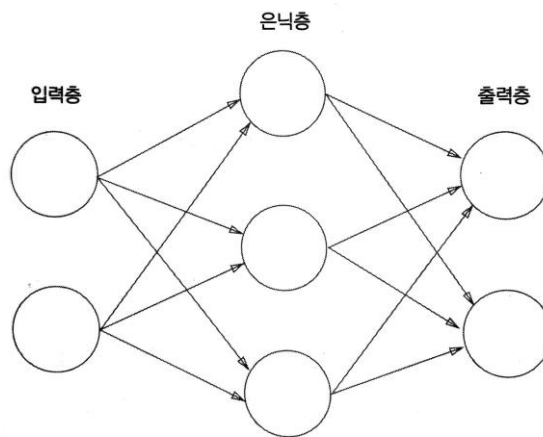


## Chapter3. 신경망

### 1. 퍼셉트론과 신경망의 차이

퍼셉트론	신경망
가중치를 사람이 직접 찾아 설정	가중치 매개변수의 값을 데이터로부터 자동 학습

### 2. 신경망의 예 (구조)



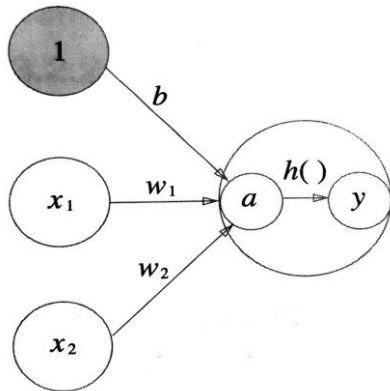
- 입력층 → 출력층 순서로 0층, 1층, 2층
- 위 신경망은 가중치를 갖는 층이 2개층으로 '2층 신경망'이라고 한다

### 3. 활성화 함수

$$y = \begin{cases} 0 & (b + w_1x_1 + w_2x_2 \leq 0) \\ 1 & (b + w_1x_1 + w_2x_2 > 0) \end{cases}$$

↓

- $y = h(x) = h(b + w_1x_1 + w_2x_2)$
- $h(x) = \begin{cases} 0 & (x \leq 0) \\ 1 & (x > 0) \end{cases}$
- 위에서  $h(x)$  = 활성화 함수 (입력 신호의 총 합이 활성화를 일으키는지 정한다)

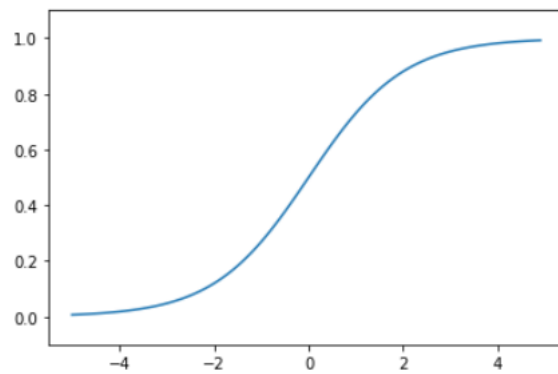


- 신경망에서 활성화 함수는 비선형 함수를 사용해야 한다.

### 3.1 시그모이드 함수 (sigmoid)

$$h(x) = \frac{1}{1 + \exp(-x)}$$

```
In [9]: import matplotlib.pyplot as plt
x = np.arange(-5.0, 5.0, 0.1)
y = sigmoid(x)
plt.plot(x,y)
plt.ylim(-0.1, 1.1)
plt.show()
```



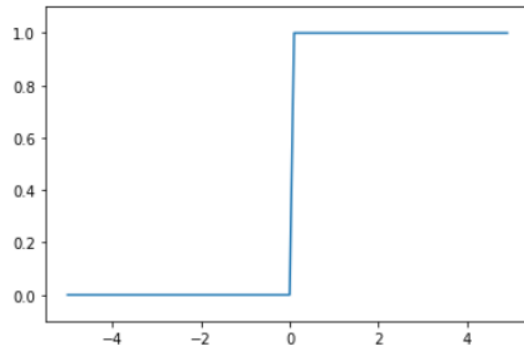
### 3.2 계단 함수 (step)

$$h(x) = \begin{cases} 0 & (x \leq 0) \\ 1 & (x > 0) \end{cases}$$

```
In [16]: import matplotlib.pyplot as plt

def step_function(x):
    return np.array(x>0, dtype=np.int)

x = np.arange(-5.0, 5.0, 0.1)
y = step_function(x)
plt.plot(x,y)
plt.ylim(-0.1,1.1) #y축 범위 지정
plt.show()
```



- 시그모이드 함수는 연속적, 계단함수는 불연속적
- 두 함수 모두 0 ~ 1사이의 값을 가진다

### 3.3 ReLu 함수

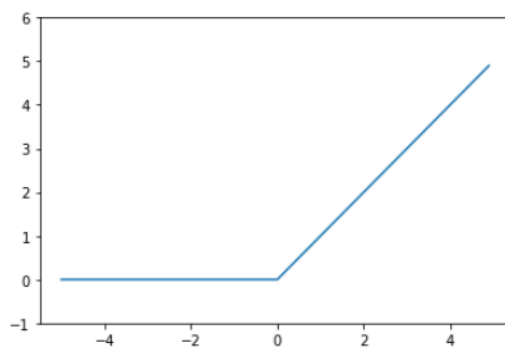
$$h(x) = \begin{cases} x & (x > 0) \\ 0 & (x \leq 0) \end{cases}$$

- 입력이 0을 넘으면 입력 그대로 출력, 0 이하면 0출력

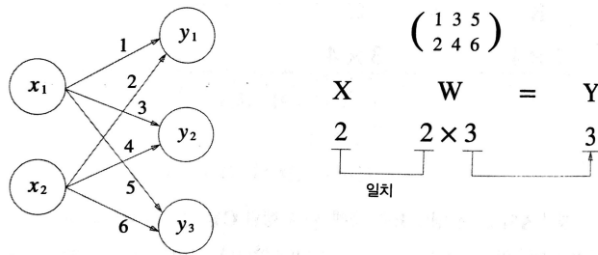
```
In [3]: import numpy as np
import matplotlib.pyplot as plt

def relu(x):
    return np.maximum(0,x)

x = np.arange(-5, 5, 0.1)
y = relu(x)
plt.plot(x,y)
plt.ylim(-1,6)
plt.show()
```



#### 4. 다차원 배열 (신경망의 행렬 곱)



\* 편향과 활성화 함수를 생략하고 가중치만 갖는 신경망

```
In [36]: X = np.array([1,2])  
X.shape
```

```
Out[36]: (2,)
```

```
In [38]: W = np.array([[1,3,5],[2,4,6]])  
print(W)  
W.shape
```

```
[[1 3 5]  
 [2 4 6]]
```

```
Out[38]: (2, 3)
```

```
In [39]: Y = np.dot(X,W)  
print(Y)
```

```
[ 5 11 17]
```