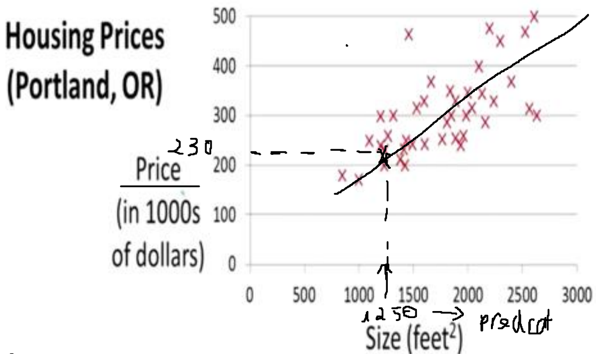
One variable

* Model representation

Example



Supervised Learning | Regression problem
given "right answer" | Predict real-valued output

training set

Size of feet^2 (x)	price (y)
2104	460
1416	232
1536	315
852	178
...	...

m = number of tranning example
x's = Input features
y's = Output features
$x^{(i)}, y^{(i)}$ - tranning example i th

How ML work

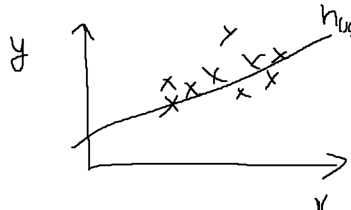
$\boxed{\text{Training set}}$
↓
$\boxed{\text{learning algorithm}}$
↓
size of house (x) → $\boxed{h}$ → estimated price (y)

hypothesis (giả thuyết)

h map $x \to y$
$h(x) = y$

represent h

$h_\theta(x) = \theta_0 + \theta_1 x$ ($f(x) = ax + b$)



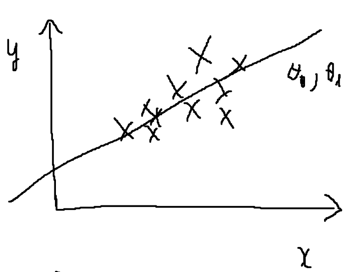
linear

linear regression with one variable
univariable linear regression

* Cost function

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

Parameter
(tham số)



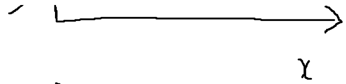
$$\underset{\theta_0, \theta_1}{\text{minimize}} \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$$

Kết quả dự đoán

Thực tế

$$\underbrace{J(\theta_0, \theta_1)}_{\text{cost function}} = \frac{1}{2m} \sum_{i=1}^{m} \left(h_{(\theta)}(x^{(i)}) - y^{(i)} \right)^2$$

Square error function



ý tưởng . tìm tham số θ_0, θ_1 cho $h_\theta(x)$ gần với y của các mẫu huấn luyện (x, y)

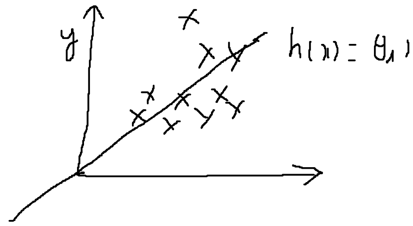
Square error function

Simplified

$h_\theta(x) = \theta_1 x$ $\qquad \theta_0 = 0$

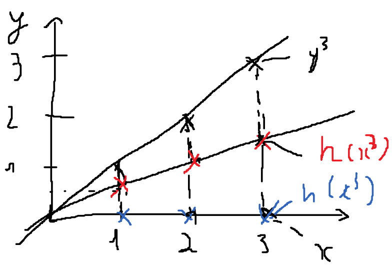
$$J(\theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^i) - y^i \right)^2$$

