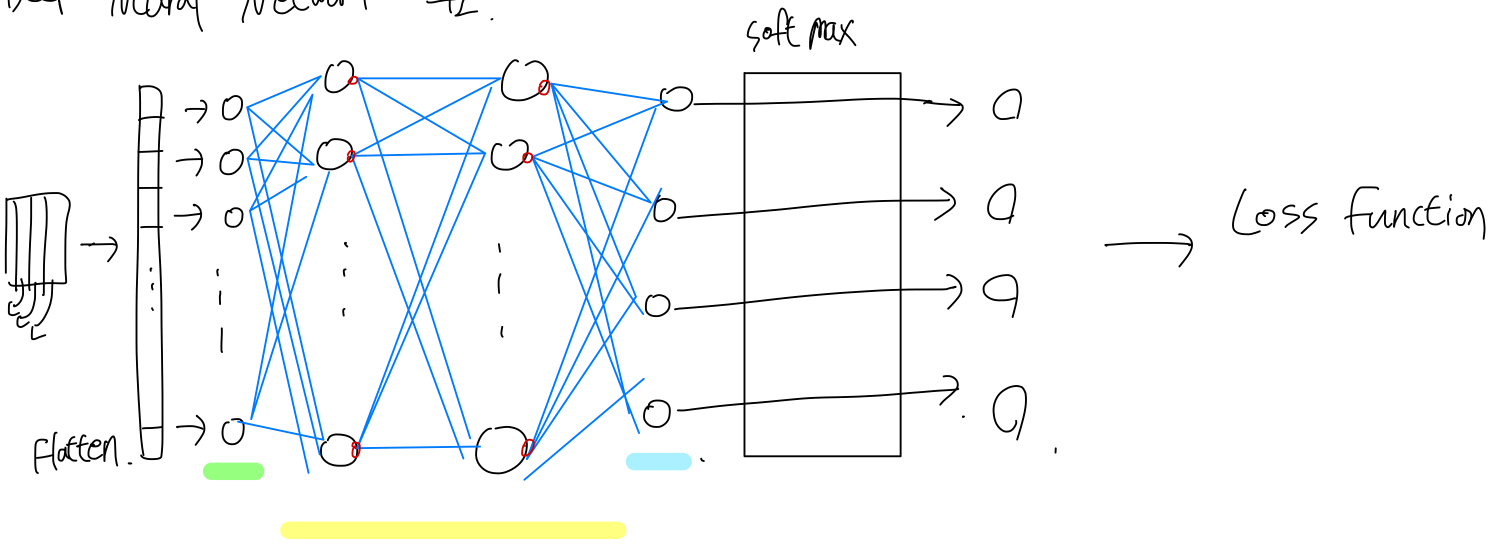
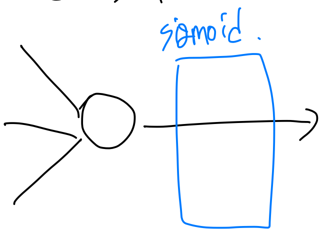
Deep Neural Network 구조.



- activation funtion.
- Loss function. (Cross-Entropy).

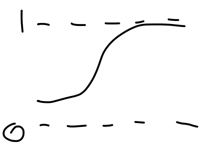
Loss Function.



$f(x, w)$.

