# 你有一份强化学习线路图，请查收。

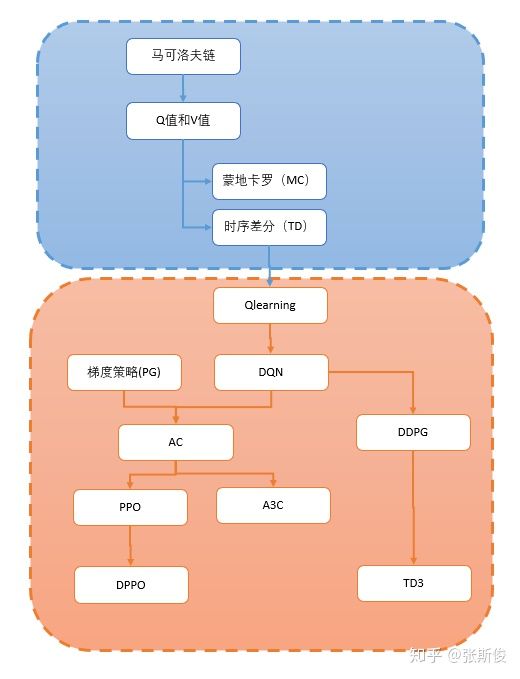
[](https://www.zhihu.com/people/zhang-si-jun-52)

[张斯俊](https://www.zhihu.com/people/zhang-si-jun-52" \t "/home/admin/文档\\x/_blank)

愿成为一把梯子，助你跨过无数的坑。

本文希望用最短的文字，介绍强化学习的入门线路图。从马可洛夫模型到DPPO。

先上线路图，我会逐点解释。



线路图分为两部分

- 第一部分是基础概念；

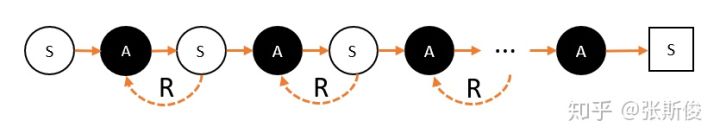
首先要弄清楚强化学习到底是干嘛的？它围绕怎样一个问题，而后面所有的所有，都是围绕这个问题的解决方案。

- 第二部分是深度强化学习的算法；

他们是怎么一直发展的，脉络是怎样的？

### 马可洛夫链

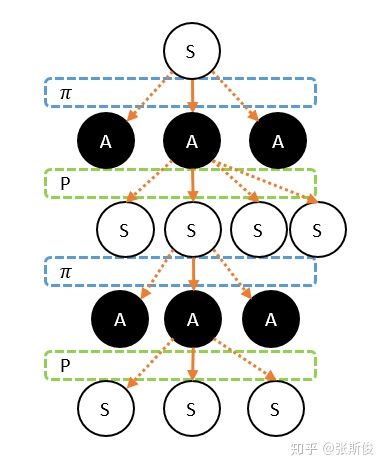
先来看马可洛夫链。马可洛夫链长这样子：



马可洛夫链描述的是智能体和环境进行互动的过程。简单说：智能体在一个状态(用S代表)下，选择了某个动作(用S代表)，进入了另外一个状态，并获得奖励(用R代表)的过程。

所以，希望通过让智能体在环境里获取**最多的奖励**，把智能体训练成想要的样子——就是能完成某些特定的任务。

所以，马上遇到第一个坑：马尔科夫链，其实应该叫马尔科夫树吧！



看到的链，是因为从现在往后看，但如果往前看，是充满不确定性的。

这里的不确定性包括两方面：

* 策略：智能体的每次选择都不是固定的，而是按照一定的策略分布。这个概率分布称为**策略**，用PI表示。
* 状态转移概率：这个只跟环境有关系。例如飞行棋的掷骰子游戏，执行同样的动作，也有可能进入不同的状态。

如果想对马尔科夫有更多了解，请看这一篇专栏：[张斯俊：怎样正确理解马尔科夫链？​](https://zhuanlan.zhihu.com/p/109217883" \t "/home/admin/文档\\x/_blank)

