Week 1: Introduction to Machine Learning

Contents

Contents

Contents

4. Train the Model with Gradient Descent
    - Gradient descent
    - Implementing gradient descent
    - Gradient descent intuition
    - Learning rate
    - Gradient descent for linear regression
    - Running gradient descent

1. Overview of Machine Learning

Applications of Machine Learning

- **Everyday Examples**
  - Web searches using Google, Bing, Baidu
  - Image recognition on Instagram, Snapchat
  - Movie recommendations on streaming services
  - Voice-to-text on the phone
  - Email spam filtering

Applications of Machine Learning

- **Industrial and Big Companies Examples**
  - Introduction to AI in big companies
  - Climate change example: Machine Learning optimizing wind turbine power generation
  - Healthcare example: Assisting doctors in accurate diagnosis
  - Industrial example: Computer vision in factories for defect inspection

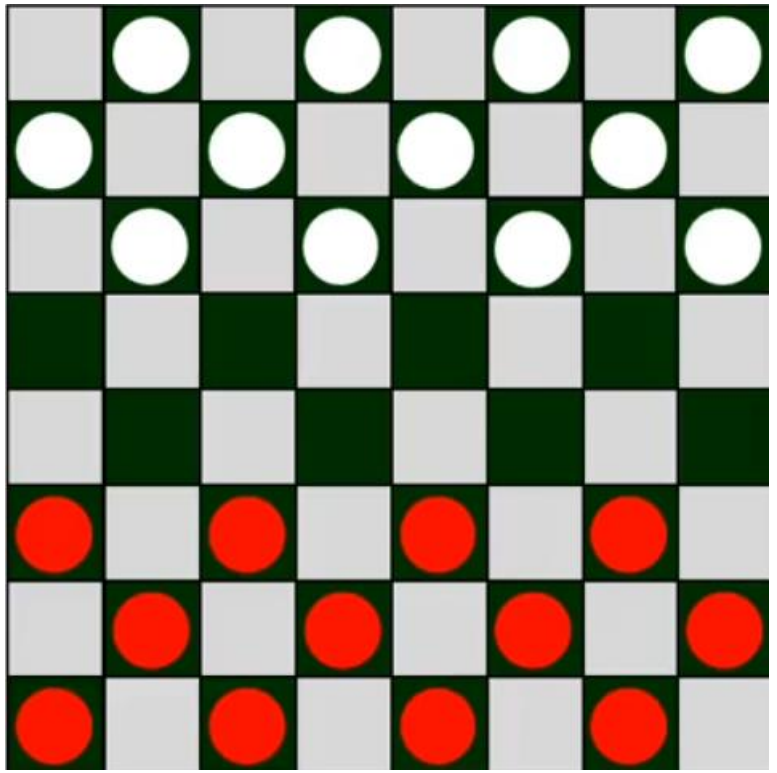
Value Creation and Opportunities

- McKinsey's estimation of AI and machine learning to create $13 trillion of value annually by 2030
- Current value creation in the software industry
- Potential for even greater value creation in various sectors: retail, travel, transportation, automotive, manufacturing, etc.
- Highlighting the vast demand for machine learning skills in diverse sectors

2. Supervised Vs Unsupervised ML

What is Machine Learning?

"Field of study that gives computers the ability to learn without being explicitly programmed" (attributed to Arthur Samuel)



What is Machine Learning?

Machine Learning Algorithms

- Supervised Learning
- Unsupervised Learning
  - Recommender Systems
- Reinforcement Learning

Practical advice for applying learning algorithms

Supervised Machine Learning-Part 1

- This method involves algorithms that learn the mapping from input (x) to output (y).
- By presenting the learning algorithm with numerous instances of input-output pairs, it eventually learns to make accurate predictions or guesses based solely on input.

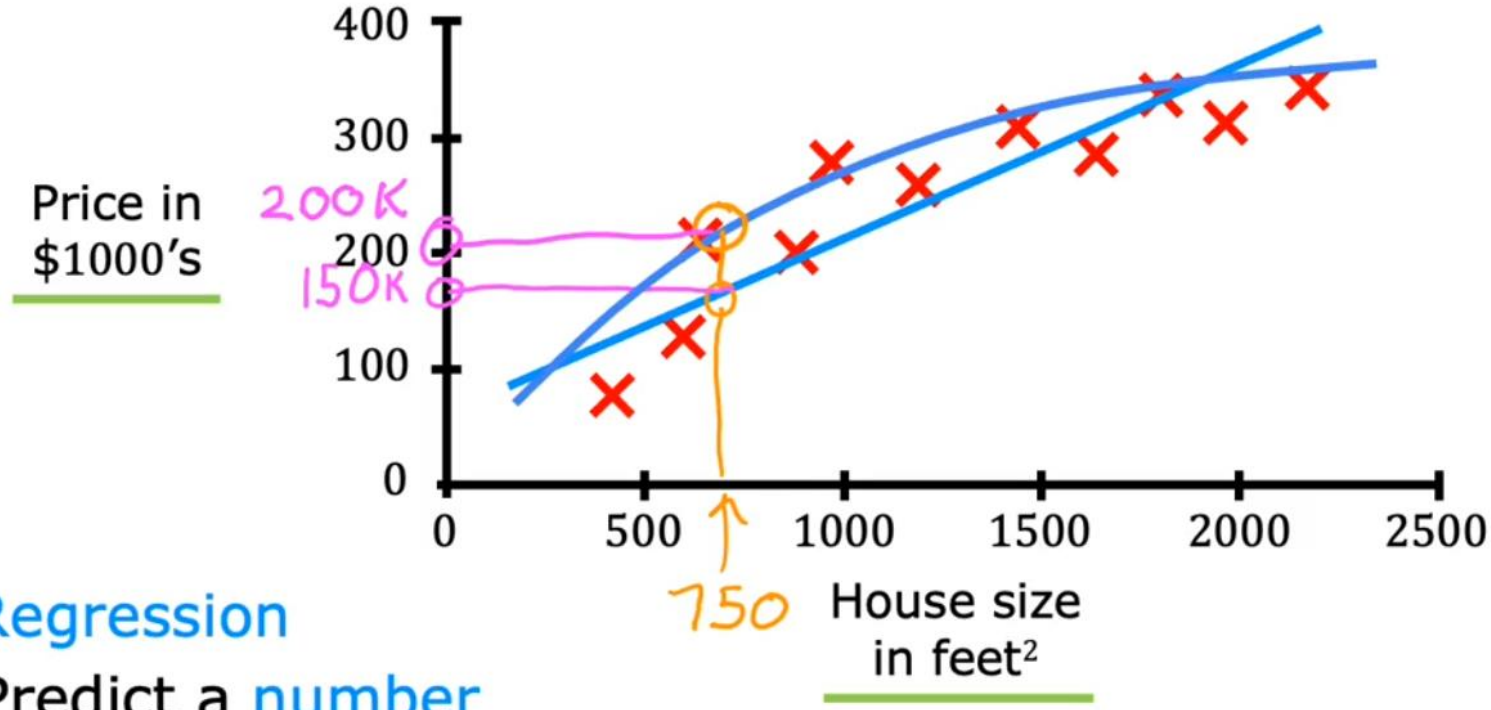
Learns from being given "**right answers**"

Supervised Machine Learning-Part 1

Input (X)	Output (Y)	Application
Email	Spam? (0/1)	Spam filtering
Audio	Text transcripts	Speech recognition
English	Spanish	Machine translation
Ad, user info	Click? (0/1)	Online advertising
Image, radar info	Position of the car	Self driving cars
Image of phone	Defect? (0/1)	Visual inspection

Supervised Machine Learning-Part 1

Regression: Housing price prediction

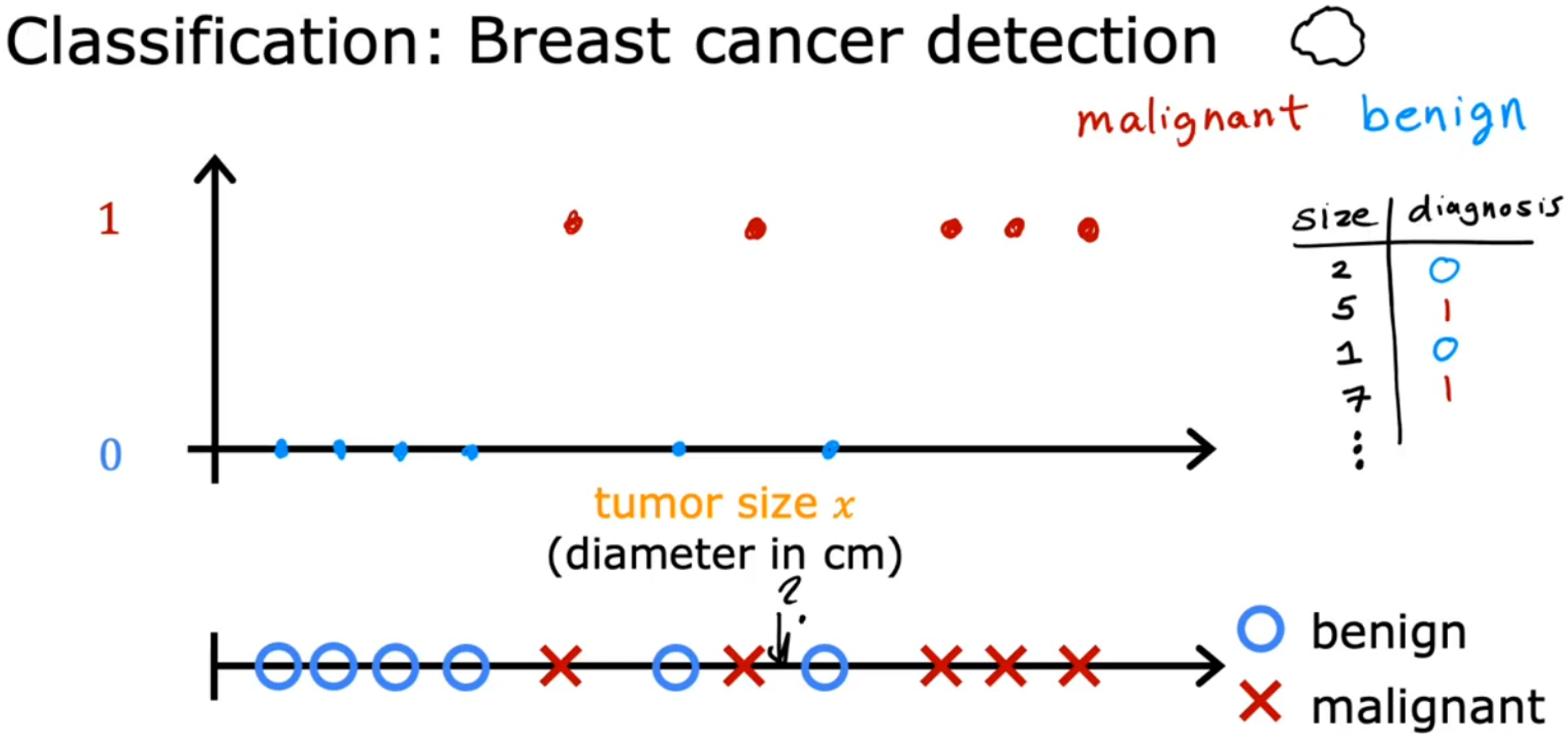


Regression

Predict a number

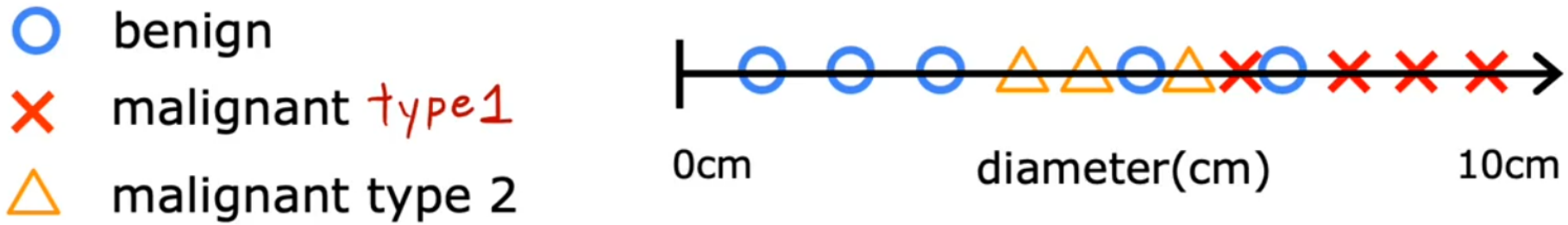
infinitely many possible outputs

Supervised Machine Learning-Part 2



Supervised Machine Learning-Part 2

Classification: Breast cancer detection



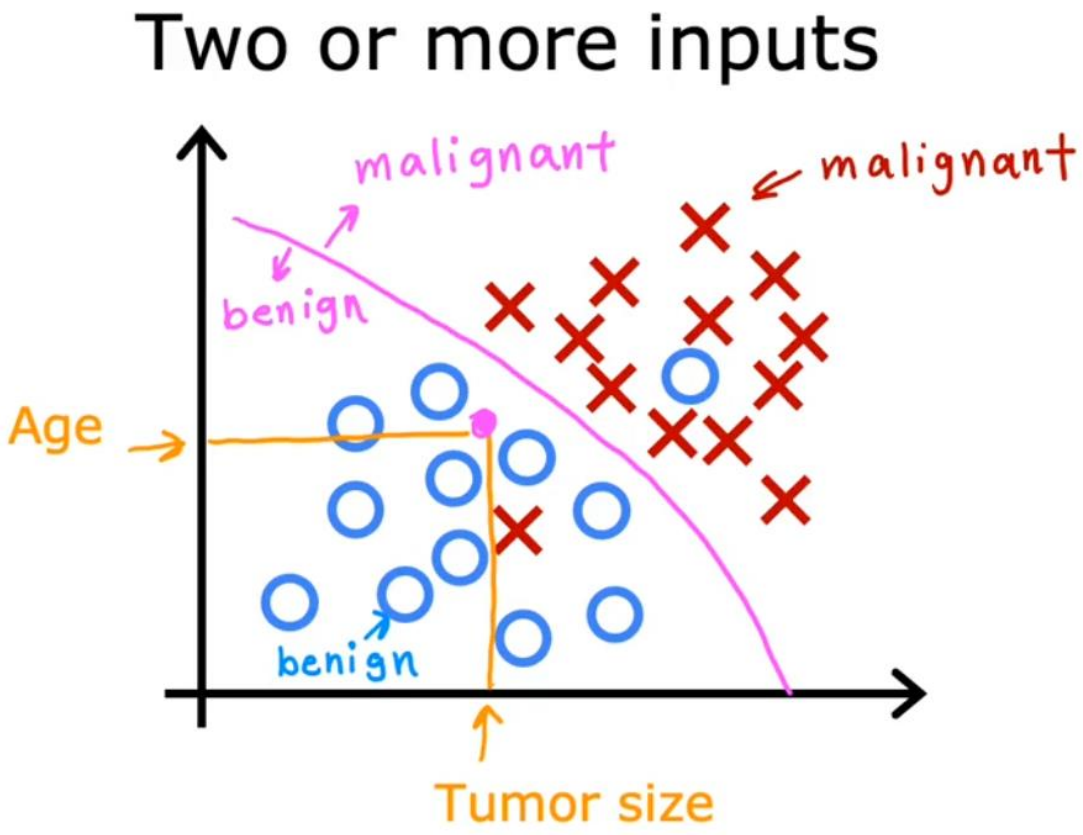
class category

Classification

predict categories cat dog benign malignant 0, 1, 2

small number of possible outputs

Supervised Machine Learning-Part 2



Supervised Machine Learning-Part 2

Supervised learning

Learns from being given "right answers"

