

## 4. 머신러닝 적용

# I. Review

---

- ✓ 분석 기획 / 문제 정의
- ✓ 데이터 수집
  - ✓ 웹 수집
- ✓ 데이터 전처리
  - ✓ 정형->정형
  - ✓ 비정형->정형
  - ✓ 반정형->정형
- ✓ 자연어 처리

## II. ML 소개

---

1. ML 개요
2. 분류 모형의 이해
3. 다양한 분류 모형
4. ANN

# 1. ML 개요

## ➤ Machine Learning 모형 구분

### 지도학습 (Supervised Learning)

Target을 **추론(Inference)**하고  
**예측(Prediction)**하는 모형

예: 회귀/분류 모형



### 비지도학습 (Unsupervised Learning)

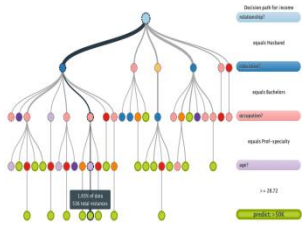
Target이 없으며, 데이터에서의 관계  
나 패턴 등을 발견하는 모형

예: 군집 분석, 연관성 분석, 주성분  
분석 등

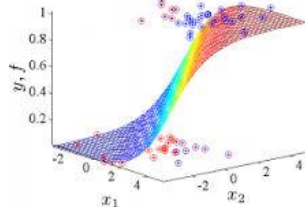


### 강화학습 (Reinforcement Learning)

[decision tree]



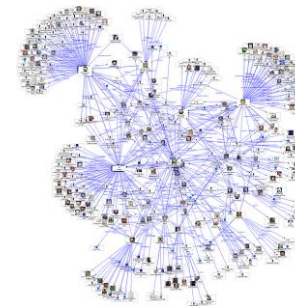
[logistic regression]



[clustering analysis]



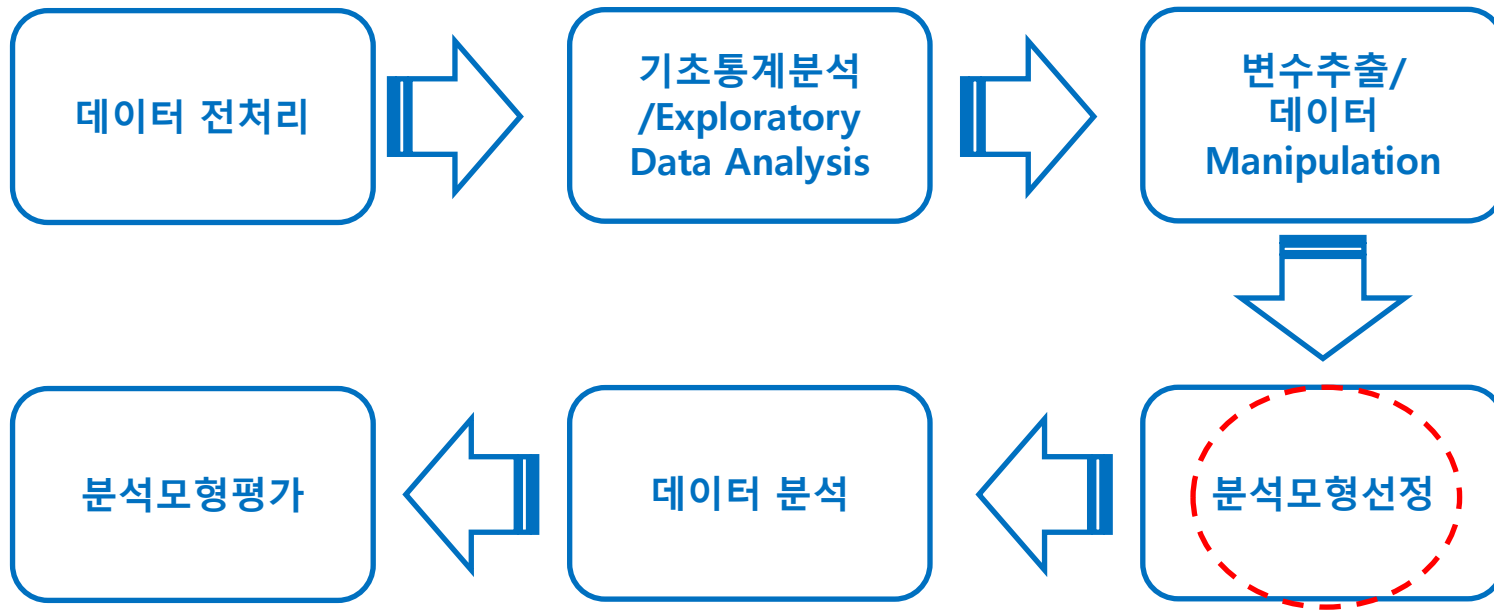
[link analysis]



# 1. ML 개요

---

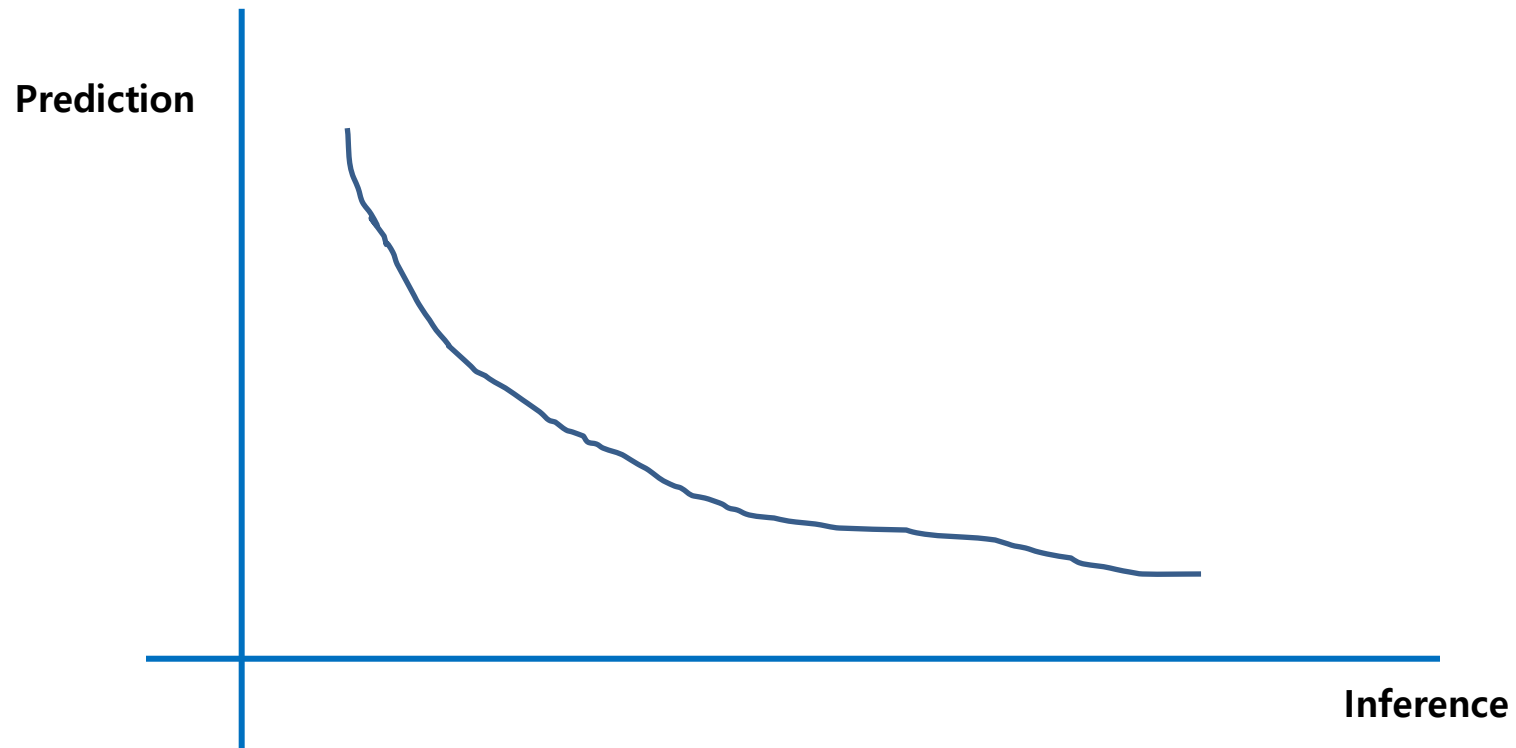
## ➤ Machine Learning 세부 절차



# 1. ML 개요

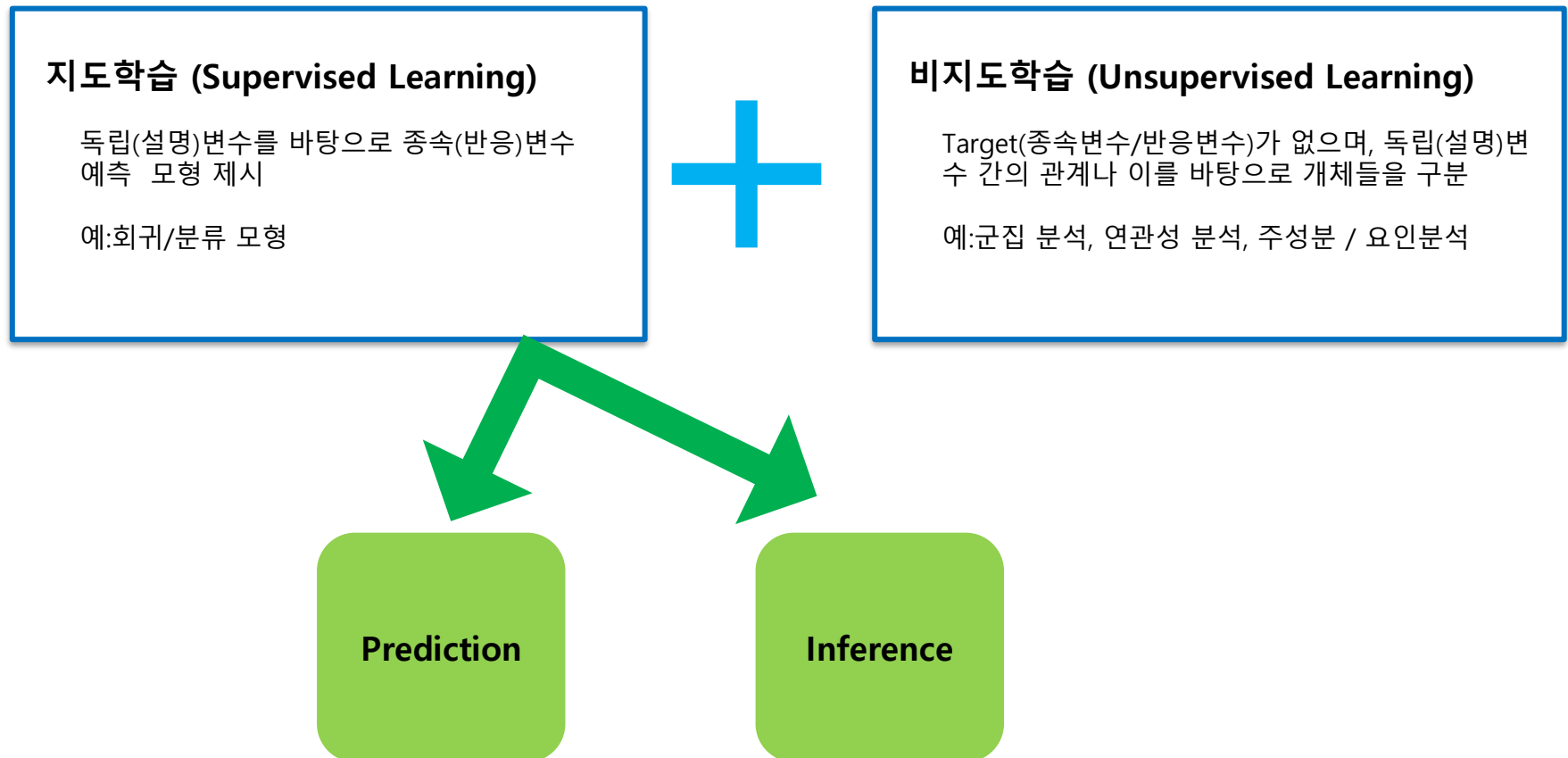
---

## ➤ 데이터 분석의 목적



# 1. ML 개요

## ➤ 데이터분석의 목적 → 예측 or 추론



## 2. 분류 모형의 이해

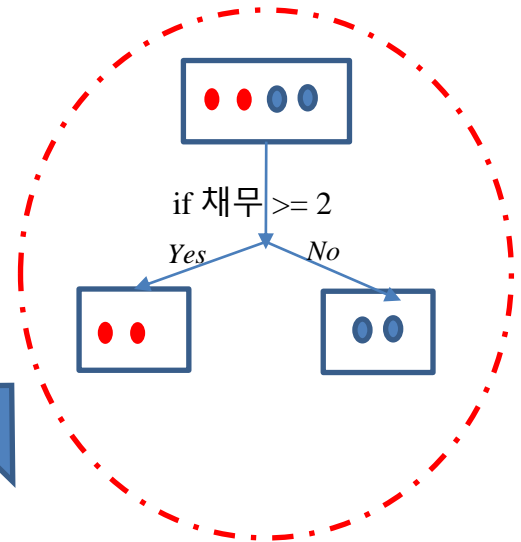
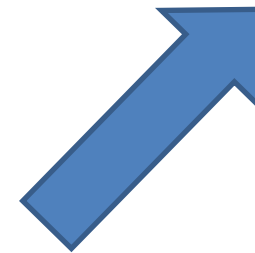
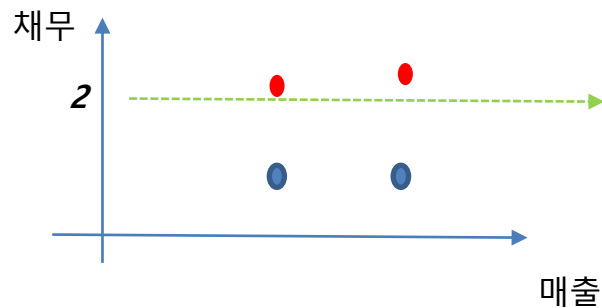
투자 대상 기업 4곳!

분류(Classification) 모형은 범주형 Target 변수를 예측하는 지도 학습의 기법!

회사	target	input	
	부도여부	채무	매출
AAA	N	1	5.5
BBB	N	1	4.5
CCC	Y	2	4.5
DDD	Y	2.1	5.5



● 부도  
● 정상



결정을 해주는 나무?



## 2. 분류 모형의 이해

회사	부도여부	채무	매출
AAA	N	1	5.5
BBB	N	1	4.5
CCC	Y	2	4.5
DDD	Y	2.1	5.5

채무 변수의 값들: 1, 2, 2.1

1을 기준으로

1 이상



VS

1 미만

2를 기준으로

2 이상



VS

2 미만

2.1을 기준으로

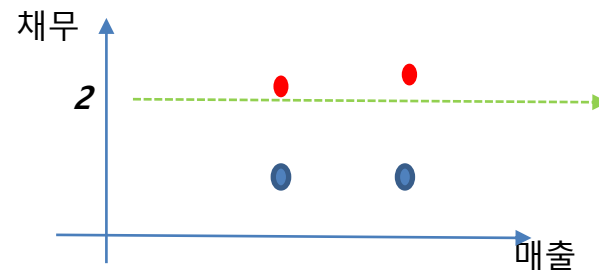
2.1 이상



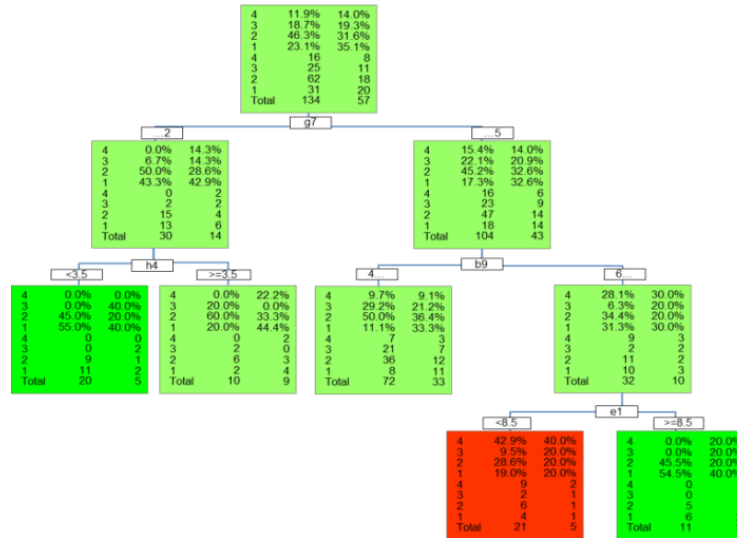
VS

2.1 미만

● 부도  
● 정상



## 2. 분류 모형의 이해



- 해석의 용이성
- 상호작용 효과의 해석
- 복잡한 가정 불필요!

VS

- 비연속성
- 선형성 또는 주효과 결여
- 안정성 부족

## 2. 분류 모형의 이해

---

### **Decision tree**

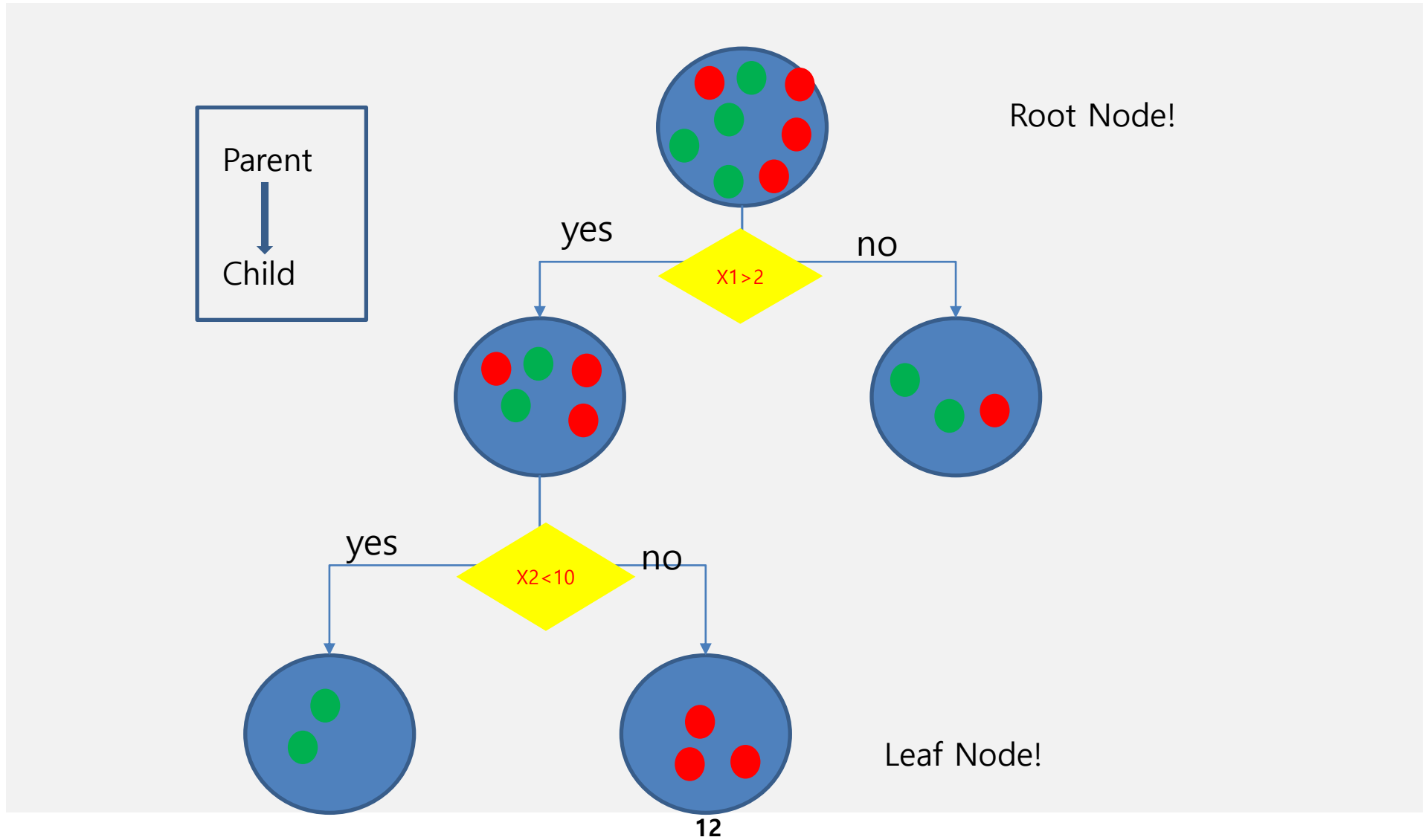
- 분류 기법 중 하나, 1980년대 부터 사용!
  - C4.5 / C5.0 : information theory, entropy, Quinlan (1983)
  - CART(Classification and regression Tree): Gini index, Breiman et al. (1984)
  - CHAID(Chi-squared Automatic Interaction Detector): Chi-square test 이용, Kass(1980)

### • **Decision Tree** 주요 구성요소

- Decision Tree는 Root부터 Leaf node 사이 여러 Node로 구성되며, 이때 Node의 분기는 Rule에 의해 이뤄짐.
- Root부터 각 Leaf Node까지를 각각 Branch라고 하며, Branch까지의 노드의 수를 Depth라고 부름.
- Decision Tree의 시각화: Text로 출력된 Rule들을 효과적으로 보기 위해 Tree구조를 시각화

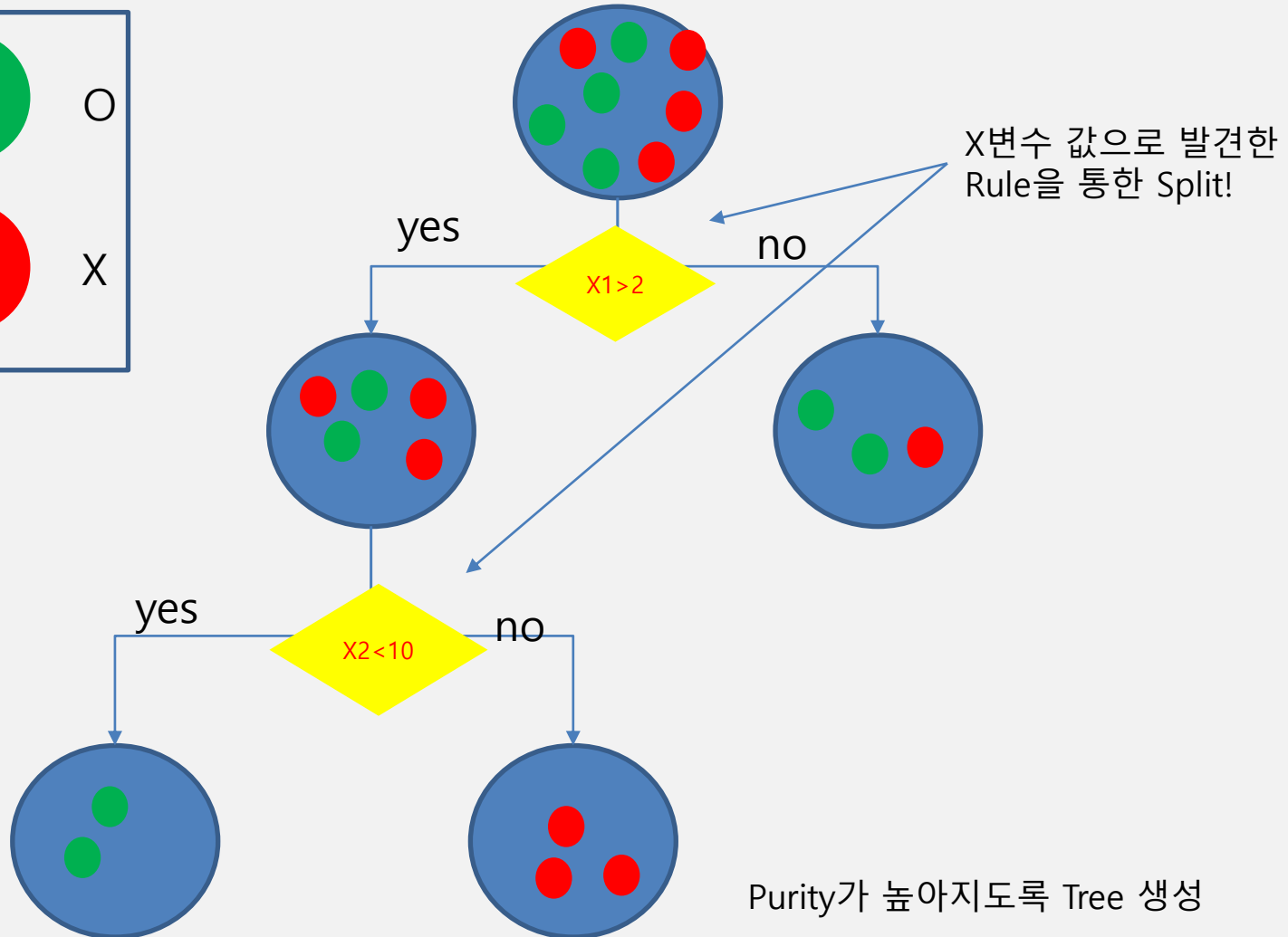
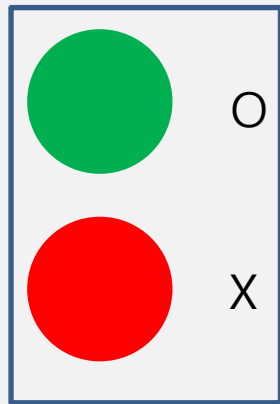
## 2. 분류 모형의 이해

Decision Tree의 구성요소: Roof, Leaf Node



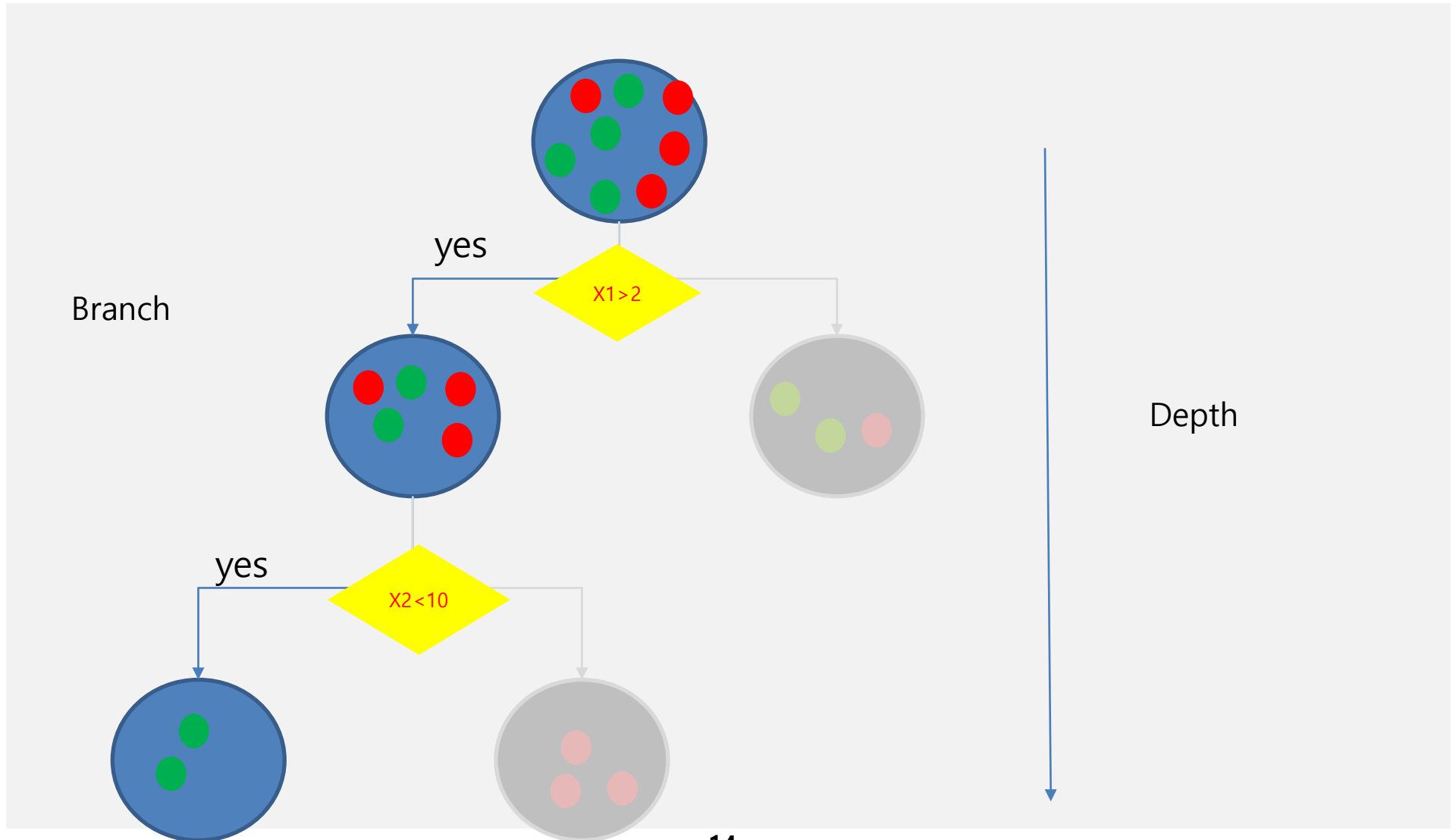
## 2. 분류 모형의 이해

### Decision Tree의 구성요소: Rule



## 2. 분류 모형의 이해

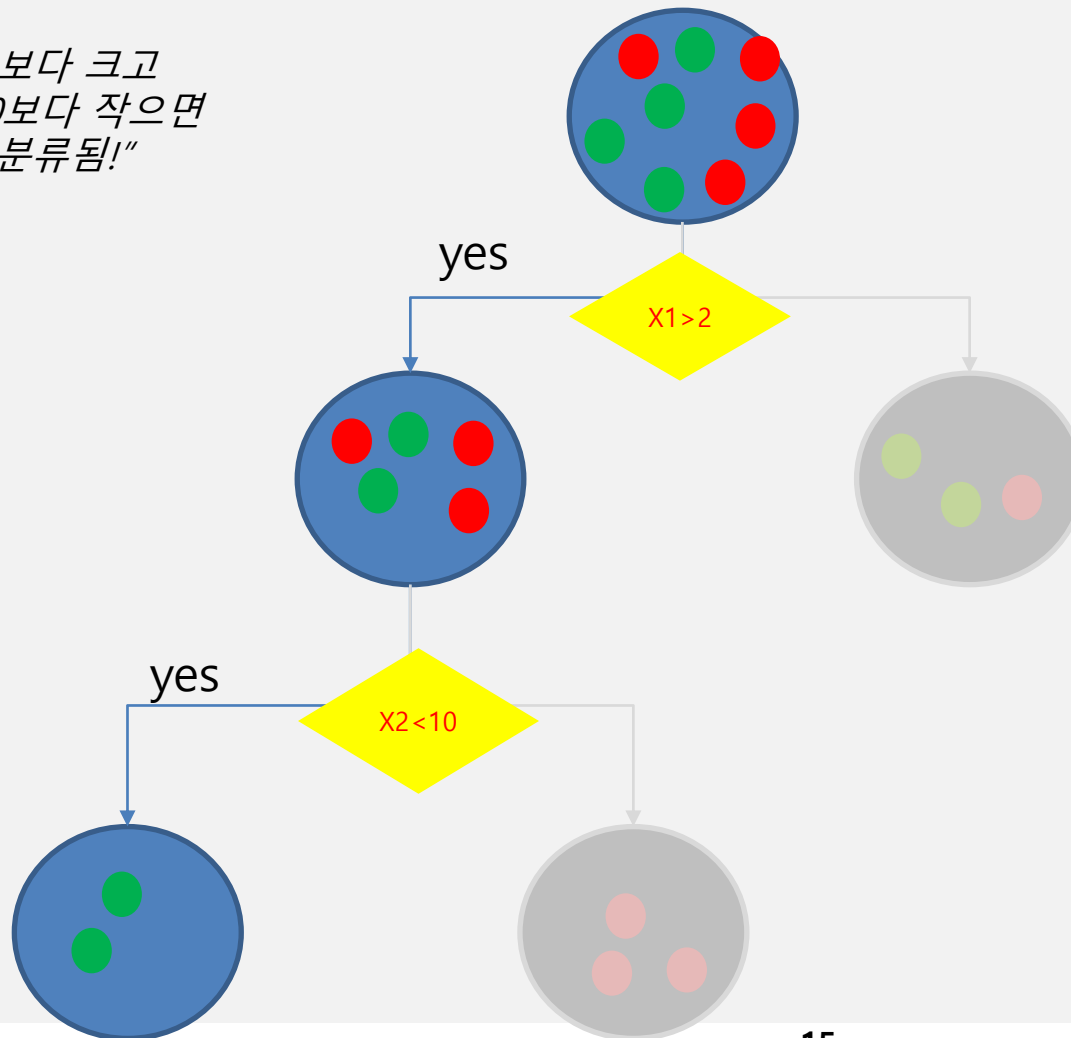
### Decision Tree의 구성요소: Branch & Depth



## 2. 분류 모형의 이해

### Decision Tree의 이해

"X1이 2보다 크고  
X2가 10보다 작으면  
●으로 분류됨!"



- Rule을 통한 Split의 이해
- 여러 Rule들을 통한 변수들의 interaction 이해
- Decision Tree의 출력!

## Decision Tree의 이해





## 2. 분류 모형의 이해

### ➤ 분류모형의 평가

#### – Confusion Matrix

	실제 Y	실제 N
예측 Y	True Positive(TP)	False Positive(FP)
예측 N	False Negative(FN)	True Negative(TN)

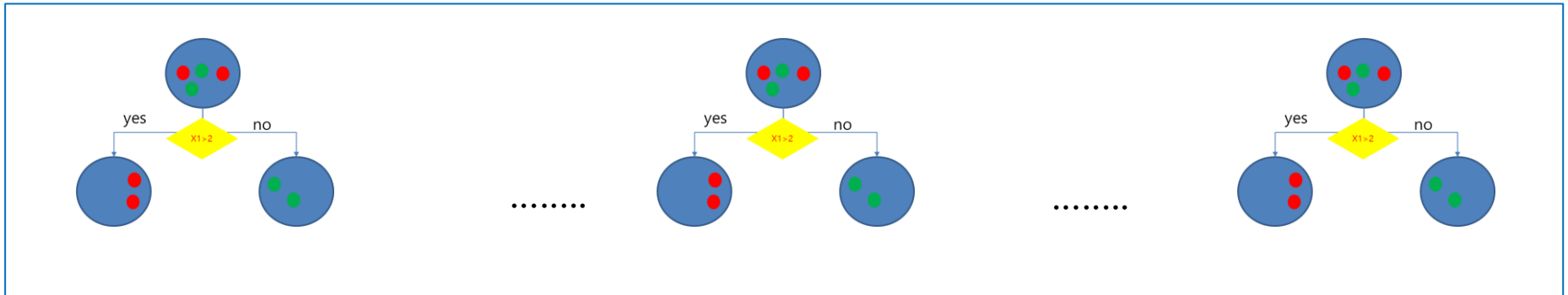
- $N = TP + FP + FN + TN$
- 예측 결과에 따라 True, False 구분
- 예측 값에 따라 Positive, Negative 구분

Metric	Formula	설명
정분류율 or Accuracy	$(TP + TN) / N$	전체 결과 중 맞게 분류한 비율
오분류율	$(FP + FN) / N$	전체 결과 중 잘못 분류한 비율
Precision	$TP / (TP + FP)$	Y로 예측된 것 중 실제로도 Y인 비율

### 3. 다양한 분류 모형

#### Random Forest

- Breiman의 " bagging " 과 변수 랜덤 선택 아이디어 기반
- 처음에는 random decision forests로 시작하여 발전
- 데이터의 다양한 경우를 반영할 수 있도록 보완
- 다양한 경우에 대한 Decision Tree를 통해 성능과 안정성을 제고



### 3. 다양한 분류 모형

---

- Random forest (or random forests)
  - Ensemble classifier that consists of many decision trees and outputs the class that is the mode of the class's output by individual trees
  - Random decision forests : first proposed by Tin Kam Ho of Bell Labs in 1995
  - Combines Breiman's "bagging" idea and the random selection of features



Machine Learning, 45, 5–32, 2001  
© 2001 Kluwer Academic Publishers. Manufactured in The Netherlands.

