2019成大數學建模黑客松

用數學打造人工智慧

蔡炎龍 政治大學應用數學系

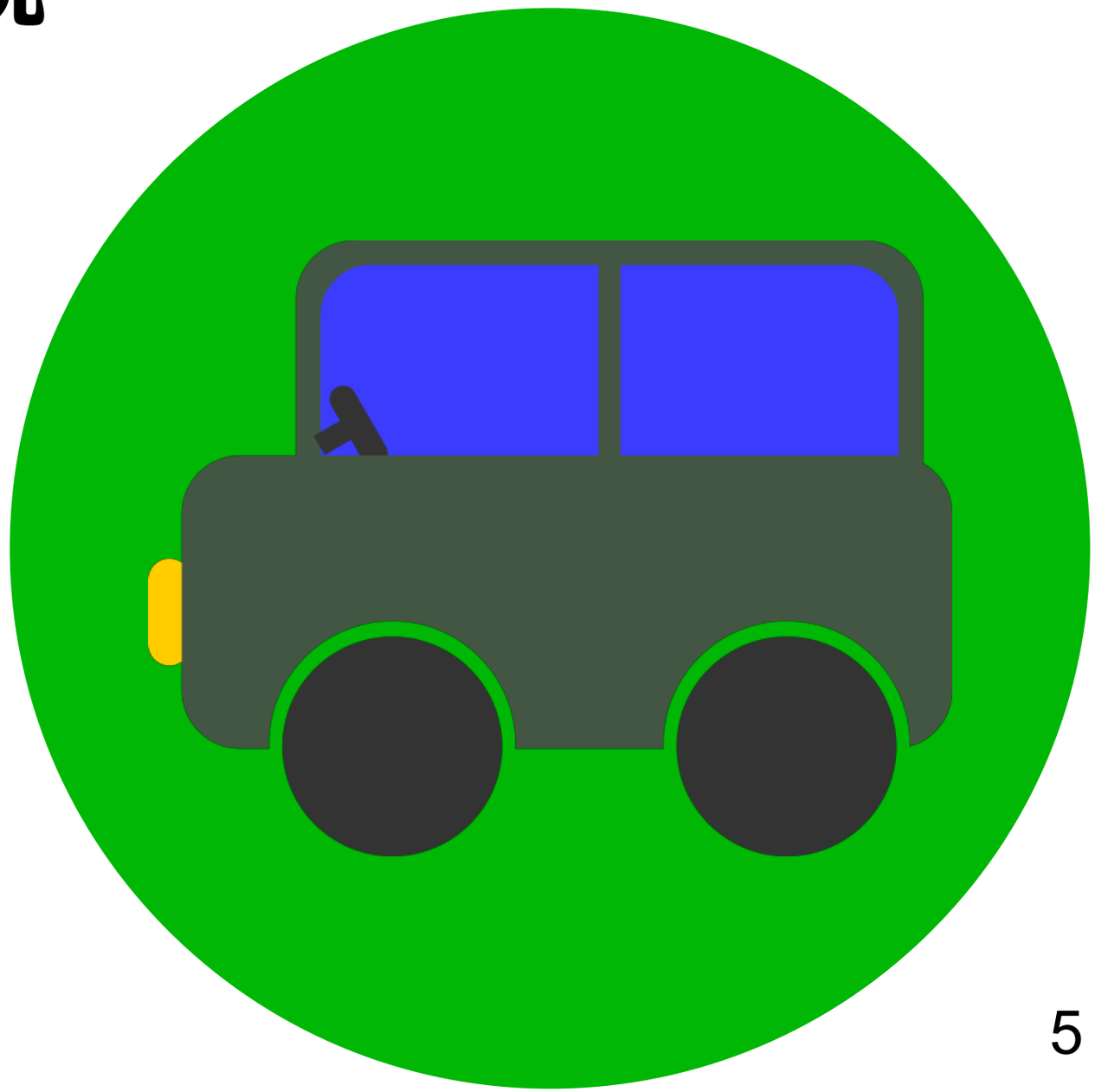
Facebook 的 AI 首席科學家 Yann LeCun 建議同學, 如果你對 AI 有興趣, 你需要...

數學、數學, 噢, 也許
更多的數學!

人類會被 AI 取代嗎?

事實上也沒那麼可怕

近 100% 機會會實現



近期內不可能

不過我們還有 94 年的時間

2112 年 9 月 3 日

假設今天你在路上撿到一個神燈...

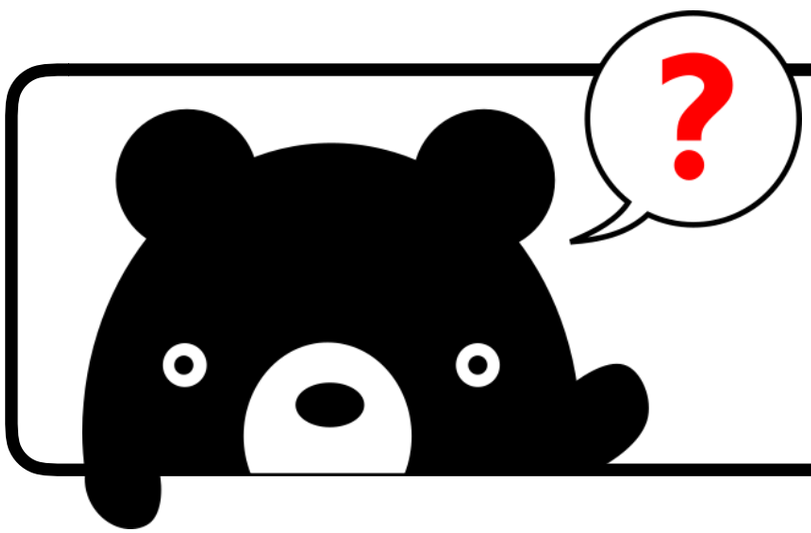
你擦拭了以後果然如傳說出現一位巨人...

為了報答你, 他給你兩個選擇, 你只能選一個...

說好的三個願望呢?

1 任意函數生成器

2 現金三千萬



你會選哪個

當然要選任意函數生成器!!

WHY?

因為我們是數學黑客松啊

函數是什麼呢?

定義

函數

一個函數 f 是由某個集合 X 到另一個集合 Y 的對映關係。x 對映到 y 我們記為 f(x) = y。要符合兩個條件...

定義

函數

條件
1

每一個在 X (我們稱為定義域) 中的元素 x, 都有一個在 Y (我們稱為值域) 裡的元素 y, 使得 f(x) = y。

定義

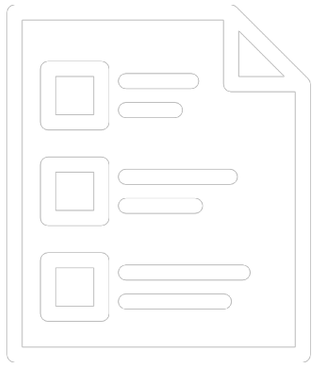
函數

條件
2

如果今天有在 X 中的兩個元素 x_1, x_2, 且 $x_1=x_2$, 則我們有

$$f(x_1) = f(x_2)$$

可以用人話說一次嗎?



函數其實是一個解答本

定義

函數

所有可能的問題

一個函數 f 是由某個集合 X 到另一個集合 Y 的對映關係。x 對映到 y 我們記為 f(x) = y。要符合兩個條件...

所有可能的答案
如 {A, B, C, D, E}

定義 函數

條件 1

每一個在 X (我們稱為定義域) 中的元素 x, 都有一個在 Y (我們稱為值域) 裡的元素 y, 使得 f(x) = y。

每個問題都要有答案!

定義 函數

條件 2

如果今天有在 X 中的兩個元素 x_1, x_2, 且 $x_1=x_2$, 則我們有

$$f(x_1) = f(x_2)$$

每個問題就一個答案!

活生生的例子

例子

股票點數預測

我想知道某支股票明天的收盤價。

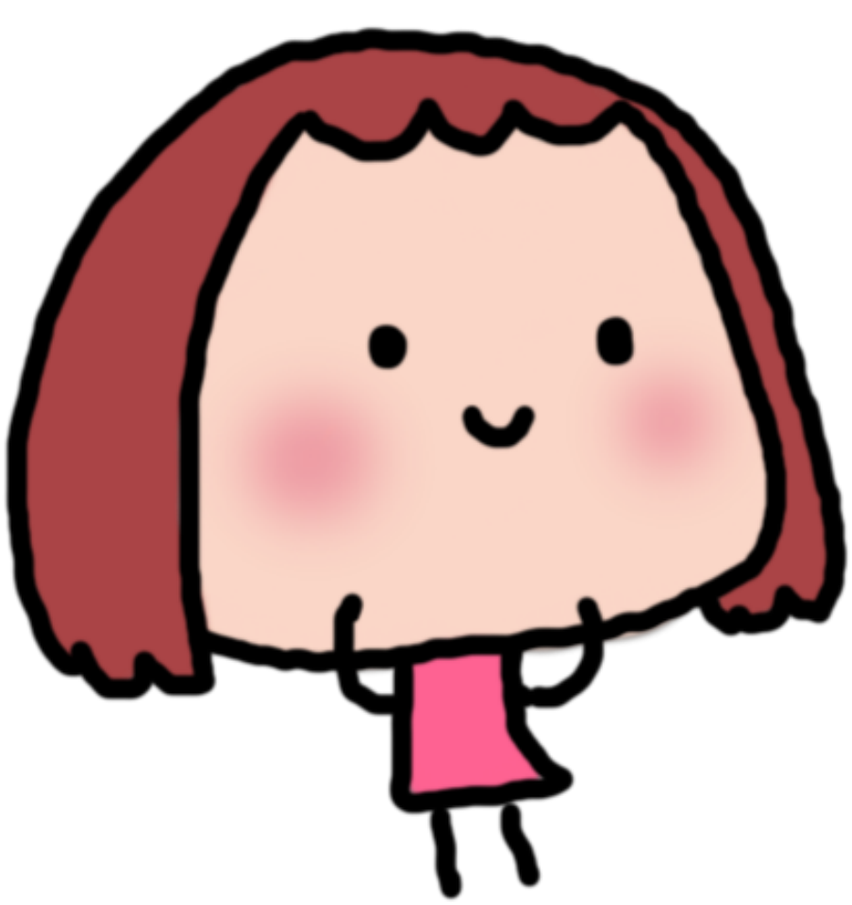
日期 x → f → 某股票 x 當天的收盤價

NN, CNN, RNN

$x_{t-1}, x_{t-2}, x_{t-3},$
x_{t-4}, x_{t-5}

f

x_t



用前 1 週的情況預測下一期。

其實還有無數種問法!

我想知道某位 MLB 選手 2019 球季可以打幾隻全壘打?

例子

全壘打數預測

15 個 features!

[Age, G, PA, AB, R, H, 2B, 3B, HR,

RBI, SB, BB, SO, OPS+, TB]

第 t-1 年資料 → f → **第 t 年全壘打數**

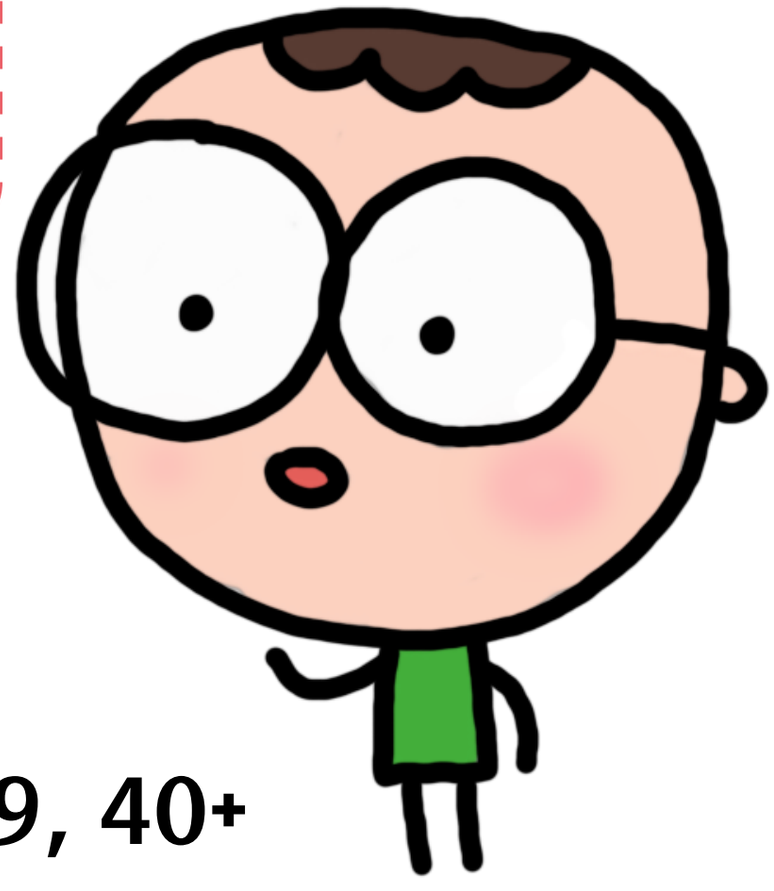
RNN

15 個 features!

[Age, G, PA, AB, R, H, 2B, 3B, HR,

RBI, SB, BB, SO, OPS+, TB]

不要猜精確數目, 猜區間即可!



分五段: 0-9, 10-19, 20-29, 30-39, 40+

2017 預測結果

(2017 年 6 月預測)

Mike Trout (LAA)
預測 30-39
實際 33

Kris Bryant (CHC)
預測 30-39 (第二高 20-29)
實際 29

Mookie Betts (BOS)
預測 20-29
實際 24

Daniel Murphy (WSH)
預測 20-29
實際 23

Jose Altuve (HOU)
預測 20-29
實際 24

Corey Seager (LAD)
預測 20-29
實際 22

我喜歡的字型有缺字, 我想要這個字!

例子

電腦造字

字型A

字型B

CNN, VAE, GAN

總之我們就是要函數

$$f : \mathbb{R}^n \to \mathbb{R}^m$$

很數學的例子

- **R 到 R 的函數**

$$f(x) = x^3 - 2x + 5$$

- **R^2 到 R 的函數**

$$f(x, y) = 3xy - 2x^2 - 3y^3$$

- **R 到 R^2 的函數**

$$f(x) = (2x^2 - 3x + 1, x - 2)$$

$f_1(x)$ $f_2(x)$

函數學習三部曲

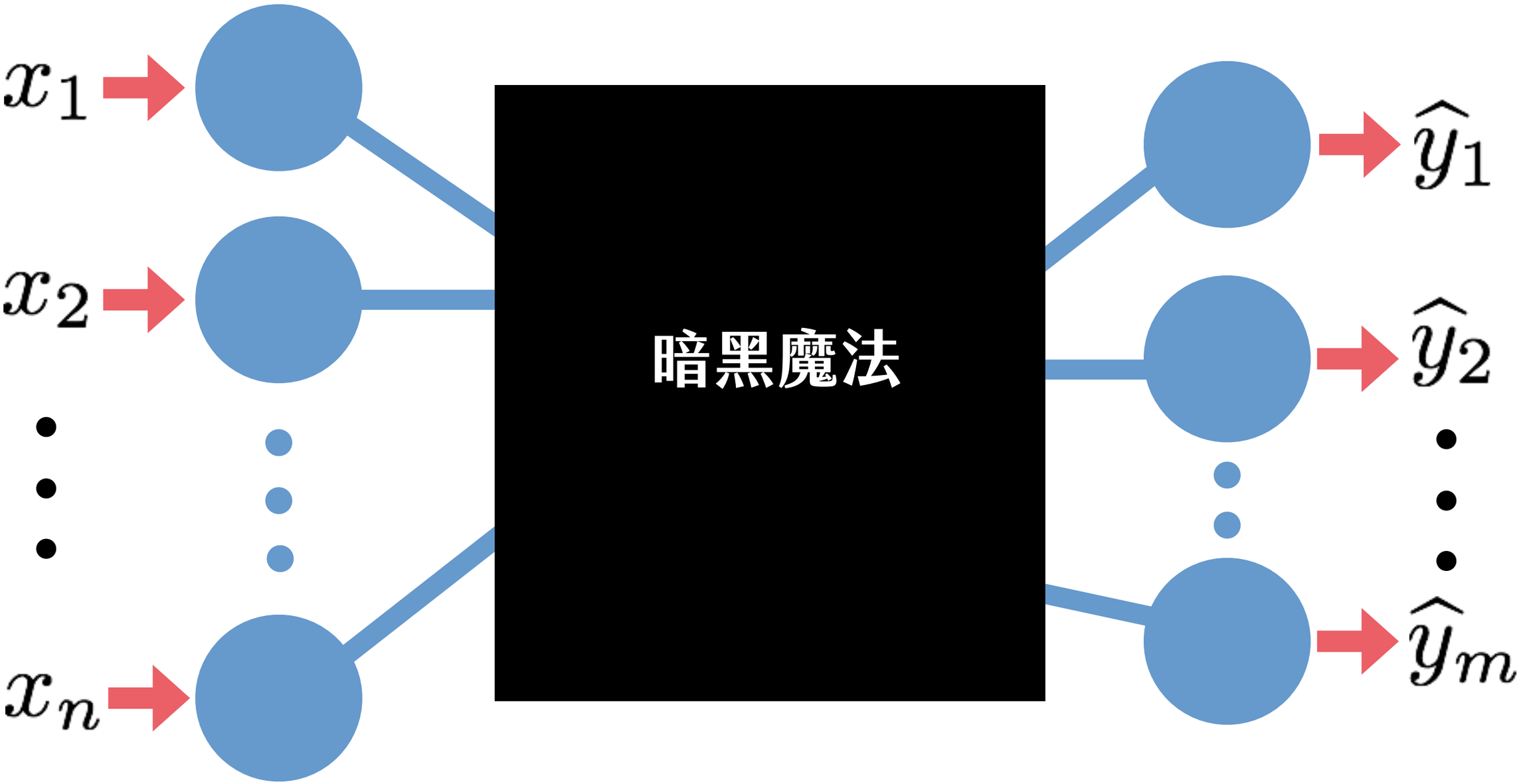
- 真實世界我們要問的問題化做函數。
- 收集我們知道「正確答案」的訓練資料。
- 找出這個函數!

