



PART 2

다층 퍼셉트론(MLP)

1장. 다층 퍼셉트론의 기본 구조

딥러닝 & 강화학습 담당
이재화 강사





이 장에서 다룰 내용

1. 다층 퍼셉트론 신경망 구조와 은닉 계층
2. 비선형 활성화 함수와 ReLU 함수
3. 민스키의 XOR 문제와 비선형 활성화 함수의 필요성
4. 다층 퍼셉트론 신경망을 지원하는 함수 구현
5. Part 1에서 다뤘던 세 문제를 다층 퍼셉트론으로 풀어보기





은닉 계층은

출력에 직접 드러나지 않는 계층을 의미하며,
은닉계층이 생성하는 중간 표현을 '은닉 벡터'라 한다.

다층 퍼셉트론은

다층 퍼셉트론에서 각각의 계층은
단층 퍼셉트론과 같은 내부구조를 갖습니다.

A 즉, 하나의 계층에 속한 퍼셉트론들은 동일한 입력을 공유,
하지만 서로 어떠한 연결도 없기에 영향을 주고 받을 수 없다.

B 인접한 계층끼리는 앞 계층의 출력이 뒷 계층의

모든 퍼셉트론에 공통 입력으로 제공.

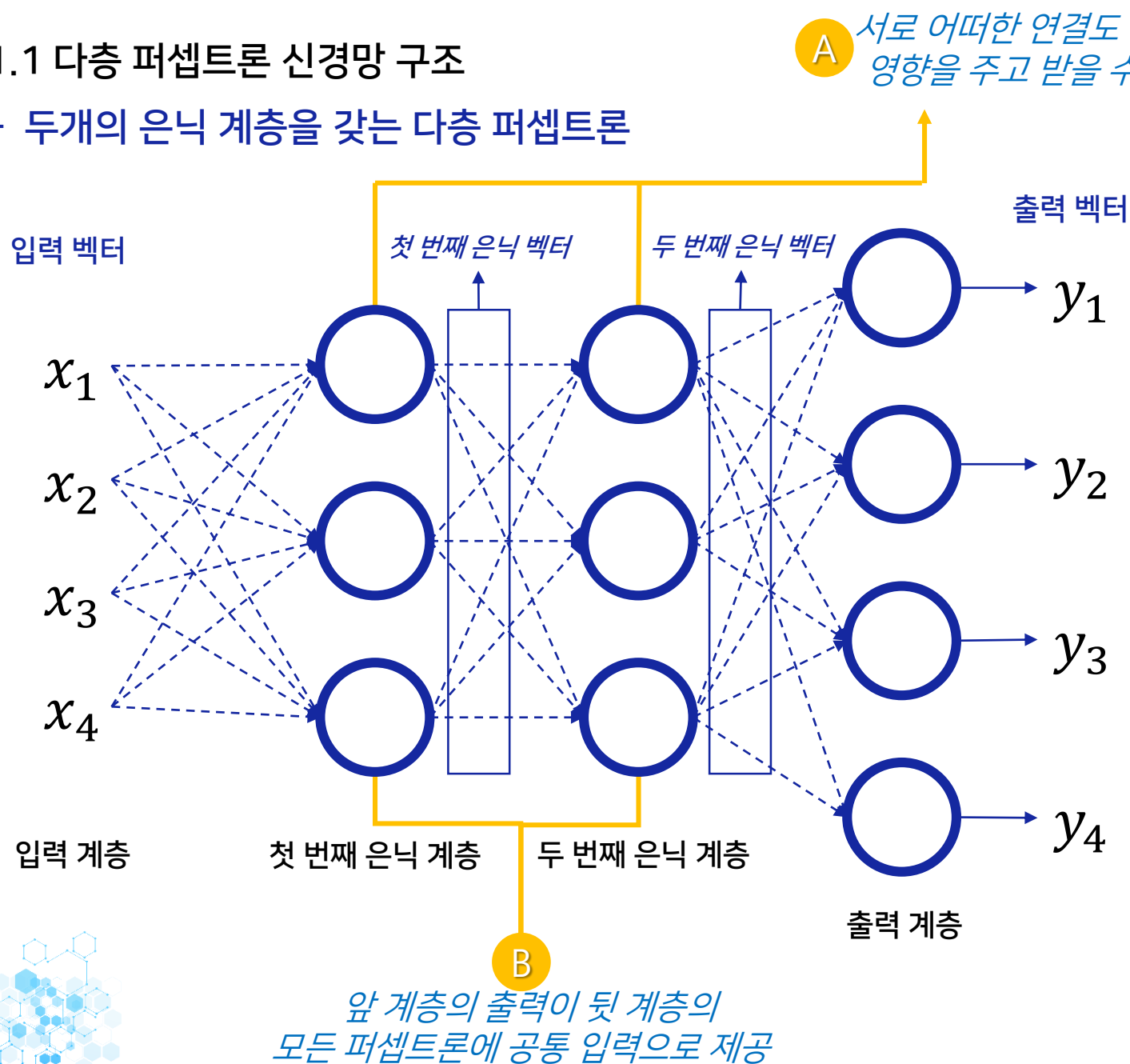
다층 퍼셉트론의 인접 계층끼리는 방향성을 갖는
'완전 연결 방식'으로 연결.