## Random Forests

LEO BREIMAN

*Statistics Department, University of California, Berkeley, CA 94720*

**Editor:** Robert E. Schapire

**Abstract.** Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. The generalization error for forests converges a.s. to a limit as the number of trees in the forest becomes large. The generalization error of a forest of tree classifiers depends on the strength of the individual trees in the forest and the correlation between them. Using a random selection of features to split each node yields error rates that compare favorably to Adaboost (Y. Freund & R. Schapire, *Machine Learning: Proceedings of the Thirteenth International conference*, \* \* \*, 148–156), but are more robust with respect to noise. Internal estimates monitor error, strength, and correlation and these are used to show the response to increasing the number of features used in the splitting. Internal estimates are also used to measure variable importance. These ideas are also applicable to regression.

**Keywords:** classification, regression, ensemble

### 1. Random forests

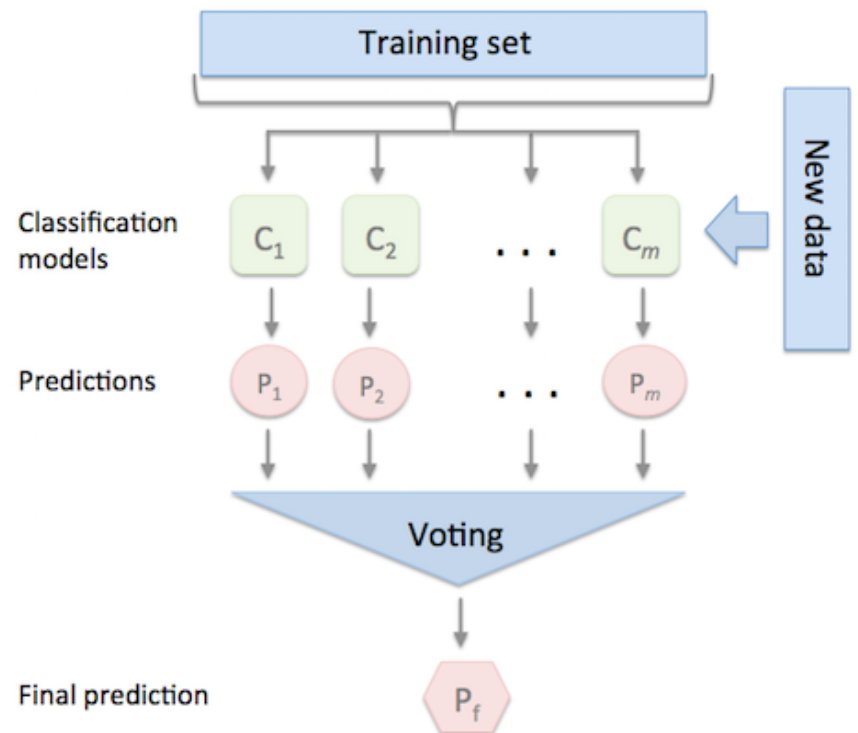
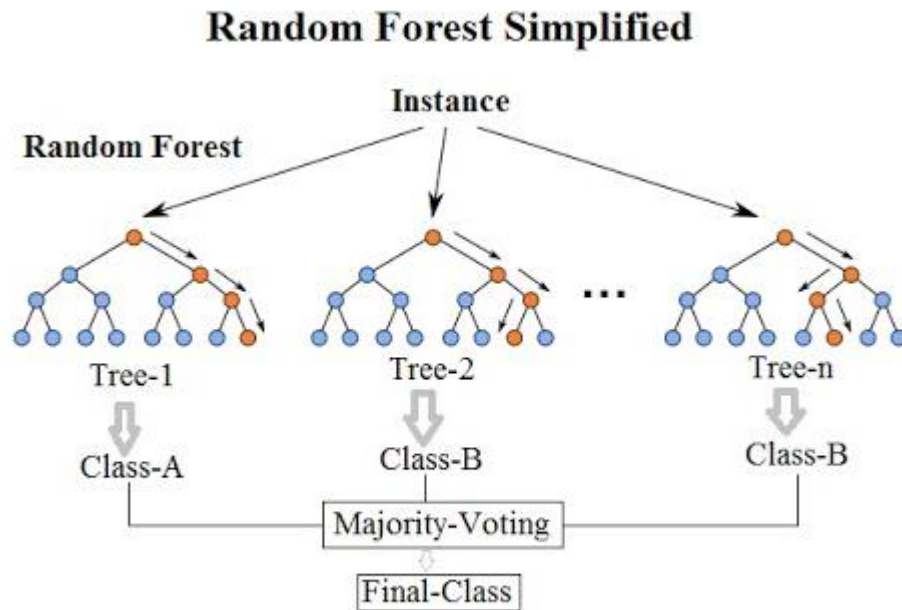
#### 1.1. Introduction

Significant improvements in classification accuracy have resulted from growing an ensemble of trees and letting them vote for the most popular class. In order to grow these ensembles, often random vectors are generated that govern the growth of each tree in the ensemble. An early example is bagging (Breiman, 1996), where to grow each tree a random selection (without replacement) is made from the examples in the training set.

### 3. 다양한 분류 모형

➤ **Random forest (or random forests)**

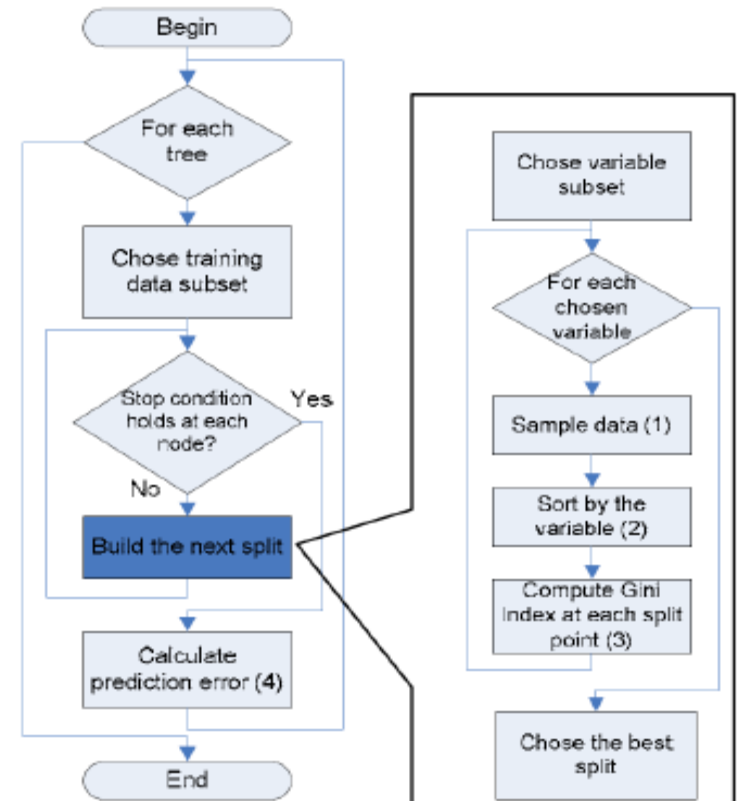
- Ensemble classifier that consists of many decision trees and outputs the class that is the mode of the class's output by individual trees



### 3. 다양한 분류 모형

- Algorithm

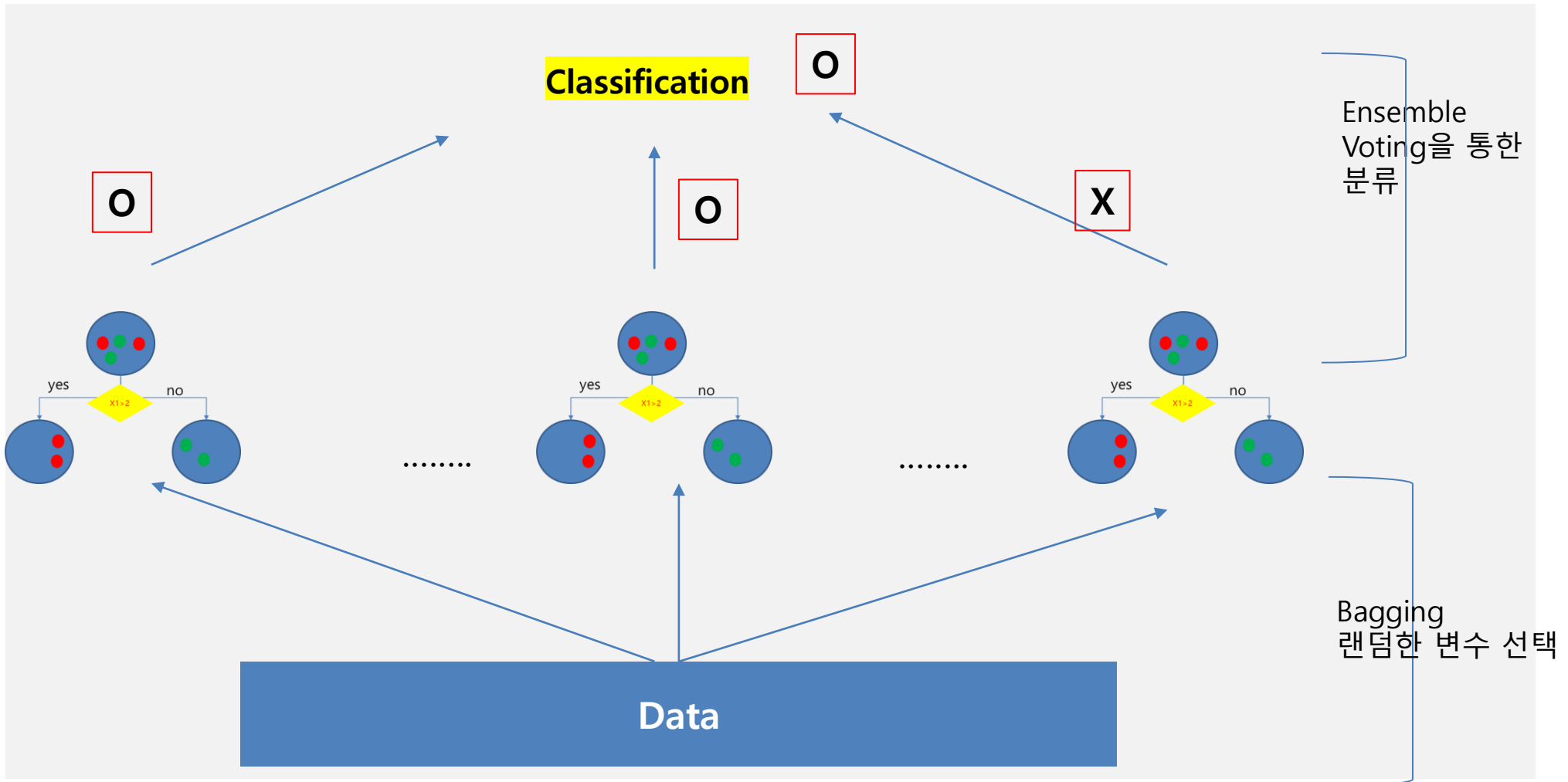
- ①  $N$ : # of training cases /  $M$ : 분류기의 변수
- ②  $M$ 개 중  $m$ 개의 변수가 Tree의 각 노드에서 분류에 사용
- ③  $N$ 개의 training case 중에서 각 tree에 사용되는  $n$ 개의 case를 선택 (예: bootstrap sample). 선택되지 않은 Case는 error 추정에 사용
- ④ 각 tree의 각 노드에서,  $m$ 개의 변수를 무작위 선택하여 분류에 사용. 이후  $m$ 개의 변수로 가장 분류를 잘하도록 계산
- ⑤ 각 Tree **fully grown and not pruned**



### 3. 다양한 분류 모형

#### *Random Forest*

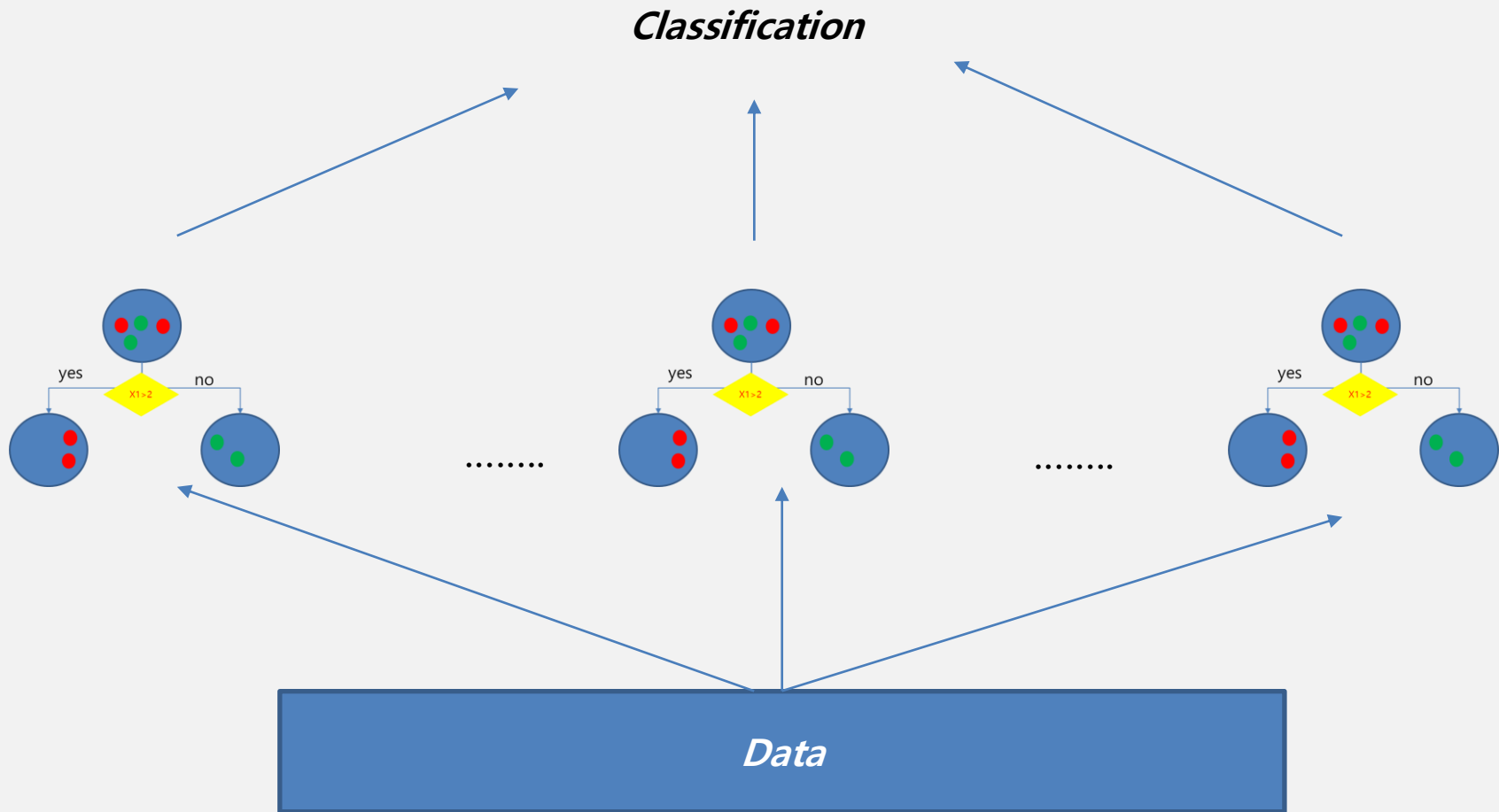
- 데이터의 다양한 경우를 반영할 수 있도록 보완
- 안정성을 제고



### 3. 다양한 분류 모형

#### Random Forest

- 몇 개의 Decision Tree를 만들 것인지?
- 몇 개의 X변수를 Random하게 선택할 것인지?



### 3. 다양한 분류 모형

---

➤ **CART**

- Classification and regression tree.
- CART: greedy, top-down binary, recursive partitioning, that divides feature space into sets of disjoint rectangular regions.
  - Regions should be pure wrt response variable
  - Simple model is fit in each region – majority vote for classification, constant value for regression.

➤ **Random forest (or random forests)**

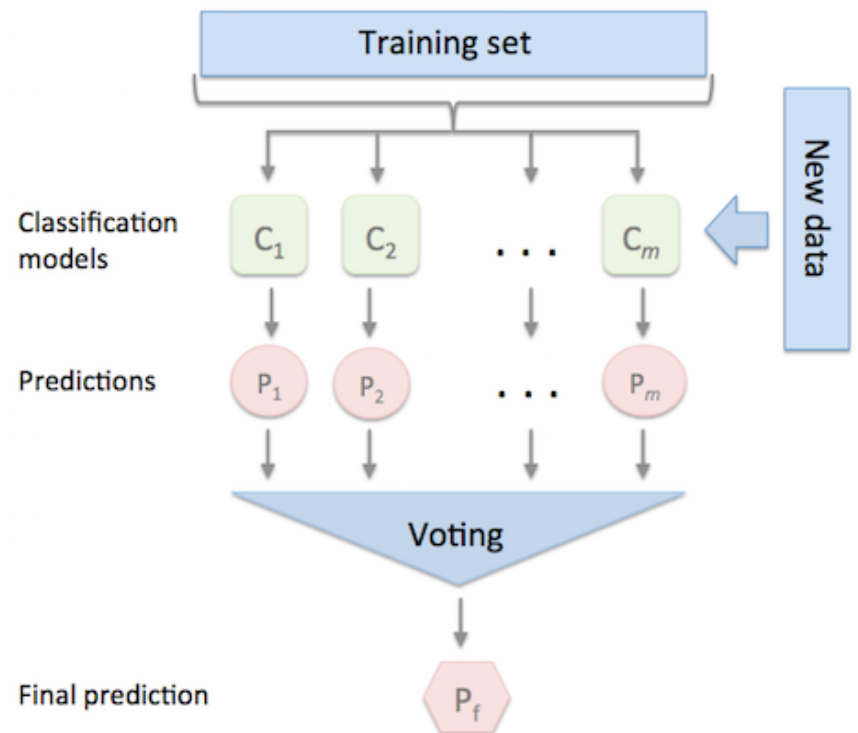
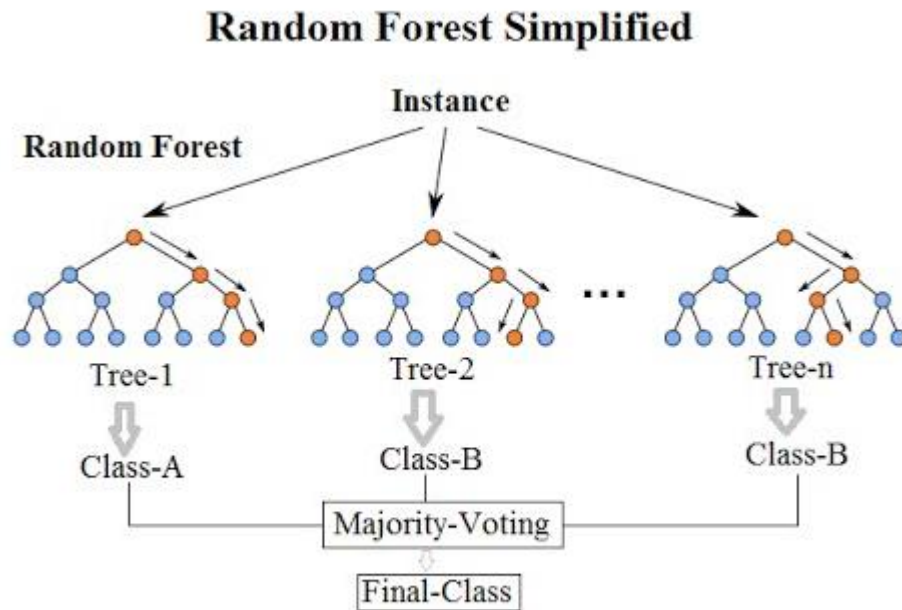
- Ensemble classifier that consists of many decision trees and outputs the class that is the mode of the class's output by individual trees
- **random decision forests** : first proposed by Tin Kam Ho of Bell Labs in 1995
- combines Breiman's "bagging" idea and the random selection of features



### 3. 다양한 분류 모형

➤ **Random forest (or random forests)**

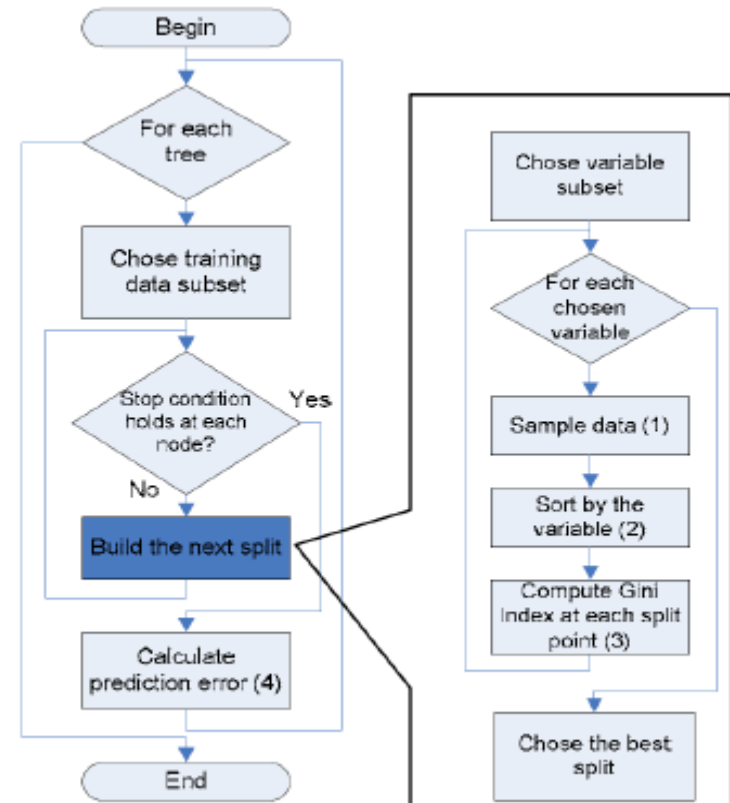
- Ensemble classifier that consists of many decision trees and outputs the class that is the mode of the class's output by individual trees



### 3. 다양한 분류 모형

#### ➤ Algorithm

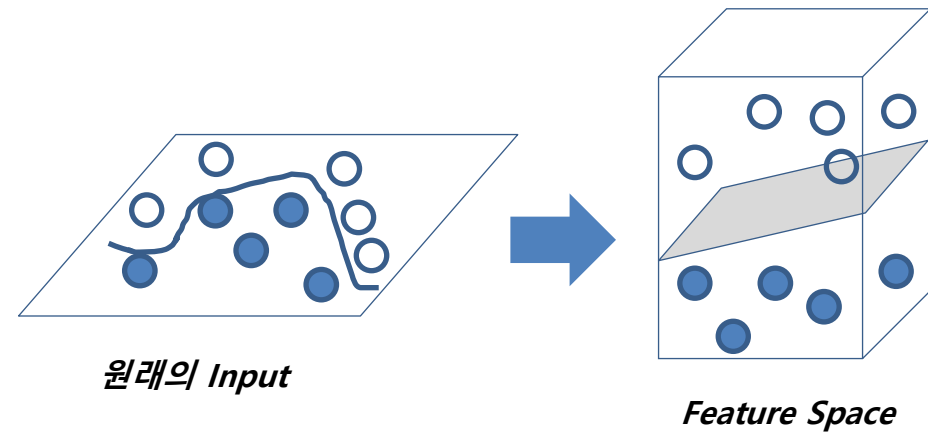
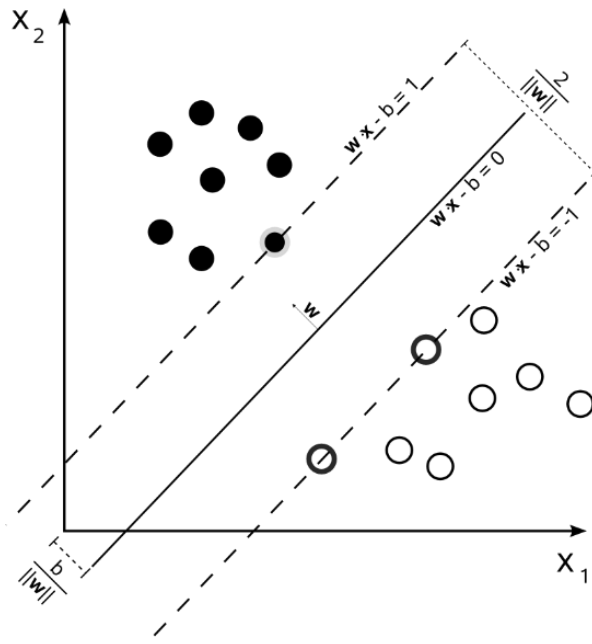
1.  $N$ : # of training cases /  $M$ : 분류기의 변수
2.  $M$ 개 중  $m$ 개의 변수가 Tree의 각 노드에서 분류에 사용
3.  $N$ 개의 training case 중에서 각 tree에 사용되는  $n$ 개의 case를 선택 (예: bootstrap sample). 선택되지 않은 Case는 error 추정에 사용
4. 각 tree의 각 노드에서,  $m$ 개의 변수를 무작위 선택하여 분류에 사용. 이후  $m$ 개의 변수로 가장 분류를 잘하도록 계산
5. 각 Tree **fully grown and not pruned**



### 3. 다양한 분류 모형

#### ➤ Support Vector Machine

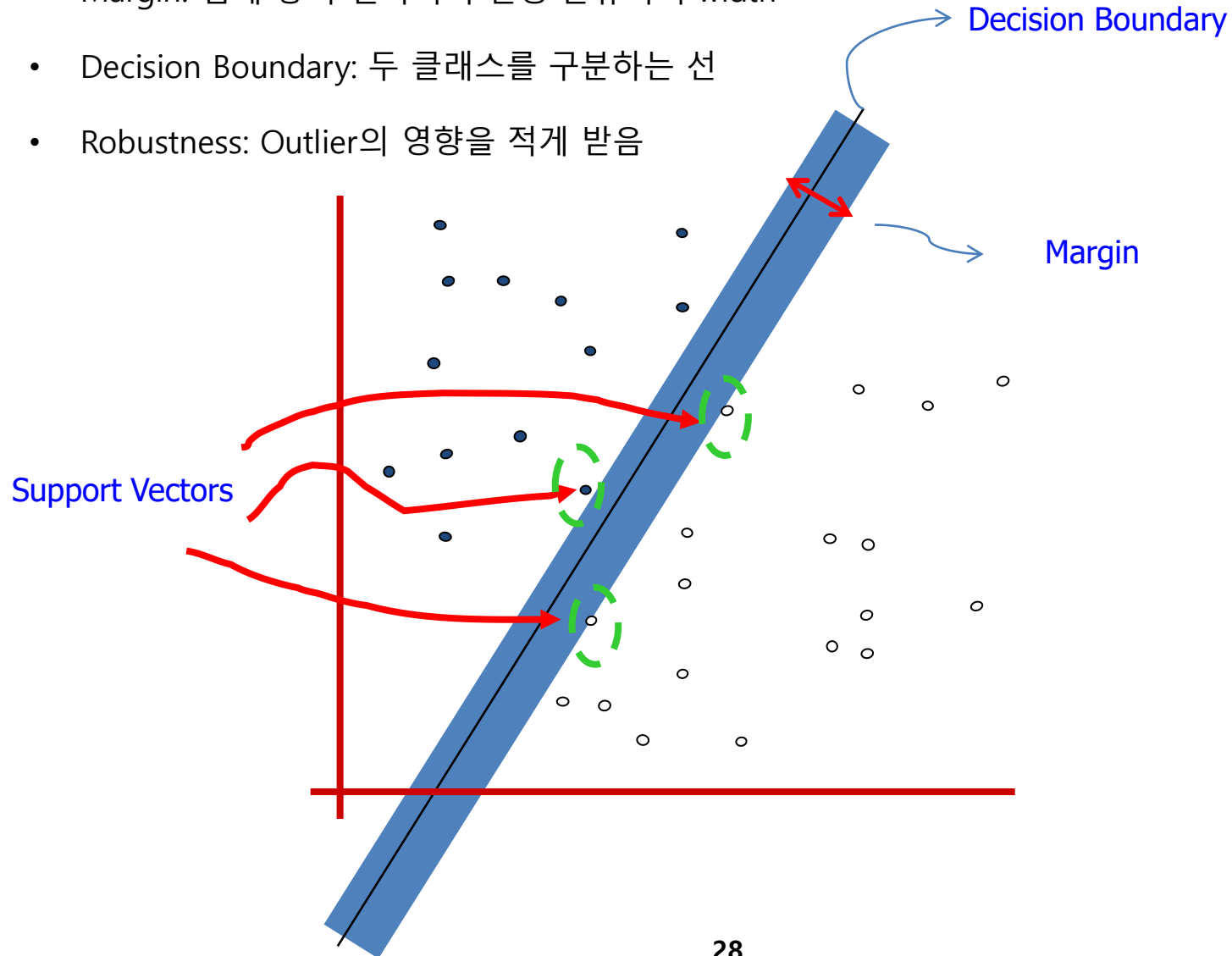
- 1990년대 개발
- *One of the best "out of the box" classifiers*
- Maximal Marginal Classifier의 Generalization 모형



### 3. 다양한 분류 모형

#### ➤ Support Vector Machine 특징

- Margin: 점에 닿기 전까지의 선형 분류기의 width
- Decision Boundary: 두 클래스를 구분하는 선
- Robustness: Outlier의 영향을 적게 받음



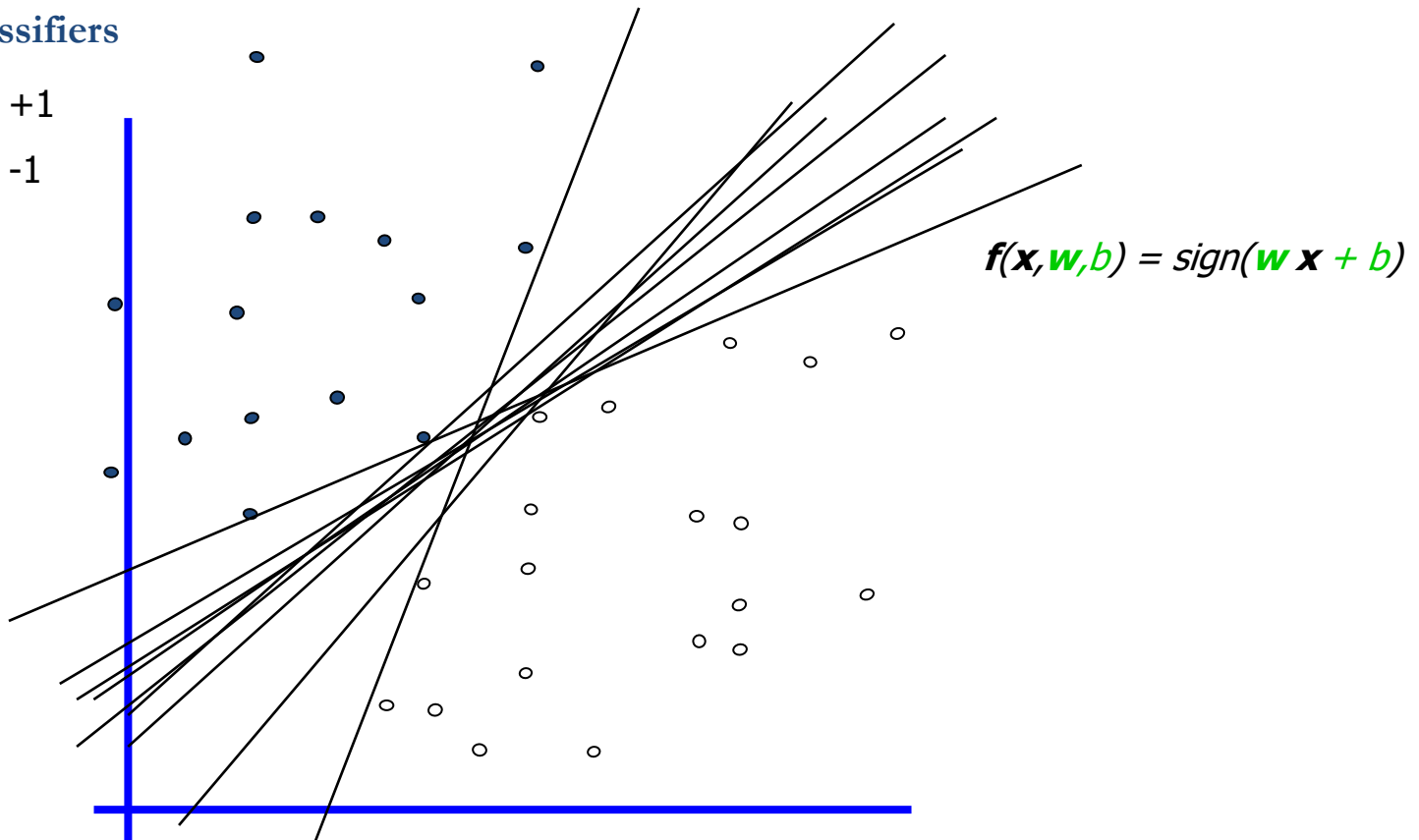
### 3. 다양한 분류 모형

#### ➤ Linear Classifier Example

- 두 종류 점을 구분하는 방법을 고민
- 아래의 직선들로 가능하지만 다양한 가능성이 존재
- 이 중에서 최선의 직선을 찾는다면?