### V值和Q值

于是，如果要想让智能体能够获得**奖励最大化**就面临两个问题。

1. 未来的路很长，不能只凭眼前的收获，就马上做决定；要考虑未来；
2. 未来的路充满不确定性：不能走一次某一条路，就下决定了。

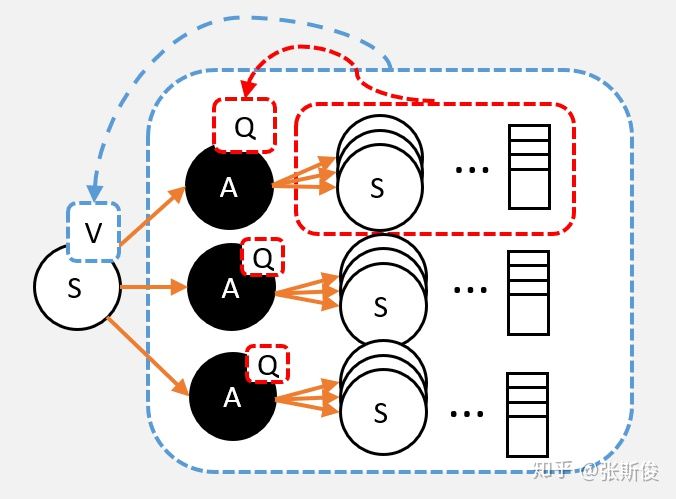
前路**长且险**，那该怎么办呢？

需要V和Q。

其实V和Q的意义是类似的，唯一的不同是**V是对状态节点的估算，Q是对动作节点的估算**。

那估算什么呢？

估算从该节点，一直到最终状态，能够获得的奖励的总和的平均值。



请记着这个意义，后面的所有算法基本都会围绕这个定义。因为，如果智能体已经知道某个动作或者状态 一直到最后状态能够一共获得多少奖励。那智能体选多的那个就完了。

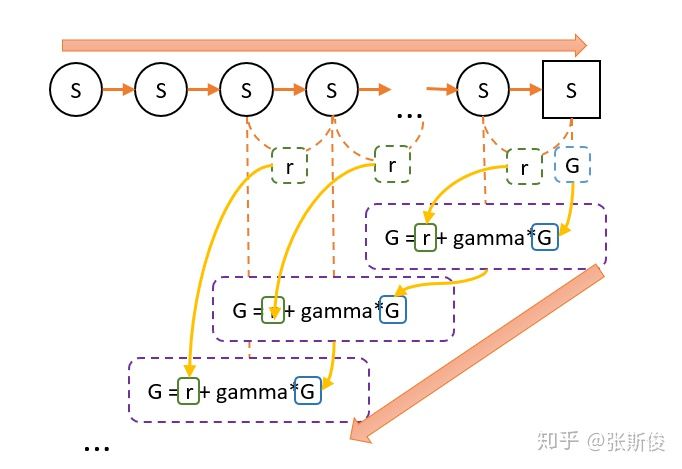
所以用个例子说明：用影分身，从某个节点出发，一直到最终节点。每个影分身途径都会获得不同的奖励。那么，平均一个影分身能够获得多少奖励，就是对这个节点的价值的估算。

如果想知道详细，可以看这篇专栏：[张斯俊：如何理解强化学习中的Q值和V值？​](https://zhuanlan.zhihu.com/p/109498587" \t "/home/admin/文档\\x/_blank)

### 蒙地卡罗(MC)和时序差分(TD)

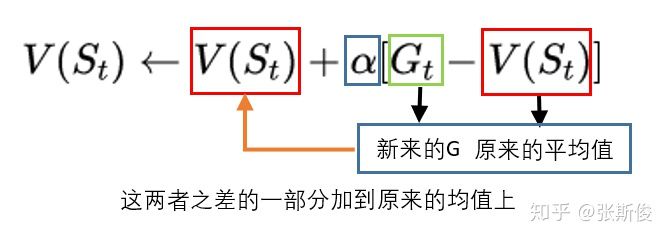
那怎么估算每个节点的价值呢？

蒙地卡罗会让智能体从某个状态S出发，直到最终状态，然后回过头来给每个节点标记这次的价值G。G代表了某次，智能体在这个节点的价值。



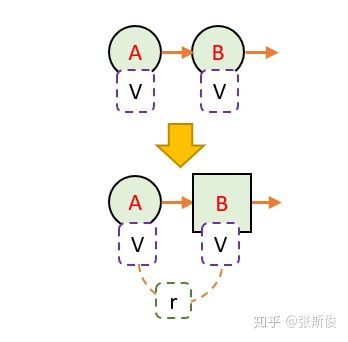
经过多次后，把每个状态的G值进行平均。这就是状态的V值。

但为了方便，对平均进行一些优化。于是获得用MC估算V值的公式：



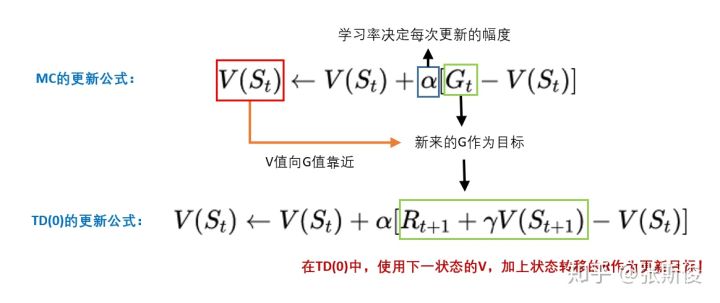
详细的理解，可以参看这篇专栏：

而时序差分是**一步一回头**。用下一步的估值，估算当前状态的估值。



这就相当于，把下一步状态直接当成最终状态。但这个状态它自己包含了这个状态的价值。

因此，可以把蒙地卡罗用到的G值，用V(St+1) + r 代替：



详细的理解，可以参看这篇专栏：

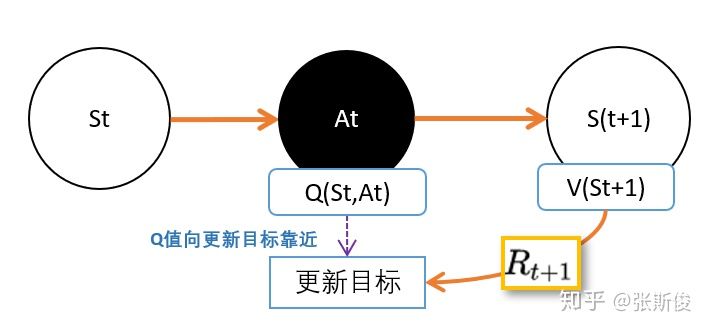
[张斯俊：如何用蒙地卡罗方法（Monte-Carlo）估算V值？​](https://zhuanlan.zhihu.com/p/109755443" \t "/home/admin/文档\\x/_blank)

