

Paradigmas de aprendizado de máquina



Paradigmas	Tarefas	
	Supervisionado	Não-supervisionado
	Classificação	Mineração de <i>itemsets</i>
	Regressão	Agrupamento (<i>clustering</i>)
	Outros	Redução de dimensionalidade Outros

7 tarefas comuns de aprendizado de máquina:

<http://vitalflux.com/7-common-machine-learning-tasks-related-methods/>

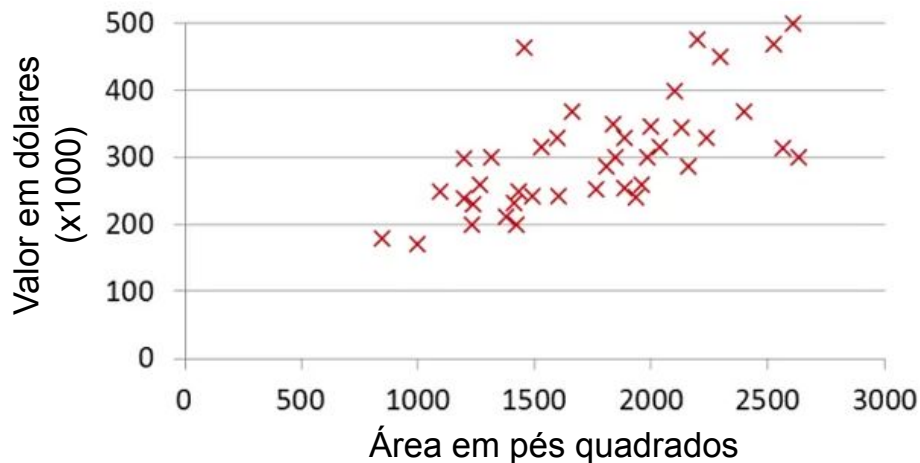


- ▶ O objetivo da regressão é aproximar um **valor** para uma instância não conhecida, com base nas instâncias conhecidas
- ▶ Diferente da classificação, onde o objetivo é aproximar uma **categoria**



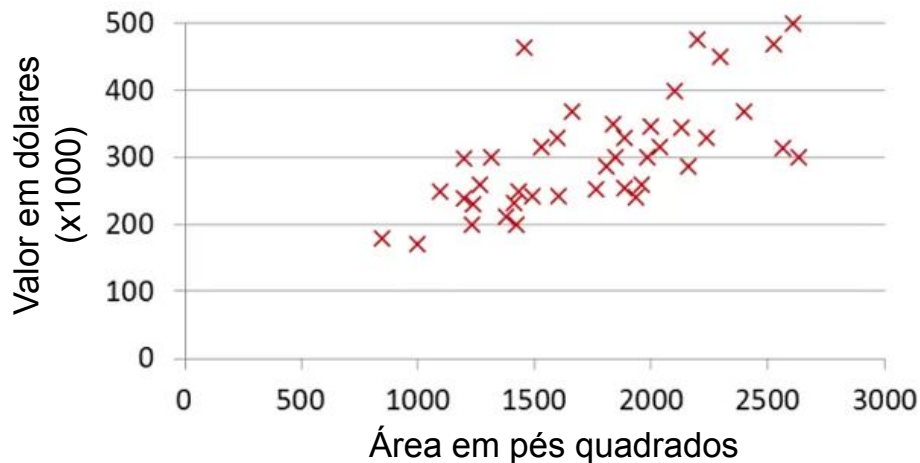


- Problema: dado o tamanho de uma casa (em pés quadrados, 1 pé = 30.48cm), qual o preço desta casa?





- ▶ Como podemos resolver o problema de aproximar um preço para uma área de casa?



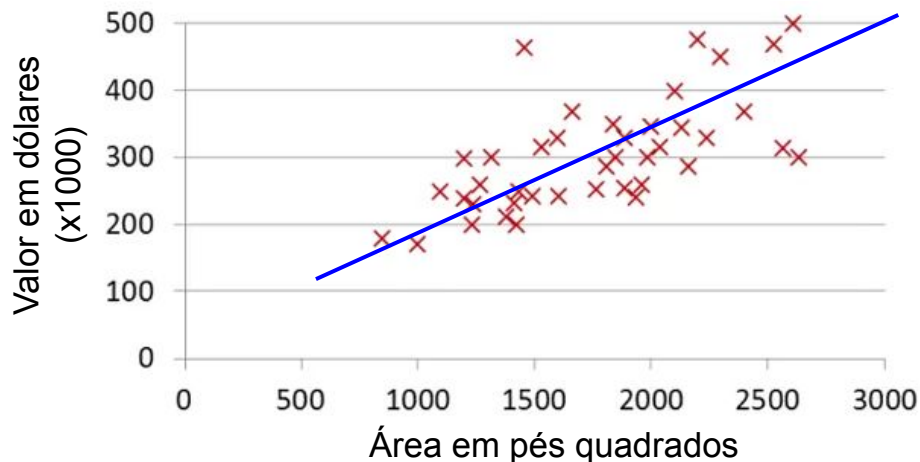


- Podemos utilizar regressão linear **univariada** (quando há apenas um atributo), onde a **hipótese** (i.e. modelo) é dada por:

$$h_{\Theta}(x) = 1 * \Theta_0 + \Theta_1 * x$$

Data table

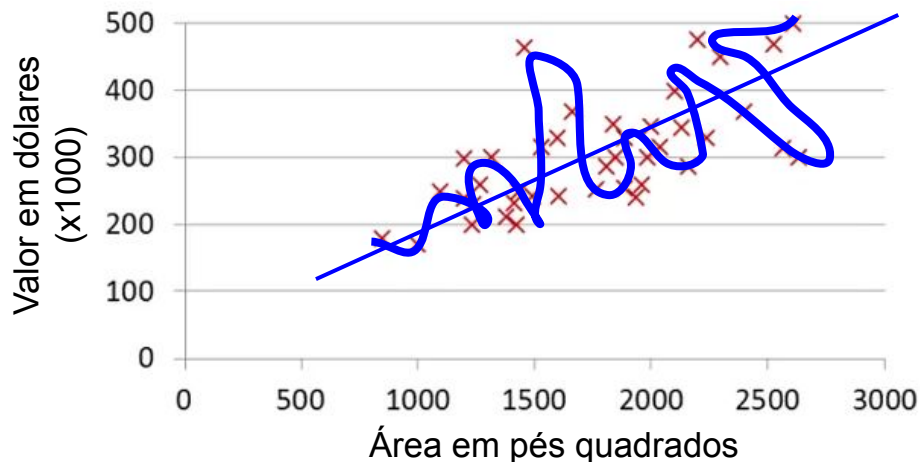
size in feet ²	price in \$1000's
2104	400
1416	232
1534	315
852	178
...	...
3210	870





- Podemos utilizar regressão linear **univariada** (quando há apenas um atributo), onde a **hipótese** (i.e. modelo) é dada por:

$$h_{\Theta}(x) = 1 * \Theta_0 + \Theta_1 * x$$





bias (**sempre** é 1)

Atributo
preditivo x

$$h_{\Theta}(x) = 1 * \Theta_0 + \Theta_1 * x$$

Predição (i.e. valor de
y, ou valor de classe)

Parâmetros theta zero e theta 1



Exercício



- ▶ Faça uma projeção para cada um dos parâmetros (i.e. pesos) a seguir:

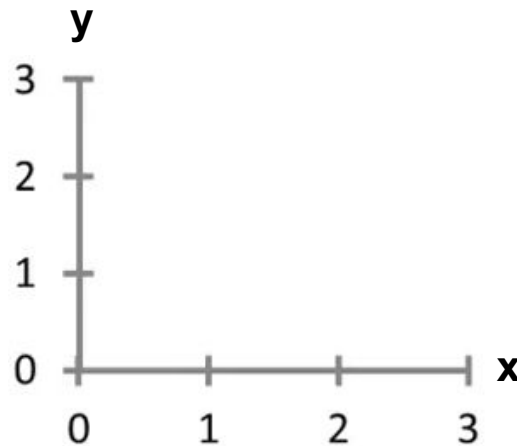
- ▶ Variando Θ_0 :

- ▶ $\Theta_0 = 0, \Theta_1 = 0.5$
- ▶ $\Theta_0 = 0.5, \Theta_1 = 0.5$
- ▶ $\Theta_0 = 1, \Theta_1 = 0.5$

- ▶ Variando Θ_1 :

- ▶ $\Theta_0 = 0.5, \Theta_1 = 0$
- ▶ $\Theta_0 = 0.5, \Theta_1 = 0.5$
- ▶ $\Theta_0 = 0.5, \Theta_1 = 1$

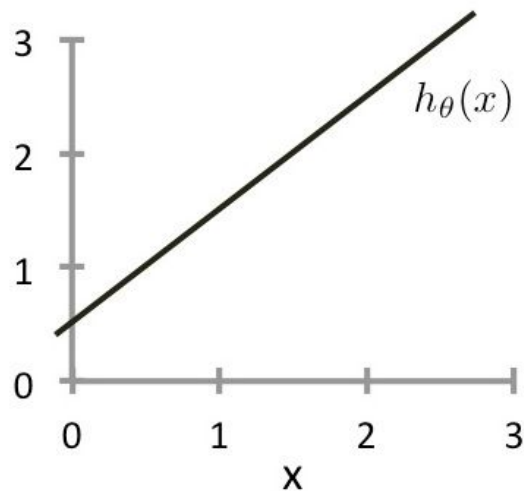
x	y
0	?
1	?
2	?
3	?



Exercício




- ▶ Quais os valores de Θ_0 e Θ_1 ?