$\underset{\theta_1}{\text{minimize}} \; J(\theta_1)$



$h_\theta(x)$

cố định θ_1, x tham số



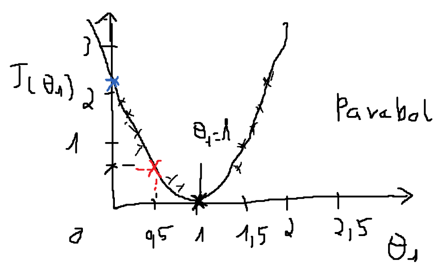
Với $\theta_1 = 1$

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

$$= \frac{1}{6} \cdot \left((1-1)^2 + (2-2)^2 + (3-3)^2 \right) = 0$$

$J(\theta_1)$

θ_1 là tham số



$J(1) = 0$

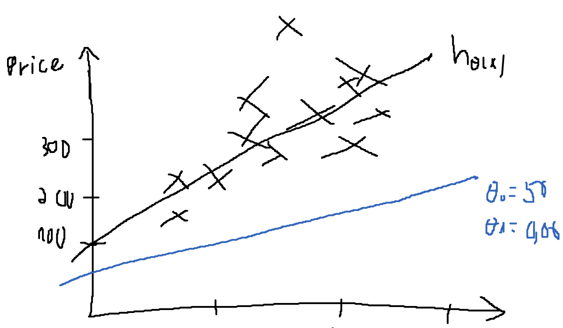
$J(0{,}5) = 0{,}58$

$J(0) = 2{,}167$

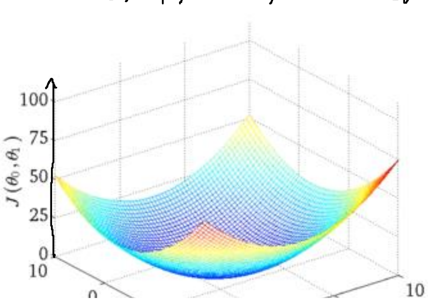
$\underset{\theta_1}{\text{minimize}} \; J(\theta_1)$

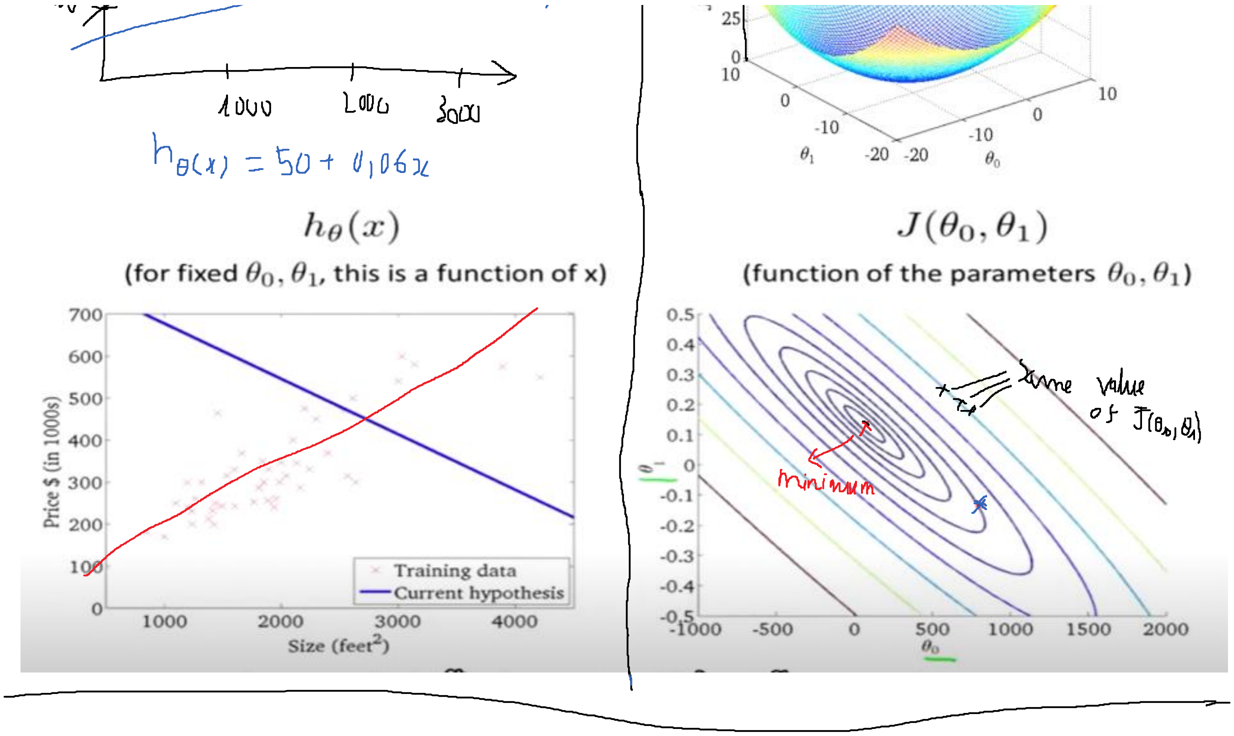
for $h_\theta(x) = \theta_0 + \theta_1 x$

