



# Machine Learning 03

Kihyun Shin  
DMSE, HBNU

# **Overview**

## **(Cost and loss function)**



# Simplified **loss** function

$$L(f(x^{(i)}), y^{(i)}) = \begin{cases} -\log(f_{w,b}(x^{(i)})) & \text{if } y^{(i)} = 1 \\ -\log(1 - f_{w,b}(x^{(i)})) & \text{if } y^{(i)} = 0 \end{cases}$$

equivalent

$$L(f(x^{(i)}), y^{(i)}) = -y^{(i)} \log(f_{w,b}(x^{(i)})) - (1 - y^{(i)}) \log(1 - f_{w,b}(x^{(i)}))$$

if  $y^{(i)} = 1$

$$L(f(x^{(i)}), y^{(i)}) = -1 \log(f_{w,b}(x^{(i)})) - \cancel{(1 - 1) \log(1 - f_{w,b}(x^{(i)}))}$$

if  $y^{(i)} = 0$

$$L(f(x^{(i)}), y^{(i)}) = \cancel{-0 \log(f_{w,b}(x^{(i)}))} - (1 - 0) \log(1 - f_{w,b}(x^{(i)}))$$



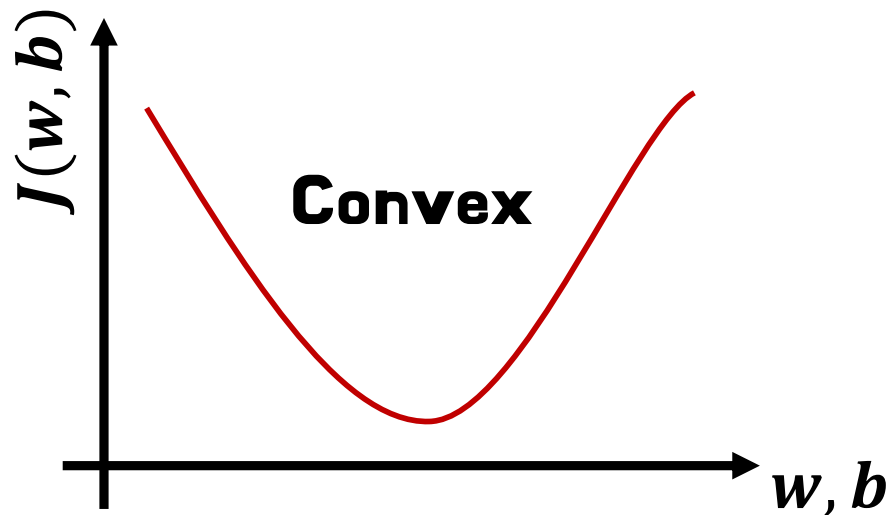
# Squared error cost

$$J(w, b) = \frac{1}{m} \sum_{i=1}^m \frac{1}{2} (f(x^{(i)}) - y^{(i)})^2$$

$= L(f(x^{(i)}), y^{(i)})$  **Loss**

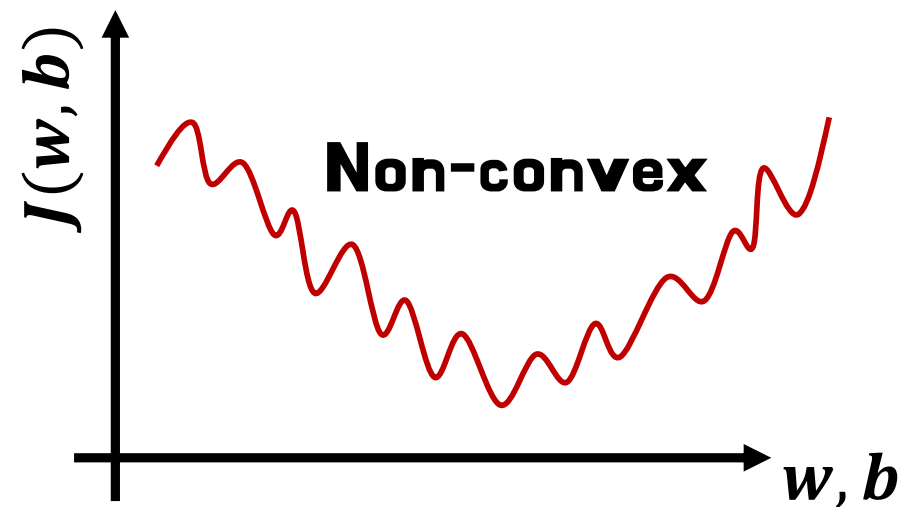
## Linear regression

$$f_{w,b}(x) = wx + b$$



## Logistic regression

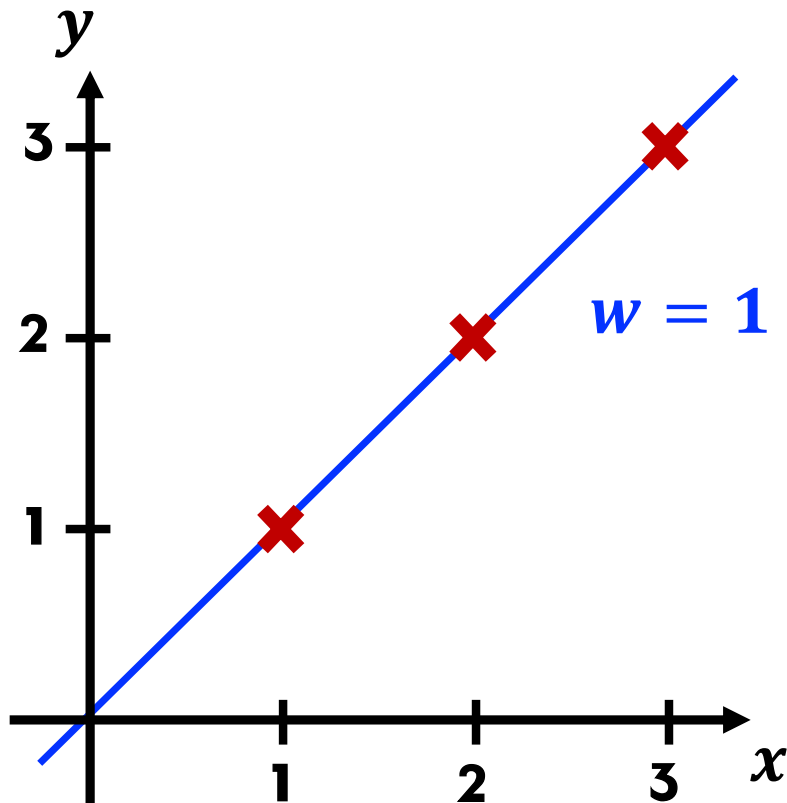
$$f_{w,b}(x) = \frac{1}{1 + e^{-(wx+b)}}$$



