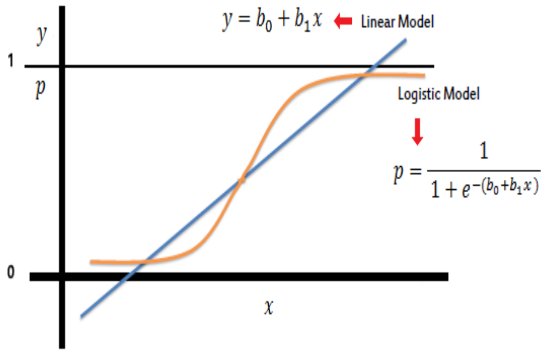
Linear Regression & Logistic Regression



Fast Campus
Start Deep Learning with Tensorflow

Let's Go to the Deep Learning World!!

쉬운 것부터 시작해봅시다

초등학교 6학년 수학 – 정비례

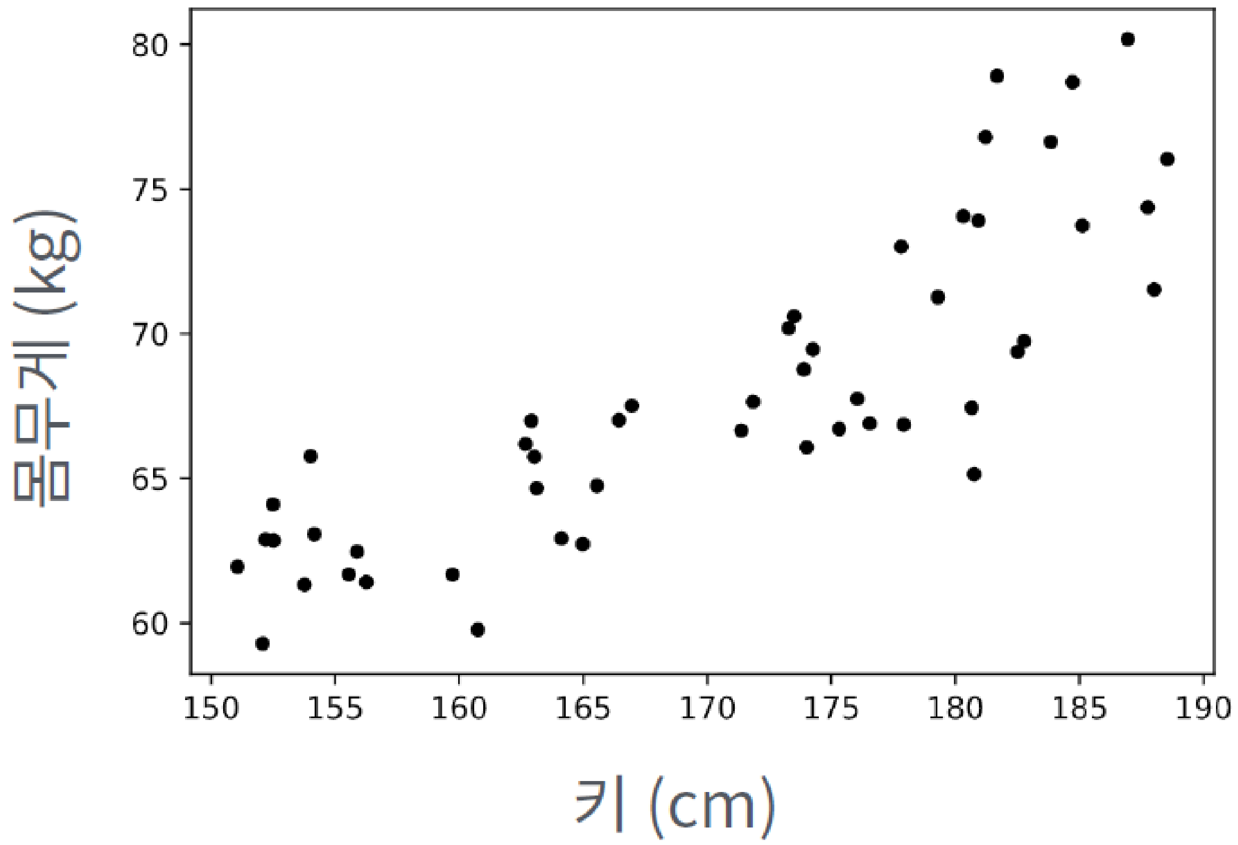
2배 3배 4배

닭 수 (마리)	1	2	3	4	5	……