暗黑學習法

真的有學任意函數的技法

就是「神經網路」！

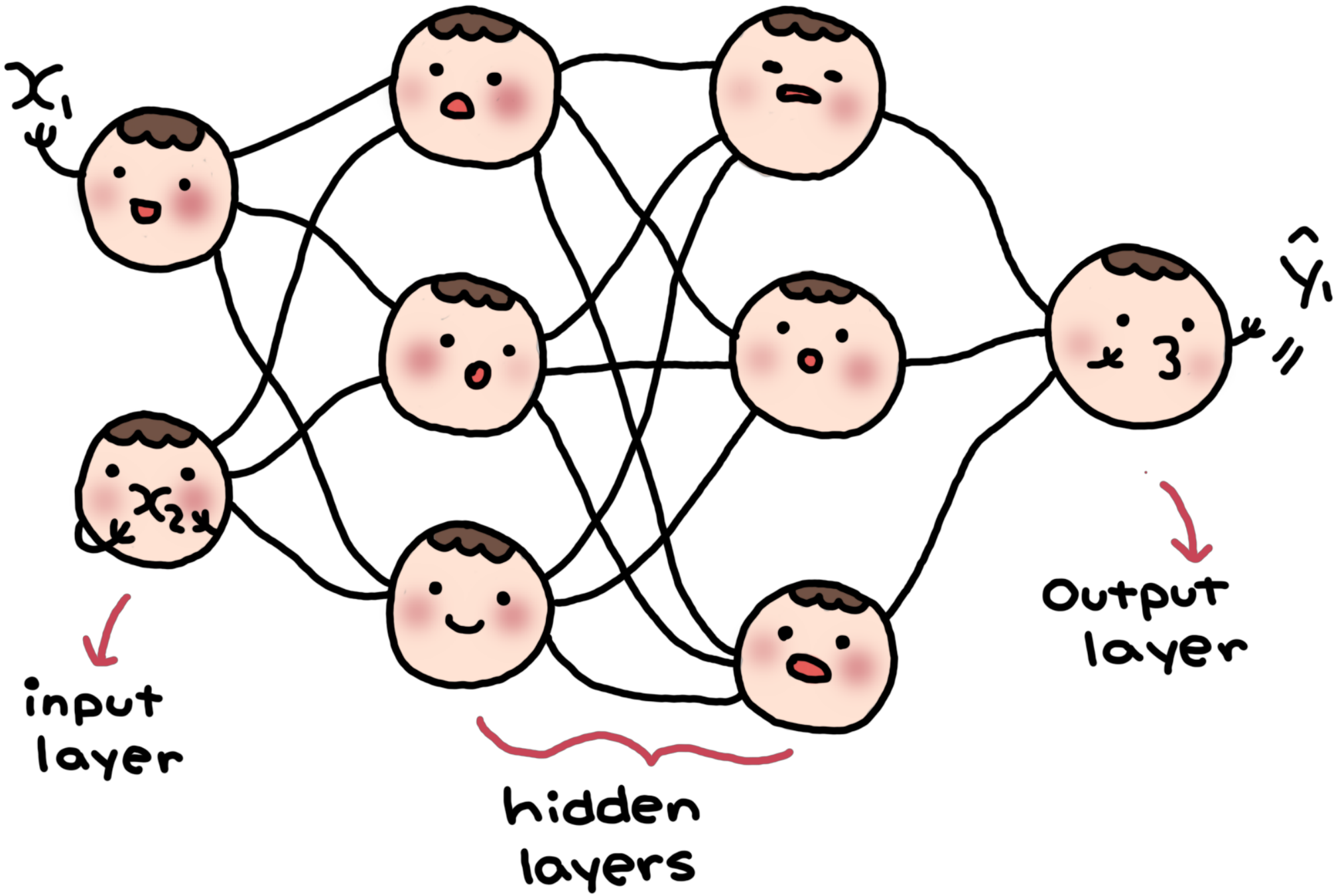
在 1980-1990 左右是很潮的東西


x_1
x_2
x_n
暗黑魔法
$\hat{y}_1$
$\hat{y}_2$
$\hat{y}_m$
Input Layer
Hidden Layer
Output Layer

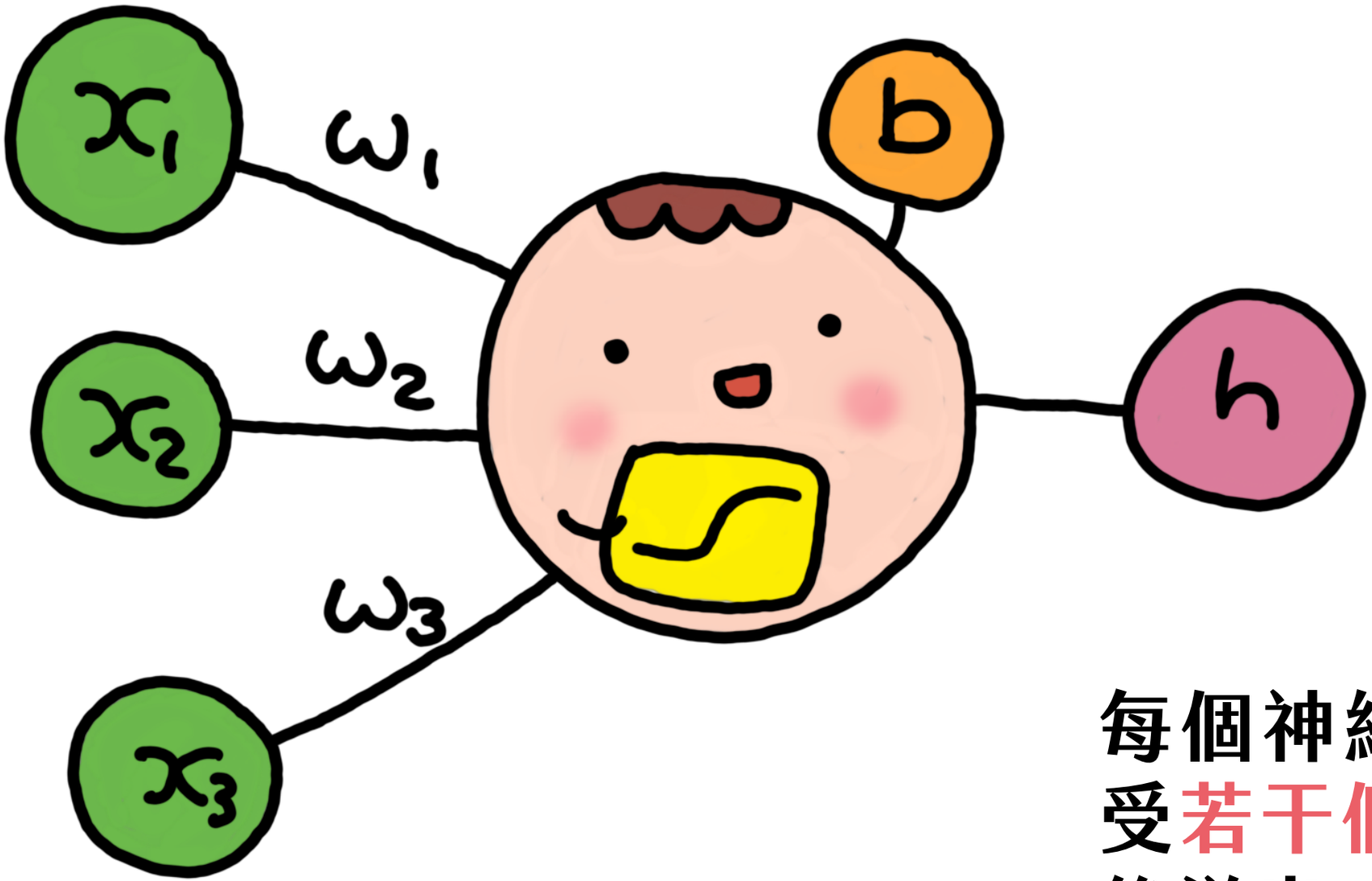
厲害的是神經網路什麼都學得會!

而且你完全不用告訴它函數應該長什麼樣子: 線性啦、二次多項式啦等等。

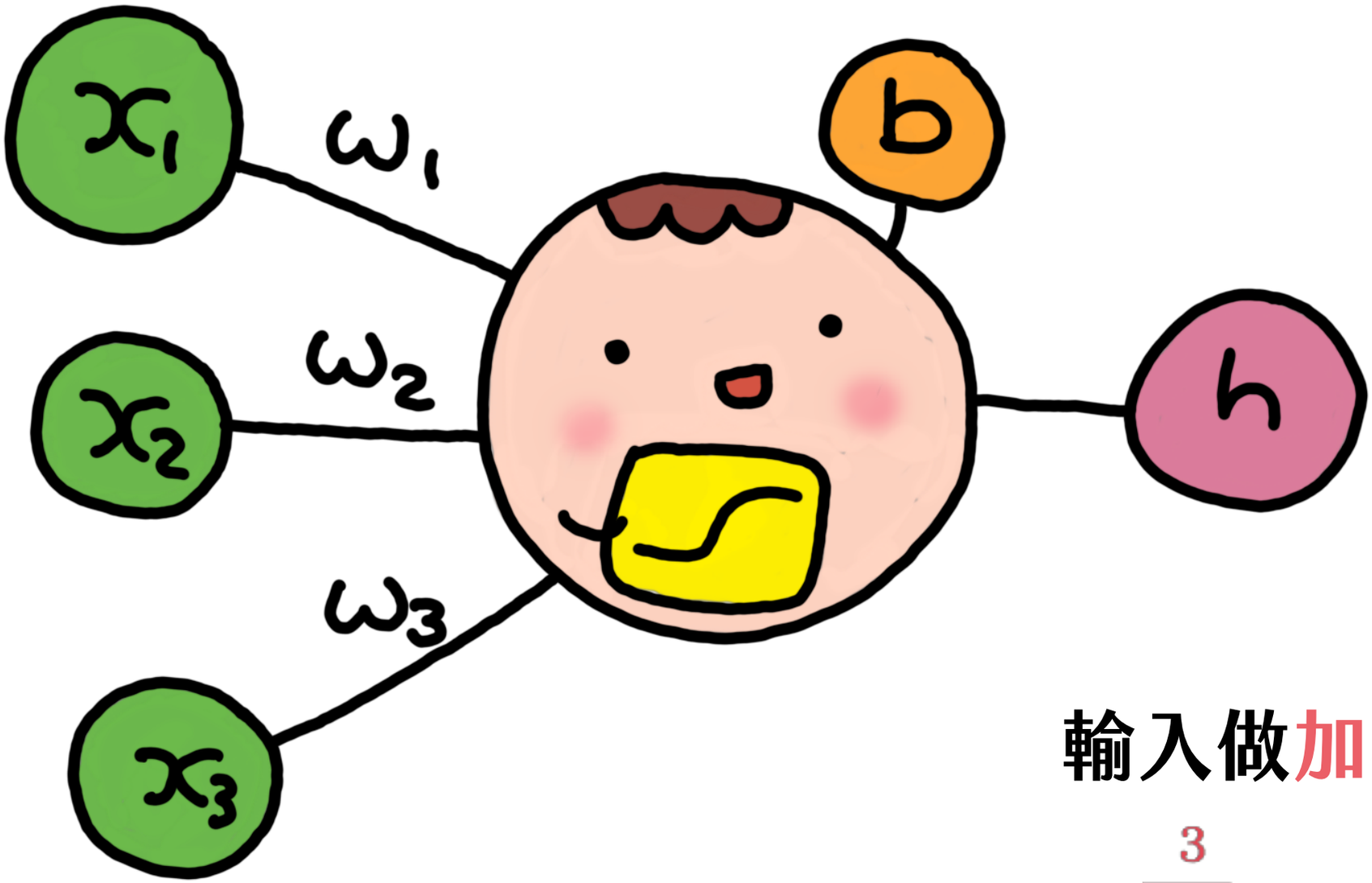
打開暗黑世界


x_1
x_2
$\hat{y}$
input layer
hidden layers
Output layer

每個神經元動作基本上是一樣的!

