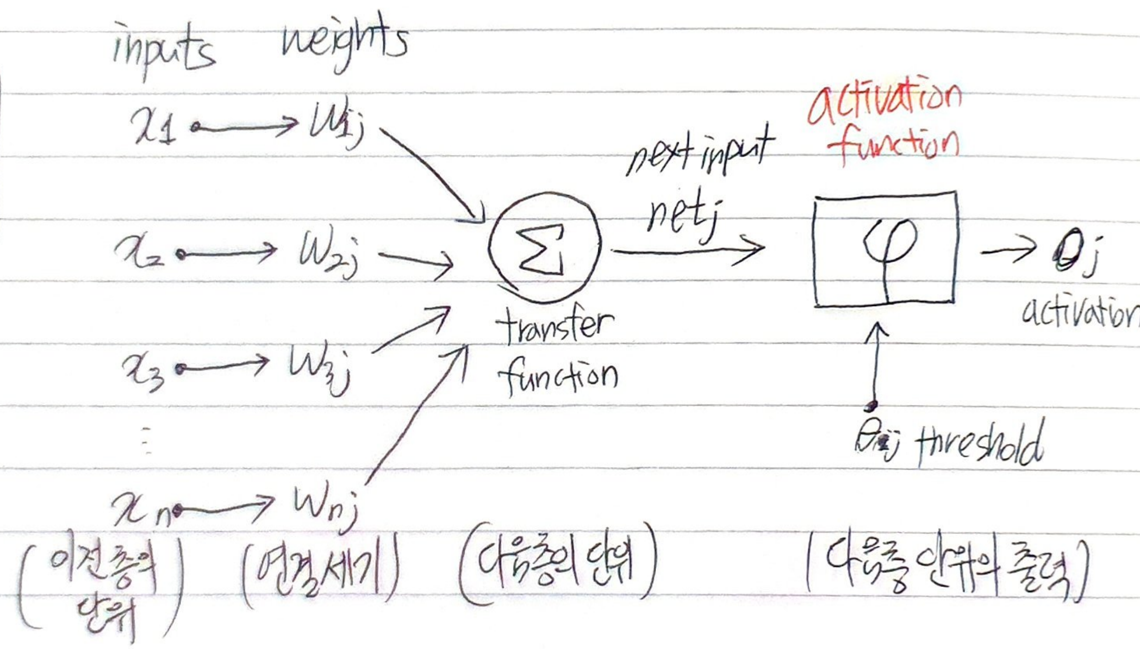
Week, 13 인공신경망

13.1 신경망

사람의 뇌에 있는 뉴런(신경세포, neuron)은 혈액 중의 아미노산으로부터 신경전달물질을 만든다. 신경의 자극은 한 방향으로만 일어나며, 만들어진 신경전달물질을 이용하여 주변의 다른 신경세포들에 신호를 전달한다.

하나의 신경세포는 여러 개의 다른 신경세포들로부터 전달받은 신호(입력신호)에 반응하여 세포막이 역치(threshold value, 임계값) 전위에 도달하면 다음 신경세포들에도 일정한 크기의 신호(출력신호)를 전달하게 된다.

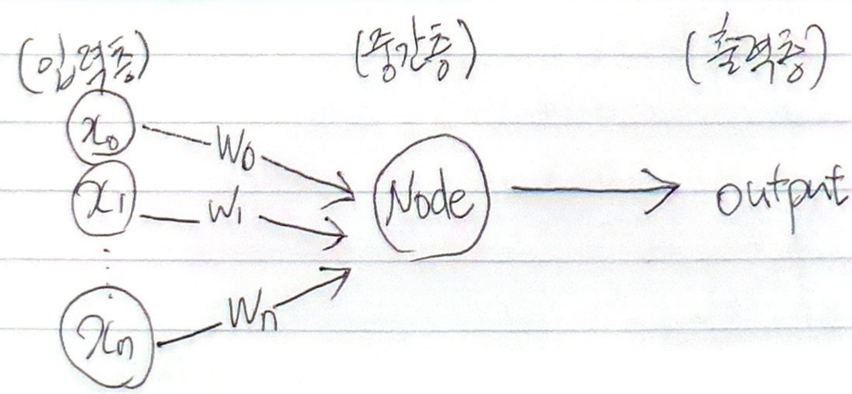
인공 신경망



- 신경망(neural network)은 신경계의 기본 단위인 뉴런을 모델화한 것
- 하나의 인공 뉴런(노드, node)에서는 다수의 입력신호 x_i를 받아서 하나의 신호를 출력

