

# 데이터 과학

## L04: Logistic Regression

Kookmin University

# 지도학습 Supervised Learning

훈련 데이터(Training Data)로부터 하나의 함수를 유추해내기 위한 기계 학습(Machine Learning)의 한 방법

## Training Data

[1.2, 3.8, -1.4, ..., 4.1]	→	1.1
[3.2, -1.2, -0.2, ..., 2.1]	→	2.7
[2.8, -1.4, -0.3, ..., 2.3]	→	2.8
[1.2, 3.4, -1.5, ..., 4.2]	→	0.9
[4.2, 2.1, 2.8, ..., -0.5]	→	-0.1
...		
[3.2, 2.2, 2.2, ..., -0.4]	→	-0.2

## Test

[1.3, 3.2, -1.5, ..., 4.1]	→	?
----------------------------	---	---

# 이진 분류 문제 Binary Classification

- 종속 변수  $y$ 가 0 또는 1인 경우의 회귀 분석

## Training Data

```
[1.2, 3.8, -1.4, ..., 4.1] → 0
[3.2, -1.2, -0.2, ..., 2.1] → 0
[2.8, -1.4, -0.3, ..., 2.3] → 1
[1.2, 3.4, -1.5, ..., 4.2] → 0
[4.2, 2.1, 2.8, ..., -0.5] → 1
...
[3.2, 2.2, 2.2, ..., -0.4] → 1
```

## Test

```
[1.3, 3.2, -1.5, ..., 4.1] → ?
```

쉬운 설명을 위해... 단순한 예제.

## Training Data

```
[1.2] → 0
[3.2] → 1
[2.8] → 0
[1.2] → 0
[4.2] → 1
...
[3.2] → 1
```

## Test

```
[1.3] → ?
```

# 이진 분류 문제의 활용

- 이메일 필터링 (스팸메일? 정상메일?)
- 고양이 분류 (이 사진은 고양이 사진인가 아닌가?)
- Instagram feed (이 피드를 보여줄 것인가?)
- 주식 시장 예측 (지금 팔아야 하는가? 사야 하는가?)



