



# 인공신경망

2023

## I. 개요 및 현황

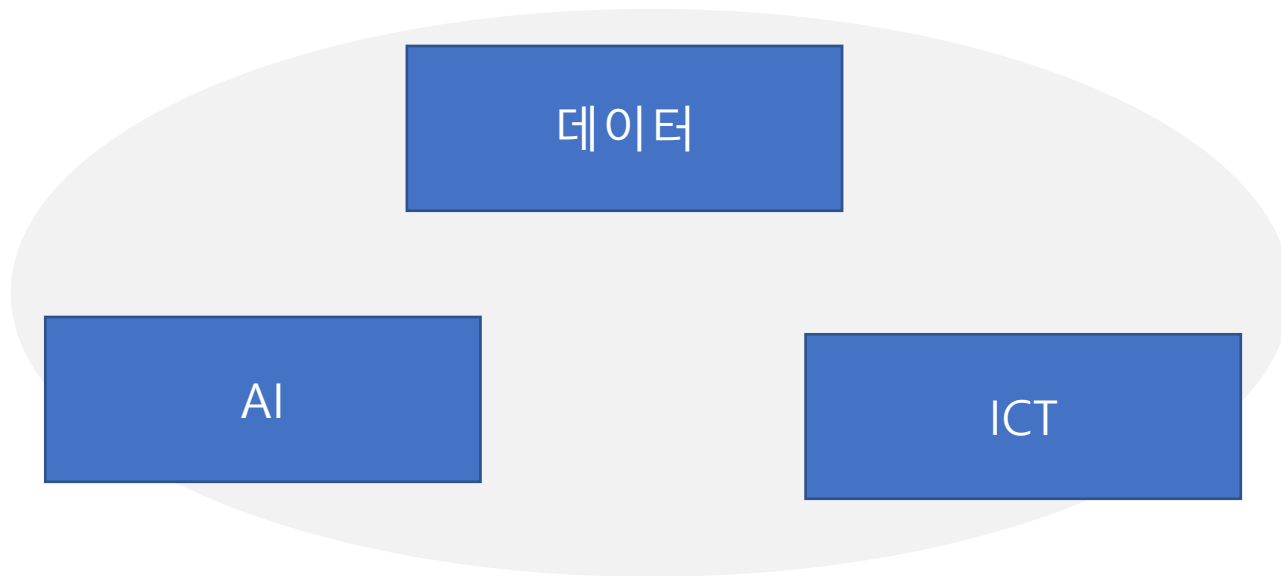
---

1. 인공신경망: 뉴런을 모방한 노드들이 각각 Input Layer, Hidden Layer, Output Layer로 구분되며 데이터를 입력받아 변환하여 원하는 결과로 출력하는 네트워크를 구축하는 것
2. 입력값에 대해 은닉층에서의 비선형 변환을 통해 출력을 예측하는 인공신경망 기법에서, 은닉층에서의 비선형변환이 여러 차례 반복되는 심층신경망에서의 학습기법이 딥러닝
3. 딥러닝은 다수의 은닉층에서도 잘 작동하며, 많은 Feature에 대해 많은 계산을 수행하며, 높은 예측력을 보여줌

## I. 개요 및 현황

---

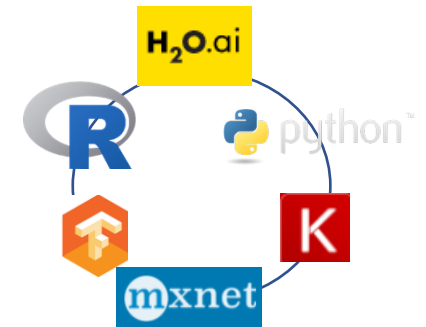
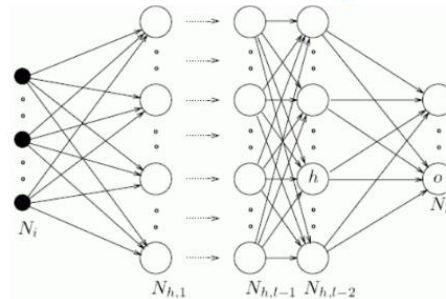
### 4차산업혁명과 DT



# I. 개요 및 현황

## AI와 머신러닝

- Machine = 컴퓨터
- Learning = 분석
- “컴퓨터가 데이터를 분석하는 알고리즘과 기술”



## I. 개요 및 현황

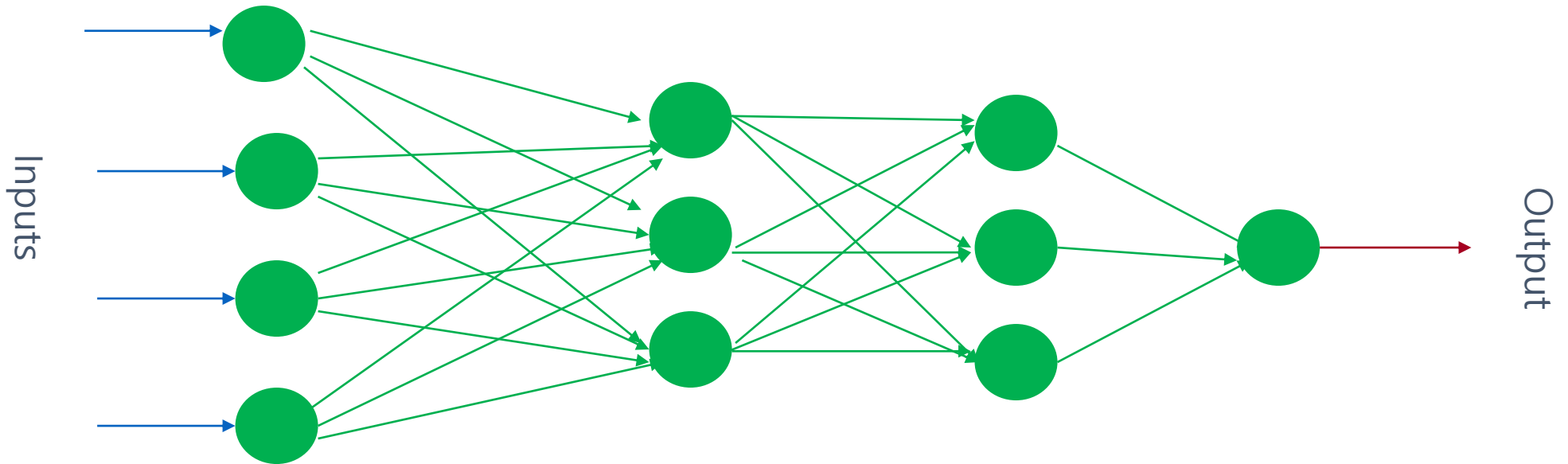
---

- 인공신경망 (Artificial Neural Network, ANN): 데이터 기반의 AI
  - 사람 뇌의 정보처리 방식을 모사한 알고리즘
  - 데이터를 분석하여 예측, 분류 등 의사결정 문제를 해결하는데 적용할 수 있는 기법



# I. 개요 및 현황

- Artificial Neural Network(ANN, 인공신경망)
  - 데이터를 입력받아 변환하여 원하는 결과로 출력: 각각 Input Layer, Hidden Layer, Output Layer로 구성
  - 예측 성능이 우수
  - 모형을 통한 추론은 어려움

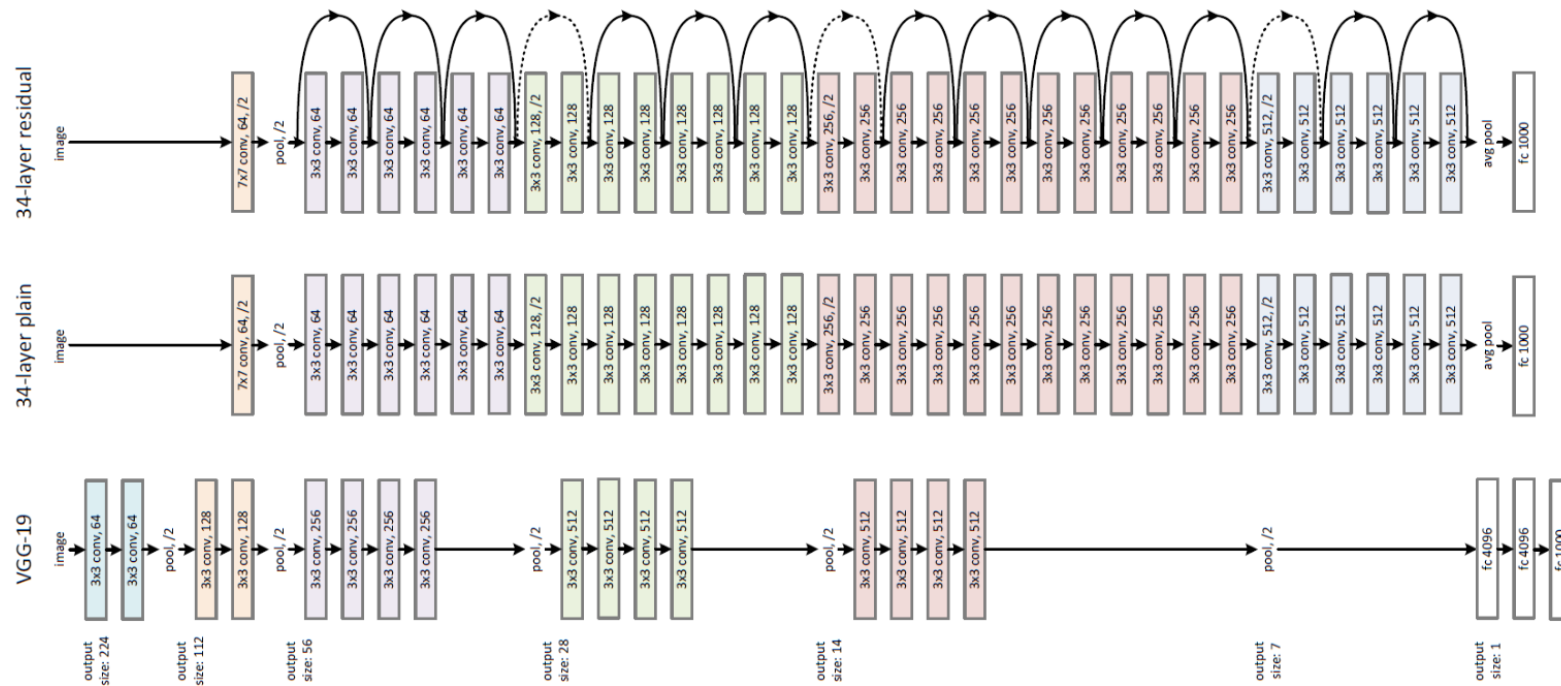


# I. 개요 및 현황

## 인공신경망+깊이

- Hidden layer가 2개 이상인 NN(Neural Network)을 Deep Neural Network(DNN)
- DNN에서의 학습을 딥러닝이라고 부름

### 비선형 변환과정들

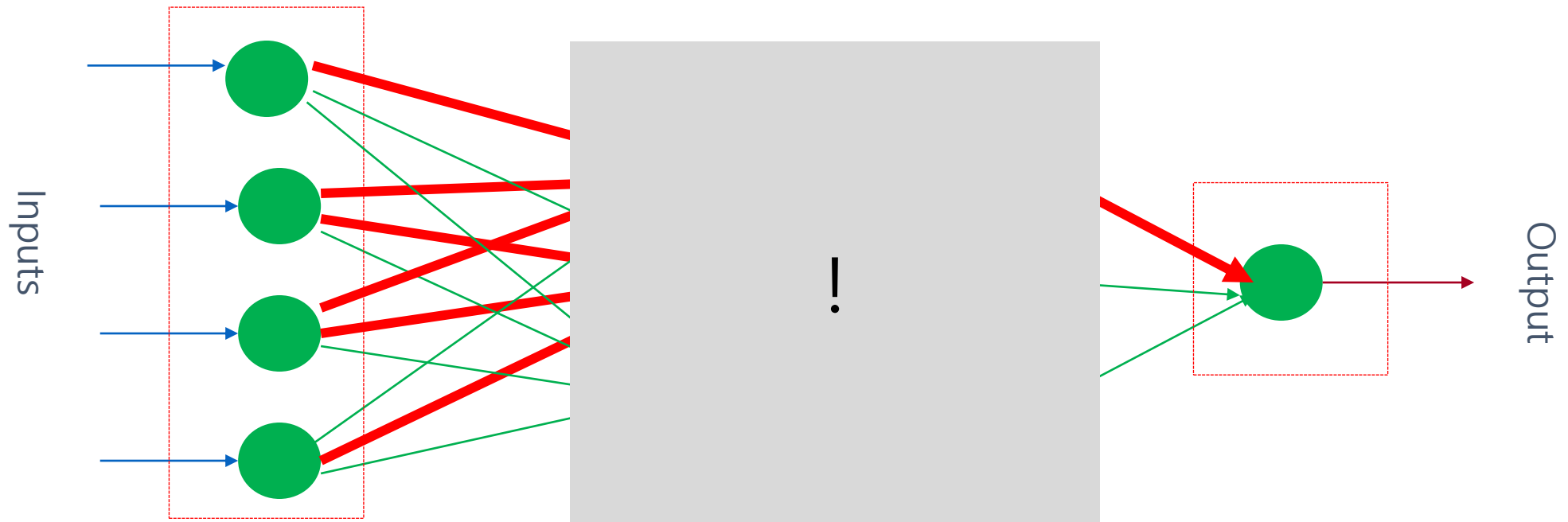


RESNET

# I. 개요 및 현황

- 딥러닝의 특성

- 다수의 은닉층 = 계산량

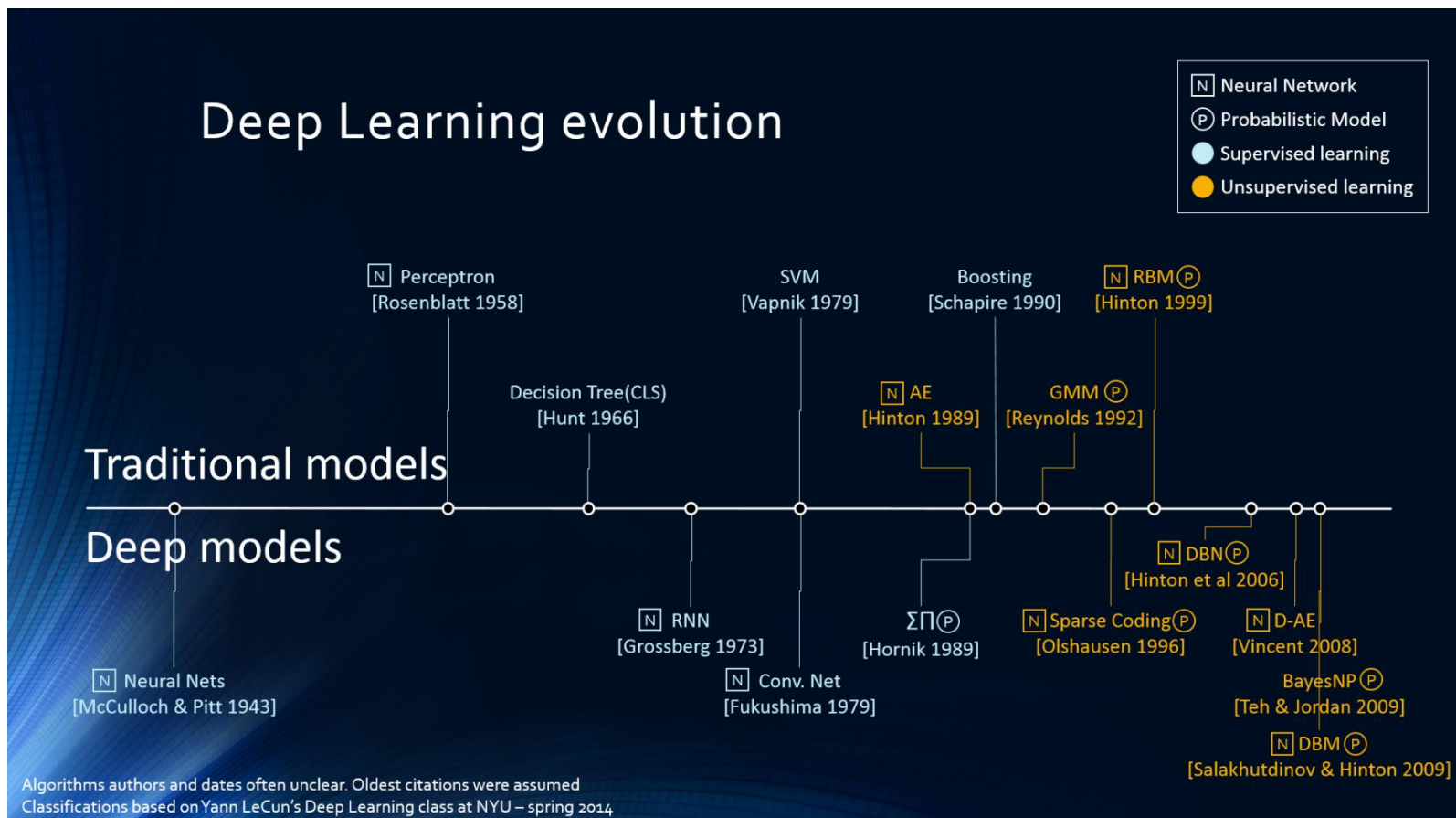




# I. 개요 및 현황

## • 딥러닝 알고리즘의 발전

- 기존 ANN의 난제들을 해결
- CNN, RNN, LSTM, Transformer, BERT 등으로 발전 중



## II. Deep VS Shallow Learning

---

1. 변수는 관측된 개체들이 갖는 특성 또는 속성으로 데이터를 구성하며, 모형은 주어진 변수를 활용하여 데이터를 분석하고 문제를 해결하는 기법이나 알고리즘
2. Shallow Learning: 다수의 비선형 변환을 거치지 않고 주어진 변수로 부터 직접적으로 모형을 도출
3. 분석의 목적이 설명인지, 예측인지에 따라, 변수를 다루는 방식이 다른 딥러닝과 셸로우러닝 중 선택해서 사용해야 함

## II. Deep VS Shallow Learning

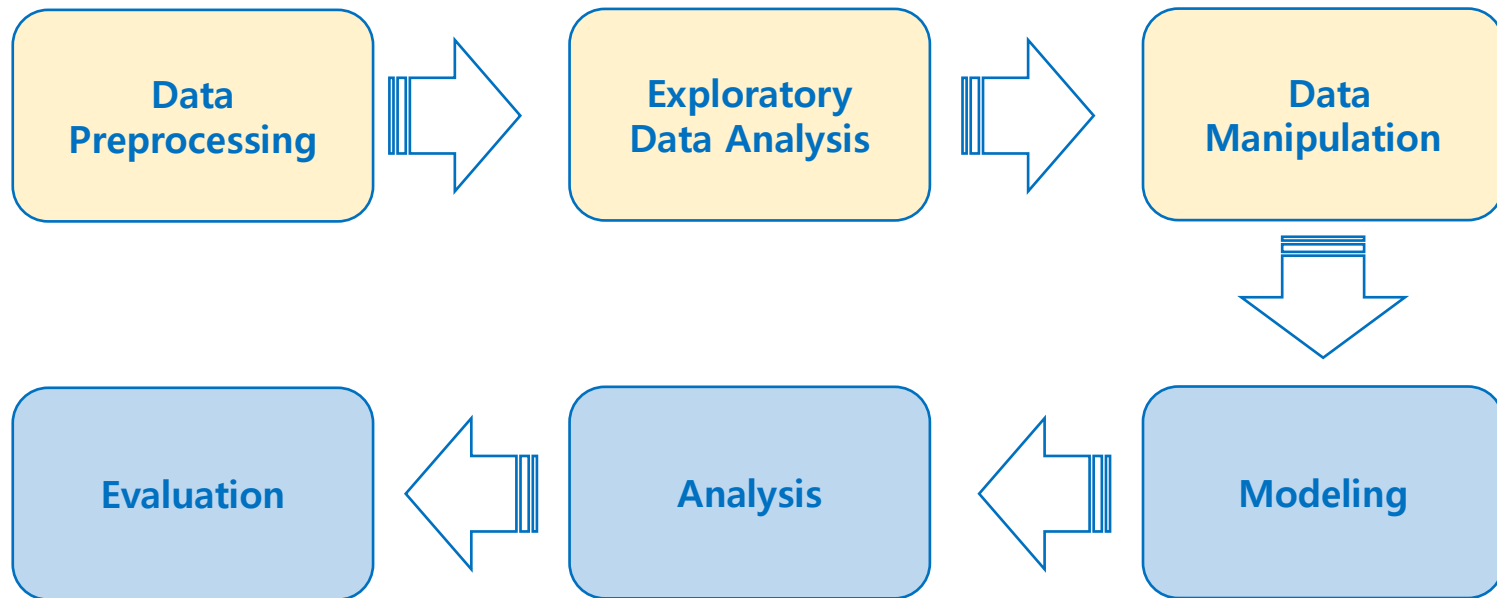
---

변수

: 관측된 개체들이 갖는 특성 또는 속성으로, 다양한 값을 갖을 수 있음, Feature라고도 지칭

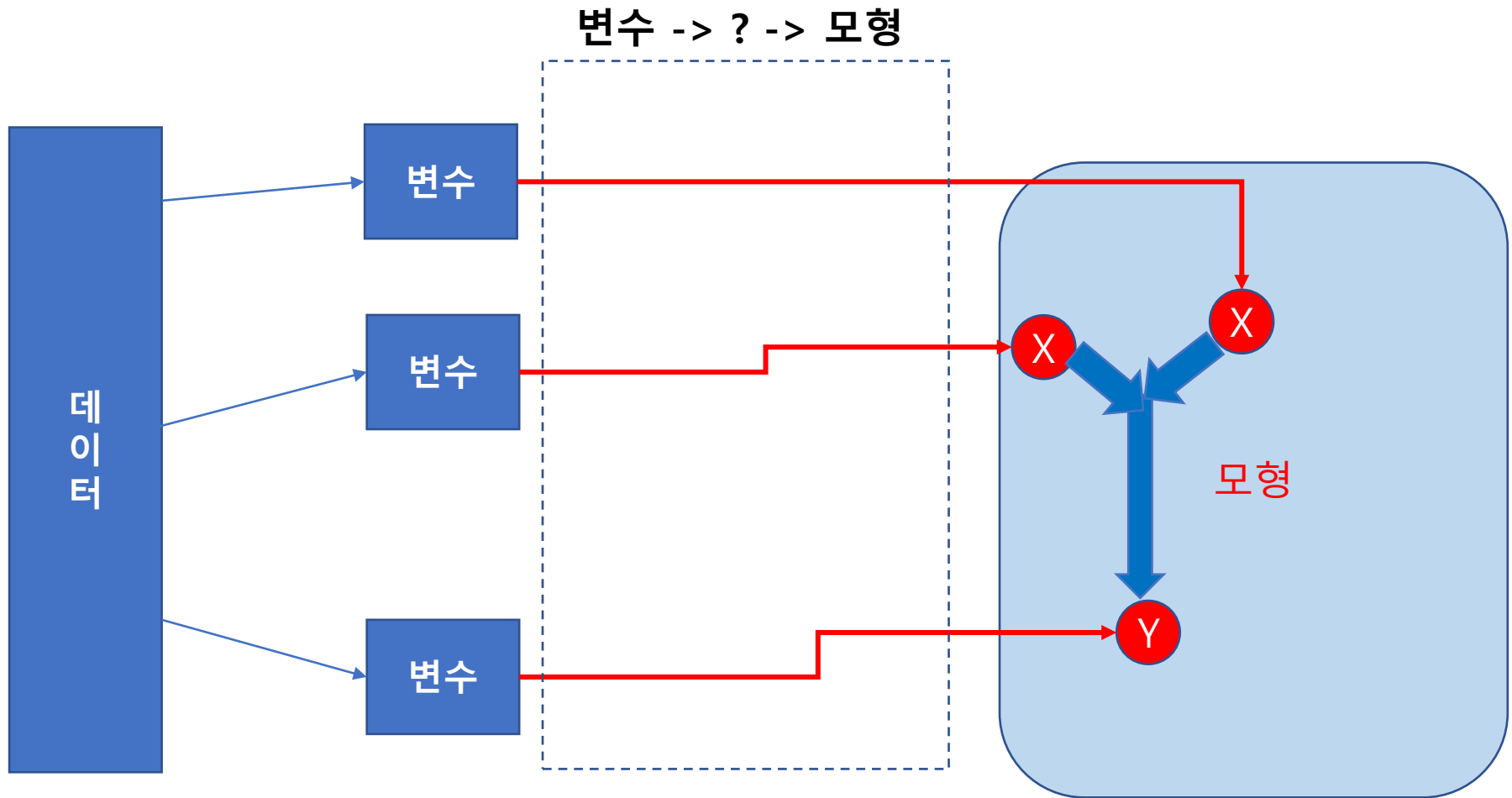
모형

: 분석 목적에 맞게 데이터에서 인사이트를 찾아내는 적절한 기법이나 알고리즘



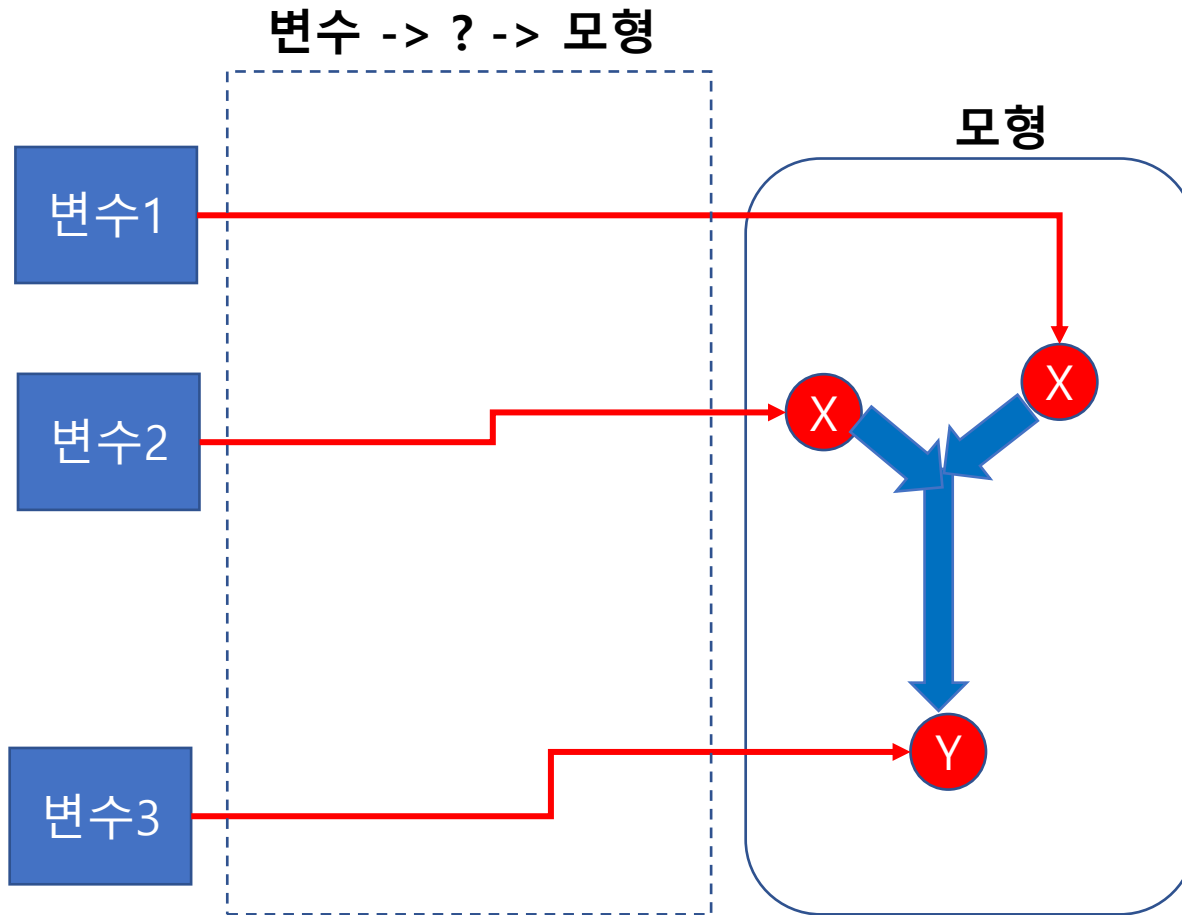
## II. Deep VS Shallow Learning

변수에서 모형으로



## II. Deep VS Shallow Learning

### Shallow Learning?



Shallow Learning:  
주어진 변수를 별도로 다수의 비  
선형 변환없이 그대로 사용해서  
모델링

## II. Deep VS Shallow Learning

---

### Shallow Learning

예:

IF

변수1 > 50,

TEHN

내일 비가 오는 것으로 예측!