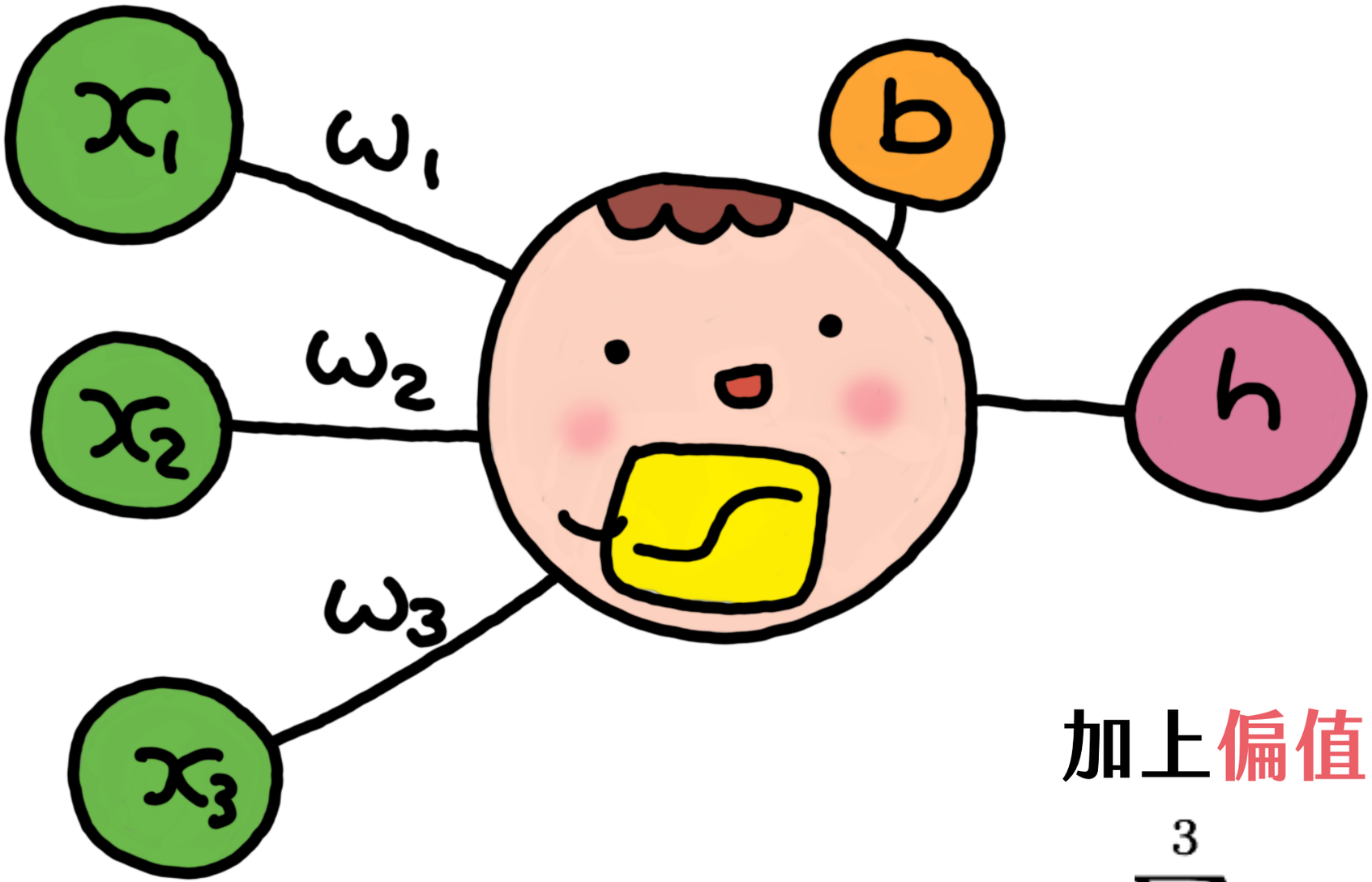


每個神經元就是接受若干個輸入，然後送出一個輸出。



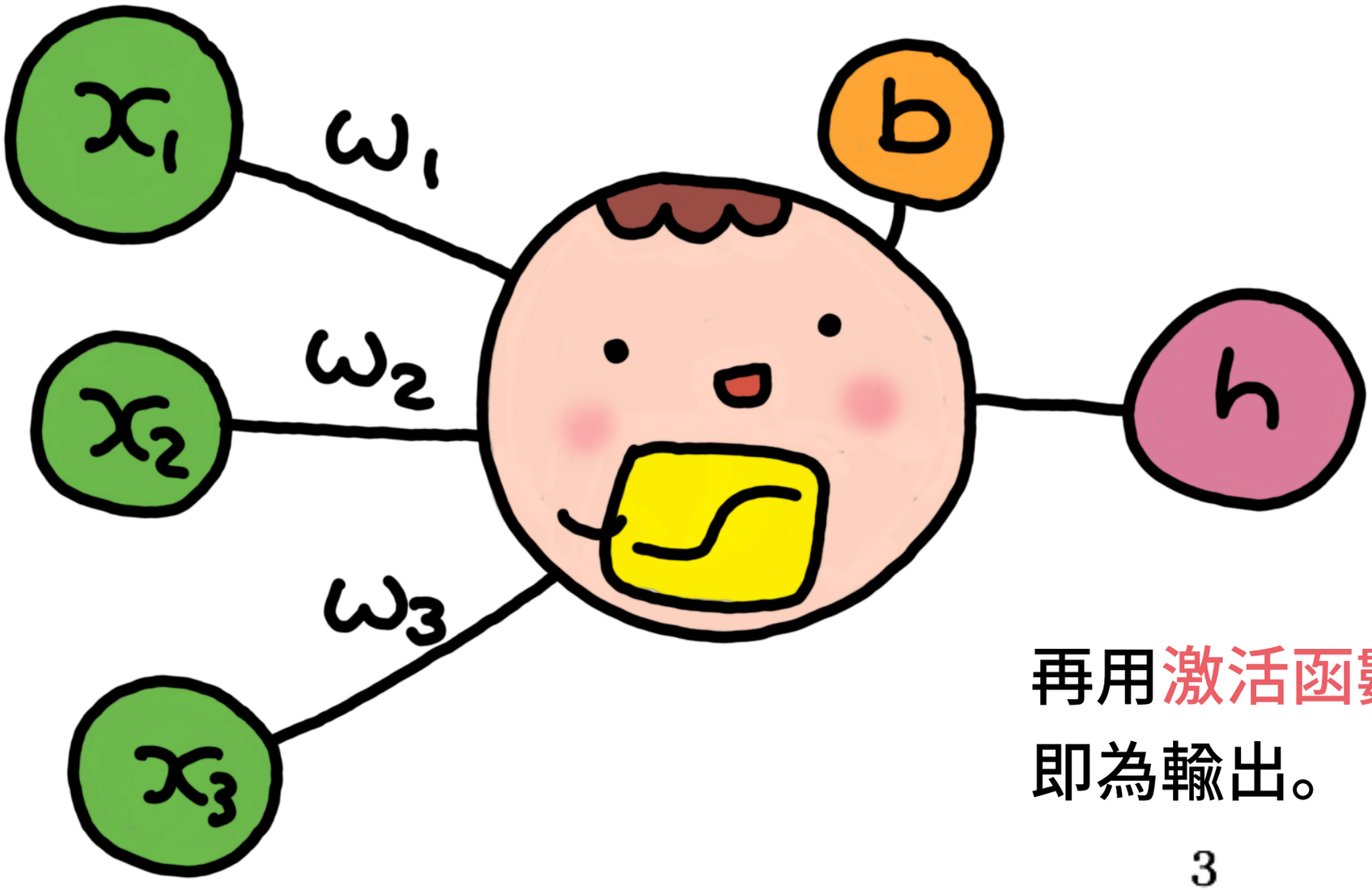
輸入做**加權和**。

$$\sum_{i=1}^{3} w_i x_i$$



加上偏值 (bias)。

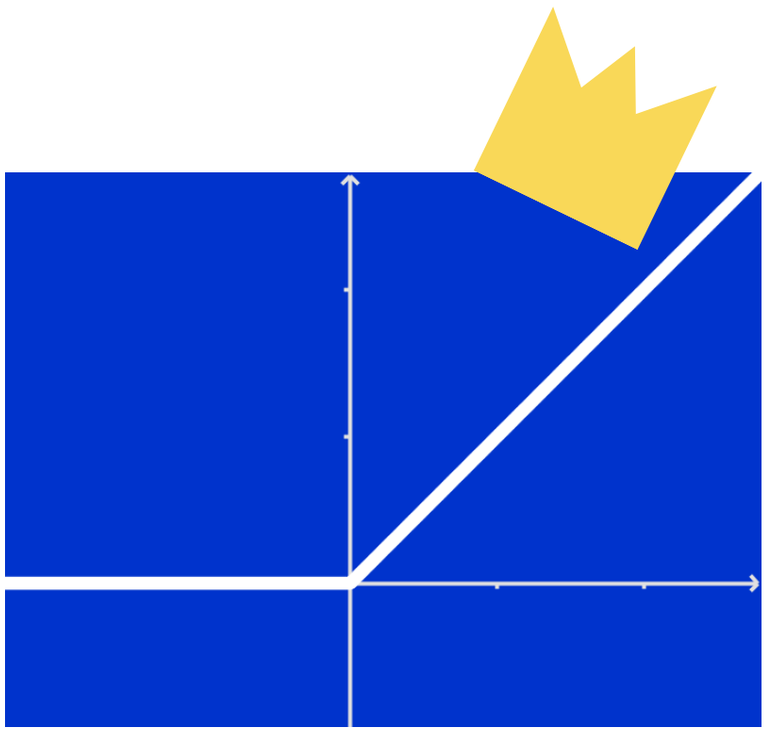
$$\sum_{i=1}^{3} w_i x_i + b$$



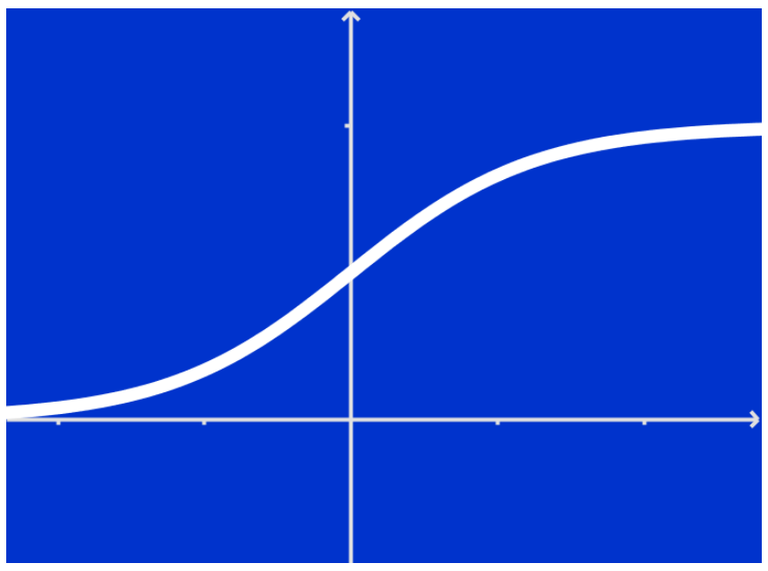
再用激活函數作用上去，即為輸出。

$$\varphi(\sum_{i=1}^{3} w_i x_i + b) = h$$

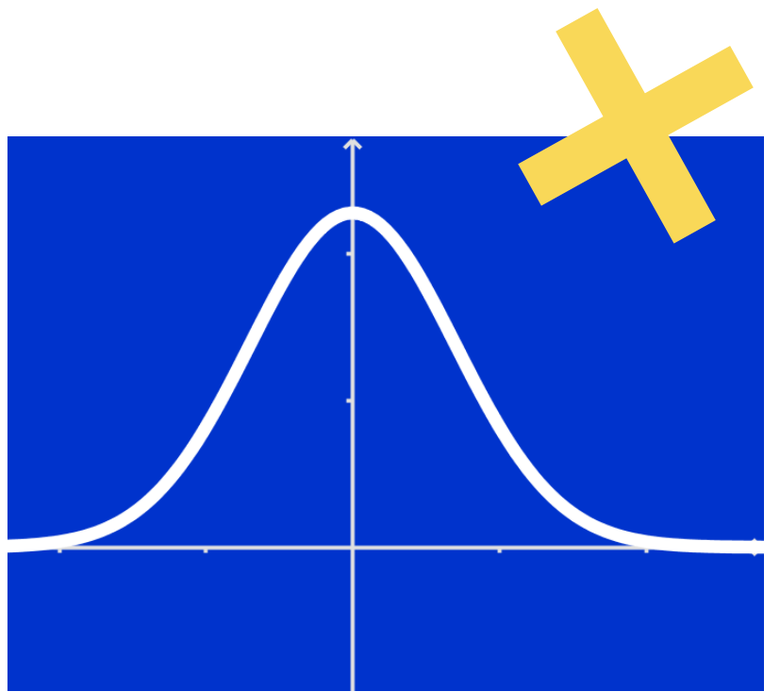
幾個 activation functions



ReLU

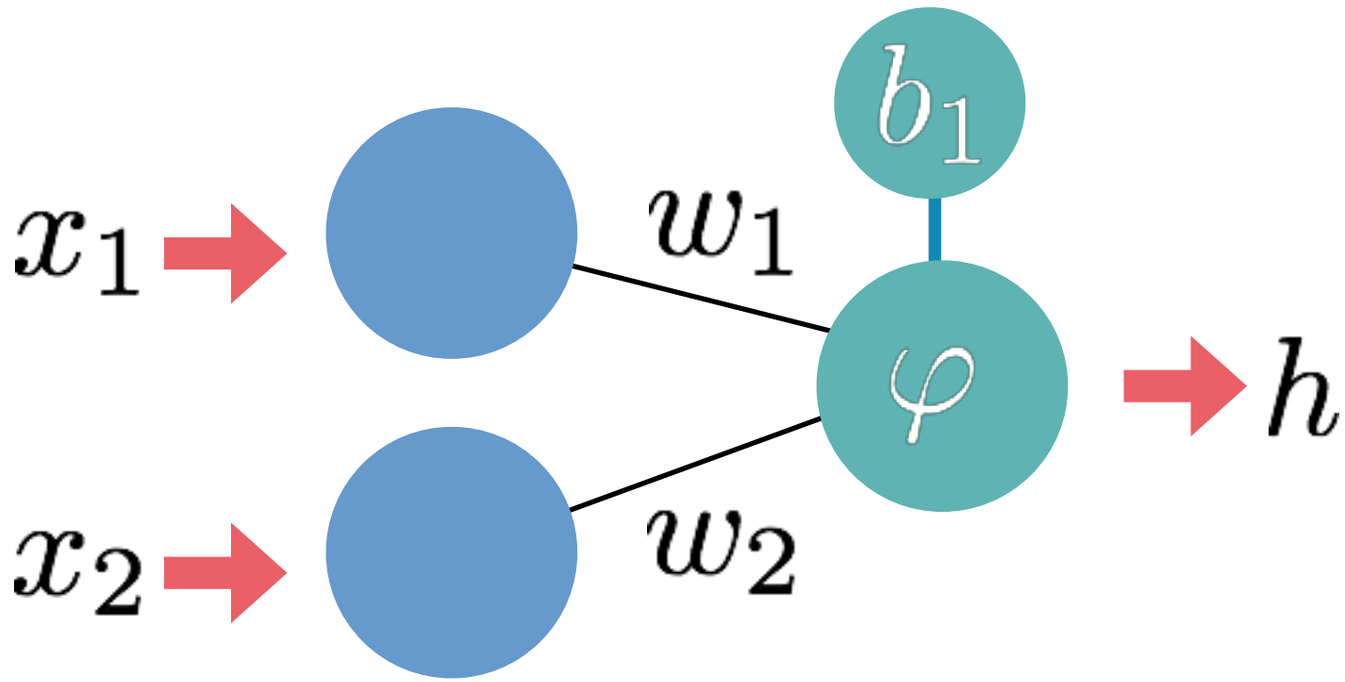


Sigmoid



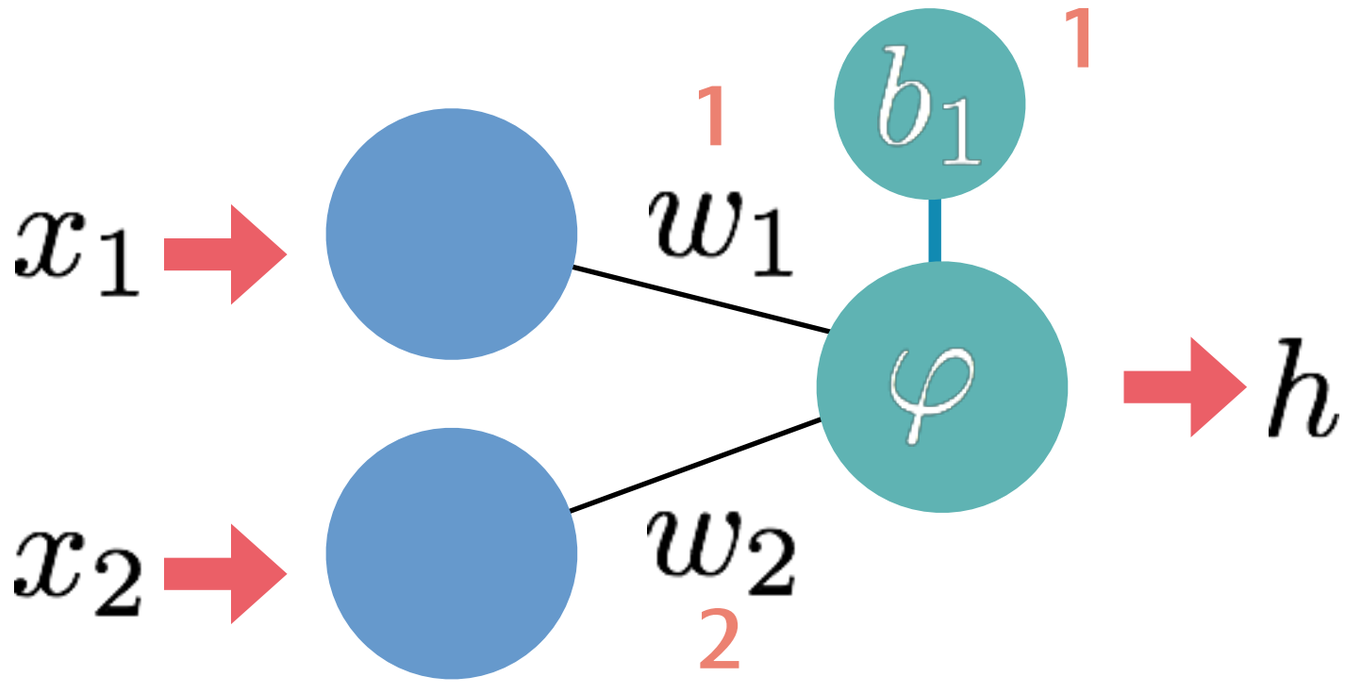
Gaussian

變數就是 weights, biases



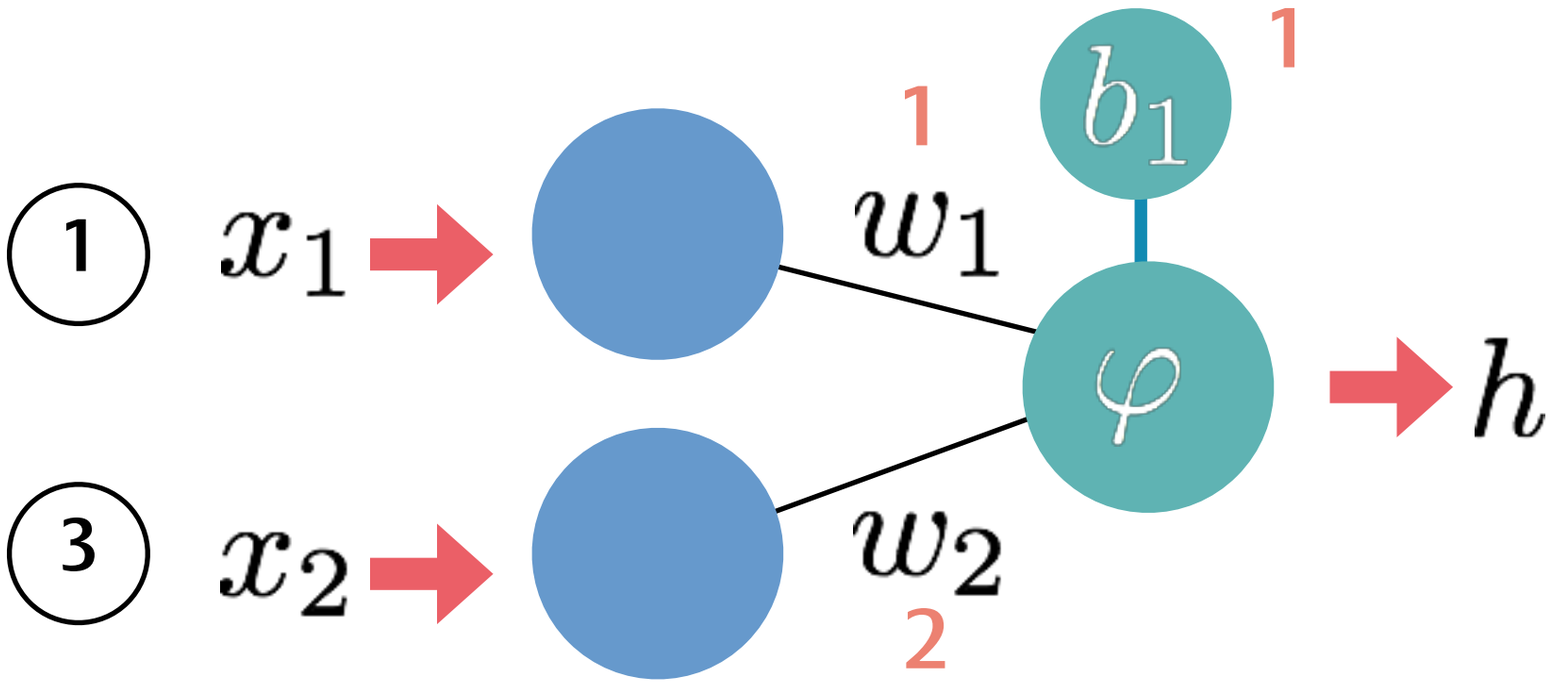
$$\varphi(w_1 x_1 + w_2 x_2 + b_1) = h$$

「學成的」神經網路



$$\varphi(\underset{1}{w_1}x_1 + \underset{2}{w_2}x_2 + \underset{1}{b_1}) = h$$

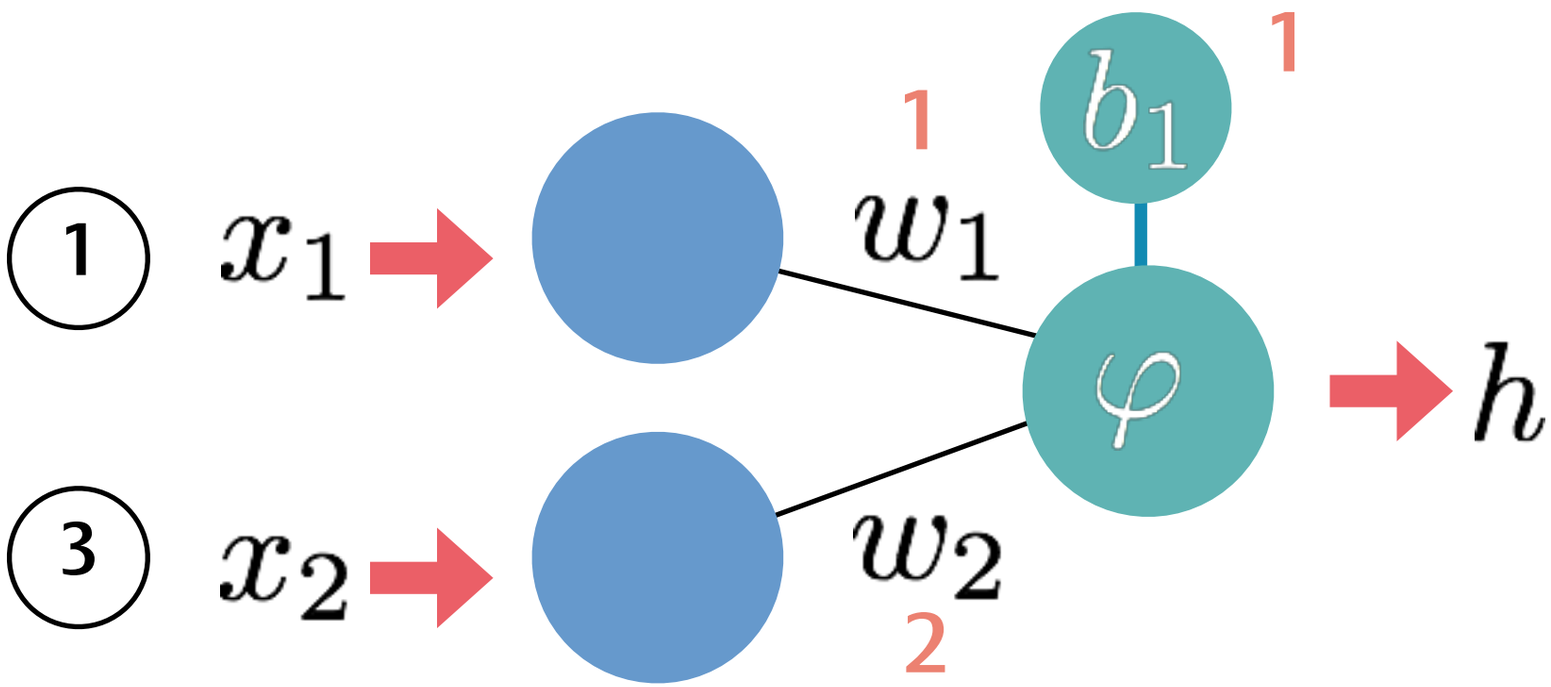
假設輸入 $(x_1, x_2) = (1, 3)$



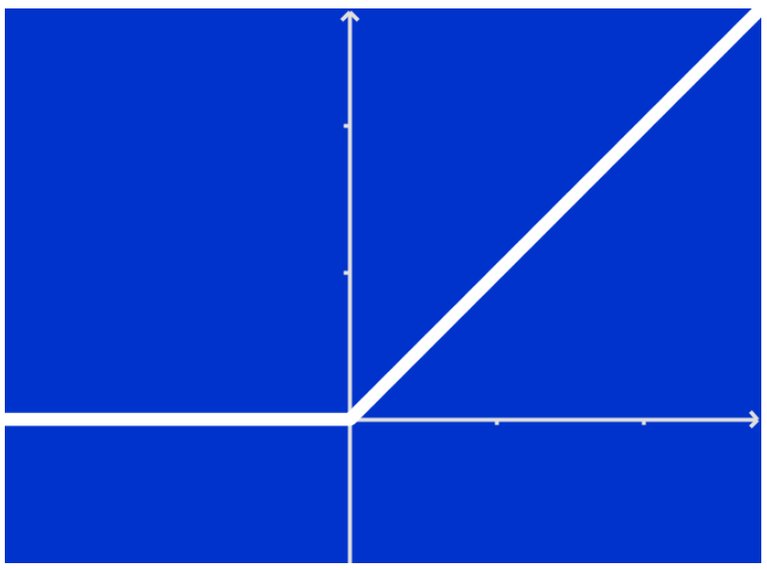
$$\varphi(\underset{1}{w_1}\underset{①}{x_1} + \underset{2}{w_2}\underset{③}{x_2} + \underset{1}{b_1}) = h$$

8

假設用 ReLU

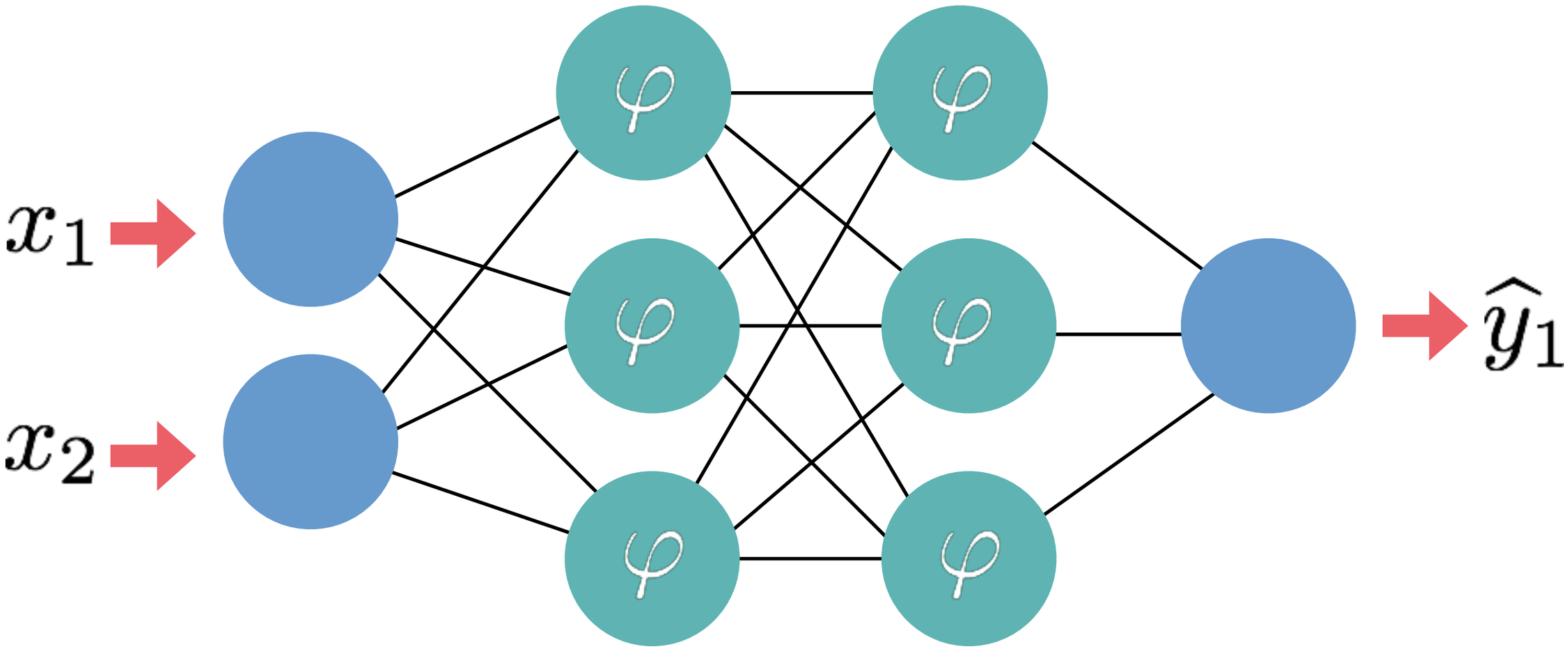


$$\varphi(8) = 8$$



練習

這樣子的神經網路變數有幾個呢?



