

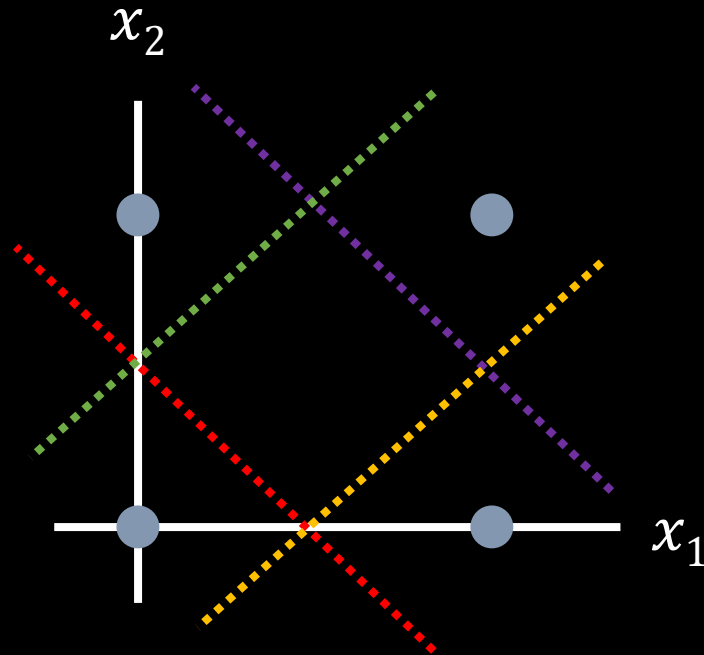
AI and Deep Learning

6. Logistic Regression(2)

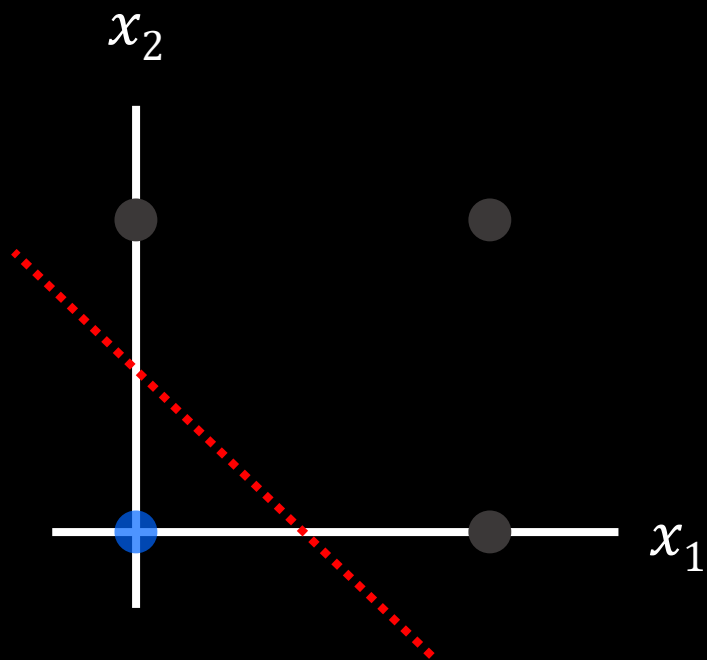
Yung-Cheol BYUN, Jeju National Univ.