Exercício



- ▶  Exercício 1: Reflexão Baseada em Cenário
- ▶ Um pesquisador quer entender a relação entre número de horas de estudo e nota final dos alunos. Ele coletou os seguintes dados:

Horas de Estudo	Nota
1	50
2	55
3	60
4	68
5	75

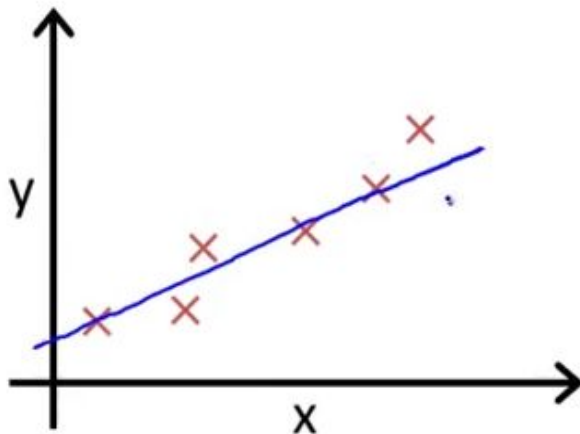


- ▶  Exercício 1: Reflexão Baseada em Cenário
- ▶  1 Se você precisasse prever a nota de um aluno que estudou 6 horas, como faria isso sem usar programação tradicional?
- ▶  2 Que padrões você consegue identificar nos dados apenas olhando para eles?
- ▶  3 Será que outros fatores além do estudo podem influenciar a nota? Como a regressão linear lida com isso?





O objetivo da hipótese é se aproximar dos valores do conjunto de treino, o que significa dizer que a distância entre $h_{\Theta}(x)$ e os dados deve ser **minimizada**





Training set

<i>features</i> size in feet ² (x)	<i>targets</i> price \$1000's (y)
2104	460
1416	232
1534	315
852	178
...	...

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

w, b : parameters
coefficients
weights

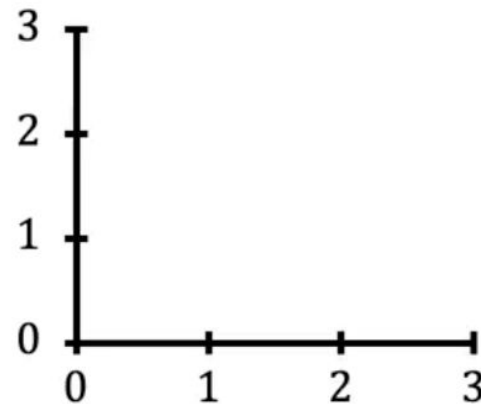
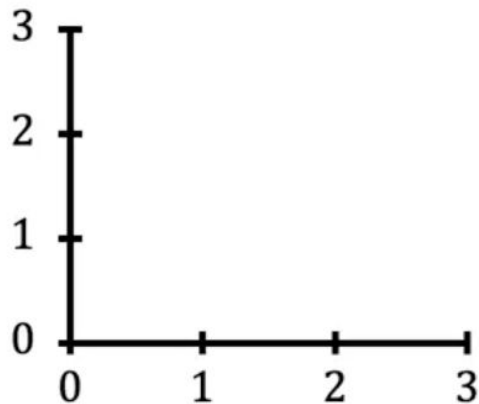
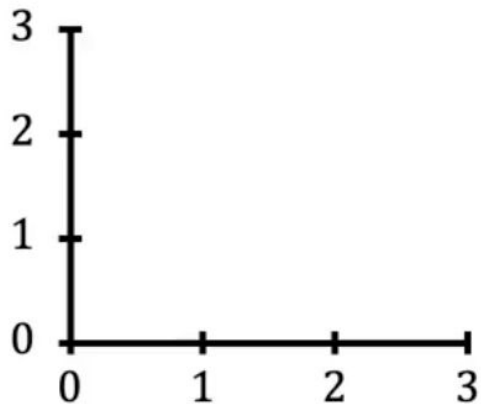
What do w, b do?





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

$f(x)$

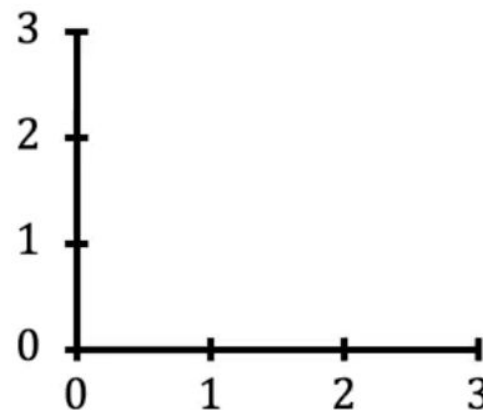
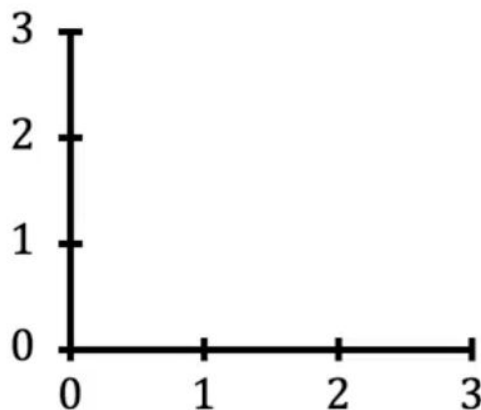
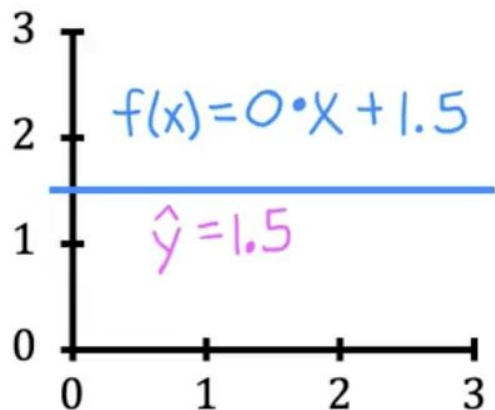


→ $w = 0$
→ $b = 1.5$





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

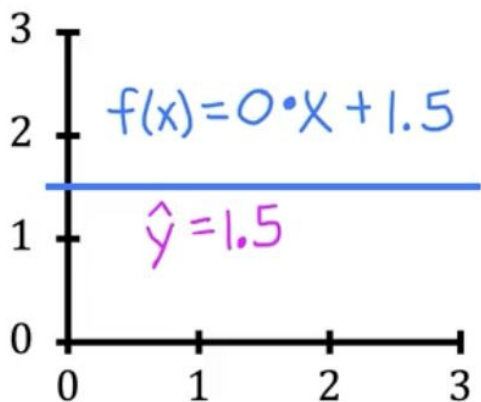
 $f(x)$ 

→ $w = 0$
→ $b = 1.5$
 y-intercept

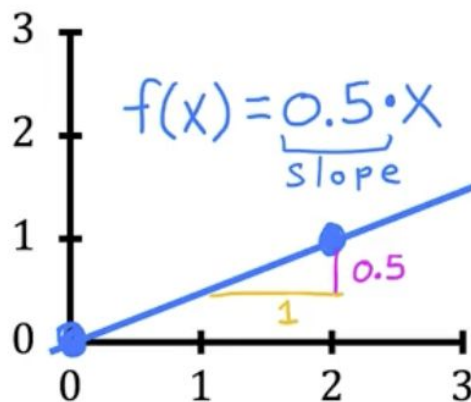
Regressão



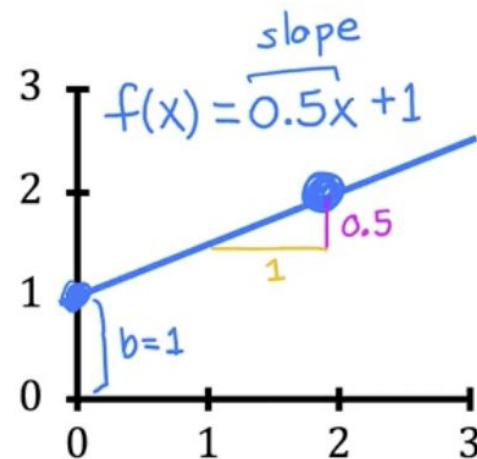
$$f_{w,b}(x) = wx + b$$

 $f(x)$ 

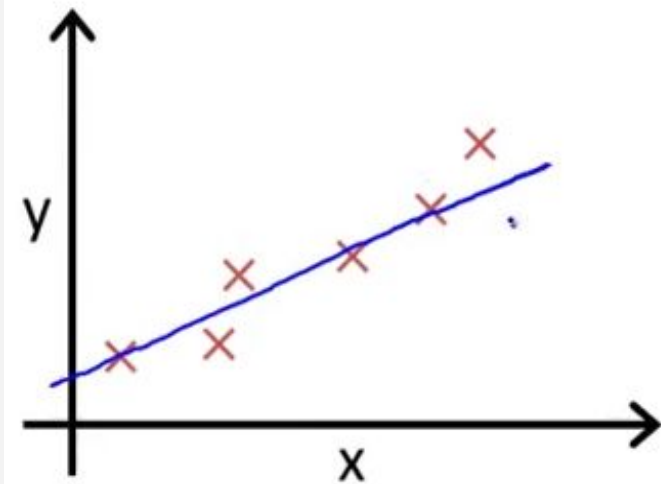
- $w = 0$
- $b = 1.5$
(y-intercept)



- $w = 0.5$
- $b = 0$



- $w = 0.5$
- $b = 1$



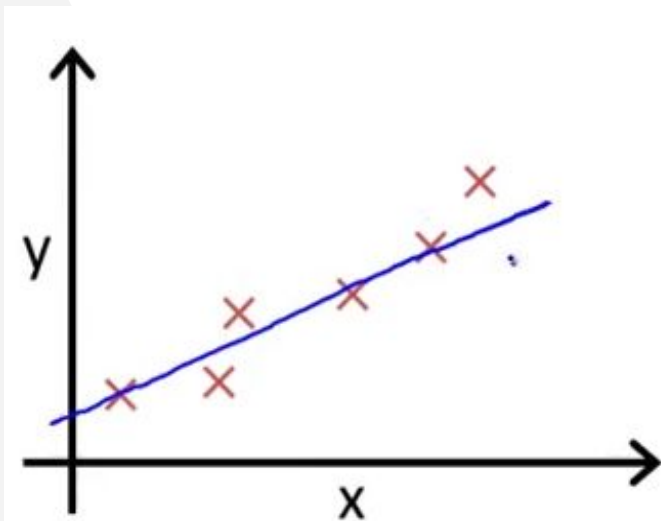
Cost function

$$\left(\hat{y} - y \right)_{\text{error}}$$

Find w, b :

