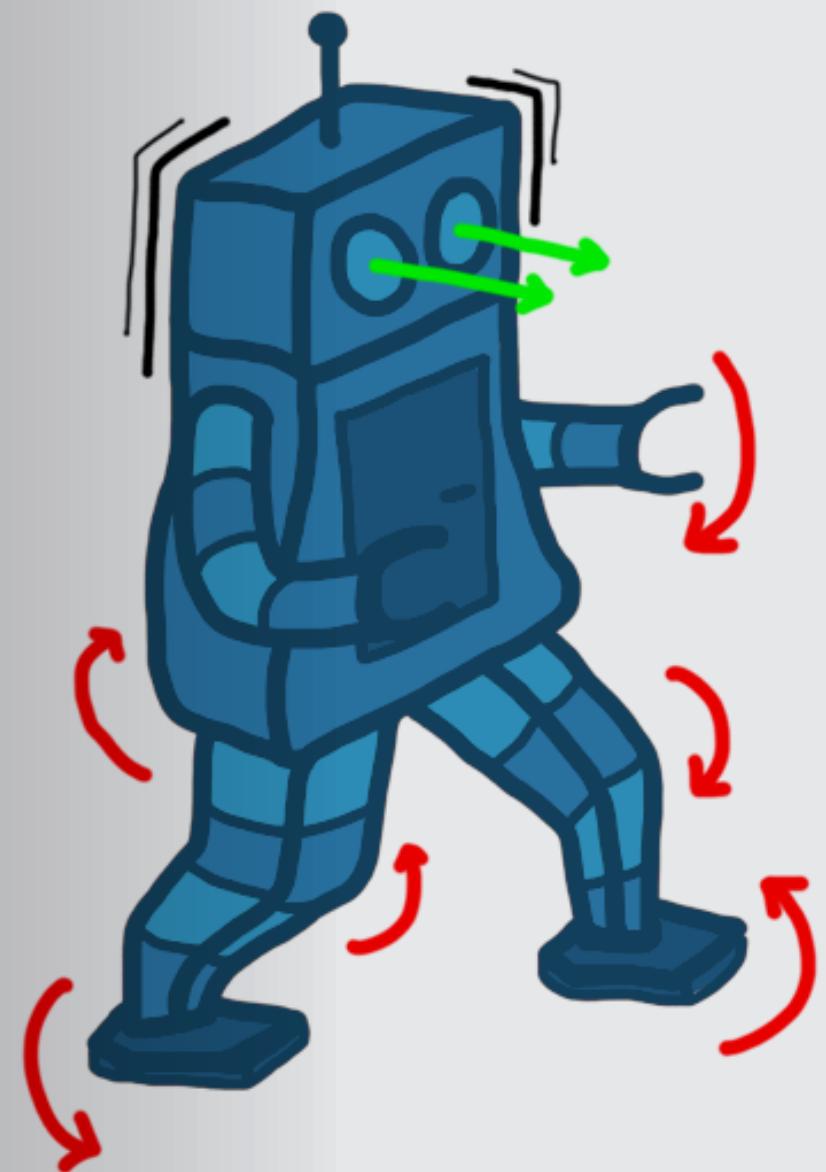


# agent

使用 MATLAB 进行强化学习

了解基础知识并设置环境

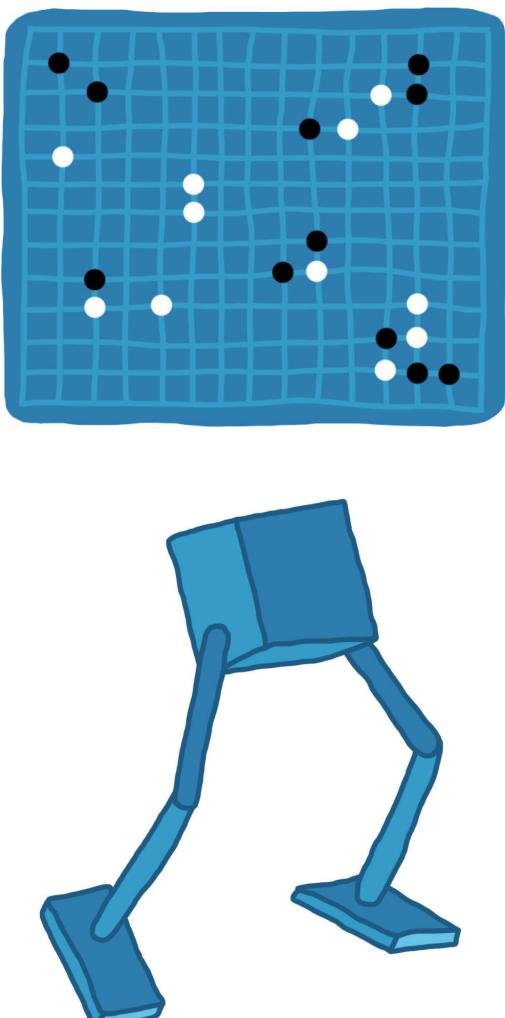
# environment



# 什么是强化学习?

强化学习旨在学习如何做,即如何根据情况采取动作,从而实现数值奖励信号最大化。学习者不会接到动作指令,而是必须自行尝试去发现回报最高的动作方案。

—Sutton and Barto, [强化学习:简介](#)

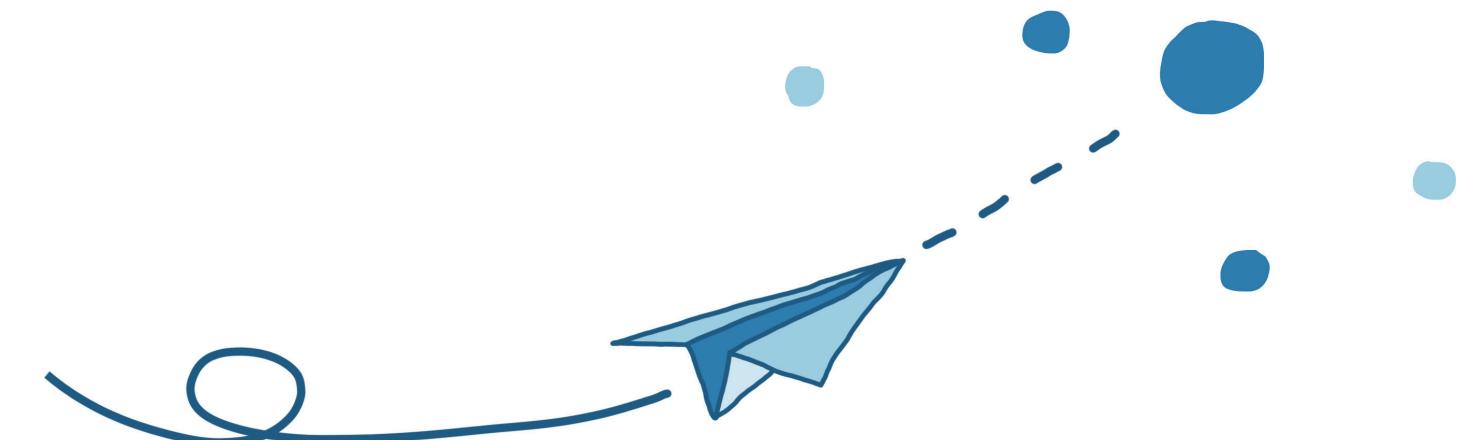


强化学习 (RL) 已成功地训练计算机程序在游戏中击败全球最厉害的人类玩家。

在状态和动作空间较大、环境信息不完善并且短期动作的长期回报不确定的游戏中,这些程序可以找出最佳动作。

在为真实系统设计控制器的过程中,工程师面临同样的挑战。另外,强化学习能否帮助解决复杂的控制问题,例如训练机器人走路或驾驶自动驾驶汽车?

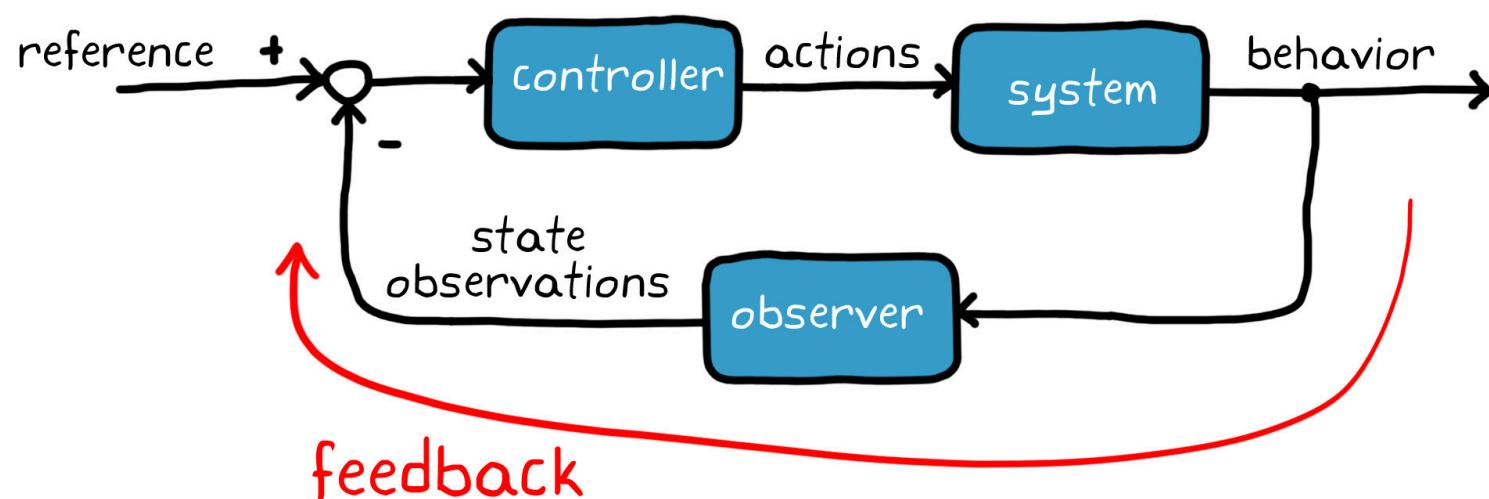
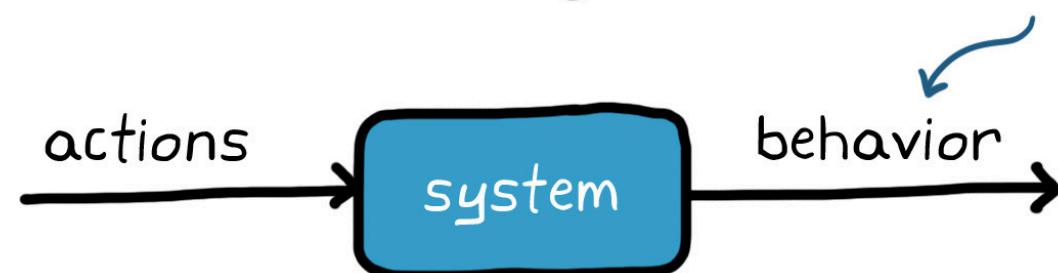
本电子书通过在传统控制问题的语境下解读什么是强化学习,帮助您了解如何设置和解决 RL 问题。



# 控制目标

从广义上而言,控制系统的目地是确定生成期望的系统行为的正确系统输入(动作)。

which actions generate the desired behavior?



