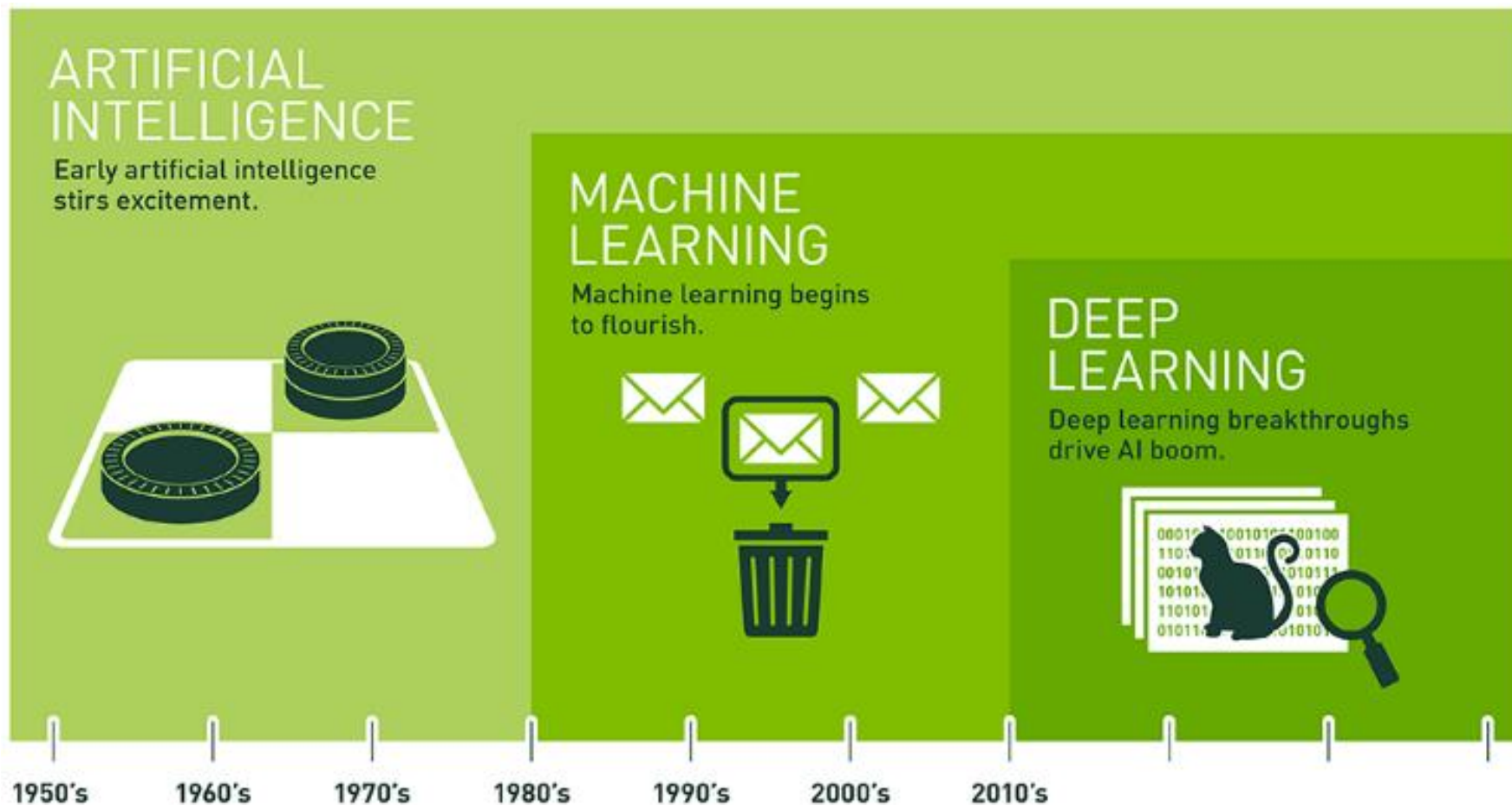


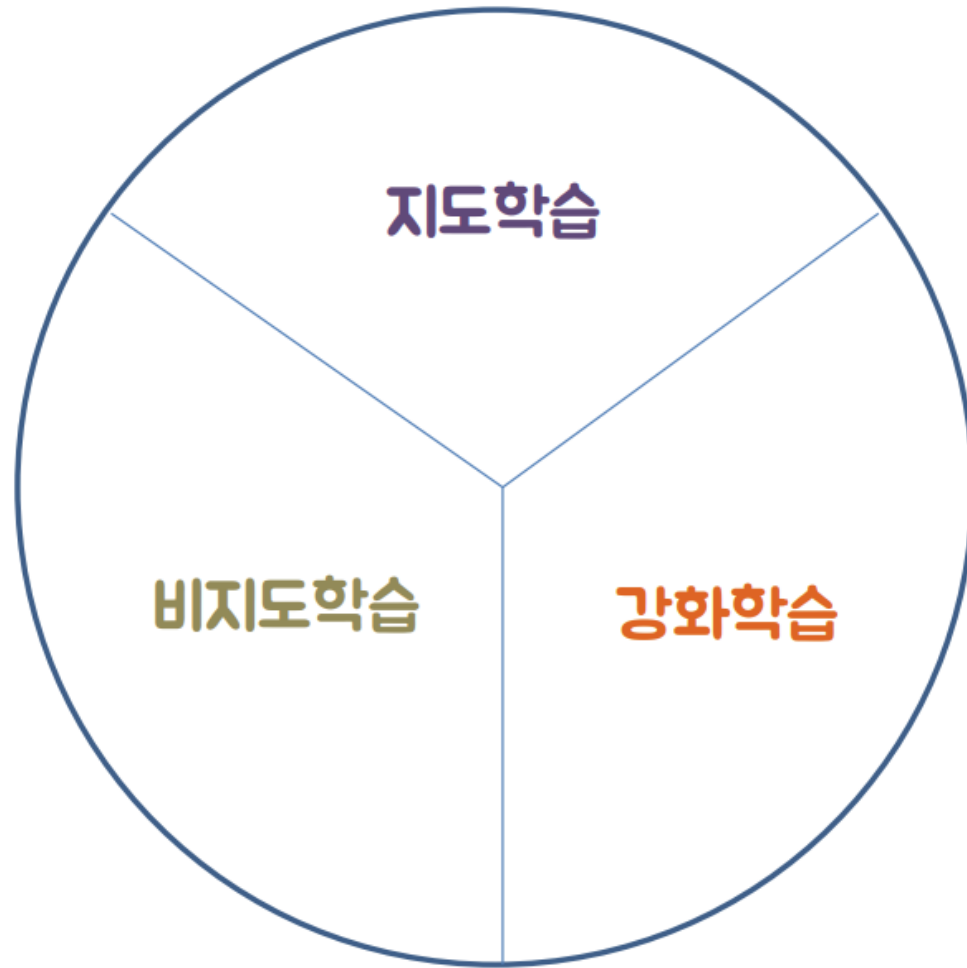
00

개요

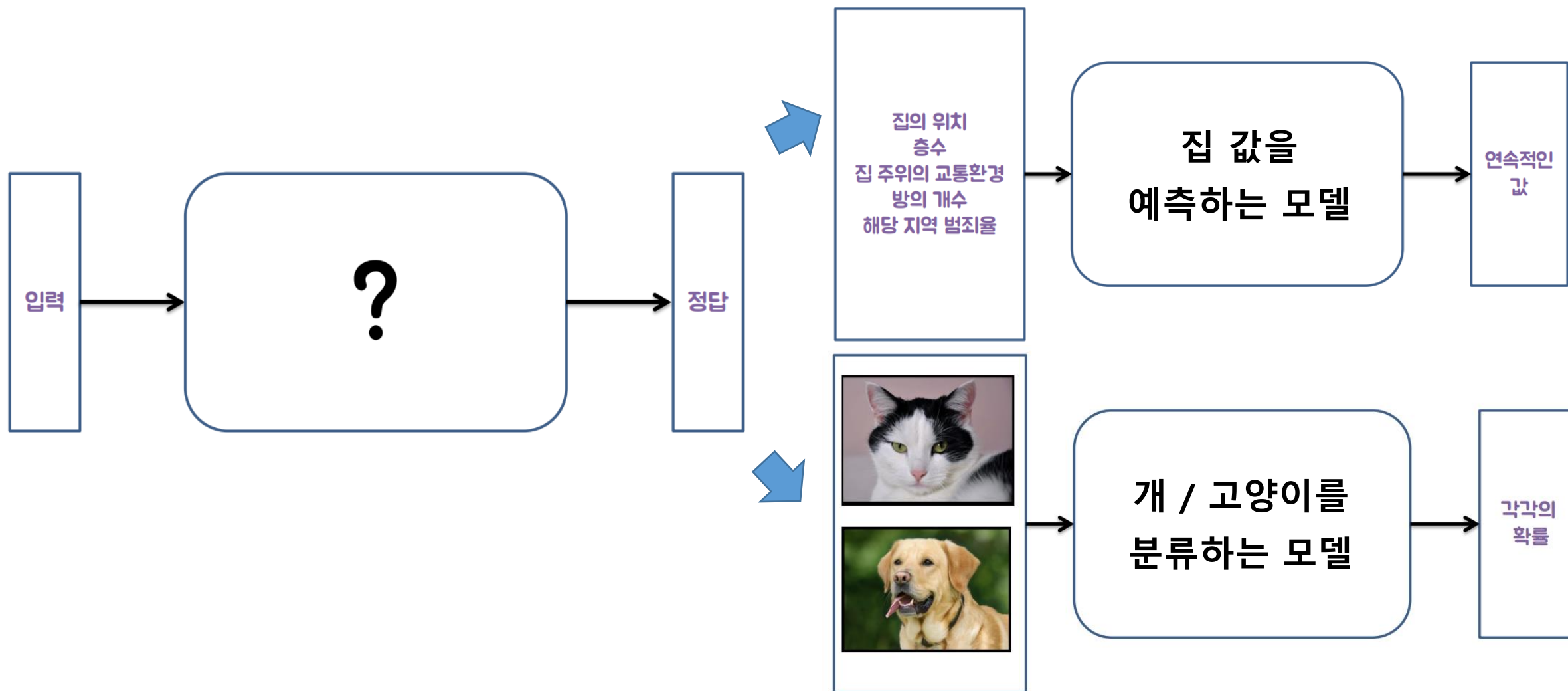
인공지능 / 머신러닝 / 딥러닝

- 딥러닝은 머신러닝의 한 종류
- 머신러닝은 인공지능의 한 종류



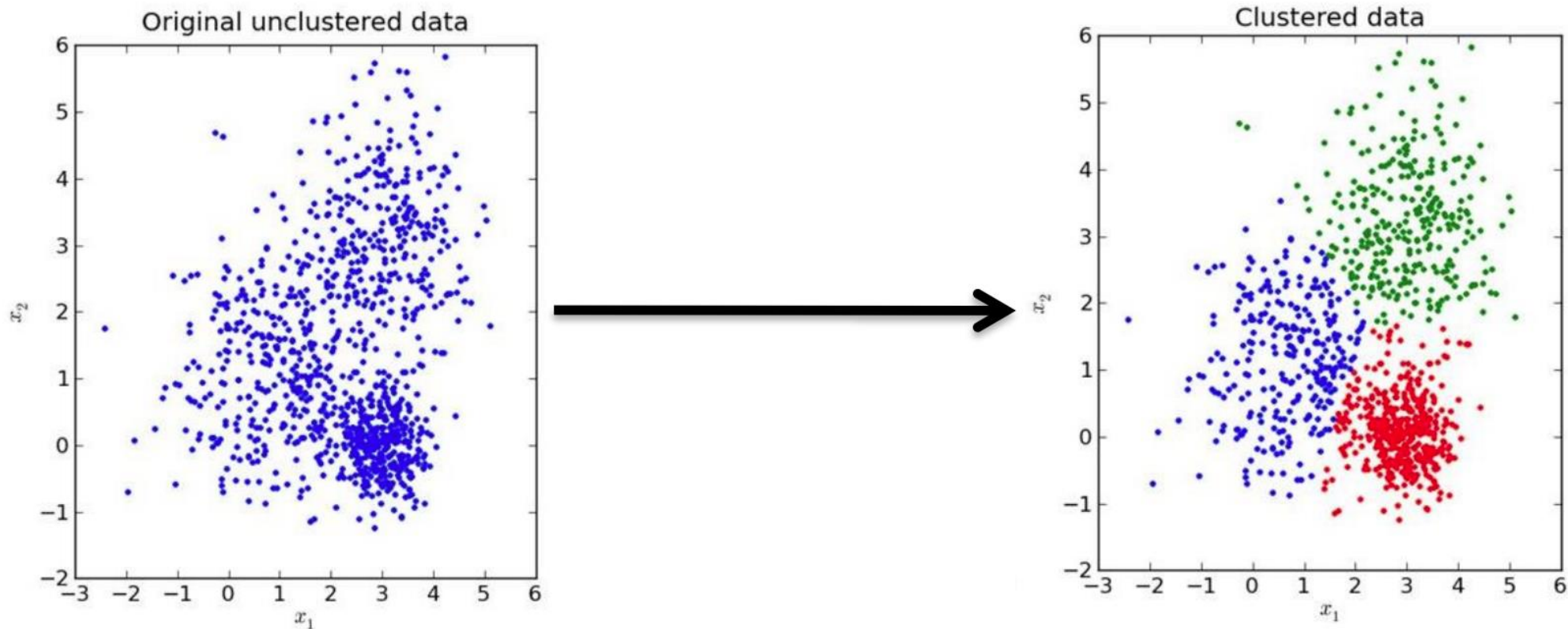


