

# #1. A.I. & Deep Learning

ICEC 2017 - Deep Learning Short Course  
Kim Jin Ho

# Definition of A.I.

AI is that activity devoted to making machines intelligent, and intelligence is that quality that enables to function appropriately and with foresight in its environments - Nill J. Nilsson (2010)

- AI는 “합리적”으로 “행동”하고, 합리적으로 환경에 반응하는 기계를 만드는 것 (Rusell, Norvig, 1995)
- 불확실성하에서 기대되는 목표를 시간에 걸쳐 최적화 하는 것

“모든 데이터”에서 발생하는 불확실성 하에서 기대되는 목표를 시간에 걸쳐 최적화 시키는 것

# 가장 일반적인?

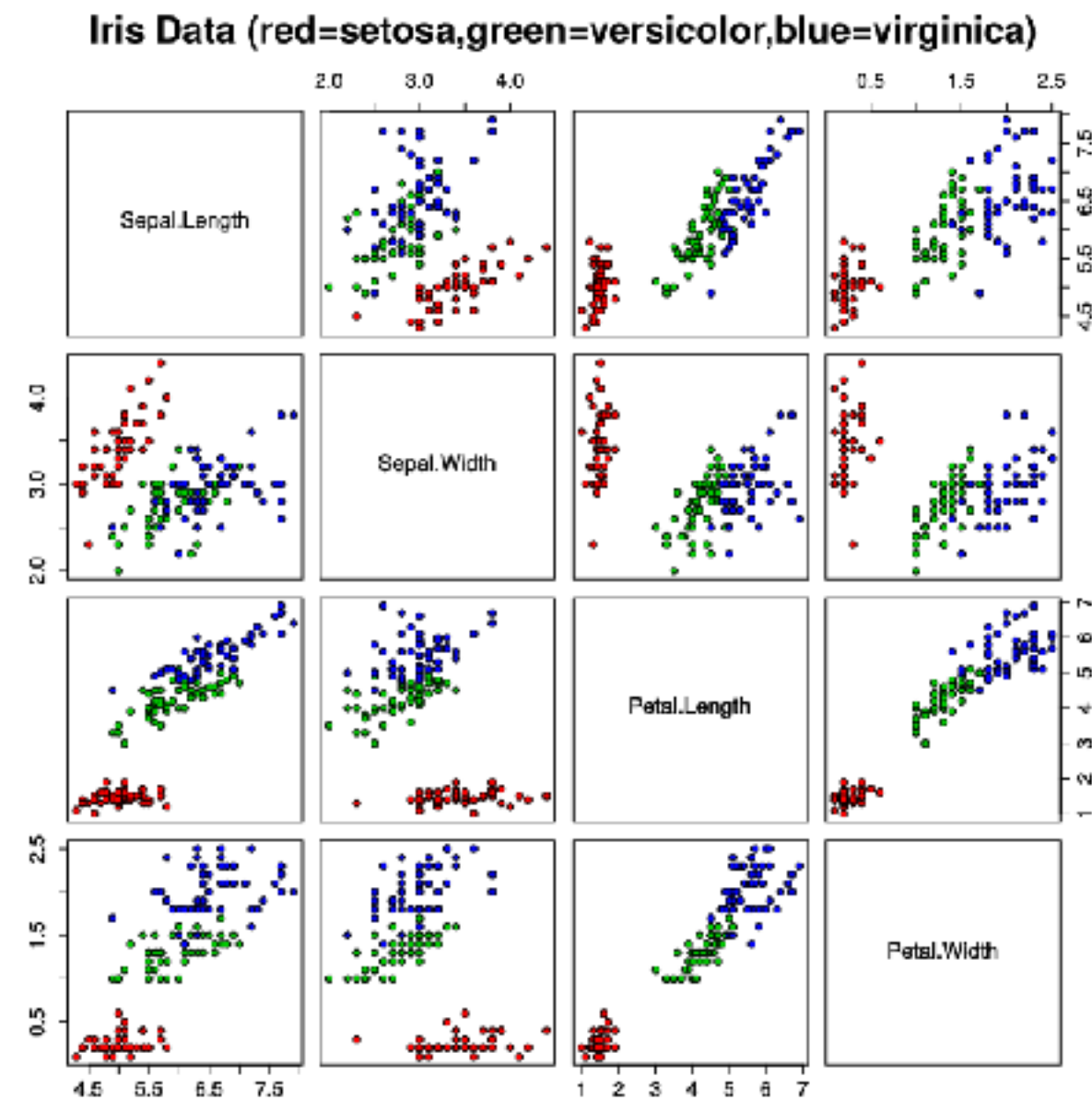


Fig. 1 IRIS Datasets



Fig. 2 MNIST Datasets

- 모든 데이터를 한꺼번에 학습할 수는 없고,
- 다양한 환경, 다양한 데이터에 대한 하나의 **Algorithm**?

