2. Linear Regression ที่มี 1 ตัวแปร

2.1 Model Representation
การเขียนอธิบายโมเดล

Krittameth Teachasrisaksakul

ราคาบ้าน (Portland, OR)

Supervised Learning: รู้คำตอบที่ถูกต้องของแต่ละตัวอย่างในข้อมูล

ปัญหา Regression: ทำนายผลลัพธ์ที่เป็นจำนวนจริงต่อเนื่อง (real-valued output)

ปัญหา Classification: ทำนายผลลัพธ์ที่เป็นค่าไม่ต่อเนื่อง (discrete-valued output)

การทำนายราคาบ้าน เป็น ปัญหา Regression

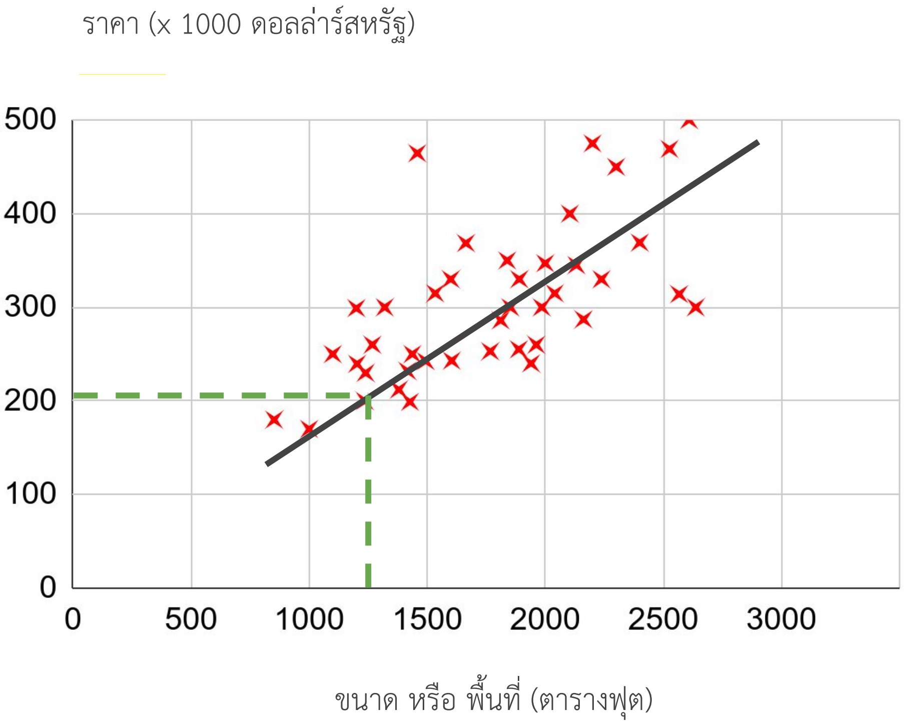


Figure source: Andrew Ng's Machine Learning course

ชุดข้อมูล Training Set ของราคาบ้าน (Portland, OR)

Training set คือ ส่วนหนึ่งของชุดข้อมูลที่แบ่งมาฝึก / สร้างโมเดล

สัญลักษณ์ / ตัวแปร

- m = จำนวนตัวอย่างใน training set
- x = variable หรือ feature ที่เป็น input
- y = variable หรือ feature ที่เป็น output หรือ target (เป้าหมาย)
- *variable: ตัวแปร, feature: คุณลักษณะ*
- (x, y): training example 1 อัน
- $(x^{(i)}, y^{(i)})$: training example ตัวที่ i

ขนาด หรือ พื้นที่ (ตร.ฟุต) (x)	ราคาบ้าน (ดอลล่าร์) x 1,000 (y)
2104	460
1416	232
1534	315
852	178
...	...

$m = 4$

$x^{(1)} = 2104$
$x^{(2)} = 1416$
$y^{(1)} = 460$

คำถาม: ชุดข้อมูล Training Set ของราคาบ้าน

ถ้าใช้ข้อมูลชุดเดิม และ $(x^{(i)}, y^{(i)})$ เป็น training example ตัวที่ i

$y^{(3)}$ = ?

ขนาด หรือ พื้นที่ (ตร.ฟุต) (*x*)	**ราคาบ้าน (ดอลล่าร์) x 1,000 (*y*)**
2104	460
1416	232
1534	315
852	178
…	…

อธิบายปัญหา Supervised Learning ด้วยคณิตศาสตร์

เป้าหมาย: มีข้อมูล training set เพื่อใช้เรียนรู้ฟังก์ชั่น $h : X \rightarrow Y$

เพื่อให้ $h(x)$ ทำนายค่า y ได้ดี

ฟังก์ชั่น $h(x)$ คือ hypothesis (สมมติฐาน)

กระบวนการ จะเป็น ดังนี้

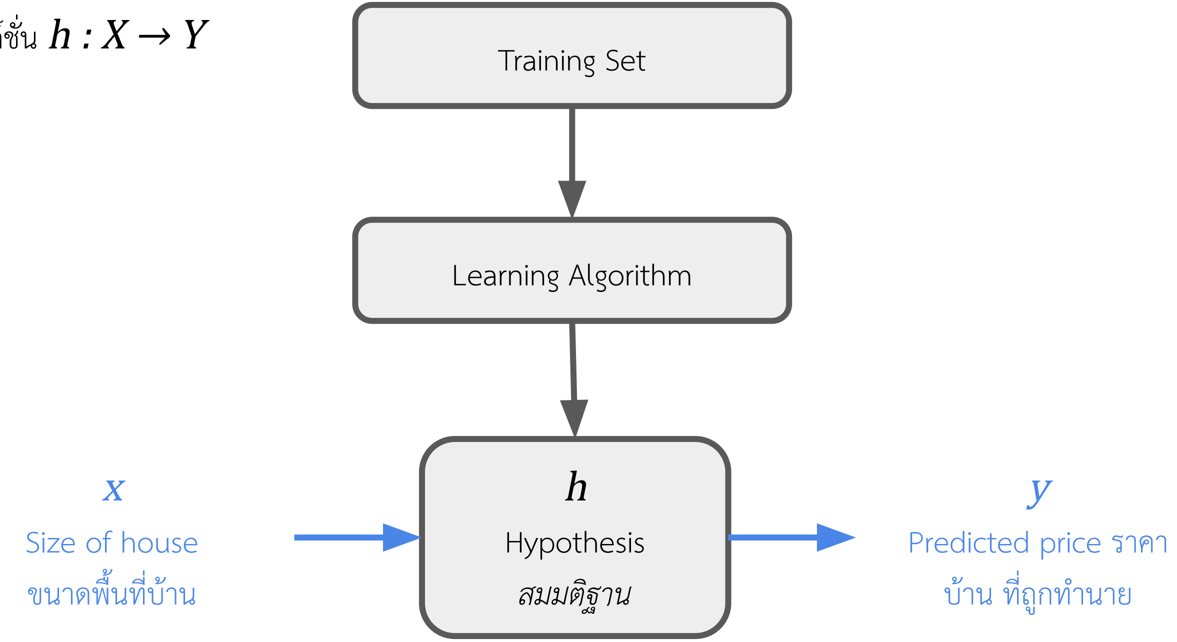


Image Credit: Andrew Ng's Machine Learning Course, Coursera

อธิบายปัญหา Supervised Learning ด้วยคณิตศาสตร์

เขียนอธิบาย $h(x)$ อย่างไร?

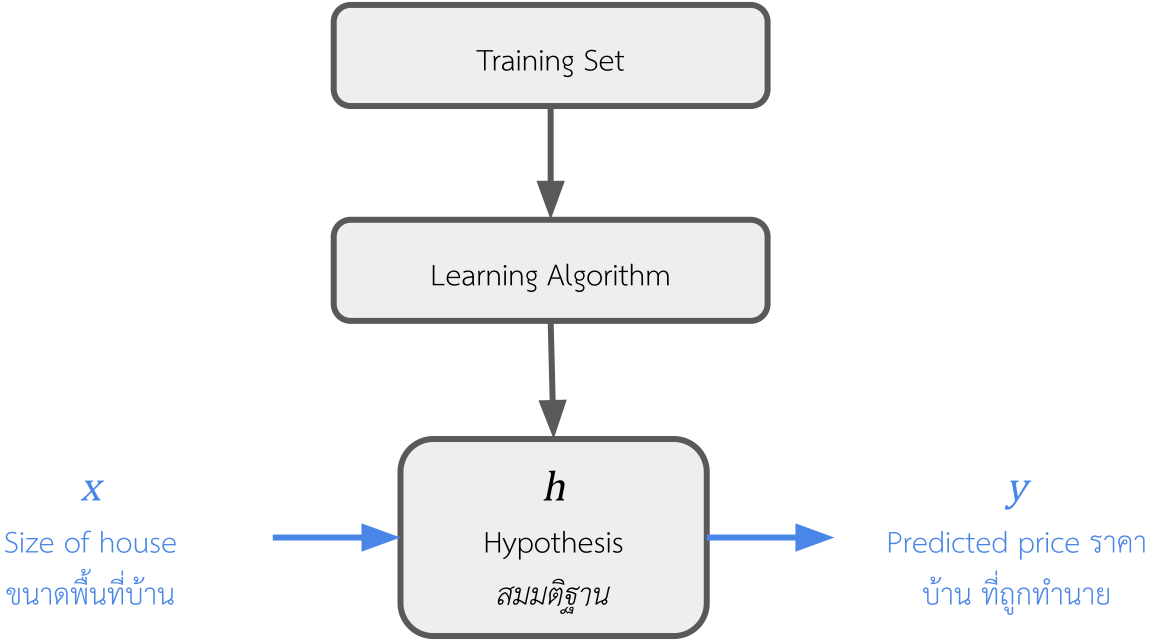


Image Credit: Andrew Ng's Machine Learning Course, Coursera

อธิบายปัญหา Supervised Learning ด้วยคณิตศาสตร์

เขียนอธิบาย $h(x)$ อย่างไร?

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

เขียนย่อเป็น $h(x)$

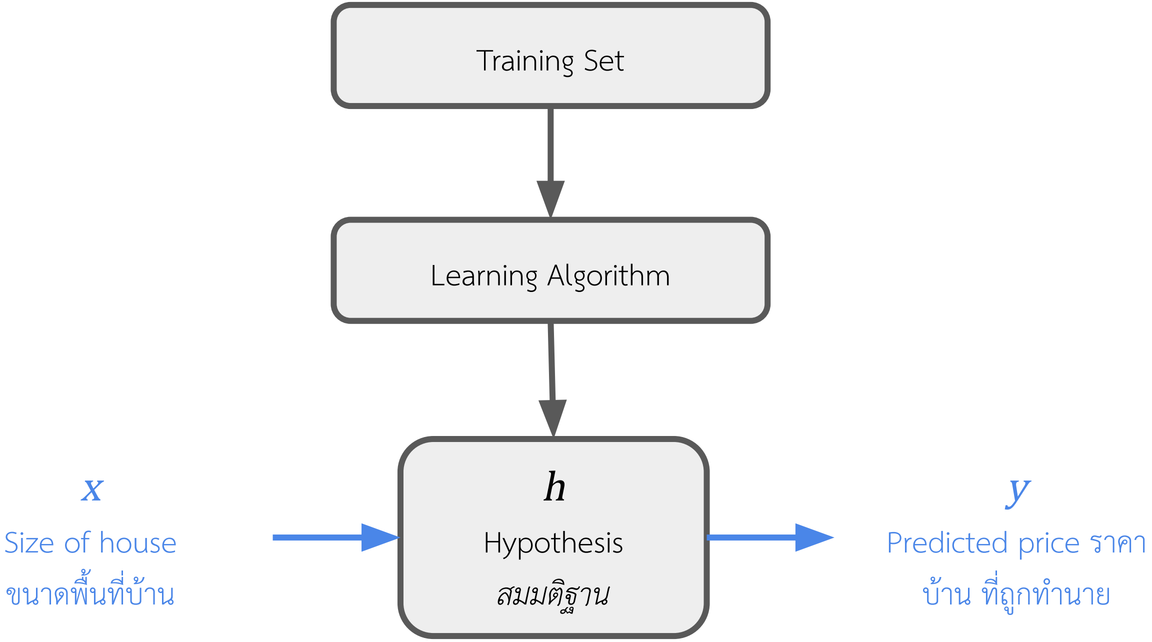


Image Credit: Andrew Ng's Machine Learning Course, Coursera

อธิบายปัญหา Supervised Learning ด้วยคณิตศาสตร์

เขียนอธิบาย $h(x)$ อย่างไร?

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

เขียนย่อเป็น $h(x)$

Linear regression ที่มี 1 ตัวแปร (หรือ univariate linear regression)

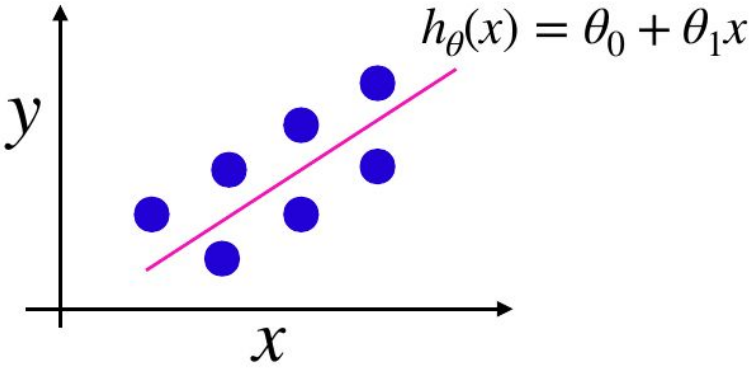


Image Credit: Andrew Ng's Machine Learning Course, Coursera

Recap: สรุป

- ถ้ามีข้อมูล training set
  เราอยากหาฟังก์ชั่น $h : X \rightarrow Y$
  ที่ทำให้ $h(x)$ ทำนายค่า y ได้ดี

- คำถาม คือ เราจะบอกได้ยังไงว่า h ที่เราหามา เป็นตัวทำนายที่ดี

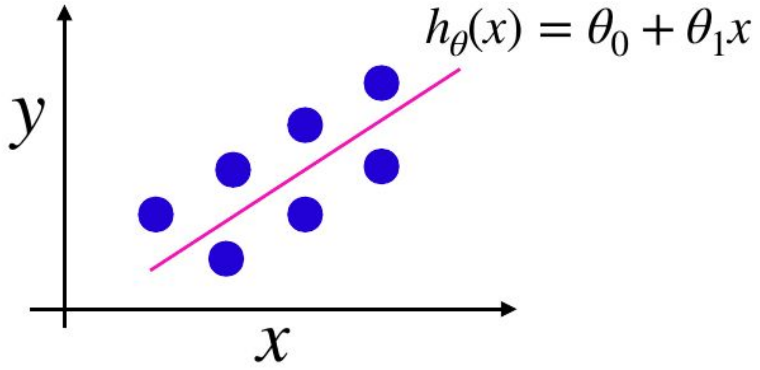


Image Credit: *Andrew Ng's Machine Learning Course, Coursera*

ประเภทของปัญหาการเรียนรู้ (Types of learning problems)

ประเภทของปัญหาการเรียนรู้	**Target variable และตัวอย่างปัญหา** (Target variable = ตัวแปรที่เราพยายามทำนาย)
ปัญหา Regression (การถดถอย)	มีค่าต่อเนื่องกัน (continuous) เช่น การทำนายราคาบ้าน (housing price)
ปัญหา Classification (การจำแนกประเภท)	ค่าที่ไม่ต่อเนื่อง (discrete values) ที่มีจำนวนจำกัด เช่น ถ้ารู้ขนาดของพื้นที่อยู่อาศัย (living area) ให้ทำนายว่าที่พักเป็นบ้านหรืออพาร์ทเมนท์

2.2 Cost Function ของ Linear Regression ที่มี 1 ตัวแปร

Krittameth Teachasrisaksakul

ชุดข้อมูล Training Set

ขนาด หรือ พื้นที่ (ตร.ฟุต) (x)	**ราคาบ้าน (ดอลล่าร์) x 1,000 (y)**
2104	460
1416	232
1534	315
852	178
...	...

$m = 47$

Hypothesis:

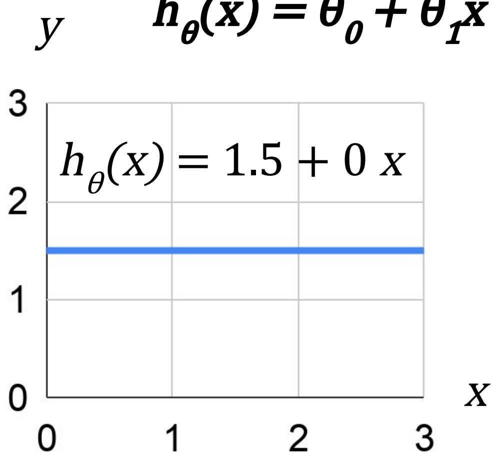
$$h_\theta(x) = \theta_0 + \theta_1 x$$

θ_i = parameters หรือ weights

เลือก θ_i อย่างไร?