#### Linear Classifiers

- denotes +1
- denotes -1

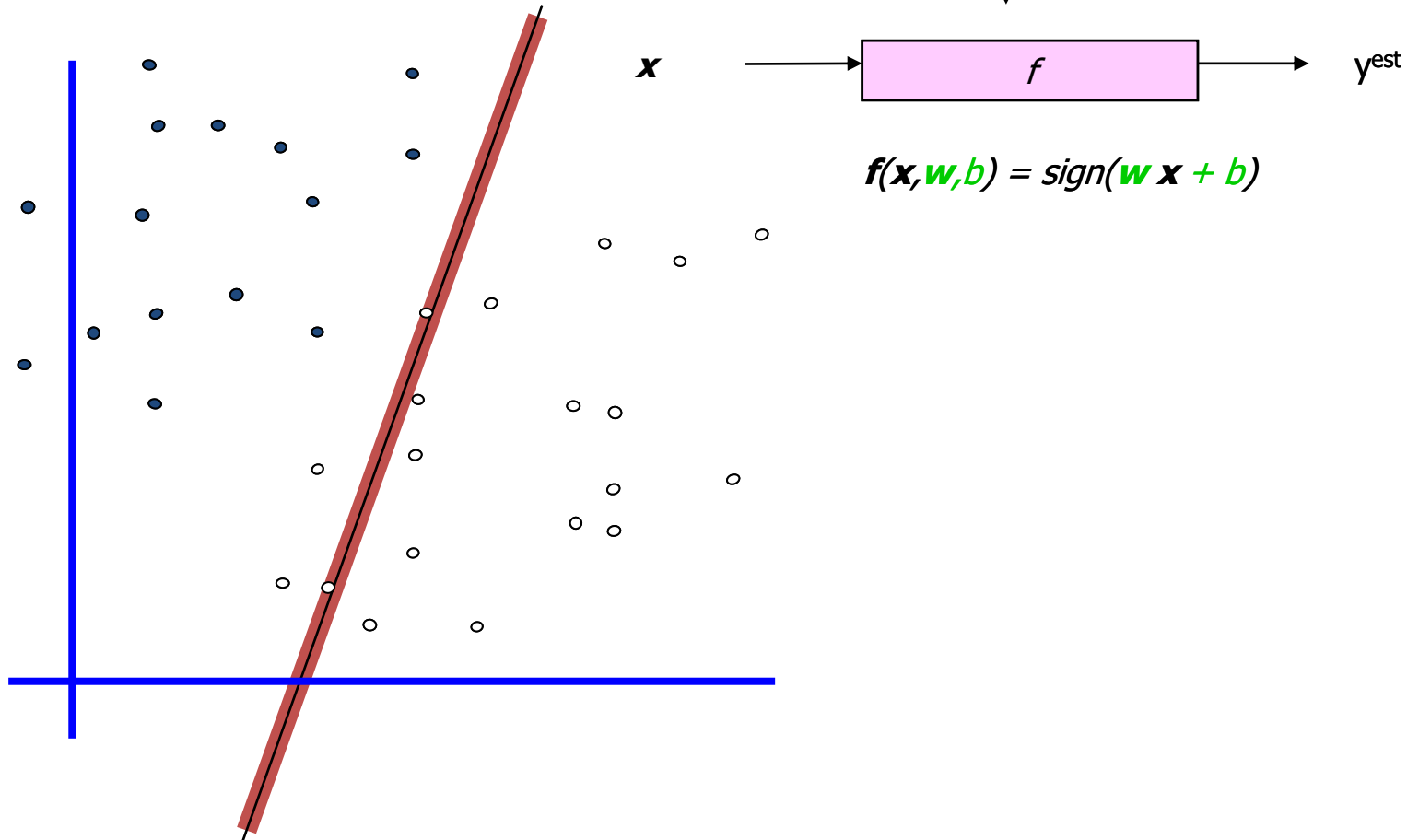


### 3. 다양한 분류 모형

- **Margin:** 점에 닿기 전까지의 선형 분류기의 **width**

#### Classifier Margin

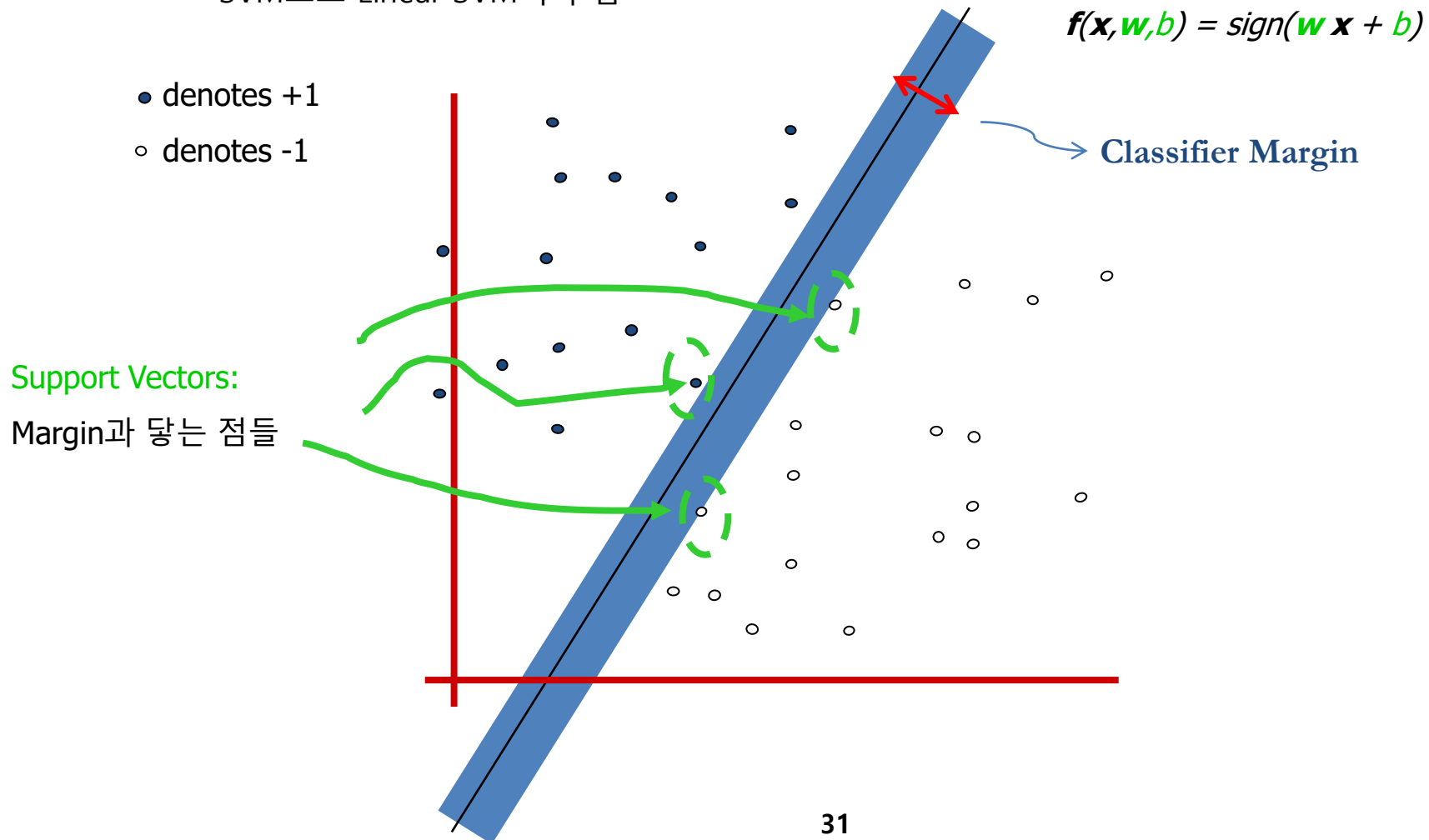
- denotes +1
- denotes -1



### 3. 다양한 분류 모형

#### ➤ Support Vector Machine

- ✓ Margin: 점에 닿기 전까지의 선형 분류기의 width
- ✓ Maximum margin linear classifier : 최대 margin을 갖는 선형 분류기, 특히 이것은 가장 단순한 형태의 SVM으로 Linear SVM이라 함

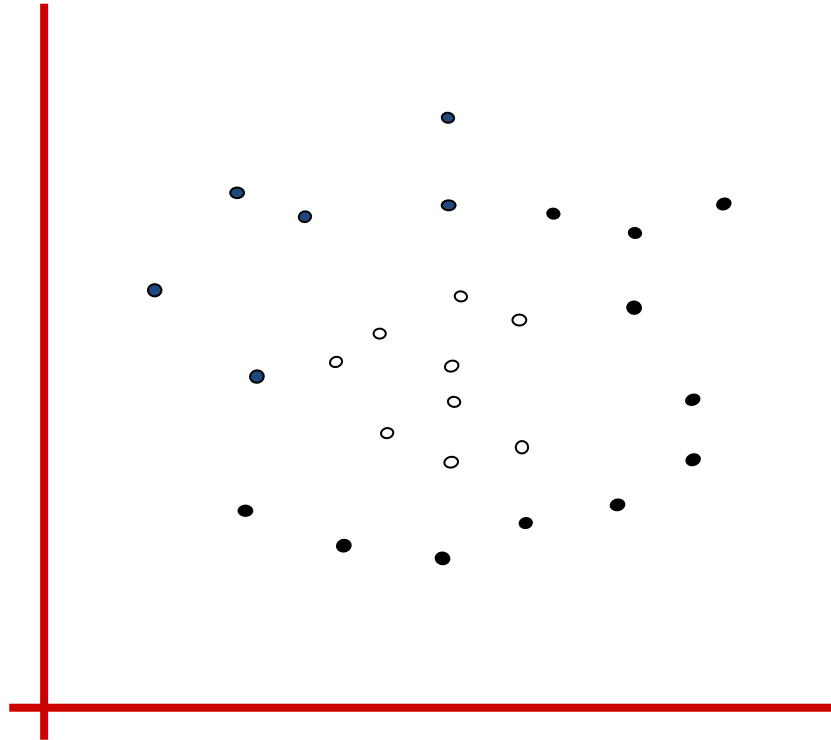


### 3. 다양한 분류 모형

---

➤ **Support Vector Machine**

- Linear Classifier의 한계: Non-linear separation!

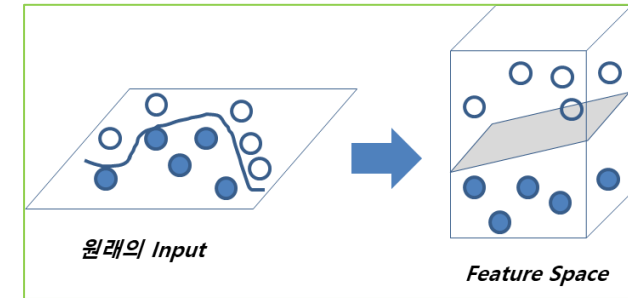




# 1. 다양한 분류 모형과 SVM

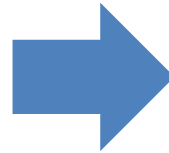
## ➤ Support Vector Machine

- 차원의 확장
- $Z = X_1^2 + X_2^2$



X2

Z



X1

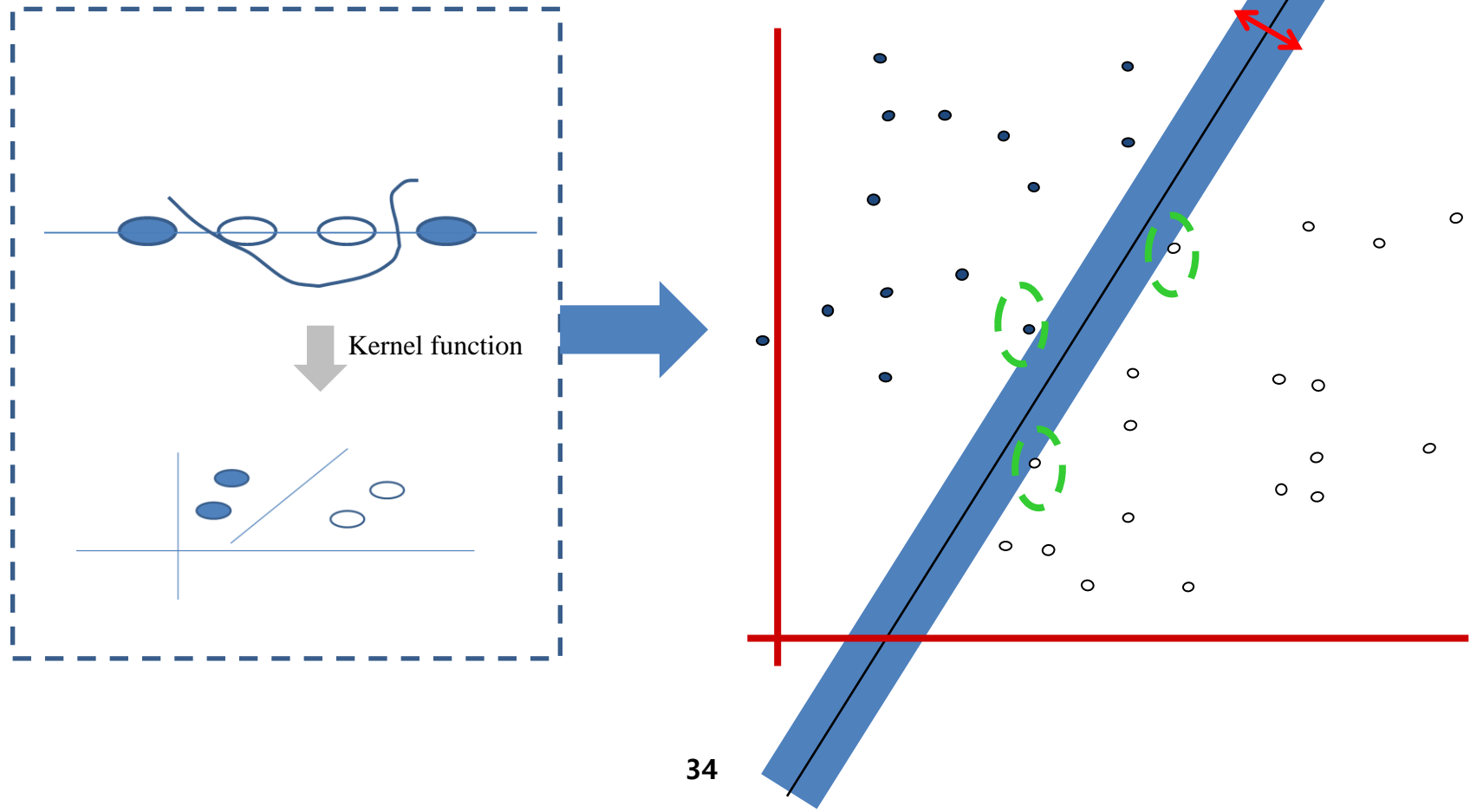
X1

### 3. 다양한 분류 모형

#### ➤ Support Vector Machine

##### ✓ Kernel Trick

- ✓ 모든 데이터를 항상 초평면 또는 선으로 나눌 수 없음
- ✓ 이 경우 주어진 자료를 평면으로 표현할 수 있는 고차원을 변환
- ✓ 저차원 공간을 고차원 공간으로 변환: 고차원 공간에서 Linear 해를 구한 후 저차원 공간의 해를 구함



### 3. 다양한 분류 모형

---

- Support Vector Machine

- ✓ Parameter

- ✓ Kernel: Linear, Polynomial, Sigmoid, RBF...

- ✓ C: Decision Boundary의 굴곡(큰 C) 또는 직선(작은 C)을 결정

- ✓ Gamma

- ✓ Decision Boundary에 영향을 주는 데이터 범위, 클수록 Reach가 좁아 Decision Boundary 근처의 데이터에 영향을 받음

- ✓ Gamma가 큰 경우: Reach가 좁아서 더 적은 데이터에 영향, 굴곡진 Decision Boundary

- ✓ Gamma가 작은 경우: Reach가 멀어서 더 많은 데이터에 영향, 직선에 가까운 Decision Boundary

- ✓ Overfitting의 이슈 & 상대적으로 긴 Training Time

### 3. 다양한 분류 모형

---

#### ➤ 베이즈 정리

- 확률변수의 조건부(conditional) 확률분포와 주변부(marginal) 확률분포를 연관 짓는 정리. 즉, 새로운 자료에서 나온 확률에 기반하여 과거의 확률을 향상(update).

$$P(A|B) = \frac{P(B|A) P(A)}{P(B)}$$

#### ➤ 베이즈 정리 이용 예

- 어떤 공장에서 일반적으로 공장이 원활한 경우 95%의 양품을 생산하지만, 공장이 원활하게 운영이 되지 않는 경우 70%의 양품을 생산
- 공장 관리자는 생산되는 제품의 품질을 바탕으로 공장 운영이 원활한지를 모니터링하고, 이를 공장 운영에 반영
- 정리
  - O: 공장이 원활하게 운영
  - OC: 공장이 원활하게 운영되지 않음
  - S: 양품 생산
  - SC:불량 생산
  - $P(S|O)$ 와  $P(S|OC)$ 를 알고 있음
- $P(O|S)=?$ 
  - 이 확률을 바로 구할 수 없으므로 베이즈 정리 이용
  - $P(O|S) = P(S|O)P(O) / (P(S|O)P(O)+P(S|OC)P(OC))$

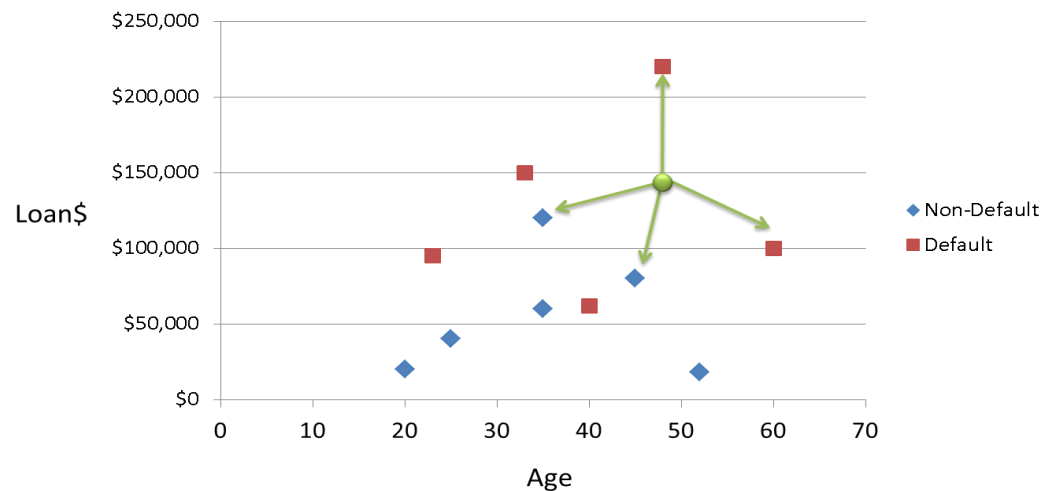
### 3. 다양한 분류 모형

#### ➤ KNN

- 1970년대 시작되었으며, 비모수적 기법
- 모든 가능한 케이스를 저장하고, 새로운 케이스를 유사도 기반하여 분류
- 모든 케이스는  $n$ 차원의 공간에서 점과 대응되며, 유클리드 혹은 맨해튼 거리 관점에서 인접한 이웃이 정의됨
- 이산형 및 연속형 가능

#### - 여러 명칭들

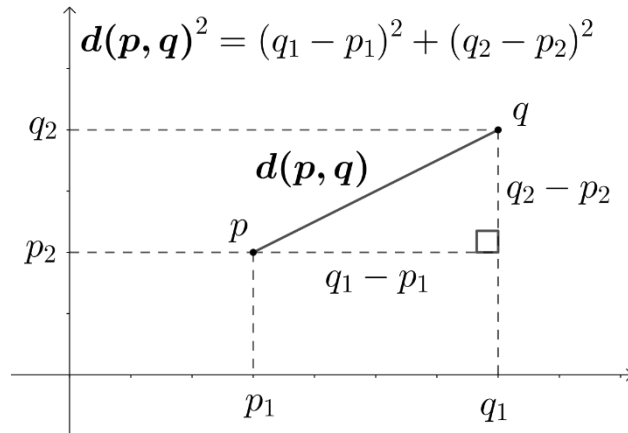
- K-Nearest Neighbors
- Memory-Based Reasoning
- Example-Based Reasoning
- Instance-Based Learning
- Case-Based Reasoning
- **Lazy Learning**



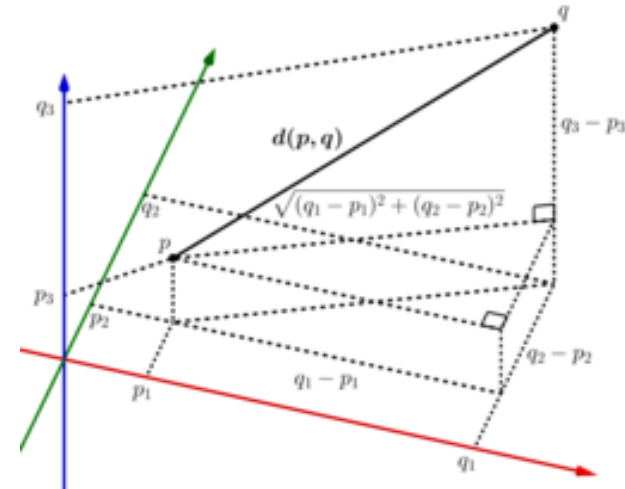
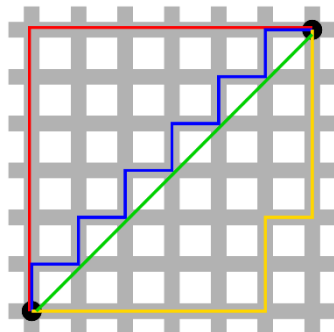
### 3. 다양한 분류 모형

#### ➤ 유클리드 거리와 맨해튼 거리 비교

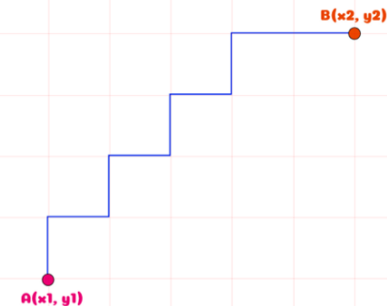
- 유클리드 거리(Euclidean Distance)



- 맨해튼 거리(Manhattan Distance)
  - 격자 모양의 경로에서 측정된 거리



$$\text{Manhattan}(A, B) = |x_1 - x_2| + |y_1 - y_2|$$



### 3. 다양한 분류 모형

---

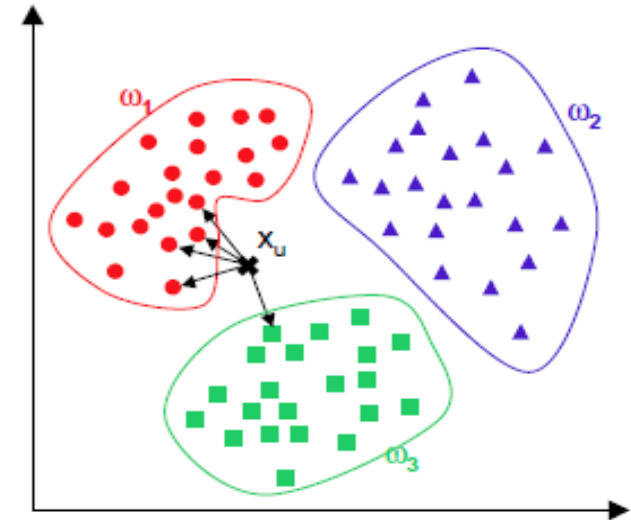
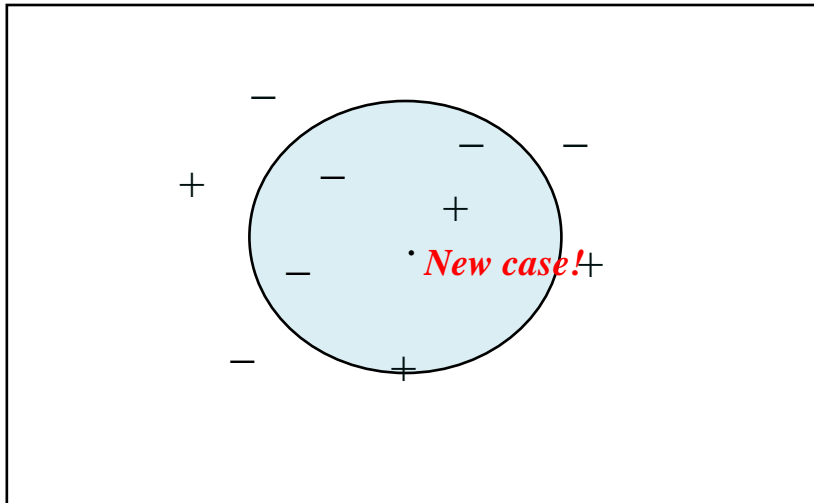
#### ➤ KNN

- Bayes Classifier의 한 종류
- K-Nearest Neighbor
  - Bayes Rule? 주어진  $x$ 에 대한  $y$ 의 확률
    - ***Classify observation to the class with largest probability***
  - 간단한 방법 & 우수한 성능
  - 최적의  $K$ 를 구하기
- **K?**
  - If  $K=1$ , select the nearest neighbor
  - If  $K>1$ ,
    - For classification select the most frequent neighbor.
    - For regression calculate the average of  $K$  neighbors.
  - $K$ 는 주로 홀수로 선택

### 3. 다양한 분류 모형

#### ➤ KNN

- 연속형 값에 대한 k-NN은 k개의 인접 이웃의 평균 값을 반환
  - Distance-weighted nearest neighbor algorithm
  - 각 k개의 이웃이 새로운 케이스에 얼마나 기여하는지를 거리를 기반으로 산출
- **Robust to noisy data** by averaging k-nearest neighbors






### 3. 다양한 분류 모형

---

- Naïve Bayes 분류기
  - **베이즈 정리 활용**
  - **Conditional independence** assumption: Feature끼리는 서로 독립!
  - 쉽고 빠른 학습이 가능
  - 이미 계산된 조건부 확률에 의해 예측이 이뤄짐
- Naïve Bayes 분류기: popular **generative** model!
  - 비교적 성능이 우수
  - 앙상블 학습에서 Base 학습기로 잘 활용됨



두 사건  $A, B$ 에 대해  $P(A \text{ and } B) = P(A)P(B)$ 이면  $A$ 와  $B$ 는 서로 독립

### 3. 다양한 분류 모형

---

➤ **Bayes classification**

- Difficulty: learning the joint probability

$$P(C | \mathbf{X}) \propto P(\mathbf{X} | C)P(C) = P(X_1, \dots, X_n | C)P(C)$$

➤ **Naïve Bayes classification**

- 가정: all input features are conditionally independent! (이런 가정이 Naïve)
- Example:

$$P(X_1, \dots, X_n | C)$$

$$\begin{aligned} P(X_1, X_2, \dots, X_n | C) &= P(X_1 | X_2, \dots, X_n, C)P(X_2, \dots, X_n | C) \\ &= P(X_1 | C)P(X_2, \dots, X_n | C) \\ &= P(X_1 | C)P(X_2 | C) \dots P(X_n | C) \end{aligned}$$

$$\mathbf{x} = (x_1, x_2, \dots, x_n)$$

$$[P(x_1 | c^*) \dots P(x_n | c^*)]P(c^*) > [P(x_1 | c) \dots P(x_n | c)]P(c), \quad c \neq c^*, c = c_1, \dots, c_L$$

### 3. 다양한 분류 모형

- Spam 필터링

Keyword	Target
대출	Spam
업무	Normal
할인	Spam
대출	Spam
연락	Normal
투자	Spam



Target	Spam	Normal	
대출	2	0	2/6
업무	0	1	1/6
할인	1	0	1/6
연락	0	1	1/6
투자	1	0	1/6
Total	4	2	
	4/6	2/6	

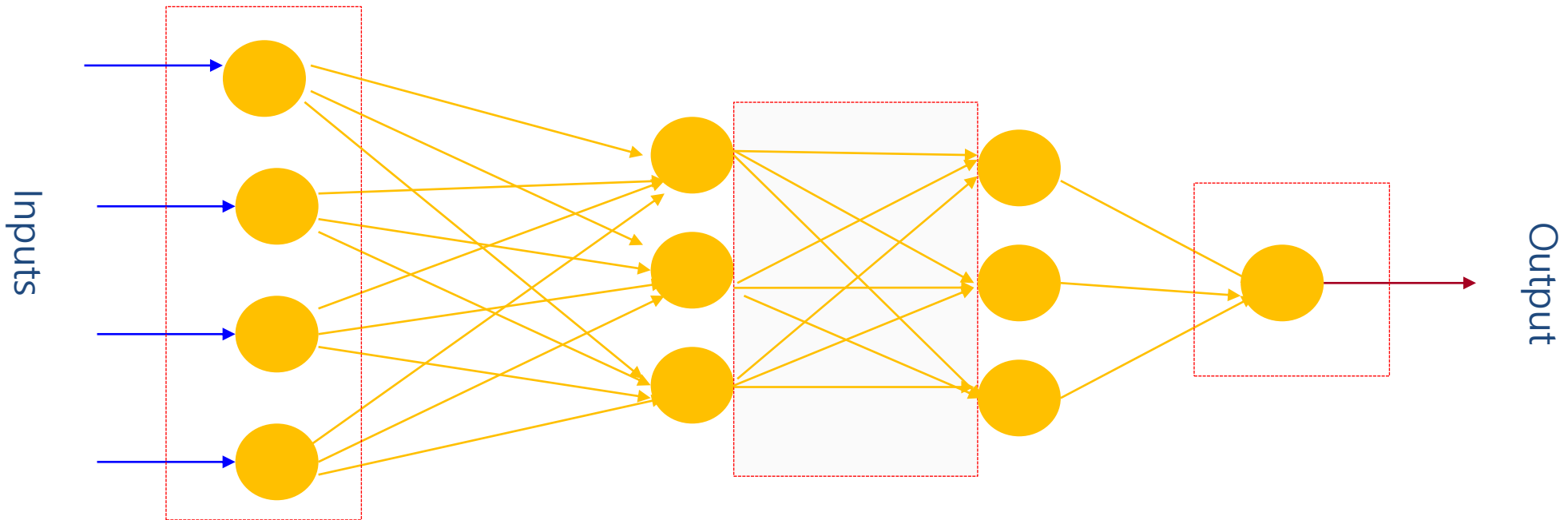


Target	Spam	Normal	Spam일 때의 확률	Normal일 때의 확률
대출	2	0	2/4	0
업무	0	1	0	1/2
할인	1	0	1/4	0
연락	0	1	0	1/2
투자	1	0	1/4	0
Total	4	2		

$$\begin{aligned}
 P(\text{Spam}|\text{대출}) &= P(\text{대출}|\text{Spam}) \times P(\text{Spam}) / P(\text{대출}) \\
 &= 2/4 \times 4/6 / (2/6)
 \end{aligned}$$

### 3. 다양한 분류 모형

- **Artificial Neural Network(ANN, 인공신경망)**
  - 생물학의 신경망(동물의 중추신경계중 특히 뇌의 뉴런)을 모사한 학습 알고리즘
  - 뉴런을 모방한 노드들이 각각 Input Layer, Hidden Layer, Output Layer로 구분되며 데이터를 입력받아 변환하여 원하는 결과로 출력하는 네트워크를 구축
  - 예측 성능이 우수하다고 알려진 반면, 모형을 직관적으로 이해하기가 어려움



노드 간의 연결

## 4. Artificial Neural Network로의 확장

인공신경망을 행렬로 표현하기!

