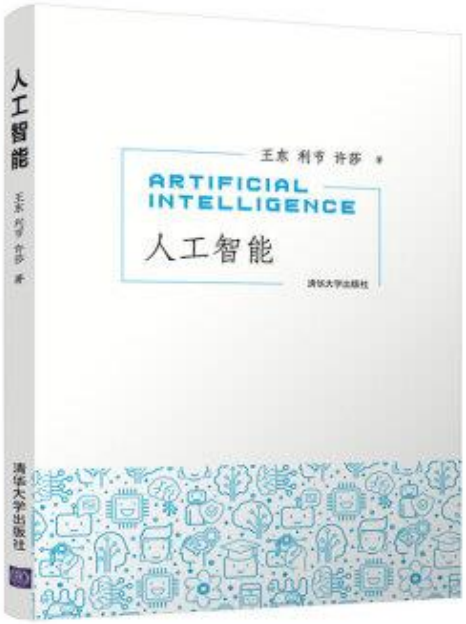
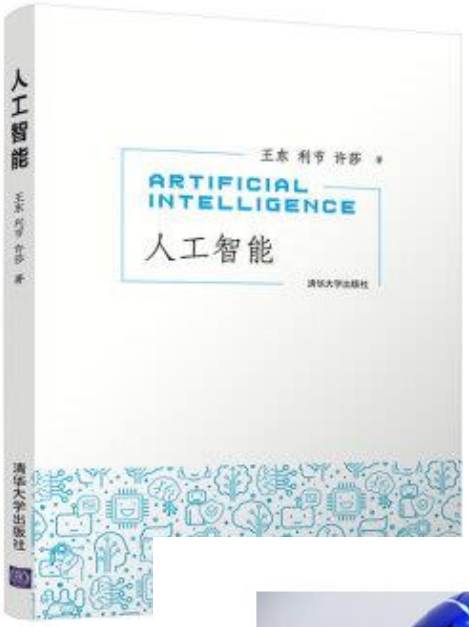
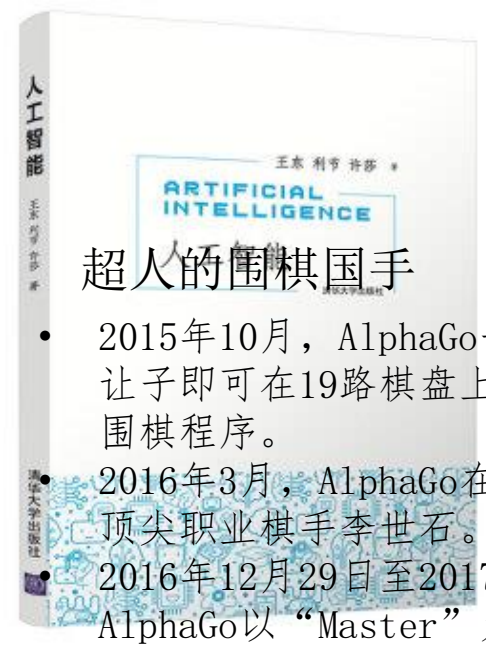