กราฟของ Hypothesis จะเป็นยังไง ถ้าเปลี่ยนค่า parameters: θ_0, θ_1

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

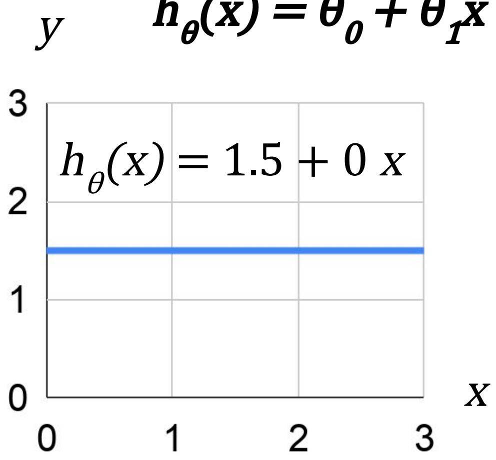


$\theta_0 = 1.5$

$\theta_1 = 0$

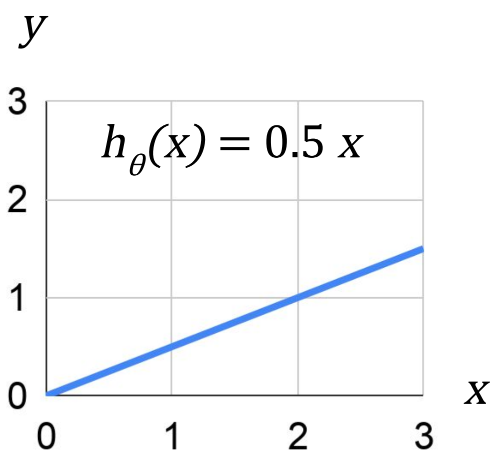
กราฟของ Hypothesis จะเป็นยังไง ถ้าเปลี่ยนค่า parameters: θ_0, θ_1

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



$\theta_0 = 1.5$

$\theta_1 = 0$

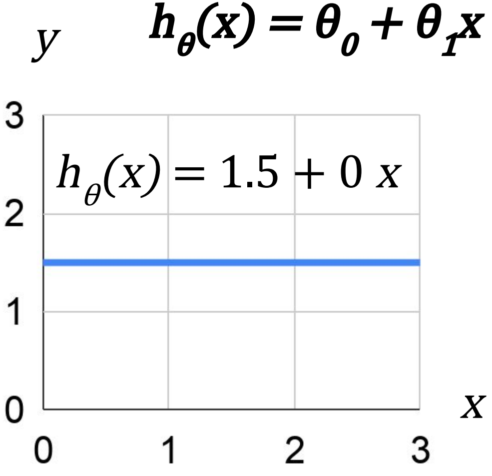


$\theta_0 = 0$

$\theta_1 = 0.5$

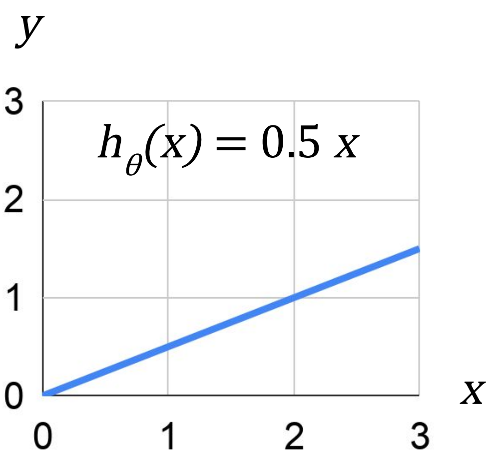
กราฟของ Hypothesis จะเป็นยังไง ถ้าเปลี่ยนค่า parameters: θ_0, θ_1

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



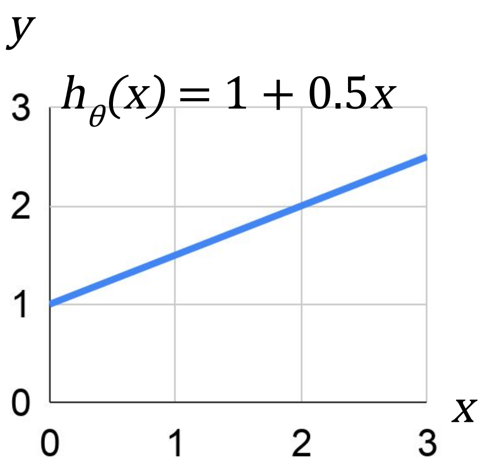
$\theta_0 = 1.5$

$\theta_1 = 0$



$\theta_0 = 0$

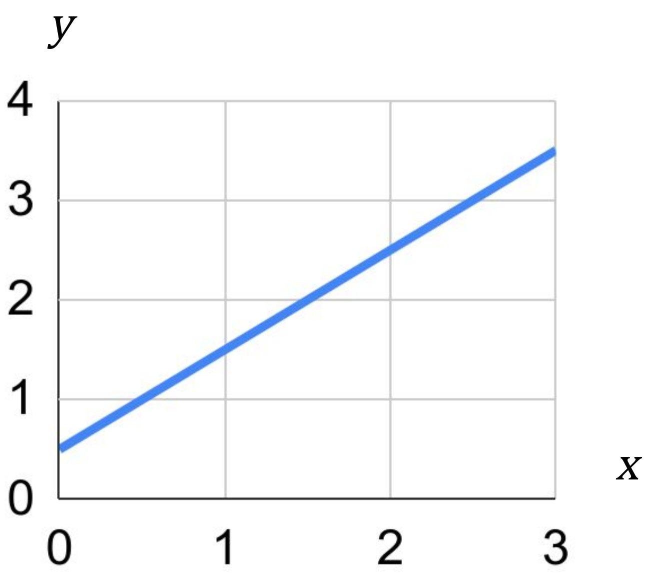
$\theta_1 = 0.5$



$\theta_0 = 1$

$\theta_1 = 0.5$

คำถาม: จาก plot ด้านล่าง ค่าของ $\boldsymbol{\theta_0}, \boldsymbol{\theta_1}$ เป็นเท่าไร



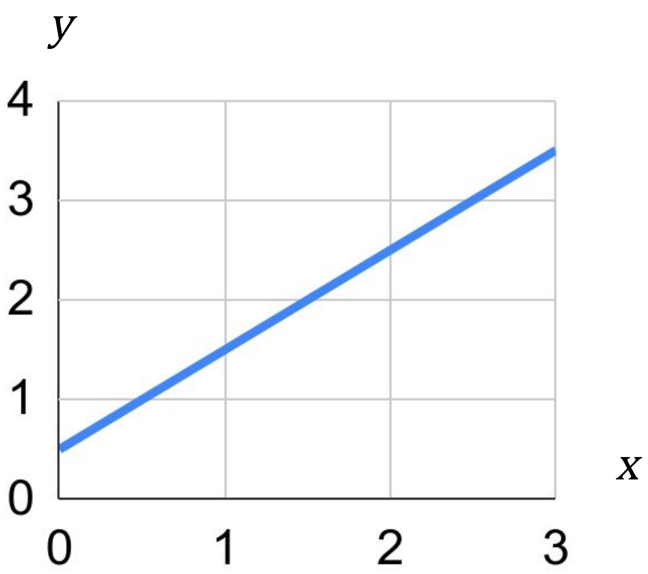
(i) $\theta_0 = 0, \quad \theta_1 = 1$

(ii) $\theta_0 = 0.5, \ \theta_1 = 1$

(iii) $\theta_0 = 1, \quad \theta_1 = 0.5$

(iv) $\theta_0 = 1, \quad \theta_1 = 1$

คำถาม: จาก plot ด้านล่าง ค่าของ $\boldsymbol{\theta_0}, \boldsymbol{\theta_1}$ เป็นเท่าไร



(i) $\theta_0 = 0, \quad \theta_1 = 1$

(ii) $\theta_0 = 0.5, \ \theta_1 = 1$

(iii) $\theta_0 = 1, \quad \theta_1 = 0.5$

(iv) $\theta_0 = 1, \quad \theta_1 = 1$

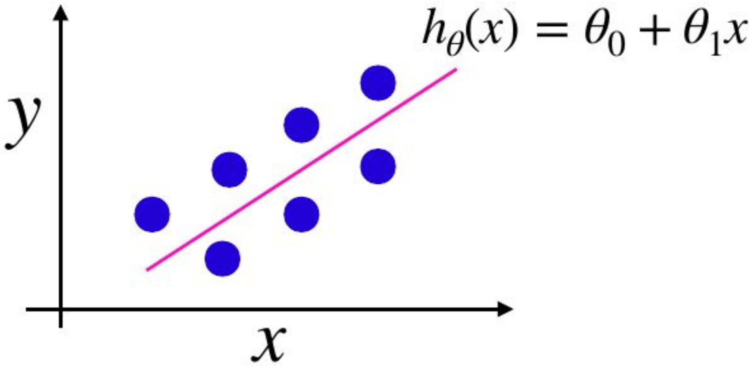
Cost Function

แนวคิด: เพื่อหาฟังก์ชั่น h ที่ดี $\rightarrow$ เราต้องหา θ_i ที่ดี

ในการทำ linear regression เราต้องมี cost function ที่

- ให้ **cost เยอะ**กับ ค่าจากการทำนาย $\boldsymbol{h(x)}$ **ที่แย่**
- ให้ **cost น้อย**กับ ค่าจากการทำนาย $\boldsymbol{h(x)}$ **ที่ดี**

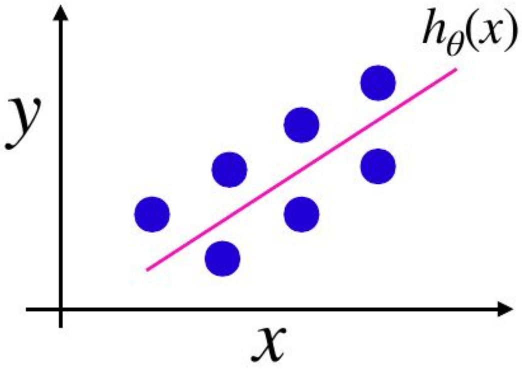
Cost Function



แนวคิด: เลือกค่า θ_0 และ θ_1 เพื่อทำให้ $h_\theta(x)$ มีค่าใกล้กับ y
เมื่อเรามีข้อมูลตัวอย่าง training example (x, y)

Cost Function

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



แนวคิด: เลือกค่า θ_0 และ θ_1 เพื่อทำให้ $h_\theta(x)$ มีค่าใกล้กับ y เมื่อใช้ตัวอย่าง (x, y) จากชุดข้อมูล training

$$\min_{\theta_0,\theta_1}(h_\theta(x^{(i)}) - y^{(i)})$$

$$\min_{\theta_0,\theta_1}(h_\theta(x^{(i)}) - y^{(i)})^2$$

$$\min_{\theta_0,\theta_1}[\boxed{\Sigma_{i=1}^{m}}(h_\theta(x^{(i)}) - y^{(i)})^2]$$

$$\min_{\theta_0,\theta_1}[\boxed{\frac{1}{2m}}\Sigma_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2]$$

Objective function ของ linear regression ที่มี 1 ตัวแปร

Source: Teeradaj Racharak; AI Practical Development Bootcamp

Cost Function

Cost function ของ Linear Regression ที่มี 1 ตัวแปร เป็น ค่าเฉลี่ย ของ ผลต่างระหว่าง $h_\theta(x_i)$ และ y_i

ใช้ cost function วัดความแม่นยำของ hypothesis function

เป็น squared error function (ฟังก์ชันความผิดพลาดยกกำลังสอง)

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

Cost Function

Cost function ของ Linear Regression ที่มี 1 ตัวแปร เป็น ค่าเฉลี่ยของ ผลต่างระหว่าง $h_\theta(x_i)$ และ y_i

เป็น squared error function (ฟังก์ชันความผิดพลาดยกกำลังสอง)

ใช้ cost function วัดความแม่นยำของ hypothesis function

จำนวนตัวอย่างจากชุดข้อมูล training

ค่าที่ถูกทำนาย

(predicted value)

= ผลทำนายจาก hypothesis ที่ใช้ input จาก X มาทำนาย

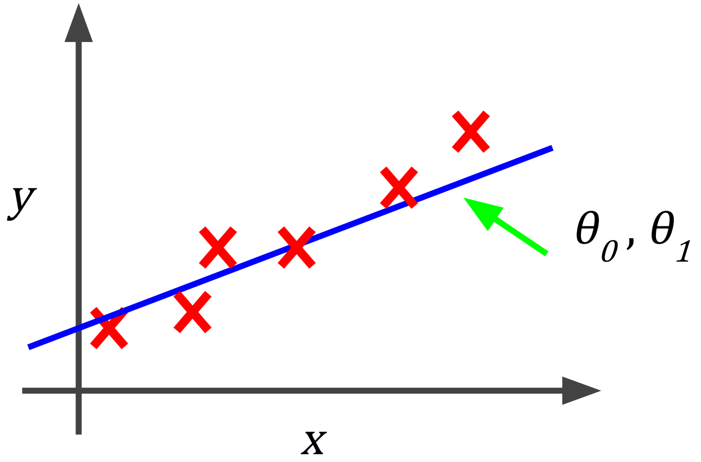
output จริง (y)

(actual value)

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

Cost Function: แนวคิด

แนวคิด: เลือก θ_0, θ_1 ให้ได้ $h_\theta(x)$ ที่ค่าใกล้กับ y เมื่อใช้ตัวอย่าง (x, y) จากชุดข้อมูล training



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

$$J(\theta_0, \theta_1) = \boxed{\frac{1}{2m}} \sum_{i=1}^{m} (\hat{y}_i - y_i)^2 = \frac{1}{2m} \sum_{i=1}^{m} (\boxed{h_\theta(x_i)} - y_i)^2$$

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

ค่าเฉลี่ยนี้ถูกหาร 2 เพื่อให้สะดวก ตอนคำนวณ Gradient Descent เพราะ ค่าอนุพันธ์ (derivative) ของ ฟังก์ชั่นกำลังสองจะตัดกับ ½ พอดี

Recap: สรุป

- Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$
- Parameters: θ_0, θ_1
- Cost function

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

- เป้าหมาย (Optimization objective)

$$\min_{\theta_0,\theta_1} J(\theta_0, \theta_1)$$

ในตัวอย่างนี้ function h เป็น**ฟังก์ชันเส้นตรง** (linear function)

θ_0, θ_1 เป็น parameters ของ model หรือ hypothesis function h

2. Linear Regression ที่มี 1 ตัวแปร

2.3 ทำให้ Cost Function น้อยที่สุด (minimize) ยังไง

Krittameth Teachasrisaksakul

ทำให้สมการง่ายขึ้น

- Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$ $\longrightarrow$ $h_\theta(x) = \theta_1 x$
- Parameters: θ_0, θ_1 $\longrightarrow$ θ_1
- Cost function

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

- เป้าหมาย (Optimization objective)

$$\min_{\theta_0,\theta_1} J(\theta_0, \theta_1) \longrightarrow \min_{\theta_1} J(\theta_1)$$

Cost function

ถ้ามองเป็นภาพ ชุดข้อมูล training กระจายตัวอยู่บน ระนาบ x-y

เราพยายามสร้างเส้นตรง $h_\theta(x)$ ที่ลากผ่านจุดข้อมูลที่กระจายตัวเหล่านี้

เป้าหมาย: หาเส้นที่ดีที่สุดที่เป็นไปได้ **ที่ทำให้** ค่าเฉลี่ยของ ค่ายกกำลังสองของ ระยะทางแนวตั้ง จากเส้นตรง ถึงจุดข้อมูลทั้งหมด หรือ $\boldsymbol{J(\theta_0, \theta_1)}$ **น้อยที่สุด**

สถานการณ์ในอุดมคติ (ที่อยากให้เป็น)

เส้นตรงลากผ่านทุกๆจุดในชุดข้อมูล training

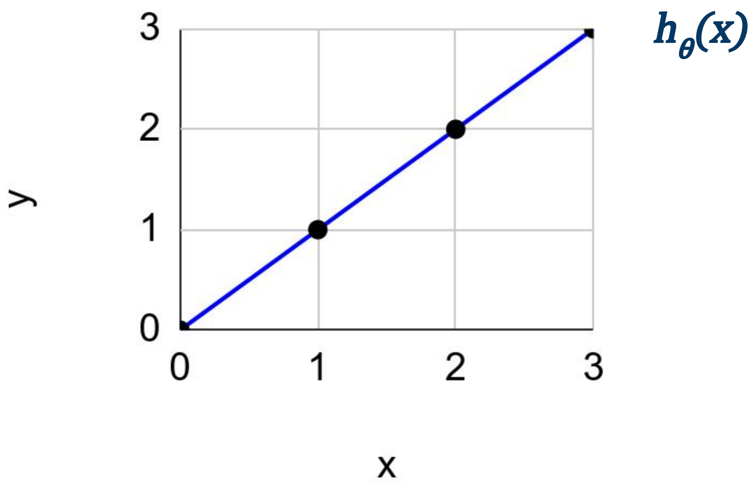
$$J(\theta_0, \theta_1) = 0$$

หรือ cost function มีค่าเป็น 0

Hypothesis vs. Cost : $\theta_1 = 1$

$h_\theta(x)$

(ถ้า θ_1 คงที่ จะเป็นฟังก์ชั่นของ x)



ชุดข้อมูล (Data set):

$\{(0,0), (1,1), (2,2), (3,3)\}$

สถานการณ์ในอุดมคติ (ที่อยากให้เป็น) : Model หรือ (ในกรณีนี้ คือ เส้นตรง) ที่มี $\theta_1 = 1, \theta_0 = 0$

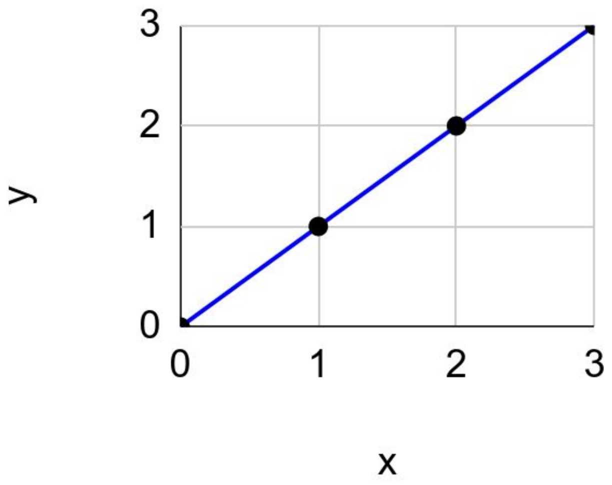
เมื่อ $\theta_1 = 1$ (Slope หรือ ความชัน = 1)