# Cost function

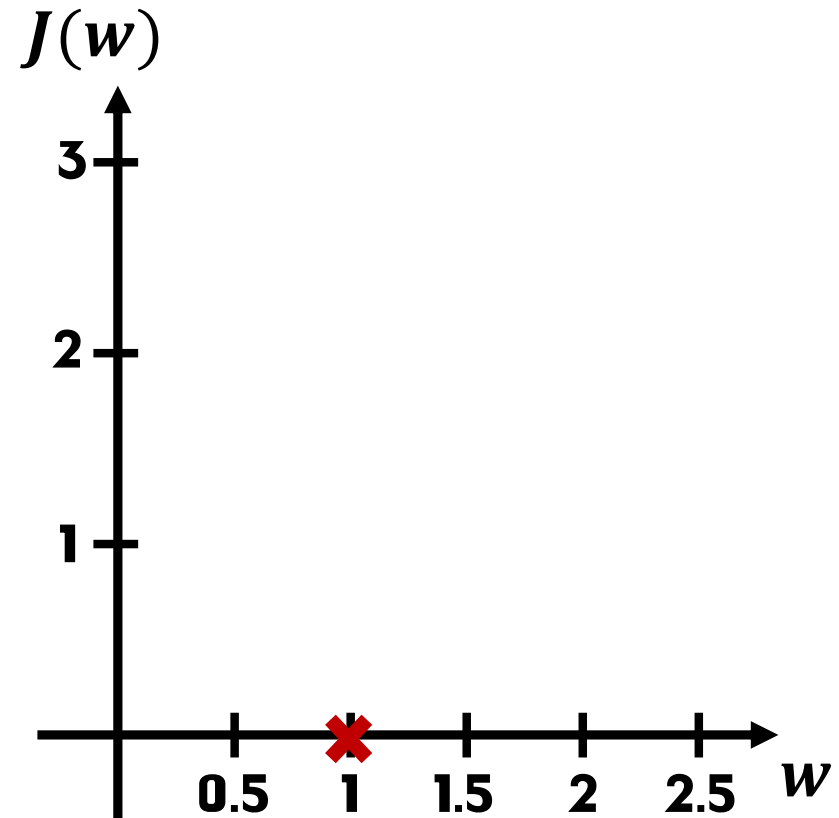
$$f_w(x)$$

For fixed  $w$ , function of  $x$



$$J(w)$$

Function of  $w$



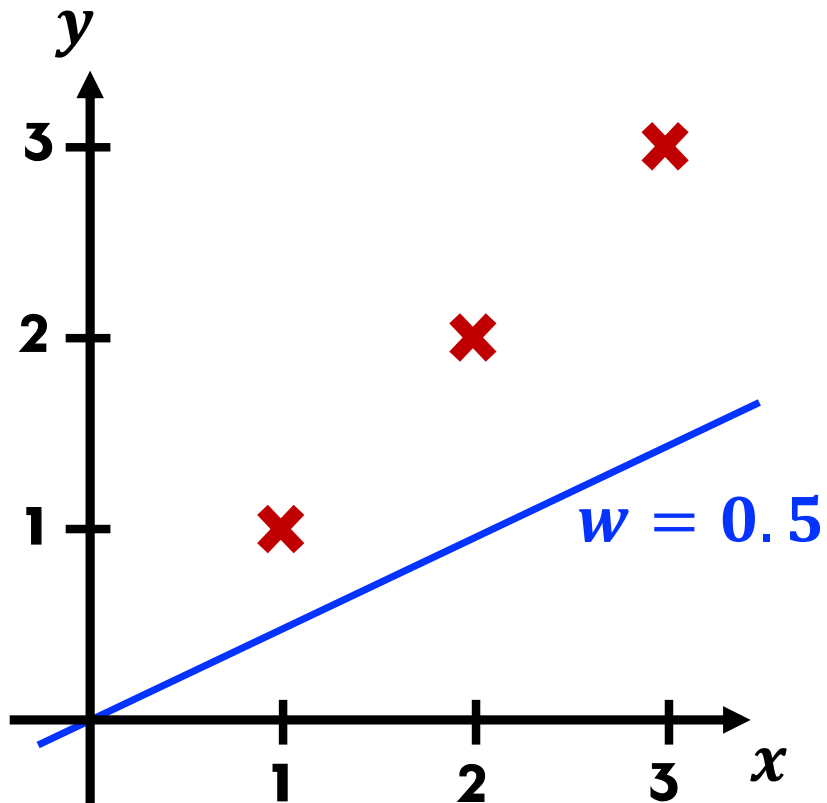
$$J(w) = \frac{1}{2m} \sum_{i=1}^m (f(x^{(i)}) - y^{(i)})^2 = \frac{1}{2m} \sum_{i=1}^m (wx^{(i)} - y^{(i)})^2 = \frac{1}{2m} (0^2 + 0^2 + 0^2) = 0$$



# Cost function

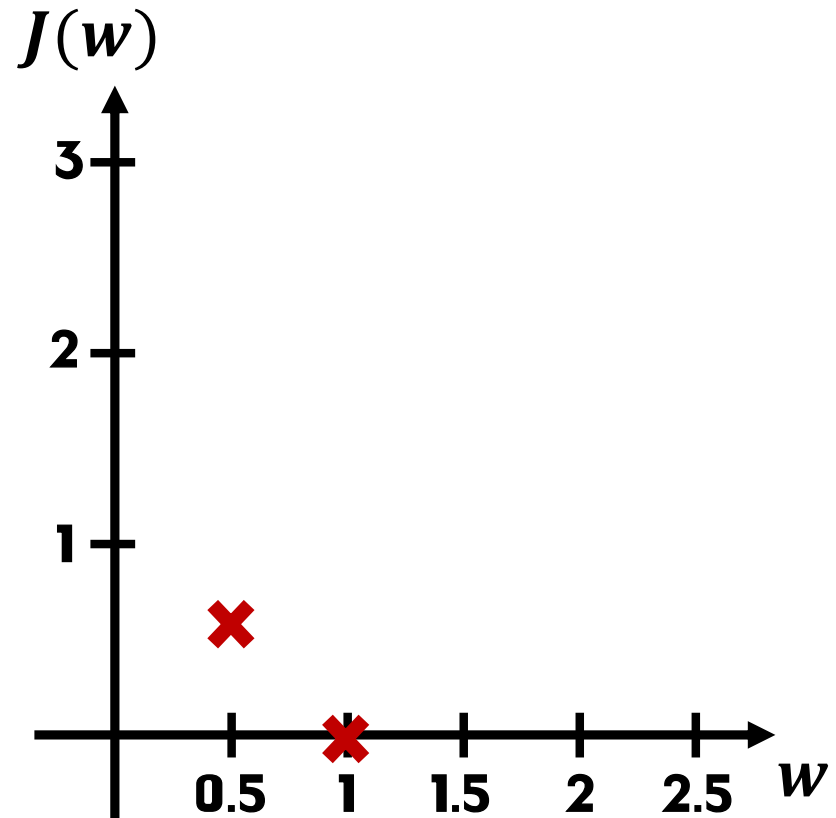
$$f_w(x)$$

For fixed  $w$ , function of  $x$



$$J(w)$$

Function of  $w$



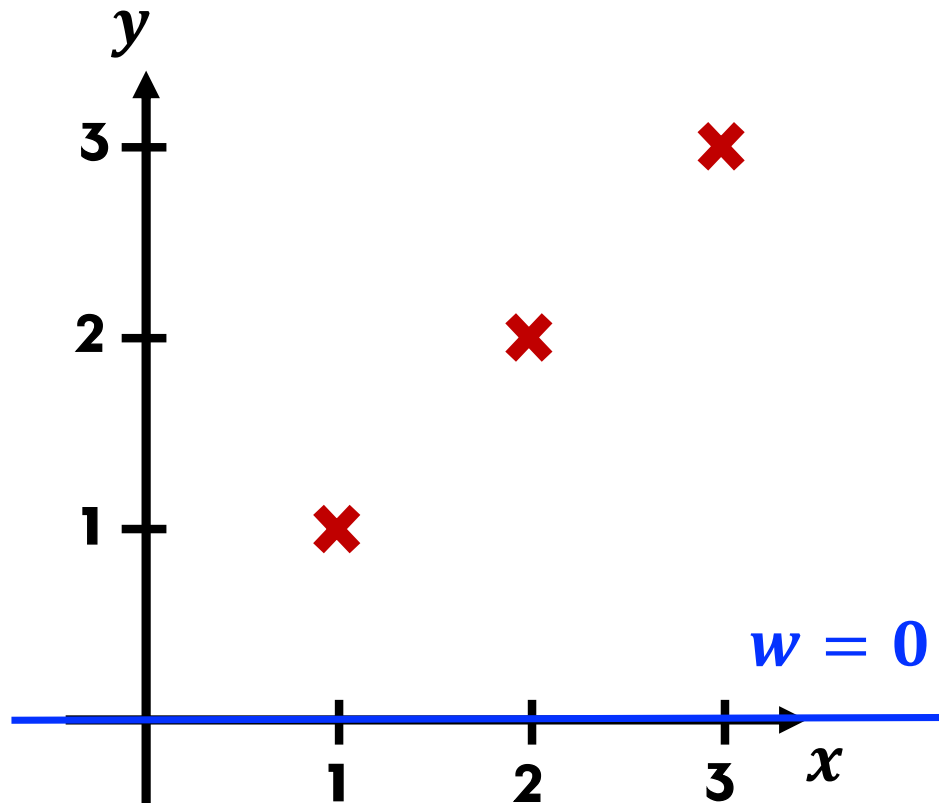
$$J(w) = \frac{1}{2m} \left( (0.5 - 1)^2 + (1 - 2)^2 + (1.5 - 3)^2 \right) = \frac{3.5}{6} = 0.58$$



