强化学习期中

项目地址：https://github.com/Light-of-Hers/rl-midterm

小组成员及分工

- 陈仁泽(1700012774)：rule-based-agent的设计与实现。
- 谭钧尹(1800012956)：rl-based-agent的设计与实现。
- 冯睿杰：调研工作及部分rl-based-agent的设计。

使用方式

```
usage: run_agent.py [-h] [-m MAP] [-r] [-e EPISODES]

optional arguments:
  -h, --help            show this help message and exit
  -m MAP, --map MAP     map name
  -r, --rule_based      whether using rule-based agent
  -e EPISODES, --episodes EPISODES
                        max episodes
```

例如用rule-based-agent打MoveToBeacon：

```
python3 ./run_agent.py -m MoveToBeacon -r -e 10
```

用rl-based-agent打FindAndDefeatZerglings：

```
python3 ./run_agent.py -m FindAndDefeatZerglings
```

注：rl-based-agent暂时只实现了FindAndDefeatZerglings, DefeatRoaches, DefeatZerglingsAndBanelings。rl-based-agent的已训练的参数保存在项目根目录下的`q_table_{3,4,5}.txt`中。

Rule Based Agent

组件设计

`Rule`

`Rule`类，表示执行用的规则，主要由两个部分组成：

- condition：一个`Observation -> Bool`的函数，用于判断前置条件。
- actions：一个元素类型为`Observation -> Action`的列表，表示动作序列。

`RuledAgent`

Rule-based Agent的基类，主要功能是：

- 设定一个`rules`列表，表示该agent基于的规则。
- 每个step开始时，检查动作队列：
    - 若空，顺序检查每个rule的condition，选取第一个匹配到的，将其actions设为动作队列。
    - 否则，从队列头取出一个动作执行。

策略简述

MoveToBeacon

- 策略较为简单，在每个step移动到最近的beacon即可。

CollectMineralShards

- 轮流选择两个枪兵(marine)，移动到离其最近的shard（同时记录前一个枪兵的目标shard，避免两者目标重复）。

FindAndDefeatZerglings

- 虽然该minigame有考察move camera相关的操作，但实际上完全基于minimap进行操作即可。
- 在minimap上设置若干个巡逻点（当前设置为靠近地图边角的四个点），组成一个可以让视野覆盖整个地图的巡逻路线。
- 初始时全选己方的三个枪兵。
- 若当前视野内存在小狗(zergling)，则攻击最近的一个。
- 若当前视野内已经没有小狗，则按设定的巡逻路线巡逻。

DefeatRoaches

- 初始全选己方枪兵。
- 每个step选择血量最少的蟑螂(roach)进行攻击（若血量相等，则选择位置最靠上的攻击）。

DefeatZerglingsAndBanelings

- 若毒爆虫(baneling)没被清完，则每个step仅派出一个枪兵去试图攻击小狗，以吸引毒爆虫自爆。
- 毒爆虫清完后，全选己方枪兵，每个step选择血量最少的小狗攻击。

CollectMineralsAndGas

- 初始时派农民(SCV)采矿，并训练农民。
- 资源足够后，在靠近右侧矿点的位置再建一个基地(command-center)。
- 然后开始训练更多的农民采矿，人口不够了就造补给站(supply-depot)。
    - 注：好像并不需要造补给站就可以吊打论文中的RL-Agent了。

BuildMarines

- 硬编码若干个点作为补给站建造点（当前设了20个）、若干个点作为兵营(barracks)建造点（当前设了8个）。
- 训练农民采矿（当前设置农民上限为20个）。
- 矿够了就建兵营。
- 兵营建到7~8个就可不再建了，开始爆兵。
- 期间人口不够了就造补给站。

效果对比

	Mean, Max (Rule-based)	Mean, Max (Worst in paper)	Mean, Max (Best in paper)
MoveToBeacon	26.34, 32	25, 33	26, 45
CollectMineralShards	110.54, 126	96, 131	104, 137
FindAndDefeatZerglings	46.68, 52	45, 56	49, 59
DefeatRoaches	100.38, 355	98, 373	101, 351
DefeatZerglingsAndBanelings	114.53, 220	62, 251	96, 444
CollectMineralsAndGas	4839.51, 5045	3351, 3995	3978, 4130

BuildMarines	133.0, 139 Mean, Max	< 1, 20 Mean, Max	6, 62 Mean, Max
	(Rule-based)	(Worst in paper)	(Best in paper)

