ 1.0 사이의 실수

true class label

predicted class label

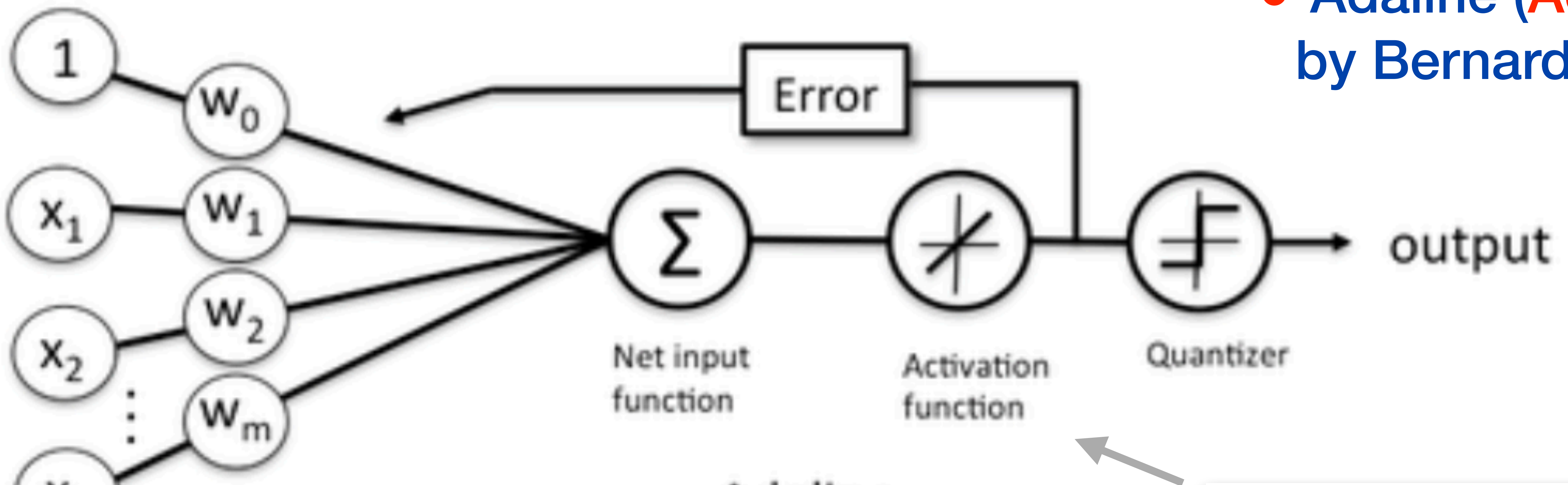
Adaline



Perceptron rule.

- Perceptron - first artificial neuron (1957) by Frank Rosenblatt

$$\phi(z) = \begin{cases} 1 & \text{if } z \geq 0, \\ -1 & \text{otherwise.} \end{cases}$$

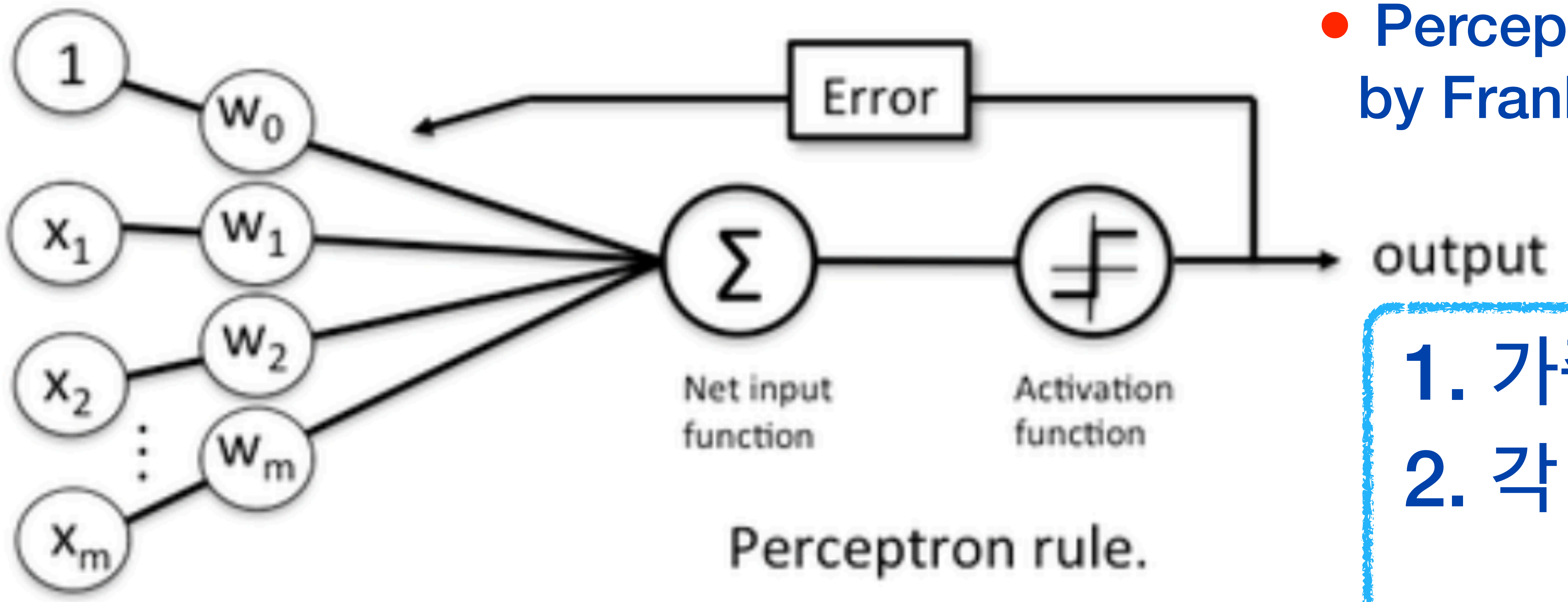


Adaline.

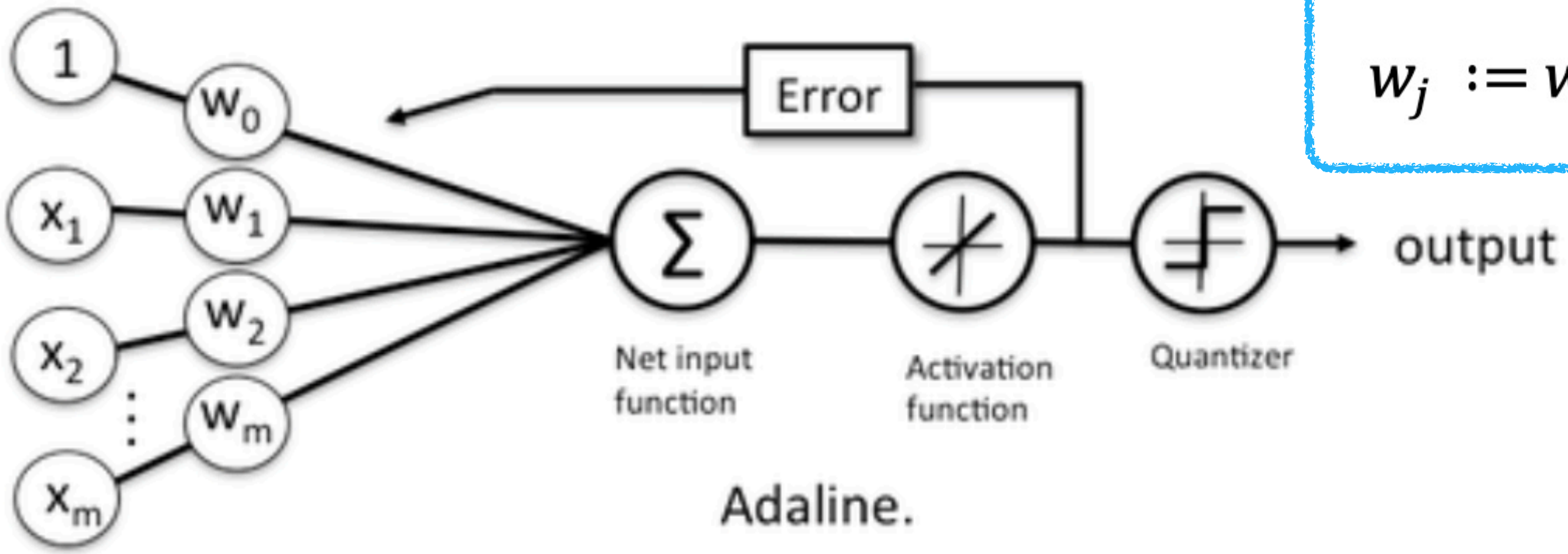
- Adaline (Adaptive linear neuron) (1960) by Bernard Widrow and Tedd Hoff

$$\phi(z) = z = a$$

Perceptron vs. Adaline



Perceptron rule.



Adaline.

- Perceptron - first artificial neuron (1957) by Frank Rosenblatt

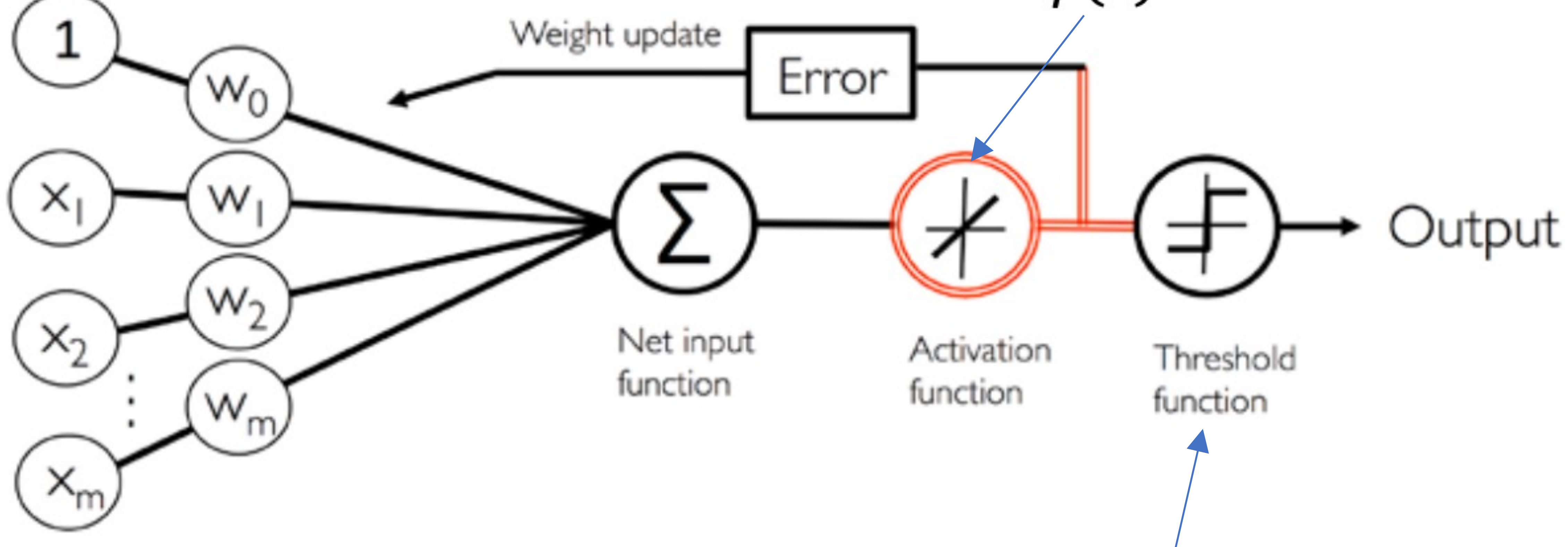
1. 가중치를 0 또는 랜덤한 작은 값으로 초기화
2. 각 훈련 샘플 $x^{(i)}$에서 다음을 계산
   - 출력값 계산
   - 가중치 업데이트

$$w_j := w_j + \Delta w_j \qquad \Delta w_j = \eta\left(y^{(i)} - \hat{y}^{(i)}\right)x_j^{(i)}$$

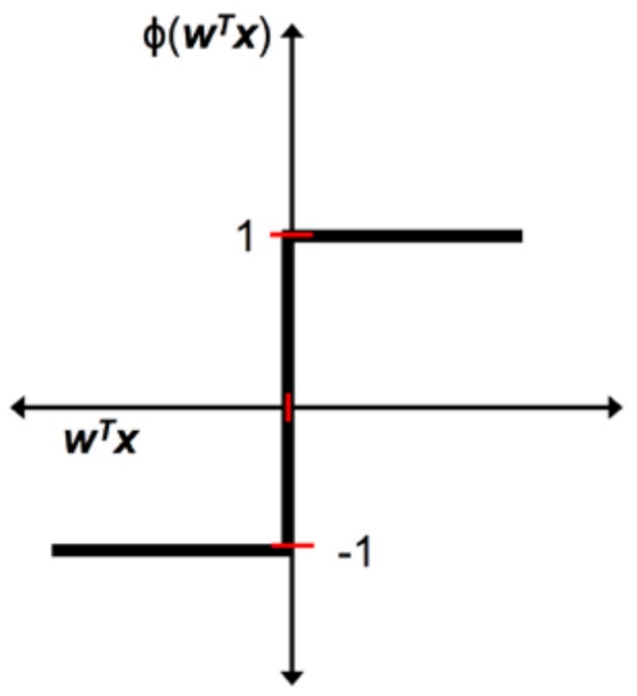
Adaline

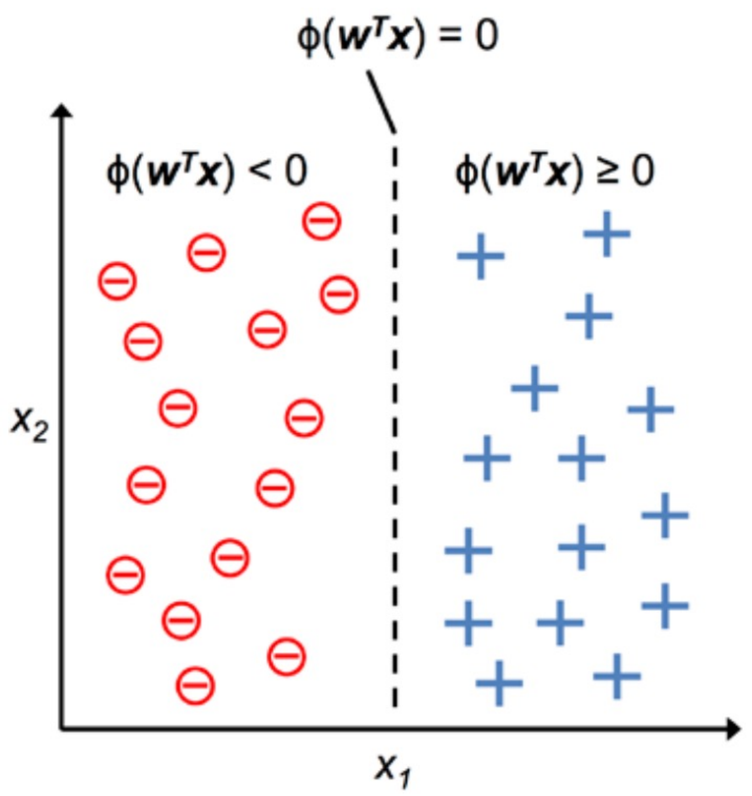
$$z = \sum_j w_j\, x_j = \boldsymbol{w}^T \boldsymbol{x} \qquad w_j := w_j + \Delta w_j$$

$$\phi(z) = z = a$$



$$\phi(z) = \begin{cases} 1 & \text{if } z \geq 0, \\ -1 & \text{otherwise.} \end{cases}$$





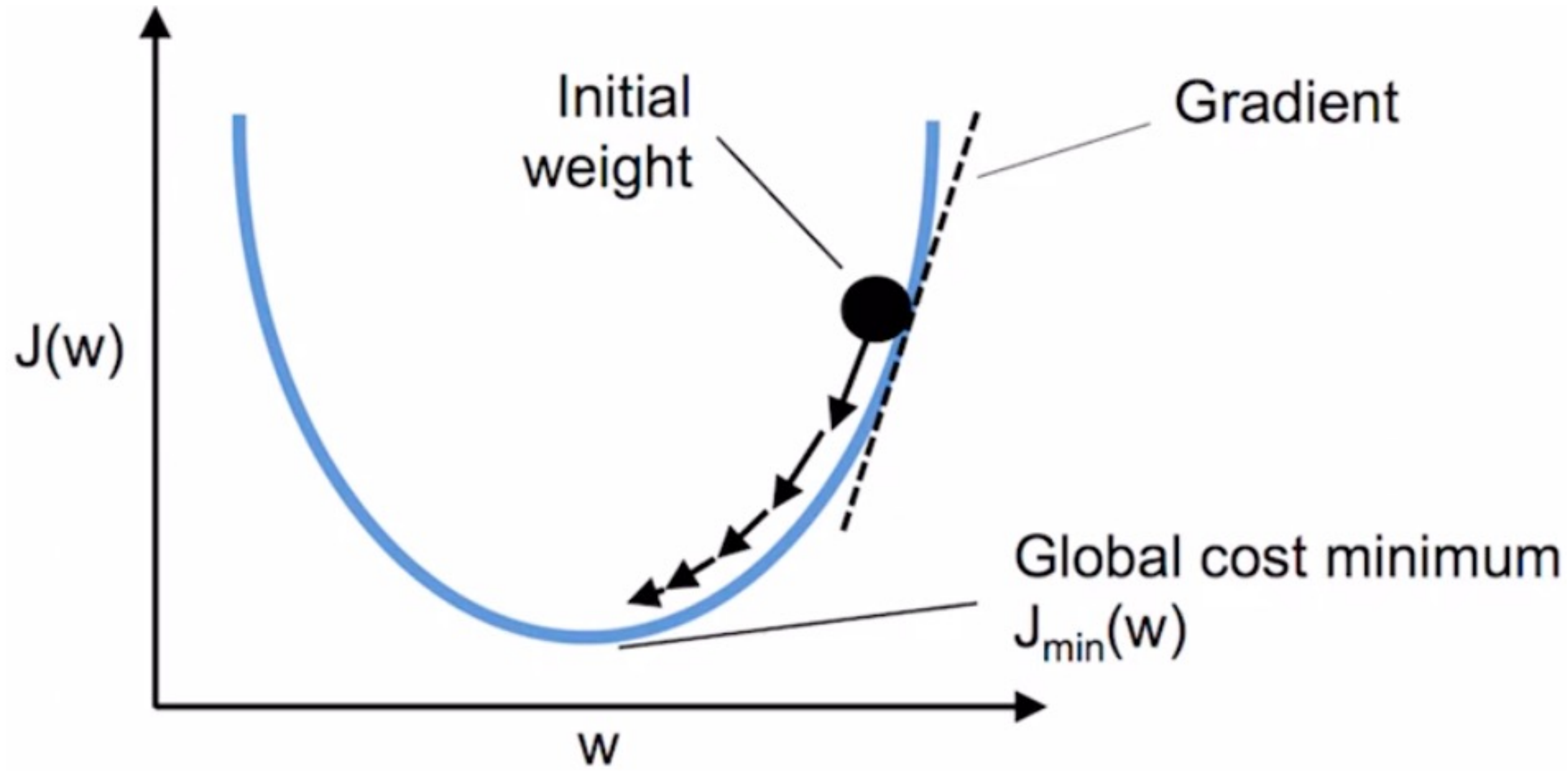
- Cost function = SSE (sum of the squared error)

$$J(\boldsymbol{w}) = \frac{1}{2}\sum_i \underbrace{\left(y^{(i)} - \phi\left(z^{(i)}\right)\right)^2}_{= \text{ Loss function}}$$

- Weight update

$$\boldsymbol{w} := \boldsymbol{w} + \Delta\boldsymbol{w}, \qquad \text{where } \Delta\boldsymbol{w} = -\eta\nabla J(\boldsymbol{w})$$

