Week_1：

關於「如何選擇損失函數」的問題：

這**不是一個 Open Problem**。選擇損失函數必須依據**任務類型**和**數據特性**。

- **迴歸問題 (Regression)：**
  - **MSE (均方誤差)** 是最標準的選擇，它對大錯誤的懲罰較重。
  - **參考資料：** Google 提供了 L1 (MAE) 和 L2 (MSE) 損失的簡潔對比。
  - **網址：** https://developers.google.com/machine-learning/crash-course/linear-regression/loss
- **分類問題 (Classification)：**
  - **Cross-Entropy (交叉熵)** 是標準選擇。它衡量模型預測的「機率」與「真實標籤」之間的差距。
  - **參考資料：** 這篇指南清楚解釋了交叉熵的原理。
  - **網址：** https://www.v7labs.com/blog/cross-entropy-loss-guide
- **不平衡數據 (Unbalanced Data)：**
  - **Weighted Loss (加權損失)** 是正確的處理方式。
  - 最常見的是 **Weighted Cross-Entropy (加權交叉熵)**，它會給予「少數類別」的錯誤更高的懲罰，強迫模型學習。
  - **Focal Loss** 是一種更進階的加權損失，它專注於學習「難分類」的樣本，對於嚴重不平衡的數據（如目標檢測）非常有效。
  - **參考資料 (Focal Loss 經典論文)：** Lin, T. Y., et al. (2017). *Focal Loss for Dense Object Detection*.
  - **網址：** https://openaccess.thecvf.com/content_ICCV_2017/papers/Lin_Focal_Loss_for_ICCV_2017_paper.pdf

Week_2：

迴歸中，若誤差**不**服從常態分佈，而是**偏態 (skewed)** 或**重尾 (heavy-tailed)** 分佈時，OLS 估計式（$\hat{\beta}$）的漸近性質（一致性、漸近常態性）會受何影響？是否有替代的穩健迴歸方法？

這**不是一個 Open Problem**。這是「穩健統計 (Robust Statistics)」領域的經典問題。

- **核心解法/分析：**
  1. **一致性 (Consistency)： 仍保持**。一致性不依賴常態假設，只依賴 $E[\epsilon|X] = 0$。
  2. **漸近常態性 (Asymptotic Normality)： 通常仍保持**（根據中央極限定理），**前提是**誤差的變異數 (variance) 必須是有限的。如果分佈重尾到變異數無限大（如柯西分佈），此性質會失效。

    iii. **主要問題（效率 Efficiency）：** OLS 不再有效率。OLS (L2 Loss) 會被重尾分佈產生的「離群值 (Outliers)」嚴重影響，導致估計結果被拉歪，變異數很大。
    iv. **替代方法 (穩健迴歸)：** 是的。核心是使用對離群值懲罰較輕的損失函數，例如：
        - **LAD (L1 迴歸 / 中位數迴歸)：** 最小化絕對誤差和，對離群值不敏感。
        - **Huber Loss (胡伯損失)：** 混合型損失。對小誤差用 L2（平方），對大誤差（離群值）用 L1（線性），兼顧效率與穩健性。

- **參考文獻與連結：**
    - **Scikit-learn (Robust Regression)：**
      展示 OLS 如何被離群值影響，以及 Huber 迴歸等方法如何抵抗。
      網址： https://scikit-learn.org/stable/modules/linear_model.html#robust-regression
    - **Huber Loss (Wikipedia)：**
      解釋 M-estimators 和 Huber 損失函數的定義。
      網址： https://en.wikipedia.org/wiki/Huber_loss

Week3：

在「學習-排序」(Learning to Rank) 任務中，如果我們**只知道雜訊的強度排序**（例如 A 區域雜訊 > B 區域），但不知道**精確的雜訊率**，模型是否仍能保持接近乾淨標籤的排序效能 (AUC)？如果不行，是否存在不可避免的排序偏差 (ranking bias)？

