
iPAS
經濟部產業人才能力鑑定

中級能力鑑定－學習指引

# AI應用規劃師

## ▶▶▶ 機器學習技術與應用

# 序

為提供授課教師及考生掌握評鑑方向，準備有所依循，本計畫委託委員會題庫組及規劃組領域專家，依據各科目評鑑內容進行重點說明與考題解析。

本手冊為學習指引，旨在提供學習方向與準備參考，並非正式教材或題庫，亦不保證考試通過之責，建議考生依循考試簡章所公告之評鑑主題內容，進行充分準備以應試。

如有相關問題，請逕自聯繫 iPAS@itri.org.tw。

經濟部產業人才能力鑑定推動小組

敬啟

# 目錄

# 職能基準

經濟部為有效提升產業人才素質，近年來持續致力於專業人才培訓發展。為了更明確產業對各類專業人才的能力需求，特別針對亟需人才的多項重點產業，邀集產官學專家，發展產業職能基準，提供各界依其內涵辦理培訓課程及規劃能力鑑定機制。

## 一、何謂職能？

為完成特定職業（或職類）工作任務，所需具備的能力組合（知識、技能、態度）。

## 二、AI 應用規劃師職能基準

| | |
|---|---|
| 職類名稱 | AI 應用規劃師 |
| 工作描述 | 了解 AI 工具的特性及具備使用經驗，以協助企業規劃與推動 AI 技術或工具導入，根據企業部門業務需求，評估並選擇適合的 AI 工具或解決方案，應用於內部流程或產品生命週期。整合跨部門團隊，共同制定與執行 AI 導入計畫，進行開發、部署及後續優化。 |
| 建議擔任此職類之學經歷或能力條件 | （建議具體以下至少 1 項）<br>1. 大專以上畢業或同等學力。<br>2. 具 1 年以上從事演算法設計、人工智慧、機器學習、深度學習、商業智慧等技術應用的工作經驗。<br>3. 具 3 年以上程式開發或專案管理經驗，並曾參與大型專案及具協助專案管理經驗。<br>4. 擔任主管職務 1 年以上。<br>5. 了解 no code/ low code、chatGTP、生成式工具。<br>6. 此項職能基準範圍為跨產業適用。 |
| 基準級別 | 5 |

完整的「AI 應用規劃師」職能基準，
可自右方 QRcode 下載：

# 第一章　考試科目與評鑑內容

| 科目 | 評鑑主題 | 評鑑內容 |
|---|---|---|
| L21<br>人工智慧技術應用與規劃 | L211<br>AI 相關技術應用 | L21101　自然語言處理技術與應用 |
| | | L21102　電腦視覺技術與應用 |
| | | L21103　生成式 AI 技術與應用 |
| | | L21104　多模態人工智慧應用 |
| | L212<br>AI 導入評估規劃 | L21201　AI 導入評估 |
| | | L21202　AI 導入規劃 |
| | | L21203　AI 風險管理 |
| | L213<br>AI 技術應用與系統部署 | L21301　數據準備與模型選擇 |
| | | L21302　AI 技術系統集成與部署 |
| L22<br>大數據處理分析與應用 | L221<br>機率統計基礎 | L22101　敘述性統計與資料摘要技術 |
| | | L22102　機率分佈與資料分佈模型 |
| | | L22103　假設檢定與統計推論 |
| | L222<br>大數據處理技術 | L22201　數據收集與清理 |
| | | L22202　數據儲存與管理 |
| | | L22203　數據處理技術與工具 |
| | L223<br>大數據分析方法與工具 | L22301　統計學在大數據中的應用 |
| | | L22302　常見的大數據分析方法 |
| | | L22303　數據可視化工具 |
| | L224<br>大數據在人工智慧之應用 | L22401　大數據與機器學習 |
| | | L22402　大數據在鑑別式 AI 中的應用 |
| | | L22403　大數據在生成式 AI 中的應用 |
| | | L22404　大數據隱私保護、安全與合規 |
| L23<br>機器學習技術與應用 | L231<br>機器學習基礎數學 | L23101　機率/統計之機器學習基礎應用 |
| | | L23102　線性代數之機器學習基礎應用 |
| | | L23103　數值優化技術與方法 |
| | L232<br>機器學習與深度學習 | L23201　機器學習原理與技術 |
| | | L23202　常見機器學習演算法 |
| | | L23203　深度學習原理與框架 |

| 科目 | 評鑑主題 | 評鑑內容 |
| --- | --- | --- |
| | L233<br>機器學習建模與參數調校 | L23301　數據準備與特徵工程 |
| | | L23302　模型選擇與架構設計 |
| | | L22303　模型訓練、評估與驗證 |
| | | L22304　模型調整與優化 |
| | L234<br>機器學習治理 | L23401　數據隱私、安全與合規 |
| | | L23402　演算法偏見與公平性 |

# 第二章　考科內容

本指引將說明中級「AI 應用規劃師」科目三之考試內容，包含「機器學習技術與應用」之評鑑主題「機器學習基礎數學」、「機器學習與深度學習」、「機器學習建模與參數調校」與「機器學習治理」，協助考生理解機器學習與深度學習理論及基礎數學，熟悉常見模型與應用情境，具備建模與參數調校能力，掌握資料處理、模型訓練與評估流程，並強化對模型治理、風險辨識與公平性等議題的理解，以提升 AI 應用實務與規劃能力。此外，為強化式學習成效，每章節將提供多樣化的練習評量，幫助考生自我測試與檢視學習成果。

# 第三章　機器學習基礎數學

在發展一套穩健且可解釋的機器學習系統前，必須奠基於堅實的數學原理。機器學習的本質，是透過資料觀察中潛藏的模式，建構能夠推論未知情況的模型。而這一過程，無論是資料的表徵、模型的建立、參數的學習、結果的評估與調整，都深深依賴於數學概念的支撐。

本章「機器學習基礎數學」將聚焦於機率統計、線性代數與數值優化三個領域，這三者分別對應機器學習中資料不確定性建模、資料與模型的數值表示、以及參數求解與訓練過程中的計算策略。透過對這些數學基礎的掌握，學習者將能更深入理解模型行為，並具備更高的能力進行模型選擇、調校與分析。本章內容安排如下：

- **機率與統計之機器學習應用：**

  探討資料中的不確定性來源，並介紹如何利用機率分佈、條件機率、假設檢定等統計方法支撐模型學習與推論。

- **線性代數之機器學習應用：**

  說明向量、矩陣等數學結構如何支持資料表示與模型運算，並引導學習者掌握特徵分解與線性轉換等進階技巧。

- **數值優化技術與方法：**

  說明損失函數最小化問題的數學基礎與解法，涵蓋梯度下降、學習率調整、正則化等訓練關鍵技術。

## 重點掃瞄

# 3.1 機率/統計之機器學習基礎應用

### 1. 前言與章節導覽

在真實世界中，資料往往受限於觀察條件、取樣變異或內在隨機性，使得預測結果不具唯一性與確定性。因此，機器學習模型的核心任務，並非僅在於尋找絕對規則，而是要能處理資料中不可避免的「不確定性」，並在此基礎上進行合理的預測與決策。這正是機率與統計在機器學習中扮演關鍵角色的原因。

本節將介紹機率與統計在建構機器學習模型時的實務應用，說明如何運用隨機變數、機率分佈來表示資料行為，進而透過條件機率、貝氏定理等概念進行推論與模型更新。同時，也會探討統計推論（如假設檢定、p 值計算）在模型評估與特徵選擇等任務中的操作方法。

### 2. 資料與隨機變數的機率表示

在建構機器學習模型時，其核心邏輯通常可表述為：「在特定觀察條件下，某結果發生的可能性有多大。」這意味著我們並非尋求唯一解，而是學習一種條件機率分佈（Conditional Probability Distribution）。

其形式可表示為：P（Y｜X），其中 X 為輸入特徵（Feature）、Y 為目標變數（Label）。這樣的機率模型有兩種意涵：

- 預測導向：模型輸出為某結果的機率（如分類機率），而非確定性分類結果。
- 不確定性評估：機率反映了模型對預測的信心程度，有助於風險控制與決策制定。

在以機率方式理解資料時，我們會根據資料型態的不同，區分為離散型與連續型隨機變數，並透過不同的機率分佈來加以建模。這些機率分佈不僅可用來表示資料的行為特性，也可作為機器學習模型假設的基礎架構，影響模型選擇與參數學習方式。根據隨機變數的特性，機率分佈可以分為離散型和連續型兩大類：

### （1）離散型機率分佈（Discrete Probability Distribution）

- 離散型隨機變數：
  - 值是有限或可數的，例如擲骰子的結果（1、2、3、4、5、6）或某電商平台每日訂單數（0、1、2…）。這些變數的值通常為整數，且可能取值集合是明確的。
- 離散型機率分佈：
  - 透過機率質量函數（Probability Mass Function, PMF）描述隨機變數「每一個特定取值的機率」。
  - 例如，擲骰子時，PMF 為 P（X=1）=1/6，P（X=2）=1/6，
  - PMF 總和等於 1：

$$\sum_{x_i} P(X = x_i) = 1$$

- 常見分佈
  - 伯努利分佈（Bernoulli）：用於表示具有兩種可能結果的事件，例如成功與失敗、點擊與否等，常見於二元分類任務。
  - 二項分佈（Binomial）：描述在 n 次獨立試驗中，某事件發生的次數，常用於模擬多次伯努利事件的累計行為。
  - 泊松分佈（Poisson）：描述在固定時間或空間區間中，某事件發生的次數。此分佈常應用於模擬稀有事件，例如單位時間內的客服來電數量、網頁伺服器的請求次數等。泊松分佈假設事件發生彼此獨立，且平均發生率為常數。

### （2）連續型機率分佈（Continuous Probability Distribution）

- 連續型隨機變數：
  - 值是無限且連續的，通常用於描述測量型變數，如身高、體重、時間或溫度。這些變數不僅包含整數，還可以是任意實數，例如某病患等待時間可能為 5.3 分鐘或 5.31 分鐘。

- 連續型機率分佈：
  - 透過機率密度函數（Probability Density Function, PDF）描述隨機變數「於某取值範圍內的機率」。
  - 對於連續型隨機變數 X，PDF 必須滿足以下條件：

$$P(a \le X \le b) = \int_a^b f(x)dx$$

其中，f（x）是隨機變數 X 的機率密度函數（PDF），描述了隨機變數在區間[a, b]內的機率。

  - 常見的分佈
    - 常態分佈（Normal）：對稱的鐘型分佈，是最常見的連續型分佈，廣泛應用於誤差建模、參數估計、特徵分數標準化與生成模型。
    - 均勻分佈（Uniform）：表示在某個固定區間內，所有數值具有相同的發生機率，常用於初始化參數或隨機抽樣。
    - 指數分佈（Exponential）：指數分佈描述的是事件發生之間的間隔時間，適用於隨機過程中等待時間的分佈，通常用來描述「等待時間」或「生存時間」。例如，機器故障時間、電話來電間隔等都可以用指數分佈來建模。該分佈的特徵是無記憶性（Memoryless），即未來的事件發生與過去的時間無關。
    - 卡方分佈（Chi-square）：主要用於描述一組獨立標準常態分佈變數平方和的分佈結果，廣泛應用於統計檢定領域，特別是在變異數分析、卡方適合度檢定、列聯表獨立性檢定等情境中。

在模型設計過程中，選用的機率分佈代表對資料生成機制的先驗假設。舉例來說，邏輯迴歸模型假設目標變數服從伯努利分佈，用以處理二元分類問題；線性迴歸則假設誤差項符合常態分佈，藉此推導參數估計與檢定的統計性質。至於生成模型（例如變分自編碼器，Variational Autoencoder），則更進一步將潛在變數與觀察變數的分佈型態納入模型架構核心，使機率分佈不只是輔助工具，而是模型運作本身的一部分。

## 3. 條件機率與貝氏推論

### （1）條件機率

在機器學習中，除了觀察變數本身的分佈外，同時關注當已知某些條件（例如輸入特徵）時，如何推估另一個變數（如目標標籤）的可能性。這種在給定條件下估算機率的行為，即為條件機率（Conditional Probability）的概念。條件機率不僅是理解機器學習的邏輯核心，也構成了貝氏推論（Bayesian Inference）的基礎架構。

條件機率的數學定義為：

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

表示在事件 B 發生的前提下，事件 A 發生的機率。這種推論形式在分類、推薦、風險預測等領域皆有廣泛應用。以下舉例條件機率在機器學習中的應用場景：

- 分類任務中的條件預測：模型的任務通常是學習條件機率 P（Y｜X），即在觀察輸入特徵 X 的情況下，預測 Y 的可能性分佈。像是邏輯迴歸、貝氏分類器等，皆以此為核心。
- 生醫與金融風控領域的風險預測：當已知某些檢驗結果或行為模式，條件機率可協助預估未來事件發生的可能性，例如罹病風險、違約機率等。
- 生成模型中的變數關聯建構：在變分自編碼器或隱馬可夫模型（Hidden Markov Model, HMM）中，條件機率用於建構潛在變數與觀察變數間的依存關係。

### （2）貝氏定理

機器學習模型的常見目標之一、是推估條件機率 P（Y｜X），也就是在已知某些輸入條件 X 的情況下，預測 Y 的可能性。這種條件推論不僅存在於分類與推薦系統，也廣泛應用於風險預測、醫療診斷與生成模型等場景。隨著模型運作過程中不斷有新資料進入，若能根據這些資料即時調整對事件的預期，就能提升預測品質與模型靈活度。

貝氏定理（Bayes' Theorem）正是處理這類條件推論問題的核心工具。建立在條件機率之上，是一種利用已知條件更新事件發生機率的方法：

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

此公式表示：在事件 B 已發生的情況下，重新評估事件 A 發生機率的方式。其四個構成元素如下：

- P（A）：
  - 先驗機率（Prior Probability）— 在尚未觀察事件 B 之前，對事件 A 發生的「初始信念」或「預設機率」。
- P（B｜A）：
  - 條件機率，或稱似然（Likelihood）— 在事件 A 發生的前提下，事件 B 發生的可能性。
- P（B）：
  - 邊際機率（Marginal Probability）— 事件 B 發生的總體機率，亦為所有可能 A 條件下 B 發生機率的加權總和。（無論 A 是否發生，B 發生的整體可能性。）
- P（A｜B）：
  - 後驗機率（Posterior Probability）— 在觀察到事件 B 之後，根據新的資訊更新後，對事件 A 發生機率的重新估計。
  - 這是貝式定理的核心輸出。

貝氏定理的關鍵意涵，在於可以將「原有知識」與「新資料觀察」整合起來，產生即時的後驗機率調整。這種更新能力，使得機器學習模型在面對不確定性與資料稀疏問題時，能保有推論的彈性與解釋力。

### 4. 假設檢定與統計推論

在機器學習與資料分析中，我們常遇到需要根據有限樣本，推論整體資料或模型是否具備某種統計性質。例如，某個特徵是否與目標變數顯著相關、兩種模

型的表現差異是否具有統計意義等。這些問題皆屬於統計推論（Statistical Inference）的範疇，而假設檢定（Hypothesis Testing）則是其中最常用的工具之一。

### （1）統計推論

統計推論的核心任務，是利用樣本資料對母體參數或模型行為進行估計與判斷，並量化不確定性。透過統計方法，我們可以推斷模型訓練的結果是否穩定、資料特徵之間是否存在顯著差異、以及模型選擇是否具有合理依據。

統計推論可概略分為兩大核心分支：參數估計（Parameter Estimation）與假設檢定（Hypothesis Testing），兩者雖然目的不同，但均依賴機率模型作為推論依據，並對樣本中的不確定性進行量化處理。

| 項目 | 參數估計 | 統計假設檢定 |
|---|---|---|
| 目的 | 推測母體參數的「值」或「區間」 | 驗證某個關於母體參數的「主張」是否成立 |
| 重點問題 | 這個母體參數大約是多少？（例如：平均收入是多少？） | 我們是否有足夠證據拒絕一個假設？（例如：新藥是否有效？） |
| 輸出結果 | 提供點估計值（如平均數）與信賴區間（如 95% CI） | 提供 p 值、檢定統計量，並根據顯著水準決定是否拒絕虛無假設 |
| 依據 | 基於樣本資料，計算出母體參數的估計值 | 基於假設前提與樣本結果，進行推論判斷 |
| 例子 | 根據樣本估計出平均體重為 68 公斤，95%信賴區間為[66, 70] | 假設新運動課程能降低體重，檢定結果 $p = 0.03$，小於設定的 $\alpha$ 值 0.05。因此拒絕虛無零假設，認為有效 |

### （2）假設檢定

假設檢定是一種以機率模型為基礎的推論方法，用於檢視樣本資料是否提供足夠證據來拒絕某一原先的假設。整體流程包含：

A. 設定虛無假設（或稱零假設）與對立假設

B. 選擇適當的檢定方法與檢定統計量

C. 決定顯著水準

D. 計算檢定統計量與 p 值

E. 比較顯著水準（$\alpha$）並進行決策

### （3）顯著水準（α）與 p 值

在進行假設檢定之前，研究者需預先設定一個可接受的錯誤機率上限，稱為顯著水準（α）。顯著水準代表在虛無假設為真的前提下，仍可能因樣本隨機波動而錯誤地拒絕該假設的機率，也就是型一錯誤（Type I Error）發生的機率。

而 p 值則是在觀察到樣本資料後所計算出的機率，用來衡量資料與虛無假設的相符程度。

## 5. 統計量與機器學習中的應用

除了選擇適當的分佈型態，統計量（Statistical Measures）是用以描述資料分佈特性的重要指標，能夠協助分析者快速掌握變數的整體趨勢、變異程度與潛在異常。這些統計量廣泛應用於機器學習各階段，從前期的資料探索與特徵工程，到後期的模型訓練與效能評估，皆是不可或缺的輔助工具。

在資料前處理與探索階段，統計量有助於確認變數的分佈型態與異常狀況，進而決定後續的標準化、轉換或篩選策略。常見應用如下：

- 期望值（Expected Value）：反映變數的平均趨勢，為許多模型的預測基準與參數估計核心，例如線性迴歸中的截距項。
- 變異數（Variance）與標準差（Standard Deviation）：衡量資料的離散程度，能判斷特徵是否需進行標準化處理，避免尺度不一致對模型訓練造成偏誤。
- 偏態（Skewness）：判斷分佈是否對稱，若偏態過大，常需對變數進行對數轉換或 Box-Cox 轉換，以改善模型收斂性與預測穩定性。
- 峰度（Kurtosis）：觀察資料是否具有尖峰或厚尾，亦可作為偵測異常值密度與風險擴散的一項參考指標。

這些統計量常透過視覺化方式（如直方圖、箱型圖、QQ-plot）輔助解釋，幫助分析者理解資料行為模式，並確認是否需採取分群處理、變數轉換或資料清理等動作。

## 重點掃瞄

# 3.2 線性代數之機器學習基礎應用

### 1. 前言與章節導覽

線性代數（Linear Algebra）是機器學習模型運算與表示的數學基礎，其核心概念貫穿於資料結構表示、模型參數計算、梯度更新與特徵轉換等環節。在現代機器學習與深度學習系統中，絕大多數演算法都以矩陣與向量為運算單位，並透過線性變換、特徵分解與最小平方估計等工具來實現模型訓練與預測。

本節從向量與矩陣表示、線性變換與特徵空間開始，到矩陣分解與維度簡化與最小平方估計與線性迴歸，介紹基礎概念背後的幾何意義與演算法對應，並建立其與實際建模流程的連結。

### 2. 向量與矩陣表示

在機器學習中，幾乎所有資料與模型參數都可以向量（Vector）與矩陣（Matrix）的形式來表示與運算。向量與矩陣不僅是資料的儲存結構，更是模型計算與訓練流程中的基本單位，包含特徵表達、線性組合、梯度運算等均仰賴這些基礎工具。

#### （1）向量在機器學習中的角色

向量是具有方向與大小的數學物件，通常用於描述單一樣本的特徵組合。例如：

一筆 5 維的樣本輸入可以表示為向量 $\mathrm{x} = [x_1, x_2, x_3, x_4, x_5]^{\mathrm{T}}$

模型的參數向量 $\theta = [\theta_1, \theta_2, \ldots, \theta_d]$ 可用於計算預測值 $\hat{y} = \theta^{\mathrm{T}}\mathrm{x}$

常見向量運算包括：

- 點積（Dot Product）：
  - 評估兩個向量在同一方向上的對應程度，為線性模型預測核心運算。
  - 其物理意義是「投影」和「相似度」。

- L2 範數（Norm）：
  - 或稱歐幾里得範數。
  - 用於計算向量的「長度」或「大小」，亦為正規化與正則化（如 L2 損失）的基礎。
- 向量加減與線性組合：
  - 可用於計算誤差向量、梯度向量等。

**（2）矩陣在機器學習中的應用**

矩陣是多個向量的集合，常用於表示多筆樣本資料、特徵轉換或神經網路中的權重。舉例如：

- 特徵矩陣 $X \in \mathbb{R}^{n \times d}$ ：
  n 筆樣本、每筆含 d 個特徵，每一行為一筆樣本向量。
- 權重矩陣 $W \in \mathbb{R}^{d \times k}$ ：
  用於多類別分類中，將 d 維輸入特徵映射為 k 維輸出機率分數。

常見矩陣運算包含：

- 矩陣乘法（Matrix Multiplication）：模型運算的核心，用於批次預測、權重更新、轉換特徵空間。
- 轉置（Transpose）：將矩陣的列與行互換，用於維度對齊與內積計算。
- 矩陣求逆（Inverse）與偽逆（Pseudo-Inverse）：用於封閉解的求解（如最小平方解），或在無法反矩陣的情況下近似解決。

在模型建構中的具體應用示例：

- 線性迴歸中，預測值可由 $\hat{y} = X\theta$ 表示，並以矩陣形式進行損失函數與導數運算。
- 神經網路的前向傳播中，層與層之間的計算本質為矩陣與向量的乘法：$z^{(l)} = W^{(l)}a^{(l-1)} + b^{(l)}$。

- 主成分分析（Principal Components Analysis, PCA）需對特徵矩陣進行協方差計算與矩陣分解，以尋找最具代表性的投影方向。

## 3. 線性變換與特徵空間

線性變換（Linear Transformation）是線性代數中的核心概念，其本質是在不破壞空間線性結構的前提下，對資料進行伸縮、旋轉或投影等操作。機器學習模型中大量的資料處理與特徵映射，其實都可視為一種線性變換，尤其在神經網路、特徵工程與降維方法中扮演關鍵角色。

### （1）向量經過矩陣運算的幾何意涵

當一個向量 $x \in \mathbb{R}^d$ 被一個矩陣 $A \in \mathbb{R}^{k \times d}$ 左乘時，所得到的新向量 $Ax \in \mathbb{R}^k$ 可視為對原始向量的一次線性變換。這個變換可能發生在同一維度空間中，也可能將向量投射至另一個維度的空間中，其幾何意義包含：

- 縮放（Scaling）：
  - 調整向量在各個方向上的長度，改變其尺度但不改變方向。
- 旋轉（Rotation）：
  - 改變向量的方向而不改變其長度，常見於正交變換或特徵對齊。
- 剪切（Shearing）：
  - 使向量方向在空間中產生傾斜變化，常出現在非對角矩陣的變換中。
- 投影（Projection）：
  - 將高維向量投射到某個子空間（如主成分空間或分類超平面），保留對任務最有意義的資訊。

這些操作可以理解為對原始特徵空間的「重構」或「重新編碼」，其目的在於讓資料在轉換後的空間中更利於模型處理。例如，透過適當的線性變換，可以強化資料的分群結構、降低維度冗餘，或提高對特定方向的敏感度。

### （2）線性變換與特徵空間重構

特徵空間（Feature Space）是指資料中各個特徵所張成的數學空間，其中每一個軸代表一個特徵維度，每一筆資料可視為空間中的一個點。這個空間的幾何結構不僅描述了資料的分佈狀態，也影響了模型如何進行分類、迴歸或聚類等任務。

透過線性變換，我們可以達到：

- 特徵重組：
  - 將原始特徵做線性組合，產生新的表示（如主成分分析）。
- 維度轉換：
  - 將資料從原始高維空間轉換至低維或嵌入空間（如投影到主成分空間或隱藏層）。
- 方向加權：
  - 強化模型對於某些方向（變數組合）的敏感性。

舉例，在主成分分析（PCA）中，我們即是透過找出一組能最大化資料變異量的正交向量基底，將原始資料透過矩陣乘法映射到這組基底所定義的空間中，達到降維與特徵重組的目的。

### （3）線性變換在機器學習模型中的出現形式

- 線性迴歸與邏輯迴歸：
  - $y = \mathrm{w}^{\mathrm{T}}\mathrm{x} + b$ 本質為一維線性投影，將多維特徵向量投射到一條直線上以進行預測。
- 神經網路中的前向傳播：
  - 每一層的運算如 $z^{(l)} = \mathrm{W}^{(l)}\mathrm{a}^{(l-1)} + \mathrm{b}^{(l)}$ 可視為將上一層輸出透過線性變換映射至下一層特徵空間，再經過非線性激活。
- 嵌入層（Embedding Layer）：
  - 將離散類別轉為連續空間的向量表示，其核心操作也是一組特定矩陣的線性查詢與轉換。

- 特徵投影與空間壓縮：
  - 如 LDA（線性判別分析, Linear Discriminant Analysis）、Autoencoder 等，皆仰賴線性變換將高維資料重構為低維潛在向量空間。

## 4. 矩陣分解與維度簡化

在高維資料分析中，資料的維度（特徵數量）往往遠高於模型所需，有時甚至導致過擬合、計算效率低下或資訊冗餘。矩陣分解（Matrix Factorization）是一種有效的數值工具，可將原始矩陣拆解為多個較小且具有結構意義的子矩陣，進而實現資料降維、壓縮與轉換的目的。

### （1）矩陣分解的核心概念

矩陣分解是指將一個高維矩陣 $X \in \mathbb{R}^{m \times n}$ 拆解為數個較小矩陣的乘積，這些子矩陣在運算上更具可解性，或在幾何上具有特定意涵。分解後的矩陣可視為資料的潛在結構（如主成分、潛在特徵）之表現，有助於後續的建模與解釋。

### （2）常見的矩陣分解方法與應用

- 特徵值分解（Eigenvalue Decomposition）
  - 原理：
    - 適用於對稱方陣，將矩陣分解為一組特徵向量與特徵值的組合形式。
    - 將矩陣 $A \in \mathbb{R}^{n \times n}$ 分解為特徵向量與特徵值的組合形式：

      $A = Q \Lambda Q^{\mathrm{T}}$

      其中：
      - $Q$ 是正交矩陣，由 $A$ 的特徵向量（Eigenvectors）構成。
      - $\Lambda$ 是對角矩陣，對角元素為特徵值（Eigenvalues）。
      - $Q^{\mathrm{T}}$ 表示矩陣 $Q$ 的轉置。簡單來說，就是將 $Q$ 的行變成列，列變成行。

- 幾何意義：
  - 特徵值分解找出一組能穩定表示資料在空間中「拉伸方向」的基底，並量化每個方向的重要性。
- 應用場景：
  - 主成分分析（PCA）：將資料投影到最大變異方向上，達到降維與資訊保留的平衡。
  - 線性判別分析（LDA）：用於找出最佳分類投影方向，以最大化類別間差異與最小化類別內變異。

- 奇異值分解（Singular Value Decomposition, SVD）
  - 原理：
    - 奇異值分解是一種可應用於任意實數矩陣（不需為方陣）的分解方法，將矩陣 $X \in \mathbb{R}^{m \times n}$ 拆解為三個部分：
      $X = U\Sigma V^{\intercal}$，其中：
      - $U \in \mathbb{R}^{m \times m}$：左奇異向量矩陣（對應樣本方向）
      - $\Sigma \in \mathbb{R}^{m \times n}$：奇異值對角矩陣（對角線為非負實數，表示各主方向的重要性）
      - $V \in \mathbb{R}^{n \times n}$：右奇異向量矩陣（對應特徵方向）
  - 幾何意義：
    - SVD 將原始矩陣轉換為不同空間基底的縮放與旋轉操作，具有極佳的數值穩定性與資訊解構能力。
  - 應用場景：
    - 資料降維：保留前 k 個奇異值與對應向量，近似原始資料（用於 PCA 計算）。
    - 推薦系統：分解使用者 - 項目矩陣，找出潛在偏好向量。
    - 潛在語意分析（Latent Semantic Analysis, LSA）：抽取語料中詞與文件間的潛在語意結構。
    - 影像壓縮：只保留主成分影像資訊，降低儲存與運算成本。

- 非負矩陣分解（Non-negative Matrix Factorization, NMF）
  - 原理：
    - 將非負矩陣 $X \in \mathbb{R}^{m \times n}$，$X \geq 0$
    - 分解為兩個非負矩陣乘積的技術：$X \approx WH$
    - 其中：
      - $W \in \mathbb{R}^{m \times k}$，$W \geq 0$：表示基底矩陣（可視為潛在特徵）
      - $H \in \mathbb{R}^{k \times n}$，$H \geq 0$：表示各基底的組合係數
  - 幾何意義：
    - NMF 將資料視為幾個「可加疊的部件」，提供具備語意解釋力的解構方式，並能自然引入稀疏性。
  - 應用場景：
    - 主題建模（Topic Modeling）：將文件–詞矩陣分解為主題與詞彙分佈
    - 生物訊號分析：如腦波分解、基因表現訊號擷取
    - 影像分析：將影像資料拆解為基本視覺元素
    - 社群分析：萃取潛在社群結構或互動關聯性

**（3）維度簡化與學習效率的關聯**

在機器學習中，資料常包含數十甚至數百個特徵，但實際上並非每個特徵都對模型預測有貢獻。若將所有特徵無差別地納入建模，不僅會造成運算成本上升，也容易導致模型過擬合，進而影響預測穩定性與泛化能力。

透過矩陣分解等技術進行「維度簡化」，可有效將資料壓縮為一組更有代表性的特徵組合。這些組合捕捉了資料的主要變異方向，同時排除了雜訊與重複資訊，有助於：

- 提升訓練效率：
  - 減少模型參數量與計算資源需求，加快訓練時間，特別適用於大型資料集或深度學習模型。

- 穩定模型表現：
  - 去除雜訊與共線性問題，有助於降低過擬合風險，提升預測準確度。
- 強化資料解釋性：
  - 轉換後的特徵常具有明確的幾何或語意意義，更容易與業務需求連結，輔助模型診斷與結果溝通。
- 利於視覺化與後續分析：
  - 在維度降低後，可將資料投影至二維或三維空間中，方便進行資料探索、群集判斷與異常偵測等任務。

## 5. 最小平方估計與線性迴歸

線性迴歸（Linear Regression）是機器學習中最基礎且最具代表性的監督式學習模型之一，其核心目的在於找出一條「最佳擬合線」，用以描述輸入變數與目標變數之間的線性關係。這條擬合線的建立，即是透過一種稱為「最小平方估計」的方法所完成。

### （1）最小平方估計的核心概念

最小平方估計（Ordinary Least Squares, OLS）是一種以「誤差最小化」為目標的參數估計方法。在進行模型訓練時，會比較模型所預測的值與實際觀測值之間的差異，並試圖找出一組參數，使這些差異的平方總和達到最小。這樣的方式不僅能提供穩定且具代表性的模型，也具備清楚的幾何與統計意義。

### （2）幾何觀點下的線性迴歸

從幾何角度來看，線性迴歸的本質是一種投影：我們將輸入資料在特徵空間中投影到一個最接近實際結果的平面上。這個平面，就是模型所學習到的線性關係。透過這樣的視角，我們可以理解為何線性迴歸如此直觀，同時又能提供具體的數學保證。

這種幾何結構也說明了為何資料的排列與變異會影響模型的準確性—資料若分佈過於分散或存在離群點，擬合出的平面可能會受到扭曲。

### （3）應用情境與特點

線性迴歸雖然簡單，但其應用場景廣泛，常見情境如：

- 銷售預測：根據廣告支出或市場活動，預測未來營收。
- 醫療風險評估：用年齡、血壓等指標預測患病機率或醫療成本。
- 房價估值：將房屋大小、樓層、地點等作為輸入，預測合理價格。
- 行為建模：描述某一變數如何受多個條件共同影響。

## 重點掃瞄

# 3.3 數值優化技術與方法

### 1. 前言與章節導覽

在機器學習的建模過程中，「訓練模型」本質上就是一個數值優化問題（Numerical Optimization Problem）。不論是調整線性模型的權重參數、深度神經網路的數千萬個連接係數，或是在強化式學習中尋找最適策略，其核心邏輯皆是：找出一組能讓目標函數（例如損失函數）達到最小或最大值的參數組合。

因此，數值優化技術不僅是模型求解的手段，更深刻地影響模型的學習效率、穩定性與泛化能力。選擇合適的優化方法、理解其收斂行為與限制，是機器學習實務中不可或缺的一環。

### 2. 最佳化問題的基本結構

在機器學習中，模型訓練可視為一個「最佳化問題」：我們希望找出一組模型參數，使得模型在訓練資料上的表現最符合預期。這通常是透過最小化某個損失函數（Loss Function）來實現的，也就是找到讓誤差最小的參數組合。

為了理解這個過程，首先需掌握最佳化問題的基本構成要素：

#### （1）目標函數

目標函數（Objective Function）是機器學習中訓練流程的核心，用來衡量模型輸出與實際答案之間的偏差程度，也稱為「損失函數」或「成本函數」。透過最小化（或最大化）這個函數，我們能讓模型持續修正參數，朝向預測更準確的方向前進。

目標函數的形式會根據任務類型而有所不同：

- 迴歸任務：
  - 如使用「均方誤差」（Mean Squared Error, MSE）作為目標函數，藉由懲罰預測值與實際值的平方差，讓模型學會輸出更接近真實的連續數值。
- 分類任務：
  - 如採用「交叉熵損失」（Cross-Entropy Loss），透過衡量預測機率分佈與實際標籤分佈之間的差距，引導模型提升判斷不同類別的信心與準確率。
- 排序與排名任務：
  - 可使用對比損失（Contrastive Loss）或排序損失（Ranking Loss），以學習資料之間相對次序的準確性。

選擇適當的目標函數，明確設定模型方向、評估「什麼是好的預測或產出表現」，對模型訓練方向與效果具決定性影響。

**（2）決策變數**

決策變數（Decision Variables）是模型中可調整的數值參數，亦即學習過程中需要被「優化」的對象。在不同模型中，這些變數的形式可能不同：

- 線性模型：包括權重係數（如迴歸係數）與偏差項。
- 神經網路：包含每一層神經元之間的權重與偏差數值，可能達數萬至數千萬個參數。
- 機率模型：如貝氏模型中的條件機率表、生成模型中的潛在變數。

這些變數的取值將決定模型對輸入資料的反應方式，優化過程的本質，就是持續調整這些參數，讓整體預測更符合學習目標。

### （3）可行域

可行域（Feasible Region）也稱為參數空間，可行域定義了決策變數的合法範圍，也就是「哪些解是允許的」。在某些最佳化問題中，我們可能會對變數施加特定條件，這些限制即構成了參數的可行域。常見情境包括：

- 非負條件：
  - 如非負矩陣分解（NMF）中，所有參數須為正數。
- 上限／下限限制：
  - 防止模型權重過大或過小，穩定訓練行為。
- 總和約束：
  - 某些模型中參數總和需為 1，例如機率分佈。
- 稀疏性限制：
  - 透過限制多數參數為 0（如 L1 正則化），促進模型簡化與可解釋性。

明確定義可行域，有助於排除無效或不可解的解答範圍，使訓練更穩定，也利於後續的規則化控制。

### （4）函數性質：凸性與可導性

一個最佳化問題的難易度，往往取決於其目標函數的數學性質，其中凸性（Convexity）與可導性（Differentiability）是兩個最關鍵的指標：

- 凸性：
  - 如果一個目標函數是凸函數，那麼從任一初始點開始，只要持續往下降的方向走，最終一定能找到全域最佳解。
  - 這讓凸問題具有可預期、穩定的求解特性。像線性迴歸、邏輯迴歸等皆屬於凸問題。
- 可導性：
  - 若函數能夠進行微分，便可透過計算「梯度」來獲得下降方向。這是大多數優化器（如梯度下降法）能正常運作的前提。

- 若函數在某些區段不可導，則可能造成訓練不穩或收斂困難。

- 非凸問題：
  - 如神經網路中的損失函數，常存在多個局部最小值與鞍點。
  - 這使得優化過程充滿不確定性，但若使用適當初始化、動量機制與調整策略，依然能取得效果良好的解。

**（5）機器學習脈絡中的應用**

在實務中，機器學習的訓練流程幾乎都可形式化為最佳化問題，根據模型結構與資料性質的不同，可大致區分為：

- 線性模型訓練：
  - 問題結構簡單、可解析求解，訓練速度快且具有理論保證。
- 深度學習模型：
  - 屬於大規模非凸問題，需依賴數值演算法進行逼近式學習，常見工具包括 SGD、Adam、RMSprop 等。
- 生成模型與策略學習：
  - 如生成對抗網路（GAN）、強化式學習等，最佳化目標可能涉及對抗損失、期望值最大化等複雜結構，需搭配啟發式搜尋或抽樣估計等技術。

