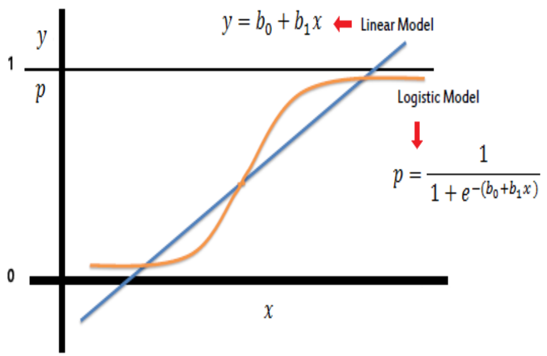
Linear Regression &

Logistic Regression



Fast Campus
Start Deep Learning with TensorFlow

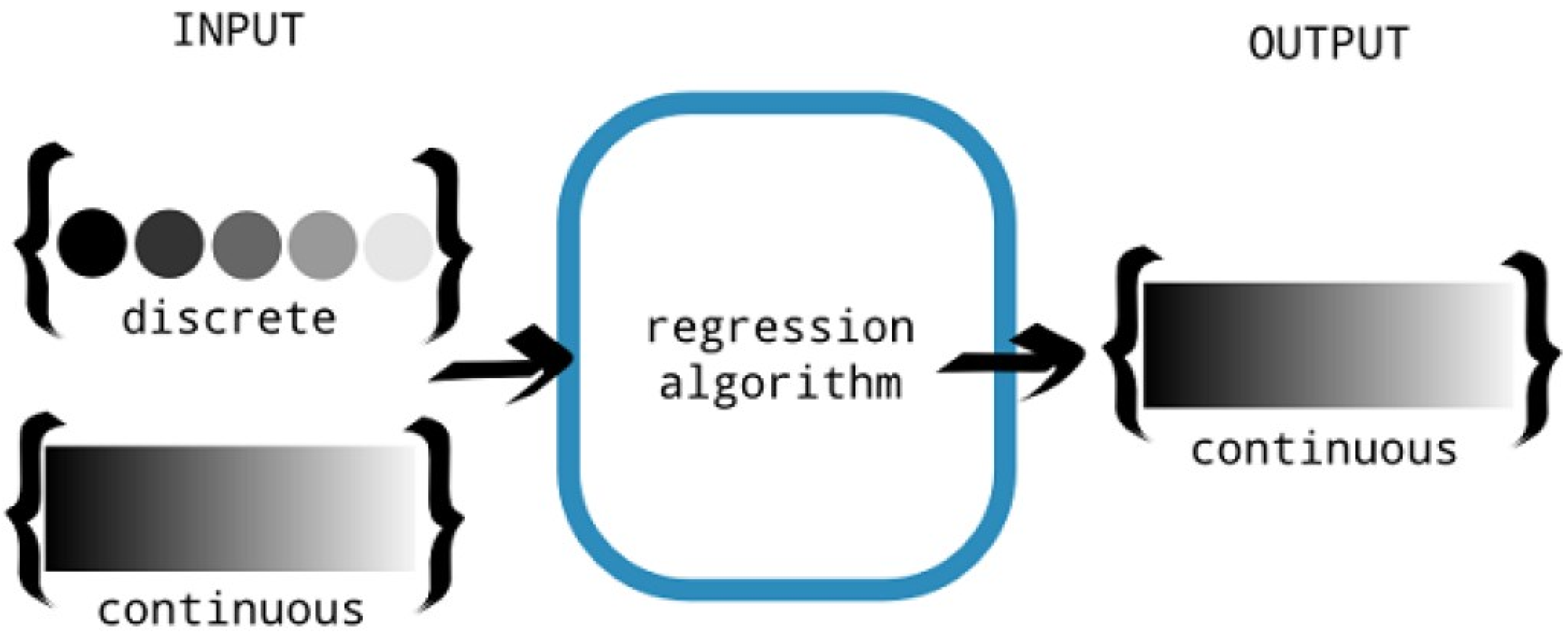
Let's Go to the Deep Learning World!!

쉬운 것부터 시작해봅시다

초등학교 6학년 수학 – 정비례

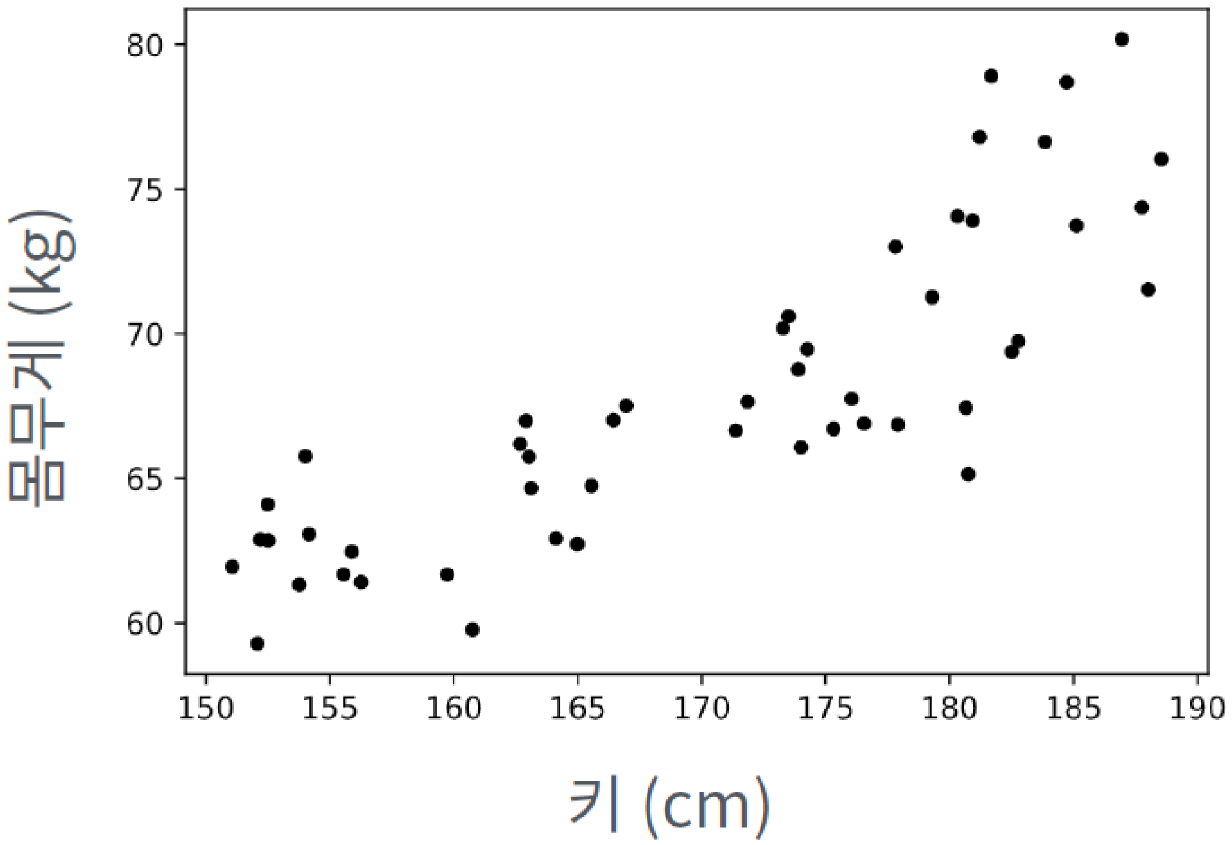
닭 수 (마리)	1	2	3	4	5	……
다리 수 (개)	2	4	6	8	10	……

Regression?


