Université Bretagne Sud

Modèle de régression linéaire

Master 1

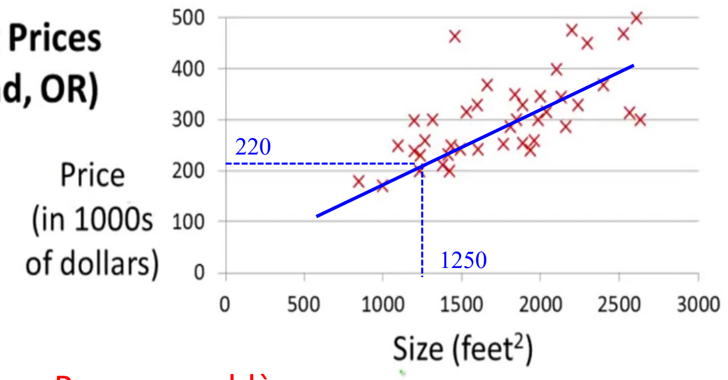
Traitement Numérique des Données

Sylvie Gibet

1

1

Housing Prices (Portland, OR)



Pour ce problème :

- Apprentissage supervisé
  - donner la réponse juste pour chaque exemple contenu dans les données (valeur réelle, valeur discrète)
- 1. Problème de régression
  - prédire la valeur réelle de sortie
- 2. Classification
  - valeur de sortie discrète

2

2

Machine learning: deux étapes

- Deux étapes dans le processus de machine learning
  - Phase d'entraînement (training) : l'algorithme apprend sur une partie des données
  - Phase de test : test sur une autre partie des données

3

3

Training set (apprentissage) prix des maisons

Taille en m2	Prix en 1000 * €
189	386
127	106
138	265
76	149
…	

- Objectif : apprendre à partir de ces données comment prédire les coûts des maisons
- Notation
  - m: nombre d'exemples d'apprentissage
  - x: variables d'entrée / features
  - y: variable de sortie/ target variable

  (x, y) : un exemple d'apprentissage

  ($x^{(i)}$, $y^{(i)}$) : i^{th} exemple de l'apprentissage (i^{th} ligne)

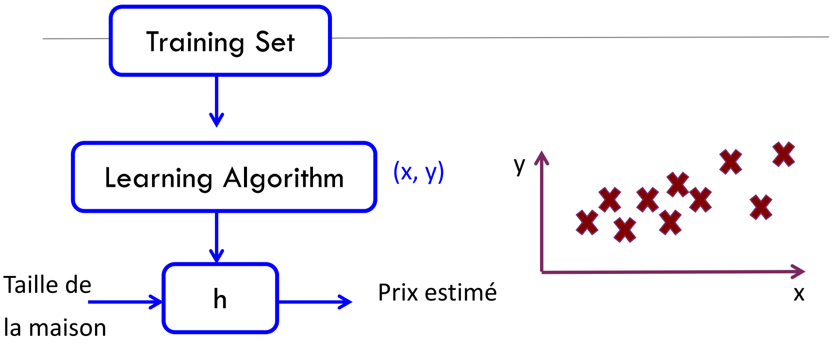
$x^{(1)}$= 189

$x^{(2)}$= 127

$y^{(1)}$= 386

4

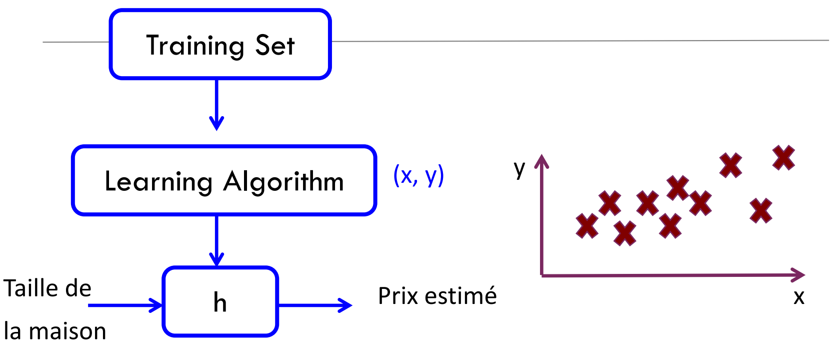
4



x: input h : prédiction ŷ : valeur estimée

5

5

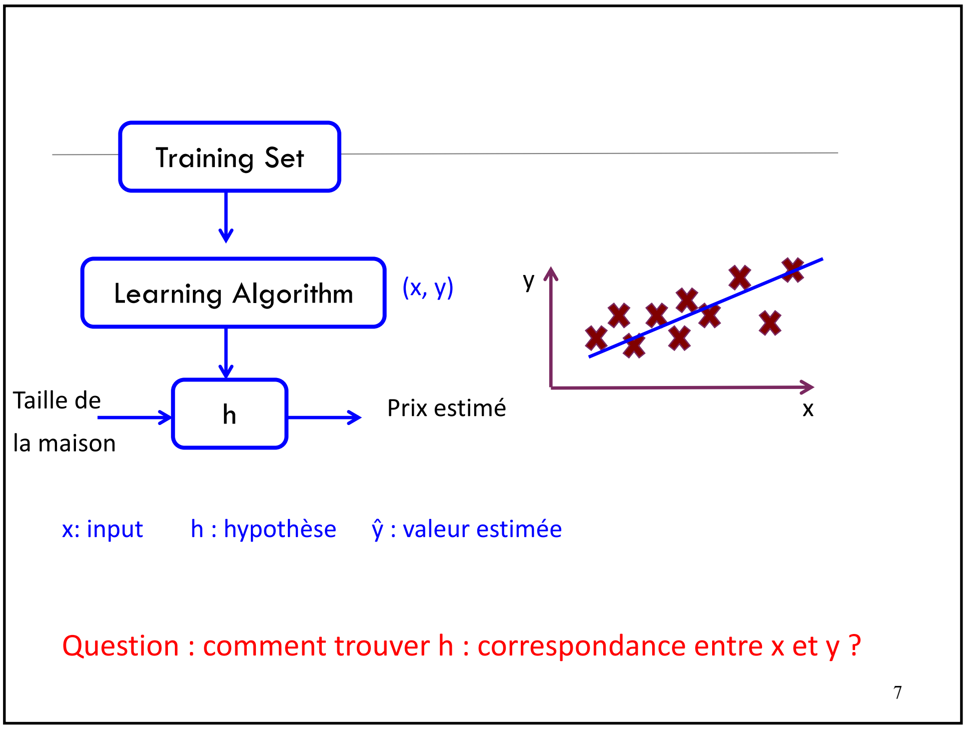


x: input h : hypothèse ŷ : valeur estimée

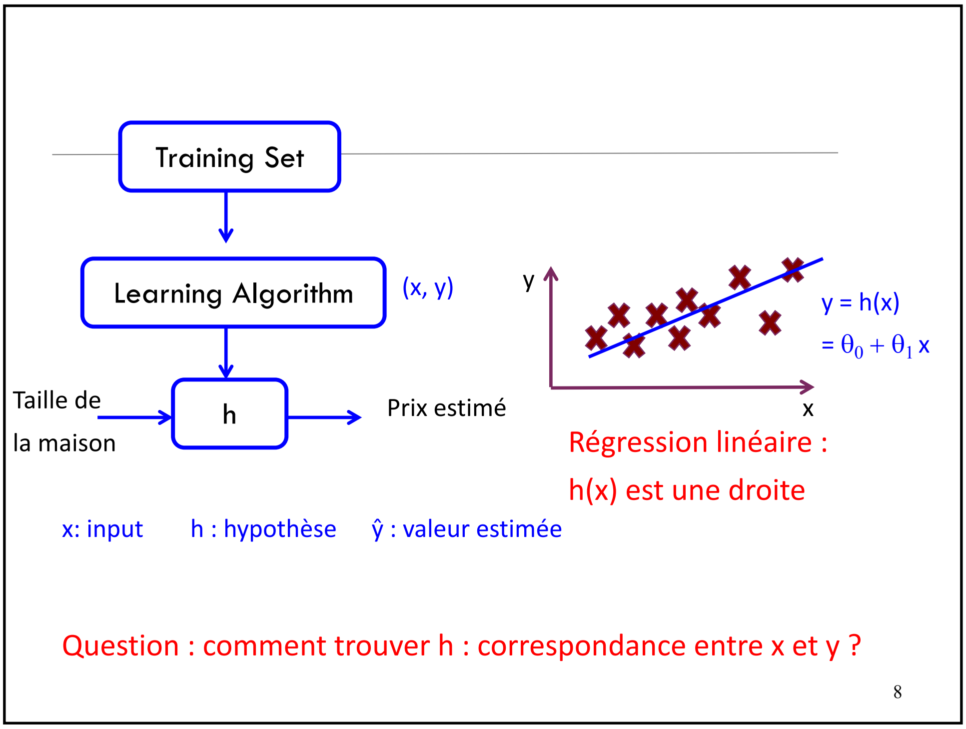
Question : comment trouver h : correspondance entre x et y ?

6

6


Training Set
Learning Algorithm
(x, y)
y
x
Taille de
la maison
h
Prix estimé
x: input
h : hypothèse
ŷ : valeur estimée
Question : comment trouver h : correspondance entre x et y ?
7

7


Training Set
Learning Algorithm
(x, y)
y
x
y = h(x)
$= \theta_0 + \theta_1 x$
Taille de
la maison
h
Prix estimé
Régression linéaire :
h(x) est une droite
x: input
h : hypothèse
ŷ : valeur estimée
Question : comment trouver h : correspondance entre x et y ?
8

8

Training set (apprentissage)
prix des maisons

Taille en m2	Prix en 1000 * €
189	386
127	106
138	265
76	149
…	

Comment choisir la fonction de mapping h ?

Hypothèse : on prend une droite (fonction linéaire de la variable x) :

$$\hat{y} = h_\theta(x) = \theta_0 + \theta_1 x$$

x : variable d'entrée de la régression linéaire, ici la taille des maisons

$\hat{y}$: variable estimée (sortie) : prix des maisons

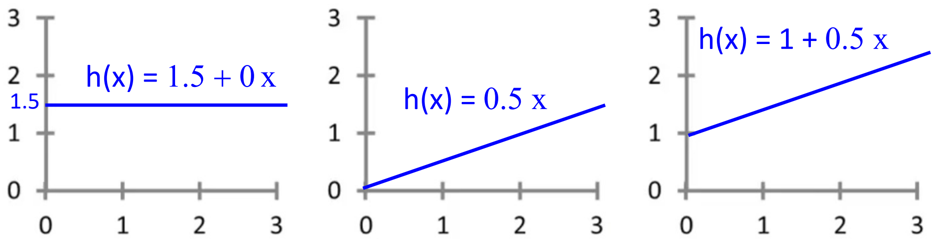
θ_i : paramètres (2 paramètres, θ_0 et θ_1)

Comment choisir les paramètres θ_0 et θ_1 ?

9

9

$$h_\theta(x) = \theta_0 + \theta_1 x$$

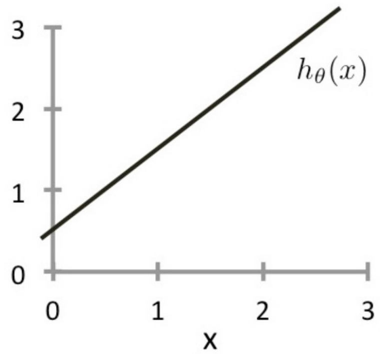


$\theta_0 = 1.5$, $\theta_1 = 0$ — $\theta_0 = 0$, $\theta_1 = 0.5$ — $\theta_0 = 1$, $\theta_1 = 0.5$

10

10

Exercice : Soit le diagramme ci-dessous : $h_\theta(x) = \theta_0 + \theta_1 x$.
Quelles sont les valeurs de θ_0 et θ_1 ?



