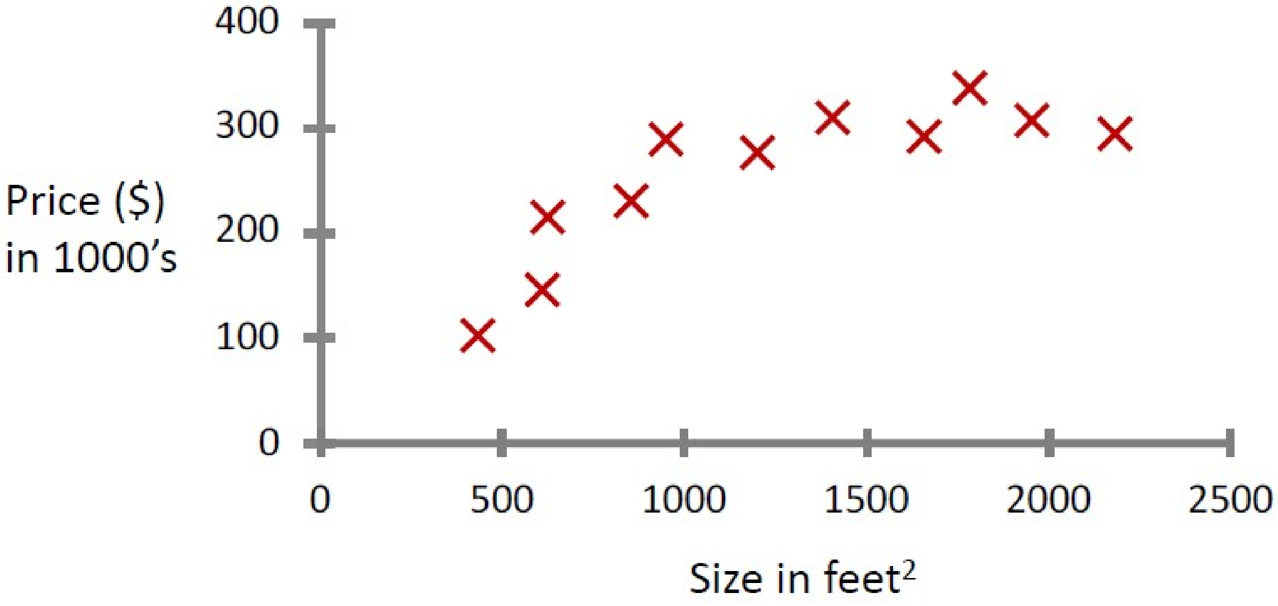
Tema 02. Regresión y clasificación

Autor: Ismael Sagredo Olivenza

2.1 Linear regression with one variable

- Initialy emerged as a statical method
- Is a linear model, e.g, a model that assumes a linear relationship between the input (X) and the output (Y)
- When there is a a single input variable is named **Simple linear regression**.
- When thre are multiple inputs is named **Multiple linear regression**

