

第2章 生成式AI相關理論

和AI做朋友補充教材

顏永進

Contents

第2章 生成式AI相關理論

- 2.1 AI 的本質：函數
- 2.2 模型訓練
- 2.3 生成式AI：一種會接龍的函數
- 2.4 文字、圖形與向量：先理解才能生成
- 2.5 編碼與解碼：邁向生成的一小步

2.1 AI 的本質：函數

第2章 生成式AI相關理論

例：利用人工智慧來預估自己這個學期的資訊科成績

- 假設老師主要依期中、期末考成績來決定學期成績，我們其實是在找一個類似這樣的函數

$$f(\text{期中考成績}, \text{期末考成績}) \rightarrow \text{學期總成績}$$

- 這樣的函數 有無限多個，例

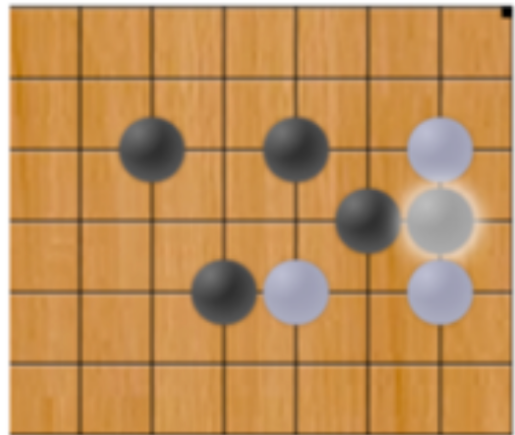
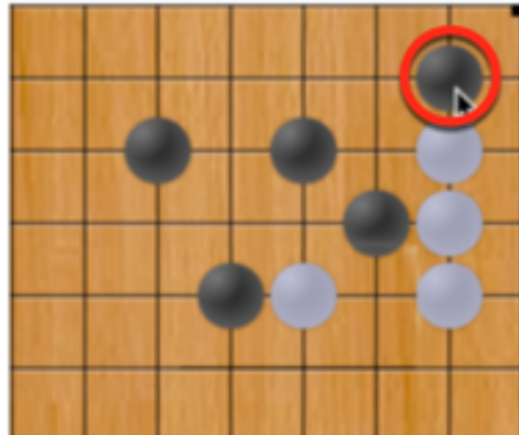
$$\text{學期總成績} = f(\text{期中考成績}, \text{期末考成績}) = 0.4 \times \text{期中考成績} + 0.6 \times \text{期末考成績}$$

此處的0.4、0.6就稱為這個函數f的參數（parameter）

- 此函數f便是用來預估分數的模型（model）

AI模型 v.s. 函數

2.1 AI 的本質：函數

- 圖形辨識的函數 $f(\text{) \rightarrow$
狗：0.82
貓：0.18
- 一個能做語音識別的函數 $f(\text{) \rightarrow$ 生成式 AI
- 一個會下五子棋的函數 $f(\text{) \rightarrow$ 

AI模型 v.s. 函數

2.1 AI 的本質：函數

- 文字生成的函數 $f(\text{什麼是台灣最美的風景?}) \rightarrow \text{台灣最美的風景是人}$

- 影像生成的函數 $f(\text{一隻飛向月球的貓}) \rightarrow$



- 這些的函數是怎麼找出來的？

2.2 模型訓練

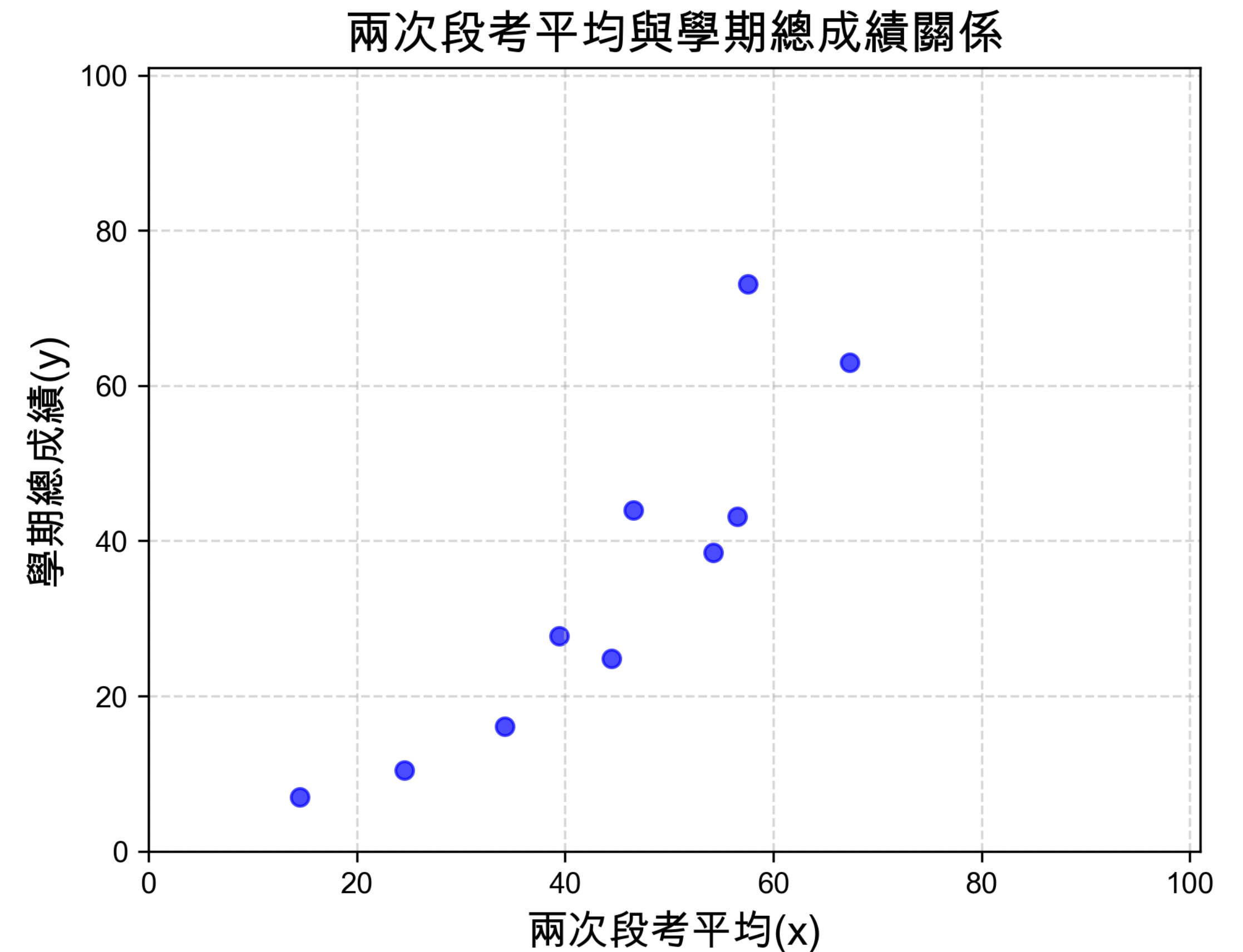
第2章 生成式AI相關理論

- 找一個函數其實也就是找到該函數的參數
- 假設老師的計分方式其實就是把兩次段考成績乘上某一個數就當成學期成績
- $y = f(x) = ax$ (x為兩次段考平均成績,y為學期總成績)
- 為了找出這個函數，我們首先應蒐集歷屆學長姐的成績資料（包括兩次段考平均成績與學期總成績），這些資料稱之為訓練資料（training data）

1. 收集訓練資料

2.2 模型訓練

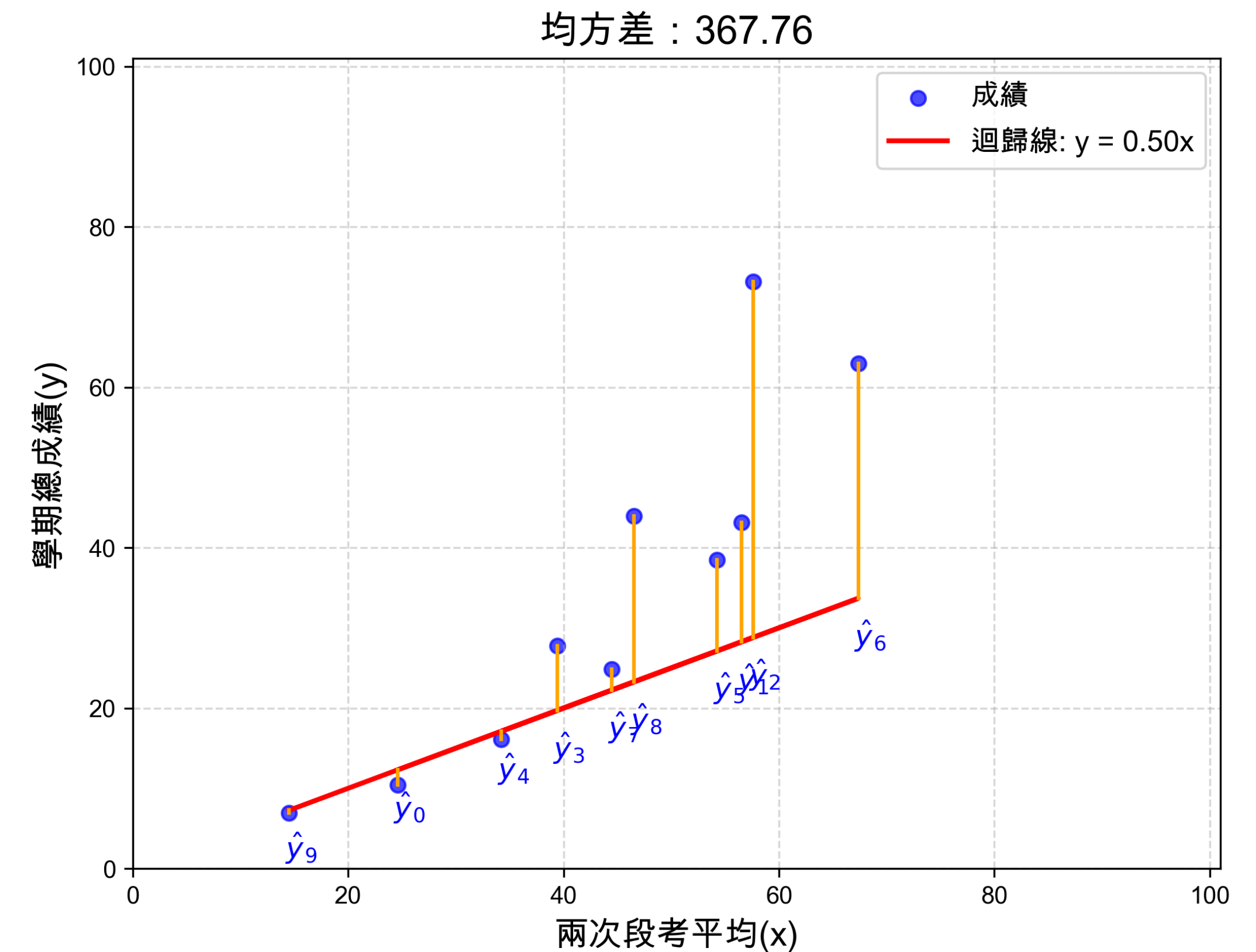
- 在將訓練資料視覺化後，我們發現 x 與 y 的分佈如右圖所示（假設我們詢問了10位學長姐、收集了10筆訓練資料），初步猜測二者可能存在線性關係。
- 接下來的任務就是畫出一條能儘量接近圖上所有資料點的線，也就是找出最佳的參數 a （即這條迴歸線的斜率）



2. 計算損失函數

2.2 模型訓練

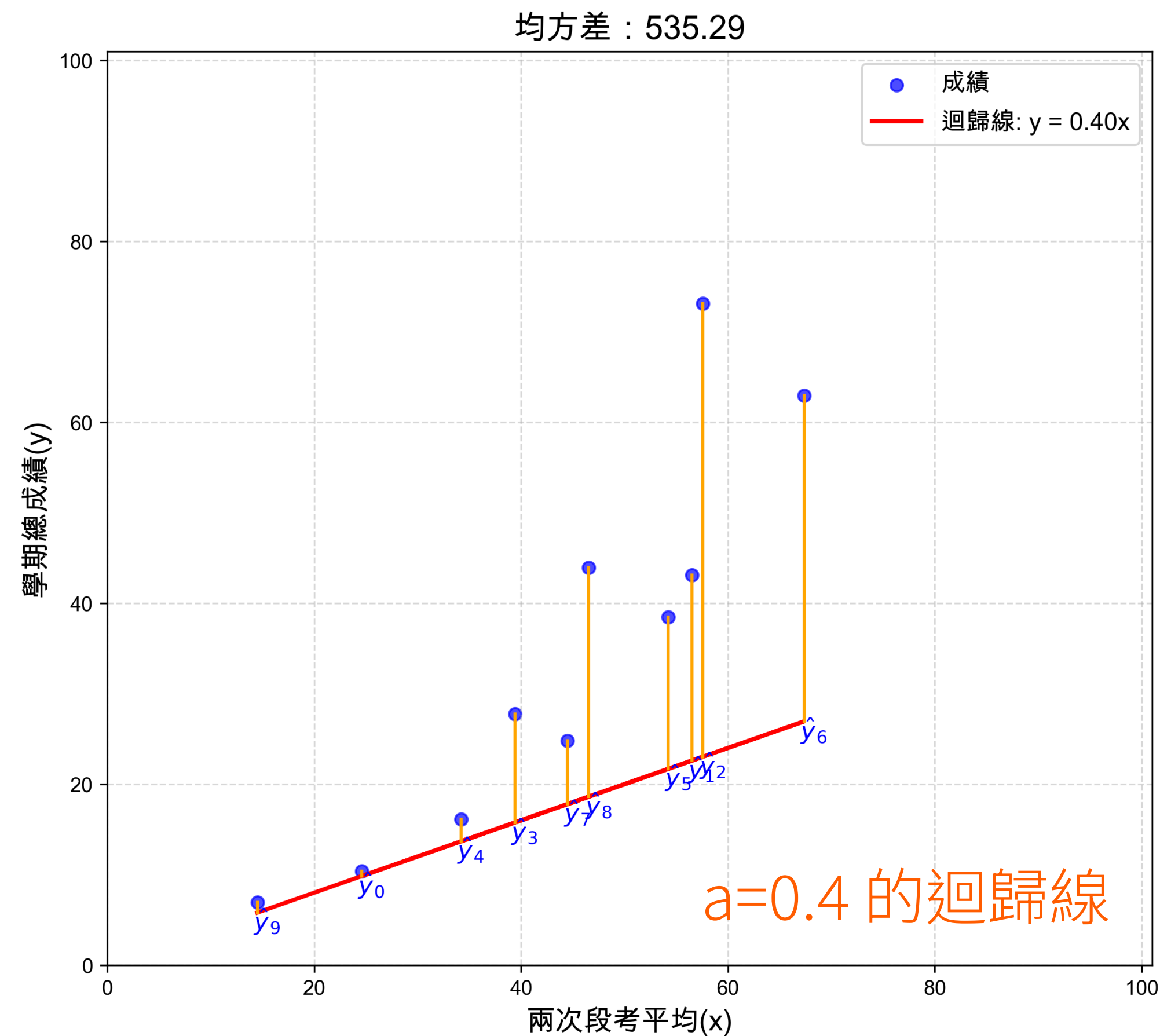
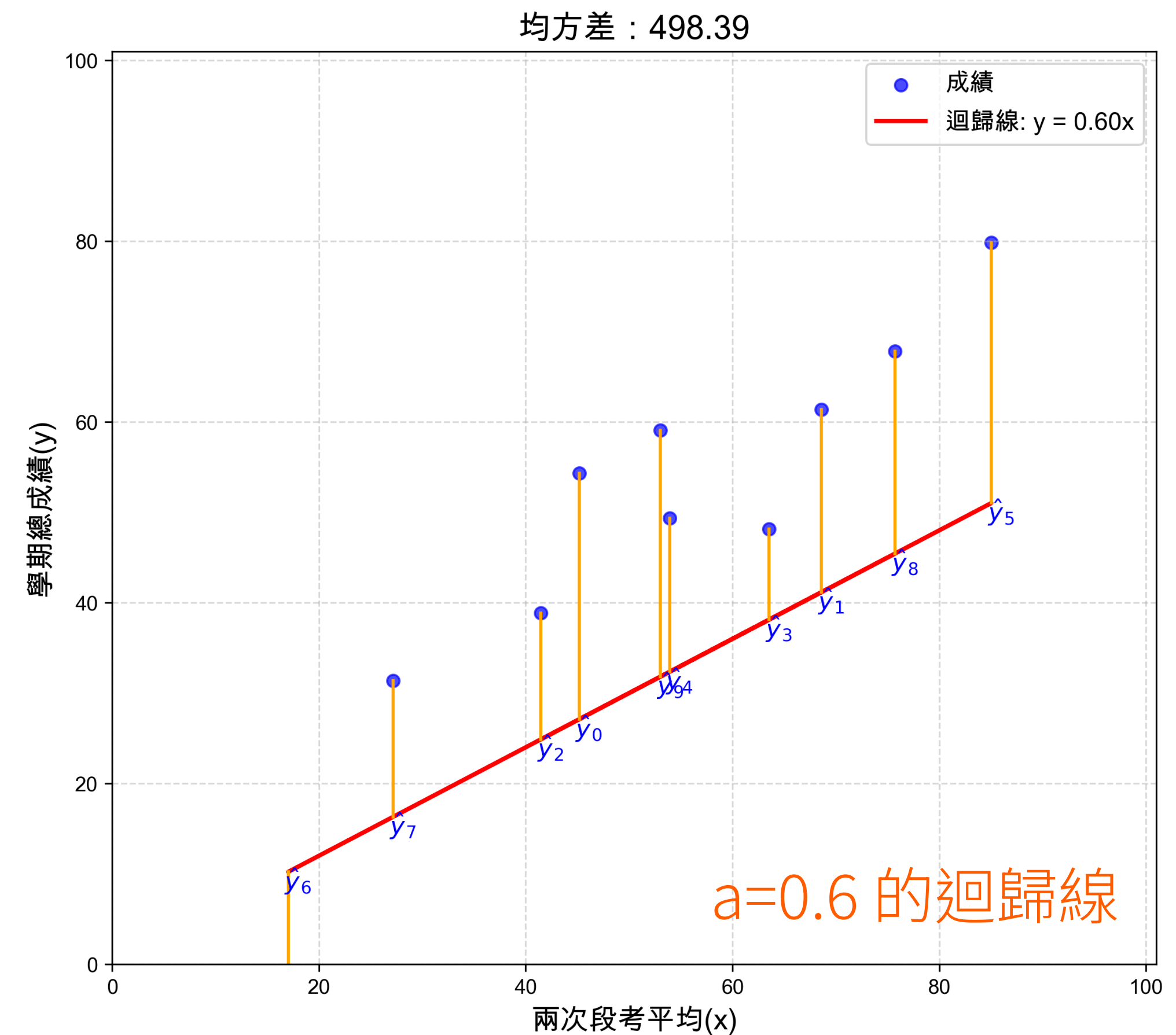
- 計算出這10筆資料與這條線的距離：把所有點與迴歸線的距離加總起來，我們把它稱為誤差。
- 此處以均方差（MSE）來計算，誤差值為367.76。
- MSE公式如下：
$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$
- 這裡的MSE就是這個模型的損失函數（Loss Function）