- A. $\theta_0 = 0, \theta_1 = 1$
- B. $\theta_0 = 0.5, \theta_1 = 1$
- C. $\theta_0 = 1, \theta_1 = 0.5$
- D. $\theta_0 = 1, \theta_1 = 1$

11

11

Fonction coût

12

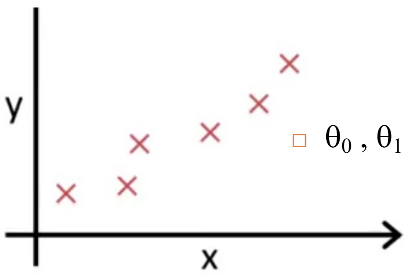
12

Fonction coût : définition

- Fonction qui va nous permettre d'évaluer la qualité de la droite qui approxime des données
- Peut s'exprimer comme une distance entre les données et la droite
  - Se ramène à la somme des distances entre chaque exemple contenu dans les données et sa "projection" sur la droite

13

13

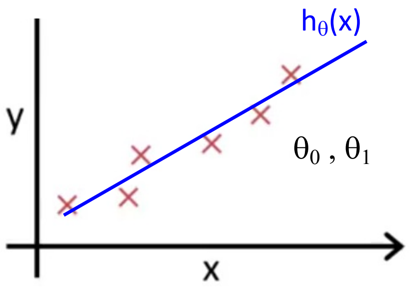


- θ_0 , θ_1

- Idée : on choisit θ_0 et θ_1 de telle façon que la prédiction $y = h_\theta(x)$ soit "proche" de y pour nos exemples de training (x, y)

14

14

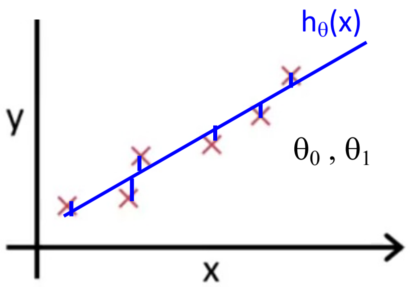


- On veut que la valeur $(h_\theta(x) - y)$ soit aussi petite que possible pour toutes les valeurs du training set

- Idée : on choisit θ_0 et θ_1 de telle façon que la prédiction $y = h_\theta(x)$ soit “proche” de y pour nos exemples de training (x, y)

15

15

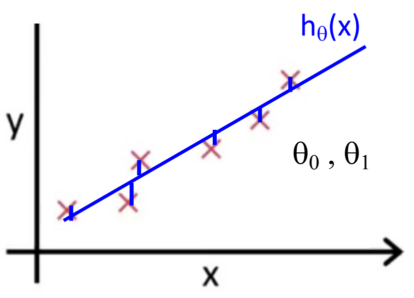


- On veut que la valeur $(h_\theta(x) - y)$ soit aussi petite que possible pour toutes les valeurs du training set

- Idée : on choisit θ_0 et θ_1 de telle façon que la prédiction $y = h_\theta(x)$ soit “proche” de y pour nos exemples de training (x, y)

16

16



- Idée : on choisit θ_0 et θ_1 de telle façon que la prédiction y = $h_\theta(x)$ soit "proche" de y pour nos exemples de training (x, y)

- On veut que la valeur

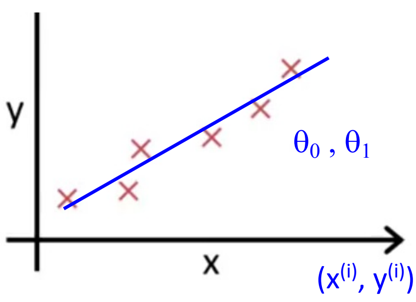
($h_\theta(x)$ – y) soit aussi petite que possible pour toutes les valeurs du training set

- Une façon de faire est de minimiser le coût quadratique :

$(h_\theta(x) - y)^2$ pour tout le training set, suivant les paramètres θ_0, θ_1

17

17



- Idée : on choisit θ_0 et θ_1 de telle façon que la prédiction y = $h_\theta(x)$ soit "proche" de y pour nos exemples de training (x, y)

- On veut minimiser / (θ_0,θ_1):

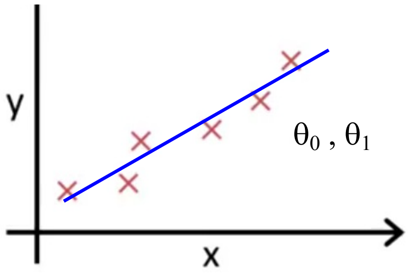
$$\frac{1}{2m} \cdot \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

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

$J(\theta_0,\theta_1)$: Fonction coût

18

18



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

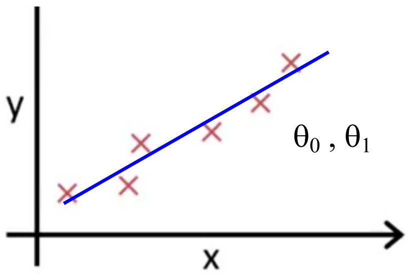
Minimiser J(θ_0 , θ_1) suivant θ_0 , θ_1

J(θ_0 , θ_1) : fonction coût encore appelée erreur quadratique

- Idée : on choisit θ_0 et θ_1 de telle façon que la prédiction y = h_θ(x) soit "proche" de y pour nos exemples de training (x, y)

19

19



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

Minimiser J(θ_0 , θ_1) suivant θ_0 , θ_1

J(θ_0 , θ_1) : fonction coût encore appelée erreur quadratique

Bonne fonction d'erreur pour beaucoup de problèmes de régression

- Idée : on choisit θ_0 et θ_1 de telle façon que la prédiction y = h_θ(x) soit "proche" de y pour nos exemples de training (x, y)

20

20

Université Bretagne Sud

Fonction de coût
Intuition I
un seul paramètre θ_1

21

21

- <u>Hypothèse</u> : $h_\theta(x) = \theta_0 + \theta_1 x$ avec

  x : input, $y = h_\theta(x)$: output

- <u>Paramètres</u> : θ_0, θ_1

- <u>Fonction coût</u> : $$J(\theta_0 \ , \theta_1) = \frac{1}{2m} \cdot \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

- <u>But</u> : $\underset{\theta_0, \theta_1}{\text{minimiser}} \ J(\theta_0, \theta_1)$

22

22

- Hypothèse : $y = h_\theta(x) = \theta_0 + \theta_1 x$
- Paramètres : θ_0, θ_1
- Fonction coût :

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

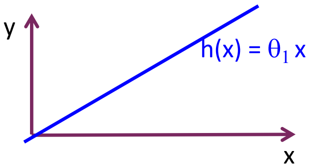
- But : minimiser $J(\theta_0 , \theta_1)$

θ_0,θ_1

Problème simplifié

- $h_\theta(x) = \theta_1 x$

$\theta_0 = 0$



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

$\theta_1 x^{(i)}$

minimiser $J(\theta_1)$

θ_1

23

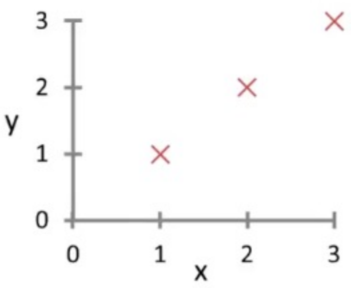
23

$h_\theta(x)$

(pour θ_1 fixe, c'est une fonction de x)

$J(\theta_1)$

(fonction du paramètre θ_1)



24

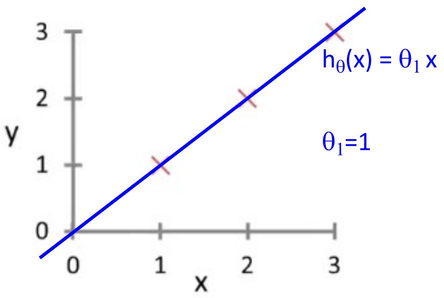
24

$h_\theta(x)$

(pour θ_1 fixe, c'est une fonction de x)

$J(\theta_1)$

(fonction du paramètre θ_1)



25

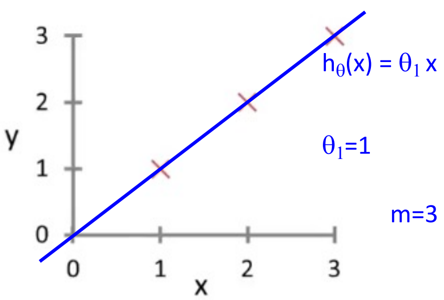
25

$h_\theta(x)$

(pour θ_1 fixe, c'est une fonction de x)

$J(\theta_1)$

(fonction du paramètre θ_1)



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

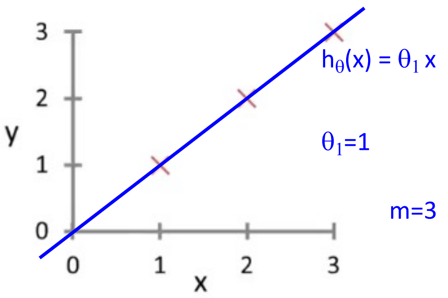
$$\frac{1}{2m}\cdot\sum_{i=1}^{m}(\theta_1 x^{(i)} - y^{(i)})^2 = \frac{1}{2m}\cdot(0^2+0^2+0^2)=0$$

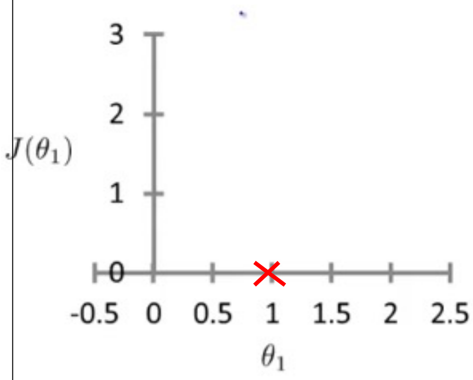
26

26

$h_\theta(x)$
(pour θ_1 fixe, c'est une fonction de x)

$J(\theta_1)$
(fonction du paramètre θ_1)





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

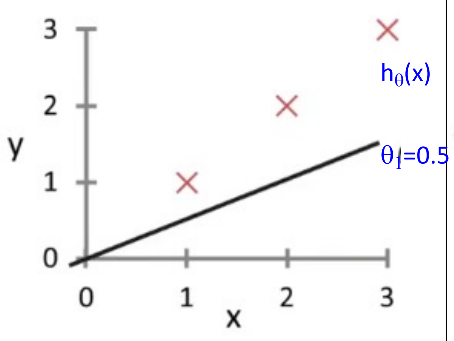
$$\frac{1}{2m} \cdot \sum_{i=1}^{m} (\theta_1 x^{(i)} - y^{(i)})^2 = \frac{1}{2m} \cdot (0^2 + 0^2 + 0^2) = 0$$

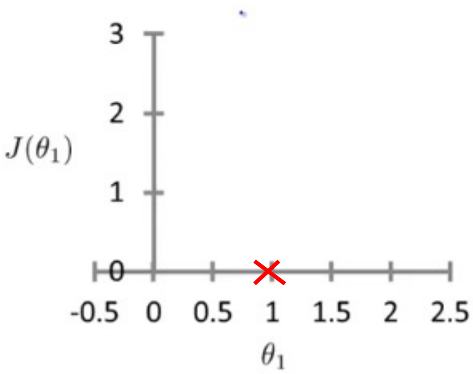
27

27

$h_\theta(x)$
(pour θ_1 fixe, c'est une fonction de x)

$J(\theta_1)$
(fonction du paramètre θ_1)





J(0.5) ?

28

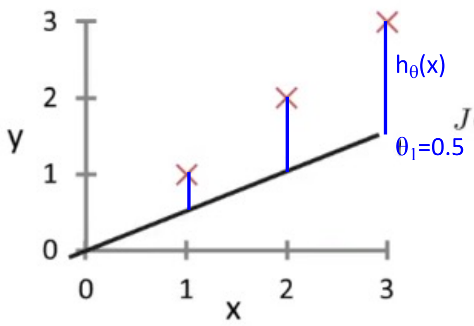
28

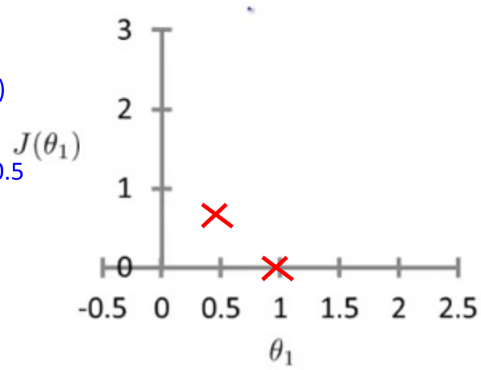
$h_\theta(x)$

(pour θ_1 fixe, c'est une fonction de x)

$J(\theta_1)$

(fonction du paramètre θ_1)





$J(0.5) = 1/2m[(0.5-1)^2 + (1 - 2)^2 + (1.5 - 3)^2]$

$= 1/6\ (3.5) = 3.5/6 = 0.58$

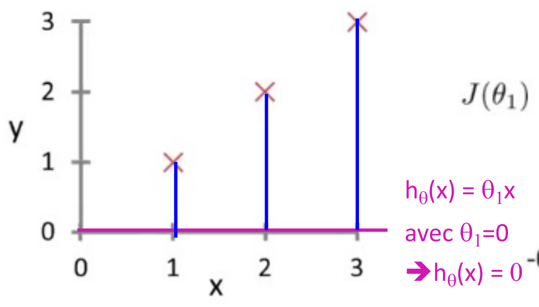
29

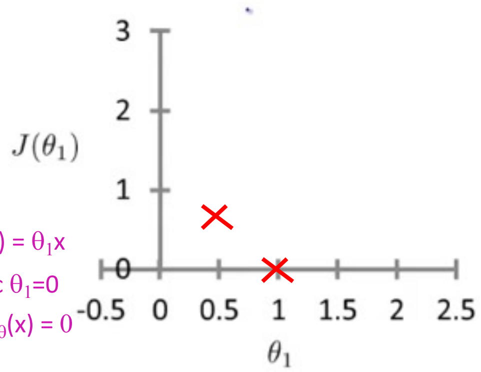
29

Supposons que nous avons un ensemble d'apprentissage de m = 3 exemples, comme sur la figure suivante. Notre hypothèse de représentation est $h_\theta(x) = \theta_1 x$, avec le paramètre θ_1. La fonction de coût $J(\theta_1)$ est :

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

Que vaut J(0) ?