// 뉴런의 돌기가 신호를 전달하듯, 인공뉴런에서는 가중치(weight)가 그 역할을 수행

// 각 입력신호에는 고유한 가중치가 부여되며, 가중치가 클수록 해당 신호가 중요



13.2 신경망의 작동원리

입력신호 x를 받아 y를 출력 → $y = wx + \underbrace{b}_{\text{bias(편향)}}$

ex) 입력신호 x 수신 → 사전부여된 가중치 w 계산 → 임계값 θ와 비교 (더 큰가?) → yes → $\boxed{1}$

$wx - \theta > 0 \leftarrow \boxed{wx > \theta \Downarrow \text{print}(1)}$

$y = wx + b > 0$

$\therefore \boxed{b = -\theta}$

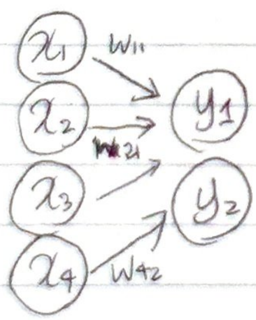
입력신호 2개 → $y = ax_1 + bx_2 + c$

입력신호 3개 → $y = ax_1 + bx_2 + cx_3 + d$

⋮

출력이 두 개인 경우를 행렬의 곱으로 나타내기

* 편의상 b(편향) = 0 이라 한다.
* w_{ij}는 입력층 i번째 노드에서 출력층의 j번째 노드로 연결된 가중치를 의미



$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ w_{21} & w_{22} & w_{23} & w_{24} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

◉ 활성화 함수 (activation function)

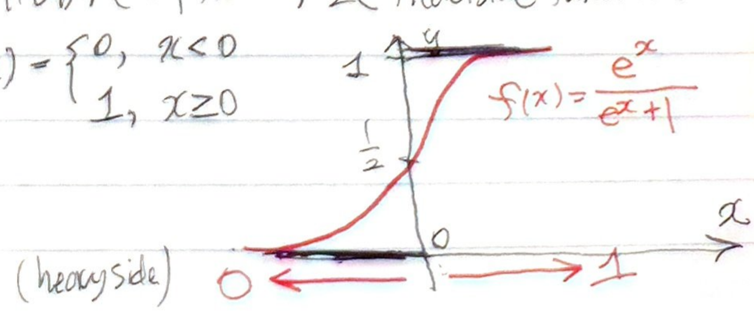
다수의 입력신호가 주어지면, 미리 부여된 가중치와 계산을 한 후 그 총합이 정해진 임계값을 넘으면 1을 출력하고, 넘지 못하면 0 or -1을 출력.

ex1. Sigmoid func.

$$f(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1}$$

[임계값을 기준으로 활성화되거나 혹은 비활성화되는 계단함수 (step function) 또는 Heaviside function]

$$H(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases}$$



Sigmoid 함수 $f(x)$에 대해 다음이 성립한다.

① 모든 x에 대하여 $0 < f(x) < 1$이다.

② $x > 0$ 일 때, $f(x) > 0.5$, $x < 0$일 때 $f(x) < 0.5$이다.

③ Sigmoid 함수의 미분공식는 다음과 같다.

$$f'(x) = (1+e^{-x})^{-2}(-e^{-x})$$

$$= \frac{e^{-x}}{(1+e^{-x})^2}$$

"$f(x)$를 그대로 활용 가능한 형태"

$$= \frac{1}{1+e^{-x}} \cdot \frac{e^{-x}}{1+e^{-x}} = \boxed{f(x)(1-f(x))}$$

Sigmoid 함수는 계단함수와 비교하여, 출력신호를 극단값 0 or 1이 아니라

· 연속적인 0과 1 사이 값으로 정규화하여 전달해준다.

· 모든 실수 x에서 미분이 가능하여 계산이 간편하다.

*그 외 활성화 함수: ReLU, $\tanh x$ 등

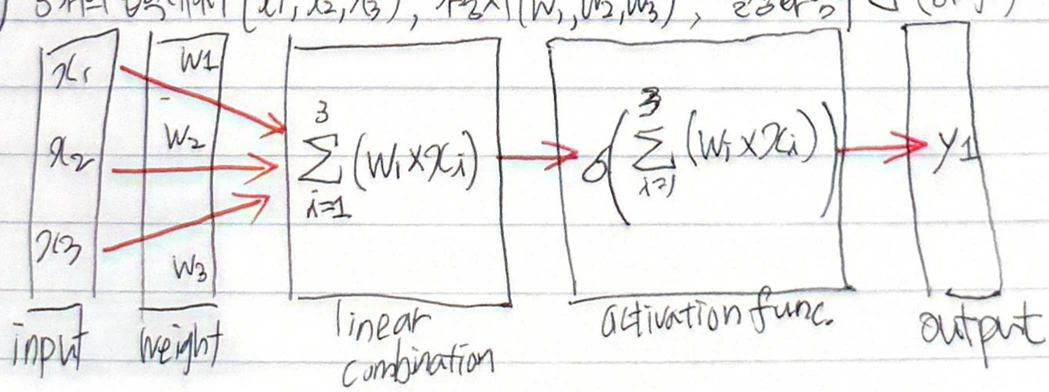
■ 인공뉴런 Process 정리

① 인공뉴런은 입력데이터 $(x_1, x_2, \cdots, x_n)$을 다른 인공뉴런/외부로부터 입력받는다.