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





Cost function

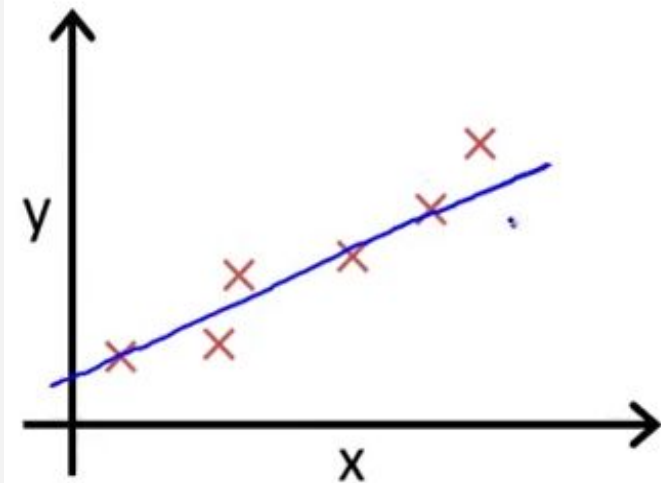
$$\sum_{i=1}^m \left(\underset{\text{error}}{\hat{y}^{(i)} - y^{(i)}} \right)^2$$

m = number of training exam

Find w, b :

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





Cost function: Squared error cost function

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

m = number of training examples

Find w, b :

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





- ▶ Podemos assumir que há um custo associado a cada hipótese realizada sobre o conjunto de dados
- ▶ O objetivo da regressão linear é traçar uma hipótese com **custo mínimo**
- ▶ A função de custo é dada por:

$$J(\theta_0, \theta_1) = \frac{1}{2N} \sum_{i=1}^N (h_{\theta}(x^{(i)}) - y^{(i)})^2$$





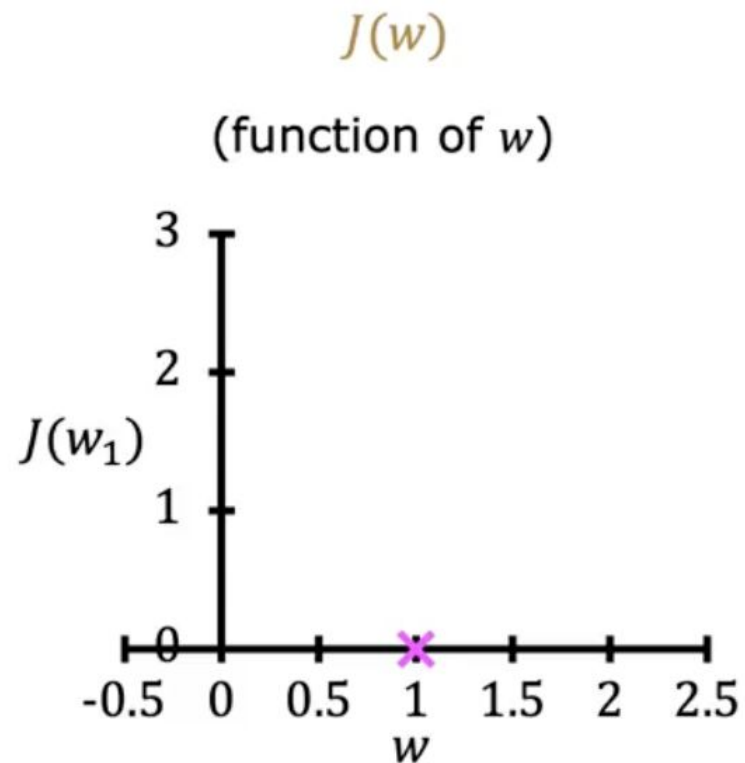
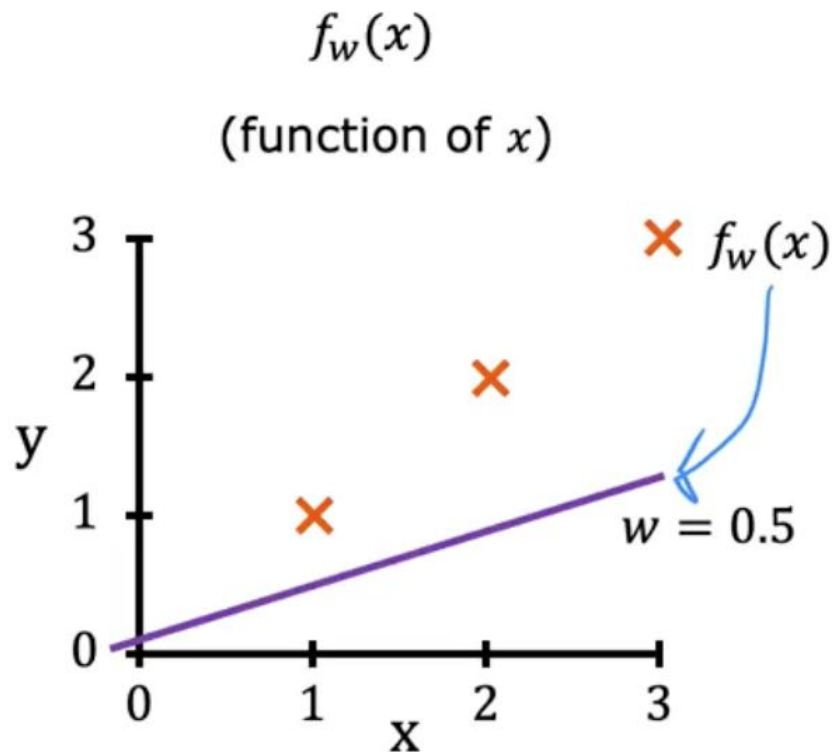
- ▶ Temos então que a **hipótese** é dada por:

$$h_{\theta}(x^{(i)}) = \theta_0 + \theta_1 x$$

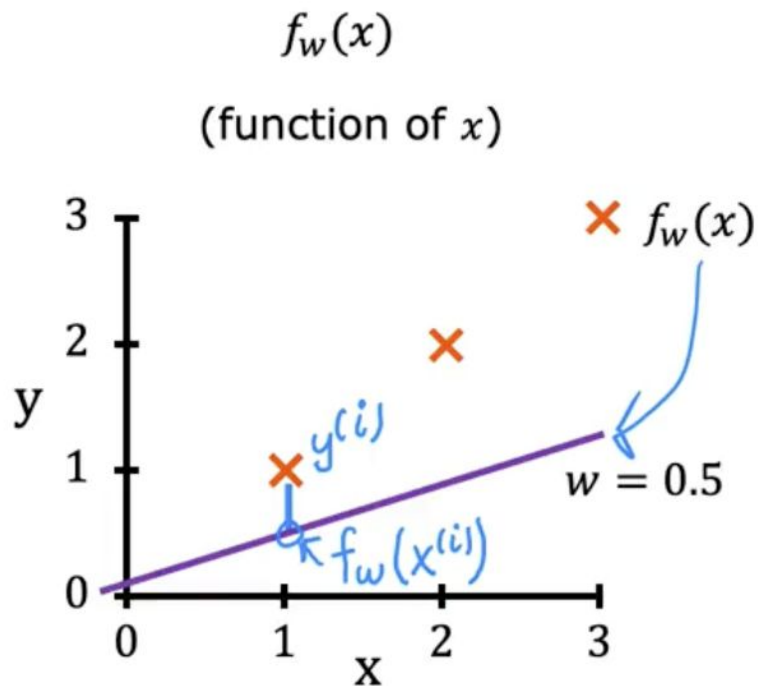
- ▶ E a função de custo é dada por:

$$J(\theta_0, \theta_1) = \frac{1}{2N} \sum_{i=1}^N (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

