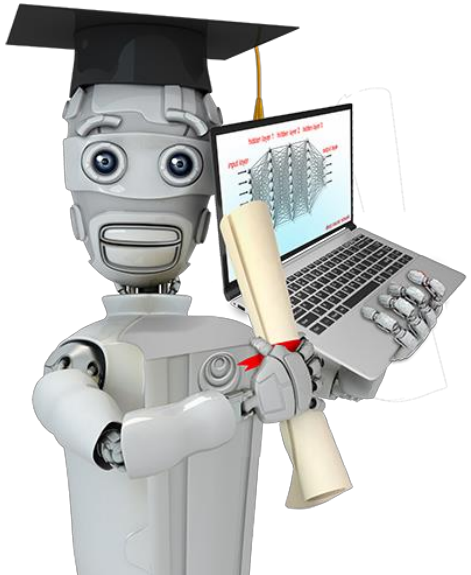
Copyright Notice

These slides are distributed under the Creative Commons License.

DeepLearning.AI makes these slides available for educational purposes. You may not use or distribute these slides for commercial purposes. You may make copies of these slides and use or distribute them for educational purposes as long as you cite DeepLearning.AI as the source of the slides.

For the rest of the details of the license, see https://creativecommons.org/licenses/by-sa/2.0/legalcode

Machine Learning Overview



Supervised Learning Part 1

Supervised learning

X

input

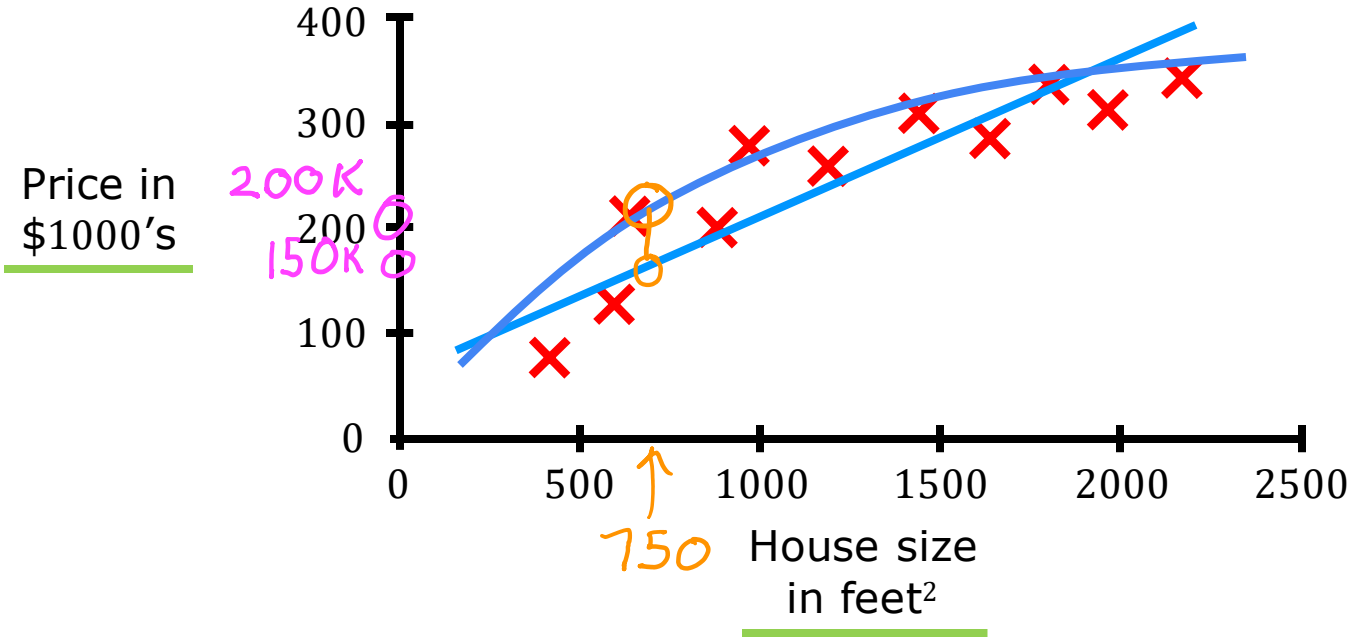
y

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

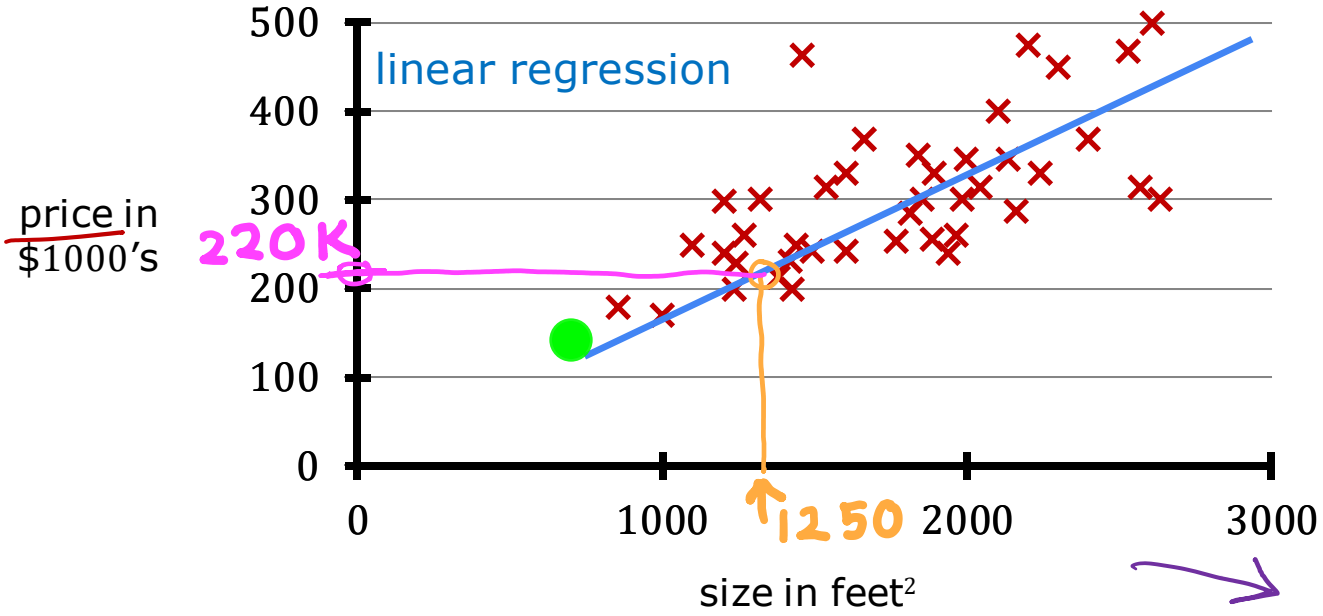
Regression: Housing price prediction



Linear Regression with One Variable

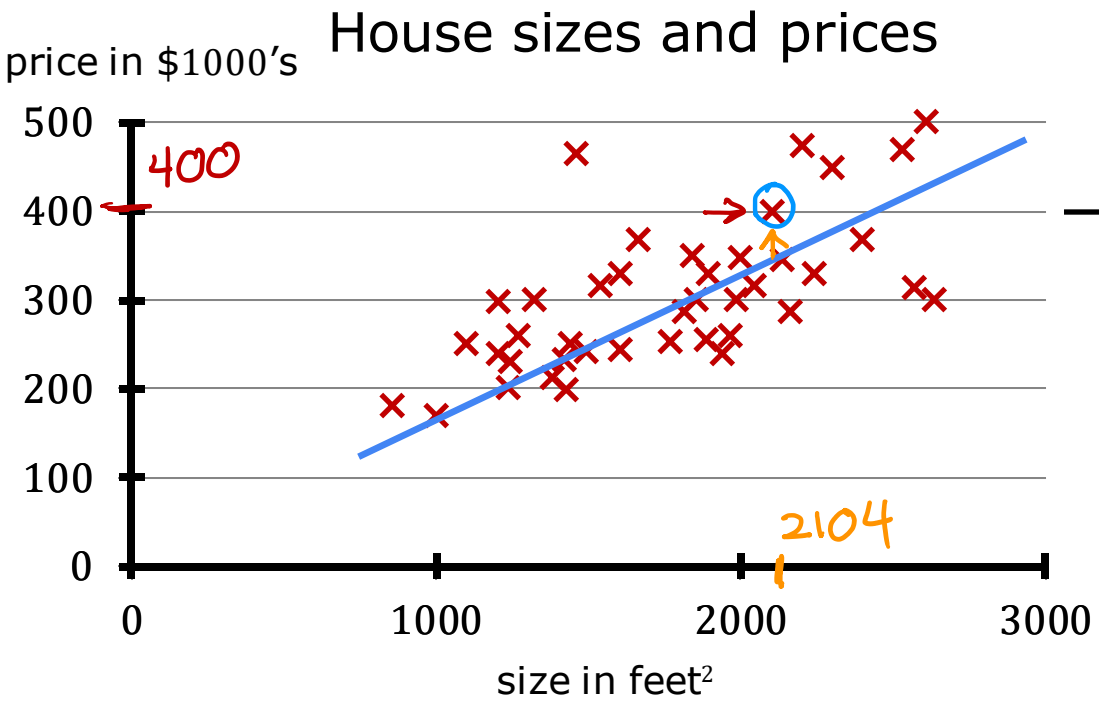
Linear Regression Model Part 1

House sizes and prices



Regression model
Predicts numbers
Infinitely many possible outputs

Supervised learning model
Data has "right answers"



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

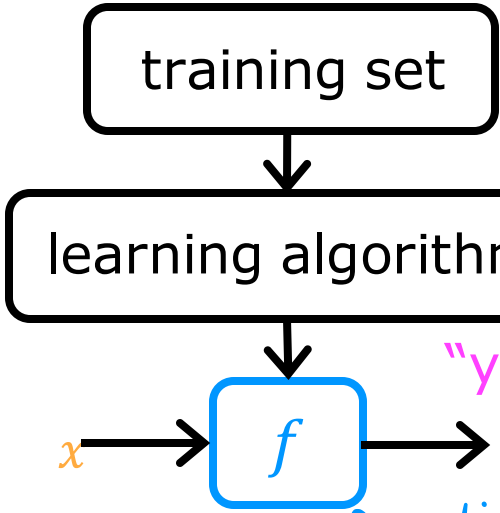
$(x^{(i)}, y^{(i)})$

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

index (1st, 2nd, 3rd …)

Linear Regression with One Variable

Linear Regression Model Part 2



features
targets

"y-hat"

function

feature

model

prediction

(estimated y)

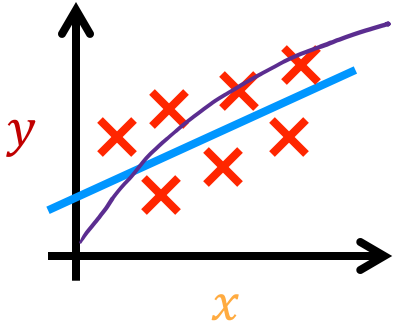
target

size → 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

Linear Regression with One Variable

Cost Function

Training set

features size in feet^2 (x)	targets price \$1000's (y)
2104	460
1416	232
1534	315
852	178
…	…

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

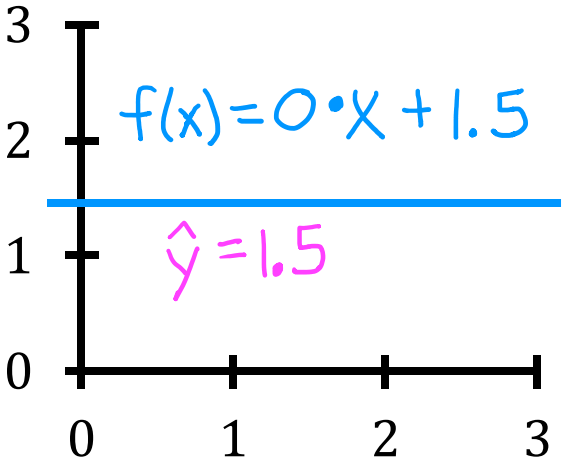
w, b: parameters

coefficients
weights

What do w, b do?

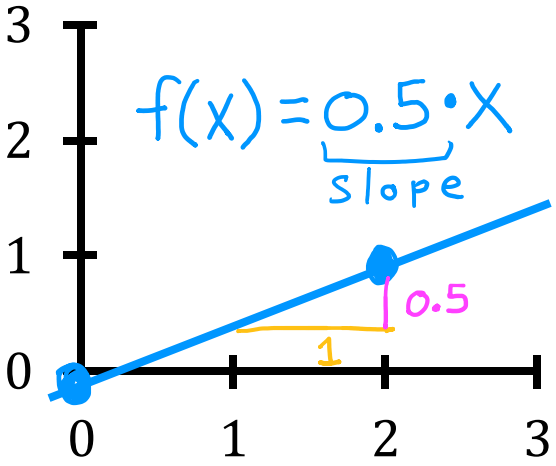
$$f_{w,b}(x) = wx + b$$

$f(x)$



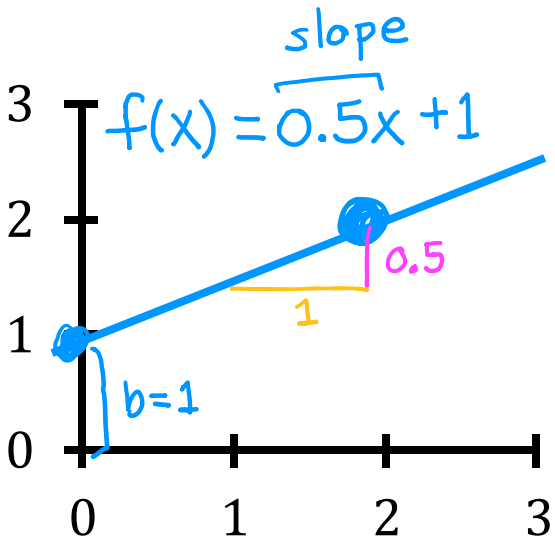
$w = 0$

$b = 1.5$



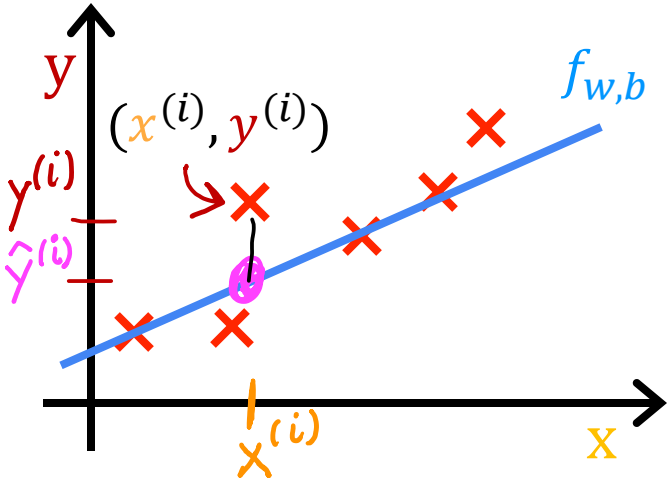
$w = 0.5$

$\mathrm{b} = 0$



$w = 0.5$

$b = 1$



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

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

Maliyet fonksiyonu

Kare hatası maliyet fonksiyonu

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

error

m = eğitim örnekleri sayısı

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

Find w, b:

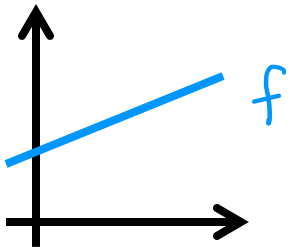
$\hat{y}^{(i)}$ is close to $y^{(i)}$ for all $(x^{(i)}, y^{(i)})$.

