

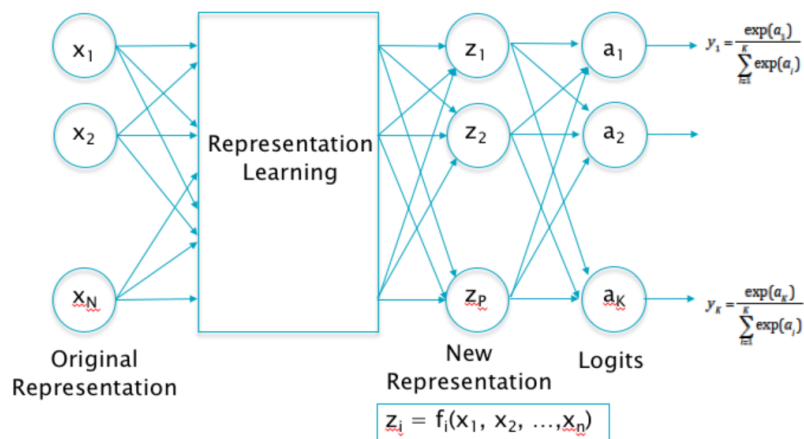
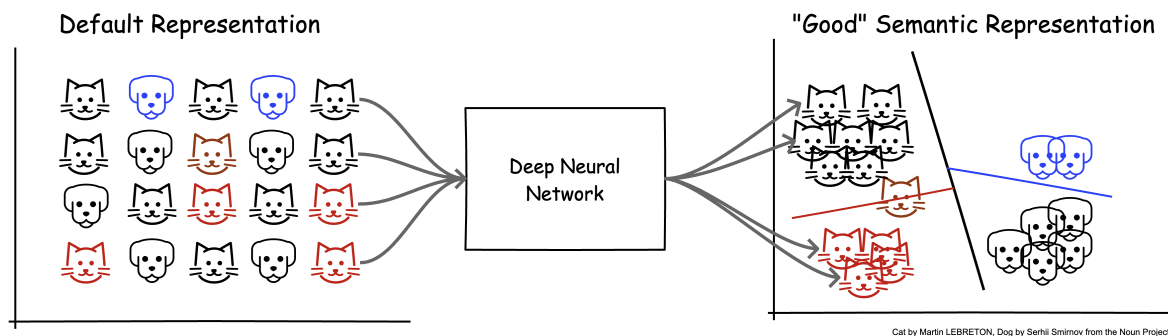
Logistic Regression

Logistic Regression

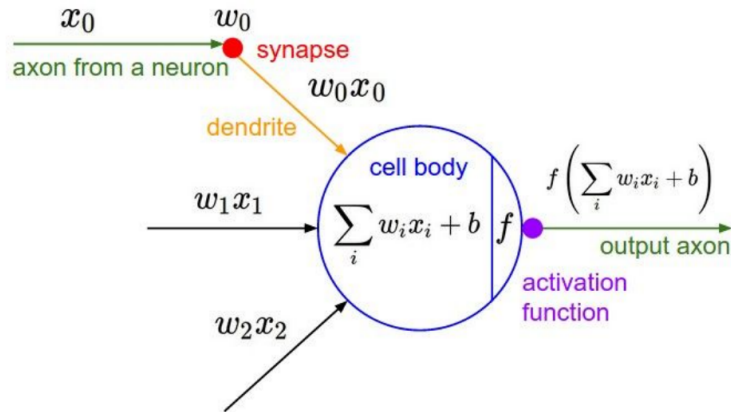
: 회귀를 사용하여 어떤 데이터가 어떤 범주에 속할 확률을 0~1의 값을 예측 & 해당 확률에 따라 가능성이 더 높은 범주에 속하는 것으로 분류하는 방법

From Generative to Discriminative Proof

Representation Learning



- DNN, CNN 등에서 최종 task 유형에 따라 "new representation"을 뽑게 하는 것
- 복잡한 데이터 공간을 선형 분류가 가능할 정도로 단순화해 표현하도록 함 → 딥뉴럴네트워크의 높은 성능
- **logit 함수**: 확률을 가지고 음과 양의 무한대 사이에서 실제 숫자 생성
- 뉴럴네트워크를 구성하는 뉴런
 - activation function으로 비선형 함수를 활용해 선형모델의 한계를 극복하고자 함 (뉴럴네트워크의 본질이 선형 모델인 이유: 가중치와 입력값 사이의 선형결합)