這**不是 Open Problem，但這是一個非常前沿且困難的研究領域**。

這個問題屬於「**依賴實例的標籤雜訊 (Instance-Dependent Label Noise, IDN)**」下的「排序學習」範疇。

- **核心解法/分析：**
    i. **一般 IDN 會破壞排序：** 在標準的 IDN 假設下（我們對雜訊一無所知），雜訊會引入一個**不可避免的偏差 (bias)**。這會導致模型學到的「含噪 posterior 機率」$P(\tilde{y}=1|x)$ 的排序，**不等於**「乾淨 posterior 機率」$P(y=1|x)$ 的排序。因此，**AUC 會下降**。
    ii. **問題 (已知排序)：** 答案是**模型可以保持排序效能**。
    iii. **關鍵假設：** 問題敘述中的「知道雜訊強度的排序」，在學術上通常被建模為「**單調性假設 (Monotonicity Assumption)**」。例如，假設雜訊率 $\eta(x)$ 是真實機率 $P(y=1|x)$ 的一個（未知的）**單調函數**（例如：越難分類的樣本 $P(y=1|x) \approx 0.5$，其雜訊率 $\eta(x)$ 越高）。
    iv. **解法 (Monotonic Denoising)：** 在上述單調性假設下，有研究證明，即使 $\eta(x)$ 的精確值未知，模型學到的「含噪機率」的排序，將**等同於**「乾淨機率」的排序（即兩者共享相同的單調變換）。因此，**真實的排序 (AUC) 可以被完美地恢復**。
- **參考文獻與連結：**
    - **Wang, Y., et al. (2023). *Learning with Instance-Dependent Label Noise: A Monotonic Denoising Perspective*. (ICML 2023).**

        - 這篇論文直接回答了問題。它證明了只要雜訊率 $\eta(x)$ 和真實機率 $P(y=1|x)$ 之間存在單調關係（即題目敘述中的「知道排序」），模型就可以恢復出真實的排序，使得 AUC 不受影響。
        - **網址：** https://proceedings.mlr.press/v202/wang23j.html

Week4：

Q1:

Generative model (生成模型, 學習 $p(x,y)$) 和 Discriminative model (判別模型, 學習 $p(y|x)$) 的應用差異為何？為什麼不全部都用判別模型來做分類就好？
這**不是 Open Problem**。這是機器學習中關於模型選擇的基礎概念。

- **核心解法/分析：**
    i. **判別模型 (Discriminative, $p(y|x)$)：**
        - **目標：** 直接學習「決策邊界」。它只問：「給定 x，它是 y 類別的機率是多少？」
        - **應用：** 專注於**分類**任務。例如 Logistic Regression, SVM, 標準神經網路。
        - **優點：** 通常在「純分類」任務上更準確、更高效。
    ii. **生成模型 (Generative, $p(x,y)$)：**
        - **目標：** 學習數據的「完整分佈」。它學習「y 類別的數據 x 長什麼樣子？」
        - **應用：**
            - **數據生成 (Generation)：** 這是最重要的應用。因為模型知道 $p(x|y)$，所以它可以**生成全新的數據**（例如：生成圖片 (GANs, Diffusion Models)、生成文章 (GPT)）。
            - **異常檢測 (Outlier Detection)：** 它可以判斷一個 x 是否「看起來很奇怪」（即 $p(x)$ 很低），這是判別模型做不到的。
            - **處理缺失值：** 由於了解 x 的完整分佈，在 x 的某些特徵缺失時，它仍能做出合理的推斷。
- **參考文獻與連結：**
    - **主題：** 判別模型 vs 生成模型的經典對比。
    - **文獻/資料：** Ng, A. Y., & Jordan, M. I. (2002). *On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes*. (NIPS 2001).
    - **簡介：** 這篇經典論文比較了 GDA（生成）和 Logistic Regression（判別）。它指出，判別模型在漸近上（數據量多時）通常表現更好，但生成模型能更快地收斂（數據量少時）。
    - **網址：** https://ai.stanford.edu/~ang/papers/nips01-discriminativegenerative.pdf