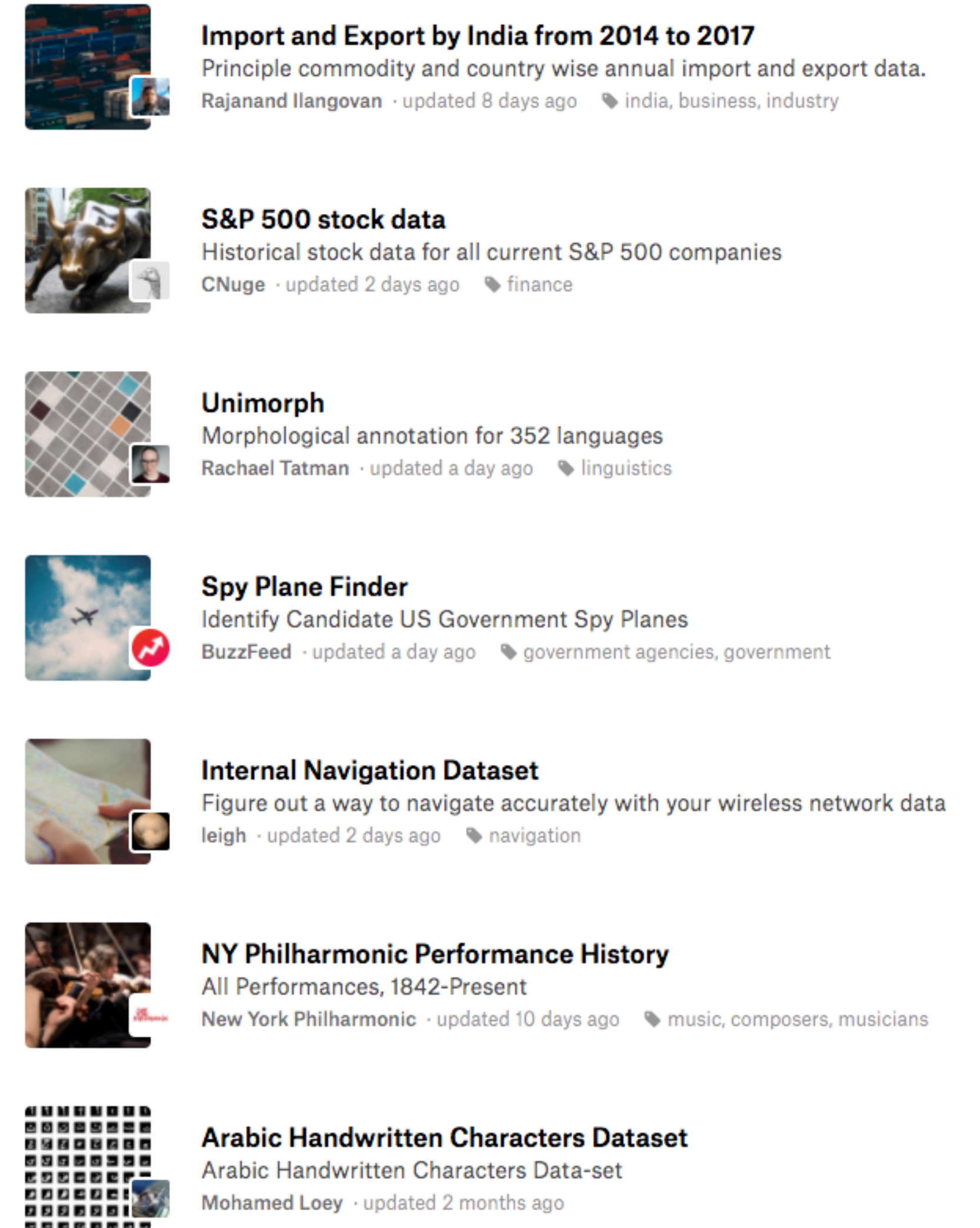
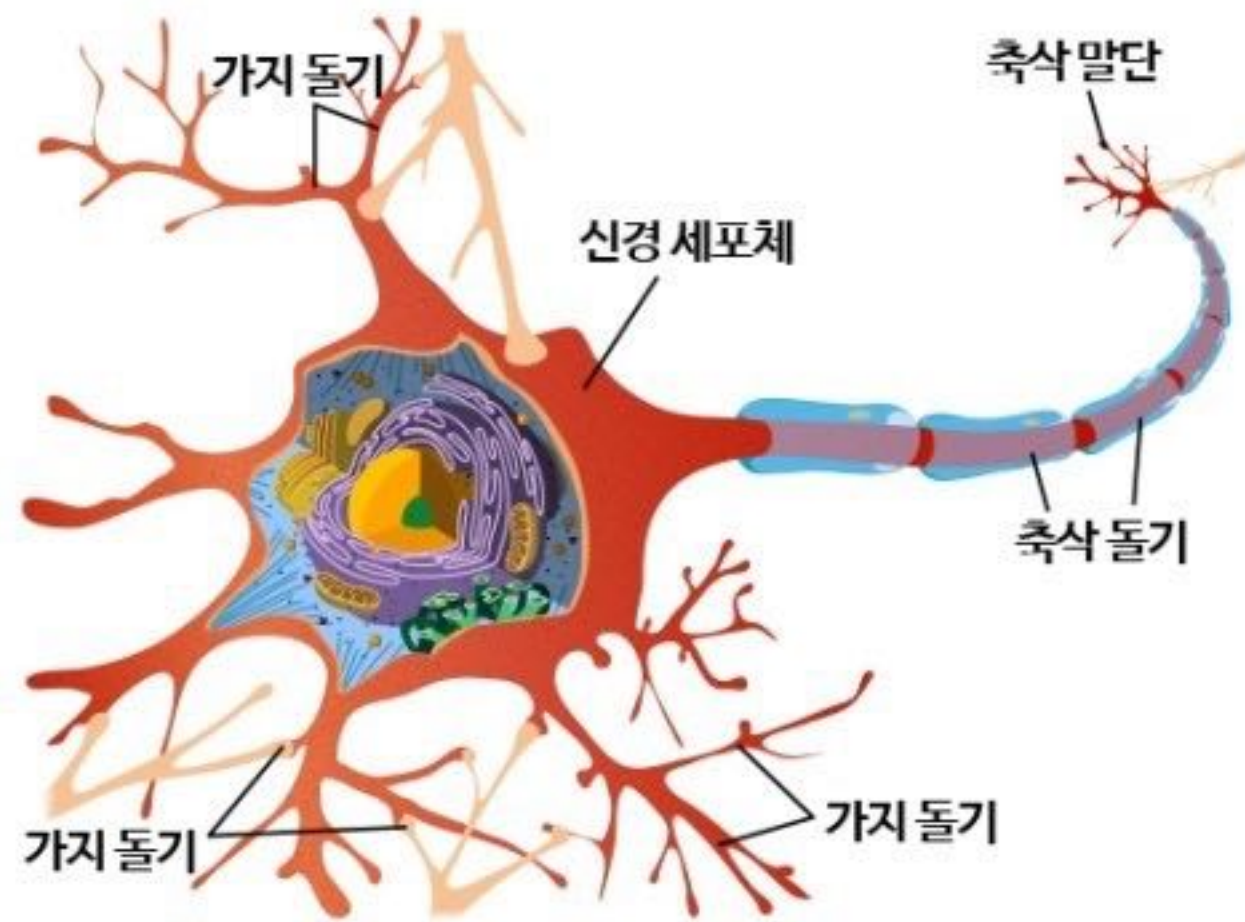


Fig. 3 Kaggle Competition List

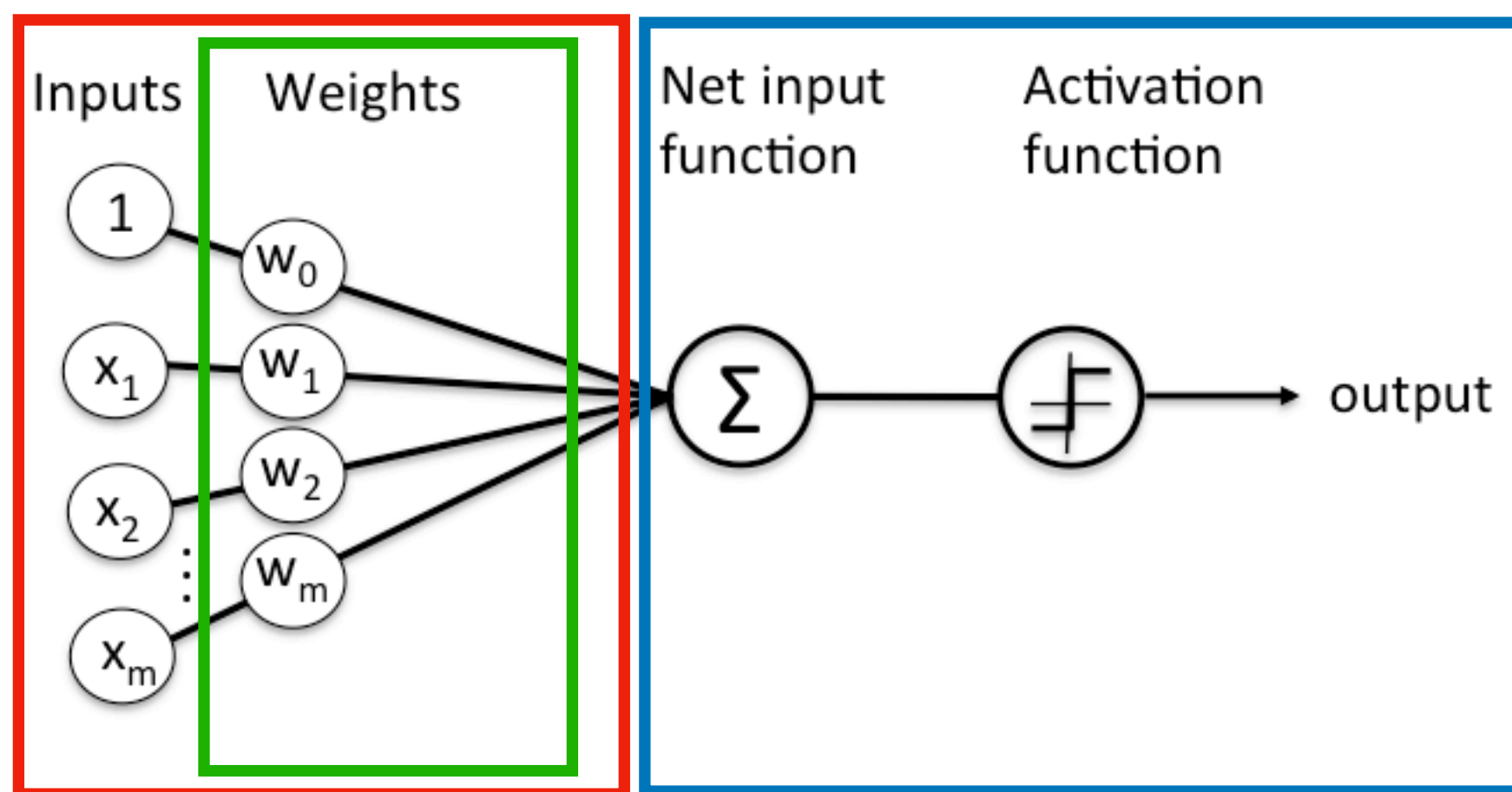


# Perceptron from Neuron



- 뉴런(Neuron)은 수상돌기(Dendrites)를 통해 다른 여러 뉴런들로부터 자극을 받아들인다.
- 하나의 뉴런은 수상돌기를 통해 다른 여러 뉴런들로부터 자극을 받아들인다.
  - 어떤 뉴런은 매우 큰 가중치로 자극을 가하고,
  - 어떤 뉴런은 매우 작은 가중치로 자극을 가한다.
- 뉴런은 수상돌기를 통해 자극을 받아들이고, 한계점(threshold)을 넘는 자극이 들어오면 축삭돌기를 통해 자극을 전달한다.

