가중치 업데이트

멀티캠퍼스

가중치 업데이트

- 가중치를 어떻게 업데이트 해야 하는가?
- 무차별 대입 (brute force)
  - 크래킹 (cracking)

    ex) -1 ~ 1 사이의 값 1000가지 중 하나일 경우

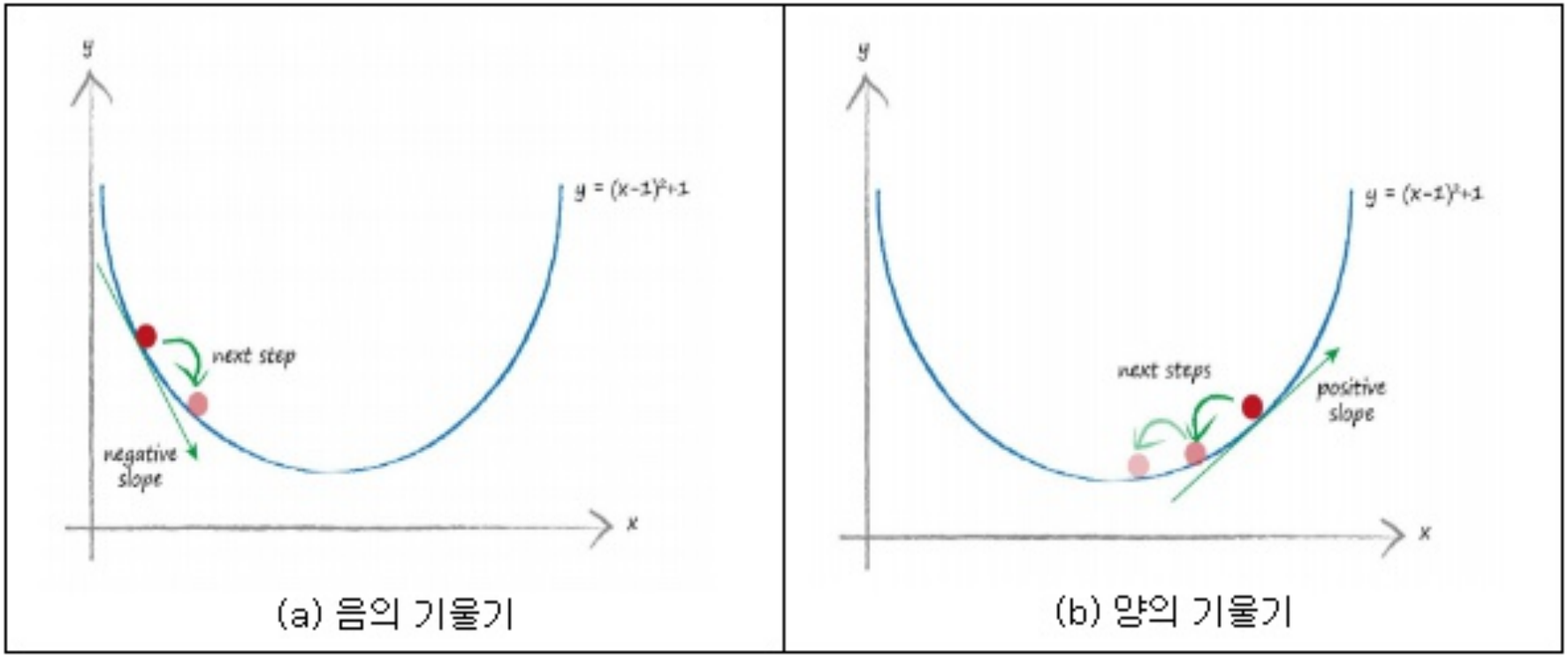
    ➔ 3개의 노드를 가지는 3개의 계층으로 구성되 신경망에는 총 18개의 가중치

