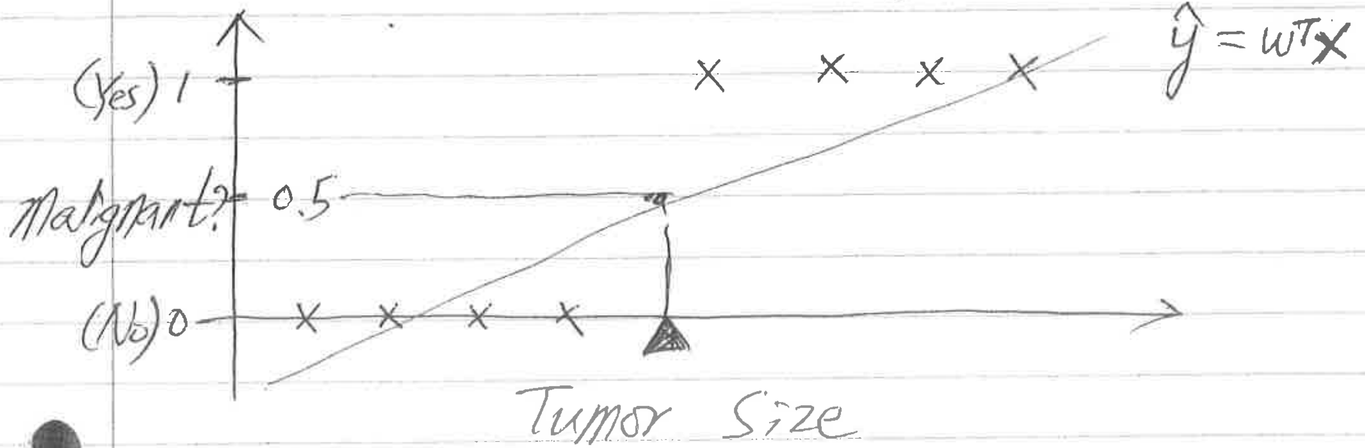
Logistic Regression

⇒ 분류(classification) 알고리즘

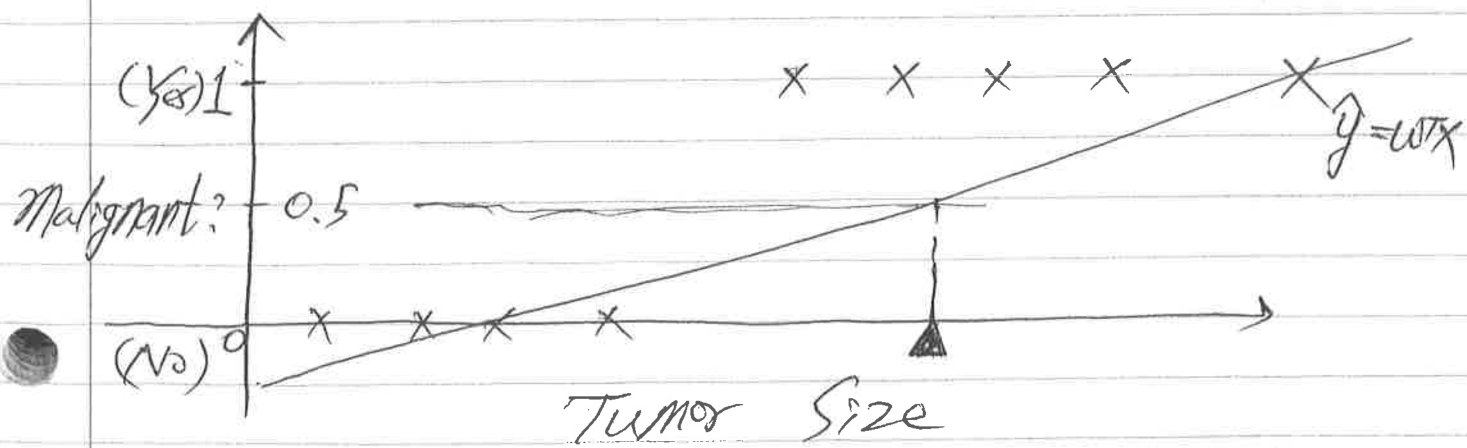
= (이진 분류 : Binary Classification) 의 예

Tumor size vs Malignancy (0 or 1)



위의 경우로 학습이 되었다면
linear regression 을 적용하여도
잘 작동이 된다.

