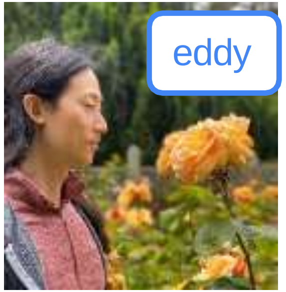
Copyright Notice

Stanford ONLINE

DeepLearning.AI

Machine Learning

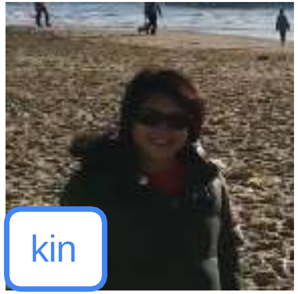
Welcome!


eddy

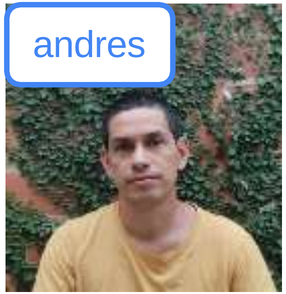

aarti


geoff


Ivy

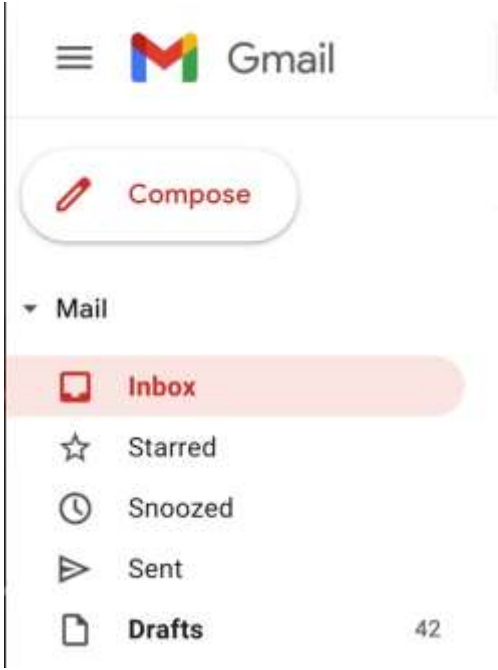

kin


robert


andres


daniel

Re: Urgent Information :) External Spam ×



Congratulations! You've won a million dollars!

Stanford ONLINE

DeepLearning.AI

Machine Learning

Applications of Machine Learning

Stanford ONLINE

DeepLearning.AI

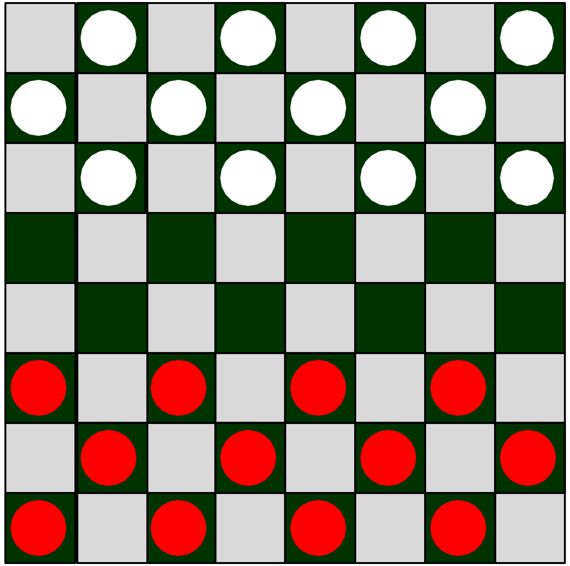
Machine Learning Overview

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

Machine learning algorithms

rapid advancements

used most in real-world applications

- Supervised learning

course 1, 2

- Unsupervised learning
- Recommender systems
- Reinforcement learning

course 3

Practical advice for applying learning algorithms

Stanford
ONLINE

DeepLearning.AI

Machine Learning Overview

Supervised Learning Part 1

Supervised learning

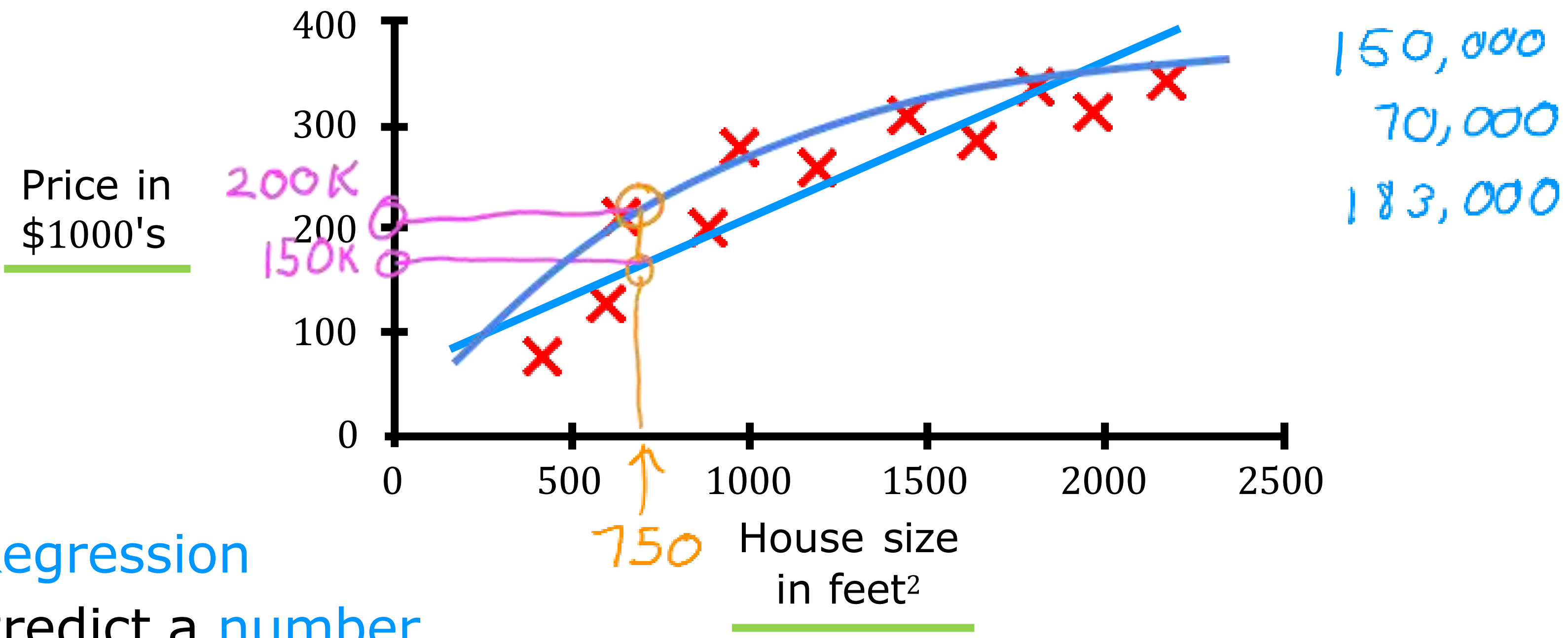
$X \longrightarrow y$

input $\longrightarrow$ output label

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

Stanford ONLINE

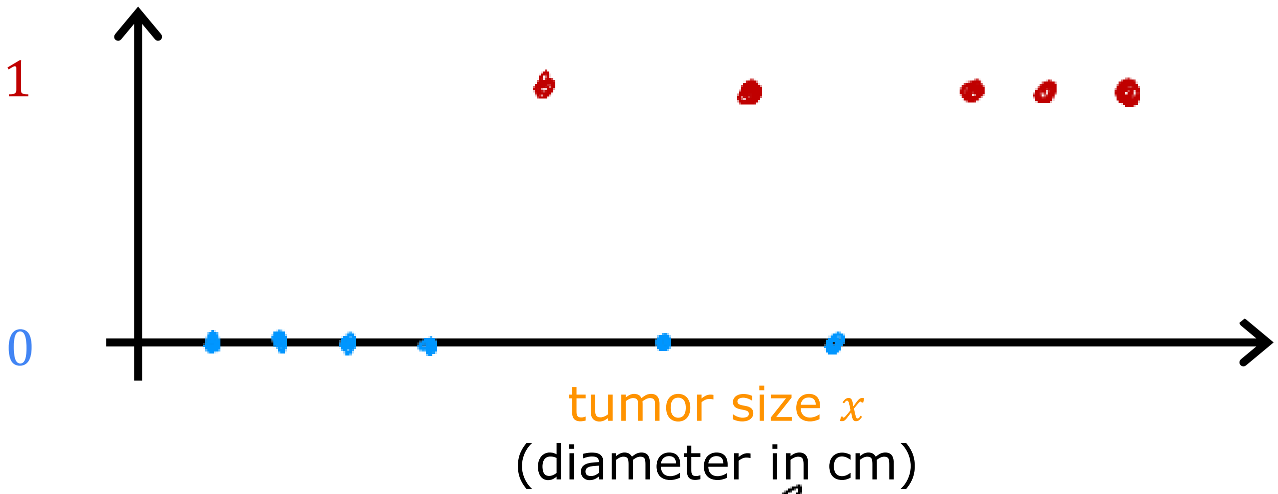
DeepLearning.AI

Machine Learning Overview

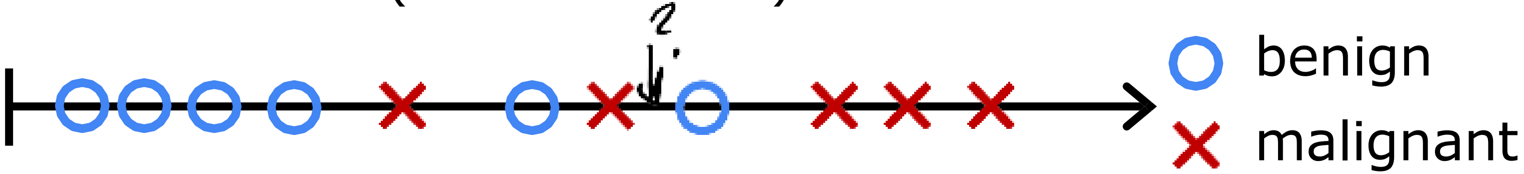
Supervised Learning Part 2

Classification: Breast cancer detection

malignant benign

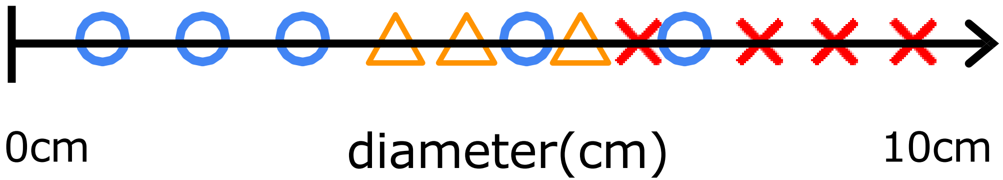


size	diagnosis
2	0
5	1
1	0
7	1
⋮	



Classification: Breast cancer detection

- ○ benign
- × malignant type1
- △ malignant type 2



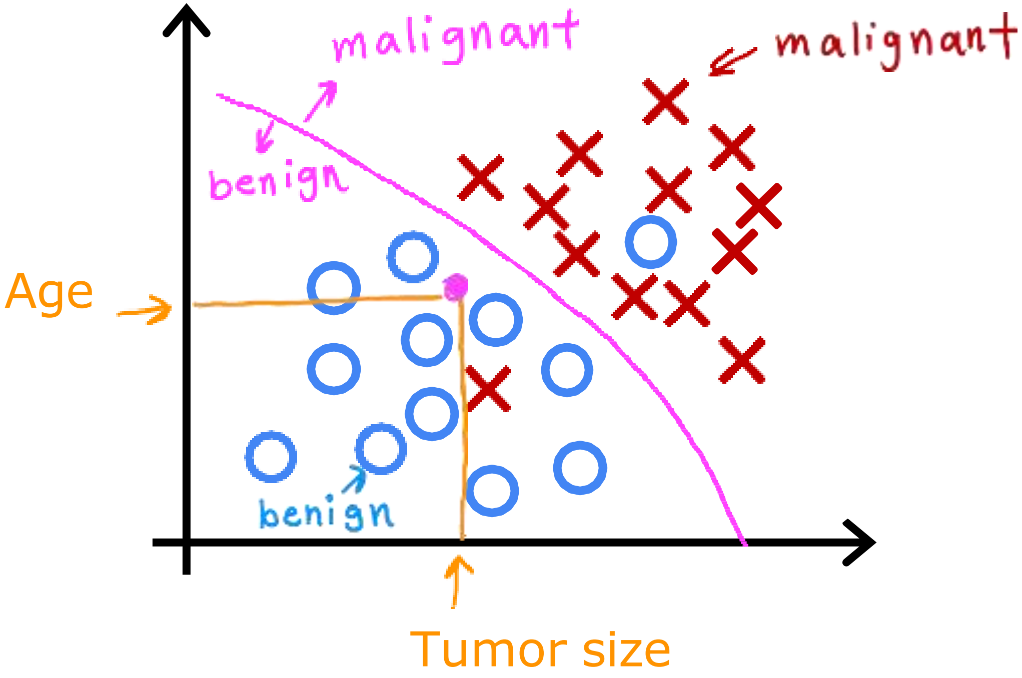
class category

Classification

predict categories cat dog benign malignant 0, 1, 2

small number of possible outputs

Two or more inputs



Supervised learning

Learns from being given "right answers"

Regression

Predict a number

infinitely many possible outputs

Classification

predict categories

small number of possible outputs

Stanford ONLINE

DeepLearning.AI

Machine Learning Overview

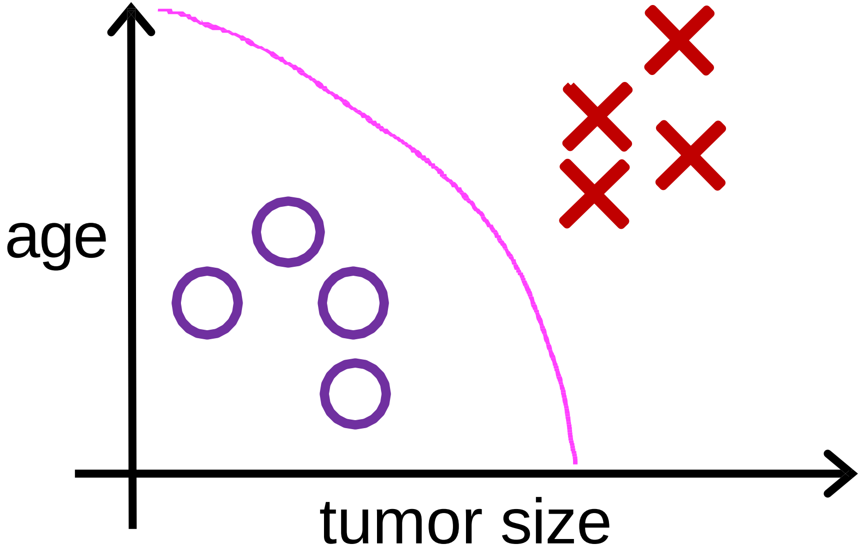
Unsupervised Learning Part 1

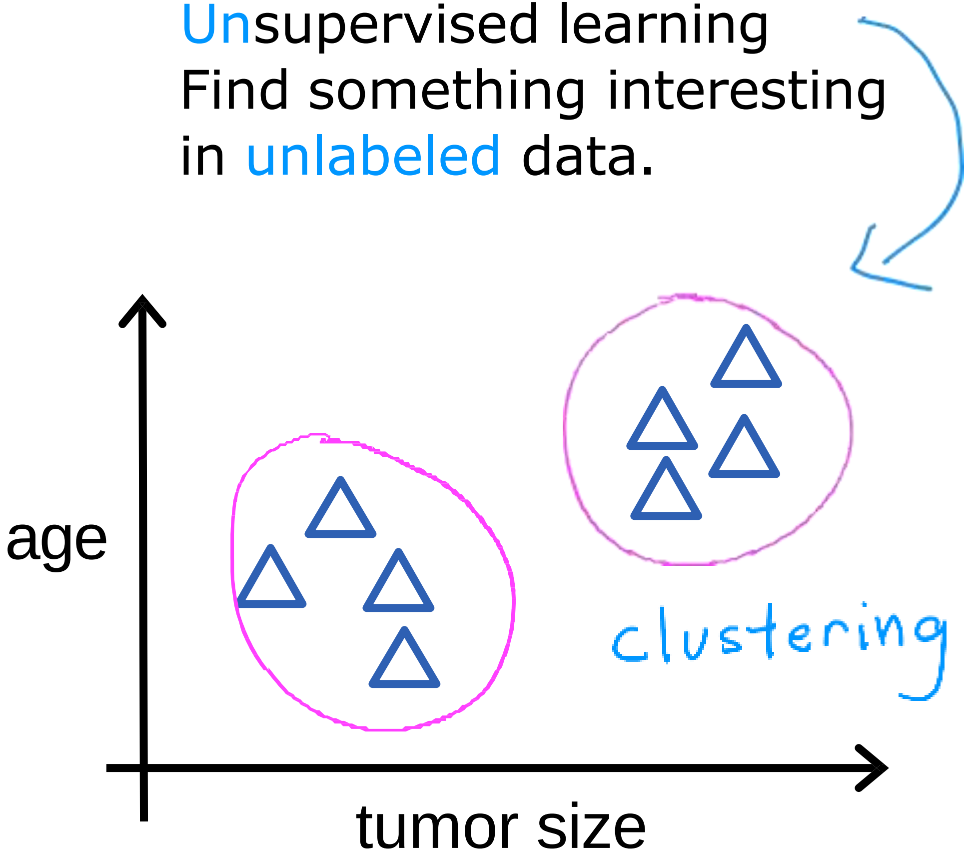
Previous: Supervised learning

Now: Unsupervised learning

Supervised learning
Learn from data labeled with the "right answers"

Unsupervised learning
Find something interesting in unlabeled data.