Feature

a	b	c
4	3	2
2	9	1
3	3	2
2	4	1

4	3	2
2	9	1
3	3	2
2	4	1

2	9	1
3	3	2
2	4	1

2	4	1
---	---	---

w1	w2
w3	w4
w5	w6

w1

w2

w3

w4

w5

w6

a

b

c

h1

h2

h1	h2
$4xw1 + 3xw3 + 2xw5$	$4xw2 + 3xw4 + 2xw6$
$2xw1 + 9xw3 + 1xw5$	$2xw2 + 9xw4 + 1xw6$
$3xw1 + 3xw3 + 2xw5$	$3xw2 + 3xw4 + 2xw6$
$2xw1 + 4xw3 + 1xw5$	$2xw2 + 4xw4 + 1xw6$

Target

y

O

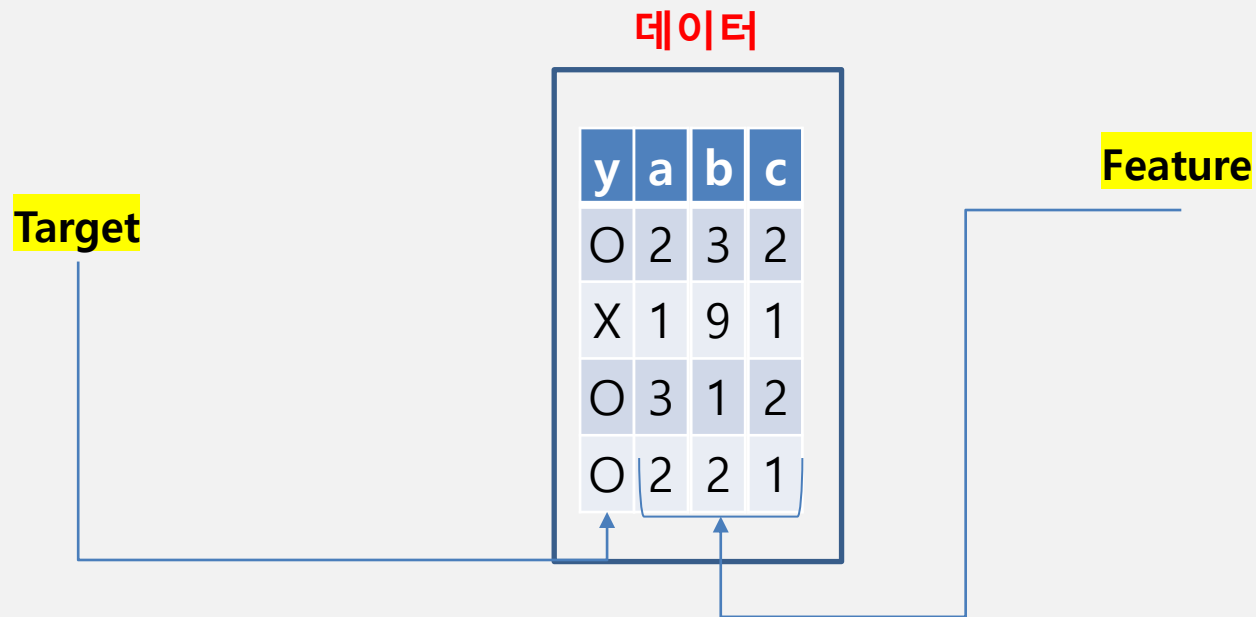
X

O

O

## 4. Artificial Neural Network로의 확장

### ANN Step by Step

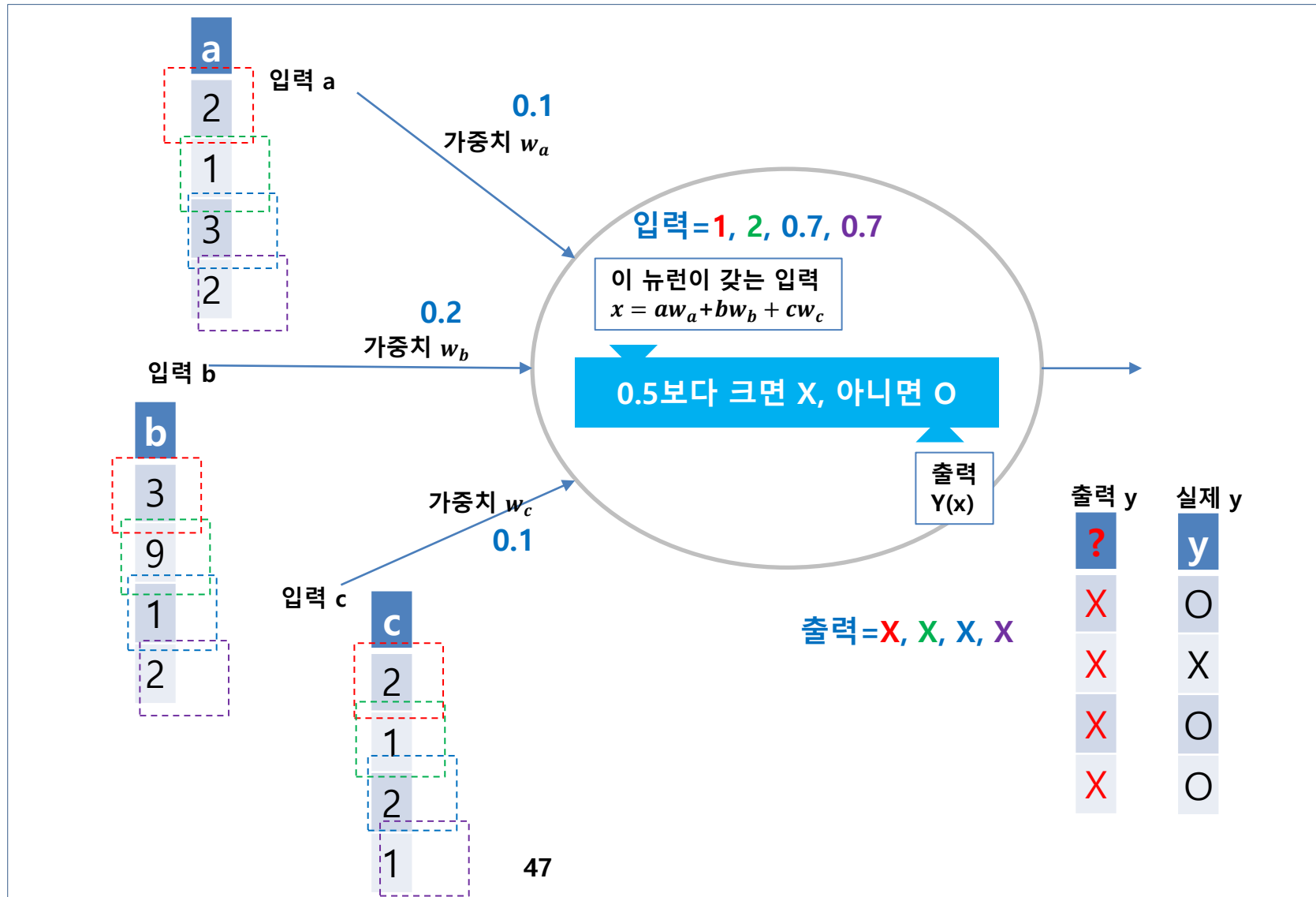


*Supervised Learning-Classification*

## 4. Artificial Neural Network로의 확장

### ANN Step by Step

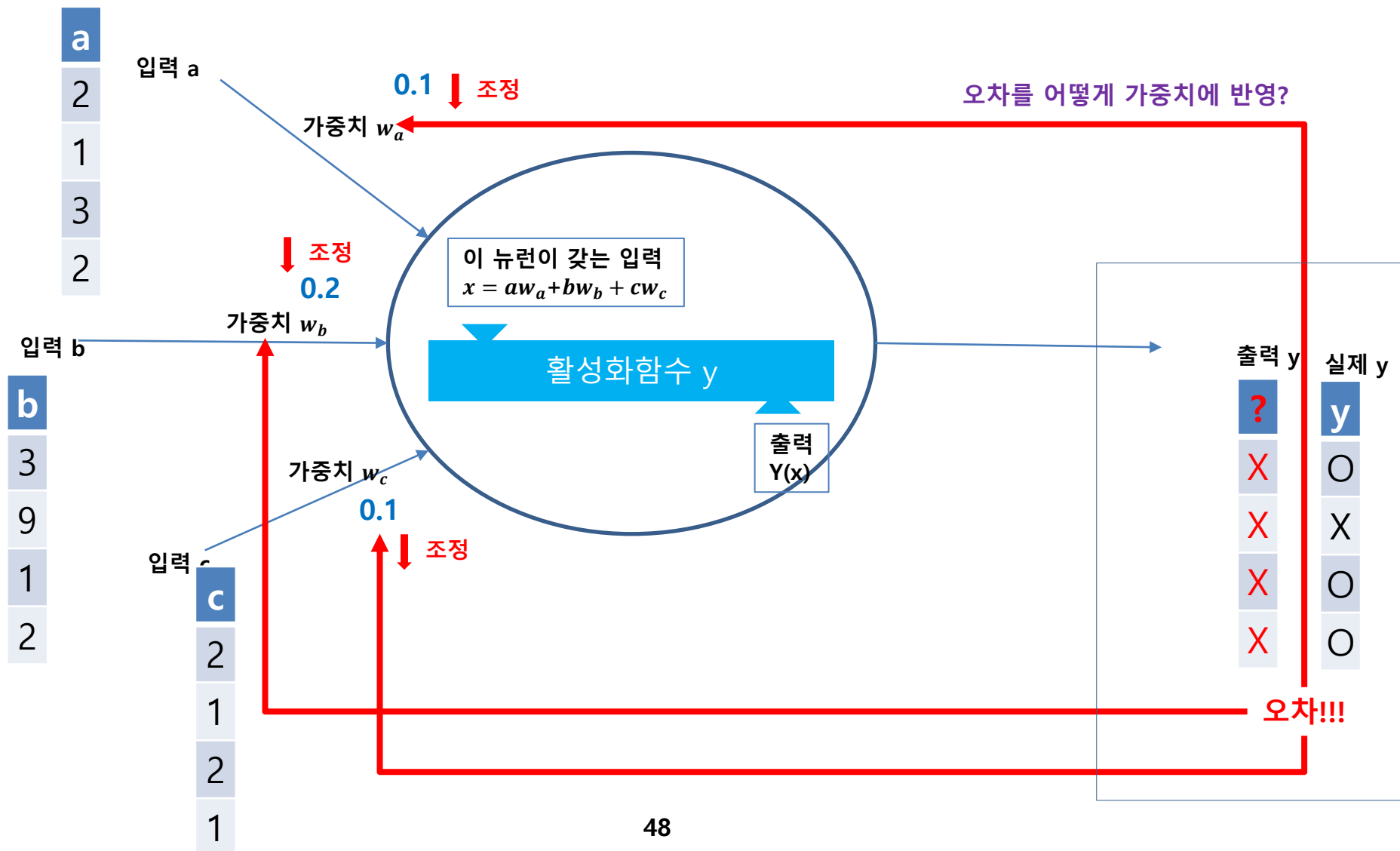
#### 인공신경망 1단계: 전파!



## 4. Artificial Neural Network로의 확장

### ANN Step by Step

#### 인공신경망 2단계: 오차의 역전파!

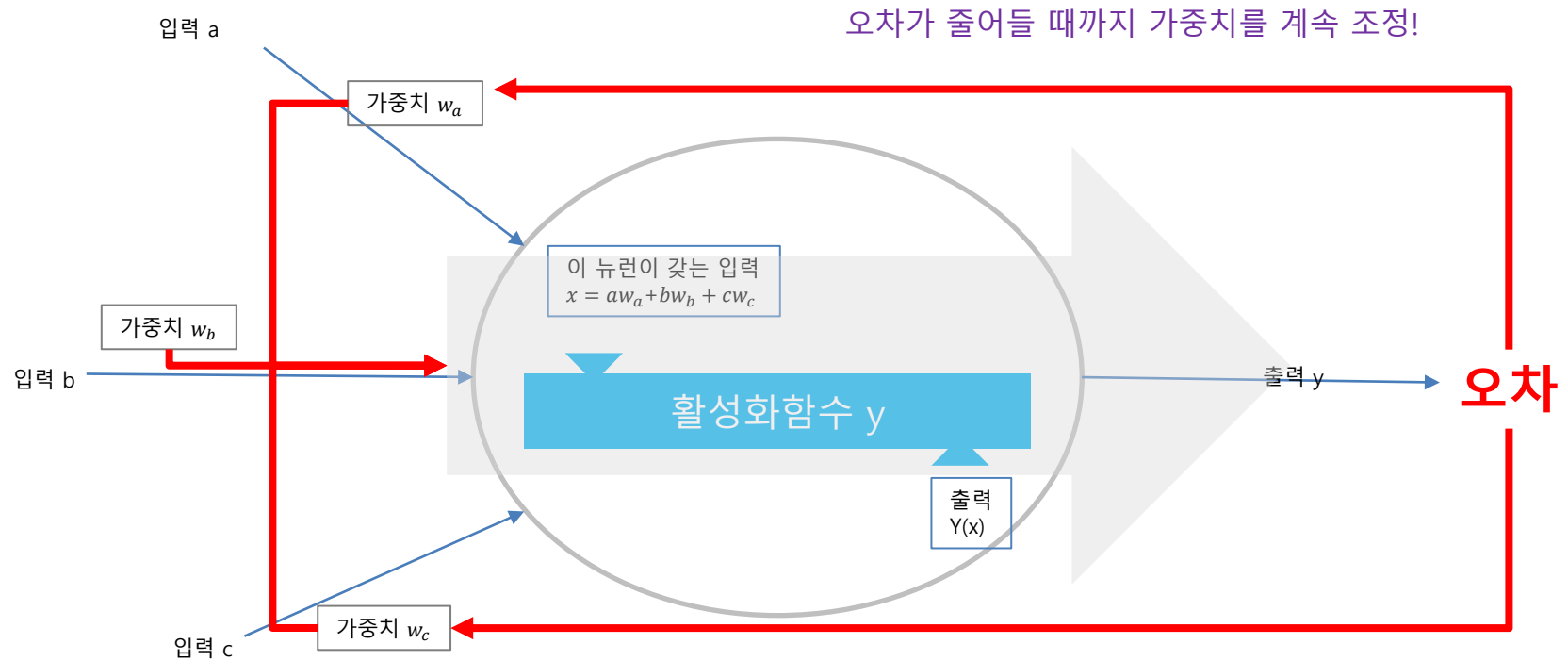




## 4. Artificial Neural Network로의 확장

### ANN Step by Step

잘 할 때까지 반복!

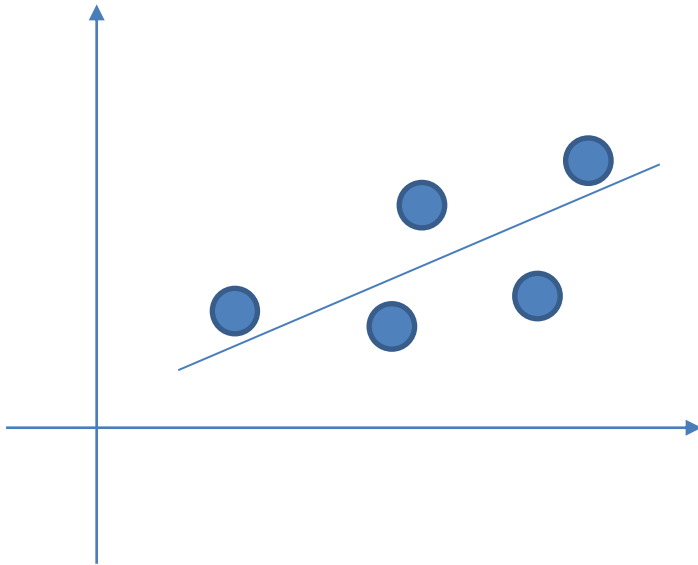


## 4. Artificial Neural Network로의 확장

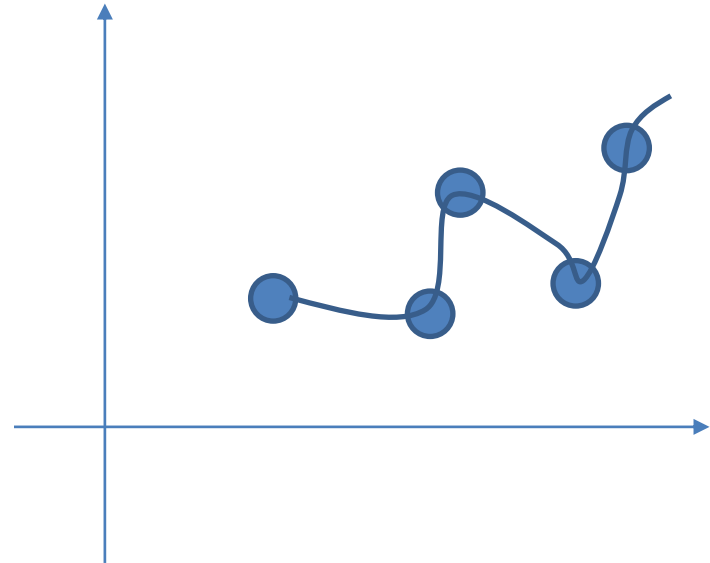
### Overfitting?

- 피팅(Fitting, 적합)이란, 주어진 데이터를 모델링하는 과정
- 주어진 데이터에만 과도하게 피팅된 것이 오버피팅
- Feature가 많고, 데이터가 많으면 겪는 이슈

일반적인 Fitting



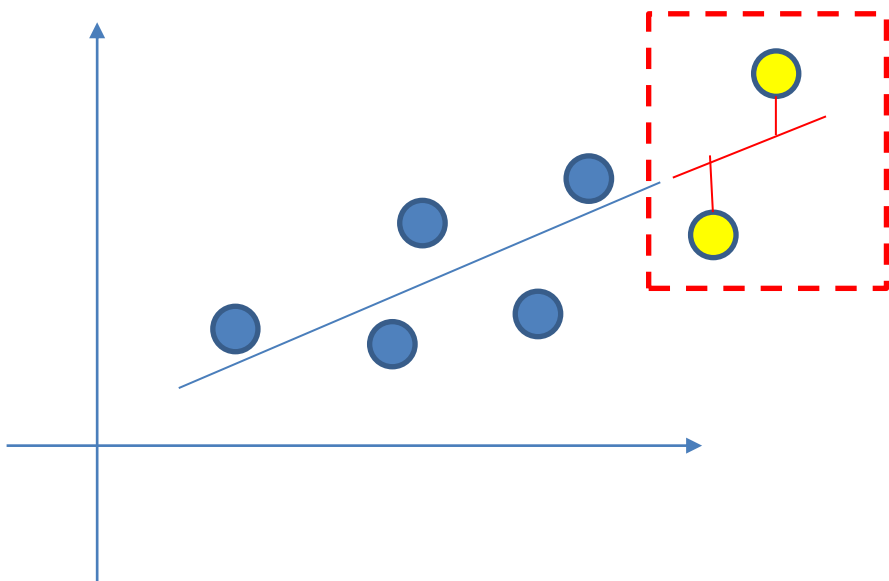
Over Fitting



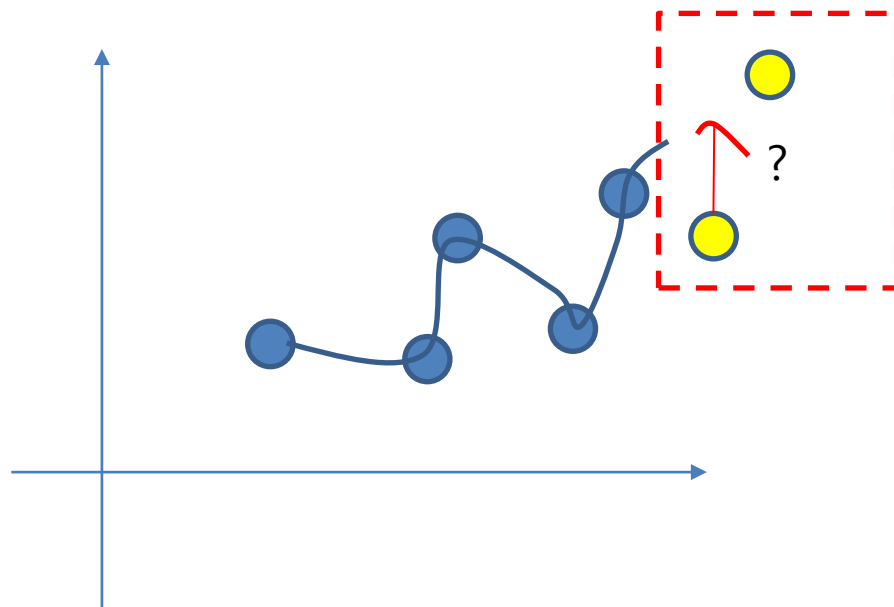
## 4. Artificial Neural Network로의 확장

Overfitting?

새로운 데이터에 대한 예측과 오차

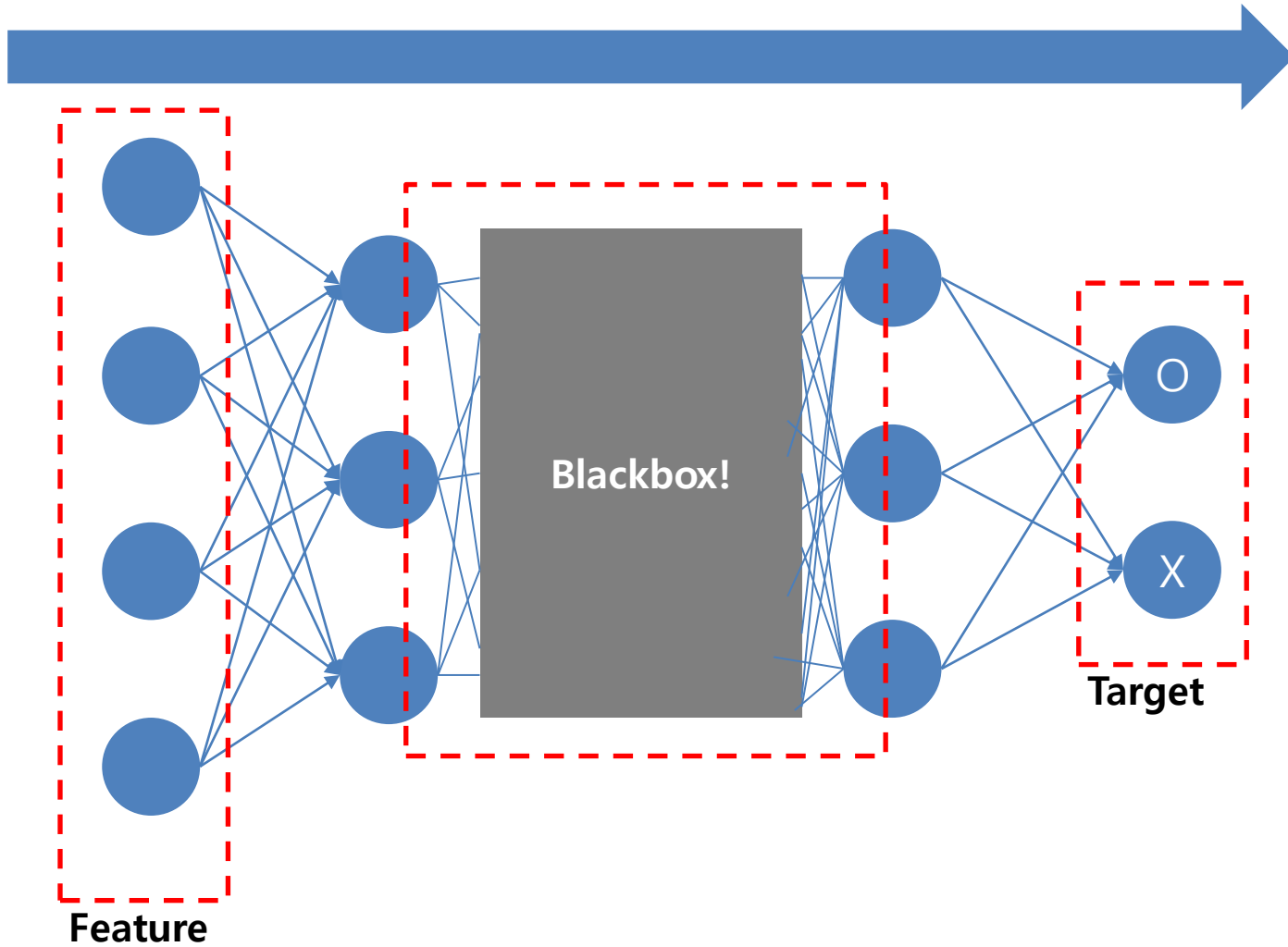


Over Fitting 시  
새로운 데이터에 대한 예측과 오차



## 4. Artificial Neural Network로의 확장

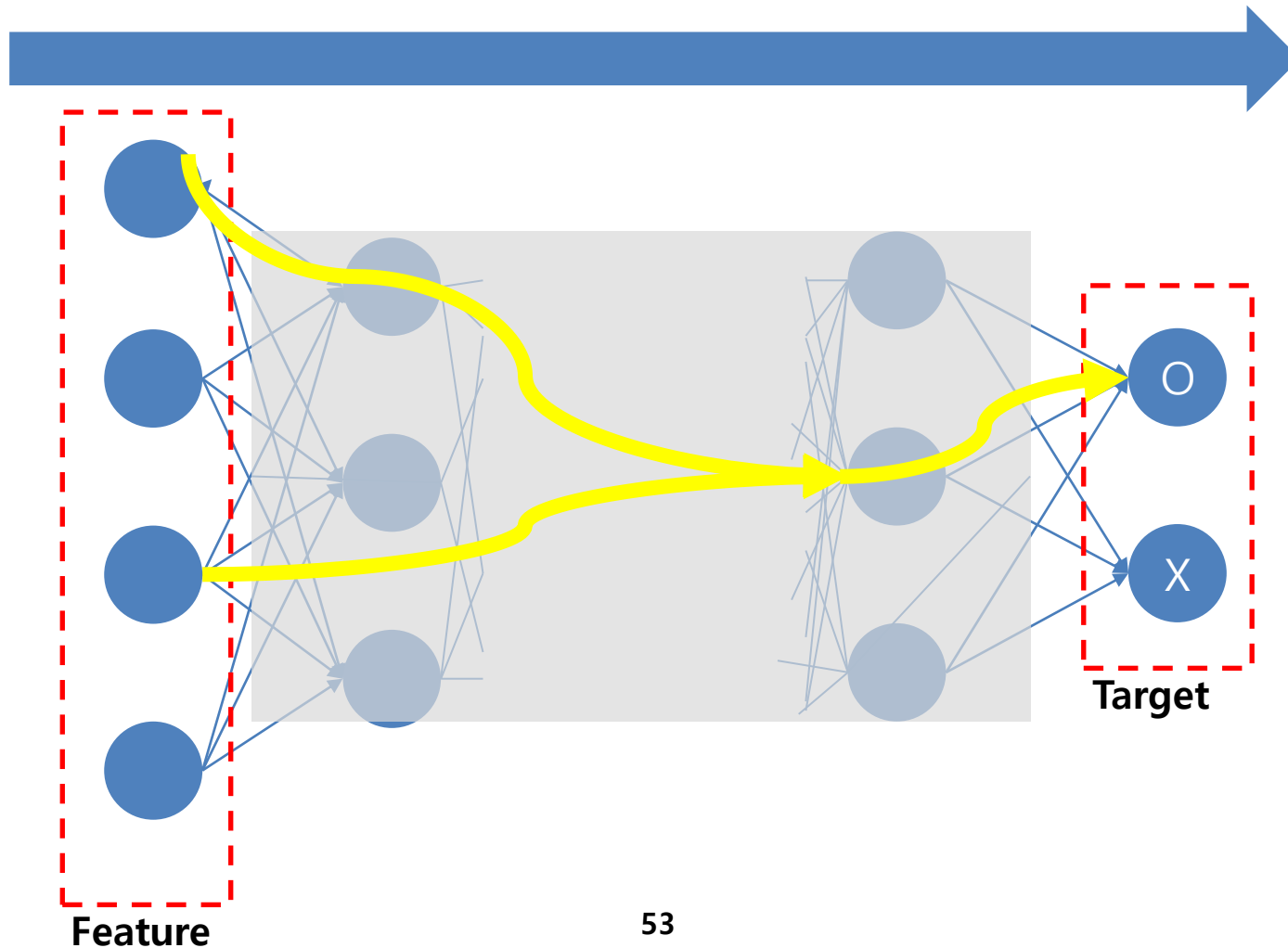
딥러닝=Blackbox model



## 4. Artificial Neural Network로의 확장

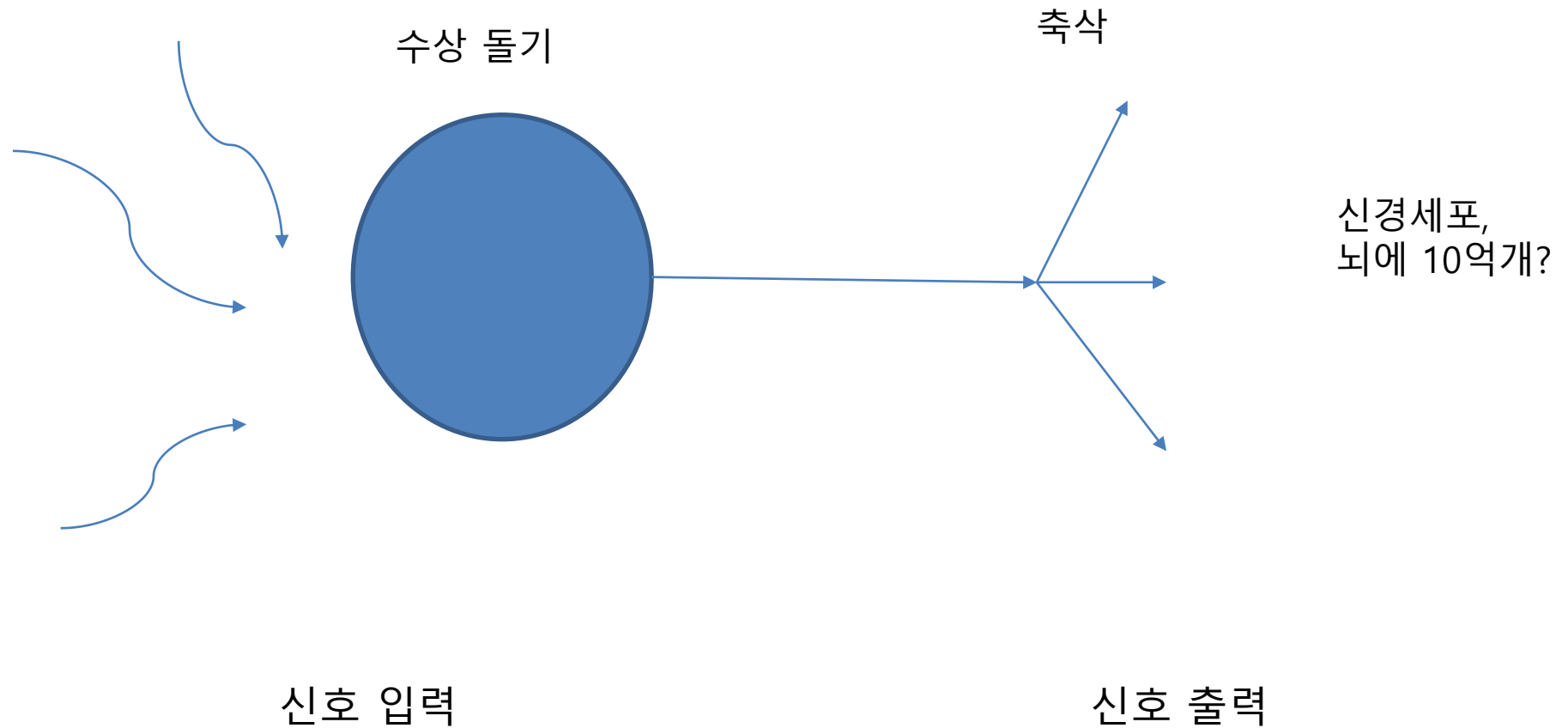
### 설명 가능한 AI(eXplainable AI)

eXplainable AI로 XAI라고도 하며,  
딥러닝과 같은 AI 모형의 예측 결과에 이르게 된 이유를 설명



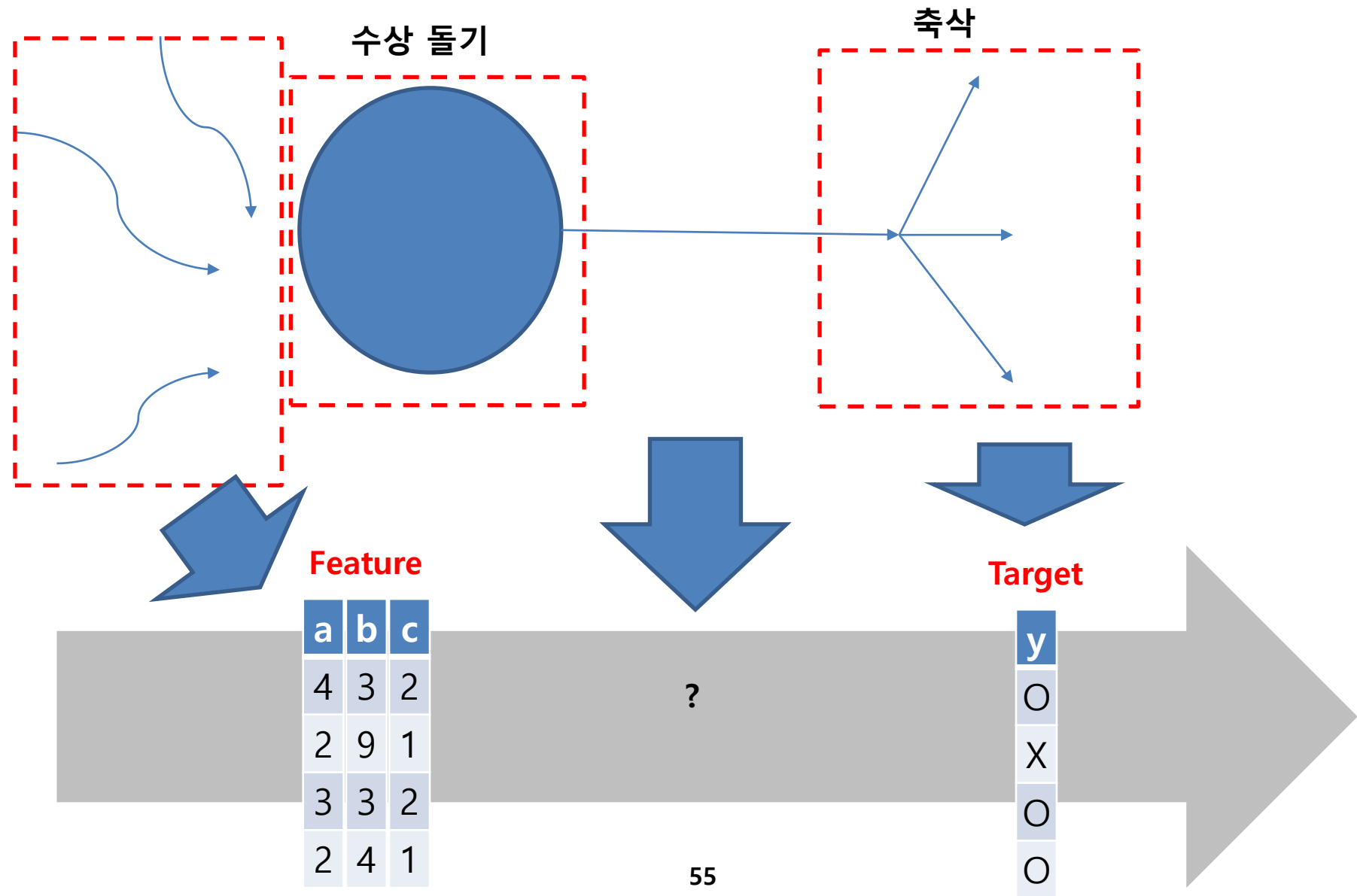
## 4. Artificial Neural Network로의 확장

뉴런?



# 4. Artificial Neural Network로의 확장

뉴런?



## 4. Artificial Neural Network로의 확장

---

### 퍼셉트론

by

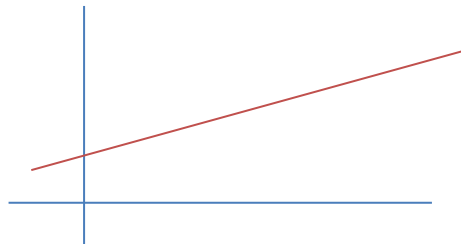
1957년

코넬 항공 연구소(Cornell Aeronautical Lab)

프랑크 로젠블라트 (Frank Rosenblatt)



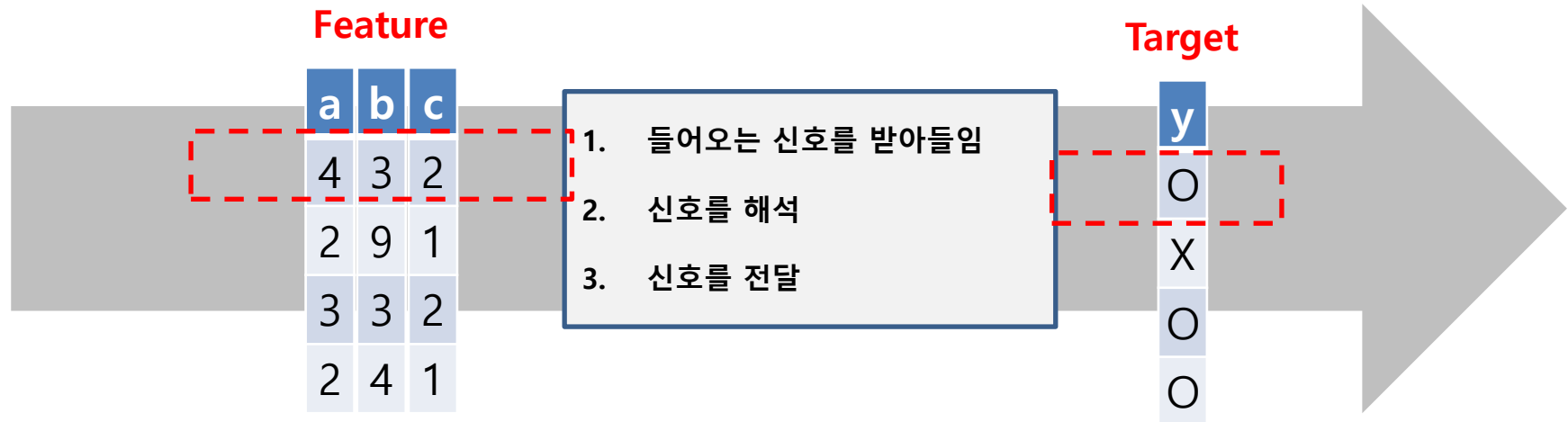
**간단한 형태의 선형분류기이자 인공신경망!**





## 4. Artificial Neural Network로의 확장

퍼셉트론: 신경세포와 같이 들어오는 신호를 바탕으로, Target을 계산



4, 3, 2



O  
범주

$$(4 \times ?) + (3 \times ?) + (2 \times ?)$$



만약 이 값이 어떤 수 보다 크면 O, 아니면 X

## 4. Artificial Neural Network로의 확장

### 퍼셉트론

$$(4 \times ?) + (3 \times ?) + (2 \times ?)$$



만약 이 값이 어떤 수 보다 크면 O, 아니면 X

- 선형 분류기: 직선식을 통한 분류
- 직선식으로 분류를 못하는 경우에는 한계
  - 예: XOR 형태의 데이터

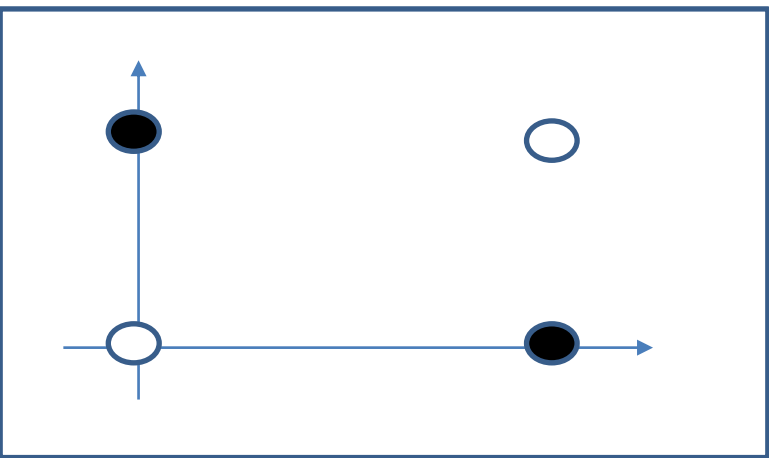
X1	X2	Y
0	0	0
0	1	1
1	0	1
1	1	0



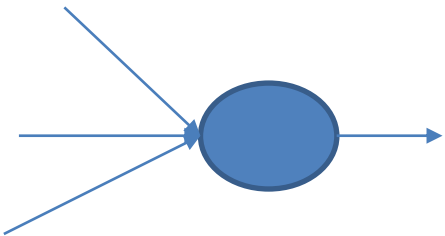
하나의 직선으로  
검은 원과 하얀원  
분류할 수 없음

# 4. Artificial Neural Network로의 확장

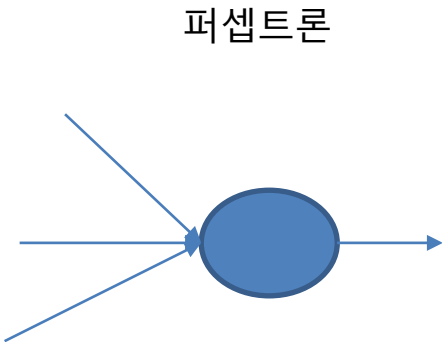
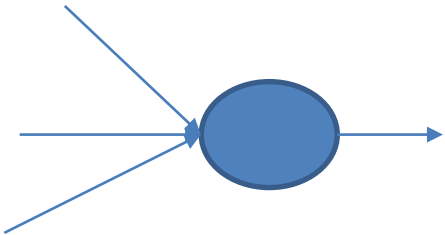
## 다층 퍼셉트론: 비선형 분류 가능!



