

AI and Deep Learning

# 로지스틱 회귀와 분류(2)

- 다중 클래스 -

제주대학교

변영철

<http://github.com/yungbyun/ml>

# 차례

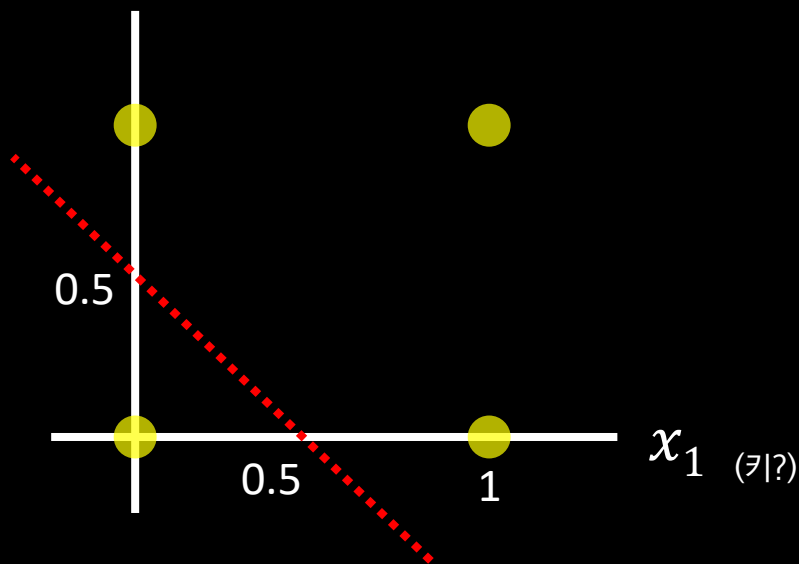
- 요즘 인공지능 이야기
- 생활 속 인공지능
- 인공지능과 4차산업혁명
- 인공지능 어떻게?
- 뉴런과 학습, 그리고 신경망
- 선형회귀와 논리회귀
- 딥러닝
- CNN과 RNN 이해하기

# 신경세포 하나

(몸부게?)

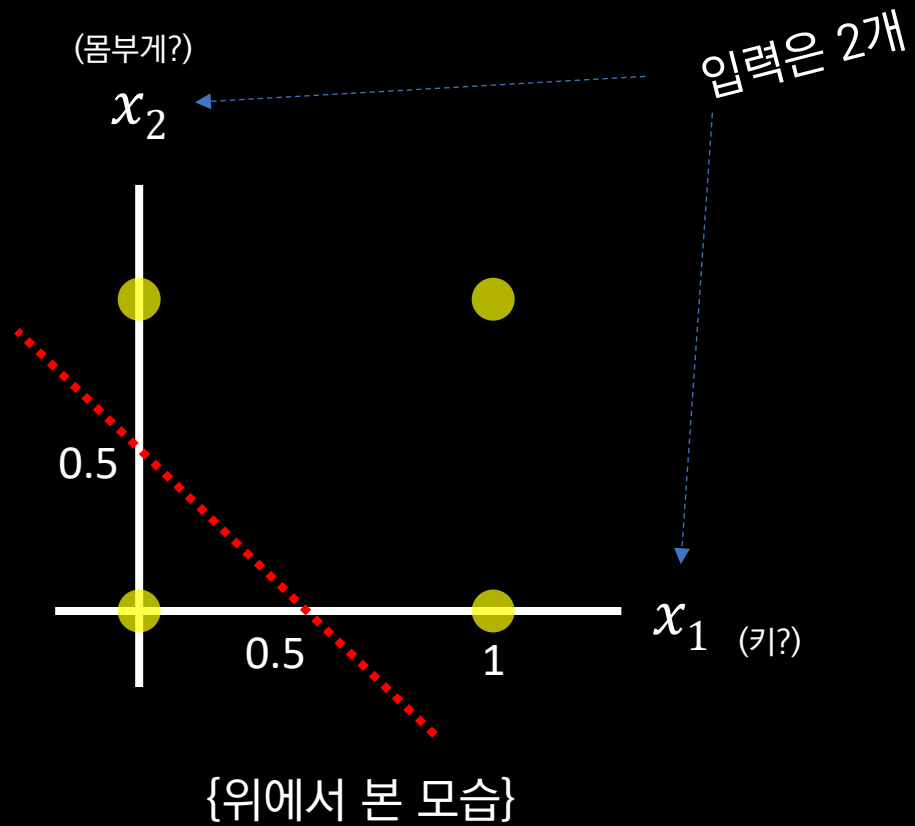
$x_2$

입력은 몇 개?



{위에서 본 모습}

# 신경세포 하나

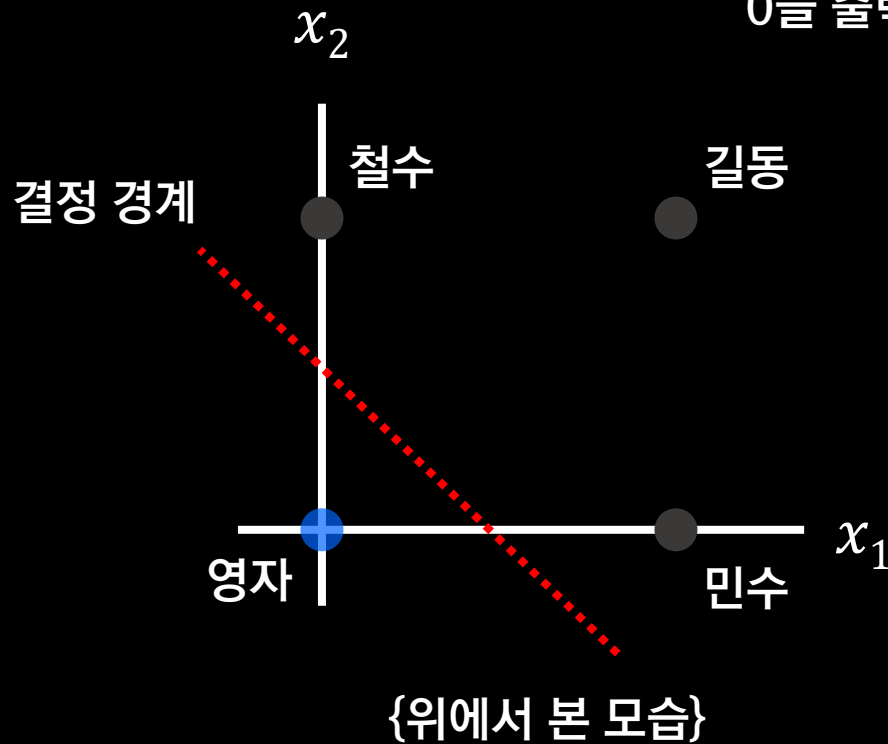


# 신경세포 하나

- 신경 세포 1개는 오직 하나의 결정 경계만을 만듦.
- 이것 아니면 저것 구분 (둘 중 하나)
- 그럼 넷 중 하나는?
  - 영자, 철수, 길동, 민수 (남자 여자 2개가 아니라 4개 중 하나로)를 인식하는 시스템을 만들려면?