- 입력데이터와 정답데이터가 주어졌을 때,
데이터의 상관관계를 가장 잘 표현하는 함수를 구하는 것



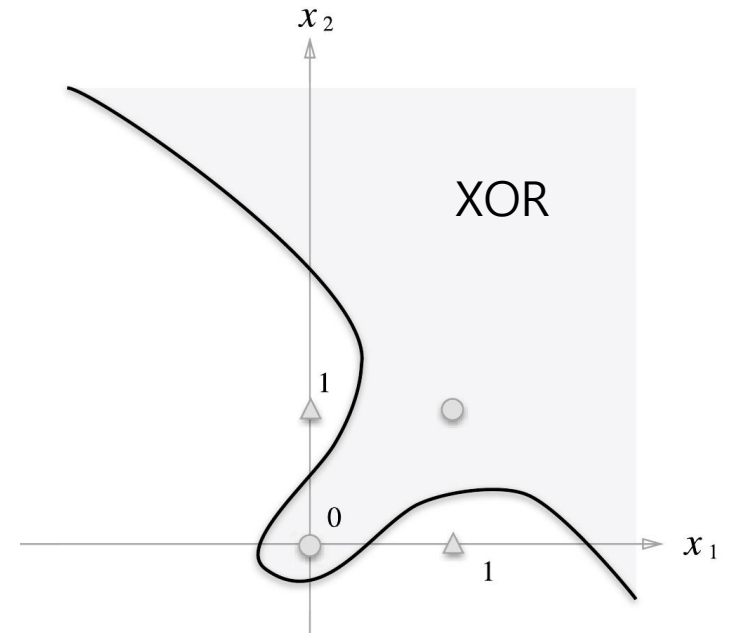
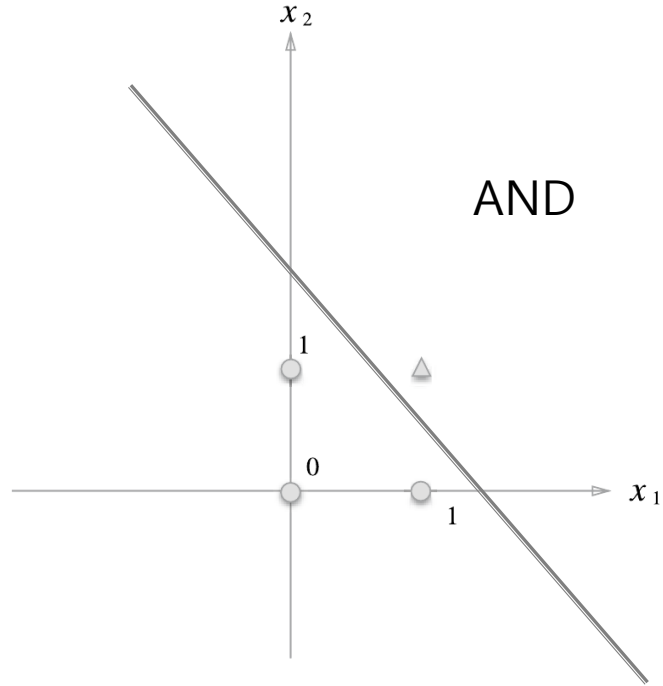
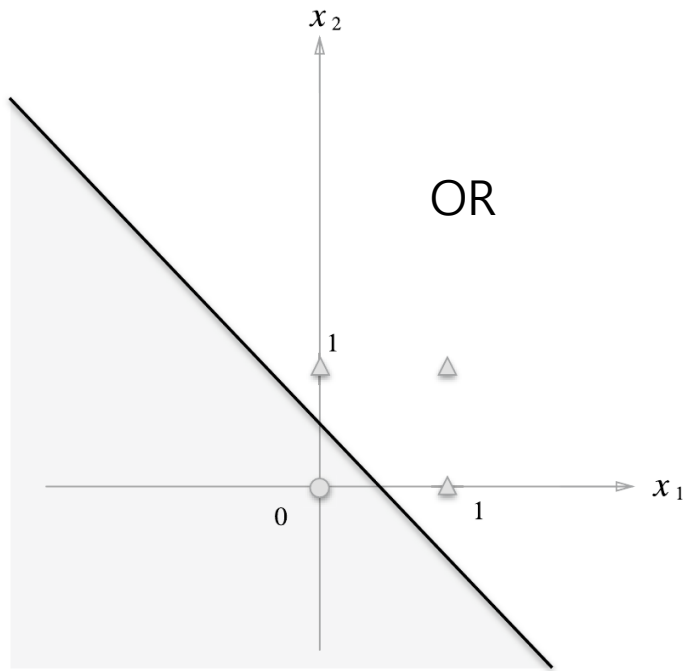
■ 입력데이터만 주어졌을 때 모델 스스로 데이터안에서 어떤 관계를 찾아냄