Linear Regression with One Variable

Cost Function Intuition

model:

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



parameters:

$$w, b$$

cost function:

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

goal:

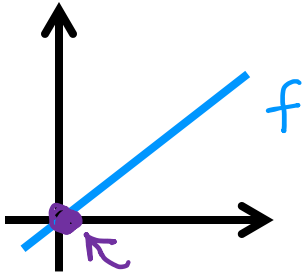
$$\underset{w,b}{\text{minimize}}\, J(w,b)$$

simplified

$$f_w(x) = wx$$

$b = \emptyset$

$$w$$



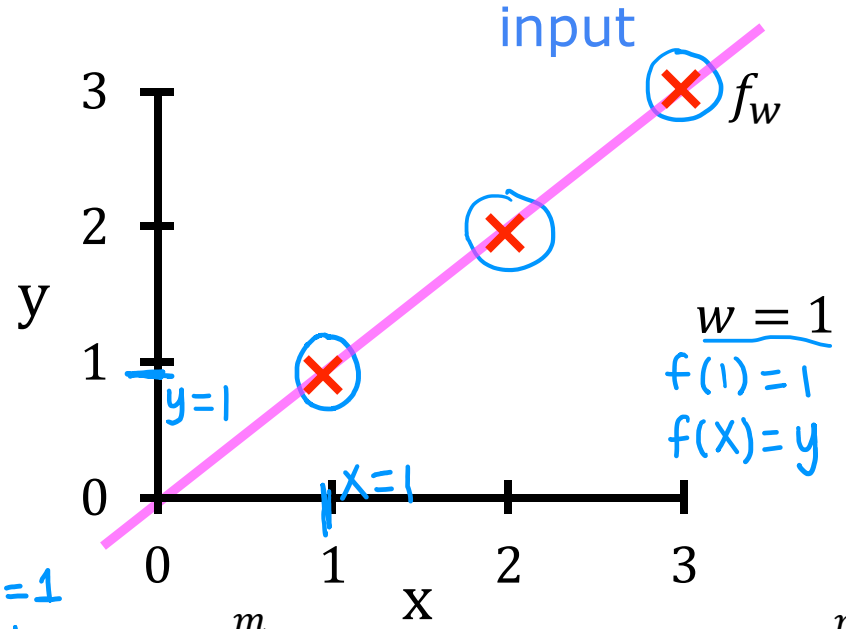
$$J(w) = \frac{1}{2m}\sum_{i=1}^{m}\left(\underline{f_w\left(x^{(i)}\right)} - y^{(i)}\right)^2$$

$wx^{(i)}$

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

$f_w(x)$

(for fixed w, function of x)



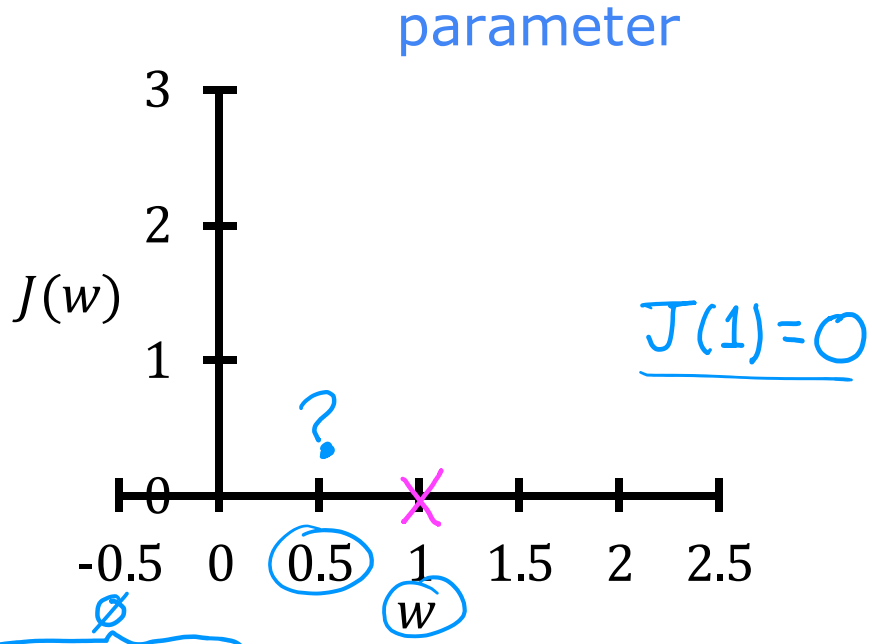
$w = 1$

$f(1) = 1$

$f(x) = y$

$J(w)$

(function of w)



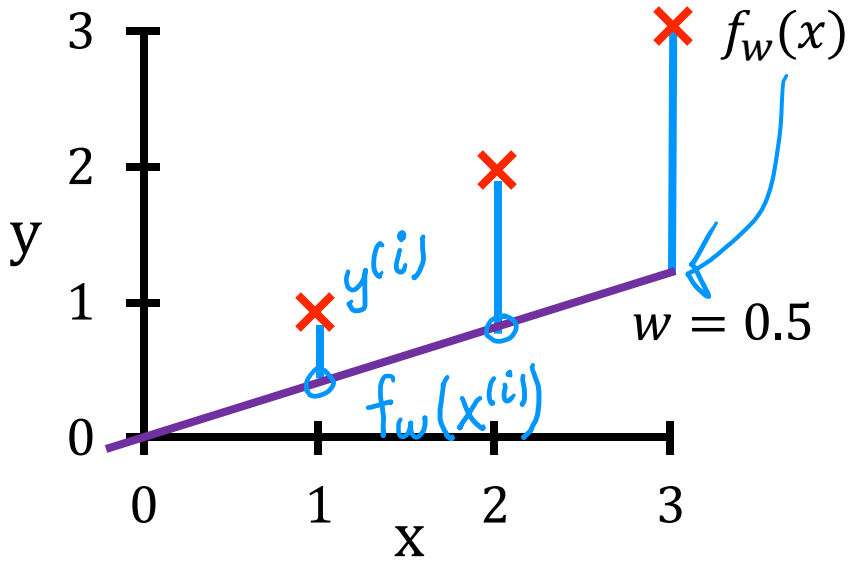
$J(1) = 0$

$w = 1$

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

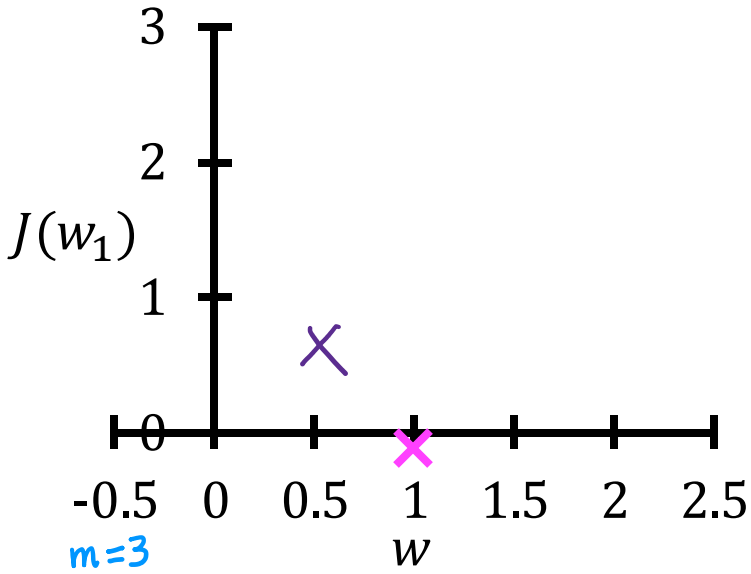
$f_w(x)$

(function of x)



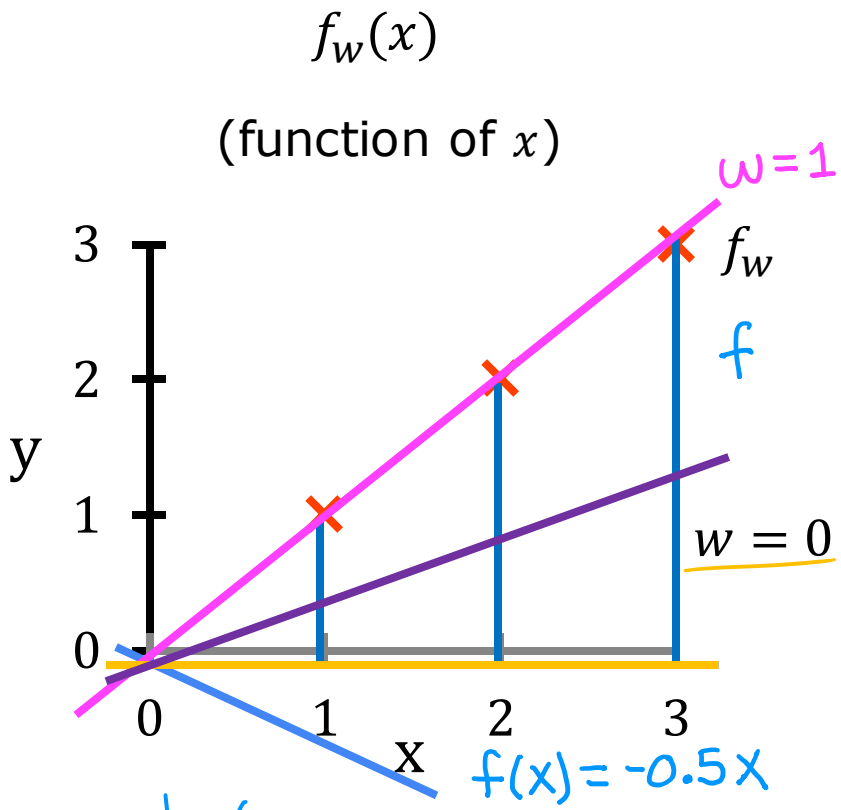
$J(w)$

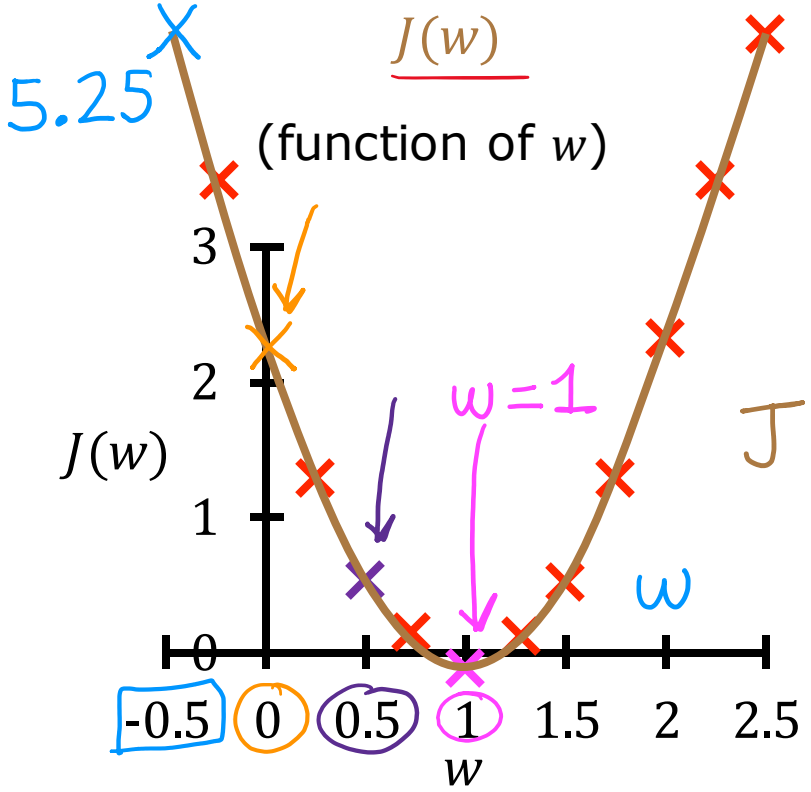
(function of w)



$m=3$

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



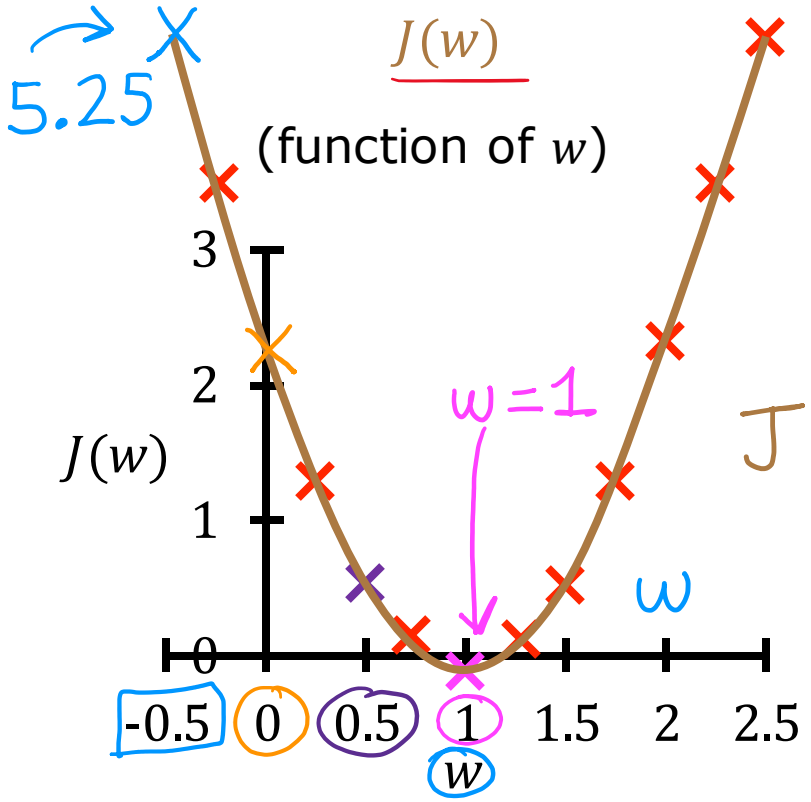


$$J(0) = \frac{1}{2m}\left(1^2 + 2^2 + 3^2\right) = \frac{1}{6}[14] \approx 2.3$$

how to choose w?

goal of linear regression:

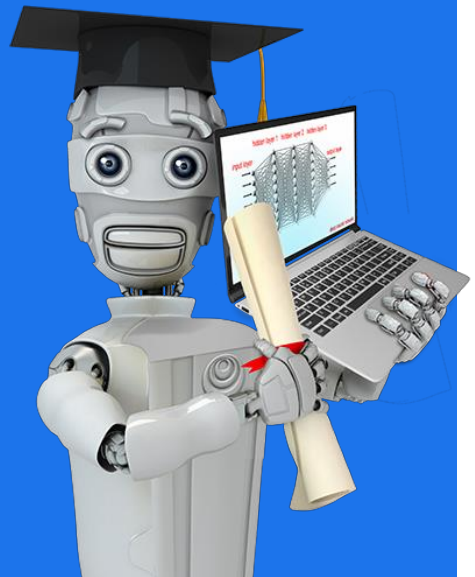
$$\underset{w}{\text{minimize}}\, J(w)$$



choose w to minimize J(w)