여러 클래스

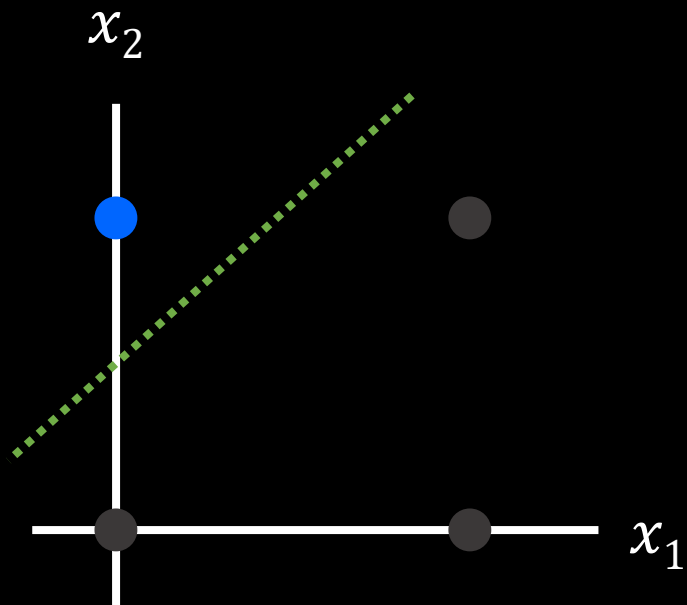
4개의 결정 경계가 필요



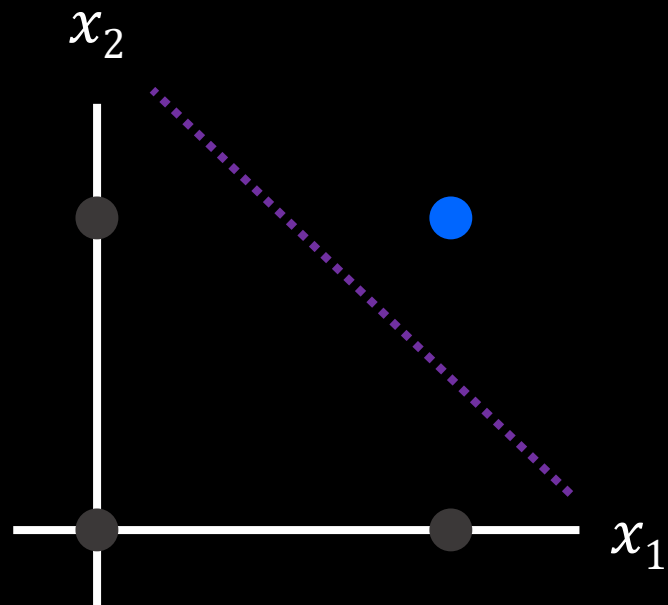
{위에서 본 모습}



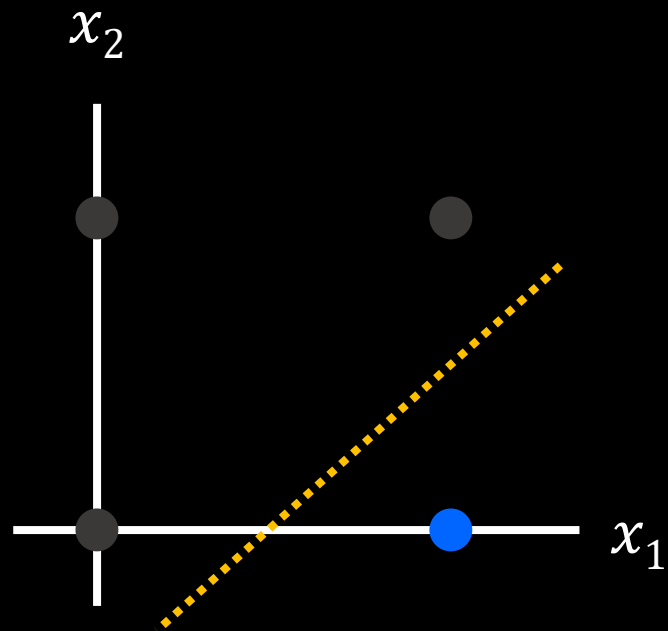