다리 수 (개)	2	4	6	8	10	……

2배 3배 4배

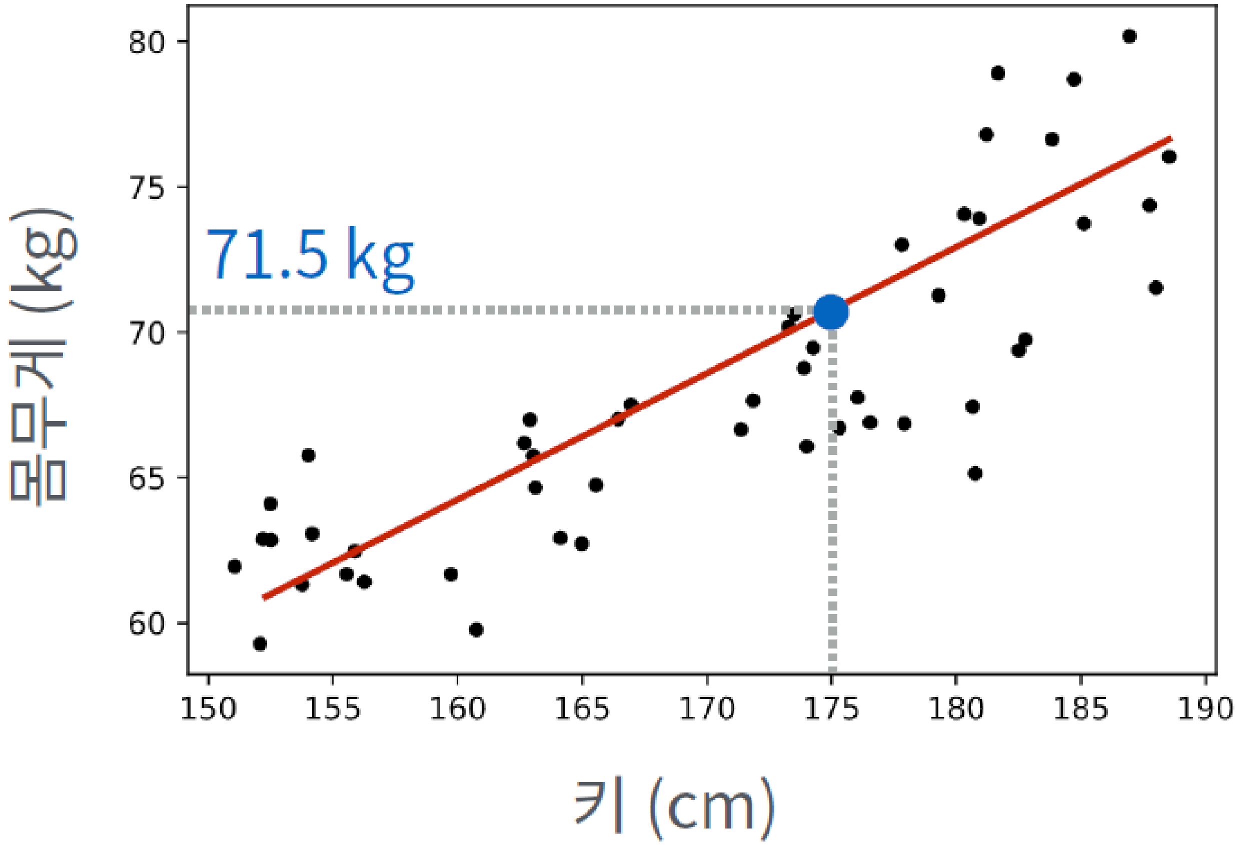
Linear Regression

- 어느 학교 학생들의 신체검사 자료
- 새로 전학온 학생 A의 키가 175cm일 때 예상 몸무게는?

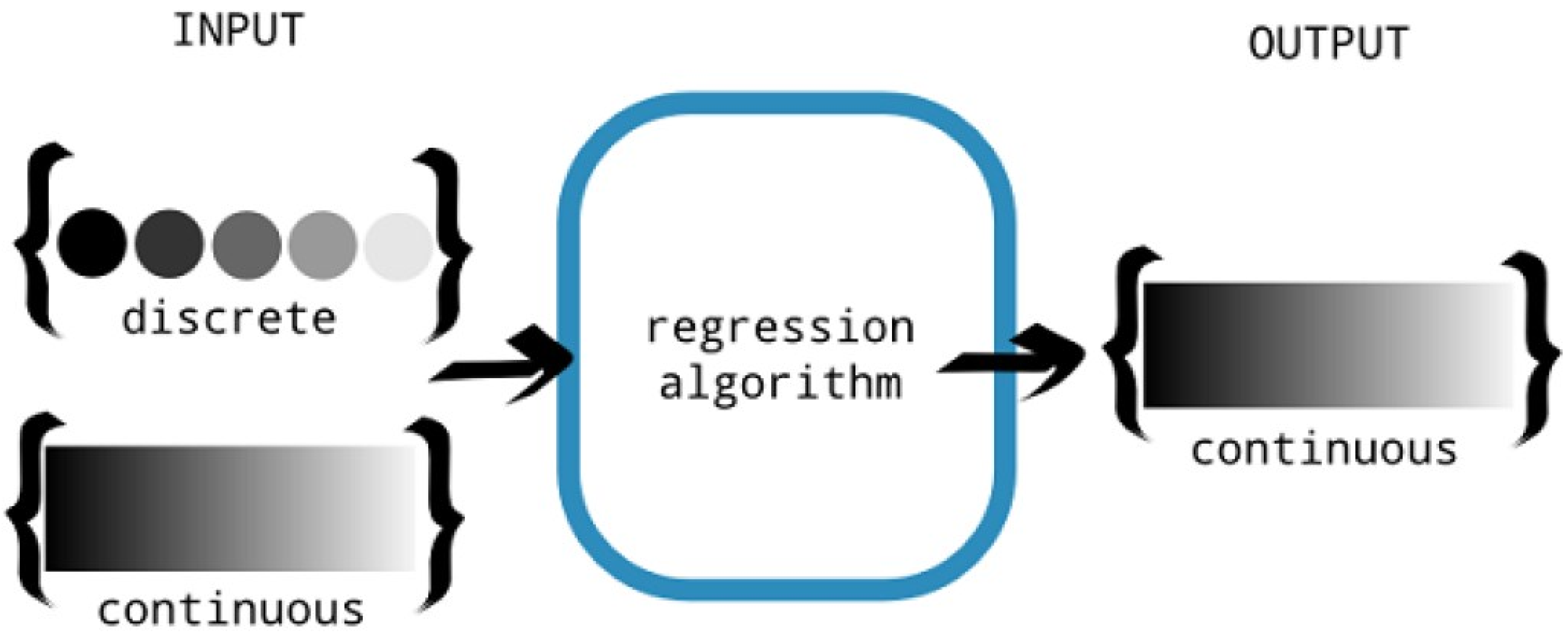


Linear Regression

- 어느 학교 학생들의 신체검사 자료
- 새로 전학온 학생 A의 키가 175cm일 때 예상 몸무게는?