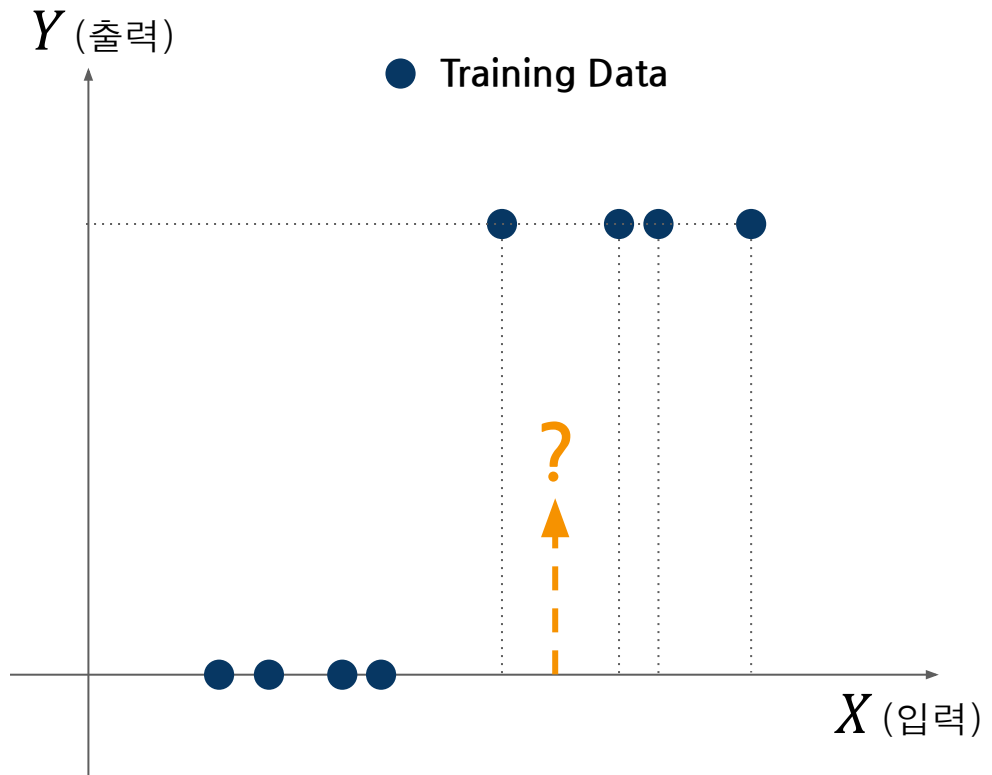
# 이진 분류 문제 Binary Classification

## Training Data

[1.2]	→	0
[3.2]	→	1
[1.6]	→	0
[1.4]	→	0
[4.2]	→	1
...		
[3.3]	→	1

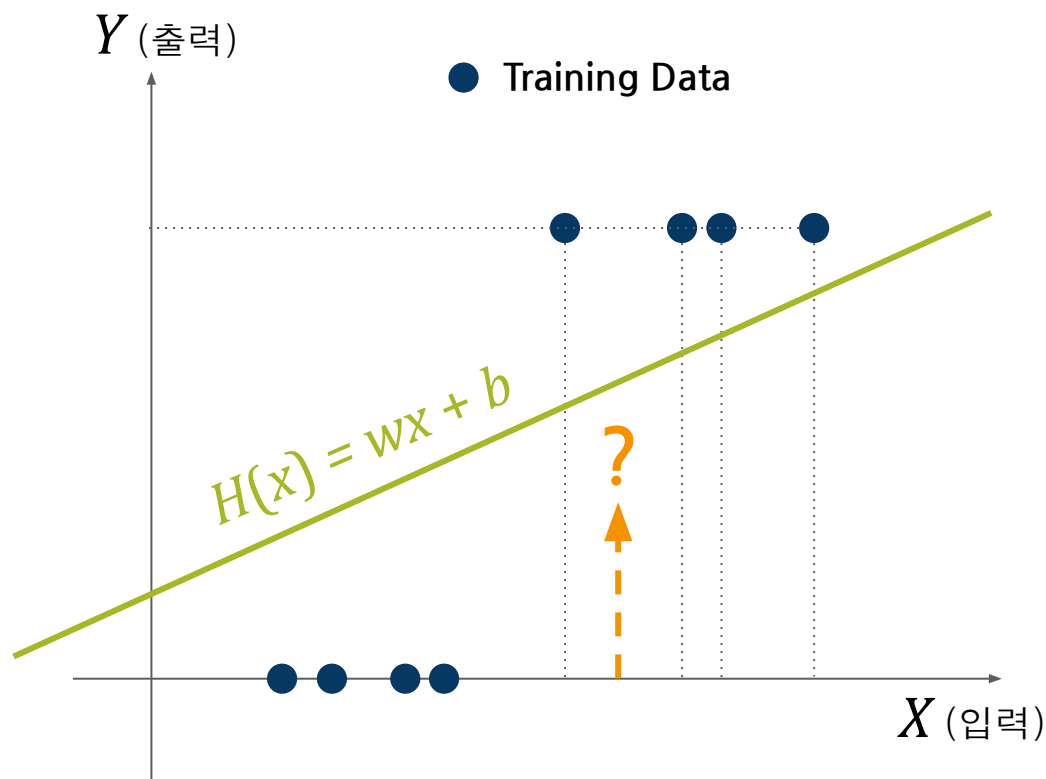
## Test

[3.9]	→	?
-------	---	---



# 분류 문제와 선형 회귀

선형 회귀 분석은 Classification 문제에 잘 동작하지 않는다..!