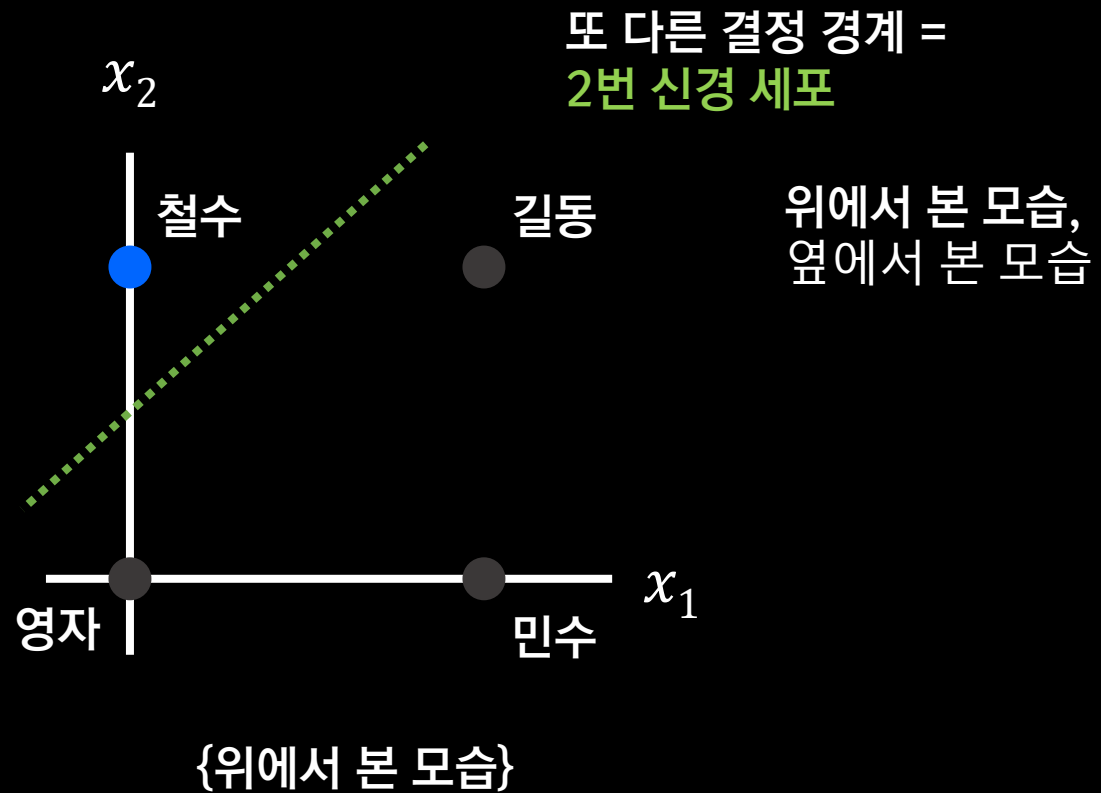
# 신경세포 #1

입력이 (0, 0)일 때만 1,  
영자일때만 불이켜지고,  
나머지에 대해서는  
0을 출력하는 1번 신경 세포

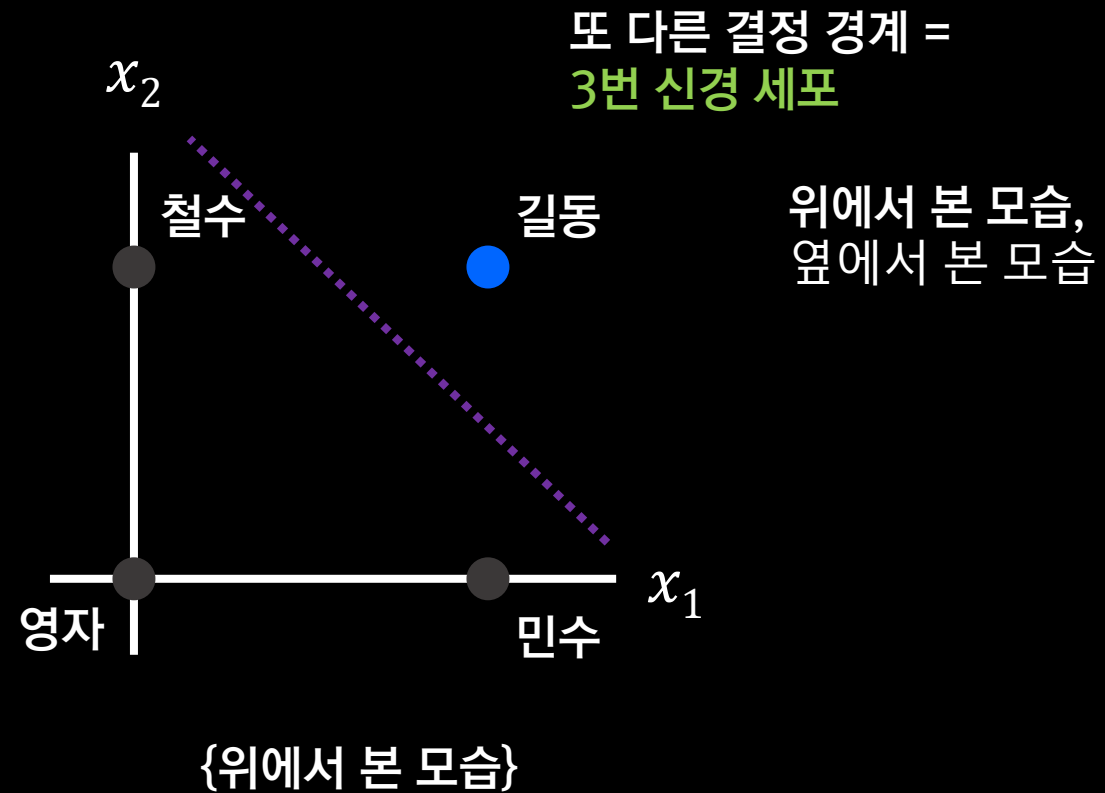


위에서 본 모습,  
옆에서 본 모습

# 신경세포 #2

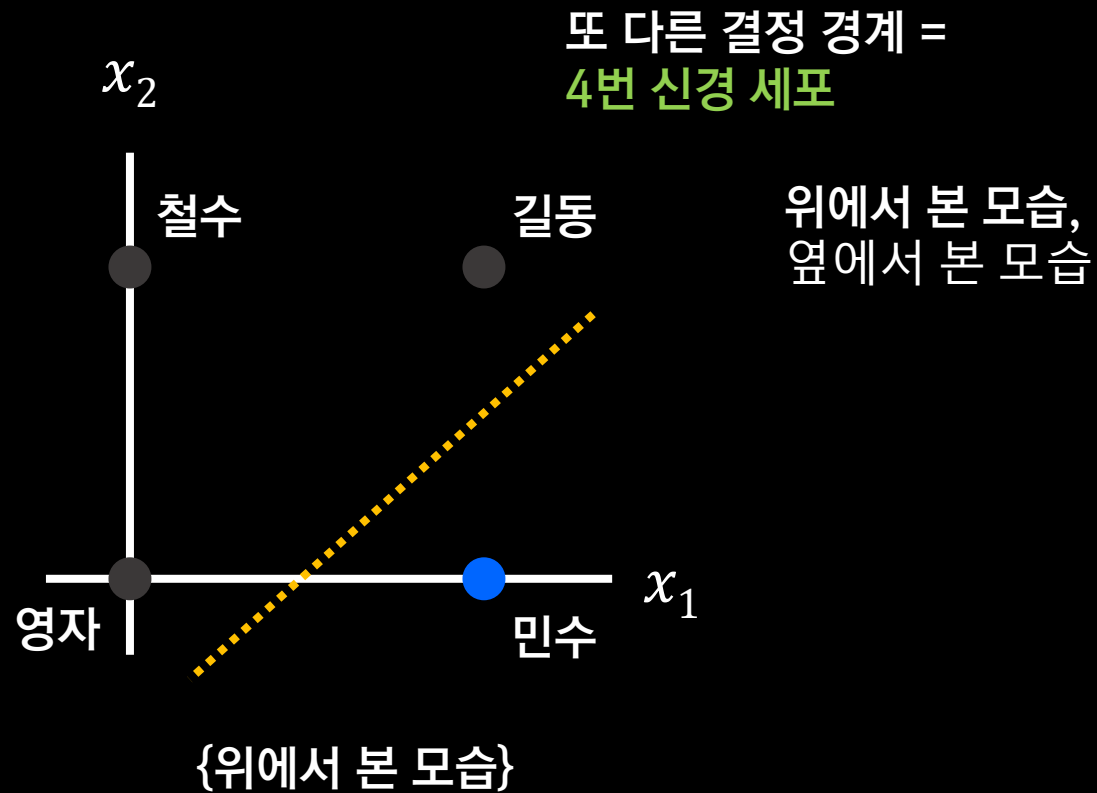


# 신경세포 #3



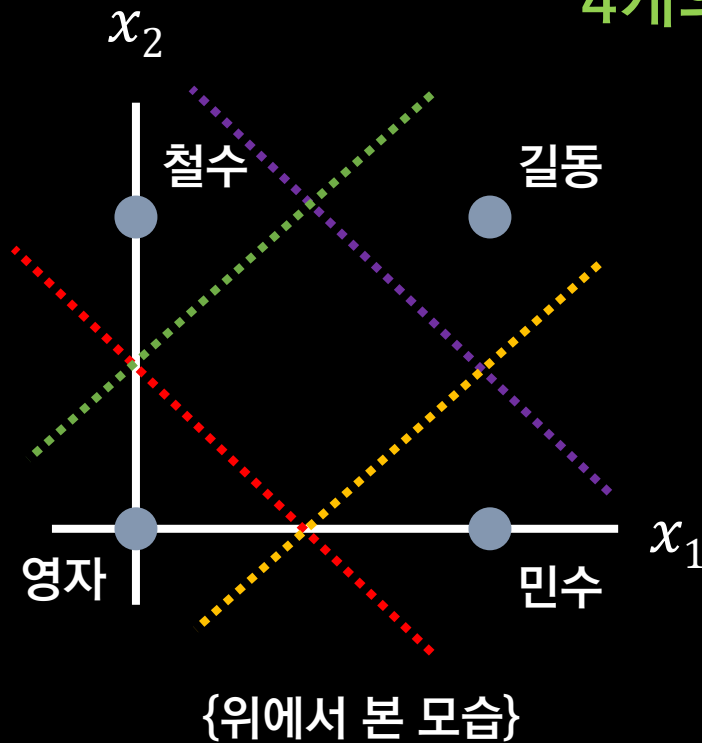


# 신경세포 #4



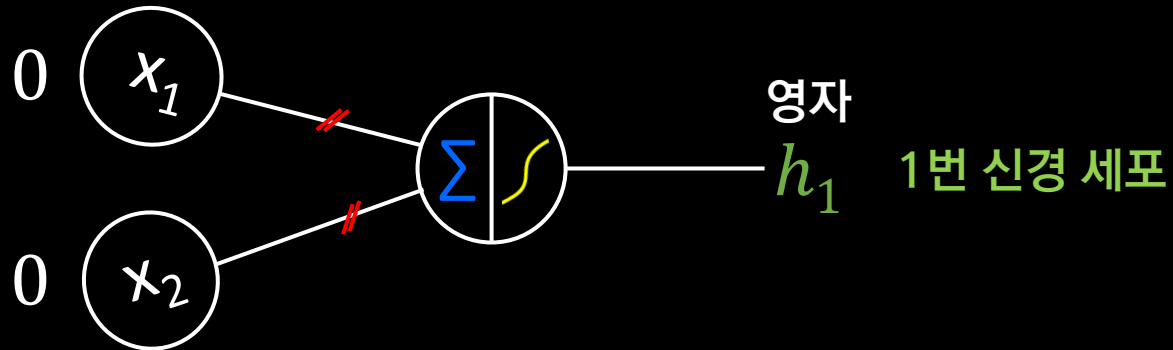
# 4개의 신경세포

각 결정경계를 만드는  
4개의 신경세포



