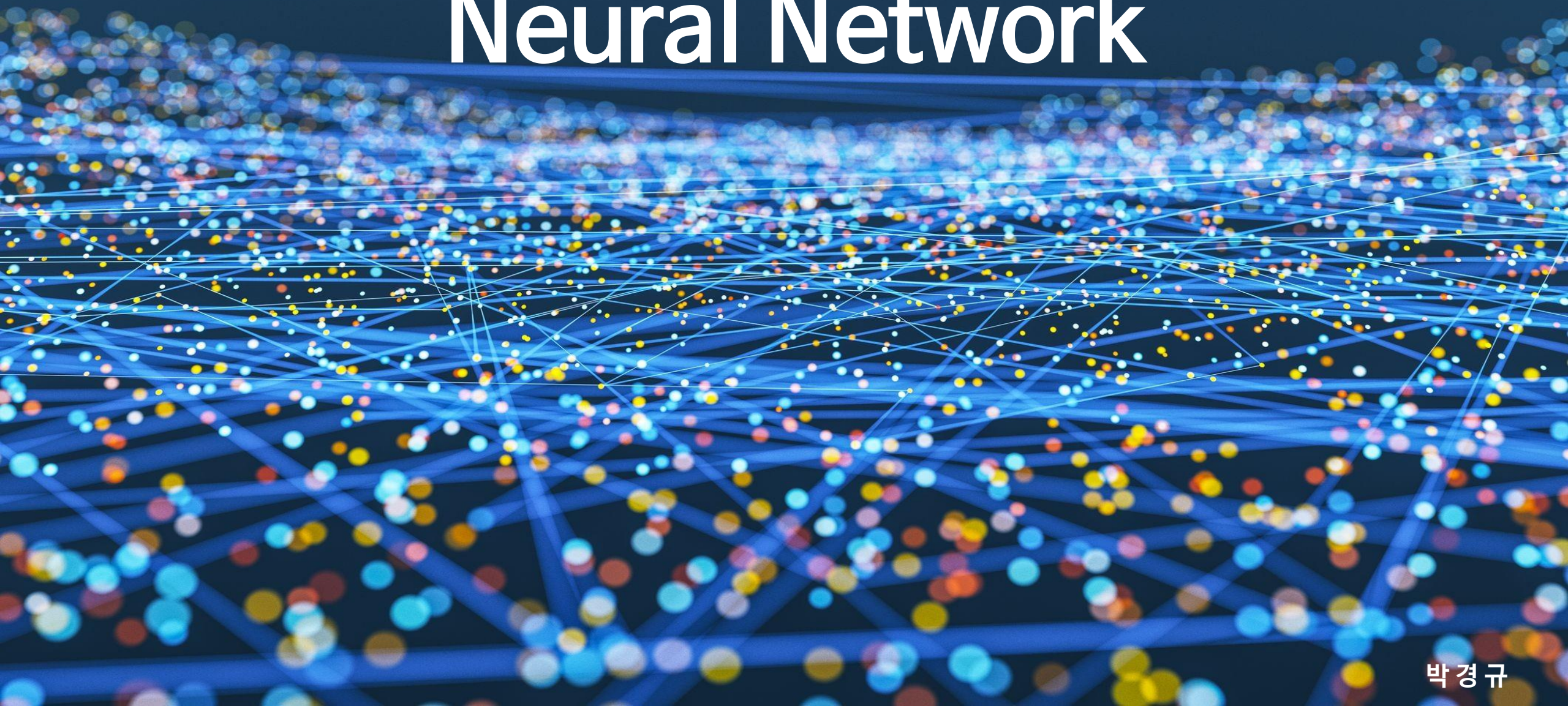
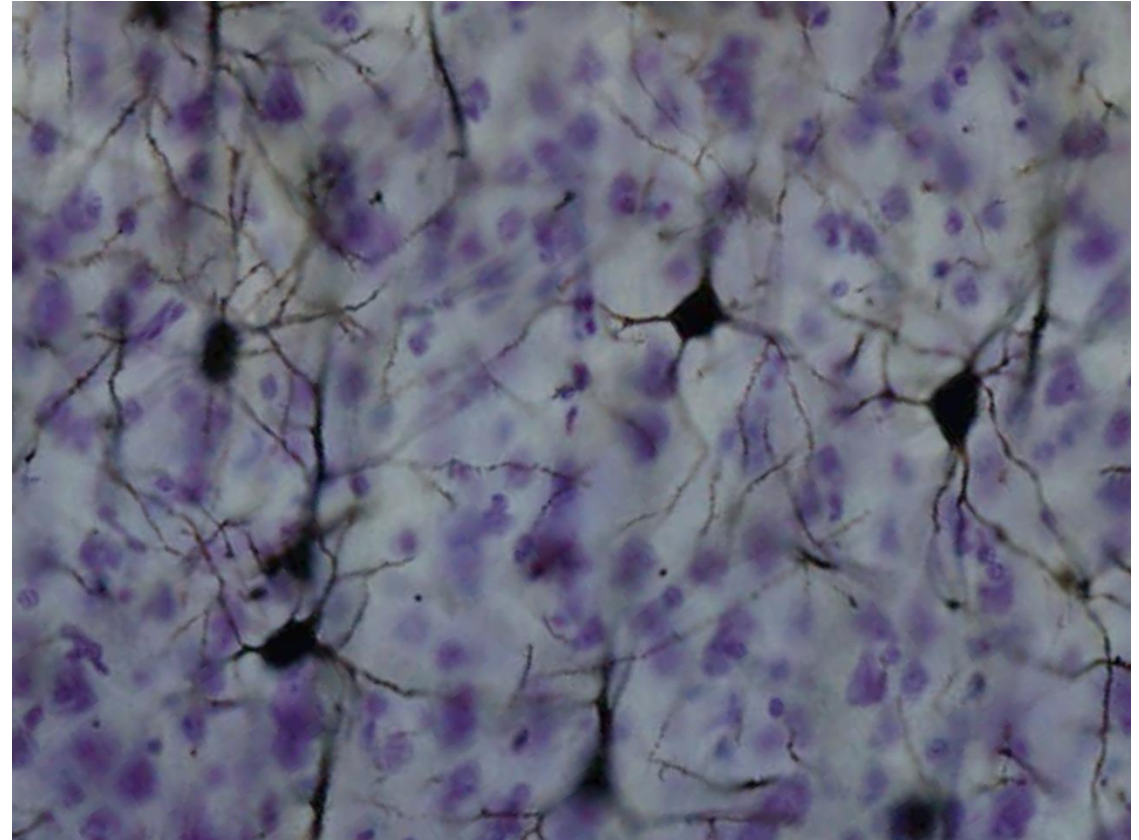
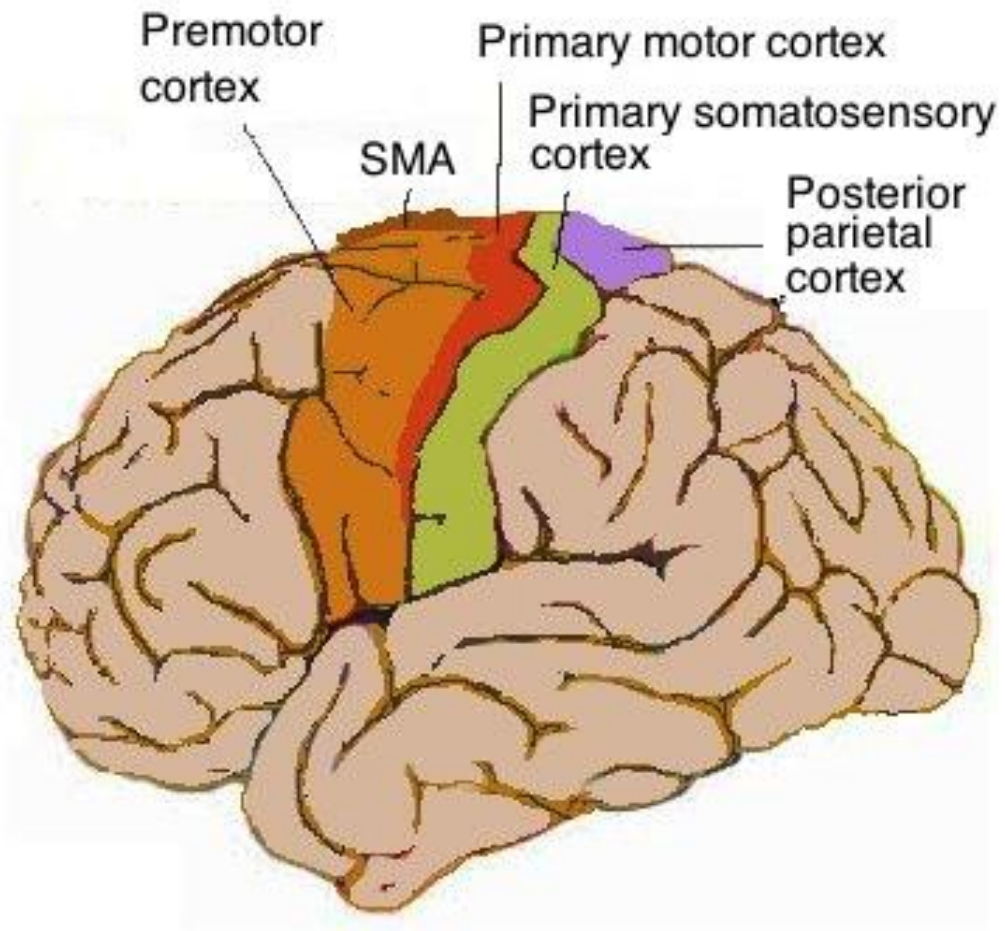