[张斯俊：[番外]蒙地卡罗MC的更新公式怎么来的？​](https://zhuanlan.zhihu.com/p/110118392" \t "/home/admin/文档\\x/_blank)

[张斯俊：如何用时序差分TD估算状态V值？​](https://zhuanlan.zhihu.com/p/110132710" \t "/home/admin/文档\\x/_blank)

### Qlearning 和 SARSA

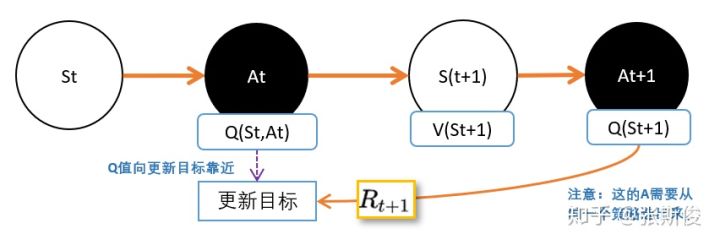
之前学习了用TD估算V值。但其实用TD预估Q值，其实会来得更方便，因为要的就是智能体选择动作嘛。



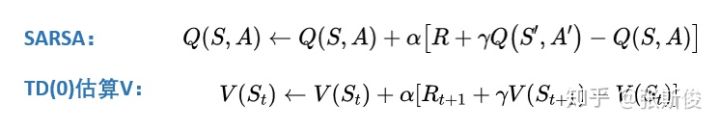
但问题是，如果既要估算V(St+1),又要估算Q(St,At)。就相当麻烦了。能不能都统一成Q值呢？也就是说V(St+1)用一个动作的Q值所代替。

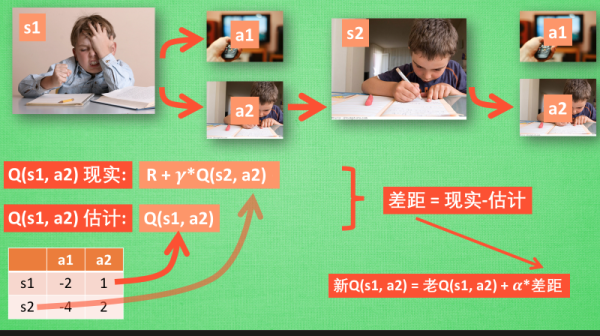
于是便有两种不同的替代方案：Qlearning和SARSA。

先说SARSA，SARSA的想法是，用同一个**策略下产生的动作A的Q值替代V(St+1)**。如下图：



于是有了，SARSA的更新公式。

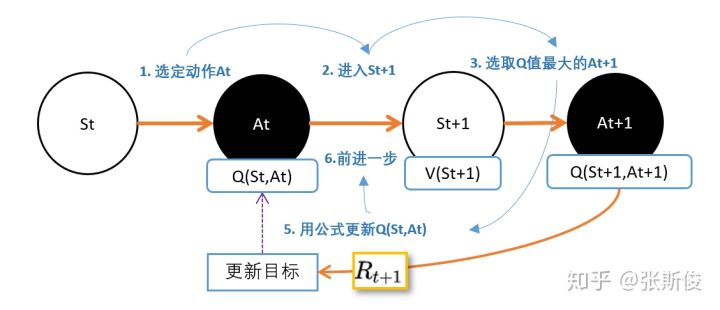


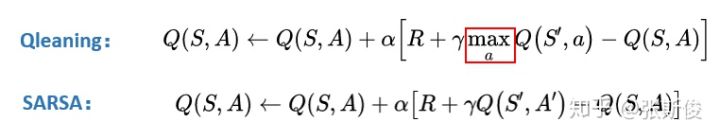


可以和TD估算V值对比一下，几乎是一模一样的，只是把V换成Q。

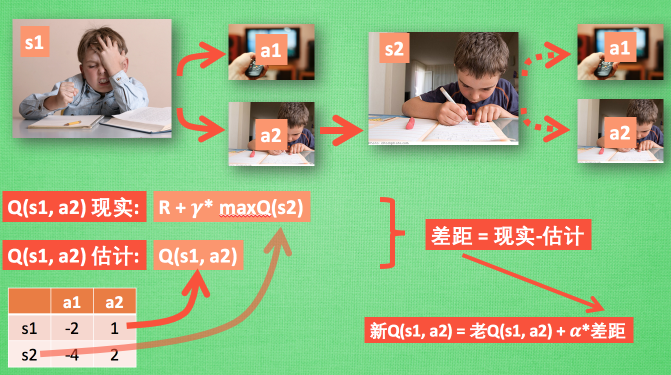
但我还是建议记得上面说的，是用Q替代V(St+1)。因为跳过这一步，就理解不了Qlearning了。