    18 * 1,000 = 18,000 개

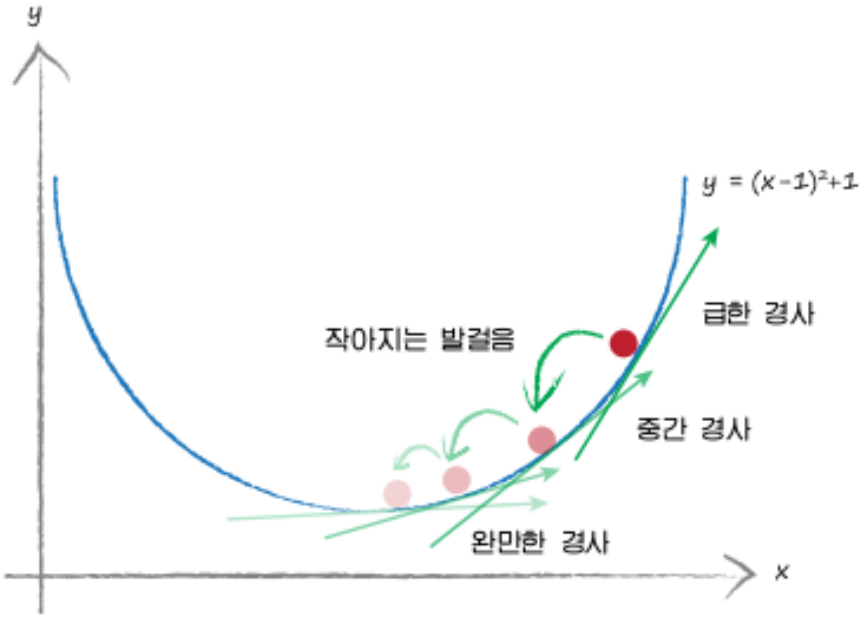
가중치 업데이트

- 어두운 밤에 험난한 산속에서 내려와야 할 경우

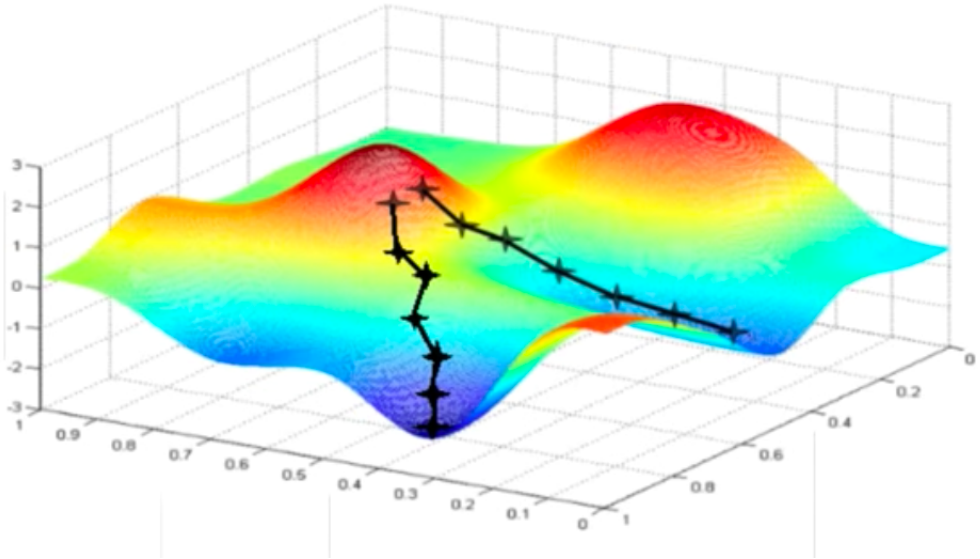
➔ ***경사 하강법 (gradient descent)***



경사하강법



매개변수가 많을 때

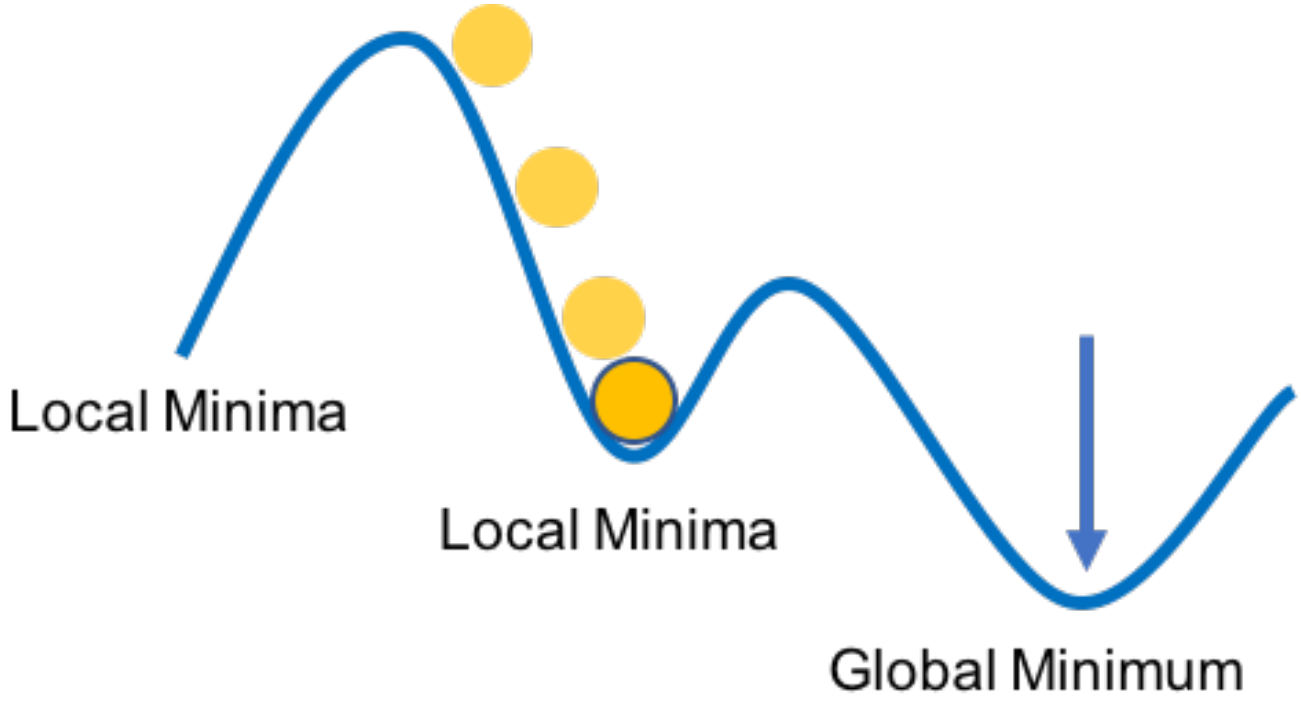


양의 기울기라면 x 값을 감소

음의 기울기라면 x 값을 증가

경사하강법

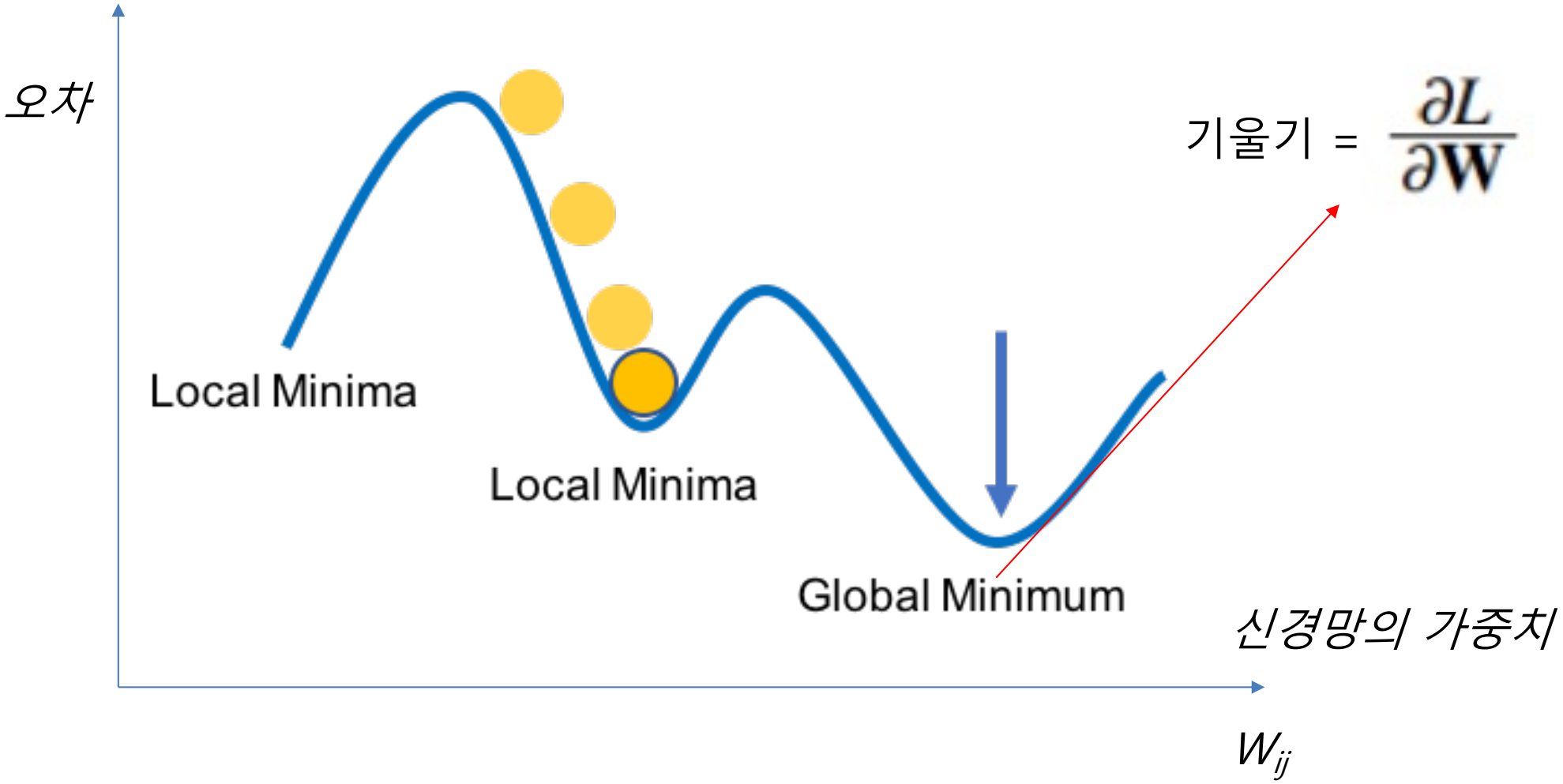
- 경사하강법 문제



미분

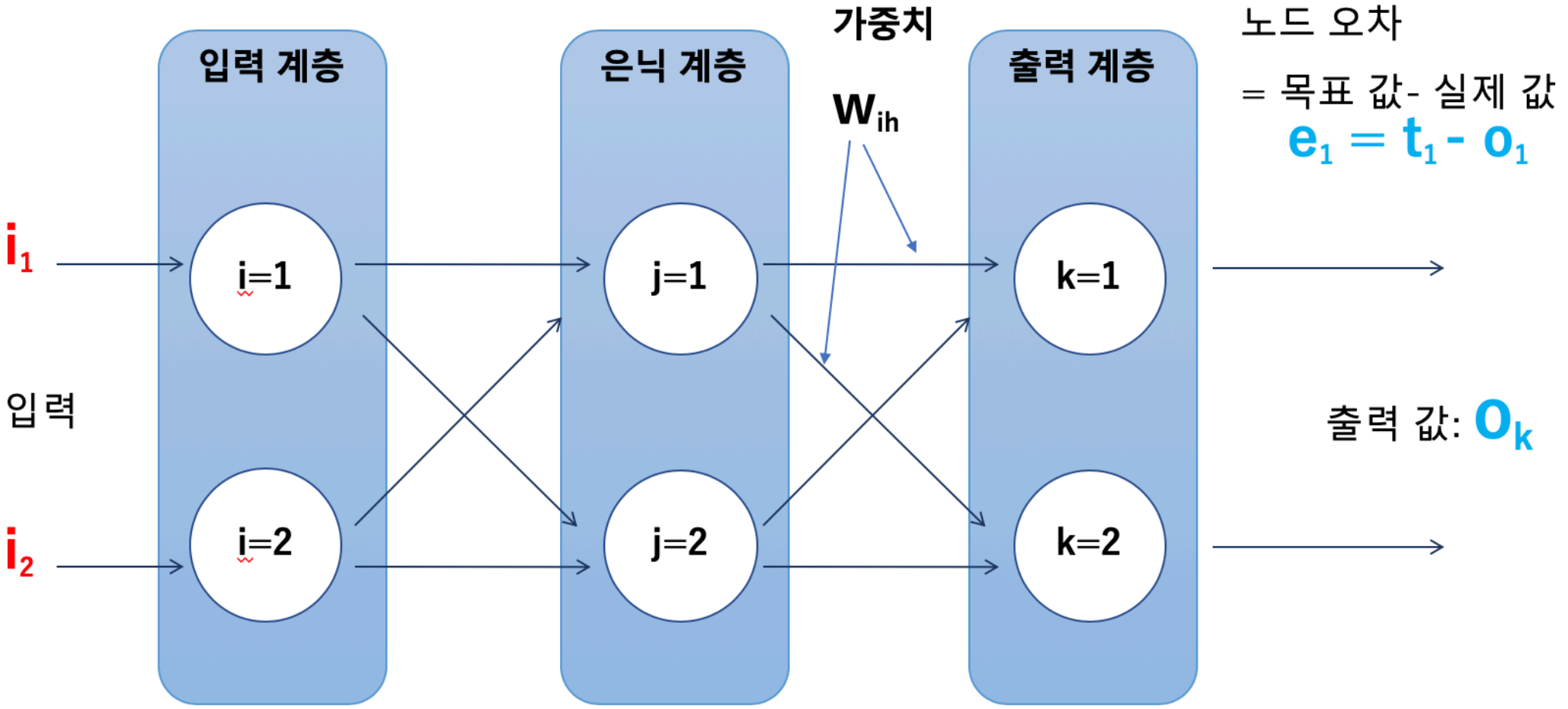
- 경사하강법 ➔ 가중치에 대한 오차함수의 기울기
- 오차는 가중치의 변화에 얼마나 민감한가

미분



미분

- 가중치 w_{ij}의 값이 변화함에 따라 오차 E의 값이 얼마만큼 변하는지?



오차함수

- 오차함수 = n개의 노드에 대해 목표 값과 실제 값의 차를 구해 이를 제곱한 다음 모두 더함

