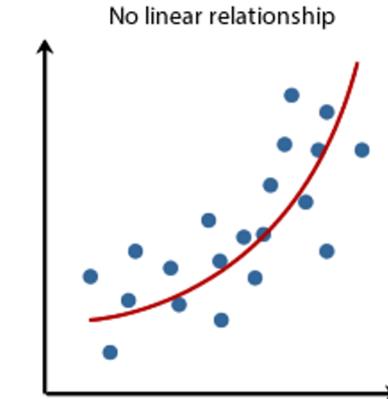
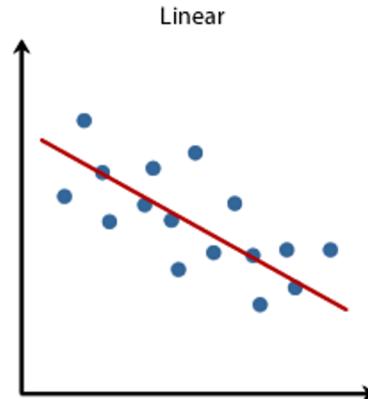
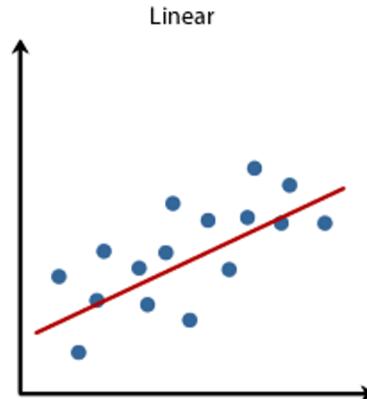


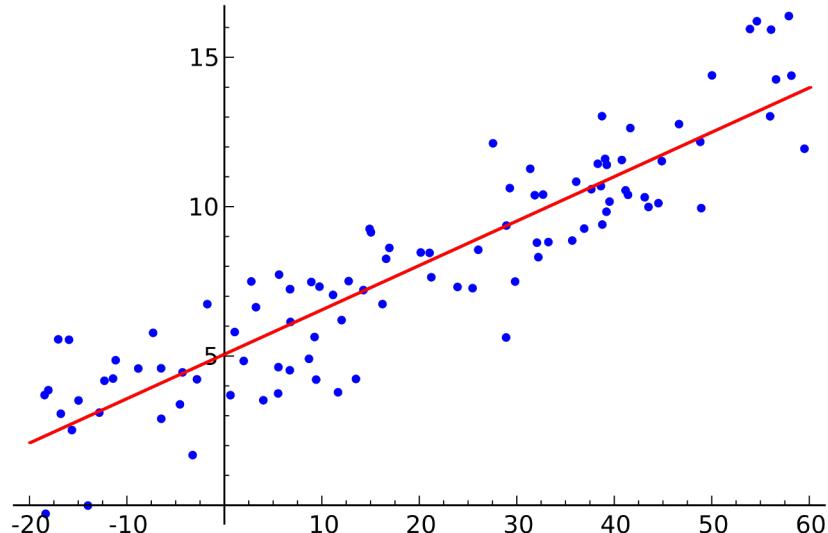
# Linear Regression

# Line of regression

- A line that can be taken as representative of the ideal variation is called as the line of best fit