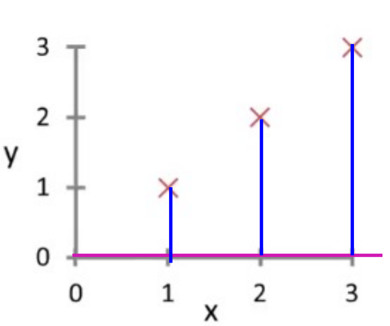
30

30

Supposons que nous avons un ensemble d'apprentissage de m = 3 exemples, comme sur la figure suivante. Notre hypothèse de représentation est $h_\theta(x) = \theta_1 x$, avec le paramètre θ_1. La fonction de coût $J(\theta_1)$ est :

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

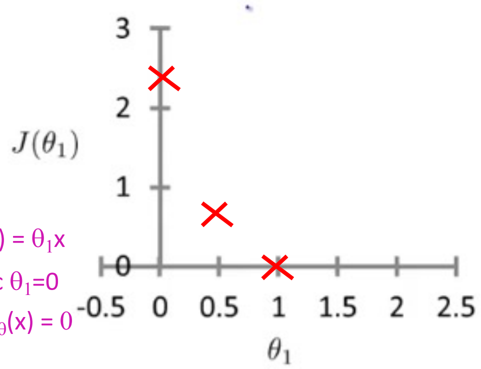
Que vaut J(0) ?



$h_\theta(x) = \theta_1 x$

avec $\theta_1=0$

$\rightarrow h_\theta(x) = 0$



$J(0) = 1/6[(0 - 1)^2 + (0 - 2)^2 + (0 - 3)^2]$

$= 1/6\ (14) = 14/6 = 2.3$

31

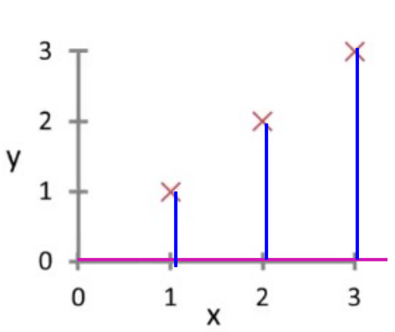
31

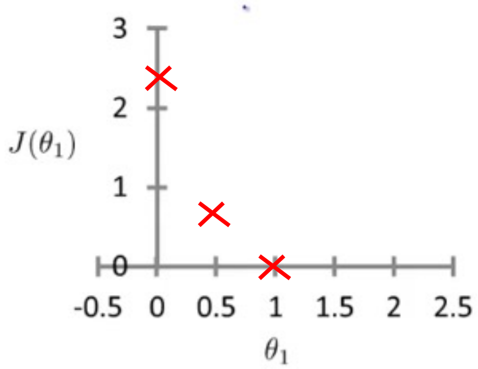
$h_\theta(x)$

(pour θ_1 fixe, c'est une fonction de x)

$J(\theta_1)$

(fonction du paramètre θ_1)





J(1) = 0

J(0.5) = 0.58

J(0) = 2.3

J(-0.5) = 5.25

32

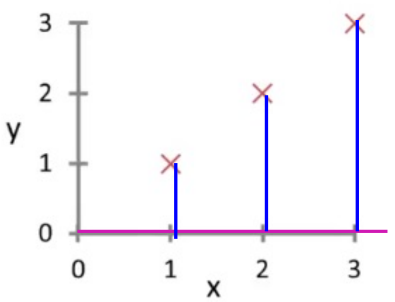
32

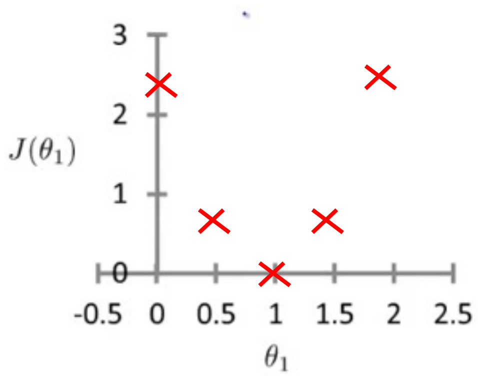
$h_\theta(x)$

(pour θ_1 fixe, c'est une fonction de x)

$J(\theta_1)$

(fonction du paramètre θ_1)





J(1) = 0

J(0.5) = 0.58

J(0) = 2.3

J(-0.5) = 5.25

33

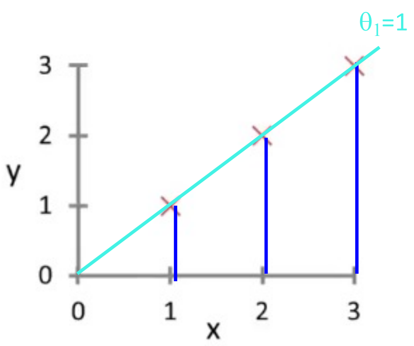
33

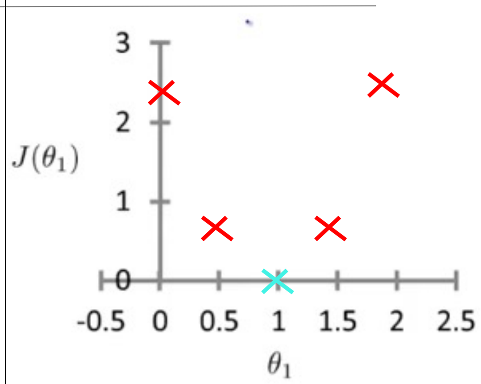
$h_\theta(x)$

(pour θ_1 fixe, c'est une fonction de x)

$J(\theta_1)$

(fonction du paramètre θ_1)





$h_{\theta 1}(x) = x$ $(\theta_1 = 1)$

J(1) = 0

$\underset{\theta_1}{\text{minimiser}}\ J(\theta_1)$

34

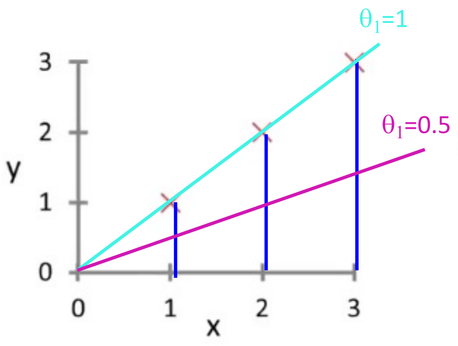
34

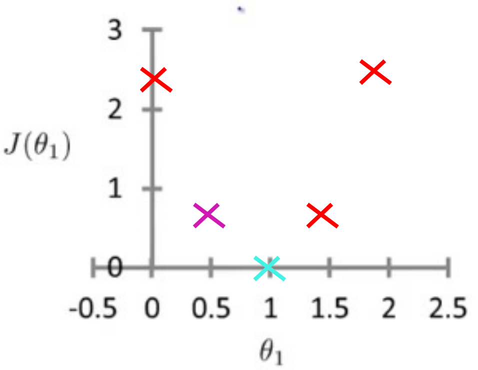
$h_\theta(x)$

(pour θ_1 fixe, c'est une fonction de x)

$J(\theta_1)$

(fonction du paramètre θ_1)





$h_{\theta 1}(x) = x/2 \quad (\theta_1 = 0.5)$

$J(0.5) = 0.58$

minimiser $J(\theta_1)$

θ_1

35

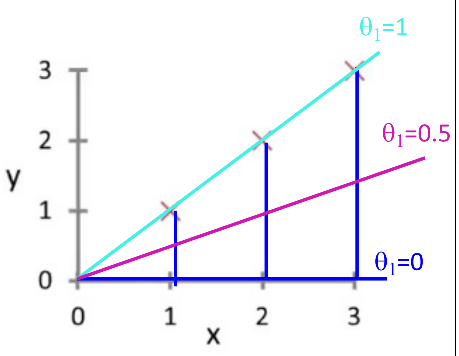
35

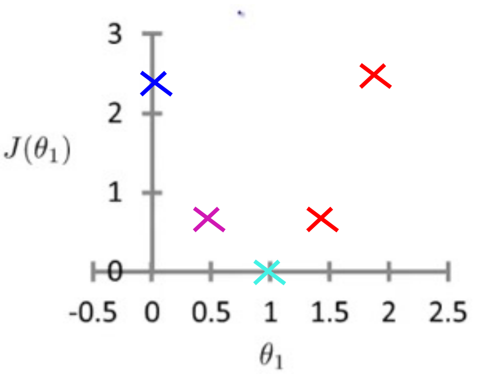
$h_\theta(x)$

(pour θ_1 fixe, c'est une fonction de x)

$J(\theta_1)$

(fonction du paramètre θ_1)





$h_{\theta 1}(x) = 0 \quad (\theta_1 = 0)$

$J(0) = 2.3$

minimiser $J(\theta_1)$

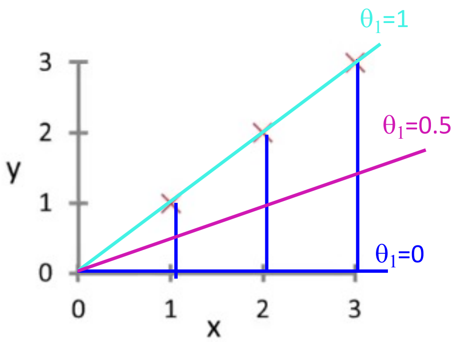
θ_1

36

36

$h_\theta(x)$

(pour θ_1 fixe, c'est une fonction de x)

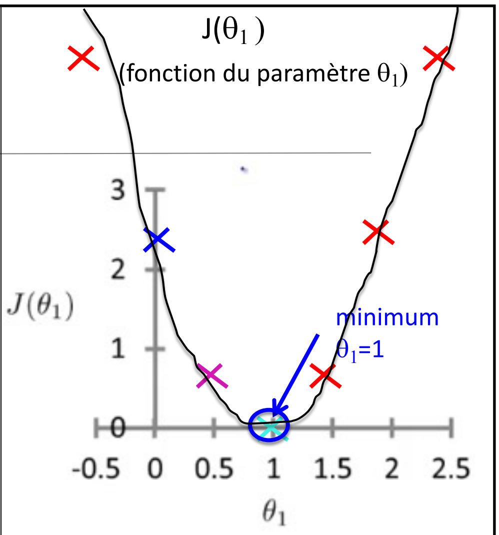


Minimum : J(1) = 0

$\theta_1 = 1$

$J(\theta_1)$

(fonction du paramètre θ_1)



minimiser $J(\theta_1)$

θ_1

37

37

Fonction de coût

Intuition II

deux paramètres θ_0, θ_1

38

38

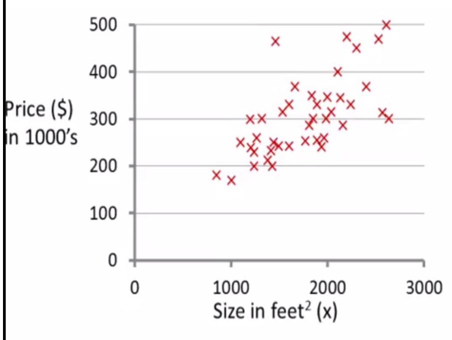
- <u>Hypothèse</u> : $h_\theta(x) = \theta_0 + \theta_1 x$
- <u>Paramètres</u> : θ_0, θ_1
- <u>Fonction coût</u> : $J(\theta_0, \theta_1) = \frac{1}{2m} \cdot \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$
- <u>But</u> : $\underset{\theta_0, \theta_1}{\text{minimiser}} \; J(\theta_0, \theta_1)$

