**第四次作业**

黄磊 计702 2022E8013282156

1. **简答题**
2. **试列举几种常见的半监督学习方法。比较有监督学习、无监督学习、半监督学习、 主动学习以及强化学习的异同；**

半监督学习：自训练方法、多视角算法、生成模型、转导SVM，基于图的算法。

监督学习最大的特点就是其数据集带有标签。换句话来说，监督学习就是在带有标签的训练数据中学习模型，然后对某个给定的新数据利用模型预测它的标签。

让学习器不依赖外界交互、自动地利用未标记样本来提升学习性能，就是半监督学习，将大量的无类标签的样例加入到有限的有类标签的样本中一起训练来进行学习，以达到提升性能的目的。

无监督学习的特点是，模型学习的数据没有标签，因此无监督学习的目标是通过对这些无标签样本的学习来揭示数据的内在特性及规律，其代表就是聚类。

自监督学习主要是利用辅助任务（pretext）从大规模的无监督数据中挖掘自身的监督信息，通过这种构造的监督信息对网络进行训练，从而可以学习到对下游任务有价值的表征。

主动学习算法可以交互式地查询用户以用真实标签标记新的数据点，被允许从尚未标记的样本池中主动选择下一个要标记的可用样本子集。其基本思想是：如果允许机器学习算法选择它想要学习的数据，它可以在使用更少的训练标签的同时实现更高的精度。

强化学习（Reinforcement learning，RL）讨论的问题是一个智能体(agent) 怎么在一个复杂不确定的 环境(environment) 里面去极大化它能获得的奖励。通过感知所处环境的 状态(state) 对 动作(action) 的 反应(reward)， 来指导更好的动作，从而获得最大的 收益(return)，这被称为在交互中学习，这样的学习方法就被称作强化学习

1. **试给出协同训练的方法步骤；**

输入：标记数据集L，未标记数据集U。

* 用L1训练视图X1上的分类器f1，用L2训练视图X2上的分类器f2；
* 用f1和f2分别对未标记数据U进行分类；
* 把f1对U的分类结果中，前k个最置信的数据（正例p个反例n个）及其分类结果加入L2；把f2对U的分类结果中，前k个最置信的数据及其分类结果加入L1；把这2（p+n）个数据从U中移除；
* 重复上述过程，直到U为空集。

输出：分类器f1和f2。其中f1和f2可以是同一种分类器也可以不是同一种分类器。

**二、 编程题**

SARSA和Q-Learning都是时序差分算法，SARSA使用on-policy策略，而Q-Learning使用的是off-policy策略。

Sarsa的思想是先基于当前状态S，使用ϵ−贪婪法按一定概率选择动作A，然后得到奖励R，并更新进入新状态S′，基于状态S′，使用ϵ−贪婪法选择A′（即在线选择，仍然使用同样的ϵ−贪婪），即：

