时序差分算法实验报告

姓名 学号

## 一、实验要求

（1）阅读《基于时序差分的路径规划-实验指导书》，尝试运行并理解时序差分算法在冰湖路径规划问题上的示例代码。

（2）在示例代码的基础上，尝试实现n-step TD算法，比较并分析n的不同取值对策略收敛速度的影响（由于随机性建议多次实验）。

（3）在示例代码的基础上，尝试实现TD(λ)算法，比较并分析λ的不同取值对策略收敛速度的影响（由于随机性建议多次实验）。

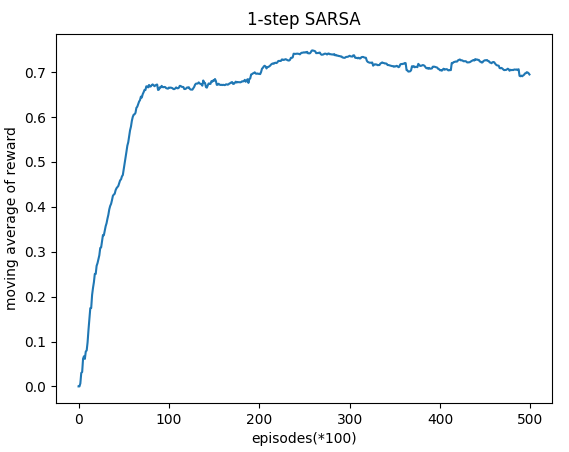
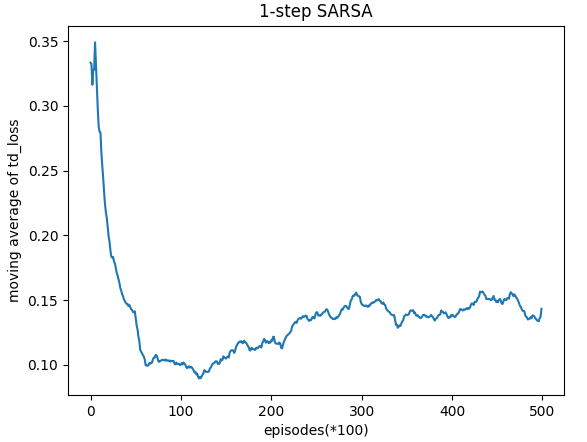
## 二、代码修改思路

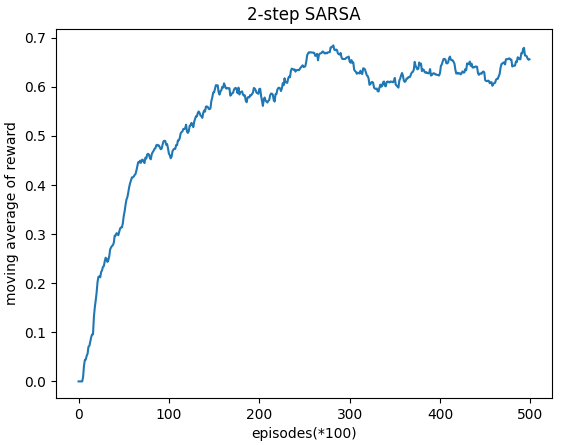
在SARSA的基础上实现了n-step SARSA算法。思路与SARSA类似，区别在于在与环境交互的过程中存储states, actions, rewards三个列表，在交互结束后通过列表计算n-step的td\_target和td\_error，对Q数组进行更新。

