



# **Natural language processing (NLP)**

## **Crash Course**

**Attawut Nardkulpat**

**Faculty of Geoinformatics**

**Burapha University**

**[attawut@buu.ac.th](mailto:attawut@buu.ac.th)**



# Staff

- Instructor: Attawut Nardkulpat
  - Email: Attawut@buu.ac.th
  - Office: SCGI Room, 5 floor QS2
  - Office hour: 8:30 – 16:30

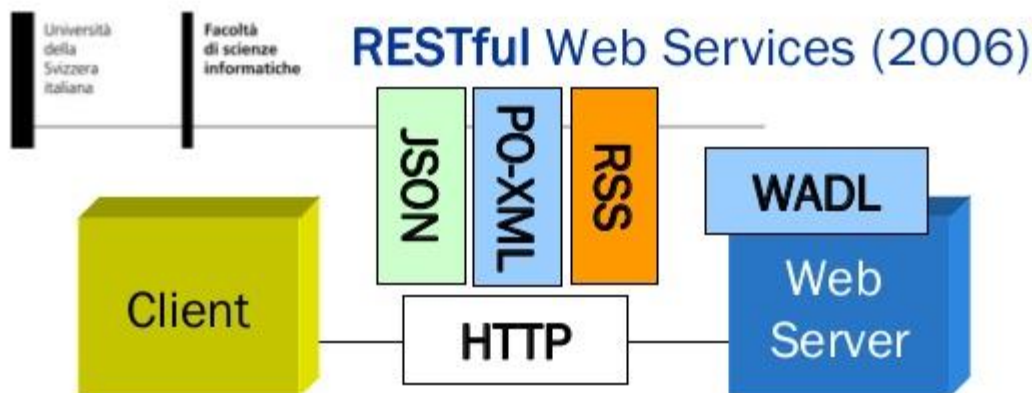


# Outline

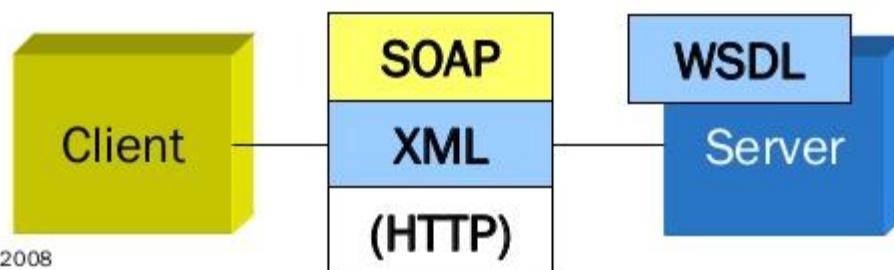
- Week 1 : Basic Python & Numpy
- **Week 2 : Pandas & Matplotlib (Exploratory Data Analysis)**
- Week 3 : API & Sentiment Analysis  
(Logistic Regression & Naïve Bayes)
- Week 4 : Sentiment Analysis (Word Embedding, Recurrent Neural Network & Long-Short Term Memory)



# REST และ SOAP



### WS-\* Web Services (2000)



7.10.2008

©2008 Cesare Pautasso



# SOAP

- SOAP ย่อมาจาก Simple Object Access Protocol ถูกกำหนดขึ้นมาตั้งแต่ปี 1998 เป้าหมายเพื่อใช้ในตลาด enterprise โดยเฉพาะ SOAP นั้นต้องสร้างให้ application logic ออกมาเป็น service โดยเป้าหมายเพื่อเป็น protocol ใหม่ของการติดต่อสื่อสาร



## SOAP (ต่อ)

- ข้อดี
  - - สามารถทำงานอยู่บน protocol ใด ๆ ก็ได้
  - - อธิบาย service ด้วย WDSL (Web Service Description Language)
  - - มีความน่าเชื่อถือ เมื่อเกิดปัญหาสามารถทำการ retry ได้
  - - สนับสนุนเรื่อง security อยู่แล้ว ทั้ง authentication, authorization และ การเข้ารหัสข้อมูล



## SOAP (ต่อ)

- ข้อเสีย
  - - ยากต่อการพัฒนา ทำให้ไม่เป็นที่นิยมสำหรับระบบ web และ mobile
  - - สนับสนุนรูปแบบข้อมูล XML เพียงอย่างเดียว
  - - เนื่องจากมันเป็น standard ทำให้มีข้อจำกัดเยอะ
  - - เนื่องจากโครงสร้างมันมีหลายส่วนทำให้มี overhead สูง หรือ ต้องใช้ bandwidth สูงกว่า REST





# SOAP (ต่อ)

- เมื่อไรถึงควรใช้ SOAP
  - เมื่อต้องการจัดการ transaction เมื่อต้องทำงานกับหลาย ๆ ระบบ
  - เมื่อต้องการความเข้มงวดในการเชื่อมต่อระหว่าง client/server
  - เมื่อตัวอย่างเช่น Financial service และ Telecommunication service



# REST

- REST ย่อมาจาก Representational State Transfer ถูกกำหนดขึ้นมาตั้งแต่ปี 2000 เป้าหมายเพื่อเป็นรูปแบบหนึ่งในการออกแบบ open web technology โดยต้องการทำให้ข้อมูลอยู่ในรูปแบบของ resource ส่วนการกระทำต่าง ๆ เป็นไปตาม HTTP Verb หรือ HTTP Method (GET, POST, PUT, DELETE) และทำงานแบบ Stateless



# REST ต่อ

- ข้อดี
  - ทำการอยู่บน HTTP และทำตามมาตรฐานของ HTTP จึงทำให้พัฒนาได้ง่าย
  - สนับสนุนรูปแบบข้อมูลมากมาย เช่น XML, JSON, Plain Text และอื่น ๆ อีกมากมาย
  - รองรับการขยายระบบได้ง่าย
  - มีประสิทธิภาพการทำงานที่ดี
  - รองรับเรื่อง caching ข้อมูล



# REST ต่อ

- ข้อเสีย
  - ทำงานได้เฉพาะ HTTP protocol เท่านั้น
  - ไม่มีเรื่องของ security และ reliability มาให้ในตัว ดังนั้นต้องทำเอง
  - รูปแบบข้อมูลที่ส่งไปมาระหว่าง client-server ไม่มีข้อจำกัดอะไรเลย



