Linear Regression

Machine Learning and Pattern Recognition

(Largely based on slides from Andrew Ng)

Prof. Sandra Avila
Institute of Computing (IC/Unicamp)

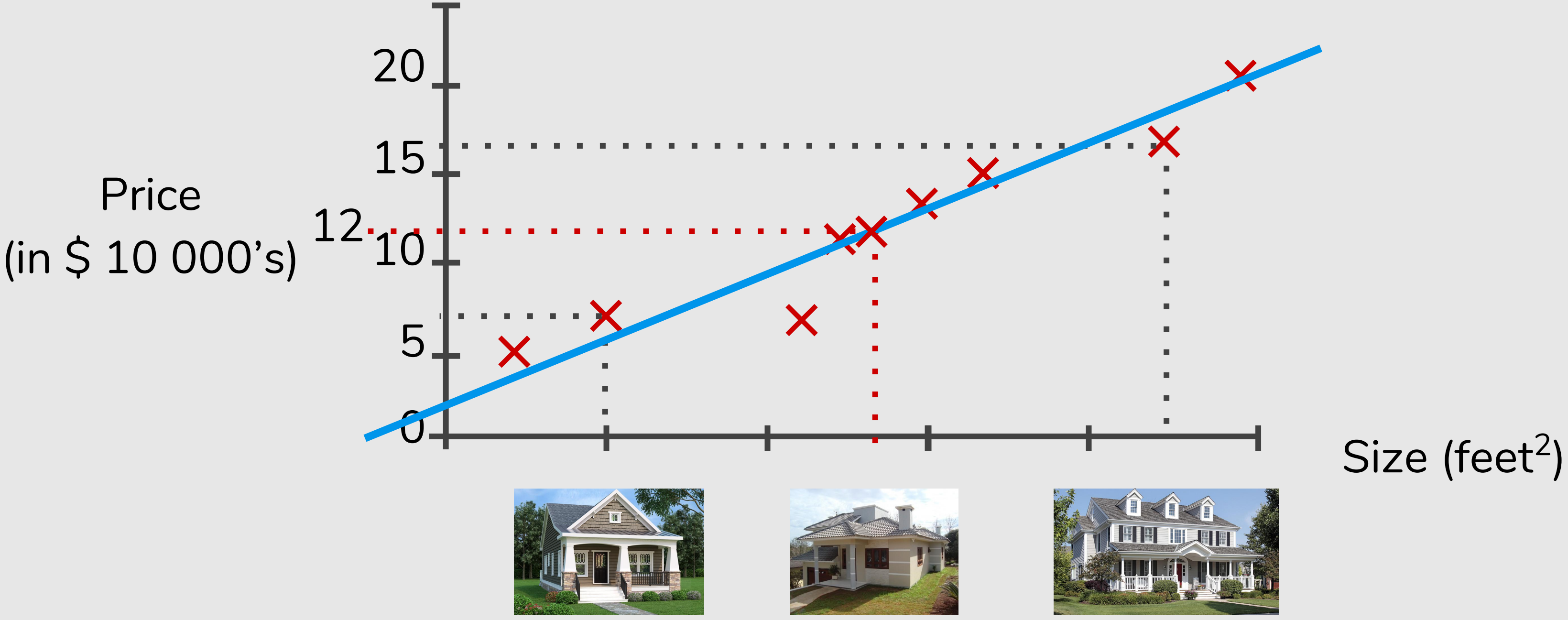
MC886/MO444, August 18, 2017

Today's Agenda

- Linear Regression with One Variable
    - Model Representation
    - Cost Function
    - Gradient Descent
- Linear Regression with Multiple Variables
    - Gradient Descent for Multiple Variables
    - Feature Scaling
    - Learning Rate
    - **Features and Polynomial Regression**
    - **Normal Equation**

Recall from last time …

House Price Prediction



Model Representation

How do we represent h ?

Training set

↓

Learning algorithm

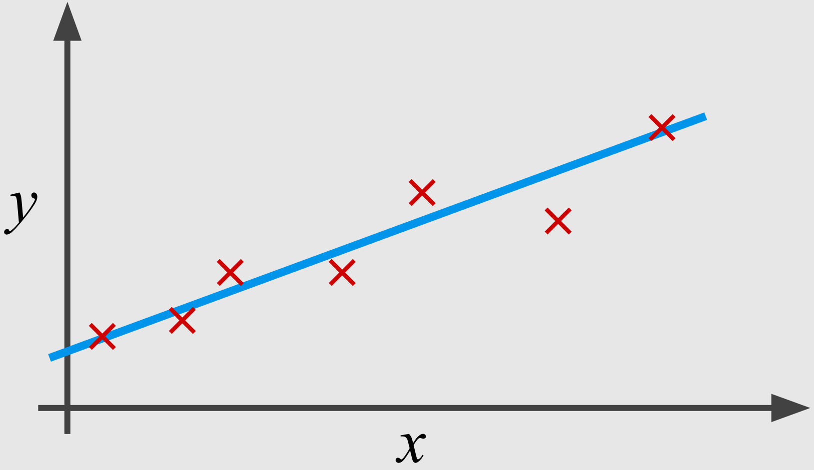
↓

Size of house → h → Estimated price

(hypothesis)

h maps x's to y's

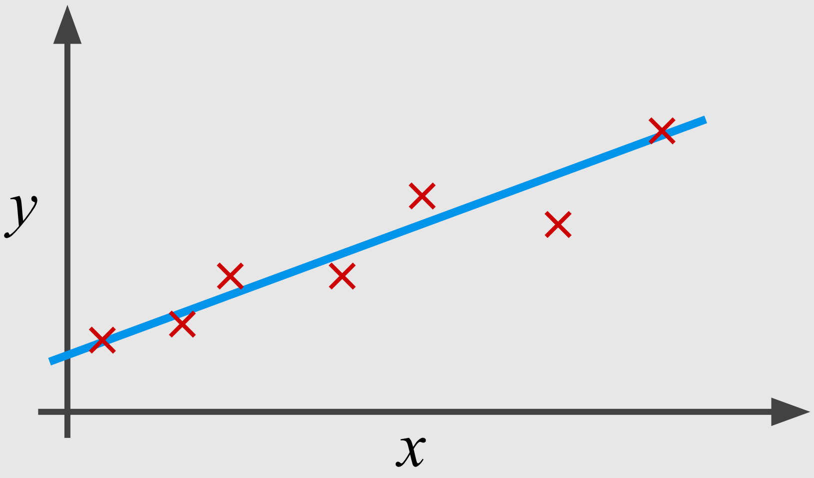
$$h_\theta(x) = \theta_0 + \theta_1 x$$



Linear regression with one variable.

Univariate linear regression.

Cost Function



Idea: Choose θ_0, θ_1 so that $h_\theta(x)$ close to y for our training examples (x,y)

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

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

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

$$\underset{\theta_0,\theta_1}{\text{minimize}} \ J(\theta_0,\theta_1)$$

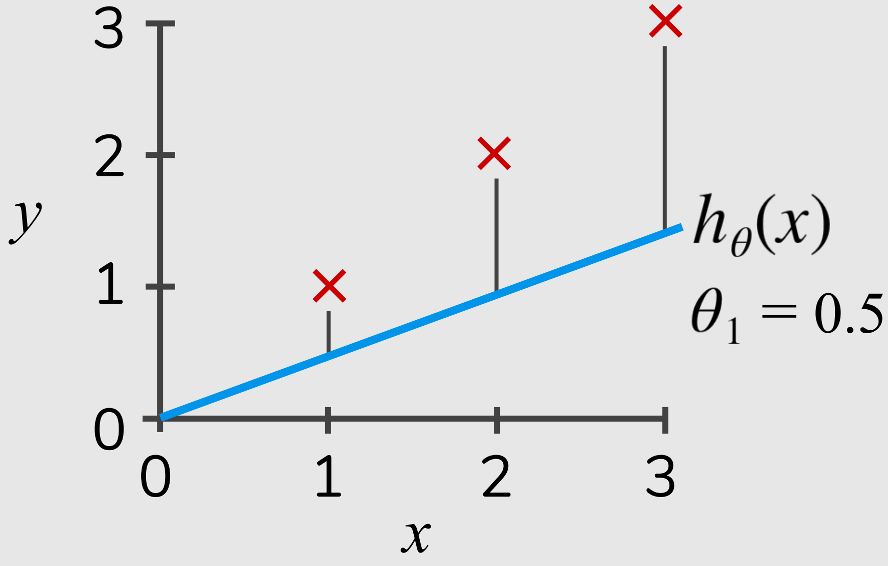
Cost function

(Squared error function)

Cost Function Intuition I

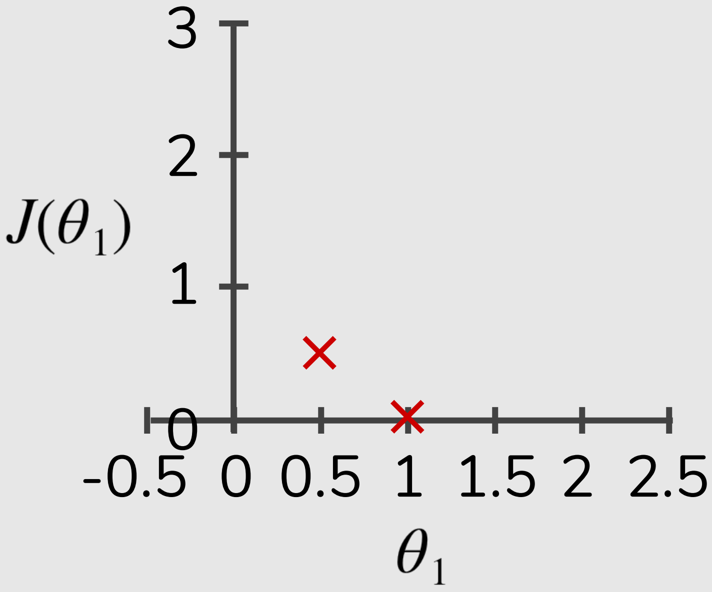
$h_\theta(x)$

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



$J(\theta_1)$

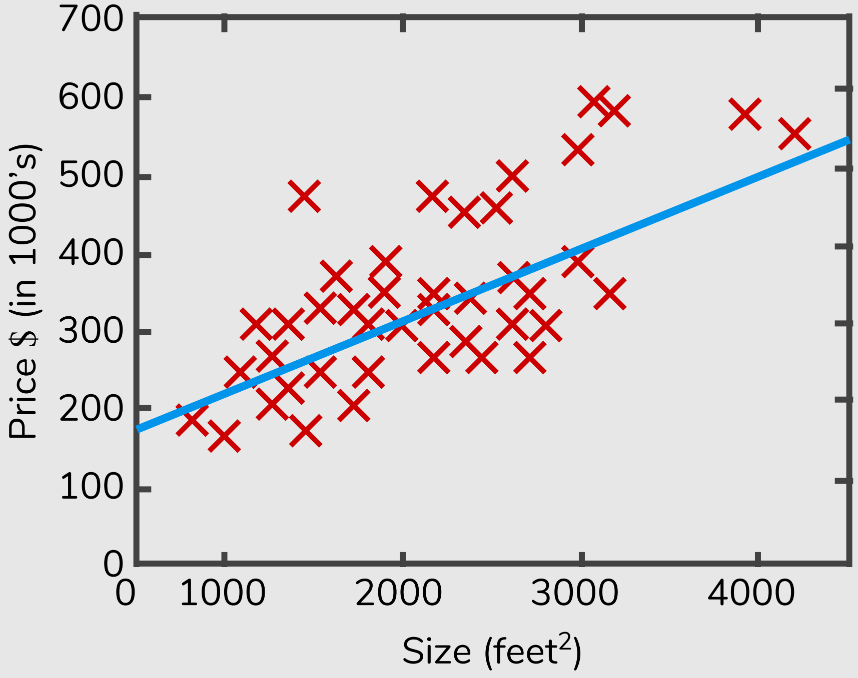
(function of the parameters θ_1)