INPUT
OUTPUT
discrete
continuous
regression
algorithm
continuous

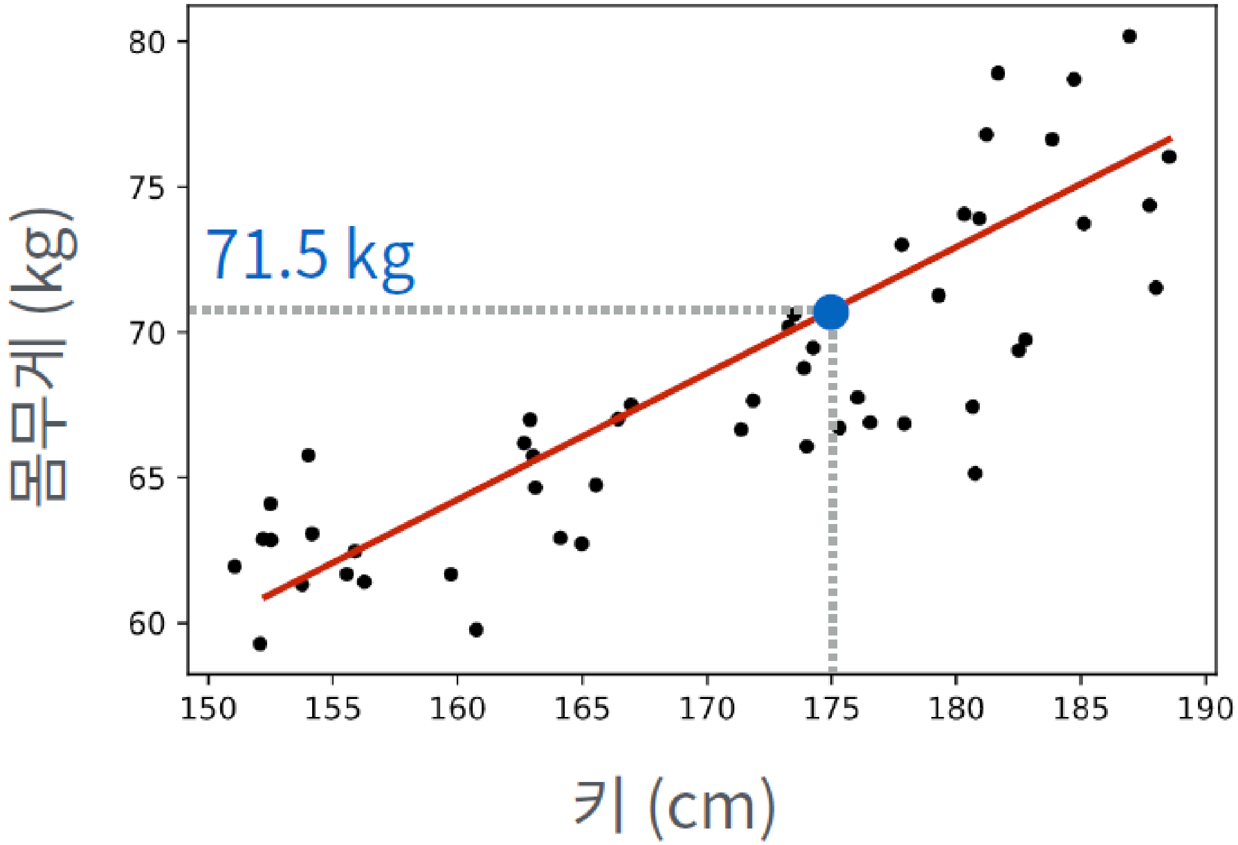
Linear Regression

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



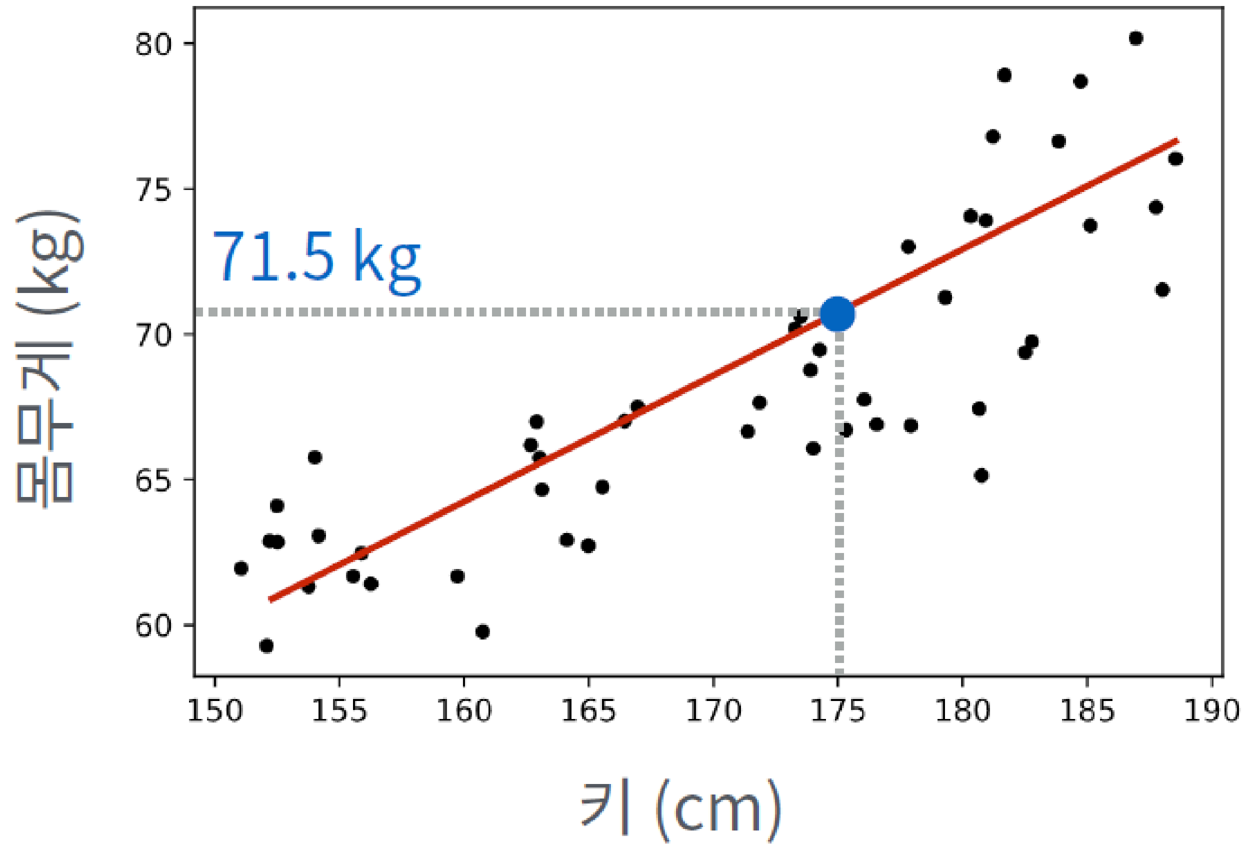
Linear Regression

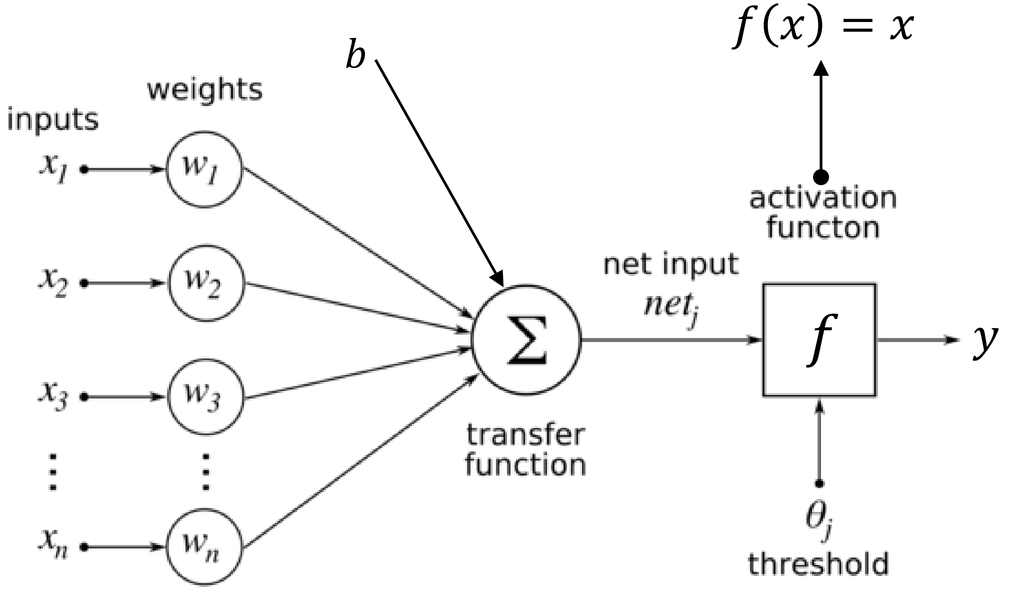
- 어느 학교 학생들의 신체검사 자료
- 새로 전학온 학생 A의 키가 175cm일 때 예상 몸무게는?



