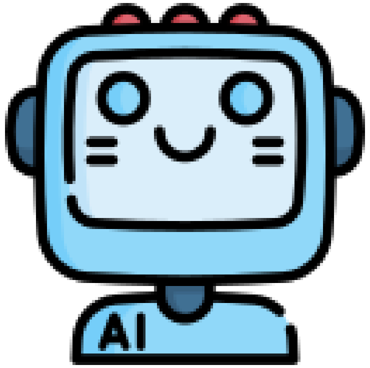
請各位同學稍待片刻

我們 14:23 開始上課

一堂課搞懂
機器學習和深度學習
的基本概念

生成式人工智慧的基本原理

未完成的句子 →  → 下一個 Token

x f $f(x)$

如何根據資料找出函式 f

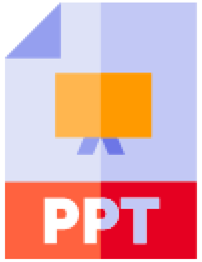
<u>**機器學習 (Machine Learning)**</u>

課程規劃

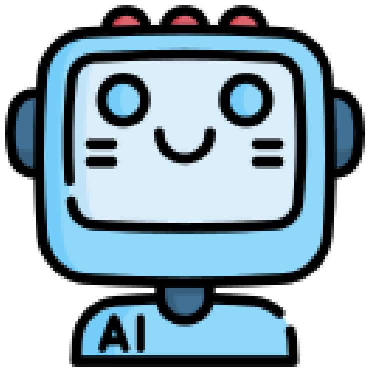
原理

實作

可以找各式各樣的函式



x



f

Regression

數字
(這堂課的長度)

$f(x)$

如何根據資料找出函式 f

這個函式有什麼用呢？

- 這個函式 f 回答一個關鍵問題

相信大家上課常常都會想的 ......

找函式步驟 3 + 1

步驟一：
我要找什麼

\+

步驟二：
我有哪些選擇

→

步驟三：
選一個最好的

學習 (Learning)、訓練 (Training)

找函式步驟 3 + 1

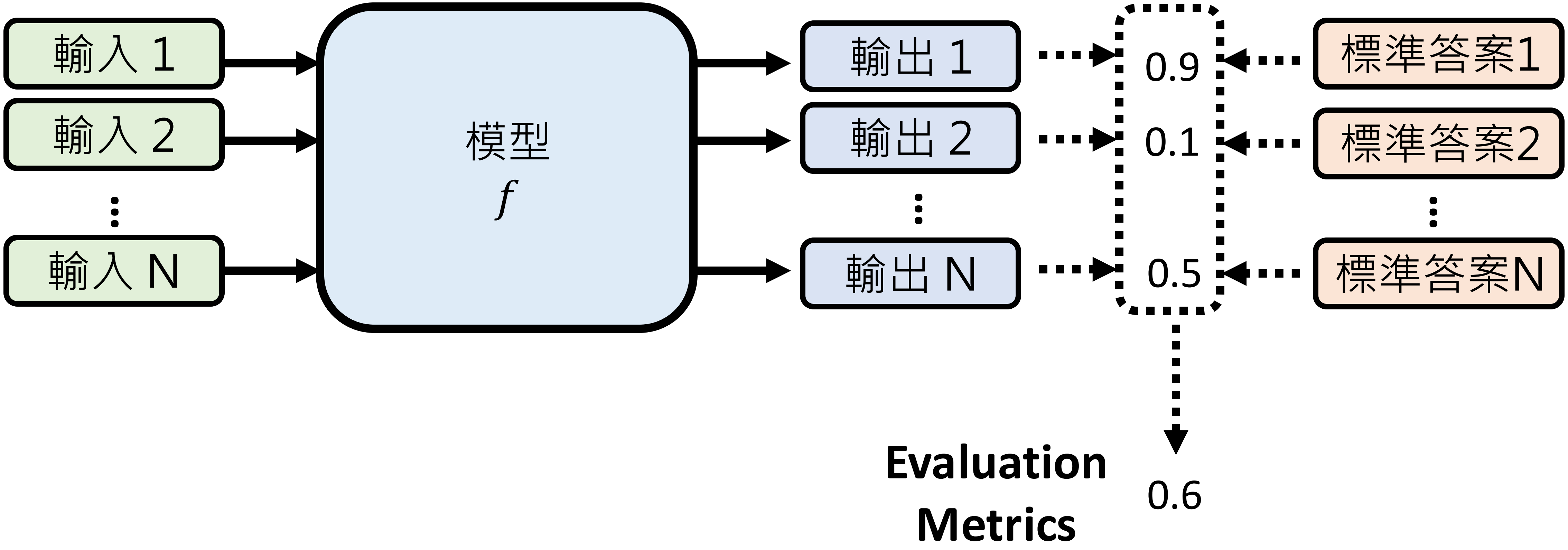
步驟一：
我要找什麼

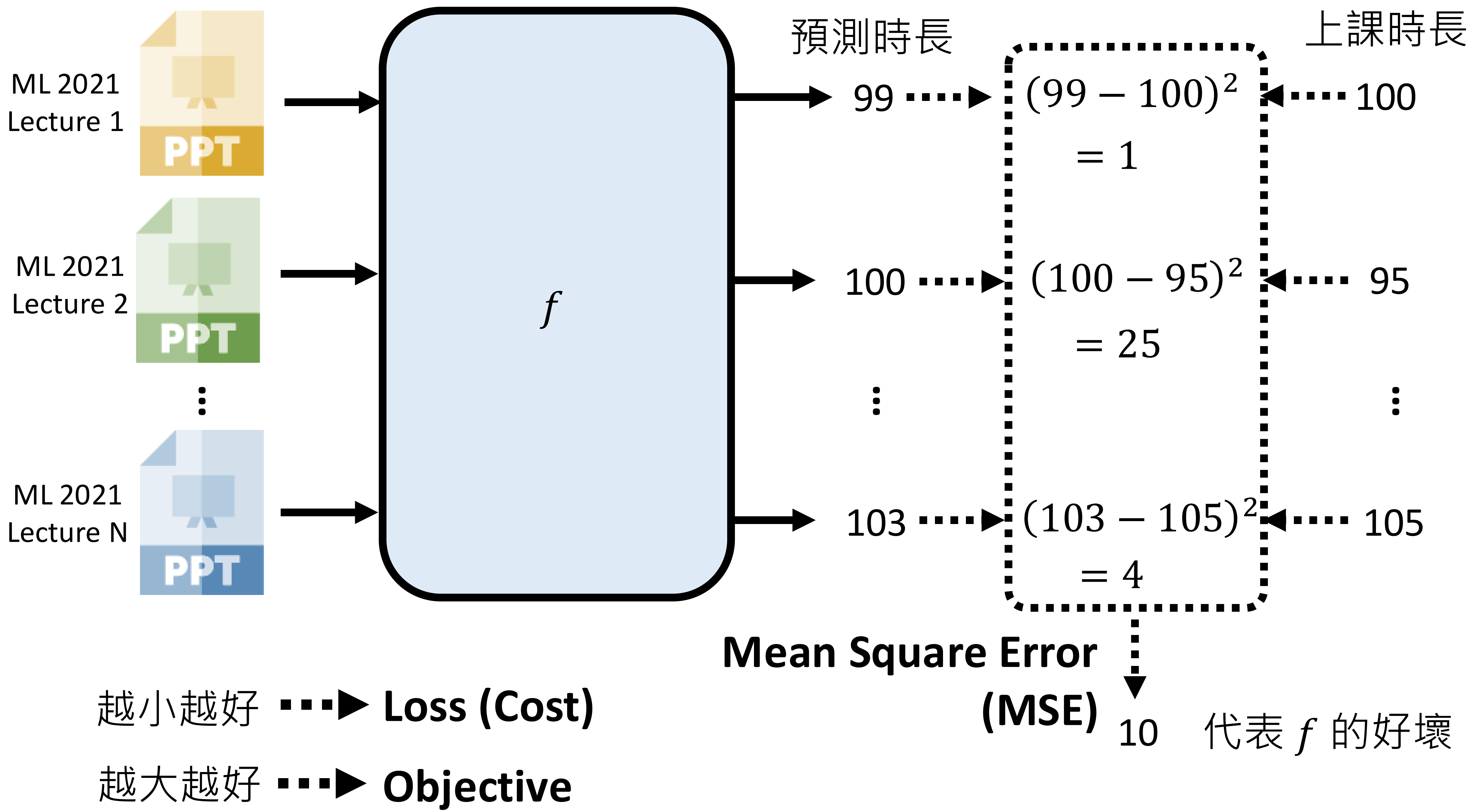
步驟二：
我有哪些選擇

步驟三：
選一個最好的

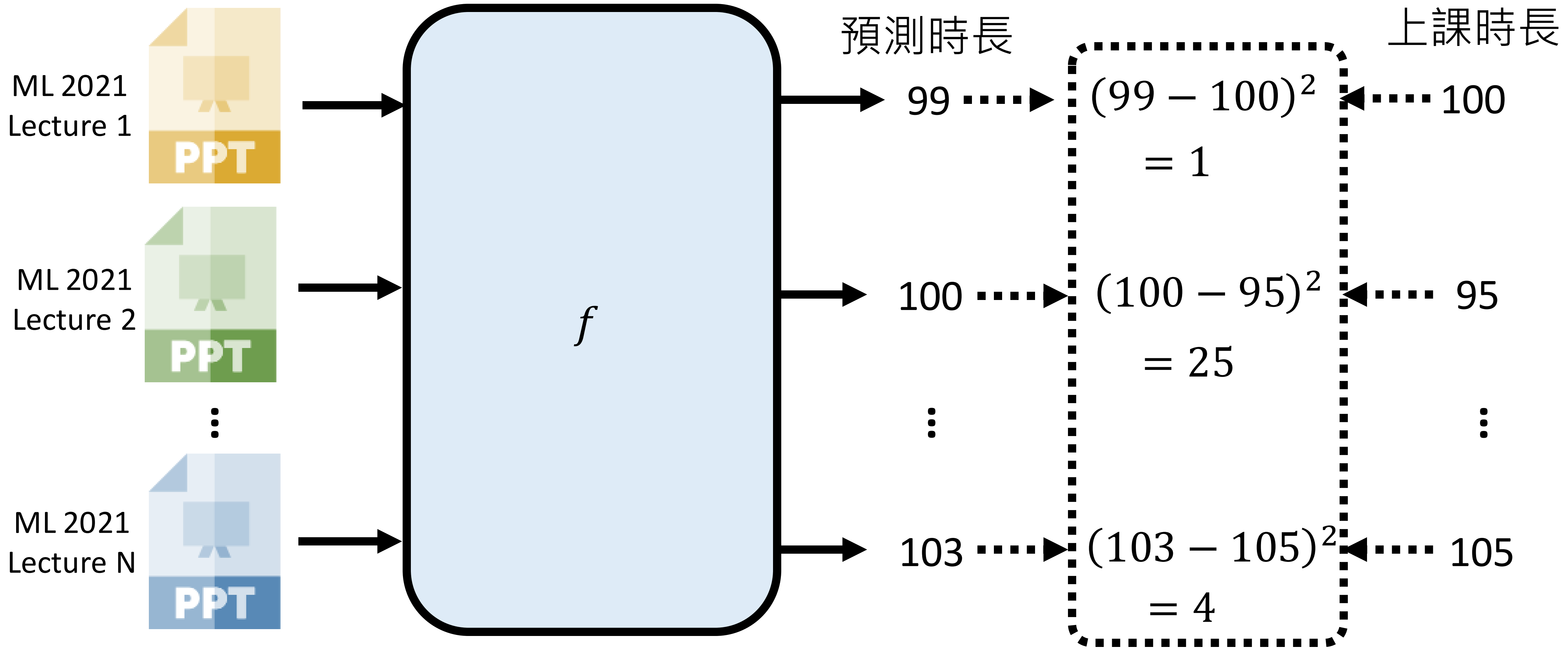
給我一個 f ，我要知道它是不是我要的

上一講：生成式人工智慧的能力檢定





ML 2021 Lecture 1
ML 2021 Lecture 2
ML 2021 Lecture N
PPT
PPT
PPT
f
預測時長
上課時長
99
100
$(99-100)^2 = 1$
100
95
$(100-95)^2 = 25$
103
105
$(103-105)^2 = 4$
Mean Square Error (MSE)
10
代表 f 的好壞
越小越好 Loss (Cost)
越大越好 Objective

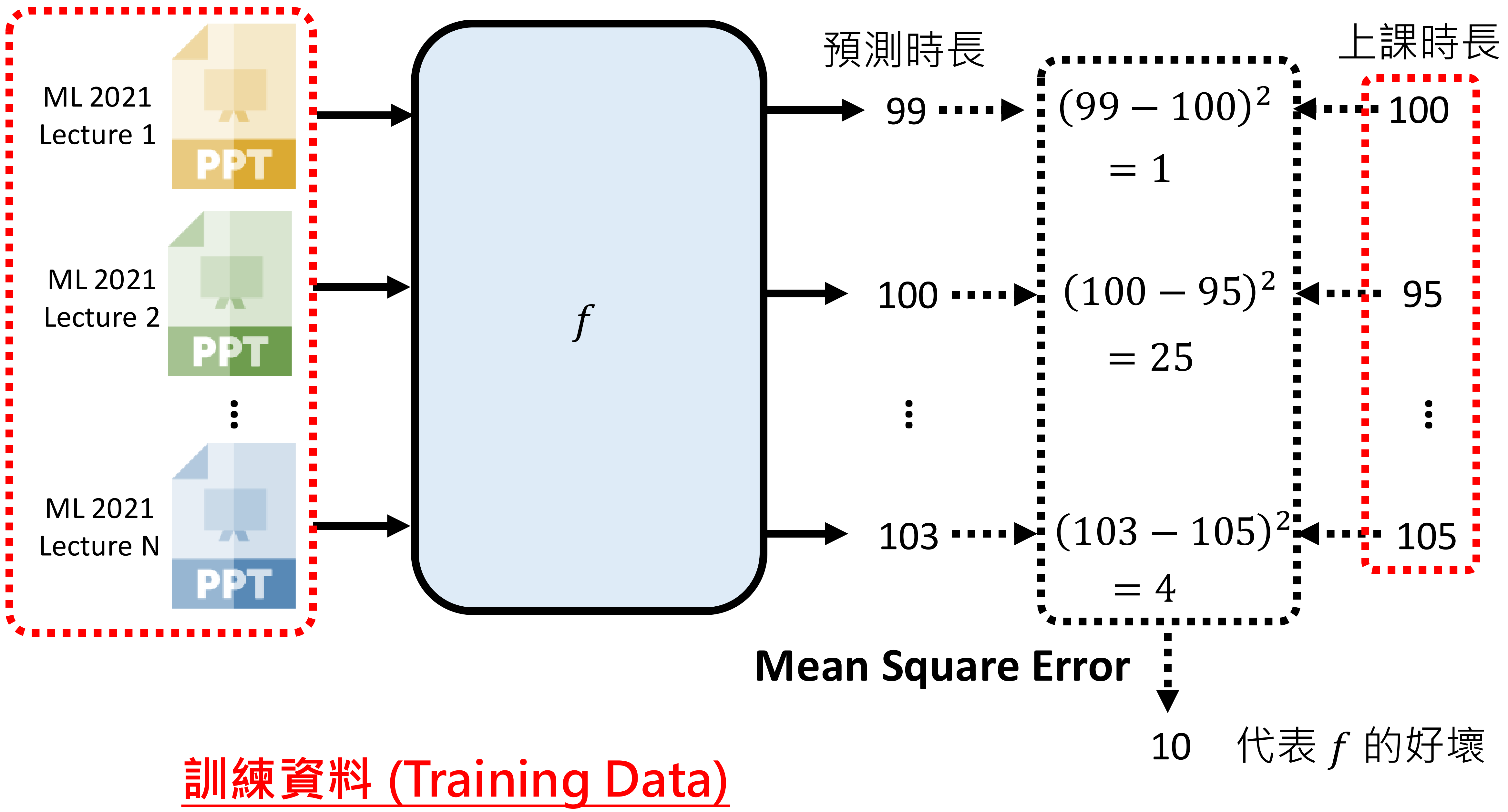


越小越好 ▶ **Loss (Cost)**

越大越好 ▶ **Objective**

這跟 Evaluation 的過程是一樣的

能不能把 Evaluation Metrics 當作 Loss (Objective)？


ML 2021
Lecture 1
PPT
ML 2021
Lecture 2
PPT
ML 2021
Lecture N
PPT
f
預測時長
99
100
103
$(99-100)^2$
$=1$
$(100-95)^2$
$=25$
$(103-105)^2$
$=4$
上課時長
100
95
105
Mean Square Error
10　代表 f 的好壞
訓練資料 (Training Data)

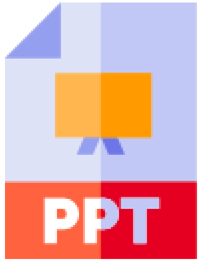
找函式步驟 3 + 1

步驟一：
我要找什麼
(定義出 Loss)

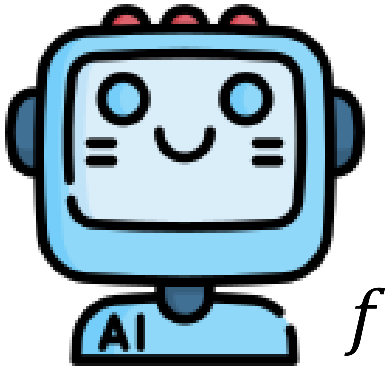
步驟二：
我有哪些選擇

步驟三：
選一個最好的

訂出候選的函式集合



x



f

數字

y

函式的輸入只
能是數字

Feature

頁數	x_1	
內容總字數	x_2	
標題長度	x_3	
有沒有 “Learning"	x_4	(0 or 1)
⋮		

<u>Linear Regression</u>

$$y = \underline{w_1} x_1 + \underline{b}$$

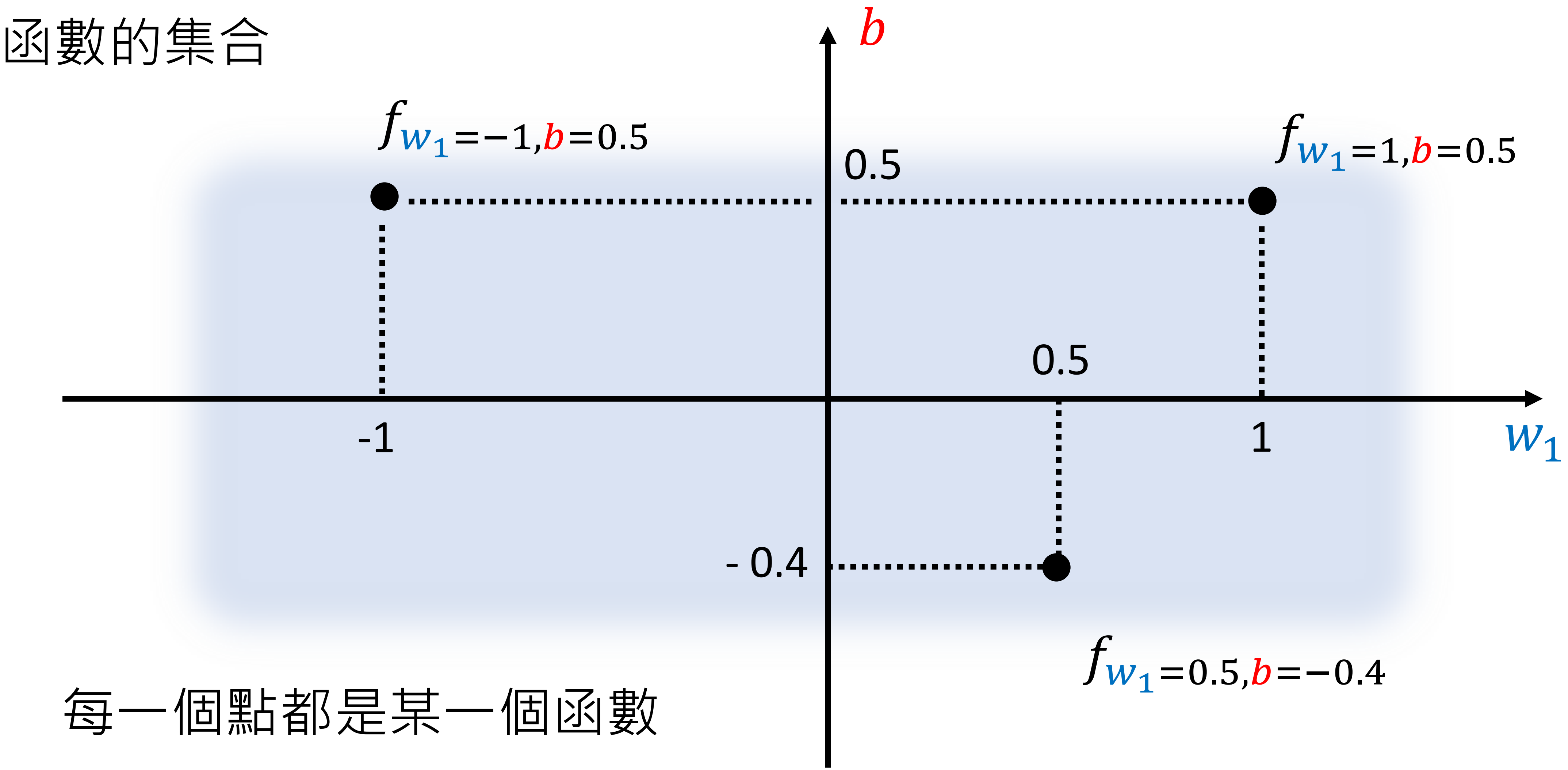
課程長度跟投影片頁數
成某種比例關係

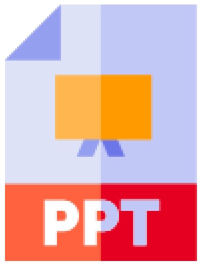
固定多出一點時間
(開場、結尾)

w_1, b 數值未知

參數 (Parameter)

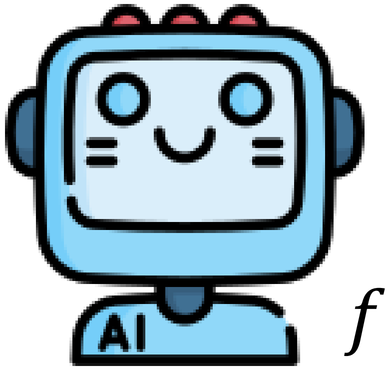
$y = w_1 x_1 + b$ w_1, b 數值未知 **參數 (Parameter)**





x

→



f

→

數字

y

函式的輸入只能是數字

頁數	x_1
內容總字數	x_2
標題長度	x_3
有沒有 “Learning"	x_4
⋮	(0 or 1)

$$y = w_1 x_1 + b$$

出自人類對於任務的理解 (domain knowledge)

模型 (Model)

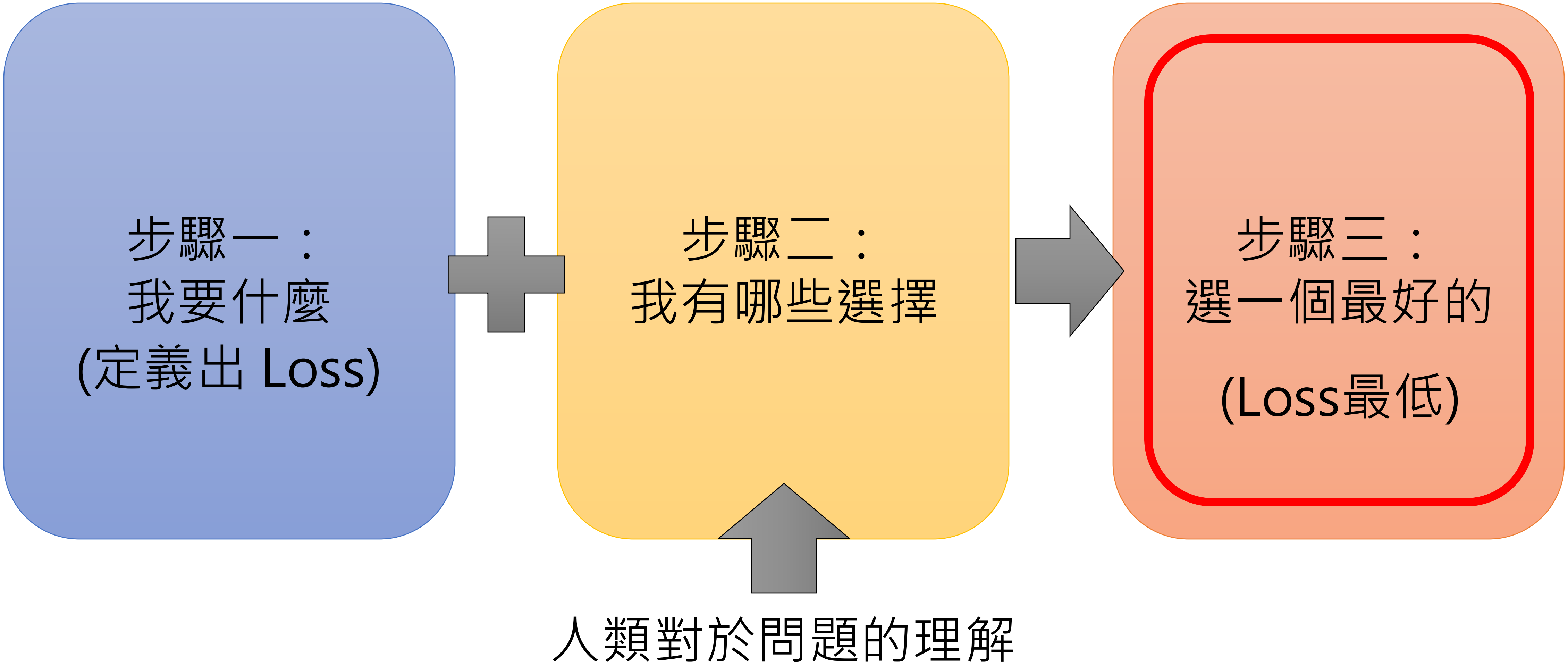
$$y = w_1 x_1 + w_2 x_2 + b$$

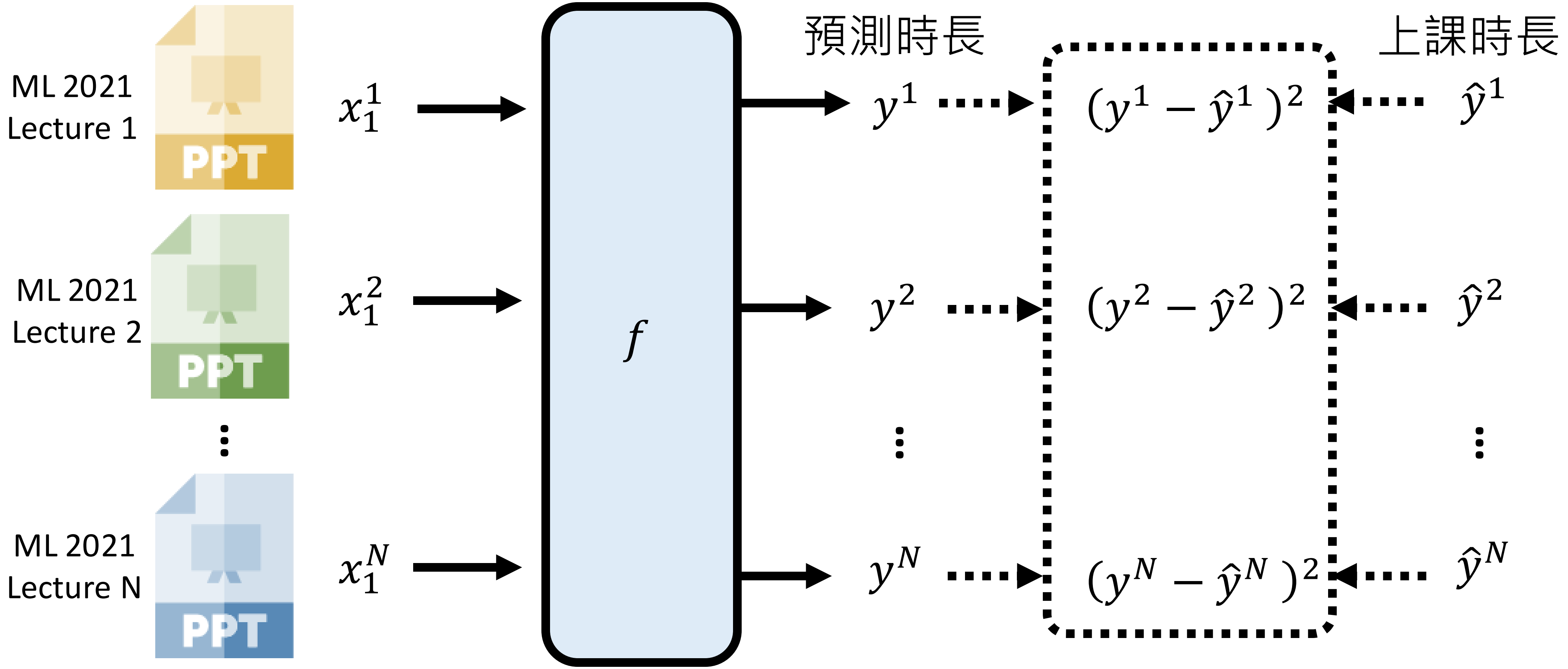
$$y = w_1 x_1 + w_2 x_2 + w_3 x_3 + b$$

$$y = w_1 x_1 + w_2 x_2 x_4 + w_3 {x_3}^2 + b$$

⋮ 有無盡的可能

找函式步驟 3 + 1





先把 Loss 的數學式寫出來

$$L = \frac{1}{N}\sum_{i=1}^{N}\left(y^i - \hat{y}^i\right)^2 = \frac{1}{N}\sum_{i=1}^{N}\left(w_1 x_1^i + b - \hat{y}^i\right)^2$$

$$y = w_1 x_1 + b$$

選一個最好 (Loss最低) 的函式

$$L(w_1, b) = \frac{1}{N}\sum_{i=1}^{N}\left(w_1 x_1^i + b - \hat{y}^i\right)^2$$