Stanford
ONLINE

DeepLearning.AI



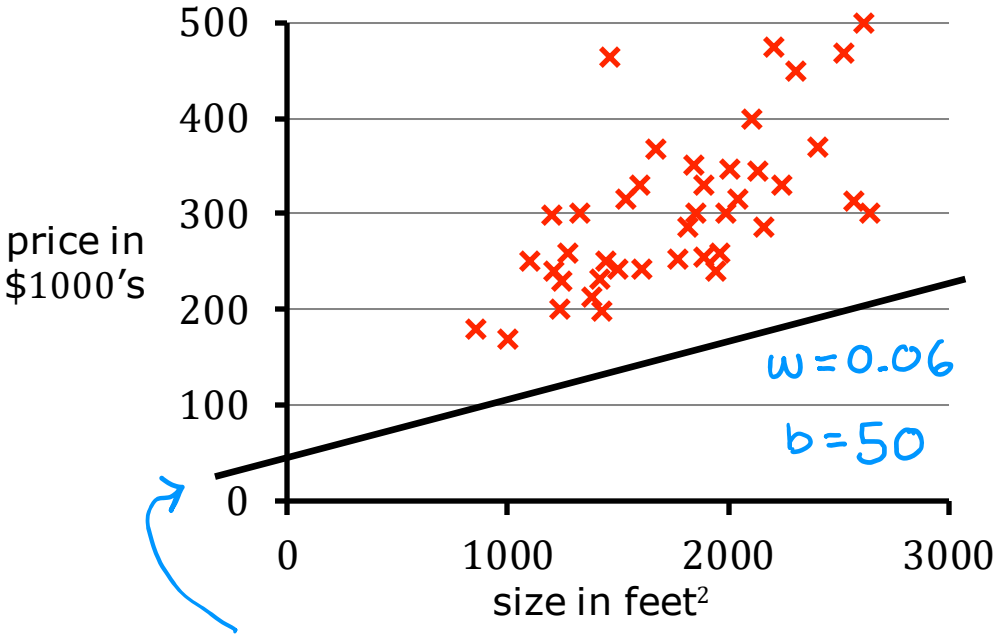
Linear Regression with One Variable

Visualizing the Cost Function

Model	$f_{w,b}(x) = wx + b$
Parameters	w, b
Cost Function	$J(w,b) = \frac{1}{2m}\sum_{i=1}^{m}(f_{w,b}(x^{(i)}) - y^{(i)})^2$
Objective	$\underset{w,b}{\text{minimize}}\, J(w,b)$

$f_{w,b}$

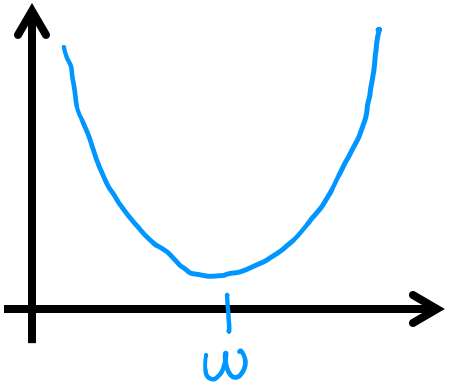
(function of x)



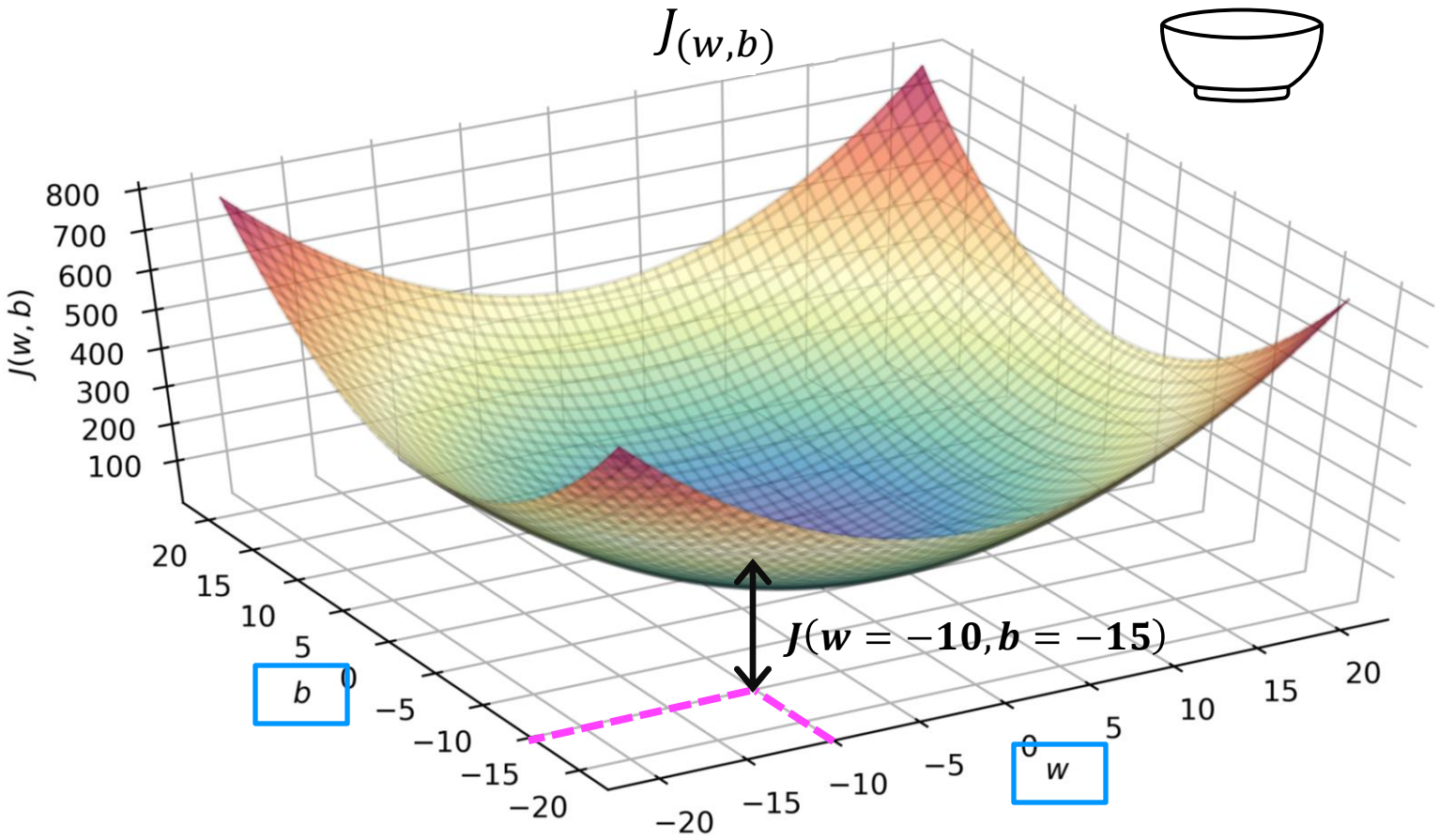
$$f_{w,b}(x) = 0.06x + 50$$

J

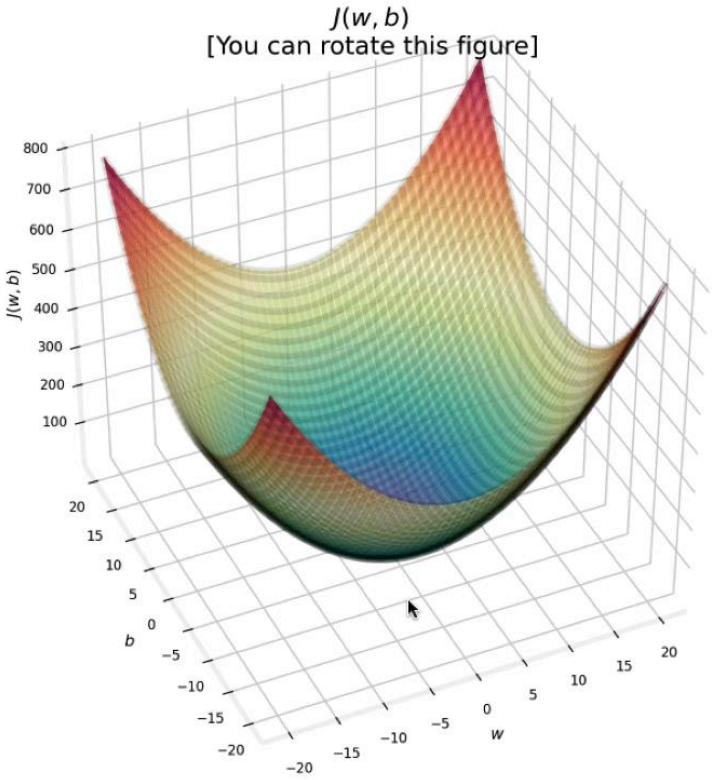
(function of w, b)



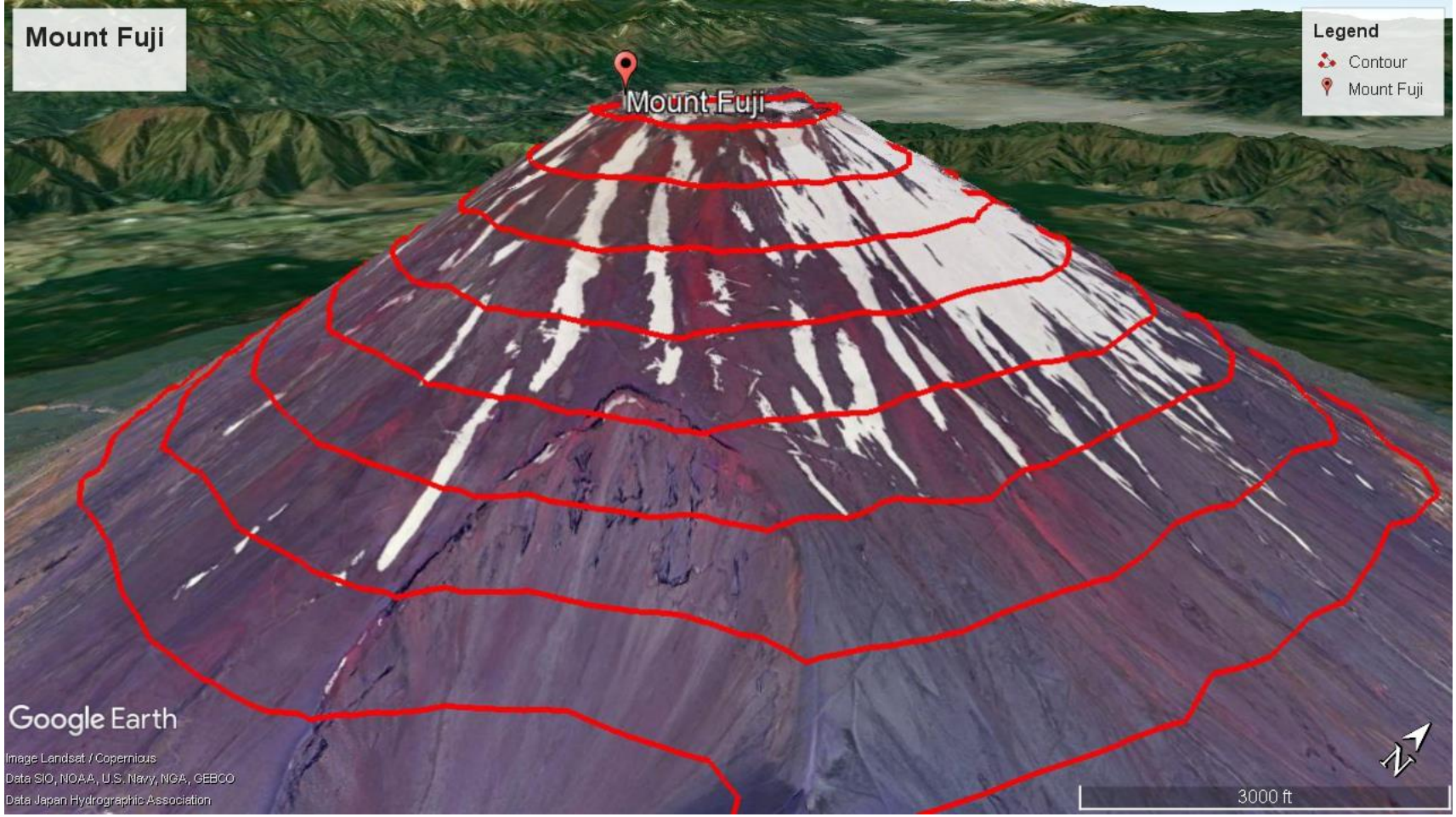
$\boxed{w, b}$

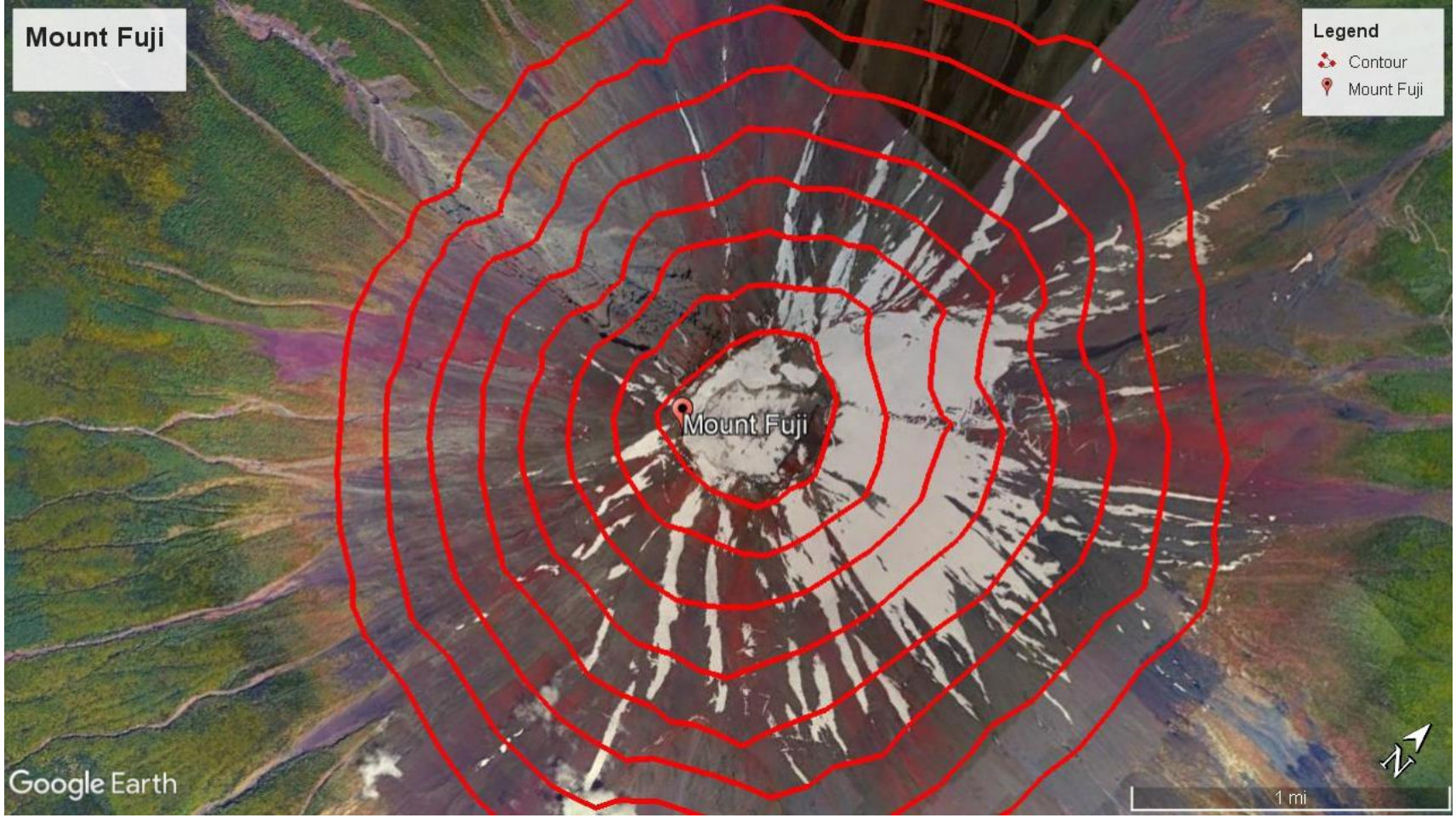

$J_{(w,b)}$
$J(w,b)$
800
700
600
500
400
300
200
100
20
15
10
5
0
−5
−10
−15
−20
b
−20
−15
−10
−5
0
5
10
15
20
w
$J(w=-10, b=-15)$

