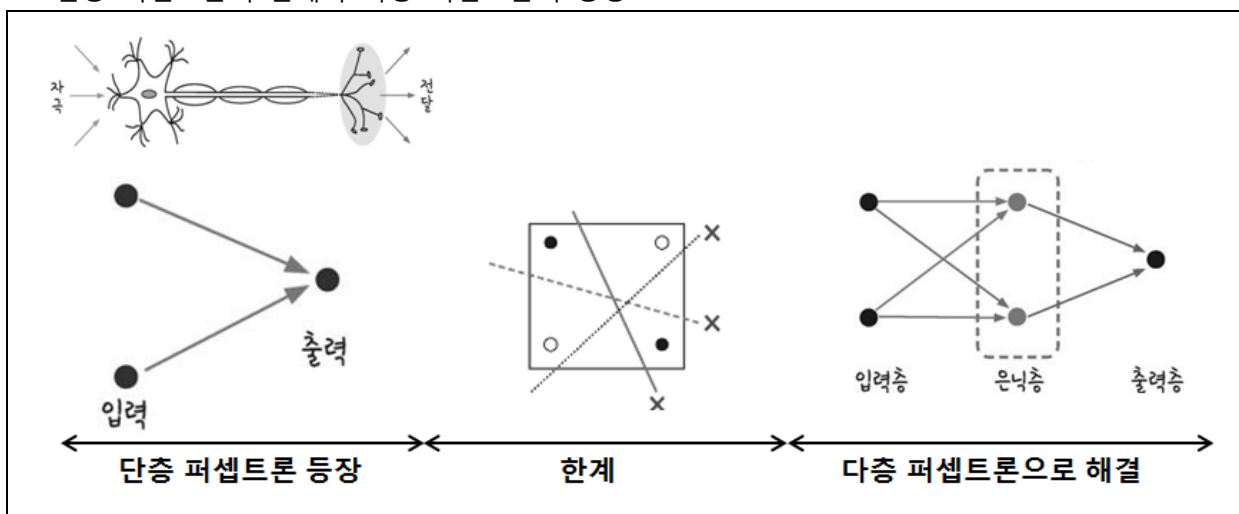


다음 문제를 풀어 제출하시오.

1. 단층퍼셉트론과 다층퍼셉트론에 대해 설명하시오.
2. 단층 퍼셉트론은 직선형 영역만 표현할 수 있다. 그러나 XOR 문제를 해결할 수 없다.
이러한 이유는 무엇이며, 이를 해결하기 위해서는 어떠한 방법이 있는가?

답)

