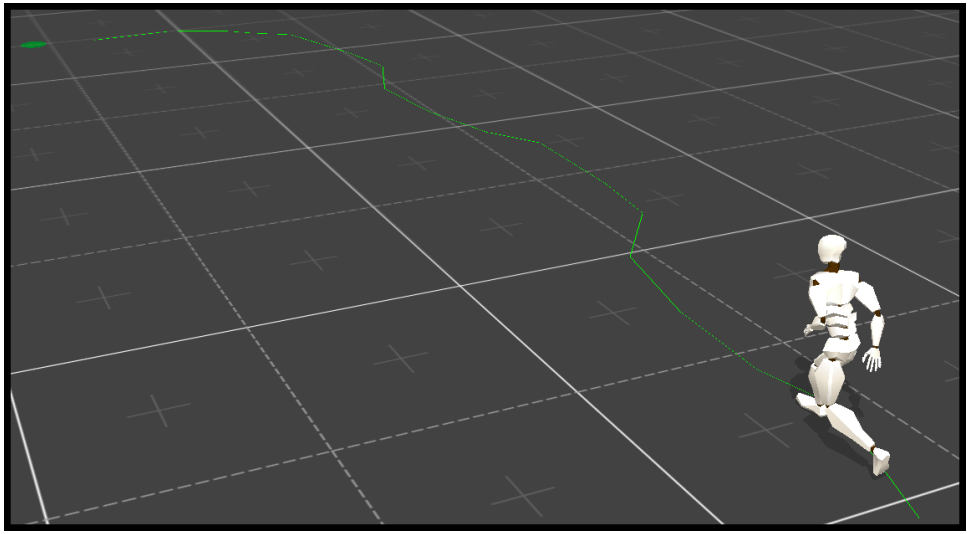
强化学习系列之八：在 OpenAI Gym 上实现 Q Learning

团队号 AlgorithmDog (/subjects/83978) 作者 @algorithmdog
原文链接 (http://www.algorithmdog.com/openai-gym-qleanring)

这周试用了下 OpenAI Gym。OpenAI Gym是一款用于研发和比较强化学习算法的工具包。强化学习和有监督学习的评测不一样。有监督学习的评测工具是数据。只要提供一批有标注的数据就能进行有监督学习的评测。强化学习的评测工具是环境。需要提供一个环境给 Agent 运行，才能评测 Agent 的策略的优劣。OpenAI Gym 是提供各种环境的开源工具包。



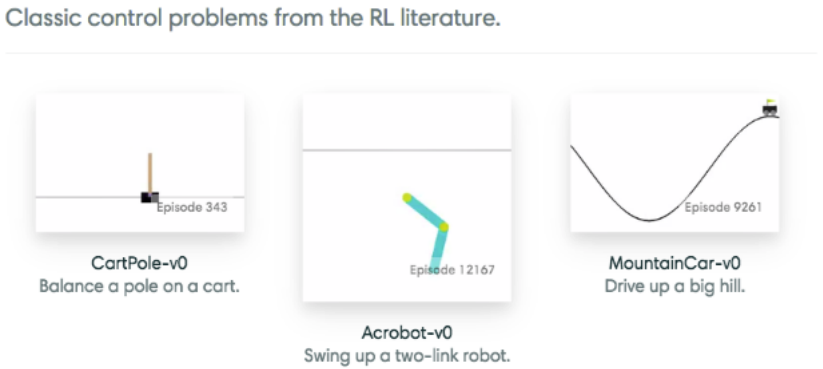
(http://7rf34y.com2.z0.glb.qiniucdn.com/c/474ab3a18b045dd671339eeb424f87b8)

1. OpenAI Gym 的基本知识

下面 OpenAI Gym 是一个示例。

```
import gym
env = gym.make('CartPole-v0') //实例化一个 CartPole 环境
for i_episode in range(20):
    observation = env.reset()
    for t in range(100):
        env.render() //更新动画
        action = env.action_space.sample()
        observation, reward, done, info = env.step(action) //推进一步
        if done:
            break
```

OpenAI Gym 的最重要的功能就是提供各种强化学习环境。上面的代码 env = gym.make('CartPole-v0') 是实例化一个 CartPole 环境。CartPole 环境要求平衡一辆车上的一根棍子，如下图的第一个环境表示。下图是 OpenAI Gym 提供的部分自动控制方面的环境。除此之外 OpenAI Gym 还提供了算法、文本和游戏方面的环境，具体可以查看官方说明 (https://gym.openai.com/envs)。



(http://7rf34y.com2.z0.glb.qiniucdn.com/c/073eddb22f49f4942e4429169a487253)

强化学习环境其实是马尔科夫决策过程，马尔科夫决策过程的四个基本元素：状态、动作、转移概率和奖励函数。

1.1 状态

代码中的 observation 就是马尔科夫决策过程的状态。更正确地说是，状态的特征。CartPole-v0 的状态特征是一维数组，比如array([-0.01377819, -0.01291427, 0.02268009, -0.0380999])。有些环境提供的状态特征是二维数组，比如 AirRaid-ram-v0 环境提供的是二维数组表示的游戏画面。

observation = env.reset() 是初始化环境，设置一个随机或者固定的初始状态。env.step(a1) 是环境接受动作 a1，返回的第一个结果是接受动作 a1 之后的状态特征。

1.2 动作

代码中的 action 就是马尔科夫决策过程中的动作。CartPole-v0 的动作是离散型特征。在 OpenAI Gym 中，离散型动作是用从 0 开始的整数集合表示，比如 CartPole-v0 的动作有 0 和 1。另一种动作是连续型，用实数表示。