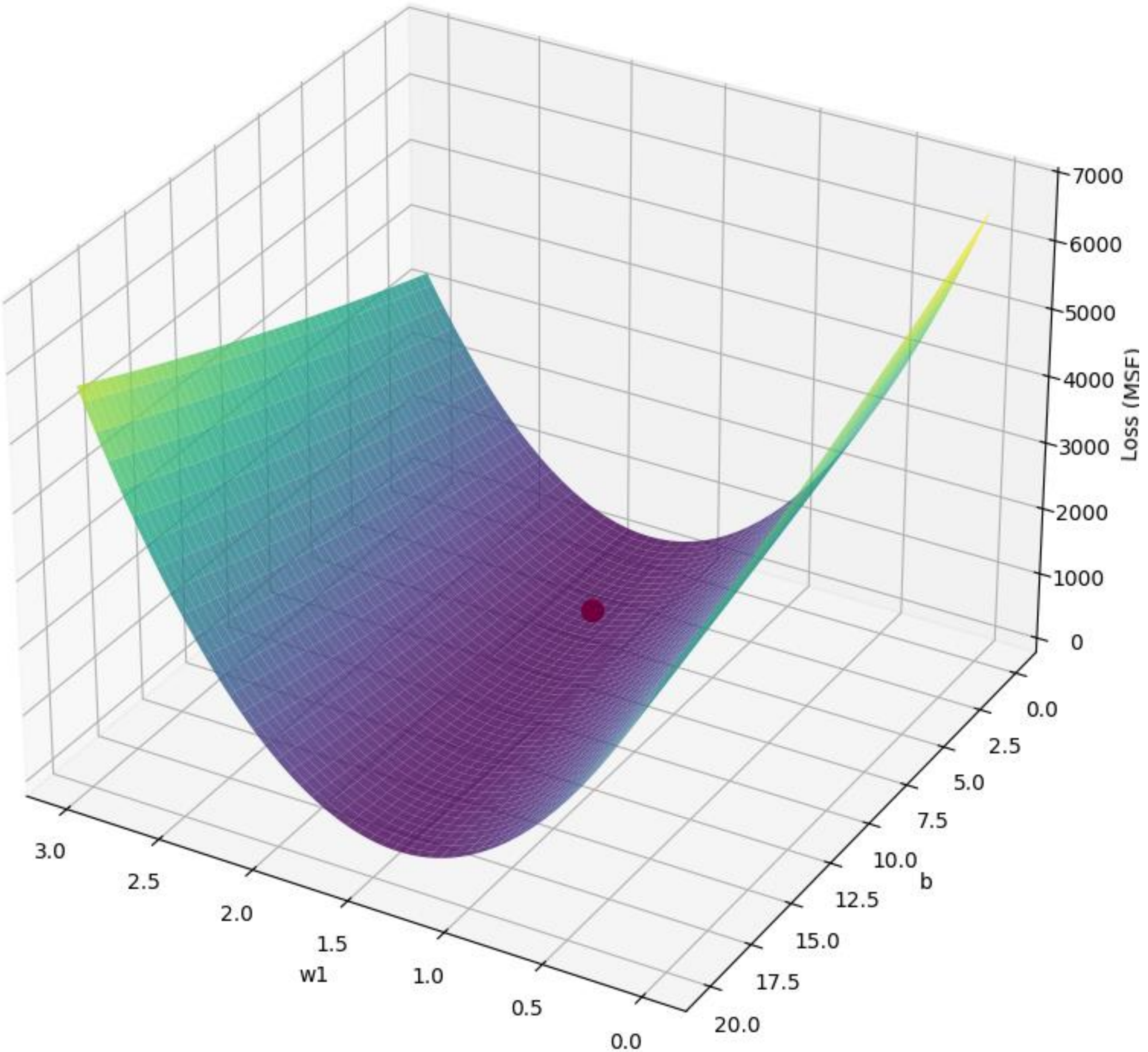
$$w_1^*, b^* = arg\min_{w_1, b} L(w_1, b)$$

這是我們真正要解的問題

Optimization

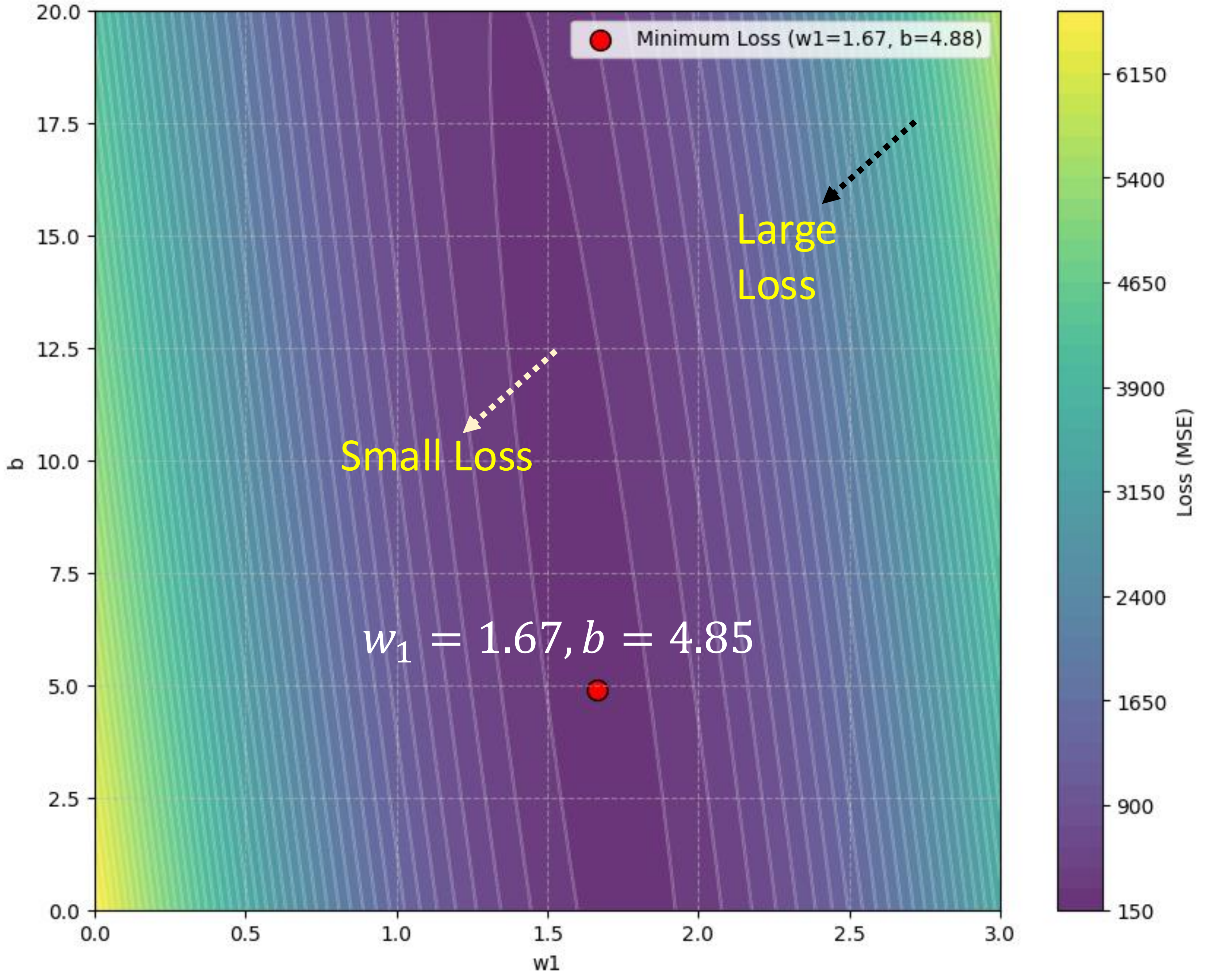
暴力算出所有候選函式的 Loss (MSE)

$$w_1^*, b^* = arg \min_{w_1, b} L(w_1, b)$$



暴力算出所有候選函式的 Loss (MSE)

$$y = w_1 x_1 + b$$



Loss Surface
(Loss 等高線圖)

選一個最好 (Loss最低) 的函式

當 Loss 是 MSE

$$L(w_1, b) = \frac{1}{N}\sum_{i=1}^{N}\left(w_1 x_1^i + b - \hat{y}^i\right)^2$$

當函式集合寫成這樣

$$y = w_1 x_1 + b$$

Linear Regression

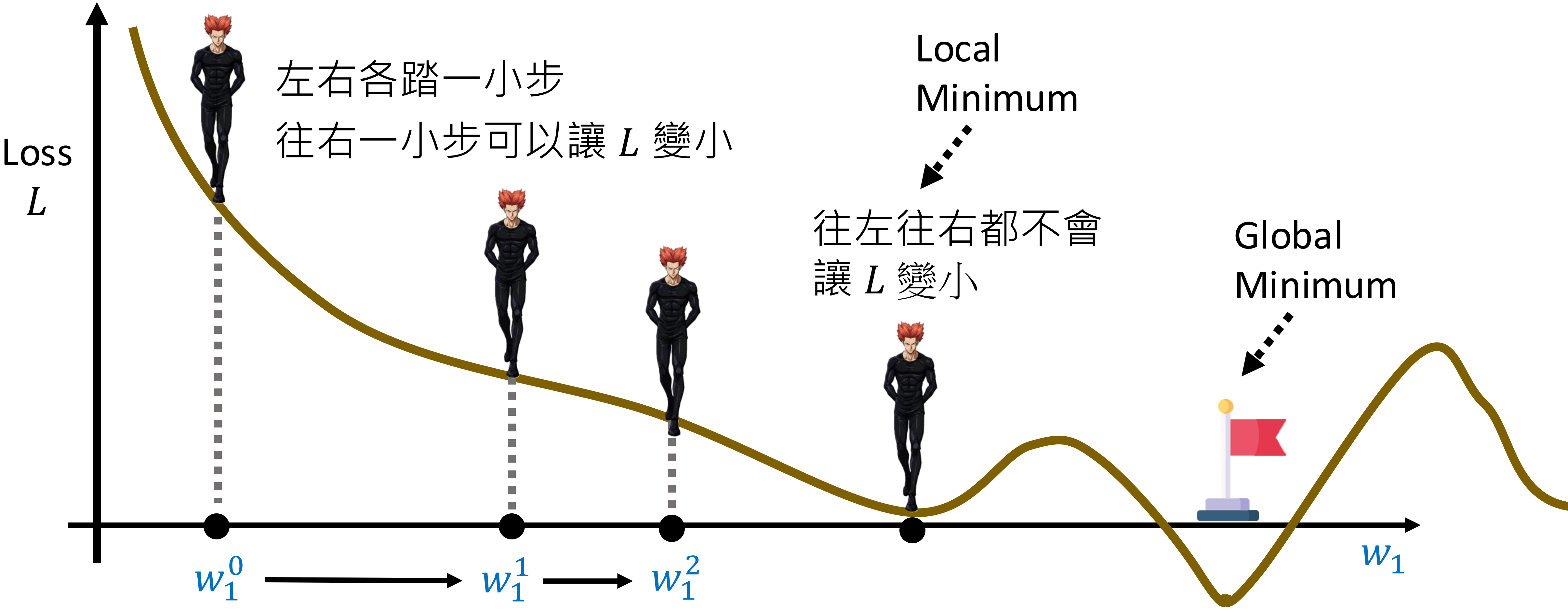
$$w_1^*, b^* = arg\min_{w_1, b} L(w_1, b)$$

線性代數告訴我們有這個問題有 Closed-form Solution (有公式解)

我們需要更通用的做法

Gradient Descent
梯度下降法

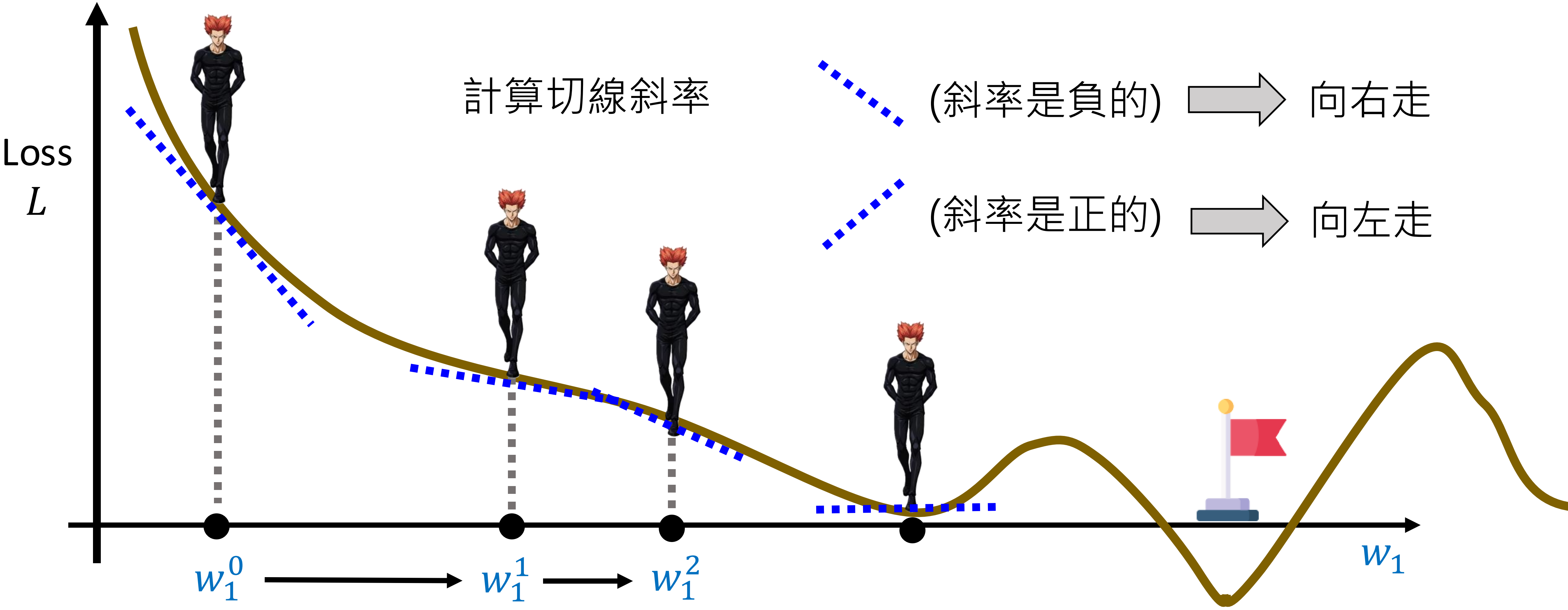
w_1^*, [redacted] $= arg \min_{w_1}$ [redacted] $L(w_1,$ [redacted]$)$



Gradient Descent

梯度下降法

$$w_1^*, \text{[redacted]} = arg \min_{w_1, \text{[redacted]}} L(w_1, \text{[redacted]})$$

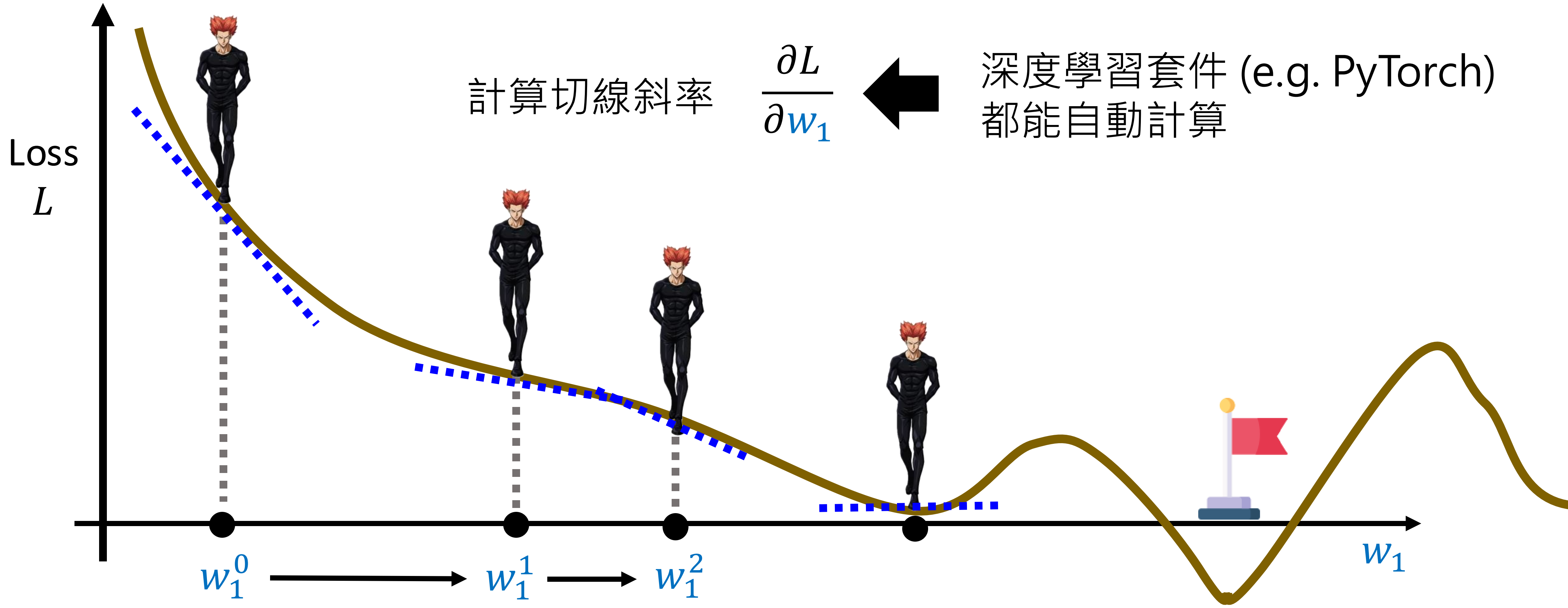


Gradient Descent

梯度下降法

w_1^*, [redacted] $= arg \min_{w_1}$ [redacted] $L(w_1,$ [redacted]$)$

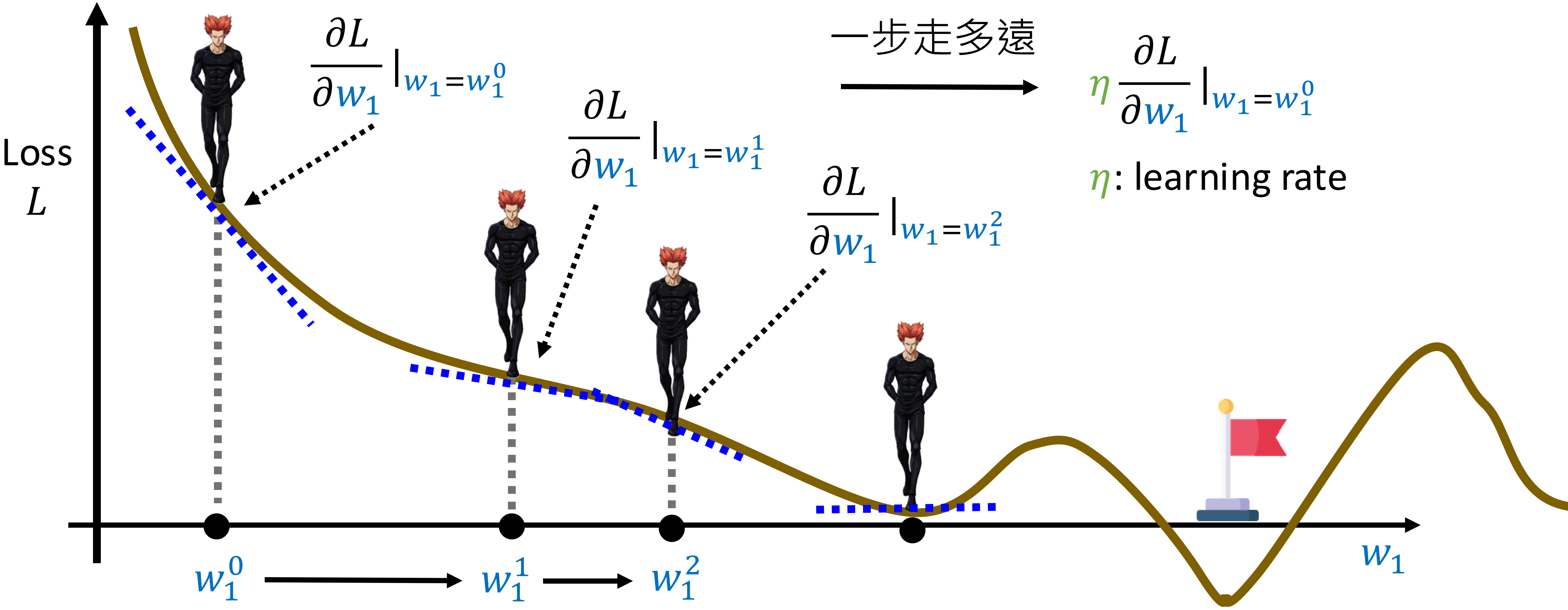
計算切線斜率 $\frac{\partial L}{\partial w_1}$ ⬅ 深度學習套件 (e.g. PyTorch) 都能自動計算



Gradient Descent

梯度下降法

$$w_1^*, \text{[redacted]} = arg \min_{w_1, \text{[redacted]}} L(w_1, \text{[redacted]})$$



Gradient Descent

$$w_1^*, b^* = arg \min_{w_1, b} L(w_1, b)$$

- (Randomly) Pick initial values w_1^0, b^0
- Compute

Gradient

$$\boxed{\begin{array}{c} \frac{\partial L}{\partial w_1}|_{w_1=w_1^0, b=b^0} \\ \frac{\partial L}{\partial b}|_{w_1=w_1^0, b=b^0} \end{array}}$$

$$w_1^1 \leftarrow w_1^0 - \eta \frac{\partial L}{\partial w_1}|_{w_1=w_1^0, b=b^0}$$

$$b^1 \leftarrow b^0 - \eta \frac{\partial L}{\partial b}|_{w_1=w_1^0, b=b^0}$$

Can be done in one line in most deep learning frameworks

- Update w_1 and b interatively

Gradient Descent

$$\boldsymbol{\theta}^* = arg \min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}) \qquad \boldsymbol{\theta} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \vdots \end{bmatrix}$$

- (Randomly) Pick initial values $\boldsymbol{\theta}^0$

$$\boldsymbol{g}^0 = \begin{bmatrix} \frac{\partial L}{\partial \theta_1}|_{\boldsymbol{\theta}=\boldsymbol{\theta}^0} \\ \frac{\partial L}{\partial \theta_2}|_{\boldsymbol{\theta}=\boldsymbol{\theta}^0} \\ \vdots \end{bmatrix} \qquad \begin{bmatrix} \theta_1^1 \\ \theta_2^1 \\ \vdots \end{bmatrix} \leftarrow \begin{bmatrix} \theta_1^0 \\ \theta_2^0 \\ \vdots \end{bmatrix} - \begin{bmatrix} \eta \frac{\partial L}{\partial \theta_1}|_{\boldsymbol{\theta}=\boldsymbol{\theta}^0} \\ \eta \frac{\partial L}{\partial \theta_2}|_{\boldsymbol{\theta}=\boldsymbol{\theta}^0} \\ \vdots \end{bmatrix}$$

gradient

$$\boldsymbol{g}^0 = \nabla L(\boldsymbol{\theta}^0) \qquad \boldsymbol{\theta}^1 \leftarrow \boldsymbol{\theta}^0 - \eta \boldsymbol{g}^0$$

Gradient Descent

$$\boldsymbol{\theta}^* = arg \min_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$$

- (Randomly) Pick initial values $\boldsymbol{\theta}^0$
- Compute gradient $\boldsymbol{g}^0 = \nabla L(\boldsymbol{\theta}^0)$

$$\boldsymbol{\theta}^1 \leftarrow \boldsymbol{\theta}^0 - \eta \boldsymbol{g}^0 \Longleftarrow \text{1 iteration (1 update)}$$

- Compute gradient $\boldsymbol{g}^{\mathbf{1}} = \nabla L(\boldsymbol{\theta}^1)$

$$\boldsymbol{\theta}^2 \leftarrow \boldsymbol{\theta}^1 - \eta \boldsymbol{g}^1$$