# Cost function

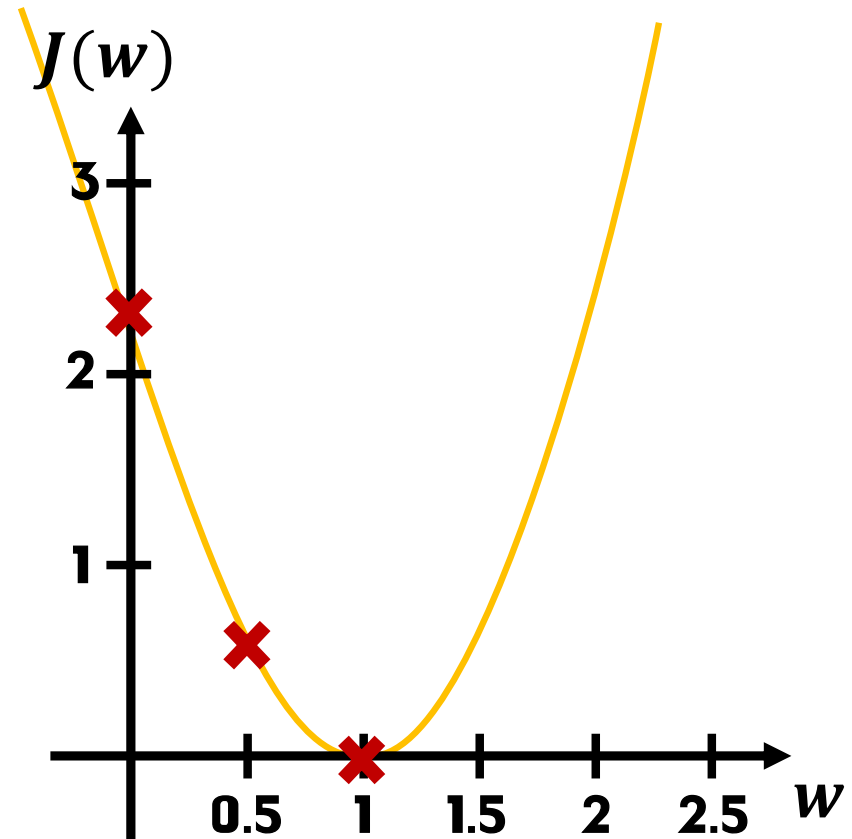
$$f_w(x)$$

For fixed  $w$ , function of  $x$



$$J(w)$$

Function of  $w$

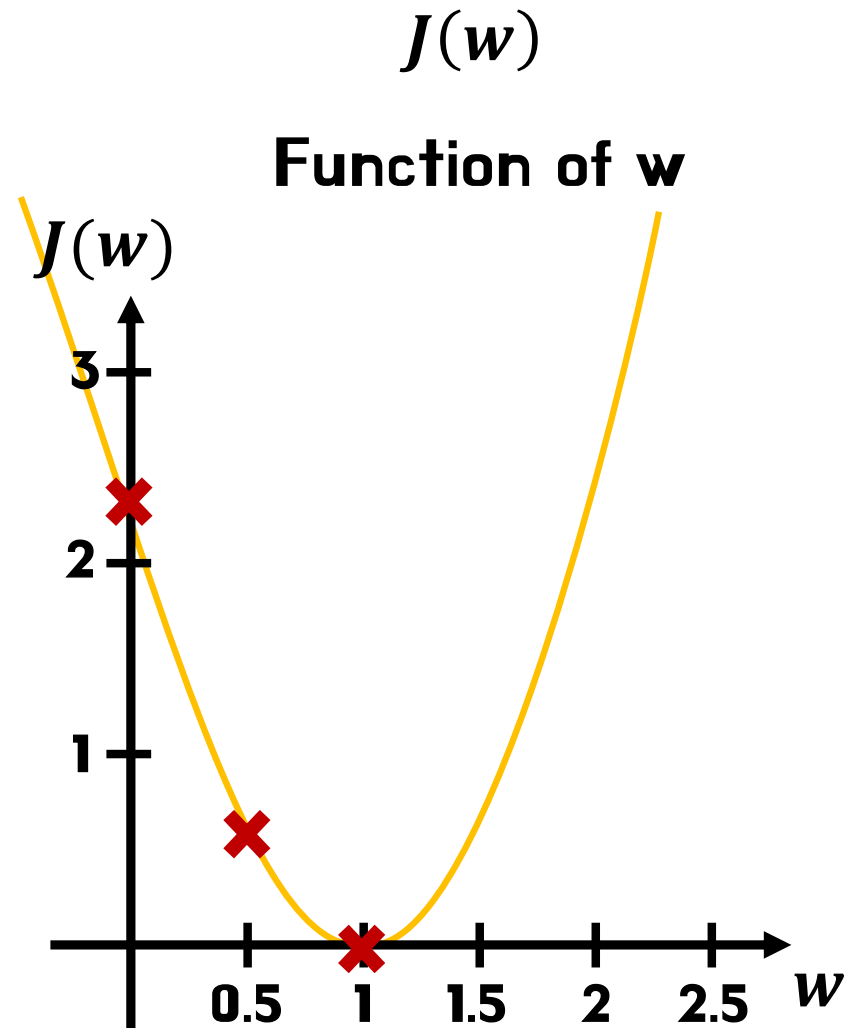


$$J(w) = \frac{1}{2m} \left( (1)^2 + (2)^2 + (3)^2 \right) = \frac{14}{6} = 2.3$$



# Cost function

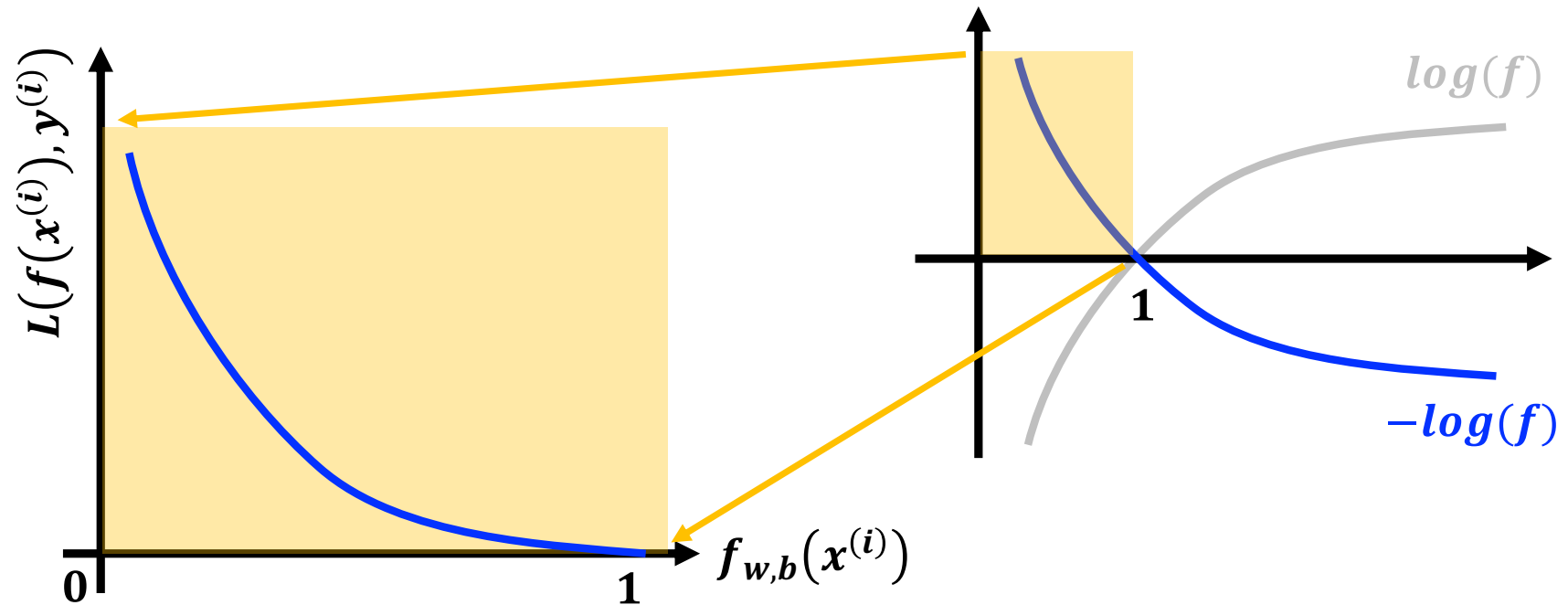
Goal of linear regression:  
**minimize  $J(w)$**



**Choose  $w$  to minimize  $J(w)$**

# Loss function

$$L(f(x^{(i)}), y^{(i)}) = \begin{cases} -\log(f_{w,b}(x^{(i)})) & \text{if } y^{(i)} = 1 \\ -\log(1 - f_{w,b}(x^{(i)})) & \text{if } y^{(i)} = 0 \end{cases}$$