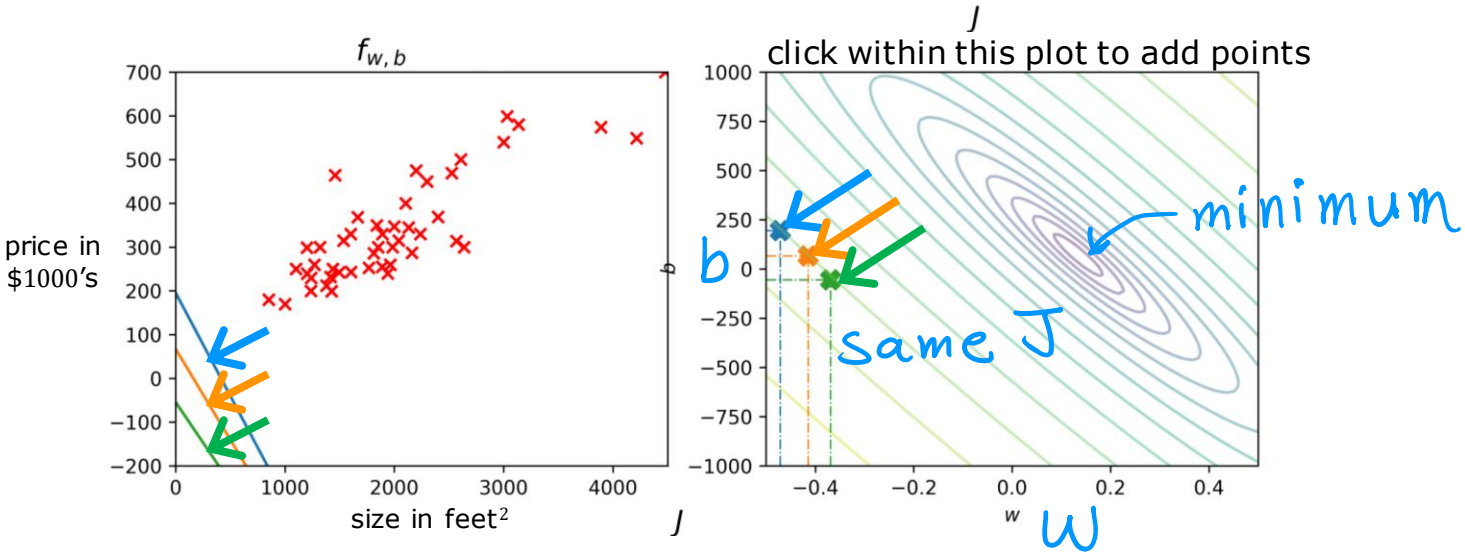
3D surface plot

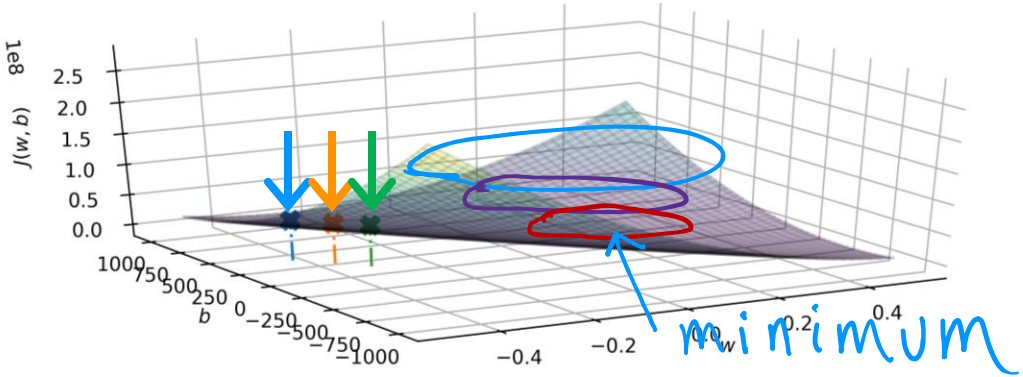


Alternative contour plot


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

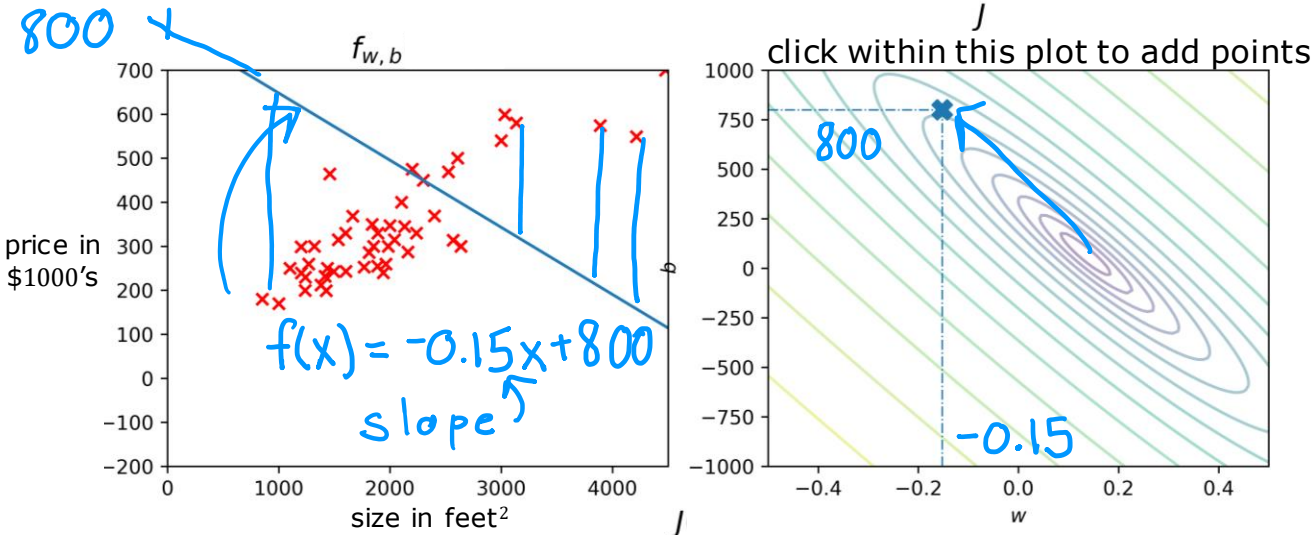

Mount Fuji
Legend
Contour
Mount Fuji
Mount Fuji
Google Earth
1 mi
N

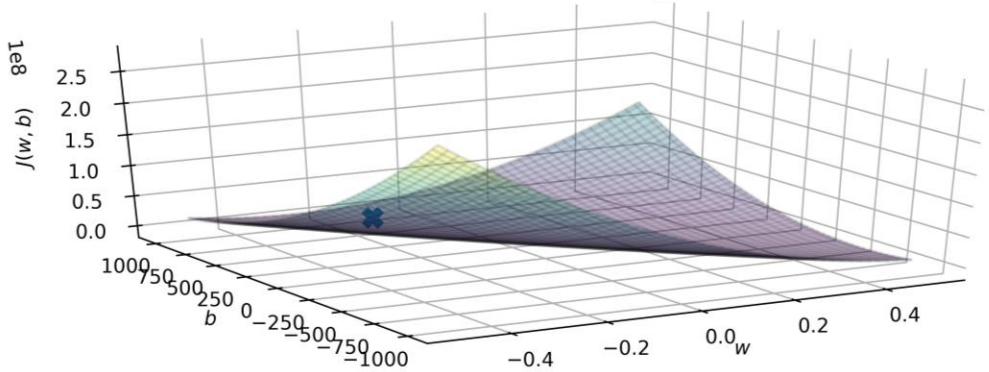

$f_{w,b}$
price in $1000's
size in feet^2
J
click within this plot to add points
b
w
minimum
same J
w

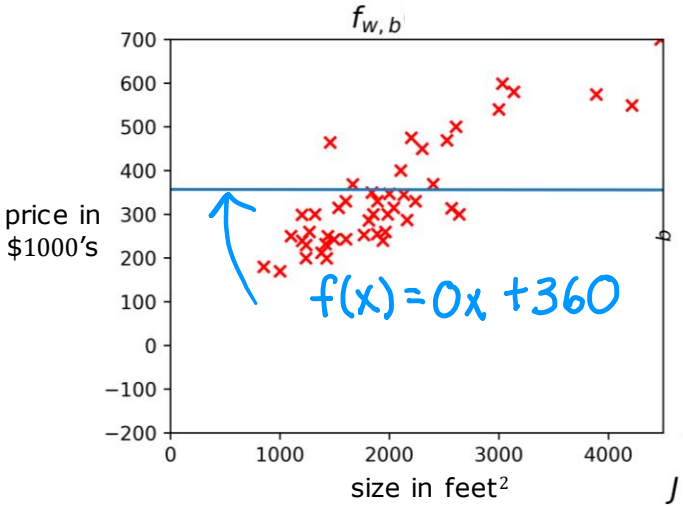

1e8
J(w, b)
b
w
minimum

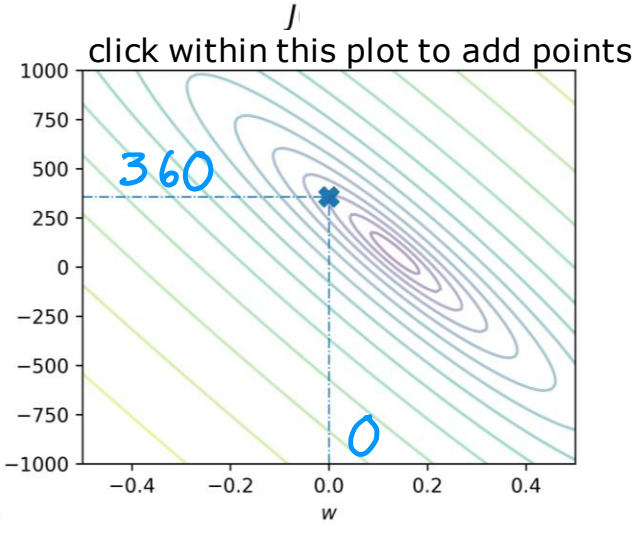
Linear Regression with One Variable

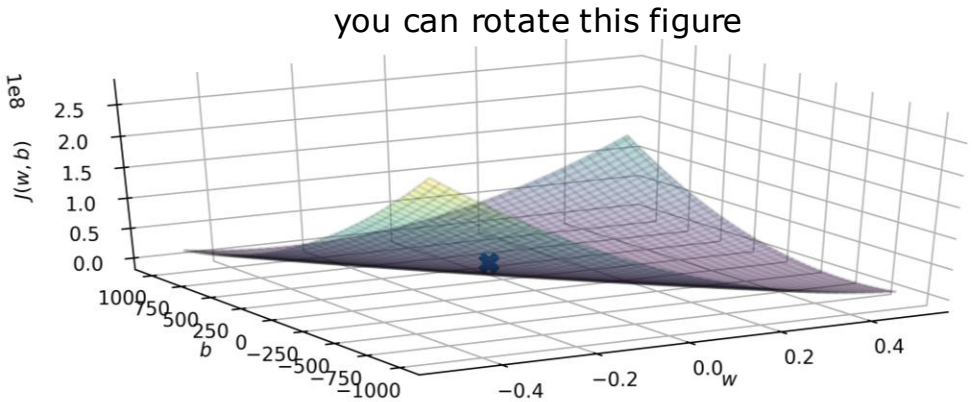
Visualization examples

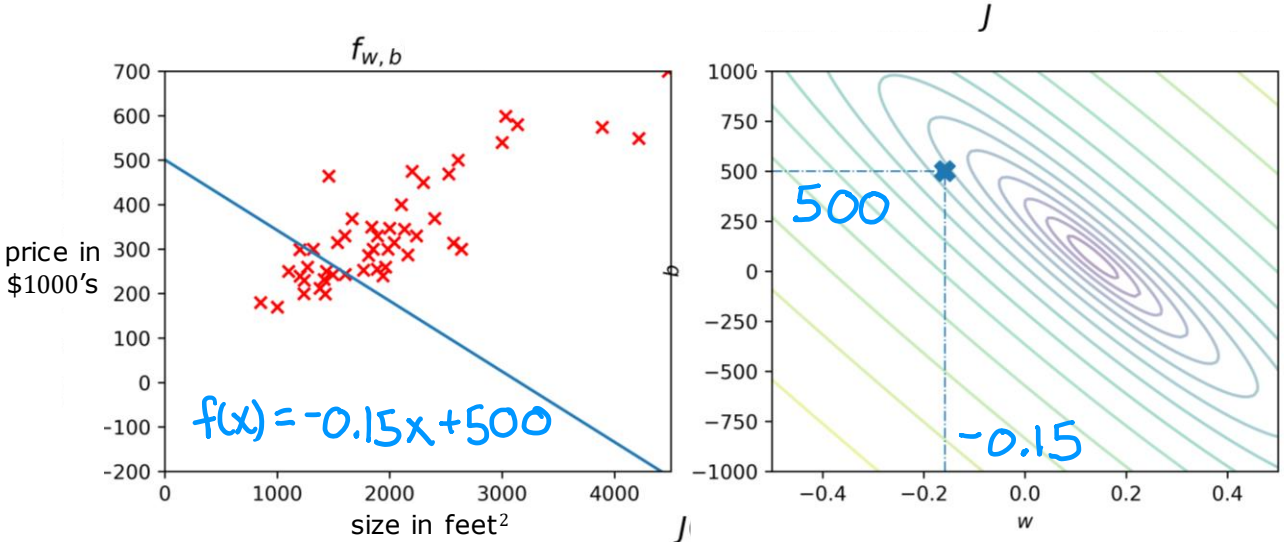

800
$f_{w,b}$
price in $1000's
size in feet^2
$f(x) = -0.15x + 800$
slope
J
click within this plot to add points
800
-0.15
b
w