$$\frac{\delta E}{\delta w_{jk}} = \frac{\delta}{\delta w_{jk}} \sum_{k=0}^{n} (t_n - on)^2$$

- 미분
  - w_{jk} 이외의 값은 삭제 가능 (Σ) 통째로 삭제 가능
    - 노드의 결과 값은 오직 연결된 가중치에 의해서만 영향 받음

$$\frac{\delta E}{\delta w_{jk}} = \frac{\delta}{\delta w_{jk}} (t_k - ok)^2$$

- t_n는 상수이므로 w_n의 값이 변해도 바뀌지 않음

오차함수

- 연쇄법칙으로 미분 분리

$$\frac{\delta E}{\delta w_{jk}} = \frac{\delta E}{\delta o_k} \cdot \frac{\delta o_k}{\delta w_{jk}}$$

$$E = (t_k - ok)^2$$

$$\frac{\delta E}{\delta w_{jk}} = -2(t_k - ok) \cdot \frac{\delta o_k}{\delta w_{jk}}$$

$$\frac{\delta E}{\delta w_{jk}} = -2(t_k - ok) \cdot \frac{\delta}{\delta w_{jk}} sigmoid(\Sigma_j w_{jk} \cdot o_j)$$

- o_k는 노드 k의 결과 값
  - 입력 신호의 가중치 합에 sigmoid 함수 적용 한 것

오차함수

$$\frac{\delta}{\delta x} sigmoid(x) = sigmoid(x)(1 - sigmoid(x))$$

$$\frac{\delta E}{\delta w_{jk}} = -2(t_k - ok) \cdot sigmoid(\Sigma_j\ wjk \cdot oj)(1 - sigmoid(\Sigma_j\ wjk \cdot oj)) \frac{\delta}{\delta w_{jk}} (\Sigma_j\ w_{jk} \cdot o_j)$$

$$= -2(t_k - ok) \cdot sigmoid(\Sigma_j\ wjk \cdot oj)(1 - sigmoid(\Sigma_j\ wjk \cdot oj)) o_j$$

$$\frac{\delta E}{\delta w_{jk}} = -(t_k - ok) \cdot sigmoid(\Sigma_j\ wjk \cdot oj)(1 - sigmoid(\Sigma_j\ wjk \cdot oj)) \cdot o_j$$

