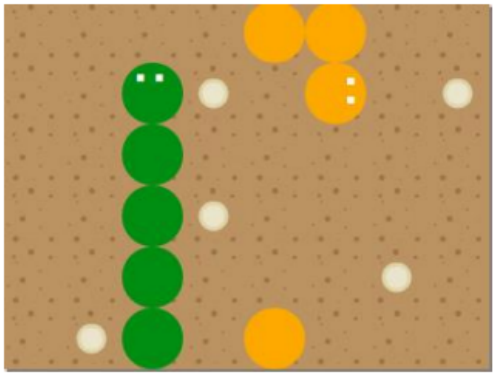
RLChina第五次习题课参考教程

芜湖~~ 最后一次习题课了喔~

贪吃蛇1v1——同构多智能体纯竞争环境

- 及第擂台http://www.jidiai.cn/compete_detail?compete=7



- 双方分别控制一条蛇，在规定步数（50）内通过吃豆子增加长度
- 若一条蛇头撞上自己或另一条蛇的蛇身会死亡，并随机以长度3重生
- 最终在第50步时，蛇身长度更长的一方获胜

题目要求:

作业的目的: 竞赛

赛程赛制

- 赛程安排
  - 热身赛1提交截止：8月23日（周一）23:00
  - 热身赛2提交截止：8月25日（周三）23:00
  - 正赛提交截止：　8月27日（周五）23:00

作业要求

- 报名本次擂台
- 实现并训练贪吃蛇(1v1)游戏的多智体对抗算法
- 将算法、模型等提交到及第擂台
- 参与最终正赛，排名高于Jidi_random

动手：

1 报名

> [竞赛详情 - 及第 Jidi (jidiai.cn)] (http://www.jidiai.cn/compete_detail?compete=7)

2 更新最新代码到本地

> [SummerCourse2021: RLChina2021] (https://gitee.com/jidiai/summercourse2021/tree/main/course_competition)

3 训练

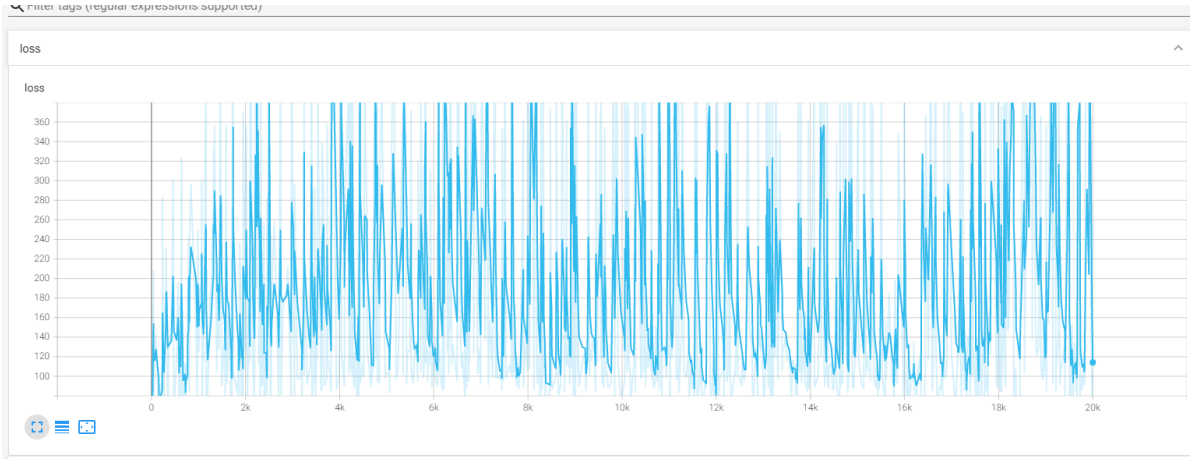
默认是对自己的agent采用DQN, 对手采用random 算法，进行训练

这版代码和snakes_3v3基本一致，可以对比的参考。

训练直接运行main.py即可。这里打印部分结果

```
[Episode 19994] score: 6.0 reward: 100.00
                loss 74.028
[Episode 19995] score: 1.0 reward: -20.00
                loss 120.191
[Episode 19996] score: 16.0 reward: 100.00
                loss 141.101
[Episode 19997] score: 3.0 reward: 100.00
                loss 103.523
[Episode 19998] score: 3.0 reward: 100.00
                loss 88.793
[Episode 19999] score: 4.0 reward: 100.00
                loss 522.823
[Episode 20000] score: 1.0 reward: 100.00
                loss 90.312
```

可视化一下训练结果：
loss:

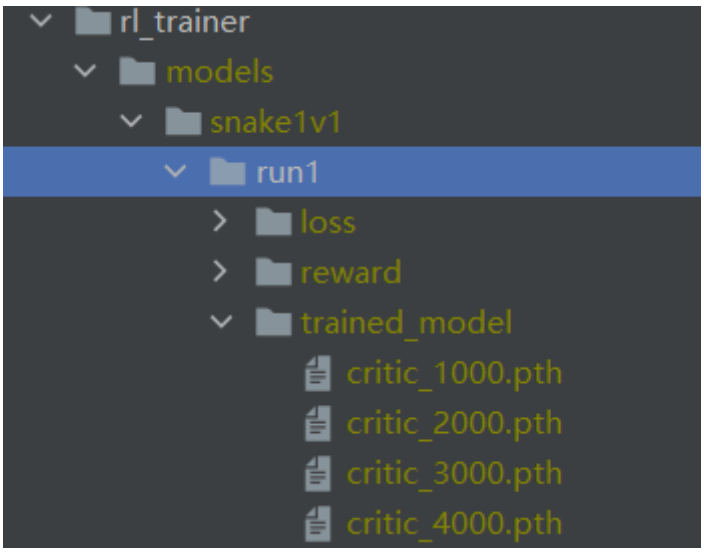


rewards:

从打印结果可以看到rewards的设计比较粗糙。

从图形化的score的结果看学习的效果并不是很显著

模型训练结果如下：



4 提交

这次的submission和`snakes_2p,snakes_3v3`很像，把训练好的模型加载进来即可。

这次最终的模型感觉也还凑合，直接提交训练20000轮的结果。

submission函数的编写，最后一次开了直播大概讲解了一下。直播内容如下

1. 第五次作业submission文件编写
    - 状态
    - 观测
    - Critic
    - 动作
2. 分享一下我理解的训练代码逻辑
3. 分享逻辑的时候用debug的方式交流一些我使用pycharm常用的功能技巧
4. 根据3V3前八大佬们的经验分享，聊一聊本次作业可以使用的优化方式
    - 优化观测
    - 优化奖励设置
    - 掩码掉不良动作
    - 调整网络结构，及其他超参数
    - 更改算法(eg: PPO等)
5. 然后大家一起唠唠嗑，交流下问题啥的

直播上传链接：

> https://www.bilibili.com/video/BV1ZQ4y1y7qT?share_source=copy_web

第五次作业 Submission

```python
import torch.nn as nn
import torch.nn.functional as F
import torch
import os
# ====================================== helper functions ======================================
from pathlib import Path
import sys
base_dir = Path(__file__).resolve().parent
sys.path.append(str(base_dir))
from common import make_grid_map, get_surrounding, get_observations


# ====================================== define algo ==========================================
# todo
class Critic(nn.Module):
    def __init__(self, input_size, output_size, hidden_size):
        super().__init__()
        self.input_size = input_size
        self.output_size = output_size
        self.linear1 = nn.Linear(input_size, hidden_size)
        self.linear2 = nn.Linear(hidden_size, output_size)

    def forward(self, x):
        x = F.relu(self.linear1(x))
        x = self.linear2(x)
        return x


# todo
class DQN(object):
    def __init__(self):
        # pass
        self.state_dim = 18
        self.action_dim = 4
        self.hidden_size = 256

        self.critic_eval = Critic(self.state_dim, self.action_dim, self.hidden_size)
    def choose_action(self, observation):
        # pass
        observation = torch.tensor(observation, dtype=torch.float).view(1, -1)
        action = torch.argmax(self.critic_eval(observation)).item()

        return action

    def load(self, file):
        # pass
        base_path = os.path.dirname(os.path.abspath(__file__))
```

```
        file = os.path.join(base_path, file)
        self.critic_eval.load_state_dict(torch.load(file))


def to_joint_action(actions, num_agent):
    joint_action = []
    for i in range(num_agent):
        action = actions
        one_hot_action = [0] * 4
        one_hot_action[action] = 1
        joint_action.append(one_hot_action)
    return joint_action


# ==================================== define agent ============================================
#todo
agent = DQN()
agent.load('critic_20000.pth')


#
================================================================================================
"""
input:
    observation: dict
    {
        1: 豆子,
        2: 第一条蛇的位置,
        3: 第二条蛇的位置,
        "board_width": 地图的宽,
        "board_height": 地图的高,
        "last_direction": 上一步各个蛇的方向,
        "controlled_snake_index": 当前你控制的蛇的序号（2或3）
        }
return:
    action: eg. [[[0,0,0,1]]]
"""
# todo
def my_controller(observation, action_space_list, is_act_continuous):
    # pass
    obs = get_observations(observation,0,18)
    action = agent.choose_action(obs)
    action_ = to_joint_action(action,1)

    # [0,0,0,1]
    return action_
```

OK ,That's all

作者：HandsomeWu(公众号同步)

也欢迎关注公众号：**RLCN** 在后台提问：

相关文章及资源:

网盘链接：永久有效 -- 或者也可以从RLChina官网获取最新版(http://rlchina.org/)

> 链接：https://pan.baidu.com/s/1EIaXGSjgn3Em1wi8Zkc_jw
> 提取码：x87v dd

RLChina第一次习题课参考教程

RLChina第二次习题课参考教程

RLChina第三次习题课参考教程

RLChina第四次习题课参考教程

RLChina2021-习题课5 -- 林舒 中科院自动化研究所