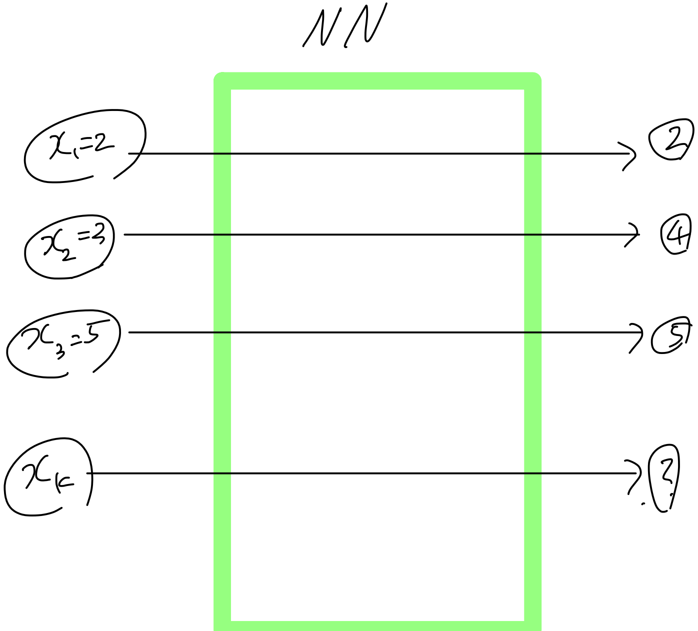
$$\left(= \frac{e^z}{1+e^z} \quad (z = b_0 + b_1 x_1 + \cdots + b_n x_n) \right).$$

sigmoid 확률적인 해석 가능.



but $\sum_{i=1}^{n} (y - f(x_i, w))^2$. 계산량 너무 많음.

⇒ 로지스틱에 맞는 Loss function "cross-entropy" 정의.



$$P(x_1) = sig(x_1, w)$$

$$P(x_2) = 1 - sig(x_2, w)$$

$$P(x_3) = sig(x_3, w)$$

$$P(x_k) = \begin{cases} sig(x_k, w), & y=1 \\ 1 - sig(x_k, w), & y=0 \end{cases}$$

$x_1, x_2, \cdots, x_n$: 독립시행 $\Rightarrow$ $\boxed{\text{We want to maximize } \prod_{k=1}^{n} P(x_k)}$

maximize $\prod_{k=1}^{n} P(x_k)$

$\Leftrightarrow$ minimize $-\prod_{k=1}^{n} P(x_k)$.

$\Leftrightarrow$ minimize $\log\left(-\prod_{k=1}^{n} P(x_k)\right)$.

log 함수 성질.

- monotone increse. ($x_1 < x_2 \Rightarrow \log x_1 < \log x_2$)
- $\log(1) = 0$. (확률이 1이면 오차 0).
- $\log(x_1 x_2) = \log x_1 + \log x_2$. (급격한 값의 변화를 완화시켜줌).

$(\times \rightarrow +)$.

$$\Rightarrow \quad Loss = -\log\left(\prod_{k=1}^{n} P(x_k)\right)$$

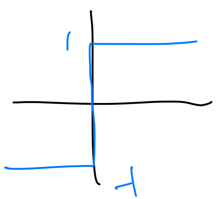
$$\Sigma\left(y - f(x, w)\right)^2$$

$$= -\sum_{k=1}^{n} \log P(x_k)$$

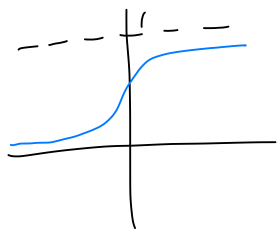
Activation Function.

- hidden layer 각 Node 마다 존재.

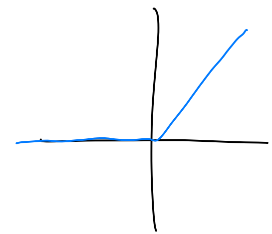
i) perceptron.



ii) sigmoid



iii) Relu.



. . .

activation function 없으면?

$a \times 3 + 2 = \boxed{b}$

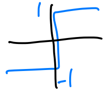
$\boxed{b} \times 5 + 4 = c$

$\longrightarrow c = (3a+2)5+4.$

b의 역할 X.

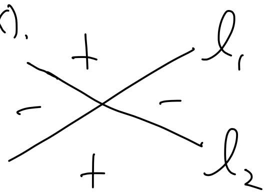
Non-linear 표현 불가.

Perceptron.



: 논리연산 (and, or, Xor, ...) 표현 가능.

ex).



