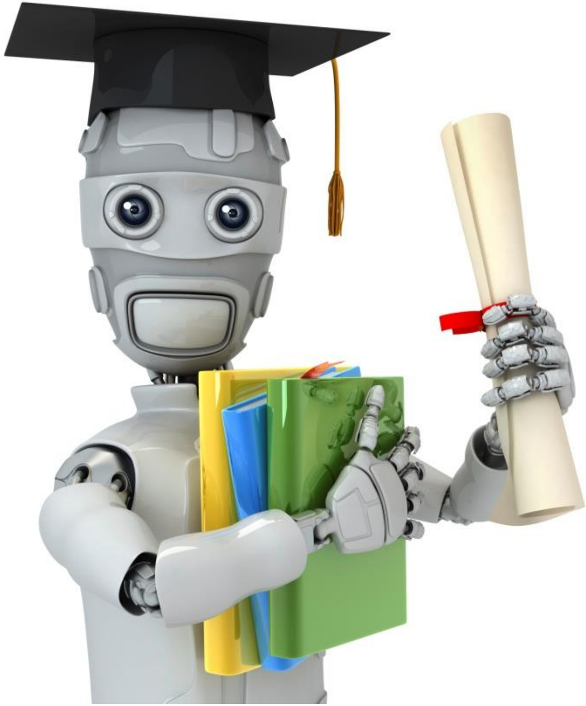
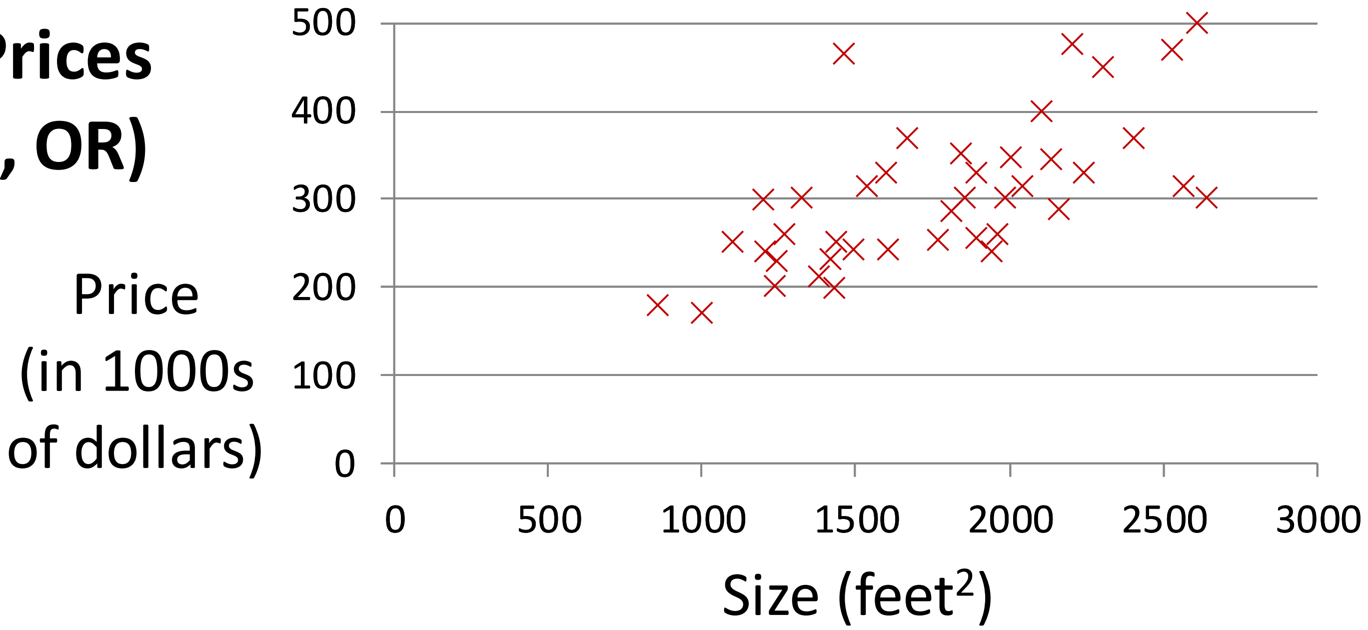


Machine Learning

Linear regression with one variable

Andrew Ng
Edited Bart Vanrumste

Housing Prices (Portland, OR)



<u>Supervised Learning</u>

Given the "right answer" for each example in the data.

<u>Regression Problem</u>

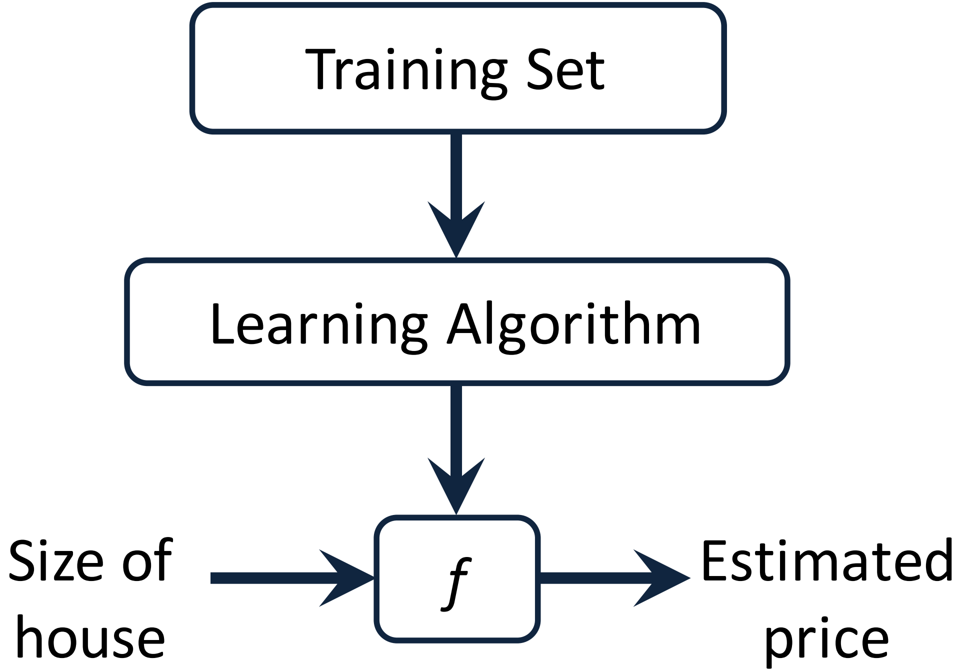
Predict real-valued output

Training set of housing prices (Portland, OR)

	Size in feet^2 (x)	Price ($) in 1000's (y)
(1)	2104	460
(2)	1416	232
(3)	1534	315
(4)	852	178
⋮	…	…

Notation:

m = Number of training examples
x's = "input" variable / features
y's = "output" variable / "target" variable



How do we represent *h* ?

Linear regression with one variable.
Univariate linear regression.

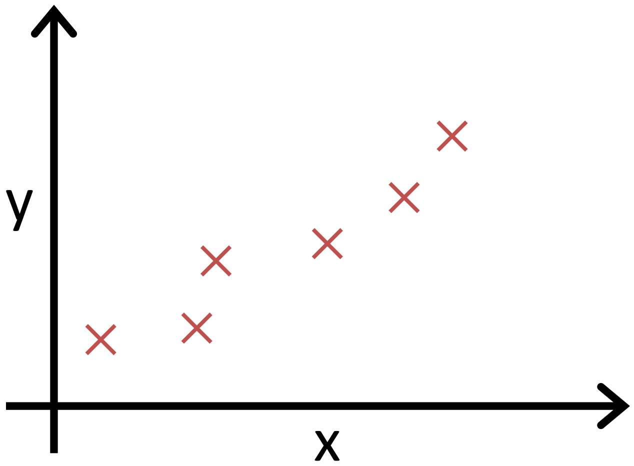
Linear regression with one variable

Cost function

Training Set

Size in feet^2 (x)	**Price ($) in 1000's (y)**
2104	460
1416	232
1534	315
852	178
…	…

How to choose the parameters



Idea: Choose w and b so that
f is close to y for our
training examples (x, y)

Linear regression with one variable

Cost function intuition

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

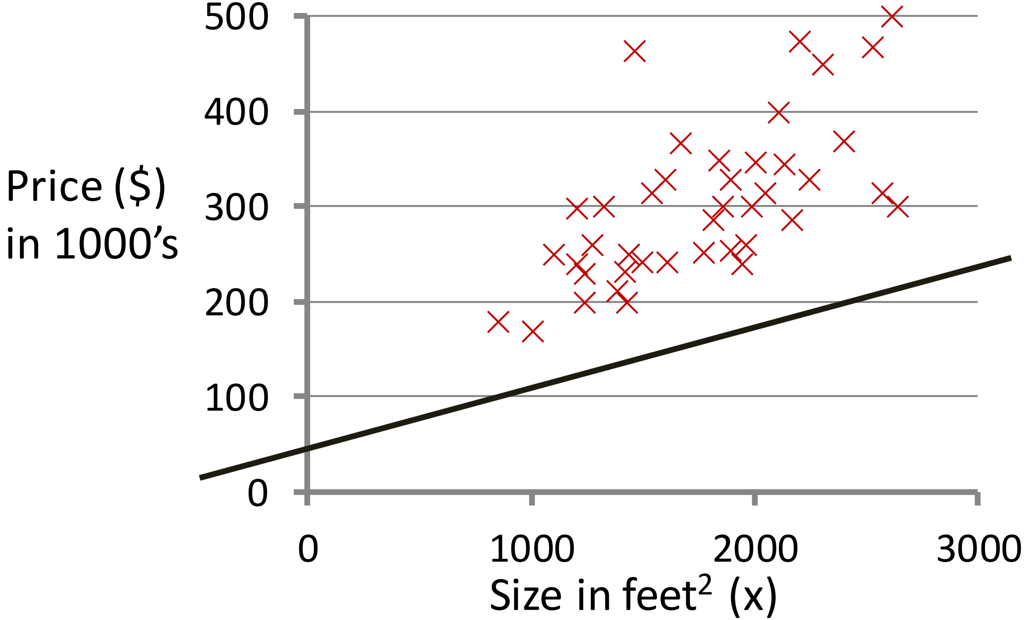
Parameters: w, b

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

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

$f_{w,b}(x)$

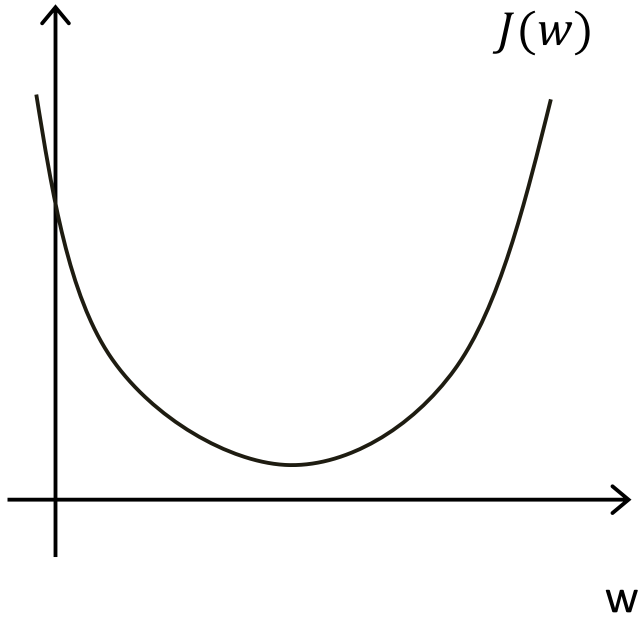
(function of x (when w and b fixed))