# Perceptron



Schematic of Rosenblatt's Perceptron

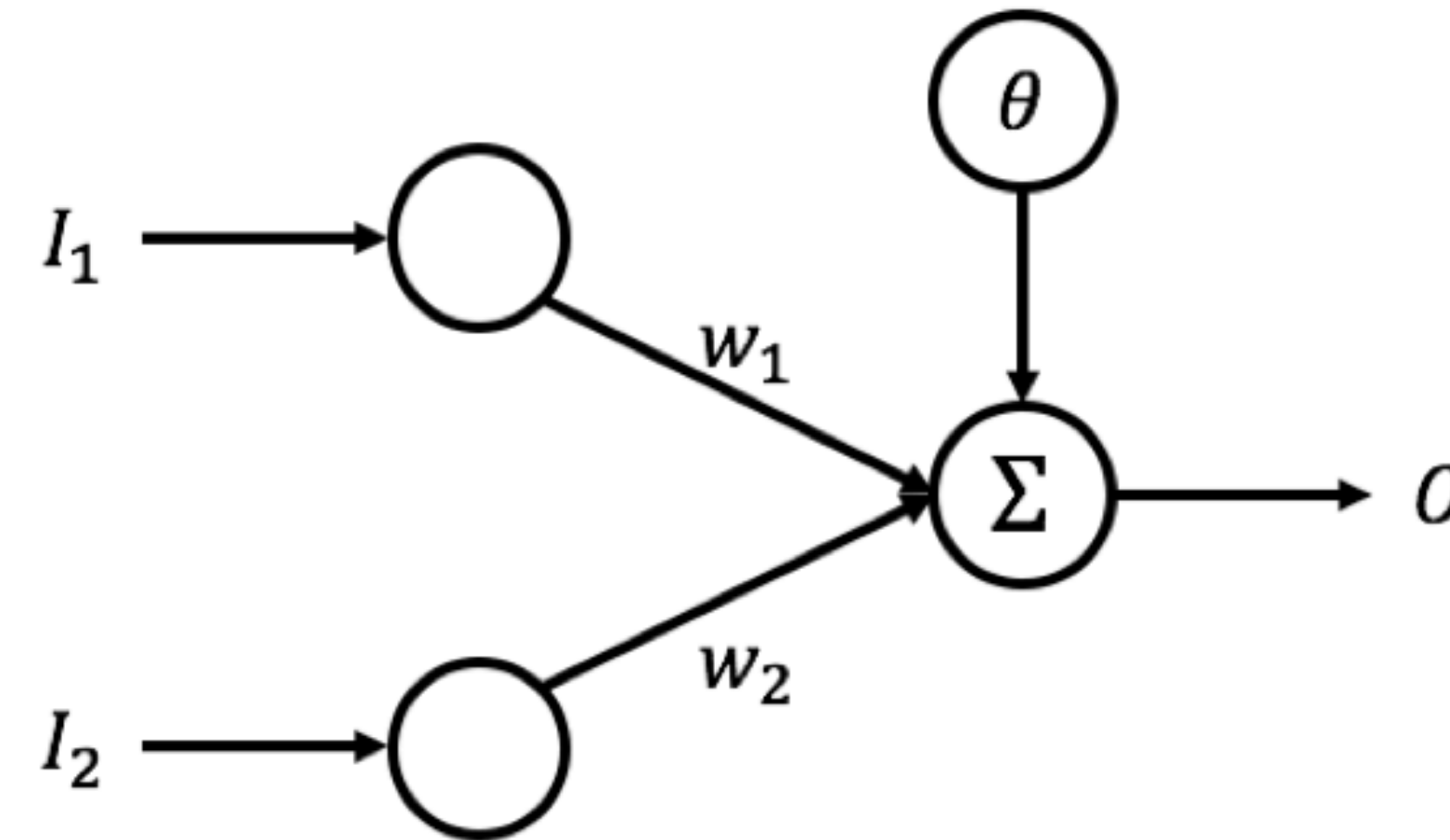
$$O = f(w_0 + \sum_{i=1}^m w_i x_i)$$

- 뉴런(Neuron)은 수상돌기(Dendrites)를 통해 다른 여러 뉴런들로부터 자극을 받아들인다.
- 하나의 뉴런은 수상돌기를 통해 다른 여러 뉴런들로부터 자극을 받아들인다.
  - 어떤 뉴런은 매우 큰 가중치로 자극을 가하고,
  - 어떤 뉴런은 매우 작은 가중치로 자극을 가한다.
- 뉴런은 수상돌기를 통해 자극을 받아들이고, 한계점(threshold)을 넘는 자극이 들어오면 축삭돌기를 통해 자극을 전달한다.