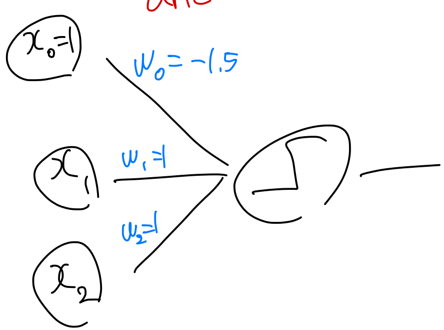
$(x_1 \ge l_1 \text{ and } x_1 \ge l_2)$

or

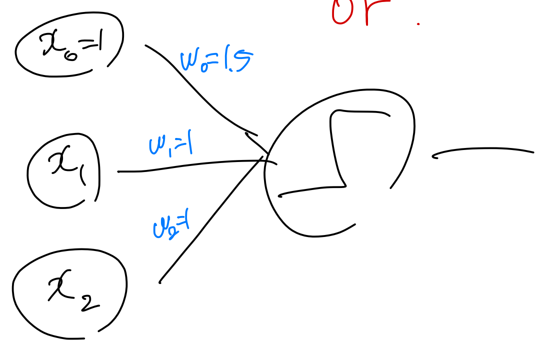
$(x_1 \le l_1 \text{ and } x_1 \le l_2)$

$\Rightarrow x_1 = +.$

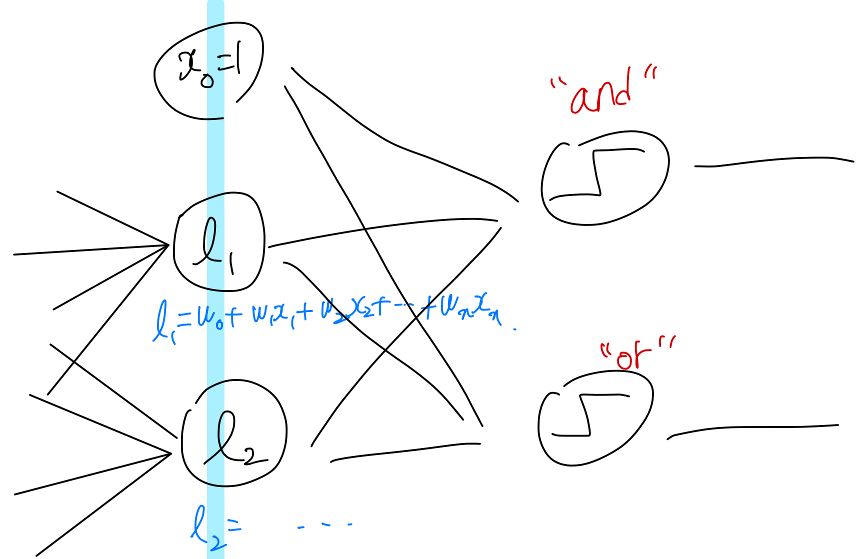
"and"



"or".



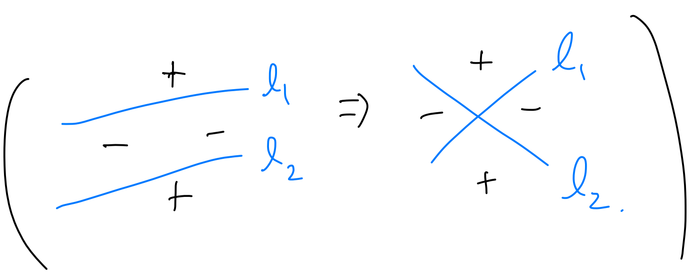
. . .

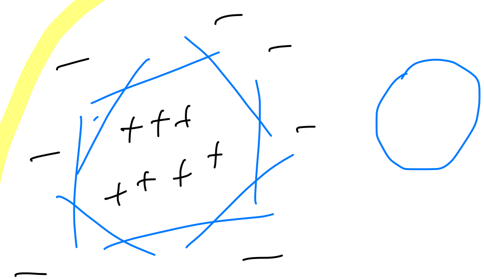


. . .

w 학습 ← → w 고정

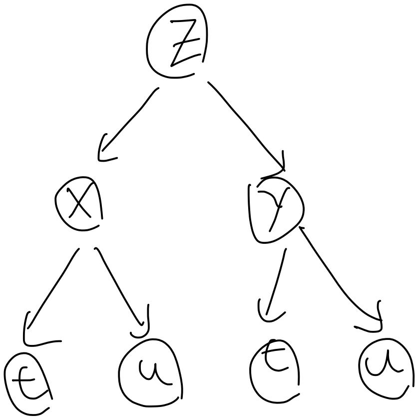
(l_n 변화)





non-linear로 표현가능.

