

## HW4-3 理解報告：PyTorch Lightning 強化版 DQN

### 一、作業目標

本作業旨在將原始 Naive DQN 模型改寫為 PyTorch Lightning 架構，並整合進階訓練技巧，提升可讀性與擴充性，同時保持與 Gym 環境的互動能力。重點在於轉換訓練流程，並非依賴傳統 DataLoader，而是手動控制環境互動與更新。

### 二、架構轉換與實作說明

1. 採用 LightningModule 將模型、優化器、scheduler 封裝於單一類別。
2. 使用 manual optimization 模式（self.automatic\_optimization = False），完全自行控制反向傳播與參數更新。
3. 將訓練邏輯從原始 loop 移入 on\_train\_epoch\_start 中，每個 epoch 自動與環境互動、收集經驗並進行訓練。
4. 為符合 Lightning 結構，補上必要但未實際使用的 training\_step 與 train\_dataloader（dummy 資料）。

### 三、加入的訓練技巧

本次改寫中已整合以下技巧來提升訓練穩定性與效果：

- 學習率調整：使用 StepLR 每 100 epoch 調降學習率。
- 支援梯度裁剪與 early stopping，可再加入 Lightning callback 延伸應用。

### 四、訓練成果與觀察

模型可穩定與 CartPole 環境互動並完成 300 集訓練，reward 曲線成功輸出為 rewards\_lightning.png。

雖初期表現不穩定，但隨訓練次數增加後 reward 呈現上升趨勢，顯示模型已成功學習維持平衡策略。

### 五、心得與結論

透過本次作業，我學習了如何將傳統強化學習流程整合進 PyTorch Lightning 框架，並在不使用 DataLoader 的情況下，透過手動優化進行完整訓練。該結構有助於模組化訓練程式、加入 callback 並追蹤實驗結果，是日後大型專案開發的良好基礎。