Regression

Predict a number

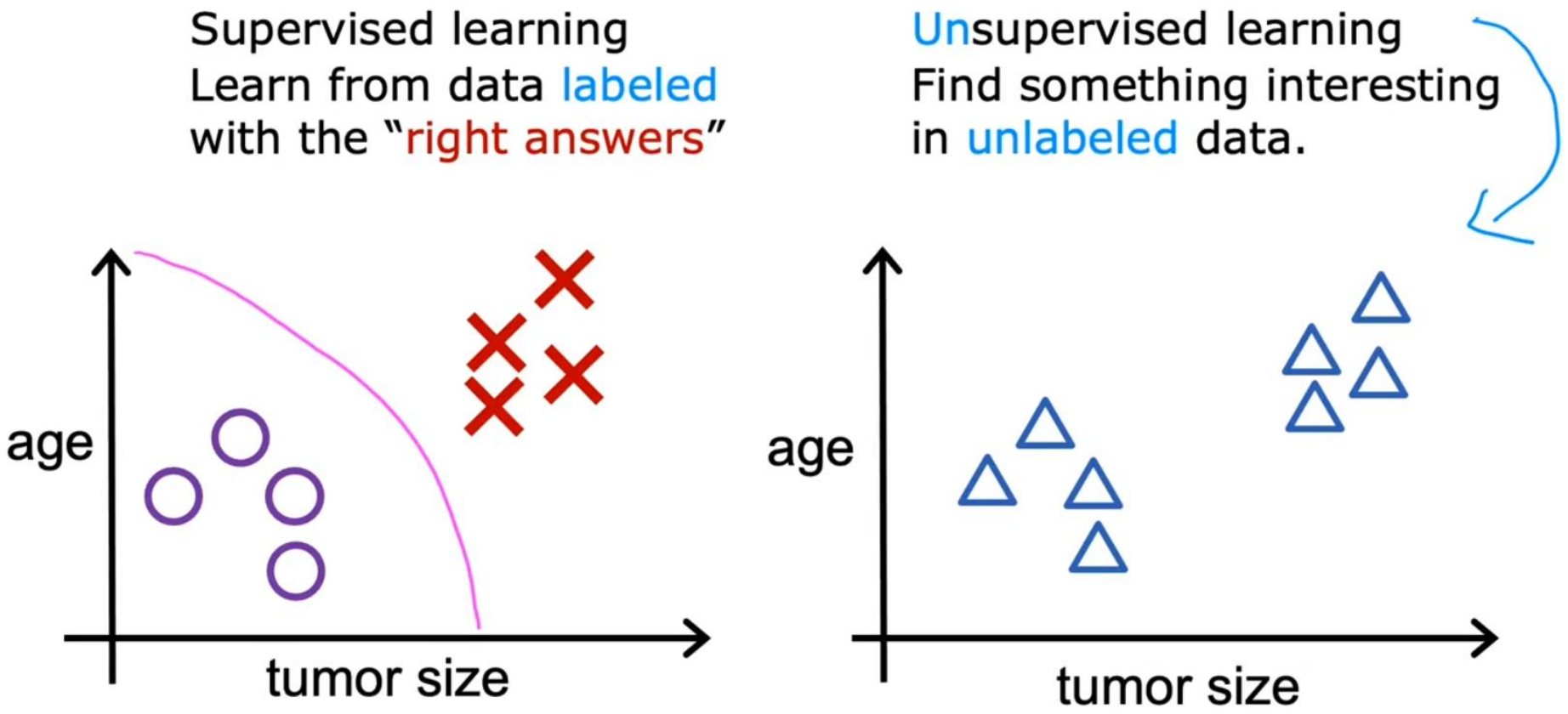
infinitely many possible outputs

Classification

predict categories

small number of possible outputs

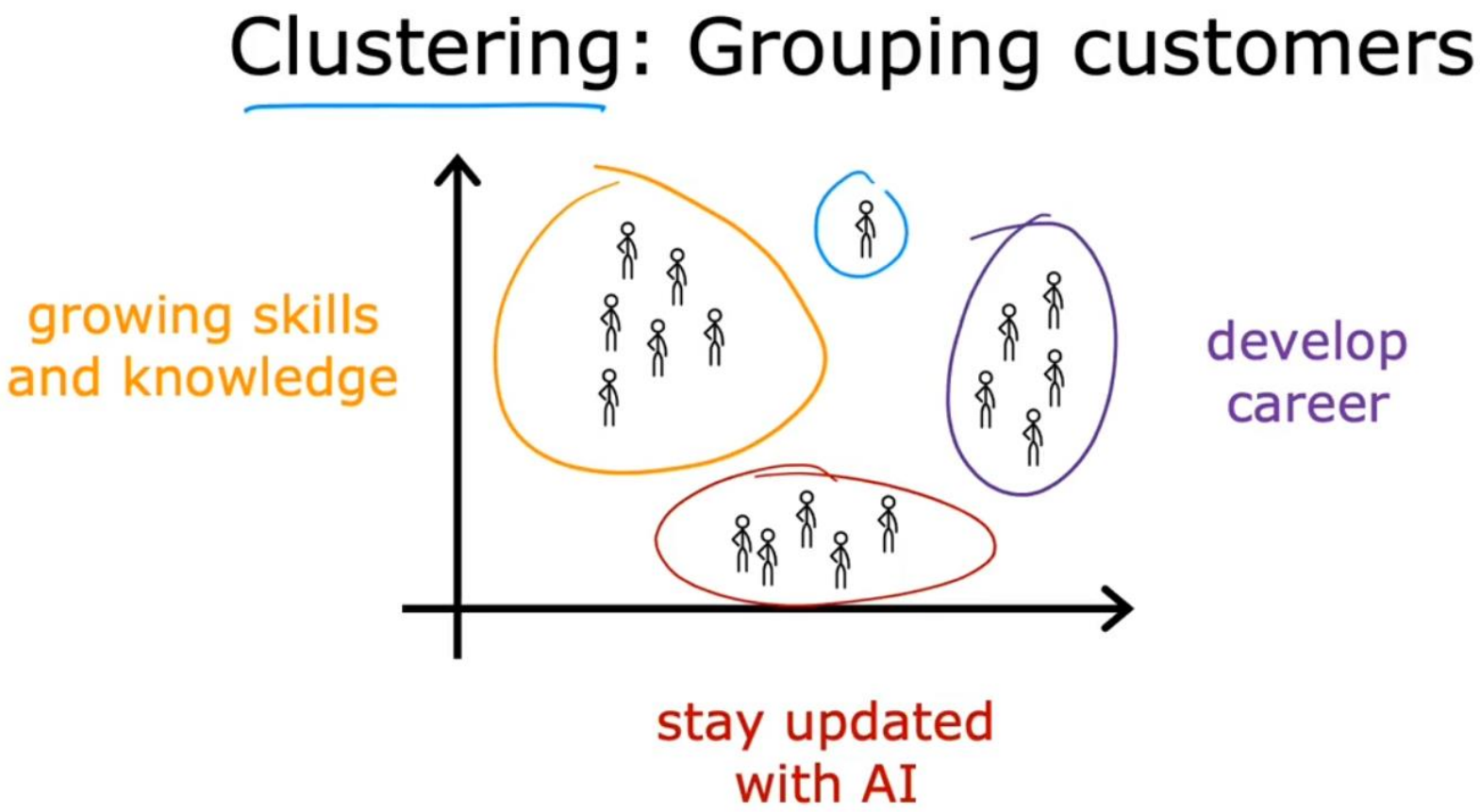
Unsupervised Machine Learning-Part 1



Unsupervised Machine Learning-Part 1

Clustering: Google news

Unsupervised Machine Learning-Part 1



Unsupervised Machine Learning-Part 1

Unsupervised Learning

a clustering algorithm is a type of unsupervised learning algorithm, which takes data without labels and tries to automatically group them into clusters

Unsupervised Machine Learning-Part 2

Unsupervised learning

Data only comes with inputs x, but not output labels y.
Algorithm has to find structure in the data.

Clustering
Group similar data points together.

Dimensionality reduction
Compress data using fewer numbers.

Anomaly detection
Find unusual data points.

Unsupervised Machine Learning-Part 2

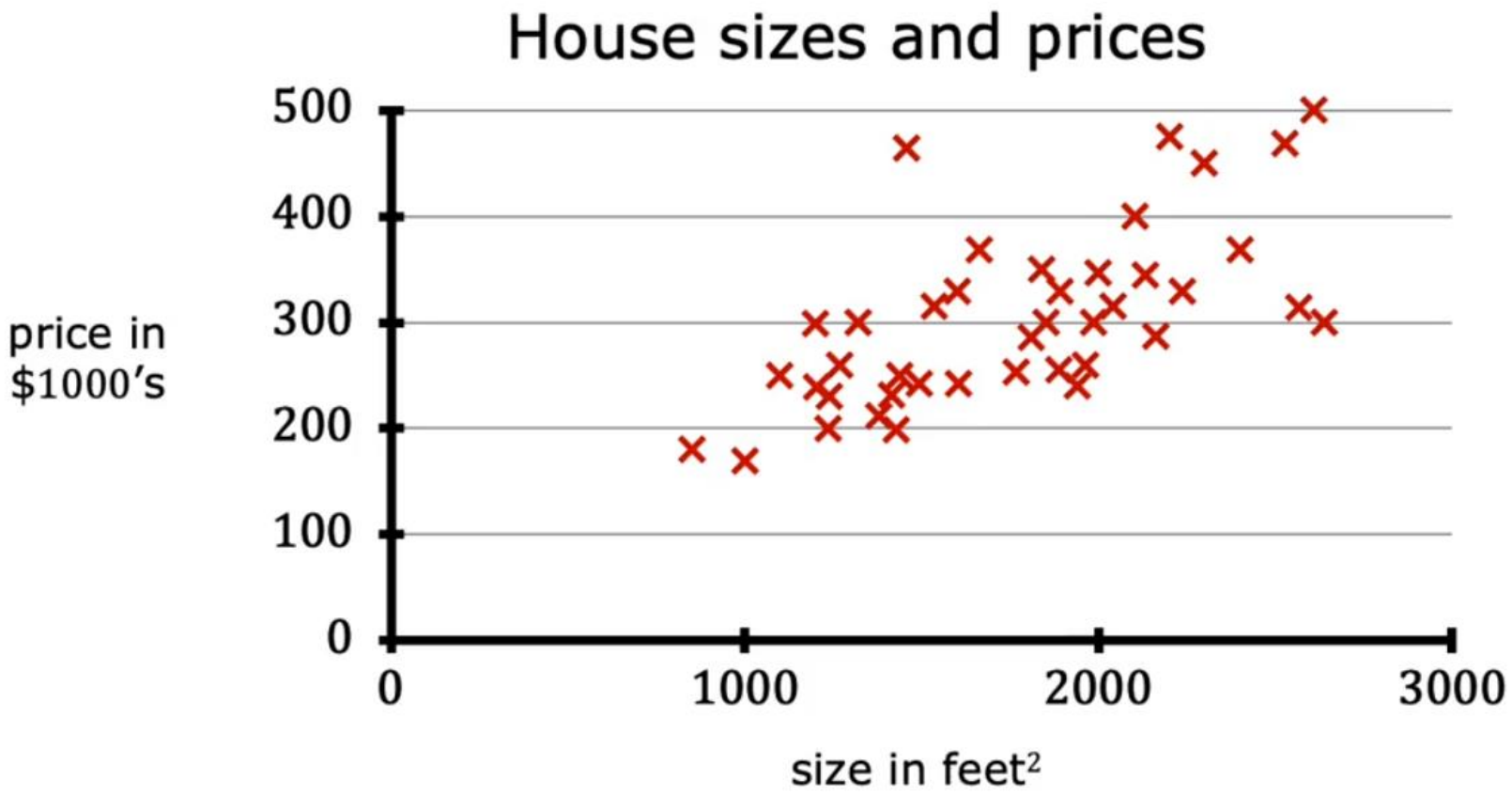
Question

Of the following examples, which would you address using an unsupervised learning algorithm?

- ☐ Given email labeled as spam/not spam, learn a spam filter.
- ☐ Given a set of news articles found on the web, group them into sets of articles about the same story.
- ☐ Given a database of customer data, automatically discover market segments and group customers into different market segments.
- ☐ Given a dataset of patients diagnosed as either having diabetes or not, learn to classify new patients as having diabetes or not

3. Regression Model

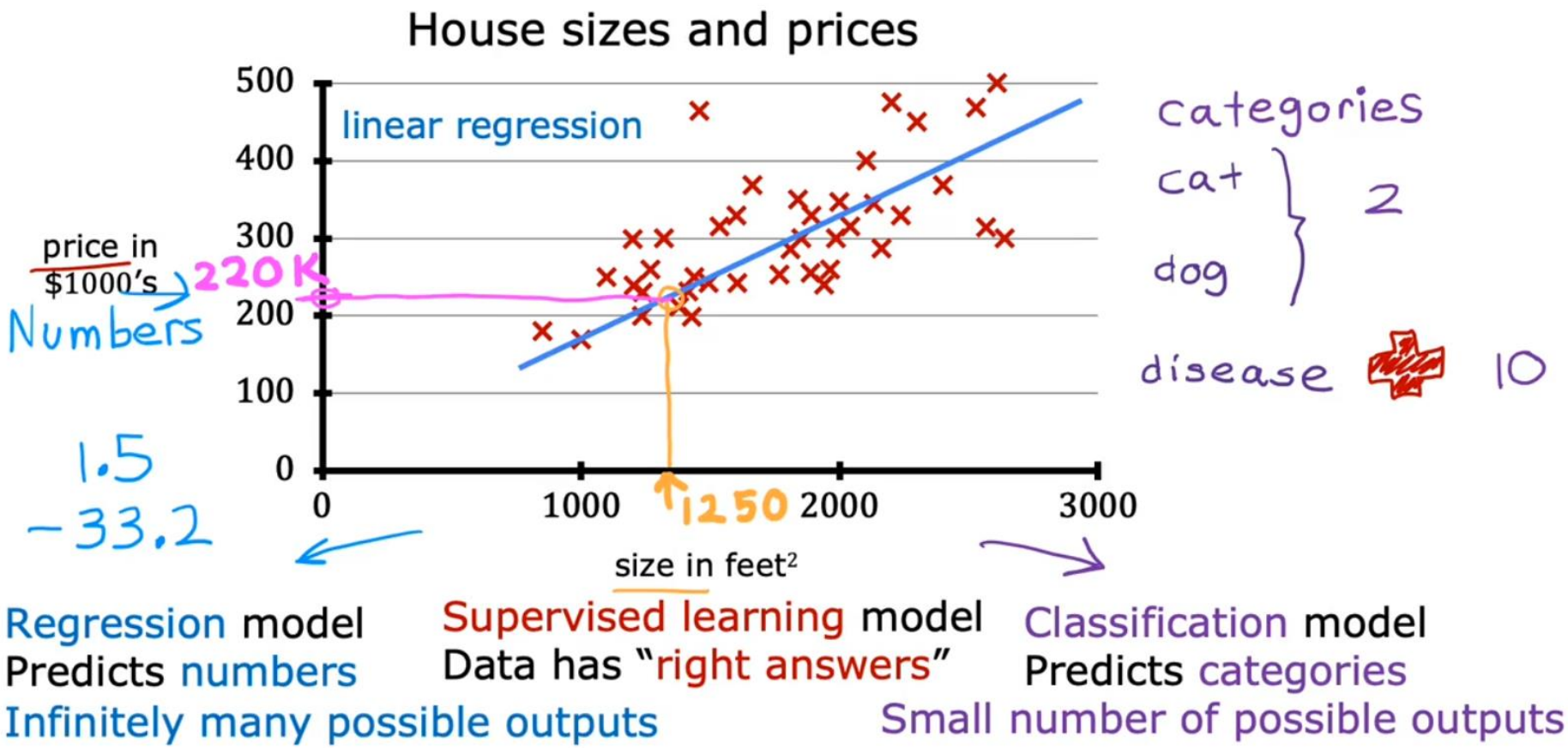
Linear Regression-Part 1



Linear Regression-Part 1

Supervised learning model
Data has "right answers"