注：BuildMarines由于reset后采光的矿不会恢复的BUG，只跑了3个episodes，其他都跑了300个episodes。

RL Based Agent

方法选择

基于打表的q_learning

训练中用时序差分TD来更新q_table：

```
q_table[prev_state][action] = 0.5 * q_table[prev_state][action] + 0.5 *
(reward + 0.99 * max_q_next)
```

训练和测试时的动作选择都采用epsilon-greedy.

```
epsilon -> action = np.argmax(q_table[curr_state][:])
1-epsilon -> action = np.random.randint(0, num_action)
```

具体设计

FindAndDefeatZerglings

- states x actions = 4 x 2
    - state_1：视野内敌人的数量，数据分箱[-inf, 1, 2, 3, inf]
    - action_1：攻击最近的敌人
    - action_2：沿给定路线巡逻
- reward：直接用pysc2库给出的obs.reward，获取每一步的即时收益

DefeatRoaches

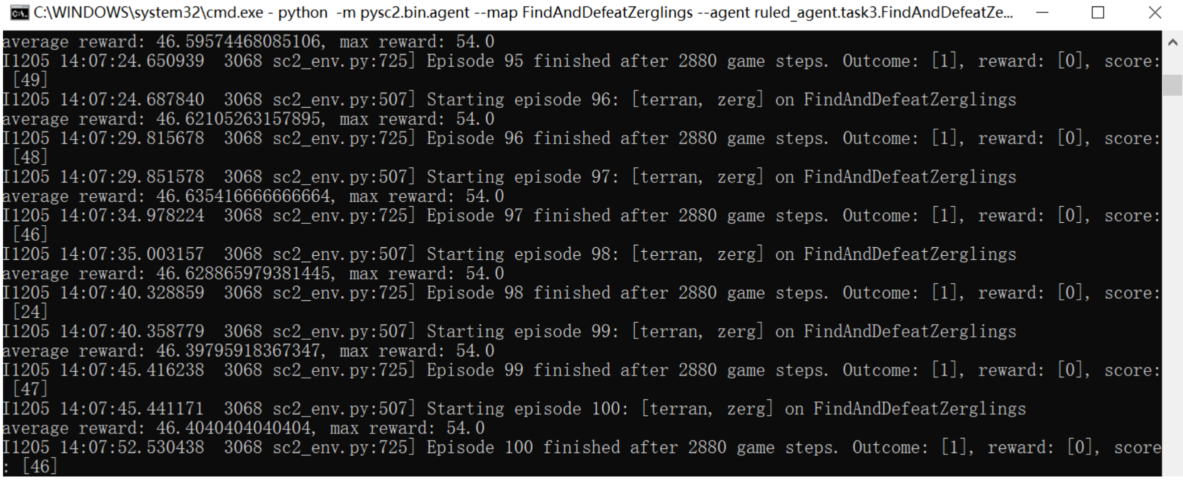
- states x actions = 4**4 x 4
    - state：四只蟑螂的血量，数据分箱[-inf, 37.5, 75, 112.5, inf]
    - action：将蟑螂按照血量排序，四种攻击分别对应攻击血量由低到高的蟑螂
- reward：直接用pysc2库给出的obs.reward，获取每一步的即时收益

DefeatZerglingsAndBanelings

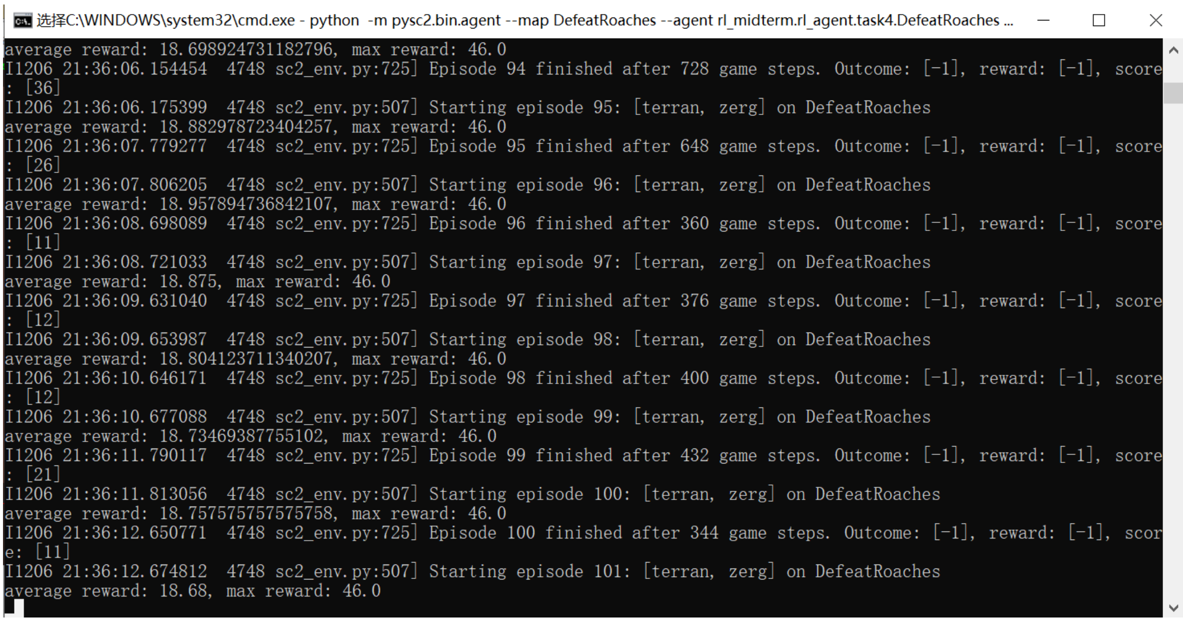
- states x actions = 4**2 x 4
    - state：小狗和毒爆虫各自的数量，数据分箱 小狗：[-inf, 1.5, 3, 4.5, inf] 毒爆虫：[-inf, 1, 2, 3, inf]
    - action：两种攻击方式：指定单体进攻或指定群体进攻，分别对应marine_attack和army_attack
- reward：直接用pysc2库给出的obs.reward，获取每一步的即时收益