# Neural Network



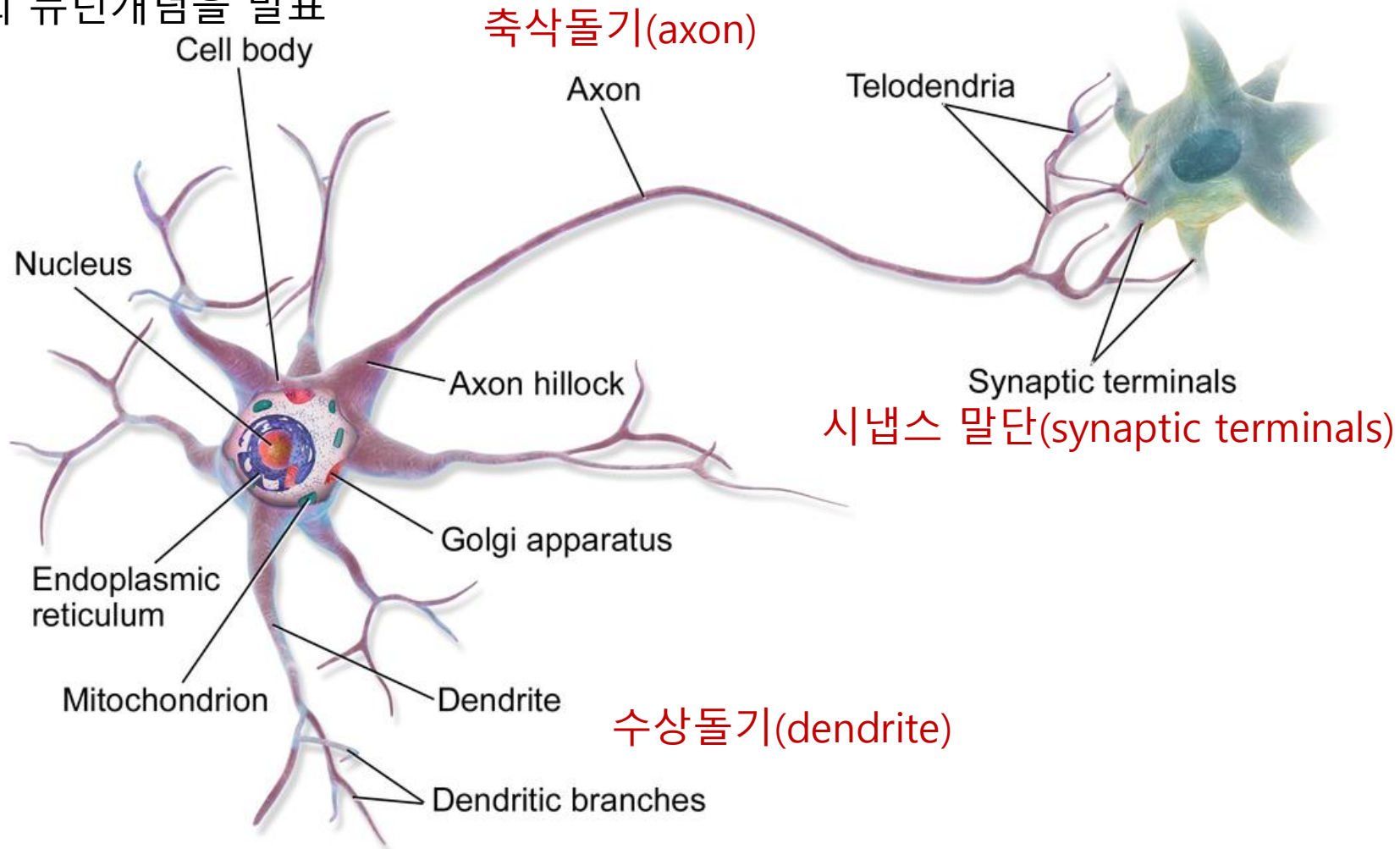


# 신경망 (Neural Network)



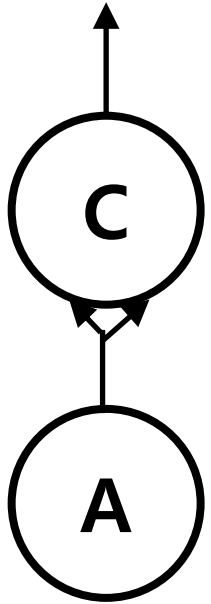
# 맥컬록-피츠(MCP) 뉴런

- 1943년 신경 생리학자 워런 맥컬록(Warren McCulloch) 과 수학자 월터 피츠 (Walter Pitts) 가 간소화된 뇌의 뉴런개념을 발표

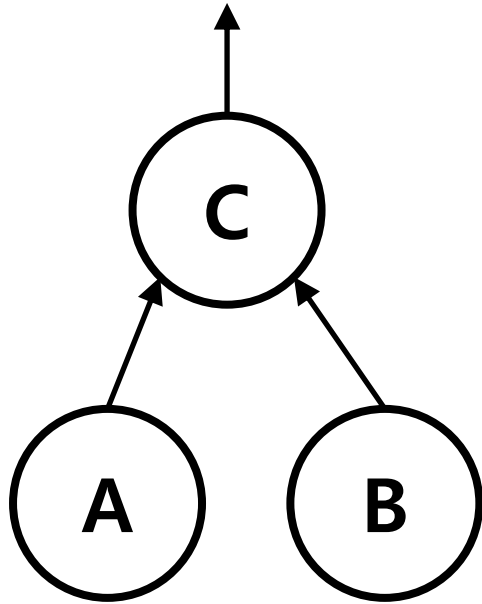


# 맥컬록-피츠(MCP) 뉴런

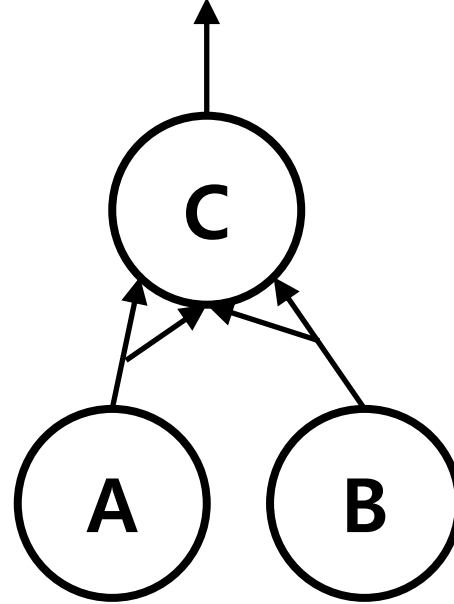
- 명제논리를 사용하여 동물 뇌의 생물학적 뉴런이 복잡한 계산을 위해 어떻게 상호작용하는지에 대한 간단한 계산 모델 제공