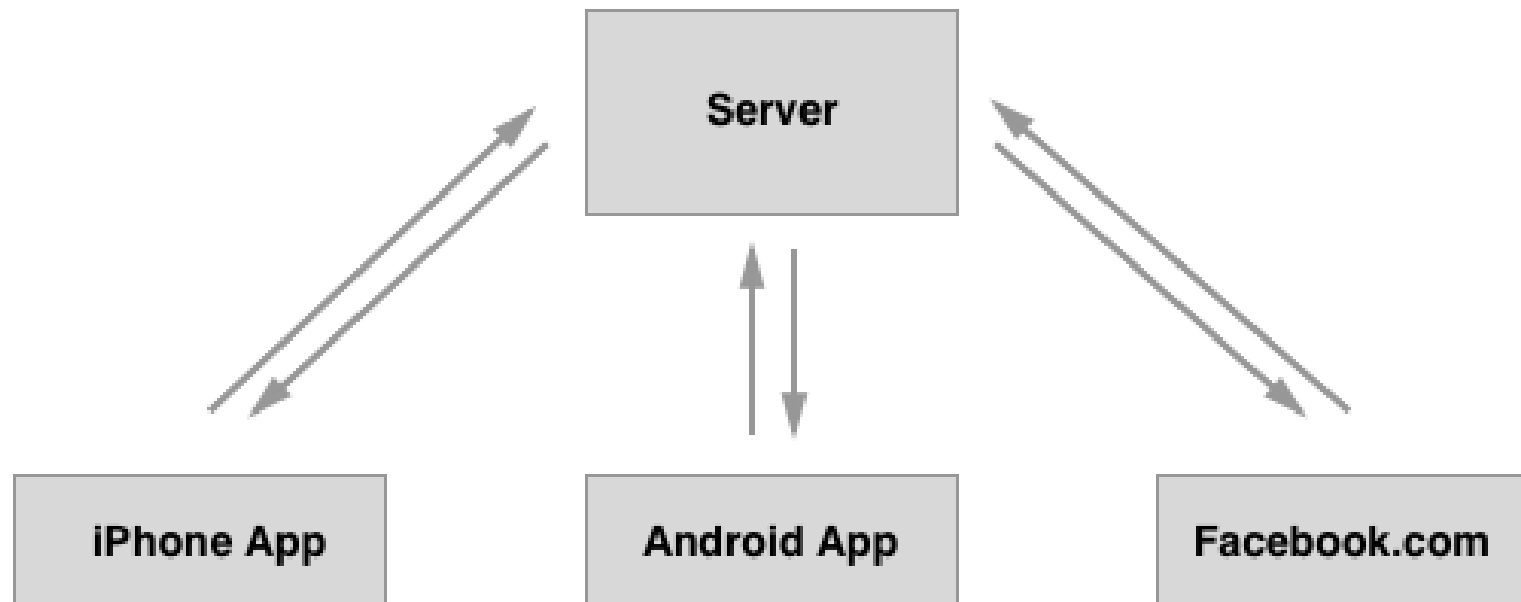
# REST ต่อ

- เมื่อไรถึงควรใช้ REST
  - เมื่อต้องการลดขนาดของข้อมูล และ จำนวน bandwidth ที่ใช้งาน
  - เมื่อต้องการเมื่อทำงานอยู่บนระบบ web และ mobile ตัวอย่างเช่น Social media service, Web Chat service



# API : Application Programming Interface

Credit : <https://medium.com/skooldio>





## API (ต่อ)

- API ซึ่งประกอบไปด้วย
  - สิ่งอะไรได้บ้าง มองแบบง่ายๆแต่ละคำสั่งคือ 1 API
  - แต่ละคำสั่งต้องบอกอะไรเพิ่ม



# Linear Regression

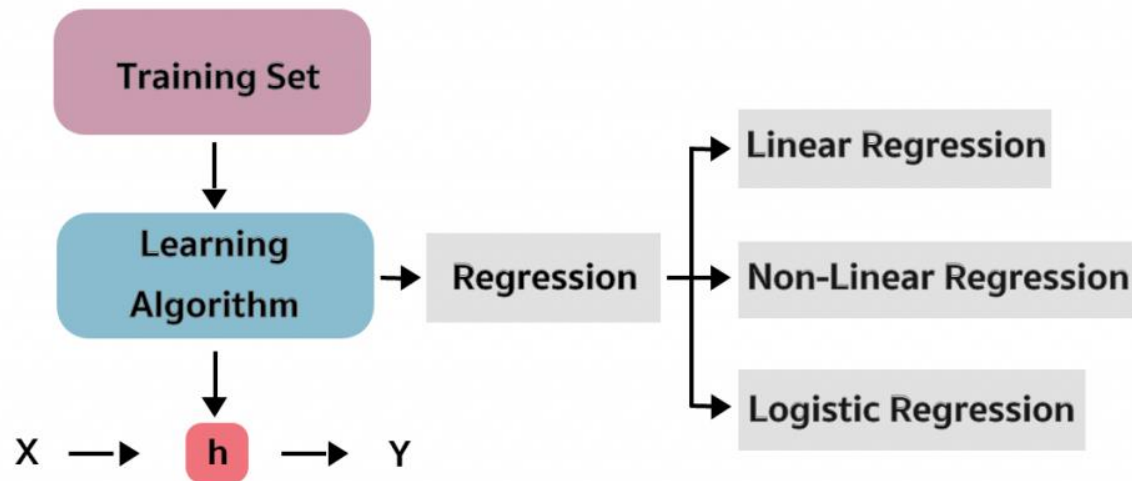




# Linear Regression

Credit : <https://ilog.ai/category/machine-learning/>

กระบวนการการเรียนรู้ของ ML แบบ Regression





# Regression

- คือ การวิเคราะห์ความสัมพันธ์ระหว่างตัวแปร โดยตัวแปรที่เราทราบค่า เรียกว่า ตัวแปรอิสระ(X) และตัวแปรที่เราจะประมาณค่า(ต้องการที่จะรู้)เรียกว่าตัวแปรตาม(Y) ซึ่งในการวิเคราะห์แบบ Regression นี้อาจจะมีตัวแปรอิสระ(X)มากกว่า 1 ตัวก็ได้