그러나 만약 학습 sample이 아래와 같다면,
linear regression 은 성능이 매우 저하된다.

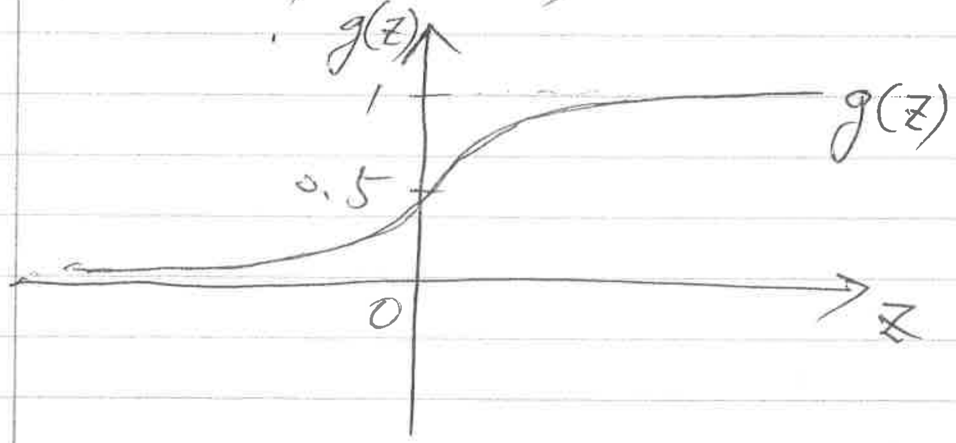


즉 Malignant 인데도 Benign으로 잘못 분류 될수도.

이와 같은 문제를 해결하기 위해서
모든 예측 값들이 0과 1 사이에 존재하게끔
linear 함수를 logistic (= sigmoid) 함수로
mapping 을 (치환을) 하면 좋다.

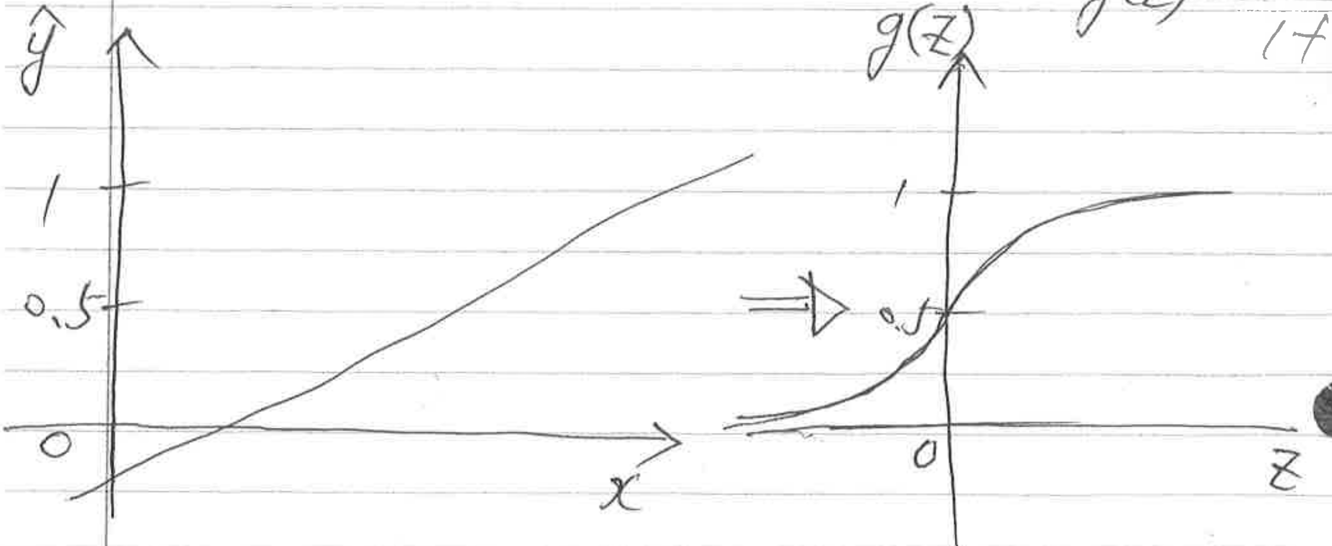
(참고) Sigmoid function = logistic function

$$g(z) = 1/(1+e^{-z})$$



(치환 즉 매핑을 하면) 즉 $z = W^T X$ 로하고

$$g(z) = \frac{1}{1+e^{W^T X}}$$



(Decision Boundary)