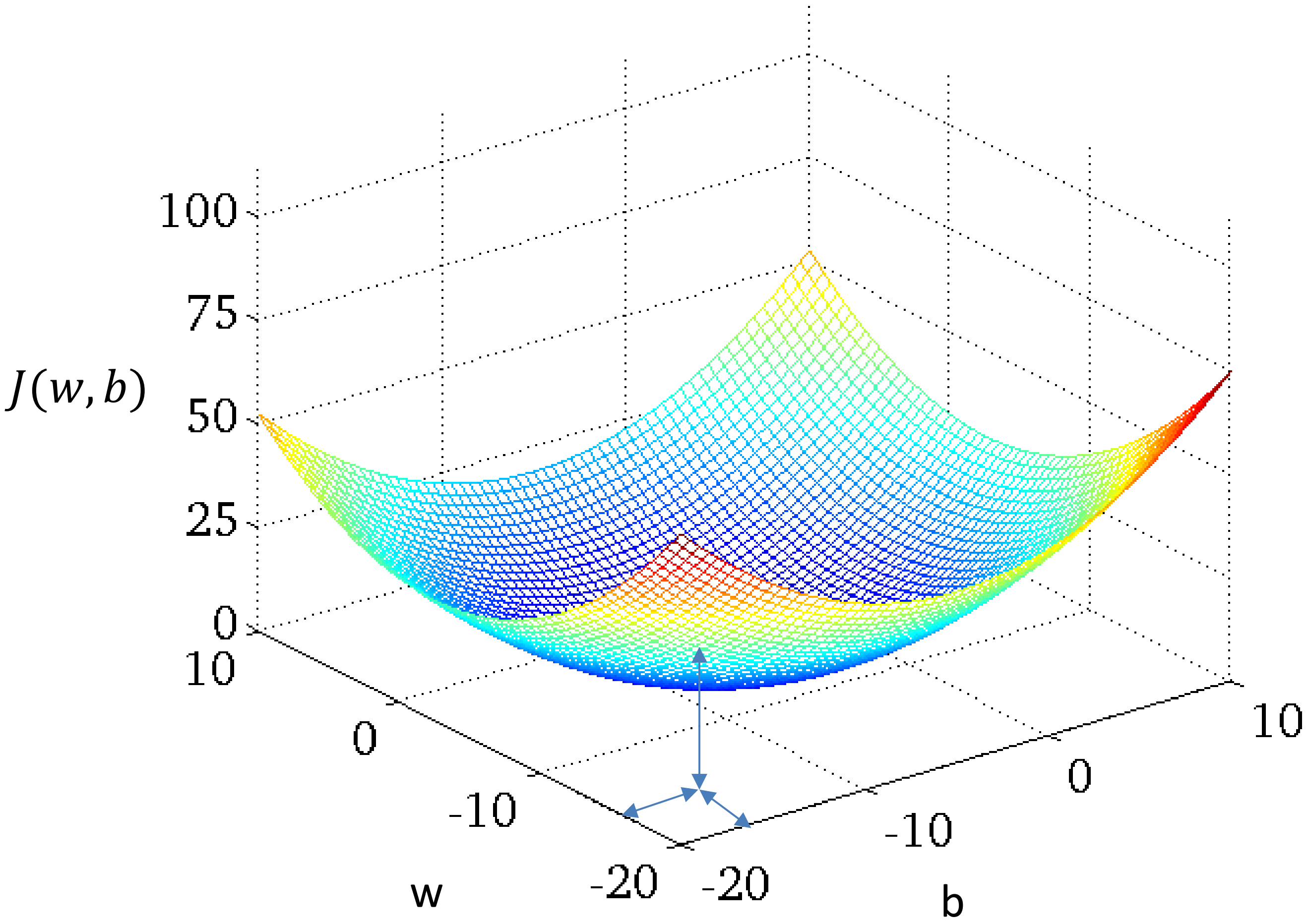


$$f_{w,b}(x) = 50 + 0.06x$$

$J(w,b)$

(function of the parameters w, b)

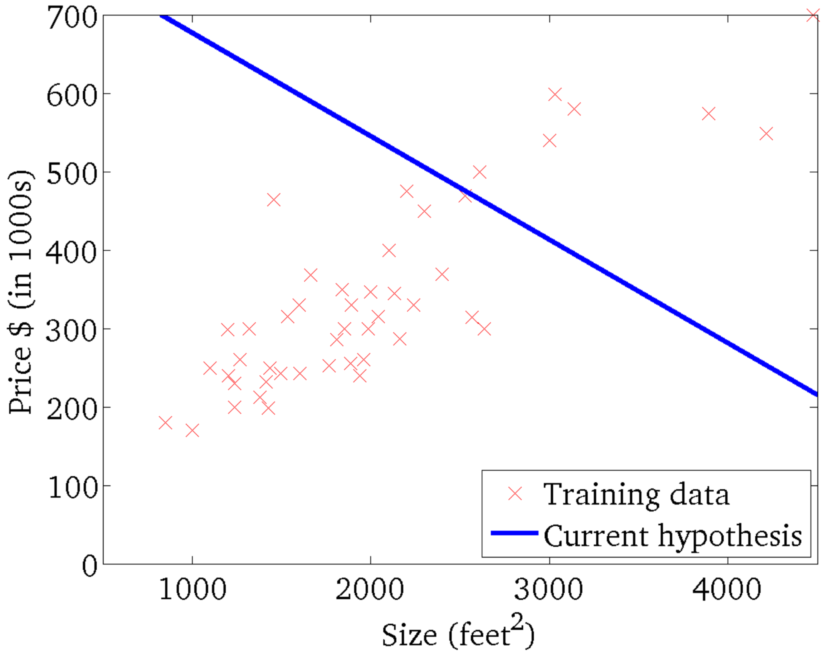




100
75
J(w, b)
50
25
0
10
0
-10
-20
w
-20
-10
0
10
b

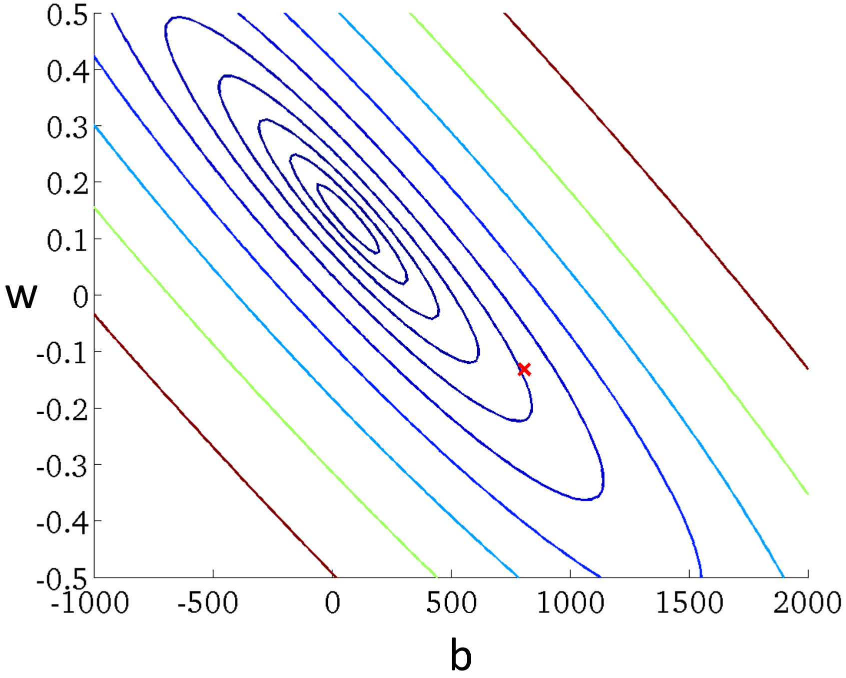
$f_{w,b}(x)$

(function of x (when w and b fixed))



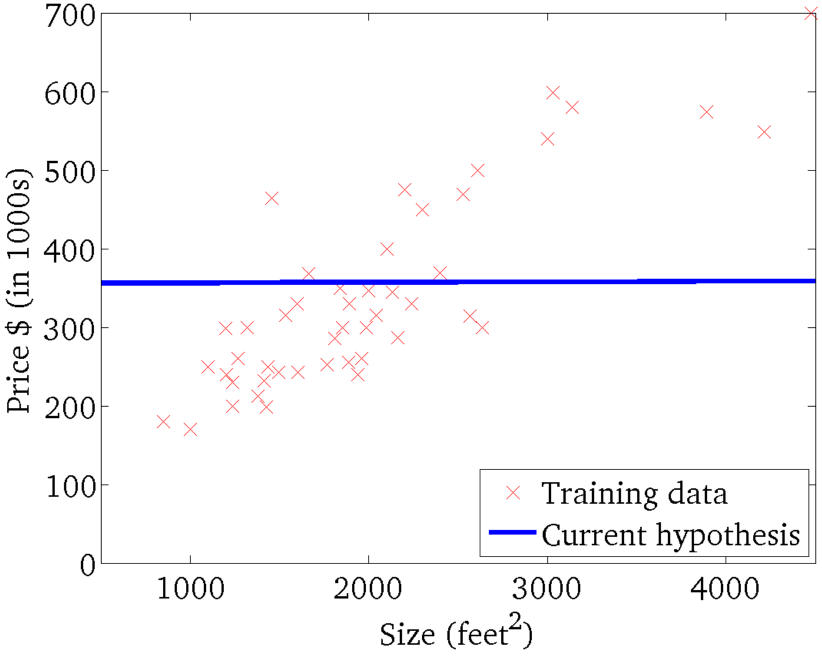
$J(w,b)$

(function of the parameters w, b)