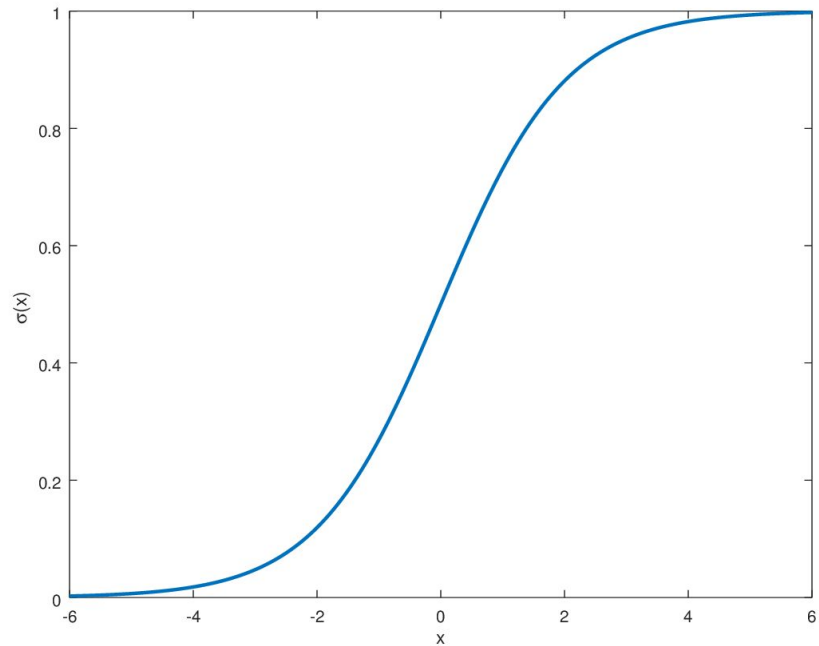


이런 데이터가  
추가된다면??

cost를 최소화 하는 선을 찾는다면 (즉,  $w$ 와  $b$ 를 찾는다면),  
과연 그 선으로 출력값들을 잘 예측할 수 있을까...?

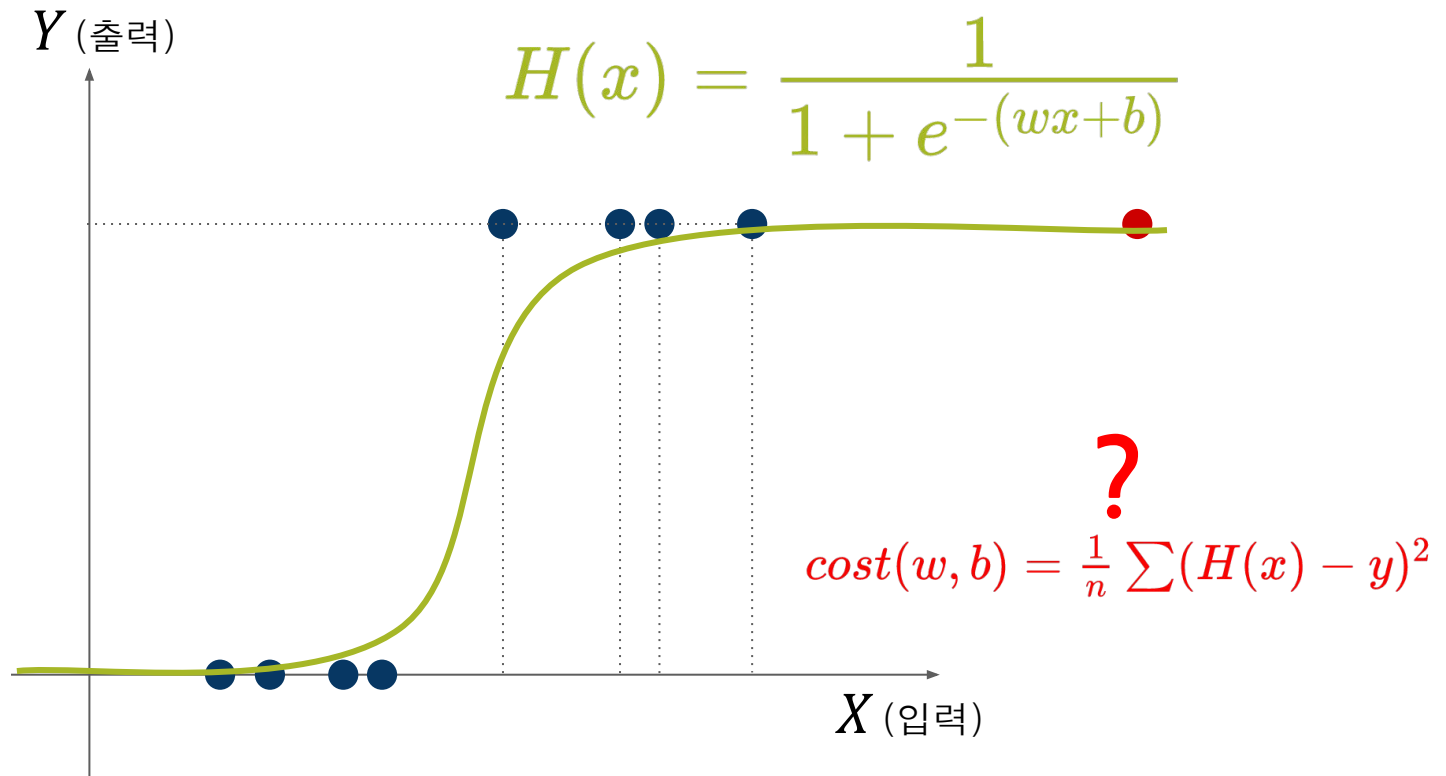
# Sigmoid 함수 (로지스틱 함수)