# AND/OR Gate using Perceptron

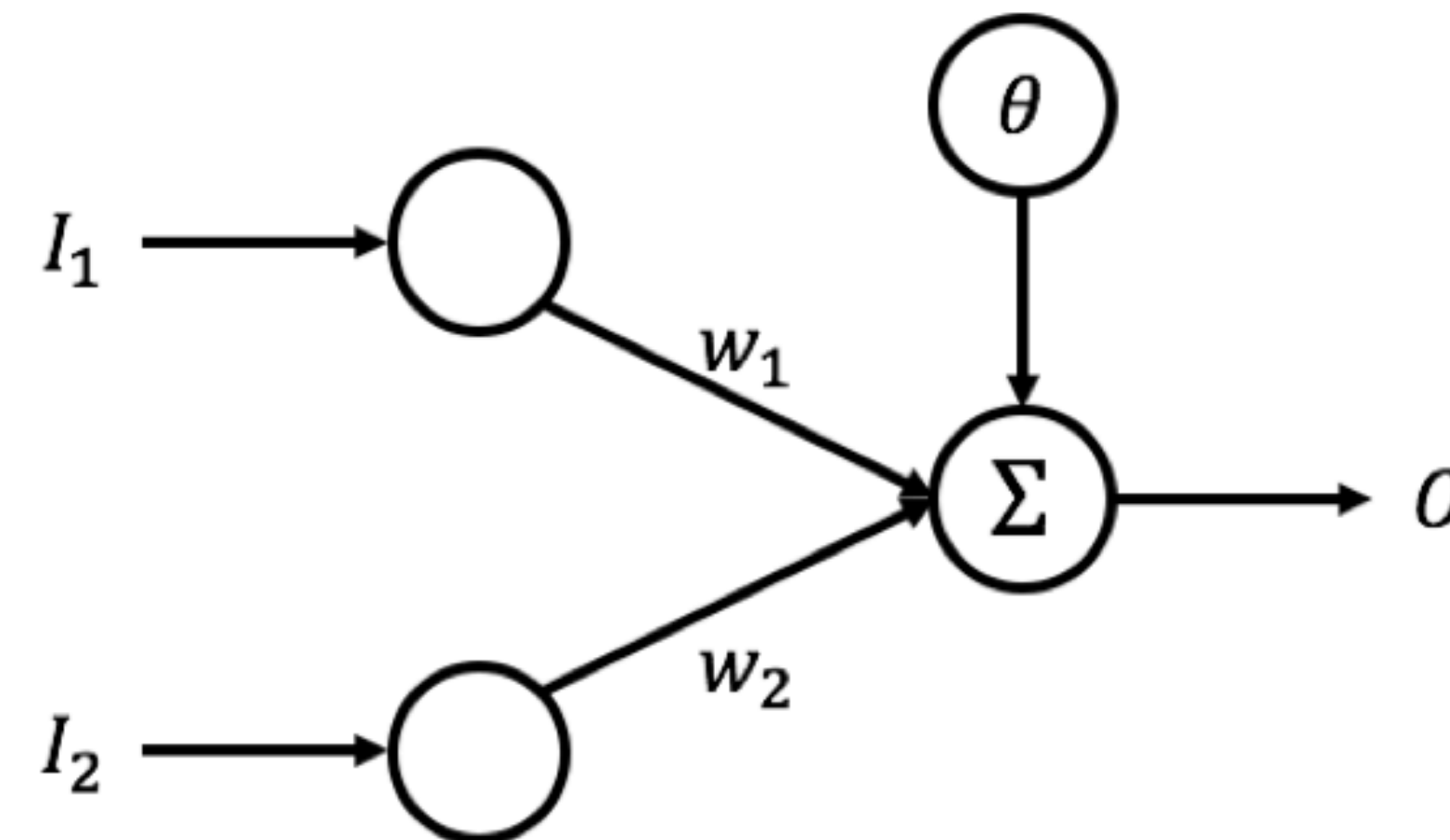
AND gate

Input A	Input B	Output
0	0	0
1	0	0
0	1	0
1	1	1



OR gate

Input A	Input B	Output
0	0	0
1	0	1
0	1	1
1	1	1





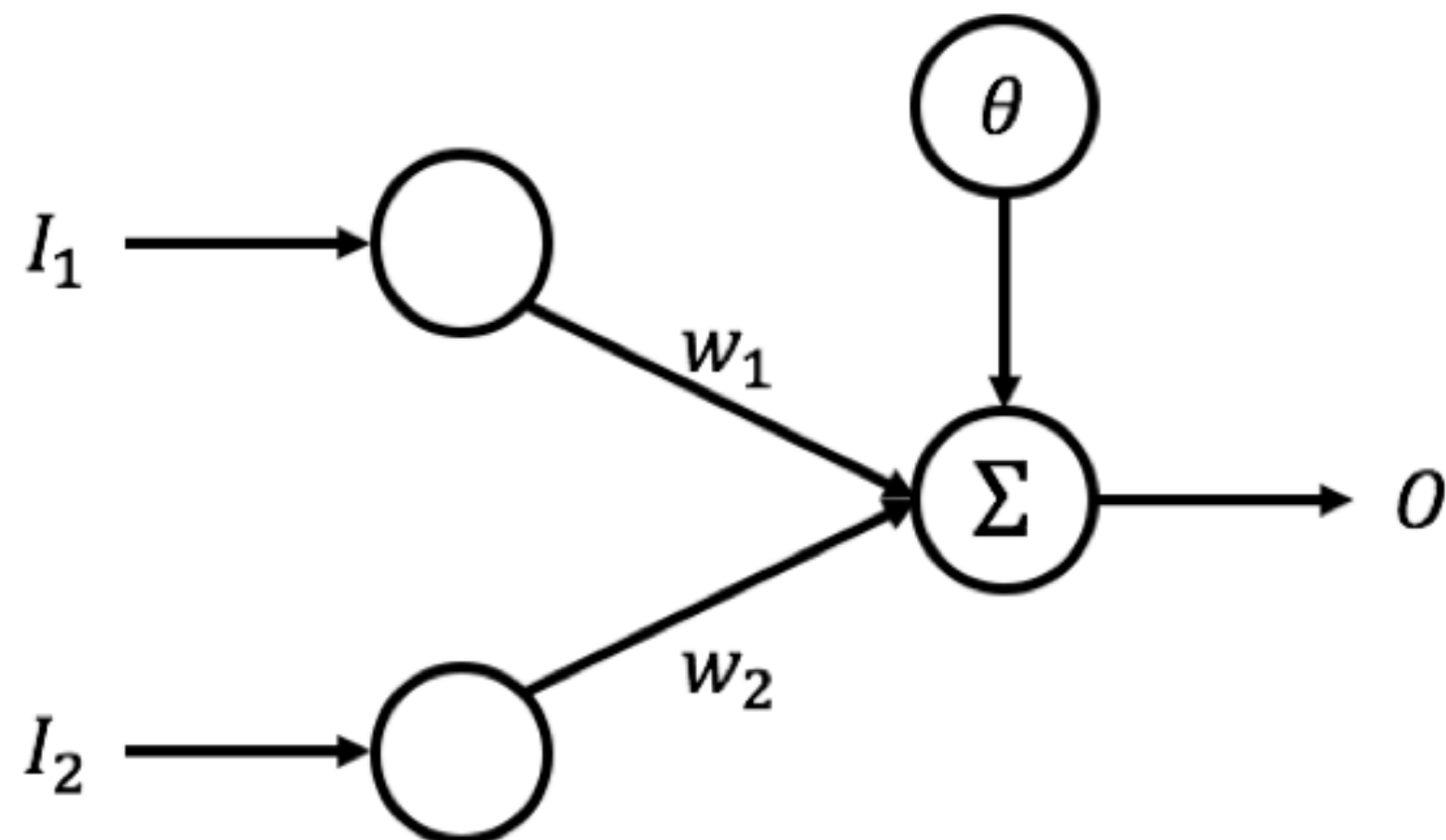
# 컴퓨터는 어떻게 풀어야 할까?

OR gate

Input A	Input B	Output
0	0	0
1	0	1
0	1	1
1	1	1

- 컴퓨터는 인간처럼 사고 할 수 없다.
  - 모든 것을 하나하나 지시해주어야 한다.
- 에러(Error)를 수정해 나갈 수 있는 방법은?