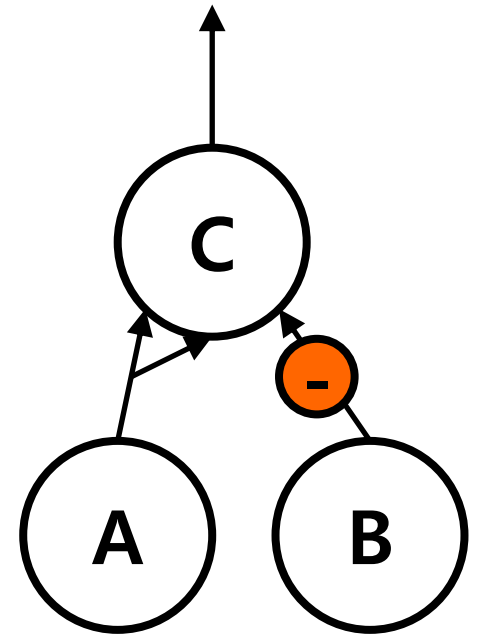
$$C = A$$



$$C = A \wedge B$$



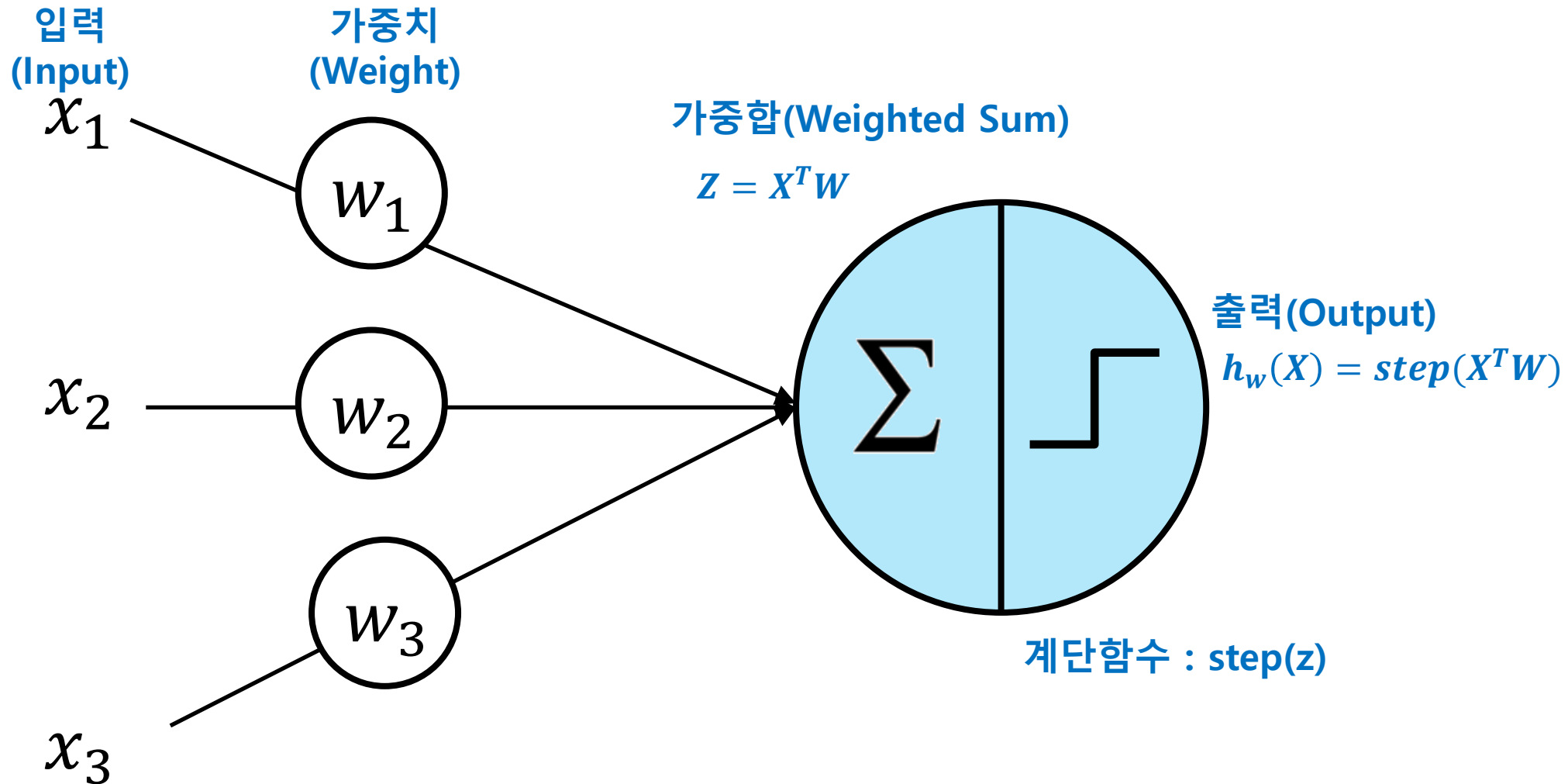
$$C = A \vee B$$



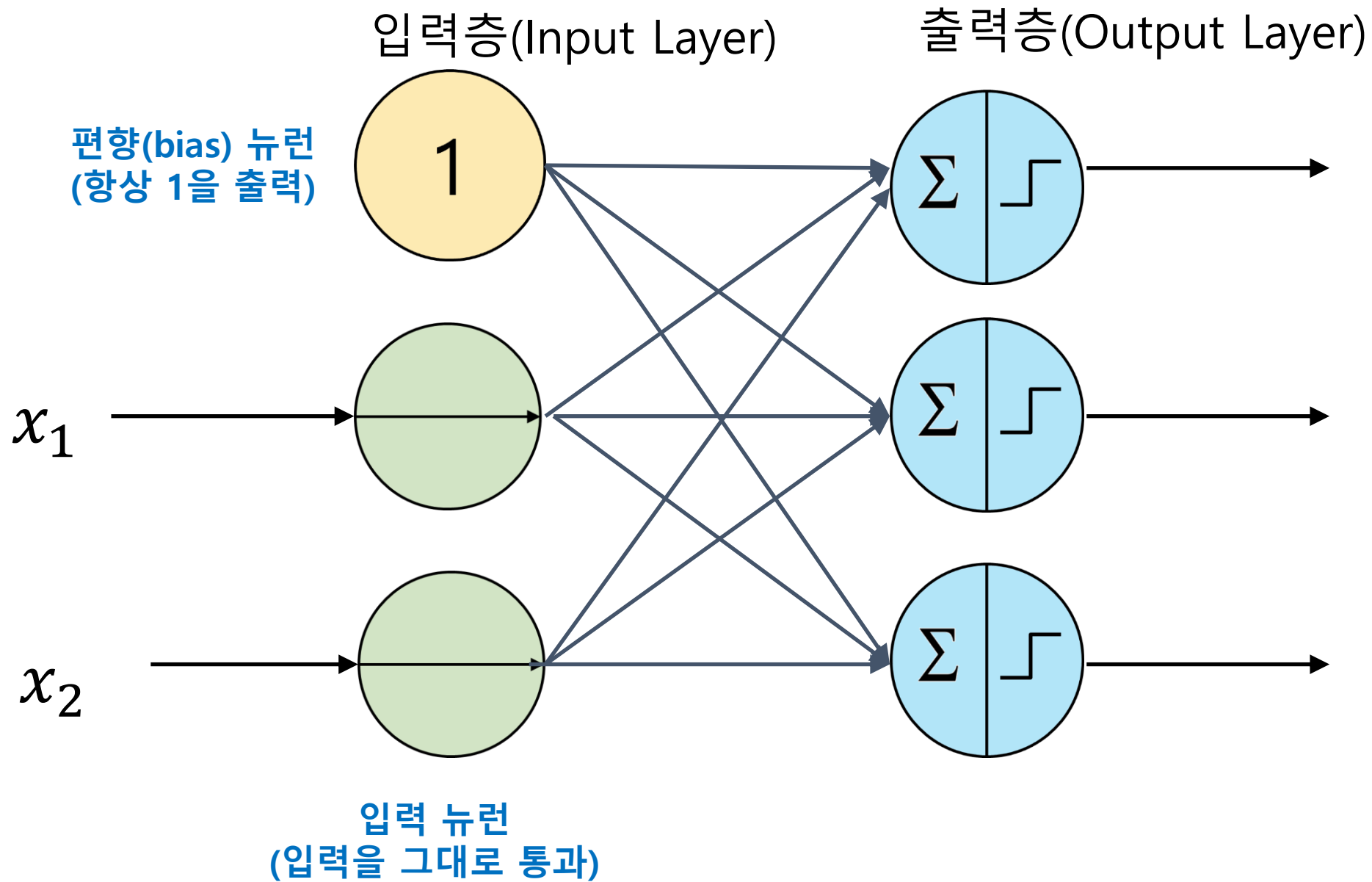
$$C = A \wedge \neg B$$

# 퍼셉트론(Perceptron)

1957년 프랑크 로젠플라트(Frank Rosenblatt)가 제안한 간단한 인공신경망 구조로 TLU(Threshold logi unit)라고 부르는 인공 뉴런을 기반으로 합니다.



# 퍼셉트론 (Perceptron)



# 퍼셉트론(Perceptron)

## ■ 완전 연결층의 출력 계산

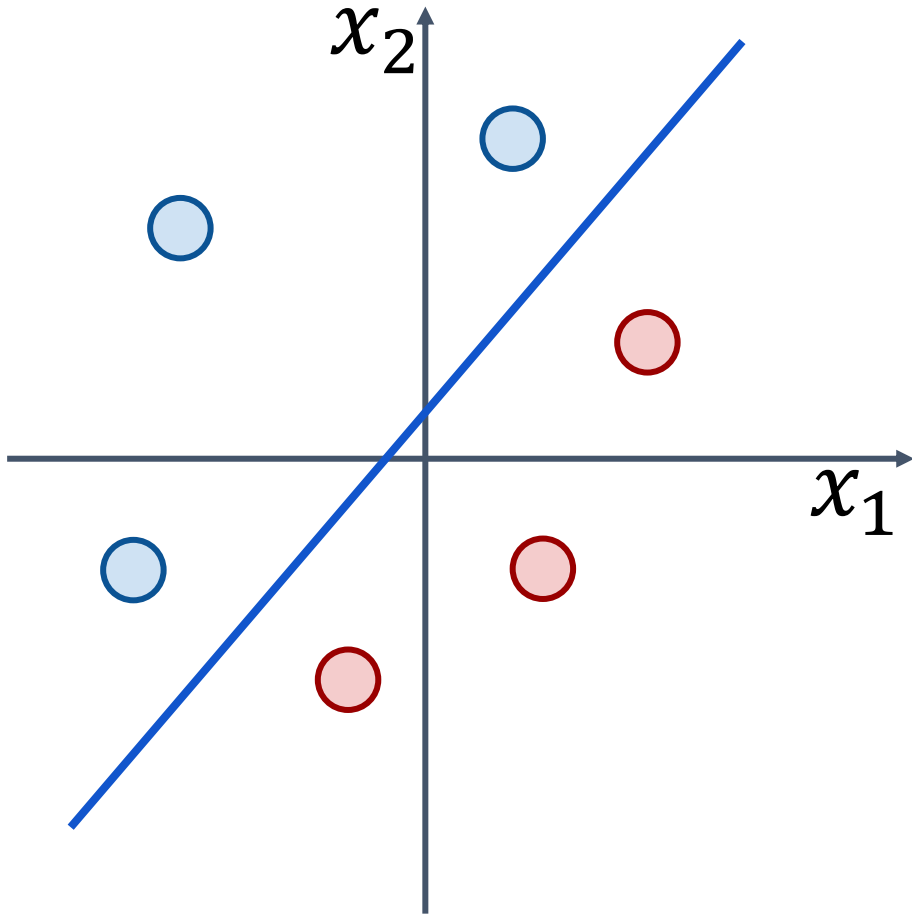
$$h_{w,b}(X) = \phi(XW + b)$$

## ■ 퍼셉트론 학습규칙(가중치 업데이트)

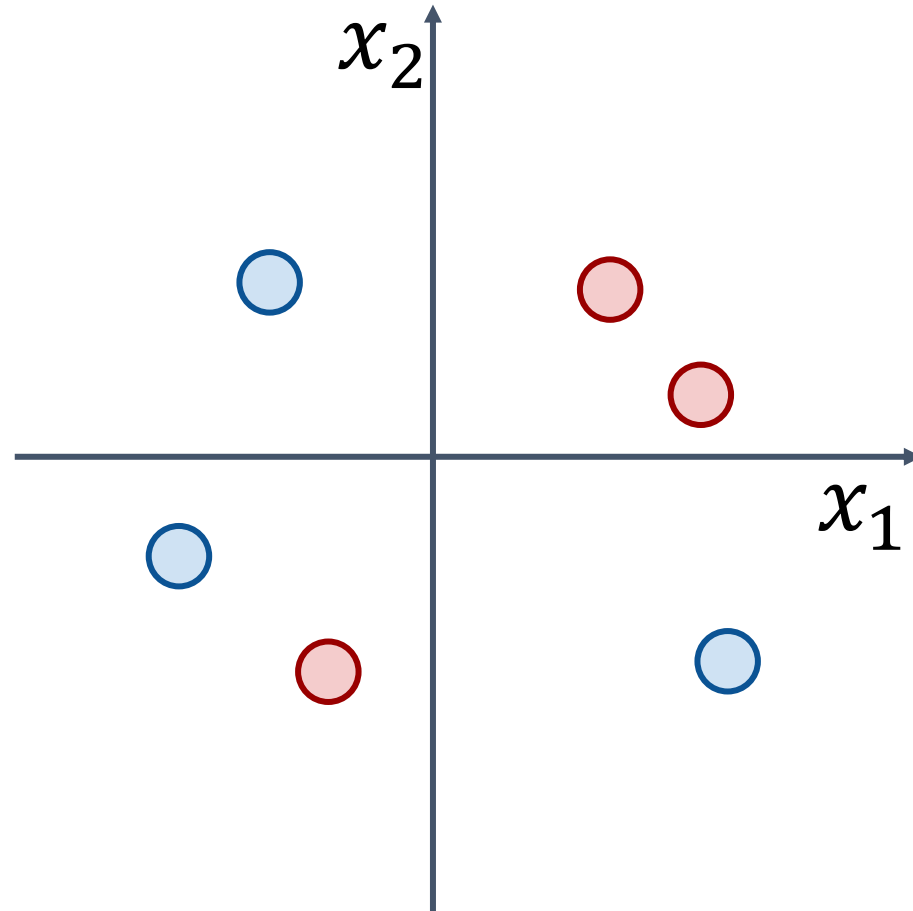
$$w_{i,j}^{(\text{next step})} = w_{i,j} + \eta(y_j - \hat{y}_j)x_i$$

