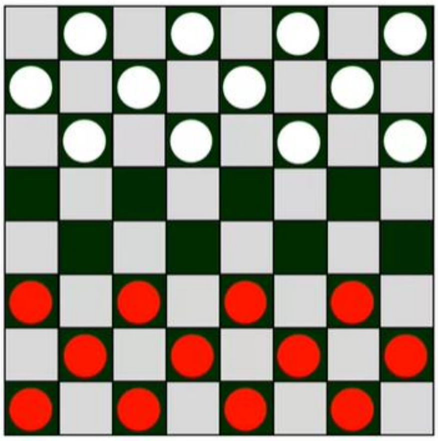
1. What is machine learning

## Machine learning

"Field of study that gives computers the ability to learn without being explicitly programmed."

Arthur Samuel (1959)



## Question

If the checkers program had been allowed to play only ten games (instead of tens of thousands) against itself, a much smaller number of games, how would this have affected its performance?

- ○ Would have made it better
- ○ Would have made it worse

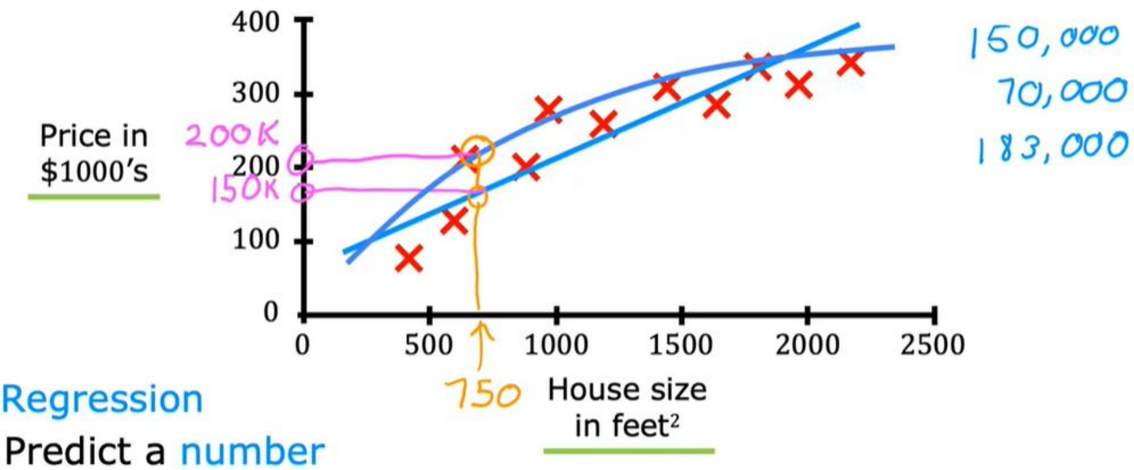
2. Supervised learning part 1

## Supervised learning


Learns from being given "right answers"

| 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 other cars | self-driving car |
| image of phone | → | defect? (0/1) | visual inspection |

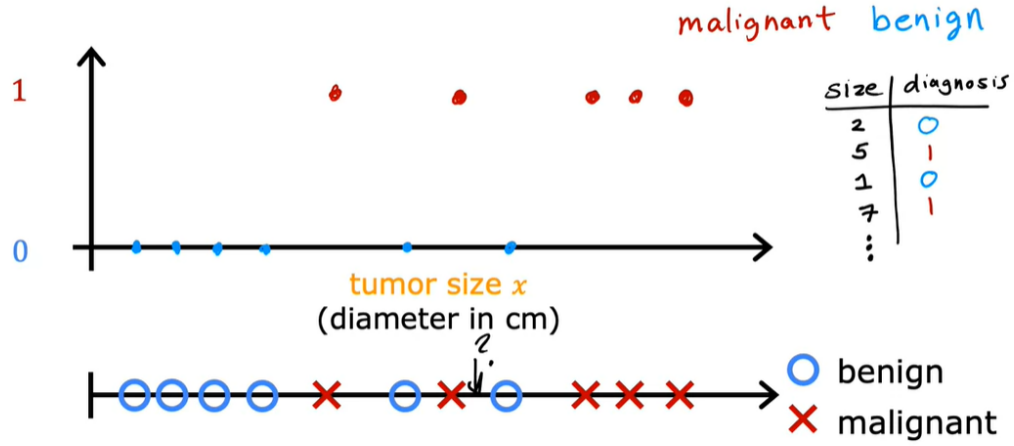
## Regression: Housing price prediction



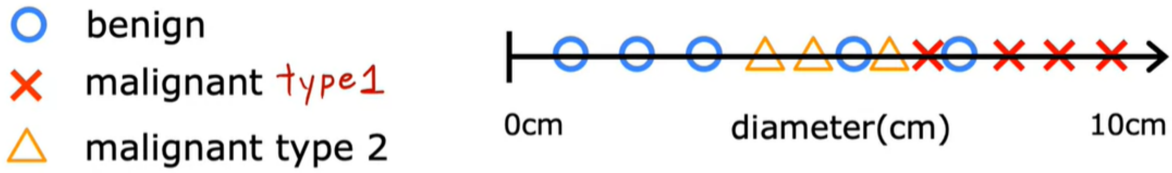

Regression
Predict a number
infinitely many possible outputs

3. Supervised learning part 2

## Classification: Breast cancer detection



## Classification: Breast cancer detection



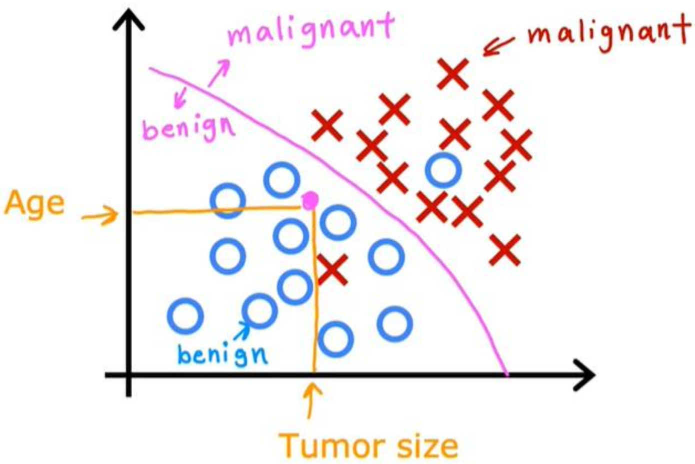

class category

Classification

predict categories cat dog benign malignant 0, 1, 2

small number of possible outputs

### Two or more inputs




Supervised learning is when we give our learning algorithm the right answer $y$ for each example to learn from. Which is an example of supervised learning?

- ○ Calculating the average age of a group of customers.

- ◉ Spam filtering.

**Correct**

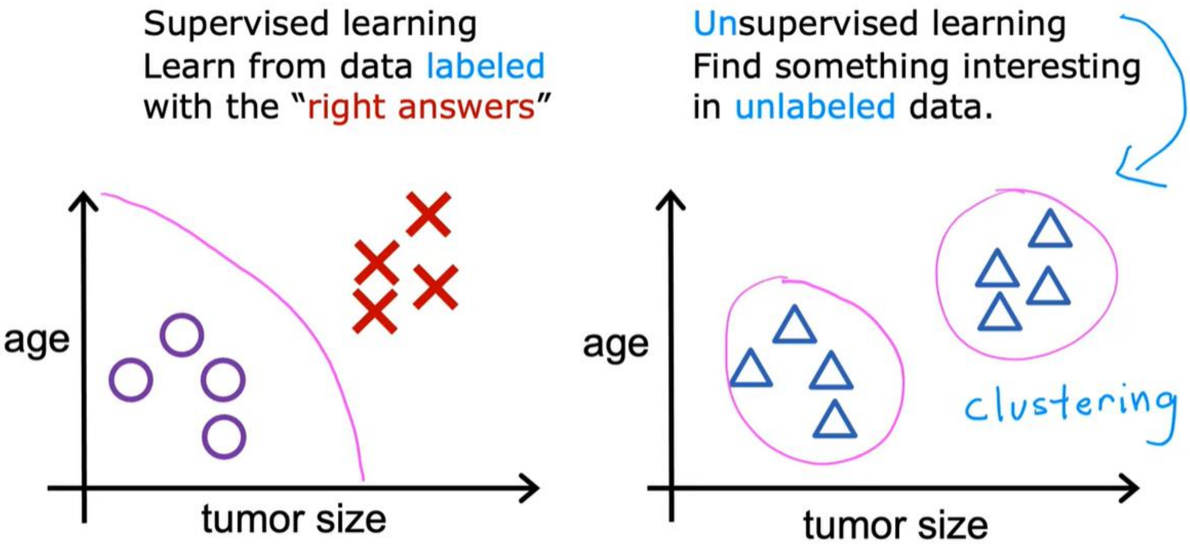
For instance, emails labeled as "spam" or "not spam" are examples used for training a supervised learning algorithm. The trained algorithm will then be able to predict with some degree of accuracy whether an unseen email is spam or not.