# Simple Linear Regression



$$Y = mX + c$$

ชี้

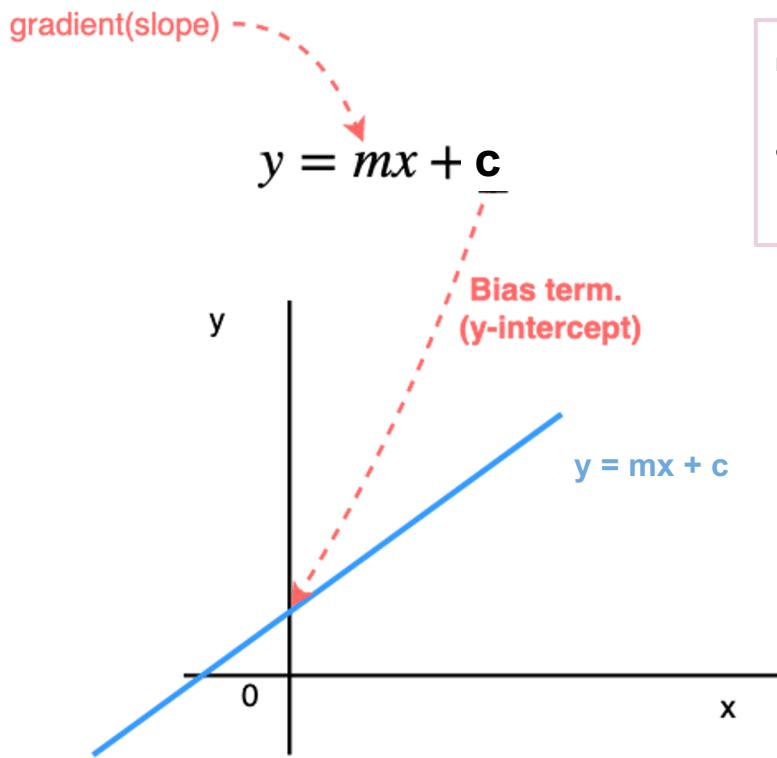
Y = ตัวแปรตาม

X = ตัวแปรต้น

m = ความชัน

c = จุดตัดแกน Y

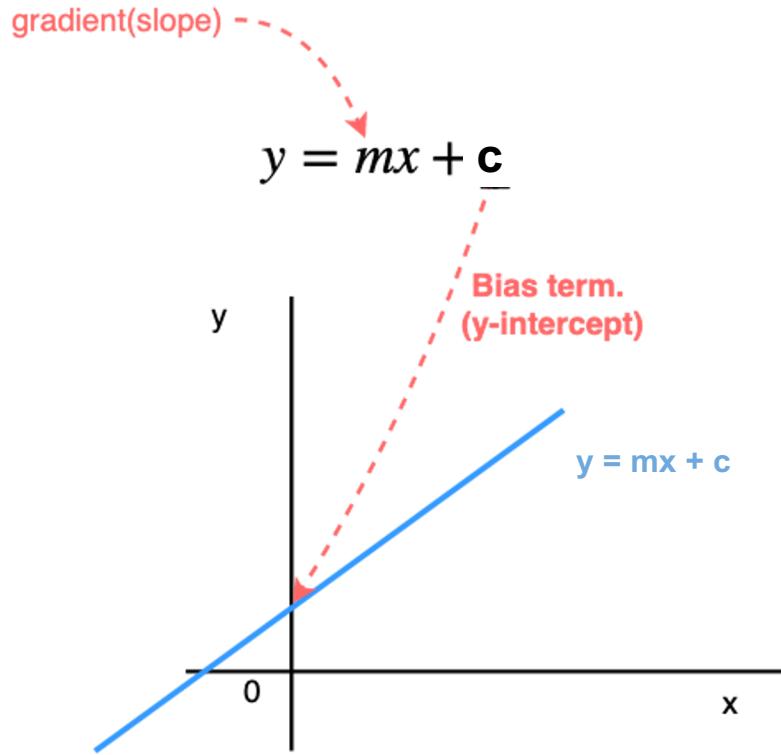
# Simple Linear Regression



ผลลัพธ์ที่ได้ คือ ผลคูณของค่า  $x$  กับ  $m$  และ บวกกับ bias  $c$

ดังนั้นเมื่อ  $x=0$  (ไม่มีข้อมูลเข้ามาในตัวแปรต้น) output จะมีค่าเท่ากับ bias  $c$ .

