强化学习：作业一

庄镇华 502022370071

2022 年 10 月 23 日

1 作业内容

在“蒙特祖马的复仇”环境中实现 Dagger 算法。

2 实现过程

2.1 相关改进

原始的 Dagger 算法初始时模型参数随机，这会导致开始时模型犯很多不必要的错误，因此本文先用专家标注数据进行有监督预训练，然后再用 Dagger 算法进行进一步训练。

2.2 主要过程

本次实验最终采用的模型是 K 近邻分类模型（尝试神经网络模型，发现数据量小的时候模型欠拟合无法达到效果）。模型的主要训练流程如下：

1. 使用预训练的参数初始化模型

2. 针对每一轮：使用当前模型采样若干轨迹存储到样本池，专家查看数据并给出标签，利用样本池中的样本和专家给出的标签对当前的模型进行更新。

3. 如此重复训练，直至模型收敛

细节部分。如对图像进行预处理，即裁剪大小并转为灰度图，另外，实验过程中发现幽灵对模型决策干扰很大，因此利用黑色掩盖掉幽灵的存在；

Initialize $\mathcal{D} \leftarrow \emptyset$.
Initialize $\hat{\pi}_1$ to any policy in Π.
for $i = 1$ **to** N **do**
  Let $\pi_i = \beta_i \pi^* + (1 - \beta_i)\hat{\pi}_i$.
  Sample T-step trajectories using π_i.
  Get dataset $\mathcal{D}_i = \{(s, \pi^*(s))\}$ of visited states by π_i and actions given by expert.
  Aggregate datasets: $\mathcal{D} \leftarrow \mathcal{D} \bigcup \mathcal{D}_i$.
  Train classifier $\hat{\pi}_{i+1}$ on $\mathcal{D}$.
end for
Return best $\hat{\pi}_i$ on validation.

图 1: Dagger 算法

此外，为了减少无效的样本标注量，将训练时每轮的步数改为 200，测试步数改为 400；最后，动作空间过大也会导致模型效果不好，因此只保留了 8 个有用的动作。

```python
def pre_process(ob, size = (128, 128)):
    obs = ob.copy()
    obs[obs == 236] = 0 # 去掉幽灵的影响
    obs_ = cv2.cvtColor(obs, cv2.COLOR_BGR2GRAY)
    obs_ = cv2.resize(obs_, size)
    obs_ = obs_.reshape((1, -1))
    return obs_
```

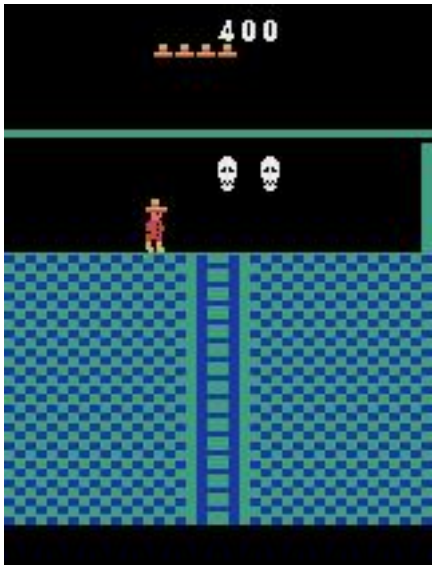
图 2: 图像预处理

3 复现方式

首先需要下载 python-opencv 库，测试模式直接运行 python main.py 命令即可复现最好结果，打开 imgs 下的 label.png 图片即可看到模型每一步的动作与结果。

训练模式需要取消 main.py 中对于询问专家标签和模型更新的注释，然后再运行 python main.py 命令即可进行模型训练。

拿到钥匙



通过墙壁

图 3: 模型效果

4 实验效果

实验效果可以查看文件夹下的短视频，模型可以拿到钥匙并到达第二个房间，即最高得分为 400 分，见图 3，累计奖励、访问专家次数和样本训练量之间的关系可见图 4。

可以看到，访问专家次数和样本训练量之间呈正比例关系，并且随着样本训练量和访问专家次数的增加，累计奖励也在不断增加，由于模型每阶段仅测试一次，因此曲线图有一些波动。

5 思考题

在玩游戏的过程中标注数据与 Dagger 算法中的标注数据方式有何不同？这个不同会带来哪些影响？

玩游戏的过程中标注数据是人标注完数据后，用这些已经被标注的数据进行训练，并且训练过程中不再新增标注数据；而 Dagger 算法中标注数据是训练过程中标注数据，即每轮都会新增标注数据，在探索和利用的过程中标注数据。

带来的影响是 Dagger 算法更容易收敛，因为模型经过训练后采样的数据分布也会发生变化，现有的数据无法覆盖新的数据，因此必须针对新数据进行专家标注并训练，仅仅利用离线的数据会导致错误累积，模型发散，这也是离线强化学习的痛点所在。

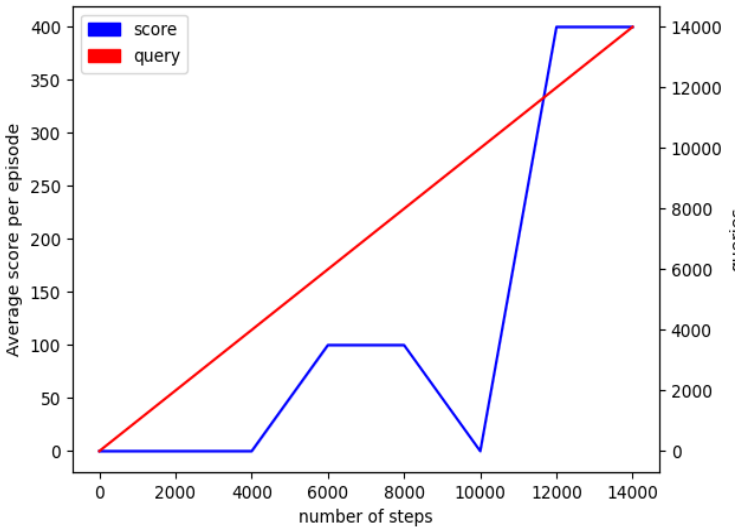


图 4: Dagger 算法

6 小结

在这次实验中，我发现传统的监督学习不适用于强化学习任务，因为模型经过训练后采样的数据分布也会发生变化，现有的数据和新的数据不满足独立同分布的要求，而 Dagger 算法的提出缓解了这一问题，但过高的专家标注量也是另一个问题。

通过亲手实验，我体会到了 Dagger 算法的优点与可行之处，同时也希望更深入地了解模仿学习的其他先进算法来解决 Dagger 算法的问题。