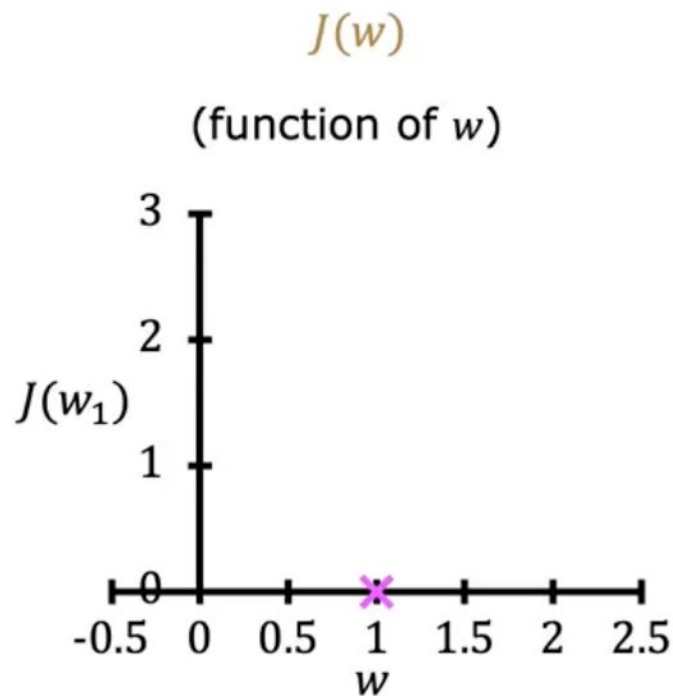




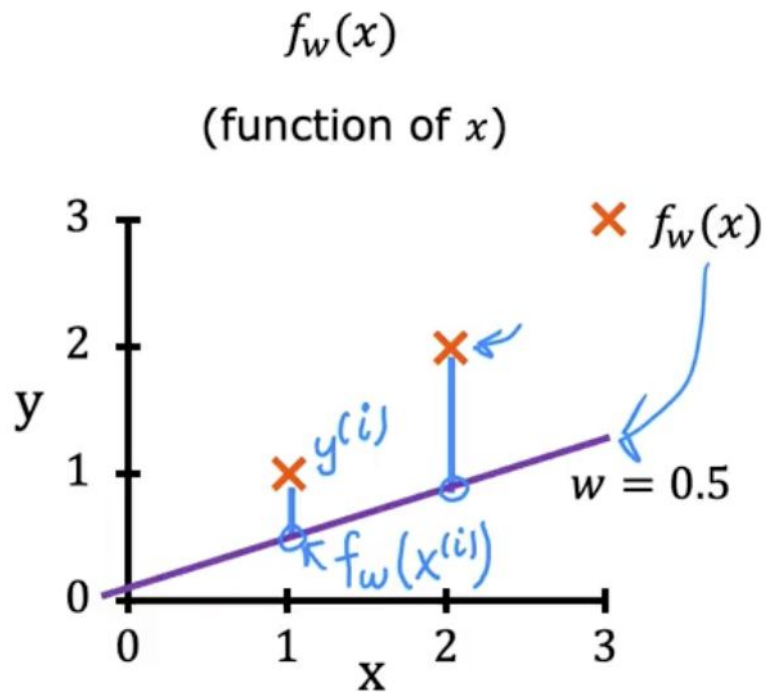
Regressão



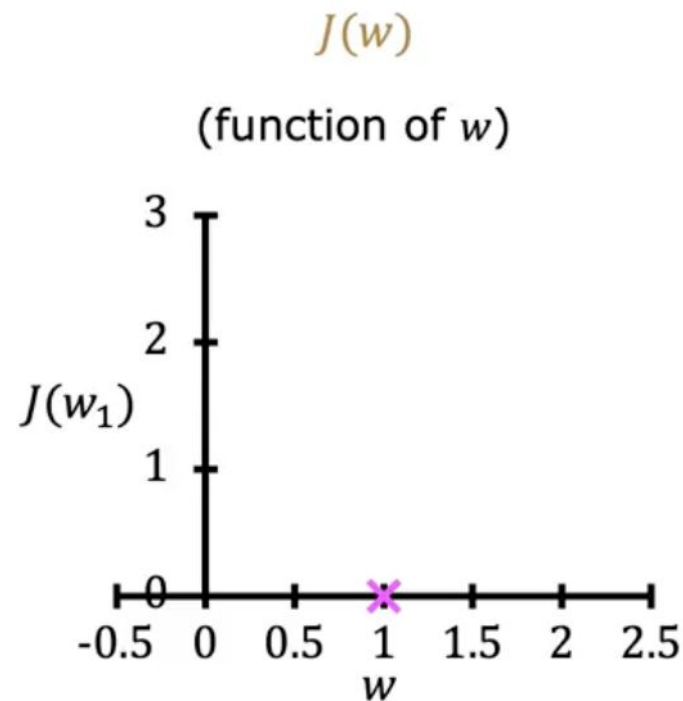
$$J(0.5) = (0.5 - 1)^2$$



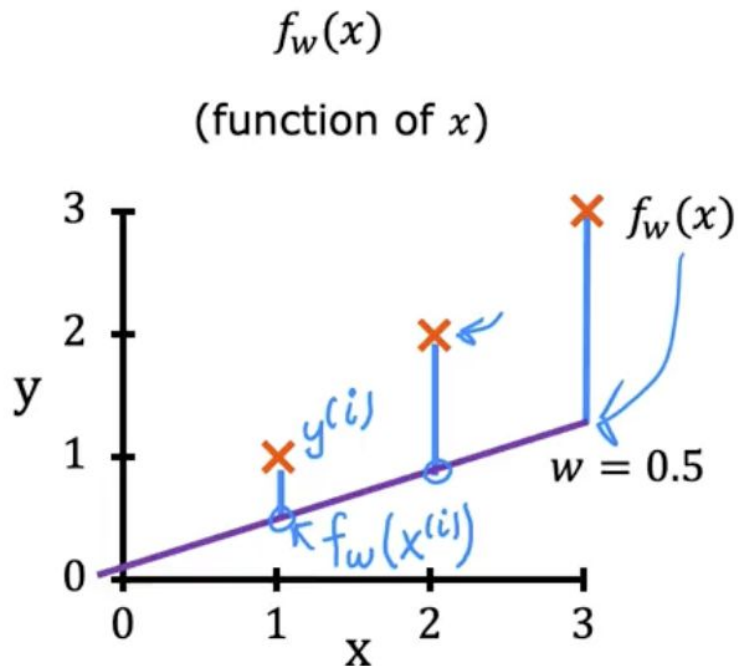
Regressão



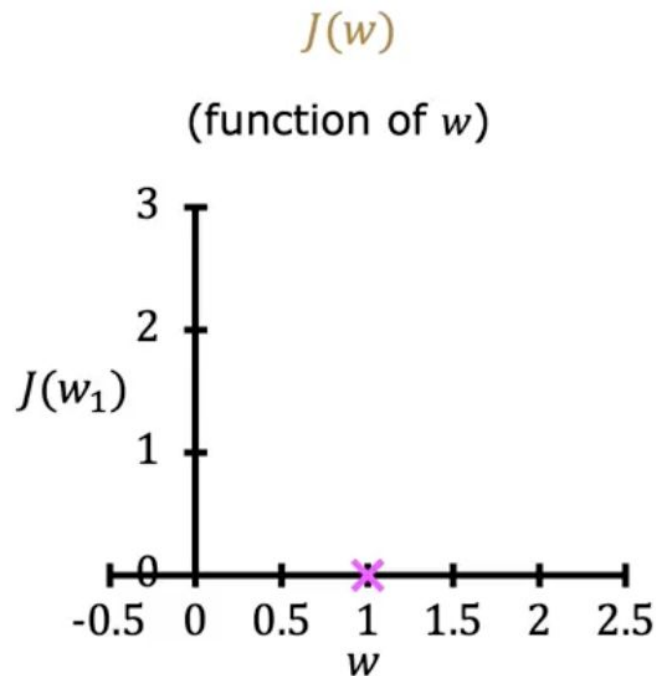
$$J(0.5) = (0.5-1)^2 + (1-2)^2$$



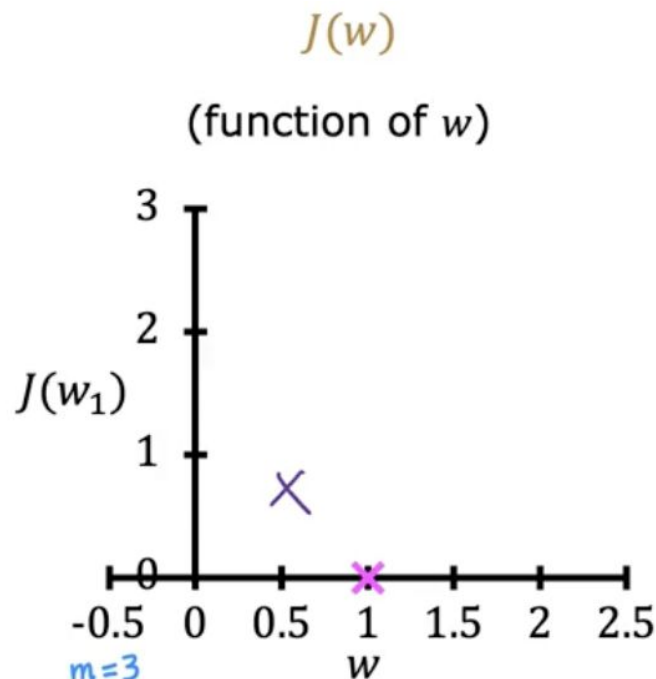
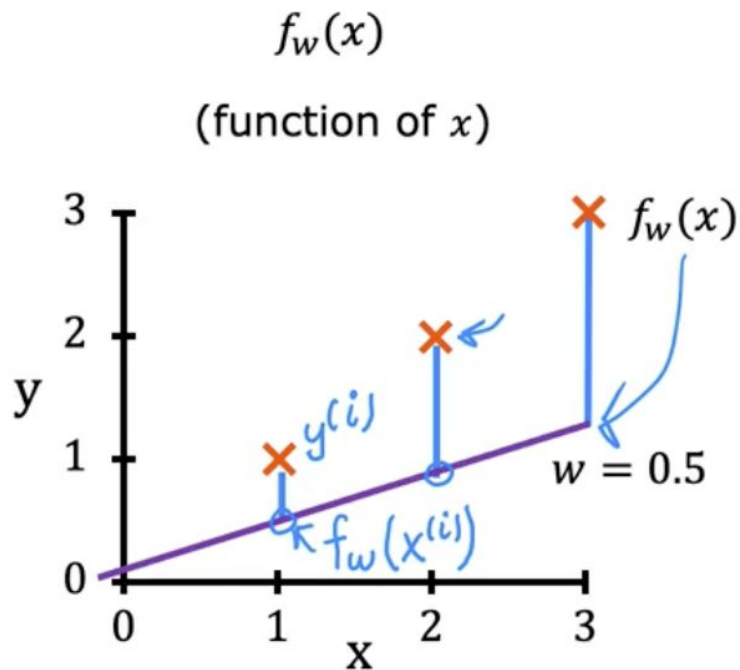
Regressão



