强化学习经典入门书的读书笔记系列--第三篇（上）

2017-05-12 小吕 神经网络与强化学习

从这一章开始，我们介绍的问题将是本书剩余部分一直要解决的问题。对于我们来说，这个问题定义了强化学习的全部领域：任何以解决这个问题为目的的方法都可以看成是强化学习方法。

我们这一章的目标是在更通用的意义上研究强化学习问题。我们试图在更广的范围里面去解释一些可以被认为是强化学习应用的任务。同时我们尽可能用数学语言来描述这些问题及其相关理论。我们也会引入一下数学上的重要元素，比如价值函数和贝尔曼方程。在人工智能领域普遍存在着应用广泛性和数学复杂性的矛盾，我们将讨论这些矛盾，并给出一些折衷的方法。

3.1 The agent-environment interface

强化学习问题如何定义呢？我们可以把它看成是从交互作用中获取某个目标的直接建模过程。学习者或者说是决策者被称作agent，和它交互的东西，是除了agent之外所有东西的组合，叫做environment。执行机构和环境之间的交互始终持续：agent依据当前environment选择一个动作，environment给予这个动作以reward，然后给出一个新的environment。后文中我们统一把agent叫做执行机构，把environment叫做环境。对环境的完整的确定定义了一个任务，这个任务就是一个强化学习实例。

更具体点说，agent和environment在一系列离散的时间点上进行交互作用，$t = 0, 1, 2, 3...$在每一个时间点上，agent会收到环境状态的表示：$S_t \in S$,S表示所有可能状态的集合，在这状态的基础上选择一个动作：$A_t \in A(S_t)$，$A(S_t)$表示在状态S_t下所有可能的动作的集合，之后，作为对动作的回应，agent会收到一个数值化的奖励（reward），然后进入新的状态：S_{t+1}。

在每一个时间点上，agent可以生成一个映射，这个映射是把当前状态映射到所有可能执行的动作的概率上。所有的映射组成一起就叫做policy，也就是策略，表示成π_t，其中$\pi_t(a|s)$就是当$S_t = s, A_t = a$的概率。强化学习方法就是用来确定agent的policy如何根据和环境交互的经验来变化的方法。agent的目标，粗略的说，就是最大化在长远过程之后得到的总奖励。

下图可以明确看出执行机构、环境、奖励之间的关系。

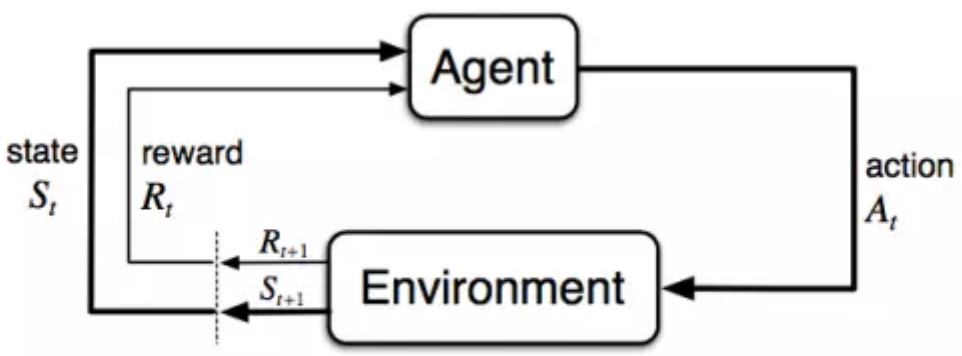


Figure 3.1: The agent–environment interaction in reinforcement learning.

这个框架很抽象，也很灵活，可以用在不同的问题上。比如，时间点不一定必须是固定间隔的真实时间，它也可以指的是任意的有关决策和动作的连续过程。动作可以是低层次的控制动作，比如机器人手臂电机的电压；也可以是高层次的决策动作，比如是否吃午餐或者是否去上学。相似的，环境状态也可以有不同的形式：它既可以低层次的传感器读数，也可以是抽象的符号表示，比如屋里的物体。甚至有的时候环境状态可以基于过往的记忆或者完全是精神层面的。比如一个执行机构可以处在“不知道物体在哪里”的状态，或者说处在被某个定义好的感觉所“震惊”的状态。比如有一些动作可以控制执行机构去选择思考某些事情，或者是应该把注意力集中在什么地方。总之，动作可以是我们能想到的任何我们想要做的决策，而状态可以是任何我们认为对做决策有用的东西。

特别要提一下的是，执行机构和环境之间的边界，并不是我们想象中的机器人或者动物的身体的物理边界。通常来说，这个边界离agent更近。比如：机器人中电机和机械的联系以及传感器的硬件应该属于环境而不是执行机构。类似的，如果考虑人和动物，那么肌肉、骨架、感觉器官应该属于环境。