在反馈控制系统中,控制器使用状态观测提高性能并修正随机干扰。工程师运用反馈信号,以及描述被控对象和环境的模型,设计控制器,从而满足系统需求。

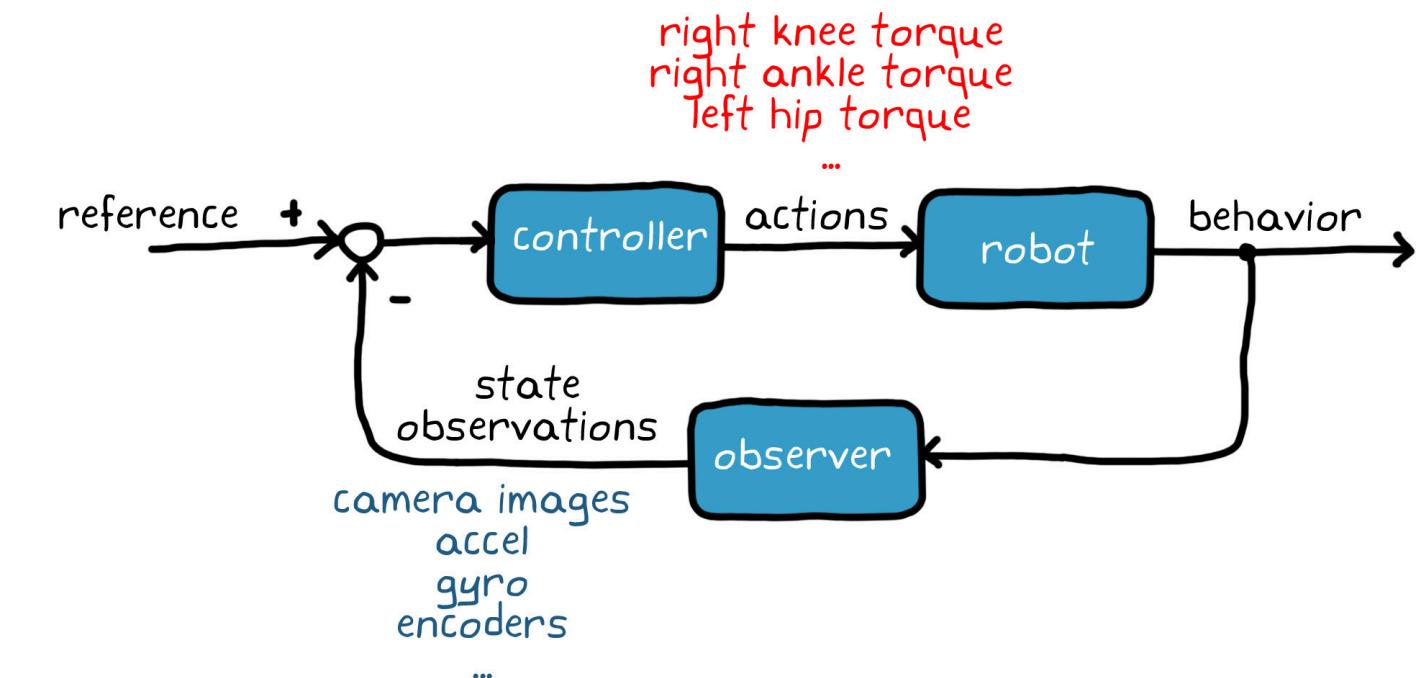
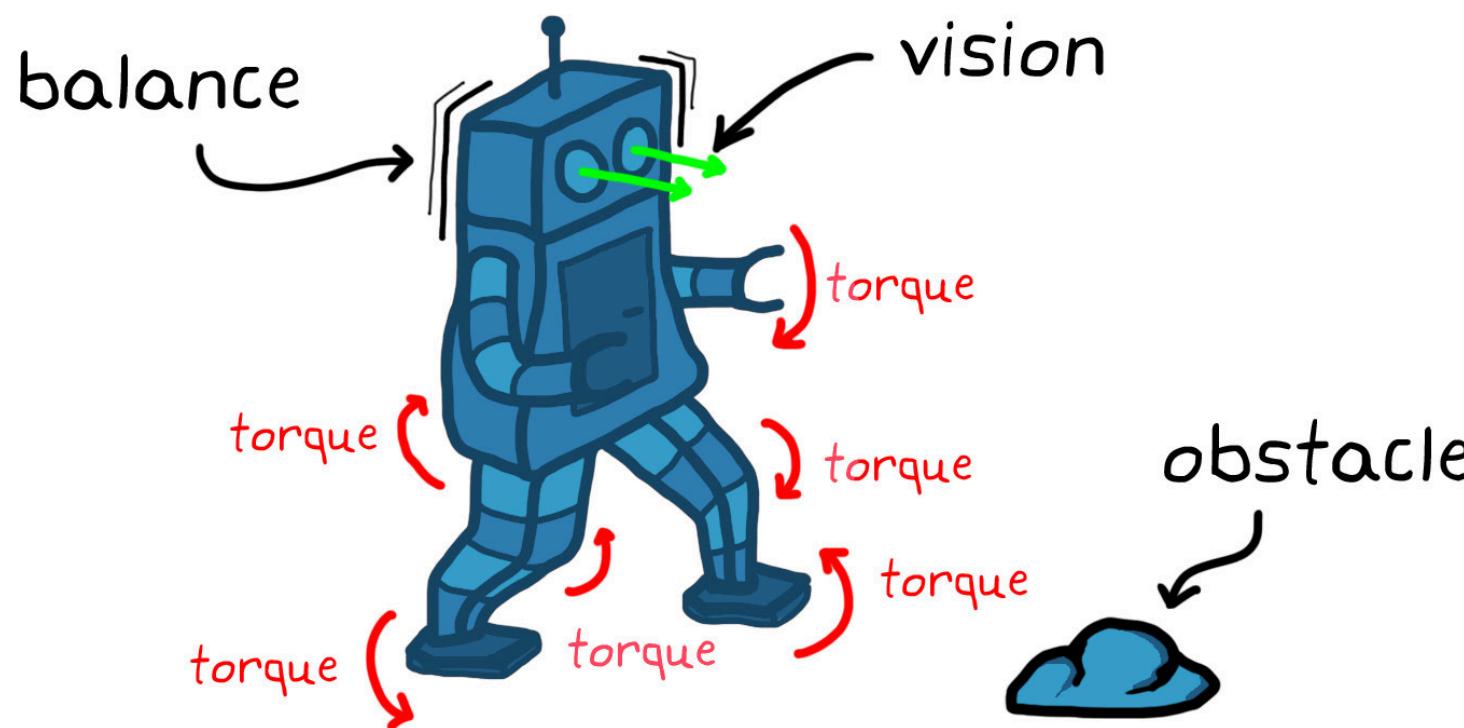
以上概念表述十分简单;然而,倘若系统难以建模、高度非线性或者状态和动作空间较大,则很难实现控制目标。

# 控制问题

为了理解此类难题对控制设计问题造成进一步后果，不妨设想一下开发步行机器人控制系统的场景。

要控制机器人（即系统），可能需要指挥数十台电机操控四肢的各个关节。

每一项命令是一个可执行的动作。系统状态观测量有多种来源，包括摄像机视觉传感器、加速度计、陀螺仪及各电机的编码器。



控制器必须满足多项要求：

- 确定适当的电机扭矩组合，确保机器人正常步行并保持躯体平衡。
- 在需要避开多种随机障碍物的环境下操作。
- 抗干扰，如阵风。

控制系统设计不仅要满足上述要求，还需满足其他附加条件，比如在陡峭的山坡或冰块上行走时保持平衡。

# 控制方案

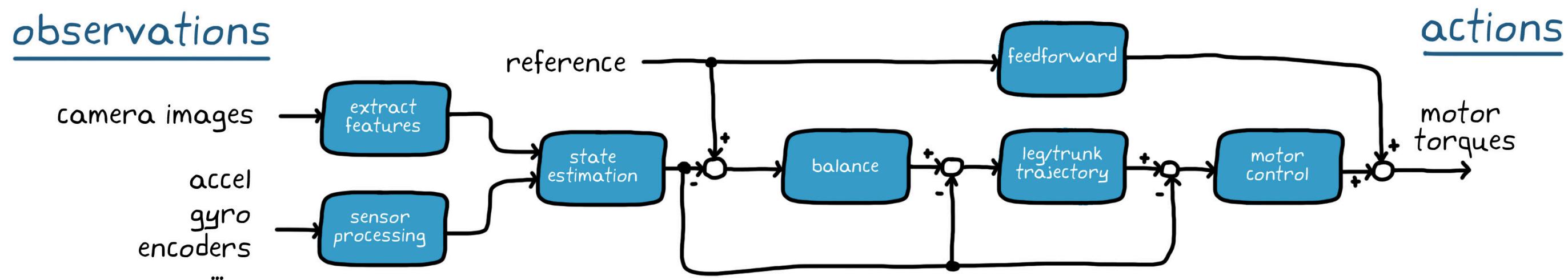
通常,解决此类问题的最佳方法是将问题分解成为若干部分,逐个击破。

例如,您可以构建一个提取摄像机图像特征的流程。比方说,障碍物的位置和类型,或者机器人在全局参照系中所处的位置。综合运用这些状态与其他传感器传回的处理后的观测值,完成全状态估测。