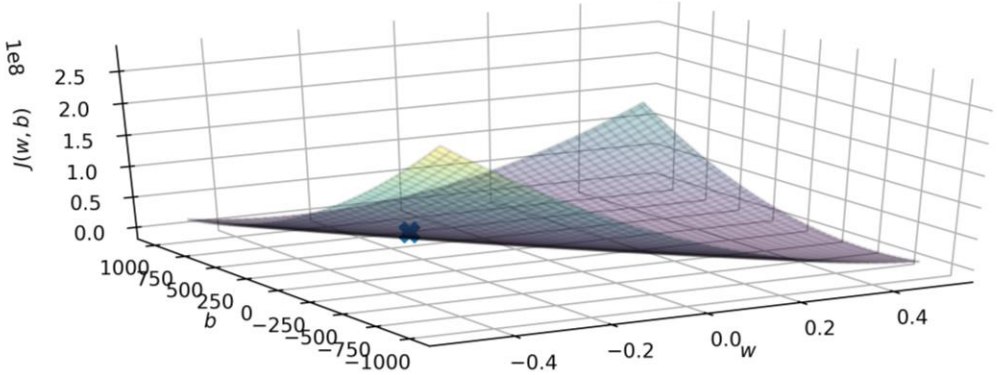

1e8
$J(w,b)$
b
w

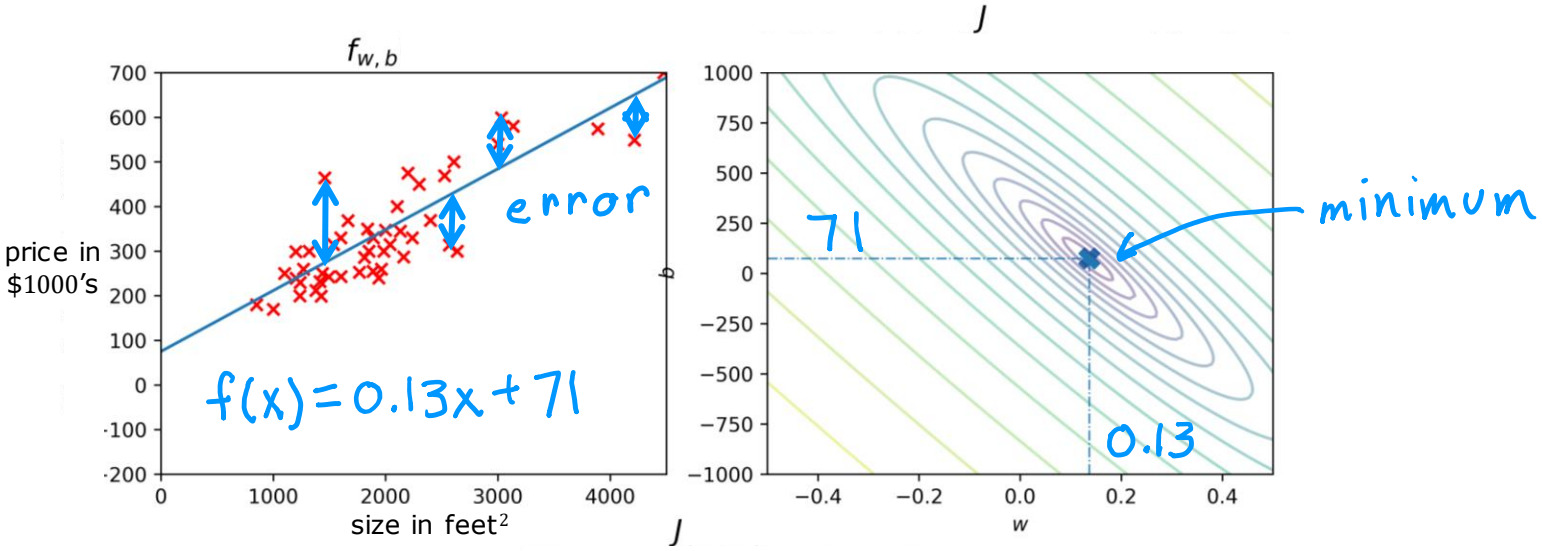

$f_{w,b}$
price in $1000's
size in feet^2
$f(x) = 0x + 360$

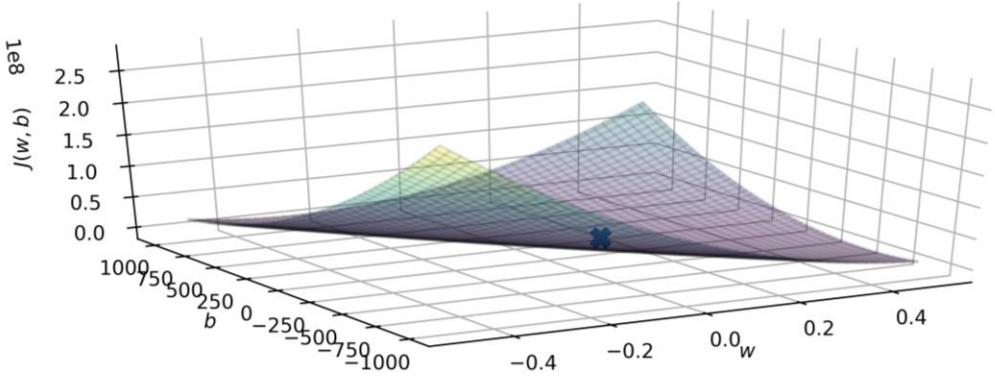

J
click within this plot to add points
360
0
w
b


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
500
-0.15
w
b


1e8
$J(w, b)$
b
w


$f_{w,b}$
price in $1000's
size in feet^2
error
$f(x) = 0.13x + 71$
J
71
0.13
minimum
b
w


1e8
J(w, b)
b
w

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 $w=0$, $b=0$)

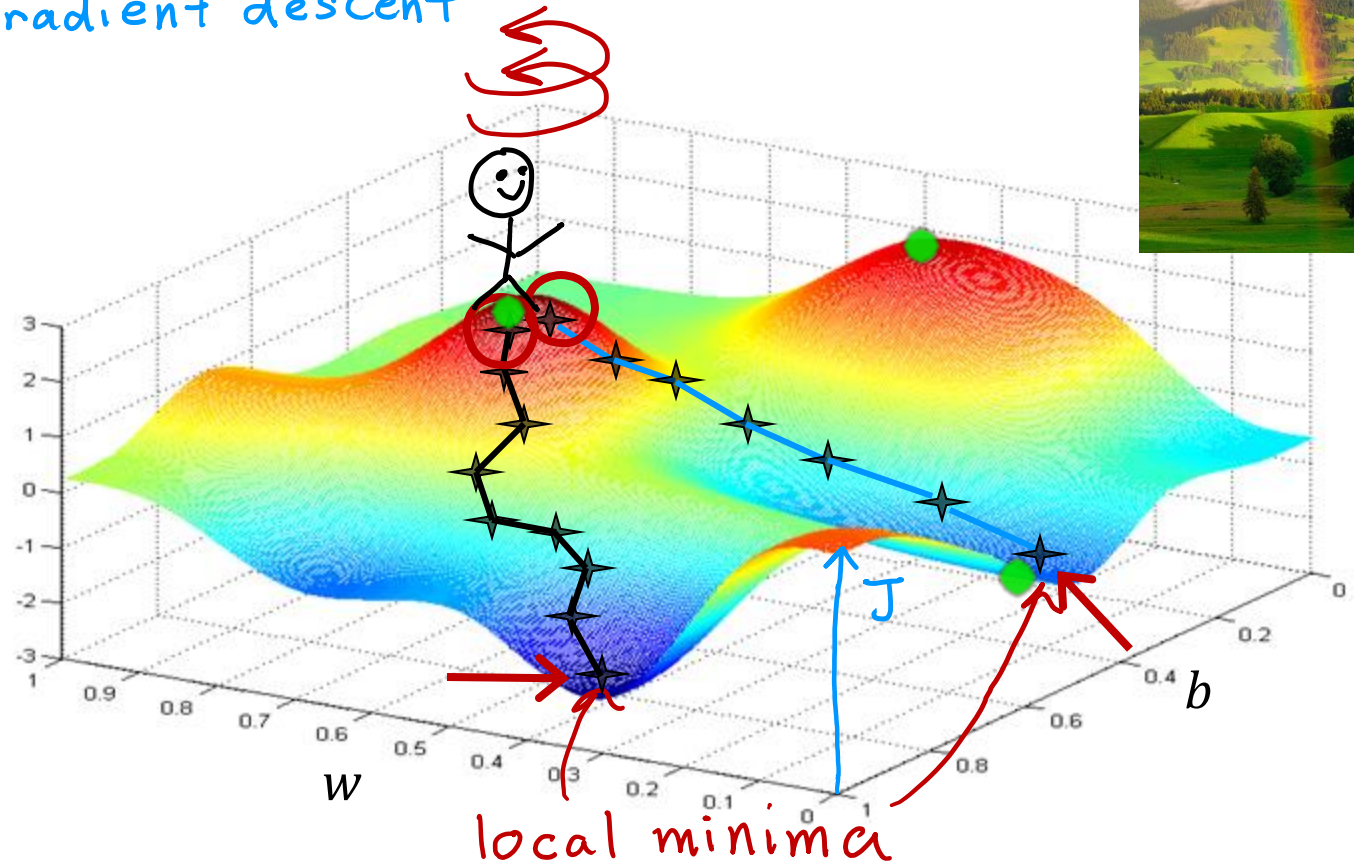
Keep changing w, b to reduce $J(w, b)$

Until we settle at or near a minimum

may have >1 minimum

gradient descent

$J(w, b)$



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

$a = c$

$a = a + 1$

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

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

$$w = tmp_w$$

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

$$b = tmp_b$$

Training Linear Regression

Gradient Descent Intuition

Gradient descent algorithm

repeat until convergence {

learning rate

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

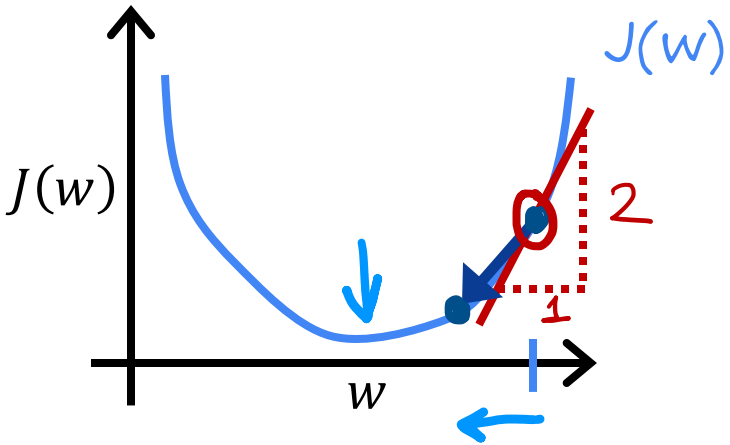
derivative

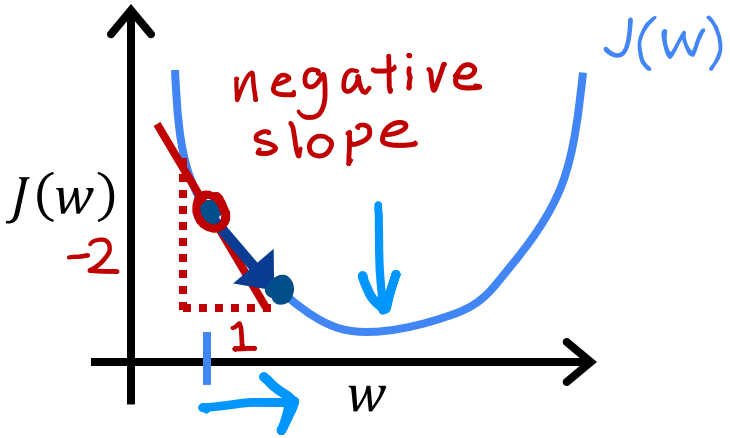
$$\underline{b} = b - \alpha \frac{\partial}{\partial b} J(w,b)$$

$$J(w)$$

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

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





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

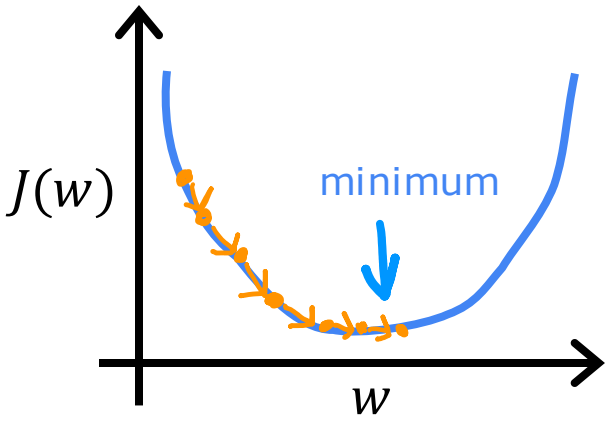
Training Linear Regression

Learning Rate

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

If α is too small…

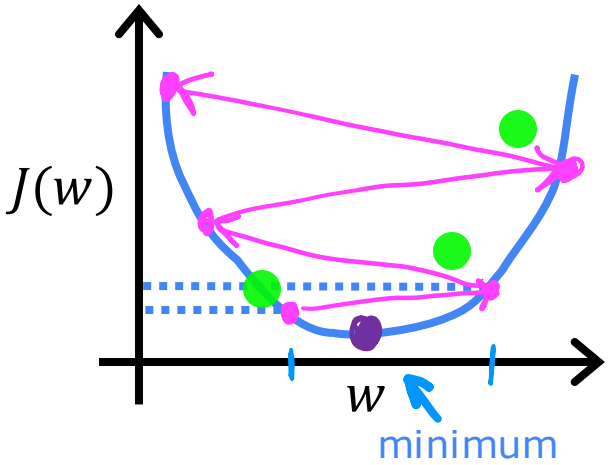
Gradient descent may be slow.

