Reflextion Report
FrozenLake 8x8 Q-Learning 實作

1. Introduction

這次的專案是 Gymnasium 的FrozenLake-8x8（Slippery Mode）環境，目標是訓練一個 Agent 能夠在這個充滿隨機性的冰湖上找到終點。透過調整參數並稍微更改程式碼來達成目標 70%的成功率。

2. Implementation Strategy

在實作 Q-Learning 的過程中，我針對環境特性做調整。我主要做了以下三個關鍵改動：

2.1 初始化的「向左」偏差

剛開始訓練時，我發現一個有趣的 Bug：因為 Q-table 一開始全是 0，程式碼中的np.argmax函數預設會一直回傳0（也就是向左走）。這導致 Agent 像著魔一樣一直往左邊撞牆，根本沒在探索。為了解決這個問題，我加了一段邏輯：「當大家分數都一樣時，就隨機選；只有分數有高低之分時，才選最高的。」這一個小改動，直接讓 Agent 在初期的探索效率提升了很多。

2.2 Epsilon 衰減 (Exploration Decay)

FrozenLake 的滑溜特性的運氣成分很重。如果太快讓 Agent 停止探索（Epsilon 降太快），它很容易誤以為某條路是最好的，結果其實只是剛好運氣好沒滑倒而已。 因此，我讓他前 60% 時間都在探索，後面認真使用探索學到的東西。總共次數是 15000 次，讓 Epsilon 慢慢地線性下降，直到第 9000 回合才降到最低。這就像是強迫 Agent 在做最後決定前，先把地圖徹底摸熟。

2.3 學習率的調整

我嘗試過很多不同的 Learning Rate，最後發現保持在0.1這種較低且固定的數值效果最好。因為環境太 Noisy，太高的學習率會讓 Agent 的判斷一直跳動，反而學不好。

3. Results & Challenges

訓練過程比我想像中漫長。一開始看著 Reward 曲線一直貼在地板上（都是 0），我一度懷疑是不是程式寫錯了。但跑到第 7000 回合左右，曲線終於開始爬升，這讓我有很大的成就感。

最終在測試階段，我的 Agent 達到了**60%以上的成功率**。 雖然離 100%還有距離，但在is_slippery=True的模式下，這其實已經是非常好的成績。因為很多時候掉進冰洞是不可控的隨機事件，Agent 選擇了正確的路，卻被環境「推」進了洞裡，這是強化學習中必須接受的無奈與現實。

4. GitHub 學習心得

在這次專案中，除了演算法本身，我也學習了如何使用 GitHub 來管理我的程式碼。這對我來說是一個很重要的軟實力提升，因為我在修這堂課程之前是完全不會使用 GitHub 的，但是老師強迫我們把檔案上傳到 GitHub 上，讓我因此學到了很多跟 GitHub 有關的操作。除此之外，我也學到了專案合作，GitHub 可以讓我們互相去分享我們的 code，讓我們溝通起來更順暢，也比較不會有進度不同的問題，每一次的上傳紀錄都可以讓我們去看說哪個時候改了什麼，就不怕會有不小心更改錯誤的情況

5. 結論與反思 (Conclusion)

這次專案給我最大的體悟是：「機器學習不只是寫 code，更多的是在觀察數據和調整策略。」

一開始我只想趕快看到結果，把參數亂調，結果完全不收斂。後來我靜下心來，去思考 Epsilon 衰減的意義，去觀察argmax的行為，才真正把問題解決。此外，配合 GitHub 的使用，讓我更有條理地管理我的嘗試過程。這些經驗對於我未來學習更複雜的深度強化學習（Deep RL）是非常寶貴的基礎。