Linear Regression

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





<Perceptron>

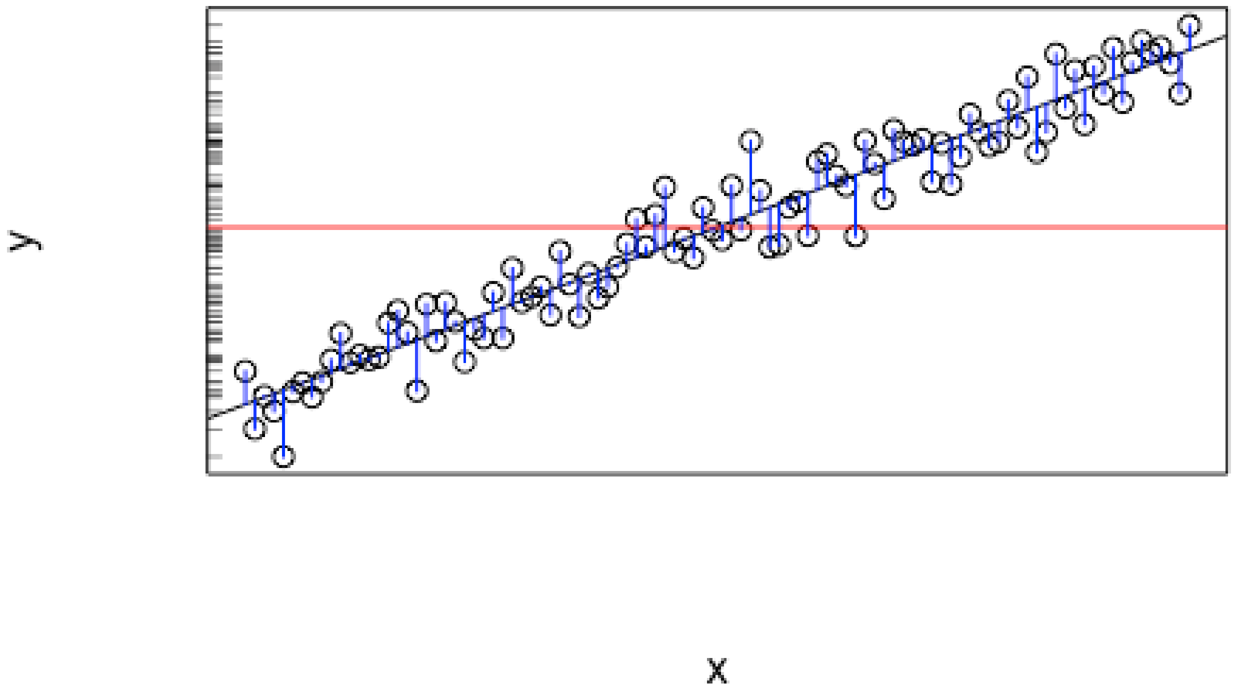
$$y = f(\mathbf{wx} + \mathrm{b})$$

$$\boldsymbol{w} = [w_1\ w_2\ w_3\ \dots\ w_n]$$

$$\boldsymbol{x} = [x_1\ x_2\ x_3\ \dots\ x_n]^T$$

Linear Regression

- 잘 예측했는지 측정할 척도(metric)가 필요함



$$y^* = wx + b \text{ (예측값)}$$

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

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

Linear Regression

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

약간의 수학을(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

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

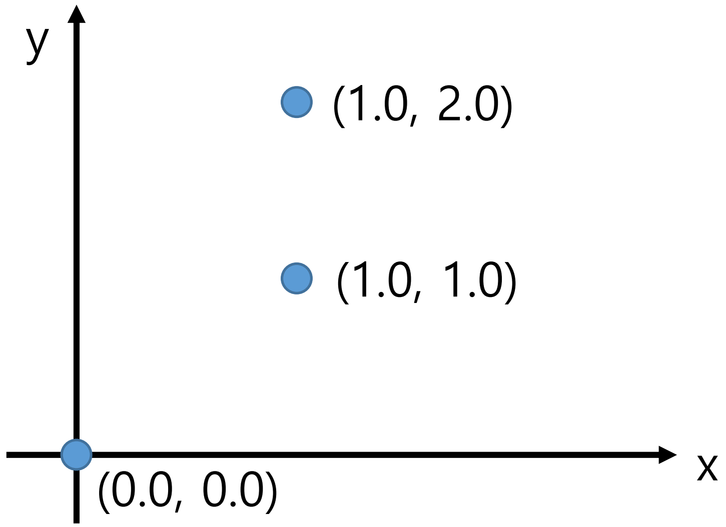
$$\frac{\delta L}{\delta w} = \frac{\delta \sum_i (y_i - wx_i - b)^2}{\delta w}$$

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

$$= 0$$

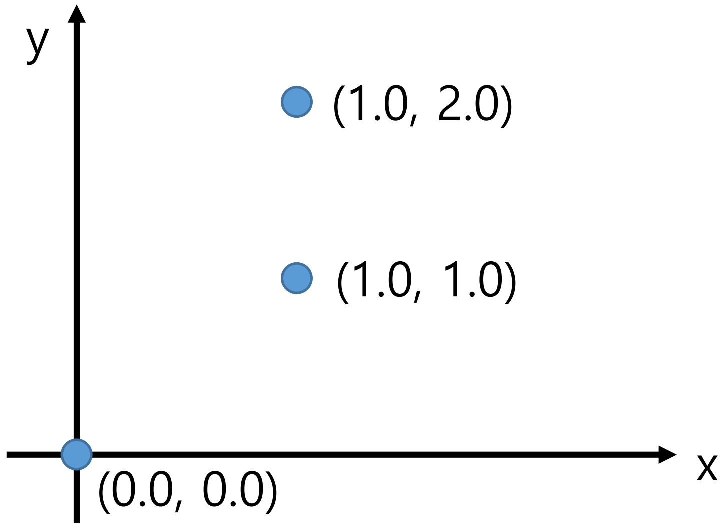
Example

- Find the linear function(f) that best describes the given data
  - $H(x, w_0, w_1) = w_1 x + w_0$



Example

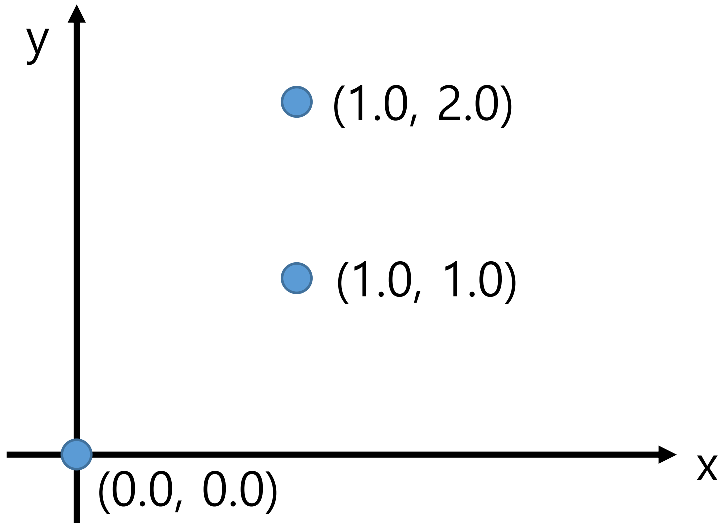
- $H(0, w_0, w_1) \approx 0.0$
- $H(1, w_0, w_1) \approx 1.0$
- $H(1, w_0, w_1) \approx 1.0$



Example

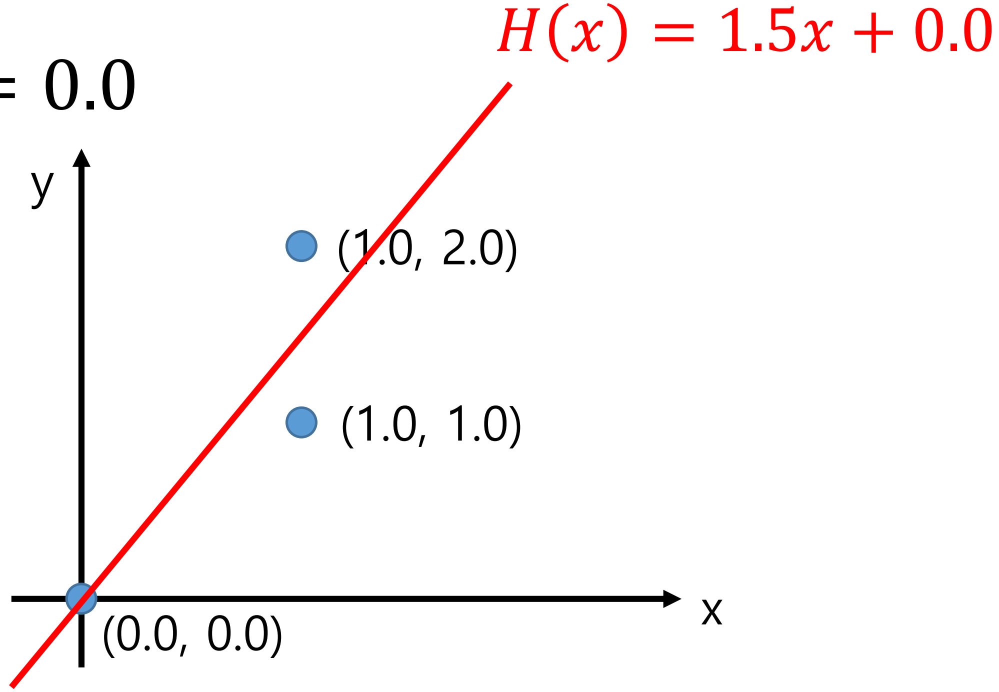
- $L = \sum_i (y_i - w_1 x_i - w_0)^2$

$$= (0.0 - w_1 \cdot 0.0 - w_0)^2 + (1.0 - w_1 \cdot 1.0 - w_0)^2 + (2.0 - w_1 \cdot 1.0 - w_0)^2$$
$$= 2{w_1}^2 + 3{w_0}^2 - 6w_1 - 6w_0 + 4w_1 w_0 + 5$$



Example

- $\dfrac{\partial L}{\partial w_1} = 4w_1 + 4w_0 - 6 = 0$
- $\dfrac{\partial L}{\partial w_0} = 4w_1 + 6w_0 - 6 = 0$
- $\therefore\ w_1 = 1.5, w_0 = 0.0$



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

$$\begin{aligned}
\mathbf{w}_{\text{lin}} &= \operatorname*{argmin}_{\mathbf{w}\in\mathbb{R}^{d+1}} \mathrm{E}_{\text{in}}(\mathbf{w}) \\
&= \operatorname*{argmin}_{\mathbf{w}\in\mathbb{R}^{d+1}} \frac{1}{N}||X\mathbf{w}-\mathbf{y}||^2 \\
&= \operatorname*{argmin}_{\mathbf{w}\in\mathbb{R}^{d+1}} \frac{1}{N}(\mathbf{w}^\top X^\top X\mathbf{w} - 2\mathbf{w}^\top X^\top \mathbf{y} + \mathbf{y}^\top \mathbf{y})
\end{aligned}$$

Multi Variable Linear Regression

- Find w that minimize $E_{in}(w)$ by requiring

$$\nabla \mathrm{E}_{\mathrm{in}}(\mathbf{w}) = \mathbf{0}$$

- From previous equation

$$\nabla \mathrm{E}_{\mathrm{in}}(\mathbf{w}) = \frac{2}{N}(X^\top X\mathbf{w} - X^\top \mathbf{y})$$

$$\begin{aligned} \mathbf{w}_{\mathrm{lin}} &= \underset{\mathbf{w}\in\mathbb{R}^{d+1}}{\operatorname{argmin}} \mathrm{E}_{\mathrm{in}}(\mathbf{w}) \\ &= \underset{\mathbf{w}\in\mathbb{R}^{d+1}}{\operatorname{argmin}} \frac{1}{N}||X\mathbf{w} - \mathbf{y}||^2 \\ &= \underset{\mathbf{w}\in\mathbb{R}^{d+1}}{\operatorname{argmin}} \frac{1}{N}(\mathbf{w}^\top X^\top X\mathbf{w} - 2\mathbf{w}^\top X^\top \mathbf{y} + \mathbf{y}^\top \mathbf{y}) \end{aligned}$$