- 정답데이터가 주어지지 않음



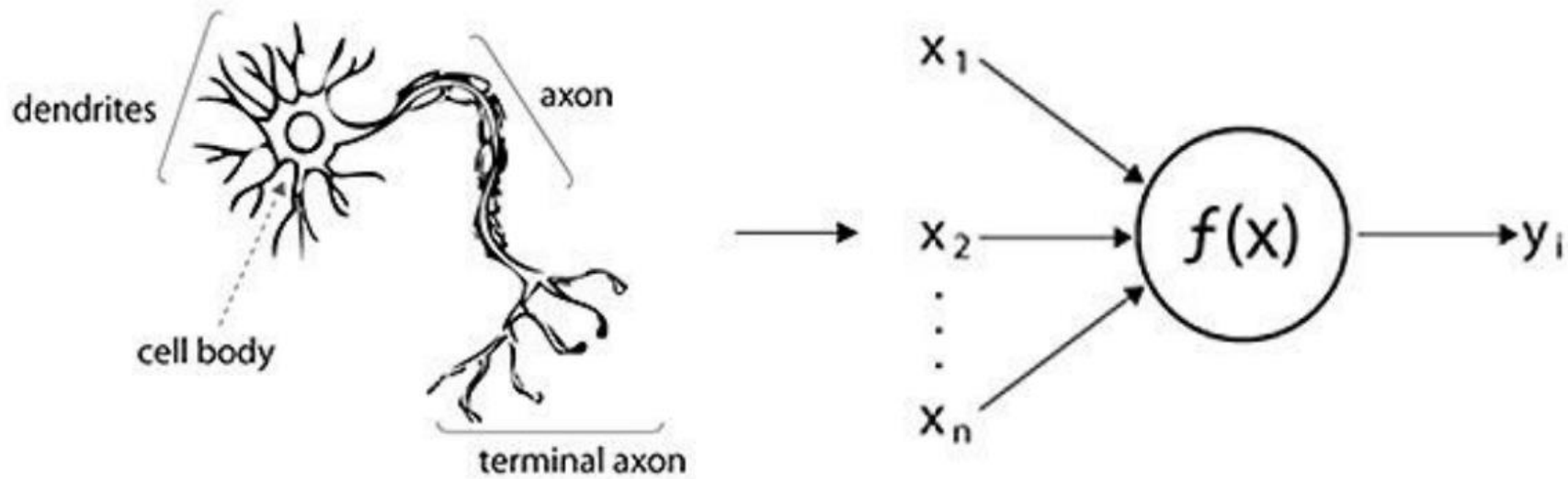
퍼셉트론

- 머신러닝을 이용한 학습방법은 오래전부터 존재
- 머신러닝이 Linear한 성질 때문에 XOR 같은 비선형 문제 해결 불가능

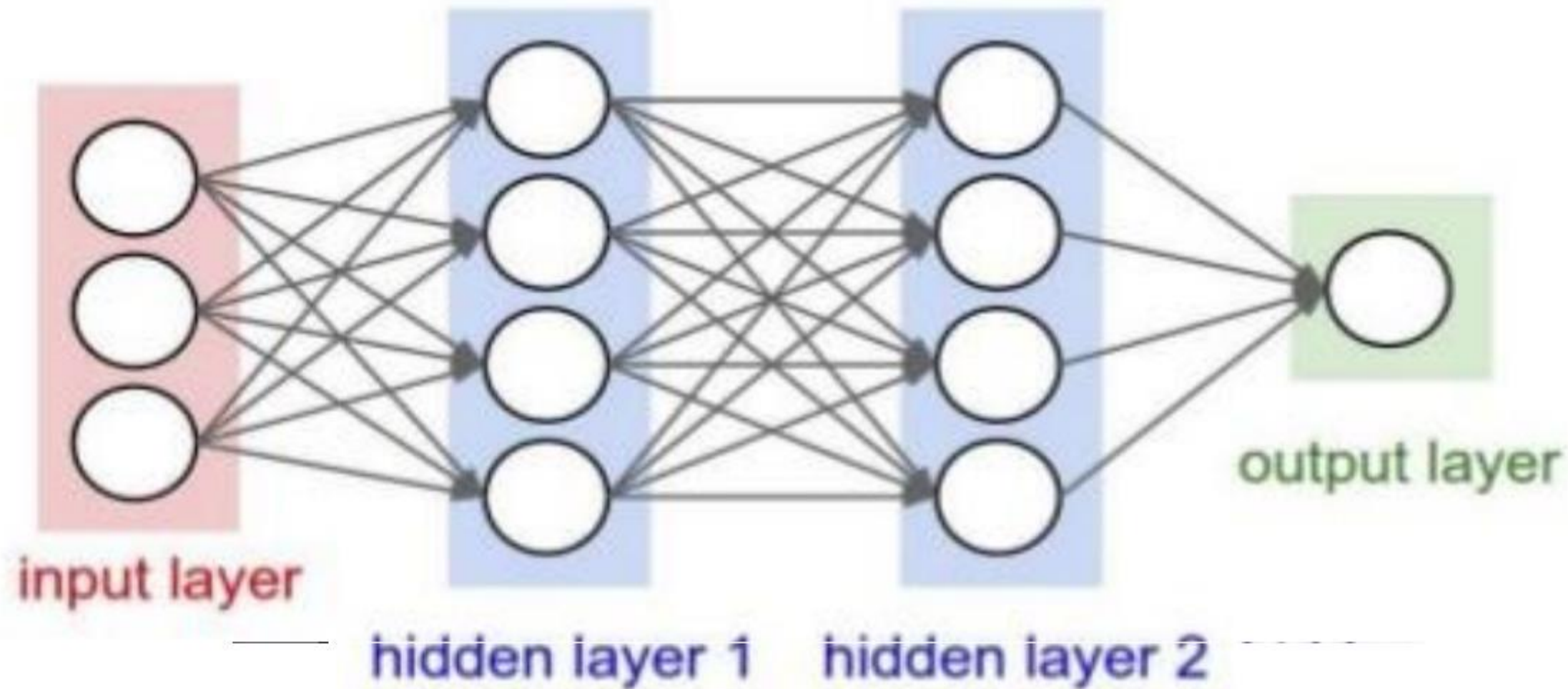


딥러닝

- 인간의 뇌를 본따서 만든 인공신경망을 이용해 학습하는 것
- 인공신경망을 넓고 다층(Deep)으로 쌓으면 딥러닝(DeepLearning)

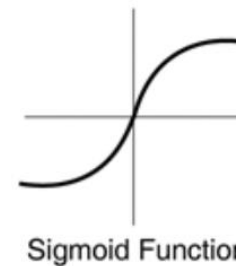