Cost Function Intuition II

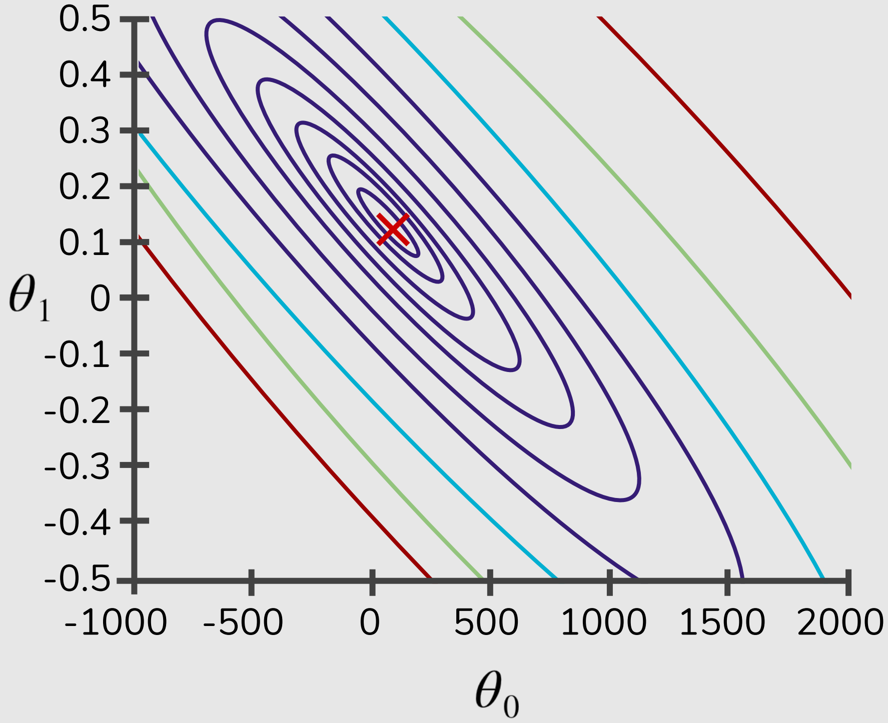
$h_\theta(x)$

(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)



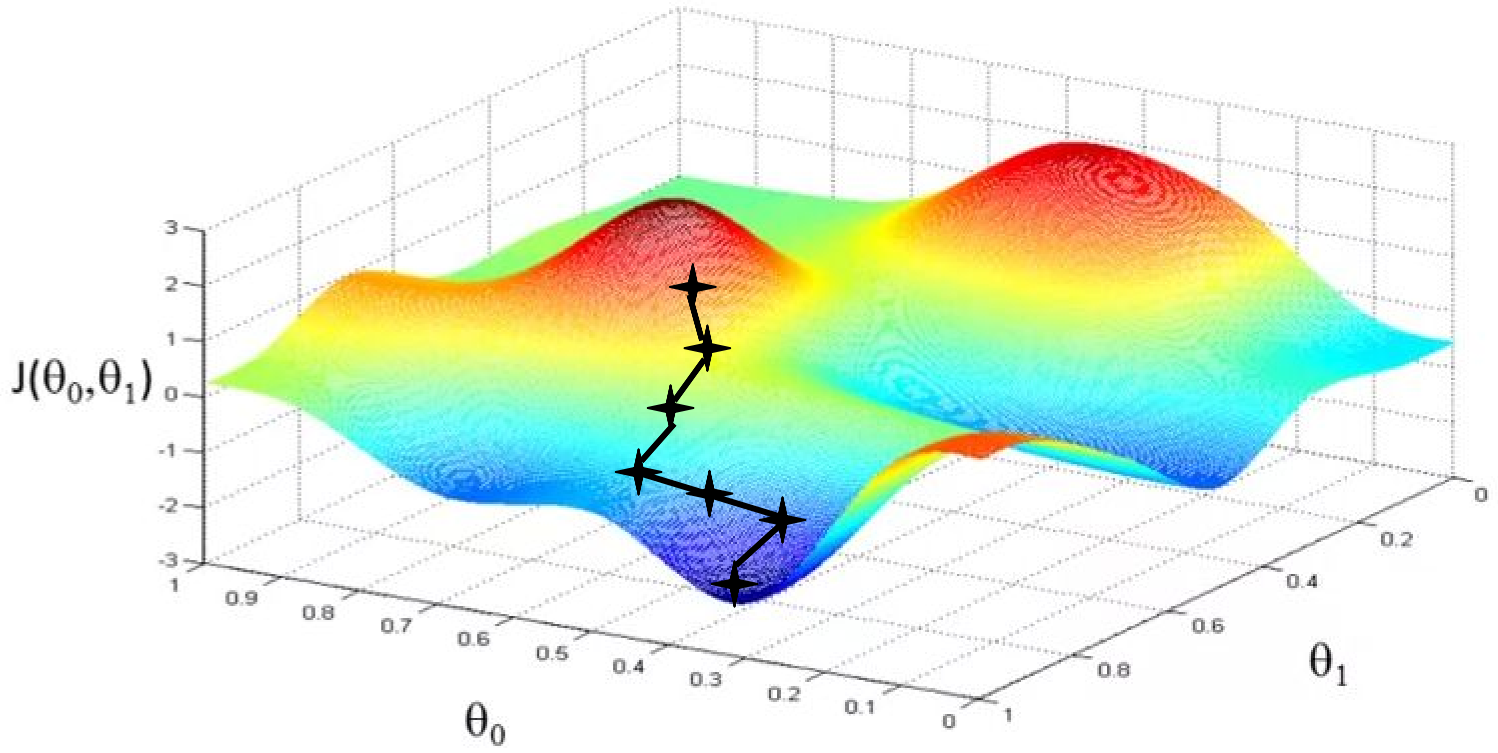
Gradient Descent

Have some function $J(\theta_0,\theta_1)$

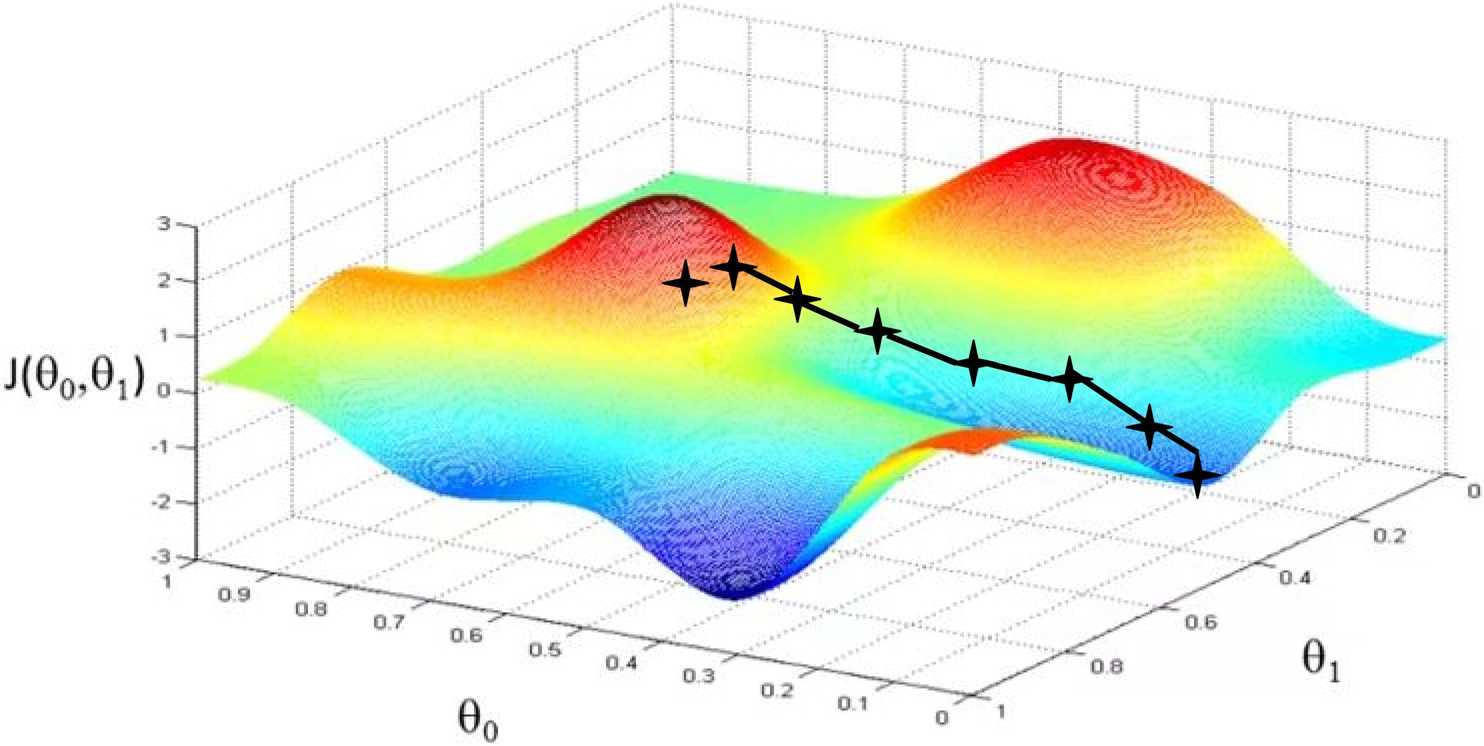
Want $\underset{\theta_0,\theta_1}{\text{minimize}}\, J(\theta_0,\theta_1)$