估算的状态值和参考值将馈送至控制器,其中很可能包含多个嵌套控制回路。外部环路负责管理高级机器人行为(如保持平衡),内部环路用于管理低级行为和各个作动器。

所有问题都解决了吗?那可未必。

各环路之间相互交互,使得设计和调优变得异常困难。同时,确定最佳的环路构造和问题分解也并不轻松。

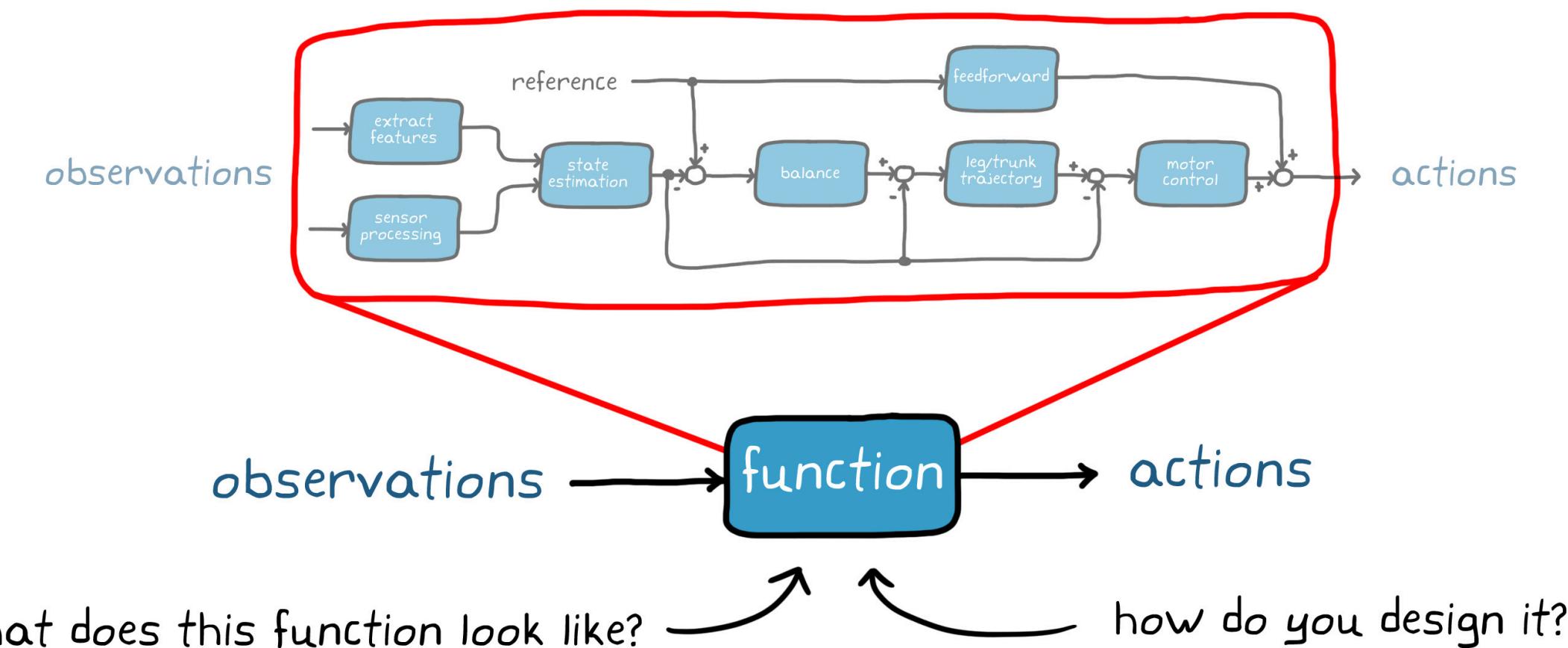


# 强化学习的魅力

不是尝试单独设计每一个组件，而是设想一下将其全部塞进一个函数里，由该函数负责接收所有观察结果并直接输出低级动作。

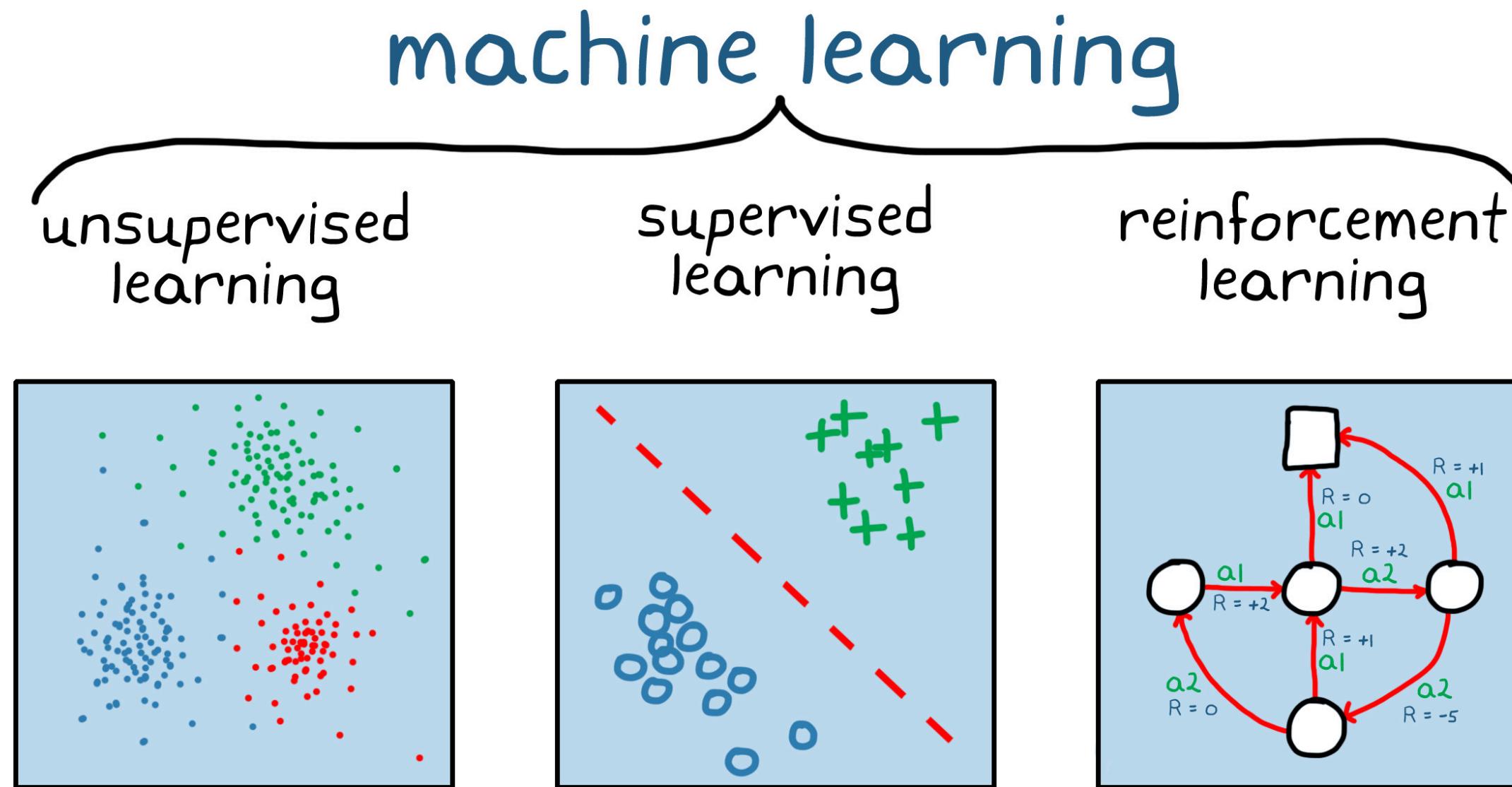
毋庸置疑，这可以简化系统方块图，但这个函数会是怎样的结构？你该如何设计这个函数呢？

创建一个单一的大函数比构建由分段子组件构成的控制系统，看起来难度要大；不过，强化学习可以助您达成目标。



# 强化学习:机器学习的子集

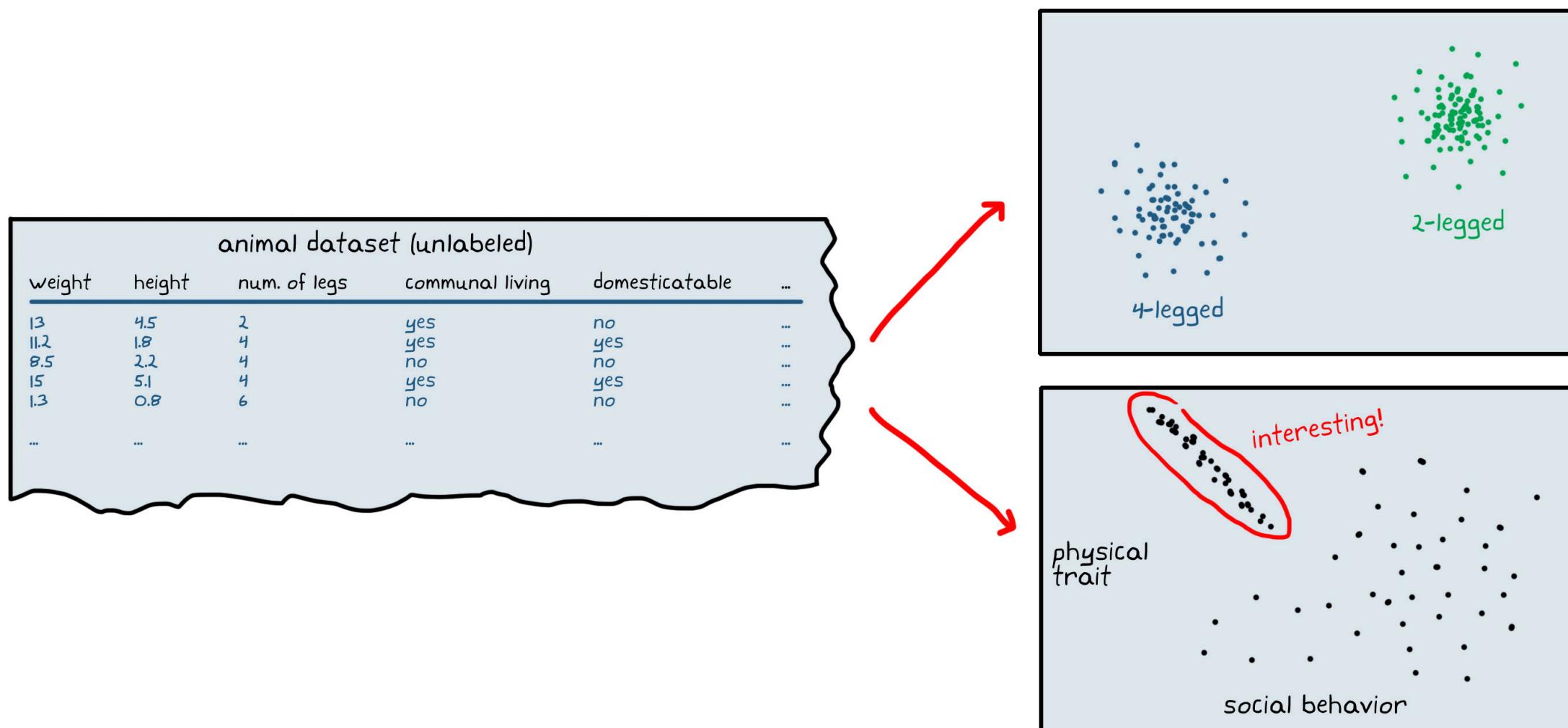
强化学习是机器学习的三个大类之一。无监督学习和监督式学习并不是本电子书关注的重点。但是理解强化学习与这二者的区别是值得的。



# 机器学习:无监督学习

无监督学习用于确定尚未被分类或标注的数据集的模式或隐藏结构。

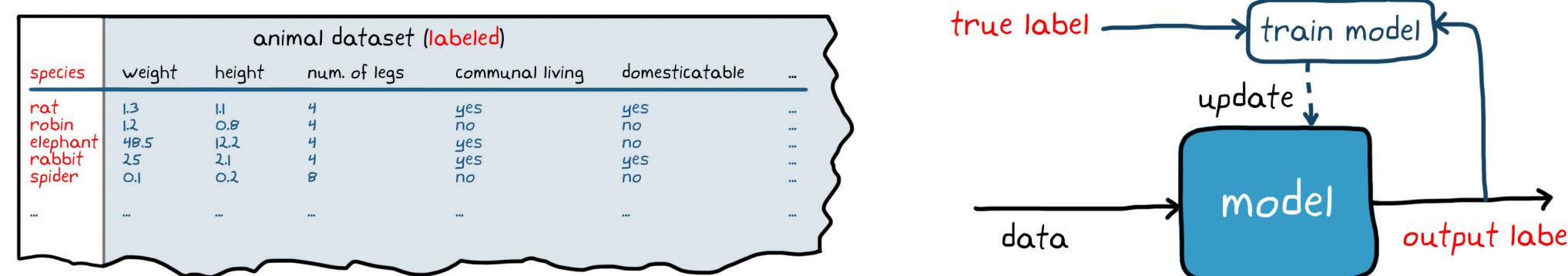
例如,假设您收集了 100,000 种动物的生理特征和社会倾向性信息。您可以使用无监督学习进行动物分组或总结相似特征。可以根据腿数进行分组,也可以根据不太显著的模式进行分组,例如,之前并不知道的生理特性和社会行为之间的关联性。



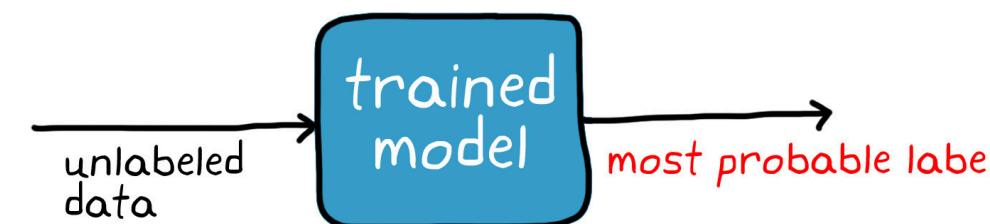
# 机器学习：监督式学习

您可以使用监督式学习训练计算机为给定输入加上标签。例如，如果动物特征数据集的其中一列是物种，则可以将物种作为标签，其余数据作为数学模型的输入。

您可以使用监督式学习训练模型，使其能够根据每一组动物的特征正确标记数据集。先由模型推断物种，再由机器学习算法系统性地调整模型。



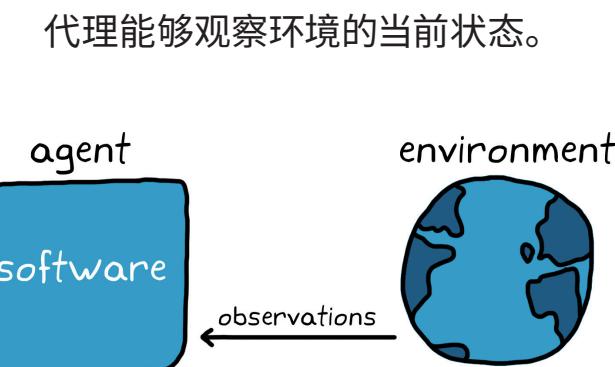
运用足够的训练数据获得可靠的模型后，再输入未标注的新动物的特征，经过训练的模型即能给出对应最有可能的物种标签。



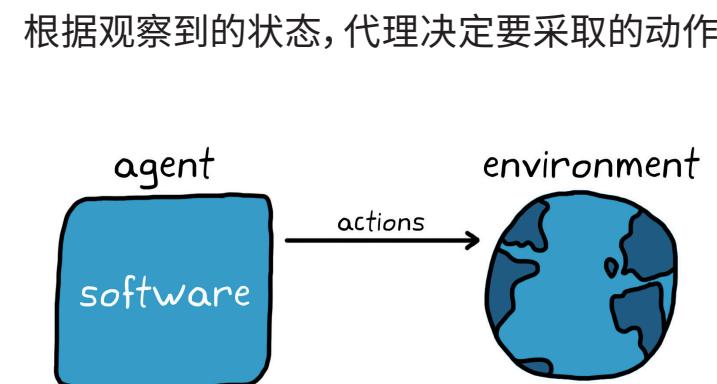
# 机器学习：强化学习

强化学习是一种截然不同的方法。不同于另外两种采用静态数据集的学习框架，RL 采用动态环境数据。其目标并不是对数据进行分类或标注，而是确定生成最优结果的最佳动作序列。为了解决这个问题，强化学习通过一个软件（即所谓的代理）来探索环境、与环境交互并从环境中学习。