3. 調整參數：單一參數

2.2 模型訓練

- 目前的a是最好的嗎？我們可以試著調整a值，對它各加、減0.1，再來觀察誤差的變化

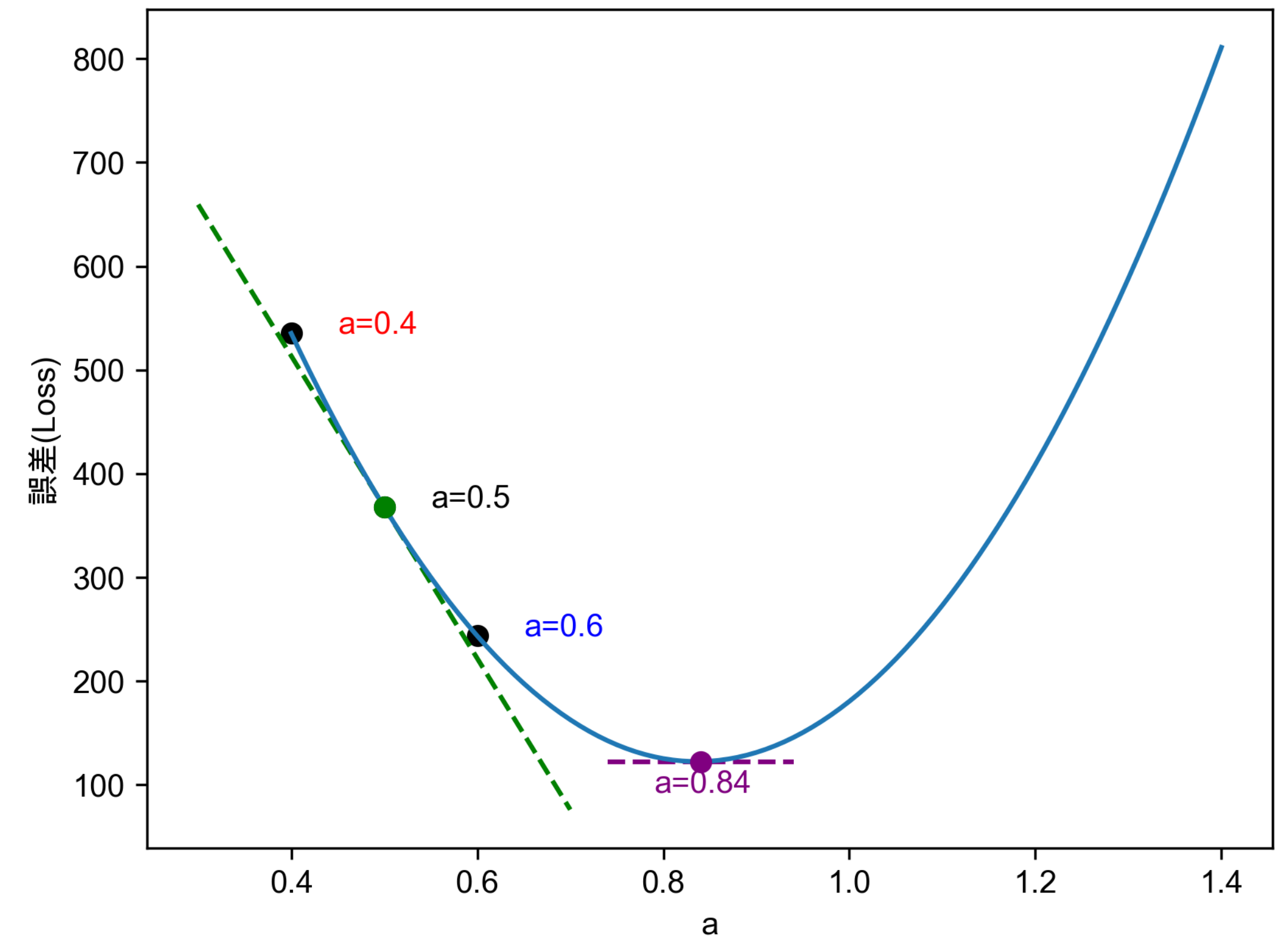


3. 調整參數：單一參數

2.2 模型訓練

- 當我們把全部合理的 a 值對應的損失函數逐一計算出來後，就可以將這些結果畫成一條如右圖的曲線，這就是這個模型的損失函數曲線（Loss Function Curve）。
- 此處以如下公式做為Loss Function:

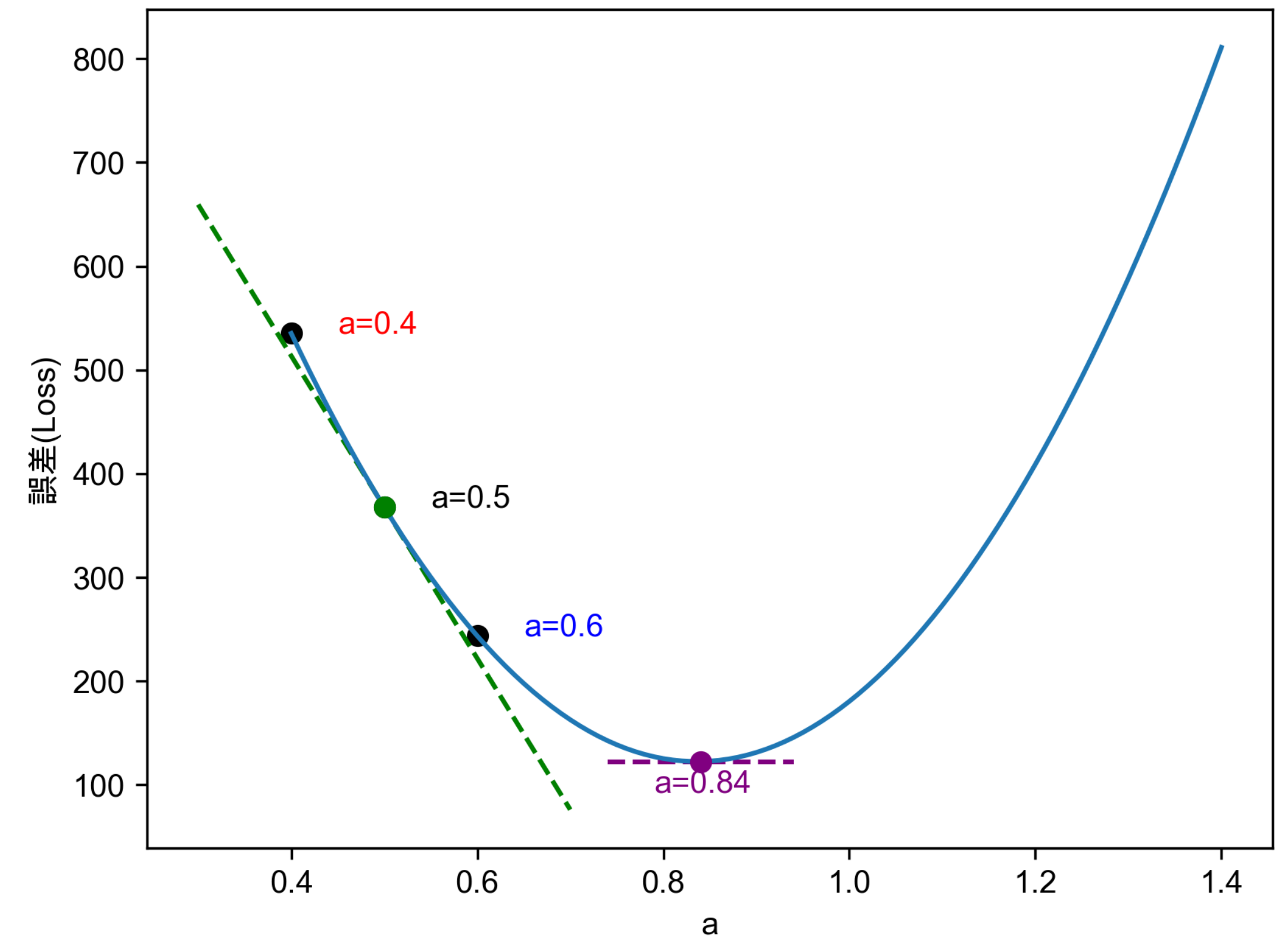
$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$



3. 調整參數：單一參數

2.2 模型訓練

- 只要將 a 逐步往損失函數較小的方向調整，這個模型的預測效果就會更好。
- 調整 a 值的方式是透過資料點在曲線上的切線斜率來判斷該往左或右移動 a 值，直到找到切線斜率為0的點，這便是曲線的最低點，也是最佳 a 值。
- 得到函數： $y = f(x) = 0.84x$



4. 調整參數：兩個參數

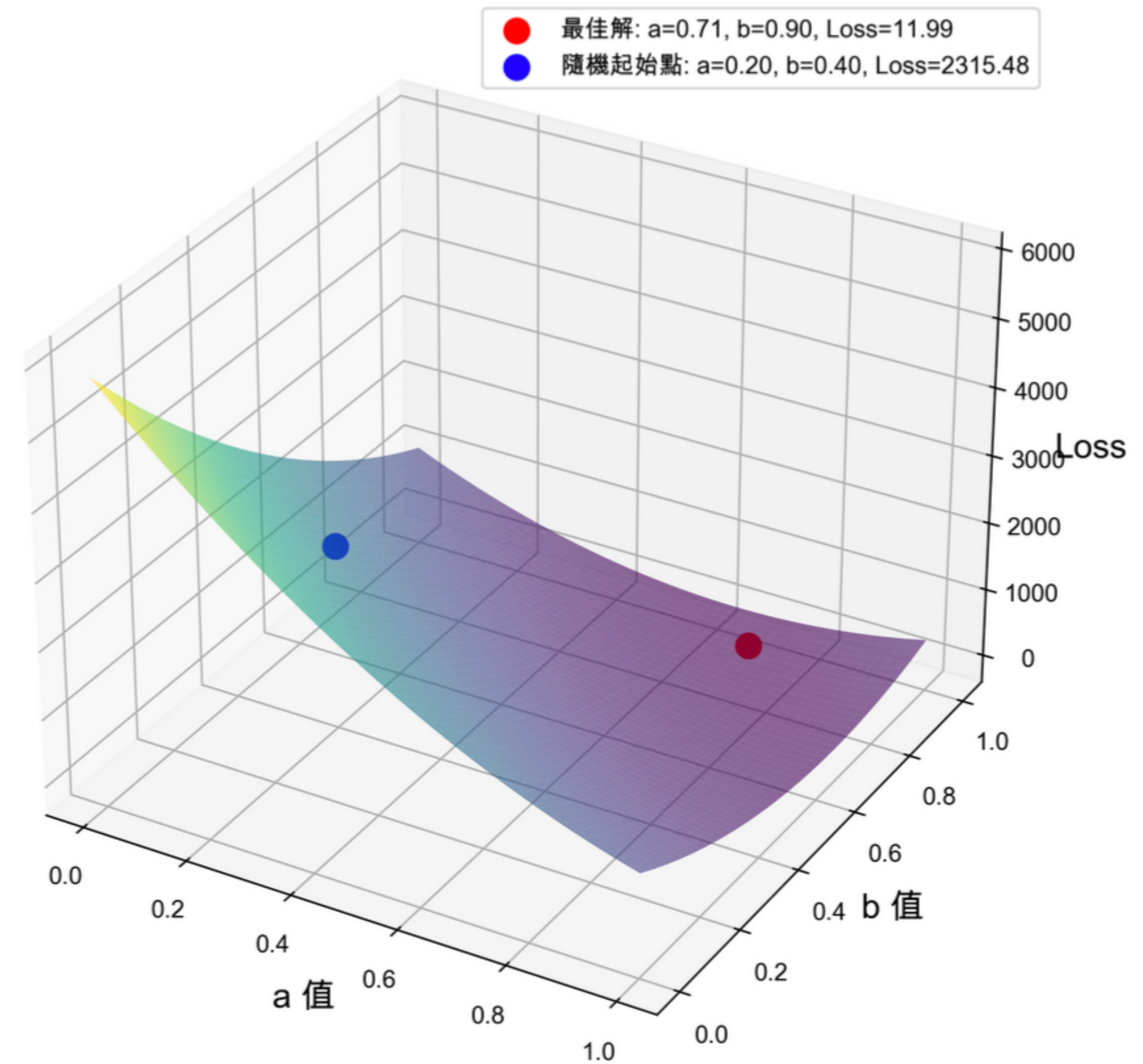
2.2 模型訓練

- 如果老師的計分是針對**期中期末考**給予不同權重呢？那我們新的目標函數就要改成如下式子
- 此時的任務在求出最佳的參數a, b
- $y = f(x) = ax_1 + bx_2$
(x_1 為兩次段考平均成績, x_2 為學期總成績)