# Supervised learning

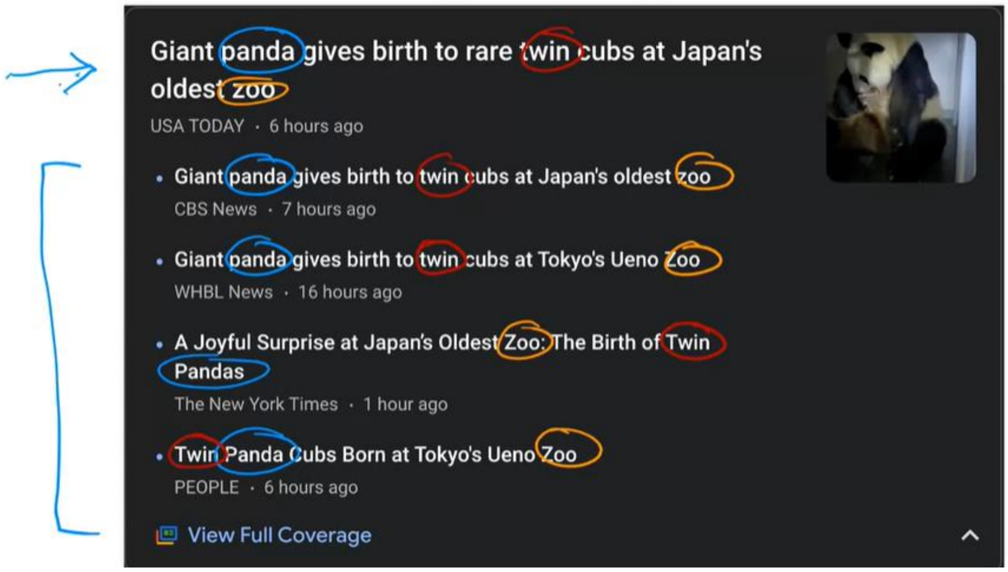
Learns from being given "right answers"

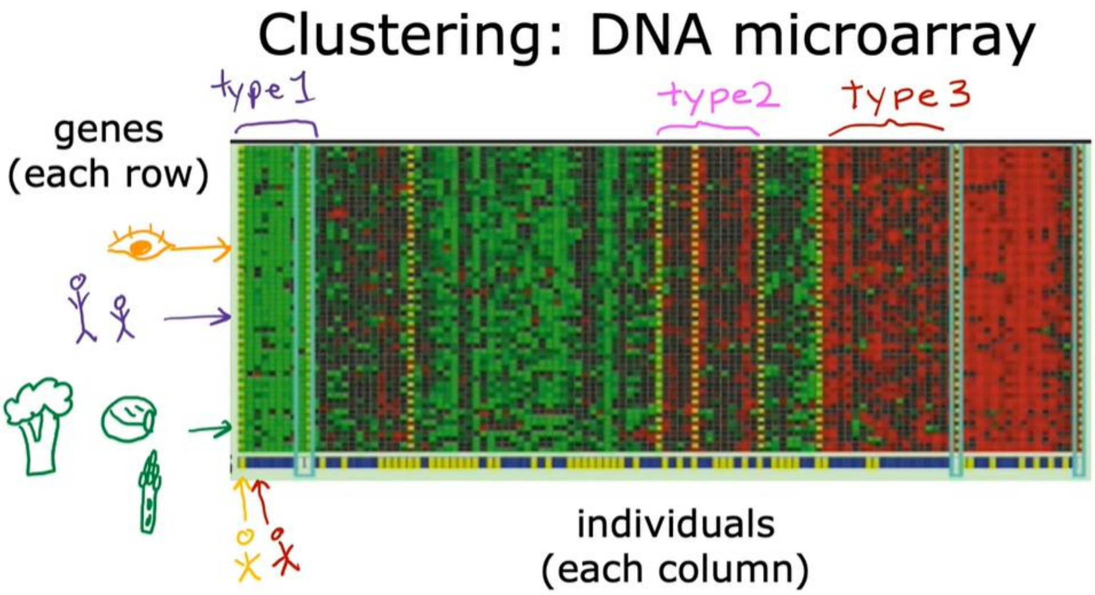
| Regression | Classification |
| --- | --- |
| Predict a number | predict categories |
| infinitely many possible outputs | small number of possible outputs |

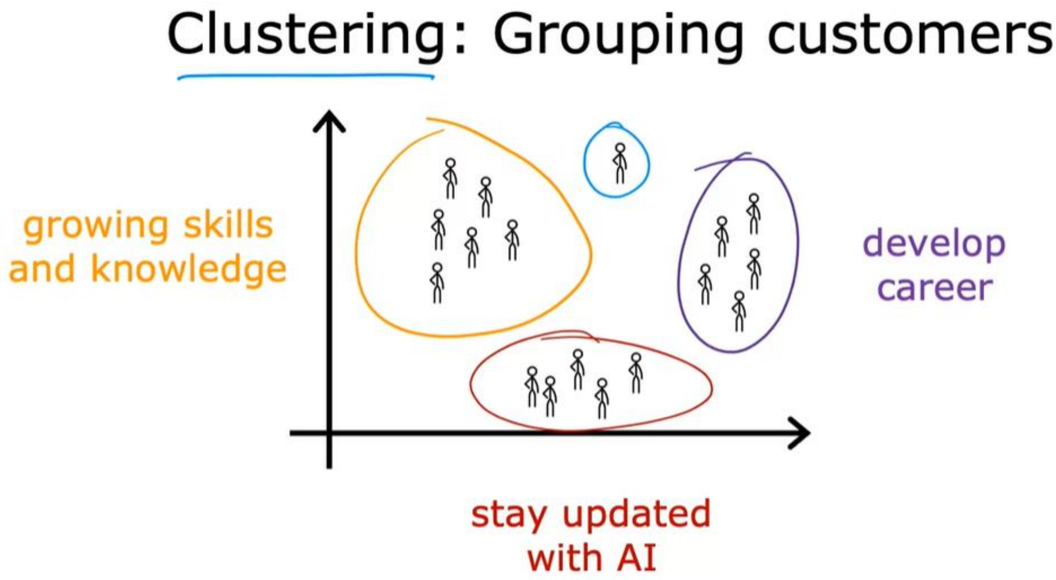
4. Unsupervised learning part 1




## Clustering: Google news






5. Unsupervised learning part 2

## Unsupervised learning

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

<u>Clustering</u>
Group similar data points together.

<u>Dimensionality reduction</u>
Compress data using fewer numbers.

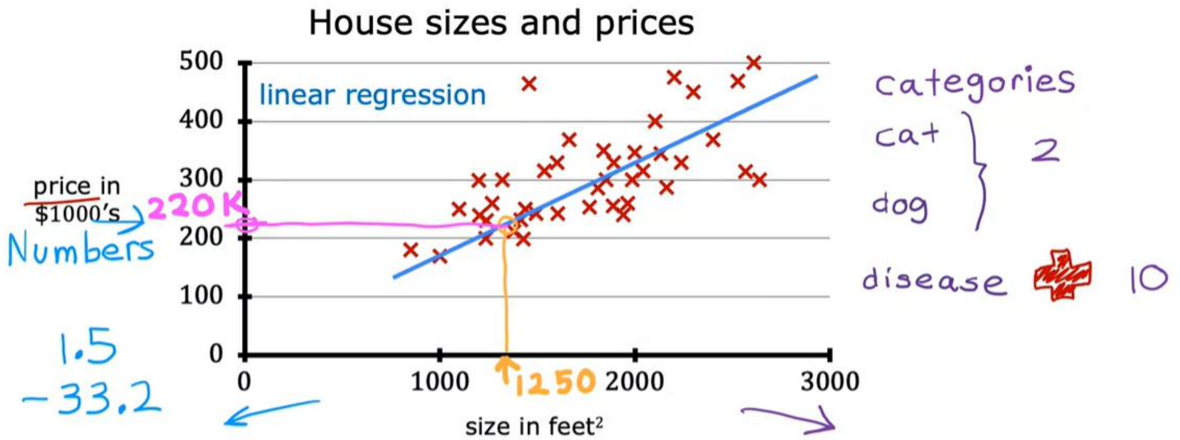
<u>Anomaly detection</u>
Find unusual data points.

## 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

6. Linear regression model part 1




Regression model
Predicts numbers
Infinitely many possible outputs

Supervised learning model
Data has "right answers"

Classification model
Predicts categories
Small number of possible outputs


Data table

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

## Terminology

**Training set:** Data used to train the model

| | $x$ size in feet² | $y$ price in \$1000's |
|---|---|---|
| (1) | 2104 | 400 |
| (2) | 1416 | 232 |
| (3) | 1534 | 315 |
| (4) | 852 | 178 |
| ... | ... | ... |
| (47) | 3210 | 870 |

$m = 47$

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

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

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

**Notation:**

$x$ = "input" variable
feature

$y$ = "output" variable
"target" variable

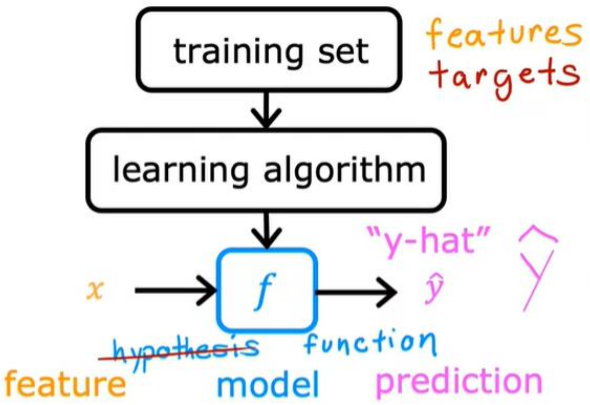
$m$ = number of training examples

$(x, y)$ = single training example

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

$(x^{(i)}, y^{(i)})$ = $i^{th}$ training example
index (1st, 2nd, 3rd ...)