House sizes and prices

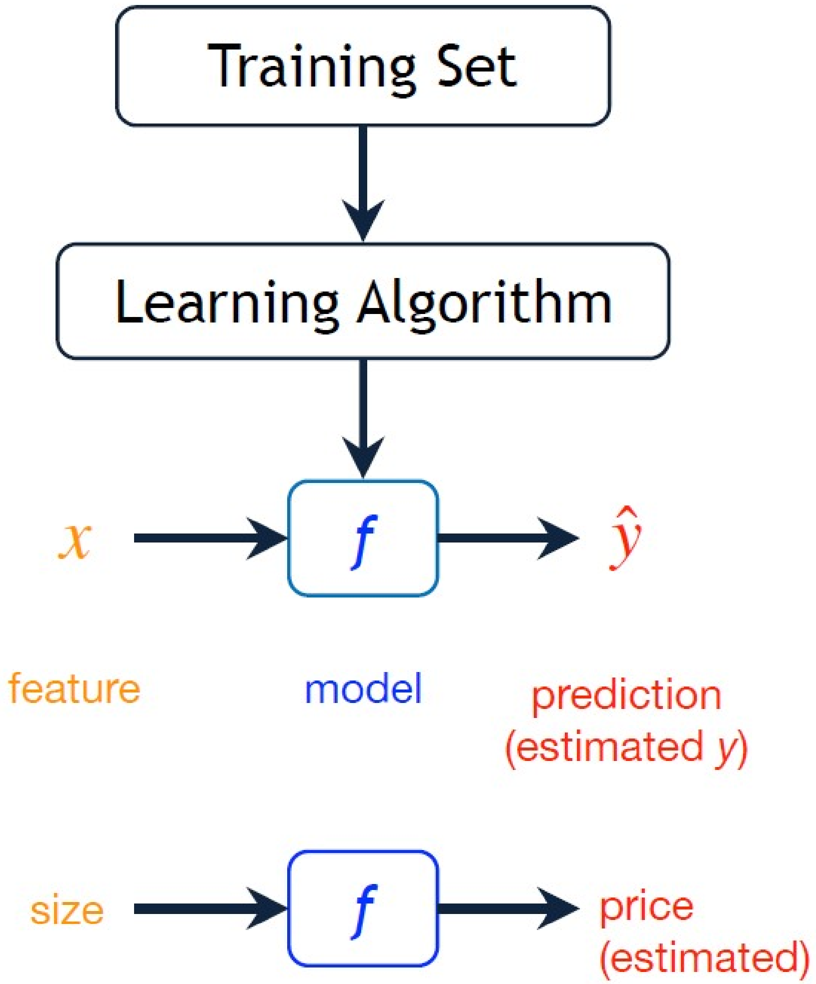


Size in feet^2	Price in \$1000's
2104	460
1416	232
1534	315
852	178
…	…
3210	870

2.1.1 Terminology

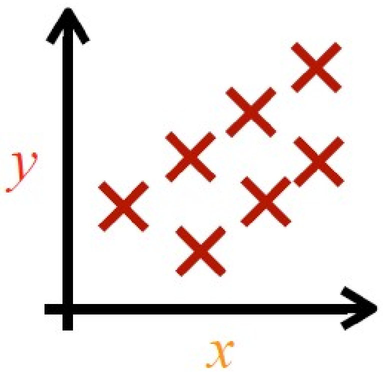
- Training Data is used to train the model
- X : input (also named feature)
- Y : output (also named target)
- Example: is one of the rows of the table that we saw in the previous image. (X,Y)

Machine learning



How to represent f ?

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



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

Linear regression with one variable

Univariate linear regression

2.1.2 Linear regression model

- Is a equation

$$Y = w * x + b => f(w, b) = w * x + b$$

- How do we calculate w and b?
- There are different techniques
  - Ordinary Least Squares
  - Gradient Descent
  - Regularization

2.1.3 Cost function

- A cost function is a mathematical formula that allows a machine learning algorithm to analyze **how well its model fits the data** given

- A cost function returns an output value, called the cost, which is a numerical value representing the deviation between the model representation and the data

- We can use a multitude of cost functions, but one of the most widely used is the **mean square error (MSE)**

2.1.3.1 Mean square error (MSE)

$$MSE(Y, Y') = \frac{1}{m} \cdot \sum_{i=1}^{m} (y_i - y_i')^2$$

- Distance between y_i' (estimated => $f_{w,b}(x)$ in linear regression) and the real values y_i squared (to keep the error always positive)
- All deviations are added together and the mean is calculated.
- In practice: we divide this by 2 for mathematical convenience when finding the partial derivative of the cost function

$$MSE(Y, Y') = \frac{1}{2m} \cdot \sum_{i=1}^{m} (y_i - y_i')^2$$

2.1.3.2 Goal of linear regression

- Minimize the cost function

Example:

Let us call the cost function **J**. The cost function for $f_{w,b}(x)$ will be $J(w, b)$ (these are the parameters we want to calculate).

Let us simplify the model by eliminating the parameter b for the example.