# 퍼셉트론의 한계

## ■ 선형 분류



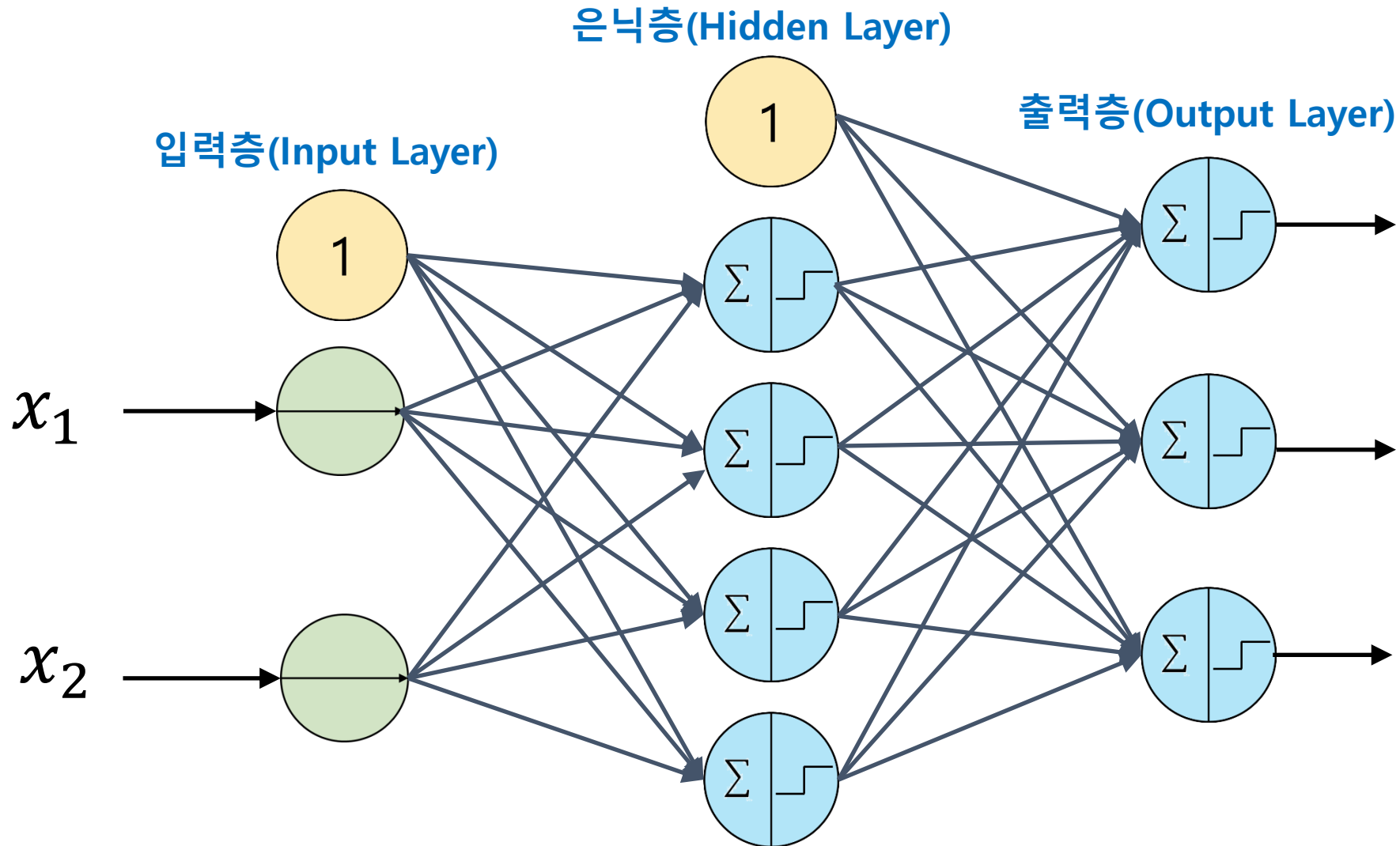
## ■ 비선형 분류





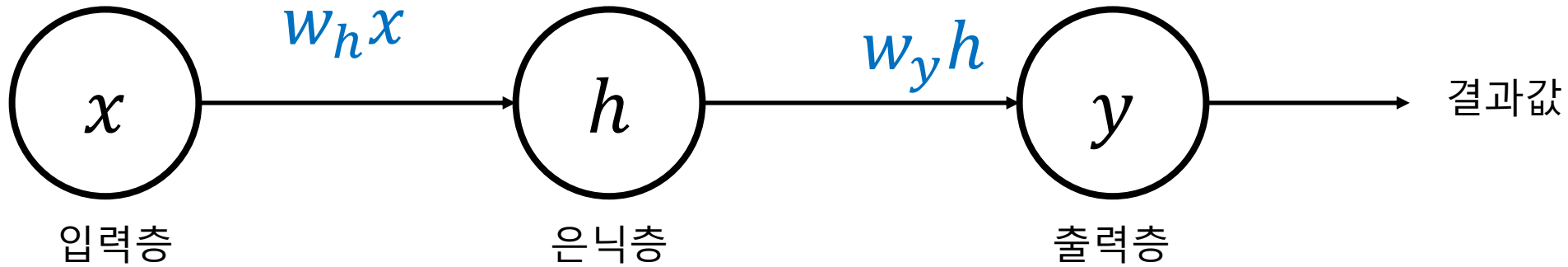
# 다층 퍼셉트론 (MLP, Multi Layer Perceptron)

입력층 하나와 은닉층(hidden layer)이라 불리는 하나 이상의 TLU층과 출력층으로 구성됩니다.

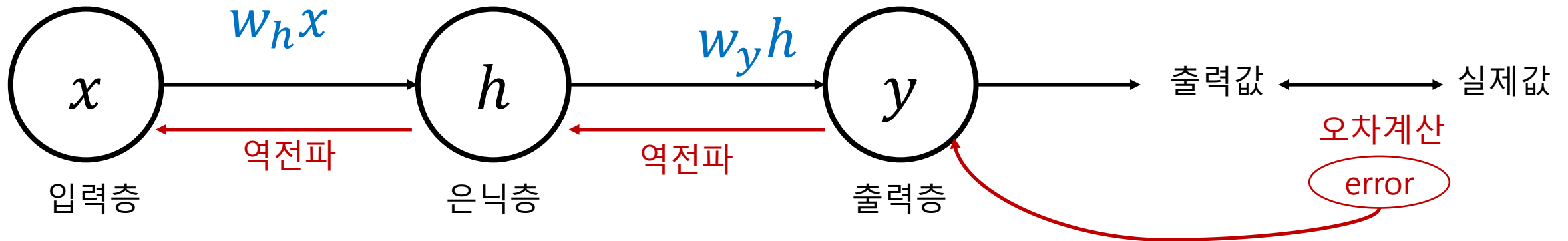


# 다층 퍼셉트론 훈련

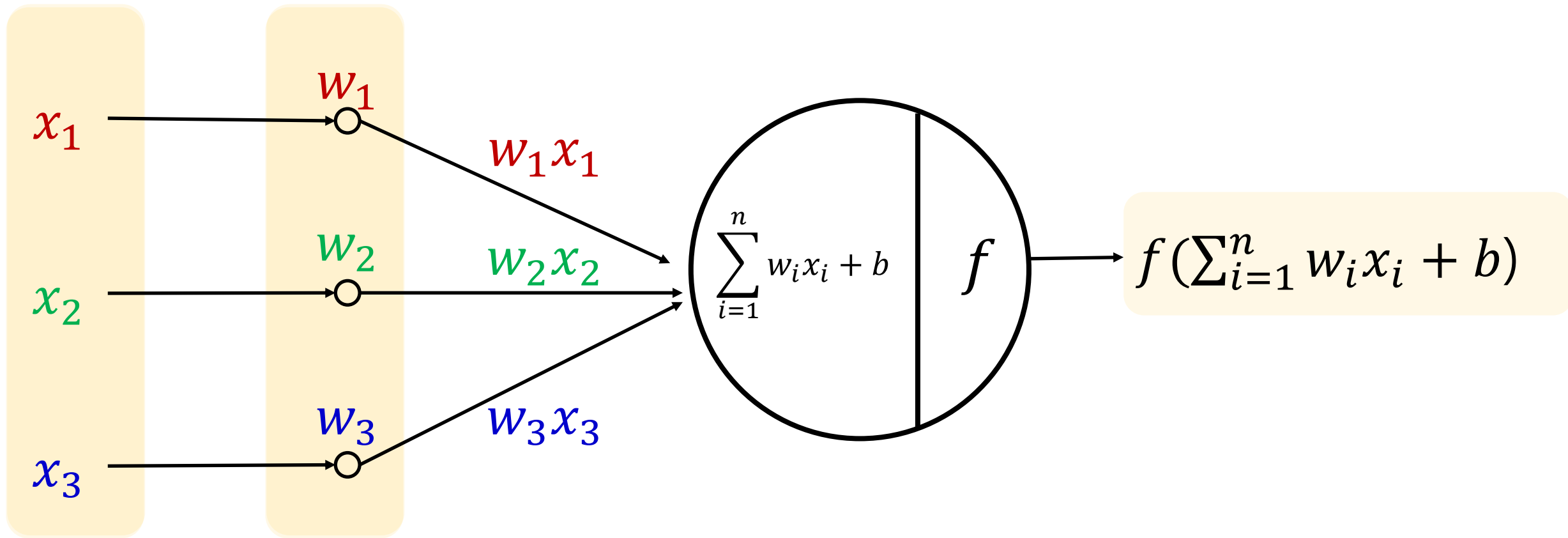
## ■ 신경망 순전파



## ■ 신경망 역전파

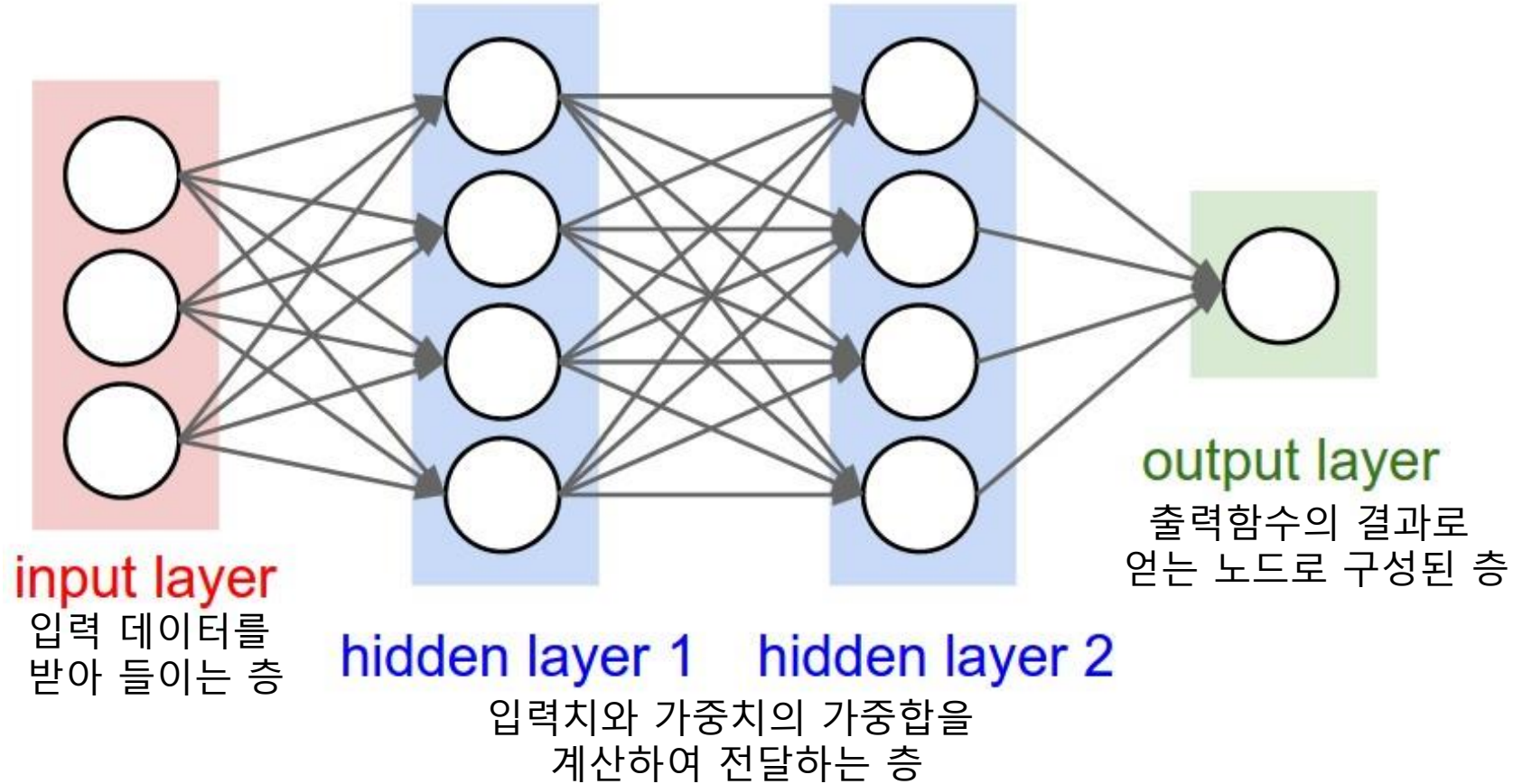


# 인공 뉴런 모델의 수학적 표현



# 심층신경망(Deep Neural Network, DNN)

- 딥러닝은 여러 층(layer)을 가진 인공신경망(Artificial Neural Network)을 사용하여 학습을 수행하는 것입니다.
- 심층신경망은 입력층과 출력층사이에 다수의 은닉층(hidden layer)을 포함하는 인공신경망입니다.
- 머신러닝에서는 비선형 분류를 위해 여러 trick을 사용하지만, DNN은 다수의 은닉층으로 비선형 분류가 가능





# 심층신경망 (Deep Neural Network, DNN)

## ■ Input Layer(입력층)

신경망의 첫 번째 레이어로서 입력 데이터를 수신합니다.