## 3. 損失函數與學習目標

在機器學習中，模型訓練的核心任務是「學習一組參數，使預測結果最符合實際資料」。這個過程仰賴損失函數（Loss Function）的設計與計算。損失函數是連結資料、模型與學習目標之間的橋梁，提供一個可度量的依據，讓演算法知道「預測得好不好」，並根據這個評價反覆修正參數。

**（1）損失函數的設計意義**

損失函數不僅是誤差的量化工具，更深層地體現了學習目標的策略偏好與風險容忍度。其設計決定了模型在學習過程中：

- 如何看待不同型態的錯誤，例如假陽性與假陰性的權重差異。
- 哪些誤差應被放大懲罰，哪些則可容忍。
- 參數調整的方向與幅度，進而影響整體的收斂行為與學習效率。

選擇合適的損失函數，不僅關係模型性能，更決定模型是否能有效理解任務本質。

**（2）常見任務與損失函數的對應關係**

根據任務性質，損失函數設計可大致歸類如下：

- 迴歸任務（預測連續數值）
  - 均方誤差（MSE）：放大較大誤差的懲罰，適合誤差分佈穩定的情況。
  - 平均絕對誤差（MAE）：對極端值較不敏感，適用於含有異常值的資料。
  - Huber 損失：結合 MSE 與 MAE 優點，在穩定性與抗雜訊之間取得平衡。
- 分類任務（預測類別標籤）
  - 交叉熵損失：衡量預測機率與實際標籤的距離，為多數分類模型的標準選擇。
  - 對比損失：學習樣本對之間的相似度關係，常用於人臉辨識、語意匹配等。
  - Focal Loss：強化對難分類樣本的學習，特別適合處理資料不平衡問題。
- 排序與重建任務
  - 排序損失：關注資料間的相對順序，常見於搜尋引擎與推薦系統。
  - 重建損失：計算輸入與輸出間的相似程度，廣泛應用於自編碼器與生成模型。

**（3）損失函數對學習行為的影響**

損失函數的選擇會直接影響模型的學習軌跡與結果品質，示例在迴歸任務中：

- 使用 MSE，模型會試圖壓低大誤差，有時會過度受到極端值影響。
- 使用 MAE，模型對所有樣本誤差給予均等權重，較穩健但學習速度可能較慢。
- 不當選擇損失函數（例如分類問題使用迴歸損失），將導致模型訓練無效，甚至完全無法收斂。

損失函數不只是效能評估的指標，更是學習過程的「導航器」，引導模型在複雜問題空間中朝正確方向學習。

## 4. 常見優化演算法與比較

在機器學習中，模型學習的過程就是一種優化問題：透過不斷調整參數，使損失函數的值最小化。為了實現這個目標，我們需要一套能有效引導參數更新的「優化演算法（Optimization Algorithm）」。這些演算法負責判斷每次應該往哪個方向移動、該移動多遠，以逐步接近最佳解。

不同的優化演算法在更新方式、計算效率、收斂行為上各有特色，選擇合適的演算法往往對訓練效率與結果表現有決定性影響。

### （1）基礎方法：梯度下降及其變形

此類方法聚焦於「基本梯度計算與參數更新流程」，透過梯度資訊找到一個能使損失函數下降的方向與步長，是機器學習中早期核心的優化技術。常見方法如下：

- 梯度下降法（Gradient Descent, GD）
  - 概念：
    - 使用整個訓練資料集計算損失函數的梯度，沿梯度方向更新參數。
  - 特點：
    - 更新穩定、能準確反映全體資料的平均方向，但計算成本高、訓練速度慢。

- 適用情境：
  - 小型資料集、高精度需求、可並行化計算的環境。

- 隨機梯度下降（Stochastic Gradient Descent, SGD）
  - 概念：
    - 每次使用一筆樣本來估算梯度並更新參數。
  - 特點：
    - 更新速度快、記憶體需求低，但梯度波動大、收斂不穩定。
  - 適用情境：
    - 大型資料集、線上學習與即時訓練場景。

- 小批次梯度下降（Mini-batch SGD）
  - 概念：
    - 將資料分成小批，每次用一批資料計算梯度。
  - 特點：
    - 在更新穩定性與效率之間取得平衡，是深度學習中最常見的選擇。
  - 適用情境：
    - 中大型模型訓練，能與 GPU 加速高度結合。

**（2）進階方法：學習率調整與收斂加速技巧**

此類方法在基本梯度更新機制上進一步引入動態調整策略與收斂加速技巧，進一步改善學習效率與穩定性，使模型更快、更穩地收斂。主要目的是解決以下實務挑戰：

- 梯度方向不穩定（如震盪或鞍點）
- 參數更新速度不一致（某些參數變動大、某些幾乎不變）
- 固定學習率難以應對不同訓練階段的需求

常見方法如下：

- 動量法（Momentum）
  - 概念：

- 模仿物理動量，將前幾次梯度的方向累積，幫助模型克服局部震盪。

◆ 優點：

- 能加速收斂並穩定學習過程，特別適合高曲率或非平滑空間。

◆ 應用場景：

- 深層神經網路訓練、收斂速度要求高的任務。

• Adagrad

◆ 概念：

- 根據每個參數的歷史梯度大小，自動調整學習率。

◆ 優點：

- 適合處理稀疏特徵或不均衡參數更新問題。

◆ 限制：

- 學習率會隨時間過度衰減，可能導致收斂停止。

• RMSprop

◆ 概念：

- 引入滑動平均，修正 Adagrad 學習率過快下降的問題。

◆ 優點：

- 能穩定訓練過程，特別適合處理非穩定梯度（如 RNN 訓練）。

◆ 應用場景：

- 語音處理、序列建模等非凸問題。

• Adam（Adaptive Moment Estimation）

◆ 概念：

- 結合動量與 RMSprop，追蹤梯度的一階與二階動量，自動調整各參數的學習率。

◆ 優點：

- 訓練快速、收斂穩定、可廣泛應用於大多數模型。

- 應用情境：
  - 目前最常用的深度學習優化器之一，適用於圖像、語言、強化式學習等各類任務。

## 5. 收斂判準與訓練穩定性

優化演算法的目標在於持續調整參數，使損失函數逐步下降，但「何時視為訓練完成」與「訓練過程是否穩定」卻並非能夠自動保證的。若沒有適當的收斂判準與穩定機制，模型可能會過早中止訓練、陷入震盪，或持續產生過擬合。

### （1）常見的收斂判準（Convergence Criteria）

- 損失函數變化趨緩
  - 當訓練損失在連續多次迭代中變化幅度極小(如變化量低於設定的 $\varepsilon$)，表示模型學習已進入平緩階段，可視為收斂。
  - 最基本且直觀的判準方式。
- 驗證集效能不再提升
  - 若模型在驗證集上的準確率、F1 分數等指標持續持平甚至下降，代表模型可能已達到其泛化能力上限。
  - 此時再繼續訓練反而可能導致過擬合。
- 梯度趨近零
  - 當模型參數的梯度值持續逼近零，表示損失函數已位於平坦區域，參數更新幅度極小，亦為常見的收斂跡象。
- 訓練步數或時間達上限
  - 在運算資源有限或訓練週期需受控的情況下，可設計固定的迭代輪數（epoch）或最大訓練時間作為訓練結束條件。
  - 此類條件應與其他指標搭配使用，避免模型尚未收斂即中止。

### （2）訓練穩定性的重要性

穩定的訓練代表模型學習曲線平順、預測逐步改善；反之，若出現劇烈震盪、發散或梯度爆炸等現象，則可能導致模型學習失敗。

以下列出幾個常見導致不穩定的因素：

- 學習率過高：
  - 更新步伐過大，導致參數在最小值附近來回震盪或完全發散。
- 初始權重設置不當：
  - 可能陷入極端值、鞍點，導致學習無法啟動或卡在局部解。
- 損失函數或資料分佈不連續：
  - 使梯度訊號不穩定，進而影響收斂路徑。
- 批次大小過小：
  - 梯度估計變異過大，造成更新方向不穩定。

### （3）常用的穩定訓練策略

為提升訓練穩定性與效率，可採取以下常見實務策略：

- 學習率調整（Learning Rate Scheduling）
  - 隨著訓練進行，自動調降學習率（如 Step Decay、Cosine Annealing、ReduceLROnPlateau），可避免後期更新過大導致震盪，提升精細收斂能力。
- 提早停止（Early Stopping）
  - 當驗證集效能在一段時間內未改善（如連續 5~10 次 epoch），即可中止訓練，以防止模型在無實質提升下持續學習而產生過擬合。
- 梯度裁剪（Gradient Clipping）
  - 為防止梯度值過大（特別是在 RNN 或深層模型中），可限制梯度的最大值，避免發散或數值不穩定。
- 批次正規化（Batch Normalization）
- 在每一層中標準化中間輸出，使輸入分佈穩定，有助於提升收斂速度與模型穩定性。

## 模擬考題

1. 若欲描述隨機變數在連續範圍內取值的機率分佈，最適合使用下列哪一種函數？

   （A）機率質量函數（PMF）

   （B）條件機率

   （C）機率密度函數（PDF）

   （D）累積分佈函數（CDF）

2. 若一個隨機變數服從伯努利分佈（Bernoulli Distribution），則該變數可能取值為下列哪一組？

   （A）任何實數

   （B）0 或 1

   （C）正整數

   （D）任意整數

3. 在主成分分析(PCA)中，常使用哪一種矩陣分解技術來找出資料的主軸方向？

   （A）矩陣求逆

   （B）矩陣轉置

   （C）特徵值分解

   （D）條件機率分解

4. 若想將高維資料映射至較低維度空間，同時保留資料的主要變異，常用下列哪一種方法？

   （A）邊際機率估計

   （B）矩陣轉置

   （C）主成分分析

   （D）累積機率計算

5. 在梯度下降（Gradient Descent）中，「梯度」代表下列哪一項？

   （A）資料點個數

   （B）模型精確度

（C）損失函數對參數的偏微分

（D）模型計算速度

6. 在進行假設檢定時，顯著水準（α）表示什麼？

（A）平均值的大小

（B）模型運算速度

（C）拒絕虛無假設時所容許的型一錯誤機率

（D）資料筆數多少

7. 若資料集中存在極端值，以下哪一種集中趨勢指標較不受影響？

（A）平均數

（B）變異數

（C）標準差

（D）中位數

8. 以下哪一種優化方法會累積每個參數的歷史梯度大小，以調整學習率？

（A）Momentum

（B）Adagrad

（C）SGD

（D）Batch Normalization

9. 當深度學習模型在訓練時出現梯度爆炸現象，應優先採用哪種技術加以處理？

（A）增大學習率

（B）梯度裁剪（Gradient Clipping）

（C）減少資料量

（D）刪除輸入特徵

10. 邏輯迴歸（Logistic Regression）模型在處理二元分類問題時，通常假設目標變數服從哪種分佈？

（A）常態分佈

（B）均勻分佈

（C）伯努利分佈

（D）泊松分佈

# 考題解析

1. **Ans（C）** 機率密度函數（PDF）

   **解析：**機率密度函數（PDF）用於描述連續型隨機變數於某個取值範圍內的機率分佈。與離散型隨機變數不同，連續變數在單一點的機率為零，PDF 通常透過積分計算區間內的機率，而 PMF 則用於離散型變數。累積分佈函數（CDF）則是累計機率的工具，而非直接描述機率密度。

2. **Ans（B）** 0 或 1

   **解析：**伯努利分佈是典型的二元分佈，僅有兩種可能結果，通常以 0 和 1 表示，例如成功與失敗、點擊與未點擊。此分佈是多數二元分類模型（如邏輯迴歸）的理論基礎。

3. **Ans（C）** 特徵值分解（Eigenvalue Decomposition）

   **解析：**PCA 透過對協方差矩陣進行特徵值分解，找出最大變異方向，並以此降低維度並保留最重要的資訊。這能讓資料投影到較少的維度上，同時盡可能維持原始資料的變異。

4. **Ans（C）** 主成分分析（PCA）

   **解析：**PCA 是常用的維度簡化工具，透過線性變換找出資料中變異最大的方向，將原始高維度資料投影到較少維度上，達到資料壓縮與特徵提取的效果。

5. **Ans（C）** 損失函數對參數的偏微分

   **解析：**梯度是損失函數相對於模型參數的導數，能指引參數應往哪個方向調整以減少損失。梯度越大，表示調整幅度應越大，以快速降低誤差。

6. **Ans（C）** 拒絕虛無假設時所容許的型一錯誤機率

   **解析：**顯著水準（α）是研究者事先設定的風險上限，代表當虛無假設為真時，錯誤拒絕該假設的容忍機率。常見設定為 0.05，意即容許 5%的型一錯誤。

7. **Ans（D）** 中位數

   **解析：**平均數會受極端值拉動而偏離中心位置。中位數則是將資料排序後取中間值，不會受到少數極端值的嚴重影響，因此在偏態分佈或極端值存在時，常用中位數描述資料集中趨勢。

8. **Ans（B）** Adagrad

   **解析**：Adagrad 根據各參數歷史梯度平方和，調整每個參數的學習率。對變動較少的參數保留較高學習率，而對變動大的參數降低學習率，特別適合處理稀疏特徵資料。

9. **Ans（B）** 梯度裁剪（Gradient Clipping）

   **解析**：梯度爆炸會導致權重變動過大甚至溢出，影響模型穩定性。梯度裁剪能限制梯度的最大值，防止數值爆炸，特別適用於深度神經網路或 RNN 訓練。

10. **Ans（C）** 伯努利分佈

    **解析**：邏輯迴歸假設目標變數為二元型態（如 0 或 1），服從伯努利分佈，並透過 sigmoid 函數將預測值映射至機率區間[0, 1]，以完成二元分類。

# 第四章　機器學習與深度學習

隨著資料規模與計算能力的提升，機器學習（Machine Learning）已成為人工智慧應用的核心基礎，而深度學習（Deep Learning）則進一步突破傳統模型的限制，在語音辨識、影像理解、自然語言處理等領域取得重大進展。本章將從機器學習的基本原理與常見技術出發，逐步延伸至深度學習的架構與實務應用，系統性建構從經典演算法到現代神經網路的技術地圖。

機器學習（Machine Learning）強調從資料中自動學習規律，主要可分為監督式學習、非監督式學習與強化式學習三大類型，分別對應分類與迴歸、聚類與降維，以及序列決策等任務。整體流程涵蓋特徵工程、模型訓練、效能評估與超參數調整等技術環節，並仰賴精確的資料處理與演算法設計，以建構具泛化能力的預測模型。

深度學習作為機器學習的子領域，透過多層神經網路進行特徵提取與結構建模，能自動學習高維資料中的抽象表示，適用於圖像、文字、語音等非結構化資料處理任務。現代深度學習架構如 CNN、RNN、Transformer 等，已成為人工智慧應用的主要模型，並依賴大規模資料與高效優化策略進行訓練。

本章將依序介紹三個主題：

- **機器學習原理與技術：**

釐清不同學習類型的基本架構與應用邏輯，並掌握偏差-變異的理論背景與模型評估方法。

- **常見機器學習演算法：**

介紹實務中常見的模型選擇，包括線性迴歸、邏輯迴歸、決策樹、支援向量機、K-means 等，並比較其適用場景與限制。

- **深度學習原理與框架：**

建構對神經網路架構與深度模型的整體理解，說明各類模型結構、運算流程與訓練技巧，並介紹主流開發框架。

## 重點掃瞄

# 4.1 機器學習原理與技術

### 1. 前言與章節導覽

機器學習（Machine Learning, ML）作為人工智慧發展的核心技術之一，強調透過資料經驗自動學習模型，使系統能根據觀察到的數據進行預測、分類或決策。與傳統規則式系統相比，機器學習不依賴人為定義的規則，而是依據資料中的統計規律，從中萃取模式與關聯，用以指導未來的判斷與行為。

隨著資料規模的快速成長與計算資源的進步，機器學習已廣泛應用於語音辨識、圖像辨識、語意理解、醫療診斷、金融預測、推薦系統等多元領域，並成為驅動智慧應用與決策系統的技術基礎。

機器學習本質上是一個涵蓋「任務類型」、「學習目標」、「資料假設」、「模型訓練」與「評估過程」的複合流程。本節將聚焦於其中最根本的知識結構與任務分類邏輯，釐清何謂監督式學習（Supervised Learning）、非監督式學習（Unsupervised Learning）與強化式學習（Reinforcement Learning），並說明在不同資料條件與應用情境下所扮演的角色。

本節亦將說明機器學習任務的核心構成，包括資料輸入與特徵空間的概念、預測目標的形式、模型學習過程的基本結構，以及學習目標如何透過損失函數進行形式化定義。

### 2. 機器學習的基本結構

機器學習系統的運作架構，建構於一系列彼此關聯的核心要素之上。從資料輸入到模型預測，整個流程涵蓋了特徵表示、任務目標定義、學習架構設計、學習目標的形式化，以及模型效能的評估。

**（1）輸入資料與特徵空間**

機器學習的輸入資料來源多樣，可能來自結構化表格、文字紀錄、影像、語音訊號或感測器資料等。這些資料必須經過適當的轉換與前處理，才能表示為模型可接受的數學結構，通常為向量或矩陣。

特徵空間（Feature Space）是指每筆資料在數學上由各個特徵構成的多維空間。每一個樣本可視為此空間中的一個點，而整體資料的幾何分佈特性（如密集程度、邊界形狀）會直接影響模型能否有效進行分類、預測或分群。因此，理解與設計適切的特徵表示，是建構機器學習模型的重要基礎。

**（2）任務目標與標籤型態**

不同的應用情境對模型的學習目標有所差異，常見的任務類型包括：

- 分類（Classification）：
  - 預測樣本所屬的類別，標籤為離散型（如 0、1 或多類別）。
- 迴歸（Regression）：
  - 預測一個連續的數值，標籤為實數。
- 聚類（Clustering）：
  - 在無標籤的情況下，根據樣本間的相似度進行分群。
- 降維（Dimensionality Reduction）：
  - 將高維資料映射至較低維度，以保留資料結構的同時簡化模型計算。
- 序列決策（Sequential Decision）：
  - 針對連續互動情境中採取的行動進行學習，以最大化長期報酬。

這些任務對應著不同的輸出形式與學習策略，並影響模型設計與評估指標的選擇。

### （3）模型與假設空間

機器學習中的模型可視為一種將輸入資料映射為預測結果的數學函數。不同的模型類型，對資料背後的分佈與結構隱含不同的假設，也稱為假設空間（Hypothesis Space）。舉例：

- 線性模型假設樣本可用線性邊界區分；
- 決策樹假設可透過一系列條件規則進行分類；
- 神經網路則傾向透過非線性結構學習抽象表示。

學習的核心任務，在於從整個假設空間中找出一個泛化能力最佳的函數，即能在未知資料上維持良好預測效果的模型。

### （4）學習目標與損失函數

為了讓模型能根據資料進行「學習」，必須透過損失函數（Loss Function）將預測與實際結果之間的誤差量化。損失函數是整個學習流程中的關鍵評價機制，其設計反映了任務的偏好與風險取捨。

常見損失函數如下：

- 均方誤差（Mean Squared Error, MSE）：
  - 用於迴歸任務，對大誤差懲罰較重；
- 平均絕對誤差（Mean Absolute Error, MAE）：
  - 用於迴歸任務，對異常值較具魯棒性；
- 交叉熵損失（Cross-Entropy Loss）：
  - 用於分類任務，衡量預測機率分佈與實際標籤的差距。

損失函數的形式不僅影響模型的數值表現，更會決定學習方向與收斂路徑。

### （5）資料分割與評估準則

為確保模型具備良好的泛化能力，訓練過程中需將資料劃分為：

- 訓練集（Training Set）：用於模型參數的學習；
- 驗證集（Validation Set）：用來調整超參數與監控過擬合；
- 測試集（Test Set）：最終評估模型表現的依據。

評估指標應依任務類型選擇，例如：

- 分類任務：
  - 準確率（Accuracy）、精確率與召回率（Precision/Recall）、F1 分數；
- 迴歸任務：
  - 平均絕對誤差（MAE）、均方誤差（MSE）、決定係數（$R^2$）。

合理的資料分割與多元評估，有助於全面觀察模型的學習品質與應用潛力。

## 3. 監督式學習

監督式學習（Supervised Learning）是機器學習中最常見且應用最成熟的學習形式，其核心目標是透過「標註資料（Labeled Data）」學習一個映射函數，讓模型能根據輸入特徵預測對應的輸出結果。

學習的目標是最小化模型輸出與實際標籤之間的誤差，同時具備良好的泛化能力。

### （1）基本架構

在監督式學習中，每筆訓練資料由一組輸入特徵（Features）與其對應的目標標籤（Label）組成。模型的任務是在學習過程中找出一個能夠近似真實對應關係的函數，使未來面對新資料時也能產生準確預測。

### （2）主要任務類型

根據預測標籤的類型，監督式學習可分為以下兩大任務：

- 分類（Classification）
  - 當標籤為離散型類別時，任務即為分類。模型需判斷輸入樣本應屬於哪一類別。常見應用包括：
    - 郵件分類：垃圾信 vs 一般信件
    - 客戶流失預測：流失 vs 留存
    - 圖像辨識：辨識影像中是貓、狗或其他物件
  - 分類模型通常輸出各類別的預測機率，再根據最高機率選擇預測結果。
- 迴歸（Regression）
  - 當標籤為連續數值時，屬於迴歸任務。模型需預測一個實數值，例如：
    - 房價預測
    - 銷售量預測
    - 使用者滿意度評分（如 1 至 5 顆星）
  - 常見的損失函數包括均方誤差（Mean Squared Error, MSE）與平均絕對誤差（Mean Absolute Error, MAE）。

### （3）訓練流程與評估方法

一個典型的監督式學習流程包含以下步驟：

- 資料準備與標記：
  - 蒐集具標籤資料，並進行清理與前處理。
- 模型訓練：
  - 根據訓練集進行參數調整，最小化損失函數。
- 驗證與調整：
  - 使用驗證集觀察模型表現，調整模型結構與超參數。
- 測試與部署：
  - 於測試集上評估最終效能，進行應用部署。

### （4）應用場景

監督式學習在實務中有廣泛應用，涵蓋多種產業與任務需求，典型例子包括：

- 圖像分類（影像 → 物件類別）
- 客戶信用風險評估（資料 → 信用等級）
- 醫療診斷（病患資訊 → 疾病類型）

在這些情境中，模型效能高度仰賴資料的標註品質與特徵設計的精準性，因此資料前處理、特徵選擇與轉換仍是監督式學習成敗的關鍵。

## 4. 非監督式學習

非監督式學習（Unsupervised Learning）是一種在無標註資料情況下進行學習的機器學習方法。其核心目的是從原始資料中發現潛在的結構、模式或分佈特性，而無需依賴人工標記的目標輸出。這類方法尤其適用於探索性分析、資料壓縮與隱含關係挖掘等任務。

### （1）基本架構

與監督式學習不同，非監督學習模型的輸出通常不是具體的預測值，而是：

- 對資料樣本的分組、歸類
- 對高維資料的降維、投影
- 對潛藏變數或生成機制的估計

這種學習方式強調對資料本質的理解與內部表示的建構，常作為數據前處理、特徵抽取或結構探索的關鍵步驟。

### （2）主要任務類型

非監督式學習依據學習目標與應用需求，劃分為幾個主要任務類型：

- 聚類（Clustering）
  - 將資料依據其相似性自動劃分為若干群組，群內樣本彼此相似，群間差異明顯。
- 降維（Dimensionality Reduction）
  - 將高維資料轉換為低維表示，以保留重要結構並便於視覺化或後續建模。
- 關聯規則學習（Association Rule Learning）
  - 從大量資料中找出項目間的關聯性或共現規律
- 潛在表示學習（Representation Learning）
  - 透過非監督方式學習可解釋或有用的資料內部結構，特別常見於語言模型、影像編碼等深度學習應用。

**（3）訓練流程與評估方法**

非監督式學習缺乏明確的標籤，因此其訓練與評估方式與監督式學習有所不同：

- 訓練流程：
  - 資料準備與標準化（常需進行特徵縮放、中心化等）
  - 模型選擇與超參數設定（如聚類數 k）
  - 模型擬合與重複迭代（許多方法依賴初始條件）
- 評估方式：
  - 內部評估指標：根據資料本身的結構特徵來評估學習效果，如：
    - Silhouette score（輪廓係數）
    - Davies-Bouldin（指數）
    - Reconstruction Error（重建誤差）
  - 外部評估指標：若有部分標籤可參考，則可使用：
    - Rand Index
    - Adjusted Mutual Information（AMI）

- 視覺化輔助分析：
  - 將高維結果降維至 2D/3D 空間進行圖形檢視，是非監督式學習中常見的理解方式。

**（4）應用場景**

非監督式學習雖不依賴標註資料，但其應用廣泛，常見領域包括：

- 客戶分群與行為分析（電商、行銷）
- 社群偵測與異常發現（社群網路、資安）
- 影像壓縮與重建（自編碼器、壓縮技術）
- 主題模型與語意分析（自然語言處理）
- 特徵工程與前處理（資料探索與建模前步驟）

非監督式學習不僅在探索未知資料結構中具有關鍵價值，也常用於模型初始化、資料標註支援與半監督學習的先備處理階段。

## 5. 強化式學習簡介

強化式學習（Reinforcement Learning,RL）是一種與監督式與非監督式學習並列的核心機器學習範式，其特徵在於模型透過與環境互動學習決策策略，目標是最大化長期累積報酬（Reward）。

強化式學習不僅廣泛應用於自走車、遊戲 AI、機器人控制等任務，也在資源分配、推薦系統與金融決策等動態環境中展現出高效潛力。

**（1）基本架構**

強化式學習問題可形式化為馬可夫決策過程（Markov Decision Process, MDP），基本組成包括：

- 代理人（Agent）：執行動作並從經驗中學習的人工智慧。
- 環境（Environment）：代理人互動的外部系統。

- 狀態（State）：代理人於某一時刻觀察到的環境資訊。
- 動作（Action）：代理人在特定狀態下可採取的行為。
- 報酬（Reward）：環境對代理人某一行為的回饋，用來指引學習方向。
- 策略（Policy）：代理人根據當前狀態選擇動作的規則。
- 價值函數（Value Function）：衡量某一狀態或狀態 - 動作對的長期獎勵期望值。

代理人的學習目標是根據歷史互動經驗，調整其策略，使在不同狀態下選擇的行動能獲得最大的長期累積報酬。

**（2）主要任務類型**

強化式學習的任務類型依據任務設計與學習方式的不同，可分為：

- 策略學習（Policy Learning）：
  - 直接學習最佳策略，例如策略梯度法（Policy Gradient）。
- 價值學習（Value-Based）：
  - 學習狀態或狀態 - 動作對的價值函數，如 Q-learning。
- 模型式學習（Model-Based RL）：
  - 嘗試學習環境轉移與回饋機制，提升策略更新效率。
- 模型無關學習（Model-Free RL）：
  - 無需環境模型，依賴試誤與經驗回放。

另可依決策空間區分，例如：

- 離散動作空間：適用於分類型選擇，如遊戲動作、導航決策。
- 連續動作空間：適用於控制類任務，如機械臂運動、車輛轉向控制。

### （3）訓練流程與評估方法

強化式學習的訓練流程具高度互動與回饋循環性，包含以下步驟：

A. 初始化策略或價值函數

B. 與環境互動並收集經驗

C. 根據報酬更新策略或價值估計

D. 重複試誤學習，逐步提升決策表現

評估方式則不同於分類或迴歸任務的靜態指標，需透過模擬或實際環境測試長期績效，指標包括：

- 平均累積報酬（Average Reward per Episode）
- 成功率或任務達成率
- 策略穩定性與收斂速度

因訓練過程涉及動態回饋、延遲效應與探索策略，強化式學習對收斂控制與樣本效率的要求特別高。

### （4）應用場景

強化式學習具備適應動態環境與策略自我調整能力，特別適合下列情境：

- 遊戲 AI 與對弈系統：AlphaGo、OpenAI Five、DeepMind Atari 等。
- 自駕車與機器人控制：動作序列學習、導航決策、連續控制。
- 推薦系統與廣告分發：即時反饋優化、長期使用者價值最大化。
- 金融投資與資源配置：連續決策、風險控制、強化投資策略。
- 運輸與物流排程：多階段決策與最短路徑規劃。

強化式學習技術在策略最適化與動態決策領域展現高度潛力，但同時也面臨如樣本效率低、訓練不穩定、實務部署門檻高等挑戰。

## 重點掃瞄

# 4.2 常見機器學習演算法

### 1. 前言與章節導覽

機器學習演算法的核心任務，是學習一個從輸入特徵映射至目標結果的規則，並具備良好的泛化能力。根據學習任務的不同，機器學習方法大致可分為兩大類型：監督式學習（Supervised Learning）與非監督式學習（Unsupervised Learning），兩者在資料需求、訓練方式與應用情境上具有明顯差異。

### 2. 監督式學習- 迴歸任務

在機器學習中，迴歸分析（Regression）主要用於預測連續數值結果的常見任務。其核心目標是建立數學模型，描述輸入變數（自變數、特徵）與目標變數（應變數、目標值）之間的關係。迴歸任務在各種應用領域中扮演重要角色，如房價預測、股票價格預測、醫療風險評估、能源消耗預測等主題。不同的迴歸演算法，能處理不同的資料特性與挑戰，如多重共線性、非線性關係、異常值影響等。因此，選擇合適的迴歸模型，是建構精準預測系統的關鍵。

#### （1）線性迴歸

- 定義
  - 線性迴歸（Linear Regression）是最基礎、最常用的監督式學習方法之一，主要用於預測連續型的目標變數（例如價格、銷售額、醫療數值等）。
  - 線性迴歸之核心概念，是透過一條直線或多維超平面，描述自變數（輸入特徵）與應變數（目標值）之間的線性關係，並找出最佳迴歸係數，使預測值與實際值之間的誤差最小化。

- 模型公式
  - 多變量線性迴歸的一般形式為：

    $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n + \varepsilon$

    其中：
    - y：目標變數（應變數）
    - $\beta_0$：截距（Intercept）
    - $\beta_1, \beta_2, \ldots, \beta_n$：各自變數的迴歸係數（Slope），代表每個自變數對預測值的影響
    - $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$：自變數（輸入特徵）
    - $\varepsilon$：誤差項，用於表示模型無法解釋的隨機變動或噪聲
- 模型評估指標（詳見第三章 3.3）
  - 平均平方誤差（MSE, Mean Squared Error）
    - 測量預測值與實際值之間誤差的平方平均，數值越小表示預測越準確。
  - 均方根誤差（RMSE, Root Mean Squared Error）
    - 為 MSE 的平方根，單位與目標變數相同，更易於解讀。
  - 平均絕對誤差（MAE, Mean Absolute Error）
    - 預測值與實際值之間誤差的絕對值平均，相較於 MSE、MAE 對離群值的敏感度較低。
  - 決定係數 $R^2$
    - 代表模型能解釋目標變數變異的比例，範圍在 0~1，越接近 1 表示模型解釋力越強。
  - 調整後 $R^2$
    - 考慮自變數數量對模型複雜度的影響，適合用於比較不同複雜度模型。
- 模型假設
  - 線性關係（Linearity）

- 自變數與應變數之間應存在線性關係。

- 誤差常態分佈（Normality of Errors）
  - 誤差項 $\varepsilon$ 須符合常態分佈。
- 變異數齊一性（Homoscedasticity）
  - 誤差項的變異數應在不同自變數取值下保持相同。
- 誤差獨立性（Independence of Errors）
  - 各觀測值之間的誤差應獨立無關。
- 無多重共線性（No Multicollinearity）
  - 自變數之間不應高度相關，以免影響係數估計的穩定性。

- 適用情境
  - 預測連續型數值。
  - 變數間關係接近線性。
  - 需要模型具可解釋性，了解每個自變數對結果的影響。
  - 資料規模不過於龐大，特徵數量適中。
- 使用限制
  - 對離群值敏感
    - 少數極端值可能對模型係數產生巨大影響。
  - 無法捕捉非線性關係
    - 若資料關係呈現明顯非線性，線性迴歸無法有效建模。
  - 多重共線性問題
    - 當自變數間高度相關時，會造成迴歸係數不穩定，影響解釋力。
  - 假設違反風險
    - 若模型假設不成立，預測結果與統計推論都可能失真。

**（2）Lasso 迴歸與嶺迴歸**

- 定義
  - Lasso 迴歸（Lasso Regression）
    - 在損失函數中加入 L1 正則化項（權重絕對值和），不僅限制係數大

小，還能將部分係數直接縮減為零，達到特徵選擇（Feature Selection）的效果。

- Lasso 全名為：Least Absolute Shrinkage and Selection Operator。

- 嶺迴歸（Ridge Regression）
  - 在損失函數中加入 L2 正則化項（權重平方和），防止模型產生過大的係數，特別適合多重共線性嚴重的情況。

- 模型公式
  - Lasso 迴歸

    $$\text{Loss} = \sum (y_i - \hat{y}_i)^2 + \lambda \sum |\beta_j|$$

  - 嶺迴歸

    $$\text{Loss} = \sum (y_i - \hat{y}_i)^2 + \lambda \sum \beta_j^2$$

  - 其中
    - $y$：目標變數（應變數）
    - $\hat{y}_i$：預測值
    - $\beta_j$：各自變數的迴歸係數
    - $\lambda$：正則化係數，控制限制強度
    - $\sum |\beta_j|$：所有係數絕對值和（Lasso）
    - $\sum \beta_j^2$：所有係數平方和（Ridge）

- 模型評估指標

  Ridge 與 Lasso 在評估上，仍使用與線性迴歸相同的指標：
  - 平均平方誤差（MSE, Mean Squared Error）
  - 均方根誤差（RMSE, Root Mean Squared Error）
  - 平均絕對誤差（MAE, Mean Absolute Error）
  - 決定係數 $R^2$（Coefficient of Determination）
  - 調整後 $R^2$（Adjusted R-squared）

- 模型假設

  Ridge 與 Lasso 與一般線性迴歸相同，依賴以下假設條件：

  - 線性關係（Linearity）
    - 自變數與應變數之間應存在線性關係。
  - 誤差常態分佈（Normality of Errors）
    - 誤差項 $\varepsilon$ 須符合常態分佈。
  - 變異數齊一性（Homoscedasticity）
    - 誤差項的變異數應在不同自變數取值下保持相同。
  - 誤差獨立性（Independence of Errors）
    - 各觀測值之間的誤差應獨立無關。
  - 無多重共線性（No Multicollinearity）
    - 自變數之間不應高度相關，以免影響係數估計的穩定性。

- 適用情境

  Ridge 與 Lasso 特別適用於以下情況：

  - 資料存在多重共線性時。
  - 變數數量遠大於樣本數時（高維資料）。
  - 希望防止模型過度擬合。
  - 希望同時進行特徵選擇（Lasso）。
  - 預測精度比模型解釋度更重要的情境。

- 使用限制

  - Lasso
    - 若變數之間高度相關，可能只保留其中一個變數，忽略其他。
    - λ 選擇不當可能過度壓縮模型，導致欠擬合。
  - Ridge
    - 不會將係數壓縮為零，因此無法自動進行特徵選擇。
    - 無法直接簡化模型結構。

- 共通限制
  - 對於非線性關係仍無法建模，需要搭配其他方法。
  - 正則化係數 λ 的設定需透過交叉驗證或其他方法選擇，無法一次設定好。

### （3）支援向量迴歸（SVR）

- 定義
  - 支援向量迴歸（Support Vector Regression, SVR）是支援向量機（Support Vector Machine, SVM）的延伸，用於處理迴歸問題。
  - SVR 核心概念是在多維空間中尋找一條最能描述資料趨勢的超平面，使預測誤差落在允許範圍（$\varepsilon$-tube）內，同時將超出範圍的誤差最小化。
  - SVR 與傳統線性迴歸不同，SVR 不追求最小化所有點的平方誤差，而是盡可能忽略誤差小於 ε 的資料點，只關注誤差超過 ε 的點，藉此提升對離群值的抵抗力。
- 模型公式
  - SVR 的目標是求解以下最小化問題：

$$\min_{w,b,\xi,\xi^*} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*)$$

  - 其中：
    - $w$：權重向量，定義超平面的方向
    - $\xi_i$、$\xi_i^*$：用於計算超過 $\varepsilon$ 容忍範圍外的誤差
    - $C$：正則化參數，控制模型複雜度與對誤差的容忍度。（也可稱為懲罰參數、看作懲罰超出 $\varepsilon$ 的誤差）
    - ε：誤差容忍範圍（$\varepsilon$-tube）
  - 迴歸函數為：

$$f(x) = w \cdot x + b$$

- 若配合核函數（Kernel Function），則變為非線性形式：

$$f(x) = \sum_{i=1}^{n} \alpha_i K(x_i, x) + b$$

- 模型評估指標
  - 平均平方誤差（MSE, Mean Squared Error）
  - 均方根誤差（RMSE, Root Mean Squared Error）
  - 平均絕對誤差（MAE, Mean Absolute Error）
  - 決定係數 $R^2$（Coefficient of Determination）
  - 調整後 $R^2$（Adjusted R-squared）
- 模型假設