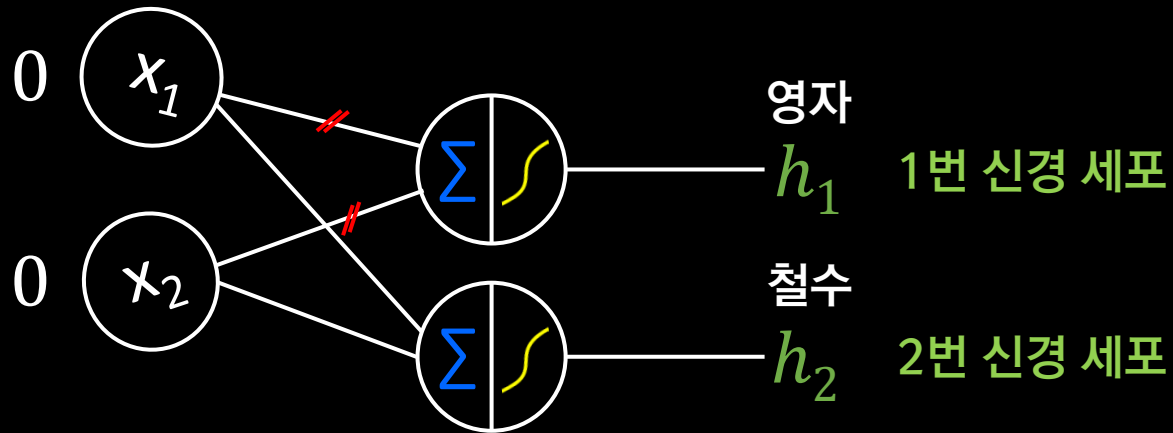
# 4개의 신경세포

$$(0, 0) \begin{pmatrix} w_{11} \\ w_{12} \end{pmatrix}$$



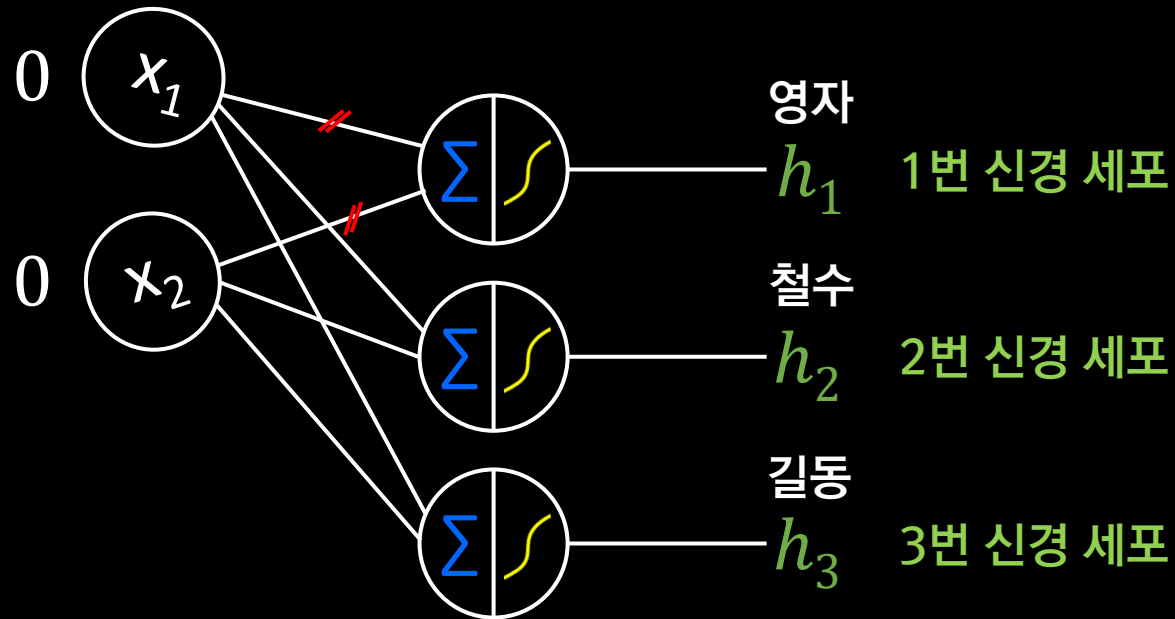
# 4개의 신경세포

$$(0, 0) \begin{pmatrix} w_{11} & w_{21} \\ w_{12} & w_{22} \end{pmatrix}$$



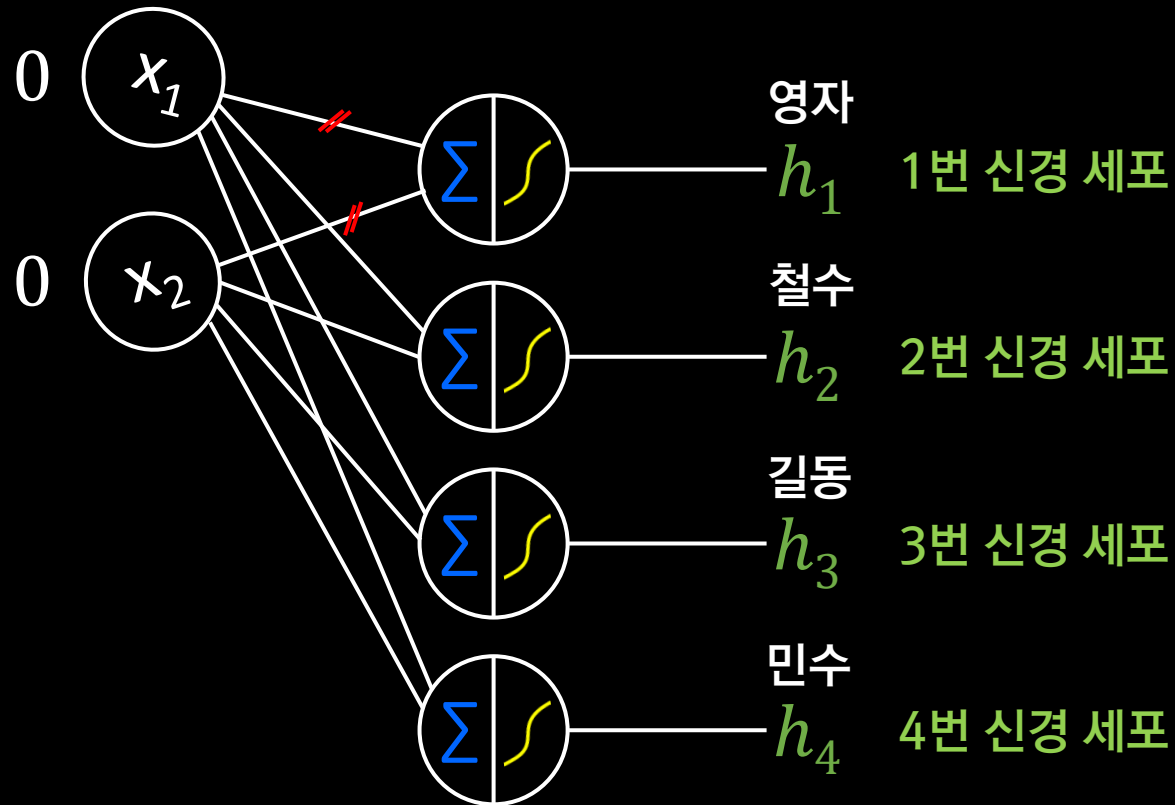
# 4개의 신경세포

$$(0, 0) \begin{pmatrix} w_{11} & w_{21} & w_{31} \\ w_{12} & w_{22} & w_{32} \end{pmatrix}$$