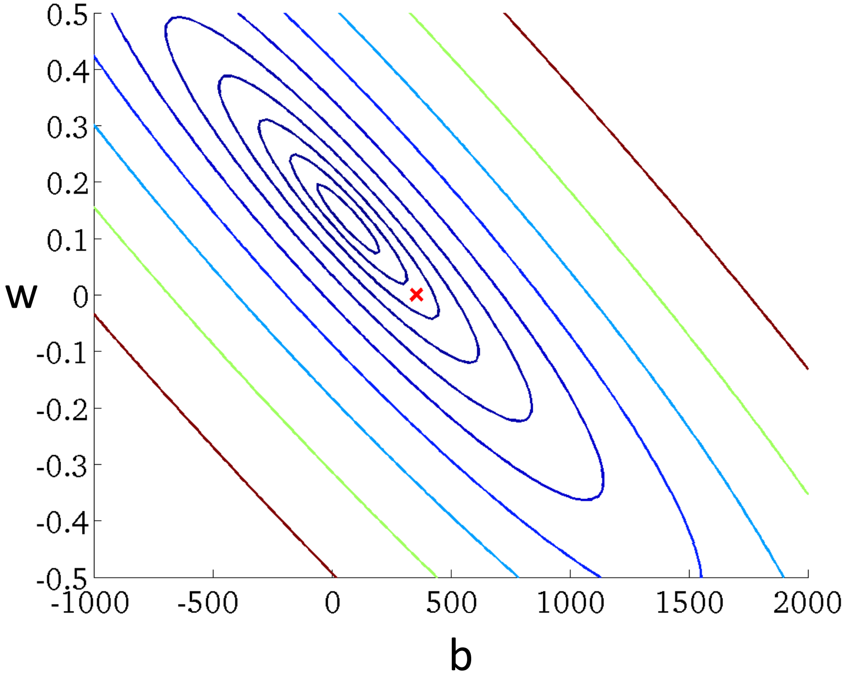
$f_{w,b}(x)$

(function of x (when w and b fixed))



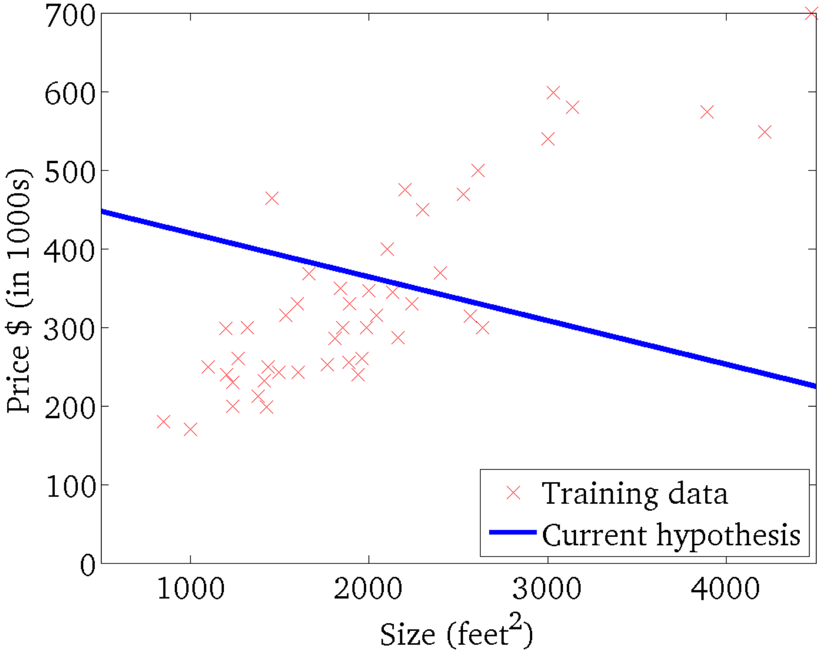
$J(w,b)$

(function of the parameters w, b)



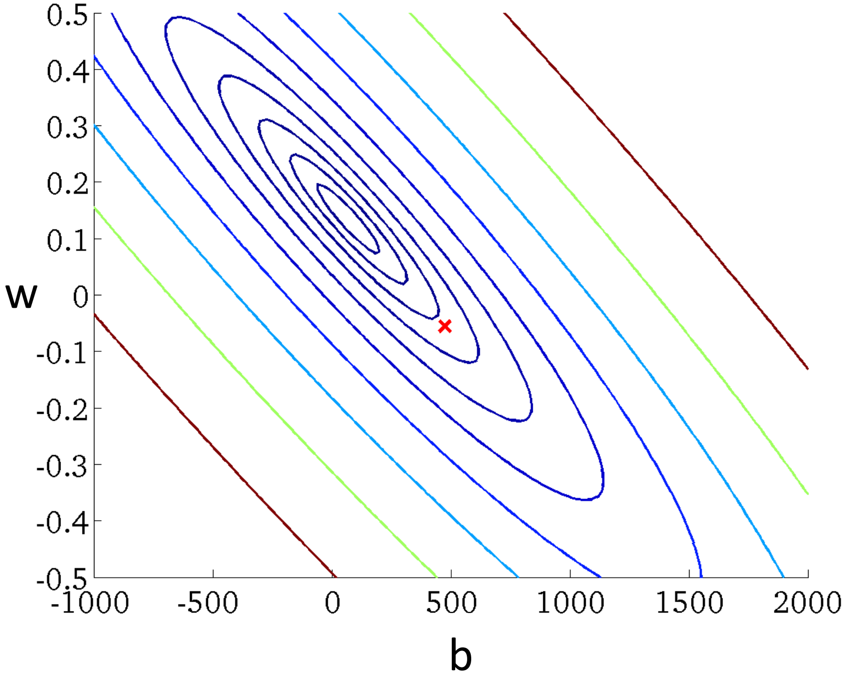
$f_{w,b}(x)$

(function of x (when w and b fixed))



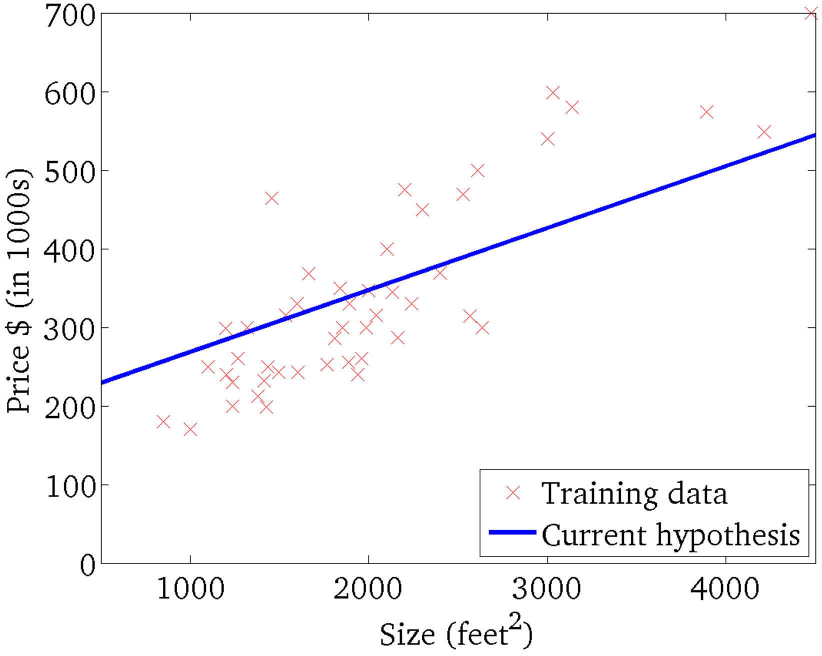
$J(w,b)$

(function of the parameters w, b)



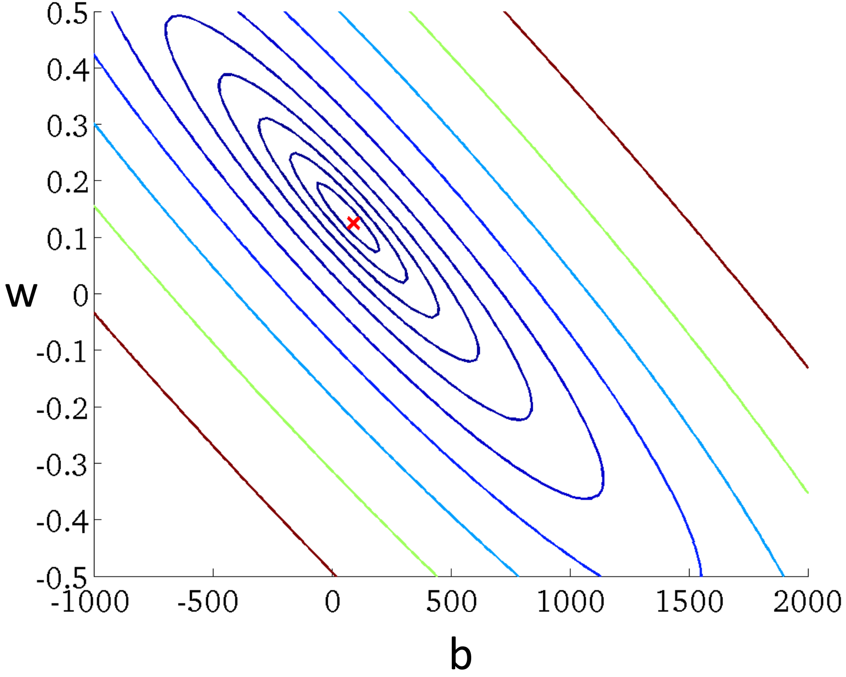
$f_{w,b}(x)$

(function of x (when w and b fixed))



$J(w,b)$

(function of the parameters w, b)