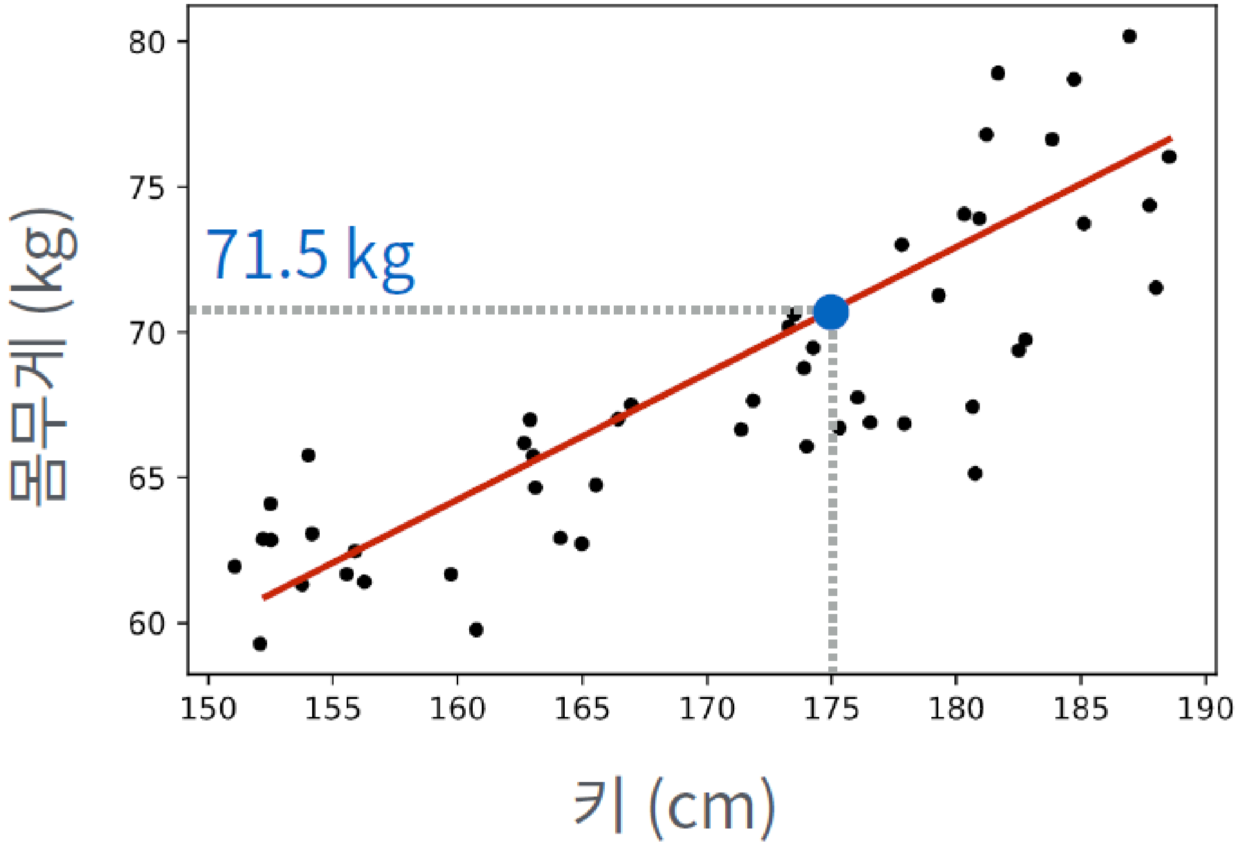


Regression?


INPUT
OUTPUT
discrete
continuous
regression
algorithm
continuous

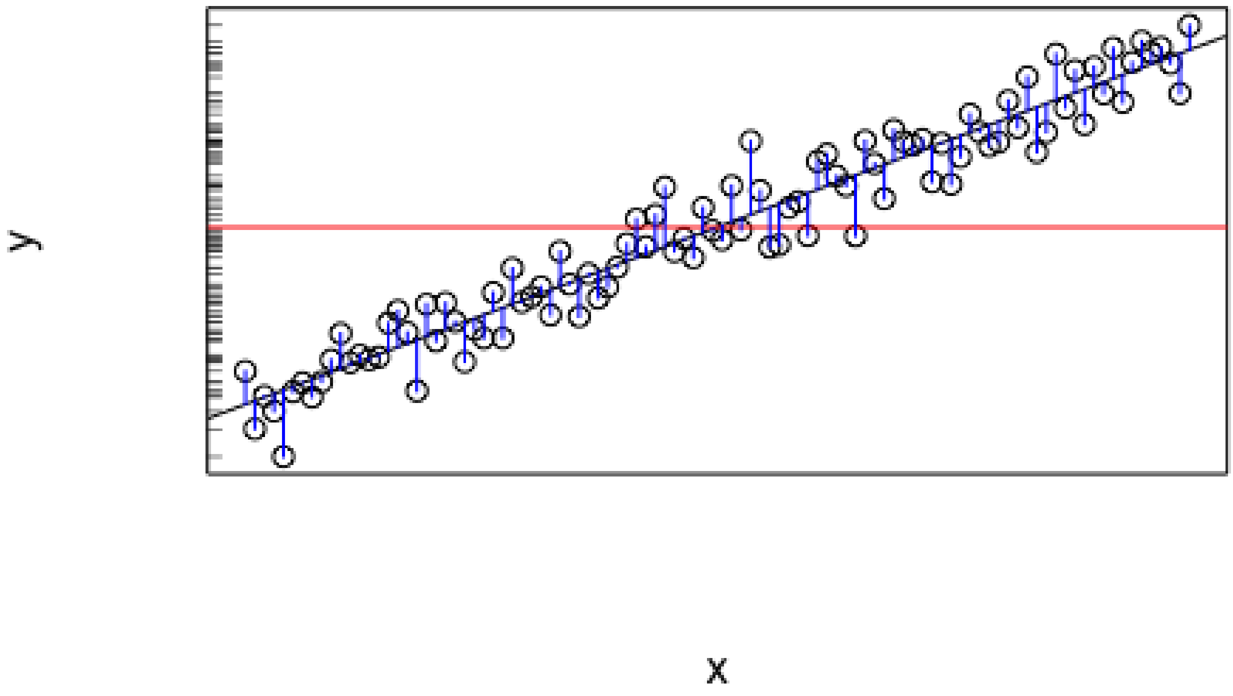
Linear Regression

- 선형함수(예 : 1차함수)로 주어진 data를 근사한다
- $y = wx + b$



Linear Regression

- 잘 예측했는지 측정할 척도(metric)가 필요함



$y^* = wx + b$ (예측값)

$$Cost(Loss) = \sum_i (y_i - y_i^*)^2$$

$$= \sum_i (y_i - wx_i - b)^2$$

Linear Regression

- Cost(Loss) 값을 minimize하는 w와 b를 구하면 될텐데.... 어떻게?
  - Random Search – 가능????
  - Cost function을 미분해서 최솟값(미분=0이되는 점)을 찾자!

산수를(미분을…) 조금 해야겠습니다

b 구하기

$$L = \sum_i (y_i - wx_i - b)^2$$

$$\frac{\delta L}{\delta b} = \frac{\delta \sum_i (y_i - wx_i - b)^2}{\delta b}$$

$$= -2\sum_i (y_i - wx_i - b) = ny_{avg} - nwx_{avg} - nb = 0$$

$$\therefore \boldsymbol{b = y_{avg} - wx_{avg}}$$

w 구하기

$$L = \sum_i (y_i \ - wx_i \ - b)^2$$

$$\frac{\delta L}{\delta w} = \frac{\delta \sum_i (y_i \ - wx_i \ - b)^2}{\delta w}$$

$$= -2 \sum_i x_i (y_i \ - wx_i \ - b) = -2 \sum_i x_i \left(y_i \ - wx_i \ - y_{avg} + wx_{avg}\right)$$

$$= 0$$

Multi Variable Linear Regression

- x가 scalar값(1개)가 아니라 vector가 된다면??