梯度下降法

2.2 模型訓練

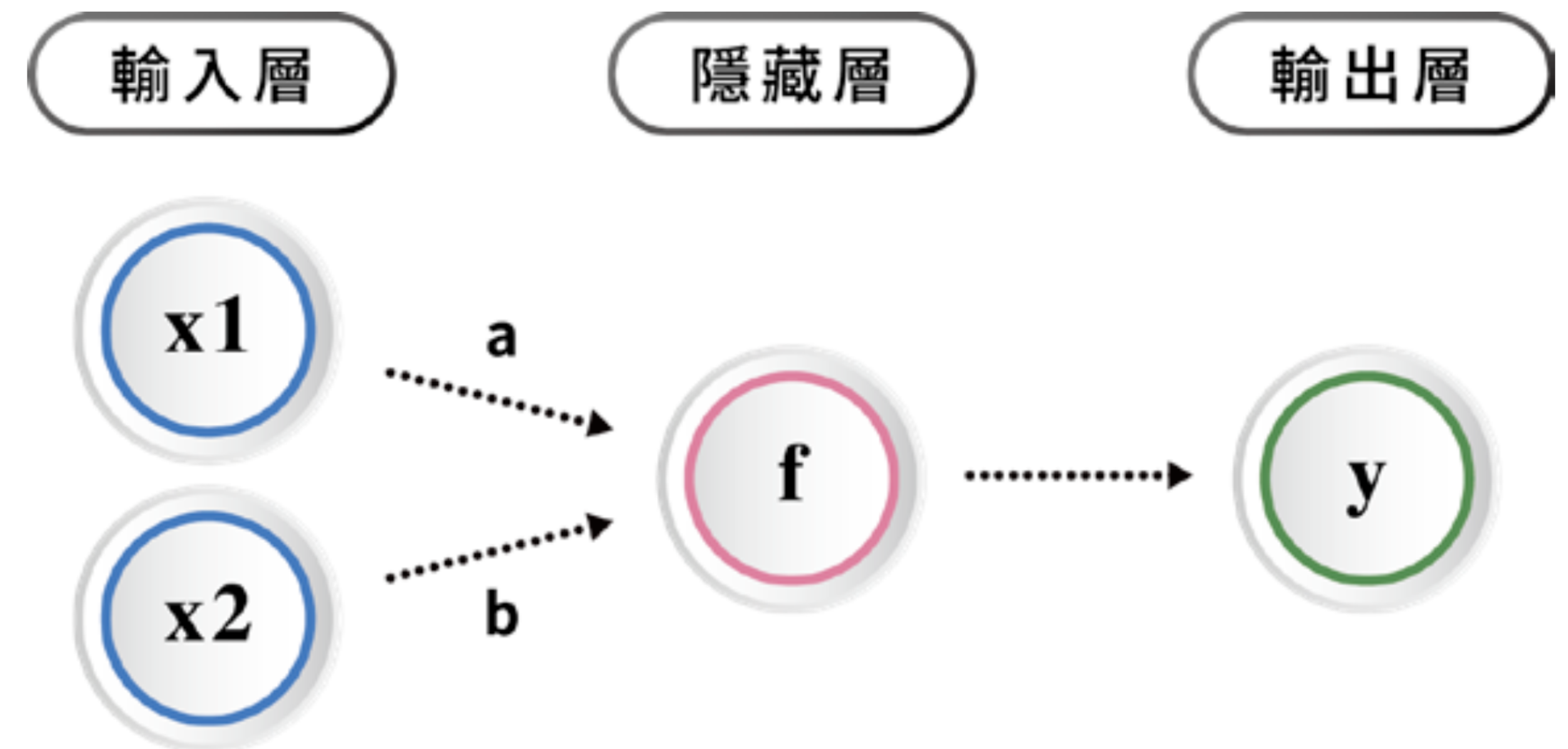
- 比照上述步驟4.的做法，我們可以求出所有合理的 a, b 值（介於0~1間）及其所對應的損失函數，將結果畫成如右圖的曲面。
- 實際求解時，一樣先隨機指定一組權重值 a, b （**藍色點**），依該點在曲面上的斜率（此處稱為梯度，Gradient），沿著梯度的反方向往下找，就能找到這個曲面的最低點（**紅色點**），該點便是函數的最佳參數。
- 此種做法也就是神經網路中找出模型最佳參數的核心思想：**梯度下降法（Gradient Descent）**



4. 調整參數：兩個參數

2.2 模型訓練

- $y = f(x) = ax_1 + bx_2$
(x_1 為兩次段考平均成績, x_2 為學期總成績)
- 從神經網路的角度來看，上述函數也可以視為如右圖的模型，模型中只有一層隱藏層，隱藏層裡面只有一個神經元，輸出結果為期末成績，這便是一個能進行迴歸計算的神經網路。



迴歸與分類

2.2 模型訓練

- 如果我們只關心能否順利取得學分，當初收集的訓練資料也只有學長姐的兩次段考與期末成績是否及格（Pass/Fail），則對應的函數如下式

$$f(\text{期中考, 期末考}) \rightarrow \begin{matrix} \text{Fail : 0.37} \\ \text{Pass : 0.63} \end{matrix}$$

- 函數輸出結果為Pass與Fail的機率，這就是一個能進行分類的模型。

圖形辨識中的參數

2.2 模型訓練

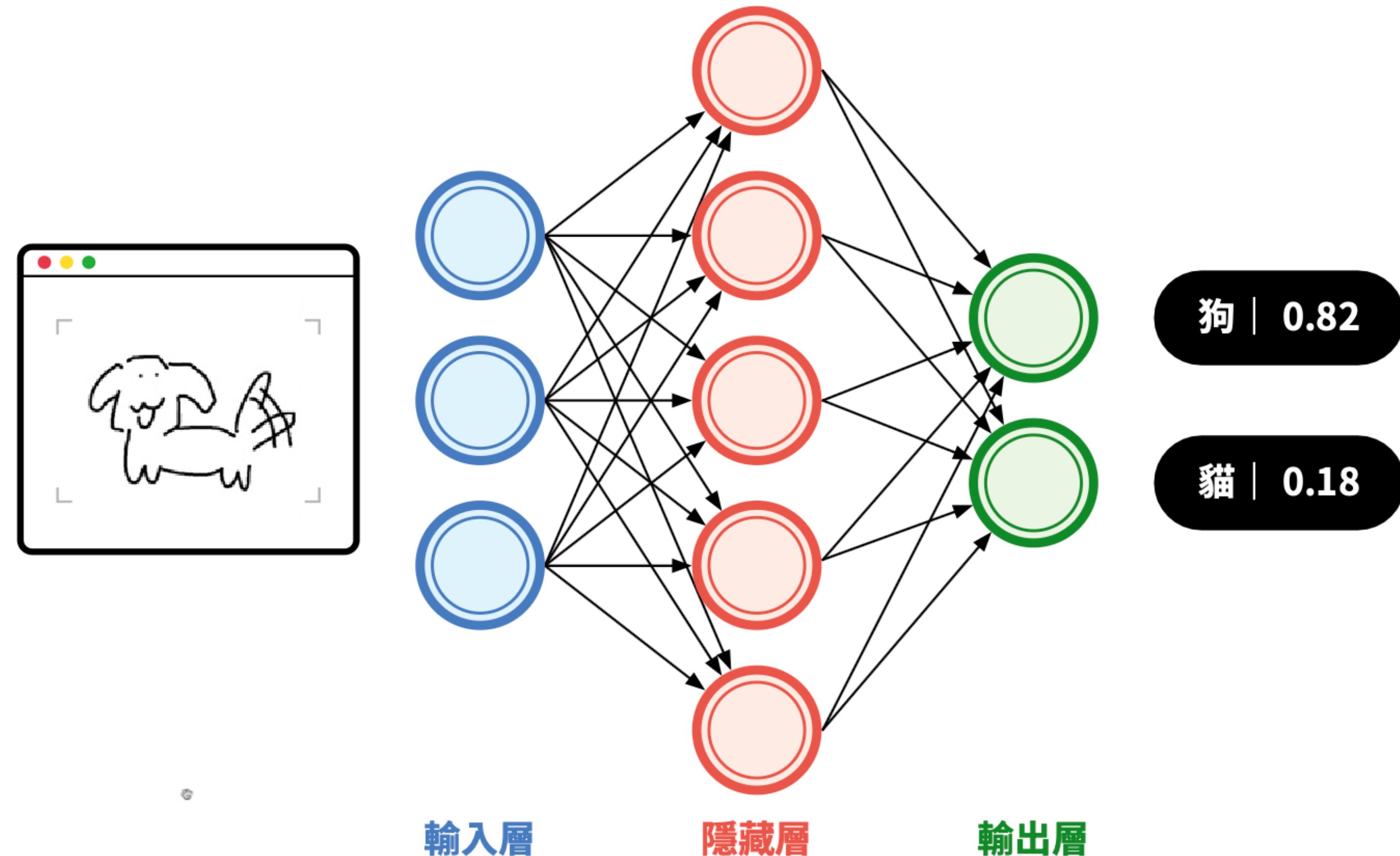
- 圖形辨識也是典型的分類問題，回顧《和AI做朋友》第八章中辨識「/」、「\」兩個符號的矩陣K（圖中綠色框線），矩陣K中的4個數值也可以視為這組模型的4個參數，由於圖形內容過於簡單，我們可以用人力嚐試找出能順利完成任務的4個參數值。

矩陣乘積	矩陣乘積後之元素和	判定圖像
矩陣A與矩陣K之乘積 $\begin{bmatrix} +1 & -1 \\ -1 & +1 \end{bmatrix} \circ \begin{bmatrix} +1 & -1 \\ -1 & +1 \end{bmatrix} = \begin{bmatrix} +1 & +1 \\ +1 & +1 \end{bmatrix}$	$(+1) + (+1) + (+1) + (+1) = 4$	\
矩陣B與矩陣K之乘積 $\begin{bmatrix} -1 & +1 \\ +1 & -1 \end{bmatrix} \circ \begin{bmatrix} +1 & -1 \\ -1 & +1 \end{bmatrix} = \begin{bmatrix} -1 & -1 \\ -1 & -1 \end{bmatrix}$	$(-1) + (-1) + (-1) + (-1) = -4$	/

圖形辨識中的參數

2.2 模型訓練

- 這個模型裡有一個隱藏層、隱藏層中有五個神經元。
- 假設每個神經元裡有一個 3×3 的卷積核負責擷取照片的不同特徵，光這個簡單的小模型就至少有45個參數要估計（未考慮偏移值bias）
- 找出45個最佳參數顯然已非人力所及
- 現今常見的圖形識別模型參數量動輒百萬起跳。



圖形辨識中的參數

2.2 模型訓練

- 而所謂的模型訓練（training）就是利用蒐集到的訓練資料（例如圖像與其對應標籤、或是前例中的兩次段考成績與期末成績），透過不斷更新模型參數，使得模型預測的結果盡可能接近真實標籤的過程。
- 這個過程的核心就是反覆應用梯度下降法來最小化模型的損失函數。換言之，就是找到一組最佳的參數。

2.3 生成式AI：一種會接龍的函數

第2章 生成式AI相關理論

- 為什麼生成式AI能針對問題生成內容，例如

$f(\text{什麼是臺灣最美的風景？}) \rightarrow \text{臺灣最美的風景是人。}$

生成式AI：接龍

2.3 生成式AI：一種會接龍的函數

1. 收到輸入的提示詞「什麼是臺灣最美的風景？」後，函數傳回的結果為下一個字及其出現機率，在此例中，「臺」的機率較高，是最有可能出現的下一個字。

$f(\text{什麼是臺灣最美的風景？}) \rightarrow$

我	: 64%
臺	: 87%
欸	: 21%
⋮	
你	: 78%

生成式AI：接龍

2.3 生成式AI：一種會接龍的函數

2. 繼續以「什麼是臺灣最美的風景？臺」作為函數的輸入，函數傳回的結果可能如下，其中「灣」的出現機率較高，是最有可能出現的下一個字。

$f(\text{什麼是臺灣最美的風景？臺}) \rightarrow$

灣	52%
北	47%
南	38%
⋮	
中	42%

生成式AI：接龍

2.3 生成式AI：一種會接龍的函數

3. 依此類推，接下來函數可能依如下的輸入 / 輸出順序生成最終結果。

1. $f(\text{什麼是臺灣最美的風景？臺灣}) \rightarrow \text{最}$

2. $f(\text{什麼是臺灣最美的風景？臺灣最}) \rightarrow \text{美}$

3. $f(\text{什麼是臺灣最美的風景？臺灣最美}) \rightarrow \text{的}$

4.

5. $f(\text{什麼是臺灣最美的風景？臺灣美的風景是}) \rightarrow \text{人}$

生成式AI：接龍

2.3 生成式AI：一種會接龍的函數

- 透過這種策略，我們已經把生成文字的任務自「從無數種可能去挑出一種最好的答案」簡化為一種「**分類任務**」（和圖形辨識一樣）。

1. $f(\text{什麼是臺灣最美的風景？臺灣}) \rightarrow \text{最}$
2. $f(\text{什麼是臺灣最美的風景？臺灣最}) \rightarrow \text{美}$
3. $f(\text{什麼是臺灣最美的風景？臺灣最美}) \rightarrow \text{的}$
4.
5. $f(\text{什麼是臺灣最美的風景？臺灣美的風景是}) \rightarrow \text{人}$

生成式AI幻覺現象 (hallucination)

2.3 生成式AI：一種會接龍的函數

$f(\text{什麼是台灣最美的風景?}) \rightarrow$

我	: 64%
台	: 87%
欸	: 21%
⋮	
你	: 78%

- $f(\text{什麼是台灣最美的風景?}) \rightarrow$ 我
- $f(\text{什麼是台灣最美的風景?我}) \rightarrow$ 覺
- $f(\text{什麼是台灣最美的風景?我覺}) \rightarrow$ 得
- ...
- $f(\text{什麼是台灣最美的風景?我覺得是阿爾卑斯}) \rightarrow$ 山

Token[實作]

2.3 生成式AI：一種會接龍的函數

- 目前的NLP模型（如GPT、BERT）以token做為函數的輸入 / 輸出的單位，token代表生成式AI用來理解其生成內容的基本單位（同時也是一些生成式API的收費依據）。

GPT-4o & GPT-4o mini		GPT-3.5 & GPT-4	GPT-3 (Legacy)	GPT-4o & GPT-4o mini		GPT-3.5 & GPT-4	GPT-3 (Legacy)
這是一段要求openai協助計算並識別token的文字				這是一段要求openai協助計算並識別token的文字			
Tokens	Characters			Tokens	Characters		
16	27			37	27		
這是一段要求openai協助計算並識別token的文字				??是一?????openai????????????token的????			