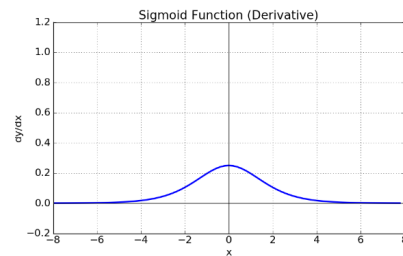
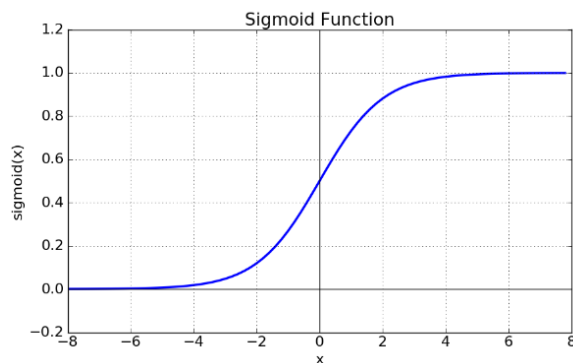


Activation Function

: 결과를 카테고리 값으로 변환시키는 함수

1) Sigmoid

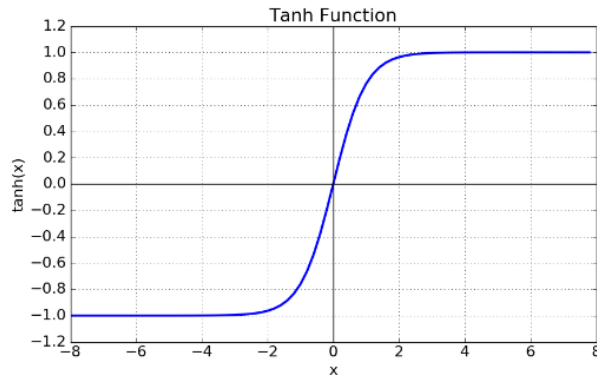
$$\sigma(x) = \frac{1}{1 + e^{-x}}$$



- 결과값이 0과 1 사이의 값을 가지도록 제한함
- zero-centered가 아님
 - 함숫값이 항상 양수이므로 한 노드에 대한 모든 파라미터 w 의 미분값은 같은 부호를 갖게 됨 → 같은 방향으로 업데이트 ("지그재그" 형태) → 학습이 느려짐
- 기울기 소실 (vanishing gradient) 문제
 - 입력값이 일정 이상 올라가면 미분값이 거의 0에 수렴하게 됨. 역전파에서 출력값이 현저하게 감소됨 (gradient들이 매우 작은 값을 갖게 됨)

2) tanh (hyperbolic tangent activation function)

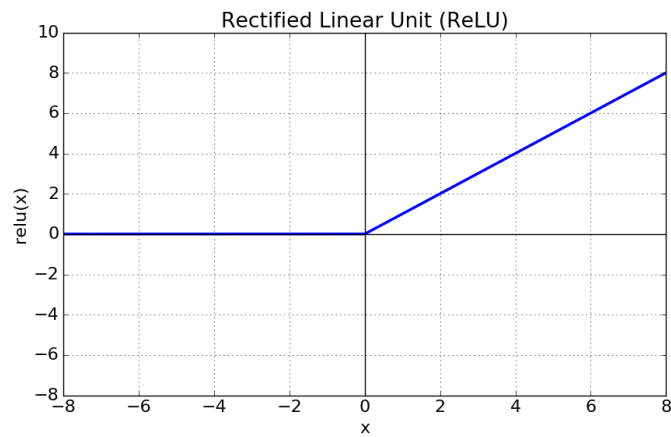
$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$



- zero-centered: 함수의 중심값을 0으로 옮김
- vanishing gradient 문제 여전히 존재

3) ReLU (rectified linear unit function)