效果对比

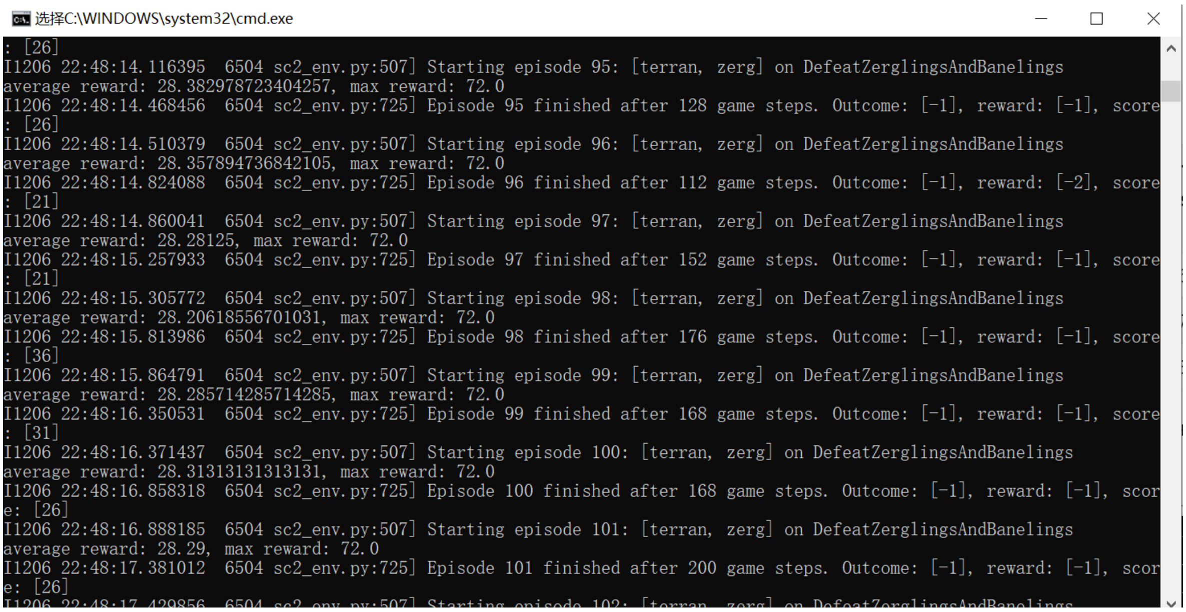
	Mean, Max (RL-based)	Mean, Max (Rule-based)	Mean, Max (Worst in paper)	Mean, Max (Best in paper)
FindAndDefeatZerglings	46.4，54	46.68, 52	45, 56	49, 59
DefeatRoaches	18.8，46	100.38, 355	98, 373	101, 351
DefeatZerglingsAndBanelings	28，72	114.53, 220	62, 251	96, 444


