

# Ocaml

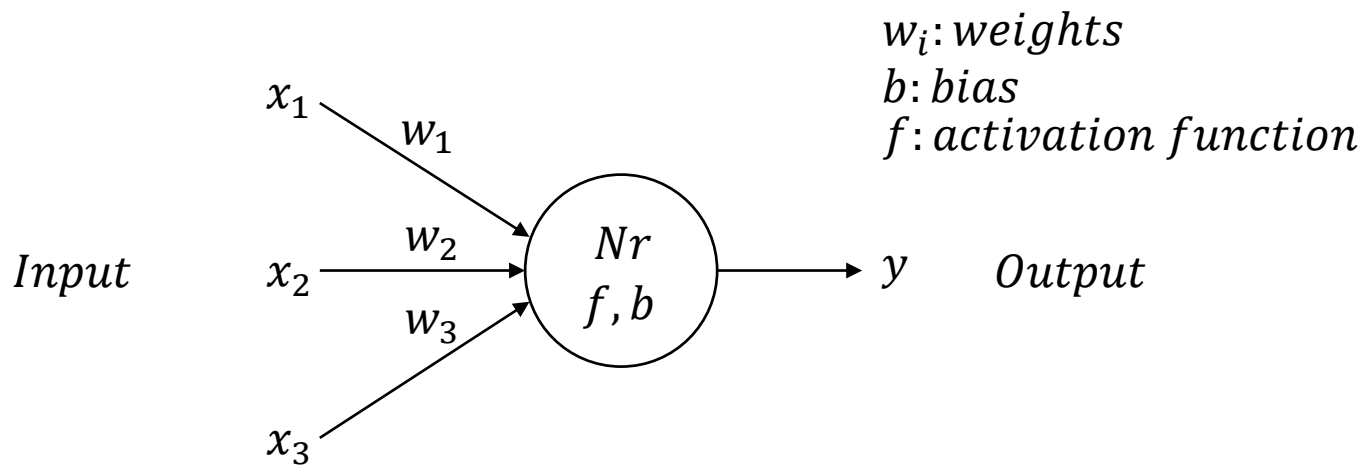
---

KOSMOS

# Neural Network가 뭐..쥬..?

기계학습(Machine Learning)의 한 분야로 매우 간단한 기능을 하는 뉴런들이 연결되어 값을 도출해내는 프로그램입니다.

뉴런은 아래처럼 동작해요.