关于执行机构和环境的区分边界的一个通用的规则就是：但凡不能被执行机构任意改动的部分都属于外界环境。然而我们并非假设环境中的所有东西对于执行机构来说都是未知的。比如，通常执行机构都会知道一些关于其得到的奖励是如何根据动作和环境状态的方程计算出来的。不过通常来说我们还是把reward的计算看成是属于外部环境的，因为reward决定了执行机构面对的任务，所以必须超出执行机构的控制范围才行（如果执行机构可以控制其得到的reward，就乱套了，环境状态的变化意义就消失了）。有时执行机构完全知道环境如何变化，然而仍然面临很困难的学习任务。这其中的原因就在于control和know之间的区别。也就是说，执行机构和环境的边界，在于其是否对该部分有完全的控制能力，而并非是否有完全的了解。

执行机构和环境的边界，在不同的应用场景下有不同的表现。比如，一个执行机构所做的高等决策，可以正是组成了环境状态的一部分，而这个环境状态又正是一个低等执行机构面临的，而这个低等执行机构，又帮助生成了高等的决策。在实际操作中，这个边界当我们选定了环境状态、动作和奖励的时候就被确定下来了。

强化学习使用action，state和reward三个元素，作为一种广泛的抽象。虽然对于实际问题的对应并不是百分百契合，但是仍然几乎是可以概括所有问题的。

在具体的应用中，如何正确的表示state和action，会对学习效果有很大影响。（这里或许可以类比成在写面向对象的程序时，如何封装一个类，有时好的封装定义会简洁，坏的定义会很复杂）。然而本书的主要目标还是关注当表示已经确定下来时，后续如何学的更好的问题。

接下来是书中给出了几个具体例子，表示在不同应用条件下的action和state区分，这里就不再一一解释。

3.2 Goals and rewards

在强化学习中，执行机构的目标是获取最大化的奖励总和。这意味着不能单单只看眼前的奖励，而要看长远过程下的奖励之和。

使用reward这个概念来形式化地表示目标是强化学习最重要的特点之一。尽管这种表示刚开始看起来有局限性，但是在实际应用中却被证明是及其灵活且具有广泛适用性。理解这一点最好的方式就是去看几个例子：假如我们想让一个机器人学会走路，研究人员提供了和机器人向前移动距离成比例的reward。为了让机器人学会走出迷宫，只有当机器人成功逃出迷宫时，reward为1,而其他大多数时间都为0；迷宫问题的另一种reward设置方法是：在成功逃出迷宫之前的每一步的reward都设置成-1，这样会让机器人更快的找到办法。

你可以从这些例子中看到，执行机构总是尽可能最大化它的reward。如果我们想让执行机构完成什么事情，我们就必须设法在执行机构正确完成我们的目标的一系列过程中给予奖励。因此很重要的一点是，我们设置的奖励要明确表明我们想完成的目标。说的更明白一点：奖励信号是用来告诉执行机构你最终想要得到什么，而不是你认为的怎么去得到。举例：下棋的时候，你应该在当你要真正赢得比赛的时候给予适当的奖励，而不应该在某个好的局部局势下给予正向奖励，比如吃掉对方的棋子，占据某个主动等等，虽然这些情况对于最终赢得胜利或许有用，但是如果总是在这些情况下给予奖励，那么执行机构就会倾向于获得局部优势，而忽略了全盘局面。

刚接触强化学习的同学，有时会惊讶于奖励是由环境决定的而不是由执行机构决定的。的确对于动物来说，大部分最终的目标都来源于自己身体内的计算：比如识别食物、饥饿、疼痛和快乐的传感器。然而，正如我们前面章节所说的，我们在理解强化学习问题时，最好把身体部分归于环境因素。比如，如果机器人的目标是其内部电池电量，那么这些就被看成是环境的一部分（尽管电池处于机器人内部，仍被看作外界因素）；如果我们的目标是机器人手臂的位置，那么手臂位置也看作是环境的一部分，也就是说agent和environment的边界在控制系统和机器人手臂之间。为了便于我们更好的理解强化学习问题，我们在理解这个边界概念的时候，不要被物理上的局限限制住，而是关注控制系统在哪里。

我们这么做的理由是agent的最终目标是一些它不能完美控制的东西，而不应该让agent可以随意改变reward。因此，我们把reward放在agent不能控制的范围，这并非是禁止agent去自我定义内在的reward。事实上，这正是强化学习要解决的问题。