예:

변수1 X 20 + 10 = 변수3

## II. Deep VS Shallow Learning

### Shallow Learning VS Deep Learning

예:

IF  
    변수1 > 50,  
TEHN  
    내일 비가 오는 것으로 예측!

예:

변수1 X 20 + 10 = 변수3

예:

변수1의 20%, 변수2의 80%를 반영해서  
합산한 값에  
활성화함수를 통해 비선형 변환을 적용해서  
얻어진 값이 새로 만든 변수 1

변수1의 40%, 변수2의 60%를 반영해서  
합산한 값에  
활성화함수를 통해 비선형 변환을 적용해서  
얻어진 값이 새로 만든 변수 2

새로 만든 변수 1과 2에 대해 비선형 변환을  
적용

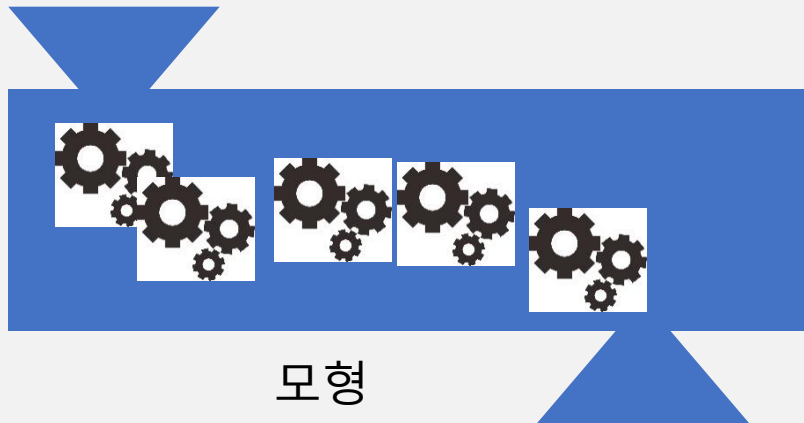
....

그 결과로 Y변수를 모델링

## II. Deep VS Shallow Learning

분석의 목적: 설명? 예측?

딥러닝과 쉬로우러닝의 선택!



엄선된 적은 수의 변수 하나하나가  
중요하고, 모형의 설명이 중요한 경우!

**Shallow**

장점: 설명 가능성  
단점: 모형의 데이터 Fitting

?

**Deep**

단점: 설명 가능성  
장점: 모형의 데이터 Fitting

변수(=feature)가 너무 많아 설명보다는  
피팅이 중요한 경우!



## II. Deep VS Shallow Learning

---

### 딥러닝의 활용 분야

많은 Feature가 제공되며, 예측을 잘 해야 하는 분야들!



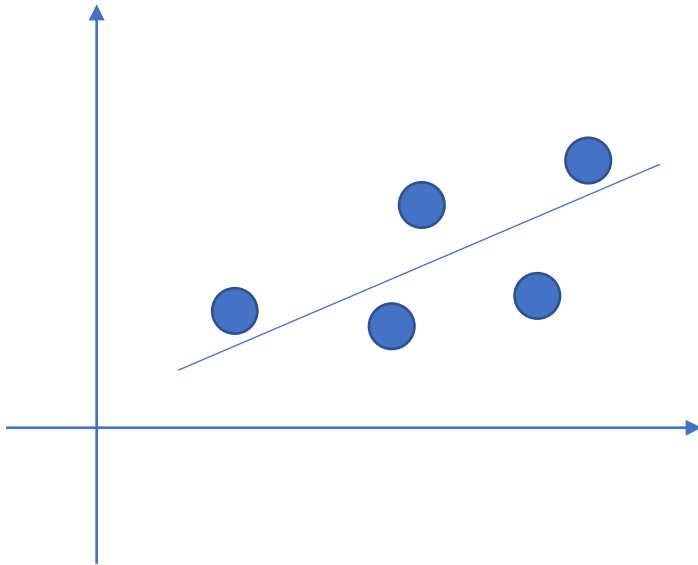
## II. Deep VS Shallow Learning

---

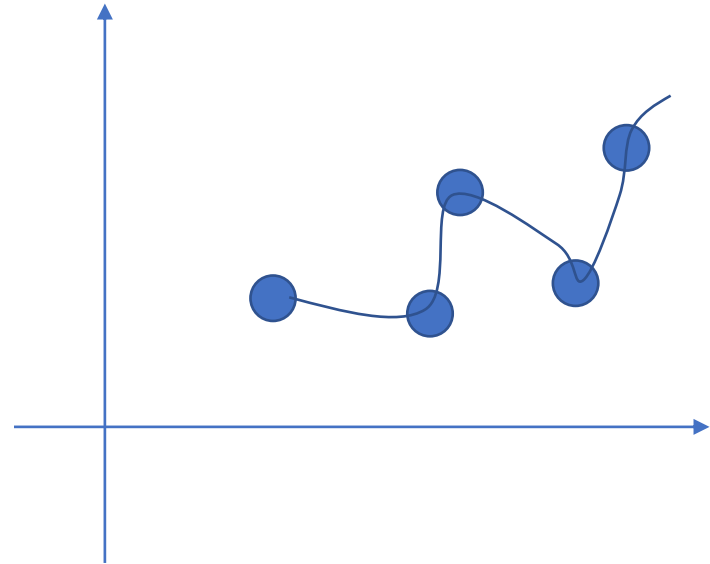
### Overfitting?

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

일반적인 Fitting



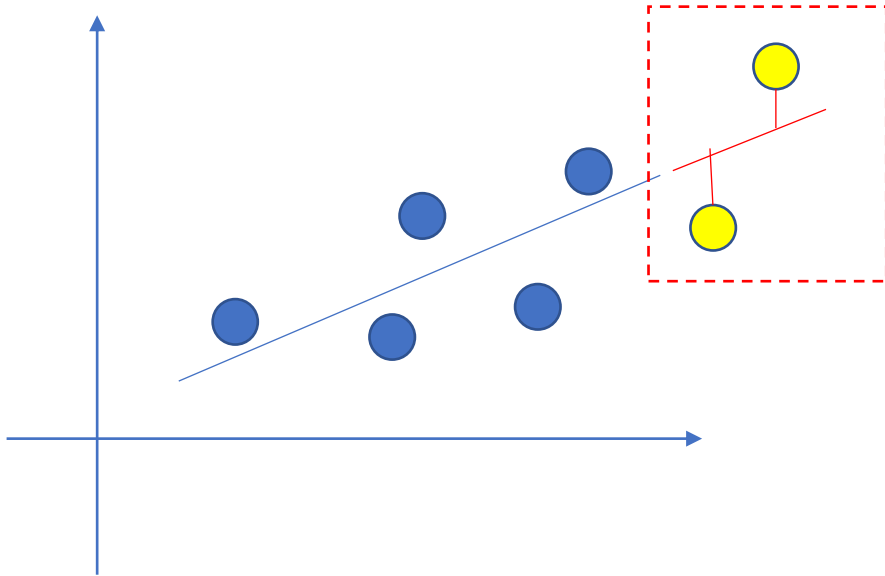
Over Fitting



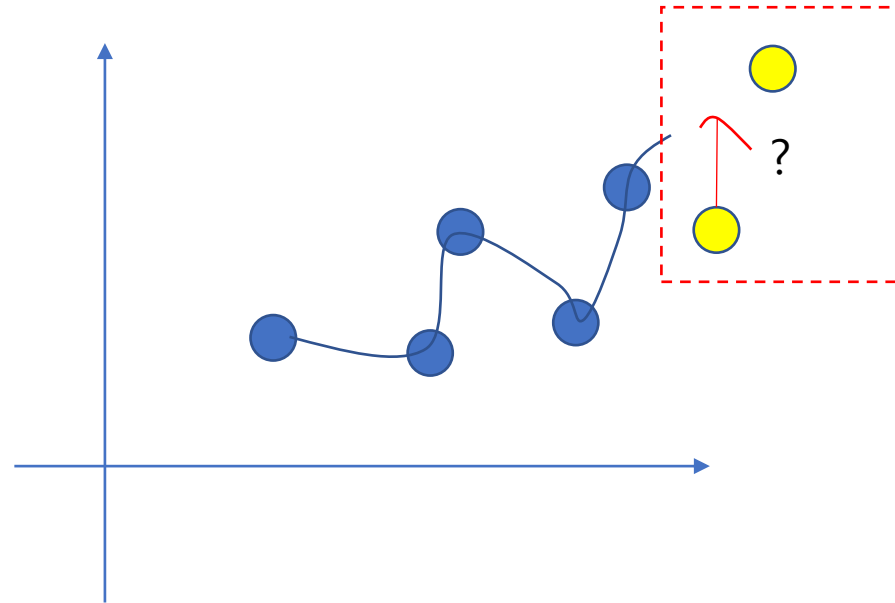
## II. Deep VS Shallow Learning

### Overfitting?

새로운 데이터에 대한 예측과 오차

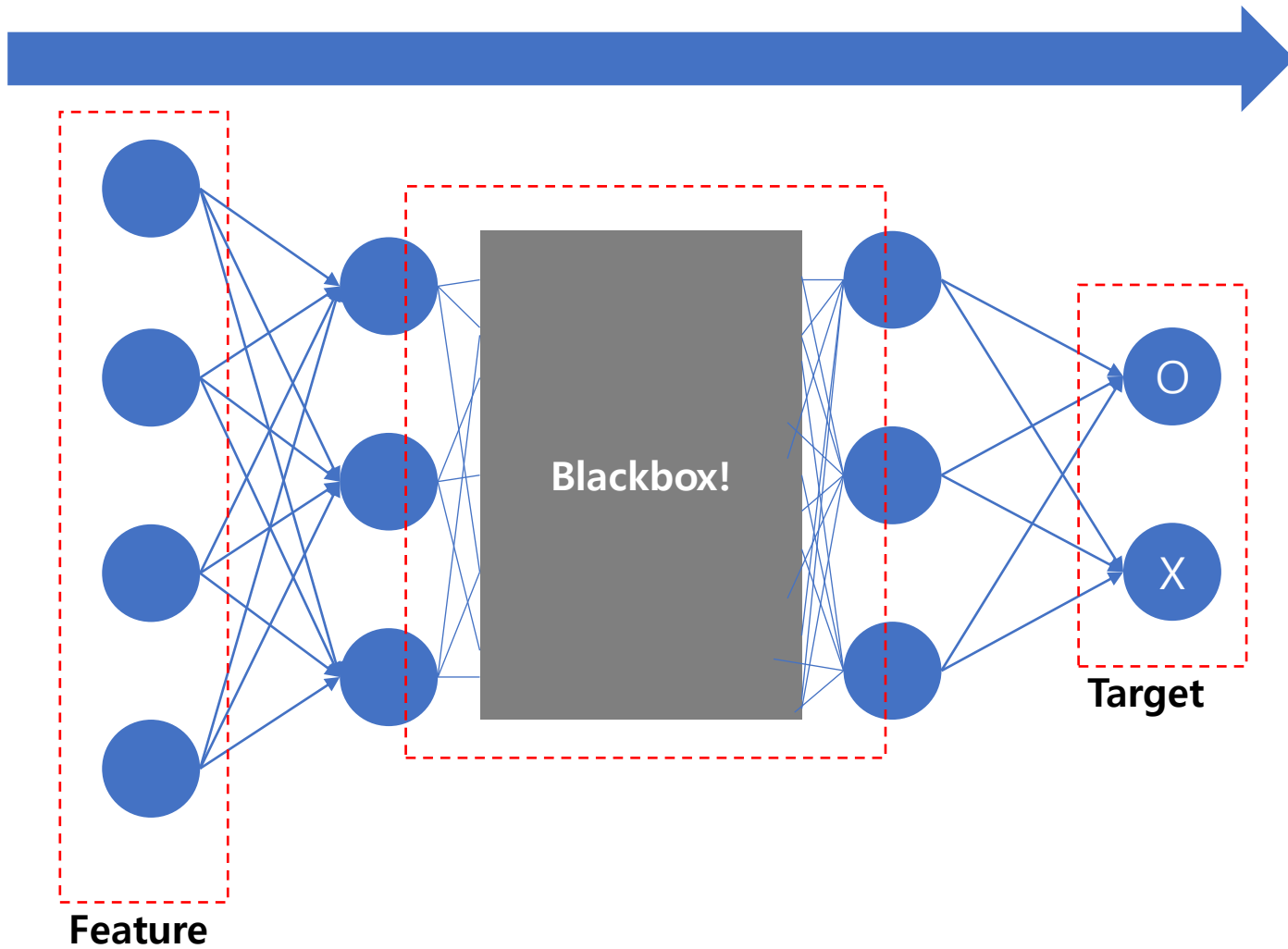


Over Fitting 시  
새로운 데이터에 대한 예측과 오차



## II. Deep VS Shallow Learning

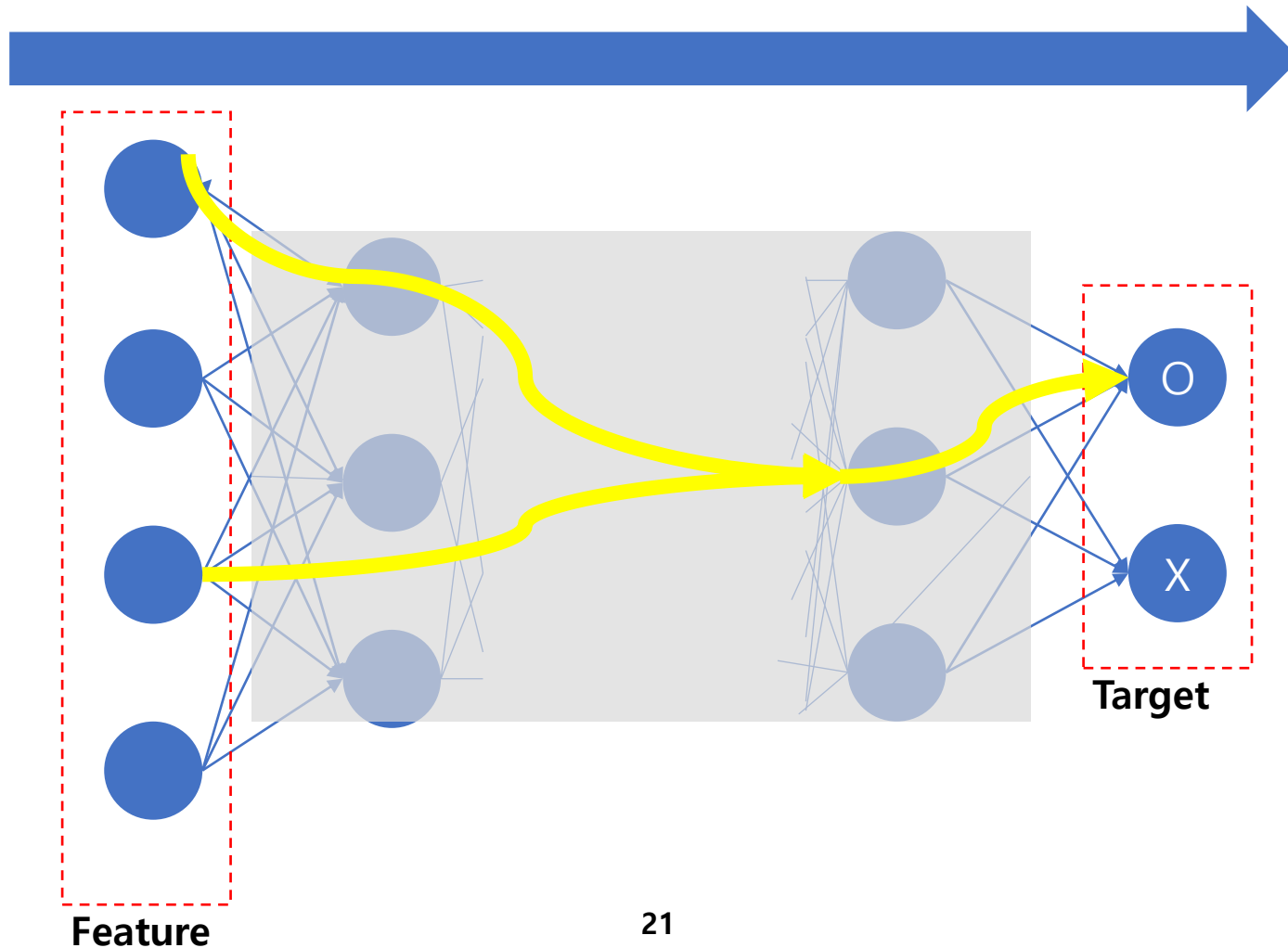
딥러닝=Blackbox model



## II. Deep VS Shallow Learning

### 설명 가능한 AI(eXplainable AI)

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



### III. 인공신경망 구성

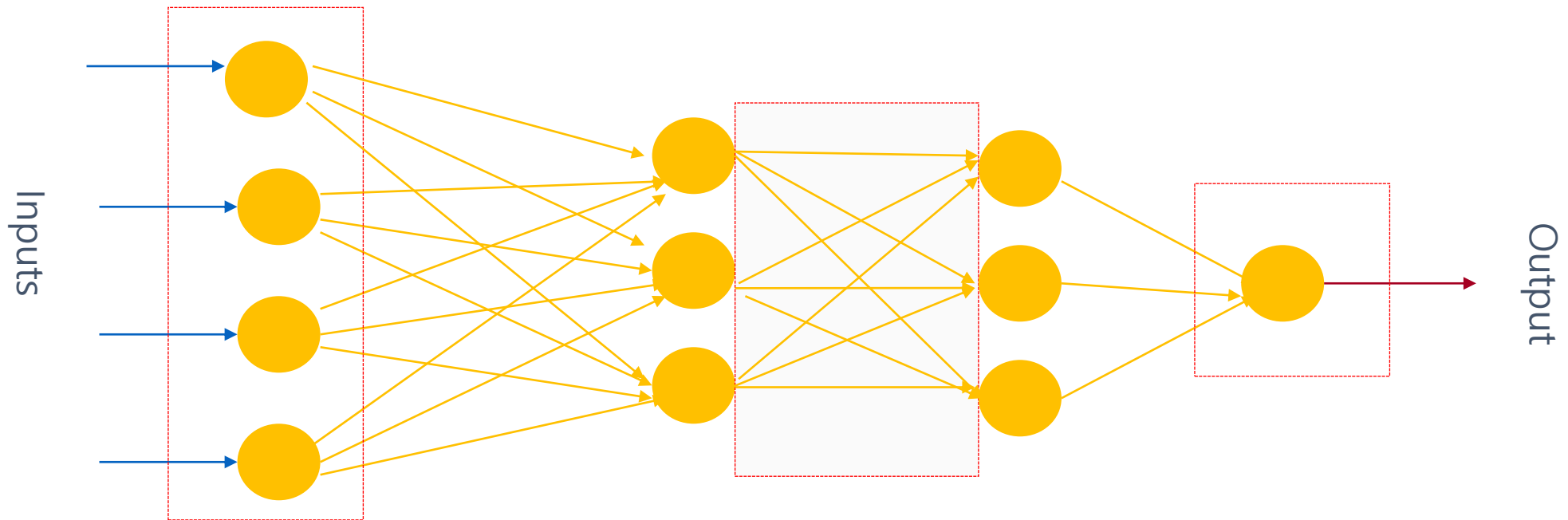
---

1. 퍼셉트론(Perceptron)이란: 1957년에 코넬 항공 연구소의 프랑크 로젠블라트에 의해 개발,  
단순한 형태의 피드포워드 네트워크이며 선형분류기
2. 퍼셉트론의 작동 원리: 입력값들의 가중합을 활성화함수를 거쳐 분류하도록 함
3. 퍼셉트론 한계 및 다층 퍼셉트론: 퍼셉트론은 비선형 분류에 한계가 있으며, 다층퍼셉트론으로 확장하면 비선형분류도 가능해짐.