{위에서 본 모습}



{위에서 본 모습}

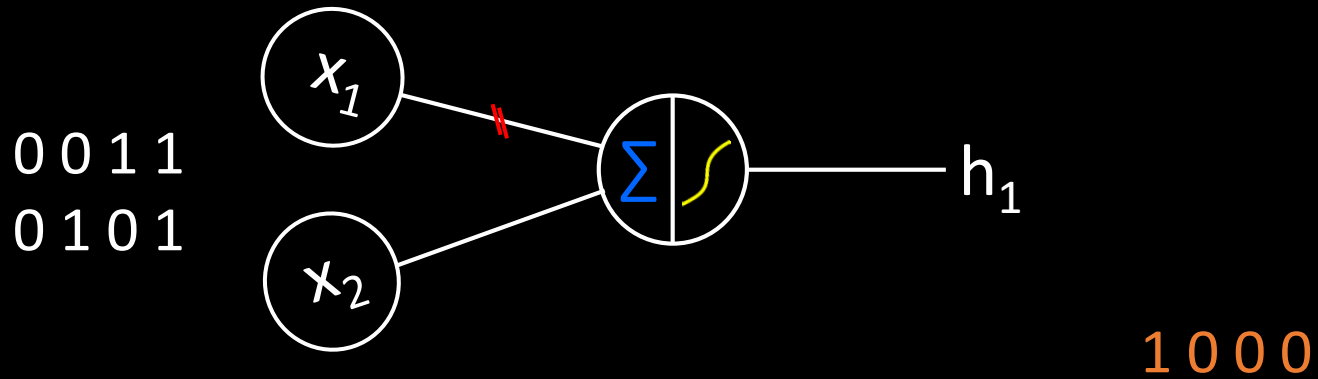


{위에서 본 모습}

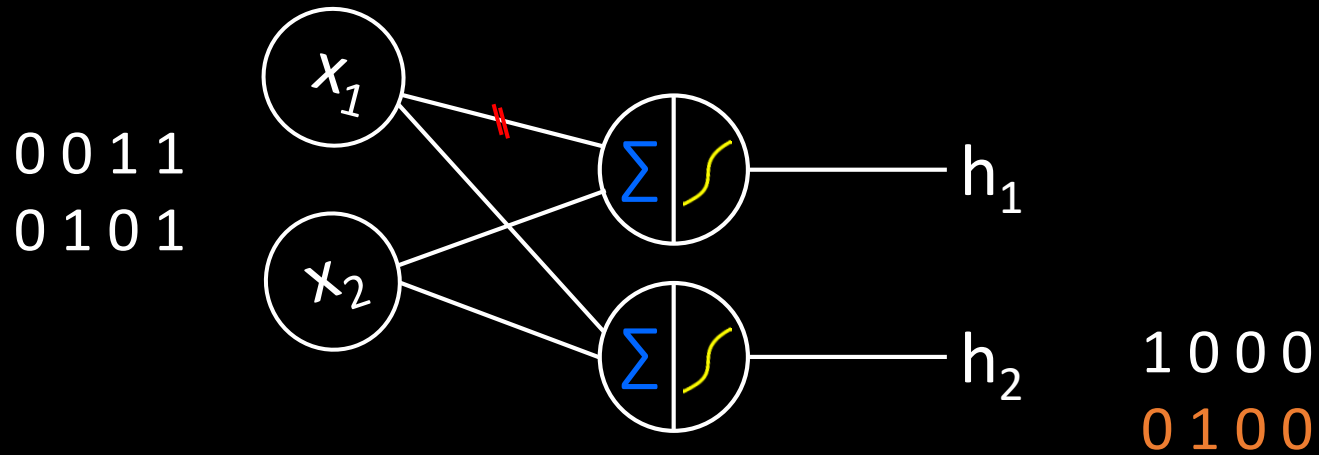


{위에서 본 모습}

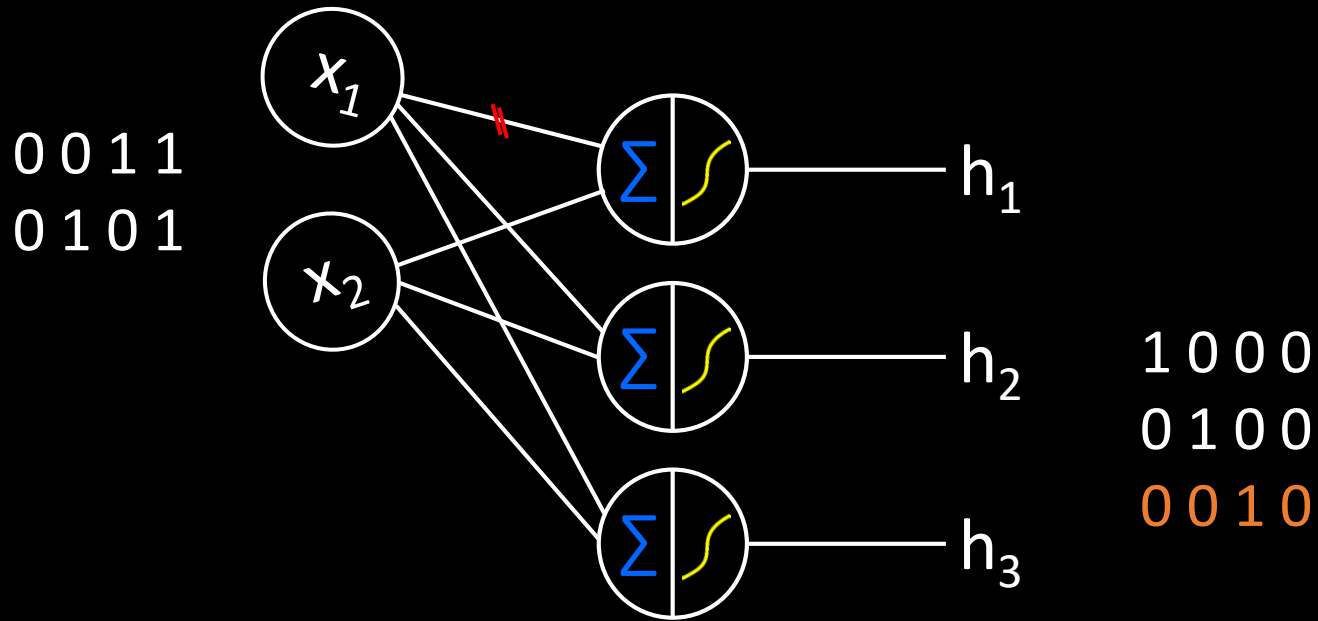