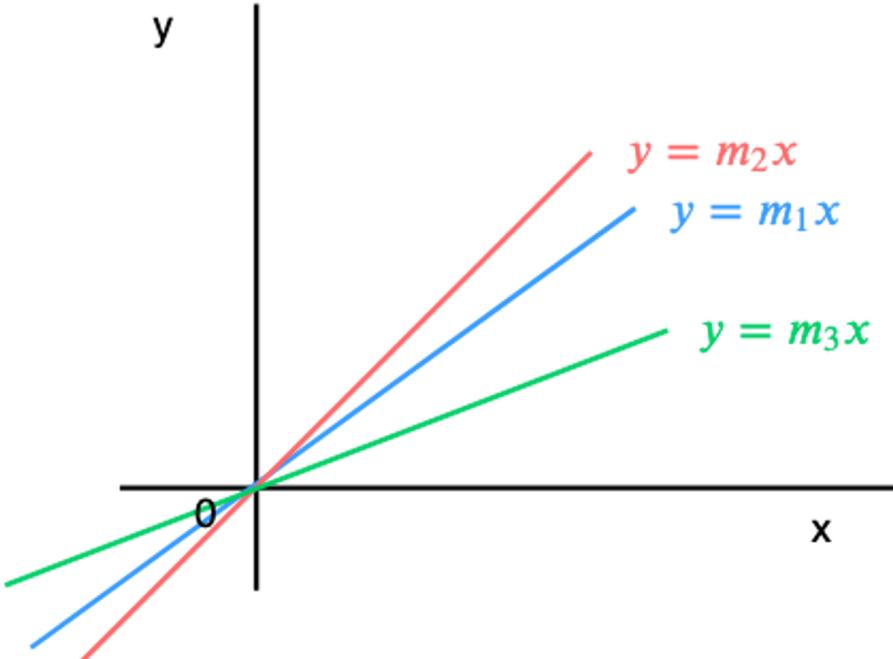
# Simple Linear Regression



$$Y = mX + c$$

$$h_{\theta}(x) = \theta_0 + \theta_1 x_1$$

# Simple Linear Regression



ถ้าหากไม่มี bias term นั้นหมายความว่า  
เส้นตรงจะตัดผ่านจุด origin  $(0,0)$  เสมอ<sup>2</sup>  
จะต่างเพียงแค่ความชัน  $m$  เท่านั้น

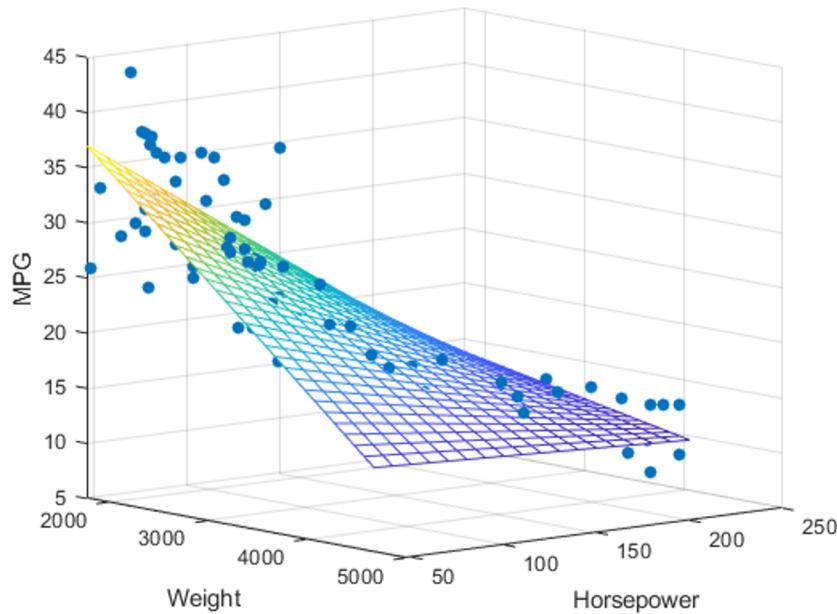
# Simple Linear Regression

Years of work	Salary
2.3	7,432 Bath
4.5	14,143 Bath
7.2	22,175 Bath
10.5	32,108 Bath
20.7	? Bath

Can we create a model that predict the amount of salary?

- What is the output ?
- What is the input (feature) ?