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



# Logistic Regression

선 대신에 sigmoid 함수를 가설로써 활용

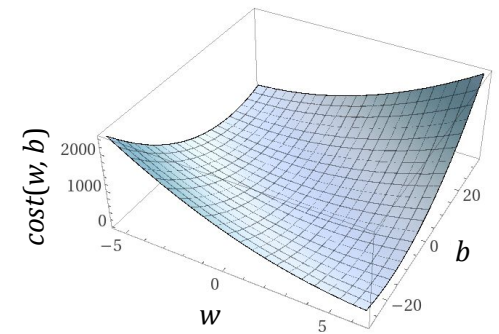
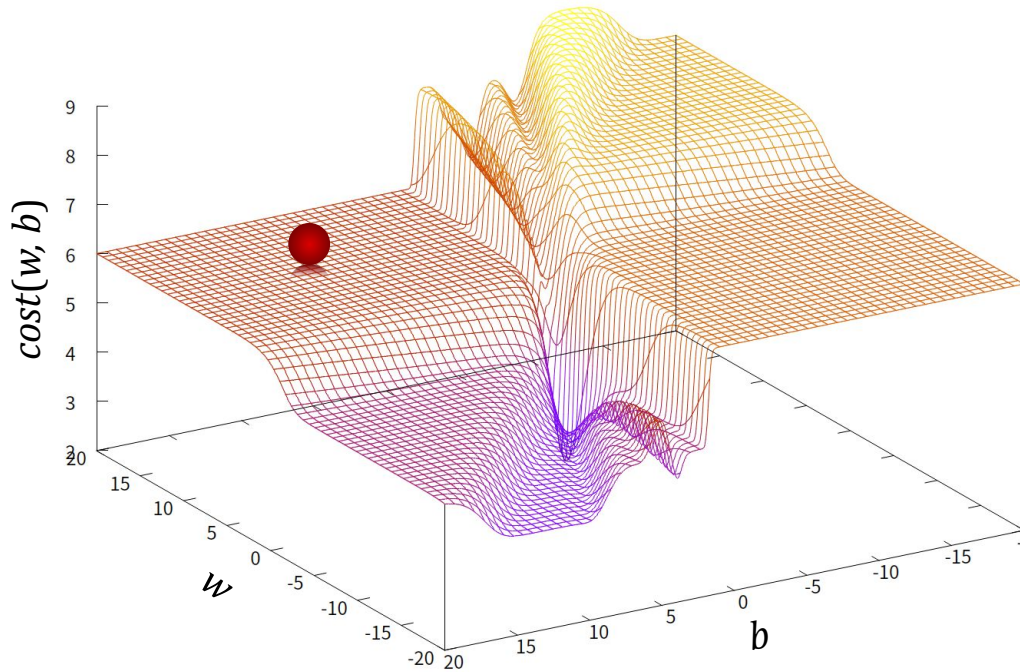




