

/* elice */

양재 AI School 인공지능 캠프

퍼셉트론과 다층 퍼셉트론



박상수 선생님

커리큘럼

1 ○ 신경망 기초

신경망의 기본 내용인 퍼셉트론을 배우기 전에
신경망의 역사와 이와 관련된 배경지식에 대해 간단하게 살펴봅니다.

2 ○ 퍼셉트론 및 다층 퍼셉트론 (MLP), 그리고 학습 방법

퍼셉트론의 구조와 동작, 학습에 대해 살펴보고
선형 분류기로서의 퍼셉트론의 한계에 대해 배웁니다.
다층 퍼셉트론의 특성을 간단하게 파악합니다.

목차

1. **신경망 기초**
2. **퍼셉트론 (Perceptron)**
3. **다층 퍼셉스론 (Multi Layer Perceptron)**

현재 신경망의 연구는 ?

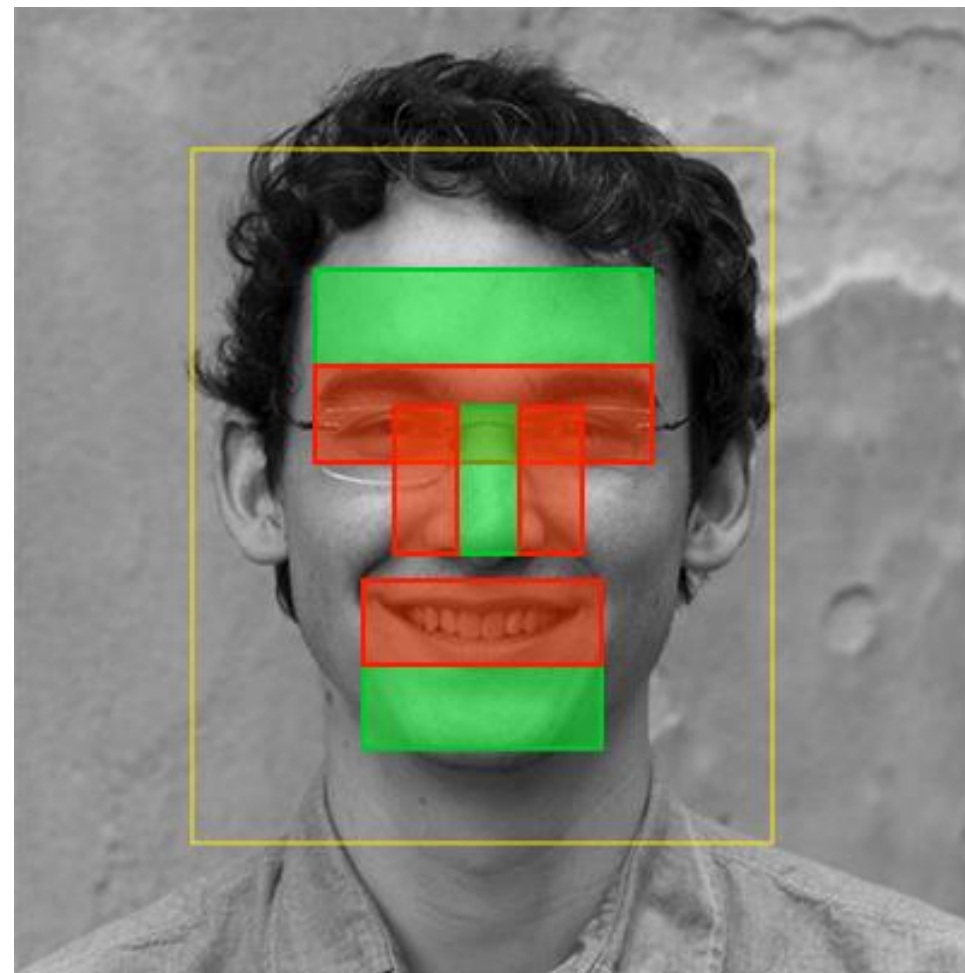


GAN을 활용한 Style transfer

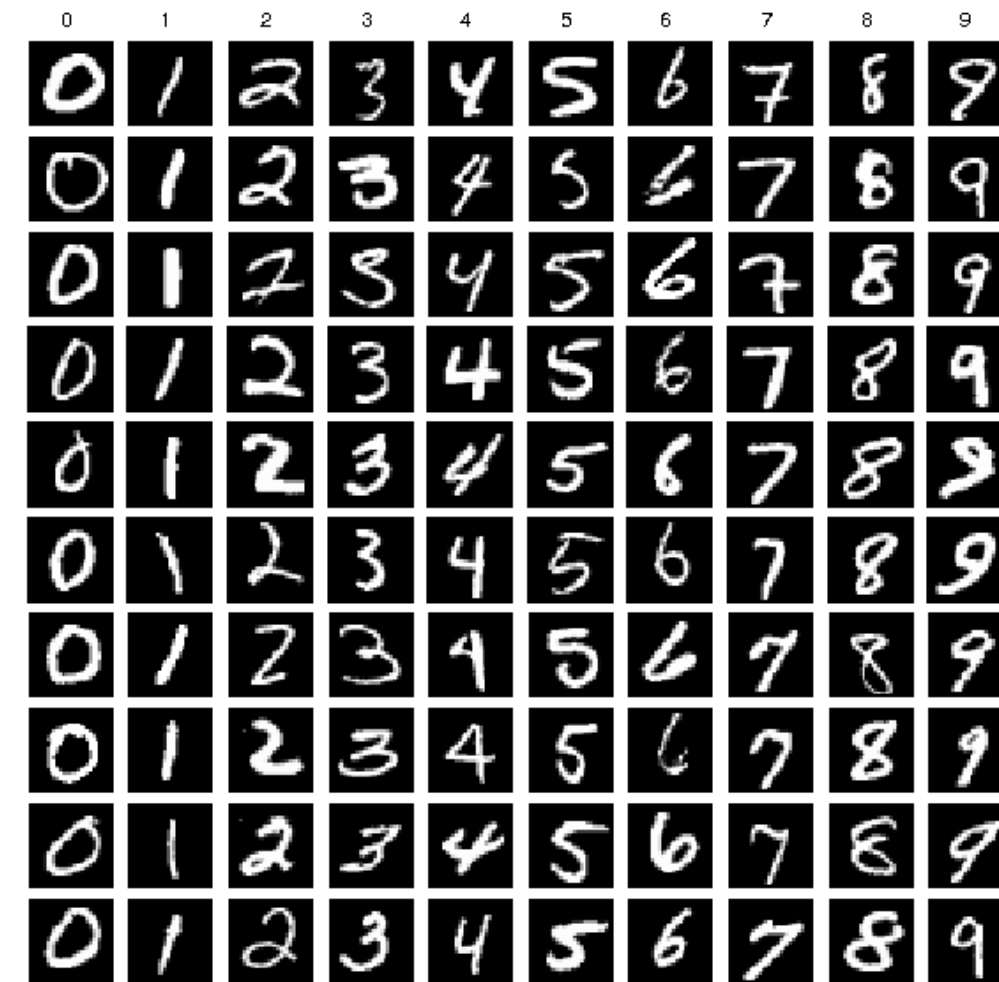
신경망 기초

인공신경망과 생물신경망은 무슨 관계가 있는 것인가 ?

신경망 이전의 연구는 ?



얼굴 인식



숫자 및 문자 인식

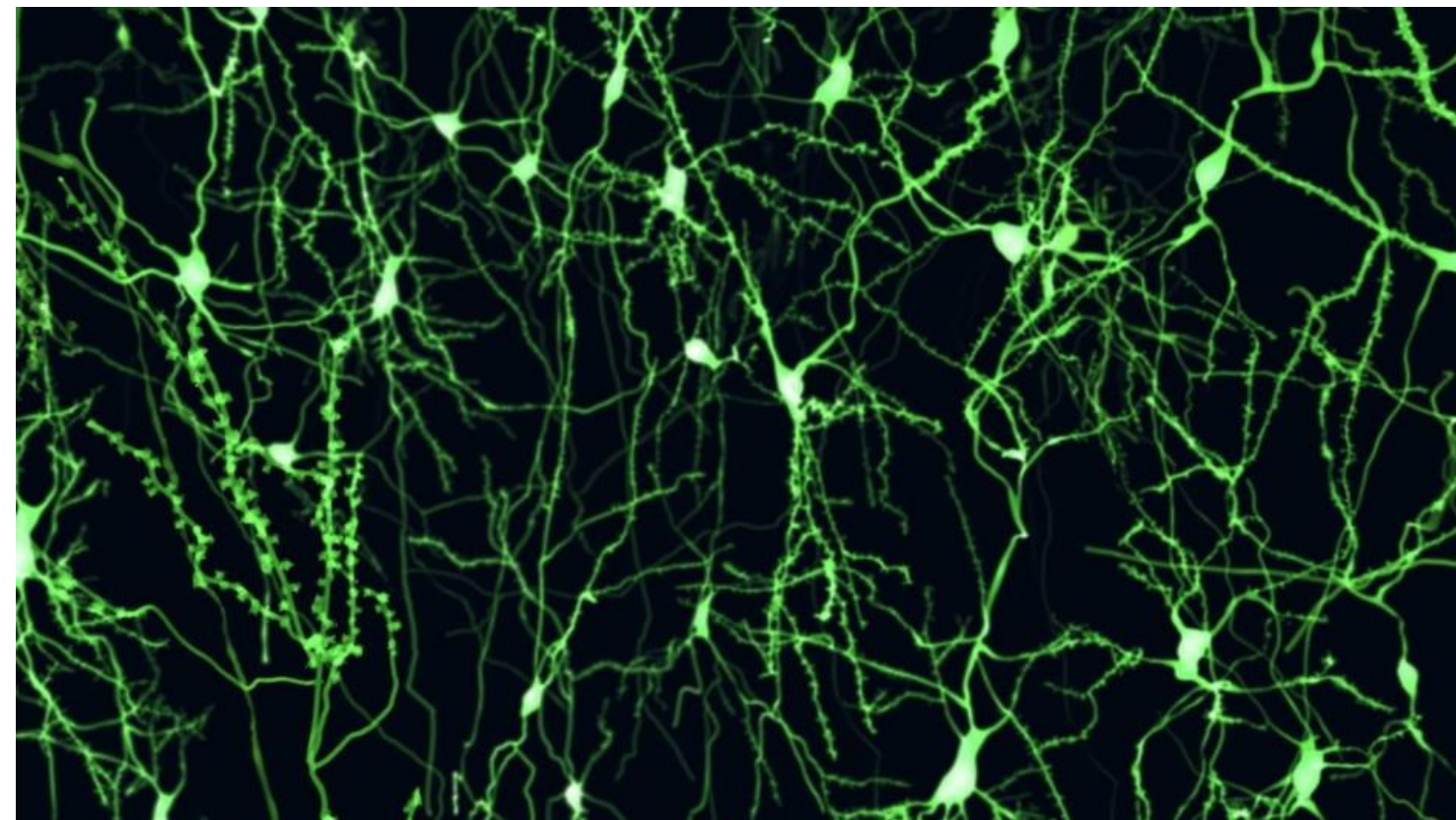
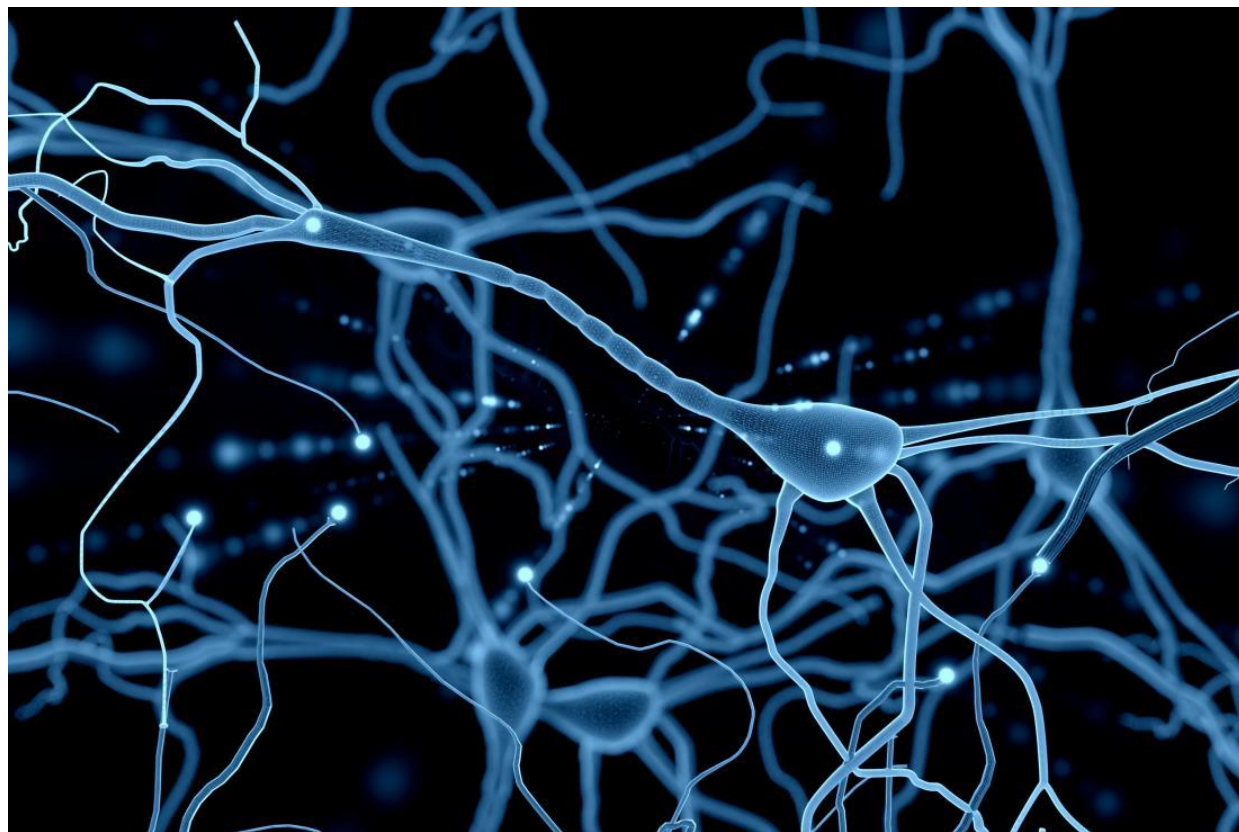
프로그래밍으로 풀 수 없는 문제들