固定結構神經網路的函數空間

當一個神經網路結構決定、activation functions 也決定，那可以調的就是 weights, biases。我們把這些參數的集合叫θ，每一個θ就定義一個函數，我們把它看成一個集合。

$$\{F_\theta\}$$

我們就是要找 θ^*

使得 F_{θ^*} 和目標函數最接近

「最近」是什麼意思

就是 “loss function” 的值最小

假設我們有訓練資料

$$\{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \ldots, (\mathbf{x}_k, \mathbf{y}_k)\}$$

最常用 loss function

我們希望它越小越好

$$L(\theta) = \frac{1}{2}\sum_{i=1}^{k} \|\mathbf{y}_i - F_\theta(\mathbf{x}_i)\|^2$$

對每個 $i = 1,2,\ldots,k$

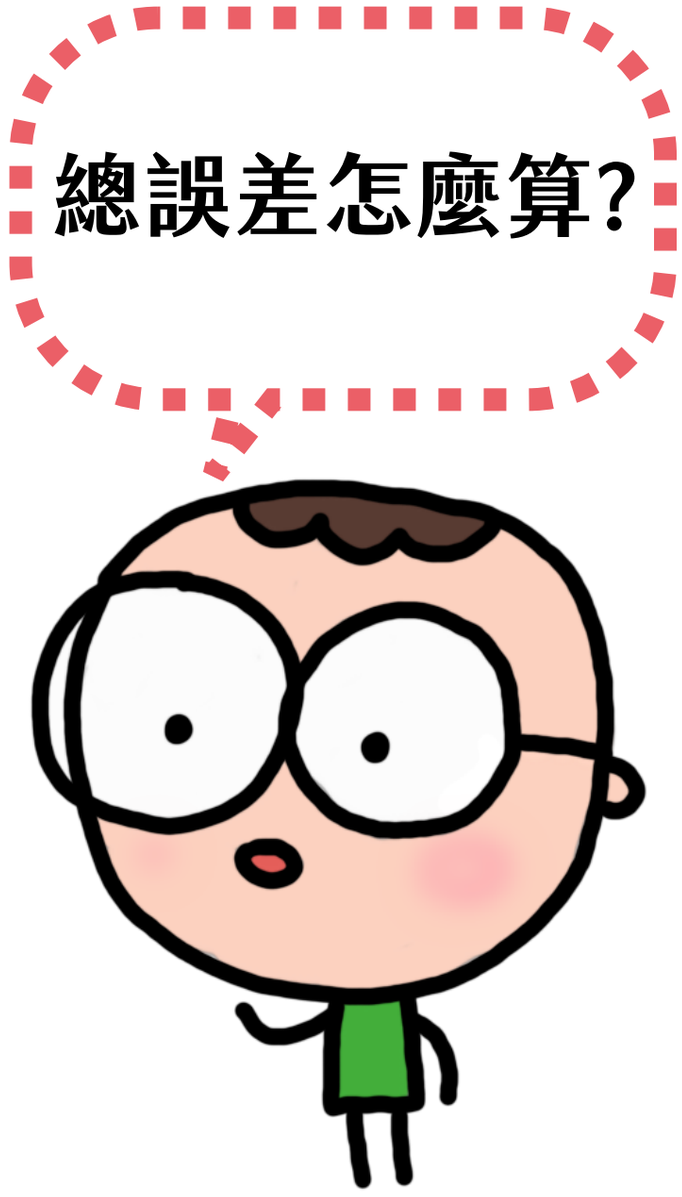
$\mathbf{x}_i$ → f → $\mathbf{y}_i$

正確答案

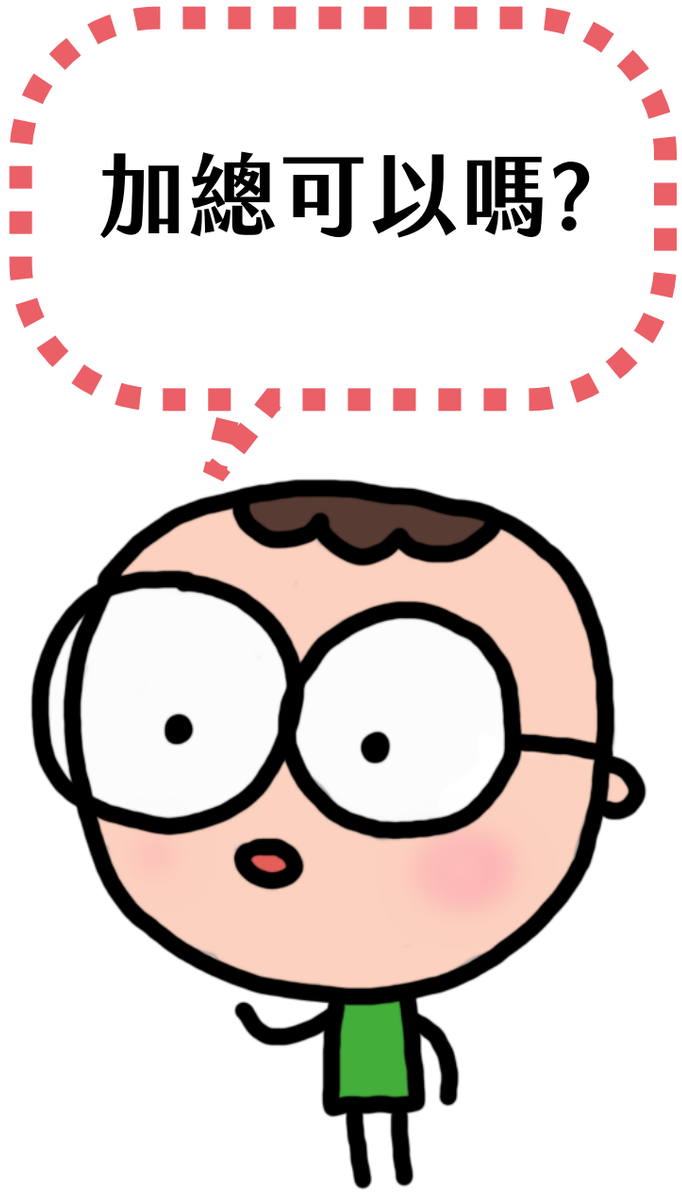
$\mathbf{x}_i$ → F_θ → $F_\theta(\mathbf{x}_i)$

學習機的答案

x_i	$F_\theta(x_i)$	y_i	誤差
1	2	3	1
2	5	2	-3
3	6	8	2
4	7	2	-5
5	6	6	0
6	4	8	4
7	8	9	1



x_i	$F_\theta(x_i)$	y_i	誤差
1	2	3	1
2	5	2	-3
3	6	8	2
4	7	2	-5
5	6	6	0
6	4	8	4
7	8	9	1



加總 = 0!!

x_i	$F_\theta(x_i)$	y_i	誤差
1	2	3	1
2	5	2	-3
3	6	8	2
4	7	2	-5
5	6	6	0
6	4	8	4
7	8	9	1

1
9
4
25
0
16
1

平方和

加總是 56 64

基本上這樣調

learning rate

$$-\eta \frac{\partial L}{\partial w_{ij}}$$

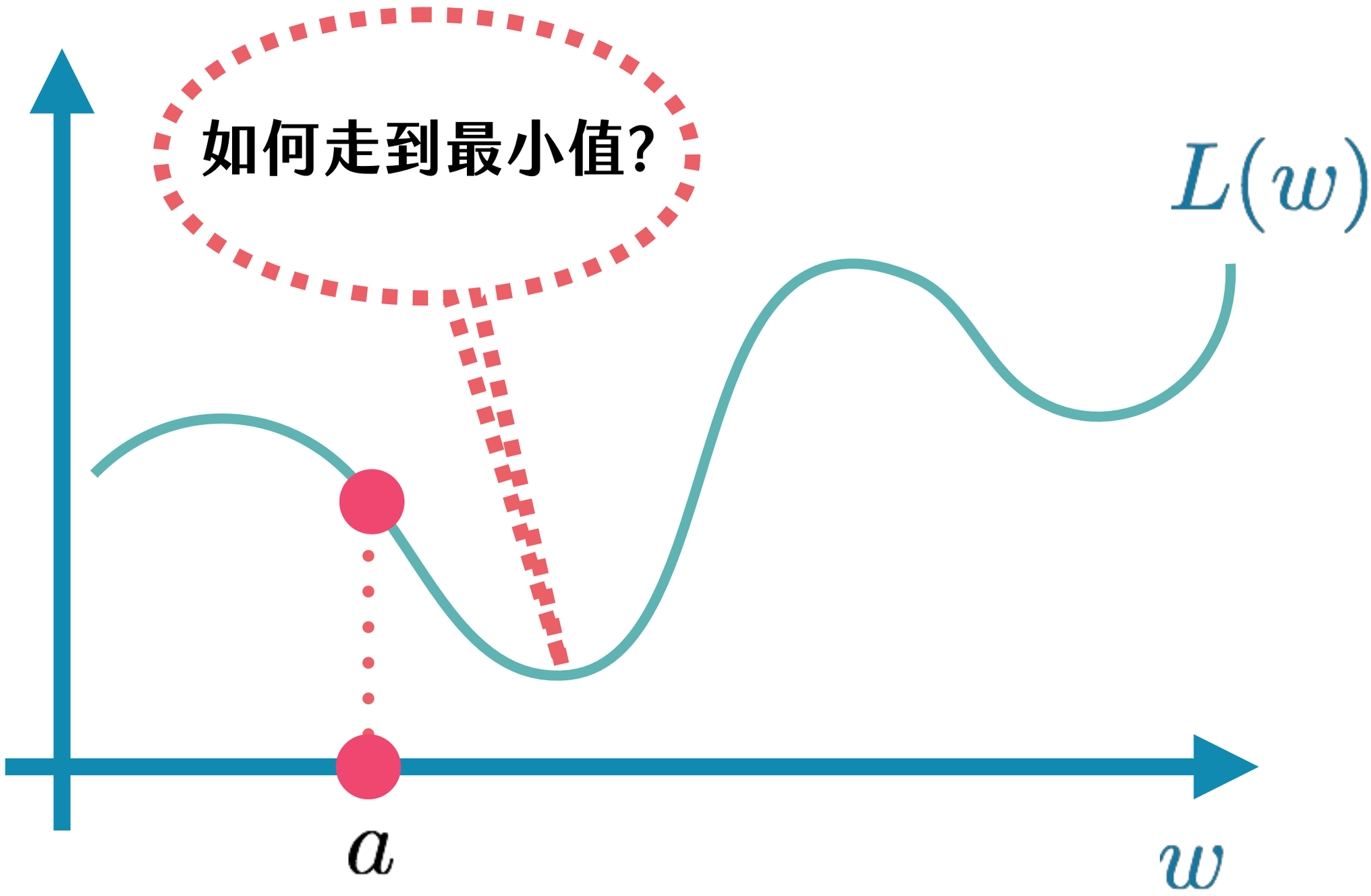
這可怕的東西是什麼意思?