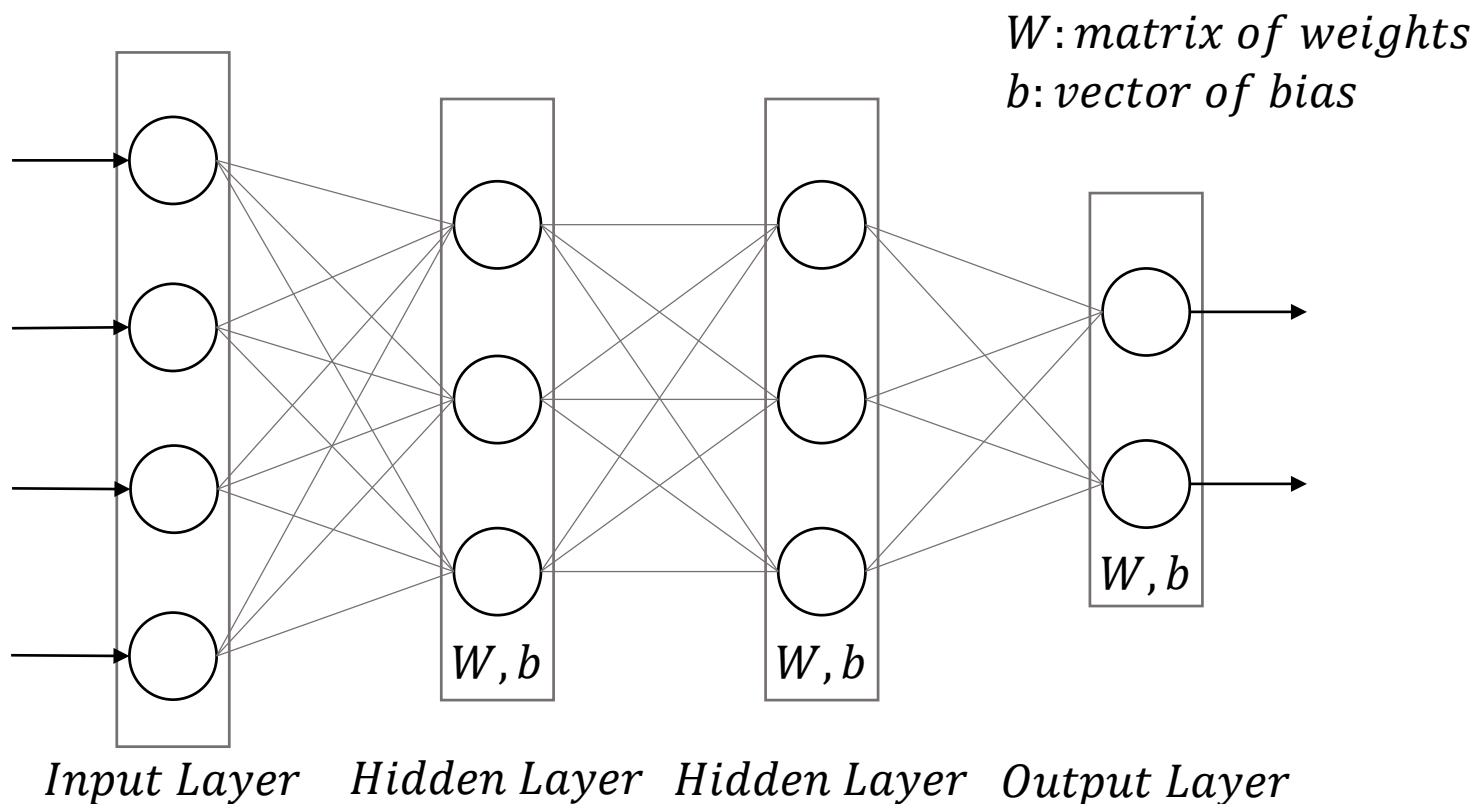


$$y = f(w_1x_1 + w_2x_2 + w_3x_3 + b)$$

# Fully Connected Neural Network

FCNN은 레이어 안에 있는 모든 뉴런들이 다음 레이어의 모든 뉴런들로 이어져 있는 NN이에요.

그림으로 나타내면 아래처럼 나타낼 수 있어요.



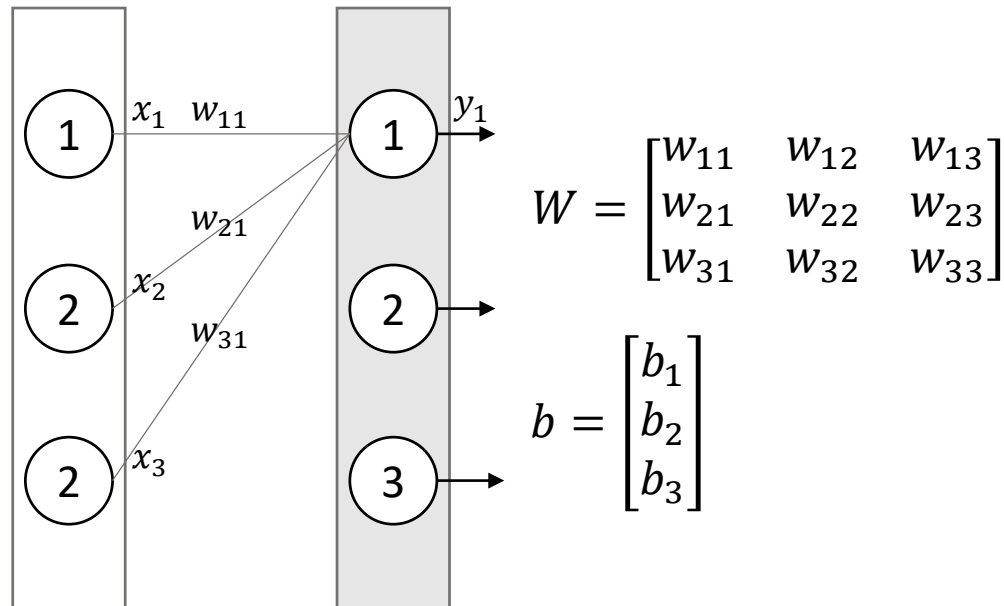
# 어떻게 계산하는 거죠?

앞서 공부한대로 ( $y_1 = \sum_i w_{i1}x_i + b_1$ )과 같이 표현 할 수 있어요.

위 식을 일반화 하면 ( $y_j = \sum_i w_{ij}x_i + b_j$ )와 같이 표현 할 수 있어요.

$x$ 를  $x_i$ 들로 이루어진 벡터라고 하고,  $y$ 를  $y_j$ 들로 이루어진 벡터라고 하면,  $y$ 는 아래식으로 계산 할 수 있어요.