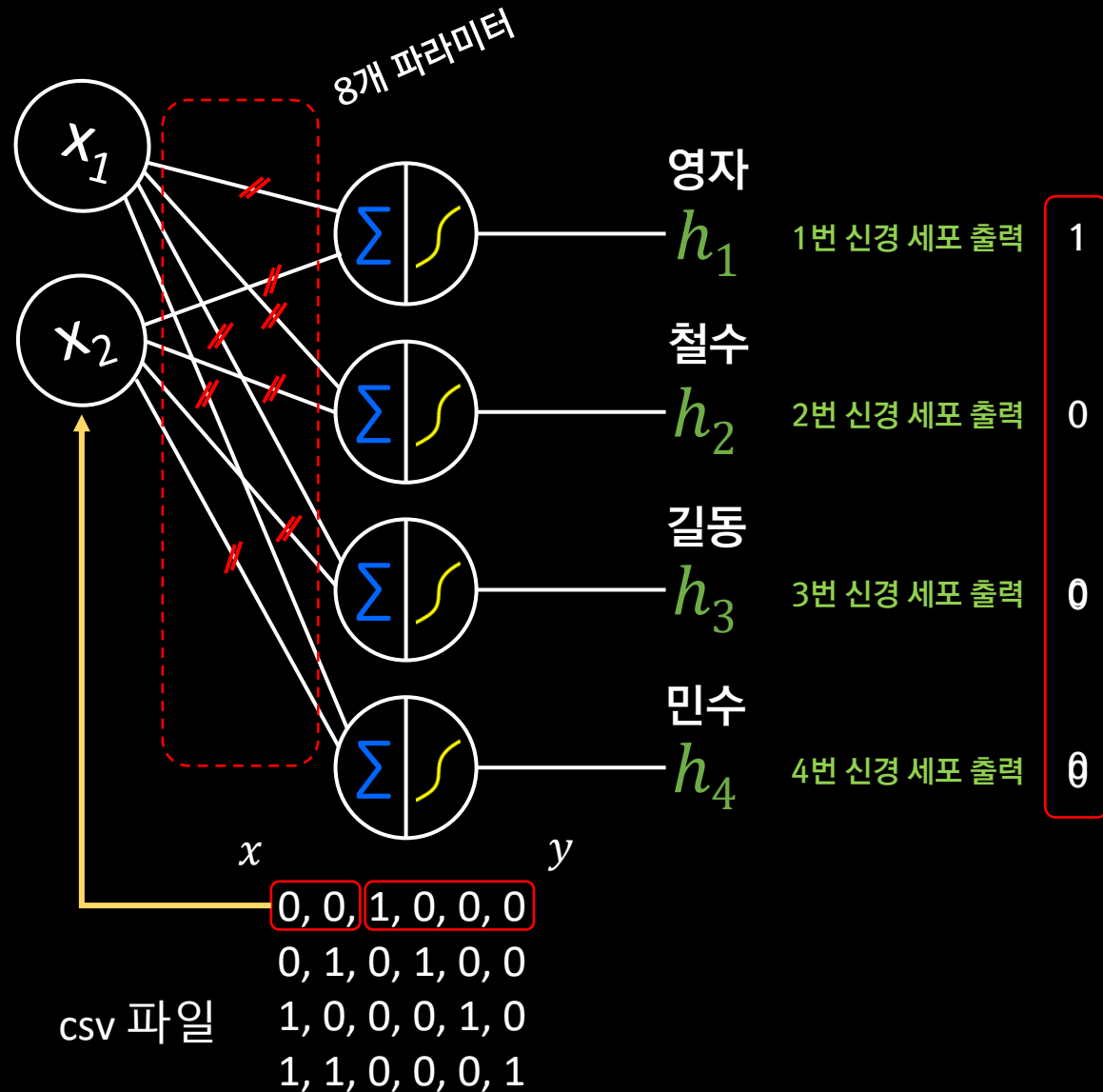


# 4개의 신경세포

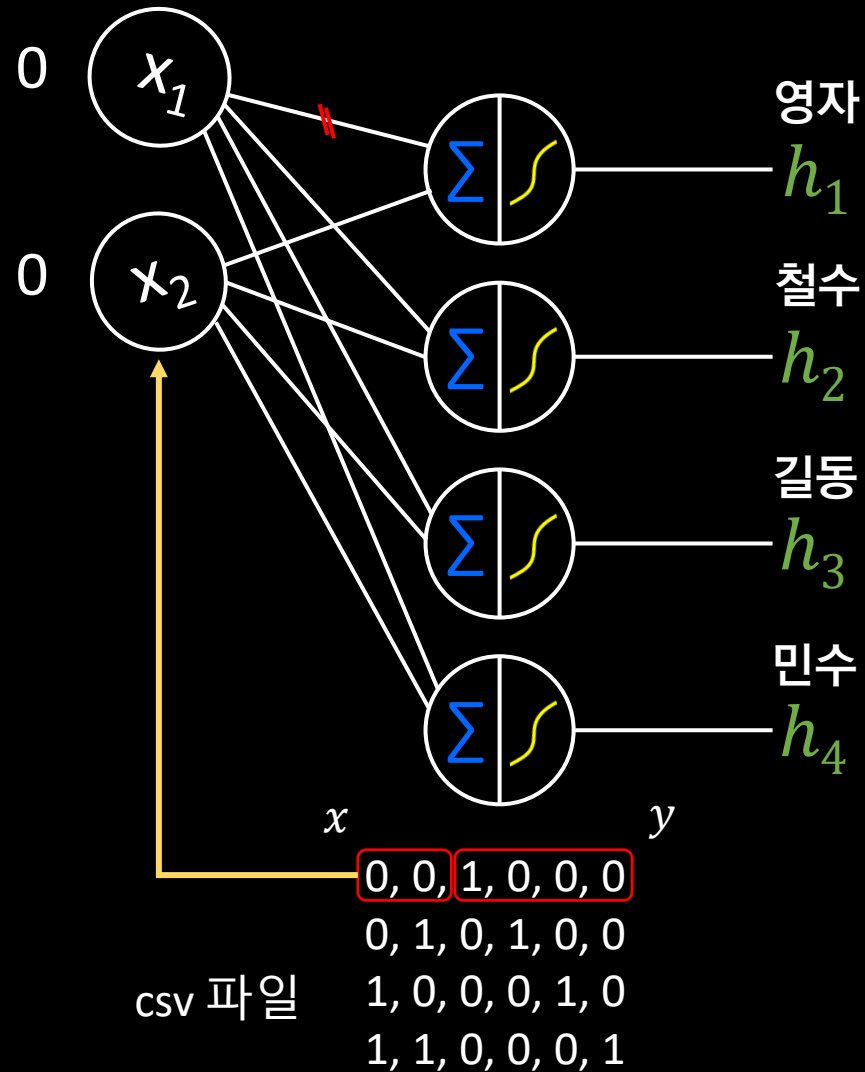
$$(0, 0) \begin{pmatrix} w_{11}, w_{21}, w_{31}, w_{41} \\ w_{12}, w_{22}, w_{32}, w_{42} \end{pmatrix}$$