프로그래밍으로 풀려면

```
if () then {  
  else if () then {  
    } else if () then {  
  } else if () then {} else if () then {}  
  .....  
}
```

발생하는 **모든 경우**에 대비
그래도 예외 상황은 존재함

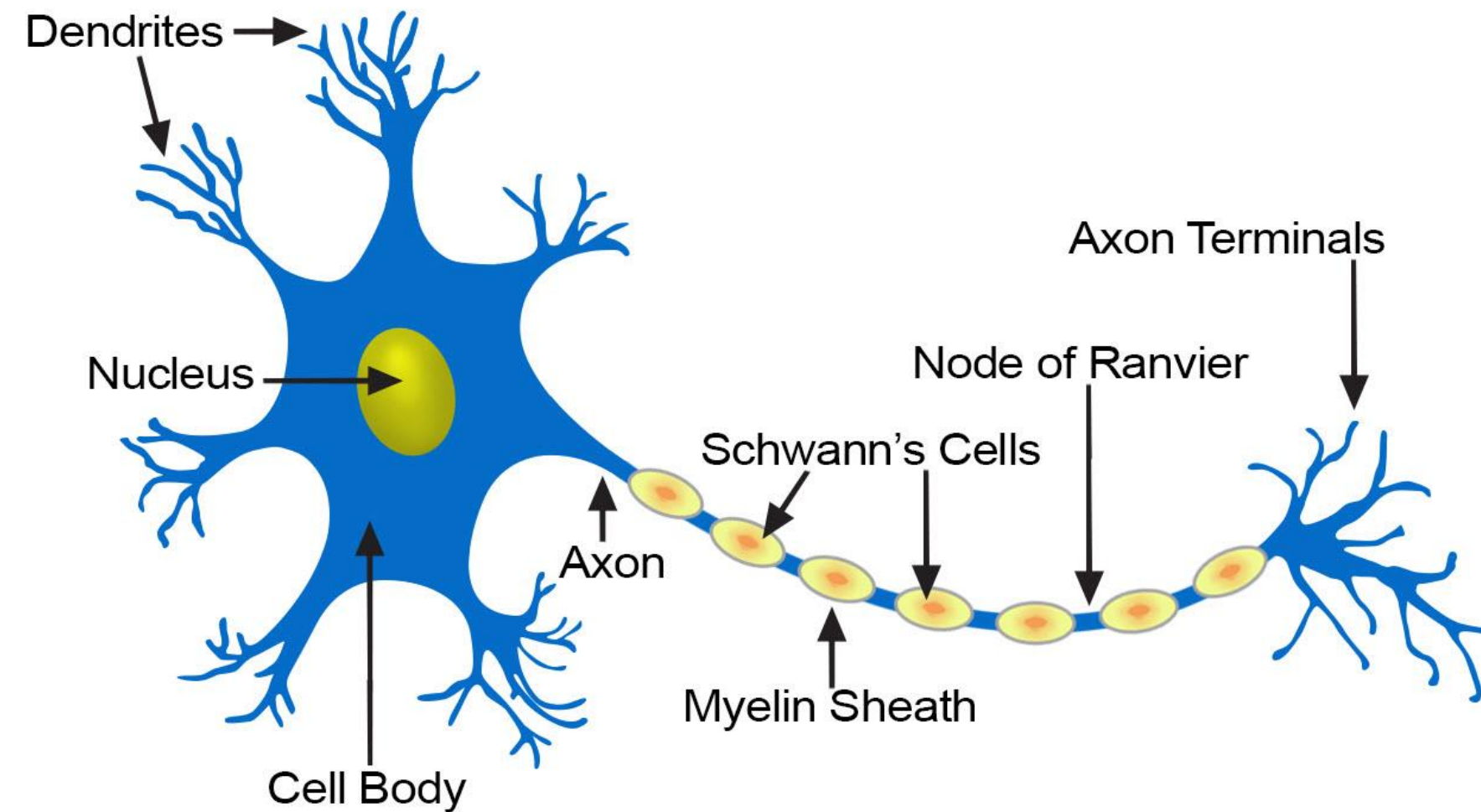
신경망



사람의 신경 시스템 (Neuron System)

사람의 신경 시스템 (Neuron)

Structure of a Typical Neuron



두뇌의 가장 작은 정보처리 단위

세포체 (cell body): 간단한 연산, 수상돌기 (dendrite): 신호 수신,

축삭 (axon): 처리 결과를 전송

사람은 10^{11} 개의 뉴런을 가지며, 뉴런은 10^3 개 가량 다른 뉴런과 연결

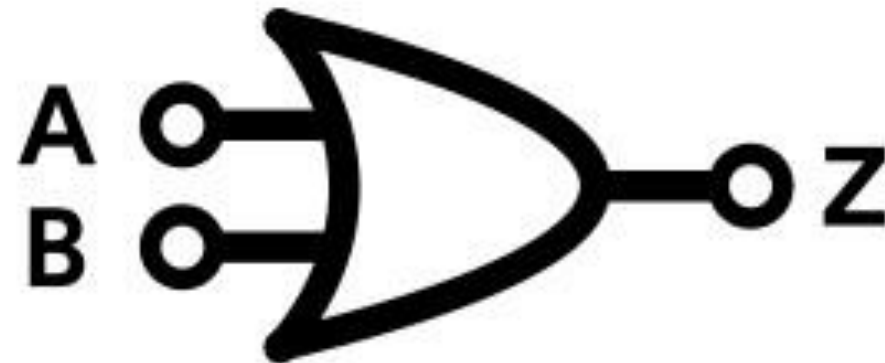
최초의 신경망

AND gate



A	B	Z
0	0	0
0	1	0
1	0	0
1	1	1

OR gate

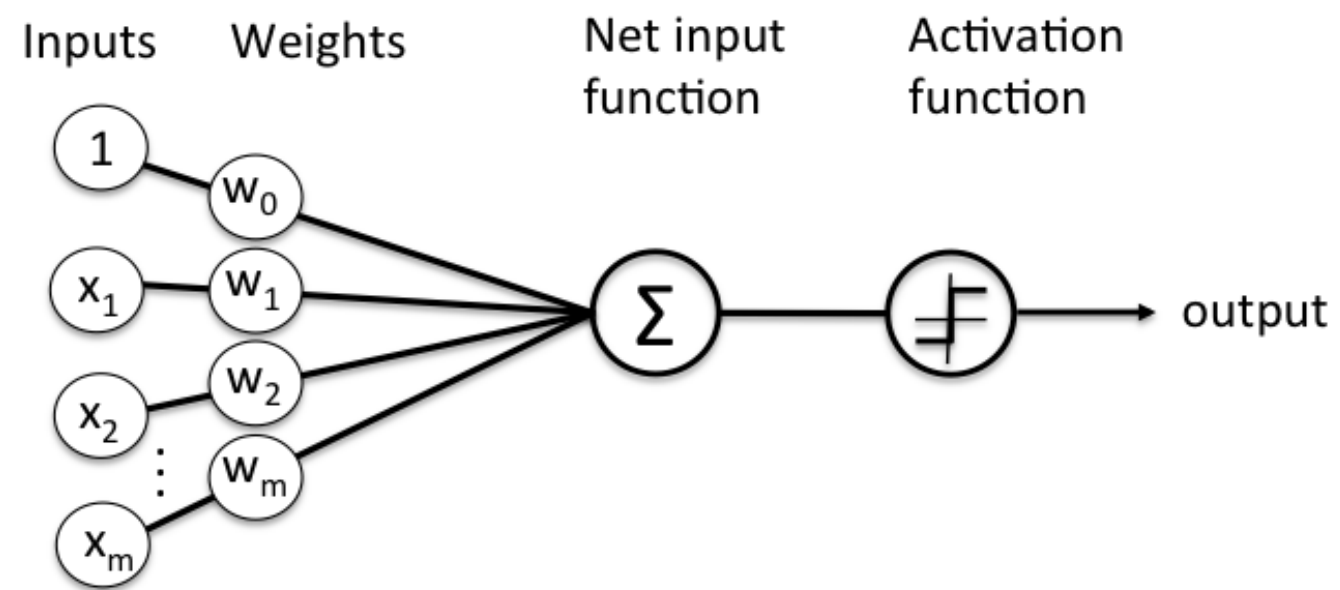


A	B	Z
0	0	0
0	1	1
1	0	1
1	1	1

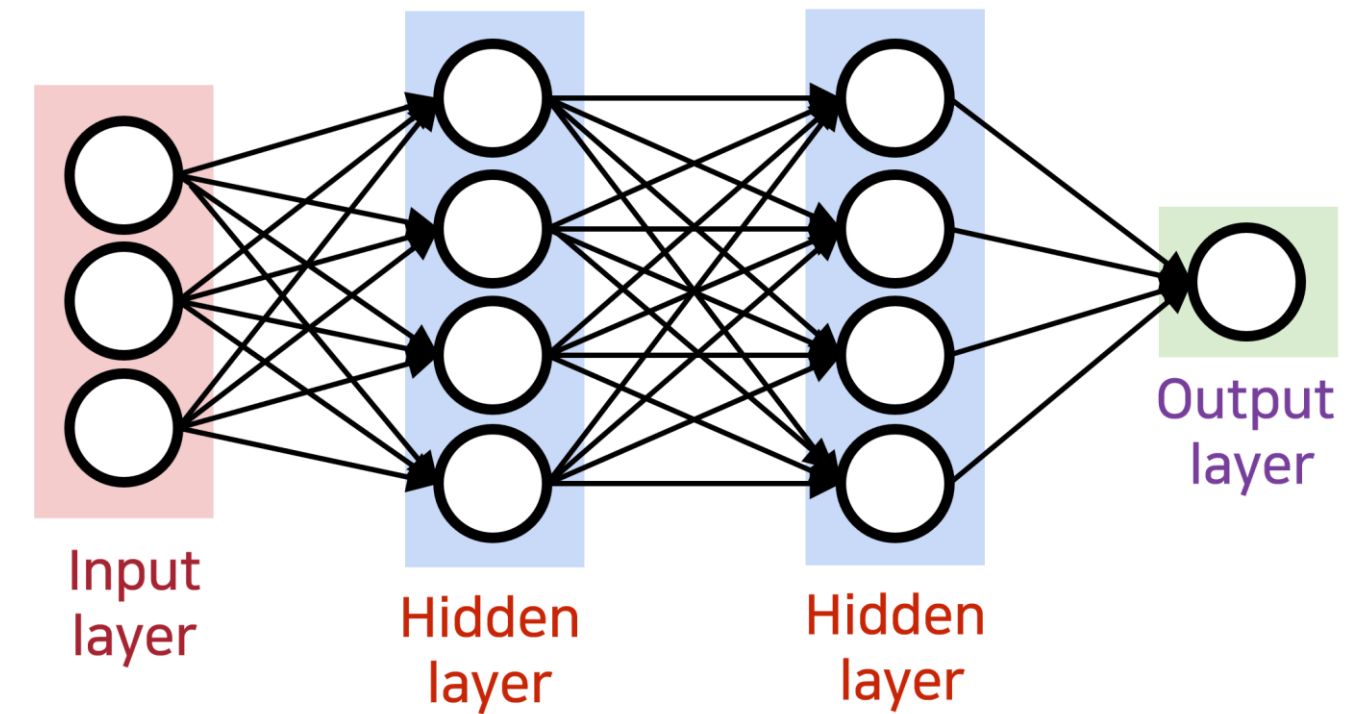
1943년 맥클락-피츠 모델 (뉴런 세포를 논리적인 값으로 모델링)

1. 뉴런은 활성화되거나 활성화되지 않는 2가지 상태
2. 뉴런 흥분, 2개 이상의 고정된 수의 시냅스가 활성화

퍼셉트론/다층 퍼셉트론



퍼셉트론

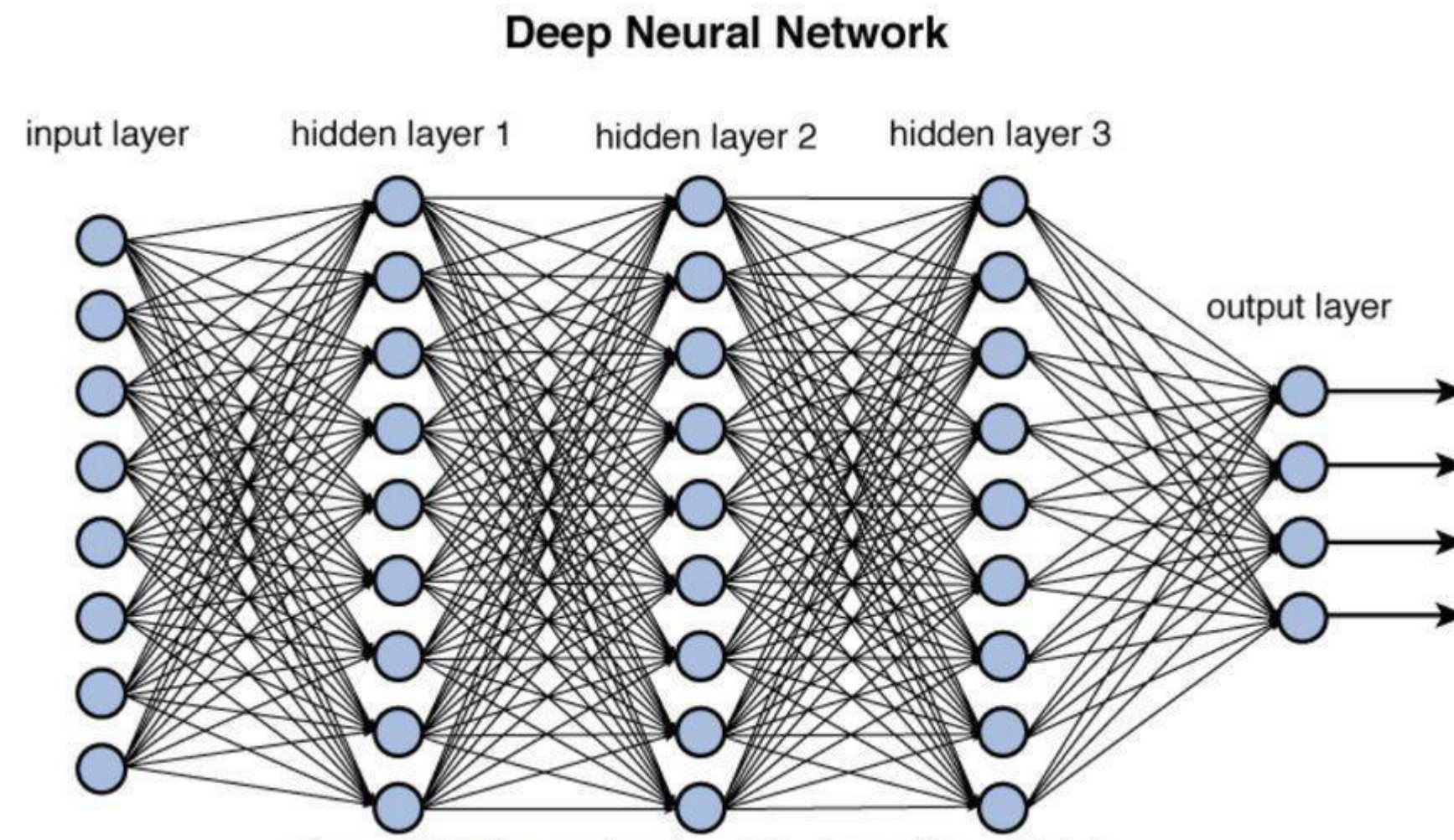


다층 퍼셉트론

1958년 퍼셉트론 (선형분류기)