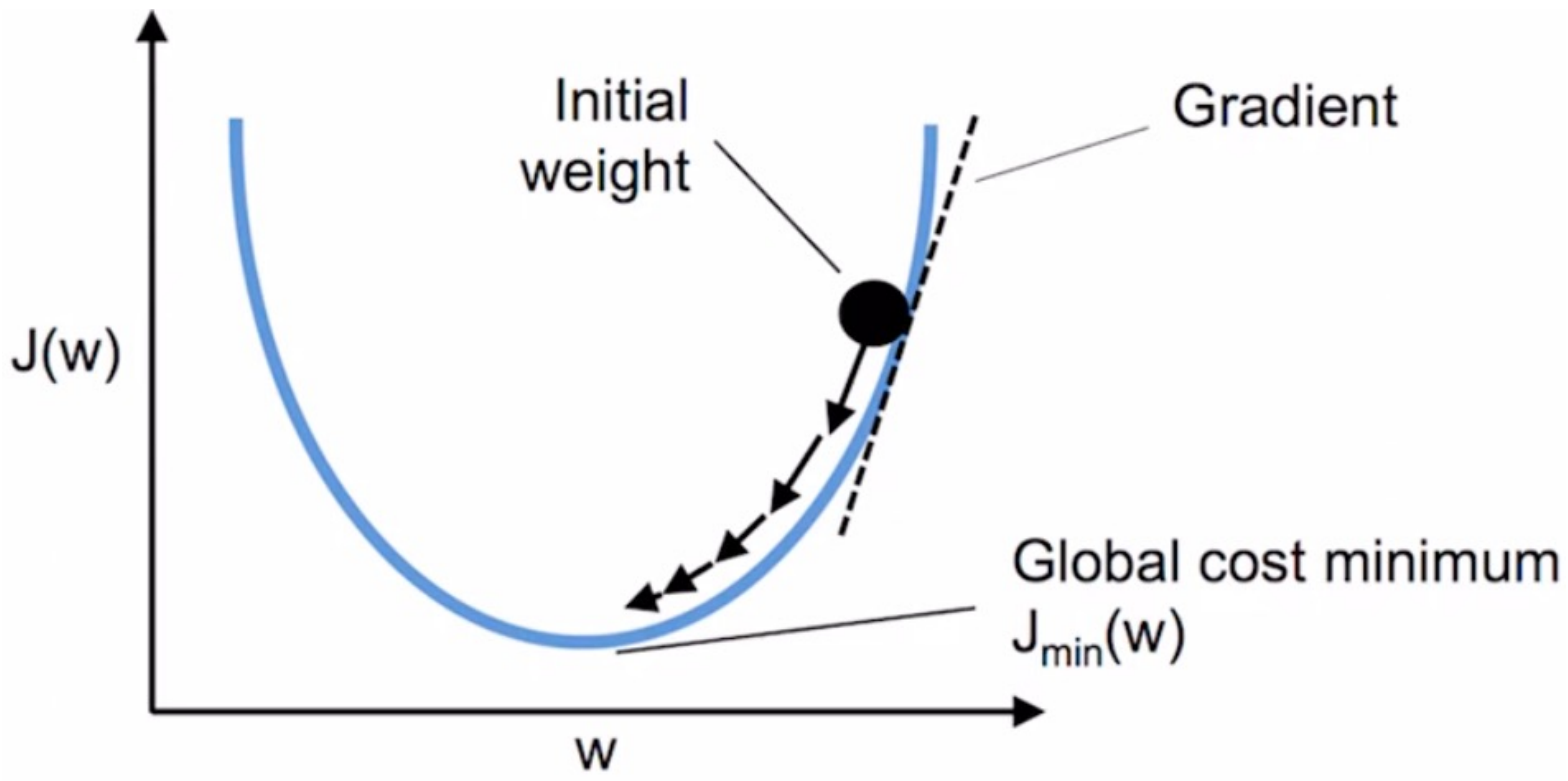
$$\frac{\partial}{\partial w_j} J(\boldsymbol{w}) = -\sum_i \left(y^{(i)} - a^{(i)}\right) x_j^{(i)}$$



Adaline

$$\boldsymbol{w} := \boldsymbol{w} + \Delta\boldsymbol{w}, \qquad \text{where } \Delta\boldsymbol{w} = -\eta\nabla J(\boldsymbol{w})$$

$$\frac{\partial}{\partial w_j} J(\boldsymbol{w}) = -\sum_i \left(y^{(i)} - a^{(i)}\right) x_j^{(i)}$$



$$\frac{\partial J}{\partial w_j} = \frac{\partial}{\partial w_j} \frac{1}{2} \sum_i \left(y^{(i)} - \phi(z^{(i)})\right)^2$$

$$= \frac{1}{2} \frac{\partial}{\partial w_j} \sum_i \left(y^{(i)} - \phi(z^{(i)})\right)^2$$

$$= \frac{1}{2} \sum_i 2\left(y^{(i)} - \phi(z^{(i)})\right) \frac{\partial}{\partial w_j} \left(y^{(i)} - \phi(z^{(i)})\right)$$

$$= \sum_i \left(y^{(i)} - \phi(z^{(i)})\right) \frac{\partial}{\partial w_j} \left(y^{(i)} - \sum_i \left(w_j^{(i)} x_j^{(i)}\right)\right)$$

$$= \sum_i \left(y^{(i)} - \phi(z^{(i)})\right)\left(-x_j^{(i)}\right)$$

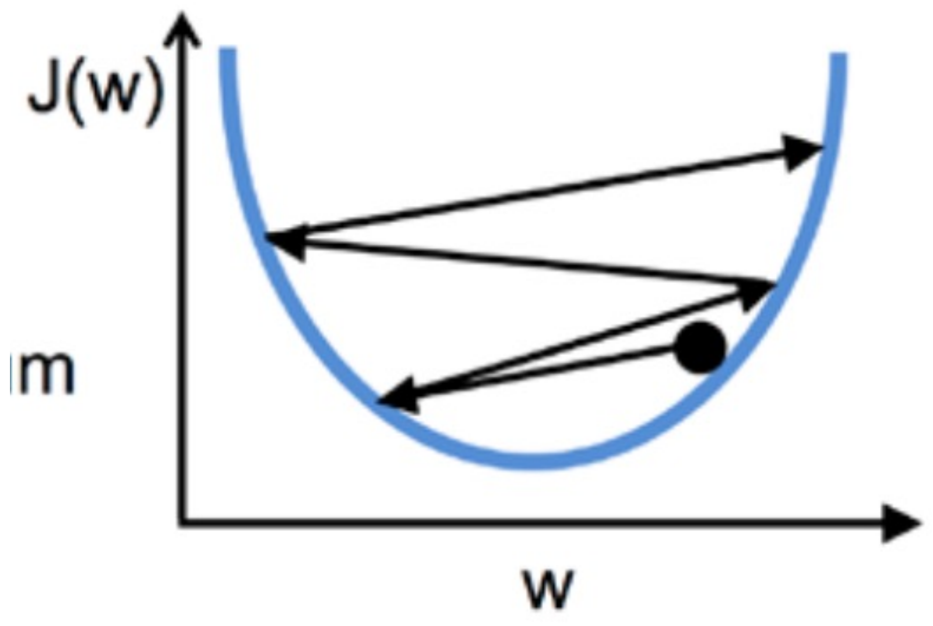
$$= -\sum_i \left(y^{(i)} - \phi(z^{(i)})\right) x_j^{(i)}$$

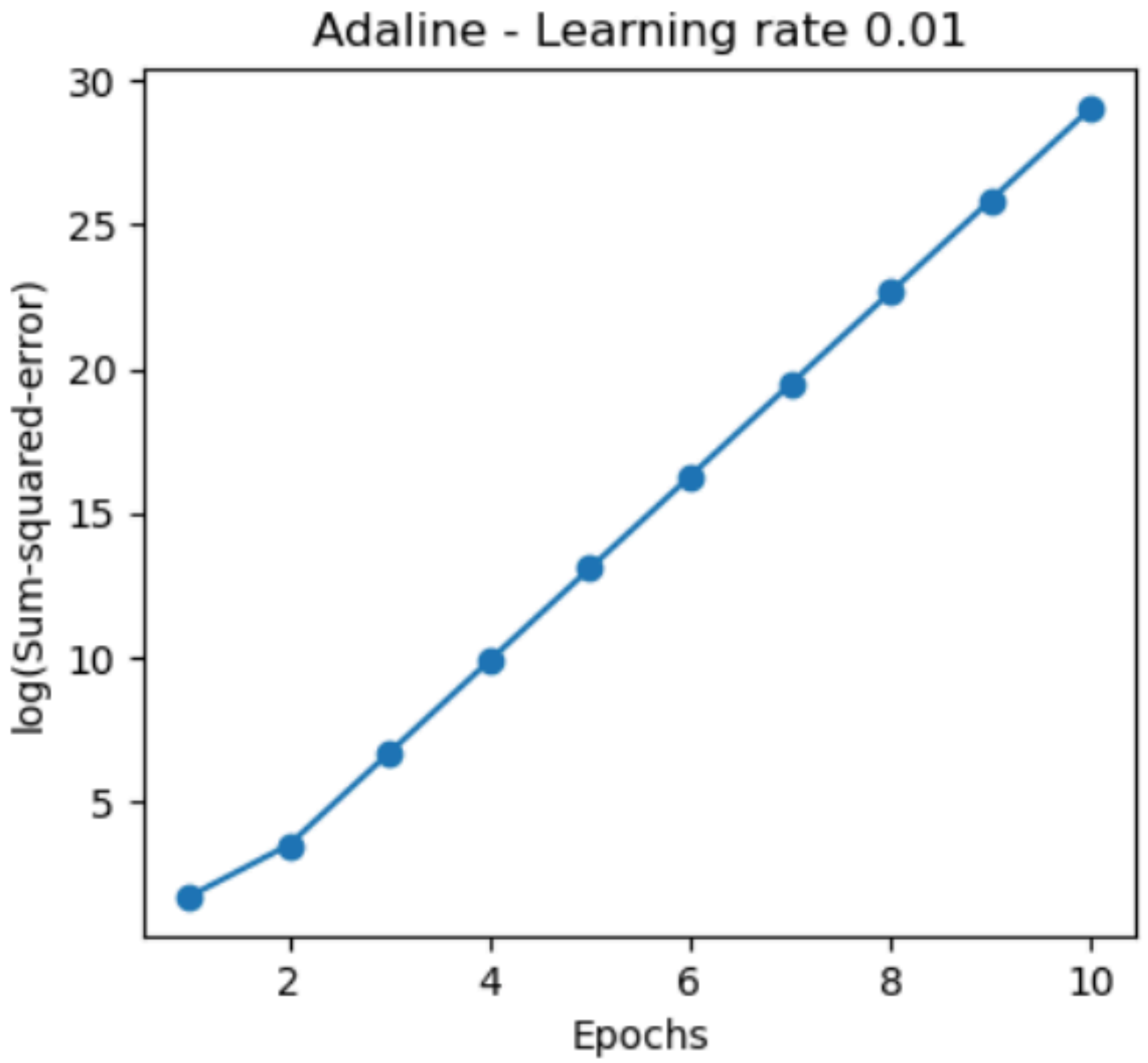
학습률의 중요성

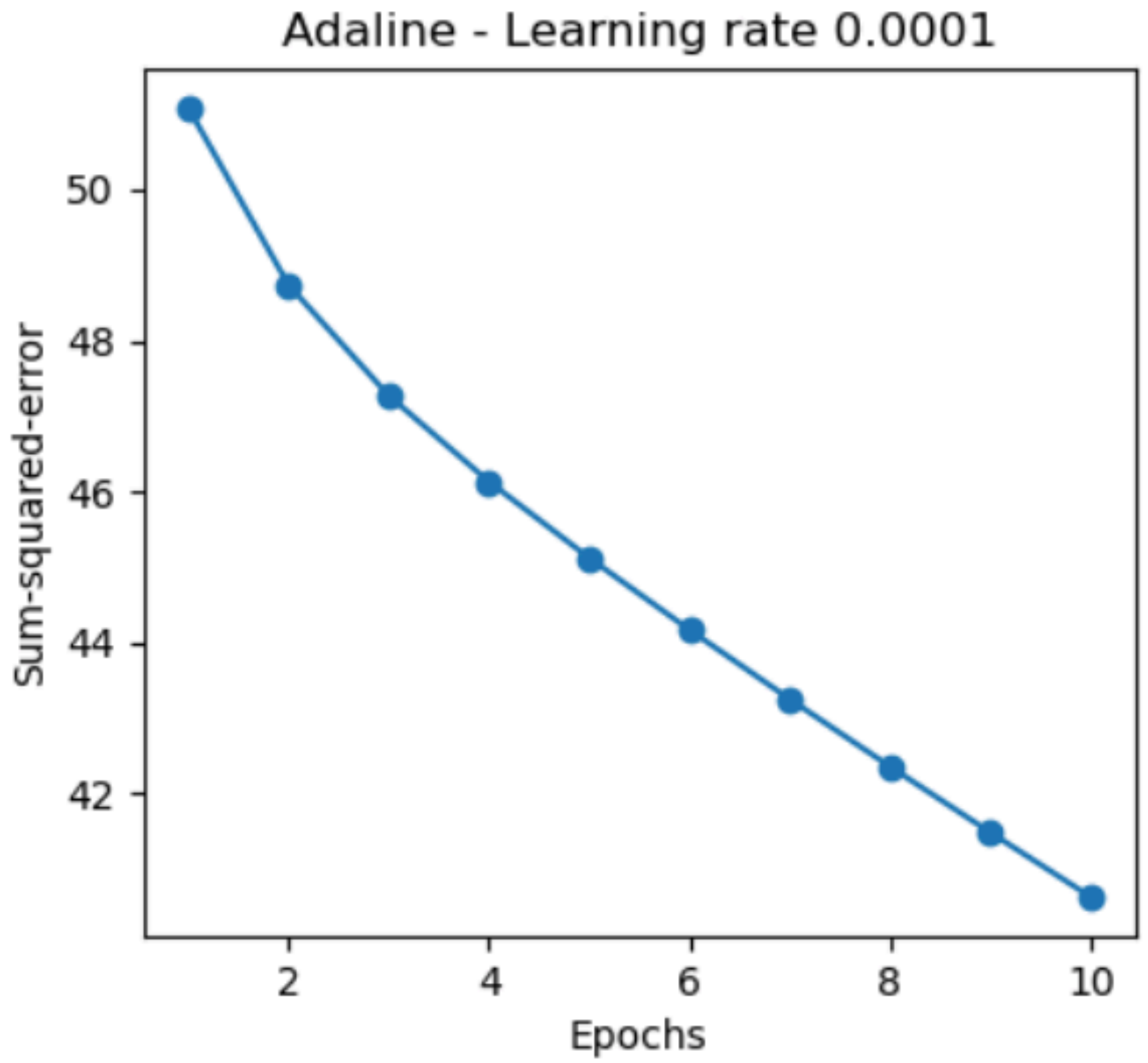
$$\boldsymbol{w} := \boldsymbol{w} + \Delta\boldsymbol{w}, \quad \text{where } \Delta\boldsymbol{w} = -\eta \nabla J(\boldsymbol{w})$$

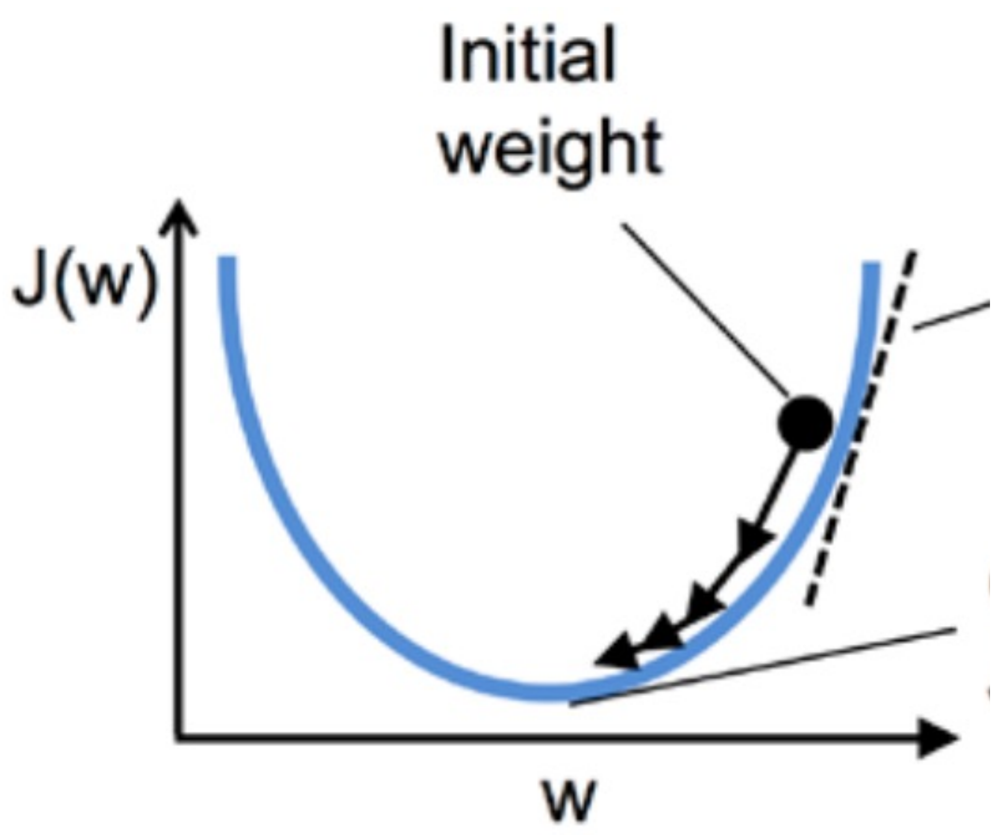
η : hyperparameter (set *by manually*)

$\boldsymbol{w}$: model parameter (obtained *by training*)









확률적 경사 하강법 (Stochastic Gradient Decent)

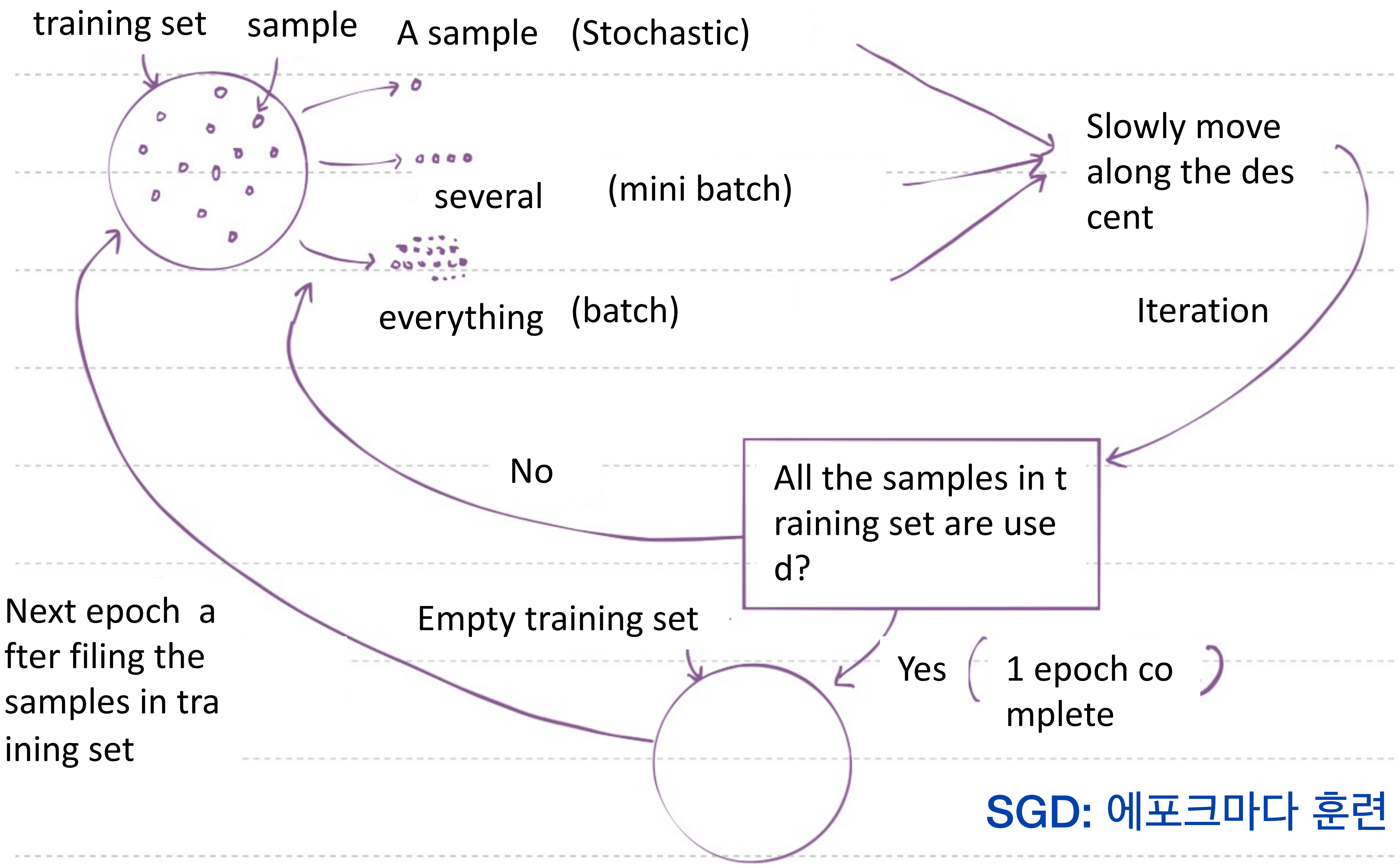
- Stochastic gradient decent (SGD)

$$\Delta \boldsymbol{w} = \eta \left(y^{(i)} - \phi\left(z^{(i)}\right) \right) \boldsymbol{x}^{(i)}$$

수렴 속도가 훨씬 빠르다!