Multi Variable Linear Regression

- Two scenarios
  - If $X^T X$ is invertible

    $$\boldsymbol{w} = (X^T X)^{-1} X^T$$

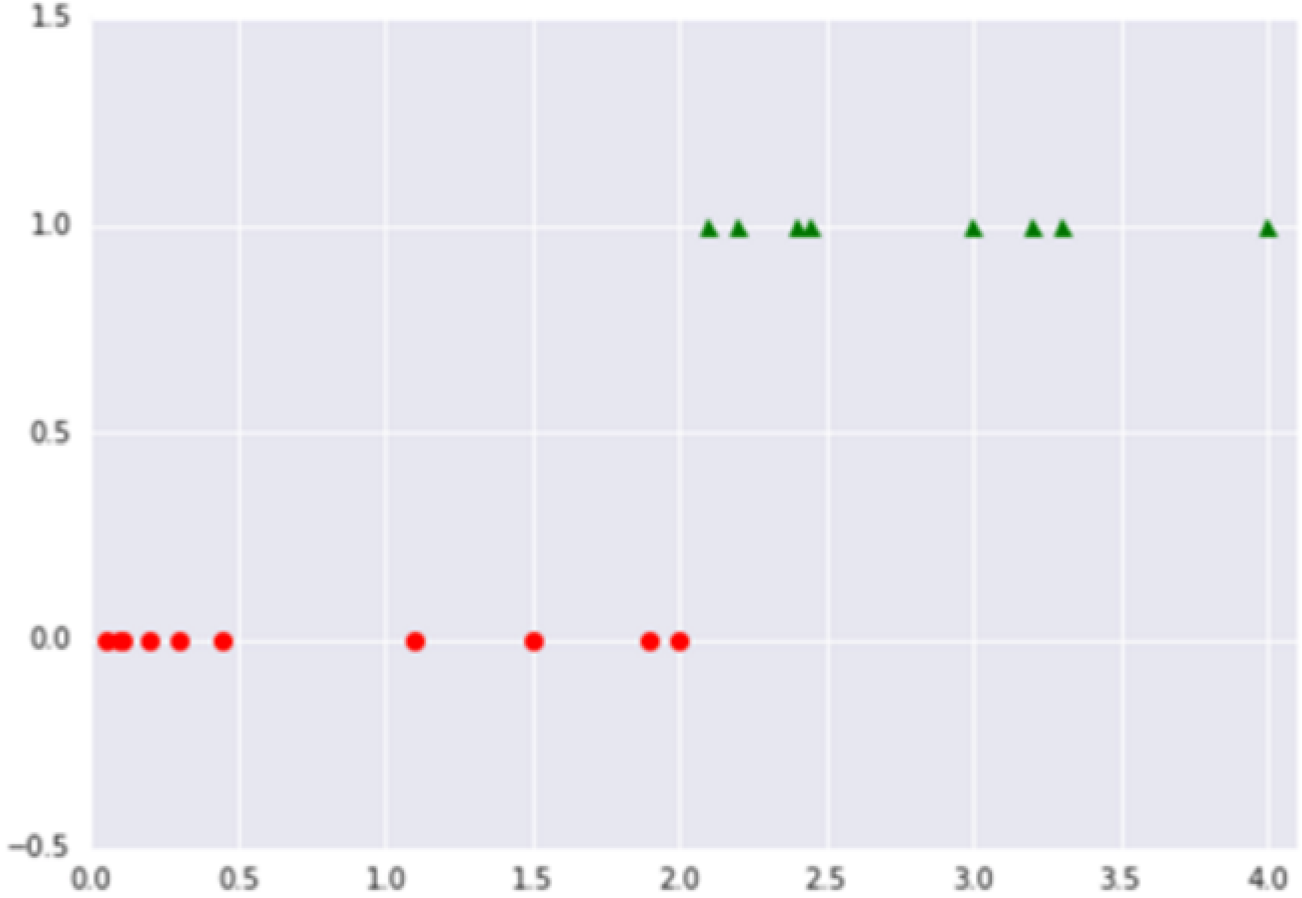
  - If $X^T X$ is not invertible

    Pseudo-inverse defined, but no unique solution

Classification도 할 수 있지 않을까요?