$h_\theta(x)$ x tham số

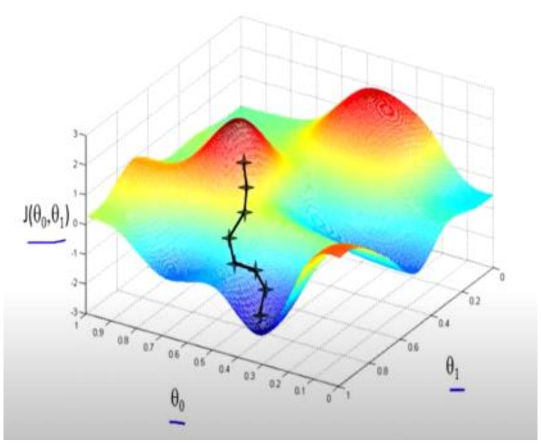


$J(\theta_0, \theta_1)$. θ_0, θ_1 tham số

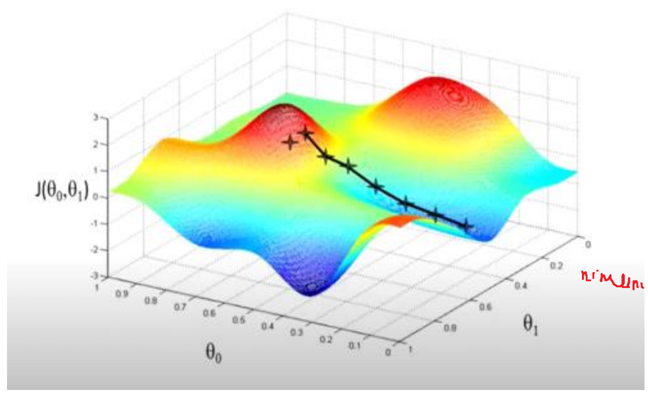




1000
2000
3000
$h_{\theta(x)} = 50 + 0,06x$
$h_\theta(x)$
(for fixed θ_0, θ_1, this is a function of x)
Price $ (in 1000s)
Size (feet^2)
Training data
Current hypothesis
$J(\theta_0, \theta_1)$
(function of the parameters θ_0, θ_1)
same value
of $J(\theta_0, \theta_1)$
minimum
θ_0
θ_1