7. Linear regression model part 2



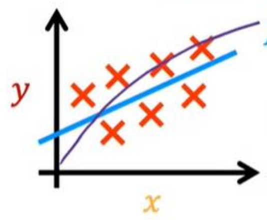

(estimated $y$)
target

size → $\boxed{f}$ → price (estimated)

**How to represent $f$?**

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

$f(x)$




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

$f(x) = wx + b$

linear

single feature x

Linear regression with one variable.
size

Univariate linear regression.
one variable

8. Cost function formula

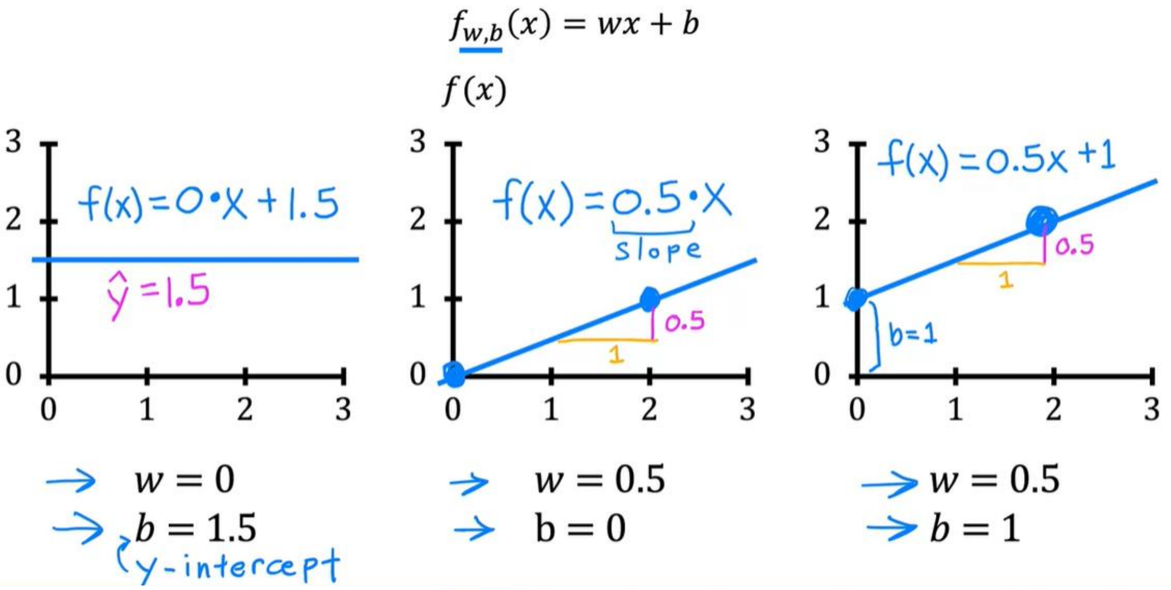
## Training set

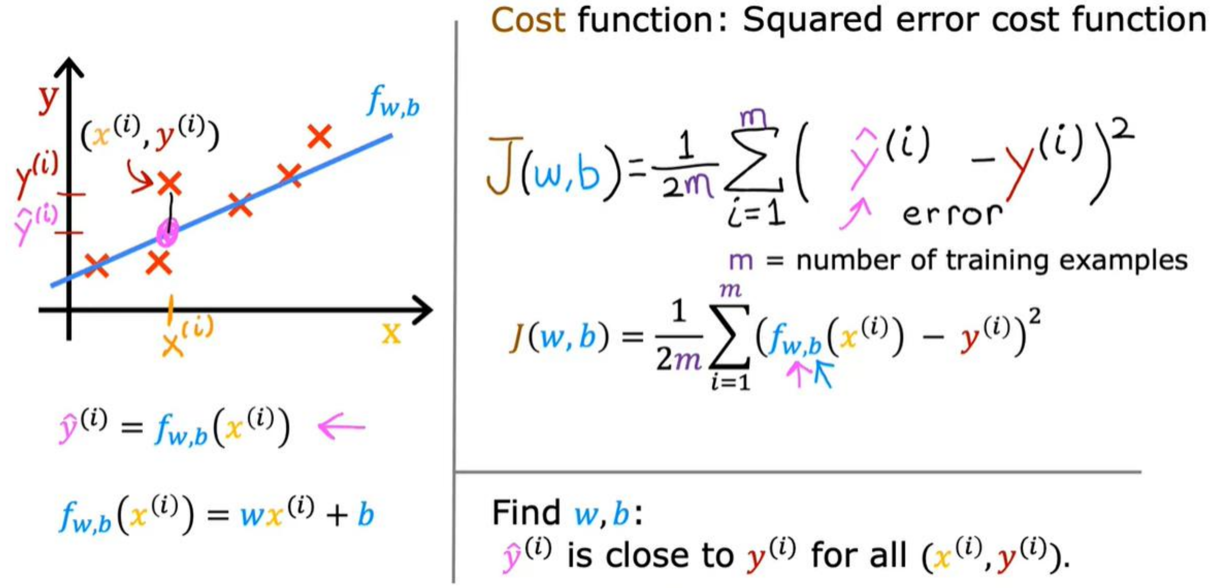
| features size in feet² ($x$) | targets price \$1000's ($y$) |
|---|---|
| 2104 | 460 |
| 1416 | 232 |
| 1534 | 315 |
| 852 | 178 |
| ... | ... |

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

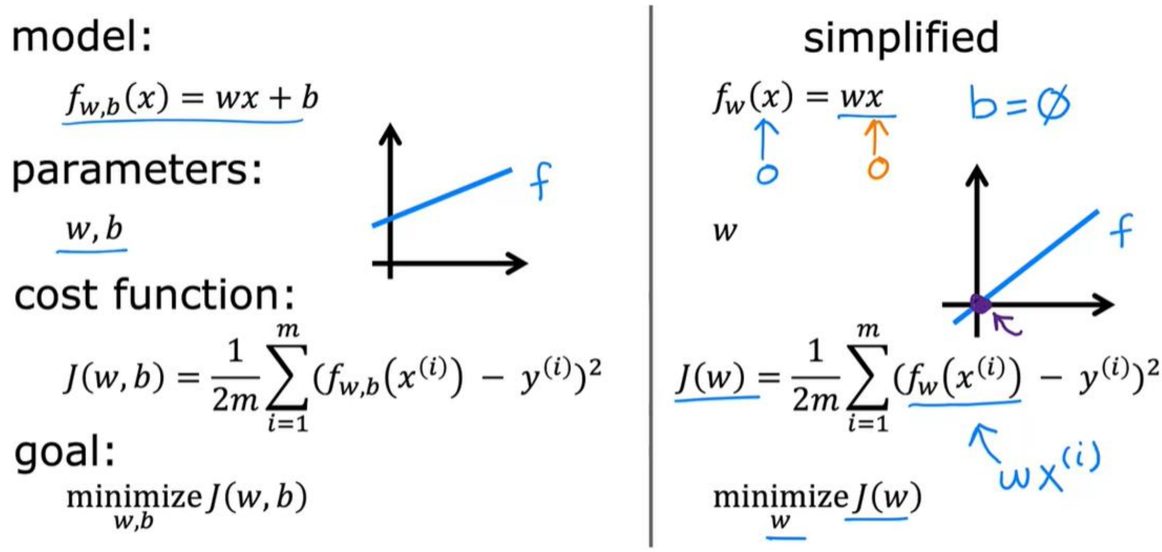
$w, b$: parameters
coefficients
weights

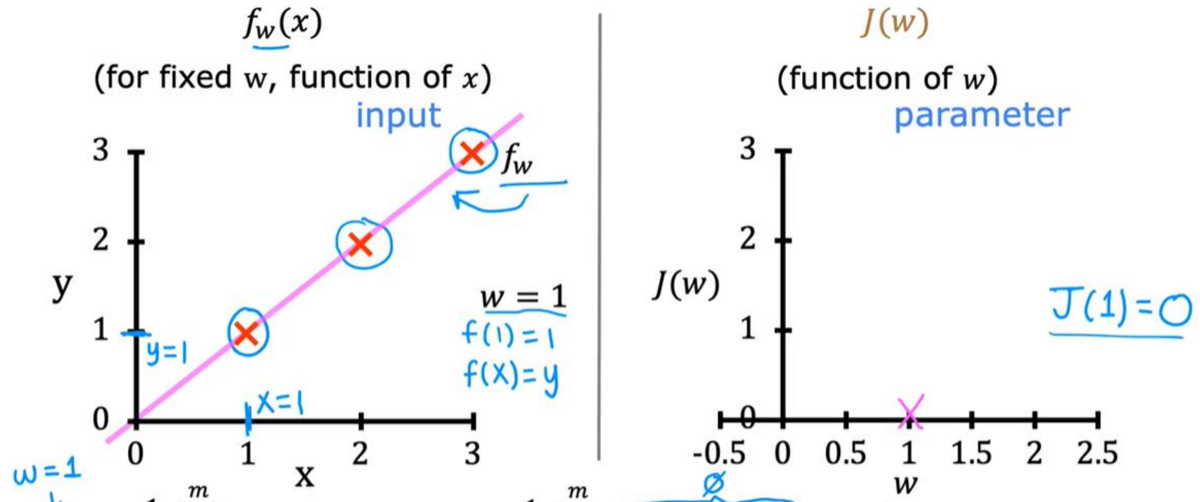
What do $w, b$ do?