เส้นตรงจะผ่านทุกๆจุดข้อมูลใน model

Hypothesis vs. Cost : $\theta_1 = 1$

$h_\theta(x)$

(ถ้า θ_1 คงที่ จะเป็นฟังก์ชั่นของ x)



$h_\theta(x)$

$J(\theta_1)$

(ฟังก์ชั่นของ parameter θ_1)

แทนค่า $\theta_1 = 1$ และ $h_\theta(x_i) = \theta_1 x$ ใน $J(\theta_1)$

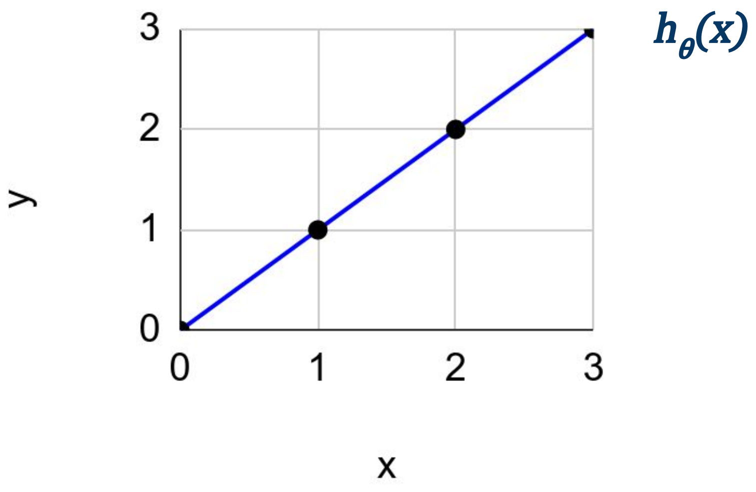
$$\because J(\theta_1) = \frac{1}{2m}[\Sigma_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2]$$

$$J(1) = (1 / 2\text{m})(0^2+0^2+0^2) = 0$$

Hypothesis vs. Cost : $\theta_1 = 1$

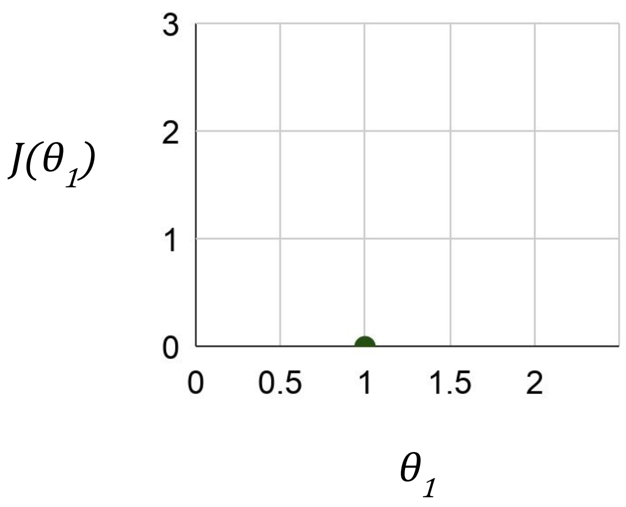
$h_\theta(x)$

(ถ้า θ_1 คงที่ จะเป็นฟังก์ชั่นของ x)



$J(\theta_1)$

(ฟังก์ชั่นของ parameter θ_1)



Question

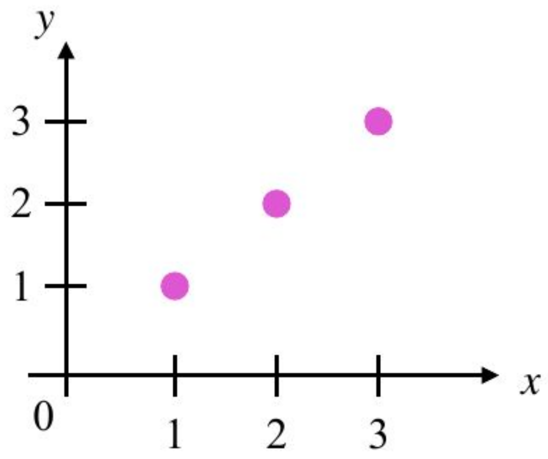
ถ้ามีชุดข้อมูล training set ที่มีจำนวนตัวอย่างข้อมูล $m = 3$ และ hypothesis คือ $h_\theta(x_i) = \theta_1 x$ ซึ่งมี parameter 1 ตัว คือ θ_1 Cost function $J(\theta_1)$ คือ

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

$J(0)$ เท่ากับเท่าไร

(a) 0 (b) $1/6$

(c) 1 (d) $14/6$



Question

ถ้ามีชุดข้อมูล training set ที่มีจำนวนตัวอย่างข้อมูล $m = 3$ และ hypothesis คือ $h_\theta(x_i) = \theta_1 x$ ซึ่งมี parameter 1 ตัว คือ θ_1Cost function $J(\theta_1)$ คือ

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

$J(0)$ เท่ากับเท่าไร

(a) 0 (b) $1/6$

(c) 1 (d) $14/6$

$$\because \theta_1 = 0$$

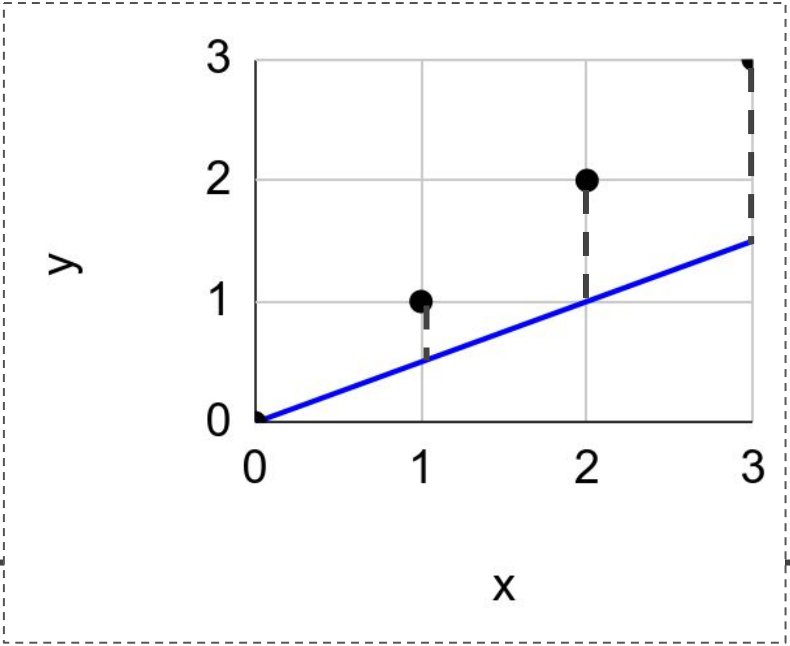
$$\therefore J(\theta_1) = \frac{1}{2 \times 3}[(0 \times 1 - 1)^2 + (0 \times 2 - 2)^2 + (0 \times 3 - 3)^2]$$

$$= \frac{1}{6}[1 + 4 + 9]$$

Hypothesis vs. Cost : $\theta_1 = 0.5$

$h_\theta(x)$

(ถ้า θ_1 คงที่ จะเป็นฟังก์ชั่นของ x)



ระยะทางแนวตั้งจากเส้นตรง (โมเดล) ของเรา ถึง จุดข้อมูล จะเพิ่มขึ้น

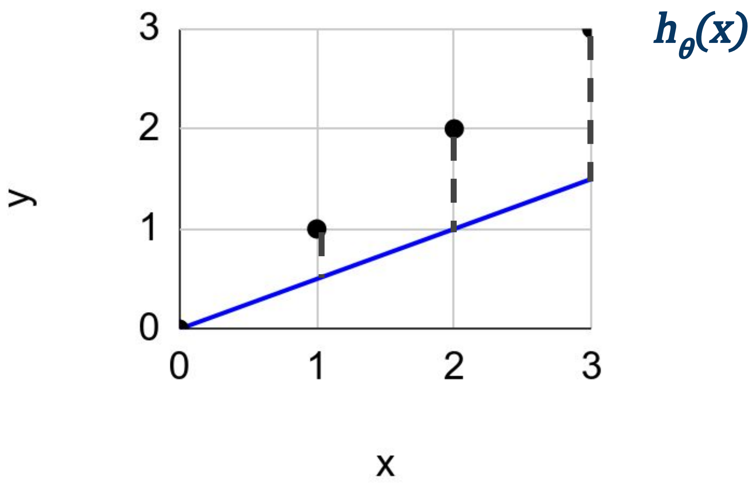
ซ้าย: $h_\theta(x)$ ที่ค่า x ต่างๆ เมื่อ $\theta_1 = 0.5$

Cost function $J(\theta_1)$ เมื่อ $\theta_1 = 0.5$ เท่ากับเท่าไร?

Hypothesis vs. Cost : $\theta_1 = 0.5$

$h_\theta(x)$

(ถ้า θ_1 คงที่ จะเป็นฟังก์ชั่นของ x)



ระยะทางแนวตั้งจากเส้นตรง (โมเดล) ของเรา ถึง จุดข้อมูล จะเพิ่มขึ้น

ซ้าย: $h_\theta(x)$ ที่ค่า x ต่างๆ เมื่อ $\theta_1 = 0.5$

Cost function $J(\theta_1)$ เมื่อ $\theta_1 = 0.5$ เท่ากับเท่าไร?

แทนค่า $\theta_1 = 0.5$ และ $h_\theta(x_i) = \theta_1 x$ ใน $J(\theta_1)$

$$J(\theta_1) = 1/(2x3)\ [(0.5\text{-}1)^2 + (1\text{-}2)^2 + (1.5\text{-}3)^2\ ]$$

$$\rightarrow \quad J(0.5) \approx 0.58$$

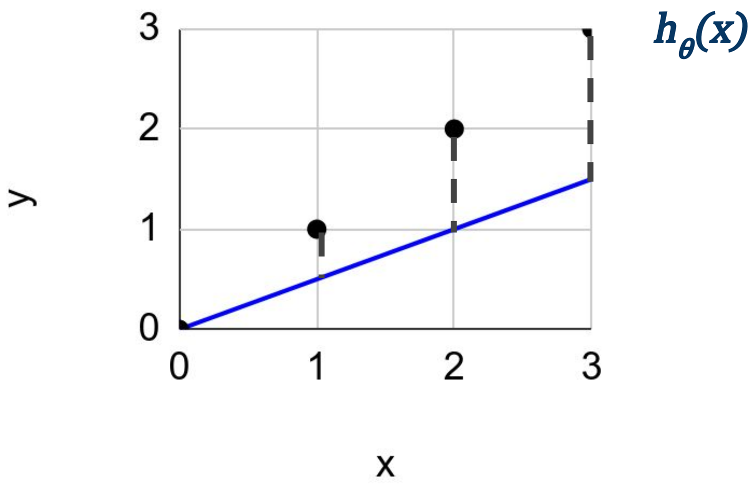
Cost function $J(\theta_0, \theta_1) \approx 0.58$

Hypothesis vs. Cost : $\theta_1 = 0.5$

plot cost ของค่า θ_1 ต่างๆ จะได้กราฟนี้

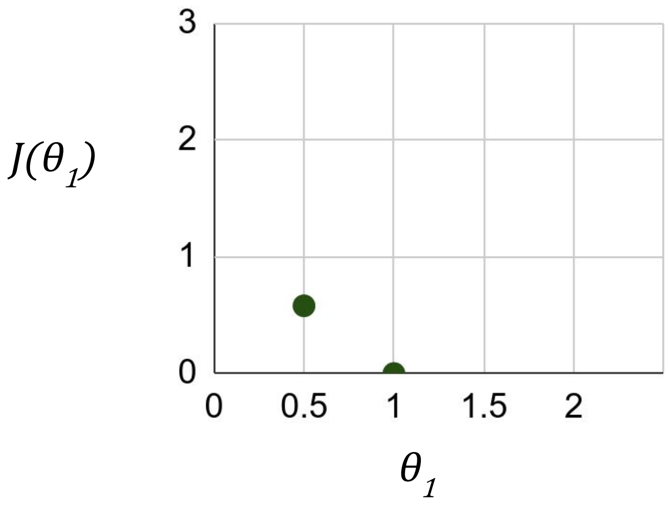
$h_\theta(x)$

(ถ้า θ_1 คงที่ จะเป็นฟังก์ชั่นของ x)

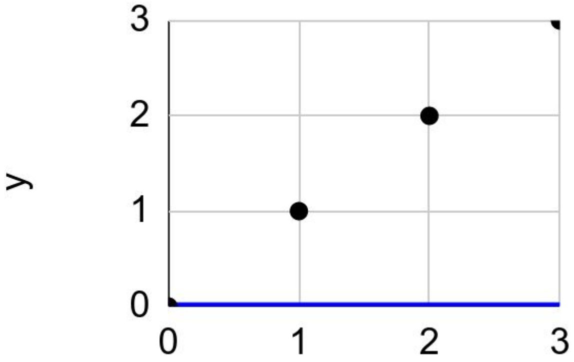


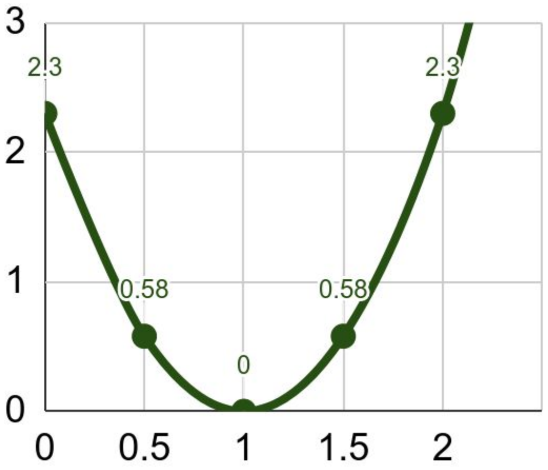
$J(\theta_1)$

(ฟังก์ชั่นของ parameter θ_1)



Hypothesis vs. Cost : สรุป





$J(\theta_1)$

θ_1

แทนค่า $\theta_1 = 0$ และ $h_\theta(x_i) = \theta_1 x$ ใน $J(\theta_1)$

$J(\theta_1) = (1 / 2\text{x}3)[(0\text{-}1)^2+(0\text{-}2)^2+(0\text{-}3)^2]$

$\rightarrow \quad J(0) \approx 2.3$

เป้าหมาย : ทำให้ cost function น้อยที่สุด

$$\min J(\theta_1)$$

ในกรณีนี้ θ_1 = 1 เป็น global minimum

2. Linear Regression ที่มี 1 ตัวแปร

2.4 Cost Function:
ความเข้าใจพื้นฐาน 2

Krittameth Teachasrisaksakul

ใช้สมการชุดเริ่มต้น

- Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$
- Parameters: θ_0, θ_1
- Cost function

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

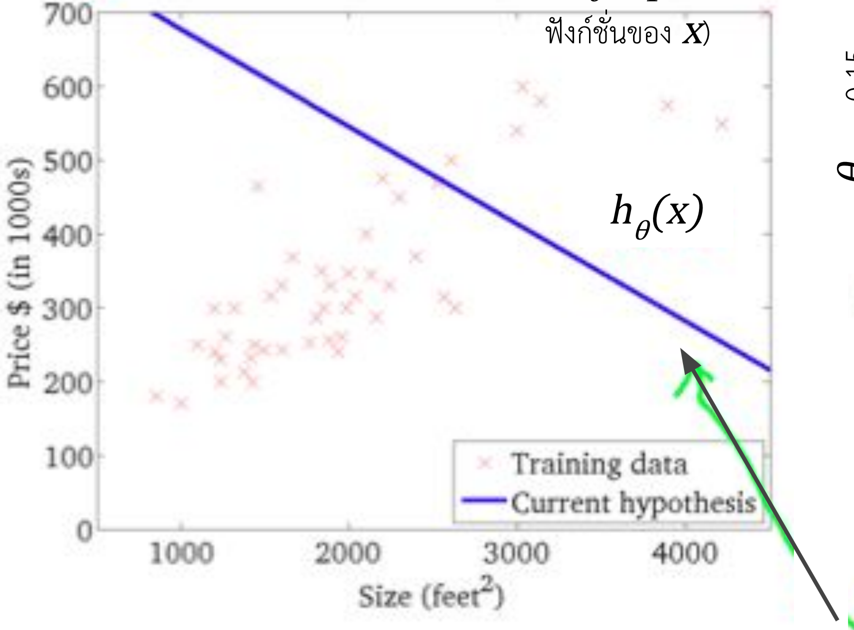
- เป้าหมาย (Optimization objective)

$$\min_{\theta_0,\theta_1} J(\theta_0, \theta_1)$$

Contour Plot

$h_\theta(x)$

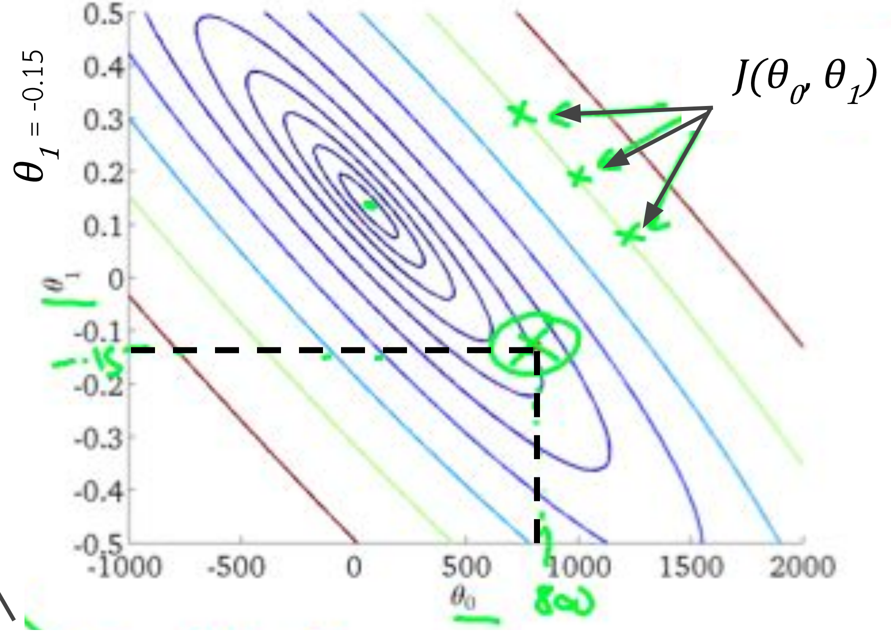
(สำหรับค่า θ_0, θ_1 ที่คงที่ จะเป็นฟังก์ชั่นของ x)