Input과 Weight의 곱을 모두 더한 뒤, 활성 함수를 통해 최종 결과

딥러닝

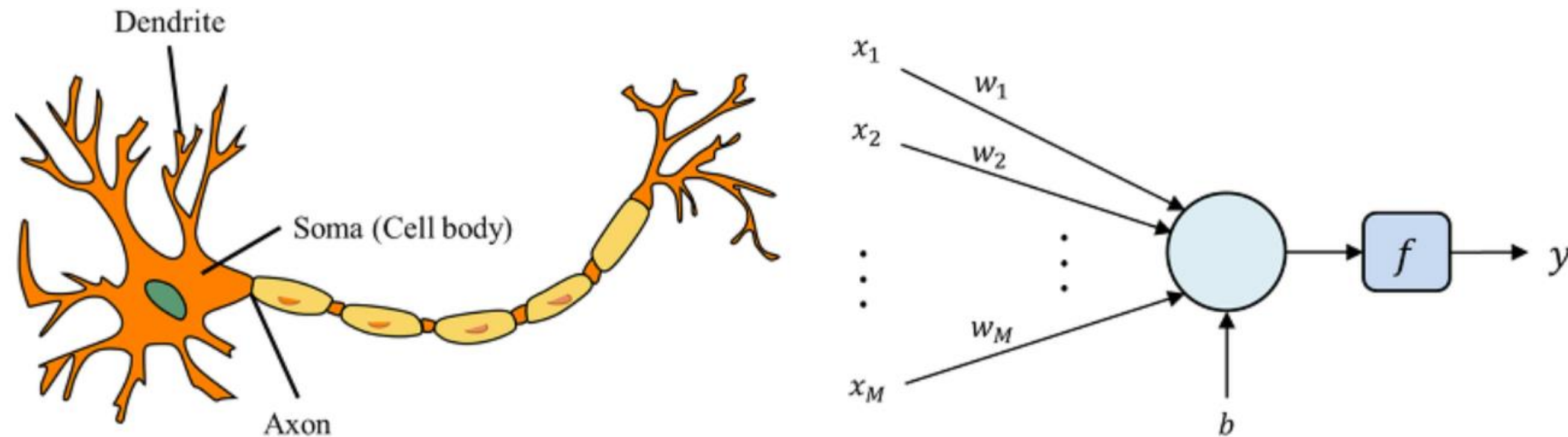


심층 신경망 (딥러닝): **망의 깊이가 깊은 신경망**
학습 알고리즘의 개선, 컴퓨팅 파워의 향상으로 구현 가능

퍼셉트론 (Perceptron)

Percept: 지각된 것; 지각[인식]의 대상;

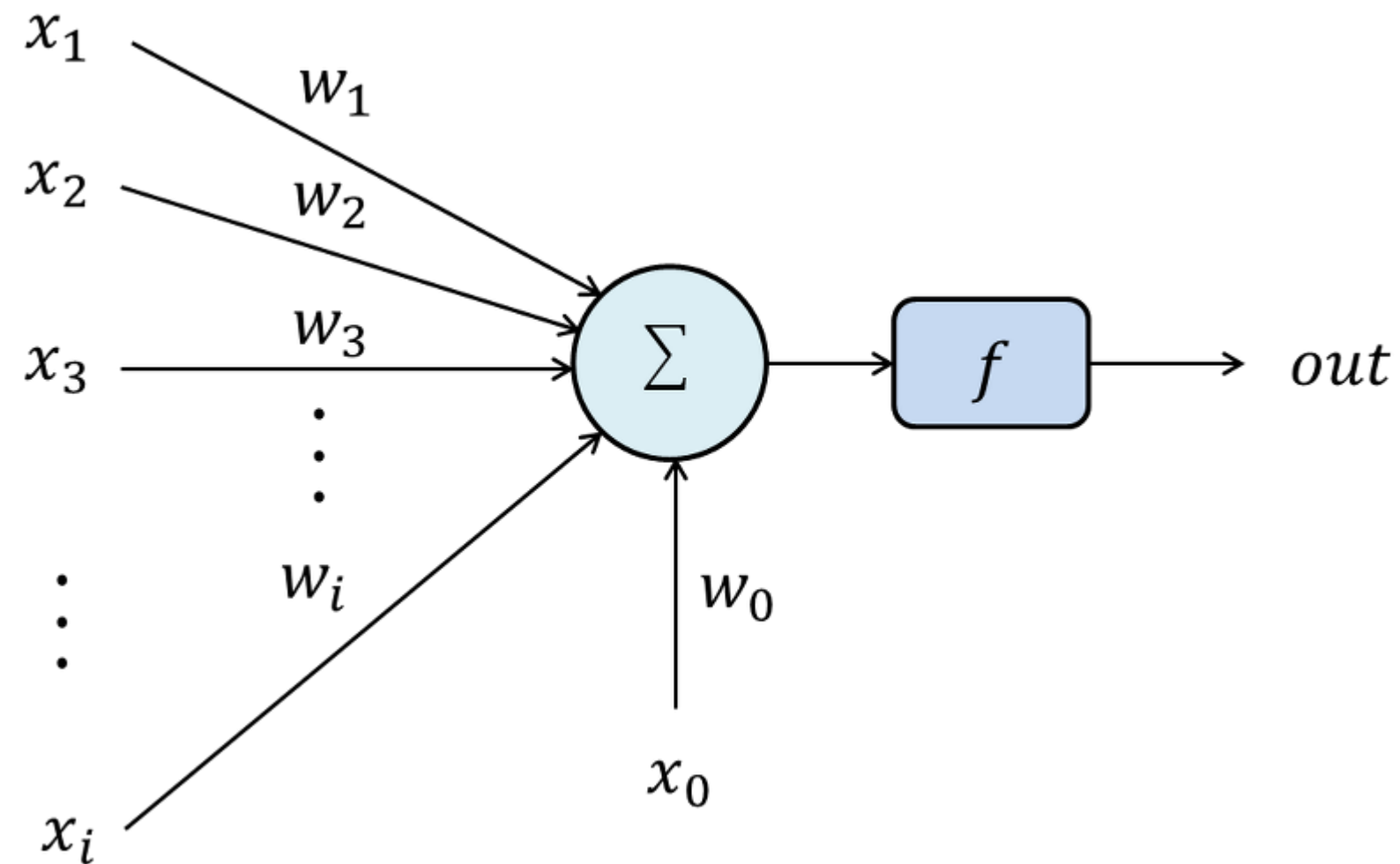
퍼셉트론 (Perceptron)



생물체의 neuron (좌)과 artificial neuron (우)

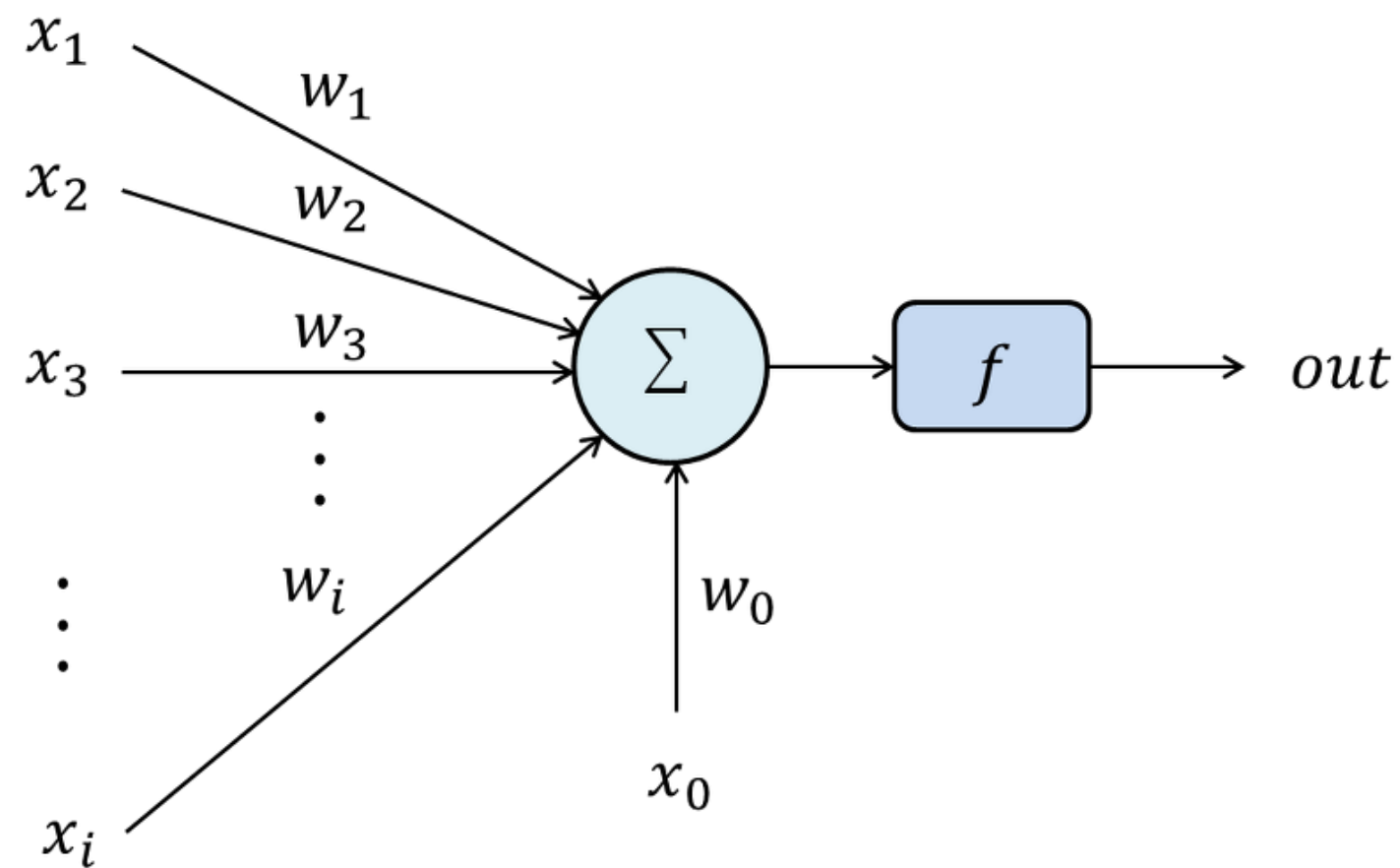
인공신경망 시스템은 **동물의 신경계 시스템을 모사**하여 설계

퍼셉트론 (Perceptron) 구조



다수의 신호 ($x_1 \sim x_i$)를 입력 받아 하나의 신호 (out)을 출력
신호를 전달하는 역할 ($w_1 \sim w_i$): 가중치 (weight)
퍼셉트론 동작 경향성 (x_0): 바이어스 (bias)
신호의 전달 여부 결정 (f): 액티베이션 함수 (activation)

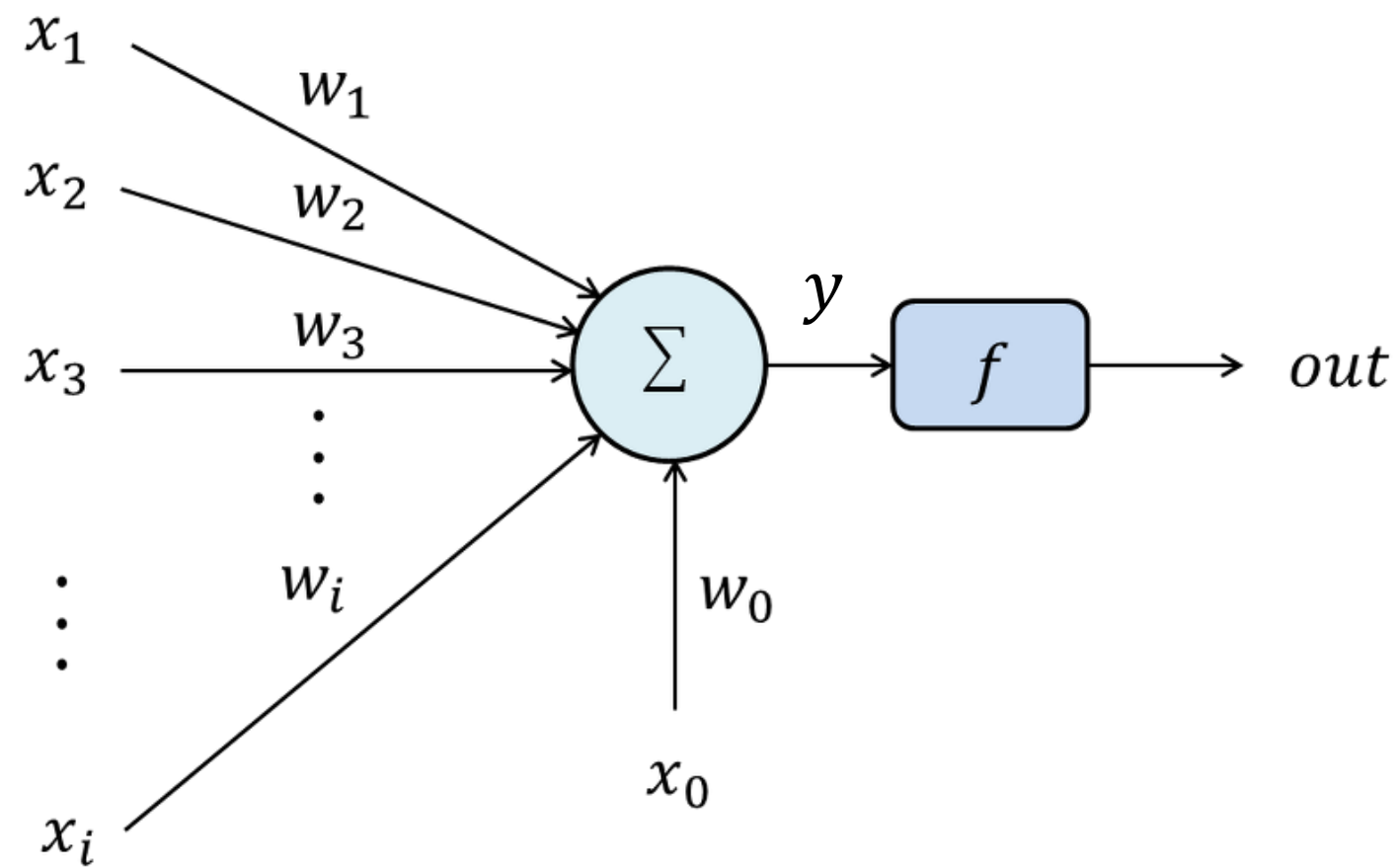
퍼셉트론 파이썬 코드



```
def perceptron (row, weights):  
    activation = weights [0]  
    for i in range(len(row)-1):  
        activation += weights[i+1]*row[i]  
    return 1.0 if activation >= 0.0 else 0.0
```