# Multivariate Linear Regression



$$y = m_1x_1 + m_2x_2 + c$$

# Multivariate Linear Regression

$$y = m_1x_1 + m_2x_2 + c$$



$$h_{\theta}(x) = \theta_0 + \theta_1x_1 + \theta_2x_2 + \theta_3x_3$$

# Multivariate Linear Regression

CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	PRICE
0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	396.90	4.98	24.0
0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	396.90	9.14	21.6
0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	392.83	4.03	34.7
0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	394.63	2.94	33.4
0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	396.90	5.33	36.2

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

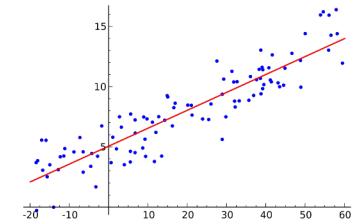
Where  $\theta_s$  are the parameter of the model

$X_s$  are values in the table

# Linear Regression

Simple

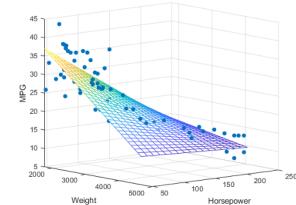
$$h_{\theta}(x) = \theta_0 + \theta_1 x_1$$



Multivariate

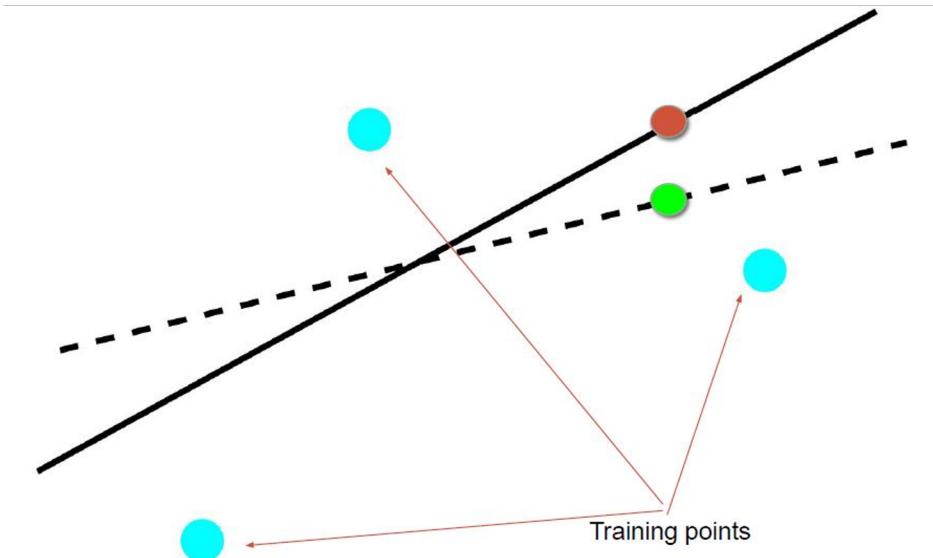
$$h_{\theta}(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \dots + \theta_n x_n$$

$$\begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}^T \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} = \theta^T x$$



# Linear Regression

$$h_{\theta}(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \dots + \theta_n x_n = \theta^T x$$



# Cost function (Loss function)

Loss function คือฟังก์ชันที่ทำหน้าที่คำนวณความผิดพลาดระหว่างข้อมูลที่ทำนายออกมากับค่าความเป็นจริง เพื่อนำไปใช้การปรับปรุงให้โมเดลพยากรณ์ที่เรียนปรับค่าที่ทำนายออกมากลับเข้าใกล้ค่าจริงมากยิ่งขึ้น

- Let's use the mean square error (MSE)

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m (y_i - \theta^T \mathbf{x}_i)^2$$



m is the number of training examples

i here is the index of the training example

Note how **x** is bolded

We want to pick  $\theta$  that minimize the loss

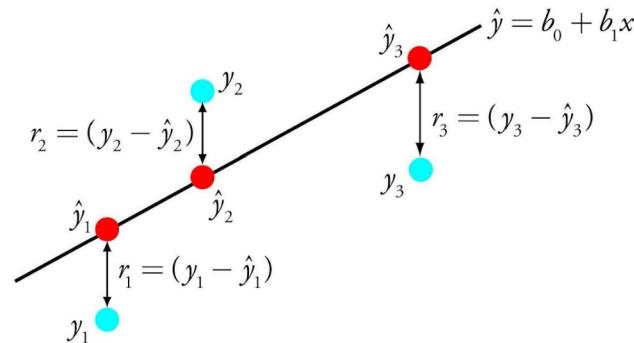
# Cost function (Loss function)

- Let's use the mean square error (MSE)

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m (y_i - \theta^T \mathbf{x}_i)^2$$