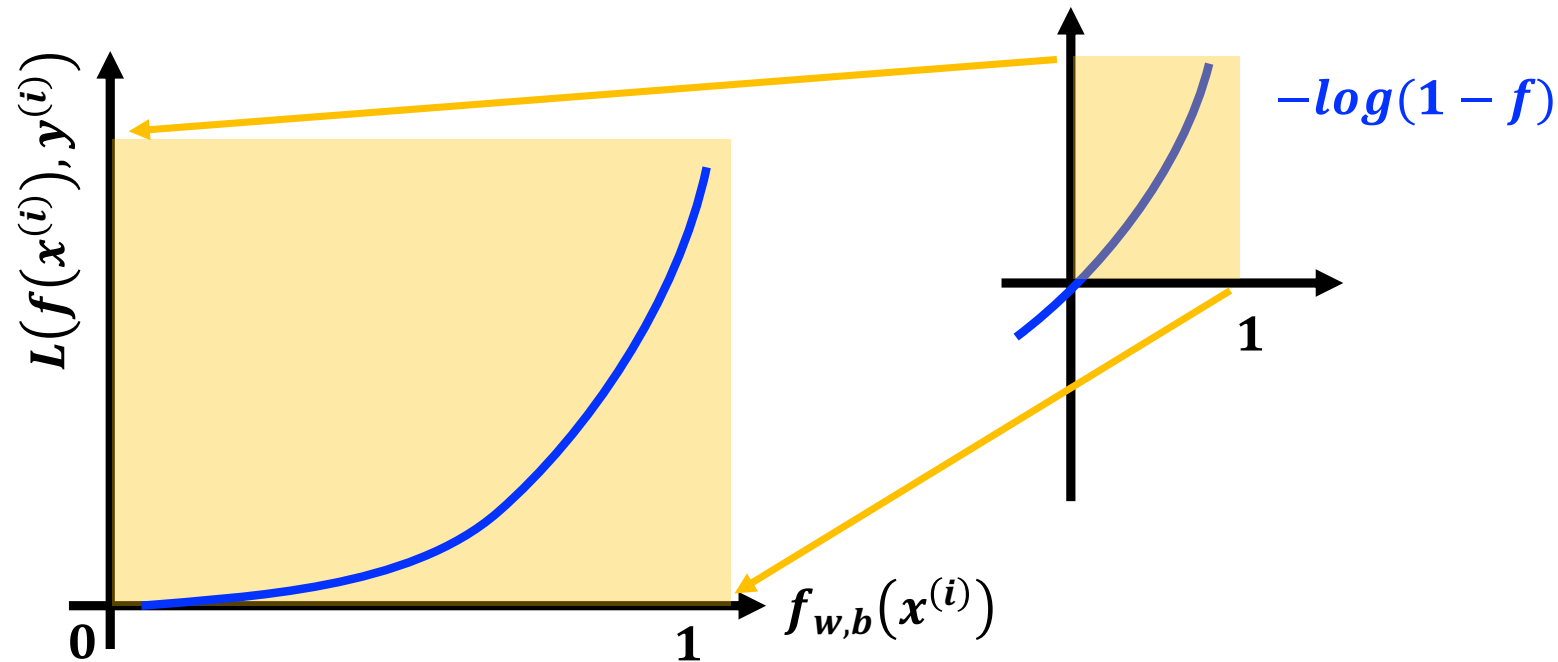


**As  $f_{w,b}(x^{(i)}) \rightarrow 1$  then loss  $\rightarrow 0$**

**As  $f_{w,b}(x^{(i)}) \rightarrow 0$  then loss  $\rightarrow$  infinite**

# Loss function

$$L(f(x^{(i)}), y^{(i)}) = \begin{cases} -\log(f_{w,b}(x^{(i)})) & \text{if } y^{(i)} = 1 \\ -\log(1 - f_{w,b}(x^{(i)})) & \text{if } y^{(i)} = 0 \end{cases}$$



**As  $f_{w,b}(x^{(i)}) \rightarrow 1$  then loss  $\rightarrow$  infinite**

**As  $f_{w,b}(x^{(i)}) \rightarrow 0$  then loss  $\rightarrow 0$**

# **Gradient descent**

**(way to minimize the cost function)**



# Outline

**Have some cost function  $J(w, b)$**  **For linear regression**  
**Or any function**

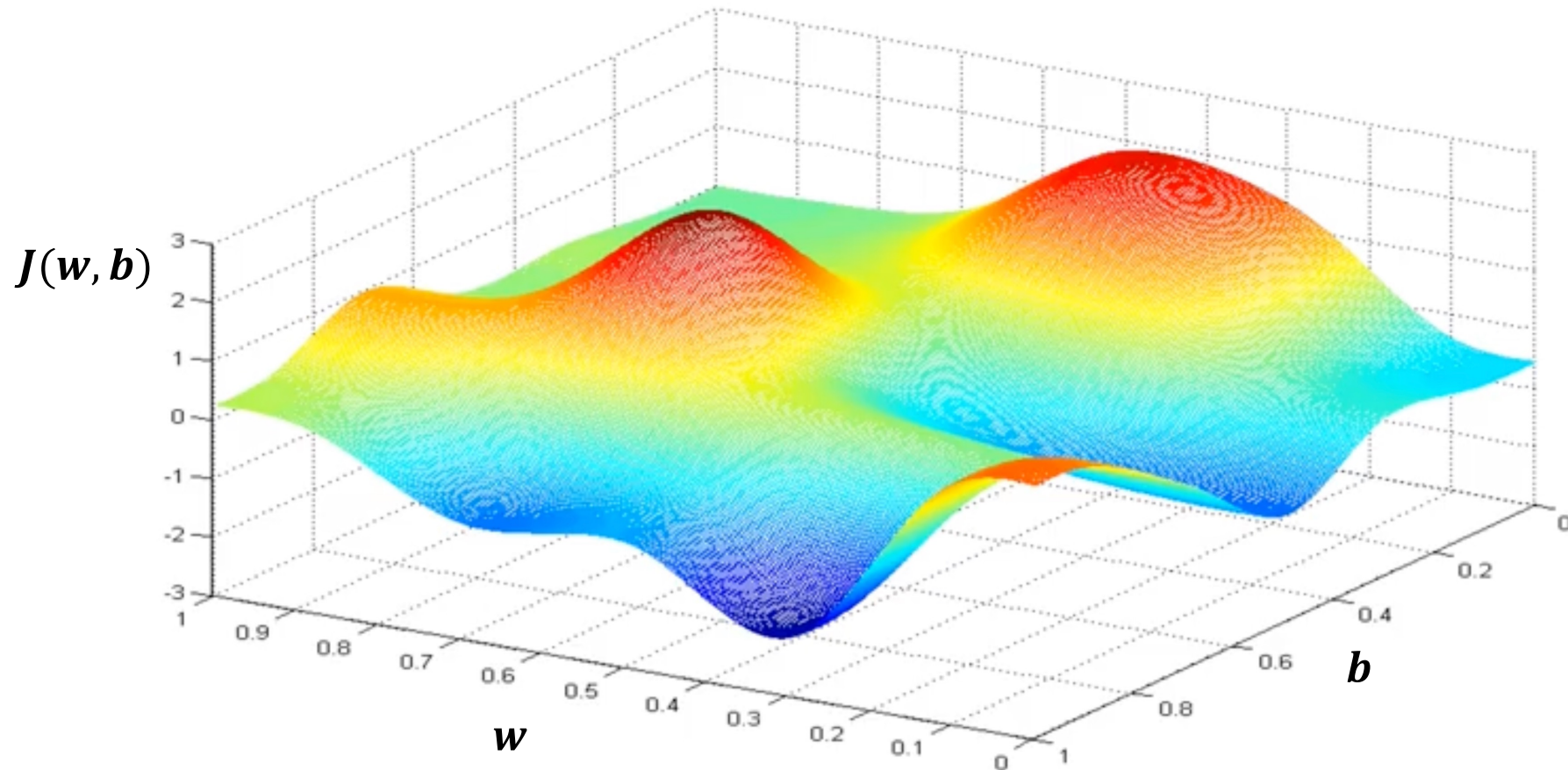
**Want  $\min_{w,b} J(w, b)$**

**Outline:**

- 1. start with some  $w, b$  (set  $w = 0, b = 0$ )**
- 2. keep changing  $w, b$  to reduce  $J(w, b)$**
- 3. until we settle at or near a minimum**

# Gradient descent

1. Finding way from hill to valley
2. local minimum(s)



# Gradient descent algorithm

Repeat until convergence

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

$$0 = -\alpha \frac{\partial}{\partial w} J(w, b) \text{ ????} \rightarrow \text{wrong}$$

In coding

$$a = c$$

$$a = a + 1 \\ \text{(O)}$$

In math

$$a = c$$

$$a = a + 1 \\ \text{(X)}$$



$$a_{n+1} = a_n + 1$$



# Gradient descent algorithm

Repeat until convergence

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

**future** →  $w$  (in the next step)  
→  $w$  (current step) **present**  
→  $\alpha$  **Learning rate**  
→  $\frac{\partial}{\partial w} J(w, b)$  **Partial derivative**