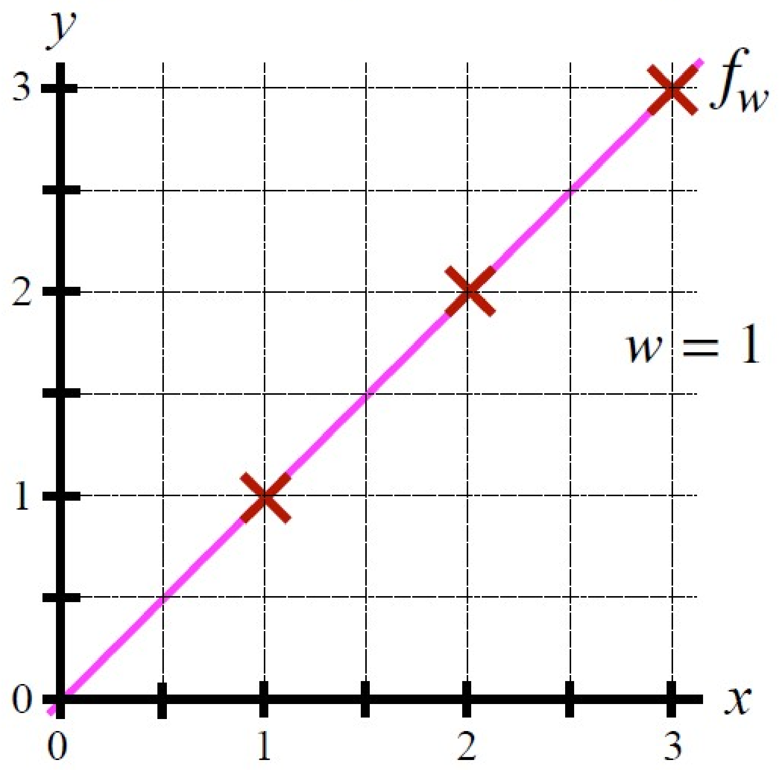
$$f(w) = w * x$$

- Let's assume some fictitious date where

X	Y
1	**1**
2	**2**
3	**3**

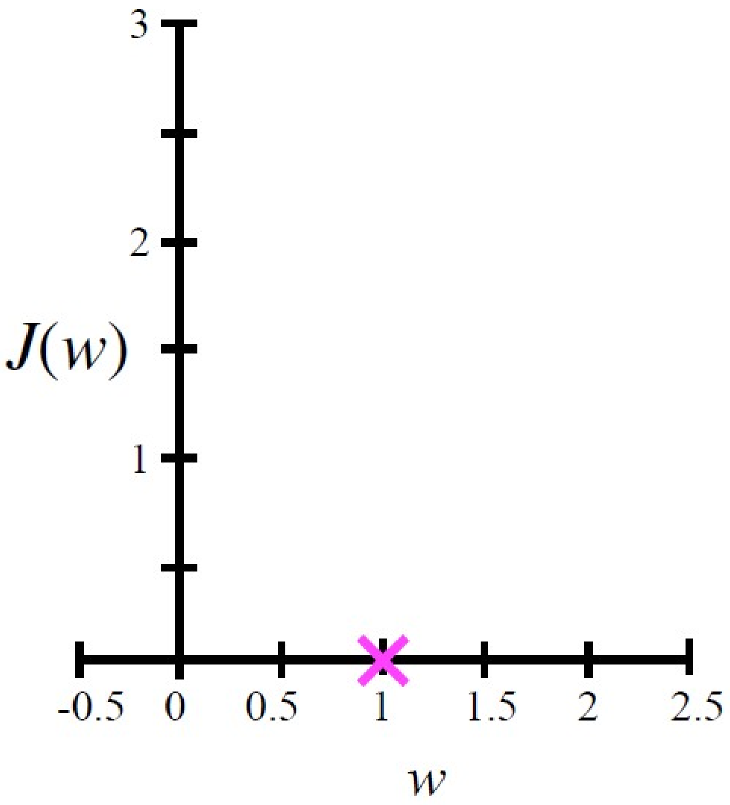
$f_w(x)$

(for fixed w, function of x)