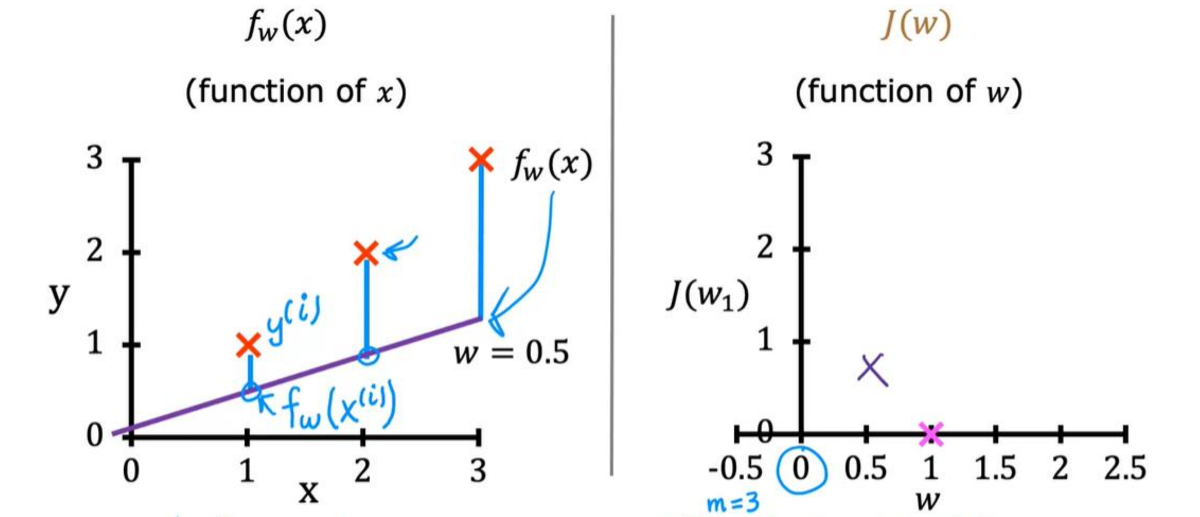




9. Cost function: intuition

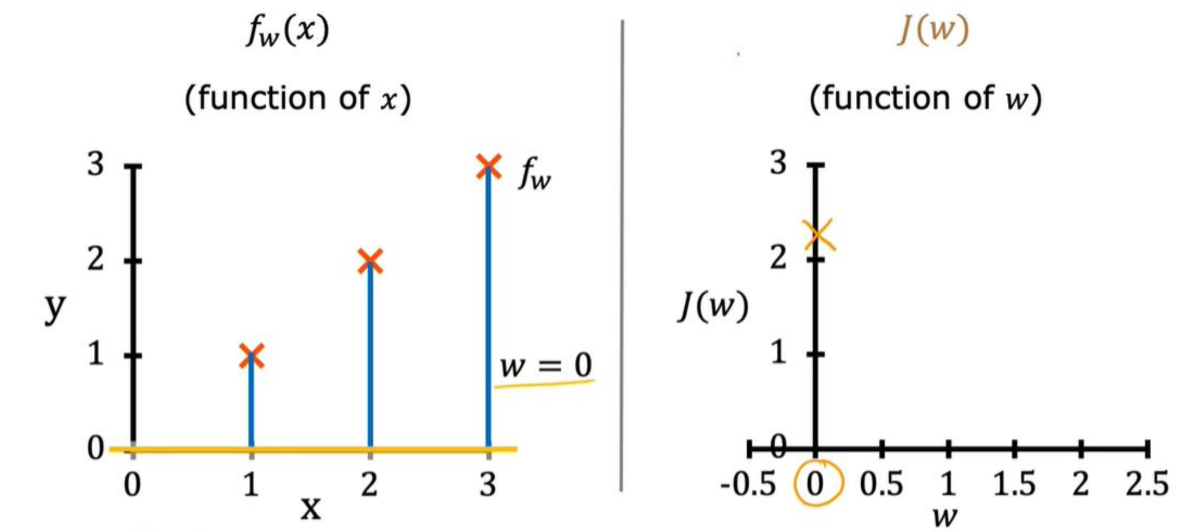




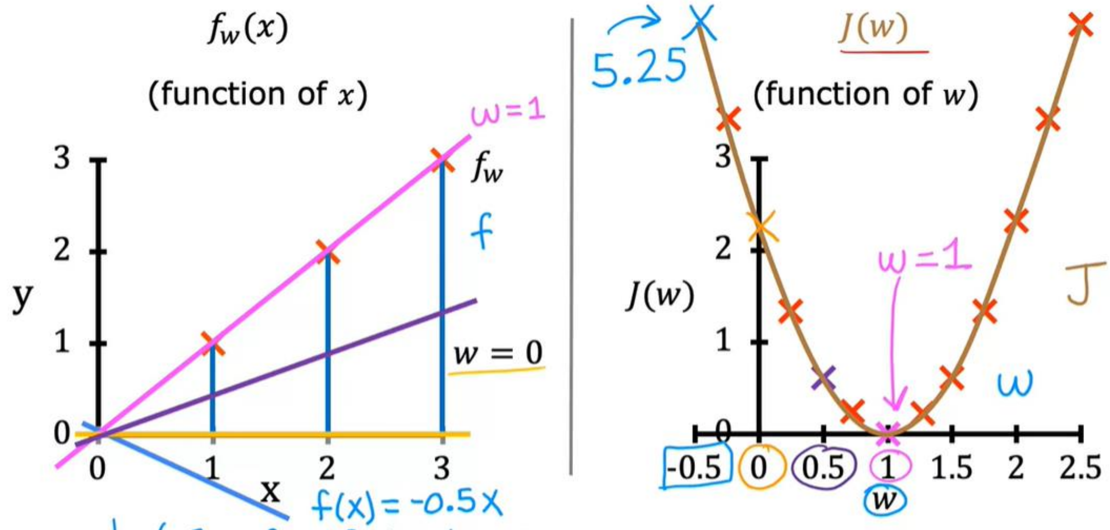

$$\underset{\substack{w=1\\\downarrow}}{J(w)} = \frac{1}{2m}\sum_{i=1}^{m}(\underbrace{f_w(x^{(i)})}_{wx^{(i)}} - y^{(i)})^2 = \frac{1}{2m}\sum_{i=1}^{m}\overbrace{(wx^{(i)} - y^{(i)})^2}^{\varnothing} = \frac{1}{2m}(0^2+0^2+0^2) = 0$$




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




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




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

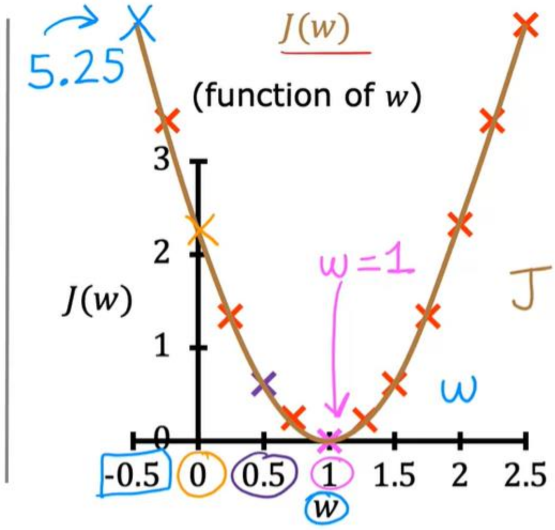
how to choose w?

goal of linear regression:

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

general case:

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




choose w to minimize J(w)