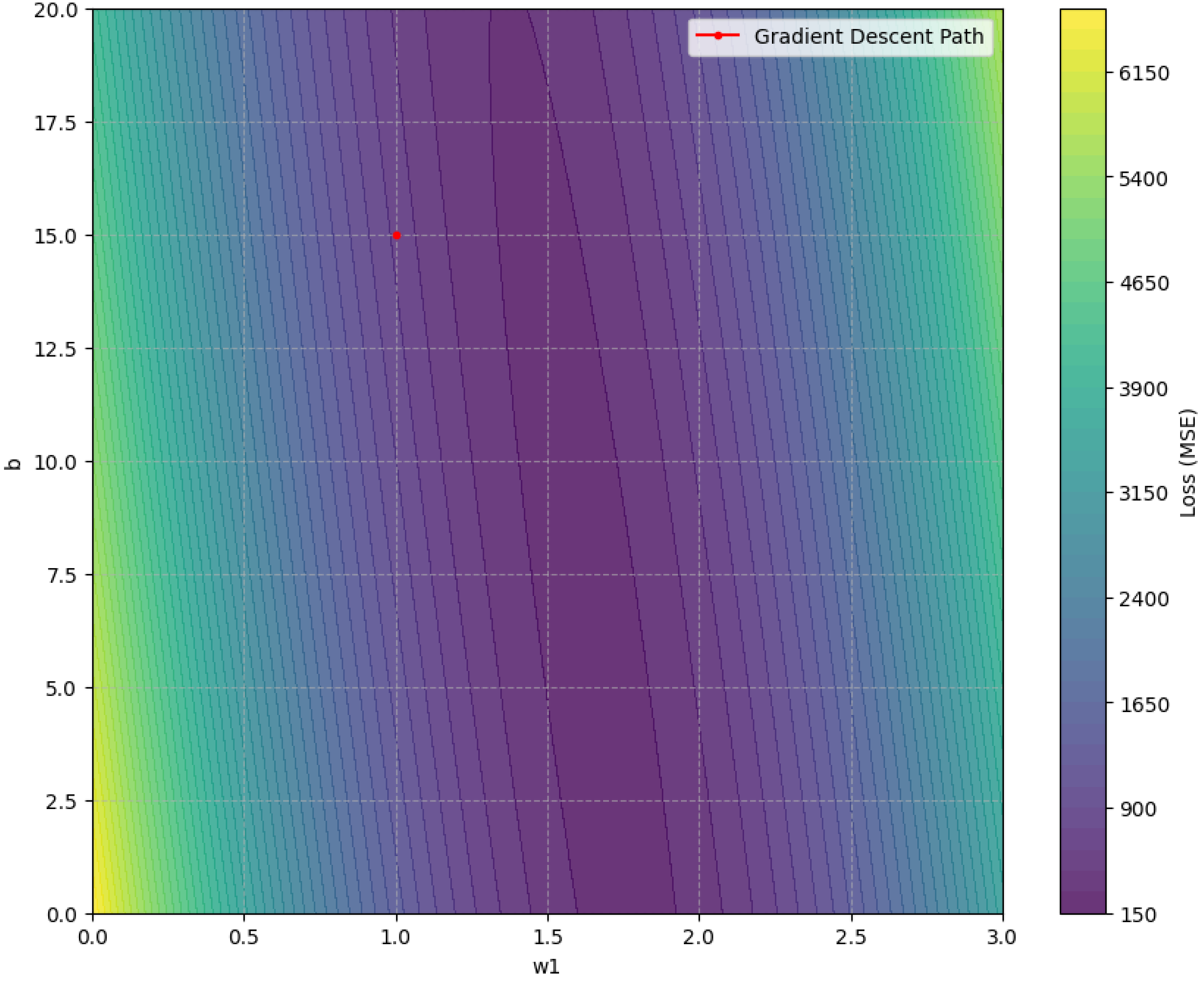
- Compute gradient $\boldsymbol{g}^2 = \nabla L(\boldsymbol{\theta}^2)$

$$\boldsymbol{\theta}^3 \leftarrow \boldsymbol{\theta}^2 - \eta \boldsymbol{g}^2$$

概念很簡單，
做起來不容易

$w_1^0 = 1.0$, $b^0 = 15.0$

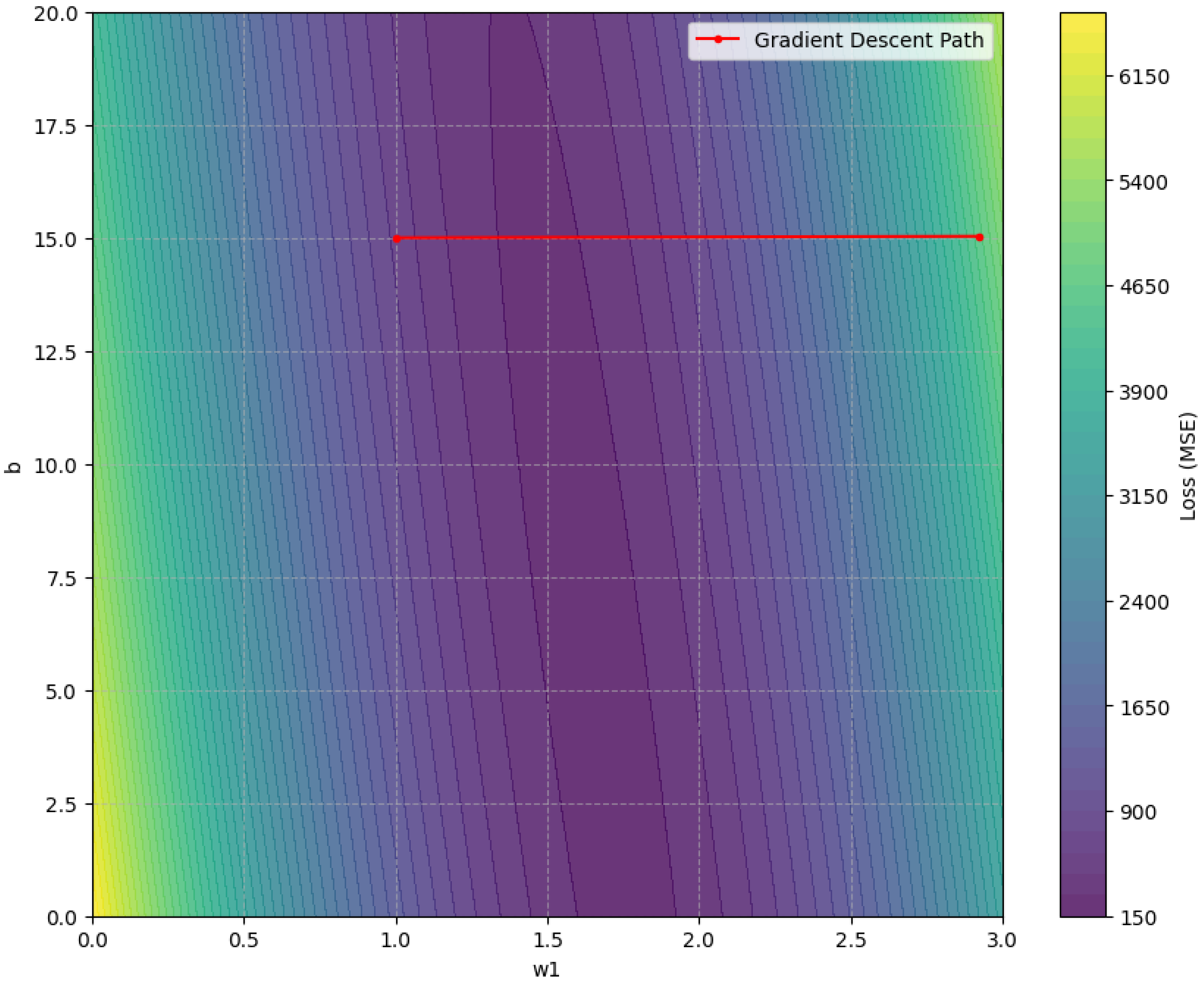
$\eta = 0.001$



概念很簡單，
做起來不容易

$w_1^0 = 1.0$, $b^0 = 15.0$

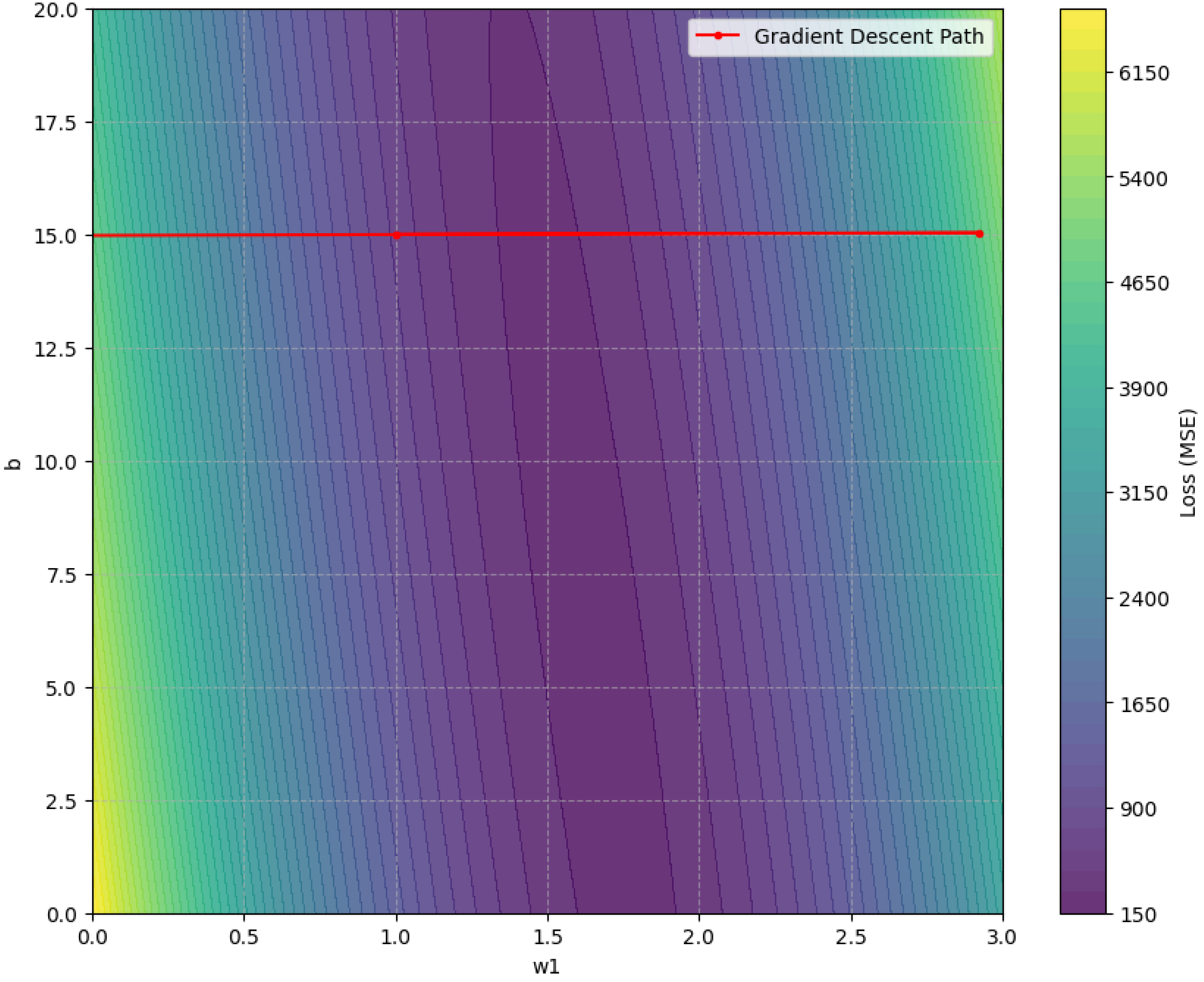
$\eta = 0.001$



概念很簡單，
做起來不容易

$w_1^0 = 1.0$, $b^0 = 15.0$

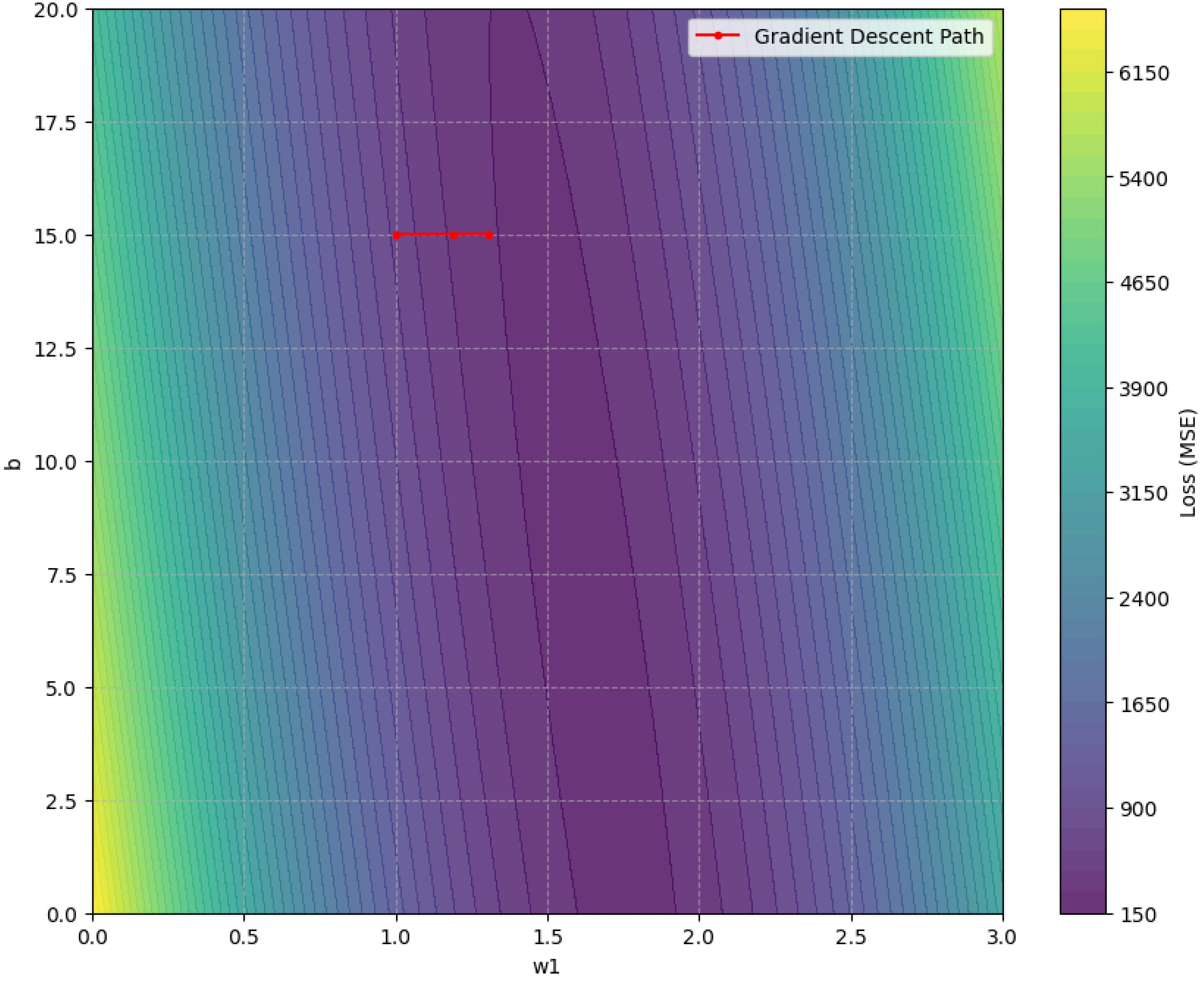
$\eta = 0.001$

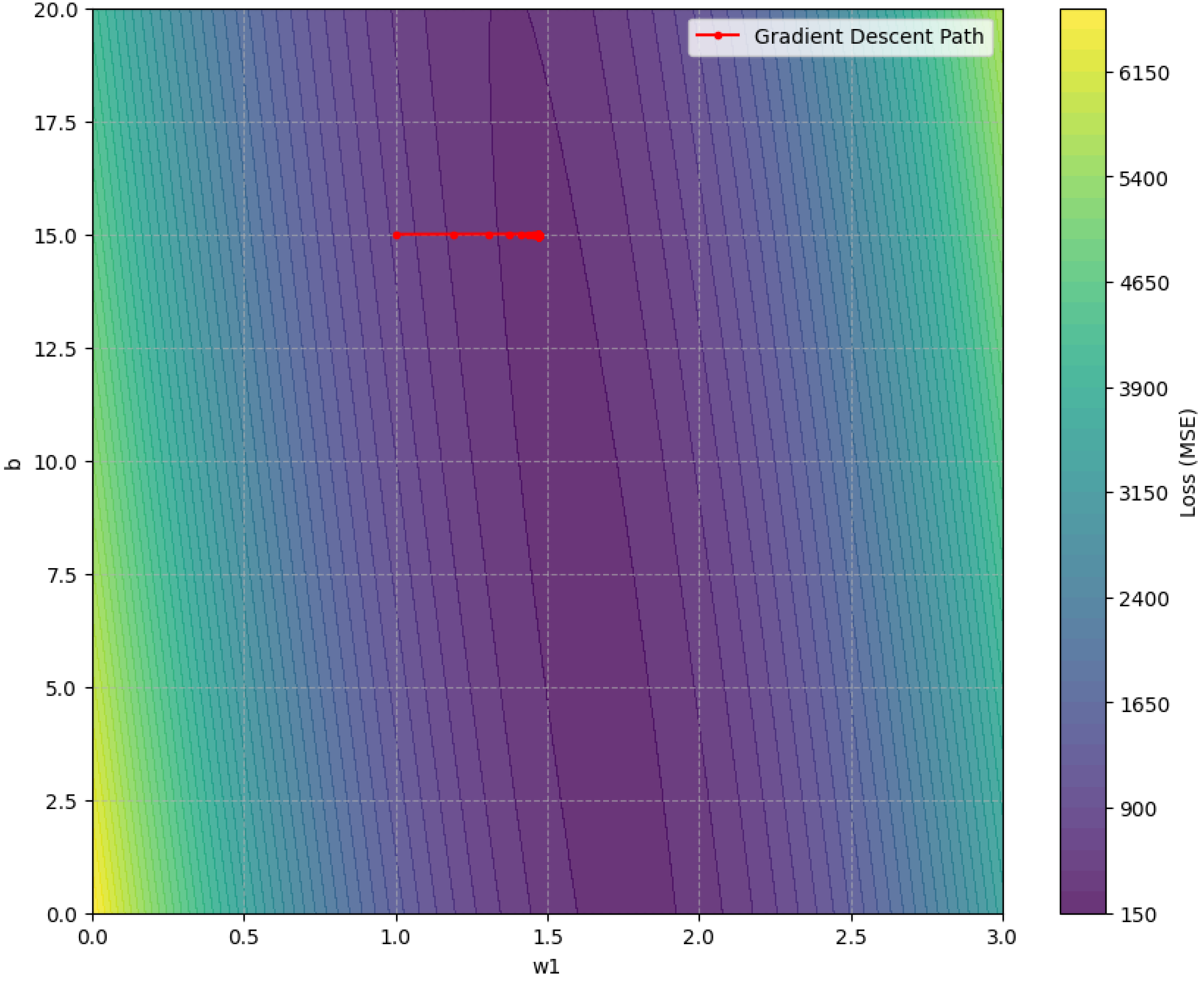


概念很簡單，
做起來不容易

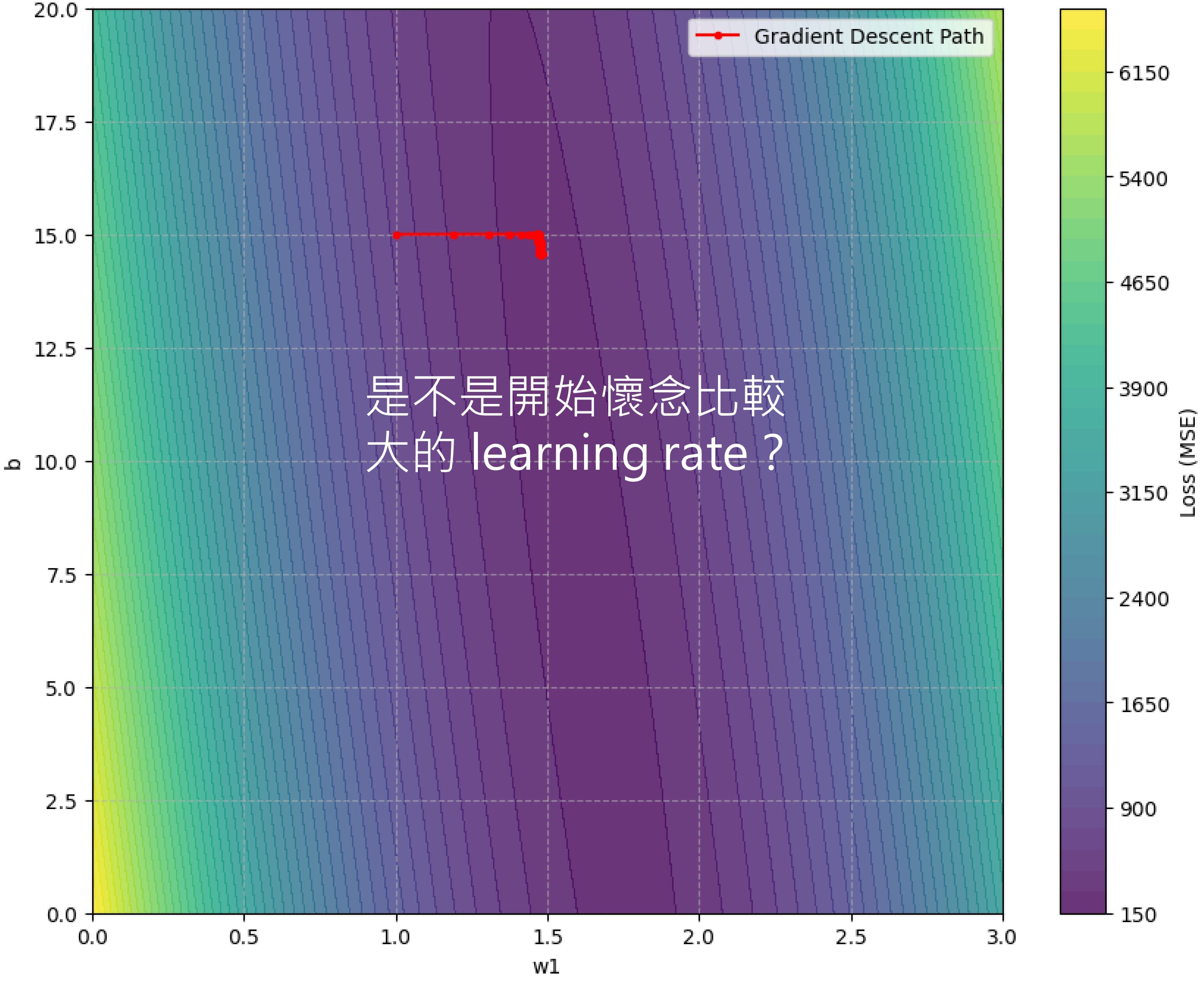
$w_1^0 = 1.0$, $b^0 = 15.0$

$\eta = 0.0001$

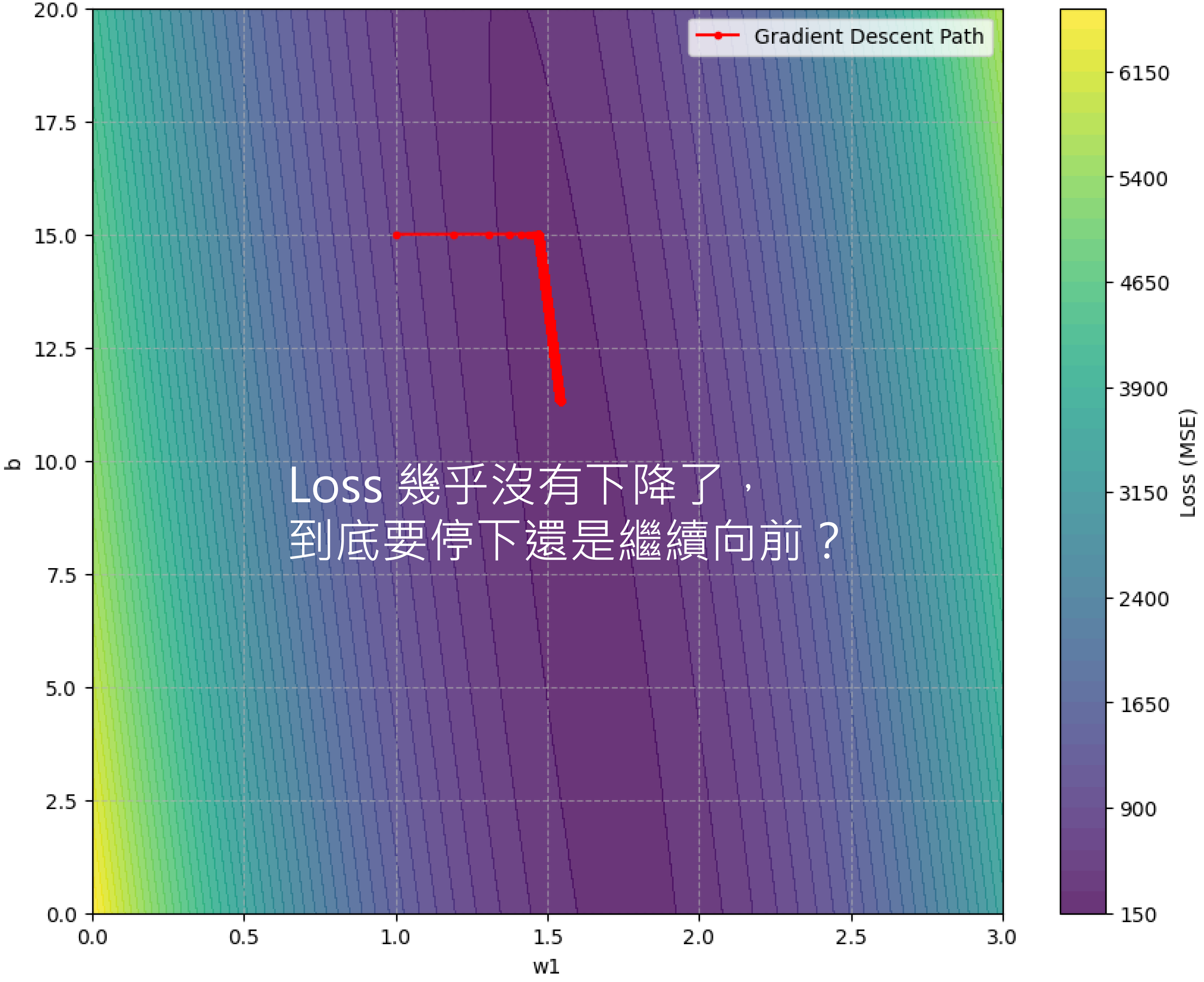




Update 100 times



Update 1,000 times



Update 10,000 times

參數更新太慢？

$$\boldsymbol{\theta}^* = arg \min_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$$

- (Randomly) Pick initial values $\boldsymbol{\theta}^0$
- Compute gradient $\boldsymbol{g}^0 = \nabla L(\boldsymbol{\theta}^0)$

$$\boldsymbol{\theta}^1 \leftarrow \boldsymbol{\theta}^0 - \eta \boldsymbol{g}^0$$

- Compute gradient $\boldsymbol{g}^1 = \nabla L(\boldsymbol{\theta}^1)$

$$\boldsymbol{\theta}^2 \leftarrow \boldsymbol{\theta}^1 - \eta \boldsymbol{g}^1$$