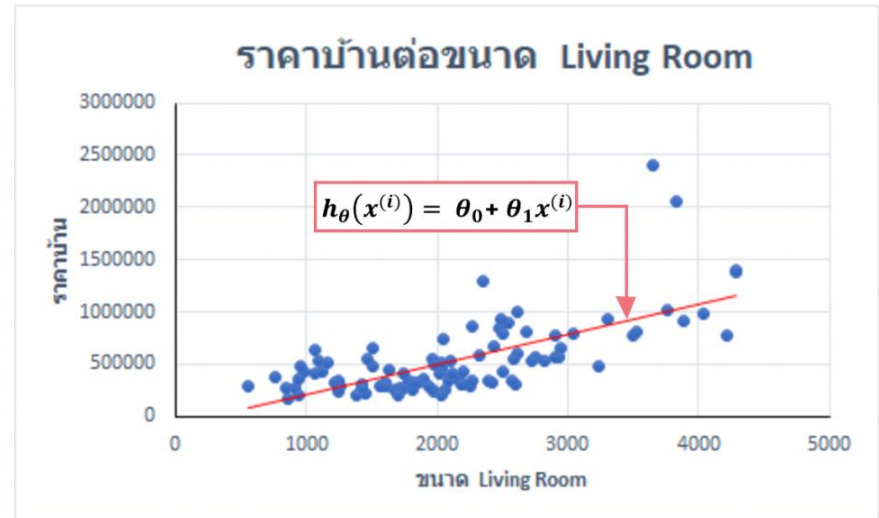
## Training Set

ขนาด living (X)	ราคา (Y)
1430	310000
2950	650000
1710	233000
2320	580500
1090	535000
2620	605000
4220	775000
2250	292500
.....	.....

$m$  = จำนวนชุดข้อมูลที่ใช้(100ชุด)

$X^{(i)}$  = ขนาด Living room( $i$ )

$Y^{(i)}$  = ราคาบ้านที่( $i$ )



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

จากการพล็อตข้อมูลดังรูป จะเห็นได้ว่าข้อมูลมีการกระจายตัวแบบกราฟเส้นตรง ซึ่งสมการเส้นตรงโดยทั่วไปจะเขียนได้ว่า  $Y = aX + b$  แต่ใน ML เราใช้สมการ  $h_{\theta}(x^{(i)}) = \theta_0 + \theta_1 x^{(i)}$  แทน โดยที่ค่าของ  $b, a = \theta_0, \theta_1$  ตามลำดับ

[illegible]

### จากรูปกำหนดให้

$(Y_{\text{จริง}})$  = ราคาบ้านที่แท้จริง



$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^m (\underbrace{h_{\theta}(x^{(i)})}_{\text{Cost Function}} - \underbrace{y^{(i)}}_{\text{Yจริง}})^2$$