39

39

$h_\theta(x)$

(pour θ_0 et θ_1 fixes, fonction de x)



$J(\theta_0, \theta_1)$

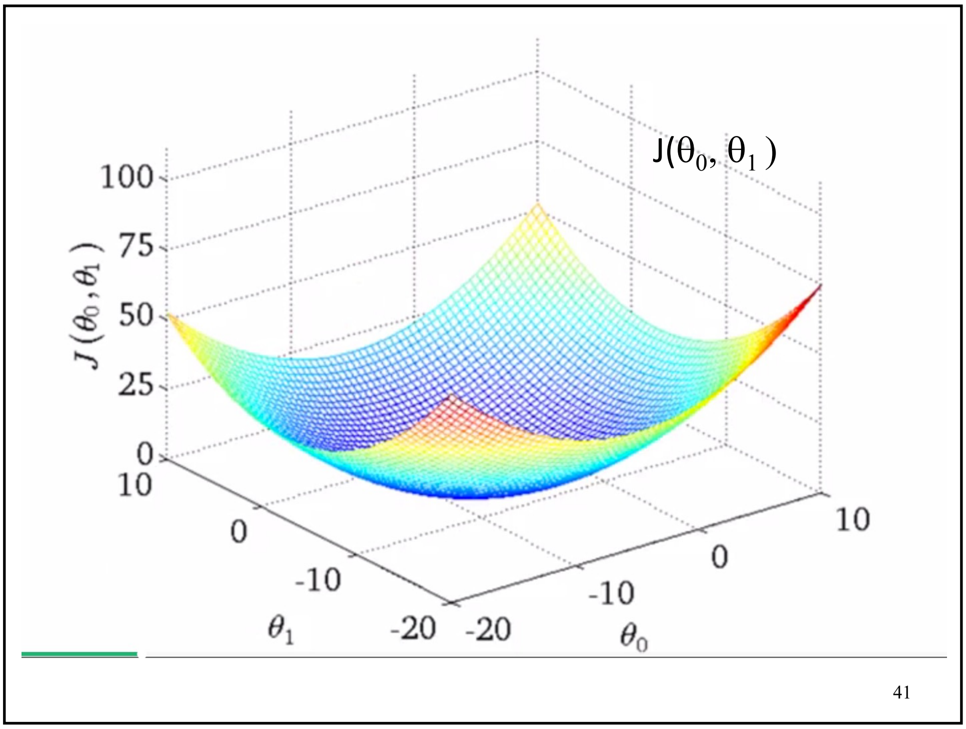
(fonction des paramètres θ_0, θ_1)

$J(\theta_0, \theta_1)$

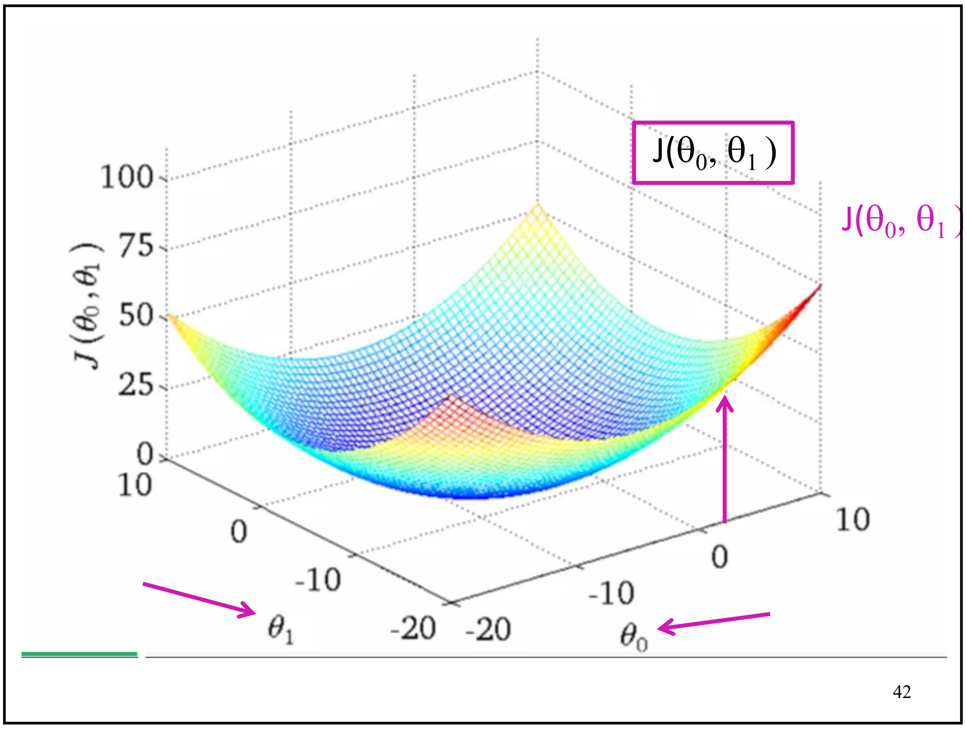
quelle forme ?

40

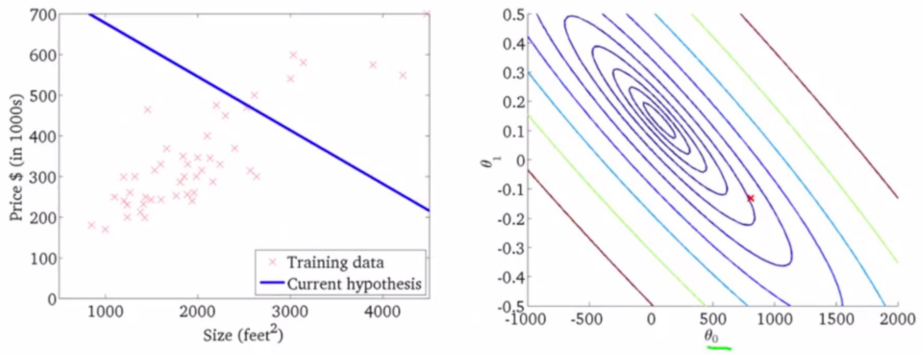
40


J(θ_0, θ_1)
$J(\theta_0, \theta_1)$
100
75
50
25
0
10
0
-10
-20
θ_1
-20
-10
0
10
θ_0
41

41


J(θ_0, θ_1)
J(θ_0, θ_1)
$J(\theta_0, \theta_1)$
100
75
50
25
0
10
0
-10
-20
θ_1
-20
-10
0
10
θ_0
42

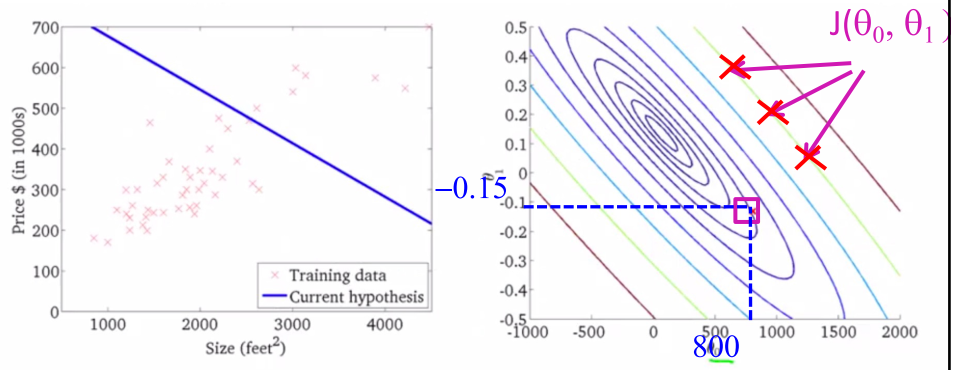
42



Contour figure

43

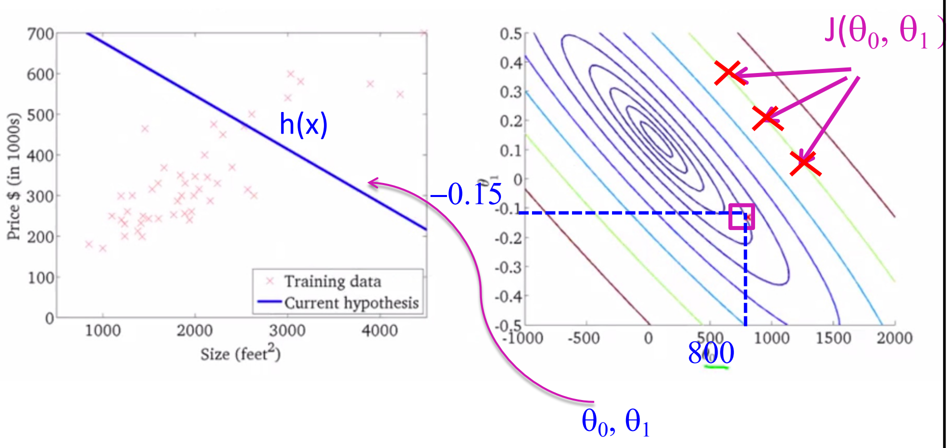
43



Contour figure

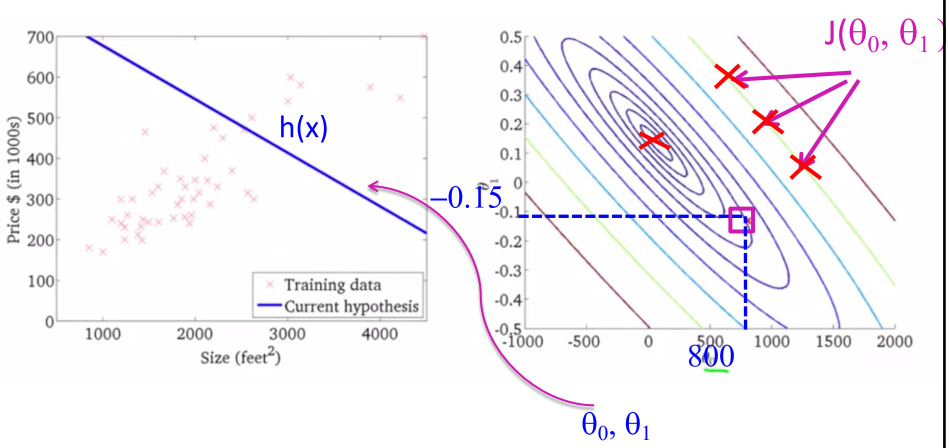
44

44


h(x)
Training data
Current hypothesis
Price $ (in 1000s)
Size (feet²)
J(θ0, θ1)
−0.15
800
θ0, θ1

