Week 7 written assignment

1 什麼是「Score」?

在統計學中,「Score Function」(或簡稱 Score)是一個非常具體的概念。

想像一個機率分佈 $p(\mathbf{x})$。這就像一個高低起伏的「地圖」,其中 $\mathbf{x}$ 是地圖上的一個點(例如,一張 512×512 的圖片就是高維空間中的一個點)。

- 山谷 (低點):代表機率 $p(\mathbf{x})$ 高的地方,例如 $\mathbf{x}$ 是一張非常清晰、漂亮的貓咪照片。
- 山頂 (高點):代表機率 $p(\mathbf{x})$ 低的地方,例如 $\mathbf{x}$ 是一張充滿雜訊、無法辨識的圖片。

Score (分數) 的數學定義是對數機率的梯度 (gradient):

$$s(\mathbf{x}) = \nabla_{\mathbf{x}} \log p(\mathbf{x})$$

它的直觀意義是:

- 在 $\mathbf{x}$ 這一點,「Score」$s(\mathbf{x})$ 是一個向量(一個箭頭),它指向地圖上「機率上升最快」的方向。
- 如果你有一張充滿雜訊的圖片(在「山頂」),它的 Score 會指向「山谷」(貓咪圖片)的方向。
- 如果你有一張有點模糊的貓咪圖片(在「山谷」的半山腰上),它的 Score 會指向「山谷」的最底部(最清晰的貓咪圖片)。

問題:我們根本不知道真實世界的「地圖」$p(\mathbf{x})$ 長什麼樣子(我們不知道所有「好圖片」的機率分佈),所以我們無法直接計算 $s(\mathbf{x})$。

2 什麼是「Score Matching」?

Score Matching (評分匹配) 的目標就是:訓練一個神經網路 $s_\theta(\mathbf{x})$,讓它能夠模仿 (估計) 這個我們無法得知的真實 Score $s(\mathbf{x})$。

2.1 天真的作法

我們想最小化「我們的網路預測」和「真實 Score」之間的差距:

$$\text{Loss} = \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})} \left[\| s_\theta(\mathbf{x}) - \nabla_{\mathbf{x}} \log p(\mathbf{x}) \|^2 \right]$$

這個損失函數無法計算,因為我們不知道 $\nabla_{\mathbf{x}} \log p(\mathbf{x})$。

2.2 巧妙的作法 (Denoising Score Matching, DSM)

這一步是關鍵。研究人員發現 (Vincent, 2011)，與其匹配乾淨資料 $\mathbf{x}$ 的 Score，不如去匹配被雜訊污染過的資料 $\tilde{\mathbf{x}}$ 的 Score。

DSM 的訓練過程如下：

1. 取樣 **(Sample)**：從你的資料集（例如，一堆貓咪圖片）中隨機拿一張乾淨的圖片 $\mathbf{x}$。
2. 加噪 **(Perturb)**：加入一個已知的隨機高斯雜訊 ϵ，得到一張「被污染」的圖片 $\tilde{\mathbf{x}} = \mathbf{x} + \epsilon$。
3. 計算目標 **(Target)**：我們雖然不知道 $p(\tilde{\mathbf{x}})$ 的 Score，但我們可以計算「給定 $\mathbf{x}$ 情況下 $\tilde{\mathbf{x}}$」的 Score，即 $\nabla_{\tilde{\mathbf{x}}} \log p(\tilde{\mathbf{x}}|\mathbf{x})$。
   - 因為 $\tilde{\mathbf{x}}$ 是由 $\mathbf{x}$ 加上高斯雜訊 ϵ 產生的，這個 Score 竟然可以被精確計算出來，它就是 $-(\tilde{\mathbf{x}} - \mathbf{x})/\sigma^2$（其中 σ^2 是雜訊的強度）。
   - 因為 $\tilde{\mathbf{x}} - \mathbf{x} = \epsilon$，所以這個目標 Score 就是 $-\epsilon/\sigma^2$。
4. 訓練 **(Train)**：我們訓練神經網路 s_θ，當它看到受污染的圖片 $\tilde{\mathbf{x}}$ 時，它必須預測出 $-\epsilon/\sigma^2$。