오차함수

- *은닉 계층과 출력 계층의 사이에 있는* ***<u>가중치 업데이트</u>***

$$\frac{\delta E}{\delta w_{jk}} = -(t_k - ok) \cdot sigmoid(\Sigma_j\ wjk \cdot\ oj)(1 - sigmoid(\Sigma_j\ wjk \cdot\ oj)) \cdot o_j$$

(목표 값 – 실제 값)

최종 계층의 노드로 들어오는 입력 신호 i_k

은닉 계층의 노드 j의 결과 값

오차함수

- *입력 계층과 은닉 계층의 사이에 있는* ***가중치 업데이트***

$$\frac{\delta E}{\delta w_{jk}} = -(e_j) \cdot sigmoid(\Sigma_i\ wij \cdot oi)(1 - sigmoid(\Sigma_i\ wij \cdot oi)) \cdot o_i$$

은닉 계층에서 재조합 된 역전파 오류

은닉 계층의 노드 j로 들어오는 입력 값에 가중치를 적용한 결과

첫번째 계층의 노드 o_i의 결과 값

학습률

- 학습률은 문제에 따라 다르게 튜닝
  - 가중치가 최저점 근처에서 오버슈팅 방지

$$new\ \boldsymbol{w_{jk}} = old\ w_{jk} - \alpha \cdot \frac{\delta E}{\delta w_{jk}}$$

오버슈팅을 방지하기 위해 변화의 강도를 조정 → ***학습률***

학습률

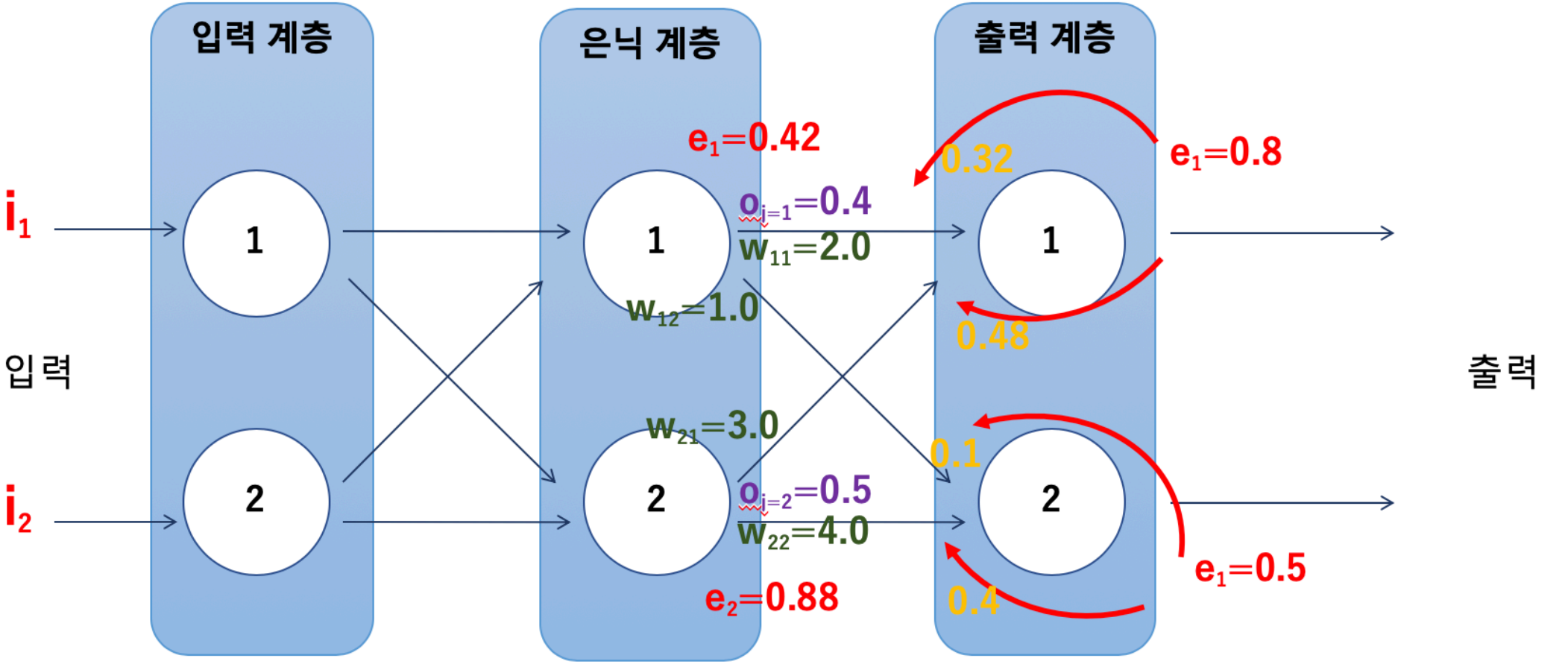
- 가중치 변화의 각 원소들의 구성 행렬

$$\begin{pmatrix} \triangle w_{1,1} & \triangle w_{2,1} & \triangle w_{3,1} & \cdots \\ \triangle w_{1,2} & \triangle w_{2,2} & \triangle w_{3,2} & \cdots \\ \triangle w_{1,3} & \triangle w_{2,3} & \triangle w_{j,k} & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{pmatrix} = \begin{pmatrix} E_1 * S_1 (1 - S_1) \\ E_2 * S_2 (1 - S_2) \\ E_k * S_k (1 - S_k) \\ \cdots \end{pmatrix} \cdot \begin{pmatrix} o_1 & o_2 & o_j & \cdots \end{pmatrix}$$