1

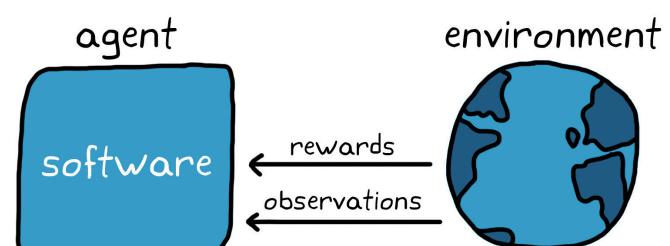


2



3

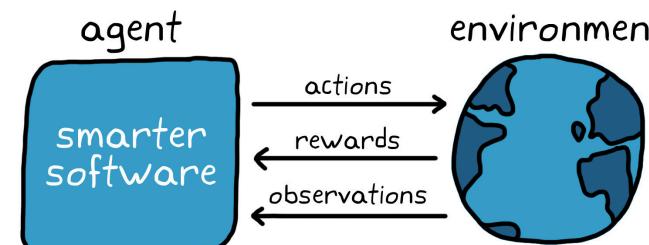
环境会改变状态并针对该动作生成奖励。  
代理接收状态变化和动作奖励。



4

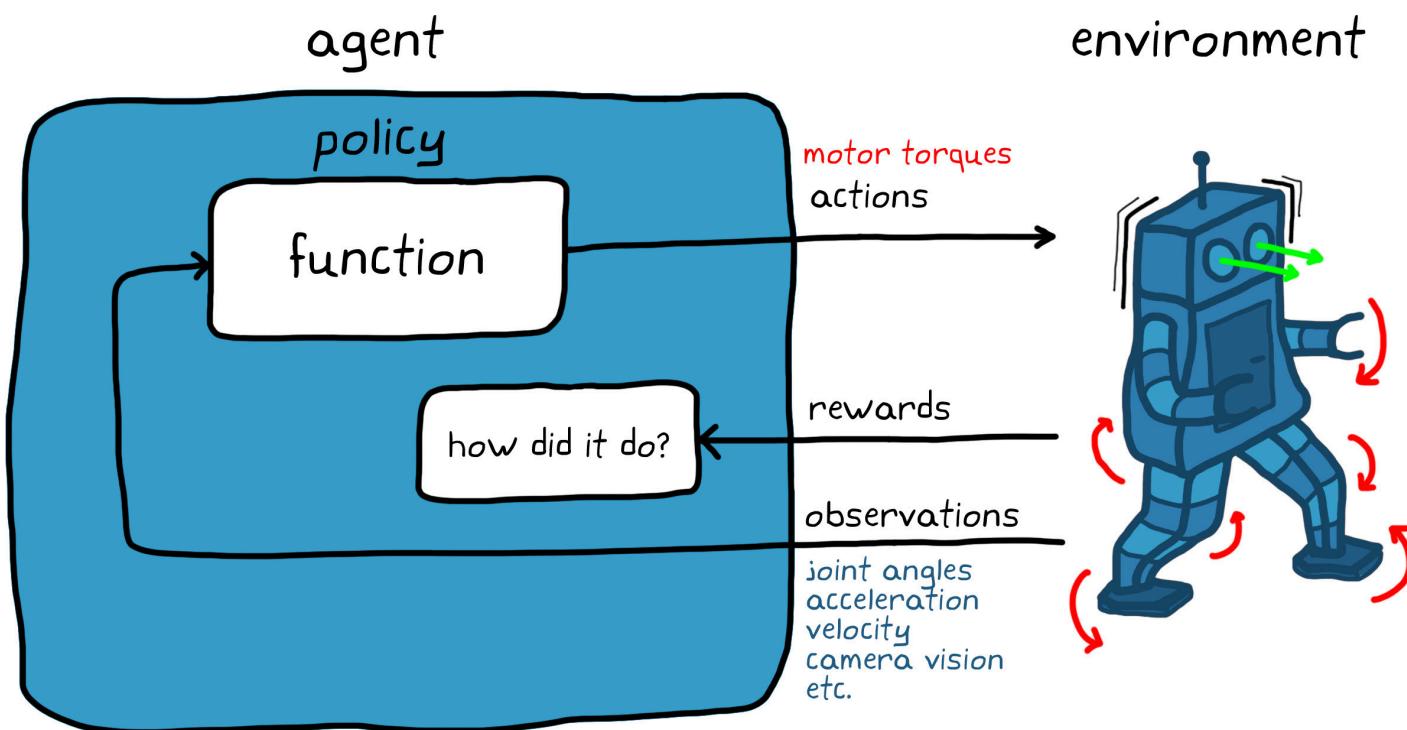
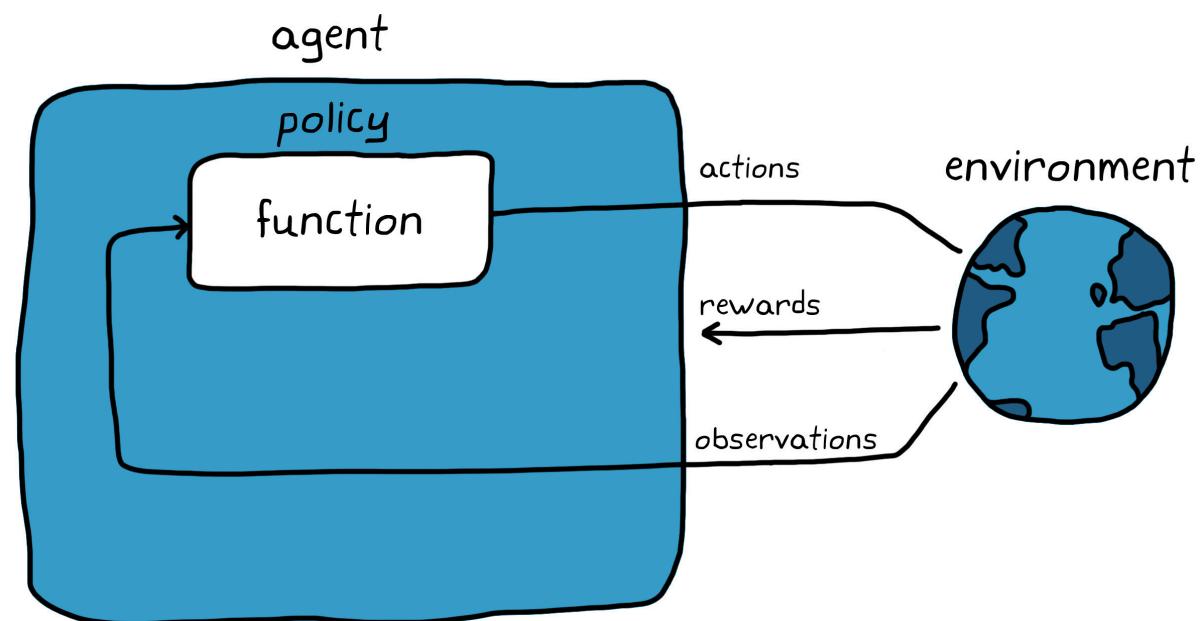
通过这些新信息，代理可以确定该动作是有用且应该重复，还是无益而应该避免。

观察-动作-奖励这个循环会一直持续，直至完成学习。



# 剖析强化学习

代理中有一个函数可接收状态观测量(输入),并将其映射到动作集(输出)。也就是前面讨论过的单一函数,它将取代控制系统的所有独立子组件。在RL命名法中,此函数称之为策略。策略根据一组给定的观测量决定要采取的动作。



以步行机器人为例,观察结果是指每个关节的角度、机器人躯干的加速度和角速度,以及视觉传感器采集的成千上万个像素点。策略将根据所有这些观测量,输出电机指令,使机器人移动四肢。

接着,环境将生成奖励,向代理反映特定作动器指令组合的效果。如果机器人能够保持直立并继续行走,则对应的奖励将高于机器人摔倒时的奖励。

# 学习最优策略

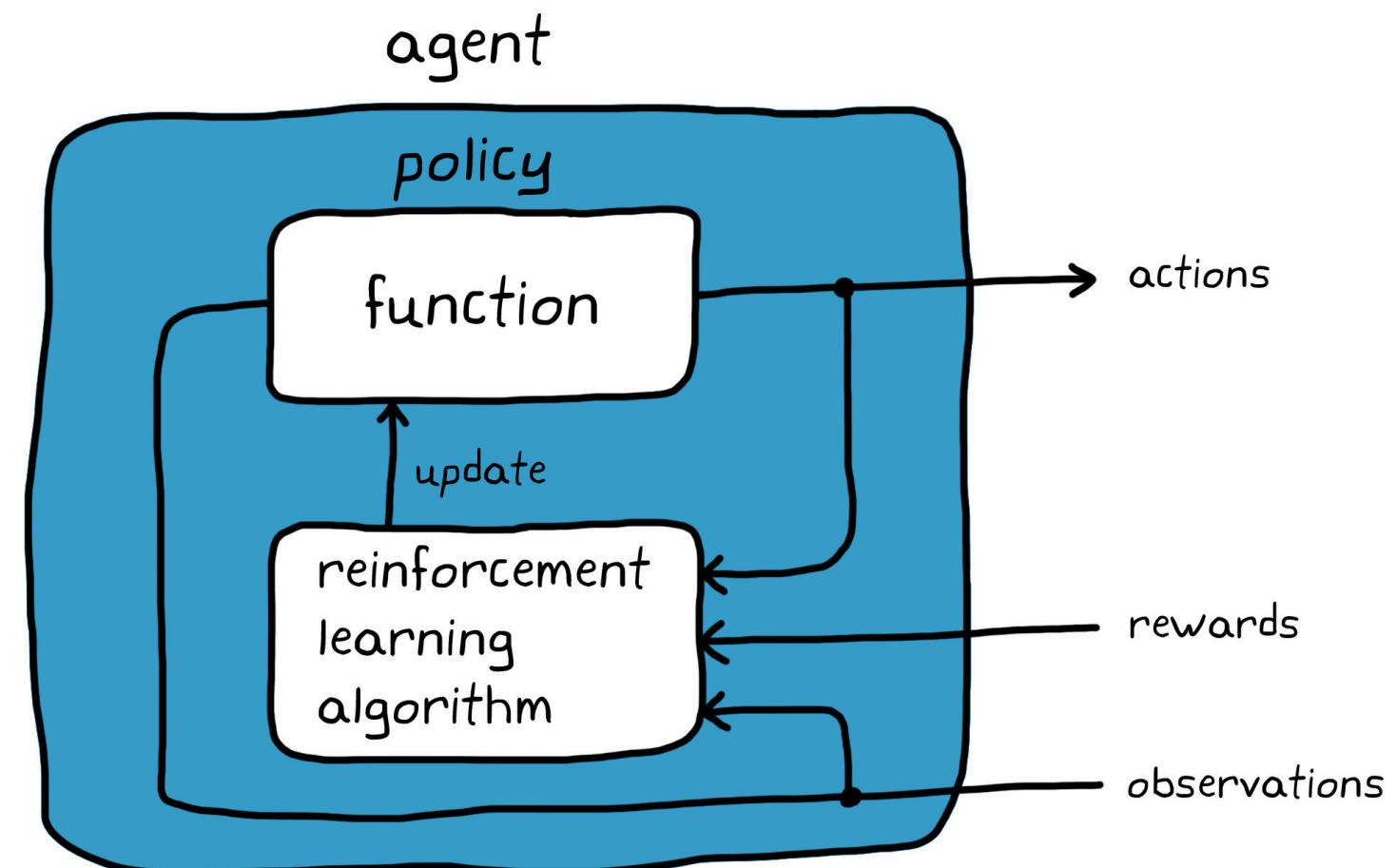
如果可以设计出一项完美的策略,针对观察到的每一种状态向适当的作动器发出适当的指令,那么目标就达成了。