Qlearning的想法其实也很直观：既然的目标是选取最大收益，所以，肯定会选择一个能够获得最大Q值的动作。也就是说，在实际选择中，我不可能选择不是最大Q值的动作。所以，**应该用所有动作的Q值的最大值替代V(St+1)。**





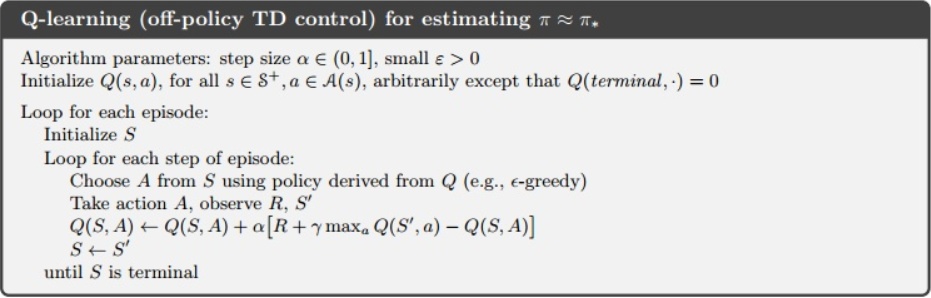
Qlearning公式和SARSA相比，就差那么一个max(SARSA中前后两个Q值是同一个**策略下产生的同一动作A?**)。



如果想对Qlearning和SARSA有更多了解。可以看一下这篇专栏：

[张斯俊：[理论篇]怎样直观理解Qlearning算法？​](https://zhuanlan.zhihu.com/p/110338833" \t "/home/admin/文档\\x/_blank)

[张斯俊：手把手教你实现Qlearning算法[实战篇]​](https://zhuanlan.zhihu.com/p/110410276" \t "/home/admin/文档\\x/_blank)

[](https://zhuanlan.zhihu.com/p/110410276)

### 深度强化学习

现在就结束了第一部分，开始进入深度强化学习的部分了。

为什么深度强化学习这么强，是因为深度强化学习增加了一个很强的武器——深度神经网络。

有人说，深度神经网络很红呀，但我不懂。后面的怎么学？

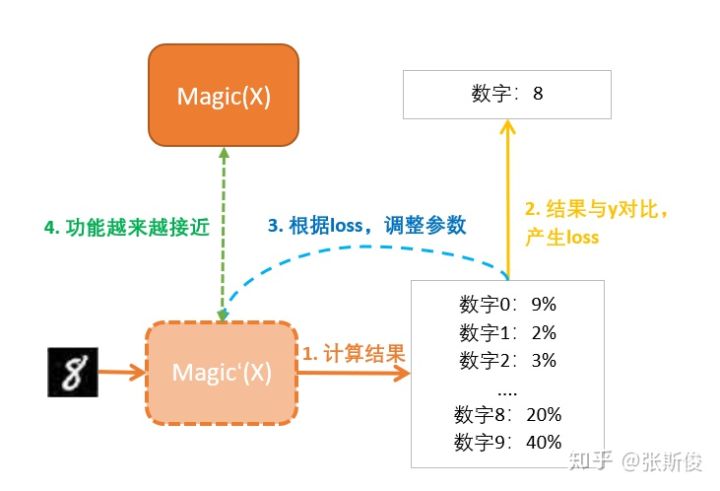
没关系，如果让我两个字概括，深度神经网络就是一个**函数**。

函数其实也很简单，就是描述两个东西的对应关系。F(x) = y , 描述的就是x和y之间的关系。

以前的函数，需要去精心设计的，要设计，就要描述其中的关系。但有些东西明明知道他们有关系，但又不好**描述**清楚。

例如，手写数字识别，一个正常人写的数字8，人类都能认出来。但却描述不出来，知道是两个圈是8，但有些人的圈明明不闭合，也认得出是8...

但深度神经网络这个工具就能自己学会这些关系。它是怎样做的呢？



要学习一个神奇函数Maigic(),辨别手写数字，也就是输入一张8的图，输出这个数字是什么。

1. 先设一个Magic'(X),其中的X就是输入的图片；
2. 计算结果是各个数字的概率。这个判断一开始通常都是错的，但没关系，会慢慢纠正它。
3. 纠正就需要有一个目标，没有目标就没有对错了。这里的目标是人类给他们标注的，告诉Magic'：这玩意儿是数字8
4. 目标和现实的输出总是有一段距离的，这段距离称为损失(loss).
5. 调整Magic'函数的参数，让损失最小化。也就是说，离目标越来越近。

最后你就发现Magci’函数的功能离心目中要找的Magci函数越来越近。

如果你想更多了解深度神经网络，可以看一下之前的专栏。

[张斯俊：一篇文章带你了解深度神经网络​](https://zhuanlan.zhihu.com/p/110531783" \t "/home/admin/文档\\x/_blank)