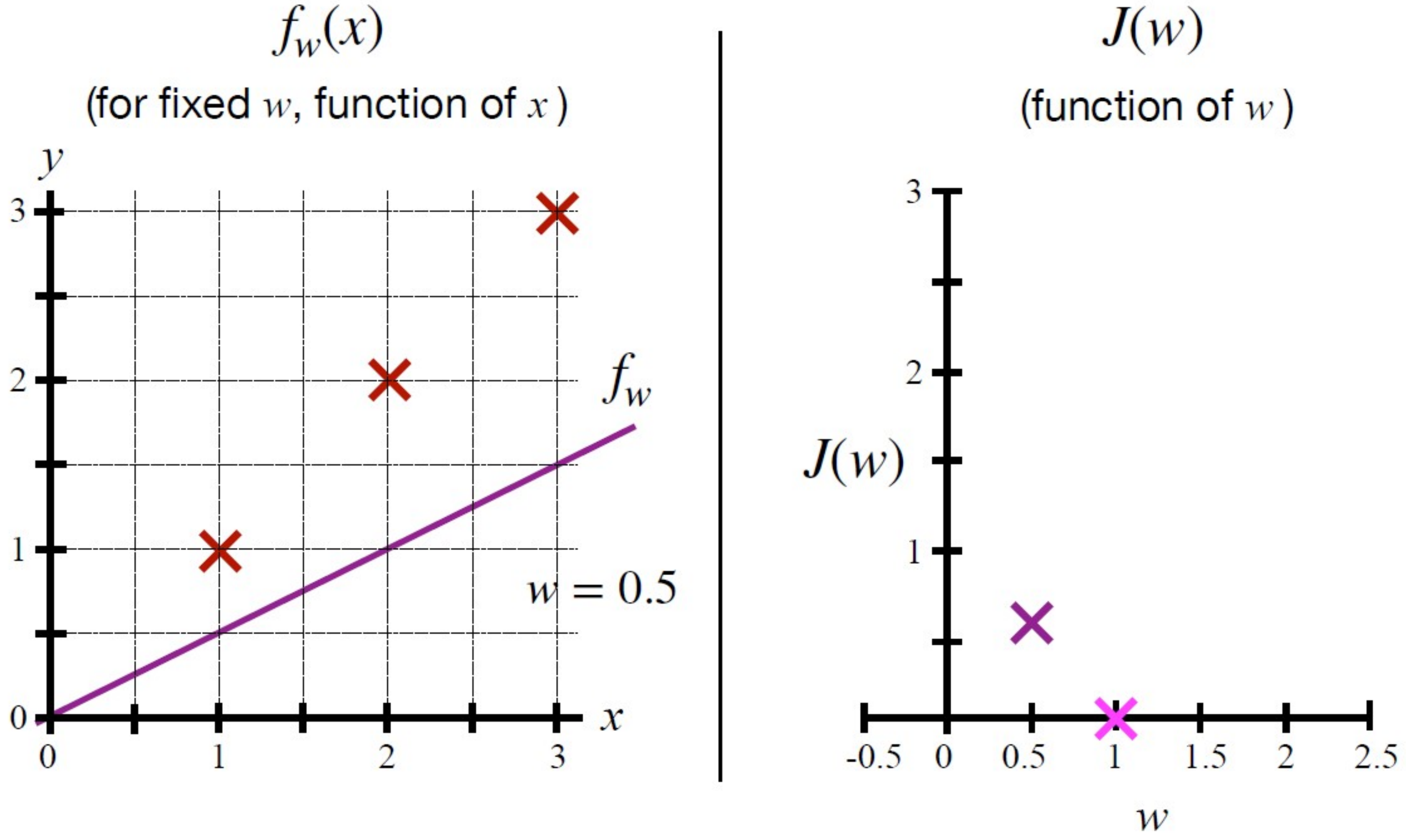


$J(w)$