当然,大多数情况下并非如此。即便你真的找到了完美的策略,环境也可能不断变化,因而静态映射不再是最优方案。

正因为如此,强化学习算法应运而生。

它可以根据已采取的动作、环境状态观测量以及获得的奖励值来改变策略。

代理的目标是使用强化学习算法学习最佳环境交互策略;这样一来,无论在任何状态下,代理都能始终采取最优动作—即长期奖励最丰厚的动作。



# 什么是学习?

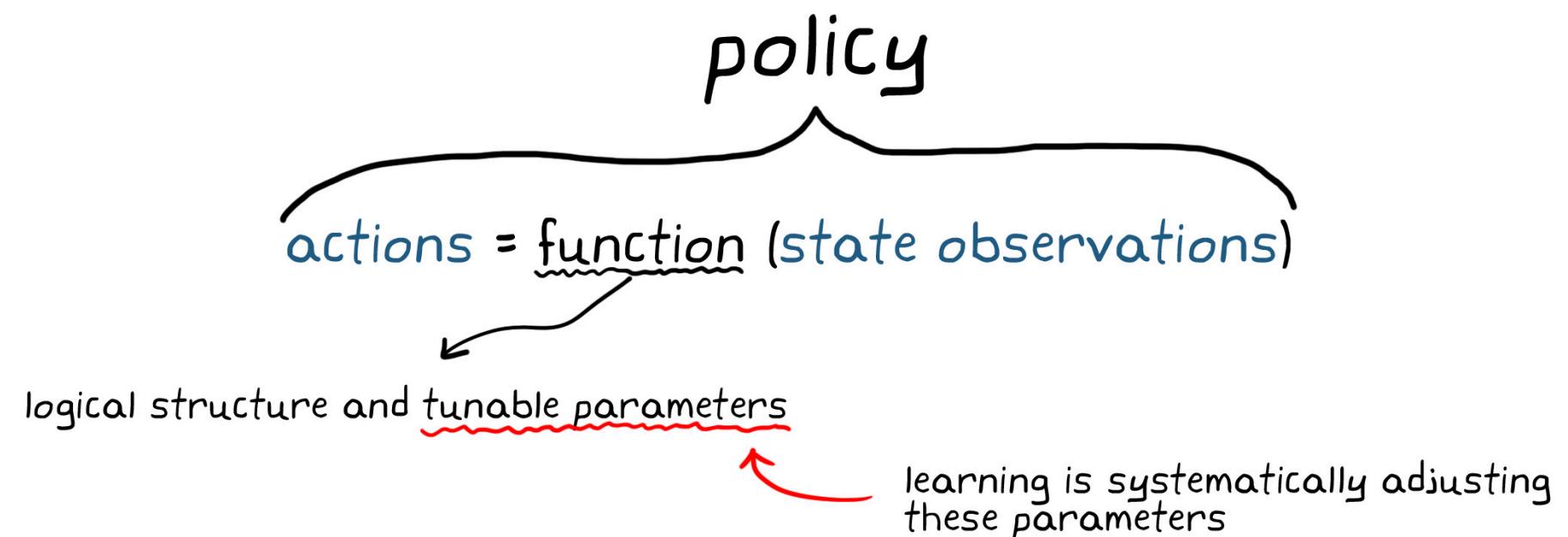
为了理解机器如何学习,请思考一下策略的含义:一个由逻辑和可调参数构成的函数。

倘若已有一套完善的策略结构(逻辑结构),对应有一组参数可生成最优策略,即可产生最丰厚的长期奖励的状态-动作的映射。

学习是指系统性调整这些参数以收敛到最优策略的过程。

这样,您将可以专注于设置适当的策略结构,而无需手动调整函数来获取确切的参数。

您可以让计算机通过稍后将要介绍的流程自行学习参数,但在现阶段,您可以将该流程视为一种复杂的试错过程。



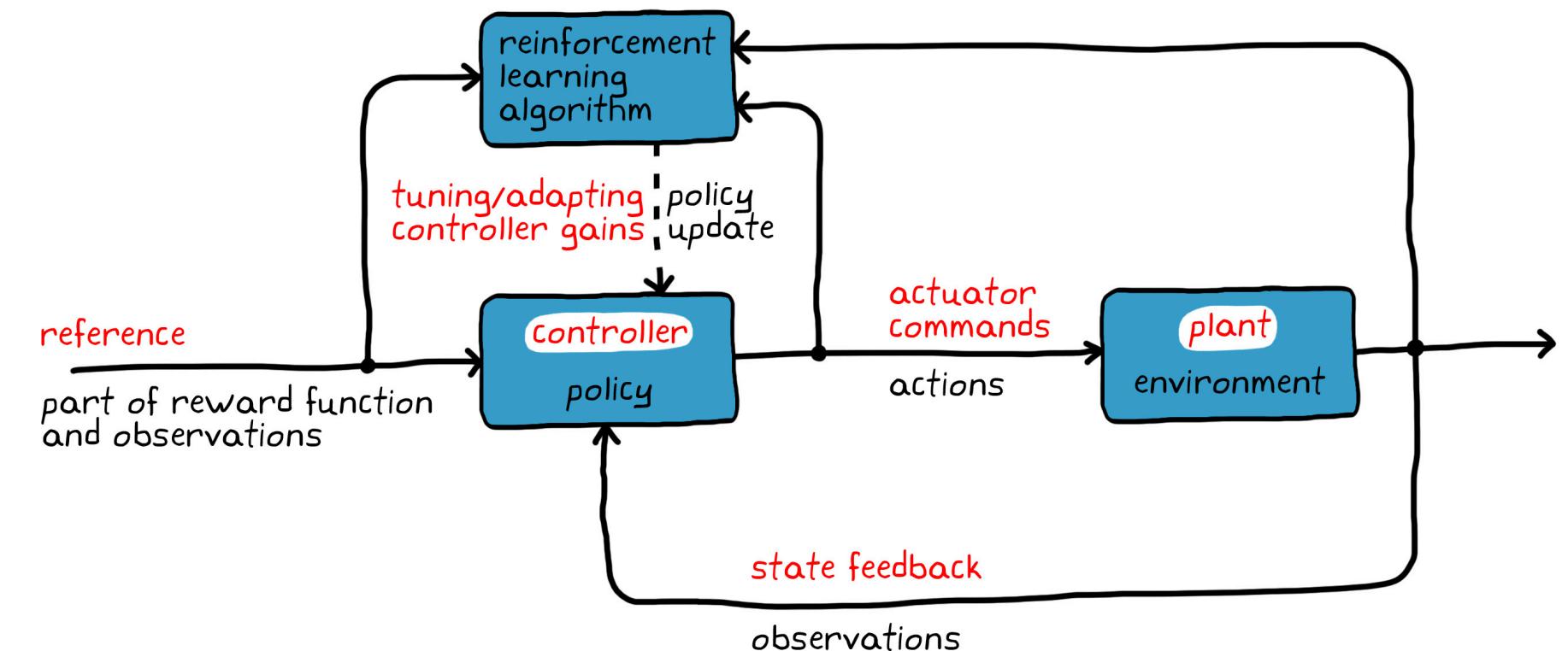
# 强化学习与传统控制有着怎样的相似之处？

强化学习的目标与控制问题相似；只不过方法不同，使用不同的术语表示相同的概念。

通过这两种方法，您希望确定正确的系统输入，以让系统产生期望的行为。

您的目的在于判断如何设计策略（或控制器），从而将环境（或被控对象）的状态观测量映射到最佳动作（作动器指令）。

状态反馈信号是指环境观察结果，参考信号则内置到奖励函数和环境观测量中。



# 强化学习工作流程概述

一般来说，强化学习涉及到五个方面。本电子书重点介绍第一个部分：建立环境。本系列的其他电子书将更深入地探索奖励、策略、训练和部署问题。

1

您需要一个环境，供您的代理开展学习。您需要选择环境里应该有什么，是仿真还是物理设置。

*environment*



2

您需要考虑最终想要代理做什么工作，并设计奖励函数，激励代理实现目标。

*reward*



3

您需要选择一种表示策略的方法。思考您想如何构造参数和逻辑，由此构成代理的决策部分。

*policy*



4

您需要选择一种算法来训练代理，争取找到最优的策略参数。

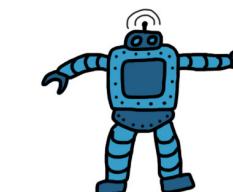
*training*



5

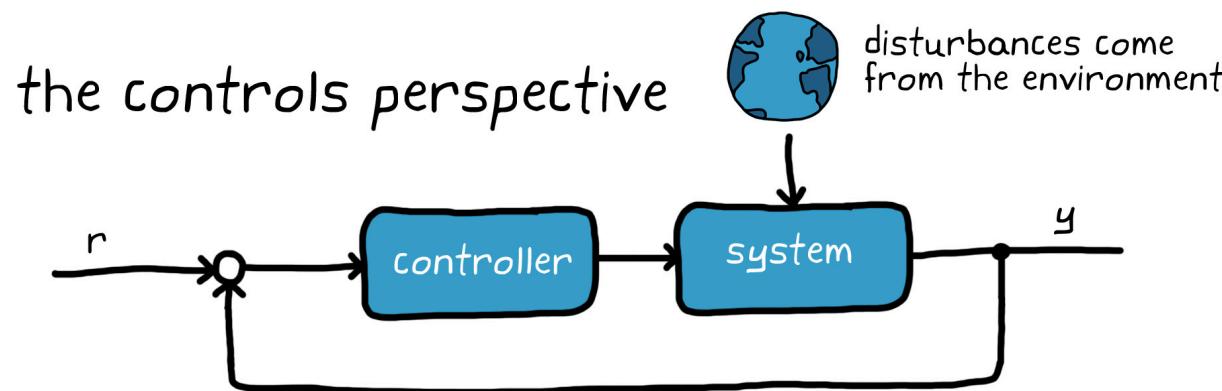
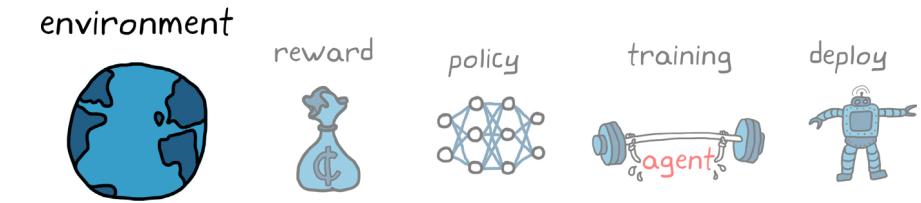
最后，您需要在实地部署该策略并验证结果，从而利用该策略。

*deploy*



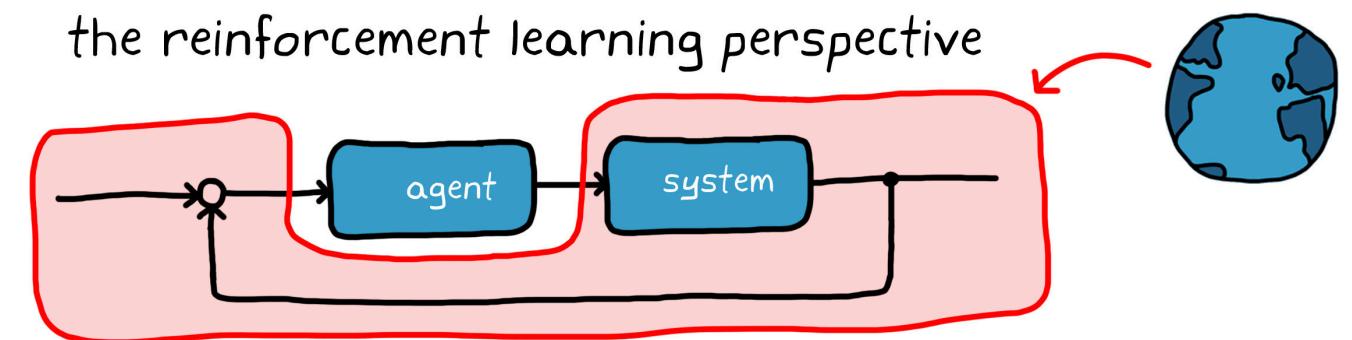
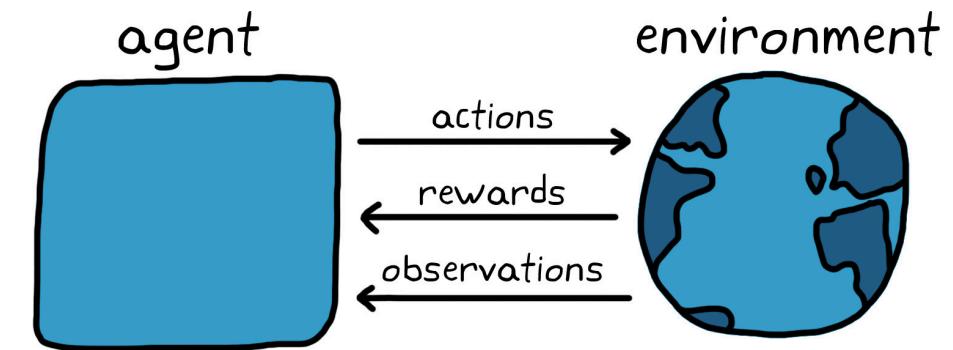
# 环境

环境是指存在于代理之外的一切元素。它既是代理动作产生作用的地方，又能生成奖励和观测量。



然而，在强化学习的术语中，环境是指除代理以外的一切元素，包括系统动态特性。因此，控制系统的一大部分实际上都属于环境。代理只不过是通过学习生成动作及更新策略的一个软件而已。