  SVR 的假設較少，不像線性迴歸依賴嚴格的統計假設：
  - 不強制自變數與應變數為線性關係，可利用核函數進行非線性映射。
  - 不需要誤差項服從常態分佈。
  - 對異常值較具抵抗力，因多數誤差在 $\varepsilon$ 範圍內會被忽略。
  - 假設資料是可分的或近似可分的。
- 適用情境
  - 資料存在非線性關係，需要更靈活的模型表達能力。
  - 資料規模中小，且維度不是過高（因計算成本較高）。
  - 需要對離群值具一定抵抗力。
- 使用限制
  - 計算複雜度高
    - 在大型資料集或高維度情境下，計算資源消耗大。
  - 參數設定複雜
    - 模型表現高度依賴：
      - $C$：懲罰參數。
      - $\varepsilon$：誤差容忍範圍。
      - 核函數與其參數（例如 RBF 核的 $\gamma$）。

- 不易解釋
  - 尤其在使用非線性核函數時，模型較難直觀理解。
- 不適合極大規模資料
  - 訓練時間和記憶體需求會隨樣本數平方增長。

### （4）決策樹迴歸

- 定義
  - 決策樹迴歸（Decision Tree Regressor）是一種監督式學習演算法，用於預測連續型目標變數。
  - 決策樹迴歸核心概念是依據輸入特徵的不同取值，將資料不斷分割成更小的子區塊，並在每個葉節點上給出一個預測值，通常是該區塊內樣本的平均值。
  - 相較於線性迴歸，決策樹迴歸能捕捉更複雜、非線性的資料結構，且對資料的分佈與尺度不敏感，具有很強的表達力。
- 模型公式
  - 決策樹迴歸本身沒有像線性模型那樣的明確數學公式，其預測流程為：
    a. 從根節點出發，根據某個特徵及其分割閾值，將資料分成左右兩部分。
    b. 對每個子節點重複步驟 a.，直到：
       - 達到樹的最大深度
       - 每個節點樣本數小於最小限制
       - 節點內樣本變異度足夠小
    c. 在葉節點，將落在該節點的所有樣本的目標值取平均，作為該節點的預測值。
  - 示例：
    - 若資料被分割至某一葉節點，其中有 10 筆資料，其目標值平均為 52.3，則所有落入此葉節點的資料預測值皆為 52.3。

- 模型評估指標
  - 平均平方誤差（MSE, Mean Squared Error）
  - 均方根誤差（RMSE, Root Mean Squared Error）
  - 平均絕對誤差（MAE, Mean Absolute Error）
  - 決定係數 $R^2$（Coefficient of Determination）
  - 調整後 $R^2$（Adjusted R-squared）
- 模型假設

  決策樹迴歸屬於非參數模型，相較於線性迴歸，不強制以下假設：
  - 不需假設自變數與應變數呈線性關係。
  - 不需誤差常態分佈。
  - 不需變異數齊一性。
  - 不需考慮多重共線性問題。
- 適用情境
  - 預測連續型目標變數。
  - 資料可能具有非線性關係或高階交互作用。
  - 希望產生易解釋、具邏輯分支的模型。
  - 特徵可能包含類別型或數值型混合資料。
  - 資料規模中小至中大。
- 使用限制

  決策樹迴歸擁有高解釋力與非線性處理能力，但也存在以下限制：
  - 容易過擬合
    - 單棵決策樹很容易完全擬合訓練資料，造成泛化能力不足。
  - 模型不連續
    - 決策樹的預測結果為分段常數，對於連續變化的目標變數，預測值可能出現不連續的跳躍。
  - 對資料微小變動敏感
    - 資料略微變動可能導致樹的結構變化很大，影響穩定性。

- 無法擅長捕捉非常複雜的邊界
  - 雖然能捕捉非線性，但在高維空間中，單棵樹表現有限。

### （5）集成式迴歸

集成式迴歸（Ensemble Regression）是一種透過結合多個模型的預測結果，來提升整體預測的準確度與穩定性的機器學習方法。其核心概念是：「群體智慧比單一模型更強大」。即使單個模型表現有限，當多個模型以不同角度學習同一筆資料，再將結果整合，就能降低錯誤、減少過擬合，並提升模型的泛化能力。

在迴歸任務中，集成方法非常受到重視，尤其是隨著資料變得龐大、複雜，單一模型往往無法應付所有挑戰。常見的集成式迴歸方法包括：

- 隨機森林迴歸

  隨機森林迴歸（Random Forest Regressor）是一種集成式演算法，透過建立多棵決策樹來進行預測。每棵樹都在不同的隨機子樣本（Bootstrap Sample）上訓練，且在每次分裂節點時，只隨機選取部分特徵進行考慮。最終預測結果是所有樹的預測值平均值，藉此降低單棵樹容易出現的過擬合問題。除了預測外，隨機森林也能計算特徵的重要性，協助了解資料中哪些變數對結果影響最大。

  - 特點
    - 抗過擬合能力強，即使資料複雜，也能維持穩定表現。
    - 可處理非線性資料與變數間的高階交互作用。
    - 可用於資料集內同時存在類別型與數值型特徵的情境。
    - 計算特徵重要性，有助於後續特徵篩選與解讀。
    - 適用於中到大型資料集。
    - 訓練可進行並行運算，加速建模過程。
  - 缺點
    - 難以直觀解釋所有樹的整體結構，屬於「黑箱模型」。
    - 訓練完成後模型體積可能較大，對記憶體需求較高。

- 單棵樹雖易視覺化，但整體模型複雜，不易解析變數交互關係。
- 預測速度比單棵樹慢，尤其樹數量多時。

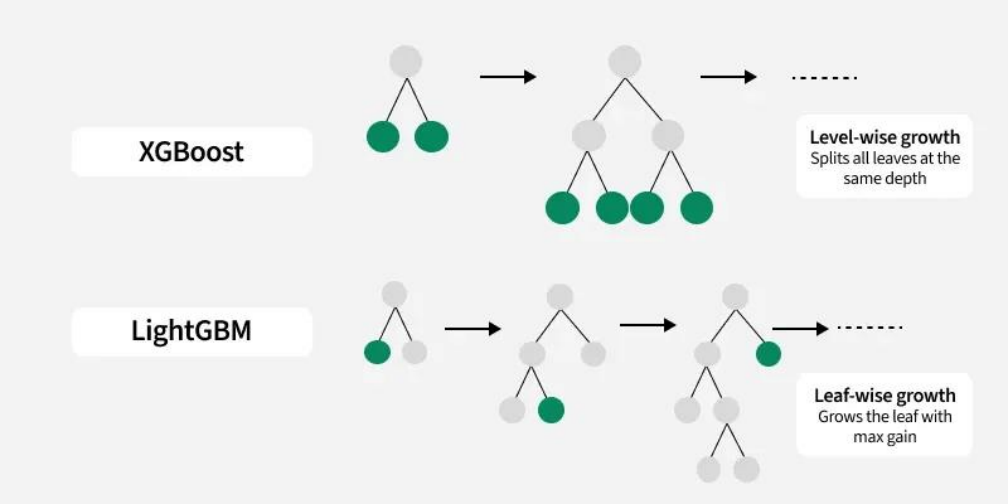
- 梯度提升迴歸

  梯度提升迴歸（Gradient Boosting Regressor）是一種序列式集成演算法，核心概念是「逐步修正誤差」。先訓練一棵簡單的決策樹，計算預測殘差後，再訓練下一棵樹專門學習修正這些殘差。如此一棵一棵地疊加，最終將多棵弱學習器（Weak Learners）加總，形成一個更好的預測模型。由於每一步都針對前一次錯誤進行調整，因此能在許多複雜問題中達到高度精確的表現。

  - 特點
    - 預測精確度通常高於隨機森林，是許多競賽常勝軍。
    - 擅長捕捉複雜的非線性關係與變數交互作用。
    - 可自訂損失函數，具高度彈性。
    - 可透過各決策樹的重要性分數來解讀變數影響。
    - 在中型到大型資料集上表現良好。
    - 訓練過程可搭配早停（Early Stopping）機制，減少過擬合風險。
  - 缺點
    - 訓練過程比隨機森林更慢，尤其在樹數多、樹深大的情況下。
    - 容易產生過擬合，尤其在學習率過高或樹太深時。
    - 超參數多，需要仔細調整（如學習率、樹深、子樣本比例等）。
    - 不易直觀解釋整體模型運作，尤其在深度很高時。

- 高效梯度提升方法（如 XGBoost、LightGBM）

  高效梯度提升方法是對傳統梯度提升樹（Gradient Boosting Decision Trees, GBDT）進行演算法層面及工程層面的優化，專門解決傳統 GBDT 在大型數據、計算速度與記憶體消耗上的限制。在眾多實作中，XGBoost 和 LightGBM 是最知名的兩種高效梯度提升方法，也成為 AI 競賽、金融、電商、醫療等領域不可或缺的工具。




（Source: https://www.geeksforgeeks.org/）

- XGBoost（eXtreme Gradient Boosting）
  - 強調運算效率與正則化，採用二階導數資訊（Hessian）來加速最佳分裂點搜尋，支援稀疏資料結構、缺失值自動處理，並內建防止過擬合的機制。
  - 採用「層級式生長」Level-wise 策略。
  - 在同一深度上同時分裂所有葉子節點。
  - 這意味著樹會橫向地擴展，每次增加一層深度時，所有當前葉子節點都會被考慮分裂。
- LightGBM（Light Gradient Boosting Machine）
  - LightGBM 採用「葉子式生長」Leaf-wise 分裂策略，而非如 XGBoost 採用的 Level-wise 分裂策略，讓演算法在更少計算下達到更好的效果。對於大數據、大特徵數的問題尤其高效，且在記憶體使用上更省。
  - 優先分裂能帶來最大增益（max gain）的葉子節點，這導致樹會垂直地生長，形成更深但不一定均勻的樹結構。
  - 相較於 XGBoost，LightGBM 的葉子式生長策略通常能更快地達到更高的準確度，同時在速度上更具優勢。

- 特點
  - 高運算效率
    - 採用先進的分裂演算法，減少計算次數。
    - XGBoost 使用 Block 結構加速計算。
    - LightGBM 使用直方圖算法（Histogram-based）降低計算複雜度。
  - 支援稀疏資料與缺值
    - XGBoost 可自動學習缺失值走向。
    - LightGBM 同樣內建缺值處理，避免額外前處理。
  - 可進行並行運算
    - 多核心 CPU、甚至 GPU 都能加速。
    - LightGBM 在分裂點搜尋上有更高並行度。
  - 正則化控制
    - XGBoost 引入 L1、L2 正則化，有助抑制過擬合。
    - LightGBM 也提供多種正則化參數。
  - 先進的分裂策略
    - LightGBM 的 Leaf-wise 分裂通常能在同樣的模型體積下取得更低的誤差。
    - 支援多種精細調參，如 max_depth、min_child_weight 等。
  - 良好的特徵重要性分析
    - 輸出各變數在模型中的重要程度，幫助後續解讀。
- 缺點
  - 參數複雜多元
    - 例如 learning_rate、max_depth、subsample、colsample_bytree 等多個參數需要仔細調整。
    - 調參過程對初學者來說學習曲線較陡。

- 解釋力有限
  - 雖然可以查看特徵重要性，但完整模型仍是黑箱，無法像線性模型那樣直觀。
- 記憶體需求仍高
  - 尤其是 Leaf-wise 策略，若沒有適當限制深度，容易導致過多節點、耗盡記憶體。
- Leaf-wise 可能導致過擬合（LightGBM 特有問題）
  - 在小型資料集上，Leaf-wise 容易生成不平衡樹，需配合 max_depth 等參數限制。
- 輸出不平滑
  - 預測結果為分段函數，對某些需要平滑輸出的應用（如價格曲線）可能不理想。

### 3. 監督式學習-分類任務

在機器學習中，分類（Classification）是另一個極為重要且常見的任務，主要用於預測離散的類別結果。其核心目標是建立一個數學模型，能夠根據輸入變數（自變數、特徵），判斷資料應屬於哪一類別。分類模型不僅用於二元分類（如判斷是或否、真或假），更廣泛應用於多類別問題，如文件主題分類、疾病診斷類別、圖像辨識、語音辨識等領域。

分類任務在現代應用中扮演關鍵角色，例如：

- 判斷電子郵件是否為垃圾信
- 預測客戶是否可能流失
- 醫學檢測中診斷疾病類
- 社群平台中自動標記圖片內容

不同的分類演算法，各自具備不同的數學假設、優勢與限制。例如，邏輯迴歸簡單易解釋，但無法捕捉複雜的非線性邊界；支援向量機能處理高維度資料，但計算成本較高；集成方法如隨機森林或梯度提升，通常能在準確度上表現優異，

但缺點是模型較難解釋。依據資料特性、規模、以及解釋需求，挑選合適的分類模型，是建置高效能機器學習系統的關鍵步驟。

### （1）邏輯迴歸

- 定義
  - 邏輯迴歸（Logistic Regression）是一種最基礎、最常用的分類演算法，雖名為「迴歸」，實際上是用於解決分類問題，尤其適合二元分類（Binary Classification）。
- 模型公式
  - 邏輯迴歸的模型形式如下：

$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n)}}$$

  - 其中：
    - p：資料屬於正類（Class 1）的機率
    - $\beta_0$：截距（Intercept）
    - $\beta_1, \beta_2, \ldots, \beta_n$：各特徵的係數
    - $x_1, x_2, \ldots, x_n$：各特徵值
- 模型評估指標（詳見第三章 3.3）
  - Accuracy（準確率）
    - 預測正確的比例
    - 包括所有正確預測的數量（正確預測為正類和正確預測為負類）除以總樣本數。
  - Precision（精確率）
    - 預測為正類中，真正為正的比例。
  - Recall（召回率）
    - 真實為正類中，被正確預測為正的比例。

- F1-score
  - Precision 與 Recall 的調和平均，用於衡量模型在不平衡資料下的表現。
- ROC-AUC
  - 衡量模型區分正、負類的能力，值越接近 1 表示模型表現越好。

- 模型假設
  - 線性可分性（Linearity in Log-Odds）
    - 特徵與對數勝率（Log-Odds）之間呈線性關係。
      - 對數勝率(Log-Odds):對事件發生機率與不發生機率比值(勝率)取自然對數後的結果。將機率的[0,1]範圍轉換為連續的實數(−∞, +∞）範圍，常用於邏輯迴歸。
  - 資料獨立性
    - 每筆觀測值彼此獨立。
  - 無多重共線性（No Multicollinearity）
    - 特徵變數之間不宜具有高度相關性，否則會導致係數不穩定。
- 適用情境
  - 預測結果是二元分類（如：是否購買、是否患病)。
  - 希望模型容易解釋，尤其在金融、醫療等需要解釋原因的領域。
  - 資料關係大致線性，特徵間獨立性較高。
  - 特徵數量不過多，資料量適中。
- 使用限制
  - 無法捕捉複雜的非線性邊界
    - 若資料具有高度非線性關係，模型效果會大幅下降。
  - 對離群值敏感
    - 離群值會顯著影響係數估計。
  - 需要滿足假設條件
    - 若資料不滿足線性關係、獨立性或變異數穩定，模型效果會受影響。

- 多類別問題需擴展
  - 對多類別問題（Multiclass Classification）需要使用如 One-vs-Rest（OvR）或 Multinomial Logistic Regression 等方法。

**（2）支援向量機（SVM）**

- 定義
  - 支援向量機（Support Vector Machine, SVM）是一種監督式學習演算法，主要用於分類任務，也能應用於迴歸問題（SVR）。
  - SVM 的核心思想是，在特徵空間中尋找一條最能區分不同類別的決策邊界（超平面），並最大化兩類之間的間隔（Margin）。
  - 若資料無法在原空間中線性分離，SVM 可利用「核函數（Kernel Function）」將資料映射到更高維度空間，使其可被線性分隔。這也是 SVM 能處理非線性分類問題的重要原因。
- 模型公式
  - SVM 建立的決策邊界可表示為：

    $w \cdot x + b = 0$
    - 參數說明：
      - w：權重向量
      - x：輸入特徵向量
      - b：偏移量（截距）
    - 分類判斷依據為：
      - 若 $w \cdot x + b > 0$，分類為正類
      - 若 $w \cdot x + b < 0$，分類為負類
  - 若採用核函數 K，則決策函數可表示為：

    $f(x) = \sum \alpha_i \cdot y_i \cdot K(x_i, x) + b$
    - 參數說明：
      - $\alpha_i$：支援向量的權重

- $y_i$：支援向量的真實標籤
- $x_i$：支援向量
- x：欲預測的輸入向量
- K（$x_i$, x）：核函數計算結果
- b：偏移量（截距）

- 模型評估指標
  - Accuracy（準確率）
  - Precision（精確率）
  - Recall（召回率）
  - F1-score
  - ROC-AUC
- 模型假設
  - 適用於線性可分或近似線性可分的資料。
  - 核函數使其能處理非線性邊界。
  - 在高維空間表現良好，即使特徵數量遠大於樣本數。
  - 對於小型、中型資料集尤為有效。
  - 常用核函數：
    - 線性核（Linear Kernel）
    - 多項式核（Polynomial Kernel）
    - 徑向基函數核（RBF Kernel, Gaussian Kernel）
    - Sigmoid 核
- 適用情境
  - 資料集規模中小，特徵維度高。
  - 資料邊界較為清晰或需要精準分界。
  - 希望有較高的分類準確率。
  - 需要在複雜邊界情境下應用。

- 使用限制
  - 計算成本高
    - 特別在大資料集下，訓練時間與記憶體消耗相當可觀。
  - 參數調整複雜
    - 成效高度依賴於：
      - C：懲罰參數（或稱正則化參數），控制誤差容忍程度。
      - γ（gamma）：核函數的參數，特別是 RBF 核函數。

### （3）決策樹分類器

- 定義
  - 決策樹分類器（Decision Tree Classifier）是一種監督式學習演算法，透過一系列的「如果…那麼…」判斷規則，將資料依特徵分割成不同群組，最終在每個葉節點上給出一個類別預測。
  - 決策樹分類器就像一個流程圖，從根節點（Root Node）開始，根據特徵值進行判斷，逐步向下分裂，直到到達葉節點（Leaf Node）。
- 模型公式
  - 決策樹不是透過傳統的數學公式建模，而是透過以下流程進行：

    a. 選擇一個特徵作為分裂依據，找到最佳切分點。

    b. 根據該特徵的切分點，把資料分成兩個子群。

    c. 重複步驟 a.~b.，直到：
      - 到達最大樹深度（max_depth）
      - 節點內的樣本數低於設定值（min_samples_split）
      - 節點內的樣本純度已經足夠高
- 模型評估指標

  決策樹透過計算以下指標，決定如何分裂：
  - 基尼不純度（Gini Impurity）

        - 用於衡量節點內的混雜程度。值越小代表節點內樣本越集中於單一類別。
    - 資訊增益（Information Gain）
        - 根據熵（Entropy）的變化，衡量分裂後的不確定性降低了多少。
    - 分類誤差（Classification Error）
        - 計算節點內樣本分錯的比例。
- 模型假設
    - 不需要特徵與目標變數呈線性關係。
    - 可處理類別型與數值型混合特徵。
    - 能夠捕捉變數之間的交互作用。
    - 對資料尺度不敏感（不需要標準化）。
- 適用情境
    - 預測結果是類別型。
    - 需要易於解釋的模型（例如：醫療、金融領域）。
    - 資料集不過於龐大。
- 使用限制
    - 容易過擬合
        - 單棵樹容易學習到資料中的噪聲，導致泛化能力不足。
    - 對資料微小變動敏感
        - 同樣的資料集若稍有變化，整棵樹可能會大幅改變。
    - 模型不連續
        - 對連續型變數，決策邊界呈階梯狀，而非平滑曲線。
    - 效能較低於集成方法
        - 單一決策樹在預測精度上通常不如集成式方法，例如隨機森林或梯度提升樹。

### （4）K 最近鄰分類

- 定義
  - K 最近鄰分類（K Nearest Neighbors, KNN）是一種非參數、惰性學習的分類演算法。
  - KNN 不建立顯式的模型，而是將新的資料點分類到其「K」個最近鄰居中佔多數的類別。就像「物以類聚」，判斷一個新樣本屬於哪個類別，就看它身邊大部分鄰居是哪個類別。
- 模型原理
  - KNN 的核心思想是基於相似性度量。當一個新的、未知的資料點出現時，KNN 會執行以下步驟：
    a. 計算距離：
       - 計算新資料點與訓練集中所有資料點之間的距離（例如歐幾里得距離、曼哈頓距離等）。
    b. 選擇最近的 K 個鄰居：
       - 找出距離新資料點最近的 K 個訓練樣本。
    c. 投票決定類別：
       - 檢查這 K 個鄰居的類別標籤，將新資料點歸類為這 K 個鄰居中出現次數最多的類別。
- 模型評估指標
  - Accuracy（準確率）
  - Precision（精確率）
  - Recall（召回率）
  - F1-score
  - ROC-AUC
- 模型假設
  - 「近朱者赤」的假設：

- 假設彼此相近的資料點具有相似的特性，因此它們的類別也可能相同。

- 特徵尺度的影響：
  - 不同尺度的特徵會影響距離計算，導致某些特徵的影響力被過度放大。因此，通常需要進行特徵縮放（如標準化或歸一化）。
- 維度詛咒：
  - 在高維度空間中，資料點之間的距離會變得均勻，導致「最近鄰居」的概念失去意義，模型效能會下降。

- 適用情境
  - 資料量適中且特徵維度不高：
    - 在資料量不大且維度不高時，KNN 表現通常不錯。
  - 決策邊界複雜但局部性強：
    - KNN 可以捕捉非線性的決策邊界，只要這些邊界在局部是清晰的。
  - 無需訓練階段（惰性學習）：
    - 模型訓練速度快、因為訓練階段基本上只是儲存資料。
    - 所有計算發生在預測階段。
  - 多類別分類問題：
    - KNN 可以直接應用於多類別分類，無需額外擴展。
- 使用限制
  - 計算成本高：
    - 由於每次預測都需要計算新樣本與所有訓練樣本的距離，當訓練資料量非常大時，預測速度會非常慢。
  - 對離群值敏感：
    - 少數離群值可能會影響 K 個最近鄰居的選擇，進而影響分類結果。
  - 需選擇合適的 K 值：
    - K 值的選擇對模型效能影響很大，太小容易受雜訊影響，太大則可能模糊類別邊界。通常需要透過交叉驗證來選擇最佳 K 值。

- 對高維度資料表現不佳（維度詛咒）：
  - 隨著特徵維度的增加，資料點之間會變得稀疏，距離差異減小，導致 KNN 的效能急劇下降。
- 對特徵尺度敏感：
  - 缺乏特徵縮放會導致結果偏差。

### （5）樸素貝式分類

- 定義
  - 樸素貝式分類（Naïve Bayes Classifier）是一種基於貝式定理（Bayes' Theorem）並假設特徵之間彼此條件獨立的機率分類演算法。
  - 之所以被稱為「樸素」，是因為簡化地假設所有用於預測的特徵彼此獨立，儘管這在現實情況中往往不完全成立，但這種簡化讓模型計算高效且容易實現。
  - 即便獨立假設未必完全符合真實世界，Naïve Bayes 在許多應用，特別是文本分類、垃圾郵件過濾、情感分析等領域，仍能展現出良好的效果，並以快速且穩定的特點受到廣泛使用。
- 模型公式
  - 樸素貝式分類器根據貝式定理計算一個樣本屬於某個類別的機率，然後將該樣本分配給機率最高的類別。貝式定理的公式如下：

$$P(C|X) = \frac{P(X|C) \times P(C)}{P(X)}$$

    - 其中：
      - P（C｜X）：在給定特徵 X 的情況下，樣本屬於類別 C 的後驗機率（Posterior Probability）。這是模型想要預測的目標。
      - P（X｜C）：在樣本屬於類別 C 的情況下，觀察到特徵 X 的似然度（Likelihood）。
      - P（C）：樣本屬於類別 C 的先驗機率（Prior Probability）。

- P（X）：觀察到特徵 X 的證據（Evidence）或邊緣機率。

- 模型假設
  - 特徵條件獨立性（Conditional Independence of Features）：
    - 「樸素」的假設：所有特徵在給定類別的情況下是相互獨立的。在現實世界中這個假設很少完全成立，但即便如此，樸素貝式在許多情況下仍能提供穩健的性能。
    - 若 X=（$x_1, x_2, \ldots, x_n$）是一組特徵

    則：$P(X|C) = P(x_1|C) \times P(x_2|C) \times \ldots \times P(x_n|C)$
    - 在這樣的獨立性假設下，大幅簡化了計算。因為我們只需要計算每個特徵與類別之間的條件機率，而不需要考慮特徵之間的複雜交互關係。
- 模型評估指標
  - Accuracy（準確率）
  - Precision（精確率）
  - Recall（召回率）
  - F1-score
  - ROC-AUC
- 適用情境
  - 文本分類（Text Classification）：
    - 如垃圾郵件過濾、情感分析、新聞分類等，儘管單詞之間並非完全獨立，但樸素貝式在這些任務中表現出色。
  - 大規模數據集：
    - 由於計算效率高且訓練速度快，非常適合處理大型數據集。
  - 多類別分類問題：
    - 可以直接應用於多類別分類任務。
  - 即時預測：
    - 預測階段的計算量小，因此適合需要快速響應的應用。

- 特徵數量多：
  - 即使特徵數量很多，只要滿足獨立性假設（或接近滿足），模型仍能有效運作。

- 使用限制
  - 強獨立性假設：
    - 這是其最大的局限性。如果特徵之間存在高度相關性，模型的性能可能會受到影響。
  - 「零機率」問題：
    - 如果訓練集中某個類別的某個特徵值從未出現過，那麼其條件機率將為零，導致整個後驗機率為零，即使該樣本在測試時出現。這通常透過拉普拉斯平滑（Laplace Smoothing）或其他平滑技術來解決。
  - 對輸入數據的敏感性：
    - 模型對輸入數據的分佈假設（例如高斯分佈）很敏感，如果數據不符合這些假設，性能會下降。
  - 不適用於迴歸問題：
    - 樸素貝式本身是一個分類演算法，不能直接用於預測連續值。

### （6）集成式分類

集成式分類（Ensemble Classification）是將多個模型的預測結果整合，提升分類準確度與穩定性的技術。其核心概念是「群體智慧通常勝過單一模型」。

在前面的「迴歸任務」中，已經介紹過集成演算法如 Random Forest、Gradient Boosting、XGBoost、LightGBM 等。這些演算法同樣可以應用在分類問題上，只是演算法內部在處理類別預測時，邏輯會略有不同。

- 在分類問題中：
  - 每個基模型預測的結果是類別標籤或類別機率。
  - 最終結果通常透過投票機制或機率平均產生。

- 而集成式分類的核心策略分為兩大類：
  - Bagging（Bootstrap Aggregating）
    - 隨機抽樣多個訓練子集，建立多個獨立模型，最終結果採多數決（Voting）。
    - 重點在降低模型的變異性（Variance）。
    - 代表演算法：Random Forest。
  - Boosting
    - 逐步建立一系列模型，每個新模型都針對前一個模型的錯誤進行修正，最後加權整合所有模型結果。
    - 重點在降低偏差（Bias）。
    - 代表演算法：Gradient Boosting、XGBoost、LightGBM。
- 集成式分類器：分類任務的決策機制與關鍵考量

  集成式學習透過結合多個獨立學習器（通常稱為基學習器或弱學習器）的預測結果，來提升模型整體性能、穩定性與泛化能力。在分類任務中，其獨特的協作與決策機制尤為突出。
  - 投票機制（Voting）
    - 在分類任務中，每棵樹或每個弱模型都會產生類別預測。
    - 最終的類別是由多數票決定，或是加權投票決定。
  - 機率輸出
    - 有些方法（如 Gradient Boosting、XGBoost、LightGBM）可以輸出每個類別的預測機率，而不是單純類別。
  - 多類別處理
    - 在分類問題，集成方法常需額外支援多類別策略，如：
      - 一對多策略（One-vs-Rest, OvR）
      - Softmax 多類別概率輸出
      - 一對一策略（One-vs-One, OvO）

- 分類評估指標：
  - Accuracy（準確率）
  - Precision（精確率）
  - Recall（召回率）
  - F1-score
  - ROC-AUC
- 常見集成分類器
  - 隨機森林分類器
    - 介紹
      - 隨機森林分類器（Random Forest Classifier）是集成方法中最具代表性的一種。透過建立多棵決策樹，並在每棵樹上隨機抽樣資料與特徵進行訓練，最後以多數投票（Voting）決定最終的分類結果。
      - 這種集成策略有助於減少單一決策樹容易出現的過度擬合問題，並提高模型的穩定性和準確性。
      - 在分類情境中：
        - 每棵樹預測一個類別。
        - 最終結果為票數最多的類別。
    - 特點
      - 抗過擬合能力強。
      - 可處理非線性資料與變數交互作用。
      - 對於不平衡資料稍具抵抗力。
      - 可計算特徵的重要性，協助特徵篩選。
      - 訓練可平行化，效率較高。
    - 缺點
      - 模型整體解釋度較低，屬於黑箱模型。
      - 模型體積大，需較多記憶體。

- 對極端不平衡資料仍可能偏向多數類別。
- 在需要精確概率輸出的應用（如醫療診斷機率）上，較無法提供平滑的機率估計。

◆ 梯度提升分類器

■ 介紹

- 梯度提升分類器（Gradient Boosting Classifier）是一種集成方法，屬於 Boosting 類演算法。其概念是：
  - 逐步建立多棵決策樹。
  - 每棵樹都專門修正前一棵樹的錯誤預測。
  - 最終透過加權整合所有弱學習器的結果。
- 在分類任務中：
  - 每棵樹預測殘差（Residual），殘差轉換為機率分數，再進行分類。
  - 尤其在二元分類時，常搭配 Logloss 作為目標函數。

■ 特點

- 預測準確度通常高於隨機森林。
- 能捕捉複雜的非線性關係。
- 支援多類別分類。
- 可自訂損失函數，靈活性高。
- 支援 Early Stopping，減少過擬合。

■ 缺點

- 訓練時間較長，尤其在樹多、深度大的情況下。
- 容易過擬合，需要謹慎調參。
- 模型解釋度差。
- 超參數多。

- XGBoost、LightGBM
  - 介紹
    - XGBoost 與 LightGBM 是 Gradient Boosting 的高效實作版本，專門解決大數據與計算瓶頸。
    - 在分類任務中：
      - 同樣透過疊加多棵樹進行預測。
      - 支援二元或多類別分類。
      - 能輸出每個類別的機率分佈（Softmax）。
  - XGBoost
    - 使用二階導數資訊（Hessian）加速分裂。
    - 支援缺值自動處理。
    - 層級式生長（Level-wise）。
  - LightGBM
    - 採用 Leaf-wise 分裂策略，能快速降低 Loss。
    - 支援 Histogram-based 計算。
    - 在大數據上速度更快。
  - 特點
    - 計算速度快。
    - 適用於大數據、高維度特徵。
    - 支援並行計算。
    - 提供特徵重要性分析。
    - 準確度高，在 Kaggle 競賽常勝軍。
  - 缺點
    - 超參數多，需要細心調整。
    - 模型解釋度差。
    - 記憶體需求仍可能高（尤其是 Leaf-wise）。
    - Leaf-wise 策略若不設限，容易造成過擬合（LightGBM 特有問題）。

## 4. 非監督式學習

在機器學習的廣泛領域中，若說監督式學習是「有老師指導的學習」，那麼非監督式學習便是「自主探索與發現」。是在沒有預先標註資料的情況下，從資料中發掘隱含模式、結構或關係的技術。其核心目標不是預測一個已知的結果，而是要呈現數據本身的內在規律，讓電腦能夠自動理解、組織和簡化複雜的資訊。

非監督式學習在許多現實應用中扮演著不可或缺的角色，例如：

- 顧客分群：找出不同消費行為的群體，以便進行精準行銷。
- 數據壓縮與視覺化：將高維度數據降維，使我們能以直觀方式理解其複雜結構。
- 異常行為偵測：辨識出網路流量中的惡意攻擊或設備故障。
- 關聯性分析：發現商品間的潛在銷售機會，例如「買尿布的通常也會買啤酒」。

不同的非監督式演算法，各自擁有獨特的數學基礎、優勢與限制。根據資料的特性、隱含模式的複雜度，以及最終的應用目標，選擇最合適的非監督式學習方法，是從未標註資料中提取重要洞察的關鍵。

### （1）分群分析

分群分析（Clustering）是將資料點依據其相似性自動分組的技術，目標是讓同一群組內的資料點彼此相似，而不同群組的資料點則彼此相異，用於發現資料中自然的群體結構。

- k-means 分群
  - 定義：
    - k-means 是一種迭代式的分群演算法，將 N 個資料點分到預設的 K 個群集中，使每個資料點都屬於離它最近的中心點（質心）所在的群集。
  - 步驟：

a. 初始化

- 隨機選取 K 個資料點作為初始質心（Centroid）。

b. 分配

- 將每個資料點分配到離其最近的質心所在的群集。

c. 更新

- 重新計算每個群集內所有資料點的平均值，將其作為新的質心。

d. 重複

- 重複步驟 b.和 c.，直到質心不再顯著移動，或達到最大迭代次數。

- 優點：
  - 簡單、快速、易於理解和實現。
  - 在處理大量數據時效率高。
- 缺點：
  - 需要預先指定 K 值。
  - 對初始質心的選擇敏感。
  - 對離群值敏感。
  - 只能形成球形或凸形群集，無法處理形狀不規則的群集。

- 階層式分群
  - 定義：
    - 階層式分群（Hierarchical Clustering）建立一個巢狀的群集序列，透過持續地合併（凝聚式，Agglomerative）或分裂（分裂式，Divisive）群集，最終形成一個樹狀結構（樹狀圖，Dendrogram），視覺化地呈現資料點的相似性層次。
  - 原理：
    - 凝聚式（Agglomerative Hierarchical Clustering）
      - 由下而上（bottom-up）的方法。
      - 從將每個資料點視為獨立群集開始，每一步迭代都將最相似的兩個群集合併，直到所有資料點最終匯聚成一個單一的大群集。

- 分裂式（Divisive Hierarchical Clustering）
  - 由上而下（top-down）的方法。
  - 從包含所有資料點的一個單一大群集開始，每一步迭代都將其中一個群集分解為更小的子群集，直到每個資料點最終都自成一群。

◆ 優點：

- 無需預先指定 K 值。
- 生成的樹狀圖提供豐富的視覺化，有助於理解不同層次的分群結構。
- 能發現不同粒度或層次的群集。

◆ 缺點：

- 計算複雜度高，尤其是處理大量數據時效率低下。
- 對離群值敏感。
- 一旦群集被合併或分裂，該操作無法撤銷，可能導致次優解。

• DBSCAN

◆ 定義：

- DBSCAN( Density-Based Spatial Clustering of Applications with Noise ) 是一種基於密度的分群演算法。
- DBSCAN 能夠辨識任意形狀的群集，並且能自動將噪聲點（離群值）從群集中分離出來。

◆ 原理：

a. 參數定義，設定兩個關鍵參數：
  - Eps（epsilon）：鄰域半徑，定義一個點周圍的搜索範圍。
  - MinPts（Minimum Points）：形成一個核心群集所需的最小點數。

b. 核心點辨識：
  - 從任意一個未被訪問的點開始，檢查其 Eps 半徑鄰域內的點數量。

- 如果鄰域內的點數 ≥MinPts，則該點被定義為核心點。從這個核心點開始，生成一個新的群集，並遞歸地將所有從核心點「密度可達」的點（包括其他核心點和邊界點）添加到該群集。
- 如果鄰域內的點數<MinPts，則該點暫時被標記為噪聲點或邊界點。

c. 重複：

- 持續這個過程，直到所有資料點都被訪問並歸類。

◆ 優點：

■ 無需預先指定群集數量。

■ 能夠發現任意形狀的群集，不受凸性限制。

■ 能有效辨識並標記噪聲點。

◆ 缺點：

■ 對於參數 Eps 和 MinPts 的選擇高度敏感，參數設定不當會嚴重影響分群結果。

■ 處理密度不均勻的群集效果不佳。

■ 對於高維數據，定義合適的「密度」和距離度量可能變得困難，表現可能不理想。

**（2）降維技術**

降維技術（Dimensionality Reduction）是將高維度數據轉換為低維度表示的方法，同時盡可能保留原始數據中的重要資訊和結構。這項技術對於數據視覺化、減少儲存空間、加速模型訓練，以及緩解「維度詛咒」（在高維空間中數據變得稀疏，導致分析困難）問題至關重要。

● 主成分分析（PCA）

◆ 定義：

■ 主成分分析（Principal Component Analysis, PCA）是一種廣泛使用的線性降維方法。透過正交變換，將原始數據投影到一組新的、不相關的座標軸上，這些新軸被稱為主成分。