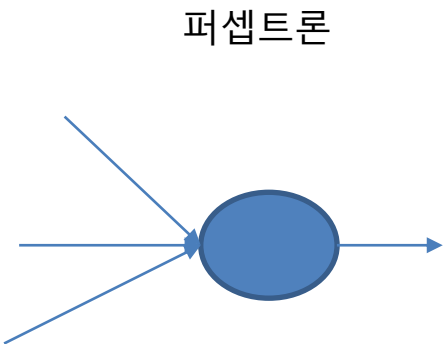
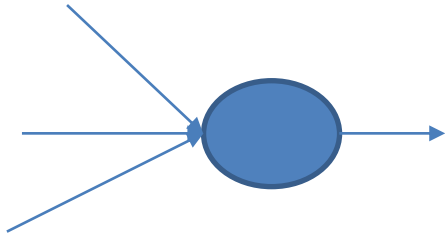
퍼셉트론



퍼셉트론



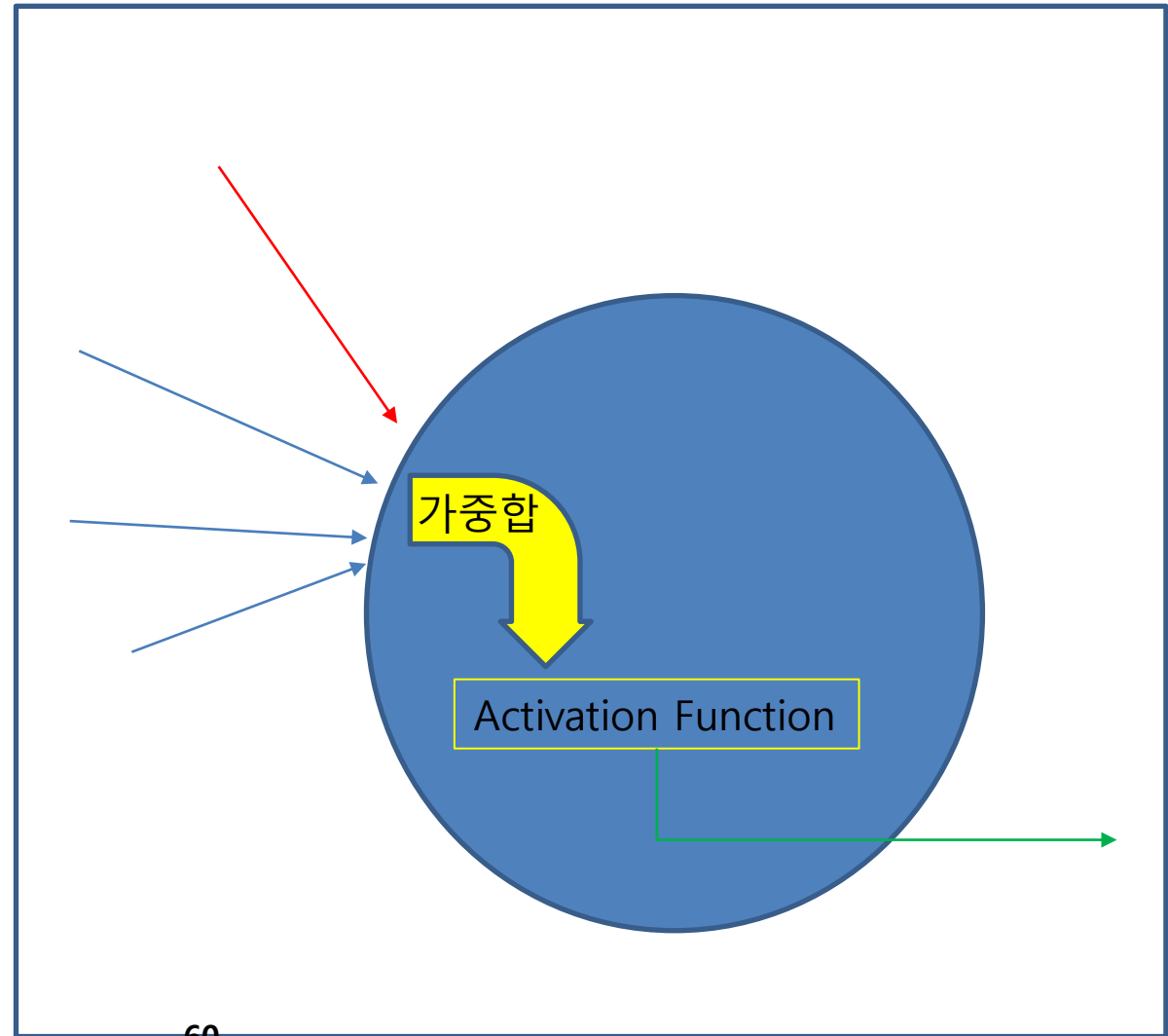
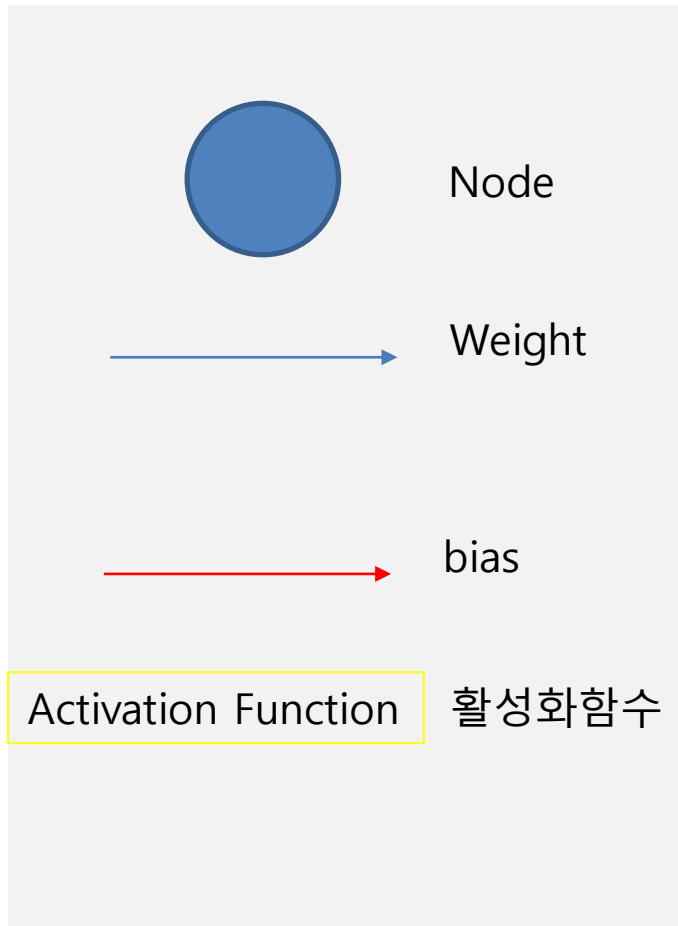
퍼셉트론



퍼셉트론

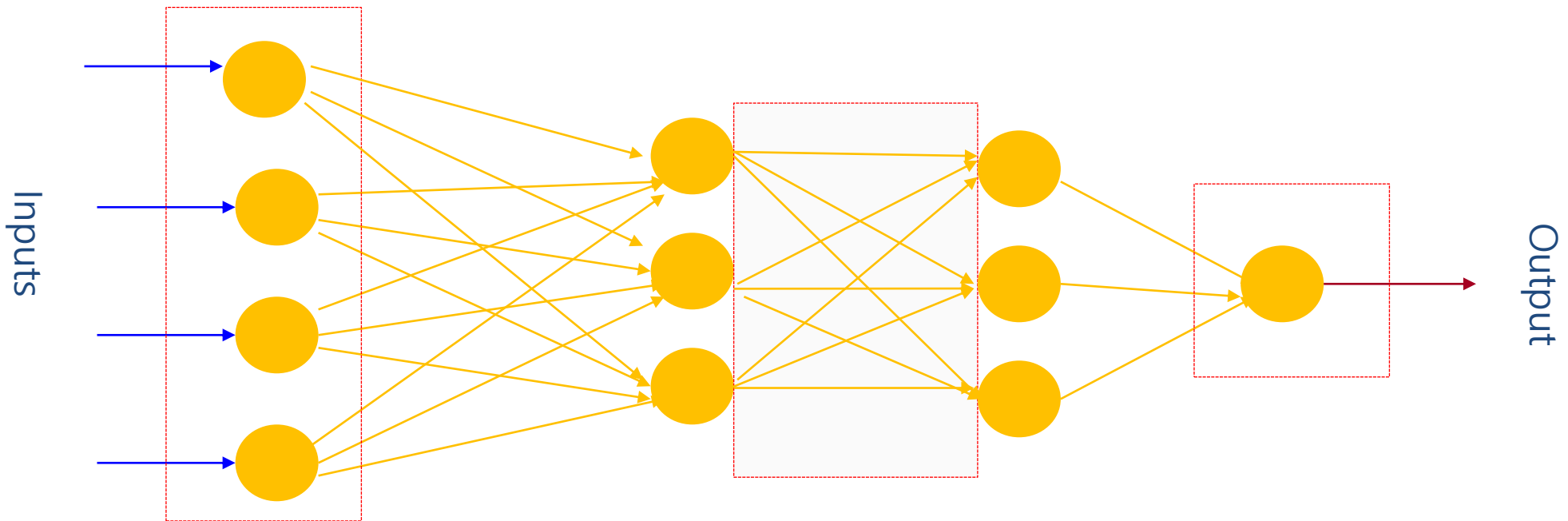
## 4. Artificial Neural Network로의 확장

### 인공신경망 구성 요소



## 4. Artificial Neural Network로의 확장

- **Artificial Neural Network(ANN, 인공신경망)**
  - 생물학의 신경망(동물의 중추신경계중 특히 뇌의 뉴런)을 모사한 학습 알고리즘
  - 뉴런을 모방한 노드들이 각각 Input Layer, Hidden Layer, Output Layer로 구분되며 데이터를 입력받아 변환하여 원하는 결과로 출력하는 네트워크를 구축
  - 예측 성능이 우수하다고 알려진 반면, 모델을 직관적으로 이해하기가 어려움



노드 간의 연결

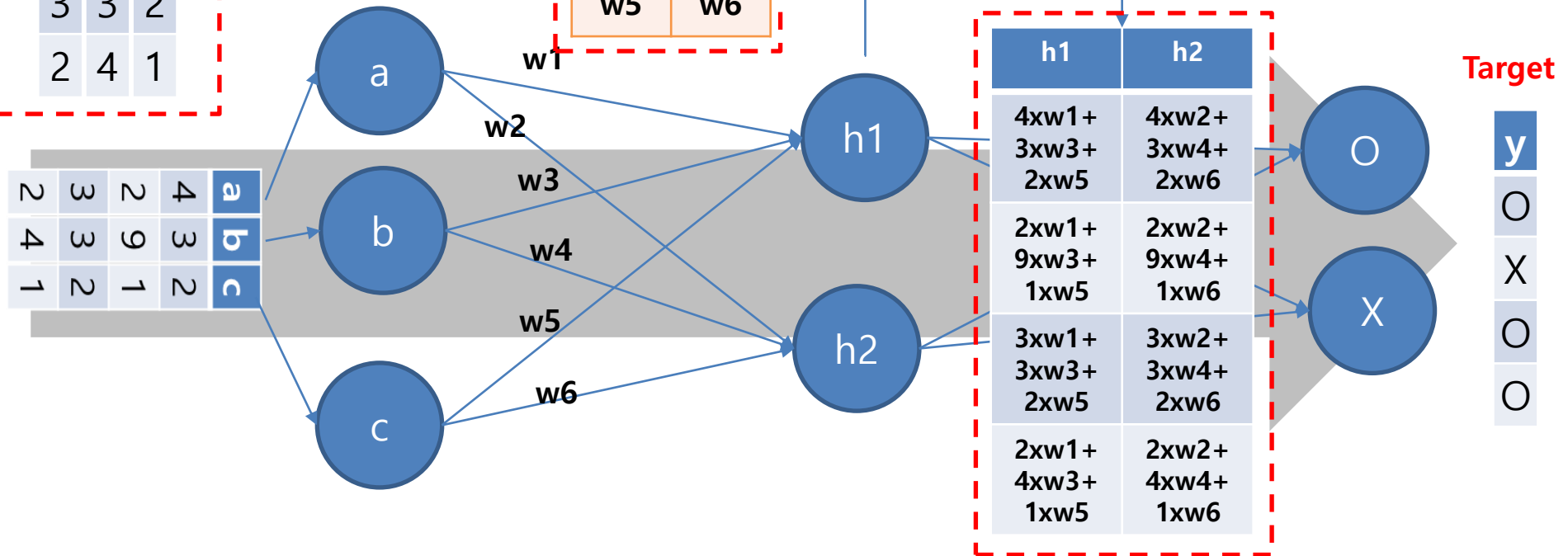
## 4. Artificial Neural Network로의 확장

### 인공신경망 in 행렬

Feature

a	b	c
4	3	2
2	9	1
3	3	2
2	4	1

w1	w2
w3	w4
w5	w6



## 4. Artificial Neural Network로의 확장

- 행렬의 곱

- 두 행렬의 곱은 각 행/열의 곱의 합으로 계산

$$\begin{matrix} \begin{pmatrix} 1 & 0 \\ 2 & 3 \end{pmatrix} & \times & \begin{pmatrix} 2 & 1 \\ 3 & 1 \end{pmatrix} & = & \begin{bmatrix} (1 \times 2) + (0 \times 3) & (1 \times 1) + (0 \times 1) \\ (2 \times 2) + (3 \times 3) & (2 \times 1) + (3 \times 1) \end{bmatrix} = \begin{pmatrix} \mathbf{2} & \mathbf{1} \\ \mathbf{13} & \mathbf{5} \end{pmatrix} \\ \mathbf{A} & & \mathbf{B} \end{matrix}$$

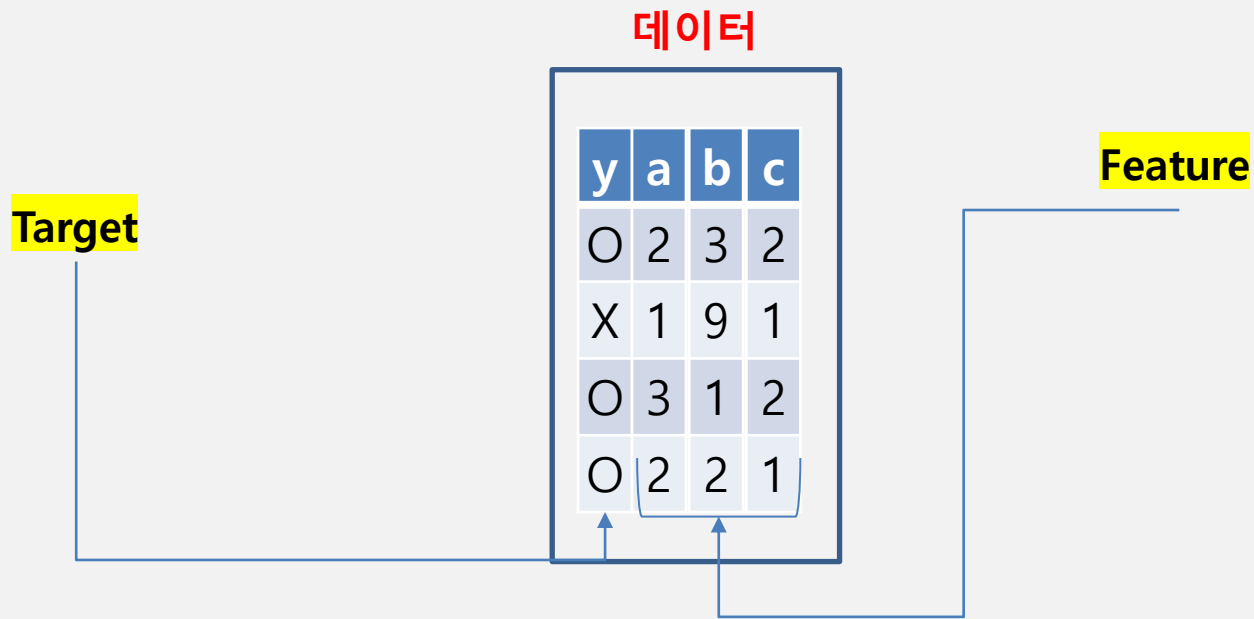


$\begin{pmatrix} 1 & 0 \\ 2 & 3 \end{pmatrix} \times \begin{pmatrix} 2 & 1 \\ 3 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 2 & 3 \end{pmatrix} \times \begin{pmatrix} 2 & 1 \\ 3 & 1 \end{pmatrix}$
$\begin{pmatrix} 1 & 0 \\ 2 & 3 \end{pmatrix} \times \begin{pmatrix} 2 & 1 \\ 3 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 2 & 3 \end{pmatrix} \times \begin{pmatrix} 2 & 1 \\ 3 & 1 \end{pmatrix}$

행렬곱의 두 행렬 중 앞 행렬의 열의 수가 뒤 행렬의 행의 수와 같아야 함

## 4. Artificial Neural Network로의 확장

### ANN Step by Step



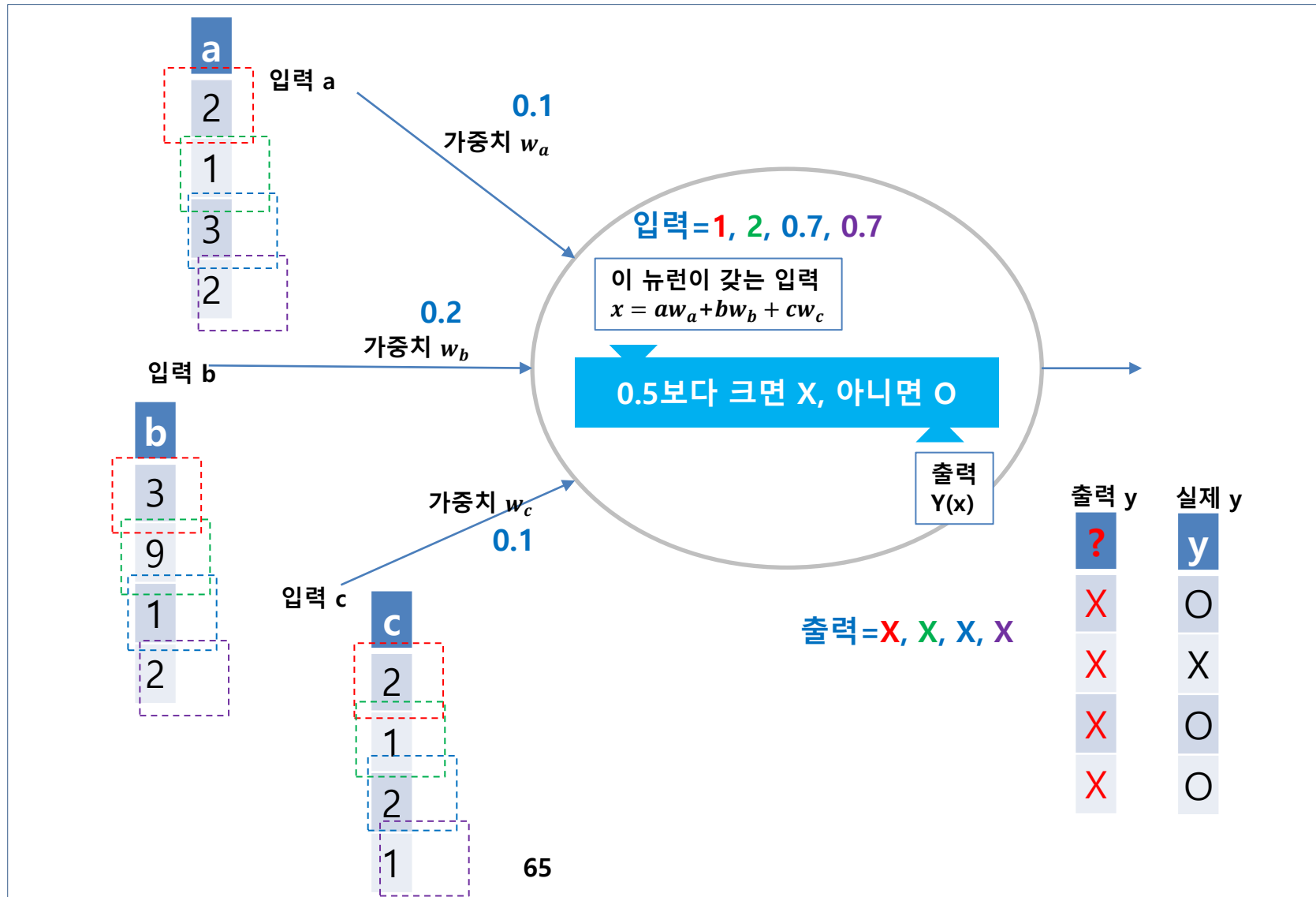
*Supervised Learning-Classification*



## 4. Artificial Neural Network로의 확장

### ANN Step by Step

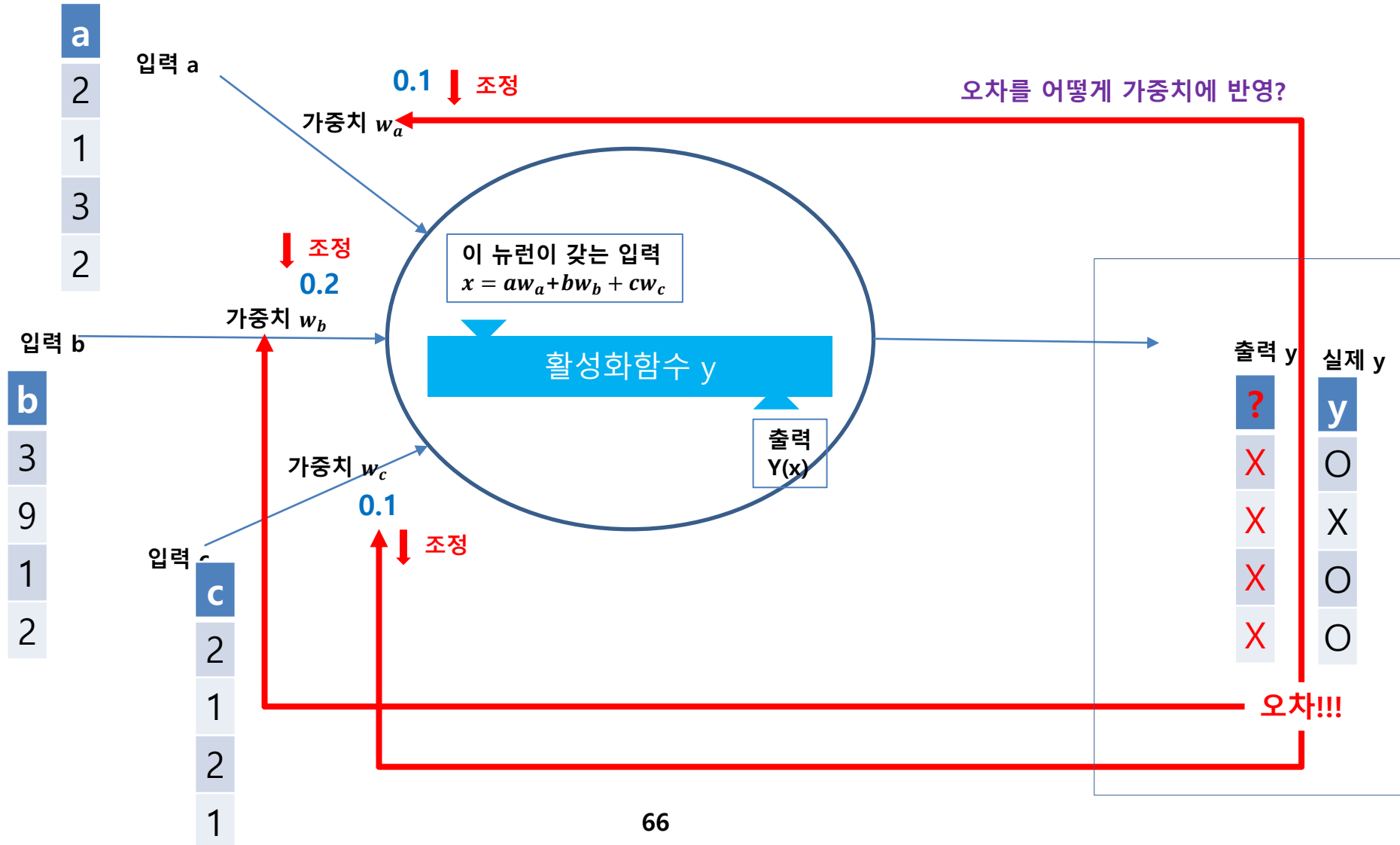
#### 인공신경망 1단계: 전파!



## 4. Artificial Neural Network로의 확장

### ANN Step by Step

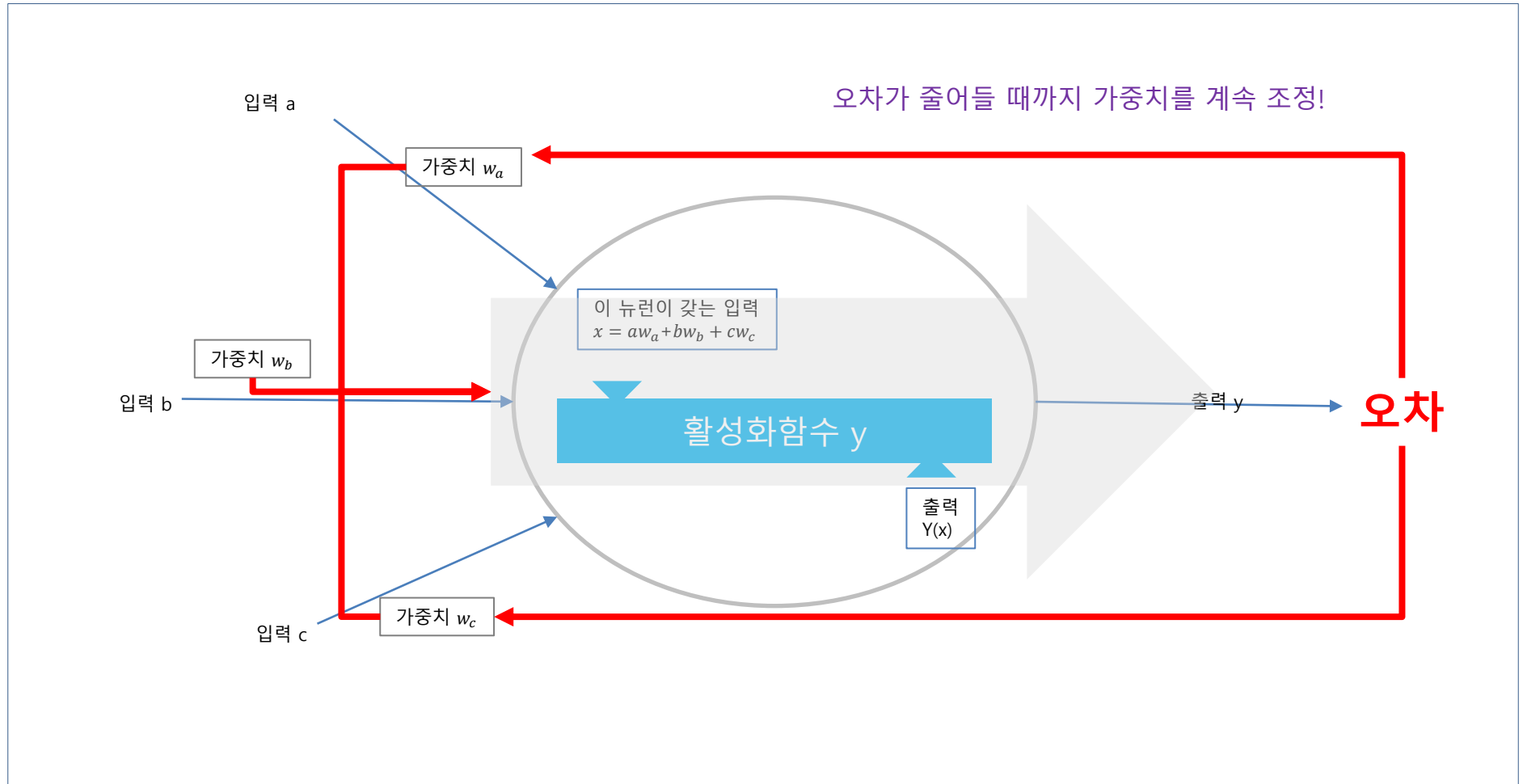
#### 인공신경망 2단계: 오차의 역전파!



## 4. Artificial Neural Network로의 확장

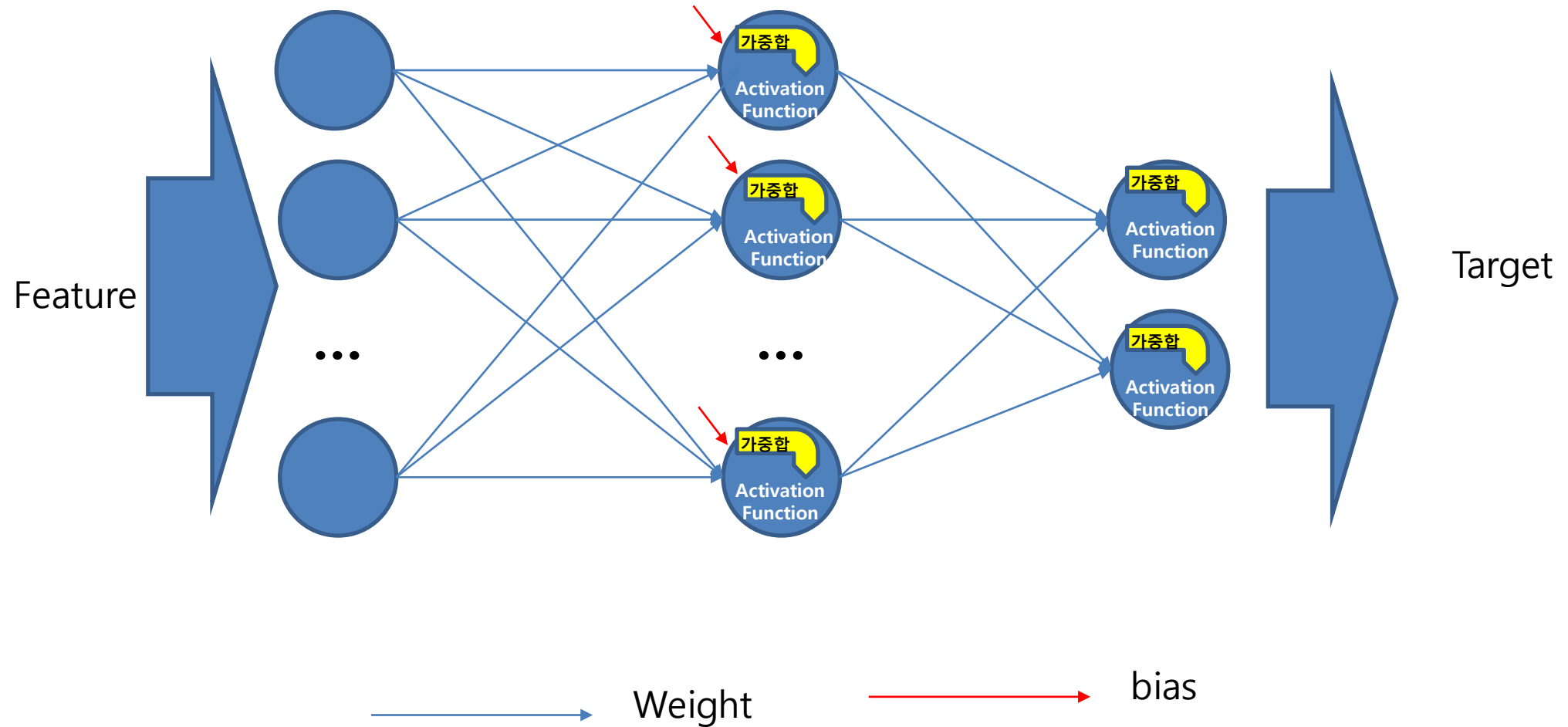
### ANN Step by Step

잘 할 때까지 반복!