从控制的角度而言，这个定义可能令人费解，因为人们普遍将环境视为影响控制系统的干扰。

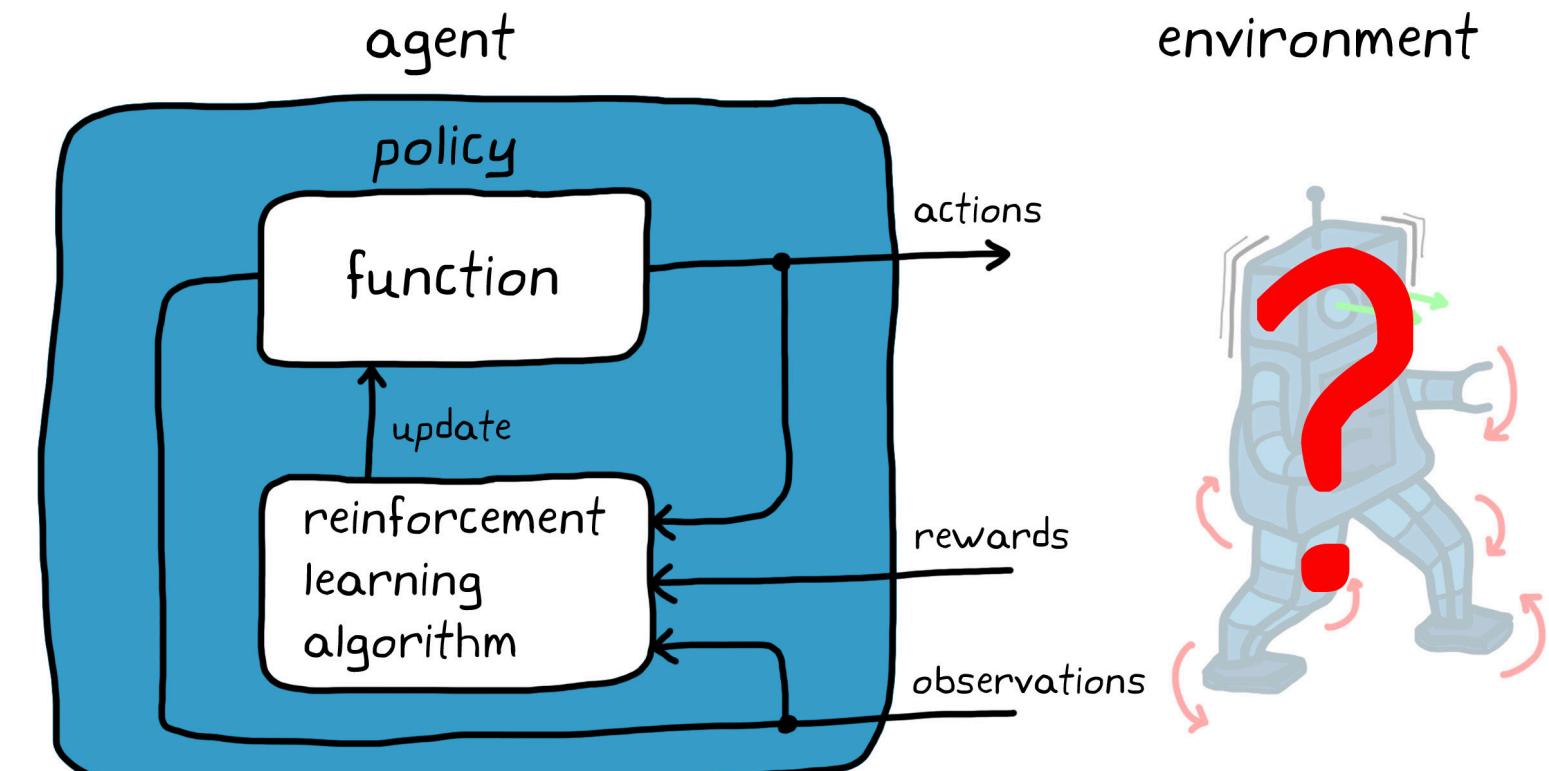
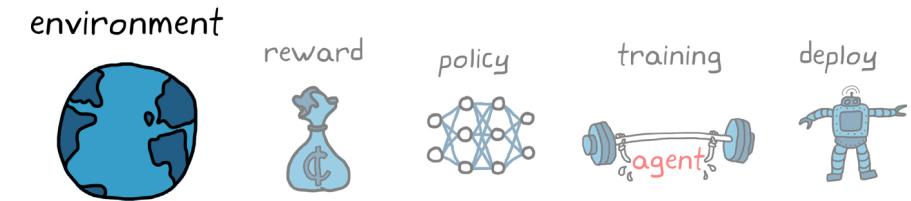


# 无模型强化学习

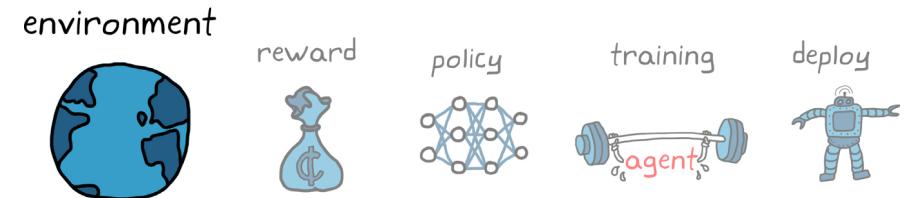
强化学习之所以功能强大,原因之一在于代理不需要对该环境有任何了解,但仍可学习如何与该环境交互。例如,代理不需要了解步行机器人的动力学或运动学原理,不必了解关节移动或附肢长度,却仍能确定如何获得最多的奖励。

这就是所谓的无模型强化学习。

在无模型 RL 中,您可以将采用 RL 的代理内置到任何系统,代理将能够学习最优策略。(假设您已给策略访问观测量、奖励、动作及足够的内部状态的权限。)

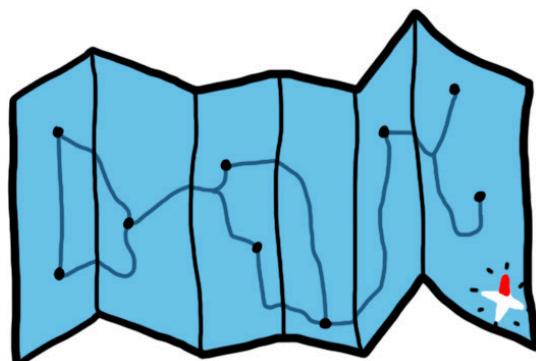


# 基于模型的强化学习



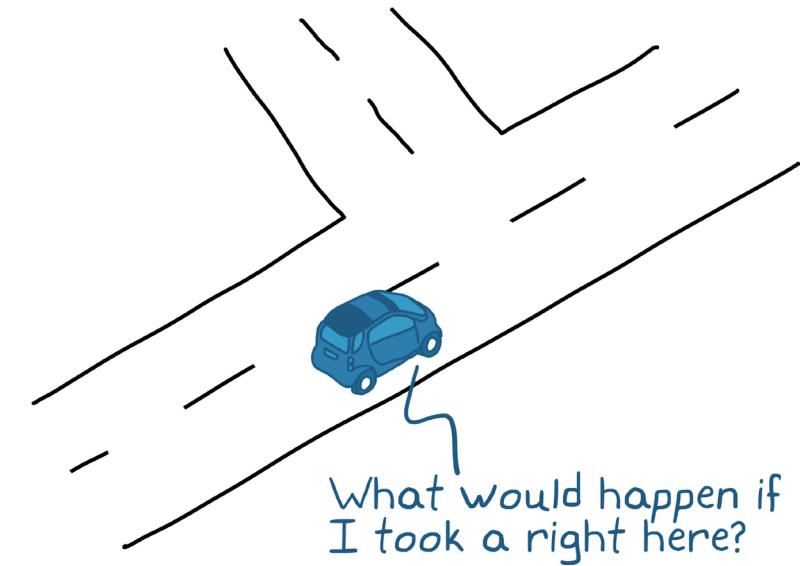
无模型 RL 面临一个问题：如果代理不了解环境，那么必须探索状态空间的所有区域，以确定如何获得最多的奖励。

这意味着，代理需要在学习过程中投入一些时间探索低奖励区域。

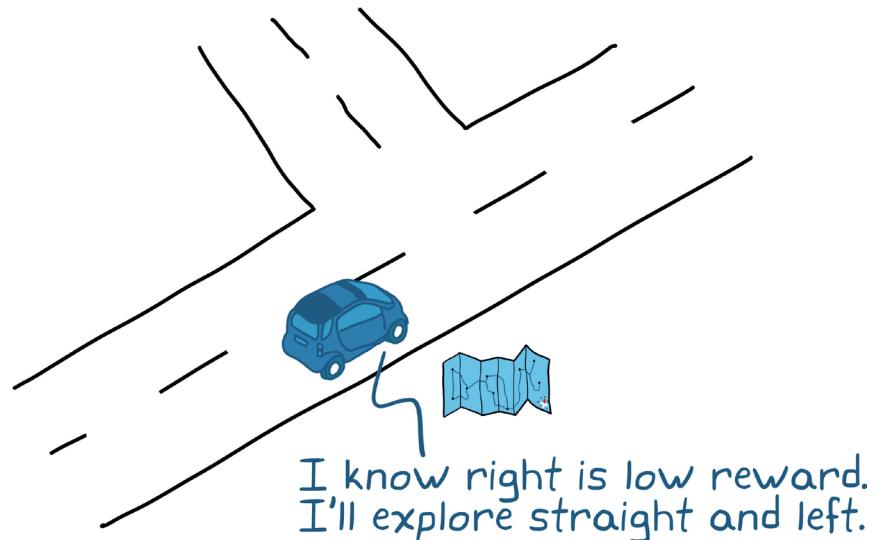


*this road map  
should help!*

代理可以使用模型探索环境的某些部分，而无需采取实际的动作。模型可以补足学习过程，使其避开已知的无益区域，而集中探索剩余部分。



但是，您可能已经知道，状态空间的某些区域不值得探索。通过提供整个环境或部分环境的模型，将已知的信息提供给代理。

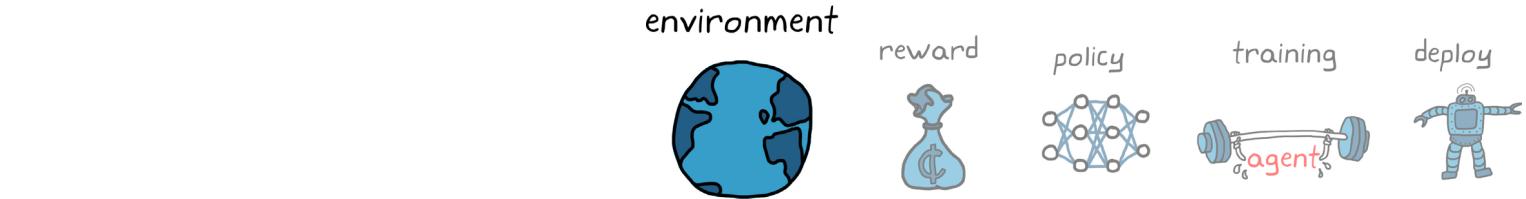


# 无模型与基于模型

基于模型的强化学习可以缩短学习最优策略所需的时间,因为您能够使用模型指导代理远离已知的低奖励状态空间区域。

首先,您不希望代理进入这些低奖励状态,所以无需浪费时间学习低奖励状态下的最佳动作。

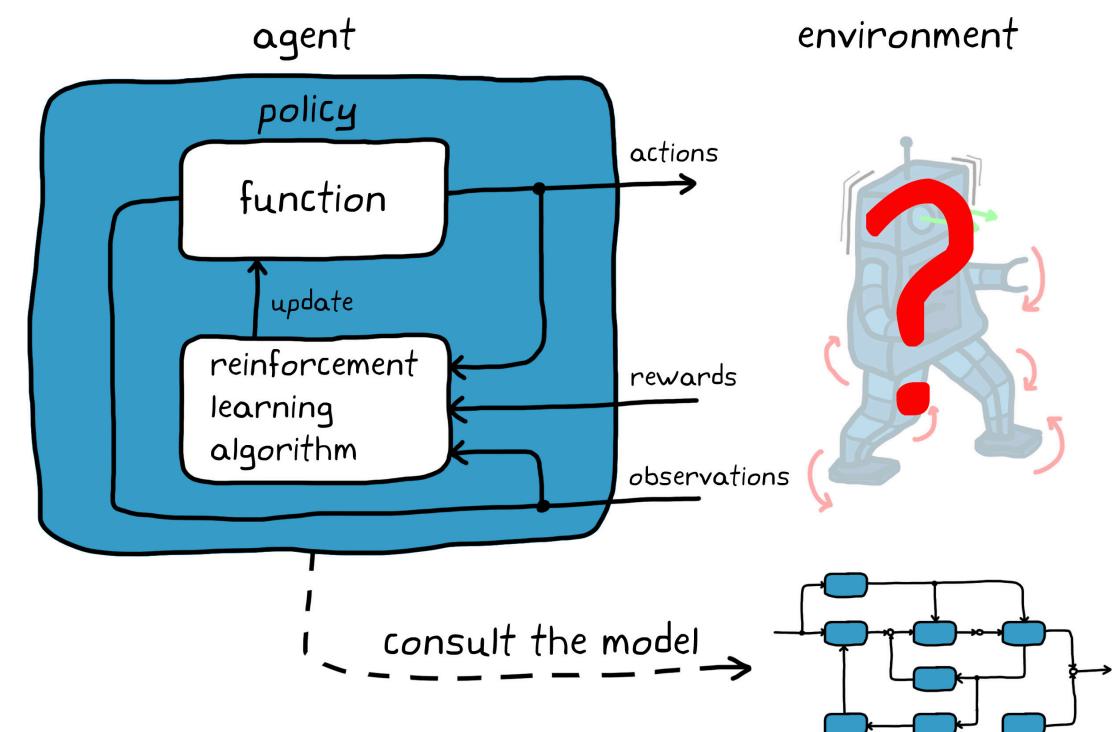
在基于模型的强化学习中,不需要了解整个环境模型;只需为代理提供您自己了解的那部分环境。



