

# 有趣的机器学习

-  Python基础 ▾
-  机器学习 ▾
-  数据处理 ▾
-  其他 ▾

什么是 Q Learning (Reinforcement Learning 强化学习)



切换到 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](#) )

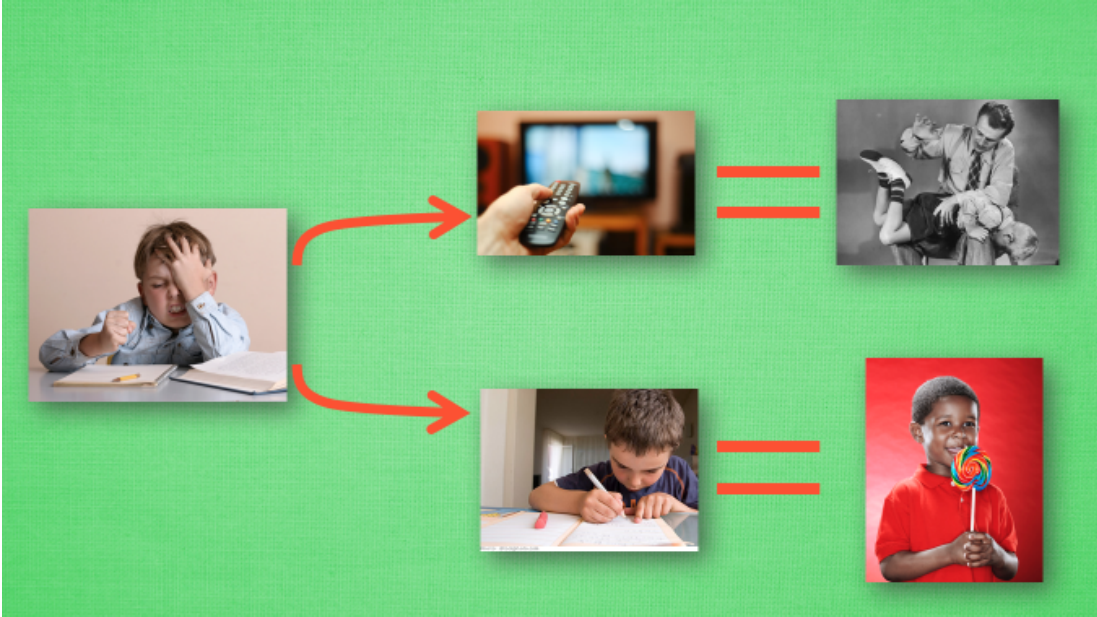
« 上一个

下一个 »

## Q Leaning

- 学习资料:
  - 强化学习教程
  - 强化学习模拟程序
  - Q learning Python 教程
  - 学习书籍 [Reinforcement learning: An introduction](http://ufal.mff.cuni.cz/~straka/courses/npfl114/2016/sutton-bookdraft2016sep.pdf)