- 每個主成分都是原始特徵的線性組合，且代表了數據中最大的變異量方向。

- 原理：
  - 數據標準化：
    - 如果特徵尺度差異大，通常會先對數據進行標準化。
  - 協方差矩陣計算：
    - 計算原始數據的協方差矩陣，該矩陣反映了各特徵之間的變異程度和協同關係。
  - 特徵值分解：
    - 對協方差矩陣進行特徵值分解，得到一組特徵值和對應的特徵向量。
  - 主成分選擇：
    - 特徵向量定義了主成分的方向，而特徵值則表示該主成分所解釋的變異量大小。選擇具有最大特徵值的 K 個特徵向量，作為新的 K 個主成分。
  - 數據投影：
    - 將原始數據投影到由這 K 個主成分定義的新的低維空間中。
- 優點：
  - 數學基礎堅實、概念相對易於理解和實作。
  - 能有效降低資料維度，並去除特徵間的冗餘資訊
  - 轉換後的特徵彼此正交。
- 缺點：
  - 只能捕捉線性關係，對於非線性結構的數據效果不佳。
  - 主成分是原始特徵的線性組合，缺乏較直觀可解釋性。
  - 對離群值敏感，因為離群值會顯著影響協方差矩陣計算。

- t-SNE
  - 定義：

- t-SNE（t-distributed Stochastic Neighbor Embedding）是非線性降維方法，尤其專為高維數據的視覺化而設計。
- t-SNE 的核心目標是將高維空間中的相似點映射到低維空間中也彼此靠近，同時將不相似的點映射到低維空間中也彼此遠離，從而呈現複雜的群集結構。

◆ 原理：

- 高維相似度計算：
  - 計算高維空間中資料點之間的機率相似性分佈（通常使用高斯分佈）。
- 低維相似度計算：
  - 在低維空間（例如 2D 或 3D）中，隨機初始化點的座標，並計算其相似性機率分佈（使用 t-分佈，以更好地解決「擁擠問題」）。
- 優化映射：
  - 使用 Kullback-Leibler（KL）散度來衡量高維和低維相似度分佈之間的差異。透過梯度下降等優化演算法，不斷調整低維點的座標，以最小化 KL 散度，使低維表示能最佳地反映高維數據的相似性結構。

◆ 優點：

- 能很好地保留局部結構，適合高維數據的視覺化，用於呈現傳統線性方法難以發現的複雜非線性關係和群集結構。

◆ 缺點：

- 計算成本高昂，不適合處理非常大的數據集（通常限制在數萬個樣本）。
- 結果可能受隨機初始化和隨機過程的影響，每次運行結果略有不同。
- 參數選擇（如「困惑度」Perplexity）對結果有較大影響，需要細緻調整。

- 主要用於視覺化，不適合作為下游機器學習任務的預處理（因為映射是非線性的，無法輕易應用到新數據）。

- UMAP
  - 定義：
    - UMAP（Uniform Manifold Approximation and Projection） 是相對較新的高性能非線性降維方法。其目標與 t-SNE 類似，都是在高維數據中尋找低維嵌入，以同時保留局部和全局結構。
  - 原理：
    - 構建高維模糊拓撲：
      - 在高維空間中，UMAP 構建一個表示數據點之間局部連通性的「模糊拓撲結構」（通過定義每個點的鄰域和它們之間的連接強度）。
    - 構建低維模糊拓撲：
      - 在低維目標空間（例如 2D 或 3D）中，建立一個類似的模糊拓撲結構。
    - 最小化交叉熵：
      - 透過優化算法，最小化高維和低維拓撲結構之間的交叉熵，以確保低維表示能最佳地近似高維數據的連通性。
  - 優點：
    - 通常比 t-SNE 運行顯著更快，能夠處理更大的數據集。
    - 在保留局部和全局結構方面表現優異。
    - 參數調整比 t-SNE 更直觀，更容易達到穩定和有意義的結果。
    - 支持增量學習。
  - 缺點：
    - 相對較新，其理論基礎更為複雜。
    - 理解門檻較高。
    - 產出結果仍可能受某些參數影響。

### （3）關聯規則學習

關聯規則學習（Association Rule Learning）旨在從大型資料集中發現不同項目之間有趣的、非平凡的關係或關聯。最典型的應用是「購物籃分析」，透過分析顧客的購買行為，找出哪些商品經常被一起購買。

- Apriori 演算法
  - 定義：
    - Apriori 是一種經典且基礎的關聯規則挖掘演算法，透過迭代和剪枝的方式，從資料集中找出所有滿足預設最小支持度（Minimum Support）的頻繁項目集（Frequent Itemset），然後從這些頻繁項目集中生成滿足預設最小信賴度（Minimum Confidence）的關聯規則。
  - 原理：
    - Apriori 演算法的核心是其著名的「Apriori 性質」：
      - 如果一個項集是頻繁的，那麼它的所有非空子集也一定是頻繁的。
      - 反之，如果一個項集不頻繁，那麼包含它的任何超集也一定不頻繁。這個性質減少了需要搜索的空間。
    - 步驟：
      - 生成頻繁 1-項集：首次掃描資料集，統計每個單一項目的頻率，並篩選出所有滿足最小支持度的 1-項集。
      - 迭代生成候選集與剪枝：使用頻繁的 k−1 項集來生成候選的 k 項集。然後，根據 Apriori 性質，剪枝掉所有包含不頻繁子集的候選 k 項集。
      - 頻率計數與篩選：再次掃描資料集，計算剩餘候選 k 項集的頻率，並篩選出頻繁 k 項集。
      - 重複：不斷重複步驟 2 和 3，直到沒有新的頻繁項目集生成為止。
      - 生成關聯規則：從所有已發現的頻繁項目集中，生成滿足最小信賴度的關聯規則。

- 衡量標準：
  - 支持度（Support）：
    - 規則 A⇒B 在總交易中發生的頻率，即同時包含 A 和 B 的交易數佔總交易數的比例。數學上表示為 P（A∩B）。
  - 信賴度（Confidence）：
    - 在包含 A 的交易中，同時也包含 B 的條件機率。
    - 衡量了規則的可靠性。
    - 數學上表示為

      P（B｜A）= Support（A∪B）/Support（A）。
  - 提升度（Lift）：
    - 規則 A⇒B 的強度，以及 A 和 B 之間相關性的指標。
    - 衡量 A 的出現對 B 的出現機率的影響。
    - Lift（A ⇒ B）= Confidence（A ⇒ B）/ P（B）。
    - 若 Lift > 1 表示 A 和 B 正相關；

      若 Lift < 1 表示負相關；若 Lift = 1 表示獨立。
- 優點：
  - 概念簡單、易於理解；能夠發現資料中有意義的、非平凡的關聯模式。
- 缺點：
  - 在處理大量數據或項數非常多（高維稀疏數據）時，其計算成本高昂且效率低下，因為需要多次掃描資料集來生成和檢驗大量的候選集。

- FP-Growth
  - 定義：
    - FP-Growth 是比 Apriori 更高效的關聯規則挖掘演算法。主要優勢在於無需生成大量的候選集，而是透過構建一個稱為 FP 樹（Frequent Pattern Tree）的緊湊資料結構來挖掘頻繁項目集的，從而大幅降低計算量。

- 原理：
  a. 第一次掃描：
     - 遍歷整個交易資料集，統計每個項目的支持度。
  b. 過濾與排序：
     - 移除所有支持度低於最小支持度的項，然後根據項目的支持度對剩餘的頻繁項進行降序排序。
  c. 構建 FP 樹：
     - 第二次掃描資料集。
     - 將每個交易映射到一條路徑上，並將這些路徑構建成一棵 FP 樹。
     - 樹中的每個節點代表一個頻繁項，其計數表示該項在路徑上出現的次數。
  d. 遞歸挖掘：
     - 從 FP 樹中，透過遞歸地構建條件模式基（Conditional Pattern Base）和條件 FP 樹（Conditional FP Tree），直接從樹中提取所有頻繁項目集。
     - 這個過程不需要生成中間候選集。
- 優點：
  - 相較於 Apriori 顯著更高效，尤其在處理大型資料集和稠密數據時表現優異。
  - 不需要生成候選集，減少了不必要的計算和記憶體消耗。
  - 只需兩次掃描資料集即可完成頻繁項目集的挖掘。
- 缺點：
  - 構建 FP 樹在處理非常龐大且複雜的資料集時可能需要大量記憶體。
  - 對於極度稀疏的資料集，FP 樹可能無法顯著壓縮，其優勢會減弱。

### （4）異常偵測

異常偵測（Anomaly Detection）或稱離群值偵測（Outlier Detection）是一種辨識資料集中與大多數資料行為顯著不同的模式或資料點的技術。這些「異常」

或「離群值」往往具有重要的意義，可能代表著錯誤、欺詐、設備故障、網路入侵，或新穎的、未曾見過的事件。

- Isolation Forest（孤立森林）
  - 定義：
    - Isolation Forest 是一種基於樹的非參數異常偵測演算法。
    - 核心思想基於一個直觀的觀察：異常點通常是少數的，並且與正常點在特徵空間中相距較遠。因此，在隨機劃分數據時，異常點更容易被快速「孤立」出來（即只需要很少的分割就能將其分開）。
  - 原理：

    a. 構建孤立樹（iTree）：
    - 隨機選擇一個特徵，並在該特徵的最小值和最大值之間隨機選擇一個分割點。

    b. 遞歸劃分：
    - 根據這個分割點將數據集分成兩部分，並對這兩部分數據遞歸地重複上述步驟。這個過程持續進行，直到每個點都被單獨孤立，或者達到預設的最大樹深。

    c. 異常分數計算：
    - 異常點因為其孤立性，通常在 iTree 中具有較短的路徑長度（很快就被孤立在樹的一個葉節點上）。

    d. 集成結果：
    - 透過構建多棵隨機生成的「孤立樹」（形成森林），並計算每個點在所有樹中的平均路徑長度，從而得出一個標準化的異常分數。分數越高，越可能是異常點。
  - 優點：
    - 效率高，尤其適合處理高維數據和大規模數據集（因為只關注將點孤立，而非計算距離或密度）。
    - 不需要任何距離度量，在高維空間中具有計算優勢。

- 能直接輸出異常分數，便於閾值設定和排序；對不平衡數據集（異常點總是非常少數）表現良好。

◆ 缺點：

- 對於非常接近正常點群集的異常值可能難以有效檢測。
- 如果數據集中正常點與異常點混雜在一起，或者異常點不是特別「孤立」，效果可能不佳。
- 結果可能受隨機性影響。

• One-Class SVM

◆ 定義：

- One-Class SVM（One-Class Support Vector Machine）是一種用於單類別分類的演算法，屬於邊界學習的方法。
- 與傳統 SVM 不同，One-Class SVM 只學習一個類別（通常是正常數據）的模式，並構建一個能將大多數正常樣本包圍起來的決策邊界。任何落在這個邊界之外的資料點都被判定為異常。

◆ 原理：

a. 映射高維空間：

○ 利用核函數（如 RBF 核），將所有的訓練數據（假設這些都是正常數據）非線性地映射到一個高維特徵空間。

b. 尋找超平面：

○ 在這個高維空間中，One-Class SVM 試圖找到一個能夠將所有正常數據點與原點（或從原點發出的向量）最大程度分離的超平面。

c. 定義決策邊界：

○ 這個超平面在原始數據空間中的投影，就定義了正常數據的邊界。

d. 異常判斷：

○ 任何新的資料點，如果其在映射後的高維空間中落在這個邊界之外，就被視為異常。

- 優點：
  - 能夠透過核函數處理非線性邊界，使其適用於各種複雜數據分佈。
  - 對於高維數據表現良好。
  - 只需要正常數據進行訓練，在異常點極難收集的場景下是優勢。
- 缺點：
  - 對於參數（如核函數的參數和異常率參數）的選擇高度敏感，這些參數直接影響模型的效能和異常檢測的嚴格程度。
  - 計算成本相對較高，不適合處理非常大的數據集。
  - 如果訓練數據中意外地混入了少量異常值，模型可能會受到這些異常值的影響，導致邊界不夠精確。

## 重點掃瞄

# 4.3 深度學習原理與框架

### 1. 前言與章節導覽

在人工智慧的發展浪潮中，深度學習（Deep Learning）扮演了極其關鍵的角色。相較於傳統機器學習依賴手工特徵工程，深度學習透過多層神經網路結構，能自動學習數據中的高階抽象特徵，大幅提升在影像辨識、語音處理、自然語言理解及生成式 AI 等多領域的表現。

深度學習不僅推動了技術創新，也成為企業數位轉型與智慧應用的重要基礎。而深度學習模型本身的複雜度與龐大的運算需求，也帶來許多實務挑戰，讓掌握其原理、架構與工具框架，成為 AI 領域從業人員不可或缺的核心能力。

本節將循序介紹深度學習的基礎概念、常見模型架構，以及主流開發框架，協助學習者建立完整的深度學習知識體系。

### 2. 深度學習基本概念

#### （1）人工神經元與感知器概念

人工神經元（Artificial Neuron）是構成深度學習模型的基礎單元，而感知器（Perceptron）則是人工神經元的一個早期且簡化的實例，為後來更複雜、功能更強的類神經網路奠定了基礎。

- 人工神經元（Artificial Neuron）
  - 定義：
    - 人工神經元是受生物神經元啟發而設計的數學模型。
    - 接收多個輸入訊號，對這些訊號進行加權求和，然後透過一個激活函數（Activation Function）產生一個輸出訊號。

- 運作原理：
  a. 輸入（$x_1, x_2, \ldots, x_n$）：
     - 接收來自其他神經元或外部數據的資訊。
  b. 權重（$w_1, w_2, \ldots, w_n$）：
     - 每個輸入訊號都與一個對應的權重相關聯。
     - 權重代表了該輸入訊號的重要性。
  c. 加權求和（Weighted Sum）：
     - 將所有輸入訊號與其對應的權重相乘，進一步求和。
     - 此外，還會加上一個偏置項（Bias, b），允許神經元在沒有任何輸入訊號時也能被激活，或者調整激活的閾值。
     - 數學表達式為：

$$Z = w_1x_1 + w_2x_2 + \cdots + w_nx_n + b = \sum_{i=1}^{n} w_1x_1 + b$$

  d. 激活函數（Activation Function）：
     - 加權求和的結果 Z 會被輸入到一個非線性的激活函數中，產生最終的輸出。
     - 激活函數決定了神經元是否被「激活」以及激活的程度。沒有激活函數，多層神經網路就等同於單層網路，無法學習複雜的非線性關係。
     - 數學表達式為：

$$Output = A(Z)$$

     - 常見激活函數：
       - 例如 ReLU、Sigmoid、Tanh。

- 感知器（Perceptron）
  - 定義：
    - 感知器是最簡單、最早提出的人工神經元模型之一，由 Frank Rosenblatt 於 1957 年提出。

- 感知器是一種二元分類器，能夠學習如何將輸入數據劃分為兩個類別。
- 感知器可以被視為一種特定類型的人工神經元，通常使用步階函數（Step Function）作為其激活函數。

◆ 運作原理：

- 輸入與加權求和：
  - 感知器與人工神經元相同，先接收輸入，並進行加權求和，公式示意為：

$$\left(Z = \sum_{i=1}^{n} w_1 x_1 + b\right)$$

- 步階激活函數（Step Function）：
  - 步階激活函數是一種二元輸出的函數。會將輸入值與一個預設的閾值（threshold）進行比較。
  - 最常見的步階函數形式如下（當閾值為 0 時）：

$$f(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$$

- 學習過程（權重更新）：
  - 感知器透過迭代調整權重和偏置來學習。
  - 如果在給定輸入下預測錯誤，感知器會根據錯誤的大小和方向來微調權重，以減少未來的錯誤，此過程被稱為感知器學習規則。

◆ 優勢與限制：

- 優勢：
  - 概念簡單，易於理解和實現。
  - 能夠解決線性可分（Linearly Separable）的分類問題。
- 限制：
  - 無法解決非線性可分的問題，例如著名的 XOR（互斥或）問題；也因此，單個感知器無法構成更複雜、功能性更強的模型。

### （2）激活函數的角色

在人工神經元和整個神經網路中，激活函數（Activation Function）扮演著核心角色。可以將激活函數想像成神經元的「守門員」或「決策者」。

如果說神經元的加權求和部分只是將所有輸入資訊的初步整合，那麼激活函數就是決定這個神經元是否要被「激活」並向下一層傳遞訊號的關鍵，同時也負責引入神經網路學習複雜模式所需的非線性能力。

以下是激活函數的主要功能：

- 引入非線性（Introduce Non-linearity）
  - 激活函數最根本、最重要的作用。若沒有激活函數，或只使用線性激活函數，那麼無論神經網路有多少層，其最終輸出都只是輸入的線性組合。這將導致多層網路的能力等同於單層線性模型，僅能解決線性可分（Linearly Separable）的問題。
  - 現實世界中的絕大多數複雜問題（如圖像、語音辨識）本質上都是非線性的。透過引入非線性激活函數，神經網路才能夠學習並逼近任意複雜的非線性函數關係，從而處理並理解更為複雜的數據模式和結構。
- 決定神經元的「激活」狀態
  - 激活函數接收神經元的加權求和結果（淨輸入），並依據此值決定該神經元的最終輸出。
  - 激活函數模擬了生物神經元在接收到足夠刺激時才會「發射」信號的行為。只有當淨輸入滿足一定條件時，神經元才會被「激活」並將訊號傳遞到下一層。舉例如：
    - Sigmoid 函數：
      - 將輸出壓縮至 0 到 1 之間，可視為激活強度或機率。
    - ReLU 函數：
      - 輸入為正時，直接輸出，為負時輸出 0，實現稀疏激活。

- 壓縮輸出範圍（Compress Output Range）
  - 某些激活函數（如 Sigmoid 和 Tanh）能將神經元的輸出值壓縮到一個特定的範圍內（如[0, 1]或[-1, 1]）。這有助於：
    - 穩定訓練：
      - 避免數值在網路傳播過程中過大或過小，進而導致梯度爆炸或梯度消失問題。
    - 解釋性：
      - 在輸出層，壓縮後的範圍有時可直接解釋為機率。

**（3）前向傳播與反向傳播原理**

神經網路的學習過程，就像人類學習新技能一樣，是一個不斷嘗試、犯錯、然後從錯誤中汲取經驗並改進的循環。在這個循環中，前向傳播（Forward Propagation）和反向傳播（Backpropagation）是兩個核心且密不可分的步驟，共同驅動著模型從數據中提取知識並提升預測能力。前向傳播產生預測，反向傳播負責修正誤差，兩者合力使神經網路具備自我學習能力，是深度學習的核心。

- 前向傳播（Forward Propagation）
  - 目標描述：
    - 前向傳播 → 做預測。
    - 把輸入數據依序送進每一層神經網路，經過計算後，產生模型的預測結果。
    - 就像把數據「往前推」到輸出層，看看模型對這筆資料怎麼判斷。
  - 定義：
    - 前向傳播是神經網路進行預測的運算過程。
  - 運作原理：前向傳播是神經網路進行預測的過程，模擬了資訊從輸入端流向輸出端的路徑。

a. 數據輸入：
   - 原始數據（例如圖片像素、文本特徵等）作為輸入，進入神經網路的第一層（輸入層）。

b. 層層計算：
   - 每個輸入的訊號會與其對應的權重相乘，然後加上偏置項，形成一個加權和。這個加權和接著會通過該層的激活函數，產生該神經元的輸出。

c. 訊息傳遞：
   - 這些輸出再作為下一層神經元的輸入，重複加權求和與激活函數的運算。這個過程一層一層地向前推進，直到資訊到達最後一層（輸出層）。

d. 輸出結果：
   - 輸出層的神經元會產生最終的預測結果。對於分類任務，這可能是一個類別標籤或每個類別的機率；對於迴歸任務，這可能是一個連續的數值。

- 反向傳播（Backpropagation）
  - 目標：
    - 反向傳播 → 調整參數。
    - 比較模型的預測結果與真實答案，計算出誤差後，把誤差「往回傳」到每一層，告訴各層權重該怎麼微調，才能讓下一次預測更準。
    - 像在告訴模型：哪裡錯了，要怎麼修。
  - 定義：
    - 反向傳播是神經網路「學習」的關鍵步驟。目標是計算：
      - 損失函數相對於每個權重的梯度。
      - 進而更新權重，讓模型預測更準確。

- 運作原理：反向傳播是神經網路進行學習和參數調整的過程，運用微積分的連鎖律（Chain Rule），將預測誤差從輸出端反向傳遞至網路的每一層，以便更新權重和偏置。
  a. 計算損失：
     - 在前向傳播得到預測結果後，我們將其與真實標籤進行比較，使用損失函數計算出一個量化的誤差值（即損失）。
     - 這個損失值代表了模型預測的「錯誤程度」。
  b. 計算梯度：
     - 計算梯度是反向傳播的核心任務，也是神經網路得以「學習」的關鍵。這個過程旨在準確地找出如何調整模型的內部參數，讓錯誤降到最低。
     - 梯度的意義：
       - 想像在一個佈滿山丘的陌生土地上，目標是找到最低點（也就是損失函數的最小值）。梯度就像是一個微型的「指南針」和「坡度計」的結合體。會告訴你當前所處位置的「坡度」有多陡。並明確指出「哪個方向是下坡，且下降最快」。
       - 在神經網路中，梯度就是損失函數對於每個權重（Weights）或偏置（Biases）的導數。代表著「如果稍微改變某個權重或偏置，損失函數會如何變化（是增加還是減少）、哪個方向的調整，能最快、最有效地降低損失」。
     - 反向傳播中的梯度計算：
       - 梯度的計算過程是從神經網路的「輸出層開始，並逐步反向傳遞到輸入層」。這個機制，正是透過微積分中的連鎖律（Chain Rule）來實現的。
       - 每一層會接收到來自其「後方」層次的梯度資訊，然後利用這些資訊（以及該層自身的計算，特別是激活函數的導數），來計算出屬於自己這一層權重和偏置的梯度，再將相關的梯度傳遞

給「前方」的層次。這樣，每個參數都能「知道」自己在導致最終錯誤中扮演的角色，以及該如何調整。

c. 權重更新：

- 獲得所有權重和偏置的梯度後，優化器（如梯度下降法）就會根據這些梯度和預設的學習率，來微調模型的權重和偏置。調整的方向總是沿著梯度相反的方向（也就是損失函數下降最快的方向）。
- 例如，如果某個權重導致損失增加，梯度會指示減少這個權重；如果導致損失減少，則指示增加這個權重。

d. 迭代學習：

- 前向傳播和反向傳播共同構成了一個訓練循環。神經網路會重複執行這兩個步驟數千甚至數百萬次，每一次循環都微調參數，使模型的預測越來越精準，損失值越來越小，直到模型達到滿意的效能。

**（4）損失函數與優化器**

在深度學習模型的訓練過程中，損失函數（Loss Function）和優化器（Optimizer）是相輔相成的核心組件，共同引導模型從錯誤中學習並不斷提升性能。

- 損失函數

  損失函數（Loss Function）是衡量模型預測結果與真實值之間誤差大小的標準。損失函數之輸出是單一的數值，值越小，表示模型的預測越接近實際情況，模型表現就越好。

  - 目的：
    - 評估誤差：
      - 損失函數充當了模型「表現好壞」的量尺，告訴我們當前的預測有多不準確。

- 指引優化：
  - 損失函數的數值直接為優化器提供依據，優化器的最終目標就是調整模型參數（權重和偏置），使損失函數的值達到最小。

- 根據不同的任務類型，我們會選擇不同的損失函數。例如：
  - 迴歸問題（預測連續值）：
    - 均方誤差（Mean Squared Error, MSE）：
      - 計算預測值與真實值之差的平方平均值。
    - 平均絕對誤差（Mean Absolute Error, MAE）：
      - 計算預測值與真實值之差的絕對值平均值，對極端值（離群值）的敏感度較低。
  - 分類問題（預測離散類別）：
    - 交叉熵損失（Cross-Entropy Loss）
      - 又稱為對數損失（Log Loss）：
      - 衡量模型預測的機率分佈與真實類別分佈之間的「距離」。
    - 二元交叉熵：用於判斷是或否的二元分類。
    - 類別交叉熵：用於多種不同類別的多分類問題。

**（5）過擬合與正則化技術**

在訓練深度學習模型的過程中，常常會面臨過擬合（Overfitting）的挑戰，導致模型在訓練資料上表現優異，卻無法有效應用於未知的新數據。為了抑制過擬合、提升模型的泛化能力，正則化技術（Regularization Techniques）應運而生，成為深度學習實務中不可或缺的重要工具。

- 過擬合

  過擬合（Overfitting）是指模型在訓練數據上表現得非常好，但在面對未見過的新數據（測試數據或實際應用中的數據）時，其表現卻顯著下降的現象。

- 發生原因：
  - 模型過於複雜：
    - 當模型的參數數量過多、網路層次過深，使其擁有過高的學習能力時，模型可能會開始「記憶」訓練數據中的噪聲和特有模式，而非學習其底層的普遍規律。
  - 訓練數據不足：
    - 如果訓練數據量相對於模型的複雜度來說太少，模型就無法從足夠多樣的例子中學習到通用的特徵。
  - 訓練時間過長：
    - 在某些情況下，即使模型和數據量都適中，訓練時間過長也可能導致模型過度適應訓練數據。
- 情況特徵：
  - 模型在訓練集上的損失很低，準確率很高。
  - 模型在驗證集（或測試集）上的損失高，準確率顯著低於訓練集。

- 正則化技術

  正則化技術（Regularization Techniques）的核心目標，是限制模型的複雜度，避免模型過度擬合訓練資料中的噪聲，從而提升在未見資料上的泛化能力。正則化技術就像一種「懲罰機制」，在模型學習的過程中，刻意對過大的參數或過度複雜的結構施加限制，藉此鼓勵模型專注於更簡潔、也更具普遍性的規律。

常見的正則化技術包括：

- L1 正則化（L1 Regularization / Lasso Regularization）
  - 在損失函數中加入「權重的絕對值總和」，促使部分權重縮小為零，達到特徵選擇的效果。
- L2 正則化（L2 Regularization / Ridge Regularization）
  - 在損失函數中加「權重平方和」，防止權重過大，讓模型更平滑、穩定。

- Elastic Net
  - 結合 L1 和 L2 正則化，兼具特徵選擇與權重平滑的優點，平衡兩種效果。
- Dropout
  - 在訓練過程中隨機屏蔽部分神經元，避免神經元彼此過度依賴，降低過擬合風險。
- 早停法（Early Stopping）
  - 在驗證集損失不再下降時提前停止訓練，以防模型在訓練集上過度擬合。

各項正則化技術的詳細原理、公式以及優缺點，請參考本指引第三章【4.3 正則化技術與模型穩定化】。

### 3. 深度學習模型架構

本小節進一步介紹常見的深度模型架構，這些架構是神經網路設計的基礎，每一種都針對特定的資料類型或任務進行優化。從專門處理結構化數據的多層感知器（Multilayer Perceptron, MLP）、擅長影像辨識的卷積神經網路（Convolution Neural Network, CNN）、能理解序列資訊的遞迴神經網路（Recurrent Neural Network, RNN），到近年在自然語言處理領域中大放異彩的 Transformer，以及具備生成能力、能創造新資料的生成式模型。

#### （1）多層感知器（Multilayer Perceptron. MLP）

- 定義
  - 多層感知器是最基礎也是最經典的深度學習模型架構之一，被視為前饋（Feedforward）神經網路的典型代表，也是許多更複雜神經網路模型的基礎。

- MLP 是一種由多層人工神經元組成的網路，其中層與層之間是全連接（Fully Connected Network）的（即每一層的神經元都與前一層的所有神經元相連），且資訊只能單向從輸入層流向輸出層，中間沒有任何迴圈或反饋。

- 模型結構

  MLP 的核心由至少三層構成：

  - 輸入層（Input Layer）：
    - 負責接收原始數據。
    - 這層的神經元數量通常與輸入數據的特徵數量相匹配。
    - 輸入層的節點通常不執行任何計算，只負責將數據傳遞給下一層。
  - 隱藏層（Hidden Layers）：
    - MLP 的核心，也是「深度」的來源。
    - MLP 可以包含一個或多個隱藏層。
    - 每個隱藏層的神經元都會接收來自前一層的所有輸出，進行加權求和，然後通過一個非線性激活函數產生輸出。
    - 隱藏層是模型學習數據中複雜模式和抽象特徵的地方。
    - 層數越多、每層神經元越多，模型的複雜度就越高，理論上能學習的模式也越複雜。
  - 輸出層（Output Layer）：
    - 產生模型的最終預測結果。
    - 輸出層的神經元數量取決於任務類型：
      - 二元分類：
        - 通常使用 1 個神經元，搭配 Sigmoid 激活函數輸出機率。
      - 多類別分類：
        - 神經元數量等於類別數量，搭配 Softmax 激活函數輸出每個類別的機率分佈。

- 迴歸問題：
  - 通常使用 1 個或多個神經元（取決於預測值的維度），通常不使用激活函數（或使用線性激活函數）。

- 運作原理
  - 前向傳播：
    - 輸入數據經過輸入層，逐層向前傳遞。
    - 在每一層，神經元進行加權求和，然後通過激活函數轉換。
    - 這個過程持續到輸出層，產生最終預測。
  - 反向傳播與優化：
    - 模型預測結果與真實值之間的誤差由損失函數計算。
    - 接著，這個誤差通過反向傳播演算法，計算出損失對所有權重和偏置的梯度。
    - 最後，優化器根據這些梯度調整模型的參數，以最小化損失。
  - 重複循環：
    - 前向傳播與反向傳播重複迭代進行，直到模型性能穩定或達到預設的訓練次數。
- 優點
  - 概念簡單，易於理解：
    - 作為神經網路的基礎，其結構和運作原理相對直觀。
  - 非線性建模能力：
    - 透過多個隱藏層和非線性激活函數，MLP 能夠逼近任意複雜的非線性函數，處理複雜的模式。
  - 廣泛適用性：
    - 可以用於處理各種表格數據的分類和迴歸任務。
- 缺點與限制
  - 對輸入數據的順序或空間關係不敏感：

- MLP 假設輸入特徵是獨立的，會將輸入數據（如圖片的像素點、文本的單詞）「攤平」成一維向量處理，因此會丟失重要的空間（像素間的相對位置）或序列（單詞的順序）資訊。

- 參數數量多，計算成本高：
  - 由於是全連接網路，當輸入特徵或隱藏層神經元數量龐大時，模型的權重數量會急劇增加，導致訓練時間長且易過擬合。
- 易受過擬合影響：
  - 由於參數眾多，如果訓練數據不足或模型過於複雜，很容易過擬合。
- 缺乏可解釋性：
  - 模型的決策過程如同一個「黑箱」，很難直接理解每個權重或神經元具體學到了什麼。

- 適用情境
  - 結構化數據的分類與迴歸：
    - 例如預測客戶流失、信用評分、房價預測等。
  - 作為其他複雜模型的前置或後置層：
    - 在許多更複雜的深度學習架構中，MLP 常被用作特徵提取後的分類器或迴歸器。
  - 簡單圖像辨識（小數據集）：
    - 儘管不擅長，但在圖像規模較小且特徵不複雜時也可應用，但效果通常不如 CNN。

**（2）卷積神經網路（Convolution Neural Network, CNN）**

- 定義
  - 卷積神經網路是一種特殊的前饋神經網路，透過引入卷積層（Convolutional Layers）、池化層（Pooling Layers）等獨特組件，能夠有效地捕捉數據的局部相關性和空間層次結構。

- 卷積神經網路在電腦視覺領域表現卓越，但也應用於語音處理和自然語言處理等其他領域。

- 模型結構

  典型的 CNN 模型通常由以下幾種類型的層次交錯組合而成：

  - 卷積層（Convolutional Layer）：
    - CNN 的核心，負責自動學習並提取輸入數據的局部特徵。
    - 卷積核/濾波器（Kernel/Filter）：
      - 每個卷積層包含多個小的、可學習的卷積核。這些核在輸入數據上進行滑動（卷積運算），每次只關注輸入的一個小區域。
    - 特徵映射（Feature Map）：
      - 每個卷積核在輸入資料上進行卷積運算後，會生成一個特徵映射，代表輸入中某種特定模式（如邊緣、紋理、顏色區塊）的激活程度。一層中通常有多個卷積核，產生多個特徵映射。
    - 權重共享（Weight Sharing）：
      - 同一個卷積核在輸入數據的不同位置上是共享權重的，減少了模型的參數數量，同時也使得 CNN 能夠辨識圖像中位置不變的特徵。
      - 例如，無論貓的眼睛在圖像的左邊還是右邊，都能被同一個濾波器辨識。
  - 激活函數層（Activation Layer）：
    - 通常接在卷積層之後。
    - 對卷積層的輸出（特徵映射）應用非線性激活函數（最常用的是 ReLU），以增加模型的非線性表達能力。
  - 池化層（Pooling Layer）：
    - 目的是縮減特徵映射的尺寸（降採樣），減少計算量，同時保留最重要的特徵資訊。
    - 有助於增加模型的平移不變性（對輸入小幅度的平移不敏感）。

- 常見類型：
  - 最大池化（Max Pooling）：從核掃過區域中提取最大值。這被認為可以捕捉區域內最顯著的特徵。
  - 平均池化（Average Pooling）：計算區域內所有值的平均值。
- 運作方式：
  - 池化操作通常在一個小型、非重疊的窗口內進行。

◆ 全連接層（Fully Connected Layer / Dense Layer）：

- 在經過多層卷積和池化操作提取出高層次抽象特徵後，這些特徵會被「扁平化」(展平為一維向量)，然後輸入到一個或多個全連接層。
- 全連接層的作用類似於傳統的多層感知器，負責將從前面層次學習到的特徵組合起來，進行最終的分類或迴歸預測。
- 通常在全連接層之後會接一個輸出層(例如，用於分類的 Softmax 激活函數)。

● 運作原理

CNN 的訓練過程與 MLP 類似，也依賴於前向傳播和反向傳播：

◆ 前向傳播：

- 輸入數據(如圖像)通過一系列的卷積層、激活層和池化層，逐步提取出越來越抽象的特徵。這些特徵最終被傳遞到全連接層，產生最終的預測結果。

◆ 反向傳播與優化：

- 模型預測與真實標籤之間的損失被計算。隨後，損失的梯度通過反向傳播算法，層層向後傳播，更新卷積核的權重、全連接層的權重和所有偏置，以最小化損失。

● 優點

◆ 自動特徵提取：

- CNN 能夠自動從原始數據中學習和提取層次化的特徵，無需人工設計特徵。這對於圖像等複雜數據尤其重要。

- 權重共享與局部連接：
  - 權重共享減少了模型的參數數量，降低了模型複雜度，減少了過擬合的風險。局部連接則利用了數據（如圖像）的局部相關性。
- 對平移、縮放、旋轉等變形具有一定不變性：
  - 池化層賦予了 CNN 對於輸入數據輕微變形的魯棒性。
- 高效處理高維數據：
  - 卷積和池化的設計，使 CNN 能夠高效地處理圖像等高維網格狀數據。

- 缺點與限制
  - 計算資源需求大：
    - 尤其在訓練深度和大型 CNN 時，需要大量的計算資源（GPU）。
  - 對數據量要求高：
    - 為了學習大量參數並防止過擬合，CNN 通常需要大量的標註數據進行訓練。
  - 模型可解釋性較低：
    - 雖然可以通過視覺化特徵映射來理解學習到的模式，但具體到每個神經元的作用，仍有一定「黑箱」性質。
  - 對旋轉和尺度變化的完全不變性有限：
    - 雖然有一定穩健性，但對於較大的角度旋轉或尺度變化，可能仍需數據增強或更複雜的網路設計。
- 適用情境
  - 圖像辨識與分類：
    - 物體辨識、人臉辨識、醫學圖像診斷、手寫數字辨識等（CNN 最主要和成功的應用領域）。
  - 圖像分割：
    - 將圖像中的每個像素點分類到特定的對象類別。

- 自然語言處理：
  - 在某些文本分類、情感分析任務中，CNN 可用於提取詞語或短語的局部特徵。
- 語音辨識：
  - 處理語音訊號的頻譜圖等。

- 衍伸模型

  卷積神經網路（CNN）在圖像辨識領域經歷了快速的演進。其中，AlexNet 是現代 CNN 發展的一個重要里程碑、於 2012 年奪得 ImageNet 大規模視覺辨識挑戰賽（ILSVRC）冠軍，首次大規模展示了深層卷積網路在圖像分類任務上的卓越效能。AlexNet 的成功不僅證明了 CNN 在處理複雜圖像數據方面的巨大潛力，也開創了後續一系列更深、更高效 CNN 架構的研究熱潮。各種創新架構相繼出現，不斷提升性能並解決不同挑戰：

  - AlexNet：
    - 作為現代深度 CNN 的奠基者，其深層架構、ReLU 激活函數、Dropout 正則化以及對 GPU 加速訓練的應用，共同開啟了圖像辨識的新時代。
  - VGG（Visual Geometry Group）：
    - 以其極深且結構簡單的特點著稱，使用多個 3×3 小型卷積核堆疊來代替大型卷積核，加深了網路深度並提升了非線性能力。
  - GoogLeNet（Inception）：
    - 引入了「Inception 模塊」，允許網路在同一層次並行執行不同大小的卷積核和池化操作，然後將結果拼接，有效利用了計算資源並捕捉多尺度特徵。
  - ResNet（Residual Network）：
    - 透過引入「殘差連接」（或跳躍連接），解決了訓練極深層網路時的梯度消失和模型退化問題，使得構建數百層的神經網路成為可能。
  - DenseNet（Dense Convolutional Network）：

