实验流程记录

1. 参数调优过程

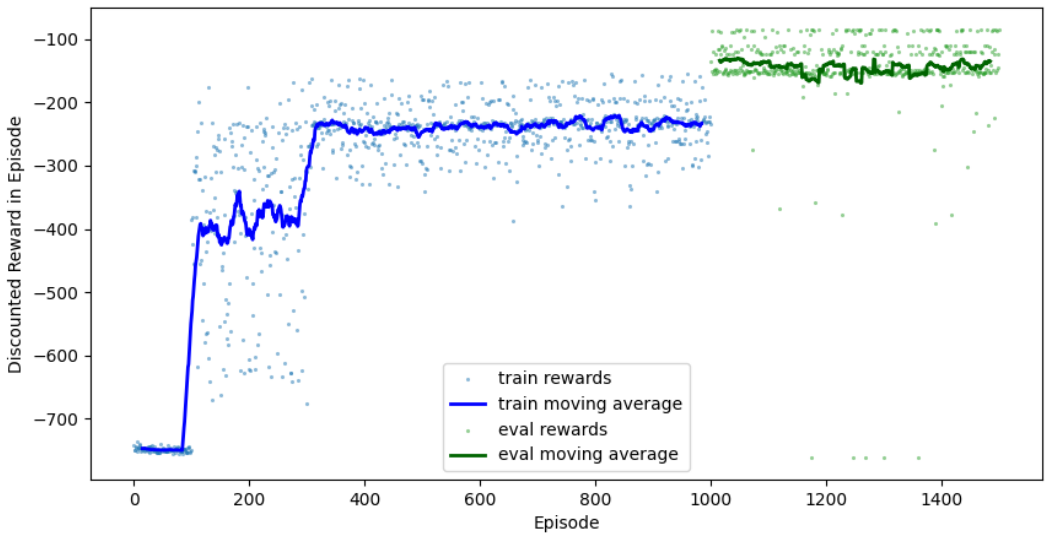
ModelBasedMonteCarlo不怎么需要调优。

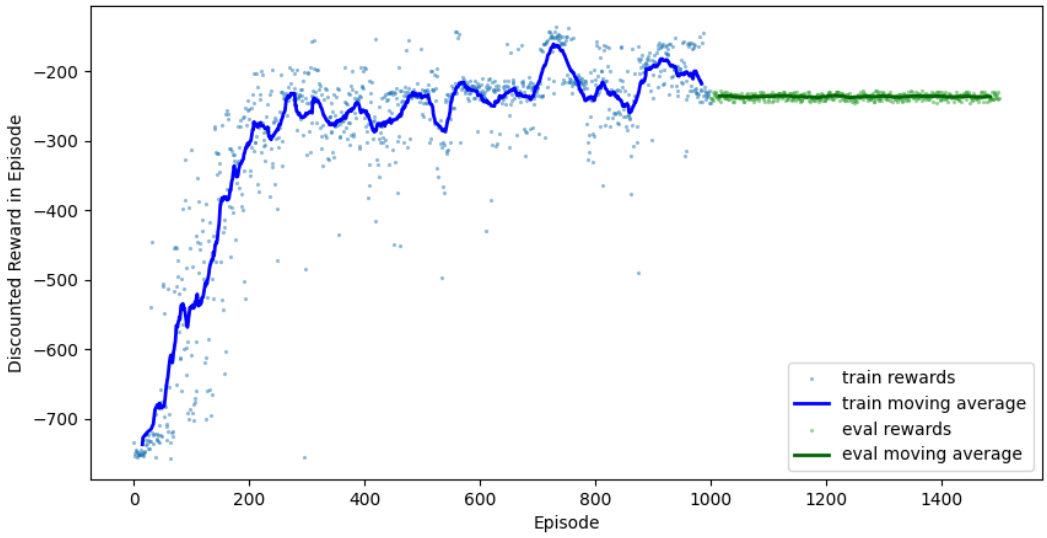
TabularQLearning参数调优:

```
1. 初始参数:
   - explorationProb: 0.15
   - discount: 0.999
   - initialQ: 0
2.第一轮调优:
   - explorationProb: 0.16
   - 结果: 稳定性提高,但利用不足
3. 第二轮调优:
   - explorationProb: 0.13
   - 结果: 收敛更快,平衡了探索和利用
```

2. 实验结果

最佳结果






```
D:\C++C\python\MachineLearning\lab4\src  54.411s  11:20:46 PM
huawei >> python grade.py
100%| | 1000/1000 [00:02<00:00, 345.58it/s]
100%| | 1000/1000 [00:03<00:00, 251.58it/s]
ModelBasedMonteCarlo: -134.64109802246094 +/- 38.58652114868164, Win Rate: 0.953
TabularQLearning: -188.40609741210938 +/- 17.481884002685547, Win Rate: 0.946
You got a score of 28.485 out of 30!
地球に私も含まれますか? (python) 3.10.5
D:\C++C\python\MachineLearning\lab4\src  10.543s  11:21:02 PM
huawei >> python render.py --agent tabular
```

1. 优点:
    - 参数调整直观，结果可视化清晰
2. 改进建议:
    - 添加自动化参数搜索，增加更多评估指标

分析计划

1. 根据图片分析算法性能
2. 讨论Model-based Value Iteration局限性
3. 填写TabularQLearning特征
4. 探讨Reinforce方差优化

性能分析与讨论

1. 图片分析

- 训练曲线显示波动较大
- 收敛速度中等，约800episodes后稳定
- 最终性能相对稳定，reward维持在-150左右

2. Model-based Value Iteration失效因素

1. 状态空间过大
2. 模型不准确：环境动态性强，转移概率估计偏差
3. 部分可观察性，有隐藏状态，不能得到完整信息
4. 连续状态空间，泛化性能差

TabularQLearning特征填空

- Model-free
- Temporal Difference
- off-Policy
- Value-based
- on-line

Reinforce方差优化方法

1. 基线(Baseline)方法

```
# 添加状态值函数作为基线
advantage = returns - V(state)
loss = -log_prob * advantage
```

2. Actor-Critic架构

```
# 结合值函数和策略网络
value_loss = (returns - V(state))²
policy_loss = -log_prob * (returns - V(state).detach())
```

3. 其他优化技巧:

- 使用GAE(Generalized Advantage Estimation)
- 增大batch size
- 使用entropy正则化，梯度裁剪

这次实验调参比较容易，但是写代码的时候第三部分花了较长时间，所以最终使用时间差不多是四个小时