Clustering: Google news

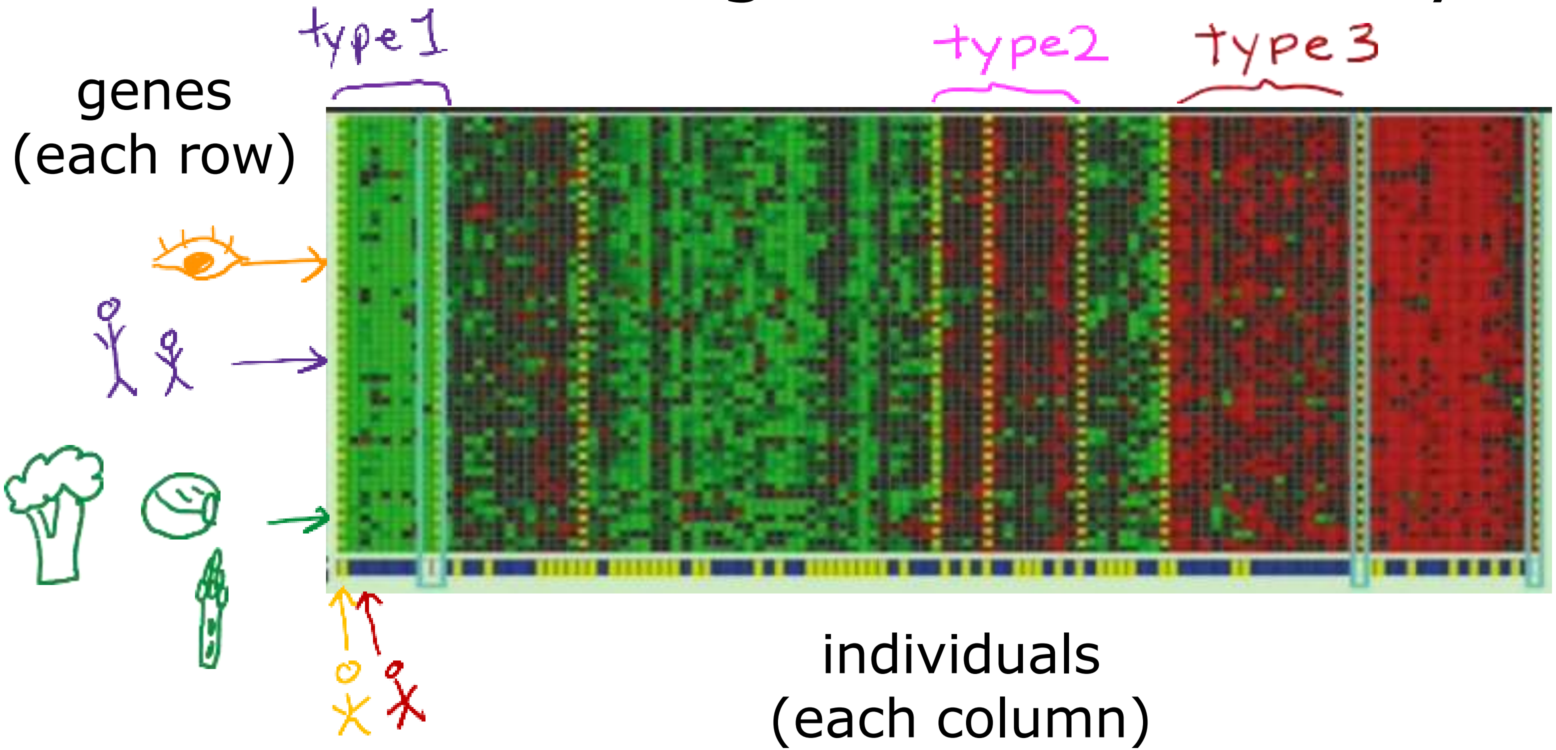
Giant panda gives birth to rare twin cubs at Japan's oldest zoo
USA TODAY · 6 hours ago

- **Giant panda gives birth to twin cubs at Japan's oldest zoo**
  CBS News · 7 hours ago
- **Giant panda gives birth to twin cubs at Tokyo's Ueno Zoo**
  WHBL News · 16 hours ago
- **A Joyful Surprise at Japan's Oldest Zoo: The Birth of Twin Pandas**
  The New York Times · 1 hour ago
- **Twin Panda Cubs Born at Tokyo's Ueno Zoo**
  PEOPLE · 6 hours ago

View Full Coverage

Clustering: DNA microarray

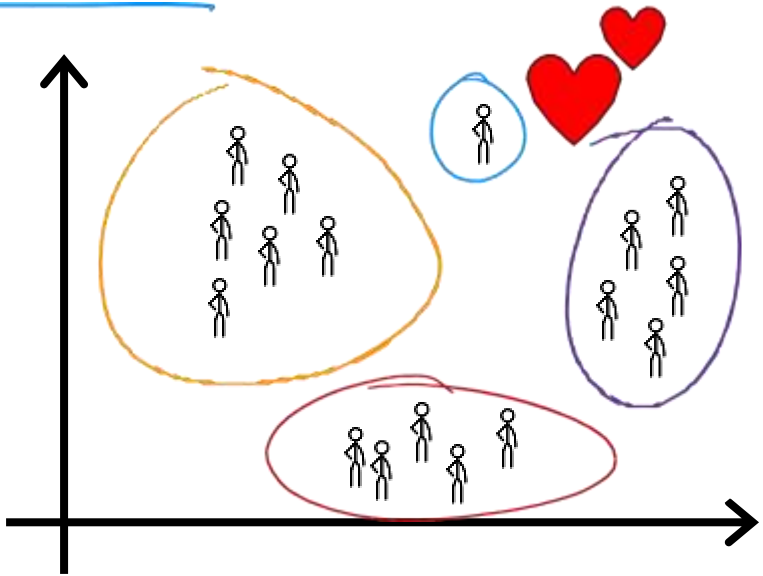
genes
(each row)



individuals
(each column)

Clustering: Grouping customers

growing skills
and knowledge



develop
career

stay updated
with AI

Stanford
ONLINE

DeepLearning.AI

Machine Learning Overview

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.
- [x] Given a set of news articles found on the web, group them into sets of articles about the same story.
- [x] 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

Stanford ONLINE

DeepLearning.AI

Machine Learning Overview

Jupyter Notebooks

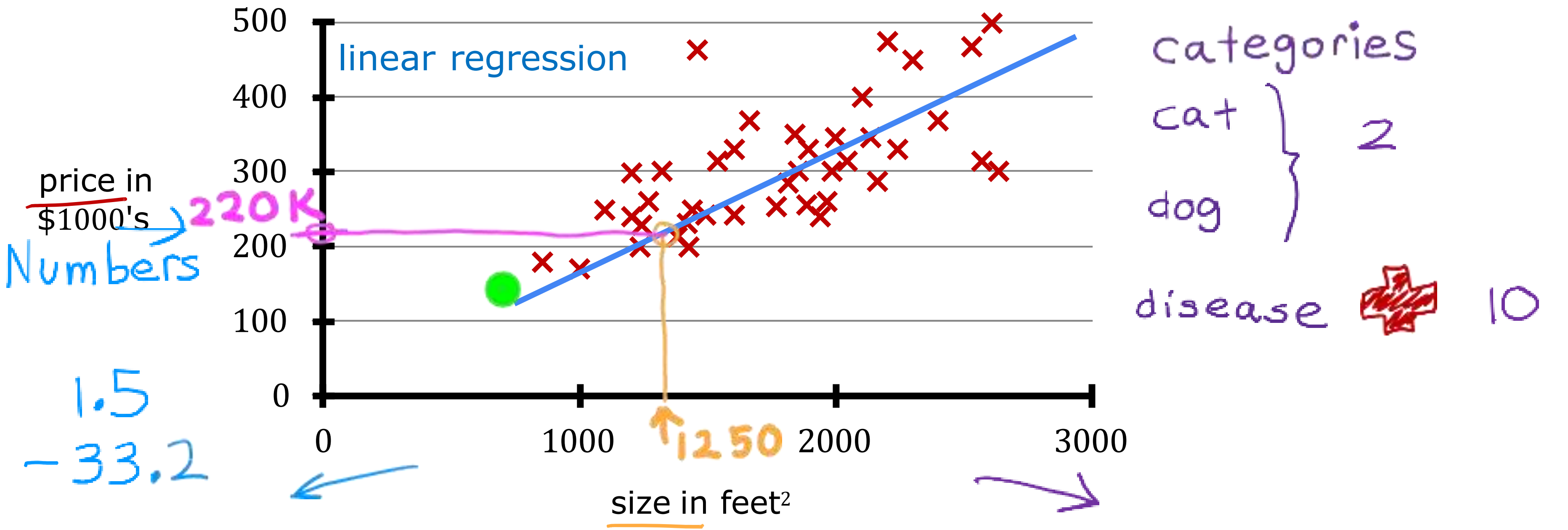
Stanford ONLINE

DeepLearning.AI

Linear Regression with One Variable

Linear Regression Model Part 1

House sizes and prices

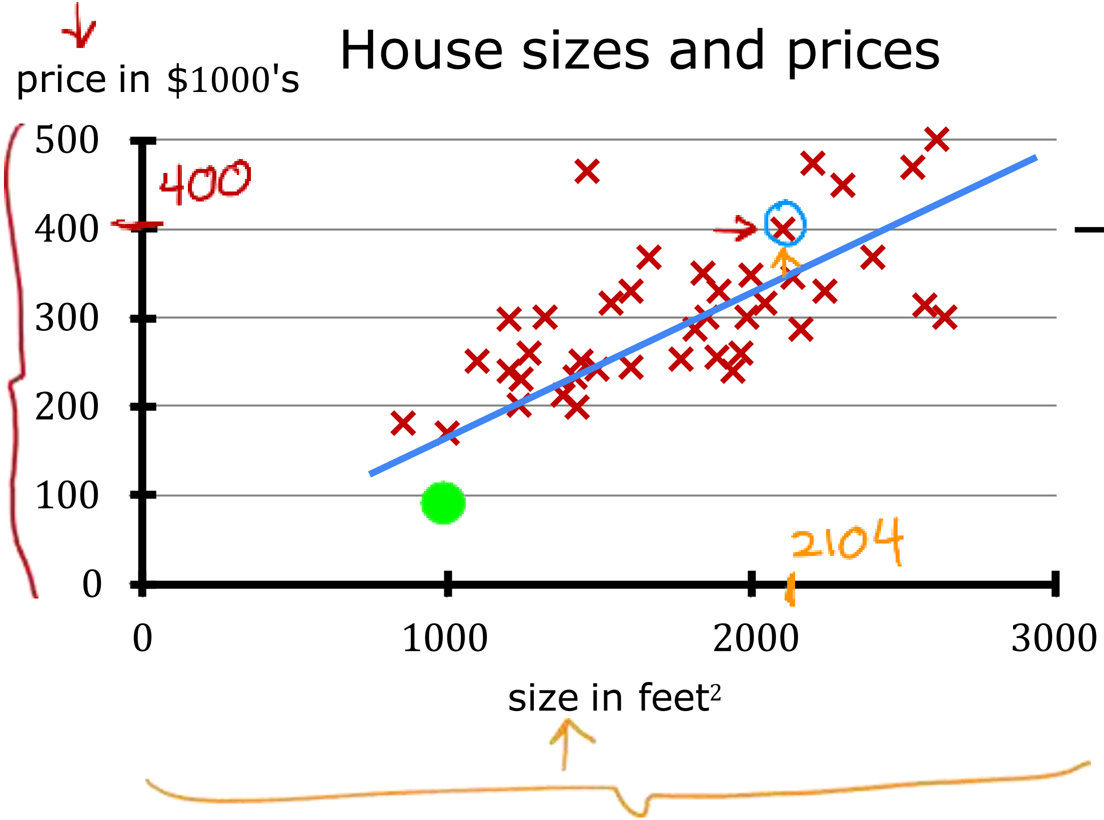


Regression model
Predicts numbers
Infinitely many possible outputs

Supervised learning model
Data has “right answers”

Classification model
Predicts categories
Small number of possible outputs

House sizes and prices



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

$m = 47$

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

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

$x^{(2)} = 1416$

$x^{(2)} \neq x^2$ 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 …)

Stanford ONLINE

DeepLearning.AI

Linear Regression with One Variable

Linear Regression Model Part 2

training set

features
targets

↓

learning algorithm

↓

$x \rightarrow \boxed{f} \rightarrow \hat{y}$

"y-hat"

~~hypothesis~~ function

feature — model — prediction (estimated y)

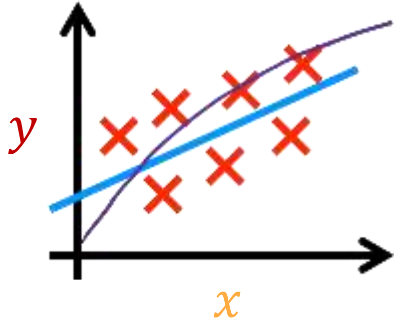
target

size $\rightarrow \boxed{f} \rightarrow$ 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

Stanford ONLINE

DeepLearning.AI

Linear Regression with One Variable

Cost Function

Training set

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

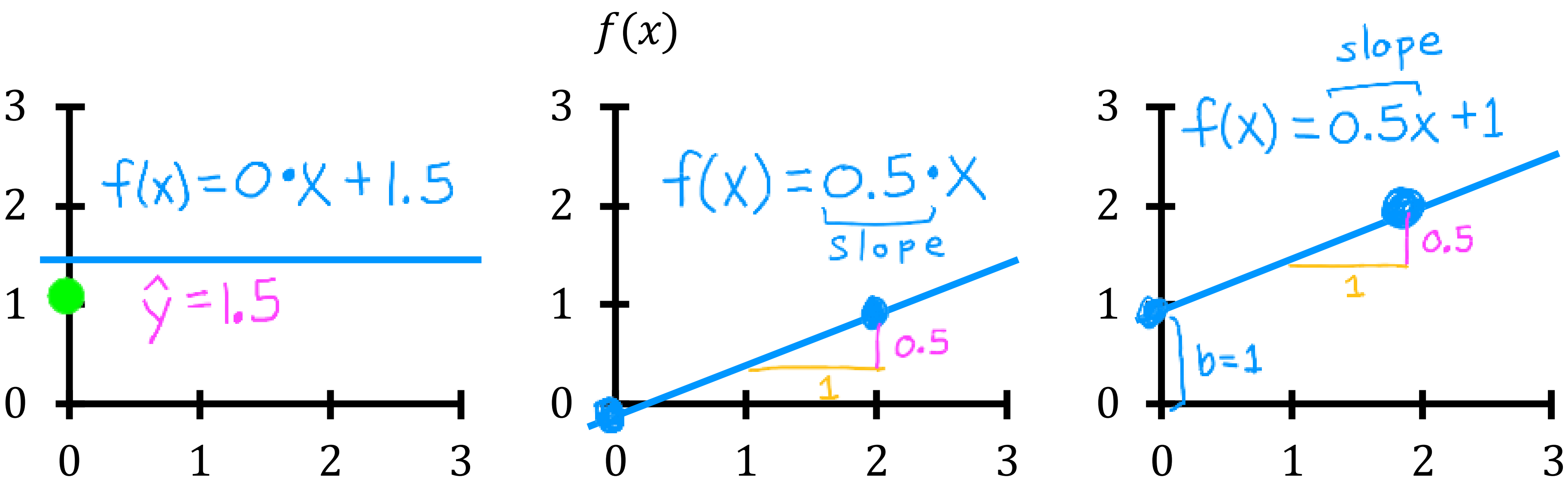
Model: $f_{w,b}(x) = wx + b$

w, b: parameters

coefficients
weights

What do w, b do?