# Gradient descent algorithm

**Repeat until convergence**

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

**Simultaneously update w and b**

**Correct**

$$tmp\_w = w - \alpha \frac{\partial}{\partial w} J(w, b)$$

$$tmp\_b = b - \alpha \frac{\partial}{\partial b} J(w, b)$$

$$w = tmp\_w$$

$$b = tmp\_b$$

**Incorrect**

$$tmp\_w = w - \alpha \frac{\partial}{\partial w} J(w, b)$$

$$w = tmp\_w$$

$$tmp\_b = b - \alpha \frac{\partial}{\partial b} J(tmp\_w, b)$$

$$b = tmp\_b$$



# Gradient descent algorithm

Repeat until convergence

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

Learning rate

Partial derivative

In simplified equation,

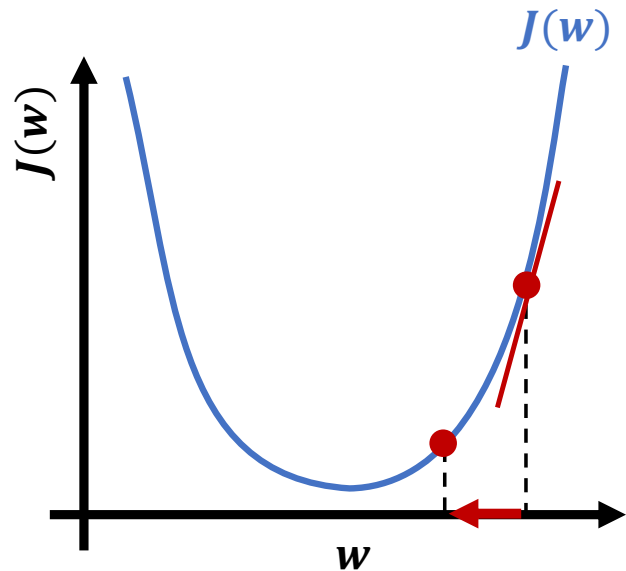
$$J(w)$$

$$w = w - \alpha \frac{d}{dw} J(w)$$

$$\min_w J(w)$$



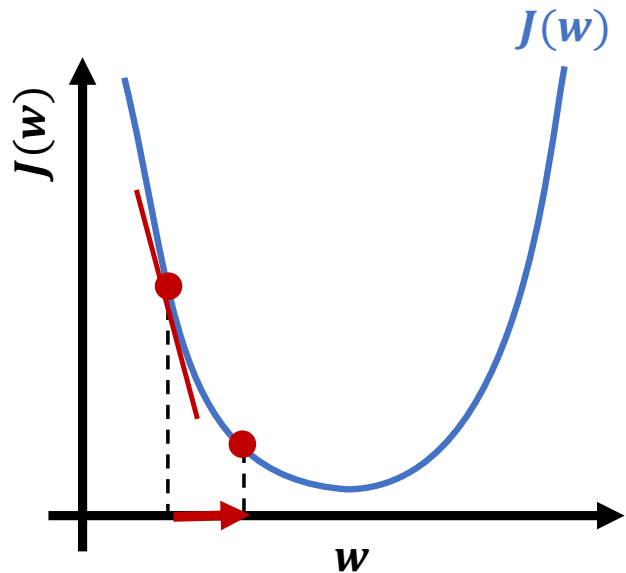
# Gradient descent algorithm



$$w = w - \alpha \frac{d}{dw} J(w)$$

$> 0$

$$= w - \alpha(\text{positive number})$$



$$w = w - \alpha \frac{d}{dw} J(w)$$

$< 0$

$$= w - \alpha(\text{negative number})$$



# Learning rate

$$w = w - \alpha \frac{d}{dw} J(w)$$

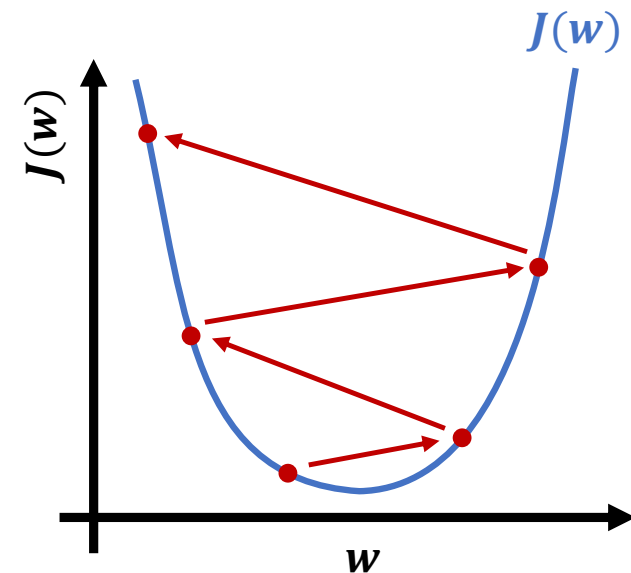
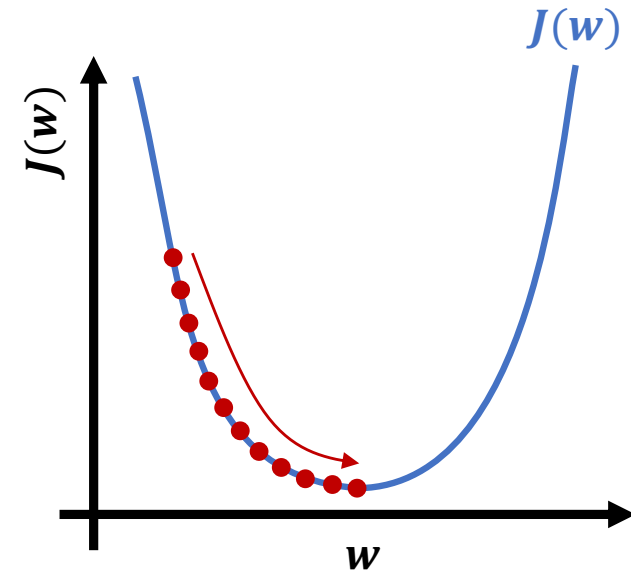
If is  $\alpha$  too small ...

**Gradient descent may be slow**

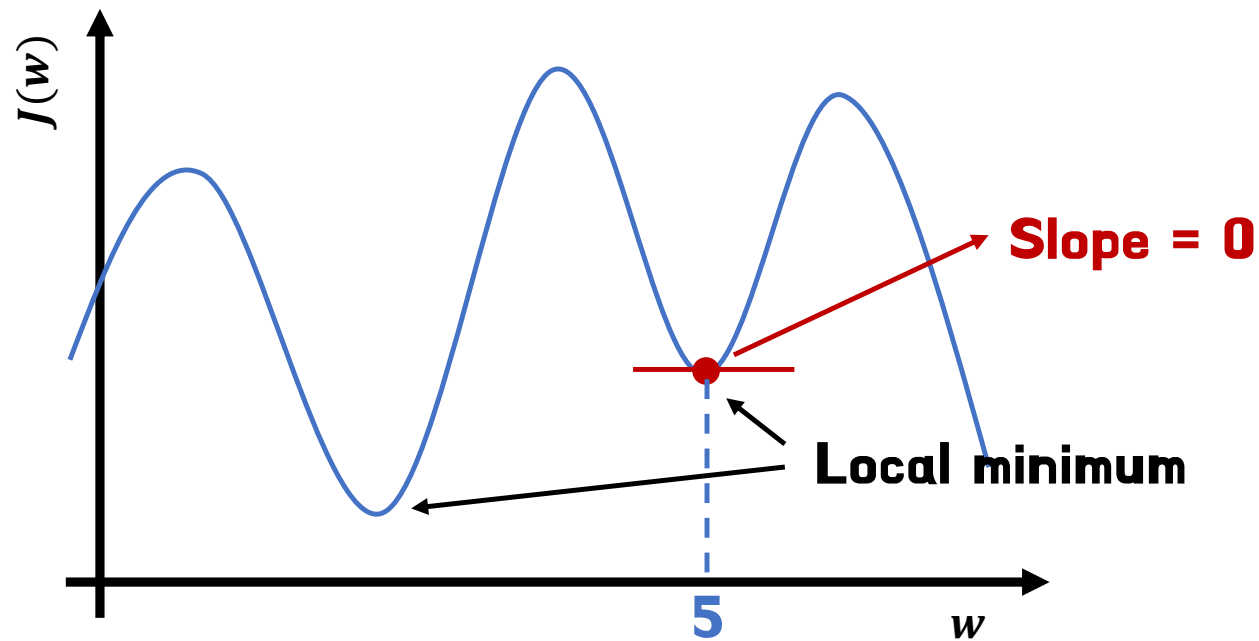
If is  $\alpha$  too large ...