- Compute gradient $\boldsymbol{g}^2 = \nabla L(\boldsymbol{\theta}^2)$

$$\boldsymbol{\theta}^3 \leftarrow \boldsymbol{\theta}^2 - \eta \boldsymbol{g}^2$$

$$L = \frac{1}{N}\sum_{i=1}^{N} \cdots\cdots$$

如果訓練資料很多，要等很久才能更新一次參數

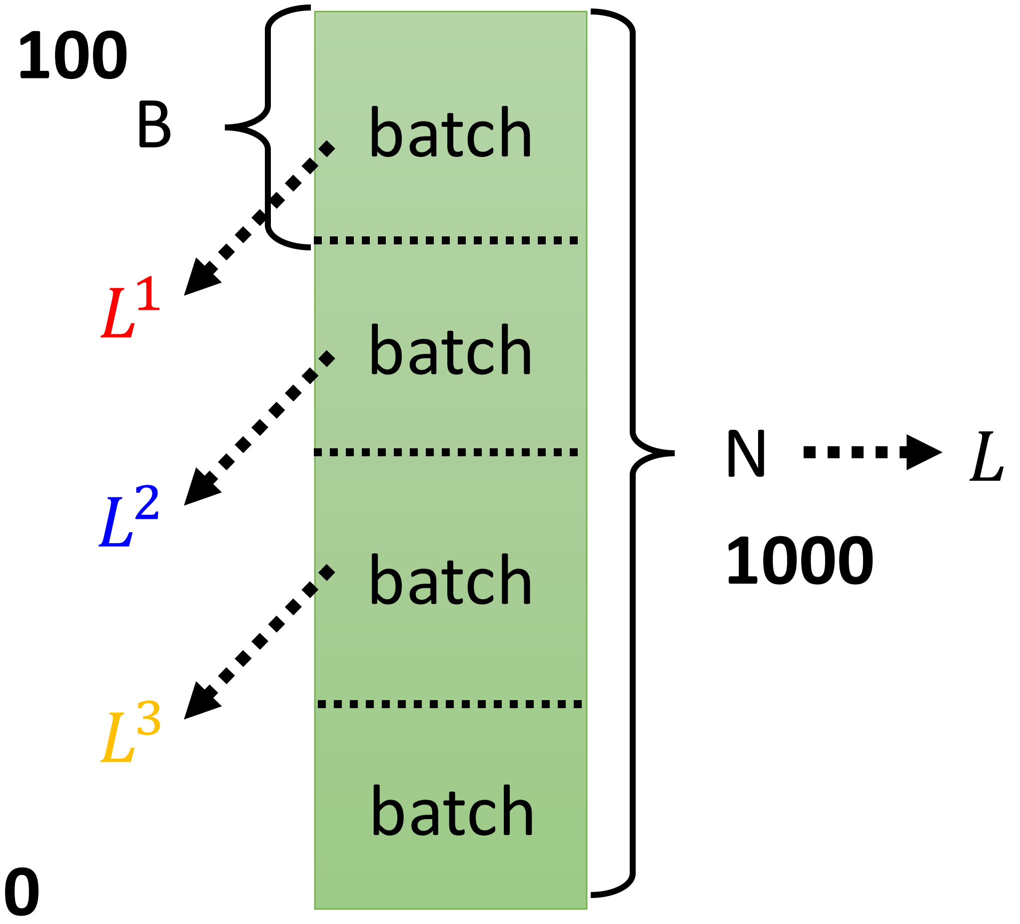
迫不及待更新參數

- (Randomly) Pick initial values $\boldsymbol{\theta}^0$
- Compute gradient $\boldsymbol{g}^0 = \nabla L^1(\boldsymbol{\theta}^0)$

  $$\boldsymbol{\theta}^1 \leftarrow \boldsymbol{\theta}^0 - \eta \boldsymbol{g}^0$$
- Compute gradient $\boldsymbol{g}^1 = \nabla L^2(\boldsymbol{\theta}^1)$

  $$\boldsymbol{\theta}^2 \leftarrow \boldsymbol{\theta}^1 - \eta \boldsymbol{g}^1$$
- Compute gradient $\boldsymbol{g}^2 = \nabla L^3(\boldsymbol{\theta}^2)$

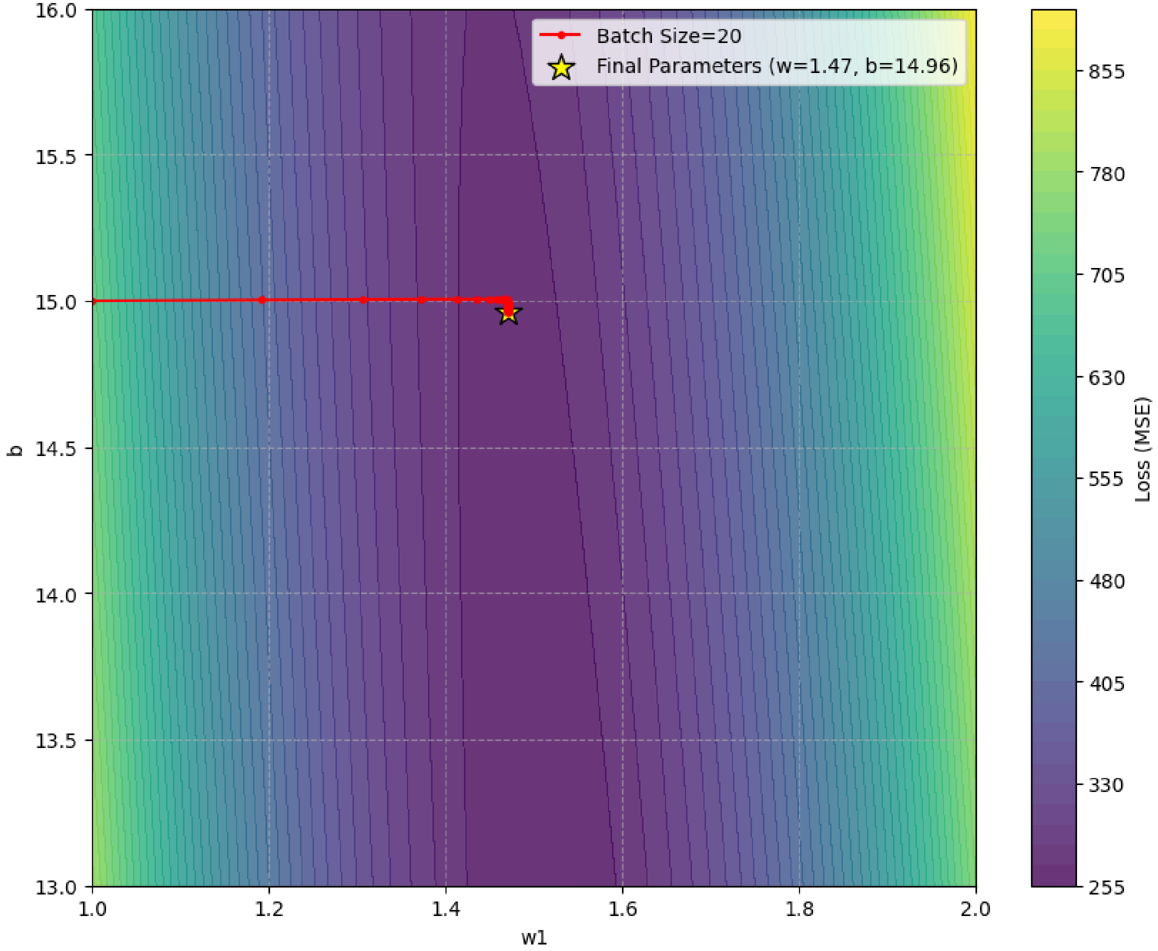
  $$\boldsymbol{\theta}^3 \leftarrow \boldsymbol{\theta}^2 - \eta \boldsymbol{g}^2$$



1 **epoch** = see all the batches once

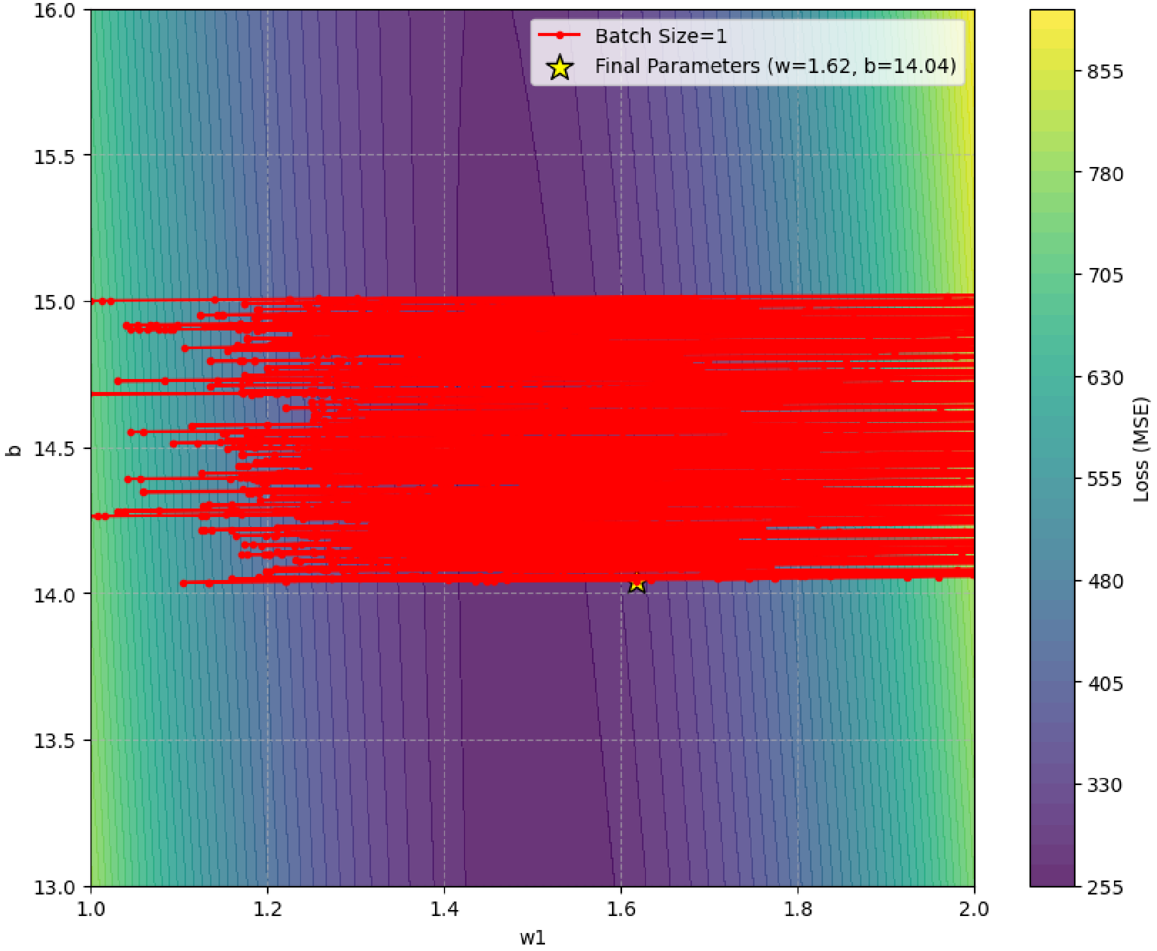
N/B updates (iterations) in one epoch

epochs = 100



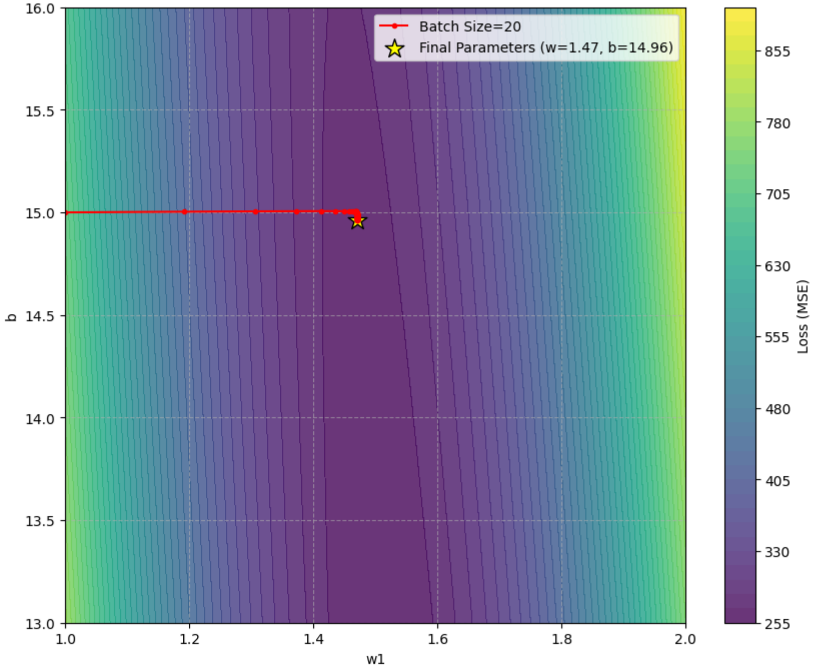
Batch size = all training data

(Full Batch)

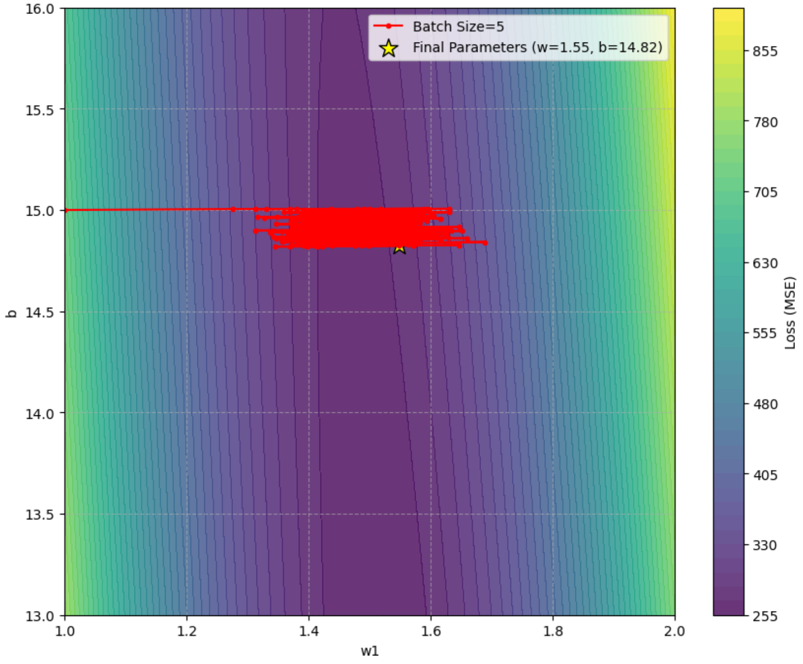


Batch size = 1

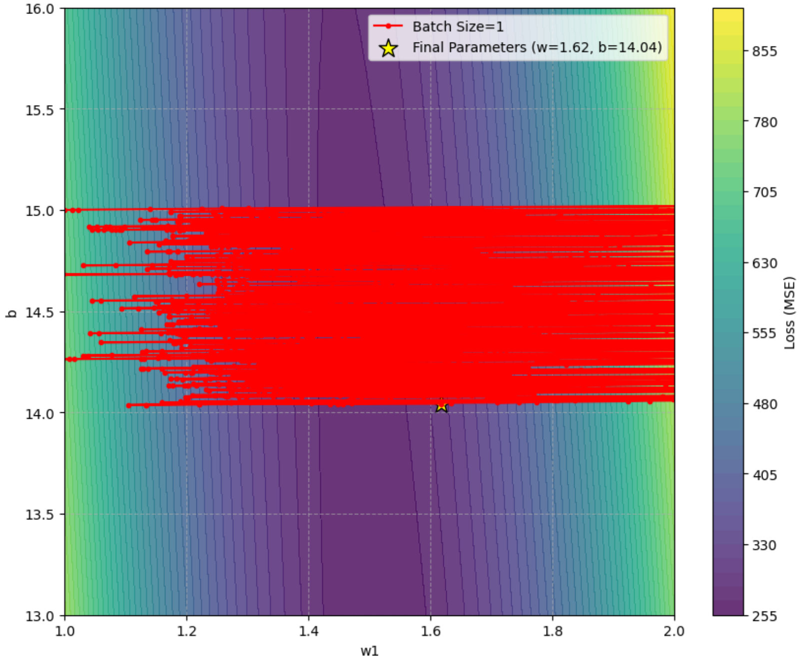
(Stochastic Gradient Descent, SGD)



Batch size
= all training data



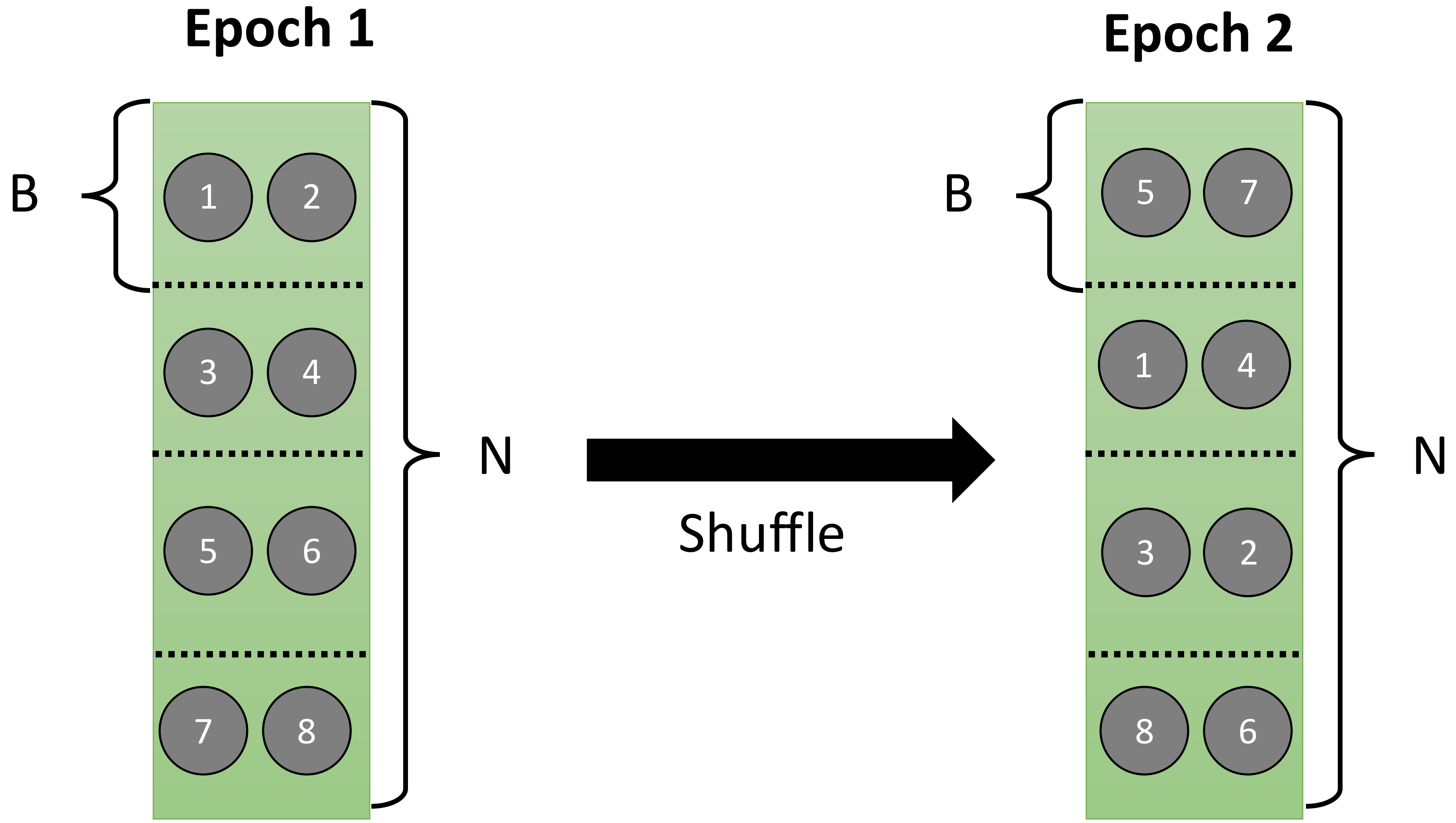
Batch size
= 5



Batch size
= 1

又多了一個可以調的 hyperparameter

Shuffle



步驟一：
我要什麼

+

步驟二：
我有哪些選擇

步驟三：
選一個最好的

$$L = \frac{1}{N}\sum_{i=1}^{N}\left(y^i - \hat{y}^i\right)^2$$ MSE

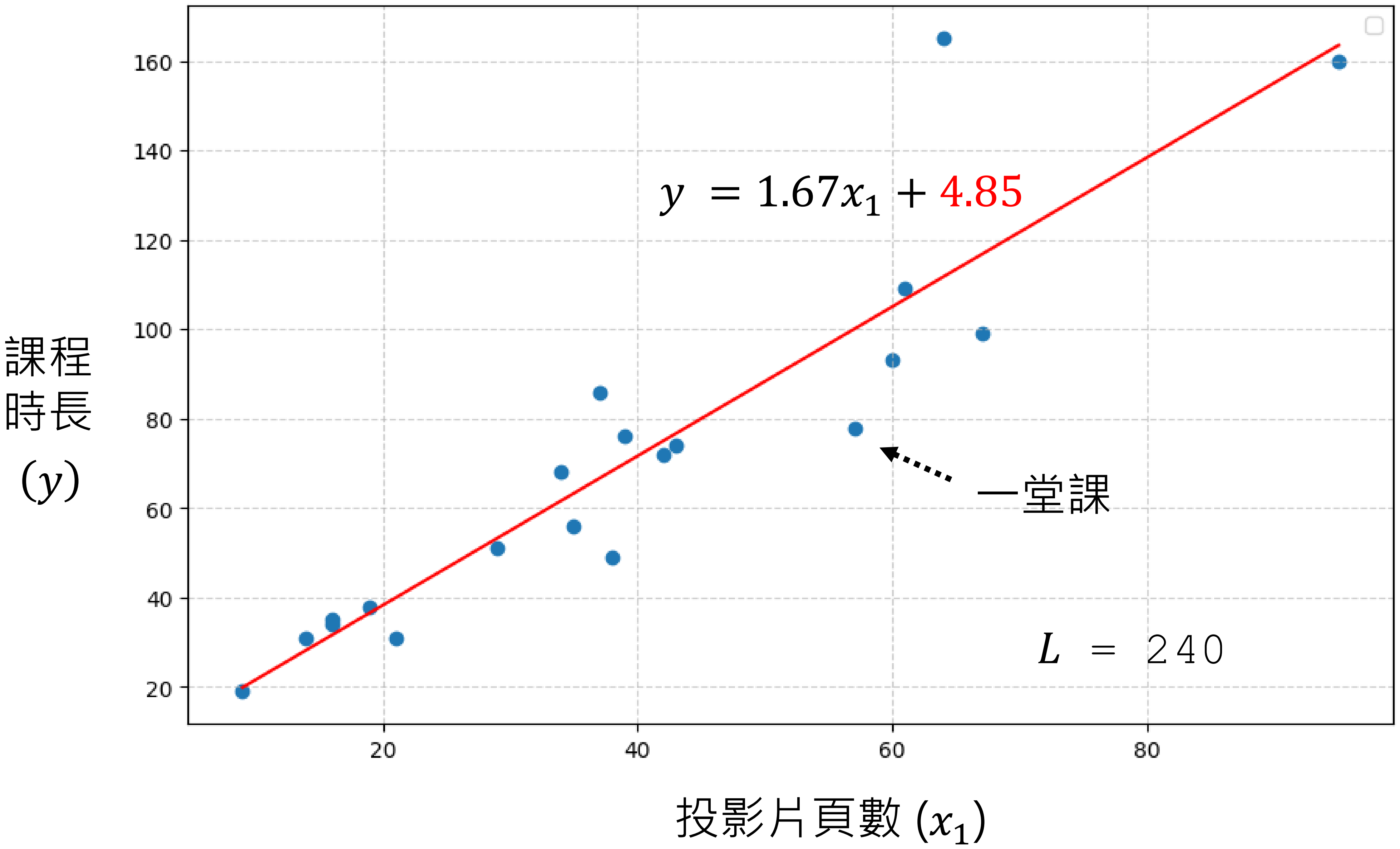
$$L(w_1^*, b^*) = 240$$

$$y = w_1 x_1 + b$$

$$y = 1.67 x_1 + 4.85$$

$$w_1^*, b^* = arg \min_{w_1, b} L(w_1, b)$$

$$w_1^* = 1.67, b^* = 4.85$$



課程時長 (y)
投影片頁數 (x_1)
$y = 1.67x_1 + 4.85$
一堂課
$L = 240$
160
140
120
100
80
60
40
20
20
40
60
80

步驟一：
我要什麼

+

步驟二：
我有哪些選擇

步驟三：
選一個最好的

$$L = \frac{1}{N}\sum_{i=1}^{N}\left(y^i - \hat{y}^i\right)^2$$ MSE

$$y = w_1 x_1 + b$$

$$w_1^*, b^* = arg \min_{w_1, b} L(w_1, b)$$

$$L(w_1^*, b^*) = 240$$

$$y = 1.67x_1 + 4.85$$

$$w_1^* = 1.67, b^* = 4.85$$

測試今天這堂課

測試 (Testing)

步驟一：
我要什麼

$$L = \frac{1}{N}\sum_{i=1}^{N}\left(y^i - \hat{y}^i\right)^2$$ MSE

$$L(w_1^*, b^*) = 240$$

步驟二：
我有哪些選擇

$$y = w_1 x_1 + b$$

$$y = 1.67 x_1 + 4.85$$

步驟三：
選一個最好的

$$w_1^*, b^* = arg \min_{w_1, b} L(w_1, b)$$

$$w_1^* = 1.67, b^* = 4.85$$

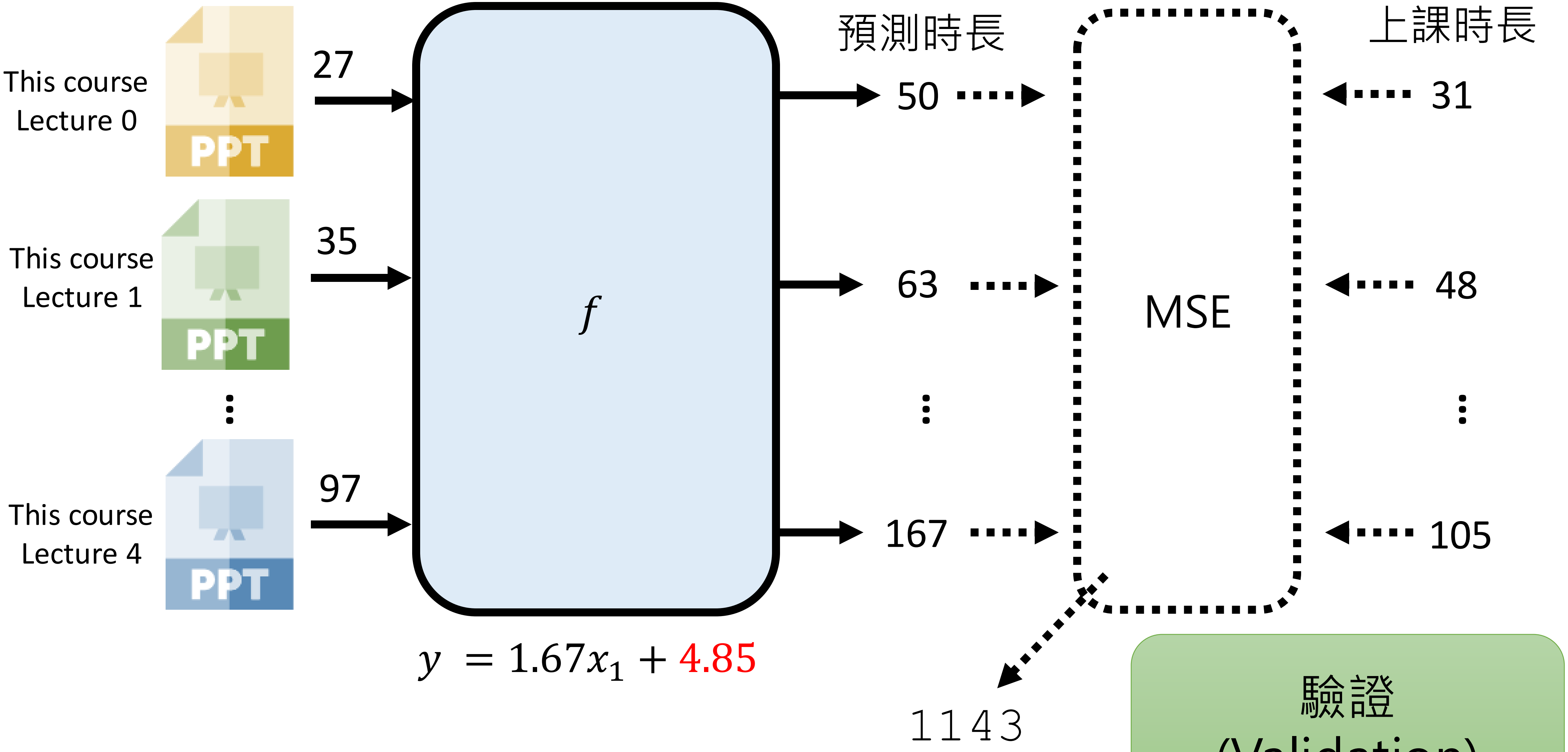
驗證
(Validation)