$$J(0.5) = (0.5-1)^2 + (1-2)^2 + (1.5-3)^2 \quad [3.5]$$



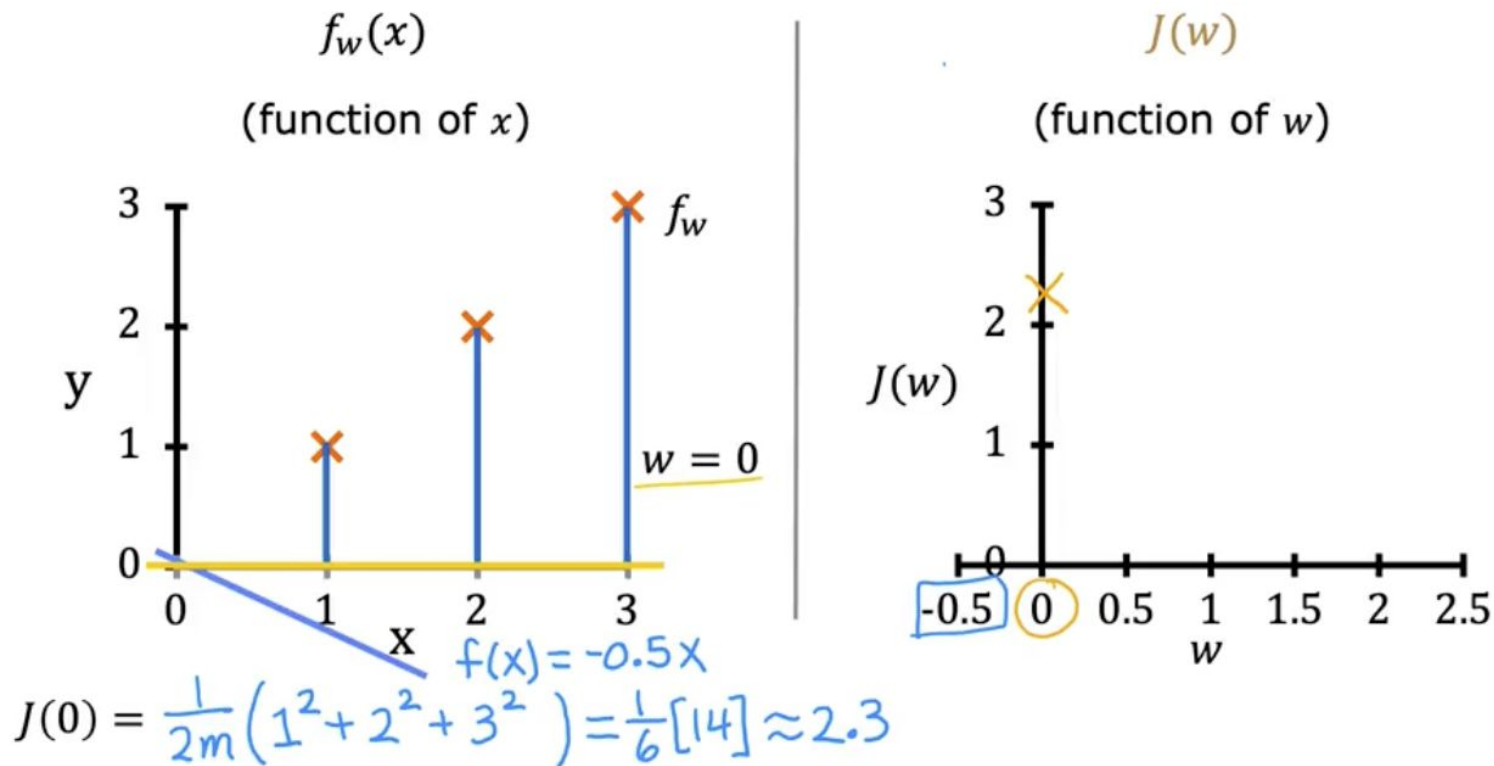
Regressão



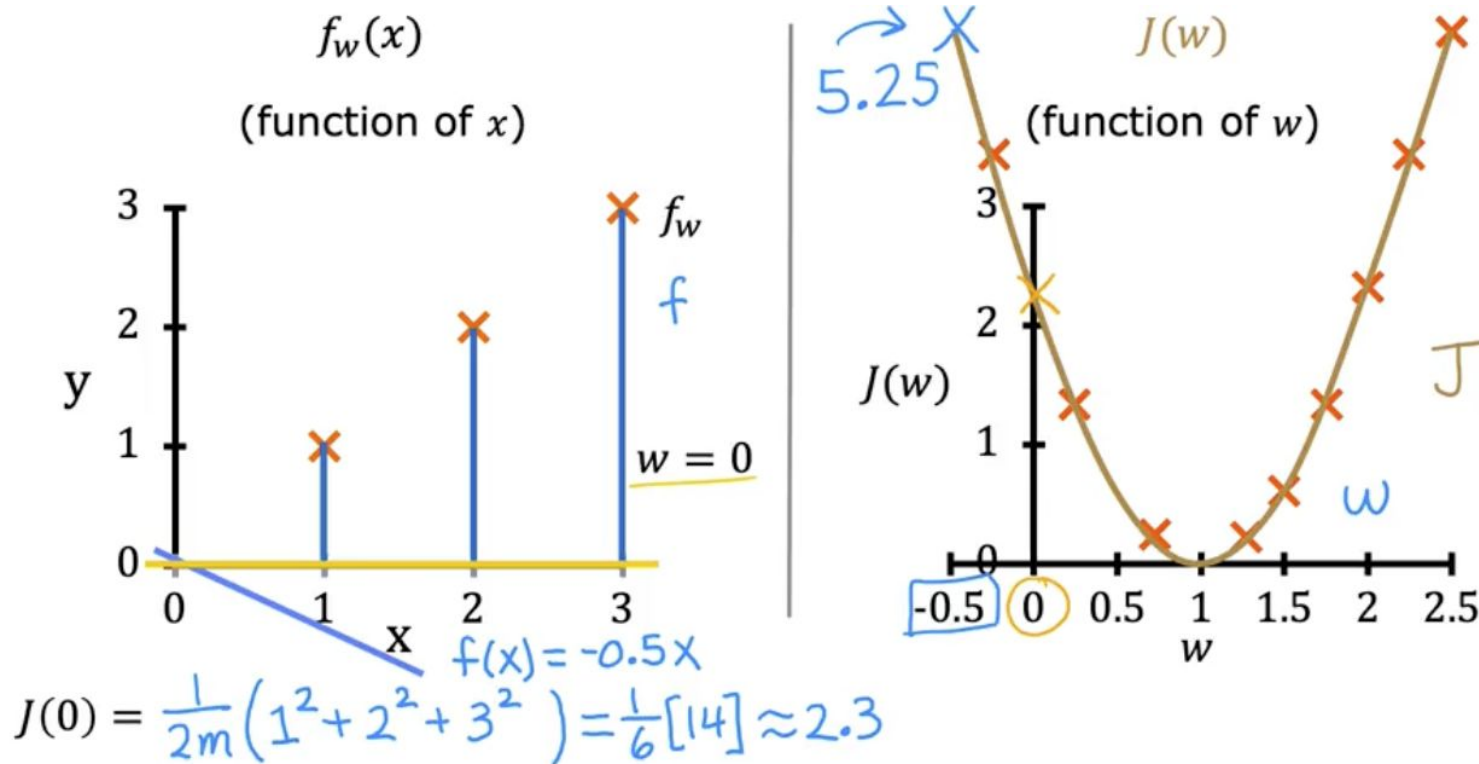
$$J(0.5) = \frac{1}{2m} \left[(0.5-1)^2 + (1-2)^2 + (1.5-3)^2 \right] = \frac{1}{2 \times \underline{3}} [3.5] = \frac{3.5}{6} \approx 0.58$$

$m=3$

Regressão



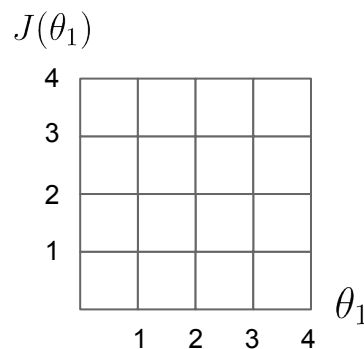
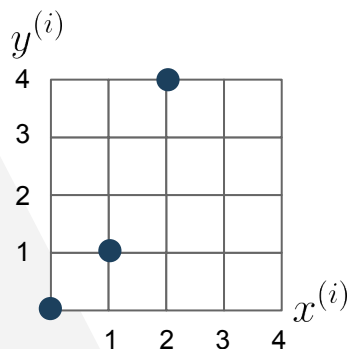
Regressão



Exercício



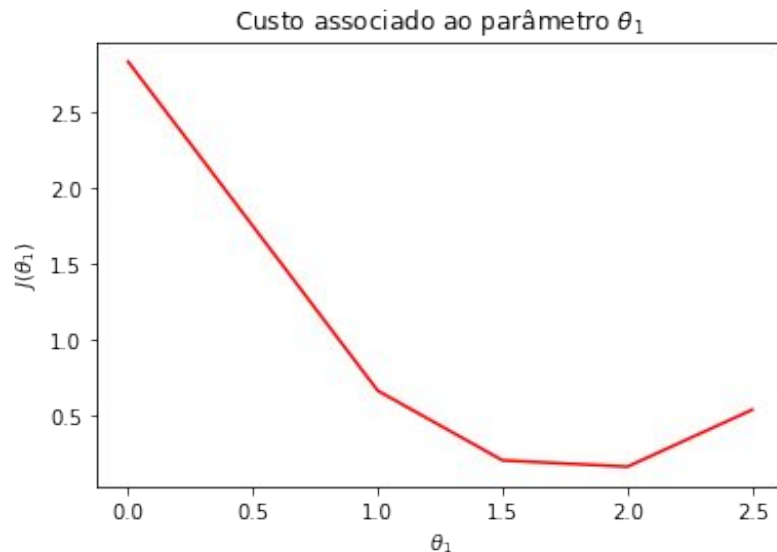
- ▶ Dado o seguinte conjunto de dados e os parâmetros, calcule o custo das predições:
 - ▶ $\Theta_0 = 0, \Theta_1 = 0$
 - ▶ $\Theta_0 = 0, \Theta_1 = 0.5$
 - ▶ $\Theta_0 = 0, \Theta_1 = 1$
 - ▶ $\Theta_0 = 0, \Theta_1 = 1.5$
 - ▶ $\Theta_0 = 0, \Theta_1 = 2$
 - ▶ $\Theta_0 = 0, \Theta_1 = 2.5$
- ▶ Preencha os gráficos a seguir:





Podemos observar que a função de custo é **convexa**

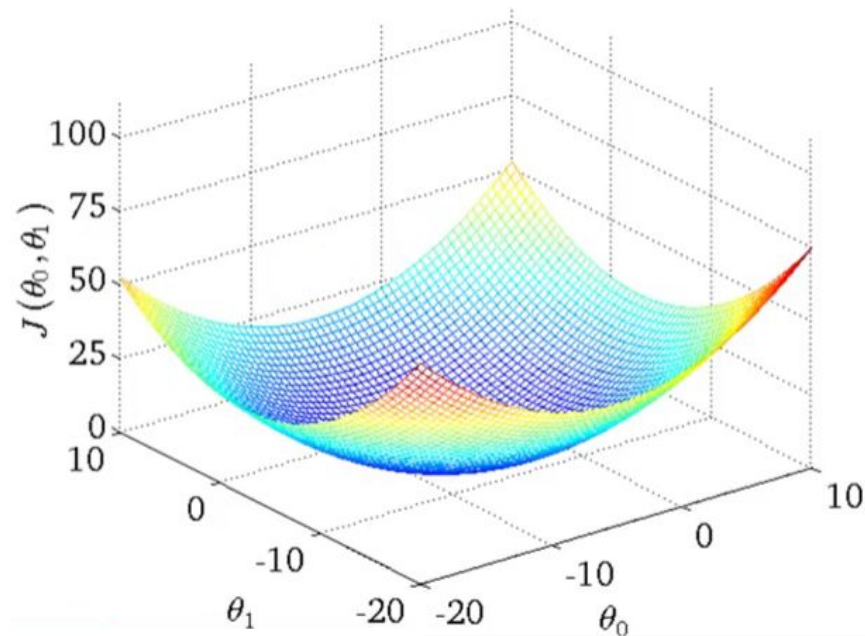
Para um
parâmetro





Podemos observar que a função de custo é **convexa**

Para dois
parâmetros





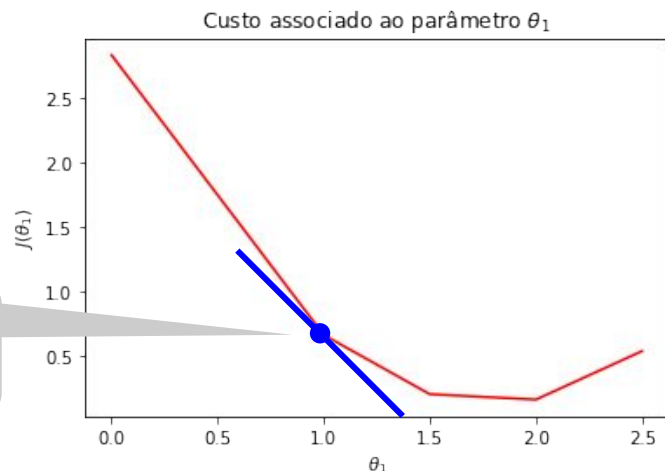
- ▶ Parâmetros têm valores iniciais arbitrários
- ▶ Como minimizar a função de custo?
 - ▷ Atualizando os pesos
- ▶ Como saber **quando** e **quanto** aumentar ou diminuir os valores dos parâmetros?
 - ▷ Devemos alterar os valores continuamente até um determinado critério de parada (convergência)





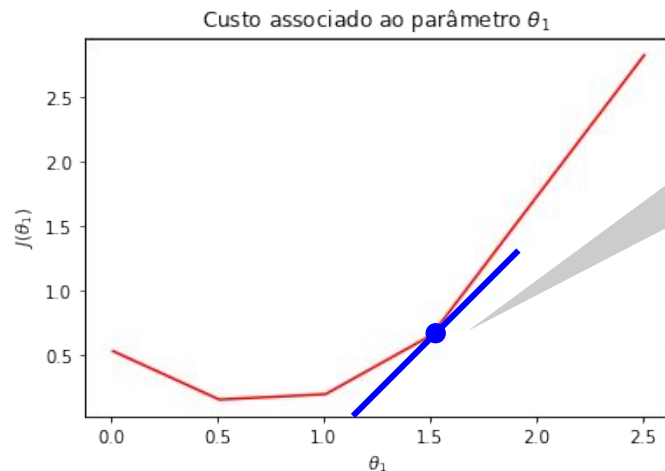
- ▶ A derivada parcial verifica como a função de custo se comporta quando modificamos o valor de um parâmetro
- ▶ Se a inclinação é **negativa**, o valor do parâmetro deve **aumentar**

Reta tangente a
função de custo





- ▶ A derivada parcial verifica como a função de custo se comporta quando modificamos o valor de um parâmetro
- ▶ Se a inclinação é **positiva**, o valor do parâmetro deve **diminuir**



Reta tangente a função de custo



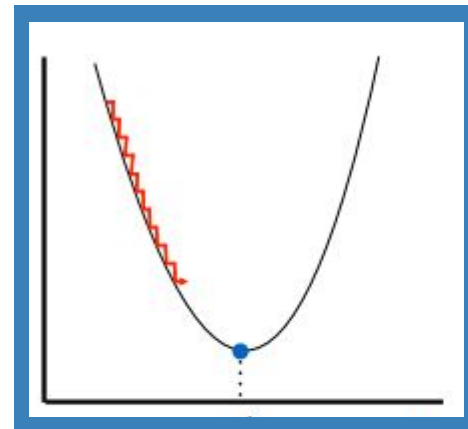
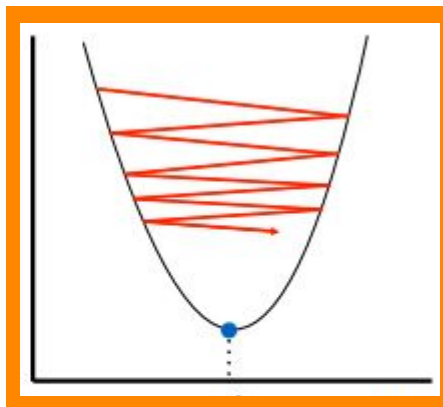


- ▶ A derivada parcial dá a **direção** que um peso deve ser atualizado e o quanto ele está **errado** em relação a um ajuste ideal
- ▶ Porém, podemos amenizar o cálculo do quanto está errado através de uma **taxa de aprendizado**
- ▶ Permite um maior controle sobre o aprendizado realizado pelo modelo





- ▶ Se a taxa de aprendizado for muito **alta**, podemos passar “por cima” do ótimo global
- ▶ Se a taxa de aprendizado for muito **baixa**, podemos demorar muito para convergir
- ▶ Não existe almoço grátis!





- ▶ Logo, para fazer a atualização de pesos, utilizamos o **gradiente descendente**
 - ▶ **Gradiente** é como chamamos o grupo de derivadas de diversos parâmetros
 - ▶ Utilizamos uma **taxa de aprendizado** para ditar o passo em torno de **melhores resultados**





O j-ésimo
parâmetro
recebe

O antigo valor
do j-ésimo
parâmetro,
menos

A derivada da função de
custo em relação ao j-
ésimo parâmetro

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

vezes a taxa de
aprendizado





A atualização dos pesos deve ser **simultânea!**

- ▶ Não atualize o valor do parâmetro $j + 1$, com base no valor **atualizado** do parâmetro j
- ▶ Em vez disso, calcule antes todos os novos valores de parâmetros, e **só então** faça as atribuições

$$\text{temp0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$

$$\text{temp1} := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$

$$\theta_0 := \text{temp0}$$

$$\theta_1 := \text{temp1}$$





- ▶ Para a função de custo da regressão linear, a atualização de pesos (com derivadas calculadas) é:

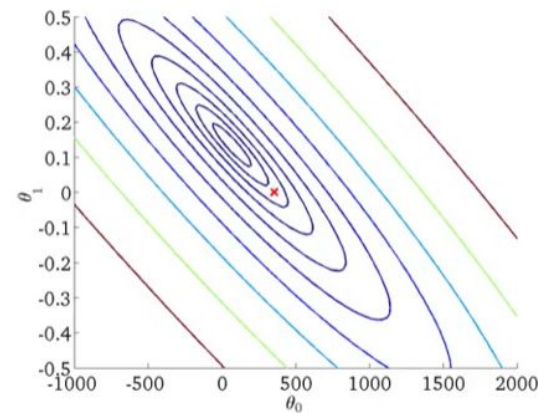
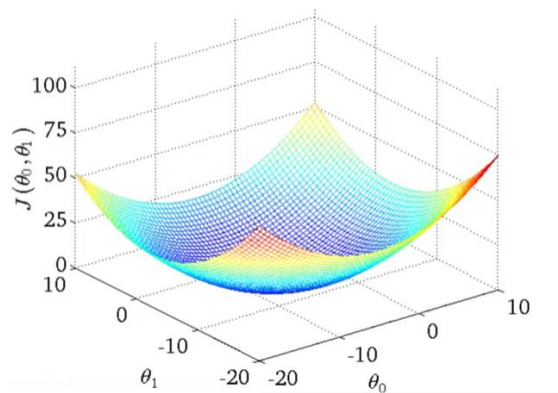
$$\theta_0 := \theta_0 - \alpha \frac{1}{N} \sum_{i=1}^N (h_{\theta}(x^{(i)}) - y^{(i)})$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{N} \sum_{i=1}^N (h_{\theta}(x^{(i)}) - y^{(i)}) \cdot x^{(i)}$$





► Achatando o *landscape*

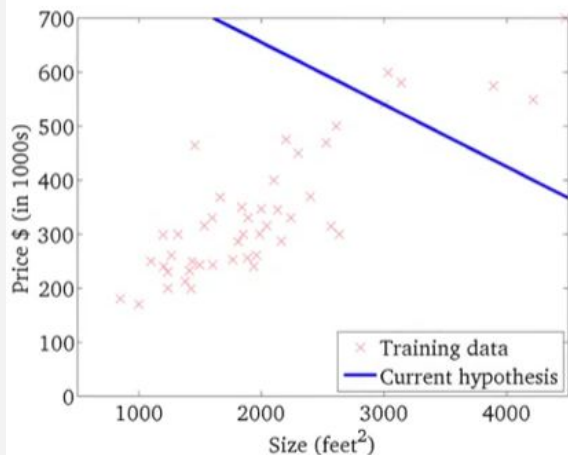




► Gradiente Descendente em ação

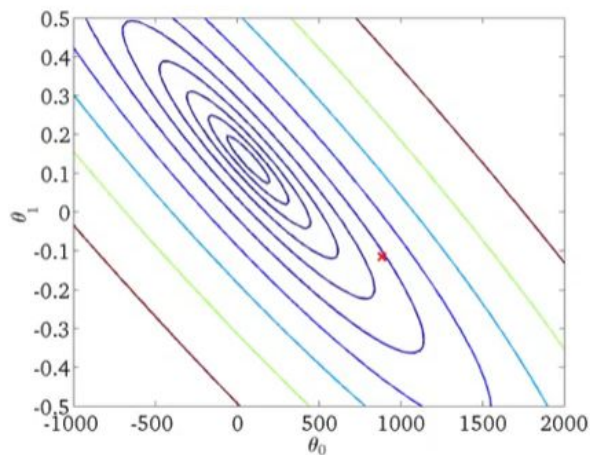
$$h_{\theta}(x)$$

(for fixed θ_0, θ_1 , this is a function of x)



$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)