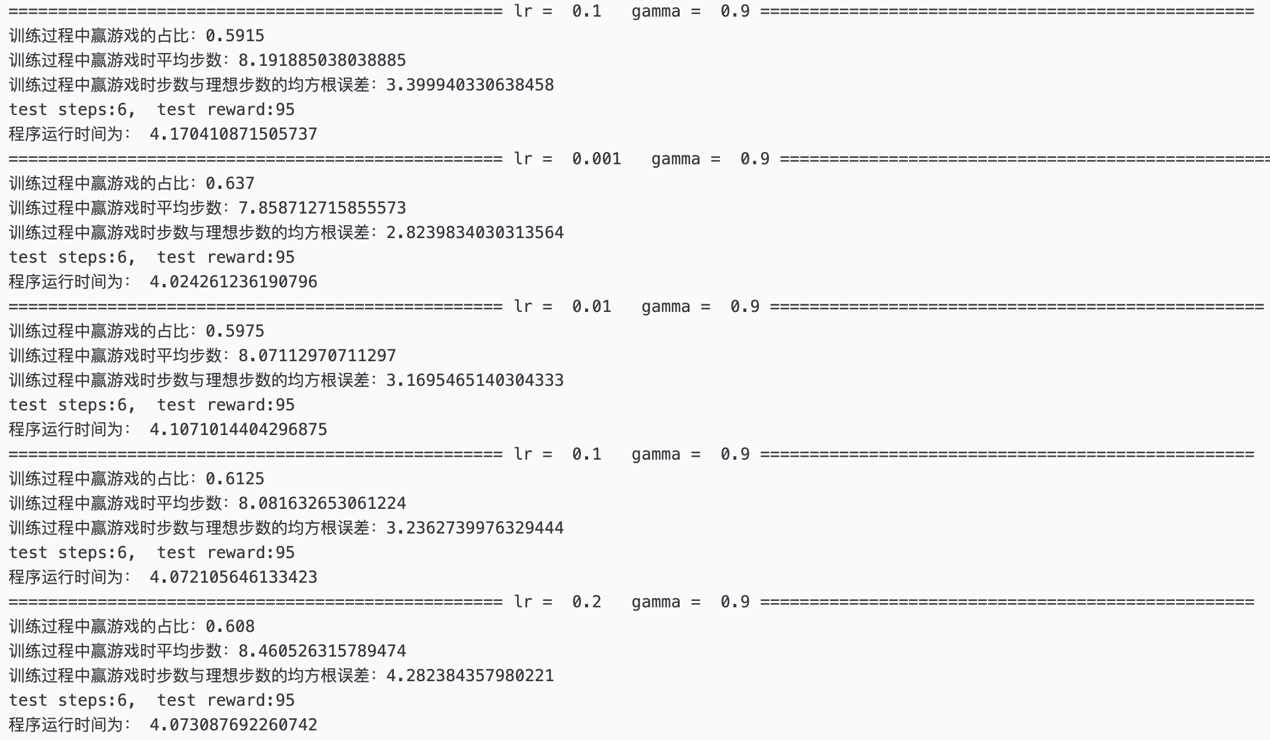
Q-Learnig的思想就是，先基于当前状态S，使用ϵ−贪婪法按一定概率选择动作A，然后得到奖励R，并更新进入新状态S′，基于状态S′，直接使用贪婪法从所有的动作中选择最优的A′（离线选择，不使用同样的ϵ−贪婪），即：

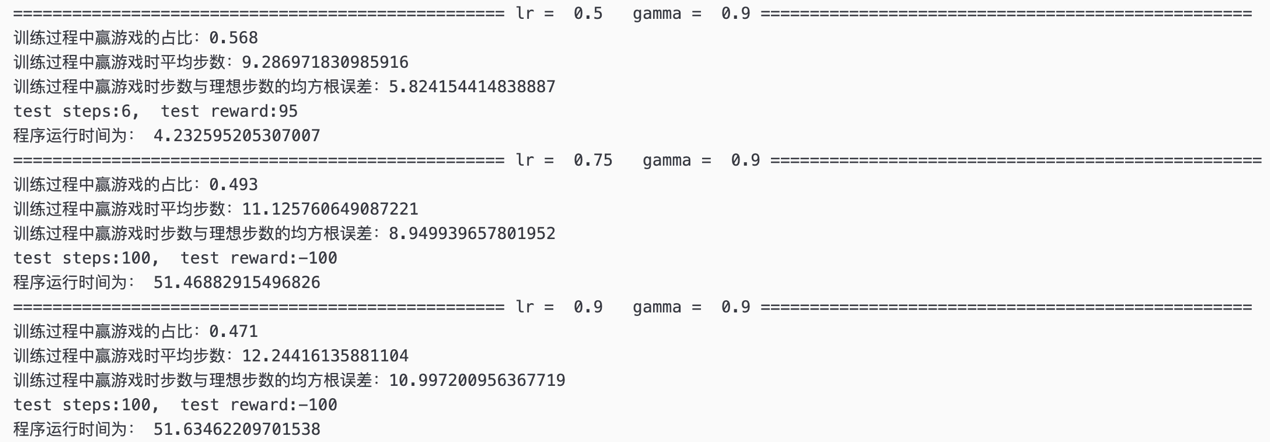
为了更加有效地训练，将reward重新设置，如果进洞则为，如果win则为100，每次行动的reward设置为。Baseline设置为：，采用控制变量分别测试其性能，即只改变学习率和折扣因子。使用三个性能指标评估各自性能：

1. 训练过程中赢得游戏的次数占比（Win Percentage）；
2. 训练过程中赢得游戏时候的平均步数；
3. 训练过程中赢得游戏时候的步数与理想步数的均方根误差