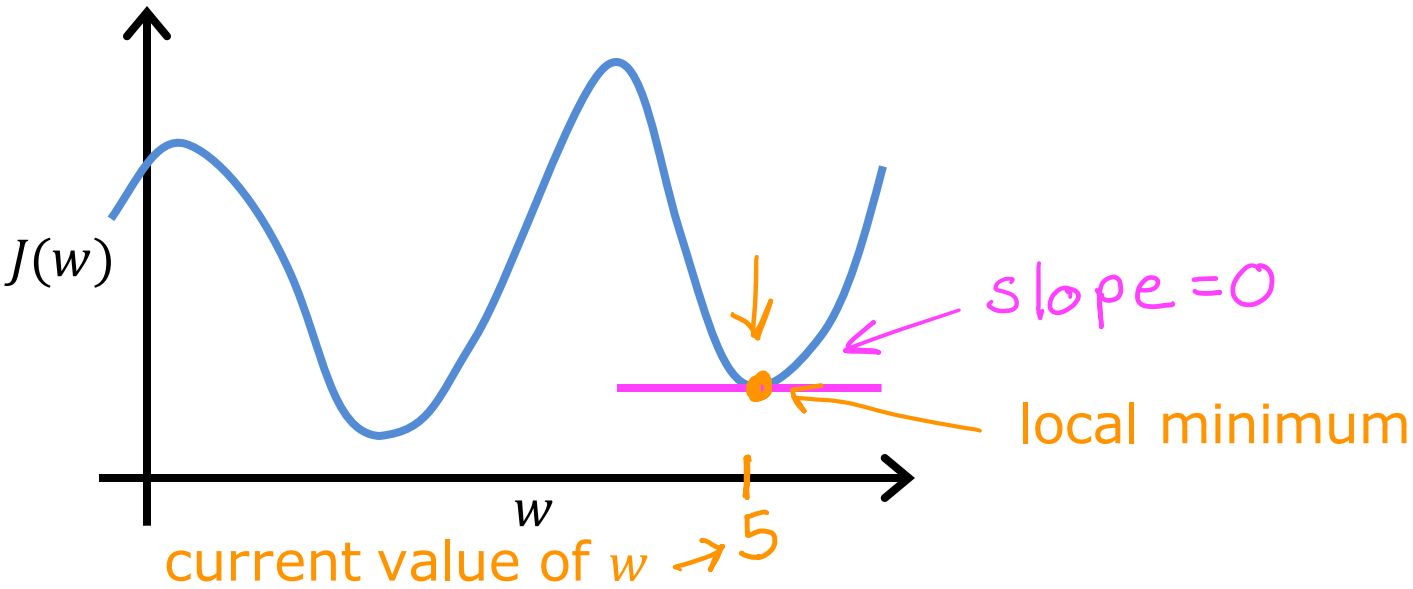


If α is too large…

Gradient descent may:

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





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

$$w = w - \alpha \cdot 0$$

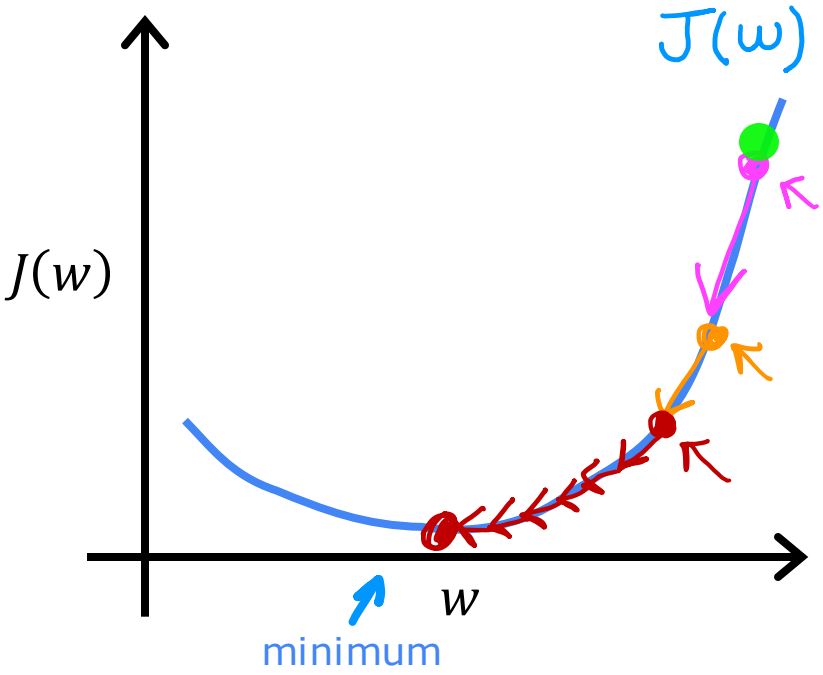
$$w = w$$

Can reach local minimum with fixed learning rate

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

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

Can reach minimum without decreasing learning rate α



Training Linear Regression

Gradient Descent for Linear Regression

Linear regression model

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

Cost function

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

Gradient descent algorithm

repeat until convergence {

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

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

}

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

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

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

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

Gradient descent algorithm

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

repeat until convergence {

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

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

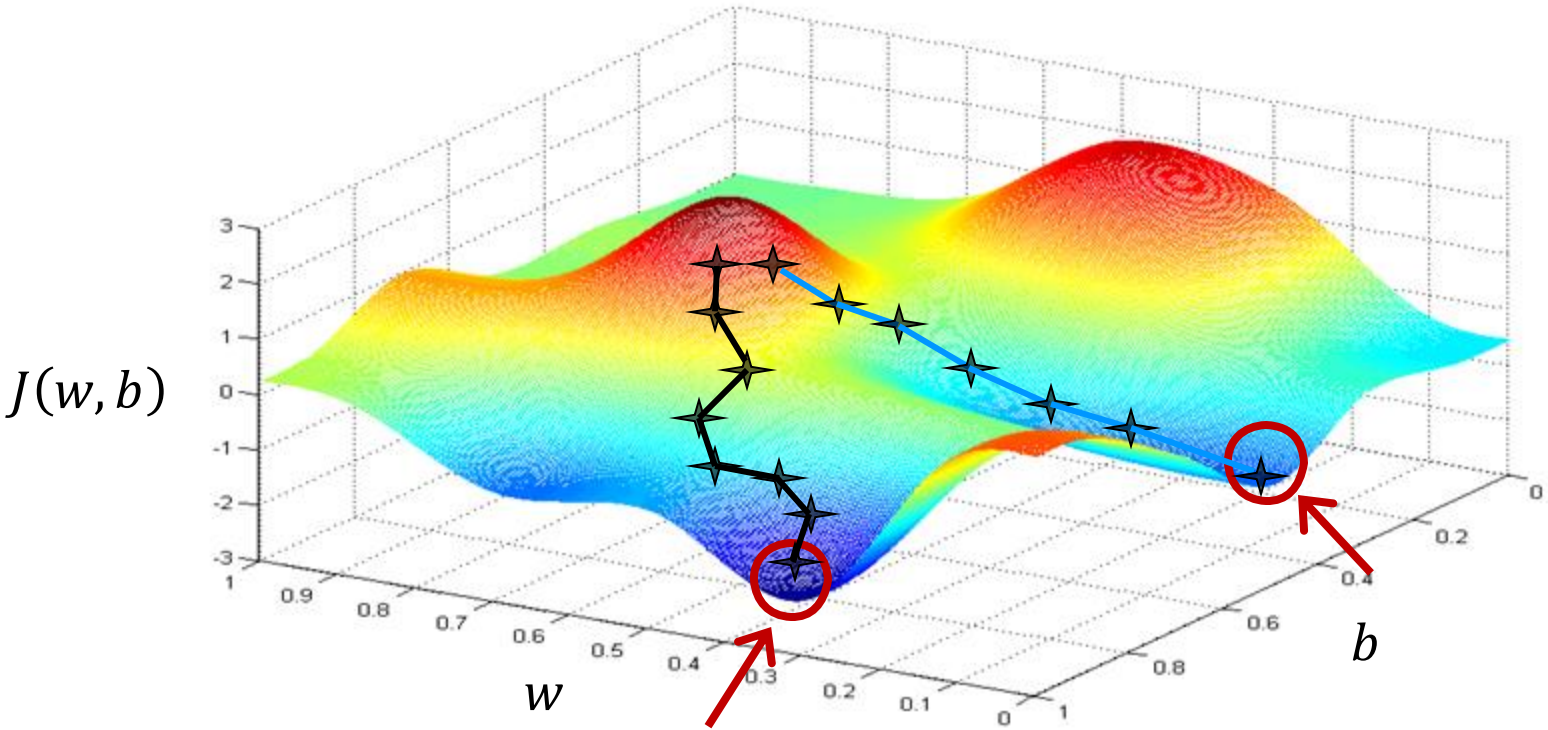
}

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

Update w and b simultaneously

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

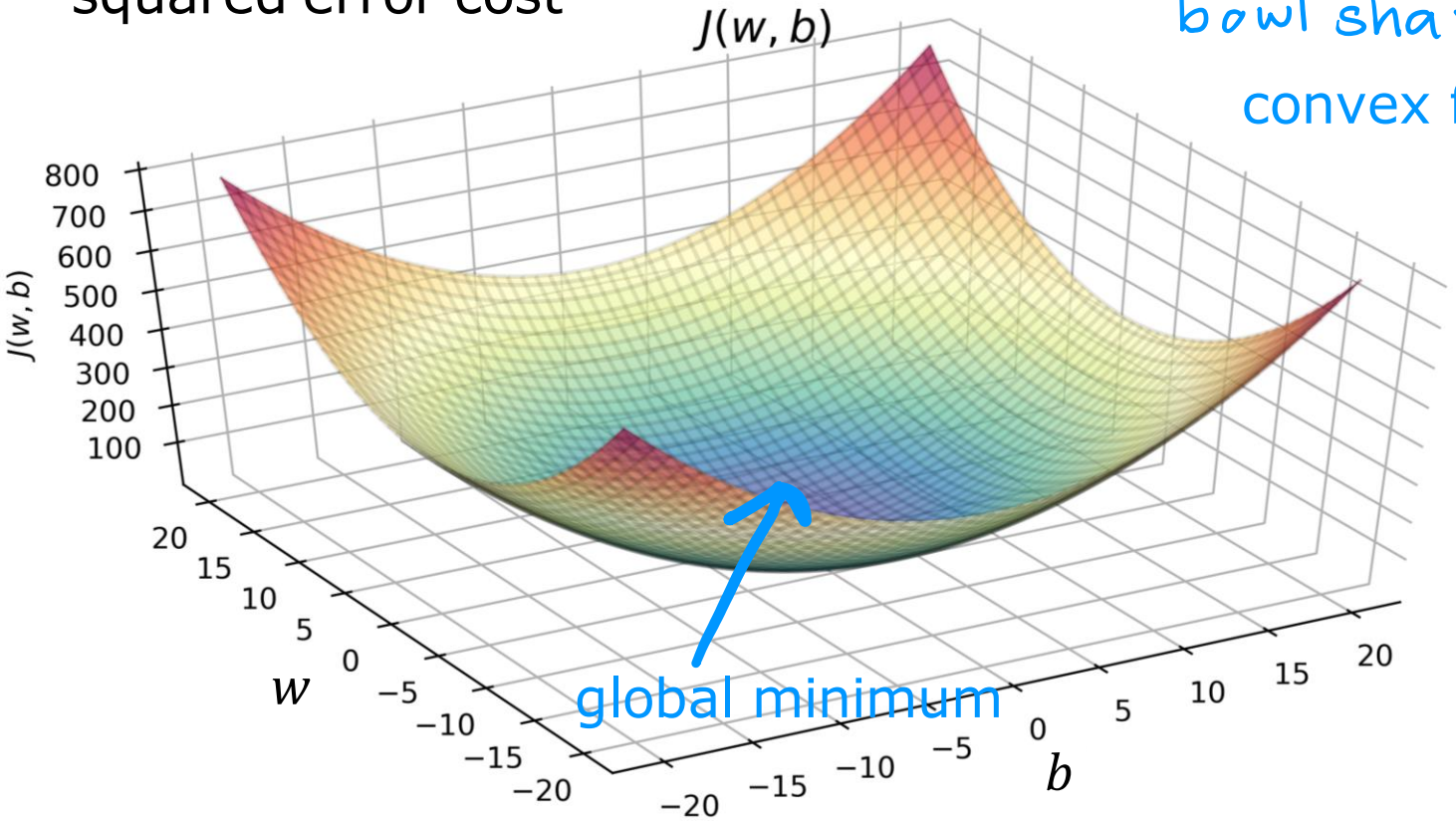
More than one local minimum



squared error cost

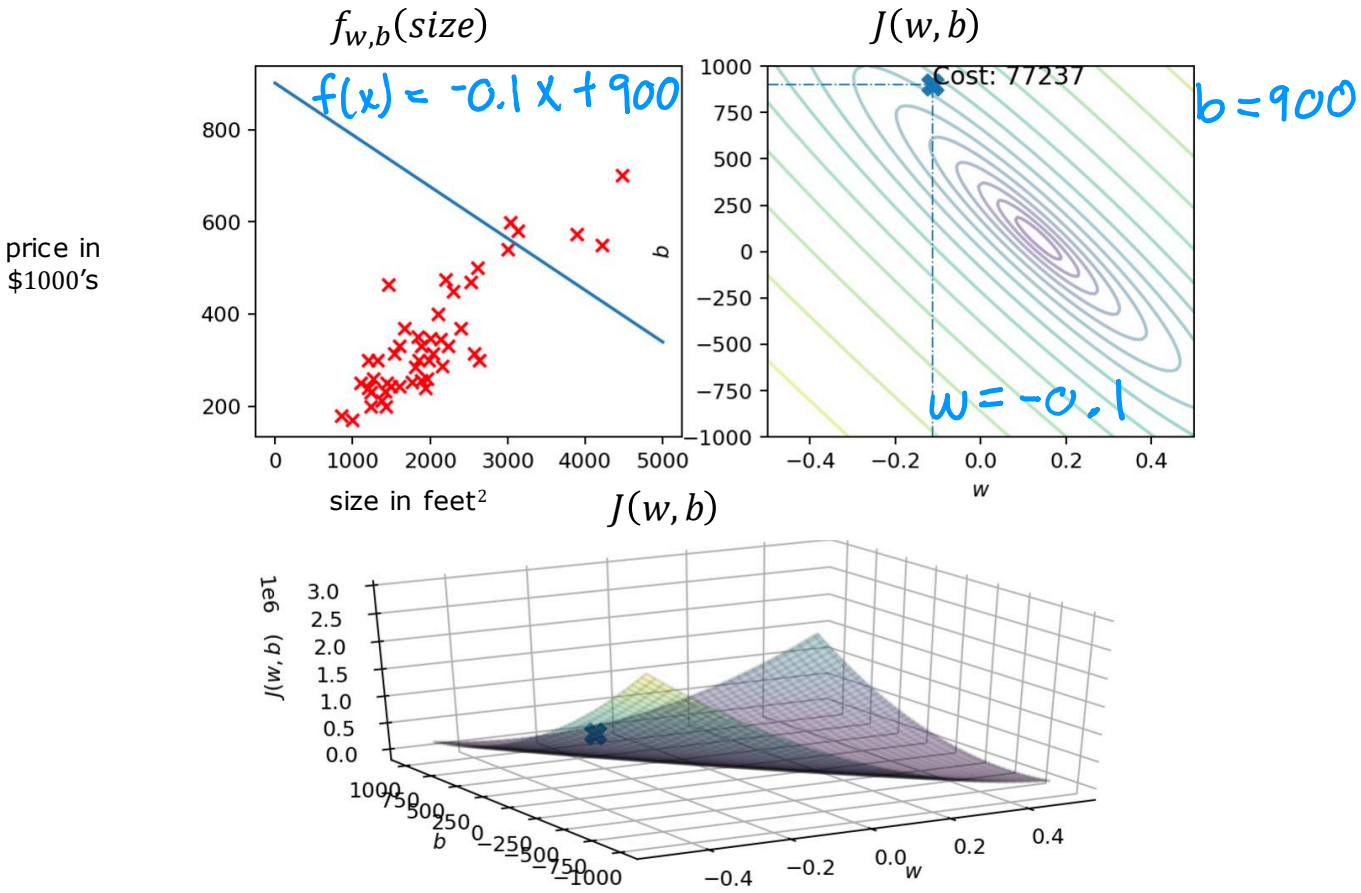
bowl shape

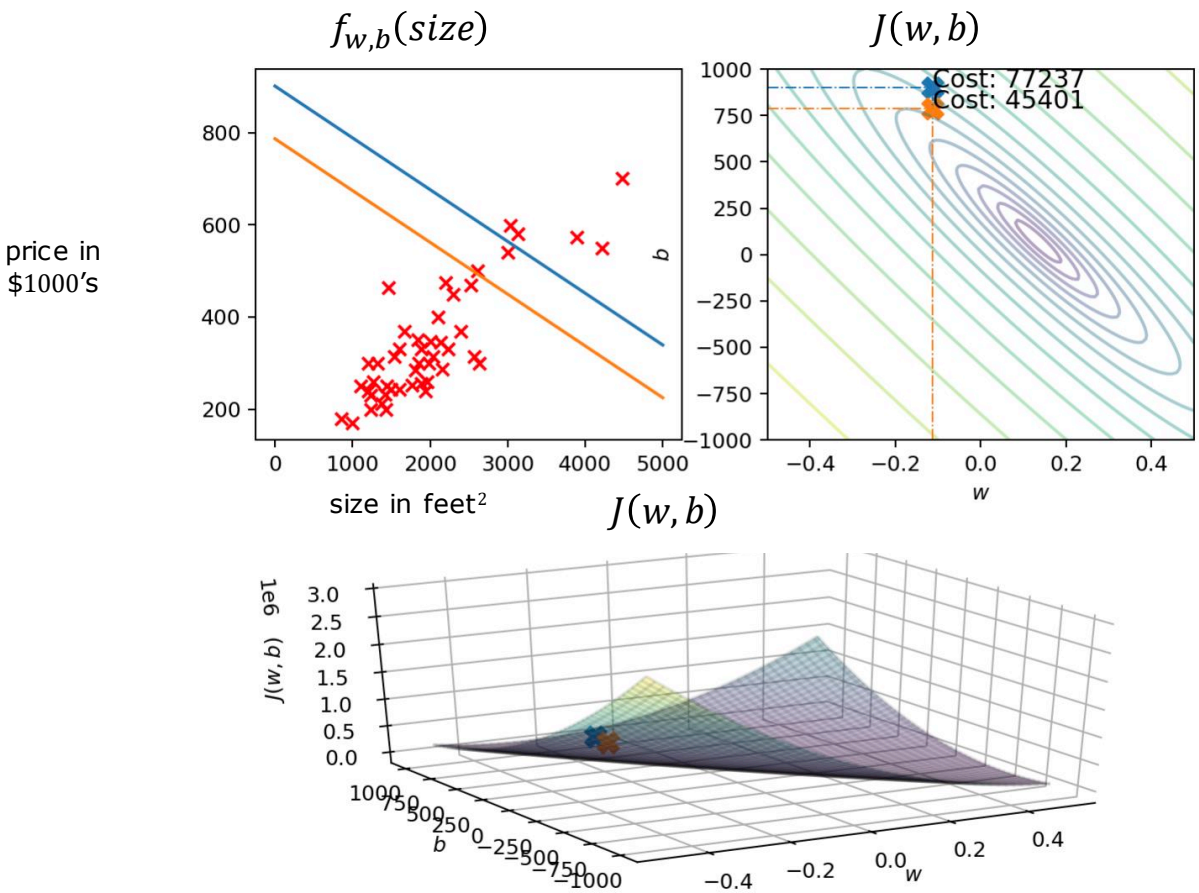
convex function

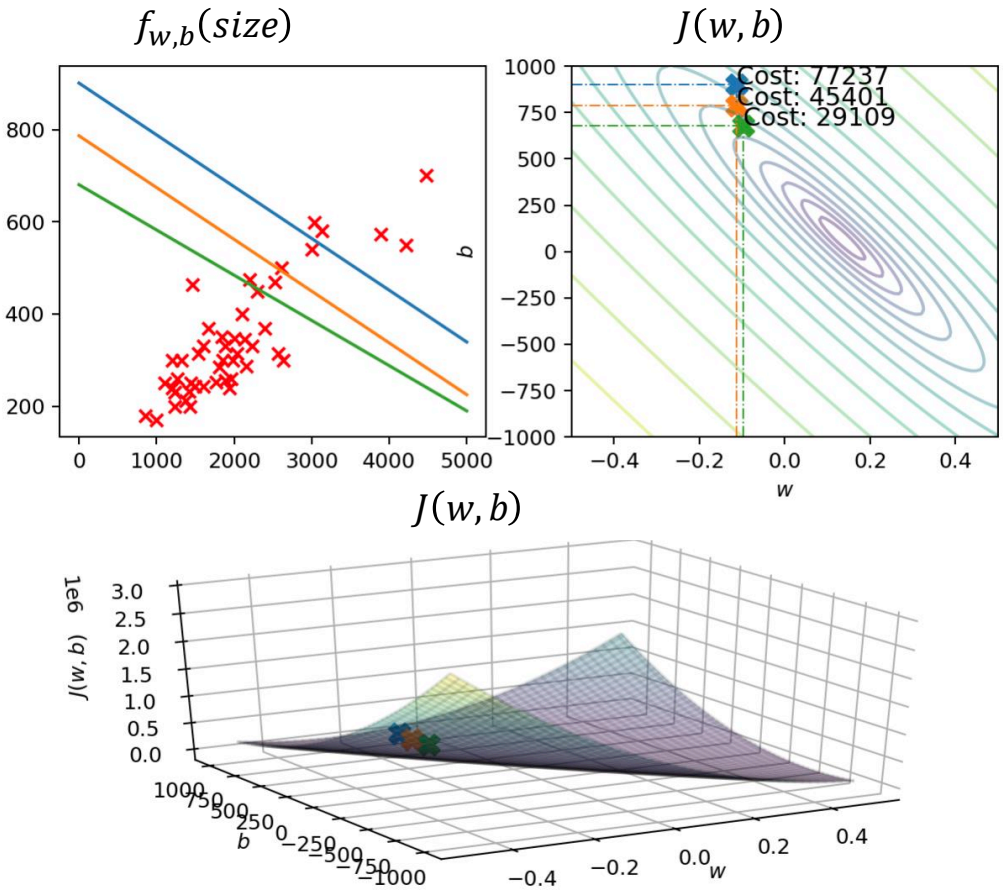


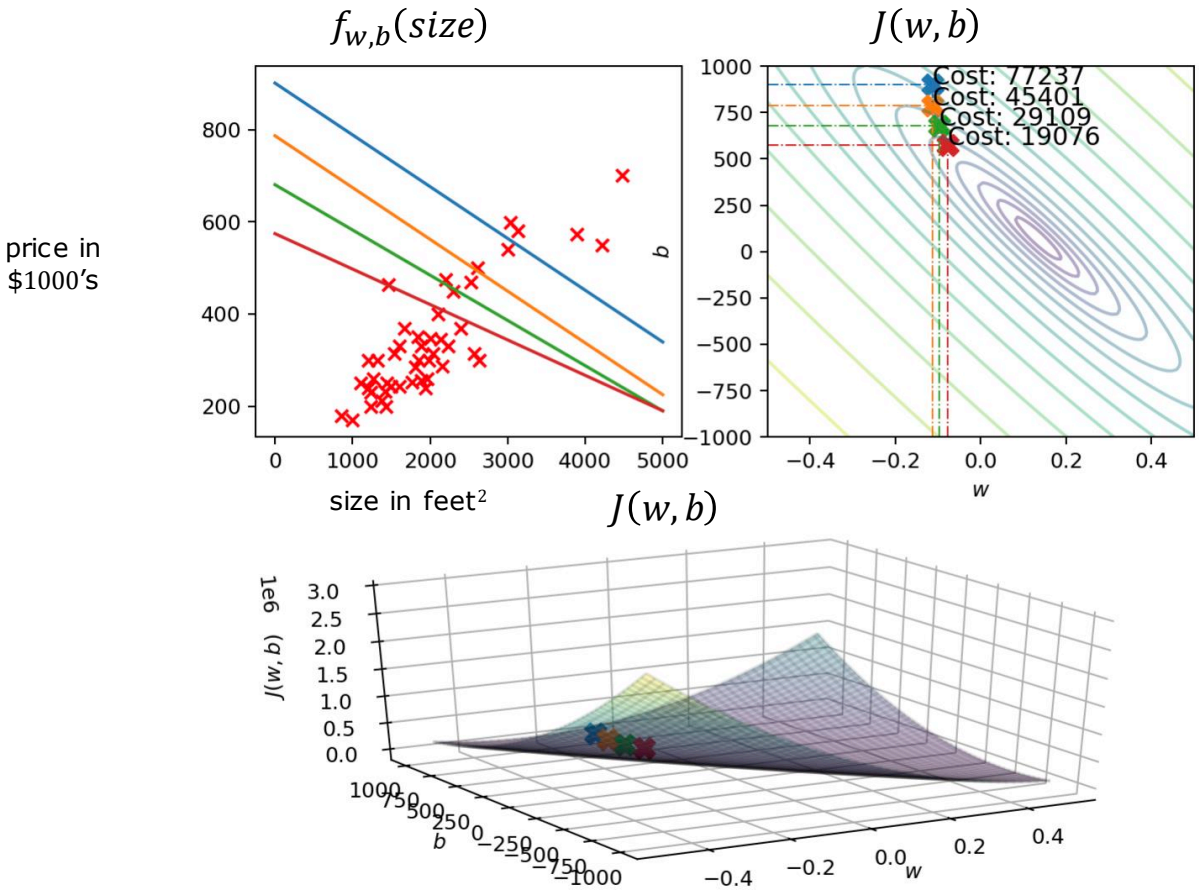
Training Linear Regression

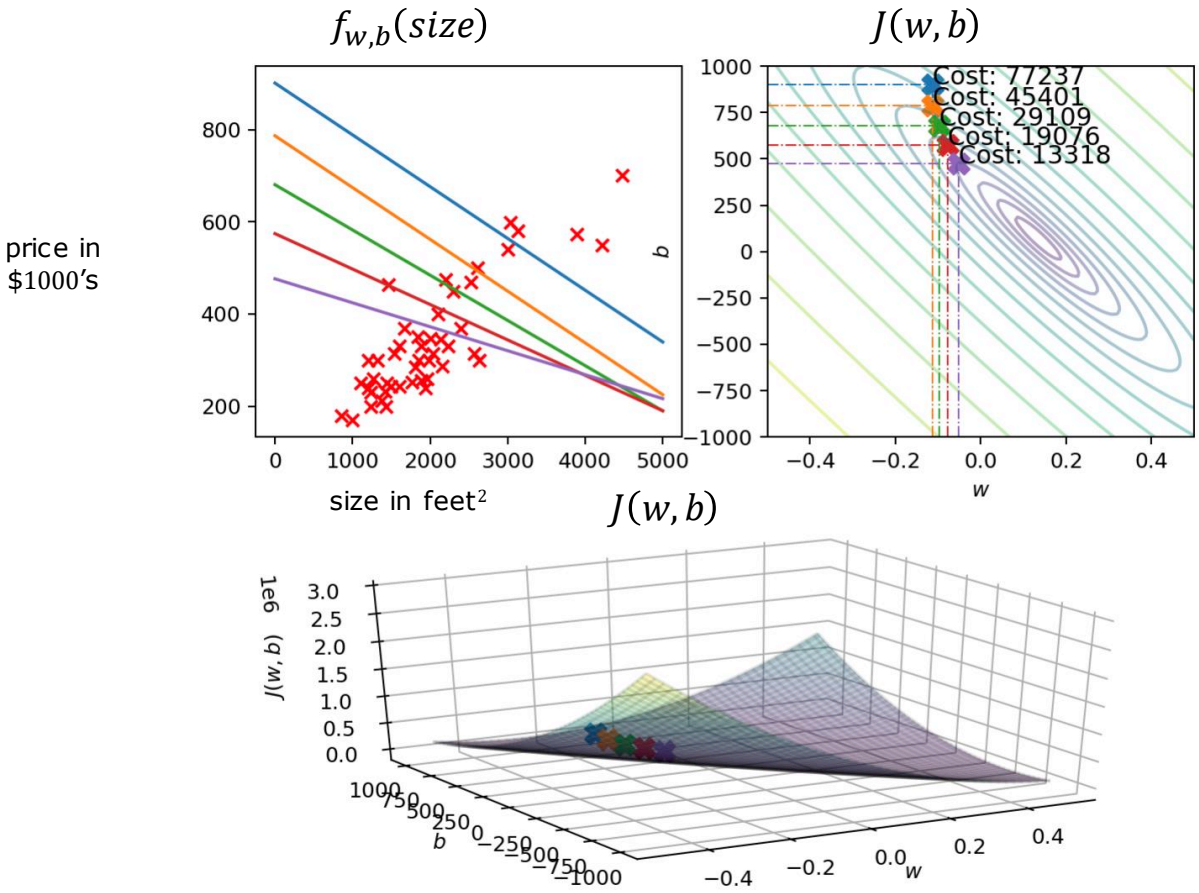
Running Gradient Descent

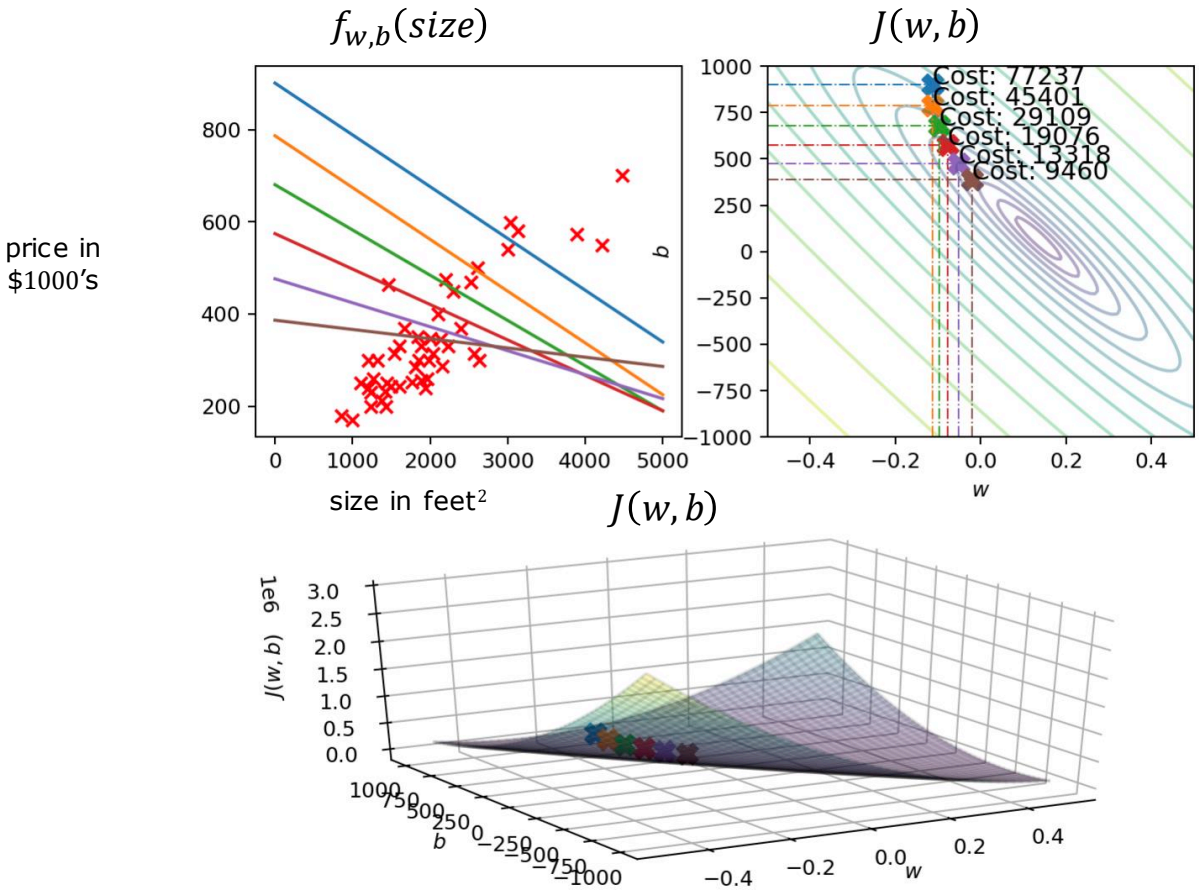

$f_{w,b}(size)$
$f(x) = -0.1x + 900$
price in $1000's
size in feet²
$J(w,b)$
Cost: 77237
$b = 900$
$w = -0.1$
$J(w,b)$
1e6