Outline:

- Start with some θ_0, θ_1
- Keep changing θ_0, θ_1 to reduce $J(\theta_0,\theta_1)$ until we hopefully end up at a minimum



$J(\theta_0,\theta_1)$
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
θ_0
0
0.2
0.4
0.6
0.8
1
θ_1



$J(\theta_0,\theta_1)$
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
θ_0
0
0.2
0.4
0.6
0.8
1
θ_1

Gradient Descent algorithm

repeat until convergence {

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

(simultaneously update $j = 0$ and $j = 1$)

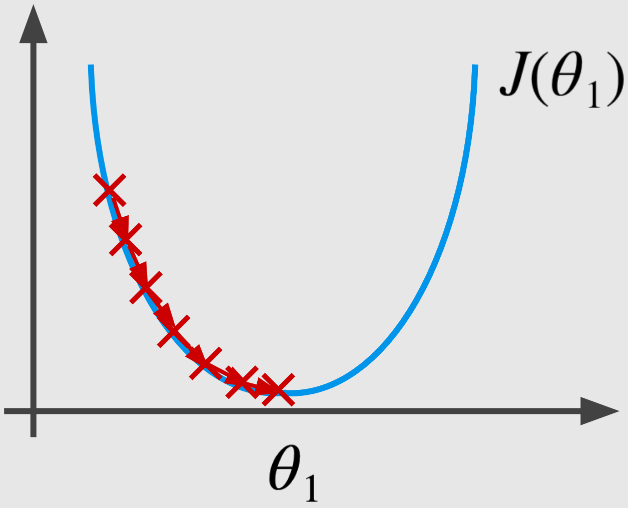
}

Learning rate

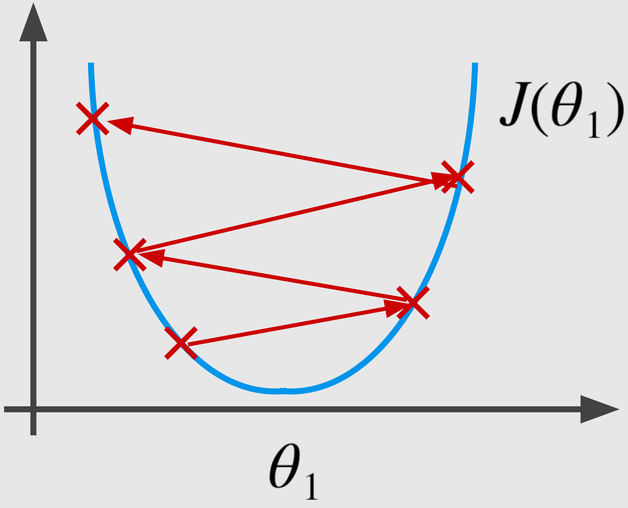
Derivative term

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

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



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



Gradient Descent algorithm

repeat until convergence {

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

(for $j = 0$ and $j = 1$)

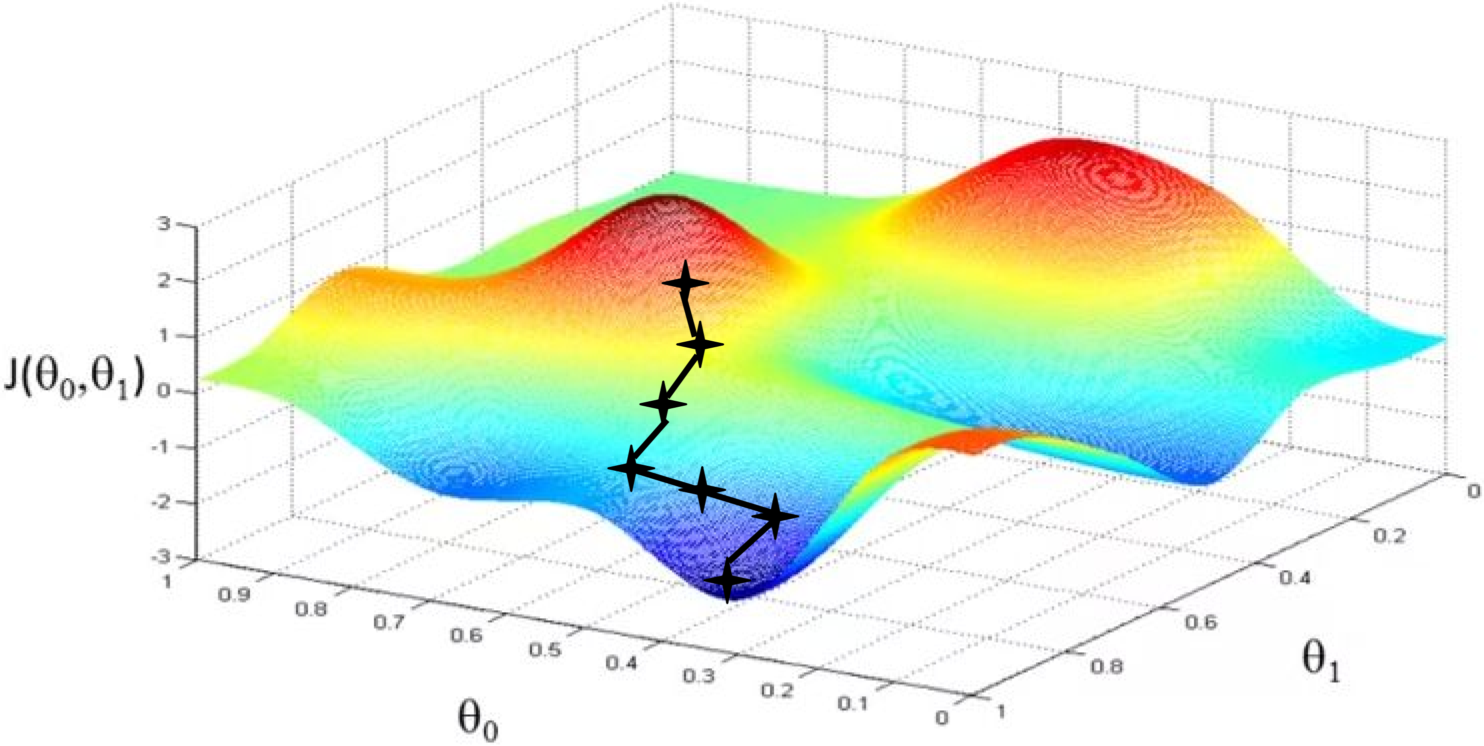
}

Linear Regression Model

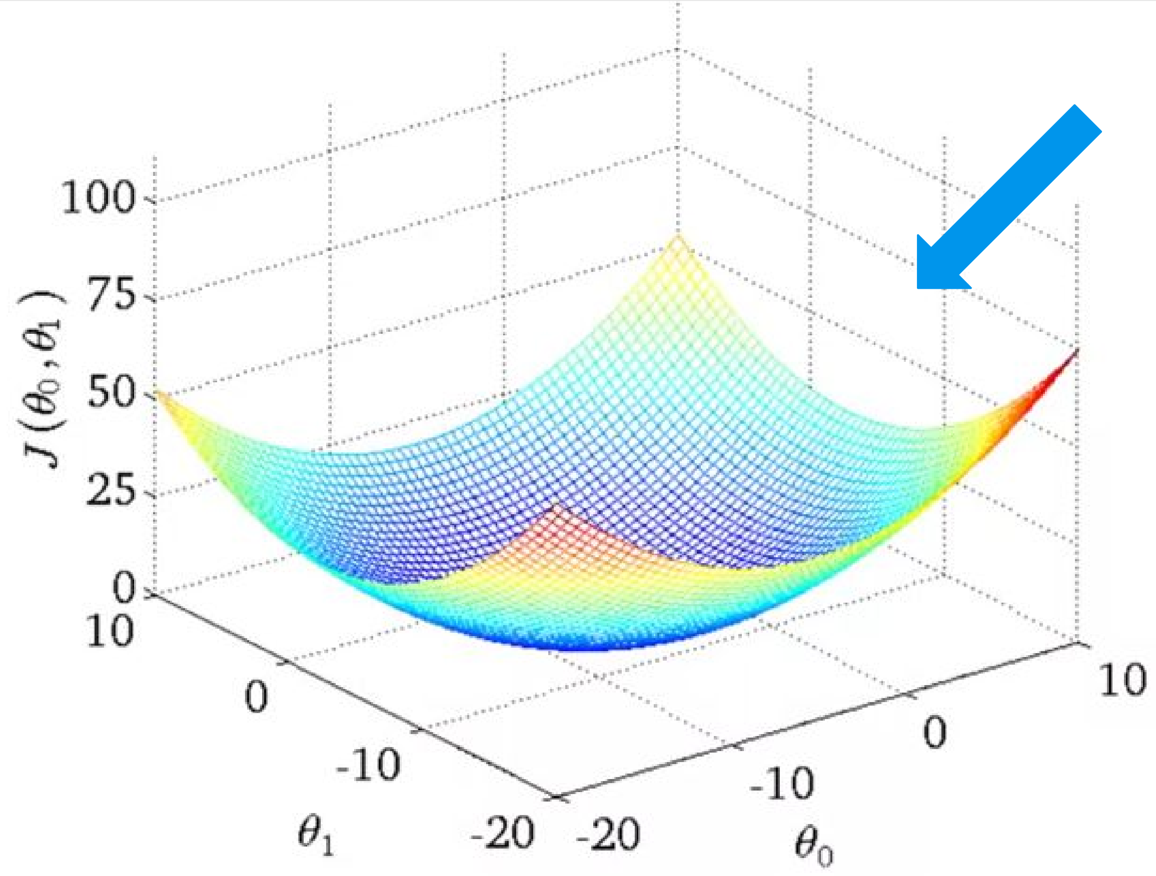
$$h_\theta(x) = \theta_0 + \theta_1 x$$

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

$$\boxed{\underset{\theta_0, \theta_1}{\text{minimize}}\, J(\theta_0, \theta_1)}$$