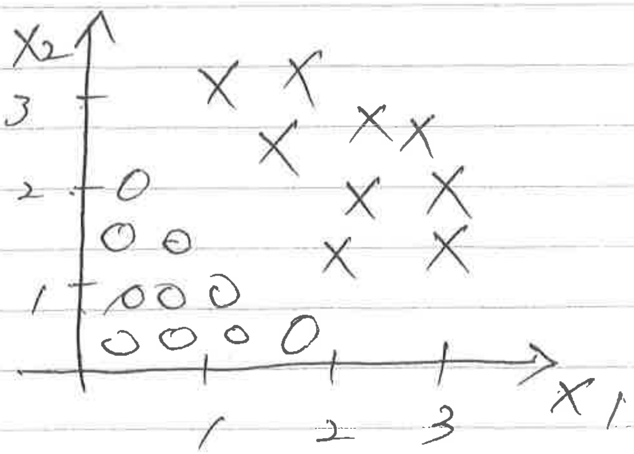
(알아두기) Sigmoid function을 활용하면
positive ($z \geq 0$)와 negative ($z < 0$)의
구분이 매우 확연하다.

따라서
$y=1$ 은 $g(z)$의 z가 0보다
큰 경우 이므로 $w^T X \geq 0$을 만족하면 되고,

$y=0$ 은 $g(z)$의 z가 0보다
작은 경우이므로 $w^T X < 0$ 이면 된다.