## 4. Artificial Neural Network의 가중치 찾기

### 인공신경망과 가중치

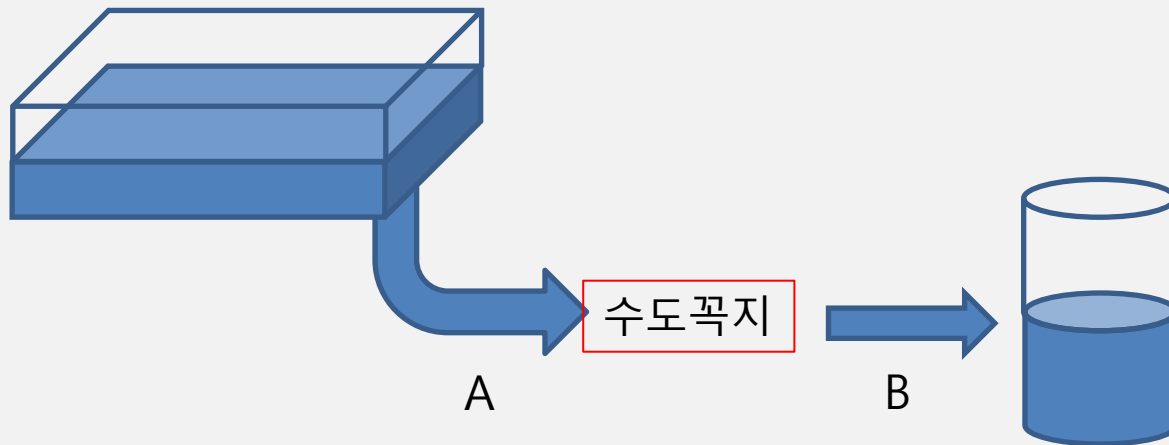


## 4. Artificial Neural Network의 가중치 찾기

### 가중치는?

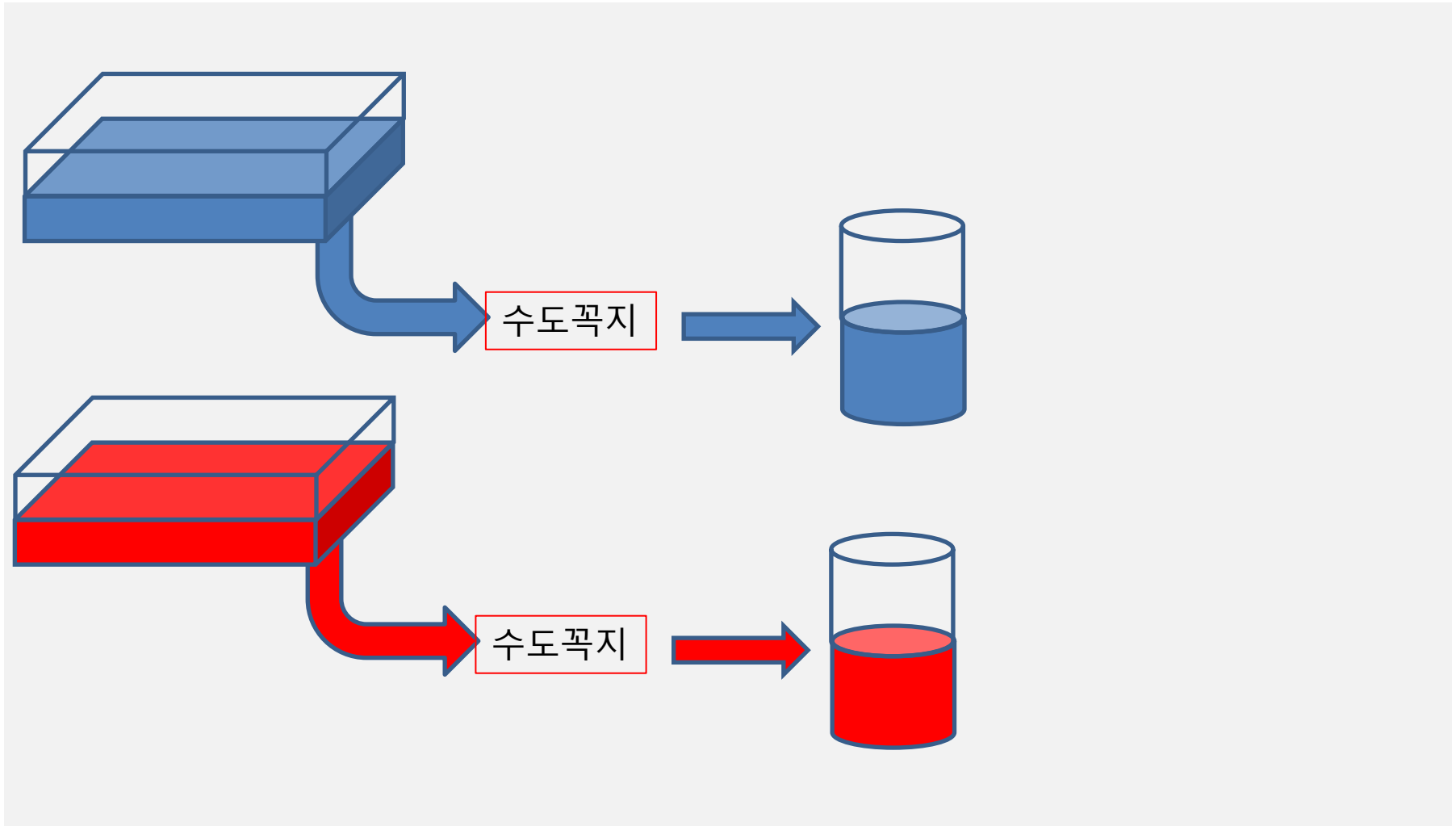
Weight, 가중치

$B = A \times \text{"수도꼭지를 통해 흘려보내는 정도"}$



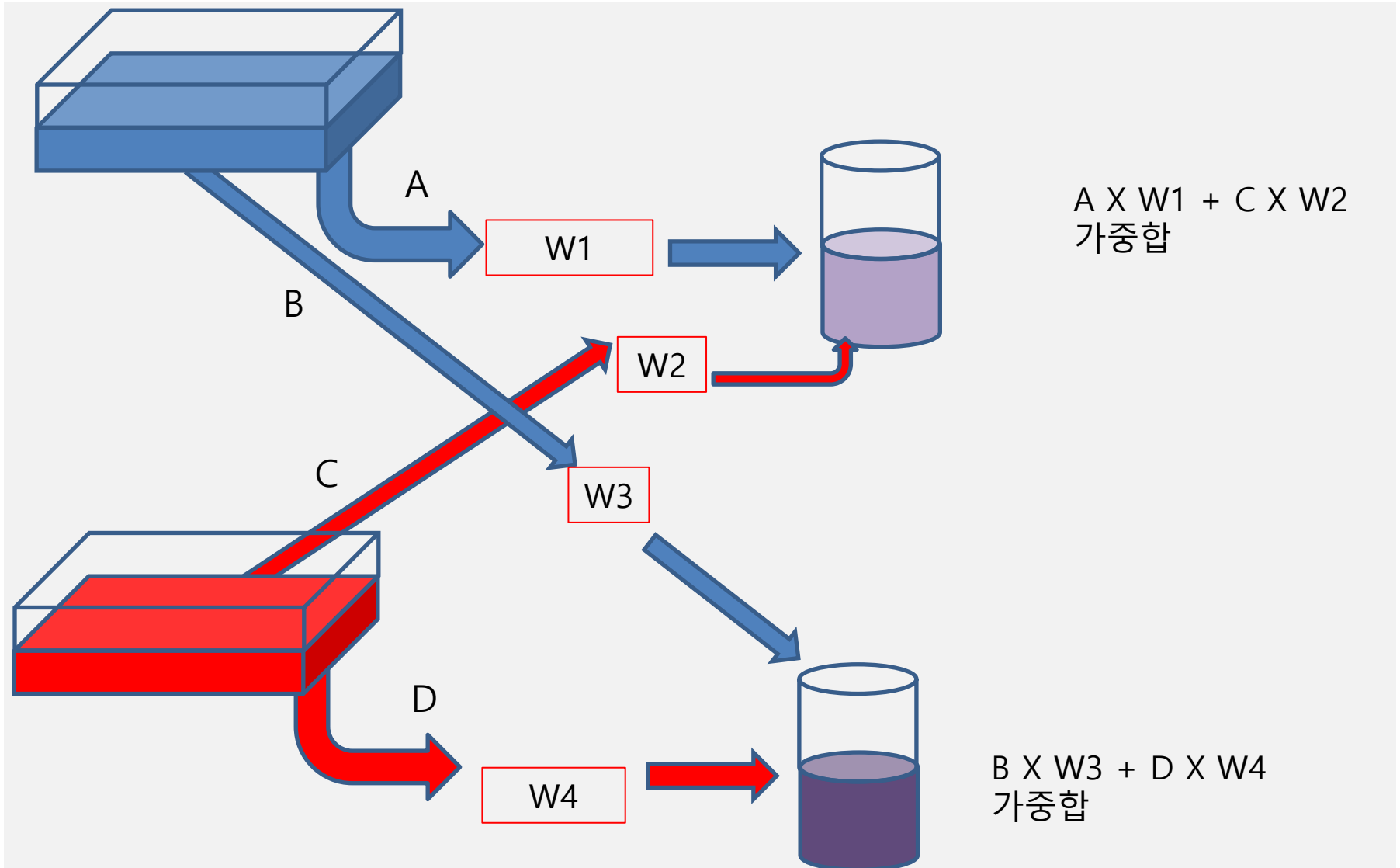
## 4. Artificial Neural Network의 가중치 찾기

가중치는?



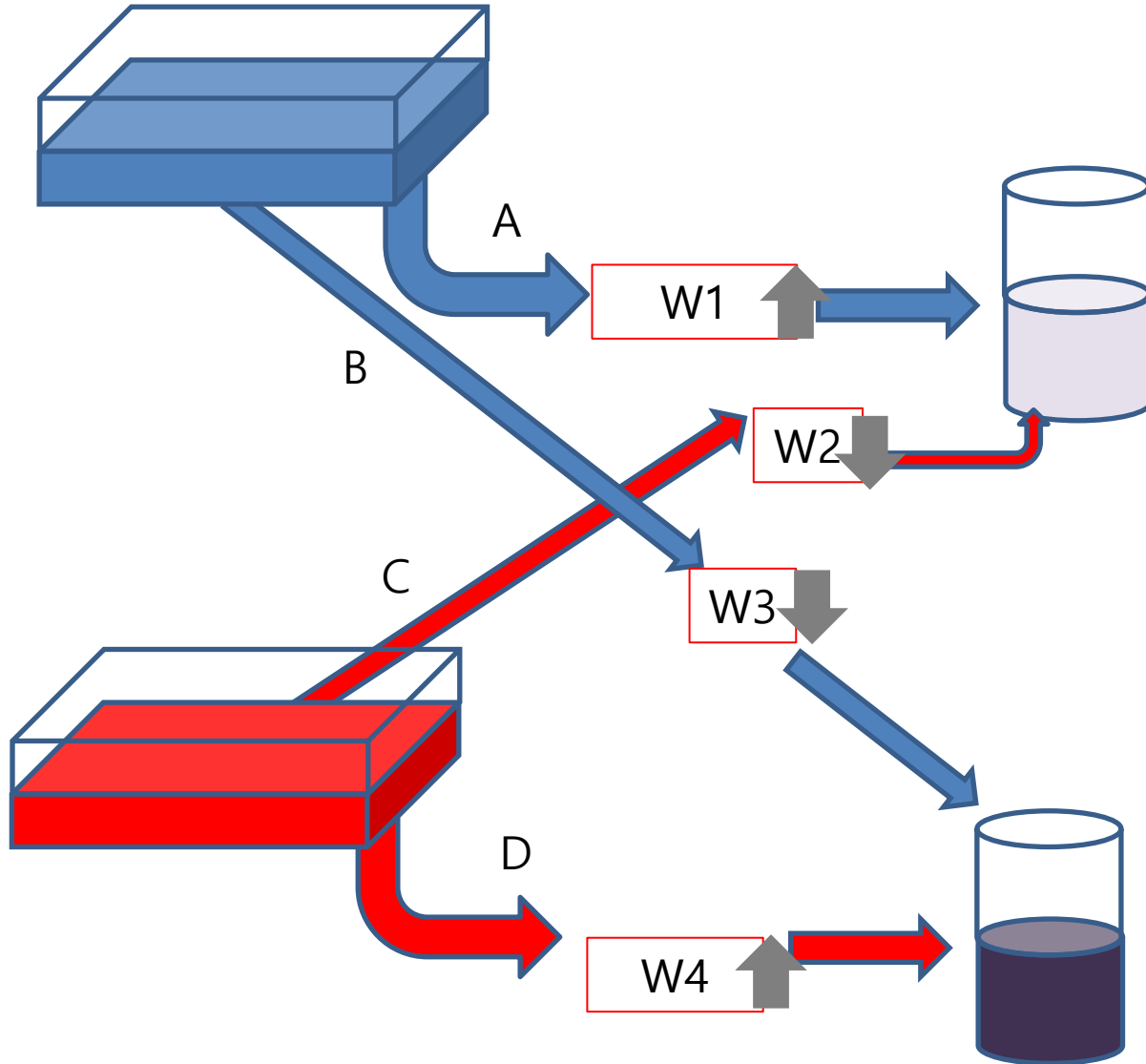
## 4. Artificial Neural Network의 가중치 찾기

### 가중치는?



## 4. Artificial Neural Network의 가중치 찾기

가중치 조정!

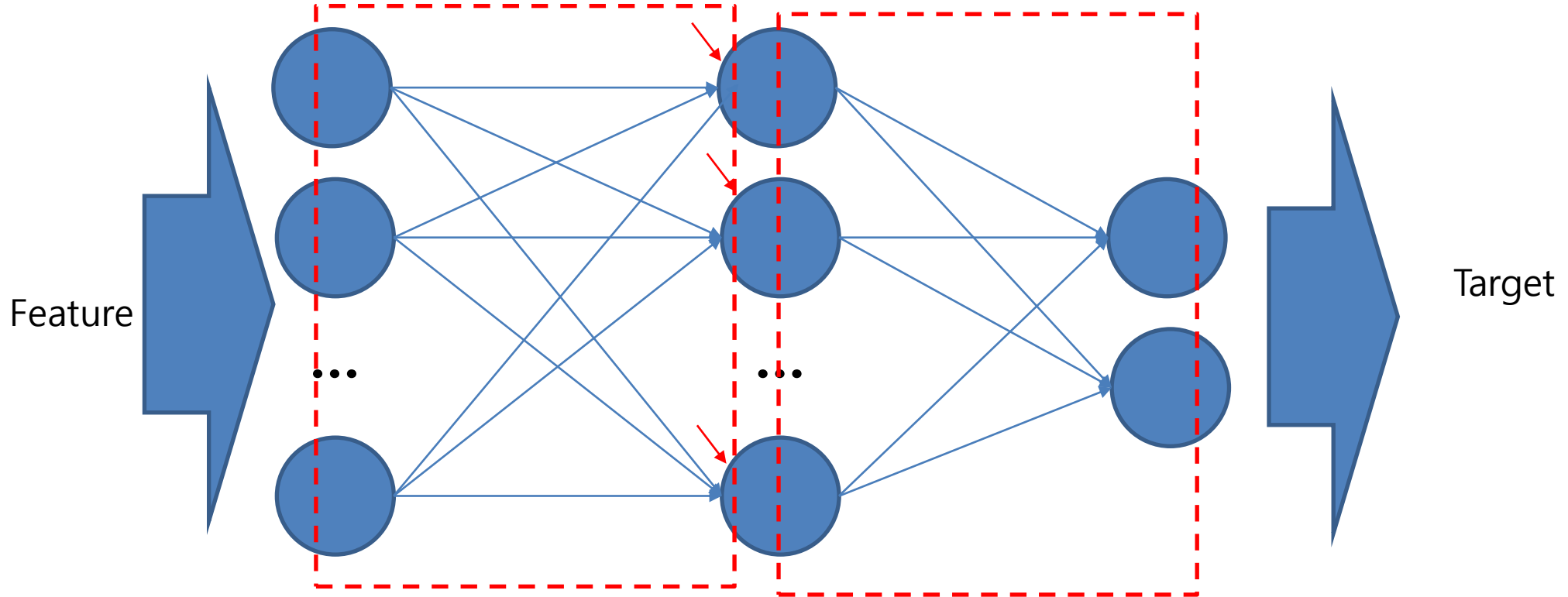


원하는 색상이 아닌 경우,  
"수도꼭지"를 조절!



## 4. Artificial Neural Network의 가중치 찾기

### 인공신경망과 가중치



원하는 결과를 얻도록 가중치를  $\pm$ 로 조정!

가중치와 관련하여 오늘 사용한 계산: 더하기, 곱하기, 크기 늘이기, 크기 줄이기

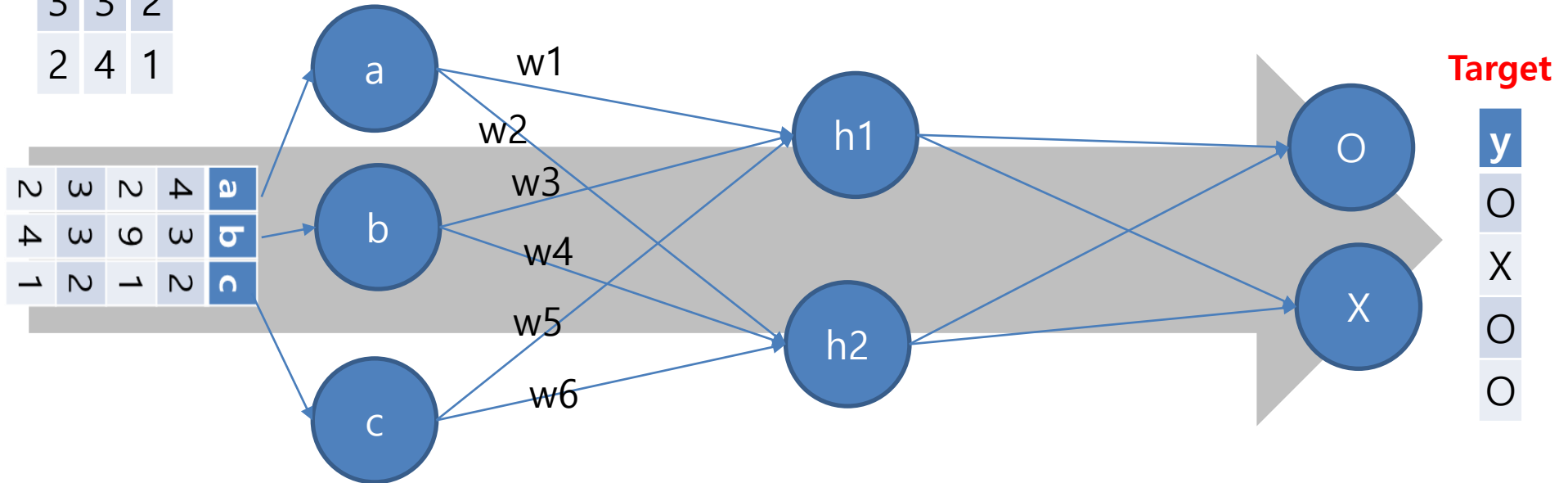
## 4. Artificial Neural Network의 가중치 찾기

### 인공신경망을 행렬로 표현

#### Feature

a	b	c
4	3	2
2	9	1
3	3	2
2	4	1

1. 입력데이터
2. 가중합
3. 은닉층 표현



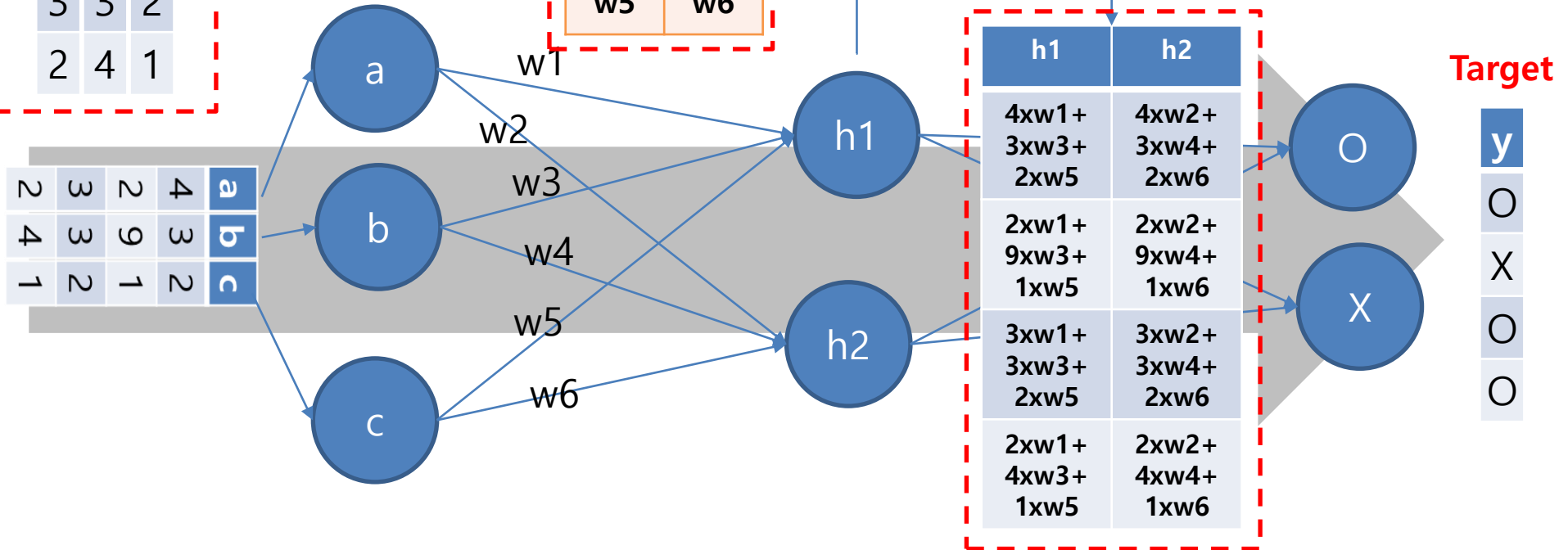
## 4. Artificial Neural Network의 가중치 찾기

인공신경망을 행렬로 표현하기!

**Feature**

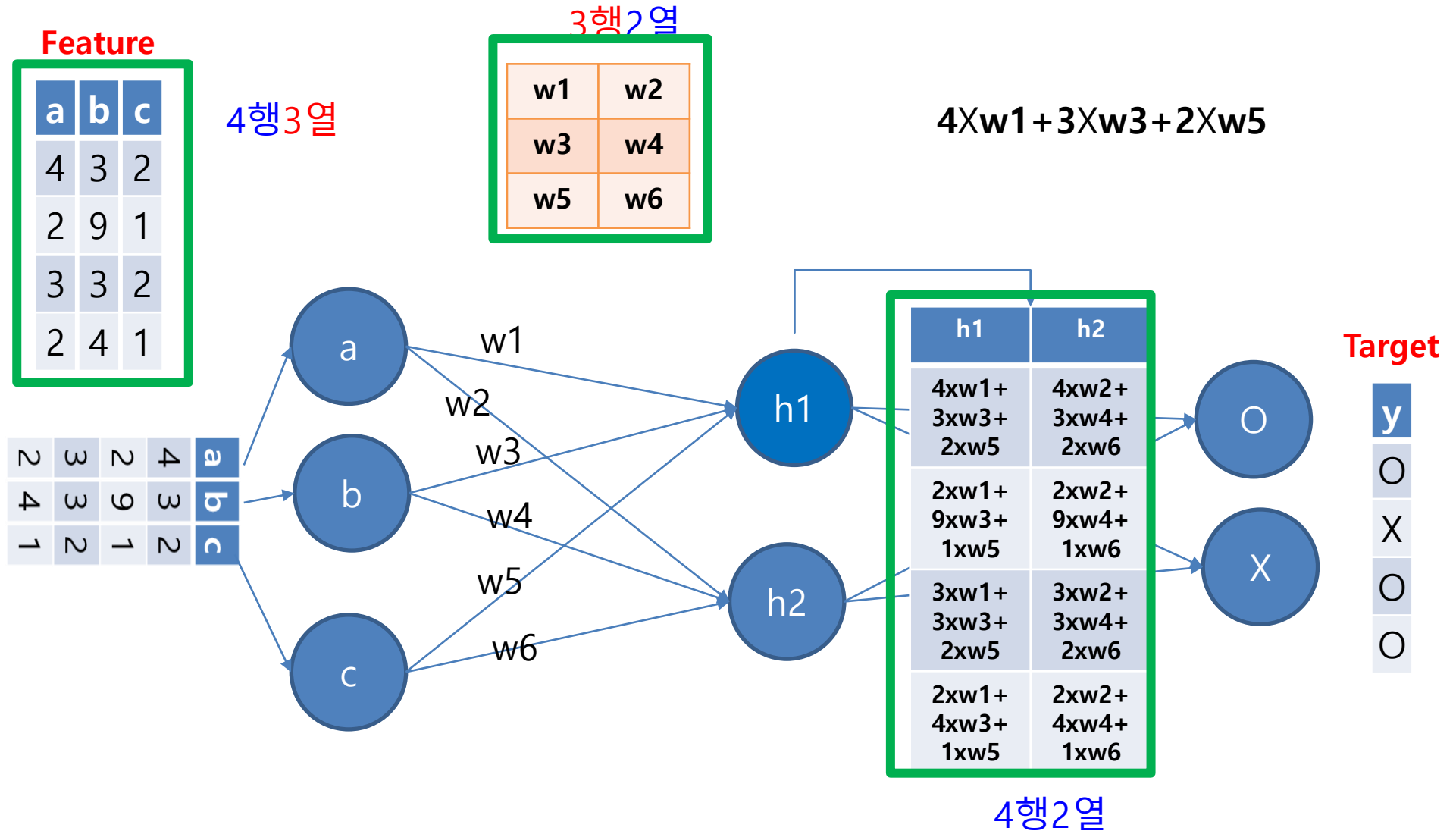
a	b	c
4	3	2
2	9	1
3	3	2
2	4	1

w1	w2
w3	w4
w5	w6



# 4. Artificial Neural Network의 가중치 찾기

## 인공신경망의 가중합



## 4. Artificial Neural Network의 가중치 찾기

### 인공신경망의 가중합

Feature

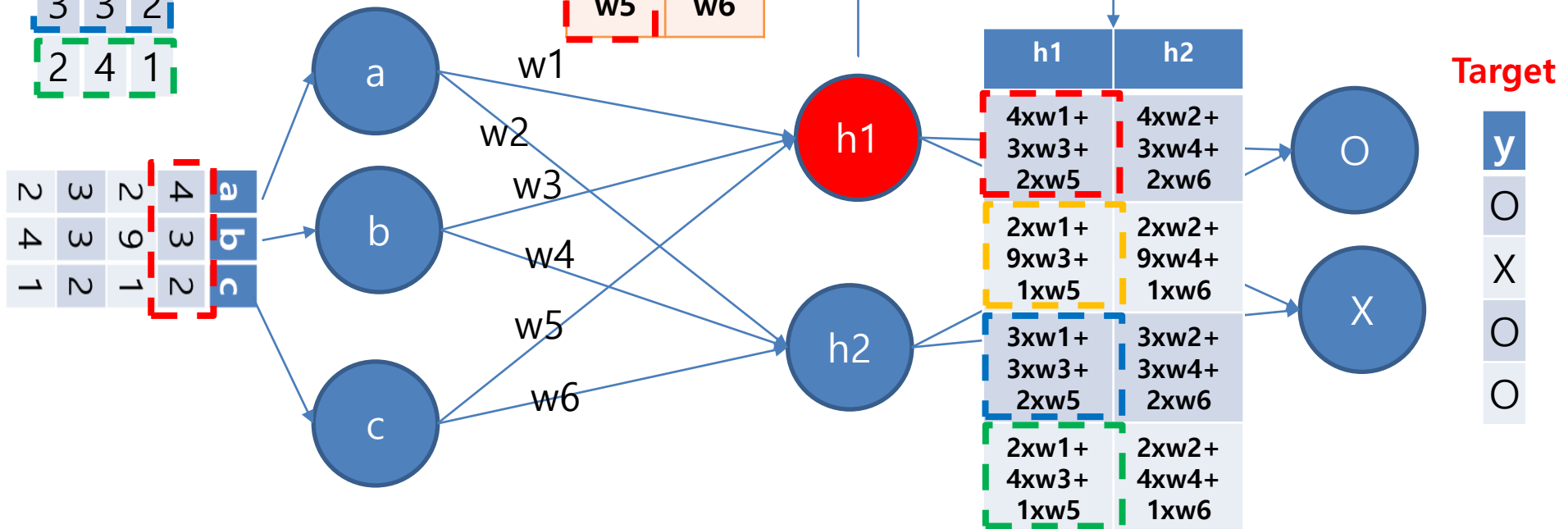
a	b	c
4	3	2
2	9	1
3	3	2
2	4	1

a	b	c
4	3	2
2	9	1
3	3	2
2	4	1

인공신경망 가중합 다시보기: 인공신경망의 가중합이란  
행렬곱 결과의 한 원소에 대한 계산과 동일

w1	w2
w3	w4
w5	w6

$$4 \times w1 + 3 \times w3 + 2 \times w5$$



## 4. Artificial Neural Network의 가중치 찾기

### 인공신경망의 행렬곱

인공신경망의 Feature와 가중치들은 모두 행렬 곱을 통해 계산됨

4행3열

a	b	c
4	3	2
2	9	1
3	3	2
2	4	1

X

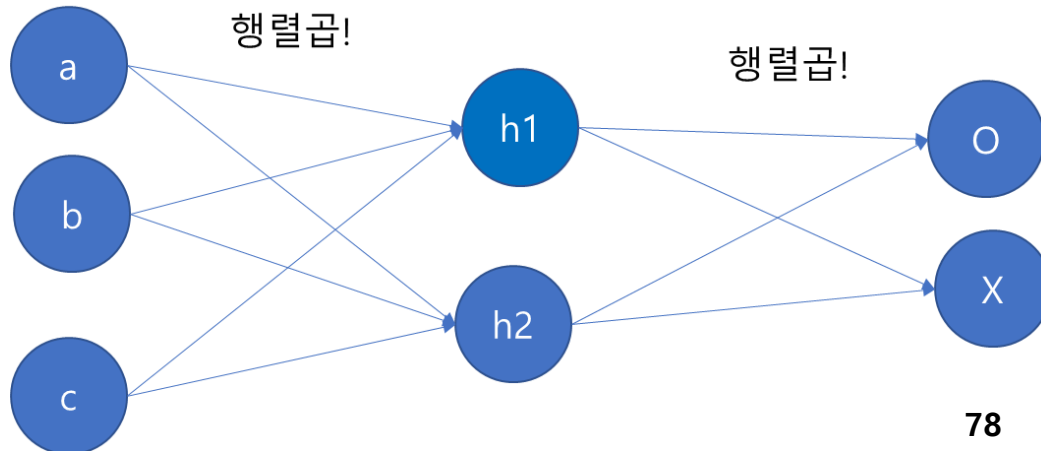
3행2열

w1	w2
w3	w4
w5	w6

=

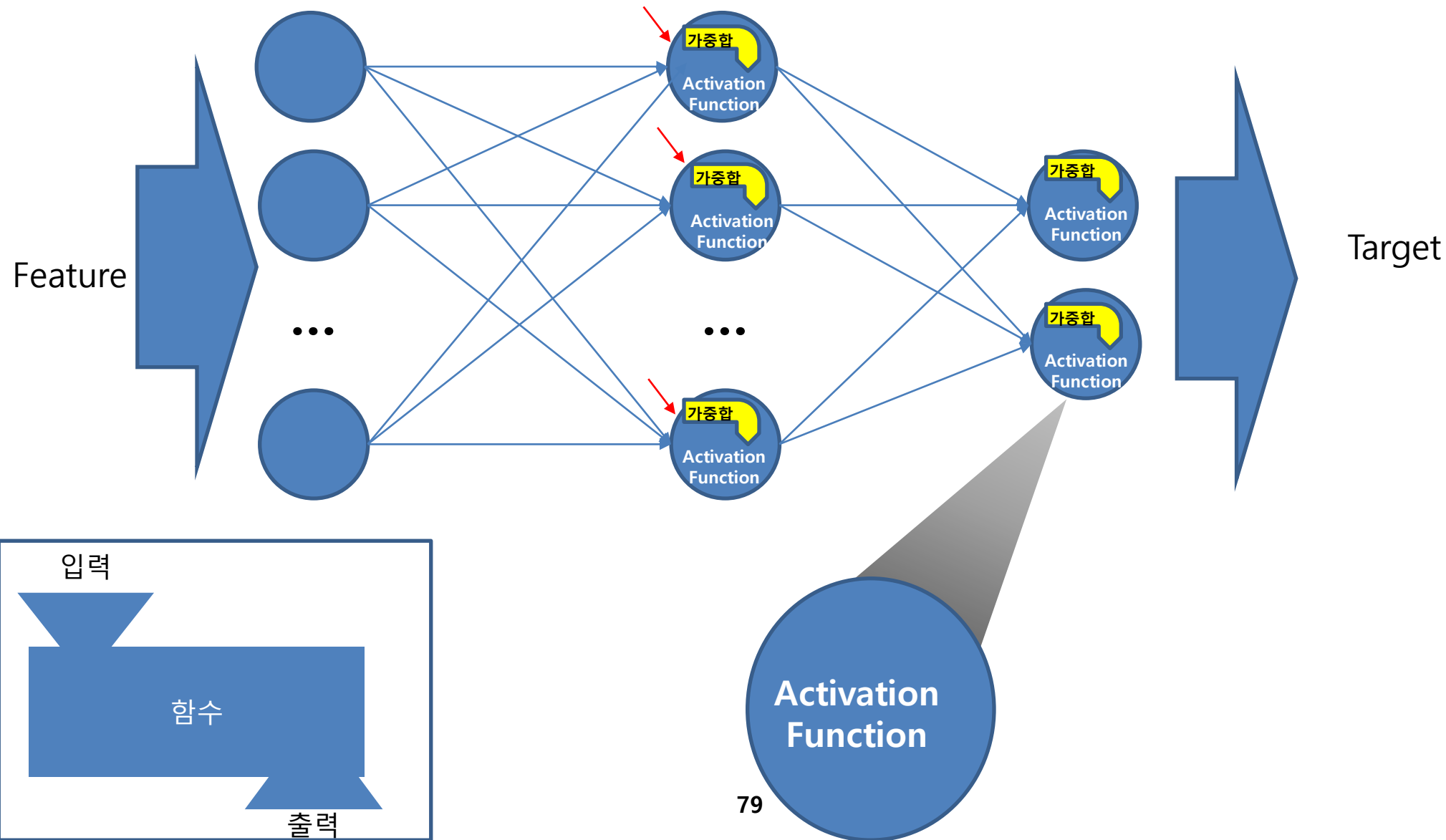
4행2열

h1	h2
4xw1+ 3xw3+ 2xw5	4xw2+ 3xw4+ 2xw6
2xw1+ 9xw3+ 1xw5	2xw2+ 9xw4+ 1xw6
3xw1+ 3xw3+ 2xw5	3xw2+ 3xw4+ 2xw6
2xw1+ 4xw3+ 1xw5	2xw2+ 4xw4+ 1xw6