記得 L 是 w_1, w_2, b_1, ⋯

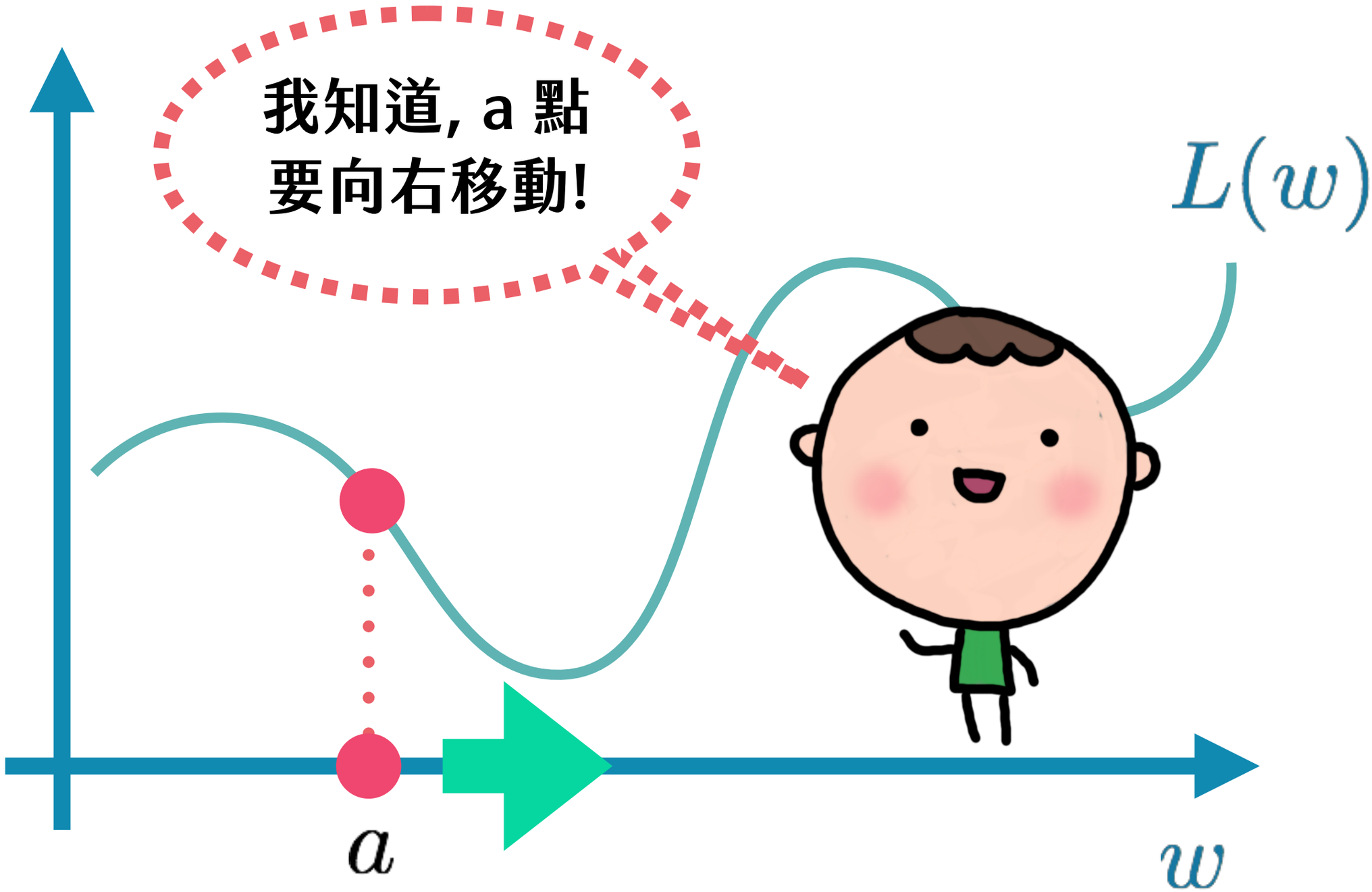
的函數

希望越小越好

為了簡化, 我們先把 L 想成只有一個變數 w

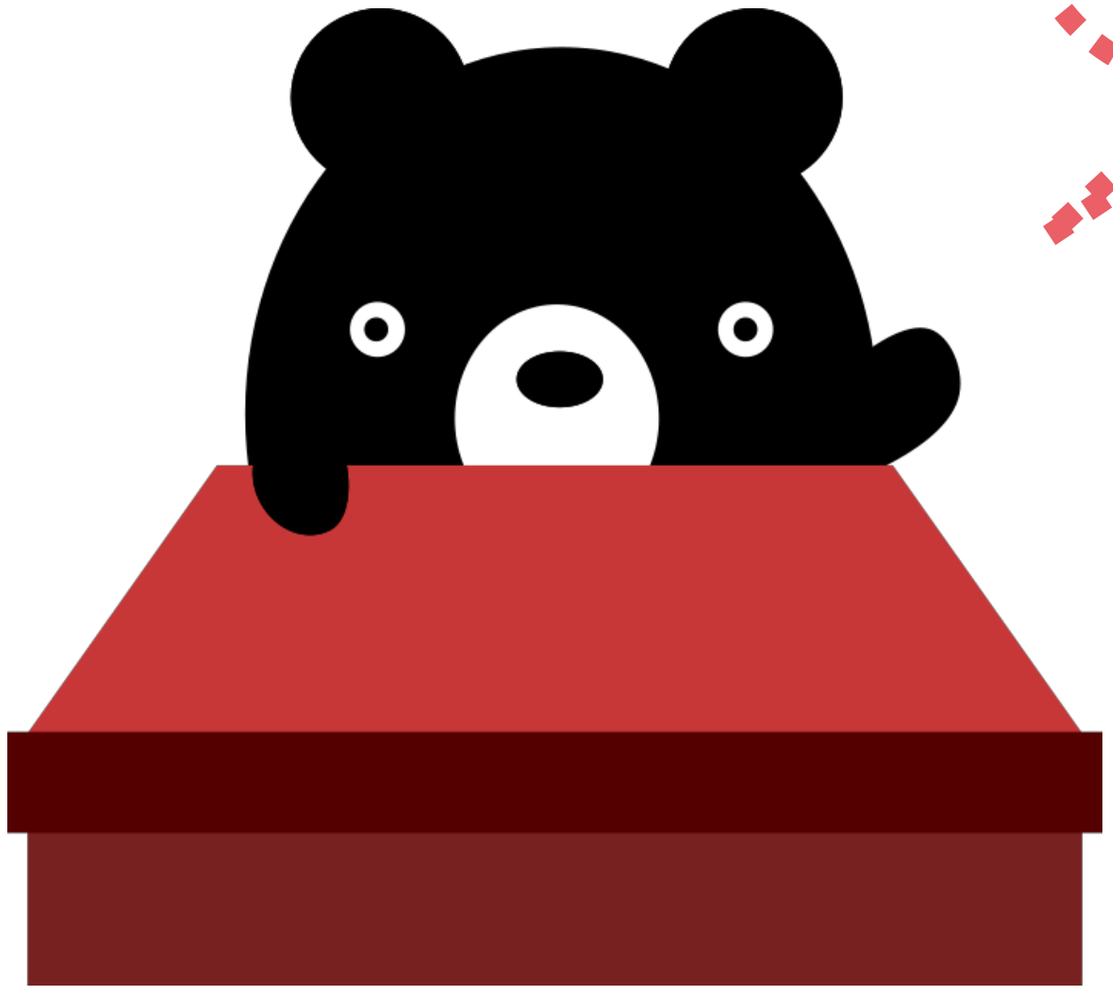


w 目前的值

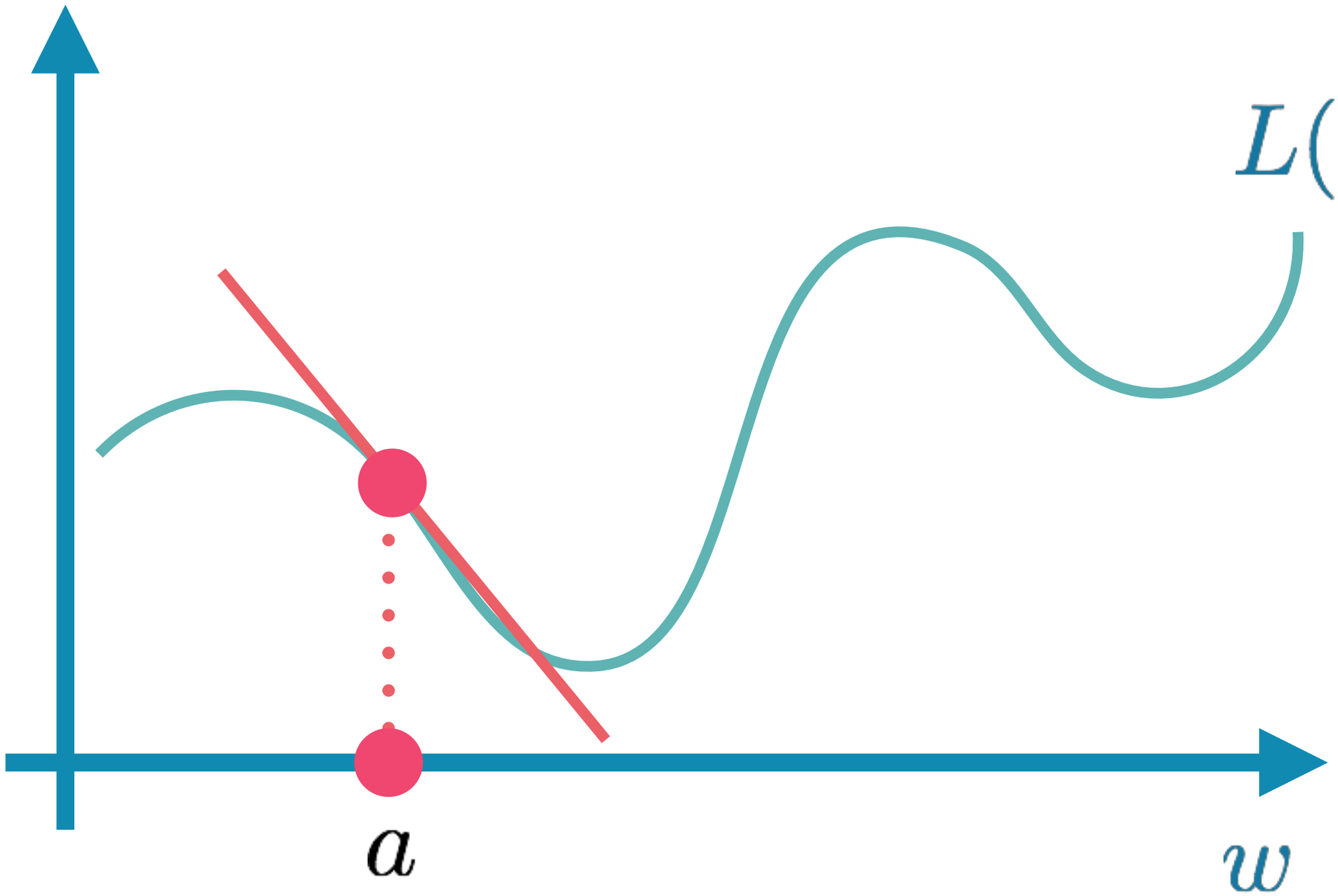


w 目前的值

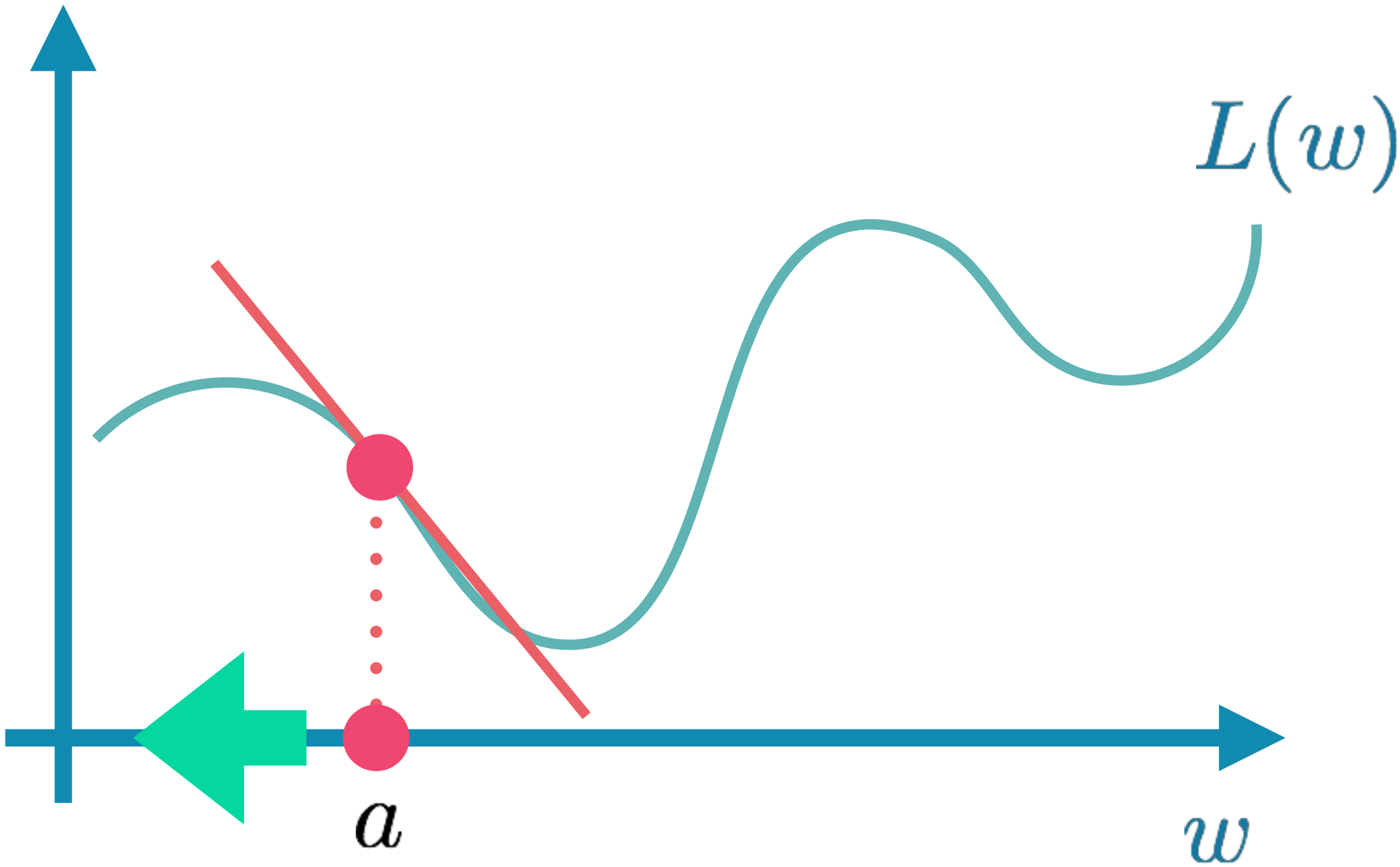
電腦怎麼「看」出來?



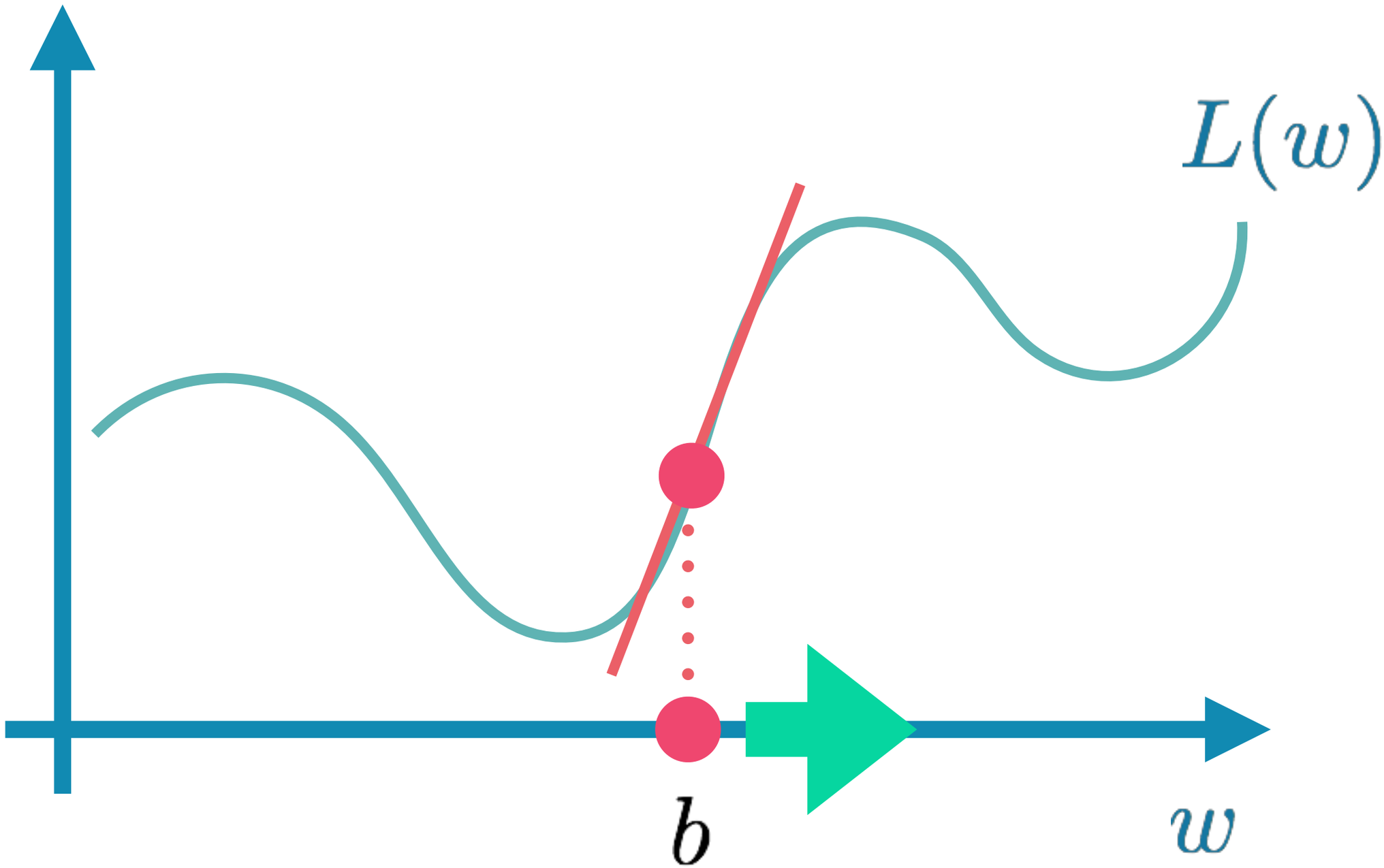
切線是關鍵!



切線斜率 < 0

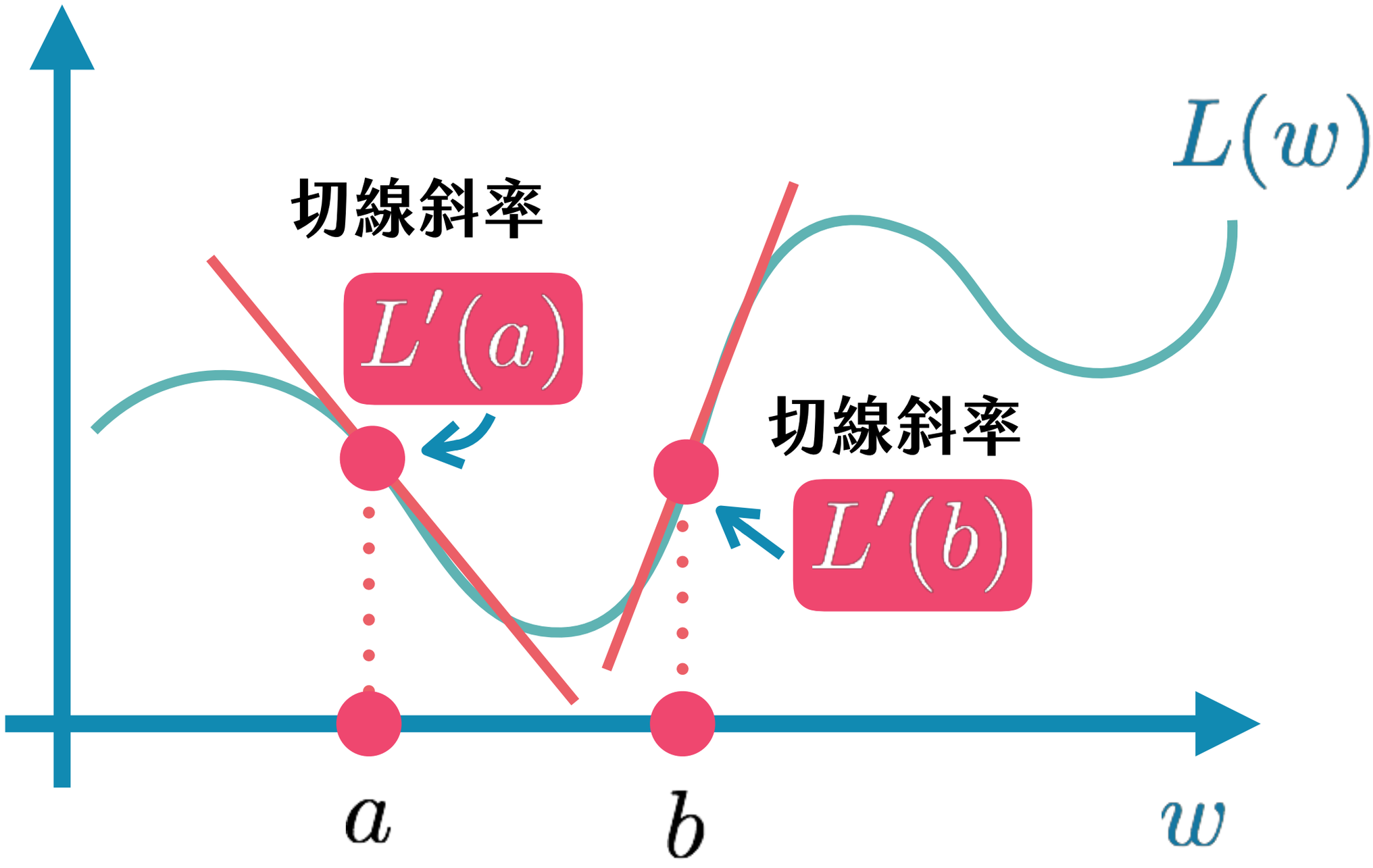


切線斜率 > 0

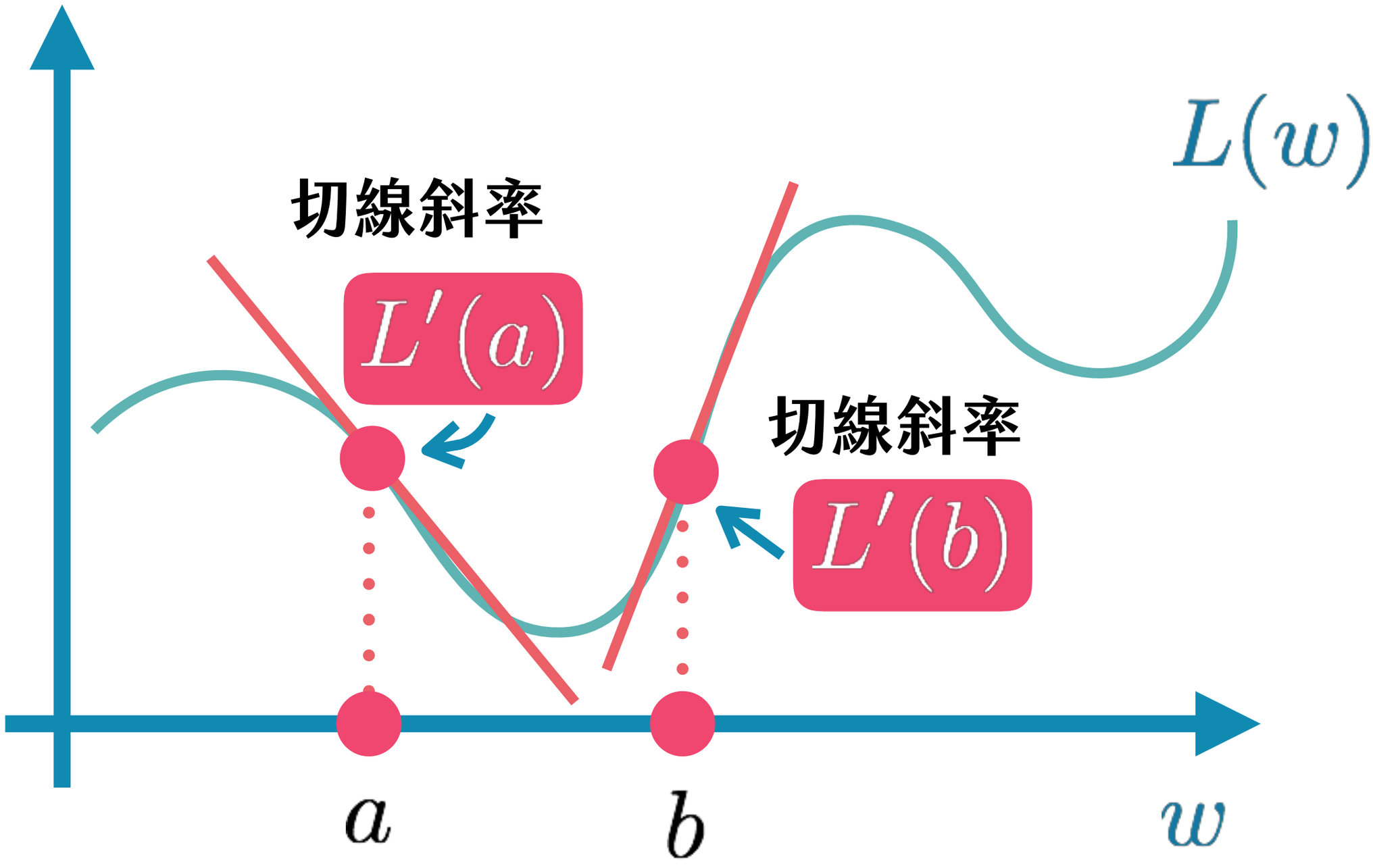


切線斜率指向 (局部) 最大值的方向!!

符號



切線斜率是變化率



符號

切線斜率合理的符號

因為是變化率, 合理的符號可以是:

$$L'(a) = \left.\frac{\Delta L}{\Delta w}\right|_{w \to a}$$

又醜又不潮!

符號

切線斜率合理的符號

萊布尼茲說我們要潮就這樣寫:

$$L'(a) = \left.\frac{dL}{dw}\right|_{w=a}$$

y 對 w 在 a 點的變化

符號

切線斜率合理的符號

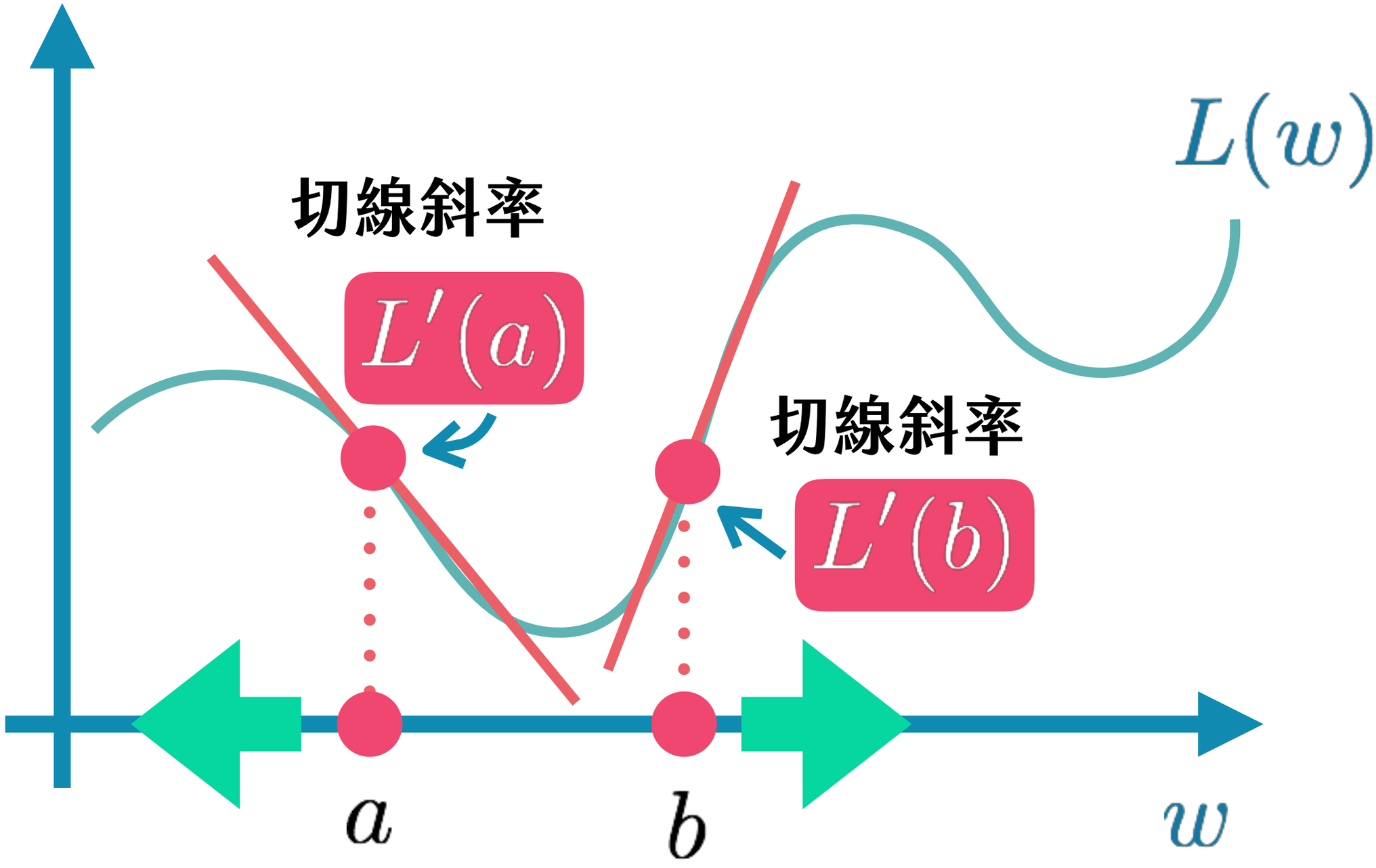
對任意的 w, 我們會寫成:

$$L'(w) = \frac{dL}{dw}$$

切線斜率函數

簡潔就是潮

正負指向 (局部) 極大



重點

往 (局部) 極小值移動

我們想調整 w 的值, 往極小值移動, 應該讓新的 w 變成:

$$w - \frac{dL}{dw}$$

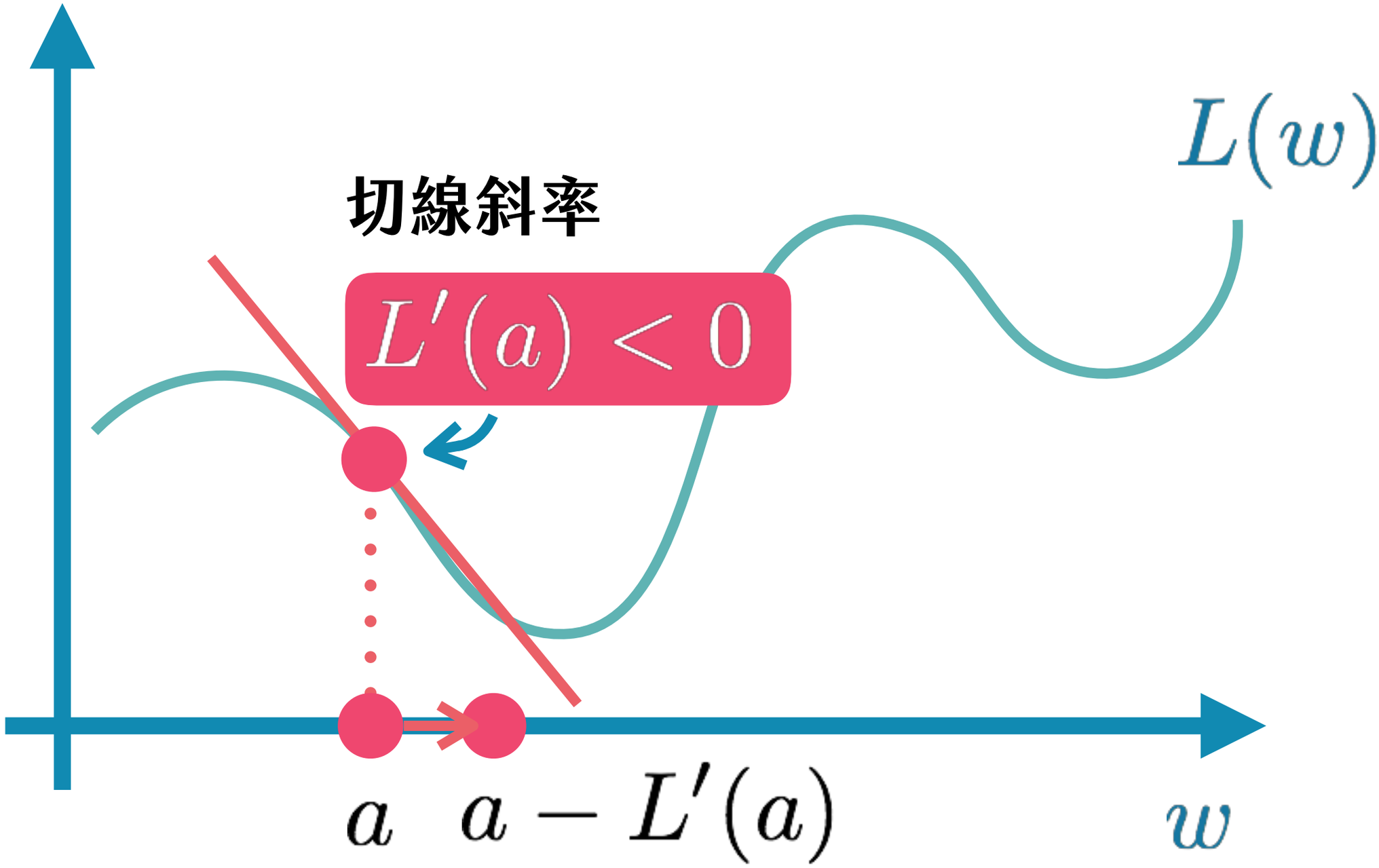
重點

往 (局部) 極小值移動

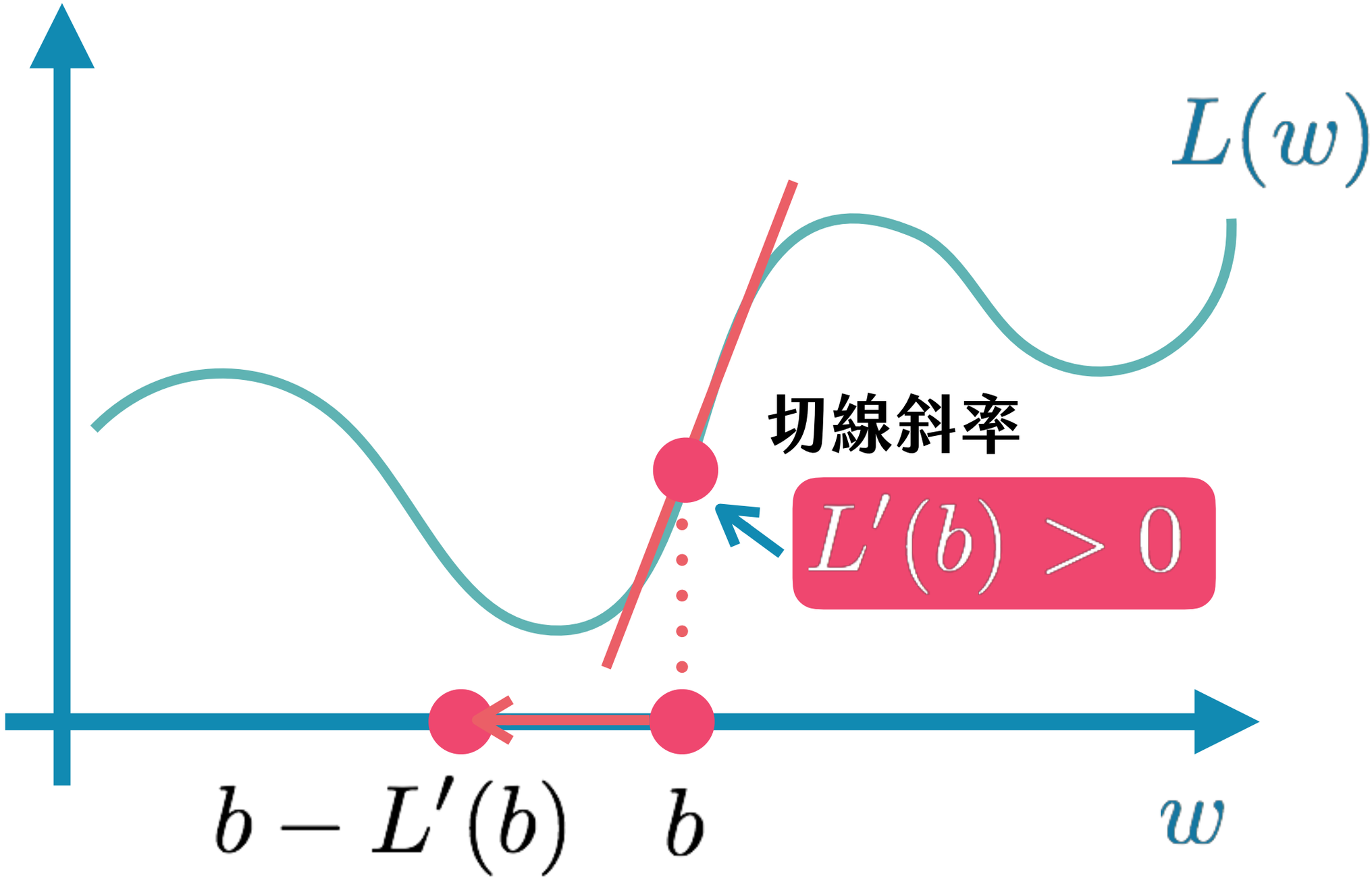
比如現在在 w=a, 要調整為

$$a - L'(a)$$

調整示意圖



有時會跑過頭!



重點

往 (局部) 極小值移動

為了不要一次調太大, 我們會乘上一個小小的數, 叫 Leraning Rate:

$$w - \eta \frac{dL}{dw}$$

求切線斜率的動作, 就是

微分

自牛頓和萊布尼茲以來我們就很會算。

王牌公式

$$L(w) = w^n$$

$$L'(w) = \frac{dL}{dw} = \frac{dw^n}{dw}$$

王牌公式

$$L(w) = w^n$$

留下 n-1

$$\frac{dw^n}{dw} = nw^{n-1}$$

n 被推下來

例子

設 $L(w) = w^2$, 求 L 在 w=3 之切線斜率。

$$L'(w) = \frac{dw^2}{dw} = 2w$$

$$L'(3) = 6$$

例子

我們來調整 w, 讓 L 變小! 這裡 leraning rate 取 0.2

$$w - \eta \frac{dL}{dw}$$

例子

我們來調整 w, 讓 L 變小! 這裡 leraning rate 取 0.2

w	L(w)
3	9
1.8	3.24
1.08	1.1664
0.65	0.4225
0.39	0.1521
0.23	0.0529
0.14	0.0196
0.08	0.0064

重要性質

微分是線性的

若 $L(w) = L_1(w) + L_2(w)$

$$\Rightarrow L'(w) = L_1'(w) + L_2'(w)$$

重要性質

微分是線性的

對於任意實數 a

$$L'(aw) = aL'(w)$$

所有多項式函數我們都會微分!!

練習

試著微微看這個函數:

$$L(w) = 3w^2 - w + 3$$

可是, 變數不只一個...

例子

$$L(w_1, w_2, b_1) = (b_1 + 2w_1 - w_2 - 3)^2$$

假設這時我們在:

$$w_1 = 1,\ w_2 = -1,\ b_1 = 2$$

一樣準備往 (局部) 極小值移動。

假裝只有一個變數!

比如說 w_1

例子

$$L(w_1, w_2, b_1) = (b_1 + 2w_1 - w_2 - 3)^2$$

$$w_1 = 1,\ w_2 = -1,\ b_1 = 2$$

如果除了 w_1, 其他變數不動...

$$L_{w_1}(w_1) = L(w_1, -1, 2) = 4w_1^2$$

一個變數的函數!

例子

$$L(w_1, w_2, b_1) = (b_1 + 2w_1 - w_2 - 3)^2$$

$$w_1 = 1,\ w_2 = -1,\ b_1 = 2$$

同理,

$$L_{w_2}(w_2) = L(1, w_2, 2) = (-w_2 + 1)^2$$

$$L_{b_1}(b_1) = L(1, -1, b_1) = b_1^2$$

例子

於是我們又會調整 w_1, w_2, b_1, 讓 L 慢慢走向 (局部) 極小。

$w_1 \leftarrow w_1 - \eta \frac{dL_{w_1}}{dw_1}$

$b_1 \leftarrow b_1 - \eta \frac{dL_{b_1}}{db_1}$

$w_2 \leftarrow w_2 - \eta \frac{dL_{w_2}}{dw_2}$

例子

寫在一起看來更有學問!

$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} - \eta \begin{bmatrix} \frac{dL_{w_1}}{dw_1} \\ \frac{dL_{w_2}}{dw_2} \\ \frac{dL_{b_1}}{db_1} \end{bmatrix}$$

新的

這叫 L 的 gradient

但這符號有點麻煩

還要新創函數$L_{w_1}, L_{w_2}, L_{b_1}$

定義

偏微分

我們有 $L(w_1, w_2, b_1)$ 這三個變數的函數, 當我們只把 w_1 當變數, 其他 w_2, b_1 當常數的微分。

$$\frac{\partial L}{\partial w_1} = \frac{dL_{w_1}}{dw_1}$$

定義 **偏微分**

同理,

$$\frac{\partial L}{\partial w_2} = \frac{dL_{w_2}}{dw_2}$$

$$\frac{\partial L}{\partial b_1} = \frac{dL_{b_1}}{db_1}$$

符號

梯度 (gradient)

函數 L 的 gradient 就變成:

$$\begin{bmatrix} \frac{\partial L}{\partial w_1} \\ \frac{\partial L}{\partial w_2} \\ \frac{\partial L}{\partial b_1} \end{bmatrix} \quad \begin{bmatrix} \frac{dL_{w_1}}{dw_1} \\ \frac{dL_{w_2}}{dw_2} \\ \frac{dL_{b_1}}{b_1} \end{bmatrix}$$

符號

梯度 (gradient)

Gradient 當然要有很酷的符號:

$$\nabla L = \begin{bmatrix} \frac{\partial L}{\partial w_1} \\ \\ \frac{\partial L}{\partial w_2} \\ \\ \frac{\partial L}{\partial b_1} \end{bmatrix}$$

符號

梯度 (gradient)

我們調整 w_1, w_2, b_1 的方法就變成:

$$\begin{bmatrix} w_1 \\ w_2 \\ b_1 \end{bmatrix} - \eta \nabla L$$

這「學習法」有個很潮的名字

Gradient Descent

梯度下降

回到我們的例子

例子

$$L(w_1, w_2, b_1) = (b_1 + 2w_1 - w_2 - 3)^2$$

在 $(w_1, w_2, b_1) = (1, -1, 2)$

例子

$$L_{w_1}(w_1) = L(w_1, -1, 2) = 4w_1^2$$
$$L_{w_2}(w_2) = L(1, w_2, 2) = w_2^2 - 2w_2 + 1$$
$$L_{b_1}(b_1) = L(1, -1, b_1) = b_1^2$$

$$\nabla L = \begin{bmatrix} \frac{\partial L}{\partial w_1} \\ \frac{\partial L}{\partial w_2} \\ \frac{\partial L}{\partial b_1} \end{bmatrix} = \begin{bmatrix} 8w_1 \\ 2w_2 - 2 \\ 2b_1 \end{bmatrix} = \begin{bmatrix} 8 \\ -4 \\ 4 \end{bmatrix}$$

例子

這次我們的 learning rate 取$\eta = 0.01$

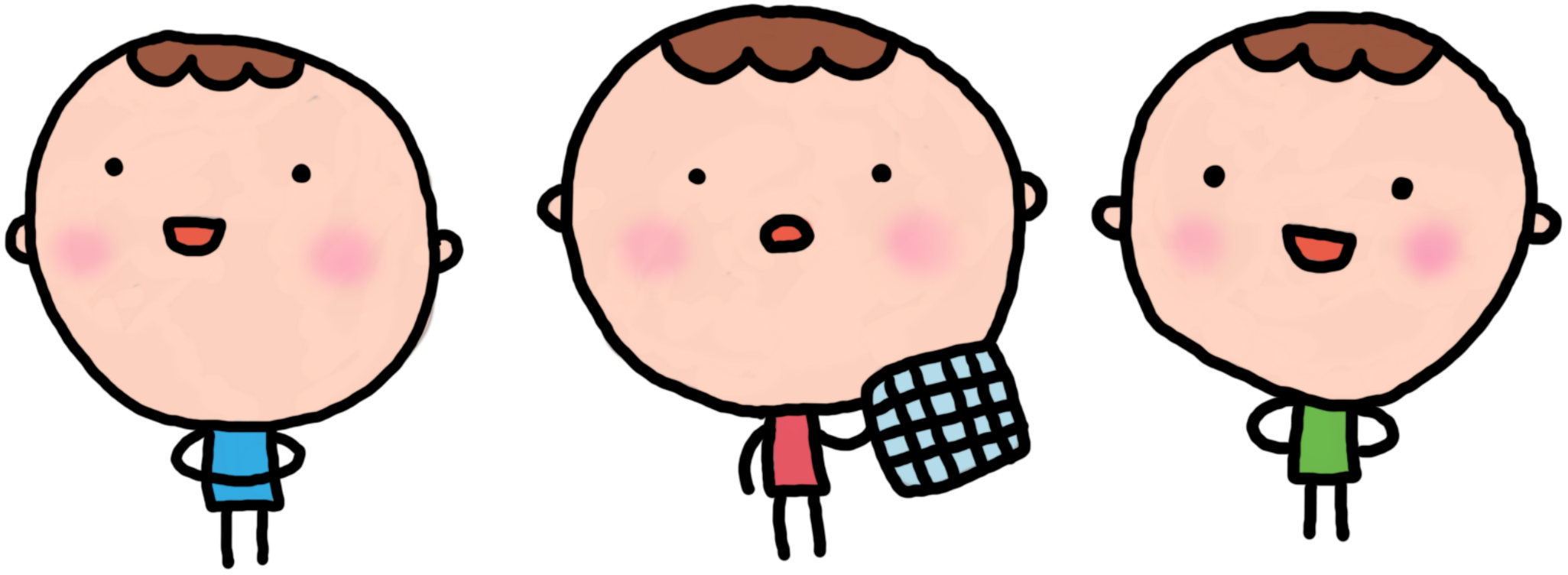
$$\begin{bmatrix} w_1 \\ w_2 \\ b_1 \end{bmatrix} - \eta \nabla L = \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix} - 0.01 \begin{bmatrix} 8 \\ -4 \\ 4 \end{bmatrix} = \begin{bmatrix} 0.92 \\ -0.96 \\ 1.96 \end{bmatrix}$$

例子

就這樣做下去。

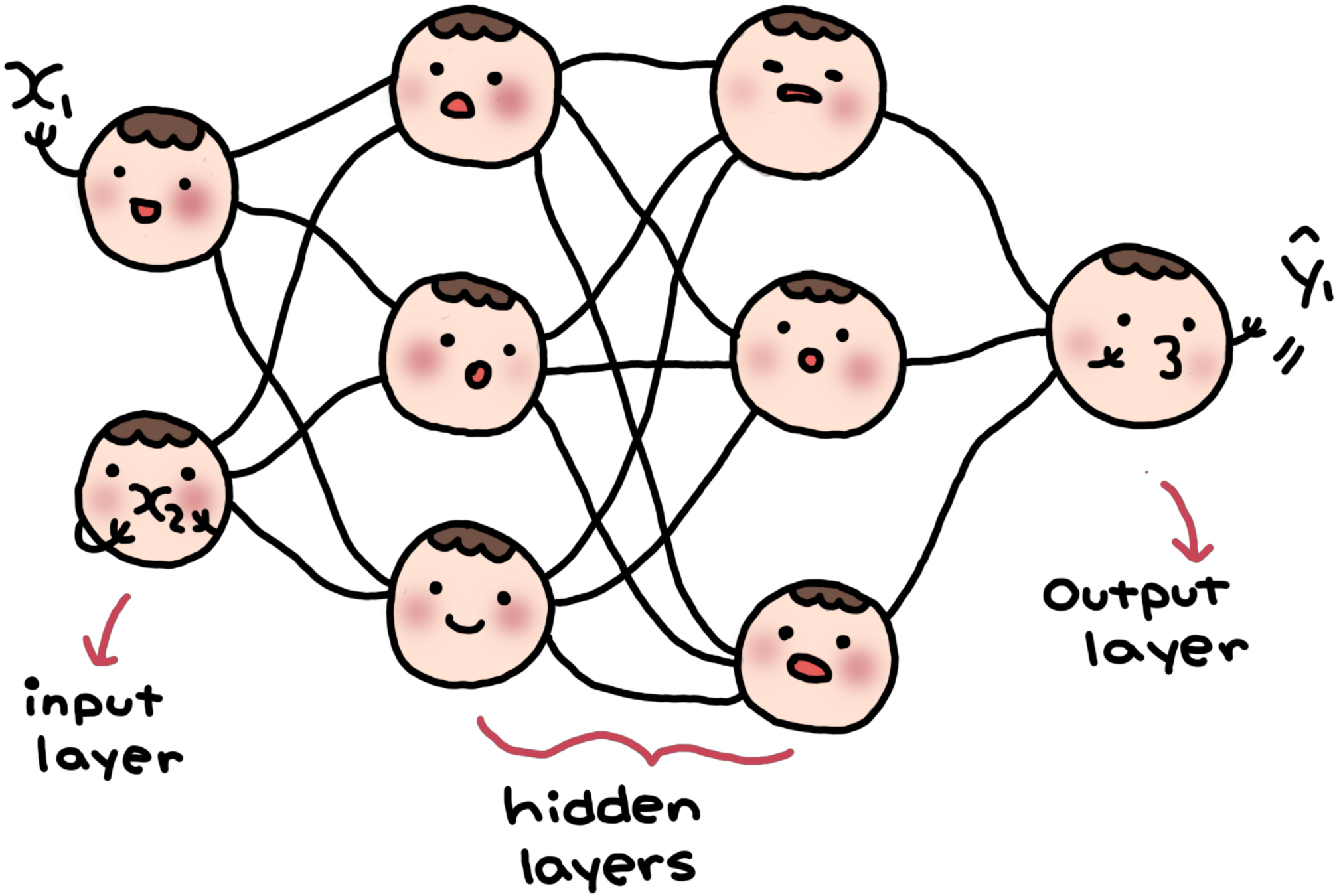
(w_1,w_2,b_1)	L
(1, -1, 2)	4
(0.92, -0.96, 1.96)	3.0976
(0.85, -0.92, 1.92)	2.3988
(0.79, -0.89, 1.89)	1.8576
(0.73, -0.87, 1.87)	1.4385
(0.69, -0.84, 1.84)	1.1140