Linear regression with one variable

Gradient descent

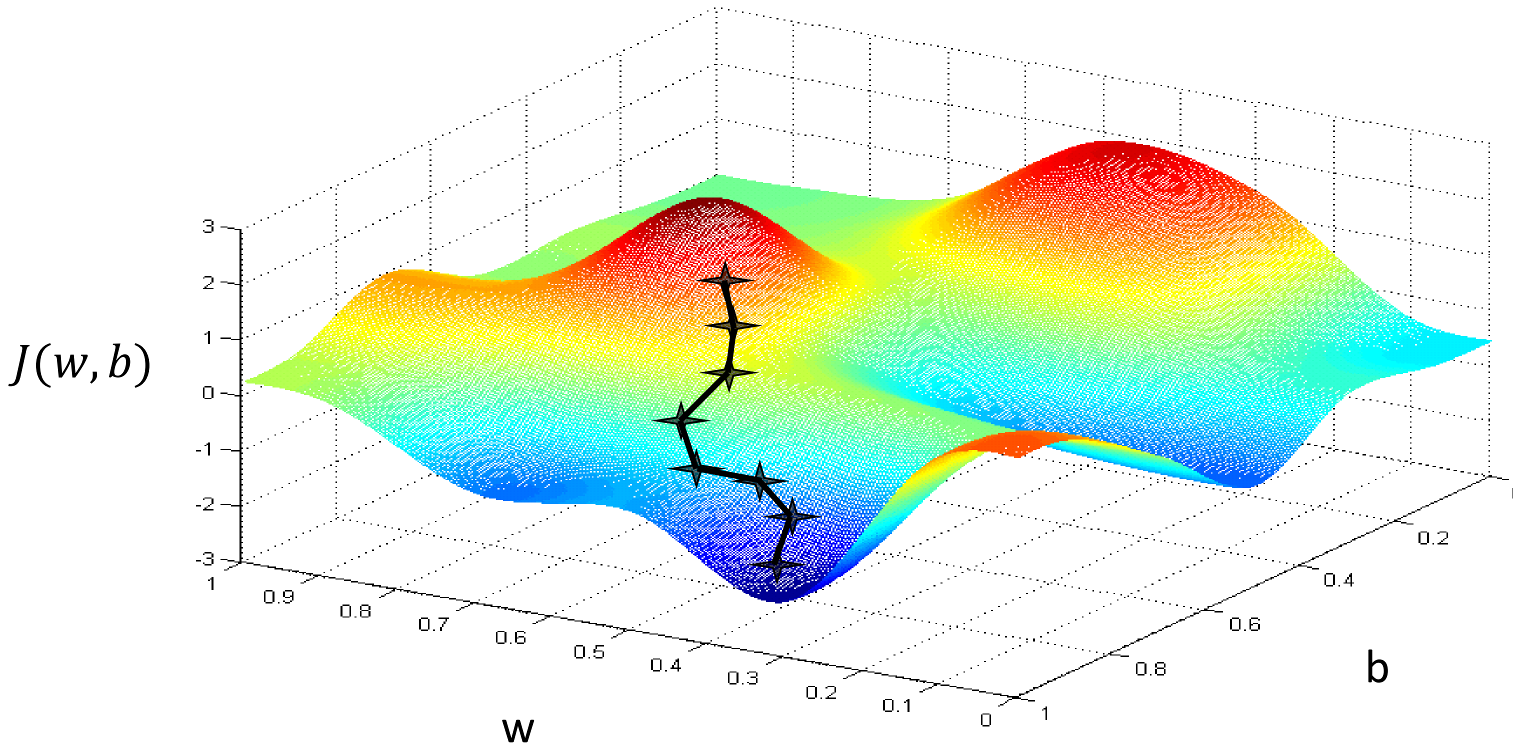
Have some function

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

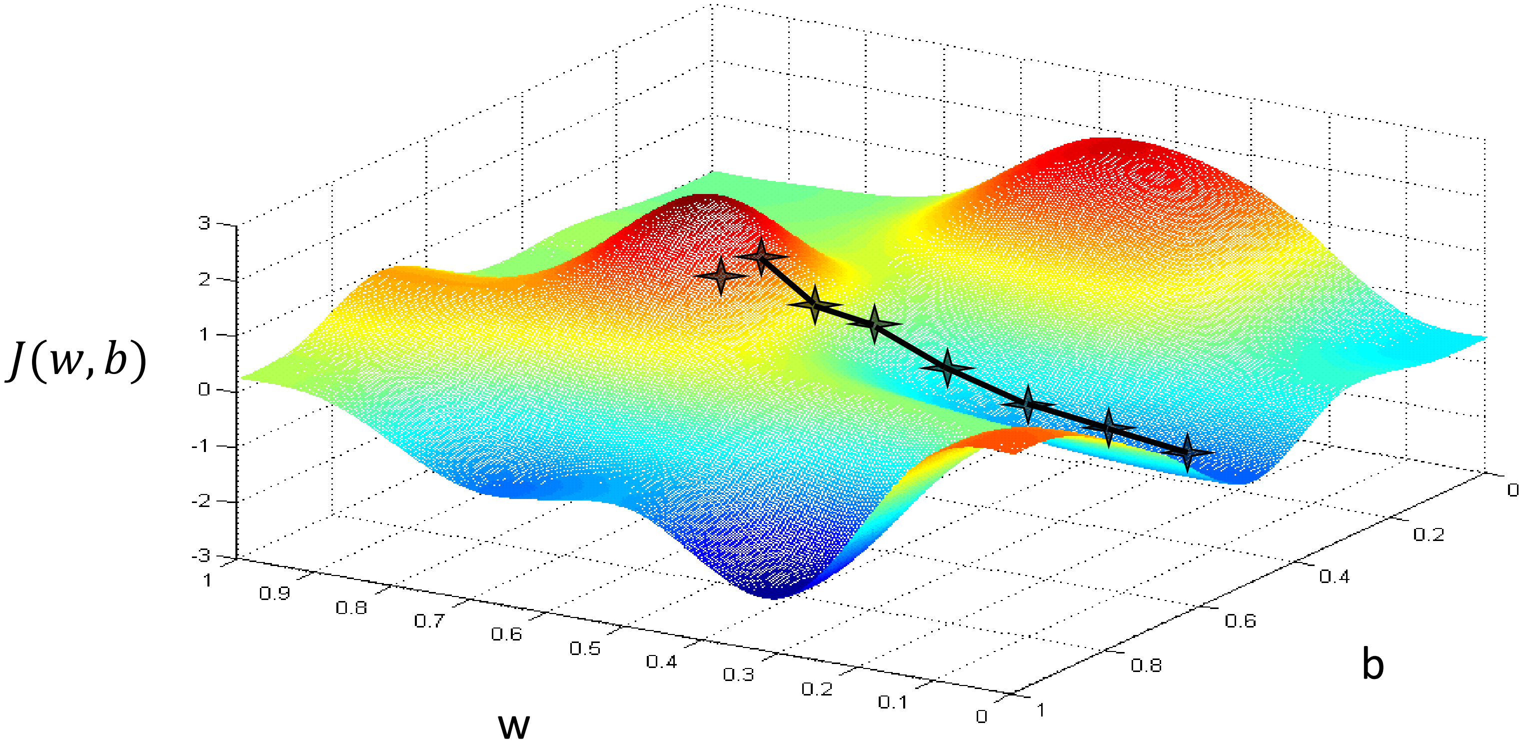
Outline:

- Start with some w,b
- Keep changing w, b to reduce $J(w,b)$

  until we hopefully end up at a minimum



$J(w,b)$
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



$J(w,b)$
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

Gradient descent algorithm

Correct: Simultaneous update

Incorrect:

Linear regression with one variable

Gradient descent intuition

Gradient descent algorithm

Repeat until convergence

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

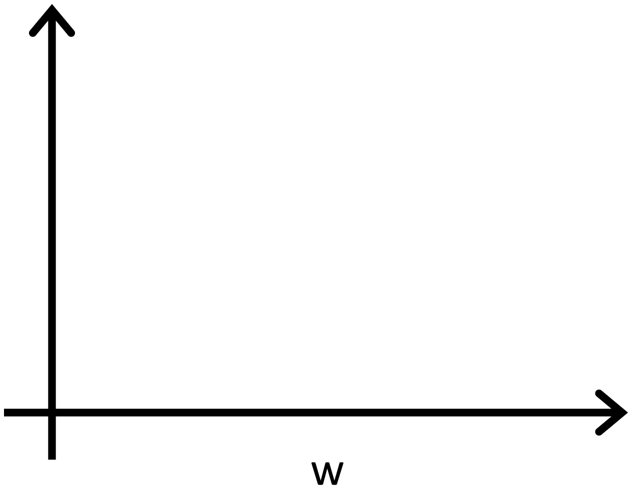
$$temp\_b = b - \alpha \frac{\partial}{\partial b} J(w, b)$$

$$b = temp\_b$$

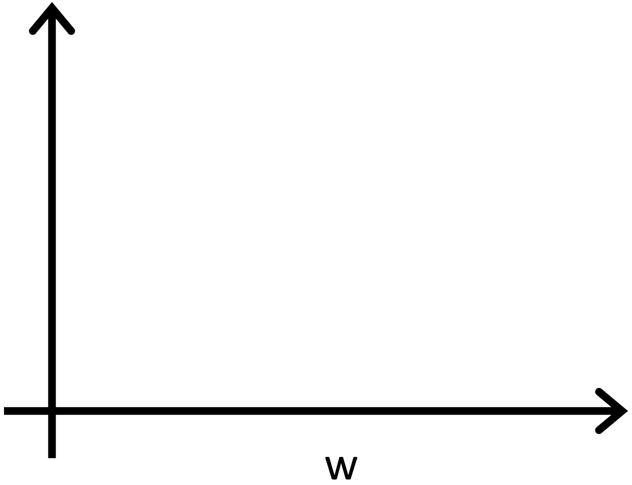
$$w = temp\_w$$

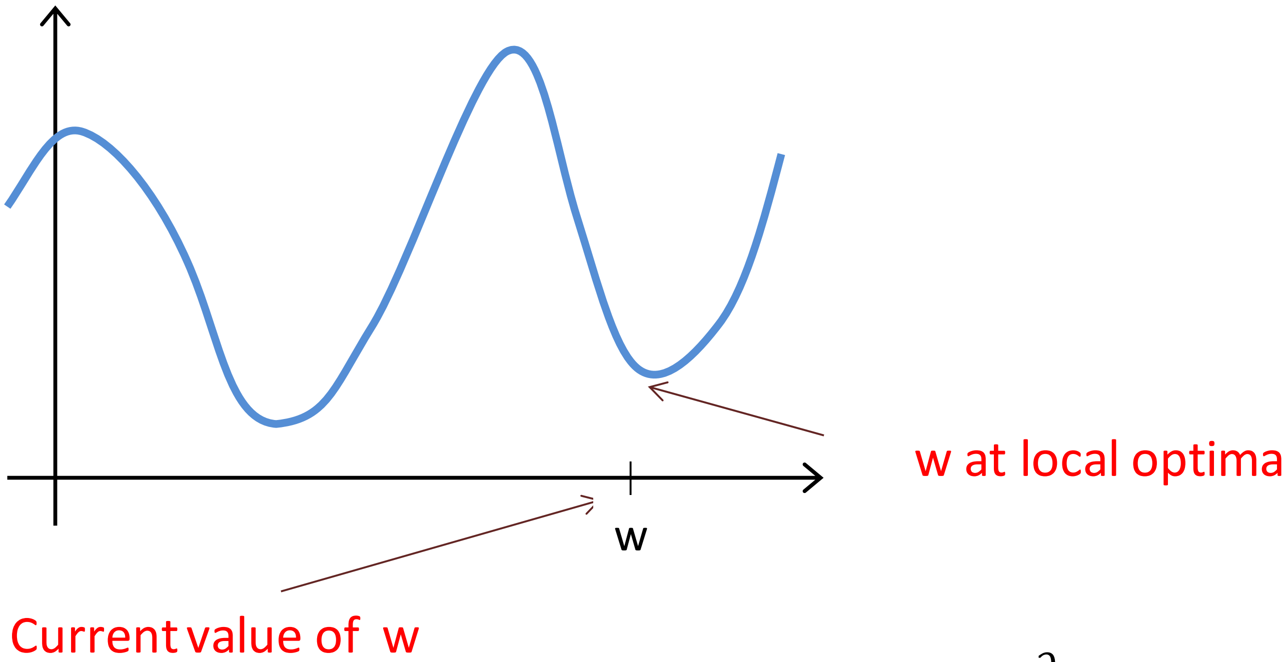
$$w = w - \alpha \frac{\partial}{\partial w} J(w)$$

If α is too small, gradient descent can be slow.