Linear Regression-Part 1



Linear Regression-Part 1

Data table

size in feet²	price in $1000's
2104	400
1416	232
1534	315
852	178
...	...
3210	870

Linear Regression-Part 1

Terminology

Training set: Data used to train the model

	x size in feet^2	y price in \$1000's
(1)	2104	400
(2)	1416	232
(3)	1534	315
(4)	852	178
...	...	...
(47)	3210	870

$m = 47$

$x^{(1)} = 2104 \quad y^{(1)} = 400$

$(x^{(1)}, y^{(1)}) = (2104, 400)$

$x^{(2)} = 1416 \quad x^{(2)} \neq x^2$ not exponent

Notation:

x = "input" variable
feature

y = "output" variable
"target" variable

m = number of training examples

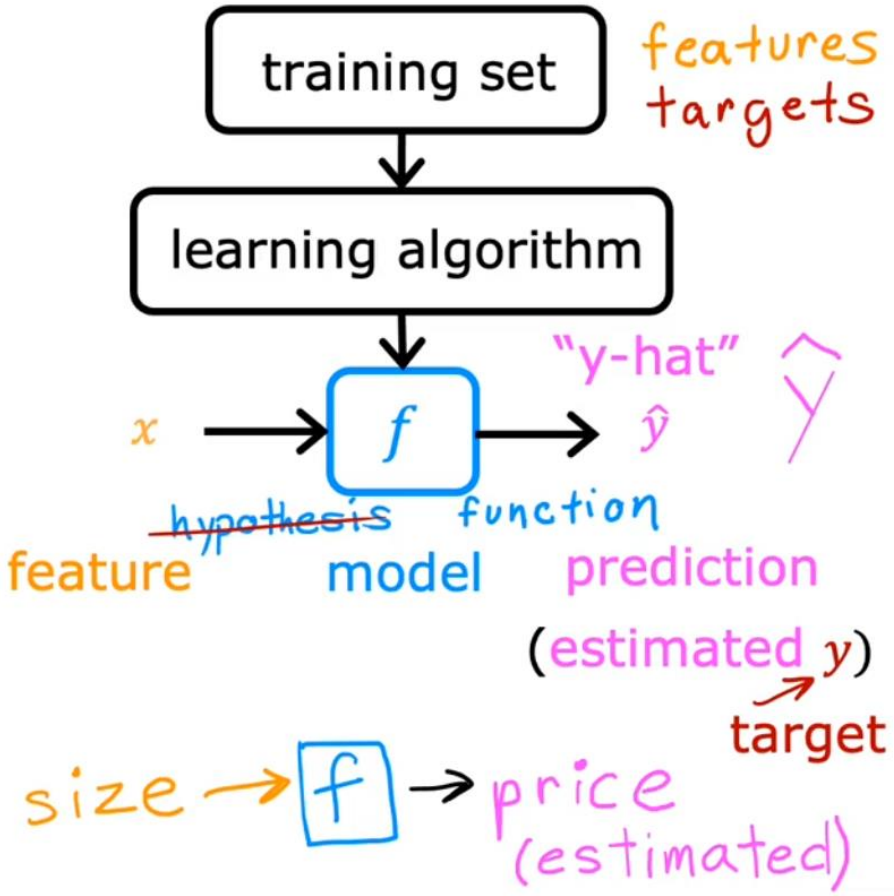
(x, y) = single training example

$(x^{(i)}, y^{(i)})$

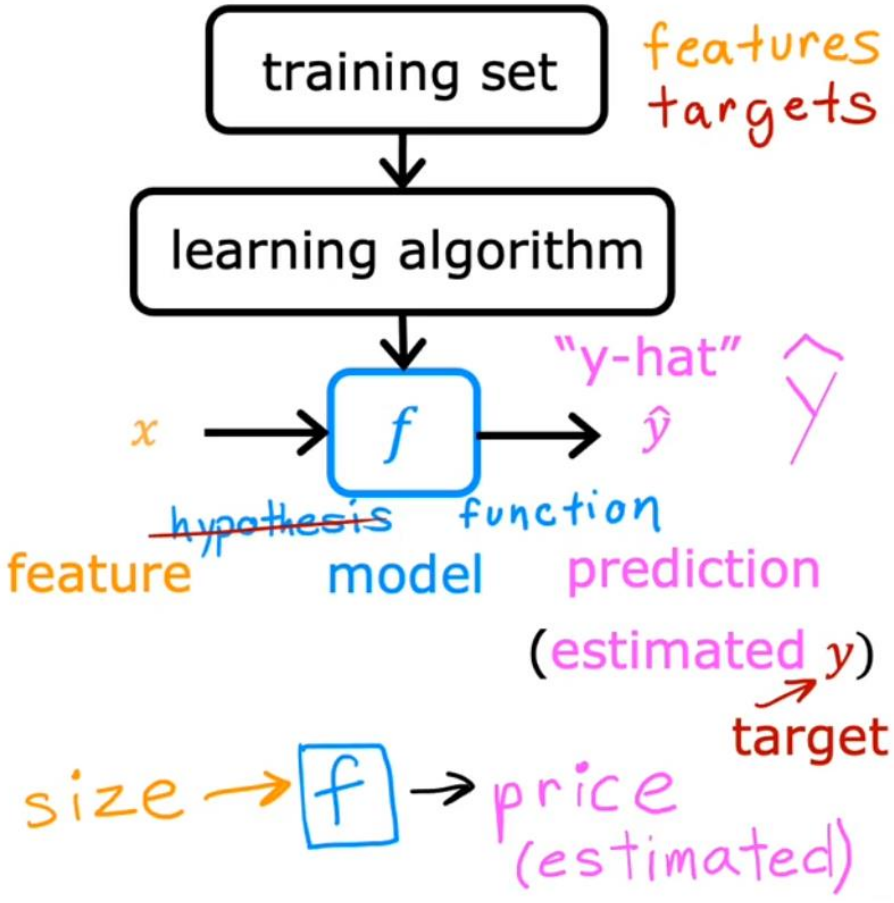
$(x^{(i)}, y^{(i)})$ = i^{th} training example

index (1^{st}, 2^{nd}, 3^{rd} ...)

Linear Regression-Part 2



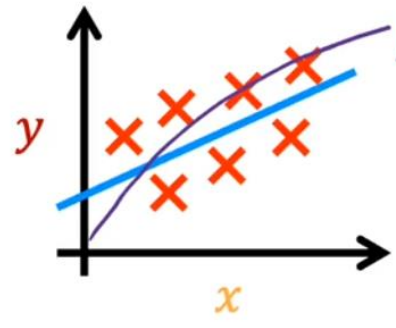
Linear Regression-Part 2



How to represent f?

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

$f(x)$



Linear regression with one variable.

size

Univariate linear regression.

one variable

Cost Function Formula

Training set

features size in feet² (x)	targets price \$1000's (y)
2104	460
1416	232
1534	315
852	178
...	...

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

w, b: parameters

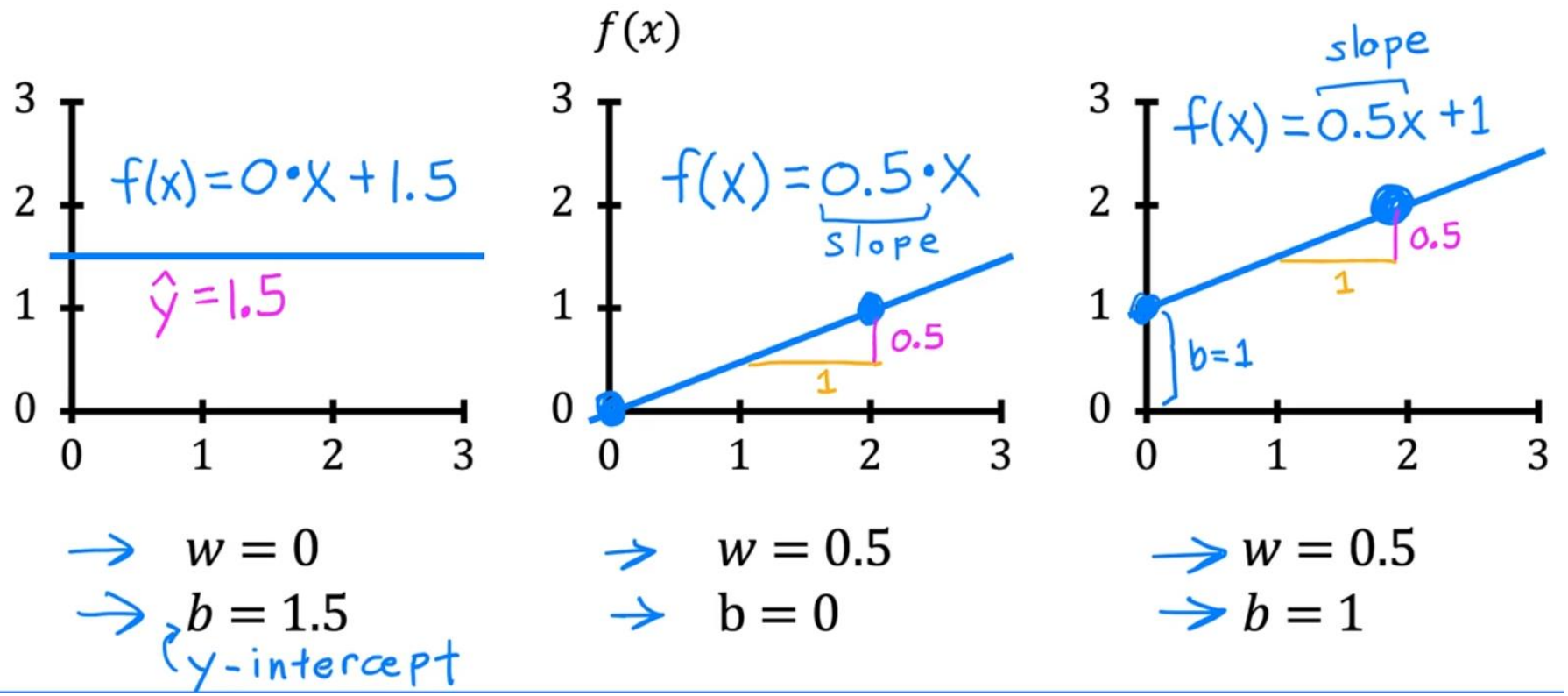
coefficients

weights

What do w, b do?

Cost Function Formula

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

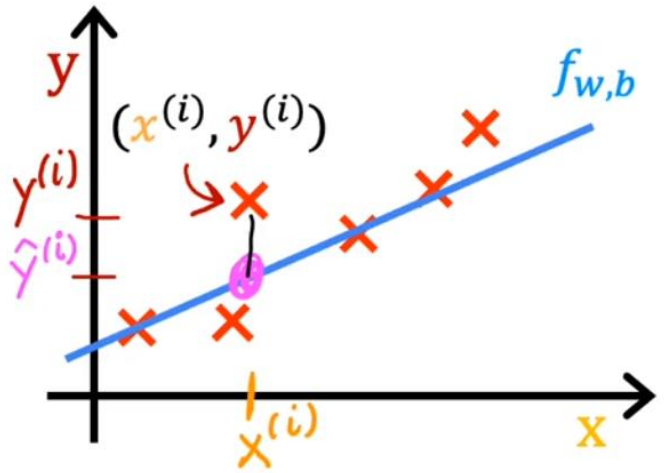


→ $w = 0$
→ $b = 1.5$
(y-intercept)

→ $w = 0.5$
→ $b = 0$

→ $w = 0.5$
→ $b = 1$

Cost Function Formula



$$\hat{y}^{(i)} = f_{w,b}(x^{(i)})$$

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

Cost function: Squared error cost function

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

$\hat{y}^{(i)}$ error

m = number of training examples

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

intuition (next!)

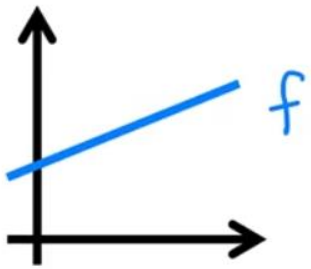
Find w, b:

$\hat{y}^{(i)}$ is close to $y^{(i)}$ for all $(x^{(i)}, y^{(i)})$.