3.3 Return

到目前为止我们明确了强化学习的目标，我们知道agent的目标是在长远时间内最大化总体reward（这里为什么不明确的说reward之和呢？读者思考一下）。那么如何正式的定义长远状况下的reward之和？假设在某个时间点t之后的一系列的reward序列为：　　。那么对于这个序列，我们选择哪些方面去最大化呢？通常，我们选择最大化“期望返回值（expected return）”
用　表示。注意这里某个时间点上的　，也就是期望返回值，是从该时间点往后所有获取的reward序列的某种函数，这个函数最简单的形式是将所有项加和，但是并非只能是加和。

这种表示形式在有着自然的时间点概念的应用中很合适，比如下棋的过程、走迷宫的过程或者任何重复性的交互过程。我们把这一类过程叫做片段。每一个片段都有一个终止状态，这个状态之后意味着整个过程重置成初始状态，或者来自初始状态标准分布的其中一个采样。带有片段式过程的任务叫做片段任务。在片段式任务中我们通常需要区分所有非终止状态，用S表示，终止状态，用S^+表示。

有了片段式任务，自然就有连续式任务。在连续式任务中，很难把交互过程看成离散的时间点。比如连续过程控制任务，或者有着很长寿命的的机器人。这样的上面片段式任务的定义就有点问题，因为最终的时间$T=\infty$。而我们希望最大化的返回值，可以很容易的就是无穷大。因此在本书中，我们将采用一种概念上很复杂但数学表示上很简单的return定义。

这个定义方式就是discounting sum。如下所示：

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ... = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

前面的系数，即discounting rate $0 \leq \gamma \leq 1$，决定了未来奖励的当前价值。

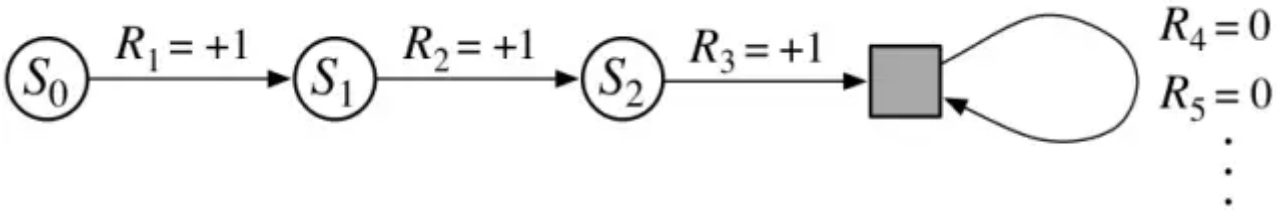
discounting rate 的含义就是，在当前时间点k步之后的奖励，在当前的价值是未来的γ^{k-1}倍。当该系数小于1，最终和会收敛，只要奖励序列是有限的。当系数为0，则执行机构是短视的，只顾眼前利益；如果系数大于1，执行机构是远视的，也就是重视长远奖励。

3.4 Unified Notation for Episodic and Continuing Tasks

在之前的内容里，我们讨论了两类任务，一种是离散性质的片段式任务，另一种是连续任务。前一种任务在数学上简单一点，因为它涉及的奖励序列是有限的。在本书中，我们有时会讨论片段式任务，有时也会讨论连续式任务，但是经常两者都会有，因此最好我们能有一种两者都适用的表示法。

为了精确表示片段式任务，我们需要额外的表示。我们需要考虑一系列的片段，每个片段包含有限的时间点序列，而不在是之前的单一的长时间序列。我们把每个片段中的时间序列的起始点都设置为0。因此我们需要把之前的S_t变成$S_{t,i}$,表明在片段i的时间点t。然而，当我们在讨论片段式任务的时候，我们几乎不需要考虑不同片段之间的不同。我们只需要考虑某一个片段的情况，然后声明这些情况对于所有片段都适用。因此，我们还是用S_t来代表$S_{t,i}$。

我们还需要另一个约定符号来统一两种任务形式。我们已经定义了有限项的返回值公式和无限项的返回值公式。这两个公式可以通过某种方式合并起来：把片段式任务的终点看成是进入了一个特殊的“吸收状态”（absorbing state），这种状态的特点是其状态转移过程始终发生在自身，并且产生的reward都是0。例如下图：



这张图中的灰色正方形就代表吸收态。这种形式在引入discounting rate的时候也适用。因此我们就可以定义了广义上的return，如下式所示：

$$G_t = \sum_{k=0}^{T-t-1} \gamma^k R_{t+k+1}$$

这个公式包含了$T=\infty$或者$\gamma = 1$的可能（但并非同时满足这两种可能）。我们在本书余下的部分会一直使用这种表示形式。

3.5 The Markov Property

在强化学习的框架中，agent做出的决策基于环境状态。在这一节，我们主要讨论什么样的状态信号是我们需要的，还有就是我们希望状态信号提供什么样的信息。特别的，我们定义了一种叫做马尔科夫性质的环境性质，这种性质的环境状态具有一些特别的特点。