여러 클래스



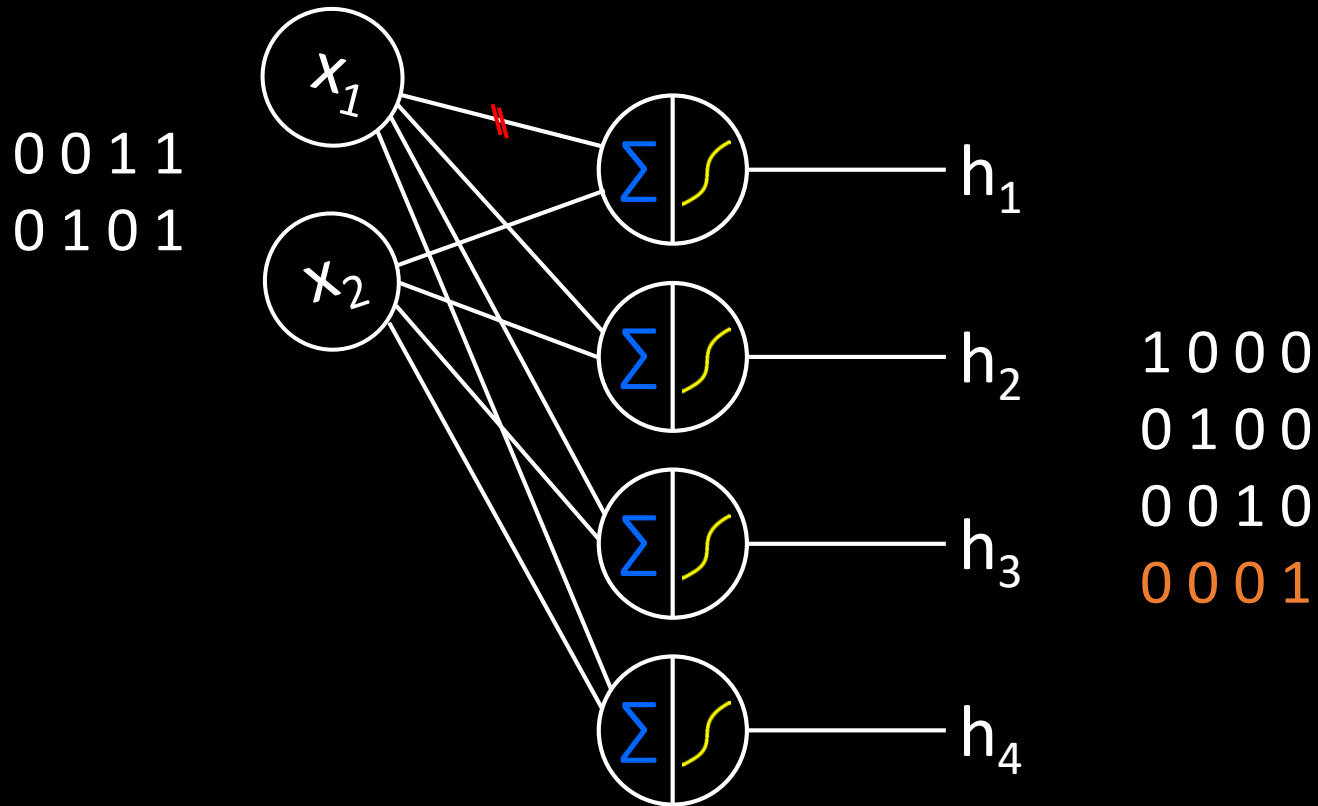
여러 클래스



여러 클래스

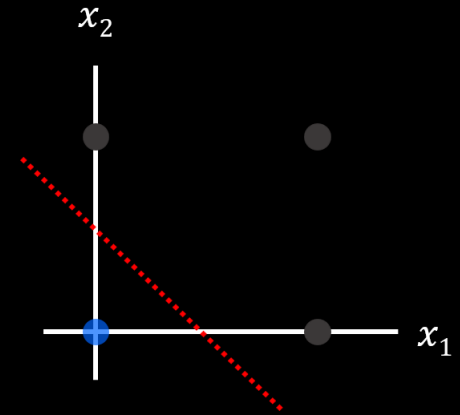
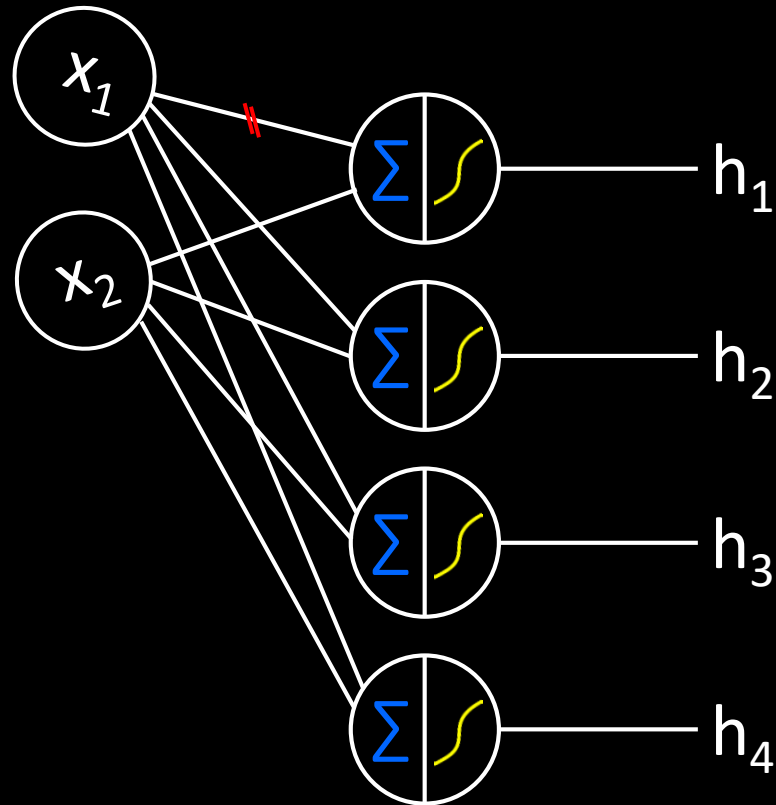