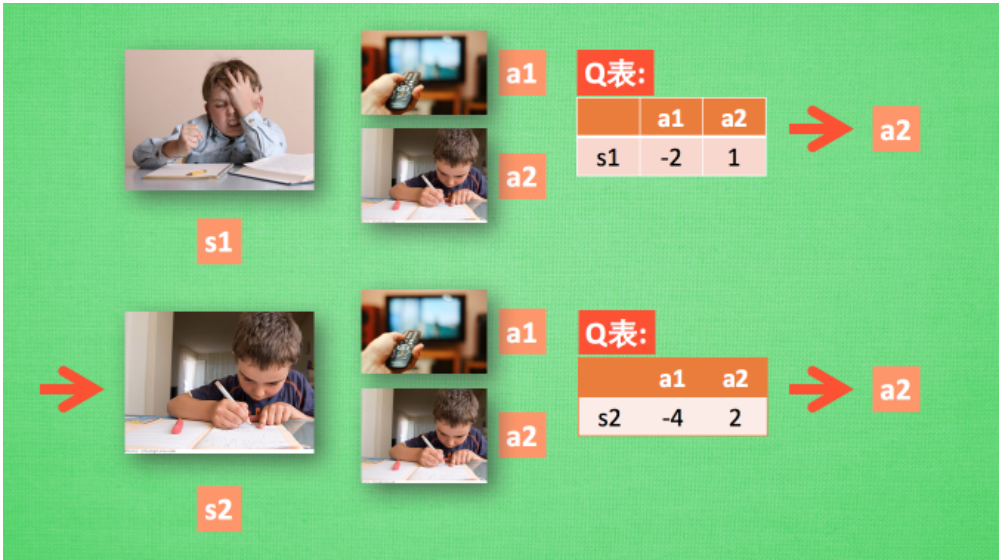
今天我们会来说说强化学习中一个很有名的算法, Q-learning



我们做事情都会有一个自己的行为准则, 比如小时候爸妈常说”不写完作业就不准看电视”. 所以我们在 写作业的这种状态下, 好的行为就是继续写作业, 直到写完它, 我们还可以得到奖励, 不好的行为 就是没写完就跑去看电视了, 被爸妈发现, 后果很严重. 小时候这种事情做多了, 也就变成我们不可磨灭的记忆. 这和我们要提到的 Q learning 有什么关系呢? 原来 Q learning 也是一个决策过程, 和小时候的这种情况差不多. 我们举例说明.

假设现在我们处于写作业的状态而且我们以前并没有尝试过写作业时看电视, 所以现在我们有两种选择 , 1, 继续写作业, 2, 跑去看电视. 因为以前没有被罚过, 所以我选看电视, 然后现在的状态变成了看电视, 我又选了 继续看电视, 接着我还是看电视, 最后爸妈回家, 发现我没写完作业就去看电视了, 狠狠地惩罚了我一次, 我也深刻地记下了这一次经历, 并在我的脑海中将 “没写完作业就看电视” 这种行为更改为负面行为, 我们在看看 Q learning 根据很多这样的经历是如何来决策的吧.