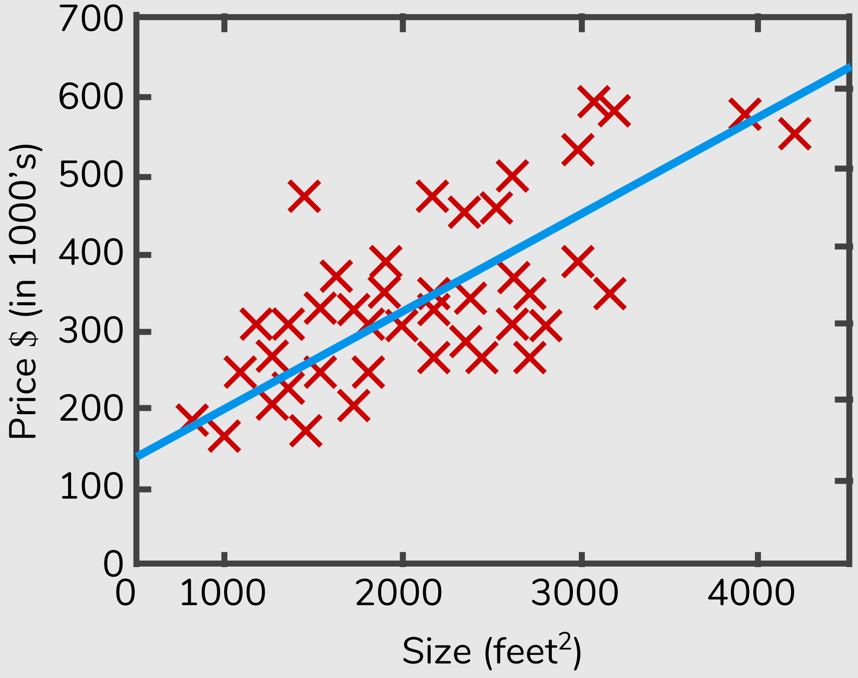
$J(\theta_0,\theta_1)$
3
2
1
0
-1
-2
-3
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
θ_0
0
0.2
0.4
0.6
0.8
1
θ_1



Convex Function

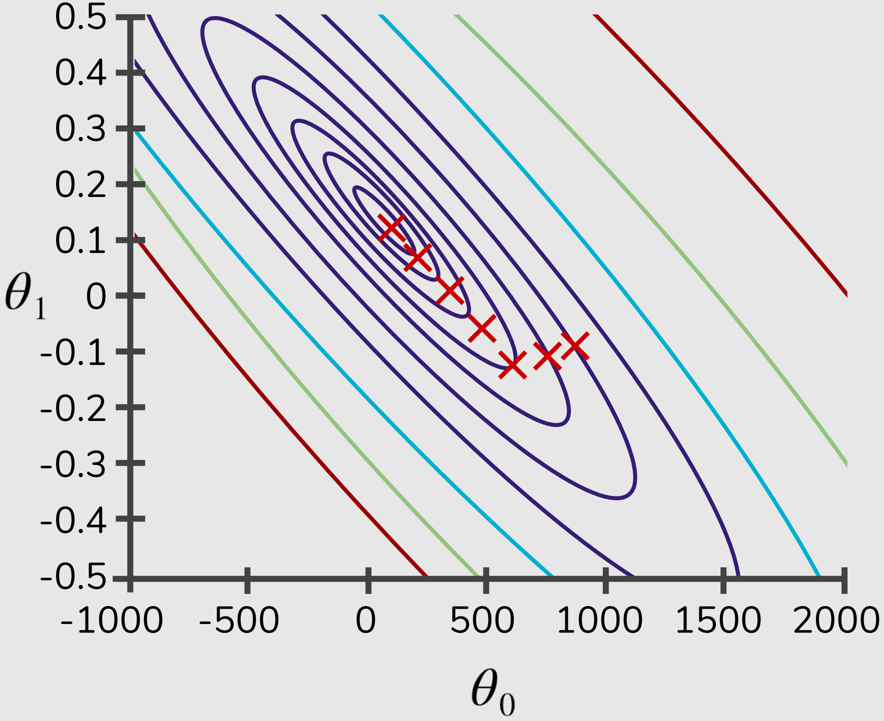
$h_\theta(x)$

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

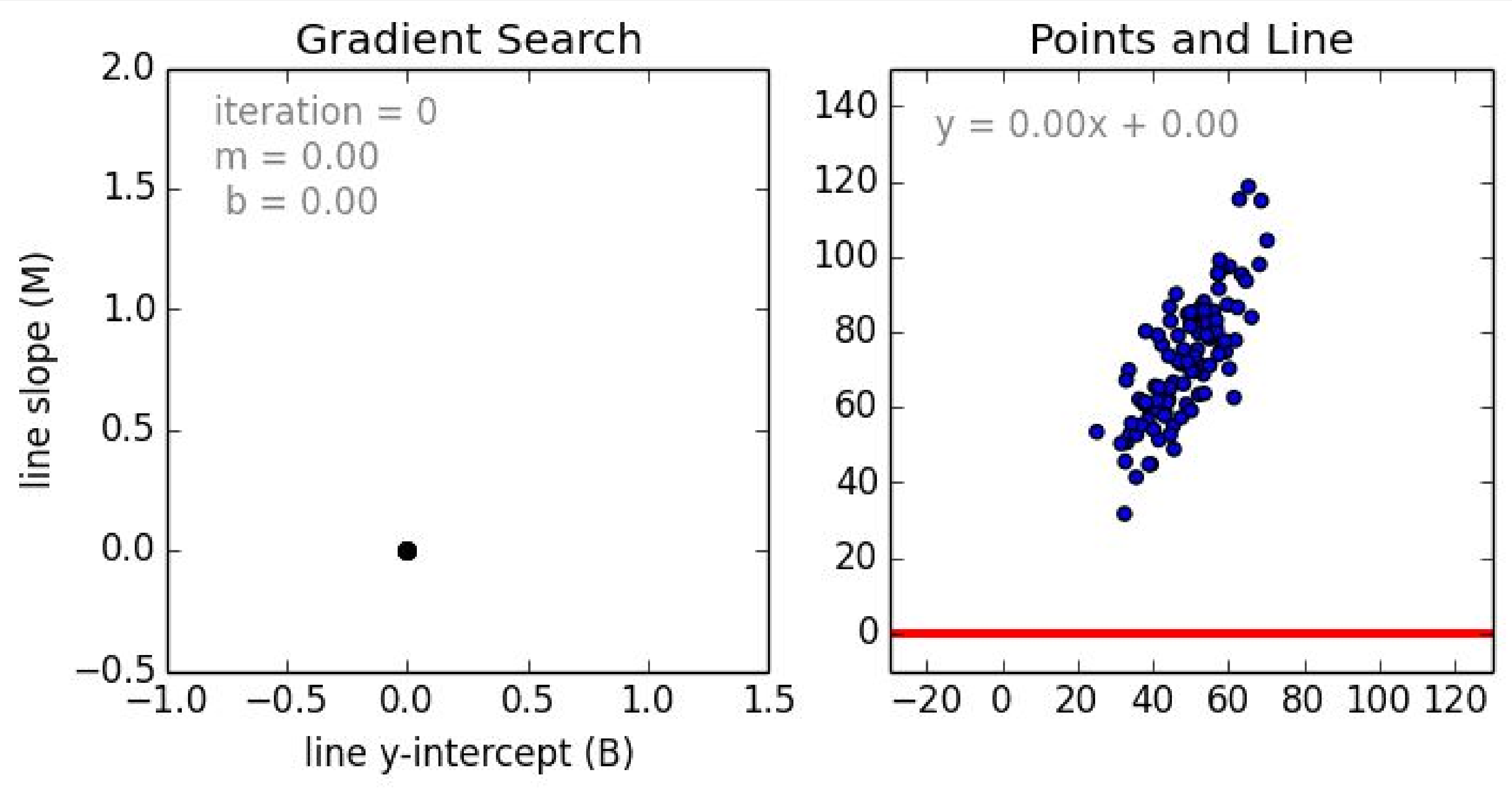


$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)

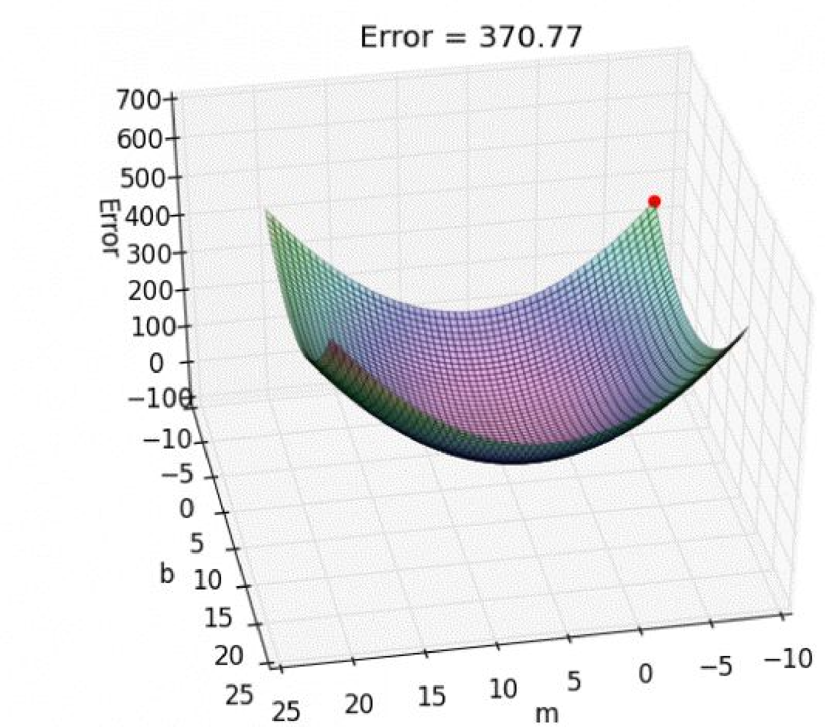


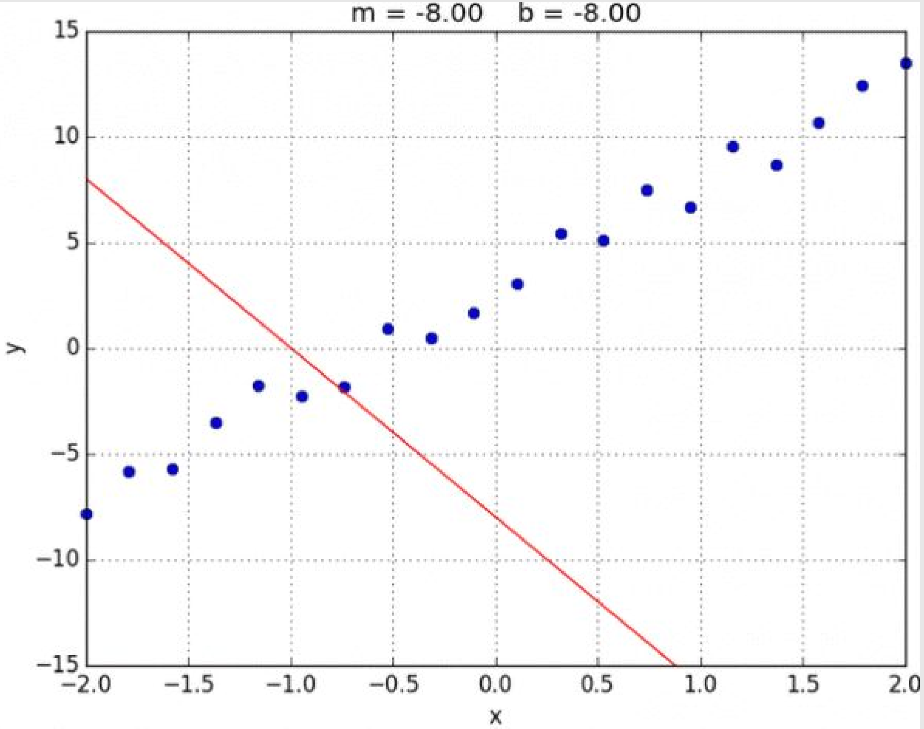
$$h_\theta(x) = \theta_0 + \theta_1 x \quad \Rightarrow \quad y = b + mx$$



