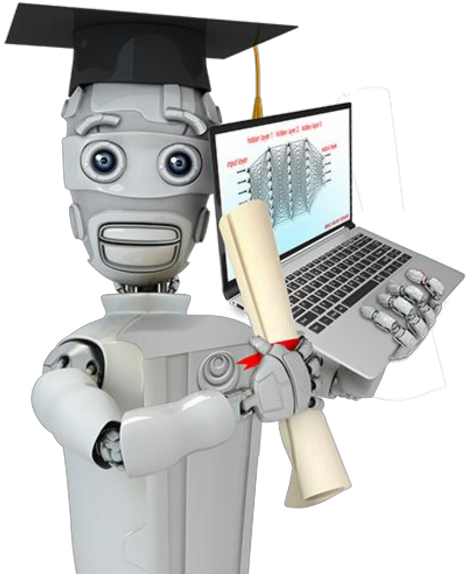
Copyright Notice

Machine Learning Overview



Supervised Learning Part 1

Supervised learning

x

input

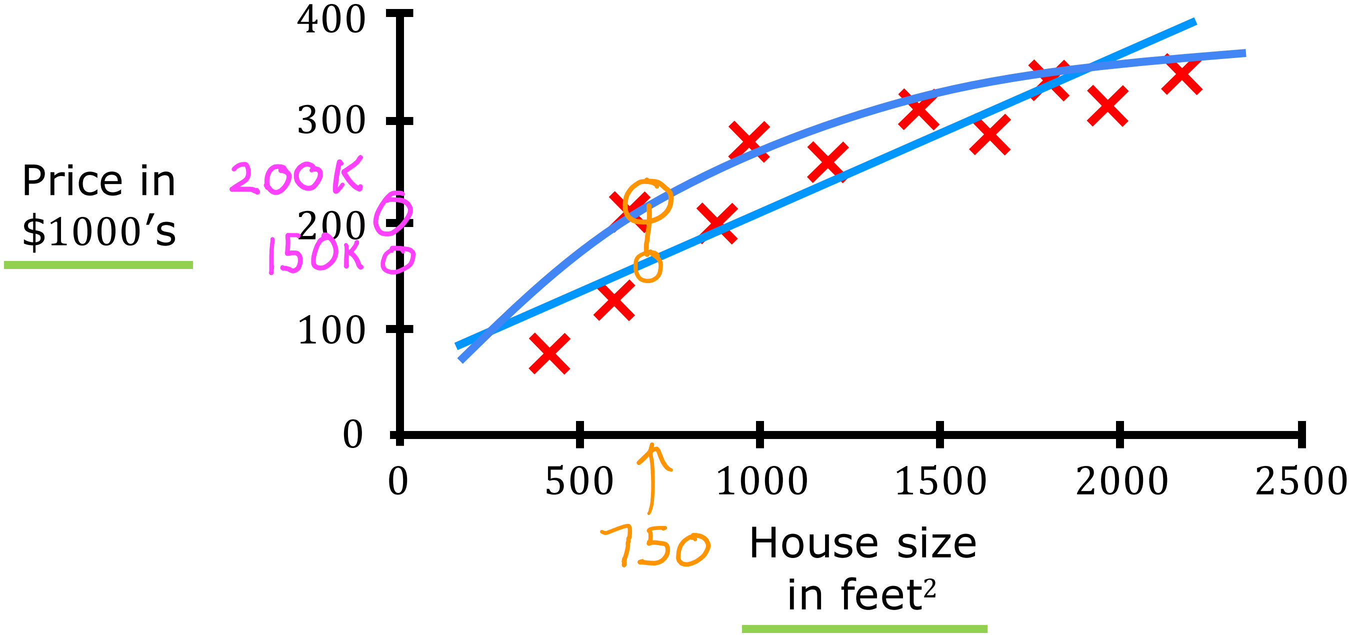
y

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

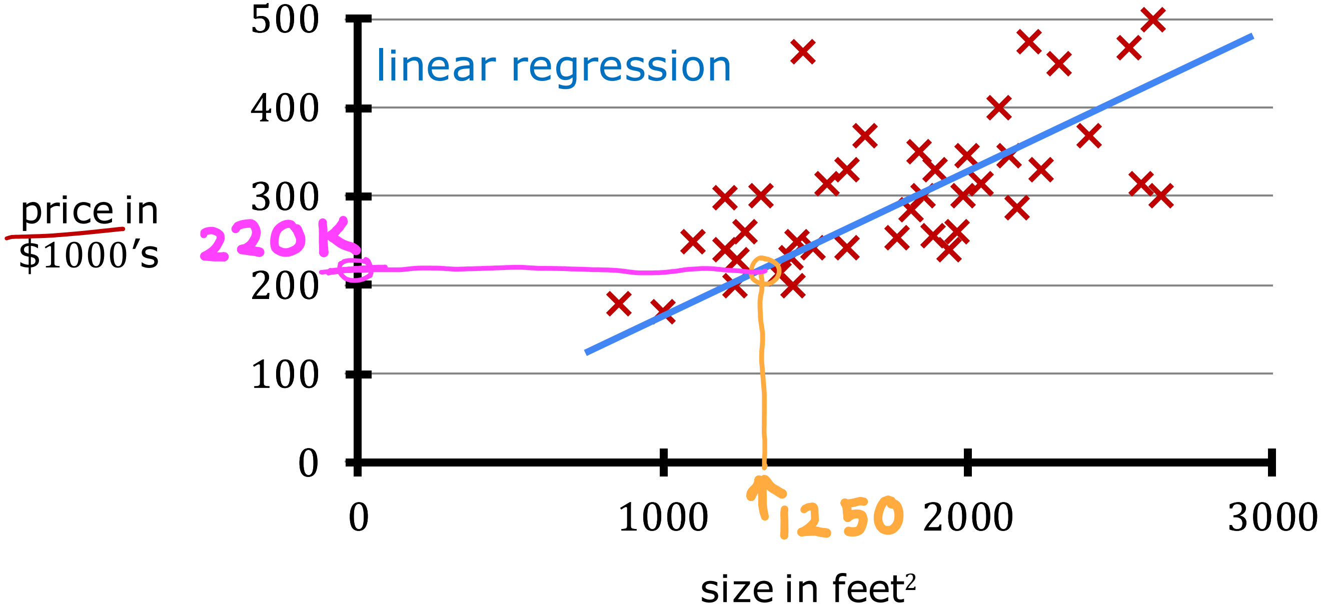
Regression: Housing price prediction



Linear Regression with One Variable

Linear Regression Model Part 1

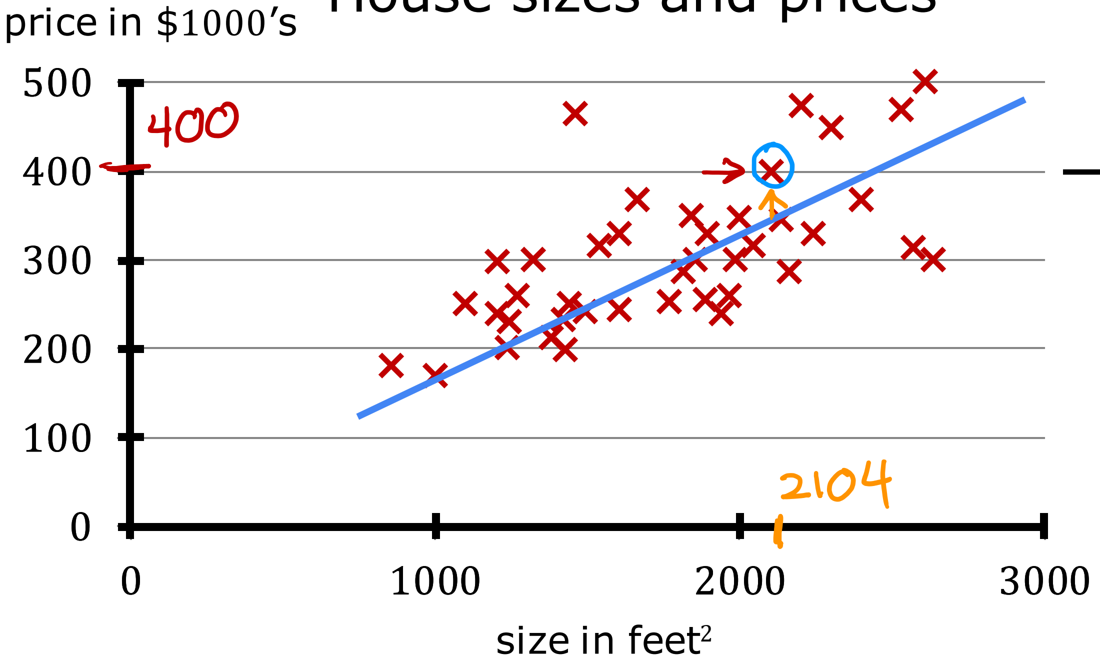
House sizes and prices



Regression model
Predicts numbers
Infinitely many possible outputs

Supervised learning model
Data has "right answers"

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

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

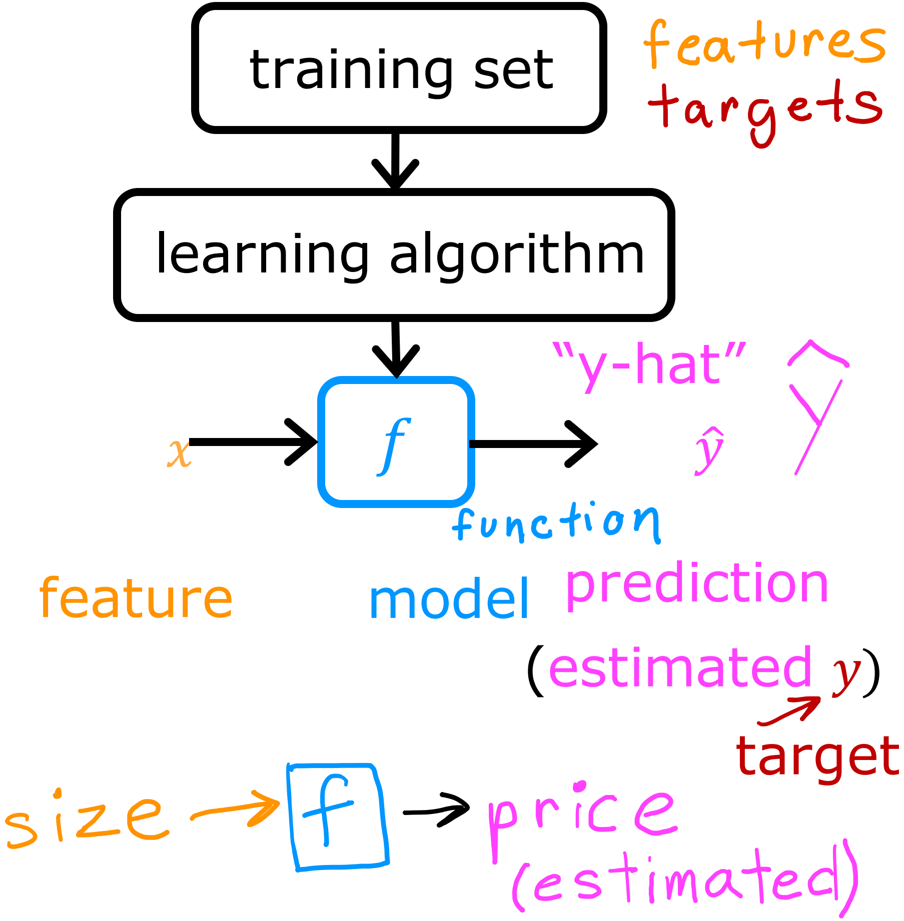
$(x^{(i)},y^{(i)})$ = i^{th} training example

index

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

Linear Regression with One Variable

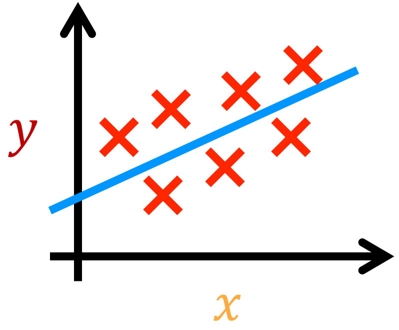
Linear Regression Model Part 2



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

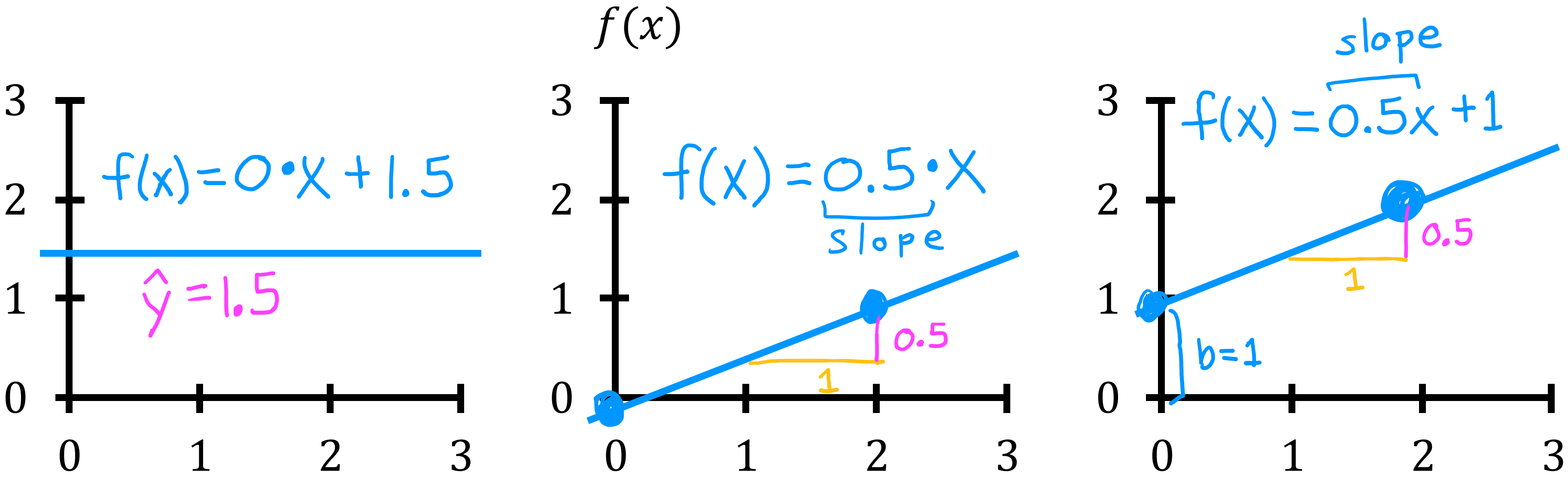
w, b: parameters

coefficients

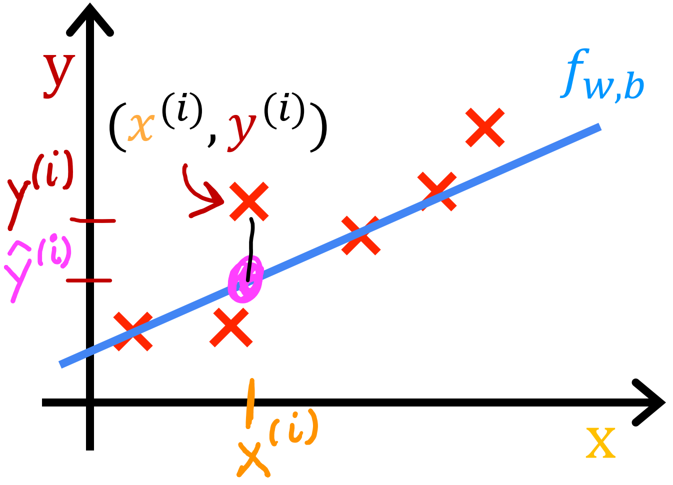
weights

What do w, b do?