1장. 다층 퍼셉트론의 기본 구조

Part 2. 다층 퍼셉트론

1.1 다층 퍼셉트론 신경망 구조

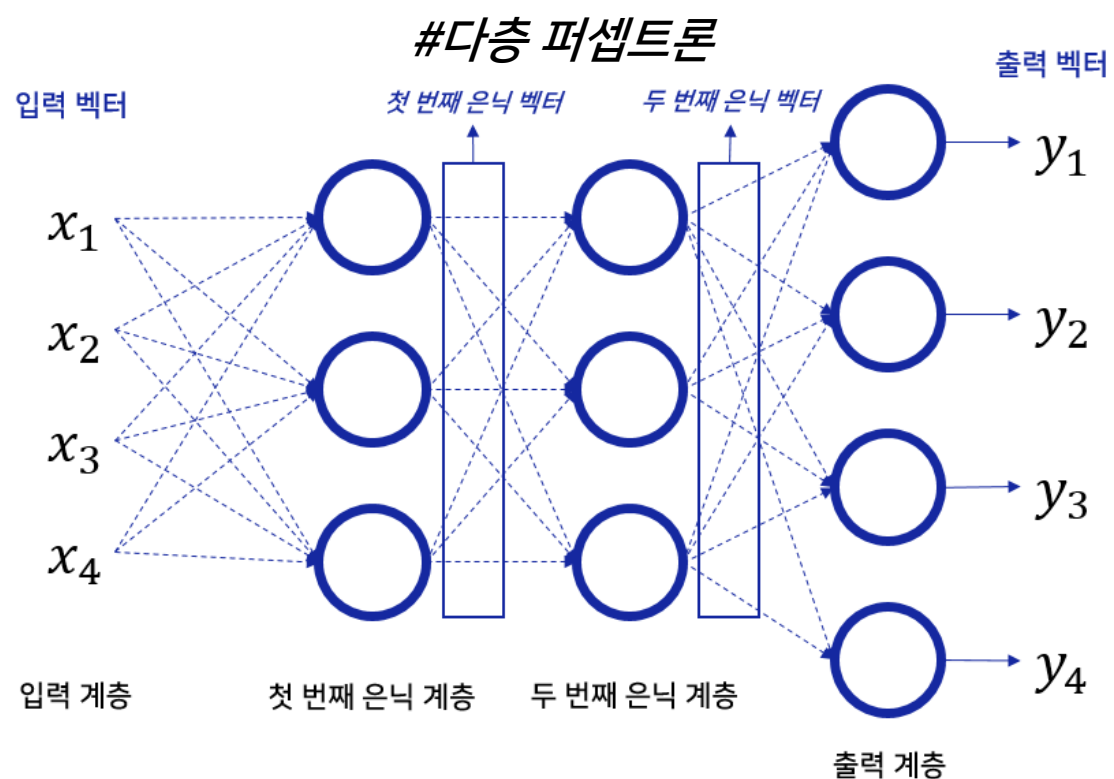
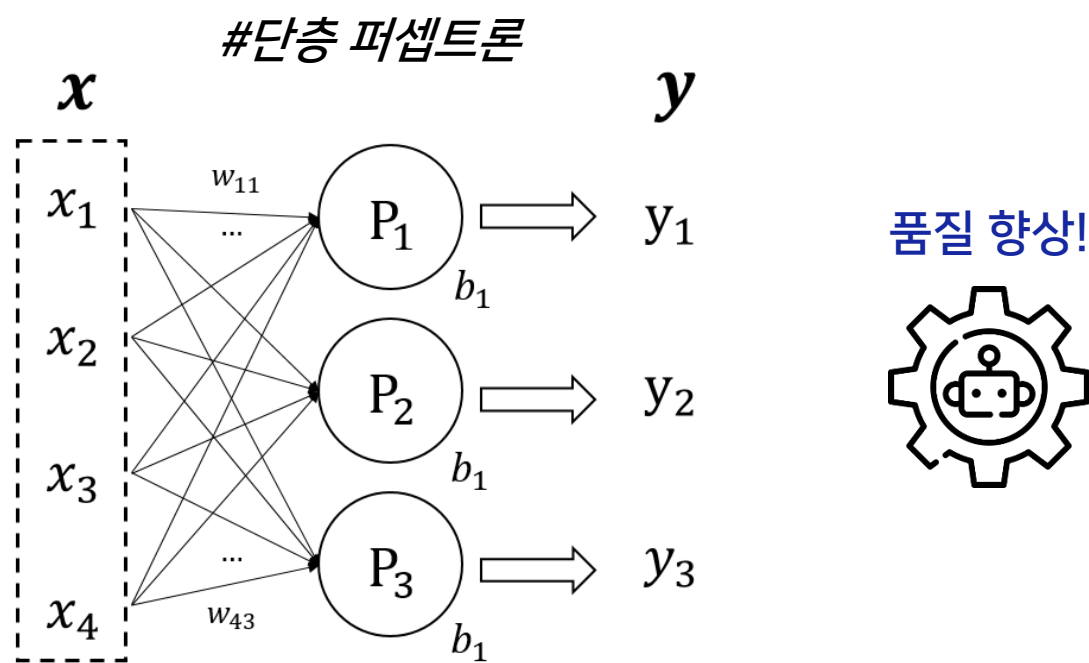
- 두개의 은닉 계층을 갖는 다층 퍼셉트론





Part 2. 다층 퍼셉트론

1.1 다층 퍼셉트론 신경망 구조



다층 퍼셉트론은

은닉계층을 **하나 이상** 갖는것이 보통.