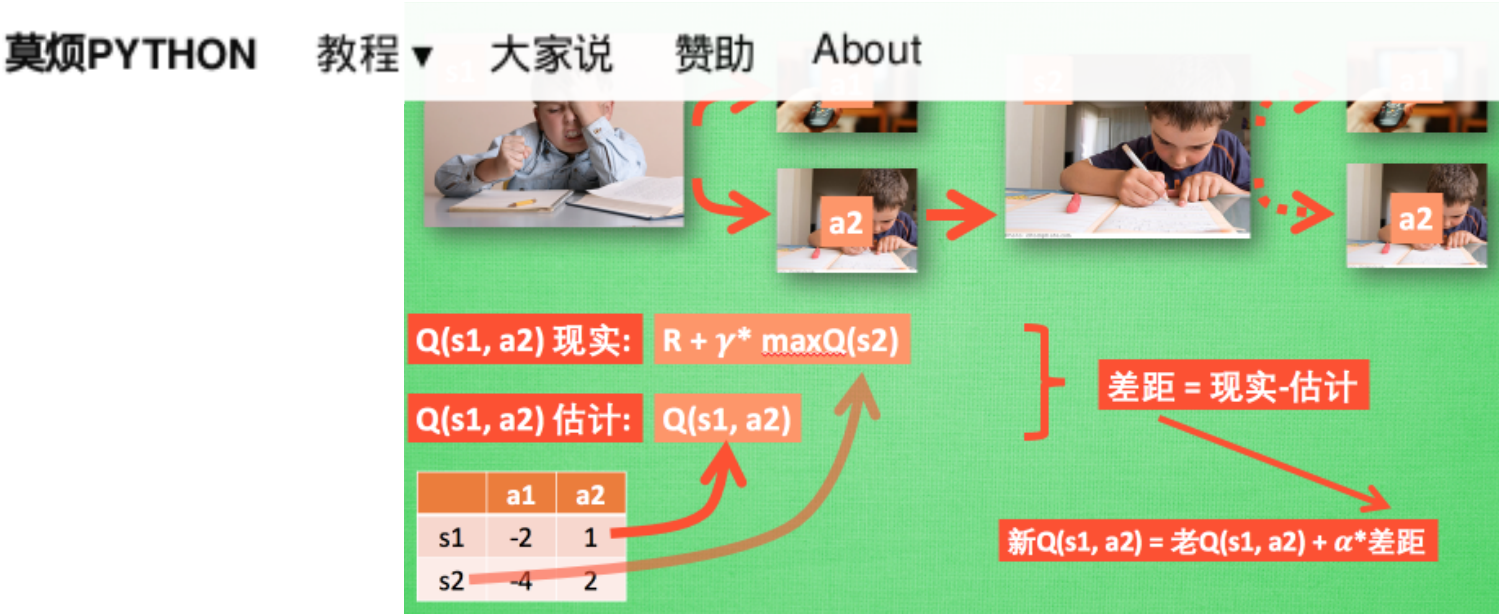
### Q-Learning 决策



假设我们的行为准则已经学习好了, 现在我们处于状态s1, 我在写作业, 我有两个行为 a1, a2, 分别是看电视和写作业, 根据我的经验, 在这种 s1 状态下, a2 写作业 带来的潜在奖励要比 a1 看电视高, 这里的潜在奖励我们可以用一个有关于 s 和 a 的 Q 表格代替, 在我的记忆Q表格中,  $Q(s1, a1)=-2$  要小于  $Q(s1, a2)=1$ , 所以我们判断要选择 a2 作为下一个行为. 现在我们的状态更新成 s2 , 我们还是有 两个同样的选择, 重复上面的过程, 在行为准则Q 表中寻找  $Q(s2, a1)$   $Q(s2, a2)$  的值, 并比较他们的大小, 选取较大的一个. 接着根据 a2 我们到达 s3 并在此重复上面的决策过程. Q learning 的方法也就是这样决策的. 看完决策, 我看来来研究一下这张行为准则 Q 表是通过什么样的方式更改, 提升的.