### III. 인공신경망 구성

- **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

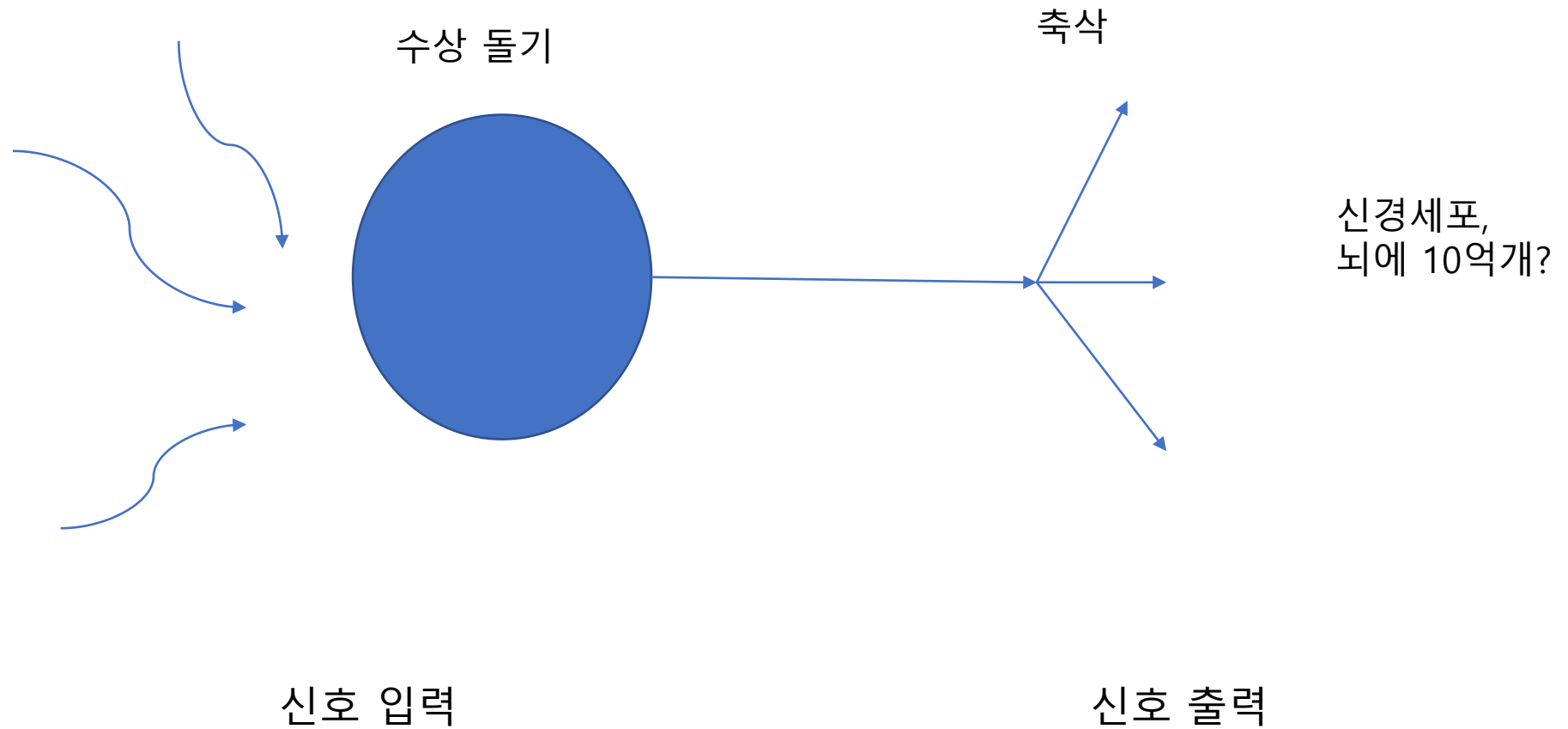


노드 간의 연결

### III. 인공신경망 구성

---

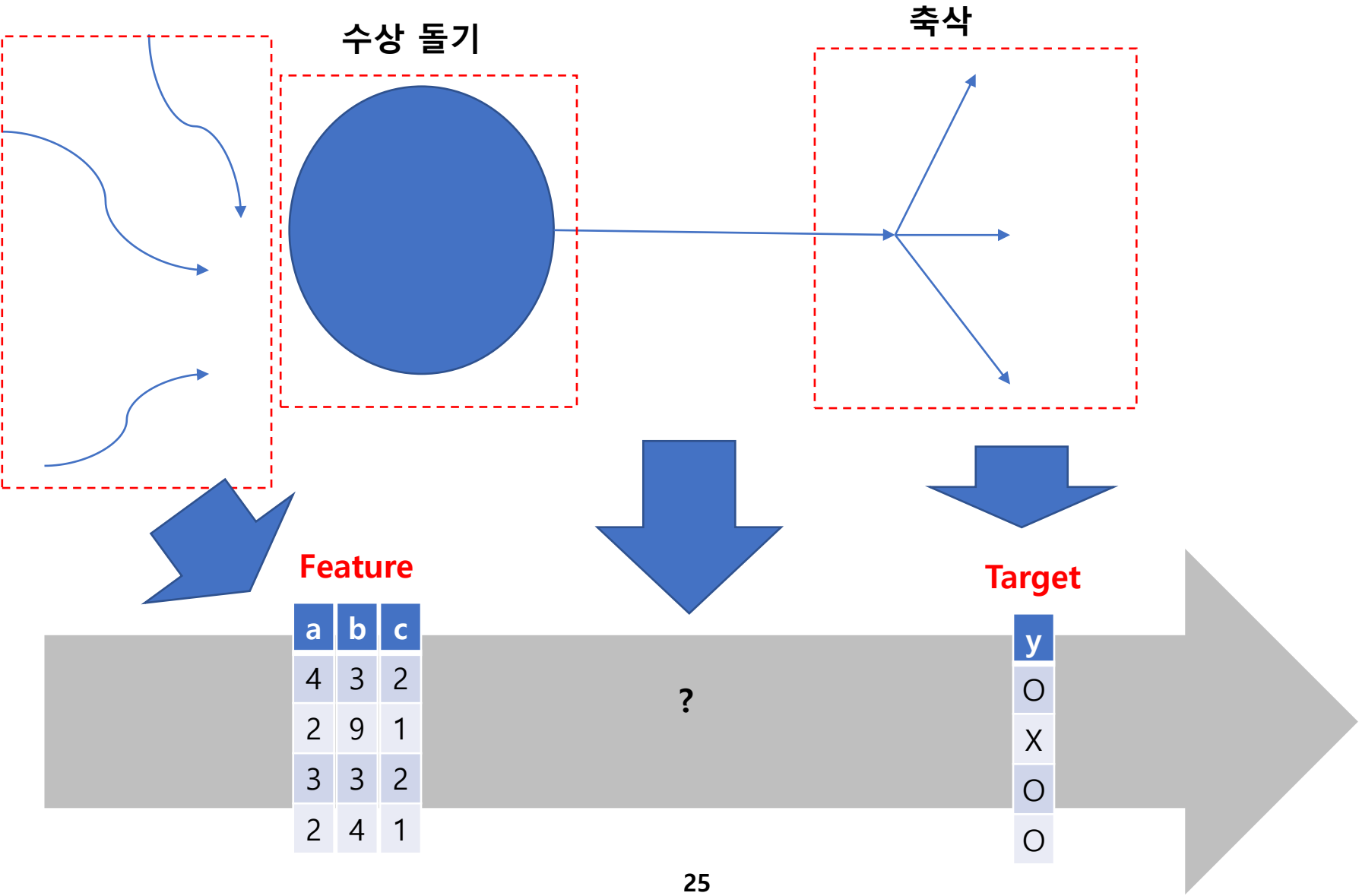
뉴런?





### III. 인공신경망 구성

#### 뉴런?



### III. 인공신경망 구성

---

#### 퍼셉트론

by

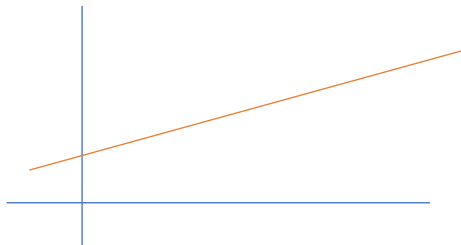
1957년

코넬 항공 연구소(Cornell Aeronautical Lab)

프랑크 로젠블라트 (Frank Rosenblatt)

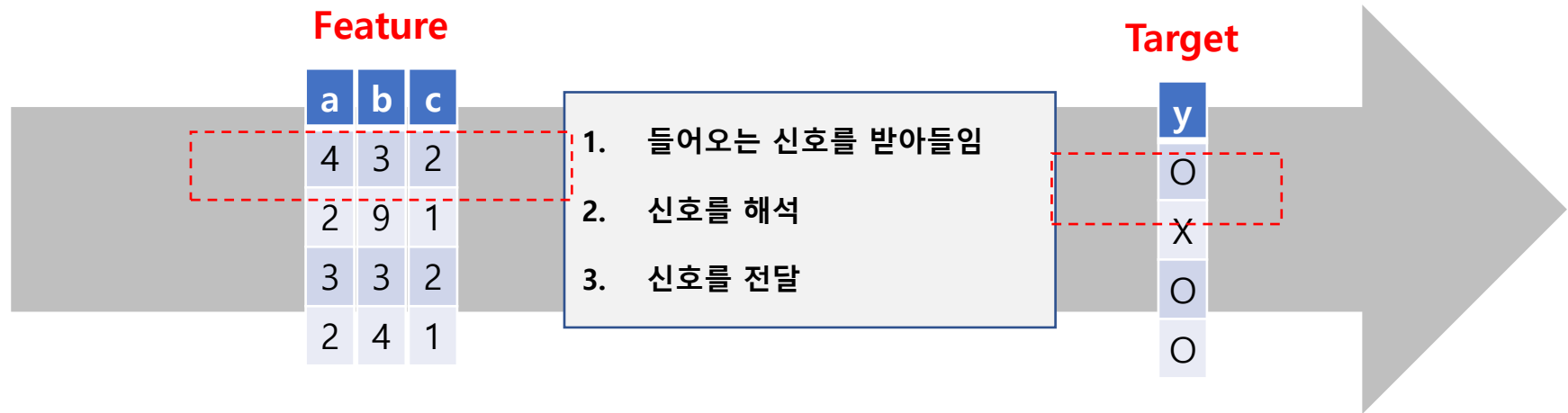


**간단한 형태의 선형분류기이자 인공신경망!**



### III. 인공신경망 구성

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

### III. 인공신경망 구성

#### 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

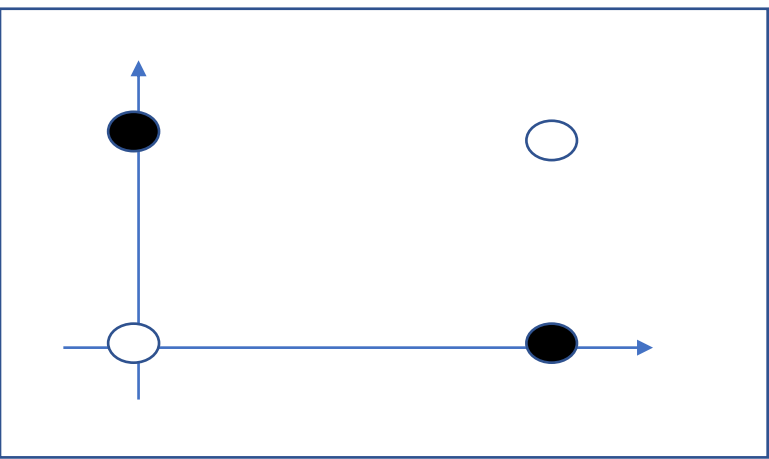
X1	X2	Y
0	0	0
0	1	1
1	0	1
1	1	0



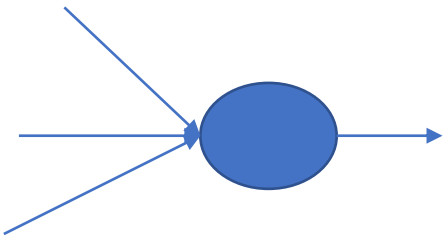
하나의 직선으로  
검은 원과 하얀원  
분류할 수 없음

### III. 인공신경망 구성

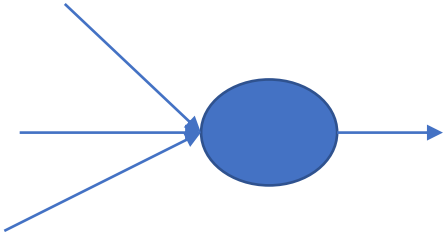
#### 다층 퍼셉트론: 비선형 분류 가능!



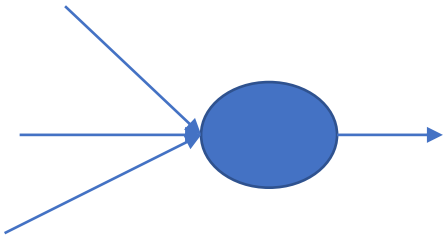
퍼셉트론



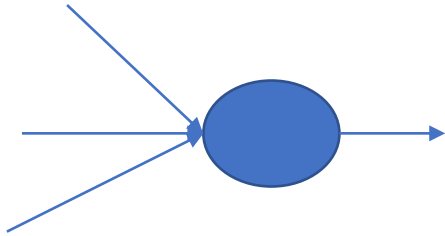
퍼셉트론



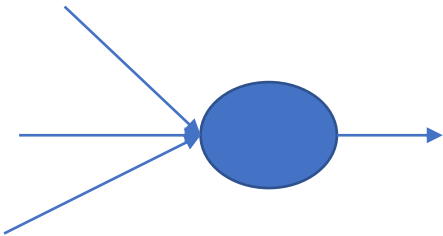
퍼셉트론



퍼셉트론



퍼셉트론



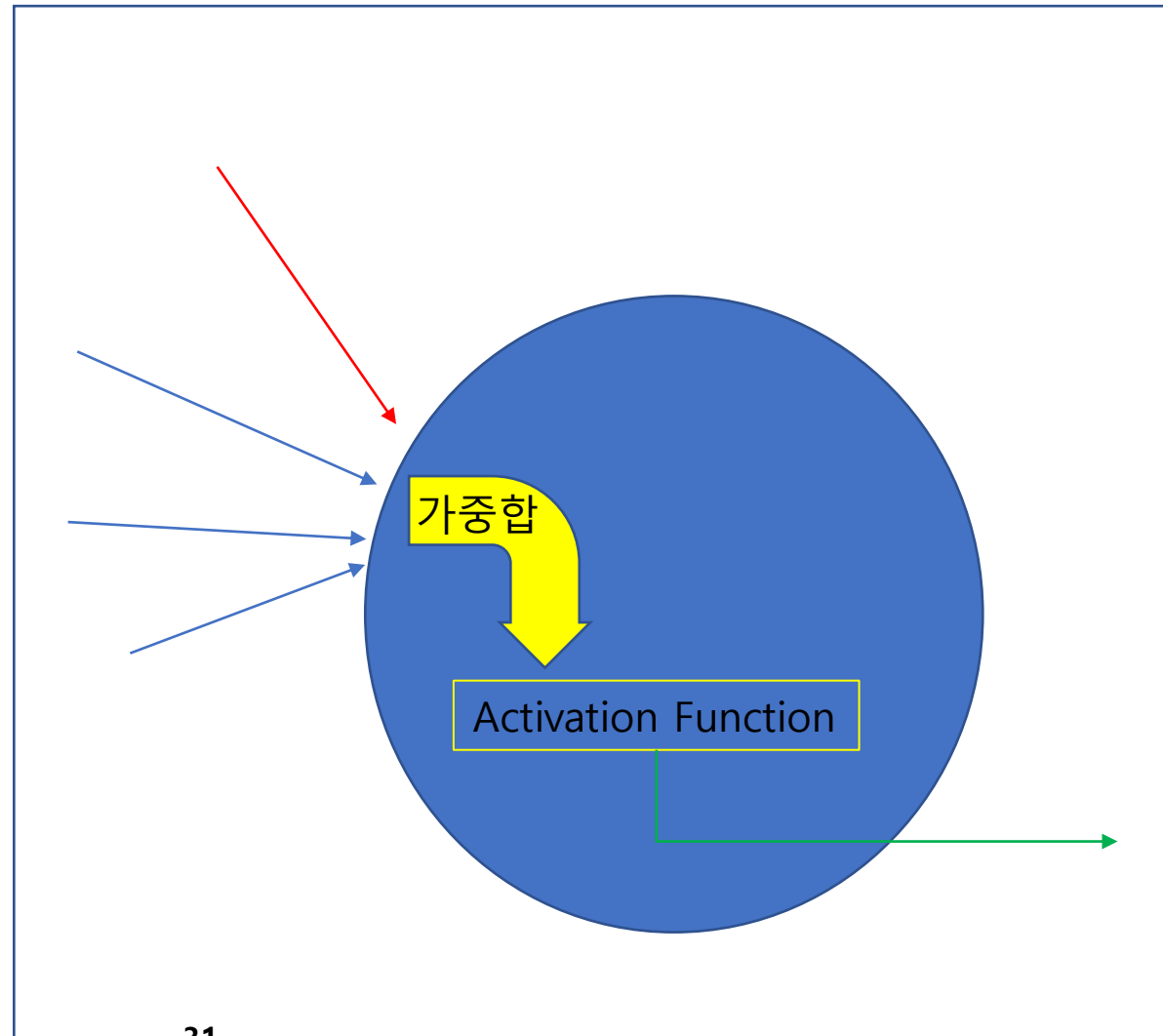
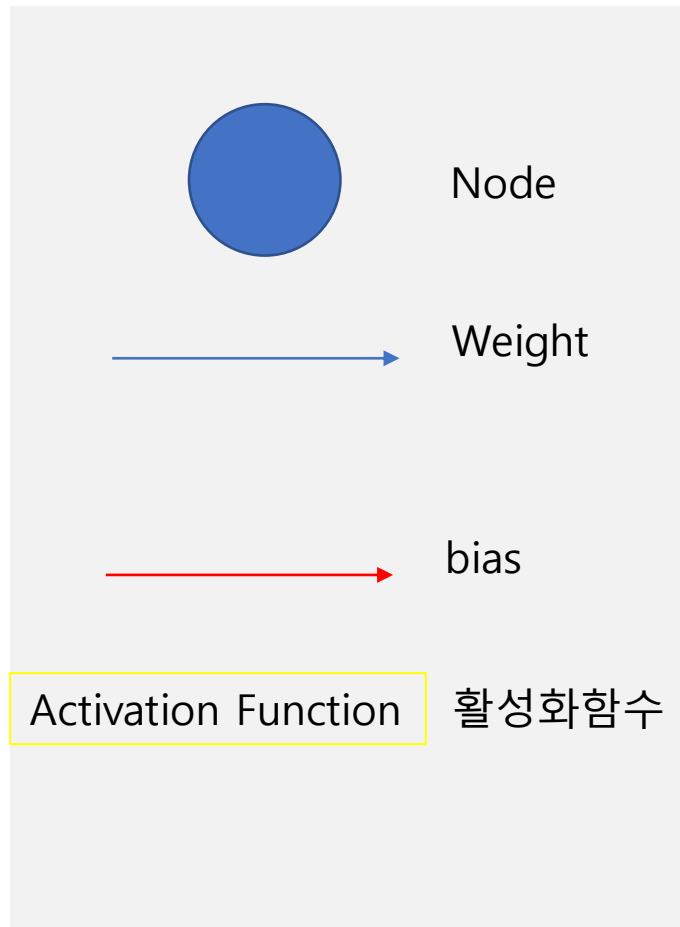
### III. 인공신경망 구성

---

인공신경망:  
뉴런의 작동방식을 모사한 기계학습 알고리즘

### III. 인공신경망 구성

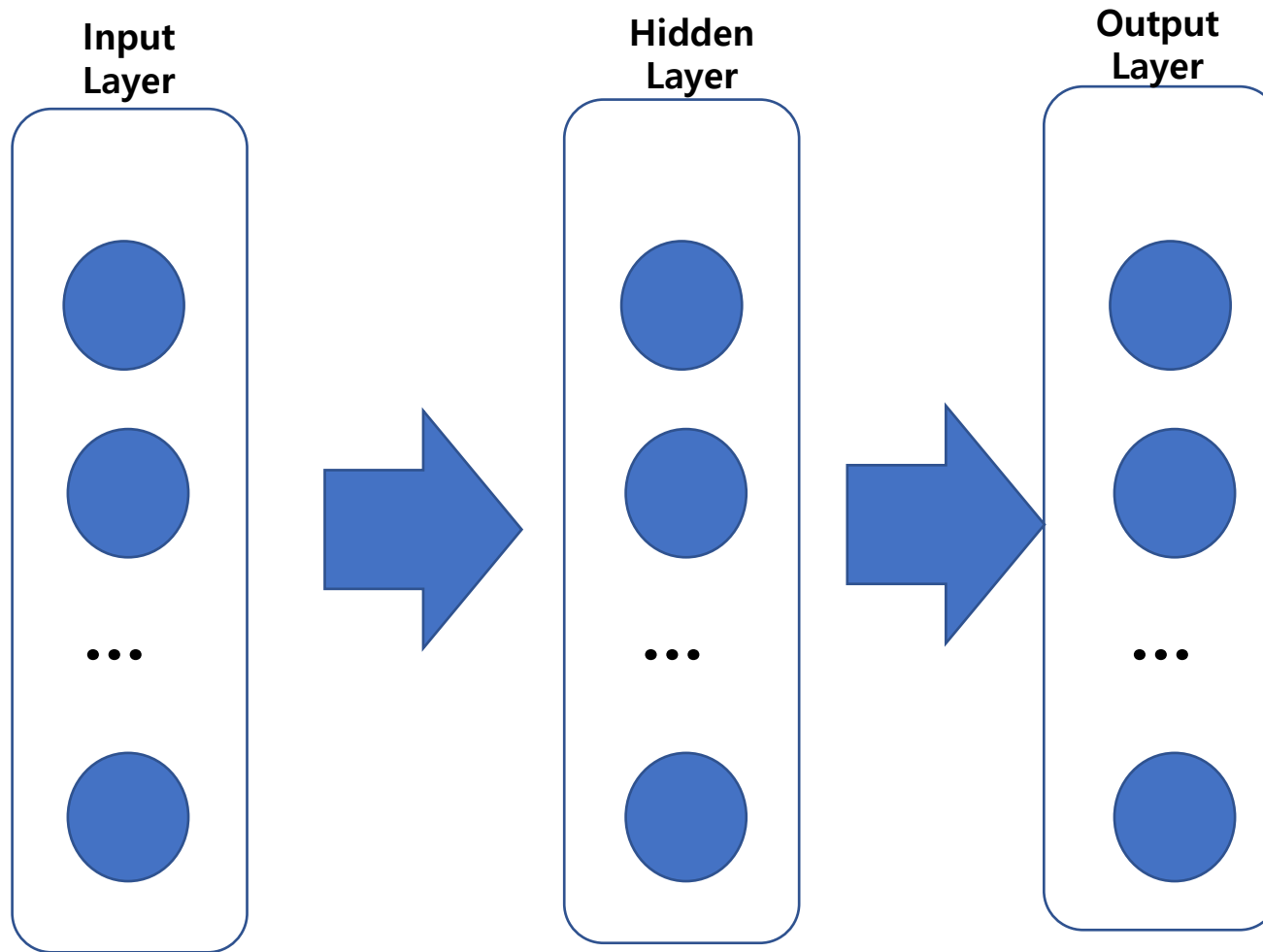
#### 인공신경망 구성 요소



### III. 인공신경망 구성

---

#### 인공신경망 구성 요소

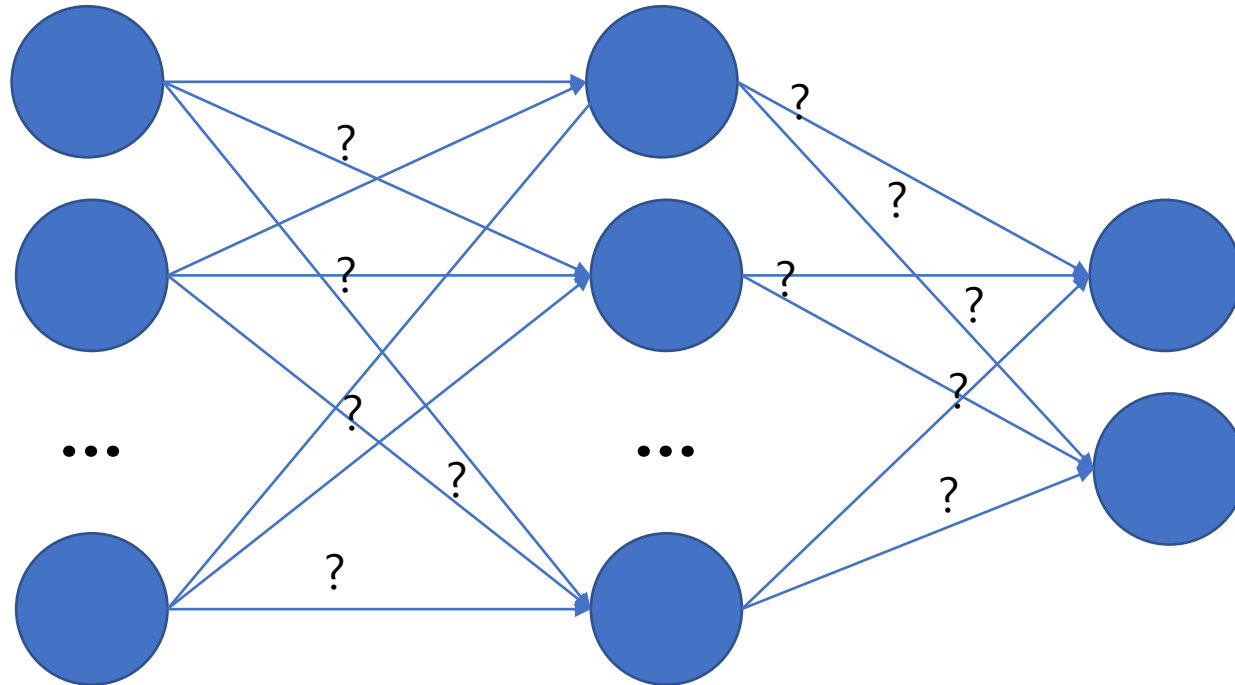




### III. 인공신경망 구성

---

#### 인공신경망: 가중치의 발견



## IV. 가중치와 행렬표현

---

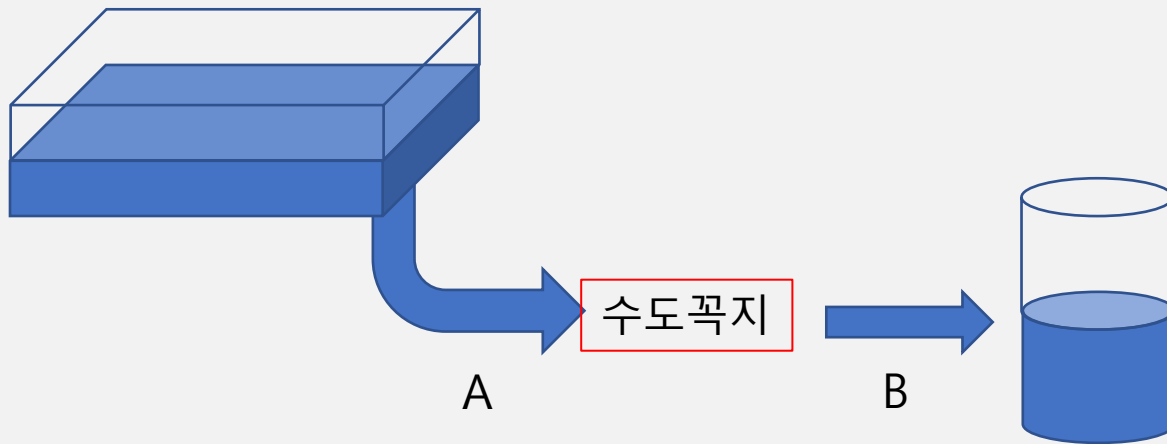
1. 가중치: 원하는 값으로 변환하기 위해 입력값에 곱해지는 수치, Weight라고 하며 값을 강조할 경우 가중치를 증가시킬 수 있음.
2. 인공신경망과 가중치: 인공신경망의 각 층의 노드를 연결하는 것이 바로 가중치이며, 이전 층의 노드값의 가중합이 다음 층의 노드의 입력값이 됨.
3. 인공신경망의 가중치 조절: 오차를 줄이거나 분류를 잘하는 것과 같이 원하는 목적을 달성하기 위해 가중치를 조절할 수 있음.