Cost Function Intuition

model:

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



parameters:

w, b

cost function:

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

goal:

$$\underset{w,b}{\text{minimize}}\, J(w,b)$$

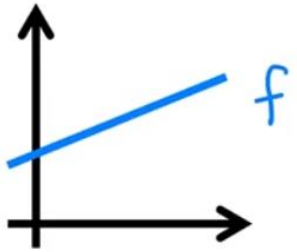
Cost Function Intuition

model:

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

parameters:

w, b



cost function:

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

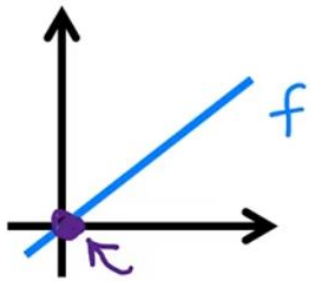
goal:

$$\underset{w,b}{\text{minimize}}\, J(w,b)$$

simplified

$f_w(x) = wx$ $\quad b = \emptyset$

w

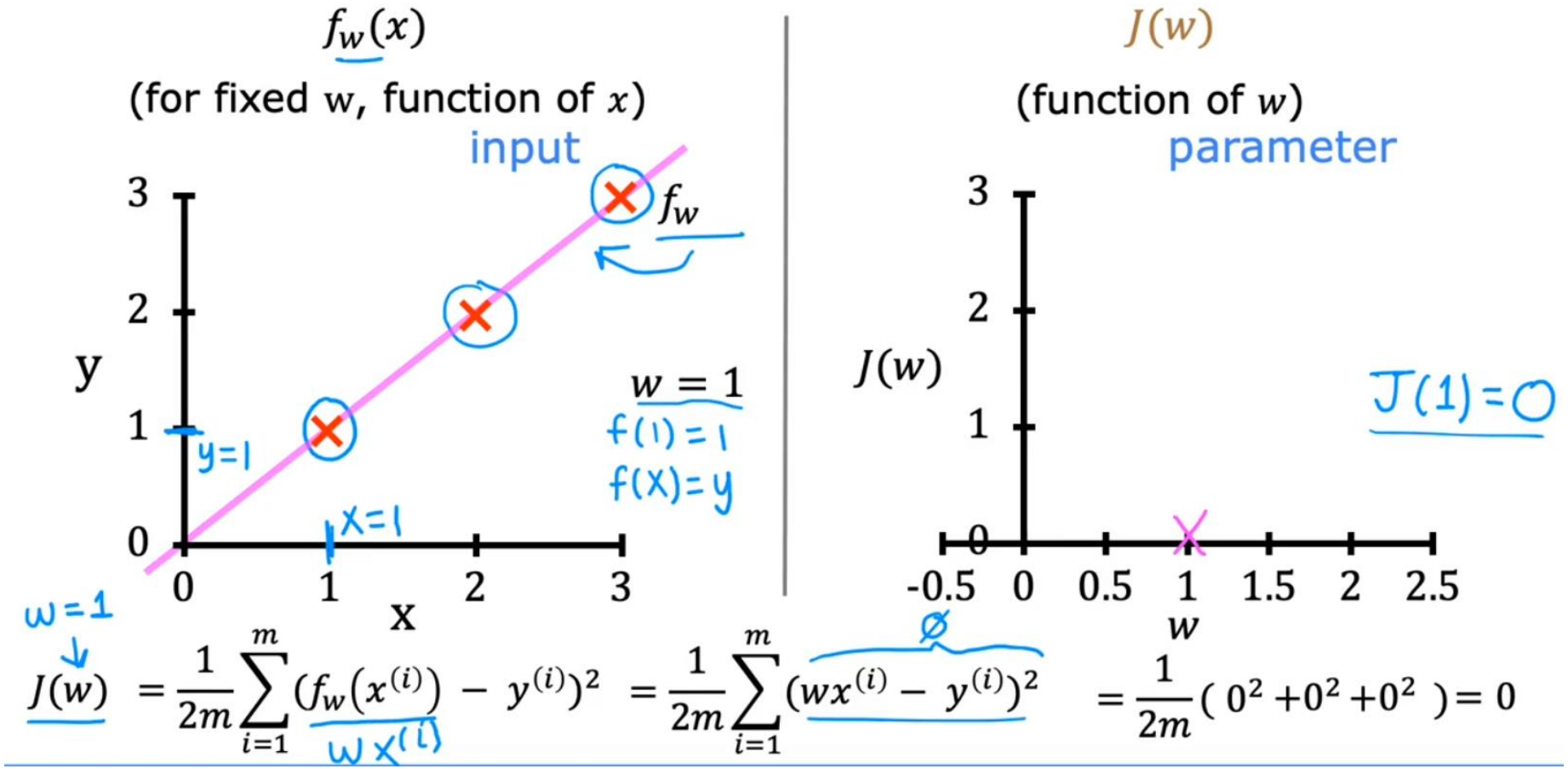


$$J(w) = \frac{1}{2m}\sum_{i=1}^{m}(f_w(x^{(i)}) - y^{(i)})^2$$

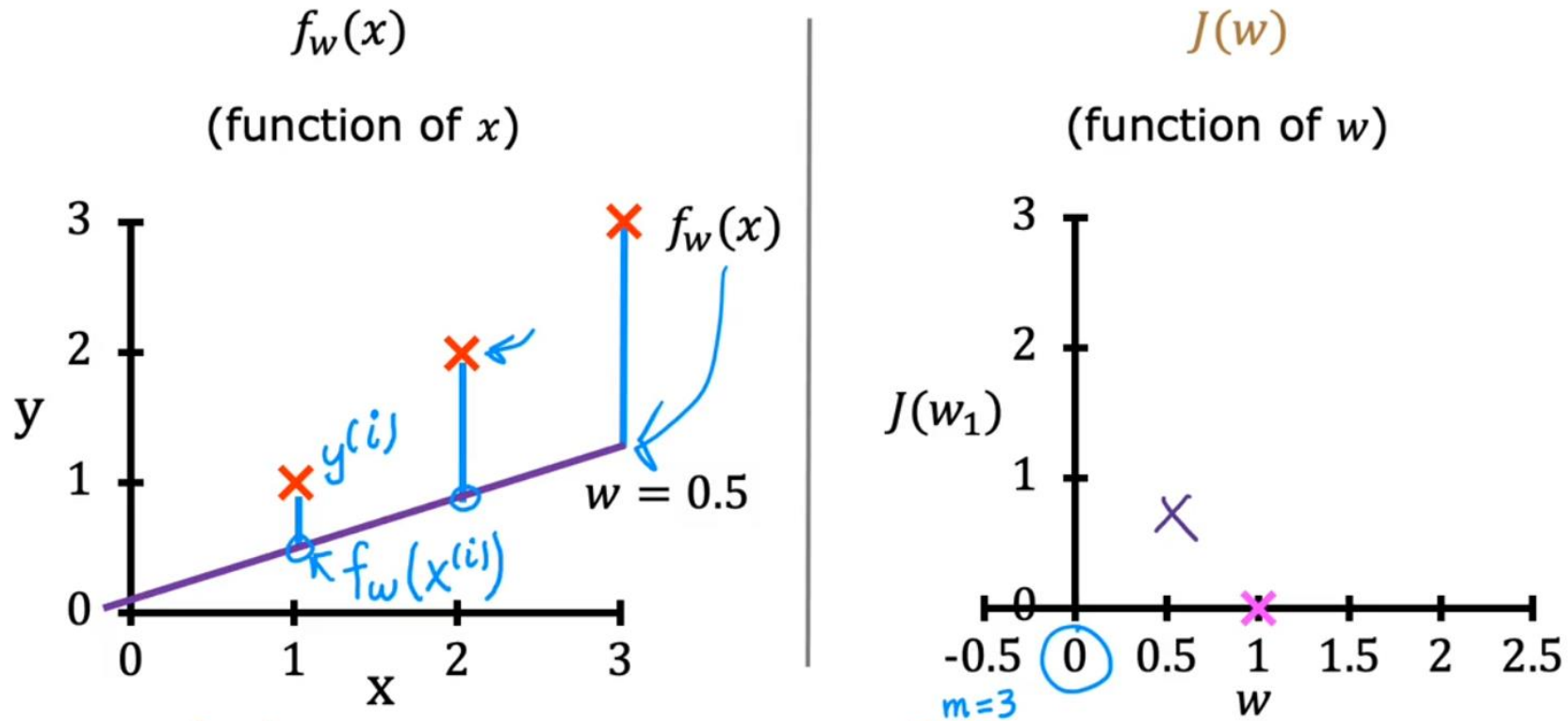
$wx^{(i)}$

$$\underset{w}{\text{minimize}}\, J(w)$$

Cost Function Intuition



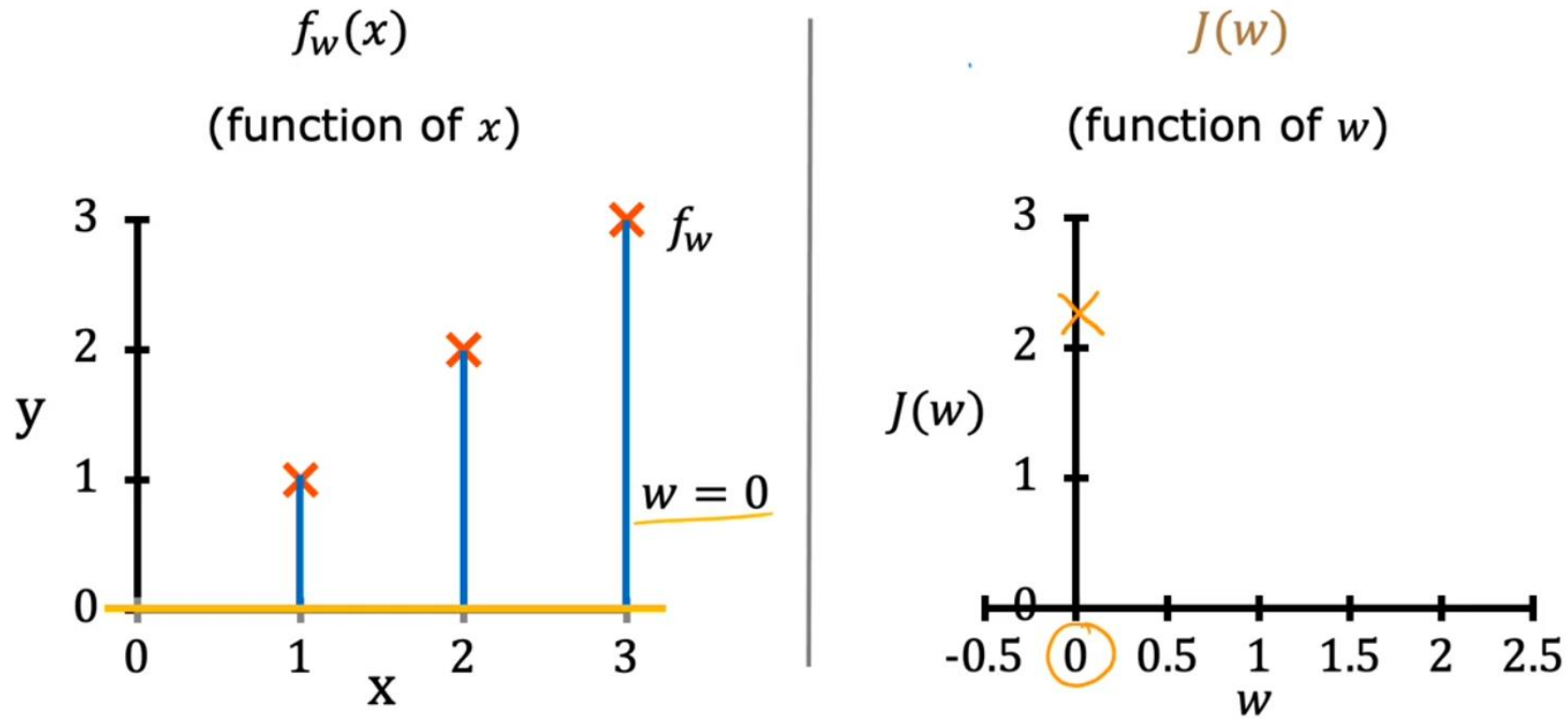
Cost Function Intuition



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

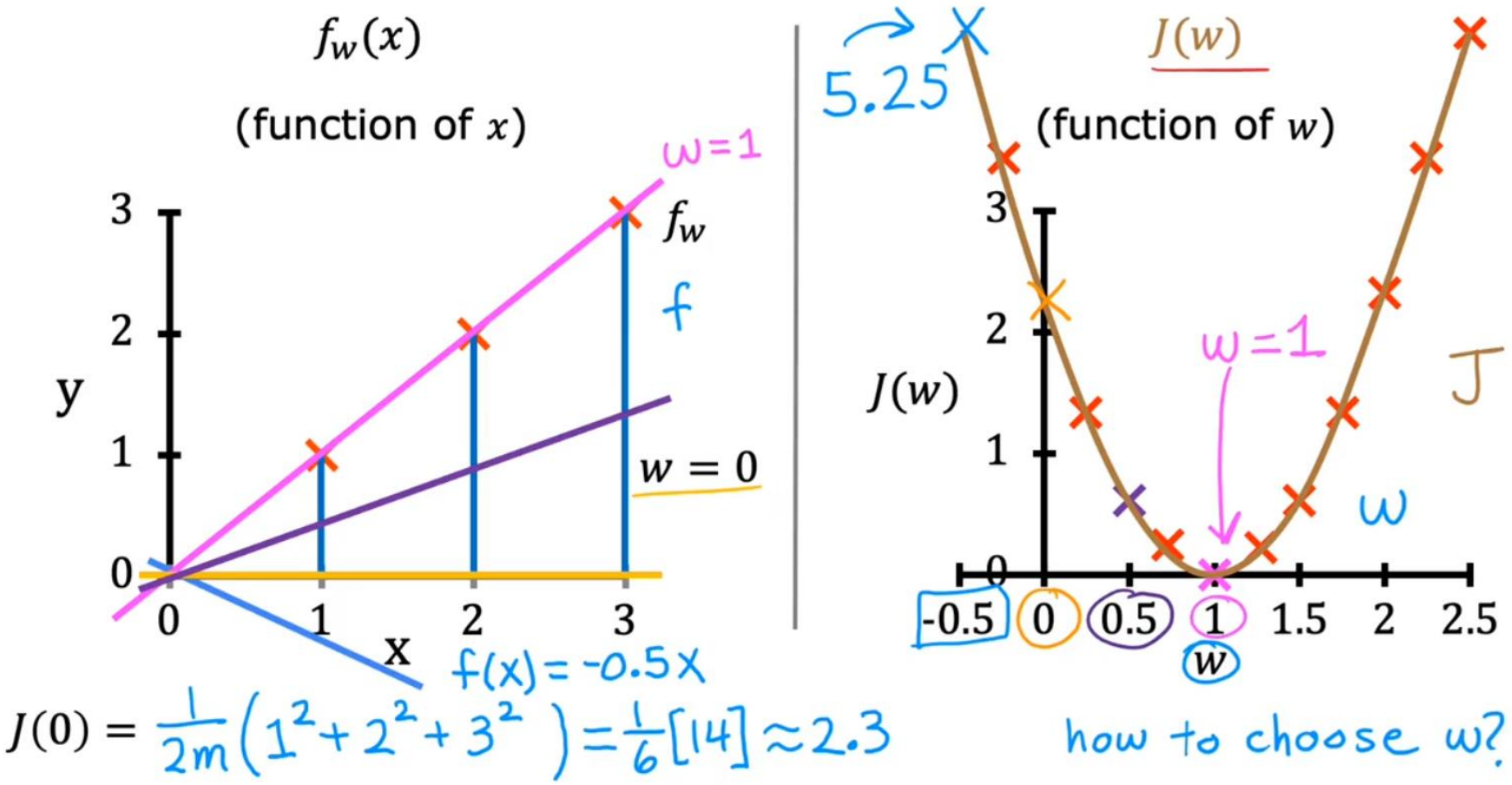
$m=3$

Cost Function Intuition



$J(0) = \frac{1}{2m}(1^2 + 2^2 + 3^2) = \frac{1}{6}[14] \approx 2.3$

Cost Function Intuition



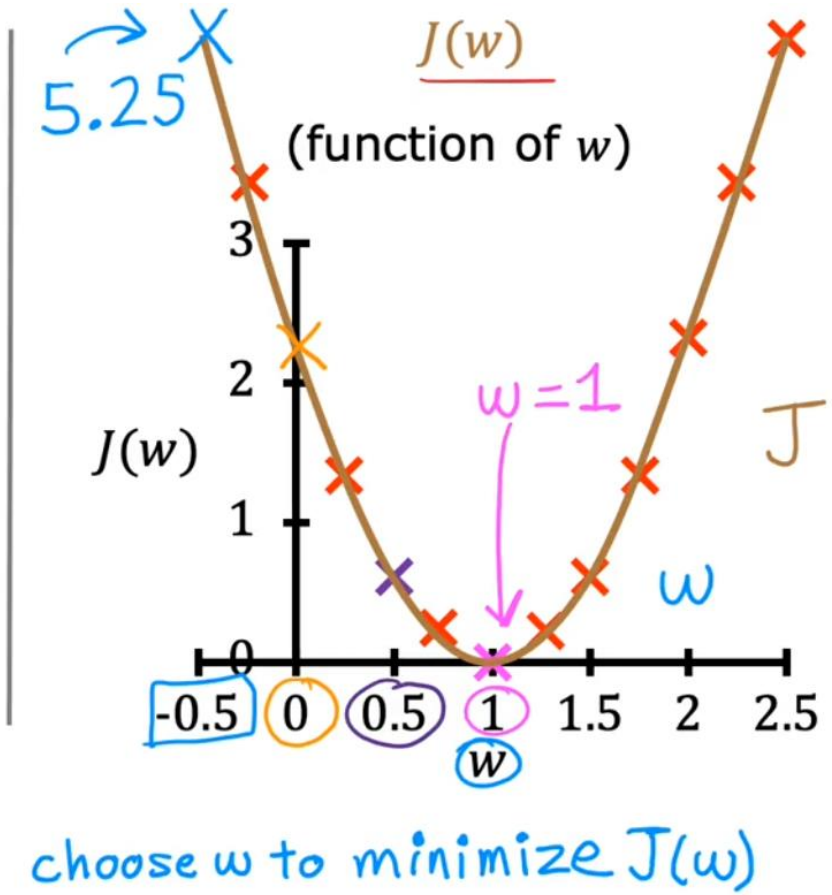
Cost Function Intuition

goal of linear regression:

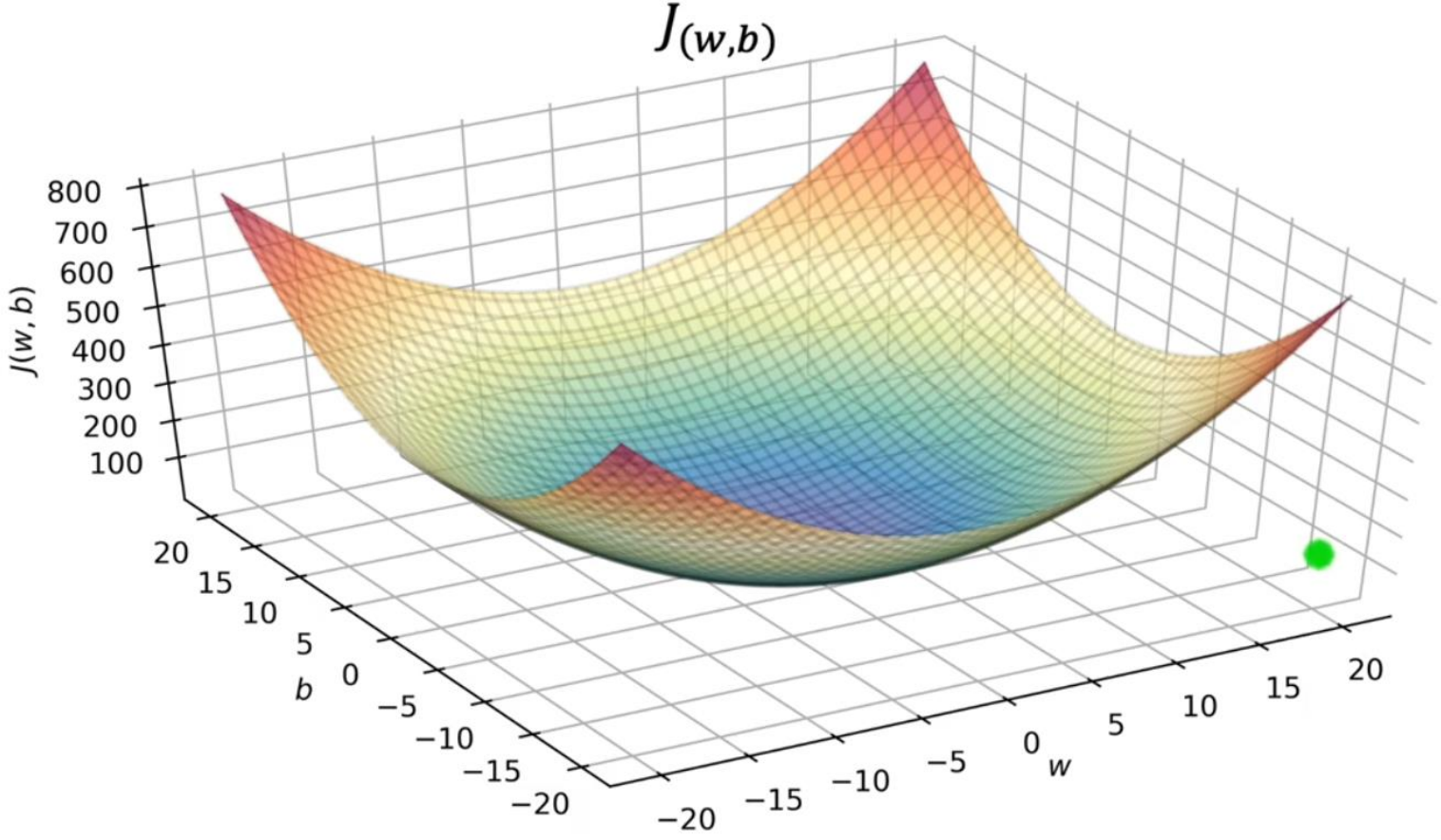
$$\underset{w}{\text{minimize}}\, J(w)$$

general case:

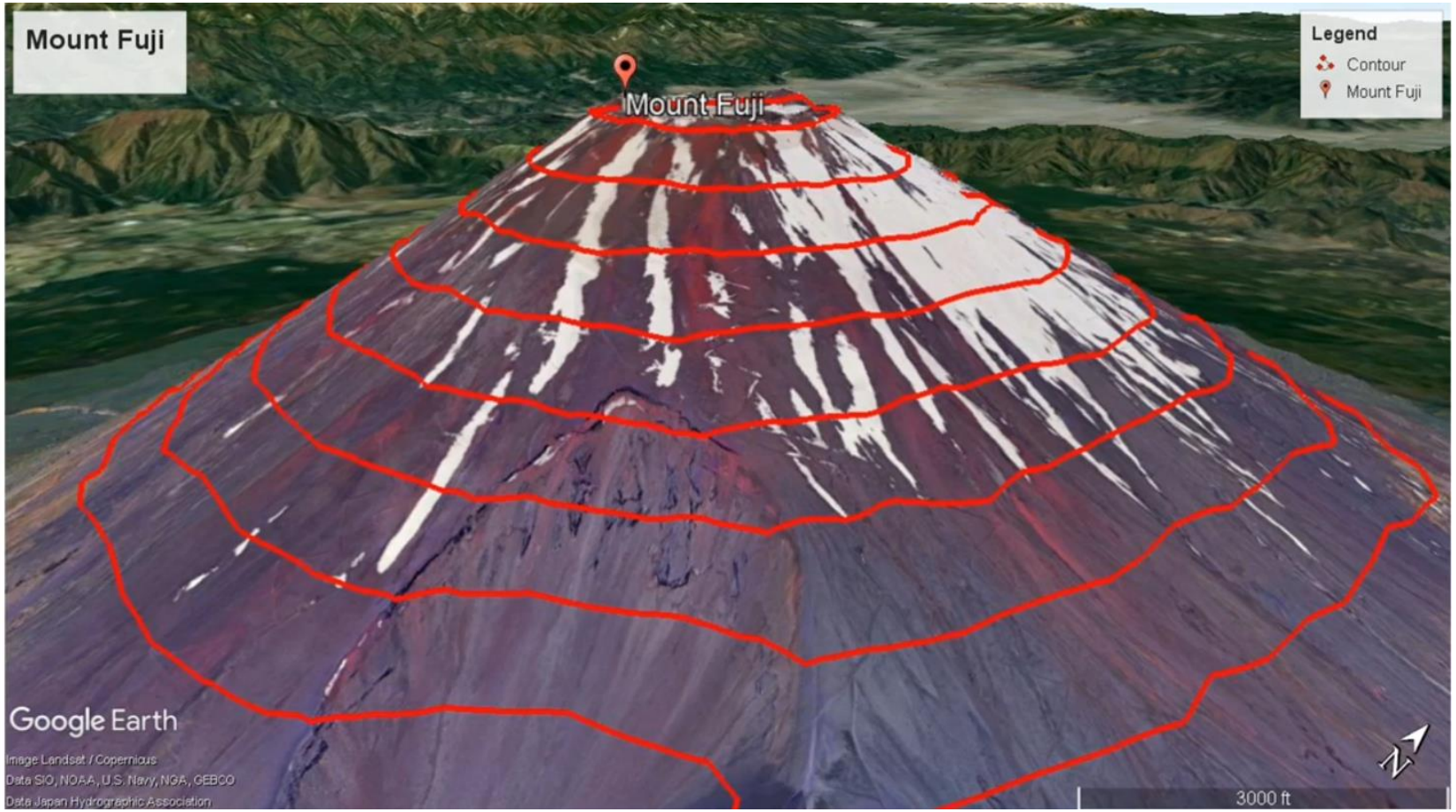
$$\underset{w,b}{\text{minimize}}\, J(w,b)$$