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

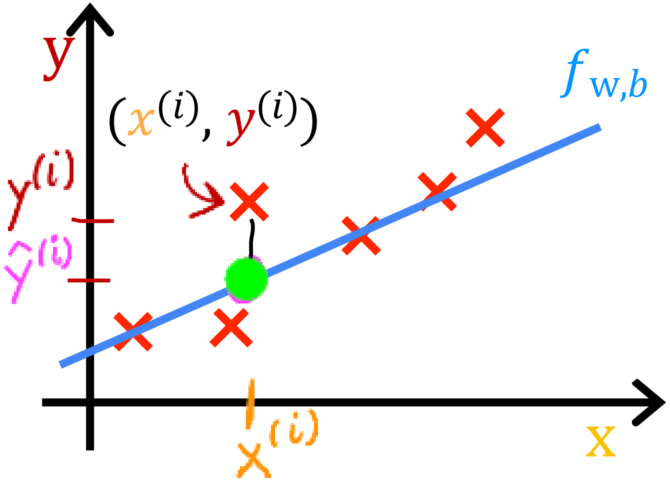


→ $w = 0$
→ $b = 1.5$
y-intercept

→ $w = 0.5$
→ b = 0

→ $w = 0.5$
→ $b = 1$

Cost function: Squared error cost function



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

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

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

error

m = number of training examples

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

intuition (next!)

Find w, b:

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

Stanford ONLINE

DeepLearning.AI

Linear Regression with One Variable

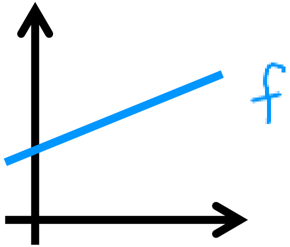
Cost Function Intuition

model:

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

parameters:

$$w, b$$



cost function:

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

goal:

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

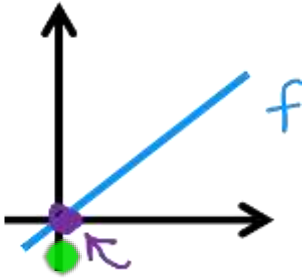
simplified

$$f_w(x) = wx$$

b = ∅

0 0

w



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

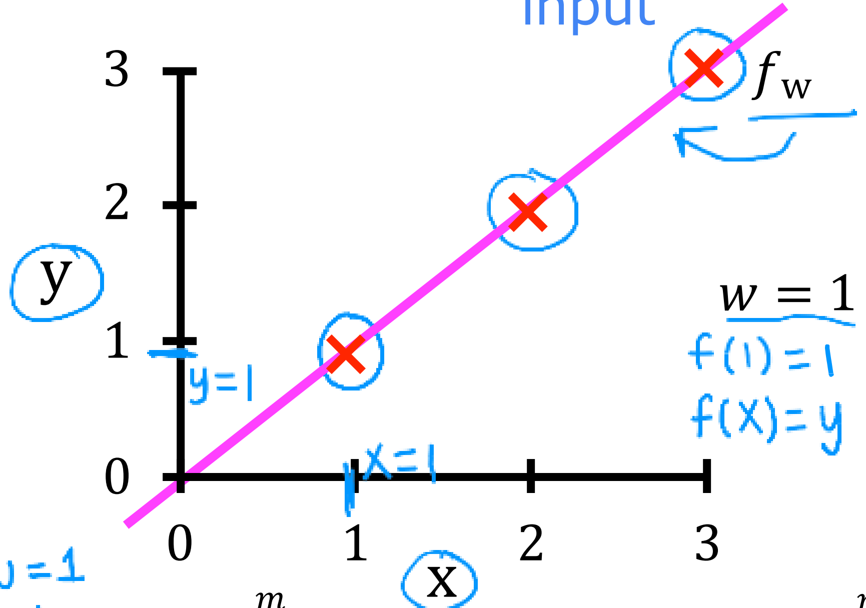
$wx^{(i)}$

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

$f_w(x)$

(for fixed w, function of x)

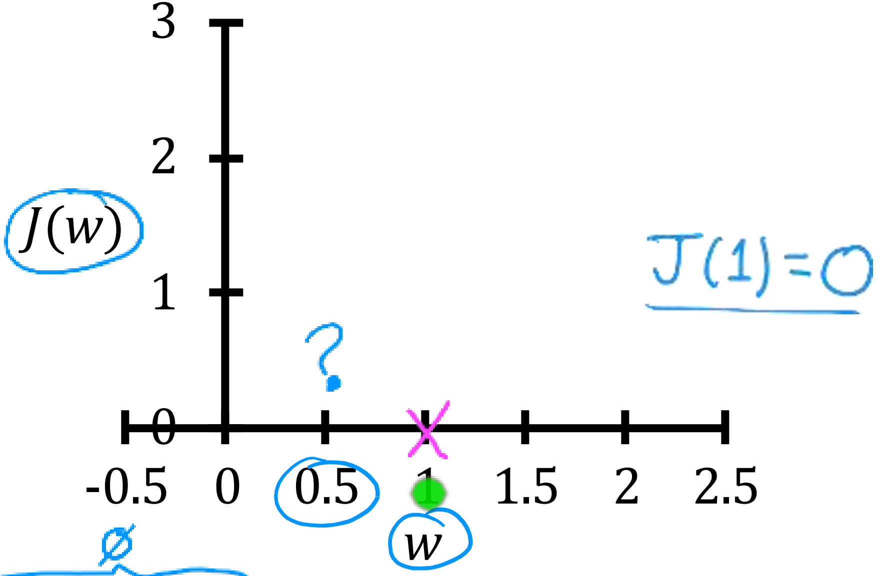
input



$J(w)$

(function of w)

parameter

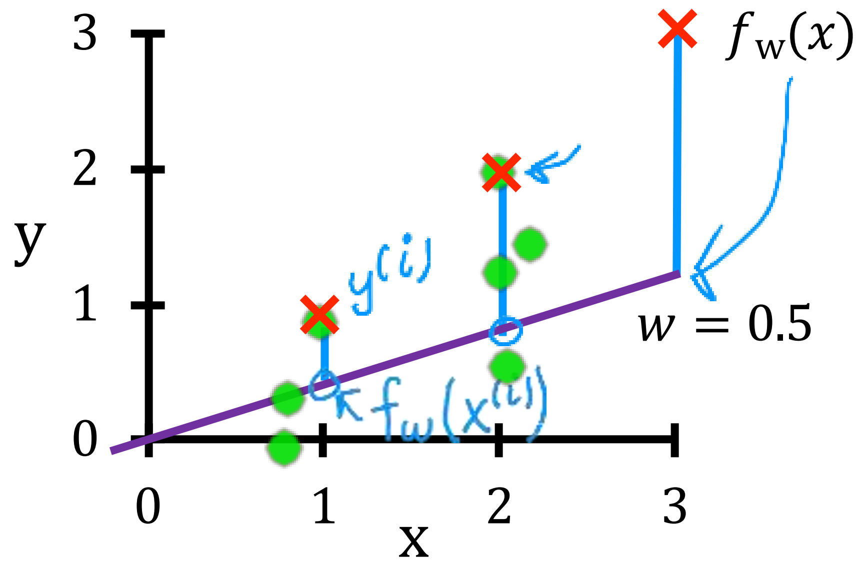


w=1

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

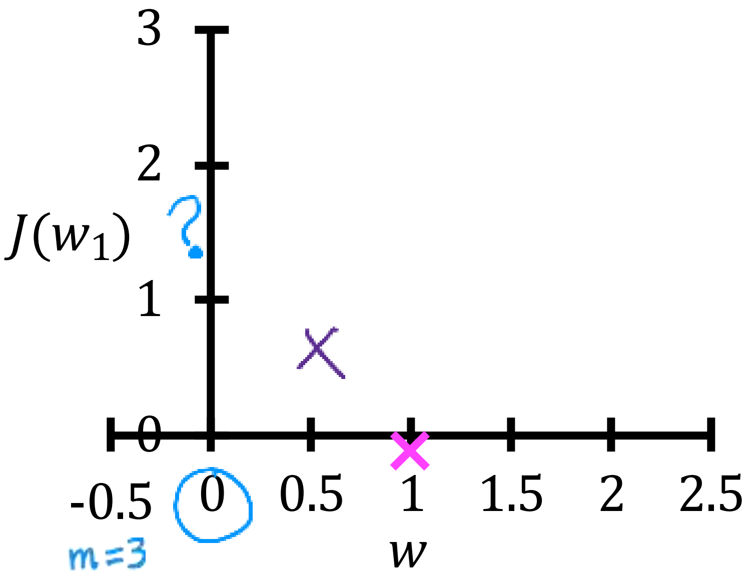
$f_w(x)$

(function of x)



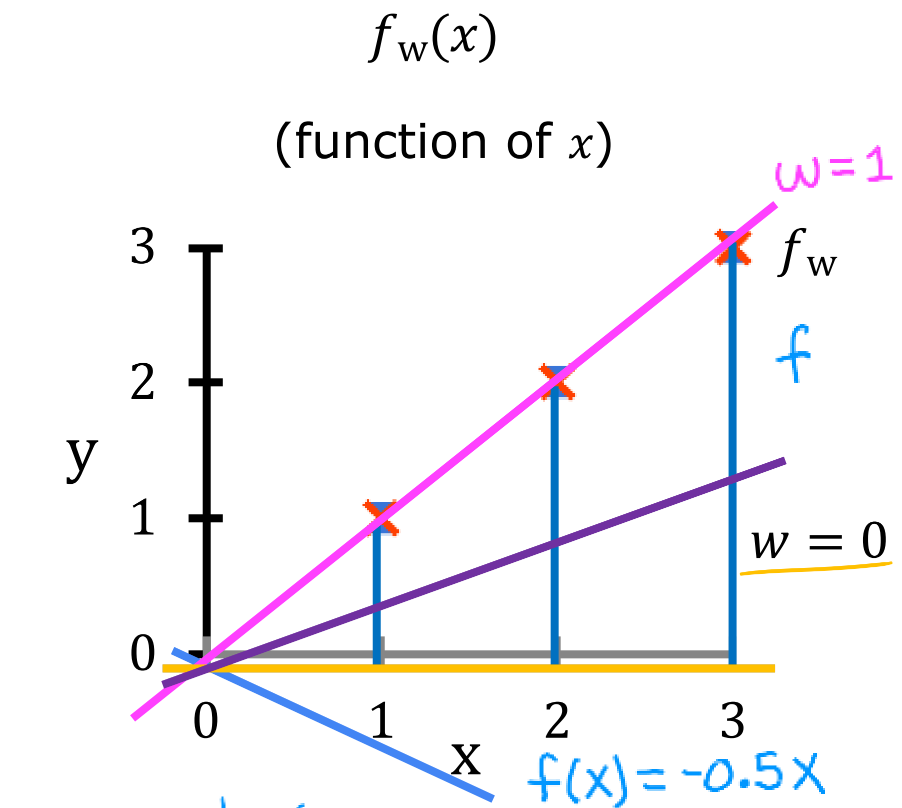
$J(w)$

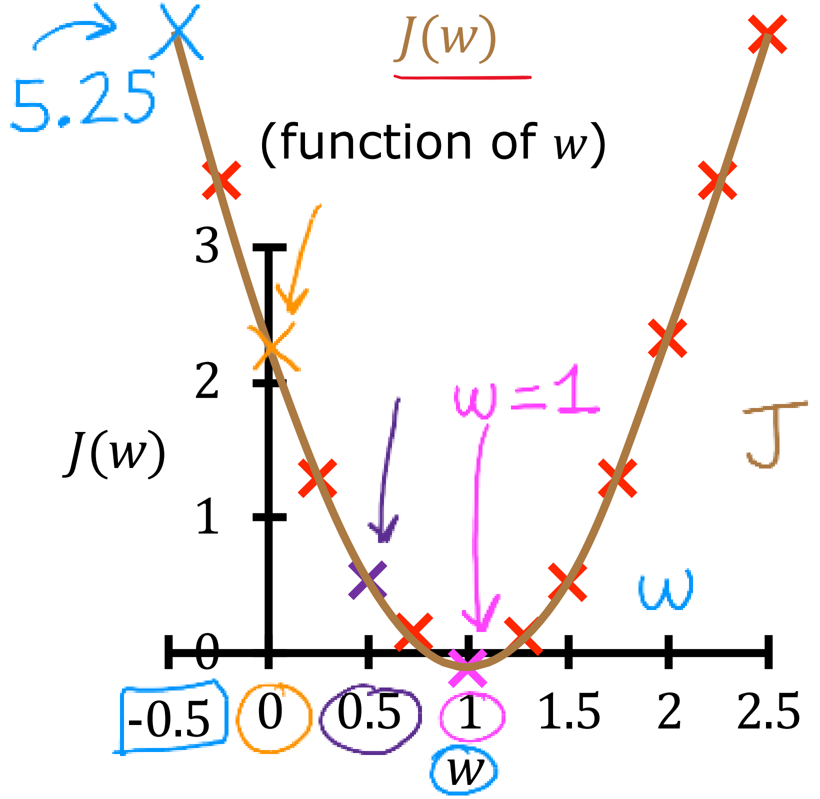
(function of w)



$m=3$

$$J(0.5) = \frac{1}{2m}\left[(0.5-1)^2+(1-2)^2+(1.5-3)^2\right] = \frac{1}{2\times3}[3.5] = \frac{3.5}{6} \approx 0.58$$





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

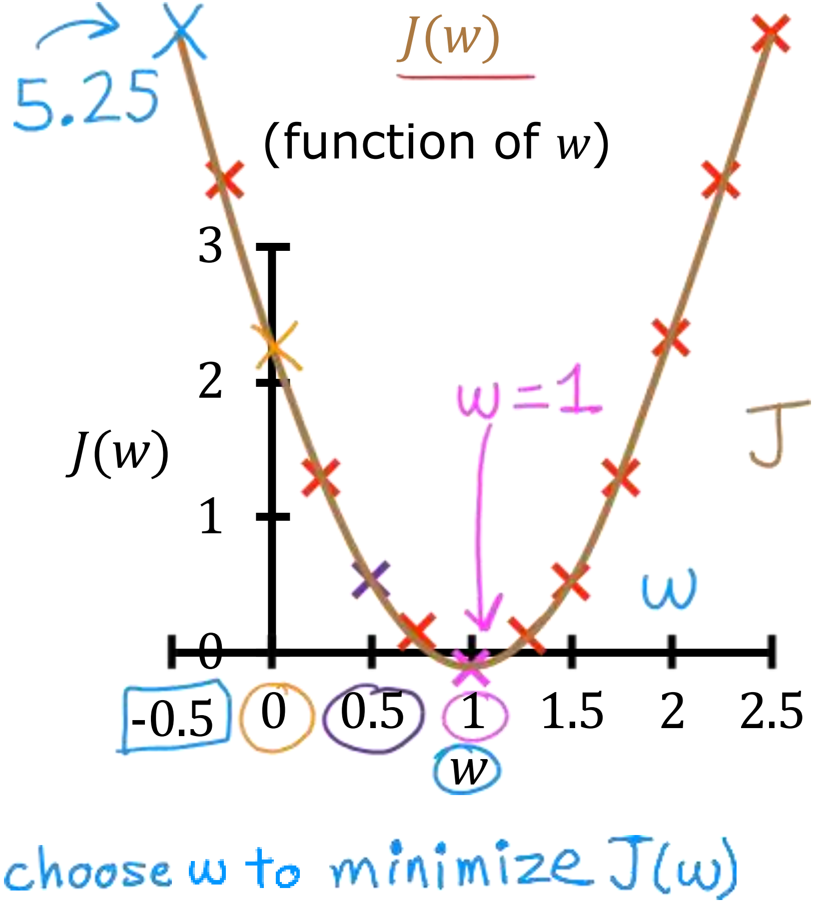
how to choose w?

goal of linear regression:

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

general case:

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



Stanford ONLINE

DeepLearning.AI

Linear Regression with One Variable

Visualizing the Cost Function

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

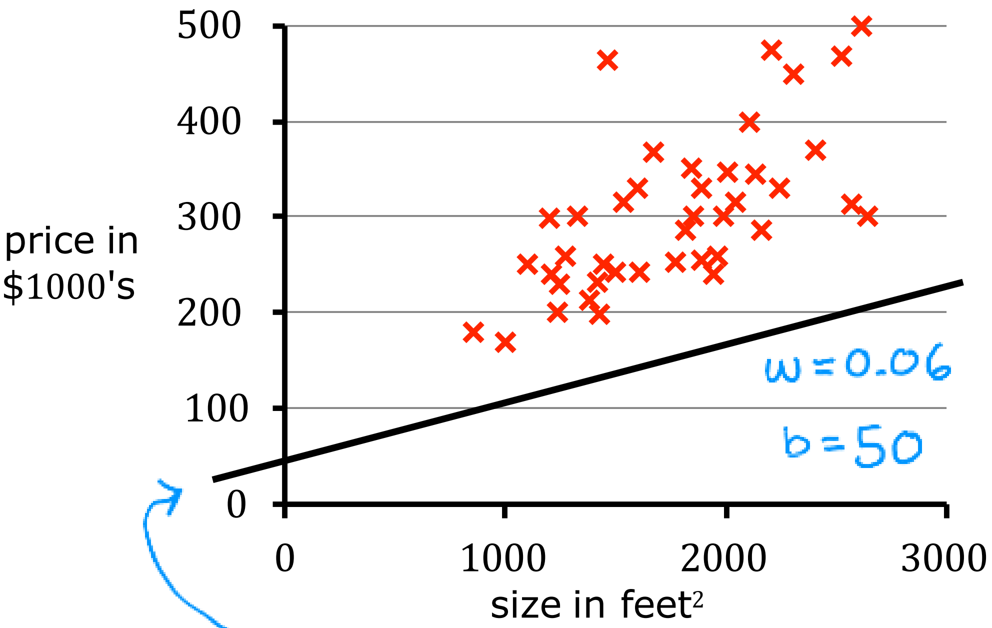
Parameters w, b ~~before: b=0~~

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

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

$f_{w,b}$

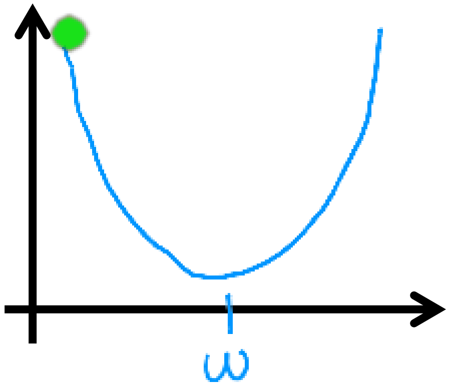
(function of x)



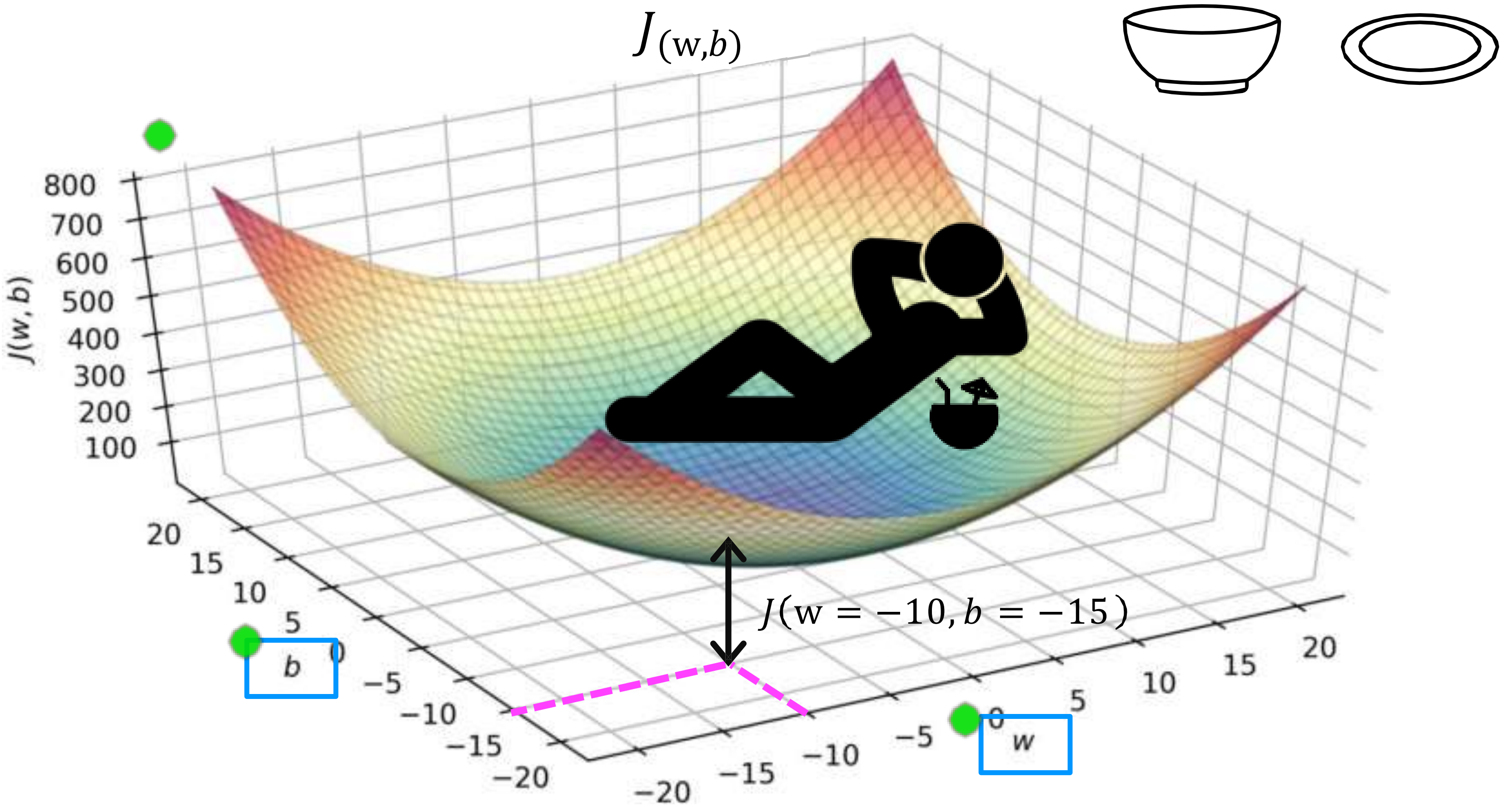
$$\boxed{f_{w,b}(x) = 0.06x + 50}$$

J

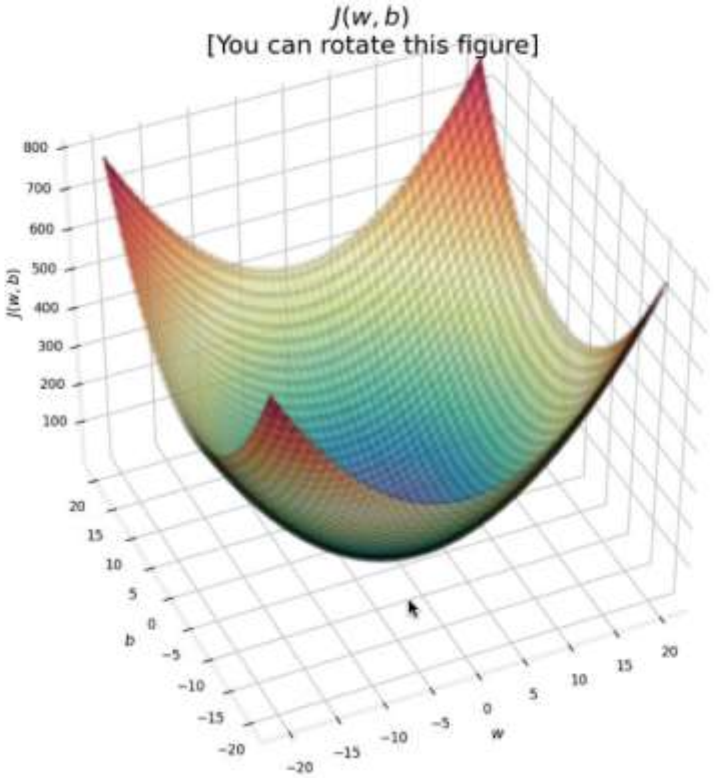
(function of w, b)



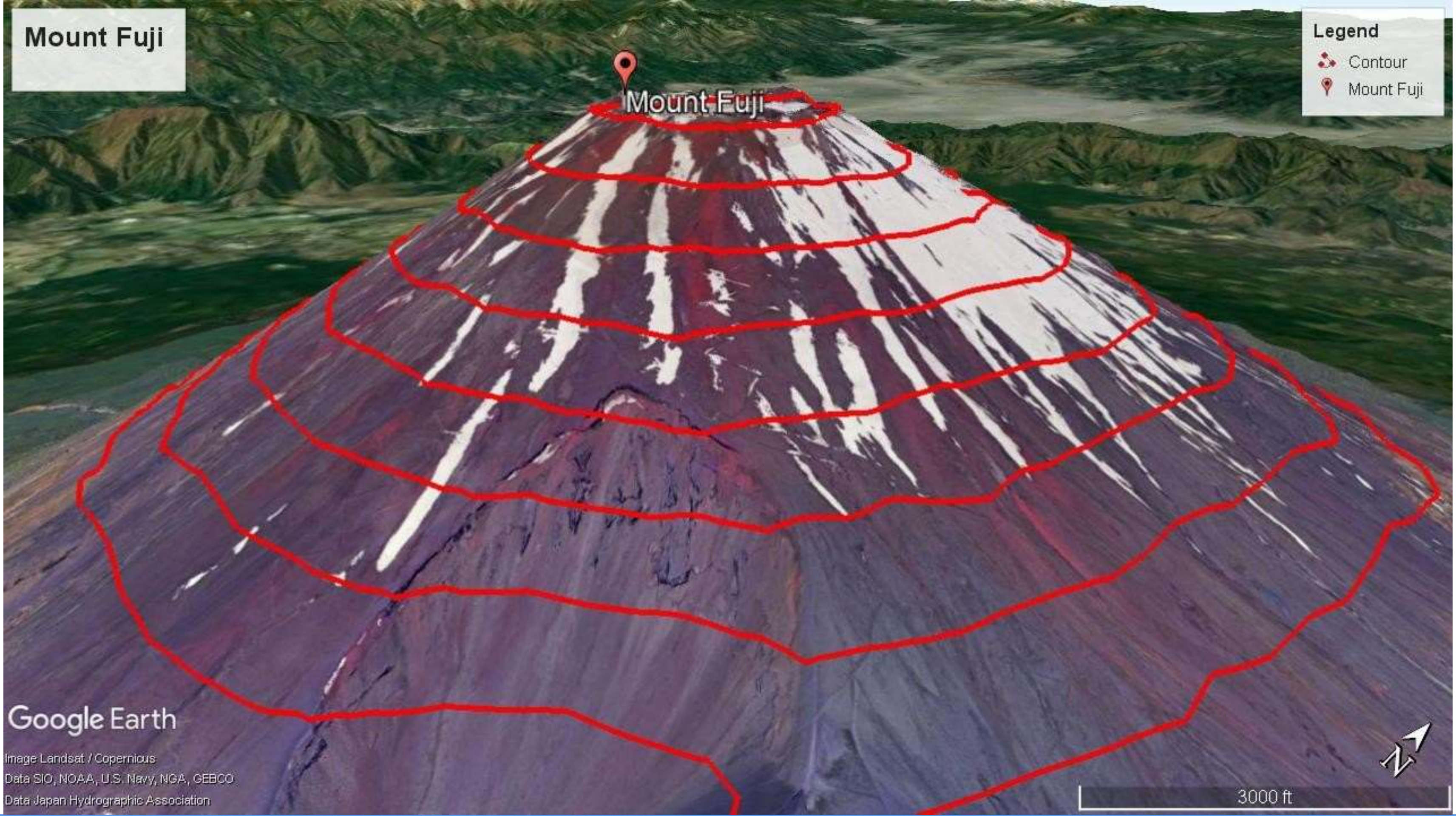
$\boxed{w, b}$

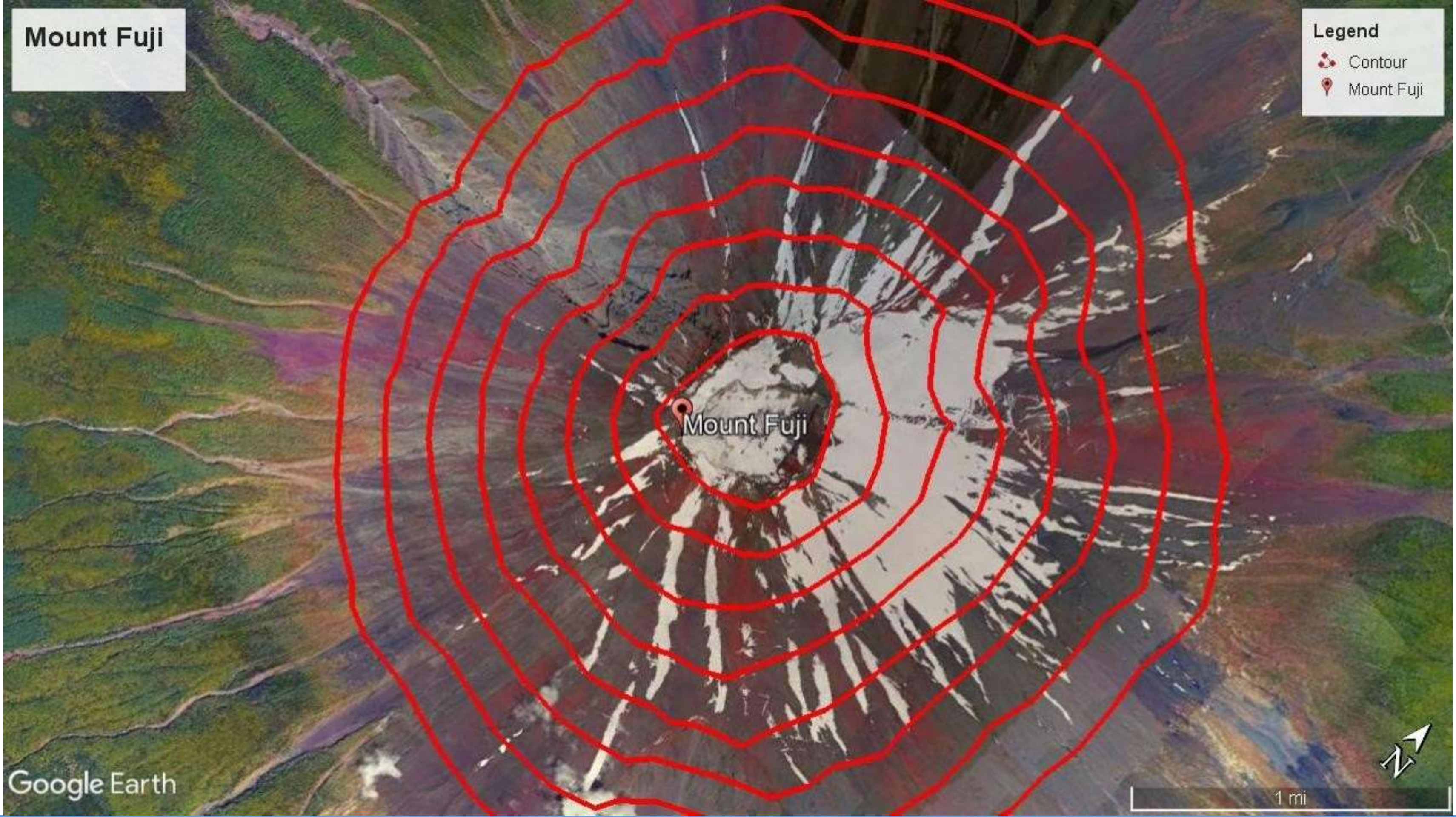

$J_{(w,b)}$
800
700
600
500
400
300
200
100
J(w, b)
20
15
10
5
0
−5
−10
−15
−20
b
w
$J(w = -10, b = -15)$