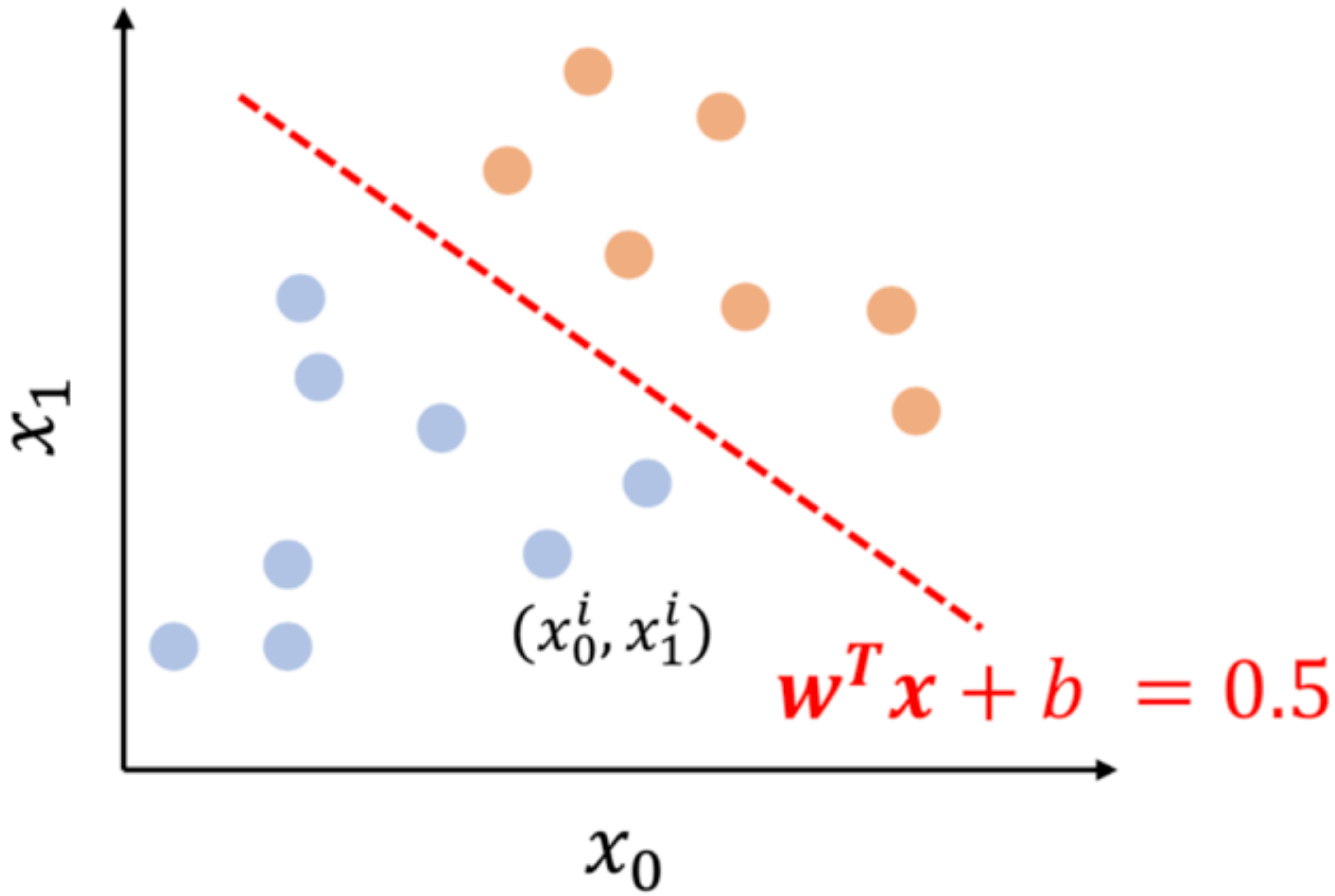
- Batch gradient decent (original)

$$\Delta \boldsymbol{w} = \eta \sum_i \left(y^{(i)} - \phi\left(z^{(i)}\right) \right) \boldsymbol{x}^{(i)}$$

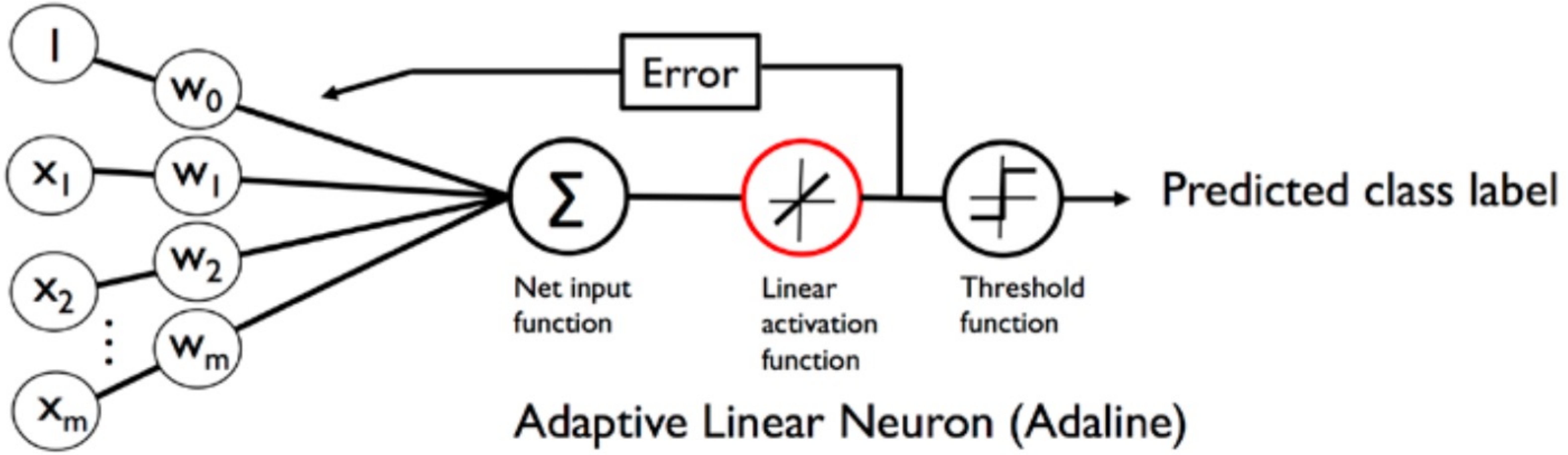


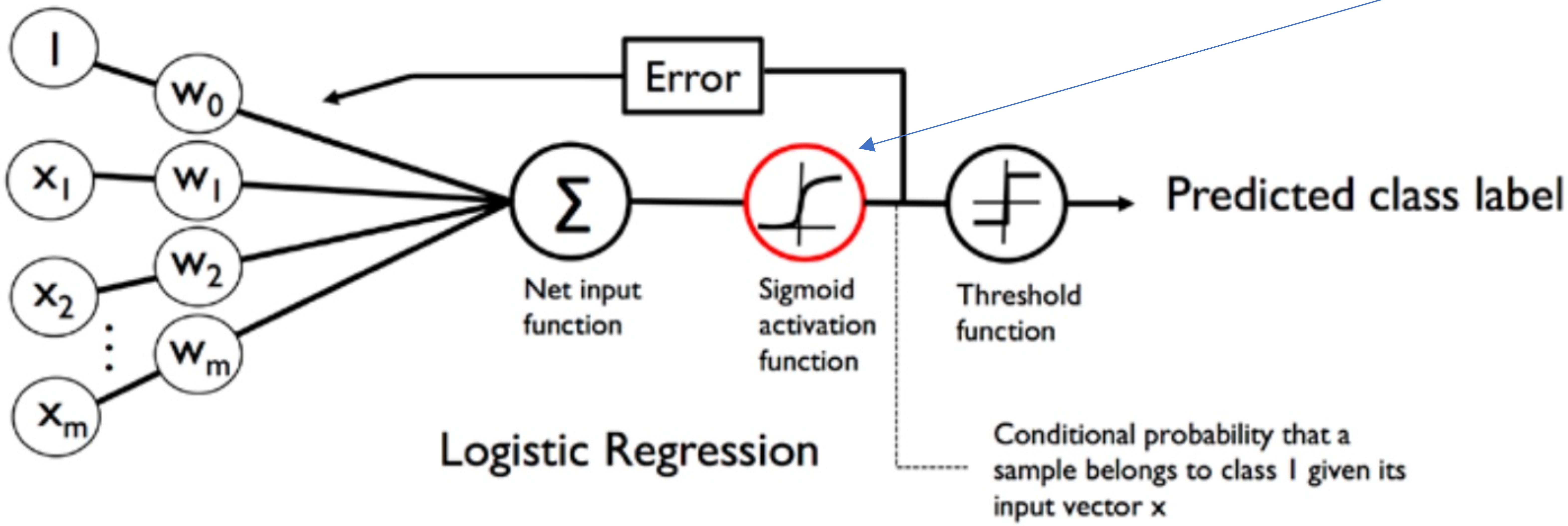
SGD: 에포크마다 훈련 데이터를 섞는 것이 좋다!

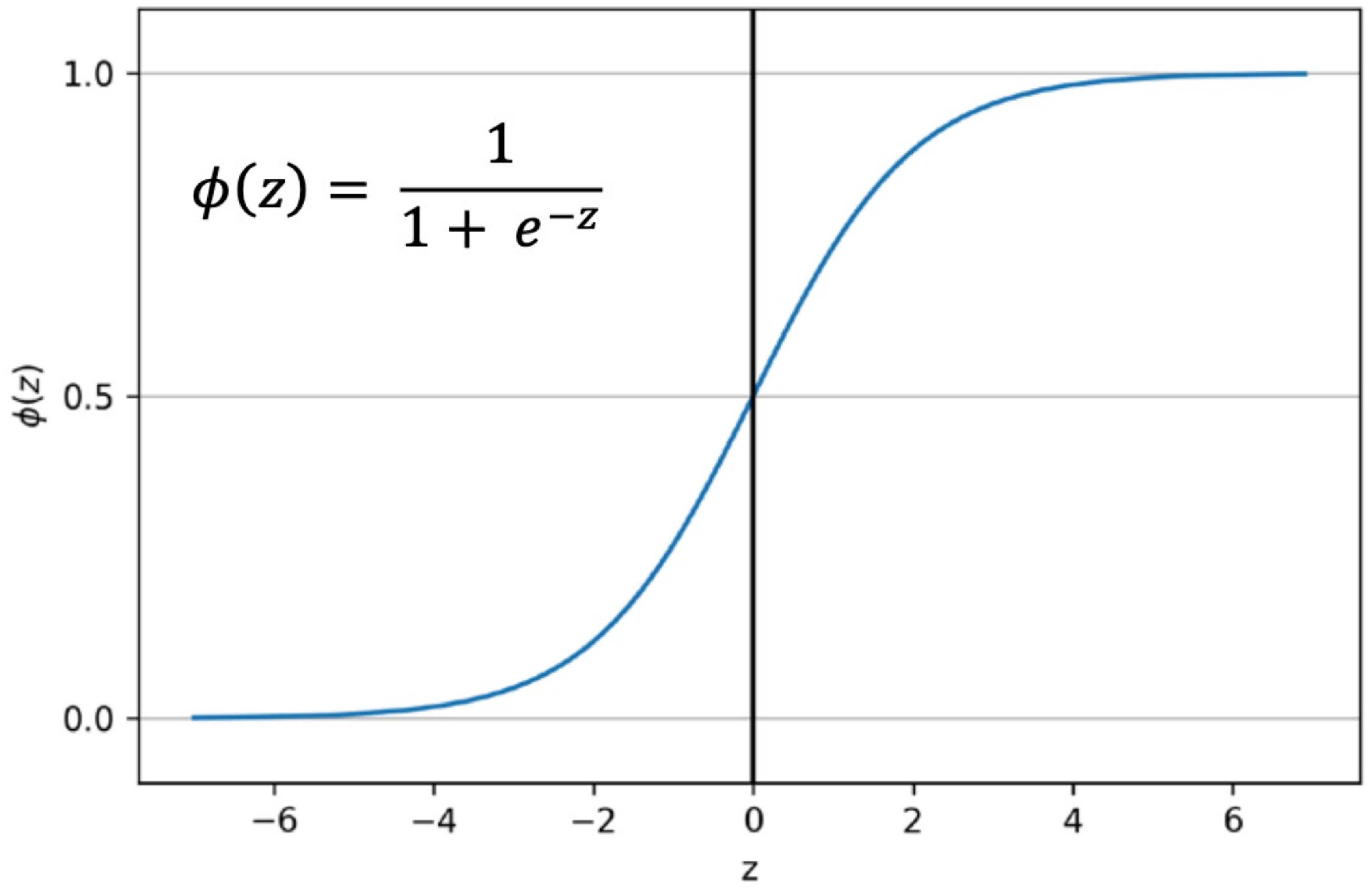
로지스틱 회귀 (Logistic regression) – 이진 분류를 위한 선형 모델



로지스틱 회귀 (Logistic regression) – 이진 분류를 위한 선형 모델



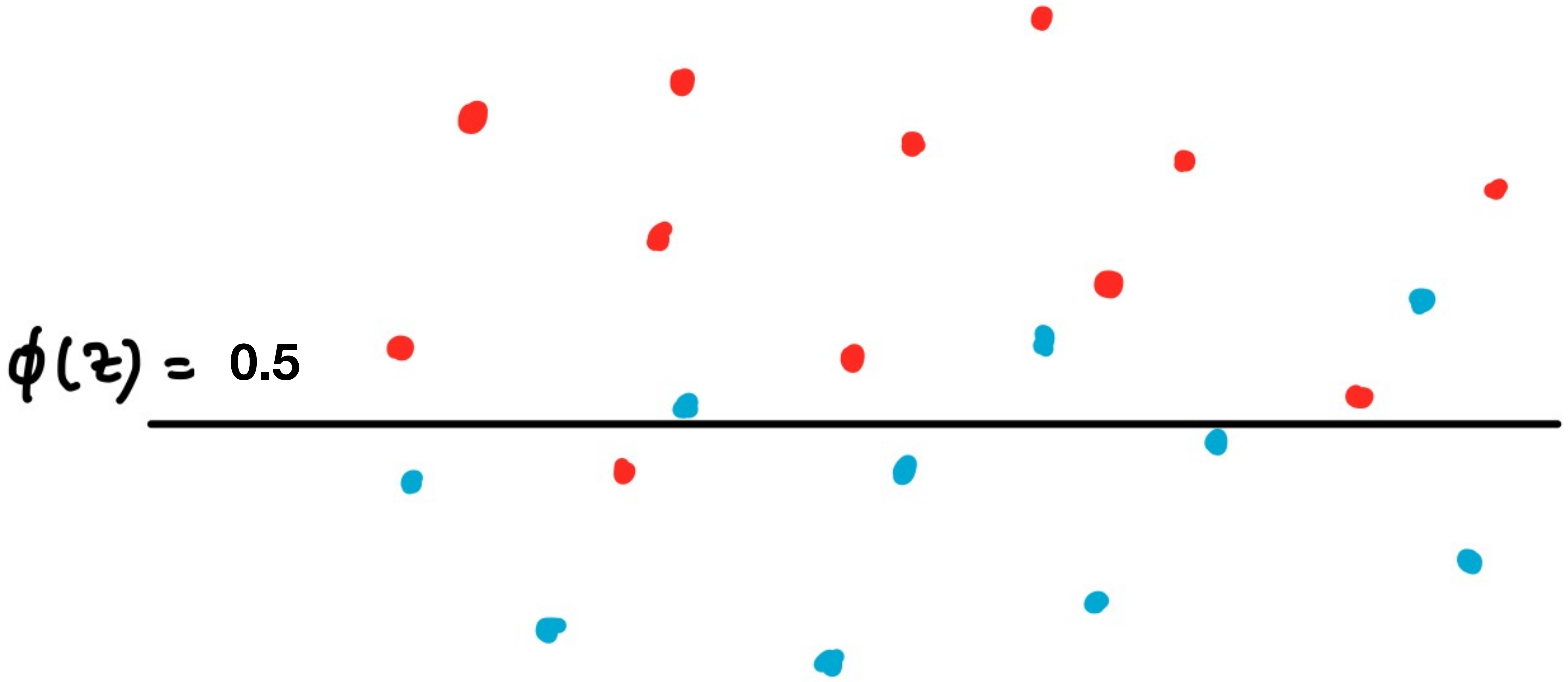


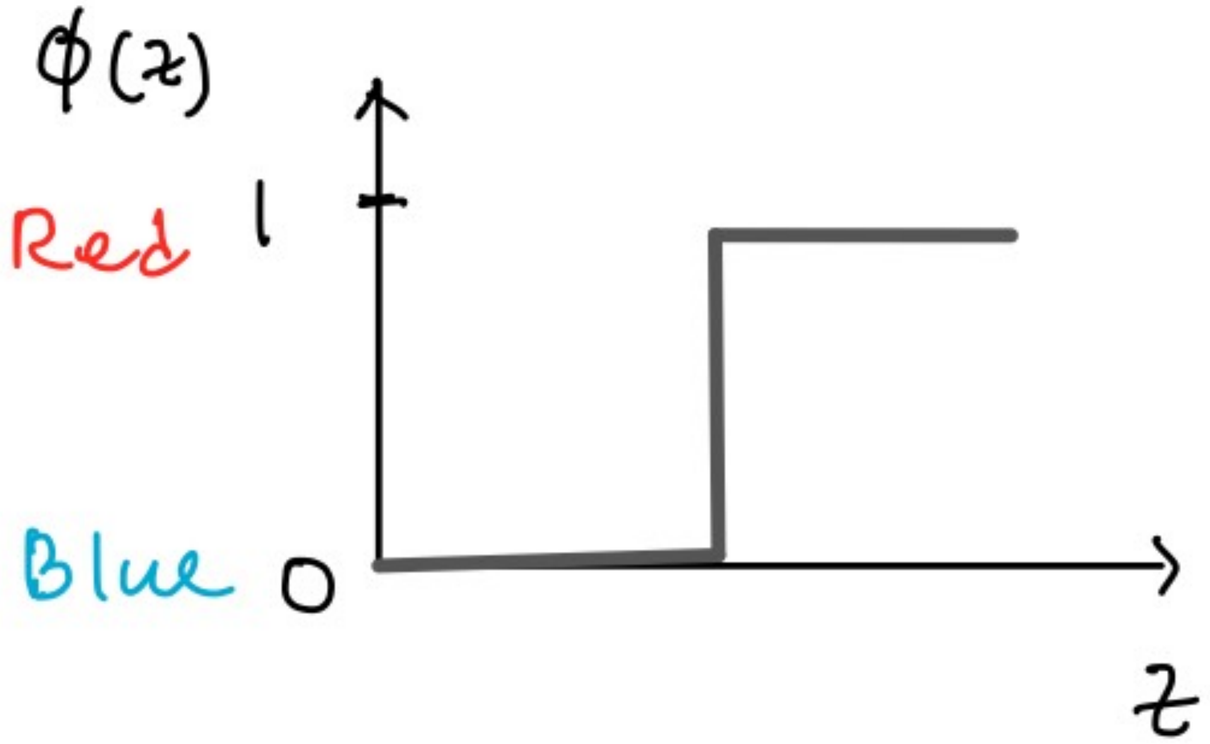


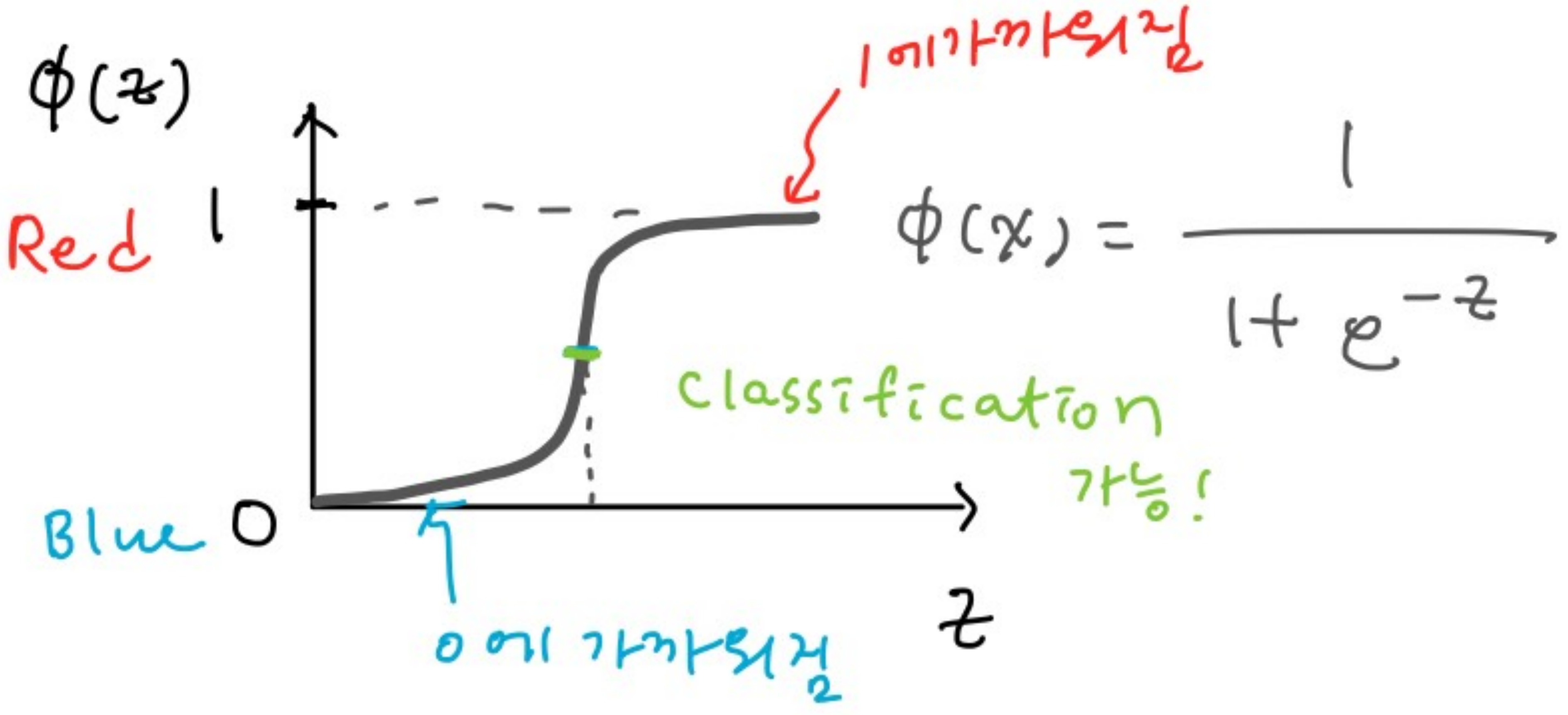
$$\hat{y} = \begin{cases} 1 & \text{if } \phi(z) \geq 0.5 \\ 0 & \text{otherwise} \end{cases}$$

$$\hat{y} = \begin{cases} 1 & \text{if } z \geq 0.0 \\ 0 & \text{otherwise} \end{cases}$$

로지스틱 회귀 (Logistic regression) – 이진 분류를 위한 선형 모델







미분가능 !