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



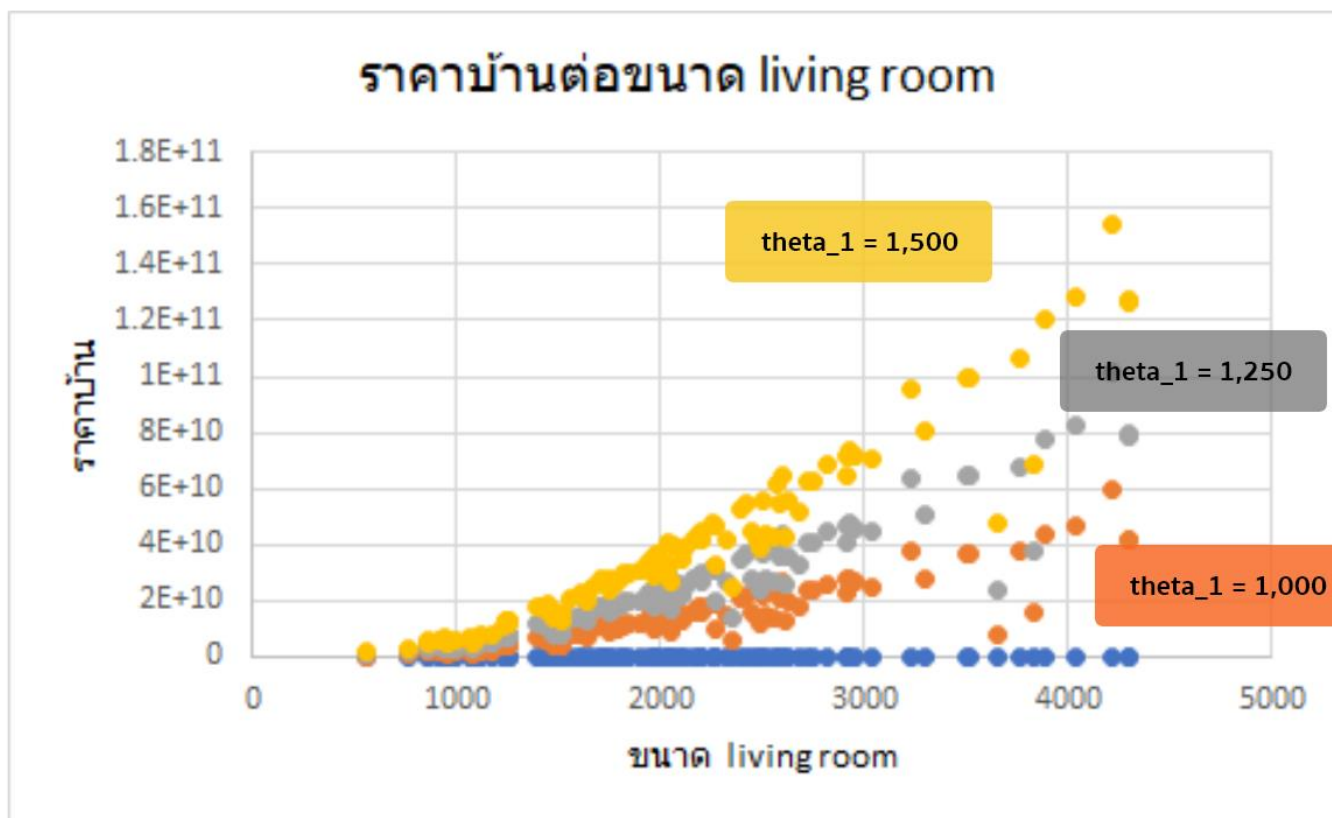
## สิ่งที่เรามีตอนนี้คือ

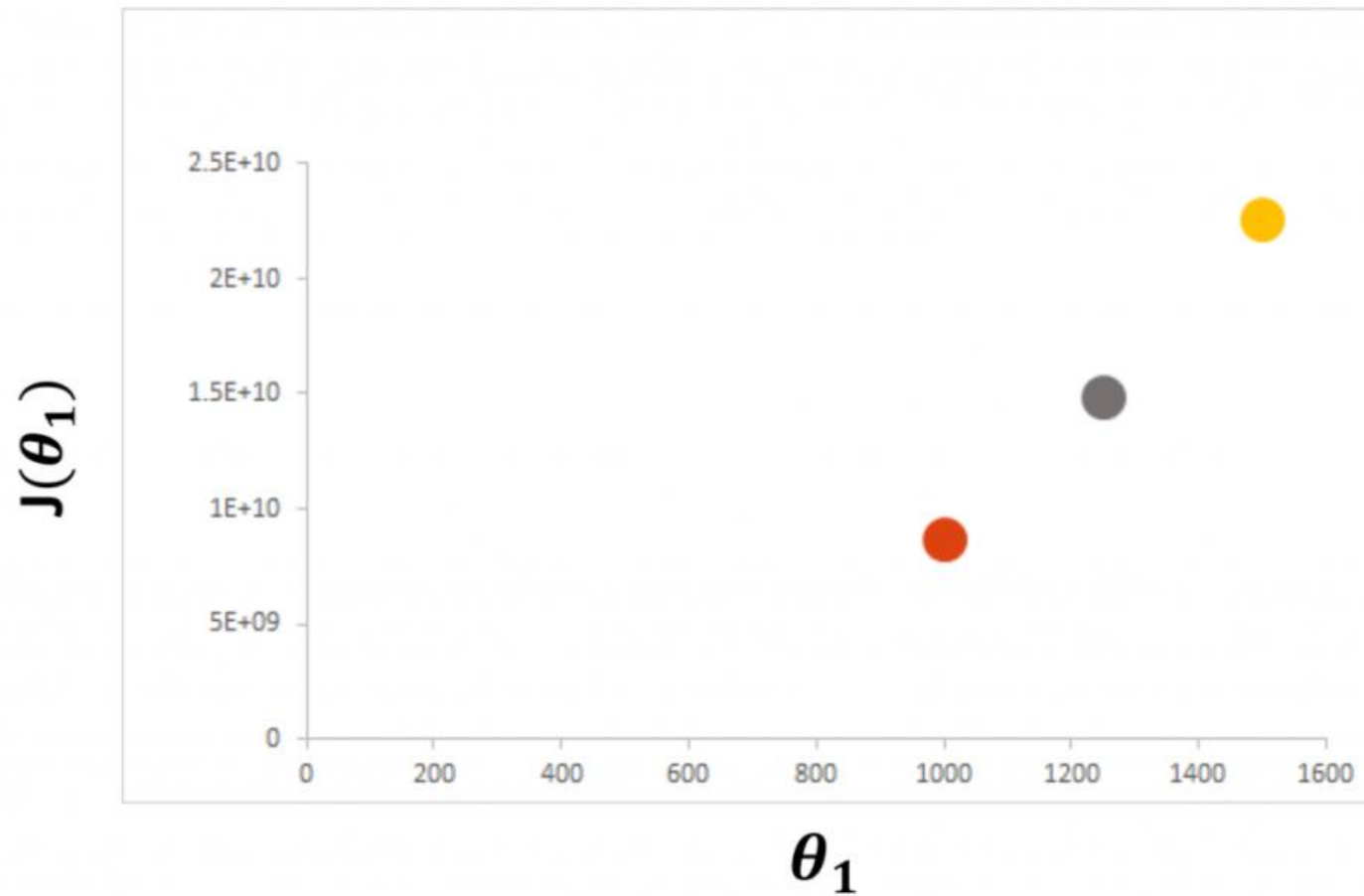
Hypothesis :  $h_{\theta}(x^{(i)}) = \theta_0 + \theta_1 x^{(i)}$

Parameter :  $\theta_0, \theta_1$

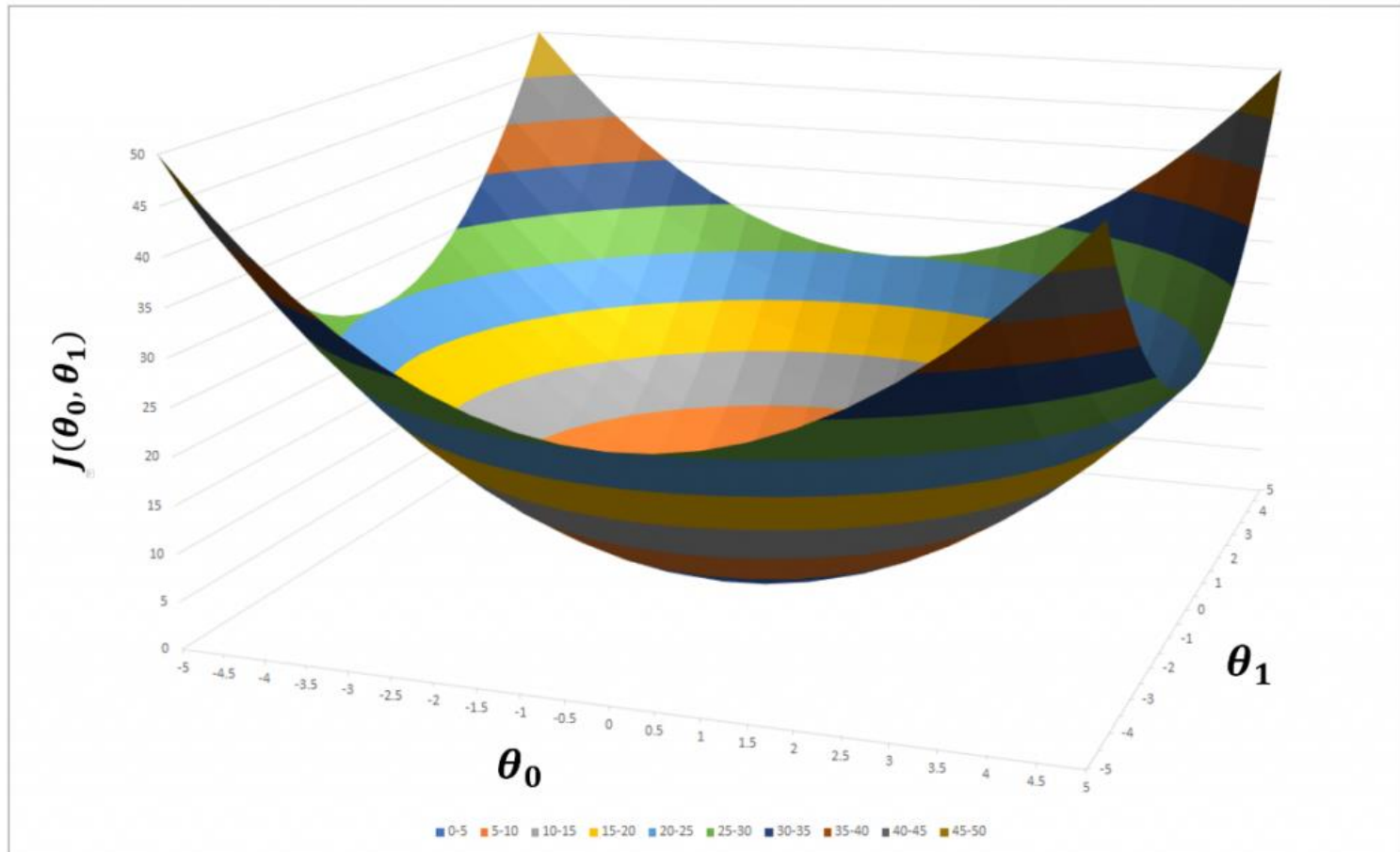
Cost Function :  $J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$