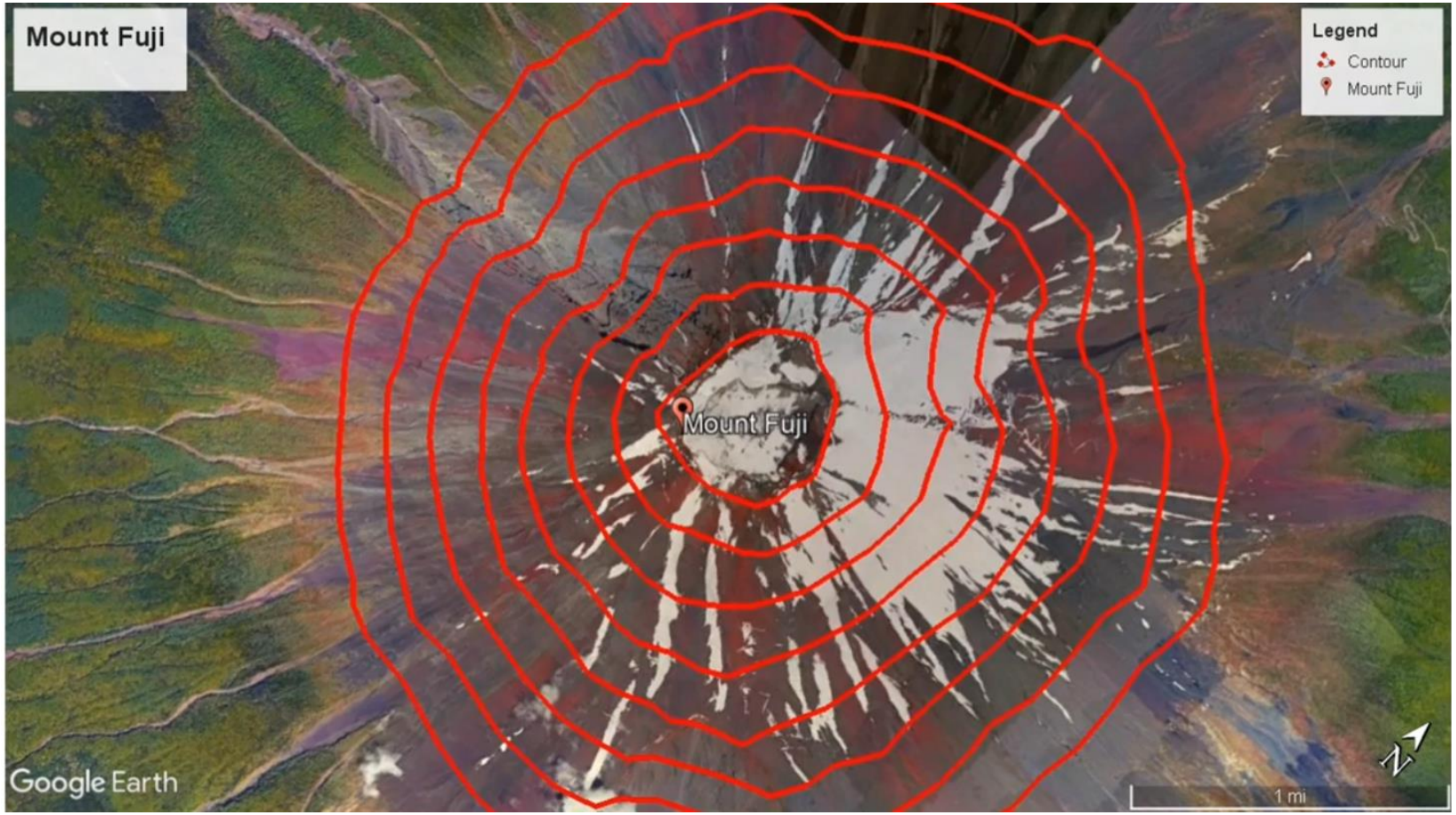
Visualizing the Cost Function



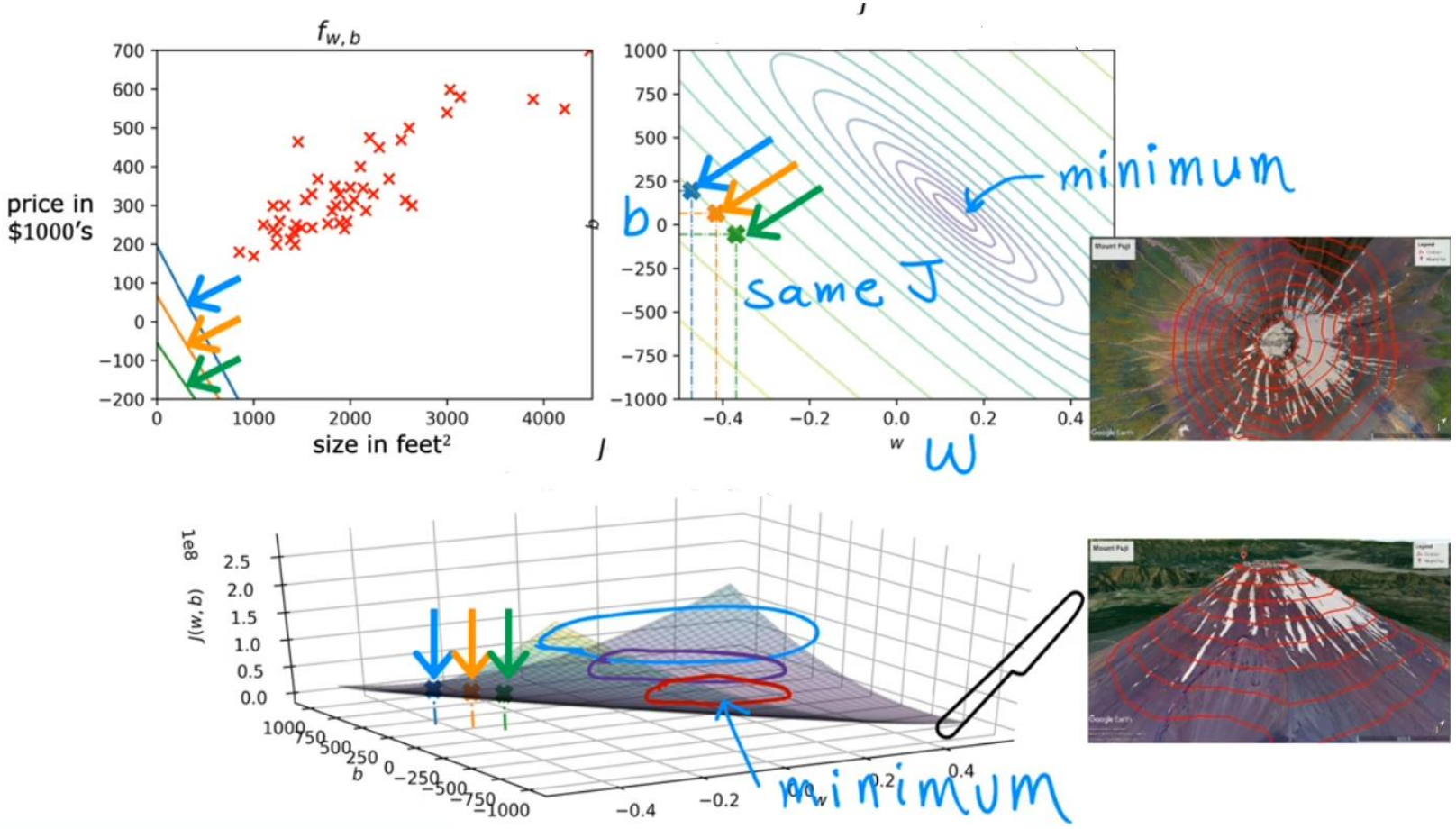
Visualizing the Cost Function



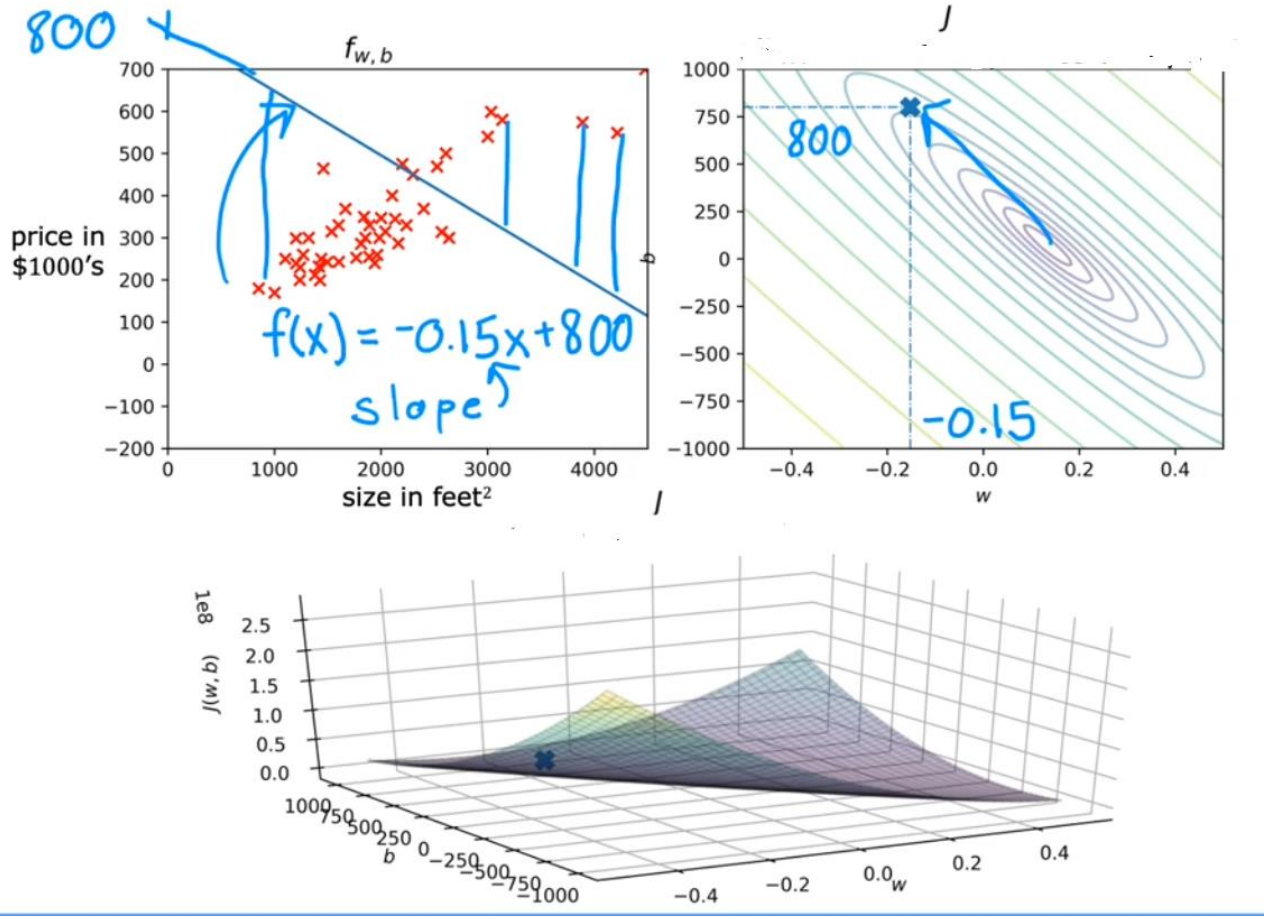
Visualizing the Cost Function



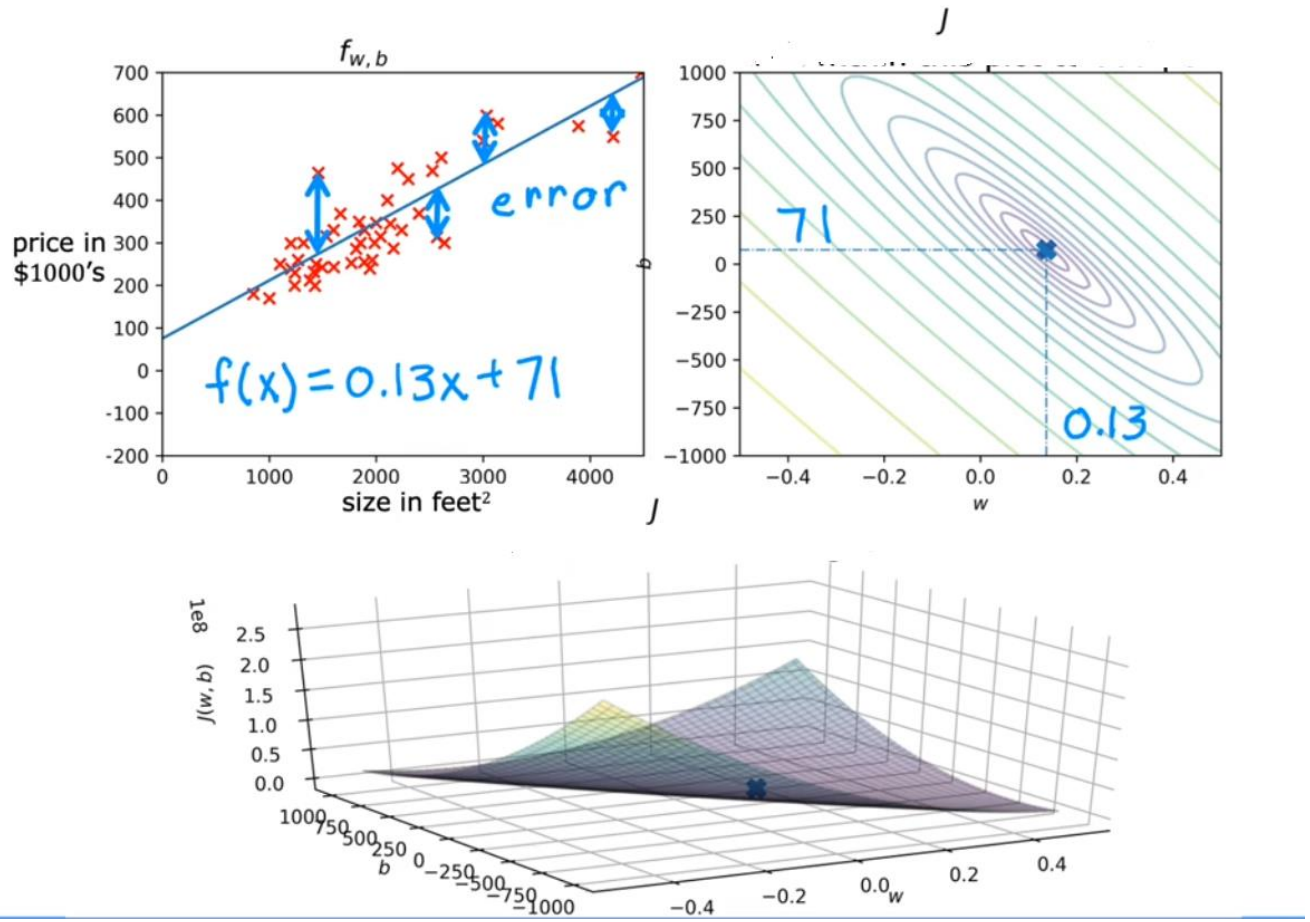
Visualizing the Cost Function



Visualization Examples



Visualization Examples



4. Train the Model with Gradient Descent

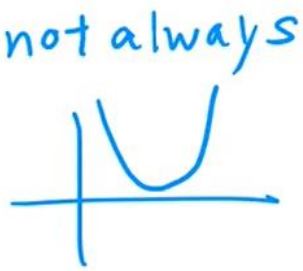
Gradient Descent

Have some function $J(w, b)$ for linear regression or any function

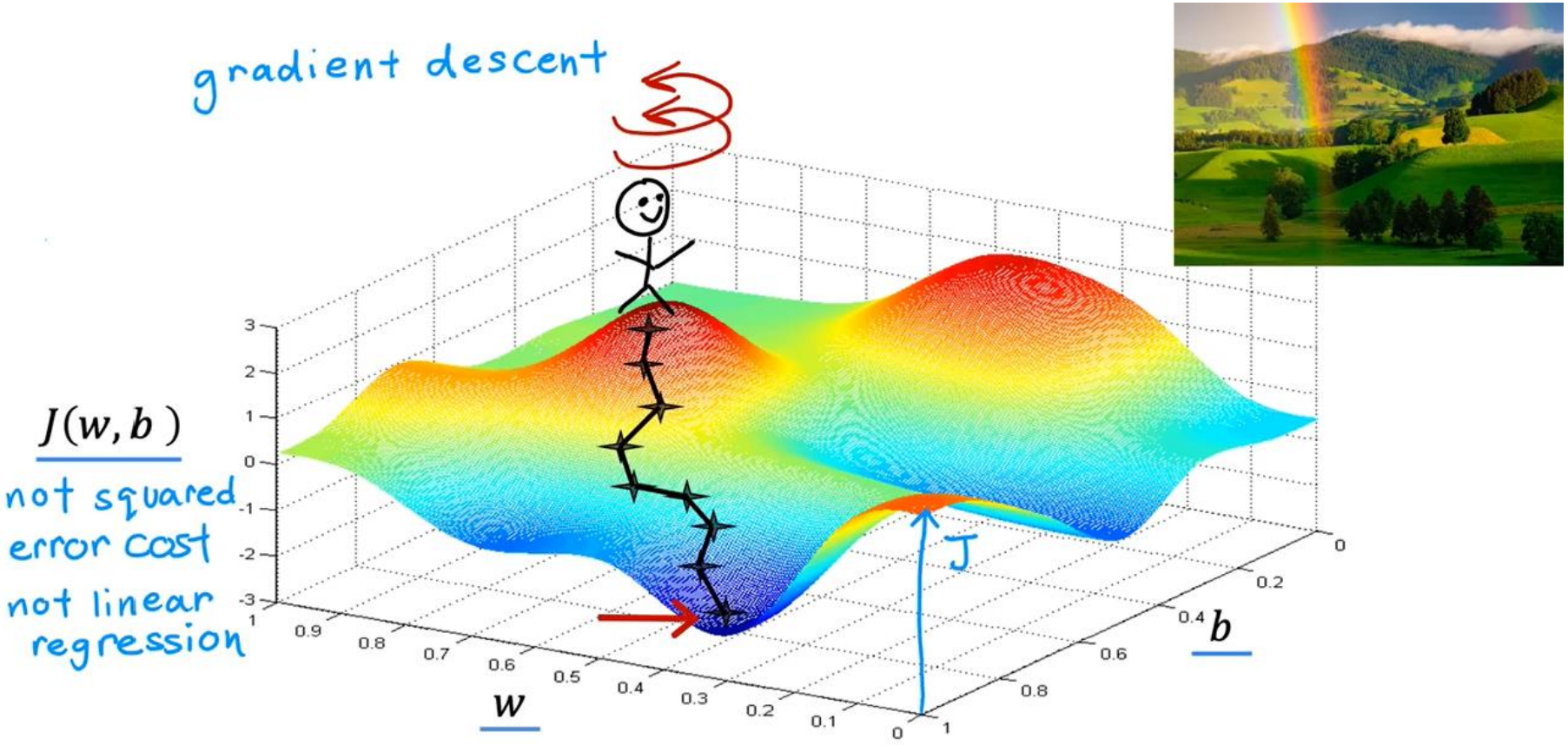
Want $\min_{w,b} J(w, b)$ $\min_{w_1, \ldots, w_n, b} J(w_1, w_2, \ldots, w_n, b)$

Outline:

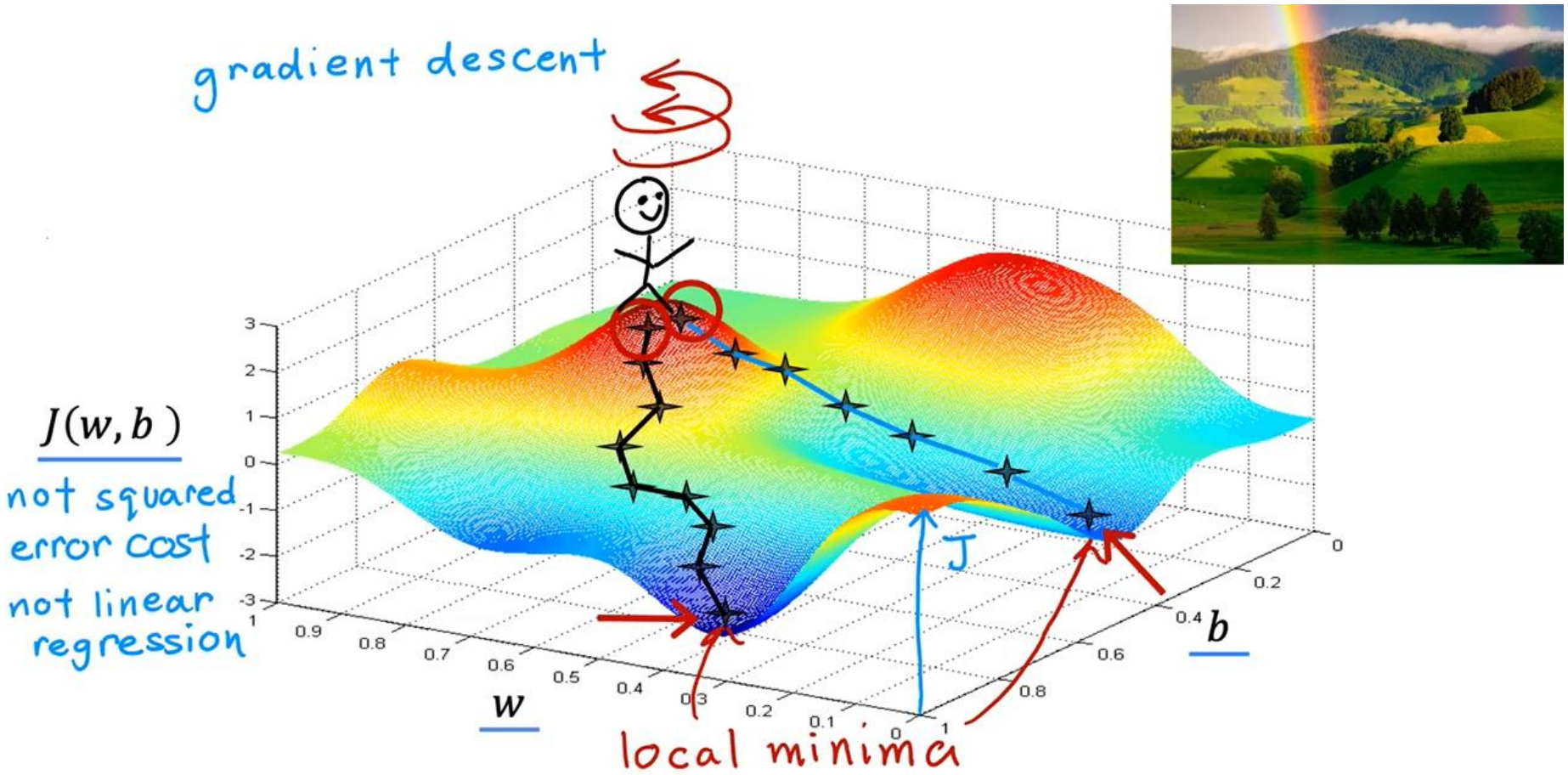
- Start with some w, b (set $w=0$, $b=0$)
- Keep changing w, b to reduce $J(w, b)$
- Until we settle at or near a minimum



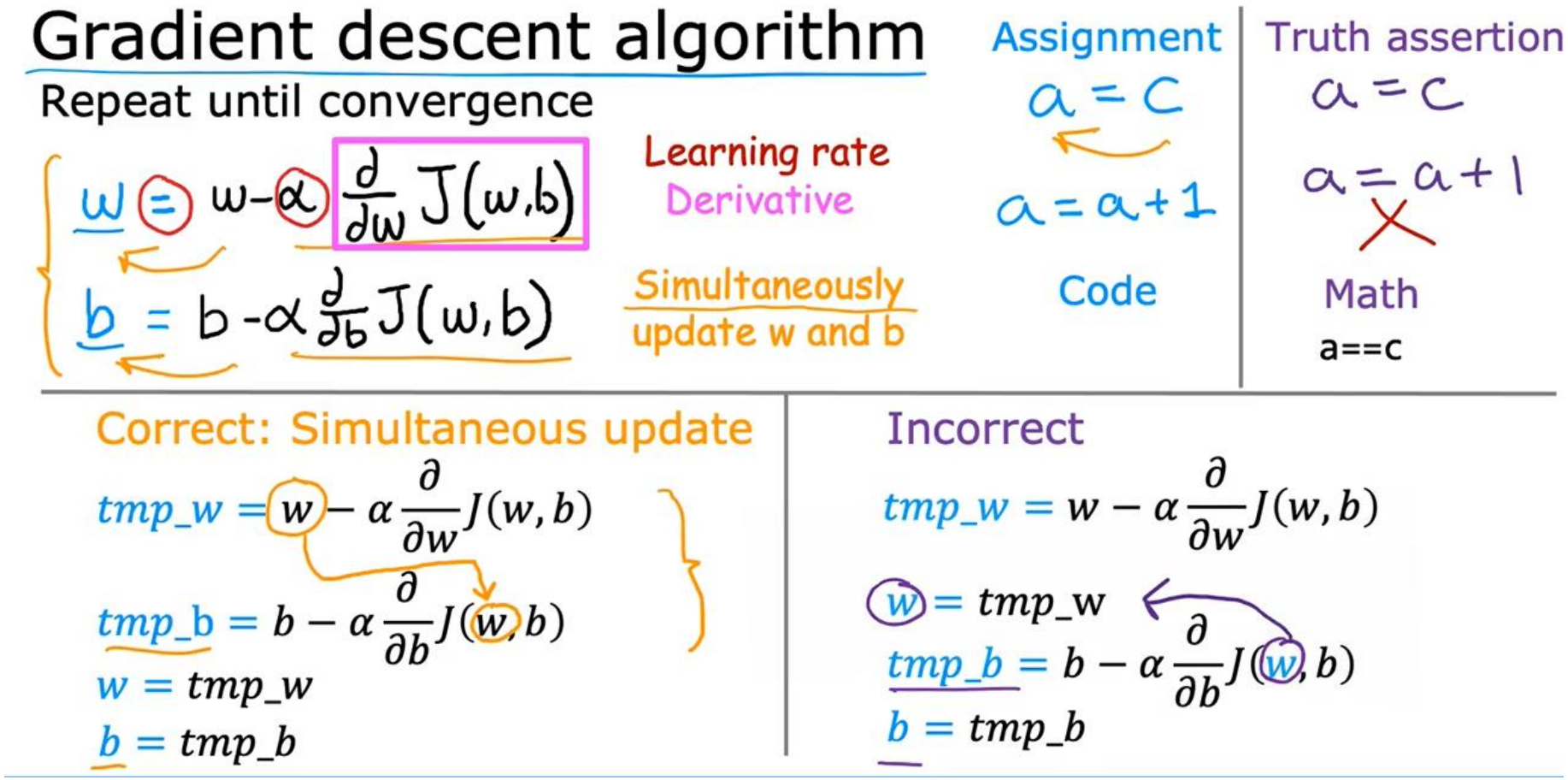
Gradient Descent



Gradient Descent



Implementing Gradient Descent



Gradient Descent Intuition

Gradient descent algorithm

repeat until convergence {

learning rate

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

derivative

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

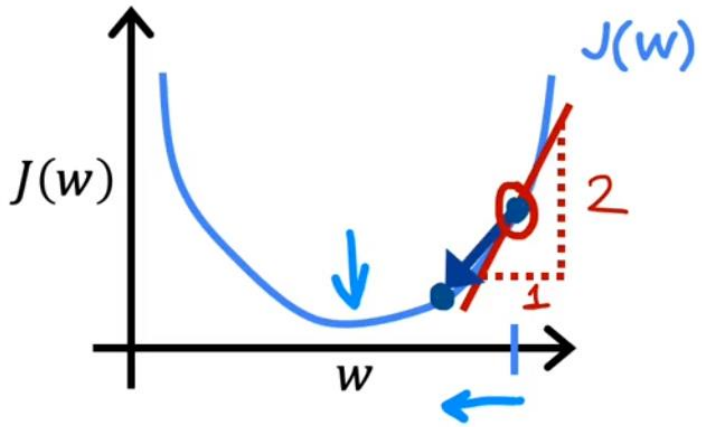
}

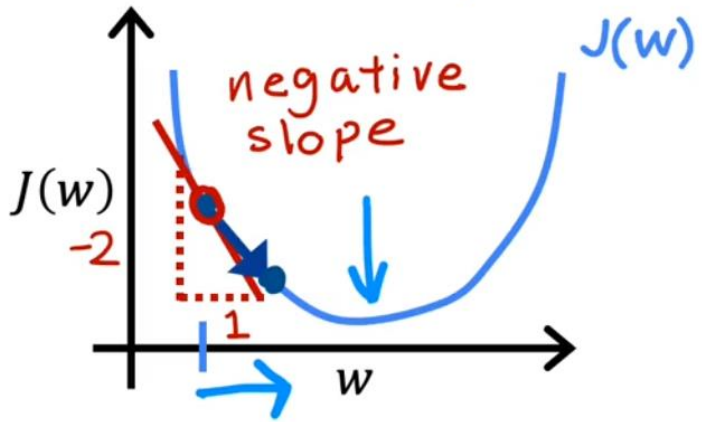
$J(w)$

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

$$\min_{w} J(w)$$

Gradient Descent Intuition





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

$$> 0$$

$$w = w - \alpha \cdot (positive\ number)$$

$$\frac{d}{dw} J(w) < 0$$

$$w = w - \alpha \cdot (negative\ number)$$

Learning Rate

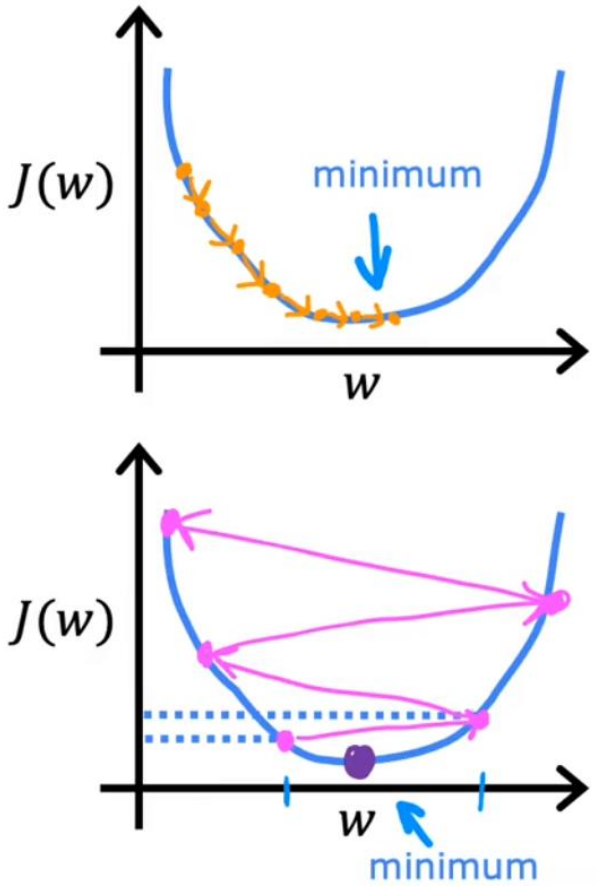
$$w = w - \alpha \frac{d}{dw} J(w)$$

If α is too small...

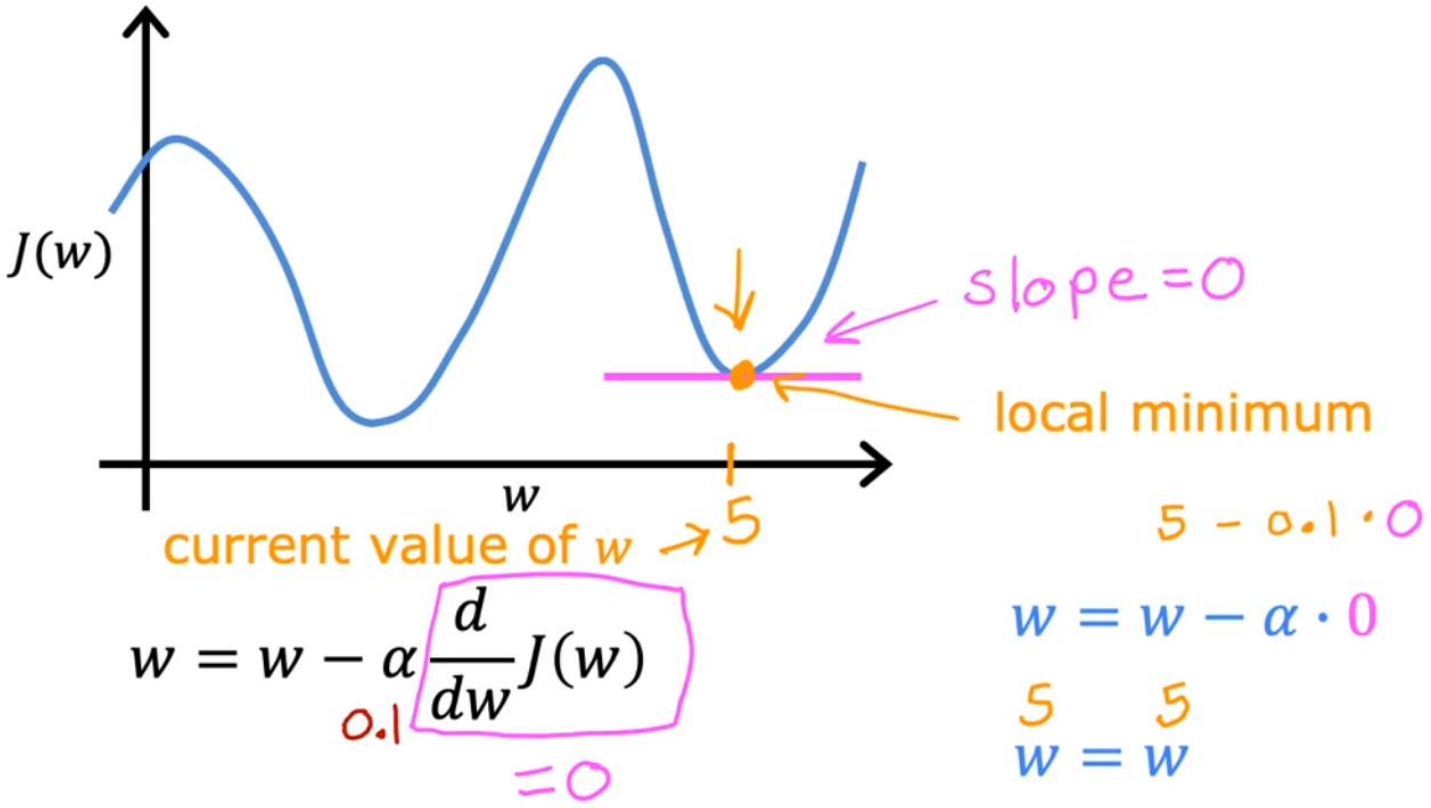
Gradient descent may be slow.

If α is too large...