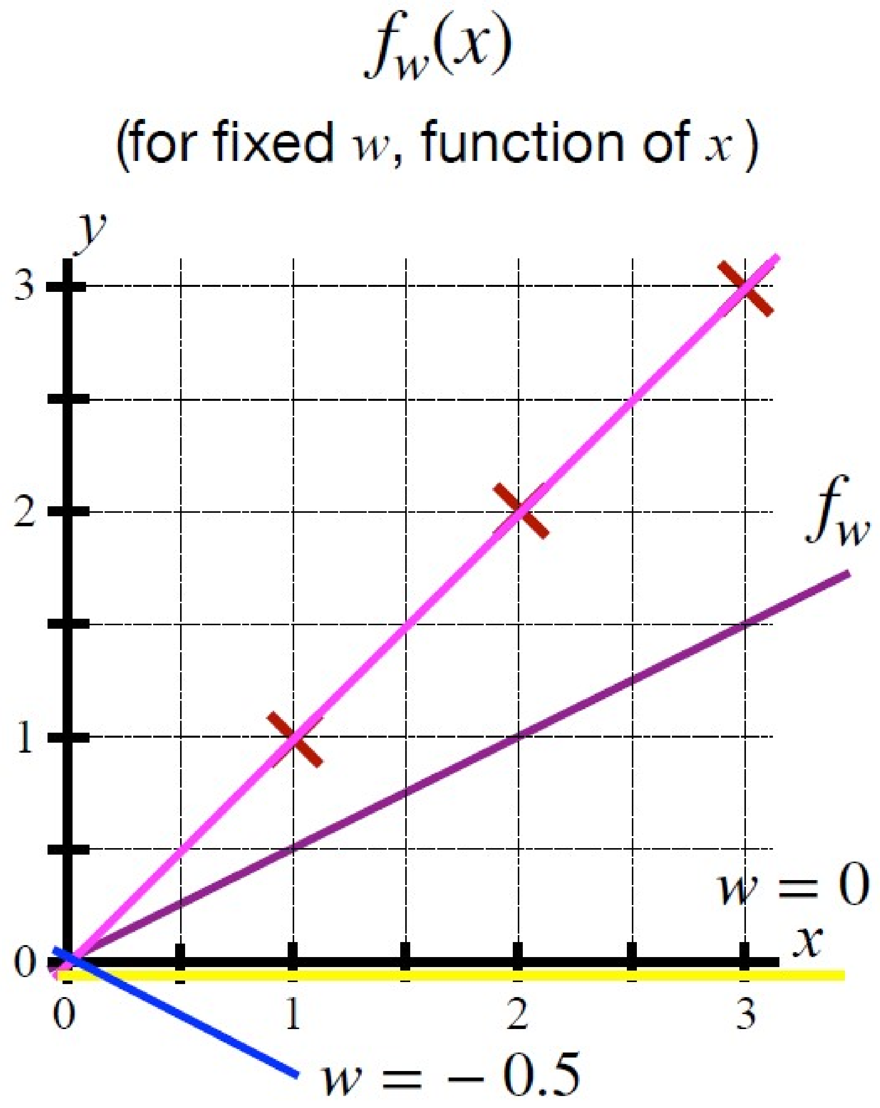
(function of w)