## ■ Hidden Layer(은닉층)

신경망에서 입력 레이어(특성)와 출력 레이어(예측) 사이에 위치하는 합성 레이어입니다.  
신경망에 하나 이상의 히든 레이어가 포함될 수 있습니다.

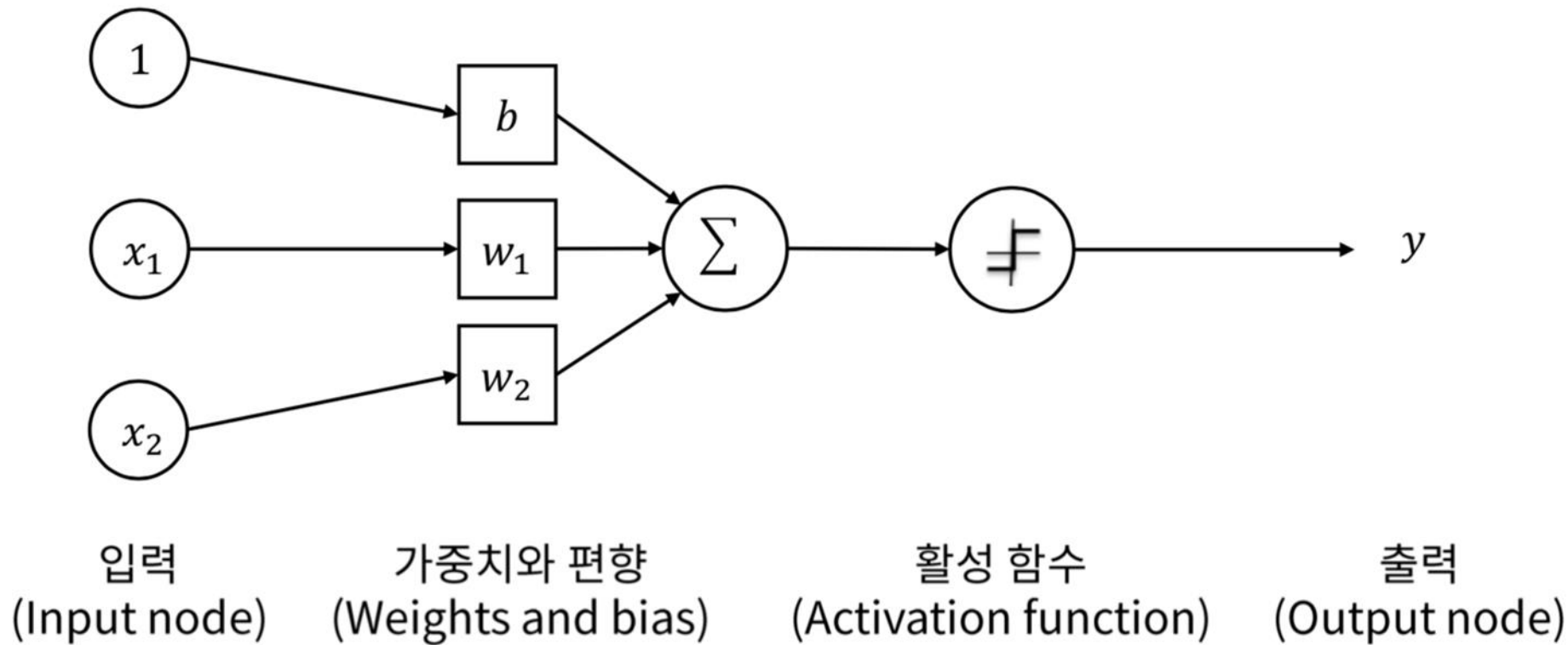
## ■ Output Layer(출력층)

신경망의 '최종' 레이어입니다. 이 레이어에 답이 포함됩니다.

## ■ Dense Layer(밀집층)

Fully Connected Layer(완전 연결층) 이나 Dense Layer(밀집층)라고 불리는 밀집 연결 층(Dense Connected Layer)

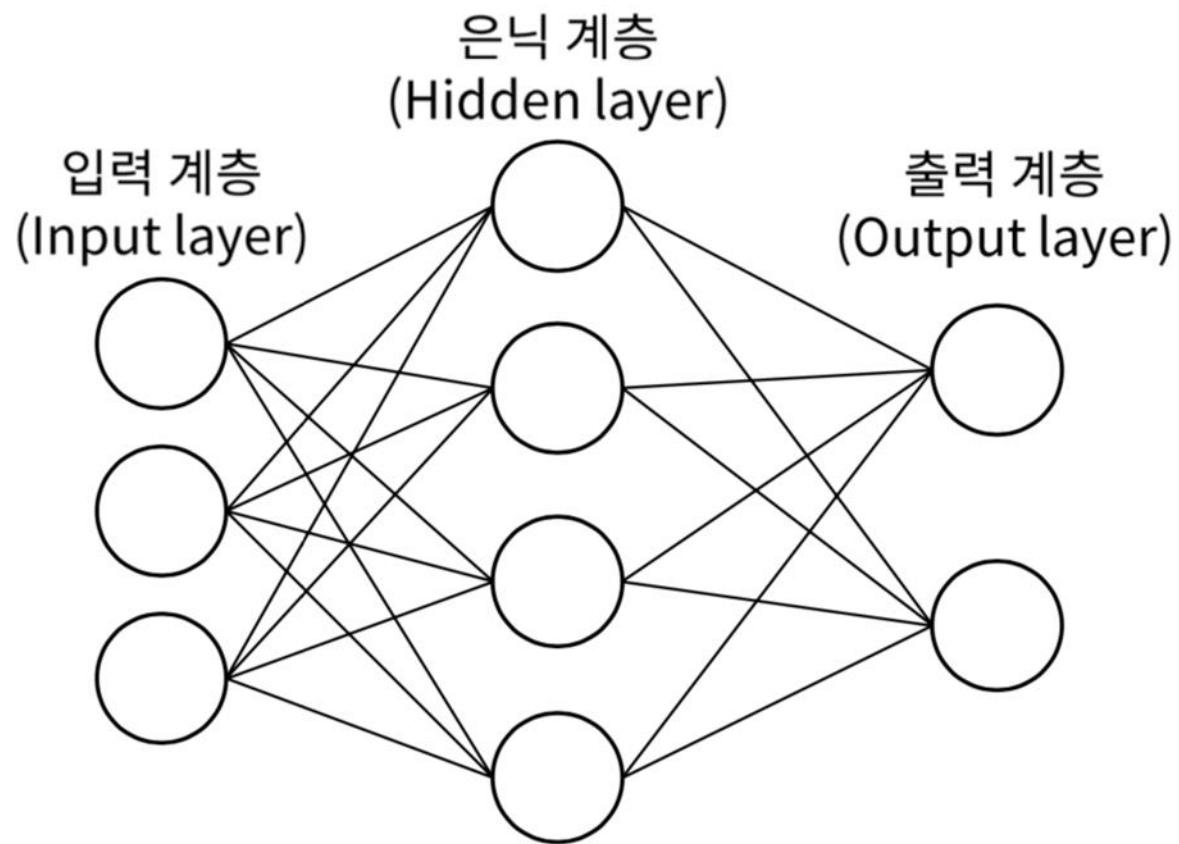
# 뉴런 Neuron



신경망은 뉴런을 기본 단위로 하며, 이를 조합하여 복잡한 구조를 이룬다.

출처 : [https://heung-bae-lee.github.io/2019/12/08/deep\\_learning\\_03/](https://heung-bae-lee.github.io/2019/12/08/deep_learning_03/)

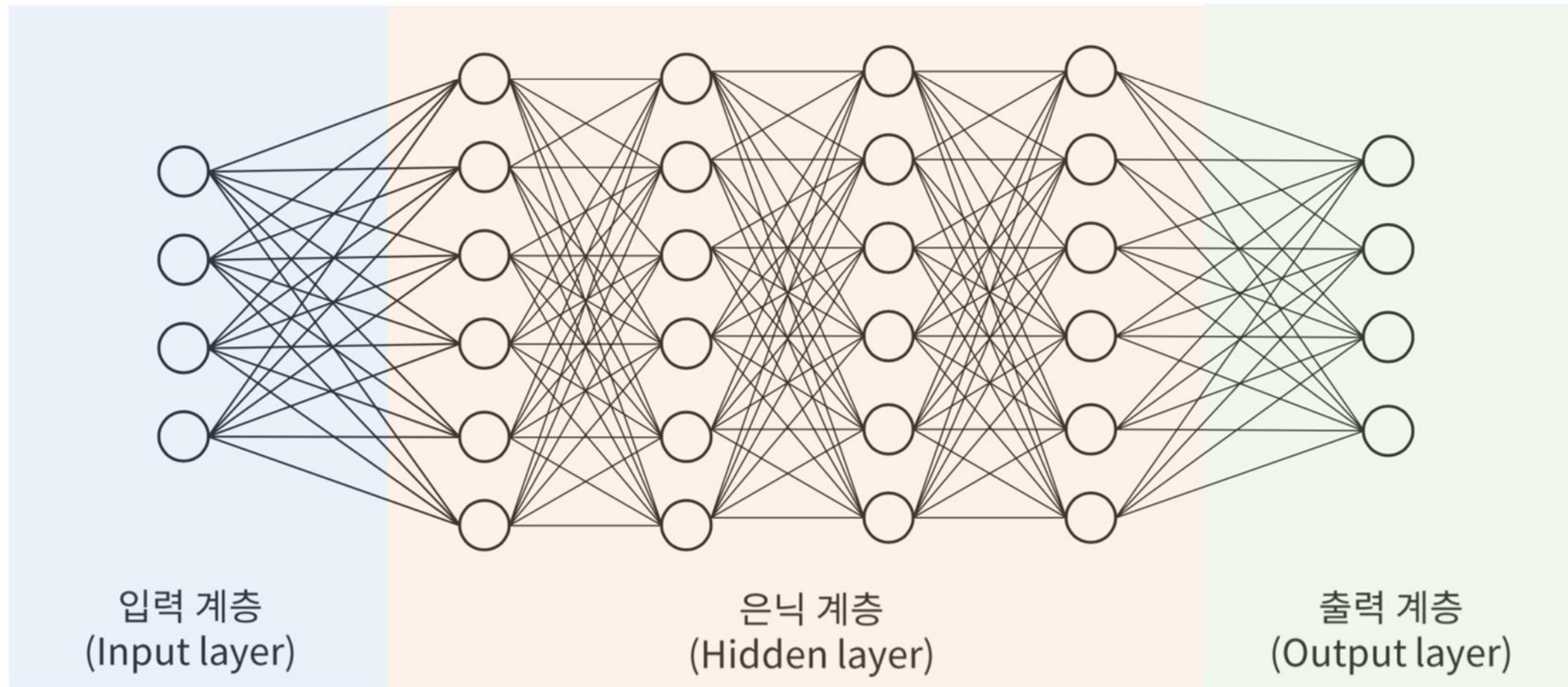
# 얇은 신경망 Shallow Neural Network



가장 단순하고 얇은(은닉 계층이 1개인) 신경망 구조를 얇은 신경망이라고 한다.