# 4개의 신경세포

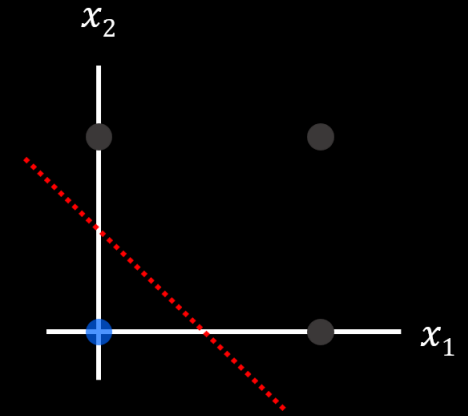
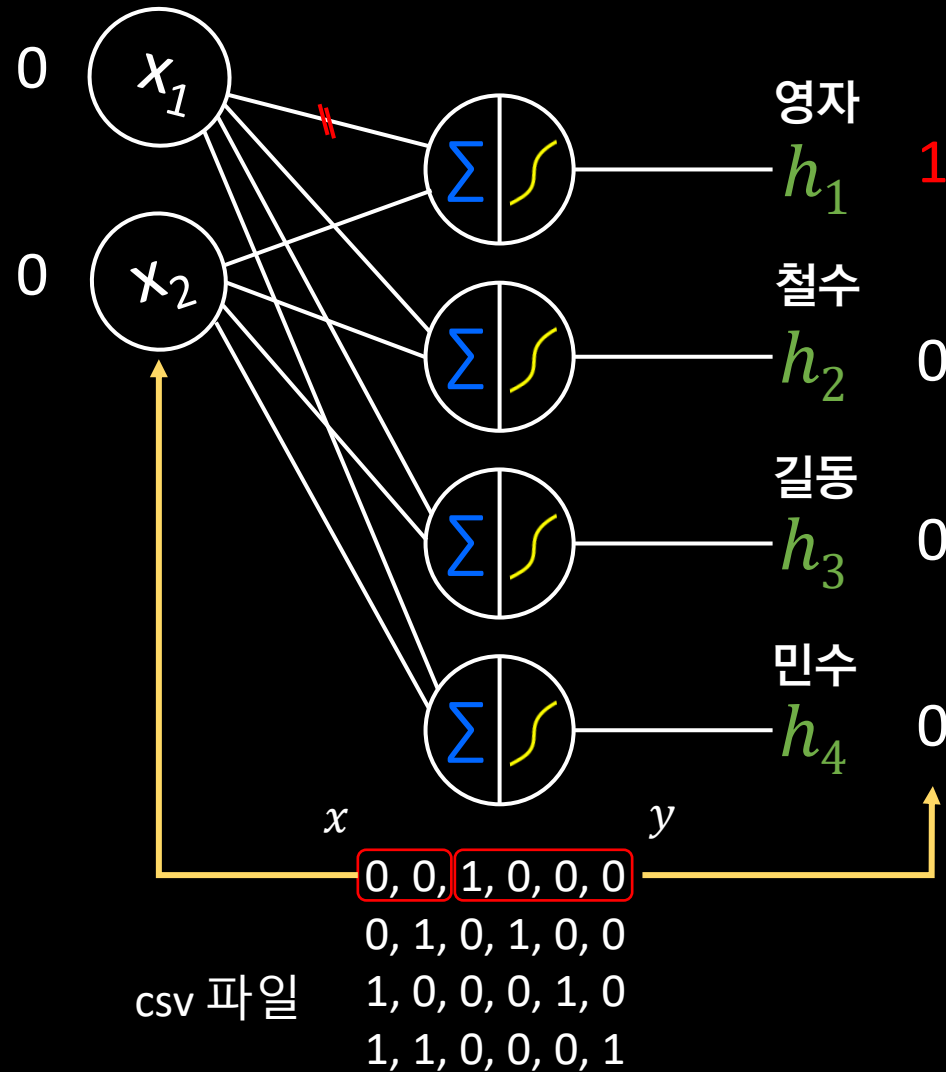


# 4개의 신경세포



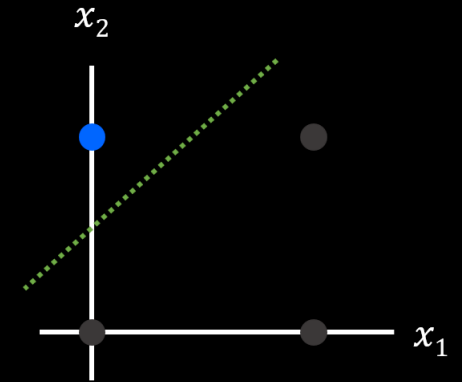
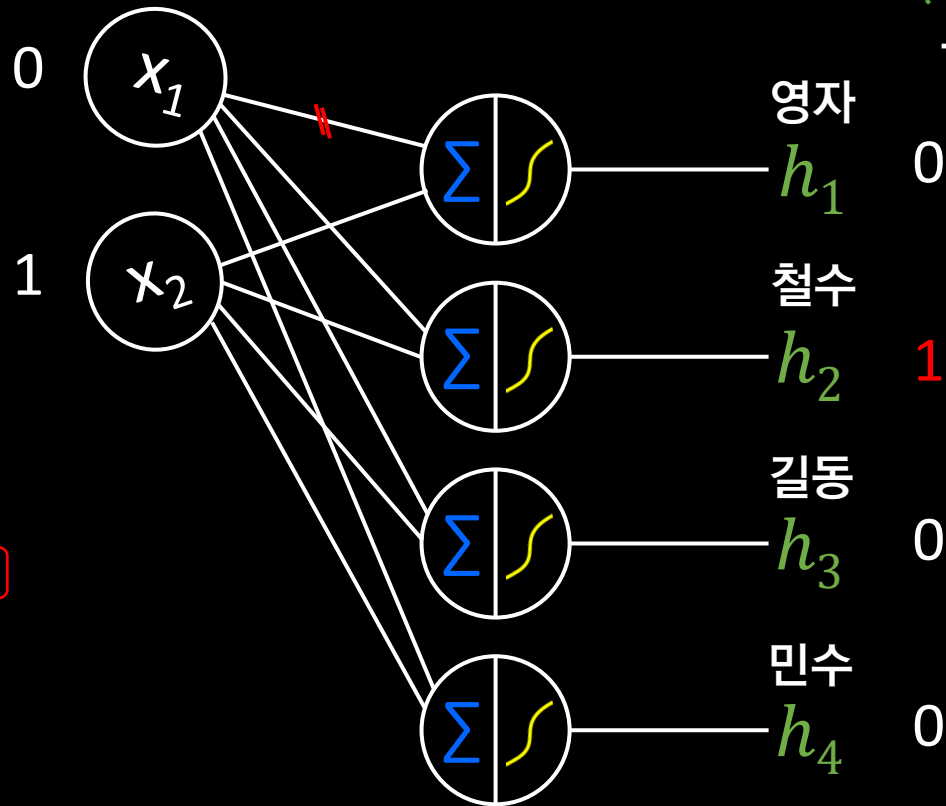


# 4개의 신경세포



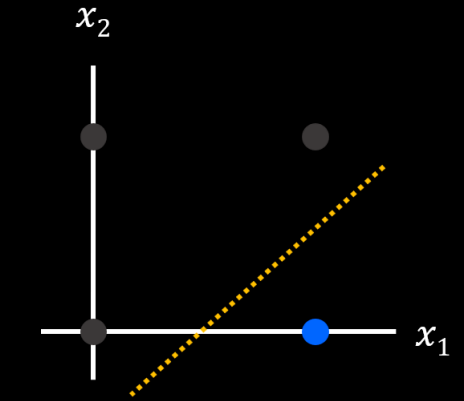
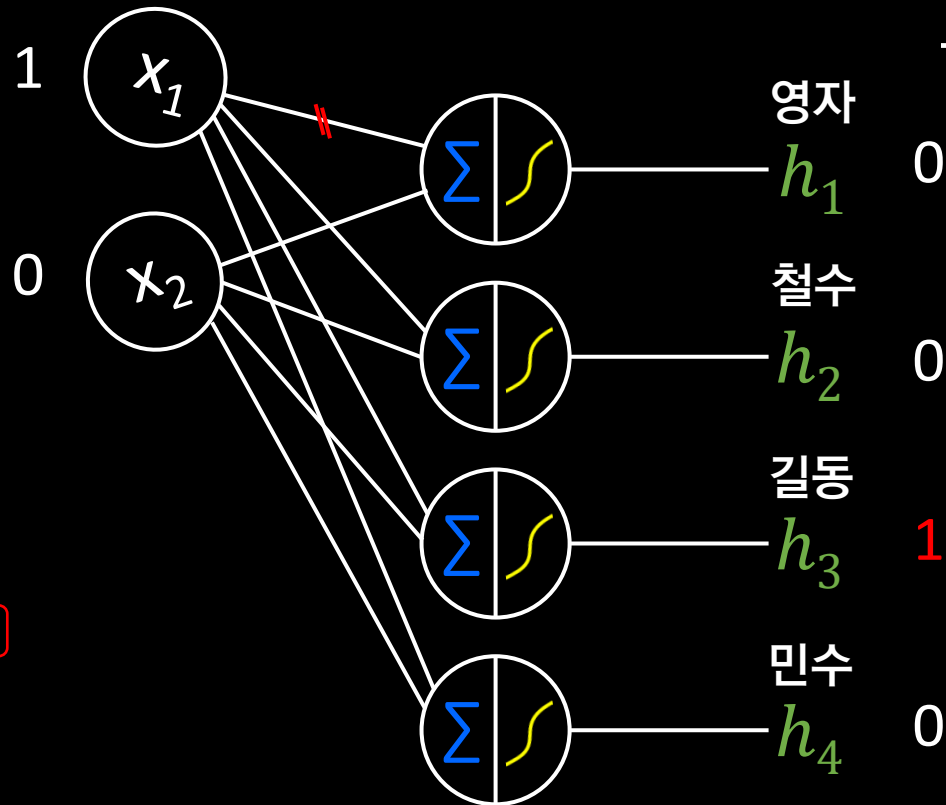
# 4개의 신경세포