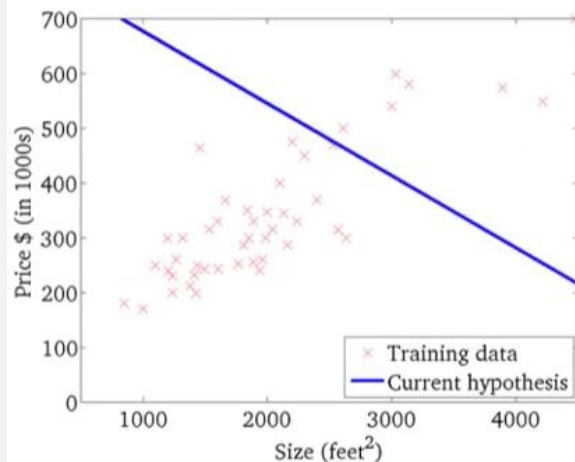




► Gradiente Descendente em ação

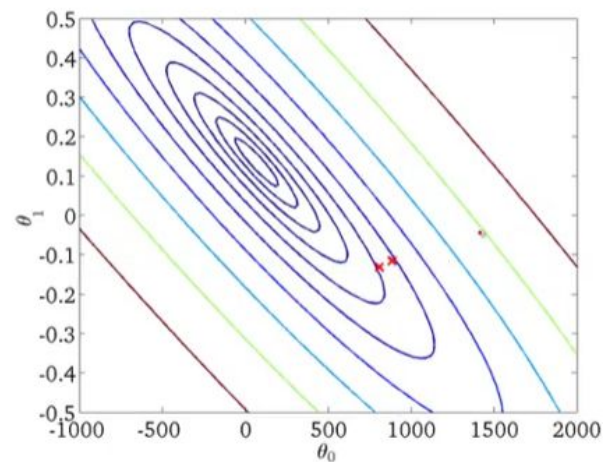
$$h_{\theta}(x)$$

(for fixed θ_0, θ_1 , this is a function of x)



$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)

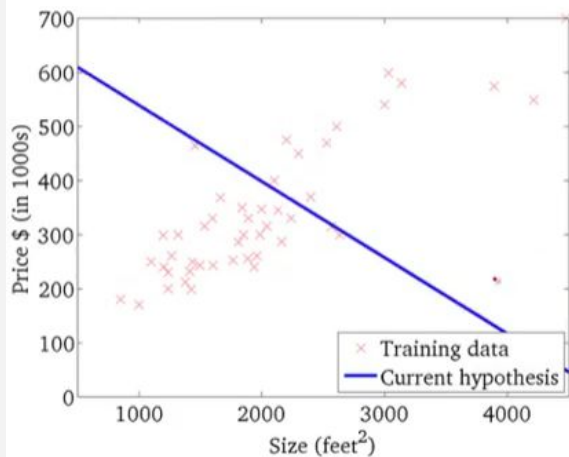




► Gradiente Descendente em ação

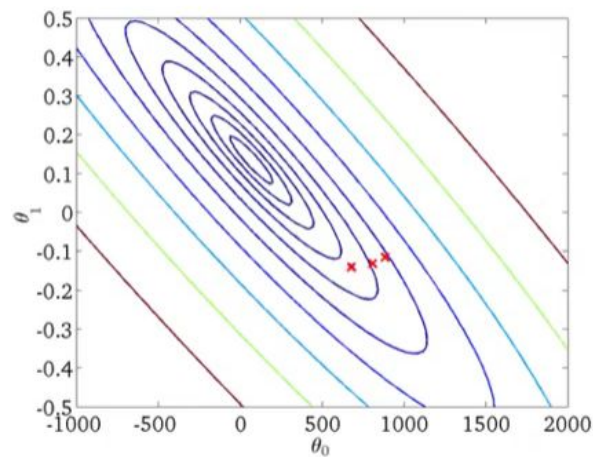
$$h_{\theta}(x)$$

(for fixed θ_0, θ_1 , this is a function of x)



$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)

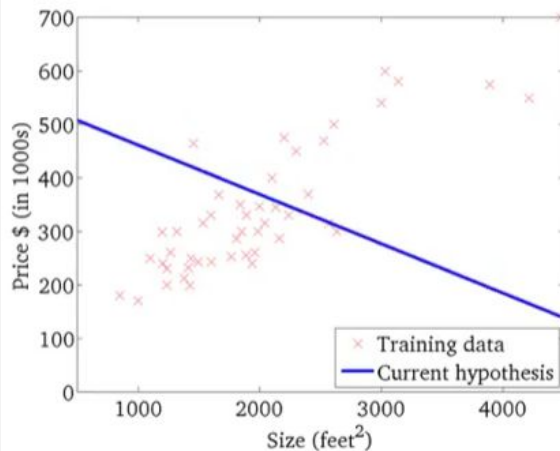




► Gradiente Descendente em ação

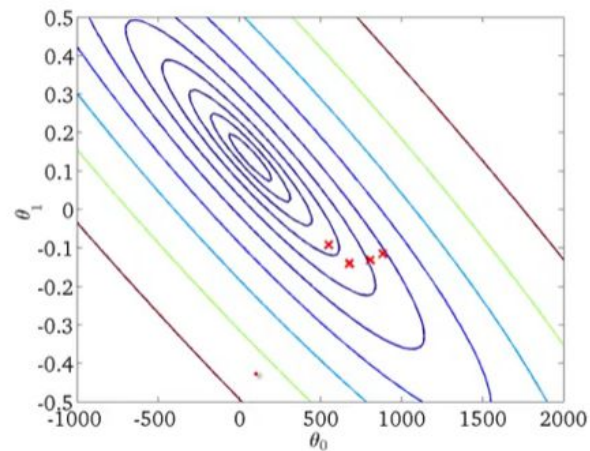
$$h_{\theta}(x)$$

(for fixed θ_0, θ_1 , this is a function of x)



$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)

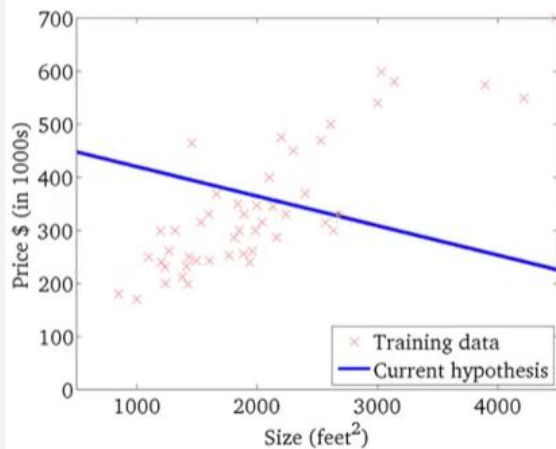




► Gradiente Descendente em ação

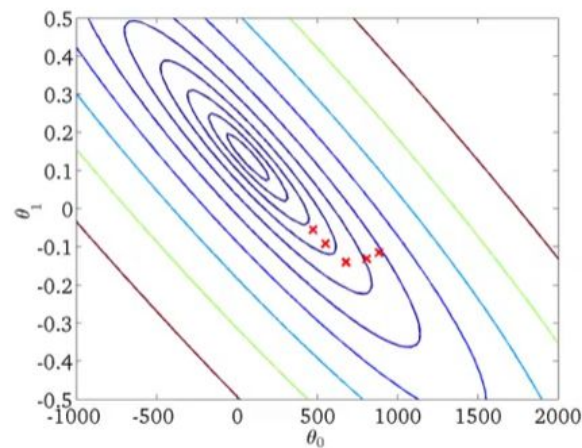
$$h_{\theta}(x)$$

(for fixed θ_0, θ_1 , this is a function of x)



$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)

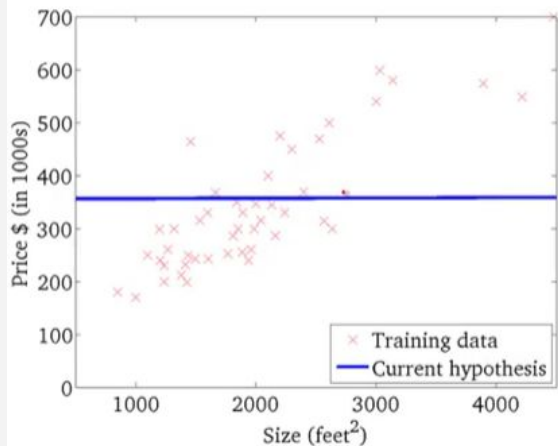




► Gradiente Descendente em ação

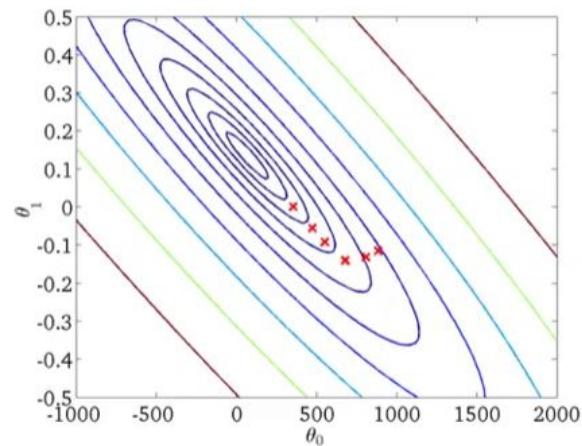
$$h_{\theta}(x)$$

(for fixed θ_0, θ_1 , this is a function of x)



$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)

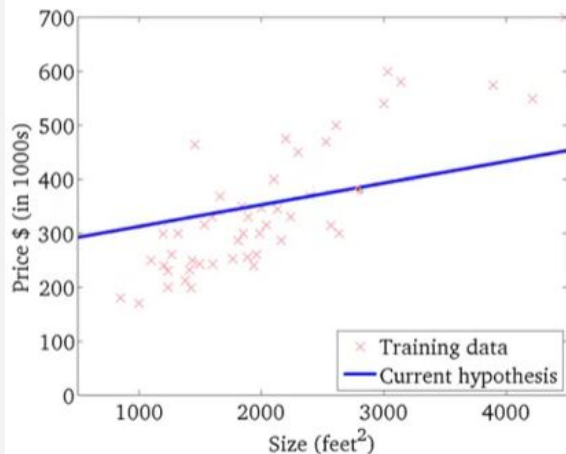




► Gradiente Descendente em ação

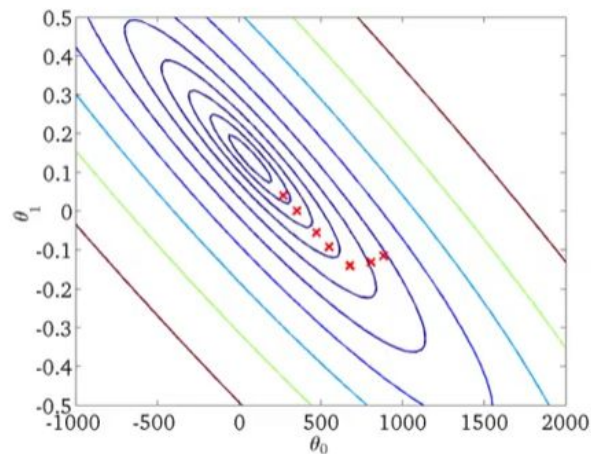
$$h_{\theta}(x)$$

(for fixed θ_0, θ_1 , this is a function of x)