If α is too large, gradient descent can overshoot the minimum. It may fail to converge, or even diverge.



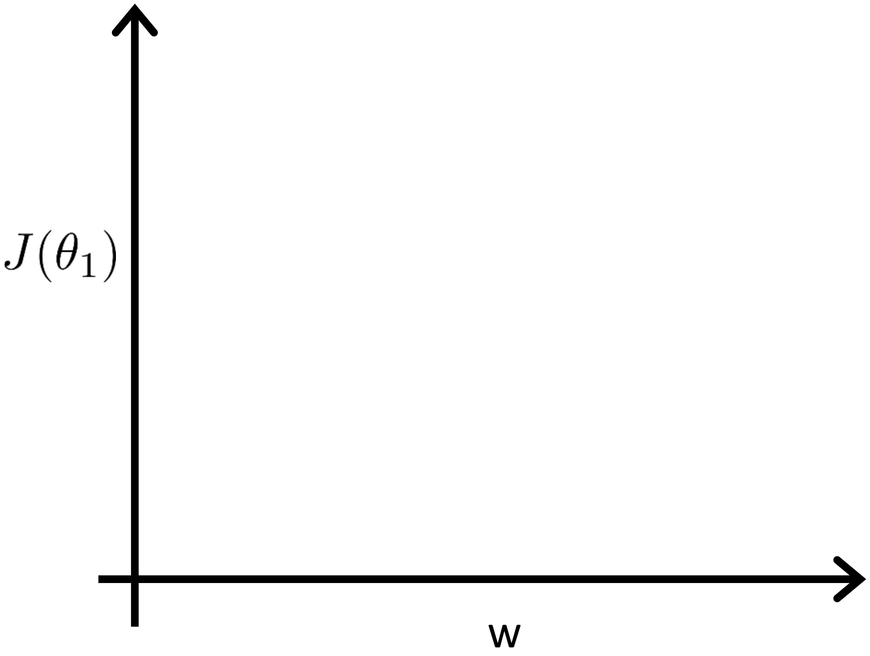


$$w = w - \alpha \frac{\partial}{\partial w} J(w)$$

Gradient descent can converge to a local minimum, even with the learning rate α fixed.

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

As we approach a local minimum, gradient descent will automatically take smaller steps. So, no need to decrease α over time.



Linear regression with one variable

Gradient descent for linear regression

Repeat until convergence

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

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

$$b = temp\_b$$

$$w = temp\_w$$

Linear Regression Model

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

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

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

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

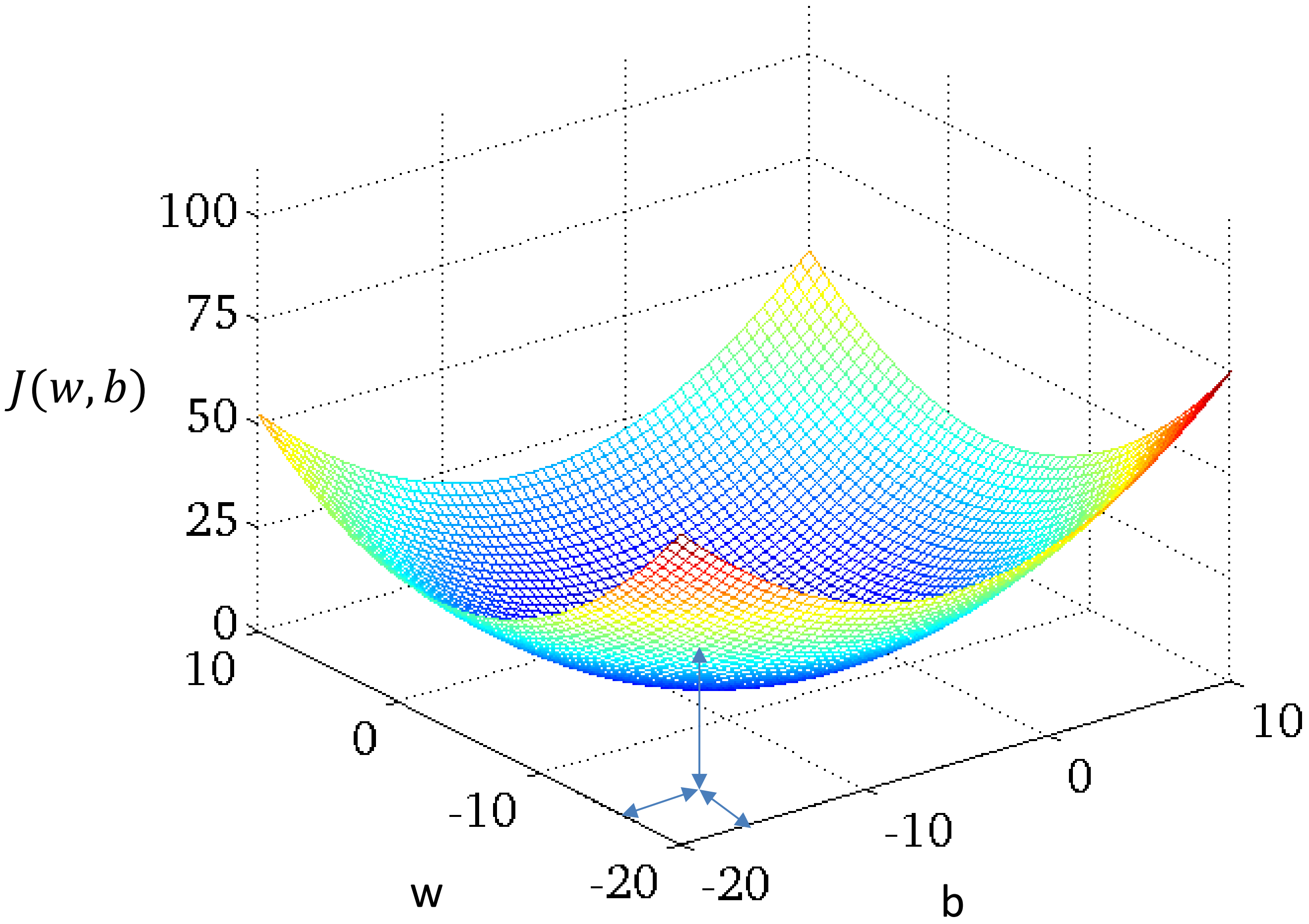
Gradient descent algorithm

Repeat until convergence

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

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

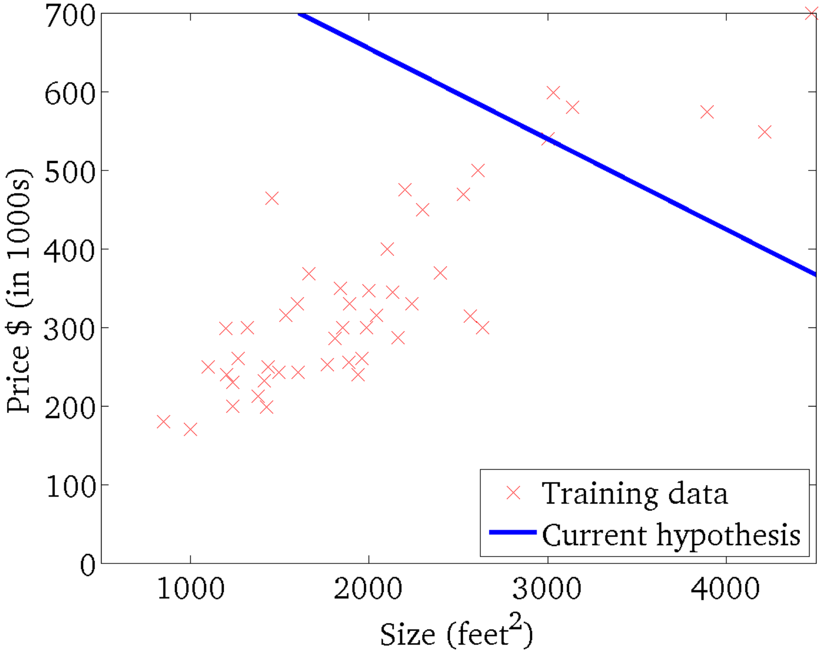
$$b = temp\_b$$
$$w = temp\_w$$



$J(w,b)$
100
75
50
25
0
10
0
-10
-20
w
-20
-10
0
10
b

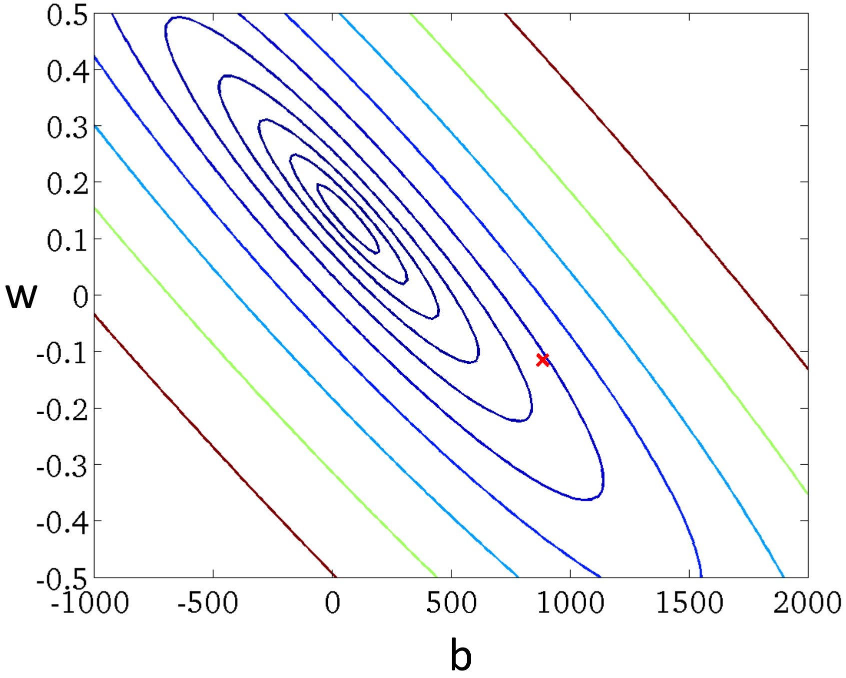
$f_{w,b}(x)$

(function of x (when w and b fixed))