- 將每個層與其所有前一層的特徵映射連接起來，實現了特徵的極大重用，進一步緩解了梯度消失，並減少了參數數量。

- MobileNet / EfficientNet：
  - 這兩類網路著重於模型效率，設計輕量級架構（如 MobileNet 的深度可分離卷積）或自動搜索最佳網路縮放比例（如 EfficientNet），旨在實現模型在移動設備和資源受限環境下的高效部署。

### （3）遞迴神經網路（Recurrent Neural Network, RNN）

- 定義
  - 遞迴神經網路是一種專門設計用於處理可變長度序列輸入數據（Sequential Data）的深度學習模型。
  - 與傳統的前饋神經網路（如 MLP 或 CNN）不同，RNN 具有內部「記憶」機制，使其能夠捕捉數據中的時間依賴性和上下文資訊。
  - RNN 的核心特點是神經元之間存在循環連接（Recurrent Connections），允許資訊在網路內部持續流動，使得當前時間步的輸出不僅依賴於當前輸入，還依賴於過去時間步的計算結果（即「記憶」）。
- 模型結構
  - RNN 的基本單元可以被想像成一個帶有迴圈的神經元。

    在每個時間步（t）：

    a. 輸入（$X_t$）：
       - 接收當前時間步的輸入數據。
       - 例如句子中的一個詞語。

    b. 隱藏狀態（Ht）：
       - RNN 的「記憶」所在，由兩個部分共同計算得出：
         - 當前輸入時間步的隱藏狀態（$X_t$）
         - 上一個時間步的隱藏狀態（$H_{t-1}$）

- 這個結合過程通過權重進行加權求和，然後透過一個非線性激活函數（f）進行轉換（如 Tanh 或 ReLU），生成新的隱藏狀態。

c. 輸出（Ot）：

- 根據當前時間步的隱藏狀態（$H_t$）產生輸出。
- 這個輸出可以是當前時間步的預測，也可以僅僅是為了傳遞給下一層或下一個時間步。

- 關鍵特性
  - 權重共享：
    - RNN 模型在處理序列的不同時間步時，會重複共享使用同一套權重參數。
    - 此權重共享（Weight Sharing）使得 RNN 模型能夠學習到在整個序列中通用的模式，無論這些模式出現在序列的哪個位置，同時也讓模型能夠處理任意長度的序列，因為參數的數量不會隨著序列長度增加而增加。
  - 「展開」視角（Unrolled View）：
    - 雖然 RNN 在結構圖上呈現為帶有迴圈的單元，但為了便於理解其在時間上的計算過程和訓練機制，我們可以將其想像成一個在時間維度上「展開」的深度前饋網路。
    - 在這種視角下，序列的每個時間步都對應著網路中的一個「層」，每個「層」共享相同的權重。
  - 反向傳播
    - RNN 的訓練也使用反向傳播，但由於其時間依賴性，需要使用一種特殊的形式，稱為時間上的反向傳播（Backpropagation Through Time, BPTT）。
    - BPTT 的過程是將整個序列在時間維度上展開，形成一個沒有循環的深層網路。

- 接著像訓練普通的前饋深度網路一樣，從最後一個時間步的損失開始，利用鏈式法則，將梯度沿著時間軸反向傳播，逐一計算並更新所有共享的權重參數。這使得 RNN 能夠學習如何利用過去的資訊來影響當前的預測。

- 優點
  - 處理序列數據：
    - 能夠有效捕捉序列數據中的時間依賴性和上下文資訊，這是 MLP 或 CNN 難以做到的。
  - 權重共享：
    - 跨時間步共享權重，減少了參數數量。
  - 處理可變長度序列：
    - 能夠處理不同長度的輸入序列。
- 缺點與限制
  - 長期依賴問題：
    - 梯度消失（Vanishing Gradient Problem）：
      - 在處理很長的序列時，梯度在反向傳播過程中會呈指數級衰減，導致網路難以學習到遠距離的依賴關係（即「記憶」太短）。
    - 梯度爆炸（Exploding Gradient Problem）：
      - 相反地，梯度也可能呈指數級增長，導致訓練不穩定。
  - 訓練速度慢：
    - 由於其序列性，RNN 的計算本質上是串行的，難以進行高效的平行計算，導致訓練速度相對較慢。
  - 難以捕捉超長距離依賴：
    - 儘管有記憶機制，但對於非常長的序列，學習和保持長期依賴仍然是一個挑戰。
- 適用情境
  - 自然語言處理（NLP）：

- 機器翻譯、語音辨識、文本生成、情感分析、命名實體辨識等（傳統上是 RNN 及其變種的主要應用領域）。

◆ 語音辨識：

- 處理音頻序列和聲學模型。
  - 時間序列預測：
    - 股票價格預測、天氣預報、傳感器數據分析、醫療數據趨勢分析等。
  - 影片處理：
    - 動作辨識、影片內容理解。

● 衍伸模型：為了克服標準 RNN 的長期依賴問題以及提升其性能，一系列更複雜且高效的循環單元被提出，成為序列建模的主流：

◆ 長短期記憶網路（Long Short-Term Memory, LSTM）：

- 引入了「門控機制」（包含輸入門 Input Gate、遺忘門 Forget Gate、輸出門 Output Gate），以及一個獨立的細胞狀態（Cell State）。
- 這些門控單元能夠選擇性地允許資訊流入、保留在細胞狀態中、或從細胞狀態中移除，從而有效地解決了梯度消失問題，使其能更好地捕捉和記憶長期依賴關係。

◆ 門控循環單元（Gated Recurrent Unit, GRU）：

- GRU 是 LSTM 的簡化版本，旨在提供類似的長期記憶能力，同時減少計算複雜度和參數數量。
- 只包含兩個門（更新門 Update Gate 和重置門 Reset Gate），結構更為緊湊，但在許多任務上仍能達到與 LSTM 相當的性能。

◆ 雙向遞迴神經網路（Bidirectional RNN, Bi-RNN）：

- Bi-RNN 透過同時訓練兩個方向（一個正向從序列開始到結束，另一個反向從序列結束到開始）的 RNN，將隱藏狀態結合起來。
- 使得模型在做預測時，能夠同時考慮到序列中過去和未來的上下文資訊，對於許多上下文敏感的任務（如命名實體辨識）非常有用。

- 深度遞迴神經網路（Deep RNN）：
  - 與前饋網路一樣，RNN 也可以堆疊多層，形成深度 RNN。每層 RNN 的隱藏狀態可以作為下一層 RNN 的輸入。這種多層結構有助於學習更複雜的時序特徵。

**（4）Transformer 架構**

- 定義
  - Transformer 架構是近年來在深度學習領域，特別是自然語言處理（NLP）領域引起革命性變革的一種模型。由 Google 在 2017 年的論文《Attention Is All You Need》中提出，徹底顛覆了以往 RNN/LSTM 在序列建模中的主導地位。
  - Transformer 是一種完全基於注意力機制（Attention Mechanism）的深度學習模型架構。捨棄了傳統 RNN 的循環結構和 CNN 的卷積結構來處理序列數據。
  - 其核心在於能夠並行化處理序列，並透過自注意力機制捕捉輸入序列中任意位置之間的長距離依賴關係。
- 設計動機
  - Transformer 的設計初衷，是為了解決傳統序列模型（如 RNN 和 LSTM）在處理長序列時所面臨的兩大核心挑戰：
    - 長期依賴問題：
      - RNN 難以有效捕捉序列中相距較遠的元素之間的依賴關係，容易出現梯度消失問題。
    - 並行化困難：
      - RNN 的循環結構導致其計算本質上是串行的，無法有效利用現代硬體的平行計算能力，訓練效率低下。
      - Transformer 透過完全基於注意力機制，提供了一種更高效且能處理長距離依賴的序列建模方法。

- 模型架構

  Transformer 架構主要由編碼器（Encoder）和解碼器（Decoder）堆疊組成，每個編碼器和解碼器內部都包含幾個關鍵組件：

  - 編碼器堆疊（Encoder Stack）：
    - 輸入：
      - 接收原始輸入序列（例如，源語言句子）。
    - 結構：
      - 由多個相同的編碼器層堆疊而成。
      - 每個編碼器層包含一個多頭自注意力機制和一個前饋網路。
    - 編碼器功能：
      - 編碼器像是一個「理解者」或「特徵提取器」，任務是深入分析原始輸入數據（例如，一個句子），並將其中的每一個元素（如詞語）轉換成一種更豐富、更能捕捉其語意和上下文關係的數學表示。
      - 編碼器的主要作用是將輸入序列中的每個元素（例如，一個句子），轉換成一個富含上下文資訊的高維度「上下文表示」，這些表示捕捉了輸入序列內部的所有依賴關係。
  - 解碼器堆疊（Decoder Stack）：
    - 輸入：
      - 接收來自編碼器的最終上下文表示，以及已生成的部分目標序列。
    - 結構：
      - 由多個（通常與編碼器層數相同）相同的解碼器層堆疊而成。
      - 每個解碼器層包含三個核心組件：
        - 帶遮罩的多頭自注意力機制（Masked Multi-head Self-Attention），確保在生成當前詞時，解碼器只能「關注」已生成的前序詞語，而不能「偷看」未來的詞語。
        - 編碼器-解碼器（Encoder-Decoder）注意力機制，允許解碼器在

生成每個詞時，根據自身當前的狀態，動態地「關注」編碼器輸出中的相關資訊。

- 前饋網路（Feed-forward Network, FFN）。

- 功能：
  - 解碼器接收「編碼器對輸入的理解（上下文表示）」，並轉化為一個連貫、有意義的輸出序列。
  - 序列生成：
    - 負責從頭開始或基於一個起始標記，一個接一個地生成目標序列中的每個元素（例如，在機器翻譯中生成目標語言的詞語，或在文本生成中生成下一個詞）。
  - 利用輸入上下文：
    - 透過編碼器-解碼器注意力機制，解碼器能夠有效地「關注」編碼器對輸入序列（原始訊息）所提取的精華表示。使得解碼器在生成每個輸出元素時，能夠充分理解輸入的語意上下文。
  - 自迴歸生成：
    - 在生成當前時間步的元素時，解碼器會利用已生成的所有前面時間步的元素作為上下文。其中透過帶遮罩的自注意力機制，確保模型在生成過程中不會「偷看」未來的資訊。
- 輸出：
  - 解碼器逐步生成目標序列中的詞語，直到生成結束標誌。

- 模型組成機制

  Transformer 的創新來自於其獨特的模塊化設計和核心組件。這些組件就像是構成 Transformer 神經網路的「基本積木」，各自負責特定的功能，彼此協作而具有良好的序列處理能力：

  - 注意力機制：

    注意力機制（Attention Mechanism）是 Transformer 最核心的創新，賦

予模型在處理序列中任何一個元素時，能夠動態地「關注」序列中所有其他相關元素，並根據相關性賦予不同權重。

- 核心思想：
  - 當模型需要處理序列中的某個元素時，不再像傳統遞迴神經網路那樣依賴固定大小的隱藏狀態獲取上下文，而是能夠計算當前元素與序列中所有其他元素之間的相似度，並據此分配注意力權重。
  - 相關性越高的元素，獲得的權重越大，最終通過加權求和得到一個綜合了相關資訊的「上下文向量」。
- 目的：
  - 解決傳統遞迴神經網路難以處理長序列的「長期依賴」問題，因為模型可以直接建立序列中任意兩個元素之間的聯繫，無需依賴逐層遞進的傳遞。

◆ 自注意力（Self-Attention）

自注意力是注意力機制的一個特例，讓模型在處理序列中的一個詞時，能夠同時「關注」並權衡「該詞語與其所屬序列中所有其他詞語」之間的關係。

- 核心原理：
  - 自注意力機制將輸入序列中的每個元素（例如詞向量）通過三個不同的線性變換，映射到三個不同的向量空間，分別產生：
    - 查詢向量（Query, Q）：代表「我在找什麼？」或「我的興趣是什麼？」
    - 鍵向量（Key, K）：代表「我能提供什麼？」或「我的內容是什麼？」
    - 值向量（Value, V）：代表「如果我被關注了，我會提供什麼資訊？」

- 特點：允許模型直接建立序列中任意兩個元素之間的關係，無論它們在序列中相距多遠，從根本上解決了遞迴神經網路的長期依賴問題。

◆ 多頭注意力機制

多頭注意力機制（Multi-Head Attention）是自注意力機制的一個擴展，並行地執行多次（多個「頭」）獨立的自注意力運算。每個「頭」都有自己獨立的 Query、Key、Value 權重矩陣，因此它們在不同的「表示子空間」中學習不同的注意力模式。

■ 核心原理：

- 拆分與並行：輸入的 Q、K、V 向量會被線性映射並拆分成 h 個較低維度的「頭」。
- 獨立注意力：每個頭獨立地執行自己的自注意力計算，學習不同類型的關係。
- 拼接與線性轉換：將所有 h 個頭的自注意力輸出拼接起來，然後再經過一個最終的線性轉換，將結果投影回原始的維度，作為多頭注意力的最終輸出。

■ 特點：

- 捕捉多樣關係：允許模型在同一時間捕捉多種不同的關係類型和上下文訊息，因為不同的頭可以學習不同的注意力模式。
- 增強表示能力：通過結合多個「視角」的注意力，模型能夠產生更豐富、更全面的上下文表示。

◆ 位置編碼：

■ 由於 Transformer 完全摒棄了循環結構，模型本身無法直接感知序列中詞語的順序或絕對位置資訊。

■ 因此設計出位置編碼（Positional Encoding），是一種透過將「位置資訊注入詞嵌入（Word Embeddings）」的方法。

- 位置編碼透過將具有特定模式（通常是正弦和餘弦函數）的位置向量加到原始的詞嵌入上，確保模型能理解序列的順序性。

◆ 前饋網路：

- 前饋網路（Feed-Forward Network）是相對簡單的、由兩個線性層構成的全連接網路（中間包含一個激活函數，如 ReLU）。
- 對序列中每個位置的輸出獨立且相同地應用這個網路，用於進一步轉換和處理注意力層所提取的資訊。

◆ 殘差連接與層歸一化：

- 殘差連接（Residual Connections）：
  - 每個子層（例如注意力層或前饋網路）的輸入都會直接加到其輸出上（即 Output = Input + Sublayer（Input））。
  - 這種「跳躍連接」能有效緩解訓練深層網路時的梯度消失問題，加速收斂，並允許模型構建極深的層次。
- 層歸一化（Layer Normalization, LN）：
  - 在每個子層的計算中，會對其輸出進行歸一化（Normalization）處理，使該層的特徵在不同維度上具有相對穩定的均值與變異數。
  - 相較於批次歸一化（Batch Normalization），層歸一化不依賴批次大小，因此特別適合自然語言處理這類序列模型。
  - 層歸一化能有效穩定訓練過程，避免數值發散，並減少內部協變偏移（Internal Covariate Shift, ICS）。
  - 內部協變偏移（Internal Covariate Shift, ICS）：隨著訓練進行，前面層的參數不斷更新，導致後續層的輸入分布不斷改變，這會迫使模型的每一層必須不停重新適應新的輸入分布，降低訓練效率並延緩收斂。

● 優點

◆ 捕捉長距離依賴：

- 自注意力機制可以直接建立序列中任意兩個位置之間的關係，有效解決了 RNN 的長期依賴問題。

◆ 高度並行化：

- 由於沒有循環結構，序列中的每個位置可以同時計算，提高了訓練效率。

◆ 良好的表示學習能力：

- 能夠學習到非常豐富和語意化的特徵表示。

◆ 遷移學習和預訓練模型：

- Transformer 成為大型預訓練語言模型（如 BERT、GPT 系列）的基石，這些模型透過在大量文本數據上預訓練，然後針對特定任務進行微調，極大推動了 NLP 的發展。

• 缺點與限制

◆ 計算複雜度高：

- 自注意力機制的計算複雜度與序列長度的平方 O（$L^2$）成正比，處理極長序列時計算量和記憶體消耗巨大。

◆ 記憶體消耗大：

- 需要儲存注意力權重矩陣，對於長序列會佔用大量記憶體。

◆ 數據飢渴：

- Transformer 及其大型變體通常需要大量的標註或未標註數據才能充分發揮效能，否則容易過擬合。

◆ 缺乏內建序列歸納偏置：

- 與 RNN 自然處理序列不同，Transformer 需要額外引入位置編碼來提供順序資訊。

• 適用情境

◆ 自然語言處理（NLP）：

- 機器翻譯、文本生成、文本摘要、問答系統、情感分析、文本分類等。

- 所有基於大型預訓練語言模型的應用。

◆ 電腦視覺（Computer Vision）：

- 圖像分類、物體檢測、圖像分割等。

◆ 語音處理：

- 語音辨識、語音合成。

- 衍伸模型

Transformer 架構因其強大的模型表達能力和並行化特性，迅速成為深度學習領域的基石，尤其在自然語言處理（NLP）領域，衍生出了一系列革命性的預訓練語言模型（Pre-trained Language Models, PLMs），這些模型通常在大量數據上進行預訓練後，針對特定任務進行微調（Fine-tuning）。以下介紹幾種衍伸模型：

◆ BERT：

- BERT（Bidirectional Encoder Representations from Transformers），由 Google 提出，是 Transformer 編碼器部分的代表性應用。
- BERT 模型採用了雙向上下文預訓練（透過 Masked Language Model 和 Next Sentence Prediction），讓模型能同時理解一個詞語在句子中左右兩邊的上下文語意。
- 擅長於語言理解任務。

◆ GPT 系列（Generative Pre-trained Transformer）：

- GPT（Generative Pre-trained Transformer）系列模型由 OpenAI 開發，如 GPT-2、GPT-3、GPT-4 、GPT-5 等。
- 主要基於 Transformer 的解碼器部分，採用單向（自迴歸）預訓練。
- 擅長於文本生成任務，能夠生成連貫、高品質的文章、故事、對話等。模型規模龐大，展現出驚人的零樣本（Zero-shot）和少樣本（Few-shot）學習能力。

- T5：
  - T5（Text-to-Text Transfer Transformer）由 Google 提出，將所有 NLP 任務統一視為「文字到文字」（Text-to-Text）問題。
  - 使用完整的 Transformer 編碼器-解碼器架構。
  - 無論是分類、摘要、問答還是翻譯，都被重新構建成生成另一個文本序列的任務。這種統一的框架使其具有很強的通用性。
- Vision Transformer：
  - Vision Transformer（ViT）將 Transformer 架構從 NLP 成功引入電腦視覺領域。
  - ViT 將圖像切分成一系列固定大小的圖像塊（Patches），然後將這些圖像塊視為序列中的「詞語」，直接輸入到 Transformer 中進行處理。
  - 證明了 Transformer 處理網格狀數據的潛力，挑戰了 CNN 在圖像任務上的主導地位。
- 長序列 Transformer：
  - 如 Longformer, Reformer, Performer 等，這類型模型旨在解決原始 Transformer 中自注意力機制計算複雜度與序列長度平方成正比 O（$L^2$）的問題。
  - 透過引入稀疏注意力（Sparse Attention）、局部注意力（Local Attention）、注意力近似計算等技術，使得 Transformer 能夠更高效地處理非常長的序列（例如數千甚至數萬個 tokens），同時大幅降低計算和記憶體消耗。

### （5）生成式模型

生成式模型的核心目標是學習訓練數據的底層分佈（Underlying Distribution），一旦模型學會了這個分佈，就能夠生成新的、與訓練數據相似但卻是前所未見的數據。以下介紹兩種重要的生成式模型：自編碼器（Autoencoder）和生成對抗網路（Generative Adversarial Network, GAN）。

- 自編碼器
  - 定義
    - 自編碼器（Autoencoder）是一種旨在學習數入數據的高效表示（Efficient Representation）或編碼（Encoding）的非監督式學習模型，透過嘗試重構自身的輸入來達到學習目的。
  - 模型結構

    自編碼器通常由兩個主要部分組成：
    - 編碼器（Encoder）：
      - 負責將高維度的輸入數據轉換（或「編碼」）為一個低維度的潛在空間向量，這個潛在空間通常被稱為瓶頸層（Bottleneck Layer）。
    - 解碼器（Decoder）：
      - 負責將編碼器生成的潛在空間向量轉換（或「解碼」）回原始輸入數據的維度。
  - 訓練目標：
    - 自編碼器透過最小化重構誤差（Reconstruction Error）來進行訓練。重構誤差衡量的是原始輸入數據與解碼器輸出（即重構數據）之間的相似度。
    - 常用的損失函數包括均方誤差（MSE）用於連續數據，或二元交叉熵用於二元數據。
    - 在訓練過程中，模型被迫學習如何將輸入數據的冗餘資訊去除，只保留最重要的特徵來完成重構任務，從而學習到一種壓縮且有意義的數據表示。
  - 優點
    - 降維：
      - 有效學習數據的低維潛在表示，達到降維的效果。

- 特徵學習：
  - 學習到的潛在表示可以被視為原始數據的抽象特徵，可用於其他機器學習任務（如分類或聚類）的前置特徵工程。
- 數據去噪：
  - 可以訓練自編碼器來將帶有噪聲的輸入中重構出乾淨的數據（去噪自編碼器）。
- 異常偵測：
  - 訓練良好的自編碼器在重構正常數據時誤差較小，而在重構異常數據時誤差較大，因此可以通過重構誤差來辨識異常。

◆ 缺點與限制

- 重構而非生成「新」數據：
  - 傳統的自編碼器主要用於重構輸入，雖然能學到分佈，但不能像 GAN 那樣憑空生成多樣化的全新樣本。
- 潛在空間缺乏結構：
  - 傳統自編碼器學到的潛在空間通常缺乏連續性或結構，這意味著在潛在空間中採樣並不能保證生成有意義的數據。
- 生成數據品質可能有限：
  - 純粹的重構任務可能無法學習到生成高品質新數據所需的全部數據分佈特性。

◆ 適用情境

- 降維與數據壓縮：
  - 為高維數據找到更緊湊的表示。
- 特徵學習：
  - 作為無監督預訓練步驟，為後續監督任務提供更好的特徵。
- 數據去噪：
  - 濾除數據中的噪聲。

- 異常偵測：
  - 辨識與訓練數據分佈不符的離群值。

- 生成對抗網路（GAN）
  - 定義
    - 生成對抗網路（Generative Adversarial Network, GAN）透過兩個神經網路的「對抗」過程，以此學習數據的分佈並生成新數據。
    - GAN 由兩個相互競爭的神經網路組成：一個生成器（Generator）和一個判別器（Discriminator）。這兩個網路在一個「零和博弈」（Zero-Sum Game）中進行訓練，直到達到平衡。
  - 模型結構
    - 生成器（Generator）：
      - 輸入：
        - 通常接收一個隨機噪聲向量（Latent Vector），這些噪聲通常從簡單的分佈中採樣（如高斯分佈）。
      - 功能：
        - 目標是將這個隨機噪音轉換成看起來像真實數據的樣本（例如，逼真的圖像、文本、音訊等）。
        - 生成器試圖「欺騙」判別器，讓判別器相信它生成的數據是真實的。
    - 判別器（Discriminator）：
      - 輸入：
        - 接收兩種類型的數據：真實的訓練數據樣本，以及生成器生成的假數據樣本。
      - 功能：
        - 一個二元分類器，目標是區分輸入數據是「真實」的還是「生成器生成（虛假）」的。
        - 判別器試圖準確地辨識出生成器生成的假數據。

- 運作原理

  GAN 的訓練是一個動態的、迭代的過程，生成器和判別器輪流進行優化：

  - 訓練判別器：
    - 在這個階段，判別器被訓練為一個分類器，目標是最大化其區分真實數據和生成數據的能力。
    - 判別器會對真實數據給出高分（接近 1），對生成數據給出低分（接近 0）。
  - 訓練生成器：
    - 在這個階段，判別器的參數被固定，生成器被訓練來最小化判別器將其生成數據辨識為假數據的能力。
    - 換句話說，生成器試圖「欺騙」判別器，讓判別器對其生成的假數據給出高分（接近 1）。
  - 重複過程：
    - 不斷重複，直到生成器能夠生成高度逼真、判別器幾乎無法區分的數據，達到一個「納什均衡」（Nash Equilibrium）狀態。

- 優點
  - 生成高品質數據：
    - 能夠生成極其逼真、具有高度視覺（或其他形式）真實感的樣本，這是許多其他生成模型難以達到的。
  - 學習複雜分佈：
    - 能夠學習非常複雜和高維的數據分佈。
  - 無需顯示概率密度函數：
    - 不需要直接計算數據的概率密度函數，這對於許多高維數據是巨大的優勢。

- 缺點與限制
  - 訓練不穩定性（Training Instability）：

- GAN 以其難以訓練而聞名。訓練過程非常敏感，容易出現模式崩潰（Mode Collapse，生成器只生成少數幾種類型的樣本）或梯度消失等問題，導致訓練失敗或生成多樣性不足。

- 超參數敏感：
  - 對於學習率、網路架構等超參數非常敏感，調參困難。
- 量化評估困難：
  - 缺乏客觀、公認的量化指標來評估生成樣本的品質和多樣性，通常依賴於人工判斷或間接指標（如 Inception Score, FID）。

- 適用情境
  - 真實圖像生成：
    - 生成人臉、動物、風景、藝術畫等。
  - 數據增強：
    - 為訓練集生成更多樣的數據以提升判別模型的魯棒性。
  - 圖像到圖像轉換：
    - 風格遷移、圖像修復、超解析度重建（Super-Resolution）、黑白圖像上色。
  - 跨模態生成：
    - 從文本生成圖像，或從圖像生成文本描述。

## 4. 深度學習主流框架

深度學習領域發展迅速，各家科技公司與研究社群推出了多種開源框架，協助研究者與工程師更快速開發、訓練與部署模型。這些框架在語法易用性、效能、靈活度及生態系統支援上各有強項。以下介紹幾個在實務與學術上最常見的主流框架：

### （1）TensorFlow 與 Keras

- TensorFlow
  - TensorFlow 是由 Google 開發並維護的開源機器學習框架，自 2015 年發布以來，迅速成為業界和學術界應用最廣泛的深度學習框架之一。TensorFlow 提供了一套全面、靈活的工具、函式庫和社群資源，用於建構和部署各種機器學習模型。
  - 核心特性
    - 計算圖（Computation Graph）：
      - TensorFlow 早期版本主要基於靜態計算圖，先定義好運算結構，再執行數據流。這提供了優化的潛力，但有時會降低開發彈性。近年來，TensorFlow 2.x 已轉向即時執行（Eager Execution），允許像 Python 那樣直觀地即時運行操作，極大地提升了開發和調試的便利性。
    - 跨平台部署：
      - 支援多種平台部署，包括 CPU、GPU、TPU（Google 自研的張量處理單元）、行動裝置（TensorFlow Lite）、物聯網設備（TensorFlow Lite Micro）和網頁（TensorFlow.js）。
    - 生產級部署：
      - 擁有豐富生態系統支援生產環境的部署，如 TensorFlow Extended（TFX）用於機器學習管道管理，以及 TensorFlow Serving 用於模型服務。
- Keras
  - Keras 是一個高階神經網路 API，以使用者友善、模組化且易於擴展為設計宗旨。自 TensorFlow 2.0 起，Keras 已被完全整合為 TensorFlow 的官方高階 API，成為其核心組成部分（tf.keras）。
  - 核心特性：
    - 極簡主義：

- Keras 的設計理念是讓用戶能夠以最少的代碼，快速而直觀地建構深度學習模型。其 API 簡潔，易於學習和使用。

- 模組化：
  - 模型由獨立的模塊（如層、激活函數、優化器、損失函數）堆疊而成，這些模塊可以自由組合。
- 易於原型開發：
  - 適合快速試驗和原型開發，讓研究者和開發者能夠迅速驗證想法。
- 靈活後端：
  - 在被 TensorFlow 整合之前，Keras 曾支援多個後端（如 TensorFlow, Theano, CNTK），展現了其靈活性。

**（2）PyTorch**

- PyTorch 由 Facebook AI Research（FAIR）開發的開源機器學習框架。以其動態計算圖（Dynamic Computation Graph）的特性而聞名，在學術研究和快速原型開發領域迅速崛起，成為許多研究者和工程師的首選。
- 核心特性：
  - 即時執行：
    - PyTorch 最大的特點是採用動態計算圖，這意味著運算在定義時立即執行，而非像早期 TensorFlow 那樣先構建完整圖形再運行。
    - 使得 PyTorch 的行為更像標準的 Python 代碼，極大地簡化了調試、開發和實驗流程。
  - Pythonic 介面：
    - PyTorch 的 API 設計非常「Pythonic」，與 Python 語言的習慣用法高度契合，讓 Python 開發者能夠更直觀、更快速地學習和使用。

- 自動微分：
  - PyTorch 內建自動微分（Autograd）引擎，能夠自動計算任何運算的梯度，這對於反向傳播訓練神經網路至關重要，簡化了梯度計算的複雜性。
- 豐富的生態系統：
  - 圍繞 PyTorch 建立了一個活躍的生態系統，包括用於視覺任務的 torchvision、用於自然語言處理的 torchtext、用於語音處理的 torchaudio，以及各種模型庫和工具（如 PyTorch Lightning, Hugging Face Transformers）。

**（3）JAX、MXNet、PaddlePaddle 等簡介**

除了 TensorFlow 和 PyTorch 這兩大主流框架之外，還有一些在特定領域或由特定組織支持的深度學習框架，各有其獨特的優勢和定位。

- JAX
  - JAX 是由 Google 開發的機器學習轉換（ML Transformations）框架，目的在於結合 NumPy 庫的易用性、自動微分（Autograd）功能以及對 GPU/TPU 的高效利用。
  - JAX 並非傳統意義上的端到端深度學習框架，更像一個數值計算庫，提供高效能數值運算和函數式程式設計的工具。
- MXNet
  - Apache MXNet 是一個靈活且高效的深度學習框架，由 Amazon Web Services （AWS）提供主要支持。
  - 支援多種程式語言綁定，並提供了混合式程式設計（兼具符號式和命令式風格）。

## 模擬考題

1. 機器學習（Machine Learning）中，若輸入資料沒有標註，欲探索潛在的群組或模式，應屬於哪種學習類型？

   （A）監督式學習（Supervised Learning）

   （B）半監督式學習（Semi-supervised Learning）

   （C）非監督式學習（Unsupervised Learning）

   （D）強化式學習（Reinforcement Learning）

2. 在監督式學習中，當模型要預測一個連續數值時，這種任務稱為什麼？

   （A）分類（Classification）

   （B）迴歸（Regression）

   （C）聚類（Clustering）

   （D）降維（Dimensionality Reduction）

3. 在深度學習中，用於學習資料中局部特徵的網路架構通常是什麼？

   （A）循環神經網路（Recurrent Neural Network, RNN）

   （B）決策樹（Decision Tree）

   （C）卷積神經網路（Convolutional Neural Network, CNN）

   （D）隨機森林（Random Forest）

4. 支援向量機（Support Vector Machine, SVM）若用於迴歸問題，稱為什麼？

   （A）Logistic Regression

   （B）Decision Tree Regression

   （C）Random Forest

   （D）Support Vector Regression

5. 機器學習模型若過度學習訓練資料中的雜訊，導致泛化能力下降，此現象稱為什麼？

   （A）欠擬合（Underfitting）

   （B）過擬合（Overfitting）

   （C）特徵縮放（Feature Scaling）

（D）梯度爆炸（Gradient Explosion）

6. 在深度學習中，哪一個是為了引入非線性，使神經網路能學習複雜模式的元件？

（A）損失函數（Loss Function）

（B）激活函數（Activation Function）

（C）隱藏層（Hidden Layer）

（D）梯度下降（Gradient Descent）

7. 若想在機器學習中降低模型過度擬合風險，可採用哪種方法？

（A）提高學習率

（B）減少資料量

（C）正則化（Regularization）

（D）隨機刪除特徵

8. 在深度學習模型的訓練過程中，計算模型輸出與實際值差距的函數稱為什麼？

（A）激活函數（Activation Function）

（B）池化函數（Pooling Function）

（C）損失函數（Loss Function）

（D）梯度函數（Gradient Function）

9. Transformer 模型中用來捕捉輸入序列不同位置間依賴關係的核心機制是什麼？

（A）池化機制（Pooling）

（B）注意力機制（Attention）

（C）決策樹分支

（D）激活函數（Activation Function）

10. 在決策樹模型中，用於判斷分裂效果好壞的指標，常見的是下列哪一種？

（A）機率密度函數（Probability Density Function）

（B）均方根誤差（Root Mean Squared Error）

（C）基尼不純度（Gini Impurity）

（D）卷積核大小（Kernel Size）

# 考題解析

1. **Ans（C）** 非監督式學習（Unsupervised Learning）

   **解析：**非監督式學習（Unsupervised Learning）不依賴標註資料，主要目的是從資料中發現潛在結構、模式或分佈，例如分群（Clustering）、降維（Dimensionality Reduction）等技術，是探索性資料分析的重要工具。

2. **Ans（B）** 迴歸（Regression）

   **解析：**迴歸（Regression）是預測連續數值的任務，如房價預測、氣温預測等，與分類不同，分類是用來預測離散類別結果。

3. **Ans（C）** 卷積神經網路（Convolutional Neural Network, CNN）

   **解析：**卷積神經網路（CNN）透過卷積層偵測局部特徵，尤其擅長處理影像、語音等有空間或時間結構的資料，是深度學習中重要的基礎架構之一。

4. **Ans（D）** Support Vector Regression

   **解析：**當支援向量機（SVM）應用於連續數值的預測問題時，稱為支援向量迴歸（Support Vector Regression, SVR）。SVR 與傳統 SVM 不同，目標是尋找一條最佳超平面，使大多數資料點落在預設的誤差範圍內。

5. **Ans（B）** 過擬合（Overfitting）

   **解析：**過擬合（Overfitting）是指模型在訓練資料上表現極好，卻無法在未知新資料上維持預測效能，通常是因為模型過於複雜，學習到不具代表性的噪聲。

6. **Ans（B）** 激活函數（Activation Function）

   **解析：**激活函數（Activation Function）如 ReLU、Sigmoid 等，為神經網路帶來非線性能力，若沒有激活函數，多層網路仍只是線性變換，無法表現複雜的關係。

7. **Ans（C）** 正則化（Regularization）

   **解析：**正則化（Regularization）透過在損失函數中加入 L1 或 L2 懲罰項，限制模型權重的大小，減少模型過度擬合訓練資料，提升泛化能力。

8. **Ans（C）** 損失函數（Loss Function）

   **解析**：損失函數（Loss Function）用來衡量模型預測值與實際值的差距，是指引優化器調整參數方向的重要依據。常見如均方誤差（MSE）、交叉熵損失（Cross-Entropy Loss）等。

9. **Ans（B）** 注意力機制（Attention）

   **解析**：Transformer 模型核心在於注意力機制（Attention），透過計算序列中各元素間的關聯權重，捕捉長距離依賴，徹底取代傳統 RNN 的循環架構，大幅提升計算效率與表現。

10. **Ans（C）** 基尼不純度（Gini Impurity）

    **解析**：基尼不純度（Gini Impurity）衡量節點內樣本的不純度，數值越低表示節點內樣本越集中於單一類別，是決策樹演算法常用的分裂判斷依據，尤其在分類任務中扮演重要角色。

# 第五章　機器學習建模與參數調校

機器學習模型的效能不僅取決於演算法本身，更仰賴整體建模流程中每一個環節的設計與執行品質。從資料整理、特徵建構，到模型選擇、訓練與優化，每一個步驟都直接影響最終預測結果的準確性與穩定性。因此，模型建構並非單一演算法的套用，而是一套完整且需反覆調整的工程流程。

本章「機器學習建模與參數調校」將聚焦於模型導入前的資料準備與特徵工程、導入中的模型設計與訓練策略，以及導入後的效能驗證與參數調整。章節主要內容如下：

- **數據準備與特徵工程：**

  處理資料品質與轉換方式，建構有意義的輸入特徵，提升模型學習效率。

- **模型選擇與架構設計：**

  依據任務性質與資料特性選擇適當模型，並針對模型結構進行初步設計與假設空間限制。

- **模型訓練、評估與驗證：**

  定義學習目標、資料切分方式與評估指標，確保模型具備泛化能力。

- **模型調整與優化：**

  進行參數調校、效能微調與策略融合，提升模型效能與部署可行性。

## 重點掃瞄

# 5.1 數據準備與特徵工程

### 1. 前言與章節導覽

在機器學習流程中「資料品質與特徵表達」往往比演算法本身更決定模型的最終表現。模型所接收的輸入，是一組經過轉換、標準化、編碼甚至衍生處理的數據向量，若這些資料本身具有錯誤、缺陷或資訊不足，即使使用最先進的演算法也難以得出良好的結果。因此，數據準備與特徵工程不僅是建模前的必要步驟，更是模型效能與穩定性的基礎建設。

本小節旨在說明資料處理與特徵建構的整體流程與重要性，並釐清其與模型選擇、參數調校之間的相依關係。

### 2. 數據清理

#### （1）缺失值處理

資料缺失（Missing Value）可能源於感測失效、人工遺漏或系統錯誤，其處理方式需視資料特性、缺失比例、缺失機制與模型特性而定：

- 刪除法（Deletion）：
  - 當缺失比例極低、樣本數充足，且缺失分佈無偏時，可直接刪除含缺值的欄位或列，以避免導入不確定性。
- 填補法（Imputation）：
  - 均值、中位數、眾數填補：
    - 適用於數值或類別型欄位，方法簡單但可能降低變異性或產生偏差。
  - 相似樣本填補：
    - 如 Hot Deck、K 最近鄰（KNN）填補，依據特徵相似度補全遺失值。