# 가설, 비용, Sigmoid

기존의 cost함수로는 gradient descent 알고리즘이 제대로 동작하지 않는다..!

$$\text{cost}(w, b) = \frac{1}{n} \sum (H(x) - y)^2$$

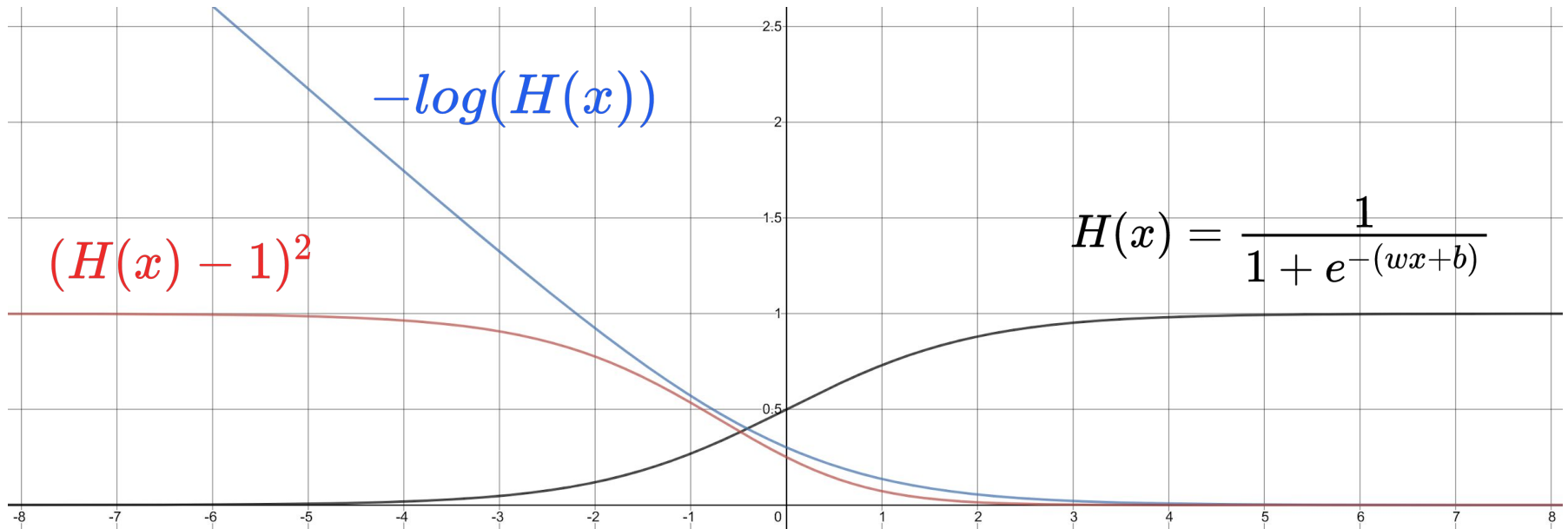
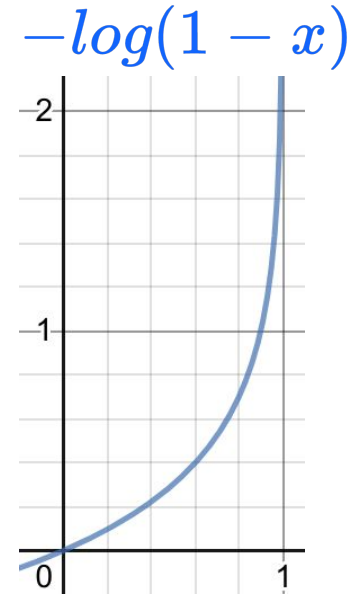