C:\WINDOWS\system32\cmd.exe - python -m pysc2.bin.agent --map FindAndDefeatZerglings --agent ruled_agent.task3.FindAndDefeatZe...

```
average reward: 46.59574468085106, max reward: 54.0
I1205 14:07:24.650939  3068 sc2_env.py:725] Episode 95 finished after 2880 game steps. Outcome: [1], reward: [0], score:
 [49]
I1205 14:07:24.687840  3068 sc2_env.py:507] Starting episode 96: [terran, zerg] on FindAndDefeatZerglings
average reward: 46.62105263157895, max reward: 54.0
I1205 14:07:29.815678  3068 sc2_env.py:725] Episode 96 finished after 2880 game steps. Outcome: [1], reward: [0], score:
 [48]
I1205 14:07:29.851578  3068 sc2_env.py:507] Starting episode 97: [terran, zerg] on FindAndDefeatZerglings
average reward: 46.635416666666664, max reward: 54.0
I1205 14:07:34.978224  3068 sc2_env.py:725] Episode 97 finished after 2880 game steps. Outcome: [1], reward: [0], score:
 [46]
I1205 14:07:35.003157  3068 sc2_env.py:507] Starting episode 98: [terran, zerg] on FindAndDefeatZerglings
average reward: 46.628865979381445, max reward: 54.0
I1205 14:07:40.328859  3068 sc2_env.py:725] Episode 98 finished after 2880 game steps. Outcome: [1], reward: [0], score:
 [24]
I1205 14:07:40.358779  3068 sc2_env.py:507] Starting episode 99: [terran, zerg] on FindAndDefeatZerglings
average reward: 46.39795918367347, max reward: 54.0
I1205 14:07:45.416238  3068 sc2_env.py:725] Episode 99 finished after 2880 game steps. Outcome: [1], reward: [0], score:
 [47]
I1205 14:07:45.441171  3068 sc2_env.py:507] Starting episode 100: [terran, zerg] on FindAndDefeatZerglings
average reward: 46.4040404040404, max reward: 54.0
I1205 14:07:52.530438  3068 sc2_env.py:725] Episode 100 finished after 2880 game steps. Outcome: [1], reward: [0], score
: [46]
```


选择C:\WINDOWS\system32\cmd.exe - python -m pysc2.bin.agent --map DefeatRoaches --agent rl_midterm.rl_agent.task4.DefeatRoaches ...

```
average reward: 18.698924731182796, max reward: 46.0
I1206 21:36:06.154454  4748 sc2_env.py:725] Episode 94 finished after 728 game steps. Outcome: [-1], reward: [-1], score
: [36]
I1206 21:36:06.175399  4748 sc2_env.py:507] Starting episode 95: [terran, zerg] on DefeatRoaches
average reward: 18.882978723404257, max reward: 46.0
I1206 21:36:07.779277  4748 sc2_env.py:725] Episode 95 finished after 648 game steps. Outcome: [-1], reward: [-1], score
: [26]
I1206 21:36:07.806205  4748 sc2_env.py:507] Starting episode 96: [terran, zerg] on DefeatRoaches
average reward: 18.957894736842107, max reward: 46.0
I1206 21:36:08.698089  4748 sc2_env.py:725] Episode 96 finished after 360 game steps. Outcome: [-1], reward: [-1], score
: [11]
I1206 21:36:08.721033  4748 sc2_env.py:507] Starting episode 97: [terran, zerg] on DefeatRoaches
average reward: 18.875, max reward: 46.0
I1206 21:36:09.631040  4748 sc2_env.py:725] Episode 97 finished after 376 game steps. Outcome: [-1], reward: [-1], score
: [12]
I1206 21:36:09.653987  4748 sc2_env.py:507] Starting episode 98: [terran, zerg] on DefeatRoaches
average reward: 18.804123711340207, max reward: 46.0
I1206 21:36:10.646171  4748 sc2_env.py:725] Episode 98 finished after 400 game steps. Outcome: [-1], reward: [-1], score
: [12]
I1206 21:36:10.677088  4748 sc2_env.py:507] Starting episode 99: [terran, zerg] on DefeatRoaches
average reward: 18.73469387755102, max reward: 46.0
I1206 21:36:11.790117  4748 sc2_env.py:725] Episode 99 finished after 432 game steps. Outcome: [-1], reward: [-1], score
: [21]
I1206 21:36:11.813056  4748 sc2_env.py:507] Starting episode 100: [terran, zerg] on DefeatRoaches
average reward: 18.757575757575758, max reward: 46.0
I1206 21:36:12.650771  4748 sc2_env.py:725] Episode 100 finished after 344 game steps. Outcome: [-1], reward: [-1], scor
e: [11]
I1206 21:36:12.674812  4748 sc2_env.py:507] Starting episode 101: [terran, zerg] on DefeatRoaches
average reward: 18.68, max reward: 46.0
```



```
: [26]
I1206 22:48:14.116395  6504 sc2_env.py:507] Starting episode 95: [terran, zerg] on DefeatZerglingsAndBanelings
average reward: 28.382978723404257, max reward: 72.0
I1206 22:48:14.468456  6504 sc2_env.py:725] Episode 95 finished after 128 game steps. Outcome: [-1], reward: [-1], score
: [26]
I1206 22:48:14.510379  6504 sc2_env.py:507] Starting episode 96: [terran, zerg] on DefeatZerglingsAndBanelings
average reward: 28.357894736842105, max reward: 72.0
I1206 22:48:14.824088  6504 sc2_env.py:725] Episode 96 finished after 112 game steps. Outcome: [-1], reward: [-2], score
: [21]
I1206 22:48:14.860041  6504 sc2_env.py:507] Starting episode 97: [terran, zerg] on DefeatZerglingsAndBanelings
average reward: 28.28125, max reward: 72.0
I1206 22:48:15.257933  6504 sc2_env.py:725] Episode 97 finished after 152 game steps. Outcome: [-1], reward: [-1], score
: [21]
I1206 22:48:15.305772  6504 sc2_env.py:507] Starting episode 98: [terran, zerg] on DefeatZerglingsAndBanelings
average reward: 28.20618556701031, max reward: 72.0
I1206 22:48:15.813986  6504 sc2_env.py:725] Episode 98 finished after 176 game steps. Outcome: [-1], reward: [-1], score
: [36]
I1206 22:48:15.864791  6504 sc2_env.py:507] Starting episode 99: [terran, zerg] on DefeatZerglingsAndBanelings
average reward: 28.285714285714285, max reward: 72.0
I1206 22:48:16.350531  6504 sc2_env.py:725] Episode 99 finished after 168 game steps. Outcome: [-1], reward: [-1], score
: [31]
I1206 22:48:16.371437  6504 sc2_env.py:507] Starting episode 100: [terran, zerg] on DefeatZerglingsAndBanelings
average reward: 28.31313131313131, max reward: 72.0
I1206 22:48:16.858318  6504 sc2_env.py:725] Episode 100 finished after 168 game steps. Outcome: [-1], reward: [-1], scor
e: [26]
I1206 22:48:16.888185  6504 sc2_env.py:507] Starting episode 101: [terran, zerg] on DefeatZerglingsAndBanelings
average reward: 28.29, max reward: 72.0
I1206 22:48:17.381012  6504 sc2_env.py:725] Episode 101 finished after 200 game steps. Outcome: [-1], reward: [-1], scor
e: [26]
I1206 22:48:17.429856  6504 sc2_env.py:507] Starting episode 102: [terran, zerg] on DefeatZerglingsAndBanelings
```