θ_0 = 800, θ_1 = -0.15

$J(\theta_0, \theta_1)$

(ฟังก์ชั่นของ parameters θ_0, θ_1)



θ_0 = 800

Contour Plot : อธิบาย

contour plot คือ graph ที่มี contour line หลายเส้น

contour line ของฟังก์ชัน 2 ตัวแปร มีค่าคงที่ ที่ทุกๆจุดบนเส้นเดียวกัน

ตัวอย่าง: กราฟด้านขวา

ถ้าไล่ไปตามเส้นวงกลมเส้นหนึ่งที่เป็นสีเดียวกัน จะได้ค่าของ cost function ค่าเดิม

ตัวอย่าง: จุดสีเขียว 3 จุดที่อยู่บนเส้นสีเขียวจะมีค่า $J(\theta_0, \theta_1)$ เดียวกัน ดังนั้น เราจึงเจอจุดพวกนี้อยู่บนเส้นเดียวกัน

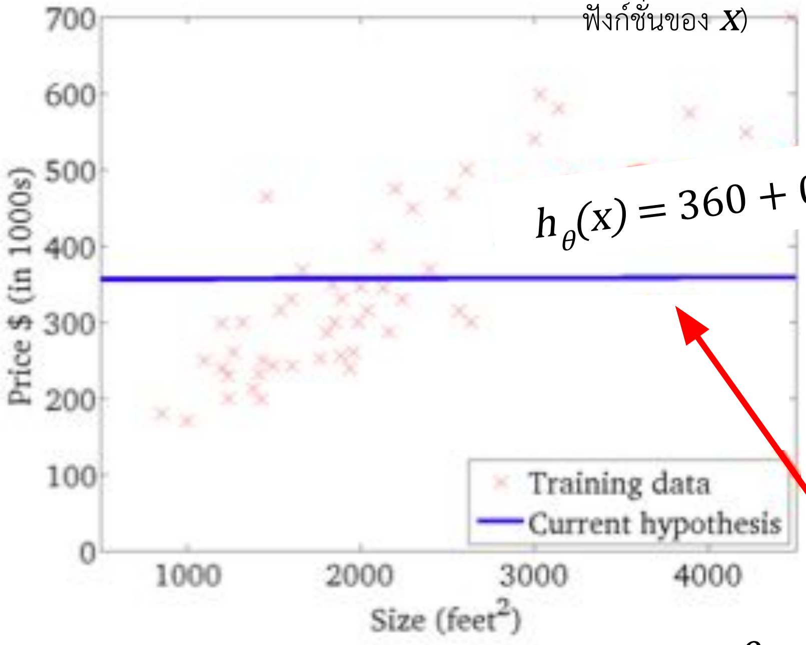
จุด $\boldsymbol{X}$ (ที่โดนวง) บอกค่าของ cost function ของกราฟด้านซ้าย เมื่อ θ_0 = 800 และ θ_1 = -0.15

Figure source: Andrew Ng, Machine Learning

Contour Plot

$h_\theta(x)$

(สำหรับค่า θ_0, θ_1 ที่คงที่ จะเป็นฟังก์ชั่นของ x)



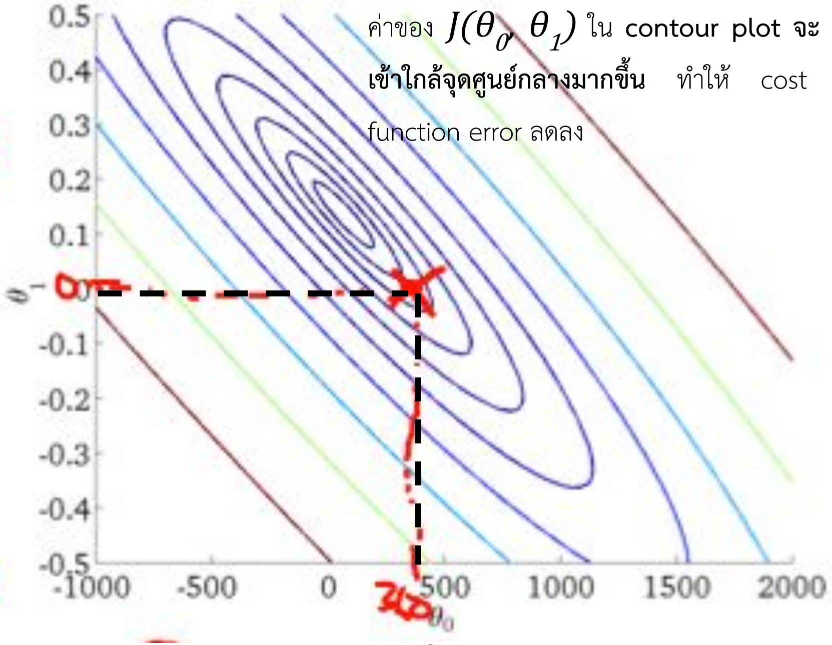
$h_\theta(x) = 360 + 0\theta_1$

θ_0 = 360, θ_1 = 0

$J(\theta_0, \theta_1)$

(ฟังก์ชั่นของ parameters θ_0, θ_1)

ค่าของ $J(\theta_0, \theta_1)$ ใน contour plot จะ **เข้าใกล้จุดศูนย์กลางมากขึ้น** ทำให้ cost function error ลดลง



θ_1 = 0

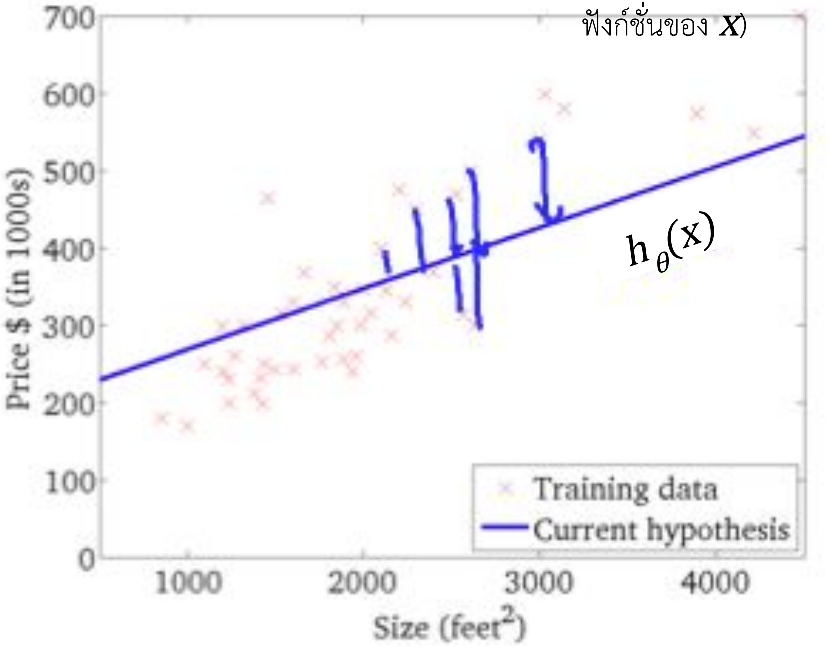
θ_0 = 360

Figure source: Andrew Ng, Machine Learning

Contour Plot

$h_\theta(x)$

(สำหรับค่า θ_0, θ_1 ที่คงที่ จะเป็นฟังก์ชั่นของ x)



$J(\theta_0, \theta_1)$

(ฟังก์ชั่นของ parameters θ_0, θ_1)

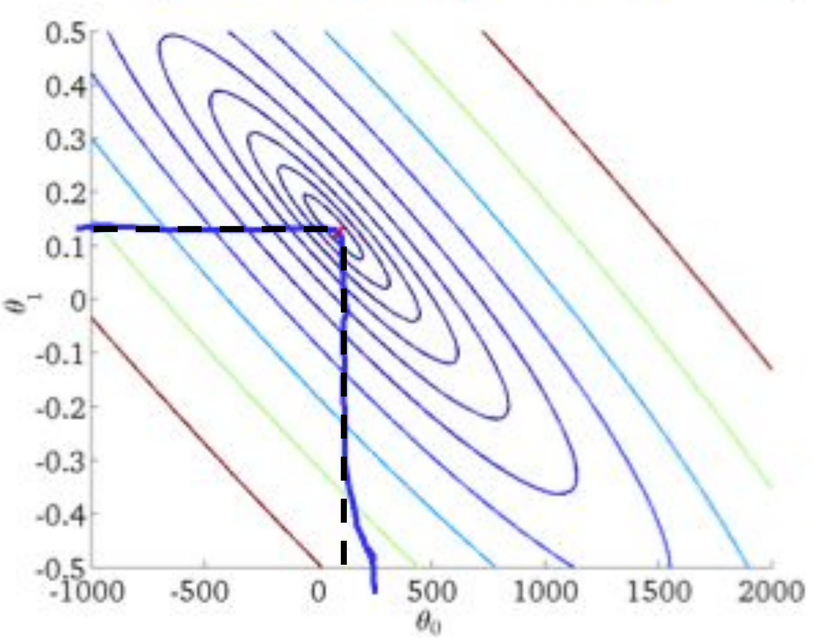


Figure source: Andrew Ng, Machine Learning

Contour Plot : อธิบาย

เลือก **hypothesis function ที่มีความชันเป็นบวกเล็กน้อย** จะได้เส้นที่เข้ากับข้อมูลได้มากขึ้น (better fit)

กราฟนี้ทำให้ cost function น้อยที่สุด เท่าที่จะเป็นไปได้

ผลของ θ_1 และ θ_0 มีแนวโน้มที่จะเป็น 0.12 และ 250 ⟶ plot ค่าเหล่านี้ลงในกราฟด้านขวา จะได้จุดที่อยู่ตรงจุดศูนย์กลางของวงกลมในสุด

2. Linear Regression ที่มี 1 ตัวแปร

2.5 Gradient Descent

algorithm สำหรับทำ Linear Regression ที่มี 1 ตัวแปร

Krittameth Teachasrisaksakul

T

เรามี

- hypothesis function
- วิธีวัดค่าว่าฟังก์ชันนี้ เข้ากับ (fit) ข้อมูลได้ดีแค่ไหน

เราต้องประมาณค่า parameter ใน hypothesis function

โดยใช้วิธี gradient descent

นึกภาพตาม: ทำกราฟของ hypothesis function โดยใช้ค่า θ_0 , θ_1 ของมัน

คือ ทำกราฟ cost function เป็น function ของค่าประมาณ (estimates) ของ parameter

ไม่ใช่ plot x, y

plot ช่วงค่าของ parameter (parameter range) ของ hypothesis function และ cost ที่ได้จากการเลือกค่าบางค่าของ parameters

ความหมายของ Gradient Descent

Gradient: ระดับความชันของกราฟ ที่จุดใดจุดหนึ่ง

Descent: การเคลื่อนต่ำลง หรือ การตกลงมา

นิยามแบบทางการ

Gradient ของ function $f(x)$ หรือ $\nabla_f(x)$

ที่หาค่า ณ จุดใดๆ x เป็นเวกเตอร์ที่มีคุณสมบัติ 2 อย่าง:

- **ทิศทาง** (direction) ของ
  คือ ทิศทางที่ $f(x)$ เพิ่มขึ้น อย่างเร็วที่สุ $\nabla_f(x)$
- **ขนาด** (magnitude) ของ
  คือ ความชันของ $f(x)$ ในทิศทางที่มีกา $\nabla_f(x)$ สูงที่สุด

Algorithm ทั่วไปของ Gradient Descent

เรามี

- Cost function: $J(\theta_0, \theta_1)$
  - วัดว่า hypothesis function เข้ากับข้อมูลได้ดีแค่ไหน
- เป้าหมาย:

$$\boxed{\min_{\theta_0,\theta_1} J(\theta_0, \theta_1)}$$

สรุป: ขั้นตอนคร่าวๆ

- เริ่มจากค่า θ_0, θ_1 บางค่า
- เปลี่ยน θ_0, θ_1 เรื่อยๆ เพื่อลด $J(\theta_0, \theta_1)$ จนกระทั่ง ได้ค่าที่น้อยที่สุด ถ้าเป็นไปได้

Algorithm ทั่วไปของ Gradient Descent:

$$\min_{\theta_0,\ldots,\theta_n} J(\theta_0, \ldots, \theta_n)$$

สรุป: ใช้ Gradient Descent

- หาค่า $\theta_0, \ldots, \theta_n$ หรือ ค่าประมาณของ parameter ใน hypothesis function
- ที่ทำให้ Cost function $J(\theta_0, \ldots, \theta_n)$ มีค่าน้อยที่สุด

Gradient Descent ทำอย่างไร?

step = การเคลื่อนที่ ที่เกิดขึ้นใน 1 iteration (การวนซ้ำ) ของ Gradient Descent

1. หา ทิศทาง ของ 1 step

เพราะเป้าหมายคือ ทำให้ cost function น้อยที่สุด ⟶ ในแต่ละ step เคลื่อนไป ใน**ทิศทางที่ cost function ลดลงเร็วที่สุด หรือ ชันที่สุด** (steepest descent)

- **ซึ่งก็คือ derivative (อนุพันธ์) ของ cost function**
- Derivative คือ ความชันของ tangential line (เส้นสัมผัส) ของ function

2. เลือก ขนาด ของ 1 step

โดยเลือก **learning rate α** (parameter อีกตัวหนึ่ง)

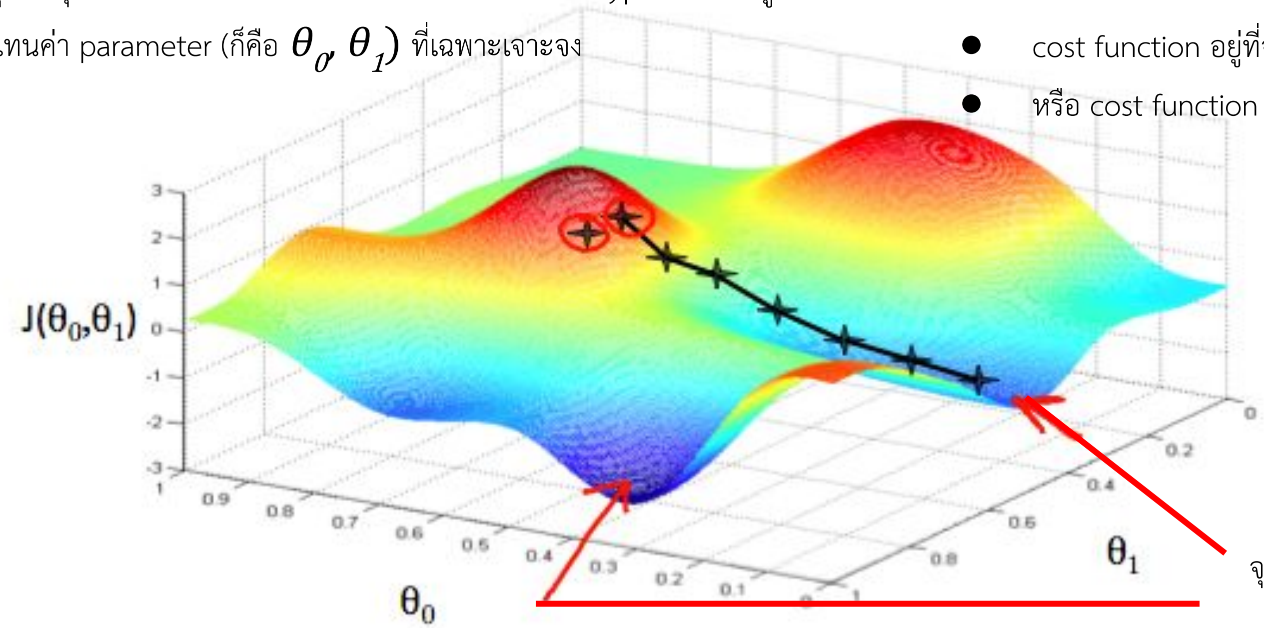
- ค่า α น้อยลง ⟶ step เล็กลง
- ค่า α มากขึ้น ⟶ step ใหญ่ขึ้น