로지스틱 회귀 (Logistic regression) – 이진 분류를 위한 선형 모델

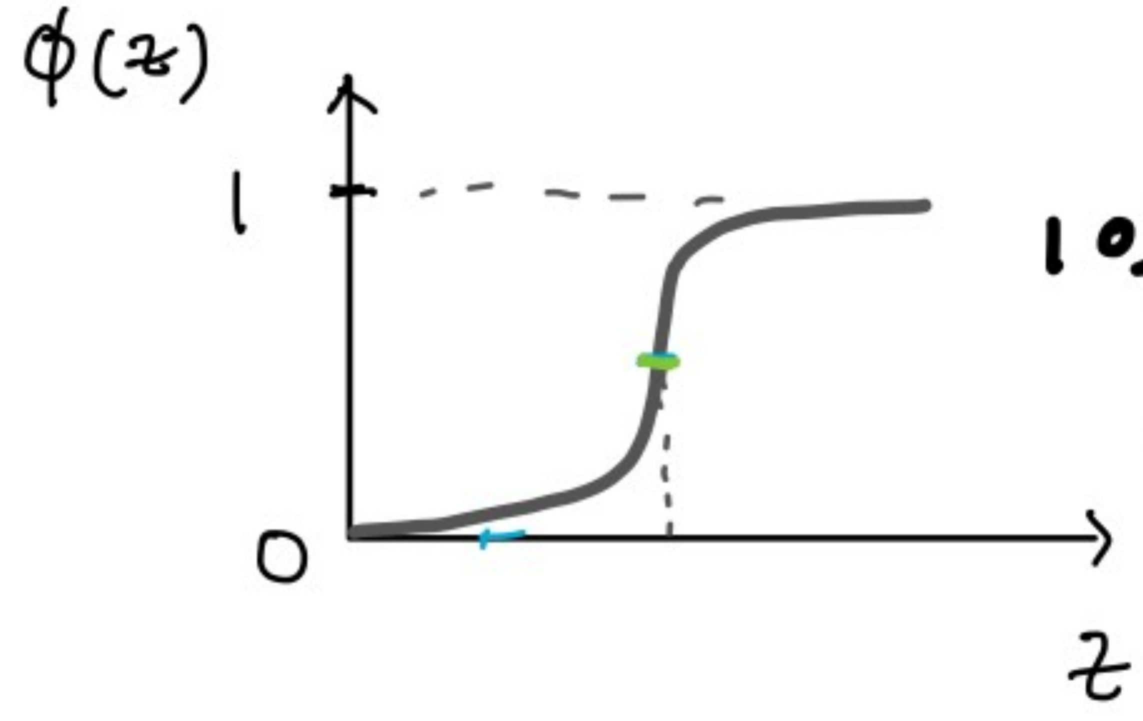
w를 찾아서 . . .

1) Loss function 정의 → 최소화하는 w 찾기

2) 결과가 truth가 될 확률 정의 → 확률 최대화 하는 w 찾기

$$P(y \mid x; w) = \begin{cases} \phi(z) & \text{if } y=1 \\ 1-\phi(z) & \text{if } y=0 \end{cases}$$

← Likelihood



1일 확률 0.7

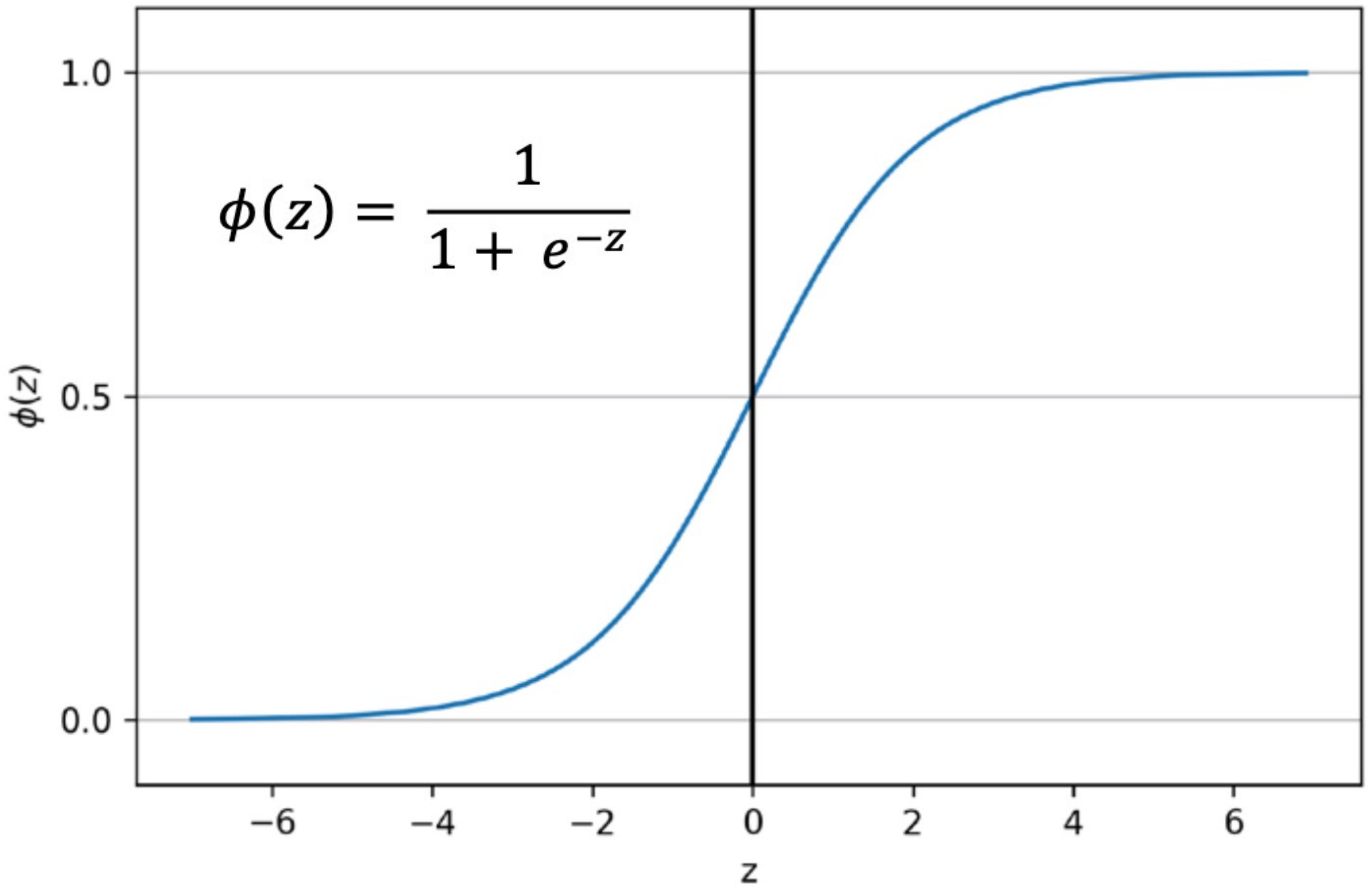
⟺ 0일 확률 = 1−0.7

Likelihood가 최대가 되는 w 찾기!

Why sigmoid function?

Odds (odds in favor of a particular event) = $\dfrac{p}{(1-p)}$ p : the probability of the positive event

$$logit(p) = \log\frac{p}{(1-p)}$$



$$0 \le p \le 1 \;\rightarrow 0 \le \frac{p}{1-p} \le \infty \rightarrow -\infty \le log\,\frac{p}{1-p} \le \infty$$

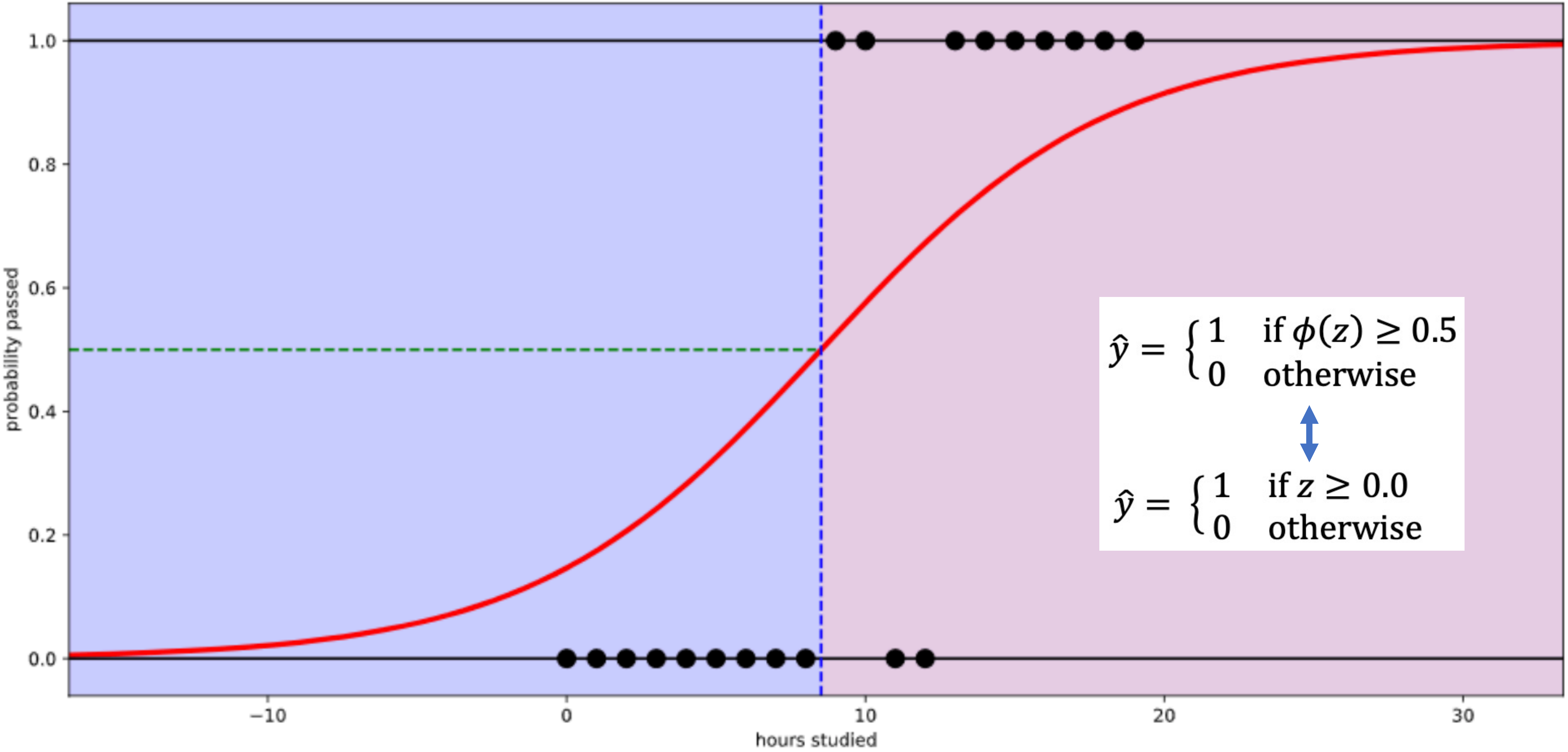
Therefore, one could say that

$$logit\big(p(y=1|\boldsymbol{x})\big) = w_0x_0 + \; w_1x_1 + \cdots + w_mx_m = \sum_{i=0}^{m} w_ix_i = \boldsymbol{w}^T\boldsymbol{x}$$

Here, $p(y=1|\boldsymbol{x})$ is the conditional probability that a particular example belongs to class 1 given its features, $\boldsymbol{x}$.

$$p = \frac{1}{1+e^{-w^Tx}} = \frac{1}{1+e^{-z}}$$

Why sigmoid function? – example



$$\hat{y} = \begin{cases} 1 & \text{if } \phi(z) \geq 0.5 \\ 0 & \text{otherwise} \end{cases}$$

$\Updownarrow$

$$\hat{y} = \begin{cases} 1 & \text{if } z \geq 0.0 \\ 0 & \text{otherwise} \end{cases}$$

로지스틱 비용 함수의 가중치 학습

Likelihood

$$L(\boldsymbol{w}) = P(\boldsymbol{y} \mid \boldsymbol{x}; \boldsymbol{w}) = \prod_{i=1}^{n} P(y^{(i)} \mid x^{(i)}; \boldsymbol{w}) = \prod_{i=1}^{n} \left(\phi(z^{(i)})\right)^{y^{(i)}} \left(1 - \phi(z^{(i)})\right)^{1-y^{(i)}}$$

Log-likelihood

$$l(\boldsymbol{w}) = \log L(\boldsymbol{w}) = \sum_{i=1}^{n} \left[y^{(i)} \log\left(\phi(z^{(i)})\right) + (1 - y^{(i)}) \log\left(1 - \phi(z^{(i)})\right)\right]$$

Negative log-likelihood as a cost function

$$\mathrm{J}(\mathbf{w}) = \sum_{i=1}^{n} \left[-y^{(i)} \log\left(\phi(z^{(i)})\right) - (1 - y^{(i)}) \log\left(1 - \phi(z^{(i)})\right)\right]$$

Optimization = minimize negative log-likelihood (maximize likelihood)

로지스틱 비용 함수의 가중치 학습

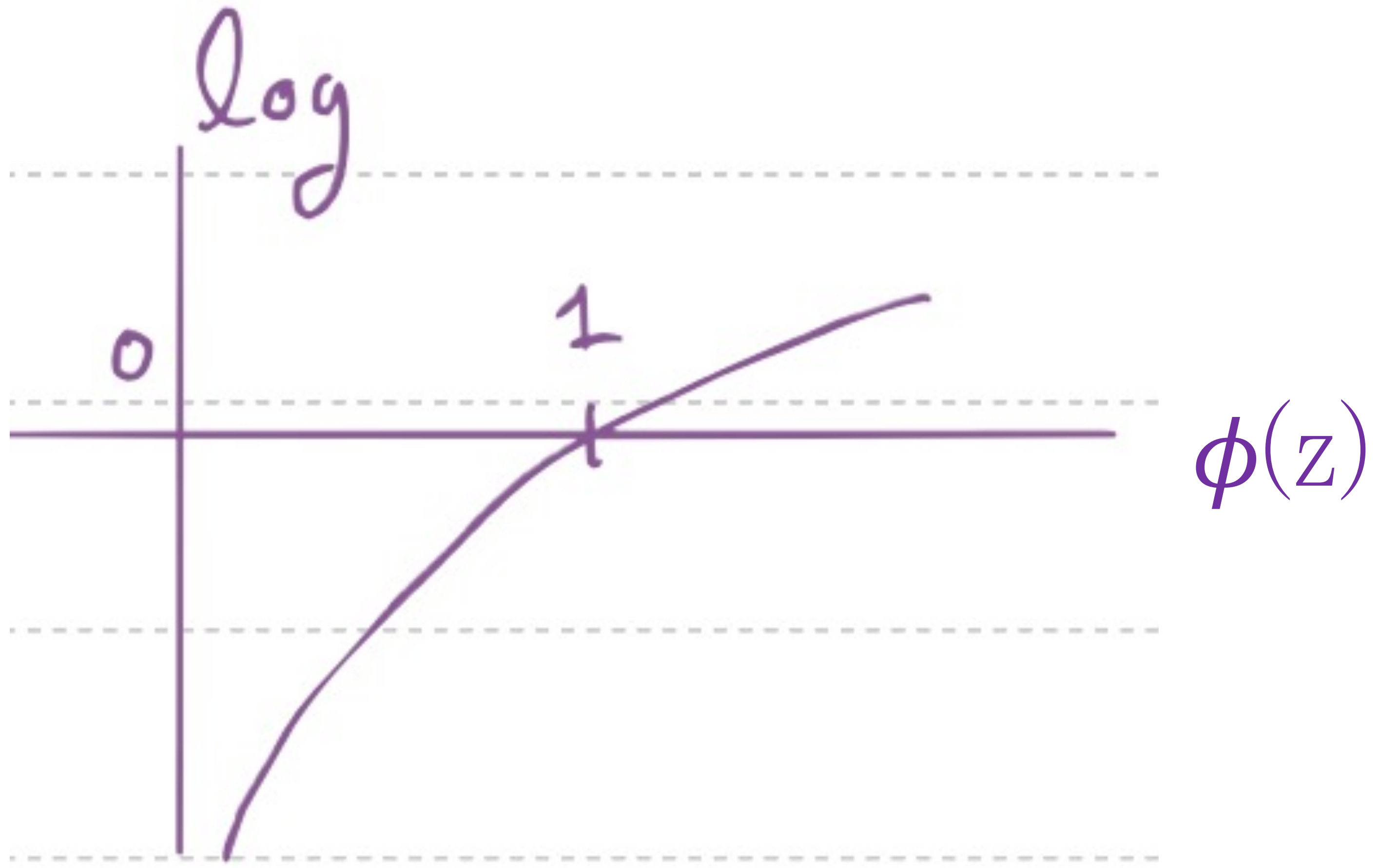
$$J(\mathbf{w}) = \sum_{i=1}^{n}\left[-y^{(i)}\log\left(\phi\left(z^{(i)}\right)\right) - \left(1 - y^{(i)}\right)\log\left(1 - \phi\left(z^{(i)}\right)\right)\right]$$