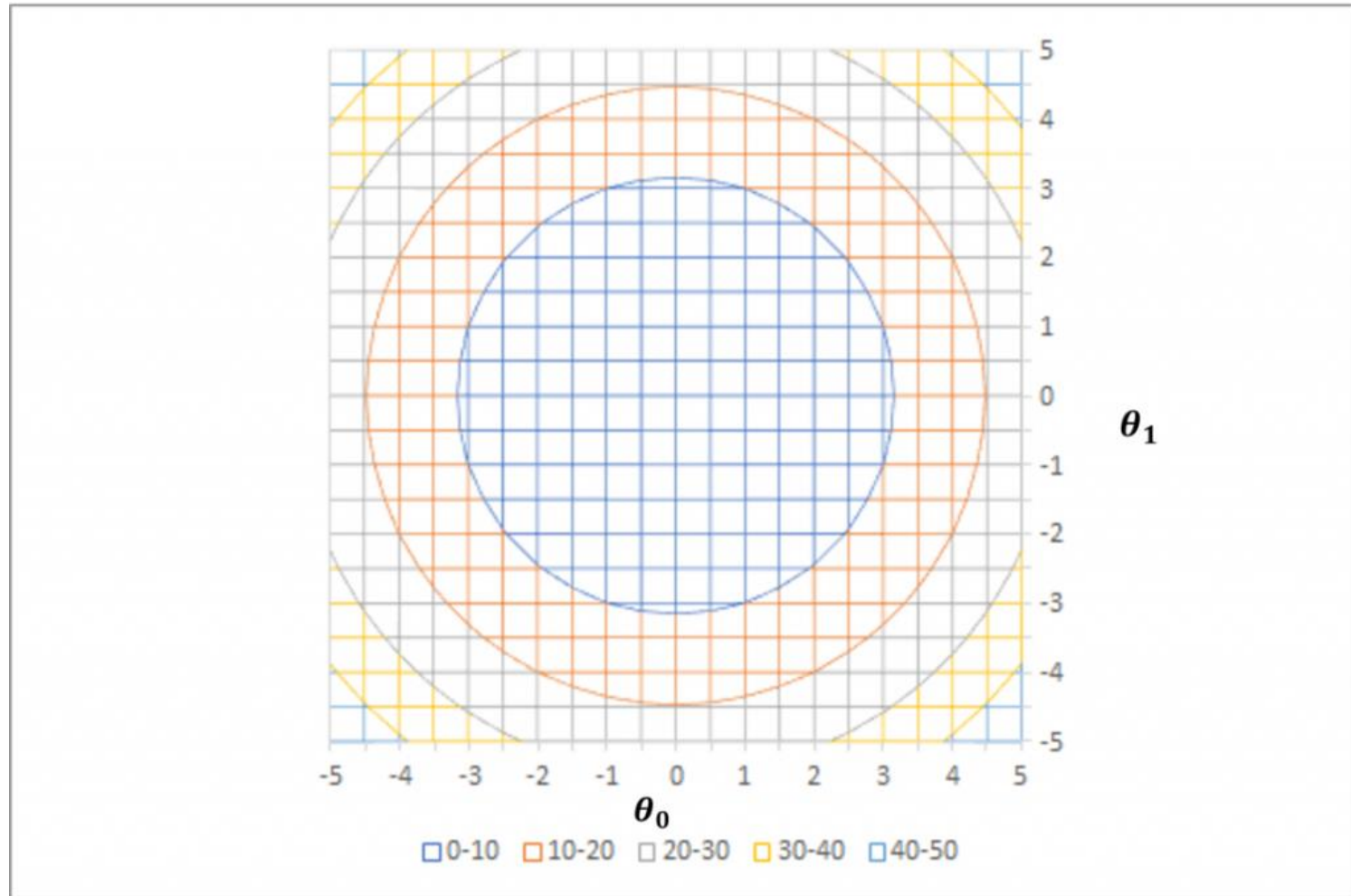
Goal : minimize  $J(\theta_0, \theta_1)$