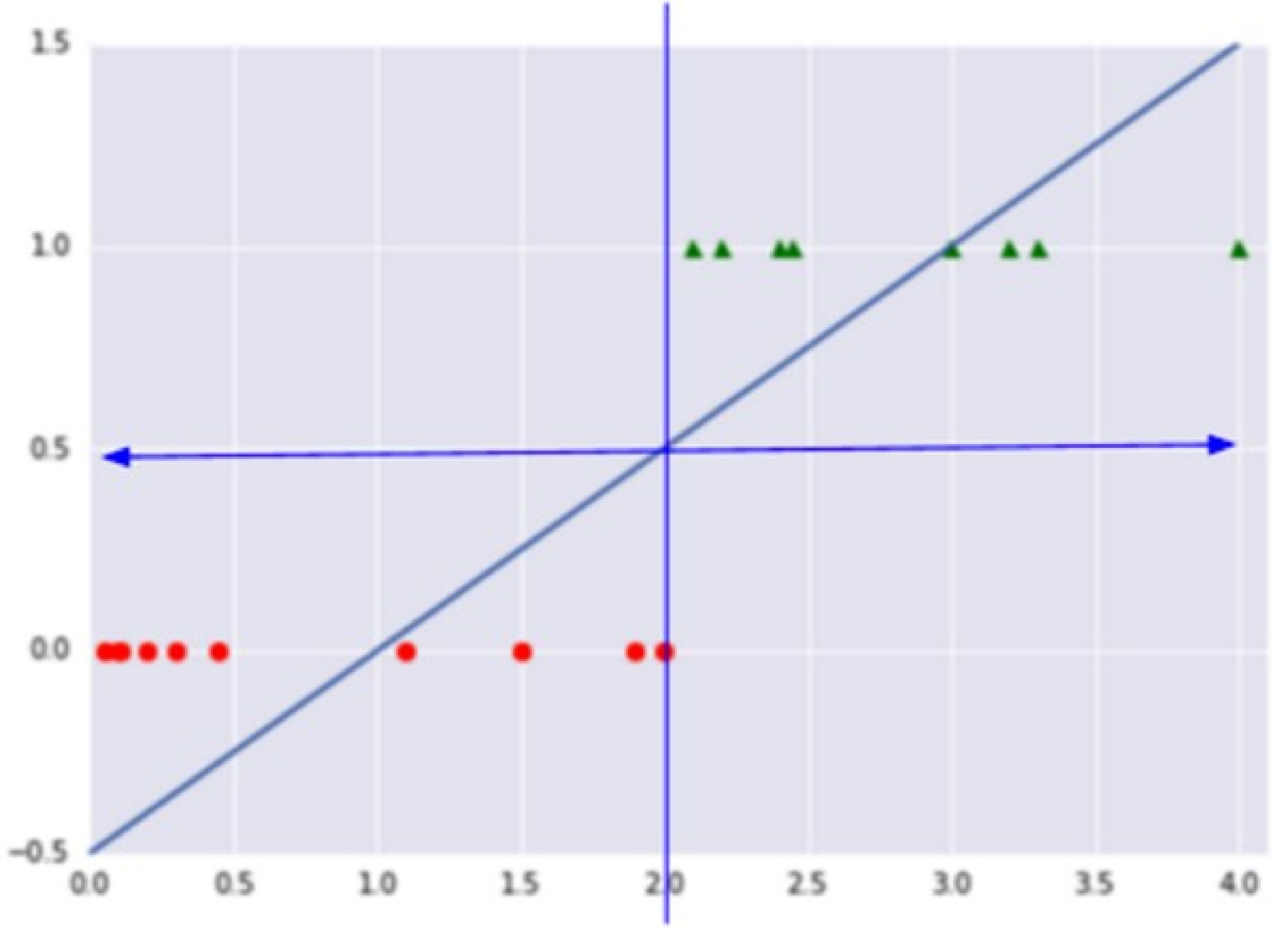
Binary Classification

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



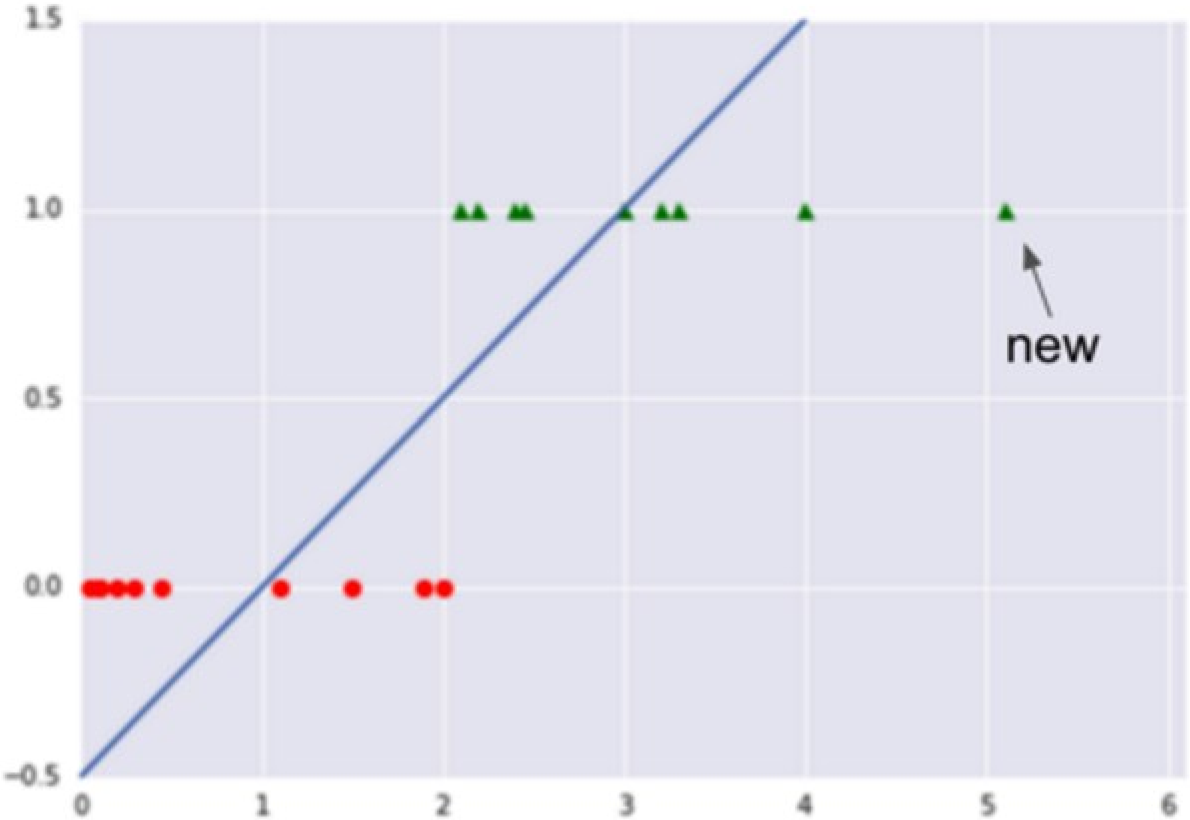
Binary Classification

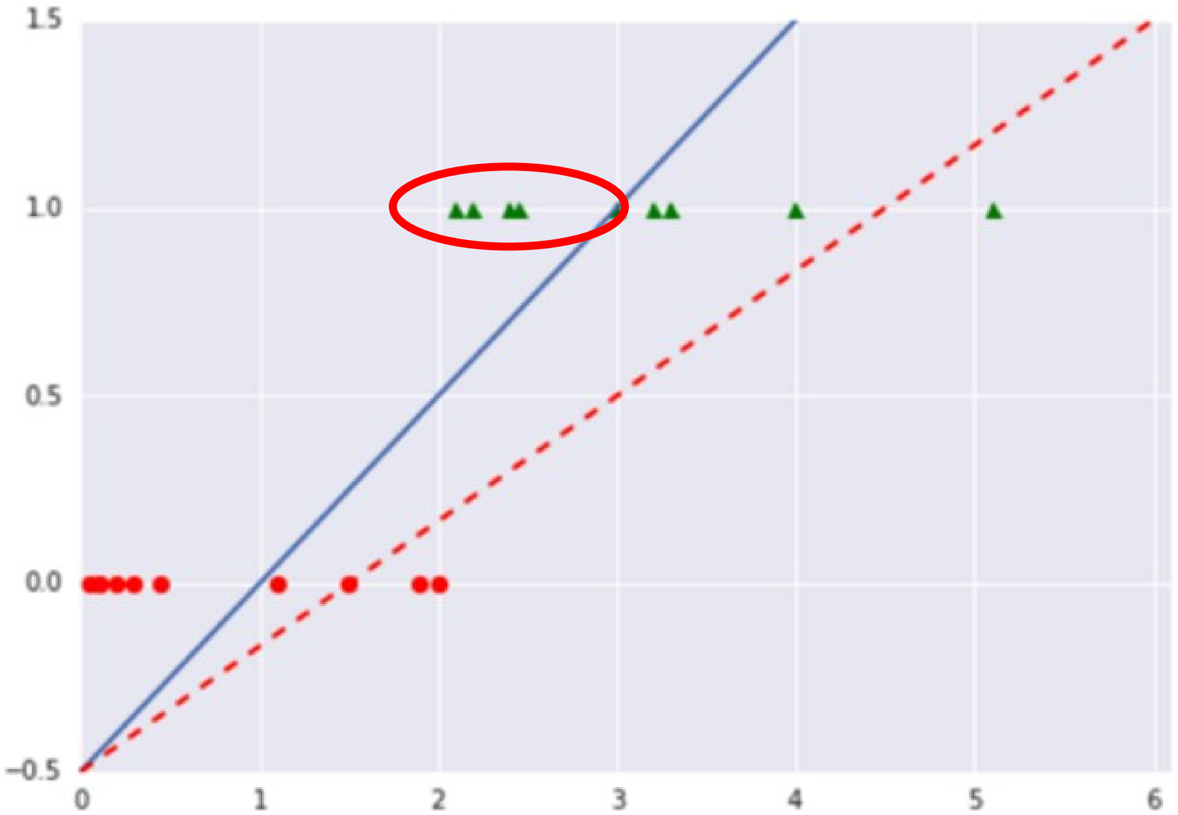
- Linear Regression으로 해봅시다
  - Regression 예측값이 0.5 이상이면 양성, 0.5 이하면 음성으로 판별



Binary Classification

- 종양의 크기가 매우 큰 data(outlier)가 추가된 경우

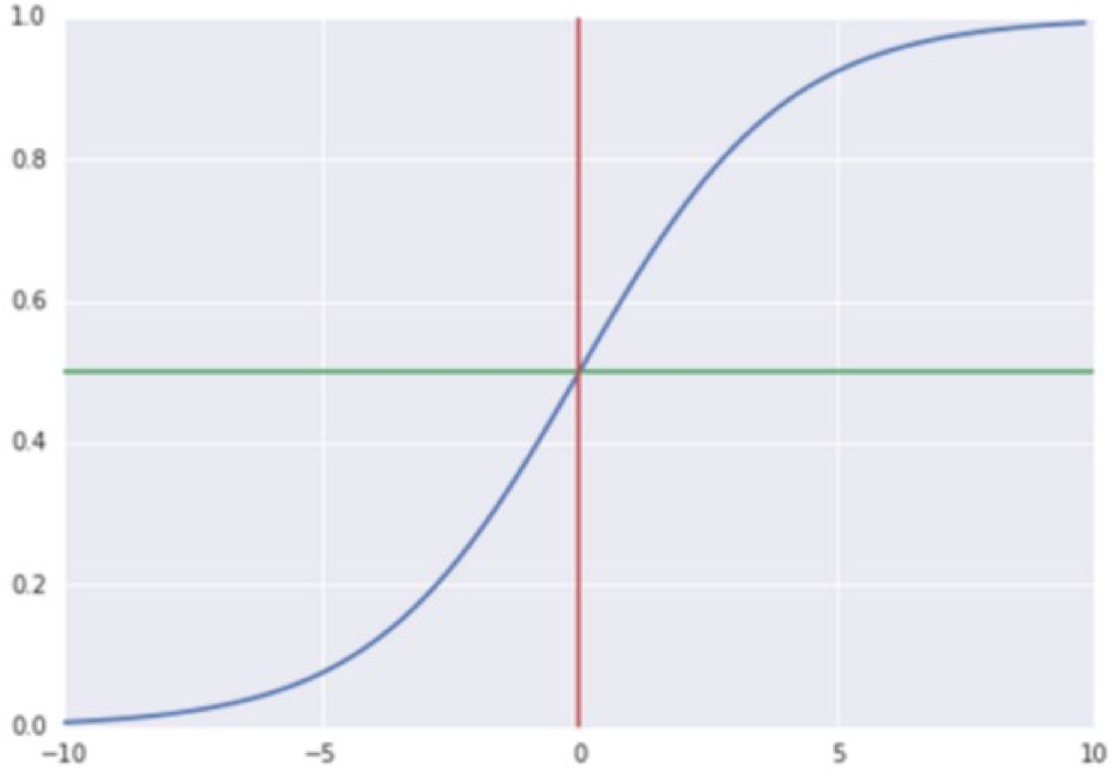




Binary Classification

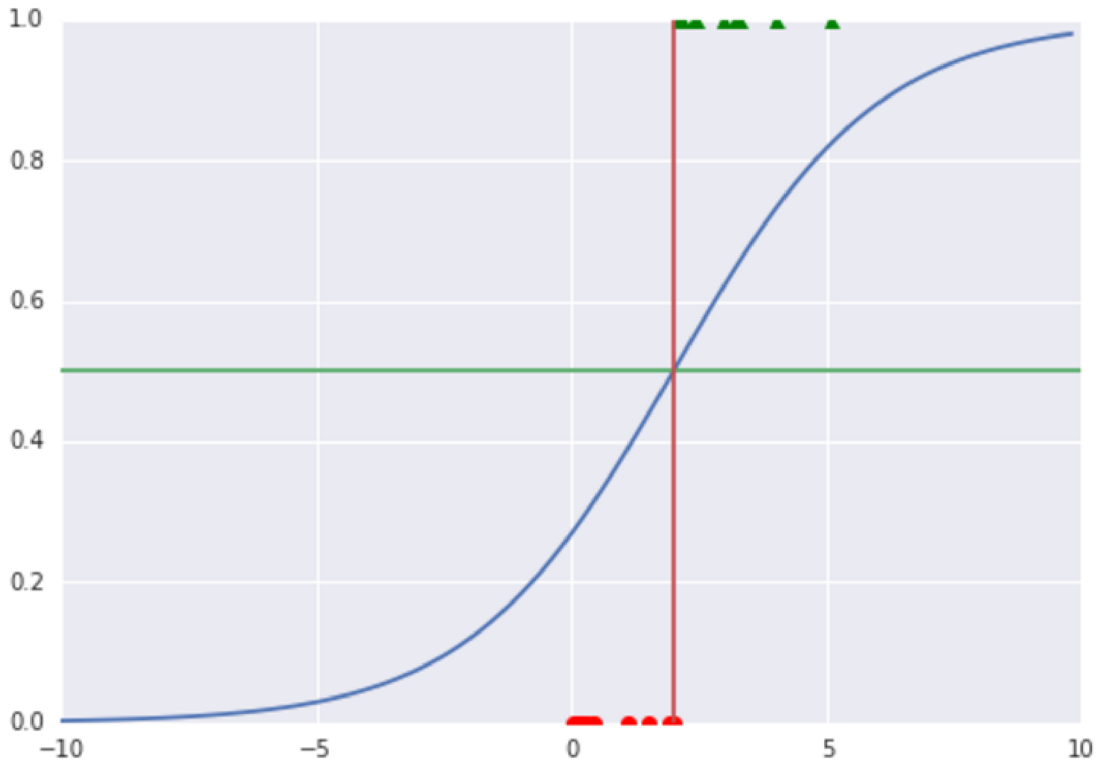
- 아주 크거나 아주 작은 data에 영향을 많이 받지 않았으면 좋겠다
- Binary classification에 맞게 0에서 1사이 값으로 나오면 좋겠다