Credit: https://github.com/mattnedrich/GradientDescentExample/raw/master/gradient_descent_example.gif

$$y = b + mx$$





Credit: https://alykhantejani.github.io/a-brief-introduction-to-gradient-descent/

“Batch” Gradient Descent

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

“Batch” Gradient Descent

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

- Stochastic Gradient Descent
- Mini-batch Gradient Descent

“Batch” Gradient Descent

repeat until convergence {

$$\left.\begin{aligned}\theta_0 &:= \theta_0 - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)}) \\ \theta_1 &:= \theta_1 - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})\,x^{(i)}\end{aligned}\right\}$$

update θ_0 and θ_1 simultaneously

}

Stochastic Gradient Descent

Each step of gradient descent uses **one training example**.

repeat until convergence {

for $i := 1, \ldots, m$ {

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

$$\theta_1 := \theta_1 - \alpha(h_\theta(x^{(i)}) - y^{(i)})x^{(i)}$$

}

}

Mini-batch Gradient Descent

Each step of gradient descent uses b **training examples**.

Say $b = 10$, $m = 1000$.
repeat until convergence {

for $i \coloneqq 1, 11, 21 \ldots, 991$ {

$$\theta_0 := \theta_0 - \alpha \frac{1}{10} \sum_{i=k}^{i+9} (h_\theta(x^{(k)}) - y^{(k)})$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{10} \sum_{i=k}^{i+9} (h_\theta(x^{(k)}) - y^{(k)})\, x^{(k)}$$

} }

Linear Regression with multiple variables

Multiple ~~Variables~~ Features

Size in feet^2 x_1	Number of bedrooms x_2	Number of floors x_3	Age of home (years) x_4	Price ($) in 1000's y
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	2	36	178
...	...	...	...	...

Notation:

n = number of features

$x^{(i)}$ = input (features) of i^{th} training example

$x_j^{(i)}$ = value of features j in i^{th} training example

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

For convenience of notation, define $x_0 = 1$.

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

$$\begin{bmatrix} \theta_0 & \theta_1 & \cdots & \theta_n \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}$$

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

Multivariate linear regression.

Hypothesis: $h_\theta(x) = \theta^T x = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$

Parameters: $\theta_0, \theta_1, \ldots, \theta_n$

Cost Function: $J(\theta_0, \theta_1, \ldots, \theta_n) = \dfrac{1}{2m} \displaystyle\sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$

Gradient Descent:

repeat {

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

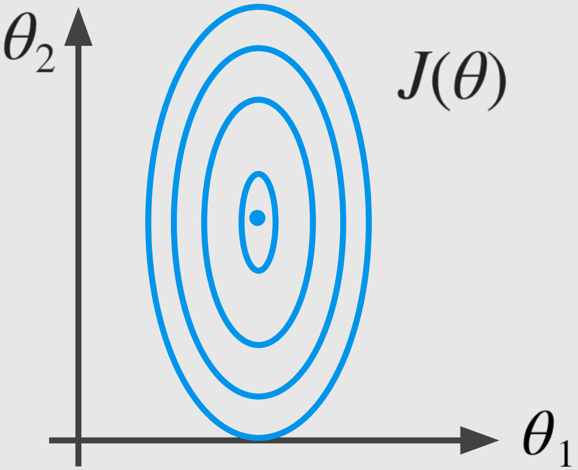
} (simultaneously update for every $j = 0, 1, \ldots, n$)

Feature Scaling

Feature Scaling

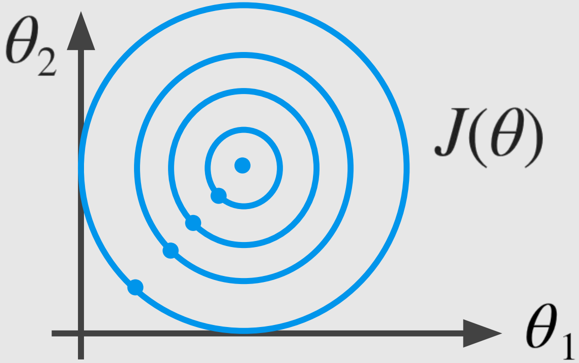
Idea: Make sure features are on similar scale.

E.g. x_1 = size (0–2000 feet^2)
x_2 = number of bedrooms (1–5)



$$x_1 = \frac{\text{size (feet}^2\text{)}}{2000}$$

$$x_2 = \frac{\text{number of bedrooms}}{5}$$



Mean Normalization

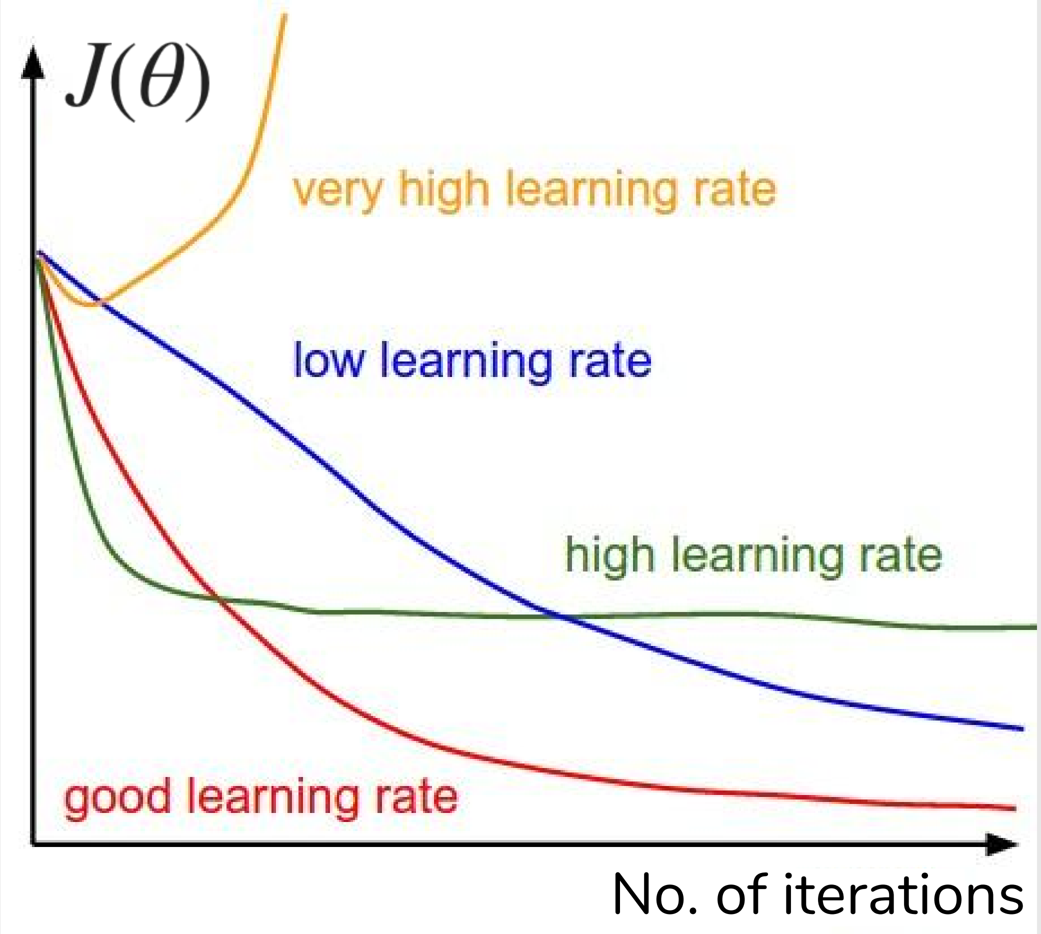
Replace x_i with $x_i - \mu_i$ to make features have approximately zero mean (do not apply to $x_0 = 1$).

E.g. $x_1 = \dfrac{\text{size} - 1000}{2000}$ $\Rightarrow$ $-0.5 \le x_1 \le 0.5$

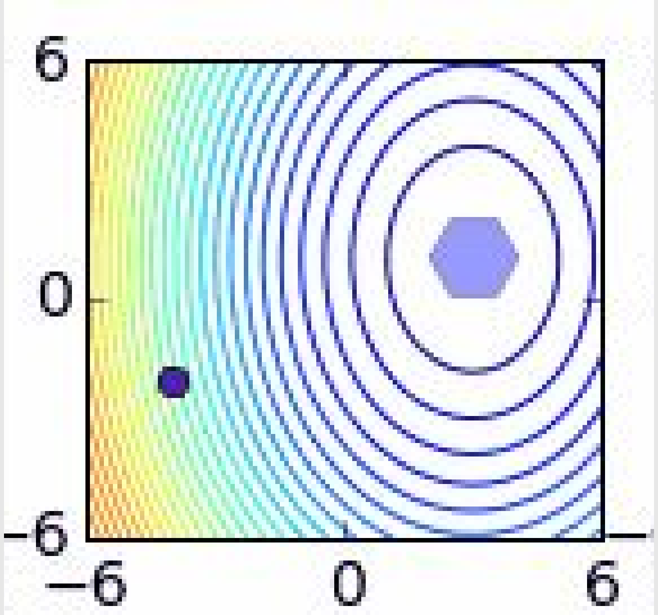
$x_2 = \dfrac{\#\text{bedrooms} - 2.5}{5}$ $\Rightarrow$ $-0.5 \le x_2 \le 0.5$

$$x_1 = \frac{x_1 - \mu_1}{s_1} \qquad x_2 = \frac{x_2 - \mu_2}{s_2}$$

Learning Rate



$J(\theta)$
very high learning rate
low learning rate
high learning rate
good learning rate
No. of iterations