### DQN

DQN = Deep learning + Qleanrning。

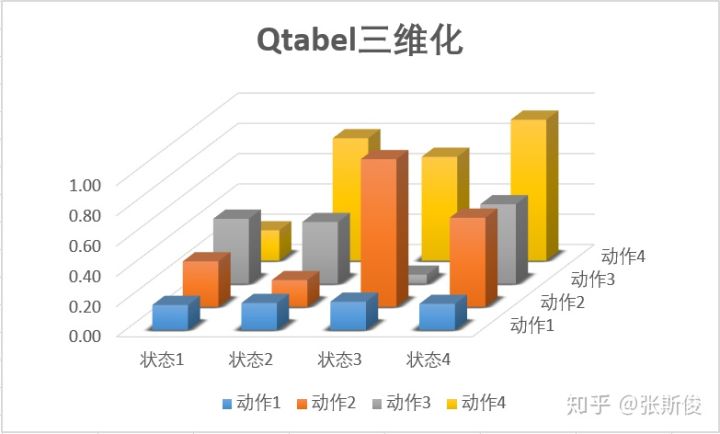
Qleanrning有一个问题：只能解决格子类型离散型状态问题，对连续型状态束手无策。

这是因为Qlearning在实做的时候用的是Q表格(Qtable)。表格这玩意儿注定就只能存离散的东西。

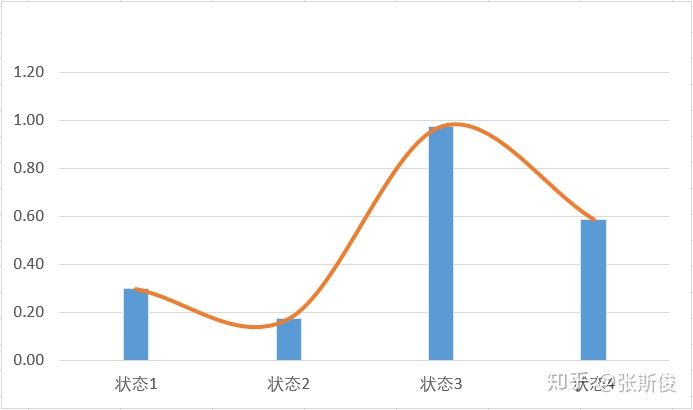
但刚才说的神经网络，正好就能解决这个问题，因为神经网络是个函数。可以处理连续型的问题。两者一拍即合！

在这我想先给大家一个不够准确，但很有用的理解方式。

用三维图，把Q表格显示出来，长这样子。



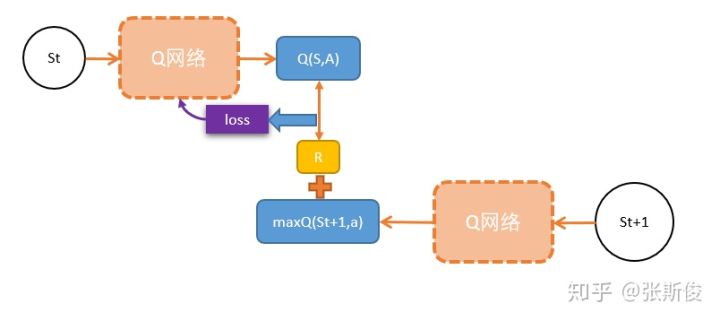
图中描述的就是某个状态下的某个动作，能够取得的Q值的大小。Q越大，柱子越高。



由于状态是连续型的，就相当于把状态用线给连起来。这样不但可以计算S1，S2，也可以计算状态S1.5了。

有一个问题，是用万能的神经网络的时候，需要解决的。就是更新的目标是什么？网络应该向什么地方优化。

其实，如果你对之前的Qlearning理解够深的话，就知道，更新用的是下一状态的Q值+奖励，作为更新的目标。如下图。



在实做上DQN还有很多需要注意的地方和技巧，欢迎看一下这篇文章：

[张斯俊：三维可视化助你直观理解DQN算法[DQN理论篇]​](https://zhuanlan.zhihu.com/p/110620815" \t "/home/admin/文档\\x/_blank)

[张斯俊：用可视化直观理解DQN[DQN实战篇]​](https://zhuanlan.zhihu.com/p/110657606" \t "/home/admin/文档\\x/_blank)

### double DQN

DQN有很多变种，这里只提一个Double DQN。主要是为了解决一个问题：DQN对Q的估值通常会过大。

直观地说，你可以这样认为： 用的是下一状态中，Q值最大的作为当前状态的估算。下一状态的Q值，以下下状态的最大Q值作为估算...这就有点像大话骰,这个Q值越传播就越大。

那怎么办？用两个网络对Q进行预估，取最小的那个。就相当于，你们尽管吹牛，大的我不要，我要小的。

当然，你会说这都可以？答案就是在试验当中，有奇效。

这就是DoubleDQN的想法。

DQN还有其他的一些变种，例如Dueling DQN。大家可以参考一下这几篇文章。

[张斯俊：Double DQN原理是什么，怎样实现？（附代码）​](https://zhuanlan.zhihu.com/p/110769361" \t "/home/admin/文档\\x/_blank)

[张斯俊：[番外篇]DuelingDQN为何那么强？(附代码)​](https://zhuanlan.zhihu.com/p/110807201" \t "/home/admin/文档\\x/_blank)