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



$w = 0$	$w = 0.5$	$w = 0.5$
$b = 1.5$	$b = 0$	$b = 1$



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

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

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

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

Find w, b:

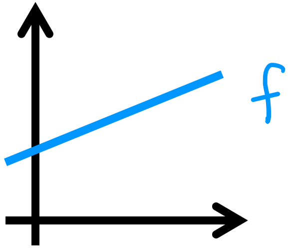
$\hat{y}^{(i)}$ is close to $y^{(i)}$ for all $(x^{(i)}, y^{(i)})$.

Linear Regression with One Variable

Cost Function Intuition

model:

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



parameters:

w, b

cost function:

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

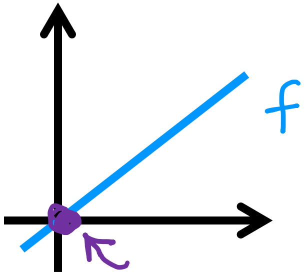
goal:

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

simplified

$$f_w(x) = wx$$

$b = \emptyset$



w

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

$wx^{(i)}$

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

$f_w(x)$

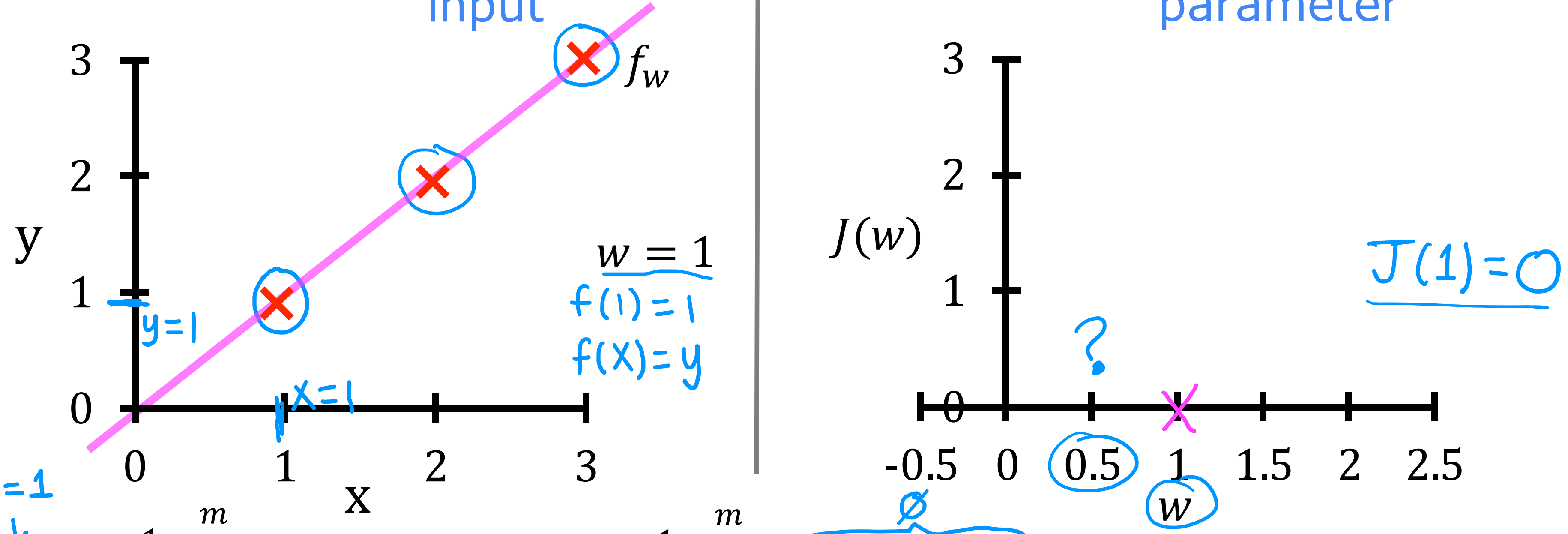
(for fixed w, function of x)

input

$J(w)$

(function of w)

parameter

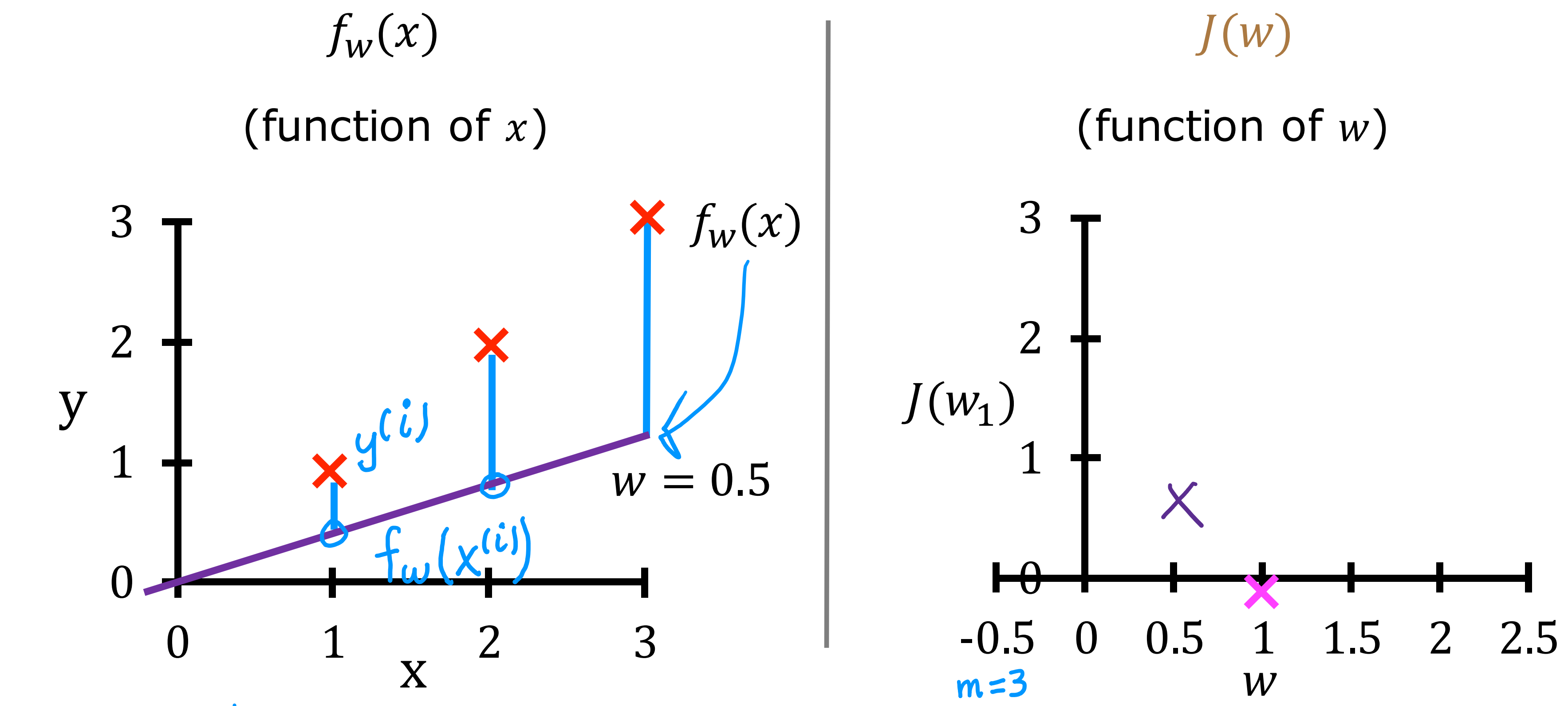


w=1

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

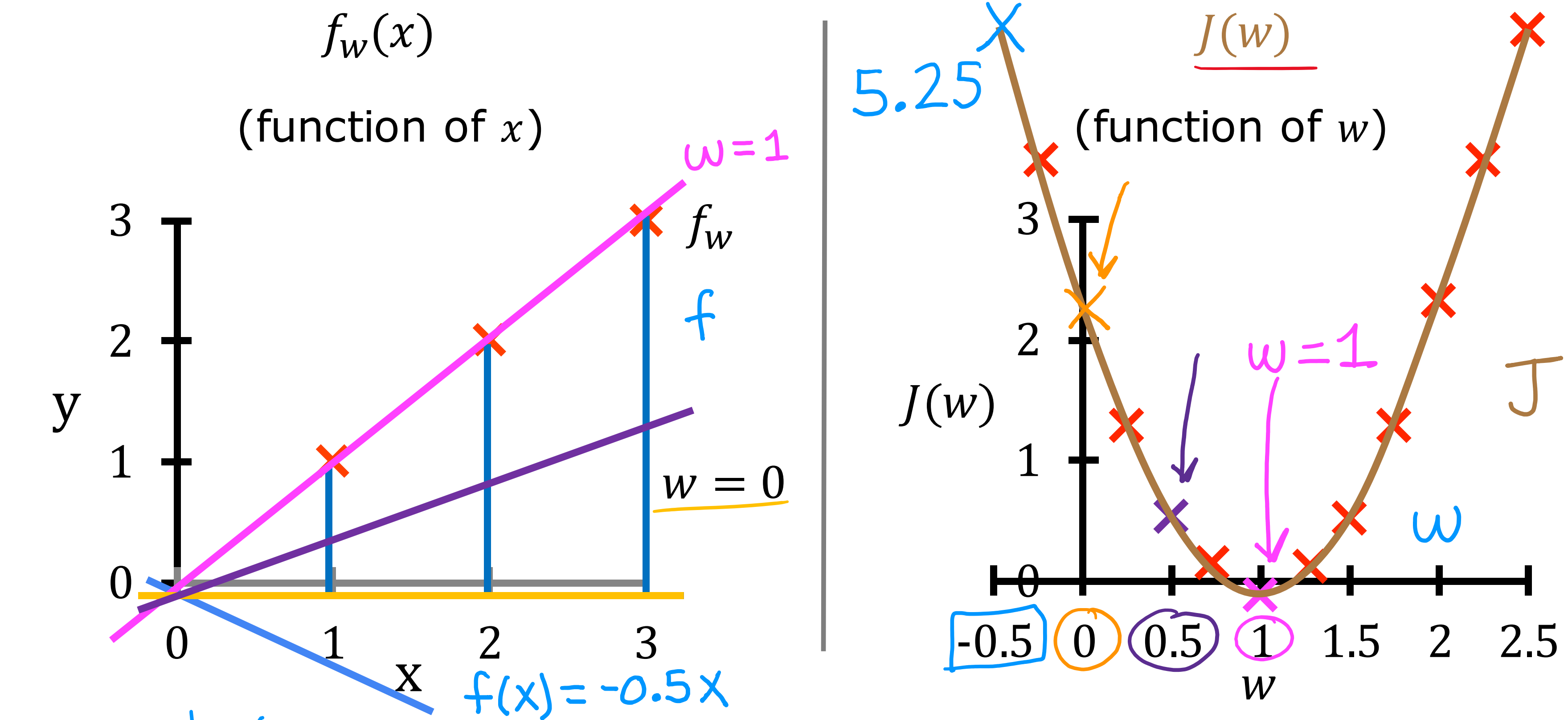
$wx^{(i)}$

$\emptyset$



$m=3$

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

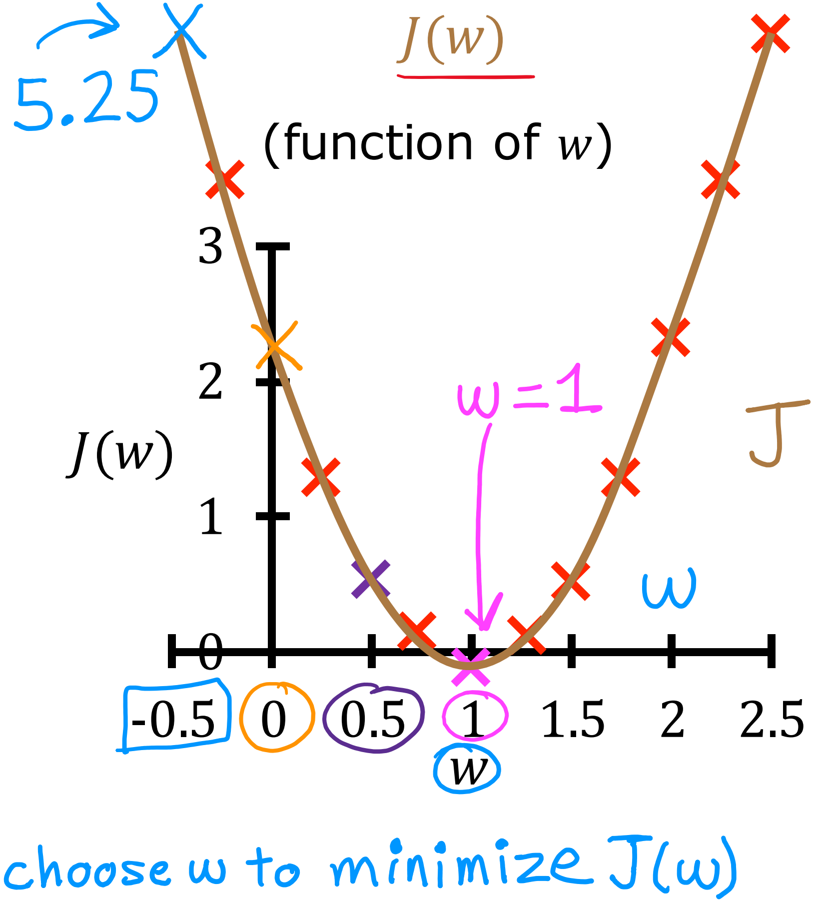


$$J(0) = \frac{1}{2m}\left(1^2 + 2^2 + 3^2\right) = \frac{1}{6}[14] \approx 2.3$$

how to choose w?

goal of linear regression:

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



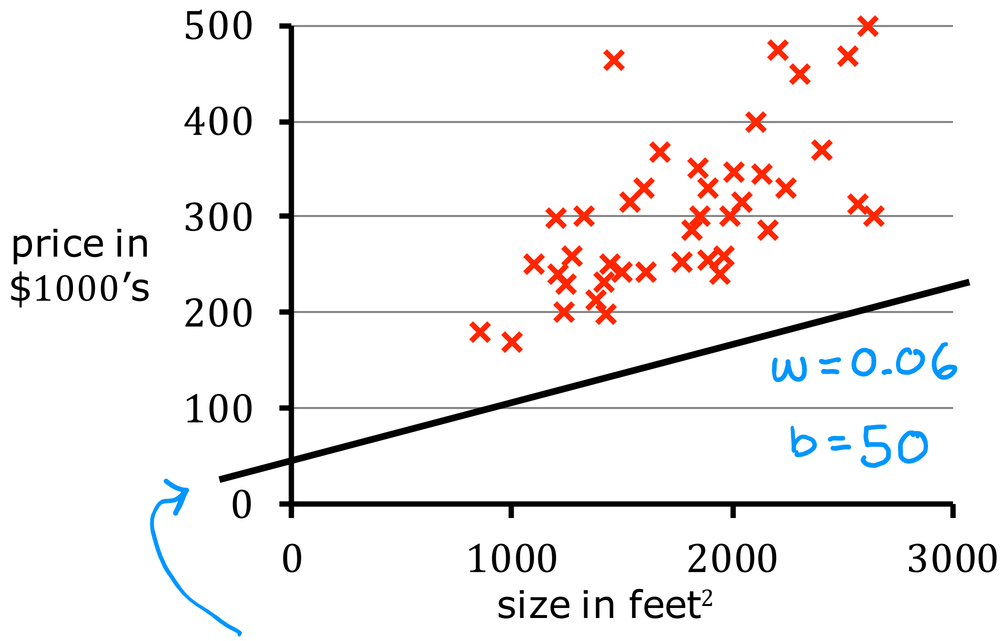
Linear Regression with One Variable

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

$f_{w,b}$

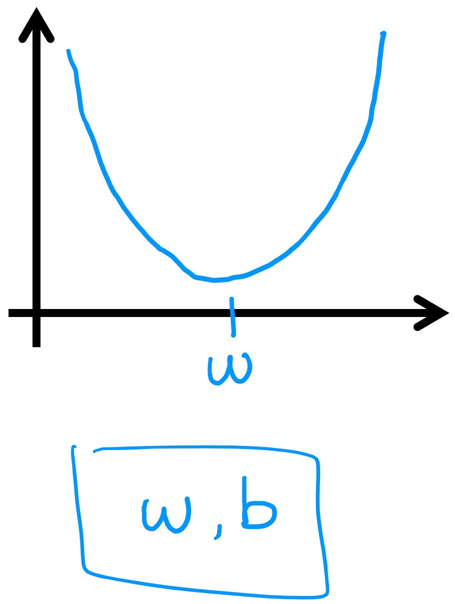
(function of x)

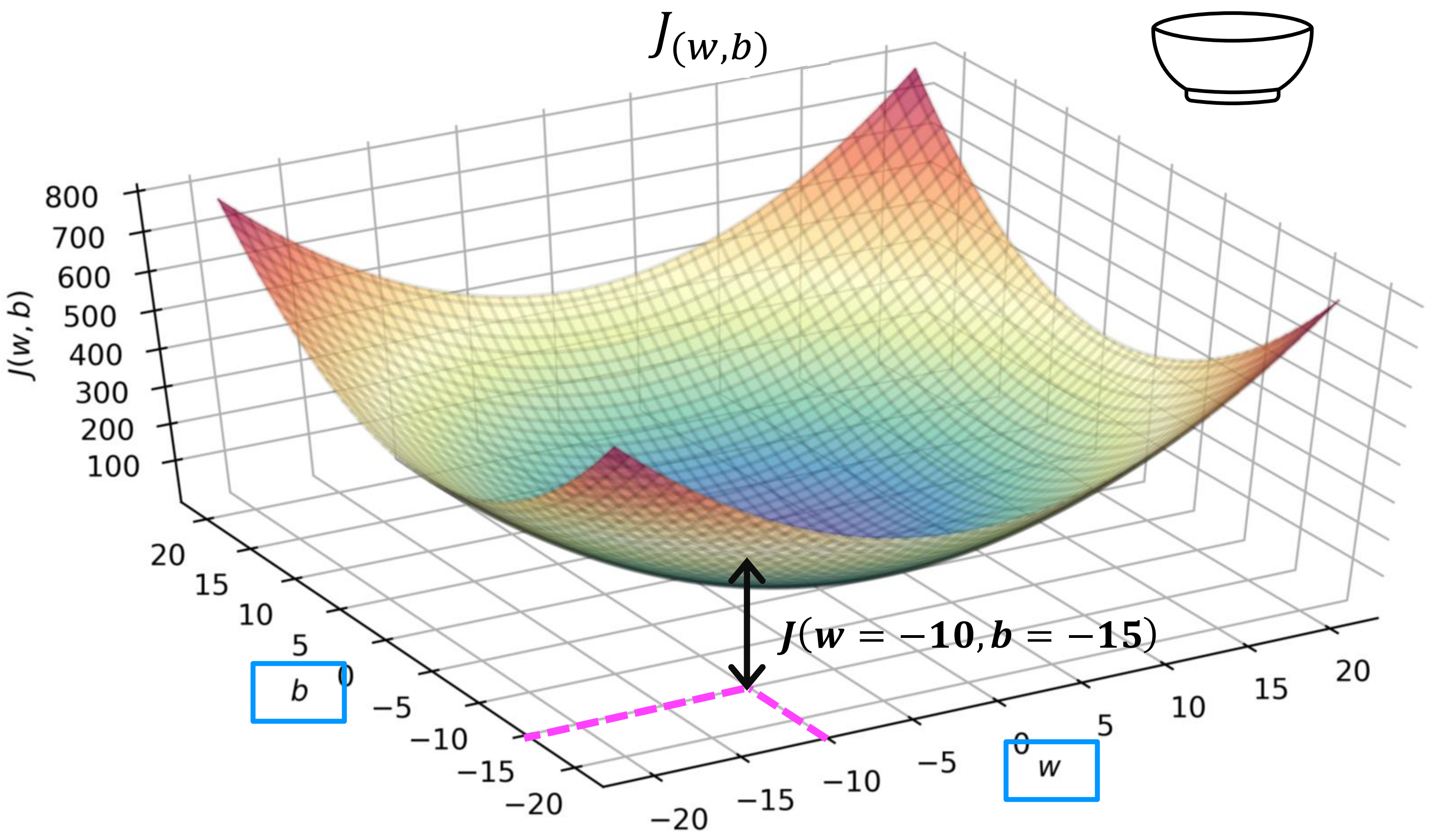


$$f_{w,b}(x) = 0.06x + 50$$

J

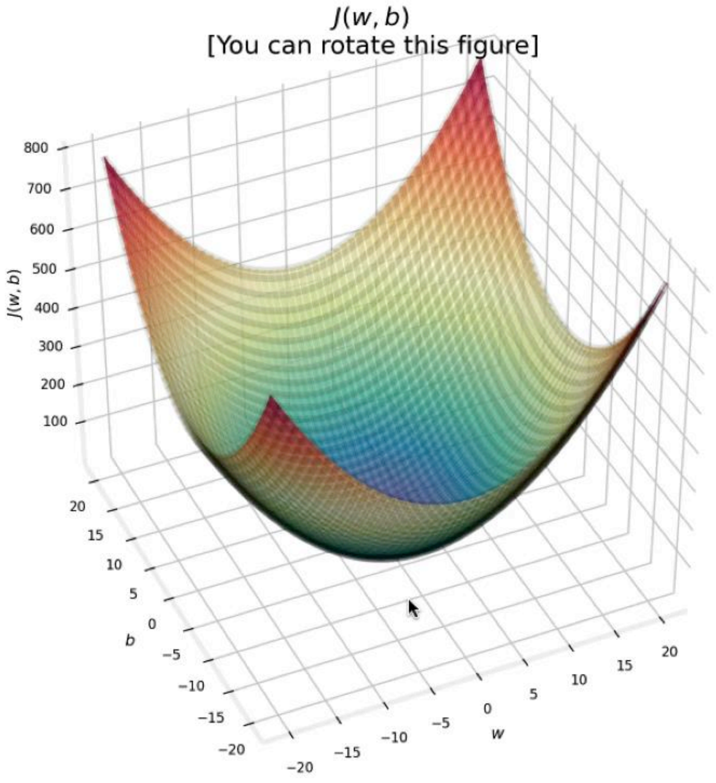
(function of w, b)



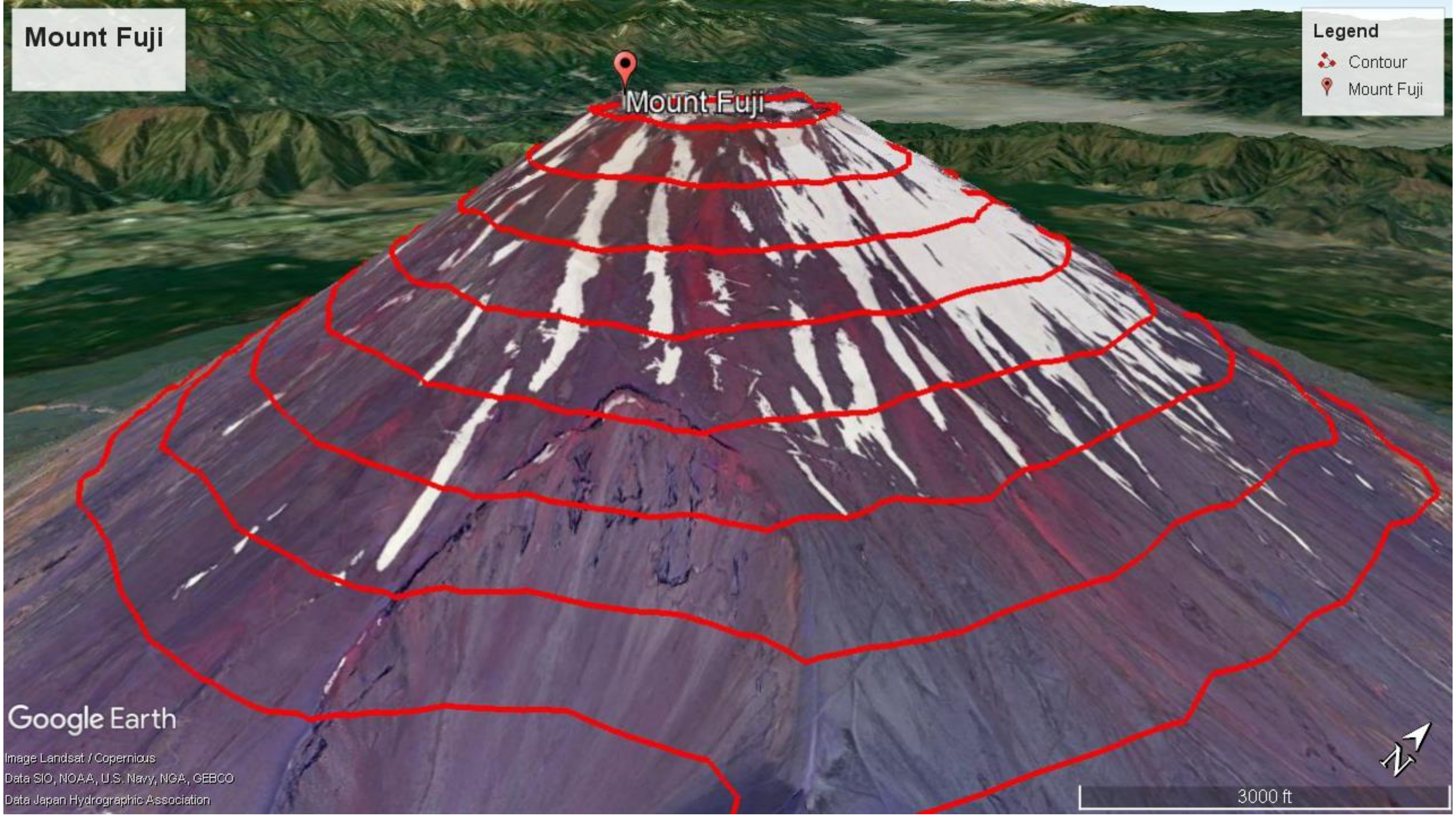


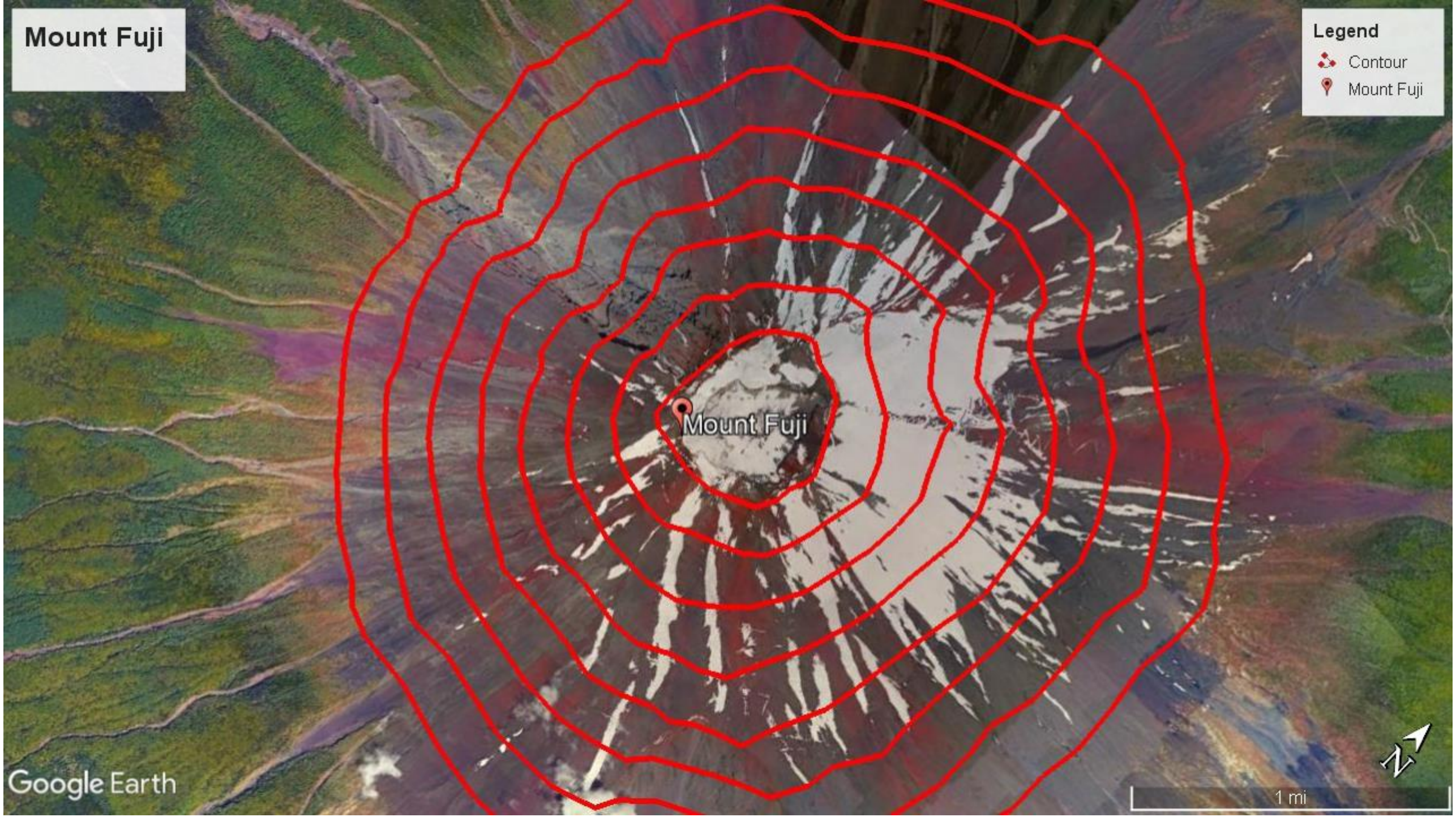
$J_{(w,b)}$
800
700
600
500
400
300
200
100
J(w, b)
20
15
10
5
0
−5
−10
−15
−20
b
$J(w = -10, b = -15)$
−20
−15
−10
−5
0
5
10
15
20
w