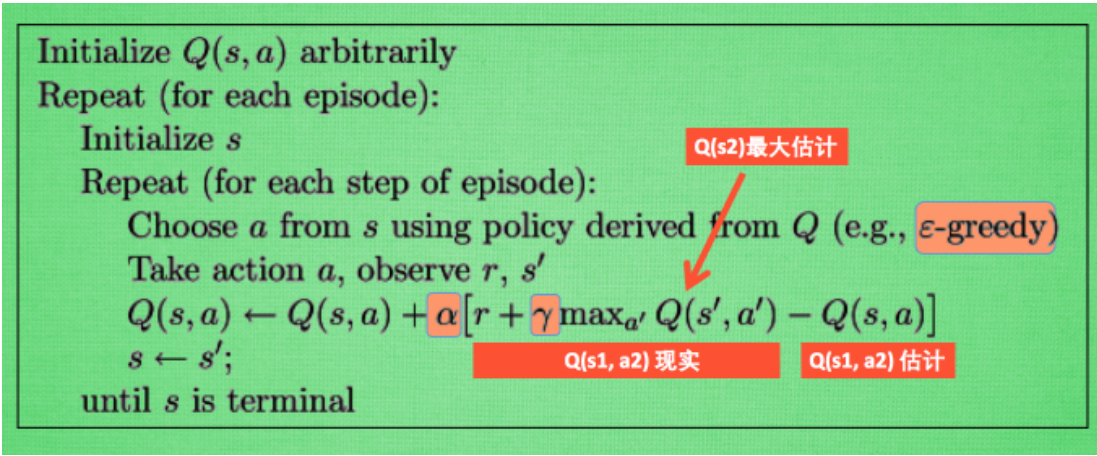
### Q-Learning 更新





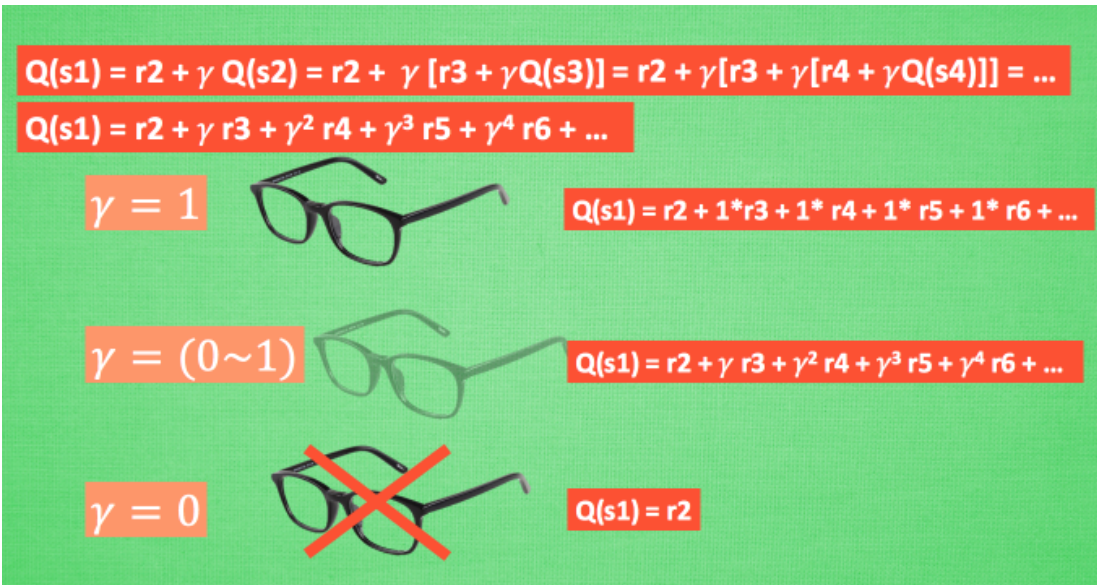
所以我们回到之前的流程, 根据 Q 表的估计, 因为在 s1 中, a2 的值比较大, 通过之前的决策方法, 我们在 s1 采取了 a2, 并到达 s2, 这时我们开始更新用于决策的 Q 表, 接着我们并没有在实际中采取任何行为, 而是再想象自己在 s2 上采取了每种行为, 分别看看两种行为哪一个的 Q 值大, 比如说  $Q(s2, a2)$  的值比  $Q(s2, a1)$  的大, 所以我们把大的  $Q(s2, a2)$  乘上一个衰减值 gamma (比如是0.9) 并加上到达s2时所获取的奖励 R (这里还没有获取到我们的棒棒糖, 所以奖励为 0), 因为会获取实实在在的奖励 R, 我们将这个作为我现实中  $Q(s1, a2)$  的值, 但是我们之前是根据 Q 表估计  $Q(s1, a2)$  的值. 所以有了现实和估计值, 我们就能更新 $Q(s1, a2)$ , 根据 估计与现实的差距, 将这个差距乘以一个学习效率 alpha 累加上老的  $Q(s1, a2)$  的值 变成新的值. 但时刻记住, 我们虽然用  $\max Q(s2)$  估算了一下 s2 状态, 但还没有在 s2 做出任何的行为, s2 的行为决策要等到更新完了以后再重新另外做. 这就是 off-policy 的 Q learning 是如何决策和学习优化决策的过程.

### Q-Learning 整体算法



这一张图概括了我们之前所有的内容. 这也是 Q learning 的算法, 每次更新我们都用到了 Q 现实和 Q 估计, 而且 Q learning 的迷人之处就是在  $Q(s1, a2)$  现实中, 也包含了一个  $Q(s2)$  的最大估计值, 将对下一步的衰减的最大估计和当前所得到的奖励当成这一步的现实, 很奇妙吧. 最后我们来说说这套算法中一些参数的意义. Epsilon greedy 是用在决策上的一种策略, 比如  $\epsilon = 0.9$  时, 就说明有90% 的情况我会按照 Q 表的最优值选择行为, 10% 的时间使用随机选行为. alpha 是学习率, 来决定这次的误差有多少是要被学习的, alpha是一个小于1 的数. gamma 是对未来 reward 的衰减值. 我们可以这样想象.

### Q-Learning 中的 Lambda




我们重写一下  $Q(s_1)$  的公式, 将  $Q(s_2)$  拆开, 因为 $Q(s_2)$ 可以像  $Q(s_1)$ 一样,是关于 $Q(s_3)$  的, 所以可以写成这样, 然后以此类推, 不停地这样写下去, 最后就能写成这样, 可以看出 $Q(s_1)$  是有关于之后所有的奖励, 但这些奖励正在衰减, 离  $s_1$  越远的状态衰减越严重. 不好理解? 行, 我们想象 Qlearning 的机器人天生近视眼,  $\gamma = 1$  时, 机器人有了一副合适的眼镜, 在  $s_1$  看到的  $Q$  是未来没有任何衰变的奖励, 也就是机器人能清清楚楚地看到之后所有步的全部价值, 但是当  $\gamma = 0$ , 近视机器人没了眼镜, 只能摸到眼前的 reward, 同样也就只在乎最近的大奖励, 如果  $\gamma$  从 0 变到 1, 眼镜的度数由浅变深, 对远处的价值看得越清楚, 所以机器人渐渐变得有远见, 不仅仅只看眼前的利益, 也为自己的未来着想.

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 0

按时间正序

还没有评论，快来抢沙发吧！

来必力是？ | 询问

支持 让教学变得更优秀

点我 赞助 莫烦

4 of 5

2017年06月06日 17:21

关注我的动向:

[Youtube频道](#) [优酷频道](#) [Github](#) [微博](#)

**Email:** morvanzhou@hotmail.com

© 2016 morvanzhou.github.io. All Rights Reserved