1. 단층 퍼셉트론의 한계와 다층 퍼셉트론의 등장

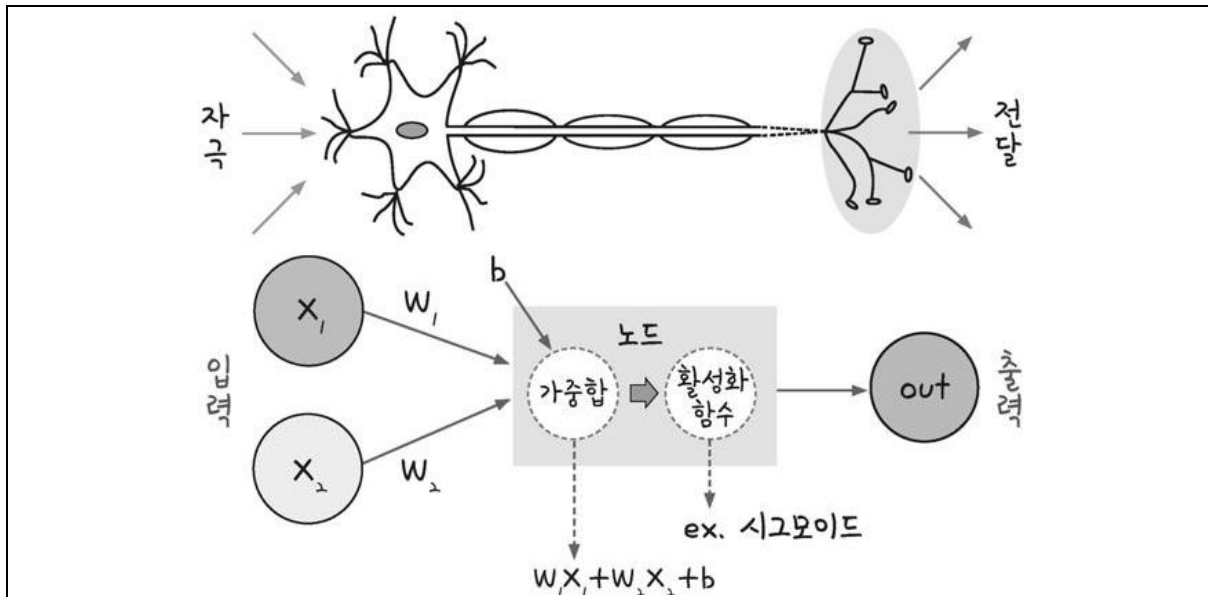


- 인간의 뉴런을 모방한 퍼셉트론 알고리즘으로 인공지능 구현이 가능할 것이라 예상했으나 XOR(Exclusive-OR) 문제조차 해결되지 않는 문제에 직면
- 이를 해결하기 위해 은닉층을 가지는 다층 퍼셉트론이 등장

2. 단층 퍼셉트론의 개념

가. 단층 퍼셉트론(perceptron)의 개념도

- 프랑크 로젠블라트가 1957년 고안한 신경망(딥러닝)의 기원이 되는 알고리즘



- 다수의 신호를 입력받아 하나의 신호(0/1)을 출력하는 알고리즘
- 자극을 받아 전위 변화를 일으킴으로써 임계값을 넘는 인간의 뉴런을 모방한 알고리즘/신경망의 단위

나. 단층 퍼셉트론의 구성요소

구성요소		세부내용
입력	Input	- 입력 노드, (단위) 데이터
바이어스	Bias	- 편향, 임계값(threshold)를 넘는데 영향을 주는 편향 정보
가중치	Weight	- 각 입력노드가 임계값에 영향을 주는 정도(영향력)
활성화 함수	Activate Function	- 출력의 형태를 조정하여 임계치를 넘을지 결정시키는 함수 - 임계값까지 입력과 가중치의 합계를 활성화 하는 함수
출력	Output	- '흐른다/안 흐른다', '켜진다/꺼진다'와 같이 이진으로 표현될 수 있는 출력 노드

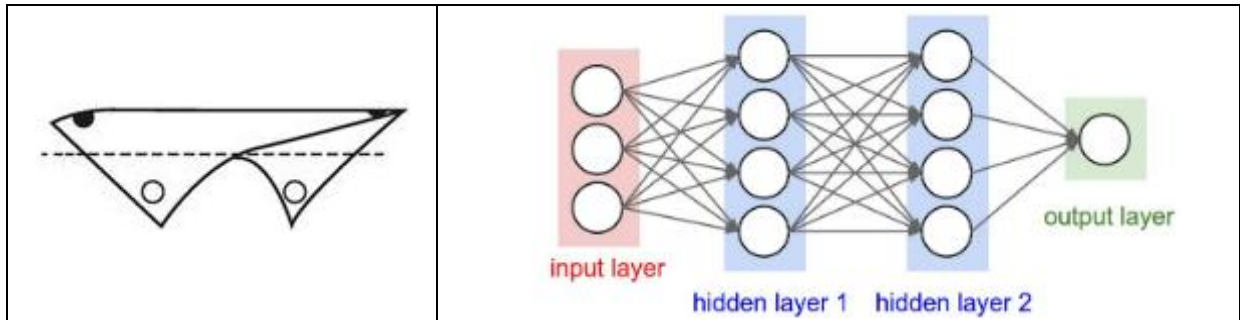
다. 진리표 구현으로 본 단층 퍼셉트론 알고리즘의 문제점

AND	OR	XOR

- 일차 방정식 형태를 가지는 Linear 한 단층 퍼셉트론으로 XOR 표현이 불가능(문제)