(예)
feature 2개, 선형으로 이루어진
모델

$$g(z) = g(b + w_1 x_1 + w_2 x_2)$$



- 만약

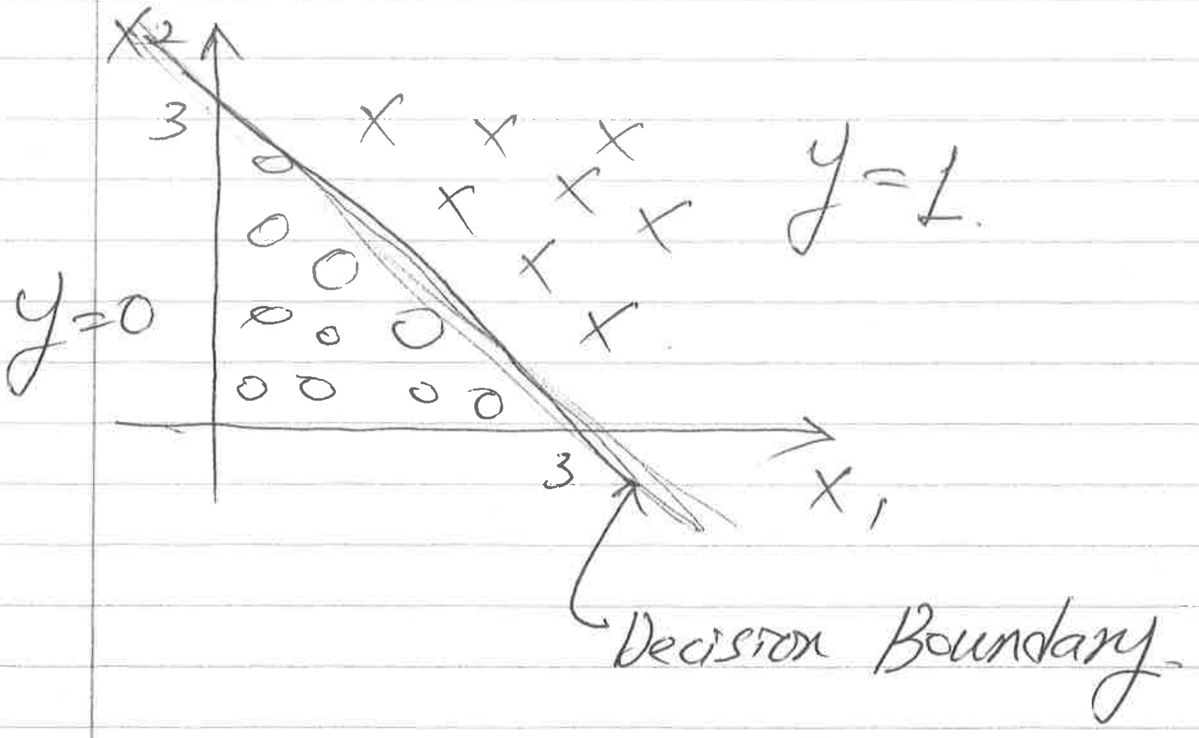
  $b = -3$

  $w_1 = 1$

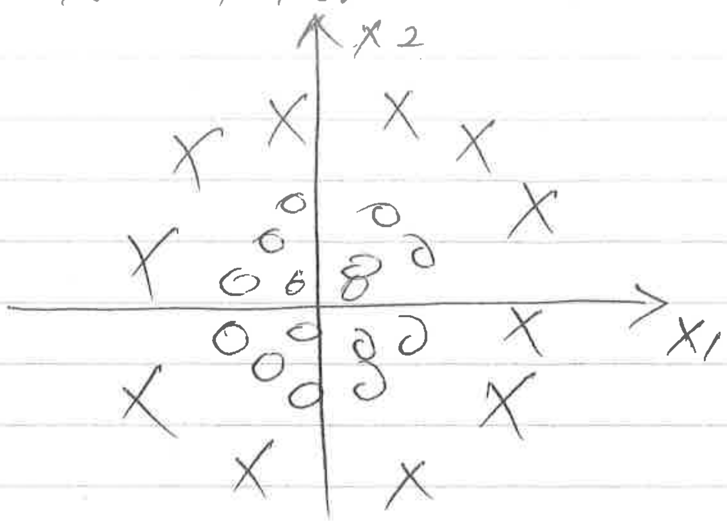
  $w_2 = 1$

  이라 하면

$$\begin{cases} -3 + x_1 + x_2 \geq 0 & \Rightarrow y = 1 \\ -3 + x_1 + x_2 < 0 & \Rightarrow y = 0 \end{cases}$$



Non-linear Decision Boundaries



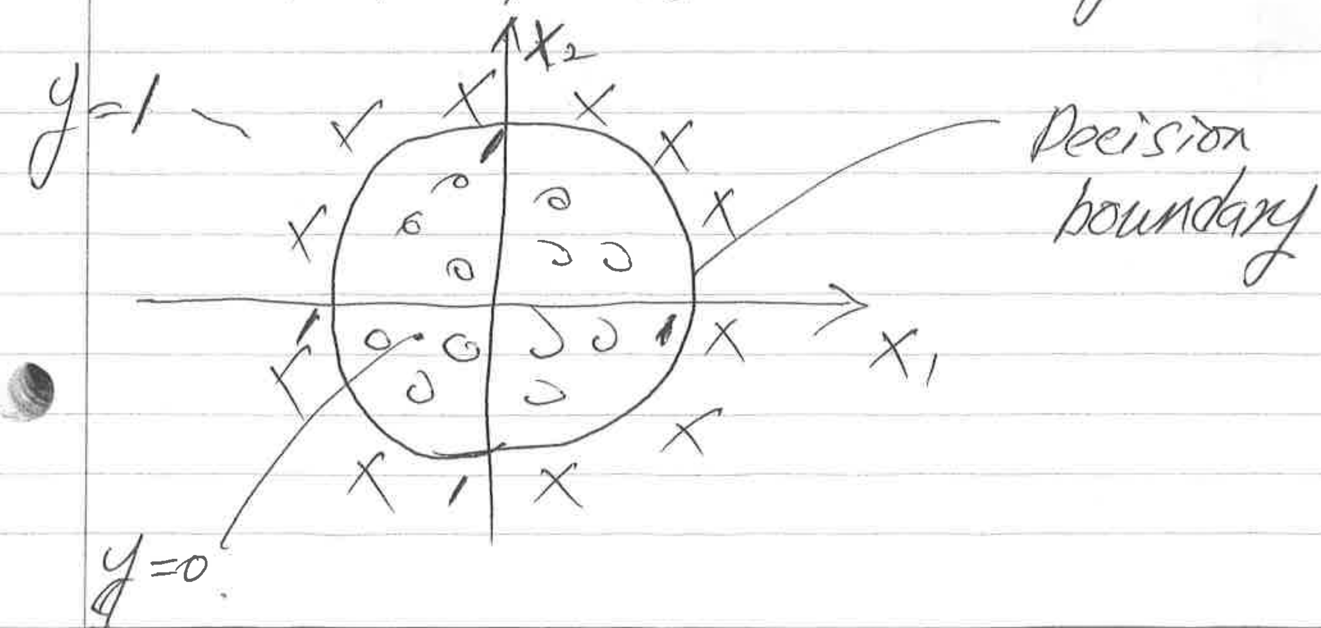
$\hat{y} = b + w_1 x_1 + w_2 x_2 + w_3 x_1^2 + w_4 x_2^2$ 이고