Q2:
GDA (高斯判別分析) 假設 $p(x|y)$ 服從高斯分佈。如果真實資料並非如此，訓練效果會變差很多嗎？

這**不是 Open Problem**。這關乎「模型錯誤設定 (Model Misspecification)」的後果。

- **核心解法/分析：**
  是的，效果很可能會顯著變差。
  i. **GDA 的本質：** GDA 依賴高斯假設來找到**最佳**的決策邊界。
     - 如果所有類別的共變異數矩陣 Σ 相同 (LDA)，GDA 會找到一條**線性**邊界。
     - 如果 Σ 不同 (QDA)，GDA 會找到一條**二次** (quadratic) 邊界。
  ii. **當假設不成立：**
     - 如果真實資料的分佈**高度非高斯**（例如：多峰分佈 (multi-modal)，像甜甜圈的形狀，或非常偏態），GDA 所找到的「最佳」線性或二次邊界，對於真實分佈來說將會是「次佳」甚至「錯誤」的。
     - **範例：** 如果 A 類資料是一個環狀，B 類資料在環的中心，GDA（無論 LDA 或 QDA）都無法正確分開它們，因為它只能畫出直線或二次曲線（如橢圓）。
  iii. **結論：** GDA 對其分佈假設**相對敏感**。在這種情況下，非參數模型（如 k-NN）或更靈活的判別模型（如神經網路、帶核的 SVM）通常會表現得更好，因為它們不對數據的潛在分佈做太強的假設。
- **參考文獻與連結：**
  - **主題：** GDA (LDA/QDA) 的假設與限制。
  - **文獻/資料：** Scikit-learn (Python 庫) 官方文檔中關於 LDA 和 QDA 的比較。
  - **簡介：** 文檔中的圖片清楚地展示了：當數據符合高斯假設時，LDA 和 QDA 表現良好；但當數據不符合時（例如 QDA 去擬合非二次曲線的數據），效果就會很差。
  - **網址：** https://scikit-learn.org/stable/modules/lda_qda.html

Week5：

Q1:
Softmax 具有「對稱性」（或稱「非唯一性」），即多組不同的參數 W（例如將所有 W_k 加上一個常數向量 c）會產生完全相同的機率輸出 $p(y|x)$。這如何影響模型的「可辨識性」 (Identifiability) 和「正則化」 (Regularization) 設計？
這**不是 Open Problem**。這是 Softmax (多類別 Logistic 迴歸) 模型一個廣為人知且已被充分理解的特性。

- **核心解法/分析：**
  i. **可辨識性：**
     - 這個特性意味著模型的參數**不具備唯一可辨識性 (Not Uniquely Identifiable)**。
     - 在訓練（例如最小化 Cross-Entropy）時，存在無限多組參數 W 都可以達到「全域最小值」(Global Minimum)。

ii. **對訓練的影響：**
   - 雖然這不影響找到「最佳解」（因為 Softmax 的損失函數是凸的 (convex)，任何局部最小都是全域最小），但如果**沒有**正則化，最佳參數 W 的具體數值是不確定的，優化器（如 SGD）可能會使權重 W 漂移到非常大（趨近無限大），導致數值不穩定。

iii. **正則化設計：**
   - **L2 正則化 (Weight Decay)** 是**必要**的。
   - L2 正則化（即在 Loss 中加入一項 $\lambda||W||^2$）會**打破這種對稱性**。
   - 在所有「能最小化 Cross-Entropy」的無限多組 W 中，L2 懲罰項會迫使優化器選擇**唯一**的那一組同時具有「最小範數 (minimum norm)」的 W（即權重值最小、最接近 0 的那一組）。
   - **結論：** L2 正則化不僅能防止過擬合，更是確保 Softmax 解的**唯一性**和**數值穩定性**的關鍵。

- **參考文獻與連結：**
  - **Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.**
    - **簡介：** 在第 6.2.2.3 節 (Softmax Units) 中，本書討論了 Softmax 的「過參數化」(overparameterized) 問題（即您所說的對稱性），並說明了 L2 正則化 (Weight Decay) 如何被用來找到一個唯一的、穩定的解。
    - **網址：** https://www.deeplearningbook.org/contents/linear_algebra.html (此書有相關章節)
  - **Stanford CS231n Course Notes on Softmax Classifier:**
    - **簡介：** 史丹佛的課程講義也經常提到這種「模糊性」(ambiguity)，並強調 L2 正則化在 Softmax 中的作用是「移除這種模糊性」。
    - **網址：** https://cs231n.github.io/linear-classify/#softmax

Q2:

Cross-Entropy (CE) Loss 對於「標註噪聲 (Label Noise)」 (即標籤打錯) 的敏感度如何？什麼情況下需要改用其他損失函數或加權？