'단층 퍼셉트론' 구조에 비해 **더 많은 퍼셉트론을 이용하여**,

'기억 용량'이나 '계산 능력'에 대한 **부담**이 커지는 대신,

'신경망의 **품질향상**'을 기대해 볼 수 있습니다.

Part 2. 다층 퍼셉트론

1.2 은닉 계층의 수와 폭

한 계층의 퍼셉트론들은 각각 해당 계층에 대한
입력 벡터 크기만큼의 가중치와 편향을 갖는다.



총 10개의 퍼셉트론



입력 m , 퍼셉트론 n

한 계층에 대한 파라미터 수

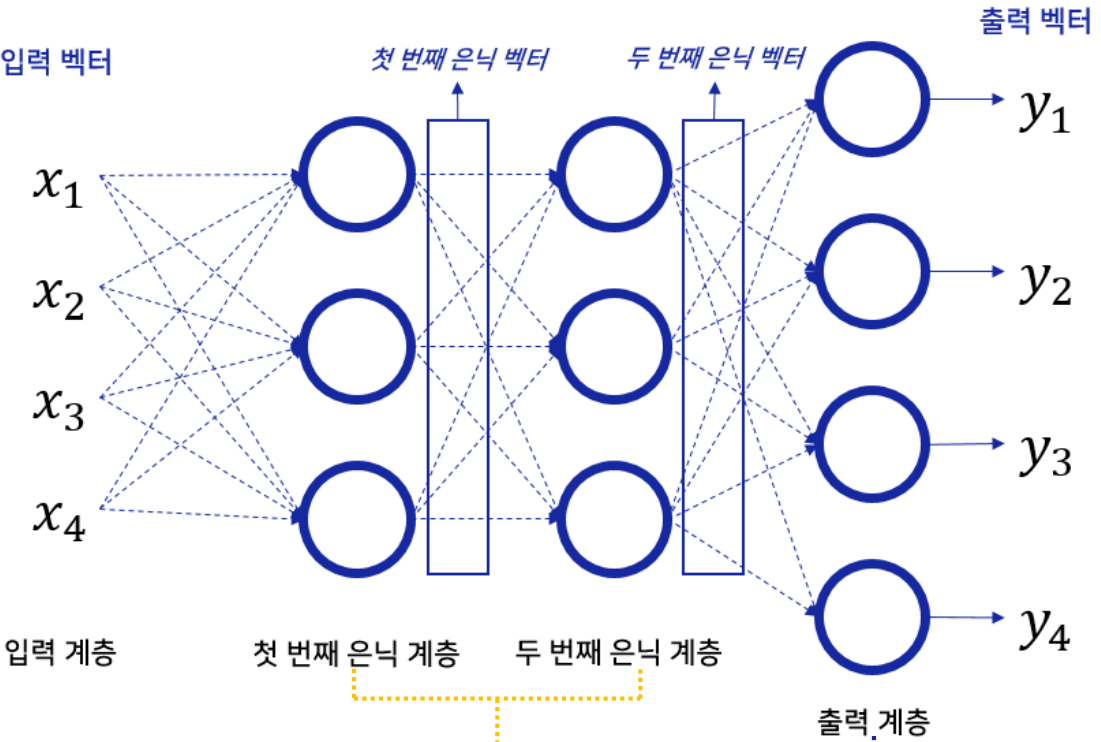
$$m * n + n$$

(입력 x 퍼셉트론 + 편향)

입력 벡터의 크기 : 4

2개의 은닉계층
3개의 퍼셉트론

출력 벡터의 크기 : 4



신경망에 주어진 원래의 임무에 따라 알맞은
형태의 출력벡터를 생성.

문제의 성격에 따라 출력 계층이 가질
퍼셉트론의 수가 달라진다.

은닉계층의 수와 각 은닉 계층의 폭은
신경망 설계자가 자유롭게 정할 수 있음.

은닉 계층의 폭
: 해당 계층이 갖는 퍼셉트론 수





1.2 은닉 계층의 수와 폭

“

은닉 계층의 수와 각 은닉 계층의 폭은 **신경망 품질을 결정짓는 중요한 요인**.

하지만 **무조건 은닉 계층 수나 폭을 늘린다고** 품질이 좋아지지 않음.

은닉계층을 추가해 파라미터 수가 늘어나게 되면 **더 많은 학습데이터가 필요**해지는 경향

→ 충분한 양의 **'양질의 데이터'**가 준비되지 않으면

다층 퍼셉트론 구조의 확장은 **오히려 품질을 떨어뜨리는 원인**이 됩니다.

”

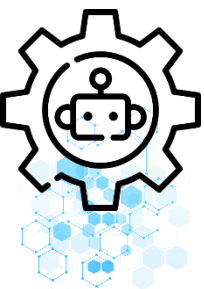
★은닉층 설정 꿀팁

문제의 규모, 데이터의 양, 난이도 종합적 고려!

다양한 **실험**과 축적된 **경험** 중요!

하이퍼파라미터 또한 중요한 원인!