$$\Delta w_i = \alpha(T - O)x_i$$

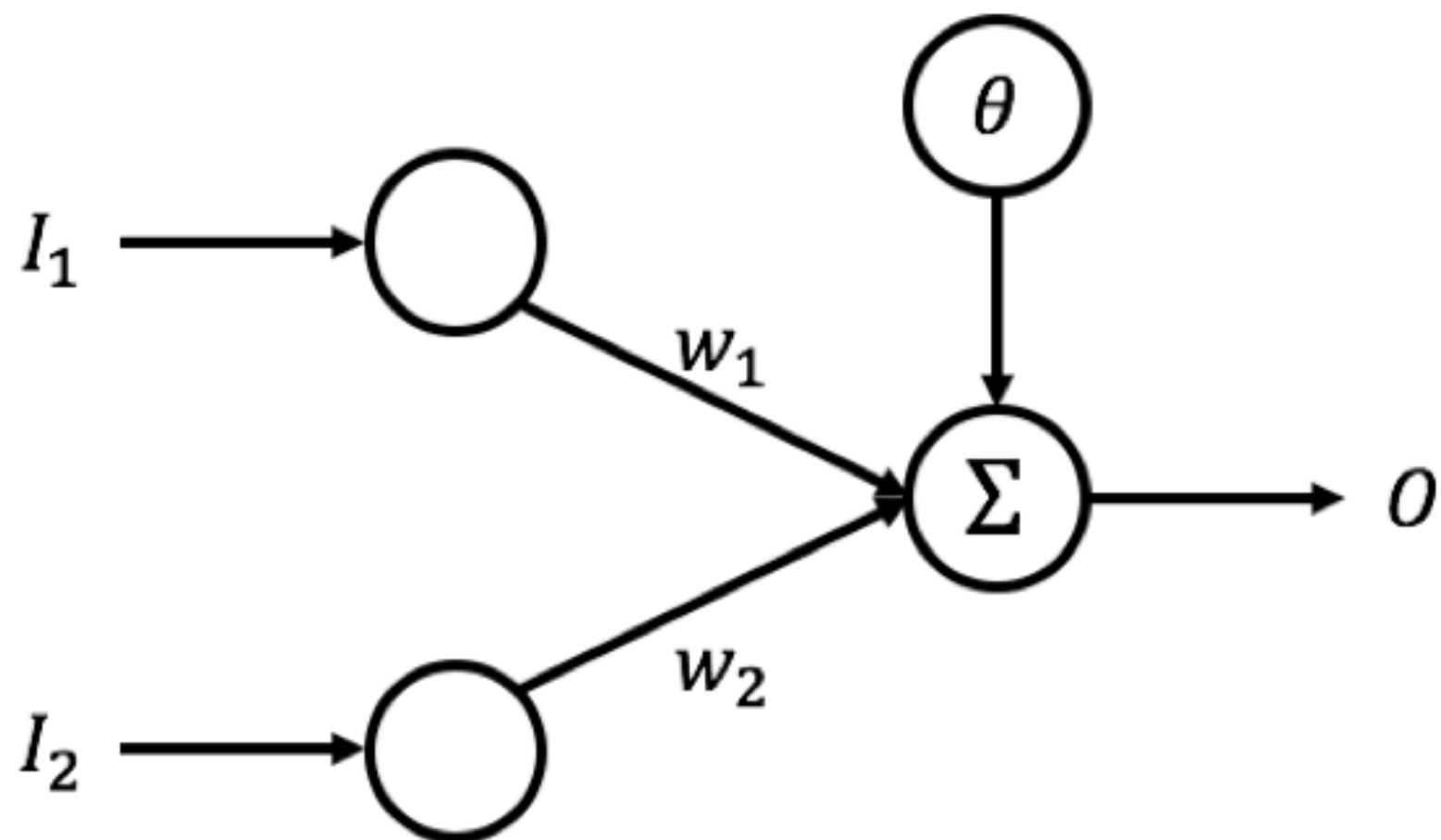


- T : Target Value
- O : Output
- $\alpha$  : Learning Rate

$$\Delta w_i = \alpha(T - O)x_i$$

### AND gate

Input A	Input B	Output
0	0	0
1	0	0
0	1	0
1	1	1

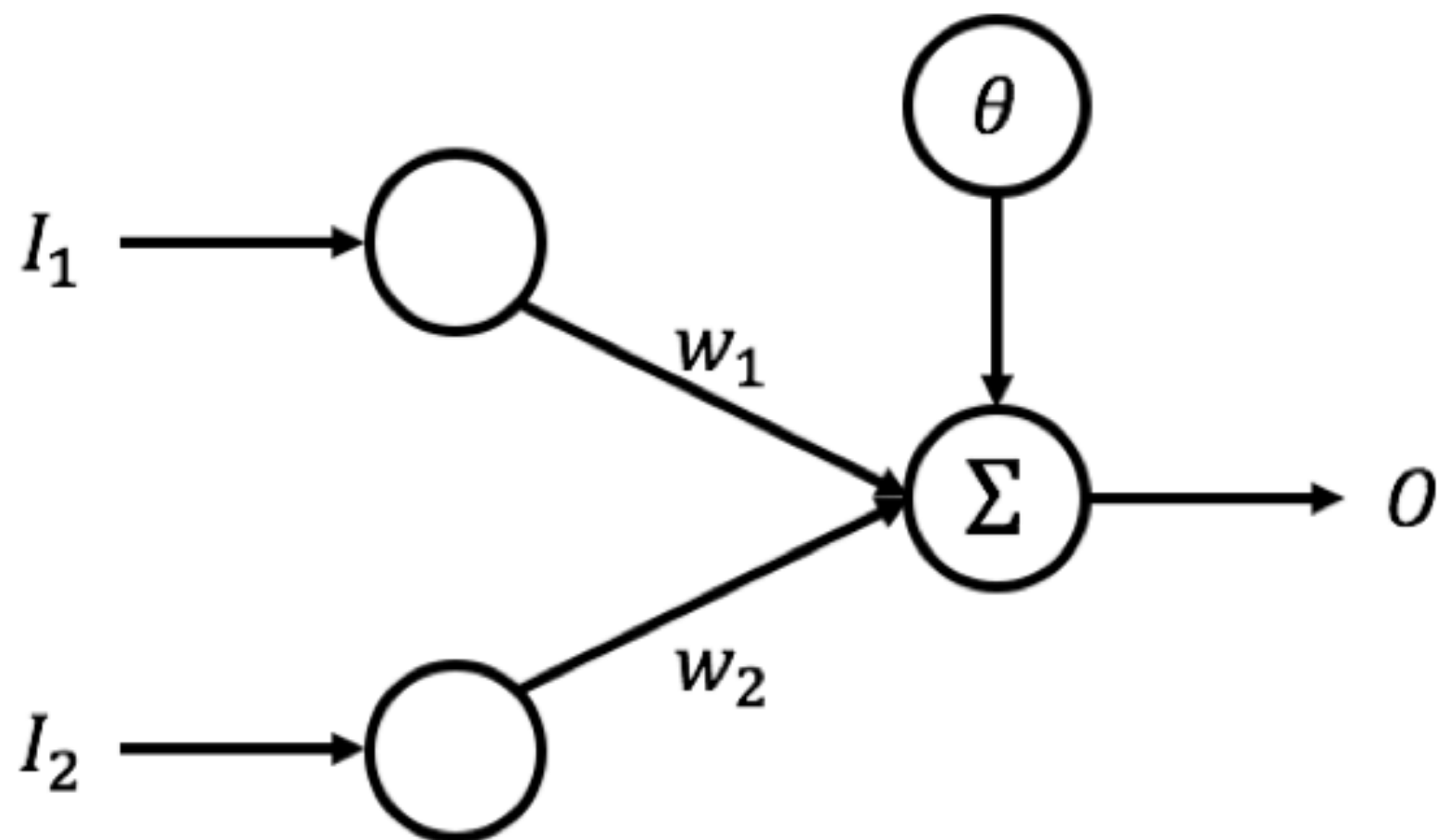
[illegible]



$$\Delta w_i = \alpha(T - O)x_i$$

OR gate

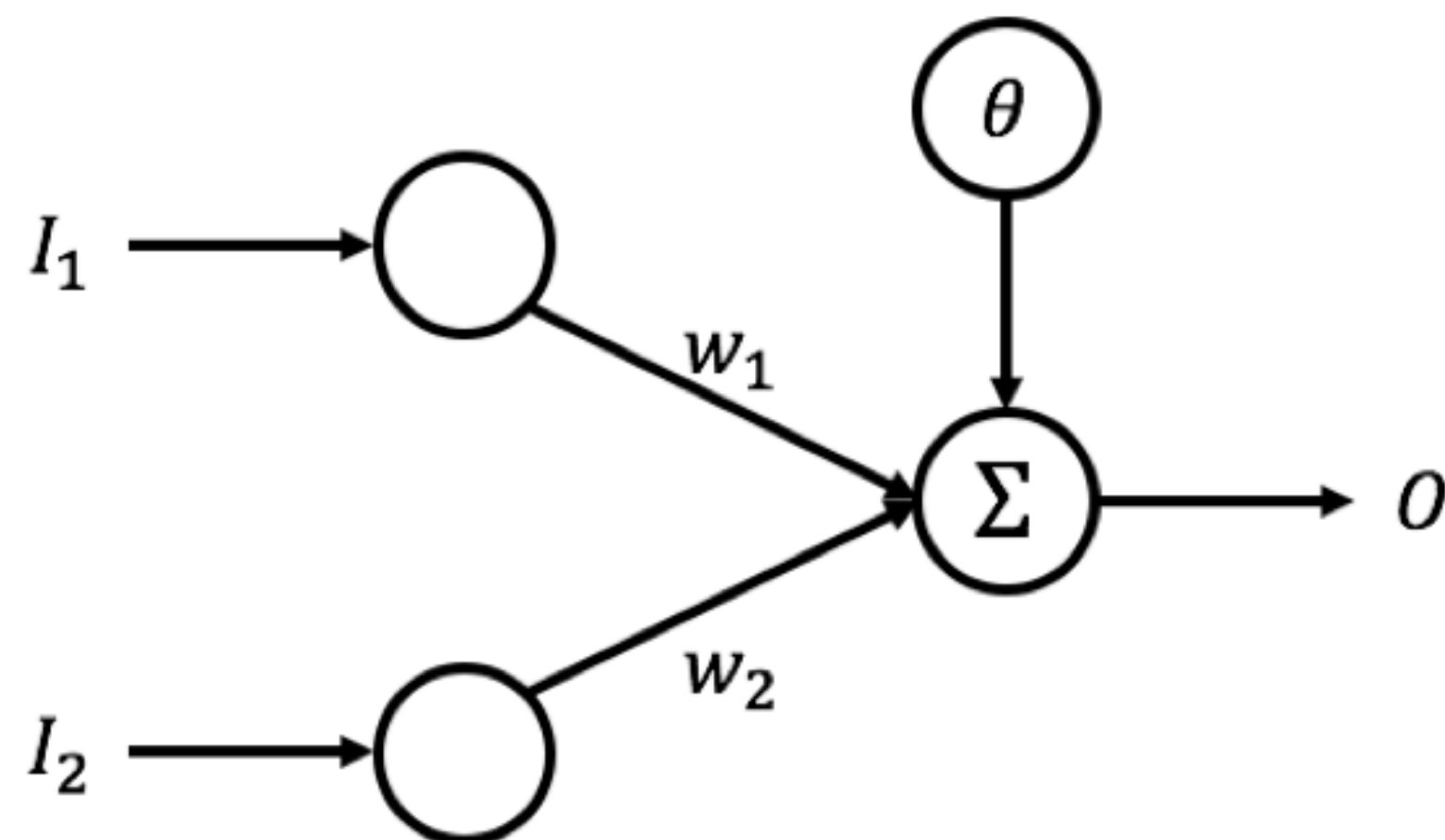
Input A	Input B	Output
0	0	0
1	0	1
0	1	1
1	1	1