1.3 转移概率和奖励函数

在 OpenAI Gym 中，转移概率并没有显式表示出来，而是通过 env.step(a1) 的结果表示。env.step(a1) 返回的 observation 满足转移概率。

代码中的 reward 就是马尔科夫决策过程中的动作，用实数表示。在 OpenAI Gym 中，奖励函数也没有显式表示出来，也是通过 env.step(a1) 的结果表示。env.step(a1) 返回的 reward 满足奖励函数。

值得一提的是，env.step 返回的第四个结果 info 是系统信息，给开发人员调试用，不允许学习过程使用。本文只介绍在 OpenAI Gym 上实现 Q Learning 算法需要的知识。想了解更多 OpenAI Gym 知识，可以参考 OpenAI Gym 官方文档 (https://gym.openai.com/docs)。

2. 实验Q Learning 算法

我们在 OpenAI Gym 的 CartPole-v0 环境上实现 Q Learning 算法 (http://www.algorithmdog.com/reinforcement-learning-model-free-learning)。Q Learning 目标是学习状态动作价值。QLearning 让 Agent 按照策略进行探索，在探索每一步都进行状态价值的更新，更新公式如下。

q(s,a)=q(s,a)+α{r+maxa′{γq(s′,a′)}−q(s,a)}

由于 OpenAI Gym 提供的状态特征，因此我们要用价值函数近似 (http://www.algorithmdog.com/reinforcement-learning-value-function-approximation)，参数更新的代码如下所示。

```
def update(policy, s_fea, a, tvalue, alpha):
    pvalue        = policy.qfunc(s_fea, a);
    error         = pvalue - tvalue;
    s_a_fea       = policy.get_state_action_fea(s_fea, a);
    policy.theta -= alpha * error * s_a_fea;
```

Q Learning 的代码如下。

```
def qlearning(env, policy, num_iter1, alpha, gamma):
    for i in xrange(len(policy.theta)):
        policy.theta[i] = 0.1

    for iter1 in xrange(num_iter1):
        s_f       = env.reset()
        a         = policy.epsilon_greedy(s_f)
        count     = 0
        t         = False

        while False == t and count < 10000:
            s_f1,r,t,i = env.step(a)
            qmax = policy.qfunc(s_f1,a) #random
            for a1 in policy.actions:
                pvalue = policy.qfunc(s_f1, a1);
                if qmax < pvalue:
                    qmax = pvalue;
            update(policy, s_f, a, r + gamma * qmax, alpha);

            s_f     = s_f1
            a       = policy.epsilon_greedy(s_f)
            count   += 1

    return policy;
```

想了解更多代码，可以参见Github (https://github.com/algorithmdog/Reinforcement_Learning_Blog/tree/master/8.%E5%BC%BA%E5%8C%96%E5%AD%A6%E4%B9%A0%E7%B3%BB%E5%88%97%E4%B9%8B%E5%
实现的 Q Learning 算法的效果如下。

3. 总结

结果好烂啊。基本的强化学习算法还是无法解决 OpenAI Gym 里面的问题啊。本文的代码可以在 Github (https://github.com/algorithmdog/Reinforcement_Learning_Blog/tree/master/8.%E5%BC%BA%E5%8C%96%E5%AD%A6%E4%B9%A0%E7%B3%BB%E5%88%97%E4%B9%8B%E5%
上找到，欢迎有兴趣的同学帮我挑挑毛病。

最后欢迎关注我的公众号 AlgorithmDog，每周日的更新就会有提醒哦~

欢迎关注

公众号讲述机器学习和系统研发的轶事，
希望讲得有趣，每周日更新~

扫描二维码即可关注。您，不关注下么?

(http://7rf34y.com2.z0.glb.qiniucdn.com/c/d7cd54ebca51cfa9cce3e1686d752ebd)

强化学习系列 (http://www.algorithmdog.com/series/rl-series)系列文章

- 强化学习系列之一:马尔科夫决策过程 (http://www.algorithmdog.com/%e5%bc%ba%e5%8c%96%e5%ad%a6%e4%b9%a0-%e9%a9%ac%e5%b0%94%e7%a7%91%e5%a4%ab%e5%86%b3%e7%ad%96%e8%bf%87%e7%a8%8b)
- 强化学习系列之二:模型相关的强化学习 (http://www.algorithmdog.com/%e5%bc%ba%e5%8c%96%e5%ad%a6%e4%b9%a0%e7%b3%bb%e5%88%97%e4%b9%8b%e4%ba%8c-%e6%a8%a1%e5%9e%8b%e7%9b%b8%e5%85%b3%e7%9a%84%e5%bc%ba%e5%8c%96%e5%ad%a6%e4%b9%a0)
- 强化学习系列之三:模型无关的策略评价 (http://www.algorithmdog.com/reinforcement-learning-model-free-evalution)
- 强化学习系列之四:模型无关的策略学习 (http://www.algorithmdog.com/reinforcement-learning-model-free-learning)
- 强化学习系列之五:价值函数近似 (http://www.algorithmdog.com/reinforcement-learning-value-function-approximation)
- 强化学习系列之六:策略梯度 (http://www.algorithmdog.com/rl-policy-gradient)
- 强化学习系列之七:逆向强化学习 (http://www.algorithmdog.com/rl-irl)
- 强化学习系列之八:在 OpenAI Gym 上实现 Q Learning

打赏

强化学习 (/tags/%E5%BC%BA%E5%8C%96%E5%AD%A6%E4%B9%A0) OpenAI (/tags/OpenAI)