- Input
  - X1 : Facebook 광고료
  - X2 : TV 광고료
  - X3 : 신문 광고료
- Output
  - 판매량

FB	TV	신문	판매량
X_1	X_2	X_3	Y
230.1	37.8	69.2	22.1
44.5	39.3	45.1	10.4
17.2	45.9	69.3	9.3
151.5	41.3	58.5	18.5
180.8	10.8	58.4	12.9
8.7	48.9	75	7.2
57.5	32.8	23.5	11.8
⋮	⋮	⋮	⋮

Multi Variable Linear Regression

- 식은 여전히 $y^* = \boldsymbol{w}^T \boldsymbol{x} + b$
  - $\boldsymbol{w}^T = [\,w_1, w_2, w_3],\ \boldsymbol{x}^T = [\,x_1, x_2, x_3]$

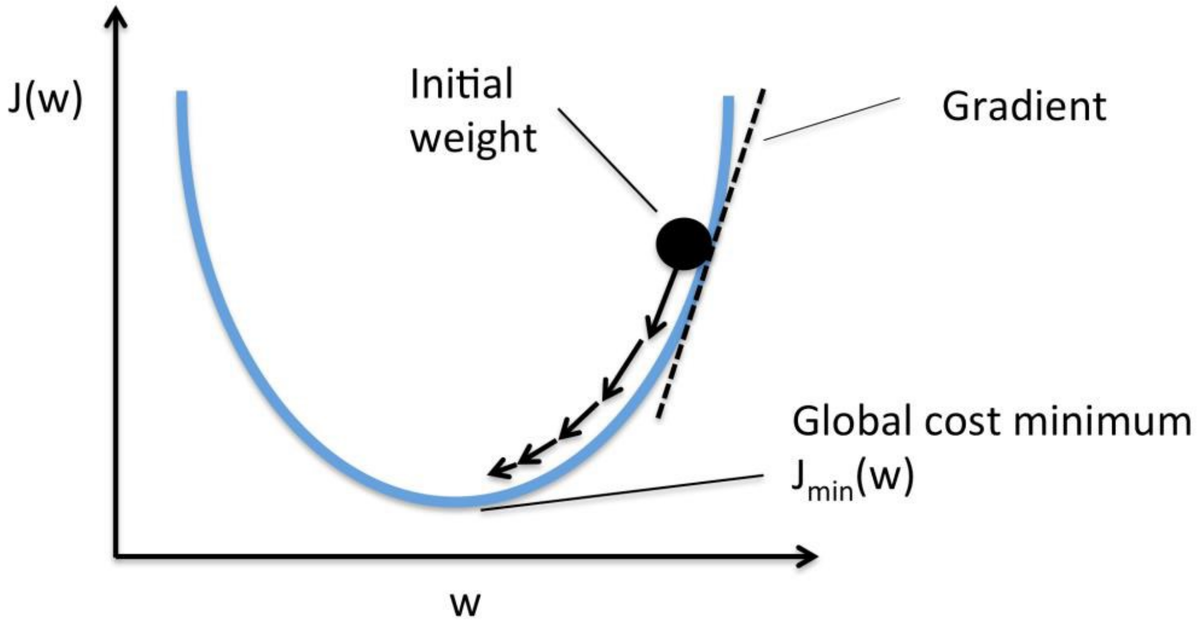
- 그런데 미분해서 0이 되는 점은 어떻게 찾아야 할까?
  - 변수가 많아지고, matrix가 커지면 복잡도가 $O(n^3)$ 이기 때문에 exponential하게 증가하여 계산이 거의 불가능함
  - minimum을 바로 구하는 게 아니라, 현재 loss에 대한 w의 gradient(경사도)를 구하여 gradient x learning rate 만큼 w를 update하자

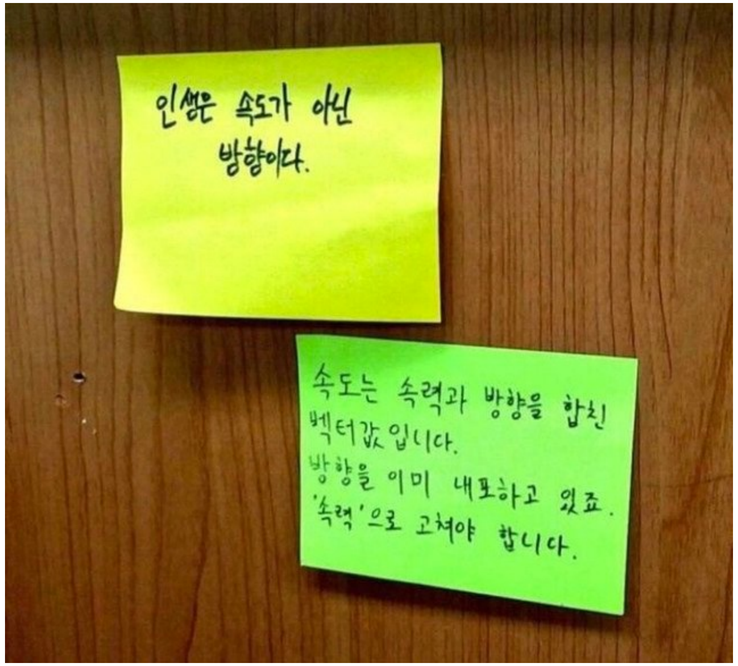
  → Gradient Descent

Gradient Descent

$$w_{new} = w - \alpha \frac{\delta L}{\delta w}$$

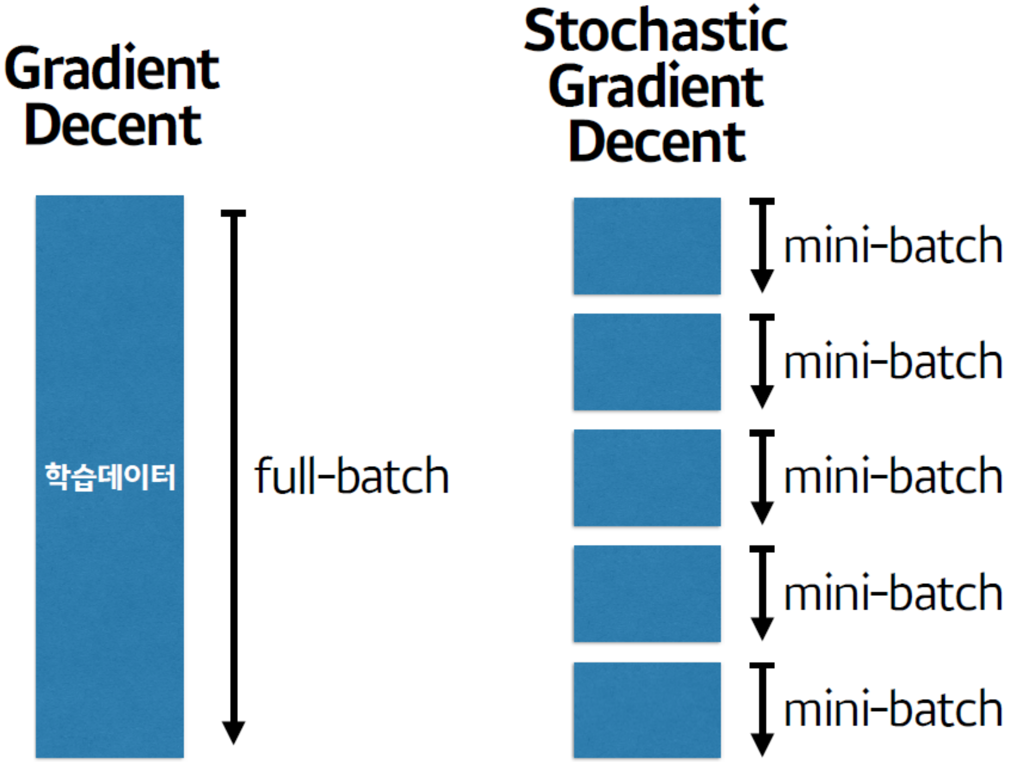
- 방향 : 그 지점에서의 – gradient
- 속력(보폭) : learning rate(α)