Token

2.3 生成式AI：一種會接龍的函數

- 不同版本的GPT中，token與文字的對映關係也有所差異。
- GPT3模型中，部份token與字元存在多對一的關係；到了GPT4，部份token反而能對映到一個以上的字元，
- 這種將字詞轉為token的過稱稱為tokenization，其常見的中譯有「斷詞」、「分詞」或「標記化」。

GPT-4o & GPT-4o miniGPT-3.5 & GPT-4GPT-3 (Legacy)

這是一段要求openai協助計算並識別token的文字

Tokens	Characters
16	27

這是一段要求openai協助計算並識別token的文字

GPT-4o & GPT-4o miniGPT-3.5 & GPT-4GPT-3 (Legacy)

這是一段要求openai協助計算並識別token的文字

Tokens	Characters
37	27

??是一?????openai????????????????????tokens的????

文字理解轉換為數學運算

2.3 生成式AI：一種會接龍的函數

- Tokenization還能讓模型靈活處理多語言文本，例如本文中就有許多中、英文夾雜的內容，透過tokenization就能對這些多國語言統一處理；
- 對於模型從未見過的陌生詞語（out-of-vocabulary, OOV），模型也能將該陌生詞語拆解為已知的基本單位再轉換為對應的token。

GPT-4o & GPT-4o miniGPT-3.5 & GPT-4GPT-3 (Legacy)

這是一段要求openai協助計算並識別token的文字

Tokens	Characters
16	27

這是一段要求openai協助計算並識別token的文字

GPT-4o & GPT-4o miniGPT-3.5 & GPT-4GPT-3 (Legacy)

這是一段要求openai協助計算並識別token的文字

Tokens	Characters
37	27

??是一??openai????token的??

接龍方式生成圖形

2.3 生成式AI：一種會接龍的函數

- OpenAI於2020年提出的Image GPT便是以像素接龍的方式來生成影像，可以在「只提供影像上半部」的狀況下生成「影像下半部」



接龍方式生成語音

2.3 生成式AI：一種會接龍的函數

- DeepMind於2016年提出的模型WaveNet便是採取逐個取樣（sample，將連續的聲音信號分割成許多細小的片段，每個片段就代表一個取樣）來生成整段語音。
- 這種逐段生成的方法能產生更加自然流暢的語音，WaveNet後來也被Google應用於其語音合成服務（Text-to-Speech）、Google助理以及Google智慧音箱。

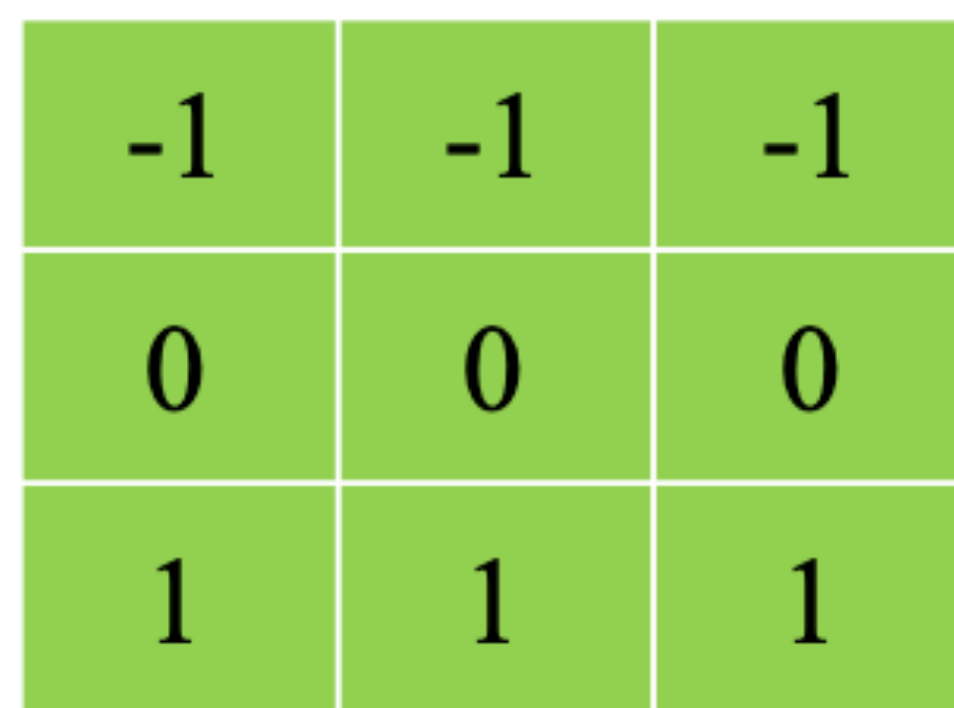
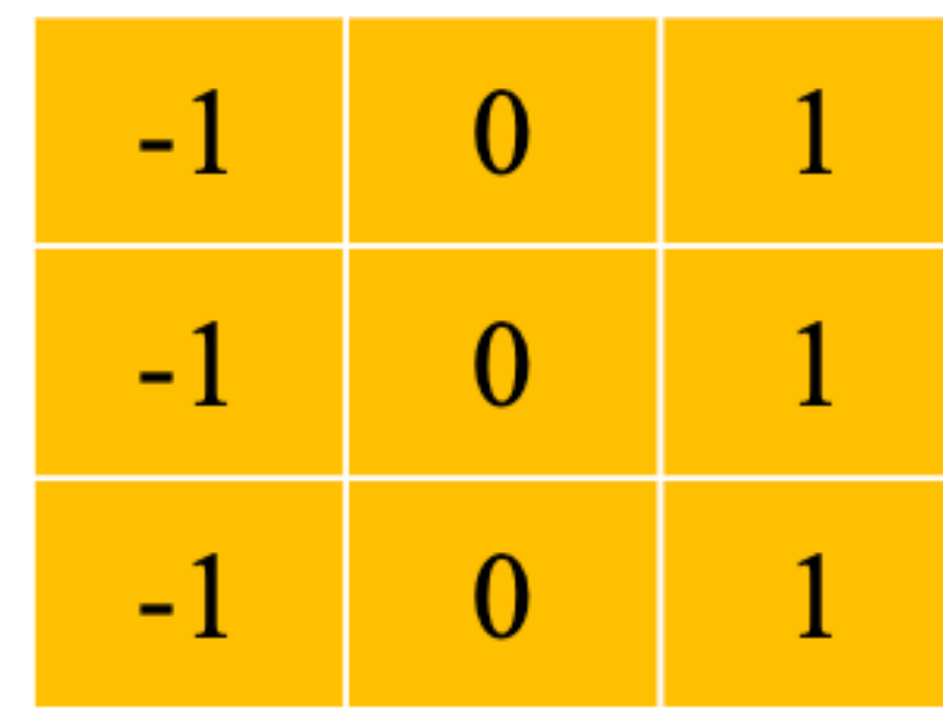
AR vs. NAR

2.3 生成式AI：一種會接龍的函數

- 本節所介紹的「接龍式生成」其架構為一種自迴歸模型（Autoregressive Model, AR），這種模型每次生成一個元素，並與之前生成的內容結合做為新的輸入，來預測下一個元素，這也是目前較常見的生成策略，GPT系列即為典型代表。自迴歸模型的優點是生成內容有良好的連貫性，缺點則是生成速度較慢，而且無法做平行處理。
- 如果要以像素接龍的方式生成一張 1024×1024 的圖，我們一共要進行104萬次像素接龍。
- 另一種生成策略為非自迴歸模型（Non-Autoregressive Model, NAR），這種模型一次生成完整內容，不依賴之前生成的元素，典型的代表模型為GAN。這種模型的優點為生成速度快，而且能夠進行平行加速處理，缺點則是生成結果可能連貫性較差。

[illegible]

- 此處以工人智慧來自訂兩個卷積核，每個卷積核理解影像的一種特徵
- 實務運作時，成千上萬個卷積核先由模型隨機生成，接下來藉由模型的訓練來逐步找出所有卷積核的最佳參數數值



[illegible]

- 要理解影像內容，我們將像素轉化為矩陣，然後通過卷積運算提取影像特徵；
- 理解文字的方法之一是詞嵌入（Word Embedding）：一種將文字轉換為數學向量的做法。
 - 詞嵌入的研究從2000年開始，在2013年Google提出了一種可大幅提升了詞嵌入模型訓練速度的方法： Word2Vec
- Word2Vec的核心思想是：將詞語以一個固定長度的向量來表示，常見的向量長度從數十到上百維不等
- 每個維度都代表了從不同角度來理解這個詞語的某種特徵，藉由對詞語向量進行數學運算，我們甚至能找到詞語間的關係。

如何理解文字

2.4 文字、圖形與向量：先理解才能生成

- 以word2vec-google-news-300把文字理解的問題轉變為向量計算的數學問題：
- The man loved the woman like queen while she adored him like king.

單字	單字對應的 Vector（前五項）				
The	-0.17285156	0.27929688	0.10693359	-0.15820312	-0.08447266
man	0.32617188	0.13085938	0.03466797	-0.08300781	0.08984375
loved	0.04052734	-0.09619141	0.14648438	0.2421875	0.02856445
...	...				
woman	0.24316406	-0.07714844	-0.10302734	-0.10742188	0.11816406
queen	0.00524902	-0.14355469	-0.06933594	0.12353516	0.13183594
...	...				
adored	0.09033203	0.10253906	0.25390625	0.05151367	-0.15917969
king	0.12597656	0.02978516	0.00860596	0.13964844	-0.02563477

如何理解文字

2.4 文字、圖形與向量：先理解才能生成

單字	單字對應的 Vector (前五項)				
The	-0.17285156	0.27929688	0.10693359	-0.15820312	-0.08447266
man	0.32617188	0.13085938	0.03466797	-0.08300781	0.08984375
loved	0.04052734	-0.09619141	0.14648438	0.2421875	0.02856445
...	...	...	...	...	...
woman	0.24316406	-0.07714844	-0.10302734	-0.10742188	0.11816406
queen	0.00524902	-0.14355469	-0.06933594	0.12353516	0.13183594
...	...	...	...	...	...
adored	0.09033203	0.10253906	0.25390625	0.05151367	-0.15917969
king	0.12597656	0.02978516	0.00860596	0.13964844	-0.02563477

- word2vec-google-news-300是一個利用 Google News 這個大規模語料庫進行訓練的Word2Vec模型，模型的訓練過程使用了Skip-Gram方法，這種方法專注於捕捉單詞與上下文之間的語義關係。
- 模型word2vec-google-news-300包含了三百萬個詞彙，每個詞彙都對應到一個長度為300的向量（300維）。
- 這些高維度的向量除了能表達該單字的語意與語法，我們還可以利用向量來計算字與字之間的相關、字與字間的距離、甚或對向量進行更複雜的運算，也就是：把文字理解的問題轉變為向量計算的數學問題。

降維

2.4 文字、圖形與向量：先理解才能生成

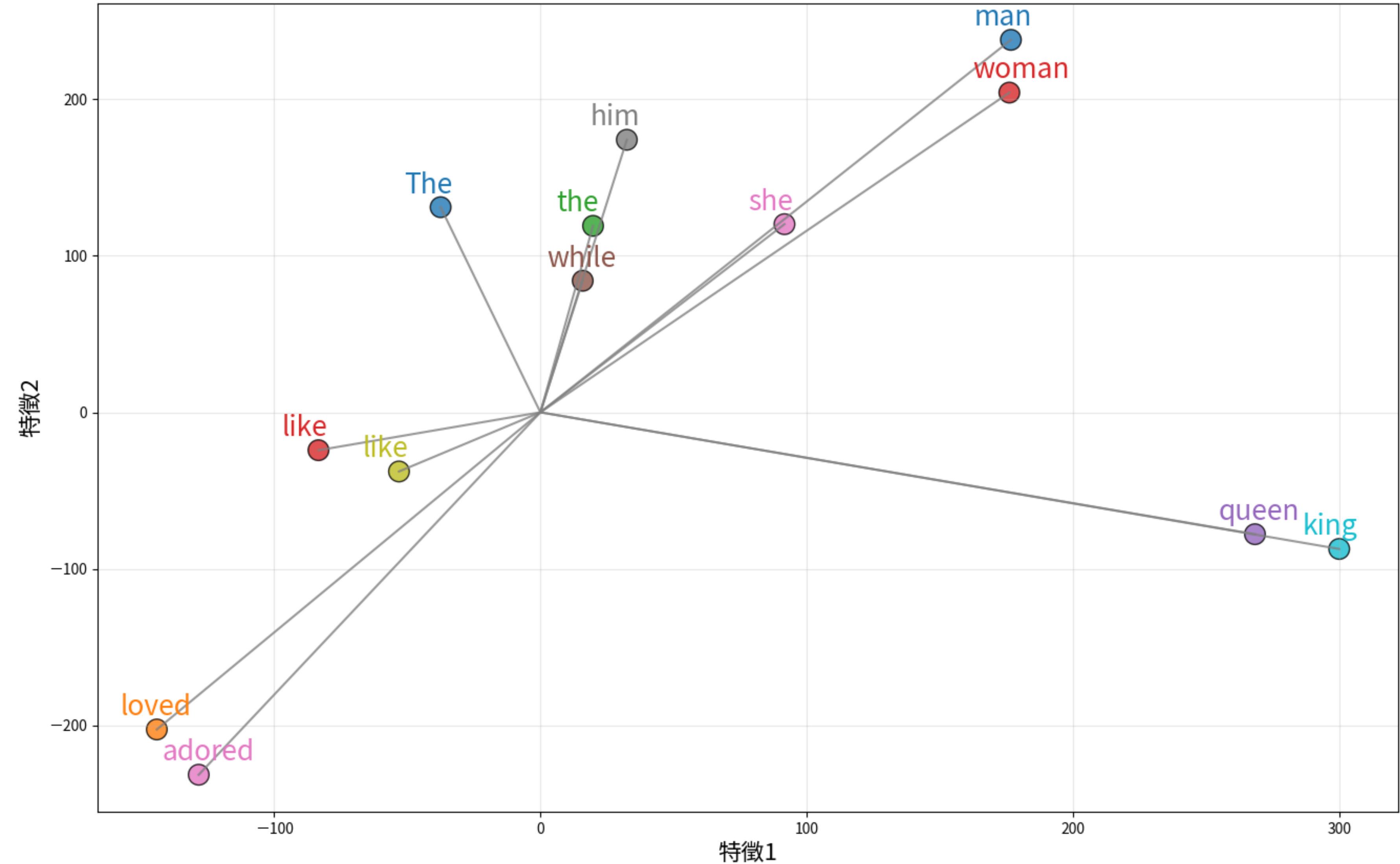
單字	單字對應的 Vector（前五項）				
The	-0.17285156	0.27929688	0.10693359	-0.15820312	-0.08447266
man	0.32617188	0.13085938	0.03466797	-0.08300781	0.08984375
loved	0.04052734	-0.09619141	0.14648438	0.2421875	0.02856445
...	...				
woman	0.24316406	-0.07714844	-0.10302734	-0.10742188	0.11816406
queen	0.00524902	-0.14355469	-0.06933594	0.12353516	0.13183594
...	...				
adored	0.09033203	0.10253906	0.25390625	0.05151367	-0.15917969
king	0.12597656	0.02978516	0.00860596	0.13964844	-0.02563477

- 由於這樣高維度的資料難以讓我們以肉眼直觀看出其相關或差異，此處我們先將這些300維的向量進行降維（dimensionality reduction）處理（降至二維），以方便我們在平面座標中標示出這些單字的相對位置。
- 降維
 - 照相，就是在把原本為三度空間的景物降維至二維平面：透過被降維的相片，我們仍能觀察出相片中景物的三維空間關係（遠方景物小、近處景物大）
 - 學校把我們在學期中各學科的作業、大小考試、報告、平時表現等分數簡化為各科期末成績：透過被降維的期末成績，我們也能大致了解自己在不同科目的能力差異，還能由期末成績了解到自己在團體中的能力排名。