여러 클래스



여러 클래스

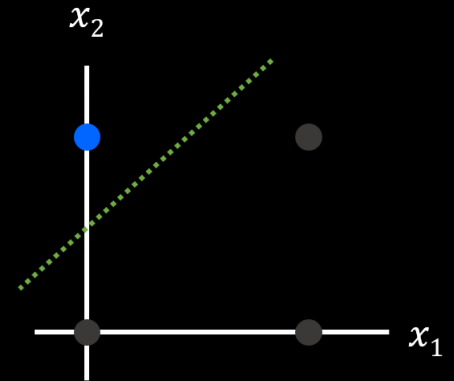
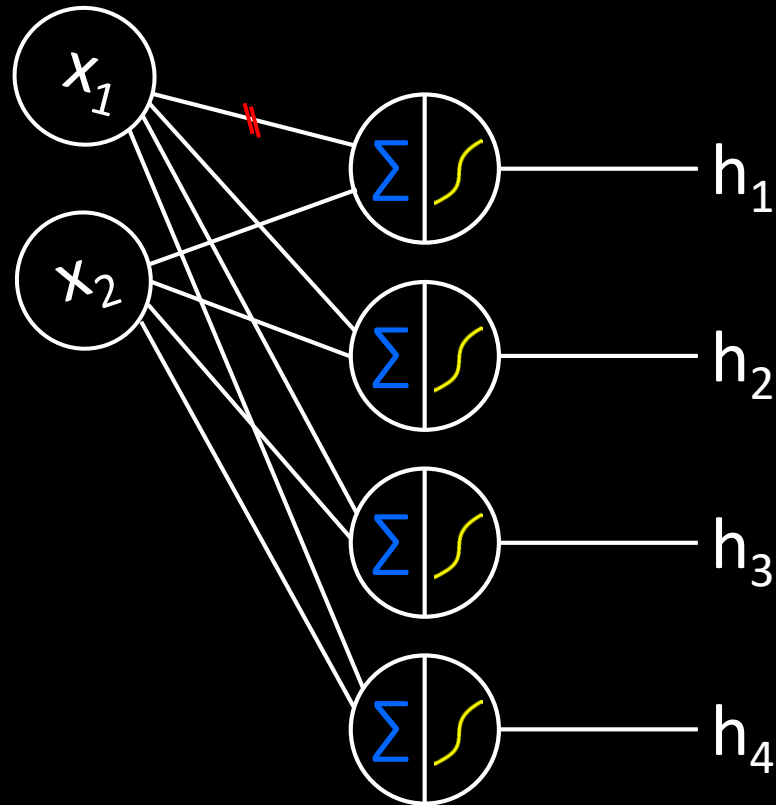
0	0	1	1
0	1	0	1