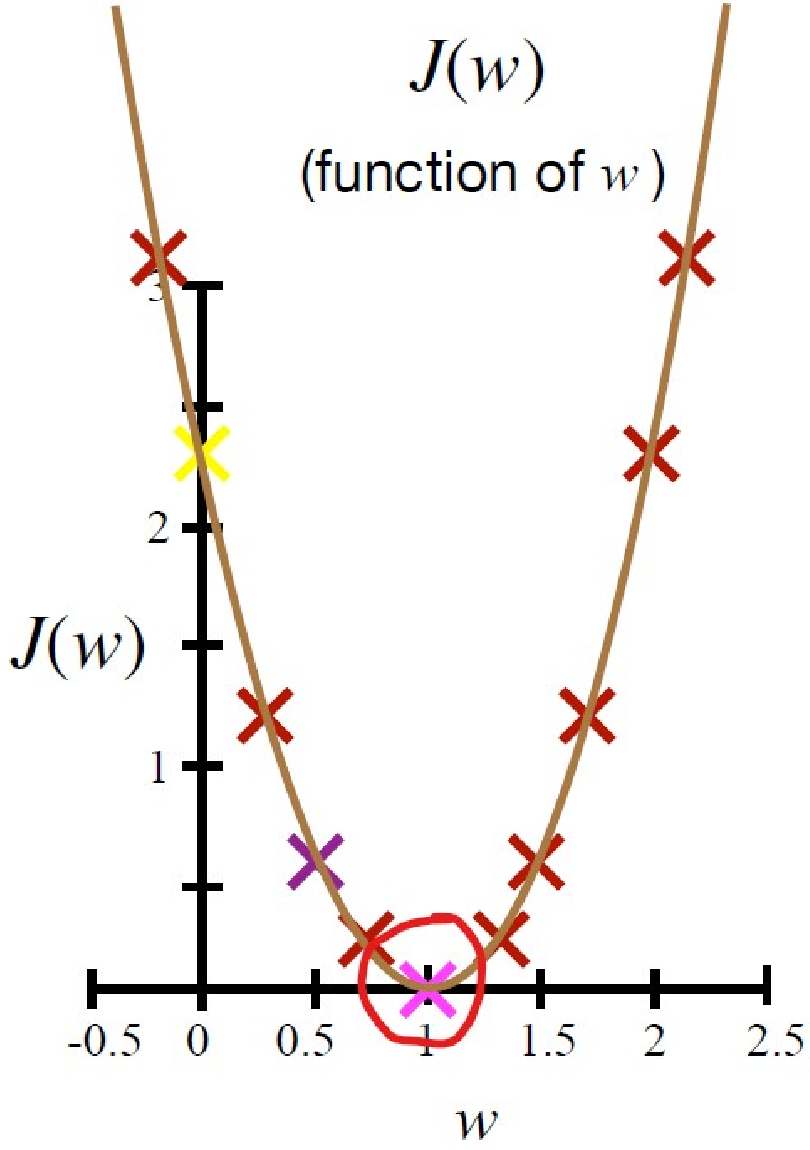
$$J(1) = \frac{1}{2m} \sum_{i=1}^{m} \left(f_w(x^{(i)}) - y^{(i)} \right)^2 = \frac{1}{2m} \sum_{i=1}^{m} \left(wx^{(i)} - y^{(i)} \right)^2 = \frac{1}{2m}(0^2 + 0^2 + 0^2) = 0$$



$$J(0.5) = \frac{1}{2m} \cdot \sum_{i=1}^{m} (wx_i - y'_i)^2 = frac12m(0.5^2 + 1^2 + 1.5^2) = \frac{1}{6}(0.25 + 1 + 2.25) = 0.58$$



$f_w(x)$
(for fixed w, function of x)
y
3
2
1
0
0
1
2
3
x
f_w
$w = 0$
$w = -0.5$



$J(w)$
(function of w)
$J(w)$
3
2
1
-0.5
0
0.5
1
1.5
2
2.5
w

2.2 Gradient Descent

- Start with random value of w and b.
- Minimize $J(w, b)$
- Keep changing w,b until we settle ar or near to minimum.

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

- α is the amount by which we shift w and b in each iteration and s name (learning rate)