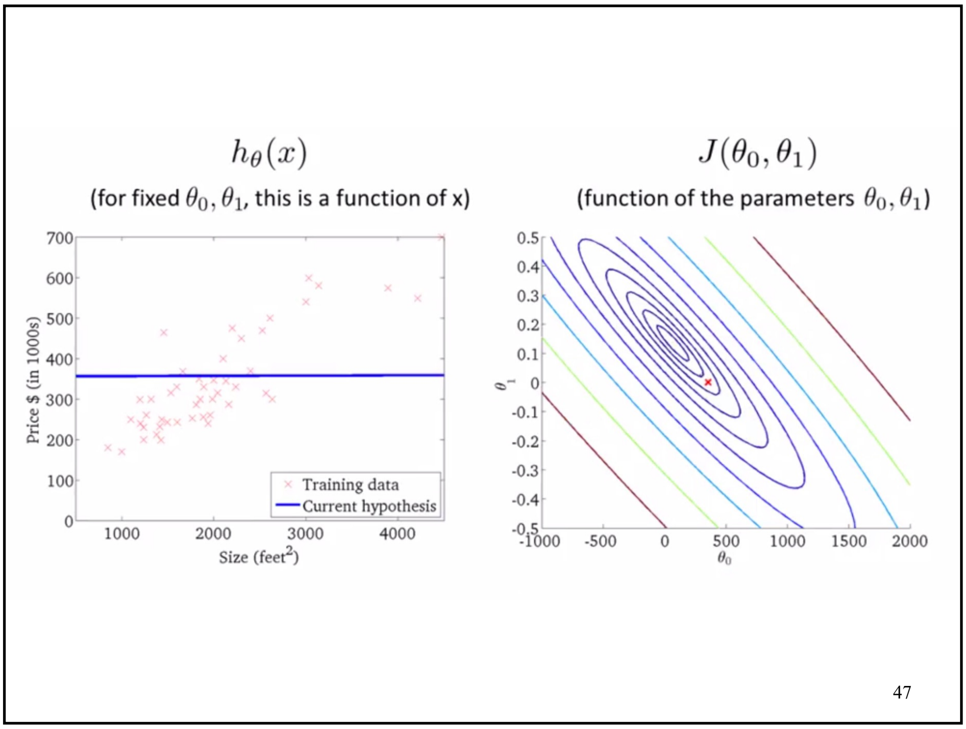
45

45

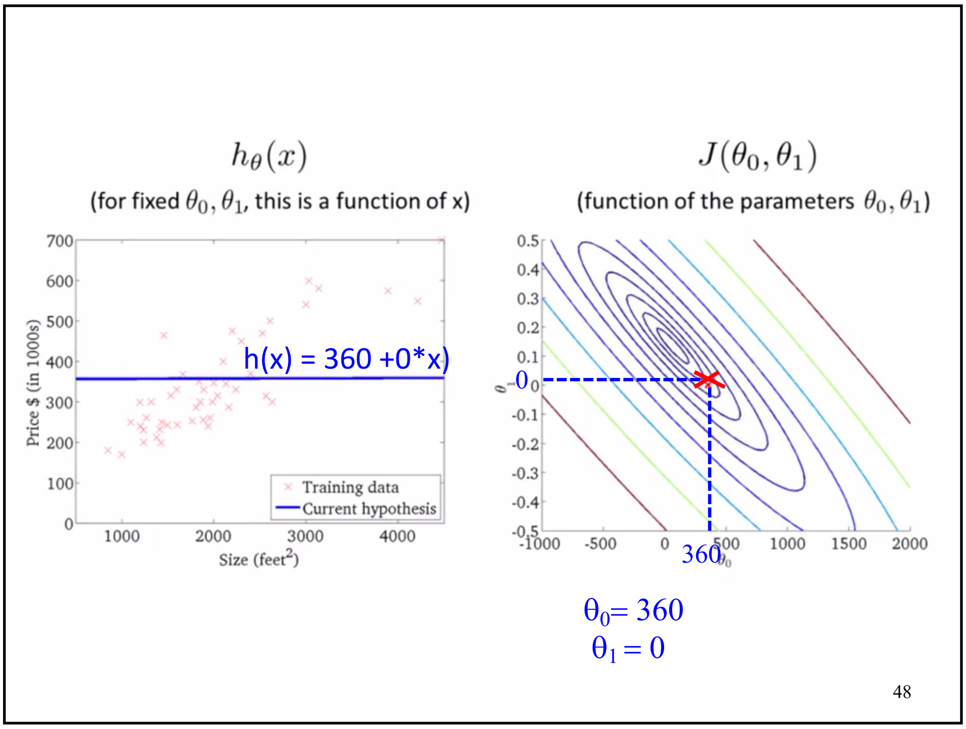

h(x)
Training data
Current hypothesis
Price $ (in 1000s)
Size (feet²)
J(θ0, θ1)
−0.15
800
θ0, θ1

46

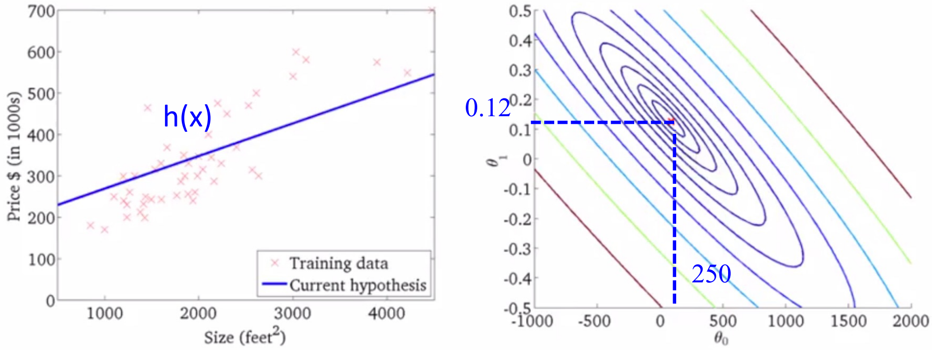
46


$h_\theta(x)$
(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)
Price $ (in 1000s)
Size (feet^2)
Training data
Current hypothesis
θ_1
θ_0
47

47


$h_\theta(x)$
(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)
h(x) = 360 +0*x)
Price $ (in 1000s)
Size (feet^2)
Training data
Current hypothesis
0
360
θ_0= 360
θ_1 = 0
48

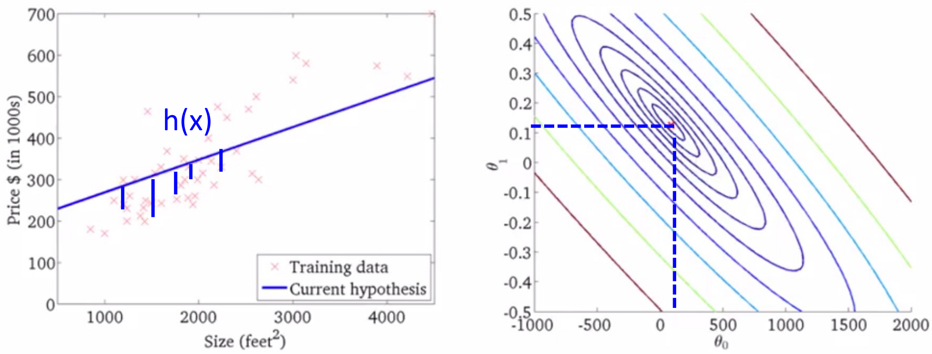
48



$\theta_0 = 250$
$\theta_1 = 0.12$

49

49

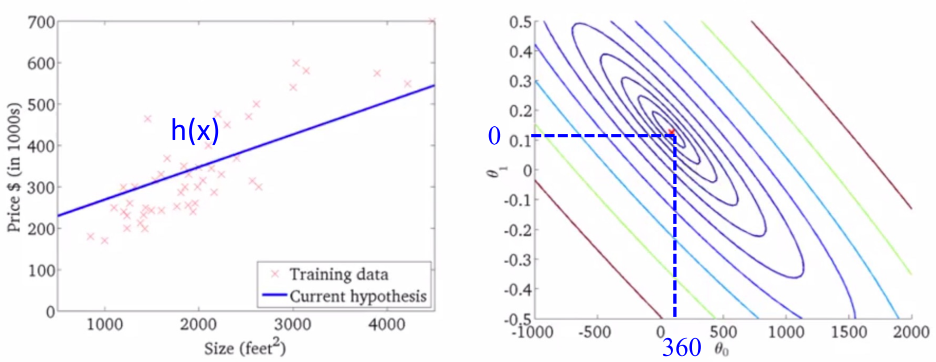


$h_\theta(x) = 250 + 0.13\ x$
Erreur quadratique

$\theta_0 = 250$
$\theta_1 = 0.13$

50

50



- <u>Hypothèse</u> : $h_\theta(x) = \theta_0 + \theta_1 x$
- Nous ne voulons pas trouver manuellement les paramètres θ_0 , θ_1 ! L'idée est d'utiliser un algorithme pour trouver automatiquement les paramètres qui minimisent la fonction coût $J(\theta_0 , \theta_1)$ par rapport à θ_0 et θ_1

51

51

Régression linéaire
Descente de gradient

52

52

- Fonction coût : $J(\theta_0, \theta_1)$ $J(\theta_0, \theta_1,\theta_2,\ldots \theta_n)$
- But : minimiser $J(\theta_0, \theta_1)$ minimiser $J(\theta_0, \theta_1,\theta_2,\ldots \theta_n)$
  θ_0, θ_1 $\theta_0, \theta_1,\theta_2,\ldots \theta_n$

53

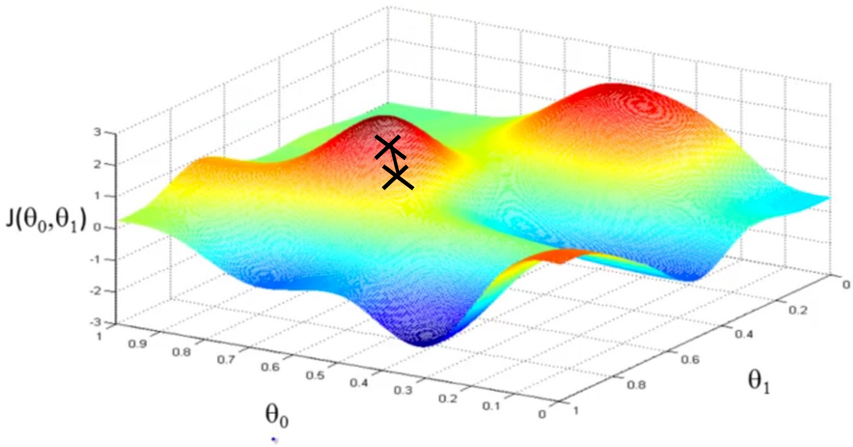
53

- Fonction coût : $J(\theta_0, \theta_1)$ $J(\theta_0, \theta_1,\theta_2,\ldots \theta_n)$
- But : minimiser $J(\theta_0, \theta_1)$ minimiser $J(\theta_0, \theta_1,\theta_2,\ldots \theta_n)$
  θ_0, θ_1 $\theta_0, \theta_1,\theta_2,\ldots \theta_n$

- Algorithme :
  - Commencer avec θ_0, θ_1 (initial guesses, e.g., $\theta_0 = 0$, $\theta_1=0$)
  - Changer θ_0, θ_1 pour réduire $J(\theta_0, \theta_1)$
    jusqu'à atteindre un minimum

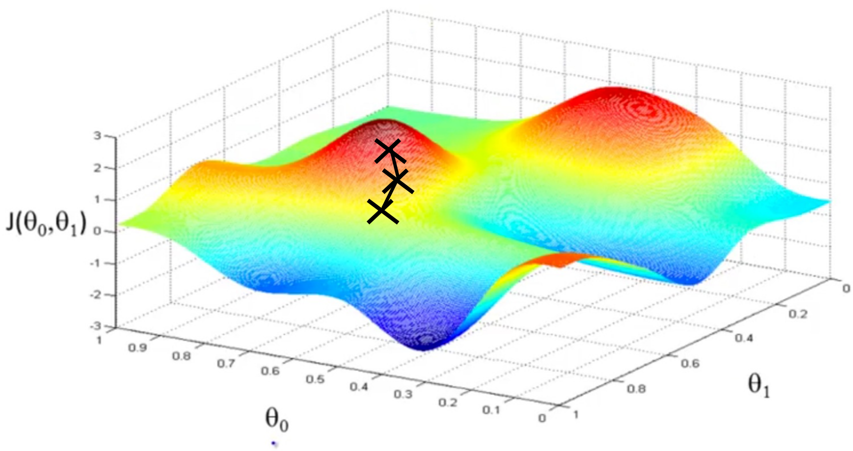
54

54


J(θ0,θ1)
θ0
θ1
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
0
0.2
0.4
0.6
0.8
1