- 預測模型填補：
  - 透過迴歸或分類模型，預測缺值欄位，適用於特徵間具有高度相關性時。
- 缺失指標編碼：
  - 新增欄位標示是否缺失，有助模型學習隱含資訊，常見於樹模型中。

**（2）異常值偵測與處理（Outlier Detection & Handling）**

異常值可能來自輸入錯誤、資料錯置或極端觀測，對模型參數與分佈估計有顯著影響：

- 統計方法：
  - 利用 Z 分數（Z-score）或四分位距（IQR）界定明顯偏離的觀測值。
- 視覺化分析：
  - 透過箱型圖、散佈圖、時間序列圖輔助觀察極端點或非典型趨勢。
- 機器學習方法：
  - 使用 Isolation Forest、Local Outlier Factor（LOF）等演算法偵測高維資料中的異常。
- 處理策略：
  - 移除：
    - 在可確認錯誤輸入時直接刪除。
  - 截尾與轉換：
    - 將值限制於上下邊界內，或進行對數、Box-Cox 等轉換。
  - 標記保留：
    - 在異常值本身具有預測價值（如欺詐偵測）時保留，並作為特徵輸入。

**（3）重複樣本與資料一致性檢查**

同一觀察單位在資料集中多次出現，常因系統重複寫入、資料整合錯誤或缺

少唯一辨識碼導致。

- 重複資料偵測：
  - 透過主鍵比對或欄位相似度判斷資料重複，避免訓練集被特定樣本主導。
- 單位與格式標準化：
  - 統一數據單位（如公克與公斤）、時間格式與類別值（如「male」「男」）以確保欄位一致性。

**（4）資料型別轉換與欄位格式調整**

- 類型轉換：
  - 確保數值型與類別型資料正確標示，以利後續特徵工程與模型處理（如類別編碼、標準化等）。
- 時間資料解析：
  - 將日期時間轉換為時間戳、週期性變數（如星期幾、月份）或進行時間差計算，有助時間序列建模。

**（5）清理流程的策略考量**

資料清理策略應根據模型類型與任務需求彈性調整，並考慮實務上的可重現性與治理要求：

- 模型對資料品質的敏感度存在差異：
  - 樹模型（如 XGBoost、Random Forest）對於缺值與異常值具較高容忍度，能自動處理部分遺失資訊。
  - 線性模型與神經網路對輸入資料較為敏感，需特別注意缺值補全與特徵正規化，否則容易造成訓練不穩或結果偏誤。
- 建立資料處理紀錄與流程可追溯性（Data Lineage）：
  - 所有清理動作應具備明確記錄，包括欄位處理邏輯、填補方法、異常值調整依據等，以確保資料處理流程可被還原、驗證與持續維護。

- 此舉不僅有助模型開發過程中的透明度與重現性，也符合資料治理與法規合規的最佳實務。

### 3. 特徵選擇與降維方法

面對大量特徵時，若直接進行建模可能導致模型複雜度提升、運算時間延長，甚至造成過度擬合的情況。特徵選擇（Feature Selection）便是指從所有可用的原始特徵中，篩選出一個最佳的子集的過程，以降低數據的維度，同時保留足以讓模型準確預測的關鍵資訊。

這個過程不創造新的特徵，而是從現有特徵中進行「選擇」。透過系統化的方法保留最具價值的特徵，進而提高模型效能與解釋性。

#### （1）特徵選擇方法

- Filter 方法（過濾法）
  - 透過統計量或相關係數，獨立於模型之外快速篩選重要特徵。
  - 常用統計方法包括：皮爾森相關係數、卡方檢定（Chi-square Test）、ANOVA 檢定。
  - 優點：速度快、不依賴特定模型。
  - 限制：無法考慮特徵間交互作用。
- Wrapper 方法（包裝法）
  - 使用模型表現（如準確率、F1-score）作為標準，透過遞迴特徵消除（Recursive Feature Elimination, RFE）或前向/後向選擇進行特徵篩選。
  - 優點：精確考量特徵互動效果。
  - 限制：計算成本高，可能過度擬合。
- Embedded 方法（嵌入法）
  - 在模型訓練過程中內建特徵選擇機制，例如決策樹模型的重要性分析或 Lasso、Ridge 等正則化方法。

- 優點：在建模過程中同時完成特徵選擇。
- 限制：需特定模型或算法支援，結果可能受模型超參數影響。

### （2）降維方法

降維（Dimensionality Reduction）是將高維特徵空間轉換為低維空間，同時保留資料中最有意義的結構或變異訊號。常見方法包括：

- 主成分分析（Principal Component Analysis, PCA）
  - 透過線性變換，找出能最大化資料變異的方向，並以這些主成分重構資料。廣泛用於視覺化、雜訊過濾與建模加速。
- 線性判別分析（Linear Discriminant Analysis, LDA）
  - 同為線性降維，但以最大化類間差異、最小化類內變異為目標，適用於分類問題。
- t-SNE、UMAP 等非線性降維方法
  - 可保留高維資料在低維空間中的鄰近關係，常用於視覺化探索，但不適合直接用於預測建模。
- 奇異值分解（Singular Value Decomposition, SVD）
  - 適用於矩陣分解，廣泛應用於文字分析（如 LSA）、推薦系統等任務。

## 4. 特徵轉換與資料標準化

原始資料中的特徵，往往具有不同的尺度、分佈型態與資料類型。若不進行適當轉換，將可能影響模型的學習效率、收斂行為與預測表現。因此，特徵轉換與標準化處理是機器學習建模前的重要前置步驟，尤其對於基於梯度或距離計算的模型（如線性模型、SVM、KNN、神經網路）更為關鍵。

### （1）資料尺度調整

不同特徵可能具有不同量級（如收入以萬元計、年齡以歲為單位），會導致模型訓練偏向高數值特徵。為避免此偏差，需進行尺度調整（Scaling）：

- Min-Max Normalization（最小 - 最大正規化）
  - 將數值線性縮放至 0～1 區間。
  - 優點：保留原始變數的分佈比例，易於解釋與視覺化。
  - 限制：對極端值敏感，異常值會壓縮其他數值的縮放範圍。
- Z-score Standardization（Z 分數標準化）
  - 將數值轉換為平均值為 0、標準差為 1 的常態分佈。
  - 優點：適用於符合常態分佈的資料，保留資料的原始形狀與相對位置。
  - 限制：對極端值仍具敏感性，若分佈偏態則標準化結果偏移。
- Robust Scaling（穩健標準化）
  - 使用中位數與四分位距（IQR）進行縮放。
  - 優點：對極端值具高度抵抗性，適用於資料分佈偏態或含離群值的情境。
  - 限制：轉換後資料不保證符合任何標準分佈，可能影響部分建模假設。

**（2）分佈轉換**

某些模型假設輸入資料近似常態分佈（如線性迴歸），因此可針對偏態資料進行分佈轉換（Transformation）：

- 對數轉換（Log Transform）
  - 常用於處理右偏分佈，降低極端值影響（如收入、銷售額）。
- 平方根 / 立方根轉換
  - 温和壓縮變異性，適用於中度偏態資料。
- Box-Cox / Yeo-Johnson 轉換
  - 自動尋找最適指數轉換參數，將資料近似常態化。

**（3）類別資料處理**

在進行機器學習建模時，大多數演算法（如邏輯迴歸、支援向量機、神經網路）無法直接處理類別變數（Categorical Variables），因此需先將這些非數值型特

徵進行類別資料轉換（Categorical Encoding），使其可被模型辨識與運算。不同的編碼方法會影響模型效能、過擬合風險與解釋性，選擇編碼策略時應依據類別性質（是否具順序）、類別數量（基數）、模型類型與資料量進行評估。

以下列出常見類別變數編碼方法：

- Label Encoding（標籤編碼）
  - 將每個類別對應到一個整數編號，例如：「小學」→ 0、「高中」→ 1、「大學」→ 2。
  - 適用情境：類別具有明確順序關係（Ordinal Variables），如教育程度、服務等級（Basic / Premium / VIP）。
  - 優點：轉換快速、佔用空間小。
  - 風險：若誤用於無序類別（Nominal Variables），模型可能誤解為數值之間具有數學意義。
- One-hot Encoding（獨熱編碼）
  - 為每個類別新增一個欄位，該類別為 1，其餘為 0。
  - 例如「紅 / 藍 / 綠」會變成三個欄位：is_red、is_blue、is_green。
  - 適用情境：無序類別變數（如城市名稱、產品類型）。
  - 優點：保留類別的完整資訊，不引入順序誤解。
  - 限制：類別數量多（High Cardinality）時會大幅增加特徵維度，造成記憶體消耗與模型訓練延遲。
- Target Encoding（目標編碼/平均編碼）
  - 將每個類別以其在目標變數上的統計量（如平均值、中位數、轉換率）取代。例如針對每個「廣告來源」，以該來源的平均轉換率作為新特徵。
  - 適用情境：高基數類別變數，且類別與目標變數具高度關聯。
  - 優點：能保留目標相關資訊，不造成維度爆炸。

### （4）時間與週期性資料轉換

時間型資料在許多應用中具有高度資訊價值，透過適當轉換，可轉化為對模型有意義的特徵輸入：

- 拆解時間欄位為結構化特徵：
  - 原始的時間戳記可分解為「年、月、日、星期幾、時段」等，這些元素往往與目標變數具有潛在關聯（例如銷售高峰常出現在假日或下班時段）。
- 建構週期性特徵表示：
  - 為保留時間的「週期性結構」(如一週七天的循環)，可使用三角函數進行週期性編碼。
  - 例如：對週期性欄位（如星期幾、月份）使用週期性編碼（sin / cos 轉換），以保留相鄰關係。

## 5. 資料增強

資料增強（Data Augmentation）方法透過產生額外的新樣本或對現有樣本進行變形，增加資料量、改善資料平衡，進一步提升模型的泛化能力與穩定性。

依照資料型態分類，常見方法包含：

- 圖像資料增強（Image Augmentation）
  - 隨機翻轉、旋轉、裁剪、縮放、色彩變換。
  - 案例：以影像辨識模型訓練為主的場景(如人臉辨識、醫學影像分析)。
- 文字資料增強（Text Augmentation）
  - 同義字替換、隨機插入、隨機刪除、隨機交換字詞位置。
  - 案例：適用於情感分析、主題分類、對話模型的訓練。
- 時序資料增強（Time-series Augmentation）
  - 透過增加噪聲、局部時段調整（Scaling, Jittering）、窗口裁切等方式擴充資料量。
  - 案例：設備故障預測、股市趨勢分析。

- 表格式資料增強（Tabular Data Augmentation）
  - SMOTE（Synthetic Minority Oversampling Technique）增加稀少樣本。
  - 案例：用於不平衡資料分類問題（如詐欺偵測、疾病診斷）。

## 6. 特徵工程策略

特徵工程並非僅是技術操作，更涉及資料理解、領域知識與建模目標的整合。並非單純地套用預設的演算法或工具，而是一個有目的、有規劃、需要深思熟慮的決策過程。常見的思考脈絡包括：

- 依任務類型設計特徵
  - 分類任務偏好具離散分群能力的特徵（如類別指標、區間編碼）。
  - 迴歸任務則偏好與數值趨勢密切相關的連續特徵。
- 依模型性質調整特徵處理
  - 線性模型需特別注意尺度與共線性。
  - 樹模型對類別編碼敏感，避免使用標籤編碼造成誤解。
  - 距離式模型（如 KNN）需保證特徵間單位一致性。
- 探索資料中的隱含結構
  - 將原始欄位進行合成（如「單價 × 數量」生成總價）。
  - 利用統計聚合生成群體行為特徵(如「使用者在過去 7 天的點擊次數」)。
- 考慮時間性與序列關聯
  - 提取滯後值（lag）、移動平均（rolling mean）等序列特徵。
  - 加入時間間隔、事件次數等動態指標。

## 重點掃瞄

# 5.2 模型選擇與架構設計

### 1. 前言與章節導覽

在機器學習專案的實務推動過程中，選擇適合任務的模型與設定適當的模型架構，是模型表現與產出符合關鍵因素。模型選擇與架構設計不僅需考量資料規模、特徵性質與任務類型，還必須平衡模型複雜度、運算資源限制，以及實務應用場景中對模型解釋性的需求。

隨著機器學習演算法的發展日益成熟，從簡單的線性模型、樹模型，到深度學習的複雜神經網路，模型架構的選擇範圍變得更廣、更靈活，但也更具挑戰性。

### 2. 模型選擇的原則與考量因素

模型選擇（Model Selection）在機器學習建模流程中，直接決定了後續訓練成效、泛化能力，以及模型部署後的可用性與效益。正確且適當的模型選擇，需兼顧多個考量因素，包括資料的特性、任務需求、模型的解釋性，以及運算資源與部署限制。

以下分別介紹模型選擇過程中，幾項常見考量因素與相關原則。

#### （1）任務類型與模型特性配對

模型選擇應根據任務的輸出型態、資料特徵與業務目標進行判斷。以下說明不同任務類型下的模型配適原則：

- 分類任務（Classification）：
  - 目標：
    - 預測資料所屬的離散類別。
  - 常見情境：
    - 垃圾郵件偵測、疾病診斷、客戶流失預測。

- 常用模型：
  - 決策樹、隨機森林：具解釋性，適用於特徵混合且分佈複雜的資料。
  - 支援向量機（SVM）：適合邊界清晰、高維小樣本的分類問題。
  - 神經網路（Neural Network）：當特徵具高度非線性或為影像、語音等資料類型，模型預測結果佳。

- 迴歸任務（Regression）：
  - 目標：
    - 預測一個連續數值輸出。
  - 常見情境：
    - 房價預測、能源消耗預測、業績估算。
  - 常用模型：
    - 線性迴歸（Linear Regression）：簡單且高解釋性，適合線性趨勢明顯的資料。
    - 決策樹迴歸、隨機森林迴歸：對於非線性與特徵交互效果有良好處理能力。
    - 神經網路：適合處理大規模、高維度或非線性強烈的資料情境。

- 非監督學習任務（Unsupervised Learning）：
  - 目標：
    - 從未標註資料中發掘潛在結構或壓縮表示。
  - 任務類型與建議模型：
    - 聚類（Clustering）
      - 目標：將資料自動分組，使群內相似、群間差異大。
      - 常用模型：
        - K-means：資料呈球狀且群數已知時效果良好。
        - DBSCAN：適合有噪聲或群大小不均的資料。
    - 降維（Dimensionality Reduction）
      - 目標：

- 將高維資料轉換為低維表示，保留主要資訊結構。

○ 常用模型：

- PCA（主成分分析）：保留最大變異方向，具解釋性。
- 自編碼器（Autoencoder）：適合非線性降維與結構重建需求。

- 序列與時間序列任務（Sequential / Time-series）
  - 目標：
    - 根據資料序列預測未來或決策行為。
  - 常見情境：
    - 股價預測、感測器數據監控、語音辨識。
  - 常用模型：
    - RNN / LSTM / GRU：
      - 適合處理長序列記憶與依賴關係。
    - 時序卷積網路（Temporal Convolutional Network, TCN）：
      - 對長期依賴具有穩定性。
    - ARIMA 等統計模型：
      - 適合短期、週期性明顯且資料量不大之應用。

**（2）資料規模與模型選擇**

資料量不僅影響模型訓練的可行性，更牽動模型選擇的策略方向。不同資料規模下，模型在表現力、穩定性與訓練效率之間需做出平衡。

- 小型資料集（數百至數千筆樣本）
  - 當樣本有限時，選擇結構簡單、參數數量少的模型能有效降低過擬合風險。例如線性迴歸、邏輯迴歸、決策樹或正則化模型（如 Lasso、Ridge）為常見選項。
  - 此時資料前處理與特徵工程的重要性提高，需以較強的先驗假設支撐模型效果。

- 中型資料集（數千至數十萬筆樣本）
  - 此區間提供模型一定的學習能力與泛化基礎，能夠支持較高表現力的模型如隨機森林、梯度提升機（XGBoost、LightGBM）等。
  - 可進一步嘗試模型集成與超參數優化，提升預測效能。同時，計算資源與訓練時間，會成為實務上必須考量的限制因素。
- 大型資料集（百萬級以上樣本）
  - 大規模資料能發揮深度模型的表現潛力，特別是在非結構化資料（如影像、語音、文字）處理上，深度神經網路成為主要選擇。
  - 大型資料訓練也對運算能力、分散式架構支援、模型調校效率有更高要求。此時應選擇具擴展性且可配合 GPU/多機訓練的架構設計。

**（3）模型解釋性需求**

模型的可解釋性決定了其在特定領域的適用性，特別是在醫療、金融、法遵等需追溯模型決策邏輯的情境下，更是不可或缺。

- 高解釋性模型
  - 線性迴歸（Linear Regression）、邏輯迴歸（Logistic Regression）與淺層決策樹（Shallow Decision Tree）等模型具備結構簡單、邏輯可視化的特性。其決策過程明確，變數對預測結果的影響可透過係數、分割規則等方式直接解釋，便於使用者進行結果溝通與審查。
  - 此類模型常被應用於對可解釋性有明確要求的場域，如醫療診斷、信貸評分與法規監管等情境。
- 低解釋性模型
  - 隨機森林（Random Forest）、梯度提升樹（Gradient Boosting Trees）與深度神經網路（Deep Neural Networks）等模型雖具備高度預測能力，但其結構高度非線性且包含大量參數，使得模型的決策過程不易直觀理解。

- 在醫療、金融或法規敏感等高風險應用場景中，建議搭配模型可解釋性技術使用，例如 SHAP（SHapley Additive exPlanations）、LIME（Local Interpretable Model-Agnostic Explanations）或偏依圖（Partial Dependence Plot），以輔助理解模型對特徵的依賴關係與輸出邏輯，進而提升模型透明度與可信度。

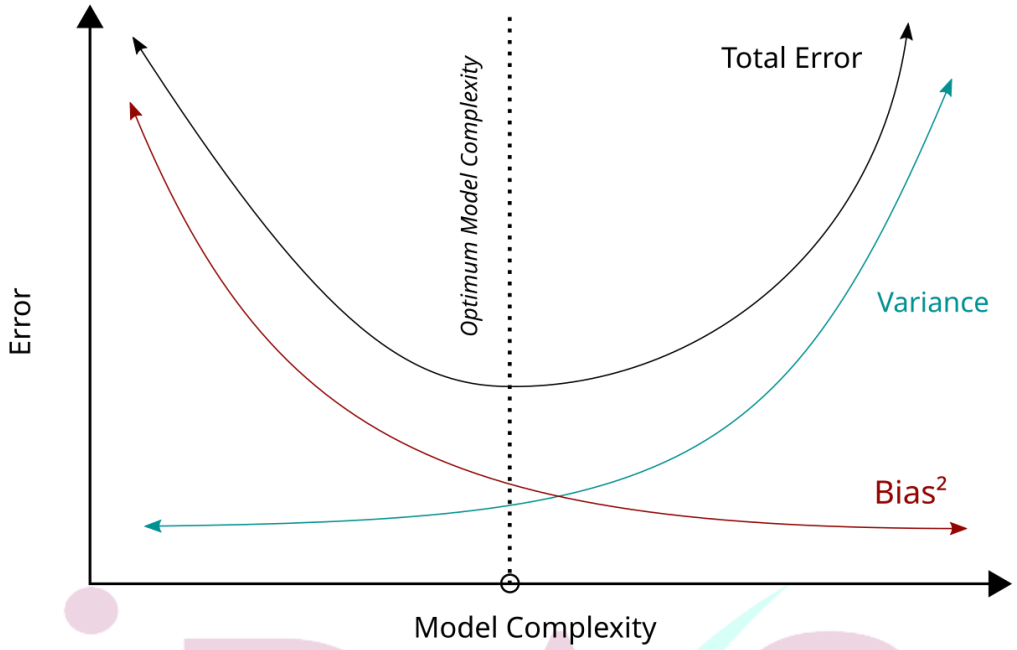
**（4）運算資源與實務部署限制**

實務應用中，模型部署往往受限於資源環境與即時反應需求。選擇模型時，除理論效能外，也應考量部署現實與維運成本。

- 資源受限場景（如 IoT、手機 App、邊緣裝置）
  - 適合選用模型結構簡單、參數少、推論快速的演算法，例如簡單決策樹、邏輯迴歸或經壓縮後的小型神經網路。亦可採用模型剪枝或量化等技術進行模型瘦身。
- 資源充足場景（如雲端部署、資料中心）
  - 可使用高效能深度模型、模型集成策略，充分發揮硬體計算能力。同時能支援較複雜的資料前處理與後處理流程。
- 即時推論需求
  - 在必須秒級或毫秒級回應的任務中（如金融交易、推薦系統），須優先考量模型的推論延遲與效能表現。可搭配 caching、模型蒸餾（knowledge distillation）等技術提升即時表現。

模型選擇與部署應同步規劃，避免訓練完成後才發現模型無法實際上線執行。

### （5）偏差-變異的權衡（Bias-Variance Tradeoff）




模型選擇的核心原則之一是權衡偏差（bias）與變異（variance），兩者之間存在固有的取捨關係：

- 偏差（Bias）：
  - 指模型對資料結構的擬合能力不足，無法有效捕捉資料中的主要趨勢，導致系統性誤差偏高（欠擬合）。
- 變異（Variance）：
  - 指模型對訓練資料的細微差異或雜訊過於敏感，導致模型在不同資料集上的表現差異大、泛化能力差（過擬合）。

模型複雜度的提升通常伴隨偏差的降低與變異的提高，因此模型選擇應在二者之間取得適當平衡：

- 低複雜度模型（如線性迴歸）：
  - 偏差較高、變異較低，穩定且具解釋性，但無法處理複雜關係。
- 高複雜度模型（如深度神經網路）：
  - 偏差低、變異高，可擬合複雜模式，但需更多資料或額外的正則化方法以控制變異。

## 重點掃瞄

# 5.3 模型訓練、評估與驗證

### 1. 前言與章節導覽

在機器學習開發流程中，即便選擇了合適的模型架構與演算法，若訓練過程未能有效執行、評估方法不嚴謹，仍可能導致模型在實際應用中表現不佳，甚至產生錯誤決策。

本節聚焦於模型訓練過程中的核心技術與方法，包括訓練資料的使用方式、模型學習策略的選擇、效能評估指標的設計，以及模型穩定性與泛化能力的檢驗流程。同時也強調透過視覺化輔助進行結果理解與分析，協助使用者掌握模型學習行為與預測邏輯。

### 2. 模型訓練流程與策略

模型訓練是構建機器學習系統中關鍵的運作環節，負責依據資料調整模型參數，使其能有效對應輸入與預測目標之間的關係。適當的訓練策略不僅有助於加速收斂過程，也能提升模型在未見資料上的泛化能力，降低過擬合風險。本節將說明模型訓練的主要流程與實務常用策略，並解析其對整體模型效能的影響。

#### （1）資料分割與準備

為評估模型在未見資料上的表現，常需將原始資料集依功能目的劃分為以下部分：

- 訓練集（Training Set）：
  - 用於模型參數的學習與內部結構調整，是整個學習流程的基礎。
- 驗證集（Validation Set）：
  - 作為調整超參數（如學習率、正則化係數）與監控訓練過程的依據，用以觀察模型的泛化能力。

- 測試集（Test Set）：
  - 僅在模型訓練完成後使用，用來進行最終效能評估，模擬模型實際部署時對未知資料的表現。
- 當資料量充足時，亦可進行 K-fold 交叉驗證，藉由輪流使用不同子集作為驗證資料，提升評估的穩定性與可信度。
  - K-fold 交叉驗證
    - K-fold 交叉驗證（K-fold Cross-Validation）是一種常用的模型評估技術，主要用於更穩健地評估模型的泛化能力，尤其當資料量有限時特別有用。
    - K-fold 交叉驗證會將原始資料集平均劃分為 K 個不重疊的子集（folds），然後進行 K 次訓練與驗證迭代。每一次：
      - 使用其中 K−1 個子集做為訓練資料。
      - 剩下的 1 個子集作為驗證資料。
      - 這個過程重複 K 次，每一個子集都會被當作一次驗證集。最終將 K 次驗證結果進行平均，作為模型的整體表現評估。

**（2）批次訓練設計與更新策略**

資料如何輸入模型，將直接影響學習速度、梯度估計的穩定性與資源使用效率：

- 全量訓練（Batch Gradient Descent）：
  - 每輪迭代使用全部資料進行梯度計算，優點是方向穩定、收斂路徑平滑，但對記憶體要求高，且不易應用於大規模資料。
- 隨機梯度下降（Stochastic Gradient Descent, SGD）：
  - 每次僅使用一筆樣本進行參數更新，適合線上學習與資料流架構，惟更新震盪較大，收斂速率不穩。

- 小批次訓練（Mini-batch SGD）：
  - 每次使用固定筆數樣本進行更新，兼具全量與隨機策略優點，是目前深度學習最常用的方法。
  - Mini-batch 的大小會影響梯度估計的準確性與訓練速度，選擇上需視 GPU 記憶體與任務特性進行調整。

**（3）學習率調整**

學習率（Learning Rate）控制模型每次更新的步伐，是訓練過程中重要的超參數之一。常見策略如下：

- 固定學習率（Constant Rate）：
  - 設定單一值，適用於簡單任務，但在收斂後期可能難以進一步提升效能。
- 遞減學習率（Step Decay / Exponential Decay）：
  - 根據訓練次數或驗證集表現，定期降低學習率，有助於穩定收斂。
- 動態調整：
  - 當驗證效能停滯時自動調降學習率，可精細控制學習節奏。
  - 如 Pytorch 內的 ReduceLROnPlateau。
- 預熱策略：訓練初期使用較低學習率，逐漸升高，有助於避免初期梯度爆炸，常見於 Transformer 類模型。

**（4）早停策略與訓練終止準則**

若訓練過久，模型可能過擬合訓練資料。Early Stopping 是透過監控驗證集效能，判斷訓練何時應中止：

- 設定容忍次數（Patience）：
  - 驗證指標若在連續 N 次迭代內無明顯改善，則終止訓練。
- 設定最小改善幅度（Minimum Delta）：
  - 效能提升若小於閾值，亦可視為無效進步。

- 配合學習率調整共同使用，能取得穩定且泛化良好的模型。

### （5）訓練過程的記錄與監控

有效的訓練流程需要全程監控與可追蹤性（Reproducibility），建議結合以下工具與紀錄項目：

- 訓練指標視覺化：
  - 使用 TensorBoard、WandB 或 MLflow 觀察損失曲線與指標趨勢。
- 超參數與模型版本管理：
  - 記錄每次訓練的超參數設定、模型結構與權重版本，便於複製與回溯。
- 資源使用狀況：
  - 追蹤 GPU、CPU 與記憶體使用率，評估效能瓶頸與部署可行性。

## 3. 評估指標與模型效果衡量

模型訓練完成後，必須透過驗證集與測試集進行效果評估，以衡量其泛化能力與實務表現。不同的任務類型（如分類、迴歸）對應不同的評估指標，每種指標各自反映模型在準確性、穩定性或風險控制上的特定面向。

### （1）分類任務的評估指標

以下是二元分類問題中的混淆矩陣（Confusion Matrix）」標準格式，以「正類（Positive）」和「負類（Negative）」作為預測與實際的對照。

| | 實際為正類（Positive） | 實際為負類（Negative） |
|---|---|---|
| 預測為正類 | 真正（TP）<br>（True Positive） | 假正（FP）<br>（False Positive） |
| 預測為負類 | 假負（FN）<br>（False Negative） | 真負（TN）<br>（True Negative） |

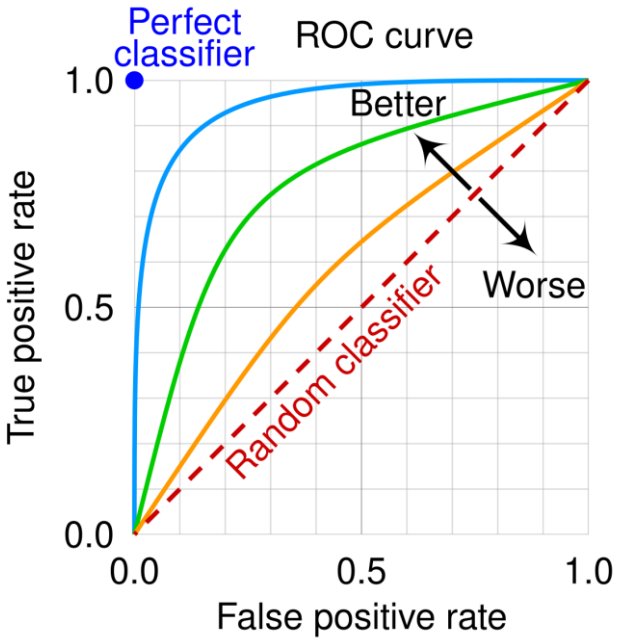
- P：實際為正類的樣本數
- N：實際為負類的樣本數

- TP：真正（True Positive）
- TN：真負（True Negative）
- FP：假正（False Positive）
- FN：假負（False Negative）

以下是常見分類任務評估指標的公式彙整，搭配說明與應用情境：

- 準確率（Accuracy）
  - 公式：$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$
  - 定義：正確預測樣本數佔總樣本數的比例。
  - 適用：類別分佈相對均衡時效果良好。
  - 限制：在嚴重類別不平衡的資料中容易誤導，例如 95%樣本為負類時，即使模型全預測為負，準確率仍高達 95%。
- 精確率（Precision）
  - 公式：$\text{Precision} = \frac{TP}{TP+FP}$
  - 定義：被預測為正類的樣本中，實際為正類的比例。
  - 適用場景：當「誤報正類」的代價高時（例如垃圾郵件分類、醫療誤診）。
  - 目的：衡量「預測為正的可信度」。
- 召回率（Recall）
  - 公式：$\text{Recall} = \frac{TP}{TP+FP}$
  - 定義：實際正類樣本中，被正確預測為正類的比例。
  - 適用場景：當「漏報正類」的代價高時（例如癌症偵測、詐騙偵測）。
  - 目的：衡量「模型能抓住多少實際正類」。
- F1 分數（F1-Score）
  - 公式：$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision}+\text{Recall}}$
  - 定義：精確率與召回率的調和平均。
  - 適用場景：需在精確率與召回率間取得平衡，且類別不平衡嚴重時。

- ROC 曲線與 AUC




  - ROC 曲線（Receiver Operating Characteristic）：
    - 橫軸：假正率。
    - 縱軸：真正率畫出的曲線。
    - AUC（Area Under Curve）：表示曲線下的面積。
  - 意義：
    - 代表模型在所有可能閾值下的整體區分能力。
    - AUC 值越接近 1，模型越好。
  - 適用：需綜觀整體預測能力時，特別是需調整分類閾值的應用。

**（2）迴歸任務的評估指標**

迴歸任務的目標為預測連續變數，模型評估需著重於預測值與實際值之間誤差的大小、穩定性與擬合程度。以下為常見指標及其特性說明：

- 均方誤差（Mean Squared Error, MSE）
  - 公式：$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$
    - $y_i$：為實際值

- $\hat{y}_i$：為預測值

- 定義：所有預測誤差平方的平均值。
- 特點：對大誤差高度敏感，會放大極端偏差的影響。
- 適用：需強調大誤差懲罰的任務，如金融風險預測或製程品質監控。

- 平均絕對誤差（Mean Absolute Error, MAE）
  - 公式：$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$
    - $y_i$：為實際值
    - $\hat{y}_i$：為預測值
  - 定義：所有預測誤差絕對值的平均。
  - 特點：對異常值較不敏感，提供穩定的誤差估計。
  - 適用：資料具偏態分佈或含少量極端值時。
- 均方根誤差（Root Mean Squared Error, RMSE）
  - 公式：$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$
    - $y_i$：為實際值
    - $\hat{y}_i$：為預測值
  - 定義：MSE 的平方根。
  - 特點：保有 MSE 的懲罰特性，同時回到與預測變數相同的單位。
  - 適用情境：作為模型精度的整體衡量，廣泛用於報告與模型比較。
- 決定係數（$R^2$ Score）
  - 意義：迴歸任務中用以衡量模型對目標變數變異解釋能力的重要指標。
  - 公式：
    $$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} = 1 - \frac{RSS}{TSS}$$
    - $y_i$：為實際值
    - $\hat{y}_i$：為預測值
    - $\bar{y}$：為實際值的平均數

- 分子：

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

  - 模型殘差平方和（Residual Sum of Squares, RSS）
  - 為資料點與模型預測值之間的差距所產生的平方總和。
  - RSS 越小，表示模型預測越接近實際資料。

- 分母：

$$TSS = \sum_{i=1}^{n}(y_i - \bar{y})^2$$

  - 總變異平方和（Total Sum of Squares, TSS）
  - 資料點與平均值之間的差距所產生的平方總和，代表資料的總變異量。
  - TSS 用於當作模型比較的基準。

◆ 適用情境：

- 適用於線性迴歸模型，可快速評估預測能力。
- 不適用於非線性或未標準化資料下的模型比較，易失真。
- 對於不同任務或資料集間的 $R^2$無法直接比較。

◆ 判別：

- $R^2 = 1$：表示模型能完全解釋資料變異，預測完美。
- $R^2 = 0$：表示模型僅與常數模型（如直接預測平均值）同等表現。
- $R^2 < 0$：表示模型比常數模型還差，可能發生在模型嚴重偏離資料趨勢或過度擬合下。

### （3）模型比較與綜合評估策略

在實務應用中，評估一個模型的優劣不應僅依賴單一指標，而需從多面向整合考量，以兼顧準確性、穩定性與應用風險。以下列出常見的模型評估策略：

- 指標組合與多角度觀察：單一評估指標往往無法全面呈現模型效能與偏誤風險。建議依據任務特性組合使用多項指標：
  - 分類任務：結合準確率（Accuracy）、F1 分數、ROC-AUC、混淆矩陣等指標，可避免單一指標掩蓋類別偏誤。
  - 迴歸任務：可綜合 MAE、RMSE 與 $R^2$，觀察誤差分佈與模型擬合能力。
- 類別不平衡處理與評估策略：面對高度不平衡資料（如欺詐偵測、醫療診斷），傳統準確率常導致誤判。可搭配資料處理技術如：
  - 類別重加權（Class Weights）
    - 模型訓練時對少數類別賦予更高權重，強化模型對其的學習能力。常用於邏輯迴歸、樹模型與神經網路中，透過損失函數加權自動調整學習焦點。
  - 過採樣（如 SMOTE）
    - SMOTE（Synthetic Minority Over-sampling Technique）透過合成新樣本方式平衡資料分佈，能保留原始樣本空間結構，降低過度複製產生的過擬合風險。適用於資料量不足的少數類別。
  - 異常值建模（如 Isolation Forest、One-Class SVM）
    - 將類別不平衡問題視為異常偵測任務，建構僅學習正常類別行為的模型，再以此辨識「異常」樣本。特別適用於極端不平衡場景（如欺詐偵測、設備故障預測等）。
- 業務導向與自定義效能衡量
- 在特定業務場景中，應設計實際風險與成本結構的客製指標：
  - 風險導向任務（如醫療）：可自定義誤判成本矩陣，強化高代價錯誤（如漏診）的懲罰。
  - 商業任務（如推薦系統、行銷）：可使用領域專屬指標，如排序任務中的 NDCG、收益導向的 Profit Score。

- 多輪驗證與效能穩定性觀察：模型效能的評估不應僅依賴單一實驗結果或平均數值，更應關注其在多次訓練下的穩定性與變異範圍，以反映模型在不同資料切分條件下的可靠性。常見方法如下：
  - K-fold 交叉驗證（K-fold Cross-Validation）：
    - 將資料集平均分為 K 份，輪流以其中一份作為驗證集，其餘作為訓練集，重複 K 次以取得平均效能。能有效減少資料分割偶然性對模型評估造成的偏誤。
  - 重複交叉驗證（Repeated K-fold CV）：
    - 在進行 K-fold 的基礎上多次隨機重複切分流程，可進一步衡量模型在隨機條件下的穩定性與泛化能力，適合評估小樣本或高變異任務。
  - 穩定性視覺化分析：
    - 搭配箱型圖（Boxplot）或平均排名圖（Mean Rank Plot）等視覺化工具，觀察不同模型在多輪驗證下的表現分佈與變異程度。這有助於篩選出在不同資料條件下都具一致性與穩定性的候選模型。

## 4. 交叉驗證

交叉驗證（Cross-Validation）是一種重要的模型評估技術，目的在於更真實地衡量模型的泛化能力，避免因為單次資料切分所產生的偏誤。透過將資料多次切分並反覆進行訓練與測試，可以有效觀察模型在不同樣本組合下的穩定性，作為模型選擇與調參的依據。以下介紹常見的交叉驗證方法：

### （1）K-fold

K-fold 交叉驗證（K-fold Cross-Validation）是最常見的通用型交叉驗證方法。其目的是減少評估受特定資料分割影響的偶然性，提供穩健且具有代表性的模型效能估計。

