

AI HW5

洪郡辰

R11944050

May 25, 2024

Q1

介紹

深度 Q 網絡 (Deep Q-Network, DQN) 是一種結合了深度學習和強化學習的算法，用於解決具有高維度狀態空間的強化學習問題。DQN 通過使用深度神經網絡來近似 Q 值函數，從而使得算法能夠在複雜的環境中進行有效的決策。以下是 DQN 的主要概念和工作流程：

核心概念

- **Q 值函數**：Q 值函數 $Q(s, a)$ 表示在給定狀態 s 下選擇動作 a 所能獲得的預期回報。DQN 通過學習 Q 值函數來讓 agent 選擇最優動作，在作業中是實作 CNN 作為 Q 值函數。
- **經驗回放 (Experience Replay)**：agent 在與環境交互過程中會生成大量的狀態-動作-獎勵-下一狀態-終止狀態 (s, a, r, s', t) 的五元組，這些五元組存儲在經驗回放緩衝區中。在訓練時，從緩衝區中隨機抽取小批量樣本進行訓練，從而打破樣本之間的時間相關性，提升訓練的穩定性和效率。
- **目標網絡 (Target Network)**：DQN 引入了一個目標網絡，該網絡的參數每隔一定步數才更新一次。這一技巧能夠緩解 Q 值估計的不穩定性，因為訓練過程中使用的目標值不會頻繁變化。

工作流程

1. **初始化**：初始化 Q 網絡的參數和目標網絡的參數，並初始化經驗回放緩衝區。
2. **交互 (act)**：agent 根據當前狀態通過 ϵ -貪婪 (epsilon-greedy) 策略選擇動作，即以概率 ϵ 隨機選擇動作，以概率 $1 - \epsilon$ 選擇當前 Q 網絡輸出 Q 值最大的動作。
3. **儲存**：將執行動作後得到的新的五元組 (s, a, r, s', t) 存儲到經驗回放緩衝區中。

4. 採樣與訓練：

- 從經驗回放緩衝區中隨機抽取一個小批量的五元組。
- 計算給定 action 及 state 的當前網絡 Q 值和 next state 的目標網路 Q 值，目標 Q 值通過目標網絡計算得到。
- 計算 td target，它是用於更新 Q 值函數的目標值。在 DQN 中，td target 是用來計算當前狀態-動作對的預期回報，從而指導神經網絡的學習過程。
- 使用損失函數（smooth l1 loss）計算當前網絡 Q 值及 td target 的誤差，並通過反向傳播算法更新 Q 網絡的參數。

5. 更新目標網絡及參數：每隔固定的步數將 Q 網絡的參數複製到目標網絡中以及將網路超參數進行更新。

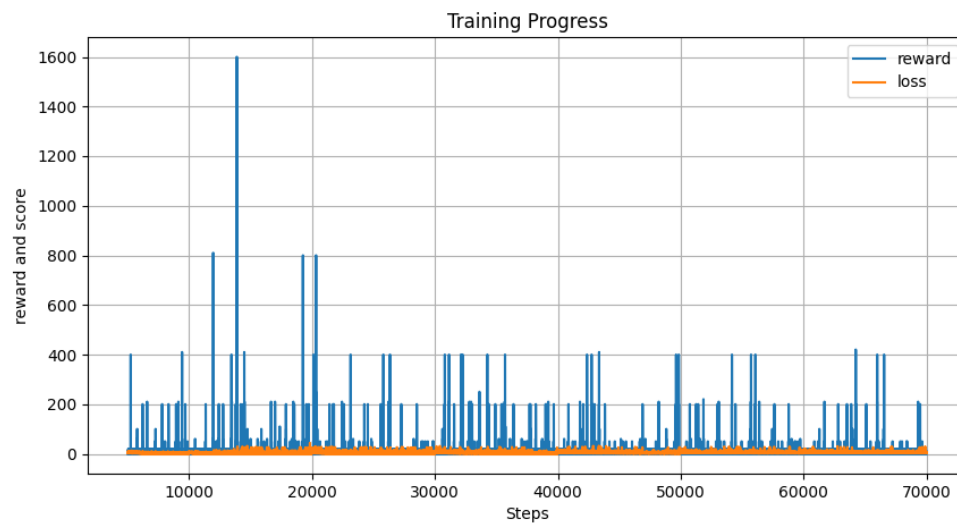
Q2

```
class PacmanActionCNN(nn.Module):
    def __init__(self, state_dim, action_dim):
        super(PacmanActionCNN, self).__init__()
        # Define the CNN architecture
        self.conv1 = nn.Conv2d(state_dim, 8, kernel_size=3, stride=1)
        self.conv2 = nn.Conv2d(8, 16, kernel_size=3, stride=1)
        self.conv3 = nn.Conv2d(16, 32, kernel_size=4, stride=1)
        self.fc1 = nn.Linear(
            32 * 77 * 77, 512
        )
        self.fc2 = nn.Linear(512, action_dim)

    def forward(self, x):
        x = F.relu(self.conv1(x))
        x = F.relu(self.conv2(x))
        x = F.relu(self.conv3(x))
        x = x.view(x.size(0), -1)
        x = F.relu(self.fc1(x))
        x = self.fc2(x)
        return x
```

我使用三層 Convolution 以及兩層 linear 建構 PacmanActionCNN，各層細節如上圖所示。經過三層卷積層後，會變成 32*77*77，透過 x.view() 將其 flatten，再接上 linear 轉成 512 維，再轉成 action dim。

Q3



在出現 avg score 大於 1500 時，就畫出當下的 reward and loss 圖。

Q4