[illegible]

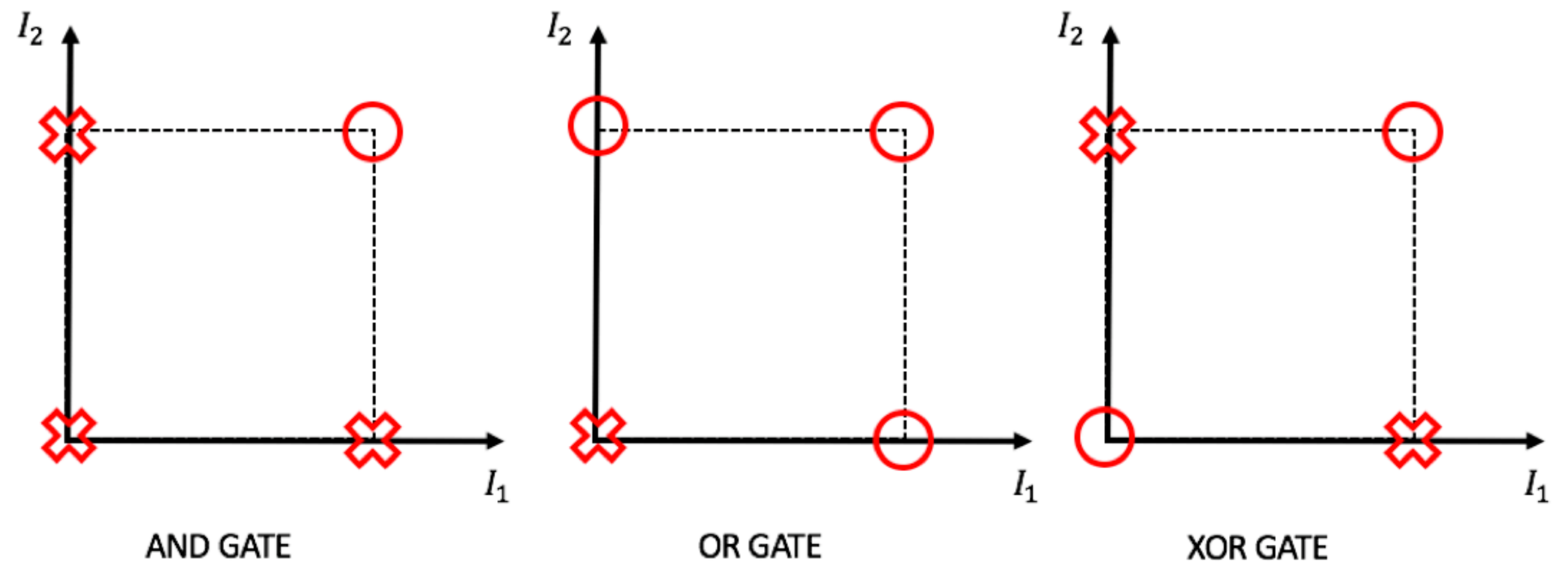
# AND/OR and so on

EX-OR gate

Input A	Input B	Output
0	0	0
1	0	1
0	1	1
1	1	0



$$O = \begin{cases} 1 & I_1 w_1 + I_2 w_2 - \theta > 0 \\ 0 & I_1 w_1 + I_2 w_2 - \theta < 0 \end{cases}$$



선을 하나만 그어 이 문제를 풀 수 있는가?

# “Perceptron”, Marvin Minsky

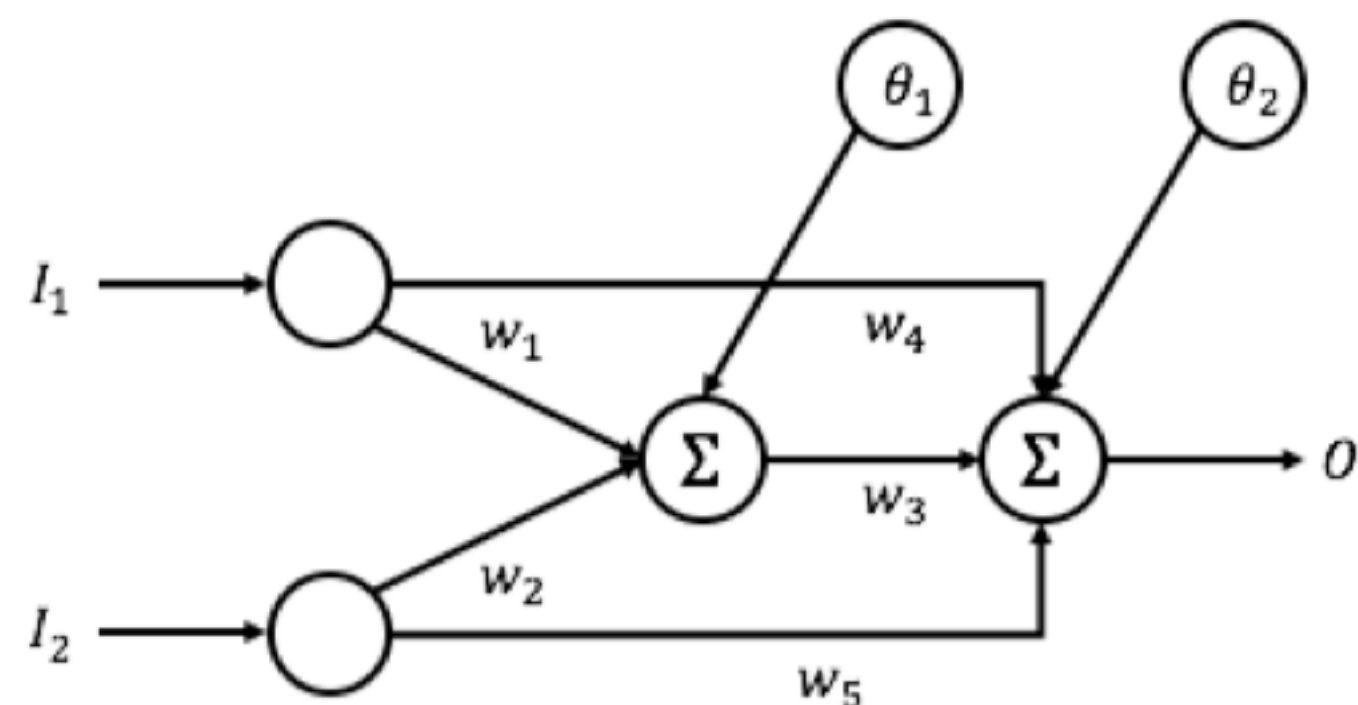
This book is the center of a long-standing controversy in the study of artificial intelligence. It is claimed that pessimistic predictions made by the authors were responsible for an erroneous change in the direction of research in AI, concentrating efforts on so-called "symbolic" systems, and contributing to the so-called AI winter.

어떻게 문제를 해결 할 수 있을까?