② 인공뉴런은 입력받은 데이터와 가중치 $(w_1, w_2, \cdots, w_n)$을 결합하여 $\sum_{i=1}^{n} w_i x_i + b$로 표현되는 하나의 값으로 만든다.

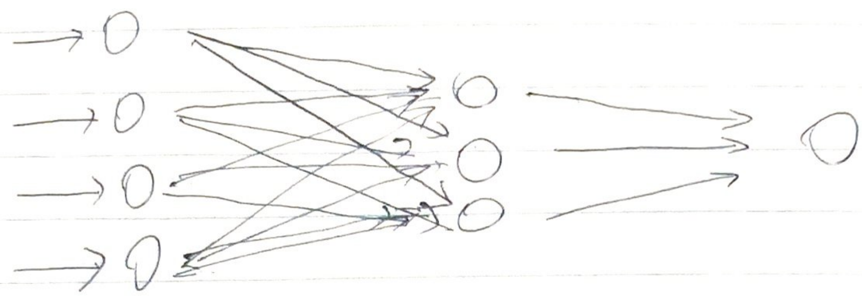
③ 위에서 구한 $(\sum_{i=1}^{n} w_i x_i)+b$ 값을 활성화 함수 $f(x)$에 대입하여 $f(\sum_{i=1}^{n} w_i x_i + b)$를 출력값으로 내보낸다.

도식화 3개의 입력데이터 (x_1, x_2, x_3), 가중치 (w_1, w_2, w_3), 활성화함수 σ (or f) → output y_1



13.3 신경망의 학습

신경망 = [입력층 input layer] + [은닉층 hidden layer] + [출력층 output layer]



[목적] 입력값과 정답을 알고있는 데이터가 주어졌을 때, 데이터를 잘 판단하도록 신경망 가중치 찾기

① 초기에 임의의 가중치 부여
② 주어진 데이터를 사용하여, 신경망 예측값과 정답간 오차를 줄이는 방향으로 가중치 수정
③ 계층 간 각각의 연결이 오차에 영향주는 정도에 비례해서 오차를 갱신, 전달

13.4 오차 역전파법 Error Back Propagation

↳ 경사하강법(gradient descent method)을 오차함수 최소화 문제에 적용한 것

ex) 각 계층에서 입력신호 x_i $(i=1,2,\cdots,N)$를 받아, 미리 부여된 가중치와 계산 후 주어진 활성함수로부터 출력된 값을 $\hat{y}_i$라 하고, 실제정답은 y_i라 하자.
그러면 제곱오차(squared error)는 다음과 같이 주어진다.

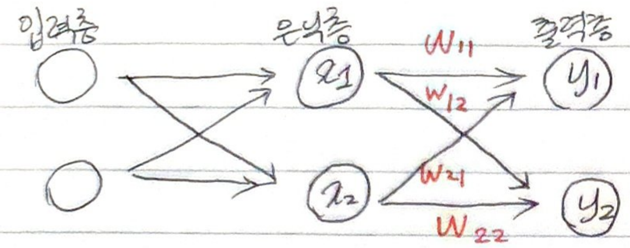
$$E = \frac{1}{2}\sum_{i=1}^{N} \| y_i - \hat{y}_i \|^2$$

그러면 각 가중치를 갱신하는 공식은 경사하강법으로부터 다음과 같이 주어진다.

$$W_{ij}^{(NEW)} = W_{ij}^{(OLD)} - \eta \frac{\partial E}{\underbrace{\partial W_{ij}}_{\text{편미분}}}$$

이제 아래와 같이 간단한 신경망의 각 계층에 전달된 오차로부터 가중치를 갱신하는 방법에 관하여 살펴보자.

① 각 계층에 전달된 오차를 계산한다. 먼저 출력층에서의 오차를 계산하기 위하여, 은닉층에서 입력신호 x_1과 x_2를 받아 가중치 $w_{11}, w_{12}, w_{21}, w_{22}$와 계산한 후 sigmoid 함수 $f(x)$를 거쳐 출력 y_1과 y_2를 얻었다고 가정하자.