這**不是 Open Problem**。這是「帶噪學習 (Learning with Noisy Labels)」領域的核心研究問題。

- **核心解法/分析：**

i. **敏感度：** Cross-Entropy 對標註噪聲「非常敏感」。

ii. **原因：** CE Loss 的目標是讓模型對「給定的標籤」輸出**接近 1.0 的機率**。如果這個標籤是錯的（是噪聲），CE Loss 會強迫模型「過度自信地」去擬合這個錯誤的答案，導致模型嚴重偏離乾淨數據應有的分佈，最終學到錯誤的特徵。

iii. **何時需要改變：** 只要懷疑數據集中存在**不可忽視**的標註錯誤時，就應考慮替代方案。

iv. **替代方法：**
   - **加權損失 (Weighted Loss)：** 如果我們能估計哪些樣本「可能」是噪聲（例如：模型對該樣本的 Loss 很高，或者模型預測的類別與標籤不一致），我們可以動態地給予這些樣本**較低的權重**。
   - **噪聲穩健 (Noise-Robust) 的損失函數：**

        - **MAE (L1 Loss)：** 理論上已被證明（在某些噪聲條件下，如對稱噪聲）是穩健的。因為 MAE 不會強迫模型輸出 1.0，它只最小化機率與 (0/1) 標籤的絕對差距，因此對錯誤標籤的懲罰是有界的、線性的，不會像 CE 那樣被錯誤標籤「拉走」。
        - **Generalized Cross-Entropy (GCE)：** 一種推廣的損失函數，它結合了 CE 和 MAE 的特性，旨在提供對噪聲的穩健性。
- **參考文獻與連結：**
    - **Zhang, Z., & Sabuncu, M. (2018).** ***Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels*****. (NIPS 2018).**
        - **簡介：** 這篇論文**直接分析了 CE 對噪聲敏感而 MAE 穩健**的現象，並提出了 GCE Loss 作為一種穩健的替代方案。
        - **網址：** https://arxiv.org/abs/1805.07836
    - **Ma, T., et al. (2020).** ***Normalized Loss Functions for Deep Learning with Noisy Labels*****. (ICML 2020).**
        - **簡介：** 提供了另一種設計「噪聲穩健」損失函數的框架，並再次討論了標準 CE 在噪聲下的失敗。
        - **網址：** https://proceedings.mlr.press/v119/ma20d.html

Week6：

Q1
如果網路的正則性（平滑度）由啟動函數決定，那麼使用在 0 點不可微的 ReLU，是否會限制網路擬合「尖點」(sharp points) 附近的能力？如何量化這個限制？
這**不是 Open Problem**。這是深度學習「逼近理論 (Approximation Theory)」中的一個核心研究領域。

- **核心解法/分析：**
    i. **結論：** 恰好相反。ReLU 在 0 點的「不可微性」（或稱「kink」，扭結點）**正是它強大的來源**，它**特別擅長**擬合具有尖點或高頻細節的函數。
    ii. **原因：** 深度 ReLU 網路的本質是建構一個「**高維度的分段線性函數 (Piecewise Linear Function)**」。網路的每一層都在前一層的基礎上，透過 ReLU 的 "ON/OFF" 行為，對輸入空間進行更複雜的切割。
    iii. **擬合尖點：** 網路正是利用其自身架構中成千上萬個「kink」的組合，來「拼湊」出目標函數的尖點。相比之下，無限可微的平滑啟動函數（如 Sigmoid 或 Tanh）反而很難（需要極多參數）去擬合一個尖銳的角落。
    iv. **量化限制：** 這通常用「逼近率 (Approximation Rate)」來量化。研究表明，深度 ReLU 網路（深度 L）在逼近某些複雜函數類（如 Sobolev 空間）時，其逼近誤差 (error) 隨網路參數 N 的增加而下降的速率，遠快於淺層網路或使用平滑啟動函數的網路。
- **參考文獻與連結：**

- **Yarotsky, D. (2017). *Error bounds for approximations with deep ReLU networks*. (COLT 2017).**
    - **簡介：** 這是量化 ReLU 網路逼近能力的經典論文之一。它證明了深度 ReLU 網路在逼近特定函數時，所需參數 N 的數量級遠少於淺層網路，這很大程度上歸功於 ReLU 的分段線性特性。
    - **網址：** https://arxiv.org/abs/1610.01145

