强化学习

Reinforcement

Learning

#6 Sarsa 算法更新 (强化学习 Reinforcement Learning 教学)

切换成 优酷 视频

(Chrome无法播放优酷? 网址框输入"chrome://settings/content/", 勾选允许 Flash Player. 实在不行? 请 点击这里)

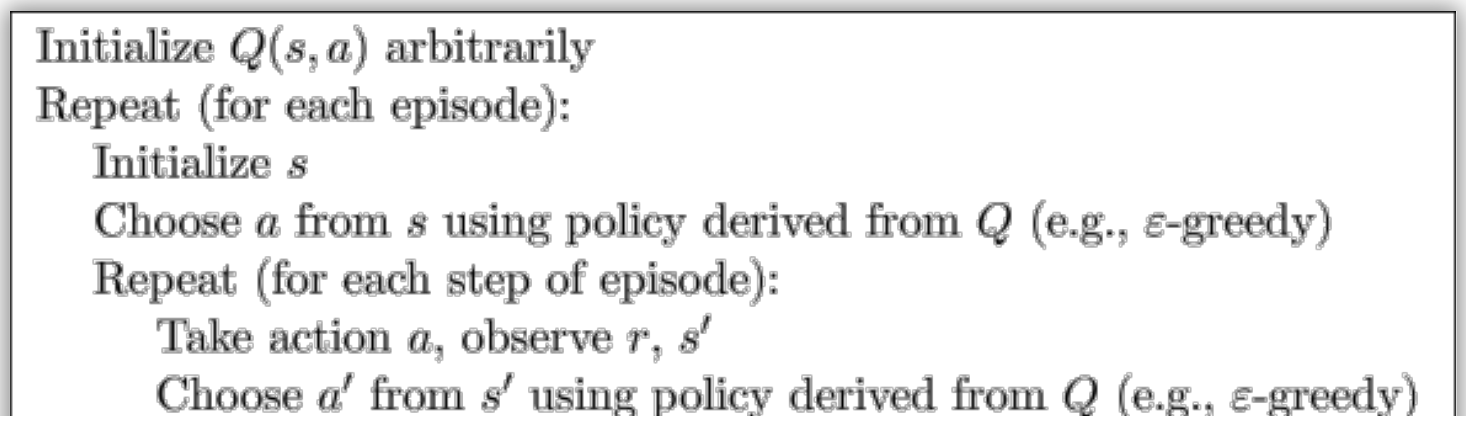
作者: **Morvan** 编辑: **Morvan**

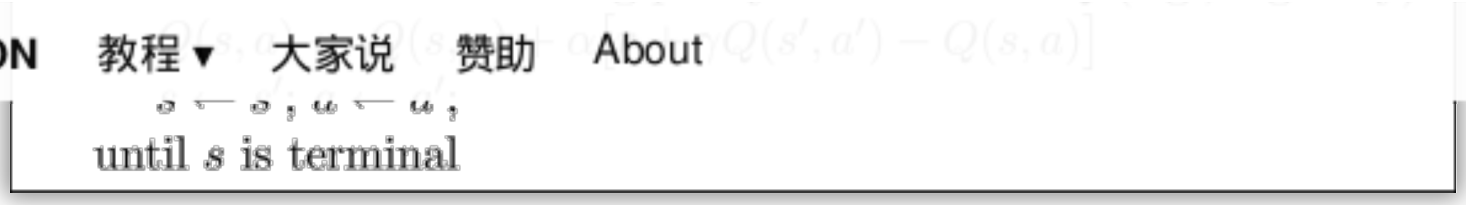
- 学习资料:
  - 全部代码
  - 什么是 Sarsa 短视频
  - 本节内容的模拟视频效果Youtube, 优酷
  - 学习书籍 Reinforcement learning: An introduction

这次我们用同样的迷宫例子来实现 RL 中另一种和 Qlearning 类似的算法, 叫做 Sarsa (state-action-reward-state_-action_). 我们从这一个简称可以了解到, Sarsa 的整个循环都将是在一个路径上, 也就是 on-policy, 下一个 state_, 和下一个 action_ 将会变成他真正采取的 action 和 state. 和 Qlearning 的不同之处就在这. Qlearning 的下个一个 state_ action_ 在算法更新的时候都还是不确定的 (off-policy). 而 Sarsa 的 state_, action_ 在这次算法更新的时候已经确定好了 (on-policy).