3. 다층 퍼셉트론의 개념

가. 다층 퍼셉트론의 개념도



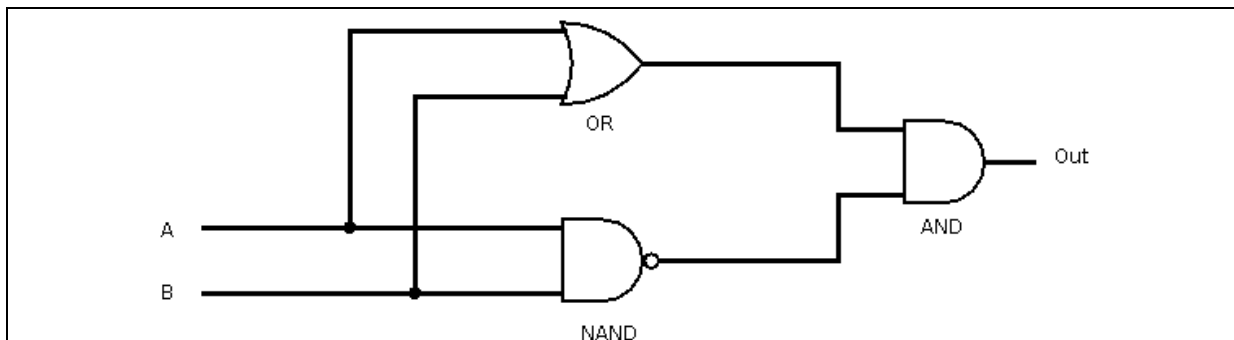
- 다층 퍼셉트론의 한계인 XOR 문제를 해결하기 위해 2차원 평면에 변화를 준 개념
- 2차원 평면에서는 Linear 하게 해결되지 않지만 3차원을 가정한다면 직선으로 XOR 문제를 해결 할 수 있으며, 은닉층을 둠으로 평면(차원)을 왜곡 시킬 수 있음(해결)

나. 다층 퍼셉트론의 구성요소

구성요소		세부내용
단층 퍼셉트론의 구성요소 +		
은닉층	Hidden Layer	<ul style="list-style-type: none"> - Input 노드와 Output 노드 사이에 은닉한 층으로 차원을 왜곡하는 결과를 가져오는 Layer - 은닉층에서 일어나는 일은 사람이 알 수 없는 Black Box

4. 논리게이트와 파이썬 소스 코드로 표현하는 다층 퍼셉트론

가. 논리게이트로 표현하는 다층 퍼셉트론



출처: <https://sullystationtechnologies.com/npnxorgate.html>

- OR, NAND, AND 를 이용하여 XOR 구현이 가능(단층 대비 Layer 증가)

