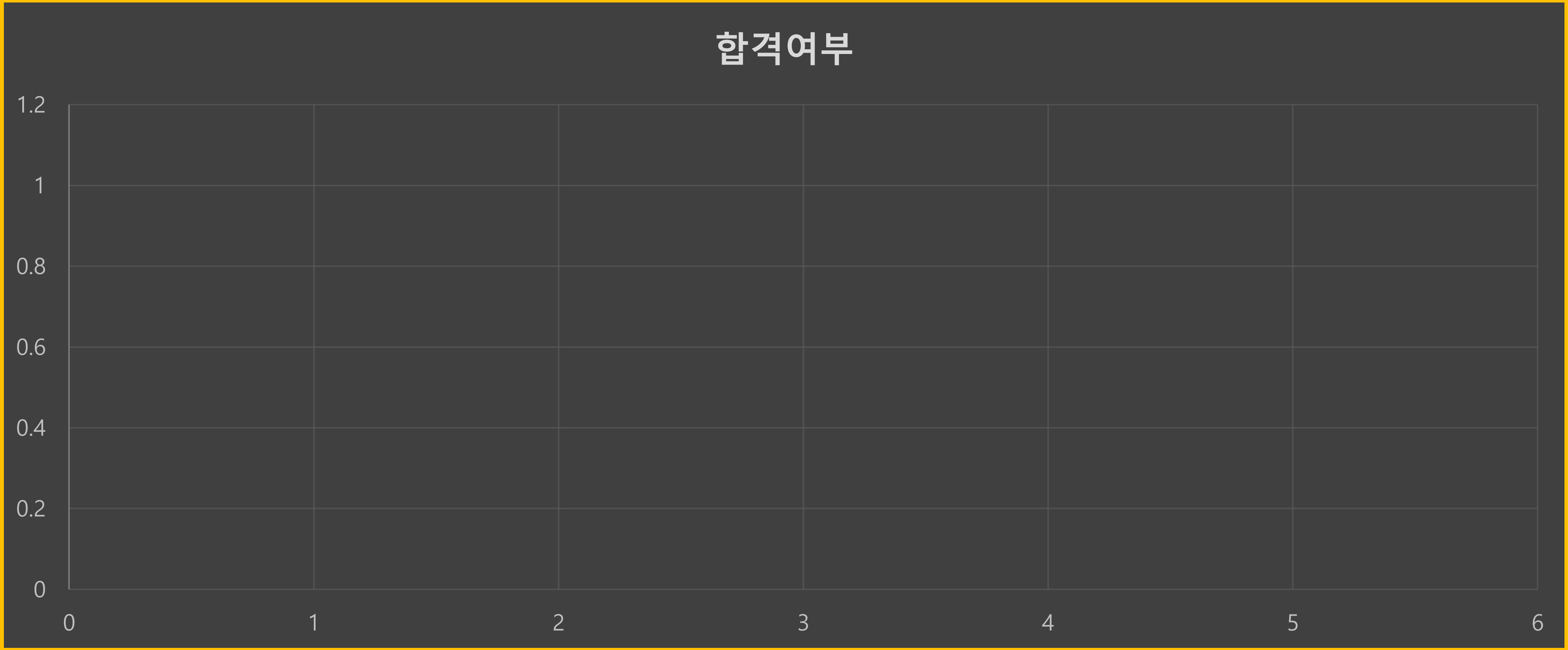
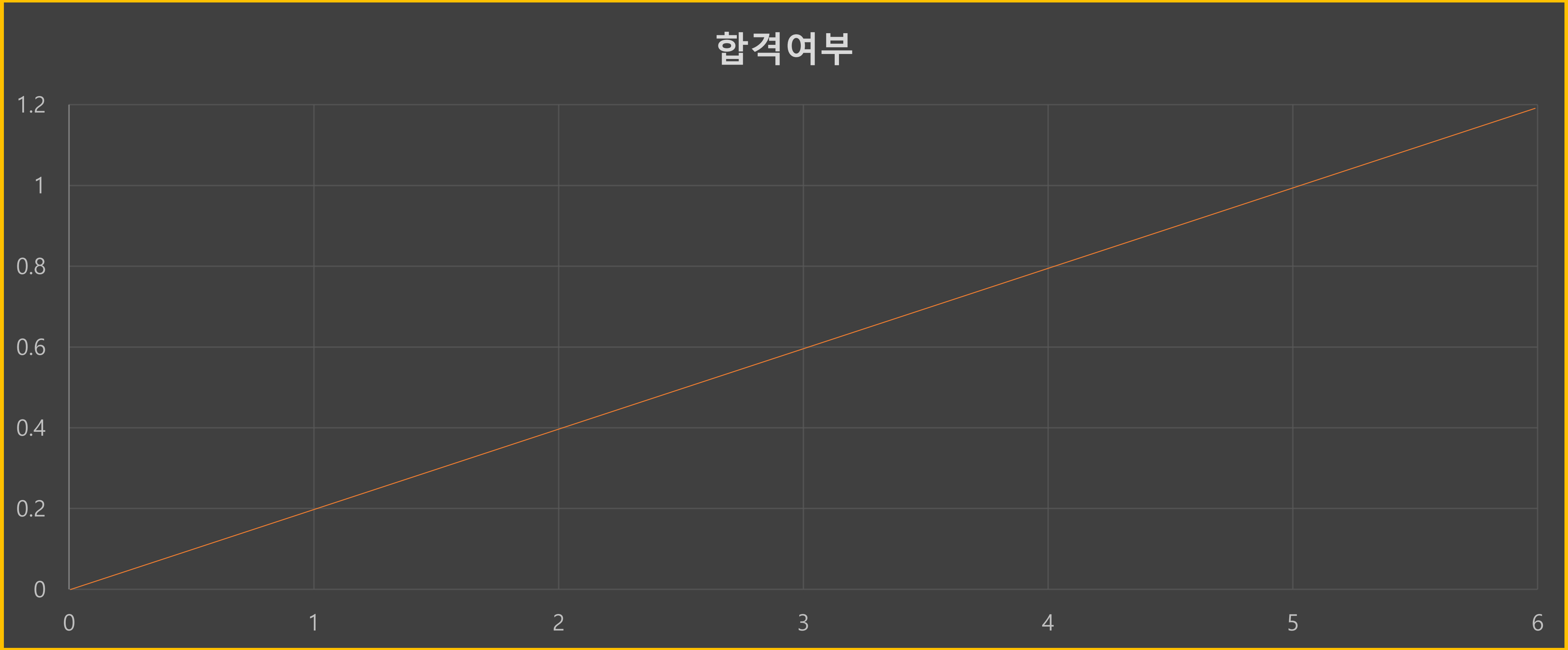
Logistic