模擬考

測試今天這堂課

測試 (Testing)

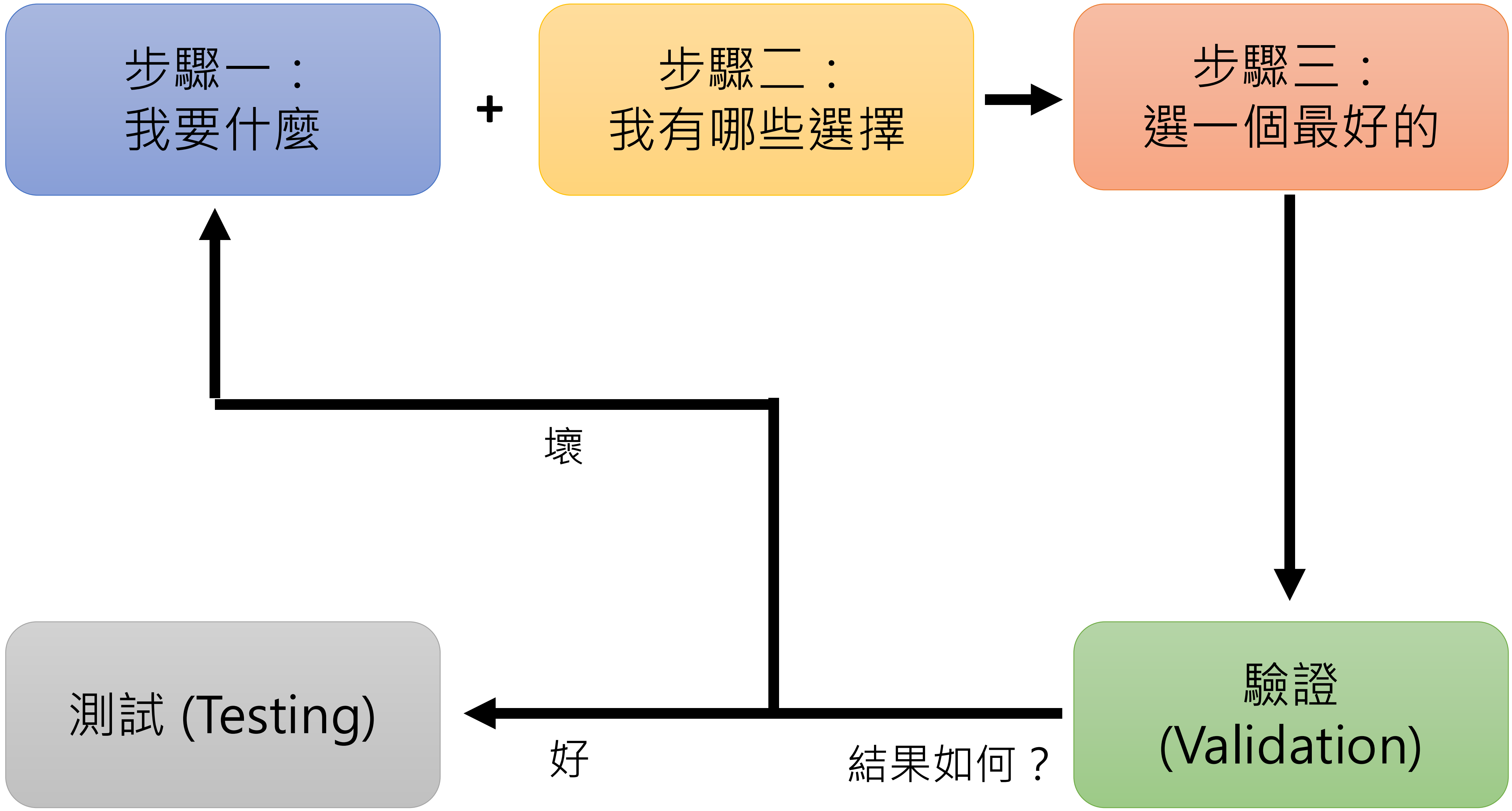
真的大考



Evaluation on the Validation Set

驗證
(Validation)

模擬考


步驟一：
我要什麼
+
步驟二：
我有哪些選擇
步驟三：
選一個最好的
驗證
(Validation)
結果如何？
壞
好
測試 (Testing)

步驟一：
我要什麼

+

步驟二：
我有哪些選擇

步驟三：
選一個最好的

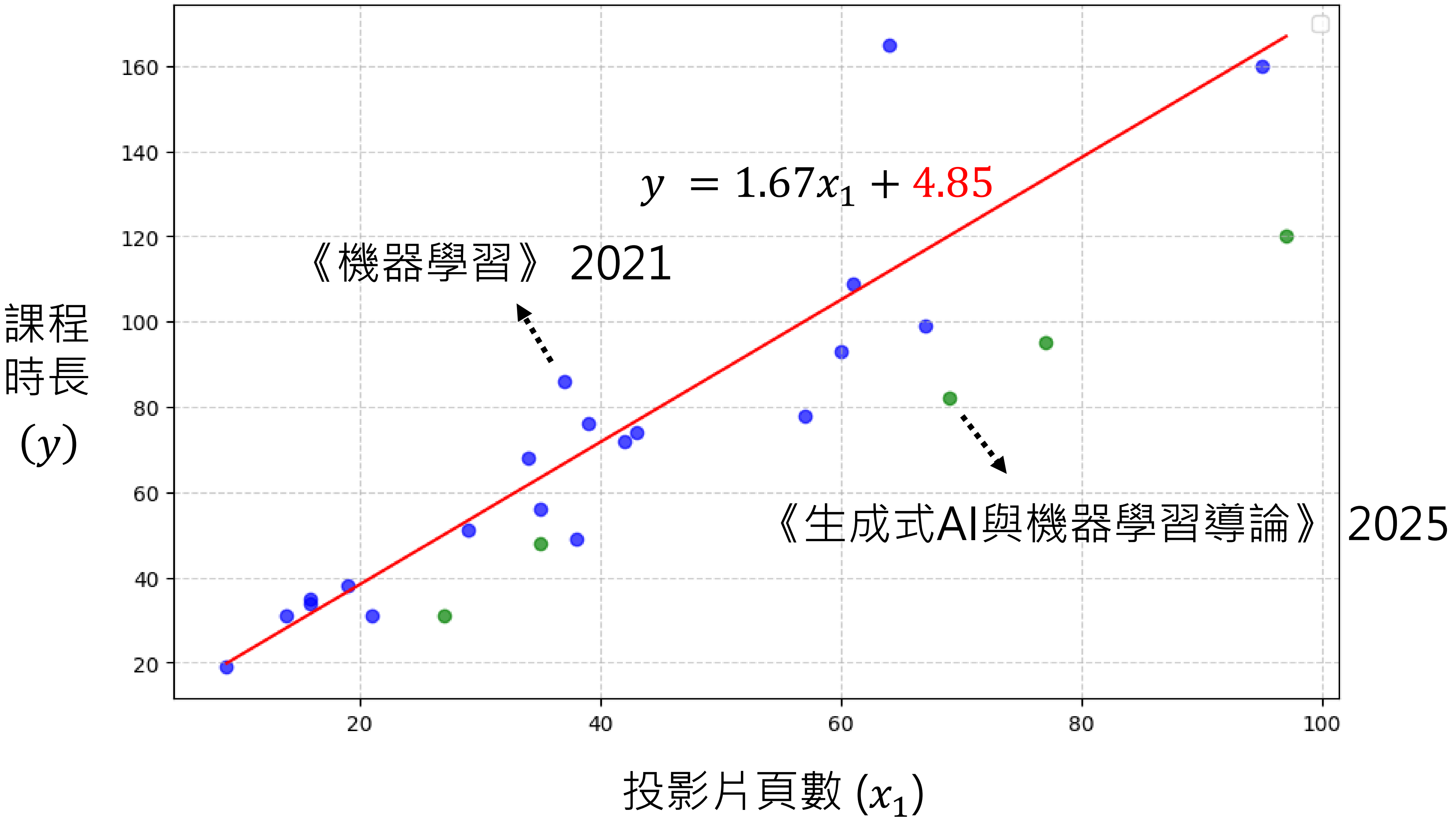
在《機器學習》2021 上計算 MSE

會不會有巨大差異？

在《生成式AI與機器學習導論》2025 上計算 MSE

驗證
(Validation)

你以為你要的目標，
跟實際上的目標不一致

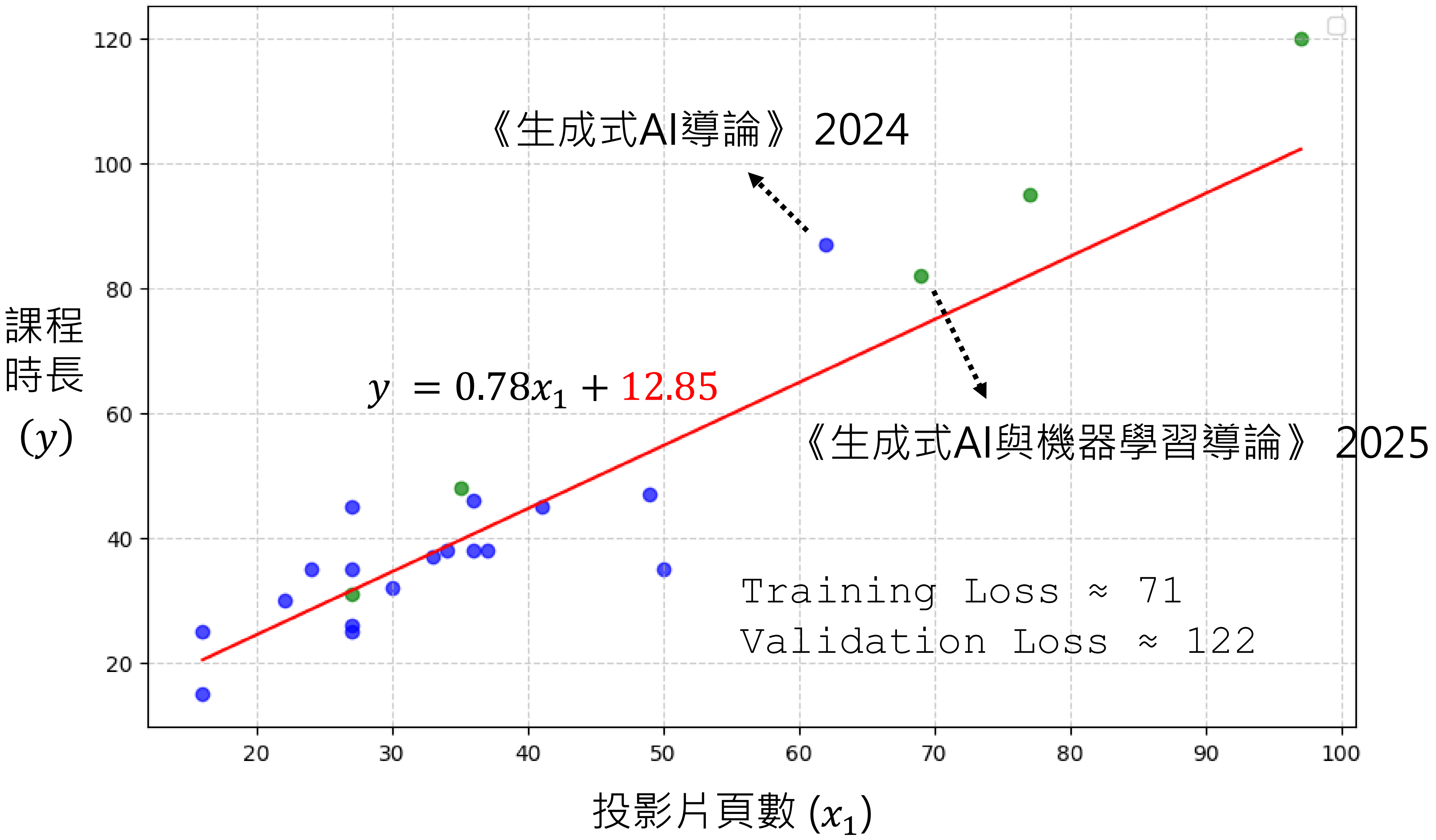


$y = 1.67x_1 + 4.85$
《機器學習》 2021
《生成式AI與機器學習導論》 2025
課程時長 (y)
投影片頁數 (x_1)
160
140
120
100
80
60
40
20
20
40
60
80
100

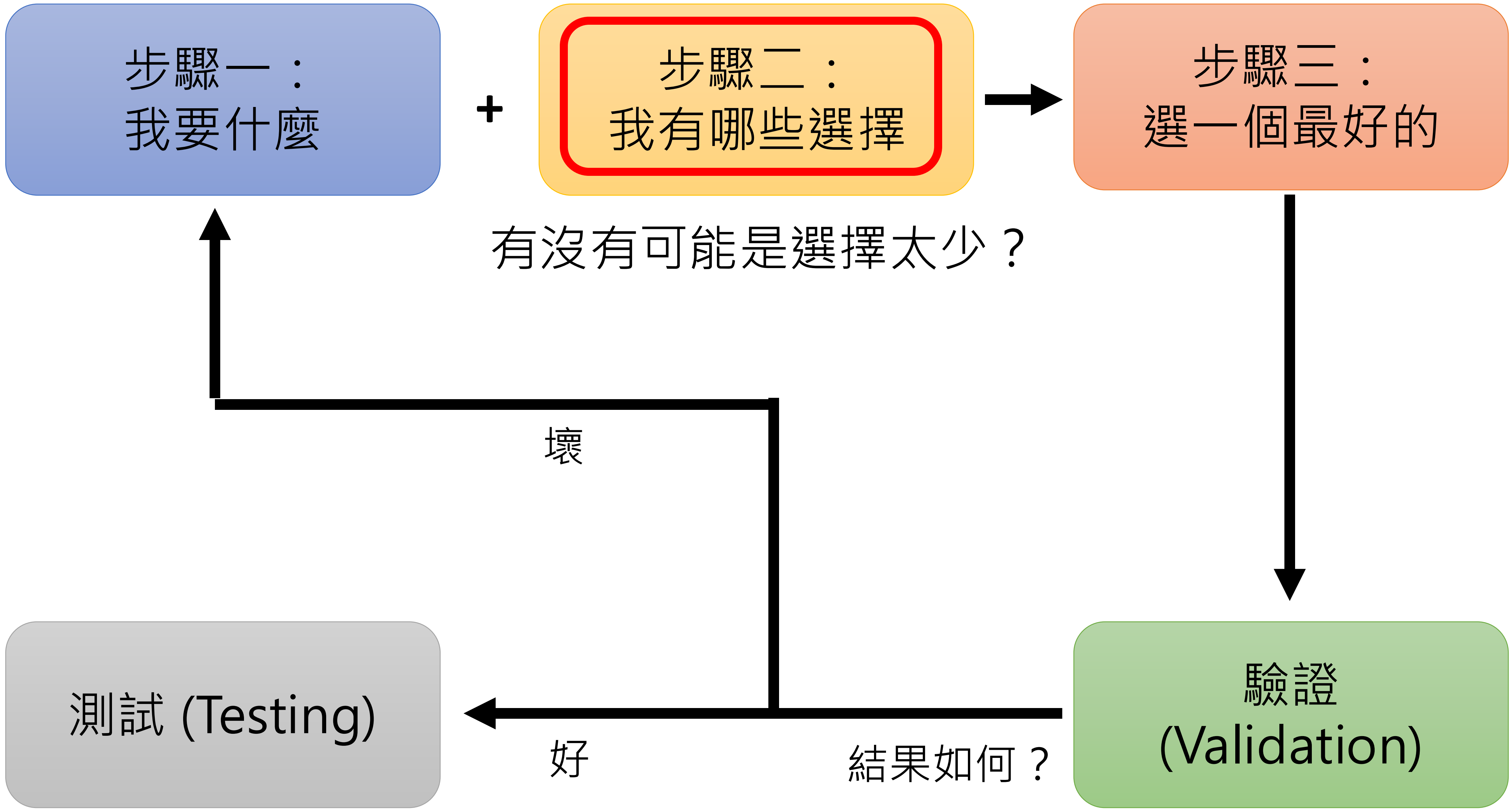
更換
訓練資料


【機器學習2021】(中文版)
Hung-yi Lee - 1/40
49:59
【機器學習2021】預測本頻道觀看人數 (上) - 機器學習基本...
Hung-yi Lee
2
58:35
【機器學習2021】預測本頻道觀看人數 (下) - 深度學習基本...
Hung-yi Lee
3
51:23
【機器學習2021】機器學習任務攻略
Hung-yi Lee
4
33:45
【機器學習2021】類神經網路訓練不起來怎麼辦 (一)： 局...
Hung-yi Lee
5
30:59
【機器學習2021】類神經網路訓練不起來怎麼辦 (二)： 批...
Hung-yi Lee
6
37:42
【機器學習2021】類神經網路訓練不起來怎麼辦 (三)：自動...
Hung-yi Lee
7
19:27
【機器學習2021】類神經網路訓練不起來怎麼辦 (四)：損失...


【生成式AI導論 2024】
Hung-yi Lee - 1/20
25:40
【生成式AI導論 2024】第0講：課程說明 (17:15 有芙莉...
Hung-yi Lee
2
29:29
【生成式AI導論 2024】第1講：生成式AI是什麼？
Hung-yi Lee
3
26:06
【生成式AI導論 2024】第2講：今日的生成式人工智慧...
Hung-yi Lee
4
34:35
【生成式AI導論 2024】第3講：訓練不了人工智慧？你...
Hung-yi Lee
5
47:22
【生成式AI導論 2024】第4講：訓練不了人工智慧？你...
Hung-yi Lee
6
25:20
【生成式AI導論 2024】第5講：訓練不了人工智慧？你...
Hung-yi Lee
7
34:26
【生成式AI導論 2024】第6講：大型語言模型修練史 — ...

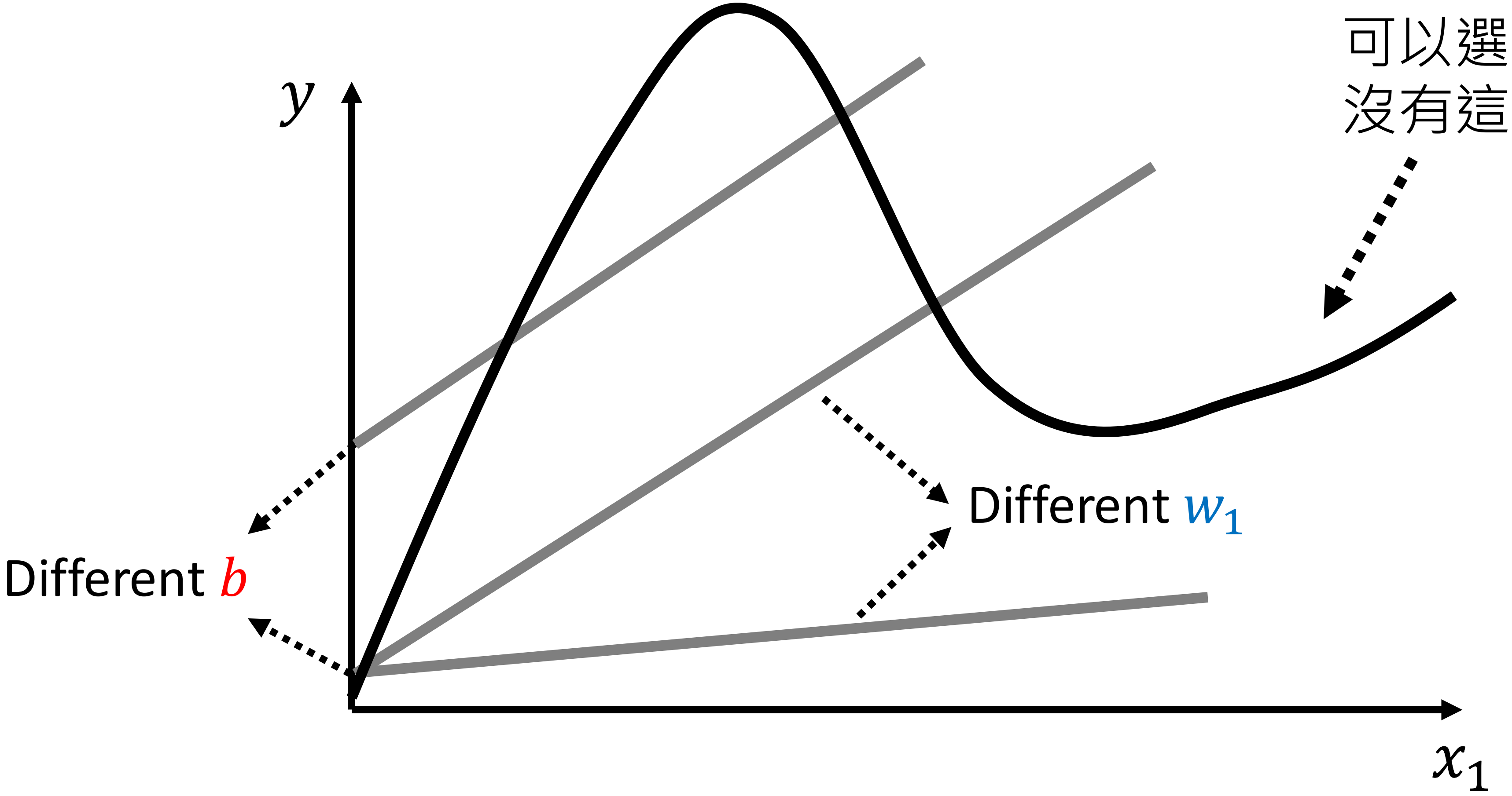


《生成式AI導論》2024
《生成式AI與機器器學習導論》2025
$y = 0.78x_1 + 12.85$
Training Loss ≈ 71
Validation Loss ≈ 122
課程時長 (y)
投影片頁數 (x_1)
120
100
80
60
40
20
20
30
40
50
60
70
80
90
100



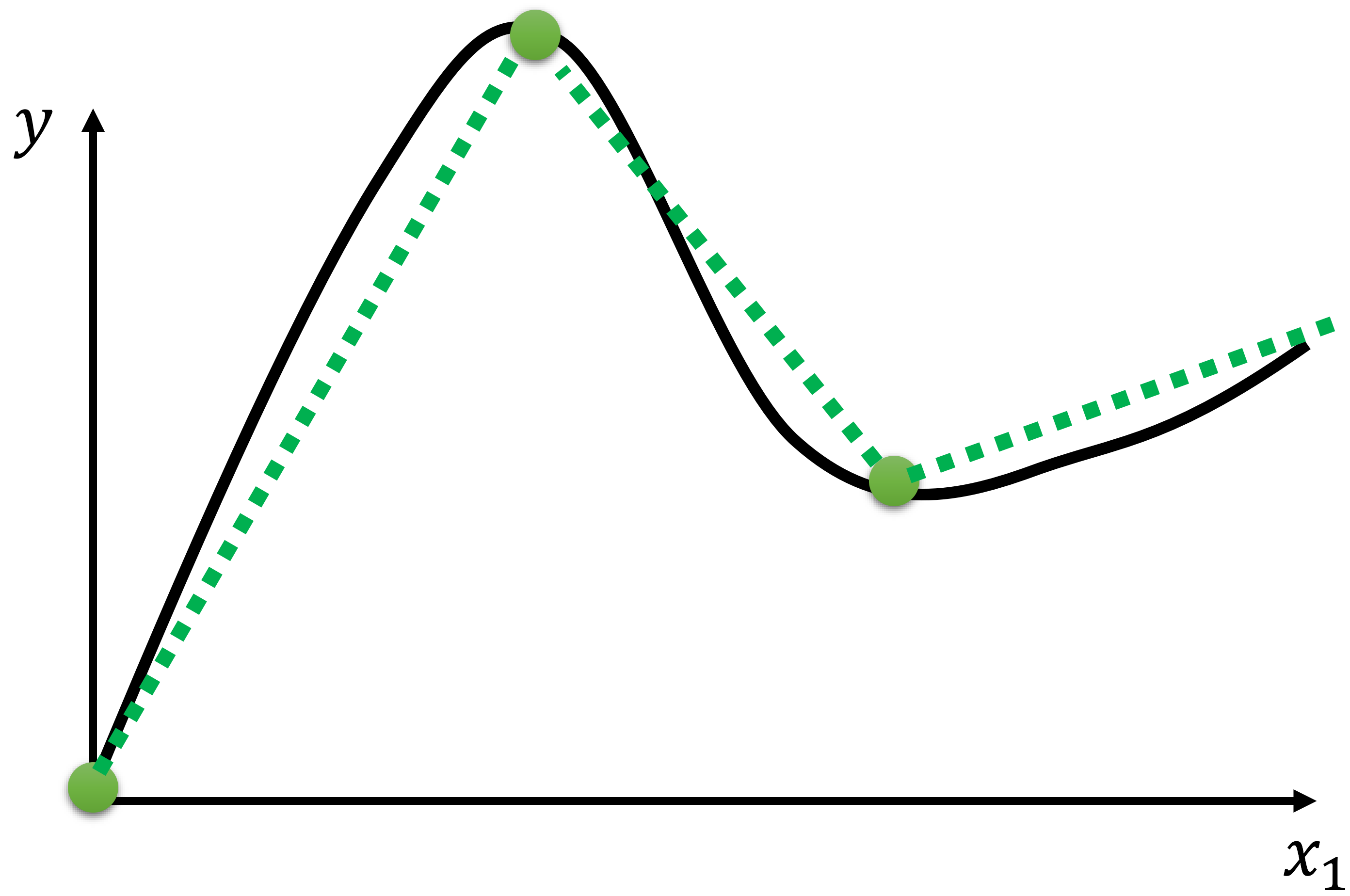
步驟一：
我要什麼
+
步驟二：
我有哪些選擇
步驟三：
選一個最好的
有沒有可能是選擇太少？
壞
驗證
(Validation)
結果如何？
好
測試 (Testing)