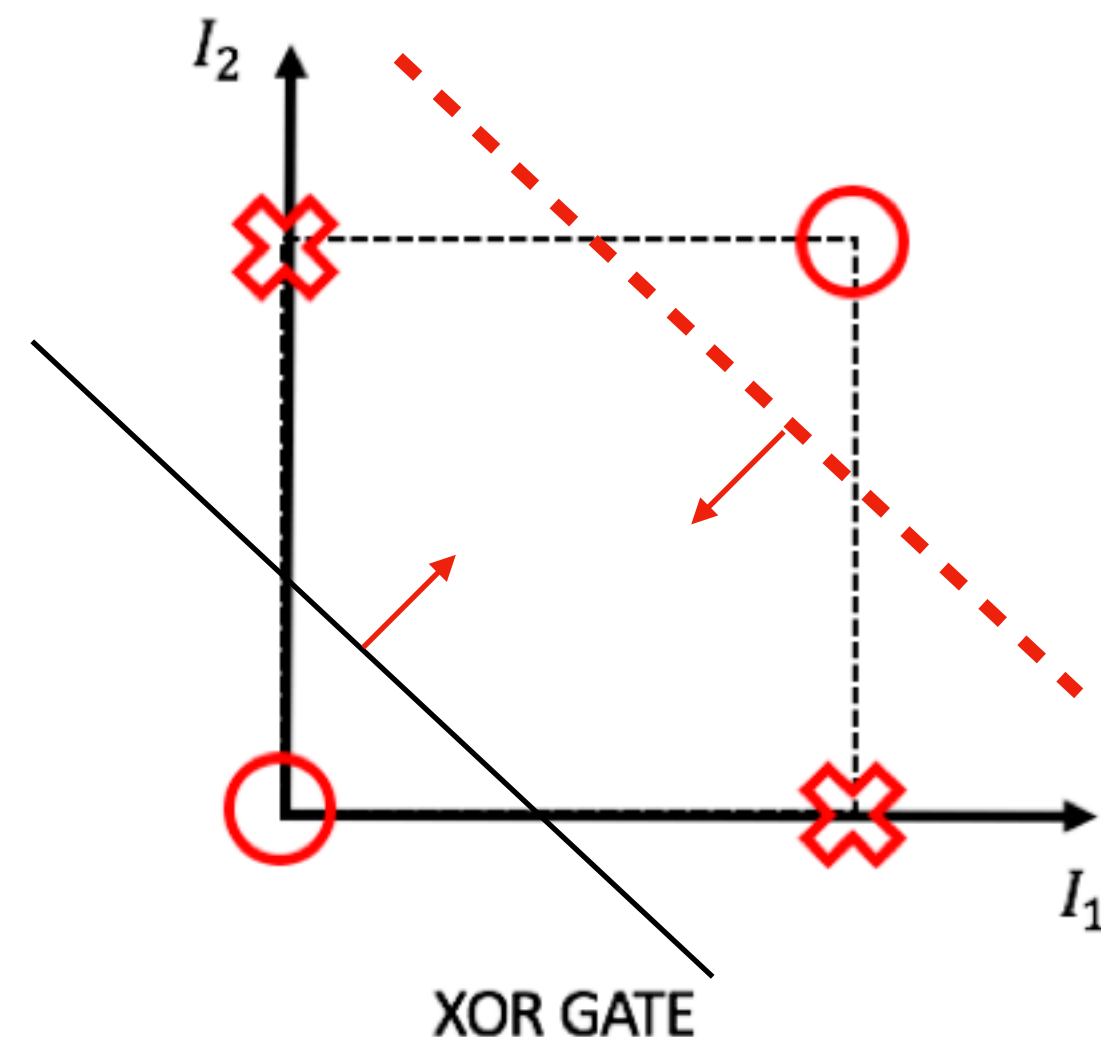
# Hidden Layer

EX-OR gate

Input A	Input B	Output
0	0	0
1	0	1
0	1	1
1	1	0



- Input과 Output사이의 하나의 은닉층 (Hidden Layer)를 넣는 것만으로 문제를 해결 할 수 있다.
- 선을 하나 더 긋고 (Perceptron 추가), 두 선 사이에 있는지 체크(Layer 추가)하는 역할을 수행한다.



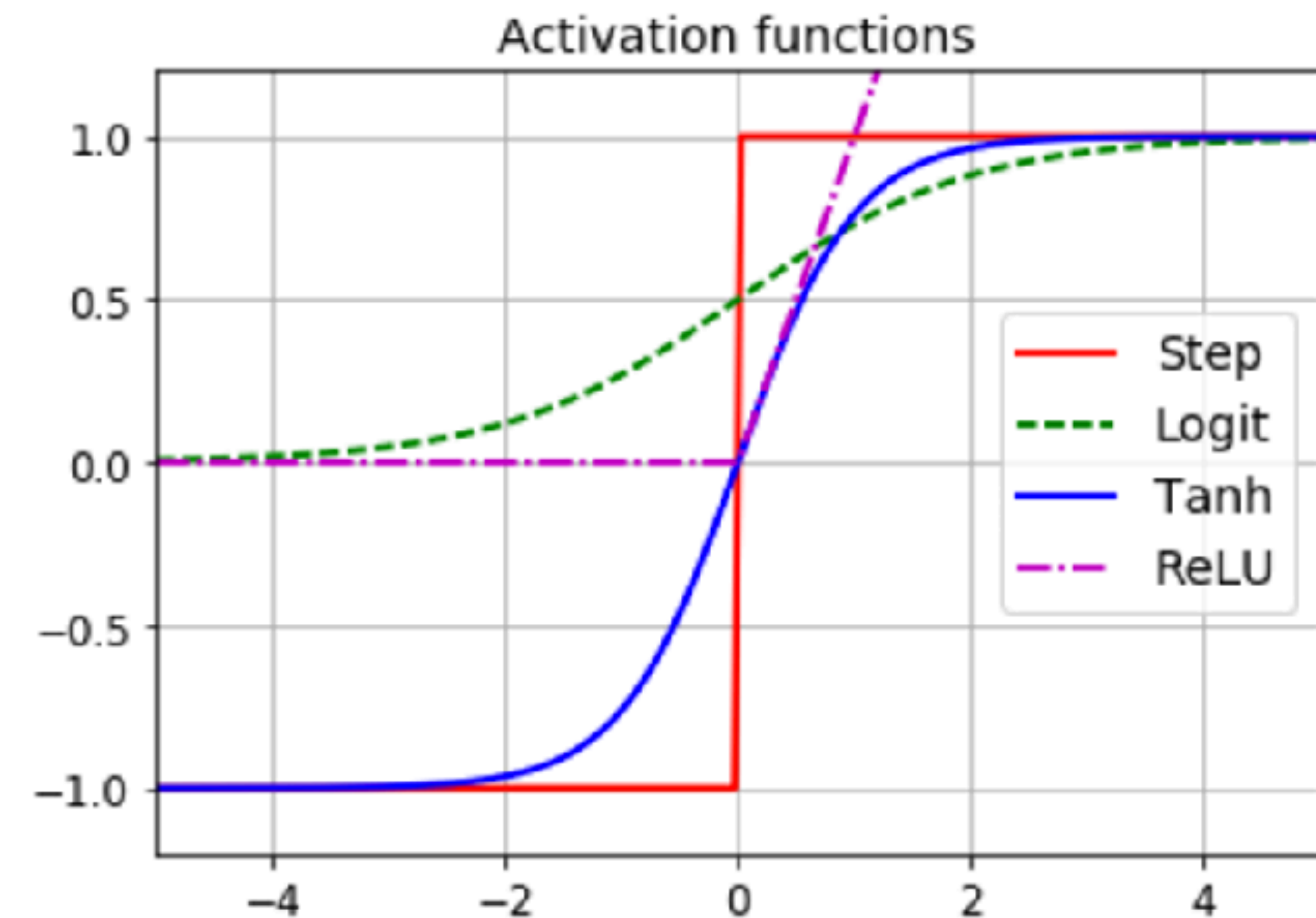
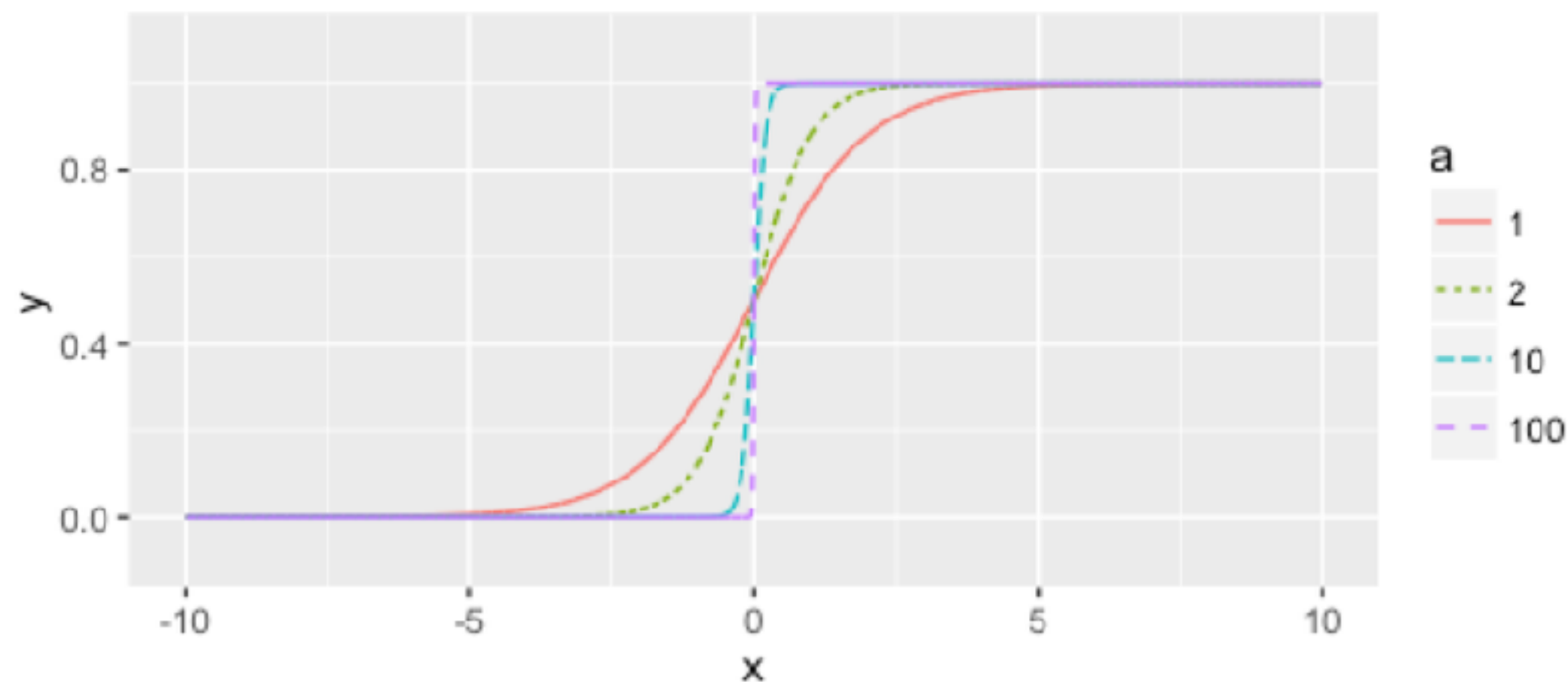
- 여러 Layer로 Perceptron을 쌓으면 non-linear 문제도 풀 수 있다!