Gradient descent may:
- Overshoot, never reach minimum
- Fail to converge, diverge



Learning Rate



Learning Rate

Can reach local minimum with fixed learning rate

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

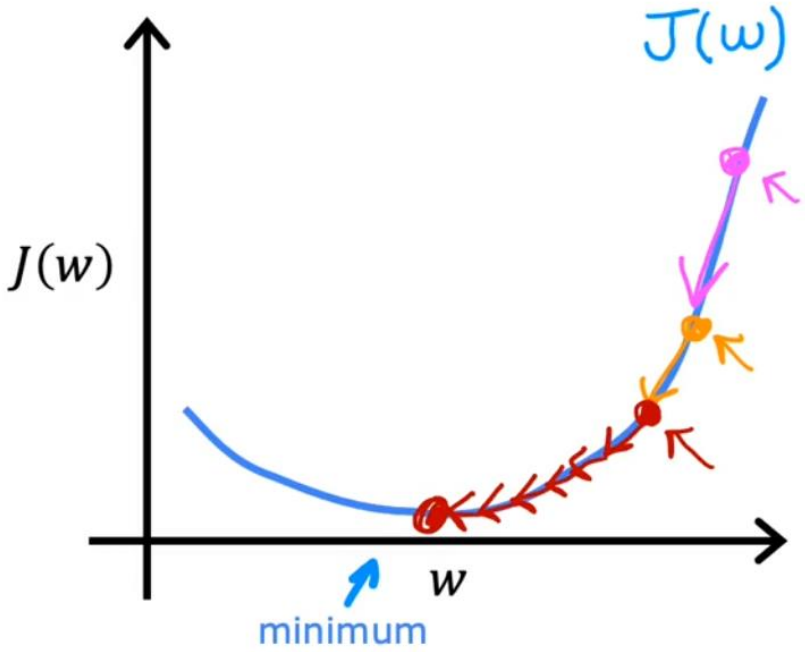
smaller

not as large

large

Near a local minimum,
- Derivative becomes smaller
- Update steps become smaller

Can reach minimum without decreasing learning rate α



Gradient Descent for Linear Regression

Linear regression model

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

Cost function

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

Gradient descent algorithm

repeat until convergence {

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

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

}

Gradient Descent for Linear Regression

(Optional)

$$\frac{\partial}{\partial w} J(w,b) = \frac{d}{dw} \frac{1}{2m} \sum_{i=1}^{m} \left(f_{w,b}(x^{(i)}) - y^{(i)}\right)^2 = \frac{d}{dw} \frac{1}{2m} \sum_{i=1}^{m} \left(wx^{(i)} + b - y^{(i)}\right)^2$$

$$= \frac{1}{2m} \sum_{i=1}^{m} \left(wx^{(i)} + b - y^{(i)}\right) 2x^{(i)} = \frac{1}{m} \sum_{i=1}^{m} \left(f_{w,b}(x^{(i)}) - y^{(i)}\right) x^{(i)}$$

$$\frac{\partial}{\partial b} J(w,b) = \frac{d}{db} \frac{1}{2m} \sum_{i=1}^{m} \left(f_{w,b}(x^{(i)}) - y^{(i)}\right)^2 = \frac{d}{db} \frac{1}{2m} \sum_{i=1}^{m} \left(wx^{(i)} + b - y^{(i)}\right)^2$$

$$= \frac{1}{2m} \sum_{i=1}^{m} \left(wx^{(i)} + b - y^{(i)}\right) 2 = \frac{1}{m} \sum_{i=1}^{m} \left(f_{w,b}(x^{(i)}) - y^{(i)}\right)$$

no $x^{(i)}$

Gradient Descent for Linear Regression

Gradient descent algorithm

repeat until convergence {

$$w = w - \alpha \frac{1}{m} \sum_{i=1}^{m} (f_{w,b}(x^{(i)}) - y^{(i)}) \; x^{(i)}$$

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

}

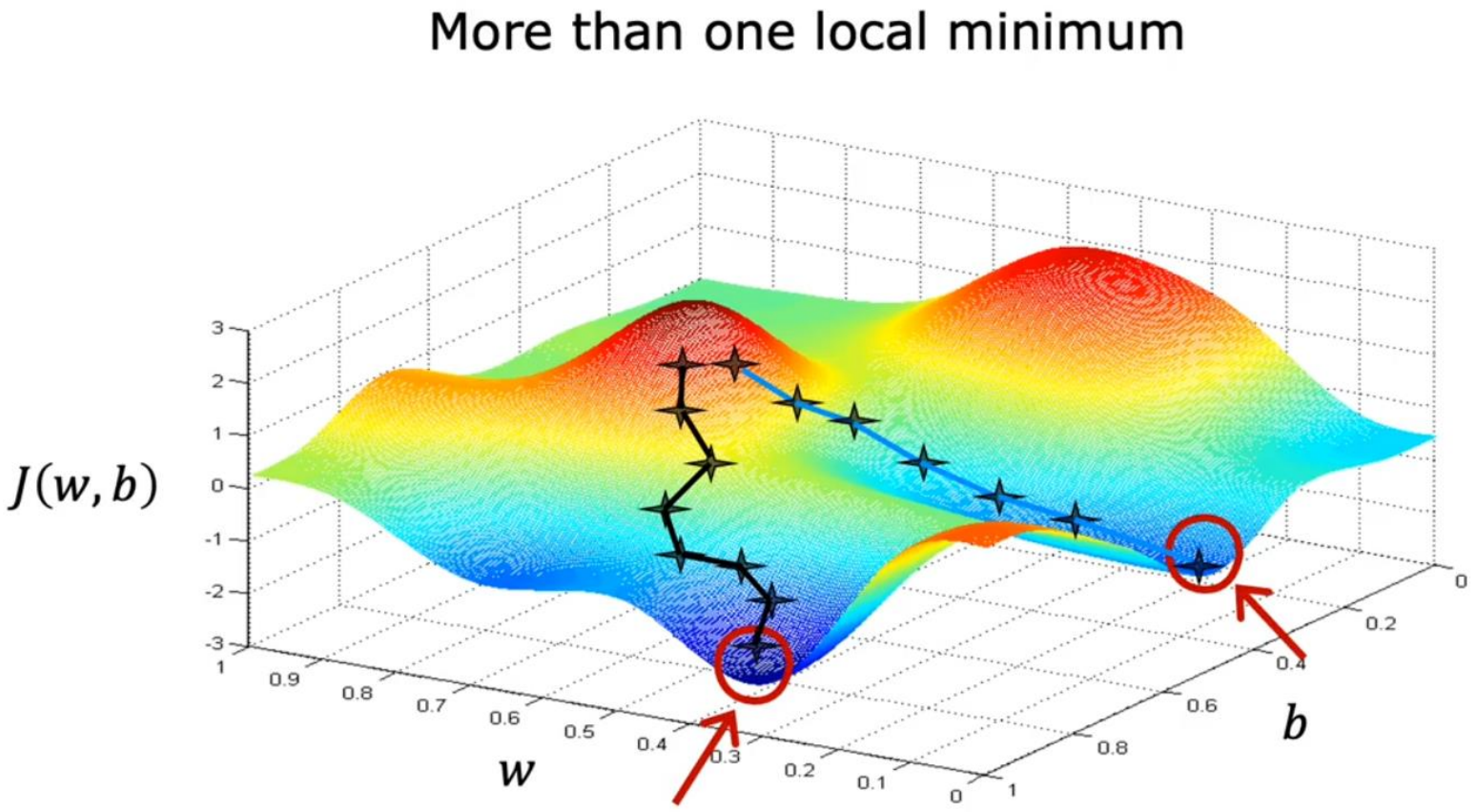
$\frac{\partial}{\partial w} J(w,b)$

$\frac{\partial}{\partial b} J(w,b)$

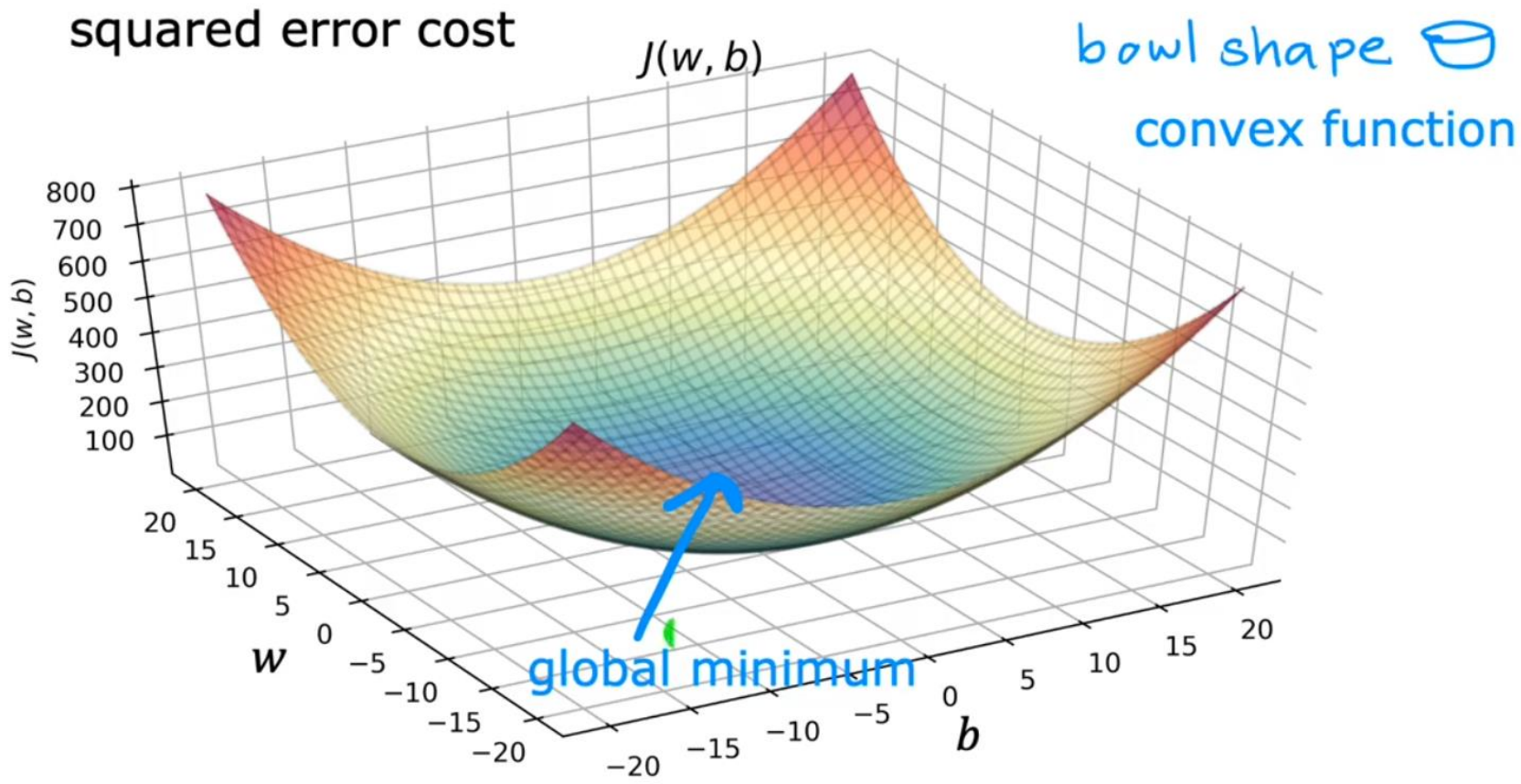
Update w and b simultaneously

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

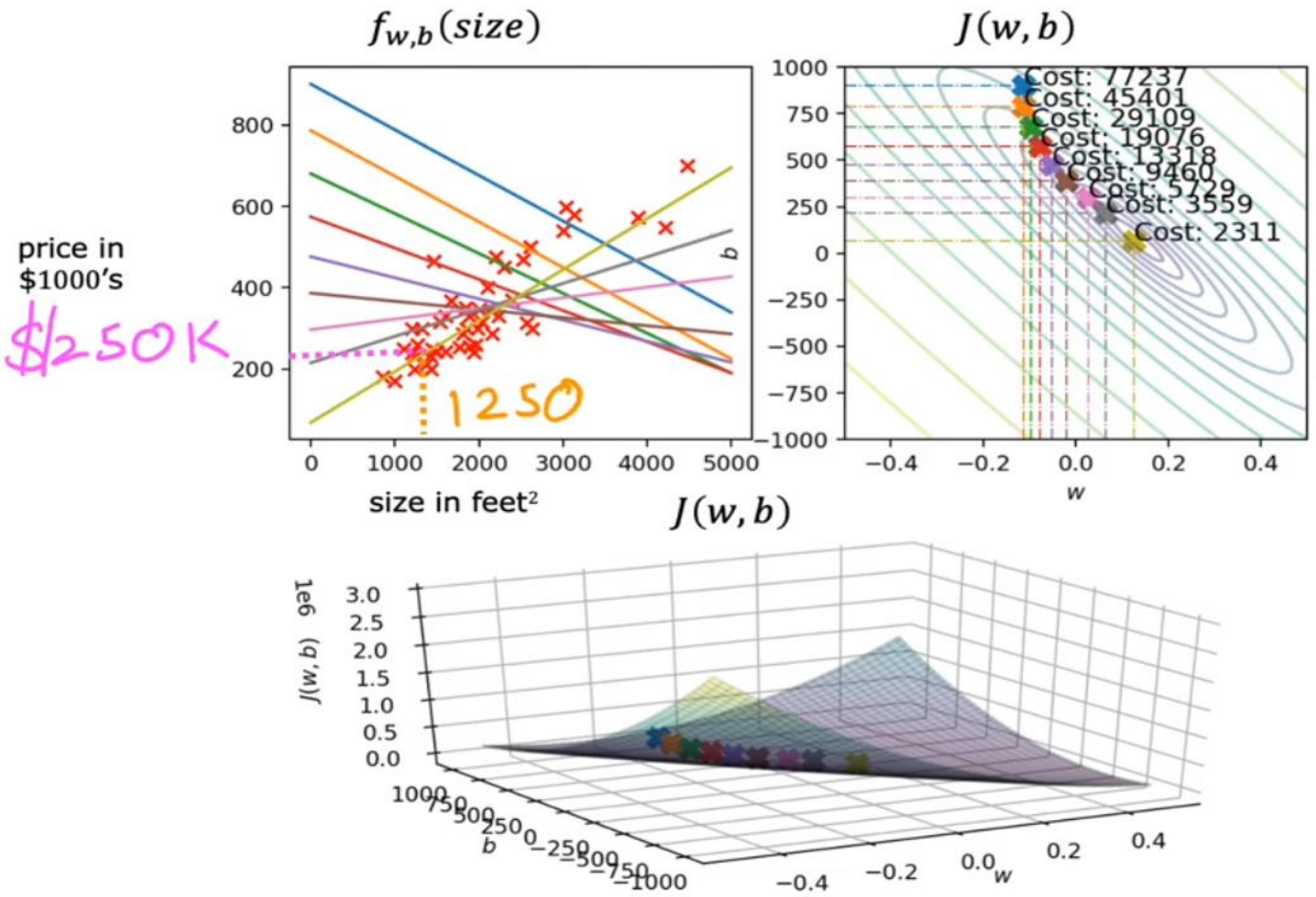
Gradient Descent for Linear Regression



Gradient Descent for Linear Regression



Running the Gradient Descent for Linear Regression



Running the Gradient Descent for Linear Regression

"Batch" gradient descent

"Batch": Each step of gradient descent uses all the training examples.

	x size in feet^2	y price in \$1000's
(1)	2104	400
(2)	1416	232
(3)	1534	315
(4)	852	178
...	...	...
(47)	3210	870

$m = 47$

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