### 策略梯度(PG)

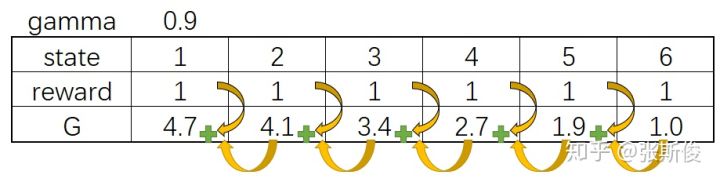
不知道大家现在有没有留意，其实进入一个大坑。

现在是把Q值和V值往死里算呀，但实际上，并不需要Q值呀，需要的是能获得最多的奖励总和呀。

既然现在发明出宇宙最强无敌的Magic函数——神经网络。那直接用神经网络magic(s)=a不行吗？

恭喜你，这就是PG的基本思想。

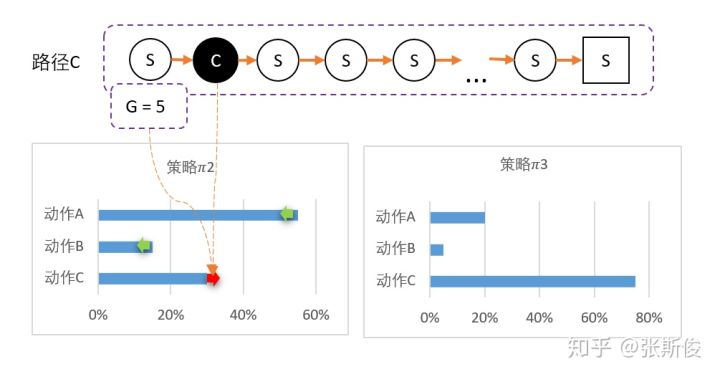
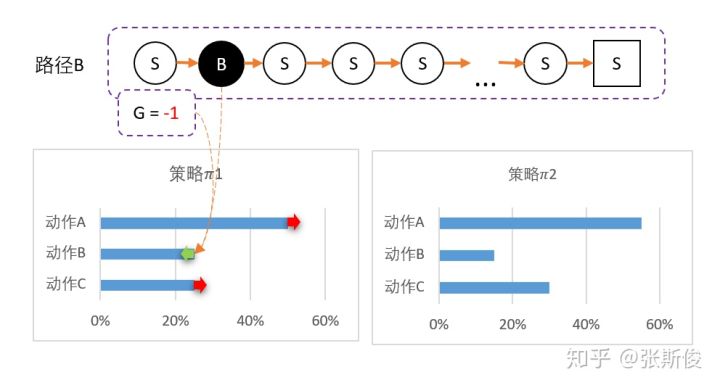
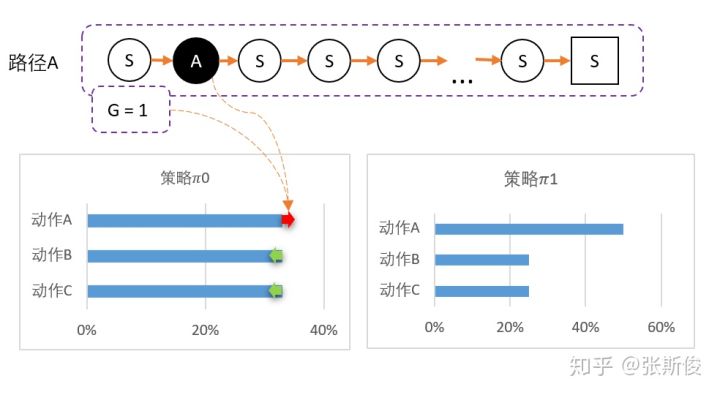
PG用的是MC的G值来更新网络。也就是说，PG会让智能体一直走到最后。然后通过回溯计算G值。



如果忘记了，可以看一下MC的专栏文章)

于是得到S - A - G 的数据。这里的G就是对于状态S，选择了A的评分。也就是说， - 如果G值正数，那么表明选择A是正确的，希望神经网络输出A的概率增加。(鼓励) - 如果G是负数，那么证明这个选择不正确，希望神经网络输出A概率减少。(惩罚) - 而G值的大小，就相当于鼓励和惩罚的力度了。

分别以ABC三条路径作为例子：



为此，可以用**带权重的梯度**。

如果对此有疑问，可以看一下这篇专栏：

[张斯俊：如何理解策略梯度（Policy Gradient）算法？[附代码]​](https://zhuanlan.zhihu.com/p/110881517" \t "/home/admin/文档\\x/_blank)