$J(\theta_0,\theta_1)$
θ_0
θ_1



$J(\theta_0,\theta_1)$
θ_0
θ_1

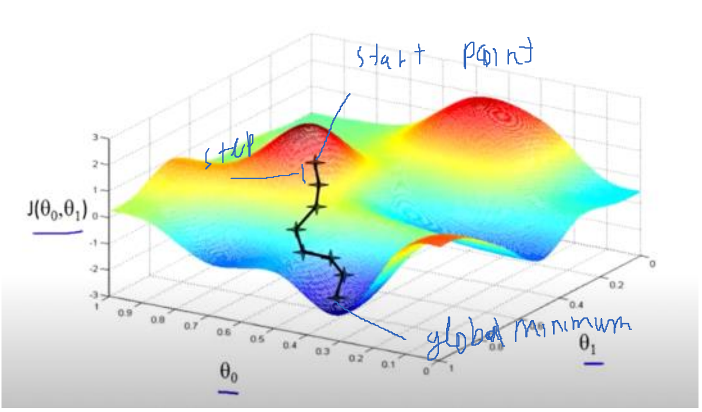
Gradient descent for univariable

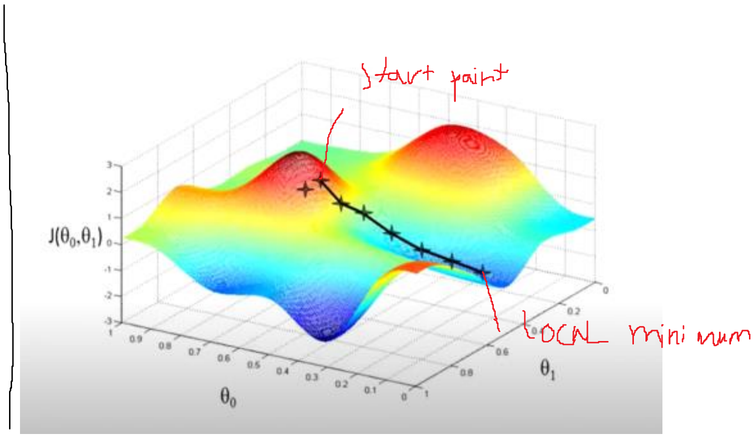
* Gradient descent

ý tưởng : có hàm $J(\theta_0, \theta_1)$, muốn $\min_{\theta_0, \theta_1} J(\theta_0, \theta_1)$

— khởi đầu với θ_0, θ_1 bất kì

— thay đổi θ_0, θ_1 để giảm $J(\theta_0, \theta_1)$ tới khi đạt được minimum





> phụ thuộc vào startpoint (tính chất của GD)

thuật toán

$$\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \quad \text{for } j=0 \text{ and } j=1$$

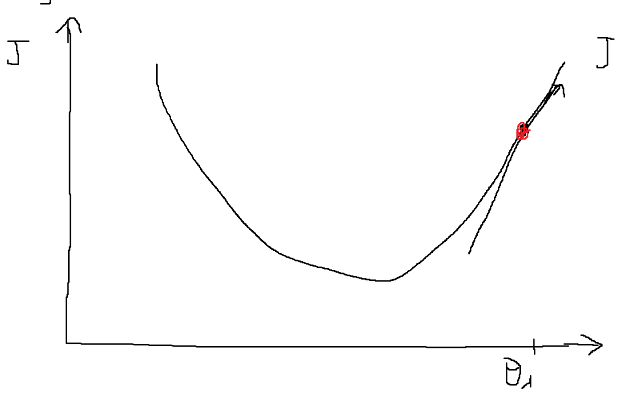
learning rate

- lặp lại cho tới khi θ_j không đổi $\Leftrightarrow \boxed{\frac{\partial J(\theta_0, \theta_1)}{\partial \theta_j} = 0} \Leftrightarrow$ cực tiểu
- cập nhật đồng thời các tham số

Simplifiend

$\min_{\theta_1} J(\theta_1) \Leftrightarrow \theta_0 = 0$

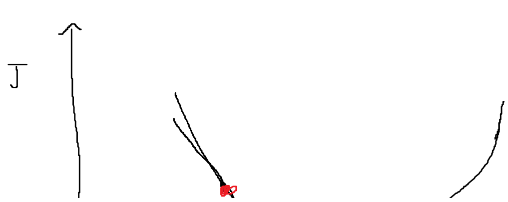
for derivate



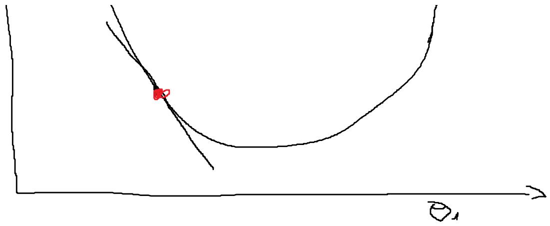
$J(\theta_1) \quad \theta_1 \in \mathbb{R}$