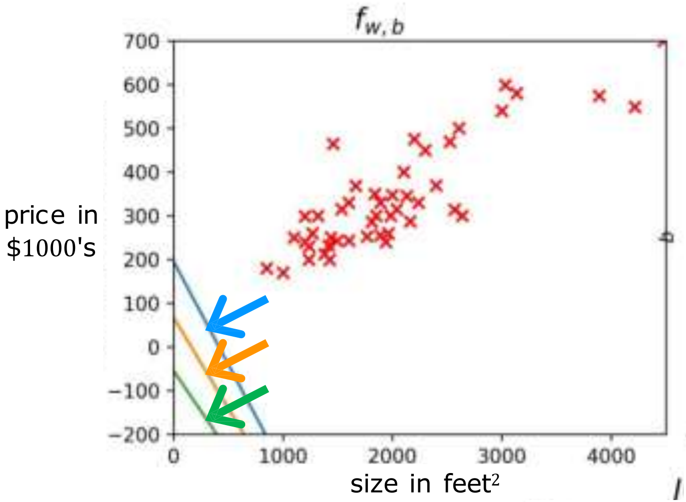
3D surface plot

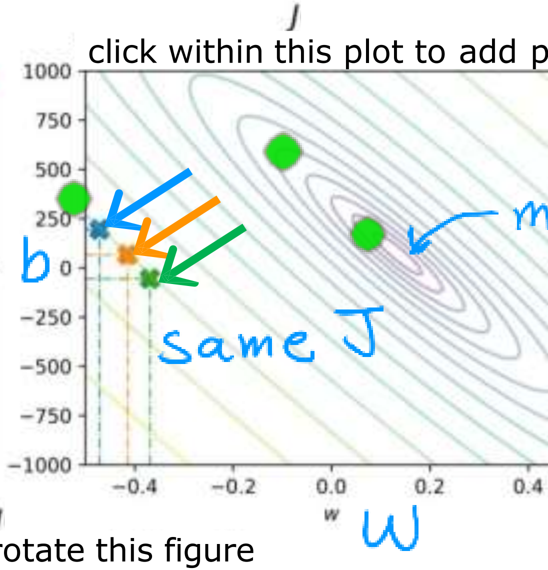


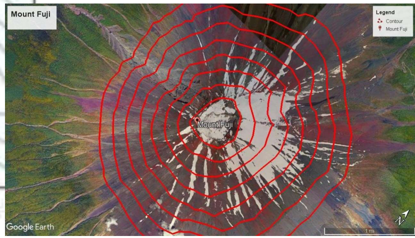
Alternative contour plot

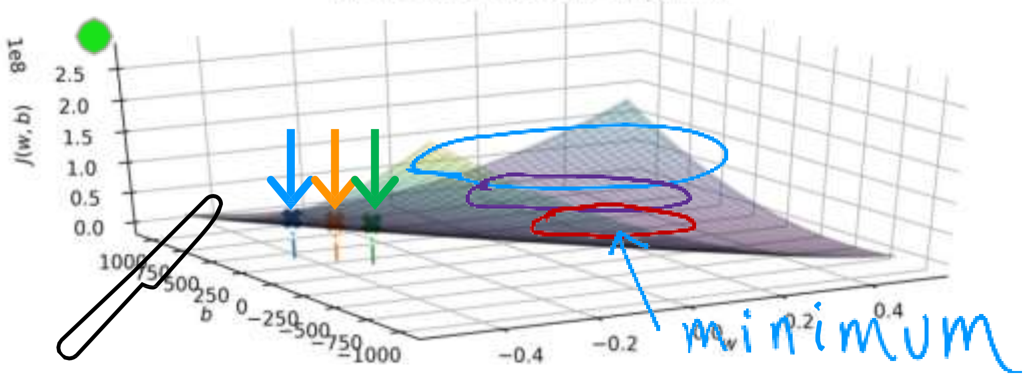

Mount Fuji
Mount Fuji
Legend
Contour
Mount Fuji
Google Earth
Image Landsat / Copernicus
Data SIO, NOAA, U.S. Navy, NGA, GEBCO
Data Japan Hydrographic Association
3000 ft
N

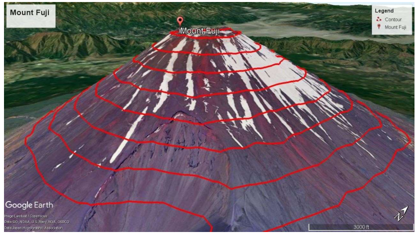

Mount Fuji
Legend
Contour
Mount Fuji
Mount Fuji
Google Earth
1 mi


$f_{w,b}$
price in $1000's
size in feet²
700
600
500
400
300
200
100
0
−100
−200
0
1000
2000
3000
4000


J
click within this plot to add points
1000
750
500
250
0
−250
−500
−750
−1000
b
−0.4
−0.2
0.0
0.2
0.4
w
minimum
same J
w


Mount Fuji
Legend
Google Earth


J
you can rotate this figure
1e8
2.5
2.0
1.5
1.0
0.5
0.0
J(w, b)
b
w
−0.4
−0.2
0.2
0.4
minimum

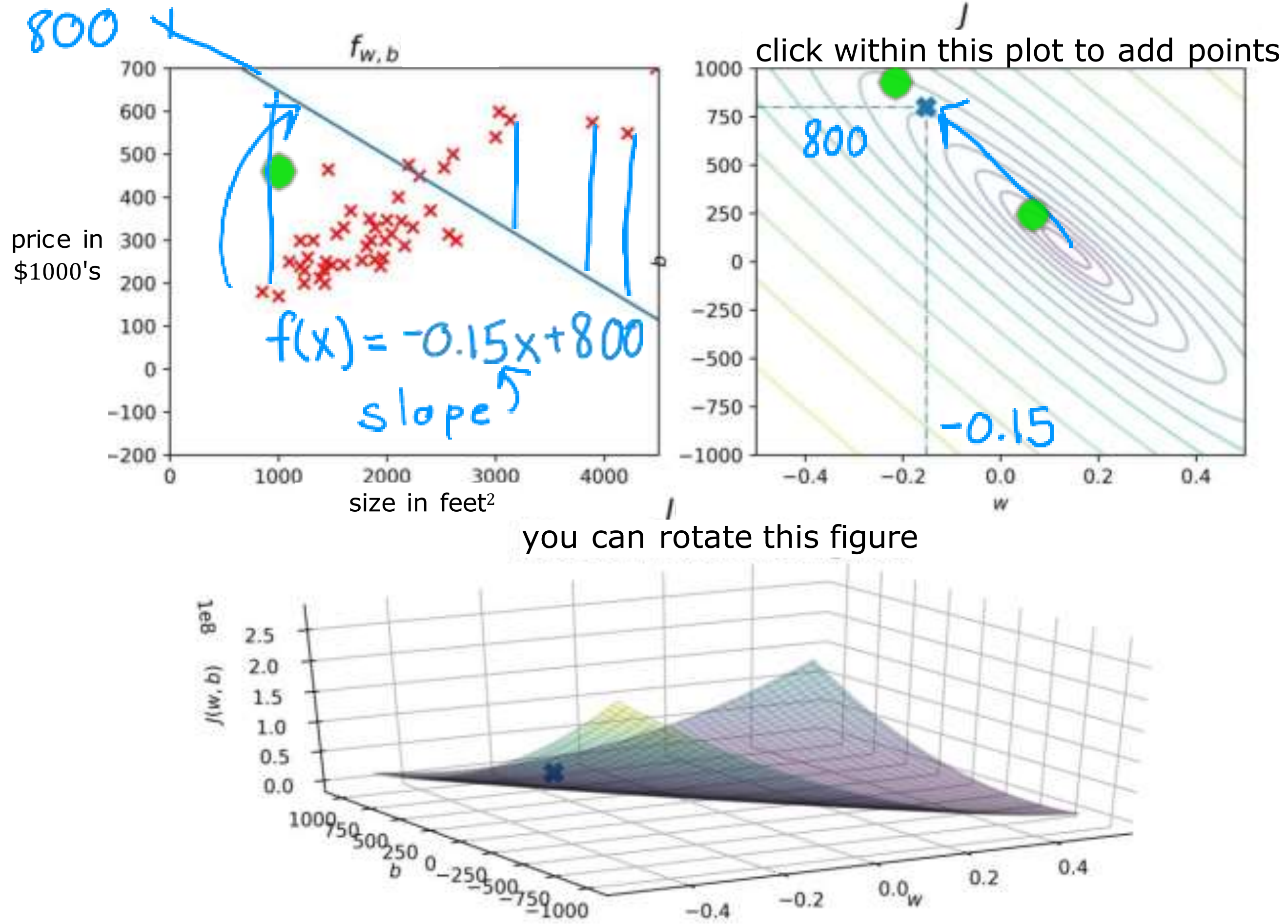

Mount Fuji
Mount Fuji
Legend
Google Earth

Stanford
ONLINE

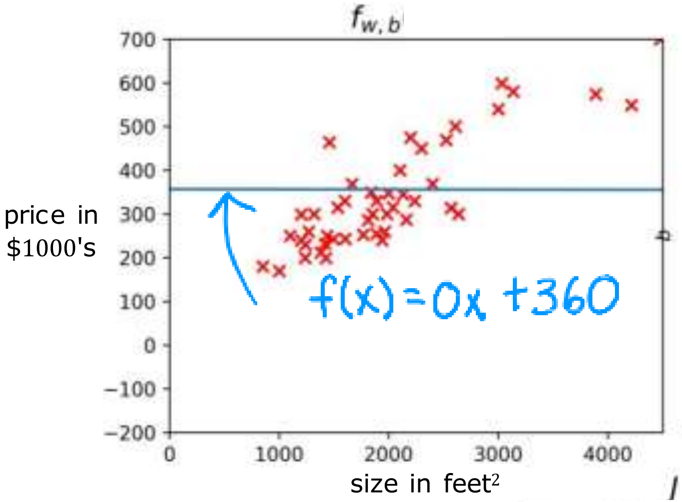
DeepLearning.AI

Linear Regression with One Variable

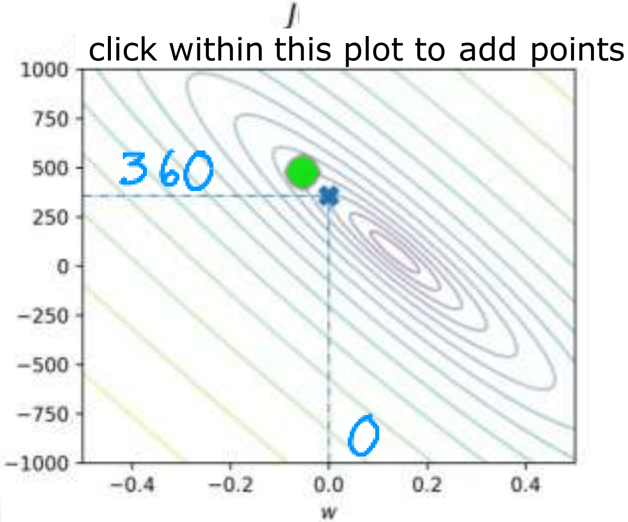
Visualization examples



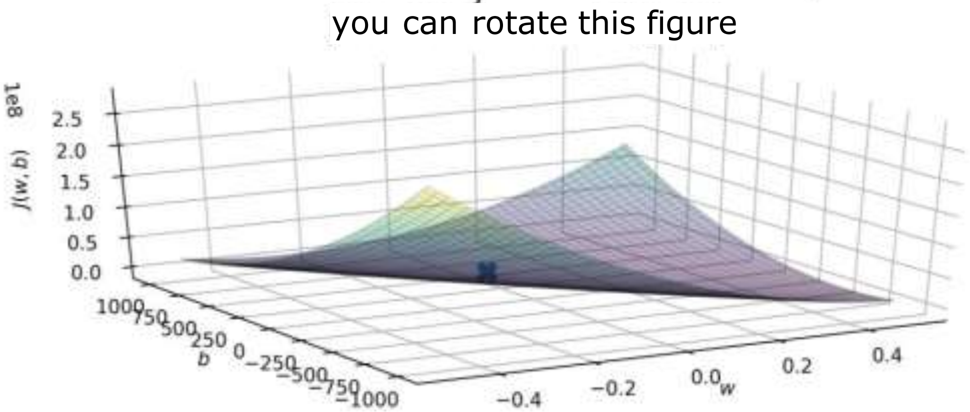
800
$f_{w,b}$
price in $1000's
size in feet²
$f(x) = -0.15x + 800$
slope
J
click within this plot to add points
800
-0.15
b
w
you can rotate this figure
1e8
J(w, b)



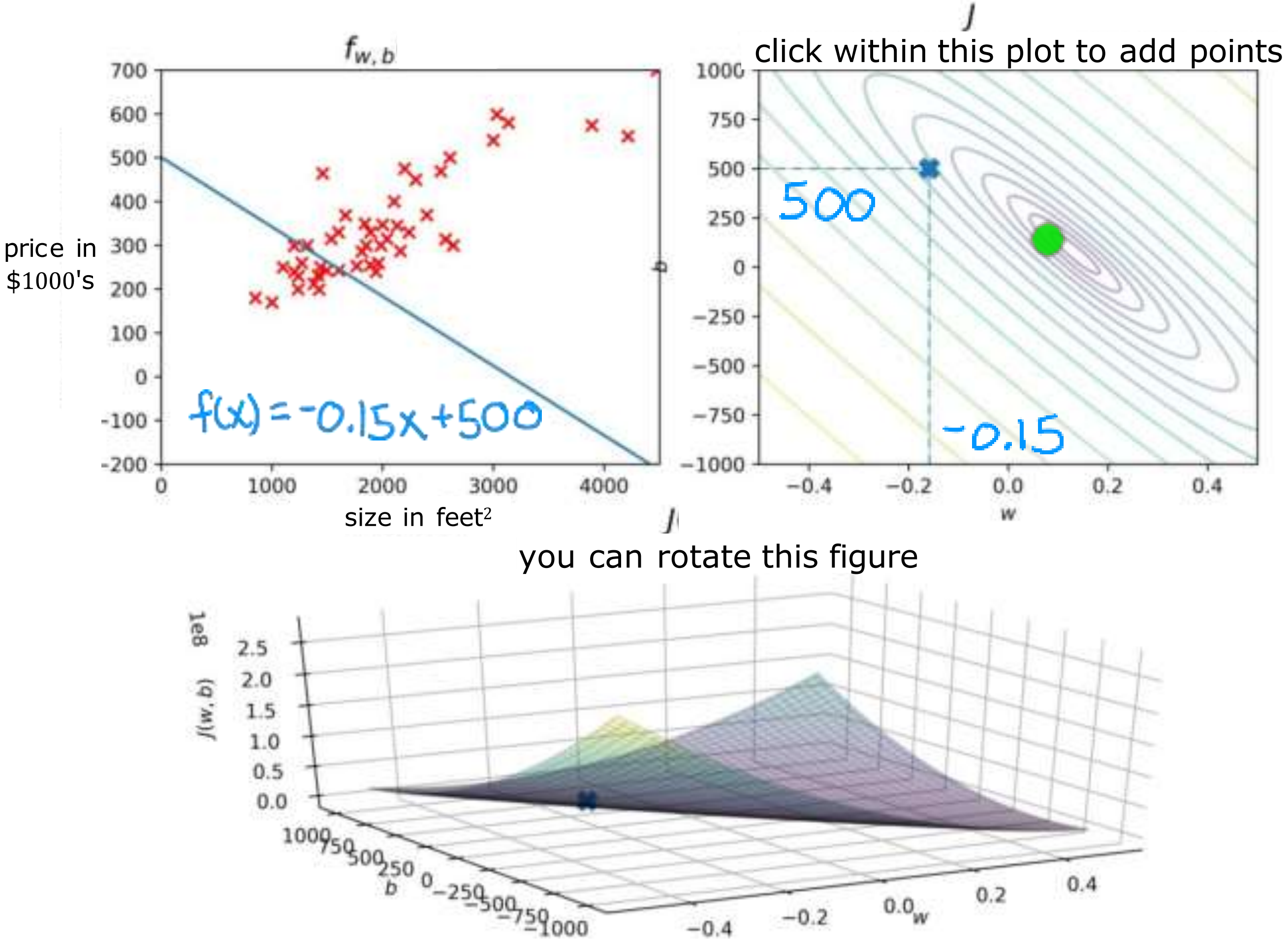
$f_{w,b}$
price in $1000's
size in feet²
f(x)=0x +360



