

Activation function (활성화 함수)

=> input을 받아 $\begin{cases} \text{활성화 시} \\ \text{비활성화 함수를} \end{cases}$ 결정할 때 쓰는 함수.

input에 따라 output은 $\begin{cases} \text{선형} \\ \text{비선형} \end{cases}$ 일 수 있다.

선형 함수는 Linear (직선 모양이고)

비선형 // 곡선이나 1개 이상 직선의 모양이고 (Relu, sigmoid) 등이 존재

∴ 문제의 종류에 따라 사용할 Activation function을

선택해야 한다.

① 직선 경향 → 선형 함수

② 1개 (0/X 분포) → Vanishing Gradient 때문에
sigmoid는 잘 쓰지 않고
Relu보다 큰 2변형을 사용.

③ 3개 이상 분포 시
→ Softmax를 주로 사용.

→ 꼭 반드시 하는 게 아니고 목표 accuracy와 learning time을 고려
'통합' 선택도 가능.

✗ Neural Network에서는

비선형 함수를 Activation function으로 사용해야 한다.

서로 다른



hidden layer를 multi layer로
구성하고 활성화 함수를 선형으로

$$h(x) = cx \text{ or } \\ h(h(h(x))) = c^3x$$



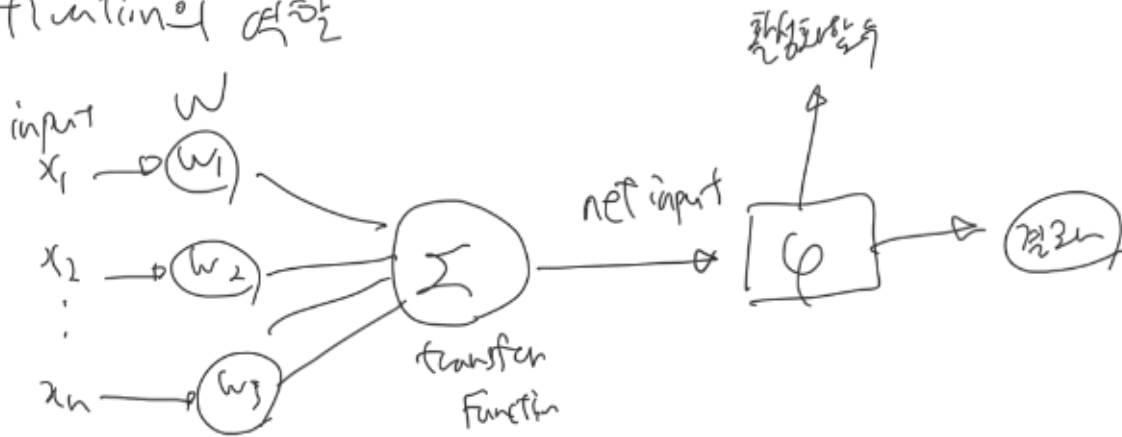
Neural

Network

Activation function

\therefore Deep한 모델을 쓸 때
 비선형 함수를 Activation Function으로 써줘야 함
 (단순 선형 모델은 증명하기 힘들어)
유니버

① Activation의 역할



활성화 함수는 train 과정에서 계산량 (많음)
역전파에서도 사용해야 함으로
 → 활성화 함수를
 연산의 효율성이 중요함.
 미분 가능 손실값을 줄이는 과정.

② 몇몇 Activation 함수 소개

* Non-linear Activation 종류 (몇몇 ex)

① Sigmoid



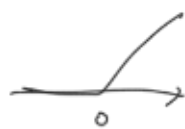
→ 출력의 중심이 0이 아니기 때문에 (단점)
 0(1) 가중치에 대한 기울기 update가
 비효율적이지 못함.
 → Vanishing gradient.