$g(z) = g(b + w_1 x_1 + w_2 x_2 + w_3 x_1^2 + w_4 x_2^2)$ 인데

여기서 $b = -1,\ w_1 = 0,\ w_2 = 0,\ w_3 = 1,\ w_4 = 1$

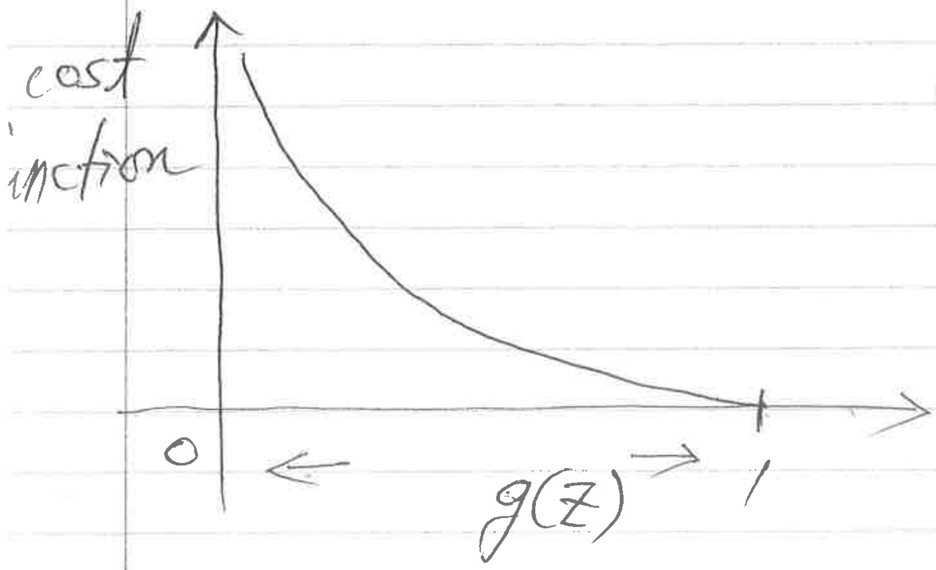
이라 가정하면

$$\begin{cases} -1 + x_1^2 + x_2^2 \geq 0 & y = 1 \\ -1 + x_1^2 + x_2^2 < 0 & y = 0 \end{cases}$$



Cost function for Logistic Regression

만약 $y=1$ 이면



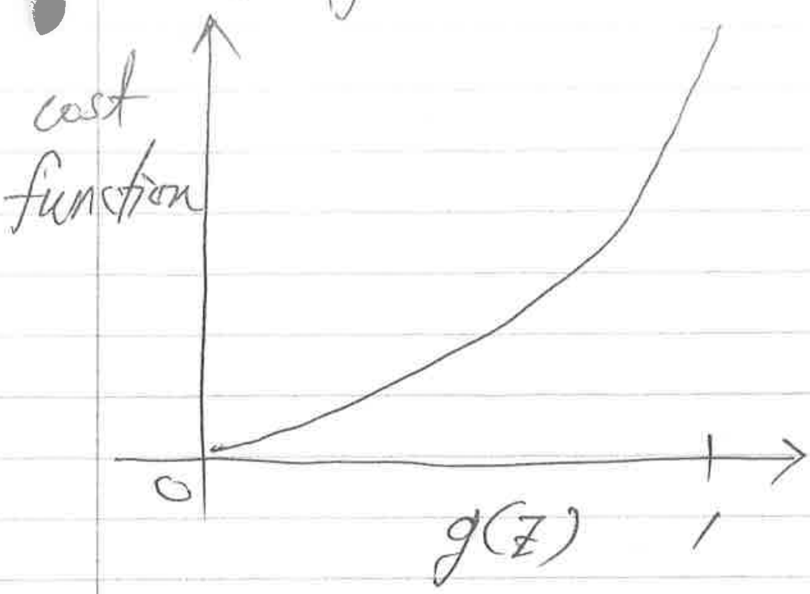
$g(Z)$가 1에 가까워지면 $y=1$일 확률이 높으므로 cost는 0에 가까워지고, $g(z)$가 0에 가까워지면 $y=1$ 일 확률이 적어지므로 cost는 무한대로 커진다.