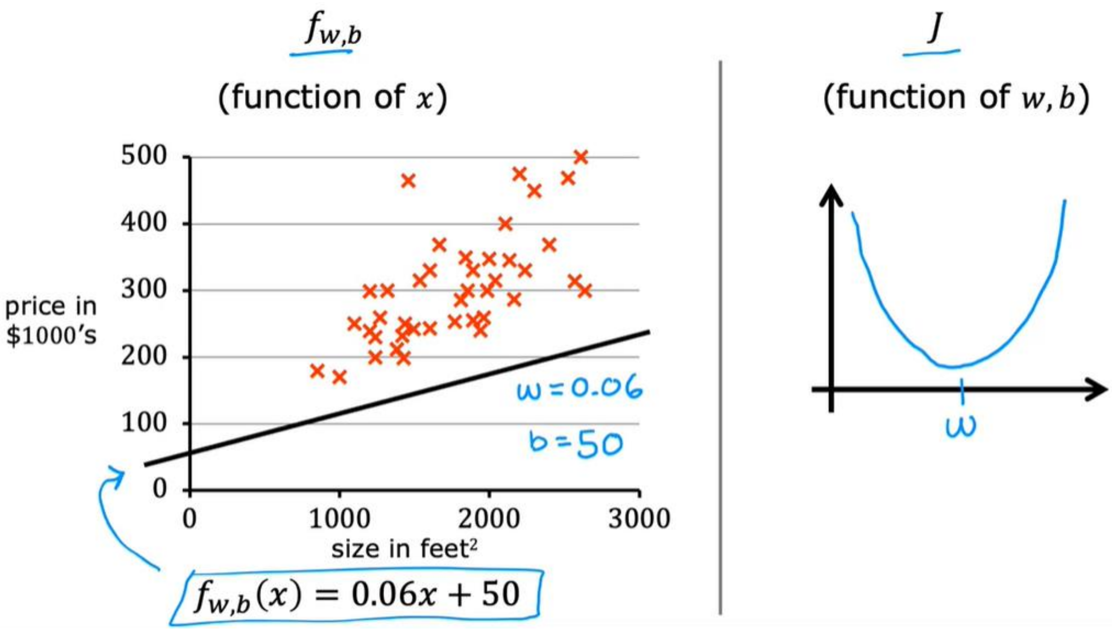
10. Visualizing the cost function

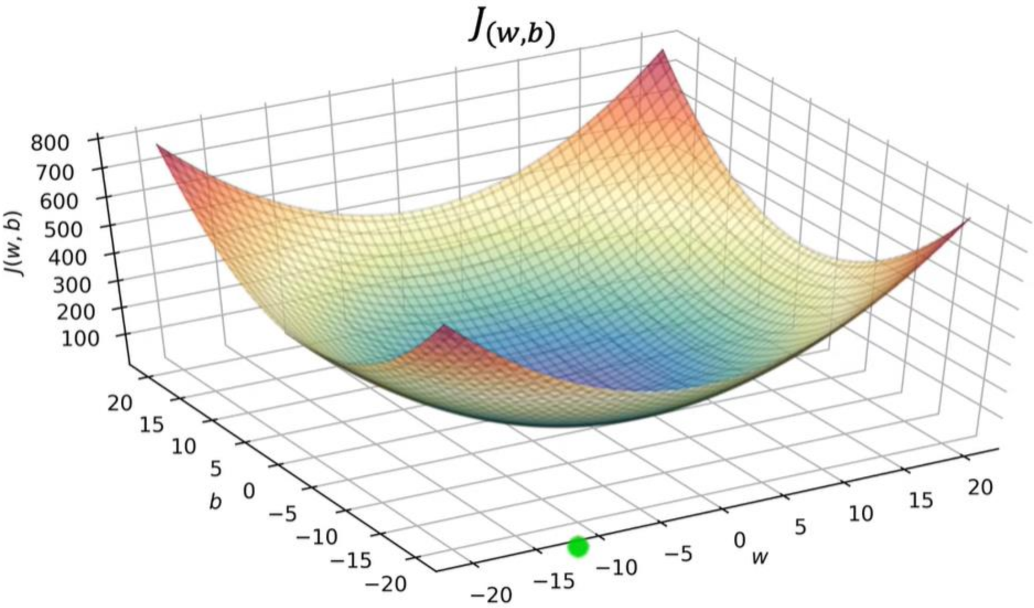
Model $f_{w,b}(x) = wx + b$

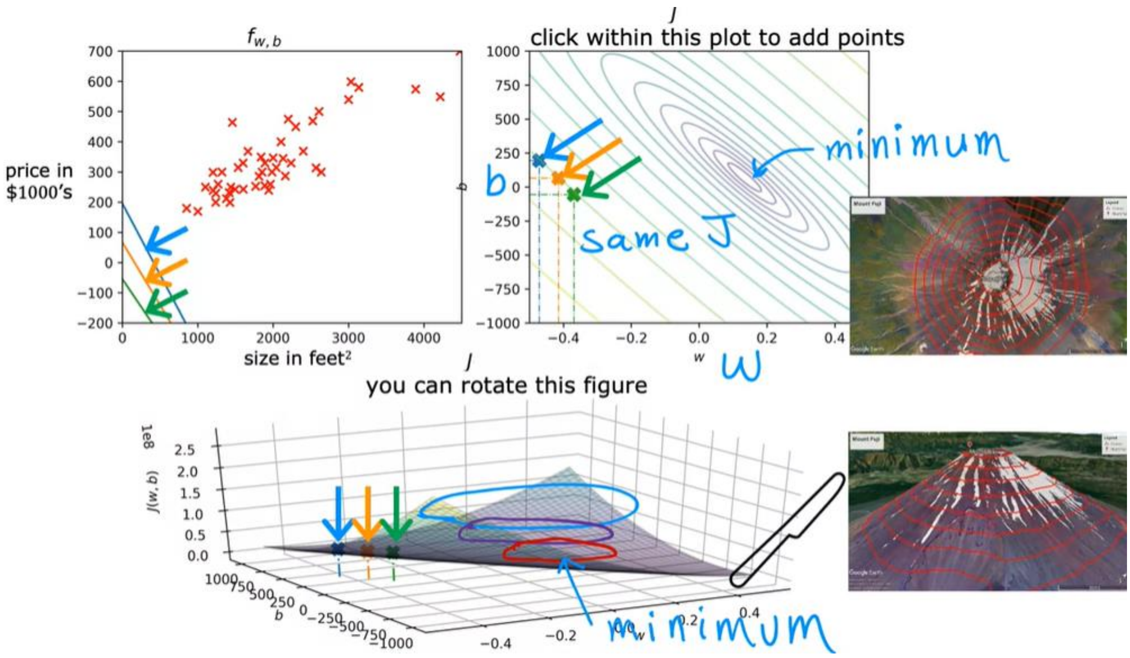
Parameters $w, b$ ~~before: b=0~~

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

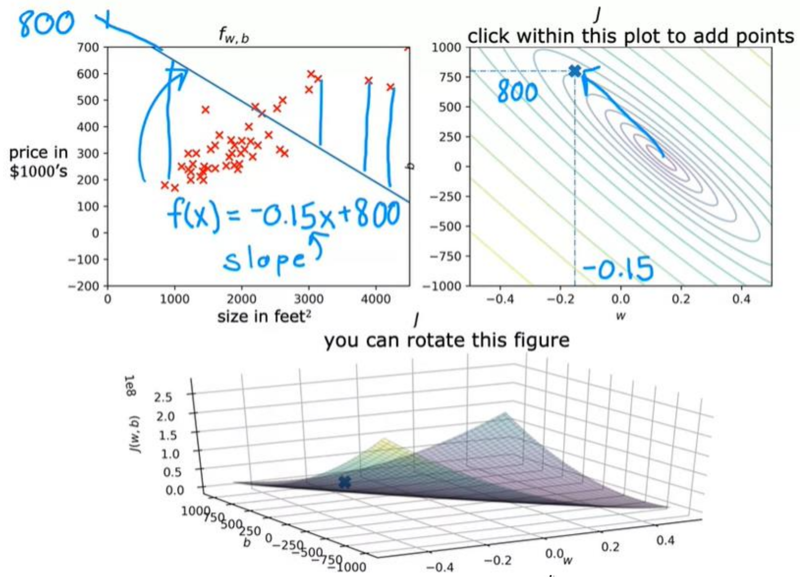
Objective $$\underset{w,b}{\text{minimize}}\, J(w, b)$$

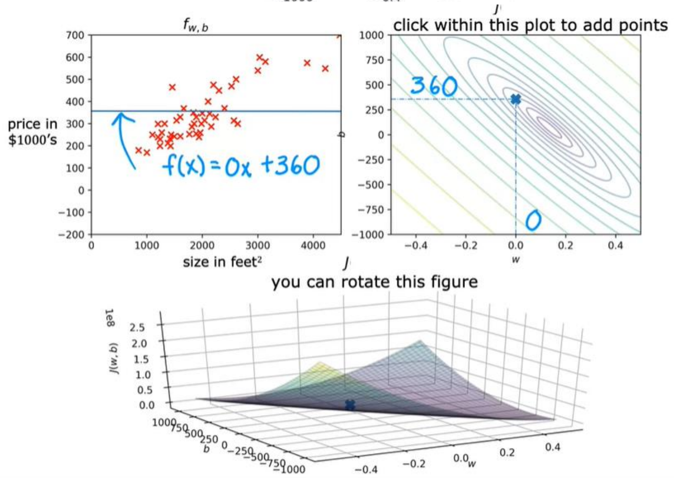

$f_{w,b}$
(function of $x$)
price in $1000's
size in feet²
500
400
300
200
100
0
0
1000
2000
3000
w=0.06
b=50
$f_{w,b}(x) = 0.06x + 50$
$J$
(function of $w, b$)
w

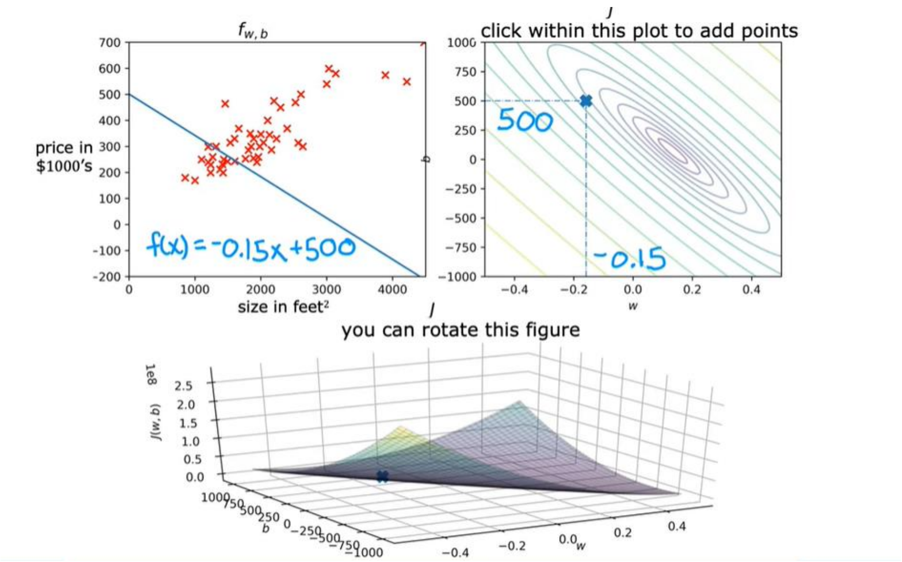

$J_{(w,b)}$
$J(w, b)$
800
700
600
500
400
300
200
100
$b$
20
15
10
5
0
−5
−10
−15
−20
$w$
−20
−15
−10
−5
0
5
10
15
20