$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)

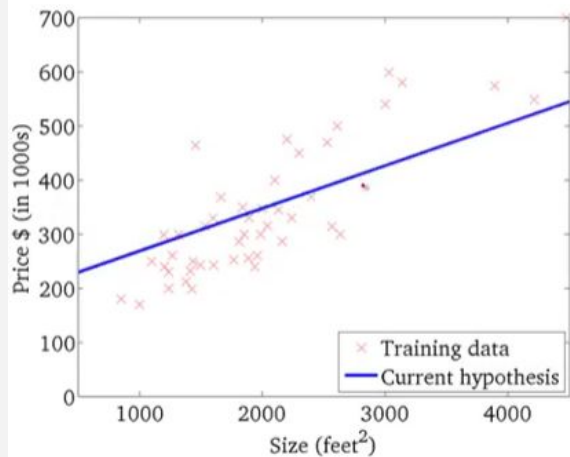




► Gradiente Descendente em ação

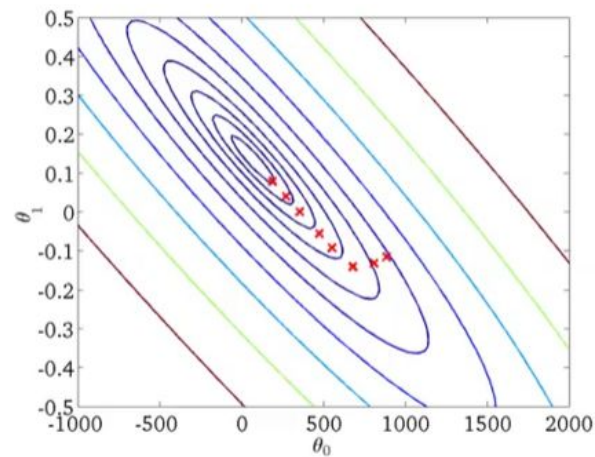
$$h_{\theta}(x)$$

(for fixed θ_0, θ_1 , this is a function of x)



$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)

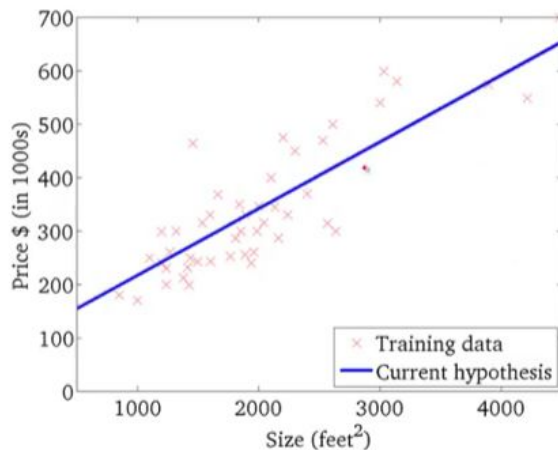




► Gradiente Descendente em ação

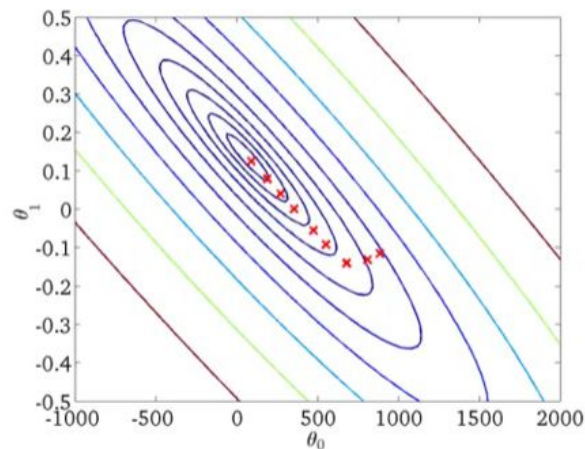
$$h_{\theta}(x)$$

(for fixed θ_0, θ_1 , this is a function of x)



$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)

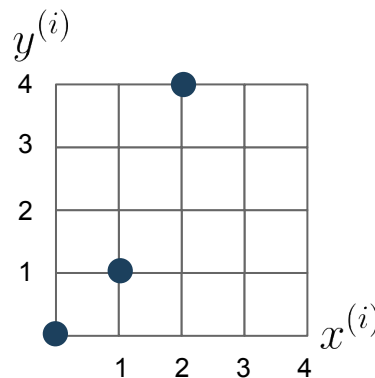


Exercício



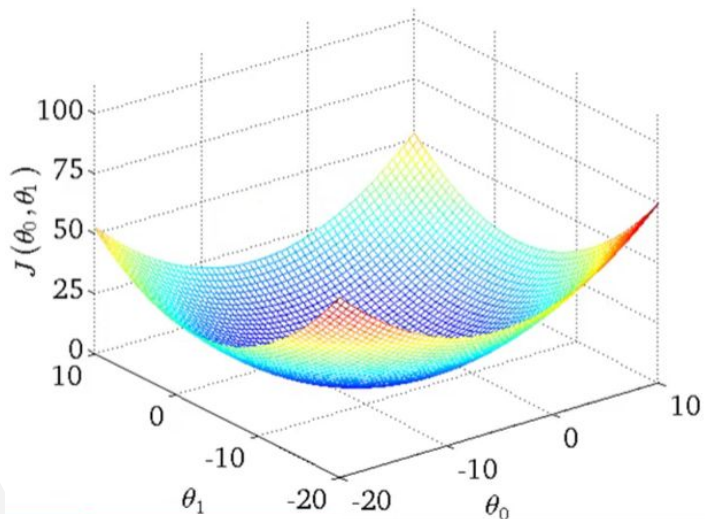
- ▶ Dados os seguintes parâmetros, realize a atualização de pesos:

- ▶ $\alpha = 0.5$
- ▶ $\Theta_0 = 0.1$
- ▶ $\Theta_1 = 1$
- ▶ $X = [0, 1, 2]$
- ▶ $y = [0, 1, 4]$



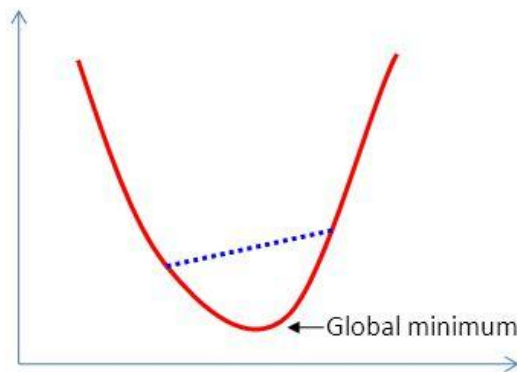


- ▶ Em um problema de regressão linear, a função de custo é **sempre convexa**

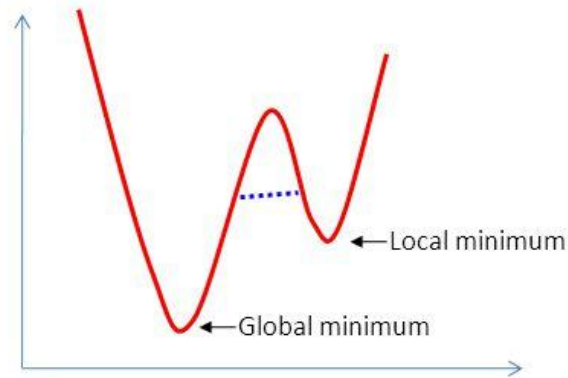




- ▶ Uma função é dita convexa se o mínimo global situa-se abaixo de uma linha reta que conecta quaisquer dois segmentos da função



Convex function



Non-convex function



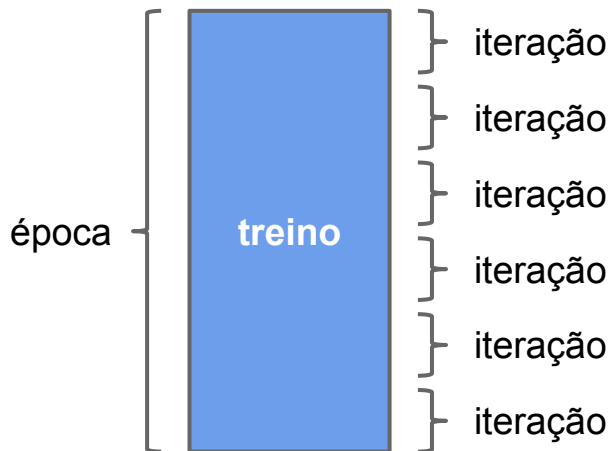


- ▶ A atualização de pesos se dá após a passada de um lote (*batch*) de dados pela **hipótese**
 - ▷ O tamanho do batch pode variar de 1 (apenas uma instância) a N (tamanho inteiro do conjunto de treino)
 - ▷ Utiliza-se múltiplos de 8, para otimizações em *frameworks*
- ▶ A passada de instâncias pode ser ainda **estocástica**
 - ▷ A ordem de passada das instâncias é aleatória





- ▶ Passada de um batch: iteração
- ▶ Passada do conjunto de treino: época





- ▶ O processo de aprendizado é parado quando alcançamos um número máximo de passadas pelo conjunto de treino (“épocas”) ou quando atingimos a convergência (diferença entre o último custo e o atual é menor do que 0.001, por exemplo)
- ▶ O Gradiente Descendente pode convergir mesmo que a taxa de aprendizado seja fixa (mas a mesma não pode ser muito alta)
- ▶ Pode ser decrementada no decorrer do treino
- ▶ Para uma escolha específica da função de custo $J(\Theta_0, \Theta_1)$ utilizada na **regressão linear**, não existe ótimo local além do ótimo global



Exercício



- ▶ Utilize a [regressão linear](#) do scikit-learn para prever o preços de imóveis (dataset `houses.csv`)





Leitura recomendada:

- ▶ Apêndice D de Introduction to Data Mining