다음 계층으로부터의 값들 전 계층으로부터의 값들

$$\triangle w_{jk} = \alpha \cdot E_k \cdot o_k (1 - o_k) \cdot o_j^T$$

가중치 업데이트


입력 계층
은닉 계층
출력 계층
i_1
i_2
입력
1
2
1
2
1
2
출력
e_1=0.42
$o_{j=1}$=0.4
w_{11}=2.0
w_{12}=1.0
w_{21}=3.0
$o_{j=2}$=0.5
w_{22}=4.0
e_2=0.88
0.32
0.48
0.1
0.4
e_1=0.8
e_1=0.5

가중치 업데이트

$$\frac{\delta E}{\delta w_{jk}} = -(t_k - o_k) \cdot sigmoid(\Sigma_j w_{jk} \cdot o_j)(1 - sigmoid(\Sigma_j w_{jk} \cdot o_j)) \cdot o_j$$

$= -(0.8 * 0.083 * 0.4)$

오차 e_1, e_1=0.8

$\Sigma_j w_{jk} \cdot o_j = (2.0 * 0.4) + (3.0 * 0.5) = 2.3$

$$\text{sigmoid} \Rightarrow \frac{1}{1 + e^{-2.3}} = 0.909$$

$$0.909 * (1 - 0.909) = 0.083$$

j=1일때, 가중치 w_{11}, $o_{j=1}$=0.4

학습률이 0.1 일때 변화량 ➔ (0.1 * -0.0265) = -0.00265

⇒ ***새로운 w_{11} 은 2.0 – (-0.00265) = 2.00265***

가중치 업데이트

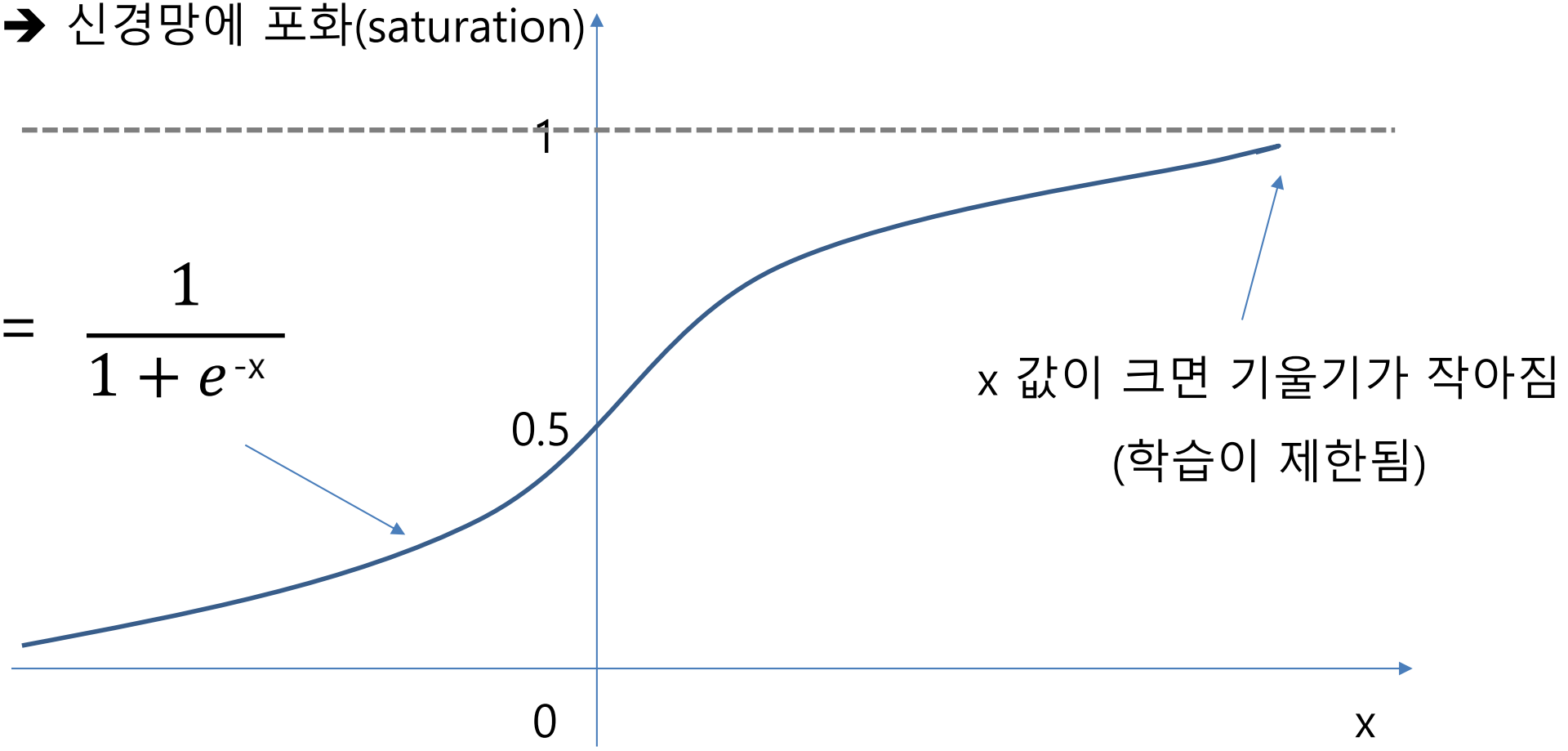
- 입력 값
  - 가중치의 변화는 활성화 함수의 기울기에 좌우
    - 작은 기울기 = 학습 능력 제한