덕근쟝

선형회귀의 한계…?



선형회귀의 한계…?



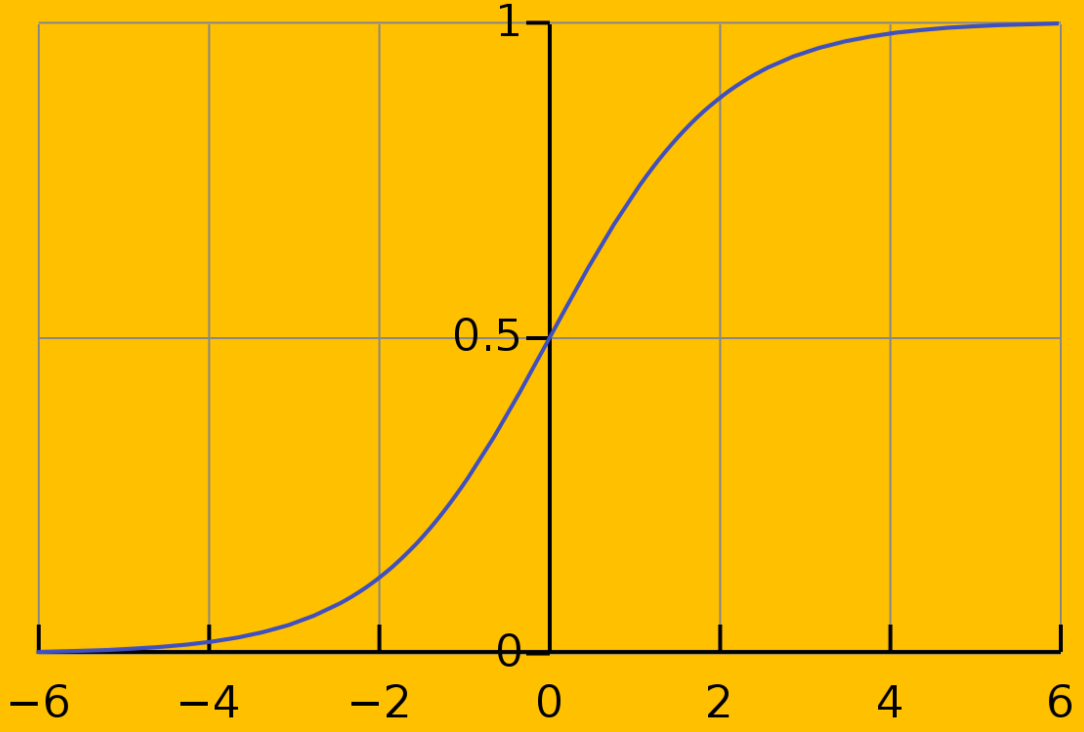
한계점

- 우선 우리가 합격(1) 불합격(0)으로 제시하려고 하는 문제에 대하여
- 자칫하면 음수가 나온다.
  예) y= 2x-5라는 가중치가 나오게 되면, 1시간만 공부하면 -3…?
- 그리고 이진 분류에 있어서 그리 좋은 그래프의 개행이 아니다.
  ->중간부분에서의 의사결정이 다소 애매하다