$$y = w_1 x_1 + b$$

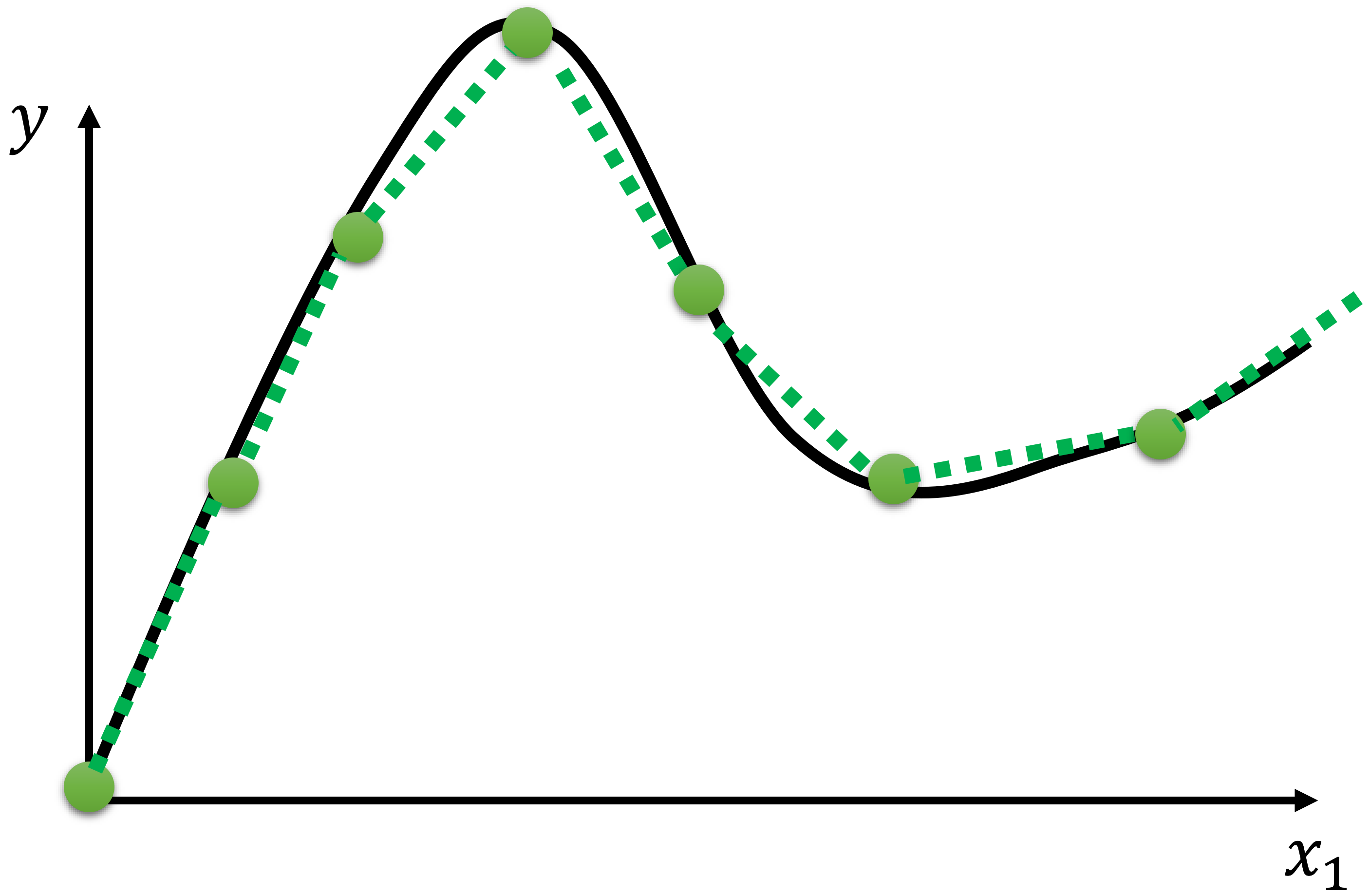


畫一個有機會包含所有函數的範圍

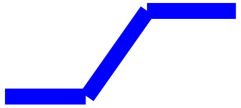
Piecewise Linear

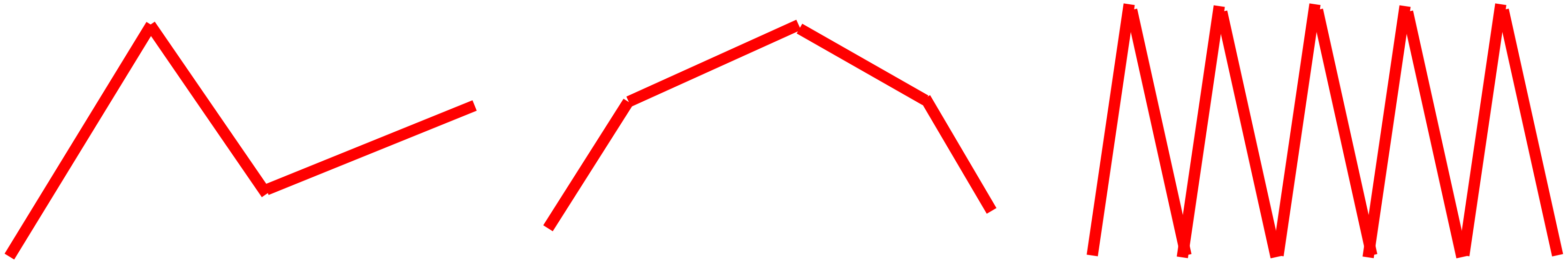


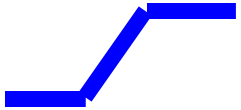
Piecewise Linear

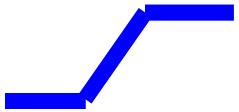


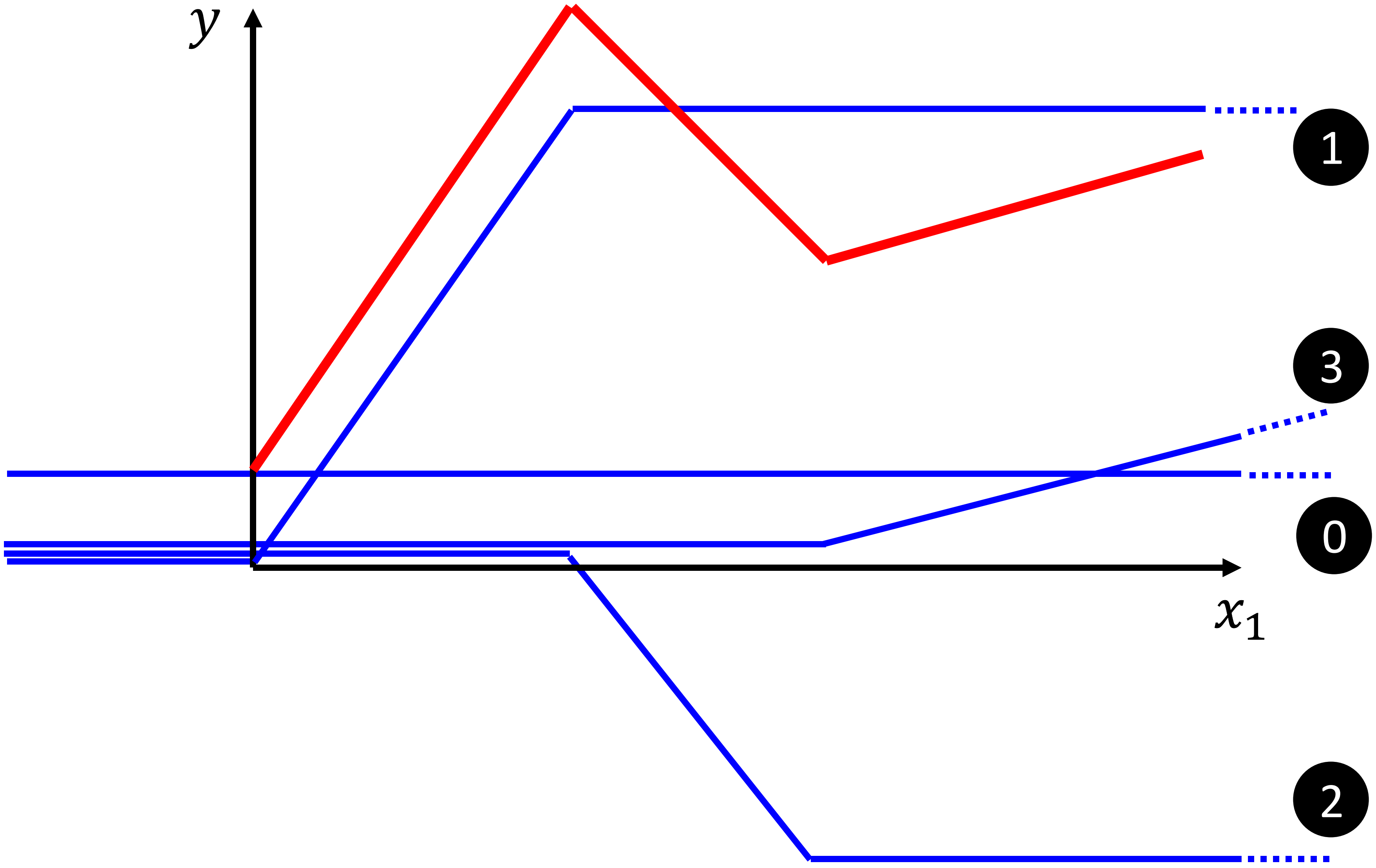
All Piecewise Linear Curves

= constant + sum of a set of 

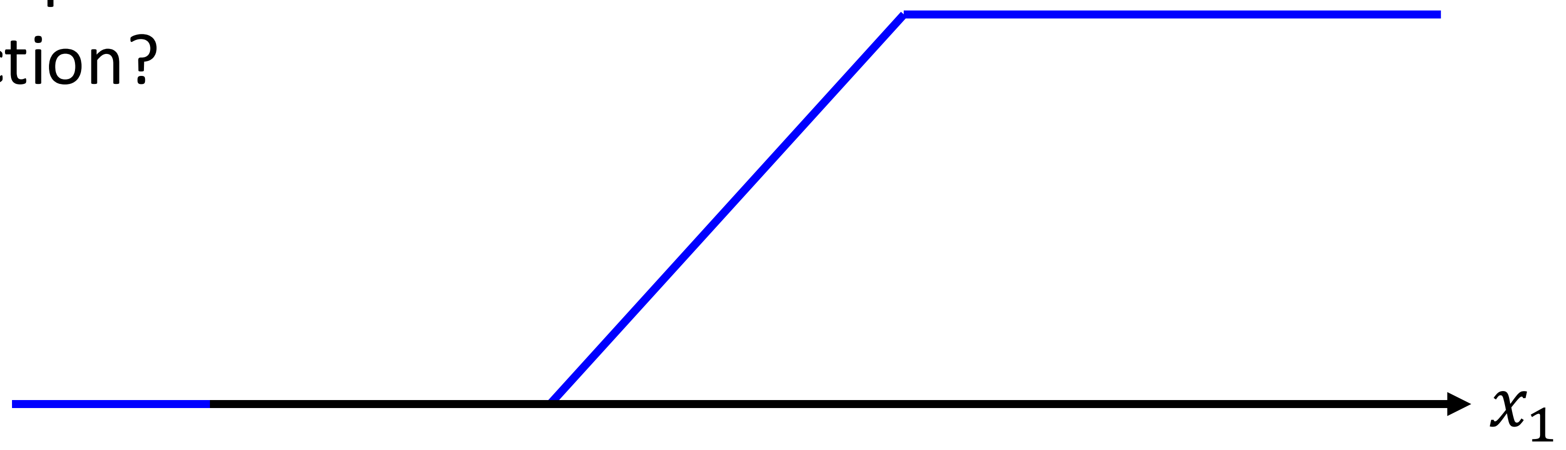


More pieces require more 

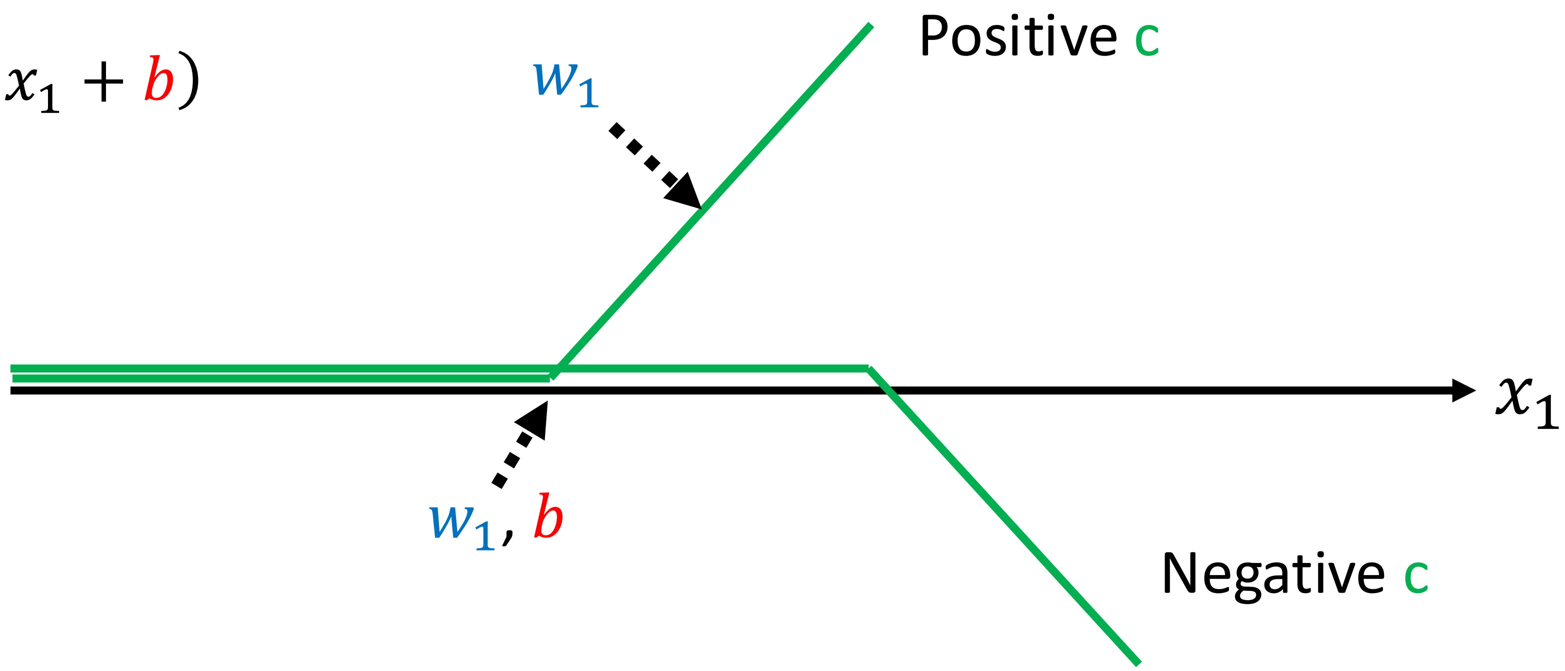
red curve = constant + sum of a set of 



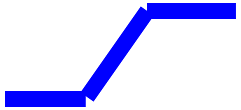
How to represent this function?



$c\,max(0, w_1x_1 + b)$



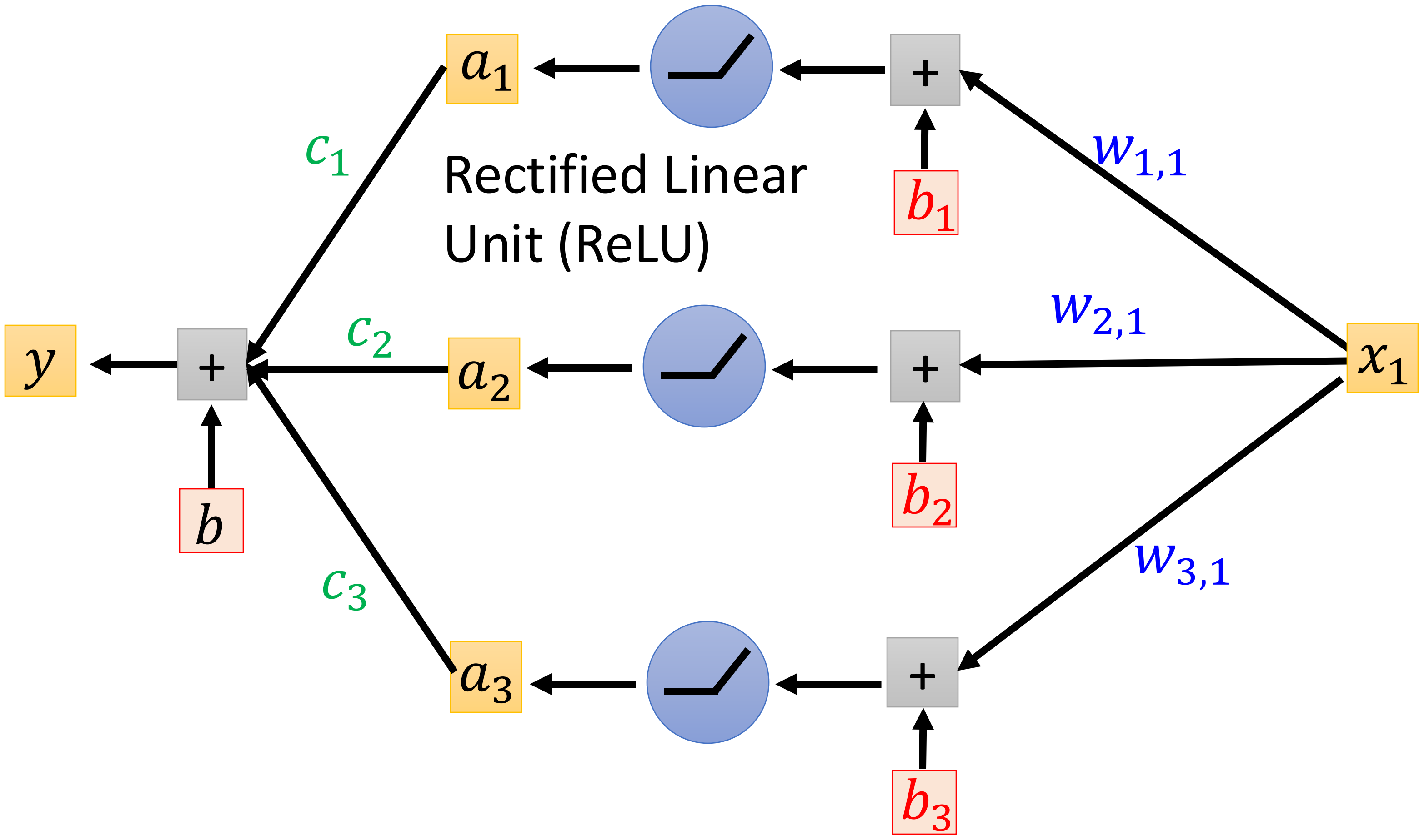
Any Curves $\approx$ Piecewise Linear Curves

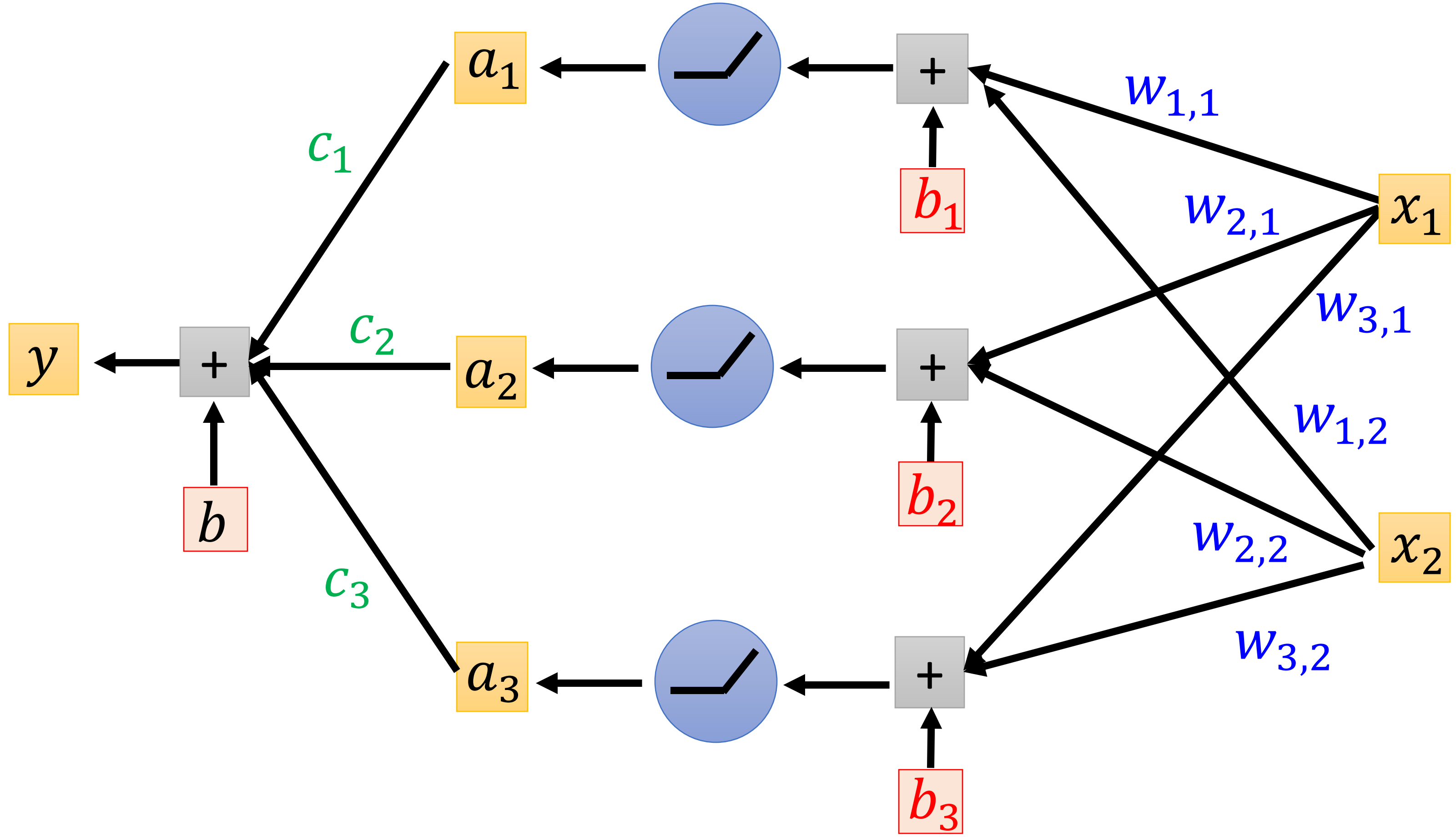
= constant + sum of a set of 

= constant + $\underbrace{\text{sum of a larger set of}}_{\mathrm{H}}$  $c\,max(0, w_1 x_1 + b)$

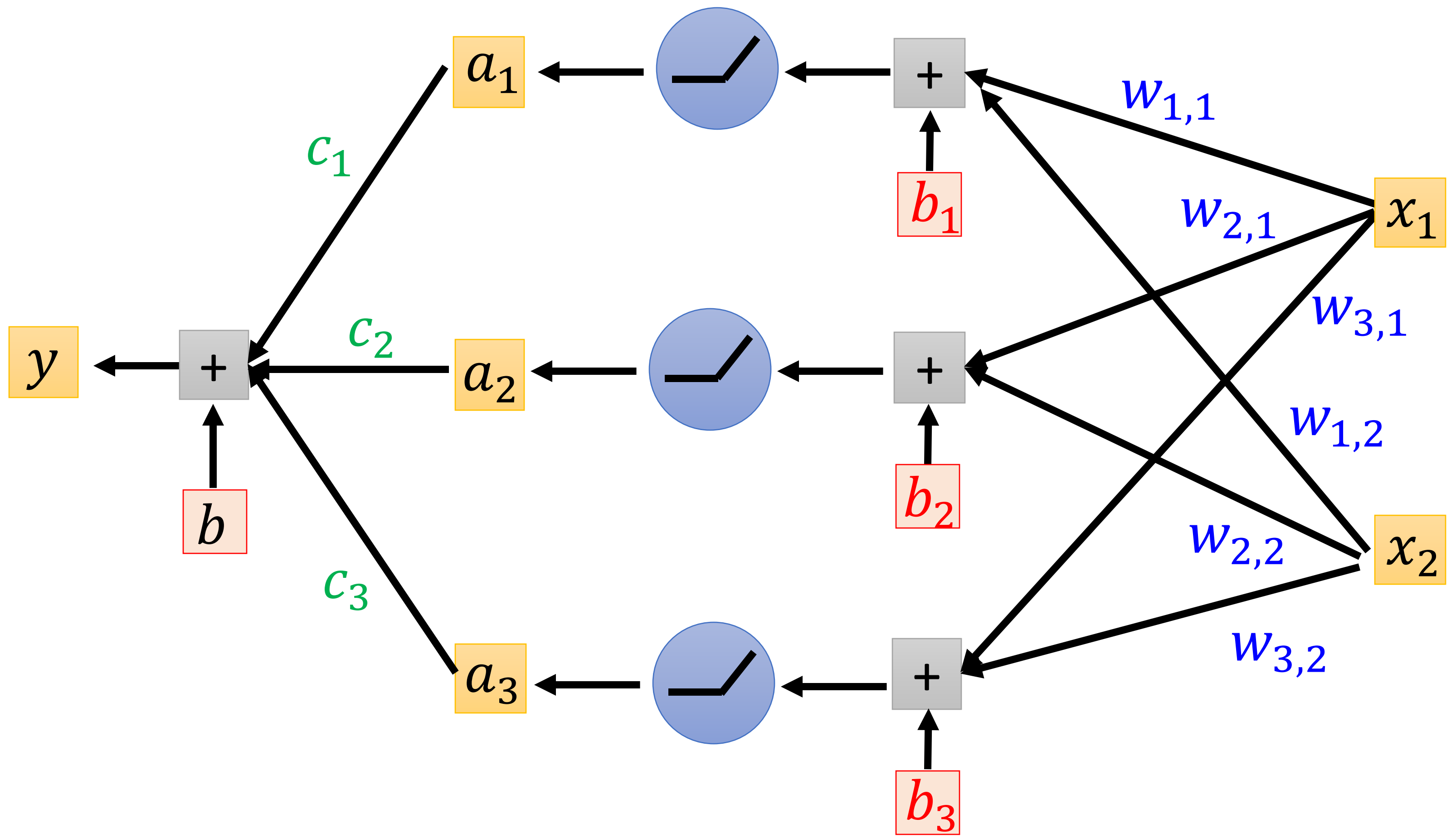
$$y = b + \sum_{i=1}^{H} c_i\, max\left(0, w_{i,1} x_1 + b_i\right)$$

$$y = b + \sum_{i=1}^{H} c_i \underbrace{max(0, w_{i,1} x_1 + b_i)}_{a_i}$$





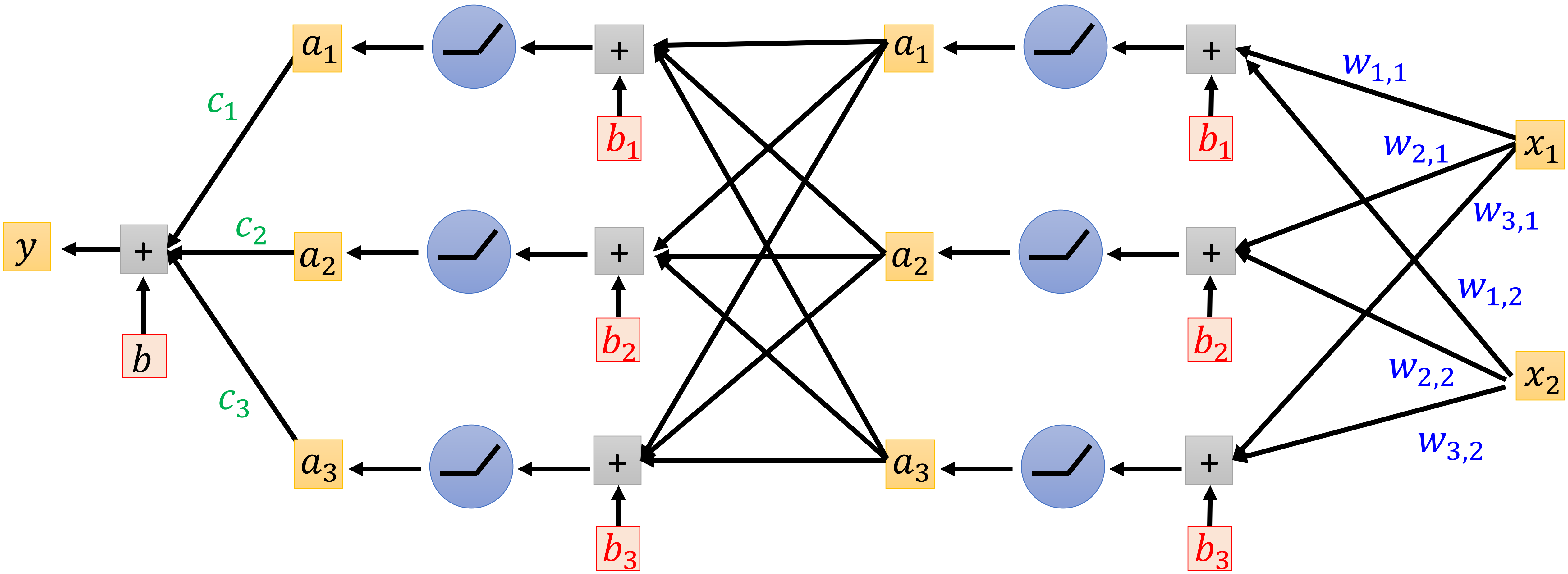
$$y = b + \begin{bmatrix} c_1 & c_2 & c_3 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \qquad \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \sigma(\qquad\qquad \begin{bmatrix} x_1 \\ x_2 \end{bmatrix})$$