→ Sigmoid 를 써보자



Logistic Regression

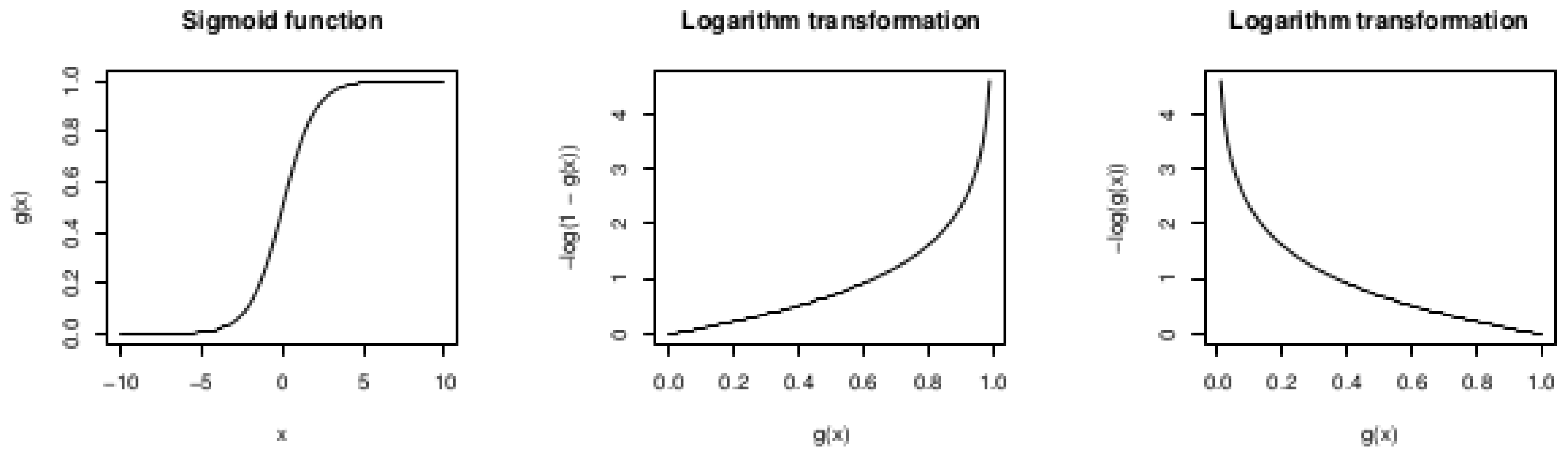
- Linear Regression 식에 Sigmoid 함수를 통과시킨 것
  - $H(x) = \sigma(\boldsymbol{w}^T\boldsymbol{x} + b) = \dfrac{1}{1+e^{-(\boldsymbol{w}^T\boldsymbol{x}+b)}} = P(y|\boldsymbol{x})$



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.*

Likelihood

- Likelihood – 2개의 class(y = +1, y = -1)의 경우,

$$P(y|\mathbf{x}) = \begin{cases} h(\mathbf{x}) & \text{for } y = +1 \\ 1 - h(\mathbf{x}) & \text{for } y = -1 \end{cases}$$

$$h(x) = \sigma(\boldsymbol{w}^T \boldsymbol{x} + b)$$

$$P(y|\mathbf{x}) = h(\mathbf{x})^{[\![y=+1]\!]}(1 - h(\mathbf{x}))^{[\![y=-1]\!]}$$

- Likelihood of data $(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_N, y_N)$

$$\prod_{n=1}^{N} P(y_n|\mathbf{x}_n) = \prod_{n=1}^{N} h(\mathbf{x}_n)^{[\![y_n=+1]\!]}(1 - h(\mathbf{x}_n))^{[\![y_n=-1]\!]}$$

Negative Log-Likelihood(NLL)

- Maximize likelihood = Minimize negative log-likelihood(NLL)

$$NLL(\mathbf{w}) \propto -\frac{1}{N}\log\left\{\prod_{n=1}^{N} P(y_n|\mathbf{x}_n)\right\}$$

$$= -\frac{1}{N}\log\left\{\prod_{n=1}^{N} h(\mathbf{x}_n)^{[\![y_n=+1]\!]}(1-h(\mathbf{x}_n))^{[\![y_n=-1]\!]}\right\}$$

$$= \boxed{\frac{1}{N}\sum_{n=1}^{N}\left\{[\![y_n=+1]\!]\log\frac{1}{h(\mathbf{x}_n)} + [\![y_n=-1]\!]\log\frac{1}{1-h(\mathbf{x}_n)}\right\}}$$

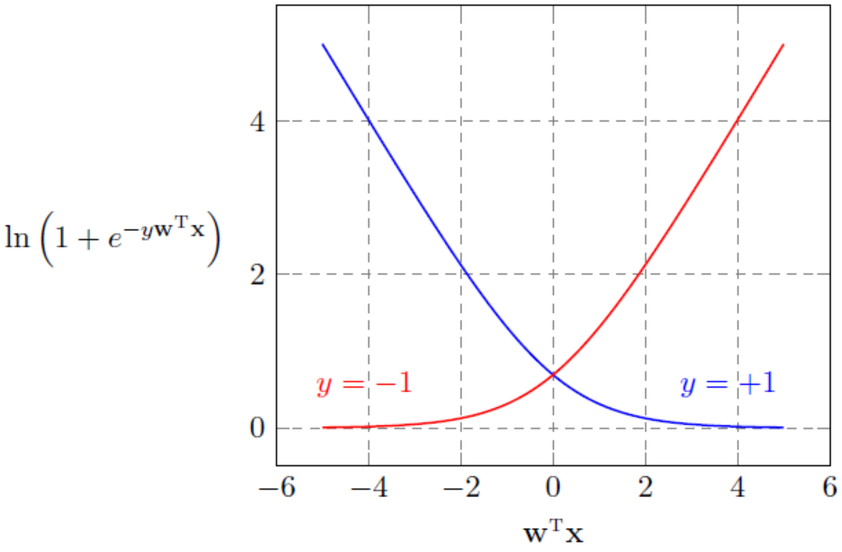
Cross-Entropy!

Minimizing NLL

- Minimize $$-\frac{1}{N}\log\left\{\prod_{n=1}^{N} P(y_n|\mathbf{x}_n)\right\} = \frac{1}{N}\sum_{n=1}^{N}\log\frac{1}{P(y_n\,|\,\mathbf{x}_n)}$$
- We can define loss(error) function as below

$$\mathrm{E_{in}}(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N}\log\left(1+e^{-y_n\mathbf{w}^{\top}\mathbf{x}_n}\right)$$

$$\mathbf{e}(h(\mathbf{x}_n), y_n) = \ln\left(1+e^{-y_n\mathbf{w}^{\top}\mathbf{x}_n}\right)$$



Minimizing NLL

- Unfortunately, not easy to manipulate analytically

$$\nabla \mathrm{E}_{\mathrm{in}}(\mathbf{w}) = -\frac{1}{N}\sum_{n=1}^{N}\frac{y_n \mathbf{x}_n}{1+e^{y_n \mathbf{w}^{\mathrm{T}}\mathbf{x}_n}}$$

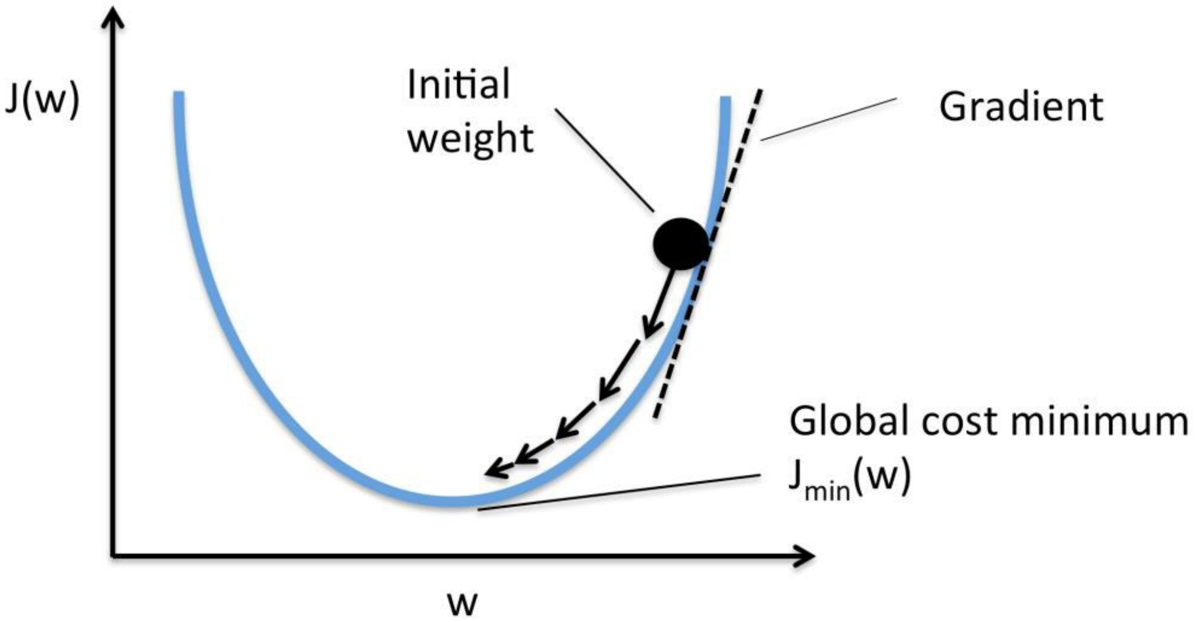
$$= \frac{1}{N}\sum_{n=1}^{N} -y_n \mathbf{x}_n \theta(-y_n \mathbf{w}^{\mathrm{T}}\mathbf{x}_n)$$