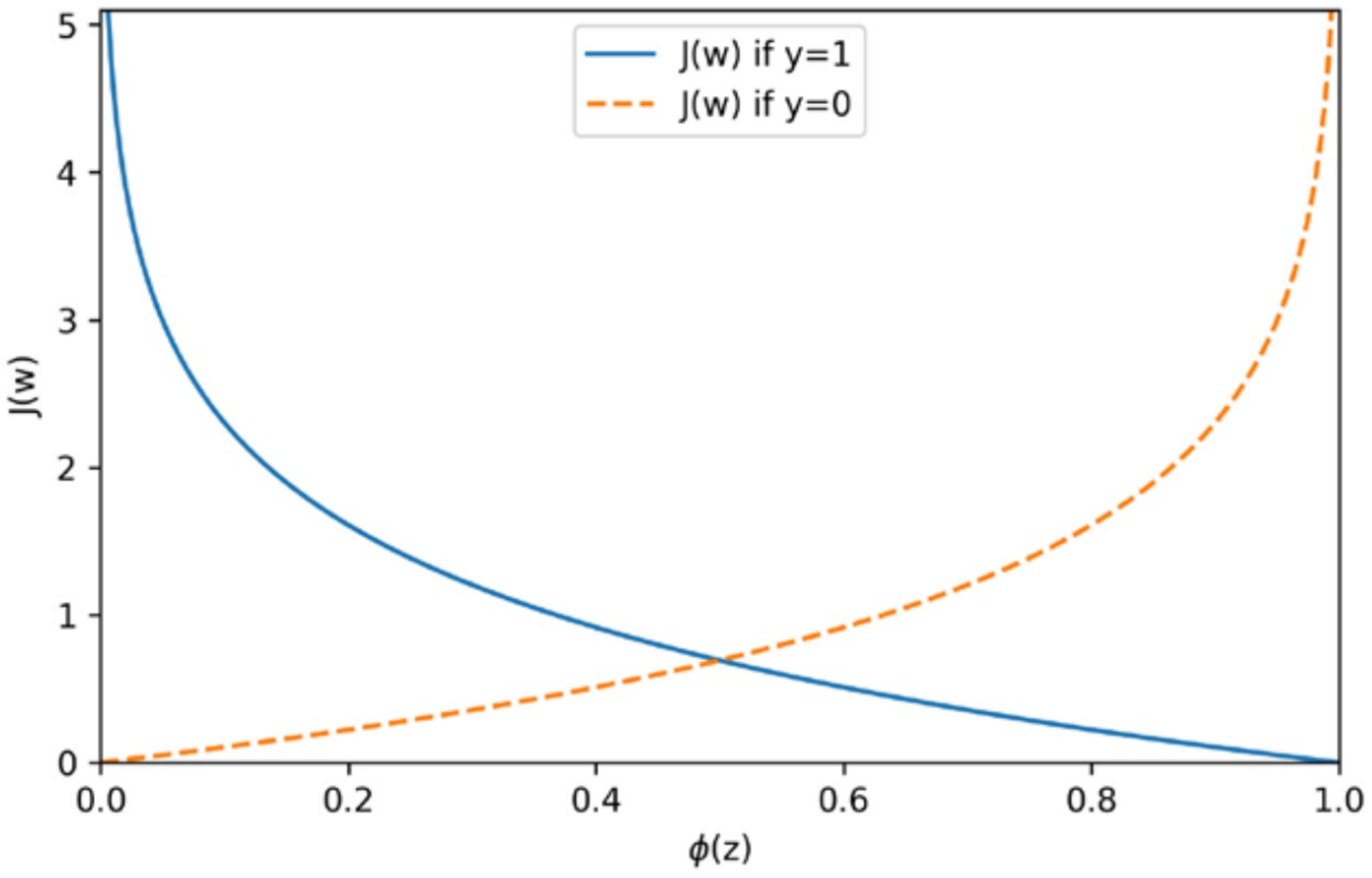
이해를 돕기 위해 샘플이 하나인 경우에 대해 계산해보면

$$J(\phi(z), y; \mathbf{w}) = -y\log(\phi(z)) - (1 - y)\log(1 - \phi(z))$$

$$J(\phi(z), y; \mathbf{w}) = \begin{cases} -\log\left(\phi(z)\right) & \text{if } y = 1 \\ -\log\left(1 - \phi(z)\right) & \text{if } y = 0 \end{cases}$$

Logistic loss function = binary cross-entropy loss function





로지스틱 비용 함수의 가중치 학습

Log-likelihood = $l(\boldsymbol{w}) = \log L(\boldsymbol{w}) = \sum_{i=1}^{n} \left[y^{(i)} \log\left(\phi\left(z^{(i)}\right)\right) + \left(1 - y^{(i)}\right) \log\left(1 - \phi\left(z^{(i)}\right)\right) \right]$

Let's start by calculating the partial derivative of the log-likelihood function with respect to the *j*th weight:

$$\frac{\partial}{\partial w_j} l(\boldsymbol{w}) = \left(y \frac{1}{\phi(z)} - (1 - y) \frac{1}{1 - \phi(z)} \right) \frac{\partial}{\partial w_j} \phi(z)$$

Before we continue, let's also calculate the partial derivative of the sigmoid function:

$$\frac{\partial}{\partial z} \phi(z) = \frac{\partial}{\partial z} \frac{1}{1 + e^{-z}} = \frac{1}{(1 + e^{-z})^2} e^{-z} = \frac{1}{1 + e^{-z}} \left(1 - \frac{1}{1 + e^{-z}}\right) = \phi(z)\left(1 - \phi(z)\right)$$

Now, we can resubstitute $\frac{\partial}{\partial z} \phi(z) = \phi(z)\left(1 - \phi(z)\right)$ in our first equation to obtain the following:

$$\begin{aligned} \left(y \frac{1}{\phi(z)} - (1 - y) \frac{1}{1 - \phi(z)} \right) \frac{\partial}{\partial w_j} \phi(z) &= \left(y \frac{1}{\phi(z)} - (1 - y) \frac{1}{1 - \phi(z)} \right) \phi(z)\left(1 - \phi(z)\right) \frac{\partial}{\partial w_j} z \\ &= \left(y\left(1 - \phi(z)\right) - (1 - y)\phi(z) \right) x_j \\ &= \left(y - \phi(z)\right) x_j \end{aligned}$$

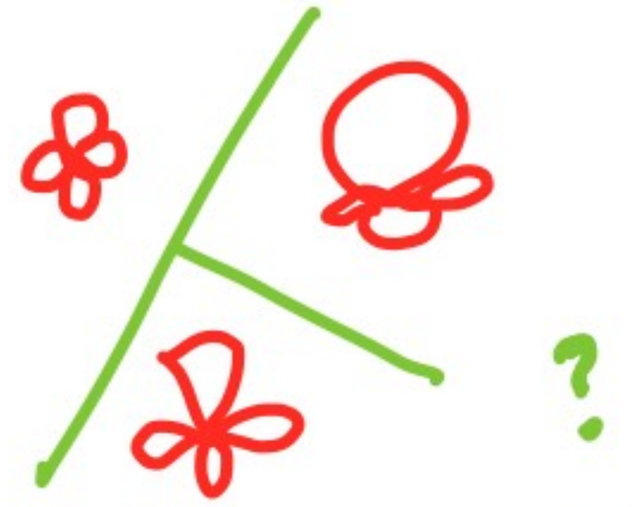
Remember that the goal is to find the weights that maximize the log-likelihood so that we perform the update for each weight as follows:

$$w_j := w_j + \eta \sum_{i=1}^{n} \left(y^{(i)} - \phi\left(z^{(i)}\right) \right) x_j^{(i)}$$

: equal to Δw in Adaline

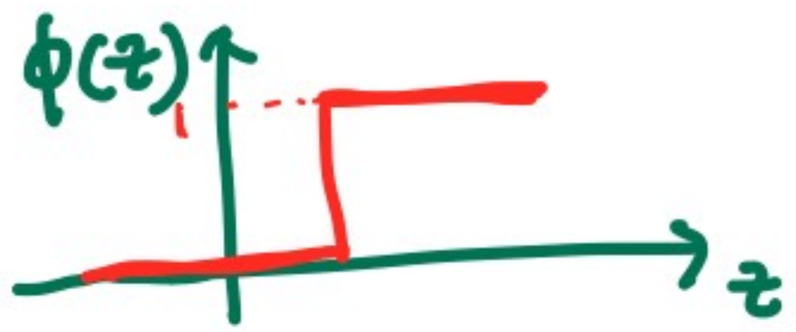
Classification – 다중 클래스 분류 모델

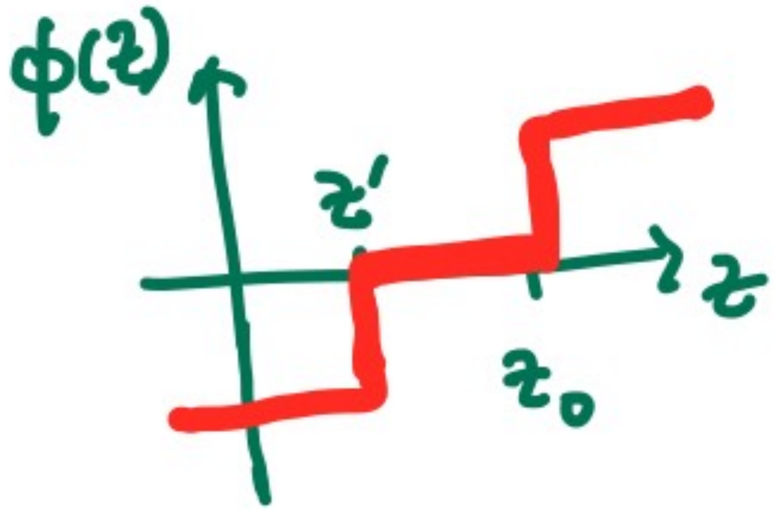
세 종류의 붓꽃 분류



$$\phi(w^T x) = \phi(z) = \begin{cases} 1 & \text{if } z \geq z_0 \\ 0 & \text{else} \end{cases}$$

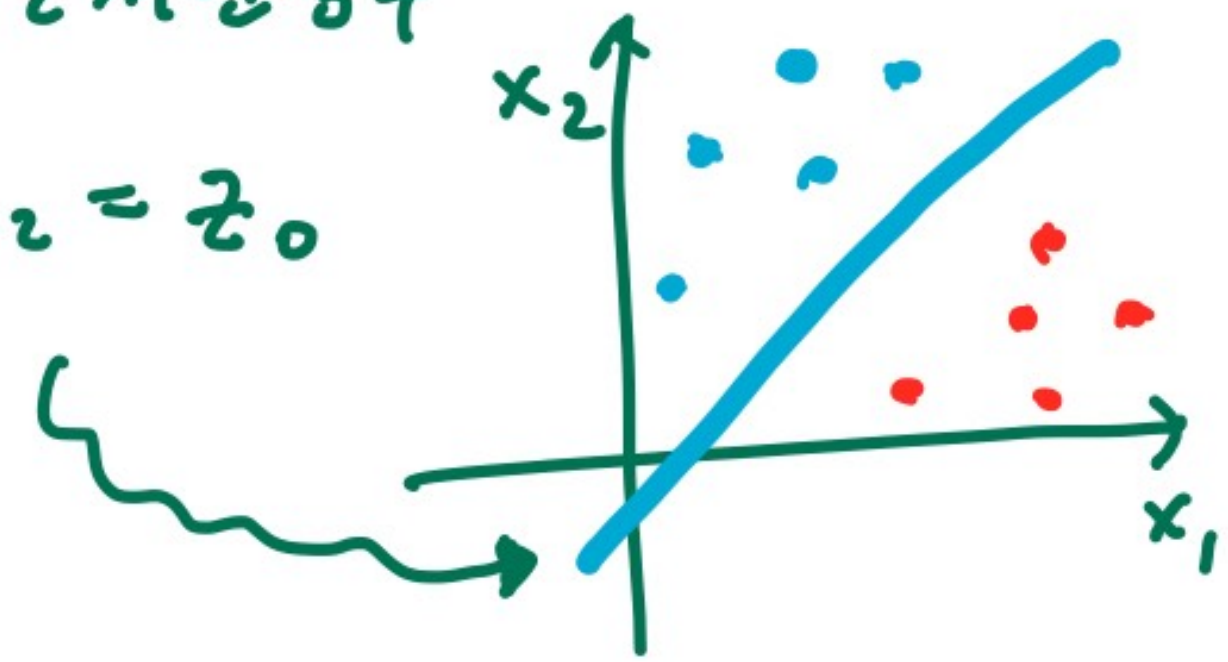
threshold

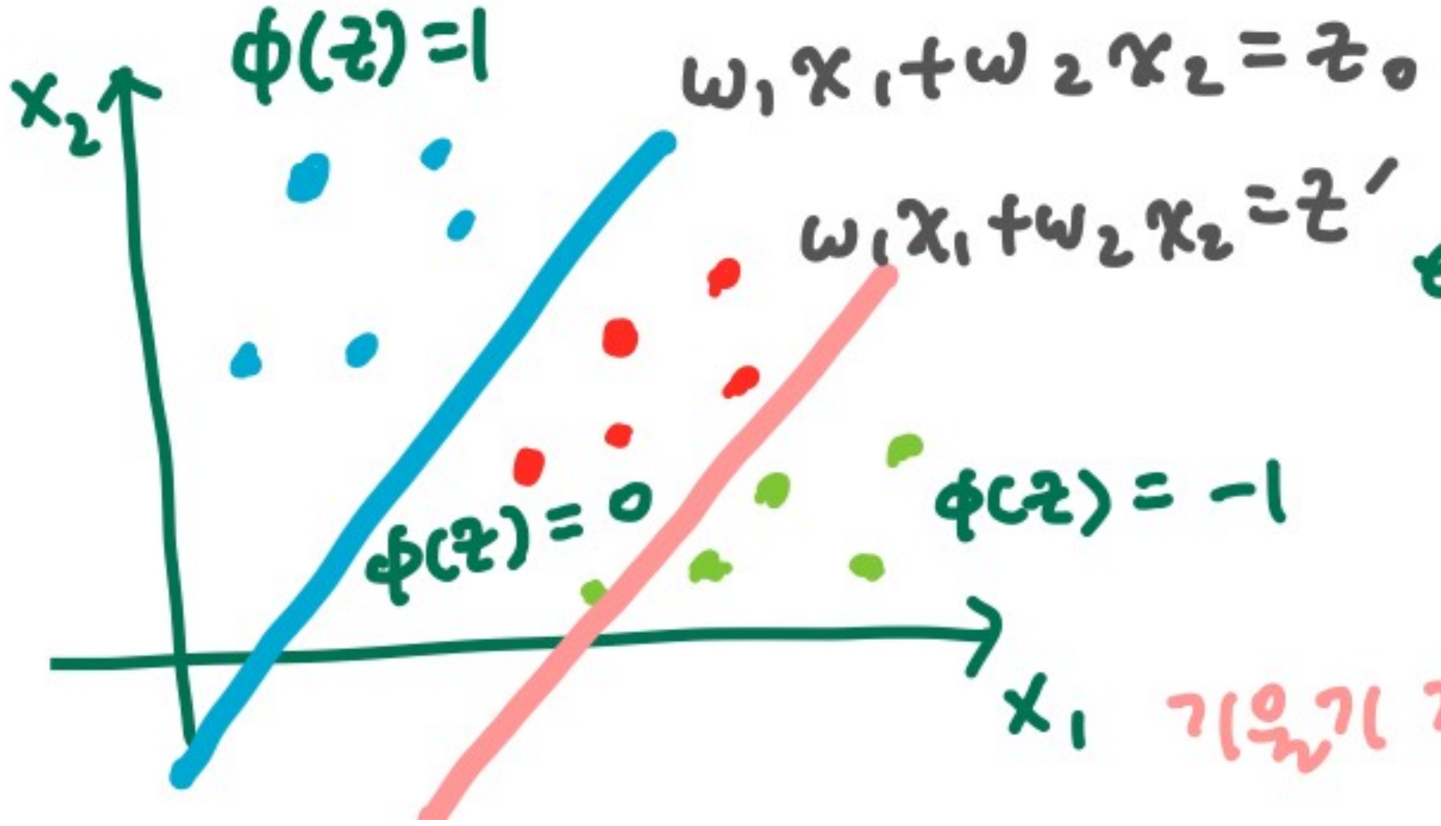




ex) feature가 2개인 경우

$$w_1 x_1 + w_2 x_2 = z_0$$

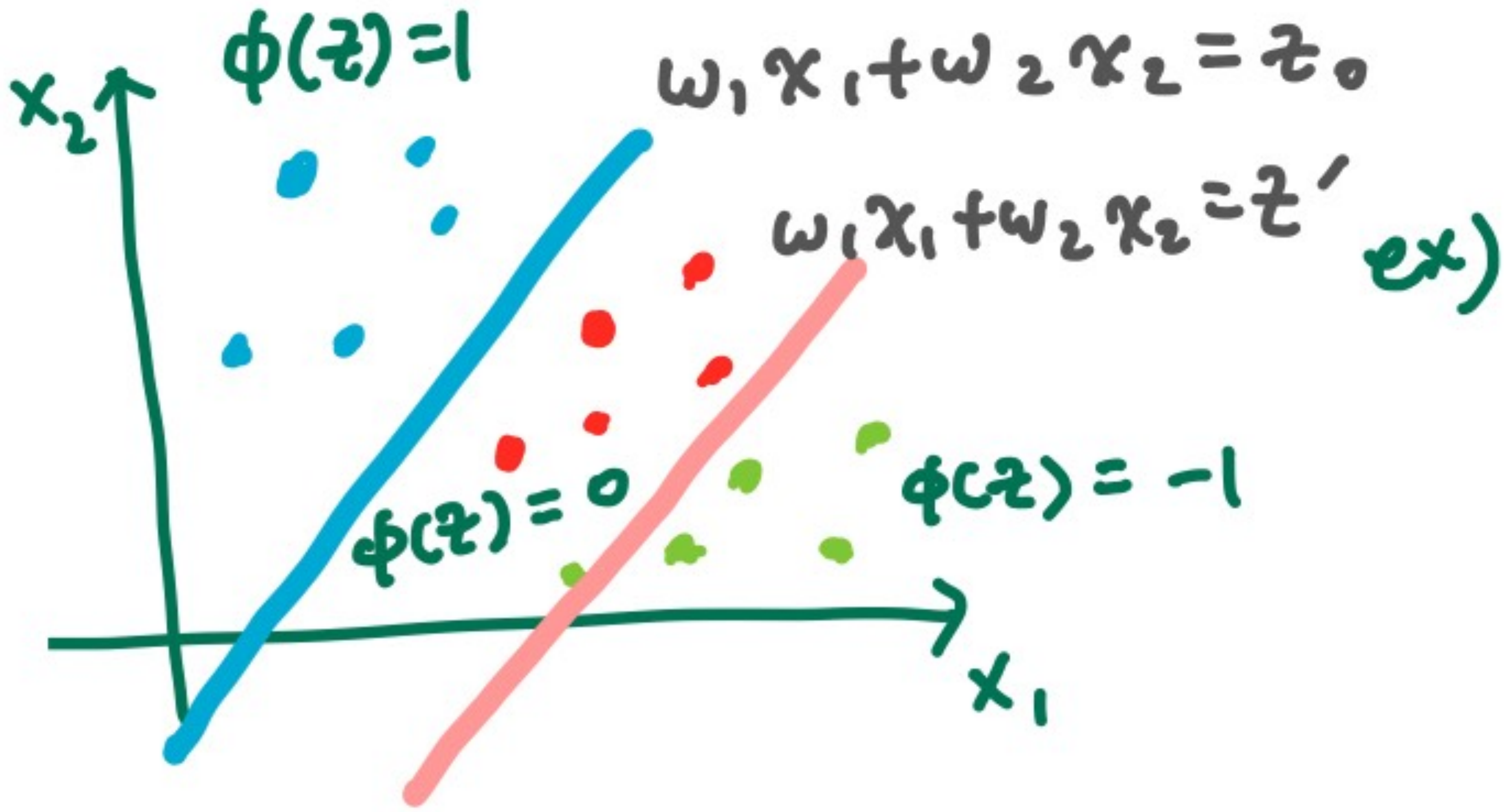




ex) 3개로 classification 하는 경우

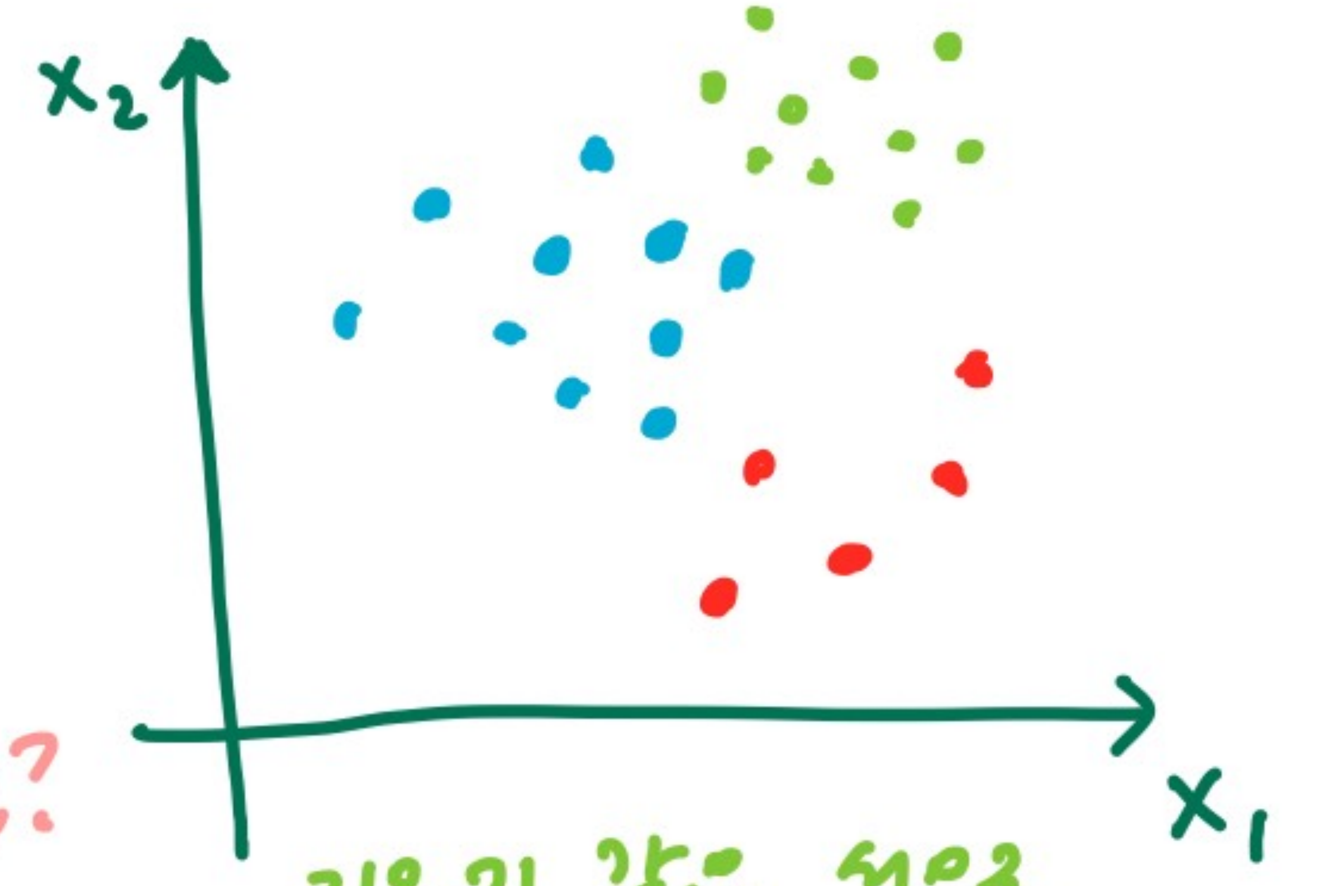
$$\phi(z) = \begin{cases} 1 & \text{if } z \geq z_0 \\ 0 & \text{else if } z' < z < z_0 \\ -1 & \text{else} \end{cases}$$

Classification – 다중 클래스 분류 모델



만일

이런 분포라면?