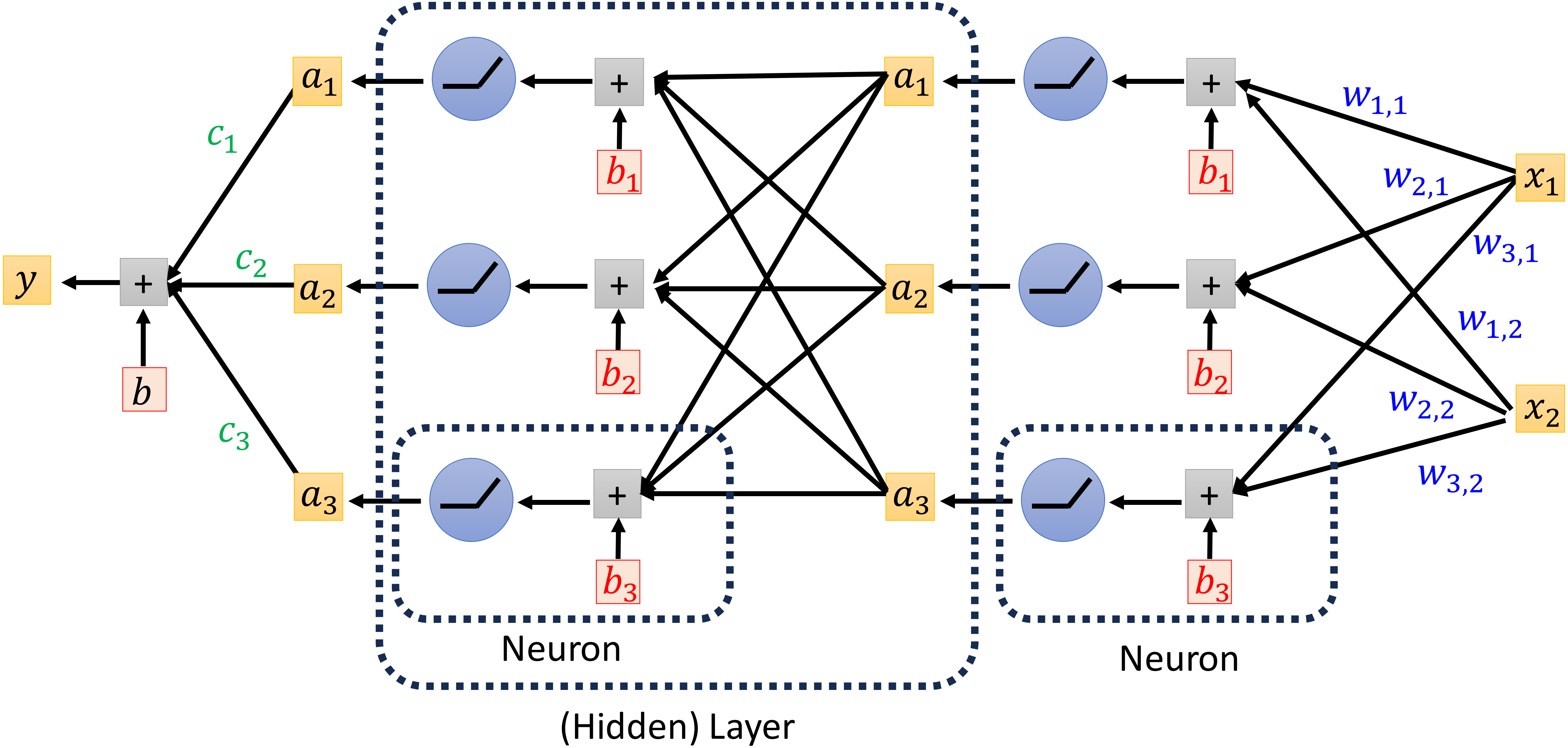


$$y = b + \boldsymbol{c}^T \boldsymbol{a}$$

$$\boldsymbol{a} = \sigma(\boldsymbol{b} + W \boldsymbol{x})$$

$$\boldsymbol{a}' = \sigma(\boldsymbol{b}' + W'\boldsymbol{a}) \qquad \boldsymbol{a} = \sigma(\boldsymbol{b} + W\boldsymbol{x})$$





Neural Network

Many Layers ⇨ **深度學習 (Deep Learning)**

Backpropagation

Computing gradients in an efficient way

https://youtu.be/ibJpTrp5mcE

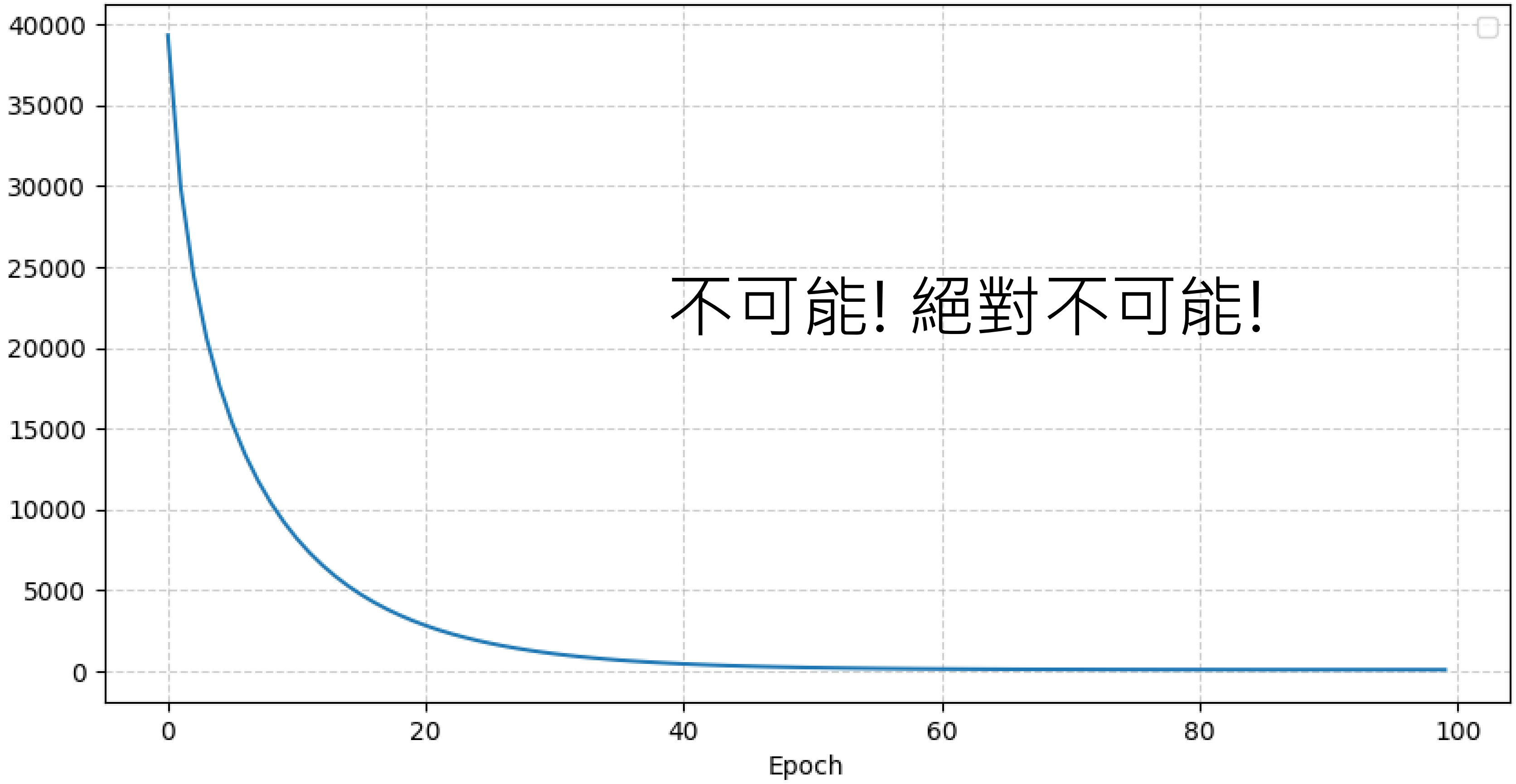
$$y = w_1 x_1 + b \longrightarrow y = b + \sum_{i=1}^{H} c_i \, max(0, w_{i,1} x_1 + b_i)$$

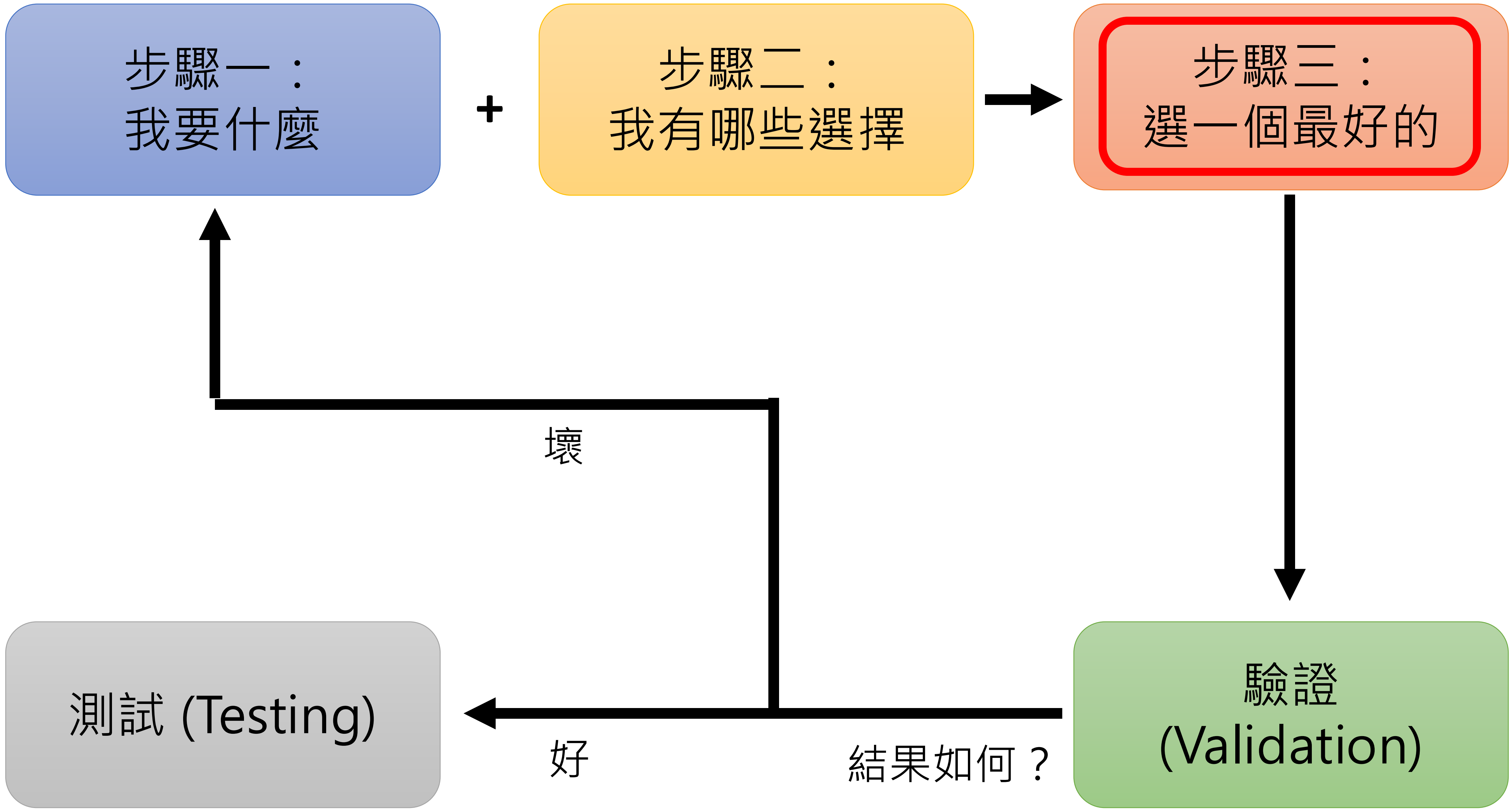
$H = 100$

Training Loss ≈ 71

Training Loss ≈ 80

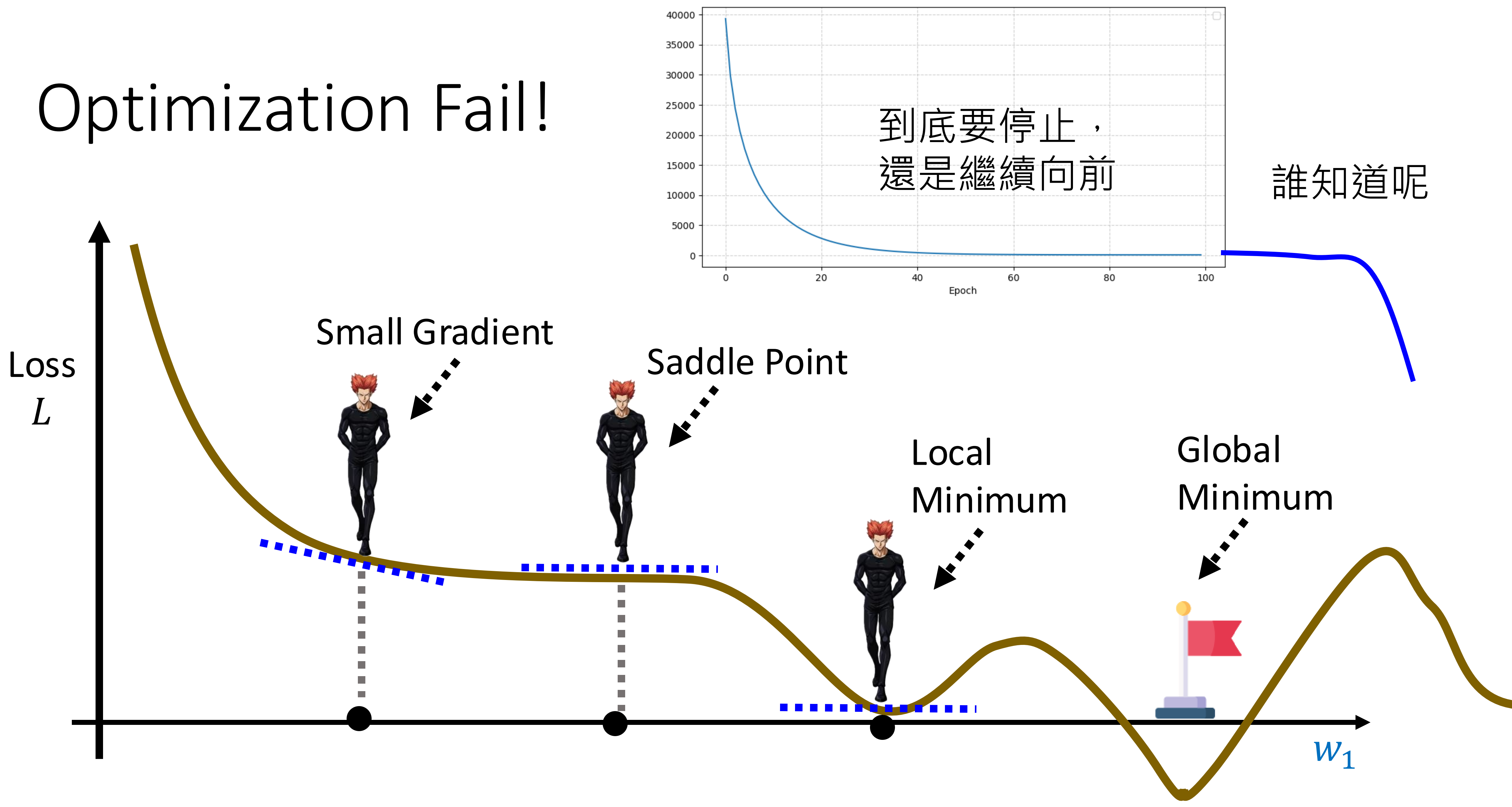
參數太多了，
只能看
Loss Curve

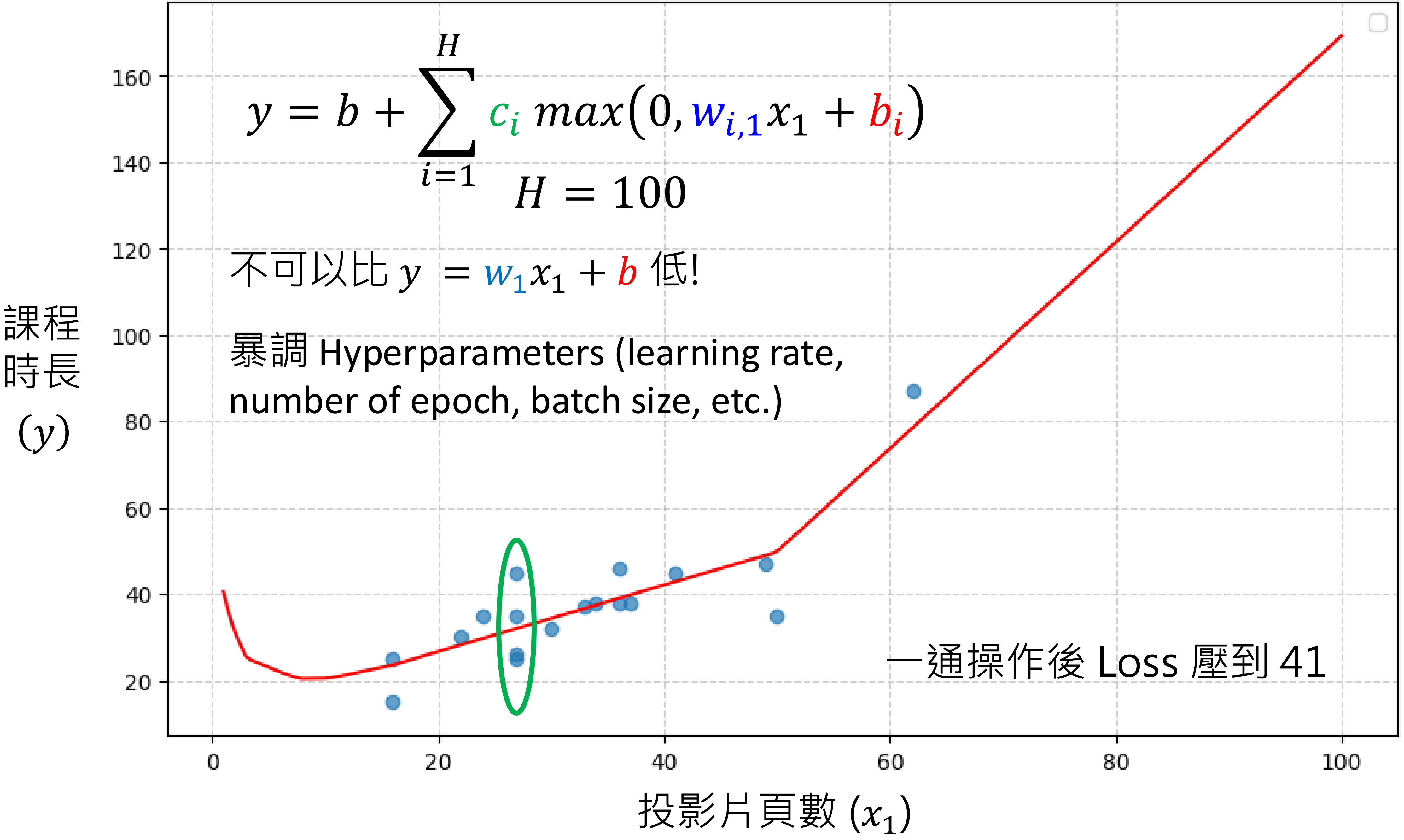




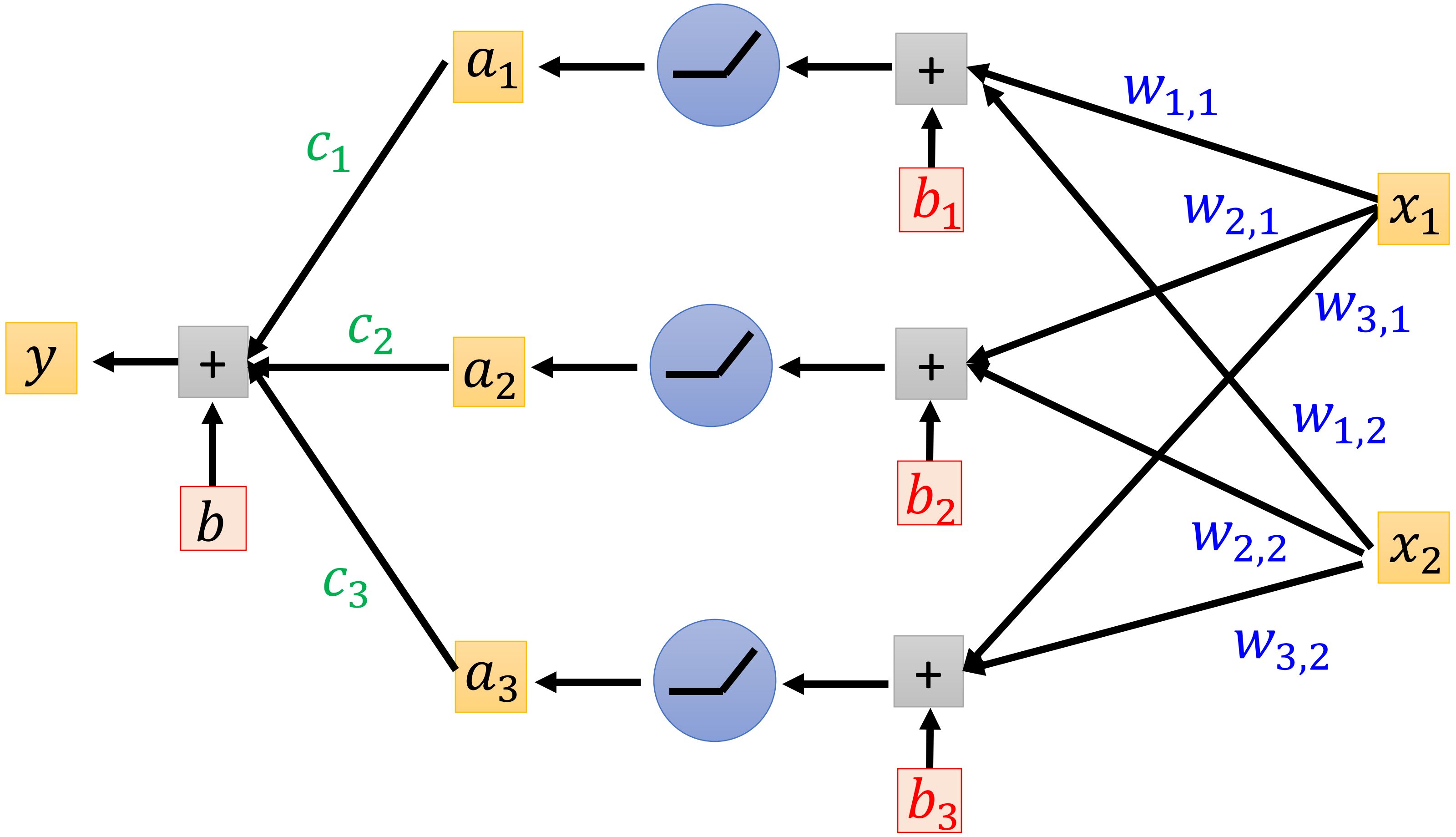
步驟一：
我要什麼
+
步驟二：
我有哪些選擇
步驟三：
選一個最好的
驗證
(Validation)
結果如何？
壞
好
測試 (Testing)

Optimization Fail!