# 가설, 비용, Sigmoid

새로운 cost function

$$\text{cost}(w, b) = \frac{1}{n} \sum_{i=1}^n C(H(x_i), y_i)$$

$$C(h, y) = \begin{cases} -\log(1 - h) & \text{if } y = 0 \\ -\log(h) & \text{if } y = 1 \end{cases}$$



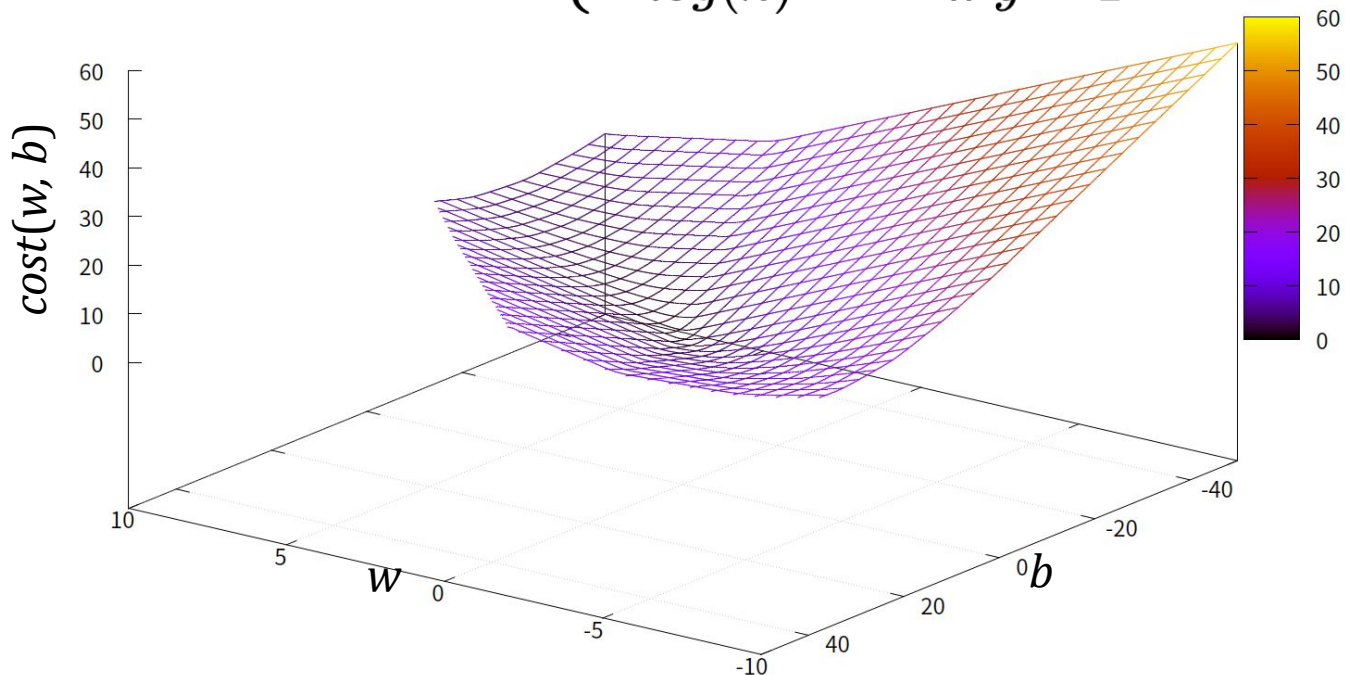
# 가설, 비용, Sigmoid

새로운 cost function

$$H(x) = \frac{1}{1 + e^{-(wx+b)}}$$

$$\text{cost}(w, b) = \frac{1}{n} \sum_{i=1}^n C(H(x_i), y_i)$$

$$C(h, y) = \begin{cases} -\log(1 - h) & \text{if } y = 0 \\ -\log(h) & \text{if } y = 1 \end{cases}$$



