强化学习：作业三

傅浩敏 MG20370012

2020年12月28日

1 作业内容

本实验需要在gridworld环境中实现model-based Q-learning算法并评估算法性能。本次实验由三部分组成，首先我们实现Dyna-Q算法，并且调整超参数以尽可能达到算法可提升的性能极限。第二部分是使用神经网络来预测环境model，同样通过调整超参数来获得最佳的性能表现。第三部分是改进model学习过程以应对稀疏奖励的问题，并且分析这些改进对算法带来了哪些变化。最后需要分析不同的学习方式和超参数对model-based算法性能的影响，并且讨论产生这些影响的可能的原因。

2 实现过程

2.1 算法描述

首先需要在`algo.py`中实现Q-learning Agent。在本实验中我直接使用HW2中实现过的Q-learning算法，我以表格的形式记录Q值并且动态地向Q表中添加新的状态-动作对。

对于普通的Dyna-Q算法，我也采用表格的形式记录状态-动作-新状态之间的转移关系。每当在环境中采样后，需要将采样到的转移关系存在转移表中。我们需要同时使用采样到的转移和转移表中储存的转移来更新Q表。

在第二个实验中，我们使用神经网络来预测状态-动作-新状态之间的转移关系，在环境中采样后都要把采样到的转移关系存储在buffer中。每经过若干轮迭代，我们从buffer中抽取若干样本来训练我们的model。同样的，我们需要同时使用采样到的转移关系和神经网络模型预测的转移关系来更新Q表。

在第三个实验中，我们对model的学习过程进行了一些改进。首先，我们在采样的同时记录了那些获得钥匙的关键转移。在训练神经网络模型时，固定这些样本在训练样本中的比例，从而保证模型始终能够对“是否获

得钥匙”做出正确的预测。第二个改进是在Q表中的值超过某个定义的上下界时进行截断。

2.2 代码实现

Algorithm 1 Dyna-Q

Initial Q_π and $Model$. Given reword function R and terminal function D. Given learning rate α, discount rate γ, and repeat number n.
for $1, 2, \cdots$ **do**
  $s = env.reset()$
  for $1, \cdots, steps$ **do**
    epsilon-greedy: $a = \pi_\epsilon(s)$
    interact with environment: $s', r, done = env.step(a)$
    update Q: $Q_\pi(s,a) += \alpha(r + \gamma(1 - done)\max_{a'} Q_\pi(s', a') - Q_\pi(s,a))$
    update Model: $Model(s,a) \leftarrow r, s'$
    $s = s'$
    if $done$ **then**
      $s = env.reset()$
    end if
  end for
  for $1, 2, \cdots, n$ **do**
    s_m, a_m, s'_m = random sample from ovserved transitions
    get reword: $r_m = R(s_m, a_m, s'_m)$
    get done: $d_m = D(s_m, a_m, s'_m)$
    update Q: $Q_\pi(s_m, a_m) += \alpha(r_m + \gamma(1 - d_m)\max_{a'_m} Q_\pi(s'_m, a'_m) - Q_\pi(s_m, a_m))$
  end for
end for

3 复现方式

在主文件夹下运行 `python main.py`.

3.1 参数介绍

4 实验效果

4.1 实验图表展示与分析

描述累计奖励和样本训练量之间的关系。

5 问题讨论

1. 根据上面实验，试讨论不同模型学习方式（table 和 neural network），不同参数对实验结果的影响和背后的原因，从而分析影响model-based的算法的性能的因素由哪些?

2. 回顾HW3的DQN中的replay buffer设置和前面的Dyna-Q实验，你觉得这两者有什么联系?

6 小结

6.1 关于算法本身

在这次实验中，我发现...

6.2 关于实验过程