2.3 更簡單的理解 (Denoising)

你會發現，要預測 $-\epsilon/\sigma^2$，其實等價於預測那個被加進去的雜訊 ϵ。

所以，Denoising Score Matching (DSM) 實際上就是在訓練一個「降噪器」(Denoising Model) $\epsilon_\theta(\tilde{\mathbf{x}})$。

- 輸入：一張 noisy 的圖片 $\tilde{\mathbf{x}}$。
- 輸出：網路 ϵ_θ 預測的雜訊 ϵ。
- 損失函數：$\text{Loss} = \mathbb{E}[\|\epsilon_\theta(\tilde{\mathbf{x}}) - \epsilon\|^2]$。（你預測的雜訊 ϵ_θ 和實際加入的雜訊 ϵ 越接近越好）。

3 Score Matching 如何用於擴散模型 (Diffusion Models)？

擴散模型巧妙地利用了這個「降噪器」來生成全新的圖片。

它包含兩個過程：

3.1 訓練過程 (Forward/Diffusion Process)

我們不只用一種雜訊強度，而是定義一個「雜訊時間表」，例如 $t = 1, 2, \ldots, 1000$。

- $t = 1$：加一點點雜訊。
- $t = 1000$：加超多雜訊，圖片變成純高斯雜訊（像電視雪花）。

我們的目標是訓練一個全能的降噪器 $\epsilon_\theta(\mathbf{x}_t, t)$，它必須能夠處理任何時間點 t 的 noisy 圖片 $\mathbf{x}_t$。

訓練步驟：

1. 隨機選一張乾淨圖片 $\mathbf{x}_0$。
2. 隨機選一個時間點 t（例如 $t = 350$）。
3. 根據 $t = 350$ 的雜訊強度，在 $\mathbf{x}_0$ 上加入雜訊 ϵ，得到 $\mathbf{x}_t$。
4. 將 $\mathbf{x}_t$ 和 t 輸入到神經網路 ϵ_θ。
5. 使用 Score Matching (DSM)：訓練網路 $\epsilon_\theta(\mathbf{x}_t, t)$，使其輸出的結果盡可能接近 ϵ。

3.2 生成過程 (Reverse/Sampling Process)

這就是我們實際「畫圖」的時候。我們反向執行這個過程：

1. 開始 **(Start)**：從 $t = 1000$ 開始。生成一張純雜訊的圖片 $\mathbf{x}_T$。
2. 迭代 **(Iterate)**：我們從 $t = 1000$ 逐步走回 $t = 0$。
3. 預測雜訊 **(Predict Noise)**：在 t 時刻，將當前的 noisy 圖片 $\mathbf{x}_t$ 和 t 輸入我們訓練好的網路 $\epsilon_\theta(\mathbf{x}_t, t)$。
4. 取得 **Score**：網路會「猜測」$\mathbf{x}_t$ 中包含的雜訊 ϵ。這個 ϵ 其實就隱含了 Score（指向更乾淨圖片的方向）。
5. 降噪一步 **(Denoise Step)**：我們使用這個預測出來的 ϵ，從 $\mathbf{x}_t$ 中「減去」一小部分雜訊，得到 $\mathbf{x}_{t-1}$。這一步在數學上被稱為 Langevin Dynamics（朗之萬動力學）或擴散模型的反向 SDE。
6. 重複 **(Repeat)**：重複這個過程（預測雜訊 -> 減去雜訊），$\mathbf{x}_{999} \to \mathbf{x}_{998} \to \cdots \to \mathbf{x}_1$。
7. 完成 **(Finish)**：當 $t = 0$ 時，$\mathbf{x}_0$ 就是一張從純雜訊中「雕刻」出來的全新、清晰的圖片。

4 Unanswered Questions

ISM 的損失函數 $L_{ISM}(\theta)$ 包含 $\nabla_x \cdot S(x;\theta)$ 這一項，也就是 score function 的 divergence。在實務上，特別是當 $S(x;\theta)$ 是一個高維度的深度神經網路時，這個散度項是如何被有效計算的？