Plot: cost function $J(\theta_0, \theta_1)$, θ_0, θ_1 บนแกน z, x, y

จุด 1 จุดในกราฟเป็นผลของ cost function ที่ได้จาก hypothesis ที่ถูกแทนค่า parameter (ก็คือ θ_0, θ_1) ที่เฉพาะเจาะจง

เรารู้ว่าทำสำเร็จ เมื่อ

- cost function อยู่ที่จุดต่ำที่สุดในกราฟ
- หรือ cost function มีค่า เป็นค่าน้อยที่สุด (minimum)



จุดที่ cost function $J(\theta_0, \theta_1)$ มีค่าน้อยที่สุด

Figure source: Andrew Ng, Machine Learning

ตัวอย่าง: Gradient Descent

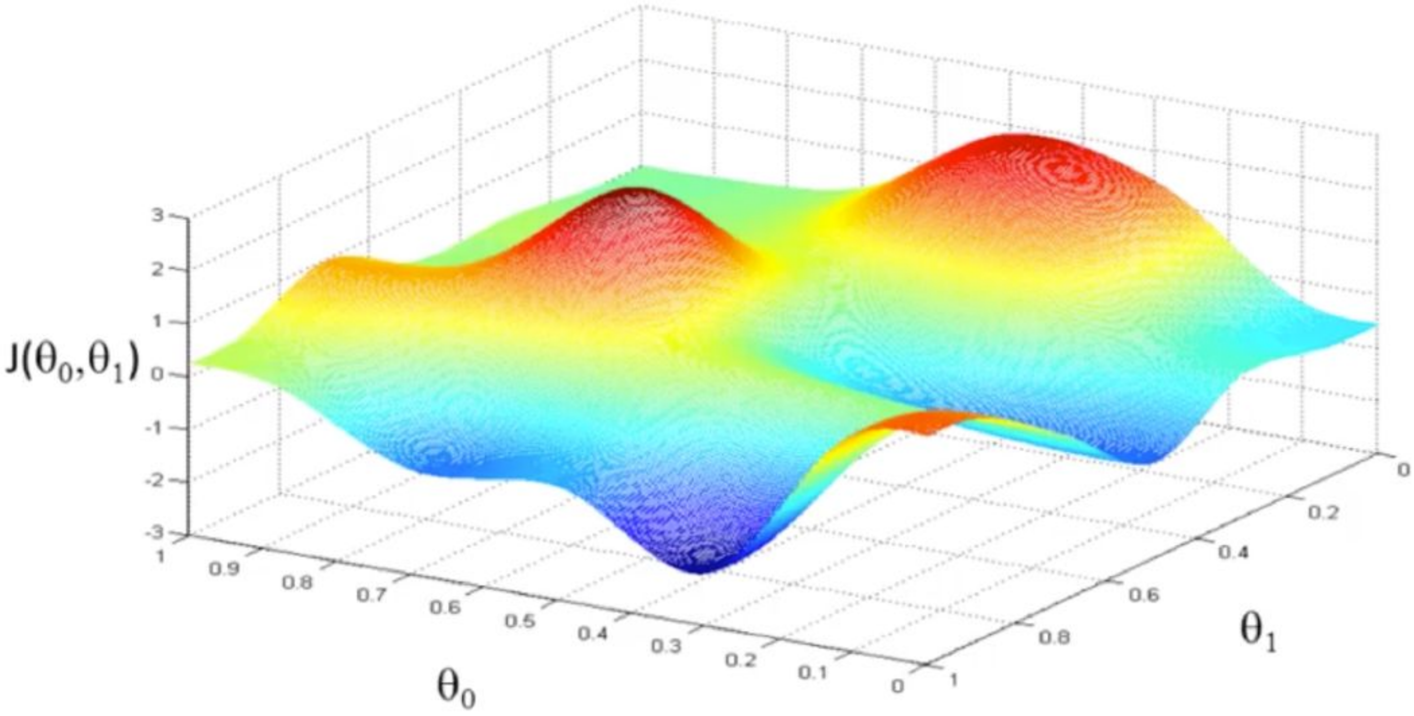


Figure source: Andrew Ng, Machine Learning

ตัวอย่าง: Gradient Descent

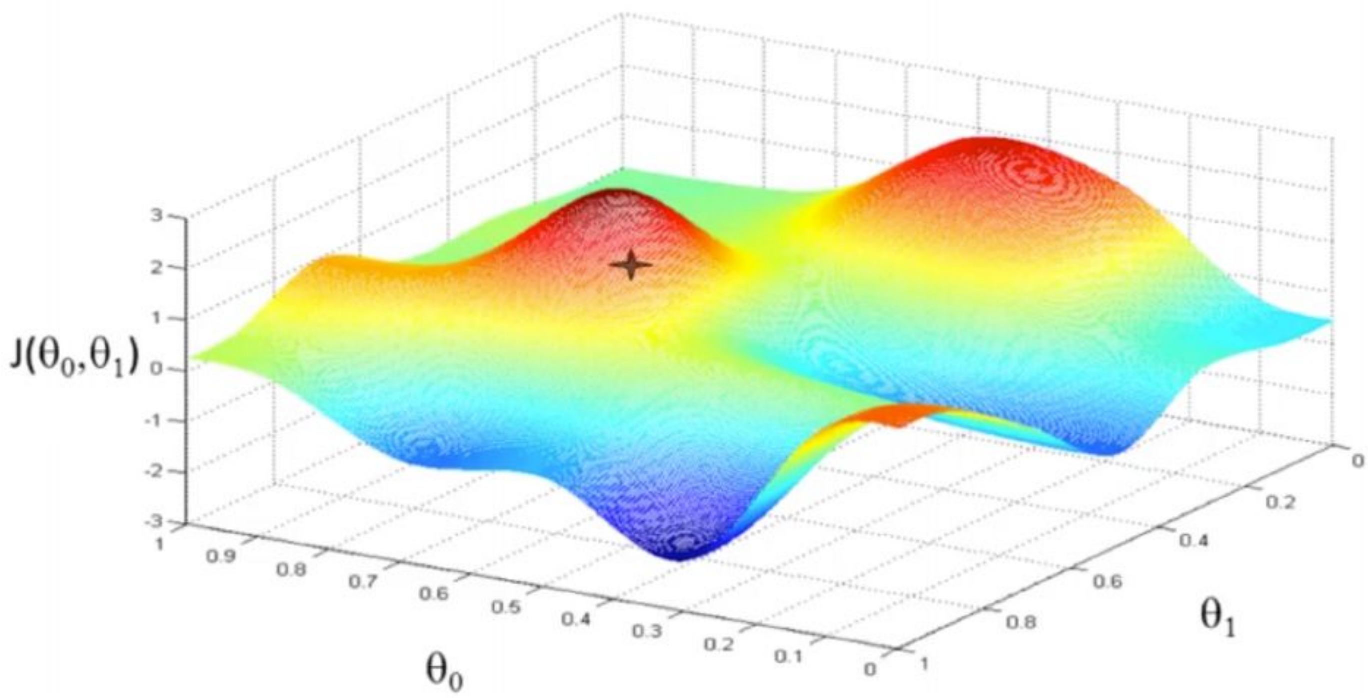


Figure source: Andrew Ng, Machine Learning

ตัวอย่าง: Gradient Descent

ระยะทางระหว่างดาวแต่ละอันในกราฟ คือ 1 step ที่ถูกกำหนดโดย parameter α หรือ learning rate

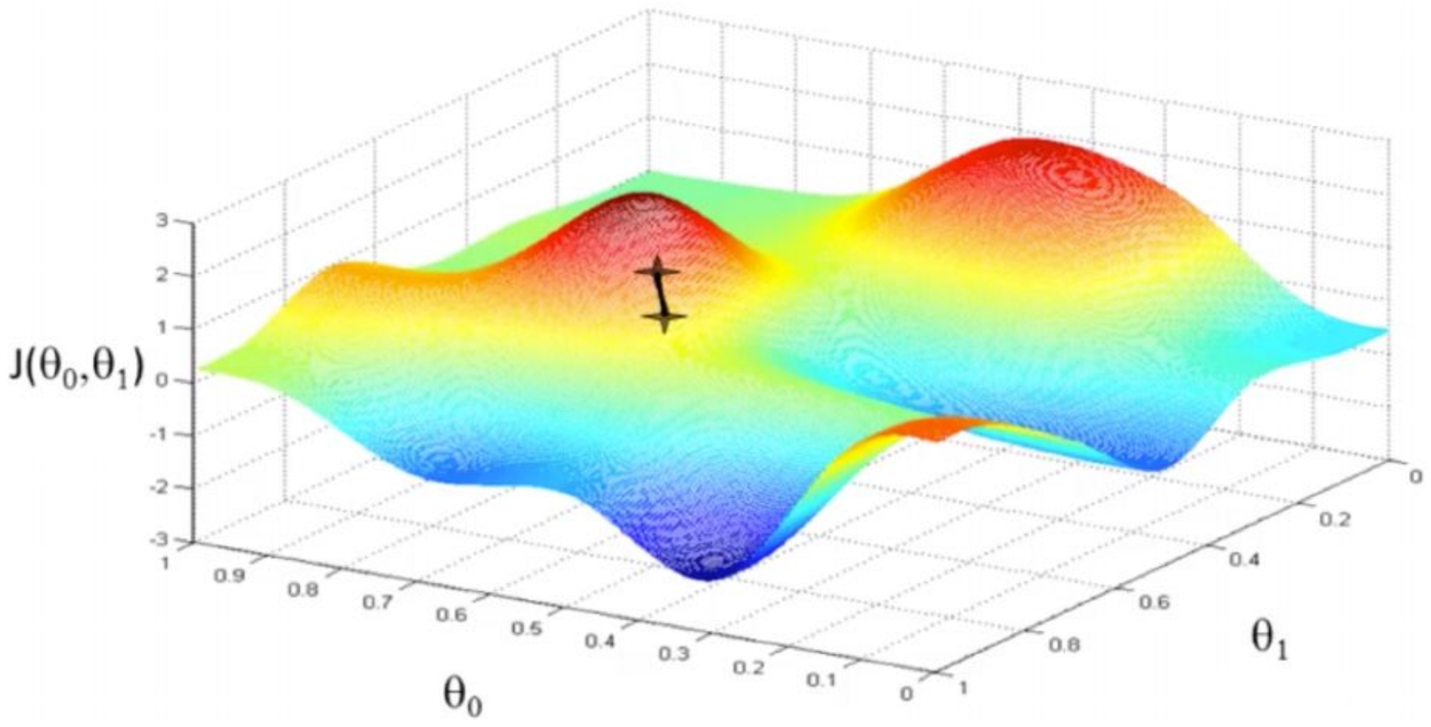


Figure source: Andrew Ng, Machine Learning

ตัวอย่าง: Gradient Descent

ทิศทางที่ step เคลื่อนไป ถูกกำหนดโดย partial derivative ของ $J(\theta_0, \theta_1)$

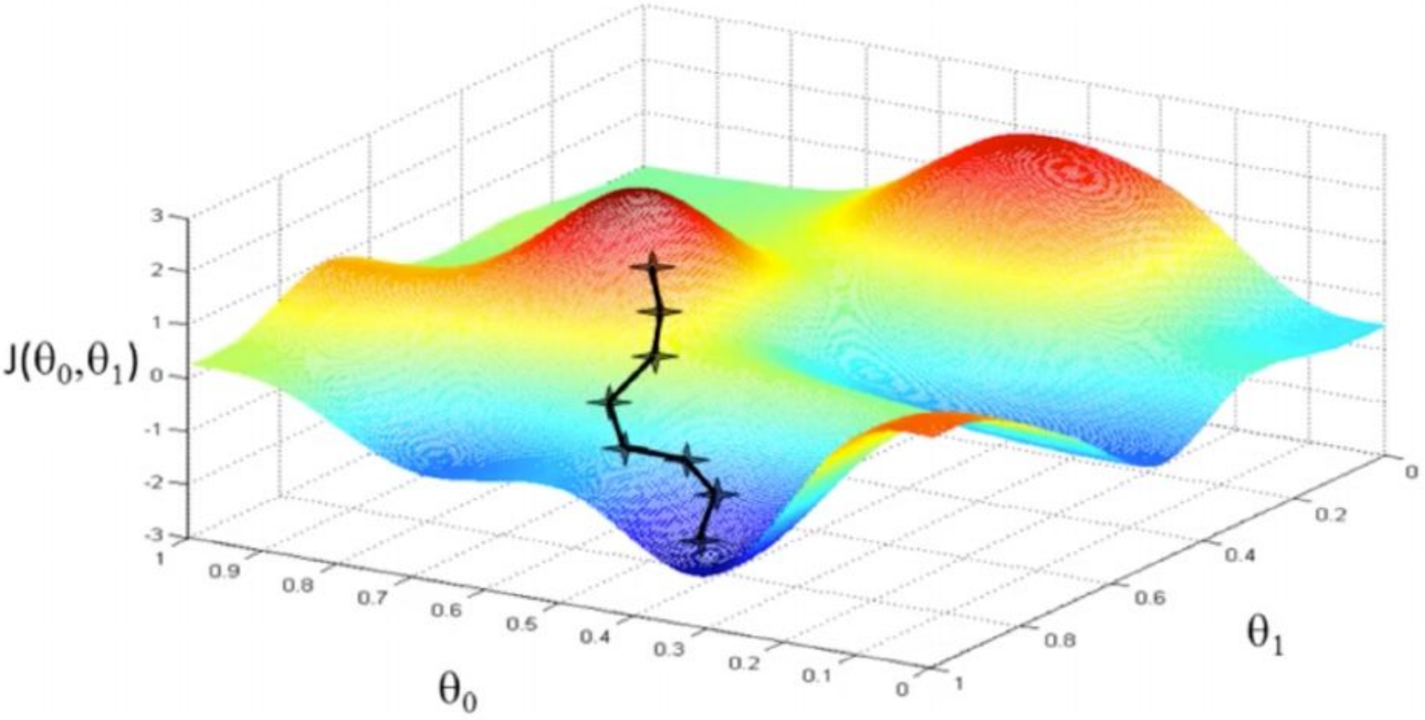


Figure source: Andrew Ng, Machine Learning

ตัวอย่าง: Gradient Descent

ผลที่ได้จากการเคลื่อนที่ ขึ้นอยู่กับว่า เริ่มที่จุดไหนบนกราฟ

กราฟแสดง จุดเริ่มต้นสองจุด (ที่ต่างกัน) ซึ่งทำให้ได้จุดจบสองจุดที่ต่างกัน

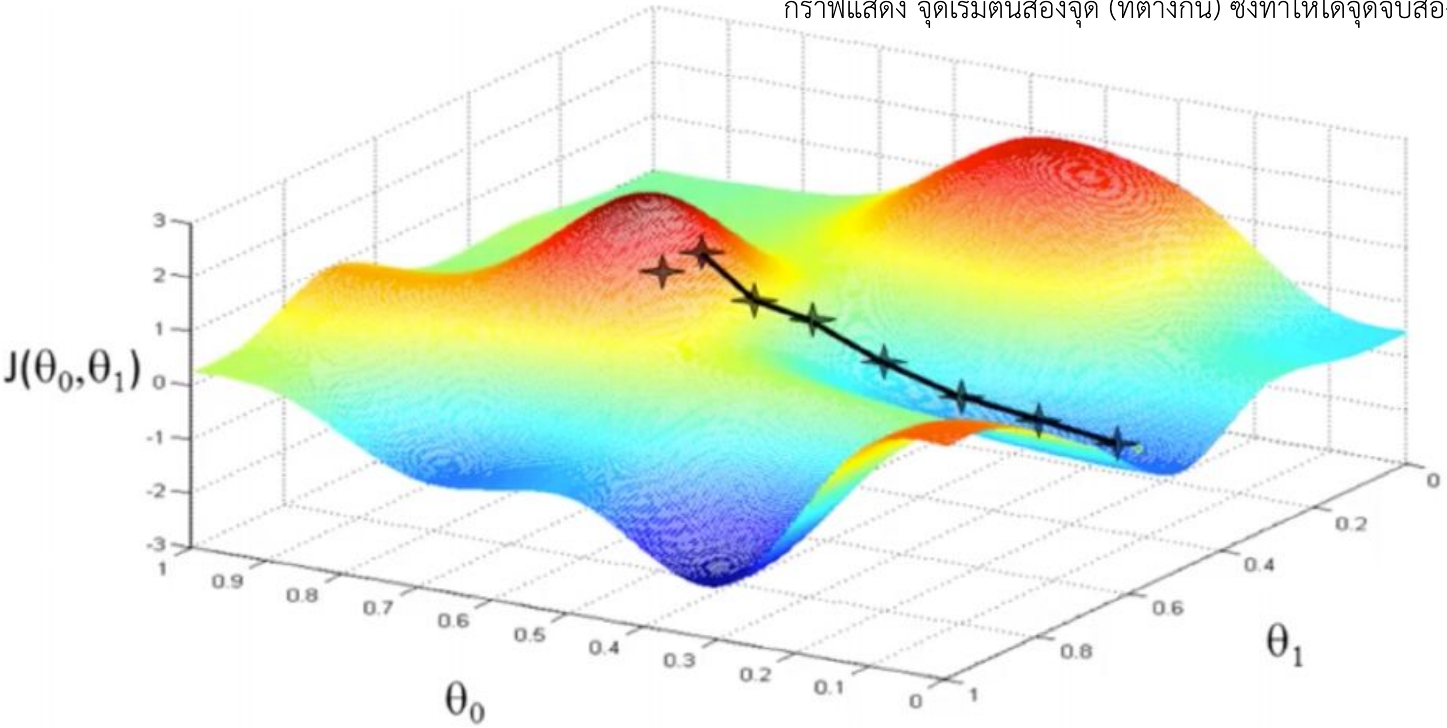


Figure source: Andrew Ng, Machine Learning

Gradient descent algorithm

repeat until convergence {

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

}

'learning rate' เป็น บวก (+) เสมอ

- เมื่อ $j = 0,1$ แทน เลข feature index
- ในแต่ละ iteration j : ควรปรับค่า parameter $\theta_1, \theta_2,$ ..., θ_n ไปพร้อมๆกัน
- ปรับค่า parameter ตัวใดตัวหนึ่ง ก่อนคำนวณอีกตัว ใน iteration ที่ $j^{(th)}$ อาจทำให้คำนวณผิด