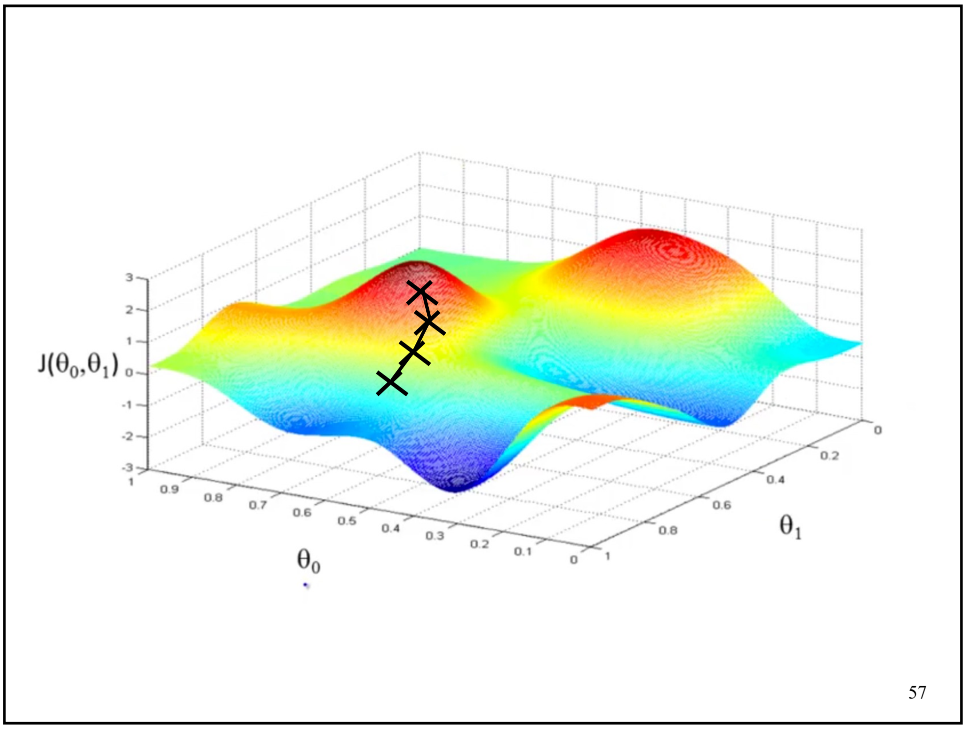
55

55

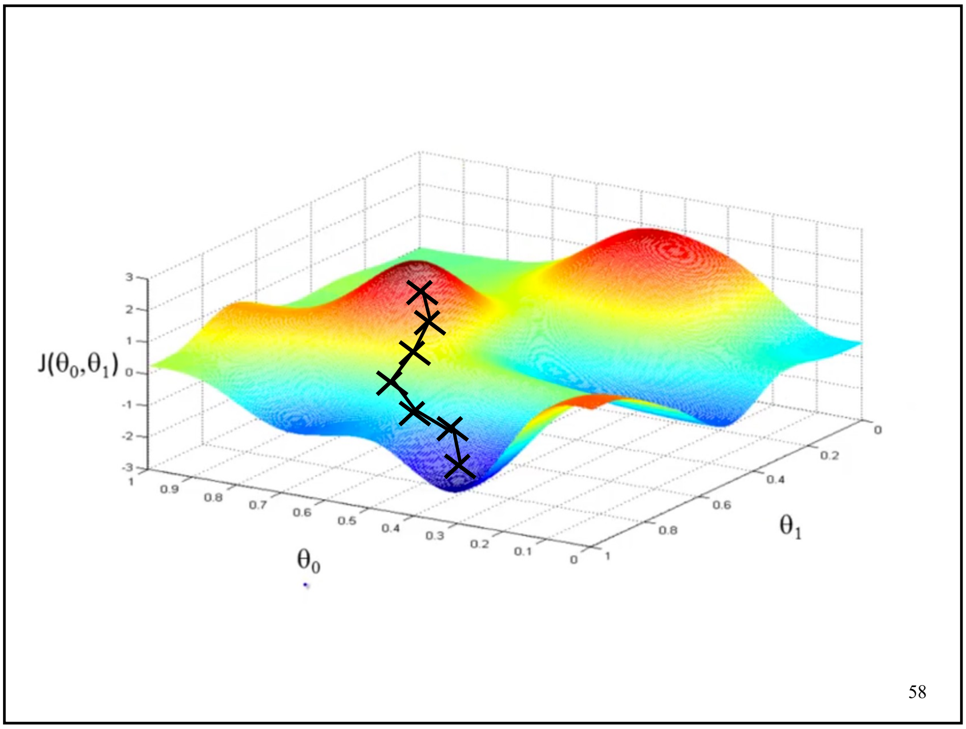

J(θ0,θ1)
θ0
θ1
3
2
1
0
-1
-2
-3
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8
1

56

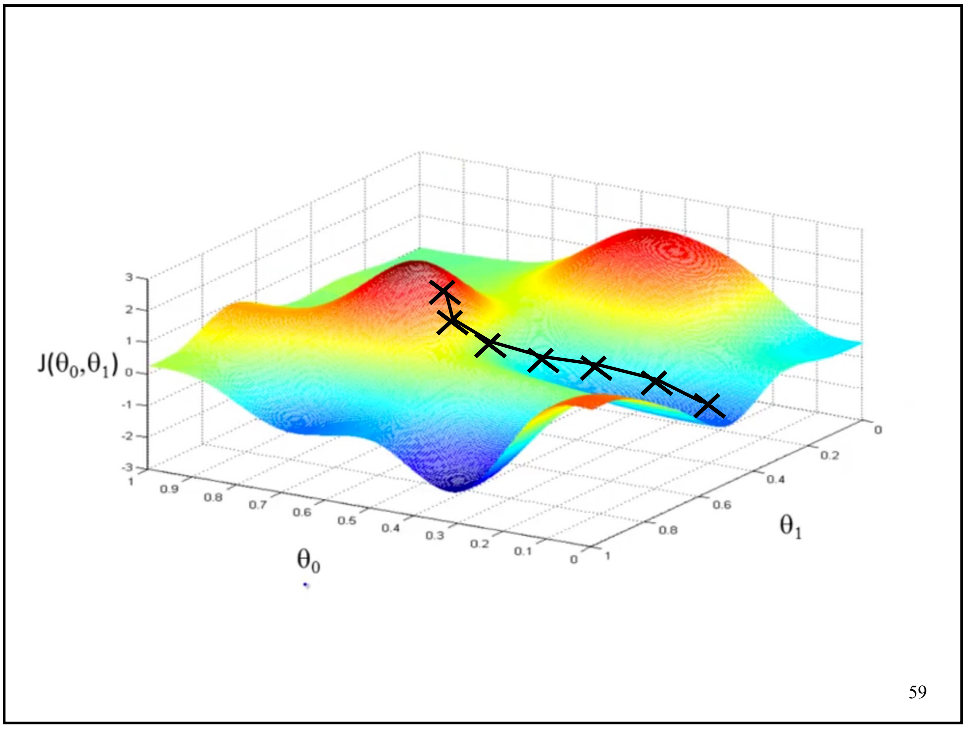
56



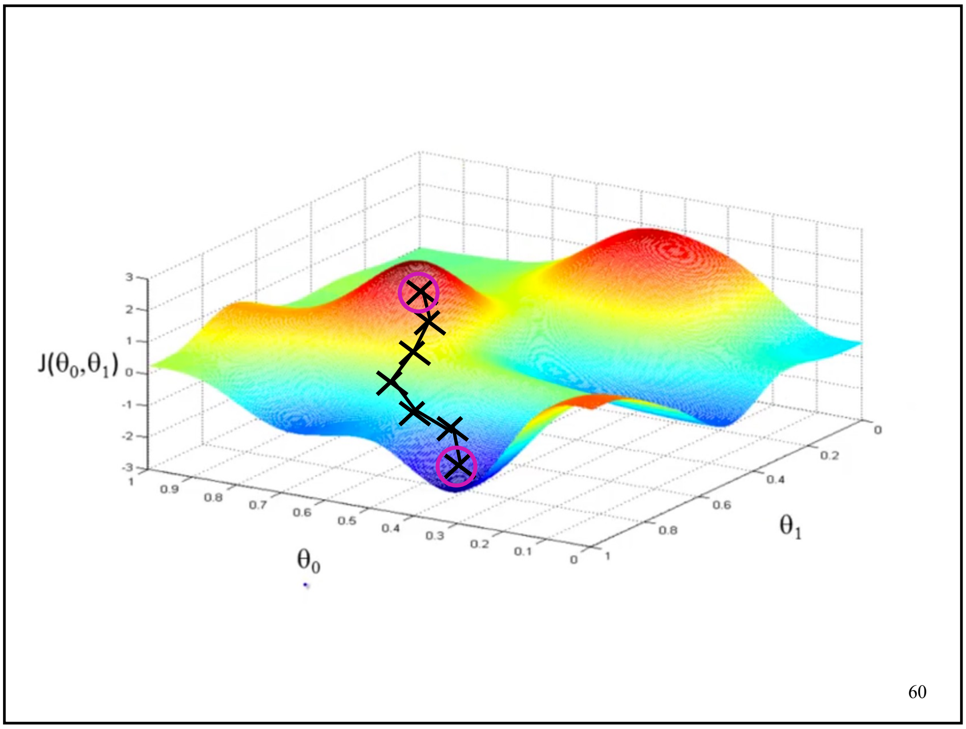
57



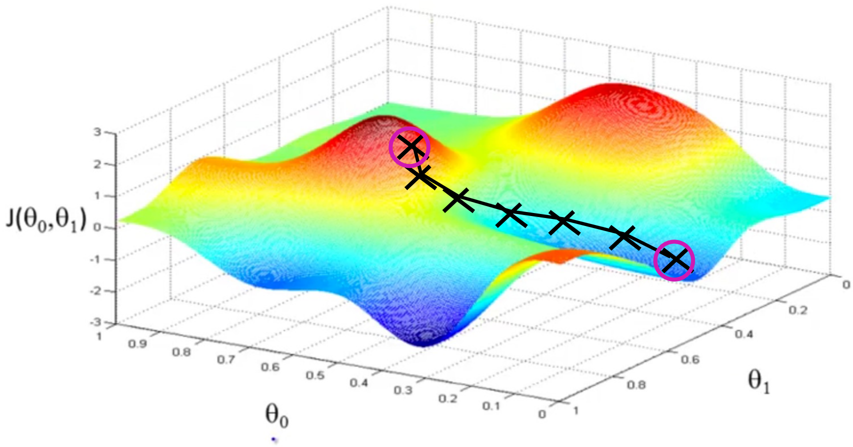
58


$J(\theta_0,\theta_1)$
θ_0
θ_1
59

59


$J(\theta_0,\theta_1)$
θ_0
θ_1
60

60



61

61

Algorithme de descente de gradient

- Algorithme :

Répéter jusqu'à convergence :

Repeat {

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

until θ_0, θ_1 converge

}

Mises à jour simultanées de θ_0, θ_1

62

62

Gradient Descent Algorithm

- <u>Algorithm</u>:

Répéter jusqu'à convergence :

Repeat {

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

}

Paramètre α : taux d'apprentissage

Contrôle la façon dont on prend la descente de plus grande pente (plus ou moins vite)

Mises à jour simultanées

$temp_0 = \theta_0 - \alpha$. dérivée$(\theta_0, \theta_1)$

$temp_1 = \theta_1 - \alpha$. dérivée$(\theta_0, \theta_1)$

$\theta_0 = temp_0$

$\theta_1 = temp_1$

Implémentation fausse

$temp_0 = \theta_0 - \alpha$. dérivée$(\theta_0, \theta_1)$

$\theta_0 = temp_0$

$temp_1 = \theta_1 - \alpha$. dérivée$(\theta_0, \theta_1)$

$\theta_1 = temp_1$

63

63

Exercice : Supposez que θ_0=1, θ_1=2, et qu'on met à jour simultanément θ_0 et θ_1 en utilisant la règle : $\theta_j=\theta_j$ + rac$(\theta_0\theta_1)$.

Quelles sont les valeurs résultantes de θ_0 et θ_1?