다수의 신호 ($x_1 \sim x_i$)와 가중치 (weight)의 곱셈, 덧셈
곱셈과 덧셈이 완료된 이후 **액티베이션 함수**

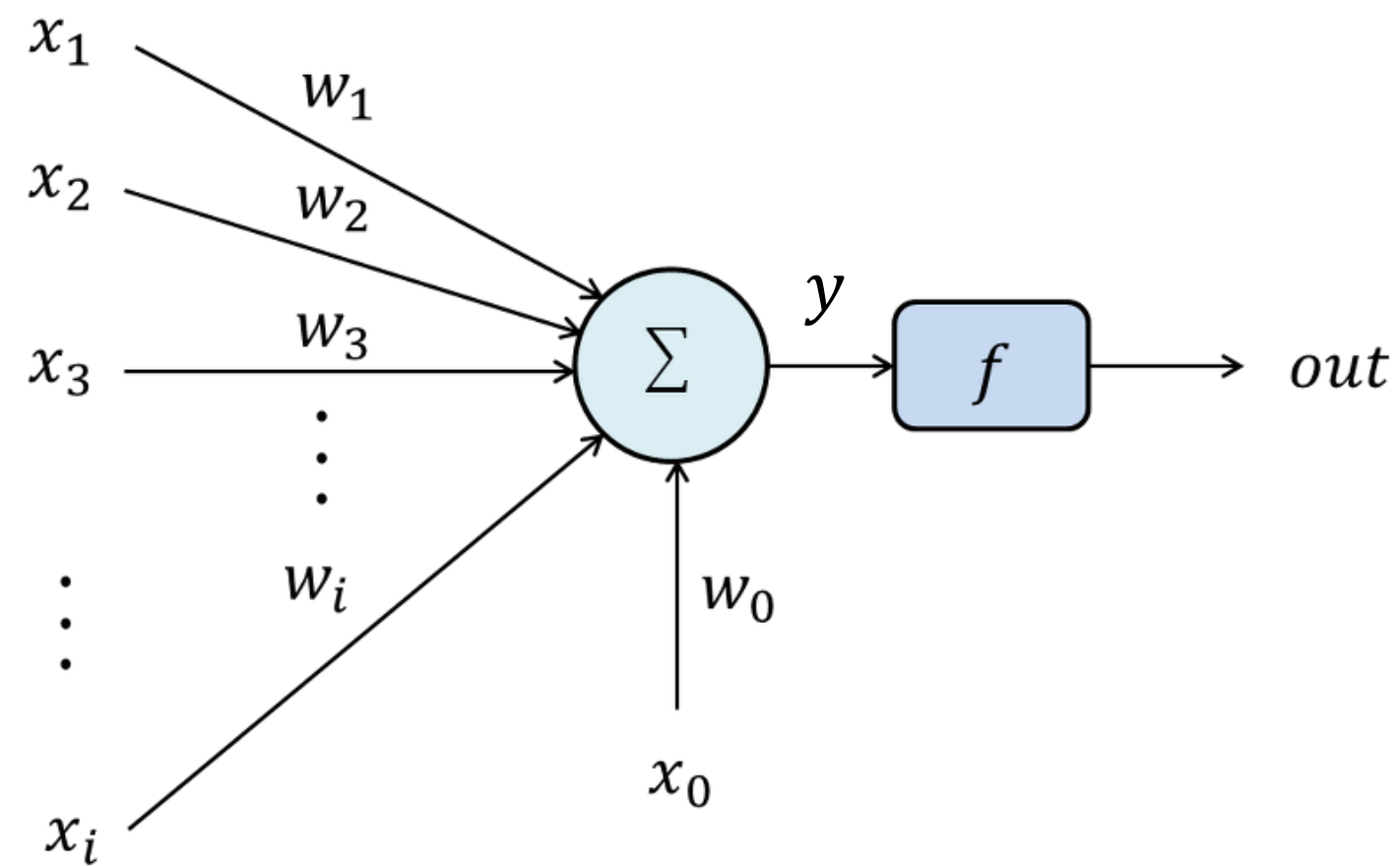
퍼셉트론 (Perceptron) 예시



x, w	의미
x_1	비가온다
x_2	여친이 만나자고 한다
w_1	비를 좋아하는 정도
w_2	여친을 좋아하는 정도
y	외출한다/안한다

액티베이션 함수: **y 값이 얼마일때** 여친을 보기 위해 외출은 하는가 ?
(1이면, 10이면, **0 보다 크면**, -1.0이면, 0.5이면)

퍼셉트론 (Perceptron) 예시



x, w, y	값	의미
x_1	1	비가온다
x_2	1	여친이 만나자고 한다
w_1	-5	비를 좋아하는 정도
w_2	6	여친을 좋아하는 정도
y		외출한다/안한다

$$y = 1 * (-5) + (1) * (6) = 1$$

난 태생적으로 비가 싫어 (bias): -2

$$y = 1 - 2 = -1$$

y 의 값이 0보다 작기 때문에, 외출 안한다

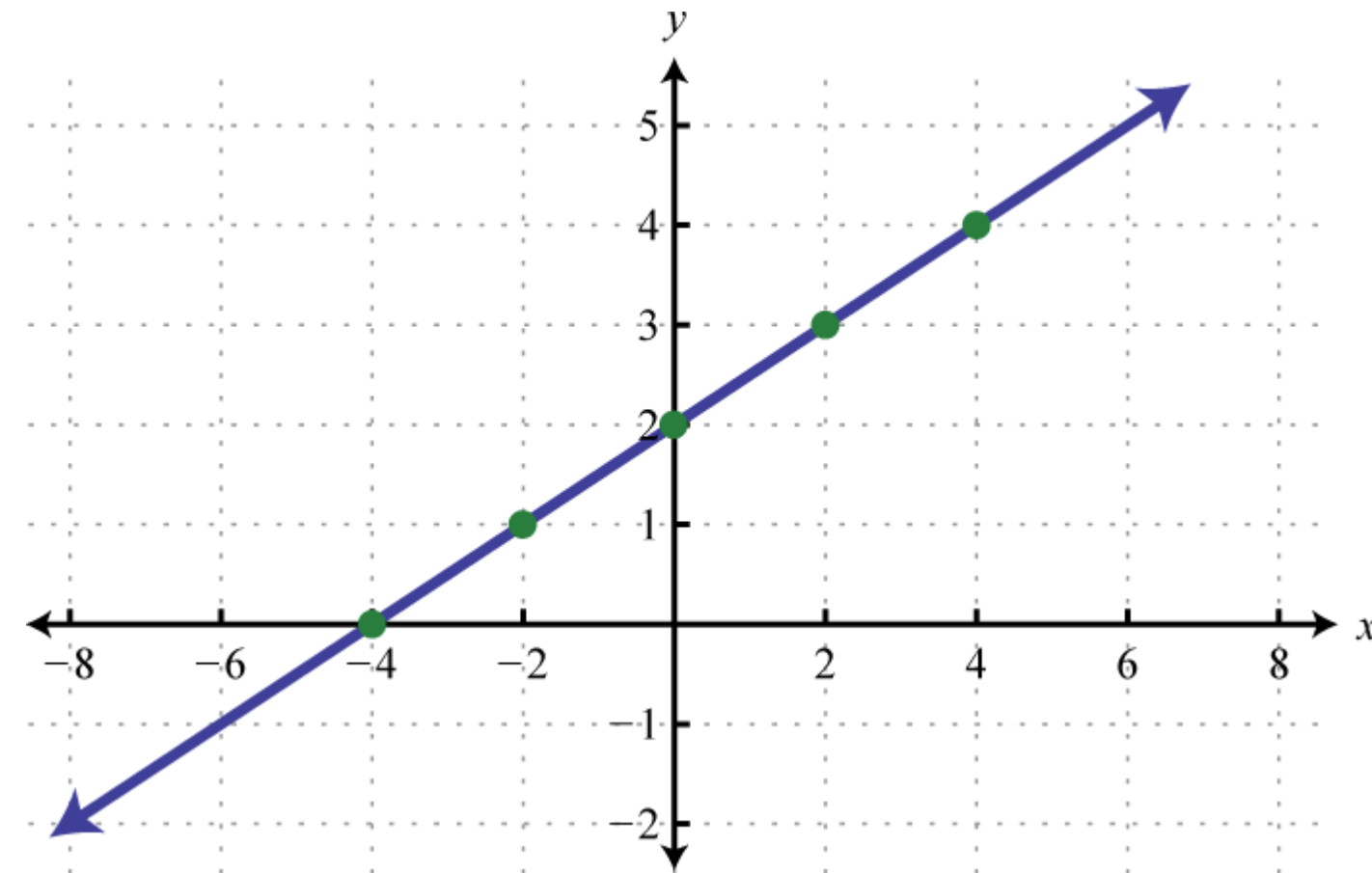
퍼셉트론 (Perceptron)

$$y = 1 * (-5) + (1) * (6) = 1$$

난 태생적으로 비가 싫어 (bias): -2

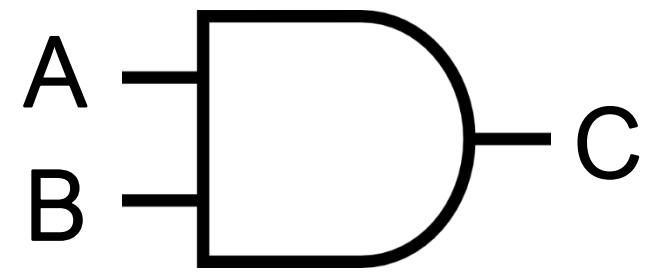
$$y = 1 - 2 = -1$$

$$y = x_1 * w_1 + x_2 * w_2 + bias$$



퍼셉트론은 **선형 방정식**으로 표현 가능함 (=선형 분류기)
퍼셉트론을 사용하여 **논리 연산 (AND, OR, NAND)** 가능

퍼셉트론 (Perceptron): 게이트



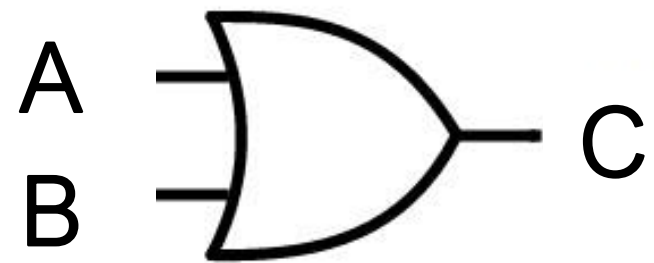
$C > threshold, 1, else 0$
 $(w1, w2, threshold) = (0.5, 0.5, 0.7)$

AND

A/B	C
0/0	0
1/0	0
0/1	0
1/1	1

$$0 * 0.5 + 0.5 * 1 < 0.7$$

$$1 * 0.5 + 0.5 * 1 > 0.7$$



$C > threshold, 1, else 0$
 $(w1, w2, threshold) = (0.5, 0.5, 0.2)$

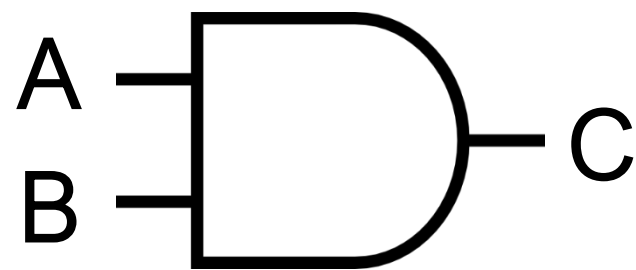
OR

A/B	C
0/0	0
1/0	1
0/1	1
1/1	1

$$0 * 0.5 + 0.5 * 1 > 0.2$$

$$1 * 0.5 + 0.5 * 0 > 0.2$$

퍼셉트론 (Perceptron): AND

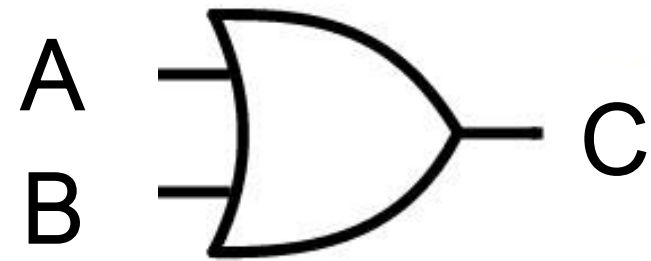


$C > \text{threshold}, 1, \text{else } 0$
 $(w1, w2, \text{threshold}) = (0.5, 0.5, 0.7)$

AND

```
def AND(x1, x2):  
    x = np.array([x1,x2])  
    w = np.array([0.5,0.5])  
    theta = 0.7  
    if np.sum(w*x) <= theta:  
        return 0  
    else:  
        return 1  
  
inputData = np.array([[0,0],[1,0],[0,1],[1,1]])  
for x in inputData:  
    print(x[0],",", x[1], " ==> ",AND(x[0],x[1]), sep = " ")
```