출처 : [https://heung-bae-lee.github.io/2019/12/08/deep\\_learning\\_03/](https://heung-bae-lee.github.io/2019/12/08/deep_learning_03/)

# 심층 신경망 Deep Neural Network (DNN)

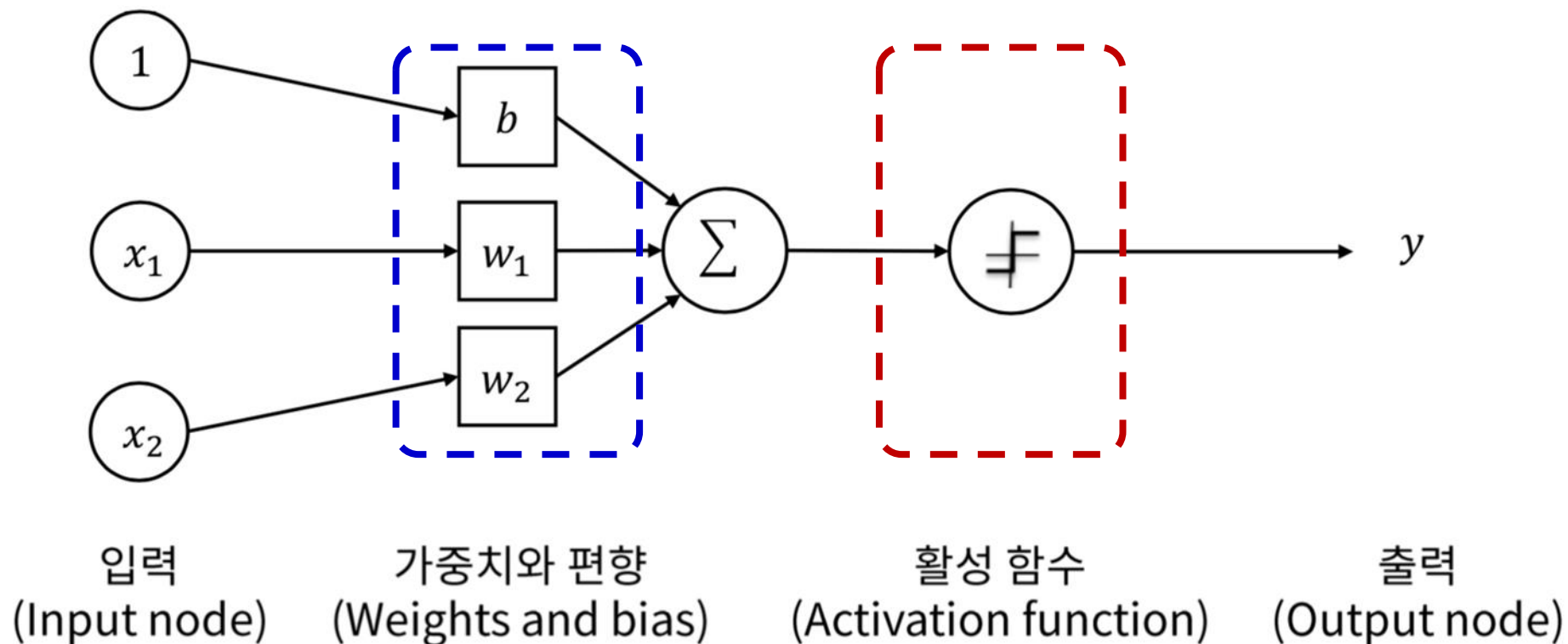


- 얇은 신경망보다 은닉 계층이 많은 신경망을 DNN이라고 부른다.



# 가중치 (Weight)

가중치는 입력값이 연산결과에 미치는 영향도를 조절하는 요소입니다.

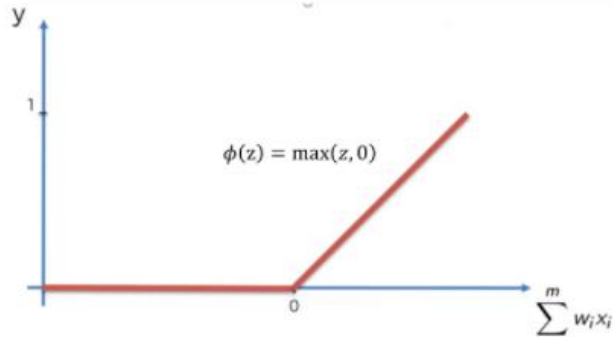


# 활성화 함수(Activation function)

입력값들의 수학적 선형결합을 다양한 형태의 비선형(또는 선형) 결합으로 변환하는 역할

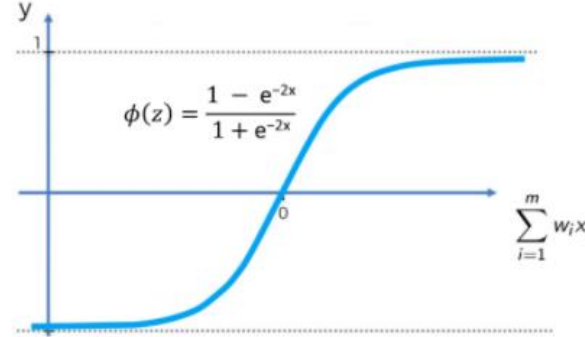
## 렐루 (ReLU)

입력이 양수일 때는  $x$ , 음수일 때는 0을 출력



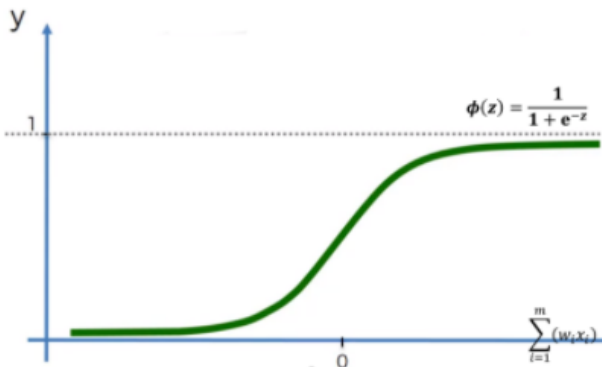
## 하이퍼볼릭 탄젠트 (Hyperbolic Tangent)

선형함수의 결과를 -1~1까지의 비선형 형태로 변경하는 함수



## 시그모이드 (Sigmoid)

0~1까지의 비선형 형태로 변경



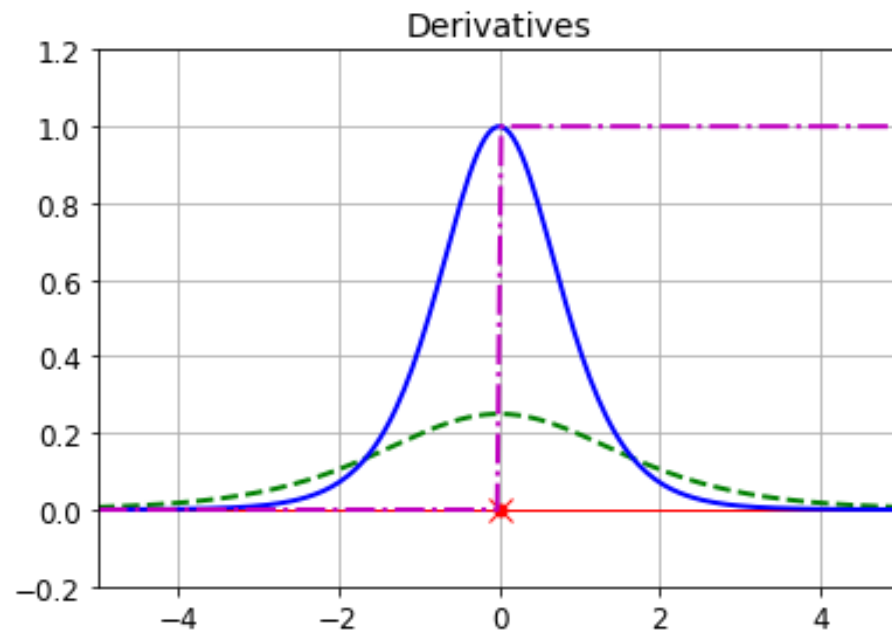
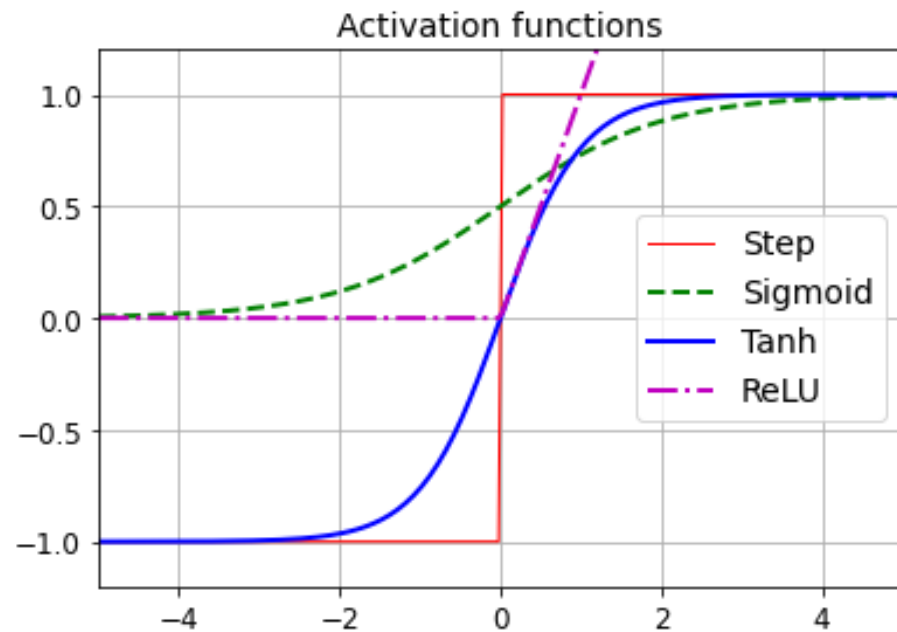
## 소프트맥스 (Softmax)

입력값을 0~1 사이 출력이 되도록 정규화, 출력값들의 총합은 항상 1

$$\phi(z) = \frac{e^i}{\sum_{j=0}^k e^j} \quad \text{where } i=0,1,\dots,k$$

# 활성화 함수(Activation function)

```
def sigmoid(z):  
    return 1 / (1 + np.exp(-z))  
  
def relu(z):  
    return np.maximum(0, z)  
  
def derivative(f, z, eps=0.000001):  
    return (f(z + eps) - f(z - eps))/(2 * eps)
```



# 손실함수 (Loss Function)

인공신경망 학습의 목적함수로 출력값(예측값)과 정답(실제값)의 차이를 계산합니다.