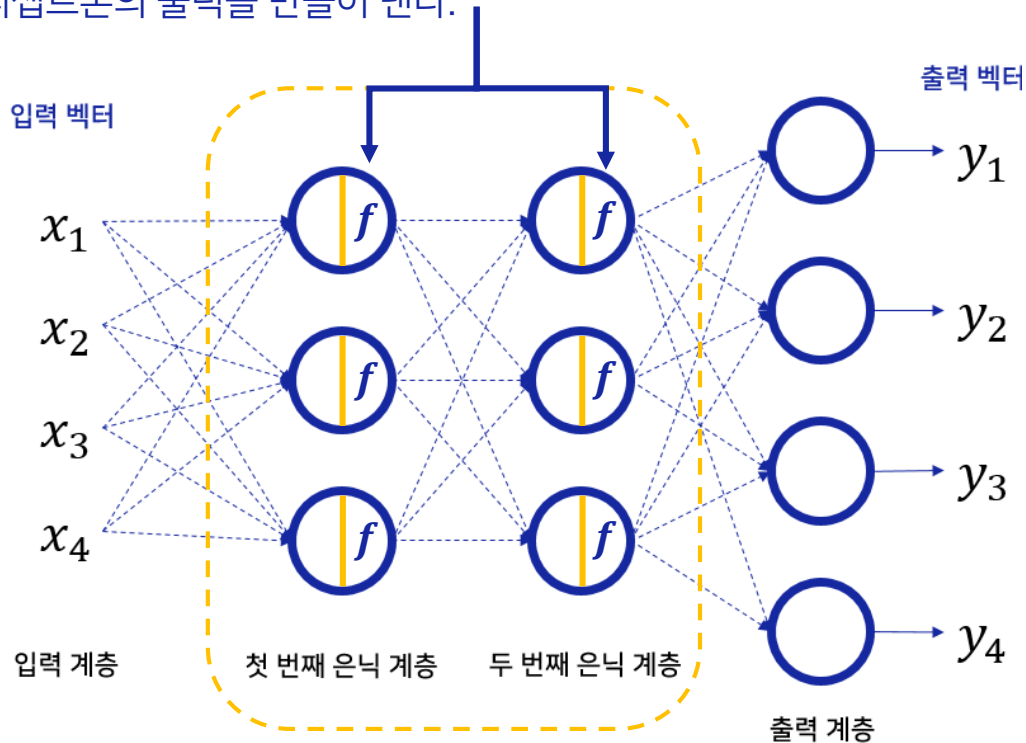
1장. 다층 퍼셉트론의 기본 구조



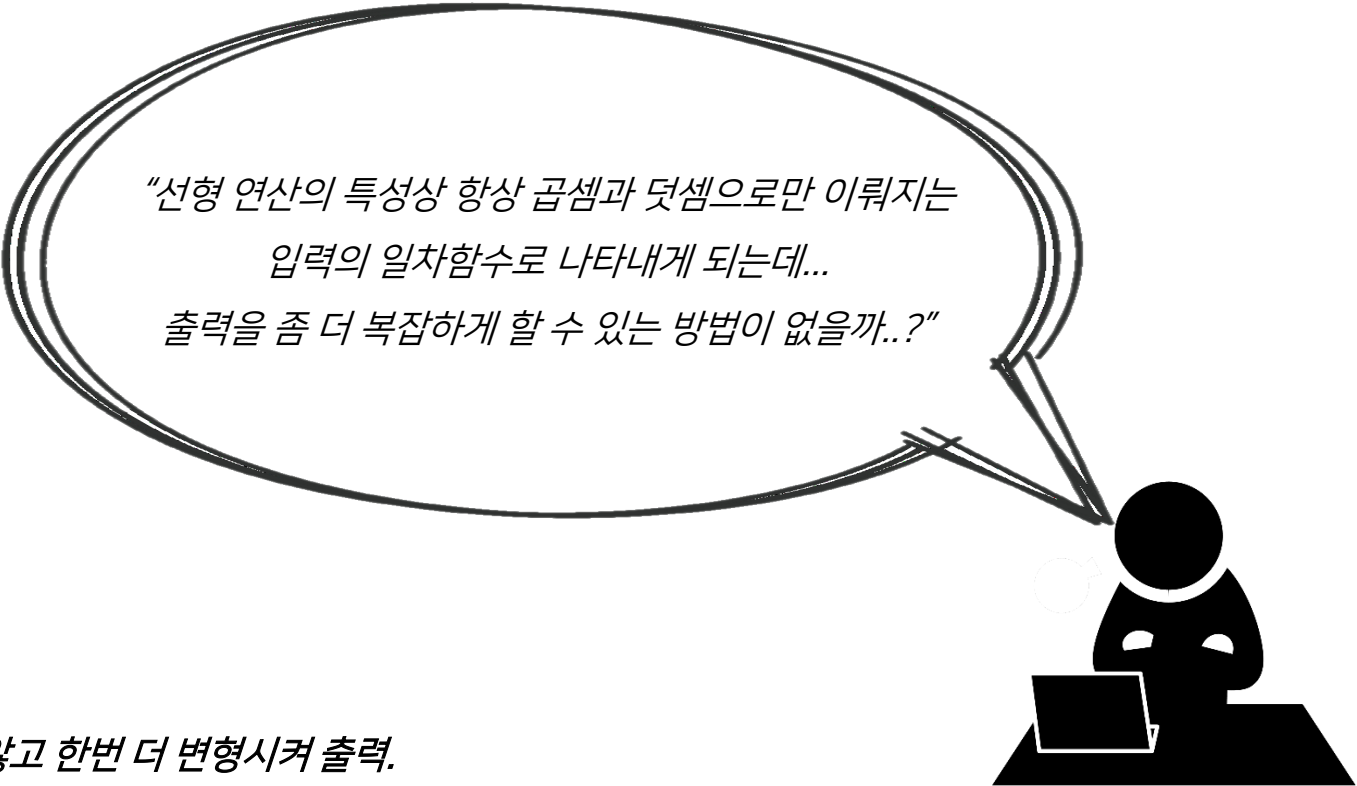
1.3 비선형 활성화 함수

비선형 활성화 함수 :
sigmoid(), softmax(), ...