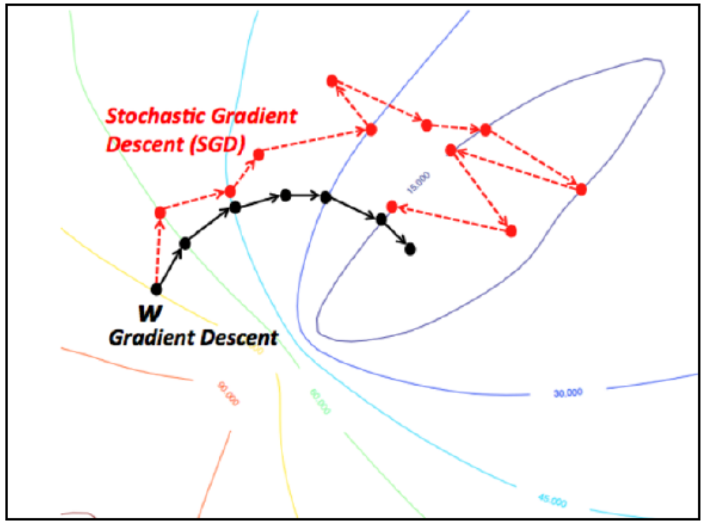




Stochastic Gradient Descent, Mini-batch Training

- Data가 너무 많아서 한번에 다 넣고 학습하면 시간도 오래걸리고, memory도 부족하게 됨





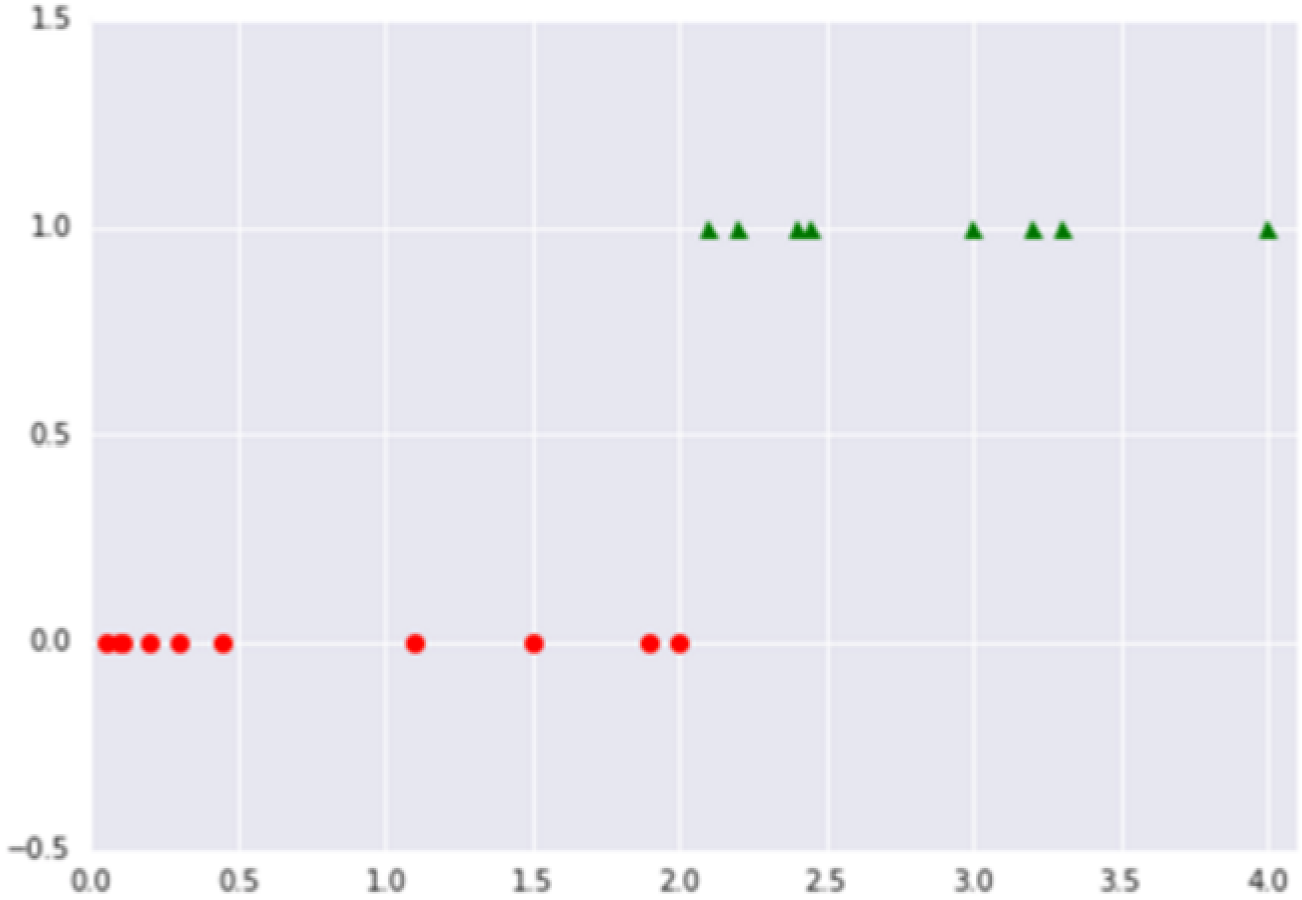
전부다 읽고나서
최적의 1스텝 간다.

작은 토막마다
일단 1스텝간다.

Classification도 할 수 있지 않을까요?