Q2:

1. 使用 $(\cos x, \sin x)$ 作為輸入時，週期長度是否可學習 (learnable)？
2. 如果週期未知，或存在多重週期（例如日週期 + 週週期），如何擴充輸入特徵？
   這**不是 Open Problem**。這是訊號處理和時間序列分析中常見的特徵工程問題，在深度學習中已有成熟的解決方案。

- **核心解法/分析：**
    i. **可學習的單一週期： 是，可學習**。
        - **方法：** 將輸入特徵改為 $(\cos(\omega x), \sin(\omega x))$。
        - 在這裡，角頻率 ω（與週期 $T = 2\pi/\omega$ 相關）可以作為網路的一個**可學習參數 (learnable parameter/weight)**，模型會透過反向傳播自動學習到最適合數據的 ω。

    ii. **未知/多重週期： 使用「傅立葉特徵 (Fourier Features)」**。
        - **方法：** 不要只用一組 $(\cos, \sin)$，而是用**一系列不同頻率**的 $(\cos, \sin)$ 組合來表示 x。
        - **擴充特徵：** 將 x 映射為一個高維向量：
          $[\cos(\omega_1 x), \sin(\omega_1 x), \cos(\omega_2 x), \sin(\omega_2 x), \ldots, \cos(\omega_N x), \sin(\omega_N x)]$
        - 這種方法讓網路可以同時捕捉到多種週期性（高頻和低頻）。這正是 **Transformer 模型**中「**位置編碼 (Positional Encoding)**」的核心思想。
        - 這些 ω_i 可以是**固定的**（例如 Transformer 中採用的指數間隔），也可以是**可學習的**（例如 NeRF 模型中的做法）。
- **參考文獻與連結：**
    - **Vaswani, A., et al. (2017). *Attention Is All You Need*. (NIPS 2017).**
        - **簡介：** 這篇 Transformer 的開創性論文。其「Positional Encoding」一節詳細說明了如何使用一組固定週期的 $\sin$ 和 $\cos$ 函數來表示序列中的位置，以捕捉不同尺度的週期性。
        - **網址：** https://papers.nips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf
    - **Tancik, M., et al. (2020). *Fourier Features Let Networks Learn High Frequency Functions...*. (NIPS 2020).**
        - **簡介：** 這篇論文（NeRF 的基礎之一）強烈建議使用「傅立葉特徵」（即多組 $\sin, \cos$）作為輸入，並展示了這種方法（無論 ω 是固定的還是可學習的）如何讓神經網路極大地提升擬合高頻訊號（複雜週期）的能力。
        - **網址：** https://arxiv.org/abs/2006.10739

Week7：

ISM (Implicit Score Matching) 因計算 $tr(\nabla_x s_\theta(x))$ 成本過高，發展出了 DSM 和 SSM 兩種可擴展的方法。問題：

1. DSM 和 SSM 之間的關係與優劣權衡？
2. 實務上如何二選一？
3. 兩者能否結合？

這**不是 Open Problem**。DSM 和 SSM 是 Score Matching 領域中兩條最主要、已被充分研究的技術路線。它們的優劣權衡（Trade-off）非常明確，是該領域的基礎。

- **核心解法/分析：**
  這兩種方法的核心差異在於它們如何處理 ISM 中昂貴的「Trace 項」($tr(\nabla_x s_\theta(x))$)，這導致了一個經典的「偏差 vs. 變異數」權衡 (Bias-Variance Trade-off)。
  i. 關係與優劣權衡
  Denoising Score Matching (DSM)
  - **核心思想： 迴避問題 (Avoidance)**。DSM 不去估計 *clean data* $p_x(x)$ 的 score，而是去估計 *noised data* $q_\sigma(\tilde{x})$ 的 score。
  - **方法：** DSM 證明，最小化 ISM 目標（針對 *noised data*）等價於一個非常簡單的**去噪迴歸任務**：
    $Loss_{DSM} = \mathbb{E}\left[\|s_\theta(\tilde{x}) - \nabla_{\tilde{x}} \log p(\tilde{x}|x)\|^2\right]$
    （當噪聲為高斯分佈 $\tilde{x} = x + \epsilon$ 時，$\nabla_{\tilde{x}} \log p(\tilde{x}|x) = -\epsilon/\sigma^2$）
  - **優點 (Pros)：**
    - **低變異數 (Low Variance)：** 是一個穩定的迴歸問題。
    - **計算高效：** Loss 計算非常簡單，不需要任何 Jacobian 或 Trace。
  - **缺點 (Cons)：**
    - **有偏差 (Biased)：** 這是最大的代價。DSM **學到的是「加噪後」分佈的 Score**，而不是「原始乾淨」分佈的 Score。它學到的是一個被「平滑化」的 Score。
    - **依賴 σ：** 效能嚴重依賴於噪聲等級 σ 的選擇。