## ■ 회귀예측(Regression)

`torch.nn.MSELoss`

## ■ 이진분류(Binary Classification)

`torch.nn.BCELoss`

## ■ 다중분류( Binary Classification)

`torch.nn.CrossEntropyLoss`



# 원핫벡터 (One-Hot Vector, One-Hot Encoding)

고유 값에 해당하는 칼럼에만 1을 표시하고 나머지 칼럼에는 0을 표시하는 방법

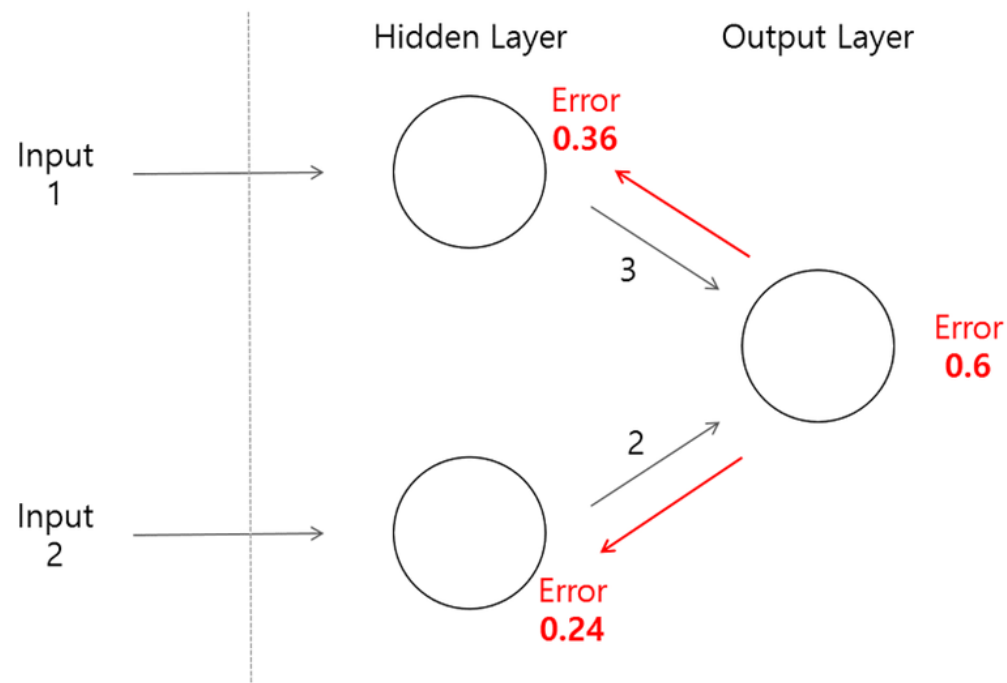
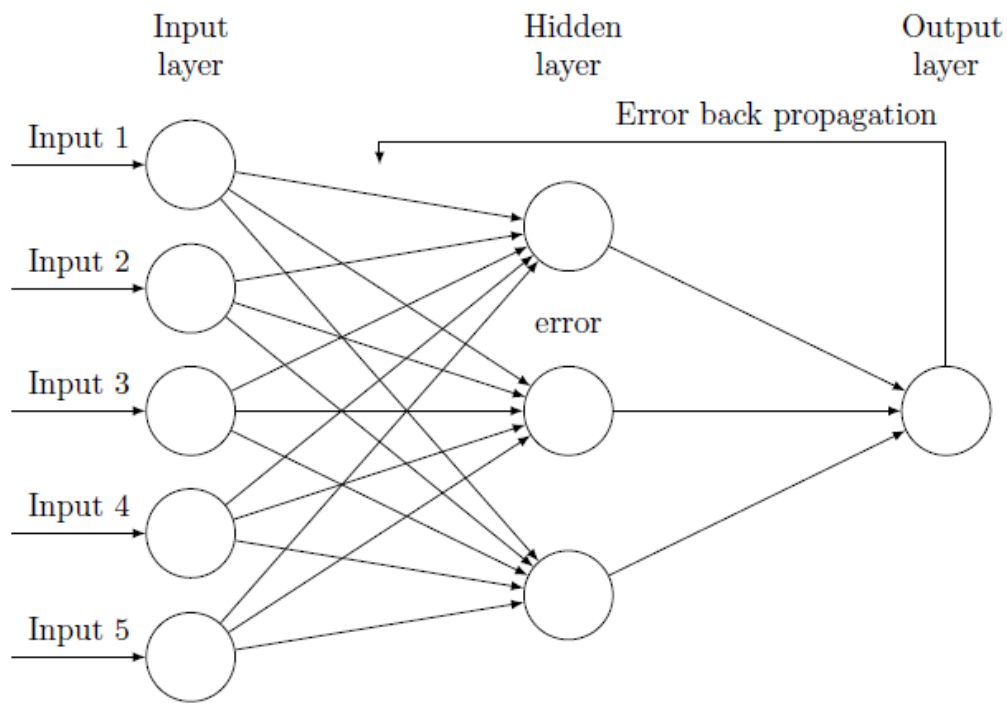
Human-Readable

Machine-Readable

Pet	Cat	Dog	Turtle	Fish
Cat	1	0	0	0
Dog	0	1	0	0
Turtle	0	0	1	0
Fish	0	0	0	1
Cat	1	0	0	0

# 딥러닝 학습방법

- 딥러닝 학습의 목표는 모델에 입력값을 넣었을 때의 출력값이 최대한 정답과 일치하게 하는 것입니다.
- 딥러닝 학습은 손실(Loss, Error)를 최소화 하는 가중치(weight)와 편향(bias)을 찾는 과정입니다.
- 모델 파라미터(weight, bias)를 무작위로 부여한 후, 반복학습(순전파-오차역전파)을 통해 모델의 출력값을 정답과 가깝게 되도록 매개변수(weight, bias)를 조금씩 조정합니다.
- 순전파(Forward Propagation)와 오차역전파(Error Back Propagation)의 반복으로 진행이 됩니다.

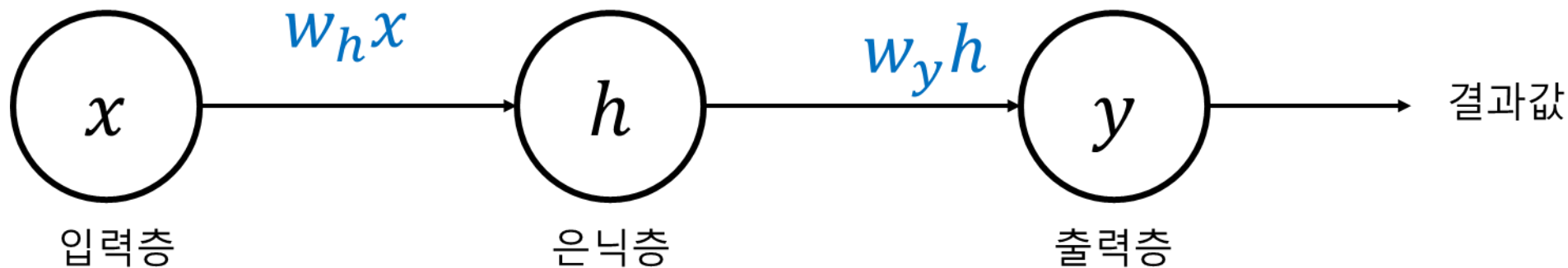


# 딥러닝 학습방법

## ■ 순전파(Forward Propagation)

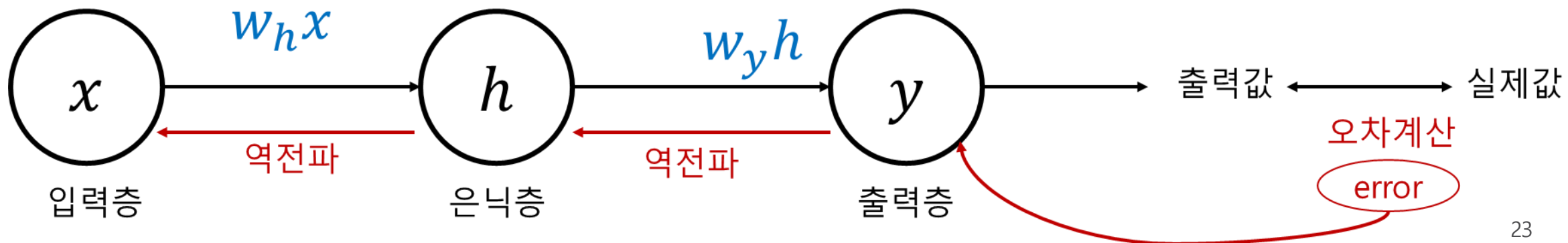
딥러닝 모델에 값을 입력해서 출력을 얻는 과정

순전파는 뉴럴 네트워크의 입력층부터 출력층까지 순서대로 변수들을 계산하고 저장



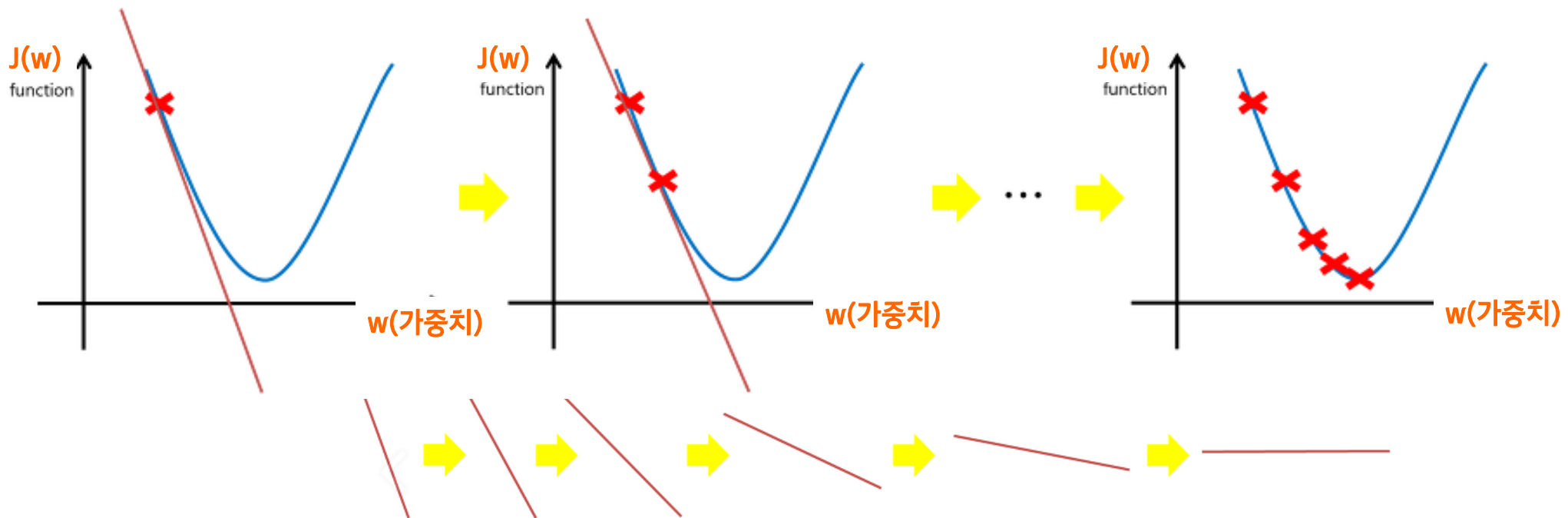
## ■ 오차역전파(Error Backpropagation)