**Gradient descent may:**

- **Overshoot, never reach minimum**
- **Fail to converge  $\rightarrow$  diverge**



# Learning rate



**What happen if the slope is zero ?**

**(Let's say  $\alpha$  is 0.1)**

$$w = w - \alpha \frac{d}{dw} J(w) = 5 - 0.1 * 0 = 5 \text{ (no update)}$$

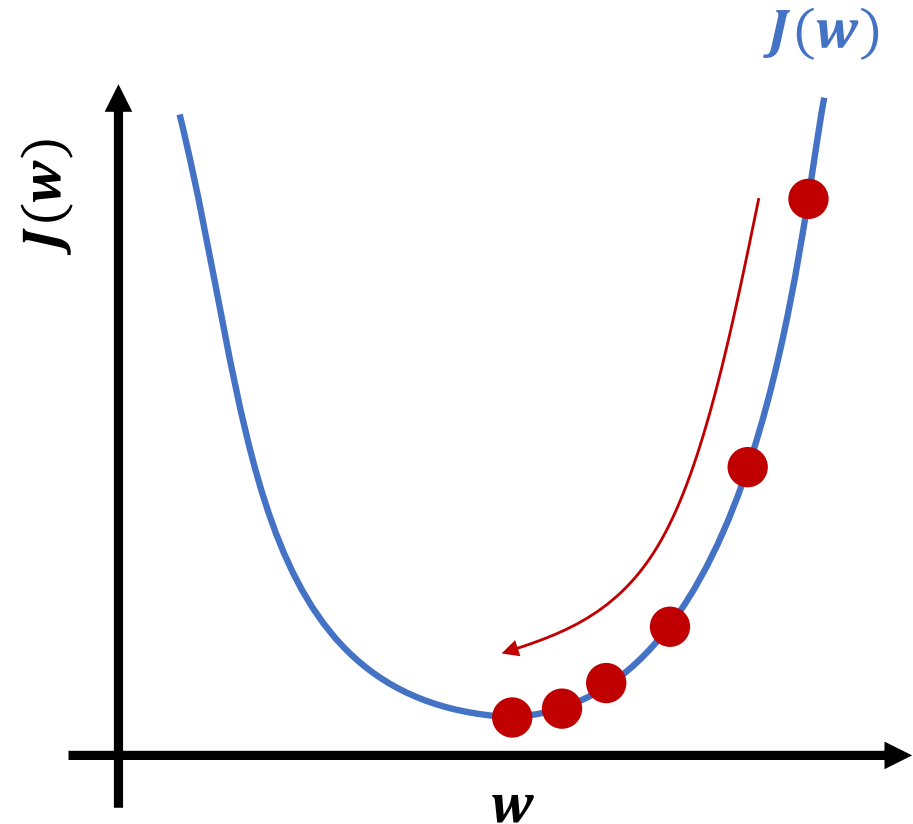
# Learning rate

Can reach local minimum with fixed learning rate ?

$$w = w - \alpha \frac{d}{dw} J(w)$$

Near a local minimum,

- **Derivative**  $\rightarrow$  smaller
- **Update steps**  $\rightarrow$  smaller



Can reach minimum without decreasing learning rate

# **Gradient descent (For linear regression)**



# Gradient descent for linear regression

- **Linear regression model**

$$f_{w,b}(x) = wx + b$$

- **Cost function**

$$J(w, b) = \frac{1}{2m} \sum_{i=1}^m (f(x^{(i)}) - y^{(i)})^2$$

- **Gradient descent algorithm (repeat until convergence)**

$$w = w - \alpha \frac{\partial}{\partial w} J(w, b) \longrightarrow \frac{1}{m} \sum_{i=1}^m (f(x^{(i)}) - y^{(i)})x^{(i)}$$

$$b = b - \alpha \frac{\partial}{\partial b} J(w, b) \longrightarrow \frac{1}{m} \sum_{i=1}^m (f(x^{(i)}) - y^{(i)})$$



# Gradient descent for linear regression

$$\begin{aligned}\frac{\partial}{\partial w} J(w, b) &= \frac{\partial}{\partial w} \left[ \frac{1}{2m} \sum_{i=1}^m (f(x^{(i)}) - y^{(i)})^2 \right] = \frac{\partial}{\partial w} \left[ \frac{1}{2m} \sum_{i=1}^m (wx^{(i)} + b - y^{(i)})^2 \right] \\ &= \left[ \frac{1}{2m} \sum_{i=1}^m (wx^{(i)} + b - y^{(i)}) 2x^{(i)} \right] = \frac{1}{m} \sum_{i=1}^m (f(x^{(i)}) - y^{(i)}) x^{(i)}\end{aligned}$$

$$\begin{aligned}\frac{\partial}{\partial b} J(w, b) &= \frac{\partial}{\partial b} \left[ \frac{1}{2m} \sum_{i=1}^m (f(x^{(i)}) - y^{(i)})^2 \right] = \frac{\partial}{\partial b} \left[ \frac{1}{2m} \sum_{i=1}^m (wx^{(i)} + b - y^{(i)})^2 \right] \\ &= \left[ \frac{1}{2m} \sum_{i=1}^m (wx^{(i)} + b - y^{(i)}) 2 \right] = \frac{1}{m} \sum_{i=1}^m (f(x^{(i)}) - y^{(i)})\end{aligned}$$



# Gradient descent for linear regression

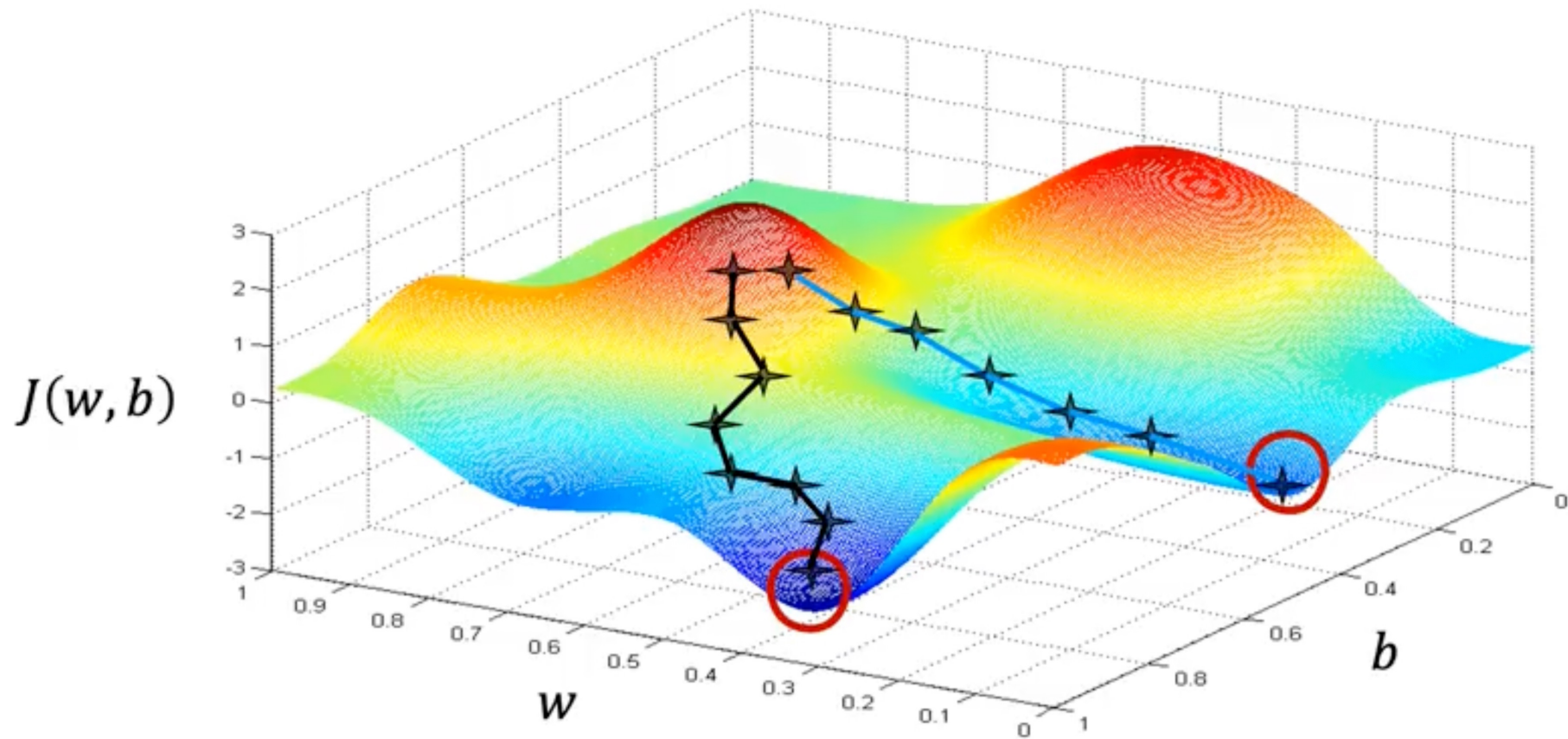
- **repeat until convergence by updating  $w$  and  $b$  in linear regression**