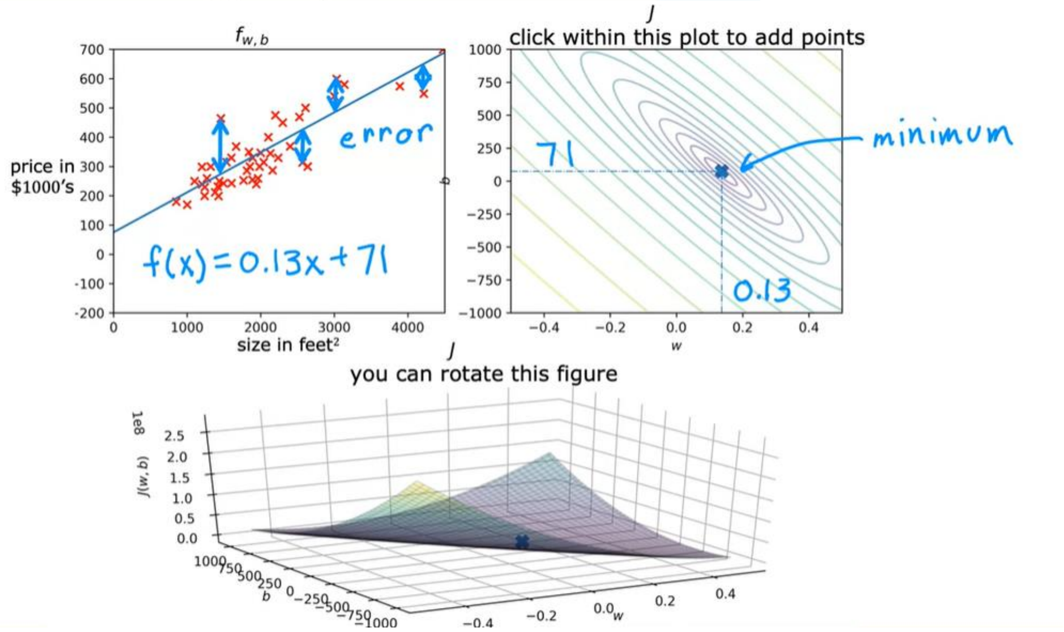



11. Visualization examples






## 12. Practice quiz

1.

For linear regression, the model is $f_{w,b}(x) = wx + b$.

Which of the following are the inputs, or features, that are fed into the model and with which the model is expected to make a prediction?

- ○ $m$
- ○ $w$ and $b$.
- ◉ $x$
- ○ $(x, y)$

**Correct**
The $x$, the input features, are fed into the model to generate a prediction $f_{w,b}(x)$

2. For linear regression, if you find parameters $w$ and $b$ so that $J(w, b)$ is very close to zero, what can you conclude?

- ○ This is never possible -- there must be a bug in the code.
- ○ The selected values of the parameters $w$ and $b$ cause the algorithm to fit the training set really poorly.
- ◉ The selected values of the parameters $w$ and $b$ cause the algorithm to fit the training set really well.

**Correct**
When the cost is small, this means that the model fits the training set well.

13. Gradient descent

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

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

Outline:

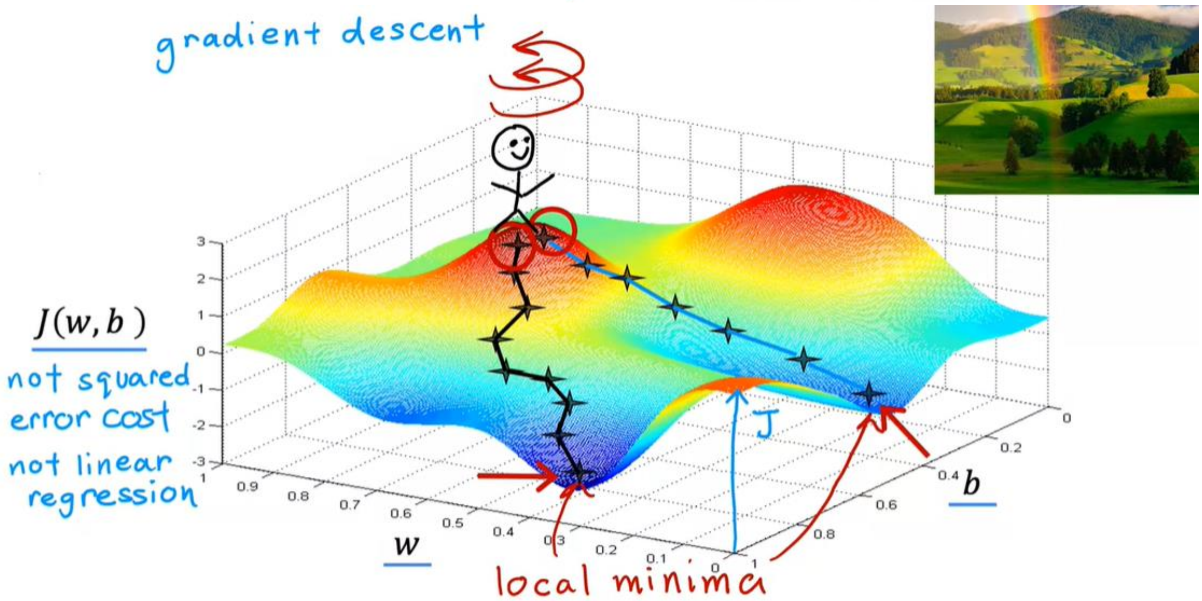
Start with some $w, b$ (set $w$=0, $b$=0)

Keep changing $w, b$ to reduce $J(w,b)$

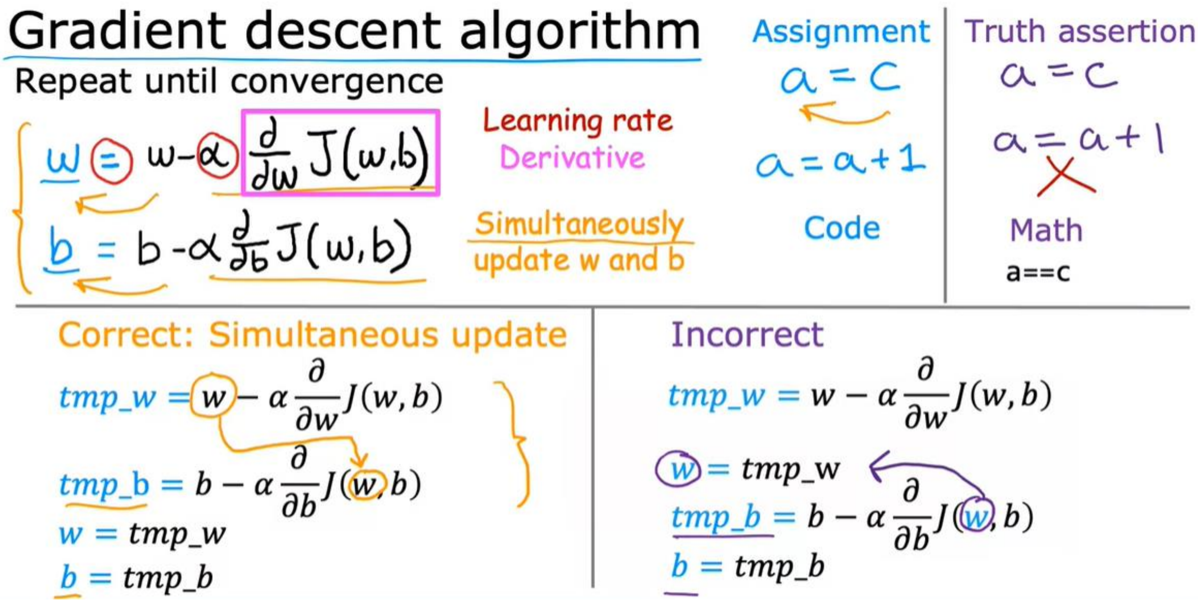
Until we settle at or near a minimum

may have >1 minimum

J not always



14. Implementing gradient descent



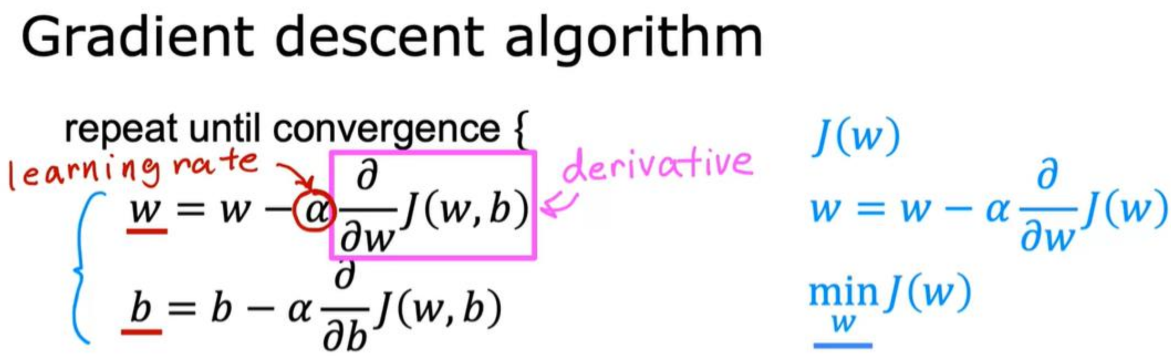

Gradient descent is an algorithm for finding values of parameters w and b that minimize the cost function J. What does this update statement do? (Assume $\alpha$ is small.)