降維

2.4 文字、圖形與向量：先理解才能生成

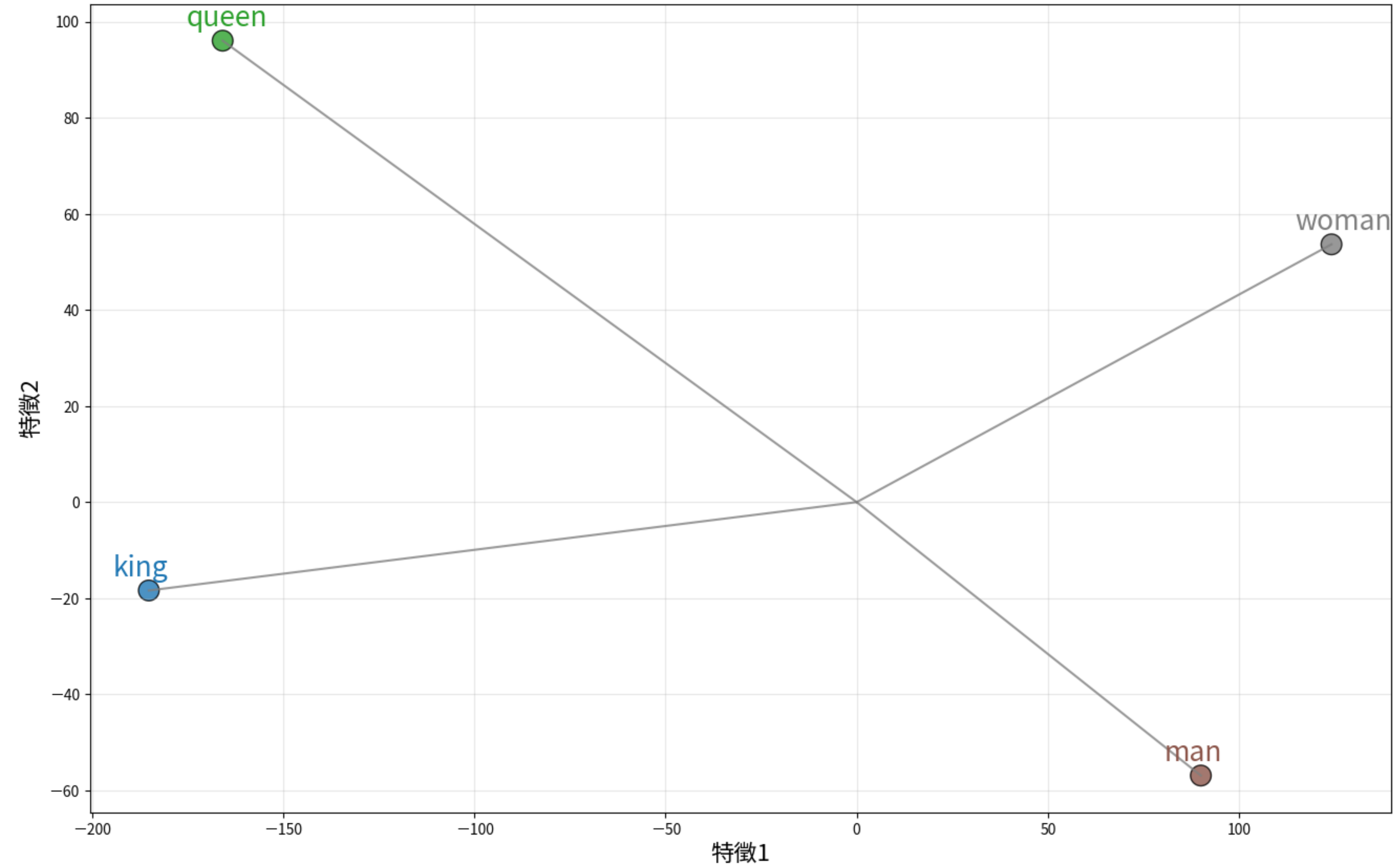
單字	單字對應的 Vector（前五項）				
The	-0.17285156	0.27929688	0.10693359	-0.15820312	-0.08447266
man	0.32617188	0.13085938	0.03466797	-0.08300781	0.08984375
loved	0.04052734	-0.09619141	0.14648438	0.2421875	0.02856445
...	...				
woman	0.24316406	-0.07714844	-0.10302734	-0.10742188	0.11816406
queen	0.00524902	-0.14355469	-0.06933594	0.12353516	0.13183594
...	...				
king	0.390625	0.05151367	-0.15917969		
...	0.1860596	0.13964844	-0.02563477		



woman, man, queen, king

2.4 文字、圖形與向量：先理解才能生成

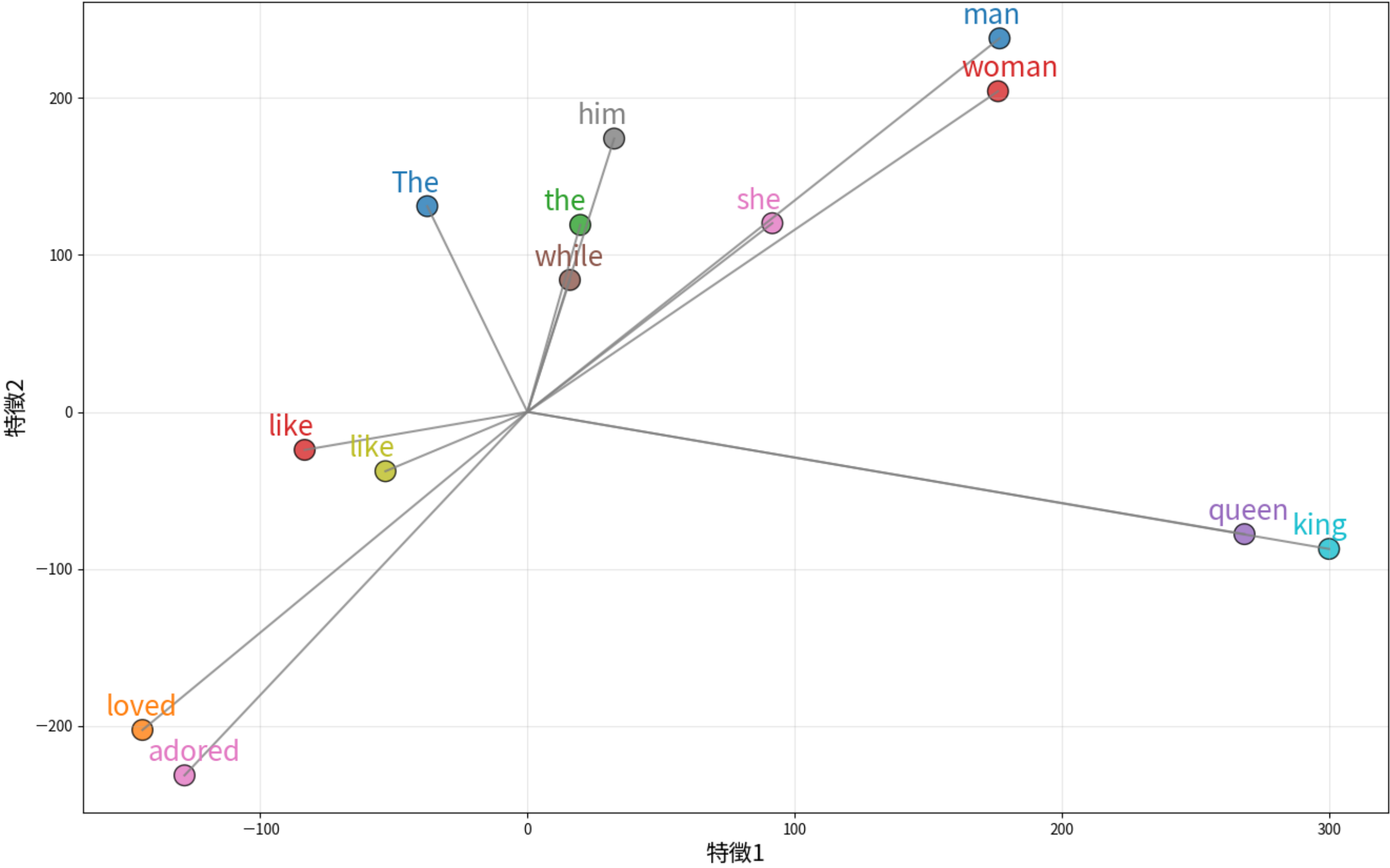
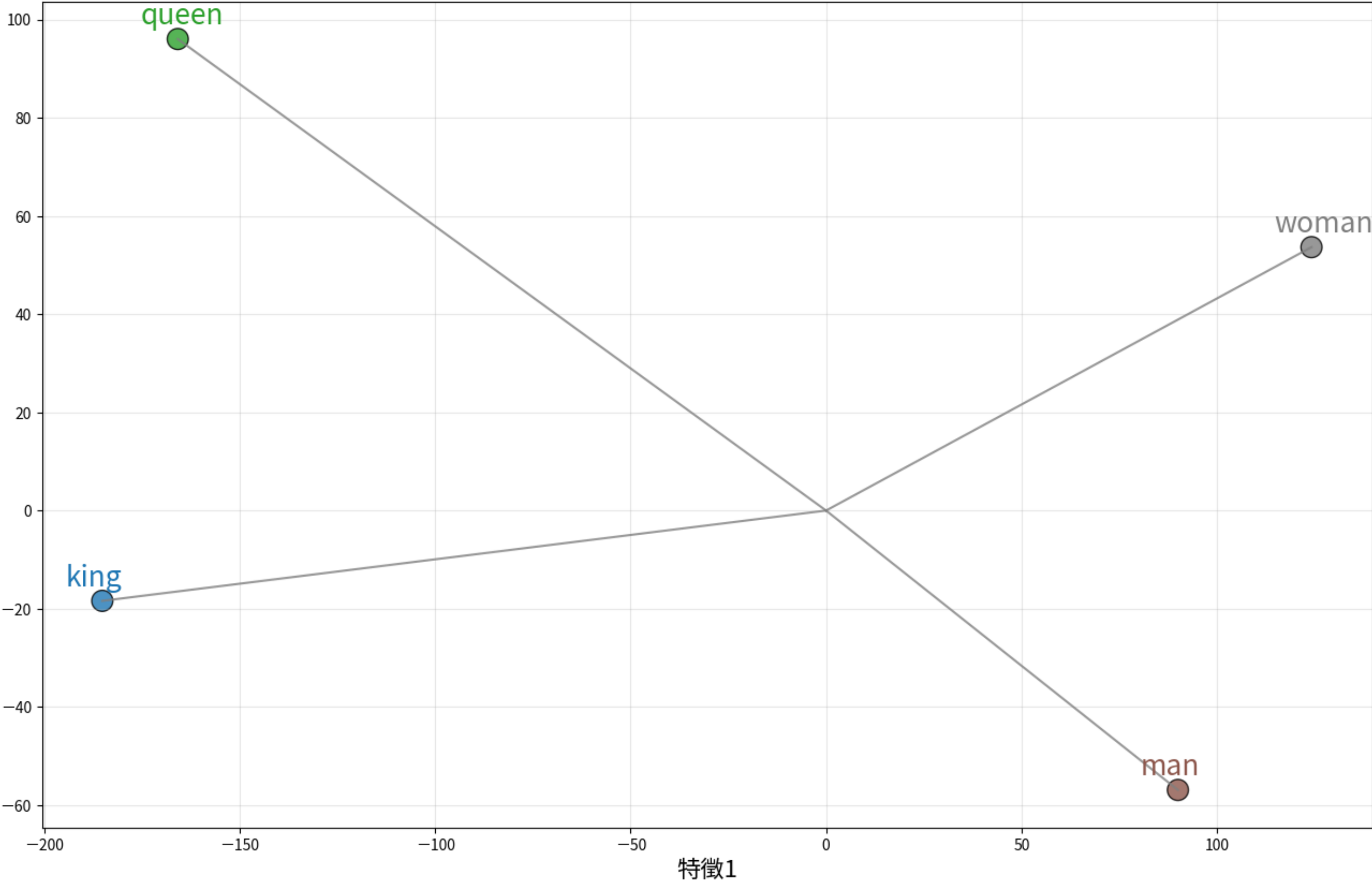
單字	單字對應的 Vector (前五項)				
The	-0.17285156	0.27929688	0.10693359	-0.15820312	-0.08447266
man	0.32617188	0.13085938	0.03466797	-0.08300781	0.08984375
loved	0.04052734	-0.09619141	0.14648438	0.2421875	0.02856445
...	...				
woman	0.24316406	-0.07714844	-0.10302734	-0.10742188	0.11816406
queen	0.00524902	-0.14355469	-0.06933594	0.12353516	0.13183594
...	...				
	0.390625	0.05151367	-0.15917969		
	0.0860596	0.13964844	-0.02563477		



為什麼向量內容改變了？

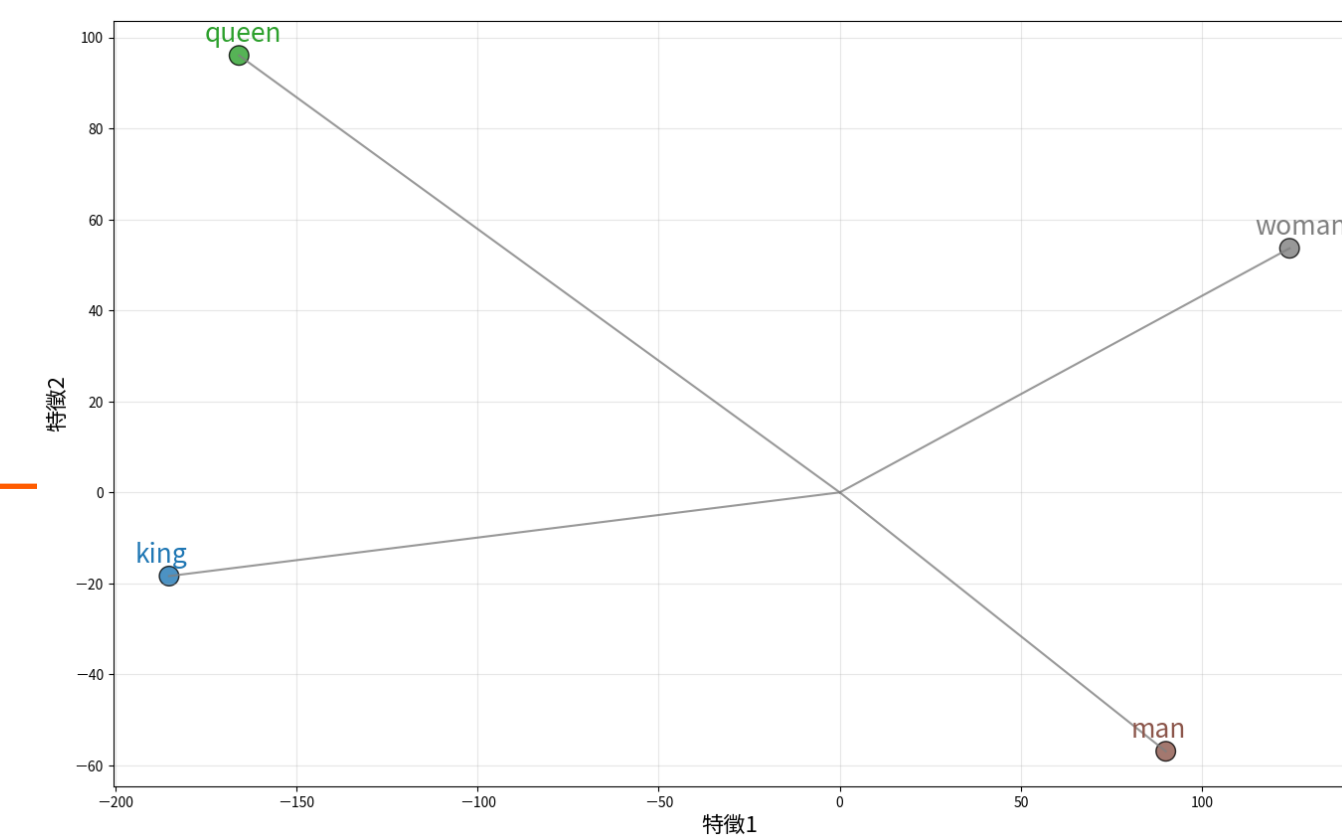
2.4 文字、圖形與向量：先理解才能生成

單字	單字對應的 Vector（前五項）				
The	-0.17285156	0.27929688	0.10693359	-0.15820312	-0.08447266
man	0.32617188	0.13085938	0.03466797	-0.08300781	0.08984375
loved	0.04052734	-0.09619141	0.14648438	0.2421875	0.02856445
...	...				
woman	0.24316406	-0.07714844	-0.10302734	-0.10742188	0.11816406
queen	0.00524902	-0.14355469	-0.06933594	0.12353516	0.13183594
...	...				
adored	0.09033203	0.10253906	0.25390625	0.05151367	-0.15917969
king	0.12597656	0.02978516	0.00860596	0.13964844	-0.02563477