$\boxed{0 < g(Z) < 1 \text{ 범위}}$

⇒ $y=1$ 이면 logistic regression 모델의 cost는 $g(z)$가 1에 가까워지면 0에 수렴하고 $g(z)$가 0에 가까워지면 무한대로 커지는 cost function을 갖는다.

수학적으로 $Cost(g(z), y) = -\log(g(z))$

만약 y=0 이면



위와는 반대로 g(Z)가 0에 가까우면 y=0이 될 확률이 높으므로 cost는 0에 가까워 지고, g(Z)가 1에 가까워 지면 y=0 일 확률은 높아지므로 cost는 무한대로 커진다.

수학적으로는

$$Cost(g(z), y) = -\log(1-g(z))$$

앞의 두개의 cost function을
합치면

$$Cost(g(z), y) = \begin{cases} -\log(g(z)) & , y=1 \\ -\log(1-g(z)) & , y=0 \end{cases}$$

$$\Rightarrow \quad = -y\log(g(z)) - (1-y)\log(1-g(z))$$

$$\boxed{* \text{ 여기서 } g(z) = \frac{1}{1+e^{-W^T X}}}$$

즉 linear regression을
logistic (sigmoid) function에
mapping 한
Logistic Regression Model.

Support Vector Machines (SVMs)

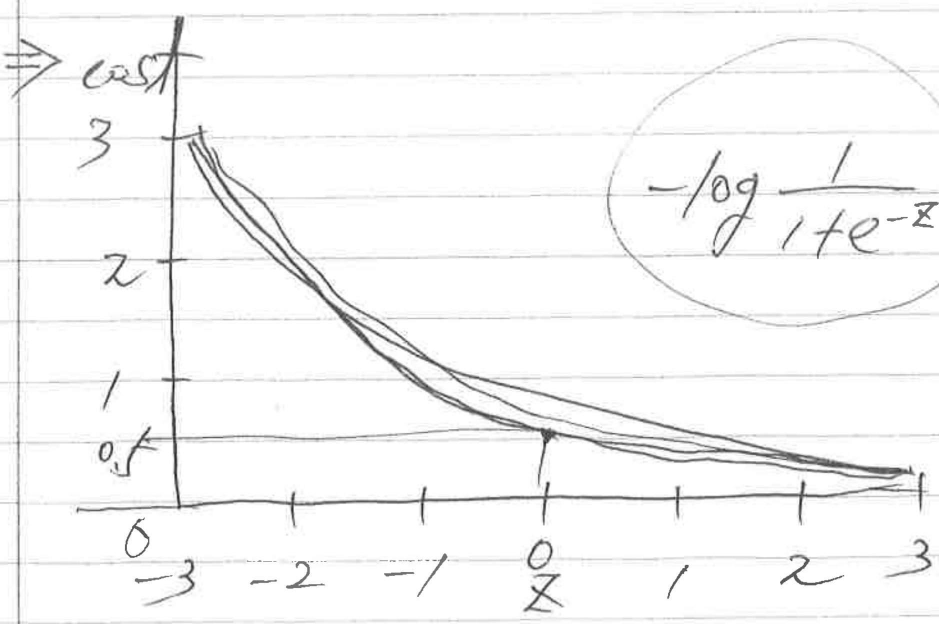
logistic Regression model의 cost function을 fully expand 하면,