➔ 신경망에 포화(saturation)

0.0 ~ 0.1

입력 값에 0.01과 같은
작은 오프셋 추가 (0 방지)

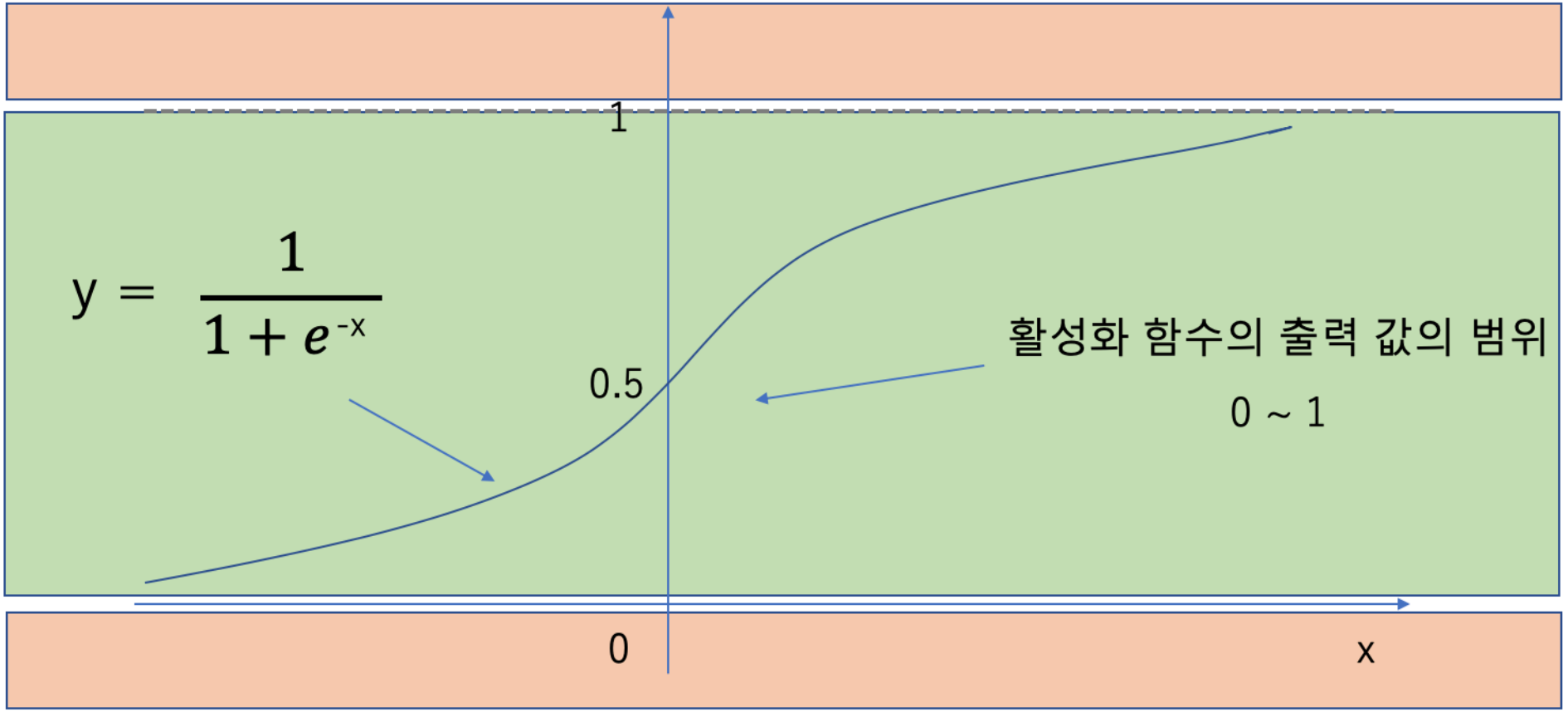
$$y = \frac{1}{1 + e^{-x}}$$



x 값이 크면 기울기가 작아짐
(학습이 제한됨)