在本书中，我们用“状态”这个词来表示agent可以获得的所有信息。我们假设这些状态来自于环境中的某些预处理系统。本书中我们不去讨论状态的构建、转化、或者获取等问题，并不是因为觉得这些不重要，而是想把主要重点放在“决策”过程上。

毋庸置疑，状态信息应该包含一些采集信息采集装置比如传感器，但实际上它包含的内容更多。状态表示可以时原始信息经过高度处理之后得到的，或者是依据原始序列信息，经过复杂变化之后得到的。比如，当我们移动目光看一片景色的时候，每一时间点，我们其实只能看到一小部分，然而最终我们的印象里得到的是整个景象和细节。用一个更普通的例子说明，一个控制系统可以通过在不同时间测量同一物体的位置而得到它的速度。在这两个例子中可以看出，状态信息都依据即时的传感信息进行了重新处理。我们没必要非得把状态信息限制为最原始的信息，事实上在某些应用中，我们需要对这些原始传感信息进行重构，获取一些高维信息。

从另一方面来说，状态信息不应该把关于环境的所有信息都提供给agent。比如agent正在玩blackjack，我们就不能告诉它下一张牌是什么。如果agent正在打电话，我们就不能让它提前知道谁打的电话。在这几个例子中，在环境中存在着隐藏信息，尽管那些信息对于agent来说是有用的，但是agent不能知道这些信息，因为它没有任何可以获取这些信息的途径。

理想情况下，状态信息应该简洁综合了历史信息 ，也就是用某种方式记住所有相关信息。能够成功保留所有相关信息的状态信息叫做具有马尔科夫状态的状态。例如棋盘的盘面状态，可以看成具有马尔科夫性质，因为它包含了所有以往可以生成当前盘面的序列信息。很多序列信息都丢失了，但是所有对于将来盘面重要的信息都被保留下来了。相似的，炮弹当前的位置和速度信息就可以完全决定将来的飞行轨迹。通常也把这种性质叫做“independence of path” ，因为所有关键的信息都存在当前的状态信息中，也就是说当前状态信息独立于所有历史状态信息。

现在我们正式定义一下马尔科夫性质。为了使数学表示尽可能的简单，我们假设有有限的状态和奖励值。这样我们就能求和和概率的数学语言来表示，而不需要用积分和概率分布。在通常情况下，当前的状态取决于所有已经发生过的历史状态，因此我们可以定义如下：

$$Pr(R_{t+1}=r, S_{t+1}=s' | S_0, A_0, R_1, ..., S_{t-1}, A_{t-1}, R_t, S_t, A_t)$$

那么如果状态信息满足马尔科夫性质，那么t+1时刻下的状态只取决于t时刻的状态和动作，因此我们有可以用另一种形式表示：

$$Pr(R_{t+1}=r, S_{t+1}=s' | S_t, A_t)$$

如果状态信息满足马尔科夫性质，那么上面两式一定相等，反之亦然。

如果环境具有马尔科夫性质，那么我们就能根据当前的状态和动作预测下一时间点的状态和期望的reward。于是乎，如果我们有了当前的状态和动作信息，通过不断迭代上面的相等关系，我们就可以把所有将来的状态和期望奖励值都算出来了。马尔科夫性质告诉我们，依据当前信息做出的最优选择，也就是依据所有历史信息能做出的最优选择。

然而，就算是有些情况下环境状态并非严格遵守马尔科夫性质，其实影响也并不是那么重要，依然可以当成马尔科夫性质来处理。

马尔科夫性质在强化学习中之所以那么重要，是因为我们假定agent做出的决策和该决策的价值（value）是当前状态信息的函数。为了让这个假设成立，我们必须让当前的状态是具有充分信息性的（也就是能完全足够代表历史信息做出决策）。在本书中的所有状态都建立在具有马尔科夫性质的假设上。然而，基于马尔科夫性质所研究的算法，也能帮助我们解决一些非严格遵守马尔科夫性质的应用中。最后要说的是，马尔科夫性质的假设，并非仅仅出现在强化学习领域，而是在很多人工智能领域都有涉及。

作者的话：这一篇结尾主要提了一下马尔科夫性质，下一篇我们将依据一个具体例子来详细阐述马尔科夫过程，以及后面几节关于价值函数、贝尔曼方程的内容。