$x$	$y$
0, 0, 1, 0, 0, 0	
0, 1, 0, 1, 0, 0	
1, 0, 0, 0, 1, 0	
1, 1, 0, 0, 0, 1	



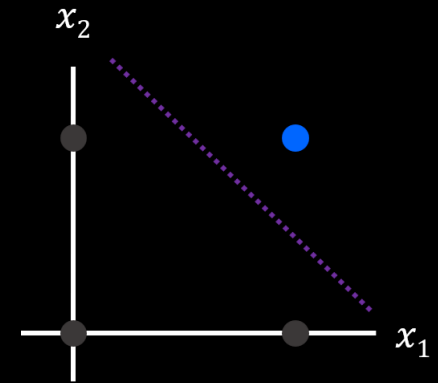
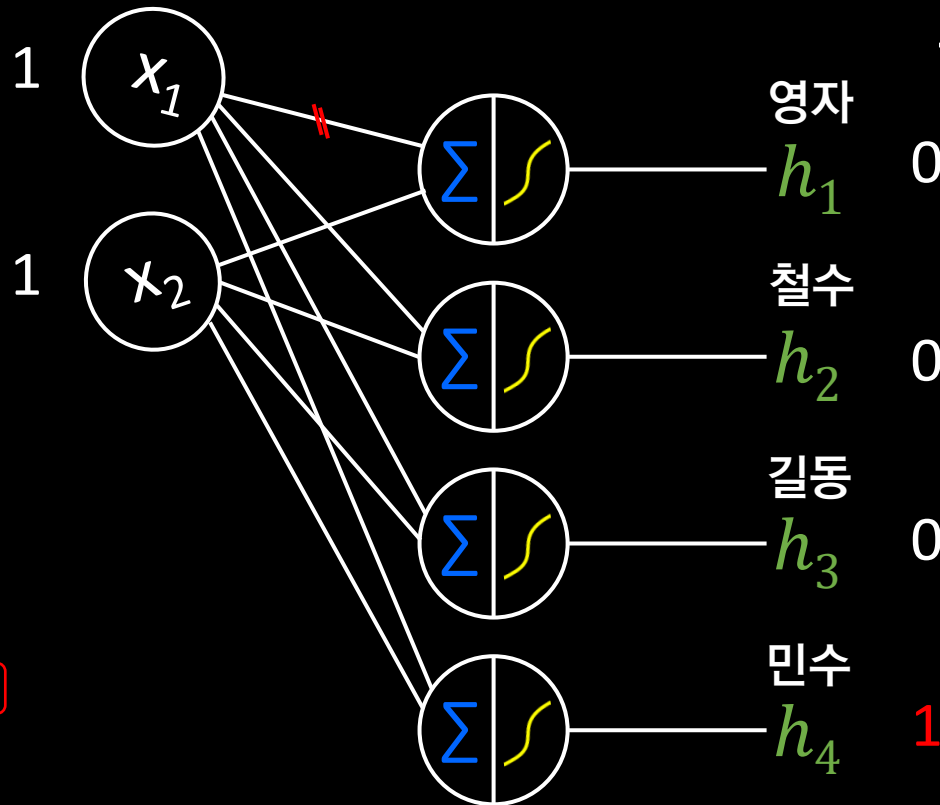
# 4개의 신경세포

$x$	$y$
0, 0, 1, 0, 0, 0	
0, 1, 0, 1, 0, 0	
1, 0, 0, 0, 1, 0	
1, 1, 0, 0, 0, 1	



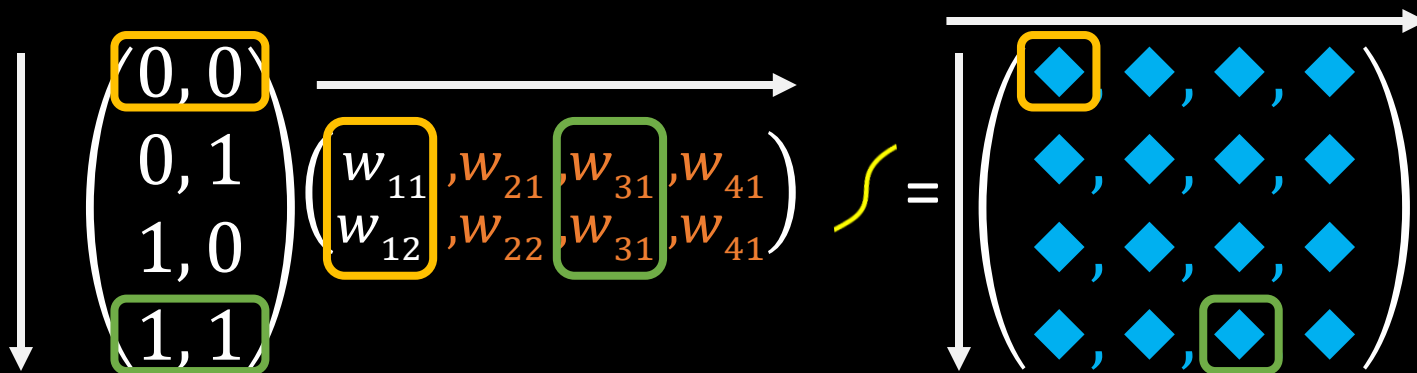
# 4개의 신경세포

$x$	$y$
0, 0, 1, 0, 0, 0	
0, 1, 0, 1, 0, 0	
1, 0, 0, 0, 1, 0	
1, 1, 0, 0, 0, 1	



신경세포 대답

$h$



$$\begin{pmatrix} 1, 0, 0, 0 \\ 0, 1, 0, 0 \\ 0, 0, 1, 0 \\ 0, 0, 0, 1 \end{pmatrix}$$

$y$

정답

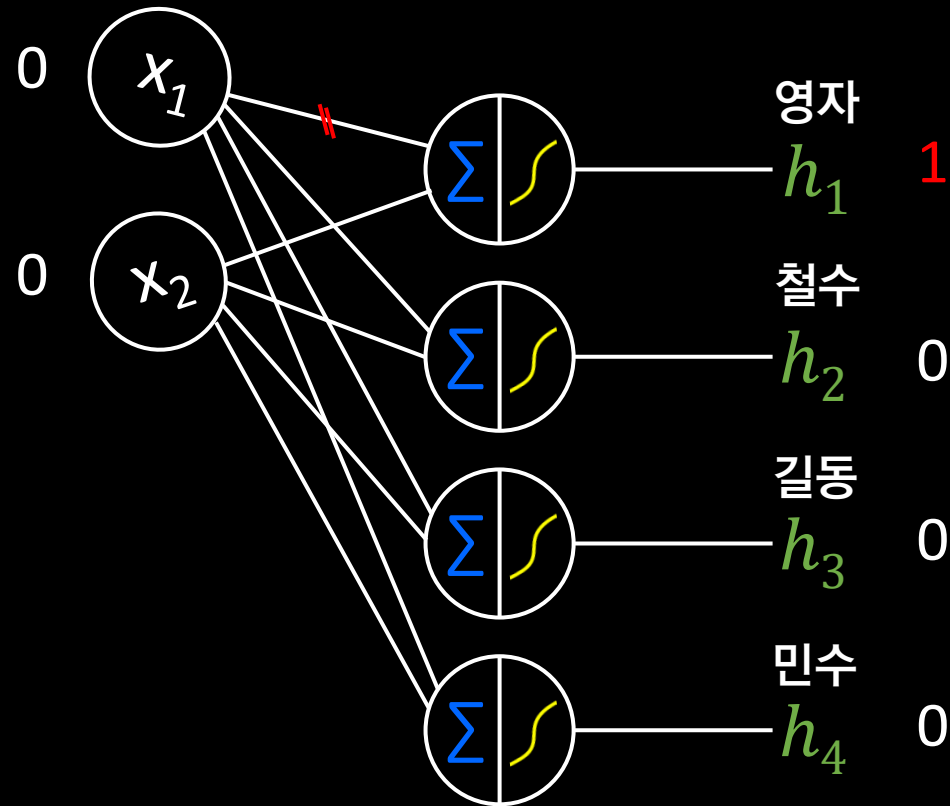
# 4개의 신경세포

- 특정 뉴런 하나만 ON, 나머지는 OFF (**one-hot** encoding)
- 직관적이고, 뇌 과학적으로 의미가 있음.

# 고려할 점

- 어느 신경 세포 하나가 1이면 나머지는 모두 0이 되어야 함.
- 하지만, 현재 4개 신경세포는 각각은 독립적으로 동작하여 그럴 수 없음.