- 過程：
  - 將資料集平均劃分為 K 個不重疊的子集（folds），每次選定其中 1 折（fold）作為驗證集，其餘 K-1 折作為訓練集。
  - 重複 K 次後計算指標平均作為整體表現。
- 特點：
  - 減少資料切分偏差，適用性廣。
  - 能有效評估模型在不同樣本上的表現穩定性。
  - 每次需重新訓練模型，計算成本為 K 倍。
  - 常見設定為 K=5 或 10，可在精確度與運算效率間取得平衡。
- 適用場景：
  - 適用於中大型資料集（數千筆以上）。
  - 模型選擇與效能比較常以此為標準方法。
  - 適用於迴歸與分類任務的泛化能力驗證。

**（2）Stratified K-fold**

Stratified K-fold（分層 K 折交叉驗證）是針對分類問題的改良版本，於劃分時確保每一折中各類別的比例與整體資料集相符，特別適用於類別分佈不均的資料（如欺詐偵測、疾病分類等）。

- 過程：
  - 依照類別比例進行分層抽樣，使每一折的類別比例與整體資料集相近。
- 特點：
  - 可有效解決類別不均所導致的模型評估失真。
  - 提升對小樣本類別的穩定性與預測準確度。
  - 與傳統 K-fold 相比，分佈一致性更高。
- 適用場景：
  - 分類資料中存在不均衡現象（如正負樣本比例懸殊）。
  - 用於詐騙偵測、醫療診斷、異常事件預測等高風險分類問題。

- 可作為分類任務交叉驗證的預設方式。

### （3）LOOCV

LOOCV（Leave-One-Out Cross-Validation）是 K-fold 的極端形式。K 等於樣本數 n，每次僅留下一筆樣本作為驗證、其餘 n-1 筆資料訓練模型，總共進行 n 次評估。

- 過程：
  - 資料集中每次僅留下 1 筆資料作為驗證集，其餘所有資料用於訓練，重複進行 n 次（n 為樣本數）。
- 特點：
  - 評估偏差最小，適合樣本珍貴或不可浪費的情境。
  - 因需訓練 n 次模型，計算成本極高，對模型複雜度與硬體要求較高。
  - 敏感於訓練集的微小變動，模型表現波動較大。
- 適用場景：
  - 樣本數極小但資料珍貴（如臨床研究、稀有病資料）。
  - 為提高模型評估的穩定性與可信度，重複 K-fold 在每次驗證前隨機劃分資料，以捕捉樣本劃分隨機性的影響。
  - 學術研究中需最大化資料利用與精準驗證。
  - 適用於須個別樣本可信度高的模型精度檢查。

### （4）Repeated K-fold

重複 K-fold（Repeated K-fold Cross-Validation）是在標準 K-fold 基礎上進行多次隨機重劃，反覆交叉驗證以取得多組評估結果，再計算平均與變異數，提升模型穩定性觀察。

- 過程：
  - 執行多輪 K-fold 驗證（如 10-fold × 5 次），每輪隨機重分 fold，計算所有輪的平均與變異。

- 特點：
  - 評估結果更加穩定與具代表性。
  - 可觀察模型在不同劃分下的表現波動與可信區間。
  - 相較標準 K-fold，計算成本更高。
- 適用場景：
  - 調參流程中模型比較與效能穩定性評估。
  - 學術研究或論文發表需報告標準差與置信範圍。
  - 需重現性高的產業建模流程，如金融風控或醫療 AI。

## 重點掃瞄

# 5.4 模型調整與優化

### 1. 前言與章節導覽

模型調整與效能優化是機器學習流程中從實驗走向應用的關鍵階段。即使完成模型選擇與初步訓練，實務上仍常面臨效能不穩、推論延遲、過擬合或資源消耗過高等問題，亟需透過精細的調整與優化策略進行改善。

本節聚焦於模型建構後的優化行動，內容涵蓋以下幾個核心面向：

- 超參數調校（Hyperparameter Tuning）：
  - 透過系統化方法調整模型的非學習參數，以獲得最佳效能。
- 正則化技術（Regularization）：
  - 減少模型過度擬合，提升泛化能力。
- 資料增強與重取樣（Data Augmentation & Resampling）：
  - 提高訓練資料多樣性與平衡性，改善模型穩定度。
- 模型壓縮與加速（Model Compression & Acceleration）：
  - 降低模型大小與推論時間，以利於部署至終端設備或邊緣裝置。

### 2. 超參數調校

機器學習與深度學習模型的表現，除了依賴資料與模型架構外，超參數（Hyperparameters）的設定更對訓練效率與泛化能力有重大影響。在機器學習建模過程中，超參數是需由開發者手動指定的參數，並不由資料自動學習得出。對模型的訓練穩定性、收斂速度與最終效能影響深遠。以下列出常見的超參數：

### （1）學習率

- 定義
  - 學習率（Learning Rate）是控制模型在每一次反向傳播（Backpropagation）後，根據梯度方向更新參數的幅度。
  - 學習率並非模型內部自我調整的參數，而是由訓練者事先指定的核心超參數。
  - 學習率的設定將直接影響訓練過程中的收斂效率、模型穩定性與最終效能，是所有深度學習流程中最敏感且具關鍵性的設定之一。
- 適用範圍
  - 學習率主要應用於採用梯度下降（Gradient Descent）或其變形演算法（如 Mini-Batch SGD、Momentum、Adam 等）的模型，尤其是深度學習架構，如：
    - 卷積神經網路（CNN）
    - 遞迴神經網路（RNN）
    - 自注意力模型（如 Transformer）
  - 在某些傳統機器學習方法（如使用 SGDClassifier 的線性模型）中也可能需手動設定學習率，但整體使用比例明顯低於深度學習。
- 作用機制
  - 學習率可視為模型在損失函數空間中「走多快」的控制因子。其作用邏輯如下：
    - 學習率大：
      - 參數每次更新幅度大，能快速接近最小值，但也容易跳過最小點或導致震盪，甚至使損失函數無限上升（發散）。
    - 學習率小：
      - 參數變化細緻，有利於穩定收斂，但可能導致學習速度極慢，特別是在高維空間中難以跳脫局部極小值。

- 在訓練初期，適度大一點的學習率可快速降低誤差；在訓練後期，降低學習率則有助於收斂至更佳的參數區域，因此實務上常會搭配學習率退火（Annealing）或調度器（Scheduler）。

- 常見問題
  - 學習率設定過高：
    - 造成模型參數大幅變動，導致訓練不穩定，甚至出現梯度爆炸。
  - 學習率設定過低：
    - 收斂速度緩慢，無法有效跳出局部極小值，導致訓練無效或資源浪費。
  - 固定學習率：
    - 未隨訓練階段調整，無法兼顧訓練初期快速下降與後期細部微調的需求。
  - 未監控學習率：
    - 忽略調整會導致模型卡在低效學習區段，錯失潛在表現提升空間。

**（2）批次大小**

- 定義
  - 批次大小（Batch Size）是指每次參與梯度更新的訓練樣本數量。與全量訓練不同，批次訓練將資料拆分為多個小群組，逐批輸入模型進行參數更新。此參數決定了單次反向傳播所見的樣本規模，是訓練流程中的核心設計之一。
- 適用範圍
  - 批次大小主要用於深度學習模型的訓練流程，尤其是在使用 GPU 進行加速時更為關鍵。
  - 在傳統機器學習中，大多數演算法採用全量訓練（batch learning），不涉及批次大小設定，僅少數使用 SGD 或 mini-batch 方法的情境會使用此參數。

- 作用機制
  - 批次大小會影響梯度估計的穩定性、模型更新的頻率、以及訓練資源的使用效率。
  - 小批次
    - 提供較高的隨機性，有助於跳脫局部極小值，但梯度波動大，可能使訓練不穩定。
  - 大批次
    - 則提供較穩定的梯度估計，有利於加速收斂，但可能降低泛化能力。
    - 批次大小也影響記憶體使用與硬體資源分配，是訓練效率與模型表現之間的重要平衡點。
- 常見問題
  - 批次過小：
    - 導致梯度估計不穩，訓練過程震盪劇烈，並增加訓練時間與迭代次數。
  - 批次過大：
    - 雖然提升運算效率，但可能降低模型泛化能力、收斂速度下降，並容易陷入局部最小值。
  - 未考慮硬體限制：
    - 若批次設定超過 GPU 記憶體上限，將導致訓練失敗或需頻繁調整。
  - 未搭配學習率調整：
    - 批次大小與學習率具高度耦合，若批次放大而未調整學習率，可能導致訓練無效或不穩。

### （3）網路深度與寬度

- 定義
  - 網路深度（Network Depth）是指神經網路中所包含的層數，包含輸入層、隱藏層與輸出層。

- 網路寬度（Network Width）則是指每層中神經元（或通道、單元）的數量。
- 兩者共同決定模型的結構複雜度與參數規模，是神經網路表達能力的基礎構面。

- 適用範圍
  - 此參數僅適用於深度學習模型，尤其是結構可調整的類型，如全連接網路（MLP）、卷積神經網路（CNN）、遞迴神經網路（RNN）與轉換器架構（Transformer）。
  - 傳統機器學習演算法（如決策樹、SVM）大部分無深度與寬度可調的概念。
- 作用機制
  - 增加深度
    - 可提升模型的非線性表達能力，使其能處理更複雜的模式辨識任務。
  - 增加寬度
    - 提供更豐富的特徵表示空間，有助於捕捉多樣化輸入資訊。
  - 深度與寬度的組合影響模型的容量（capacity），但若無適當正則化與訓練策略，過多的層數與神經元將導致參數爆炸、梯度消失或梯度爆炸問題。
- 常見問題
  - 結構過深：
    - 若未搭配合適初始化、Batch Normalization 或殘差連接，容易出現梯度消失，使模型難以收斂。
  - 結構過寬：
    - 模型容量過大，容易對訓練資料過擬合，缺乏泛化能力。
  - 結構過淺：
    - 表達能力不足，導致欠擬合，無法學習複雜資料特徵。

- 無規劃堆疊層數：
  - 單純堆疊層數並不保證效果提升，應根據任務特性與資料量進行設計。
- 計算成本增加：
  - 深層與寬層模型通常需要更多訓練時間與硬體資源，若未考量部署限制，將造成效能瓶頸。

**（4）激活函數**

- 定義：
  - 激活函數（Activation Function）是神經網路中用來將神經元加權輸入轉換為輸出的非線性函數。
  - 激活函數決定了神經元的反應方式，並賦予模型處理非線性問題的能力，是深度學習中不可或缺的核心組件。
- 適用範圍：
  - 激活函數應用於所有深度學習架構的隱藏層與輸出層。
  - 傳統機器學習模型通常不需明確指定激活函數，僅神經網路相關方法才會涉及。
- 常見激活函數：不同任務與層級會選擇不同類型的激活函數，例如分類任務輸出層常用 Softmax，二元分類用 Sigmoid，隱藏層則常用 ReLU 或其變體。
  - Softmax
    - 定義與計算
      - Softmax 函數會將一組實數輸入轉換為機率分佈，其輸出值為 0 到 1 之間，且總和為 1。
      - 公式為：

$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_i}}$$

- 適用範圍
  - 多類別單選（Multi-Class Single-Label）。
  - 例如在多分類問題中，只能從多個類別中選出一個最終預測。
- 特點
  - 輸出為機率分佈。
  - 確保類別間相互排斥。
  - 用於輸出層。

◆ ReLU

- 定義與計算
  - ReLU 函數將所有負數輸入設為 0，正值則保持不變。
  - 公式為：

    $\text{ReLU}(x) = \max(0, x)$
- 適用範圍
  - 深度神經網路的隱藏層。
  - 適合捕捉非線性特徵，同時計算效率高。
- 特點
  - 解決梯度消失問題。
  - 計算簡單快速。
  - 輸出不在固定範圍內。

◆ Sigmoid

- 定義與計算
  - Sigmoid 函數將輸入壓縮到 0 到 1 之間。
  - 公式為：

    $\sigma(x) = \frac{1}{1 + e^{-x}}$
- 適用範圍
  - 二元分類。
  - 標籤分類。

- 每個類別都是獨立的二元判斷，不彼此排斥。

- 特點
  - 每個輸出單獨表示機率。
  - 不保證輸出總和為 1。
  - 常用於輸出層。

- Tanh
  - 定義與計算
    - Tanh 函數將輸入壓縮到-1 到 1 之間。
    - 公式為：

      $$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

  - 適用範圍
    - 深度神經網路的隱藏層。
    - 數據需要正負輸出對稱時，較適合使用 Tanh。
  - 特點
    - 輸出範圍是-1 到 1。
    - 較 Sigmoid 能產生中心對稱的效果。
    - 在中心區域梯度較大，能讓模型在這裡學習得更快。

**（5）優化器**

- 定義
  - 優化器（Optimizer）是深度學習中，用來更新模型參數（權重和偏差）的演算法。
  - 優化器依據損失函數（Loss Function）的梯度資訊，調整參數方向與步幅，以最小化損失值並提升模型效能。
  - 不同優化器的演算法邏輯，會直接影響收斂速度、穩定性，以及最終模型表現。

- 適用範圍
  - 優化器幾乎適用於所有需要進行參數學習的機器學習與深度學習模型，尤其是：
    - 深度神經網路（CNN、RNN、Transformer 等）。
    - 使用梯度下降法或其變種的演算法。
    - 部分傳統機器學習演算法如 SGDClassifier（但規模通常較小）。
- 作用機制
  - 計算梯度
    - 根據損失函數對模型參數的偏導數，計算每個參數更新的方向。
  - 決定更新步伐
    - 根據學習率（Learning Rate）及演算法特性，決定每次更新的幅度。
  - 控制收斂過程
    - 是否考慮動量（Momentum）。
    - 是否採用自適應學習率（Adaptive Learning Rate）。
    - 是否累積梯度歷史以修正更新方向。
  - 平衡速度與穩定性
    - 好的優化器能兼顧快速收斂與避免震盪或發散。
- 常見問題
  - 學習率敏感
    - 大多數優化器仍高度依賴學習率的適當設定，過高或過低都可能導致模型效能下降。
  - 陷入局部最小值或鞍點（Saddle Point）
    - 某些優化器容易卡在非全域最小點，影響模型學習完整度。
  - 過度擬合風險
    - 若未搭配適當的正則化或 Early Stopping，收斂過快可能造成過擬合。
  - 不同模型適配性差異
    - 某些優化器適合淺層網路，但不適合深度模型（如單純 SGD）。

- 資源消耗
  - 高階優化器（如 Adam、Adagrad）需要額外的記憶體儲存梯度歷史或二階矩估計，對大模型可能造成額外負擔。

- 常見優化器
  - SGD
    - 定義
      - 隨機梯度下降（Stochastic Gradient Descent, SGD）是最基本的梯度下降方法，每次僅使用一筆或一小批（Mini-batch）資料來計算梯度並更新參數。
      - 公式為：

        $\theta_{t+1} = \theta_t - \eta \cdot \nabla_\theta L(\theta_t)$

        $\theta_t$：當前參數

        $\eta$：學習率

        $\nabla_\theta L$：對損失函數 L 的梯度

        $L(\theta_t)$：只用第 i 筆資料或 mini-batch 計算的 Loss
    - 適用範圍
      - 小型模型或淺層神經網路。
      - 記憶體有限的環境。
      - 資料量較小的應用。
    - 特點
      - 計算簡單、記憶體需求低。
      - 更新快速且頻繁。
      - 易受噪聲影響，收斂速度慢。
      - 高度依賴學習率設定。
  - Momentum
    - 定義
      - Momentum 在 SGD 基礎上，對梯度更新加上過去梯度的累積，產生「慣性」，讓參數更新更平滑，減少震盪。

- 公式為：

  $v_{t+1} = \gamma \cdot v_t + \eta \cdot \nabla_\theta L(\theta_t)$

  $\theta_{t+1} = \theta_t - v_{t+1}$

  $v_t$：累積的動量

  γ：動量係數，通常介於 0.5~0.9

- 適用範圍
  - 深度神經網路。
  - 梯度震盪大的情況。
  - 希望加快收斂速度的應用。
- 特點
  - 減少梯度震盪。
  - 幫助跳脫局部最小值。
  - 收斂速度較快。
  - 需要額外調整動量係數。

◆ Adagrad

- 定義
  - Adagrad 會自動為每個參數調整學習率，讓更新幅度隨參數過往累積的梯度大小而變化，適合用於處理稀疏資料或特徵。
  - 公式為：$G_t = G_{t-1} + \nabla_\theta L(\theta_t)^2$

    $$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{G_t + \epsilon}} \cdot \nabla_\theta L(\theta_t)$$

    G$t$：累積到目前為止的梯度平方和

    ϵ：避免除以零的極小值
- 適用範圍
  - 稀疏特徵場景，如文字、NLP。
  - 特徵分佈不均的情況。
- 特點
  - 為每個參數分配不同學習率。

- 特別適合稀疏特徵。
- 缺點是學習率會持續衰減到非常小，可能導致模型停止學習。

◆ Adam

■ 定義

- Adam（Adaptive Moment Estimation）結合了 Momentum 與 RMSprop 的優點，同時計算一階動量（平均梯度）與二階動量（梯度平方），並進行偏差修正。
- 公式為：

計算一階動量：

$$m_t = \beta_1 \cdot m_{t-1} + (1-\beta_1) \cdot \nabla_\theta L(\theta_t)$$

計算二階動量：

$$v_t = \beta_2 \cdot v_{t-1} + (1-\beta_2) \cdot \left(\nabla_\theta L(\theta_t)\right)^2$$

修正偏差：

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t}$$

更新參數：

$$\theta_{t+1} = \theta_t - \frac{\eta \cdot \hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

$\beta_1$、$\beta_2$：分別控制動量與梯度平方的衰減

$\epsilon$：避免除以零的極小值

■ 適用範圍

- 幾乎所有深度學習模型。
- 資料噪聲大或梯度稀疏的問題。
- 初學者作為「萬用優化器」。

■ 特點

- 收斂快且穩定。

- 自動調整學習率。
- 記憶體需求較高。
- 在部分情境下，泛化能力略輸 SGD。

### （6）正則化係數

- 定義
  - 正則化係數（Regularization Coefficient）是用來控制模型複雜度的超參數。
  - 正則化係數決定了「在損失函數中，正則化項的影響力」，用於避免模型過度擬合（Overfitting）。
  - 作用於正則化公式：

    Loss = 原始損失 + λ × Regularization Term

    λ：正則化係數
  - Regularization Term：正則化項
- 適用範圍
  - 機器學習
    - 線性迴歸、邏輯迴歸。
    - 支持向量機（SVM）。
    - Lasso 迴歸、Ridge 迴歸等模型。
  - 深度學習
    - 各類神經網路（CNN、RNN、Transformer 等）。
    - 避免模型權重過大或過度依賴特定特徵。
  - 作用機制
    - 當 λ 值變大：
      - 強化限制。
      - 權重被壓縮得更小。
      - 模型變簡單，但可能欠擬合。

- 當 λ 值變小：
  - 放鬆限制。
  - 權重可以變大。
  - 模型更靈活，但容易過擬合。

### 3. 正則化技術與模型穩定化

在機器學習與深度學習中，模型若過度追求在訓練資料上的精確度，往往會記住太多資料細節，導致在新資料上表現不佳，這種現象稱為過擬合（Overfitting）。

為了避免過擬合，並讓模型在不同情境下保持穩定表現，實務上會採用各種正則化技術（Regularization），限制模型的複雜度或引導模型學習更具泛化能力的特徵。

以下將介紹幾種常用的正則化與模型穩定化技術，包括 L1、L2 正則化、Elastic Net、Dropout，以及 Early Stopping。

#### （1）L1 正則化

- 定義
  - L1 正則化（Lasso）是在損失函數中加入所有參數（權重）絕對值的總和，限制模型過度依賴特定特徵，並促使部分權重變為零。
  - L1 正則化用絕對值限制，能把不重要的參數縮到 0
  - 公式示意：

    $\text{Loss} = \text{原始損失} + \lambda \sum |\theta_i|$

    原始損失：通常是 MSE、交叉熵等

    $\theta_i$：模型中每個權重

    λ：正則化係數，決定限制強度

    $\sum |\theta_i|$：把所有權重取絕對值後加總
- 特點
  - 可自動執行特徵選擇，讓模型更簡化。

- 常用於高維度資料或特徵數量多的情況。

- 常見問題
  - 若 λ 設定過大，可能過度刪除重要特徵，導致欠擬合。
- L1 正則化 → 為什麼叫 Lasso？
  - Lasso 全名為：Least Absolute Shrinkage and Selection Operator，名稱中的每個詞分別代表：
    - Least Absolute：用絕對值（Absolute Value）作為限制條件（L1 正則化就是絕對值加總）。
    - Shrinkage：讓權重變小。
    - Selection：會把不重要的權重變成 0，等於自動做特徵選擇。

**（2）L2 正則化**

- 定義
  - L2 正則化（Ridge）是在損失函數中加入所有參數（權重）的平方和，降低權重過大，避免模型過度擬合。
  - L2 正則化用平方限制，讓權重縮小，但不直接變 0。
  - 公式示意：

    $\text{Loss} = \text{原始損失} + \lambda \sum \theta_i^2$

    $\theta^2$：計算每個權重的平方，即使是負數也會變為正數。

    λ 越大，權重值被壓抑得越小。
- 特點
  - 能穩定權重大小，防止過擬合。
  - 不會將參數直接壓到零。
- 常見問題
  - 過大的 λ 可能造成模型欠擬合。
- L2 正則化 → 為什麼叫 Ridge？

- 在統計裡，如果資料有高度相關的特徵（多重共線性），做普通最小平方法（OLS）會讓估計變得不穩。
- 加上 L2 正則化後，估計的解會沿著某種「脊線（Ridge）」分佈，而不是像 OLS 那樣無限制變動。
- 幾何上，加入平方和限制後，權重的可能解會落在一個橢圓（或橢球）形的範圍內，形狀看起來就像脊線或山脊。

**（3）Elastic Net**

- 定義
  - 結合 L1 與 L2 的特性，同時對權重施加稀疏與穩定化的限制。
  - 公式示意：

    $\text{Loss} = \text{原始損失} + \lambda_1 \sum |\theta_i| + \lambda_2 \sum \theta_i^2$
- 特點
  - 保留 L1 的特徵選擇能力，又利用 L2 防止模型過度稀疏。
  - 對於多重共線性（Highly Correlated Features）問題特別有效。
- 常見問題
  - 需要同時調整兩個正則化係數，調參較複雜。

**（4）Dropout**

- 定義
  - 在訓練過程中，隨機將部分神經元暫時關閉，減少神經元之間的依賴，讓模型學到多種路徑的特徵，減少過擬合。
  - 公式示意（以隨機關閉為例）：

    output = dropout($x$, $p$)

    $x$：神經元輸出值

    $p$：保留機率，常見如 0.5

- Dropout 訓練階段：
  - 隨機將部分神經元輸出設為 0。
  - 讓模型學習不同「子網路」的結構。
- Dropout 測試階段：
  - 不再丟棄神經元，而是把權重按保留率縮放，保持輸出期望值一致。

- 特點
  - 能降低過擬合風險。
  - 不需要修改網路架構，僅在訓練階段生效。
  - 常用於深度神經網路。
- 常見問題
  - Dropout 機率設定過高，可能導致模型無法有效學習。
  - 測試時需關閉 Dropout 並調整輸出權重。

**（5）Early Stopping**

- 定義
  - 早停法（Early Stopping）在訓練過程中，若驗證集的效能（損失）在連續多次迭代後不再改善，便提前終止訓練。
- 特點
  - 透過「patience」參數，決定可容忍多少次不進步。
  - 防止過擬合，節省訓練時間。
  - 無需修改模型架構，是一種訓練策略。
- 常見問題
  - patience 設定過短，可能提早停止訓練，導致模型欠擬合。
  - patience 設定過長，可能仍導致過擬合。

## 4. 資料增強與重取樣

在機器學習與深度學習中，資料數量及多樣性對模型的表現至關重要。但實務上常面臨資料量不足、類別分佈不均、或資料偏差等挑戰。為提升模型泛化能力並減少過擬合，可透過資料增強（Data Augmentation）與重取樣策略（Resampling Strategies）嘗試解決這些問題。

### （1）資料增強

- 定義
  - 資料增強（Data Augmentation）是指利用各種隨機變換手段，人工擴增訓練資料，製造更多樣本，藉此提升模型對多變環境的適應力，並降低過擬合的風險。
- 常見方法
  - 影像：
    - 旋轉、平移、翻轉、縮放、裁切、添加雜訊、變更亮度等。
  - 文字：
    - 同義字替換、隨機刪詞、分詞打亂、句子順序調整等。
  - 音訊：
    - 改變播放速度、音量變化、加背景噪音、隨機靜音片段等。
- 優點
  - 增加資料多樣性。
  - 降低模型對特定樣式的過度擬合。
  - 幫助模型學習更具泛化能力的特徵。
  - 不需額外收集昂貴的實際數據。
- 常見問題
  - 不當的變換可能改變原始數據的語意或標註，導致模型學習錯誤資訊。
  - 大規模資料增強可能提高計算成本。
  - 需根據應用場景謹慎選擇增強手法。

### （2）重取樣

- 定義
  - 重取樣（Resampling）是透過調整樣本數量或權重，解決資料集中類別不平衡問題，減少模型對多數類別的偏倚，提升少數類別的辨識能力。
- 常見方法
  - 過採樣（Oversampling）
    - 原理
      - 增加少數類別的樣本數，使各類別樣本數趨於平衡。
    - 常見方法
      - 複製少數類別的現有樣本。
      - 使用 SMOTE（Synthetic Minority Over-sampling Technique）等演算法，合成新樣本。
    - 優點
      - 平衡類別分佈。
      - 提升少數類別的預測能力。
      - 不會損失原始資料。
    - 缺點
      - 單純複製樣本易導致過擬合。
      - 合成樣本可能產生不自然的資料點。
  - 欠採樣（Undersampling）
    - 原理
      - 減少多數類別的樣本數，以達到類別平衡。
    - 優點
      - 降低計算成本。
      - 簡單易行，實作快速。
    - 缺點
      - 可能丟失有價值的資料，導致模型準確度下降。
      - 在少數類別本就稀少時，不適用。

- 類別權重調整（Class Weighting）
  - 原理
    - 在模型訓練過程中，針對損失函數賦予不同類別不同的權重，使少數類別對模型的貢獻度提高。
  - 優點
    - 不需更動樣本數，保留原始資料完整性。
    - 適用於資料量有限或無法輕易生成新樣本的情況。
    - 計算資源消耗較低。
  - 缺點
    - 權重需謹慎調整，過大可能造成過度補償。
    - 權重設定過小可能無法解決不平衡問題。

## 5. 模型壓縮與加速技術

隨著 AI 模型規模日益龐大，將其部署至實際應用環境（尤其是行動裝置、IoT 或邊緣運算）時，經常面臨儲存空間不足、運算速度緩慢，以及能源消耗過高等挑戰。模型壓縮（Model Compression）與加速技術（Acceleration Techniques）的主要目標，便是有效減少模型的大小與計算需求，同時盡可能維持原有的預測精準度，確保在有限資源條件下，模型仍能穩定且高效地運作。

### （1）知識蒸餾

- 定義
  - 知識蒸餾（Knowledge Distillation）是一種將大型、高準確度的「教師模型（Teacher）」知識，傳遞給較小、較輕量的「學生模型（Student）」的技術。
  - 目標是讓學生模型在體積更小、運算更快的情況下，仍能接近教師模型的預測表現。

- 應用場景
  - 將模型部署至硬體資源有限的設備，如行動裝置或邊緣運算節點。
  - 降低運算延遲，提升即時應用效率。
- 優點
  - 學生模型體積更小、推論速度更快。
  - 在準確度上接近原本大型模型的表現。
  - 適用於模型壓縮與加速雙重目標。
- 常見問題
  - 蒸餾過程需要額外訓練時間與運算資源。
  - 若教師模型本身存在偏誤，可能也被傳遞給學生模型。

**（2）模型剪枝**

- 定義
  - 模型剪枝（Pruning）是透過移除神經網路中影響較小、貢獻度低的權重或神經元，達到減少模型大小與提升計算效率的目的。
- 類型
  - 結構化剪枝（Structured Pruning）
    - 移除整個神經元、卷積通道或層級。
    - 較容易與硬體加速器整合。
  - 非結構化剪枝（Unstructured Pruning）
    - 移除零散的單一權重參數。
    - 模型稀疏化，但硬體支援較差。
- 優點
  - 減少模型參數數量，降低儲存需求。
  - 提升模型運算效率，降低推論延遲。
  - 可與其他壓縮技術結合使用。

- 常見問題
  - 剪枝過度可能導致模型準確度明顯下降。
  - 部分硬體對非結構化稀疏矩陣支援有限，實際效益有限。

### （3）量化

- 定義
  - 量化（Quantization）是將模型中高精度的參數（如 32-bit 浮點數）轉換為較低精度的格式（如 8-bit 整數），以減少模型大小並加快運算速度。
- 優點
  - 大幅降低模型儲存空間與記憶體需求。
  - 減少運算負擔，特別適合硬體加速器（如 Edge TPU、NPU）。
  - 部分硬體平台已針對低精度運算進行最佳化。
- 常見問題
  - 直接量化可能導致模型準確度下降。
  - 常需搭配量化感知訓練 Quantization Aware Training, QAT），讓模型學習適應量化誤差，避免精度損失。
  - 不同層級的量化對模型影響程度不一，需謹慎設計。

### （4）混合精度訓練

- 定義
  - 混合精度訓練指在模型訓練過程中，同時使用不同數值精度（如 FP16 與 FP32），藉此兼顧運算速度與模型精度。
- 優點
  - 加速模型訓練過程。
  - 減少記憶體佔用，允許更大批次訓練或更深層模型。
  - 現代 GPU（如 NVIDIA Tensor Cores）已提供良好支援。

- 常見問題
  - 需要硬體支援，否則效益有限。
  - 低精度運算可能引發數值不穩定，需特別處理梯度縮放（Gradient Scaling）等問題。
  - 並非所有演算法都能無痛適用。

## 模擬考題

1. 針對缺失值（Missing Value）的處理，下列哪種方法屬於利用模型進行補值的方式？

   （A）刪除法（Deletion）

   （B）均值填補（Mean Imputation）

   （C）預測模型填補（Predictive Imputation）

   （D）缺失指標編碼（Missing Indicator）

2. 在進行異常值（Outlier）偵測時，哪種方法是利用統計特徵分佈來判斷離群點？

   （A）KNN 補值

   （B）四分位距法（IQR）

   （C）Isolation Forest

   （D）One-hot Encoding

3. 下列哪一種特徵選擇方法會直接在模型訓練過程中進行特徵挑選？

   （A）Filter 方法

   （B）Wrapper 方法

   （C）Embedded 方法

   （D）One-hot Encoding

4. 在進行特徵轉換時，若資料存在極端偏高的右偏分佈，常用哪種轉換以降低偏態？

   （A）平方根轉換（Square Root Transform）

   （B）對數轉換（Log Transform）

   （C）Label Encoding

   （D）缺失指標編碼

5. 在模型訓練過程中，若模型在訓練集表現良好，但測試集效能不佳，可能是發生了什麼問題？

   （A）欠擬合

（B）特徵縮放失敗

（C）過擬合

（D）批次大小過小

6. 交叉驗證中，哪一種方法在每次驗證中只留下一筆樣本作為驗證集，其餘皆用於訓練？

（A）K-fold

（B）Stratified K-fold

（C）LOOCV

（D）Repeated K-fold

7. 在訓練深度學習模型時，隨機將部分神經元暫時關閉以減少過擬合的技術是什麼？

（A）Early Stopping

（B）Dropout

（C）Batch Normalization

（D）Quantization

8. 在模型壓縮技術中，知識蒸餾（Knowledge Distillation）的主要目的是什麼？

（A）減少資料維度

（B）降低參數數量使模型更小

（C）將大型模型的知識轉移至小型模型

（D）增強資料平衡

9. 在優化器中，哪一個方法會自動調整每個參數的學習率，特別適用於稀疏資料？

（A）Momentum

（B）Adagrad

（C）Adam

（D）SGD

10. One-hot Encoding 在什麼情況下可能造成模型的運算負擔加重？

（A）缺失值多的資料

（B）資料集中某類別種類 10 種

（C）類別變數具有高基數（High Cardinality）

（D）資料集中無重複樣本

# 考題解析

1. **Ans（C）** 預測模型填補（Predictive Imputation）

   **解析：**預測模型填補是利用其他特徵變數訓練模型，如迴歸或分類器，預測缺失值的位置或值，特別適用於特徵之間關聯性高的情況。相對於簡單的刪除或均值填補，模型補值能更好地捕捉數據內在關聯性，降低資訊流失。

2. **Ans（B）** 四分位距法（IQR）

   **解析：**四分位距法是常見的統計方法，透過計算資料的第一與第三四分位數（Q1 與 Q3），並判定距離超過 IQR（四分位距）1.5 倍以上的數據點為異常。這種方法直觀且適用於大多數數值型資料的初步檢查。

3. **Ans（C）** Embedded 方法

   **解析：**Embedded 方法會在模型訓練過程中同步進行特徵挑選。例如決策樹模型會產生特徵重要性（Feature Importance），或像 Lasso、Ridge 透過正則化直接控制特徵權重，實現同時建模與特徵選擇的目標。

4. **Ans（B）** 對數轉換（Log Transform）

   **解析：**對數轉換可有效降低右偏分佈的極端值影響，使資料分佈更接近常態。常用於收入、銷售額等具極端高值的變數。平方根轉換則較適用於中度偏態資料。

5. **Ans（C）** 過擬合

   **解析：**過擬合發生在模型學習了太多訓練集的細節與雜訊，導致在未見過的測試資料上表現不佳。常見解決方案包括使用正則化、降低模型複雜度或早停策略（Early Stopping）。

6. **Ans（C）** LOOCV

   **解析：**LOOCV（Leave-One-Out Cross-Validation）是 K-fold 的極端形式，當 K 等於樣本數 n 時，每次只留一筆做驗證。此方法能最大化資料利用，但計算成本極高，不適用於大樣本情境。

7. **Ans（B）** Dropout

   **解析**：Dropout 透過隨機將神經元暫時設為零，迫使模型在每次訓練都以不同的「子網路」進行學習，有效降低模型對特定神經元的過度依賴，減少過擬合風險。

8. **Ans（C）** 將大型模型的知識轉移至小型模型

   **解析**：Knowledge Distillation 是利用大型、高性能的教師模型，將其知識以軟標籤（soft labels）的形式傳遞給小型學生模型，讓學生模型在參數較少的情況下，仍維持接近的預測能力。

9. **Ans（B）** Adagrad

   **解析**：Adagrad 能針對每個參數調整不同的學習率，累積梯度平方和後，自動減小常被更新參數的學習步伐。適合處理特徵稀疏的場景，如文字分析，但學習率會不斷衰減至極小值，需留意訓練是否停滯。

10. **Ans（C）** 類別變數具有高基數（High Cardinality）

    **解析**：One-hot Encoding 會為每個不同類別產生獨立欄位，若類別數過多，會導致特徵維度爆炸，增加記憶體使用與模型訓練的負擔，尤其在資料量大時更明顯。

# 第六章　機器學習治理

隨著機器學習技術與 AI 相關服務廣泛應用於各類產業，資料治理議題隨之成為企業部署 AI 系統時的核心關鍵。除技術實作挑戰外，企業更需面對隱私保護、資訊安全與法規遵循的壓力，以及演算法偏見與公平性的社會責任。

機器學習治理的目的在於系統性控管資料與模型風險，確保 AI 應用在合法、安全、公平的原則下運作。本節將從兩個主要面向進行探討：

- **數據隱私、安全與合規**

辨識及管理資料治理相關的風險，並透過合適的技術方法及治理框架，確保 AI 系統的數據安全性、隱私保護與法規遵循。

- **演算法偏見與公平性**

介紹機器學習模型中的偏見來源，公平性評估方法及降低偏見的具體策略，並提供組織層面的公平性治理機制，以確保 AI 應用的公正性與社會責任。

## 重點掃瞄

# 6.1 數據隱私、安全與合規

### 1. 前言與章節導覽

在機器學習系統的建構與部署過程中，資料的合法使用與安全管理是不可或缺的基礎。AI 模型的效能高度依賴於大量且高品質的訓練數據，這些數據往往涉及個人資訊與敏感內容，若處理不當，將導致隱私洩漏、法規違反，甚至企業信譽損害與法律責任。

本節將針對「數據隱私、安全與合規」三大面向進行系統性說明，強調企業在 AI 應用中應具備的風險辨識能力、技術應用能力與治理規範遵循能力。本節目標如下：

- 說明數據隱私風險的來源與評估方法，協助企業預先辨識潛在風險；
- 介紹常用的隱私強化與匿名化技術，並說明其原理與實務應用限制；
- 比較不同國際與本地資料保護法規，建立合規處理數據的基本原則；
- 提出企業內部資料治理的實務建議，包含控管模型與稽核制度等措施。