  Sliced Score Matching (SSM)
  - **核心思想： 隨機估計 (Stochastic Estimation)**。SSM 試圖**無偏地**估計那個昂貴的 Trace 項。
  - **方法：** 基於 Hutchinson's Trick，將 $tr(A)$ 用 $\mathbb{E}_v[v^T A v]$ 來估計，其中 v 是隨機投影向量。SSM 的 Loss 變為：
    $Loss_{SSM} = \mathbb{E}_v \mathbb{E}_{p_x}\left[v^T \nabla_x s_\theta(x) v + \frac{1}{2}\|s_\theta(x)\|^2\right]$
    （$v^T \nabla_x s_\theta(x) v$ 這一項可以透過一次「vector-Jacobian product」高效算出，成本 $O(D)$
    ）

- **優點 (Pros)：**
  - **無偏差 (Unbiased)：** 這是最大的優勢。SSM 在期望值上**等於**昂貴的 ISM，因此它能準確學習**「原始乾淨」分佈的 Score** $\nabla_x \log p_x(x)$。
- **缺點 (Cons)：**
  - **高變異數 (High Variance)：** 作為一個隨機估計，SSM 的變異數非常高。在訓練中，它需要大量的隨機投影向量 v（或大 batch size）才能穩定收斂。

權衡總結：

- **DSM：** 低變異數、有偏差（學到的是平滑後的 Score）。
- **SSM：** 高變異數、無偏差（學到的是真實的 Score）。

ii. 實務上的選擇

- **選擇 DSM 的情況：**
  - **主要應用： 生成模型 (Generative Modeling)**，特別是**擴散模型 (Diffusion Models)**。
  - **原因：** 擴散模型的本質就是學習一系列「不同噪聲等級」下的 Score Function。DSM 的「偏差」（即學習加噪後的 Score）在 diffusion 框架下**反而成為了「特性」(feature) 而非「缺陷」(bug)**。你需要的就是在 σ_t 噪聲等級下，準確估計出 $\nabla \log q_{\sigma_t}(x)$。DSM 由於其低變異數和穩定性，成為了訓練擴散模型的完美選擇。
- **選擇 SSM 的情況：**
  - **主要應用：** 當你需要**準確估計「原始乾淨數據」的 Score** $\nabla_x \log p_x(x)$ 時。
  - **原因：** 例如在**科學建模**（如物理系統的能量函數 $U(x)$，其梯度 $-\nabla_x U(x)$ 就是 Score）、**變分推斷 (VI)** 或**密度估計 (Density Estimation)** 中，你關心的是數據本身的真實分佈，而不是加噪後的。在這種情況下，SSM 的「無偏差」特性至關重要。

iii. 是否可以結合？

是的，這是一個活躍的研究領域。

核心思想是利用 DSM 的「低變異數」特性來**改進** SSM 的「高變異數」問題。

- **方法：** 將 DSM 視為一個「控制變量 (Control Variate)」。
- **概念：** 我們知道 DSM 是一個有偏差但低變異數的估計，而 SSM 是無偏差但高變異數的。研究者們已經提出了一些方法，使用 DSM 作為 SSM 的一個「基線」(baseline)，然後只用 SSM 來估計兩者之間的「殘差」(residual)。
- **目的：** 這樣做的目標是得到一個**新的估計式**，它既具備 SSM 的「無偏差」（或漸近無偏差）特性，又具備 DSM 的「低變異數」特性。

- **參考文獻與連結：**
  - **DSM 原始論文 (奠定 Diffusion 基礎)：**
    - Vincent, P. (2011). *A Connection Between Score Matching and Denoising Autoencoders*. (Neural Computation).
    - **網址：** https://www.mitpressjournals.org/doi/abs/10.1162/NECO_a_00142
  - **SSM 原始論文 (解決 Trace 成本)：**
    - Song, Y., et al. (2019). *Sliced Score Matching: A Scalable Approach to Density and Score Estimation*. (UAI 2019).