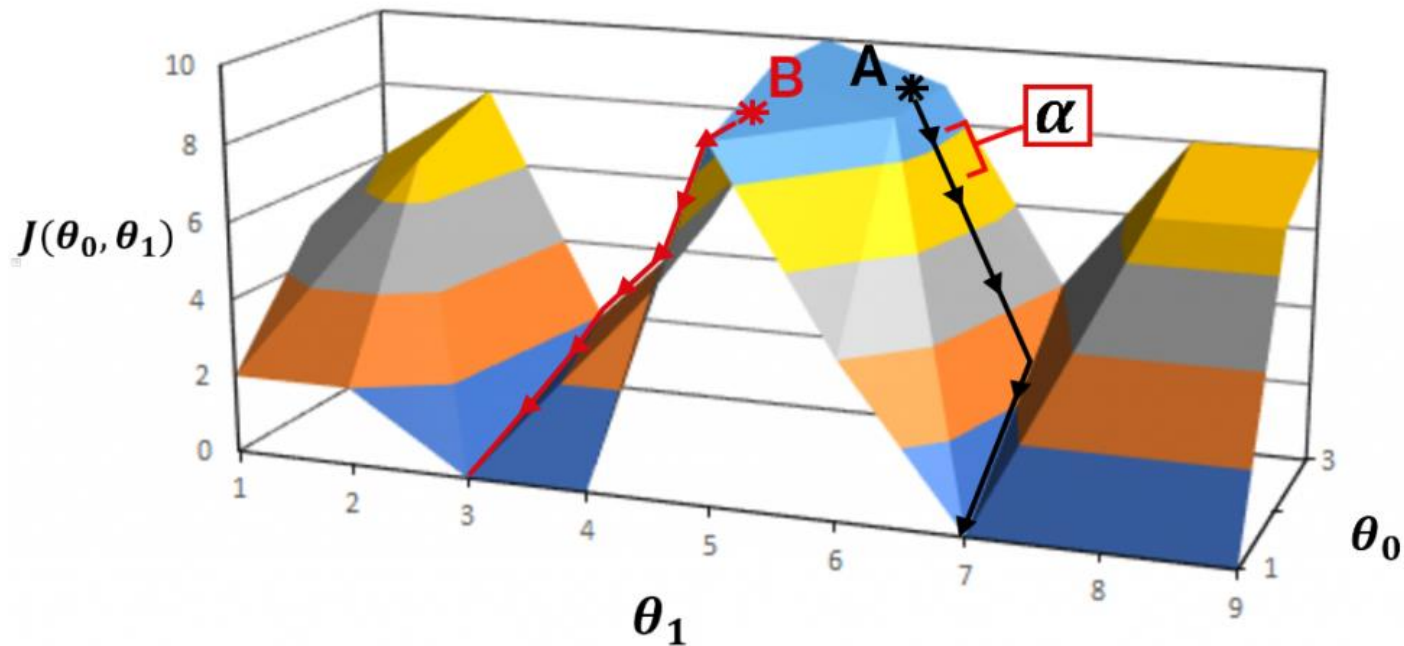








# Gradient Descent





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



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

Assignment

j=Feature Index Number

Learning Rate

Partial Derivative

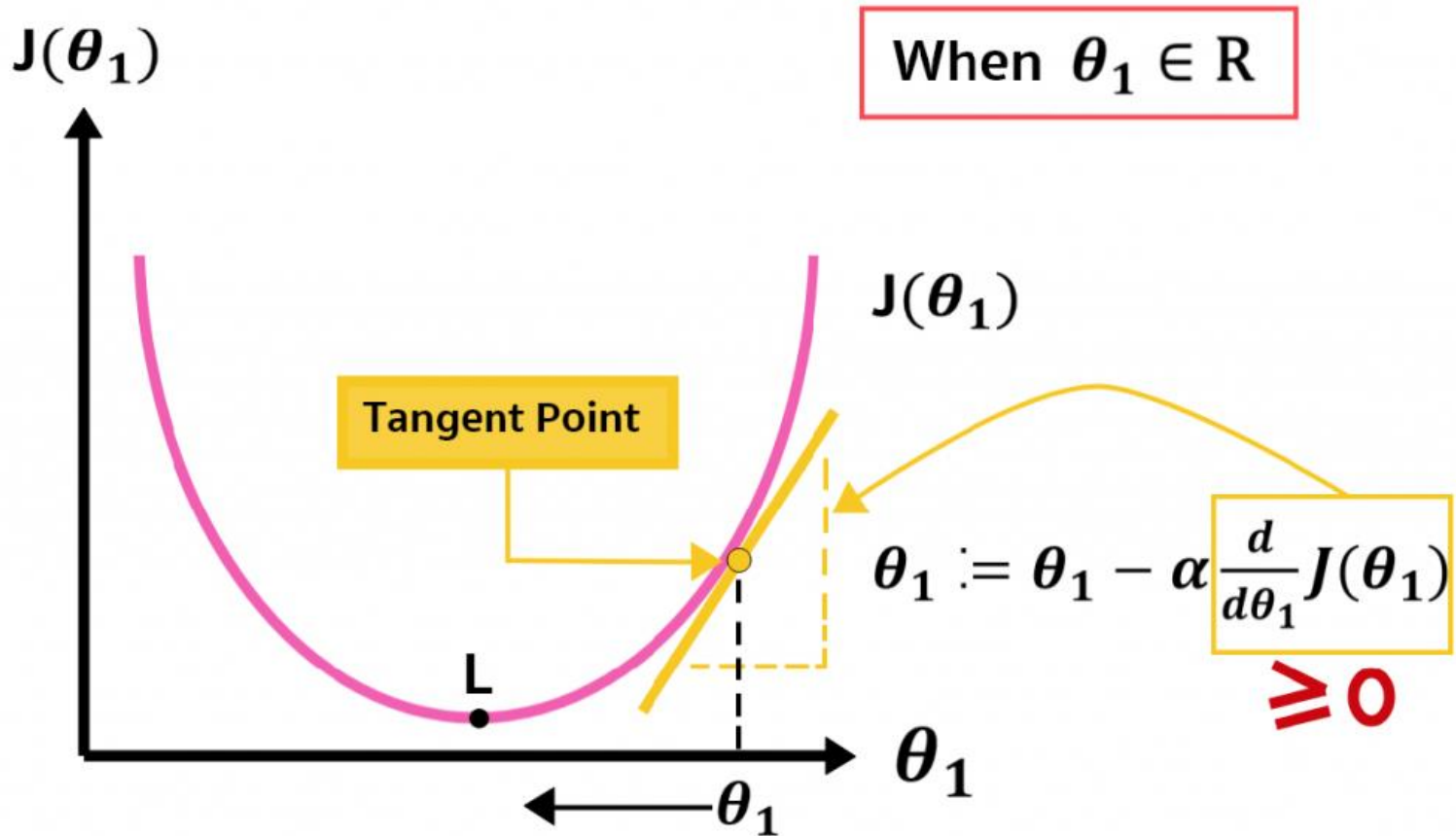


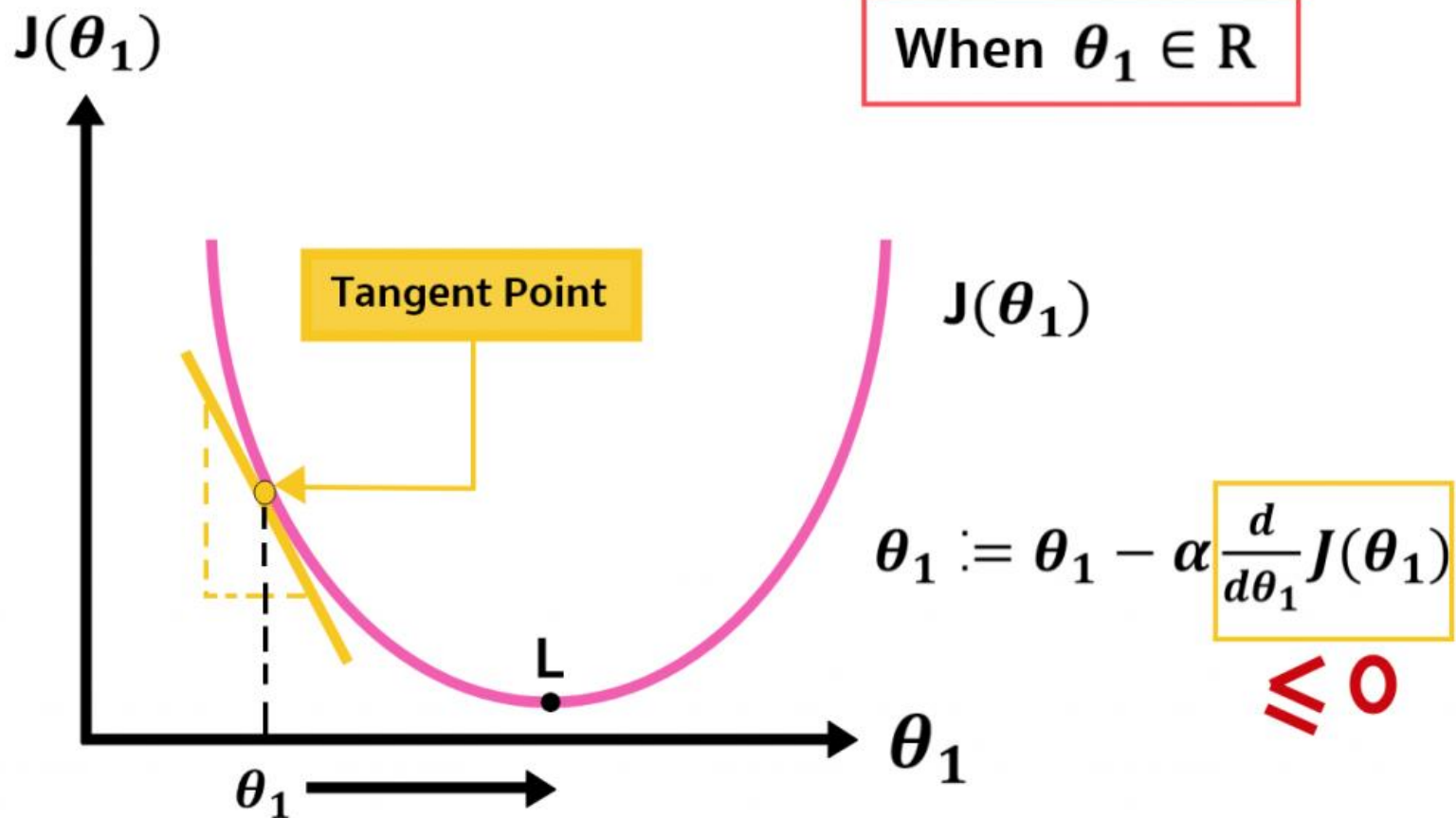
# Derivative

- คือ ความชันของเส้นตรงที่สัมผัสจุดใดๆบนเส้นโค้ง (Tangent Point) โดยทั่วไป ถ้าหากสมการเป็นเส้นตรง เราสามารถหาความชันเส้นตรงได้โดยใช้สูตร