woman, man, queen, king

2.4 文字、圖形與向量：先理解才能生成



- 如果我們以woman, man, queen, king這四個單字的向量（參考表 2.4-1）進行下列運算：

- $\overrightarrow{king} - \overrightarrow{man} + \overrightarrow{woman}$

- 以計算所得到的向量搜尋除了這3個單字以外、最接近這個計算結果的向量

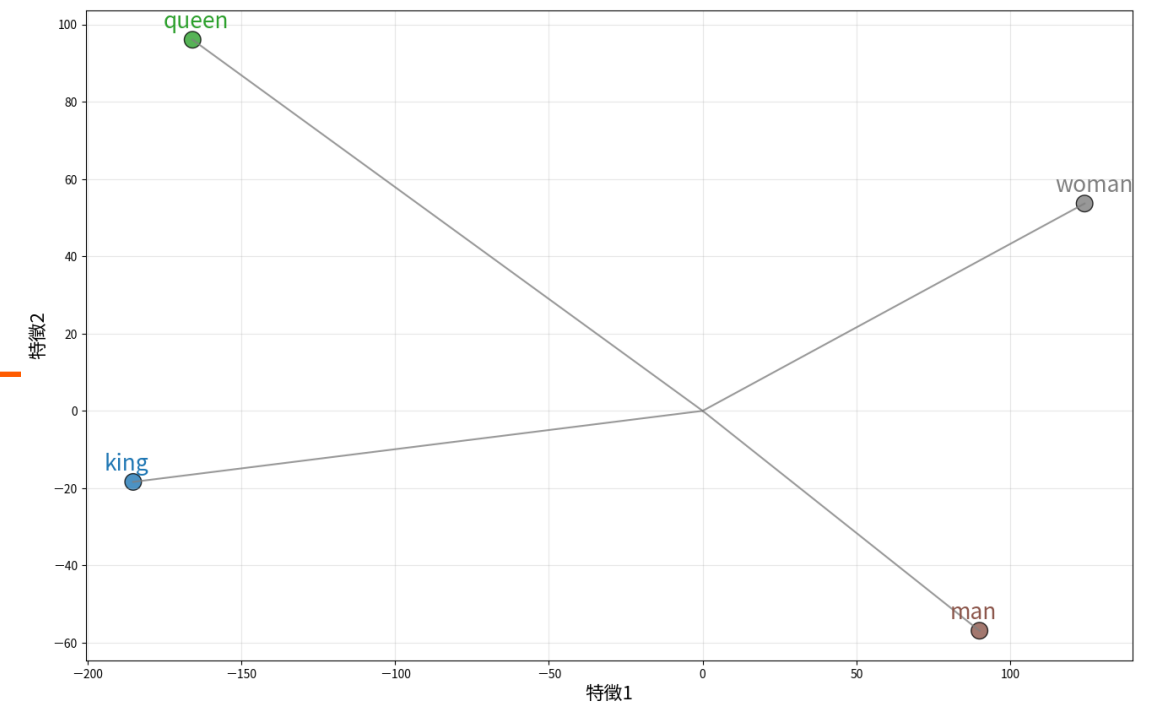
- 答案會是：queen

- 這就是自然語言處理中的經典等式

- $\overrightarrow{king} - \overrightarrow{man} + \overrightarrow{woman} \approx \overrightarrow{queen}$

探討活動：字詞關係

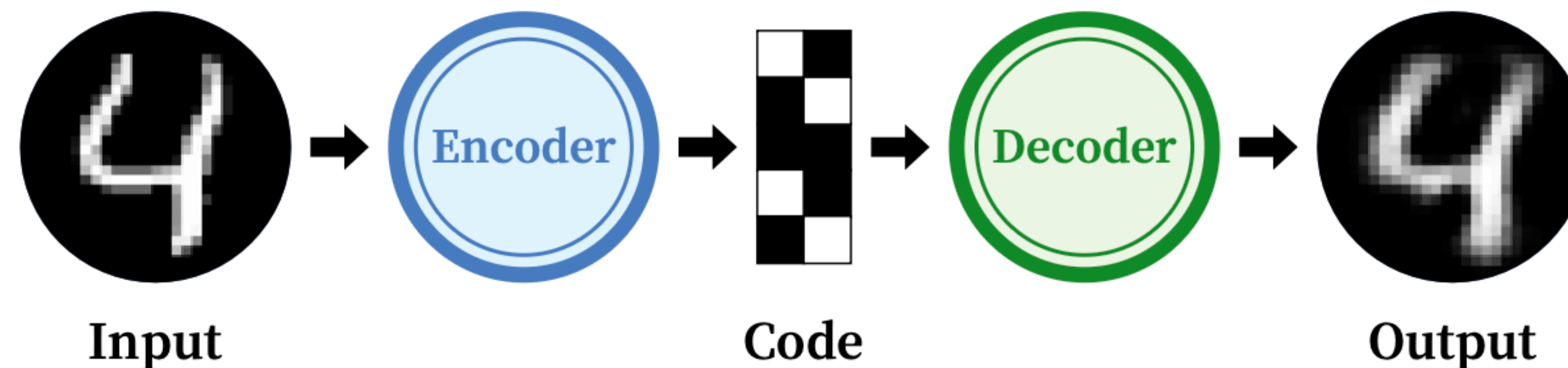
2.4 文字、圖形與向量：先理解才能生成



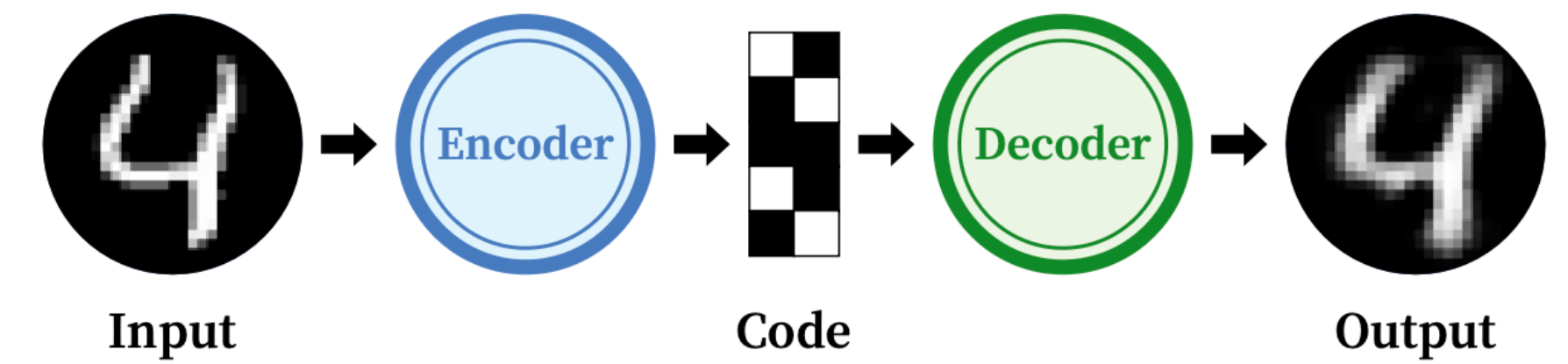
- 閱讀程式碼與註解，理解範例程式的原理。
- 請小組組員自行找出另外的單字組合，輸出這些單字的前五個向量值。
- 將這些單字向量降維，以圖形化的方式呈現單字間的關係，並解釋這些關係。
- 請組員試著找出這些單字間是否存在著類似 的關係。
- 程式碼網址：<https://tinyurl.com/GAI-W2V-CODE>

2.5 編碼與解碼：邁向生成的一小步

- 編碼與解碼為生成式AI的核心機制，是一種將輸入模型的影像編碼後再解碼還原回原圖的過程，此一生成的過程也叫**重建**（Reconstruction）
- 負責擷取特徵並將特徵壓縮降維的神經網路稱為編碼器（Encoder）
- 被壓縮降維的向量稱為Code
- 負責根據Code生成原圖的神經網路則稱為解碼器（Decoder）
- 這整個進行編碼、解碼的架構便是**自動編碼器**（AutoEncoder，AE）

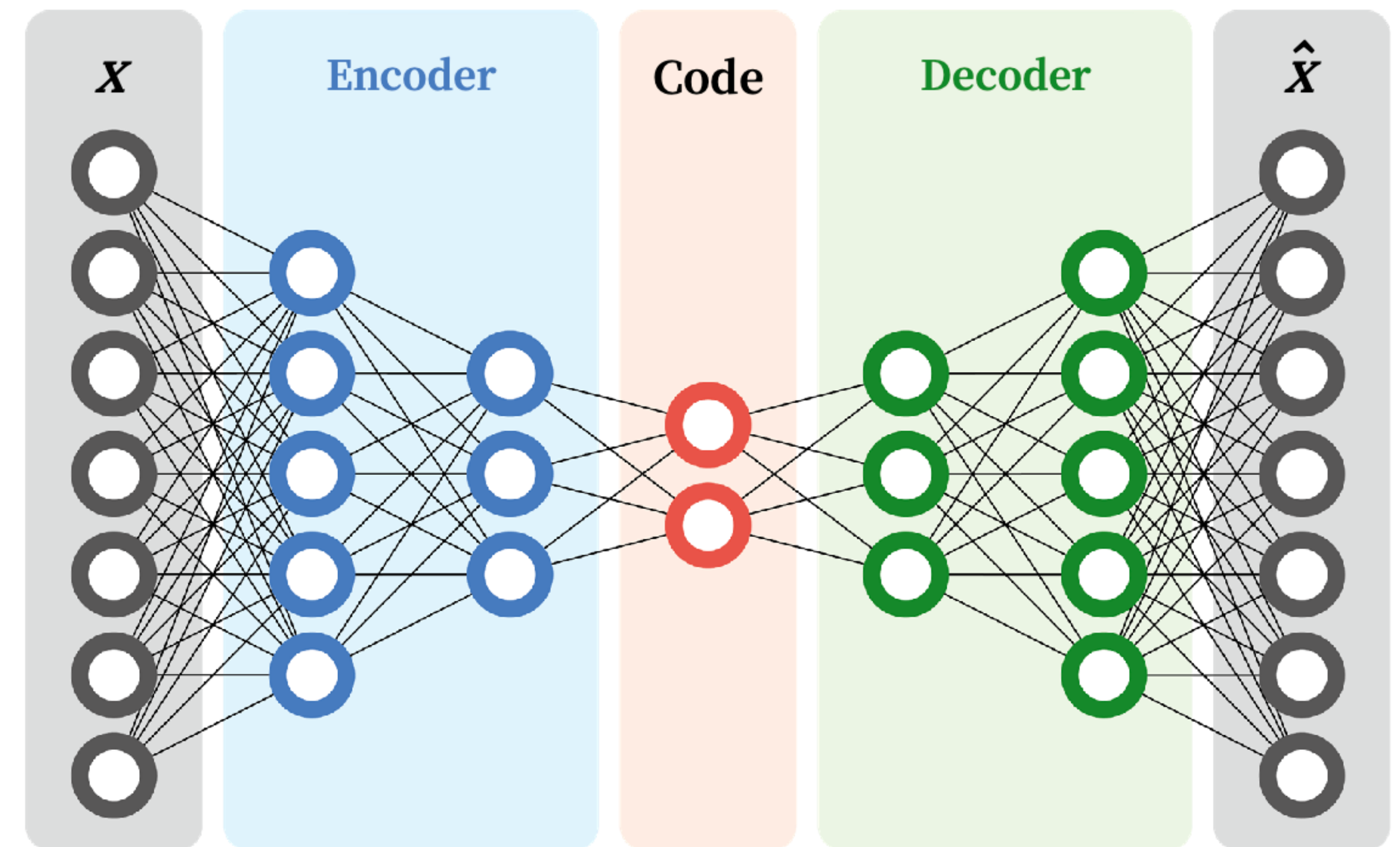


Code

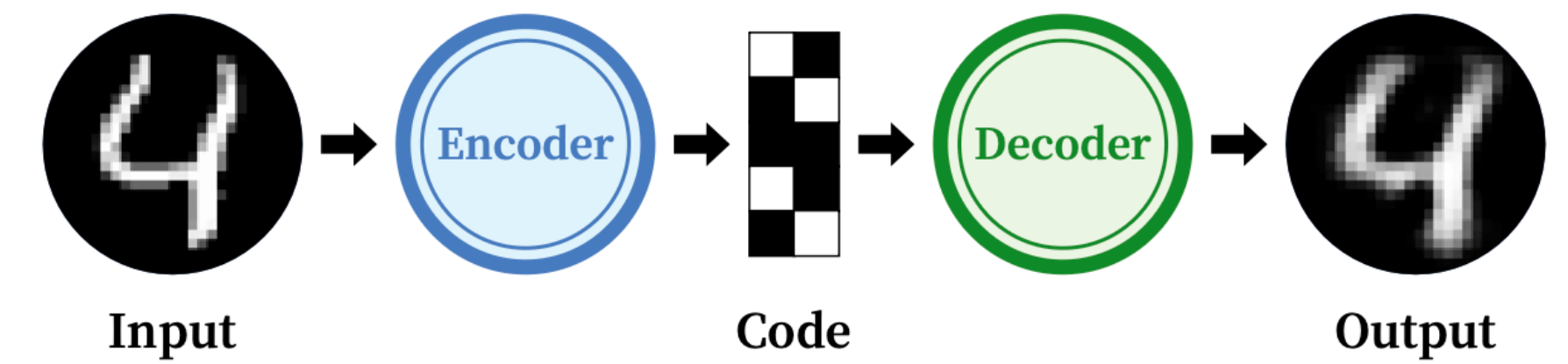


2.5 編碼與解碼：邁向生成的一小步

- Code在不同情境下有時也有其他名稱，如：
 - Latent Representation：強調此向量是原始資料在低維空間中的潛在表達方式。
 - Latent Vector / Latent Space（潛在向量、潛在空間）：強調其為模型中的一層與其向量型態。
 - Latent Code（潛在向量）：結合Latent Vector與Code。
 - Embedding（嵌入）：常見於表示詞向量或文本嵌入時。