### Actor-Critic

知道，MC的效率是相对比较低的，因为需要一直走到最终状态。所以希望用TD代替MC。那么可不可以把PG和DQN结合呢？

注意：这里是一个大坑。个人更倾向于把AC理解成PG的TD版本，而不是PG+DQN。

这是为什么呢？

Critic网络负责估算Q值 Actor网络负责估算策略

这不是很完美吗？

但要注意，Q值都是正数，容易掉进“正数陷阱”。

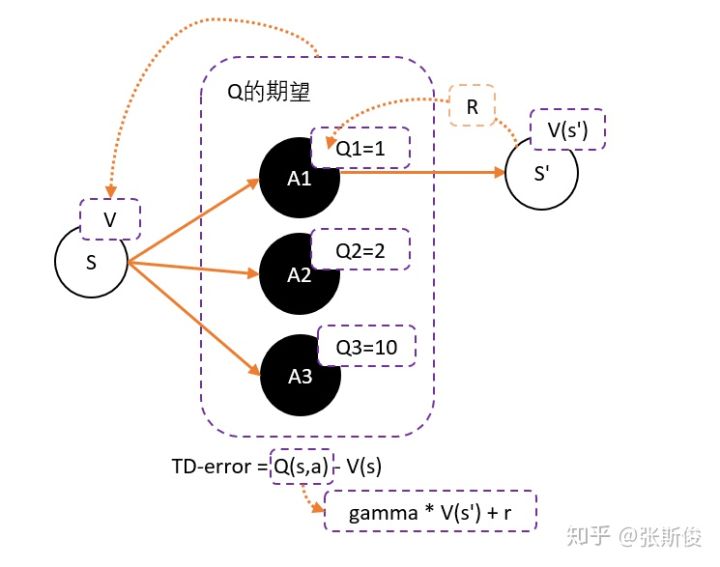
假设用Critic网络，预估到S状态下三个动作A1，A2，A3的Q值分别为1,2,10。

但在开始的时候，采用平均策略，于是随机到A1。于是用策略梯度的带权重方法更新策略，这里的权重就是Q值。

于是策略会更倾向于选择A1，意味着更大概率选择A1。结果A1的概率就持续升高...

那要怎么办？把Q值弄成有正有负就可以了。一堆数减去他们的平均值一定有正有负吧！Q减去Q的期望值，也就是V值，就可以得到有正有负的Q了。

也就是说Actor用Q(s,a)-V(s)去更新。但之前也说过Q和V都要估算太麻烦了。能不能只统一成V呢？



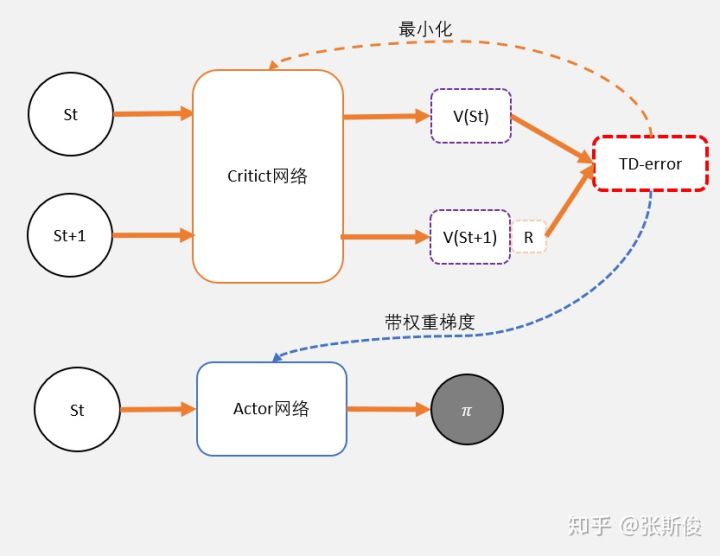
Q(s,a)用gamma \* V(s') + r 来代替，于是整理后就可以得到：

gamma \* V(s') + r - V(s) —— 把这个差，叫做TD-error

这个和之前DQN的更新公式非常像，只不过DQN的更新用了Q，而TD-error用的是V。

眼尖的同学可能已经发现，如果Critic是用来预估V值，而不是原来讨论的Q值。那么，这个TD-error是用来更新Critic的loss了！

所以，强烈建议大家不要把AC看成PG+DQN，而是看成是PG的TD版本。



AC还有个很好的特性，因为用两个网络，所以也把连续动作控制型问题解决了。详细可以看看本专栏这篇文章。

[张斯俊：理解Actor-Critic的关键是什么？​](https://zhuanlan.zhihu.com/p/110998399" \t "/home/admin/文档\\x/_blank)

### PPO

在强化学习中，数据来自智能体和环境互动。所以，数据都弥足珍贵，希望尽量能够利用好每一份数据。

但AC是一个在线策略的算法，也就是行为策略跟目标策略并不是同一个策略。

为了方便讨论，先理清楚两个概念： - **行为策略**——不是当前策略，用于**产出数据** - **目标策略**——会更新的策略，是需要**被优化的策略**

如果两个策略是**同一个策略**，那么称为**On Policy**，**在线策略**。如果**不是同一个策略**，那么**Off Policy**，**离线策略**。

这样说有点难以理解，举个例子：

如果在智能体和环境进行互动时产生的数据打上一个标记。标记这是第几版本的策略产生的数据,例如 1， 2... 10

现在的智能体用的策略 10，需要更新到 11。如果算法只能用 10版本的产生的数据来更新，那么这个就是在线策略；如果算法允许用其他版本的数据来更新，那么就是离线策略。

所以，需要用到重要性更新的，就可以用上之前策略版本的数据了。

详细大家可以看一下这篇专栏：

[张斯俊：如何直观理解PPO算法?[理论篇]​](https://zhuanlan.zhihu.com/p/111049450" \t "/home/admin/文档\\x/_blank)