퍼셉트론 (Perceptron): OR

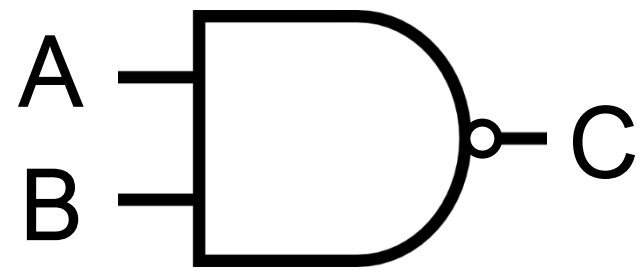


$C > \text{threshold}, 1, \text{else } 0$
 $(w1, w2, \text{threshold}) = (0.5, 0.5, 0.2)$

OR

```
def OR(x1, x2):  
    x = np.array([x1,x2])  
    w = np.array([0.5,0.5])  
    theta = 0.2  
    if np.sum(w*x) <= theta:  
        return 0  
    else:  
        return 1  
  
inputData = np.array([[0,0],[1,0],[0,1],[1,1]])  
for x in inputData:  
    print(x[0],", ",x[1]," ==> ",OR(x[0],x[1]), sep = " ")
```

퍼셉트론 (Perceptron): NAND

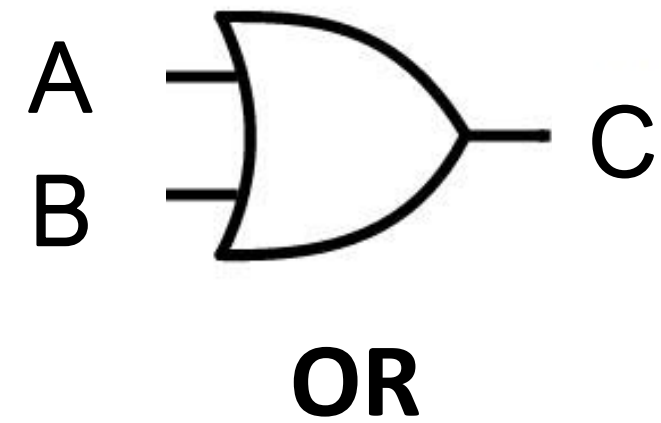


$C > \text{threshold}, 1, \text{else } 0$
 $(w1, w2, \text{threshold}) = (-0.5, -0.5, -0.7)$

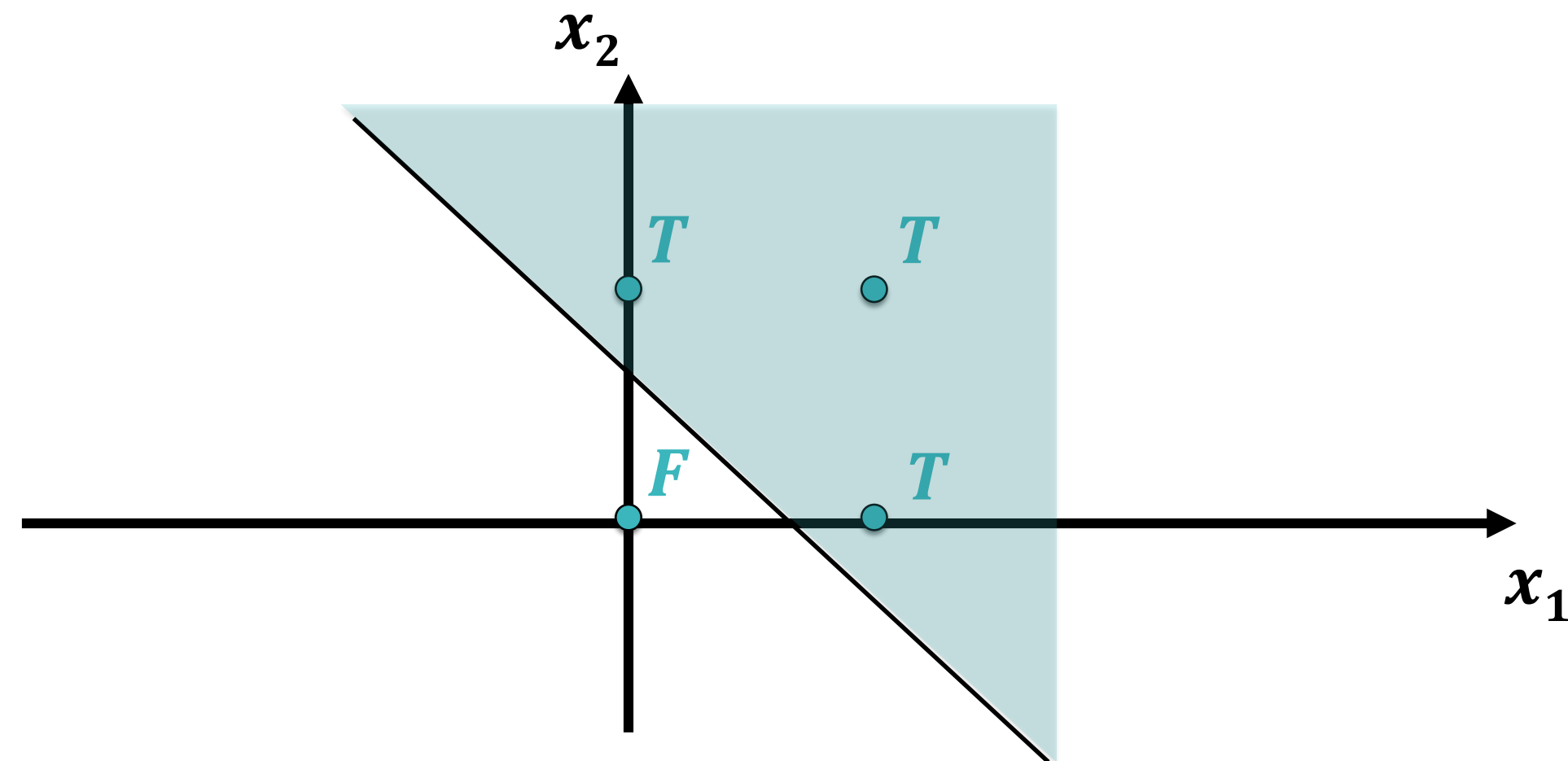
NAND

```
def NAND(x1, x2):  
    x = np.array([x1,x2])  
    w = np.array([-0.5,-0.5])  
    theta = -0.7  
    if np.sum(w*x) <= theta:  
        return 0  
    else:  
        return 1  
  
inputData = np.array([[0,0],[1,0],[0,1],[1,1]])  
for x in inputData:  
    print(x[0],", ",x[1]," ==> ",NAND(x[0],x[1]), sep = " ")
```

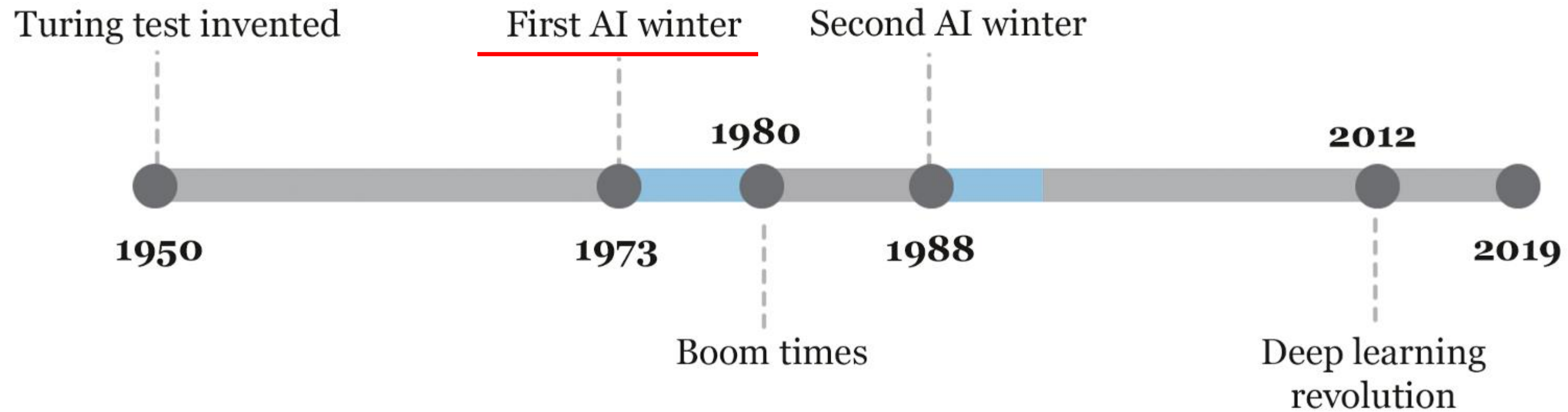
퍼셉트론 (Perceptron): OR게이트



A/B	C
0/0	0
1/0	1
0/1	1
1/1	1

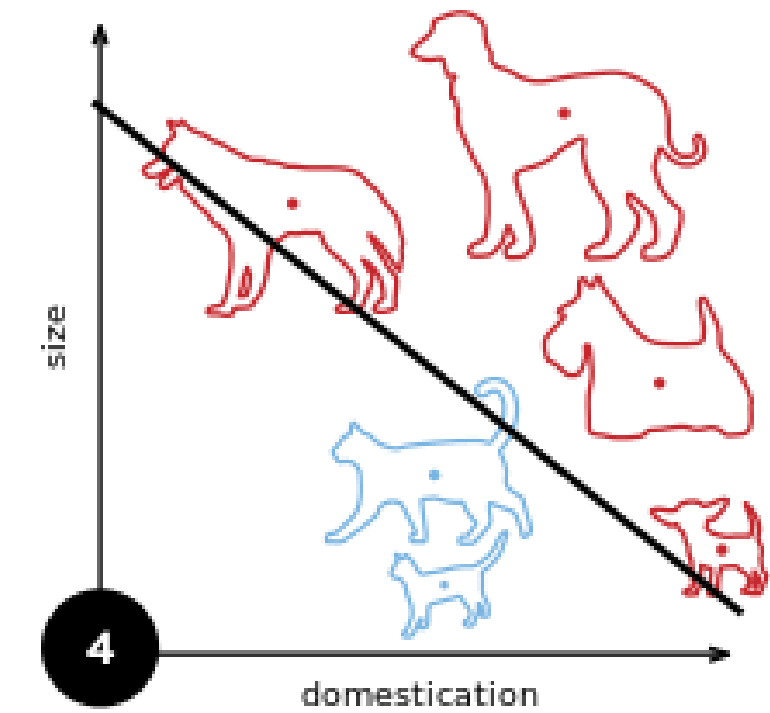
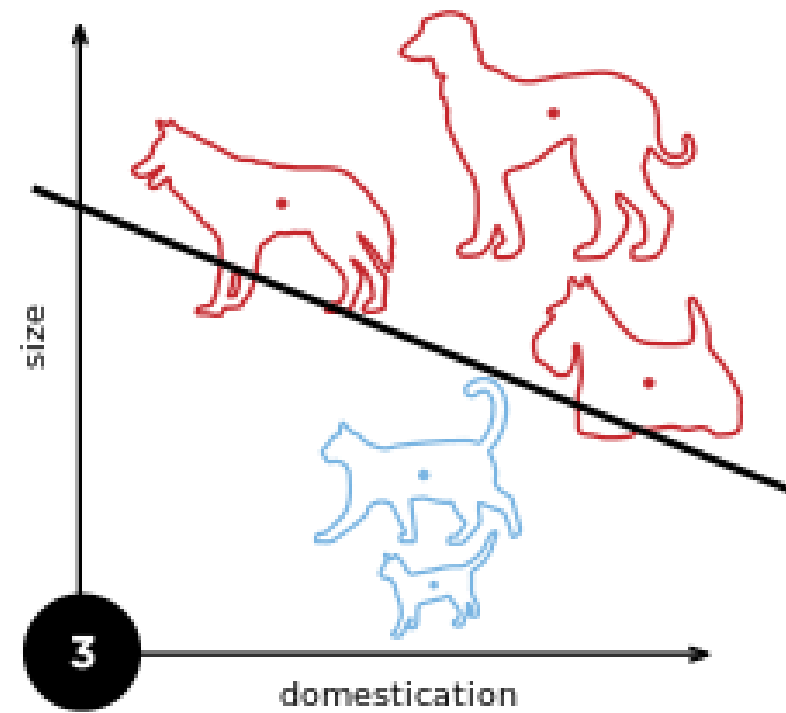
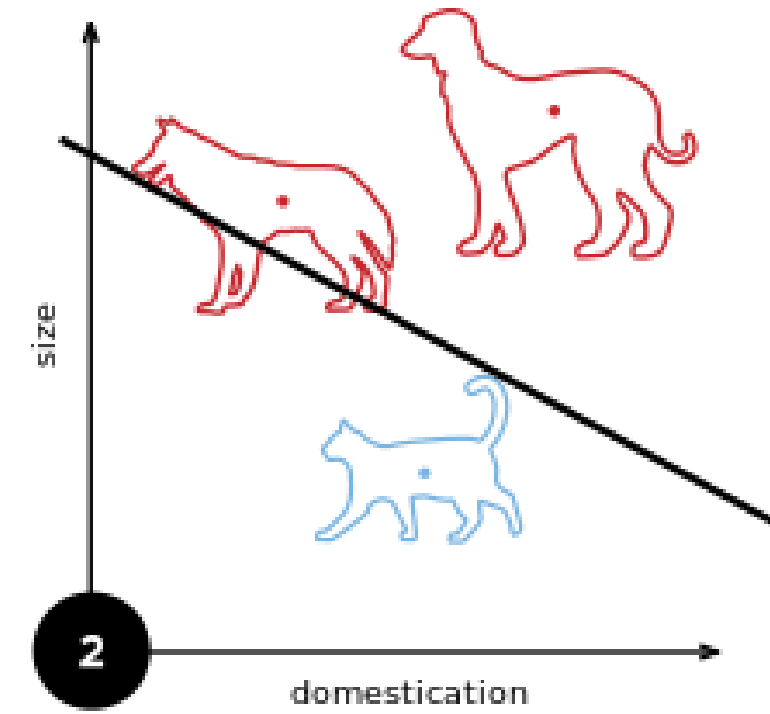
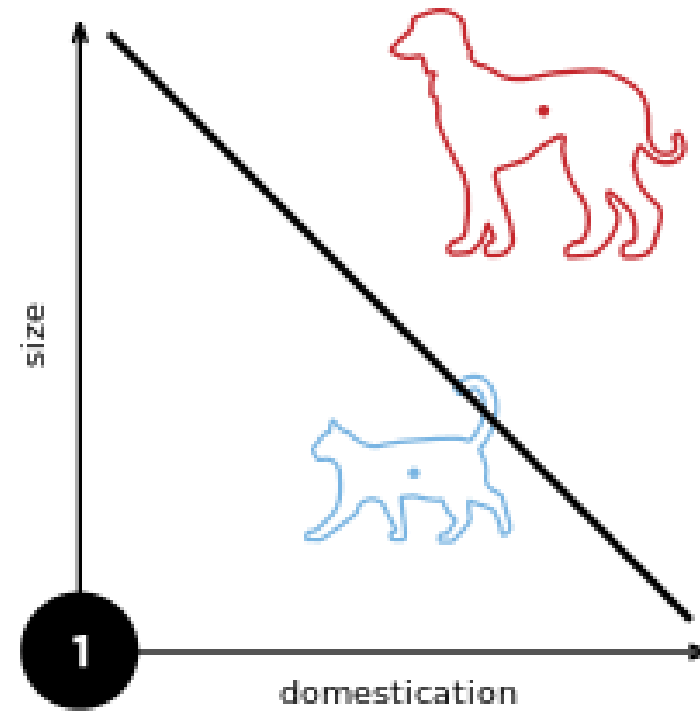