X1	X2	OR
0	0	0
0	1	1
1	0	1
1	0	1

X1	X2	NAND
0	0	1
0	1	1
1	0	1
1	0	0

OR	NAND	AND
0	1	0
1	1	1
1	1	1
1	0	0

XOR
0
1
1
0

- OR, NAND 층을 추가하여 단층 퍼셉트론은 해결 못한 XOR 결과를 도출할 수 있음

나. 파이썬 소스코드로 표현하는 다층 퍼셉트론

- 4-가. 논리 게이트를 파이썬으로 작성
- AND, OR, NAND 를 먼저 정의

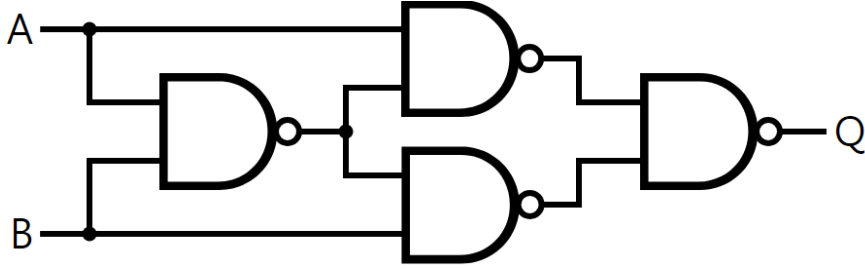
AND		OR		NAND	
<pre>import numpy as np def AND(x1, x2): x = np.array([x1, x2]) w = np.array([0.5, 0.5]) b = -0.7 tmp = np.sum(w * x) + b if tmp <= 0: return 0 else: return 1</pre>		<pre>def OR(x1, x2): x = np.array([x1, x2]) w = np.array([0.5, 0.5]) b = -0.3 tmp = np.sum(w * x) + b if tmp <= 0: return 0 else: return 1</pre>		<pre>def NAND(x1, x2): x = np.array([x1, x2]) w = np.array([-0.5, -0.5]) b = 0.7 tmp = np.sum(w * x) + b if tmp <= 0: return 0 else: return 1</pre>	
<pre>print('AND') print(AND(0, 0)) print(AND(1, 0)) print(AND(0, 1)) print(AND(1, 1))</pre>	AND 0 0 0 1	<pre>print('OR') print(OR(0, 0)) print(OR(1, 0)) print(OR(0, 1)) print(OR(1, 1))</pre>	OR 0 1 1 1	<pre>print('NAND') print(NAND(0, 0)) print(NAND(1, 0)) print(NAND(0, 1)) print(NAND(1, 1))</pre>	NAND 1 1 1 0

XOR		
<pre>def XOR(x1, x2): # layer1 s1_1 = OR(x1, x2) s1_2 = NAND(x1, x2) # layer2 s2 = AND(s1_1, s1_2) return s2</pre>	<pre>print('XOR') print(XOR(0, 0)) print(XOR(1, 0)) print(XOR(0, 1)) print(XOR(1, 1))</pre>	XOR 0 1 1 0

- 첫번째 레이어에서 OR, NAND를 수행하고 두번째 레이어에서 AND 를 수행한 결과
- XOR 출력을 얻을 수 있음

다. 계층을 추가하여 구현한 XOR

- 4-가/나 의 형태가 아니라도 XOR 구현이 가능함
- 계층을 추가하고 NAND 만을 이용하여 XOR 구현이 가능함



출처: https://ko.wikipedia.org/wiki/%ED%8C%8C%EC%9D%BC:XOR_from_NAND.svg

X1	X2	NAND
0	0	1
0	1	1
1	0	1
1	1	0

X1	NAND	NAND
0	1	1
0	1	1
1	1	0
1	0	1

NAND	NAND	NAND
1	1	0
1	0	1
0	1	1
1	1	0

XOR
0
1
1
0

X2	NAND	NAND
0	1	1
1	1	0
0	1	1
0	0	1

```
def XOR(x1, x2):
    # layer1
    s1 = NAND(x1, x2)
    # layer2
    s2_1 = NAND(x1, s1)
    s2_2 = NAND(x2, s1)
    # layer3
    return NAND(s2_1, s2_2)
```

```
print('XOR')
print(XOR(0, 0))
print(XOR(1, 0))
print(XOR(0, 1))
print(XOR(1, 1))
```

XOR

0

1

1

0

⇒ **결론:** 인간의 뉴런을 모방한 단층 퍼셉트론 모델이 가지는 직선형 성질(일차방정식 형태, Linear)의 한계로 XOR 조차 표현이 안 되는 문제가 존재했고, 이를 극복하기 위한 모델로 은닉층을 추가한 다층 퍼셉트론이 등장함