วิธีที่ถูก: update parameter พร้อมๆกัน

$$t_1 := \theta_0 - \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$
$$t_2 := \theta_1 - \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$
$$\theta_0 := t_1$$
$$\theta_1 := t_2$$

วิธีที่ผิด

$$t_1 := \theta_0 - \frac{\partial J}{\partial \theta_0}(\theta_0, \theta_1)$$
$$\theta_0 := t_1$$
$$t_2 := \theta_1 - \boxed{\frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)}$$
$$\theta_1 := t_2$$

Source: Andrew Ng, Machine Learning

คำถาม

ถ้า $\theta_0 = 1, \theta_1 = 2$ และเราปรับค่า θ_0 และ θ_1 พร้อมๆกัน โดยใช้

$$\theta_j := \theta_j + \sqrt{\theta_0 \theta_1}$$ (สำหรับ $j = 0, j = 1$)

ค่าของ θ_0 และ θ_1 จะเป็นเท่าไร

(i) $\theta_0 = 1, \ \theta_1 = 2$

(ii) $\theta_0 := 1 + \sqrt{2}, \ \theta_1 := 2 + \sqrt{2}$

(iii) $\theta_0 := 2 + \sqrt{2}, \ \theta_1 := 1 + \sqrt{2}$

(iv) $\theta_0 := 1 + \sqrt{2}, \ \theta_1 := 2 + \sqrt{(1 + \sqrt{2}) \cdot 2}$

Source: Andrew Ng, Machine Learning

คำถาม

ถ้า $\theta_0 = 1, \theta_1 = 2$ และเราปรับค่า θ_0 และ θ_1 พร้อมๆกัน โดยใช้

$$\theta_j := \theta_j + \sqrt{\theta_0 \theta_1}$$ (สำหรับ $j = 0, j = 1$)

ค่าของ θ_0 และ θ_1 จะเป็นเท่าไร

(i) $\theta_0 = 1, \quad \theta_1 = 2$

(ii) $\theta_0 := 1 + \sqrt{2}, \quad \theta_1 := 2 + \sqrt{2}$

(iii) $\theta_0 := 2 + \sqrt{2}, \quad \theta_1 := 1 + \sqrt{2}$

(iv) $\theta_0 := 1 + \sqrt{2}, \quad \theta_1 := 2 + \sqrt{(1 + \sqrt{2}) \cdot 2}$

Source: Andrew Ng, Machine Learning

2. Linear Regression ที่มี 1 ตัวแปร

2.6 Gradient Descent: ความเข้าใจพื้นฐาน

Krittameth Teachasrisaksakul

ทำความเข้าใจ Gradient Descent Algorithm

เราจะสำรวจสถานการณ์ที่ ใช้ parameter 1 ตัว

สูตร Gradient Descent ที่มี parameter 1 ตัว

ทำซ้ำจนกระทั่ง convergence

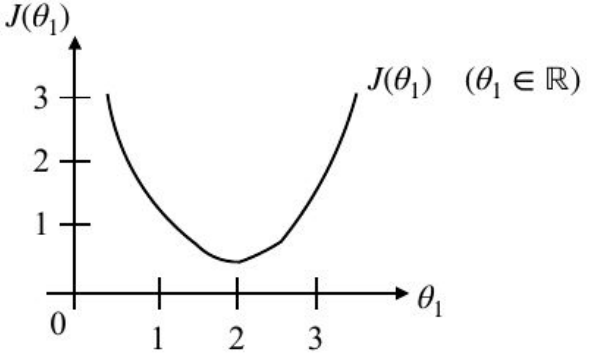
$$\theta_1 := \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta_1)$$

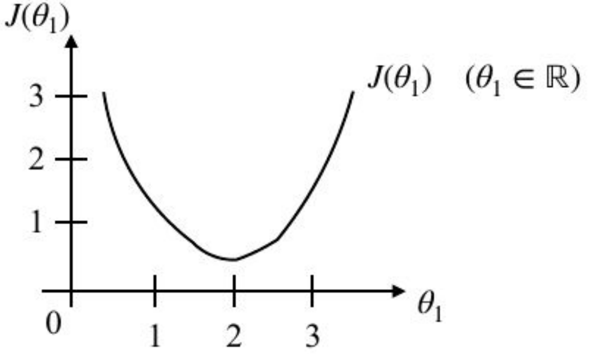
ถ้าไม่คำนึงถึงเครื่องหมายของ $\frac{d}{d\theta_1} J(\theta_1)$

ท้ายที่สุด θ_1 จะ converge ที่ค่าน้อยที่สุด (minimum)

- converge คือ ลงมาที่จุดต่ำสุดของ cost function

เข้าใจ Gradient Descent Algorithm จากกราฟ

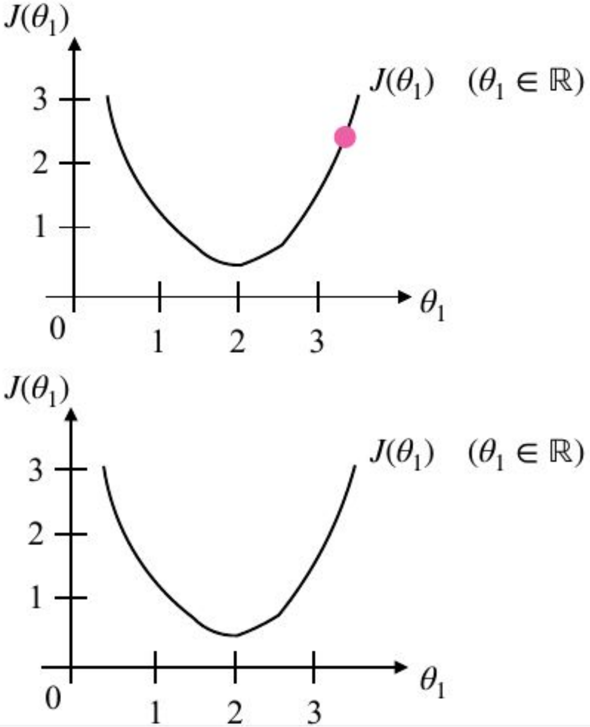




Plot cost function เพื่อทำ Gradient Descent

Figure Source: Teeradaj Racharak; AI Practical Develop- ment Bootcamp

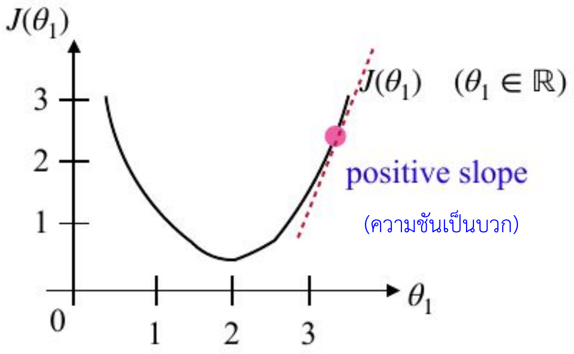
เข้าใจ Gradient Descent Algorithm จากกราฟ

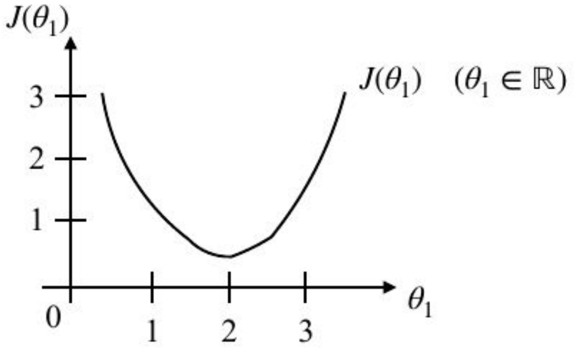


$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$

Figure Source: Teeradaj Racharak; AI Practical Develop- ment Bootcamp

เข้าใจ Gradient Descent Algorithm จากกราฟ

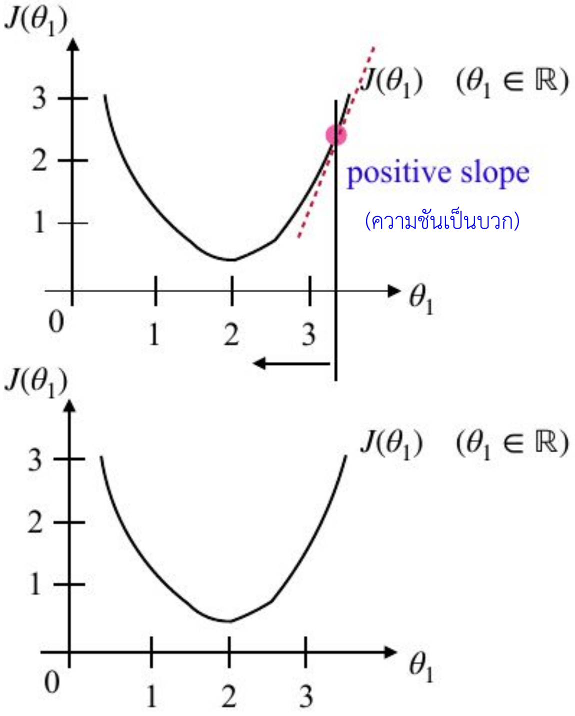




$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$

Figure Source: Teeradaj Racharak; AI Practical Develop- ment Bootcamp

เข้าใจ Gradient Descent Algorithm จากกราฟ



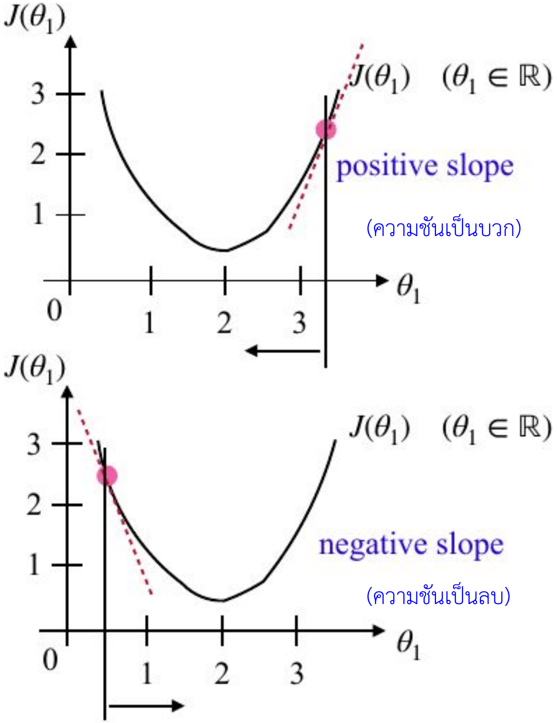
$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$

$$\theta_1 := \theta_1 - \alpha \cdot (+)$$

ความชัน / slope (+) : ค่าของ $\boldsymbol{\theta_1}$ ลดลง

Figure Source: Teeradaj Racharak; AI Practical Develop- ment Bootcamp

เข้าใจ Gradient Descent Algorithm จากกราฟ



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

$$\theta_1 := \theta_1 - \alpha \cdot (+)$$

ความชัน / slope (+) : ค่าของ $\boldsymbol{\theta_1}$ ลดลง

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

$$\theta_1 := \theta_1 - \alpha \cdot (-)$$

ความชัน / slope (-) : ค่าของ $\boldsymbol{\theta_1}$ เพิ่มขึ้น

Figure Source: Teeradaj Racharak; AI Practical Develop- ment Bootcamp

เข้าใจ Gradient Descent Algorithm จากกราฟ

รู้ได้ยังไงว่า ขนาด step ผิด ?

- ไม่ converge
- ใช้เวลามากเกินไปเพื่อหาค่าน้อยที่สุด (minimum)

เราควรปรับค่าของ parameter α เพื่อทำให้แน่ใจว่า Gradient Descent Algorithm converge ในเวลาที่เหมาะสม

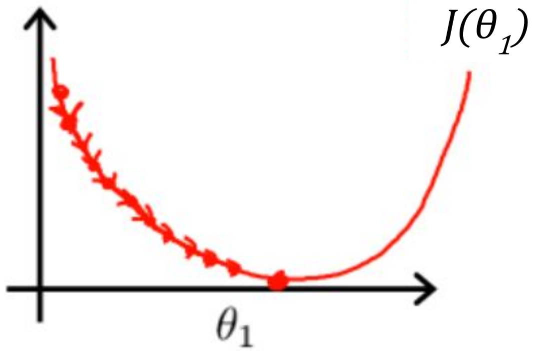
เข้าใจ Gradient Descent Algorithm จากกราฟ

ถ้า α น้อยไป: gradient descent จะ converge ช้า

ถ้า α มากไป: gradient descent อาจกระโดดเลยค่าต่ำสุด (minimum) และอาจไม่ converge หรืออาจ diverge (ลู่ออก หรือ ไม่ไปถึงค่าต่ำสุดของ cost function)

เข้าใจ Gradient Descent Algorithm จากกราฟ

ถ้า α น้อยไป: gradient descent จะ converge ช้า



ถ้า α มากไป: gradient descent อาจกระโดดเลยค่าต่ำสุด (minimum) และอาจไม่ converge หรืออาจ diverge (ลู่ออก หรือ ไม่ไปถึงค่าต่ำสุดของ cost function)

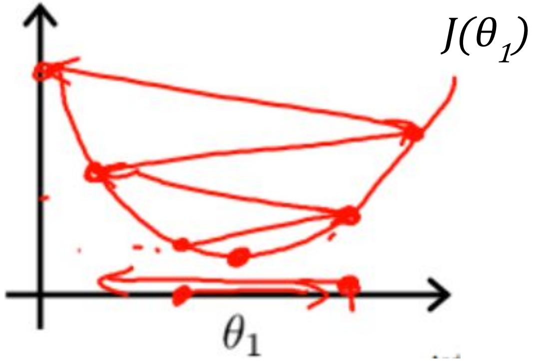


Figure source: Andrew Ng, Machine Learning

คำถาม

ถ้า θ_1 อยู่ที่ local optimum ของ $J(\theta_1)$ อย่างที่แสดงในกราฟ step 1 step ของ gradient descent

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

จะทำให้เกิดผลอะไร?

(i) ไม่เปลี่ยน θ_1

(ii) เปลี่ยนค่า θ_1 ในทิศทางที่สุ่ม

(iii) เปลี่ยนค่า θ_1 ในทิศทางของ global minimum

(iv) ลด θ_1

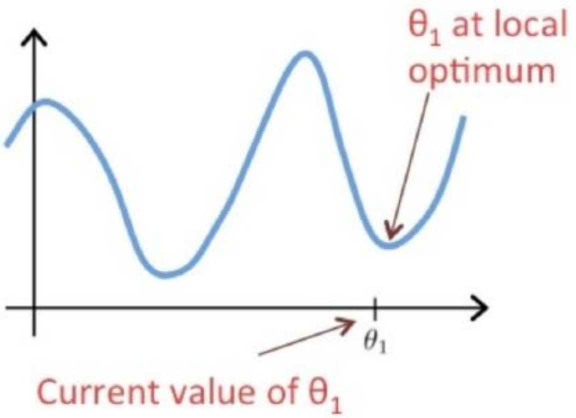


Figure Source: Teeradaj Racharak; AI Practical Develop- ment Bootcamp [L 1.2]

คำถาม

ถ้า θ_1 อยู่ที่ local optimum ของ $J(\theta_1)$ อย่างที่แสดงในกราฟ step 1 step ของ gradient descent

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

จะทำให้เกิดผลอะไร?