- θ_0 = 1 and θ_1=2
- θ_0 = 1 + rac(2) and θ_1=2 + rac(2)
- θ_0 = 2 + rac(2) and θ_1=1 + rac(2)
- θ_0 = 1 + rac(2) and θ_1=2 + rac((1 + rac(2)).2))

64

64

Descente du gradient
Intuition

65

65

Algorithme de descente du gradient

- Algorithm:

  Répéter jusqu'à convergence

Repeat {

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

}

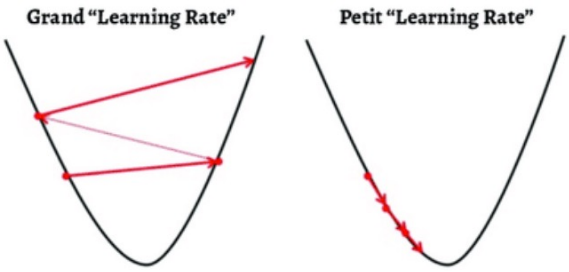
Mise à jour simultanée de θ_0 et θ_1

Supposons que nous ayons uniquement θ_1

66

66

Algorithme de descente du gradient



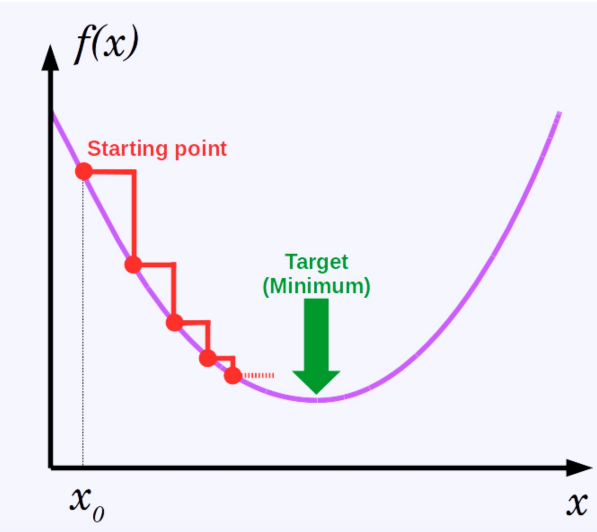
- Si α est trop grand, la descente de grandient peut "overshooter" le minimum (osciller de part et d'autre). l'algoritme peut ne pas converger, ou même diverger.
- Si α est trop petit, la descente de gradient peut être (trop) lente

67

67

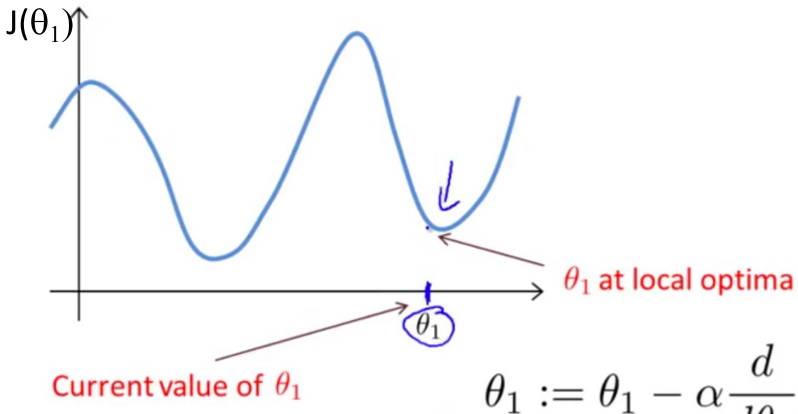
Algorithme de descente du gradient

- La descente de gradient peut converger vers un minimum local, avec un taux α fixé.
- Lorsqu'on approche le minimum local, la descente de gradient va prendre automatiquement de plus petits incréments : ainsi, pas besoin de diminuer α au cours du temps.



68

Algorithme de descente du gradient



$$\theta_1 := \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta_1)$$

69

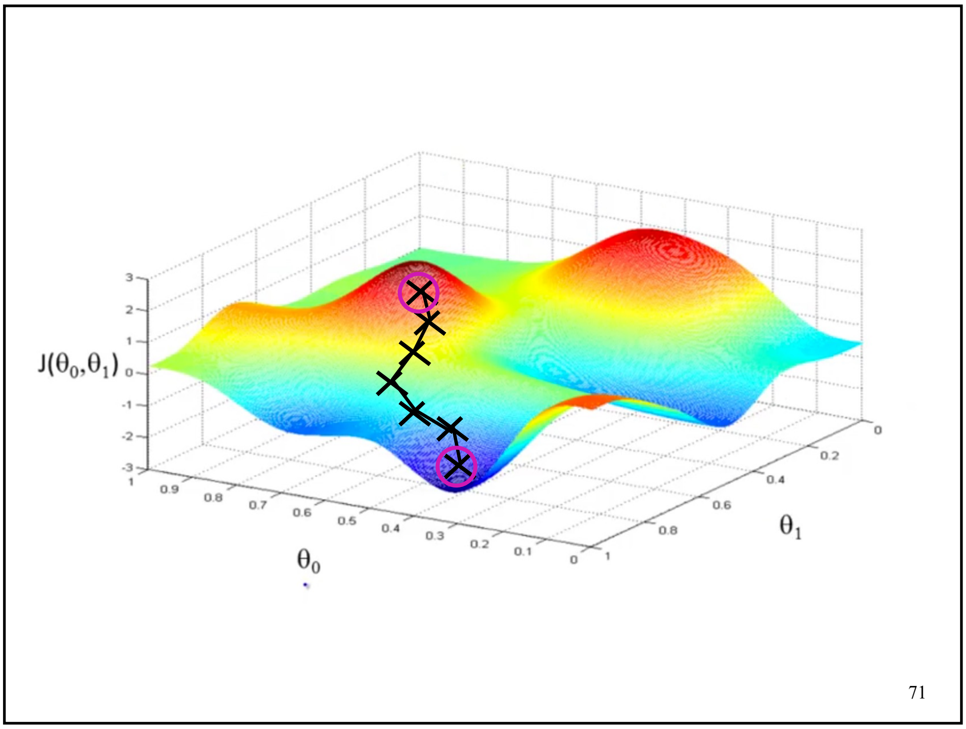
69

Université Bretagne Sud

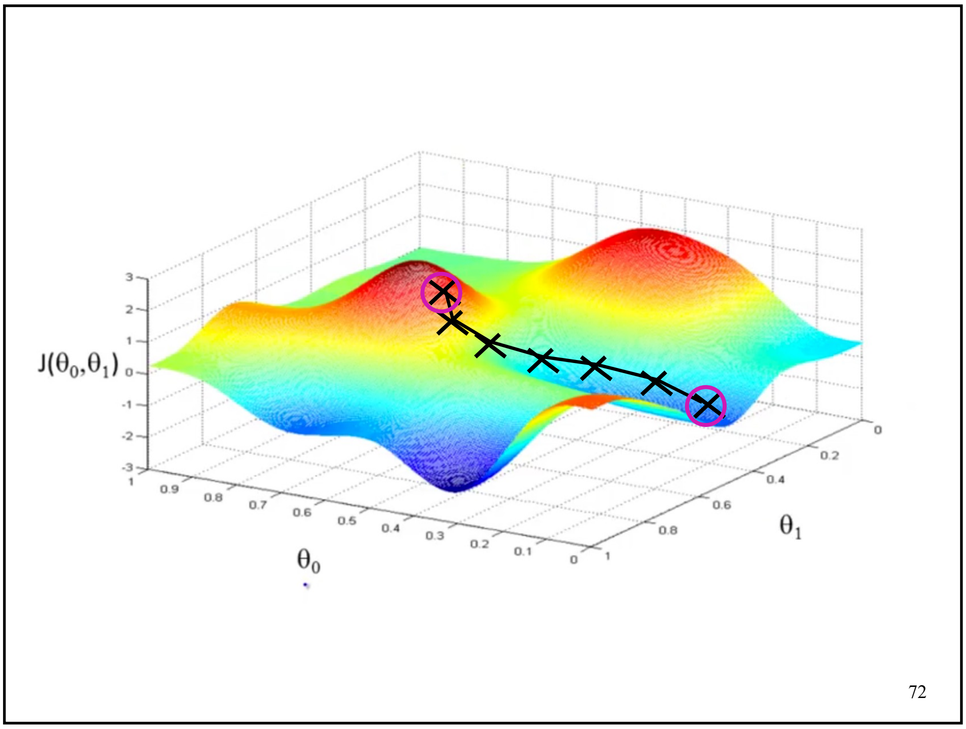
Descente du gradient pour la régression linéaire

70

70

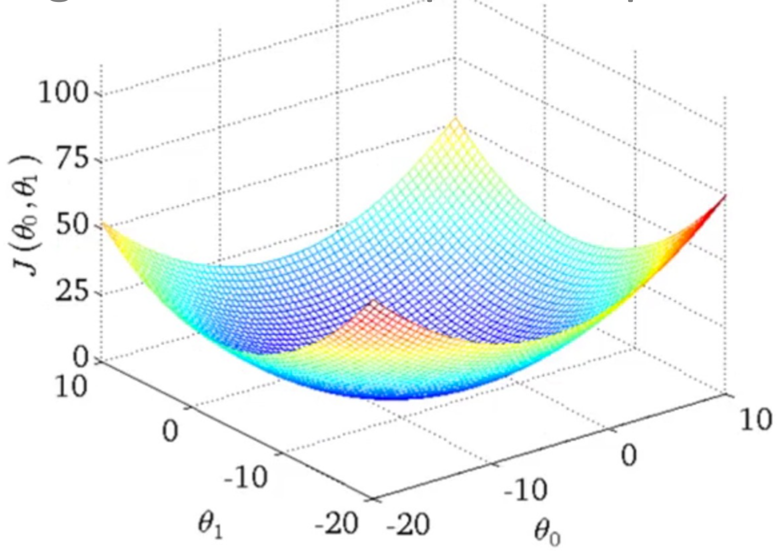


71



72

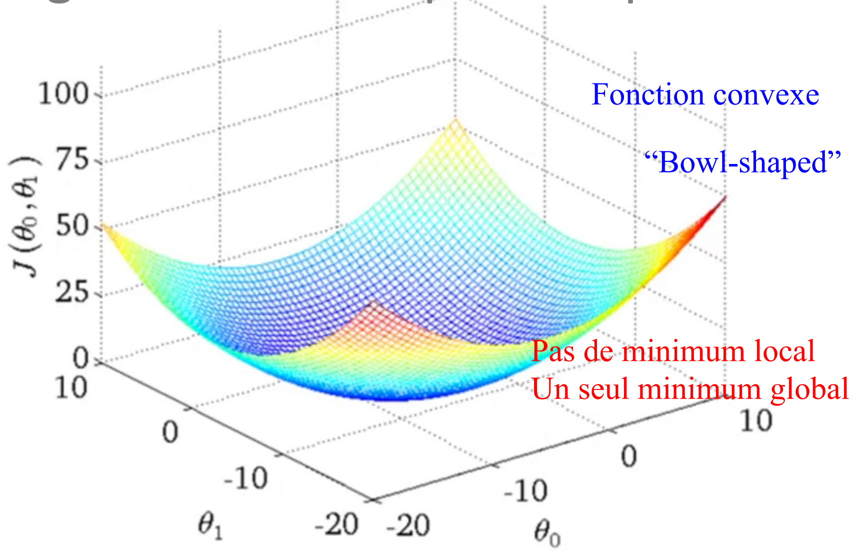
Régression : coût quadratique



73

73

Régression : coût quadratique



74

74

Régression : coût quadratique

- Algorithme de descente de gradient : peut tomber dans un minimum local, dépendant du point de départ de l'algorihtme (initialisation des paramètres θ_0 et θ_1).
- Régression linéaire : la descente de gradient appliquée à une fonction coût quadratique :
  - La fonction coût est toujours une fonction convexe (bowl-shaped)
  - Avec une telle méthode, impossible de tomber dans un minimum local, il n'y a qu'un seul minimum global.

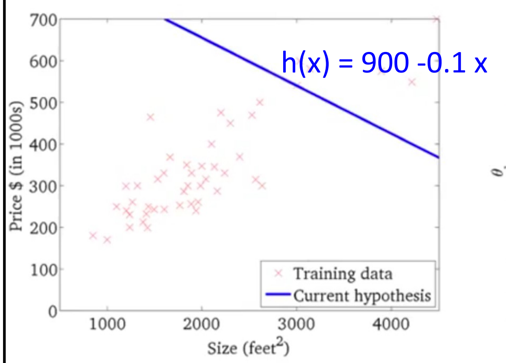
75

75

Algorithme en action

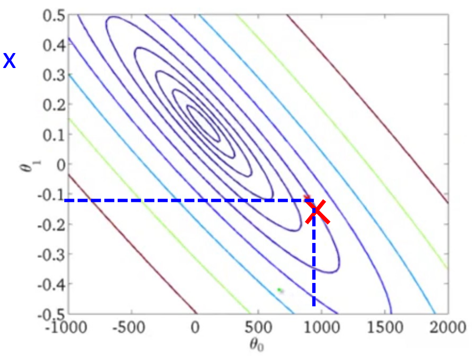
$h_\theta(x)$

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



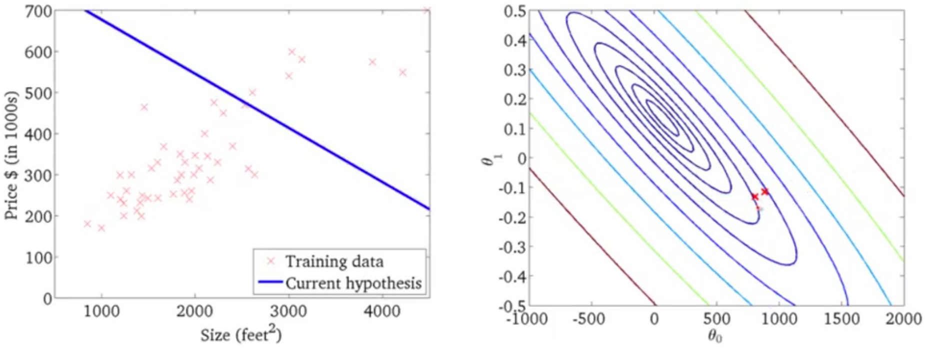
$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