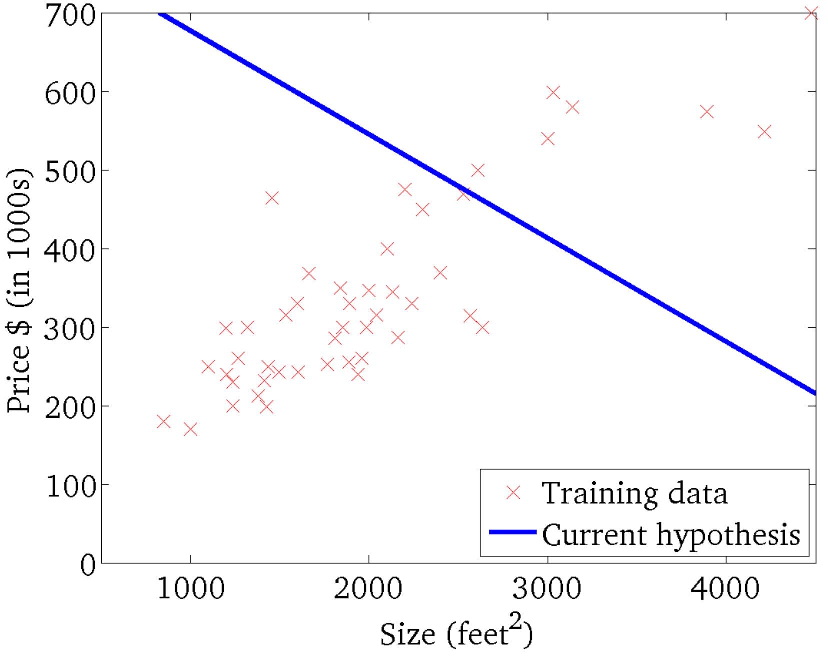
$J(w,b)$

(function of the parameters w, b)



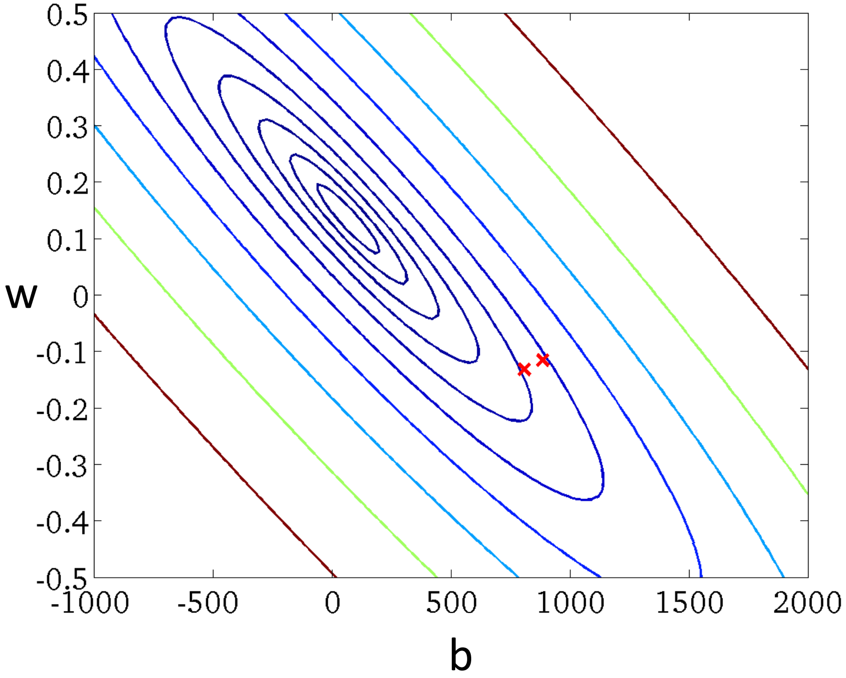
$f_{w,b}(x)$

(function of x (when w and b fixed))



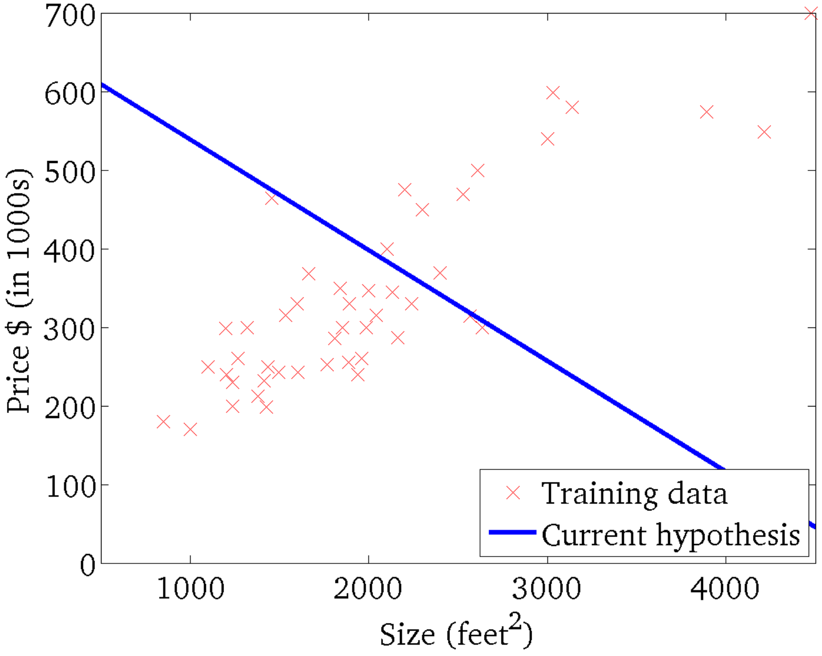
$J(w,b)$

(function of the parameters w, b)



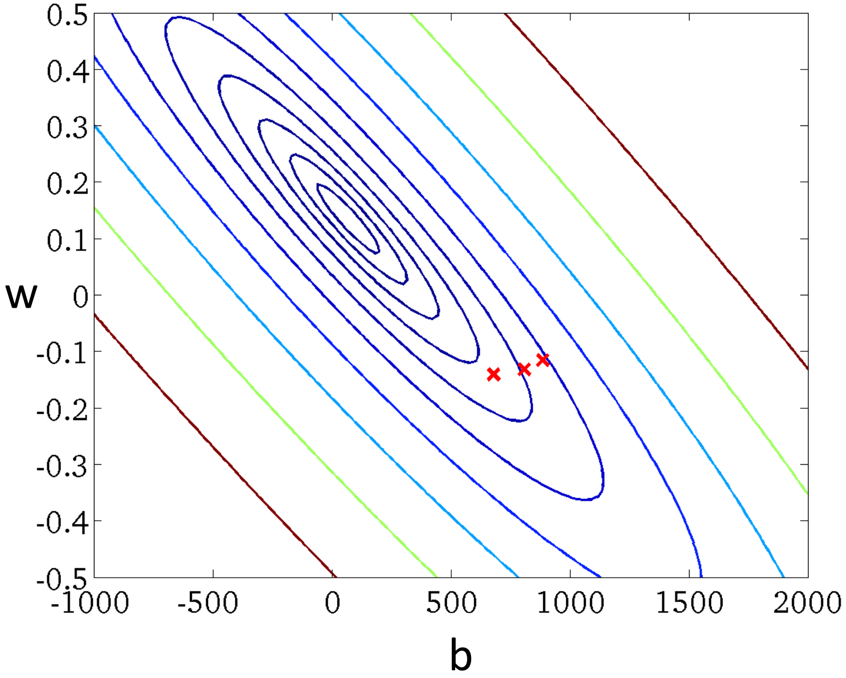
$f_{w,b}(x)$

(function of x (when w and b fixed))



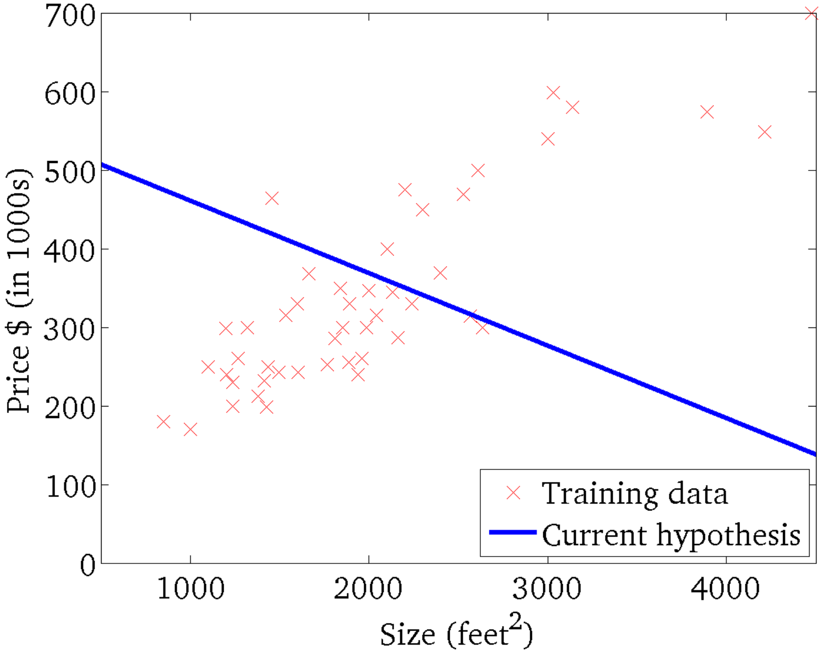
$J(w, b)$

(function of the parameters w, b)