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

$\Leftrightarrow \theta_1 - \alpha$ (số dương) $\Rightarrow \theta_1$ giảm



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

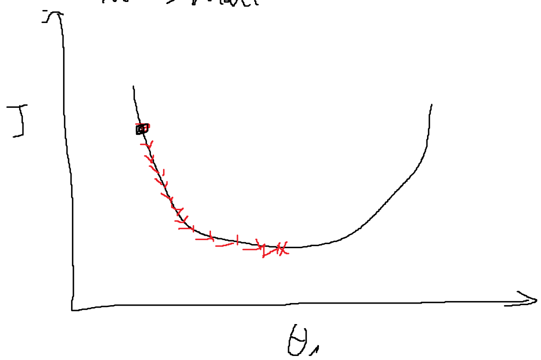


$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$

$\Leftrightarrow \theta_1 - \alpha . (\text{số âm})$

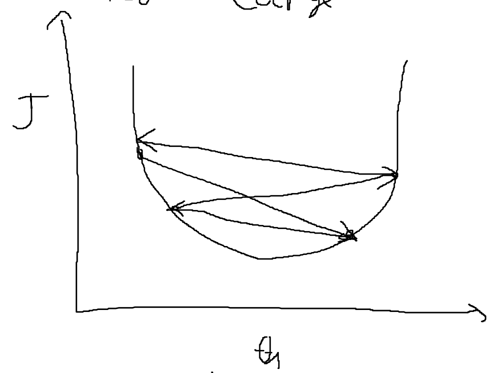
$\Rightarrow \theta_1$ tăng

for α too small



can be slow

too large



may fail

Khi gần điểm cực tiểu thì các bước càng nhỏ nên không cần cập nhật lại α

Gradient descent for Linear regression

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

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

$$\theta_0 \quad \frac{\partial}{\partial \theta_1} = \frac{1}{m} \sum_{i=1}^{m} \left(\theta_0 + \theta_1 x - y^i \right)$$

$$\theta_1 \quad \frac{\partial}{\partial \theta_1} J = \frac{1}{\cancel{2}m} \sum_{i}^{m} \cancel{2} . x^i \left(\theta_0 + \theta_1 x^i - y^i \right)$$

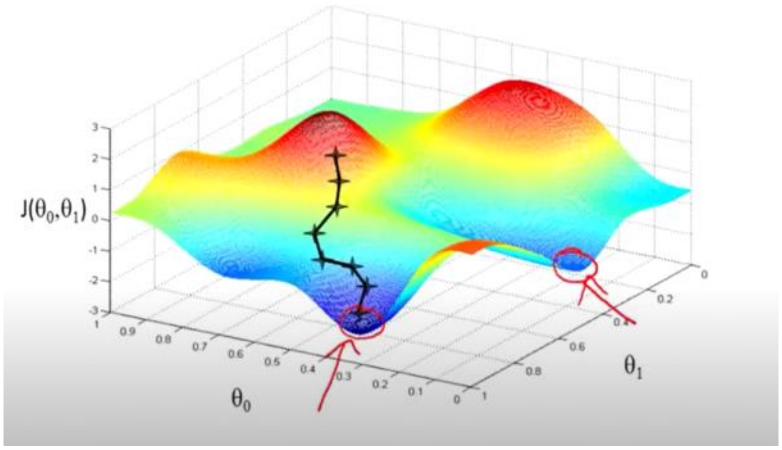
Gradient descent Algorithms

$$\theta_0 = \theta_0 - \alpha . \frac{1}{m} \sum_{i=1}^{m} \left(\theta_0 + \theta_1 (x^i) - y^i \right)$$

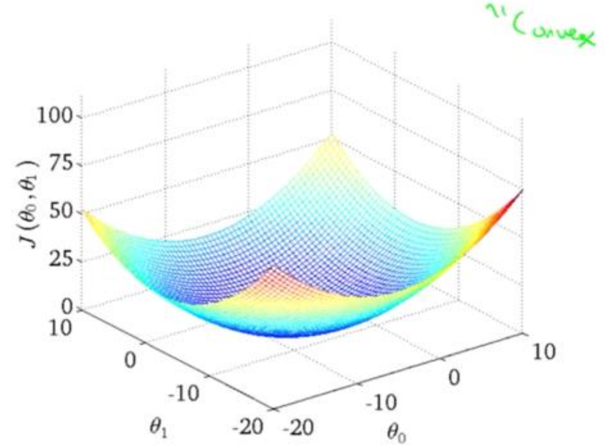
$$\theta_1 = \theta_1 - \alpha . \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^i) - y^i \right) . x^i$$