Gradient descent algorithm

Repeat until convergence

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

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

Code assignment	Math
$a = c$	$a = c$
$a = a + 1$	$a \neq a + 1$

Correct: simultaneous update

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

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

$$w = tmp\_w$$

$$b = tmp\_b$$

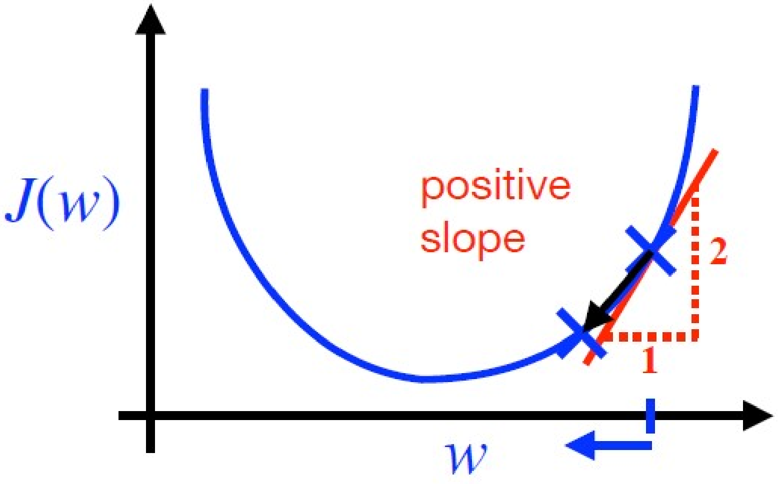
Incorrect

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

$$w = tmp\_w$$

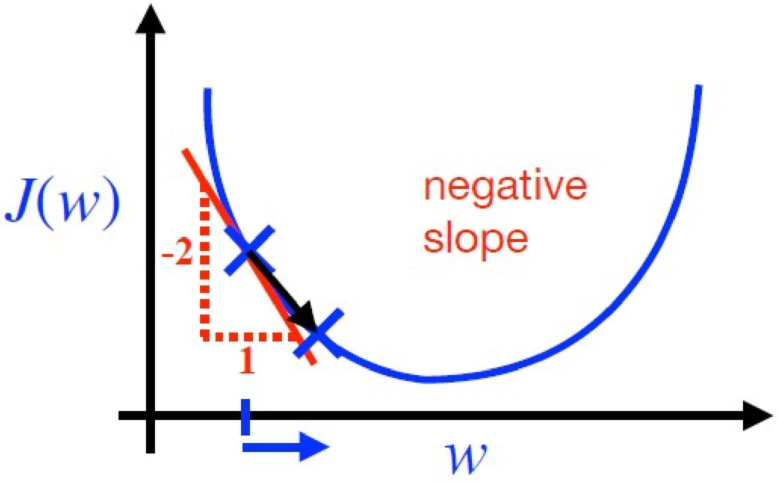
$$tmp\_b = b - \alpha \frac{\partial}{\partial b} J(w, b)$$

$$b = tmp\_b$$



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

$$w = w - \alpha \cdot (positive\ number)$$



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

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

2.2.1 Gradient descent in linear regression

- Cost function

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

- Partial derivate:

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

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

2.2.1.1 Apply derivate

Remember: $\partial x^2 = 2x$

$$\frac{\partial J(w,b)}{\partial w} = \frac{1}{2m} \cdot \sum_{i=1}^{m} (wx_i + b - y_i)^2 = \frac{1}{2m} \cdot \sum_{i=1}^{m} ((y_i\prime - y_i) \cdot 2x_i) = \frac{1}{m} \sum_{i=1}^{m} (y_i\prime - y_i) * x_i$$