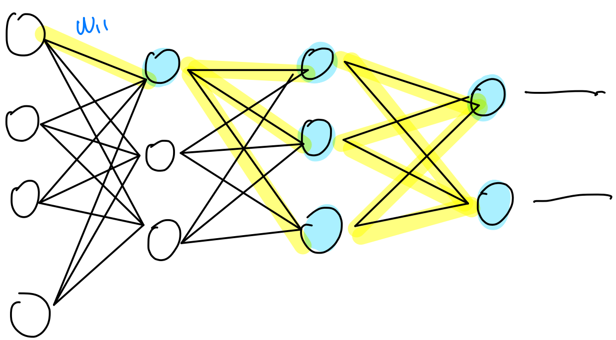
Chain Rule.



$$\frac{\partial z}{\partial t} = \frac{\partial z}{\partial x} \cdot \frac{\partial x}{\partial t} + \frac{\partial z}{\partial y} \cdot \frac{\partial y}{\partial t}.$$

$$\frac{\partial z}{\partial u} = \frac{\partial z}{\partial x} \cdot \frac{\partial x}{\partial t} + \frac{\partial z}{\partial y} \cdot \frac{\partial y}{\partial u}.$$

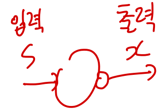
In Neural network...



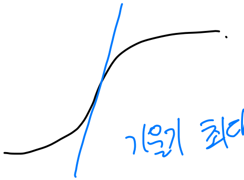
$$\frac{\partial f}{\partial w_{ij}^{(l)}} = \delta_j^{(l)} \cdot x_i^{(l)}$$

$$\delta_j^{(l)} = \left(\delta^{(l+1)}\right)^T w_j^{(l+1)} \cdot \boxed{\frac{\partial x_j^{(l+1)}}{\partial s_j^{(l)}}}$$

activation function의 기울기.



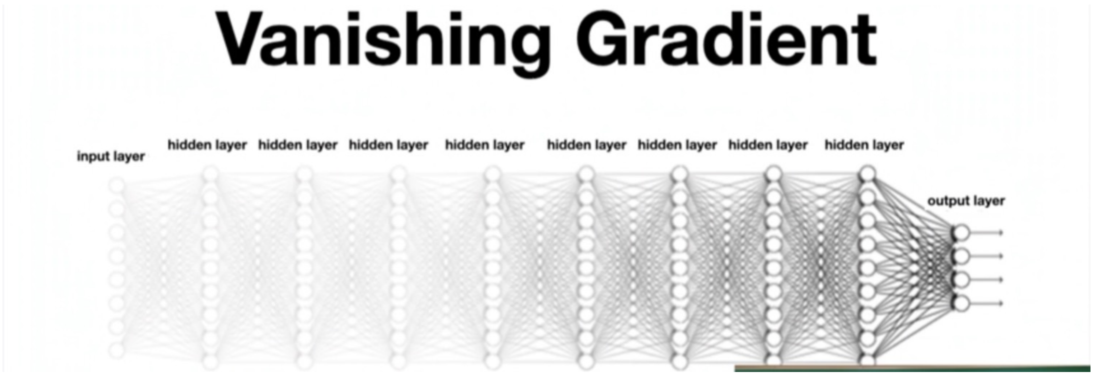
sigmoid.



기울기 최대값 : 0.25.

⇒ $\times \frac{1}{4} \times \frac{1}{4} \times \cdots$ 값이 쭉쭉 떨어짐.

⇒ "Gradient Vanishing".



sigmoid를 activation function 으로 사용했기 때문에 성능이 떨어졌음.

⇒

"Relu".