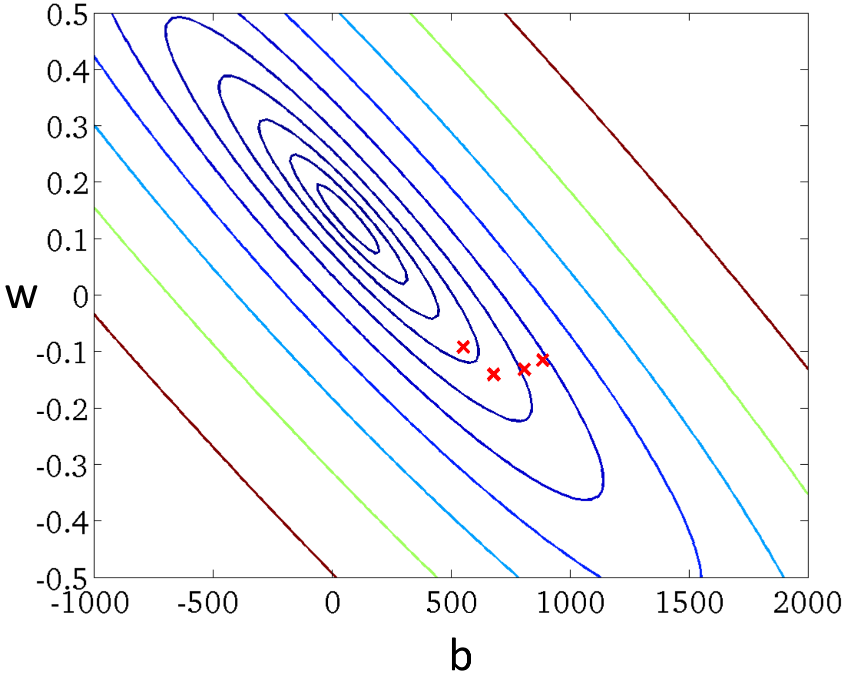
$f_{w,b}(x)$

(function of x (when w and b fixed))



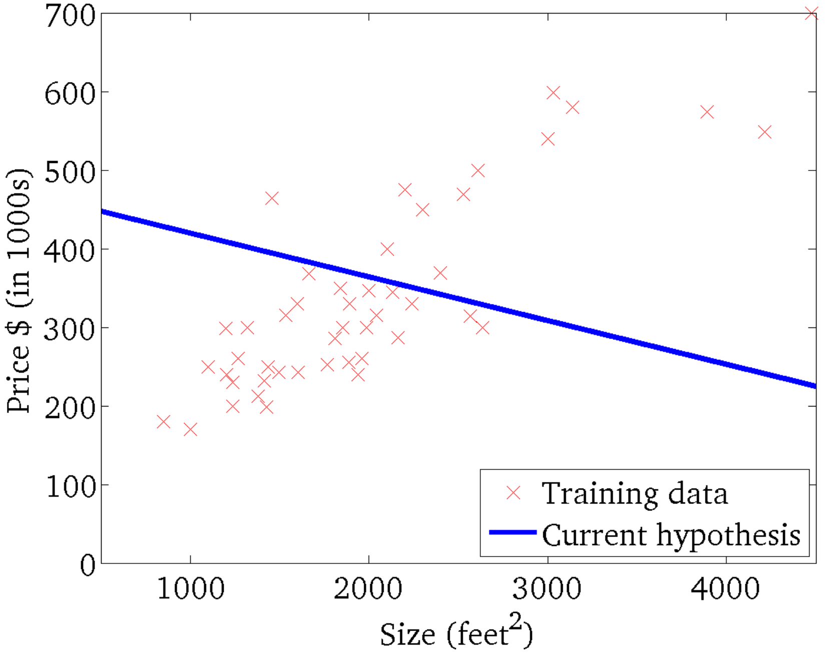
$J(w,b)$

(function of the parameters w, b)



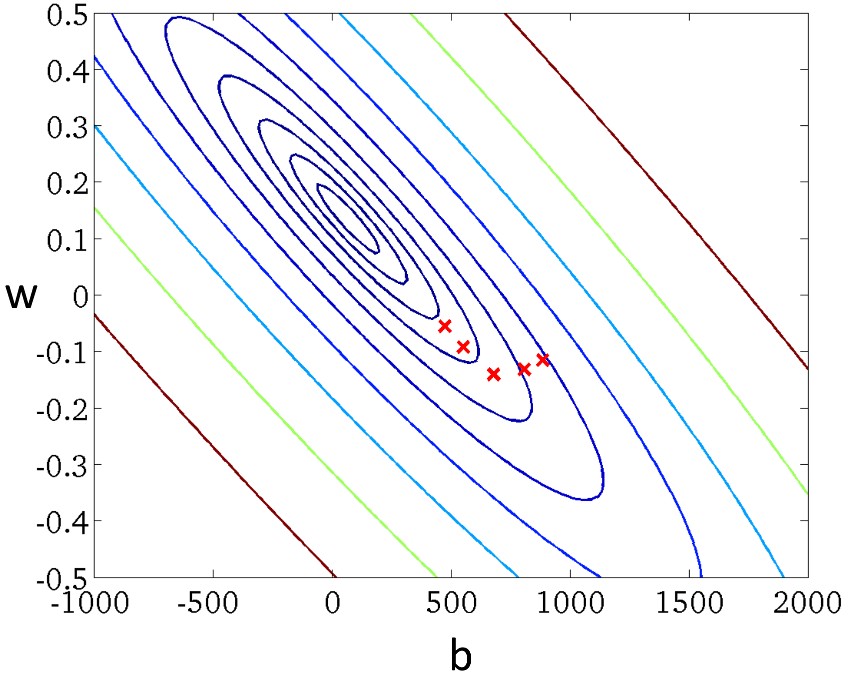
$f_{w,b}(x)$

(function of x (when w and b fixed))



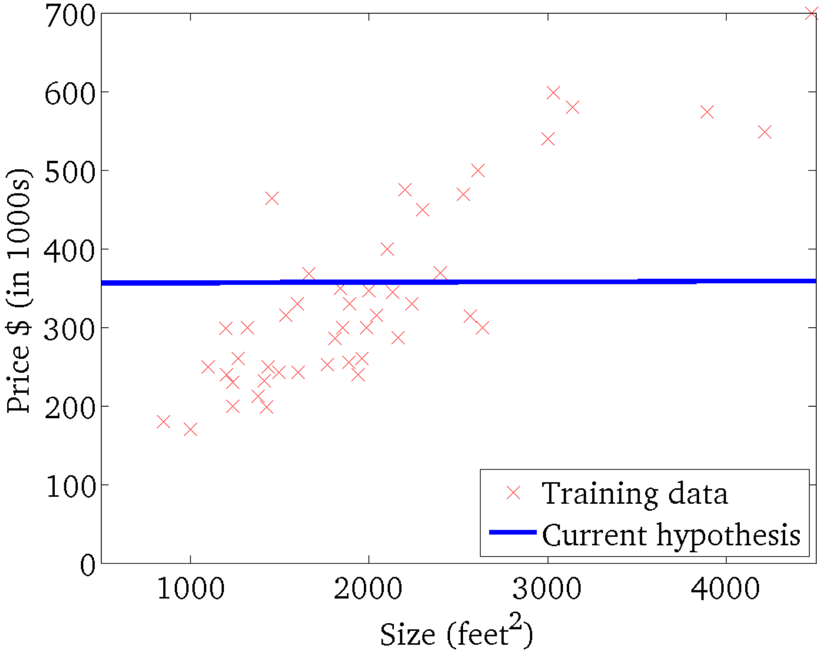
$J(w,b)$

(function of the parameters w, b)



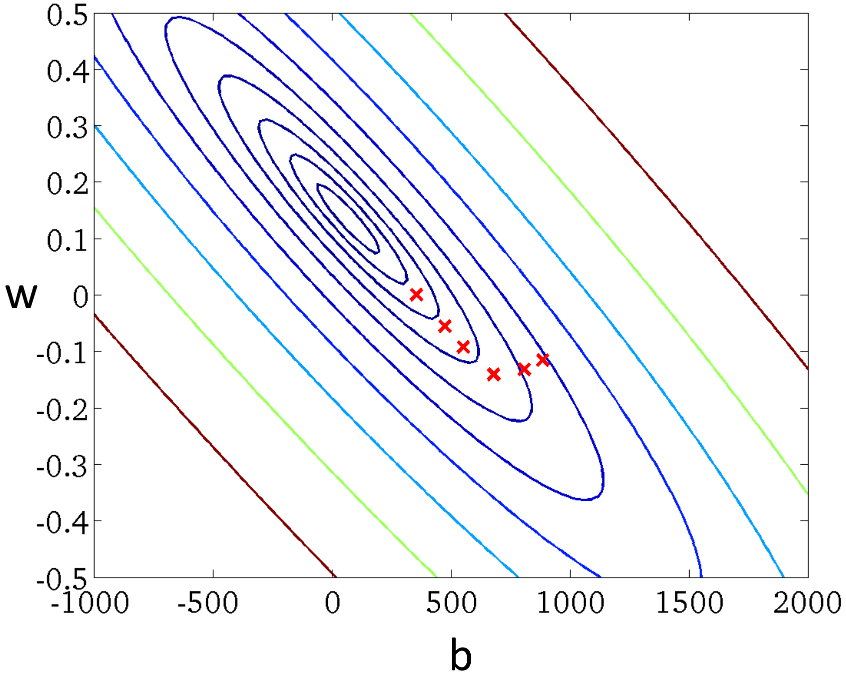
$f_{w,b}(x)$

(function of x (when w and b fixed))



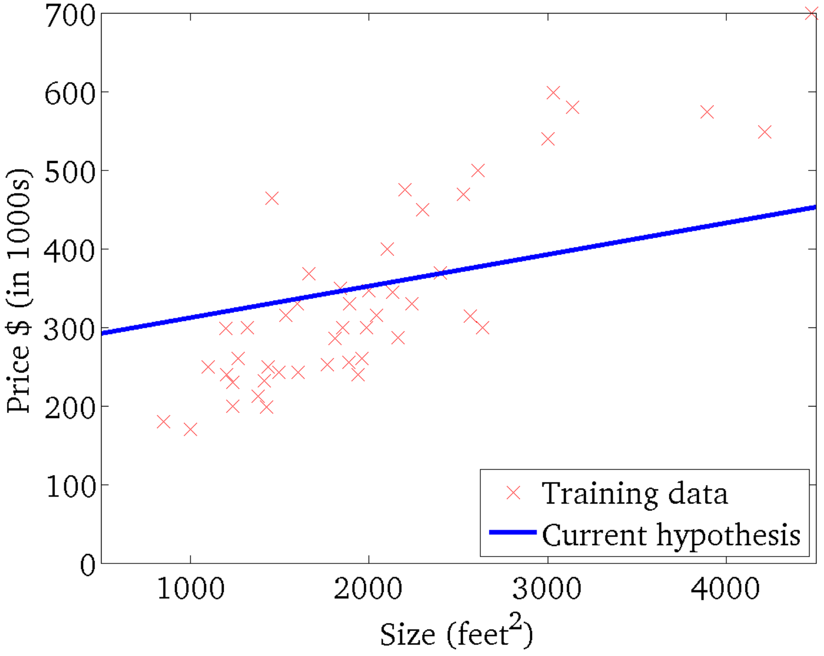
$J(w,b)$

(function of the parameters w, b)