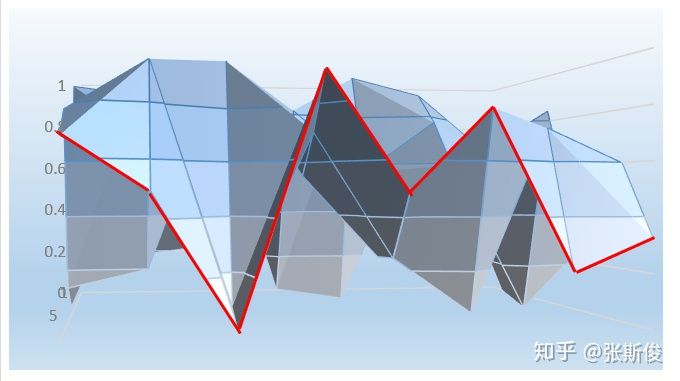
[张斯俊：如何直观理解PPO算法[实战篇]​](https://zhuanlan.zhihu.com/p/111068310" \t "/home/admin/文档\\x/_blank)

### DDPG

这里也有大坑，因为往上推荐的学习路径都这样，都是先学AC和PPO。再学DDPG，DDPG也是AC框架下的，所以经常会拿DDPG和AC一起比。

但我认为DDPG虽然是AC框架，但理解的时候，应该从DQN开始理解。

DDPG就是为了解决DQN连续控制型问题而产生的。



之前说，DQN的神经网络就相当于用线把Qtable的状态连起来。那翻到DDPG中，Critic网络就相当于用一张布，把整个Qtable的所有柱子都覆盖了。

DDPG的Actor接受输入一个状态，就相当于在这块布切沿着这个状态S切一个面。Actor的任务就是希望在这个面上找寻最高点，也就是最大的Q值。

所以和AC不同，DDPG预估的是Q而不是V。而DDPG的Actor采用的是梯度上升的方式找出最大值。而AC和PPO的Actor采用带权重更新的方法。

如果想对DDPG有更深入的了解，那么请看这一篇专栏：

[张斯俊：一文带你理清DDPG算法​](https://zhuanlan.zhihu.com/p/111257402" \t "/home/admin/文档\\x/_blank)

### TD3

学习了DDPG，那么TD3的理解就很简单了。

说DDPG源于DQN，DQN有一个大问题，就是高估Q值，所以DDPG也有这个问题。

和DQN一样，采用双Q网络，取最小值的方式就可以了。

DDPG用了4个网络，而TD3，用了6个网络。

...上传图片失败...

想了解更多TD3的知识，可以看这一篇专栏。

[张斯俊：什么是TD3算法？​](https://zhuanlan.zhihu.com/p/111334500" \t "/home/admin/文档\\x/_blank)

### A3C和DPPO

说过，强化学习中，最珍贵就是数据。

A3C和DPPO都使用了“影分身的方式”在最短的时间获取最多的数据，从而让智能体更快地学习。

这有点像一个班长和同学之间的故事。

邪恶的老师给聪明的班长一个任务，让班长一天之内交出一份十万字的莎士比亚全集的读书心得。

这怎么可能？但班长深得班里同学们喜爱。班长决定发动全体同学来完成这个任务。

但怎么分配任务呢？

聪明的班长给每个同学一套莎士比亚全集，然后公布了任务分配的规则：

1. 同学们就随便看任何一个你们感兴趣的段落就可以了，就算是重复了也没关系；
2. 但同学们需要把读过的提炼成**心得**，每隔一段时间汇报给班长；
3. 班长会负责汇总大家的心得; 但同学提交自己的心得之后，要看一下当前被汇总的最新版本的**心得**，因为这是集体的智慧，有助于大家提高阅读水平。
4. 最后，班长会把最新版本的**心得**提交给老师。

邪恶的老师看到这篇凝聚了大众智慧的读书心得，感动得流下泪水。而这位聪明的班长在毕业后投身AI事业，发明出A3C算法。

...故事我编不下去了...

相信大家在这个故事中，已经对A3C和DPPO的思路有个大致的想法。

在A3C中，“同学”不仅要和环境互动，产生数据，而且要自己从这些数据里面学习到“心得”。这里的所谓心得，其实就是计算出来的梯度；需要强调的是，worker向全局网络汇总的是**梯度**，而不是自己探索出来的**数据**。

在DPPO中，“同学”会各自看书，然后在书上划重点给“班长”看。这就相当于只提供和环境交互的数据，并不需要自己计算梯度了。

同学们想想，这是为什么呢？如果你还不太清楚，那么可以看看这一篇专栏:

[张斯俊：AC：看我的影分身之术[A3C]​](https://zhuanlan.zhihu.com/p/111336330" \t "/home/admin/文档\\x/_blank)

[张斯俊：PPO:看我的影分身之术[DPPO]​](https://zhuanlan.zhihu.com/p/111346592" \t "/home/admin/文档\\x/_blank)

感谢各位的阅读，我的专栏希望从更显浅直白的方式，绕过深奥的数据，直达算法思想的核心。有兴趣的同学请关注和转给有需要的同学吧！

[白话强化学习​](https://zhuanlan.zhihu.com/c_1215667894253830144" \t "/home/admin/文档\\x/_blank)