### 2. 數據隱私風險的辨識與評估

機器學習模型高度依賴資料作為訓練基礎，特別是在開發監督式學習模型時，往往需仰賴大量標註過的個體層級資料，例如顧客行為記錄、醫療檢驗數據、用戶互動紀錄等。這些資料不僅蘊含可觀的模型價值，也潛藏重大的隱私風險。

若缺乏有效的風險識別與管理機制，極可能導致個人資料外洩或濫用。除可能侵害當事人權益外、也可能違反資料保護相關法規，如《一般資料保護規則》（GDPR）、《加州消費者隱私法案》（CCPA）或《個人資料保護法》（PDPA）等。

### （1）常見數據隱私風險分類

數據隱私風險可依其對個體識別的潛在威脅程度，劃分為三種類型：

- 直接識別風險
  - 指資料中含有足以「直接」辨識特定個人的欄位。
  - 例如姓名、身分證號碼、電子郵件、聯絡電話、金融帳號等。
  - 此類資料一旦外洩，往往立即構成隱私侵害，且多數隱私法規（如GDPR、CCPA、PDPA）均將其視為高度敏感的「個人識別資訊（Personally Identifiable Information, PII）」。因此，這類風險在企業治理中，通常被列為最高優先處理層級。
- 間接識別風險（準識別資訊）
  - 此風險源自於資料中雖無單一能辨識個體的欄位，但透過多項資訊交叉比對，仍可能「推導」出個人身份。
  - 例如性別、出生年月、職業、地理位置、消費習慣、瀏覽紀錄等，都屬於典型的準識別資訊（Quasi-identifiers）。
  - 實務顯示，即使資料中未包含如姓名或身分證字號等直接識別欄位，僅憑出生年月、性別及行政區域等資訊，若結合其他公開或商業資料，仍可能推測出個人身分。在台灣，尤其在人口密度較低或特定職業、族群較少的地區，間接識別風險更為顯著，顯示在大數據時代，保護準識別資訊的重要性不可忽視。
- 再識別風險
  - 即便資料已經過去識別化處理，例如移除姓名或以代碼替代敏感欄位，仍可能因外部資料的豐富性與可取得性，而被「重新還原」出個人身分。
  - 在實務中，若有人將不同資料來源進行交叉比對，即使是匿名化的資料集，也可能被重新識別，造成個資外洩風險。尤其在開放數據、資料共享或用於 AI 模型訓練的場景中，這類再識別風險更應受到高度重視。

### （2）隱私風險辨識與風險評估

為有效掌握並降低上述各類數據隱私風險，企業與系統開發者應建構系統化的風險評估流程。以下為實務中常見的四個風險評估項目方法：

#### A. 資料盤點與分類

建立完整的資料清冊或資料地圖（Data Map），是隱私風險管理的首要步驟。這份文件應記錄並清楚標示每一類資料的以下資訊：

- 資料來源：
  - 資料從哪裡收集而來，如用戶填寫表單、感測器紀錄、外部購買資料等。
- 處理流程：
  - 資料如何被處理，包括收集、整理、分析、儲存、傳輸及銷毀等過程。
- 欄位型態與內容：
  - 記載資料包含哪些欄位、每個欄位的意義及資料格式。
- 接觸單位或使用部門：
  - 哪些部門、職務或人員可以存取或使用該筆資料。
- 流通路徑：
  - 資料在組織內部或外部之間的流動情形及交換方式。
- 儲存位置：
  - 資料儲存在何處，例如內部伺服器、雲端平台、第三方服務商等。
- 保留期限：
  - 資料應保存多久，以及在超過保存期限後如何處置（如刪除、匿名化）。

盤點完成後，應進一步對資料進行層級分類，以便有效控管風險：

- 開放層級（Access Level）
  - 公開資料：
    - 對外公開、無涉個資，例如政府發布的統計資料、年報等。
  - 非公開資料：
    - 僅限內部使用，不對外發布，需要特定權限才能存取。

- 敏感度層級（Sensitivity Level）

  對於非公開資料，還需進一步分級管理，例如：

  - 一般資料
    - 不涉及個資或機敏業務資訊，外洩風險較低。
  - 機密資料
    - 涉及商業機密、內部策略、合約等，若外洩可能對營運造成衝擊。
  - 個人資料
    - 包含可直接或間接識別個人的欄位，如姓名、身分證號、聯絡資訊等，需要依個人資料保護法規嚴格管理。
  - 高度敏感個資
    - 涉及健康、財務、族群、宗教、政治傾向等，外洩恐造成當事人重大損害，也常受到法律特別規範。

**B. 隱私影響評估（Privacy Impact Assessment, PIA）**

隱私影響評估是國際間廣泛採用的隱私治理工具，用來系統性分析資料處理活動對個人隱私可能產生的影響，並提出相應的風險緩解策略。PIA 不僅是許多國家隱私法規的要求，也已成為企業降低隱私風險的重要實務做法。

PIA 的執行流程通常包含以下五個步驟：

a. 資料流程盤點

- 徹底釐清資料從收集、處理、儲存、使用到刪除的完整流程，掌握資料流動的每一環節。

b. 風險辨識

- 分析各個流程中，可能對個人隱私造成風險的環節或因素。

c. 影響程度分析

- 評估若風險發生，可能對個人權益或組織造成的法律、營運或信譽衝擊。

d. 策略擬定

- 擬定技術性或管理性的防範措施，以降低風險發生的機率或減輕衝擊程度。

e. 治理責任分工

- 明確界定內部各部門、角色在隱私保護工作中的責任與權限。

**C. 風險矩陣與風險等級：**

將辨識出的風險按發生機率與影響程度進行初步排序，有助於企業優先處理高風險區域，集中資源進行應對。

- 風險矩陣（Risk Matrix）

  風險矩陣是一種風險視覺化工具，常用於將「風險發生的可能性（Likelihood）」與「風險影響程度（Impact）」交叉評估，形成二維矩陣，進行風險分類與優先處理判斷。常見的風險矩陣分為 3x3 或 5x5 格式。

- 風險等級（Risk Level）

  風險等級是對風險嚴重程度的總體評分指標，通常結合以下兩個要素：

  - 發生機率（Probability）：
    - 此風險在示例如出現的可能性
    - 例如：低（Rare）、中（Possible）、高（Likely）
  - 影響程度（Impact）：
    - 此風險一旦發生，對業務或系統的衝擊強度
    - 例如：輕微（Minor）、重大（Major）、災難性（Critical）
  - 風險等級
    - 風險等級 = 發生機率 × 影響程度
    - 不同等級可分為：
      - 低風險（Low Risk）：可接受，可監控
      - 中風險（Medium Risk）：需規劃因應對策
      - 高風險（High Risk）：應優先處理，必要時迴避或延後導入

**D. 再識別模擬與滲透測試（Re-identification Simulation）**

對於計畫開放、共享或應用於 AI 模型平台的資料集，企業應執行再識別模擬與滲透測試，以驗證匿名化或去識別化措施的有效性。常見的測試方法包括：

- 交叉比對測試
  - 嘗試利用公開社群資料、政府開放資料或商業數據，進行比對，以判斷是否能還原匿名化後的個體身分。
- 欄位組合分析
  - 評估多個欄位在特定情境下，是否可能具備足夠的推導能力，進而辨識出個人身分。

若測試結果顯示再識別風險仍然偏高，企業應採取以下對策：

- 強化匿名化或去識別化技術，如加大數據模糊化程度、降低數據精細度。
- 調整資料釋出範圍或限制使用情境，減少外部識別風險。
- 審慎評估是否適合對外公開該筆資料集。

## 3. 隱私保護與匿名化技術實務應用

在機器學習模型訓練與應用過程中，常涉及大量個人資料與敏感資訊。為降低隱私洩漏風險，除了在資料取得階段進行合法合規把關外，技術層面也需導入資料匿名化及隱私強化技術。以下分別從 基礎技術 與 進階技術 兩個層面，介紹常用方法及其應用考量。

**（1）基礎資料匿名化技術：**

資料匿名化（Data Anonymization）旨在移除或模糊能直接或間接識別個人的資訊（Personally Identifiable Information, PII），降低再識別風險，同時保留數據的分析價值。以下為常見基礎技術：

- 遮蔽（Masking）
  - 定義：
    - 以符號或虛構數據替換敏感欄位的部分或全部內容，例如將姓名「王大明」處理為「王〇〇」，或將身分證號「A123456789」顯示為「A12******89」。
  - 應用場景：
    - 報表展示、非正式分析、測試環境數據生成。
  - 優點：
    - 實作簡單、快速，能保留資料格式（如電話號碼長度）。
  - 限制：
    - 僅隱藏部分資訊，若搭配其他資料仍可能被推測還原。
- 雜湊處理（Hashing）
  - 定義：
    - 對身分類欄位（如帳號、Email）進行單向雜湊（如 SHA-256），產生固定長度、不可逆的字串，用於比對而非顯示。
  - 應用場景：
    - 匿名化用戶 ID、跨資料庫比對、資料去重。
  - 優點：
    - 不可逆、支援一致性比對，安全性高。
  - 限制：
    - 若原始資料種類有限（如短 ID），易受彩虹表攻擊；不適合用於數值分析。
- 泛化（Generalization）
  - 定義：
    - 降低資料精度，例如將出生日期「1987-03-12」泛化為「1980 年代」，或將地址「台北市信義區基隆路」簡化為「台北市」。

- 應用場景：
  - 公開數據集、統計分析、降低精細定位風險。
- 優點：
  - 簡單有效，能保留資料的分佈特性。
- 限制：
  - 精度降低可能影響分析準確度（如年齡分群分析）。

- 分桶（Bucketing）或分組
  - 定義：
    - 將連續數值轉換為區間，如將收入「58,000 元」轉為「50K–60K」；或將年齡「32 歲」歸類為「30–39 歲」區間。
  - 應用場景：
    - 統計報表、人口統計分析、降低數值精確度風險。
  - 優點：
    - 保留數據趨勢，減少個體識別風險。
  - 限制：
    - 若分桶設計過細，仍可能造成再識別風險。
- 隨機擾動（Noise Injection）
  - 定義：
    - 為數值資料加入隨機噪聲（如高斯噪聲），例如將薪資「50,000」擾動為「50,123」，使單筆數據難以精準推算。
  - 應用場景：
    - 數值型資料分享、統計分析。
  - 優點：
    - 保留整體統計特性（如平均數、標準差）。
  - 限制：
    - 噪聲需精心設計，幅度過大會影響數據分析，幅度過小則難以達到保護效果。

### （2）進階隱私強化技術

隨著再識別攻擊技術提，以及生成式 AI 模型訓練與應用時可記憶個資的風險增加，進階的隱私強化技術（PETs）已成為敏感領域（如醫療、金融）中的重要手段。以下介紹常用技術：

- K-匿名、L-多樣性、T-接近性
  - K-匿名（K-Anonymity）
    - 確保每筆紀錄至少與其他 K-1 筆紀錄在準識別欄位（如年齡、性別）上相同，降低個體識別風險。
  - L-多樣性（L-Diversity）
    - 在 K-匿名基礎上，進一步要求每個群組內，敏感欄位（如疾病）必須具有至少 L 種不同值，以避免屬性推測。
  - T-接近性（T-Closeness）
    - 要求群組內敏感欄位的分佈與全體資料集相近，防止因分佈偏差而推測個體特徵。
  - 應用場景：
    - 公開數據集、醫療研究、金融風險分析。
  - 優點：
    - 在結構化資料中保護效果佳，實務中易於實施。
  - 限制：
    - 計算複雜度較高，K 值過大可能導致資料精度降低；對非結構化資料（如文字、影像）的適用性有限。
- 聯邦學習（Federated Learning）
  - 定義：
    - 模型在各個客戶端（如使用者裝置或不同機構）本地進行訓練，只將模型參數更新（如梯度）傳回中央伺服器，避免原始資料集中存放或傳輸。

- 應用場景：
  - 醫療聯盟（跨院數據建模）、手機鍵盤輸入預測。
- 優點：
  - 保留資料在本地，降低外洩風險；支援跨機構合作。
- 限制：
  - 通訊成本高，可能面臨參數逆向推導的攻擊風險。

- 同態加密（Homomorphic Encryption）
  - 定義：
    - 允許在加密資料上直接執行運算（如加法、乘法），解密後結果與在明文上運算相同，確保計算過程中資料全程保密。
  - 應用場景：
    - 雲端 AI 模型訓練、金融風控計算、醫療研究中的外包運算。
  - 優點：
    - 即使數據外包處理，也無需解密，提升機密保障；基於密碼學提供強安全保證。
  - 限制：
    - 計算效能較低，尤其是完全同態加密（FHE），需高效能硬體支援。

## 4. 合規實務建議

在 AI 訓練過程中，企業應全面考量合規風險，不僅滿足法規要求，更要兼顧技術實務及倫理責任。以下為企業應重點關注的實務建議：

- 合法來源與告知同意

  在蒐集個人資料之前，必須確認資料來源合法，無論是直接向當事人蒐集，或是透過第三方取得，都需審視其取得過程是否合法合規。若基於當事人同意，該同意必須具備以下要素：
  - 自由性：
    - 不可因服務限制、經濟利益或壓力而被迫同意。

- 明確性與具體性：
  - 應清楚載明蒐集的資料項目、利用目的、範圍、保存期間等，不可使用籠統條款。
- 可撤回性：
  - 當事人應有權隨時撤回同意，企業須說明撤回方式與後續影響。
- 若無法取得同意，應檢視是否符合法規授權的其他合法依據，如履行契約、法定義務或正當利益等。

- 資料最小化與目的限制
  - 蒐集資料時應遵循「必要性原則」，即僅收集實現 AI 訓練或預期功能所需的最低限度資料，避免無關資訊進入系統。
  - 不得將資料用於未經告知或未獲同意的其他目的，即使該用途對企業有商業價值。
  - 對於敏感資料（如健康、族群、宗教信仰等），更須謹慎評估蒐集必要性與比例原則，並尋求替代方式（例如使用泛化後的統計資料）。
- 去識別化或匿名化處理
  - 若資料計畫對外共享、用於模型發布、研究公開或與第三方合作，應優先採取去識別化或匿名化技術，減少個人識別風險。
  - 去識別化應確保無法輕易回推個人身份，並結合再識別風險測試，驗證匿名化效果。
  - 匿名化雖可降低法規適用程度，但不同法規對匿名化的標準認定有所差異，企業仍須保留風險評估紀錄。
  - 在某些情境下，應考慮先對敏感欄位進行泛化、分桶或差分隱私處理，以平衡隱私保護與資料效用。
- 透明度與紀錄保存
  - 建立完整的資料處理紀錄，紀載以下資訊：
    - 資料來源及收集方式
    - 資料蒐集與利用的法律基礎

- 資料處理過程、傳輸及外部共享情況
- 受影響的資料類別與當事人群體
- 所採用的保護技術與風險緩解措施

◆ 保持對外資訊透明，必要時應提供隱私聲明或模型說明文件，讓使用者瞭解其資料如何被用於 AI 訓練。

◆ 定期進行內部稽核與政策檢視，確保所有作業符合最新法規及業界標準。

- 跨境傳輸規範

◆ 若 AI 訓練或服務涉及跨國資料流通，企業需確認是否觸及不同國家或地區的個資傳輸限制。如 GDPR（歐盟通用資料保護規則）規定若將歐盟居民個資移轉至歐盟以外國家，需符合足夠保護措施、標準合約條款或其他合法機制。

◆ 因此在資料使用上，須檢視與規劃以下項目：

- 明確定義跨境資料流動的範圍、用途與國家。
- 評估接收國的隱私保護水準及潛在法律風險。
- 制定跨境傳輸協議或標準條款，並保存紀錄以供監管機關查驗。

## 重點掃瞄

# 6.2 演算法偏見與公平性

### 1. 前言與章節導覽

隨著人工智慧與機器學習在各領域迅速普及，演算法已成為影響人們生活與決策的重要工具。AI 系統並非完全中立，而是高度依賴於輸入的資料與演算法設計，若缺乏謹慎規劃，極易在無意間放大既有的社會偏見或不公平，造成歧視性結果。

AI 偏見（Bias）並非單純的技術瑕疵，而是涉及社會公平、法律合規、倫理責任與企業聲譽的多面向議題。從歷史數據的偏誤、資料代表性的不足，到模型內部的計算邏輯，都可能在不經意間使 AI 系統產生對特定群體不利的決策，影響公平性並引發法律與道德風險。

### 2. 偏見的成因與類型

人工智慧技術在各領域展現潛力，但其核心運作高度依賴資料的質與量。若資料本身存在偏誤，或演算法設計未充分考量公平性，極易在不知不覺間導致模型產生偏見。這些偏見不僅影響模型的預測準確度，更可能導致社會不公、弱勢群體受害，甚至引發法律與商譽風險。

AI 偏見的成因可分為資料層面與模型層面，並會對企業及社會造成深遠影響，以下分別說明：

#### （1）資料代表性與偏誤風險

資料代表性是衡量訓練資料能否忠實且全面反映目標群體特性的核心指標。如果資料在收集、篩選或標註過程中過度集中於特定群體、文化、語言或社經背景，便可能產生系統性偏見，使 AI 模型學習到偏頗的模式、價值觀或敘事方式，進而影響應用結果的公平性與可靠性。

資料偏見大致可分為三大類、包括來源偏誤、內容偏誤、製程偏誤：

- 來源偏誤（Source Bias）

  來源偏誤發生在資料蒐集階段，核心問題在於資料無法均衡涵蓋所有應被代表的群體或情境，導致模型學到的知識或模式無法普遍適用。常見情況與例子如下：

  - 群體分佈不均
    - 醫療模型缺乏某年齡層或性別數據，導致診斷準確度差異。
  - 社經或地理偏重
    - 資料過度集中在都市、高收入群體，忽略偏鄉或中低收入者的需求。
  - 文化與語言侷限
    - 語料僅來自北美，模型無法理解亞洲語境。
  - 來源平台侷限
    - 資料僅來自特定社群平台，無法應對正式場合語言。
- 內容偏誤（Content Bias）

  內容偏誤存在於資料本身的內容或敘事方式，核心問題是資料內含不公平或歧視的觀點，若未修正，會被模型學習並複製到應用中。

  - 定義：
    - 資料本身包含刻板印象或歷史不平等，反映社會偏見。
  - 例子：
    - 資料將「醫師」預設為男性，「護士」預設為女性，強化性別刻板印象。
    - 信用審核歷史資料中，隱含種族或性別歧視。
- 製程偏誤（Process Bias）

  製程偏誤出現在資料標註或編輯過程，核心問題是人為主觀判斷導致不一致或偏差，進而影響模型的準確度與公平性。

  - 定義：
    - 資料標註或處理過程中，因標註者的主觀或文化差異造成不一致。

- 例子：
  - 不同標註者對同一句話在情感分析上的判斷不同。
  - 標註者因個人觀念產生刻板印象的標註偏誤。

**（2）模型偏見與歧視**

即使資料本身具有良好的代表性，AI 模型在訓練或運算過程中，仍可能因演算法設計、目標設定或學習邏輯，額外引入偏見，或放大原本隱藏於資料中的微弱偏誤。這類偏見屬於模型偏見，其特徵是：即使輸入資料相對平衡，模型本身仍可能產生不公平的預測結果。

常見的模型偏見來源包括：

- 演算法偏見（Algorithmic Bias）
  - 某些演算法在追求整體預測效能時，可能忽略少數群體的需求。例如，推薦系統通常依據多數使用者的偏好生成結果，導致少數群體的興趣與需求被邊緣化。
- 目標函數偏誤（Objective Function Bias）
  - 多數模型在訓練時，以整體平均精度作為優化目標，若未加入公平性約束，就可能在提升整體效能的同時，犧牲特定群體的預測準確度。
- 正規化與簡化偏誤（Regularization Bias）
  - 為避免過度擬合，模型往往會簡化變數間的關聯，可能降低對少數群體特徵的敏感度，導致模型在該群體上的表現較差。
- 對抗式訓練不足（Insufficient Adversarial Training）
  - 即使導入對抗式公平學習（Adversarial Fairness），若設計不完善或參數設定不當，仍可能保留部分偏見，使模型對特定群體的預測不公。

**（3）偏見的潛在影響**

無論是資料偏見還是模型偏見，若企業未能即時辨識並修正，都可能帶來多重風險與負面影響，涉及技術層面、法律責任、商譽及社會信任。常見的潛在影

響包括：

- 弱勢群體受歧視
  - AI 模型可能對特定群體（如女性、少數族群、高齡者、身心障礙者）產生不公平或不利的決策，導致這些群體在就業、金融服務、醫療診斷等領域遭受差別待遇，進一步加劇社會不平等。
- 企業品牌與信譽受損
  - 若模型偏見被揭露，容易引發社會輿論反彈與媒體關注，導致用戶抵制、合作夥伴疏遠，嚴重損害企業形象、信譽及市場競爭力。
- 法律與監管風險
  - 偏見結果若涉及歧視，企業可能違反反歧視法、個人資料保護法或消費者保護法，面臨巨額罰款、法律訴訟，或受到監管機關調查與處分。

## 3. 公平性指標與評估工具

在 AI 系統的開發與部署過程中，公平性已成為全球監管與企業治理的重要議題。「公平」並不是單一概念，而是需要明確的衡量標準與工具，才能有效檢驗模型在不同群體間是否存在不公平的差異。選擇適用的公平性指標，通常取決於應用場景、業務目標及法律法規的要求。

以下介紹常用的公平性衡量指標與實務上常見的公平性評估工具。

### （1）常見公平性指標

公平性指標主要用來衡量不同群體之間，在 AI 模型預測結果上的系統性差異。以下是國際間常見且廣泛使用的指標：

- Demographic Parity（群體平等率）
  - 定義：
    - 不同群體獲得正向預測（例如核准貸款、錄取面試等）的比例應大致相同。
    - 強調結果的均等分配（Equality of Outcome）。

- 比較對象：
  - 各群體獲得正向預測的整體比例（不論實際是否符合條件）。
- 適用情境：
  - 對結果平等有高度要求的場景，如招聘、入學機會等。
- 限制：
  - 可能為達到比例公平而犧牲個別個案的預測準確性。
  - 無法考量實際資格差異，只追求比例相等。

- Equal Opportunity（機會平等）
  - 定義：
    - 在實際應獲得正向預測的個案中（如真正應核准貸款的人），不同群體被正確預測的機率應相同。
    - 強調真正有資格的人不能漏掉。
  - 比較對象：
    - 各群體中「真實應該被選擇」者的正確預測比例。
  - 適用情境：
    - 必須確保真正該被服務者不被忽略的任務，例如醫療診斷、社會福利核准等。
  - 限制：
    - 僅關注正例的正確預測，未考量負例的錯誤比例。
    - 無法全面涵蓋所有可能的不公平情況。
- Equalized Odds（均衡機率）
  - 定義：
    - 要求模型在不同群體間，對「正例」與「負例」都有相同的預測機率，即 True Positive Rate（正確核准比例）與 False Positive Rate（錯誤核准比例）皆需一致。
    - 強調所有群體在各種情境下都公平。

- 比較對象：
  - 各群體的正確預測率和錯誤預測率兩種情境。
- 適用情境：
  - 同時考量預測正確率與誤判率公平的場景，例如司法判決、信用評估等敏感領域。
- 限制：
  - 實務上難以完全達成，且可能需要犧牲部分整體效能作為妥協。
  - 實現方式較為複雜。

- Disparate Impact（不利影響比）
  - 定義：
    - 比較群體間獲得正向結果的比例，若某群體的比例未達另一群體的80%（80% Rule），可能構成間接歧視或不利待遇。
  - 適用場景：
    - 主要用於法律合規審查，例如招聘、公平貸款等領域的歧視檢驗。
  - 限制：
    - 僅考量結果比例差異，無法指出產生偏差的具體原因或細微偏見來源。

**（2）公平性評估工具**

隨著 AI 公平性議題備受關注，許多開源工具與商業解決方案已問世，協助開發者評估並修正模型偏見。以下介紹兩個常用工具：

- IBM AI Fairness 360（AIF360）

  由 IBM Research 開發的開源 Python 工具包，支援超過 70 種公平性指標及多種去偏技術。
  - 功能：
    - 計算群體間各類公平性指標。
    - 提供資料前處理、模型內部處理及結果後處理的去偏方法。

- 產生公平性分析報告及視覺化圖表。

◆ 優點：

- 支援多種指標與方法。
- 文件完整，適合研究及企業試驗性應用。

◆ 限制：

- 對大型商業模型或複雜系統需進行額外整合與測試。

• Microsoft Fairlearn

由微軟開發的 Python 工具，專注於衡量及降低 AI 系統中的公平性問題。

◆ 功能：

- 計算群體間公平性指標。
- 提供公平性約束下的模型再訓練工具。
- 支援可解釋性分析。

◆ 優點：

- 與 scikit-learn 等 Python 生態系統高度相容。
- 易於整合進現有機器學習流程。

◆ 限制：

- 提供的去偏功能較 AIF360 少，適合中小型或輕量化專案。

## 4. 降低演算法偏見的方法論與技術方案

當識別 AI 系統中的偏見之後，下一步便是採取技術方法來減輕或消除這些偏見。在實務上，降低演算法偏見的方法大致可分為三個階段：

### （1）資料前處理

資料前處理指的是在將資料投入模型訓練前，即針對資料本身進行調整或修正，以降低資料中隱含的偏見。常見的方法包括：

- 資料重新抽樣（Re-sampling）
  - 定義：
    - 透過增加或減少特定群體資料的方式，使資料分佈更加均衡。
  - 實例：
    - 招聘模型訓練前，調整不同性別的履歷數量達成平衡。
- 特徵去偏處理（Feature Neutralization）
  - 定義：
    - 移除或調整可能引發偏見的敏感特徵（如性別、族群），或其高度相關的特徵。
  - 實例：
    - 銀行信用評分移除客戶族群特徵，以避免種族偏見。
- 資料匿名化與泛化（Data Anonymization and Generalization）
  - 定義：
    - 降低敏感特徵的精確性（如年齡轉換成年齡區間），減少因敏感特徵產生的偏見。
  - 實例：
    - 將特定族群改為一般化標籤，降低偏見產生。

**（2）模型內部處理**

模型內部處理是指在模型訓練過程中，即將公平性指標或約束條件加入到模型演算法中，以達成更公平的預測結果。常見的方法包括：

- 公平性約束訓練（Fairness Constraints）
  - 定義：
    - 在訓練模型時，同時加入公平性指標作為約束條件，例如 Demographic Parity、Equal Opportunity 等。
  - 實例：
    - 貸款模型在訓練時強制約束不同性別的核准率相似。

- 對抗式去偏模型（Adversarial Fairness）
  - 定義：
    - 同時訓練一個去預測敏感特徵（如性別）的「對抗式網路」，迫使主模型學到不受敏感特徵影響的特徵表現。
  - 實例：
    - 招聘模型透過對抗網路訓練，使履歷分析結果無法推斷申請者性別。
- 公平性正規化（Fairness Regularization）
  - 定義：
    - 在模型訓練目標函數中加入額外的公平性損失項（Fairness Loss），同時平衡準確度與公平性。
  - 實例：
    - 信用評分模型透過加入公平性損失函數，確保各族群錯誤率接近。

**（3）模型後處理**

模型後處理指的是模型訓練完成後，針對模型輸出的結果進行修正，以達到更公平的結果。常見的方法包括：

- 結果門檻調整（Threshold Adjustment）
  - 定義：
    - 模型產出預測分數後，針對不同群體調整正向預測的閾值，以達到公平性。
    - 直接改變判斷標準（閾值），例如改變「通過」或「不通過」的門檻。
  - 實例：
    - 司法判決風險評估模型，調整不同族群的判斷閾值，使不同族群錯誤率公平。
    - 將某群體的貸款評估「核准門檻」，從 60 分降到 55 分。

- 結果校準（Calibration）
  - 定義：
    - 針對不同群體的模型預測分數進行重新校準，使得相同分數在不同群體間代表相同的意義。
    - 不改變判斷標準（閾值），但調整不同群體的預測分數，使同一分數在不同群體具有相同意義。
  - 實例：
    - 信用評分模型在男女族群分別進行校準，避免同樣分數下不同群體的待遇差異。
    - 將某群體的貸款評估「分數」，從 60 分重新校準成 65 分。

## 5. 組織面向的 AI 公平性治理策略

確保 AI 模型在實際運作中維持公平性，不能僅依賴技術層面的解決方案。企業必須從治理與管理層面著手，建立完善的制度、流程與文化，將公平性融入整體營運與決策體系。

本小節將從企業組織的角度，提出系統化的公平治理策略與實務做法，協助企業在符合法律規範的同時，落實社會責任，並確保 AI 系統能在各層面達成更高的公平標準。

### （1）建立 AI 公平性治理機制

企業需建立清晰的 AI 公平性治理架構，確保 AI 系統在設計、開發與部署各階段，均符合法規與社會期待。具體作法包括：

- 建立公平性政策與標準
  - 制定明確的 AI 公平性原則，界定在模型開發與應用過程中應遵守的公平性目標（如 Demographic Parity、Equal Opportunity 等）。
  - 訂定具體的公平性衡量指標及可接受範圍，作為各專案的遵循標準。

- 設立跨部門公平性審查委員會
  - 集合法務、技術、產品、倫理及風險管理等部門，負責定期審查 AI 系統的公平性風險及績效。
  - 於 AI 專案早期即參與，預防潛在偏見，而非僅在事後處理。
- 明確責任與問責機制
  - 將 AI 公平性納入高階主管（如資訊長、法務長）的關鍵績效指標（KPI），或企業 ESG 報告範疇。
  - 建立公平性審查流程的紀錄機制，確保可追溯與透明，利於內外部稽核。

**（2）多元化團隊與公平意識教育訓練**

多元化團隊及公平意識，是降低演算法偏見的重要基礎。企業需確保開發團隊與決策團隊具備不同的視角與背景，減少潛在偏見的發生。

具體作法包括：

- 招募多元化團隊成員
  - 積極聘用不同性別、族群、文化、專業背景的人才，降低團隊盲點風險。
  - 鼓勵跨部門合作與交流，提升討論的多元性與敏銳度。
- 推動公平性與倫理培訓
  - 定期舉辦 AI 公平性及倫理課程，讓員工瞭解偏見的成因、影響及風險。
  - 建立企業內部公平性文化，提升員工對公平議題的重視與敏感度。

**（3）公平性評估與稽核制度**

企業需建立制度化、持續性的公平性評估與稽核機制，確保 AI 系統在運行過程中能即時發現問題並進行調整。

具體作法包括：

- 定期公平性稽核（Fairness Audits）
  - 每年定期對關鍵 AI 系統進行公平性稽核，並產出公開報告與改善計畫。
  - 對於新上線或重大更新的 AI 系統，應進行公平性審查與測試。
- 導入公平性指標監控系統
  - 將公平性指標納入 AI 系統日常監控報告。
  - 建立即時警示機制，當指標超出預設範圍時，能即時啟動風險應變措施。

**（4）AI 公平性資訊揭露與溝通策略**

透明度是建立外界信任的關鍵。企業應透過主動公開資訊與溝通，展現對 AI 公平性的重視與承諾。

具體作法包括：

- 主動揭露 AI 公平性報告
  - 定期公開 AI 系統的公平性衡量結果、績效分析與改善措施，展現企業在公平性議題上的責任感。
  - 內容包括：主要公平性指標數據、偏見發現情形、已採取或預計採取的修正措施。
- 引入第三方獨立驗證
  - 邀請外部專業機構進行公平性稽核或審查，並將驗證結果對外公布，以提高外部信任度與公信力。
- 與利益關係人保持溝通
  - 積極聆聽不同群體、用戶或社會大眾的關注與建議，將其納入 AI 系統的持續改進中。

## 模擬考題

1. 在個人資料保護法規的框架下，下列哪一組資訊單獨使用時，最明確地被歸類為「直接識別個人身份資訊（Directly Identifiable Information, PII）」？

   （A）出生日期、郵遞區號、職業

   （B）網站 Cookie ID、設備 IP 位址、地理定位資訊

   （C）姓名、電子郵件地址、身分證字號

   （D）網路行為模式、匿名化統計數據、加密後的密碼雜湊值

2. 以下哪一種資料處理方式屬於「隨機擾動（Noise Injection）」？

   （A）把姓名改成代號

   （B）將薪資加入隨機誤差

   （C）將地址改成縣市級別

   （D）以星號遮蔽身分證號

3. 在 AI 偏見治理中，對不同群體調整模型預測閾值，屬於哪一種偏見修正方法？

   （A）資料泛化

   （B）結果門檻調整

   （C）模型剪枝

   （D）資料增強

4. 若企業想確保 AI 模型在不同群體間「真正該被選擇者」皆有同等機會被正確預測，應採用哪一個公平性指標？

   （A）Equal Opportunity

   （B）Demographic Parity

   （C）K-Anonymity

   （D）T-Closeness

5. 若企業欲在 AI 專案中降低敏感特徵（如性別）的影響，可以採取哪種模型內部處理方式？

   （A）對抗式去偏模型

（B）模型蒸餾

（C）雜湊處理

（D）結果門檻調整

6. 若資料集中存在群體樣本數不均，造成模型偏向多數群體，適合採用哪種技術？

（A）權重初始化

（B）權重衰減

（C）類別重加權

（D）全量訓練

7. T-接近性（T-Closeness）技術主要是為了降低什麼風險？

（A）機率分佈差異造成的識別風險

（B）資料重複風險

（C）訓練時間過長

（D）演算法震盪問題

8. 在 AI 公平性治理中，企業應建立什麼跨部門組織，確保公平性納入日常決策流程？

（A）模型剪枝委員會

（B）AI 蒸餾中心

（C）公平性審查委員會

（D）隨機抽樣團隊

9. 若企業欲降低 AI 模型因少數群體樣本過少而造成的過擬合風險，可以採用哪種資料增強方法？

（A）SMOTE

（B）Softmax

（C）Momentum

（D）Dropout

10. 下列哪一項不是常見的隱私保護基礎技術？

（A）遮蔽（Masking）

（B）泛化（Generalization）

（C）知識蒸餾（Knowledge Distillation）

（D）分桶（Bucketing）

# 考題解析

1. **Ans（C）** 姓名、電子郵件地址、身分證字號

   **解析**：「直接識別個人身份資訊（PII）」是指無需透過其他資訊輔助，單獨即可明確辨識出特定個人的資料。

   （A）出生日期、郵遞區號和職業等資訊單獨無法直接識別特定個人，通常需要結合其他資訊才能達到識別效果，屬於準識別資訊（Quasi-Identifiers）。

   （B）網站 Cookie ID、IP 位址和地理定位資訊雖然屬於識別符號，且在特定情境下可被追溯或結合其他資訊後具有識別性，但其本身通常不被視為單獨可直接指向特定個人的 PII。通常被歸類為間接識別資訊。

   （C）姓名、電子郵件地址和身分證字號是典型的直接識別資訊，這些資訊單獨就能指向一個特定且唯一的人。

   （D）網路行為模式、匿名化統計數據和加密後的密碼雜湊值均屬於非直接識別資訊。匿名化數據已去除識別性，加密雜湊值雖然源於個人密碼但已不可逆地處理，網路行為模式本身也不直接指向個人。

2. **Ans（B）** 強化匿名化技術

   **解析**：即便進行匿名化，外部資料比對仍可能造成再識別風險，因此須強化匿名化措施或限制資料使用情境，才能降低風險。

3. **Ans（B）** 結果門檻調整

   **解析**：結果門檻調整（Threshold Adjustment）屬於模型後處理方式，透過改變不同群體的決策門檻，以平衡模型在不同群體間的預測公平性。

4. **Ans（A）** Equal Opportunity

   **解析**：Equal Opportunity 著重在實際應獲得正向結果的個案，在不同群體間應具有相同的被正確預測機率，尤其適用於醫療、社會福利等情境。

5. **Ans（A）** 對抗式去偏模型（Adversarial Fairness）

   **解析**：對抗式去偏透過訓練一個對抗網路，迫使主模型無法預測敏感屬性，從而降低模型對敏感特徵的依賴，是常見的公平性處理技術。

6. **Ans（C）** 類別重加權（Class Weighting）

   **解析**：類別重加權能在模型訓練時賦予少數類別更高權重，提升模型對少數群體的辨識能力，是處理不平衡資料常見的方法。

7. **Ans（A）** 機率分佈差異造成的識別風險

   **解析**：T-Closeness 要求群組內敏感屬性的分佈，需與全體資料集分佈相近，以防止透過分佈偏差推測個人特徵，是進階的隱私保護技術。

8. **Ans（C）** 公平性審查委員會

   **解析**：公平性審查委員會匯集法務、技術、產品與倫理等部門，負責評估 AI 系統的偏見風險，是企業治理 AI 公平性的重要機制。

9. **Ans（A）** SMOTE

   **解析**：SMOTE（Synthetic Minority Over-sampling Technique）透過合成新樣本，擴增少數類別資料，有助於平衡類別分佈，降低過擬合及模型偏誤風險。

10. **Ans（C）** 知識蒸餾（Knowledge Distillation）

    **解析**：知識蒸餾是用於模型壓縮與加速的技術，並非基礎隱私保護技術。隱私保護常用技術包括遮蔽、泛化及分桶，用來降低個資識別風險。

主辦單位 經濟部產業發展署 Industrial Development Administration, MOEA

執行單位 工業技術研究院 Industrial Technology Research Institute