## IV. 가중치와 행렬표현

### 가중치는?

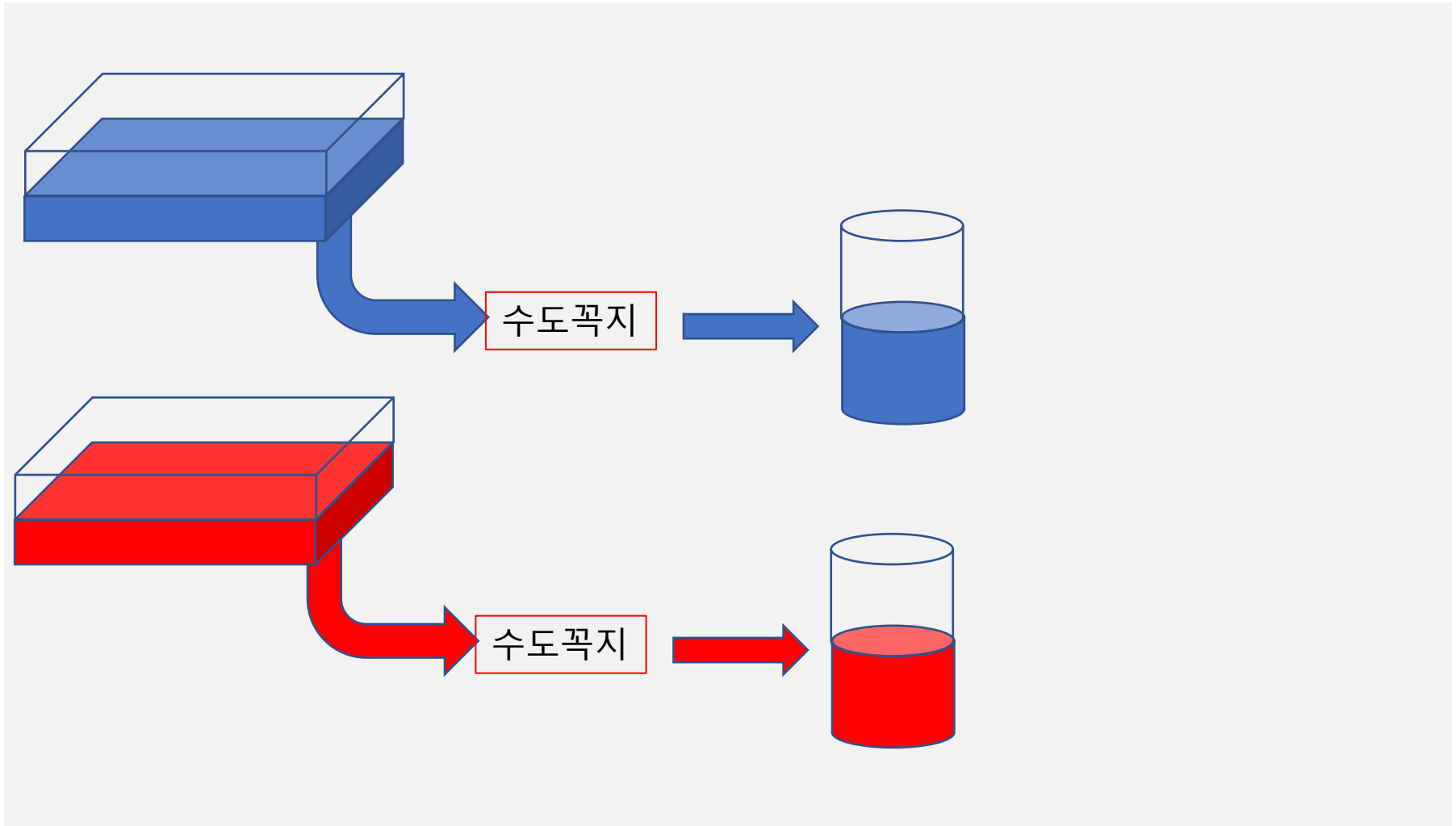
Weight, 가중치

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



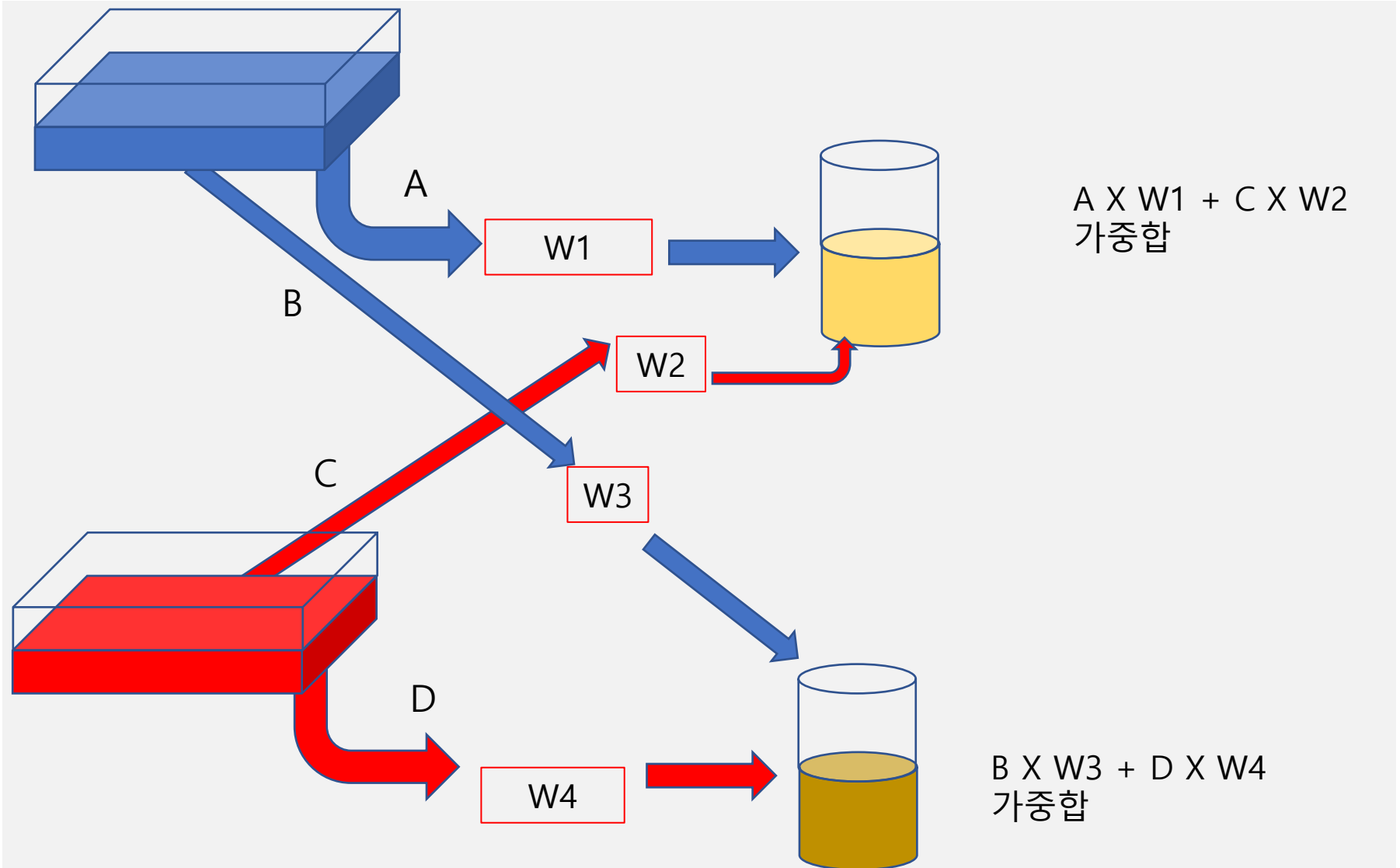
## IV. 가중치와 행렬표현

### 가중치는?



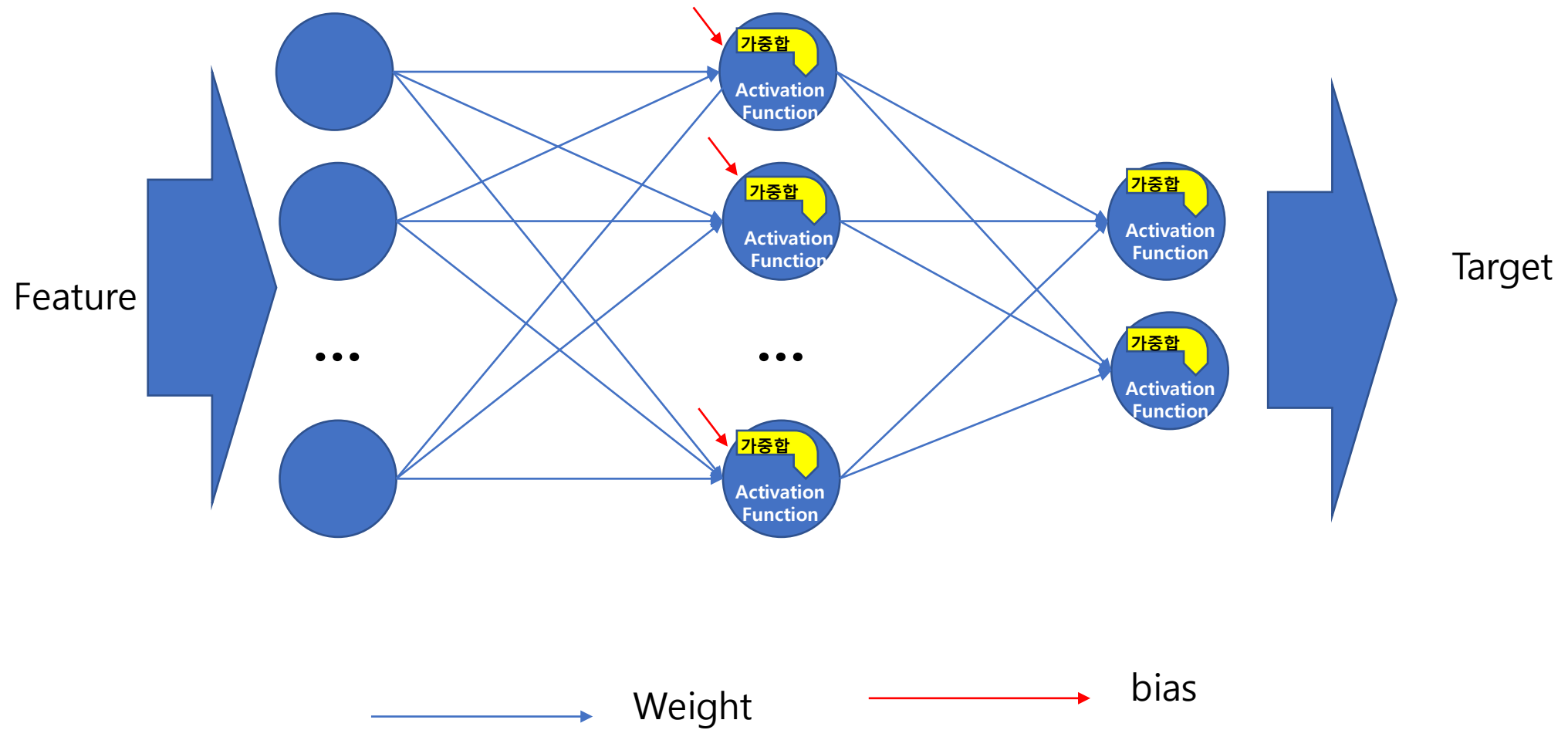
## IV. 가중치와 행렬표현

### 가중치는?



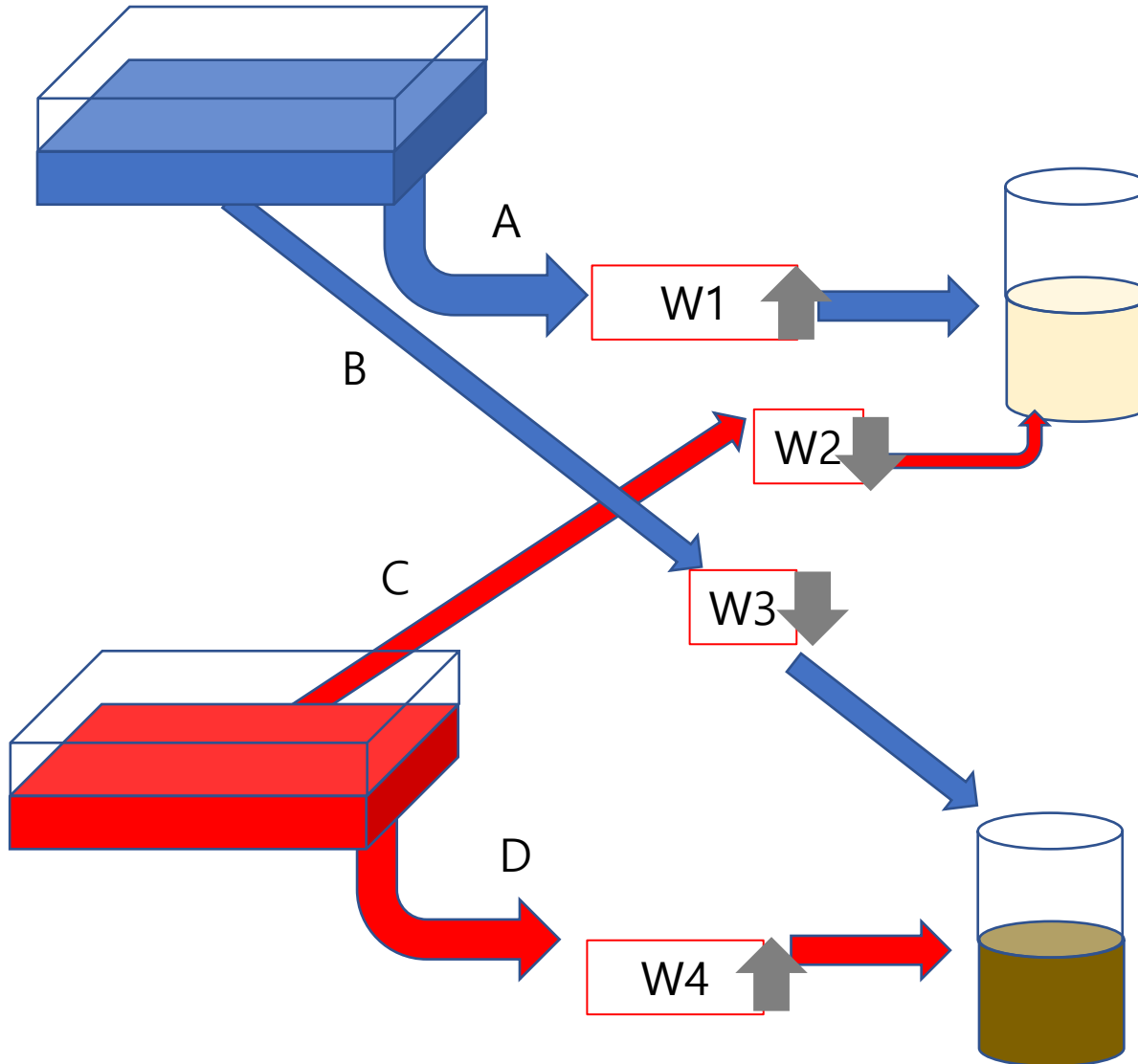
## IV. 가중치와 행렬표현

### 인공신경망과 가중치



## IV. 가중치와 행렬표현

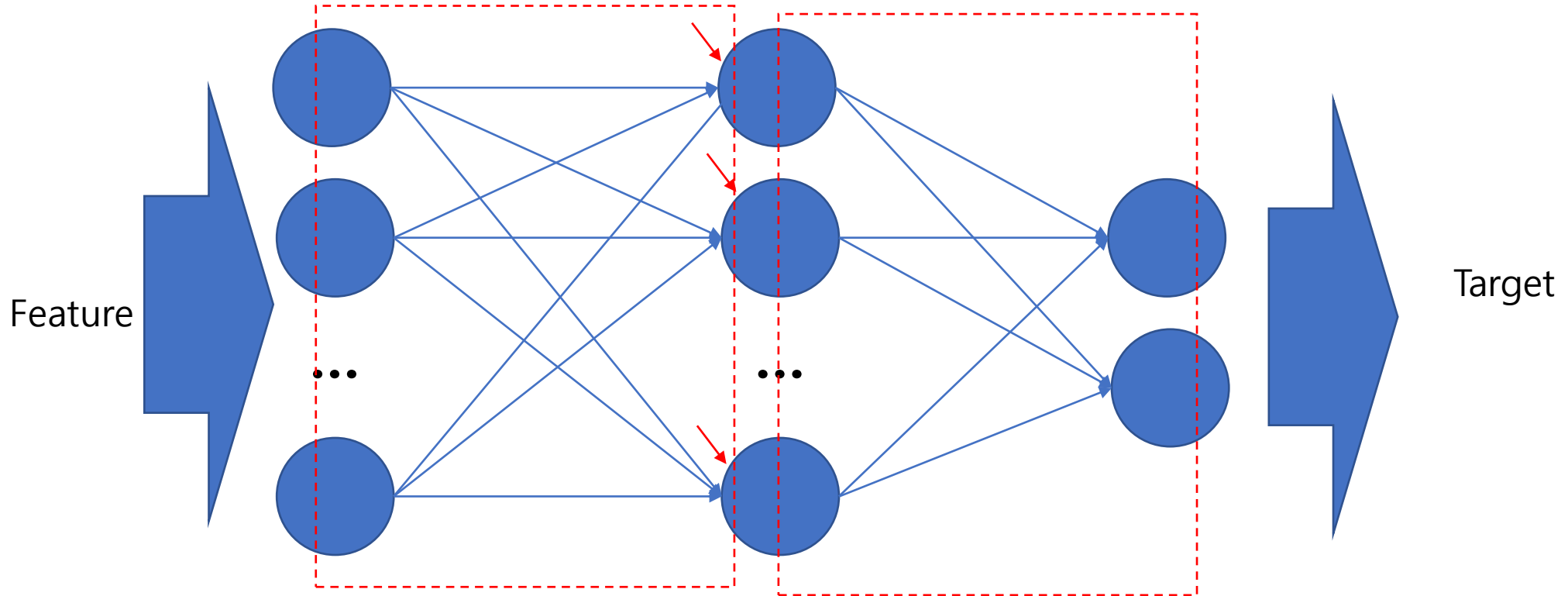
### 가중치 조정!



원하는 색상이 아닌 경우,  
"수도꼭지"를 조절!

## IV. 가중치와 행렬표현

### 인공신경망과 가중치



*원하는 결과를 얻도록 가중치를 +-로 조정!*

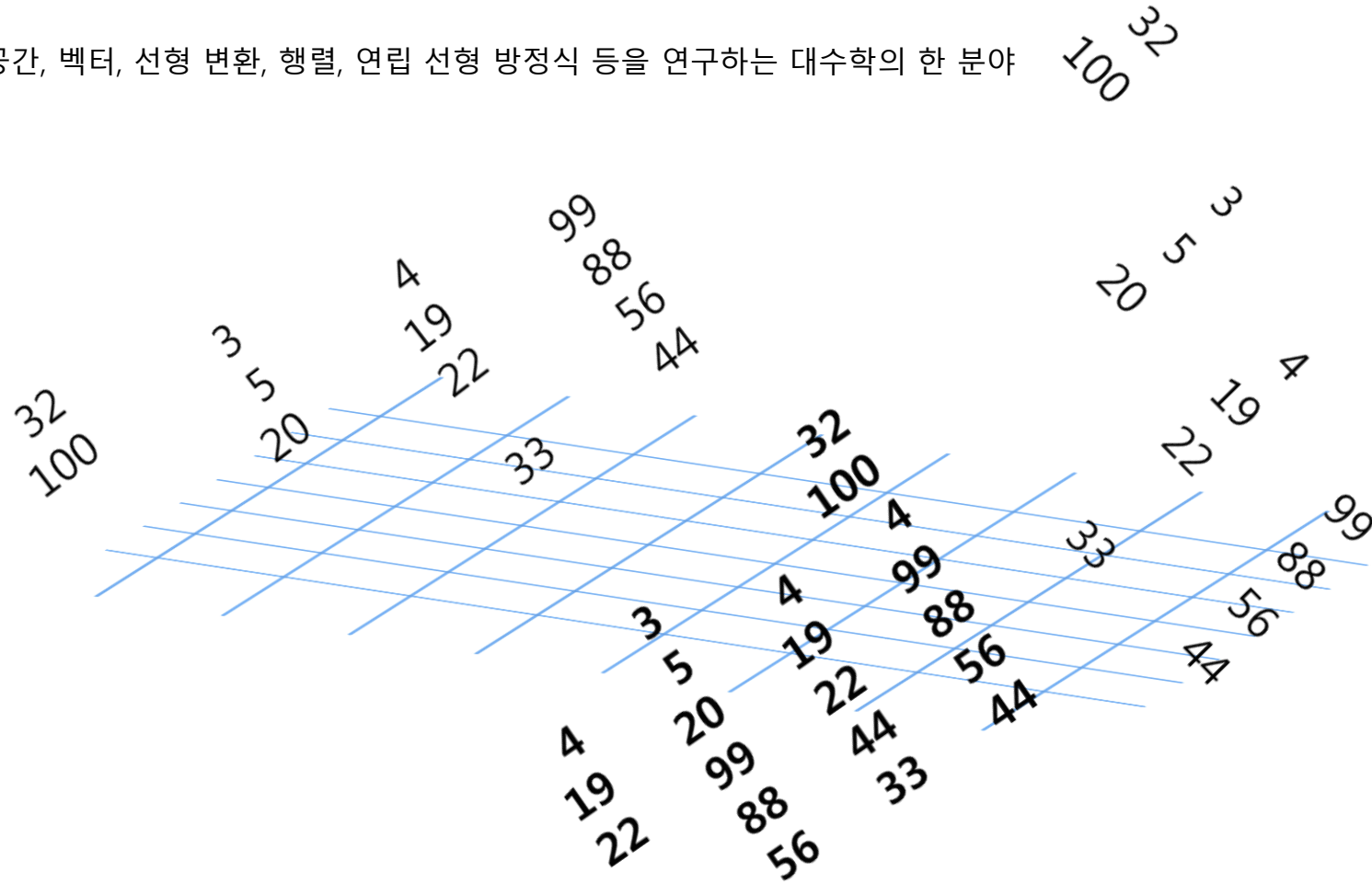
가중치와 관련하여 오늘 사용한 계산: 더하기, 곱하기, 크기 늘이기, 크기 줄이기



## IV. 가중치와 행렬표현

### 선형대수와 “행렬”

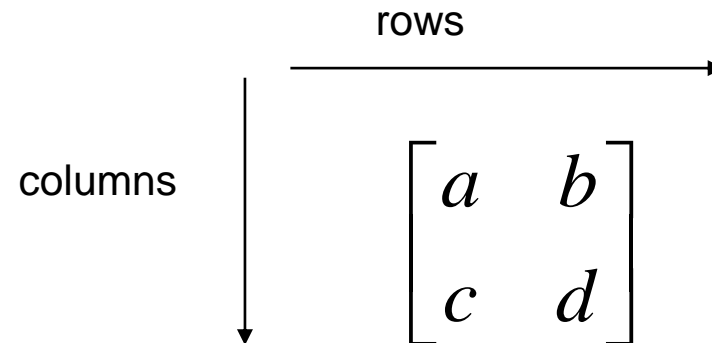
벡터 공간, 벡터, 선형 변환, 행렬, 연립 선형 방정식 등을 연구하는 대수학의 한 분야