1차 AI 겨울

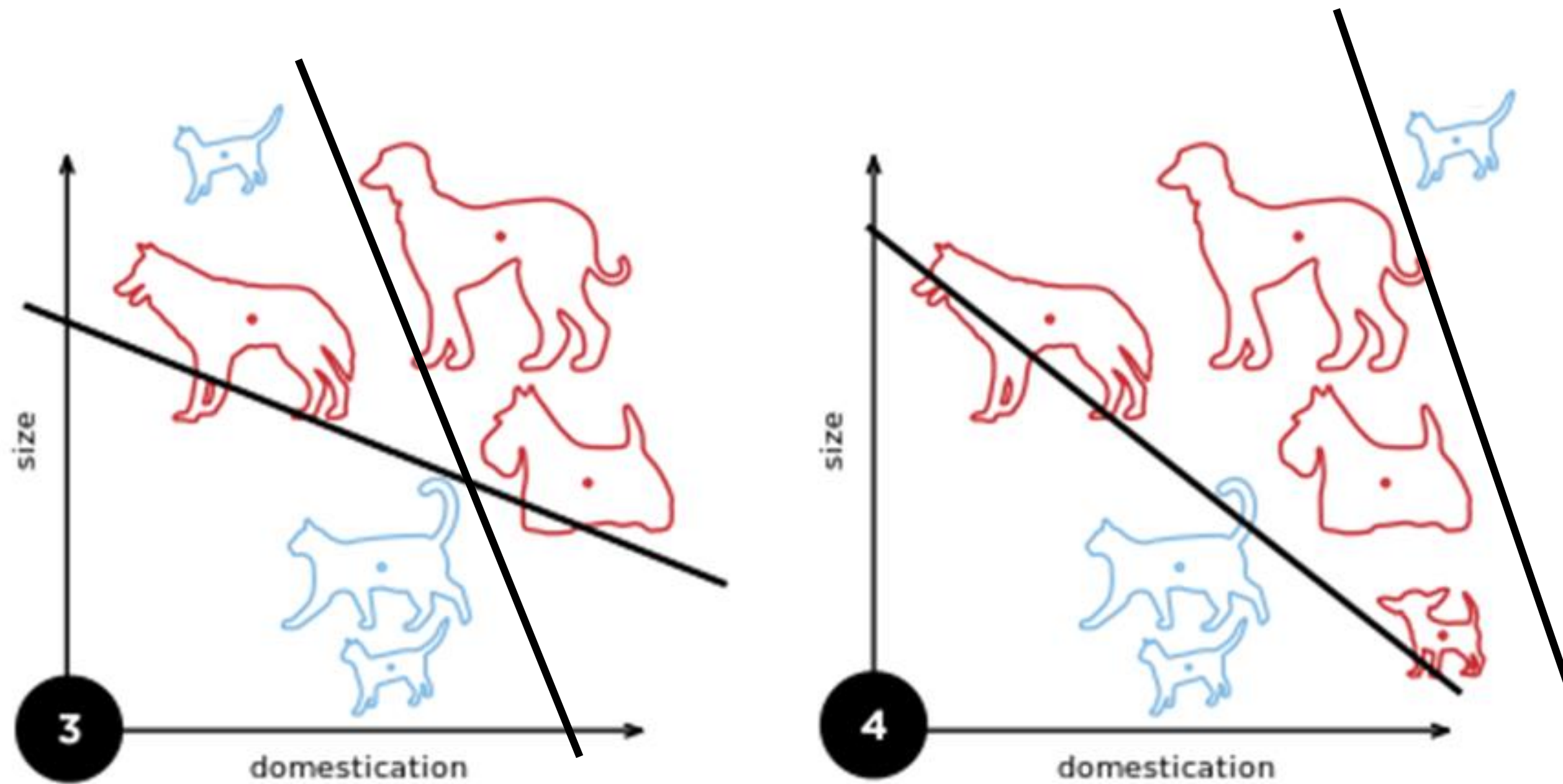


그렇다면 비 선형적인 문제도 해결이 가능할까 ?

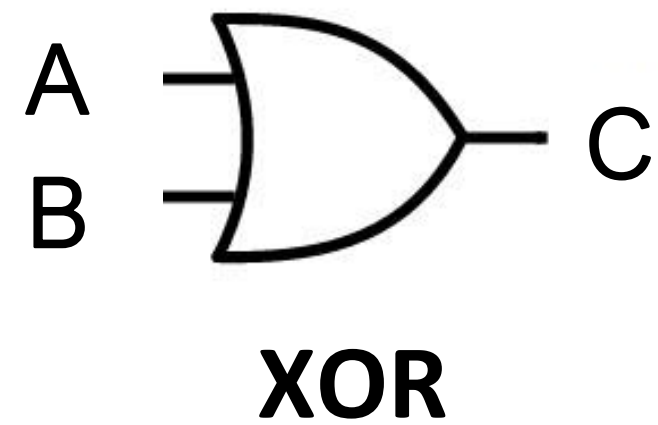
선형적인 문제는 ?



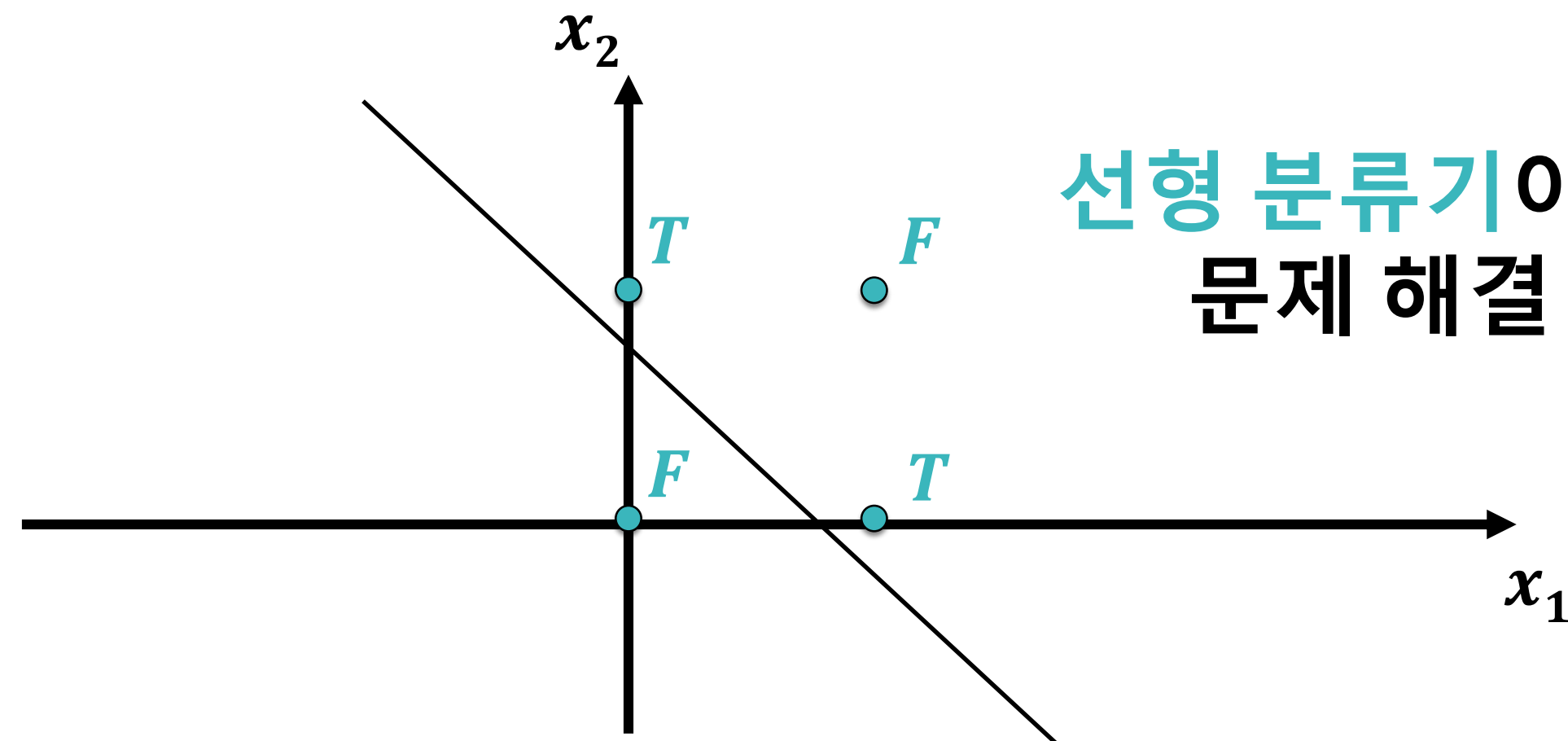
비 선형적인 문제는 ?