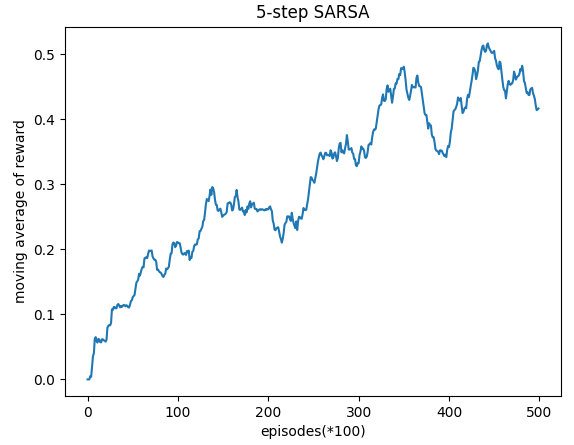
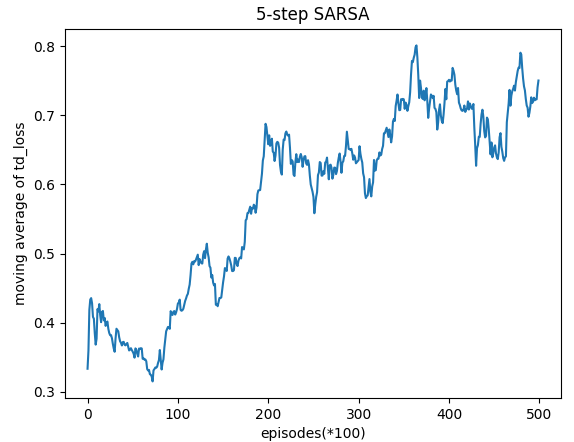
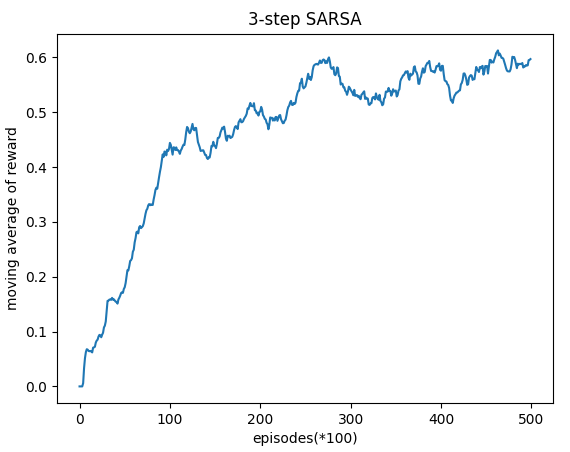
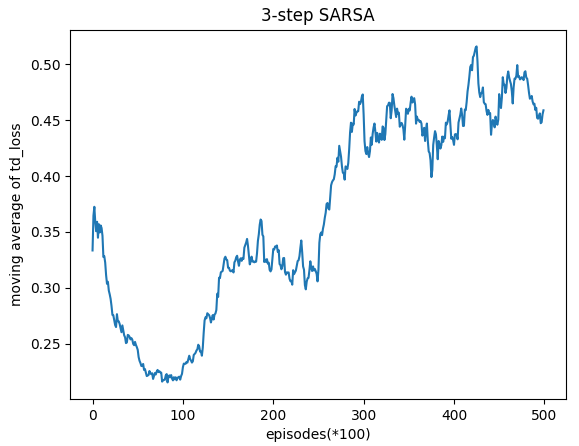
在SARSA的基础上实现了SARSA(lambda)算法。思路与SARSA类似，区别在于加入了资格迹矩阵，这个矩阵的大小为state\_number\*action\_number，表示了在从前向后进行更新时，需要为前面走过的Q[s, a] 加上的系数。

为了评估收敛程度，加入Getloss函数，这个函数以Q, env.P, gamma作为参数，枚举每个s,a对，计算Q[s, a] 与 target 的差值的平方并加和。

## 三、实验结果和分析







以上是对于n-step SARSA在n=1,2,3,5下的测试结果。首先可以发现的是收敛速度（收敛到相同td\_loss所需的episodes数量）随着n的增大而降低，收敛到的策略效果也随着n的增大而变差。可能的原因是本次的实验环境具有马尔可夫性，且只在终点有大小为1的reward。另一个可以发现的是td\_loss在开始逐渐降低，但在约10000个episodes之后开始升高。可能的原因有两方面，一是从reward可以发现此时收敛到的策略并不是最优策略，在此之后开始了对更优策略的探索；二是td\_loss计算的是所有s,a对的误差，随着episodes的增加epsilon不断降低，对一些不在较优策略中的状态和动作的探索越来越少，这些位置的Q值长期得不到更新。

下面是对于SARSA(lambda) 在lambda=0.1, 0.4, 0.7, 1下的测试结果。观察是与n-step SARSA类似的，背后的原因也是如此。lambda参数越大就会越多的使用多个step的信息，这导致了收敛速度和收敛效果等不佳。

