中山大学计算机院本科生实验报告
（2024学年秋季学期）

课程名称：强化学习与博弈论　　批改人：

实验	Assignment3	专业（方向）	计算机科学与技术计科一班
学号	21307099	姓名	李英骏
Email	liyj323@mail2.sysu.edu.cn	完成日期	2024 年 12 月 3 日

目录

1 应用案例说明 **2**

2 算法实现核心思路 **2**

2.1 算法公式 2

2.2 算法伪代码 2

3 实验结果及分析 **3**

3.1 训练曲线 3

3.2 测试表现 3

3.3 GIF生成 3

4 结论 **4**

5 代码链接 **4**

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

1 应用案例说明

CarRacing是OpenAI Gym提供的一个经典强化学习环境，用于测试智能体在连续动作空间中控制车辆的能力。本实验使用DQN（Deep Q-Network）算法解决该问题，训练智能体使其能够在给定赛道上尽可能长时间保持行驶并获得高分。

2 算法实现核心思路

DQN算法结合了深度学习和强化学习的思想，通过卷积神经网络（CNN）近似动作值函数（Q函数）。

2.1 算法公式

DQN的核心公式如下：

$$Q(s_t, a_t; \theta) = r_t + \gamma \max_{a'} Q(s_{t+1}, a'; \theta^-), \tag{1}$$

其中：

- s_t 和 s_{t+1} 分别表示当前状态和下一状态；
- a_t 和 a' 分别表示当前动作和下一动作；
- r_t 表示即时奖励；
- γ 为折扣因子，控制未来奖励的权重；
- θ 和 θ^- 分别表示当前网络和目标网络的参数。

目标是通过最小化均方误差（MSE）优化网络参数：

$$L(\theta) = \mathbb{E}\big[(y_t - Q(s_t, a_t; \theta))^2\big], \tag{2}$$

其中：

$$y_t = r_t + \gamma \max_{a'} Q(s_{t+1}, a'; \theta^-). \tag{3}$$

2.2 算法伪代码

以下为DQN的核心伪代码：

```
Initialize policy network Q with weights \theta
Initialize target network Q' with weights \theta^- = \theta
Initialize replay buffer D

for episode in num_episodes do
    Initialize state s
```

```
    for step in max_steps do
        With probability \epsilon select a random action a
        Otherwise select a = argmax Q(s, a; \theta)
        Execute action a, observe reward r and next state s'
        Store (s, a, r, s') in replay buffer D
        Sample random mini-batch from D
        Compute target y and loss L
        Perform gradient descent to update \theta
        Update target network \theta^- periodically
    end for
end for
```

3 实验结果及分析

3.1 训练曲线

在训练过程中，我们记录了每一回合的总奖励（Reward）。如图2所示，随着训练的进行，智能体的表现逐渐提升。

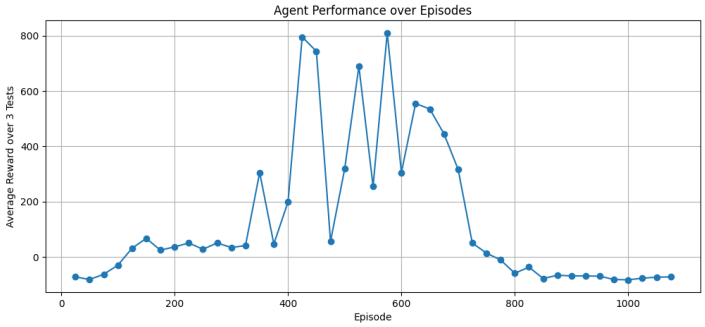

图 1: 训练过程中每回合的奖励曲线

3.2 测试表现

在测试阶段，使用训练好的模型对智能体进行评估。下图**??**展示了智能体在CarRacing环境中的表现。

3.3 GIF生成

我们进一步生成了智能体在测试过程中的行为GIF，用于直观地展示智能体的表现。

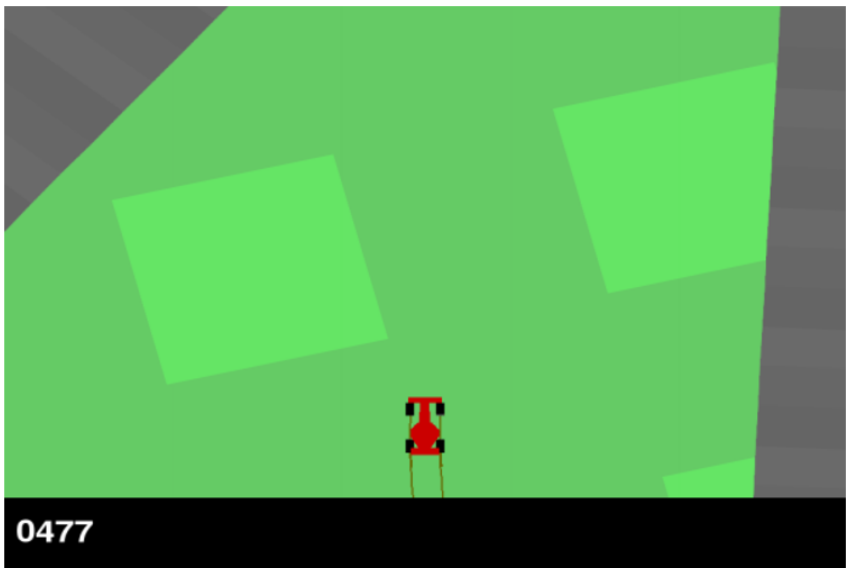

图 2: 训练过程中每回合的奖励曲线

4 结论

本实验通过DQN算法成功训练了一个能够在CarRacing环境中表现良好的智能体。实验结果表明：

- DQN算法能够有效解决连续动作空间的强化学习问题；
- 增加Replay Buffer和目标网络的设计可以提升训练稳定性。

未来可以考虑扩展为双重DQN（Double DQN）或使用更复杂的强化学习算法以进一步提升性能。

5 代码链接

完整代码已上传至GitHub仓库。