$$w = w - \alpha \frac{1}{m} \sum_{i=1}^m (wx^{(i)} + b - y^{(i)})x^{(i)}$$

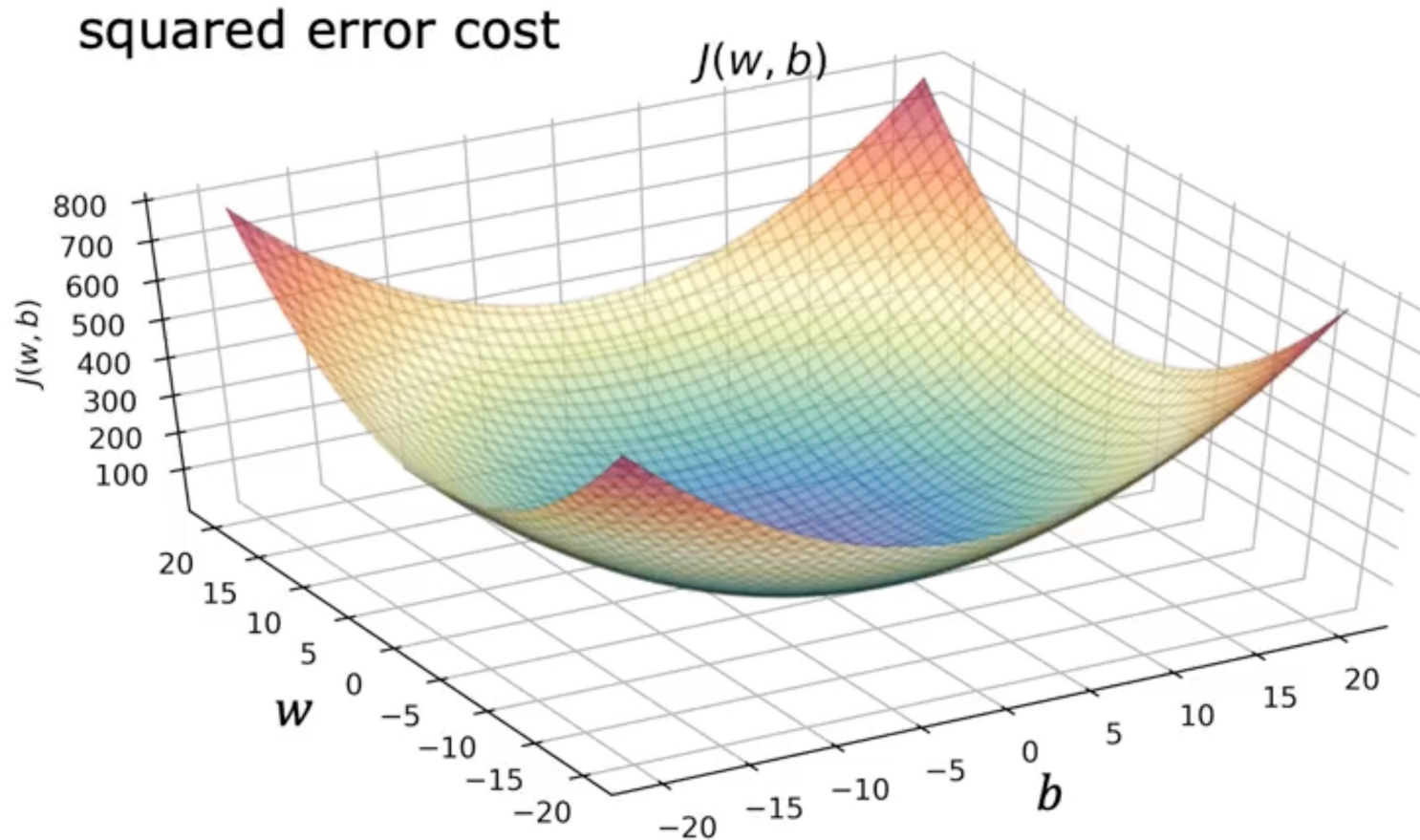
$$b = b - \alpha \frac{1}{m} \sum_{i=1}^m (wx^{(i)} + b - y^{(i)})$$

# Non-linear cost function

## Multiple local minimums



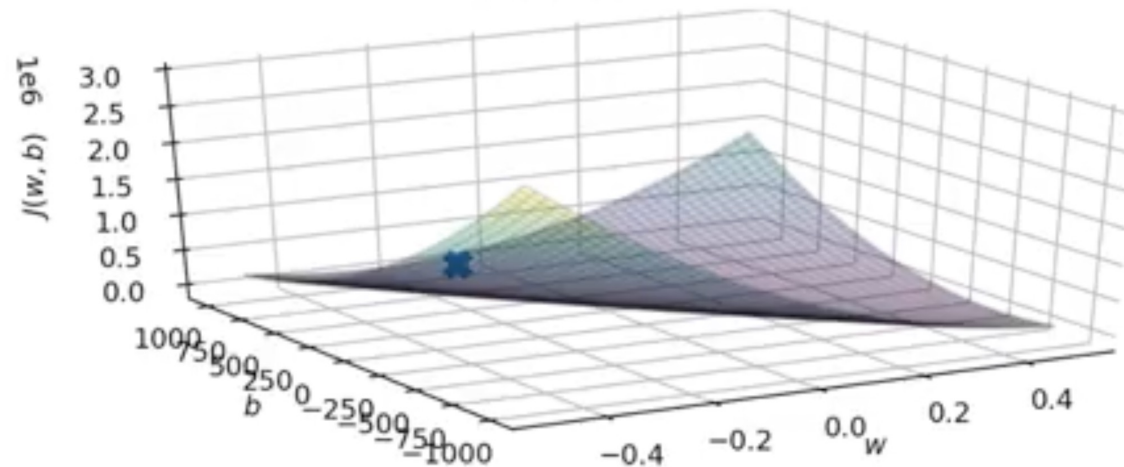
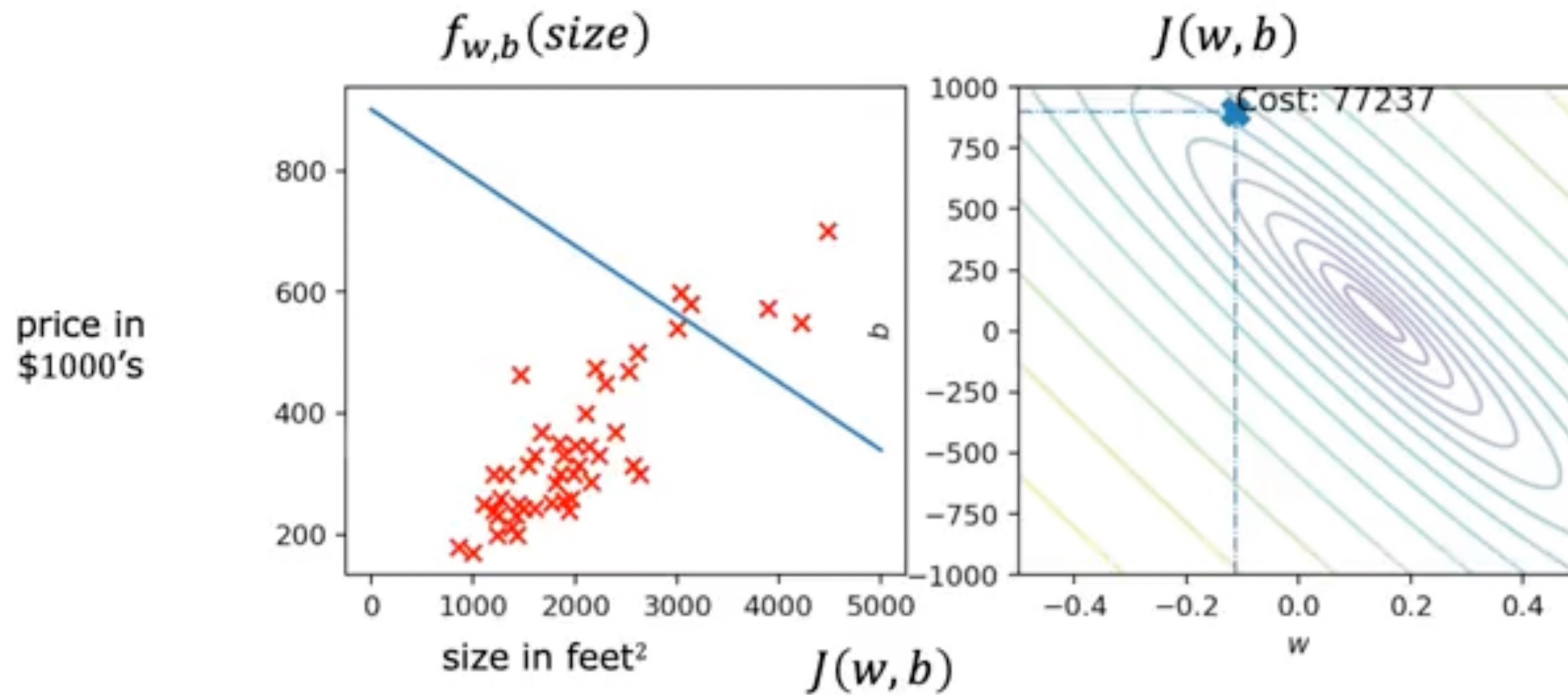
# Squared error cost function