Remember: $\partial x = 1$

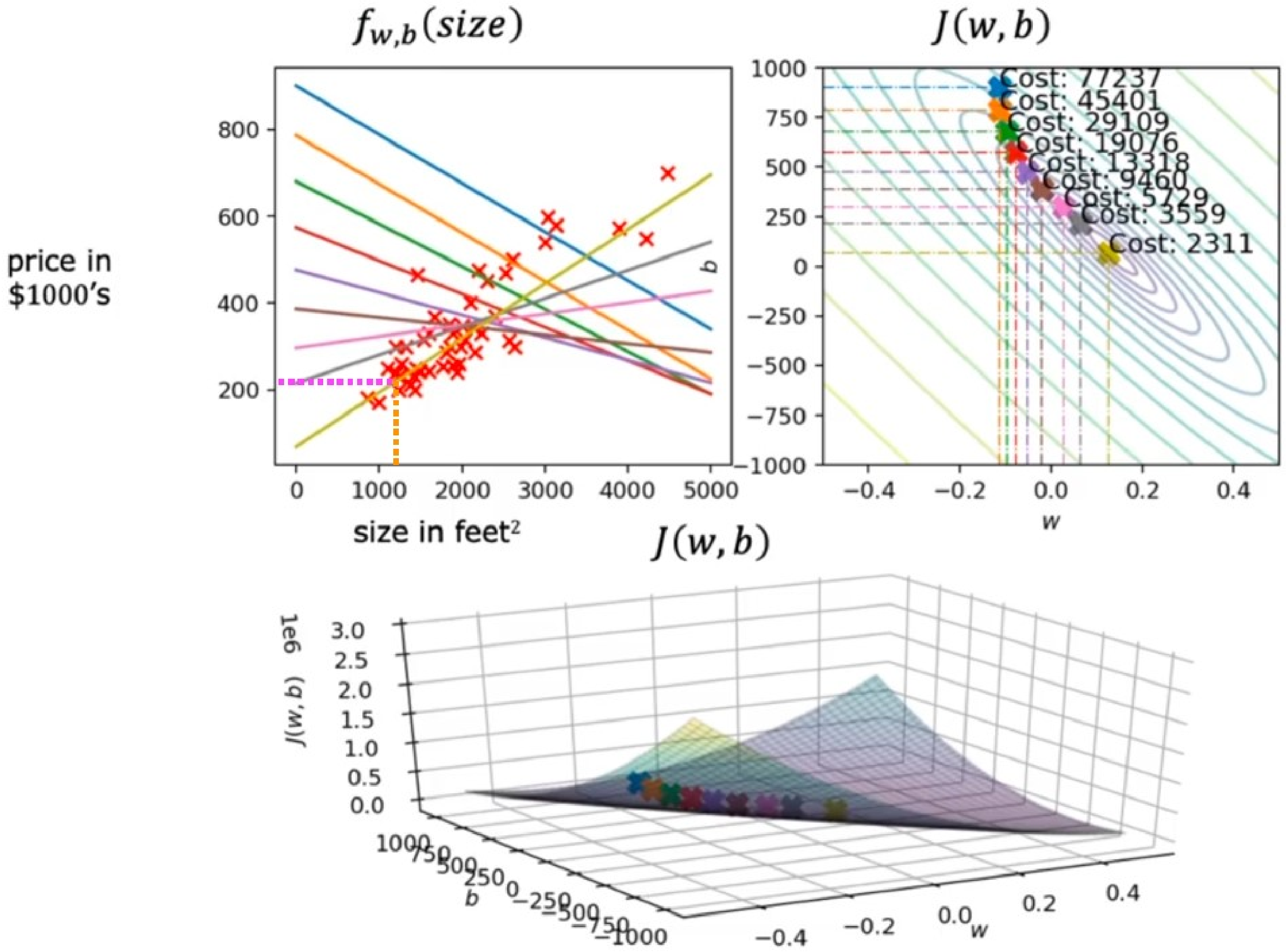
$$\frac{\partial J(w,b)}{\partial b} = \frac{1}{2m} \cdot \sum_{i=1}^{m} (wx_i + b - y_i)^2 = \frac{1}{2m} \cdot \sum_{i=1}^{m} ((y_i\prime - y_i) \cdot 2) = \frac{1}{m} \sum_{i=1}^{m} (y_i\prime - y_i)$$

2.2.2.2 Gradient Descent algorithm

Repeat until convergence.

$$w = w - \alpha \frac{1}{m} \sum_{i=1}^{m} (y_i\prime - y_i) * x_i$$

$$b = b - \alpha \frac{1}{m} \sum_{i=1}^{m} (y_i\prime - y_i)$$



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

2.2.2.2 Bach

Depending on the number of training examples considered in updating the model parameters, we have 3-types:

- **Batch Gradient Descent**: Parameters are updated after computing the gradient of the error with respect to the **entire training set**
- **Stochastic Gradient Descent**: Parameters are updated after computing the gradient of the error with respect to a **single training example**
- **Mini-Batch** Gradient Descent: Parameters are updated after computing the gradient of the error with respect to **a subset of the training set**

2.2.2.2 Mini-Bach Algorithm

- **Forward pass** on selected examples in the batch.
  - Make prediction on the mini-batch
    Compute error in prediction (J)
- **Backward pass** compute de gradient descent
- Update parameters