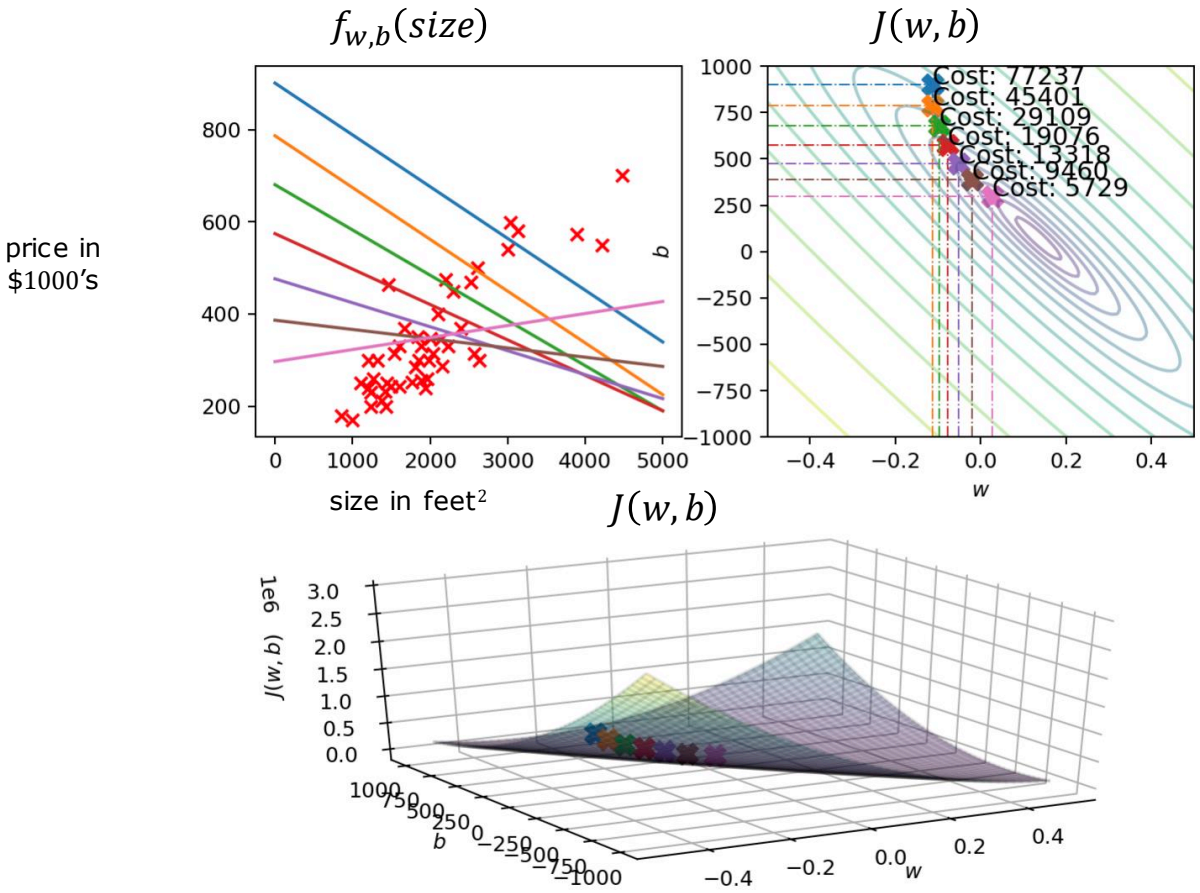

$f_{w,b}(size)$
price in $1000's
size in feet²
$J(w,b)$
Cost: 77237
Cost: 45401
$J(w,b)$
1e6

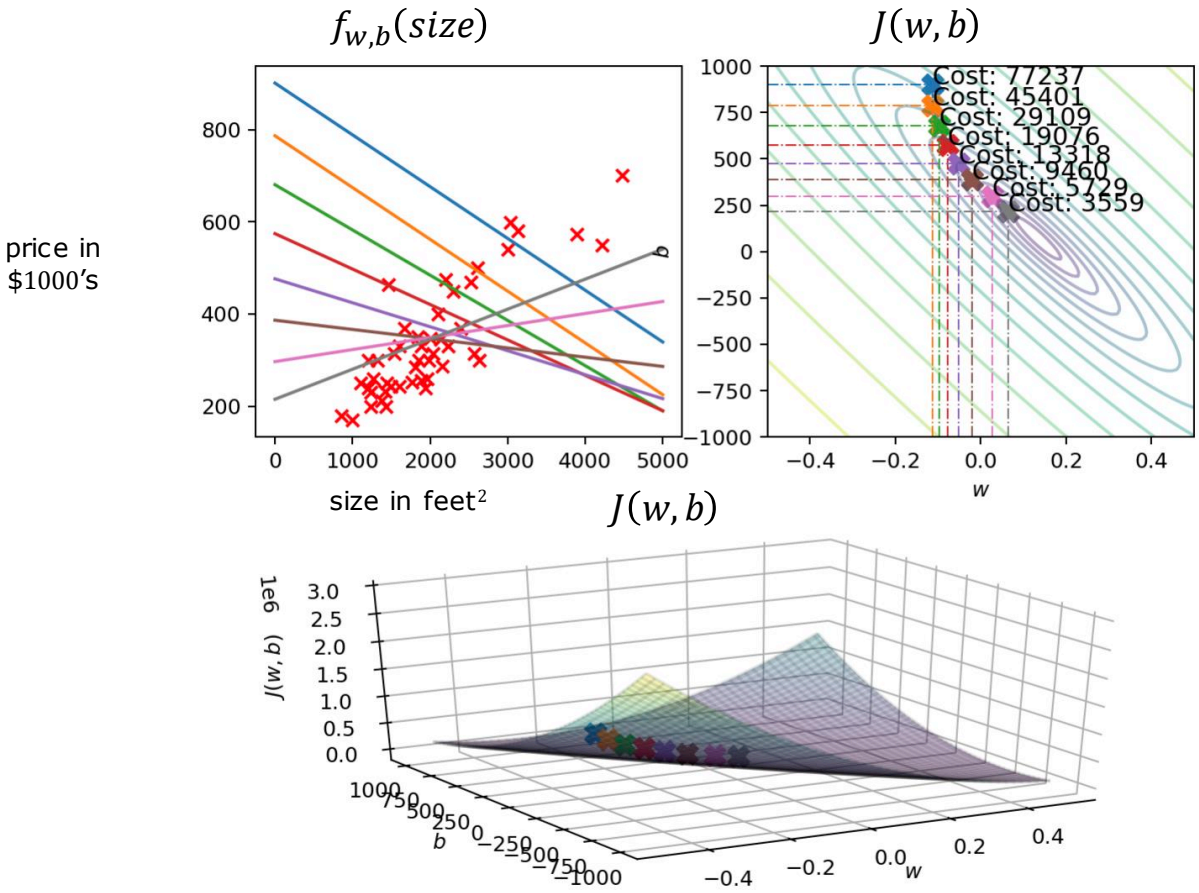

$f_{w,b}(size)$
800
600
400
200
0
1000
2000
3000
4000
5000
b
$J(w,b)$
Cost: 77237
Cost: 45401
Cost: 29109
1000
750
500
250
0
−250
−500
−750
−1000
−0.4
−0.2
0.0
0.2
0.4
w
$J(w,b)$
1e6
3.0
2.5
2.0
1.5
1.0
0.5
0.0
$J(w,b)$
1000
750
500
250
0
−250
−500
−750
−1000
b
−0.4
−0.2
0.0
0.2
0.4
w

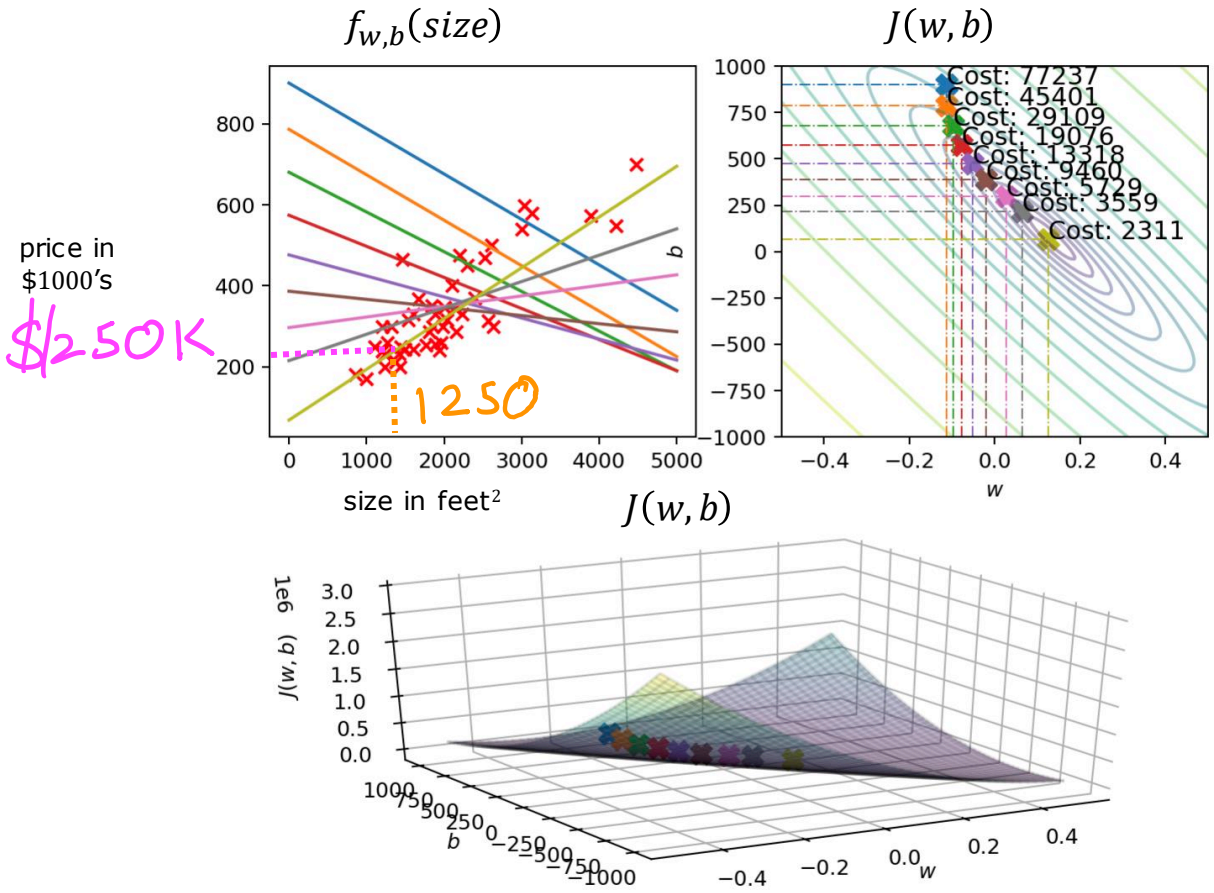

$f_{w,b}(size)$
price in $1000's
size in feet²
$J(w,b)$
Cost: 77237
Cost: 45401
Cost: 29109
Cost: 19076
$J(w,b)$
1e6


$f_{w,b}(size)$
price in $1000's
size in feet²
b
$J(w,b)$
Cost: 77237
Cost: 45401
Cost: 29109
Cost: 19076
Cost: 13318
w
$J(w,b)$
1e6
J(w, b)
b
w


$f_{w,b}(size)$
price in $1000's
size in feet²
b
$J(w,b)$
Cost: 77237
Cost: 45401
Cost: 29109
Cost: 19076
Cost: 13318
Cost: 9460
w
$J(w,b)$
1e6
$J(w,b)$
b
w


$f_{w,b}(size)$
price in $1000's
size in feet²
b
$J(w,b)$
Cost: 77237
Cost: 45401
Cost: 29109
Cost: 19076
Cost: 13318
Cost: 9460
Cost: 5729
w
$J(w,b)$
1e6
$J(w,b)$
b
w

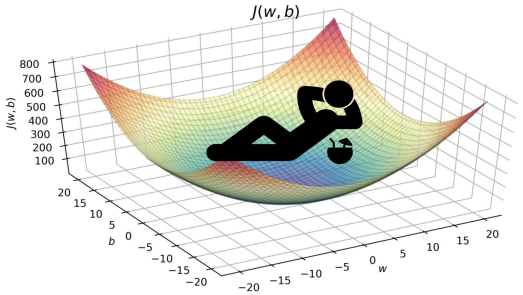

$f_{w,b}(size)$
price in $1000's
size in feet²
b
$J(w,b)$
Cost: 77237
Cost: 45401
Cost: 29109
Cost: 19076
Cost: 13318
Cost: 9460
Cost: 5729
Cost: 3559
w
$J(w,b)$
1e6
$J(w,b)$
b
w


$f_{w,b}(size)$
price in $1000's
$250K
1250
size in feet²
b
$J(w,b)$
Cost: 77237
Cost: 45401
Cost: 29109
Cost: 19076
Cost: 13318
Cost: 9460
Cost: 5729
Cost: 3559
Cost: 2311
w
$J(w,b)$
1e6
$J(w,b)$
b
w

“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

$m = 47$

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