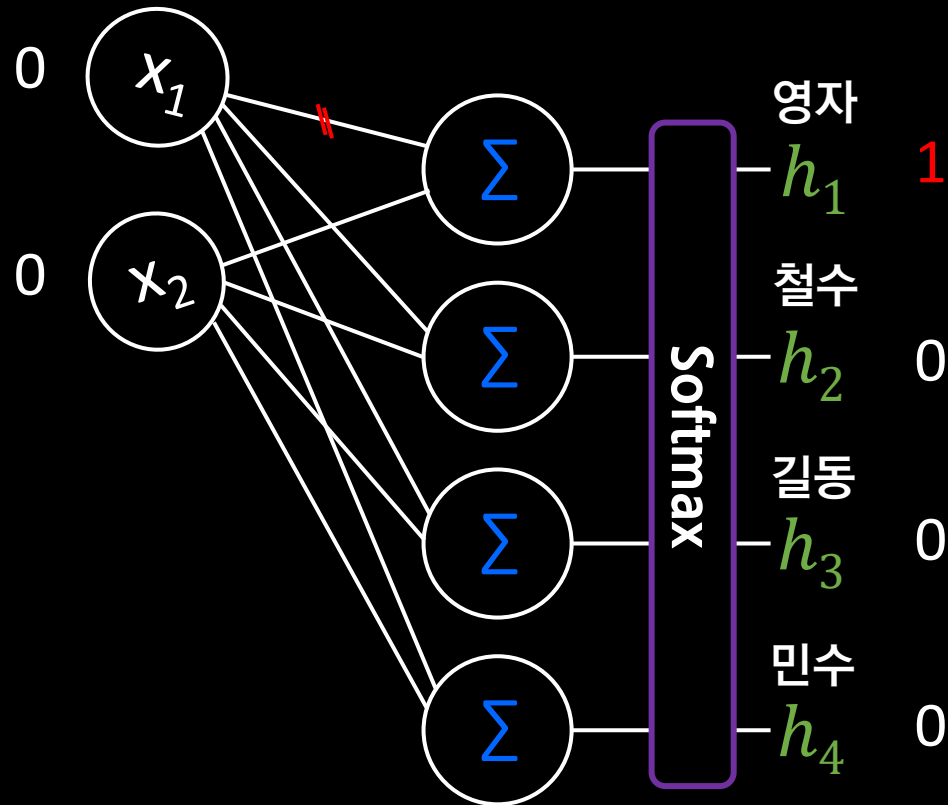
# 4개의 신경세포





(소맥-정규)

# Softmax?



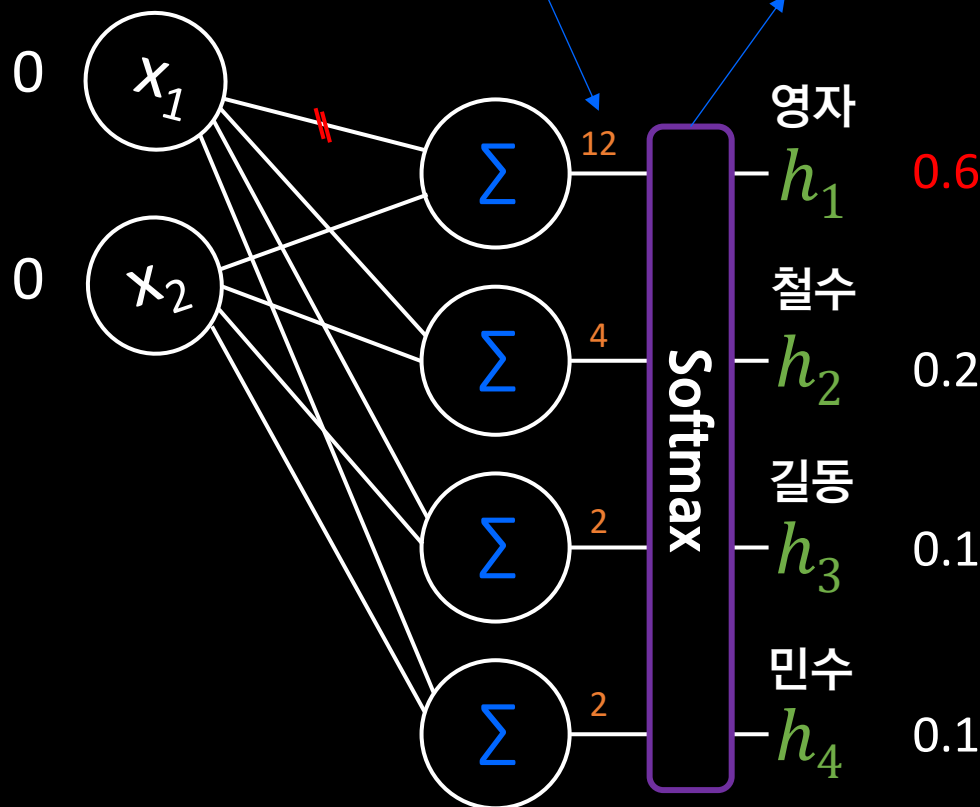
0과 1사이 값으로  
모두 합쳤을 때 1이 되도록

# Softmax?

(소맥-정규)

로짓(logits): 가공하지 않은, squash하지 않은, 생(raw) 출력값

모두 더한 값 20으로 각각을 나눔.  
0~1 사이의 값, 확률



0과 1사이 값으로  
모두 합쳤을 때 1이 되도록

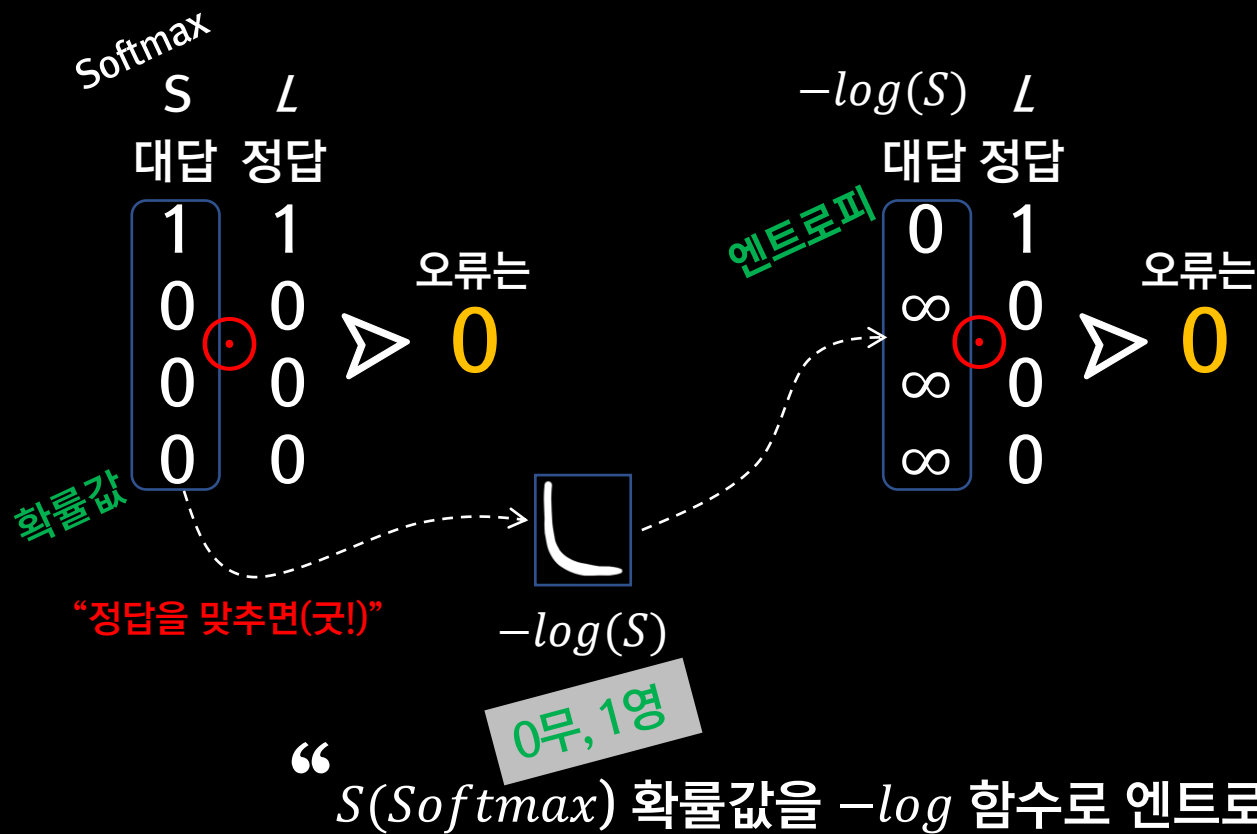
# Softmax 의미

- 신경망 출력 값을 모두 합쳤을 때 1이 되면 이해하기 쉬움(확률).
- 따라서 신경세포가 출력하는 4개의 값을 0과 1 사이의 값을 갖도록 하고 모두 합쳤을 때 1이 되도록 조절하는 기능이 있다면!
- Softmax (소프트맥스) : 소맥정규