## IV. 가중치와 행렬표현

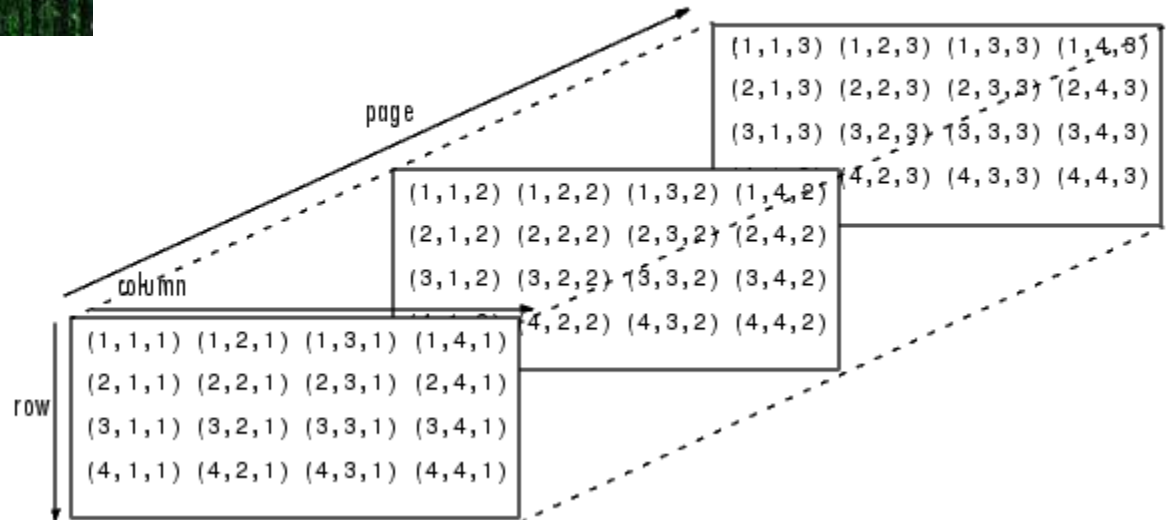
행렬이란: 어떤 값을 행과 열을 갖는 2차원 형태로 배열한 것

- 행렬
  - A matrix is a set of elements, organized into rows and columns
  - 대문자로 표현하며, 각 원소는  $A_{ij}$ 로 표현



## IV. 가중치와 행렬표현

### 인공신경망을 행렬로 표현하기!



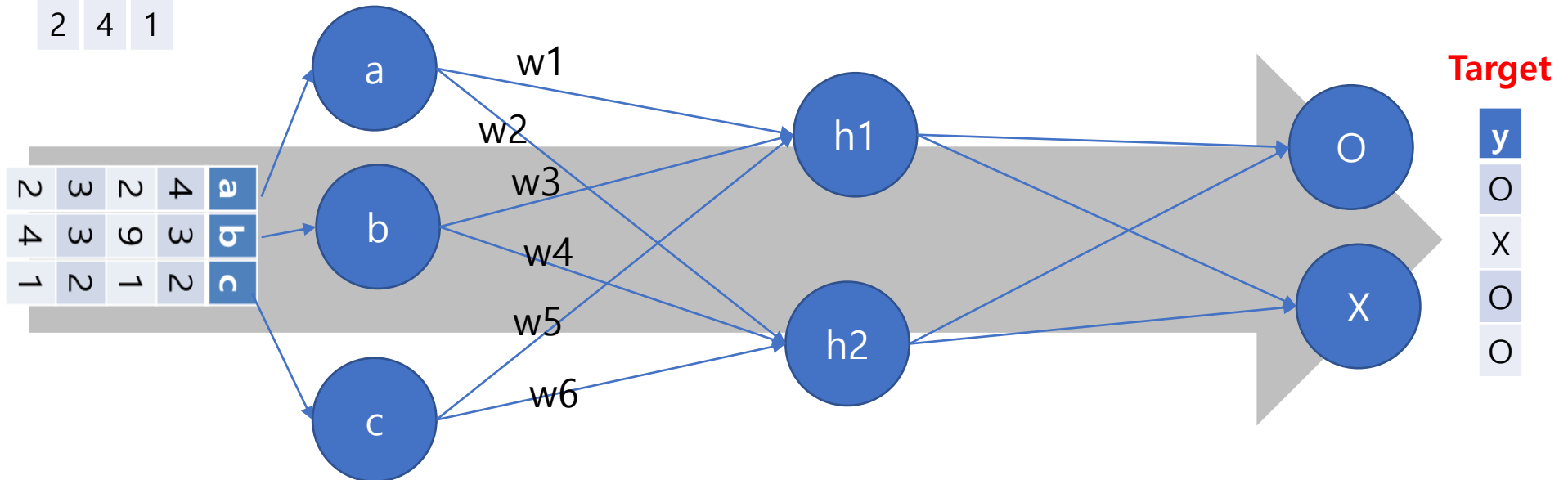
## IV. 가중치와 행렬표현

### 인공신경망을 행렬로 표현하기!

#### Feature

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

1. 입력데이터
2. 가중합
3. 은닉층 표현



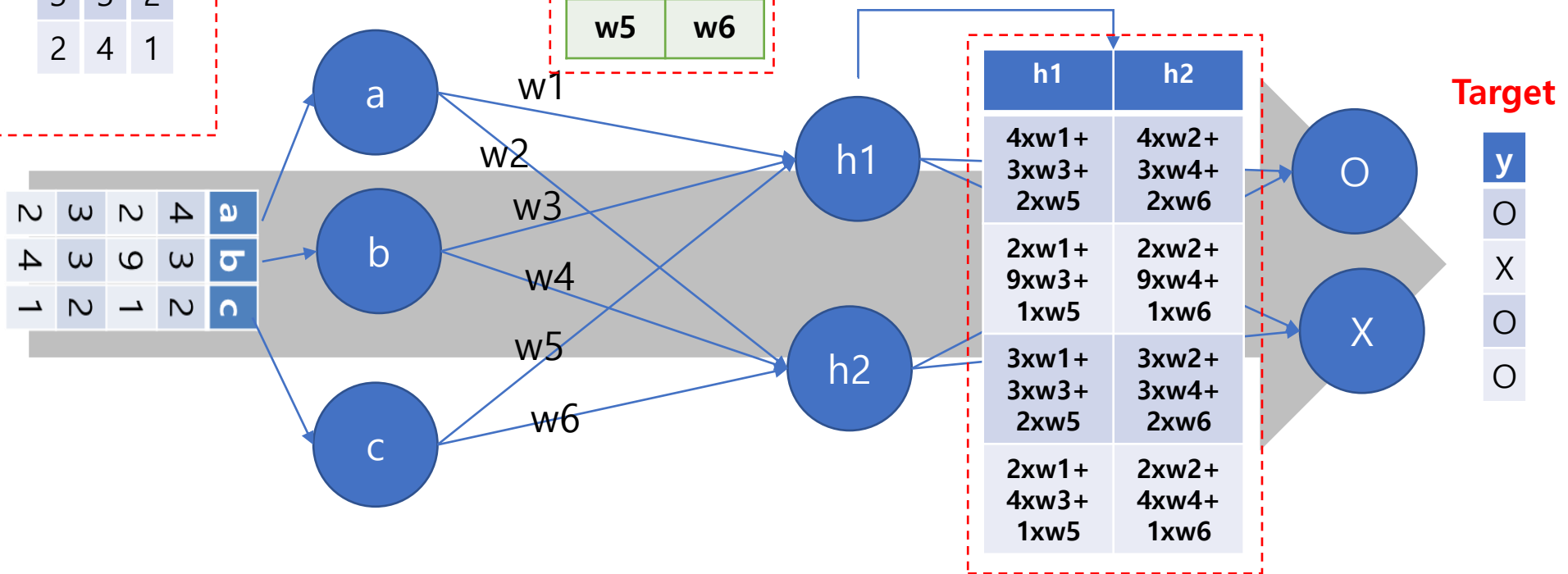
## IV. 가중치와 행렬표현

### 인공신경망을 행렬로 표현하기!

#### Feature

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

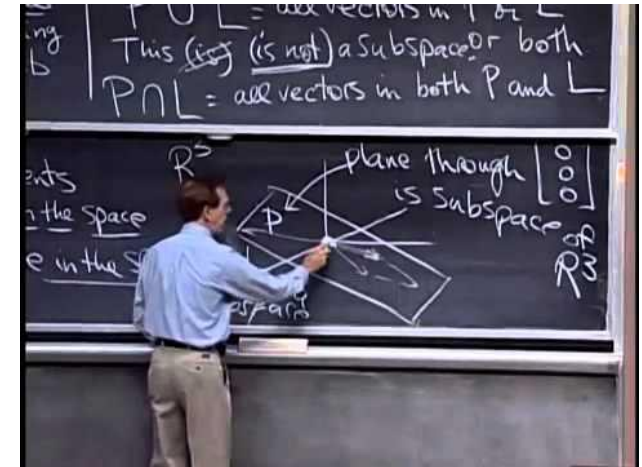
w1	w2
w3	w4
w5	w6



## IV. 가중치와 행렬표현

### 행렬로 할 수 있는 것들!

- 데이터 표현
  - 많은 데이터가 **행렬**을 통해 자연스럽게 표현
  - 정형 데이터, 비정형 데이터
- 데이터 변환
  - 다른 벡터 공간을 사용한 데이터 표현
  - 좌표계 변환
  - 차원변환 : 차원축소
- 데이터처리
  - 특징추출 - 행렬 분해
  - 수학적 기술의 편의성
  - 명확하고 간결한 표현



*Linear Algebra has become as basic and as applicable  
as calculus, and fortunately it is easier.*

**Gilbert Strang, MIT**

## IV. 가중치와 행렬표현

### • 행렬 계산!

- 합/차/곱

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} a+e & b+f \\ c+g & d+h \end{bmatrix}$$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} - \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} a-e & b-f \\ c-g & d-h \end{bmatrix}$$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} ae+bg & af+bh \\ ce+dg & cf+dh \end{bmatrix}$$

## IV. 가중치와 행렬표현

### • 행렬의 곱

- 두 행렬의 곱은 각 행/열의 곱의 합으로 계산

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

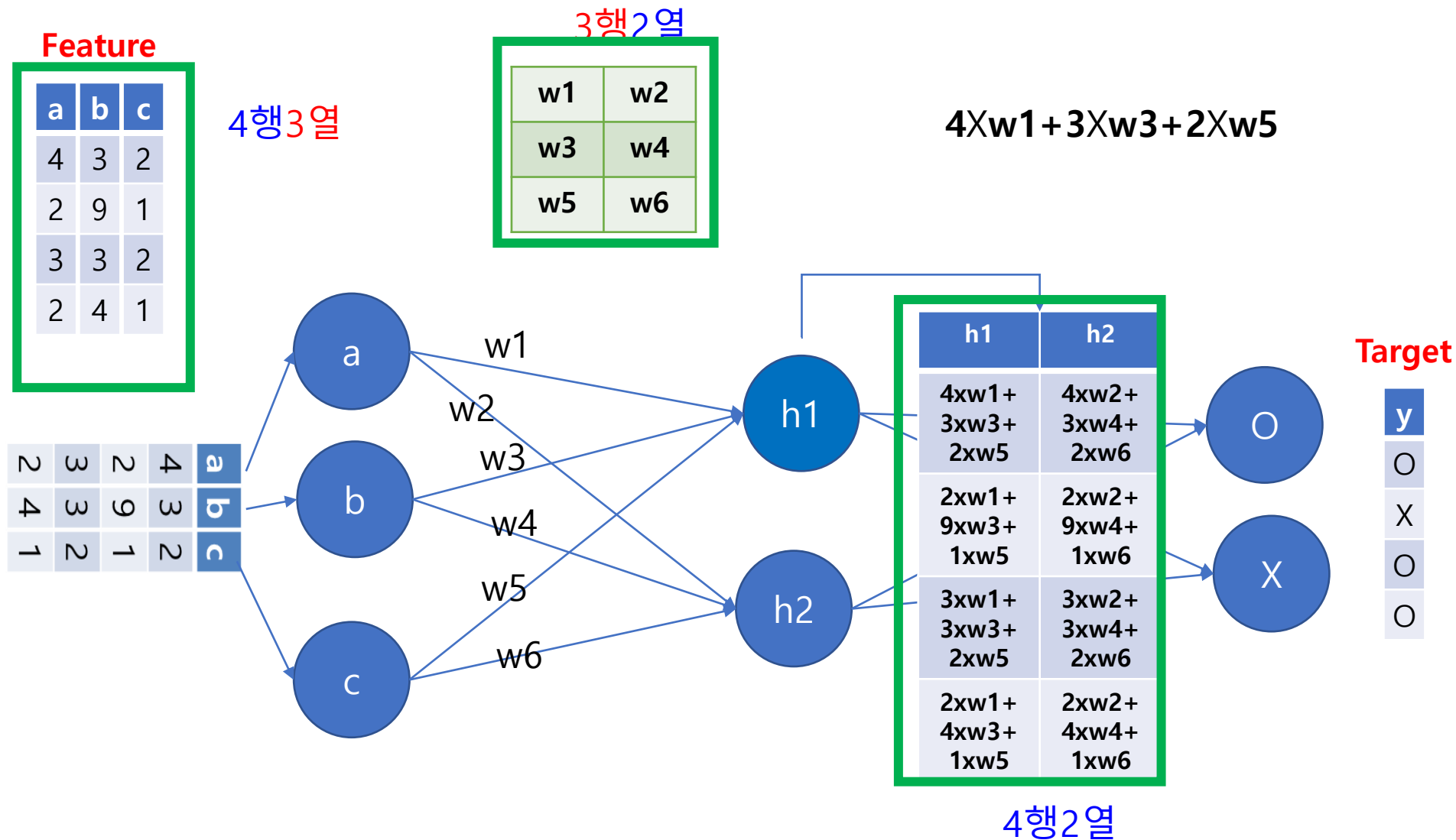
$\begin{pmatrix} 1 & 0 \\ 2 & 3 \end{pmatrix} \times \begin{pmatrix} 2 & 1 \\ 3 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 2 & 3 \end{pmatrix} \times \begin{pmatrix} 2 & 1 \\ 3 & 1 \end{pmatrix}$
$\begin{pmatrix} 1 & 0 \\ 2 & 3 \end{pmatrix} \times \begin{pmatrix} 2 & 1 \\ 3 & 1 \end{pmatrix}$	$\begin{pmatrix}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

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



# IV. 가중치와 행렬표현

## 인공신경망의 가중합



# IV. 가중치와 행렬표현

## 인공신경망의 가중합

Feature

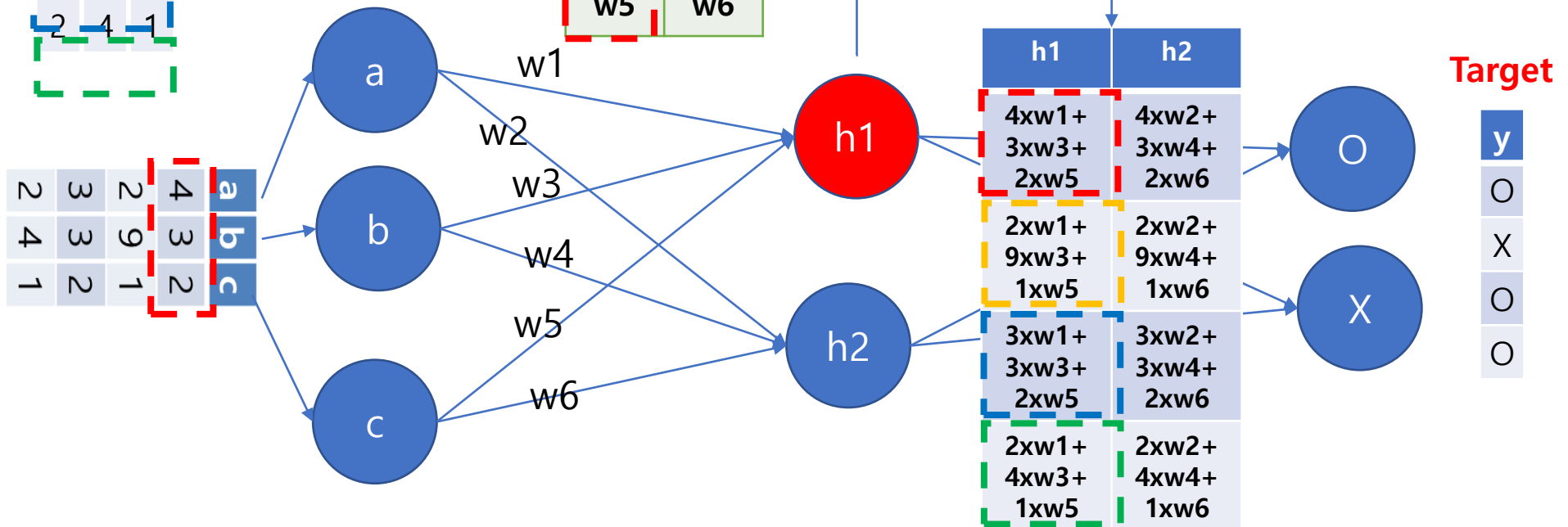
a	b	c
4	3	2
2	9	1
3	3	2
2	4	1

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

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

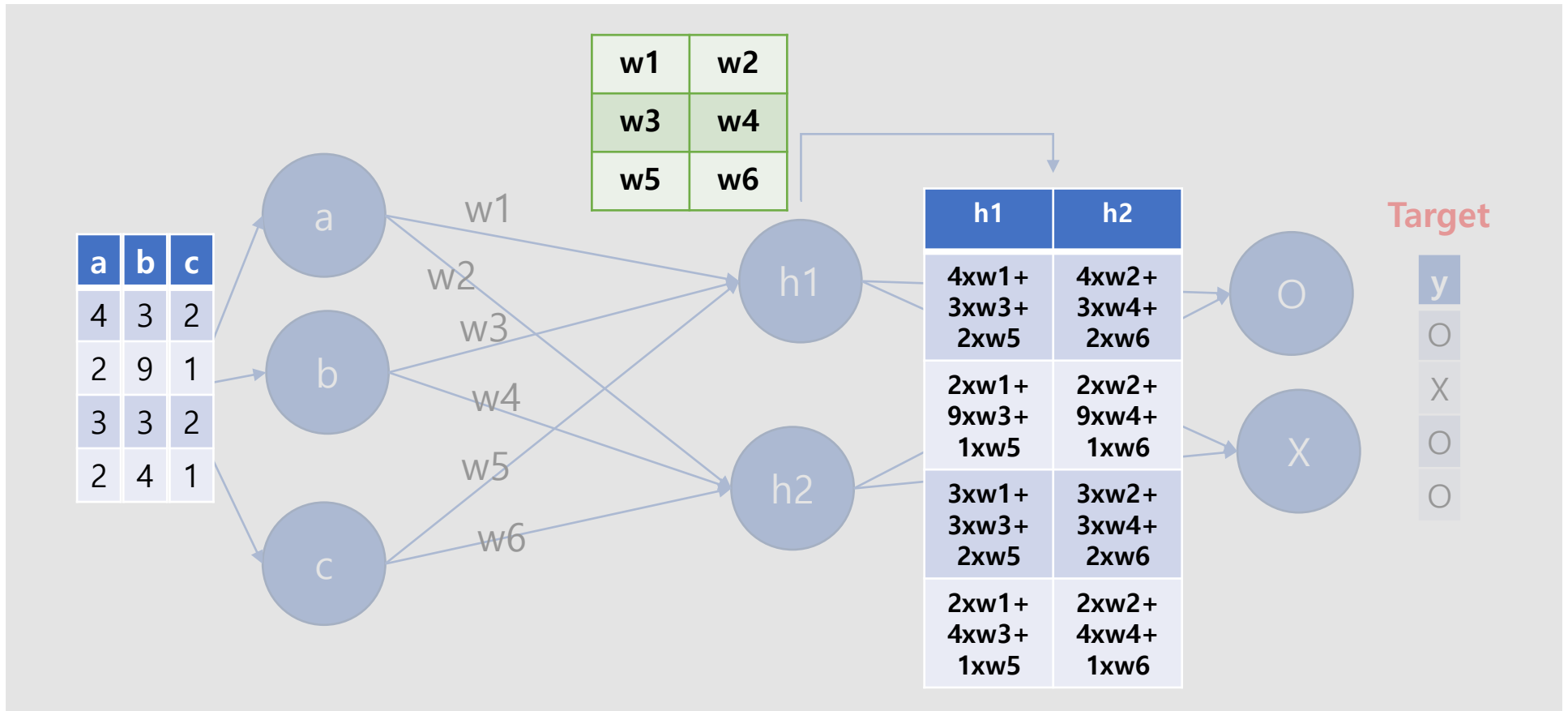
$$4 \times w_1 + 3 \times w_3 + 2 \times w_5$$

