① What is machie learning ?

exact, clear

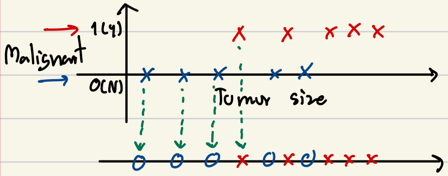
- Samel (1959) : Field of studying that give a computer ability to learn without being explicity programmed

- Mitchell (1998) : " A computer program is said to learn from Experience with respect to some Task if its performance on T and some Performance measure, as measured by P, improved by E.

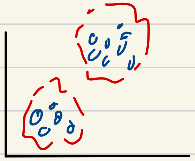
② Supervised learning → * right answer *

(fatal, deadly)

e.x. Breast cancer (Malignant , benign) → Classification problem
Discrete valued output
(0 or 1)
No cancer Cancer



③ Unsupervised learning



Learning without labelling → e.g. Social network analysis, Market segmentation

Cocktail party problem Algorithm → [W,s,v] = svd((repmat(sum(x.*x,1), size(x,1), 1).*x)*x');

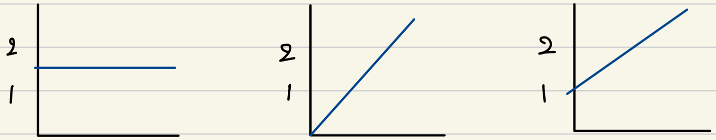
④ Quiz ✓

(5) Cost function

Hypothesis = $h_\theta(x) = \theta_0 + \theta_1 x$ ← Parameters

(Linear? → $y = mx + c$)

$\theta_0 = 1.5$, $\theta_1 = 0$ | $\theta_0 = 0$, $\theta_1 = 0.5$ | $\theta_0 = 1$, $\theta_1 = 0.5$



Regression problem : Choose θ_0, θ_1 so that $h_\theta(x)$ (Predicted) close to y (GT) for traing examples (x, y)

↳ Minimize θ_0, θ_1 ⟶ $\frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^i) - y^i\right)^2$ (m → Traing sets)

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

Cost function = Square error function

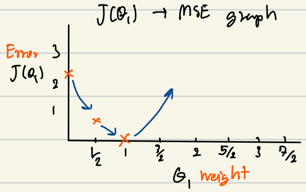
* $h_\theta(x)$ → Predicted | $\Sigma(h_\theta(x) - y)$ = MSE
  y → Real value |

5.1 Cost function : intution 1

Given Simplified / Simple Liner equation : $h_\theta(x) = \theta_1 x$; $\theta_0 = 0$

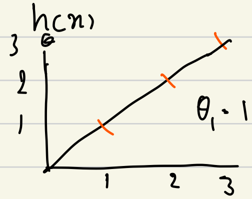
Cost function : $J(\theta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(\boxed{h_\theta(x^i)} - y^i\right)^2$

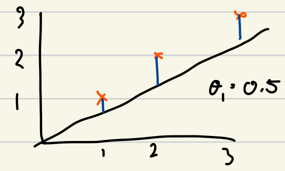
Goal Minimize : $J(\theta_1)$



Real value

x_1	y
1	1
2	2
3	3





$J(\theta_1) = \frac{1}{2m}\sum_{i}^{m}(h(x^i) - y^i)^2$

$J(1) = \frac{1}{2m}\sum_{1}^{3}(\theta_1 x^i - y^i)^2$

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

$J(0.5) = \{[(0.5\cdot1) - 1]^2 + [(0.5\cdot2) - 2]^2 + [(0.5\cdot3) - 3]^2\}\frac{1}{2\cdot3}$

$J(0.5) \approx 0.58$

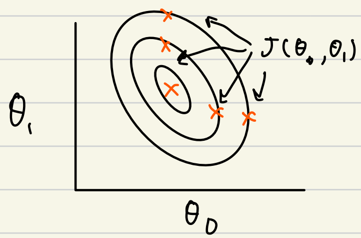
... $J(0) \approx 2.3$

5.2 Cost function : intution II

Given $h_\theta(x) = \theta_0 + \theta_1 x^i$

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

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



⑥ Gradient descent

Outline

① Start with random θ_0, θ_1

② Changing θ_0, θ_1 to reduce $J(\theta_0, \theta_1)$ until minimum

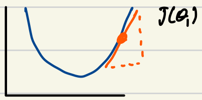
Gradient descent algorithm

repeat until convergence {

$$\theta_j := \theta_j - \underbrace{\alpha}_{\text{Learning rate}} \underbrace{\boxed{\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)}}_{\text{derivitive}} \quad (\text{for } j=0 \ \& \ j=1)$$

}

6.1 Gradient descent intuition 1

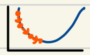


$\theta_1 := \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta_1)$

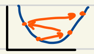
$\theta_1 := \theta_1 - \alpha(\text{positive number})$

$\theta_1 := \theta_1 - \alpha(\text{negative number})$

$\alpha =$ Step size → small



→ Large



When we found the minimum; $\theta_1 = 0$ → $\theta_1 := \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta_1)$

$\theta_1 = \theta_1 - 0$

$\theta_1 = \theta_1$

6.2 Gradient descent for Linear Regression → $h_\theta(x) = \theta_0 + \theta_1 x$

$y = mx + c$

$$\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) = \frac{\partial}{\partial \theta_j} \cdot \frac{1}{2m} \sum_1^m \left(h_\theta(x^i) - y^i\right)^2$$

$$= \frac{\partial}{\partial \theta_j} \cdot \frac{1}{2m} \sum_1^m \left(\theta_0 + \theta_1 x^i - y^i\right)^2$$

Partial diff

$$\theta_0 : c : j=0 \rightarrow \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) = \frac{\cancel{2}}{\cancel{2}m} \left(h_\theta(x^i) - y^i\right) \cdot \frac{\partial}{\partial \theta_0} \left(\theta_0 + \theta_1 x^i - y^i\right) = \frac{1}{m} \sum_1^m \left(h_\theta(x^i) - y^i\right)$$

$$\theta_1 : x : j=1 \rightarrow \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) = \frac{\cancel{2}}{\cancel{2}m} \left(h_\theta(x^i) - y^i\right) \cdot \frac{\partial}{\partial \theta_1} \left(\theta_0 + \theta_1 x^i - y^i\right) = \frac{1}{m} \sum_1^m \left(h_\theta(x^i) - y^i\right) \cdot x^i$$

Use these to update

θ_0, θ_1

b, w

"Batch Gradient Descent"

- Batch : Each step of gradient descent uses all the training example.