시그모이드 함수

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

- 보시다시피, 종속 변수는 0과 1 사이로 고정이라는 것.
- 그리고 선형회귀의 경우에는 결과가 연속적인 숫자이나,
- 로지스틱의 경우는 범주형
- 또한 input 값을 log-odds로 받는다.



수식적인 접근

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

- 우선 우리는 자연상수 e를 대충 2.7로 상정한다.
- 그리고 지수x는 일차함수 형태의 가중치 이다.
- 또한 지수 x앞에 -가 붙은 이유 : y축을 기준으로 대칭을 시켜주었다.

odd

$$\text{Odds} = \frac{\text{P(event occurring)}}{\text{P(event not occurring)}}$$

사건이 발생할 확률을 발생하지 않을 확률로
나눈 값이 odds이다.

odd

$$\text{Odds} = \frac{\text{P(event occurring)}}{\text{P(event not occurring)}}$$

위와 같은 식에서, p의 범위는 (0,1)사이 이다.

odd(p)는 (0,∞)가 된다. (분모의 크기가 0에 수렴한다고 생각해봐라)

근데 여기서 log를 씌우면?

범위는 (−∞,∞)가 된다. 실수 전체가 범위가 되기 때문에,

$$\log\left(Odds(p)\right) = Wx + b$$

으로 선형회귀의 식을 가져올 수 있게 되는 것.

일차함수가 쓰임으로써 생기는 특징

일차함수에서의 기울기는
-> 경사도에 관여

일차함수에서의 y절편은

-> 역시나 편차에 기여

그리고 후 일에 딥러닝 노드간의
상호작용이 유리하다.

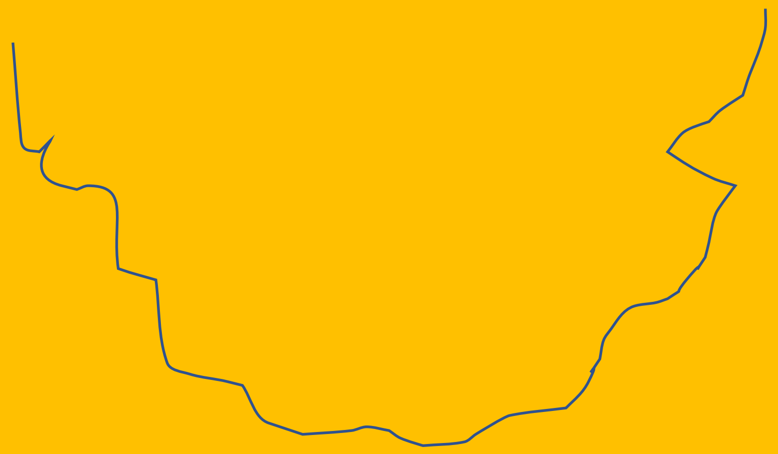
최적화에 들어가기에 앞서서

우리는 일차함수와는 다르게,
0과1사이의 값, 즉 확률의 가중치를
구하기 위해서 로지스틱을
쓴다는 것을 명심할 것.

또한, cost function의 경우,

선형이 아니기 때문에

Linear regression과 다르게
경사하강법으로 최적화가 힘들다는
점.



함수의 볼록성

손실함수

종속변수가 1과 0밖에 없으므로, 저런 형태의 함수를 가져온다.

로그의 성질은 조금만 생각해봐도 위가 0일 수는 없으니까…
(아니 뭐 된다고 쳐도 0이면 별도의 연산이 불가능..)

잘 보면 함수 자체는 똑같다.

$$cost(W) = \frac{1}{m}\sum \mathbf{c}(H(x), y)$$

$$c(H(x), y) = \begin{cases} -log(H(x)) & : y = 1 \\ -log(1 - H(x)) & : y = 0 \end{cases}$$