- ※ 출력계층에는 '비선형 활성화 함수'가 사용되지 않는다.
- ※ 비선형 활성화 함수는 은닉계층에서 선형 연산결과를 변형시켜 퍼셉트론의 출력을 만들어 낸다.



퍼셉트론의 출력을 한번 더 변형시키려고 추가한 장치.
입력의 일차함수 표현을 넘어서는 다양하고 복잡한 형태의 퍼셉트론
출력을 만들 수 가 있게 됩니다.



- ※ 가중치와 편향을 이용해, 계산된 선형 연산 결과를 바로 출력으로 보내지 않고 한번 더 변형시켜 출력.
- ※ 은닉계층 다음에는 또 다른 은닉계층이나 출력계층이 있기 때문에, 다층 퍼셉트론에서는 계층과 계층사이, 즉 각 계층의 선형연산 사이 마다, 비선형 활성화 함수가 놓이게 됨.

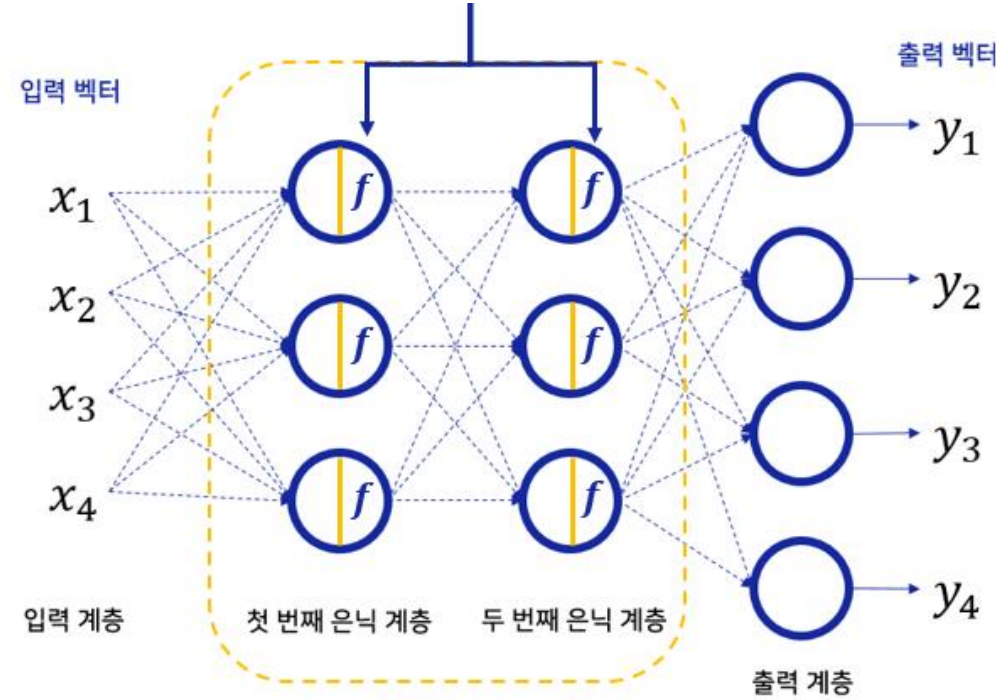


Part 2. 다층 퍼셉트론

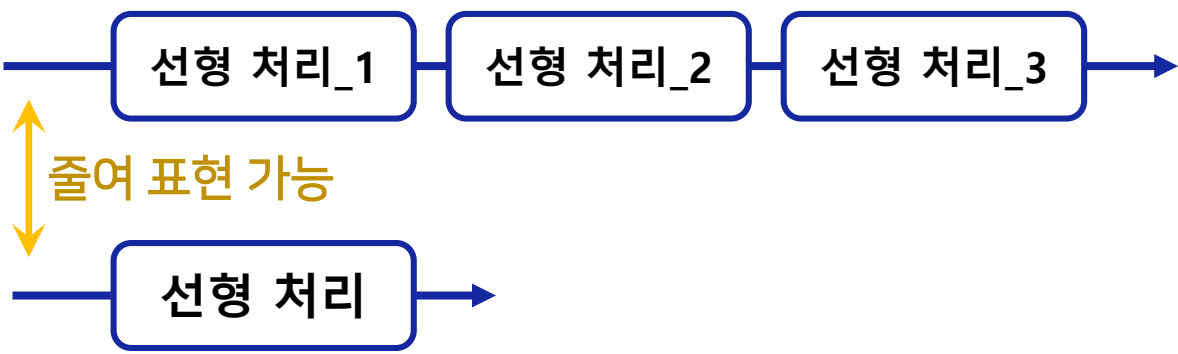
1.3 비선형 활성화 함수

다층 퍼셉트론에서 필수적인 구성 요소

활성화 함수의 도입으로 선형성의 한계에서 벗어날 수 있다.



선형처리의 수학적 원리



단 두 계층의 다층 퍼셉트론 구조만으로도,
어떠한 수학적 함수든 원하는 오차 수준 이내로 근사하게 동작!