Cost function

$$= -\left(y \log(g(z)) + (1-y)\log(1-g(z))\right)$$

$$= -y \log \frac{1}{1+e^{-z}} - (1-y)\log\left(1 - \frac{1}{1+e^{-z}}\right)$$

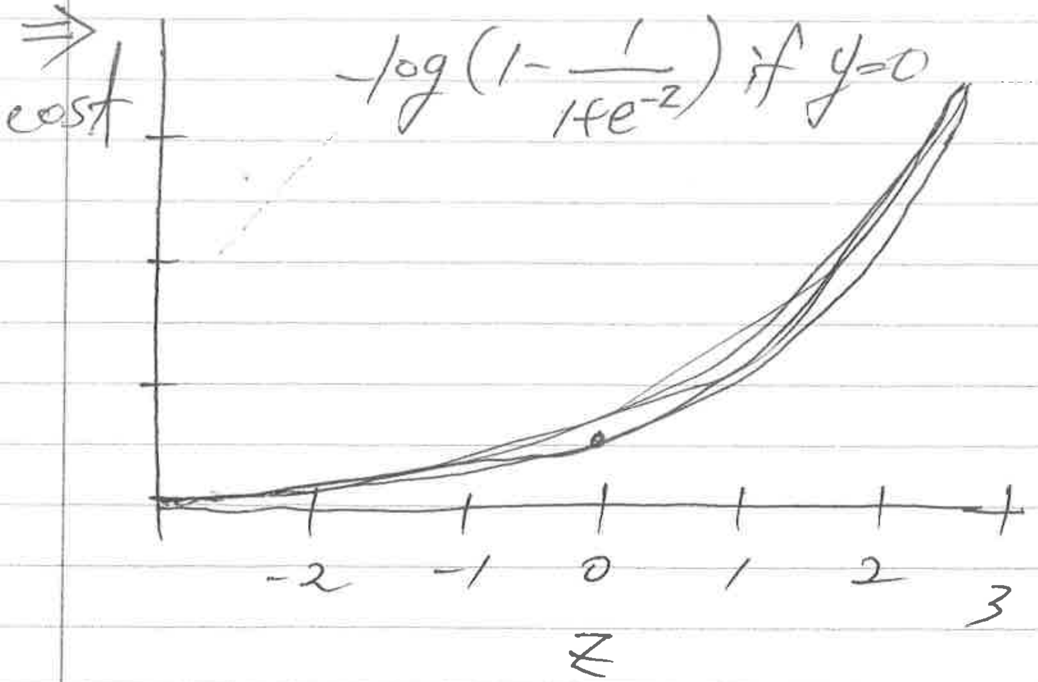
※ 여기서 $z = W^T x$ 이고 범위는 자연수 전체 (linear regression 이라 가정하면)

⇒ 

$-\log \frac{1}{1+e^{-z}}$ if $y=1$

만약 z가 충분히 크다면
즉 $\hat{y} = w^T x$ 가 충분히 크다면
비용은 충분히 낮아지므로 문제가 없고,

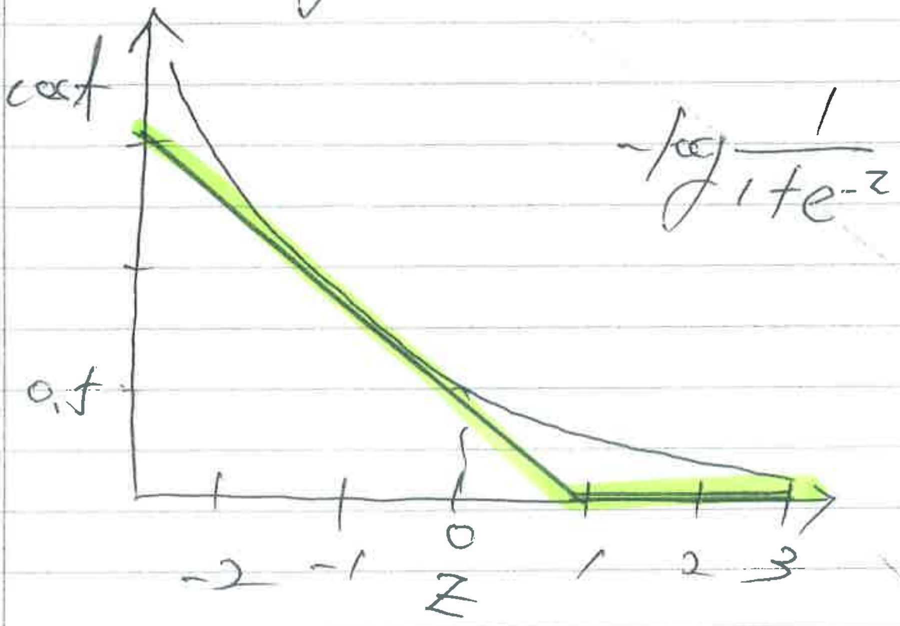
만약 z가 작아진다면
즉 $\hat{y} = w^T x$ 가 음수이 영역이면,
모델의 cost는 매우 커질 것이다.