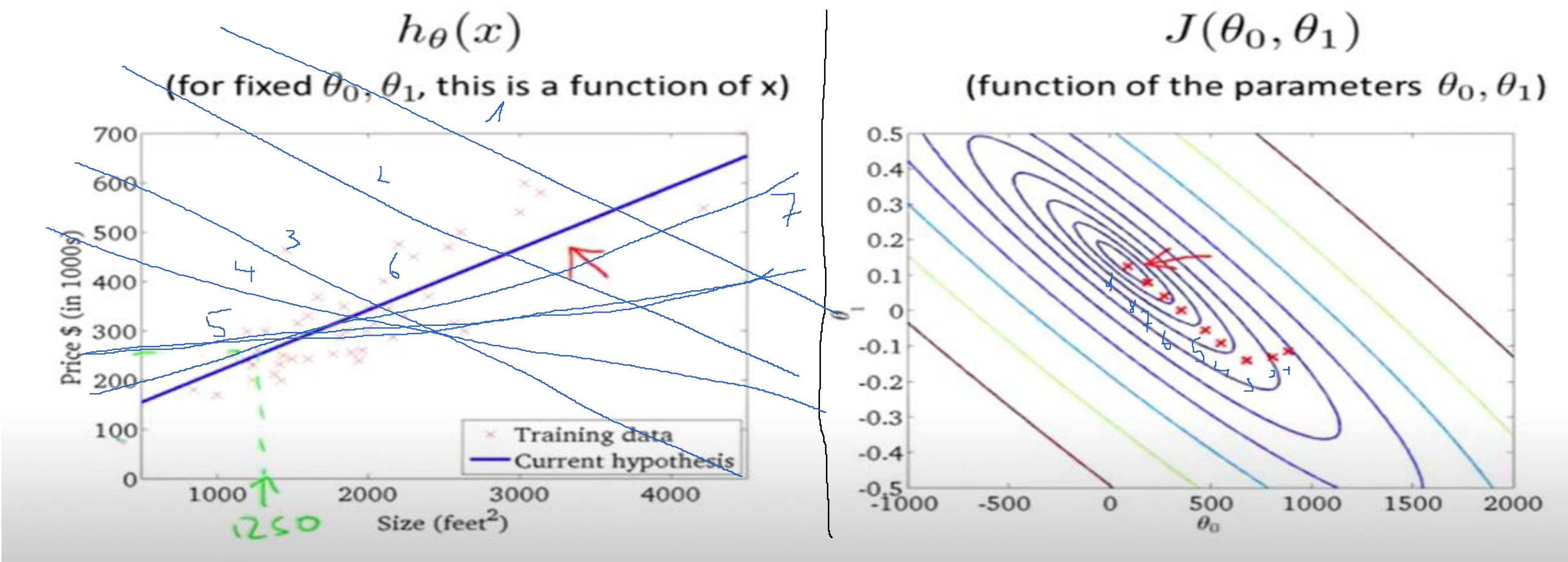
Lặp lại tới khi θ_0, θ_1 không đổi

Normal Gradient Linear descent



Linear regression Gradient descent





other name: batch Gradient descent

Multiple variables

one variable

Size x	Price y
2014	460
1416	232
1532	301

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

Mutiple Var

Size x_1	No bedroom x_2	No floor x_3	age of home x_4	price y
2104	5	1	45	460
1416	3	2	40	232
1534	3	1	30	315

Note:

n = number of feature

x^i = Input of i^{th} training exampl

x_j^i = value of feature j in i^{th}

Ex: $x^2 = \begin{bmatrix} 1416 \\ 3 \\ 2 \\ 40 \end{bmatrix} \in \mathbb{R}^4$

$x_4^2 = 40$

hypothesis.

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4$$

define $x_0 = 1 \Leftrightarrow x_0^i = 1$

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n$$

$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^{n+1} \qquad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix} \in \mathbb{R}^{n+1}$$

$$h_\theta(x) = \boxed{\theta^T x}$$

multivariable Linear regression

Gradient descent for multiple variables

$h_\theta(x) = \theta^T x$

tham số: $\theta_0, \theta_1, \theta_2, \dots \Leftrightarrow \theta$

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

$J(\theta)$

Gradient descent

Lặp lại tới khi tham số không đổi

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

$$= \theta_j - \alpha . \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^i) - y^i \right) . x_j$$