기울기 같은 선으로 Classify 하기 어려움.

even in hyperspace?

ex) k 개로 classification

ex) 3개 → different weight for 3 different category

j × k weight matrix 이용!

$$W = \begin{pmatrix} w_{11} & w_{12} & \cdots & w_{1k} \\ w_{21} & \ddots & & \\ \vdots & & \ddots & \\ w_{j1} & & & w_{jk} \end{pmatrix}, \quad W^T x = \begin{pmatrix} w_{11} & w_{12} & \cdots & w_{1j} \\ w_{21} & \ddots & & \\ \vdots & & \ddots & \\ w_{k1} & & & w_{kj} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_j \end{pmatrix} = \begin{pmatrix} w_{11}x_1 + \cdots w_{11}x_j \\ \vdots \\ w_{k1}x_1 + \cdots w_{kj}x_j \end{pmatrix}$$

Classification – 다중 클래스 분류 모델

j 개의 feature를 가진 x 에 대해 k 개의 값이 나옴

cost function을 정의하는 방법에 따라 ω matrix는 달리 결정되겠으나,

붓꽃 예제라 같이 3개의 category 가 있는 경우 (feature는 4개라하자)

A: $w_{11}x_1 + w_{12}x_2 + w_{13}x_3 + w_{14}x_4 = 1$ (1에 가까운 수)

B: $w_{21}x_1 + w_{22}x_2 + w_{23}x_3 + w_{24}x_4 = 1$ "

C: $w_{31}x_1 + w_{32}x_2 + w_{33}x_3 + w_{34}x_4 = 1$ "

$\rightarrow \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} = z$

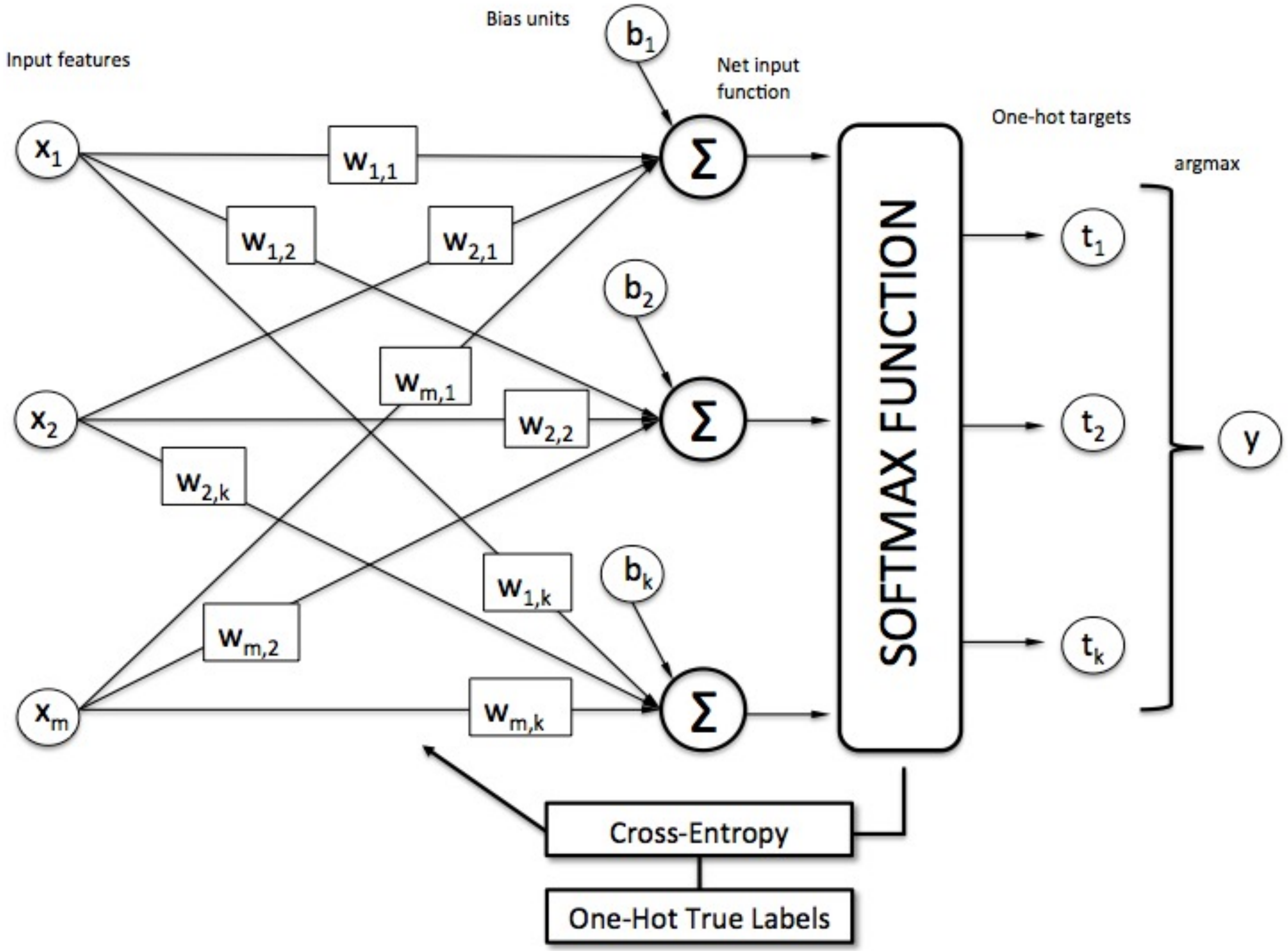
인 3개의 $\underbrace{\text{hyper}}_{\text{기울기가 다른}}$ 평면을 찾을 수 있을 것이다.

여전히 평면!

Test 붓꽃이 $z = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$ 이면 A 카테고리에 있는 붓꽃!

일반적으로 $\overbrace{\text{Softmax function}}^{\text{activation function}}$ 사용 $\phi(z^{(k)}) = \dfrac{e^{z^{(k)}}}{\sum_{j=0}^{k} e^{z_j^{(k)}}}$

Classification – 다중 클래스 분류 모델

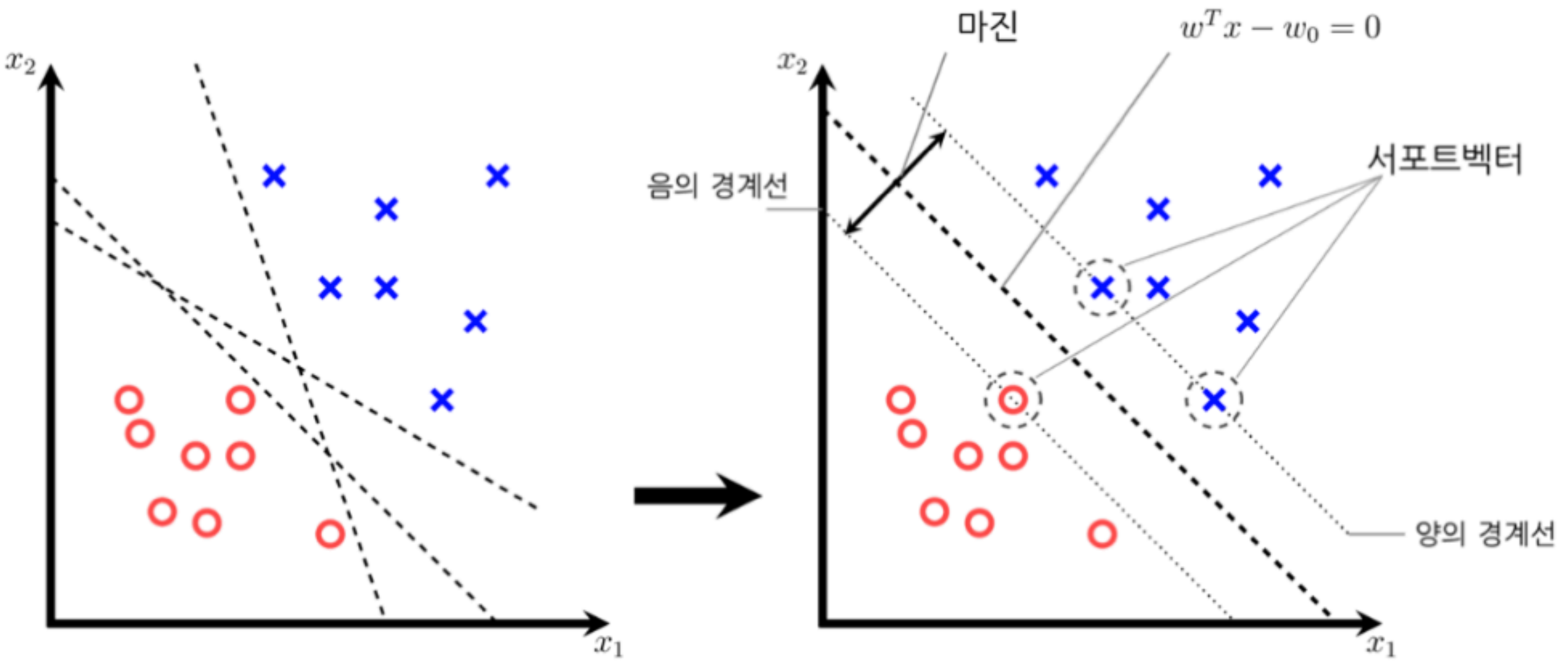


$$p_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_j}}$$

$$j = 1, 2, \ldots, K$$

서포트 벡터 머신을 사용한 분류

- 마진: 클래스를 구분하는 초평면과 이 초평면에 가장 가까운 훈련 샘플 사이의 거리
- 서포트 벡터 (support vector): 가장 가까운 샘플

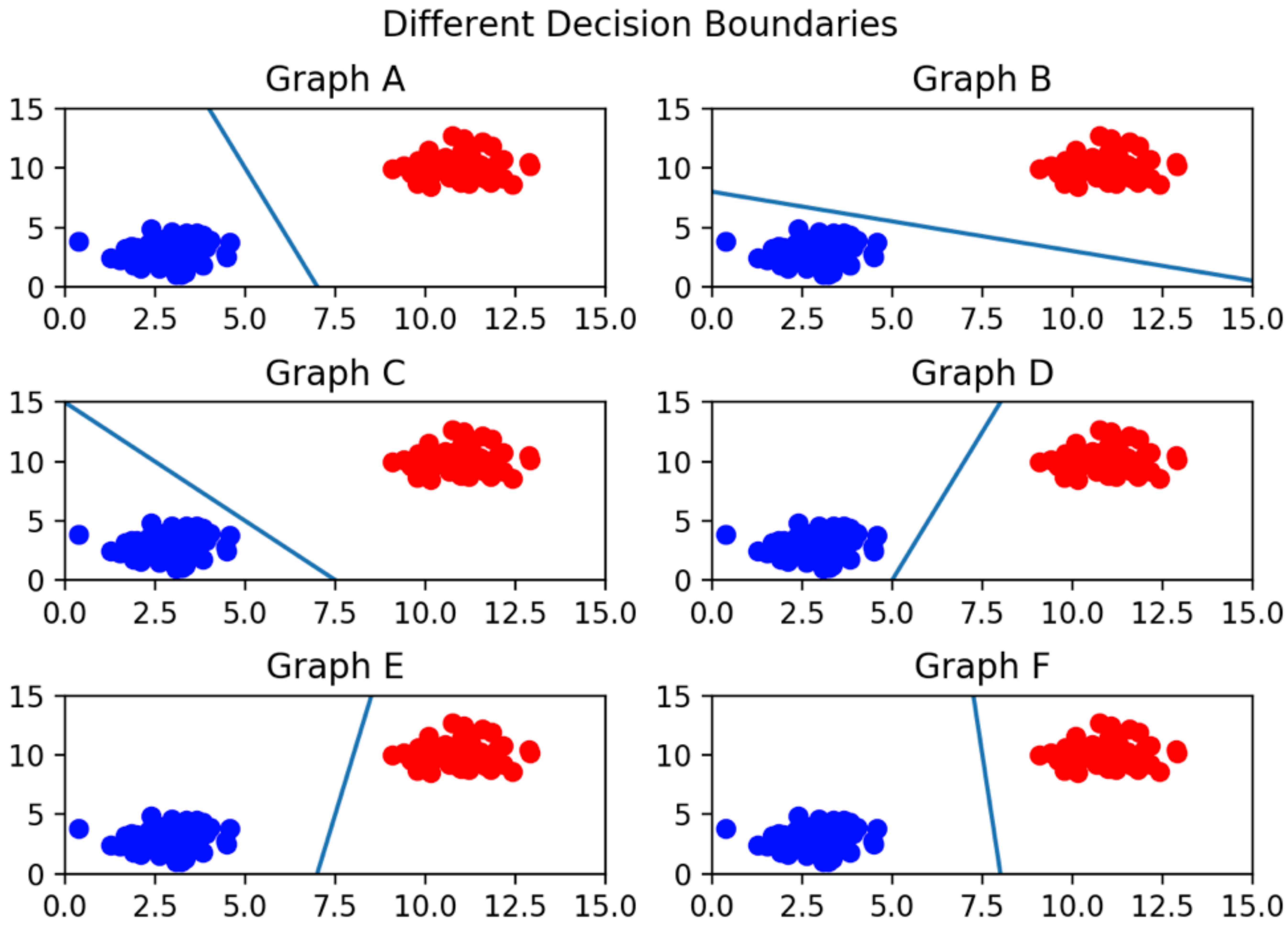


초평면

$$w_0 + w^T x_{pos} = 1$$

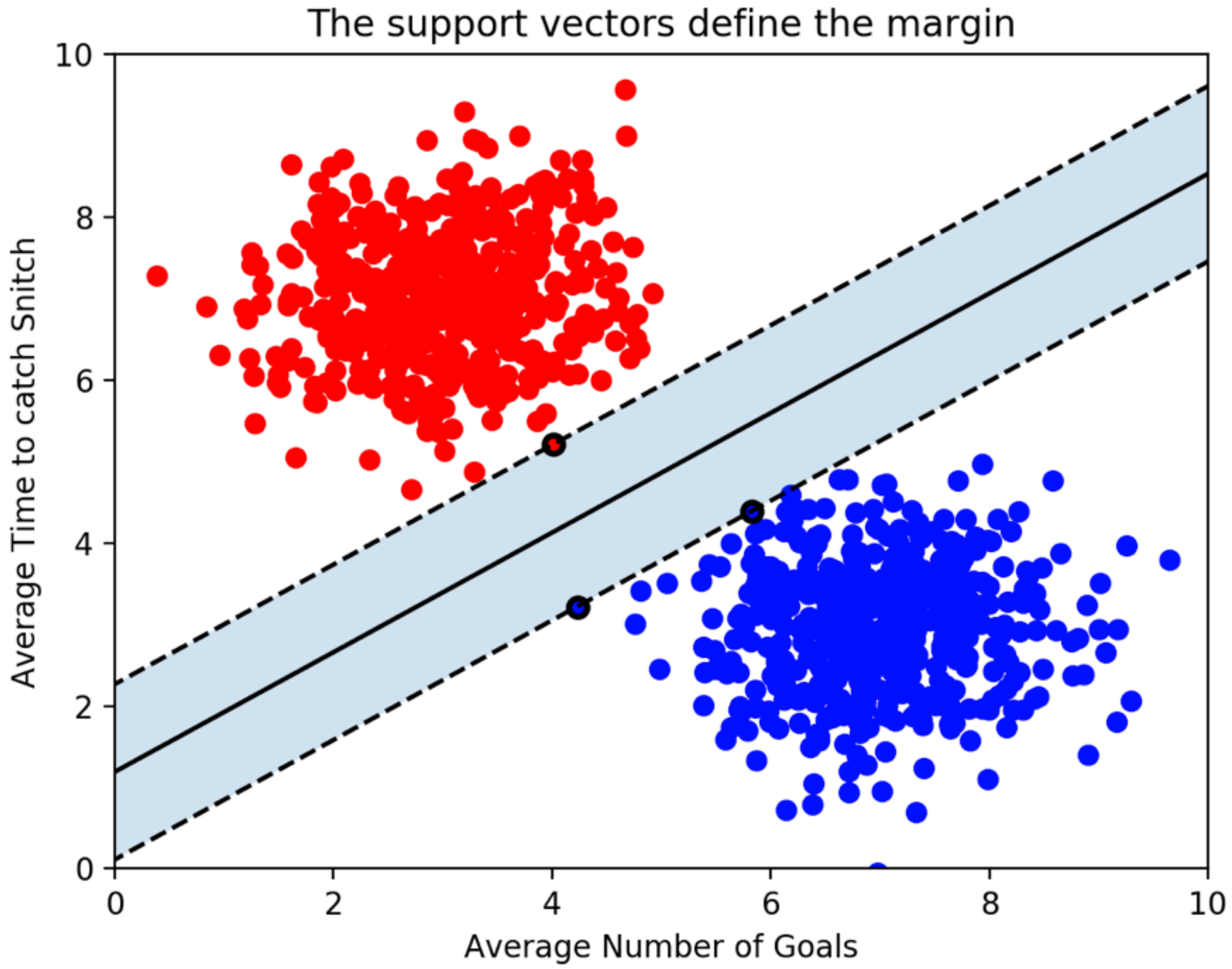
$$w_0 + w^T x_{neg} = -1$$

서포트 벡터 머신을 사용한 분류



- 두 데이터로부터 가장 멀리 떨어진 decision boundary가 가장 적합

➡ boundary와 데이터 간의 거리 최대화: 마진 최대화 (잘못 분류할 가능성을 낮춰 줌 →support vector에 덜 민감하다)



서포트 벡터 머신을 사용한 분류

(1) - (2)

$$w^T(x_{pos} - x_{neg}) = 2$$

$$\|w\| = \sqrt{\sum_{j=1}^{m} w_j^2}$$

두 평면 사이의 거리 $$\frac{w^T(x_{pos} - x_{neg})}{\|w\|} = \underbrace{\frac{2}{\|w\|}}_{\text{최대화!}}$$

: 양성 초평면과 음성 초평면 사이의 거리!