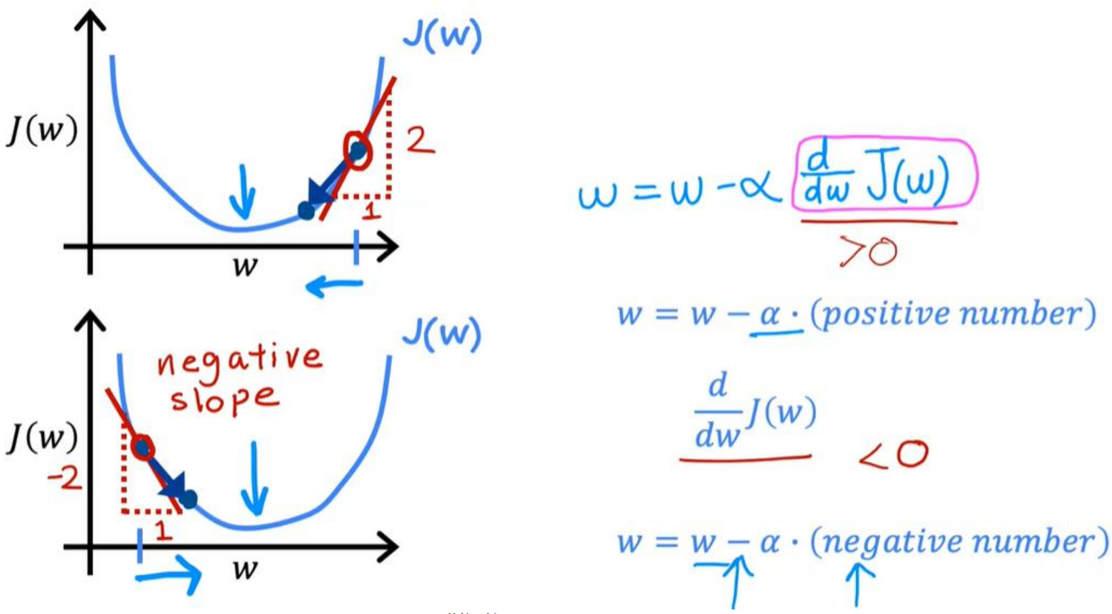
$w = w - \alpha \frac{\partial J(w,b)}{\partial w}$

- ○ Checks whether $w$ is equal to $w - \alpha \frac{\partial J(w,b)}{\partial w}$
- ◉ Updates parameter $w$ by a small amount

✓ **Correct**
This updates the parameter by a small amount, in order to reduce the cost $J$.

15. Gradient descent: intuition



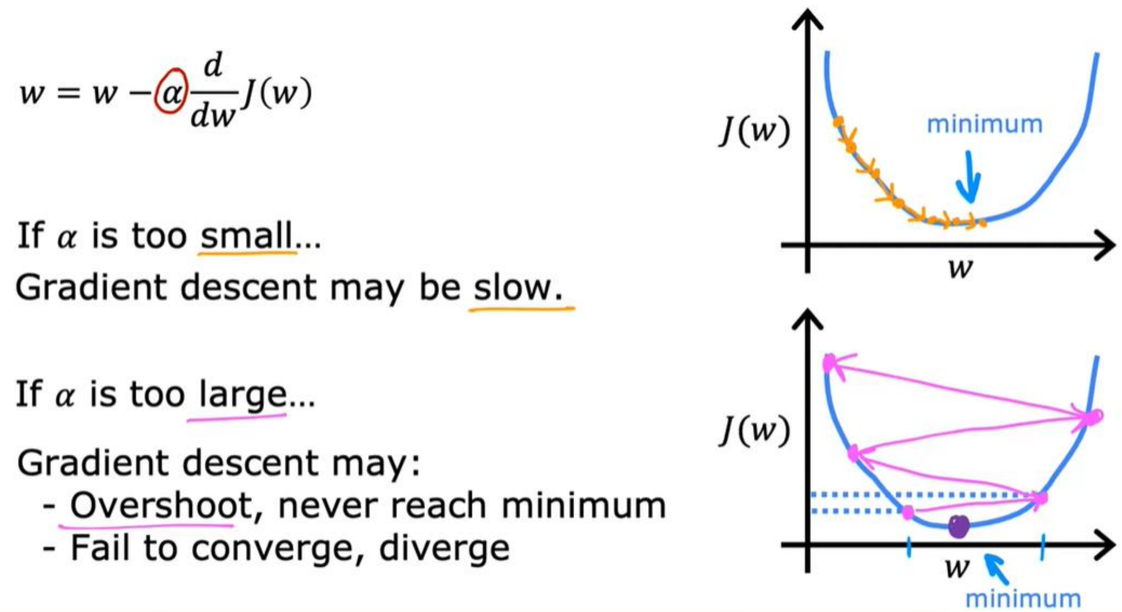


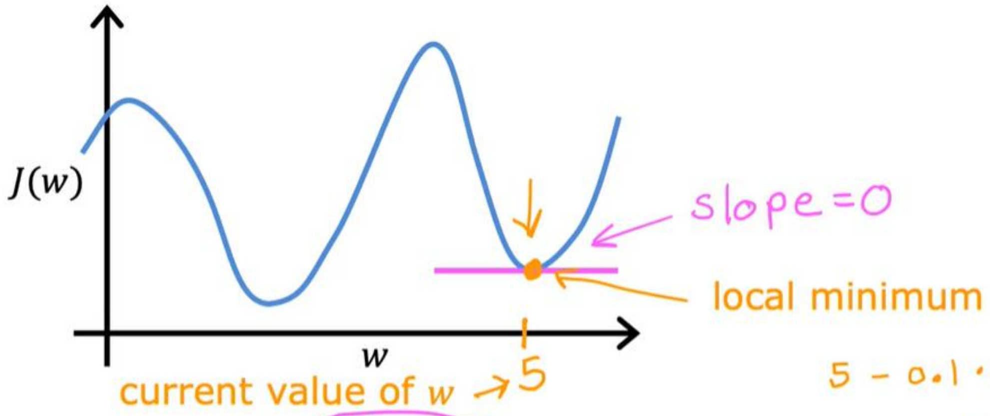

Assume the learning rate $\alpha$ is a small positive number. When $\frac{\partial J(w,b)}{\partial w}$ is a positive number (greater than zero) -- as in the example in the upper part of the slide shown above -- what happens to $w$ after one update step?

- ○ $w$ increases
- ○ It is not possible to tell if $w$ will increase or decrease.
- ◉ $w$ decreases.
- ○ $w$ stays the same

**Correct**
The learning rate $\alpha$ is always a positive number, so if you take W minus a positive number, you end up with a new value for W that is smaller

16. Learning rate






$$w = w - \underset{0.1}{\alpha}\underbrace{\boxed{\frac{d}{dw}J(w)}}_{=0}$$

$5 - 0.1 \cdot 0$

$w = w - \alpha \cdot 0$

$\overset{5}{w} = \overset{5}{w}$

Can reach local minimum with fixed learning rate $\alpha$

$$w = w - \underbrace{\overbrace{\overbrace{\boxed{\alpha \frac{d}{dw}J(w)}}^{\text{large}}}^{\text{not as large}}}_{}$$

smaller

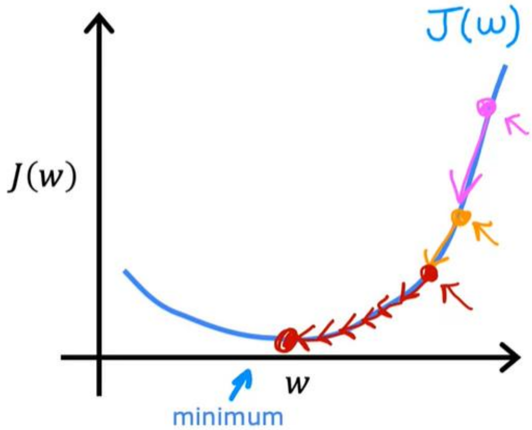
not as large

large

Near a local minimum,

- Derivative becomes smaller
- Update steps become smaller

Can reach minimum without decreasing learning rate $\alpha$




17. 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 {

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

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

}

(Optional)