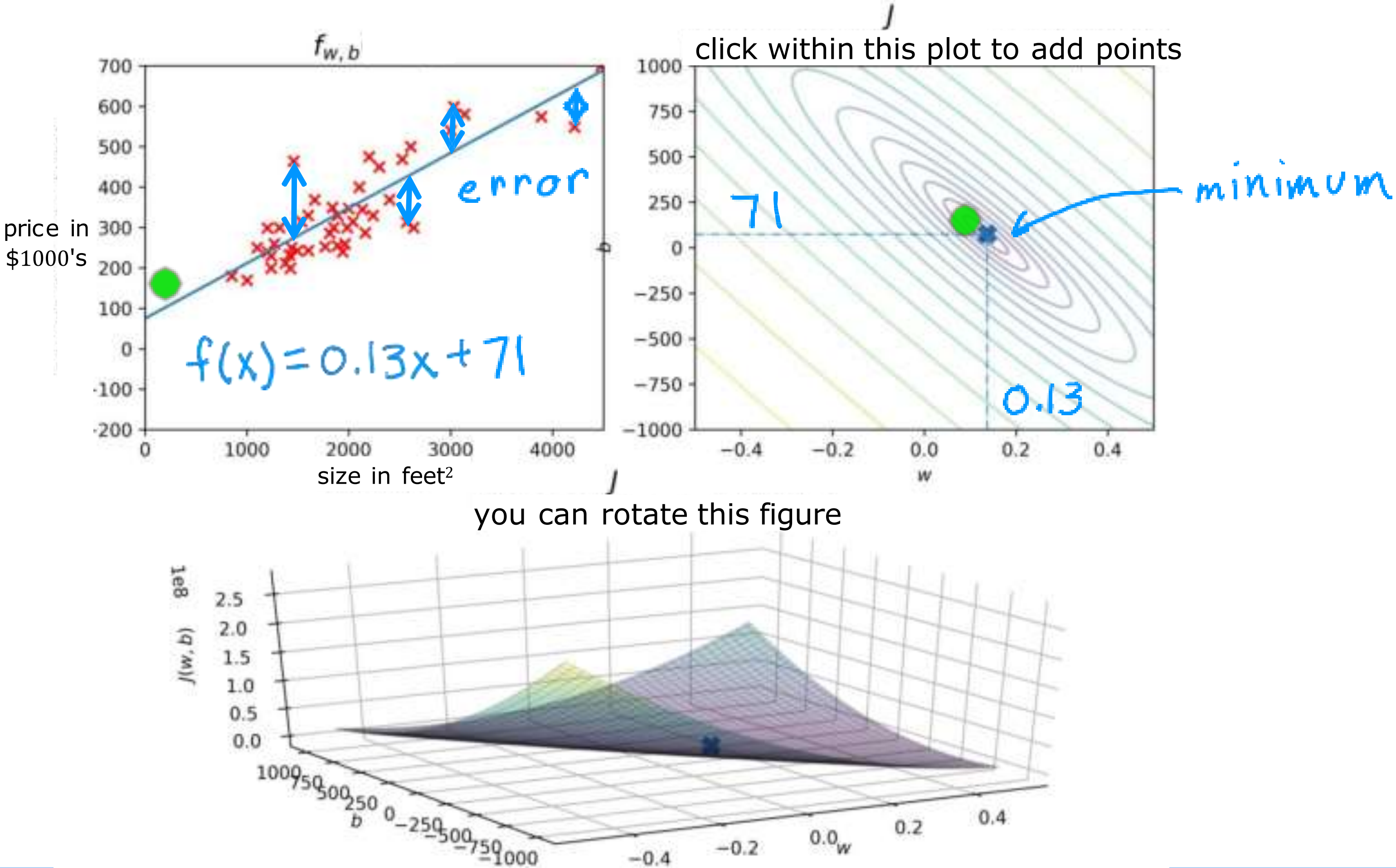
J
click within this plot to add points
360
0
w
b



J
you can rotate this figure
1e8
J(w, b)
b
w



$f_{w,b}$
price in $1000's
size in feet^2
$f(x) = -0.15x + 500$
J
click within this plot to add points
500
-0.15
w
b
you can rotate this figure
1e8
J(w, b)



$f_{w,b}$
price in $1000's
size in feet²
error
f(x)=0.13x+71
J
click within this plot to add points
71
minimum
0.13
w
b
J
you can rotate this figure
J(w, b)
1e8
w
b

Stanford
ONLINE

DeepLearning.AI

Training Linear Regression

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

may have >1 minimum

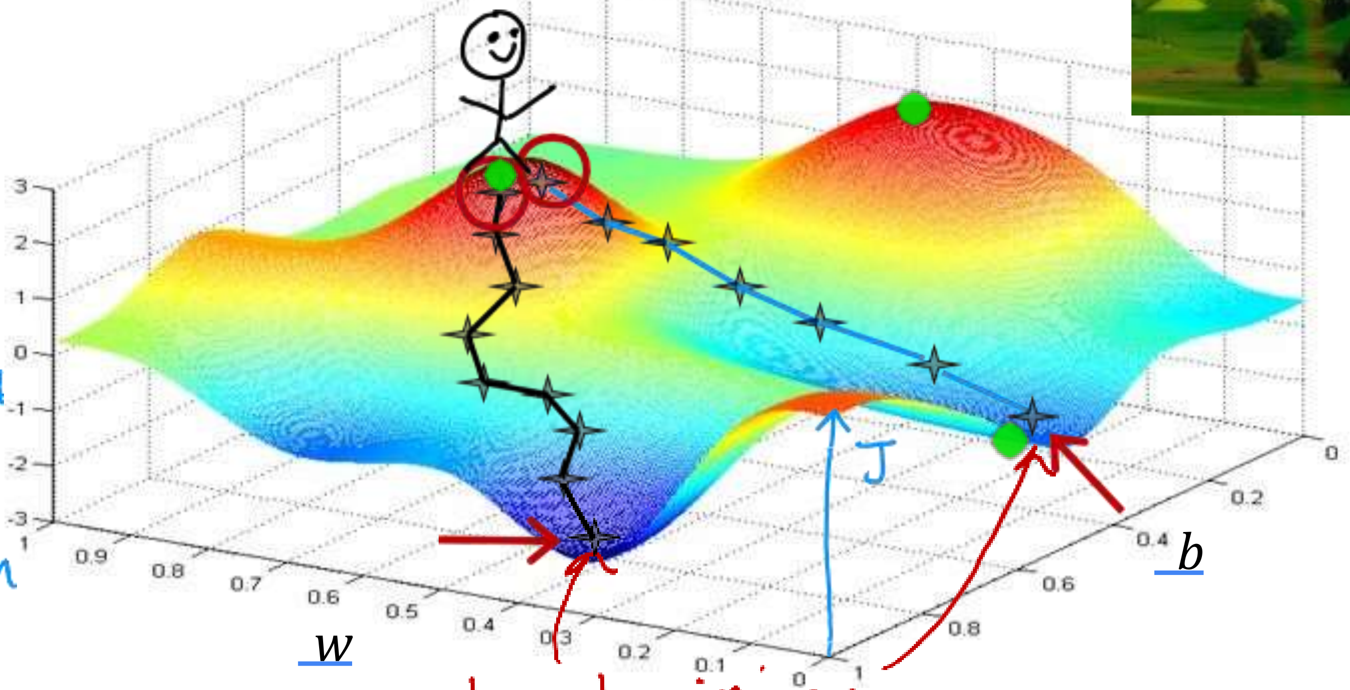
J not always

gradient descent

$J(w,b)$

not squared error cost

not linear regression



w

b

local minima

Stanford ONLINE

DeepLearning.AI

Training Linear Regression

Implementing Gradient Descent

Gradient descent algorithm

Repeat until convergence

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

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

Learning rate

Derivative

Simultaneously update w and b

Assignment

$a = c$

$a = a + 1$

Code

Truth assertion

$a = c$

$\cancel{a = a + 1}$

Math

a==c

Correct: Simultaneous update

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

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

$$w = tmp\_w$$

$$b = tmp\_b$$

Incorrect

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

$$w = tmp\_w$$

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

$$b = tmp\_b$$

Stanford
ONLINE

DeepLearning.AI

Training Linear Regression

Gradient Descent Intuition

Gradient descent algorithm

repeat until convergence {

learning rate

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

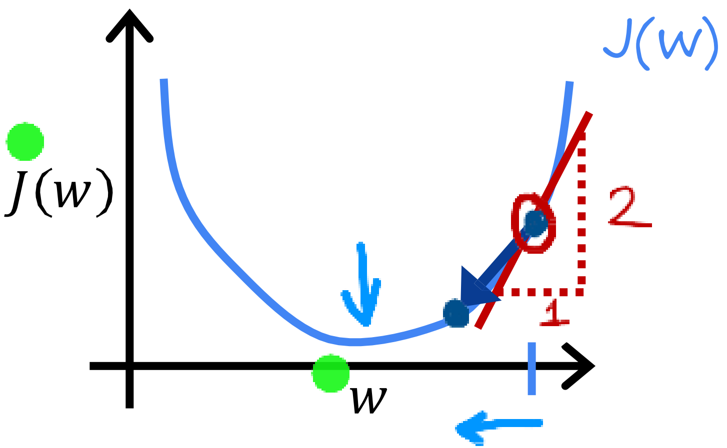
derivative

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

$J(w)$

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

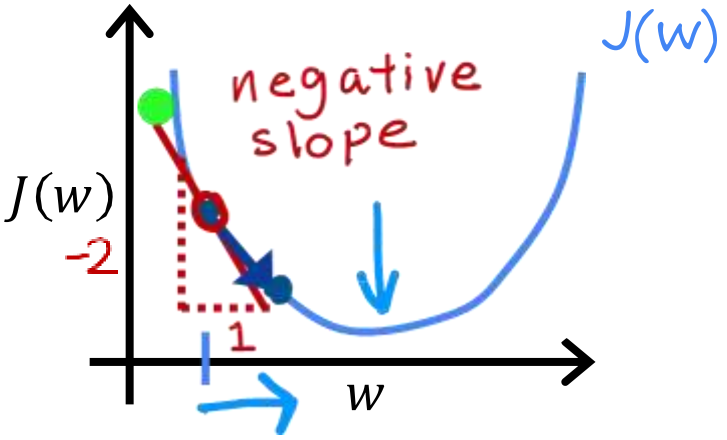
$$\min_{w} J(w)$$



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

>0

$$w = w - \underline{\alpha \cdot}\,(positive\; number)$$



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

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

Stanford
ONLINE

DeepLearning.AI

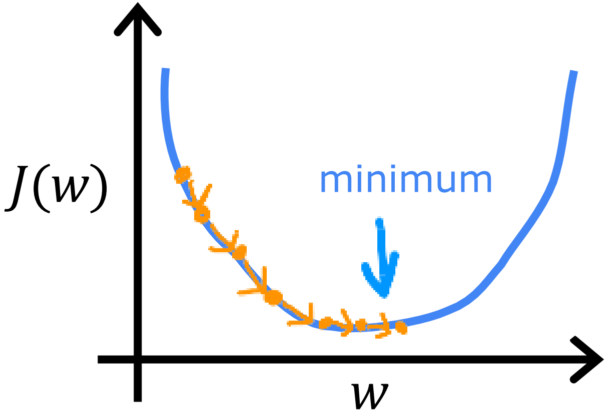
Training Linear Regression

Learning Rate

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

If α is too small...

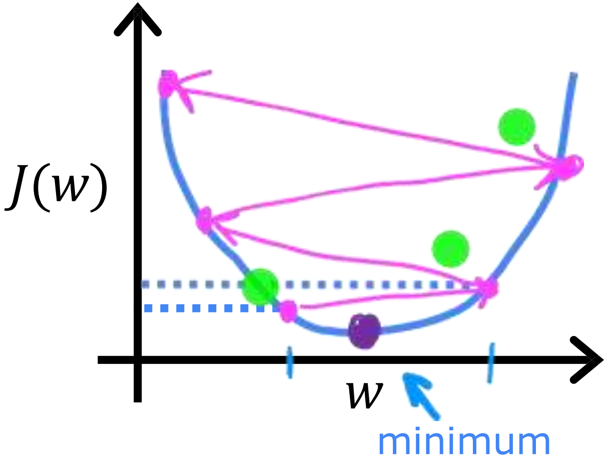
Gradient descent may be slow.

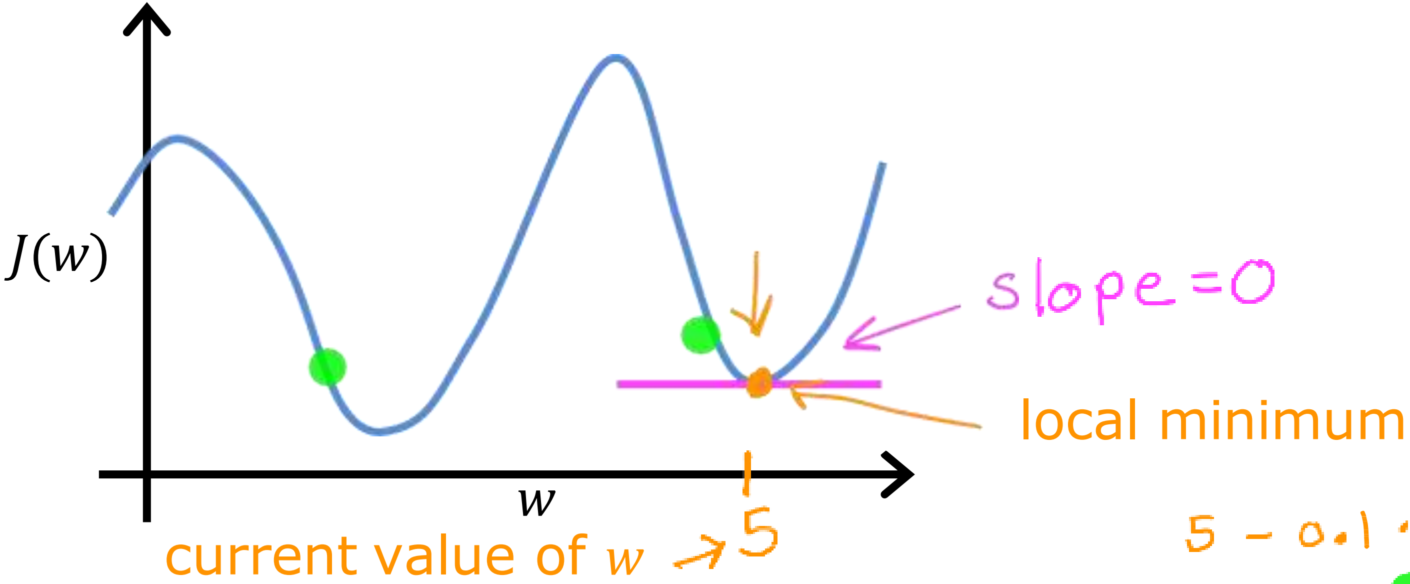


If α is too large...

Gradient descent may:

- Overshoot, never reach minimum
- Fail to converge, diverge





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

$5 - 0.1 \cdot 0$

$w = w - \alpha \cdot 0$

$w = w$

Can reach local minimum with fixed learning rate

α

smaller

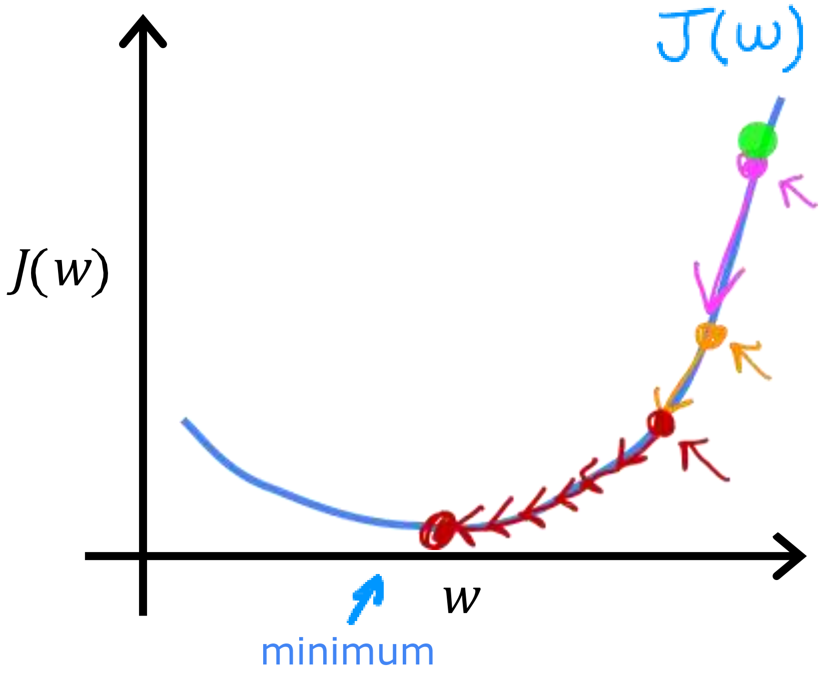
not as large

large

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

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

Can reach minimum without decreasing learning rate α



Stanford ONLINE

DeepLearning.AI

Training Linear Regression

Gradient Descent for Linear Regression

Linear regression model

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

Cost function

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

Gradient descent algorithm

repeat until convergence {

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

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

}

next slide is optional!