비 선형적인 문제

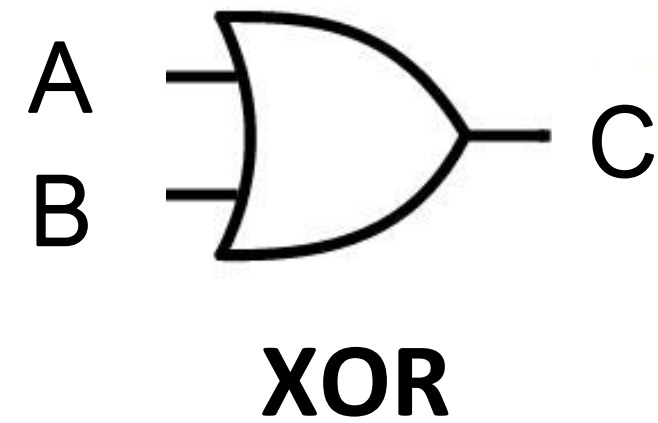


A/B	C
0/0	0
1/0	1
0/1	1
1/1	0

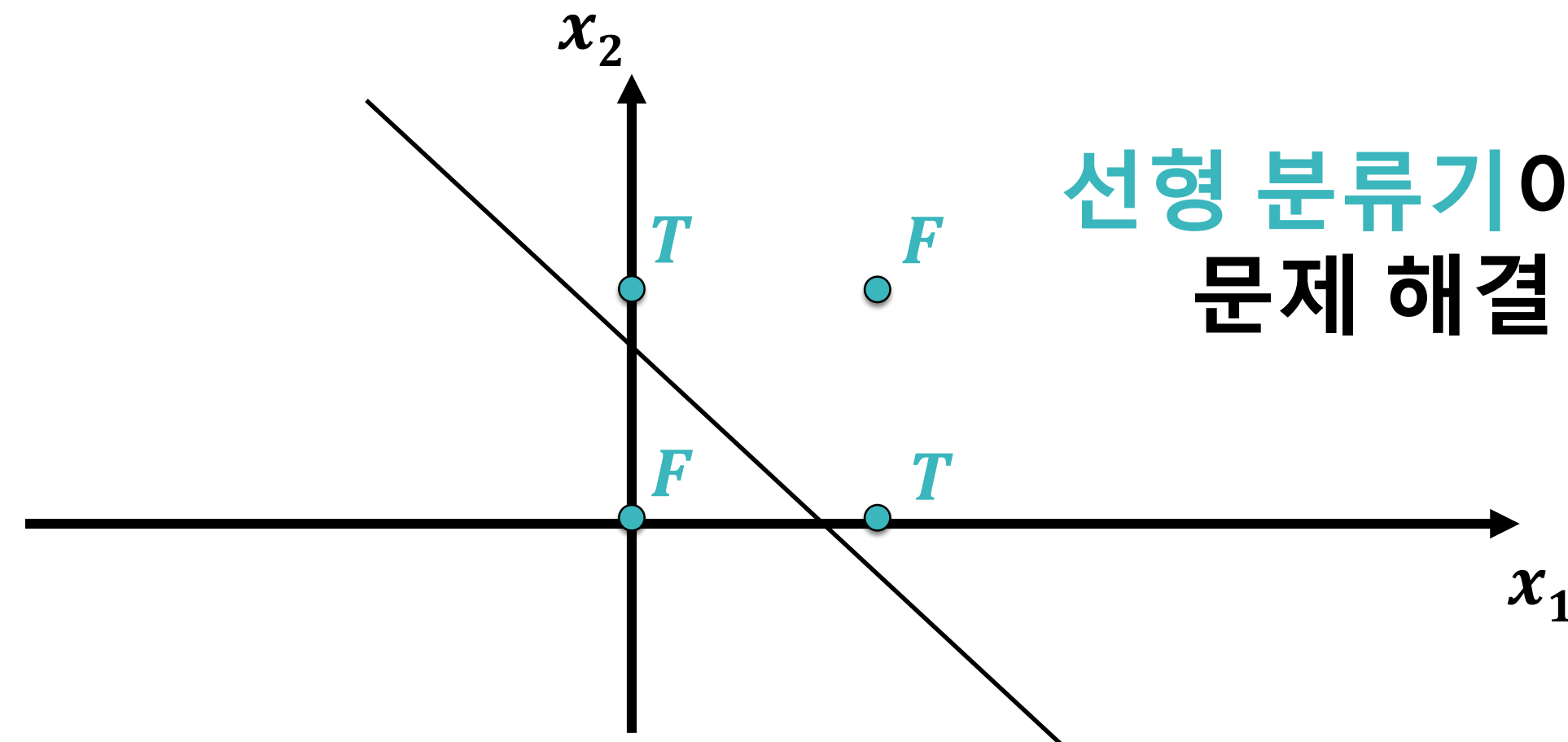


선형 분류기이기 때문에
문제 해결 불가능

퍼셉트론 (Perceptron): XOR게이트

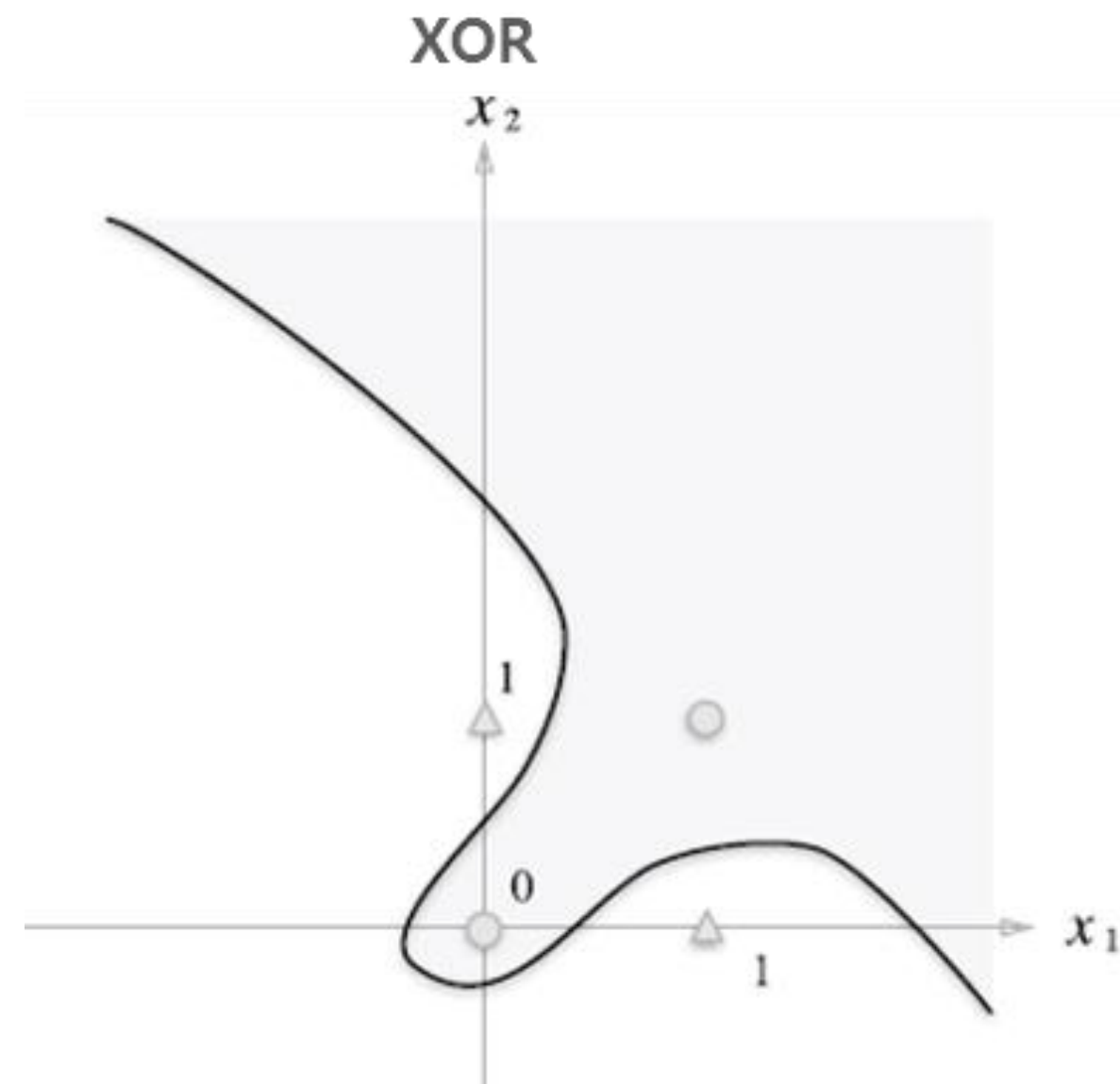


A/B	C
0/0	0
1/0	1
0/1	1
1/1	0



선형 분류기이기 때문에
문제 해결 불가능

퍼셉트론 (Perceptron) 문제점

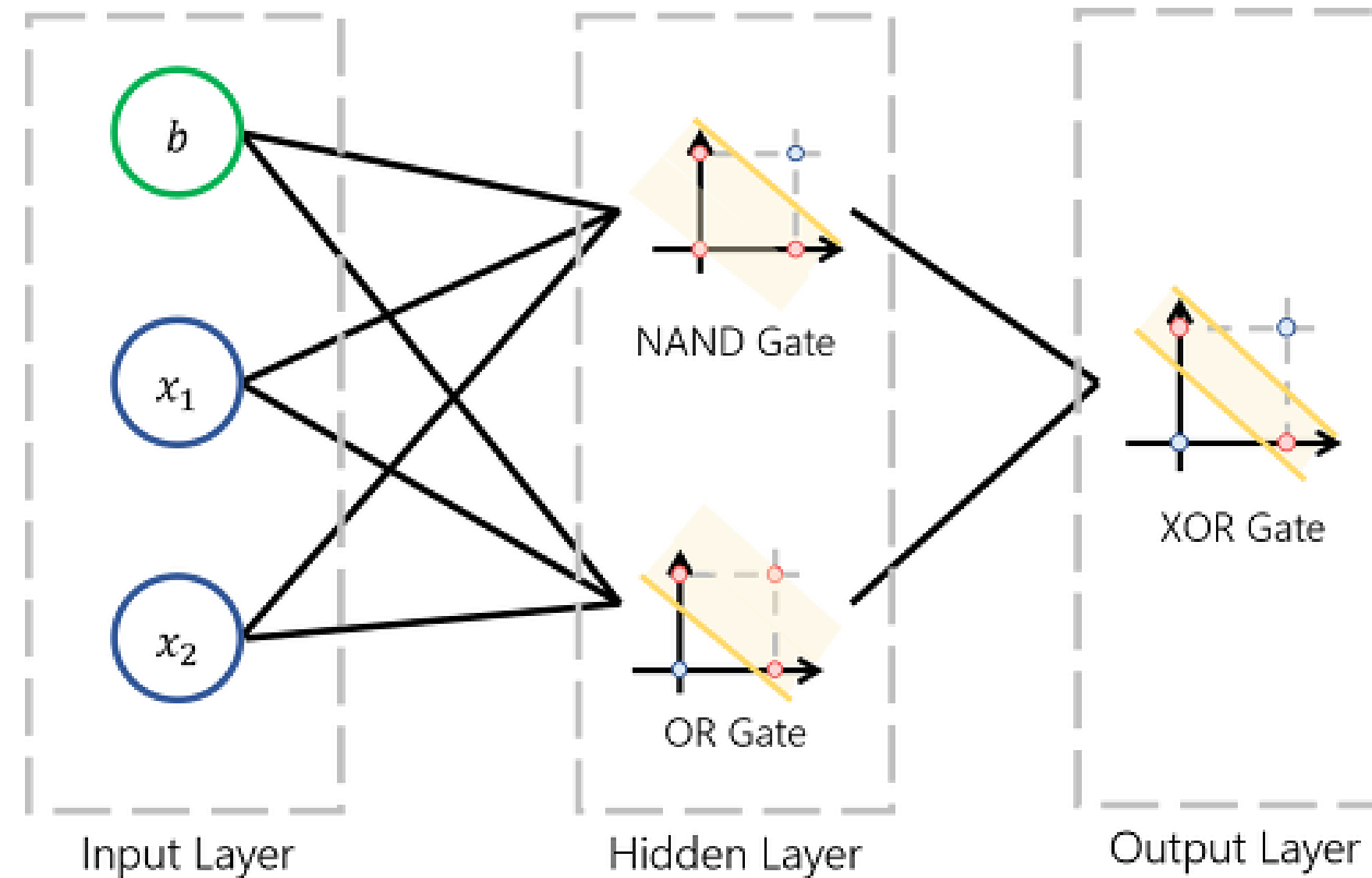


단층 퍼셉트론은 XOR 문제 뿐만 아니라 다양한 문제의 해결 불가능
선형적인 특성을 벗어난 **비선형 적인 접근 방법** 필요 !

다층 퍼셉트론 (Multi Layer Perceptron)

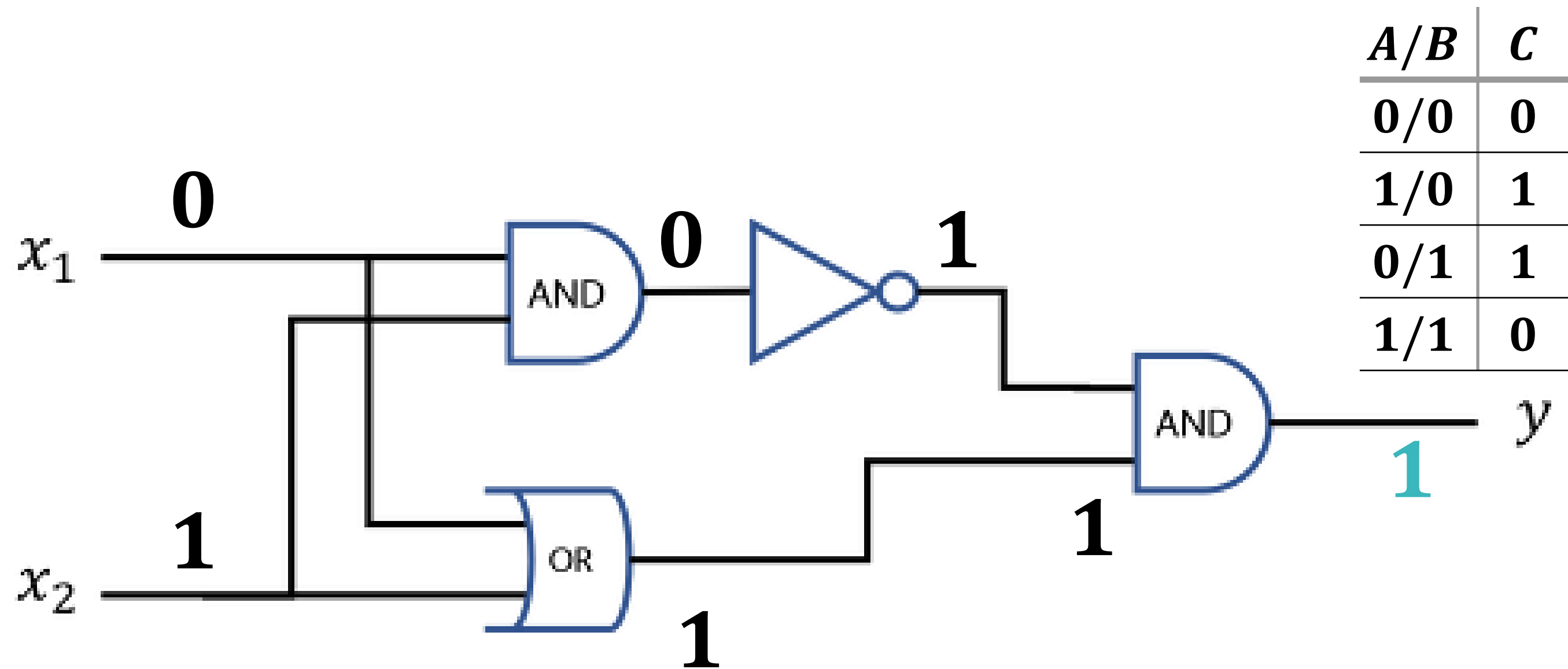
더 많은 문제를 해결하는 방법

퍼셉트론을 여러 개 쌓는 다면 ?



XOR 연산은 하나의 레이어를 사용하여 표현하는 것은 불가능
하지만, **NAND와 OR 연산**을 사용하여 표현 가능

XOR = NAND + OR + AND



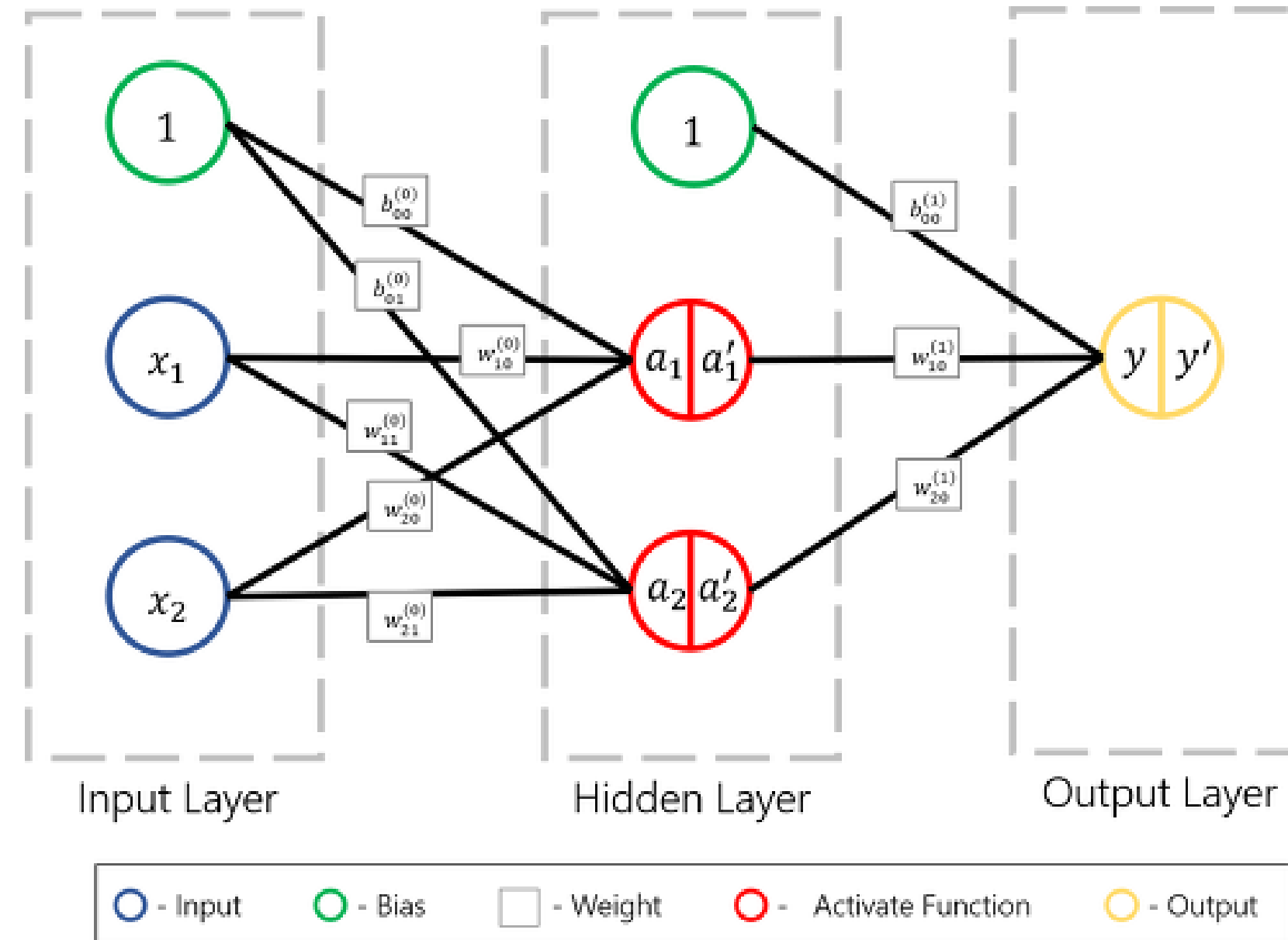
여러 개의 퍼셉트론을 조합하면 XOR 게이트를 만들 수 있음

퍼셉트론 (Perceptron): XOR

```
def XOR(x1, x2):  
    return AND(NAND(x1, x2), OR(x1, x2))  
  
inputData = np.array([[0,0],[1,0],[0,1],[1,1]])  
  
for x in inputData:  
    print(x[0],", ",x[1]," ==> ",XOR(x[0],x[1]), sep = ")
```