Purple: $\alpha = 0.016$

Black: $\alpha = 0.1$

Red: $\alpha = 0.6$

Credit: https://blog.sigopt.com/posts/tensorflow-convnets-on-a-budget-with-bayesian-optimization

Features and Polynomial Regression

Housing prices prediction

$$h_\theta(x) = \theta_0 + \theta_1 \times \text{frontage} + \theta_2 \times \text{depth}$$

Housing prices prediction

$$h_\theta(x) = \theta_0 + \theta_1 \times \underbrace{\text{frontage}}_{x_1} + \theta_2 \times \underbrace{\text{depth}}_{x_2}$$

Housing prices prediction

$$h_\theta(x) = \theta_0 + \theta_1 \times \text{frontage} + \theta_2 \times \text{depth}$$

$$\text{frontage} \rightarrow x_1 \qquad \text{depth} \rightarrow x_2$$

Area $x = \text{frontage} \times \text{depth}$

Housing prices prediction

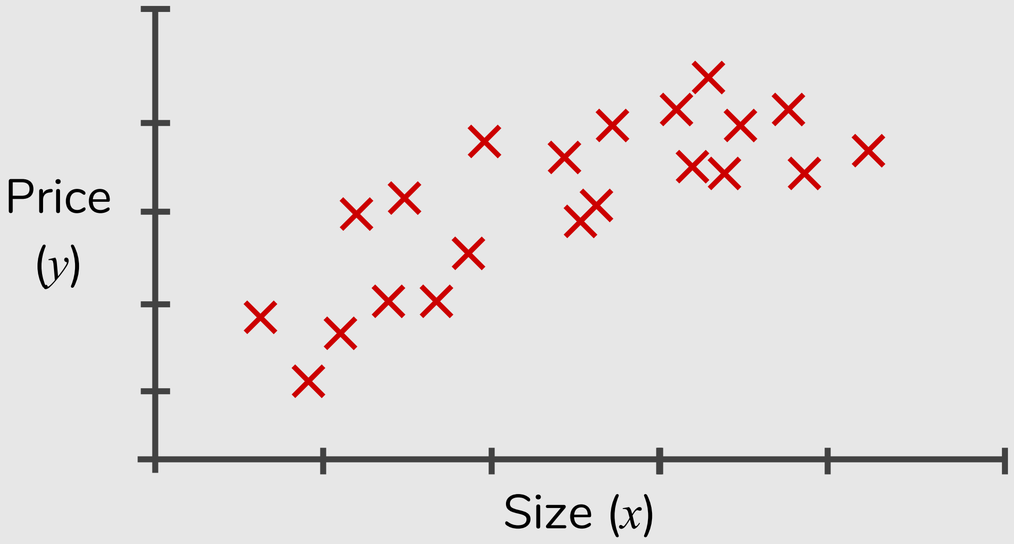
$$h_\theta(x) = \theta_0 + \theta_1 \times \text{frontage} + \theta_2 \times \text{depth}$$

$$x_1 \qquad\qquad x_2$$

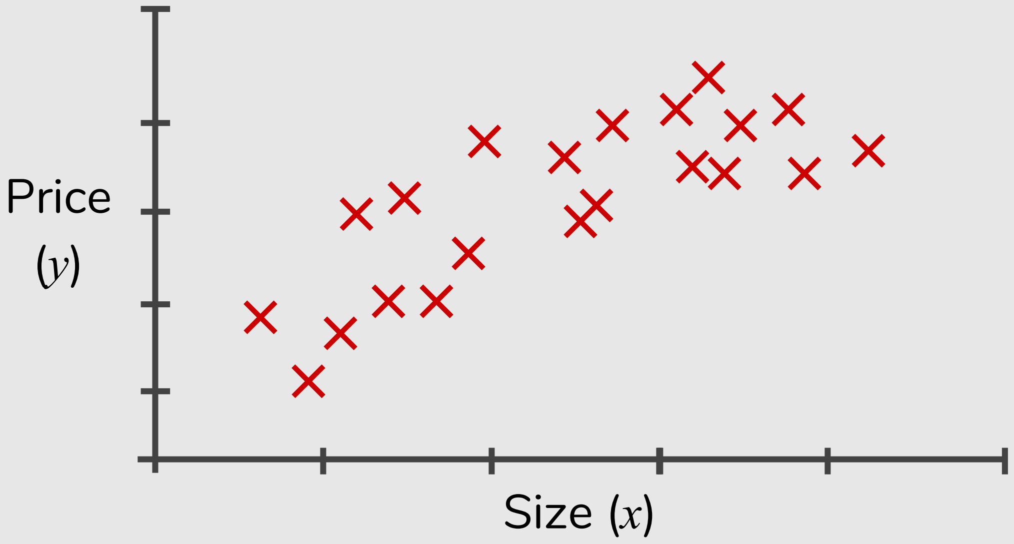
Area x = frontage $\times$ depth

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

Polynomial Regression

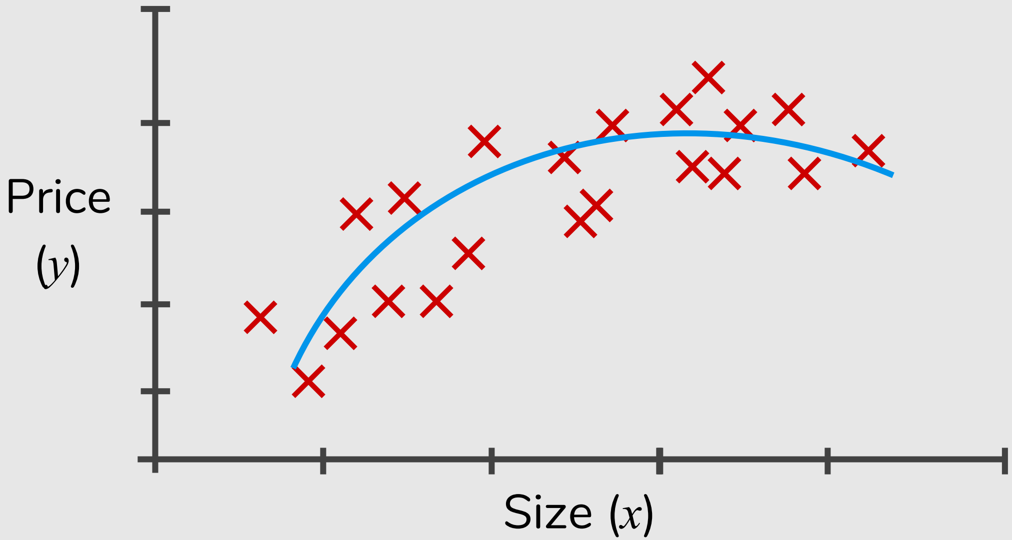


Polynomial Regression



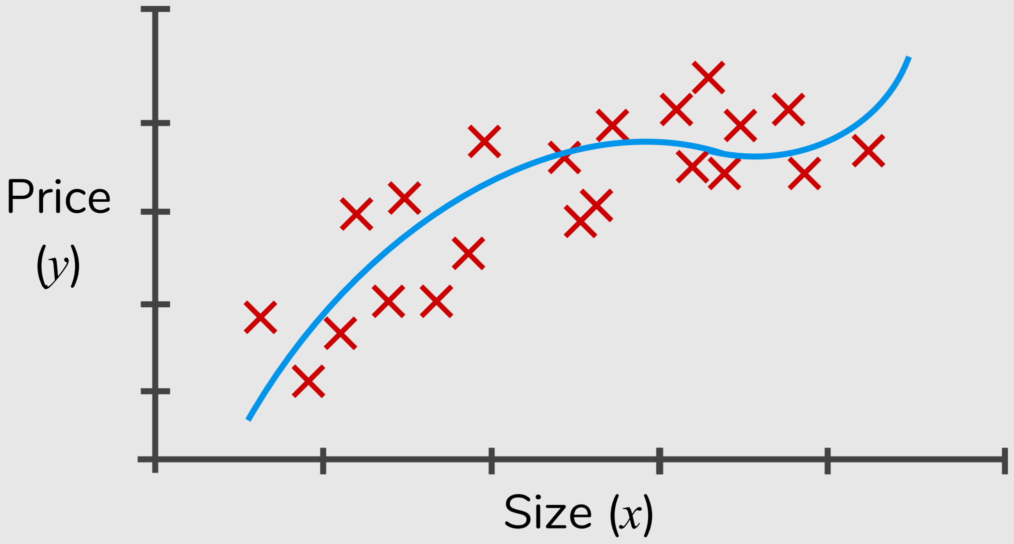
$$\theta_0 + \theta_1 x + \theta_2 x^2$$