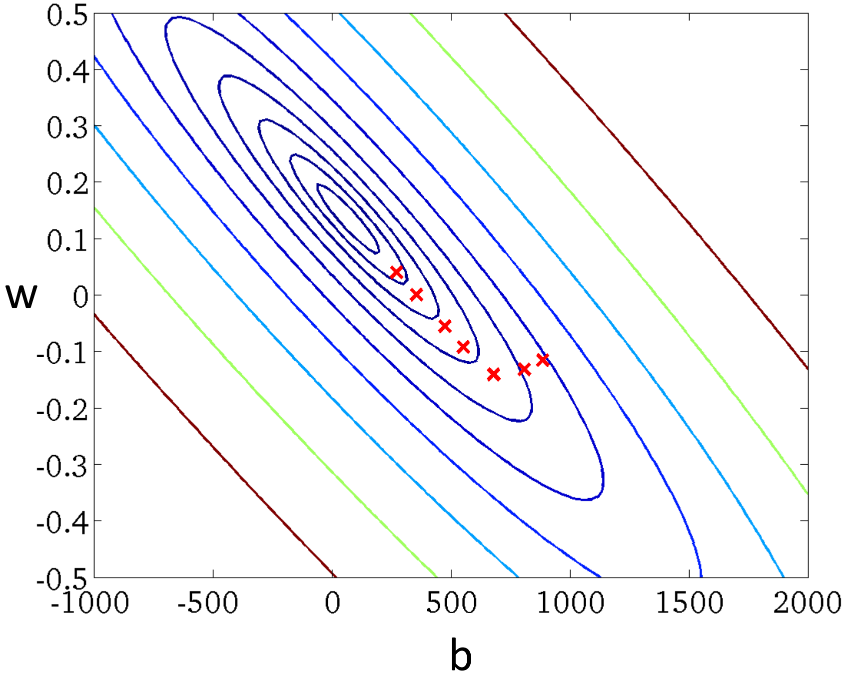
$f_{w,b}(x)$

(function of x (when w and b fixed))



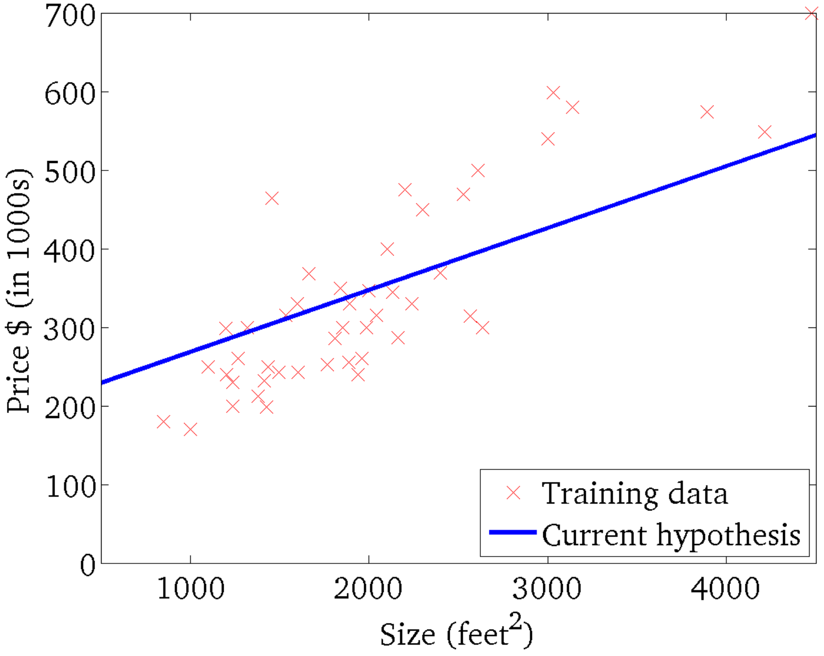
$J(w,b)$

(function of the parameters w, b)



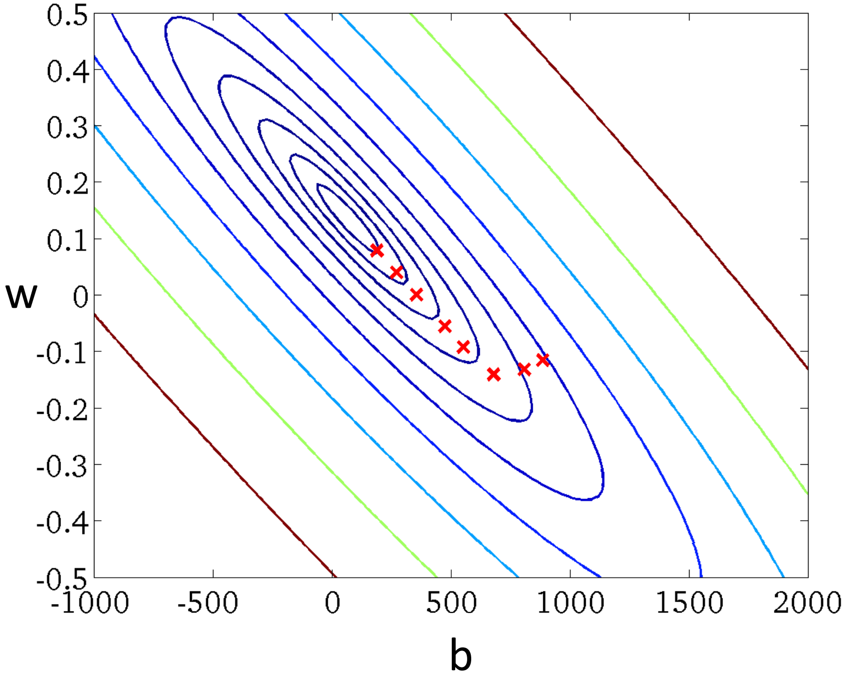
$f_{w,b}(x)$

(function of x (when w and b fixed))



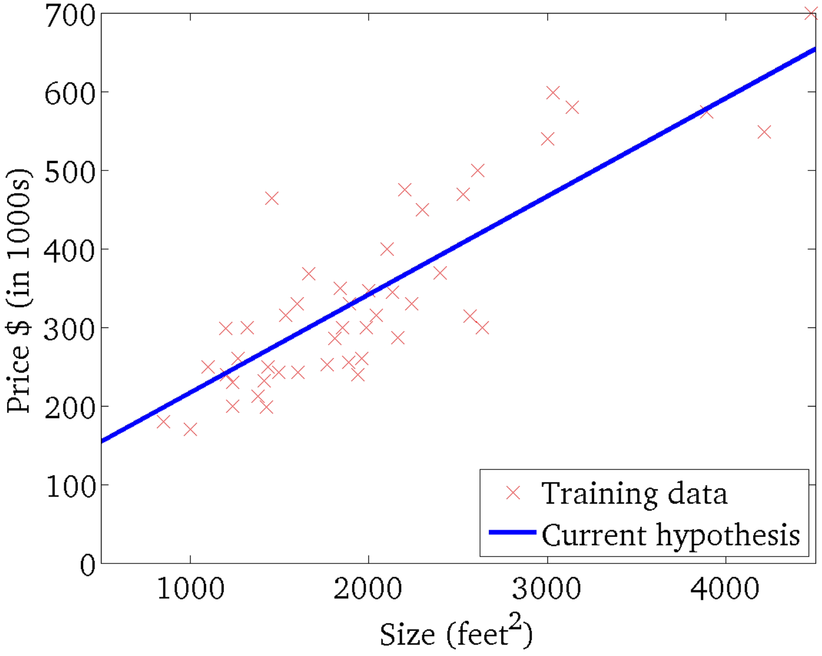
$J(w,b)$

(function of the parameters w, b)



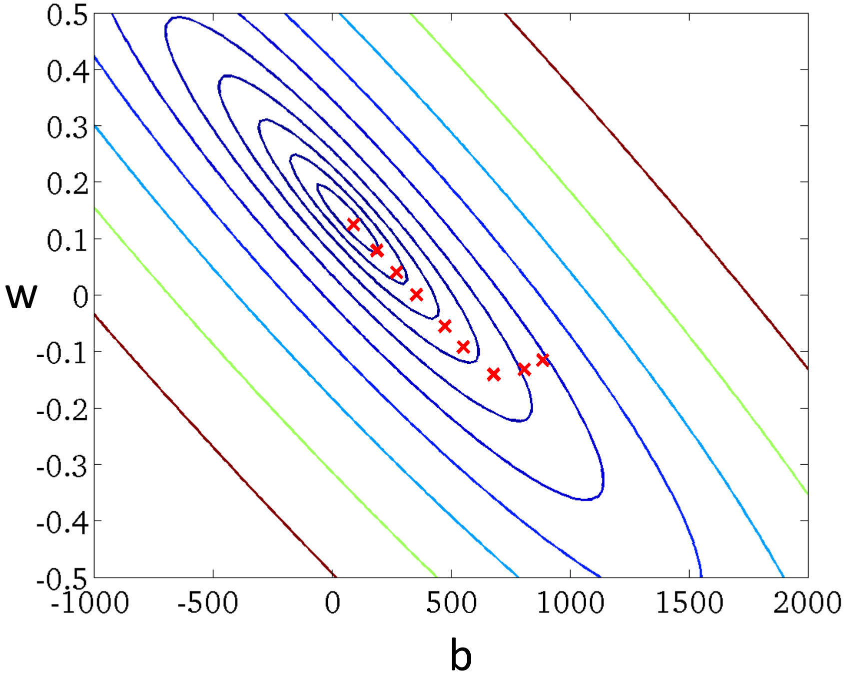
$f_{w,b}(x)$

(function of x (when w and b fixed))



$J(w,b)$

(function of the parameters w, b)



“Batch” Gradient Descent

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