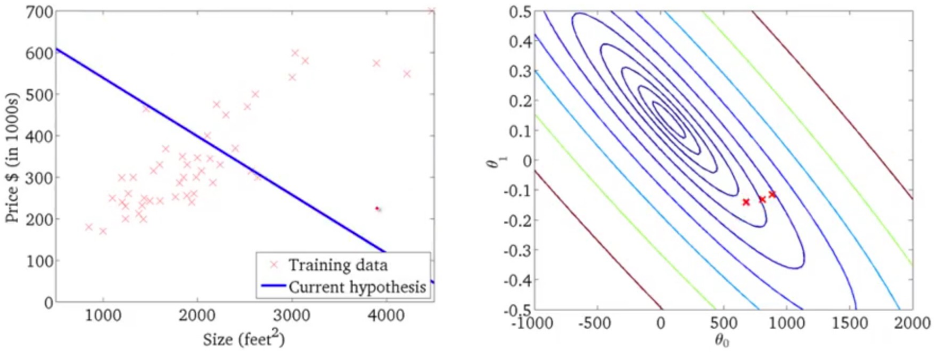
76

76


700
600
500
400
300
200
100
0
Price $ (in 1000s)
1000
2000
3000
4000
Size (feet^2)
Training data
Current hypothesis
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
θ_1
-1000
-500
0
500
1000
1500
2000
θ_0

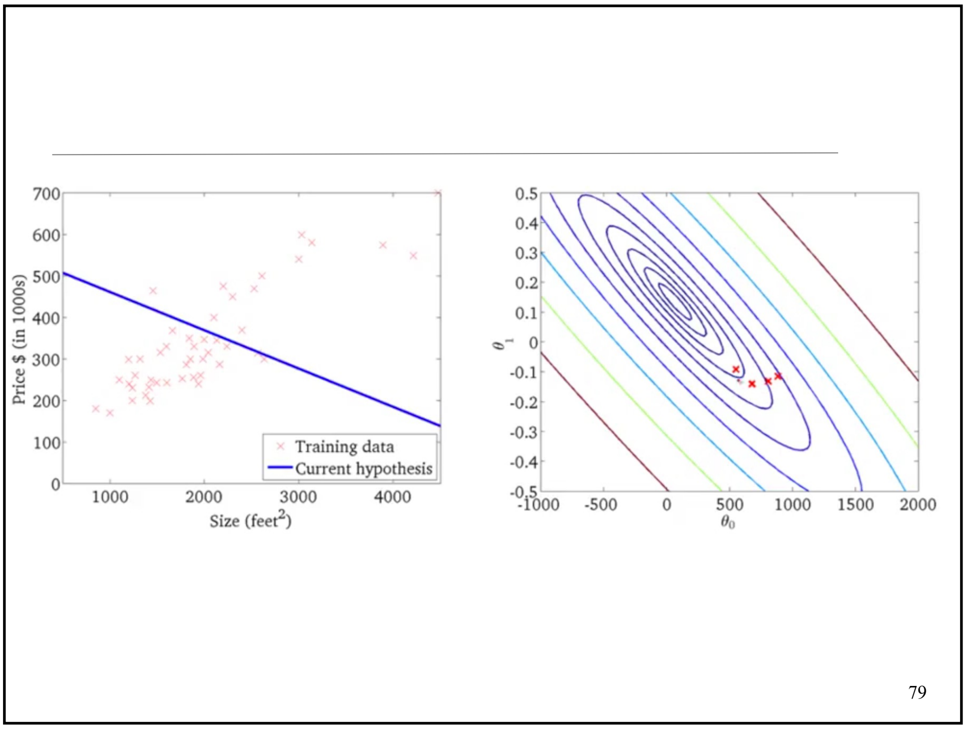
77

77

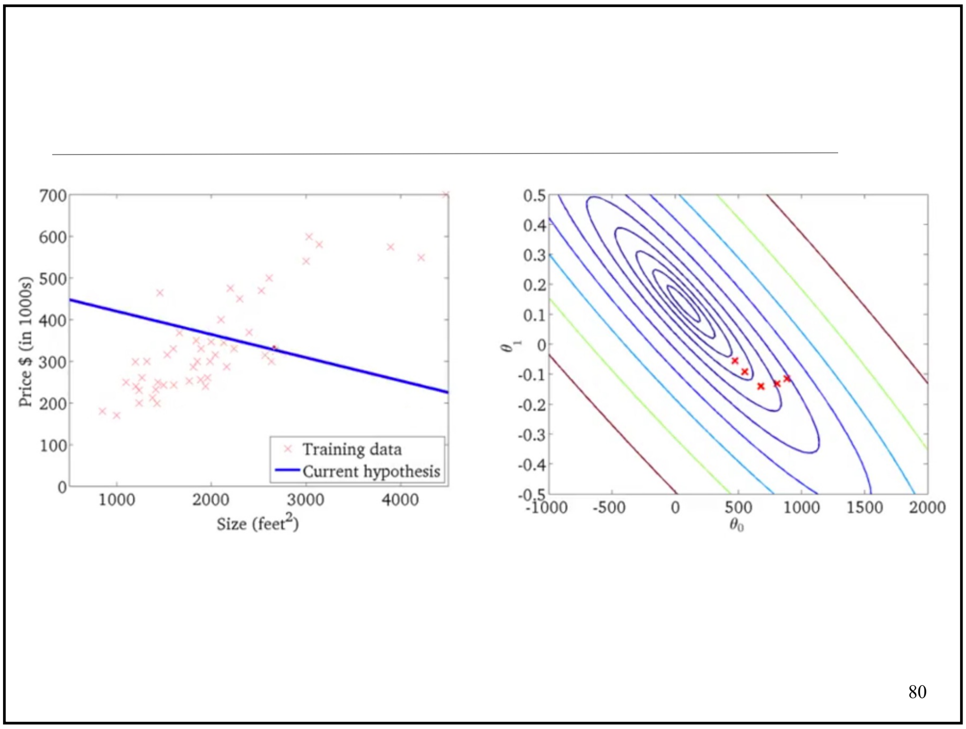

700
600
500
400
300
200
100
0
Price $ (in 1000s)
1000
2000
3000
4000
Size (feet^2)
Training data
Current hypothesis
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
θ_1
-1000
-500
0
500
1000
1500
2000
θ_0

78

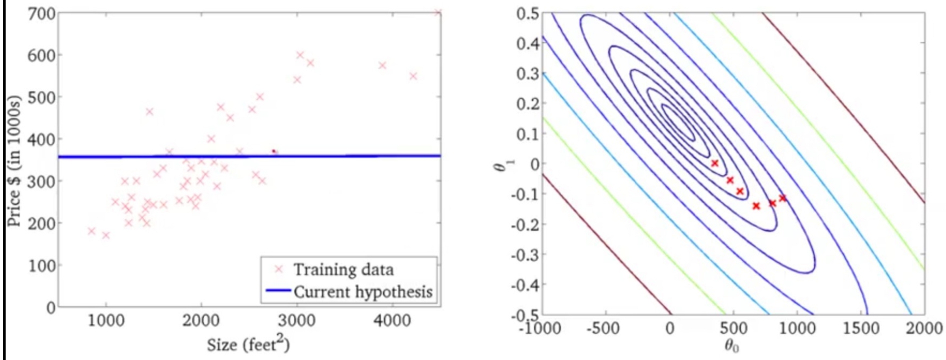
78


700
600
500
400
300
200
100
0
Price $ (in 1000s)
1000
2000
3000
4000
Size (feet^2)
Training data
Current hypothesis
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
θ_1
-1000
-500
0
500
1000
1500
2000
θ_0
79

79

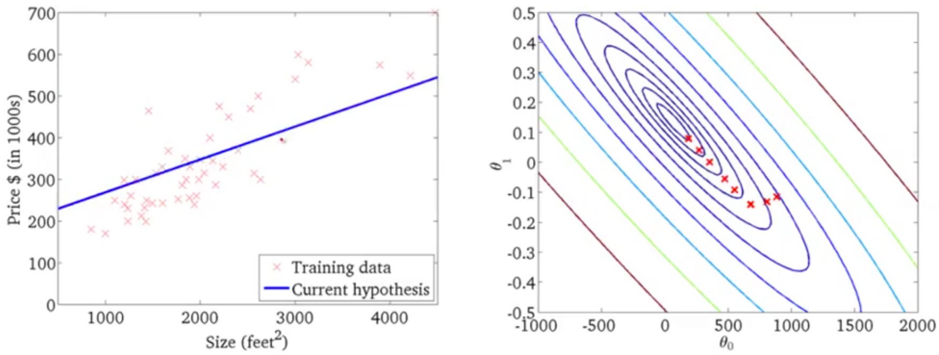

700
600
500
400
300
200
100
0
Price $ (in 1000s)
1000
2000
3000
4000
Size (feet^2)
Training data
Current hypothesis
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
θ_1
-1000
-500
0
500
1000
1500
2000
θ_0
80

80


700
600
500
400
300
200
100
0
Price $ (in 1000s)
1000
2000
3000
4000
Size (feet²)
Training data
Current hypothesis
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
θ1
-1000
-500
0
500
1000
1500
2000
θ0

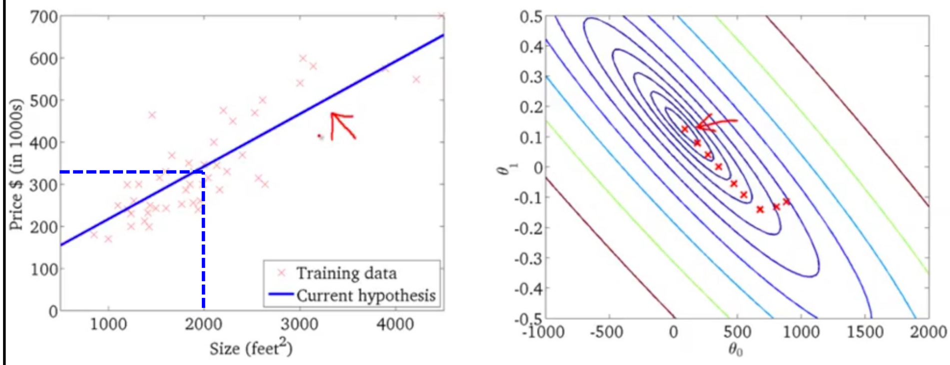
81

81


700
600
500
400
300
200
100
0
Price $ (in 1000s)
1000
2000
3000
4000
Size (feet²)
Training data
Current hypothesis
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
θ1
-1000
-500
0
500
1000
1500
2000
θ0

82

82



83

83

Parmi les propositions suivantes, lesquelles sont vraie ?
Sélectionnez toutes les bonnes réponses

- □ Pour que la descente de gradient converge, il faut que le taux α décroisse au cours du temps.
- □ La descente de gradient trouve toujours le minimum global de toute fonction coût $J(\theta_0, \theta_1)$.
- □ La descente de gradient peut converger même si α est maintenu fixe (mais α ne peut pas être trop grand, sinon l'algorithme peut ne pas converger).
- □ Pour le choix spécifique de fonction coût utilisé dans la régression linéaire, il n'y a pas d'optimum local (autre que l'optimum global).

84

84

- Algorithm de descente du gradient

Repeat {

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

(for j = 1 and j = 0)

until convergence

}

- Linear Regression Model

$h_\theta(x) = \theta_0 + \theta_1 x$

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

$$= \frac{1}{2m} \cdot \sum_{i=1}^{m} (\theta_0 + \theta_1 x^{(i)} - y^{(i)})^2$$

85

85

Algorithme de descente du gradient

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

$$= \frac{\partial}{\partial \theta_j} \cdot \frac{1}{2m} \cdot \sum_{i=1}^{m} (\theta_0 + \theta_1 x^{(i)} - y^{(i)})^2$$

$$j = 0 : \frac{\partial}{\partial \theta_0} J(\theta_0\ ,\ \theta_1) = \frac{1}{m} \cdot \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})$$

$$j = 1 : \frac{\partial}{\partial \theta_1} J(\theta_0\ ,\ \theta_1) = \frac{1}{m} \cdot \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}).x^{(i)}$$

86

86

Algorithme de descente du gradient

Repeat until convergence {

$$j = 0 : \theta_0 = \theta_0 - \frac{1}{m} \cdot \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})$$

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

} ➔ update θ_0 and θ_1 simultaneously

87

87