$$\text{Slope} = \frac{Y_2 - Y_1}{X_2 - X_1}$$

- แต่ในกรณีนี้ เราต้องการหาความชันของเส้นโค้ง ดังนั้นเราต้องใช้การ Derivative หรือการ Diff

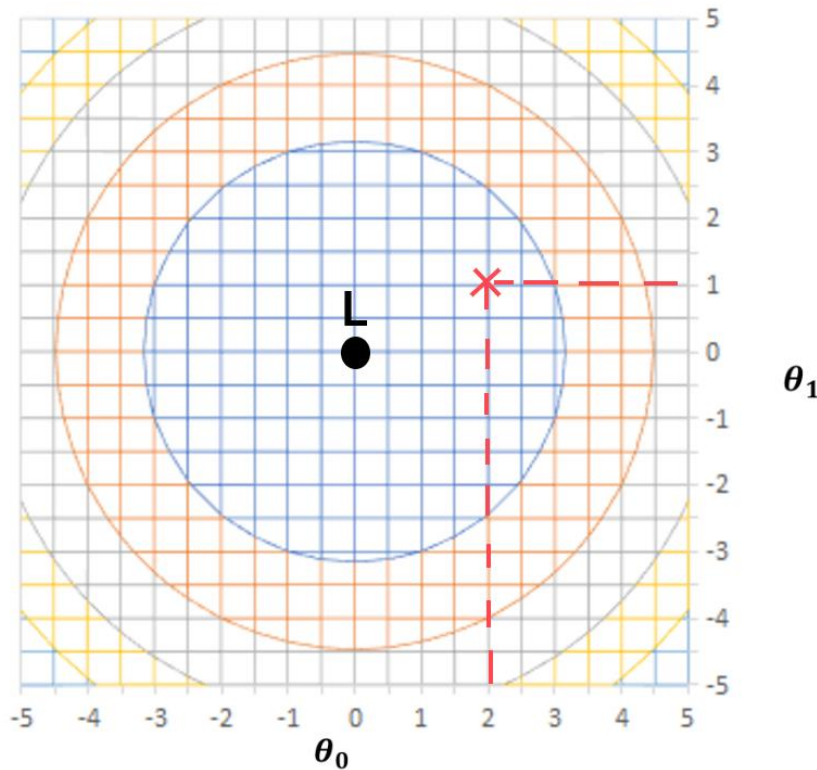








# Partial Derivative



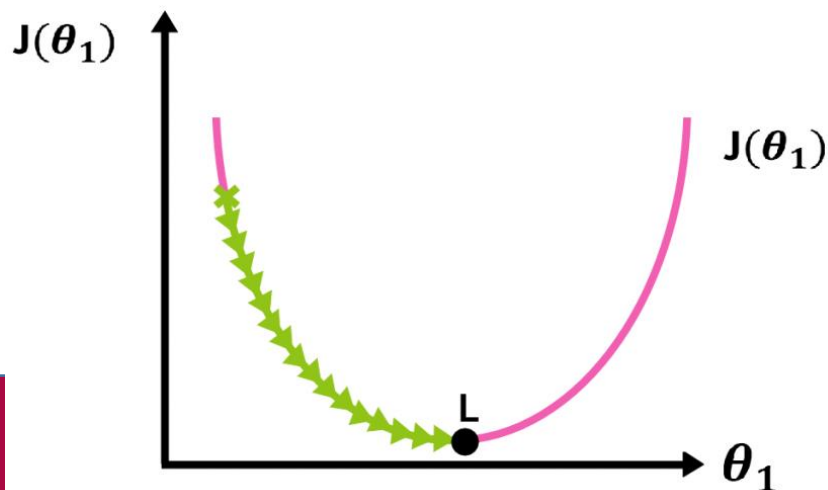
\*จะหยุดเมื่อ  $\text{Derivative}=0$  (Slope = 0)  
หรือ ถึงจุดที่ค่า  $\text{Derivative}$  มันเล็กมากๆ จน  
แทบไม่เปลี่ยนแปลงเราก็สามารถหยุดได้ ถึงแม้ว่า  
 $\text{Derivative}$  มันจะไม่เท่ากับ 0