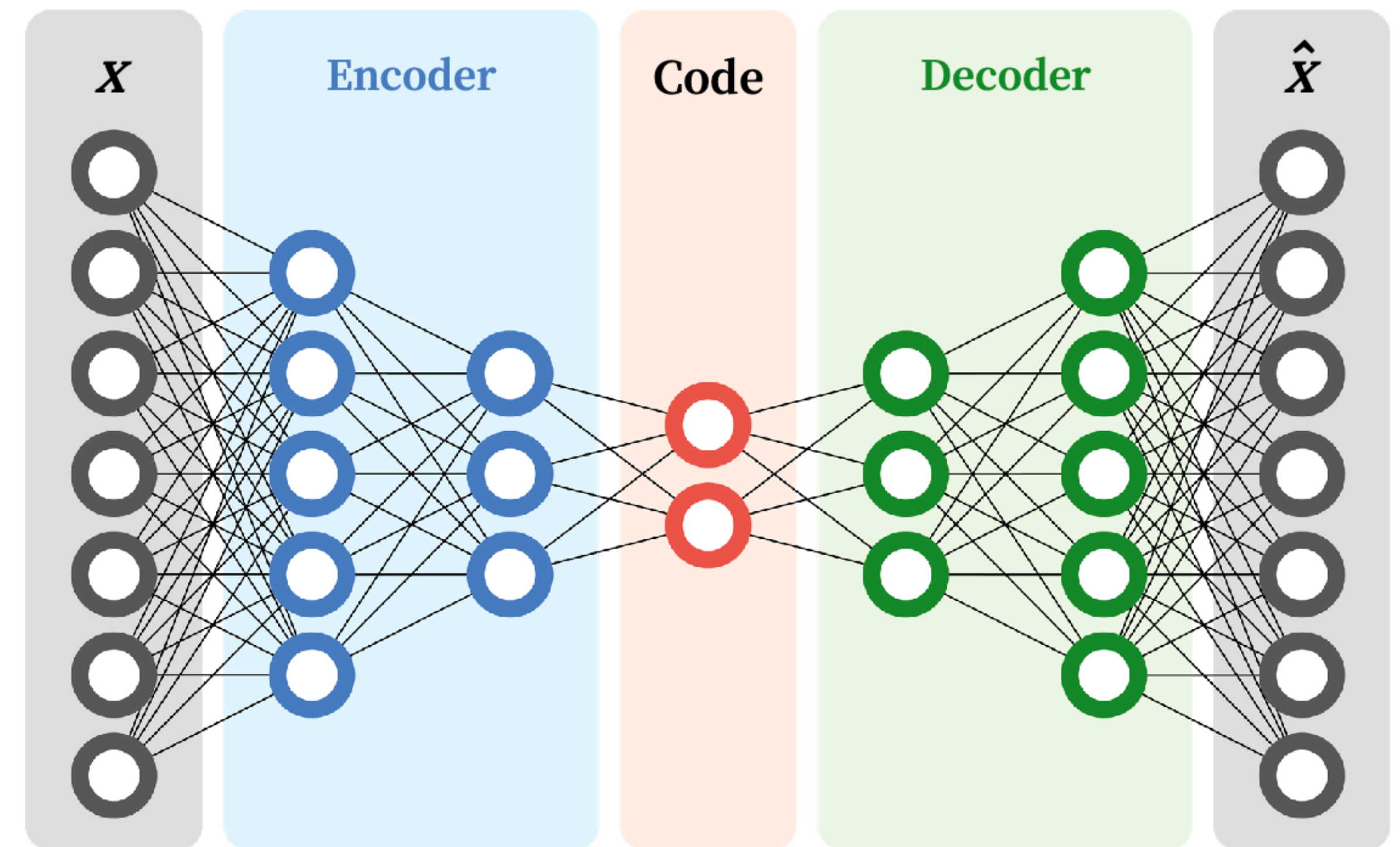


Code

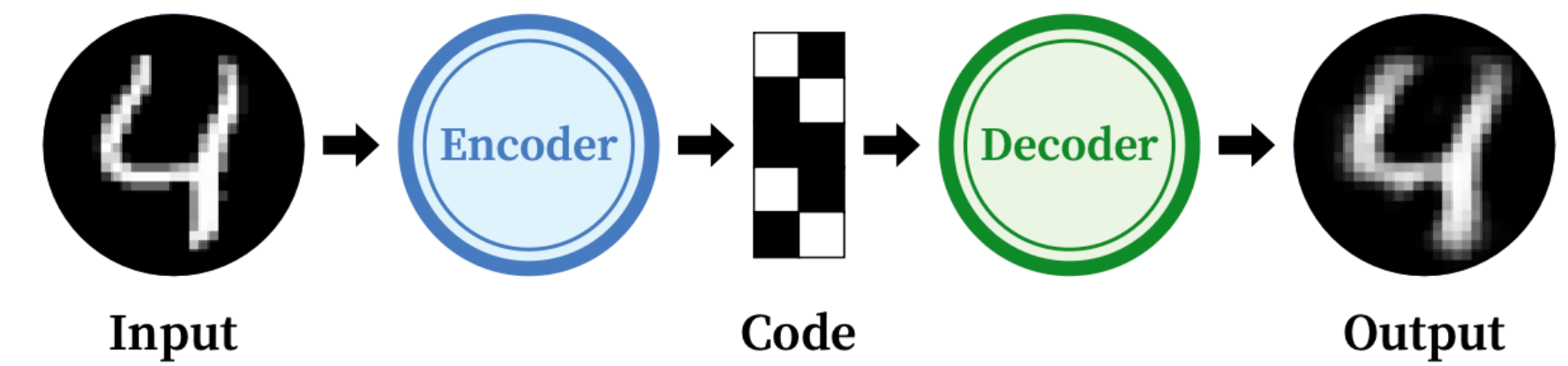


2.5 編碼與解碼：邁向生成的一小步

- Code在不同情境下有時也有其他名稱，如：
 - Compressed Representation（壓縮表示）：強調這個向量是對原始資料進行壓縮後的結果。
 - Z ：數學表示式
 - Bottleneck Layer（瓶頸層）：強調這個向量位於網絡的瓶頸位置，其維度通常比輸入層和輸出層都小

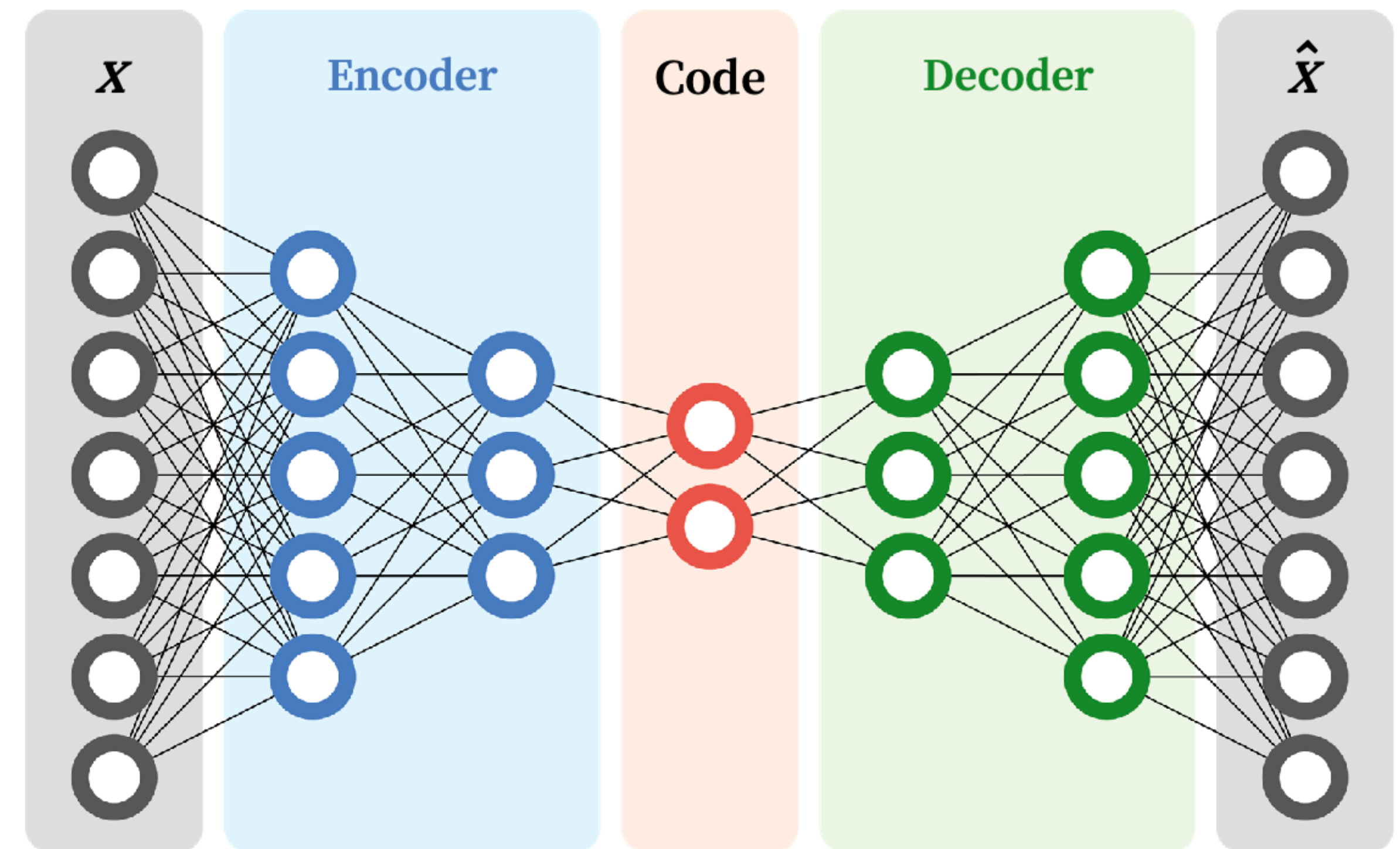


Code



2.5 編碼與解碼：邁向生成的一小步

- 此處的Encoder、Decoder都是神經網路，Encoder中可能包含數層卷積層與池化層，Decoder則可能包括了反卷積層與上採樣層。
- 中間的Latent Layer相對於模型兩端顯得特別窄，相當於整個模型中資料流動的瓶頸(Bottleneck)。
- 反卷積層：類似對卷積做相反的動作，如果我們把卷積視為由資料中擷取出特徵，那麼反卷積可以說是依據特徵重建出儘量符合原始資料的內容。
- 上採樣層：類似於對池化層做相反的動作，池化是對資料進行降維與壓縮；上採樣是將壓縮後的資料透過插值等方法放大回原始尺寸，從而近似還原到資料的原始結構。



如何進行編碼、還原

2.5 編碼與解碼：邁向生成的一小步

- 舉例來說，我們可以將學生的國、英、數三科成績（80、90、85）以如下方式壓縮成二維向量

$$\begin{aligned} z_1 &= 0.5 \times \text{國文} + 0.2 \times \text{數學} + 0.3 \times \text{英文} \\ &= 0.5 \times 80 + 0.2 \times 90 + 0.3 \times 85 = 83.5 \end{aligned}$$

$$\begin{aligned} z_2 &= -0.4 \times \text{國文} + 0.6 \times \text{數學} + 0.1 \times \text{英文} \\ &= -0.4 \times 80 + 0.6 \times 90 + 0.1 \times 85 = 30.5 \end{aligned}$$

- 壓縮後的Code 向量為 $\begin{bmatrix} 83.5 \\ 30.5 \end{bmatrix}$

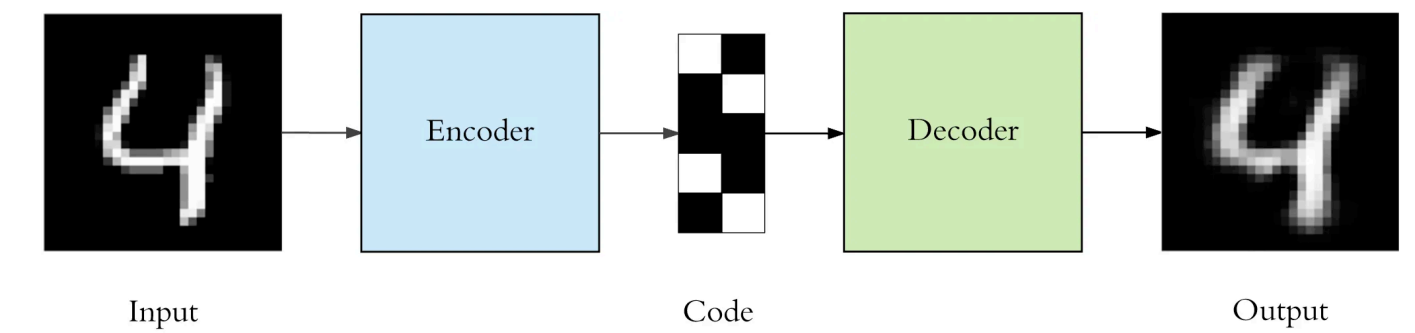
- 根據這組Code，其中一可能的解碼方式：

$$\text{國文} = 0.6 \times z_1 + 0.1 \times z_2 = 53.15$$

$$\text{數學} = 0.2 \times z_1 + 0.7 \times z_2 = 38.05$$

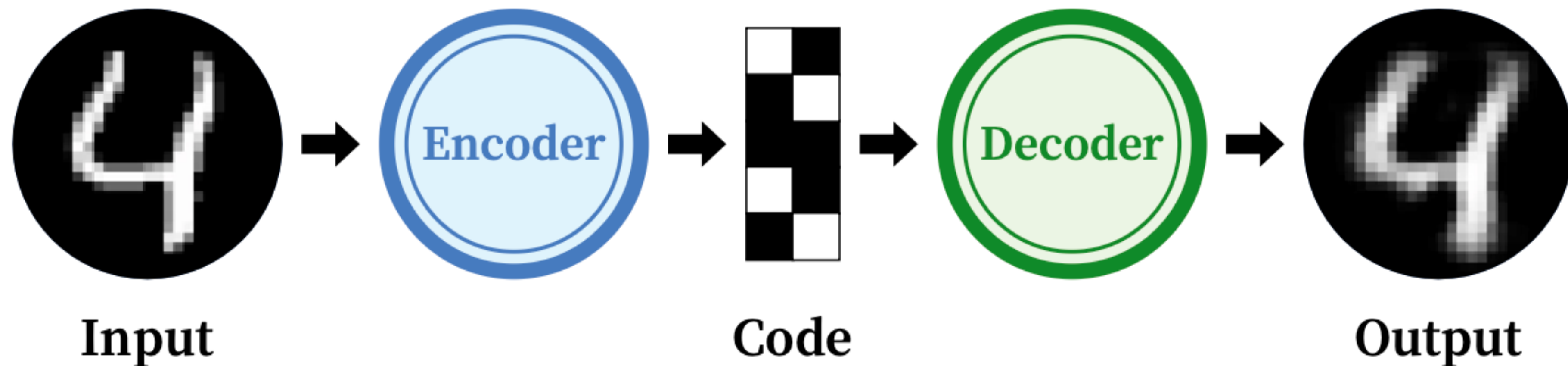
$$\text{英文} = 0.2 \times z_1 + 0.2 \times z_2 = 22.80$$