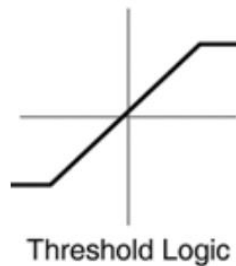
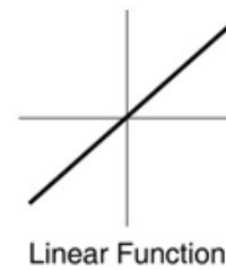
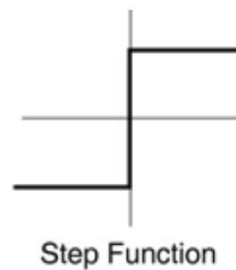
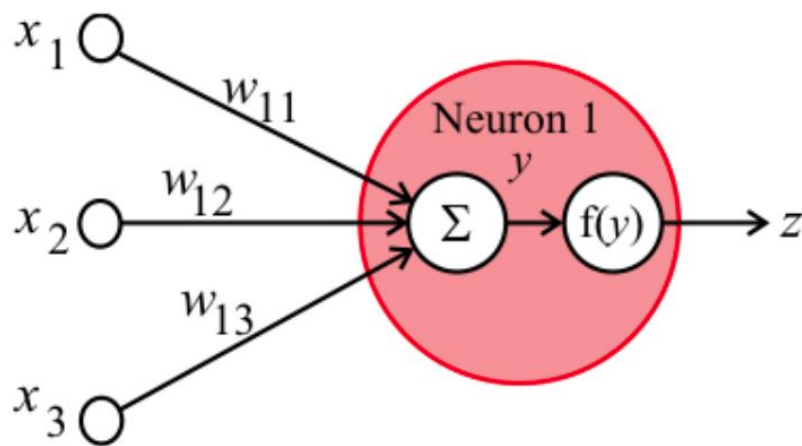


-
-

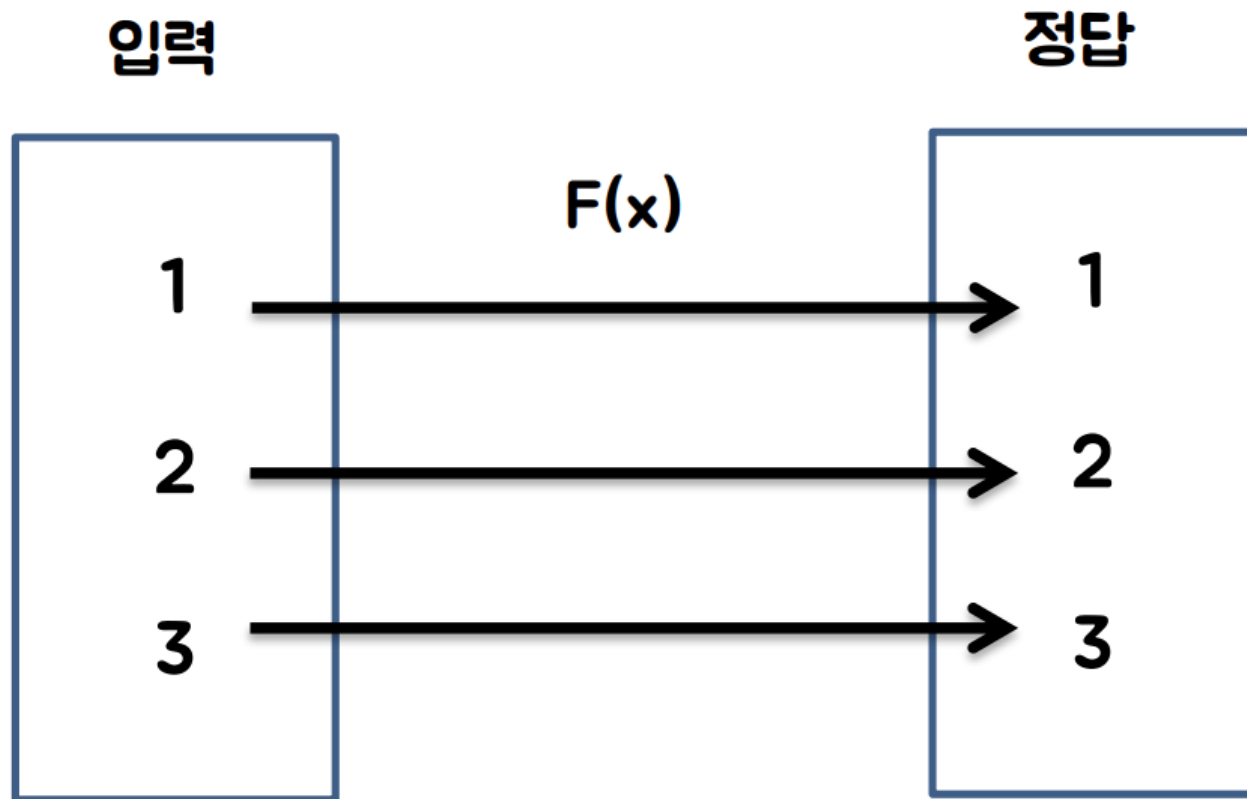


활성화 함수

- 어떤 입력이 들어와서 함수 출력이 나왔을때, 이함수를 활성화 하는 함수
- Linear 함수를 다층으로 쌓아도 선형함수
- 활성화 함수를 사용함으로써 비선형이 됨