(i) ไม่เปลี่ยน θ_1

(ii) เปลี่ยนค่า θ_1 ในทิศทางที่สุ่ม

(iii) เปลี่ยนค่า θ_1 ในทิศทางของ global minimum

(iv) ลด θ_1

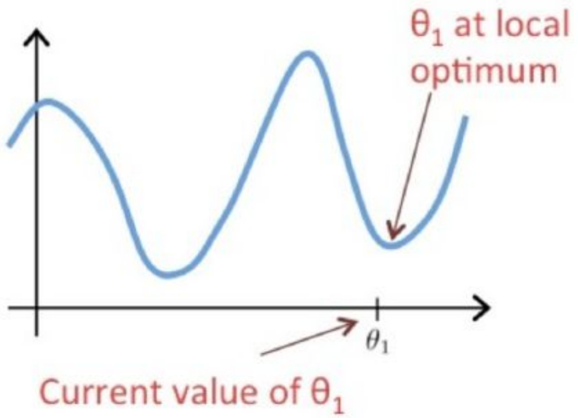


Figure Source: Teeradaj Racharak; AI Practical Develop- ment Bootcamp [L 1.2]

Gradient descent converge ได้ แม้ตั้งขนาด step α ให้คงที่

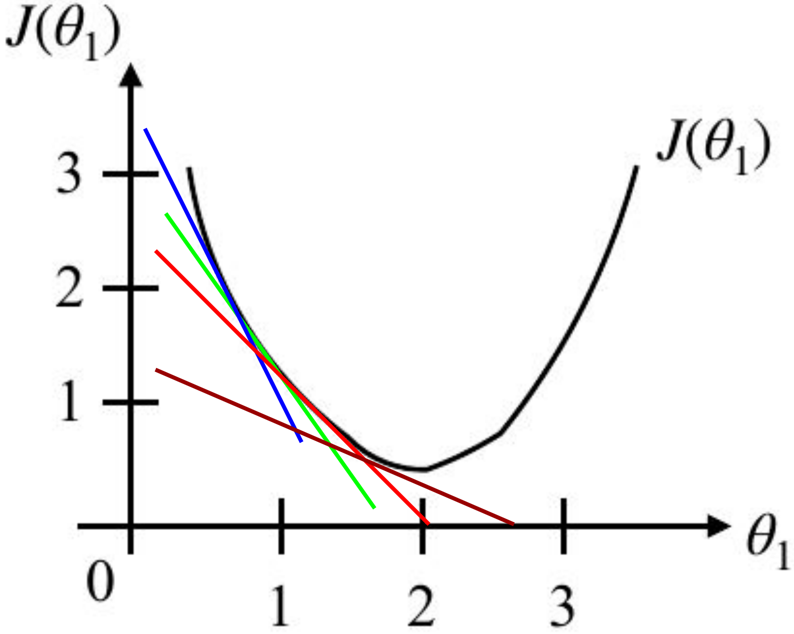
แม้ learning rate α จะคงที่ (ไม่เปลี่ยน α) $\longrightarrow$ gradient descent สามารถ converge เข้าสู่ค่า local minimum ได้

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

เมื่อเข้าใกล้ค่า lc [illegible] t จะปรับ step ให้เล็กลง โดยอัตโนมัติ

- เพราะค่า gradient หรือ derivative ของ cost function จะน้อยลง (เพราะความชันของเส้นสัมผัสกราฟ น้อยลง)

ดังนั้น เราจึงไม่ต้องลดค่า α ด้วยตัวเอง



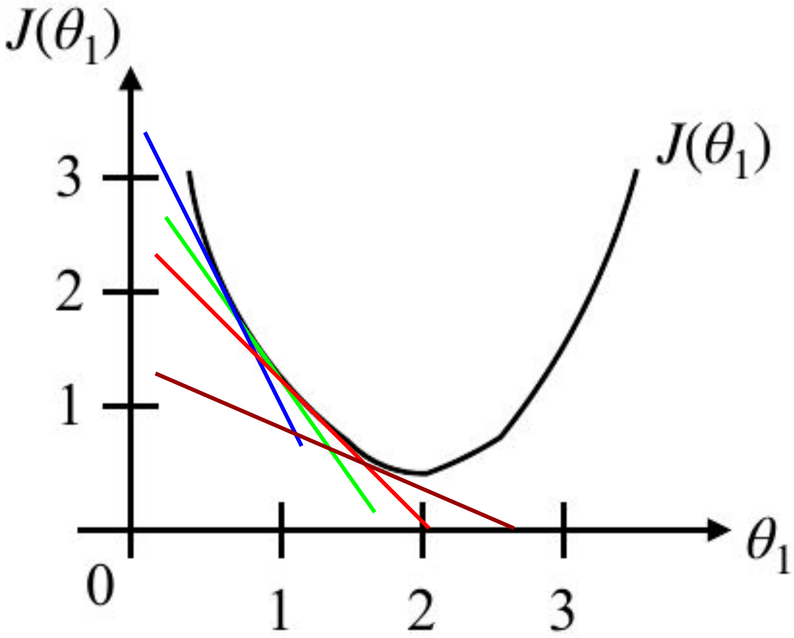
ทำไม gradient descent converge ได้ แม้ตั้งขนาด step α ให้คงที่

Derivative ของ cost function $\frac{\partial}{\partial \theta_1} J(\theta_1)$

เข้าใกล้ 0 เมื่อเราเข้าใกล้จุดต่ำสุดของ convex function

ที่ค่าน้อยที่สุด (minimum) derivative จะเป็น 0 เสมอ และจะได้

$$\theta_1 := \theta_1 - \alpha * 0$$



2. Linear Regression ที่มี 1 ตัวแปร

2.7 Gradient Descent สำหรับ Linear Regression

Krittameth Teachasrisaksakul

Gradient descent algorithm & linear regression model

Gradient Descent Algorithm

repeat until convergence {

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

}

Linear Regression Model

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

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

เป้าหมาย: ใช้ gradient descent เพื่อ (minimize) ทำให้ squared error function น้อยที่สุด

Gradient descent algorithm & linear regression model

$$\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

Gradient descent algorithm & linear regression model

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

Gradient descent algorithm & linear regression model

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

$$= \frac{\partial}{\partial\theta_j}[\frac{1}{2m}\Sigma_{i=1}^{m}(\boxed{\theta_0+\theta_1 x^{(i)}}-y^{(i)})^2]$$

Gradient descent algorithm & linear regression model

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

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

ใช้ Chain Rule (กฎลูกโซ่)

กรณี 1: เมื่อ $j = 0$

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

กรณี 2: เมื่อ $j = 1$

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

https://math.stackexchange.com/questions/70728/partial-derivative-in-gradient-descent-for-two-variables

Gradient descent algorithm สำหรับ linear regression 1 ตัวแปร

repeat until convergence {

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

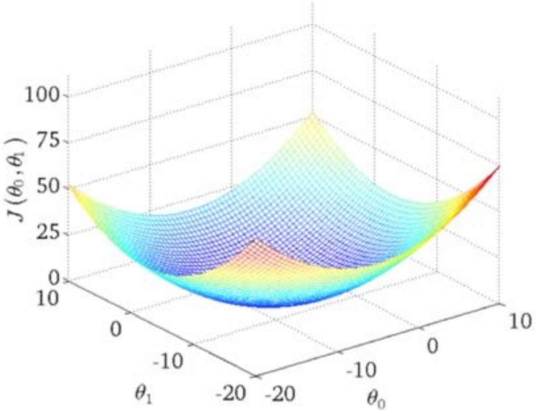
$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \Sigma_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x^{(i)}$$

}

ปรับค่า θ_0, θ_1 พร้อมๆกัน

Cost function ของ linear regression มีรูปร่าง เป็นชามเสมอ

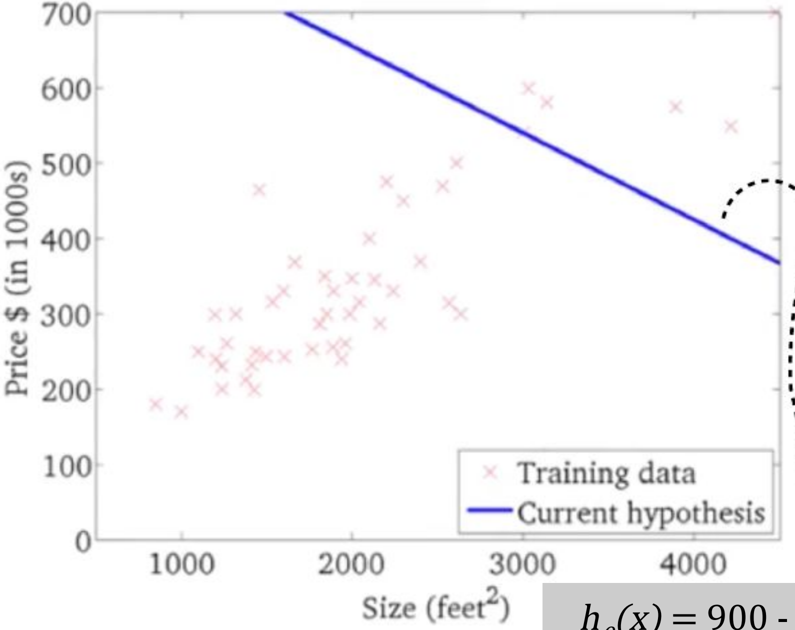
ก็คือ **"convex function"**



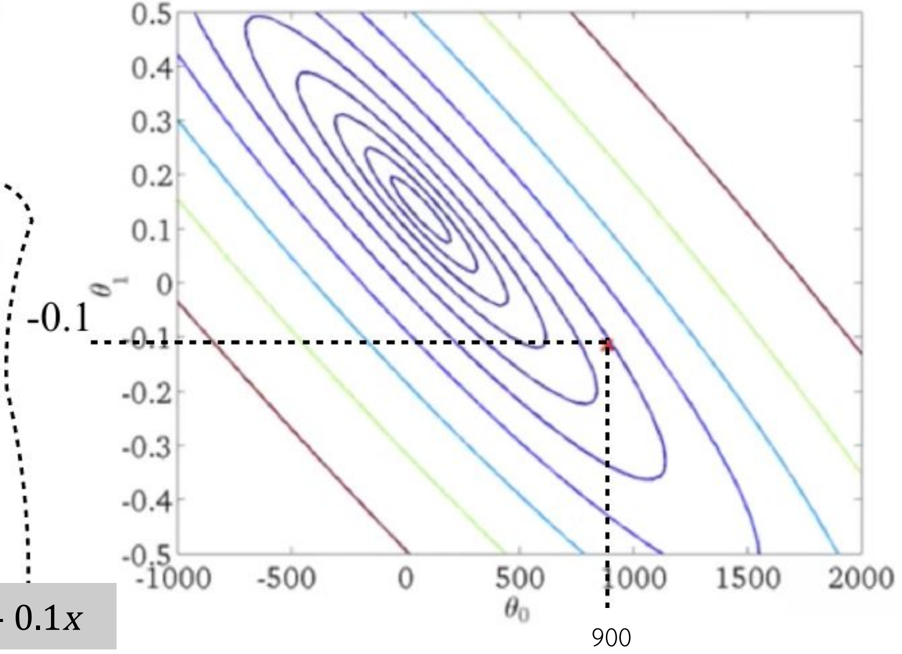
ตัวอย่างการใช้ Gradient Descent

$\boldsymbol{h_\theta(x)}$ (สำหรับค่า θ_0, θ_1 ที่คงที่ จะเป็นฟังก์ชั่นของ x)

$\boldsymbol{J(\theta_0, \theta_1)}$ (ฟังก์ชั่นของ parameters θ_0, θ_1)



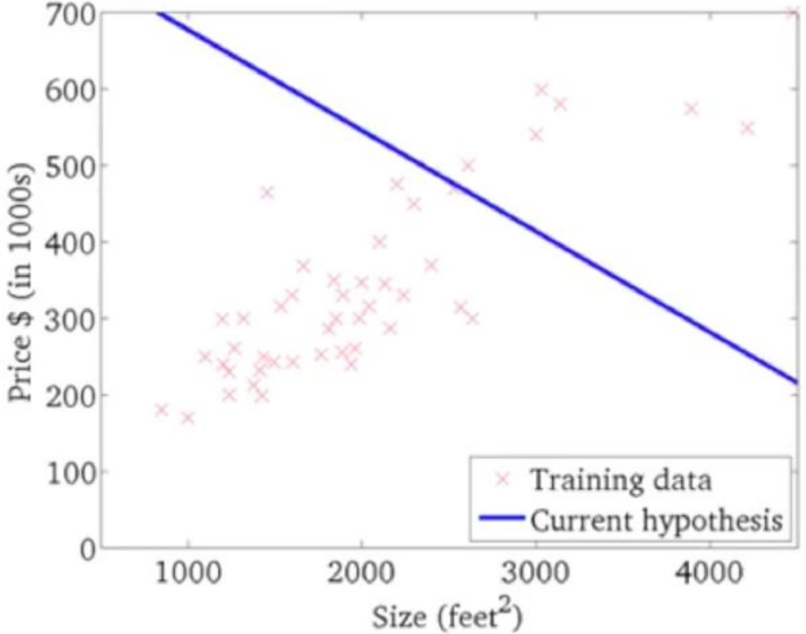
$h_\theta(x) = 900 - 0.1x$



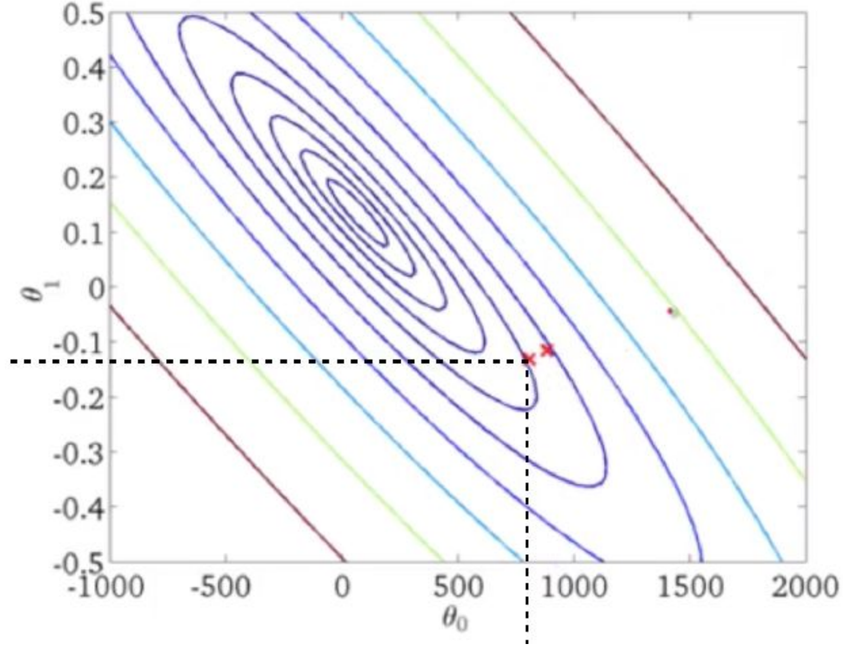
Source: Andrew Ng, Machine Learning

ตัวอย่างการใช้ Gradient Descent

$h_\theta(x)$ (สำหรับค่า θ_0, θ_1 ที่คงที่ จะเป็นฟังก์ชั่นของ x)



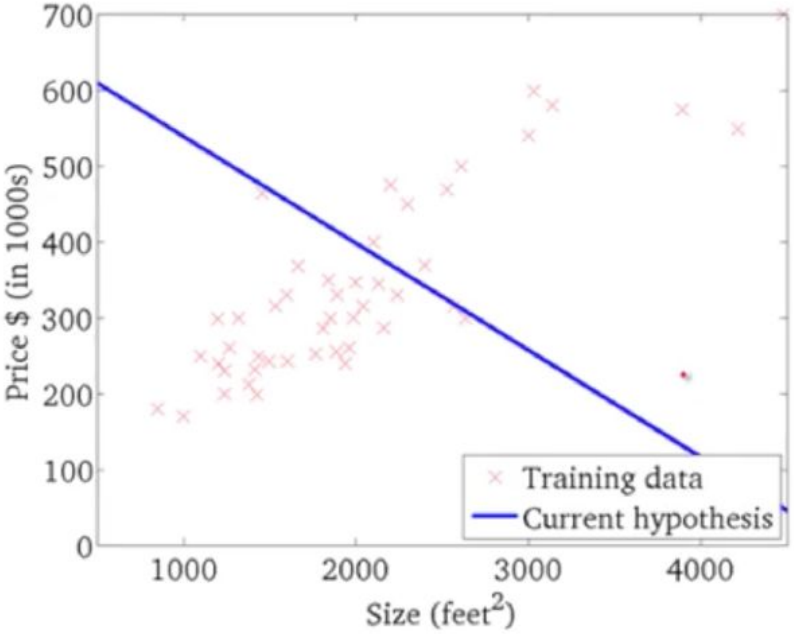
$J(\theta_0, \theta_1)$ (ฟังก์ชั่นของ parameters θ_0, θ_1)



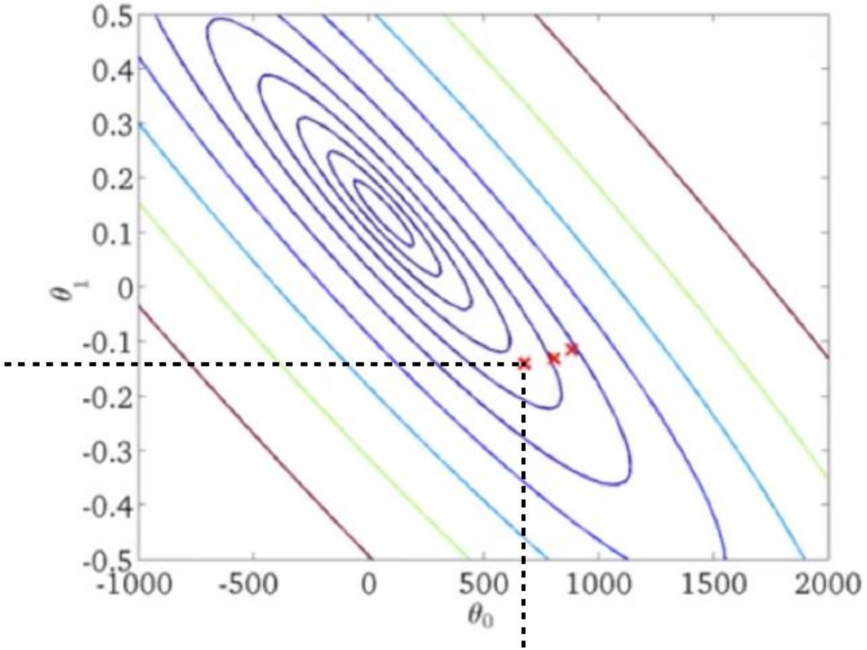
Source: Andrew Ng, Machine Learning

ตัวอย่างการใช้ Gradient Descent

$h_\theta(x)$ (สำหรับค่า θ_0, θ_1 ที่คงที่ จะเป็นฟังก์ชั่นของ x)