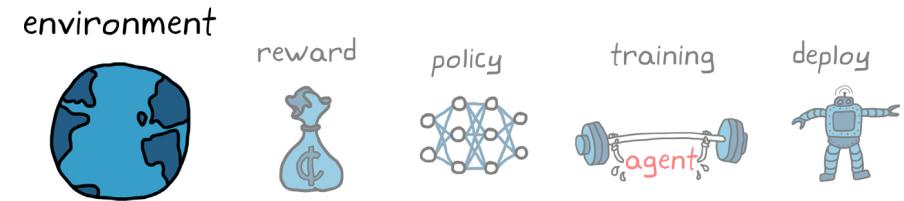
无模型强化学习应用更广泛,本电子书接下来将重点介绍这种方法。

如果您对无模型强化学习的基础知识有所了解,那么继续研究基于模型的 RL 也会更为直观。

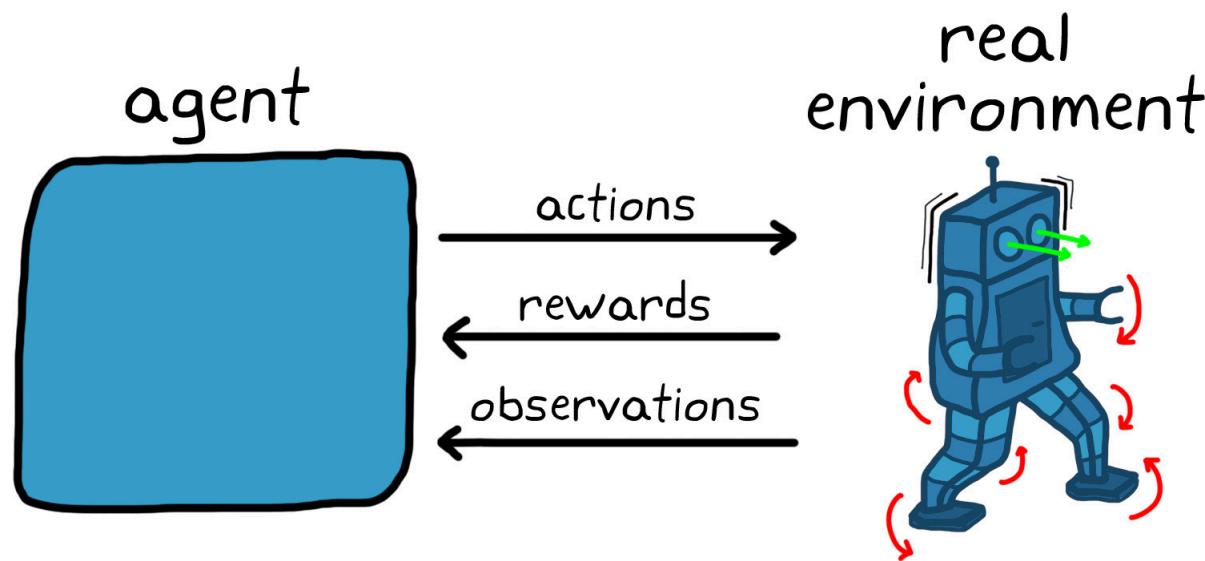
现阶段,无模型 RL 更受欢迎,因为人们希望通过它来解决一些难以开发模型(甚至是简单模型)的问题。例如,通过像素观测来控制汽车或机器人。在大多数情况下,像素强度与汽车或机器人动作之间的关系并不明显。



# 真实环境与仿真环境



鉴于代理通过与环境交互来开展学习，您需要设法使代理与环境进行实际交互。可以采用真实物理环境，也可以通过仿真，需根据具体情况加以选择。

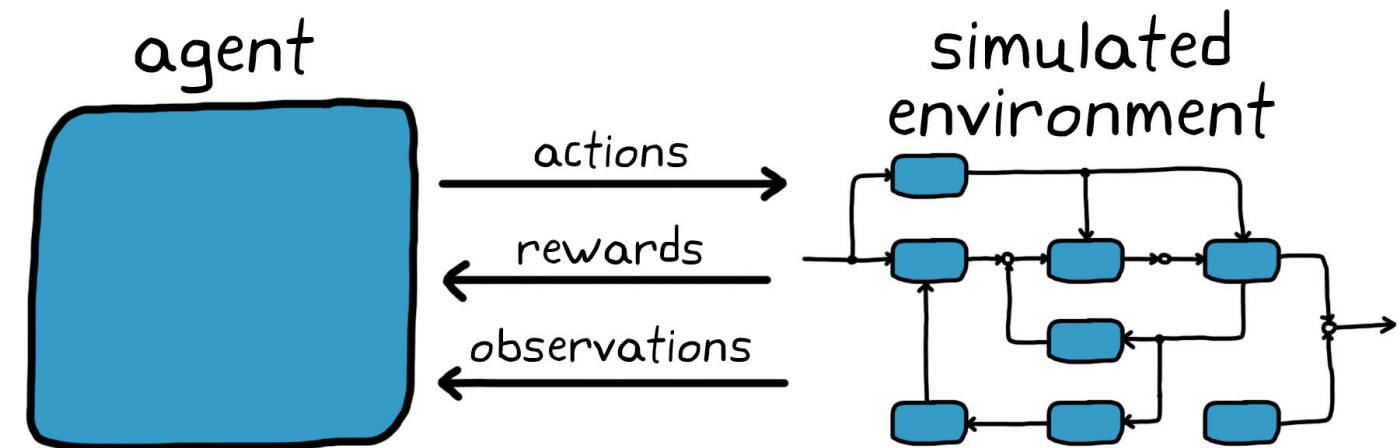


## 真实

**准确性**: 没有什么能比真实环境更全面地反映环境状态了。

**简便性**: 无需花时间创建和验证模型。

**必要性**: 如果真实环境不断变化或难以准确建模，可能需要根据该真实环境进行训练。



## 仿真

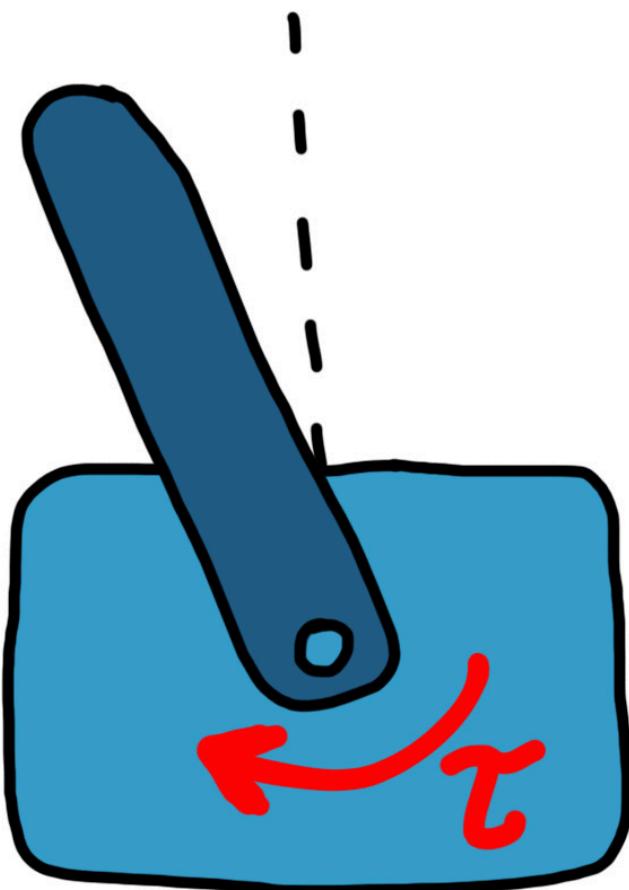
**高速性**: 仿真的运行速度可能高于实时环境或可并行化，从而可以加快缓慢的学习过程。

**仿真条件**: 对于难以测试的情况，建模更容易。

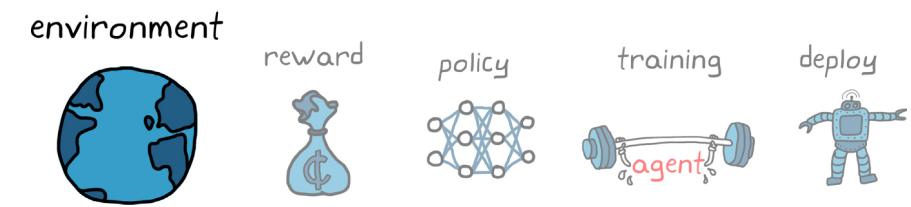
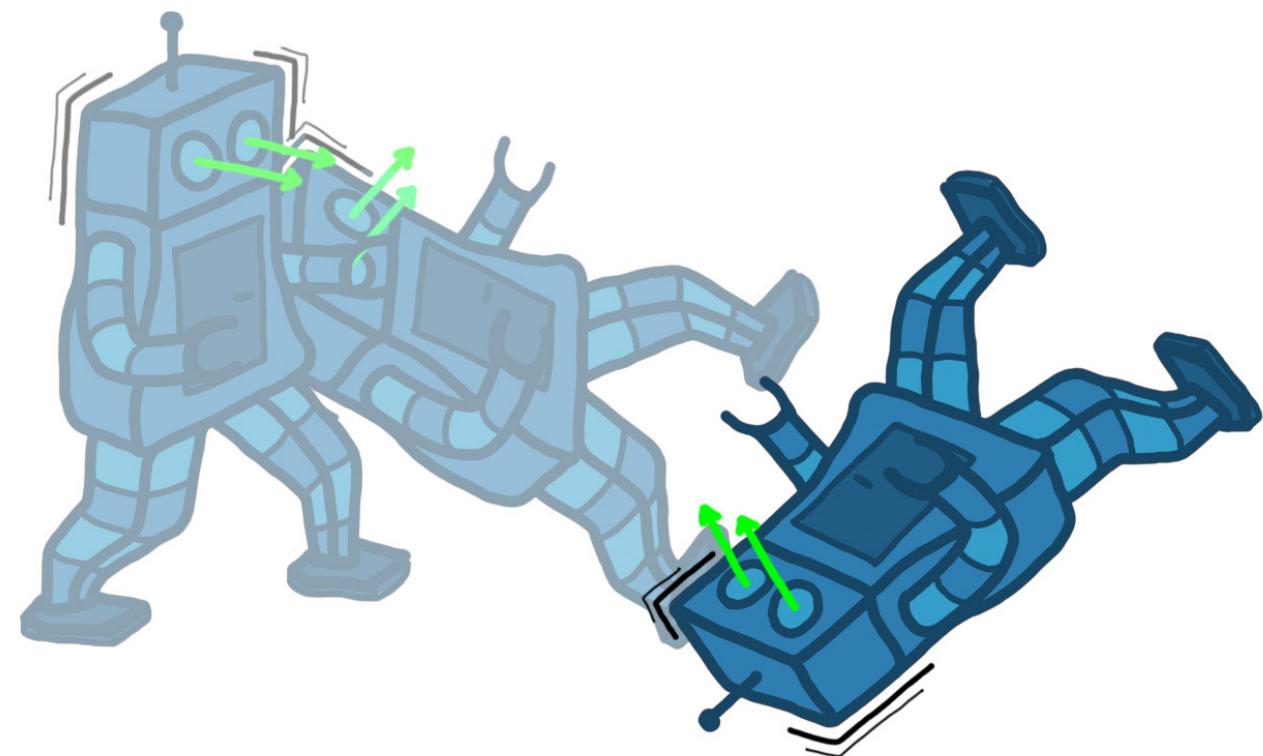
**安全性**: 不存在损坏硬件的风险。