Why



2.5 編碼與解碼：邁向生成的一小步

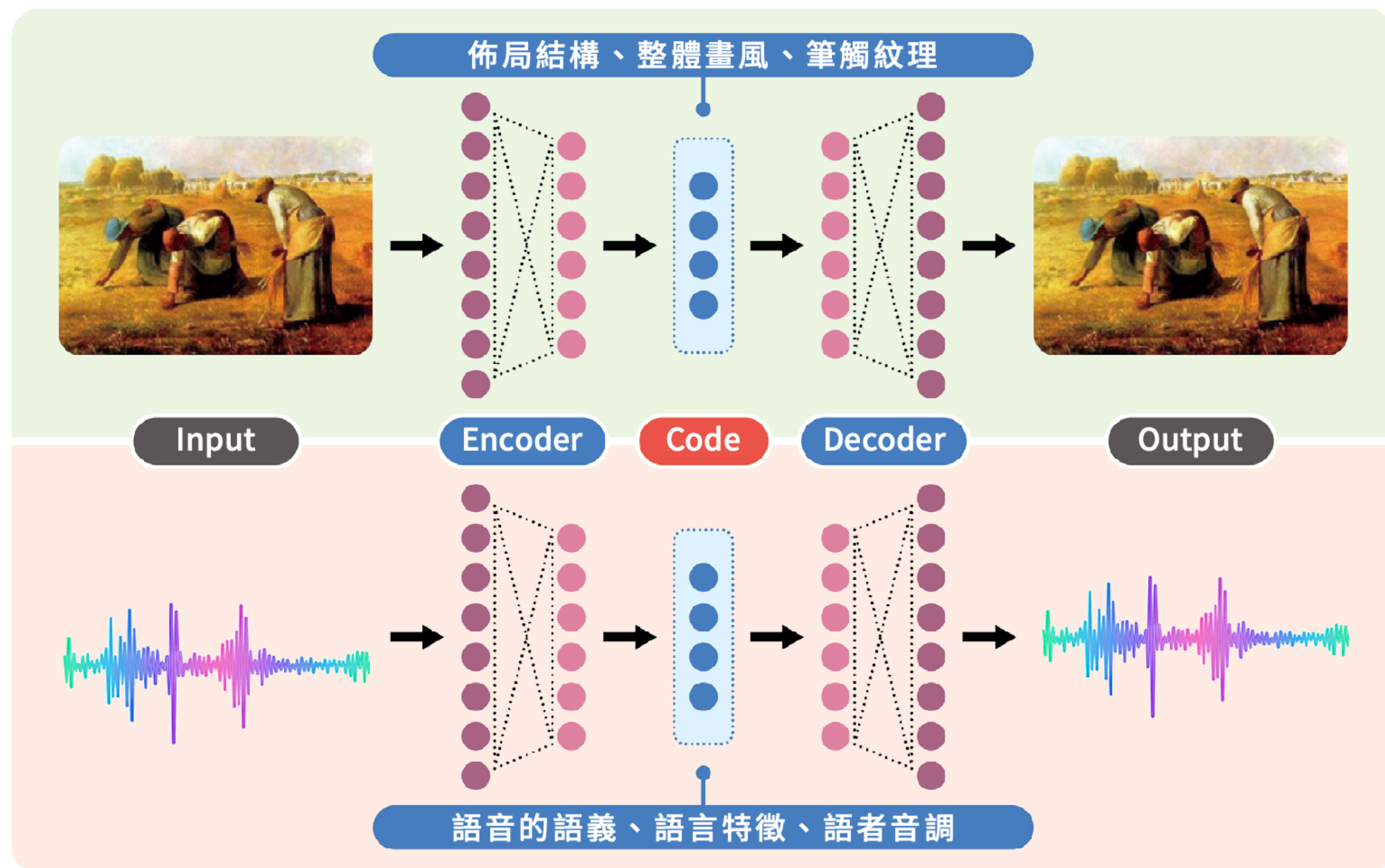
- 這個模型設計如此複雜、中間經歷大量運算，最終其輸出結果竟只是希望能儘量與原資料相符
- 如此大費周張所為何來？



Why

2.5 編碼與解碼：邁向生成的一小步

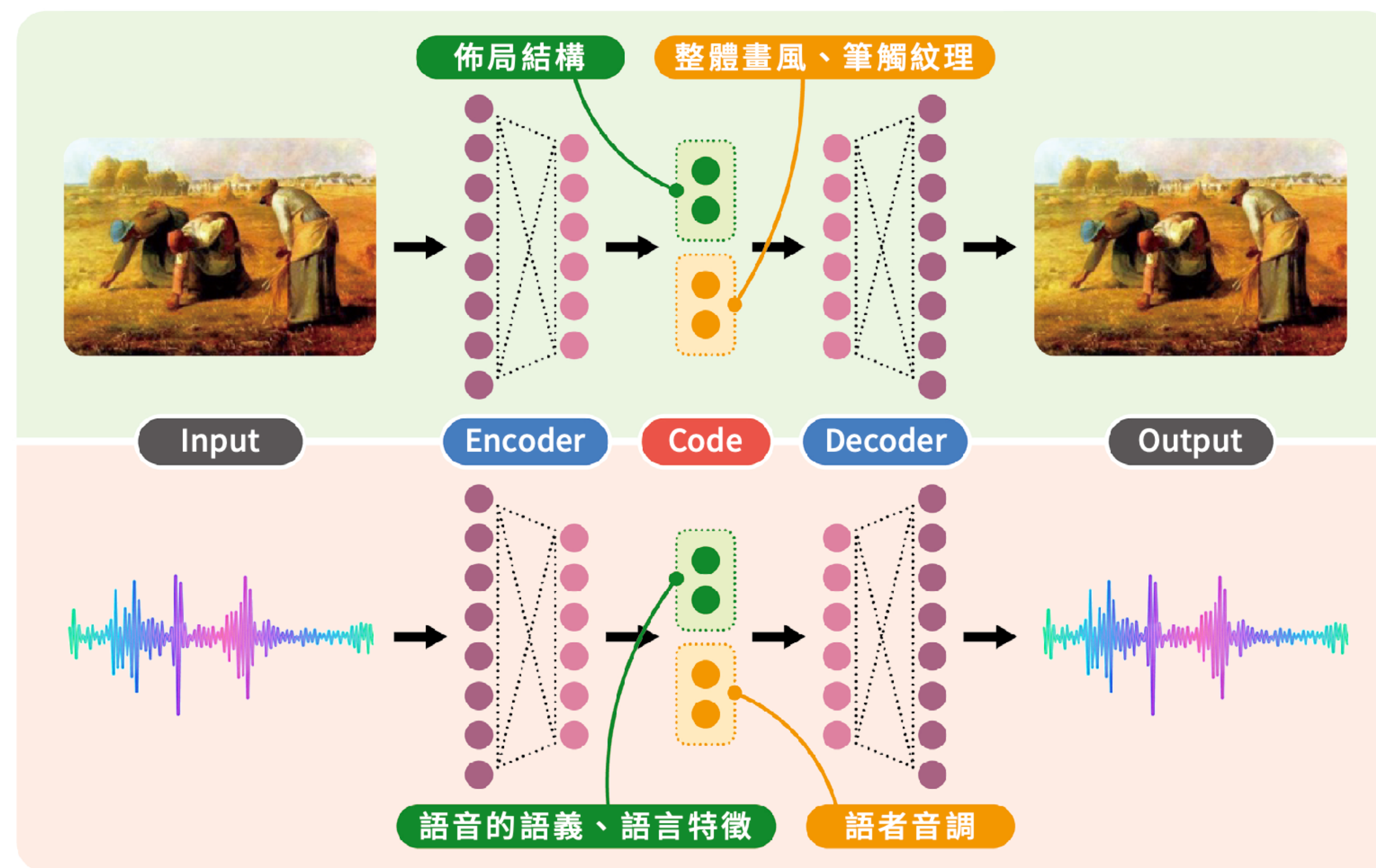
- Decoder 能依據被壓縮降維的 Code 重建資料
- 我們可合理推估 Code 中必然儲存了輸入資料的重要特徵
- 當輸入資料為一幅畫時，Code 中可能儲存了有關畫中佈局結構、整體畫風、筆觸紋理等資訊
- 當輸入資料為一段語音時，Code 中可能儲存了語音的語義、語言特徵、語者音調等資訊。



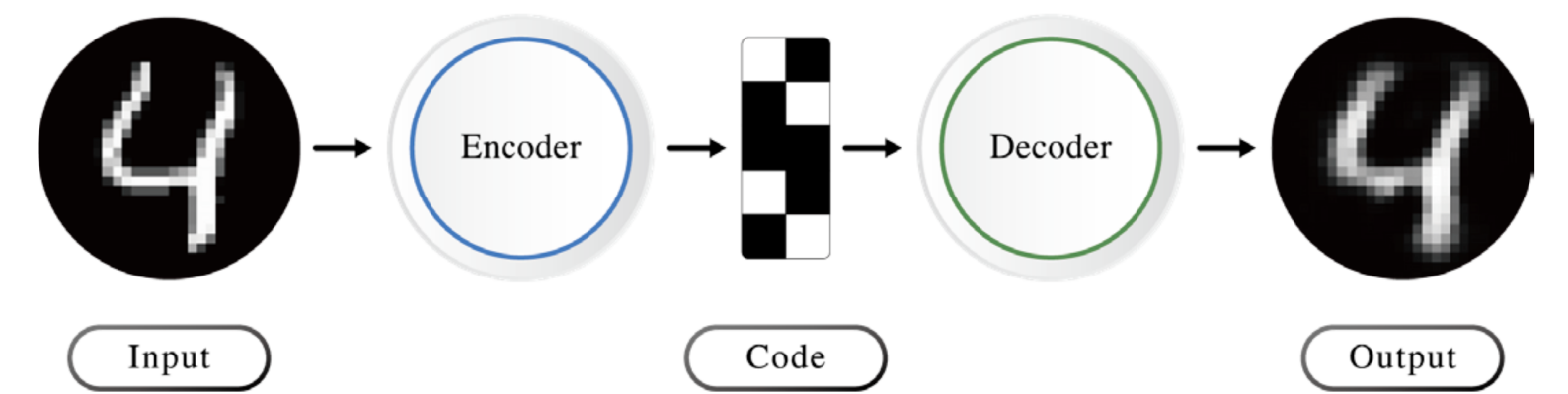
Why

2.5 編碼與解碼：邁向生成的一小步

- 如果我們能將 Code 中儲存的資訊進一步將原本交織在一起的各種特徵分離成以下兩部分：
 1. 該圖的佈局結構（Code 中的綠色部分向量）
 2. 整體畫風、筆觸紋理（Code 中的橙色部分向量）
- 那麼是不是可以藉由修改後者來改變最後生成的原圖畫風呢？



Why



2.5 編碼與解碼：邁向生成的一小步

- 風格轉移示例，藉由修改Code中的畫風特徵，一幅以暖色系呈現「金黃色麥田中辛勤工人」的米勒作品，最終就轉變為一幅以冷色調呈現出眩人光彩的梵谷風畫作。
- 我們也可以利用自動編碼器來對一段語音進行語者轉換（例如以導演吳念真的聲音來當語音導航）、或是對一段文本進行寫作者風格轉換（例如以網路作家九把刀的筆觸來重寫《西遊記》）



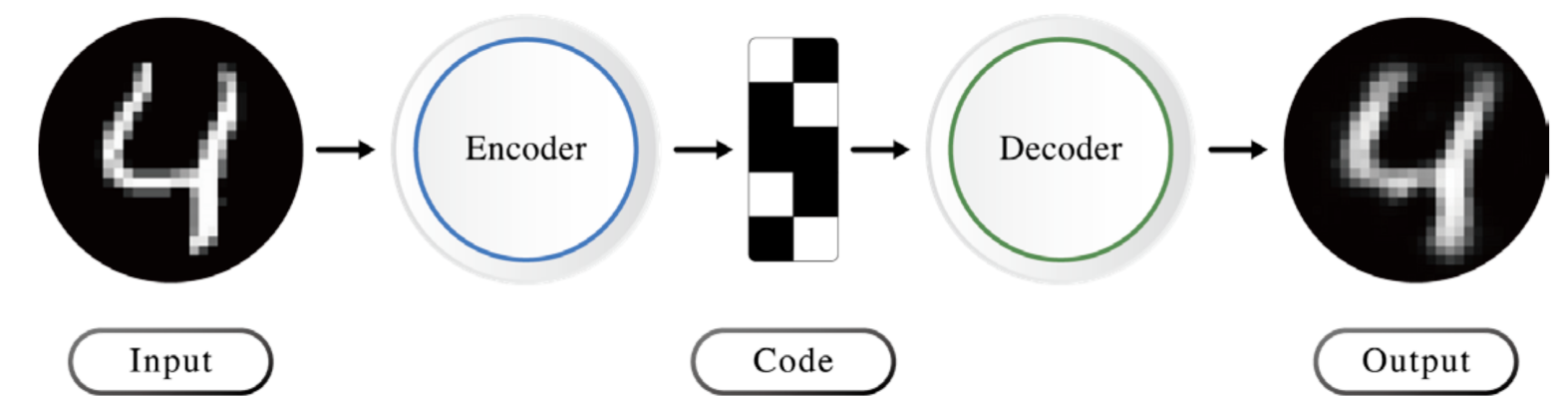
(a) 米勒的拾穗



(b) 梵谷的星空



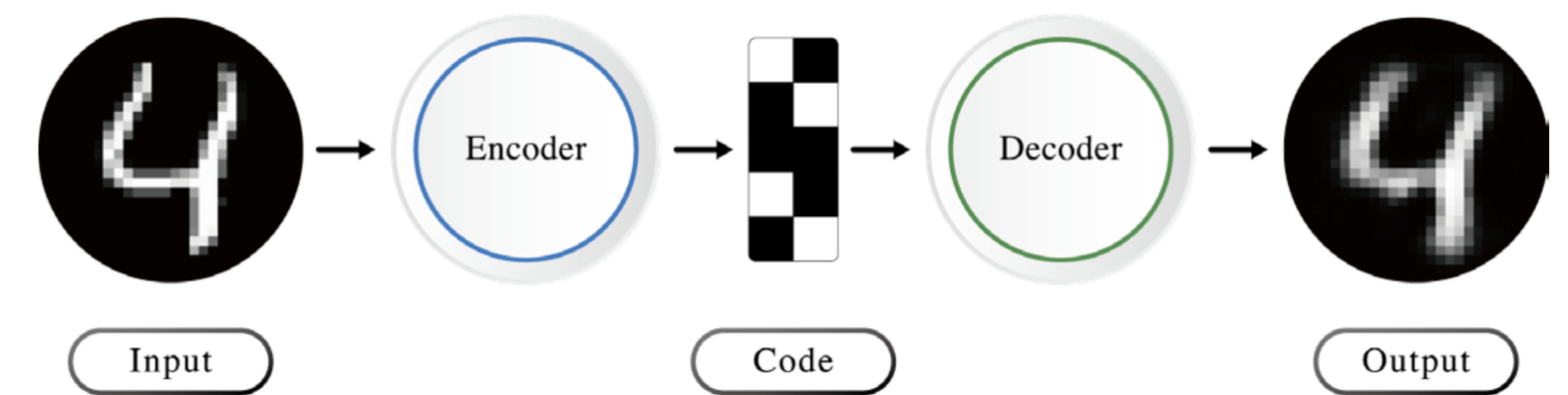
(c) 梵谷畫風的拾穗



2.5 編碼與解碼：邁向生成的一小步

- 自動編碼器的常見應用領域：
 - 降維（Dimensionality Reduction）：將高維度的資料映射到低維空間，從而實現資料降維，對於模型來說，處理被降維的資料會比處理原始輸入資料更節省所需的運算量，自然也能提高處理速度。
 - 降噪（Denoising）：通過訓練自動編碼器對有雜訊的資料（如有雜點的影像、有噪聲的語音）重建原始數據。做法是從輸入的帶噪點資料中提取Code、再依據該Code重建無雜訊的資訊。
 - 異常值檢測（Anomaly Detection）：這種類型的任務似乎透過監督式學習也能達成，但訓練一個用來檢測異常值的監督式模型最大的問題以於異常資料本身就較為稀少（例如信用卡詐欺、網路攻擊行為），當訓練資料集中的「正常」與「異常」資料數量差異過大時，監督式學習模型可能會偏向學習主要的「正常」類別，而忽略稀少的「異常」類別，導致模型在實際應用中無法有效檢測異常值。

Code



2.5 編碼與解碼：邁向生成的一小步

- 自動編碼器的常見應用領域：
 - 內容生成（Content Generation）：如本節中的風格轉換。
 - 影像修復：利用Decoder來重建影像中缺失的部分（神經修復（Neural Inpainting）），這就是如今許多手機中必備的、可以用來無痕移除相片中路人或雜物的P圖神器背後的運作原理。