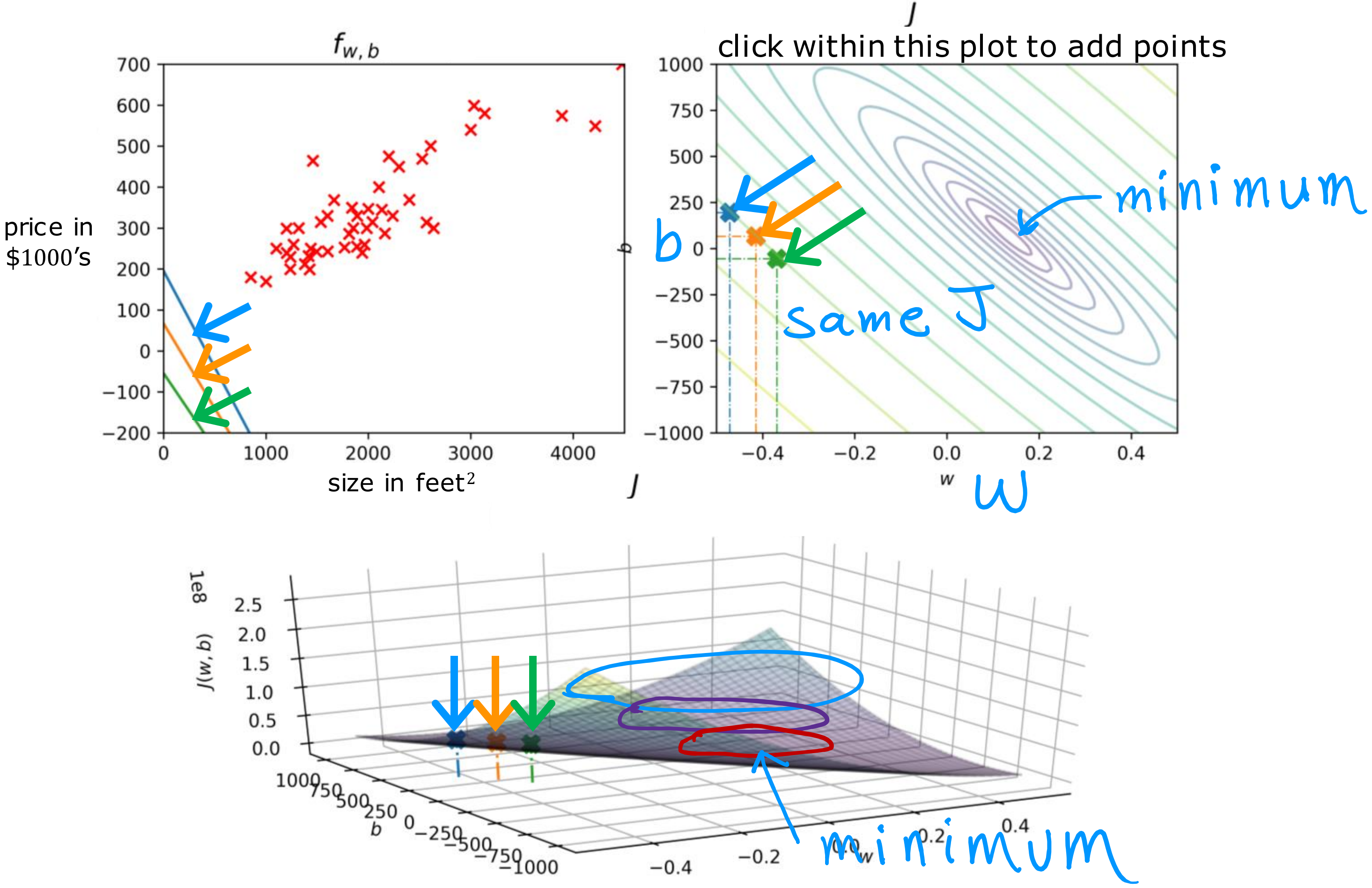
3D surface plot



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
N

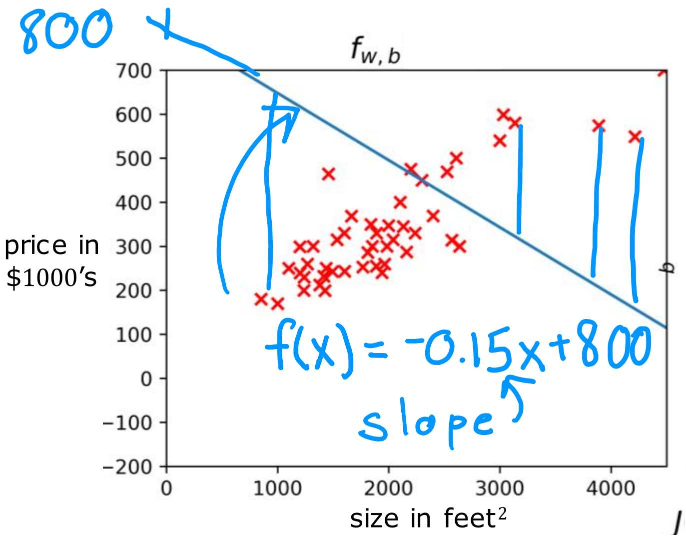

Mount Fuji
Legend
Contour
Mount Fuji
Mount Fuji
Google Earth
1 mi
N



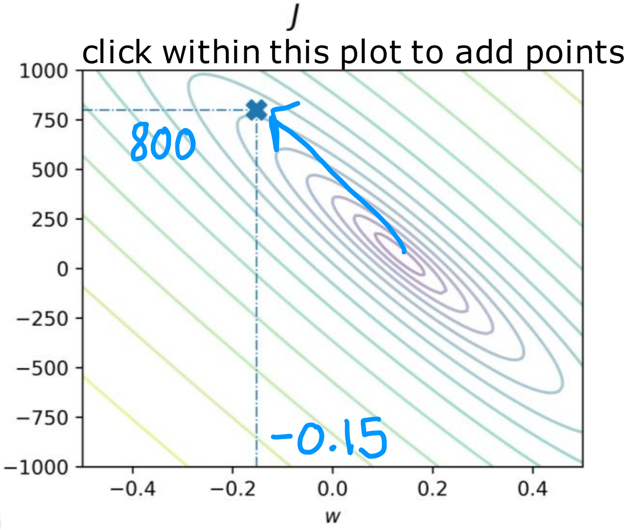
$f_{w,b}$
price in $1000's
size in feet²
J
click within this plot to add points
b
w
minimum
same J
J
1e8
J(w, b)
b
w
minimum

Linear Regression with One Variable

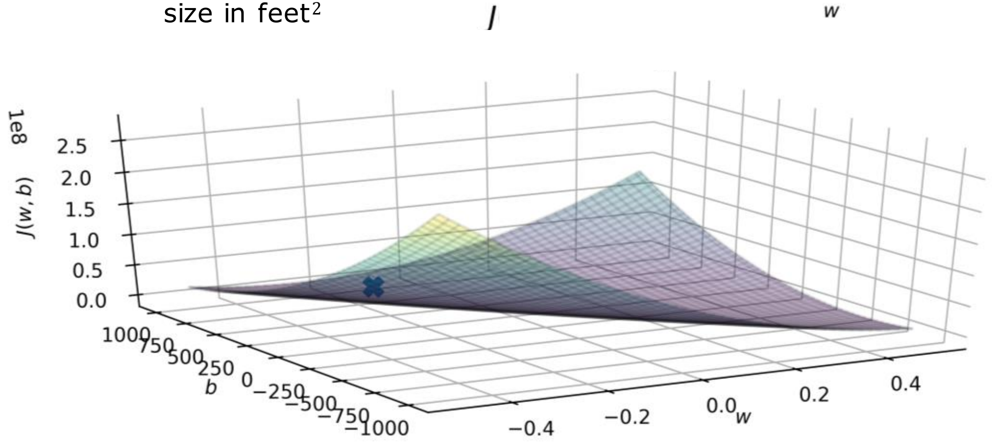
Visualization examples



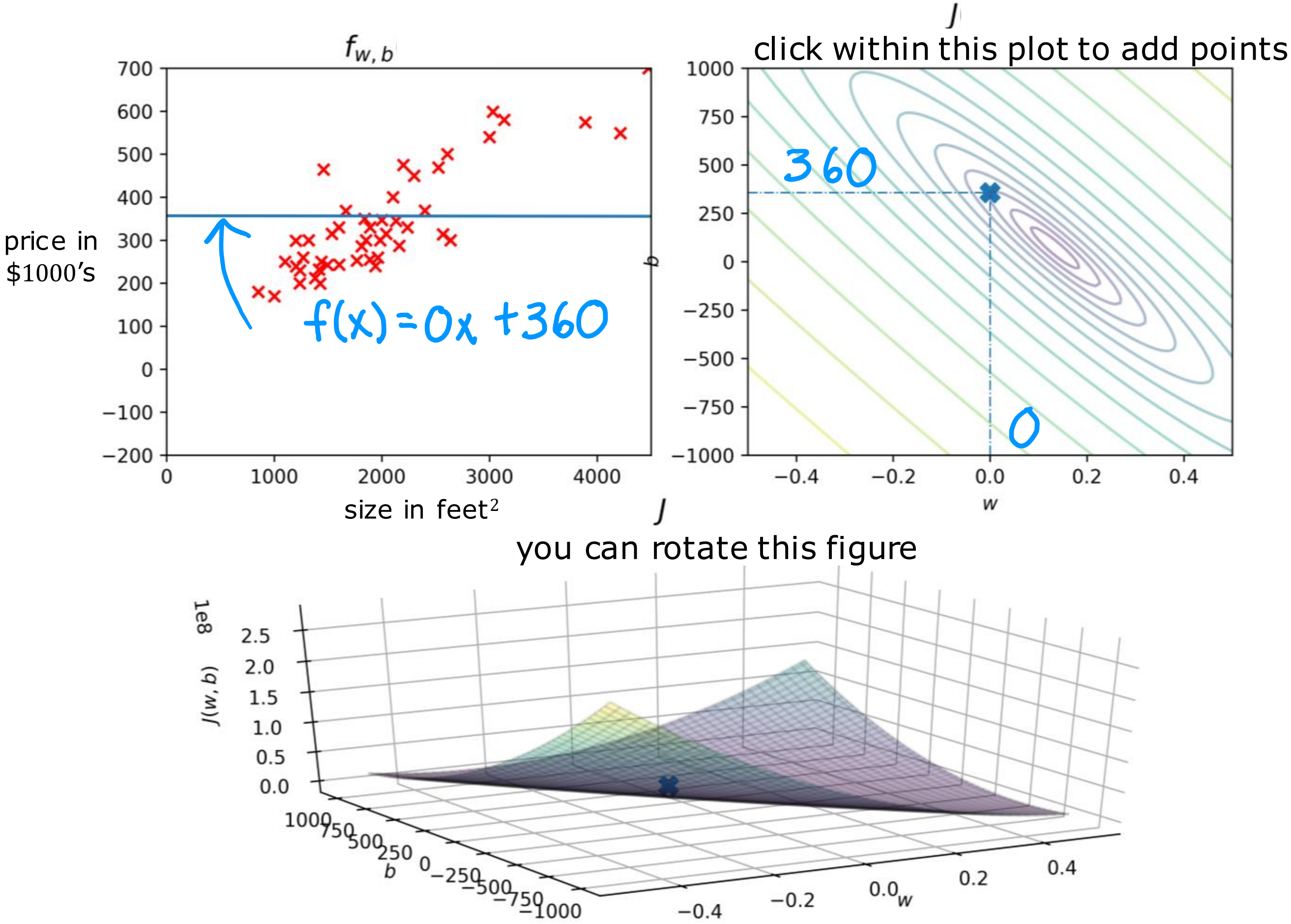
800
$f_{w,b}$
700
600
500
400
300
200
100
0
−100
−200
price in $1000's
0
1000
2000
3000
4000
size in feet²
f(x) = -0.15x+800
slope



