四川大学计算机学院、软件学院

实验报告

学号： 姓名： 专业： 班级：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 课程名称 | 深度强化学习 | | 实验课时 | 2 |
| 实验项目 | gym实验 | | 实验时间 | 3.24 |
| 实验目的 | 1. 探究DQN强化学习算法在carpole-v0实验中的表现； 2. 研究如何在强化学习中使用深度学习模型； 3. 了解强化学习中的agent、state、reward、action等概念。 | | | |
| 实验环境 | Windows + VsCode + PyTorch + OpenAI Gym + Python | | | |
| 实验内容 | 1. 首先安装Python、OpenAI Gym以及PyTorch等库。并且需要实现一个强化学习算法，来实现OpenAI Gym中的carpole-v0实验。 2. ReplayBuffer类：用于存储智能体的经验，并在训练时提供经验回放机制。将每个状态转移保存在ReplayBuffer中，包括当前状态、动作、奖励、下一个状态以及是否终止等信息。 3. Args\_settings 文件，定义了args相关参数 4. Qnet类：定义网络的结构 5. DQN类：实现强化学习算法。里面包括了： 6. def init(self): # 定义DQN的初始化参数 7. def select\_action(self, state): # epsilon-贪婪策略采取动作 8. def update(self, transition): # 更新函数 9. def run(self): # 训练函数 10. def plot\_reward(self, reward\_list): # 绘制图像 | | | |
| 结论  （结果） |  | | | |
| 小结 | 本次实验中，实现了基于深度强化学习的DQN算法，使用该算法解决了OpenAI Gym中的carpole-v0实验。在实验过程中，我们首先介绍了DQN算法的原理，并对其核心概念进行了讲解，包括Q值函数、经验回放和目标网络等。接着，我们实现了DQN算法中的关键组件，包括神经网络、回放缓存和DQN代理等。  在实验中，我们通过修改代码中的超参数，如学习率、批次大小和训练轮数等，来寻找最佳的模型超参数，从而提高DQN算法在carpole-v0的性能。通过实验结果的分析，我们发现模型超参数的调整能够对模型性能产生重大影响。这表明了DQN算法在解决强化学习问题上的有效性。  总体而言，本次实验对于我们理解深度强化学习的原理和应用具有重要的意义。通过实践，我们更深入地理解了DQN算法，并学会了如何在实际中使用该算法解决强化学习问题。同时，我们也学会了如何调整模型超参数以优化算法性能。这些知识和技能将对我们今后在深度强化学习领域的研究和实践中起到重要的指导作用。 | | | |
| 指导老师  评议 |  | | | |
| 成绩评定： | 指导教师签名： | | |