실제값과 모델 결과값에서 오차를 구해서, 오차를 입력(input) 방향으로 보내서 가중치를 재업데이트 하는 과정



# 경사하강법 (Gradient Descent)

- 손실함수  $J(w)$ 는 가중치( $w$ )의 함수로, 볼록함수 형태라면 미분으로 손실이 가장 작은 가중치를 찾을 수 있습니다.
- 하지만, 딥러닝에서는 손실함수가 복잡하고 계산량이 매우 크고, 미분이 0이 되는 값이 여러 개 존재하므로 미분만으로 최소값을 찾기 어려워 경사하강법(Gradient Descent)을 사용합니다.
- 경사하강법은 손실함수의 현 가중치에서 기울기를 구해서 손실(Loss)을 줄이는 방향으로 업데이트 해 나갑니다.



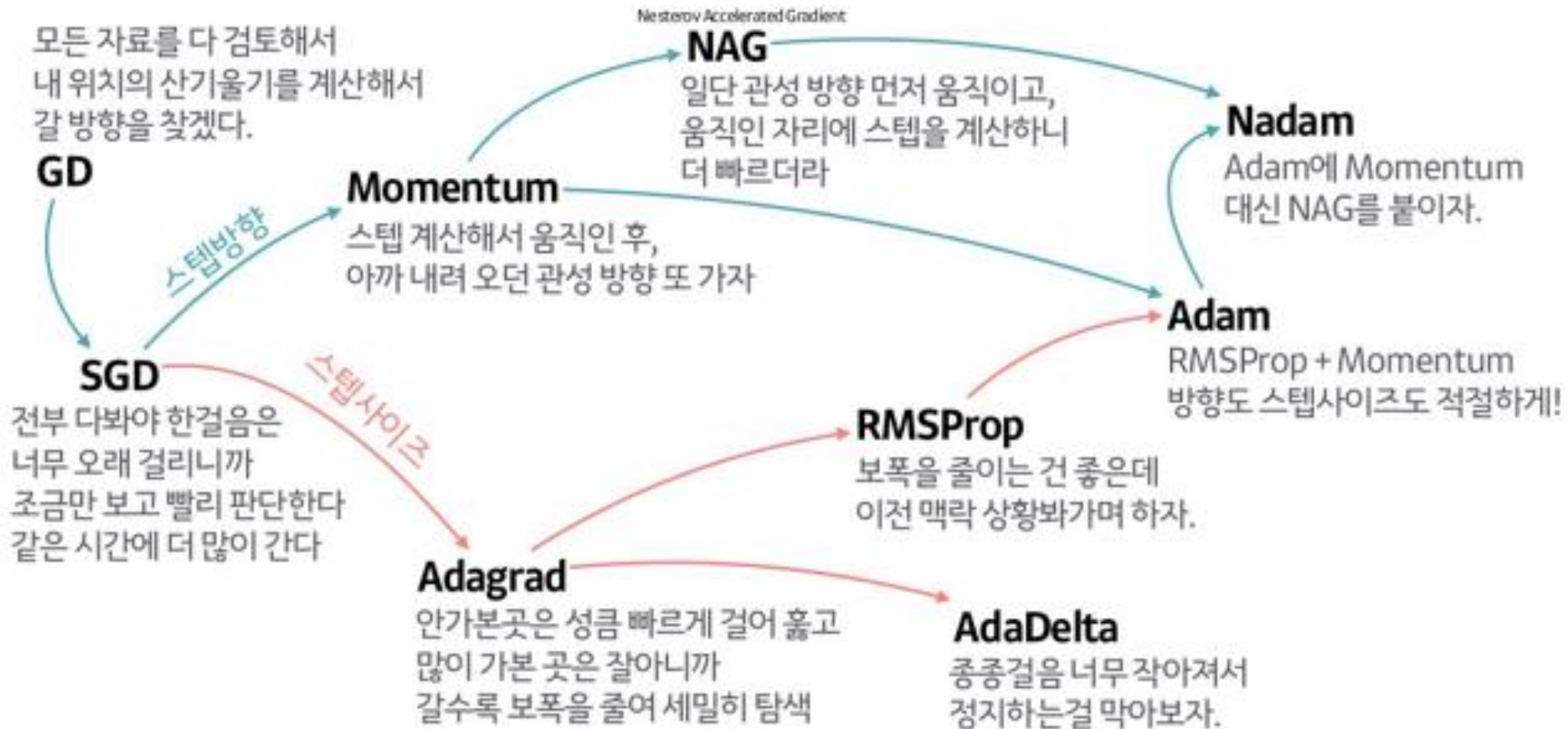
경사가 점차 감소되는 현상을 이용하므로 경사감소법!

참고 : [https://angeloyeo.github.io/2020/08/16/gradient\\_descent.html](https://angeloyeo.github.io/2020/08/16/gradient_descent.html)  
[https://angeloyeo.github.io/2020/08/16/gradient\\_descent.html](https://angeloyeo.github.io/2020/08/16/gradient_descent.html)



# 옵티마이저 (Optimization Algorithm)

손실함수를 최소화하는 방향으로 가중치를 갱신하는 알고리즘입니다.



# 옵티마이저 (Optimization Algorithm)

## ■ 손실함수(Loss Function)

손실함수는 신경망 학습의 목적으로(목적함수) 모델의 출력값(예측값)과 정답(실제값)의 차이를 계산합니다.

## ■ 최적화(Optimization)

딥러닝 모델의 파라미터(weight, bias)를 조절해서 손실함수의 값을 최저로 만드는 과정  
경사하강법(Gradient Descent)이 대표적입니다.

## ■ 경사하강법(gradient descent)

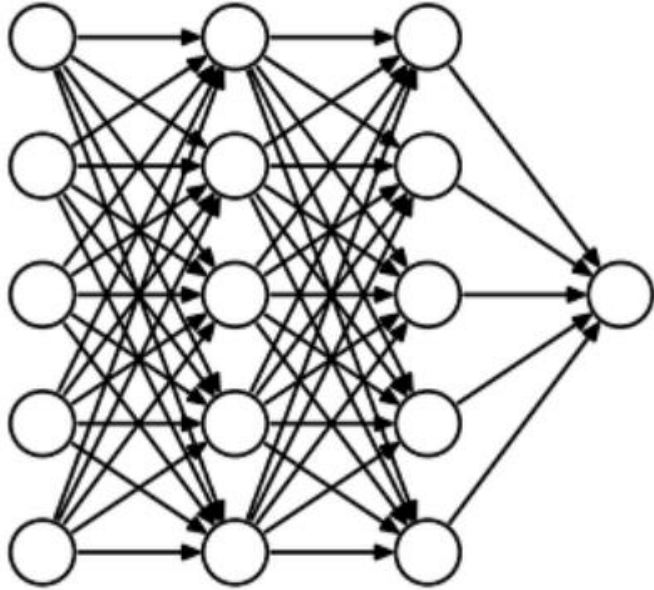
학습 데이터의 조건에 따라 모델의 매개변수를 기준으로 손실의 경사를 계산하여 손실을 최소화하는 기법입니다.  
경사하강법은 매개변수를 반복적으로 조정하면서 손실을 최소화하는 가중치와 편향의 가장 적절한 조합을 점진적으로 찾는 방식입니다.

## ■ 경사(gradient)

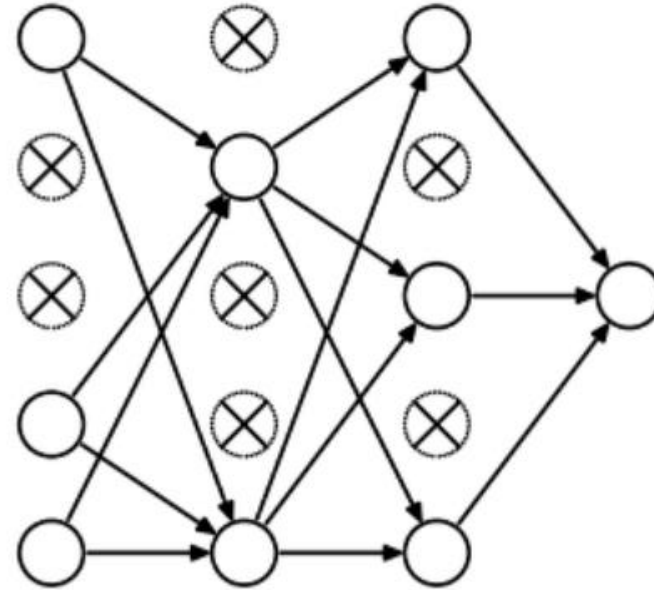
모든 독립 변수를 기준으로 한 편미분의 벡터입니다. 머신러닝에서 경사는 모델 함수의 편미분의 벡터입니다.

# 드롭아웃(Dropout)

Hidden Layer의 일부 유닛이 동작하지 않게 하여 overfitting(과적합)을 막는 방법



일반 심층신경망



Dropout이 적용된 심층신경망

## ■ 과적합(overfitting)

생성된 모델이 학습 데이터와 지나치게 일치하여 새 데이터를 올바르게 예측하지 못하는 경우입니다.

## ■ 일반화(generalization)

모델학습에 사용된 데이터가 아닌 이전에 접하지 못한 새로운 데이터에 대해 올바른 예측을 수행하는 능력을 의미합니다.



긴급과학  
Emergency Sci

이재원  
이철호

# 인공지능이란 무엇인가?

출처 : <https://www.youtube.com/watch?v=Ym3IM9y4U5U>

이재원  
이철호

# Thank you