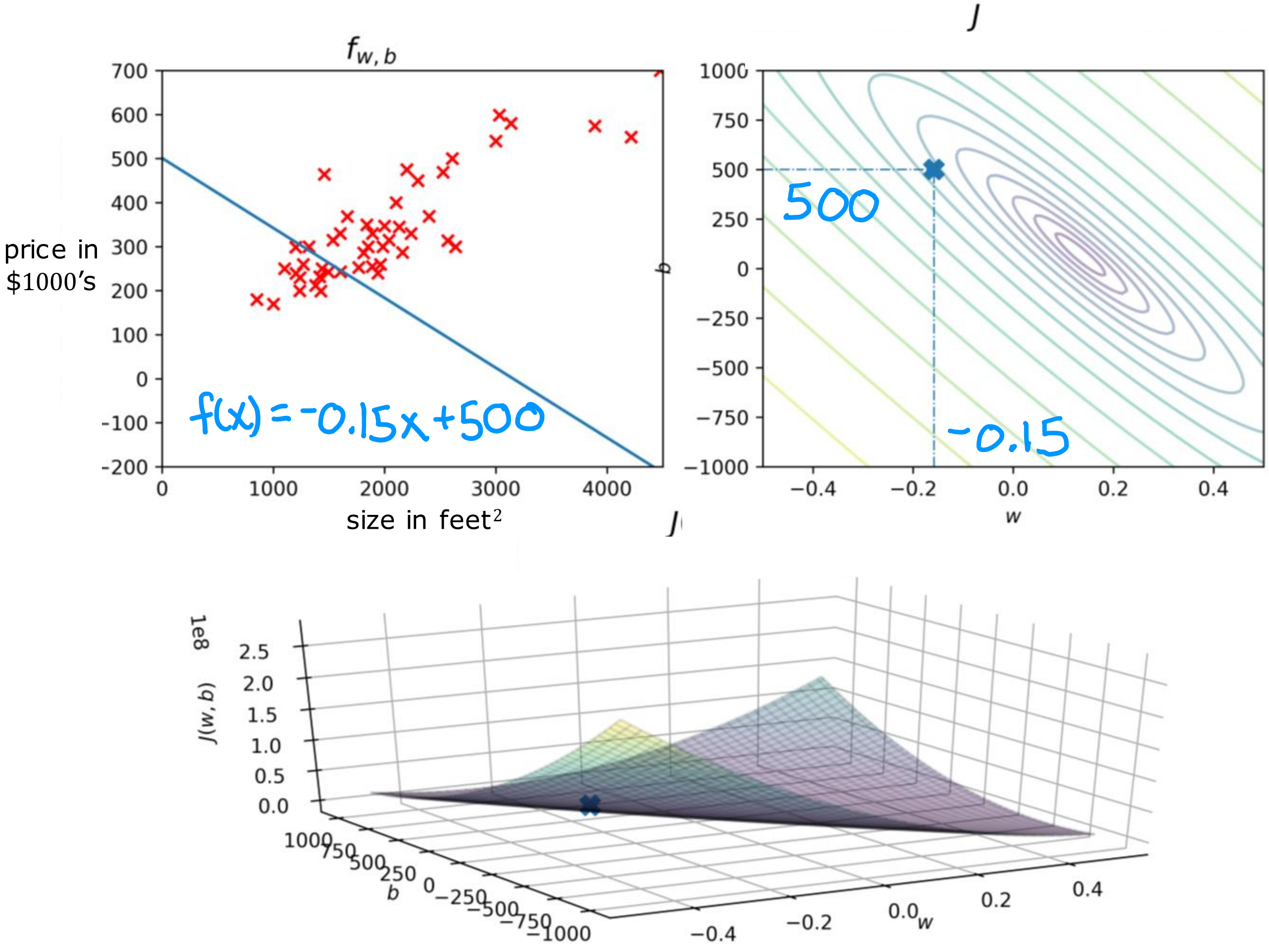
J
click within this plot to add points
1000
750
500
250
0
−250
−500
−750
−1000
b
800
-0.15
−0.4
−0.2
0.0
0.2
0.4
w

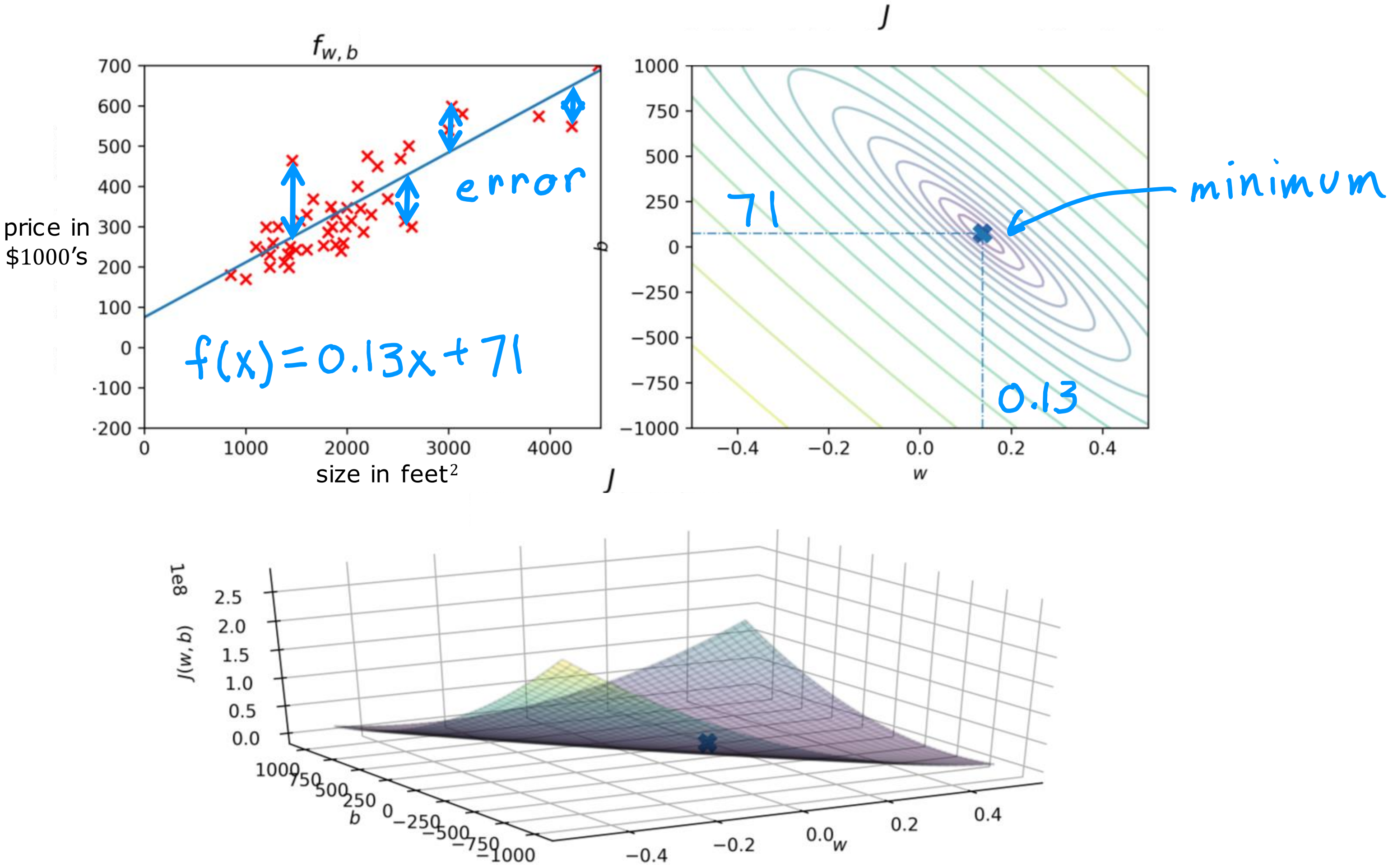


J
1e8
2.5
2.0
1.5
1.0
0.5
0.0
J(w, b)
1000
750
500
250
0
−250
−500
−750
−1000
b
−0.4
−0.2
0.0
0.2
0.4
w



$f_{w,b}$
price in $1000's
size in feet^2
f(x)=0x +360
J
click within this plot to add points
360
0
b
w
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
b
w
500
-0.15
J(w, b)
1e8



$f_{w,b}$
price in $1000's
size in feet²
error
f(x)=0.13x+71
J
b
w
71
0.13
minimum
J
1e8
J(w, b)
b
w

Training Linear Regression

Gradient Descent

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

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

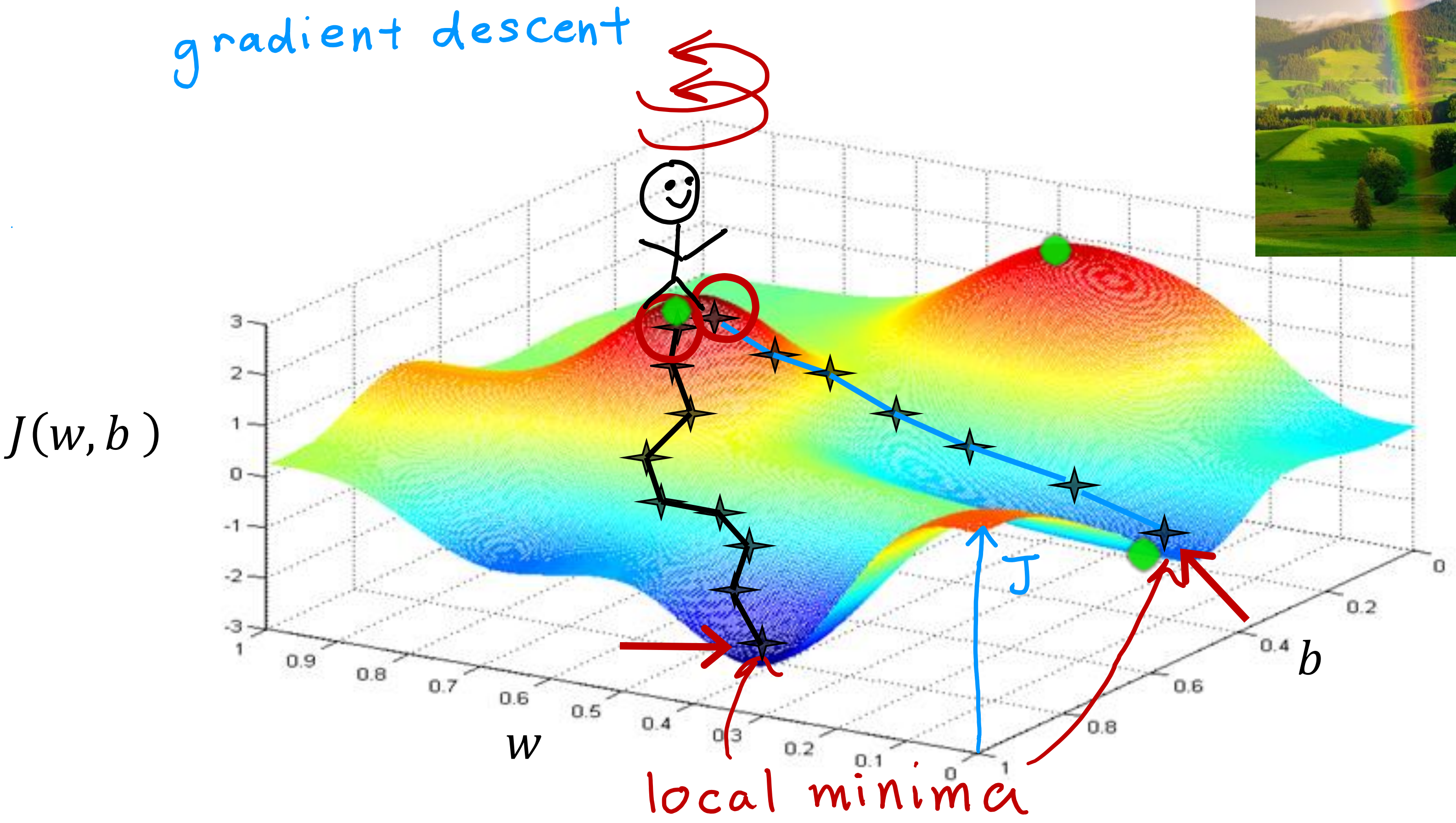
Outline:

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)
3
2
1
0
-1
-2
-3
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
w
0
0.2
0.4
0.6
0.8
1
b
J
local minima

Training Linear Regression

Implementing Gradient Descent

Gradient descent algorithm

Repeat until convergence

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

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

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

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

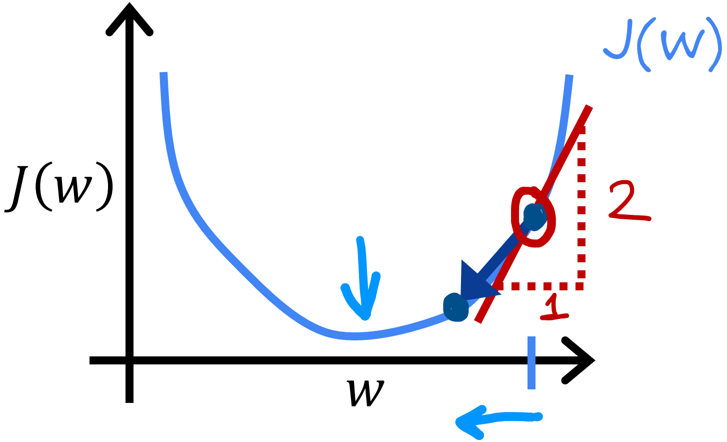
derivative

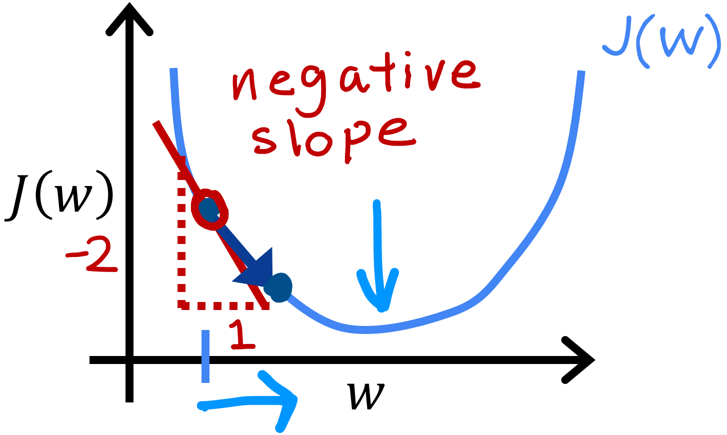
$$b = b - \alpha \frac{\partial}{\partial b} J(w, b)$$

$J(w)$

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

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





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

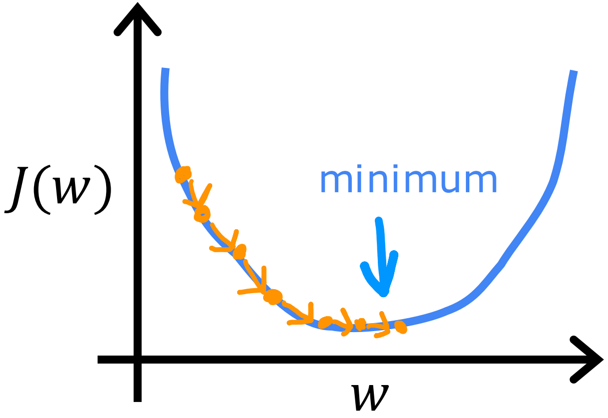
Training Linear Regression

Learning Rate

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

If α is too small…

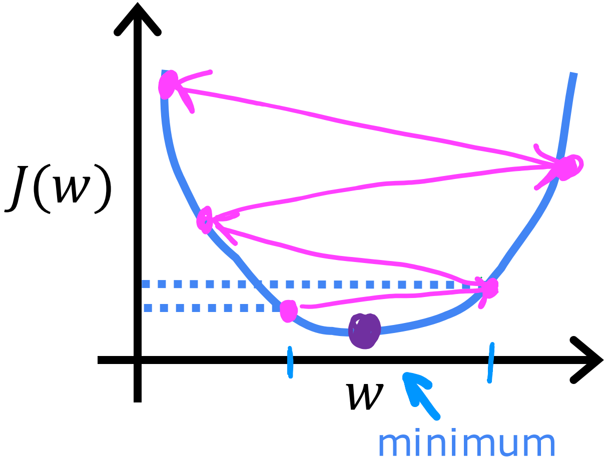
Gradient descent may be slow.