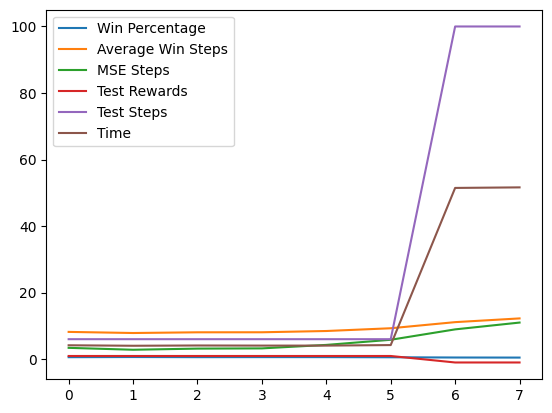
画图的时候对test\_reward进行了一些比例上调整，使得图像更加直观。

1. ﻿编程实现 Sarsa 算法实现 Agent 穿越冰湖，分析不同学习率和折扣因子下算法的表现；

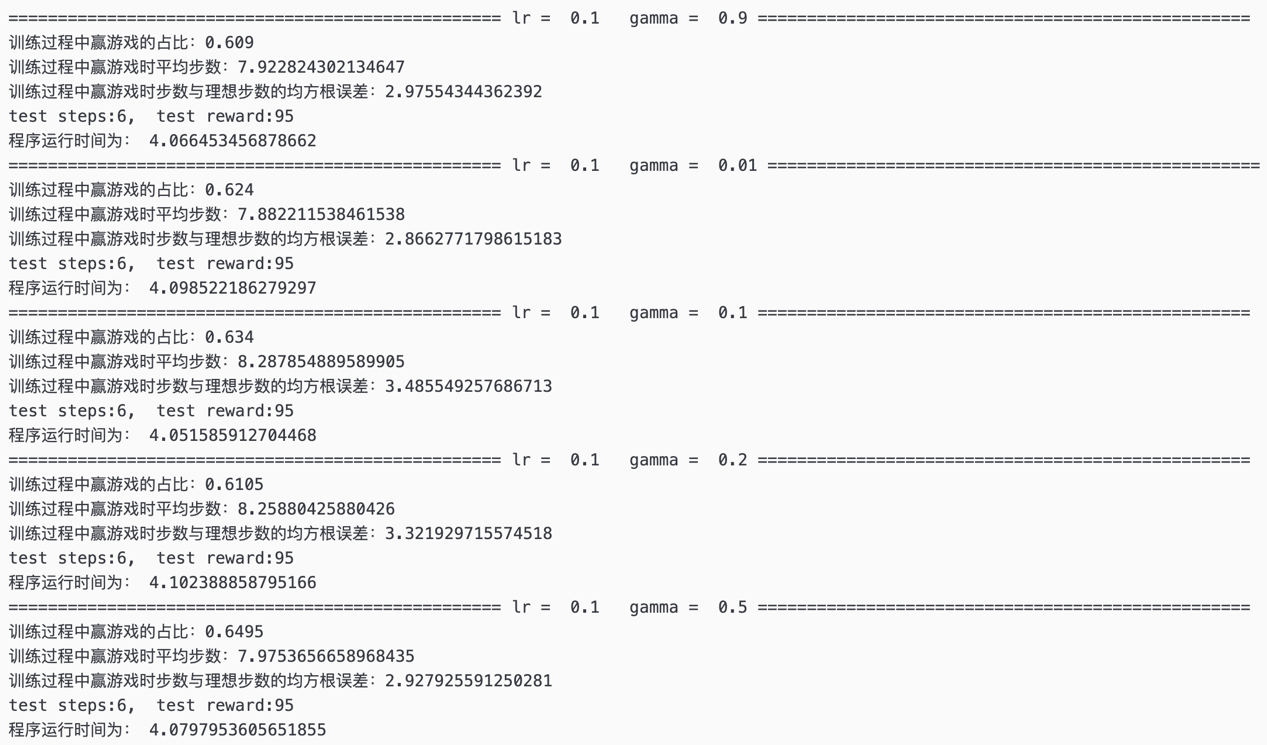
1.1 首先，固定 ，比较不同的值对性能的影响：

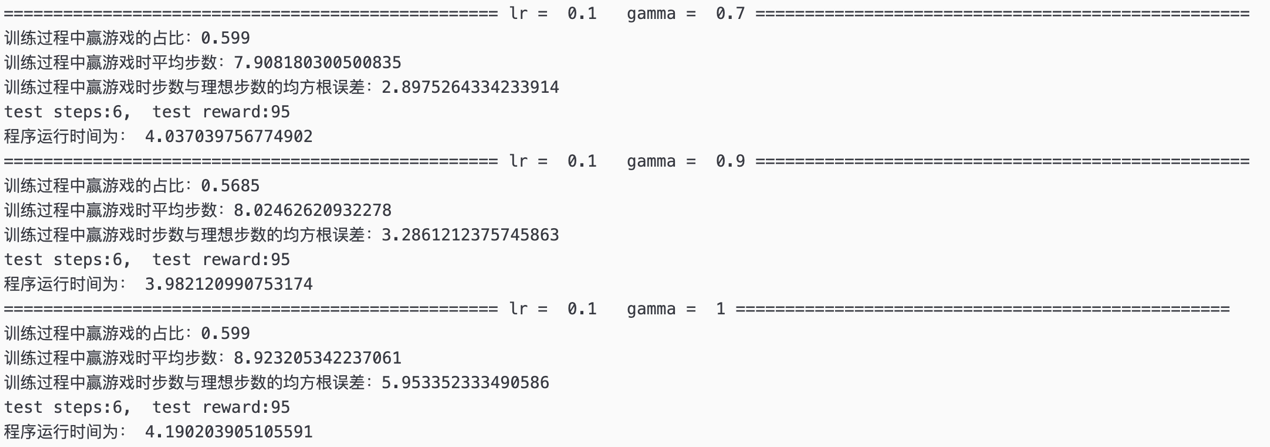


可以看出，随着的增大，SARSA算法变得更加不稳定，平均步数和均方根误差均有增大，赢游戏概率有下降趋势，甚至会出现游戏失败的情况，这是因为学习率增大有可能导致欠学习的情况；在程序运行时间上差别不大。