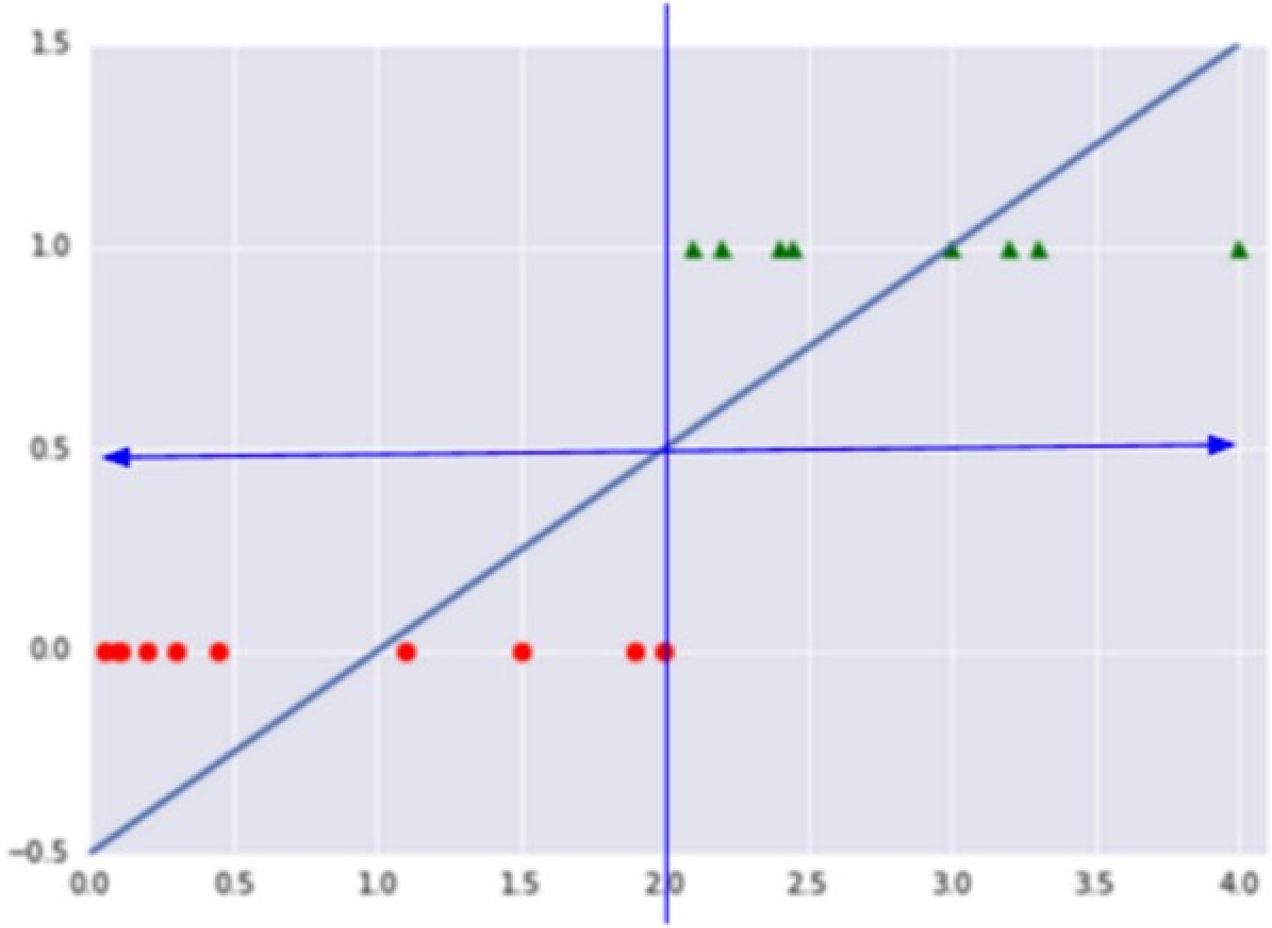
Binary Classification

- 종양의 크기에 따른 양성/음성 판별 문제
  - 1 : 양성(암), 0: 음성(정상)