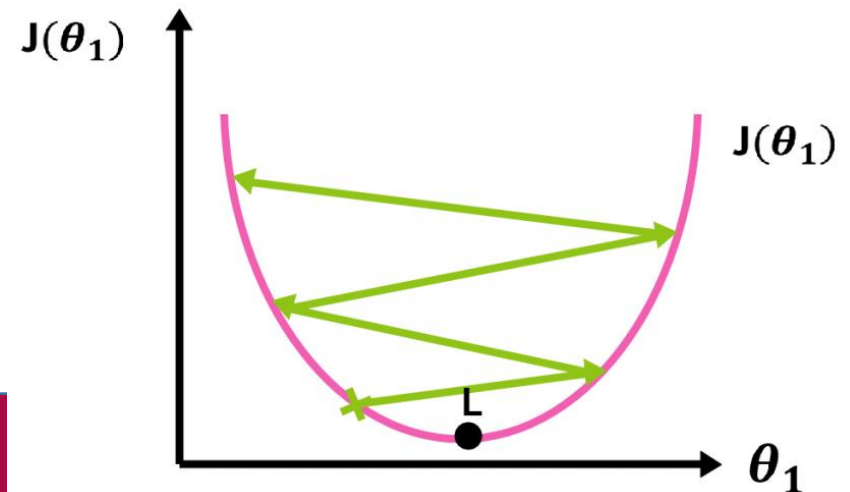
# Learning Rate

- Learning Rate ( $\alpha$ ) คือ ขนาดของการก้าว ซึ่งเป็นค่าคงที่ (Fixed Learning Rate)  
เมื่อเราได้มีการกำหนดค่า(สมมุติ)  $j=0, j=1$  แล้ว (Simultaneous Update)

Learning Rate ขนาดเล็ก -> ช้า



Learning Rate ขนาดใหญ่ -> Divergence





สรุป

สิ่งที่เรามี

Gradient Descent	Linear Regression
<p>Repeat until convergence</p> $\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$ <p>when <math>j=0</math> and <math>j=1</math></p>	$h_{\theta}(x^{(i)}) = \theta_0 + \theta_1 x^{(i)}$ $J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$ <p>แทนค่า</p>



1. คำนวณตามสูตร Gradient โดยคำนวณค่า **Y predict** จาก ค่า Slope ( $\widehat{\theta}_1$ ) และ Intercept ( $\widehat{\theta}_0$ ) ปัจจุบัน ( **Y predict** =  $\theta_0 + \theta_1 x^i$  )
2. คำนวณ Prediction errors (ความห่างระหว่างค่าที่ได้จากสมการประมาณการกับค่า Y จริง) จาก (**Y prediction** – Y)
3. Update the Intercept ( $\widehat{\theta}_0$ ) :
  - หาค่า Derivative of Intercept ( $\widehat{\theta}_0$ ) ได้จาก sum(errors) :
$$\sum_{i=1}^m ((\theta_0 + \theta_1 x^{(i)}) - y^{(i)})$$
  - หาค่าที่เปลี่ยนแปลงไป (Adjustment) : **Adjustment** = Learning rate \* **Derivative of intercept** ( $\widehat{\theta}_0$ )
  - หาคัดัดใหม่ (New Intercept) : **New Intercept** ( $\widehat{\theta}_0$ ) = intercept เดิม – ค่า **Adjustment**



#### 4. Update the slope ( $\widehat{\theta}_1$ ) :

- หาค่า Derivative of slope ( $\widehat{\theta}_1$ ) ได้จาก  $\text{sum}(\text{errors} * \text{input})$  :

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

- หาค่าที่เปลี่ยนแปลงไป (Adjustment) : **Adjustment = Learning rate \* Derivative of Slope**
- หาคความชันใหม่(New Slope) : **New Slope ( $\widehat{\theta}_1$ ) = Slope เดิม – ค่า Adjustment**



## สรุป (ต่อ)

- ทำจนกว่า Derivative จะเท่ากับที่ตั้งค่าไว้ เช่น 0
- แต่บางครั้งมันเสียเวลา เราเลยกำหนด จุด tolerance ที่ใกล้ 0 มาก ๆ แทน ดังนั้นเมื่อเราทำ Derivative ไปเรื่อยๆ จนค่า Magnitude มากกว่า ค่า Tolerance แสดงว่าเรา Convergence แล้ว วิธีนี้เรียกว่า Gradient Descent Approach
- โดยใช้ Magnitude คือ ขนาดของการ converge เข้าสู่จุด Local Minimum ซึ่งบางครั้งมันก็มีทั้งค่าที่เป็นบวกและเป็นลบ แต่ว่าเราไม่สนใจว่ามันจะเป็น -0.1 หรือ 0.1 เพราะมัน converge ทั้งคู่



4. Update the slope ( $\hat{\theta}_1$ ) :

- หาค่า Derivative of slope ( $\hat{\theta}_1$ ) ได้จาก  $\text{sum}(\text{errors} * \text{input})$  :

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

- หาค่าที่เปลี่ยนแปลงไป (Adjustment) : **Adjustment = Learning rate \* Derivative of Slope**

- หาค่าความชันใหม่(New Slope) : **New Slope ( $\hat{\theta}_1$ ) = Slope เดิม - ค่า Adjustment**

5. คำนวณหา Magnitude of the gradient

$$\sqrt{(\text{Derivative of slope})^2 + (\text{Derivative of intracept})^2}$$

6. ตรวจสอบว่ามันลู่เข้าแล้วหรือไม่ : ถ้า  $\text{Magnitude} < \text{Tolerance} = \text{Convergence!}$