= 마진

단, $w_0 + w^T x^{(i)} \geq 1$, $y^{(i)} = 1$ 일때

$w_0 + w^T x^{(i)} \leq -1$, $y^{(i)} = -1$ 일때

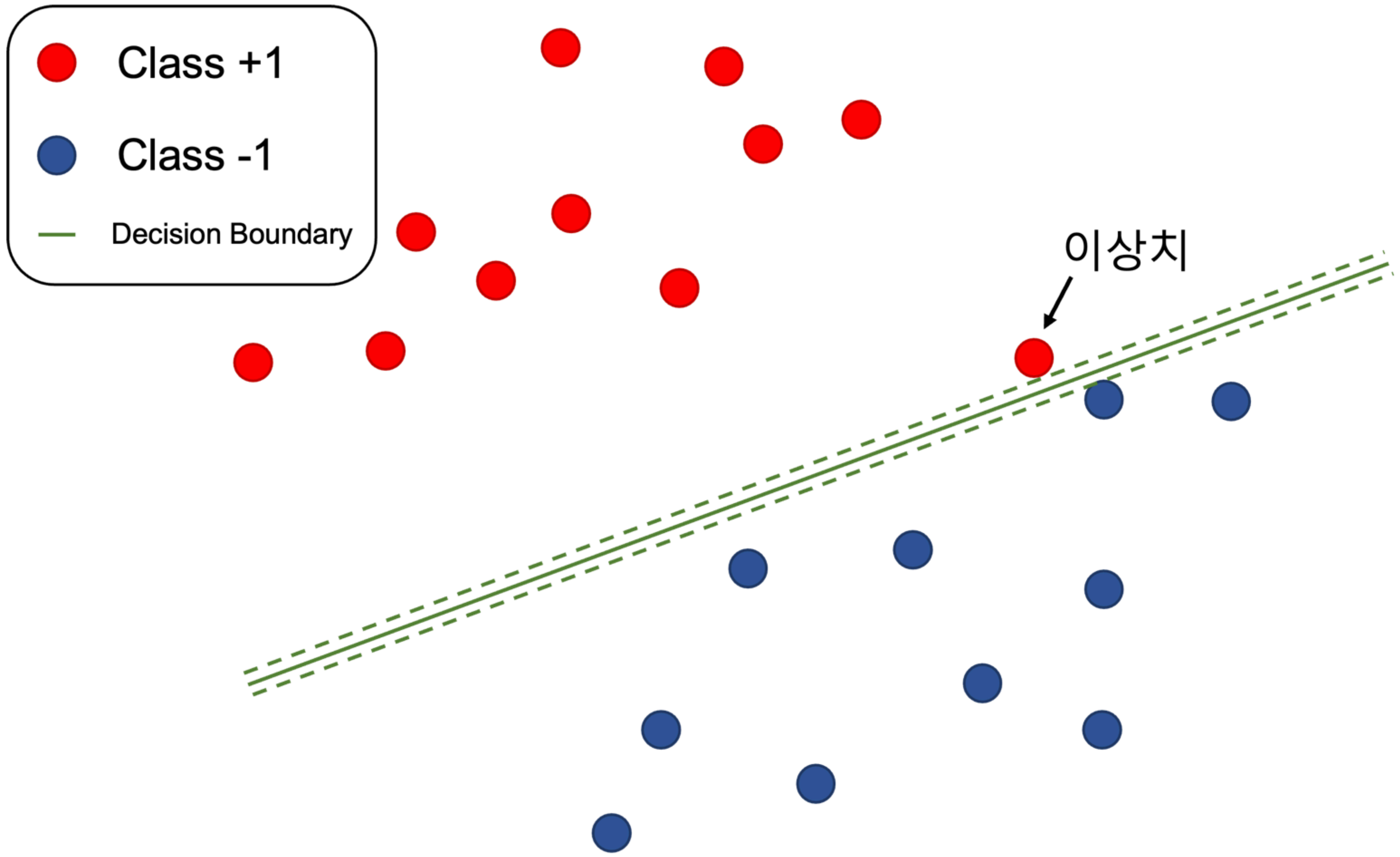
$i = 1 \ldots N$

샘플이 정확하게 분류된다는 제약조건하에!

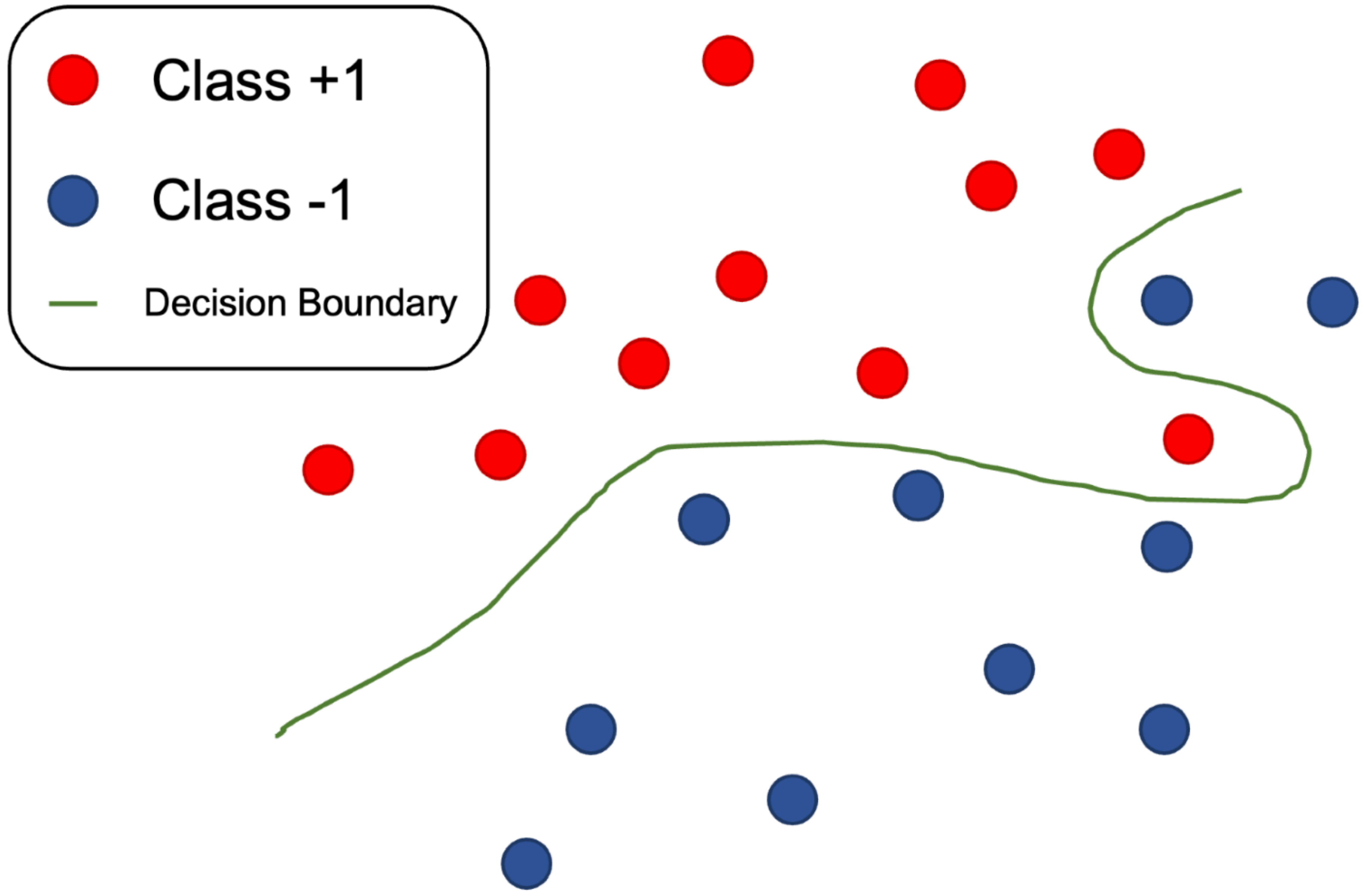
➡ $y^{(i)}(w_0 + w^T x^{(i)}) \geq 1 \quad \forall i$

$\Leftrightarrow$ $\frac{1}{2}\|w\|^2$ 을 최소화!

Outliers

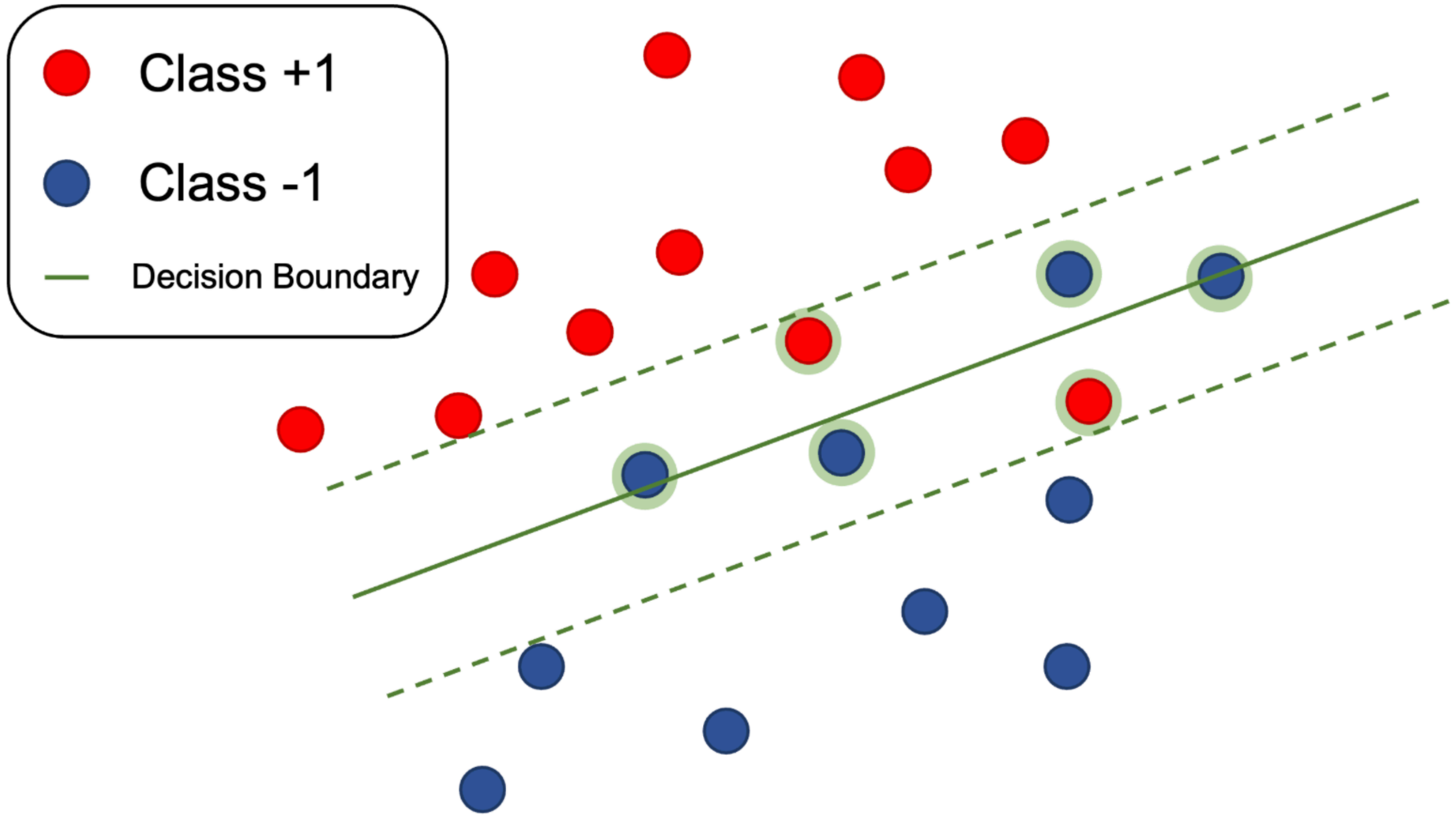


선형분류 불가능!

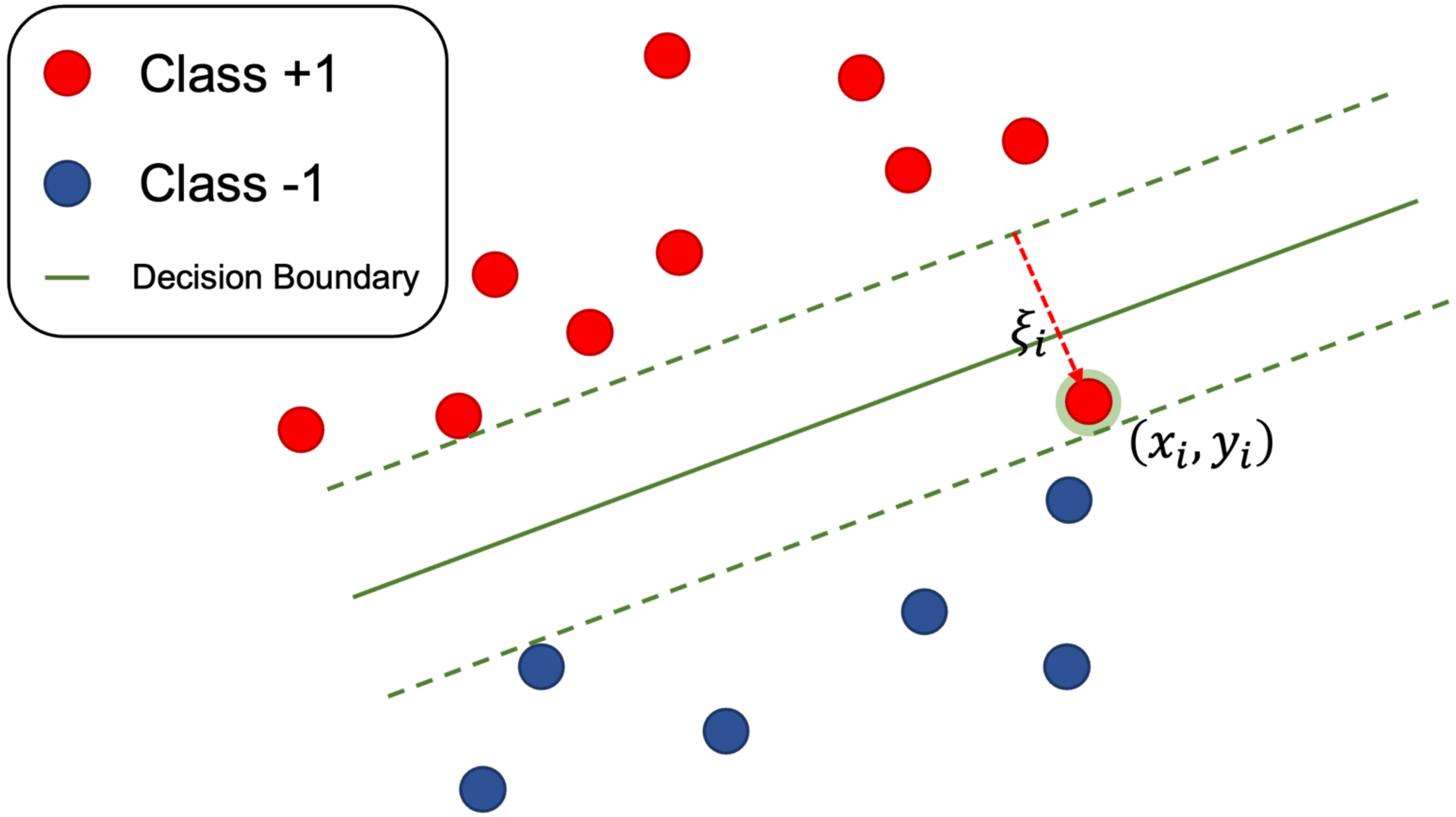


Outliers

융통성이 좀 있어야지하지 않아?

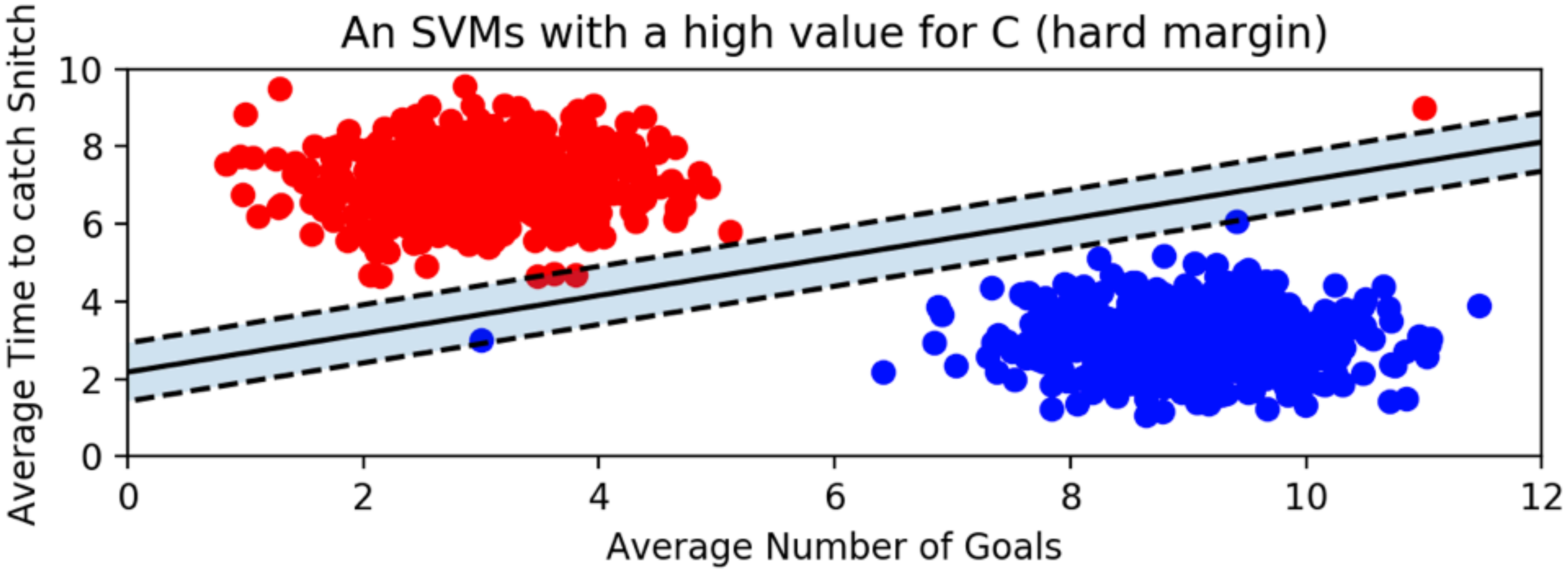


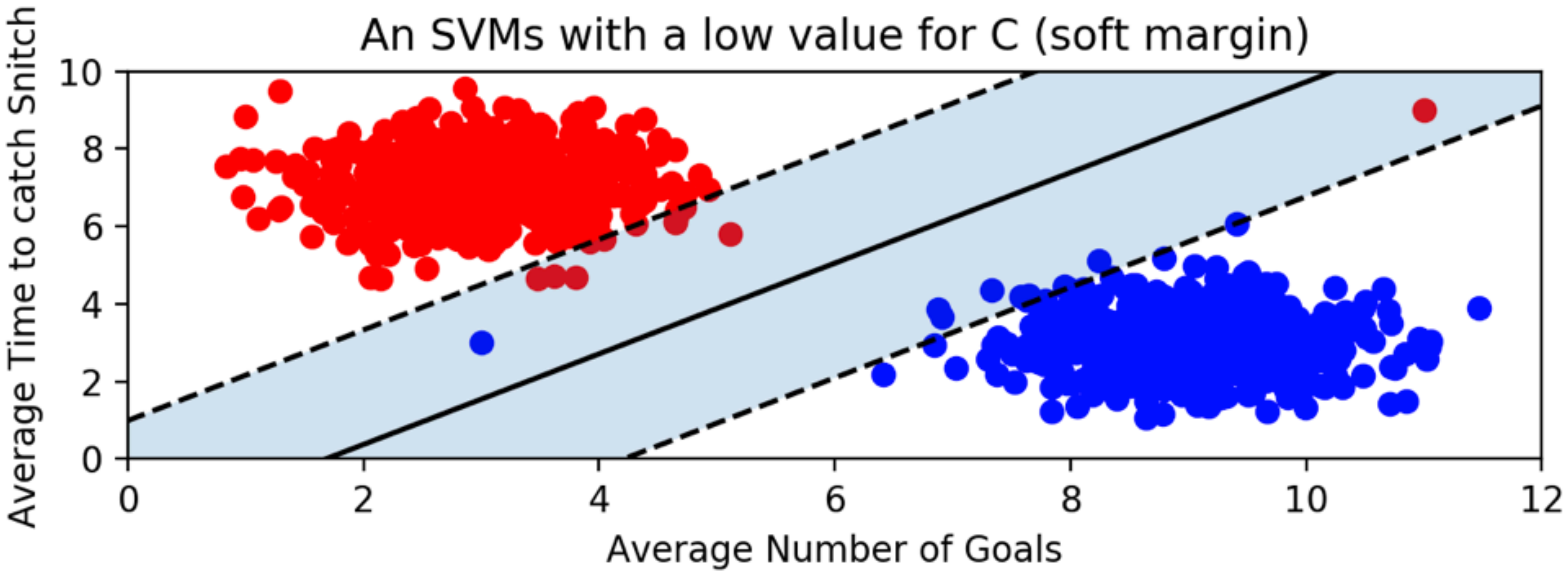
잉여 변수 (Slack variable) 도입
→ξ 정도는 용인해주자!



