# 報告：在 Atari 上的 Decision Transformer 作業

## 1. 簡介

本作業實現並驗證 Decision Transformer 在 Atari 遊戲環境的應用。Decision Transformer 將強化學習問題轉化為序列建模，使用自回歸 Transformer 在已知狀態與期望回報條件下，預測下一步動作，而非傳統的價值函數或策略梯度方法。本報告將涵蓋：

1. Decision Transformer 的核心概念。

2. 在 Atari 環境下所做的主要改動。

3. 實作版本與論文版本之間的差異比較。

## 2. Decision Transformer 核心概念

### 2.1 序列建模視角

- 傳統強化學習：透過 Q-learning、策略梯度、Actor–Critic 等方法，最大化累積回報。

- Decision Transformer：將強化學習視為「條件序列生成」問題：

- 輸入序列包含「狀態 (state)」、「動作 (action)」、「回報到目前為止 (returns-to-go)」及「時間步 (timestep)」。

- 模型自回歸地預測下一步動作，條件是已觀測到的狀態和對未來回報的期望。

### 2.2 模型架構

1. 輸入嵌入 (Input Embeddings)：

- 狀態嵌入：將原始影像觀測映射到嵌入維度。

- 動作嵌入：將離散動作編號映射到同維度向量。

- 回報嵌入：將回報到目前為止向量化。

- 時間步嵌入：位置編碼 (Position Embedding)。

2. Transformer 編碼器：多層自注意力堆疊，捕捉序列中跨時間的依賴。

3. 預測頭 (Head)：線性層輸出下一步動作的 logit。

4. 損失 (Loss)：均方誤差 (MSE) 用於動作或回報預測誤差。

## 3. Atari 環境下的主要改動

|  |  |  |
| --- | --- | --- |
| 改動項目 | 論文版本（MuJoCo 連續控制） | 本實作（Atari 離散動作） |
| 動作空間 | 連續動作 | 離散動作（Breakout、Pong、Qbert 等） |
| 觀測空間 | 低維數值狀態 | 84×84 灰階畫面幀 |
| 狀態嵌入 | 小型 MLP 或直接向量 | 展平後影像張量線性映射 |
| 回報計算 | 小尺度實數回報 | 遊戲得分逐步累加 |
| 資料生成 | 隨機蒐集長軌跡 | 批次讀取 .npz replay logs，清洗 + padding |
| 批次大小 | 較小 (e.g.16) | 128 |
| 學習率 | 較小 (e.g.1e-4) | 6e-4 |

## 4. 實作版本與論文差異

1. 模型細節：

- 狀態嵌入：使用 nn.Linear(84\*84\*history, n\_embd) vs. 小型 MLP。

- 時間步編碼：nn.Embedding vs. 正弦/餘弦位置編碼。

2. 訓練流程：支援多 GPU DataParallel、warmup/cosine decay vs. 論文 schedule。

3. 資料處理：自動移除空檔 + padding vs. 手動整理。

4. 可視化：loss\_history.npy + loss\_curve.png vs. 論文圖表。

## 5. 未來改進方向

- 評估管線：定期執行 get\_returns() 並繪製回報曲線。

- 超參數搜尋：自動化搜尋最佳層數、頭數、學習率與批次大小。

- 儀表板：整合 TensorBoard 或 Weights & Biases。

結論：本實作成功將 Decision Transformer 遷移至 Atari 離散動作環境，在保持核心演算法架構不變下，完成了影像嵌入、資料前處理與訓練流程優化，提升訓練穩定性與可視化能力。