1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

여러 클래스

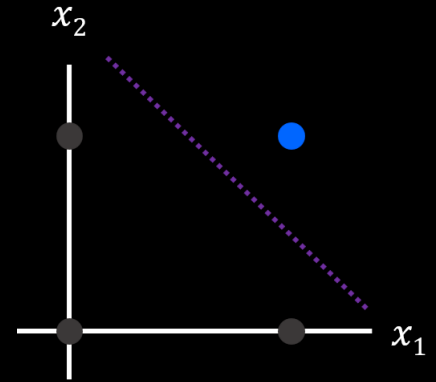
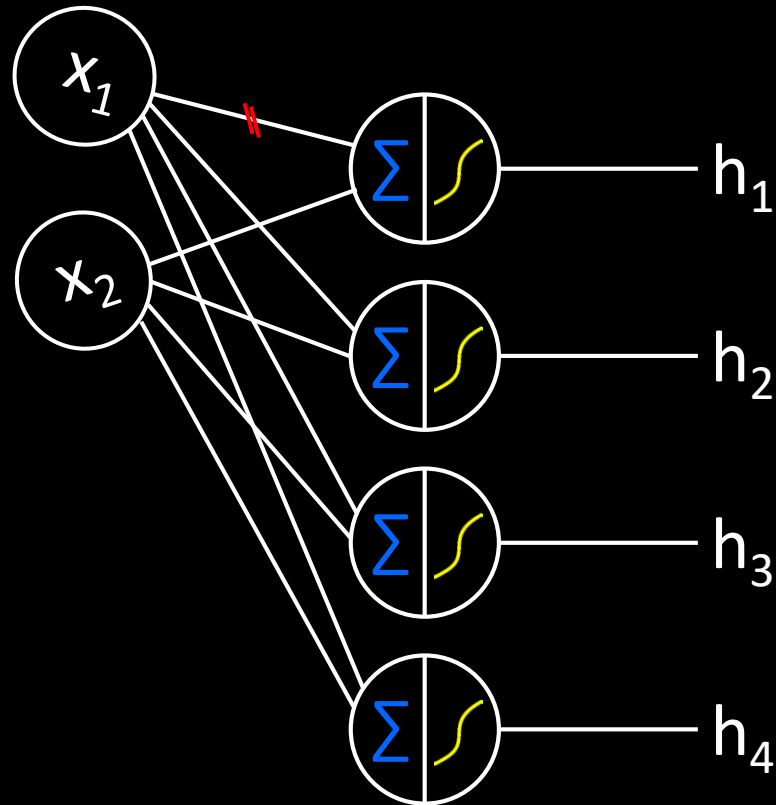
0	0	1	1
0	1	0	1



1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

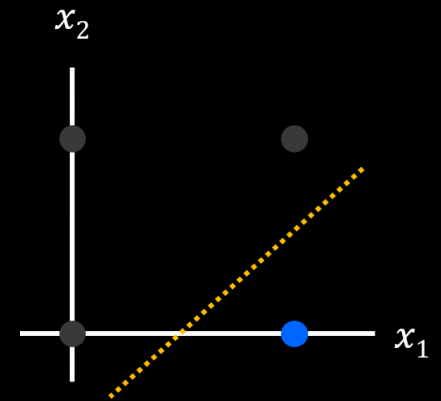
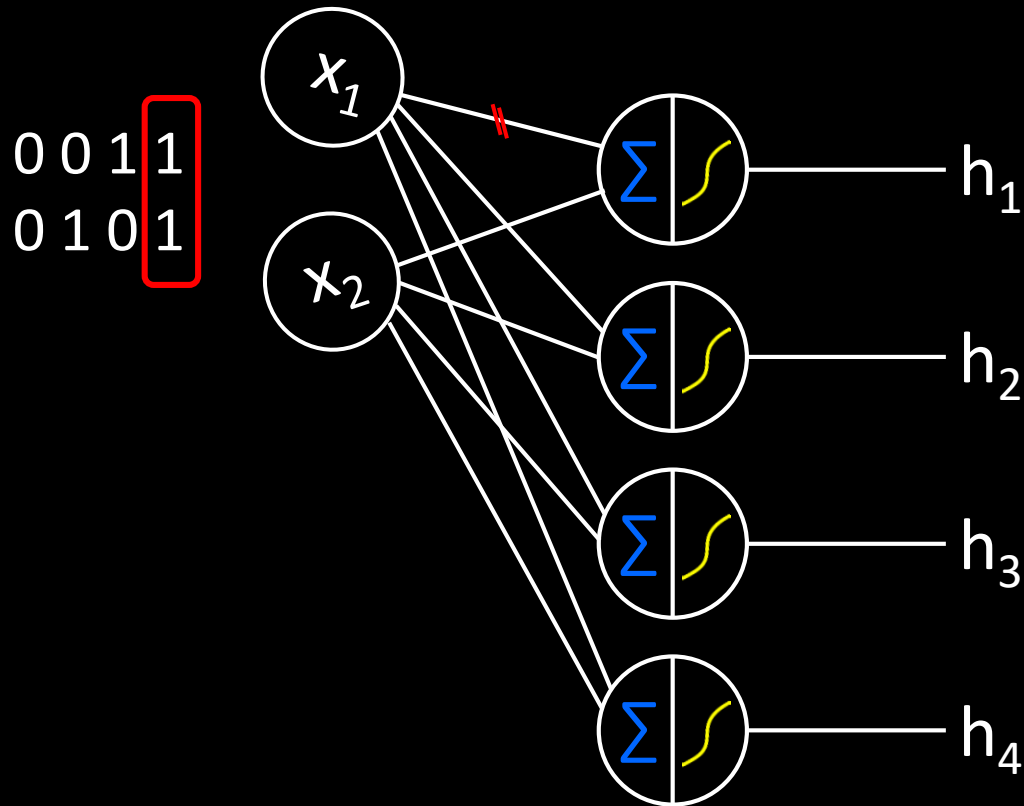
여러 클래스