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

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

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

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

no $x^{(i)}$

# Gradient descent algorithm

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

repeat until convergence {

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

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

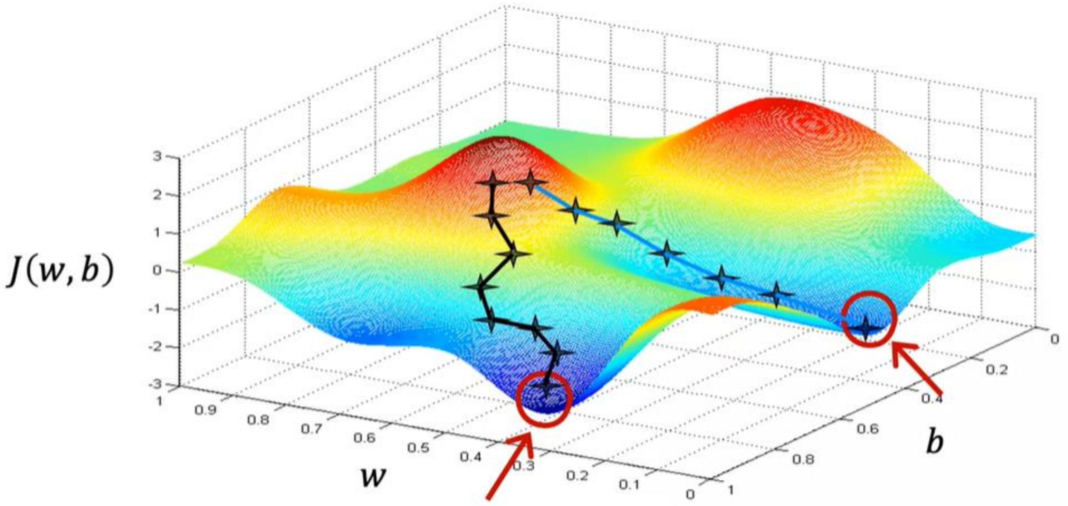
}

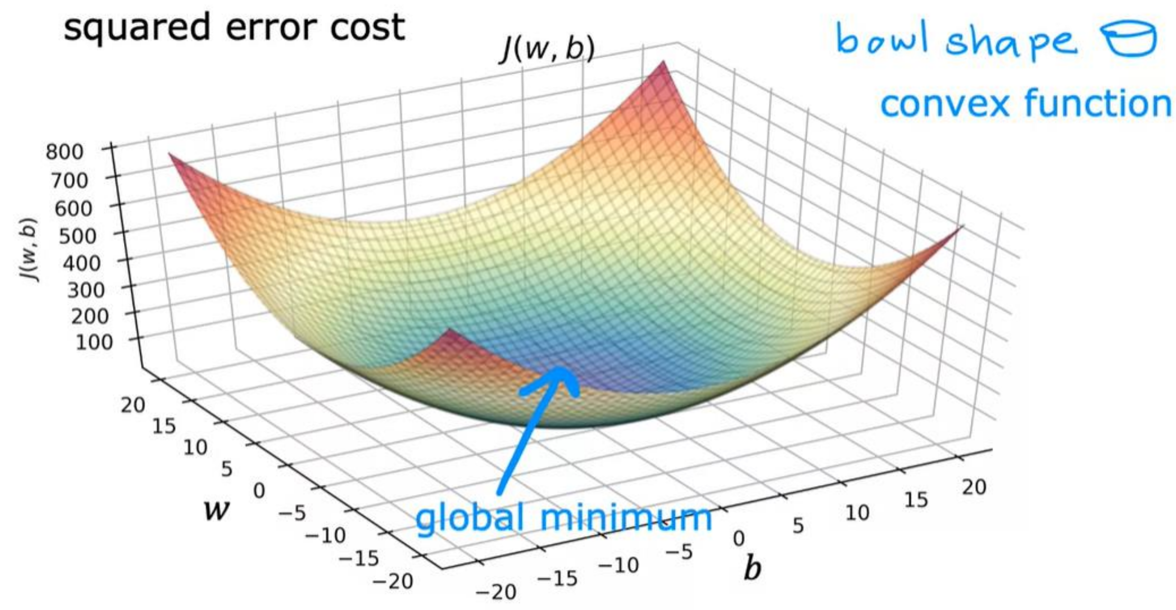
Update $w$ and $b$ simultaneously

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

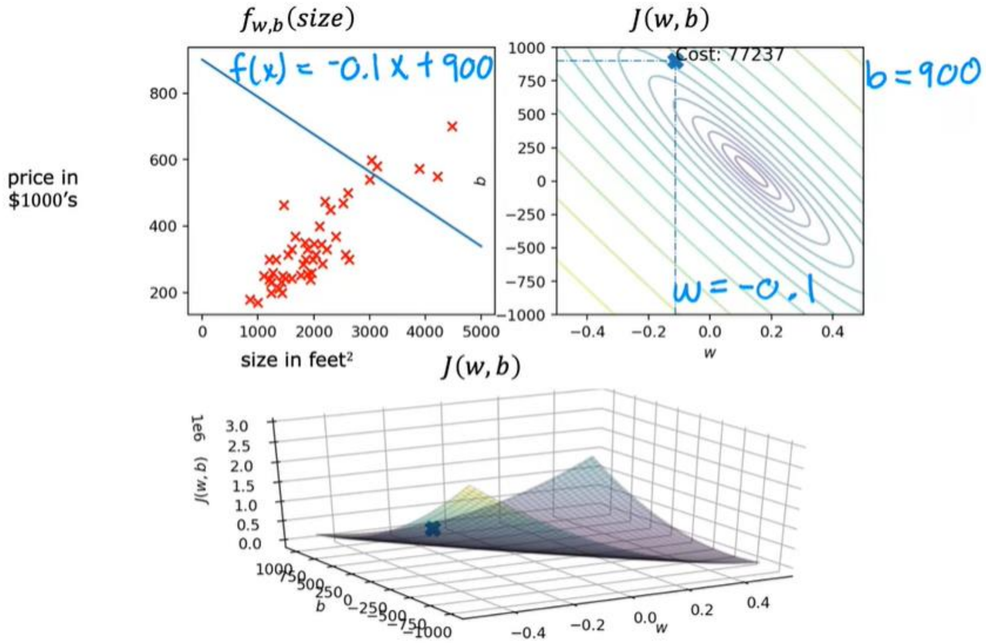
$f_{w,b}(x^{(i)}) = wx^{(i)}+b$

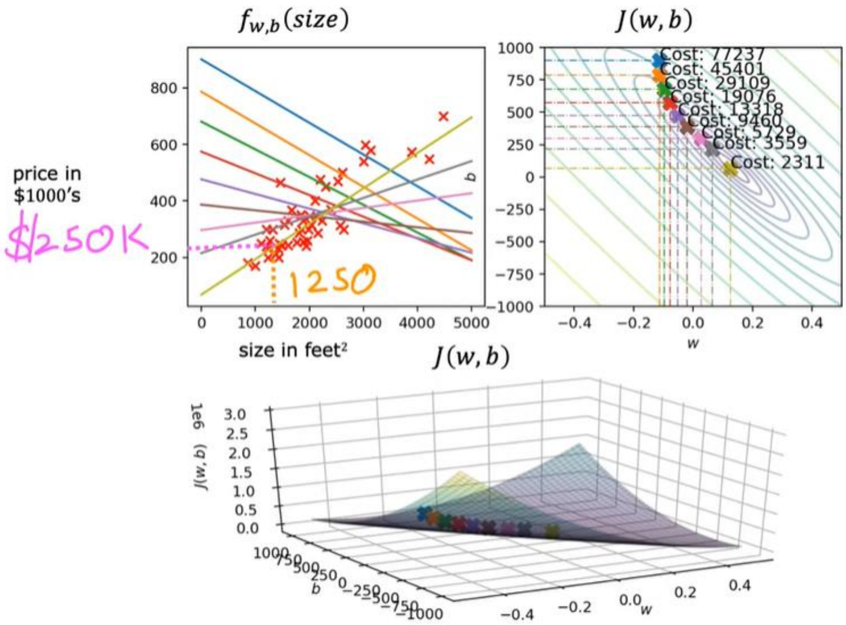
More than one local minimum






18. Running gradient descent






# "Batch" gradient descent

other gradient descent: subsets

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

| | $x$ size in feet² | $y$ price in $1000's |
|---|---|---|
| (1) | 2104 | 400 |
| (2) | 1416 | 232 |
| (3) | 1534 | 315 |
| (4) | 852 | 178 |
| ... | ... | ... |
| (47) | 3210 | 870 |

$m=47$

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

19. Practice quiz

1.

Gradient descent is an algorithm for finding values of parameters w and b that minimize the cost function J.

repeat until convergence {

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

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

When $\frac{\partial J(w,b)}{\partial w}$ is a negative number (less than zero), what happens to $w$ after one update step?

- (●) $w$ increases.
- ( ) $w$ stays the same
- ( ) $w$ decreases
- ( ) It is not possible to tell if $w$ will increase or decrease.

**Correct**
The learning rate is always a positive number, so if you take W minus a negative number, you end up with a new value for W that is larger (more positive).

2.

For linear regression, what is the update step for parameter b?

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

**Correct**
The update step is $b = b - \alpha \frac{\partial J(w,b)}{\partial w}$ where $\frac{\partial J(w,b)}{\partial b}$ can be computed with this expression: $\sum_{i=1}^{m} (f_{w,b}(x^{(i)}) - y^{(i)})$