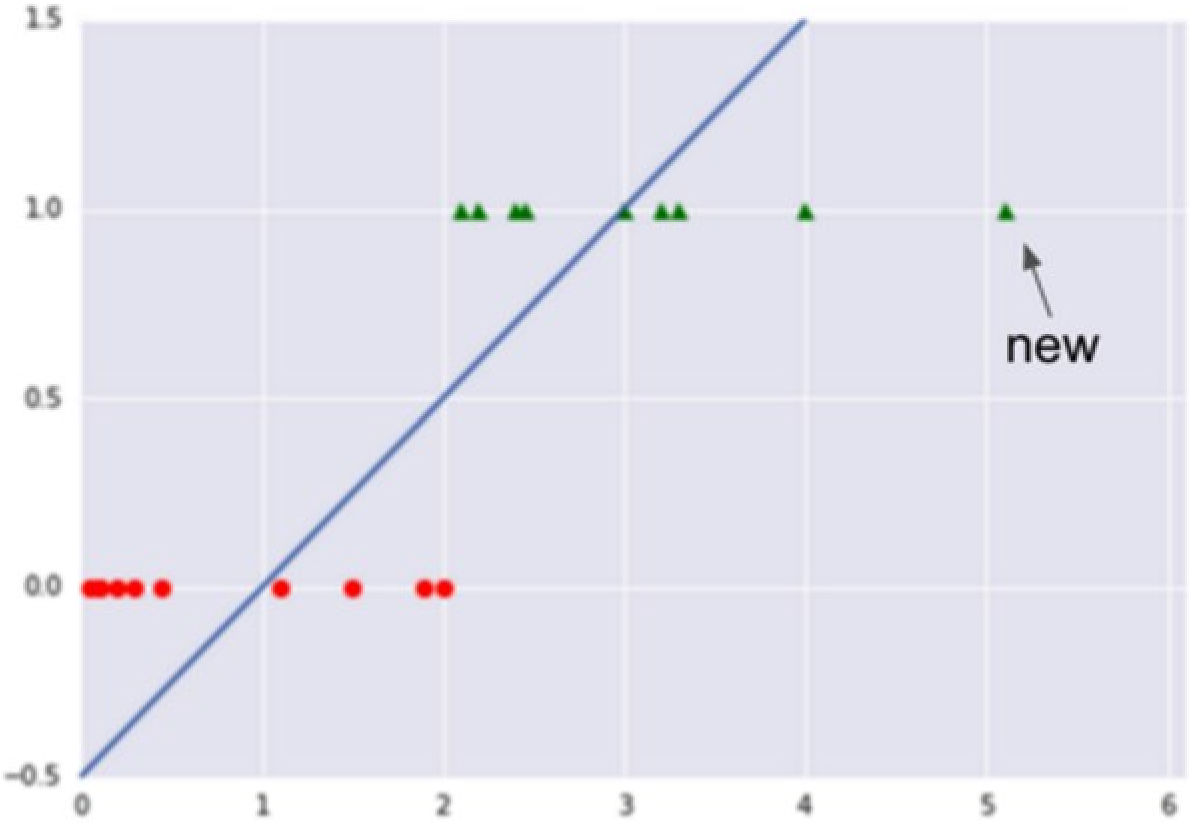
Binary Classification

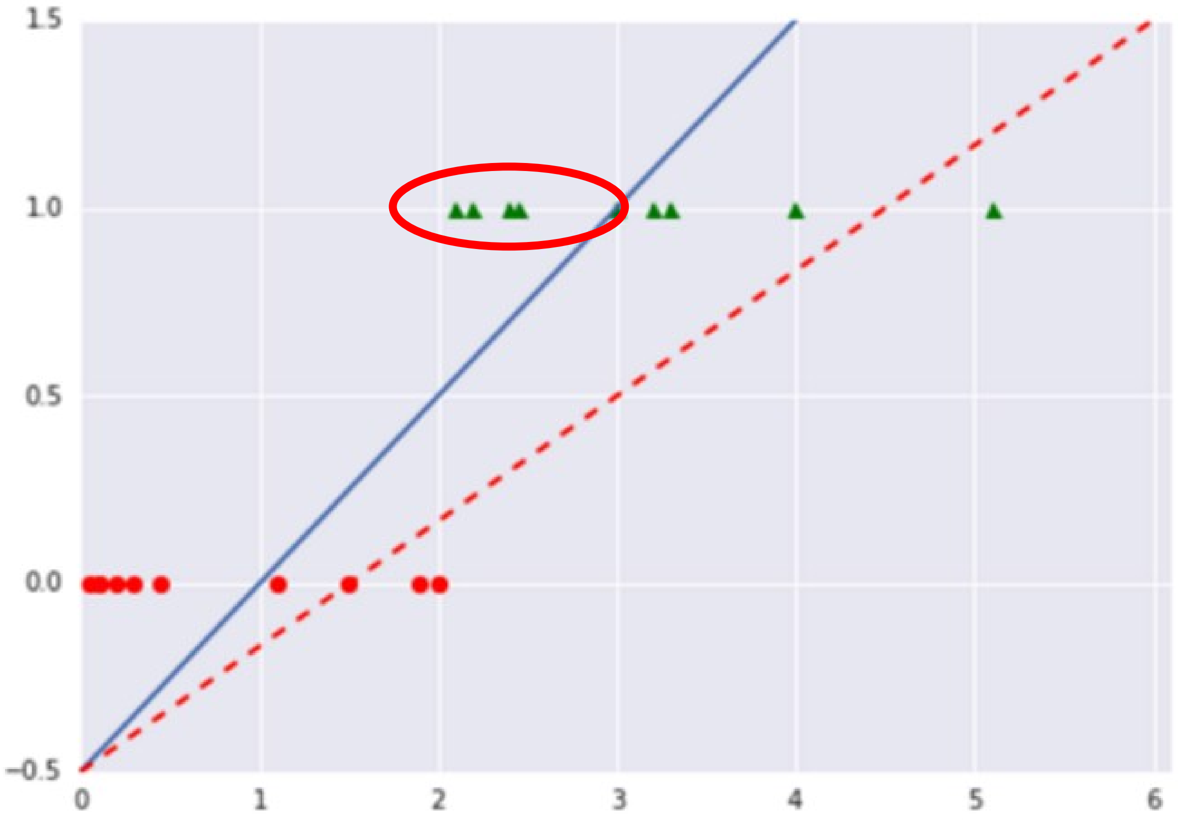
- Linear Regression으로 해봅시다
  - Regression 예측값이 0.5 이상이면 양성, 0.5 이하면 음성으로 판별