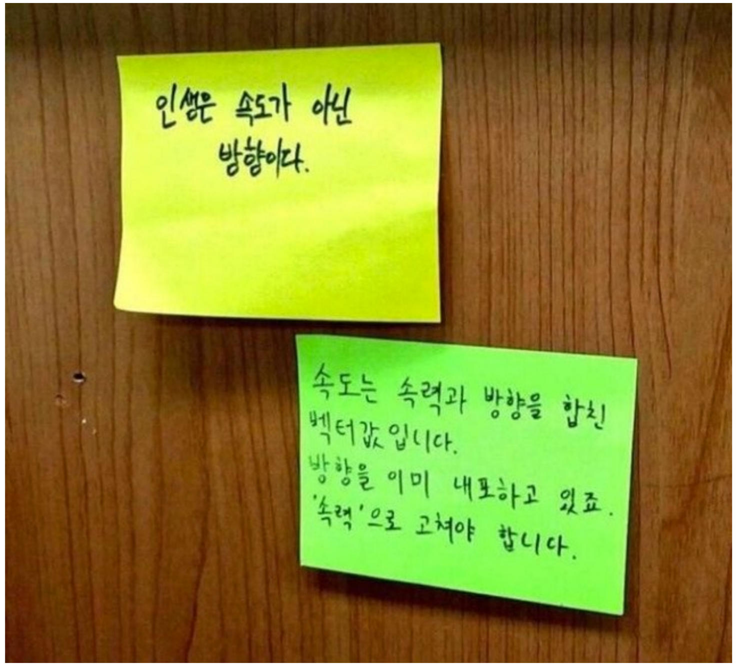
- We need iterative optimization

Gradient Descent

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

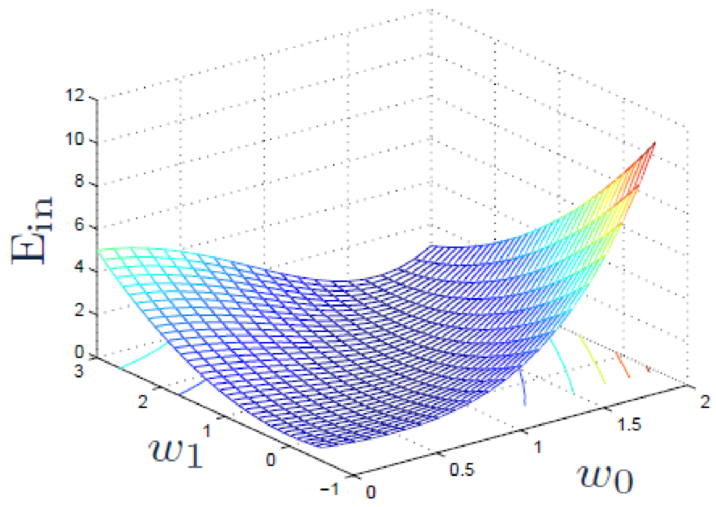
- 방향 : 그 지점에서의 – gradient
- 속력(보폭) : learning rate(η)