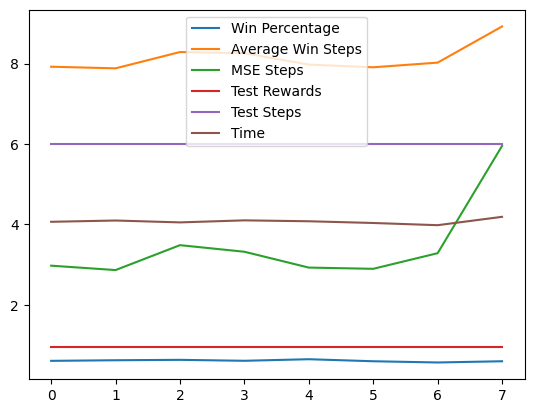


1.2 固定 ，比较不同的值对性能的影响：

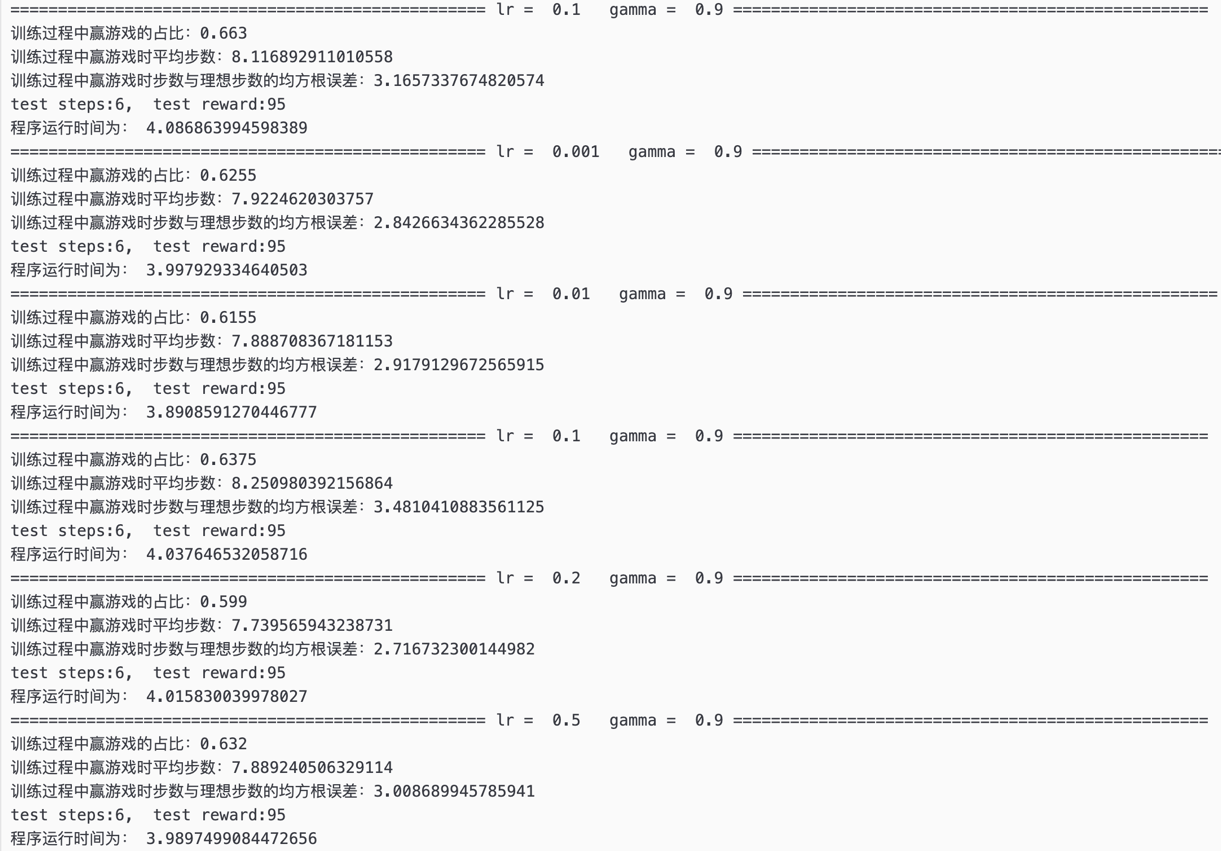


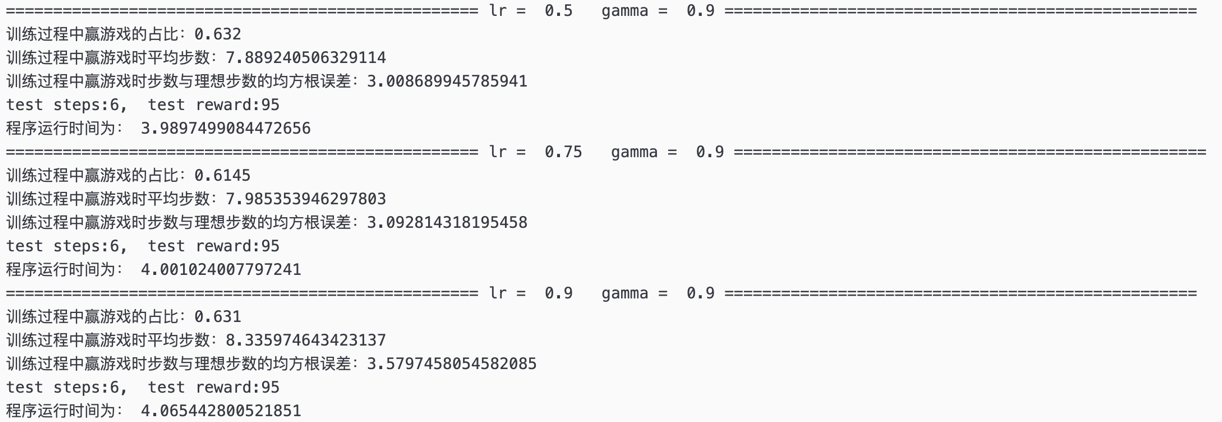


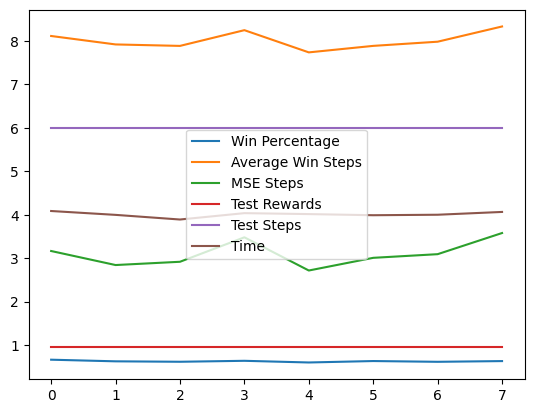
从结果看出，折扣因子对于模型性能影响并不大。但是如果折扣因子太小或者太大，模型可能会出现训练不稳定的情况，收敛步数的方差可能会较大。