여러 개의 퍼셉트론을 조합하면 XOR 게이트를 만들 수 있음

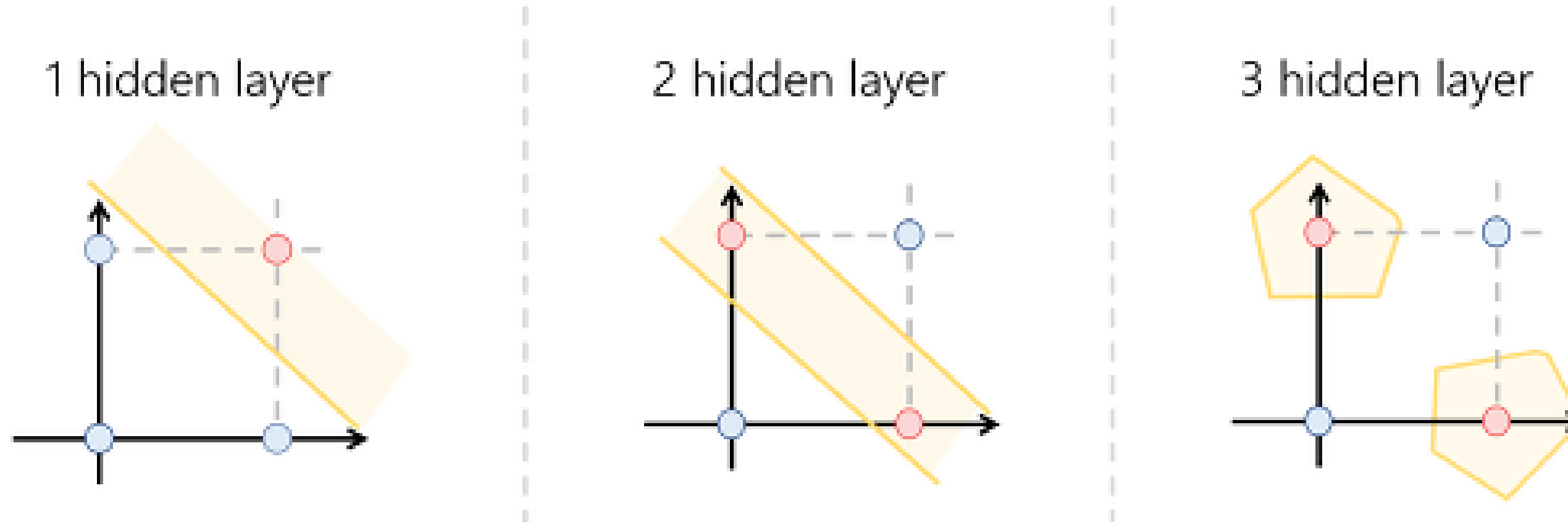
Multi Layer Perceptron (MLP)



처음 값 (Input), 결과 출력 (Output), **중간 부분 (Hidden)**
Hidden Layer가 3층 이상 되면 **Deep NN (DNN)**

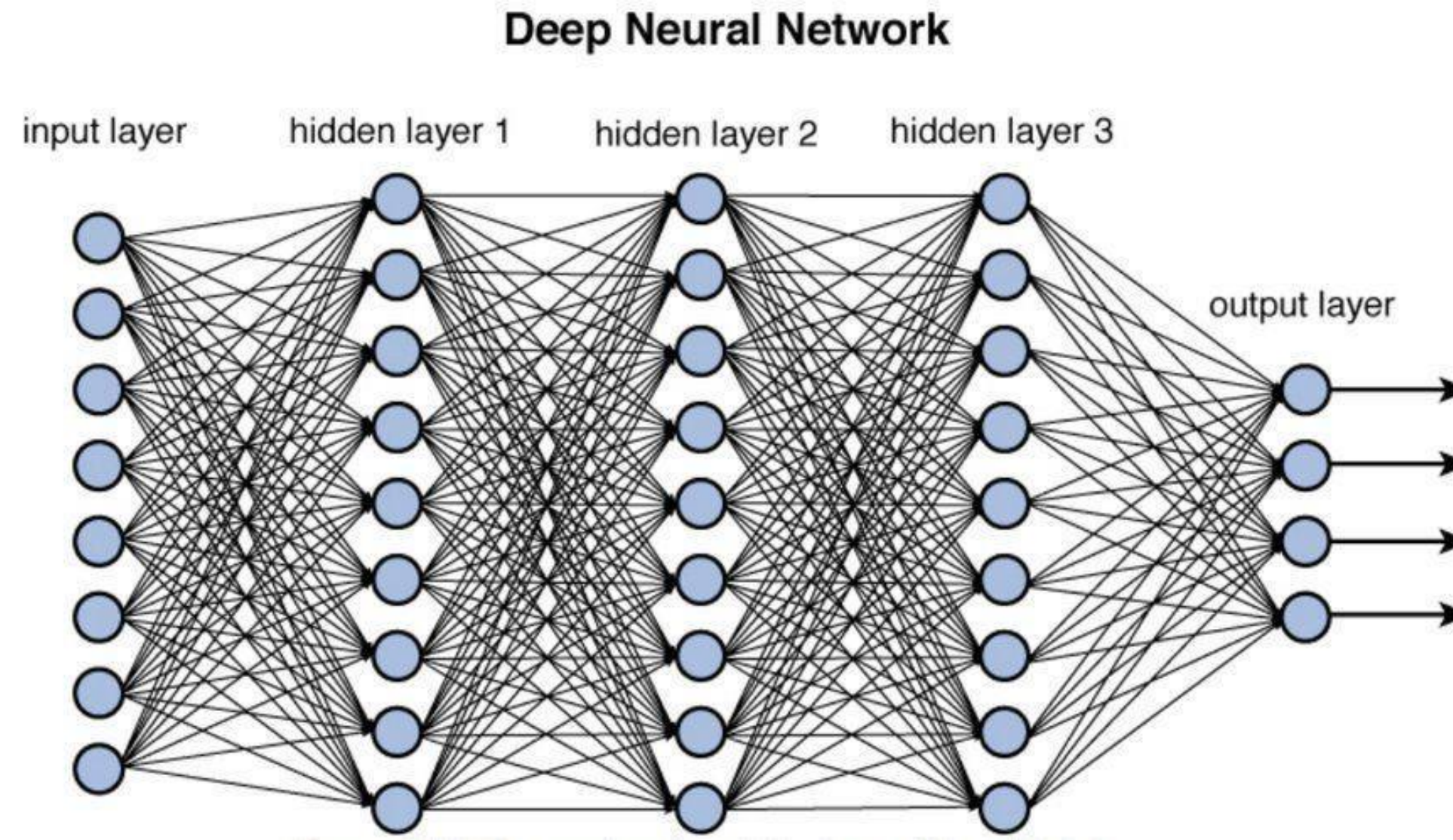
MLP가 결정할 수 있는 영역

Layer 의 개수에 따라 결정할 수 있는 영역



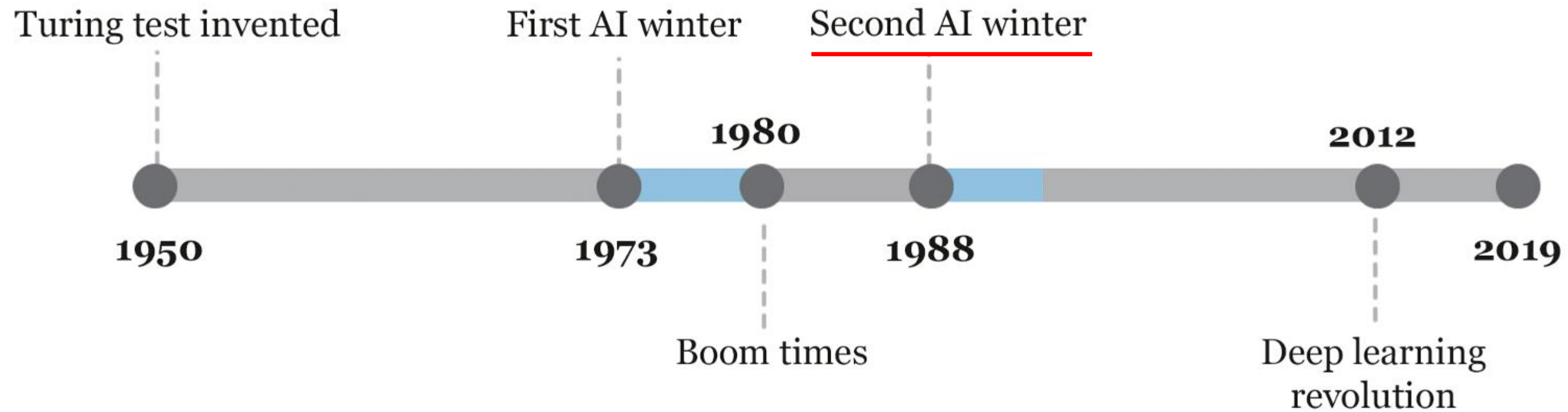
1층일 경우는 선형분리만 가능, 2층은 구역분리 가능
3층은 더 세분화된 분리 가능

망의 깊이가 깊어진다면 ?



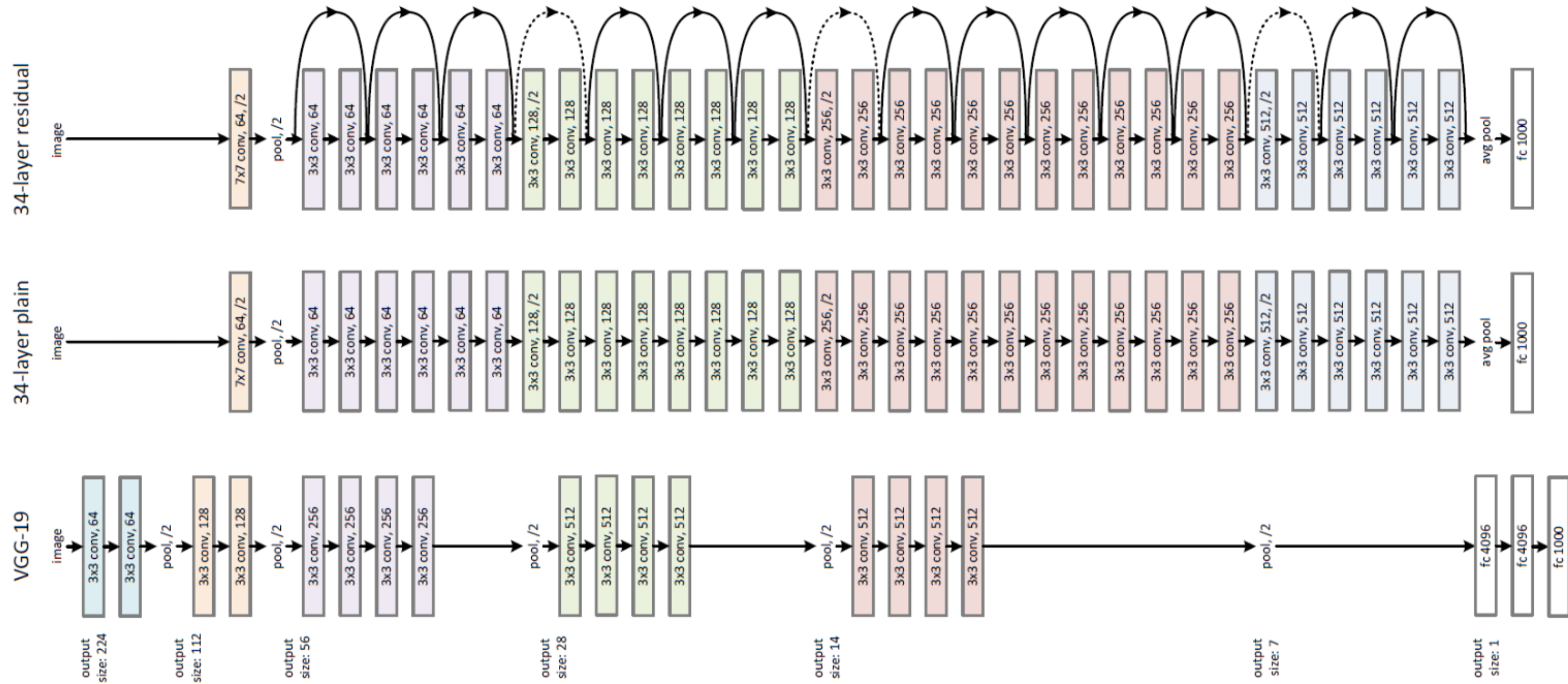
층수가 깊어지고 한 층의 노드 수가 많아진다면, **Parameter 증가**
Parameter: 가중치 (weight), 바이어스 (bias)

2차 AI 겨울



층의 깊이가 깊어지는 만큼 증가하는 **계산량**
깊이가 깊어지는 만큼 증가하는 변수는 **학습의 수렴 효과** 감소

AI 겨울이 끝난 이후에는 ?



망의 깊이가 152 이상의 신경망의 학습 가능

`/* elice */`

문의 및 연락처

academy.elice.io

contact@elice.io

facebook.com/elice.io

medium.com/elice