w1	w2
w3	w4
w5	w6



## IV. 가중치와 행렬표현

### 인공신경망의 행렬곱



# IV. 가중치와 행렬표현

## 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

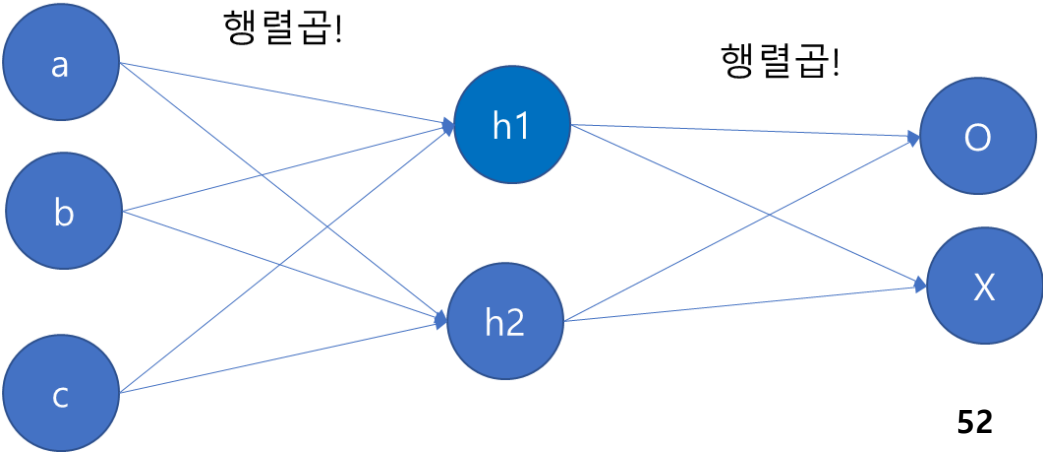
3행2열

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



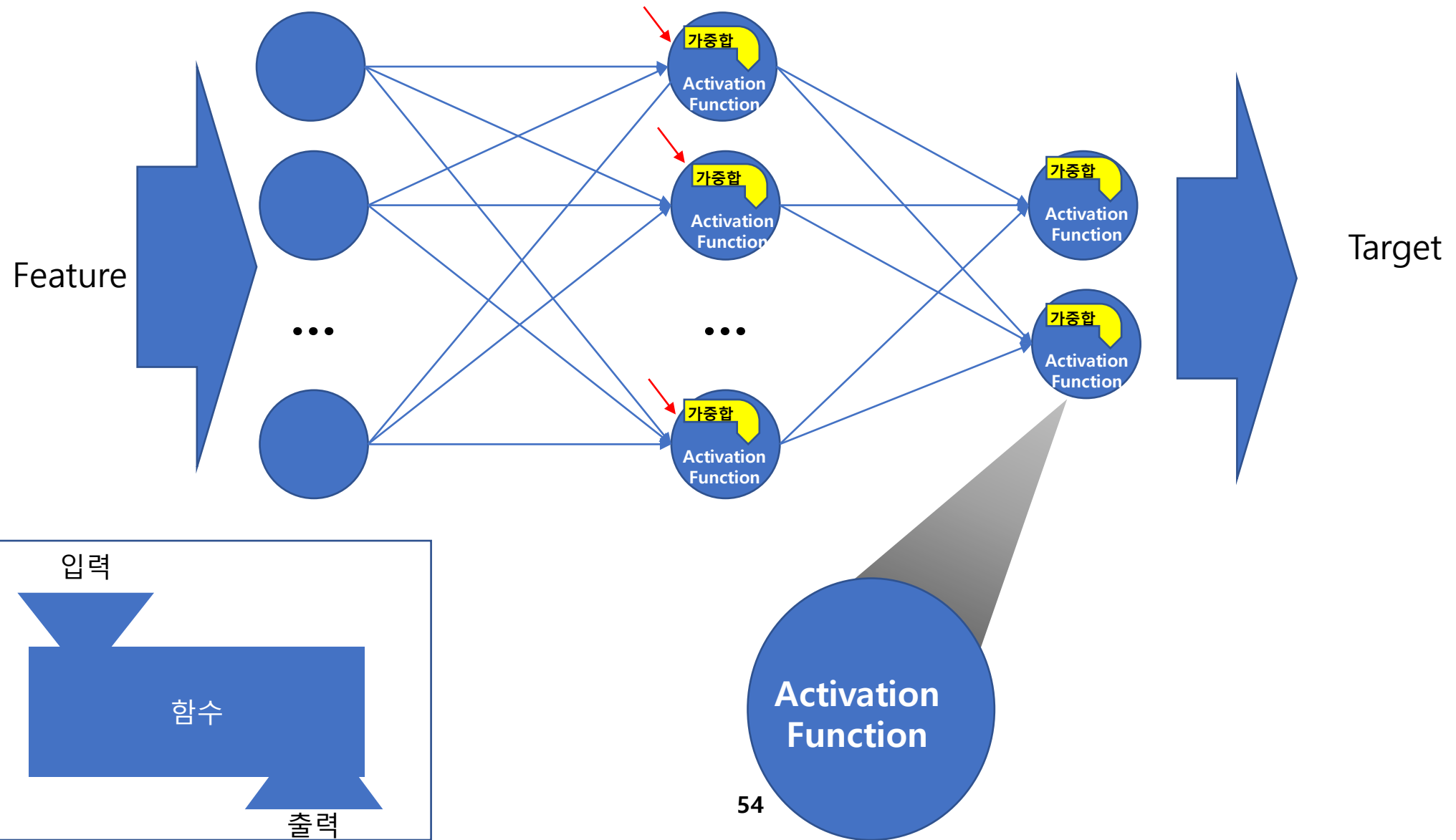
## V. 활성화함수와 가중치 업데이트

---

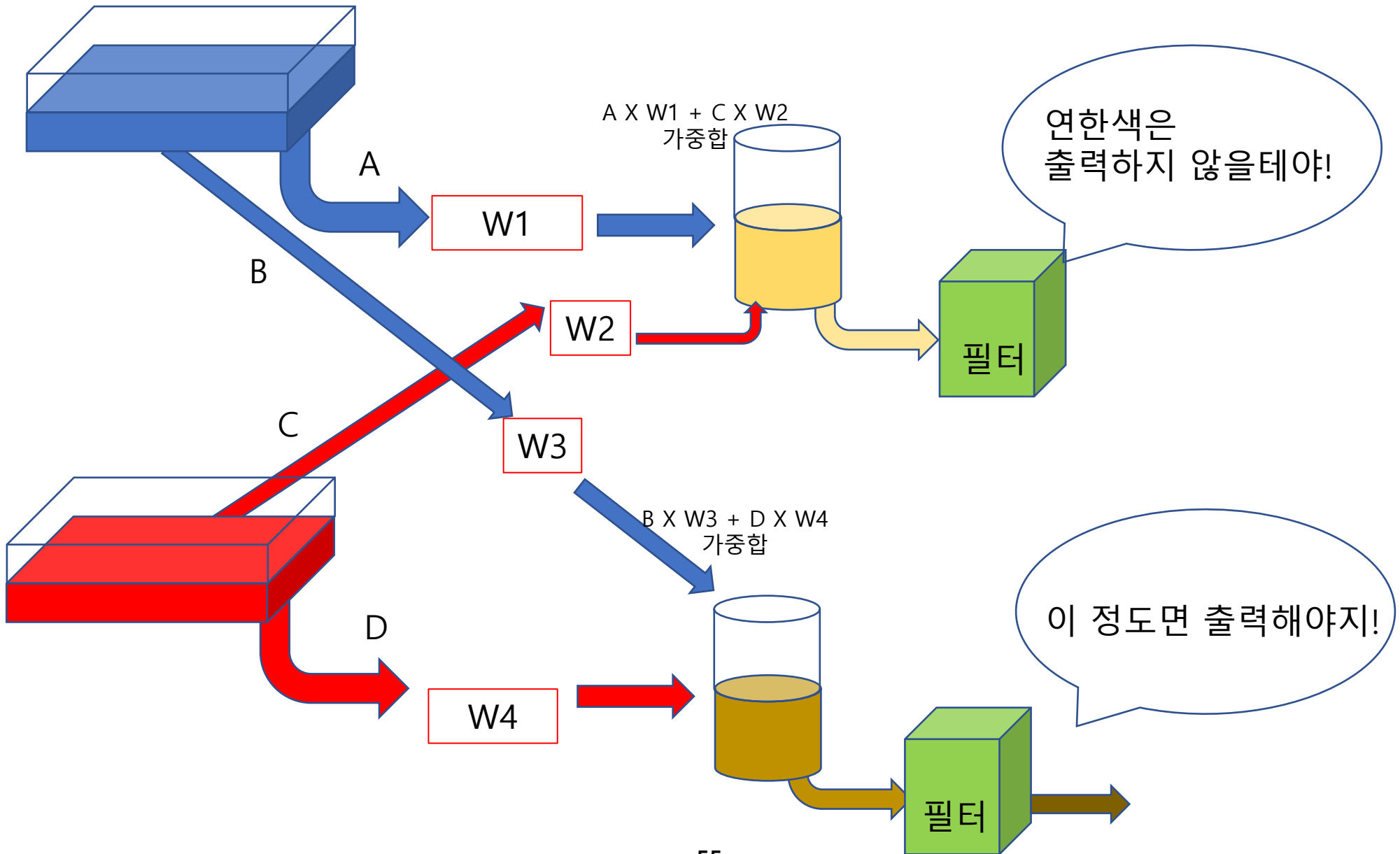
1. 활성화함수: 은닉층의 노드에서 입력 신호에 대해 적절한 처리를 하여 출력해주는 함수
2. 다양한 활성화함수: Step, Sigmoid, Linear, ReLU 등 다양한 형태의 활성화 함수가 있음
3. Feedforward 신경망: 노드 간의 연결에서 순환이나 루프가 없는 기본적인 인공신경망
4. 가중치 업데이트: 랜덤하게 주어진 가중치부터 시작하여 인공신경망이 좋은 성능을 보이도록 최적화된 가중치를 찾는 과정
5. 에포크: 인공신경망에서 순전파(Forward Propagation)와 역전파(Back Propagation)를 마친 것을 의미

## V. 활성화 함수와 가중치 업데이트

### 활성화 함수

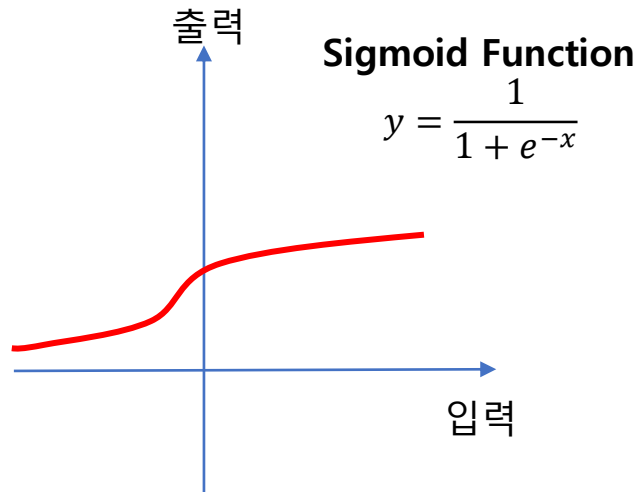
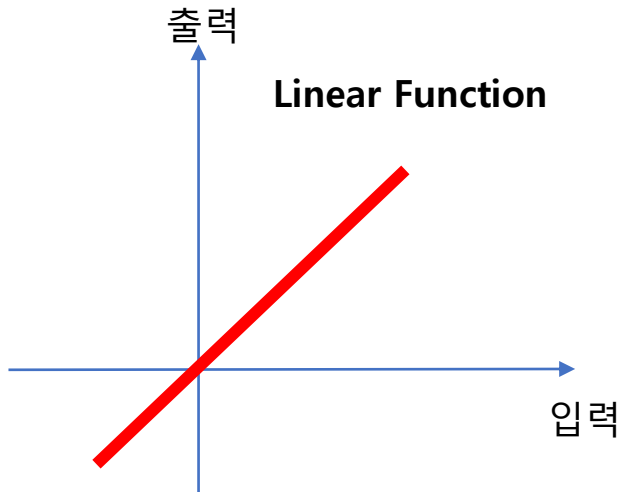
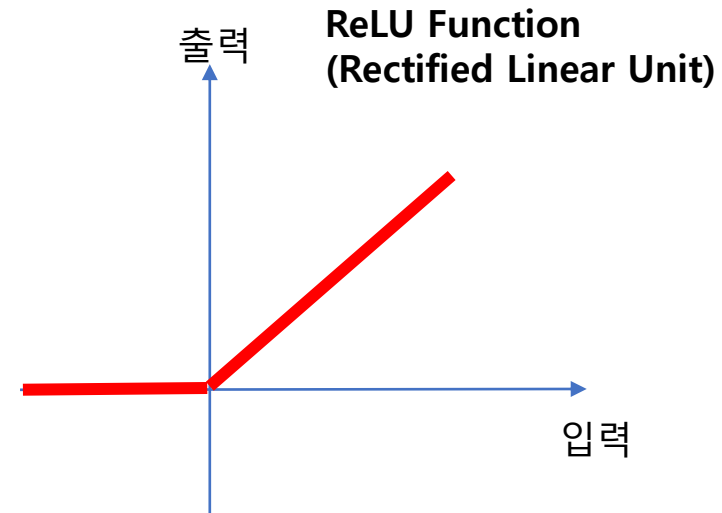
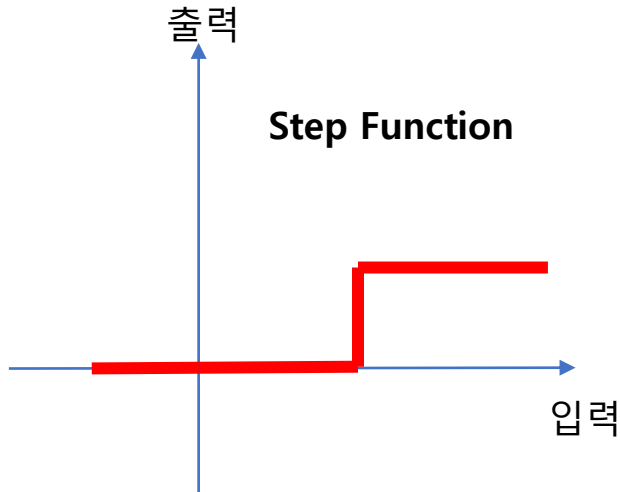


## V. 활성화함수와 가중치 업데이트



## V. 활성화 함수와 가중치 업데이트

### 다양한 활성화 함수





## V. 활성화함수와 가중치 업데이트

---

### Feedforward Neural Network

- 노드 간의 연결에서 순환이나 루프가 없는 기본적인 인공신경망



- Feedforward = 순방향



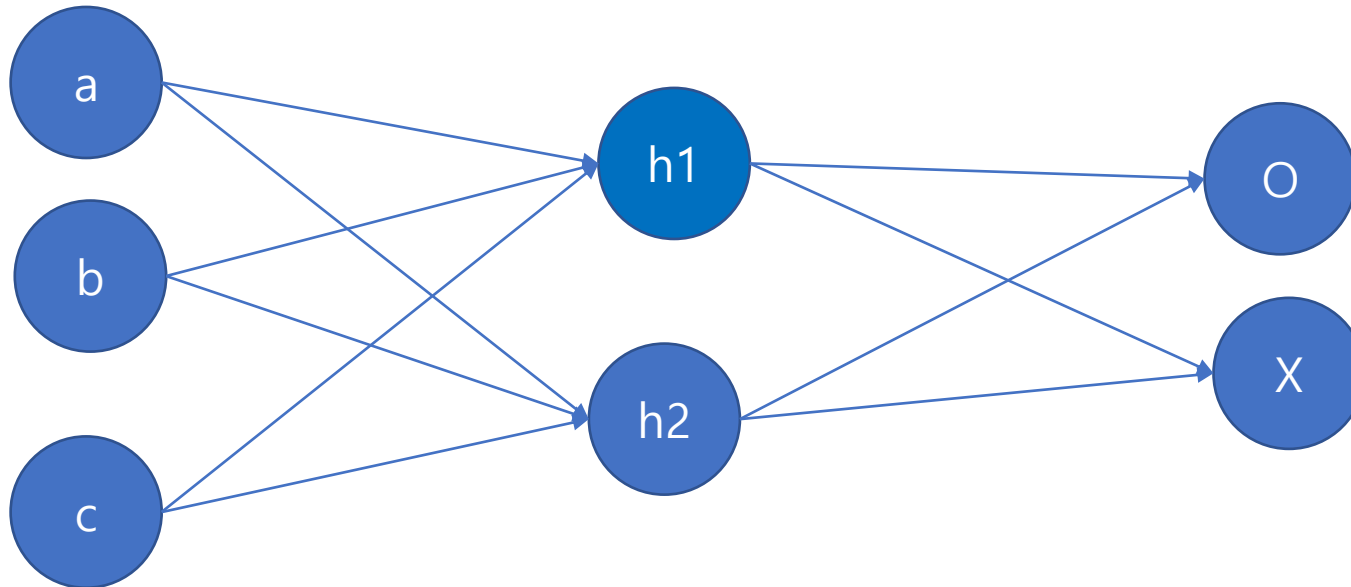
- Forward Propagation? Back Propagation?

## V. 활성화 함수와 가중치 업데이트

---

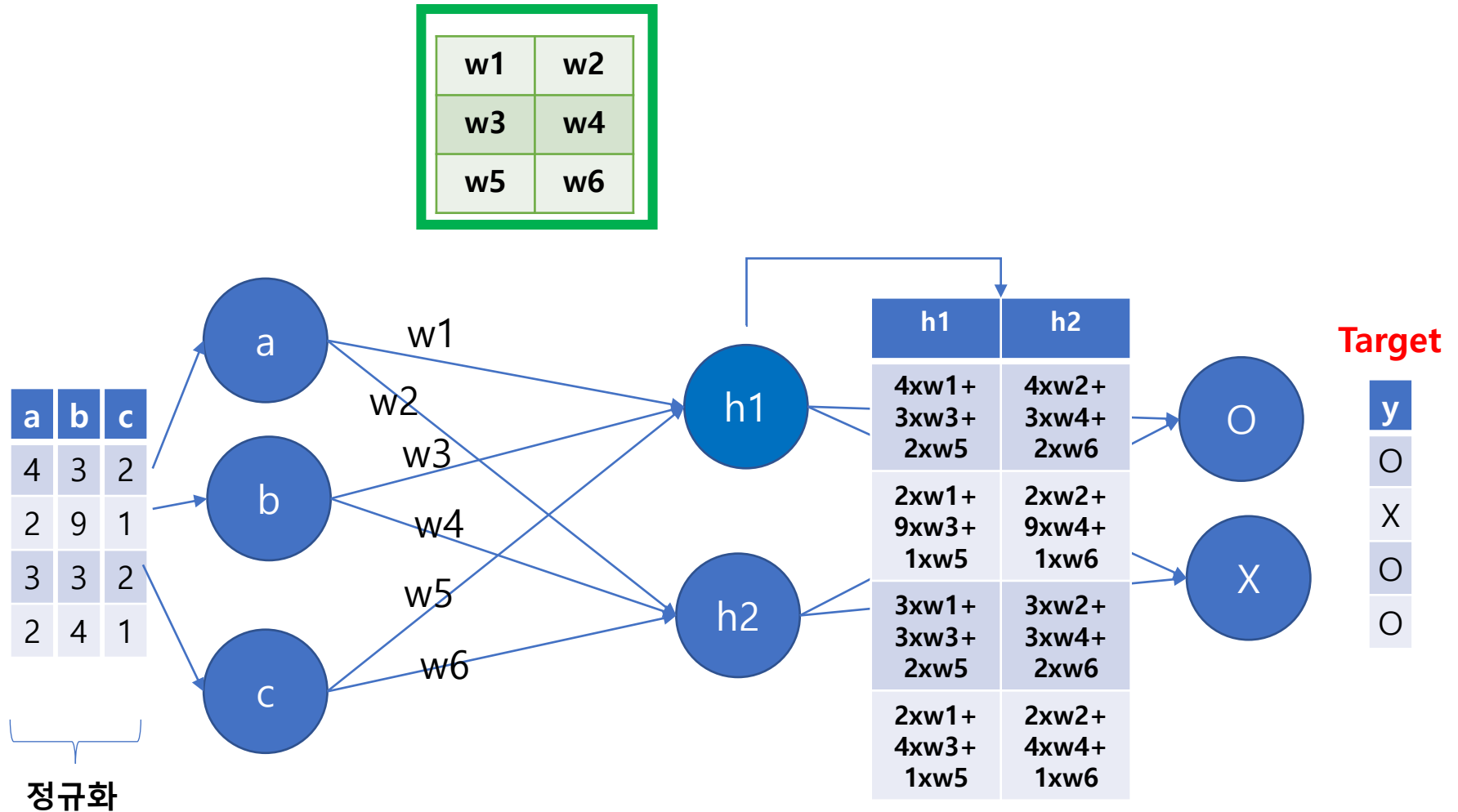
### Forward Propagation = Matrix Multiplication

가중치!



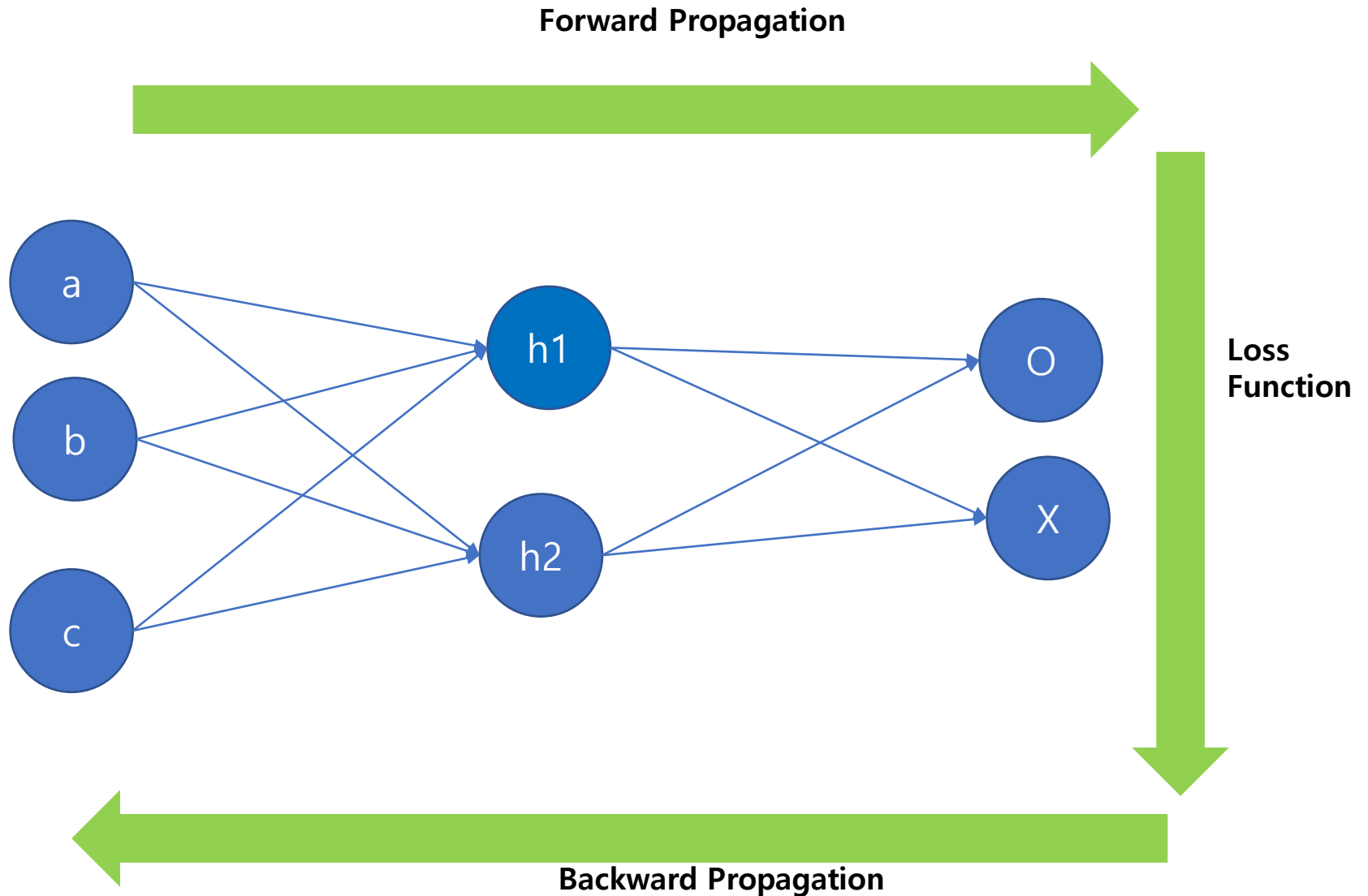
## V. 활성화함수와 가중치 업데이트

가중치: 처음엔 Random (0~1)



## V. 활성화 함수와 가중치 업데이트

### Epoch(에포크): Forward Propagation + Back Propagation



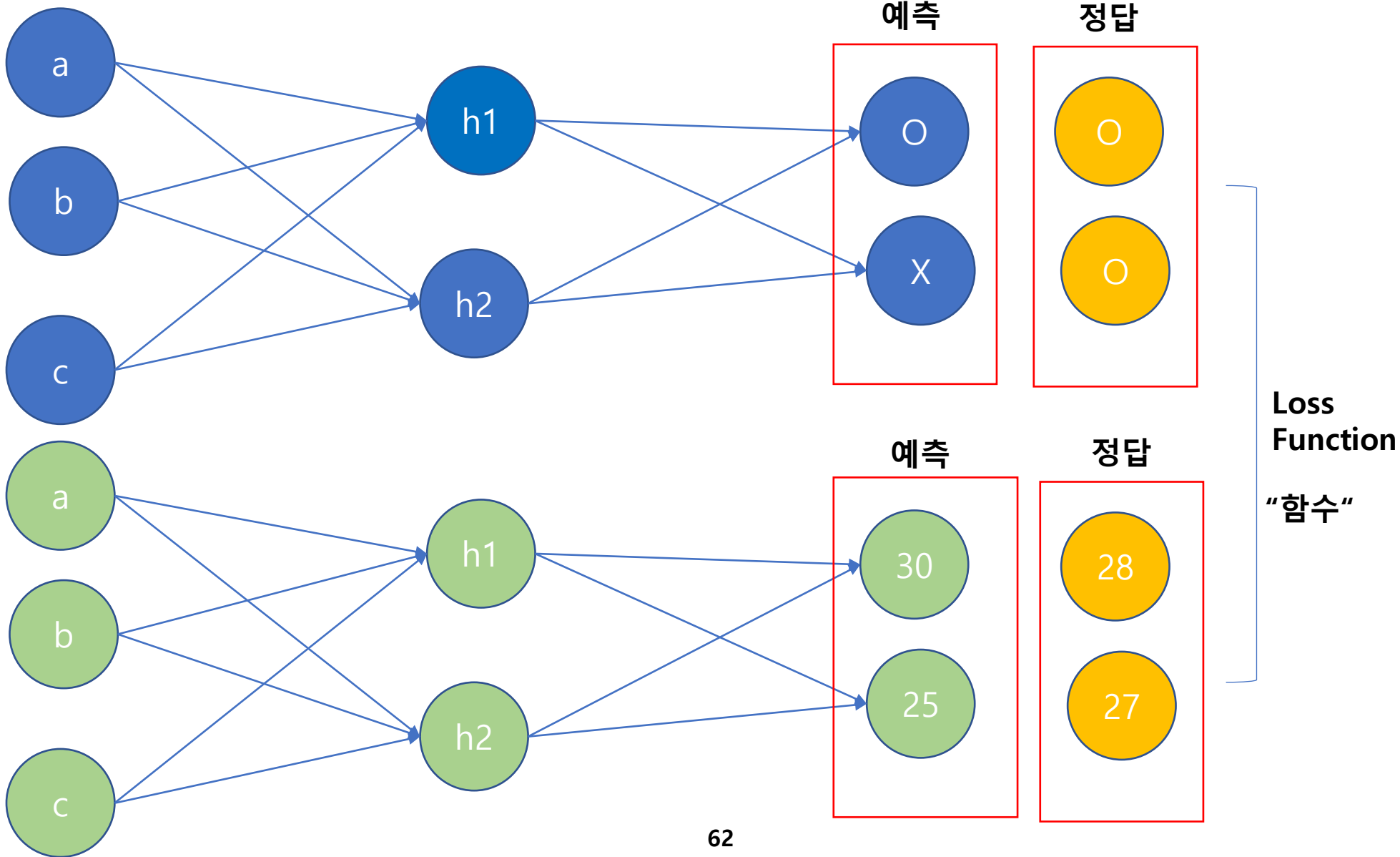
## VI. 역전파 알고리즘

---

1. 인공신경망의 오차: 인공신경망의 순전파 과정을 통해 계산된 예측값과 실제값과의 차이값
2. 역전파알고리즘: 오차를 활용하여 가중치를 업데이트하는 알고리즘
3. 경사하강법 : 가중치에 대한 오차 함수가 최소가 되는 지점을 발견하여 가중치를 최적화하는 방법
4. 학습율: 한 번의 에포크에서 얼마나 학습되는지를 알려주며, 이 값이 너무 크거나 작으면 최적의 가중치를 발견하기가 어려움