Polynomial Regression



$$\theta_0 + \theta_1 x + \theta_2 x^2$$

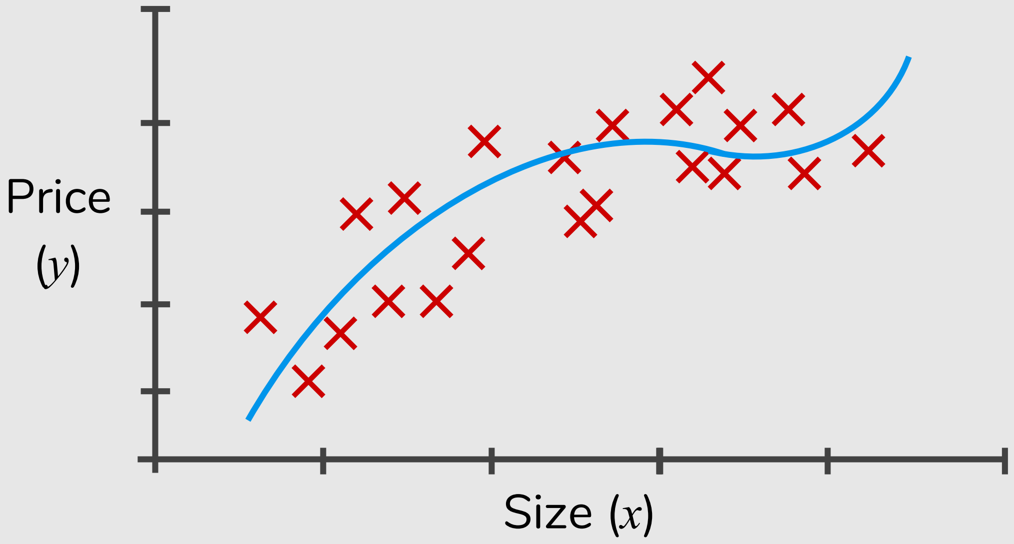
Polynomial Regression



$$\theta_0 + \theta_1 x + \theta_2 x^2$$

$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$$

Polynomial Regression



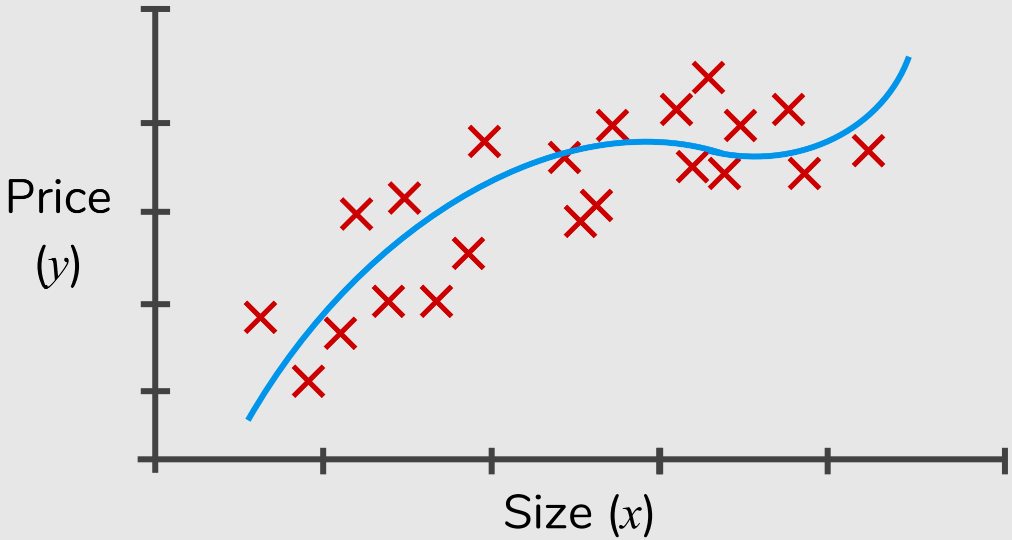
$$\theta_0 + \theta_1 x + \theta_2 x^2$$

$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$$

$$x_1 = (\text{size})$$
$$x_2 = (\text{size})^2$$
$$x_3 = (\text{size})^3$$

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$$
$$= \theta_0 + \theta_1(\text{size}) + \theta_2(\text{size})^2 + \theta_3(\text{size})^3$$

Polynomial Regression



$$\theta_0 + \theta_1 x + \theta_2 x^2$$

$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$$

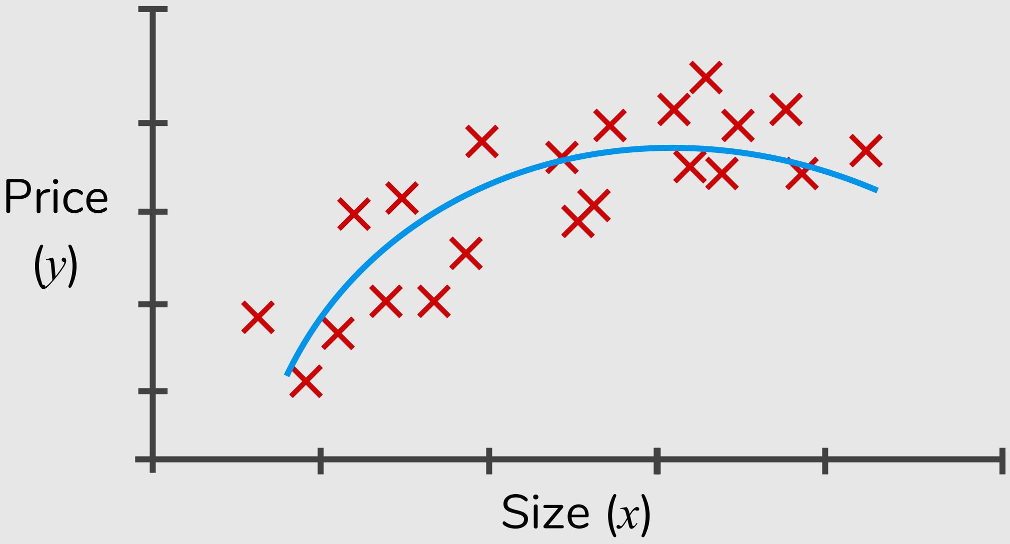
$$x_1 = (\text{size}) \quad : 1 - 1{,}000$$

$$x_2 = (\text{size})^2 \quad : 1 - 1{,}000{,}000$$

$$x_3 = (\text{size})^3 \quad : 1 - 10^9$$

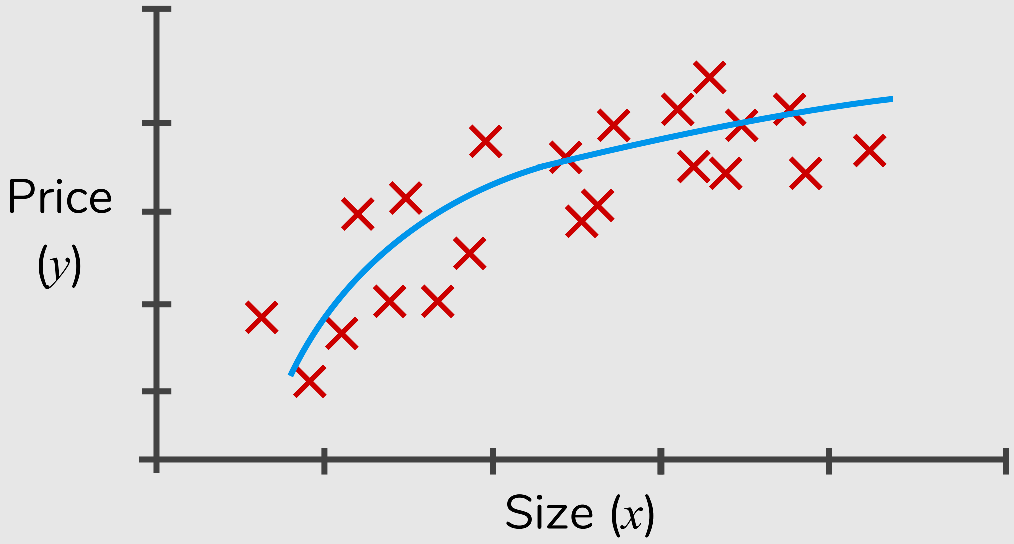
$$\begin{aligned} h_\theta(x) &= \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 \\ &= \theta_0 + \theta_1(\text{size}) + \theta_2(\text{size})^2 + \theta_3(\text{size})^3 \end{aligned}$$

Choice of Features



$$h_\theta(x) = \theta_0 + \theta_1(\text{size}) + \theta_2(\text{size})^2$$

Choice of Features

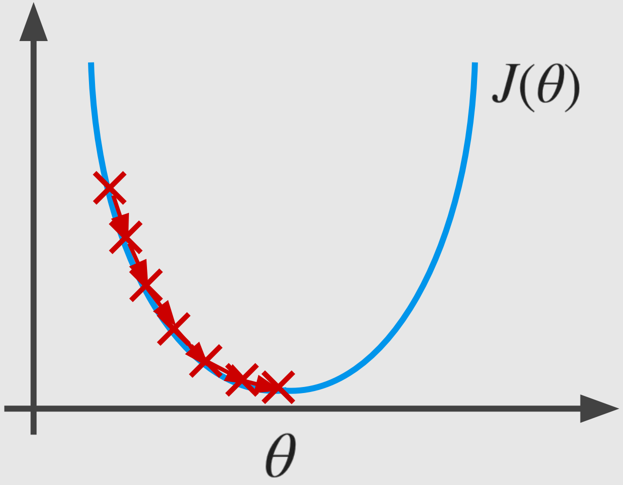


$$h_\theta(x) = \theta_0 + \theta_1(\text{size}) + \theta_2(\text{size})^2$$

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

Normal Equation

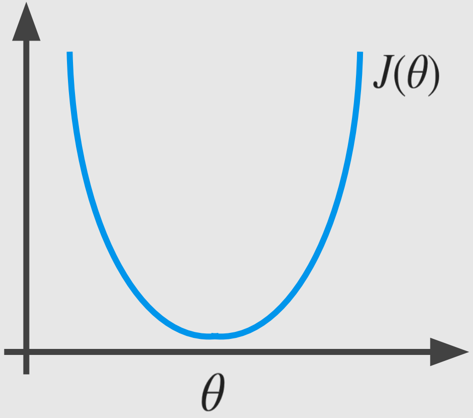
Gradient Descent



Normal equation: Method to solve θ **analytically**.

Intuition: If 1D ($\theta \in \mathbb{R}$)

$$J(\theta) = a\theta^2 + b\theta + c$$

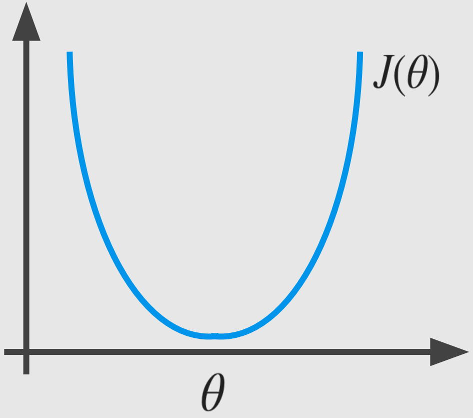


Intuition: If 1D ($\theta \in \mathbb{R}$)

$$J(\theta) = a\theta^2 + b\theta + c$$

$$\frac{d}{d\theta} J(\theta) = \ldots = 0$$

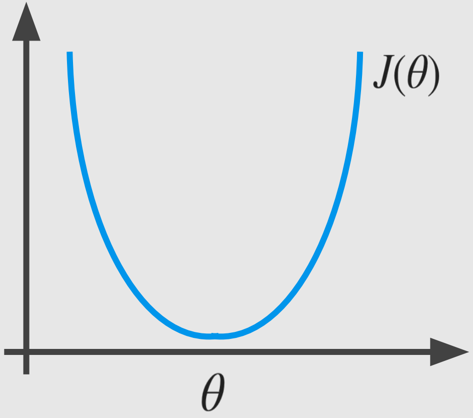
Solve for θ



Intuition: If 1D ($\theta \in \mathbb{R}$)

$$J(\theta) = a\theta^2 + b\theta + c$$

$$\frac{d}{d\theta} J(\theta) = \ldots = 0$$ Solve for θ



$\theta \in \mathbb{R}^{n+1}$ $$J(\theta_0, \theta_1, \ldots, \theta_n) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

$$\frac{\partial}{\partial \theta_j} J(\theta) = \ldots = 0$$ Solve for $\theta_0, \theta_1, \ldots, \theta_n$

Examples: $m = 4$.