We want to pick  $\theta$  that minimize the loss



# Cost function (Loss function)

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m (y_i - \theta^T \mathbf{x}_i)^2$$

X <sub>1</sub>	X <sub>2</sub>	θ <sub>0</sub>	θ <sub>1</sub>	θ <sub>2</sub>	y_pred	y_real	loss
3	-7	-10	7	3	-10	-8	4
-2	4	-10	7	3	-12	-2	100
13	-8	-10	7	3	57	43	196
0	9	-10	7	3	17	12	25
-1	2	-10	7	3	3	5	4

SUM = 329

Average = 329/5 = 65.8

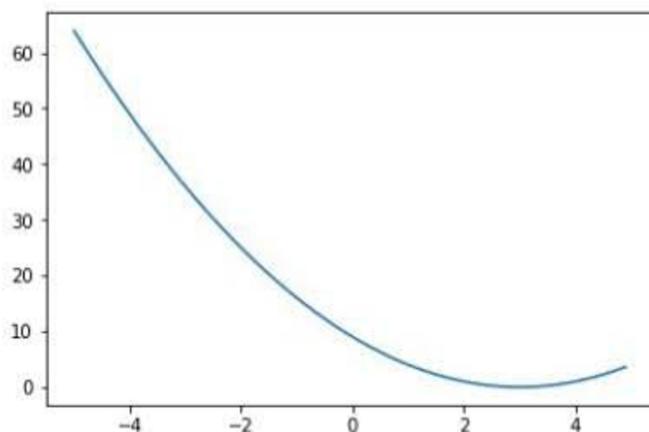
# Picking $\theta$

- Random until you get the best performance?
  - Can we do better than random chance?
- How to quantify best performance?

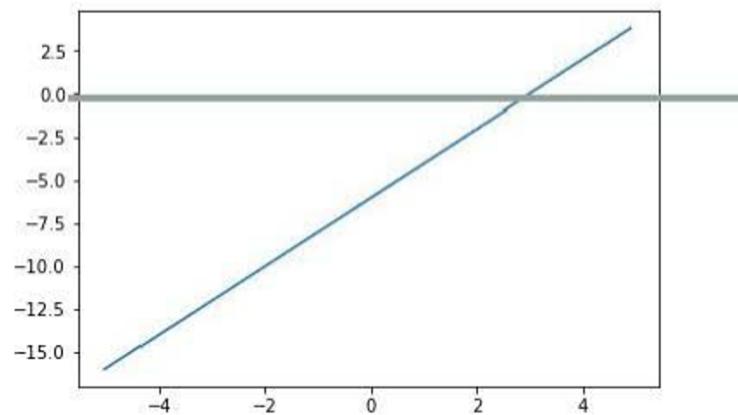
# Minimizing a function

- You have a function
  - $y = (x - a)^2$
- You want to minimize Y with respect to x
  - $dy/dx = 2x - 2a$
  - Take the derivative and set the derivative to 0
    - (And maybe check if it's a minima, maxima or saddle point)
- We can also go with an iterative approach

# Gradient descent



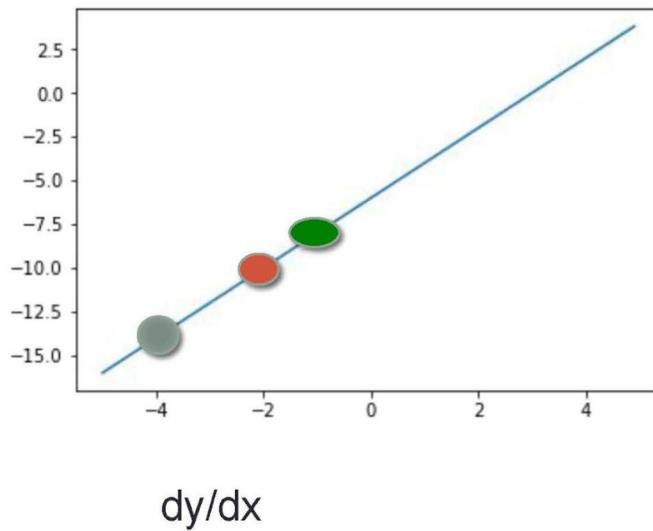
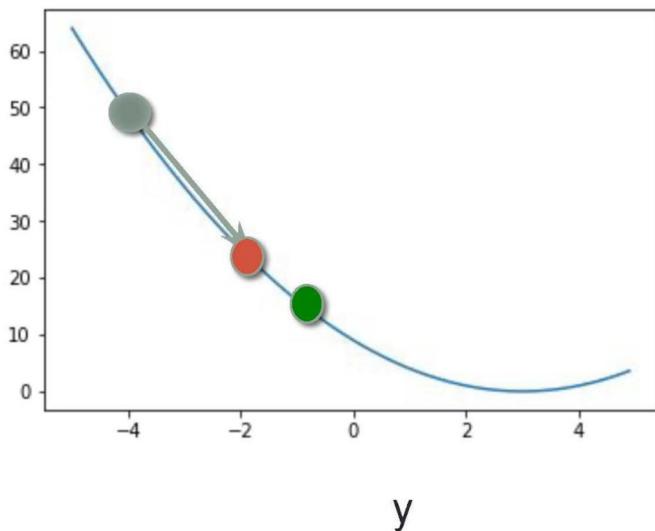
$y$