---

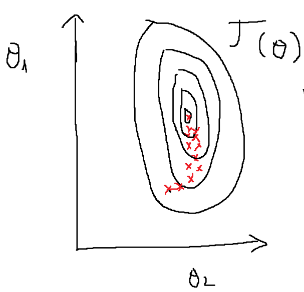
trick feature scaling

ý tưởng: biết miền giá trị của các đặc trưng

VD.

x_1 = size (0 − 2000)

x_2 = No bedroom (1 − 5)

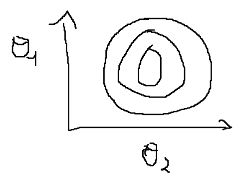


$J(\theta)$ (bất cân đối)

hình elip mảng nên hội tụ chậm

$x_1 = \frac{\text{size}}{2000}$

$x_2 = \frac{\text{No bedroom}}{5}$



(cân đối)

hình tròn nên đi đến điểm cực tiểu nhanh hơn

làm cho các đặc trưng có giá trị trong khoảng $-1 \le x_i \le 1$

Mean normalization

thường là giá trị trung bình

ý tưởng: thay x_i bằng $x_i - \mu_i$ để làm đặc trưng xấp xỉ 0

VD:

$$x_1 = \frac{x_1 - 1000}{2000} \qquad -0,5 \le x_1 \le 0,5$$

$$x_2 = \frac{x_2 - 2}{5} \qquad -0,5 \le x_2 \le 0,5$$

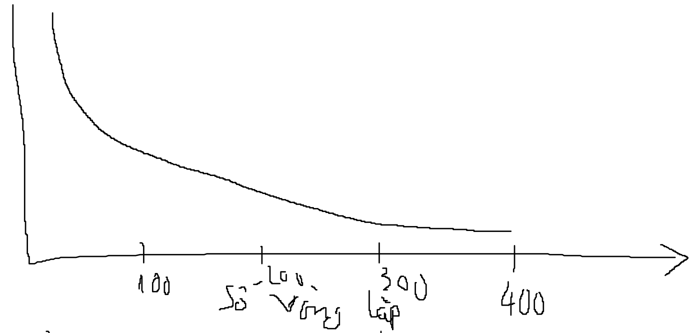
$$\boxed{x_i = \frac{x_i - avg}{Max - Min}}$$

---

debugging

$\min_\theta J(\theta)$

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



$J(\theta)$ hội tụ nếu $J(\theta)$ giảm $< 10^{-3}$ trong 1 lần lặp

nếu Gradient descent không hoạt động tốt:

- giảm α
- α quá nhỏ làm Gradient descent bị chậm

Lựa chọn feature và hồi quy đa thức

Example

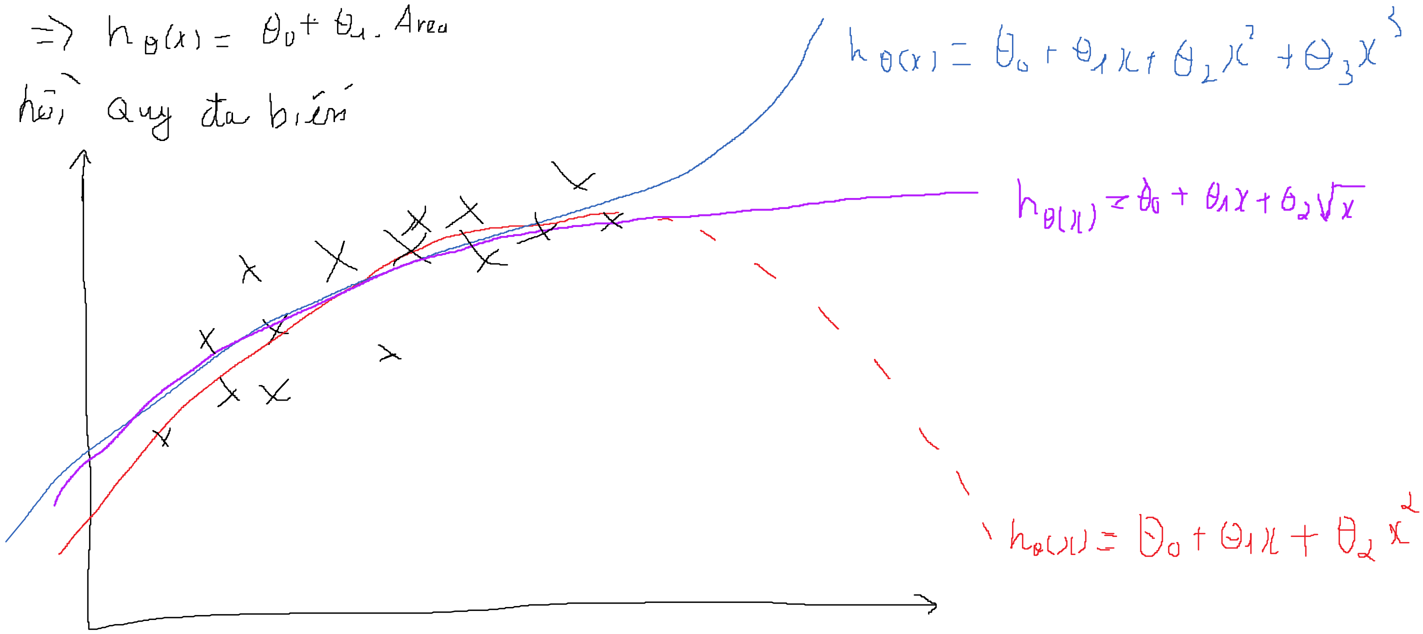
dự đoán giá nhà

$h_\theta(x) = \theta_0 + \theta_1 . front + \theta_2 . depth$

Area = front . depth

$\Rightarrow h_\theta(x) = \theta_0 + \theta_1 . Area$

hồi quy đa biến



áp dụng vào mutiple variable

$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$

$= \theta_0 + \theta_1 size + \theta_2 size^2 + \theta_3 size^3$

$x_1 = size$

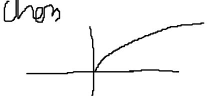
$x_2 = size^2$

$x_3 = size^3$

chú ý Feartture scaling để tối ưu

Thay vì bậc ba thì có thể chọn

$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 \sqrt{x}$



Normal equation

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n$$

đặt $X = \underbrace{[1, x_1, x_2, \dots, x_n]}_{1 \times (n+1)} \in R^{m \times (n+1)}$

$$\theta = \underbrace{[\theta_0, \theta_1, \theta_2; \dots; \theta_n]}_{1 \times (n+1)} \in R^{1 \times (n+1)}$$

$$h_\theta(x) = X . \theta^T$$

$$J_\theta = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^i) - y^i)^2 = \frac{1}{2m} \|X\theta^T - Y\|_2^2$$

$$\frac{\partial}{\partial_{\theta_n}} J_\theta = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^i) - y^i) x_n$$

X like
$$\begin{bmatrix} x_0^1 & x_1^1 & x_2^1 & \dots & x_n^1 \\ x_0^2 & x_1^2 & x_3^2 & \dots & x_n^2 \\ \dots \\ x_0^m & \dots & & & x_n^m \end{bmatrix}$$
$m \times (n+1)$

θ like $[\theta_0 \; \theta_1 \; \theta_2 \dots \theta_n]$
$1 \times (n+1)$

Y like
$$\begin{bmatrix} y^1 \\ y^2 \\ y^3 \\ \vdots \\ y^m \end{bmatrix} \quad m \times 1$$

$$\frac{\partial J_\theta}{\partial_\theta} = X^T(X\theta^T - y) = 0$$

$$\Leftrightarrow X^T X \theta^T - X^T y = 0$$

$$\Leftrightarrow X^T X \theta^T = X^T y$$

$$\boxed{\theta^T = (X^T X)^{-1} X^T y}$$

ma trận nghịch đảo của $X^T X$

đk $X^T X$ khả nghịch $\Leftrightarrow$ det $\neq 0$

So sánh với Gradient descent

GD	NE
- cần chọn α	- không cần α
- nhiều vòng lặp	- không cần vòng lặp
- hoạt động tốt ngay cả khi có nhiều feature	- chậm nếu có nhiều feature
- $O(kn^2)$	- $X^T X$ $O(n^3)$

* nếu $X^T X$ không khả nghịch

- có feature bị thừa $\Rightarrow$ hai cột, hàng tỉ lệ $\Rightarrow$ det $= 0$
- có quá nhiều feature và ít train set