$$y = W^T x + b$$



# 아직 안 쓴 게 하나 있어요

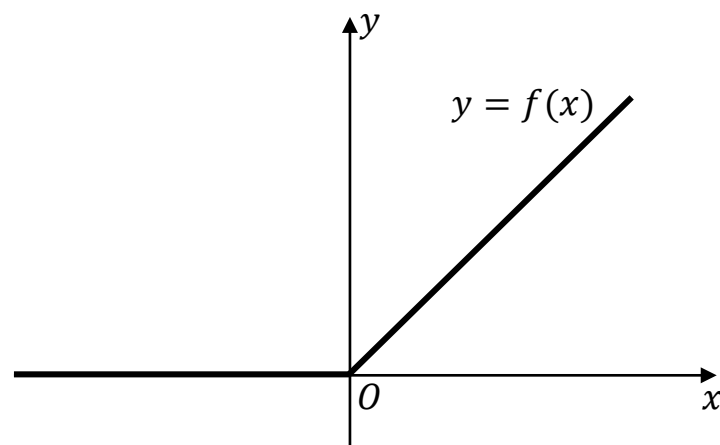
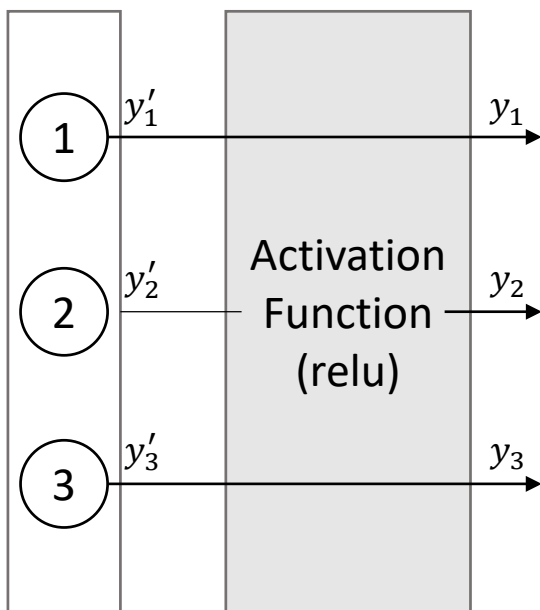
이렇게 해서 계산한 값에 활성화 함수를 사용하면 끝나요!

가장 많이 사용하는 활성화 함수인 relu는 아래의 식으로 표현할 수 있어요.

$$f(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases}$$

아래와 은 식으로 이 활성화 함수를 이용하면 최종적으로 Layer의 출력 값을 구할 수 있어요.

$$y = f(y')$$



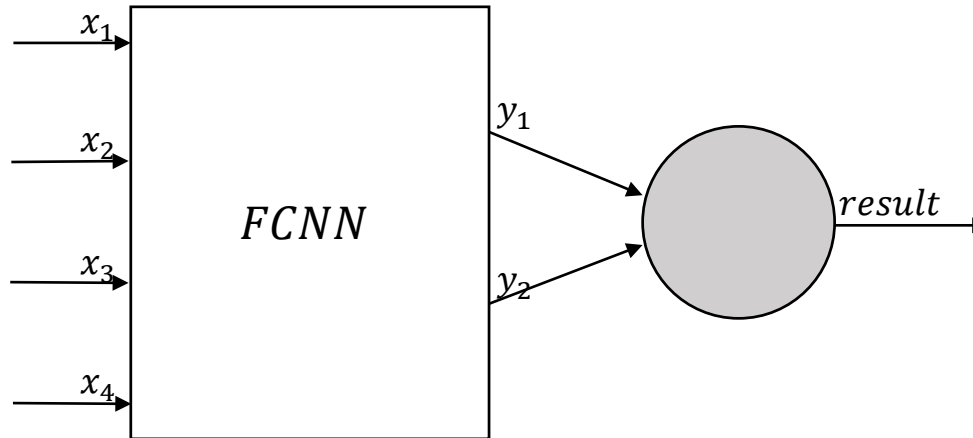
# 이제 고르기만 하면 돼요!

---

FCNN의 내부는 앞서 공부한 것처럼 계산하면 돼요.

이제 출력을 처리하기만 하면 되는데 제일 쉬운 부분이에요!

아래의 그림처럼 출력된 벡터의 원소 중에서 가장 큰 값의 인덱스를 계산하기만 하면 돼요.



식으로는 아래와 같이 쓸 수 있어요.

$$result = \arg \max_i y_i$$

# Ocaml로 간단한 계산기 만들기

---

Ocaml을 이용해서 간단한 계산기를 만들어 볼 거예요!

일단 이번 주는 아래의 가장 간단한 기능 4가지부터 만들어 볼 게요.

$+, -, *, /$

우선 계산 할 식( $E$ )이 어떻게 생겼는 지부터 정의해 볼 게요.

$$\begin{aligned} F &\rightarrow E \\ E &\rightarrow n \\ &\quad | E + E \\ &\quad | E - E \\ &\quad | E * E \\ &\quad | E / E \end{aligned}$$

# Ocaml로 간단한 계산기 만들기

---

앞서 정의한 식을 계산하면 어떤 값이 나올 수 있는지 정의해 볼 게요.

$$Val = R$$

앞서 정의한 식을 어떻게 계산하는 지 정리해 볼 게요.

$$\frac{}{n \Rightarrow n} \qquad \frac{E_1 \Rightarrow n_1 \quad E_2 \Rightarrow n_2}{E_1 + E_2 \Rightarrow n_1 + n_2} \qquad \frac{E_1 \Rightarrow n_1 \quad E_2 \Rightarrow n_2}{E_1 - E_2 \Rightarrow n_1 - n_2}$$

$$\frac{E_1 \Rightarrow n_1 \quad E_2 \Rightarrow n_2}{E_1 * E_2 \Rightarrow n_1 * n_2} \qquad \frac{E_1 \Rightarrow n_1 \quad E_2 \Rightarrow n_2}{E_1 / E_2 \Rightarrow n_1 / n_2} n_2 \neq 0$$



# Ocaml로 간단한 계산기 만들기

---

앞에서 정의 한 식들을 코드로 옮기기만 하면 계산기가 완성돼요!