# 真实环境与仿真环境

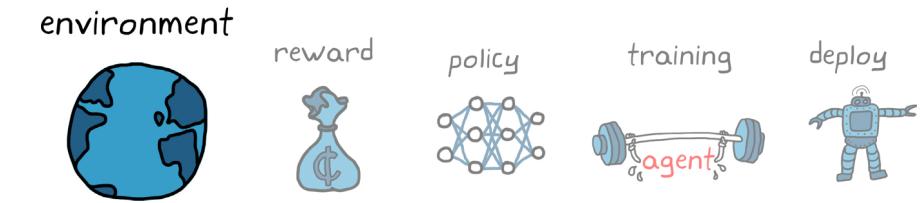
例如，您可以通过运行真实的物理系统，让代理学习如何平衡倒立摆。这应该是个不错的解决方案，因为硬件损坏的情况不大可能发生。鉴于状态和动作空间相对较小，训练大概不会花费太长时间。



但若是训练步行机器人，这种方法可能并不那么奏效。如果开始训练时策略不够理想，机器人会不停地摔倒或踉跄，以至于无法学会移动双腿，更不用说学习如何走路了。这不仅会损坏硬件，而且每次还得扶起机器人，相当耗时。效果不理想。



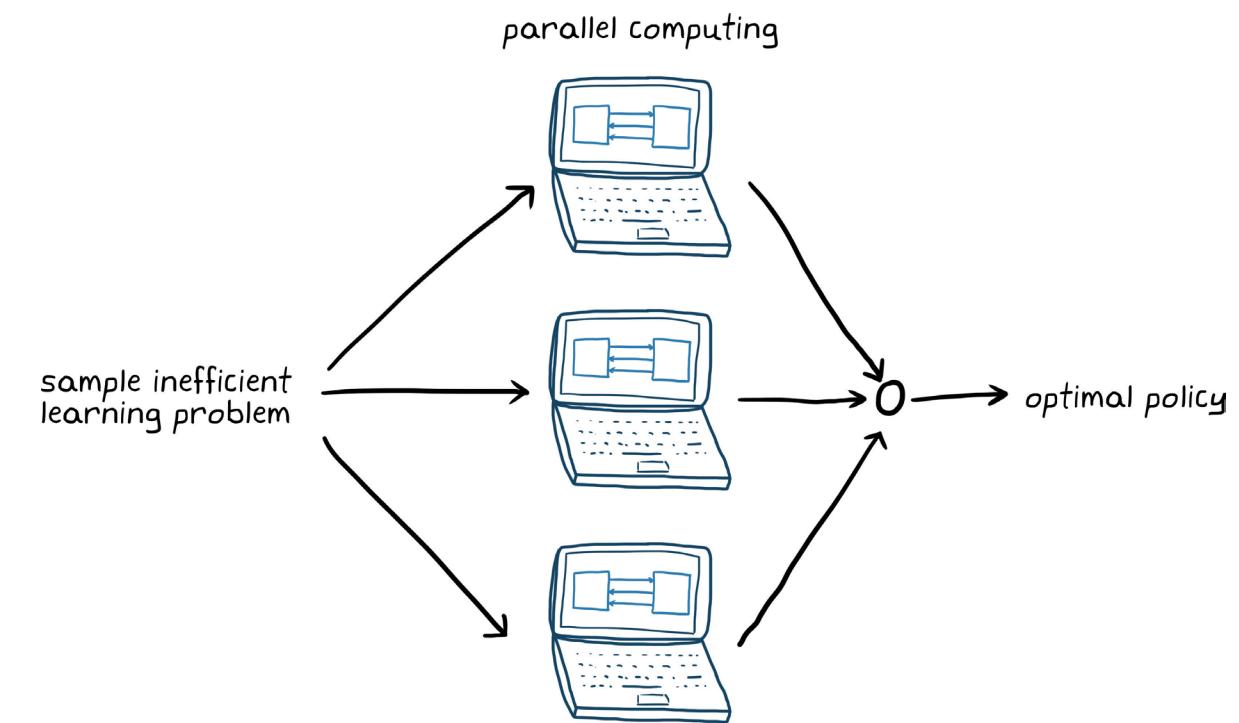
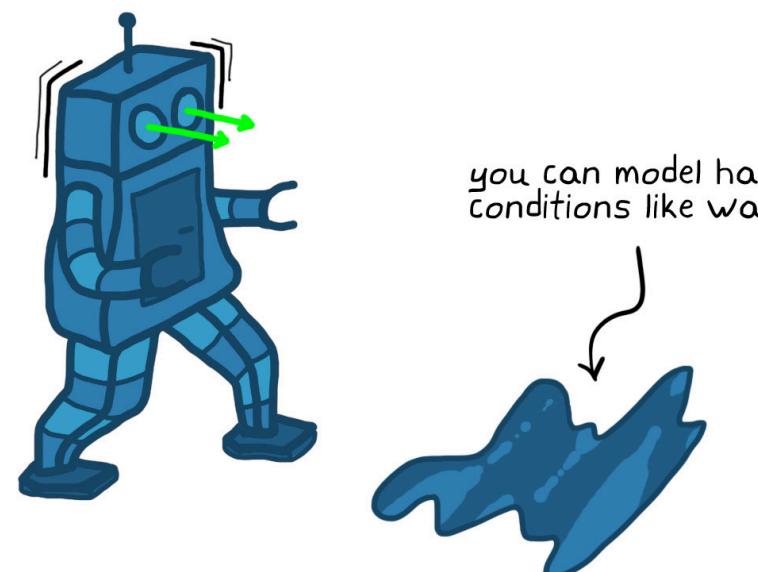
# 仿真环境的优势



仿真环境是最常用的代理训练方法。这对于控制问题的一大优势在于，通常已经具备良好的系统和环境模型，因为您往往需要系统和环境模型来进行传统控制设计。如果已在 MATLAB® 或 Simulink® 中搭建好模型，您可以将现有控制器替换为强化学习代理，向环境添加奖励函数，然后启动学习过程。

学习过程需要大量样本：试验、误差和修正。从这个意义上而言，学习过程效率极低，因为这期间可能需要经历数千乃至数百万个片段，才能收敛到最优解决方案。

环境模型的运行速度可能比实时环境快，您可以启动大量仿真让其并行运行。这两种方法都可以加快学习过程。



相较于在真实世界中让代理暴露在环境中，您对仿真状况的控制力要大得多。

例如，您的机器人或许必须能够在任意数量的不同表面上行走。通过仿真技术模拟在低摩擦表面（如冰面）上行走比实际冰面测试容易得多。此外，在低摩擦环境中训练代理其实还有助于机器人在各种平面上保持直立姿势。通过仿真可以营造更适宜的训练环境。

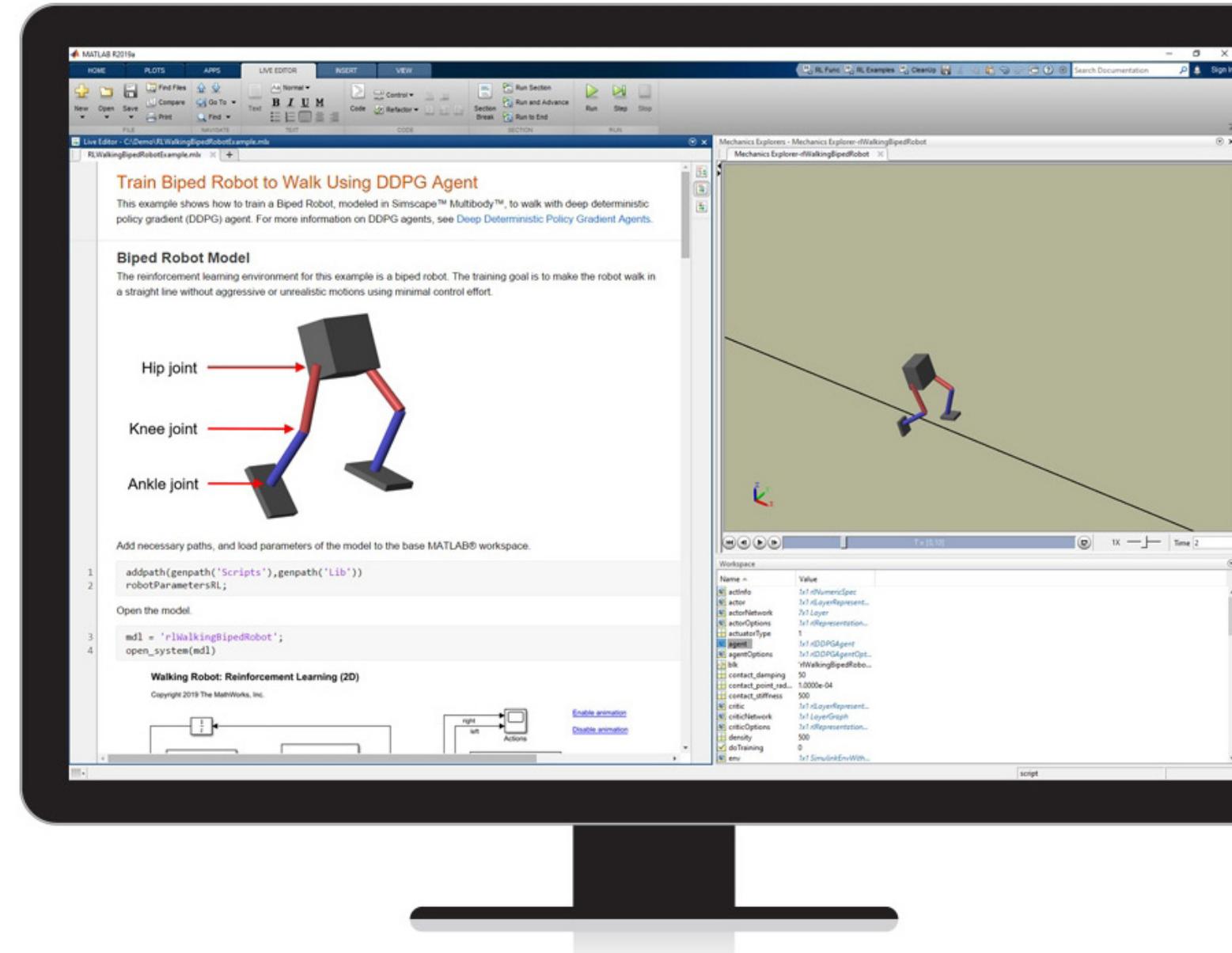
# 使用 MATLAB 和 Simulink 进行强化学习

Reinforcement Learning Toolbox 为使用强化学习算法进行策略训练提供了一些函数和模块。您可以使用这些策略为复杂系统(如机器人和自主系统)实现控制器和决策算法。使用该工具箱,您可以通过让代理与 MATLAB 或 Simulink 模型表征的环境进行交互,来训练策略。

例如,如需在 MATLAB 中定义强化学习环境,您可以使用现成的模板脚本和类,根据应用场合适当修改环境动态特性、奖励、观测量和动作。

在 Simulink 中您可以模拟大量不同的环境,以用于解决控制或强化学习问题。例如,您可以进行车辆动力学和飞行动力学建模;使用 Simscape™ 进行多种物理系统建模;使用 System Identification Toolbox™ 进行基于测量数据的近似动力学建模;对雷达、激光雷达和惯导模块等传感器建模等等。

[mathworks.com/products/reinforcement-learning](http://mathworks.com/products/reinforcement-learning)



# 了解更多 agent

## 观看

[什么是强化学习? \(14:05\)](#)

[了解环境和奖励 \(13:27\)](#)

[对步行机器人进行建模与仿真 \(21:19\)](#)

## 深入了解

[Reinforcement Learning Toolbox 入门](#)

[在 MATLAB 中创建环境](#)

[在 Simulink 中创建环境](#)

[在 Simulink 中进行飞行动力学建模](#)

[在 Simulink 中进行整车动力学仿真](#)

# environment