$dy/dx$

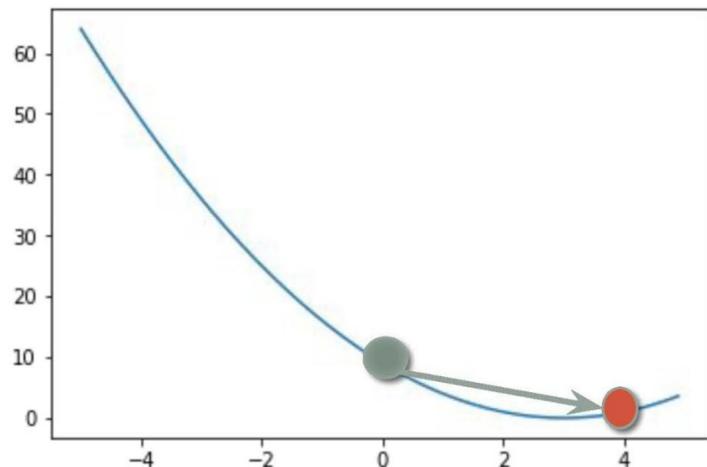
First what does  $dy/dx$  means?

# Gradient descent

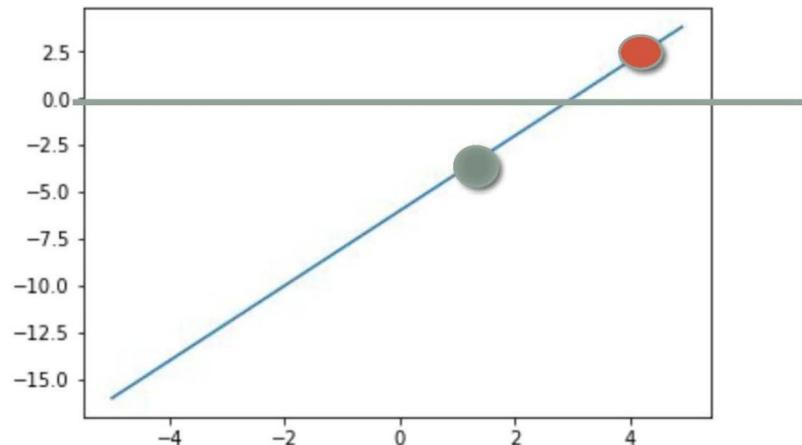


Move along the negative direction of the gradient  
The bigger the gradient the bigger step you move