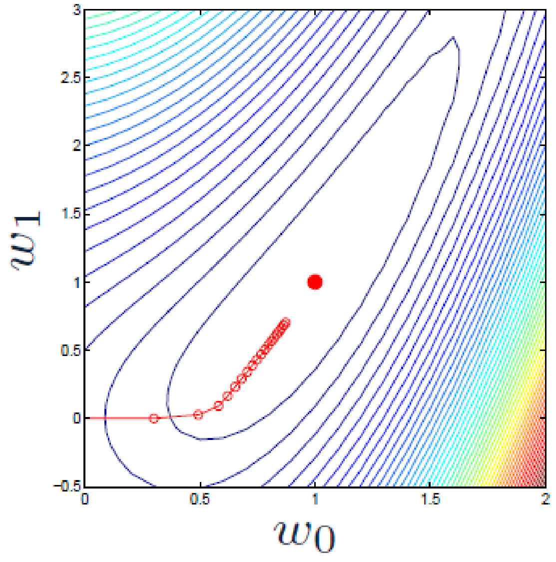


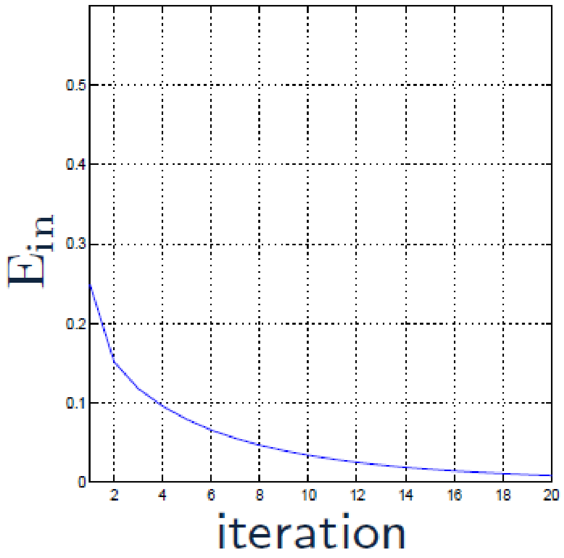


Gradient Decent Example

- **Global minimum 0 at (1,1)**

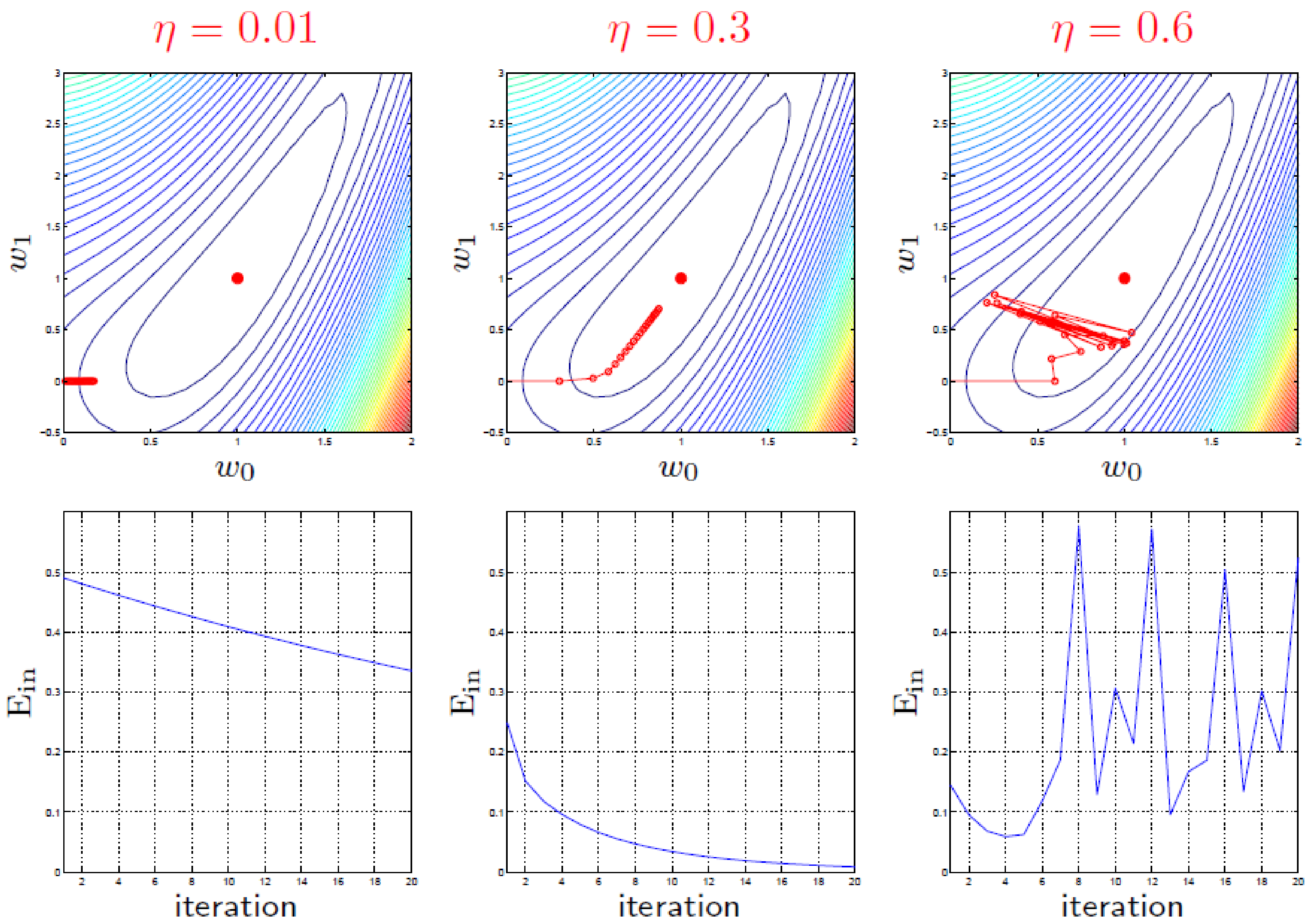






- **Start at (0,0)**
  - # iterations (steps) $= 20$
  - step size $\eta = 0.3$

Learning Rate

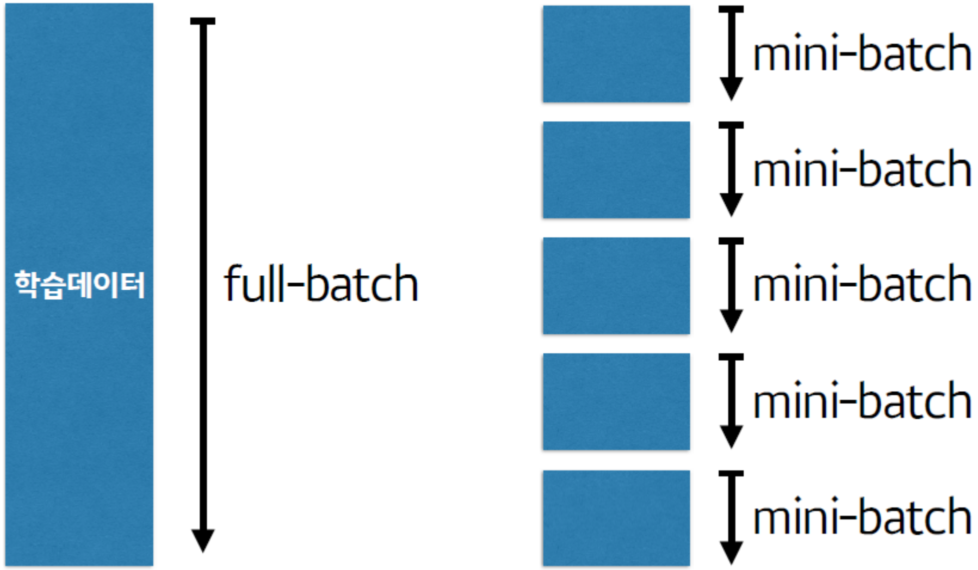


Stochastic Gradient Descent, Mini-batch Training

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

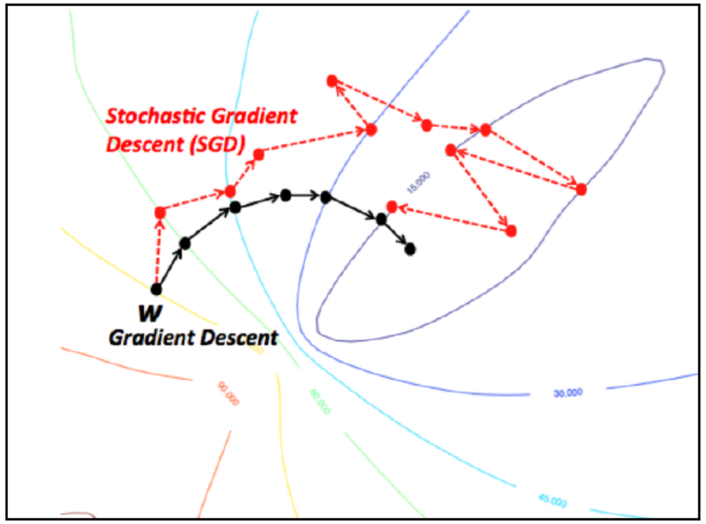
Gradient Decent

Stochastic Gradient Decent

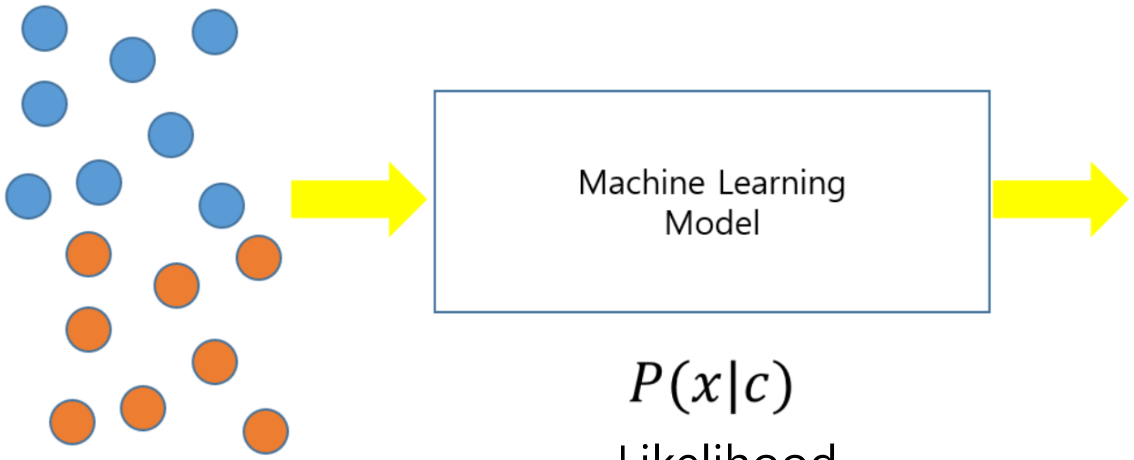


전부다 읽고나서
최적의 1스텝 간다.

작은 토막마다
일단 1스텝간다.



Appendix: Bayes Rule



$P(c)$
Prior

$P(x|c)$
Likelihood

$P(c|x)$
Posterior

$P(c|x) \times P(x) = P(x|c) \times P(c)$
Posterior Evidence Likelihood Pri

or

x: data
c: classification

$P(c|x;\theta)$: Posterior
→data x 가 들어왔을 경우 어떤 class c 에 mapping 될 확률
즉, 주어진 데이터를 보고 최적의 parameter set (e.g., $\theta = \{w, b\}$)을 찾아 내는 것

$P(x|c)$: Likelihood → class c 에 속해있는 data x 의 분포
$P(x)$:Evidence → data x 자체의 분포
$P(c)$:prior → class c 의 분포가 어떨 것이라는 우리의 사전적인prior 지식

Appendix: Logistic Regression (2-class classification)

$X: Y_1$ if $P(Y_1|X)>P(Y_2|X)$, $X: Y_1$ if $P(Y_1|X)>P(Y_2|X)$

$$P(Y_1|X) = \frac{P(X|Y_1)P(Y_1)}{P(X)}$$

$$P(Y_2|X) = \frac{P(X|Y_2)P(Y_2)}{P(X)}$$

$$P(X) = P(X \cap Y_1) + P(X \cap Y_2)$$

$$= P(X|Y_1)P(Y_1) + P(X|Y_2)P(Y_2)$$

Let $a_k = ln\left(P(X|Y_k)P(Y_k)\right), k = 1, 2$

: Log odd, Decision boundary function

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)} \propto P(x|c) \times P(c)$$

Posterior likelihood Prior

x: data
c: classification

→

$$P(Y_1|X) = \frac{\exp(a_1)}{\exp(a_1) + \exp(a_2)} = \frac{1}{1 + \exp(a_2 - a_1)}$$

→

Let $a = -(a_2 - a_1)$, $\quad P(Y_1|X) = \dfrac{1}{1 + \exp(a)}$

Appendix: Logistic Regression (2-class classification)

$a = -(a_2 - a_1), \qquad a_k = ln\big(P(X|Y_k)P(Y_k)\big), k = 1,2$

when Prior $P(c)$ is constant, and both classes have same covariance matrix with Gaussian distribution,

→

$$a = ln\left(\frac{\exp\left(-\frac{1}{2}(X-\boldsymbol{\mu_1})^T\Sigma^{-1}(X-\boldsymbol{\mu_1})\right)P(Y_1)}{\exp\left(-\frac{1}{2}(X-\boldsymbol{\mu_2})^T\Sigma^{-1}(X-\boldsymbol{\mu_2})\right)P(Y_2)}\right)$$

→

$$a = \underbrace{\boldsymbol{\mu_1}\Sigma^{-1}X - \boldsymbol{\mu_2}\Sigma^{-1}X}_{\boldsymbol{W}^T X} - \underbrace{\frac{1}{2}\boldsymbol{\mu_1}^T\Sigma^{-1}\boldsymbol{\mu_1} + \frac{1}{2}\boldsymbol{\mu_2}^T\Sigma^{-1}\boldsymbol{\mu_2} + \ln\left(\frac{P(Y_1)}{P(Y_2)}\right)}_{\boldsymbol{B}}$$

→

$$a = \boldsymbol{W}^T X + \boldsymbol{B}$$

$$P(Y_1|X) = \frac{1}{1+\exp(a)}$$

: Logistic Regression

Appendix: Likelihood function

어떠한 Parameter를 선택하였을 경우 주어진 Data가 가장 잘 설명이 될까? : Likelihood $P(x|c)$

→

$$a = \boldsymbol{W}^T X + \boldsymbol{B}, \text{Let} \quad \sigma(w^T x) = P(Y_1|X) = \frac{1}{1 + \exp(a)}$$

→

Let

$$y_n = P(t_n = 1|x_n) = P(Y_1|X),$$
$$1 - y_n = 1 - P(t_n = 1|x_n) = P(t_n = 0|x_n) = P(Y_2|X)$$

$$Likelihood = \prod_{n=1}^{N} P(t_n = 1|x_n, w) = \prod_{n=1}^{N} y_n{}^{t_n}(1 - y_n)^{1-t_n}$$

→

$$LogLikelihood = \sum_{n=1}^{N} t_n log(y_n) + (1 - t_n) log(1 - y_n)$$

$$Loss = -LogLikelihood = -\sum_{n=1}^{N} t_n \log(\sigma(w^T x)) + (1 - t_n)\log(1 - \sigma(w^T x))$$

That is Maximizing Likelihood is Minimizing Cross-Entropy