0	0	1	1
0	1	0	1



1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

여러 클래스



1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

여러 클래스

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

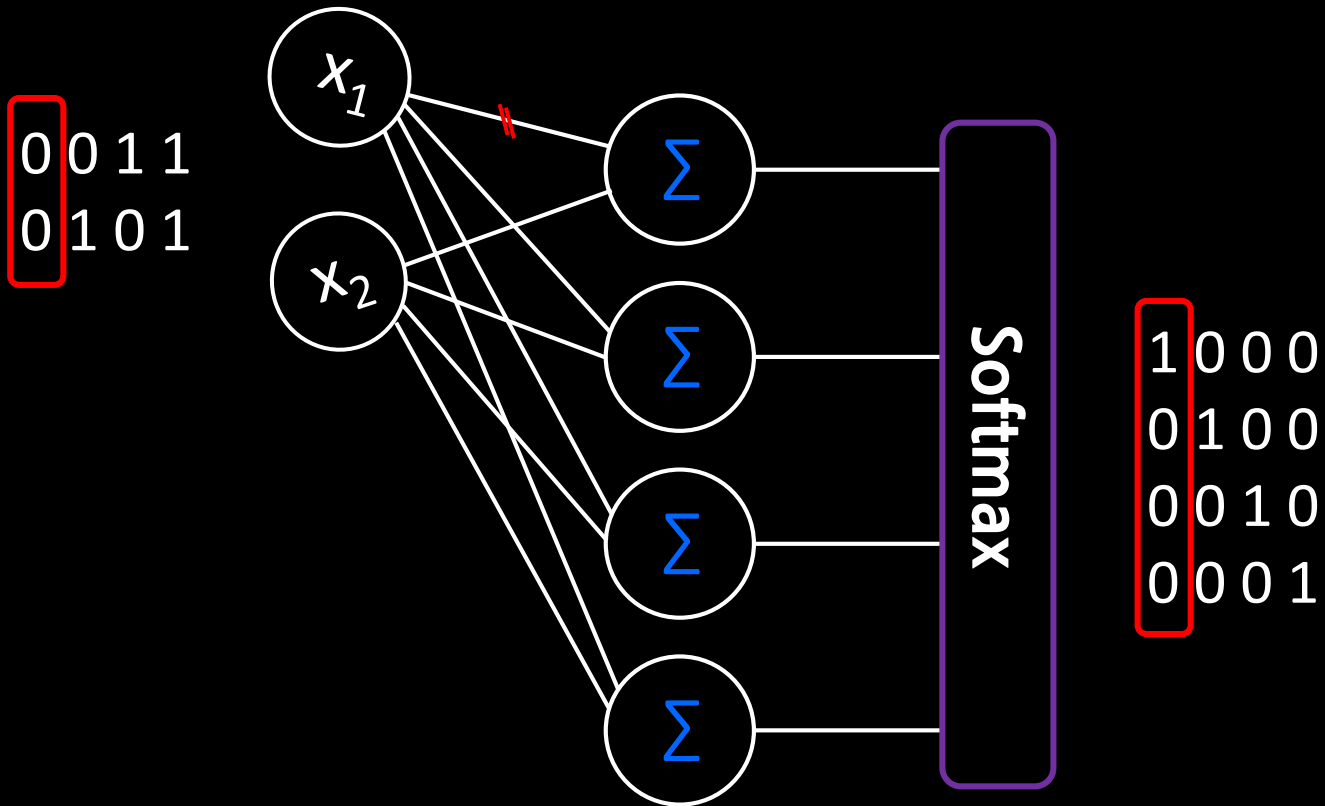
문제점

- 신경세포는 각각 독립적으로 출력값을 냄
- 어느 신경 세포 하나가 1이면 나머지는 모두 0이 되어야 하지만 이를 제어할 수 없음.
- 출력값이 0과 1사이의 값으로 제각각임.

문제점

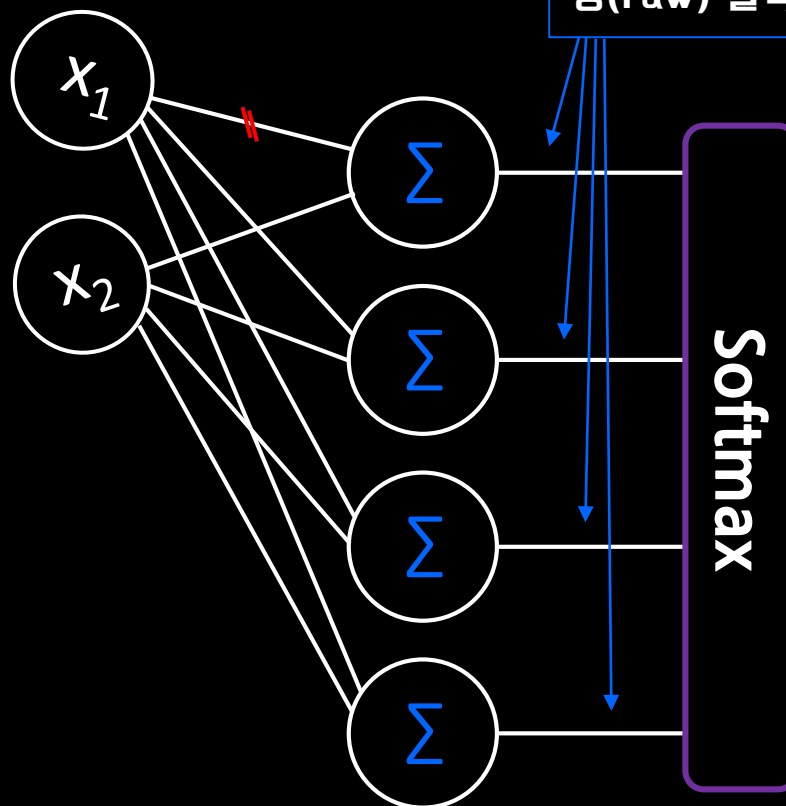
- 출력값을 모두 합쳤을 때 10이 되면 이해하기 쉬움(확률).