$y = b + \sum_{i=1}^{H} c_i \, max\left(0, w_{i,1} x_1 + b_i\right)$
$H = 100$
不可以比 $y = w_1 x_1 + b$ 低!
暴調 Hyperparameters (learning rate, number of epoch, batch size, etc.)
一通操作後 Loss 壓到 41
課程時長 (y)
投影片頁數 (x_1)
160
140
120
100
80
60
40
20
0
20
40
60
80
100

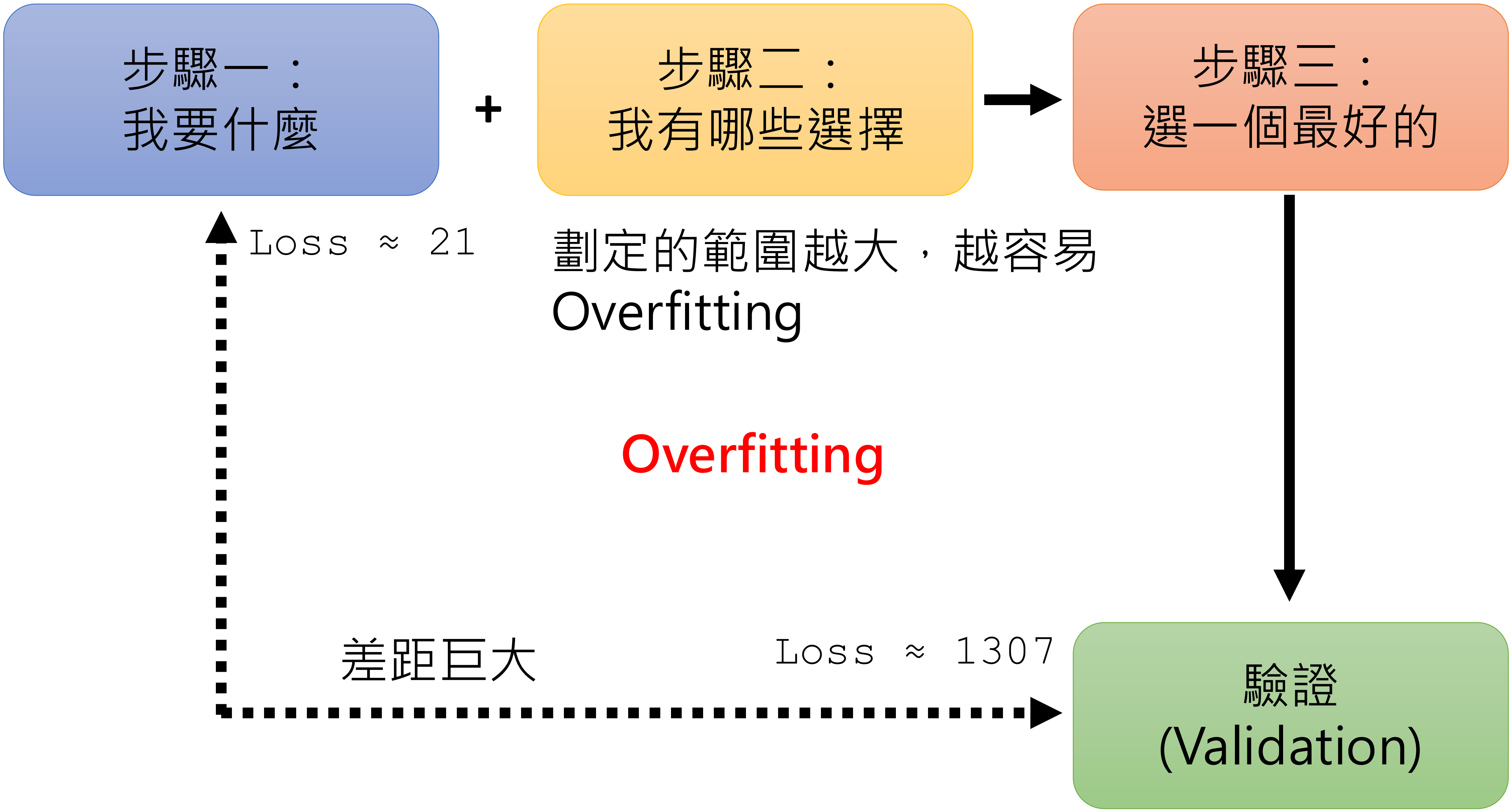


x_1: 有多少頁投影片

x_2: ~~投影片中總共有多少字~~

每頁投影片平均多少字

	Linear	Deep	+ No. of Word	+ Avg of Word
Training:	Loss ≈ 71	41	40	22
Validation:	Loss ≈ 122			1307


步驟一：
我要什麼
+
步驟二：
我有哪些選擇
步驟三：
選一個最好的
Loss ≈ 21
劃定的範圍越大，越容易
Overfitting
Overfitting
差距巨大
Loss ≈ 1307
驗證
(Validation)

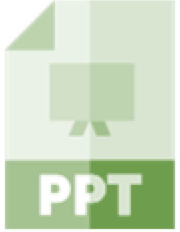
如果世上一切函數都可以選，會怎樣？

訓練資料

Duration: 10

Duration: 20

Duration: 30

$f_{lazy}($ 

$) = 10$

$f_{lazy}($ 

$) = 20$

$f_{lazy}($ 

$) = 30$

$f_{lazy}($ **Other** $) = 0$

在訓練資料上的 Loss 為 0

你說這是不是在訓練資料上Loss 最低的函式？

$f_{lazy}(\quad) = 10$

$f_{lazy}(\quad) = 20$

$f_{lazy}(\quad) = 30$

$f_{lazy}($ **Other** $) = 0$

驗證資料

Duration: 15

Duration: 32

Duration: 33

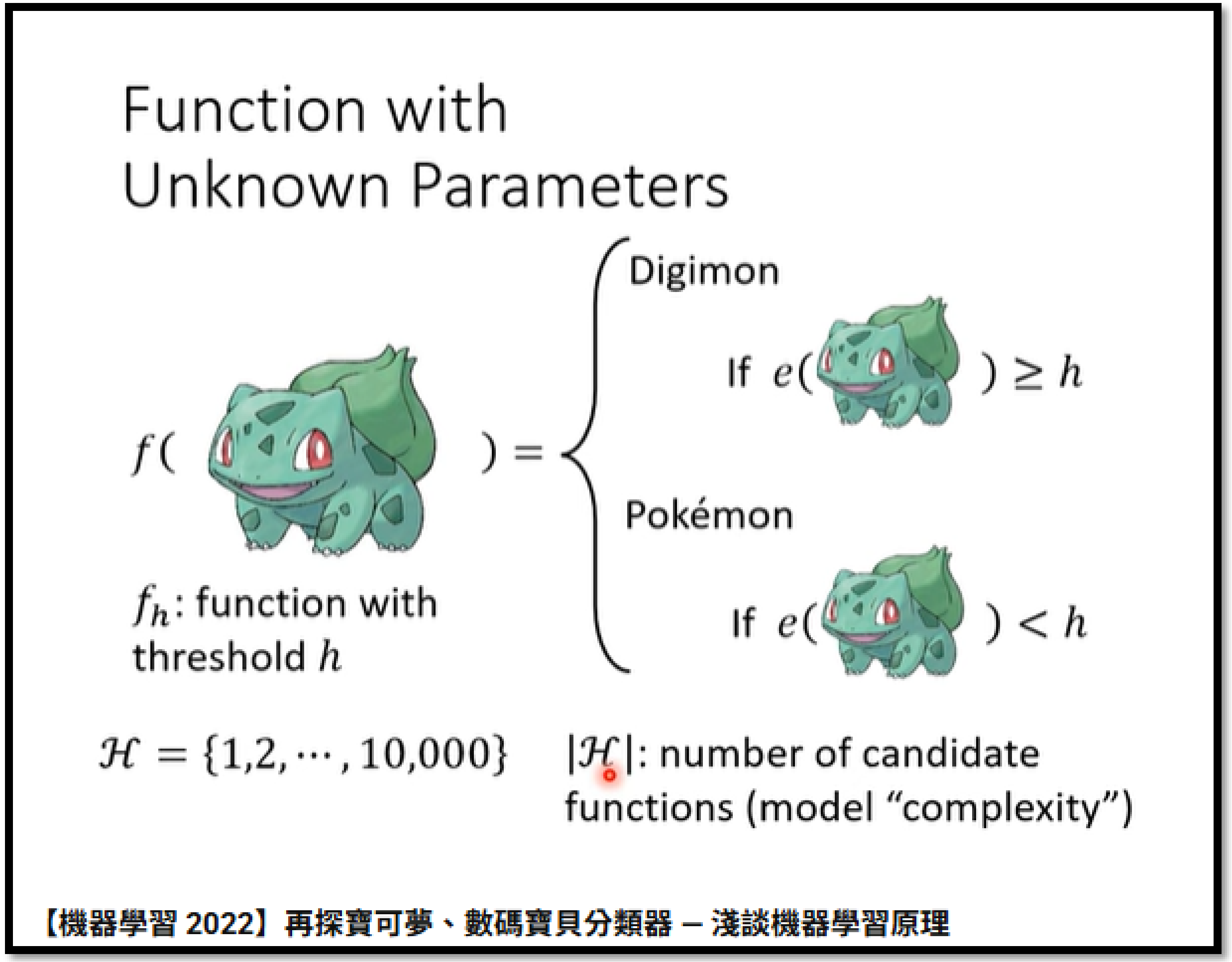
$f_{lazy}(\quad) = 0$

$f_{lazy}(\quad) = 0$

$f_{lazy}(\quad) = 0$

在訓練資料上的
Loss 為 0

在驗證資料上的
Loss 為炸裂



https://youtu.be/_j9MVVcvyZI?si=cKWY8QmyS3-wX4l9

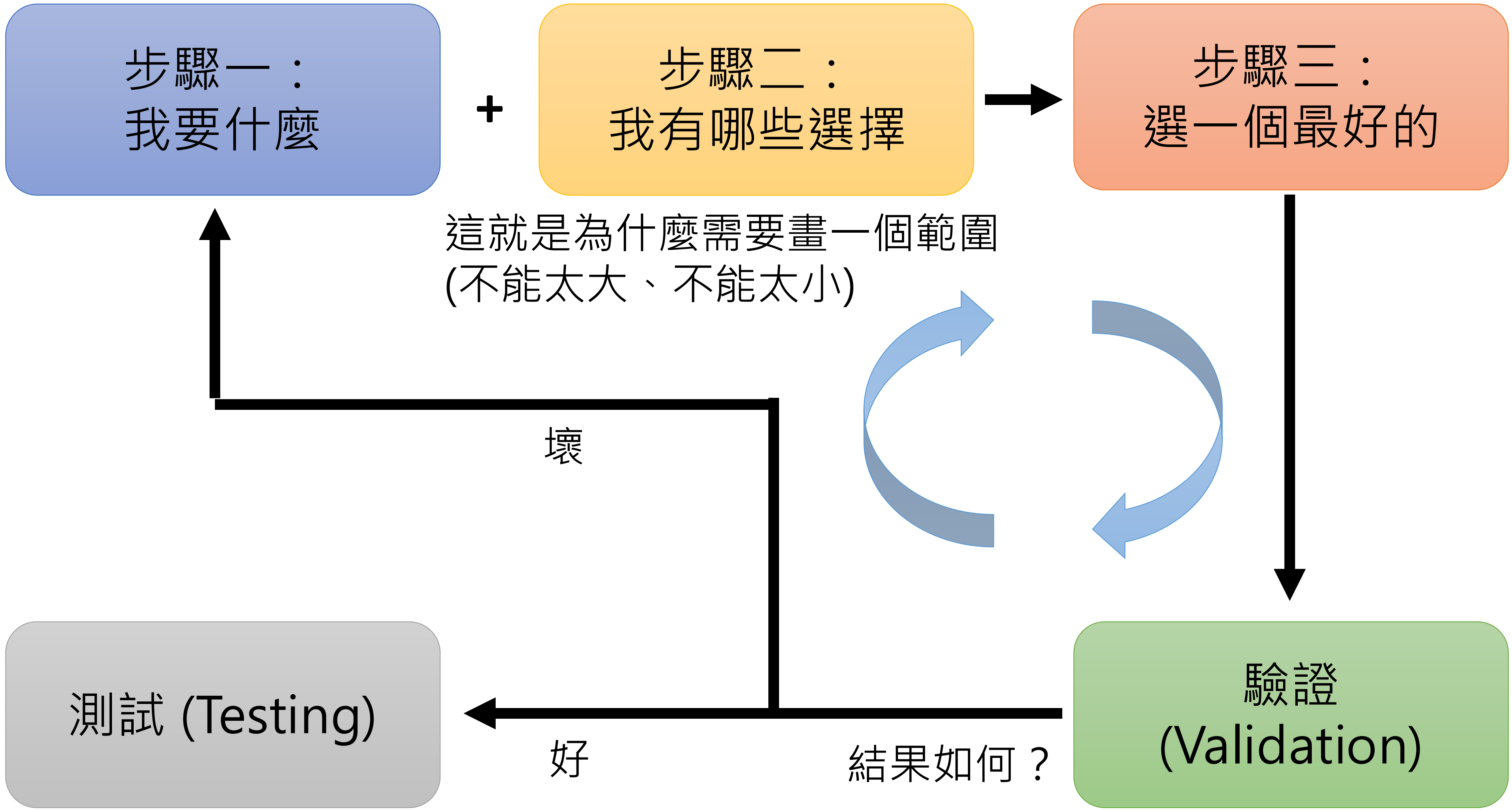
Overfitting

- 選擇越多，訓練和驗證的差距越大

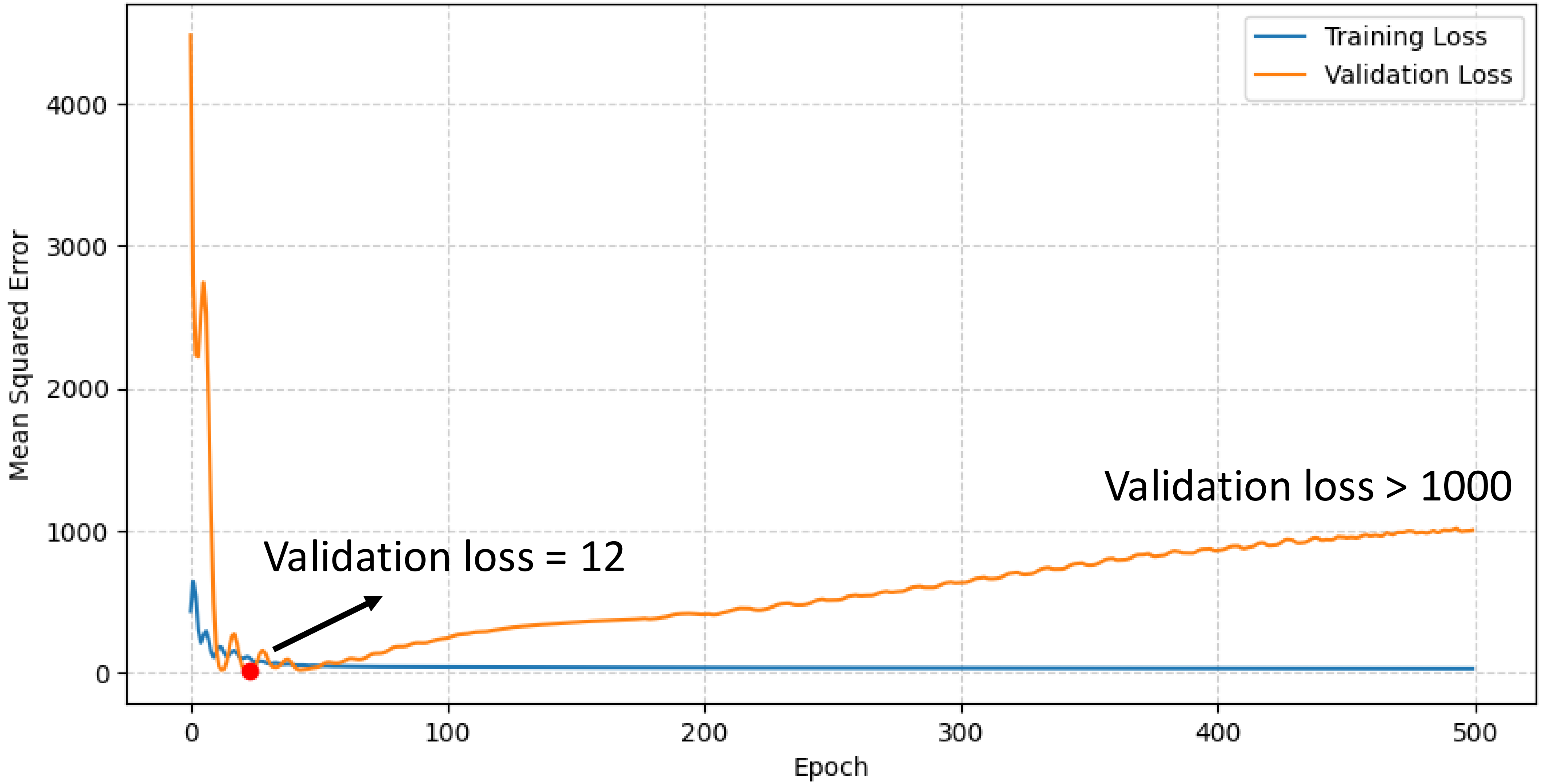
看著路開車

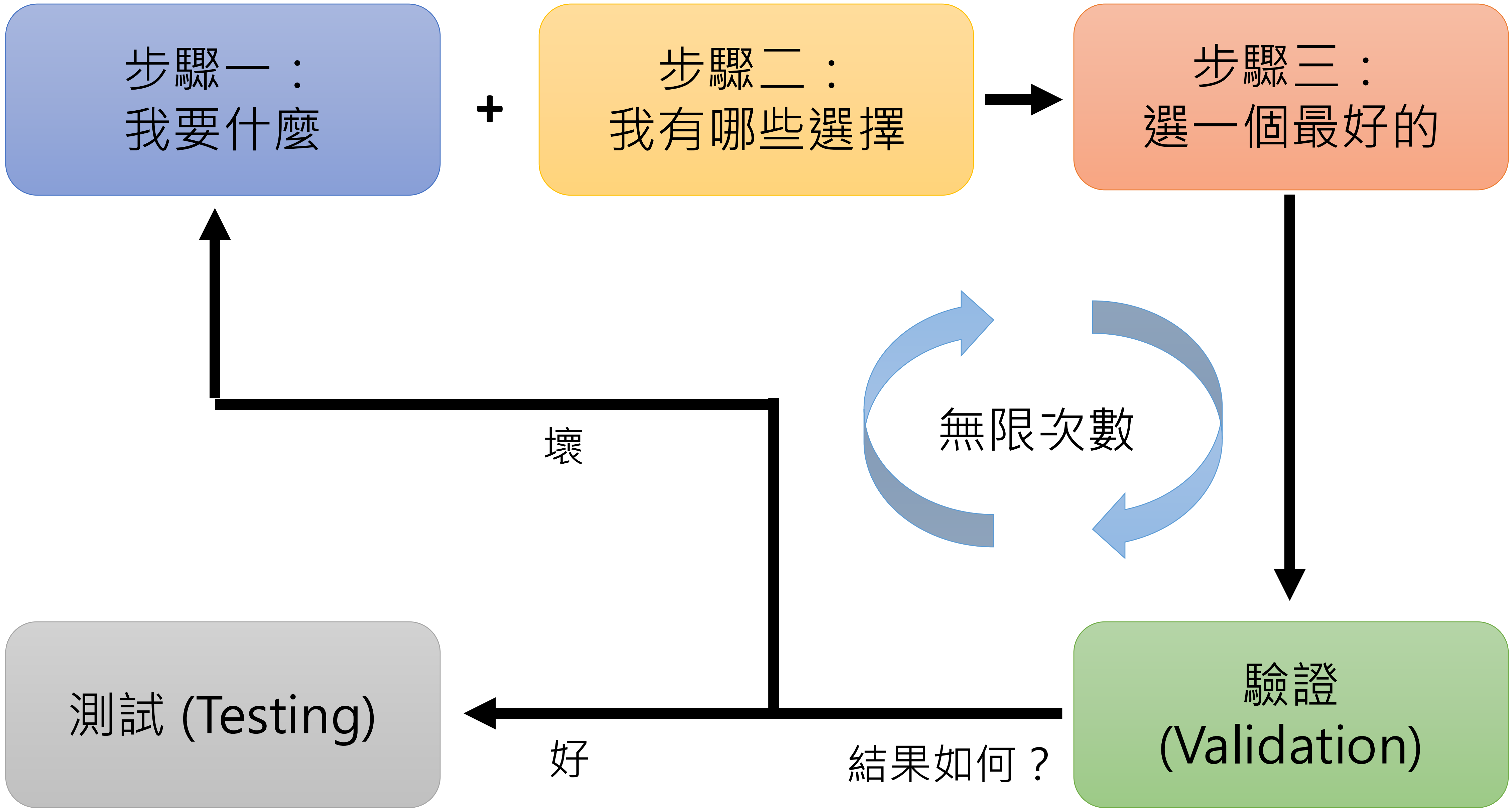
看著貼紙開車

只在駕訓班才能開車


步驟一：
我要什麼
+
步驟二：
我有哪些選擇
步驟三：
選一個最好的
這就是為什麼需要畫一個範圍
(不能太大、不能太小)
壞
驗證
(Validation)
結果如何？
好
測試 (Testing)

每一個 Epoch 結束都去量 Validation Loss




步驟一：
我要什麼
+
步驟二：
我有哪些選擇
步驟三：
選一個最好的
無限次數
壞
驗證
(Validation)
結果如何？
好
測試 (Testing)

如果可以無限的使用驗證資料

訓練資料

$f_{lazy2}(\ ) = 10$

$f_{lazy2}(\ ) = 20$

$f_{lazy2}(\ ) = 30$

跟訓練資料一樣

驗證資料

$f_{lazy2}(\ ) = 15$

$f_{lazy2}(\ ) = 32$

$f_{lazy2}(\ ) = 33$

跟驗證資料一樣

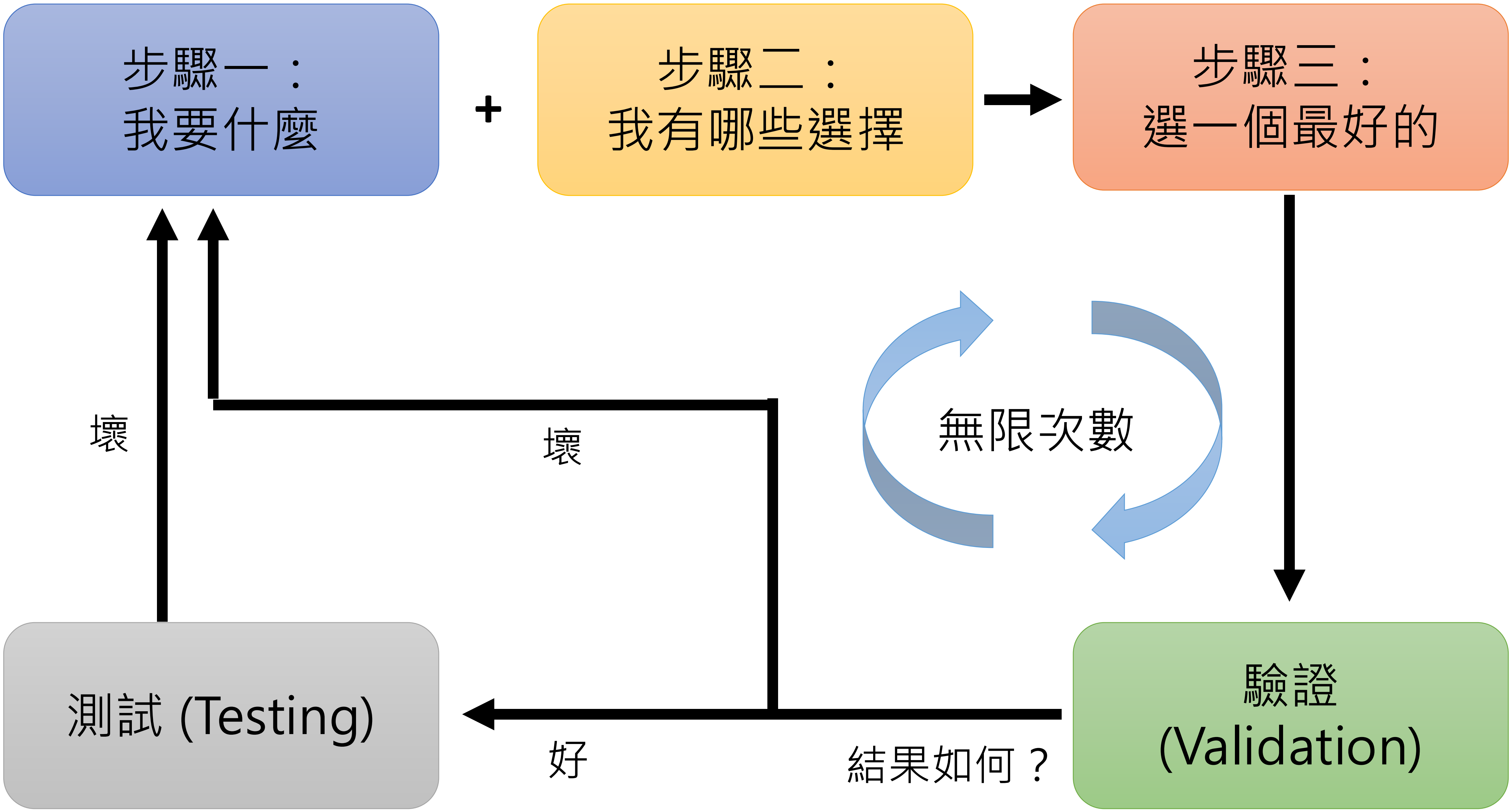
測試資料

$f_{lazy2}(\ ) = 0$

$f_{lazy2}(\ ) = 0$

$f_{lazy2}(\ ) = 0$

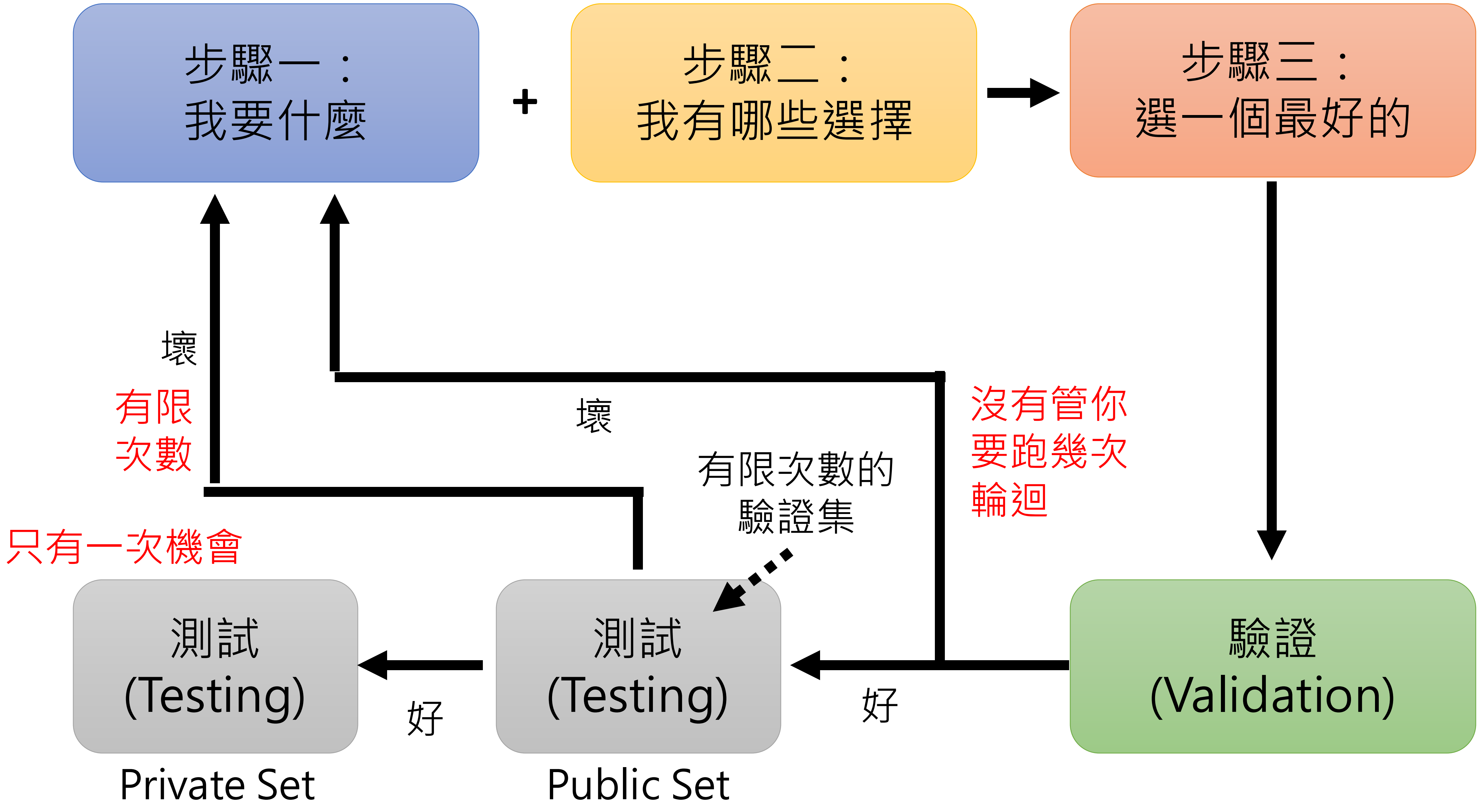
亂給答案



假設可以做無限制的測試 ....

Rank	Model	EM	F1
	Human Performance *Stanford University* (Rajpurkar & Jia et al. '18)	86.831	89.452
1 Mar 20, 2019	BERT + DAE + AoA (ensemble) *Joint Laboratory of HIT and iFLYTEK Research*	**87.147**	**89.474**
2 Mar 15, 2019	BERT + ConvLSTM + MTL + Verifier (ensemble) *Layer 6 AI*	86.730	89.286
3 Mar 05, 2019	BERT + N-Gram Masking + Synthetic Self-Training (ensemble) *Google AI Language* https://github.com/google-research/bert	86.673	89.147
4 May 21, 2019	XLNet (single model) *XLNet Team*	86.346	89.133
5 Apr 13, 2019	SemBERT(ensemble) *Shanghai Jiao Tong University*	86.166	88.886

這就是為什麼人工智慧常常在 Benchmark 上打敗人類 ......


步驟一：
我要什麼
+
步驟二：
我有哪些選擇
步驟三：
選一個最好的
壞
有限
次數
壞
沒有管你
要跑幾次
輪迴
有限次數的
驗證集
只有一次機會
測試
(Testing)
Private Set
好
測試
(Testing)
Public Set
好
驗證
(Validation)

課程規劃

原理

實作

範例程式

連結：
https://colab.research.google.com/drive/1SFtkeDL9jp5LtaVsj-2JApsGpOltILi9?usp=sharing