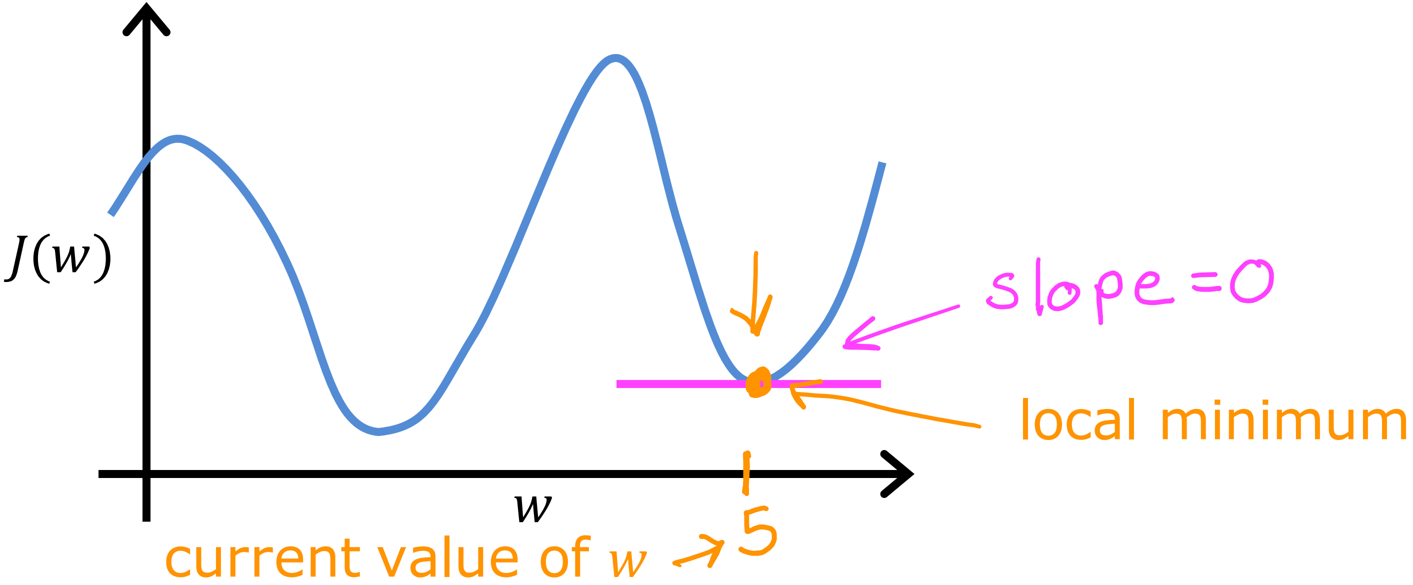


If α is too large…

Gradient descent may:

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





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

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

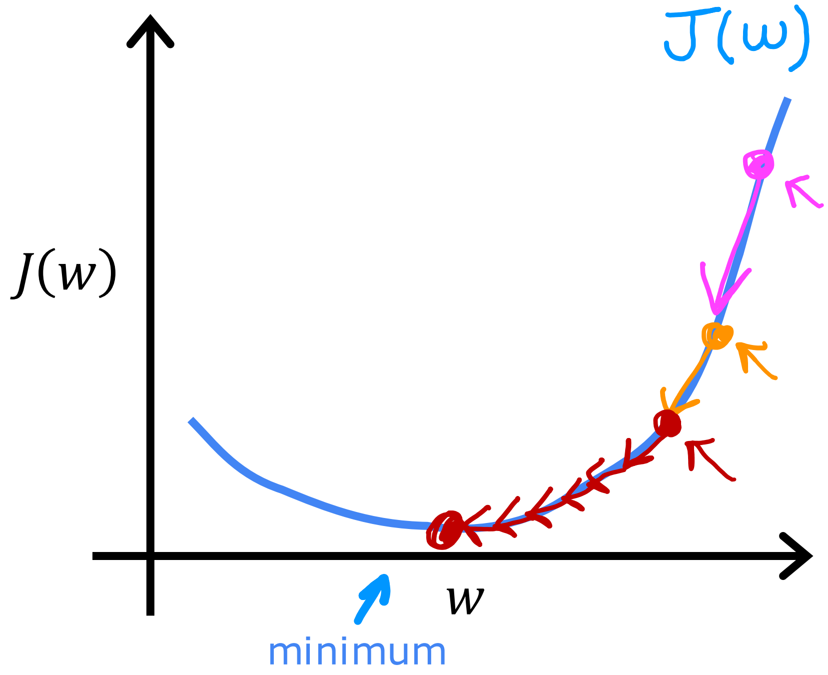
$$w = w$$

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

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

Gradient descent algorithm

repeat until convergence {

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

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

}

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

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

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

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

Gradient descent algorithm

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

repeat until convergence {

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

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

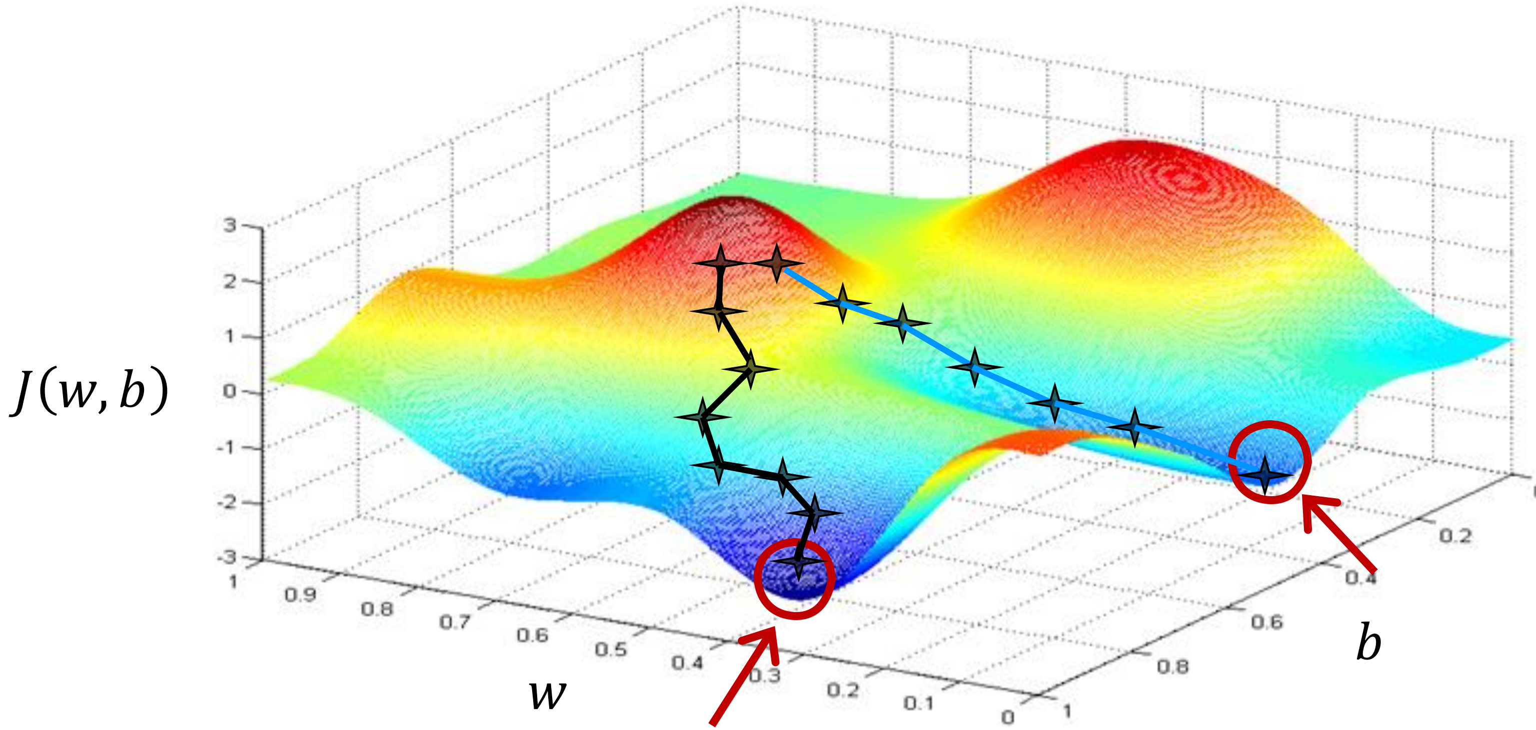
}

Update w and b simultaneously

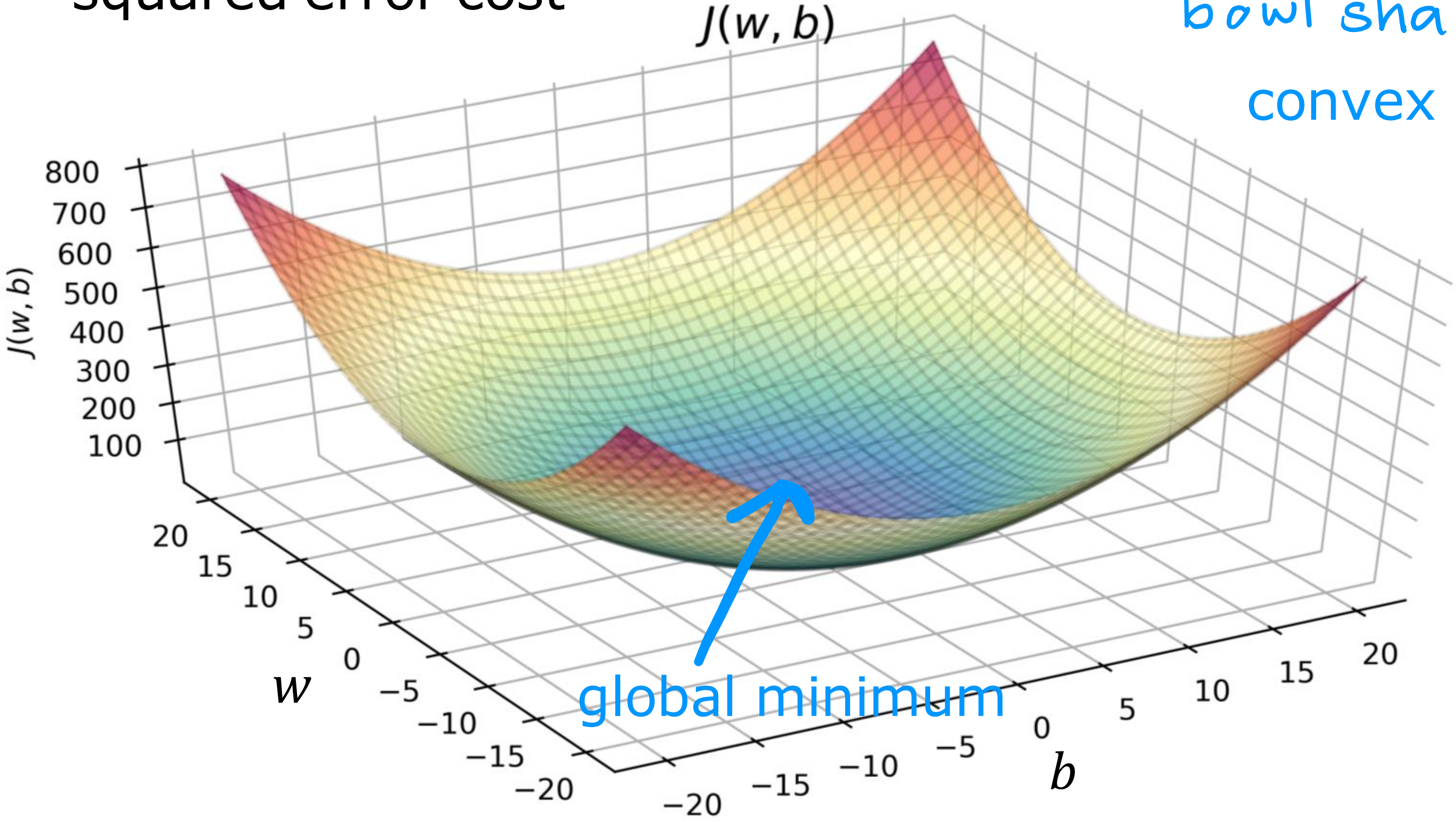
$\frac{\partial}{\partial b} J(w,b)$

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

More than one local minimum



squared error cost

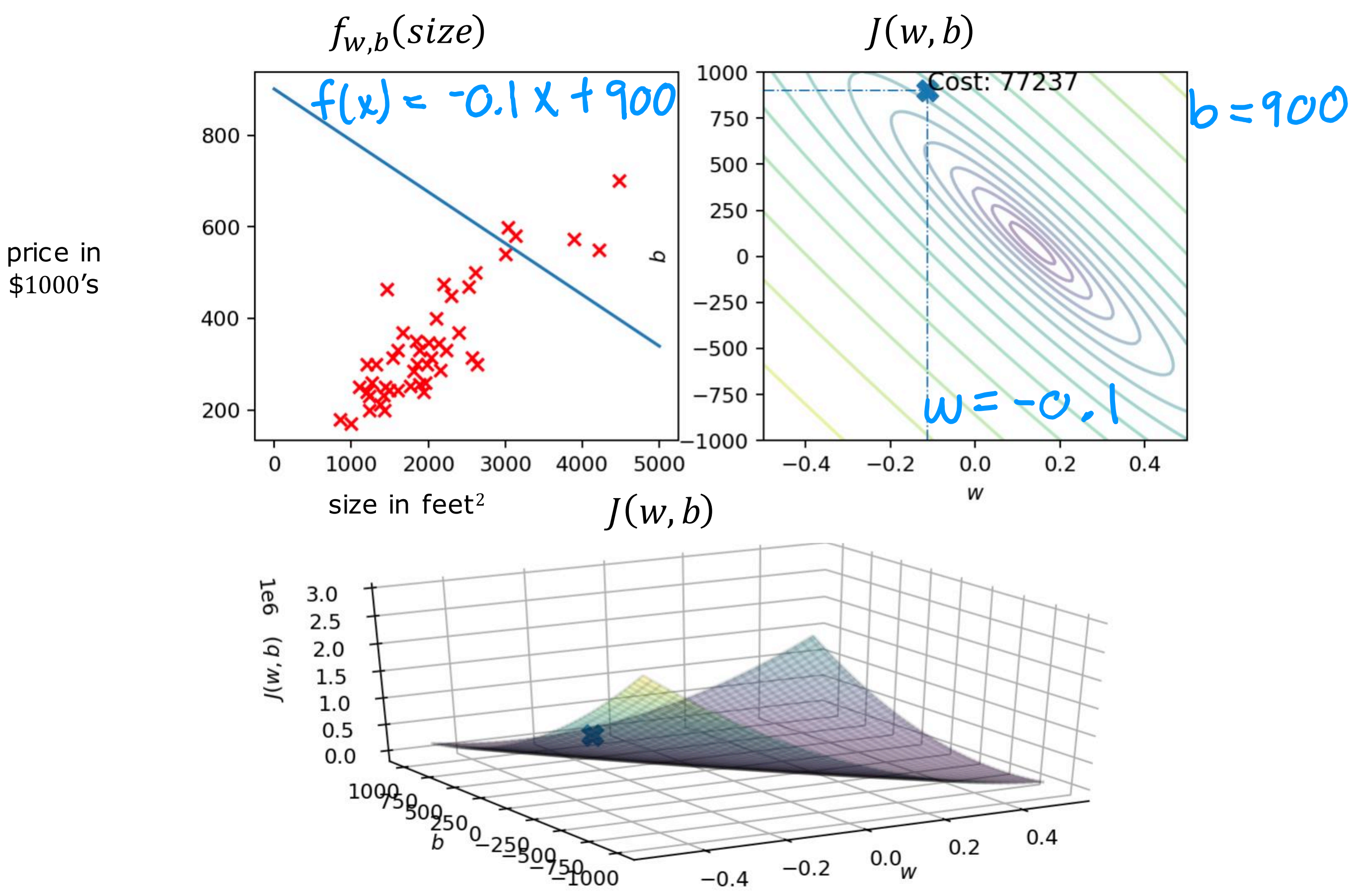


bowl shape

convex function

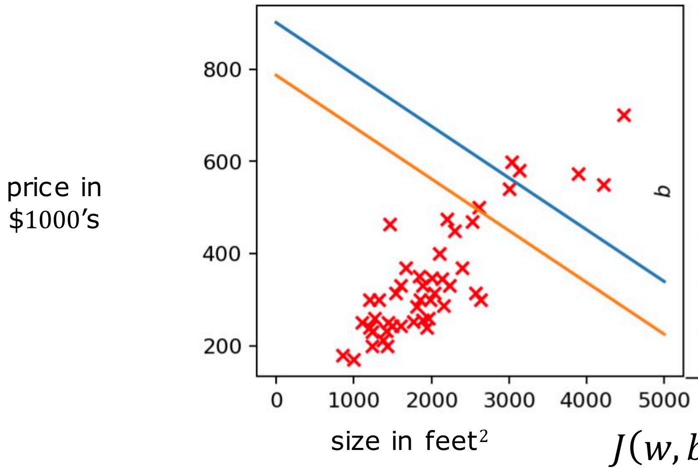
Training Linear Regression

Running Gradient Descent

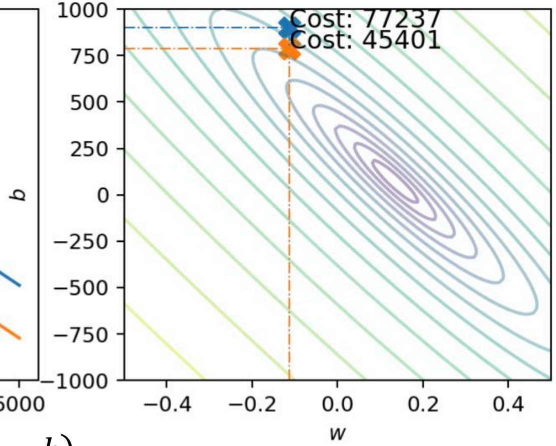


$f_{w,b}(size)$
f(x) = -0.1 x + 900
price in $1000's
size in feet²
$J(w,b)$
Cost: 77237
b = 900
w = -0.1
$J(w,b)$
1e6