Size (feet^2) x_1	Number of bedrooms x_2	Number of floors x_3	Age of home (years) x_4	Price ($\$$) in 1000's y
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178

Examples: $m = 4$.

↓ x_0	Size (feet^2) x_1	Number of bedrooms x_2	Number of floors x_3	Age of home (years) x_4	Price ($) in 1000's y
1	2104	5	1	45	460
1	1416	3	2	40	232
1	1534	3	2	30	315
1	852	2	1	36	178

Examples: $m = 4$.

x_0	Size (feet^2) x_1	Number of bedrooms x_2	Number of floors x_3	Age of home (years) x_4	Price (\$) in 1000's y
1	2104	5	1	45	460
1	1416	3	2	40	232
1	1534	3	2	30	315
1	852	2	1	36	178

Examples: $m = 4$.

x_0	Size (feet^2) x_1	Number of bedrooms x_2	Number of floors x_3	Age of home (years) x_4	Price ($) in 1000's y
1	2104	5	1	45	460
1	1416	3	2	40	232
1	1534	3	2	30	315
1	852	2	1	36	178

$$X = \begin{bmatrix} 1 & 2104 & 5 & 1 & 45 \\ 1 & 1416 & 3 & 2 & 40 \\ 1 & 1534 & 3 & 2 & 30 \\ 1 & 852 & 2 & 1 & 36 \end{bmatrix}$$

Examples: $m = 4$.

x_0	Size (feet^2) x_1	Number of bedrooms x_2	Number of floors x_3	Age of home (years) x_4	Price (\$) in 1000's y
1	2104	5	1	45	460
1	1416	3	2	40	232
1	1534	3	2	30	315
1	852	2	1	36	178

$$X = \begin{bmatrix} 1 & 2104 & 5 & 1 & 45 \\ 1 & 1416 & 3 & 2 & 40 \\ 1 & 1534 & 3 & 2 & 30 \\ 1 & 852 & 2 & 1 & 36 \end{bmatrix}$$

Examples: $m = 4$.

x_0	Size (feet^2) x_1	Number of bedrooms x_2	Number of floors x_3	Age of home (years) x_4	Price (\$) in 1000's y
1	2104	5	1	45	460
1	1416	3	2	40	232
1	1534	3	2	30	315
1	852	2	1	36	178

$$X = \begin{bmatrix} 1 & 2104 & 5 & 1 & 45 \\ 1 & 1416 & 3 & 2 & 40 \\ 1 & 1534 & 3 & 2 & 30 \\ 1 & 852 & 2 & 1 & 36 \end{bmatrix} \qquad y = \begin{bmatrix} 460 \\ 232 \\ 315 \\ 178 \end{bmatrix}$$

Examples: $m = 4$.

x_0	Size (feet^2) x_1	Number of bedrooms x_2	Number of floors x_3	Age of home (years) x_4	Price ($) in 1000's y
1	2104	5	1	45	460
1	1416	3	2	40	232
1	1534	3	2	30	315
1	852	2	1	36	178

$$X = \begin{bmatrix} 1 & 2104 & 5 & 1 & 45 \\ 1 & 1416 & 3 & 2 & 40 \\ 1 & 1534 & 3 & 2 & 30 \\ 1 & 852 & 2 & 1 & 36 \end{bmatrix}_{m \times (n+1)} \qquad y = \begin{bmatrix} 460 \\ 232 \\ 315 \\ 178 \end{bmatrix}_{m}$$

Examples: $m = 4$.

x_0	Size (feet^2) x_1	Number of bedrooms x_2	Number of floors x_3	Age of home (years) x_4	Price (\$) in 1000's y
1	2104	5	1	45	460
1	1416	3	2	40	232
1	1534	3	2	30	315
1	852	2	1	36	178

$$X = \begin{bmatrix} 1 & 2104 & 5 & 1 & 45 \\ 1 & 1416 & 3 & 2 & 40 \\ 1 & 1534 & 3 & 2 & 30 \\ 1 & 852 & 2 & 1 & 36 \end{bmatrix}_{m \times (n+1)} \qquad y = \begin{bmatrix} 460 \\ 232 \\ 315 \\ 178 \end{bmatrix}_{m}$$

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

m **examples** $(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})$ **and** n **features**

$$x^{(i)} = \begin{bmatrix} x_0^{(i)} \\ x_1^{(i)} \\ x_2^{(i)} \\ \vdots \\ x_n^{(i)} \end{bmatrix} \in \mathbb{R}^{n+1}$$

m **examples** $(x^{(1)}, y^{(1)}), \ \dots, \ (x^{(m)}, y^{(m)})$ **and** n **features**

$$x^{(i)} = \begin{bmatrix} x_0^{(i)} \\ x_1^{(i)} \\ x_2^{(i)} \\ \vdots \\ x_n^{(i)} \end{bmatrix} \in \mathbb{R}^{n+1}$$

$$X = \begin{bmatrix} \\ \\ \\ \end{bmatrix}$$

m **examples** $(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})$ **and** n **features**

$$x^{(i)} = \begin{bmatrix} x_0^{(i)} \\ x_1^{(i)} \\ x_2^{(i)} \\ \vdots \\ x_n^{(i)} \end{bmatrix} \in \mathbb{R}^{n+1}$$

$$X = \begin{bmatrix} \text{---} \; (x^{(1)})^{\mathrm{T}} \; \text{---} \\ \\ \\ \\ \end{bmatrix}$$

m **examples** $(x^{(1)}, y^{(1)}), \ \ldots, \ (x^{(m)}, y^{(m)})$ **and** n **features**

$$x^{(i)} = \begin{bmatrix} x_0^{(i)} \\ x_1^{(i)} \\ x_2^{(i)} \\ \vdots \\ x_n^{(i)} \end{bmatrix} \in \mathbb{R}^{n+1}$$

$$X = \begin{bmatrix} \text{——} \ (x^{(1)})^{\mathrm{T}} \ \text{——} \\ \text{——} \ (x^{(2)})^{\mathrm{T}} \ \text{——} \\ \\ \\ \end{bmatrix}$$

m **examples** $(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})$ **and** n **features**

$$x^{(i)} = \begin{bmatrix} x_0^{(i)} \\ x_1^{(i)} \\ x_2^{(i)} \\ \vdots \\ x_n^{(i)} \end{bmatrix} \in \mathbb{R}^{n+1}$$

$$X = \begin{bmatrix} \text{———} & (x^{(1)})^{\mathrm{T}} & \text{———} \\ \text{———} & (x^{(2)})^{\mathrm{T}} & \text{———} \\ \text{———} & \vdots & \text{———} \\ \text{———} & (x^{(m)})^{\mathrm{T}} & \text{———} \end{bmatrix}$$

m **examples** $(x^{(1)}, y^{(1)}), \ \ldots, \ (x^{(m)}, y^{(m)})$ **and** n **features**

$$x^{(i)} = \begin{bmatrix} x_0^{(i)} \\ x_1^{(i)} \\ x_2^{(i)} \\ \vdots \\ x_n^{(i)} \end{bmatrix} \in \mathbb{R}^{n+1}$$

$$X = \begin{bmatrix} \text{——} & (x^{(1)})^{\mathrm{T}} & \text{——} \\ \text{——} & (x^{(2)})^{\mathrm{T}} & \text{——} \\ \text{——} & \vdots & \text{——} \\ \text{——} & (x^{(m)})^{\mathrm{T}} & \text{——} \end{bmatrix}$$

E.g. $x^{(i)} = \begin{bmatrix} 1 \\ x_1^{(i)} \end{bmatrix}$

m **examples** $(x^{(1)}, y^{(1)}), \ \dots, \ (x^{(m)}, y^{(m)})$ **and** n **features**

$$x^{(i)} = \begin{bmatrix} x_0^{(i)} \\ x_1^{(i)} \\ x_2^{(i)} \\ \vdots \\ x_n^{(i)} \end{bmatrix} \in \mathbb{R}^{n+1}$$

$$X = \begin{bmatrix} \text{——} & (x^{(1)})^{\mathrm{T}} & \text{——} \\ \text{——} & (x^{(2)})^{\mathrm{T}} & \text{——} \\ \text{——} & \vdots & \text{——} \\ \text{——} & (x^{(m)})^{\mathrm{T}} & \text{——} \end{bmatrix}$$

E.g. $x^{(i)} = \begin{bmatrix} 1 \\ x_1^{(i)} \end{bmatrix}$ $\quad X = \begin{bmatrix} 1 & x_1^{(1)} \\ \vdots & \vdots \\ 1 & x_m^{(1)} \end{bmatrix}_{m \times 2}$

m **examples** $(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})$ **and** n **features**

$$X = \begin{bmatrix} \text{———} & (x^{(1)})^{\mathrm{T}} & \text{———} \\ \text{———} & (x^{(2)})^{\mathrm{T}} & \text{———} \\ \text{———} & \vdots & \text{———} \\ \text{———} & (x^{(m)})^{\mathrm{T}} & \text{———} \end{bmatrix} \qquad y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix}$$

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

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

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

$(X^T X)^{-1}$ is inverse of matrix $X^T X$.

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

$(X^T X)^{-1}$ is inverse of matrix $X^T X$.

Deriving the Normal Equation using matrix calculus ...

☞ https://ayearofai.com/rohan-3-deriving-the-normal-equation-using-matrix-calculus-1a1b16f65dda

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

$(X^T X)^{-1}$ is inverse of matrix $X^T X$.

Deriving the Normal Equation using matrix calculus ...

☞ https://ayearofai.com/rohan-3-deriving-the-normal-equation-using-matrix-calculus-1a1b16f65dda

What if $X^T X$ is noninvertible?

What if $X^T X$ is noninvertible?

The common causes might be having :

- Redundant features, where two features are very closely related (i.e. they are linearly dependent).
- Too many features (e.g. $m \leq n$). In this case, delete some features or use “regularization”.

Gradient Descent

- Need to choose α.
- Needs many iterations.

m examples and n features

Normal Equation

- No need to choose α.
- Don't need to iterate.

Gradient Descent

- Need to choose α.
- Needs many iterations.
- Works well even when n is large.

Normal Equation

- No need to choose α.
- Don't need to iterate.
- Need to compute $(X^T X)^{-1} \rightarrow O(n^3)$.
- Slow if n is very large.

m examples and n features

References

Machine Learning Books

- Hands-On Machine Learning with Scikit-Learn and TensorFlow, Chap. 2 & 4
- Pattern Recognition and Machine Learning, Chap. 3
- Machine Learning: a Probabilistic Perspective, Chap. 7

Machine Learning Courses

- https://www.coursera.org/learn/machine-learning, Week 1 & 2