집중하세요~

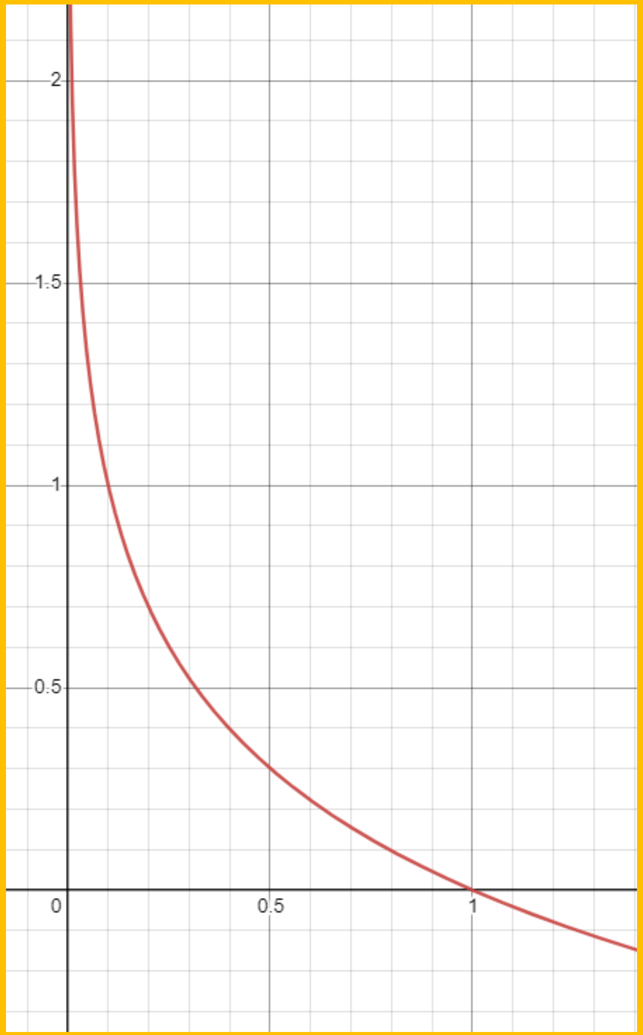
$$-\log(H(x))$$

Y=1일 때의 함수이니,

모델에서 예측한 것이 1이면
cost가 0으로 가겠고,

0으로 가면 log 0이니,

Cost가 무한대로 치고 올라간다.



집중하세요~

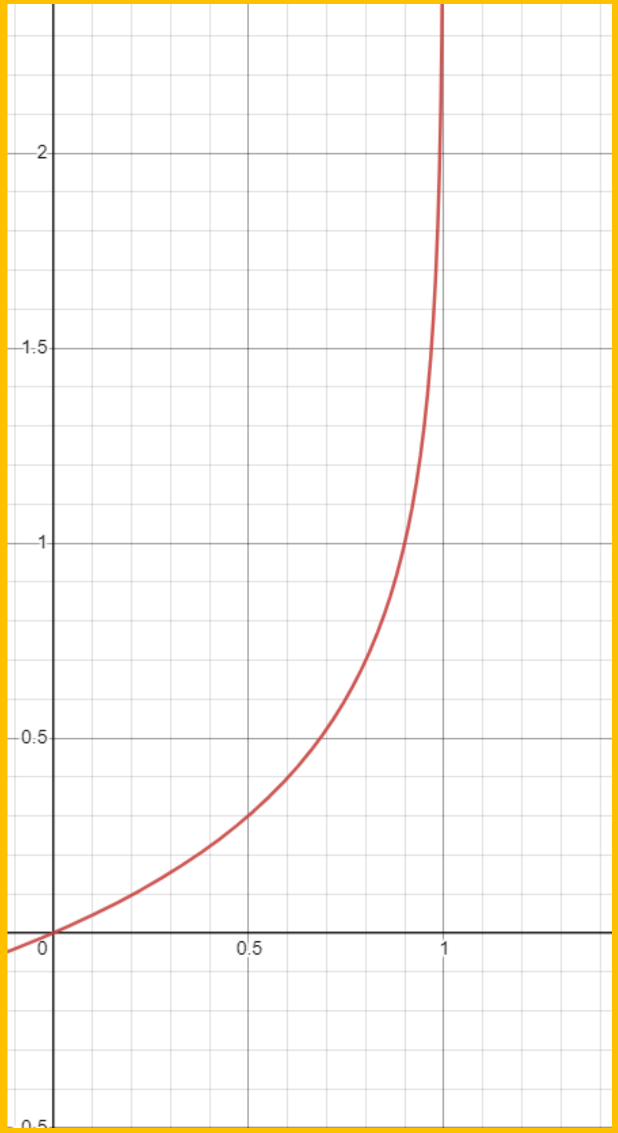
$$-\log(1-H(x))$$

Y=0일 때의 함수이니,

모델에서 예측한 것이 0이면
cost가 0으로 가겠고,

1으로 가면 log 0이니,

Cost가 무한대로 치고 올라간다.



로그 손실

따라서 이러한 함수가 생겨났다.
그리고 이를 통해 나온 sigmoid로 범주를 나누면 된다.

$$-\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}log(h(z^{(i)})) + (1 - y^{(i)})log(1 - h(z^{(i)}))]$$