## VI. 역전파 알고리즘

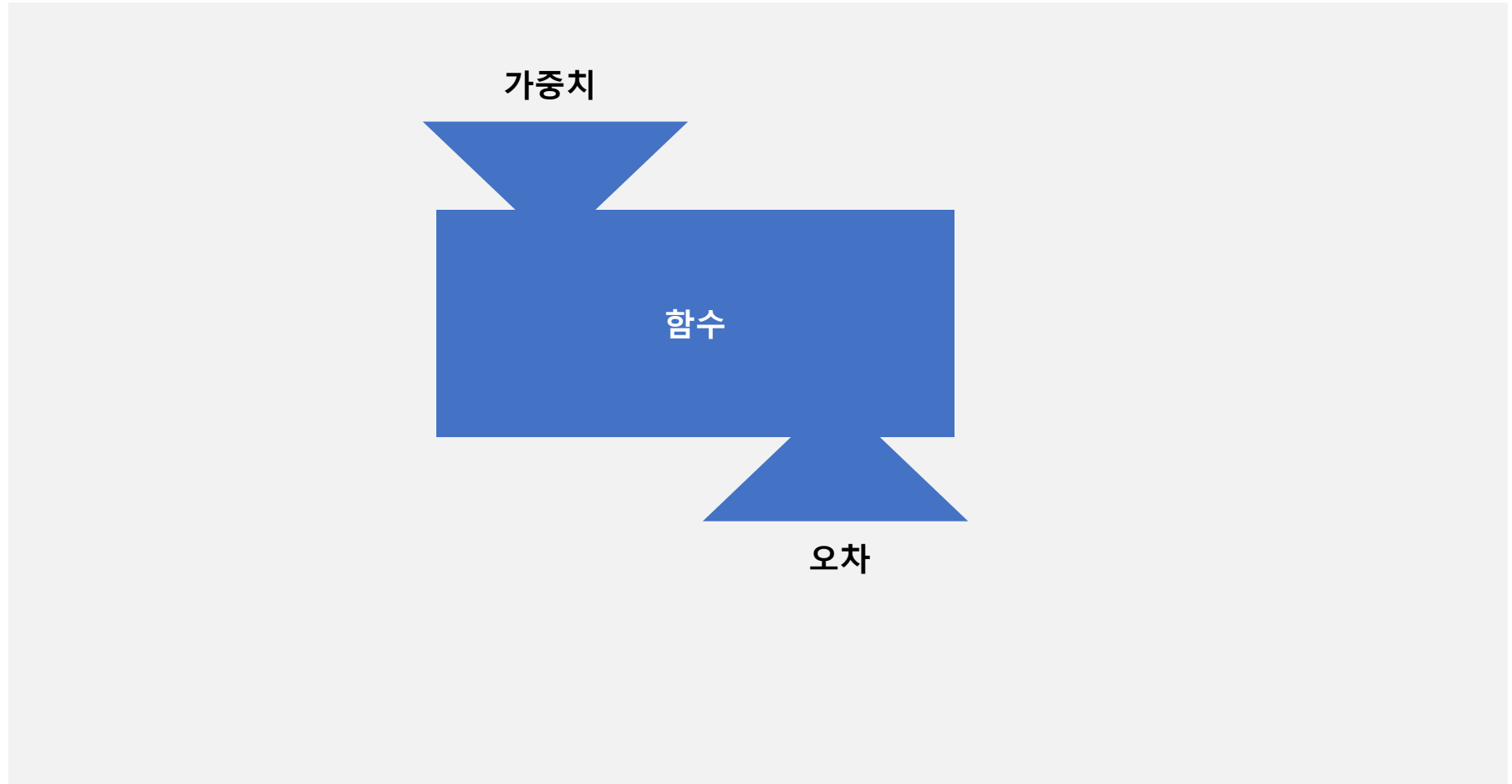
### 모형의 오차: Classification VS Regression



## VI. 역전파 알고리즘

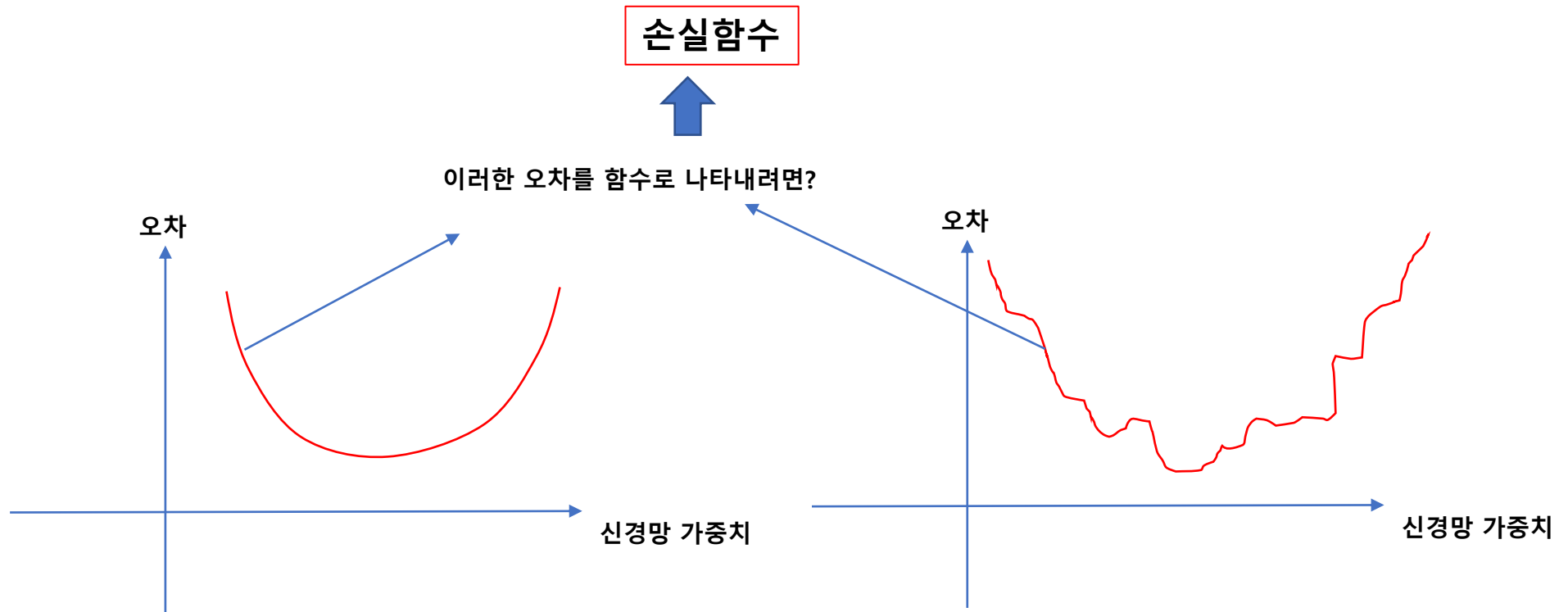
---

### 가중치와 오차의 함수



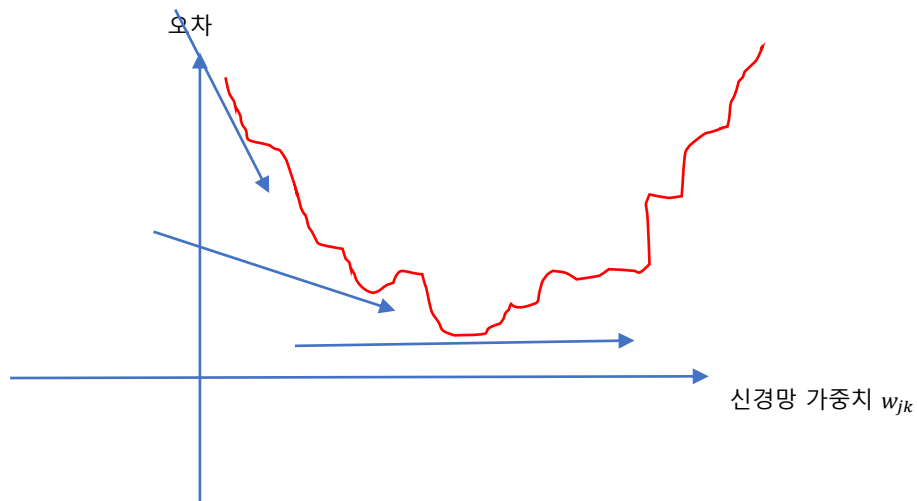
## VI. 역전파 알고리즘

### 오차에 대한 함수





## VI. 역전파 알고리즘



오차를 구하는 함수

$$E = (\text{실제값} - \text{오차})^2$$

기울기가 작아지거나 0이 되는 지점의 가중치

$$\frac{\partial E}{\partial w_{jk}} = \frac{\partial}{\partial w_{jk}} \sum_n (\text{목표값}_n - \text{출력값}_n)^2$$

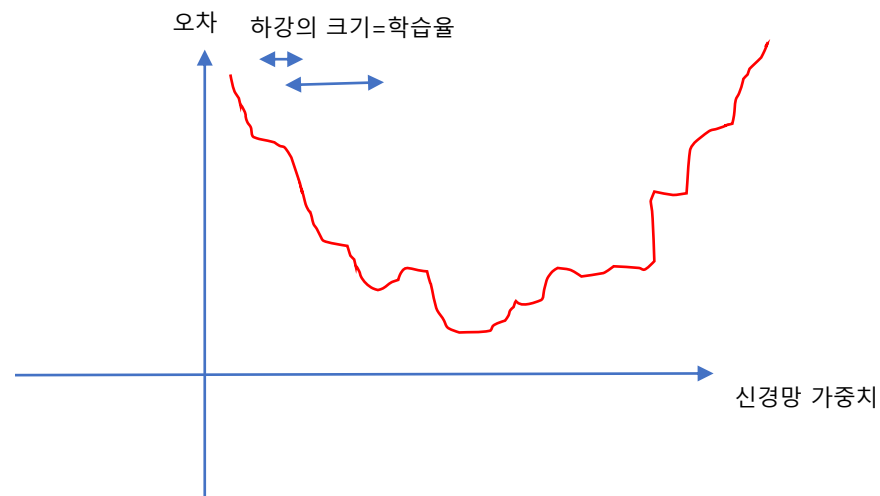
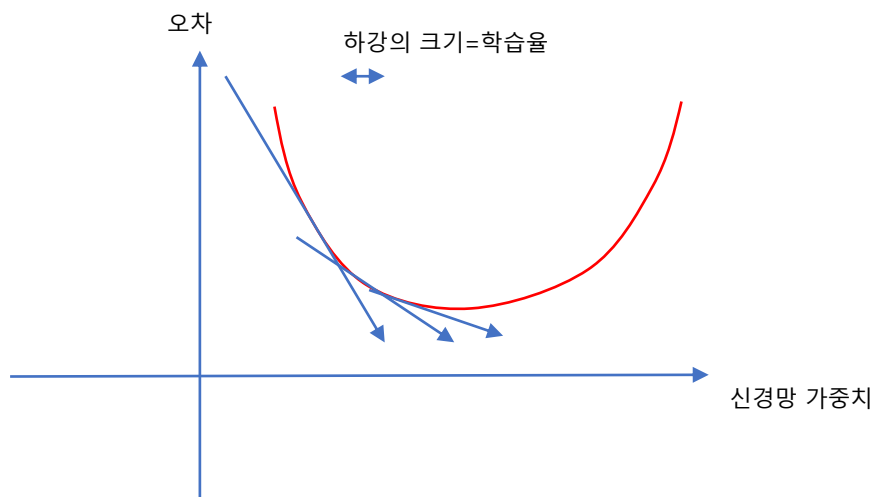
출력층 직전 은닉층이 n개의 노드  
j: 은닉층의 노드 수  
k: 결과층의 노드 수

$$\frac{\partial E}{\partial w_{jk}} = \text{오차 함수를 미분한 기울기}$$

$$\frac{\partial E}{\partial w_{jk}} = -2(t_k - o_k) \times \text{sigmoid}(\sum_j w_{jk} o_j) (1 - \text{sigmoid}(\sum_j w_{jk} o_j)) \times o_j$$

## VI. 역전파 알고리즘

- 신경망 가중치가 입력이고 오차가 출력인 함수 대상
- 경사하강법: 오차의 최소화
- 학습율: 하강의 크기



## VI. 역전파 알고리즘

---

### 학습률(Learning Rate)

가중치 조정과 학습율

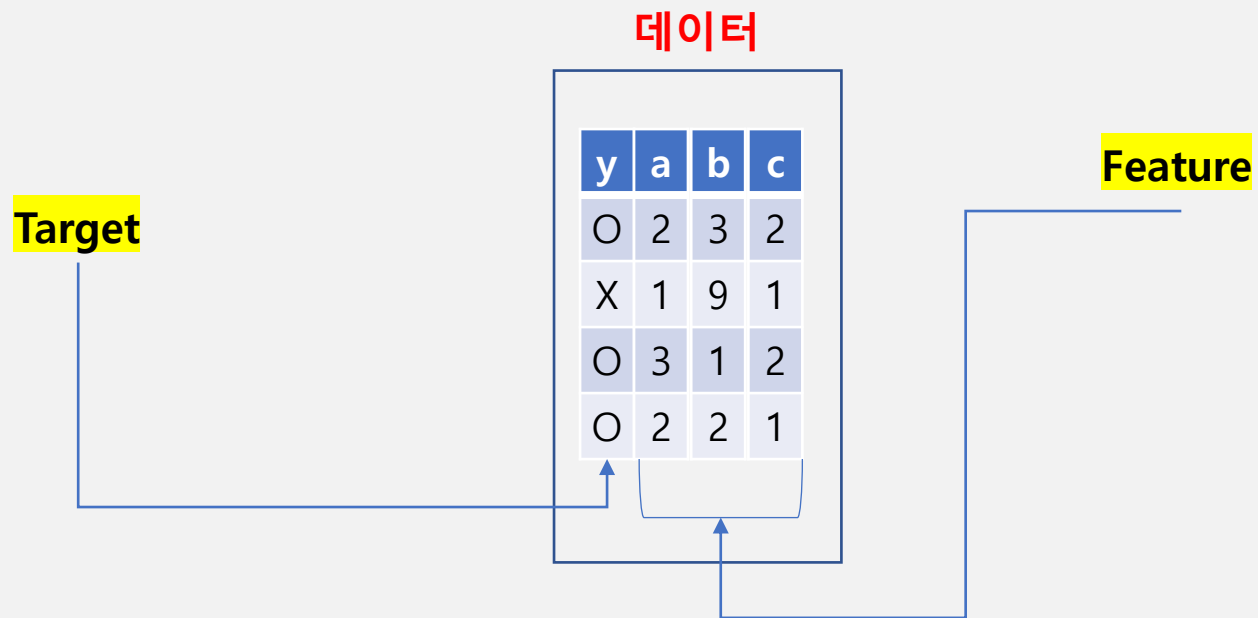
$$\text{새로운 } w_{jk} = \text{이전 } w_{jk} - \alpha \times \frac{\partial E}{\partial w_{jk}}$$



- 이전 가중치에서 오차의 변화율을 빼주기
  - 양의 기울기(오차 증가시킴)이면 이전 가중치에서 빼주어 영향을 덜 받게 하고, 음의 기울기(오차 감소)이면 이전 가중치를 더 크게 해주는 효과(-의-)
- $\alpha$ 는 가중치 변화하는 정도를 조정:학습률

## VI. 역전파 알고리즘

### ANN Step by Step

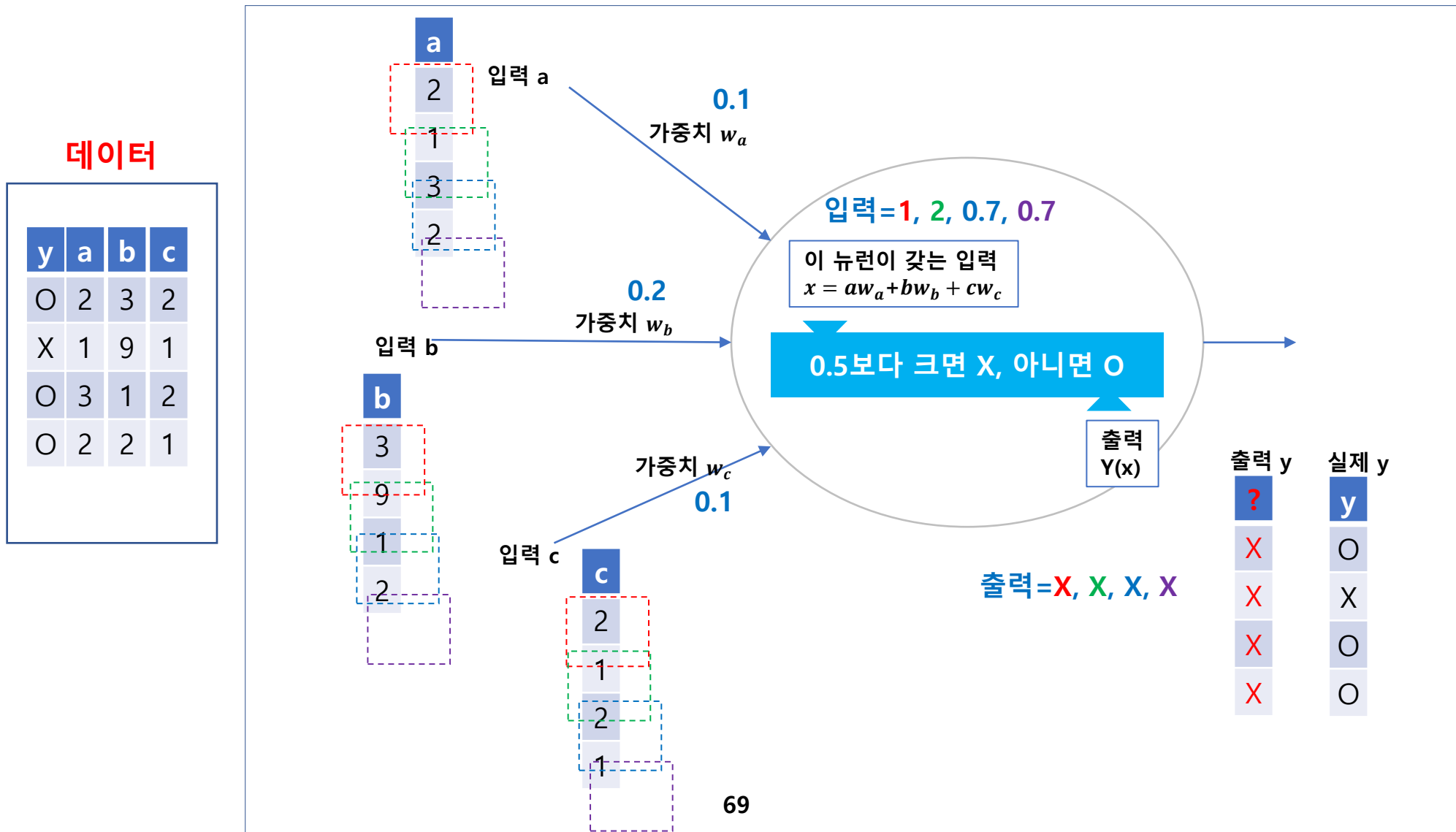


*Supervised Learning-Classification*

## VI. 역전파 알고리즘

### ANN Step by Step

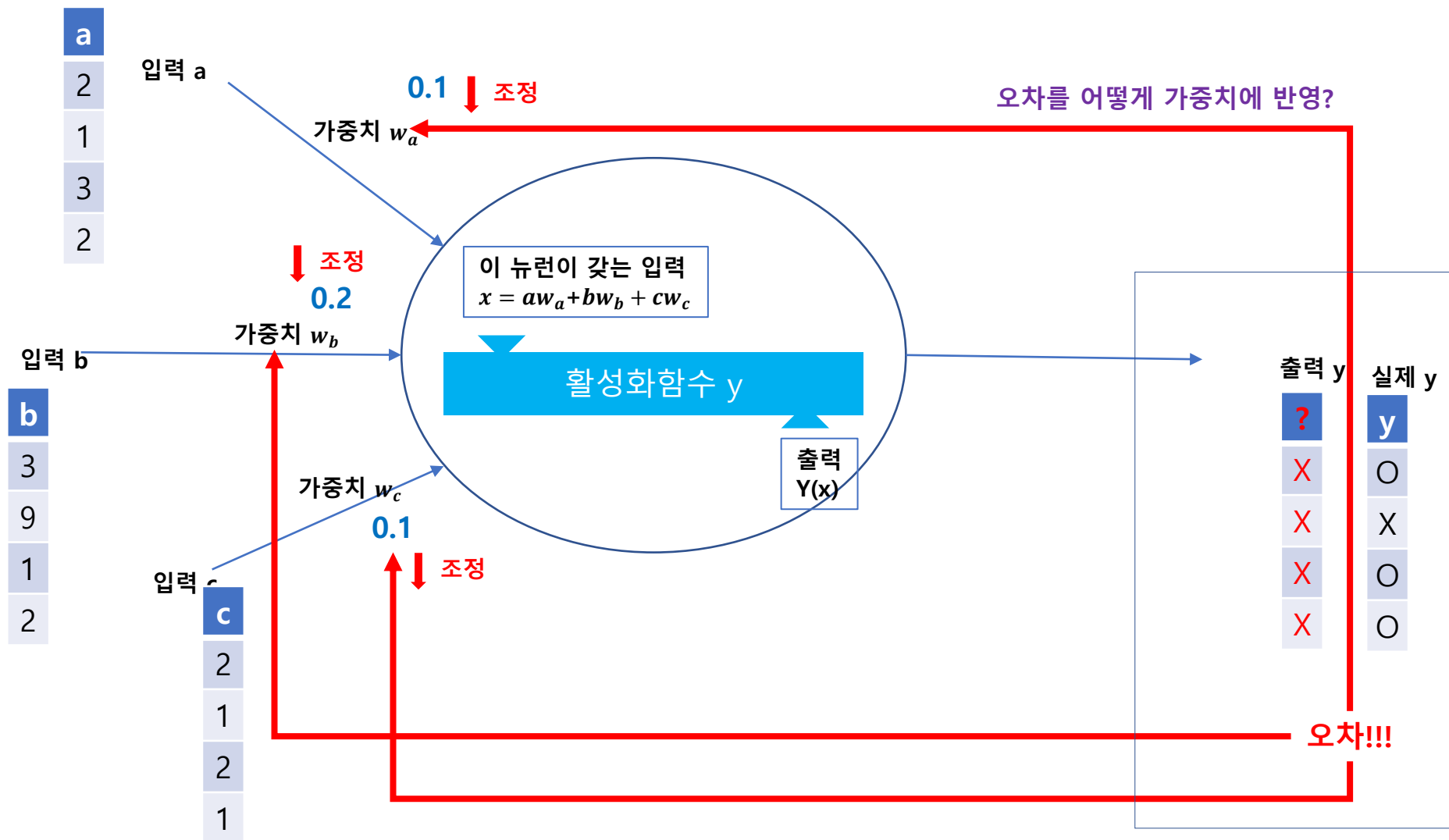
#### 인공신경망 1단계: 전파!



## VI. 역전파 알고리즘

### ANN Step by Step

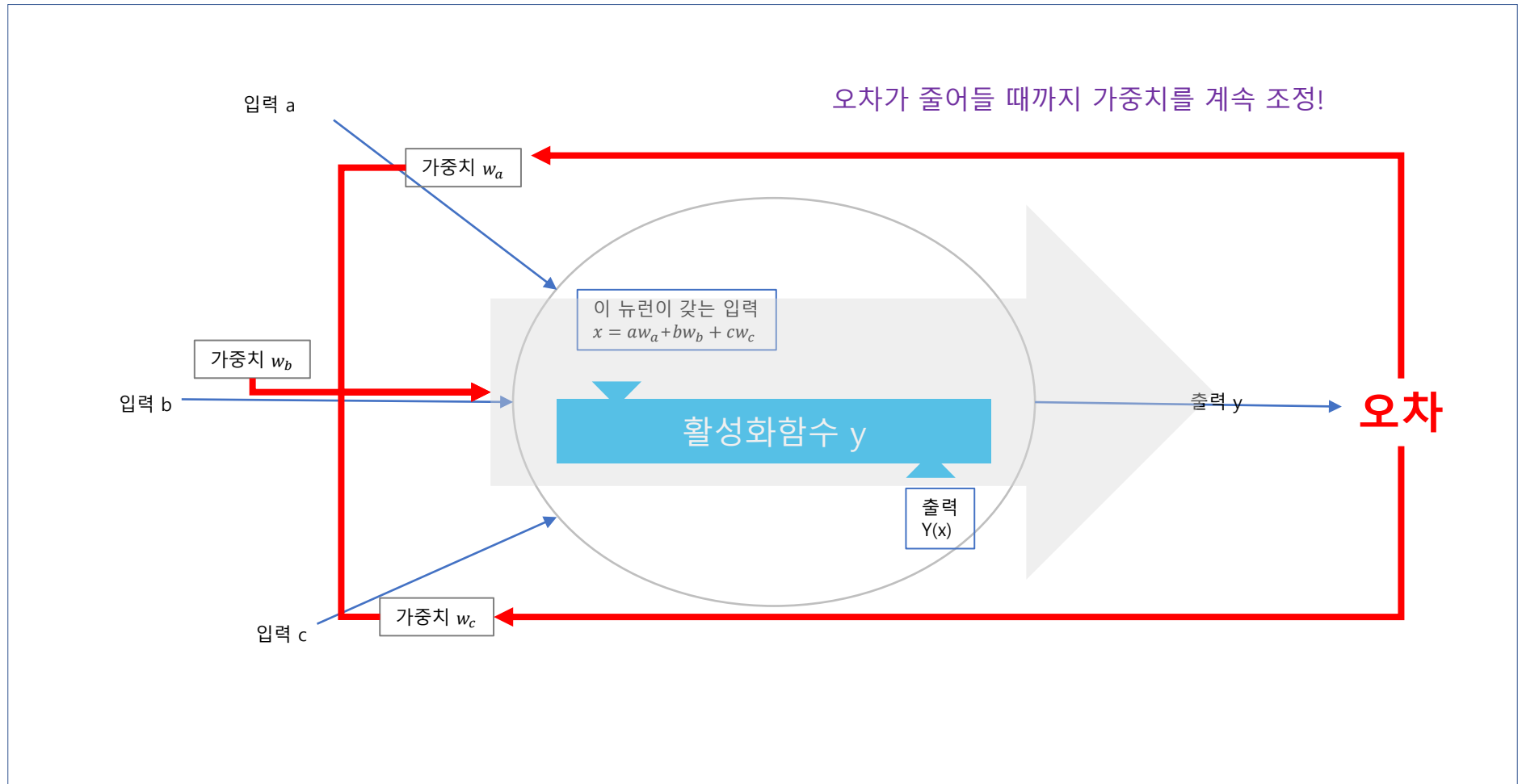
#### 인공신경망 2단계: 오차의 역전파!