本节内容包括:

算法





整个算法还是一直不断更新 Q table 里的值, 然后再根据新的值来判断要在某个 state 采取怎样的 action. 不过于 Qlearning 不同之处:

- 他在当前 state 已经想好了 state 对应的 action , 而且想好了 下一个 state_ 和下一个 action_ (Qlearning 还没有想好下一个 action_)
- 更新 Q(s,a) 的时候基于的是下一个 Q(s_, a_) (Qlearning 是基于 maxQ(s_))

这种不同之处使得 Sarsa 相对于 Qlearning, 更加的胆小. 因为 Qlearning 永远都是想着 maxQ 最大化, 因为这个 maxQ 而变得贪婪, 不考虑其他非 maxQ 的结果. 我们可以理解成 Qlearning 是一种贪婪, 大胆, 勇敢的算法, 对于错误, 死亡并不在乎. 而 Sarsa 是一种保守的算法, 他在乎每一步决策, 对于错误和死亡比较铭感. 这一点我们会在可视化的部分看出他们的不同. 两种算法都有他们的好处, 比如在实际中, 你比较在乎机器的损害, 用一种保守的算法, 在训练时就能减少损坏的次数.

算法的代码形式

首先我们先 import 两个模块, maze_env 是我们的环境模块, 已经编写好了, 大家可以直接在这里下载, maze_env 模块我们可以不深入研究, 如果你对编辑环境感兴趣, 可以去看看如何使用 python 自带的简单 GUI 模块 tkinter 来编写虚拟环境. 我也有对应的教程. maze_env 就是用 tkinter 编写的. 而 RL_brain 这个模块是 RL 的大脑部分, 我们下节会讲.

```
from maze_env import Maze
from RL_brain import SarsaTable
```

下面的代码, 我们可以根据上面的图片中的算法对应起来, 这就是整个 Sarsa 最重要的迭代更新部分啦.

```
def update():
    for episode in range(100):
        # 初始化环境
        observation = env.reset()

        # Sarsa 根据 state 观测选择行为
        action = RL.choose_action(str(observation))

        while True:
            # 刷新环境
            env.render()

            # 在环境中采取行为, 获得下一个 state_ (obervation_), reward, 和是否终止
            observation_, reward, done = env.step(action)

            # 根据下一个 state (obervation_) 选取下一个 action_
            action_ = RL.choose_action(str(observation_))

            # 从 (s, a, r, s, a) 中学习, 更新 Q_tabel 的参数 ==> Sarsa
            RL.learn(str(observation), action, reward, str(observation_), action_)

            # 将下一个当成下一步的 state (observation) and action
            observation = observation_
```

```
            action = action_
            # 终止时跳出循环
            if done:
                break

        # 大循环完毕
        print('game over')
        env.destroy()

if __name__ == "__main__":
    env = Maze()
    RL = SarsaTable(actions=list(range(env.n_actions)))

    env.after(100, update)
    env.mainloop()
```

下一节我们会来讲解 SarsaTable 这种算法具体要怎么编.

如果想一次性看到全部代码, 请去我的 Github

如果你觉得这篇文章或视频对你的学习很有帮助, 请你也分享它, 让它能再次帮助到更多的需要学习的人.

莫烦没有正式的经济来源, 如果你也想支持 莫烦**Python** 并看到更好的教学内容, 请拉倒屏幕最下方, 赞助他一点点, 作为鼓励他继续开源的动力.

« 上一个 下一个 »

撰写评论

使用社交网站账户登录 或使用来必力便捷评论

邮件

写评论

总评论数 7 按时间正序

一张天舒 2017年6月2日 · 已分享的SNS(1)

莫老师,用你的Sarsa代码跑了一下,为什么最后程序根本停不下来,说明Q没有收敛,程序应该有问题吧,最后while True陷入死循环根本到不了最终状态 因为if done满足不了,麻烦你看一下

1 0 0

莫烦Python 20小时前

@一张天舒 是的, 到了最后 Sarsa 很难往下走, 基本上都停在上面了, 所以一直在 while loop 里面. 如果修改一下 epsilon 可能会好点. 可以使用一个不断递增的 epsilon 来控制探索度.

0 0 0

James 2017年4月18日

你好，似乎SARSA的算法描述和Q-Learning的一模一样？
参见：https://morvanzhou.github.io/tutorials/machine-learning/reinforcement-learning/2-1-general-rl/

1 0 0

莫烦Python 2017年4月18日

@James 不一样的， 你仔细看一下

0 0 0

NO NICKNAME 2017年1月10日

你好，这里的视频好像看不了

2 0 0

莫烦Python 2017年1月10日

@NO NICKNAME 这里的视频还没做. 文字先写好了, 等我有时间就做视频. 最近又搬家什么的, 比较忙

1 0 0

NO NICKNAME 2017年1月11日

@莫烦Python 嗯好，辛苦了

0 0 0

来必力是？ | 询问

支持 让教学变得更优秀

点我 赞助 莫烦

关注我的动向:

Youtube频道 优酷频道 Github 微博

Email: morvanzhou@hotmail.com