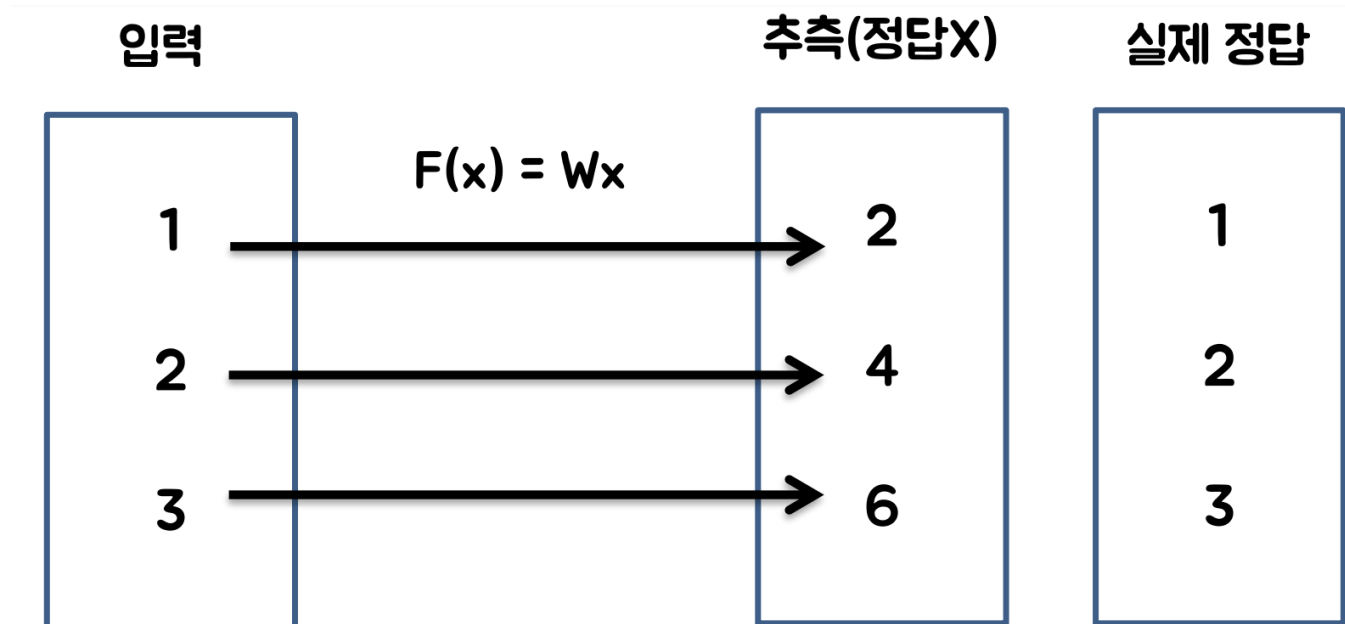


- 입력데이터와 정답데이터가 필요(지도학습)
- 입력데이터와 정답데이터에 맞는 최적의 함수를 찾아야 함



■ $F(x) = Wx$ 라는 일차식으로 표현

- 구해야 할 것은 가중치 w (weight) 이고 이 w 가 2라면 다음과 같은 결과가 나옴



■ 오차(loss / cost)를 줄이는 방법 사용

- Regression문제 : MSE(Mean Square Error)를 사용
- Classification문제 : Cross-Entropy 사용

출력(정답X)

2
4
6

정답

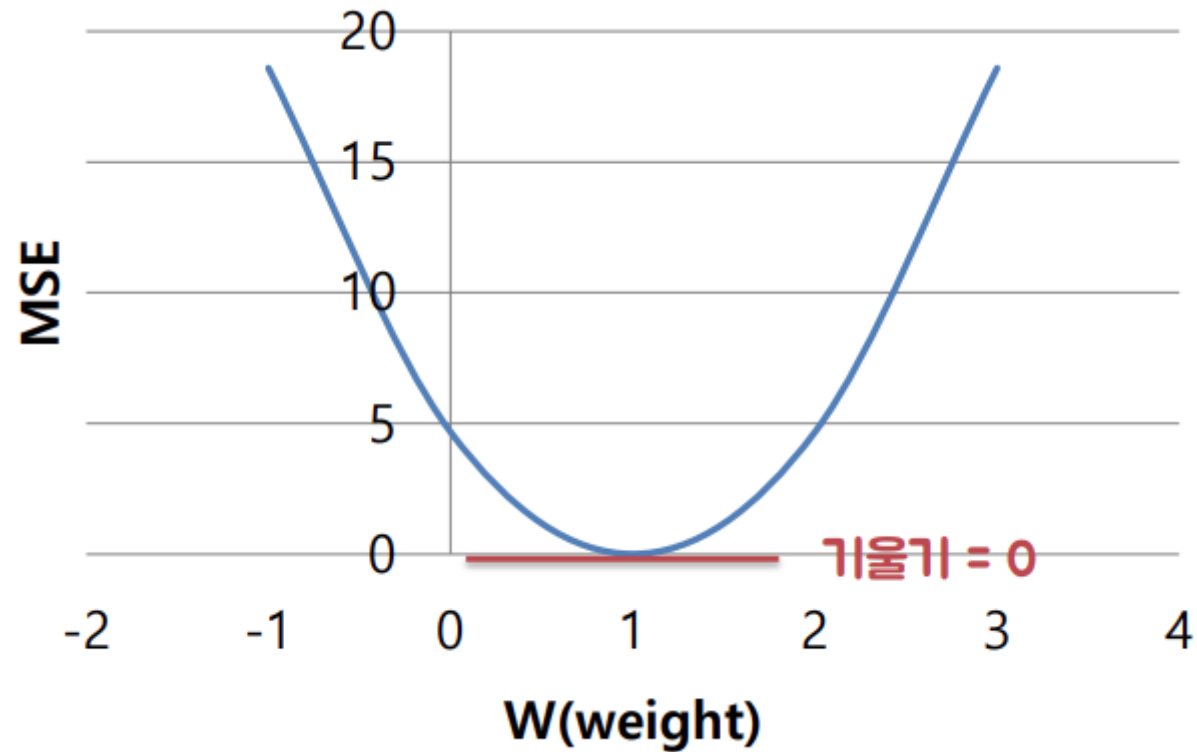
1
2
3

$$F(x) = 2(w) * X$$

$$\begin{aligned} \text{MSE} &= \frac{1}{n} \sum_{i=1}^n (y_i - \tilde{y}_i)^2 \\ &= \frac{1}{3} [(2 - 1)^2 + (4 - 2)^2 + (6 - 3)^2] \\ &= \frac{14}{3} = 4.67 \end{aligned}$$