Part 2. 다층 퍼셉트론

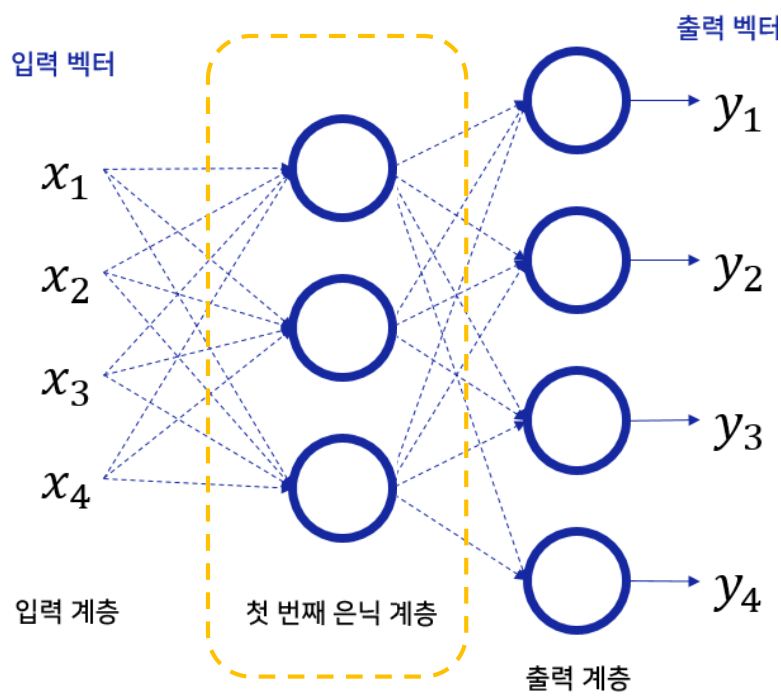
1.3 비선형 활성화 함수

『노드수가 많은』 『단층 구조 신경망』 보다

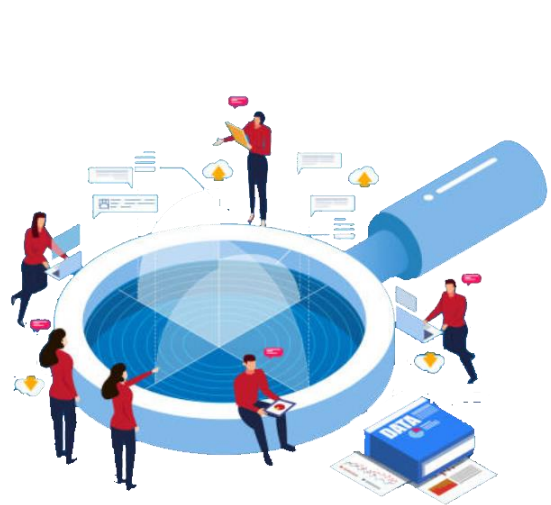
『노드수가 적은』 『다층 구조 신경망』 성능이 훨씬 우수한 경우가 많다!

복잡한 문제라고 가정!

- 퍼셉트론 수가 기하급수적으로 늘어날 수 있다.



최적의 파라미터 조합을 찾는 방법을 찾자!
(아담 알고리즘, 경사하강법,...)



※ 계층 수는 많아도 노드가 적은 신경망으로 문제를 해결하자!
(딥러닝의 급격한 성장의 시작)





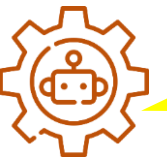
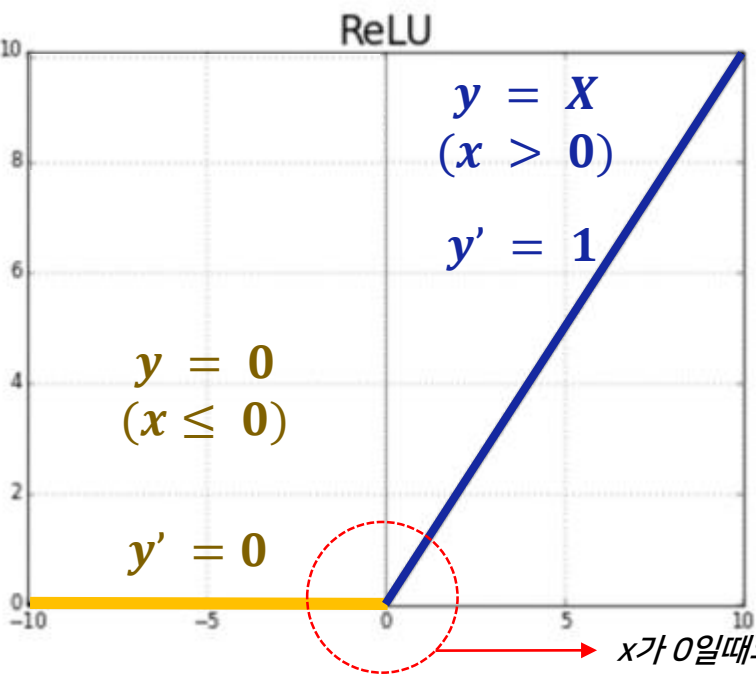
Part 2. 다층 퍼셉트론

1.4 ReLU 함수

BEST
은닉계층 비선형 활성화 함수 ReLU!
(Rectified Linear Unit)

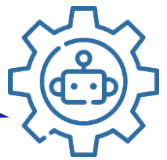
지수연산 포함 → 처리부담 증가

↔ softmax()
sigmoid()