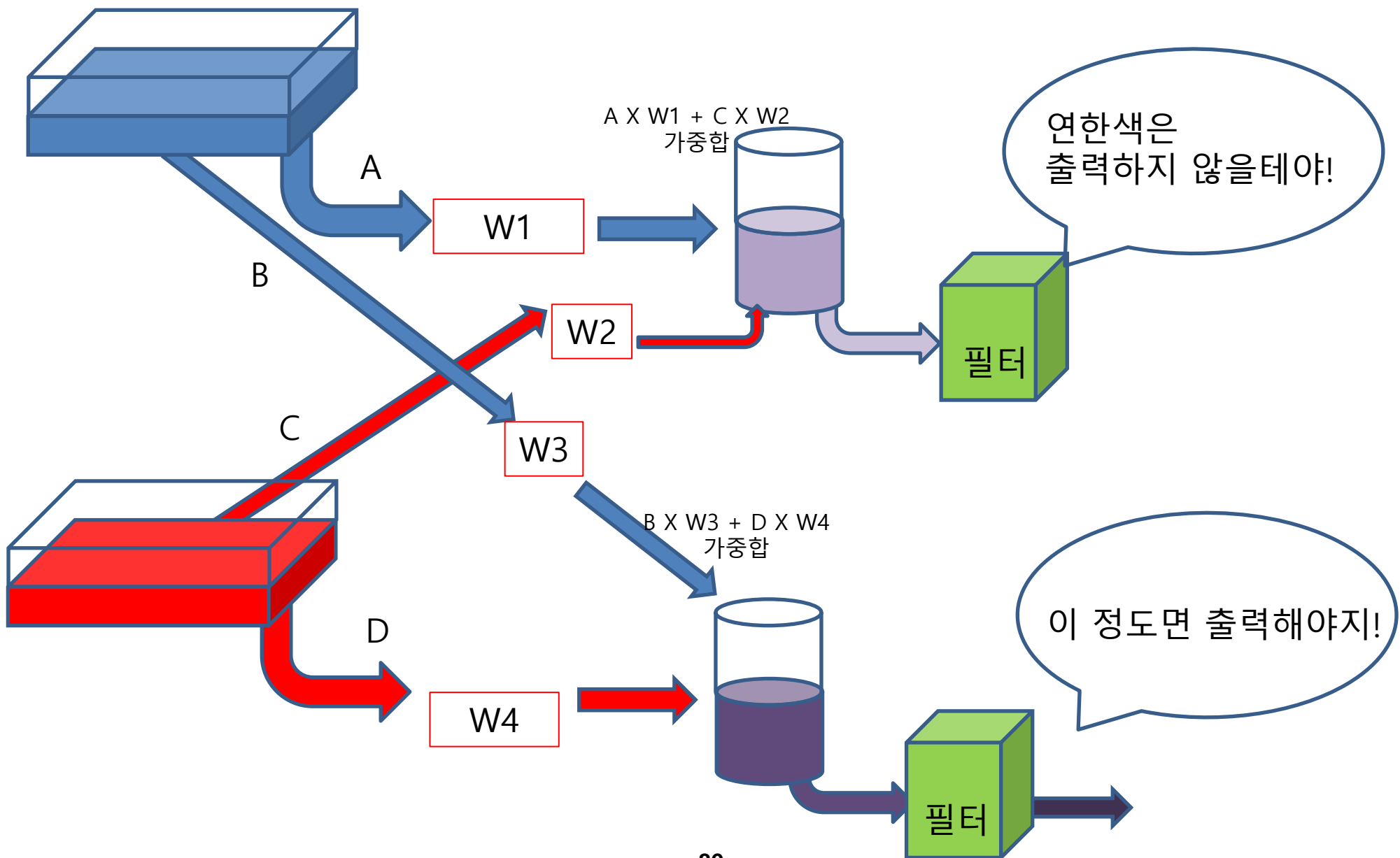


## 4. Artificial Neural Network의 가중치 찾기

### 활성화 함수



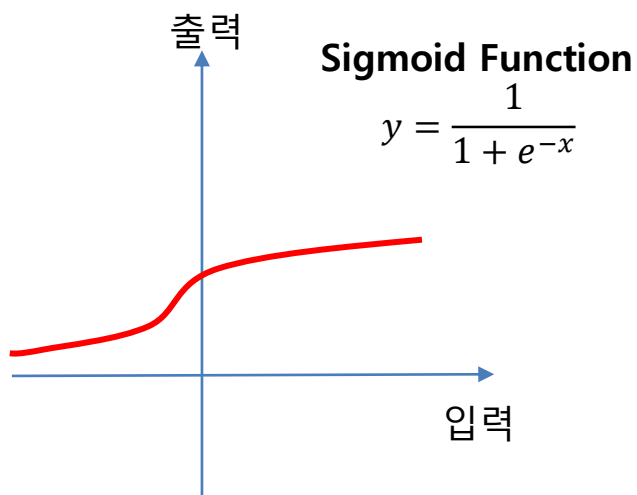
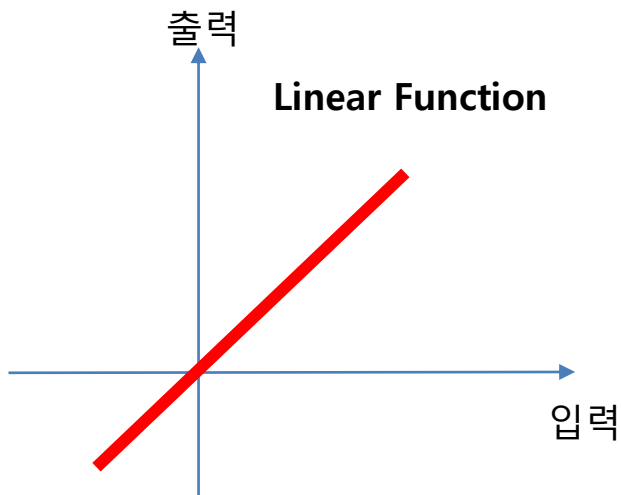
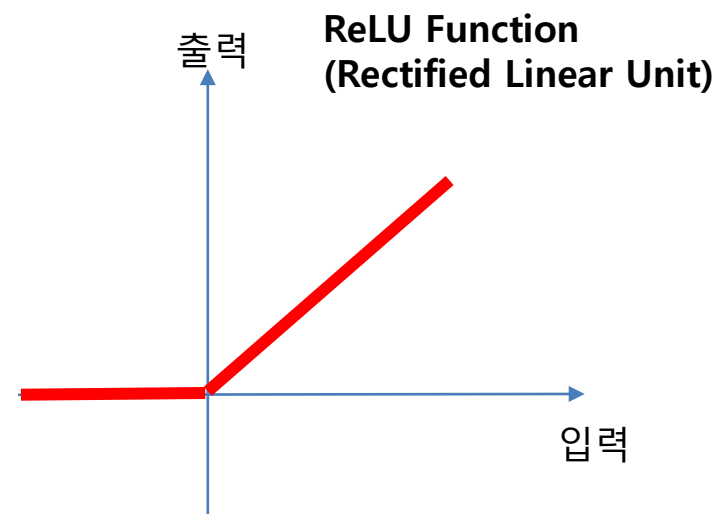
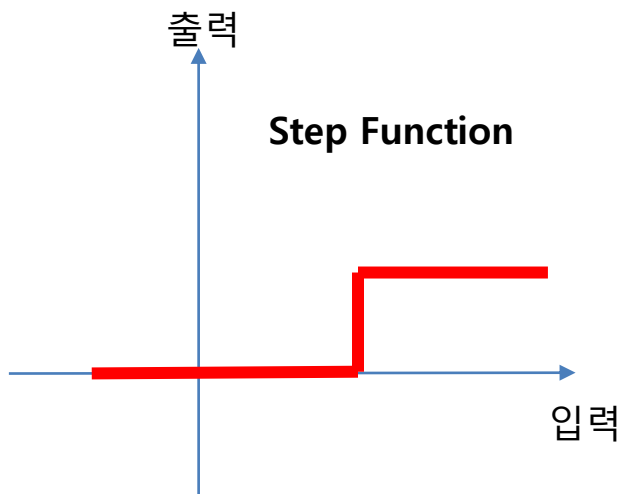
## 4. Artificial Neural Network의 가중치 찾기





## 4. Artificial Neural Network의 가중치 찾기

### 다양한 활성화 함수



## 4. Artificial Neural Network의 가중치 찾기

---

### Feedforward Neural Network

- 노드 간의 연결에서 순환이나 루프가 없는 기본적인 인공신경망



- Feedforward = 순방향



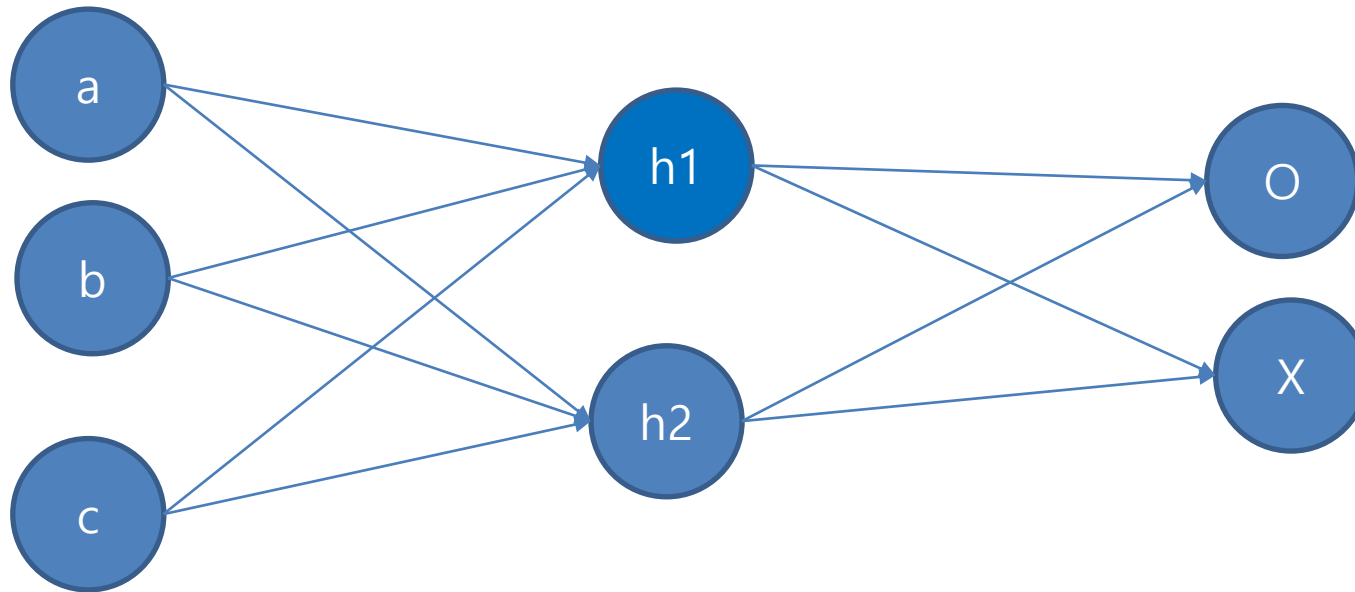
- Forward Propagation? Back Propagation?

## 4. Artificial Neural Network의 가중치 찾기

---

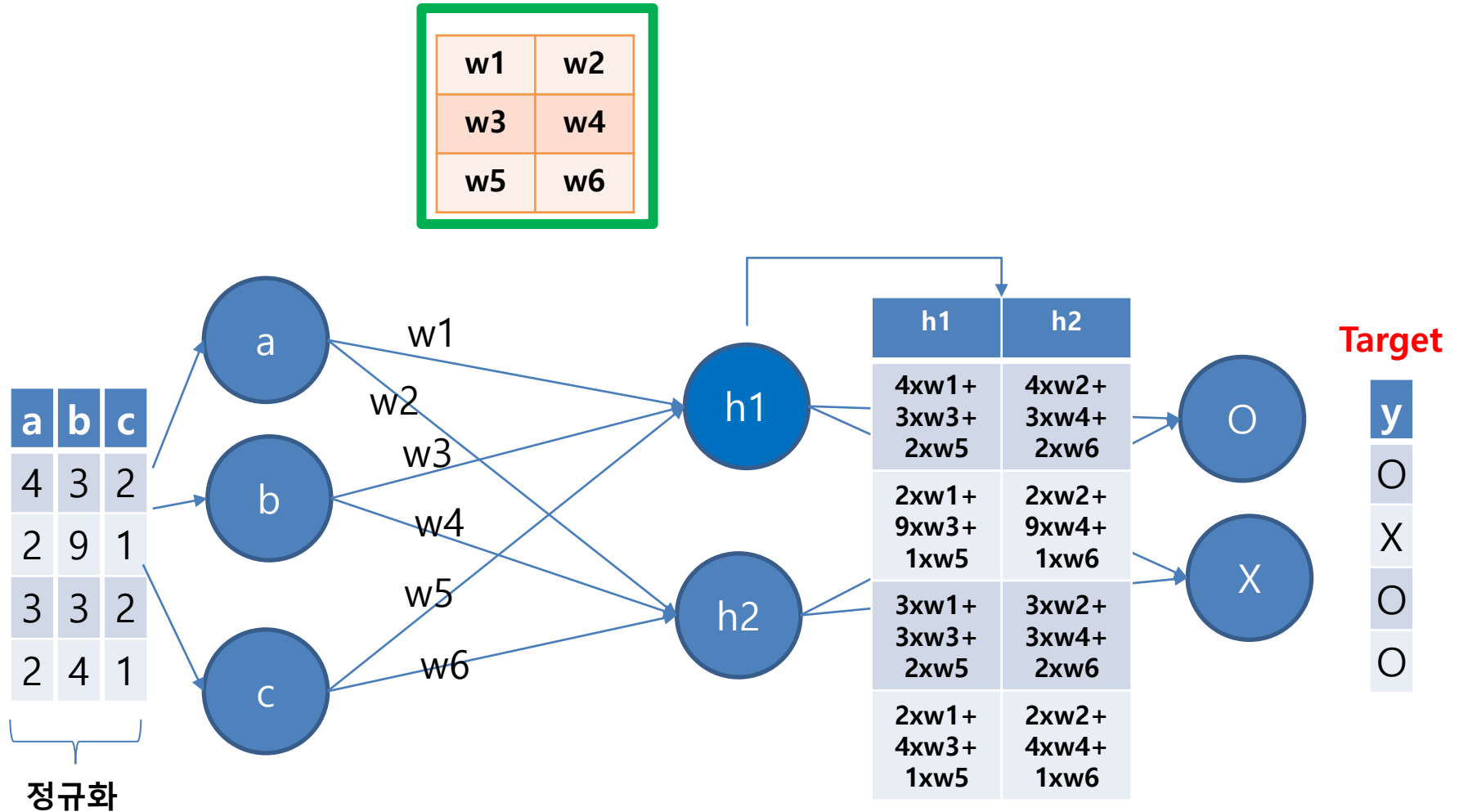
Forward Propagation = Matrix Multiplication

가중치!



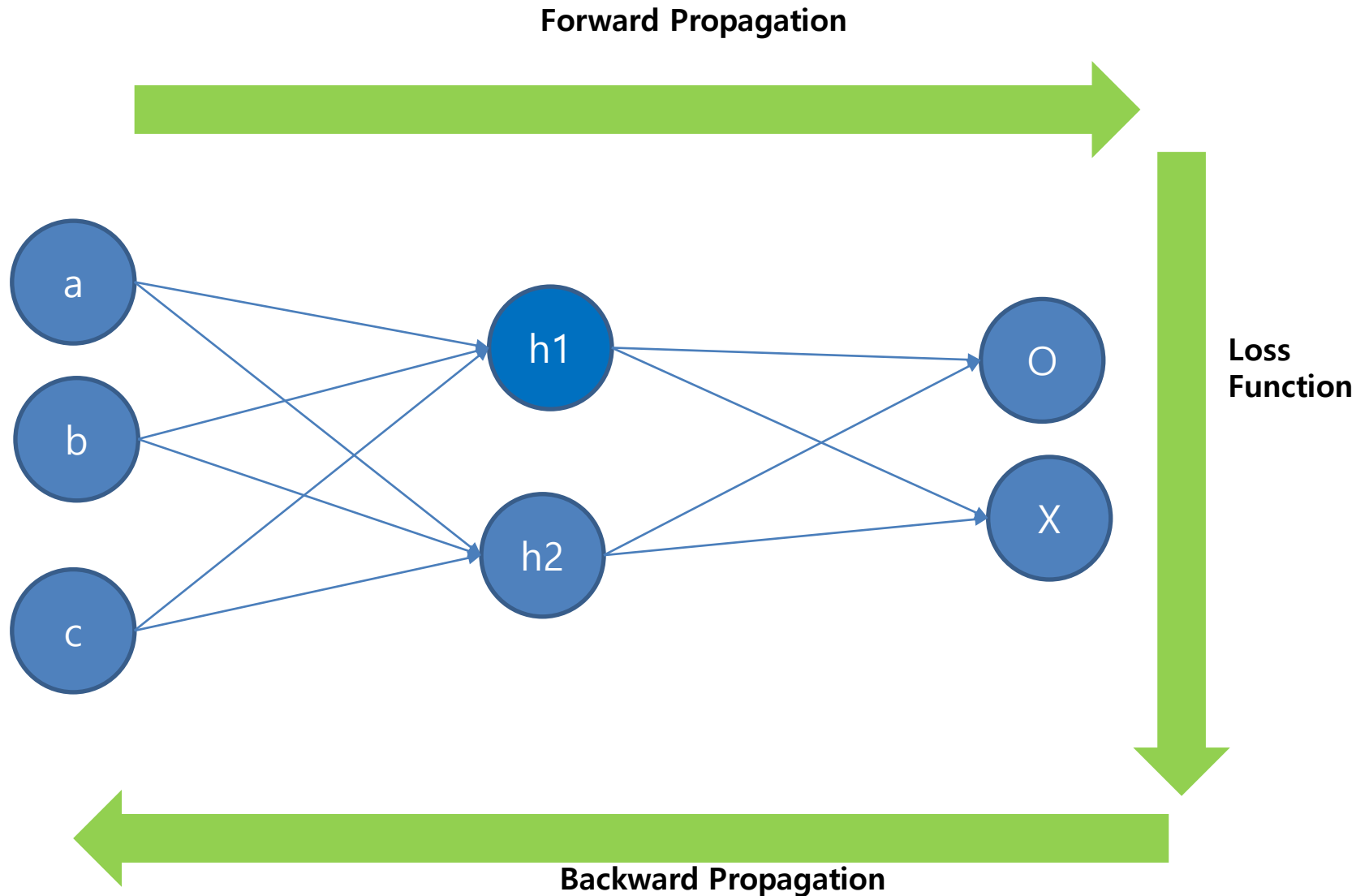
## 4. Artificial Neural Network의 가중치 찾기

가중치: 처음엔 Random (0~1)



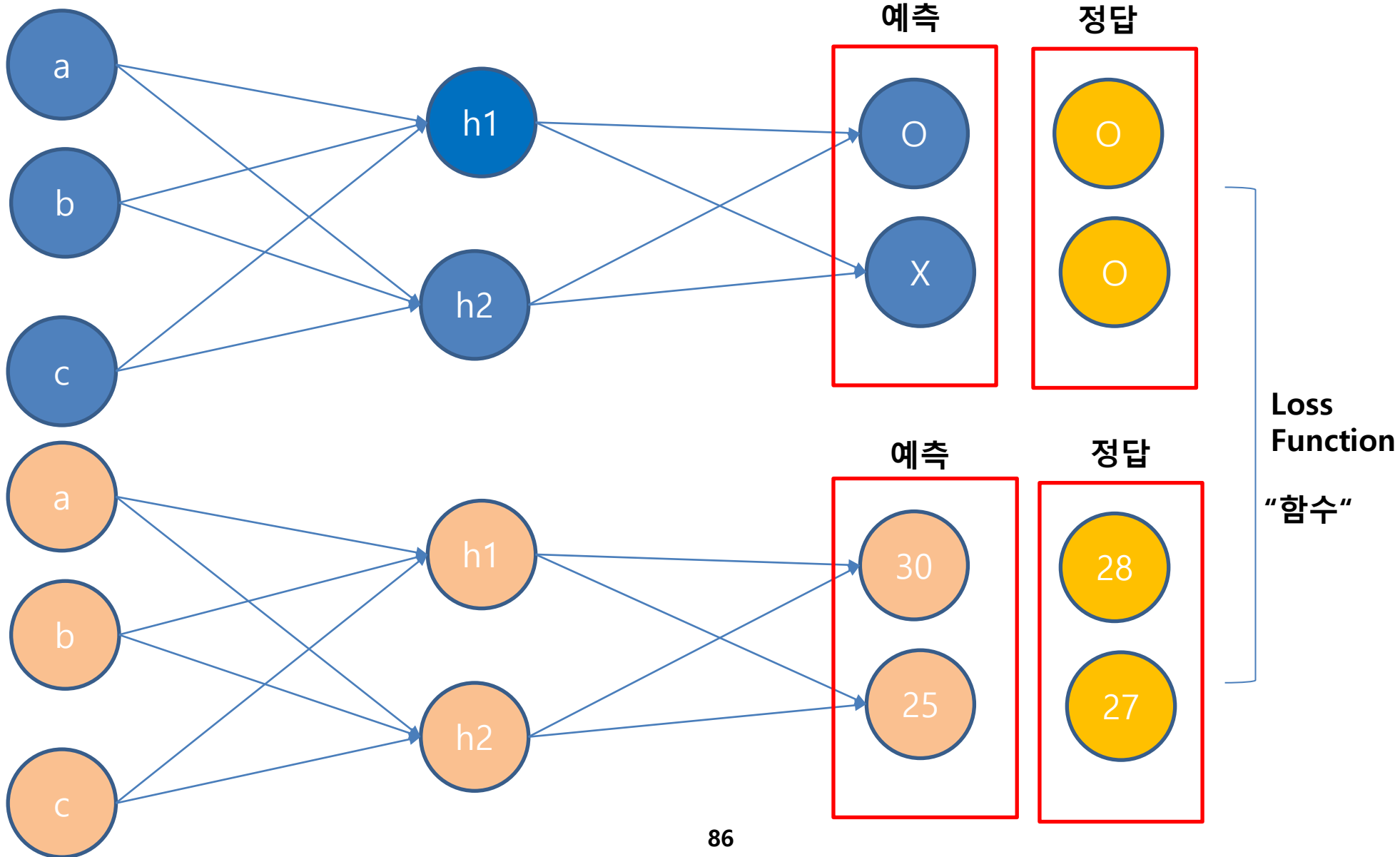
## 4. Artificial Neural Network의 가중치 찾기

Epoch(에포크): Forward Propagation + Back Propagation



## 4. Artificial Neural Network의 가중치 찾기

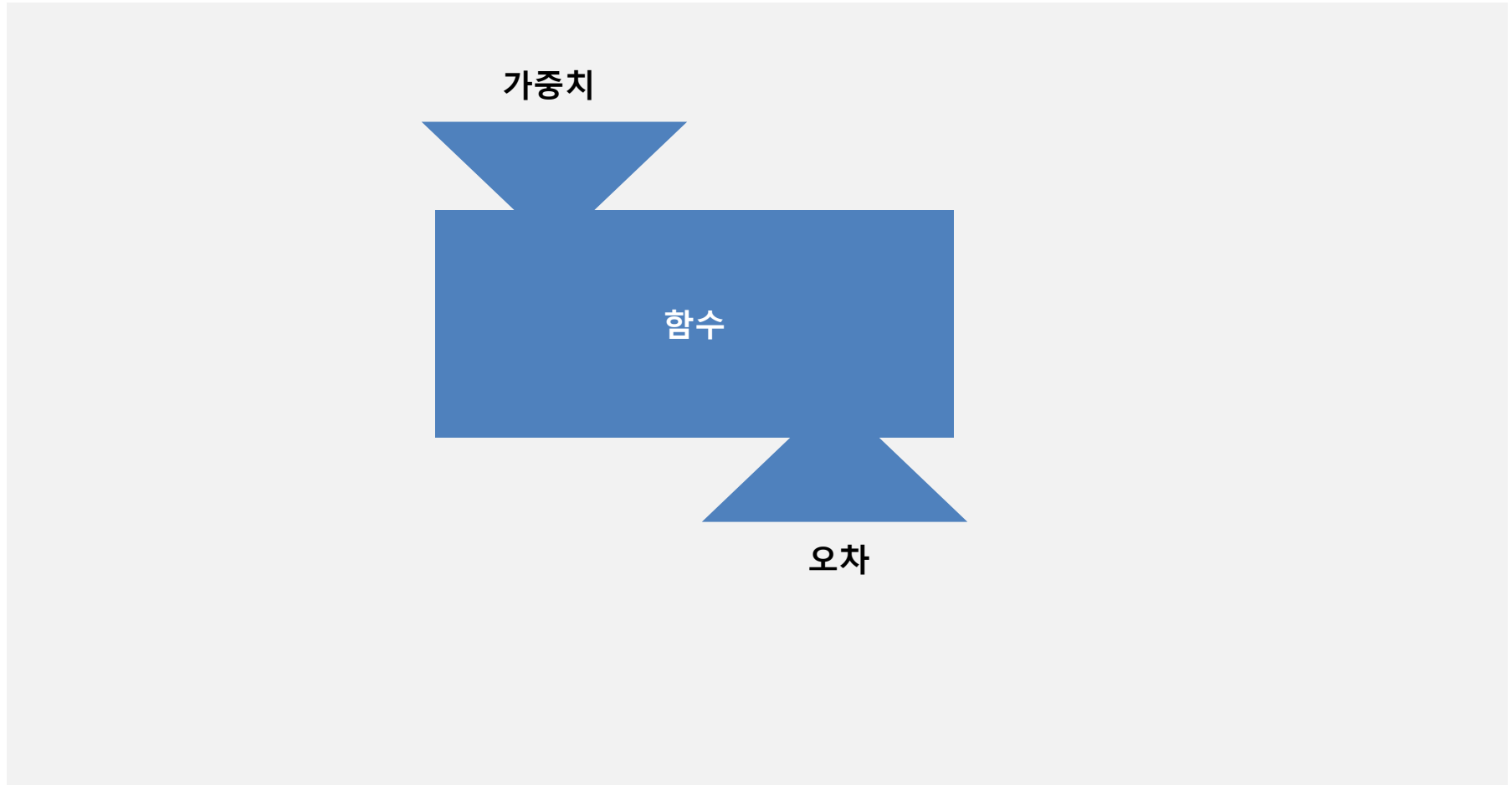
### 모형의 오차: Classification VS Regression



## 4. Artificial Neural Network의 가중치 찾기

---

### 가중치와 오차의 함수

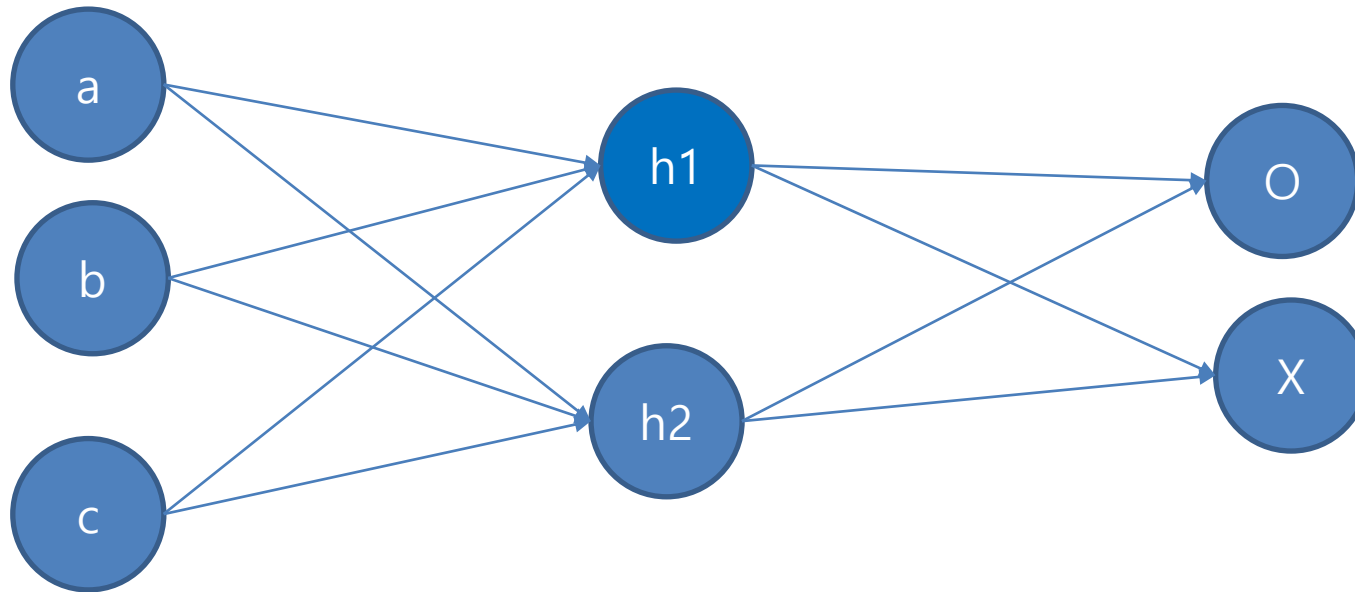


## 4. Artificial Neural Network의 가중치 찾기

---

### 하이퍼파라미터:

학습율(Learning rate), 모멘텀 파라미터, 은닉층과 노드의 수,  
미니배치 크기, 학습율 감쇠 정도, 최적화알고리즘 파라미터 등





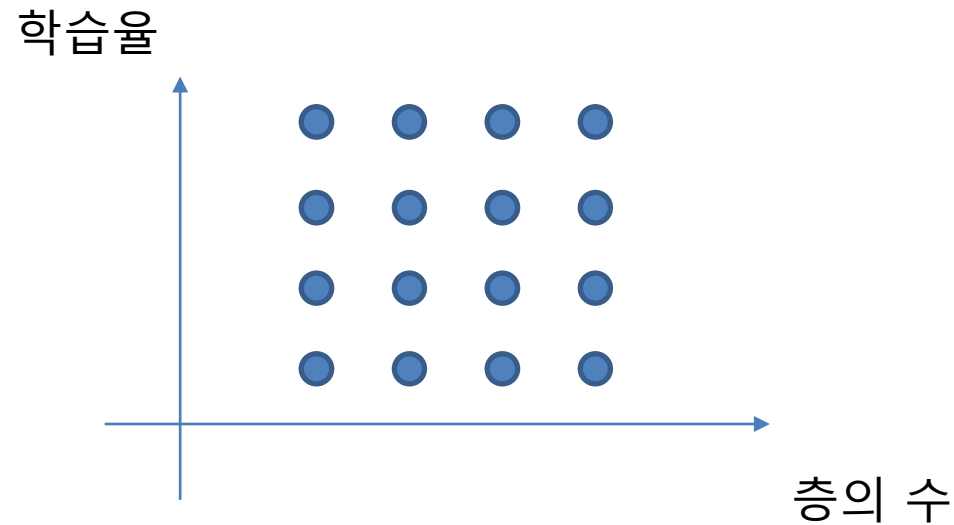
## 4. Artificial Neural Network의 가중치 찾기

---

### Grid Search?

모든 가능한 하이퍼파라미터의 조합을 체계적으로 하나씩 탐색하는 방식

예: 2개의 파라미터의 조합

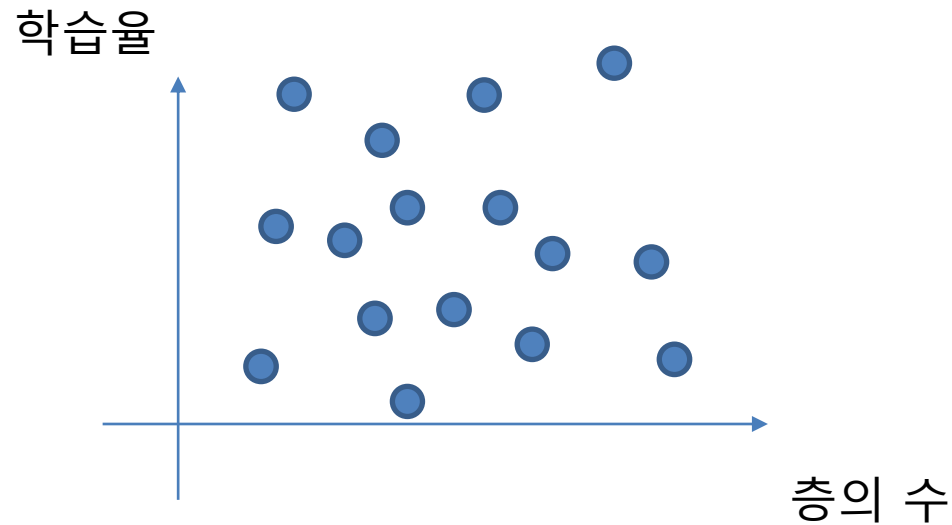


## 4. Artificial Neural Network의 가중치 찾기

### Random Search?

Random하게 하이퍼파라미터의 조합을 탐색

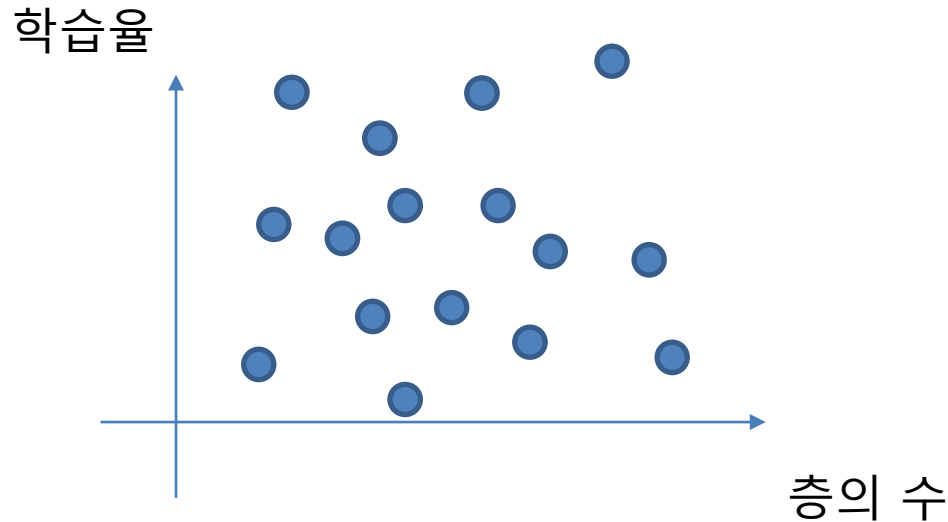
예: Random하게 생성되는, 파라미터 2개에 의한 값의 조합



**Grid Search보다 우수할 수도!**

## 4. Artificial Neural Network의 가중치 찾기

### Random Search?



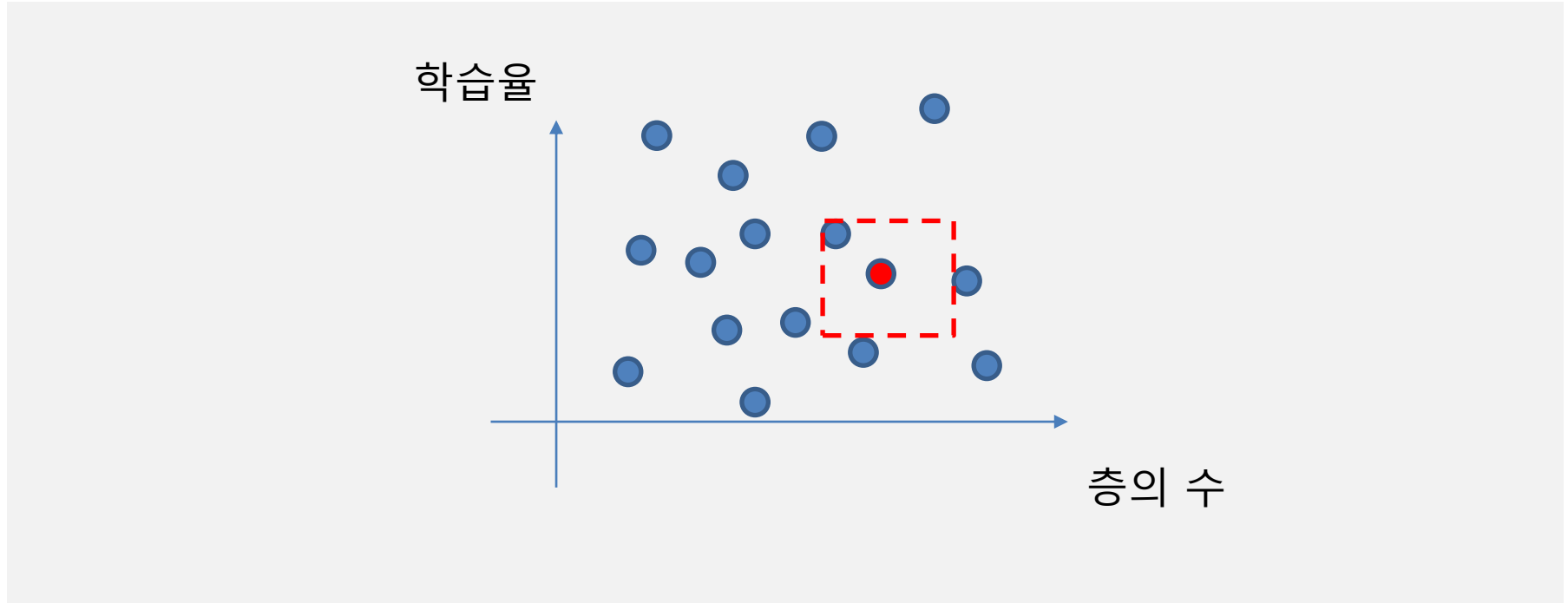
***Don't use Grid!***  
from Andrew Ng

- 왜? 인공신경망에서는 어떤 파라미터가
- 성능에 어느 정도 영향을 주는지를 알 수 없음
- 중요한 하이퍼파라미터 관점에서 먼저 탐색!