■ MSE가 최소가 되는 지점을 구함

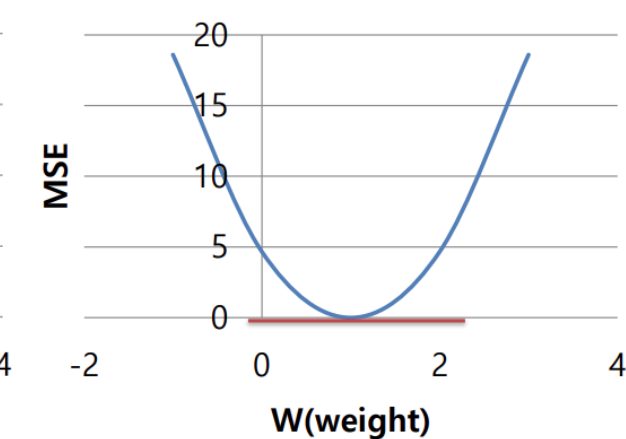
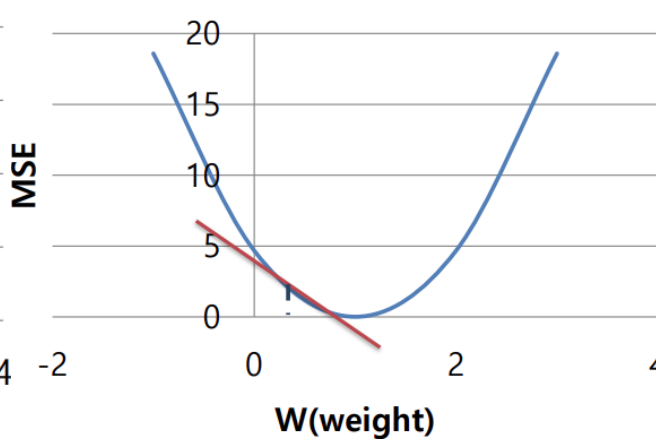
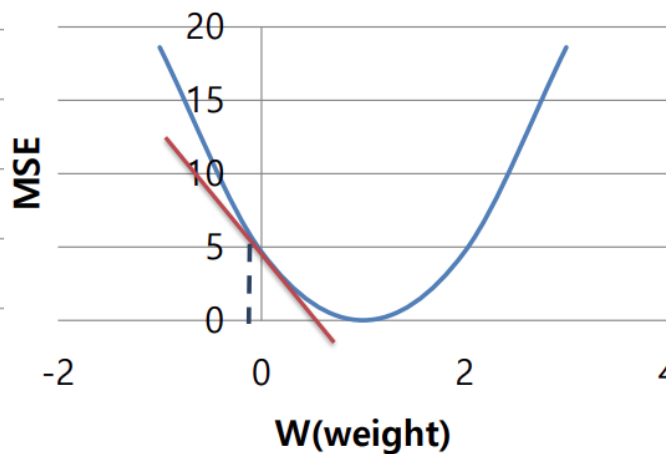
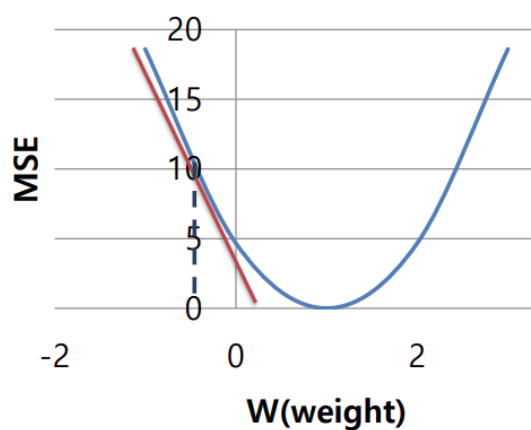
- 미분하여 그 기울기가 0이되는 w 가 해당 그래프의 최소값



경사하강법(Gradient Descent Algorithm)

■ 경사하강법

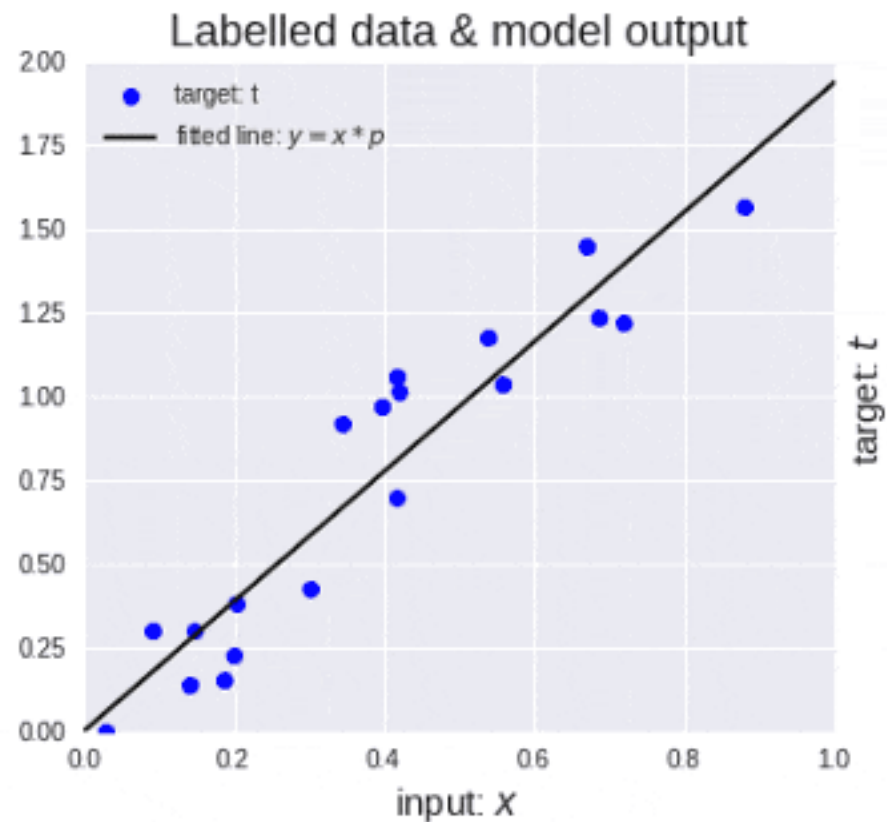
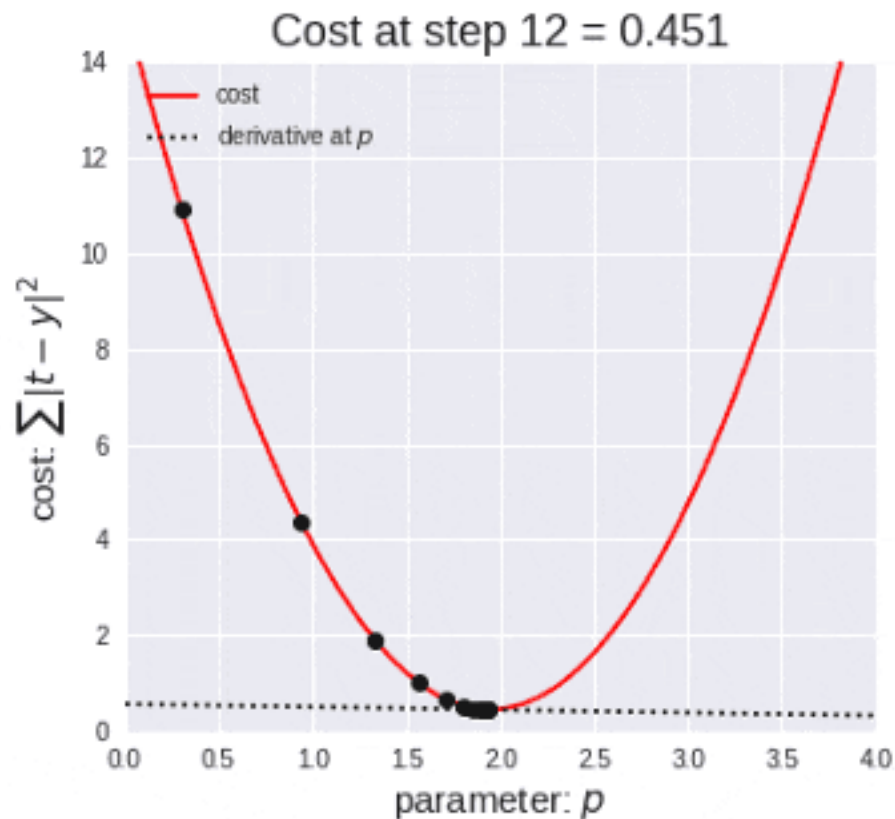
$$W = W - \alpha \frac{\partial E}{\partial W} \quad (\alpha : \text{Learning Rate 학습률})$$