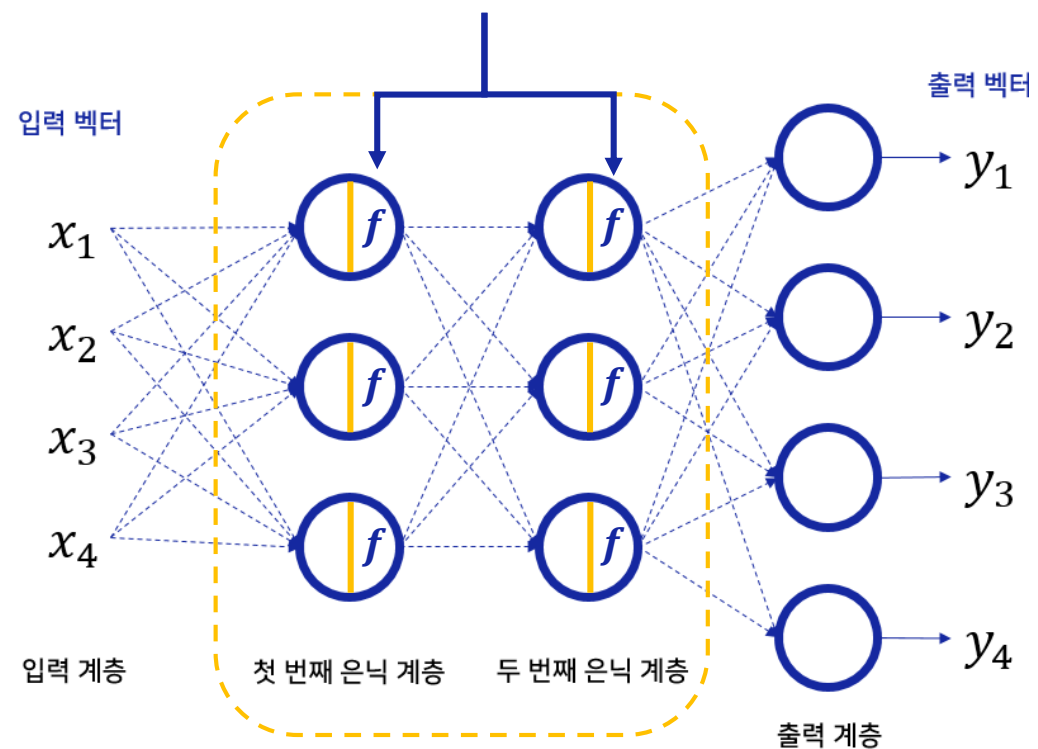


$x = 0$ 미분 불가!

미분 불가능한 지점에서는
미분값을 강제로 정해주고 넘어가도 괜찮아!



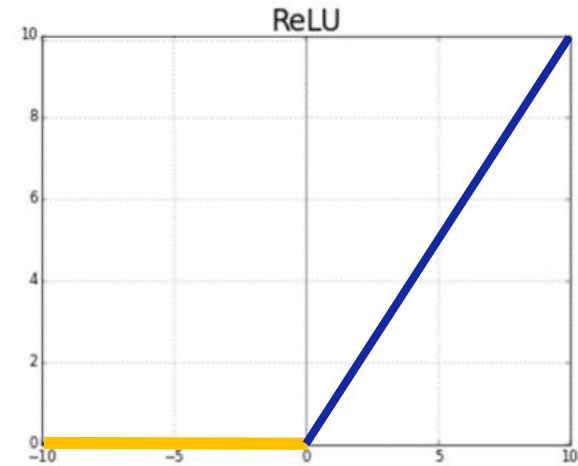
쉽고 빠른 ReLU !!





Part 2. 다층 퍼셉트론

1.4 ReLU 함수



ReLU 함수의 미분
y값으로 더 쉽게 계산가능



$$\begin{array}{ll} (y > 0) & y' = 1 \\ \text{otherwise} & y' = 0 \end{array}$$

※ ReLU의 출력(y)이 음수일 수 없기 때문에
y가 음수인 경우를 제외할 수 있어서 가능



np.sign()

```
print(np.sign(123))  
print(np.sign(456))  
print(np.sign(0))  
#print(np.sign(-789))
```

```
1  
1  
0
```

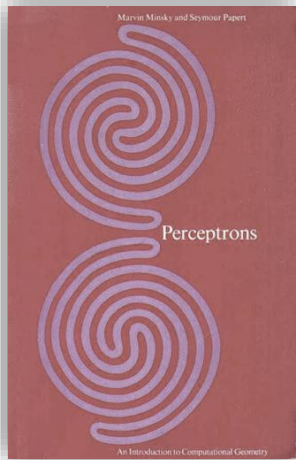
※ np.sign() 사용하게 되면, ReLU의 미분값을
쉽게 구할 수 있게 된다.





Part 2. 다층 퍼셉트론

1.5 민스키의 XOR 문제와 비선형 활성화 함수

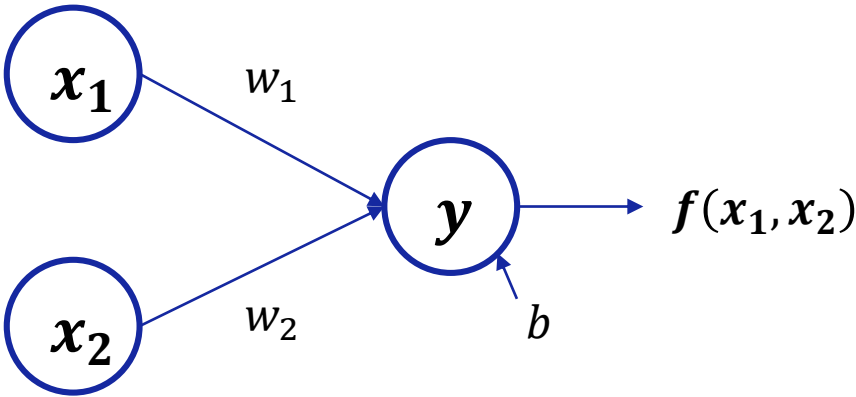


Marvin Lee Minsky
(1927년 8월 9일 ~ 2016년 1월 24일)