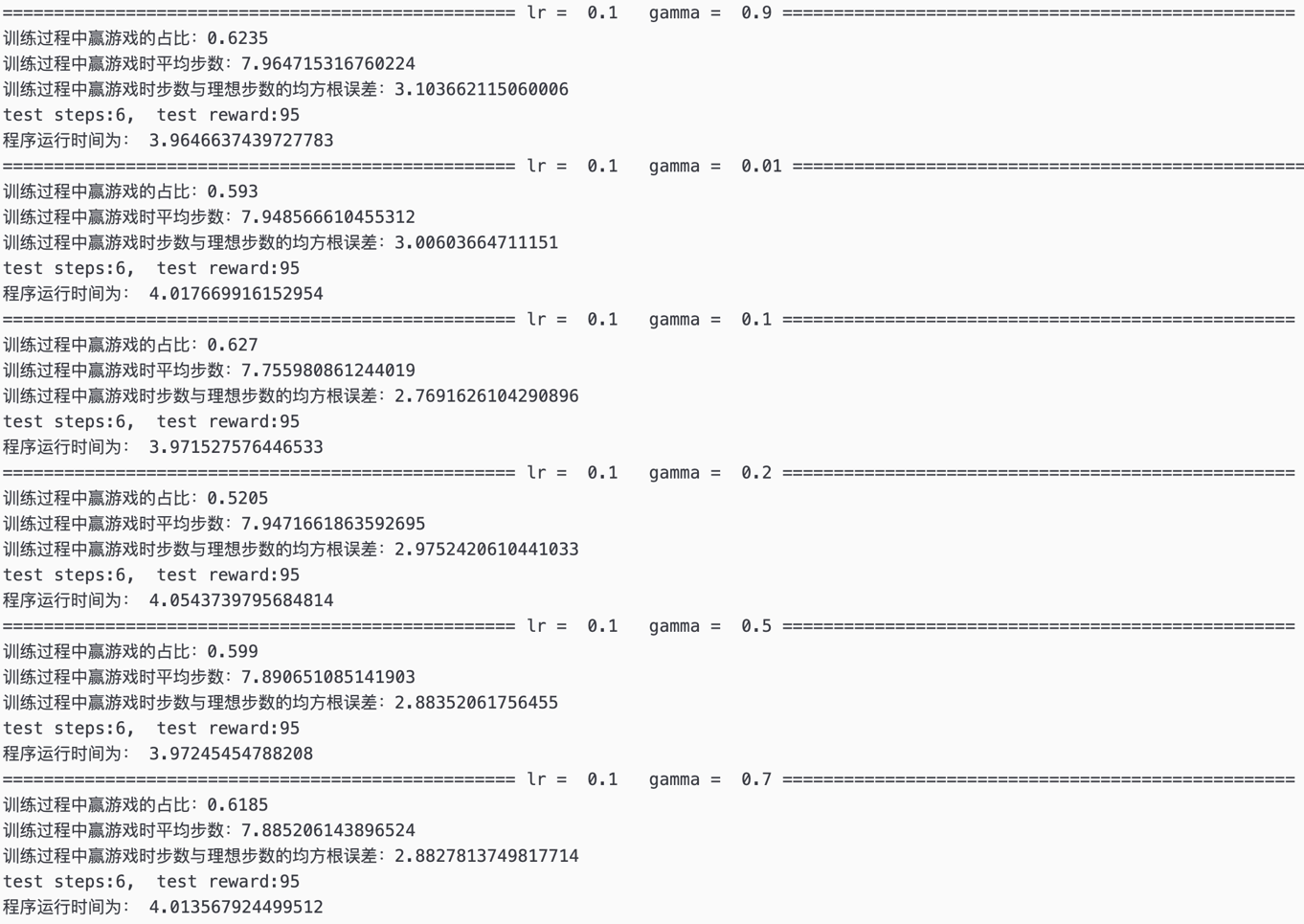
上述结果可以看出来，会是比较好的参数选择。

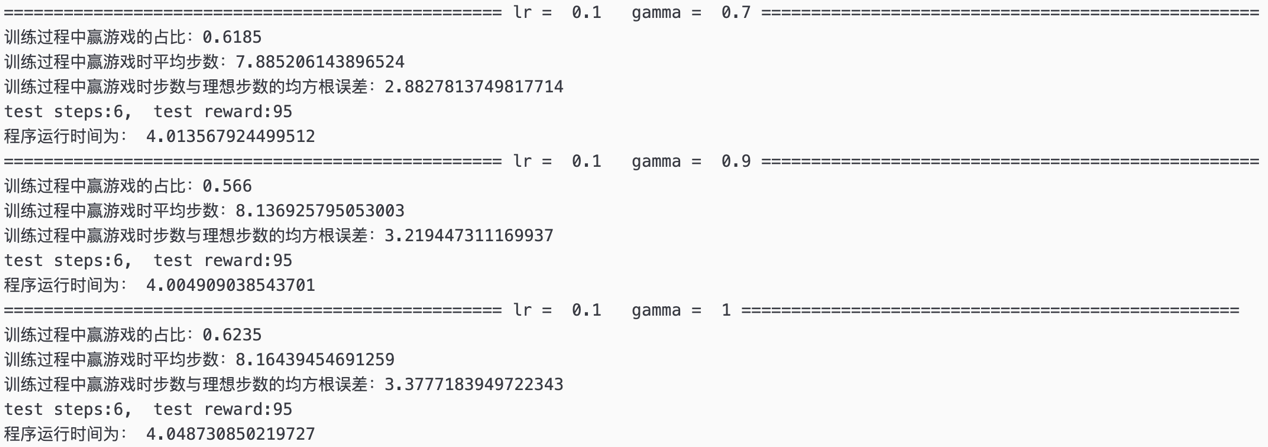
1. 编程实现Qlearning算法实现Agent穿越冰湖分析不同学习率和折扣因子下算法的表现；

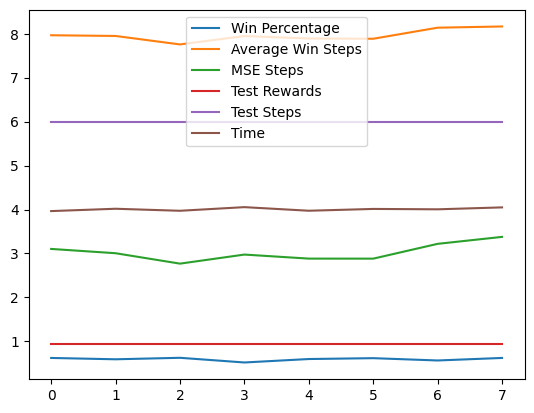




QLearning算法会更加稳定一些，的改变对整体性能影响并不算很大。但lr不能过大，不然会出现模型不稳定的趋势。

固定 ，比较不同的值对性能的影响:

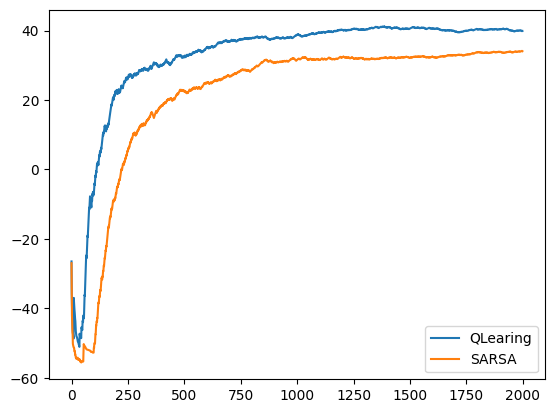




的值对模型性能影响也不是很大。但如果设置过小或过大的话，同样可能会有模型不太稳定、收敛慢的问题。

1. ﻿分析并实验对比上述两个算法的性能表现

得到其每个轮次前的平均回报图像：



从上述分析可以看出，QLearning算法相较于SARSA更加稳定、收敛更快。本质上是因为Q-Learning的Target Policy是绝对的greedy策略，绝对的greedy策略保证了Agent在Q值进入收敛后不会也不可能再记录可能失败的状态动作的Q值，也可以说是Off-Policy类的控制方法并不会受到greedy策略无探索性的影响，所以才能够产生Optimal Policy。然而SARSA算法使用的epsilon-greedy策略来更新Q值，也就是说不论何时，Agent总是有epsilon的概率选择非最优的动作，其中失败的动作也始终有一定的概率被选中，并在Q函数更新时被记录下来。