(Optional)

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

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

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

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

no $x^{(i)}$

Gradient descent algorithm

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

repeat until convergence {

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

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

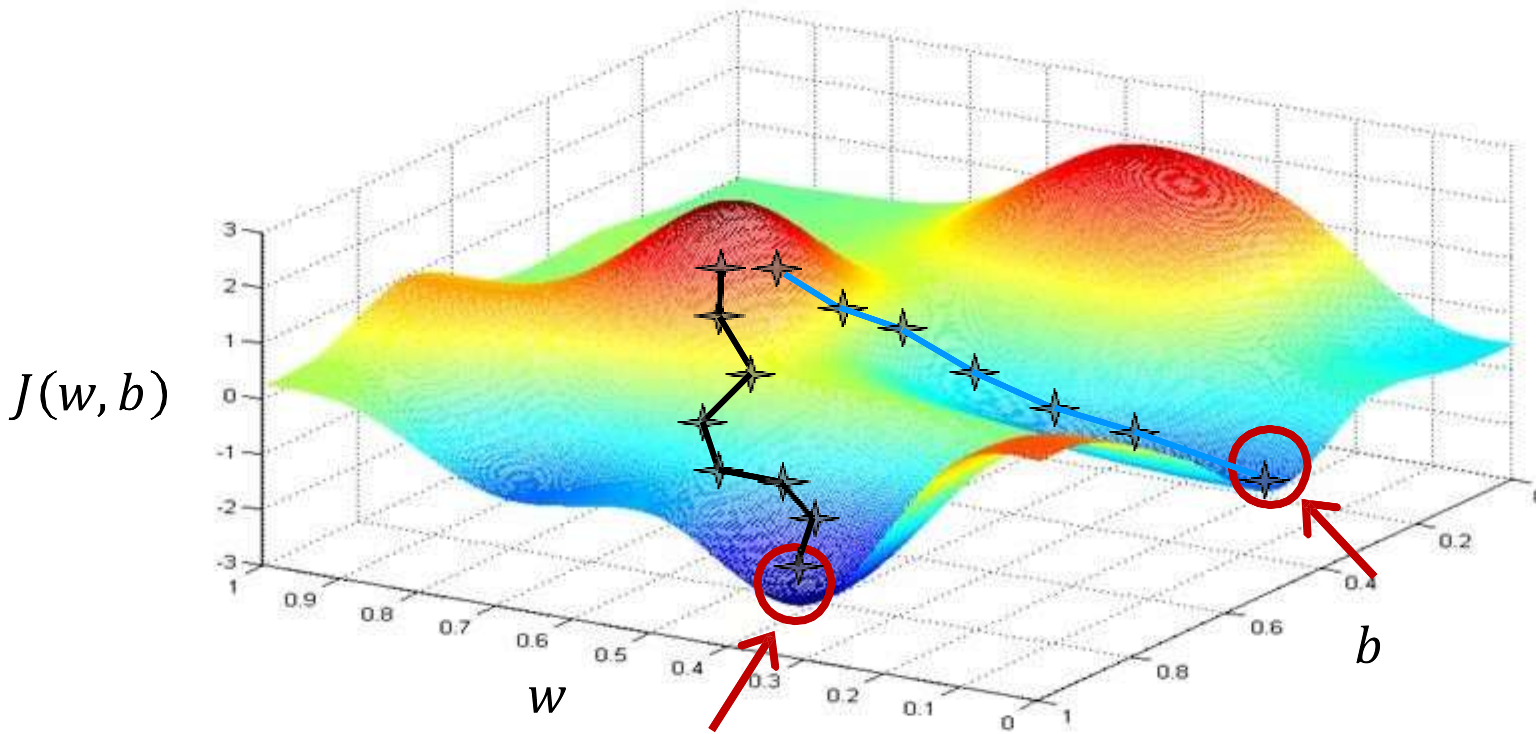
}

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

Update w and b simultaneously

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

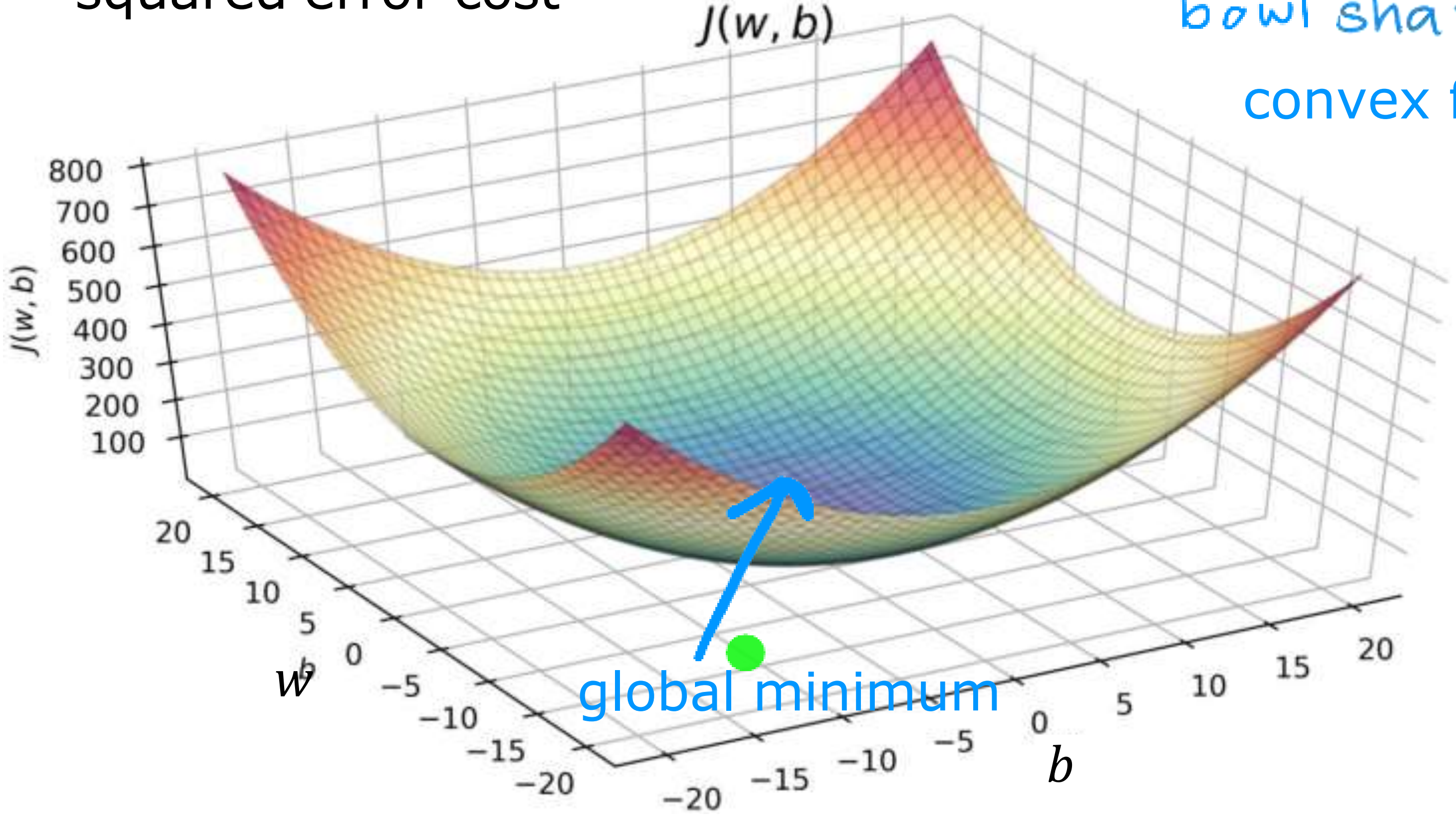
More than one local minimum



squared error cost

bowl shape

convex function

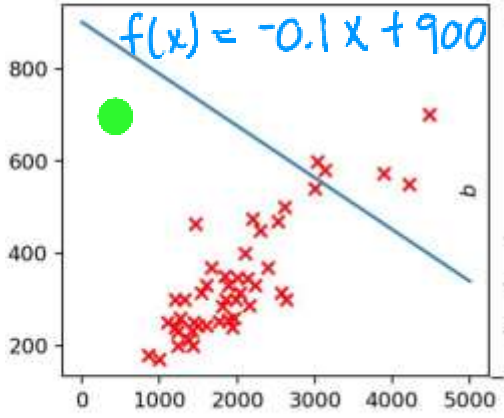


Training Linear Regression

Running Gradient Descent

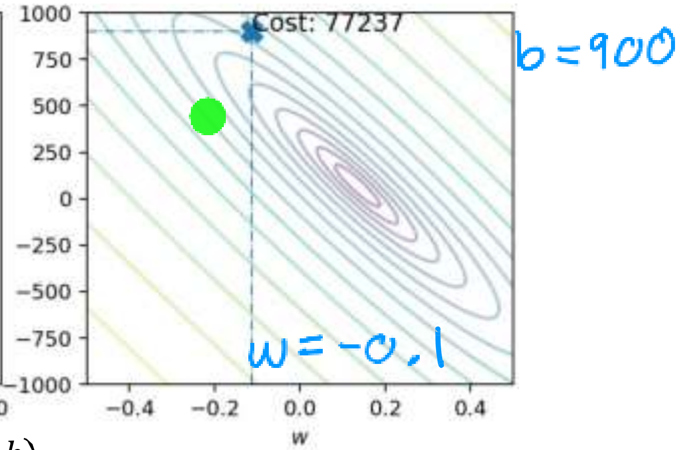
$f_{w,b}(size)$

price in $1000's

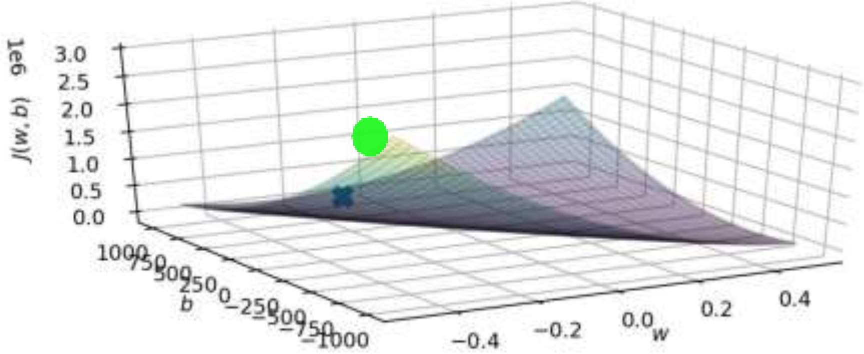


size in feet²

$J(w, b)$

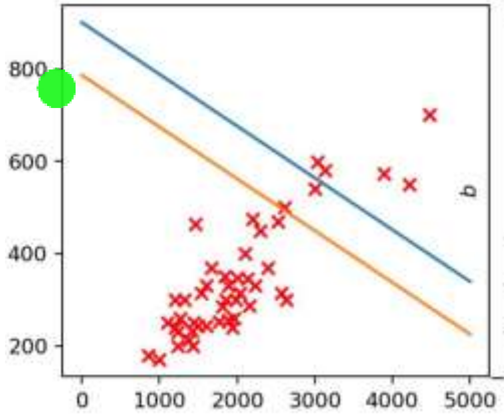


$J(w, b)$



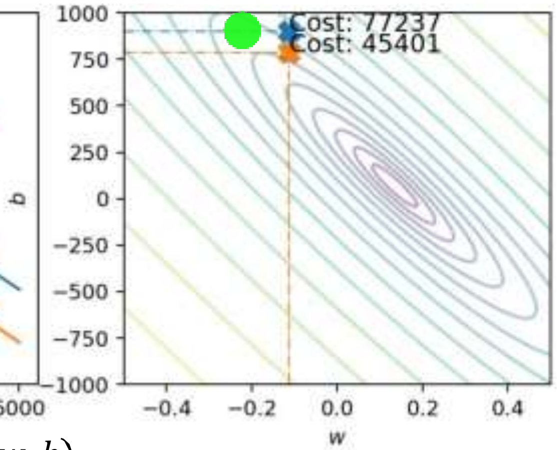
$f_{w,b}(size)$

price in $1000's

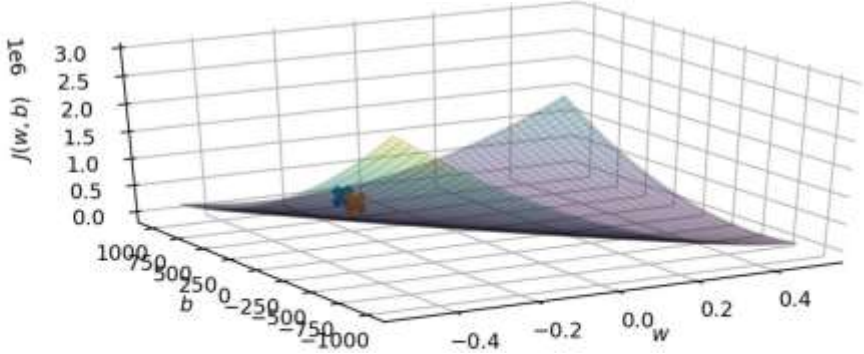


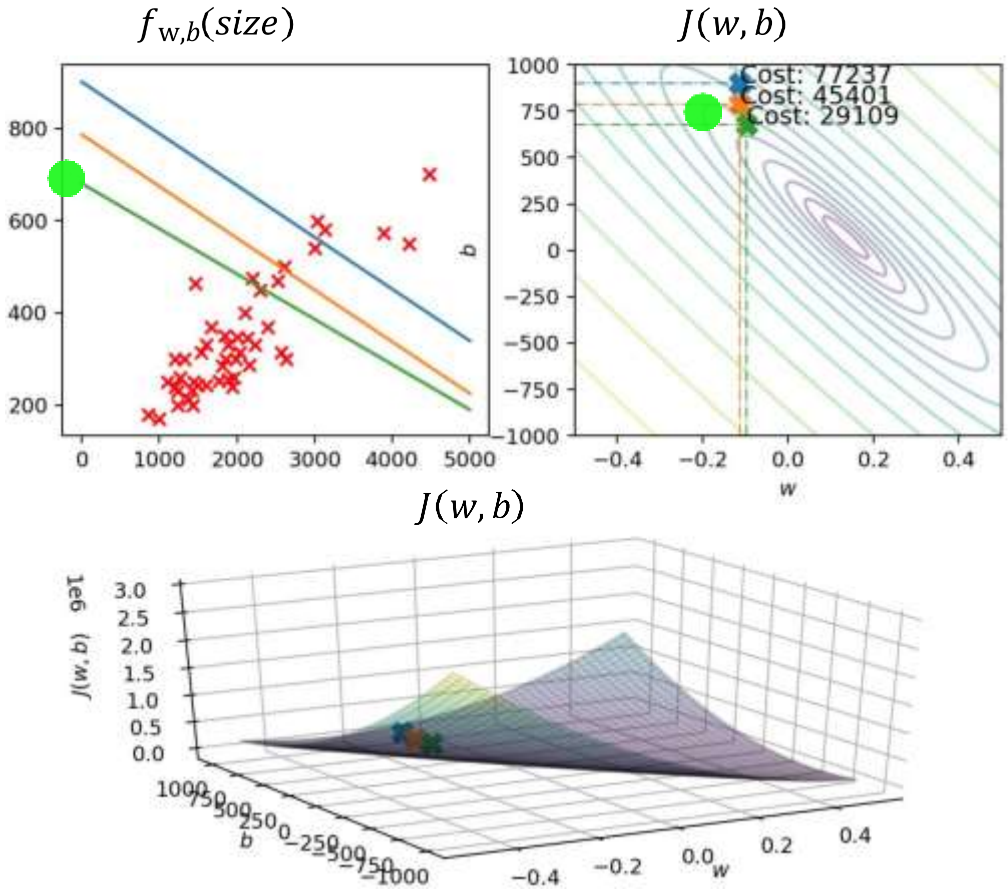
size in feet^2

$J(w,b)$

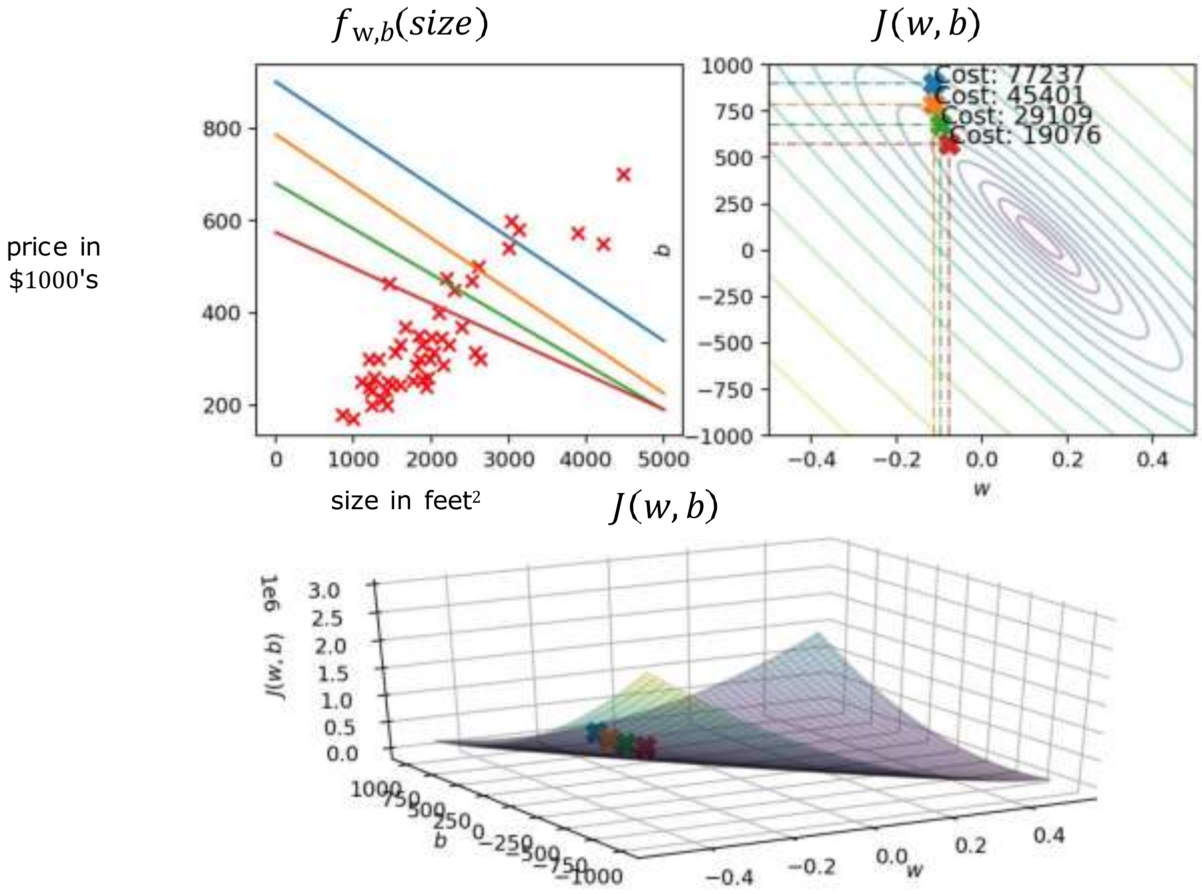


$J(w,b)$



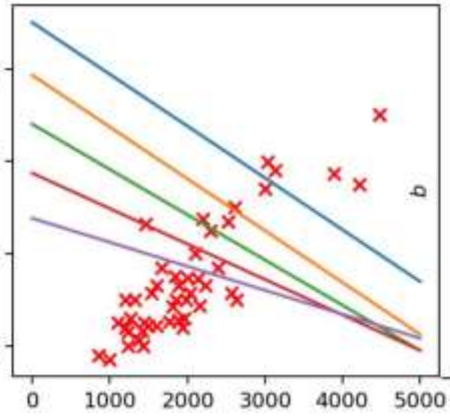


$f_{w,b}(size)$
$J(w,b)$
Cost: 77237
Cost: 45401
Cost: 29109
b
w
$J(w,b)$
1e6
J(w, b)
b
w