# Gradient descent



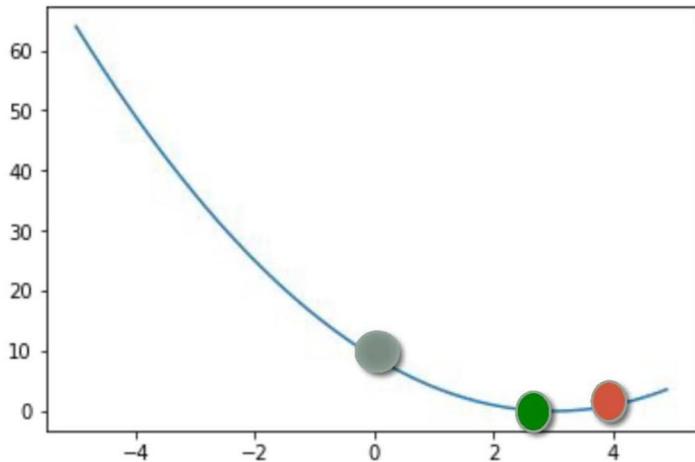
$y$



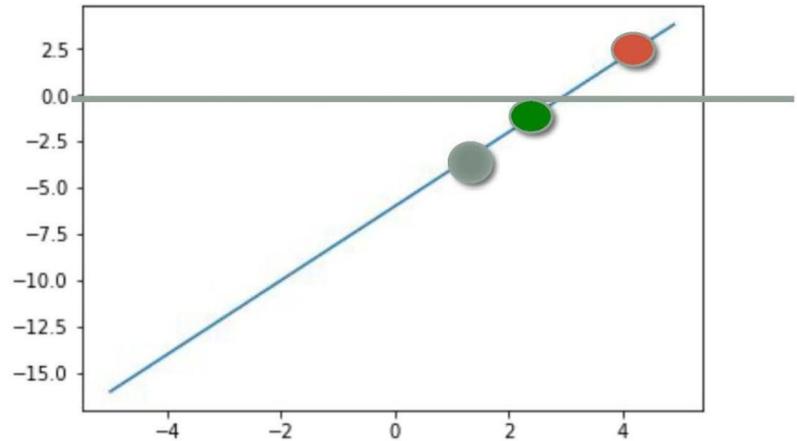
$dy/dx$

What happens when you overstep?

# Gradient descent



$y$



$dy/dx$

If you over step you can move back

# Backpropagation

## Simple Linear Regression

$$\frac{d\text{Loss}}{d\theta_0} = \sum_i h(\theta_0, \theta_1, x_{i,1}) - y_i$$

$$\frac{d\text{Loss}}{d\theta_1} = \sum_i (h(\theta_0, \theta_1, x_{i,1}) - y_i)^* x_{i,1}$$

# Derivatives

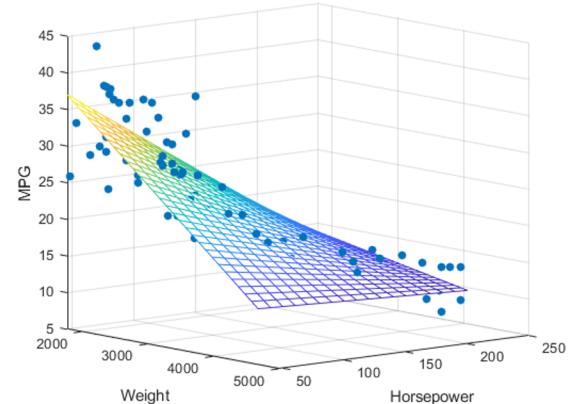
## Multivariate Linear Regression

$$\frac{d\text{Loss}}{d\theta_0} = \sum_i h(\theta_0, \theta_1, \theta_2, \theta_3, x_{i,1}, x_{i,2}, x_{i,3}) - y_i$$

$$\frac{d\text{Loss}}{d\theta_1} = \sum_i (h(\theta_0, \theta_1, \theta_2, \theta_3, x_{i,1}, x_{i,2}, x_{i,3}) - y_i) * x_{i,1}$$

$$\frac{d\text{Loss}}{d\theta_2} = \sum_i (h(\theta_0, \theta_1, \theta_2, \theta_3, x_{i,1}, x_{i,2}, x_{i,3}) - y_i) * x_{i,2}$$

$$\frac{d\text{Loss}}{d\theta_3} = \sum_i (h(\theta_0, \theta_1, \theta_2, \theta_3, x_{i,1}, x_{i,2}, x_{i,3}) - y_i) * x_{i,3}$$



$X_1$	$X_2$	$\theta_0$	$\theta_1$	$\theta_2$	$y_{\text{pred}}$	$y_{\text{real}}$	loss
3	-7	-10	7	3	-10	-8	4
-2	4	-10	7	3	-12	-2	100
13	-8	-10	7	3	57	43	196
0	9	-10	7	3	17	12	25
-1	2	-10	7	3	3	5	4