Softmax



Softmax

0	0	1	1
0	1	0	1



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

1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

Softmax (소맥정규)

- Softmax는 logits 값들을 0~1 사이 값으로 만들고(squash) 모두 더했을 때 1이 되도록 정규화
- 값이 12, 4, 2, 2일 경우 모두 더한 값 20으로 각각을 나누면 0.6, 0.2, 0.1, 0.1

오류함수

- 오류 함수는 Softmax가 생성하는 값과 정답 사이의 차이(오류)를 계산
- 정답을 맞추면 차이(오류) = 0, 맞추지 못하면 차이(오류) = ∞

오류함수

“Softmax 출력값과 정답의
차이(오류) 계산 방법”

Softmax

1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1

소프 정답

1 1

0 0

0 0

0 0

?

> 0

소프 정답

0 1

1 0

0 0

0 0

?

>

무한대

0무1영, 곱

소프 정답

0 1

∞ 0

∞ 0

∞ 0

⊙

> 0

소프 정답

∞ 1

0 0

∞ 0

∞ 0

⊙

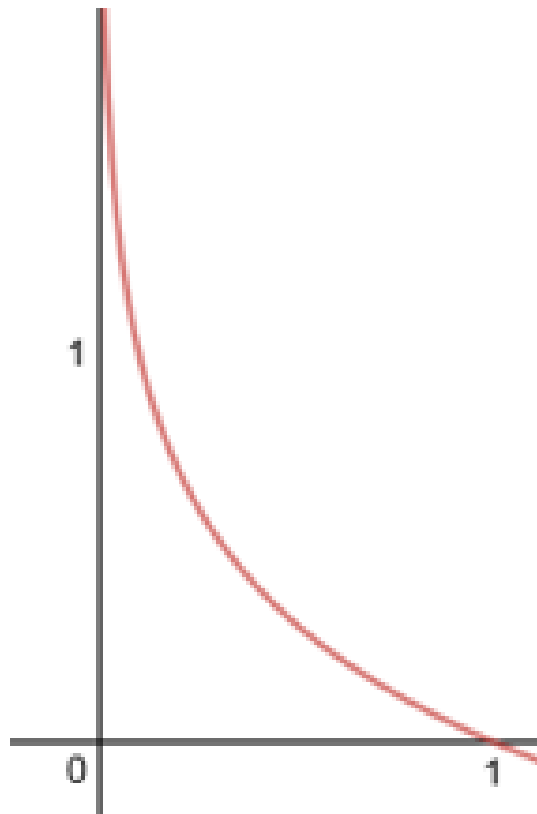
>

무한대

오류함수

0무1영

$$-\log(s)$$



Softmax 출력값 s

$$-L \log(s)$$

$$-\sum_i L_i \log(s_i)$$

오류함수

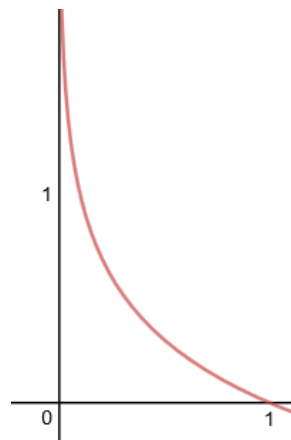
$$D(S, L) = - \sum_i L_i \log(S_i)$$

0.7
0.2
0.1

$S(y)$

1.0
0.0
0.0

L



로짓값과 정답을 주면
로짓값을 정규화한 후(softmax)
이를 정답과 비교하여
오류(무질서량, 엔트로피)를
계산해주는 함수가 있으면 좋겠다!

`softmax_cross_entropy_with_logits(logits, y_data)`

(실습) 14.py

- 2개의 입력을 갖는 신경세포 4개
- 각 신경세포에 1개의 바이어스 입력