경사하강법(Gradient Descent Algorithm)

■ 경사하강법

$$W = W - \alpha \frac{\partial E}{\partial W} \quad (\alpha : \text{Learning Rate 학습률})$$



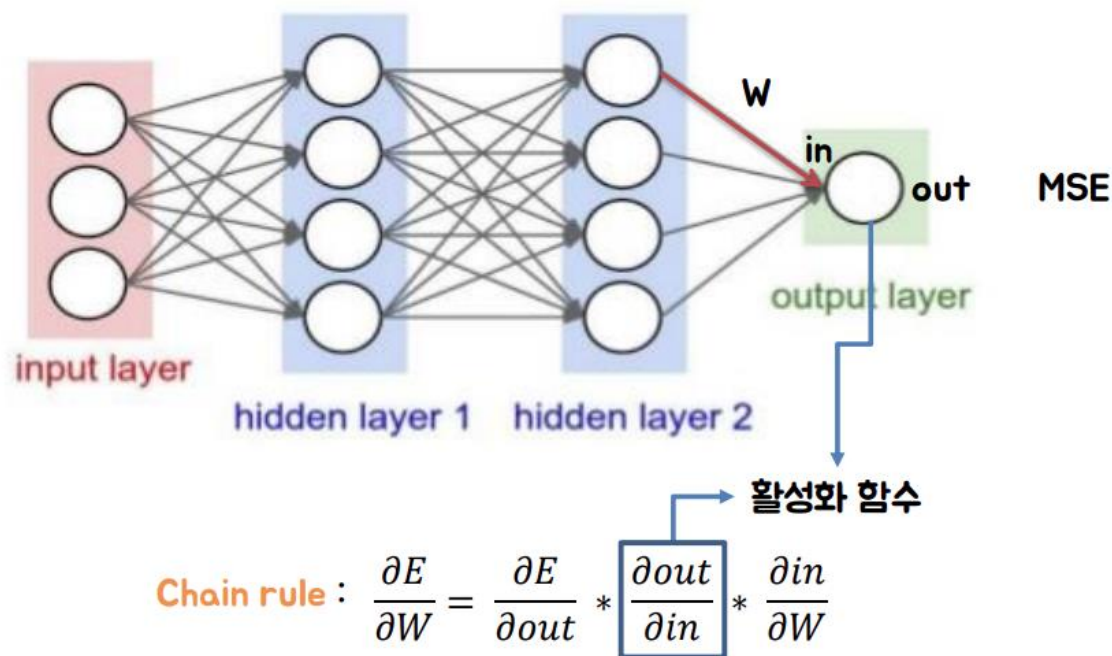
딥러닝의 문제점

■ 문제점

- 기존의 퍼셉트론은 층이 하나라서 미분가능
- 딥러닝은 층이 깊어지면서 사실상 미분이 불가능
- 더 이상 발전못함 -> 역전파(backpropagation)으로 해결

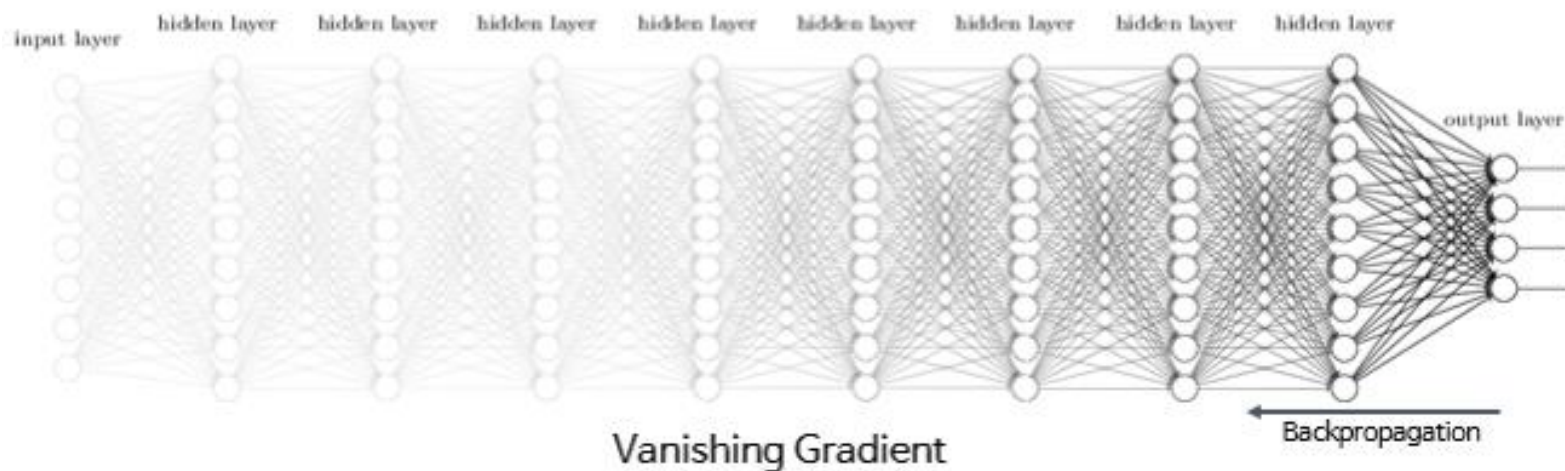
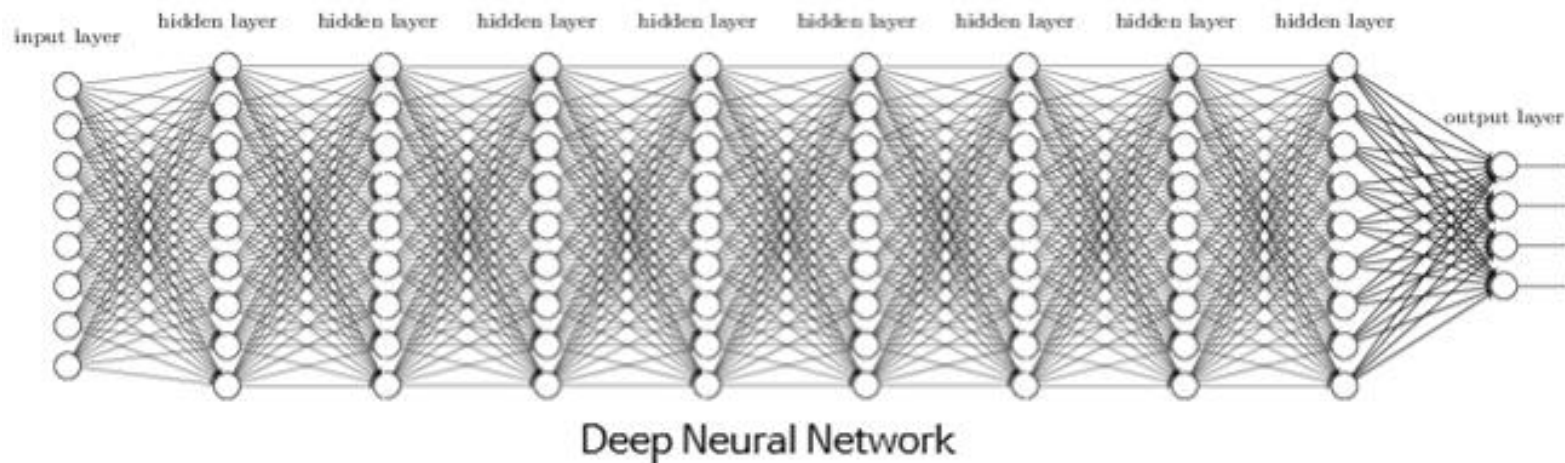
■ 역전파