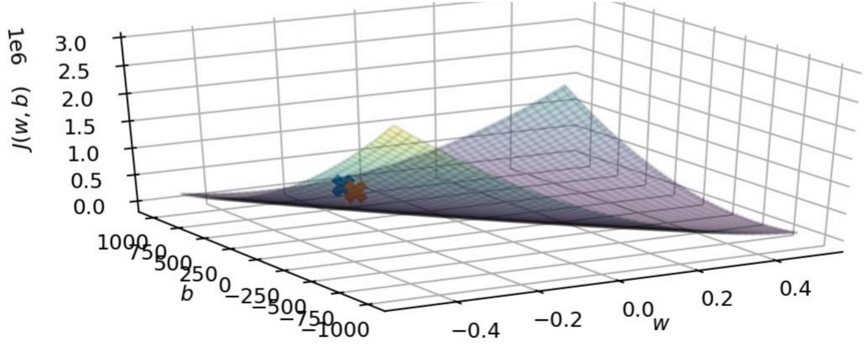
$f_{w,b}(size)$

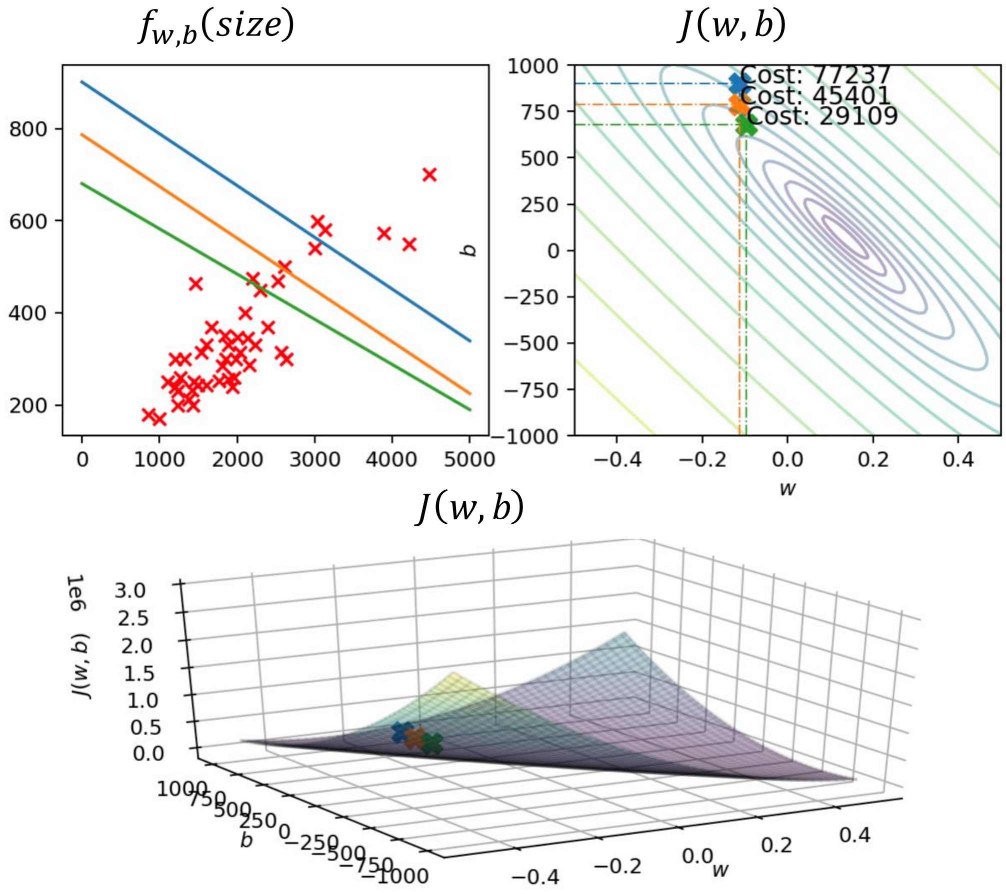


$J(w,b)$

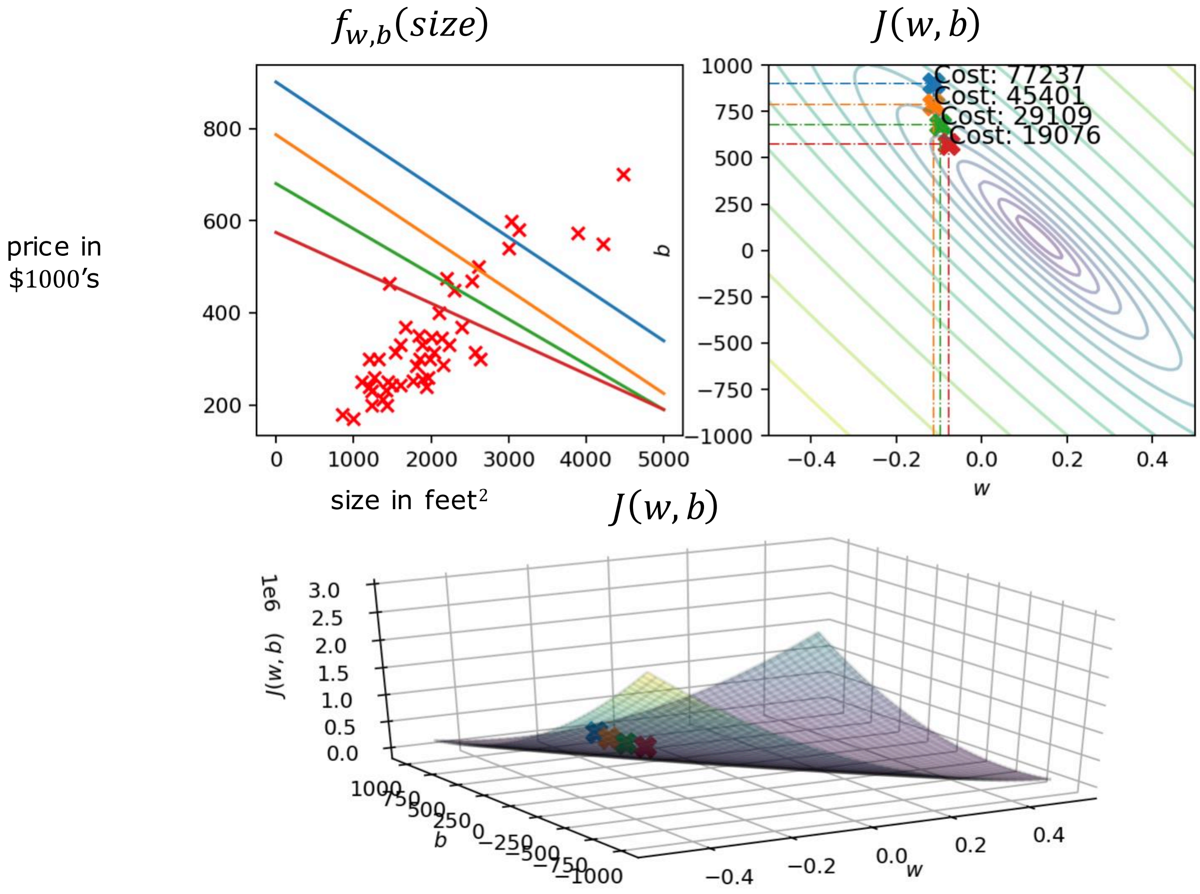


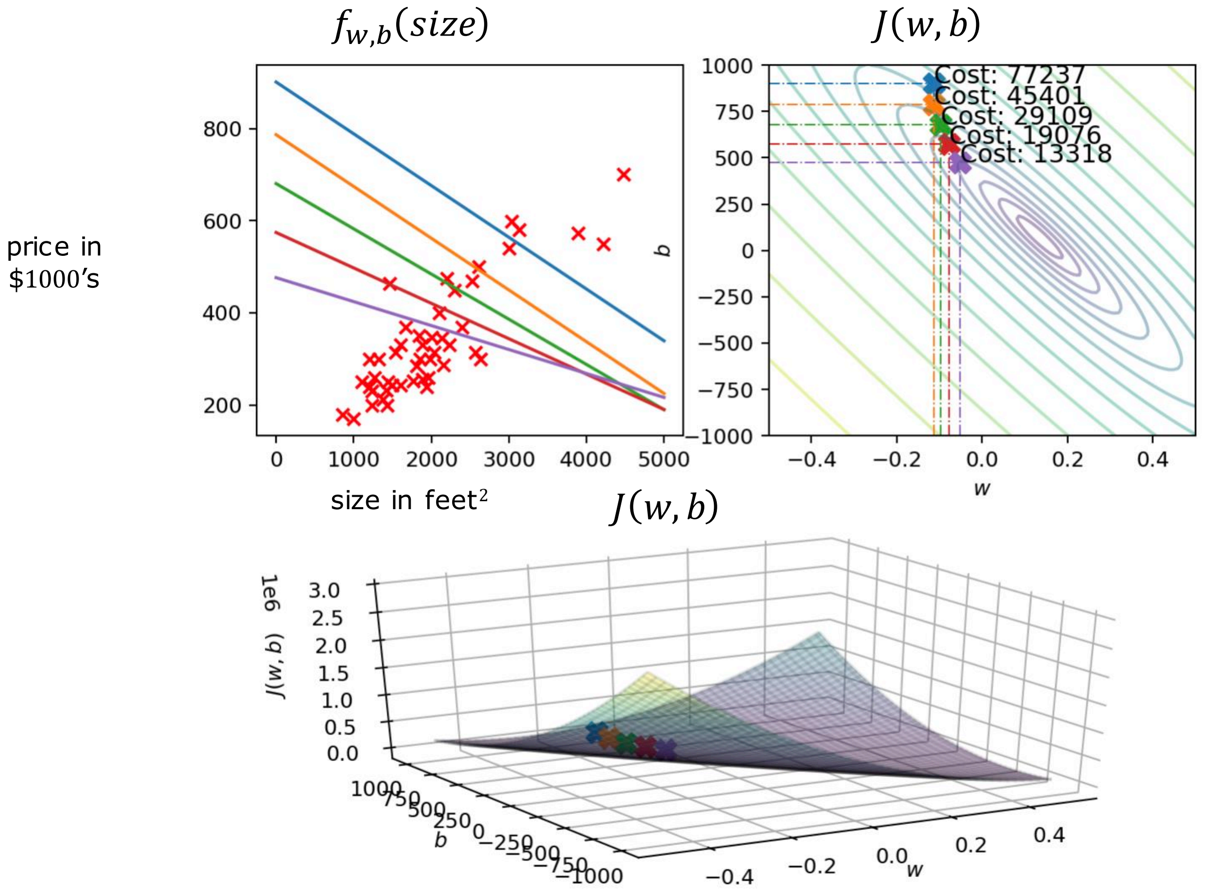
$J(w,b)$



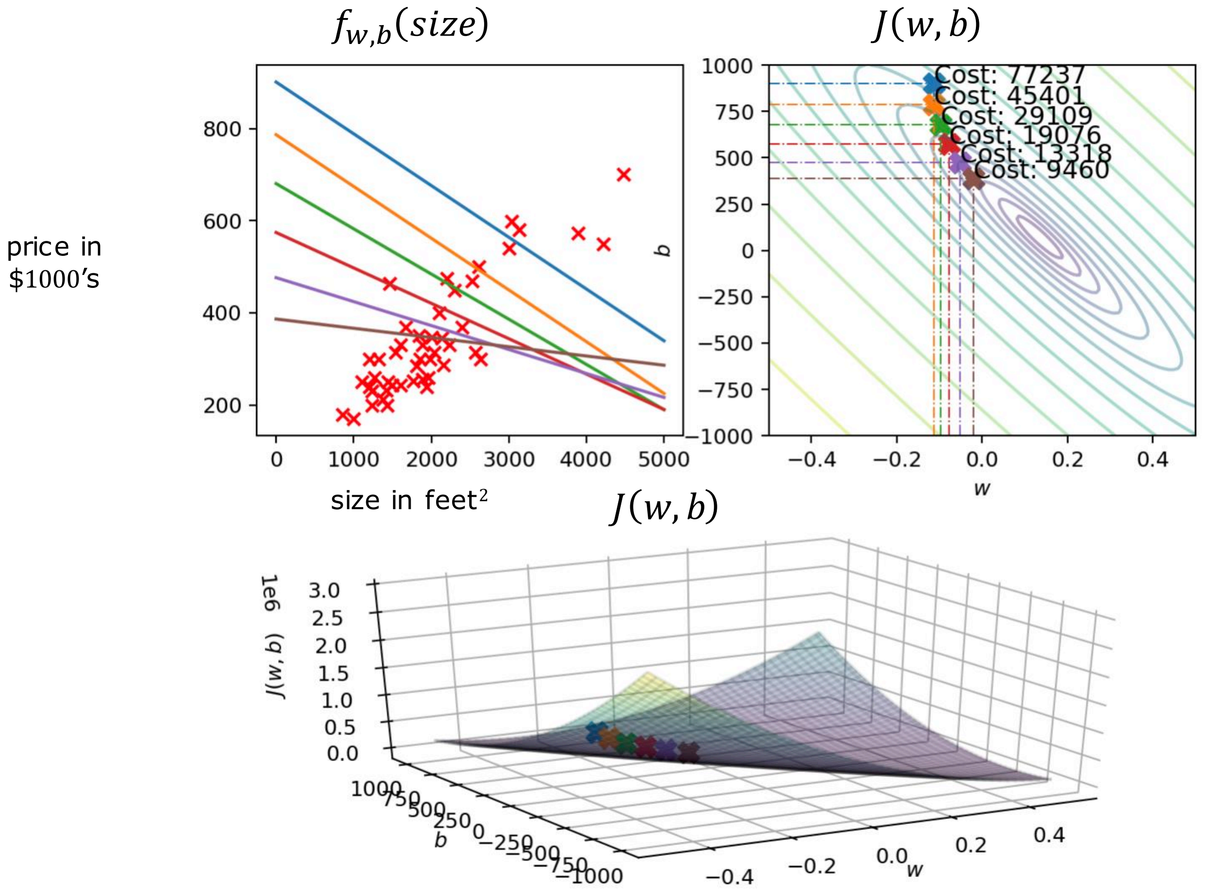


$f_{w,b}(size)$
$J(w,b)$
Cost: 77237
Cost: 45401
Cost: 29109
b
w
$J(w,b)$
1e6
J(w, b)
b
w

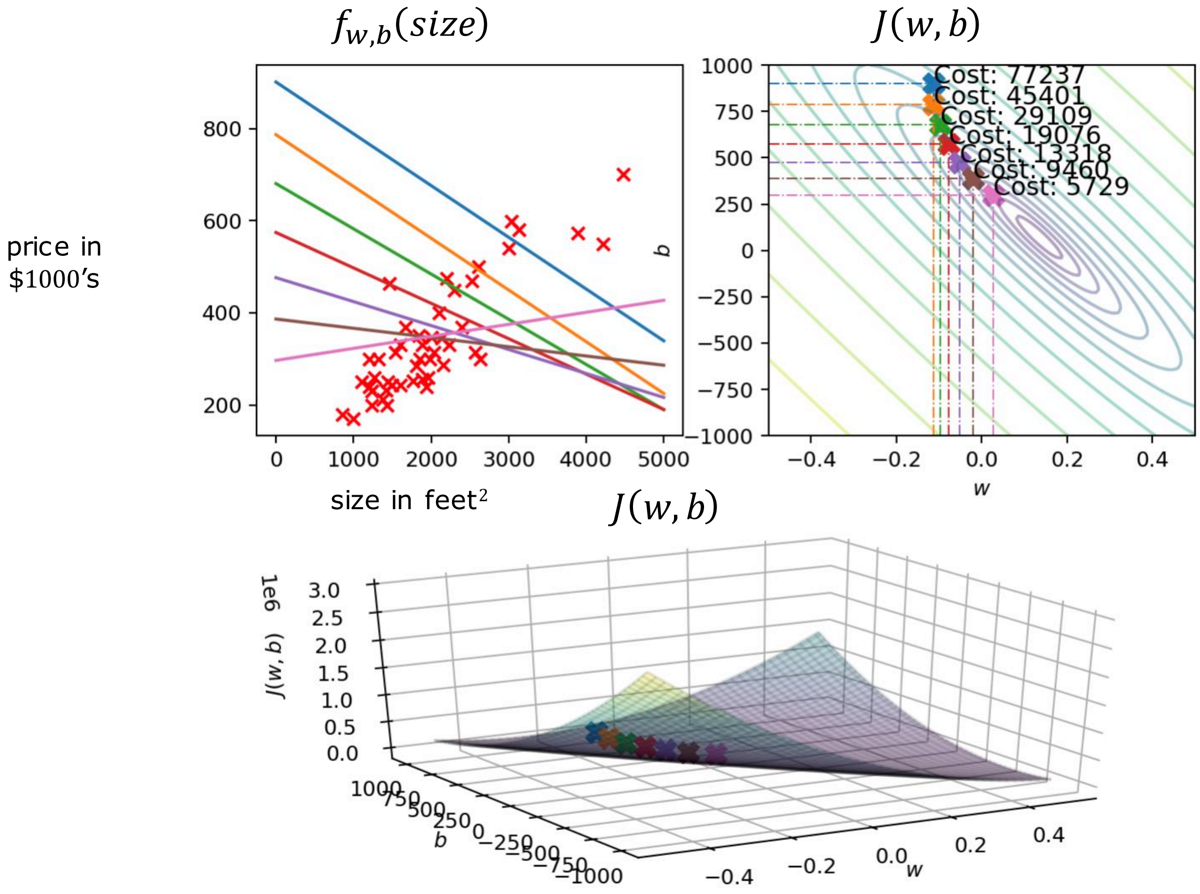

f_{w,b}(size)
J(w,b)
price in $1000's
size in feet²
Cost: 77237
Cost: 45401
Cost: 29109
Cost: 19076
b
w
J(w,b)
J(w, b)
1e6



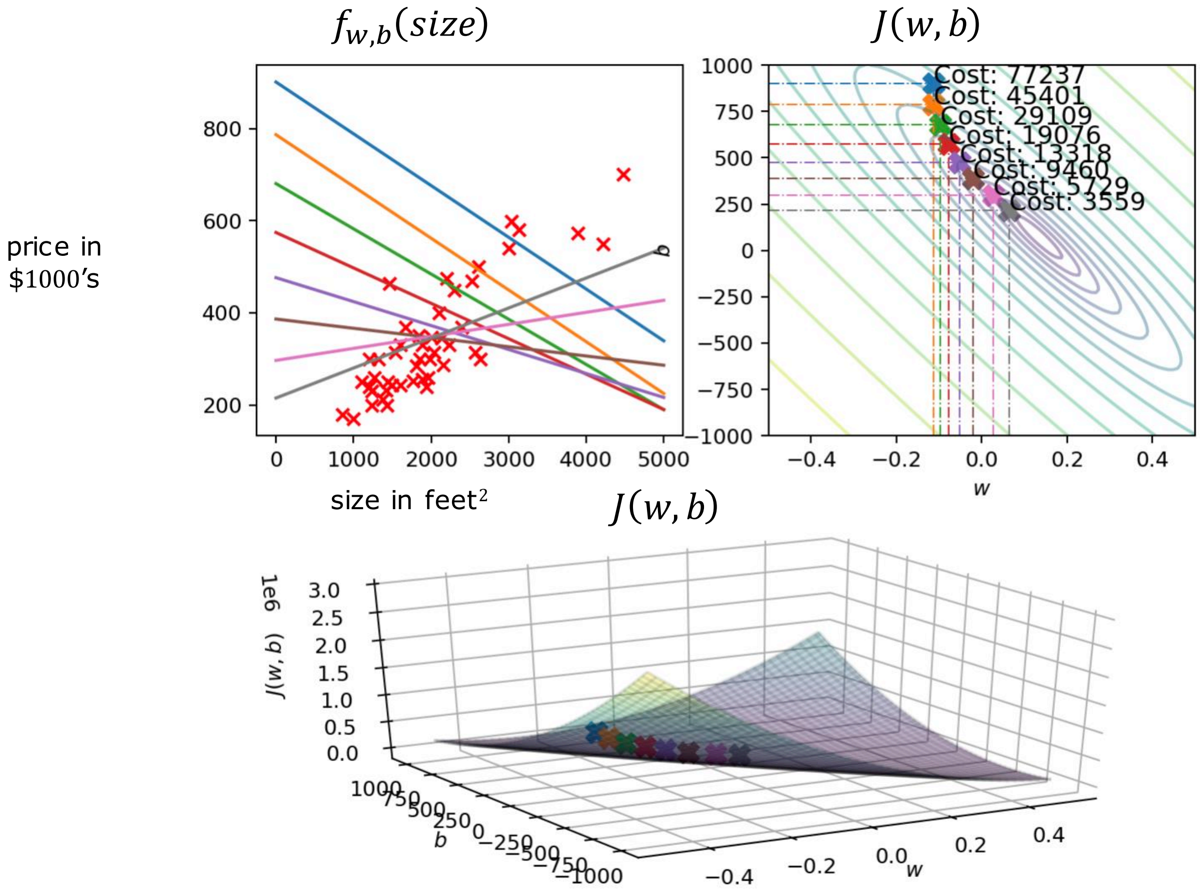
$f_{w,b}(size)$
price in $1000's
size in feet²
$J(w,b)$
Cost: 77237
Cost: 45401
Cost: 29109
Cost: 19076
Cost: 13318
b
w
$J(w,b)$
1e6
J(w, b)



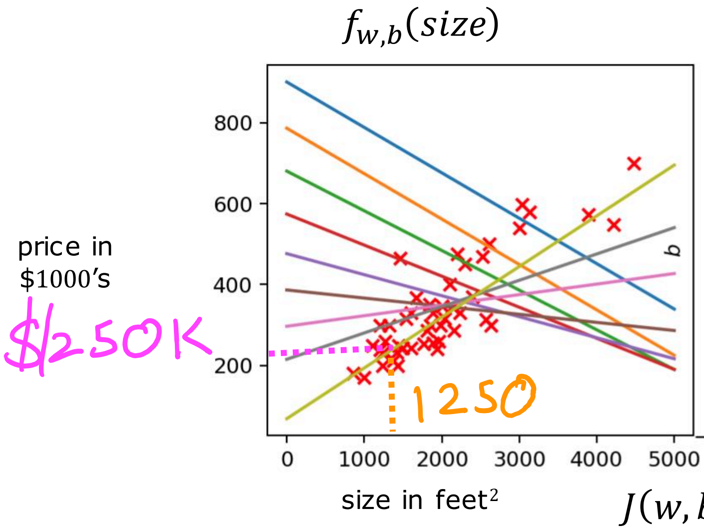
$f_{w,b}(size)$
price in $1000's
size in feet²
$J(w,b)$
Cost: 77237
Cost: 45401
Cost: 29109
Cost: 19076
Cost: 13318
Cost: 9460
$J(w,b)$

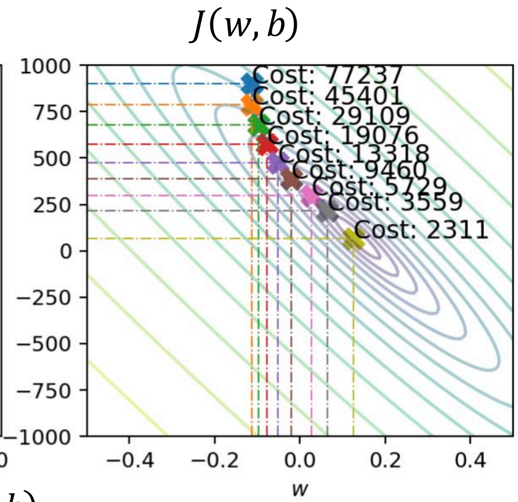


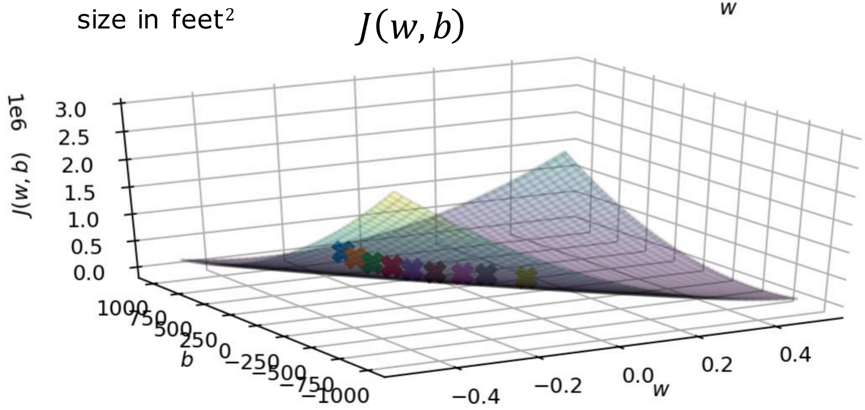
$f_{w,b}(size)$
$J(w,b)$
price in $1000's
size in feet^2
Cost: 77237
Cost: 45401
Cost: 29109
Cost: 19076
Cost: 13318
Cost: 9460
Cost: 5729
b
w
$J(w,b)$
1e6
J(w, b)



$f_{w,b}(size)$
price in
$1000's
size in feet^2
$J(w,b)$
Cost: 77237
Cost: 45401
Cost: 29109
Cost: 19076
Cost: 13318
Cost: 9460
Cost: 5729
Cost: 3559
w
b
$J(w,b)$
1e6
J(w, b)


$f_{w,b}(size)$
price in
$1000's
$250K
1250
size in feet²


$J(w,b)$
Cost: 77237
Cost: 45401
Cost: 29109
Cost: 19076
Cost: 13318
Cost: 9460
Cost: 5729
Cost: 3559
Cost: 2311
b
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

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