XOR 연산이 만족시켜야 하는 입출력 관계

$$y = f(x_1, x_2) = w_1x_1 + w_2x_2 + b$$

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0





1.5 민스키의 XOR 문제와 비선형 활성화 함수

XOR 연산이 만족시켜야 하는 입출력 관계

$$y = f(x_1, x_2) = w_1x_1 + w_2x_2 + b$$

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

$$x_2 = 0$$

$x_1 \uparrow \ y \uparrow \ \& \ x_1 \downarrow \ y \downarrow$

$$x_2 = 1$$

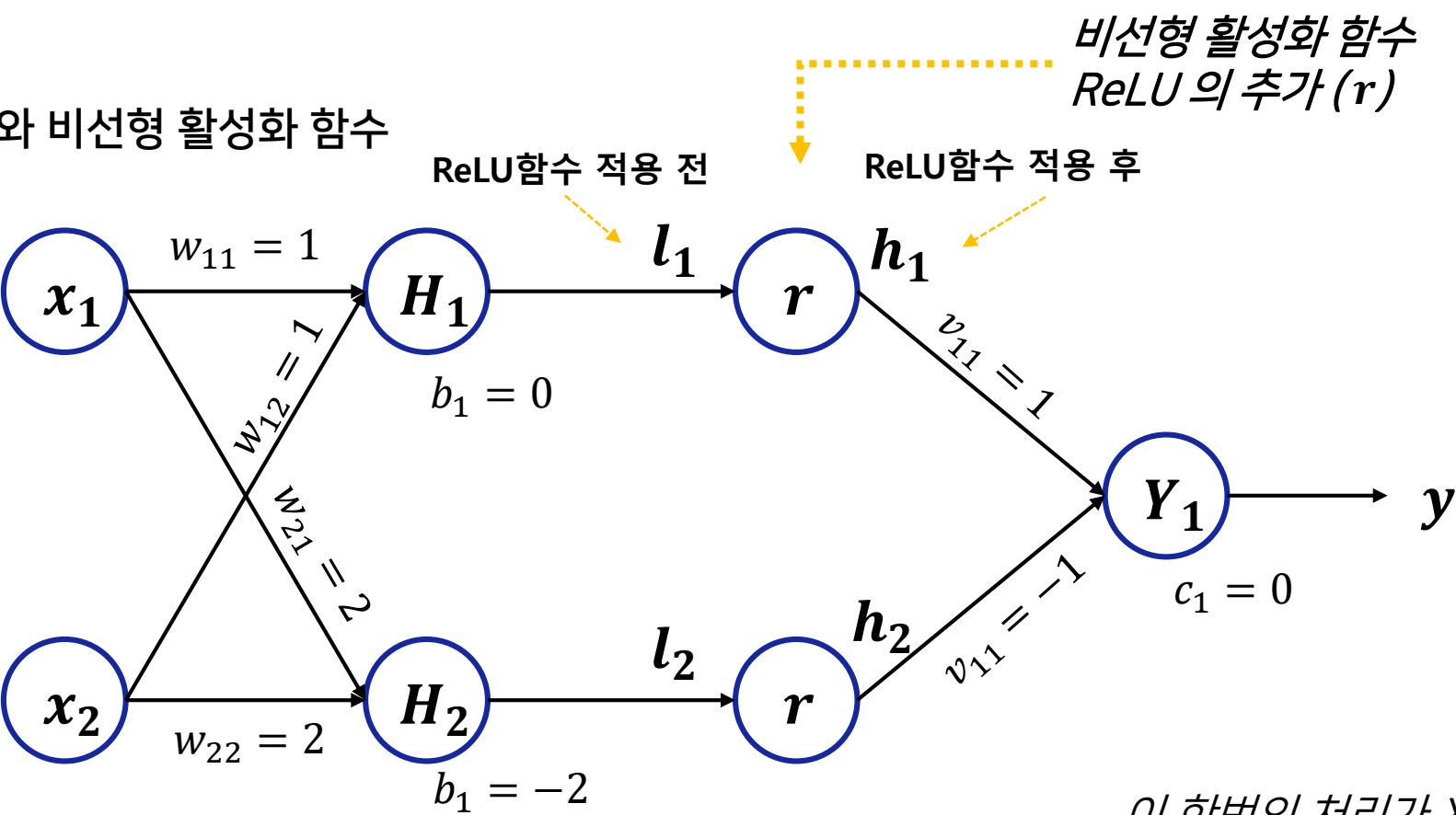
$x_1 \uparrow \ y \downarrow \ \& \ x_1 \downarrow \ y \uparrow$

*과연 x_2 값에 따라 x_1, y 가 이렇게 값이 바뀔 수 있도록
파라미터를 설정할 수 있는가?*





1.5 민스키의 XOR 문제와 비선형 활성화 함수



이 한번의 처리가 XOR연산을 완벽하게 처리할 수 있게 만듦

x_1	x_2	l_1	l_2	h_1	h_2	y
0	0	0	-2	0	0	0
0	1	1	0	1	0	1
1	0	1	0	1	0	1
1	1	2	2	2	2	0