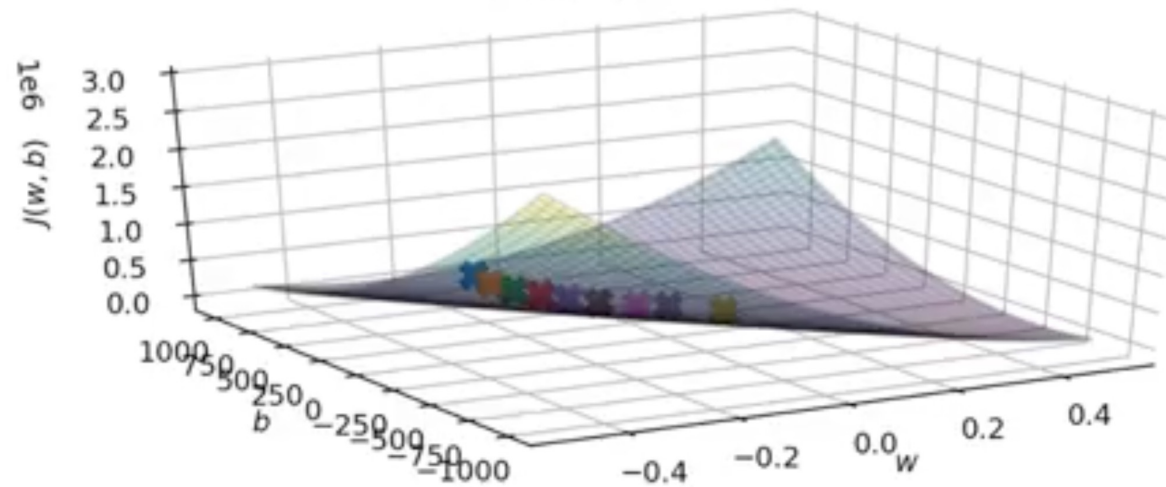
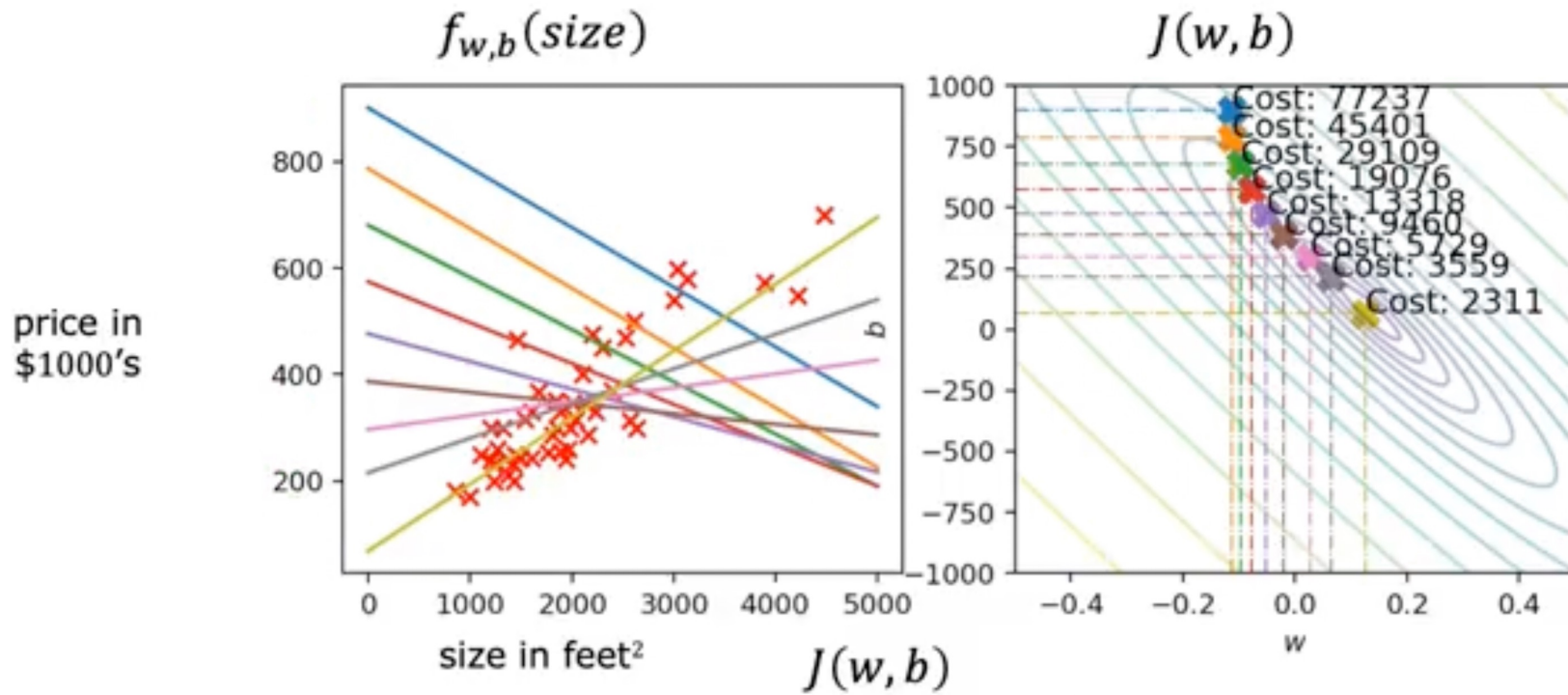
**One global minimum in convex function**

**If you choose proper learning rate, it will always converge to global minimum**

# Running gradient descent



# Running gradient descent



# "Batch" gradient descent

**"Batch": each step of gradient descent uses all the training examples**

	x Size in feet <sup>2</sup>	y Price in \$1000's
(1)	2104	400
(2)	1416	232
(3)	1534	315
(4)	852	178
...	...	...
(47)	3210	870

$$m = 47$$

$$J(w, b) = \frac{1}{2m} \sum_{i=1}^m (f(x^{(i)}) - y^{(i)})^2$$



# **Gradient descent (For logistic regression)**



# Training logistic regression

Find  $w, b$

Given new  $x$ , output  $f_{w,b}(x) = \frac{1}{1+e^{-(wx+b)}}$

$$P(y = 1|x; w, b)$$

# Gradient descent for logistic regression

$$J(\mathbf{w}, b) = -\frac{1}{m} \sum_{i=1}^m y^{(i)} \log(f_{\mathbf{w},b}(x^{(i)})) + (1 - y^{(i)}) \log(1 - f_{\mathbf{w},b}(x^{(i)}))$$

**Repeat until convergence (simultaneous  $w, b$  update)**

$$w = w - \alpha \frac{\partial}{\partial w} J(w, b) \longrightarrow \frac{\partial}{\partial w} J(w, b) = \frac{1}{m} \sum_{i=1}^m (f(x^{(i)}) - y^{(i)}) x^{(i)}$$

$$b = b - \alpha \frac{\partial}{\partial b} J(w, b) \longrightarrow \frac{\partial}{\partial b} J(w, b) = \frac{1}{m} \sum_{i=1}^m (f(x^{(i)}) - y^{(i)})$$

**Not proving this one**



# Gradient descent for logistic regression

$$w = w - \alpha \frac{1}{m} \sum_{i=1}^m (f(x^{(i)}) - y^{(i)}) x^{(i)}$$

$$b = b - \alpha \frac{1}{m} \sum_{i=1}^m (f(x^{(i)}) - y^{(i)})$$

**Looks like linear regression !**

**Linear regression**       $f_{w,b}(x) = wx + b$

**Logistic regression**       $f_{w,b}(x) = \frac{1}{1 + e^{-(wx+b)}}$