        - **網址：** http://proceedings.mlr.press/v115/song20a.html
    - **SSM vs DSM 的討論 (可參考 Fig 1)：**
        - Song, Y., & Ermon, S. (2020). *Improved Techniques for Training Score-Based Generative Models*. (NIPS 2020).
        - **簡介：** 這篇是 NCSN++ 論文，它雖然主要使用 DSM（因為是生成模型），但在開頭和附錄中詳細討論了為何 DSM 在實務上（特別是流形數據）優於 SSM。
        - **網址：** https://arxiv.org/abs/2006.09011

Week8：

隨機微分方程 (SDE) 在機器學習中有哪些具體應用？它如何與擴散模型 (Diffusion Models) 產生關聯？

這**不是 Open Problem**。SDE 不僅與擴散模型相關，它已成為現代生成模型（Score-Based Generative Models）的**核心數學框架**，統一了先前如 DDPM 和 NCSN 等離散時間的模型。

- **核心解法/分析：**
  SDE 提供了將「離散時間」的擴散過程（一步步加噪）推廣到「**連續時間**」的強大工具。
  i. **Forward SDE (前向 SDE / 加噪過程)：**
    - 傳統擴散模型 (DDPM) 使用 T 個離散步驟將數據 x_0 轉化為噪聲 x_T。
    - SDE 框架將 $T \to \infty$，將這個過程描述為一個**連續時間**的 SDE。這個 SDE 描述了數據分佈 $p_0(x)$ 如何隨時間 t 平滑地演化成一個純高斯噪聲分佈 $p_T(x)$。
    - $dx = f(x,t)dt + g(t)dw$（w 是維納過程）
  ii. **Reverse SDE (反向 SDE / 生成過程)：**
    - **核心洞察：** 任何 SDE 都存在一個對應的「**反向時間 SDE**」，可以將 $t = T$ 的噪聲 $x(T)$ 逆轉回 $t = 0$ 的數據 $x(0)$。
    - **與 ML 的關聯：** 這個「反向 SDE」的公式**必須依賴一個關鍵項：Score Function** ($\nabla_x \log p_t(x)$)，即在 t 時刻加噪數據分佈的對數機率梯度。
    - $dx = [f(x,t) - g(t)^2 \nabla_x \log p_t(x)]dt + g(t)d\bar{w}$
  iii. **SDE 在擴散模型中的應用總結：**
    - **訓練：** 機器學習模型（神經網路）的任務就是**學習**（估計）所有 t 時刻的這個 **Score Function** $\nabla_x \log p_t(x)$（通常使用 Week 7 提到的 DSM 方法）。
    - **生成：** 從 $t = T$ 的純噪聲開始，使用數值 SDE 求解器（如 Euler-Maruyama）來**模擬**這個「反向 SDE」的軌跡，直到 $t = 0$ 時，輸出的就是一個生成的樣本。

  優勢： SDE 框架統一了 DDPM 和 NCSN，並允許更靈活的採樣（例如改變求解步數而無需重新訓練）以及更高級的求解器（如 Probability Flow ODE）。
- **參考文獻與連結：**
    - **Song, Y., et al. (2021). *Score-Based Generative Modeling with Stochastic Differential Equations*. (ICLR 2021).**

    - **簡介：** 這是**奠定 SDE 框架的關鍵論文**。它將「加噪」和「去噪」過程嚴謹地表述為前向和反向 SDE。
    - **網址：** https://arxiv.org/abs/2011.13456
- **Lilian Weng (2021).** ***What are Diffusion Models?***
    - **簡介：** 一篇非常清晰的部落格文章，詳細解釋了 SDE 如何作為 DDPM 和 NCSN 的統一框架。
    - **網址：** https://lilianweng.github.io/posts/2021-07-11-diffusion-models/