# Activation Function : Non-linear

Sigmoid Function

$$\sigma(x) = \frac{1}{1 + \exp(-ax)}$$



- Activation Function이 Step function인 경우 최종적인 결과는 linear들의 집합이기 때문에, Non-linear function을 붙인다.

# BackPropagation(BP)

- Hidden Layer와 Non-Linear Function을 Activation Function으로 갖는 경우 어떻게 학습을 시켜야 할까?

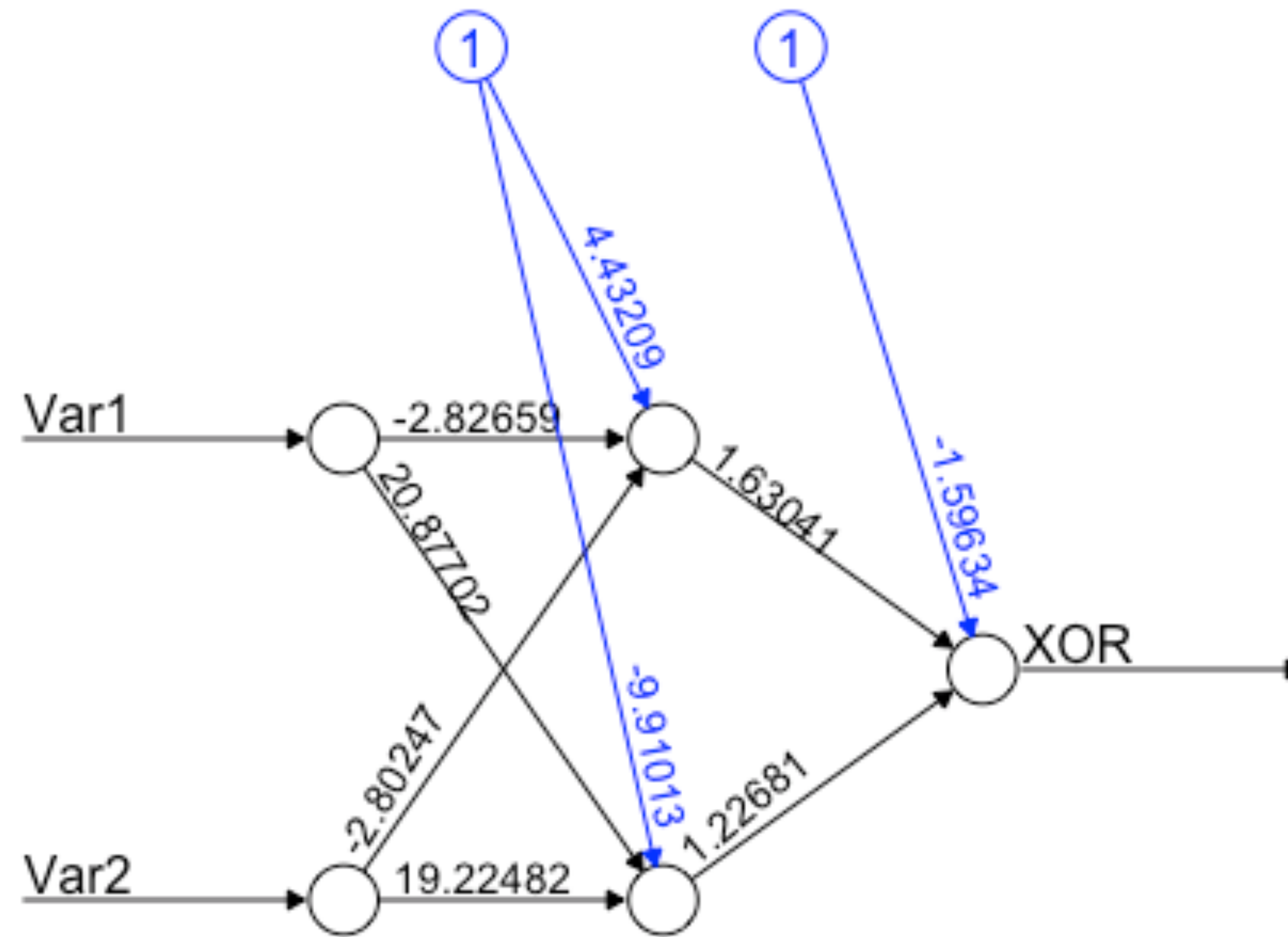
$$\Delta w_i = \alpha(T - O)x_i$$

Activation Function에 대한 변화량을 추가해주면 된다

$$\Delta w_i = \alpha(T - O)g'(h_j)x_i$$

- $g(x)$  is the neuron's activation function
- $h(x)$  is the weighted sum of the neuron's inputs

# R Code Example



Error: 0.00024 Steps: 200

# Python Code Example