- 딥러닝의 가장 중요한 개념
- 복잡한 형태의 함수를 Chain rule을 이용하여 기울기를 구할 수 있음



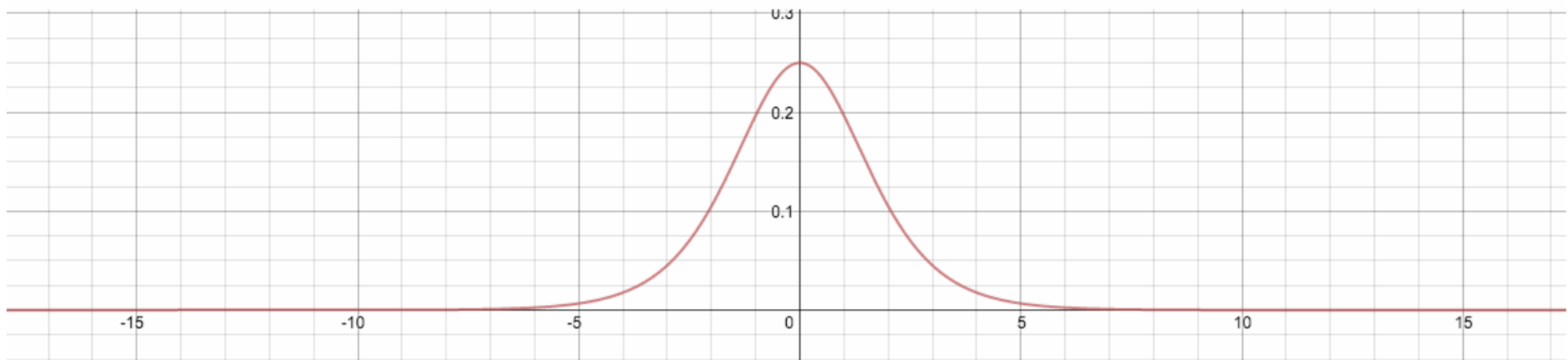
■ Vanishing Gradient

- 층을 깊이 쌓고 역전파를 통해 복잡한 문제를 해결
- 층이 깊어지면서 역전파로 오류가 입력단까지 전달되지 못하는 문제 발생



■ Vanishing Gradient의 원인

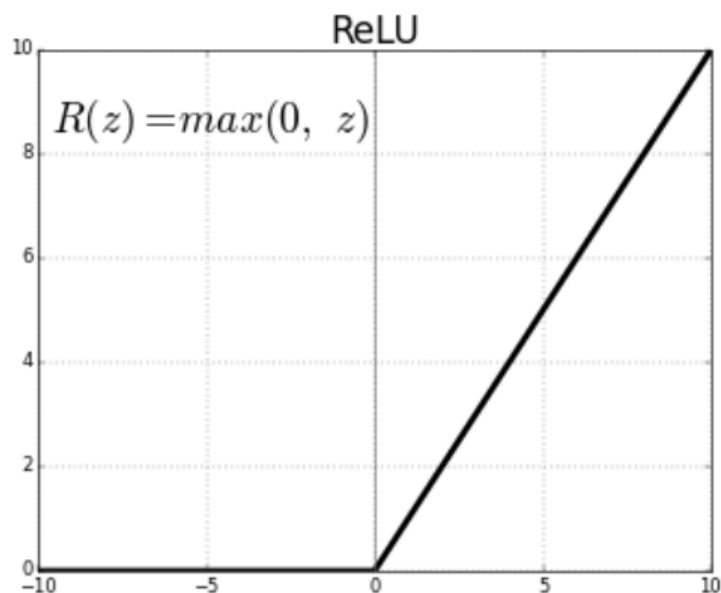
- 비선형 문제를 풀기위한 활성화 함수
- 모든 출력을 $[0,1]$ 사이로 압축해 버리는 sigmoid특성때문에 기울기가 거의 사라짐
- 미분값의 최대가 0.25
- 연속해서 곱하면 0으로 수렴



Sigmoid의 미분

■ Relu의 출현

- 아주 간단한 활성화 함수 추가
- 0보다 작으면 0, 0보다 크면 그값을 그대로 출력해 주는 활성화 함수
- ReLU(Rectified Linear Unit: ReLU) : 미분하면 계단함수가 나옴, 기울기가 사라지지 않음

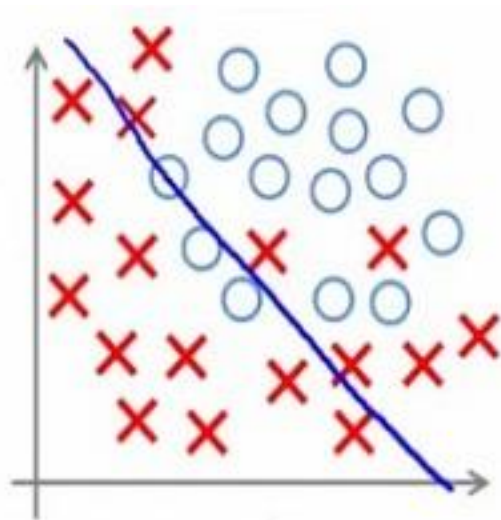


ReLU의 미분

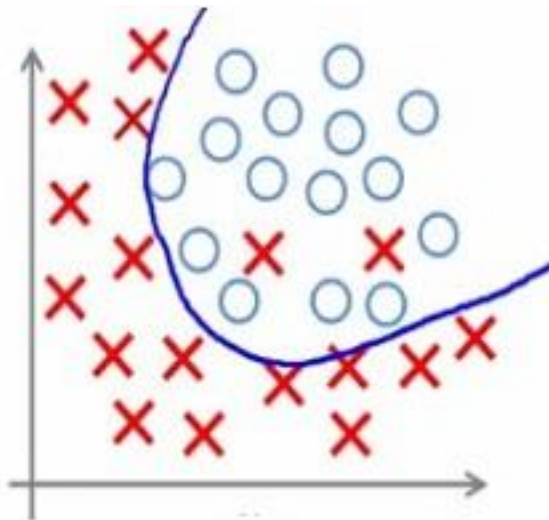
과적합(Overfitting)

■ 학습 데이터(테스트 셋)으로 학습을 계속하면 과적합 문제가 발생

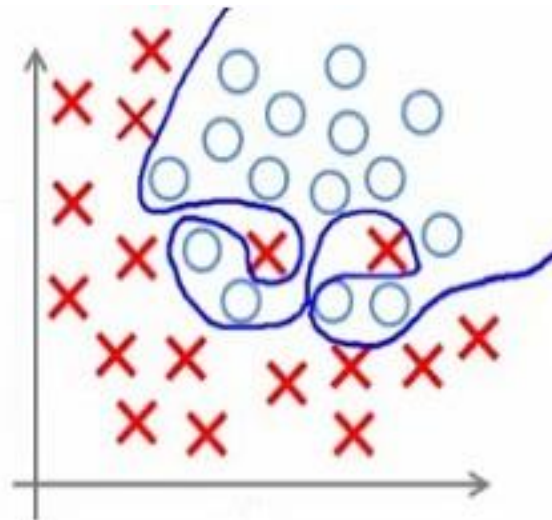
- 학습데이터에 최적화 됨 : 실제 데이터와 차이가 발생하는 모델이 만들어짐
- 과적합 방지하는 기법들을 적용해야함
- 반대의 경우로 언더피팅(Underfitting)



Under-fitting



Appropriate-fitting



Over-fitting