가중치 업데이트

- 결과 값
  - 1에 가까운 값을 가짐 ➔ 로지스틱 함수는 1.0보다 크거나 0보다 작은 출력 값 불가능



가중치 업데이트

- 임의의 값으로 ***가중치*** 초기화
  - -1.0 ~ 1.0 사이의 임의 값, 많은 연결 노드를 가질 수록 가중치의 범위를 줄여야 함
  - 노드로 오는 연결 노드의 개수에 루트를 씌운 다음 역수를 취해 얻은 값 범위
  - ex) 노드가 3개의 연결 노드 가질 경우

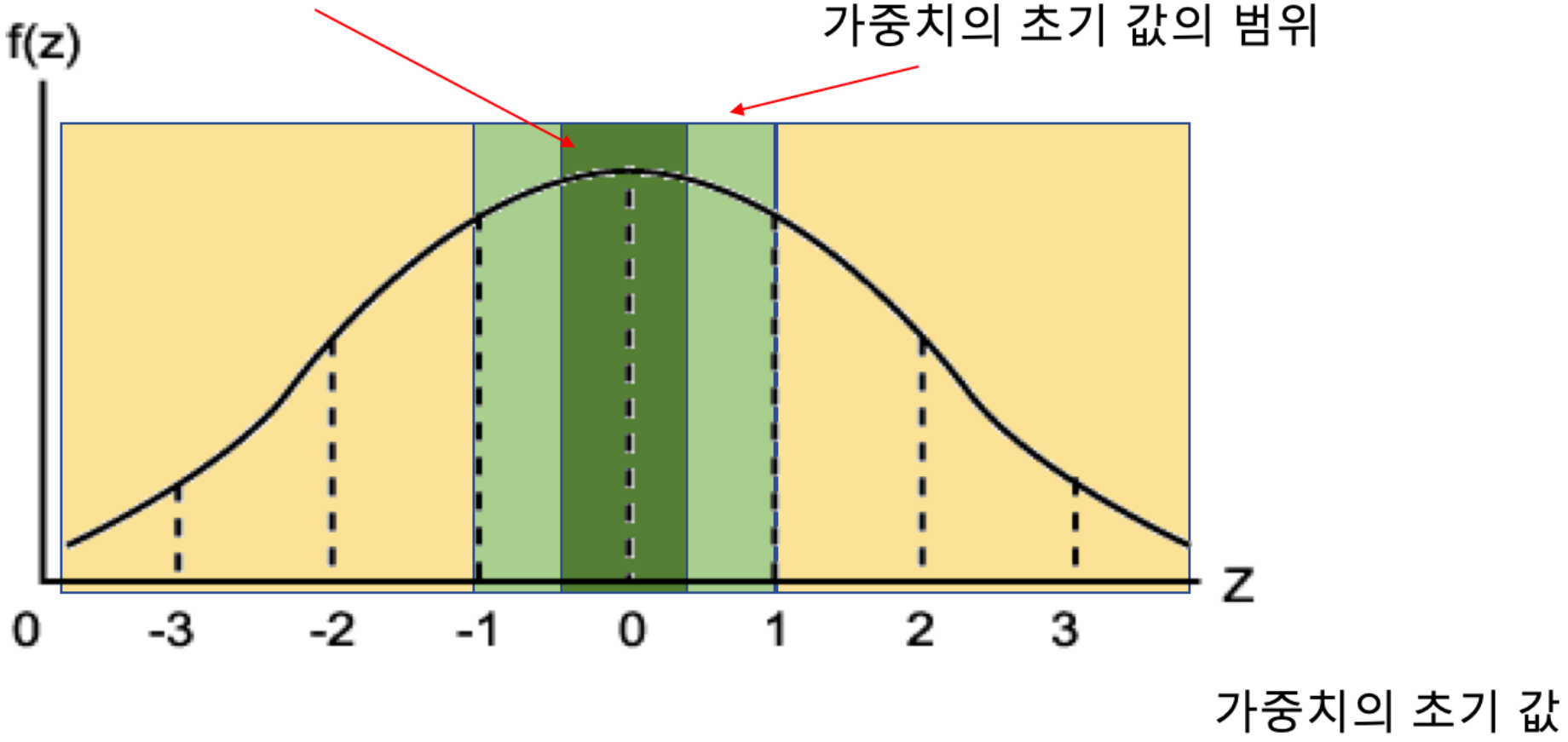
  - 가중치의 초기값을 모두 같은 값으로 설정할 경우 **(X)**

    → 모든 노드들은 같은 신호를 받으며, 출력 값도 동일

가중치 업데이트

- 임의의 값으로 ***가중치*** 초기화

0에 가까운 가중치를 선택하는 편향



"Automatic" Feature Engineering

- W와 b의 초기값은 랜덤 → Gradient descent로 최적화
- Feature Engineering은 Domain Knowledge가 필요
  - 종양 이미지를 보고 종양 분류하는 작업에는 컴퓨터 전문가와 종양 이미지를 보고 양성 음성을 판단할 수 있는 전문가가 필요
  - 주가 예측을 위해서는 금융 지식이 필요
  - 이상 탐지, 사기 탐지를 위해서는 보험 관련 분야의 지식이 필요
- Deep learning은 도메인 전문가가 아니어도 모델을 생성할 수 있음
  - 비 방사선 전문의도 의료 진단을 위한 최첨단 이미지 분류 모델을 구축할 수 있음

Tensorflow PLAYGROUND

http://playground.tensorflow.org/