# 가설, 비용, Sigmoid

새로운 cost function

$$\text{cost}(w, b) = \frac{1}{n} \sum_{i=1}^n C(H(x_i), y_i)$$

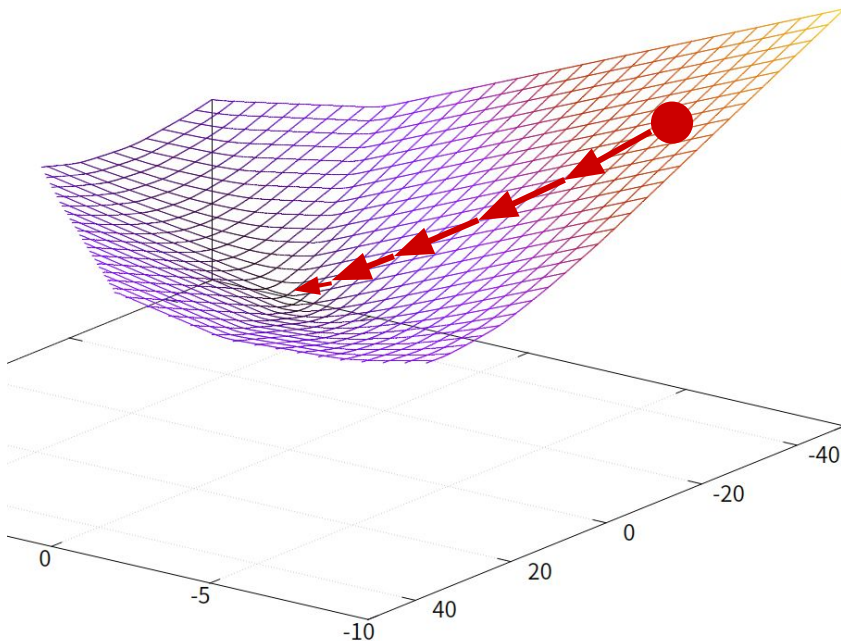
$$C(h, y) = \begin{cases} -\log(1 - h) & \text{if } y = 0 \\ -\log(h) & \text{if } y = 1 \end{cases}$$

$$C(H(x), y) = -y \log H(x) - (1 - y) \log(1 - H(x))$$

# 경사 하강법 Gradient Descent

우리의 목표: cost를 최소화 하자! = cost를 최소로 만드는  $w$ ,  $b$  값을 찾자!

$$\arg \min_{w,b} cost(w, b)$$



경사Gradient:

$$\left( \frac{\partial cost(w,b)}{\partial w}, \frac{\partial cost(w,b)}{\partial b} \right)$$

업데이트:

Learning Rate

$$w = w - \alpha \frac{\partial cost(w,b)}{\partial w}$$

$$b = b - \alpha \frac{\partial cost(w,b)}{\partial b}$$

# Logistic Regression (2)

입력이 조금 더 복잡할 때 (입력 차원이 1 이상일 때)?

## Training Data

[1.2, 3.8, -1.4, ..., 4.1]	→	0
[3.2, -1.2, -0.2, ..., 2.1]	→	0
[2.8, -1.4, -0.3, ..., 2.3]	→	1
[1.2, 3.4, -1.5, ..., 4.2]	→	0
[4.2, 2.1, 2.8, ..., -0.5]	→	1
...		
[3.2, 2.2, 2.2, ..., -0.4]	→	1

## Test

[1.3, 3.2, -1.5, ..., 4.1]	→	?
----------------------------	---	---

# Logistic Regression (2)

가설함수:

$$H(x) = \frac{1}{1 + e^{-(\mathbf{w}^\top \mathbf{x} + b)}}$$

비용:

$$cost(\mathbf{w}, b) = \frac{1}{n} \sum_{i=0}^n C(H(\mathbf{x}_i), y_i)$$

$$C(h, y) = \begin{cases} -\log(1 - h) & \text{if } y = 0 \\ -\log(h) & \text{if } y = 1 \end{cases}$$

업데이트:

$$\mathbf{w} = \mathbf{w} - \alpha \frac{\partial cost(\mathbf{w}, b)}{\partial \mathbf{w}}$$

$$b = b - \alpha \frac{\partial cost(\mathbf{w}, b)}{\partial b}$$

# Questions?