Deep Learning
三大天王

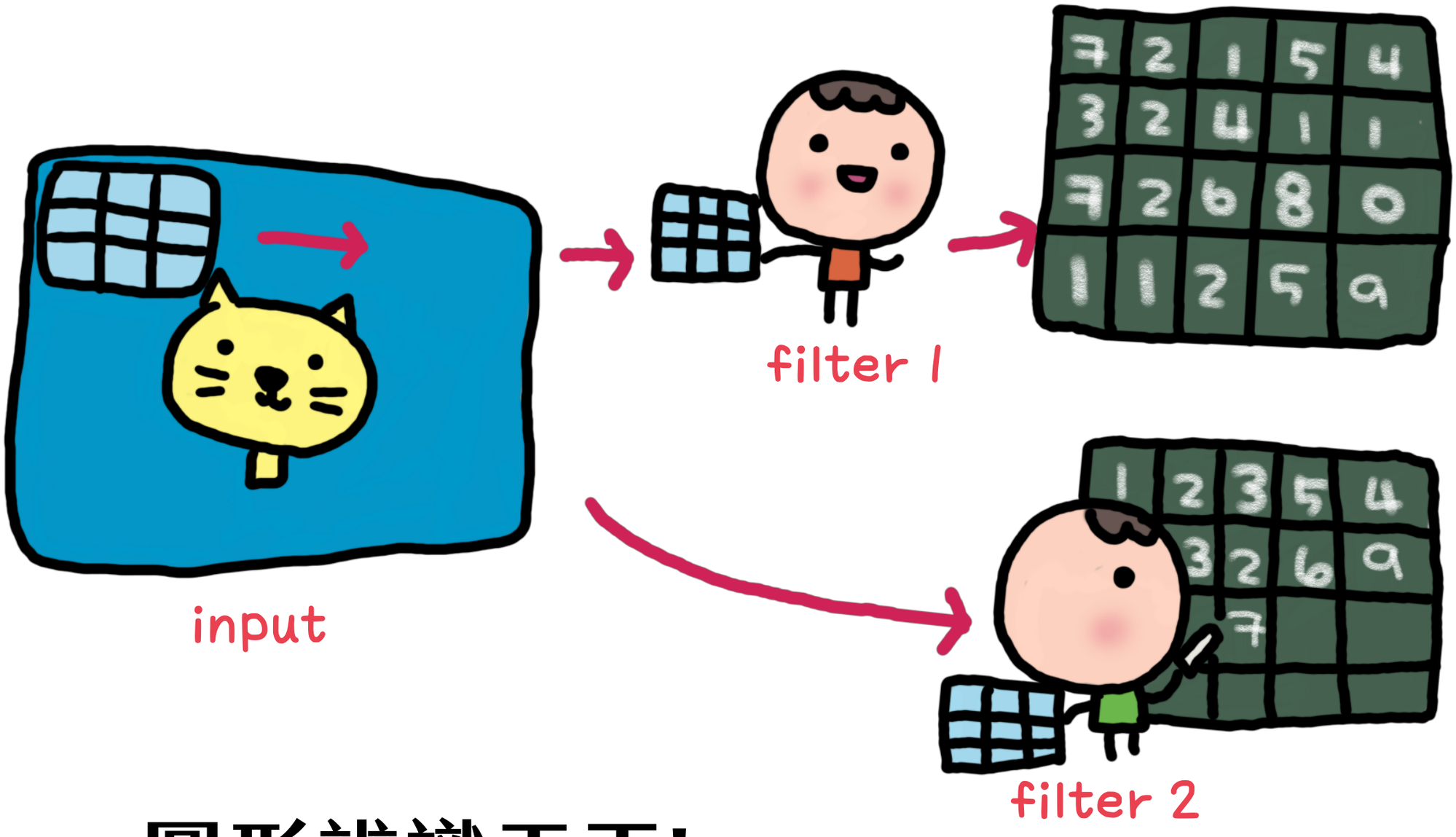


標準 NN　　CNN　　RNN

標準的神經網路



Convolutional Neural Network (CNN)



圖形辨識天王!

1

Convolutional Layer

我們要做 filters

例如 3x3 的大小

內積

$$W = \begin{bmatrix} 1 & 0 & 1 \\ 2 & 1 & 3 \\ 1 & 1 & 2 \end{bmatrix}$$

filter

這學來的

2	5	5	2	5	2	0	1
2	3	4	0	4	2	1	5
4	3	1	3	5	5	4	3
5	3	4	5	0	2	1	5
2	3	1	1	1	0	1	3
4	4	1	1	5	1	1	4
2	3	2	2	0	4	2	4
0	5	4	5	3	4	1	4

35					

想成這是一張圖所成的矩陣

$$W = \begin{bmatrix} 1 & 0 & 1 \\ 2 & 1 & 3 \\ 1 & 1 & 2 \end{bmatrix}$$

filter

還是一樣的矩陣

2	5	5	2	5	2	0	1
2	3	4	0	4	2	1	5
4	3	1	3	5	5	4	3
5	2	1	5	0	2	1	5
2	3	1	1	1	0	1	3
4	4	1	1	5	1	1	4
2	3	2	2	0	4	2	4
0	5	4	5	3	4	1	4

右移一格

35	**27**				

filter

$$W = \begin{bmatrix} 1 & 0 & 1 \\ 2 & 1 & 3 \\ 1 & 1 & 2 \end{bmatrix}$$

2	5	5	2	5	2	0	1
2	3	4	0	4	2	1	5
4	3	1	3	5	5	4	3
5	3	4	5	0	2	1	5
2	3	1	1	1	0	1	3
4	4	1	1	5	1	1	4
2	3	2	2	0	4	2	4
0	5	4	5	3	4	1	4

35	27	44	32	36	38
36	36	37	36	36	43
37	37	23	26	17	35
29	25	22	18	14	27
27	25	24	21	24	32
31	38	27	34	25	**40**

一路到最後

最後就是一個 6x6 的矩陣
有時我們會把它弄成還是 8x8
基本上和原矩陣一樣大
而且我們通常 filter 會很多!

35	27	44	32	36	38
36	36	37	36	36	43
37	37	23	26	17	35
29	25	22	18	14	27
27	25	24	21	24	32
31	38	27	34	25	40

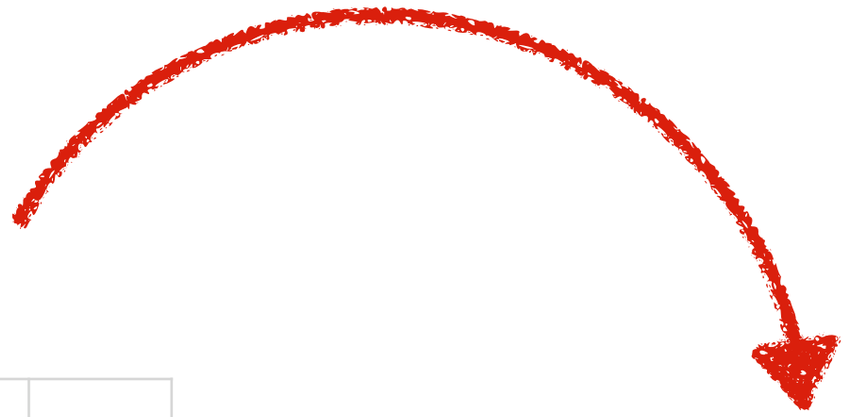
2

Max-Pooling Layer

基本上就是「投票」

看我們要多大區選一個代表, 例如 2x2

35	27	**44**	32	36	38
36	**36**	37	36	36	**43**
37	37	23	**26**	17	**35**
29	25	22	18	14	27
27	25	24	21	24	32
31	**38**	27	**34**	25	**40**



36	**44**	**43**
37	**26**	**35**
38	**34**	**40**

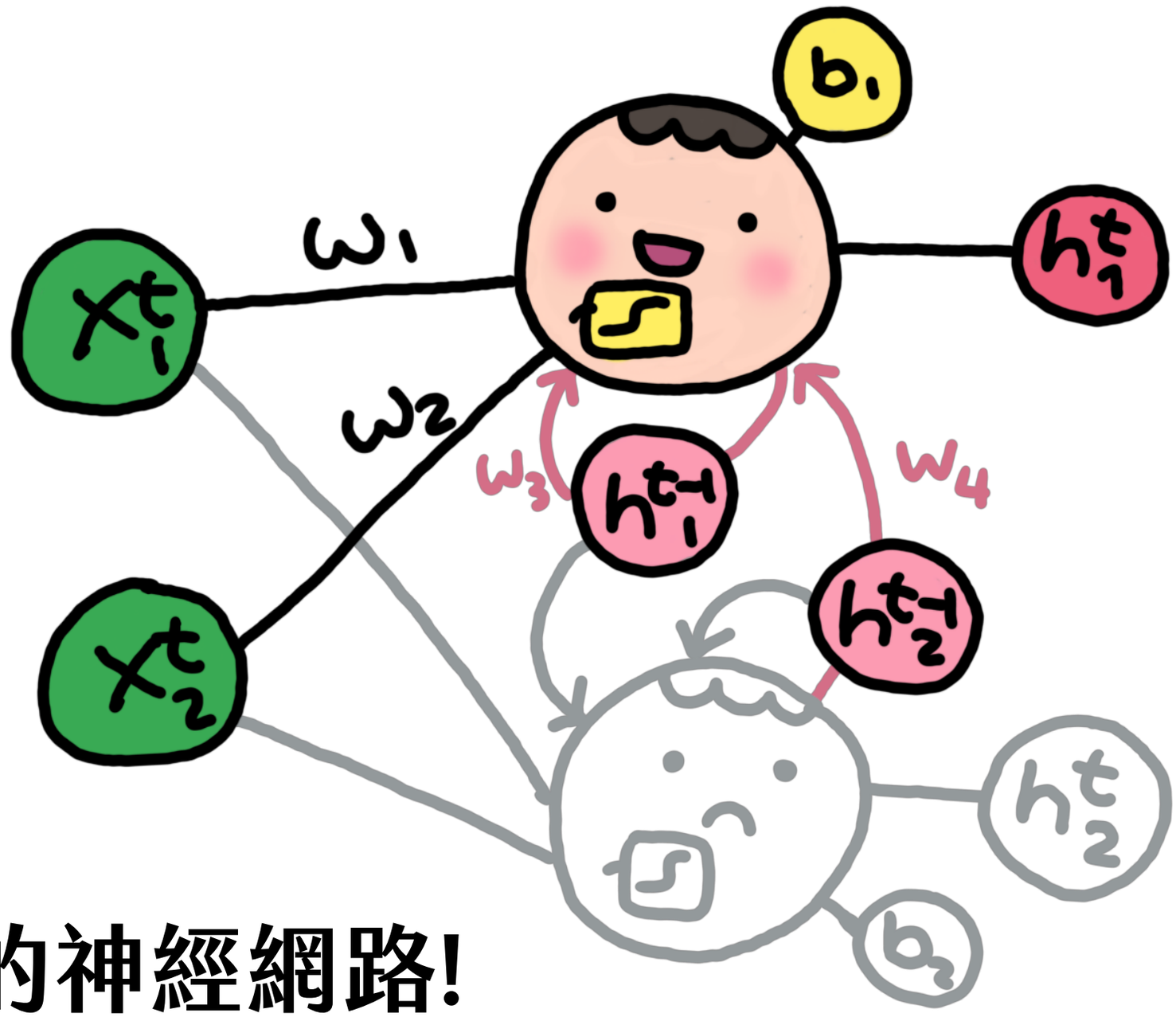
每區投出最大的!!

可以不斷重覆

convolution, max-pooling, convolution, max-pooling…

做完再送到「正常的」神經網路

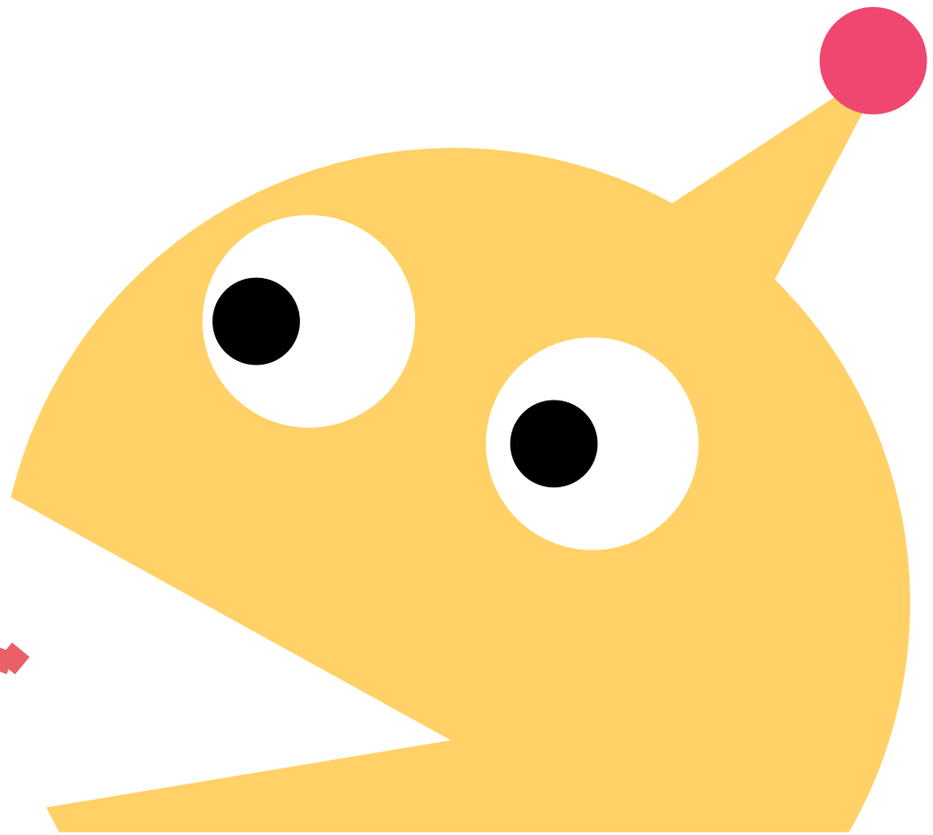
Recurrent Neural Network (RNN)



有記憶的神經網路!

$$h^t_1 = \sigma(w_1 x^t_1 + w_2 x^t_2 + w_3 h^{t-1}_3 + w_4 h^{t-1}_4 + b_1)$$

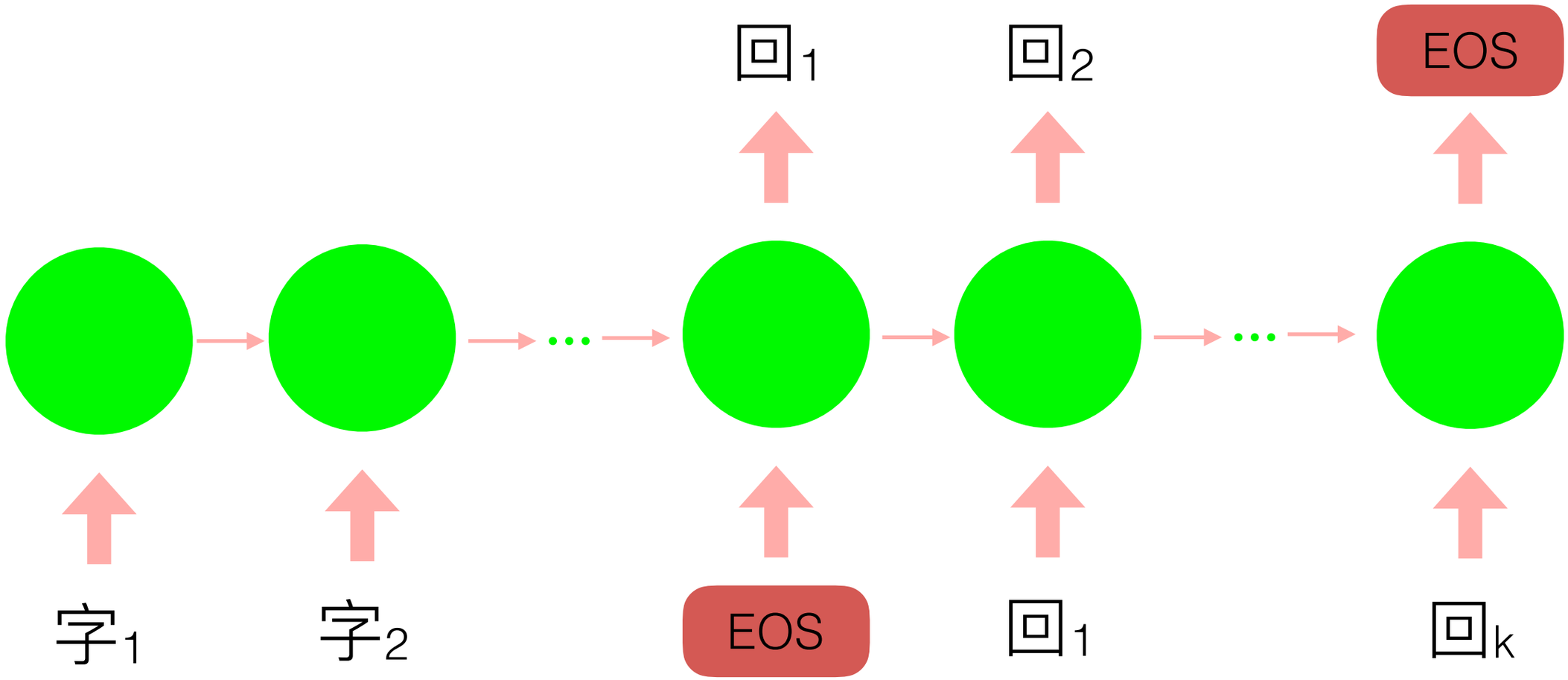
對話機器人



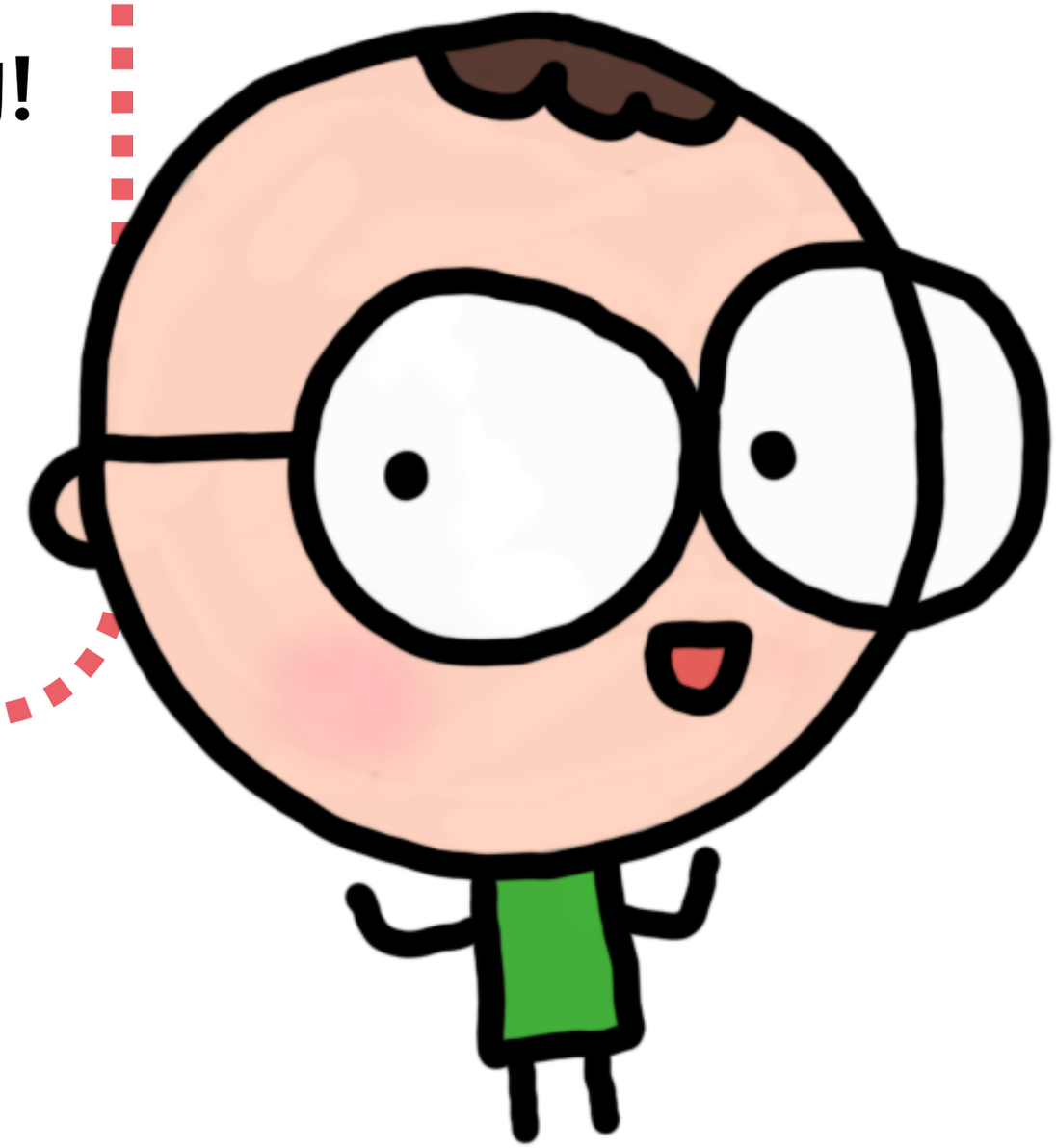
f(目前的字) = 下一個字

応用

対話機器人



其實輸入不一定要文字, 是影片 (一張一張的圖) 也是可以的!
輸出還是可以為文字, 最常見的大概是讓電腦說影片中發生什麼事。



同樣型式的應用

- 翻譯。
- Video Captioning 生成影片敘述。
- 生成一段文字。
- 畫一半的圖完成它。

To prove study we see that $\mathcal{F}|_U$ is a covering of $\mathcal{X}'$, and $\mathcal{T}_i$ is an object of $\mathcal{F}_{X/S}$ for $i > 0$ and $\mathcal{F}_p$ exists and let $\mathcal{F}_i$ be a presheaf of $\mathcal{O}_X$-modules on $\mathcal{C}$ as a $\mathcal{F}$-module. In particular $\mathcal{F} = U/\mathcal{F}$ we have to show that

$$\widetilde{M}^\bullet = \mathcal{I}^\bullet \otimes_{\mathrm{Spec}(k)} \mathcal{O}_{S,s} - i_X^{-1}\mathcal{F})$$

is a unique morphism of algebraic stacks. Note that

$$\mathrm{Arrows} = (Sch/S)^{opp}_{fppf}, (Sch/S)_{fppf}$$

and

$$V = \Gamma(S, \mathcal{O}) \longmapsto (U, \mathrm{Spec}(A))$$

is an open subset of X. Thus U is affine. This is a continuous map of X is the inverse, the groupoid scheme S.