Week10：

在 1D OU 過程（擴散過程）的 Score Matching 中，訓練目標為何從「學習**邊際機率 (marginal)** $\nabla \log p(x_t, t)$」被轉換為「學習**條件機率 (conditional)** $\nabla \log p(x_t|x_0)$」？

1. 這兩個 score 相同嗎？
2. 若不同，為何可以用 $p(x_t|x_0)$ 的 score 來訓練？
3. 講義提到的 C_t 是否就是它們的差異？
   這**不是 Open Problem**。這個轉換是 Denoising Score Matching (DSM) 的核心，也是所有現代擴散模型（如 DDPM、SDE）能夠被高效訓練的**關鍵數學基礎**。

- **核心解法/分析：**
  i. **這兩個 Score 相同嗎？**
    - **不相同**。
    - $\nabla \log p(x_t|x_0)$ 是**條件 score** (Conditional Score)。它代表「給定一個**明確**的起點 x_0」，在 t 時刻 x_t 分佈的 score。這個分佈很簡單（就是一個高斯分佈），因此它的 score **計算非常容易**。
    - $\nabla \log p(x_t, t)$ 是**邊際 score** (Marginal Score)。它代表在 t 時刻 x_t 的**「混合」**分佈 $p(x_t, t) = \int p(x_t|x_0)p(x_0)dx_0$。這個分佈是所有可能的 x_0（來自你的真實數據分佈 $p(x_0)$）所產生的高斯分佈的**無限混合**，它非常複雜，其 score **無法直接得知**。

  ii. **為什麼可以用條件 Score 來訓練？**
    - **關鍵結論：** 雖然這兩個 **Loss 函數值**（$Loss_{marginal}$ 和 $Loss_{conditional}$）不相同，但它們對於模型參數 θ 的**梯度 (Gradient) 是完全相同的**。
    - $Loss_{marginal}(\theta) = \mathbb{E}_{p(x_t,t)}[||s_\theta(x_t) - \nabla \log p(x_t, t)||^2]$ （*目標，但很難*）
    - $Loss_{conditional}(\theta) = \mathbb{E}_{p(x_0)}\mathbb{E}_{p(x_t|x_0)}[||s_\theta(x_t) - \nabla \log p(x_t|x_0)||^2]$ （*實作，很簡單*）
    - 數學上可以證明： $\nabla_\theta Loss_{marginal}(\theta) = \nabla_\theta Loss_{conditional}(\theta)$
    - **因此：** 我們最小化「困難的」邊際 Loss，**等價於** 最小化「簡單的」條件 Loss。我們用 s_θ 去擬合 $\nabla \log p(x_t|x_0)$，就等於間接地（且正確地）學會了 $\nabla \log p(x_t, t)$。

  iii. **C_t (或 C_0) 是否就是它們的差異？**

- **完全正確**。
- 這兩個 Loss 函數之間的關係可以寫成：
$Loss_{marginal}(\theta) = Loss_{conditional}(\theta) + C_t$
- C_t 是一個**常數**，它只依賴於數據分佈 $p(x_0)$ 和噪聲過程 $p(x_t|x_0)$，**與模型參數 θ 無關**。
- 因為 C_t 是常數，它在求梯度 ∇_θ 時會被消除（$\nabla_\theta C_t = 0$），這再次印證了第 2 點（兩者的梯度相同）。
- 在優化中，$\arg\min_\theta(Loss_{conditional} + C_t)$ 和 $\arg\min_\theta Loss_{conditional}$ 的解是完全一樣的。因此，我們可以安全地忽略 C_t，只優化 $Loss_{conditional}$。

- **參考文獻與連結：**

  - **Vincent, P. (2011). *A Connection Between Score Matching and Denoising Autoencoders*. (Neural Computation).**
    - **簡介：** 這是**奠定 DSM 基礎**的論文。它首次證明了「Score Matching 目標（邊際）」等價於「Denoising（條件）」任務。
    - **網址：** https://www.mitpressjournals.org/doi/abs/10.1162/NECO_a_00142
  - **Song, Y., et al. (2021). *Score-Based Generative Modeling with Stochastic Differential Equations*. (ICLR 2021).**
    - **簡介：** 這篇 SDE 論文在其附錄 (Appendix B/C) 中也詳細推導了這個等價關係，並將其作為 SDE 框架的訓練基礎。
    - **網址：** https://arxiv.org/abs/2011.13456