예: 두 조합 중 학습율이 있다면, 학습율 우선으로 고려,  
이후 전체 조합의 개수 산출, 다른 파라미터 값을 검색

## 4. Artificial Neural Network의 가중치 찾기

정밀화 접근!

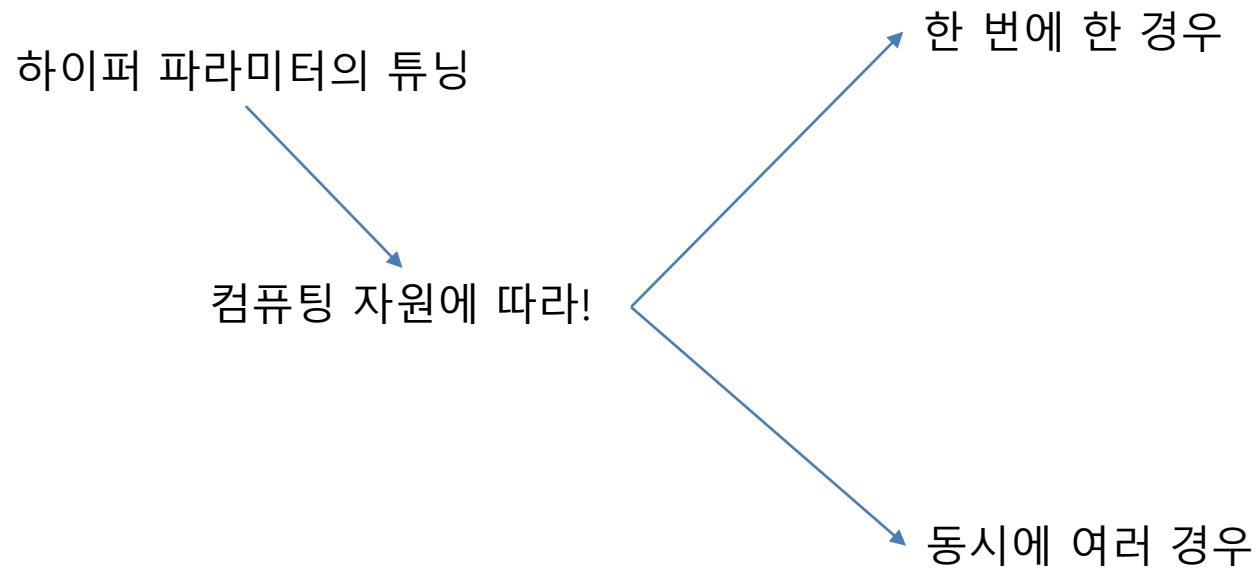


*Random한 탐색에서 발견한 최적의 하이퍼파라미터 주변을 더 탐색!*

## 4. Artificial Neural Network의 가중치 찾기

---

한 번에 하나! VS 한 번에 여러 개!



### III. ML 소개2

---

1. 패턴: Association Rule
2. 군집: Clustering (K & H)
3. 그래프: Graph mining
4. 추천: Recommendation



# 1. 패턴: Association Rule

---

## ➤ Association Rule

- 기저귀와 맥주?
  - 다수의 거래 내역 각각에 포함된 품목(Item)의 관찰을 통해 규칙 발견
    - Find all rules that satisfy the user-specified criterias, minimum support (minsup) and minimum confidence (minconf)
    - Agrawal et al. (1993)에 의해 소개 : R. Agrawal, T. Imielinski and A.N. Swami, Mining association rules between sets of items in large databases. Proceedings of SIGMOD 1993
    - 모든 데이터를 Categorical 가정(Numeric data에는 적절하지 않음)
    - 장바구니 분석에 처음 사용됨
      - » 예: Bread → Milk [sup = 5%, conf = 100%]
      - » 빵을 사는 사람은 우유도 산다
- 특징
  - 모든 규칙을 찾음(Completeness)
  - 특정한 Target 변수 없음
  - 다소 간단하지만 많이 사용됨 (많이 사용되는 데이터마이닝 방법 상위 10위 안에...)



# 1. 패턴: Association Rule

## ➤ Association Rule

### - 지지도(Support)

- 전체 자료에서 관련성이 있다고 판단되는 품목들을 포함하고 있는 거래나 사건의 확률( 두 항목이 동시에 일어날 확률)인  $P(A \text{ and } B)$

$$support = P(X \cap Y) = \frac{n(X, Y)}{N}$$

- 아무리 X나 Y가 연관성이 높은 유의미한 Rule이라 해도, X나 Y가 전체 거래에서 Coverage 가 작으면 값이 작게 나옴
- X를 구매할 경우, Y도 구매할 확률은 얼마인가와 같은 패턴을 찾기 위해 다른 기준이 필요

### - 신뢰도(Confidence)

- 어떤 항목 X가 구매되었을 때, Y가 추가로 구매될 조건부 확률

$$confidence = P(Y | X) = \frac{P(X \cap Y)}{P(X)} = \frac{X \text{와 } Y \text{를 동시에 포함하는 거래의 수}}{X \text{의 거래의 수}}$$

### 예시: Transaction data

- Clothes → Milk, Chicken : 지지도: 3/7, 신뢰도: 3/3
- Clothes, Chicken → Milk : 지지도: 3/7, 신뢰도: 3/3

t1:	Beef, Chicken, Milk
t2:	Beef, Cheese
t3:	Cheese, Boots
t4:	Beef, Chicken, Cheese
t5:	Beef, Chicken, Clothes, Cheese, Milk
t6:	Chicken, Clothes, Milk
t7:	Chicken, Milk, Clothes

# 1. 패턴: Association Rule

---

## ➤ Association Rule

- Lift(향상도)

- A를 고려한 B의 구매확률을 A를 고려하지 않은 B의 구매확률로 나눈 것으로, 이 값이 높다면 우연에 의해 연관성이 나타남
- 두 값이 독립인 경우에는 분모 분자가 동일하고 Lift=1, 독립이 아닌 경우(연관된 경우)에는 분모, 분자 값이 다르게 됨

$$Lift(A \Rightarrow B) = \frac{P(B | A)}{P(B)} = \frac{P(A \cap B)}{P(A)P(B)}$$

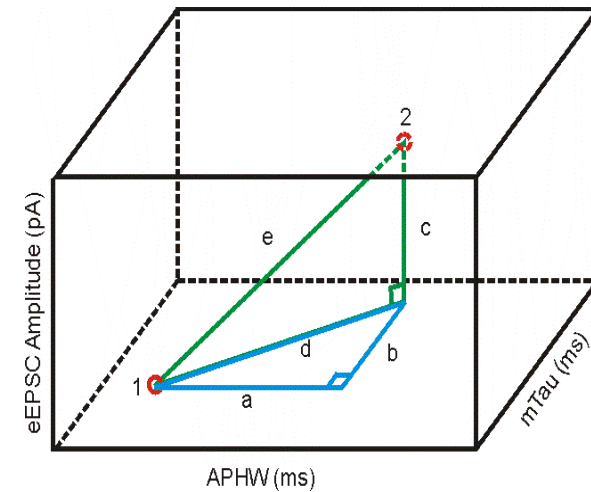
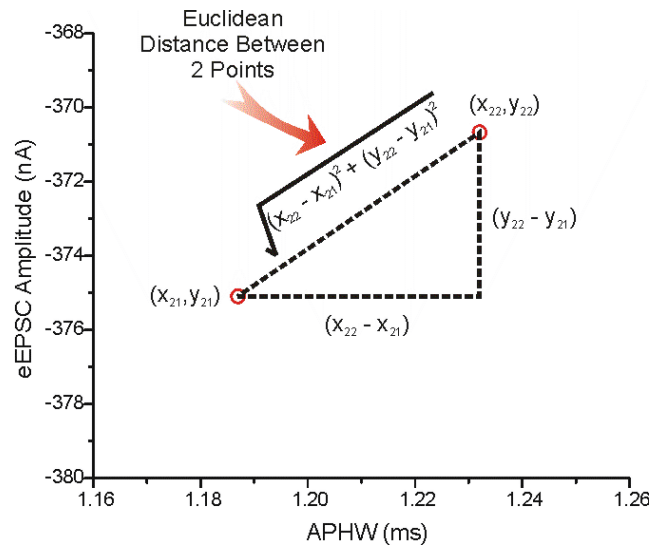
- 실제 분석 시,

- 지지도로 일정 구매비율 이상의 항목을 찾기
- 대신 실제로 연관성이 크지만 구매비율이 낮은 규칙은 신뢰도로 찾기
- Lift가 1 이상인 규칙을 찾기

## 2. 군집: Clustering

### ➤ Clustering?

- Cluster의 개수나 구조에 관한 특별한 사전 가정없이, 개체들 사이의 유사성/거리에 근거해 cluster를 찾고 다음 단계의 분석을 하게 하는 기법
- 유사한 개체들을 cluster로 그룹화하여 각 집단의 성격을 파악



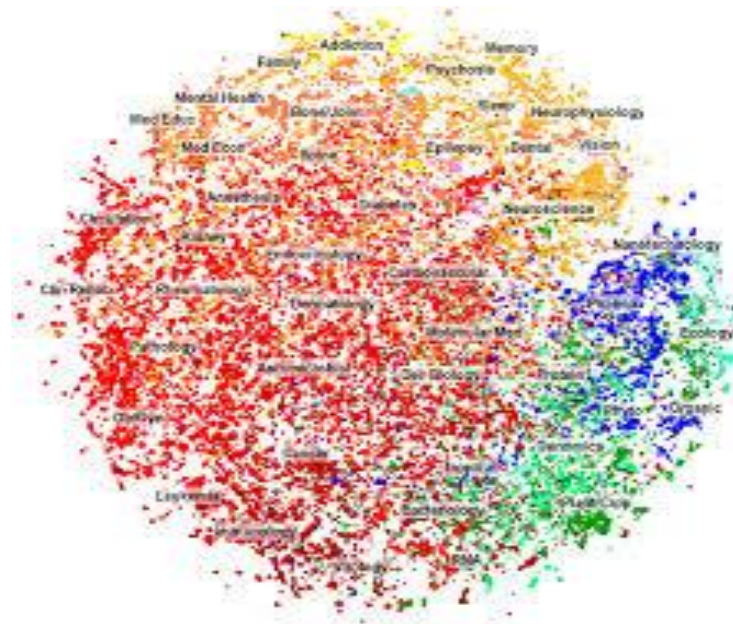
## 2. 군집: Clustering

## ➤ Clustering 장점

- 데이터를 탐색하는 기법
- 추가적인 분석을 위해 사용할 수 있음
- 유사성, 비유사성만 계산할 수 있다면 여러 형태 데이터 적용 가능
- 분석 용이한 장점

## ➤ Clustering 단점

- 자료유형이 혼합된 경우, 거리 정의 등이 어려울 수 있음
- 초기군집수 설정이 중요
- 결과해석에 주의

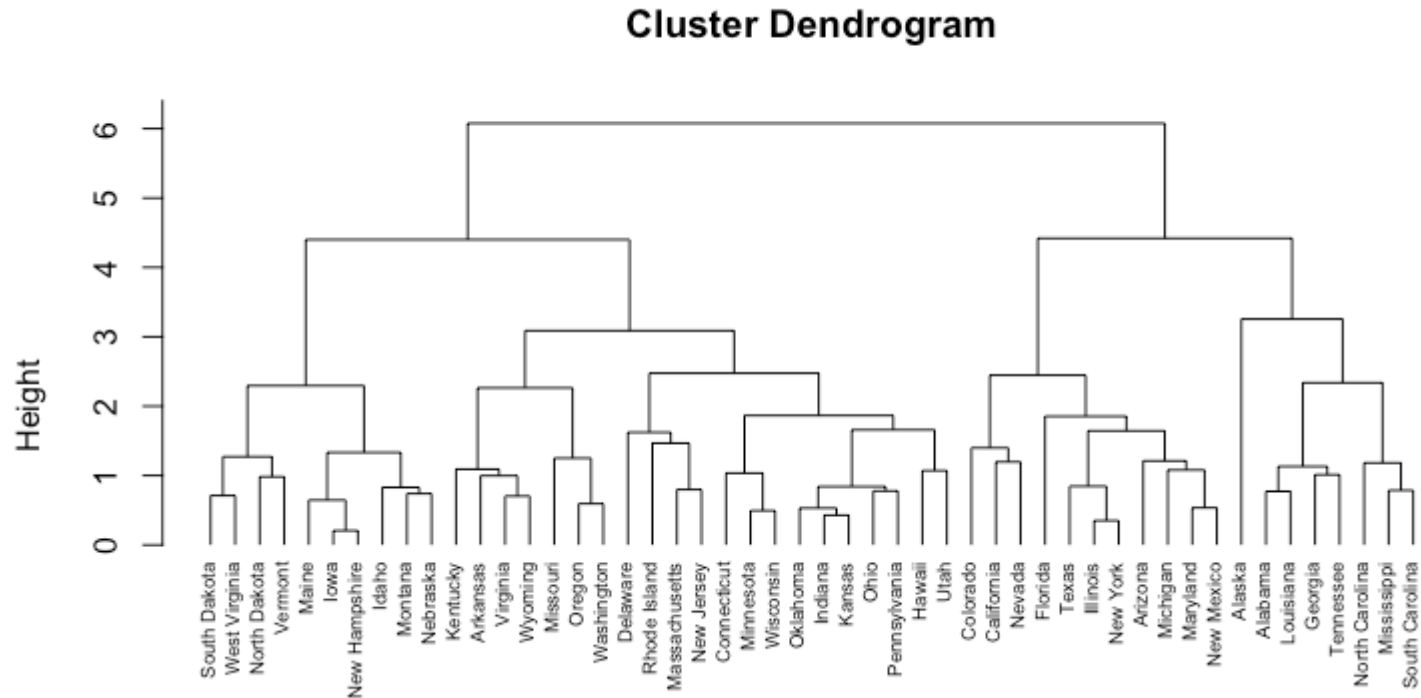


## 2. 군집: Clustering

### ➤ Clustering

#### – Hierarchical clustering

- 가까운 개체들을 하나씩 묶어나가면서 클러스터 형성
- 예:  $n$ 개의 개체 중 가장 가까운 개체를 묶고, 다른  $n-2$ 는 각각이 하나의 군집,  $n-1$ 개 군집이 생성, 반복하여 1개의 군집이 되도록 함, dendrogram으로 표현

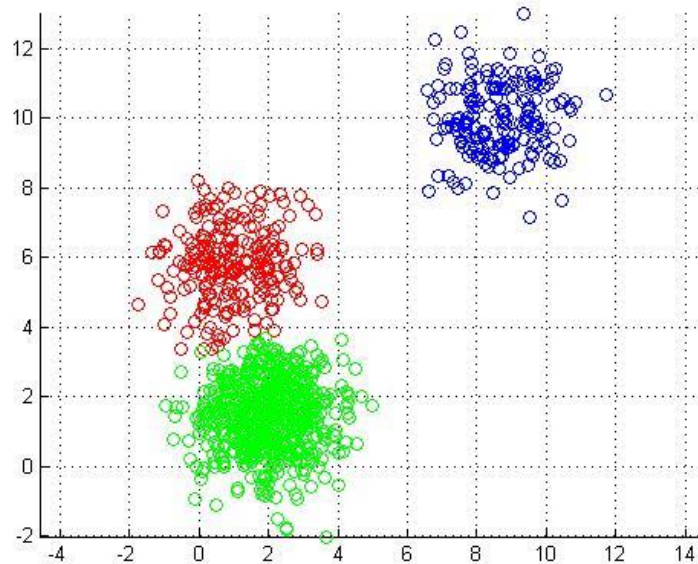


## 2. 군집: Clustering

### ➤ Clustering

#### – K-means clustering

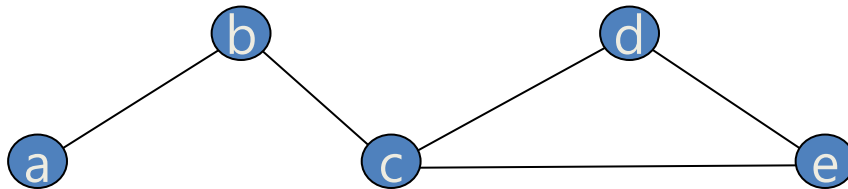
- K개: 주어진 cluster의 개수
- 절차: k개 만큼의 cluster seed 선택(지정/임의로 선택 등)
- 각 개체들에 대해 cluster seed와 거리 계산하여, 가장 가까운 seed에 개체 할당
- 각 개체가 seed에 할당될 때마다, 군집의 중심이 그 군집에 속하는 개체들의 평균 벡터로 다시 계산
- 개체들을 가장 가까운 Cluster seed에 재할당하고, 이 과정을 cluster center의 변화가 일정 수준 이하가 되도록 반복
- 초기 군집수 결정이 중요: EDA를 통해 파악(시각화), 주성분 분석을 통해 2,3차원 그래프로 분석



### 3. 그래프: Graph mining

#### ➤ Graph?

- Actors와 relations (또는 "nodes" and "edges")으로 구성
- Graph는 Node들의 연결에 대한 패턴에 대해 두 가지 방법으로 표현: graphs와 matrix
  - 특히 graphics는 "socio-grams " 으로 불리기도 함
  - 수학자들은 "directed graphs" "signed graphs" or simply "graphs " 등으로 지칭



Undirected, binary

	a	b	c	d	e
a		1			
b	1		1		
c		1		1	1
d			1		1
e			1	1	

### 3. 그래프: Graph mining

---

➤ 주요 Measure1

- Degree: Number of links to a vertex(indegree, outdegree...)
  - 노드가 갖는 연결의 수
  - Indegree: 노드로 들어오는 연결의 수
  - Outdegree: 다른 노드로 나가는 연결의 수
- Density: sum of tie values divided by the number of possible ties
  - 전체 연결의 수를 전체 가능한 연결의 수로 나눈 정도
- Reciprocity: the proportion of dyads which are symmetric
  - 노드에 회귀적인 연결의 수



### 3. 그래프: Graph mining

---

➤ 주요 Measure2

- Transitivity: the total number of transitive triads is computed
  - 노드 i에 모두 영향을 받는 노드 j와 노드 j가 있을 때, 노드 j가 노드 h에 영향을 주는 연결의 수
- Betweenness Centrality: Number of shortest paths pass it
  - 서로 다른 두 노드를 연결해주는 정도
- Closeness Centrality: Length to all other vertices
  - 각 노드에서 모든 다른 노드까지의 거리

### 3. 그래프: Graph mining

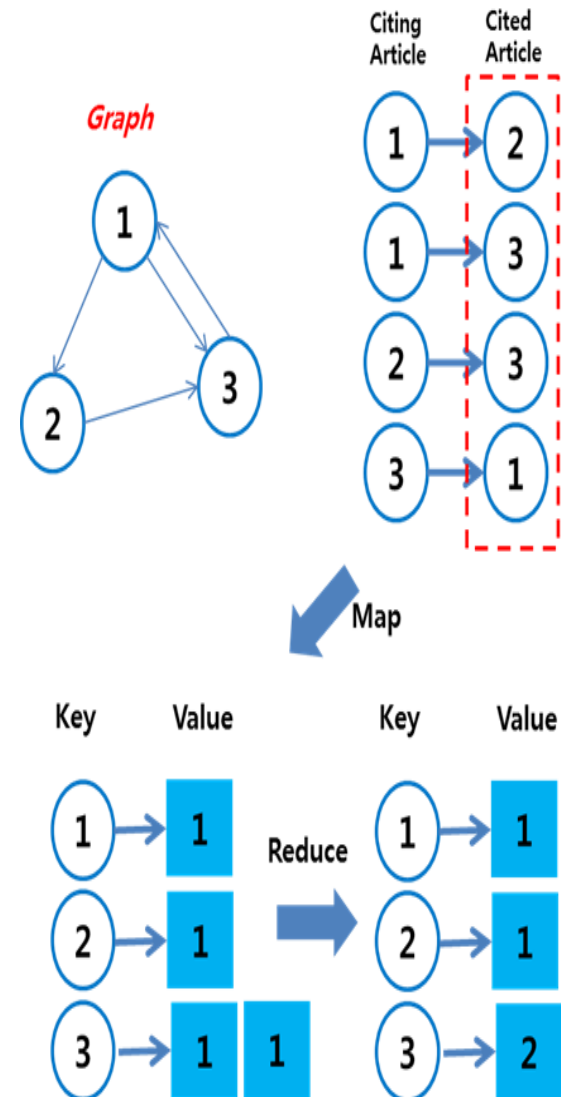
#### ➤ Graph?

- Web scale의 그래프 표현/분석

#### ➤ Graph degree 계산

- 그래프 Degree 계산은 그래프 분석의 시작
- 그래프는 행렬로 표현이 되는데, 크기가 급증한 행렬의 분석에는 큰 규모의 계산과 자원이 필요

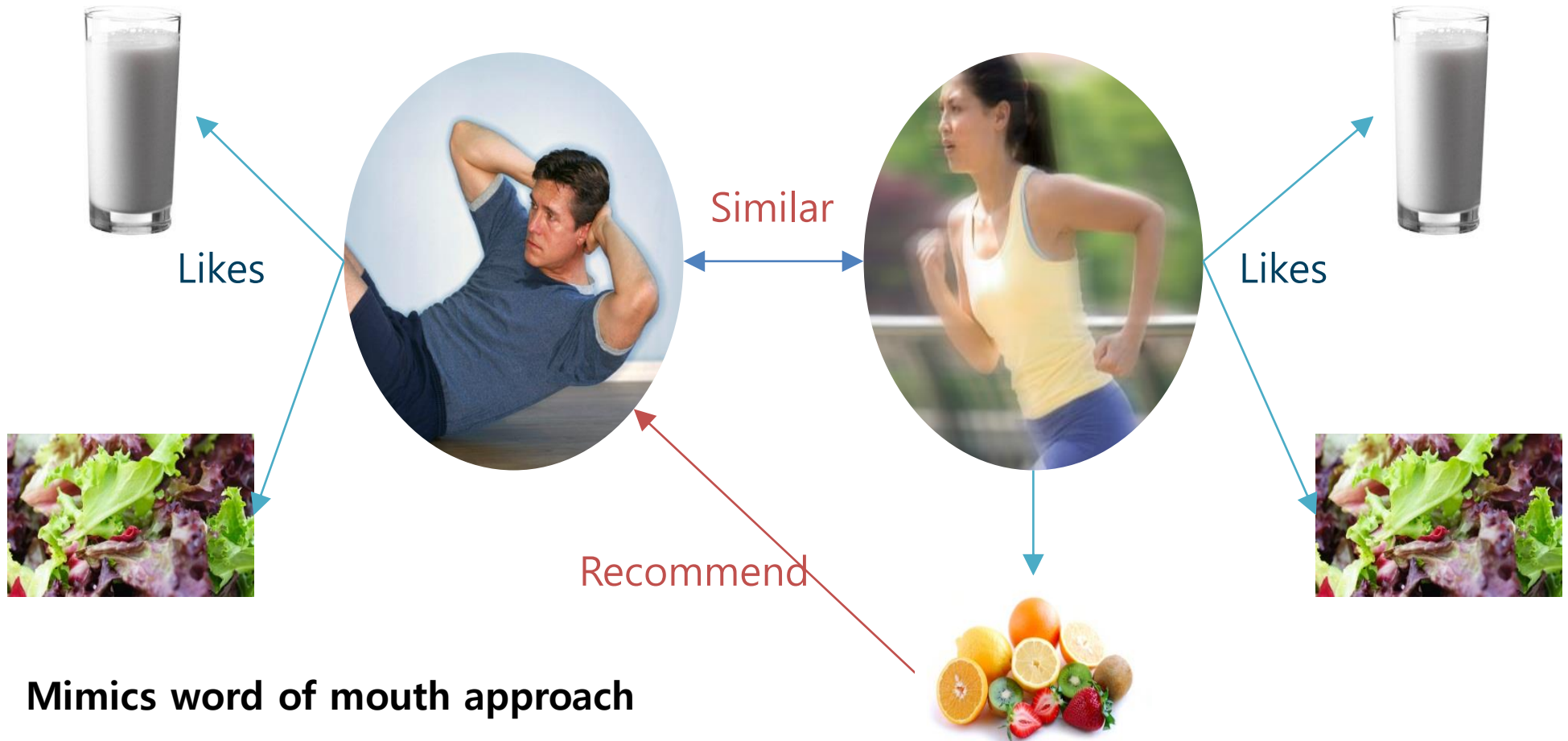
#### ➤ Graph degree 계산: In-degree



## 4. 추천: Recommendation

- Collaborative Filtering

### User Based Collaborative Filtering

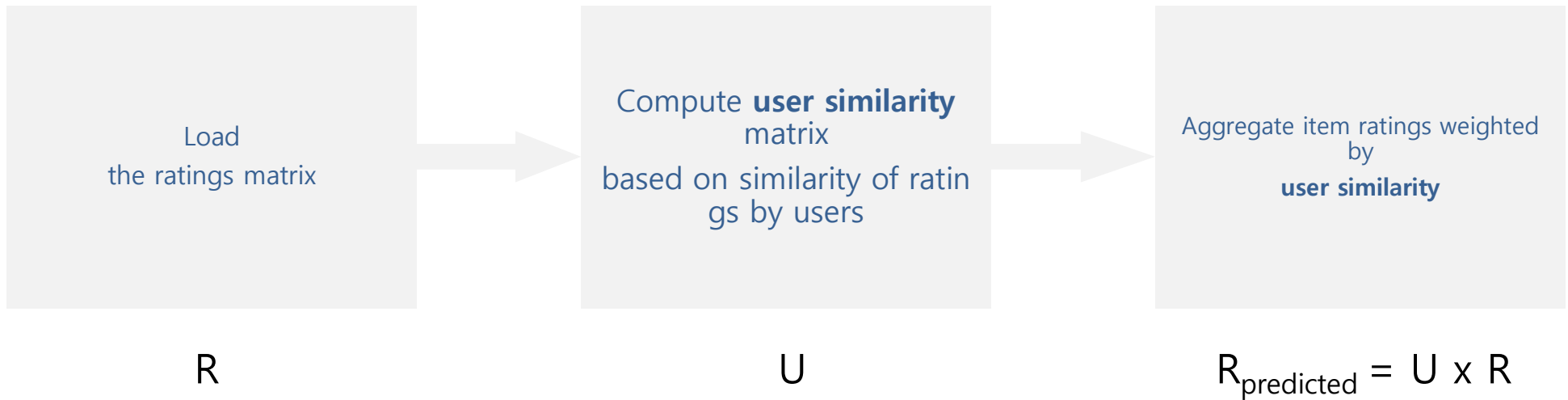


## 4. 추천: Recommendation

### ➤ Collaborative Filtering

- 유사한 선호를 갖는 이용자는 아이템도 유사하게 평가

#### User Based Collaborative Filtering



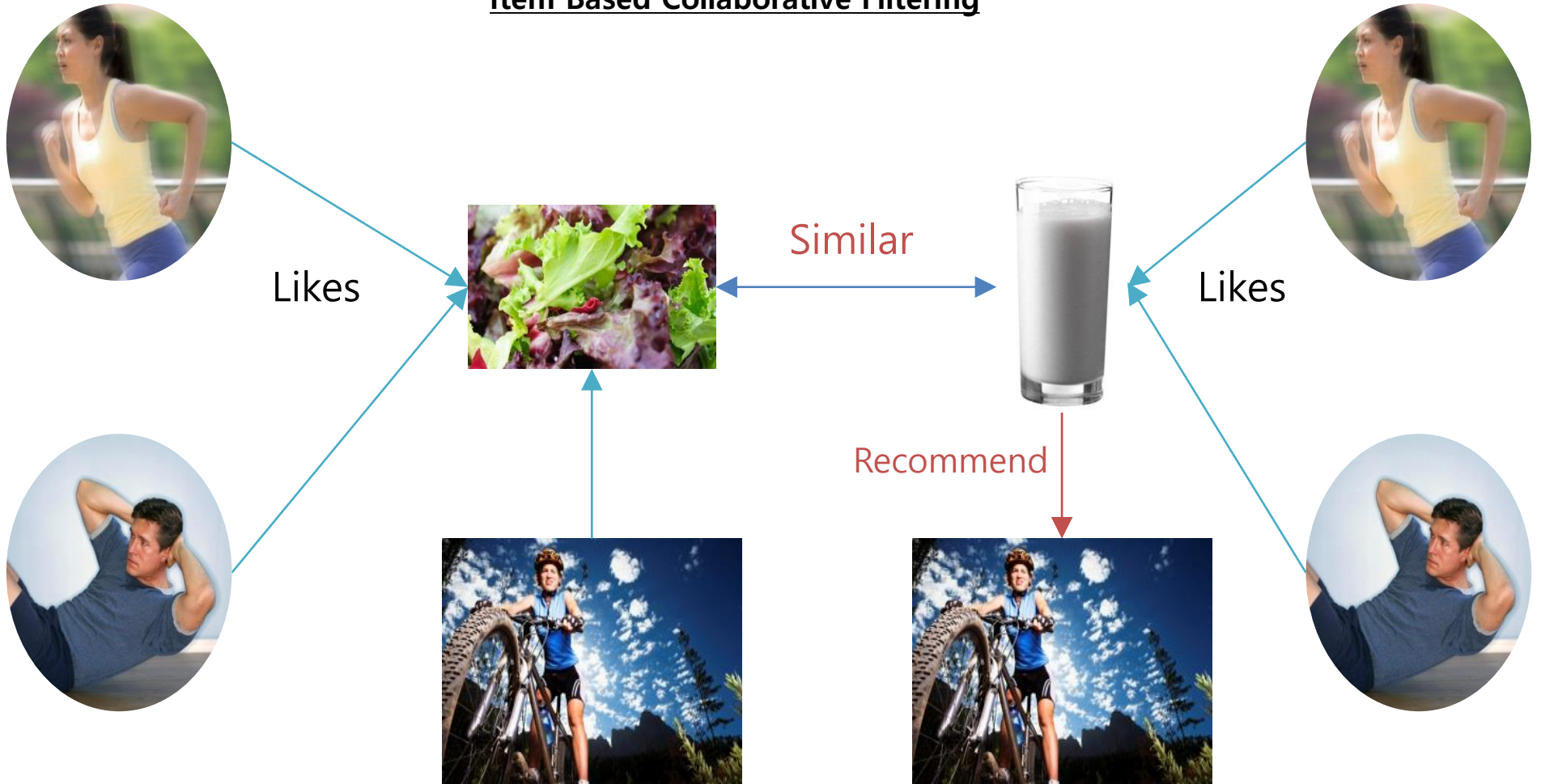
Rating for user i, for item k

$$r'_{ik} = \frac{\sum_j s_{ij} * r_{jk}}{\sum_j s_{ij}}$$

## 4. 추천: Recommendation

- Collaborative Filtering

### Item Based Collaborative Filtering

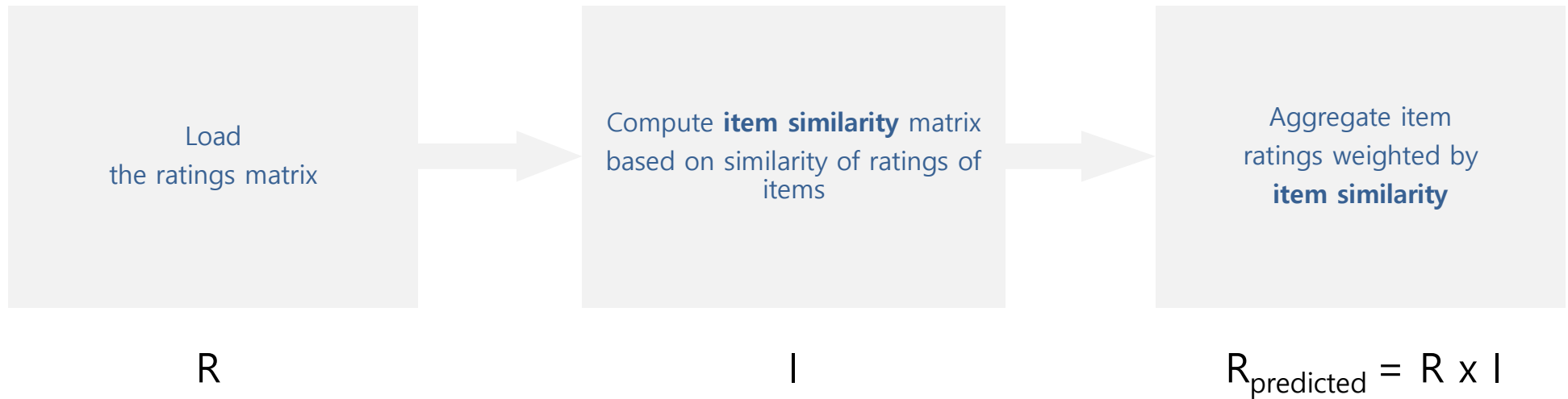


## 4. 추천: Recommendation

### ➤ Collaborative Filtering

- 이용자들은 이미 좋아하는 아이템과 유사한 아이템을 선호함

#### Item Based Collaborative Filtering



Rating for user  $i$ , for item  $k$

$$r'_{ik} = \frac{\sum_j r_{ij} * s_{jk}}{\sum_j s_{jk}}$$

---

# Q&A