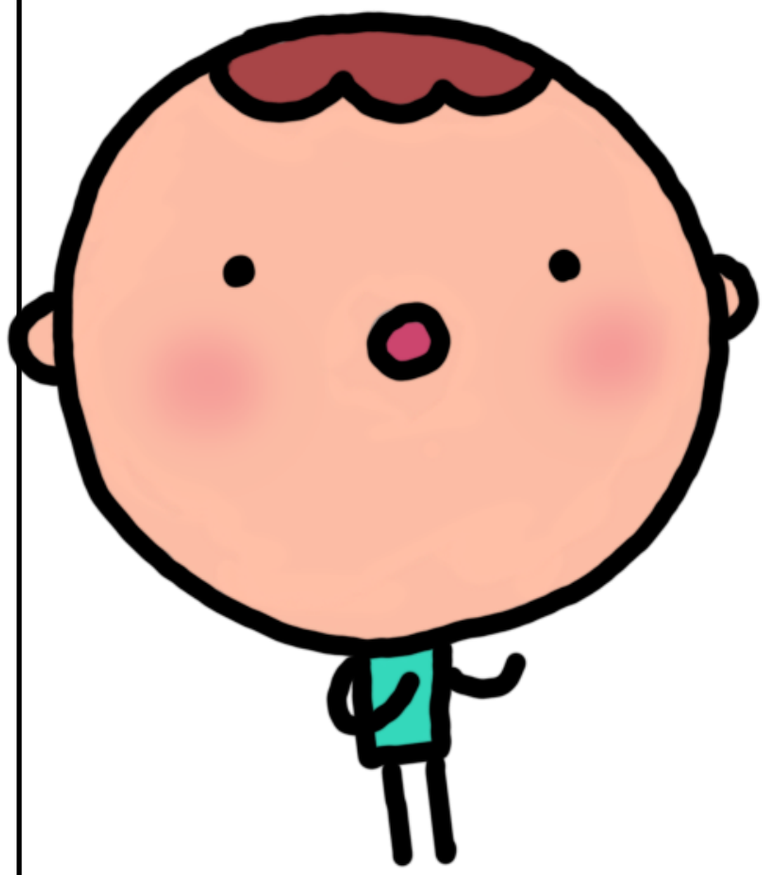
Proof. See discussion of sheaves of sets. □

Andrej Karpathy 生出代數幾何介紹 "Stacks" 的文字

http://karpathy.github.io/2015/05/21/rnn-effectiveness/

潘達洛斯：
唉，我想他應該過來接近一天
當小的小麥變成從不吃的時候，
誰是他的死亡鏈條和臣民，
我不應該睡覺。

第二位參議員：
他們遠離了我心中產生的這些苦難，
當我滅亡的時候，我應該埋葬和堅強
許多國家的地球和思想。

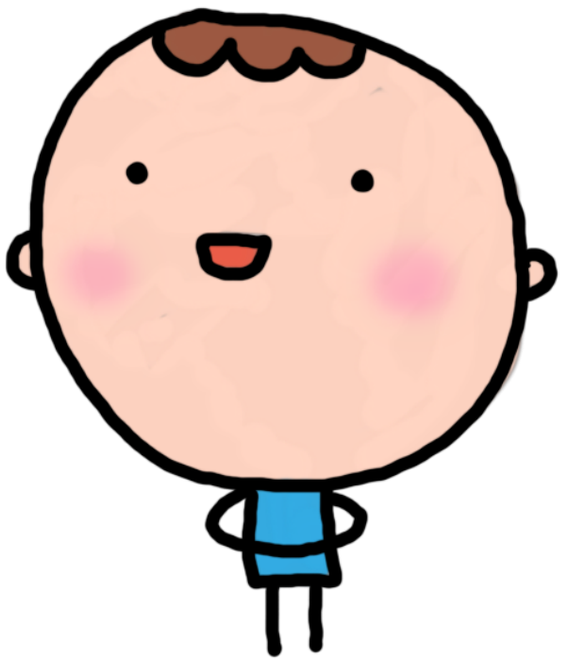


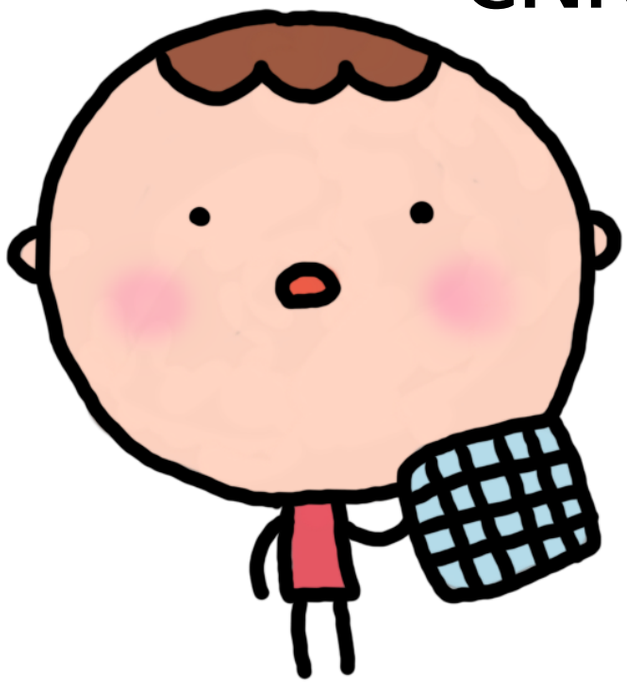
電腦仿的莎士比亞作品。

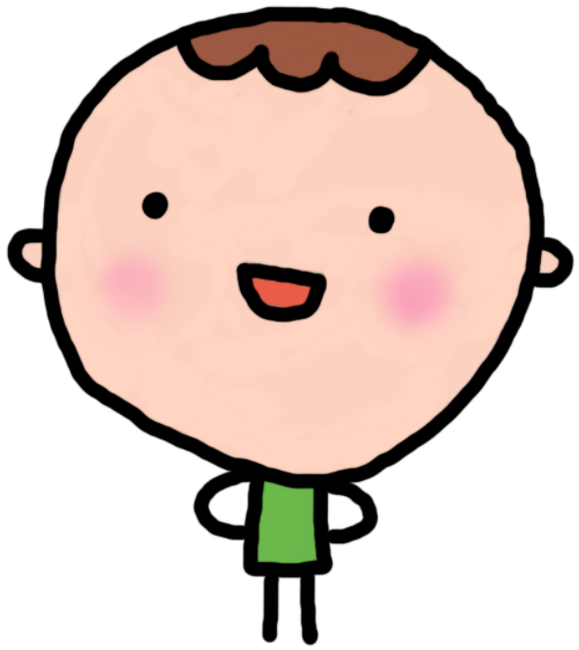
標準 NN

CNN

RNN

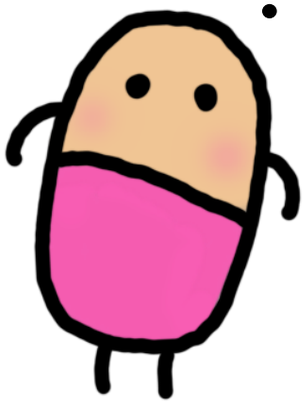




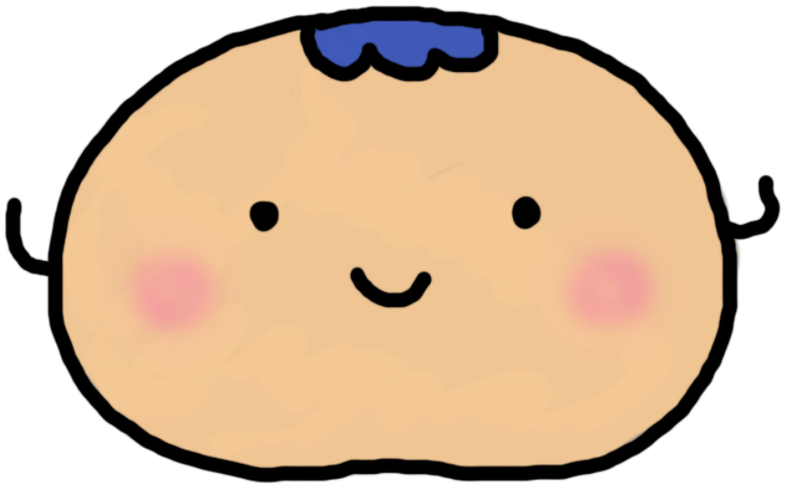


?

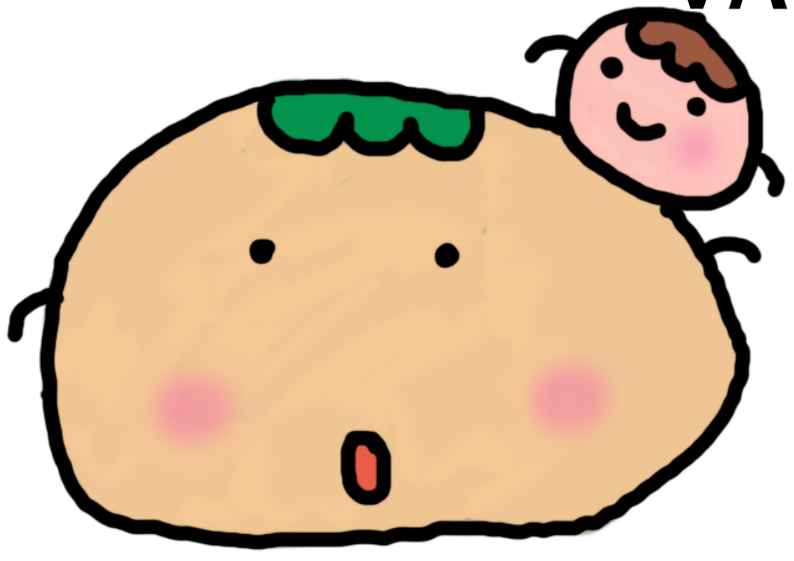
VAE



膠囊



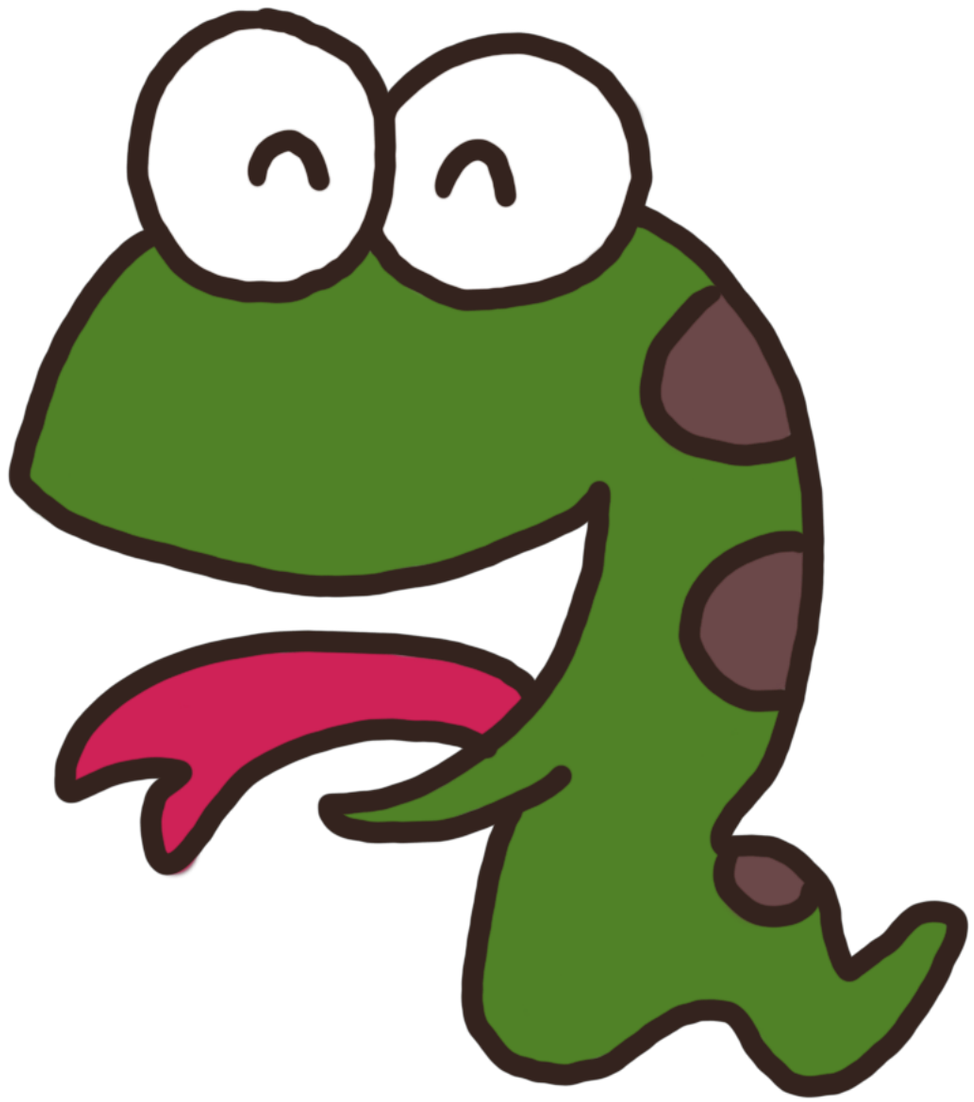
強化學習



生成對抗模式 (GAN)

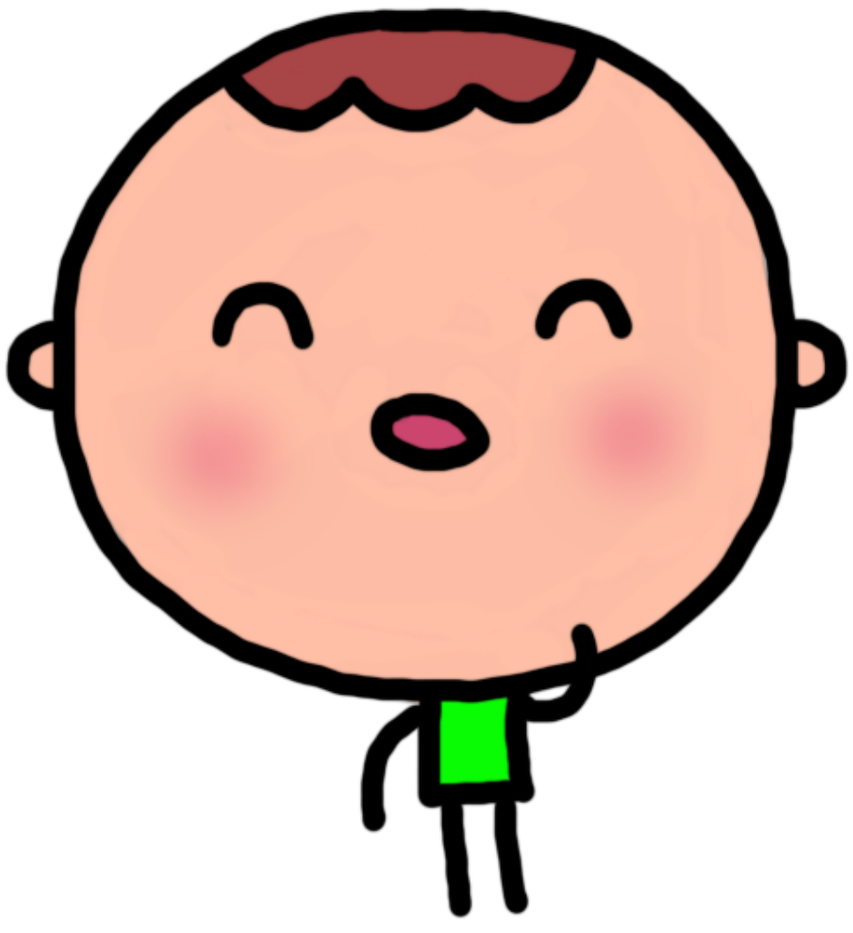
真的把程式寫出來!

Python 程式語言



下載 Anaconda

https://www.anaconda.com/download/

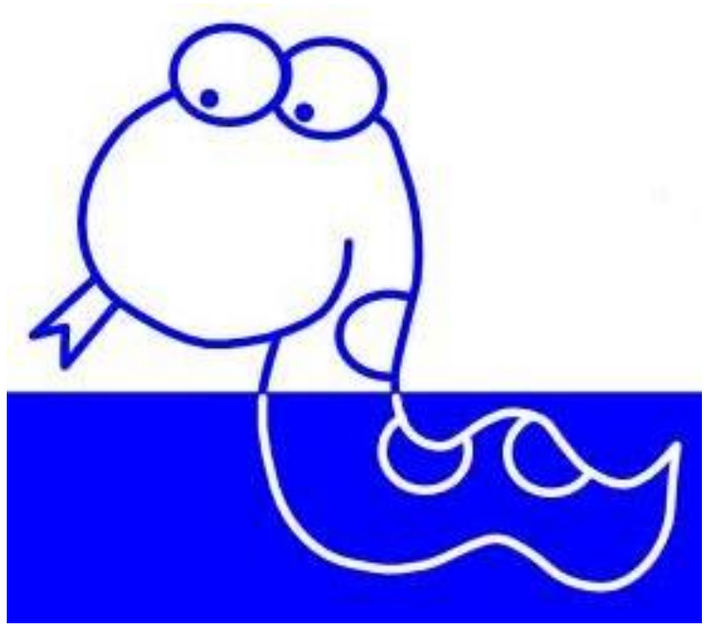


請裝 Python 3 的版本。

3.6

moocs.nccu.edu.tw

moocs.nccu.edu.tw



政大數理資訊學程

fb.me/nccumit

魔法程式家 FB 社團

fb.me/yenlung