## VI. 역전파 알고리즘

### ANN Step by Step

잘 할 때까지 반복!



## VII. ANN의 이슈와 성능 개선

---

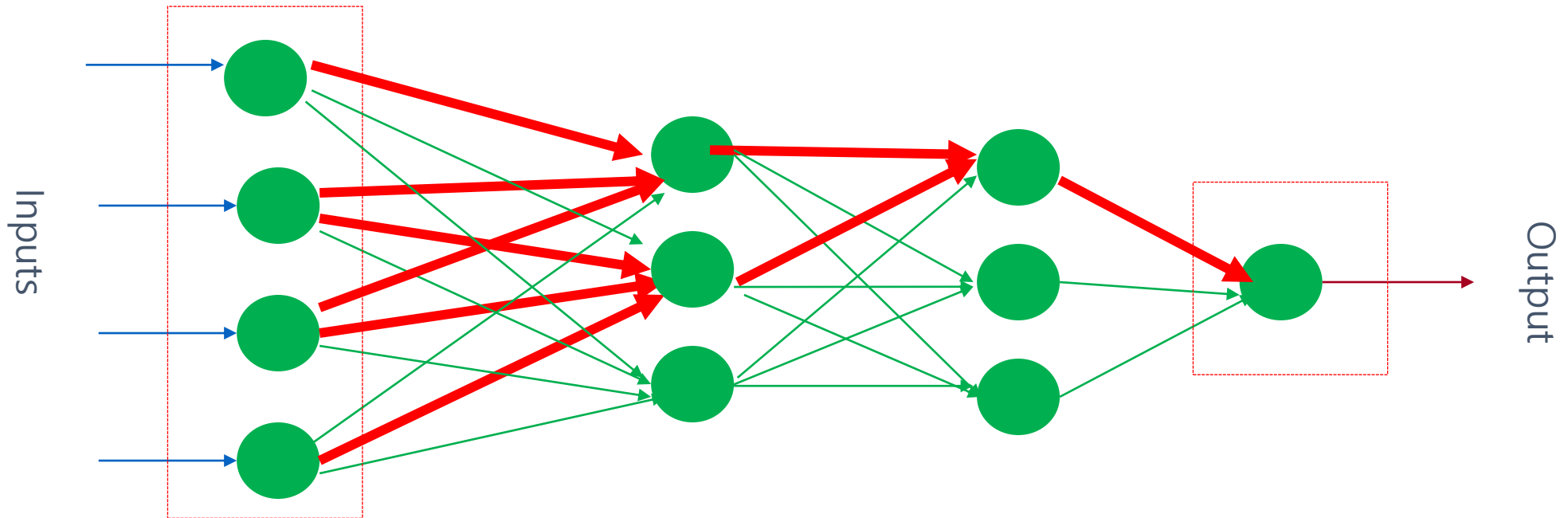
1. ANN 난제1-Overfitting과 해석의 어려움: 주어진 데이터에 적합한 ANN의 가중치들로 인한 Overfitting, 가중치 해석의 어려움
2. ANN 난제2-네트워크 구조: 입력층-은닉층-결과층으로 구성되는 ANN에서 은닉층의 노드의 수 등을 결정하기 위해 시행착오를 거치게 됨
3. ANN 난제3-Vanishing Gradient: 은닉층의 노드에서는 활성화함수를 통해 입력된 값이 변환되고, 역전파 과정에서 활성화함수의 미분값이 사용되며, 학습을 거듭할 수록 기울기가 작아지는 현상



## VII. ANN의 이슈와 성능 개선

- **Artificial Neural Network(ANN, 인공신경망)**

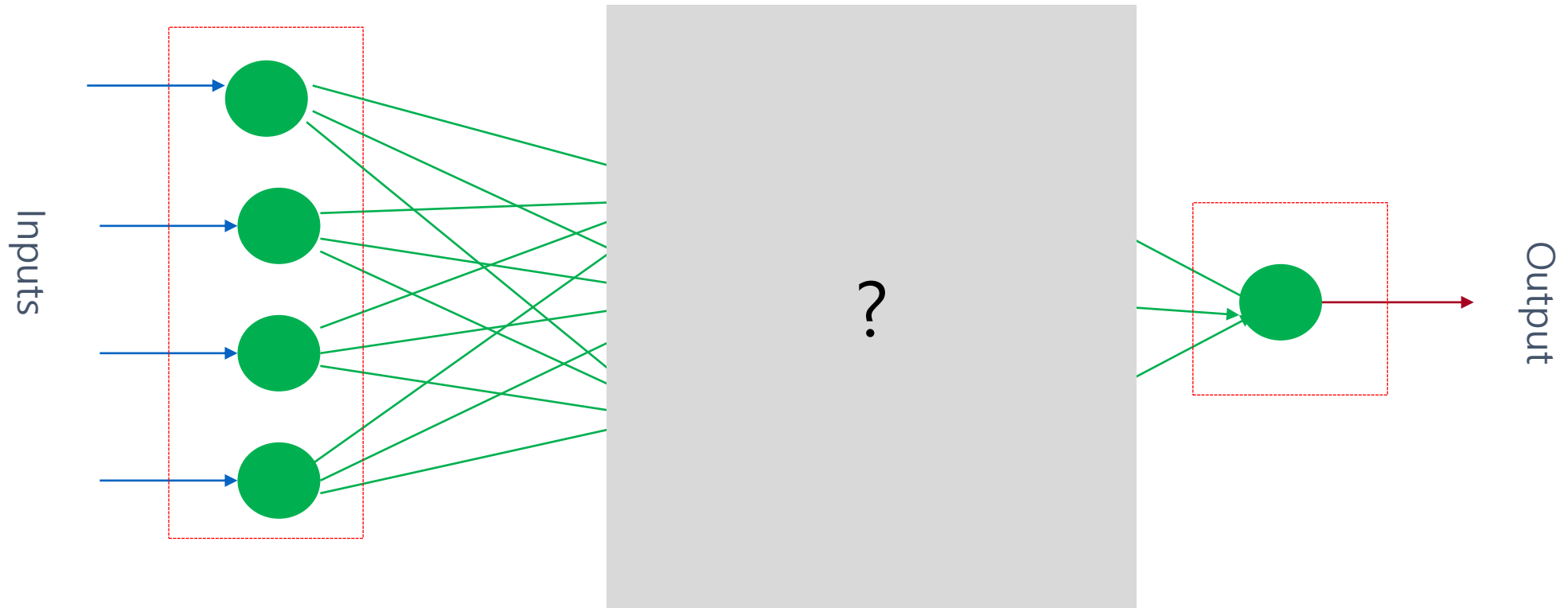
- 주어진 데이터에 적합한 다수의 가중치들이 최적값을 갖도록 학습
- 어떤 X변수의 값의 단위크기 변화에 따른 결과에의 영향을 아는 것이 사실상 불가



## VII. ANN의 이슈와 성능 개선

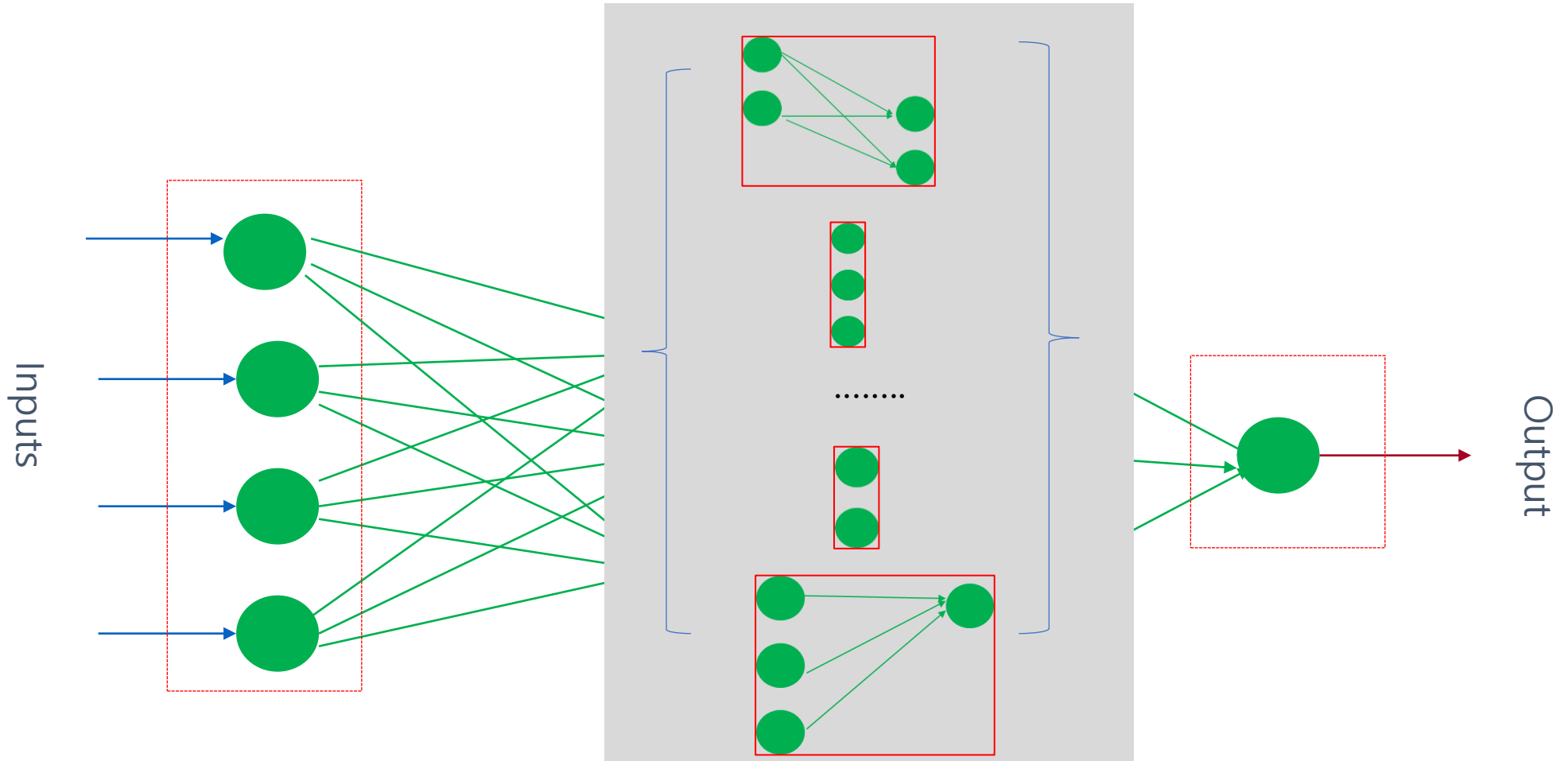
- **Artificial Neural Network(ANN, 인공신경망) = Black Box**

- ANN은 입력층-은닉층-결과층으로 이루어짐
- 은닉층은 어떻게 구성되는지에 따라 ANN의 성능이 결정



## VII. ANN의 이슈와 성능 개선

- 은닉층 구성은 Trial and Error를 통해 접근해 나감 (Grid Search)



## VII. ANN의 이슈와 성능 개선

---

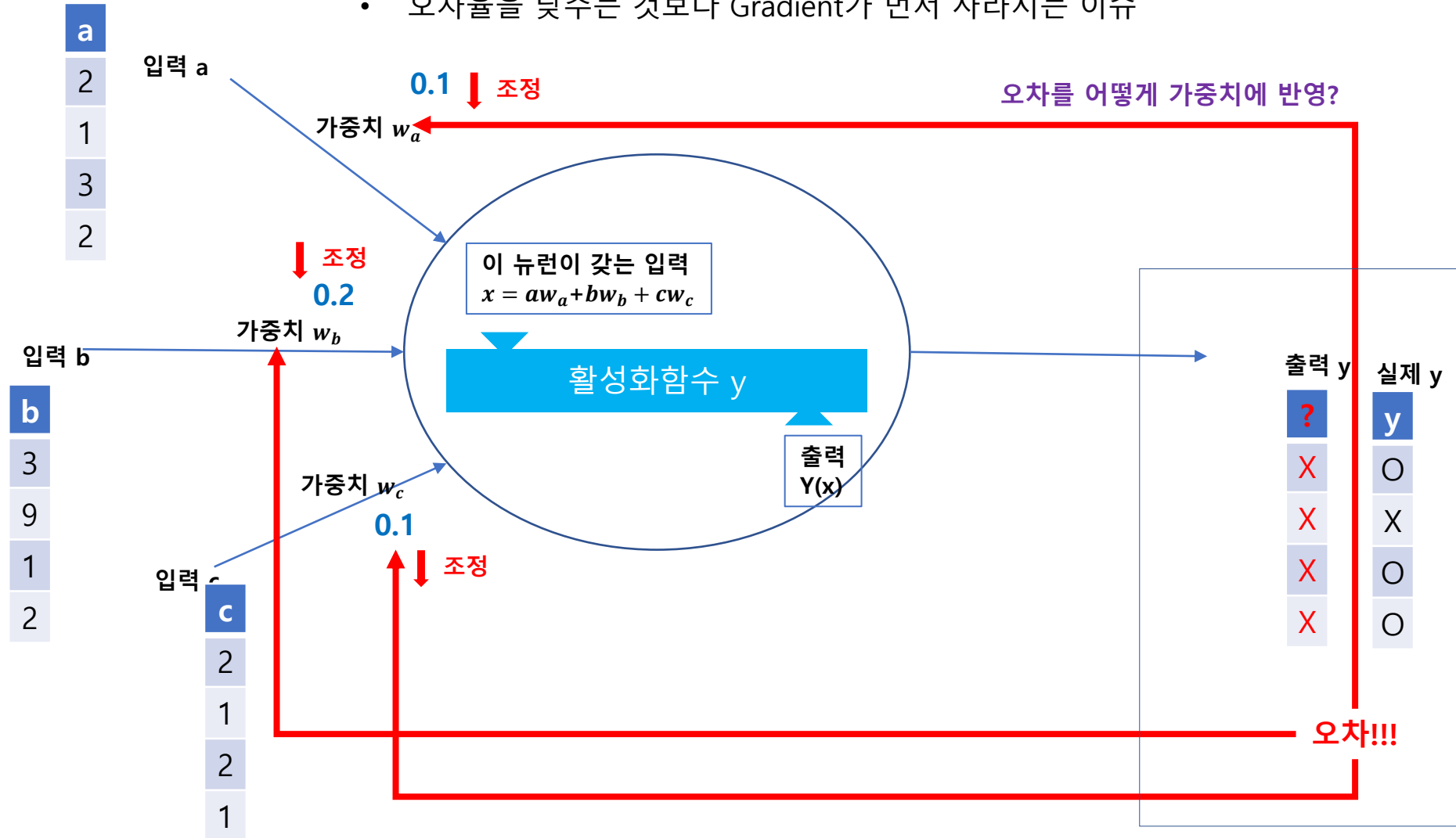
### Vanishing Gradient

- Gradient? 미분된 함수의 기울기를 의미
- ANN의 활성화 함수에 대해 기울기를 구할 수 있음
- 활성화 함수의 미분은 ANN 오차의 역전파(Back propagation) 과정에 사용

## VII. ANN의 이슈와 성능 개선

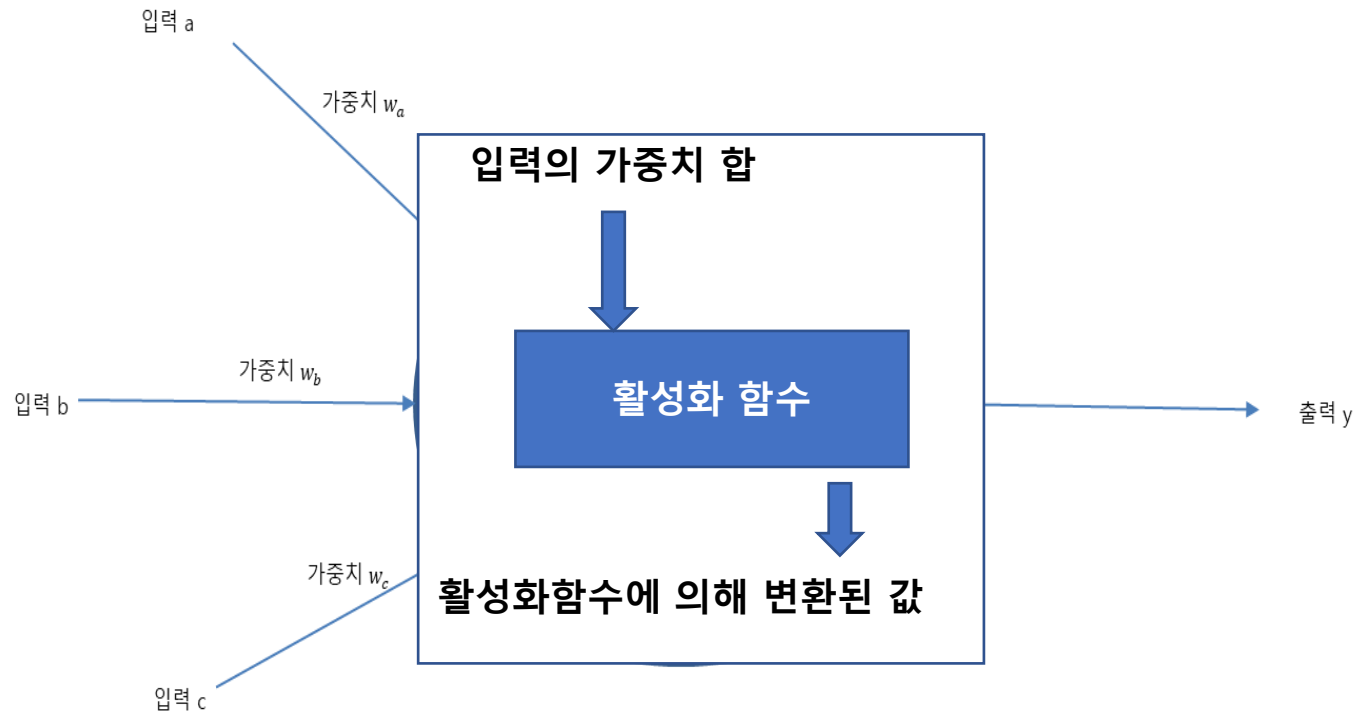
### Vanishing Gradient

- 역전파를 통한 가중치 학습이 계속될 수록 오차가 감소
- 감소된 오차는 오차의 역전파 과정에서 작은 변화량만 전달
- 오차율을 낮추는 것보다 Gradient가 먼저 사라지는 이슈



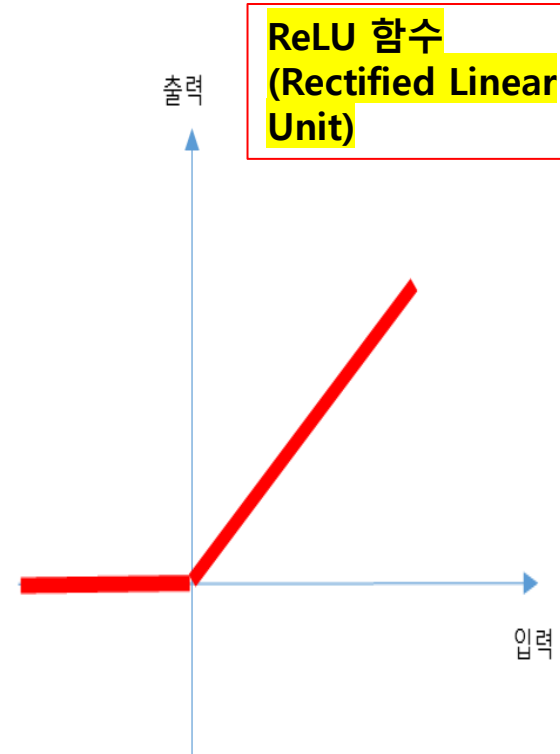
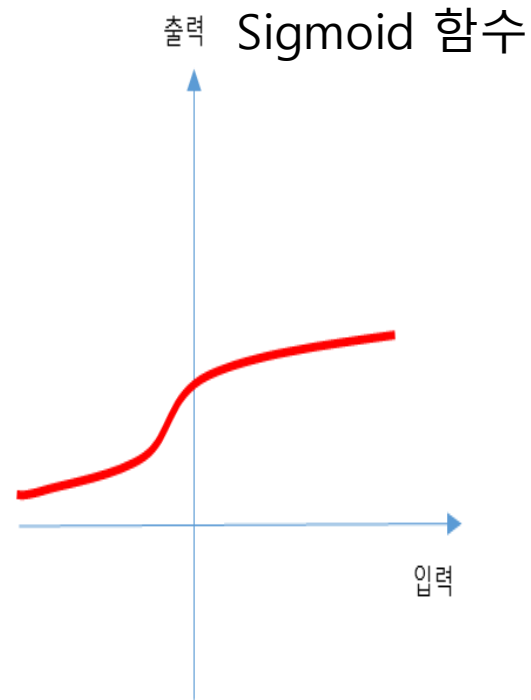
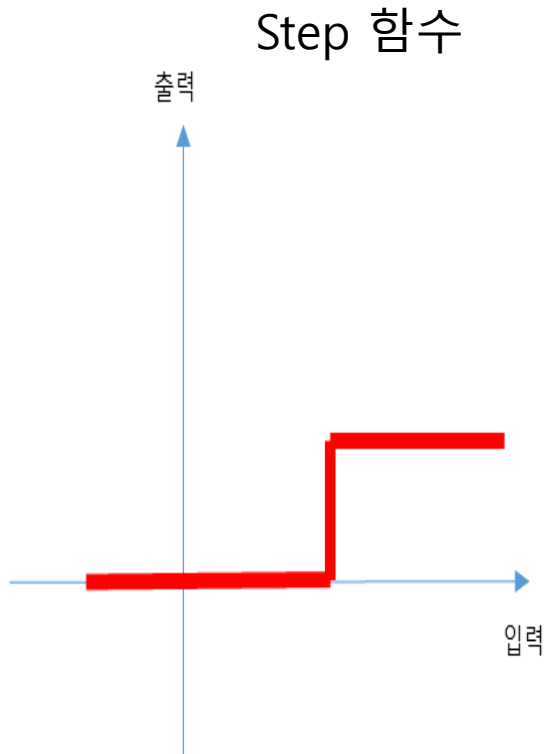
## VII. ANN의 이슈와 성능 개선

- **인공신경망 (Artificial Neural Network, ANN)의 활성화 함수**
  - 은닉층 노드에 입력되는 이전 단계 출력의 가중치 합을 변환시키는 함수
  - Activation function



## VII. ANN의 이슈와 성능 개선

- 다양한 활성화 함수
  - 여러 종류의 활성화 함수들이 사용



활성화 함수에 따라 가중치의 최적화도 영향을 받으며 모형의 성능에도 직결!

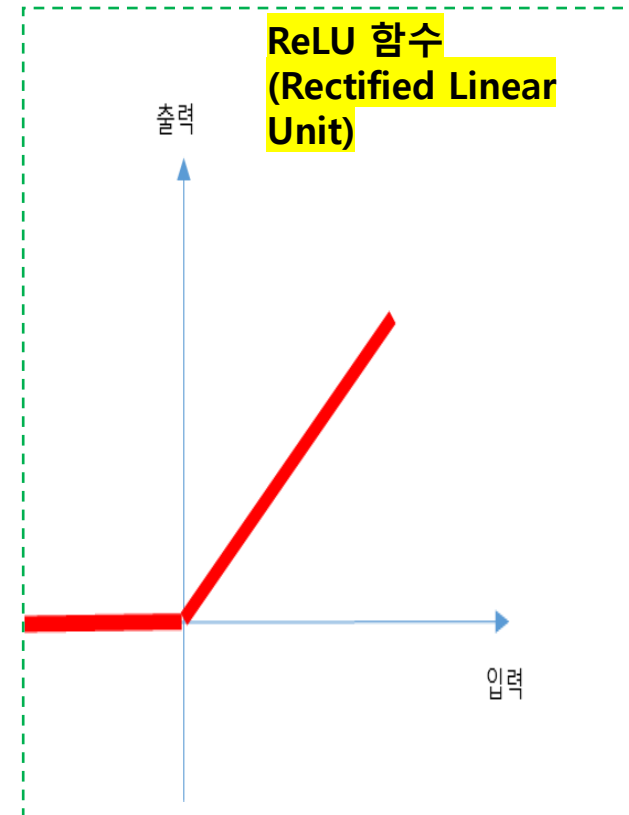
## VII. ANN의 이슈와 성능 개선

- ReLU 활성화 함수

Sigmoid 함수에 의한  
Vanishing Gradient의 문제

->

ReLU함수를 통한 개선+빨라진 학습





## VII. ANN의 이슈와 성능 개선

---

### GPU의 사용

- 다층 신경망의 성능 문제 해결
  - 2006년 힌턴(Hinton) 교수의 논문(A fast learning algorithm for deep belief nets)에서 해결의 단초가 제공, 이후 딥러닝이라는 이름
  - 딥러닝: 여러 개의 은닉층을 갖는 인공 신경망을 학습



- 다층 신경망의 성능 문제를 해결
  - 계산의 이슈!
    - 대용량의 계산, 그러나 각각의 계산이 아주 복잡하지는 않음
    - 굳이 CPU를 많이 사용할 필요는 없으며, GPU를 사용
    - GPU를 통한 딥러닝!

---

**QnA**