$f_{w,b}(size)$
$J(w,b)$
price in $1000's
size in feet²
Cost: 77237
Cost: 45401
Cost: 29109
Cost: 19076
b
w
$J(w,b)$
1e6
J(w, b)

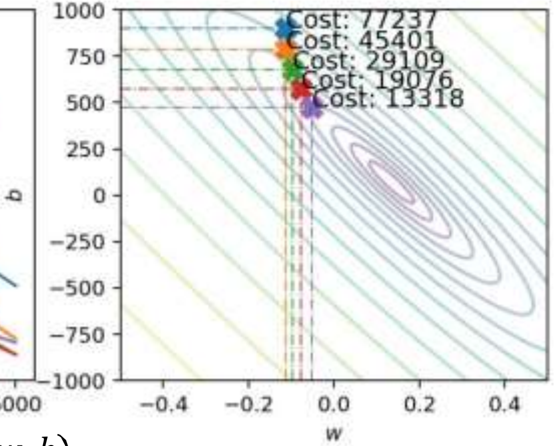
$f_{w,b}(size)$

price in $1000's

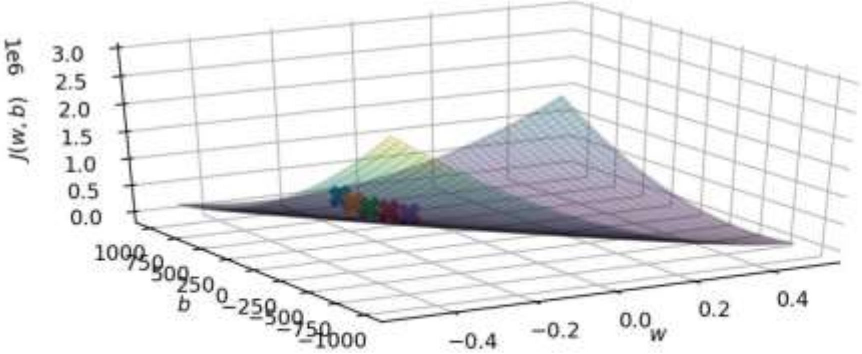


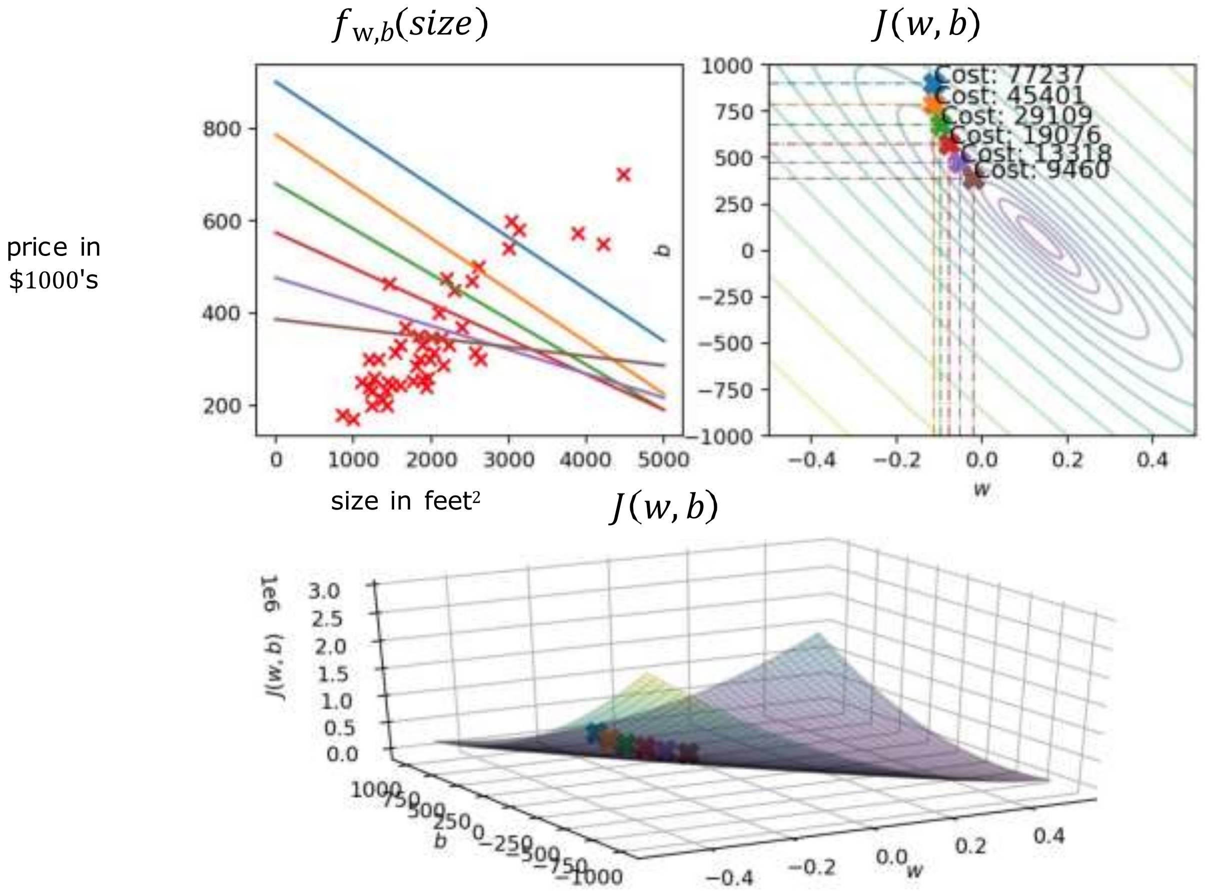
size in feet^2

$J(w,b)$



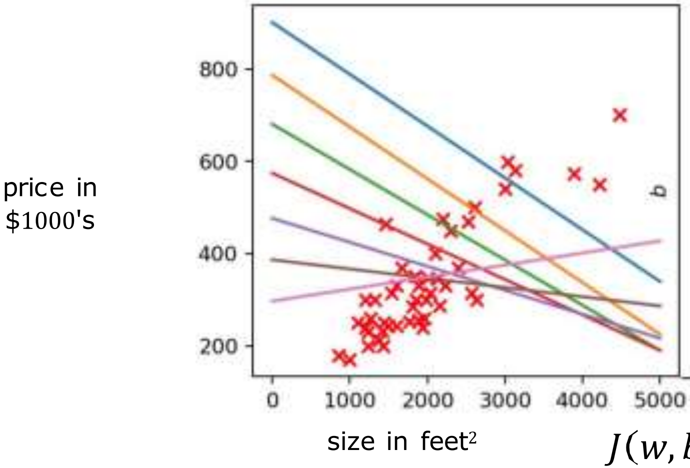
$J(w,b)$



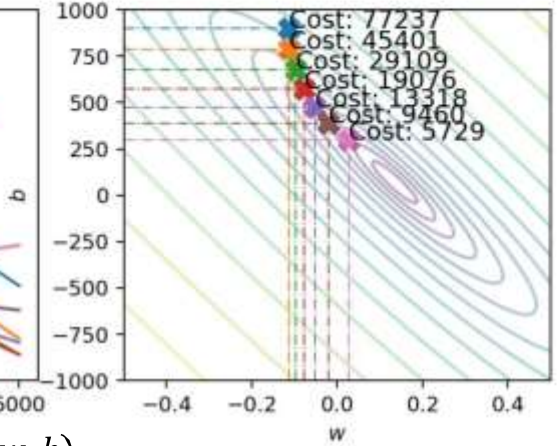


$f_{w,b}(size)$
$J(w,b)$
price in $1000's
size in feet^2
Cost: 77237
Cost: 45401
Cost: 29109
Cost: 19076
Cost: 13318
Cost: 9460
b
w
$J(w,b)$
1e6
J(w, b)

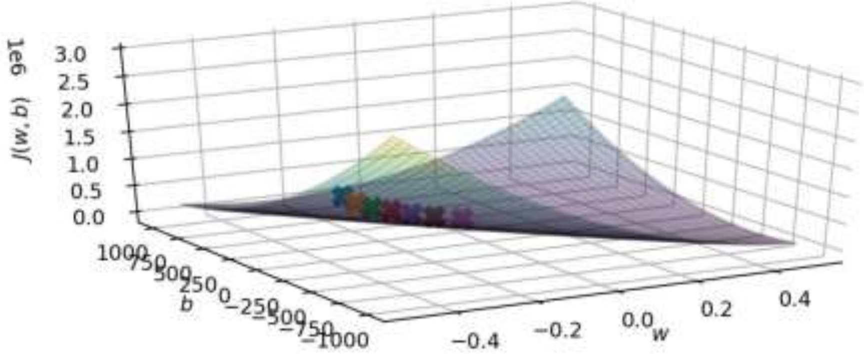
$f_{w,b}(size)$



$J(w,b)$

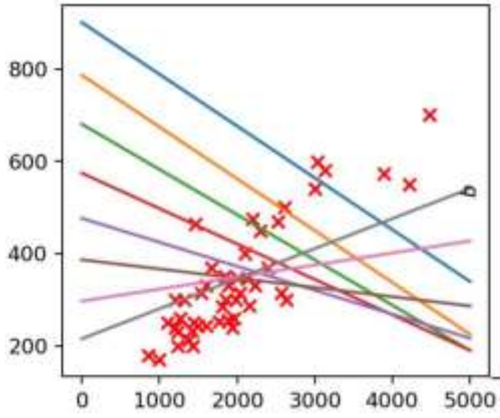


$J(w,b)$



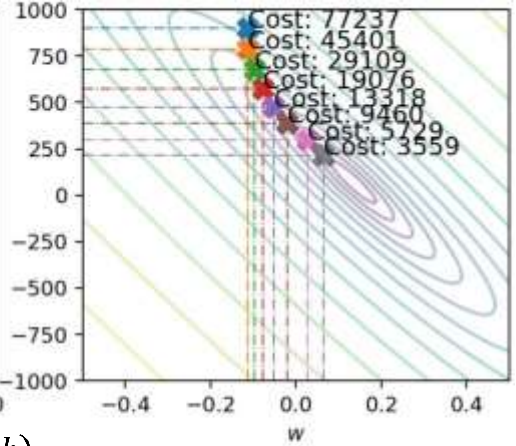
$f_{w,b}(size)$

price in
$1000's

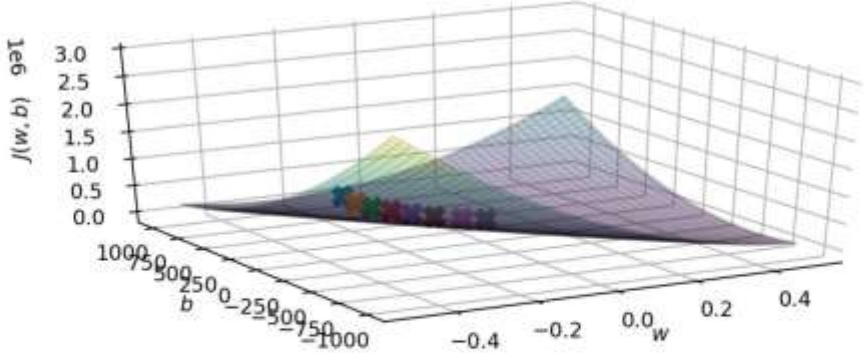


size in feet^2

$J(w,b)$



$J(w,b)$

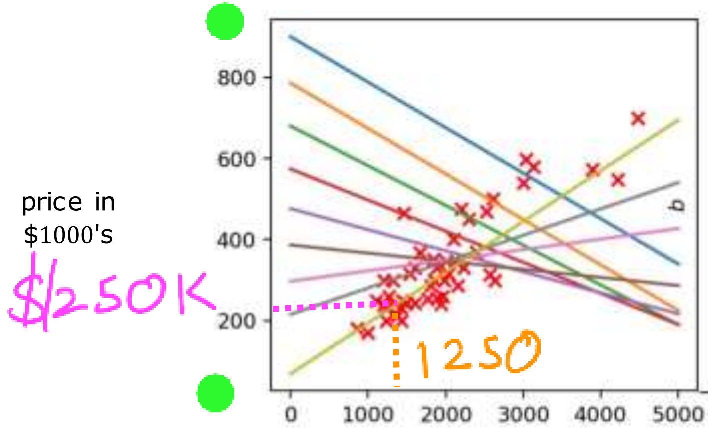


Stanford ONLINE DeepLearning.AI Andrew Ng

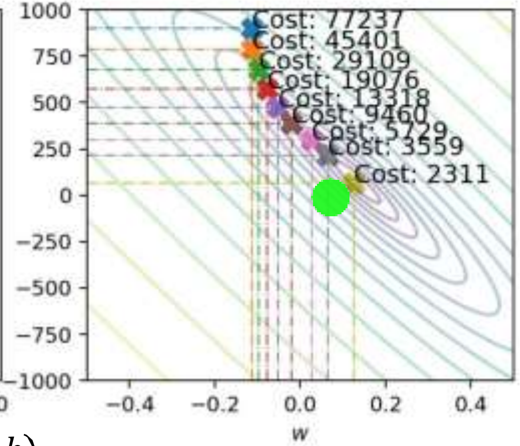
$f_{w,b}(size)$

$J(w,b)$

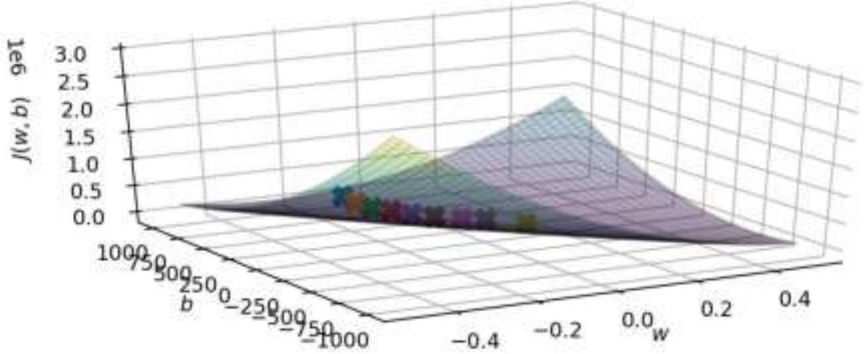
price in
$1000's



size in feet²



$J(w,b)$



“Batch” gradient descent

DeepLearning.AI
THE BATCH

other gradient descent: subsets

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

m = 47

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

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