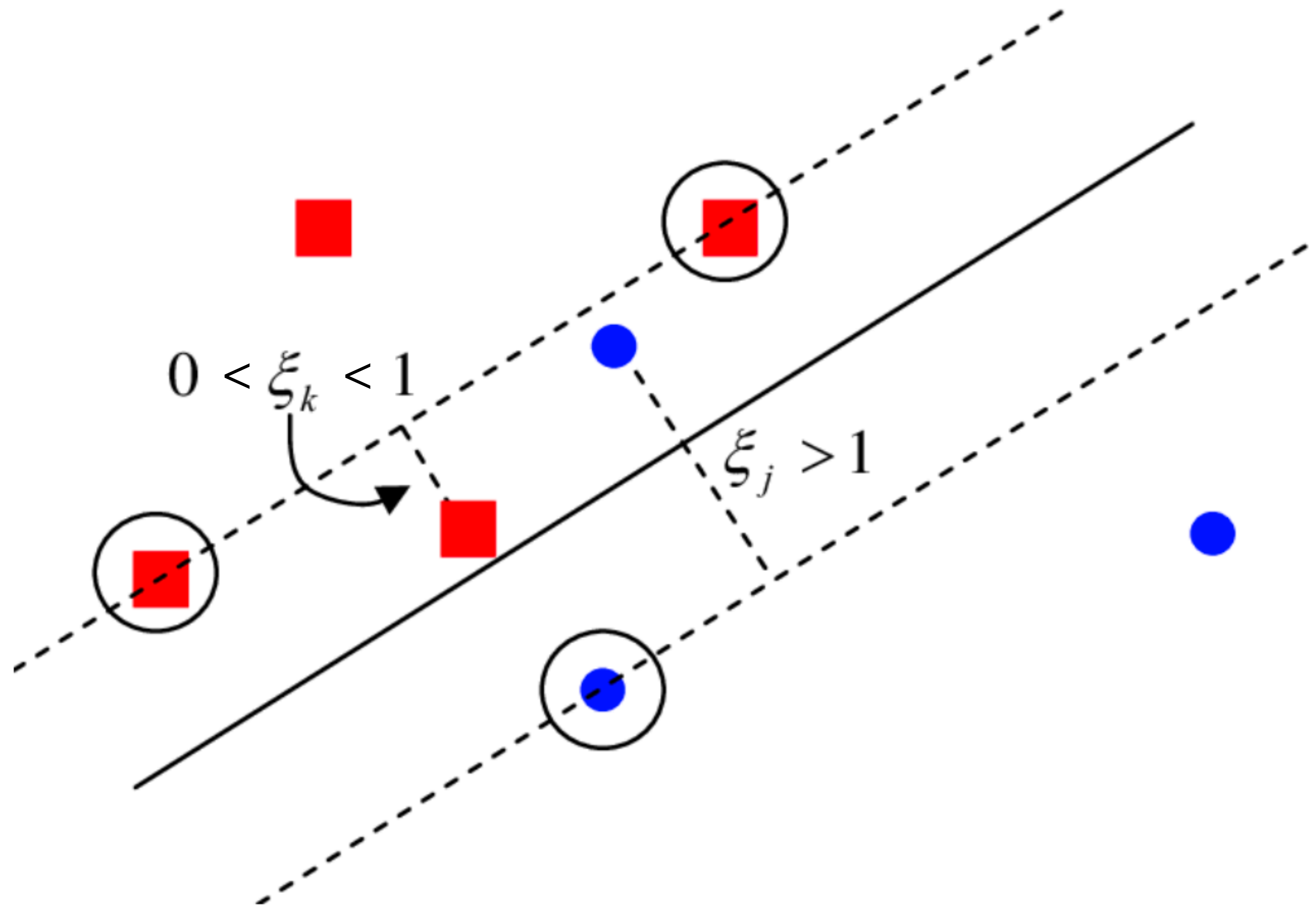
모두 잉여? No
잉여의 총 합은 최소가 되도록 하자.

Outliers





- 마진 안쪽에도 존재할 수 있도록 예외를 허용 (soft margin)



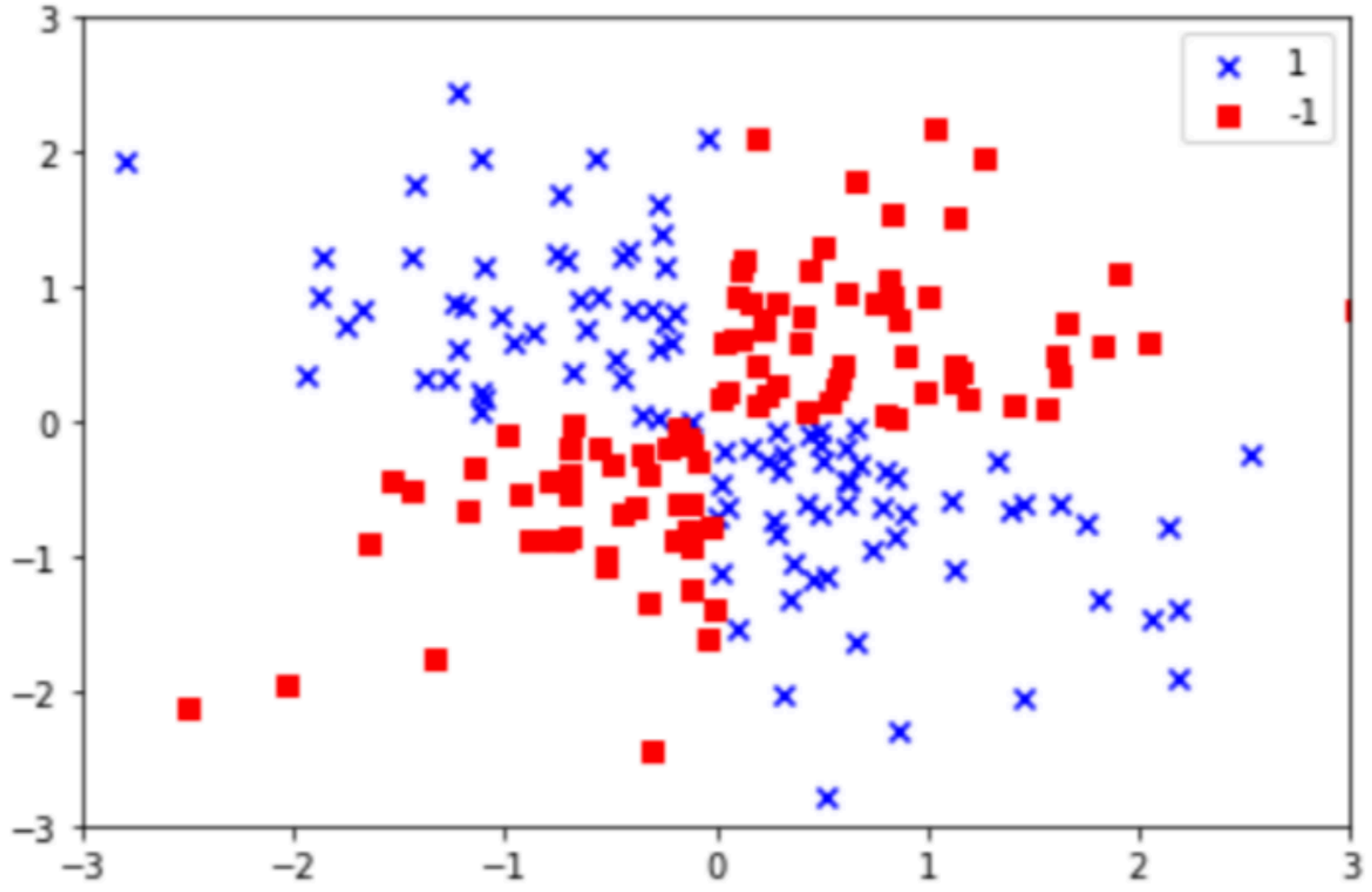
$w_0 + w^T x^{(i)} \geq 1 - \xi^{(i)}$ $y^{(i)} = 1$ 일때

$w_0 + w^T x^{(i)} \leq -1 + \xi^{(i)}$ $y^{(i)} = -1$ 일때

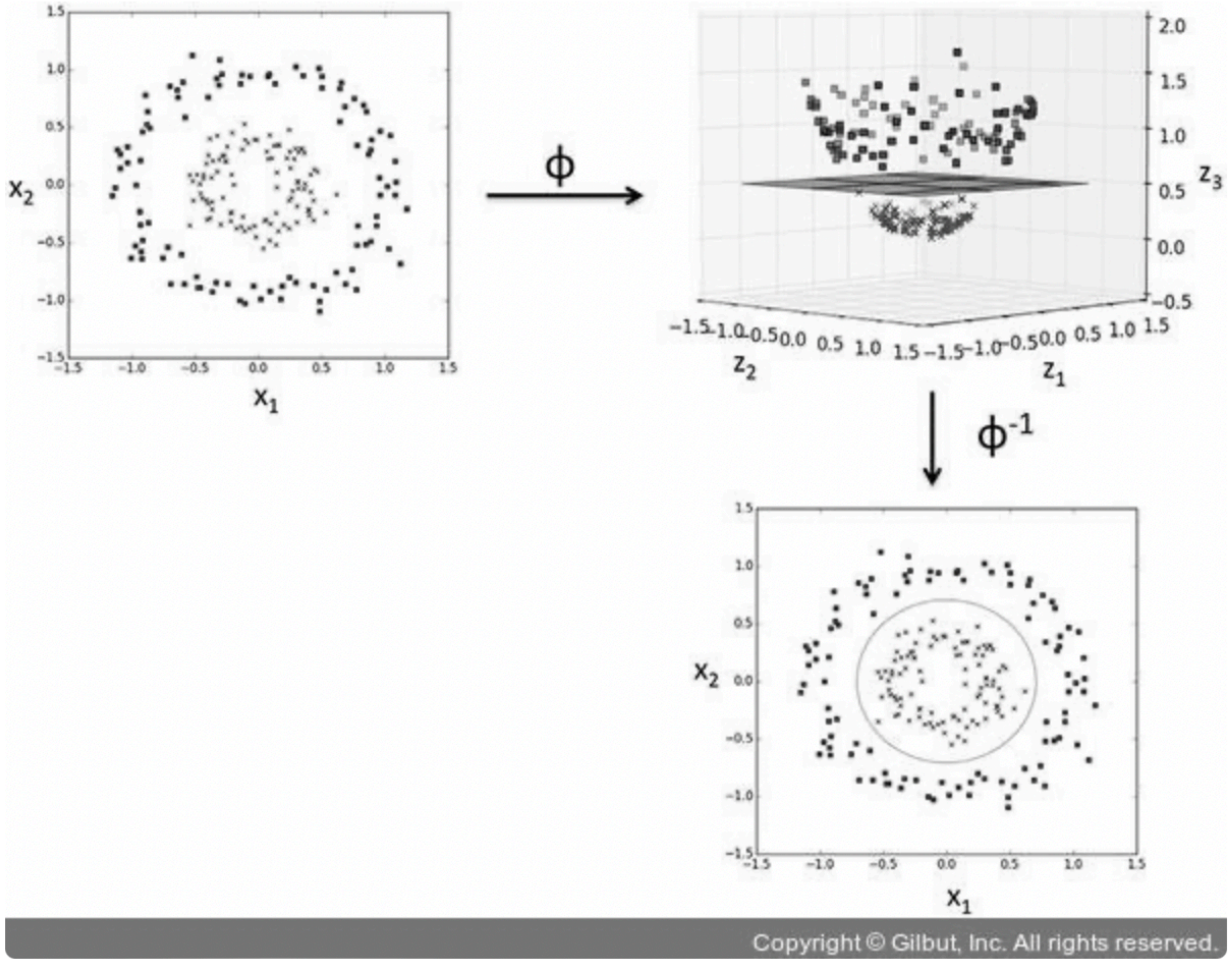
$\rightarrow \frac{1}{2}\|w\|^2 + C\left(\sum_i \xi^{(i)}\right)$

SVM을 사용하여 비선형 문제풀기

XOR data set

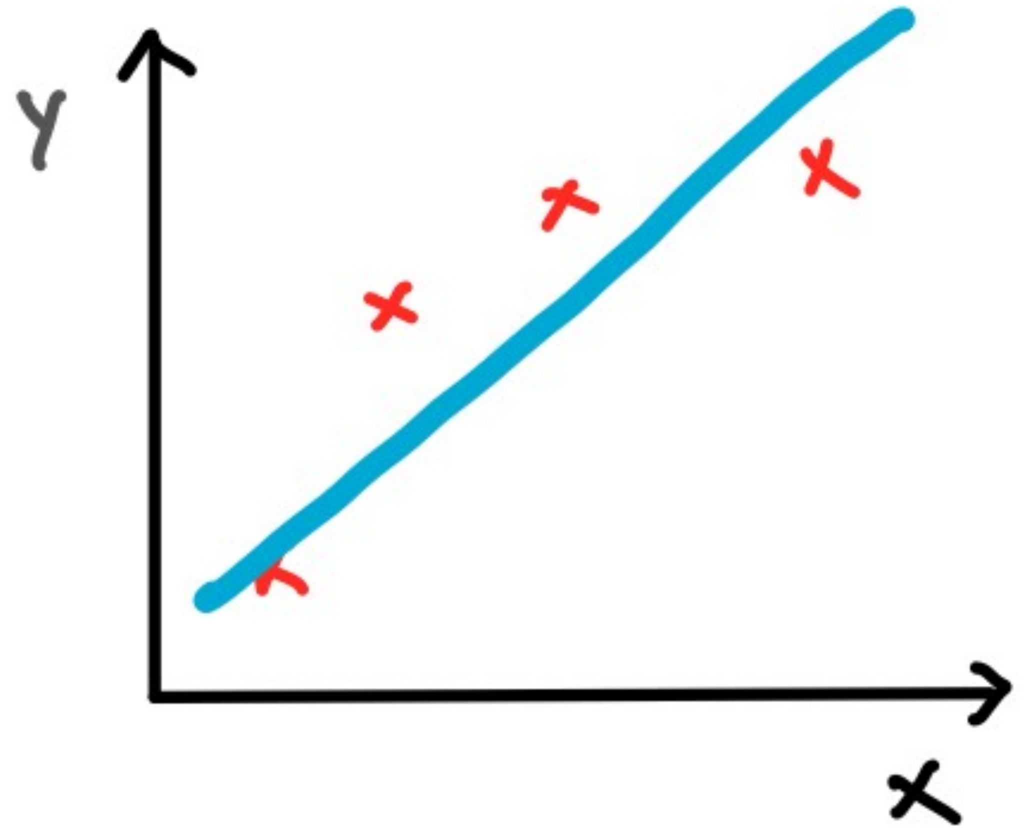


훈련데이터를 고차원 특성 공간으로 변환 → 커널기법 사용



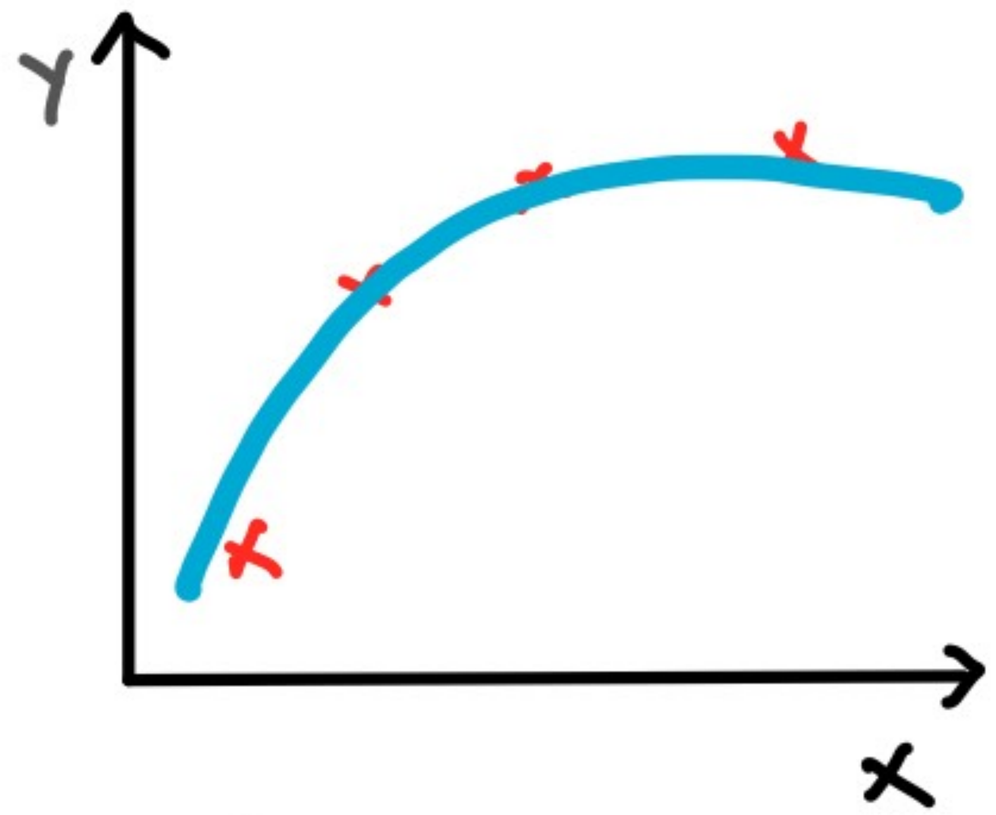
Extra Slides

Regularization

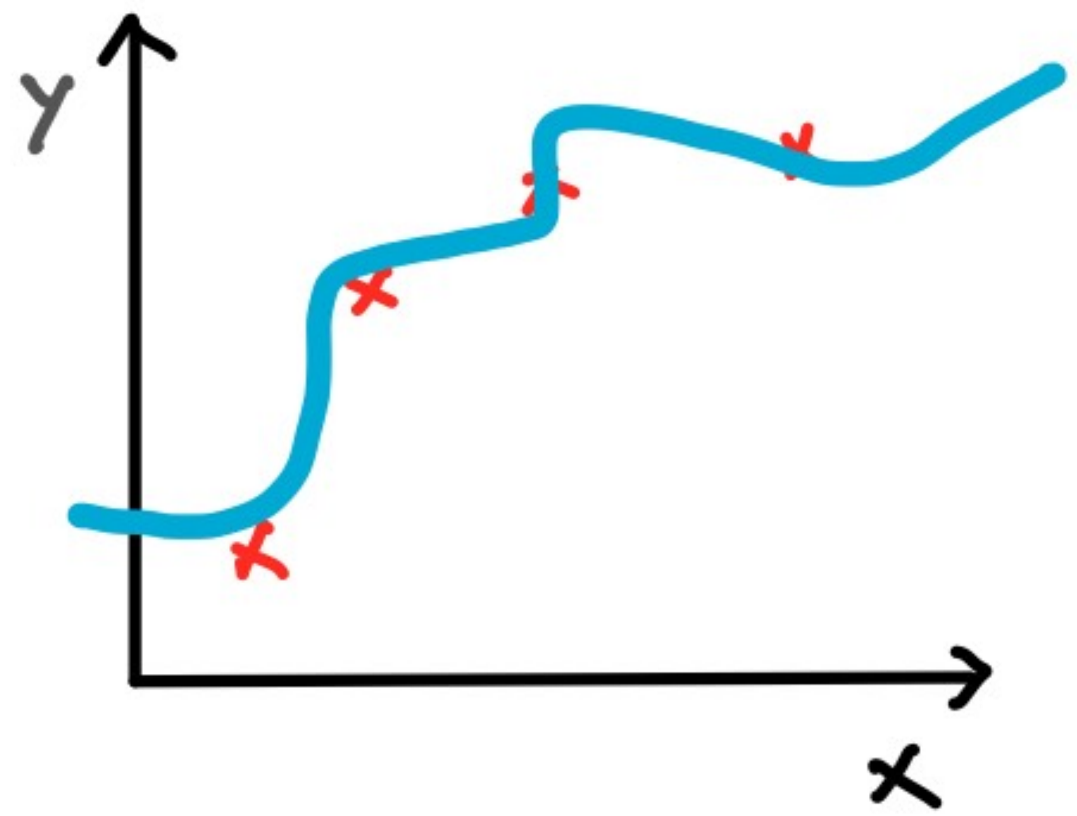


$\theta_0 + \theta_1 x$

Underfitting



$\theta_0 + \theta_1 x + \theta_2 x^2$



$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 +$

Over fitting

새로운 data를 잘 맞출 수 X

과적합 해결방법

1) feature의 갯수를 줄인다

2) Regularization

ex) $\min(\theta_0 + \theta_1 x + \theta_2 x^2 + \underbrace{10^3 \theta_3^2 + 10^3 \theta_4^2}_{})$

가중치를 부여해 cost function에 더함!

Regularization

Cost function can be regularized by adding a simple term

regularization hyperparameter

$$J(\mathbf{w}) = \underbrace{\sum_{i=1}^{n}\left[-y^{(i)}\log\left(\phi\left(z^{(i)}\right)\right) - \left(1-y^{(i)}\right)\log\left(1-\phi\left(z^{(i)}\right)\right)\right]}_{\text{Original cost function}} + \underbrace{\frac{\lambda}{2}\|\mathbf{w}\|^2}_{\text{L2 regularization term}}$$

Original cost function

L2 regularization term

$$\frac{\lambda}{2}\|\mathbf{w}\|^2 = \frac{\lambda}{2}\sum_{j=1}^{m} w_j^2$$