② Tanh → Sigmoid를 중심이 0으로 바뀔 형태

③ ReLU 가장 많이 사용

장점) 연산 빠름 → 비교 연산 (0)만 하여 학습률 구하기 가능

비선형 \Rightarrow 2 이상의 가산, 역전과 가산
 $f(x) = \max(0, x)$
 $f'(x) = \begin{cases} 0 & (x \leq 0) \\ 1 & (x > 0) \end{cases}$

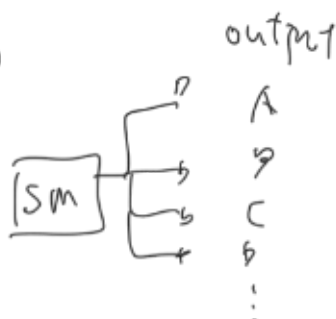


relu 문제점 \rightarrow input 값이 0 이하의 음수이면
 $f'(x)$ 즉 기울기가 0으로 연산되므로
 학습을 못하는 경우 발생 가능

④ Softmax

\rightarrow MNIST 등 기본적인 다중분류 문제에서 사용
 Softmax 특징 \hookrightarrow multi classification

[input을 0 ~ 1 사이로 정규화
 출력의 합이 1 (행 0(1))]



단점 $\rightarrow f_i(\vec{x}) = \frac{e^{x_i}}{\sum e^{x_j}}$

즉 사수함수를 사용해서 연산이 커지면
 overflow 발생 가능

이외에도 ReLU의 dying ReLU 해결책은

Leaky ReLU \rightarrow 음수일 때 0으로 상수로 곱해준다

$$f(x) = \max(0, x)$$

$$f'(x) \rightarrow \begin{cases} 0 & (x \leq 0) \\ 1 & (x > 0) \end{cases}$$

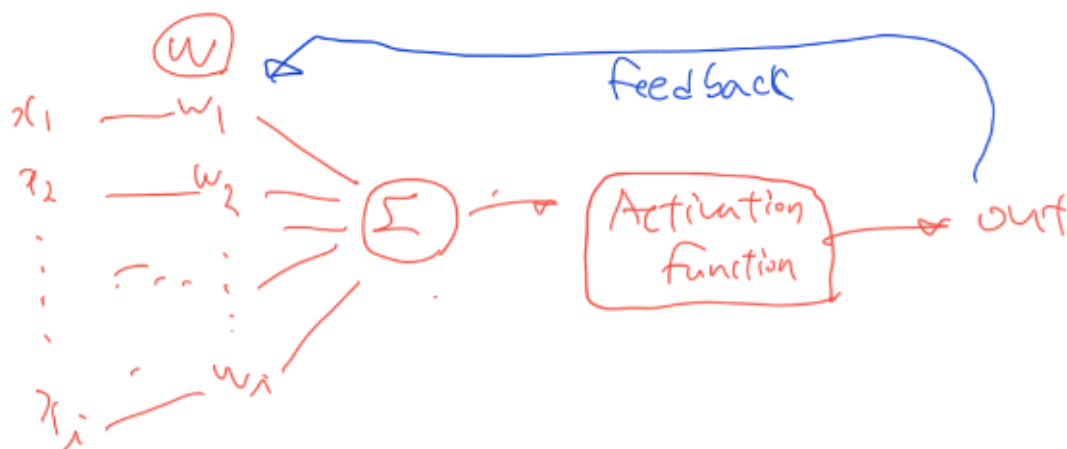
\rightarrow 0은 아니어서
 기울기 0이 아닌
 양수.

\Rightarrow 그렇기때문에 ReLU보다 좋은 성능을 내는 것은 아니라고 할

결론 : '모든 문제에 최적화된 활성화 함수는 없다'

↓
문제에 알맞은 Activation function을 선택하여
좋은 성능을 낼 수 있도록 하는 것이 중요하다

★ 선형과 활성화 함수의 문제는 2점 문제?



활성화 함수는 이전 Layer층의 결과값을
다음 층의 입력값으로 변환해주는 역할
[or]
전체 층의 출력값을 나타내는 함수이다.

⇓

이 활성화 함수를 문제의 유형, 특징에 맞게 선택 하는 것이
중요 → 그래야 최대한 이전 층들의 data들을 보다
정확하고 학습이 되도록, 좋은 성능을 보이도록
사용가능하다고 생각.

이때 이 Activation function을 통해 좋은
결과를 얻는다.

output이 나오게

Loss Function에서 정확한 error를
계산하고 cost에서 최적의 gradient를
찾아내어 역전파를 통해 W 와 B 를
update 해주는 것이 중요한 역할을 갖는 것
으로 생각.

Loss Function과
Activation Function

나 활성화 함수를 알맞게 선택하여 손실함수를
알맞게 CEG 중선택하여 W, B 를 가감
하면
좋은 학습효과를 보일 수
있다고 생각하여
바로 keyword로 선정하여
발표해 보았습니다.