Binary Classification

- 종양의 크기가 매우 큰 data(outlier)가 추가된 경우

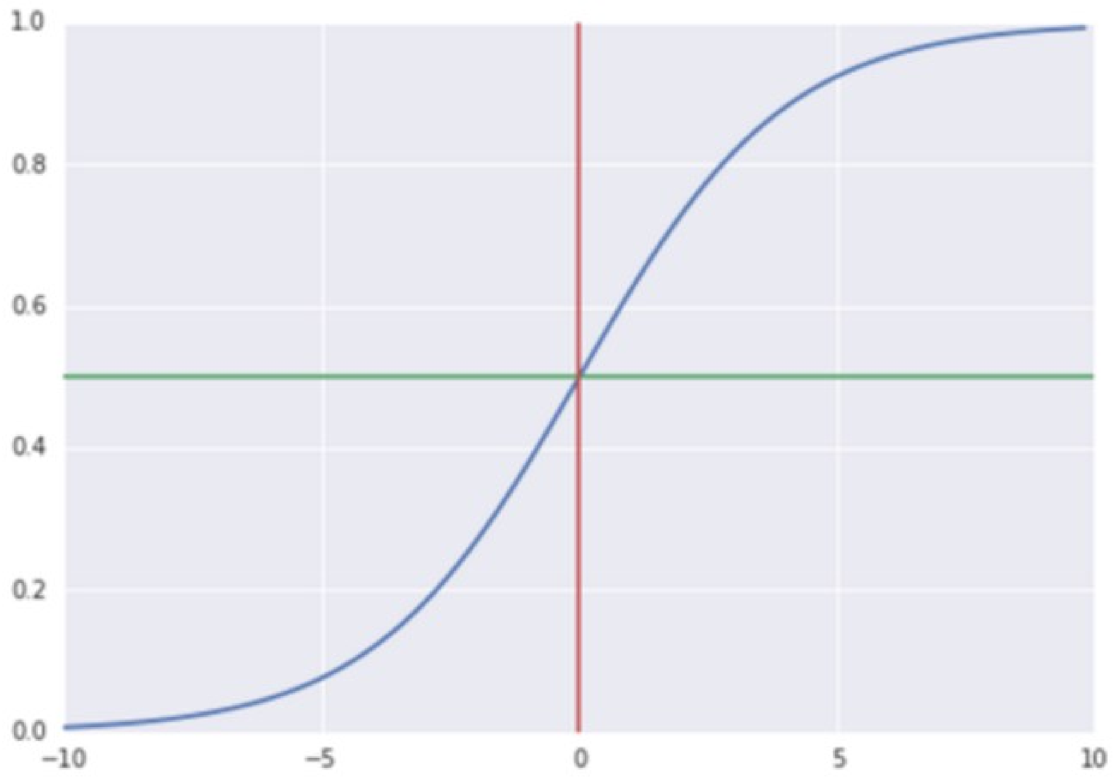




Binary Classification

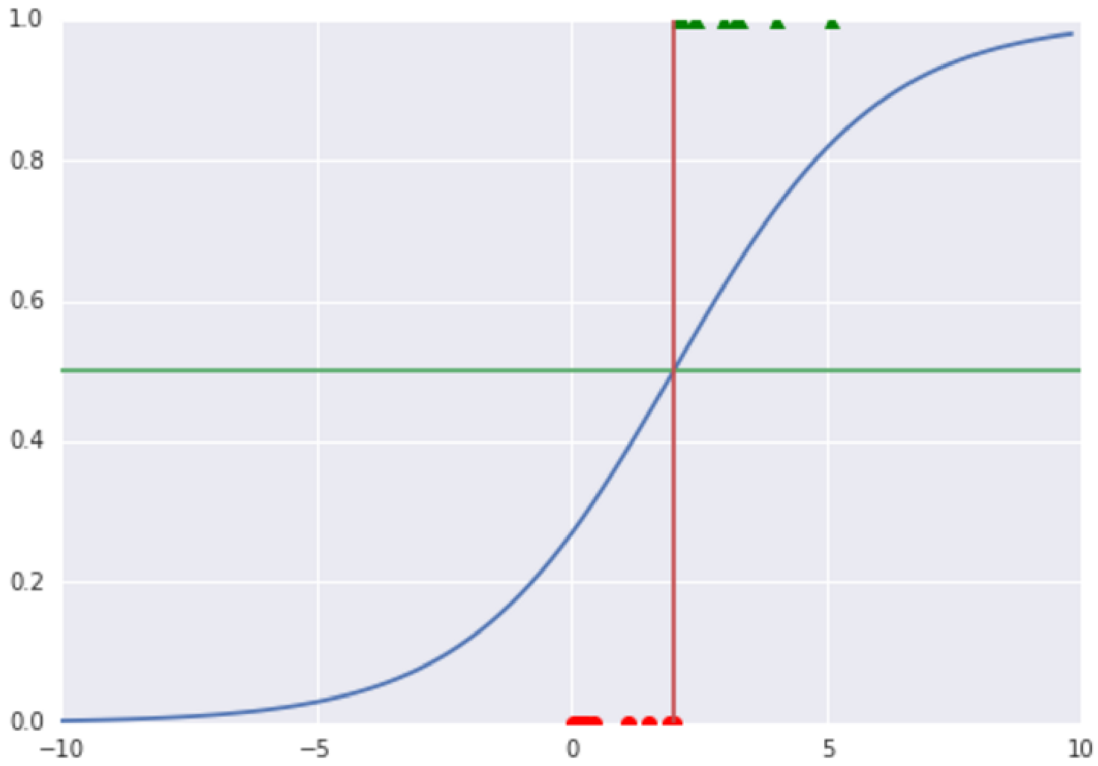
- 아주 크거나 아주 작은 data에 영향을 많이 받지 않았으면 좋겠다
- Binary classification에 맞게 0에서 1사이 값으로 나오면 좋겠다

→ Sigmoid 를 써보자



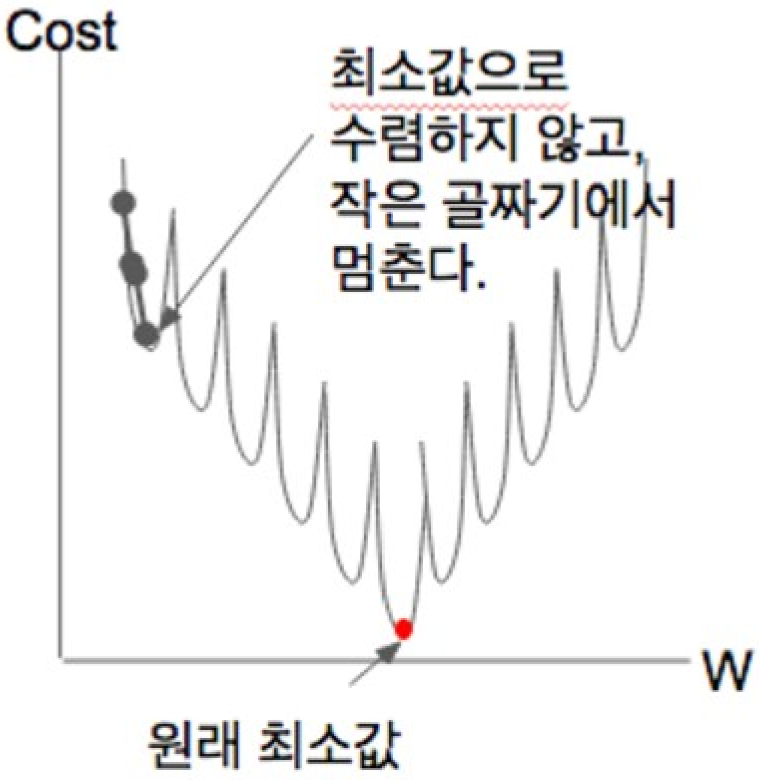
Logistic Regression

- Linear Regression 식에 Sigmoid 함수를 통과시킨 것(odds의 logit을 linear regression)
  - $y = \sigma(w^T x + b) = \dfrac{1}{1+ e^{-(w^T x+b)}}$



Logistic Regression

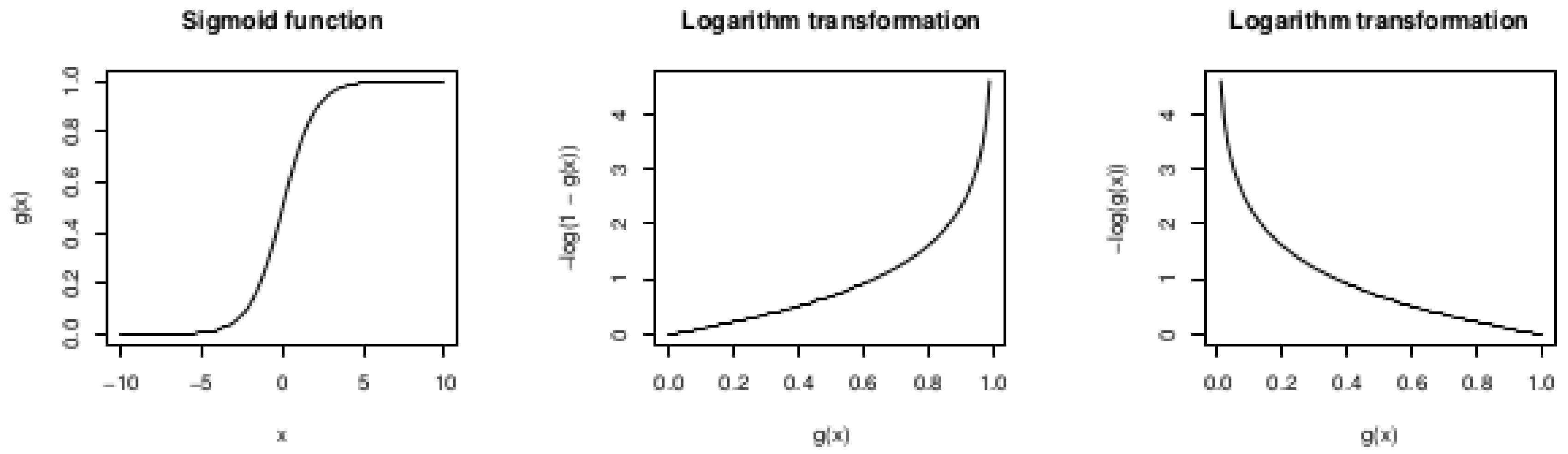
- Loss function으로 MSE(Mean Squared Error)를 쓰면?
  - Linear regression에서는 convex function이었지만 Logistic Regression에서는...



Logistic Regression

- 새로운 Cost(Loss) function을 정의(maximum likelihood estimation)

$$cost(W) = -\frac{1}{m}\sum ylog(H(x)) + (1-y)log(1-H(x))$$



(a) *Sigmoid function.* **(b)** *Cost for* $y = 0$. **(c)** *Cost for* $y = 1$.

Figure B.1: *Logarithmic transformation of the sigmoid function.*