episode：100

效果分析

FindAndDefeatZerglings

将视野内敌人的数量作为状态，Rule-Based认为只要有敌人就都选择attack，没有敌人就scout，相当于它的策略如下：

	scout	attack
0 zerg	√	
1 zerg		√
2 zerg		√
≥3 zerg		√

RL-Based学出的是类似的效果，q_table可以反映出来策略偏好：

15.55	18.69
20.44	19.22
21.27	18.72
21.39	20.30

只是注意到过分偏离路线的Attack会导致遍历地图的速度变慢，所以测试时选择epsilon-greedy而非greedy，保留了在视野里有敌人的情况下仍然前行的可能性，这是RL_Based在Max_score上比Rule_Based表现较好的原因。

DefeatRoaches

在实验阶段我先把动作空间设为仅有attack1，即只能选择攻击血量最低的蟑螂，这样跑出来跟Rule-Based是一样的效果，这说明了代码的写法上是没有问题的。

之后加入了attack2~attack4之后，发现300个episode之后发现代码还是没有获取reward的好的策略，从q_table的数据上看它也并没有收敛的意思，所以目前的攻击方式仍然偏向于是随机地攻击，Score上反馈很差。

原因应该是在于收敛速度太慢。

DefeatZerglingsAndBanelings

我们将state设为两种敌人的数量，action设为单攻和群攻，那么根据Rule_based的策略，我们希望训练出的效果是：

	marine_attack	**army_attack**
≥1 Banel	√	x ①
0 Banel		√

但是在300个episode的训练之后，marine还是会较为频繁地做出①的选择，导致全军覆没，score无法进一步提升。

分析可能的原因：最大的原因在于状态选择不佳，zerg的数量对理想策略的生成没有帮助，而按照Banel数量来设定状态，导致reward在q_table上的分布较为稀疏，q_learning很难学出来。

我们有探讨过用敌人血量作为状态的可能性，但普遍认为不适用。好的状态还需要进一步探索。