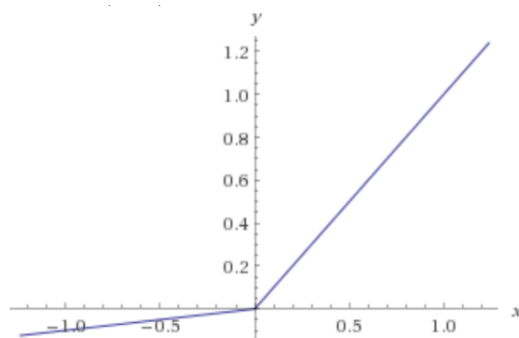
$$f(x) = \max(0, x)$$



- 가장 많이 쓰이는 함수
- $x > 0$ 이면 기울기가 1인 직선, $x < 0$ 이면 함수값이 0
- vanishing gradient 문제 해결
- $x < 0$ 인 값들에 대해서는 항상 기울기가 0이기 때문에 뉴런이 죽는 “dying ReLU” 현상 나타남

4) Leaky ReLU

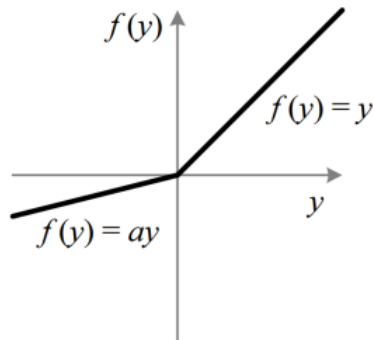
$$f(x) = \max(0.01x, x)$$



- dying ReLU 현상 해결하기 위한 함수
- $x < 0$ 인 값에 대해서도 미분값이 0이 되지 않음
- 위 식에서 0.01 대신 매우 작은 값 사용 가능

5) PReLU

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

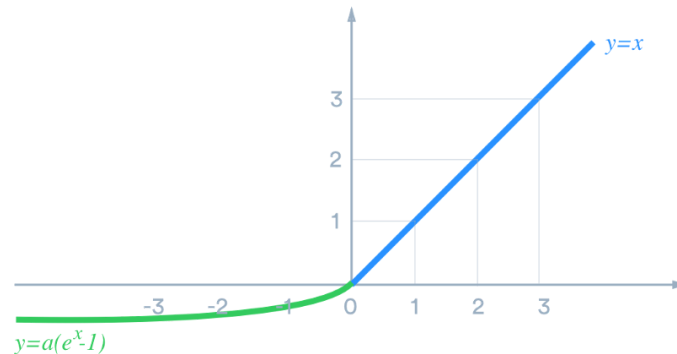


- 새로운 파라미터 α 를 추가 $\rightarrow x < 0$ 에서 기울기 학습

6) ELU (Exponential Linear Unit)

$$f(x) = x \quad \text{if } x > 0$$

$$f(x) = \alpha(e^x - 1) \quad \text{if } x \leq 0$$



- ReLU의 모든 장점 포함
- dying ReLU 해결
- 출력값이 zero-centered에 가까움
- exp 함수 계산 비용 발생

7) Maxout

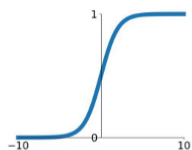
$$f(x) = \max(w_1^T x + b_1, w_2^T x + b_2)$$

- ReLU의 모든 장점 포함, dying ReLU 해결
- 계산량이 복잡함

Activation Functions

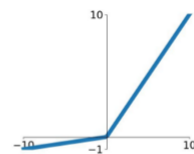
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



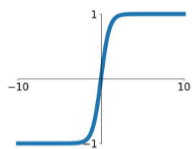
Leaky ReLU

$$\max(0.1x, x)$$



tanh

$$\tanh(x)$$

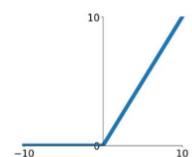


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

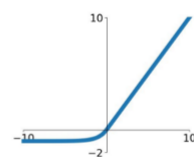
ReLU

$$\max(0, x)$$



ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



- sigmoid, tanh은 큰 성능 내기 어려움
- 우선적으로는 간단한 ReLU 사용 → 그 이후 다른 계열의 ReLU 사용