정답	오차
t_1	$e_1 = t_1 - y_1$
t_2	$e_2 = t_2 - y_2$

출력층// 제곱오차 $E = \frac{1}{2}(e_1^2 + e_2^2) = \frac{1}{2}\left((t_1 - y_1)^2 + (t_2 - y_2)^2\right)$

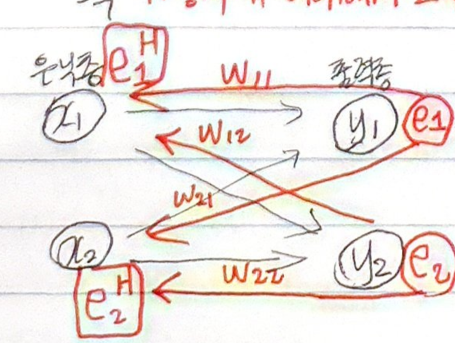
$$= \frac{1}{2}\left((t_1 - f(x_1 w_{11} + x_2 w_{21}))^2 + (t_2 - f(x_1 w_{12} + x_2 w_{22}))^2\right)$$

은닉층 오차계산

→ 오차 역전파법 활용

"신경망에서 오차가 발생했다" = 입력신호가 입력층→은닉층→출력층으로 전파될 때, 은닉층에서의 오차가 반영된 결과

되돌려주려면? ● 계층간 각각의 연결이 오차에 영향을 주는 정도, 즉 가중치에 비례해서 오차를 역으로 전달해주면 된다.



$$e_1^H = \frac{w_{11}}{w_{11} + w_{21}} e_1 + \frac{w_{12}}{w_{12} + w_{22}} e_2, \quad e_2^H = \frac{w_{21}}{w_{11} + w_{21}} e_1 + \frac{w_{22}}{w_{12} + w_{22}} e_2$$

분모 제외 표현하면, $\begin{bmatrix} e_1^H \\ e_2^H \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \end{bmatrix}$

② 출력층에서 얻은 오차로부터 은닉층과 출력층 사이의 가중치를 갱신한다.

Sigmoid 함수의 성질과 연쇄법칙으로부터 $\frac{\partial E}{\partial w_{11}}$ 를 구하면 다음과 같다.

$$\frac{\partial E}{\partial w_{11}} = (t_1 - f(w_{11}x_1 + w_{21}x_2))\, f(x_1 w_{11} + x_2 w_{21})\,(1 - f(w_{11}x_1 + w_{21}x_2))\, x_1$$

$$= -(t_1 - y_1)\, y_1 (1 - y_1)\, x_1$$

* 여기서는 $\frac{\partial E}{\partial w_{11}}$ 가 쉽게 계산됨을 확인하고 넘어가자.

(* Sigmoid 함수 $f(x)$
$f'(x) = f(x)(1 - f(x))$)

마찬가지 방법으로 다른 가중치에 대해서도 다음과 같이 계산한다.

$$\frac{\partial E}{\partial w_{12}} = -(t_2 - y_2)\, y_2 (1 - y_2)\, x_1$$

$$\frac{\partial E}{\partial w_{21}} = -(t_1 - y_1)\, y_1 (1 - y_1)\, x_2$$

$$\frac{\partial E}{\partial w_{22}} = -(t_2 - y_2)\, y_2 (1 - y_2)\, x_2$$

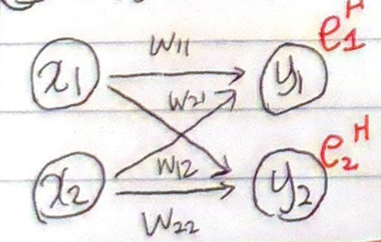
가중치 w_{ij}는 은닉층의 i번째 노드와 출력층의 j번째 노드만 연결되어있으므로, E를 w_{ij}에 대하여 편미분하면 다음과 같이 간단하게 표현됨을 알수 있다.

$$\frac{\partial E}{\partial w_{ij}} = -(t_j - y_j)\, y_j (1 - y_j)\, x_i$$

$$= -e_j\, y_j (1 - y_j)\, x_i$$

$$= \boxed{-(\text{출력층 } j \text{ 에러})(\text{출력층 } j \text{ 결과})(1 - \text{출력층 } j \text{ 결과}) \times \text{은닉층 } i \text{ 입력값}}$$

③ 은닉층에 전달된 오차로부터 입력층과 은닉층 사이 가중치를 갱신한다.



$$\frac{\partial E}{\partial w_{ij}} = -e_j^H\, y_j (1 - y_j)\, x_i$$

$$\boxed{\text{“경사하강법 적용”} \quad w_{ij}^{(NEW)} = w_{ij}^{(OLD)} - \eta \frac{\partial E}{\partial w_{ij}}}$$

입력층과 출력층 사이 은닉층이 여러개 → deep neural network (심층 신경망)

DNN을 학습하기위한 기계학습 기법 → deep learning