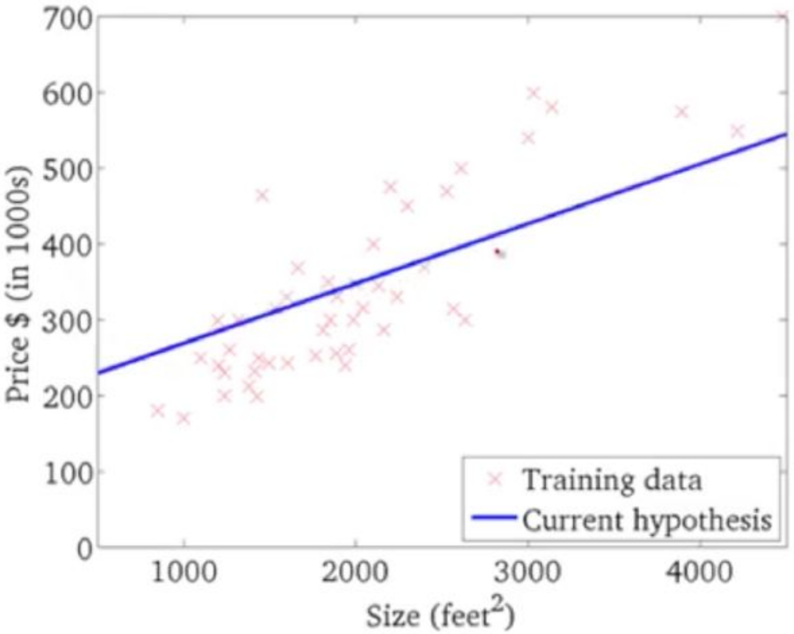
$J(\theta_0, \theta_1)$ (ฟังก์ชั่นของ parameters θ_0, θ_1)



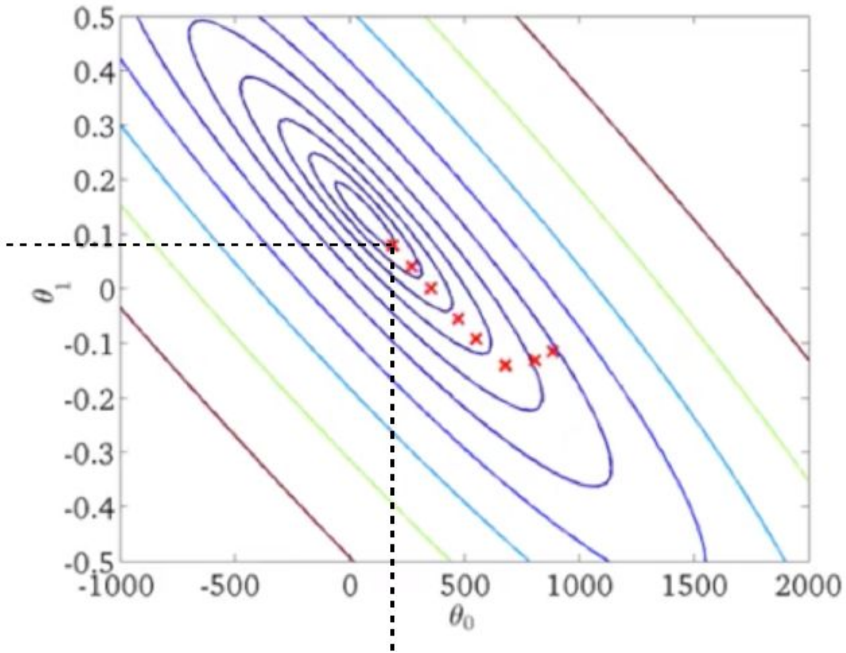
Source: Andrew Ng, Machine Learning

ตัวอย่างการใช้ Gradient Descent

$h_\theta(x)$ (สำหรับค่า θ_0, θ_1 ที่คงที่ จะเป็นฟังก์ชั่นของ x)



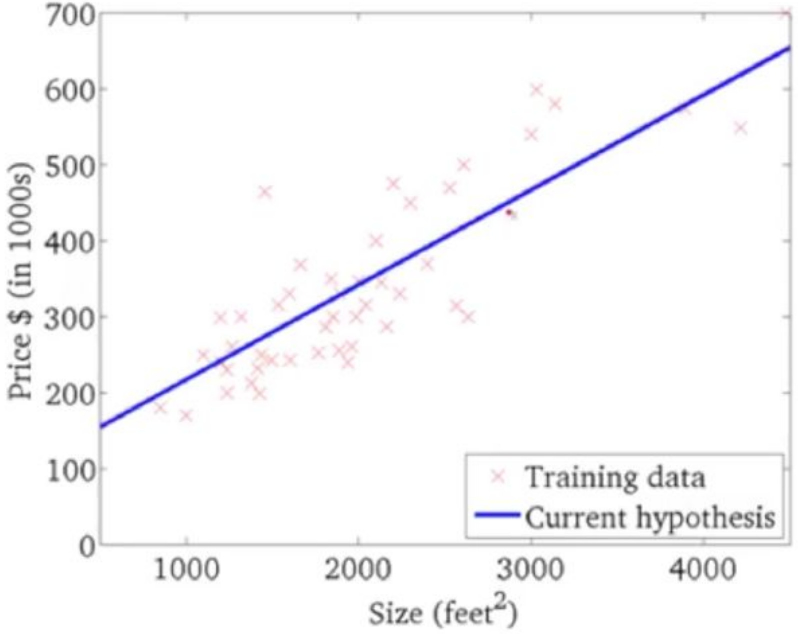
$J(\theta_0, \theta_1)$ (ฟังก์ชั่นของ parameters θ_0, θ_1)



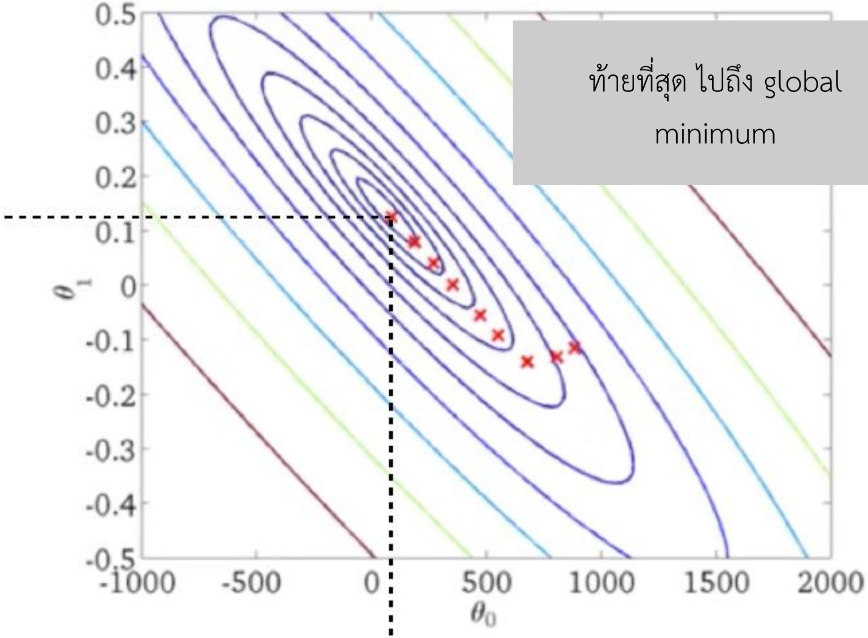
Source: Andrew Ng, Machine Learning

ตัวอย่างการใช้ Gradient Descent

$h_\theta(x)$ (สำหรับค่า θ_0, θ_1 ที่คงที่ จะเป็นฟังก์ชั่นของ x)



$J(\theta_0, \theta_1)$ (ฟังก์ชั่นของ parameters θ_0, θ_1)

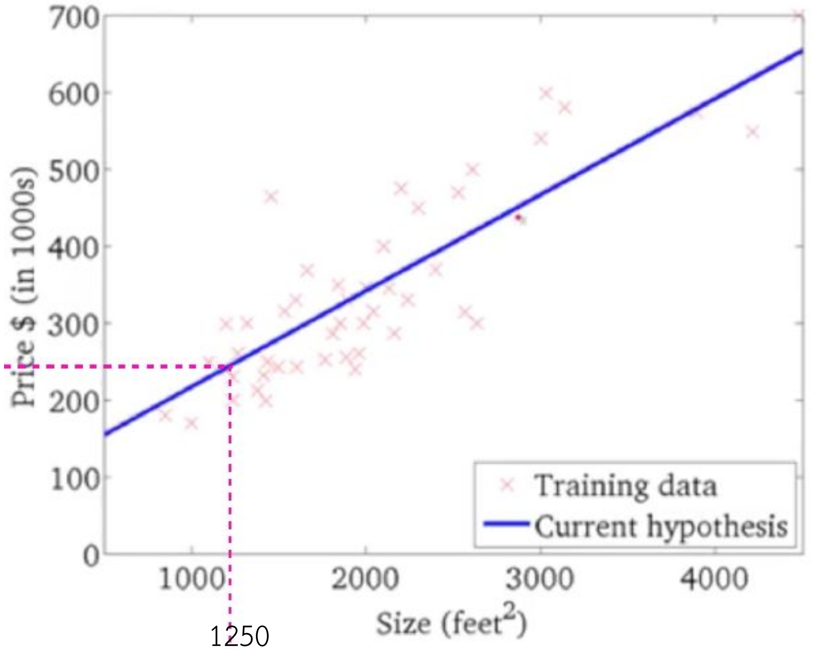


ท้ายที่สุด ไปถึง global minimum

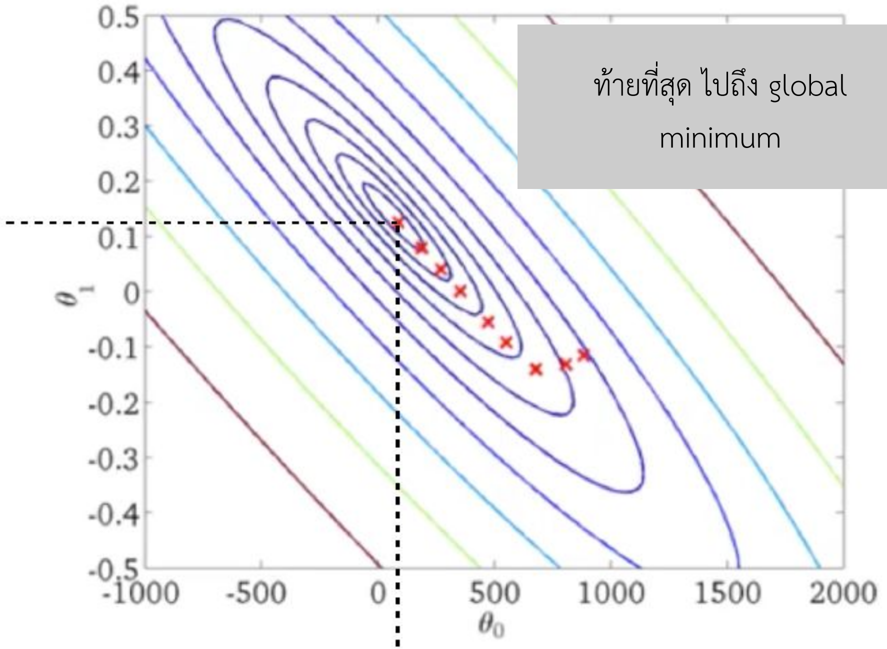
Source: Andrew Ng, Machine Learning

ตัวอย่างการใช้ Gradient Descent

$h_\theta(x)$ (สำหรับค่า θ_0, θ_1 ที่คงที่ จะเป็นฟังก์ชั่นของ x)



$J(\theta_0, \theta_1)$ (ฟังก์ชั่นของ parameters θ_0, θ_1)



ท้ายที่สุด ไปถึง global minimum

Source: Andrew Ng, Machine Learning

Batch Gradient Descent

Batch gradient descent

“Batch” หมายความว่า ในแต่ละ step ของ gradient descent ใช้**ตัวอย่างทั้งหมด**จาก training set (all the training examples) ก็คือ

$$\Sigma_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})$$

Source: Teeradaj Racharak; AI Practical Development Bootcamp [L 1.2]

วิธีที่ใช้แทน Batch Gradient Descent

1. Stochastic gradient descent (SGD)

อีกหนึ่งทางเลือกที่ใช้บ่อยใน machine learning

ปรับ parameter ซ้ำๆ (หลายๆครั้ง) โดยใช้ gradient ที่คำนวณจาก **ตัวอย่าง 1 ตัวอย่าง**จาก training set (training element)

2. Mini-batch gradient descent (Mini-batch)

- Stochastic = มีการกระจายของความน่าจะเป็นแบบสุ่ม (a random probability distribution)
- คำว่า "stochastic" มาจากความจริงที่ว่า gradient ของ cost function $\nabla_J(\theta)$ คิดมาจากตัวอย่าง (sample) 1 ตัวอย่างจาก training set ซึ่งเป็น stochastic approximation (การประมาณค่าแบบ stochastic) ของค่า gradient ที่แท้จริง

https://tupleblog.github.io/gradient-descent-part1/

SGD: Stochastic Gradient Descent

$\theta_0, \theta_1 :=$ some initial guess

repeat until convergence {

For $i \in \{1,\ldots,m\}$

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

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

}

Stochastic gradient descent มีแนวโน้มที่จะเข้าใกล้ค่าต่ำสุด (minimum) เร็วกว่า Batch gradient descent แต่มันอาจจะแกว่งรอบ ๆค่า minimum

มี SGD หลาย version (cf. [1, 2])

[1] Bottou, Leon (1998). "Online Algorithms and Stochastic Approximations". Online Learning and Neural Networks. Cambridge University Press. ISBN 978-0-521-65263-6

[2] Bottou, Leon. "Large-scale machine learning with SGD." Proceedings of COMPSTAT'2010. Physica-Verlag HD, 2010. 177-186.

Source: Teeradaj Racharak; AI Practical Development Bootcamp [L 2.1]

Mini-batch Gradient Descent

$\theta_0, \theta_1 :=$ some initial guess

repeat until convergence {

let $\quad k \in \{1, \ldots, m\}$

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

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

}

Source: Teeradaj Racharak; AI Practical Development Bootcamp [L 2.1]

SGD vs. Mini-batch

SGD

- โดยทั่วไป SGD สามารถหลีกเลี่ยงค่า local minimum เพราะความไม่มีแบบแผน (randomness) ของ input
- SGD อาจใช้เวลานานกว่าที่จะ converge

Mini-batch

- เป็น compromised version ของ GD และ SGD
- สามารถใช้ความสามารถในการคำนวณ (computational power) ถ้า batch size ใหญ่ (batch size ทั่วไป คือ 50)

แนะนำให้ใช้ gradient descent แบบทั่วไป เมื่อ $m <= 1000$

Source: Teeradaj Racharak; AI Practical Development Bootcamp [L 2.1]

คำถาม

วงทุกข้อที่ถูกต้อง

(a) เพื่อทำให้ gradient descent converge เราต้องลด α อย่างช้าๆ ตามเวลา

(b) Gradient descent สามารถหา global minimum ของ function $J(\theta_0, \theta_1)$ ใดๆ ก็ได้

(c) Gradient descent สามารถ converge ได้ แม้ตั้ง α ให้คงที่ (แต่ α ไม่สามารถมากเกินไป ไม่อย่างนั้น gradient descent อาจไม่ converge)

(d) สำหรับ cost function $J(\theta_0, \theta_1)$ ที่เฉพาะเจาะจงที่ใช้ใน linear regression จะไม่มีค่า local optima อื่นๆ นอกจากค่า global optimum

คำถาม

วงทุกข้อที่ถูกต้อง

(a) เพื่อทำให้ gradient descent converge เราต้องลด α อย่างช้าๆ ตามเวลา

(b) Gradient descent สามารถหา global minimum ของ function $J(\theta_0, \theta_1)$ ใดๆ ก็ได้

(c) Gradient descent สามารถ converge ได้ แม้ตั้ง α ให้คงที่ (แต่ α ไม่สามารถมากเกินไป ไม่อย่างนั้น gradient descent อาจไม่ converge)

(d) สำหรับ cost function $J(\theta_0, \theta_1)$ ที่เฉพาะเจาะจงที่ใช้ใน linear regression จะไม่มีค่า local optima อื่นๆ นอกจากค่า global optimum

สรุป

มี 3 วิธี minimize (ทำให้น้อยสุด) cost function $J(\theta)$

1. Batch gradient descent: ใช้ training example ทั้งหมด
2. Stochastic gradient descent: ใช้ training example 1 ตัว
3. Mini-batch gradient descent

นอกจากนี้ ยังมีวิธีที่ไม่ทำซ้ำ (non-iterative methods) เช่น การใช้ normal equations ซึ่งเราจะเรียนในบทต่อไป

หมายเหตุ: training example = ตัวอย่างข้อมูลจากชุดข้อมูล training set

ทำไมใช้ mean squared error เป็น cost function ของ linear regression ?

ทำไมไม่ใช้ cost function อื่น ?

สำหรับ linear regression: วิธี least squares กับ maximum likelihood เทียบเท่ากัน (equivalent)

ซึ่งได้มาจาก assumption (ข้อสมมติ) ว่า ตัวอย่าง (samples) $y^{(i)}$ มี Gaussian errors ที่กระจายตัวแบบ independently และ identically

เราจะเรียนเรื่องนี้ในบทถัดไป

The equivalence comes from the assumption of independently and identically distributed Gaussian errors in our samples $y^{(i)}$.

References

1. Andrew Ng, Machine Learning, Coursera.
2. Teeradaj Racharak, AI Practical Development Bootcamp.
3. What is Machine Learning?, https://www.digitalskill.org/contents/5