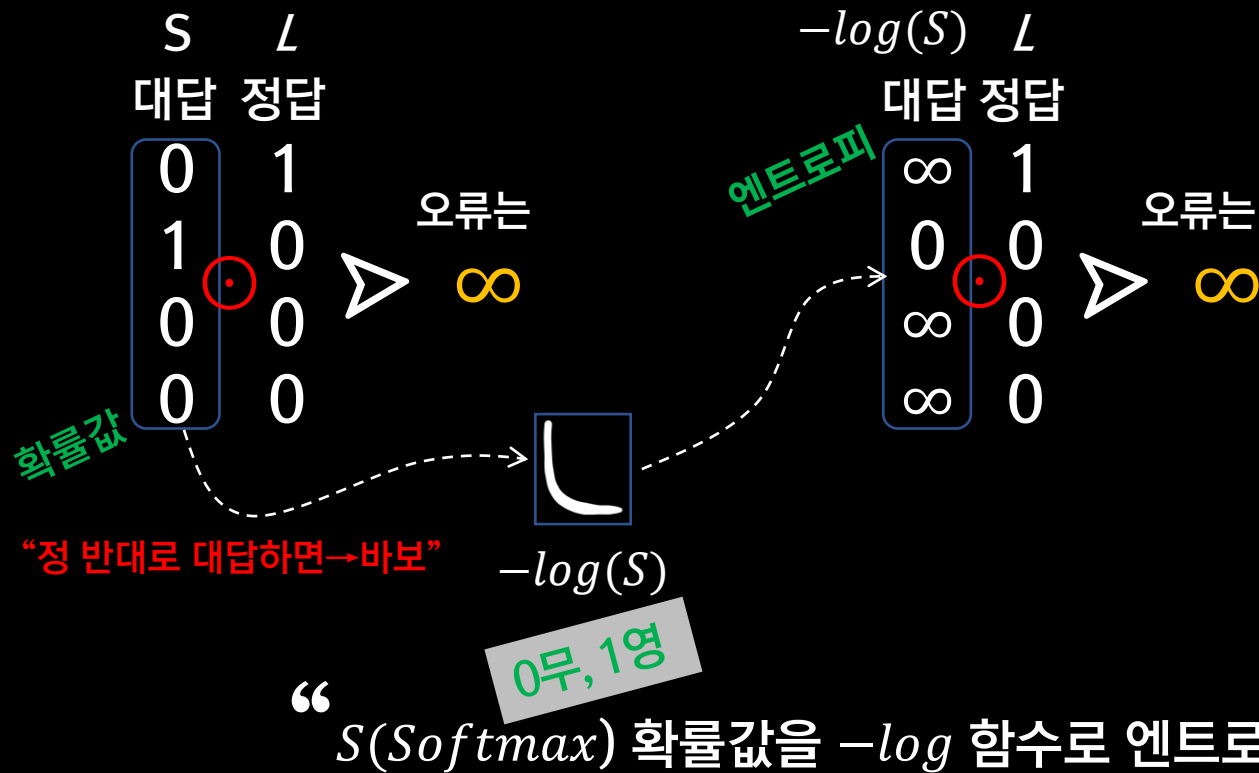
# 오류함수

- 신경망의 **대답**과 **정답** 사이의 차이  
(**오류**)
- 신경망의 **대답**은 뭐?
- 정답을 맞추면 **차이(오류)** = 0,  
맞추지 못하면 **차이(오류)** =  $\infty$

# 오류함수



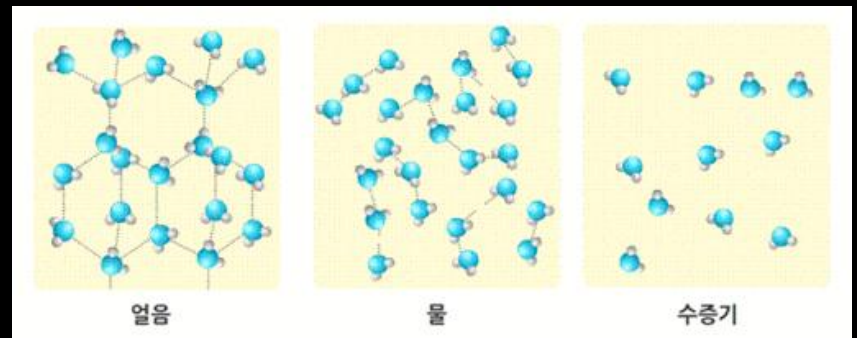
# 오류함수



# 엔트로피(독일어 entropie, 1850s)

- 물리 열역학에서 물질의 상태, 무질서도를 표현하는 말.  
얼음(엔트로피/무질서도 작음), 물(보통), 수증기(큼)
- 수학 통계학에서는 불확실성과 관련
- 매우 안정적 : (확률이 1 = 엔트로피 거의 0)
- 매우 불확실 : (확률이 0.000001 = 엔트로피 거의 무한대),
- $-\log(\text{확률}) \rightarrow \text{엔트로피 (entropy)} : \text{무질서도}$

0무, 1영



$$E = -L \log(S)$$

정답



`sess.run(train)`

$$E = -\sum_i L_i \log(S_i)$$

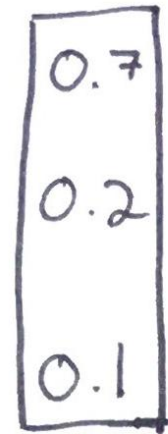
크로스 엔트로피 (*cross - entropy*) 함수에서 *cross* 는  
 ‘비교한다’는 의미로  
 모델의 예측과 정답을 서로 비교(*cross - check*)하여  
 오류를 측정한다는 의미

크로스 엔트로피 함수 = 크로스 체크하여 엔트로피 오류를 계산하는 함수



# 크로스 엔트로피 오류함수

$$D(S, L) = - \sum_i L_i \log(S_i)$$

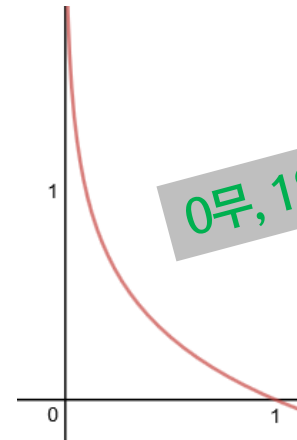


$S(y)$

확률값



$L$



확률을 엔트로피(불확실성)로  
바꿔주는 함수

$$-\sum_{i=1}^2 L_i \log(S_i)$$

$$-L_1 \log(S_1) + L_2 \log(S_2)$$

$$L_2 = 1 - L_1 \text{ (정답은 1 혹은 0 둘중 하나)}$$

$$S_2 = 1 - S_1 \text{ ('0일 확률'은 } 1 - \text{'1일 확률'})}$$

$$-L_1 \log(S_1) + (1 - L_1) \log(1 - S_1)$$

$$-y \log(h) + (1 - y) \log(1 - h)$$

논리(로지스틱) 회귀에서의 오류함수

**바이너리 크로스 엔트로피 로스(loss) 함수**

- 절대값 오류 L1
- 제곱 오류 L2
- 바이너리 크로스 엔트로피 오류  
(둘 중 하나로 분류)
- 다항 크로스 엔트로피 오류  
(여러 개 중 하나로 분류)

# 이미 있는 오류 함수

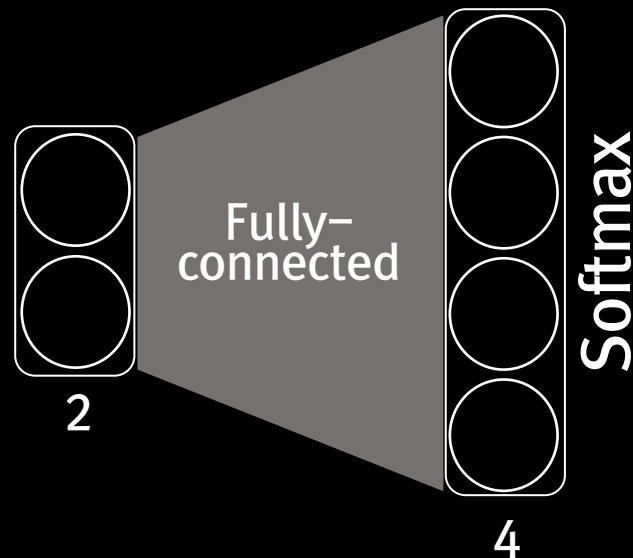
로짓(생) 값과 정답을 주면  
로짓 값을 정규화한 후(softmax)  
정답과 비교하여 오류(무질서량, 불확실성, 엔트로피)를  
계산해주는 함수

`softmax_cross_entropy_with_logits(logits, y_data)`

같으면 0, 전혀 다르면  $\infty$ 를 반환하는 함수

# (실습) 14.py

- 4가지 중 하나로 인식
- 2개의 입력을 갖는 신경세포 4개
- 각 신경세포에 1개의 바이어스



$$\begin{matrix}
 \mathbf{x} \\
 \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}
 \end{matrix}
 \begin{matrix}
 \mathbf{W} \\
 \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}
 \end{matrix}
 \begin{matrix}
 \text{Softmax} \\
 \boxed{\phantom{0000}}
 \end{matrix}
 =
 \begin{matrix}
 \mathbf{S} \text{ (대답)} \\
 \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}
 \end{matrix}$$

Handwritten annotations: A blue horizontal line under the first row of matrix  $\mathbf{x}$ . A blue circle around the first element (0) of the first row of matrix  $\mathbf{S}$ . A green horizontal line under the first row of matrix  $\mathbf{S}$ . A vertical yellow box labeled "Softmax" is placed between the weight matrix  $\mathbf{W}$  and the result matrix  $\mathbf{S}$ .

$$\begin{matrix}
 \mathbf{L} \text{ (정답)} \\
 \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}
 \end{matrix}$$

Handwritten annotation: A pink horizontal line under the first row of matrix  $\mathbf{L}$ .