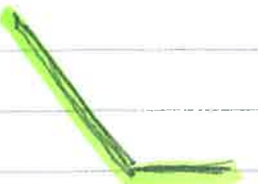
반대로 $y=0$ 이면
z가 커지면 비용이 증가하고
z가 작아지면 비용이 감소한다.

※ SVM cost functions from logistic Regression cost functions.

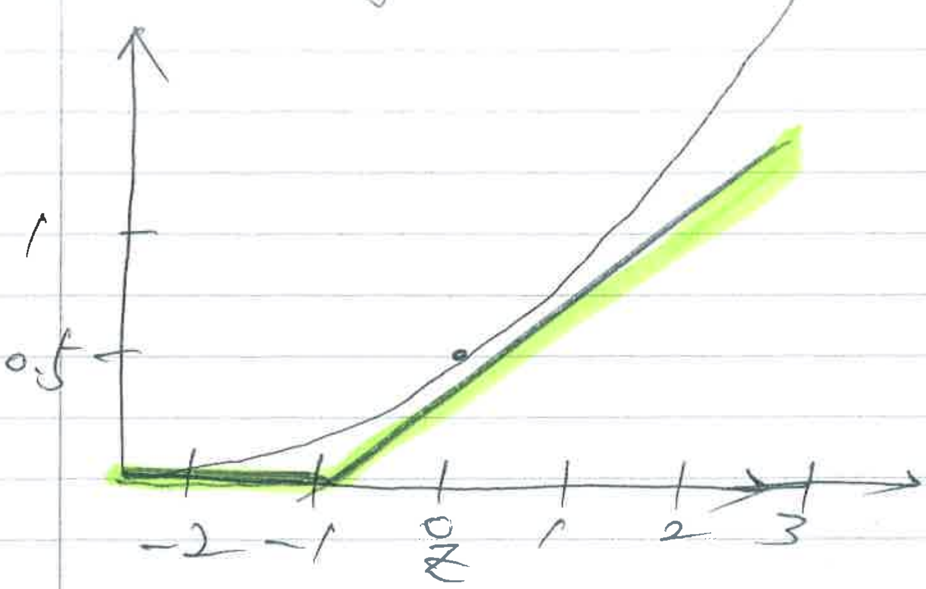
① if $y = 1$.



두개의 라인을 만들어 cost function을 조금 더 간편하게 만들면 최적화 및 계산상의 이득을 취할 수 있다.

 를 $y=1$ 일때 cost function으로 사용하자는 concept이 SVM 이다.

② if y=0 이면



비슷한 논리로 y=0 이면 아래의

모양의 cost function을 사용하라는 concept이 SVM이다.

⊛ 이런 비용함수를 제곱 힌지 (Squared Hinge) 손실 함수라 함.

(Regularization 이해하기.)

◎ linear regression 에서는

$cost = MSE + \lambda B$ (여기서 $B = \sum_{i=1}^{n} w_i^2$)

의 cost를 minimize 하는 것으로

regularization을 함으로 λ가 커지면

규제를 많이하고 λ가 작아지면

규제를 적게하는 것으로 이해를 하였다.

◎ 그러나 Logistic Regression과

SVMs 의 경우는

$C = 1/\lambda$ 이라는 매개변수를 사용하여

규제를 하므로 C가 커지면 규제를

적게하고 C가 작으면 규제를 많이함.

$\min_{w} C \sum_{i=1}^{m} [$ 비용함수 $] + \sum_{i=1}^{n} w_i^2$ 의형태.