$X_{i,j}$

j គឺ ផែវវីរីលេខាដែលត្រូវមិន

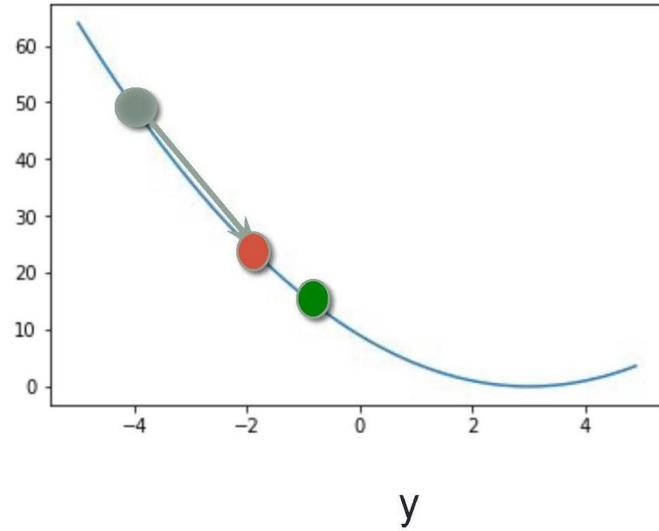
j គឺ គណន៍នៃវិសាលិនៅពេលចុះឈើមិន

# Update weight

$$\theta_0 \leftarrow \theta_0 - \alpha \frac{d\text{Loss}}{d\theta_0}$$

$\theta_0$  คือ ค่า weight ที่ต้องการปรับ

$\alpha$  (แอลฟ่า) คือ learning rate



# Learning rate

The steps which are taken to reach optimal point decides the rate of gradient descent.  
It is often referred to as 'Learning rate'

- **Too big**

bounce between the convex function and may not reach the local minimum

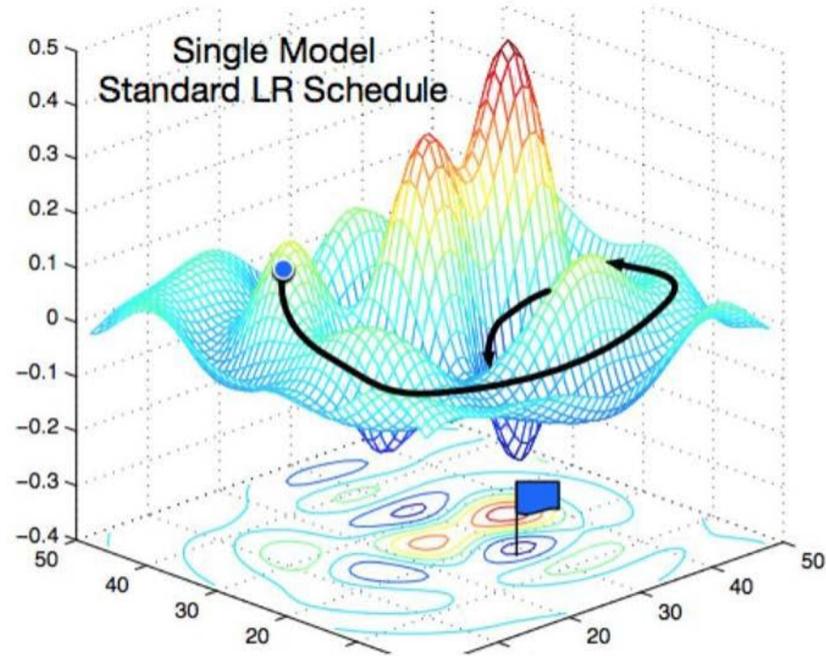
- **Too small**

gradient descent will eventually reach the local minimum but it will take too much time  
for that

- **Just right**

gradient descent will eventually reach the local minimum but it take too much time for  
that

# Gradient descent in 3D



<https://openreview.net/pdf?id=BJYwwY9II>

# Training Step

1. Choose hypothesis function.
2. Create loss function.
3. Calculate loss from hypothesis and ground truth.
4. Compute gradients from loss value.
5. Update weight.