模仿你的行为

利节

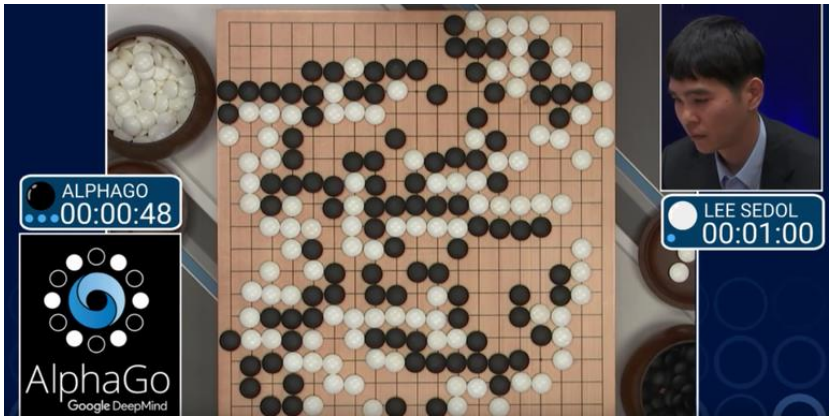


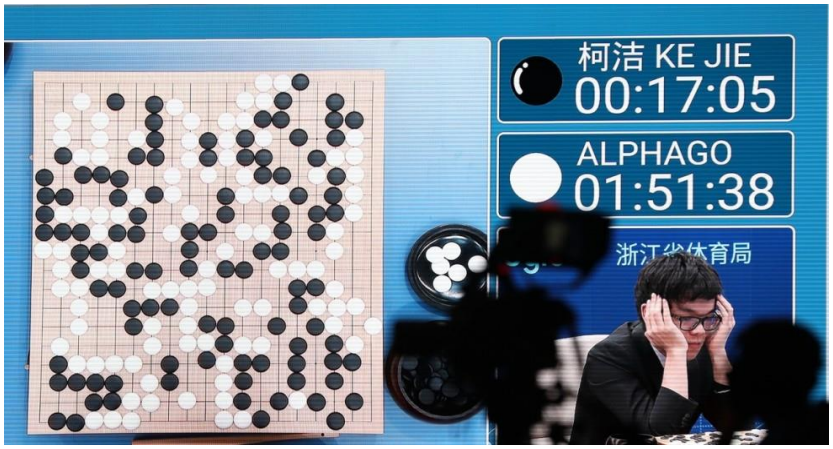
经过近一百年的发展，机器人经历了模仿、感知和智能三个阶段，现在已经成为人类的重要合作伙伴，在工业生产、自然探索、抢险救援、教育娱乐等各个方面扮演着重要角色。

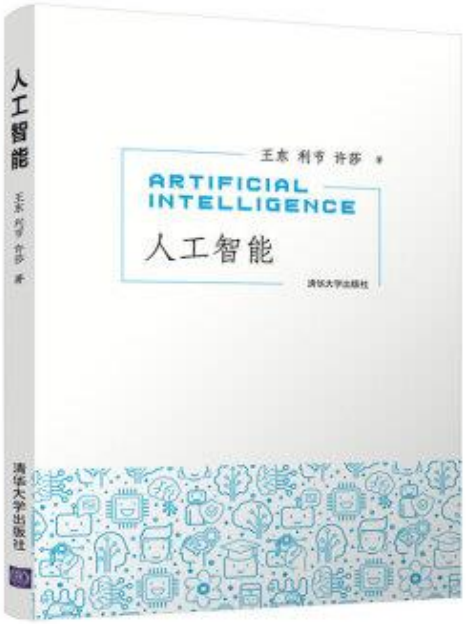


超人的围棋国手

- 2015年10月，AlphaGo击败樊麾，成为第一个无需让子即可在19路棋盘上击败围棋职业棋手的电脑围棋程序。
- 2016年3月，AlphaGo在一场五番棋比赛中4:1击败顶尖职业棋手李世石。
- 2016年12月29日至2017年1月4日，再度强化的AlphaGo以“Master”为账号名称，借非正式的网络快棋对战进行测试，挑战中韩日台的一流高手，测试结束时60战全胜。
- 2017年5月23至27日在乌镇围棋峰会上，最新的强化版AlphaGo和世界第一棋手柯洁比试、并配合八段棋手协同作战与对决五位顶尖九段棋手等五场比赛，获取3比零全胜的战绩，团队战与组队战也全胜.
- 2017年10月19日《自然》介绍AlphaGo Zero，这是一个没有用到人类数据的版本。AlphaGo Zero经过3天的学习，以100:0的成绩超越了AlphaGo Lee的实力，21天后达到了AlphaGo Master的水平，并在40天内超过了所有之前的版本。

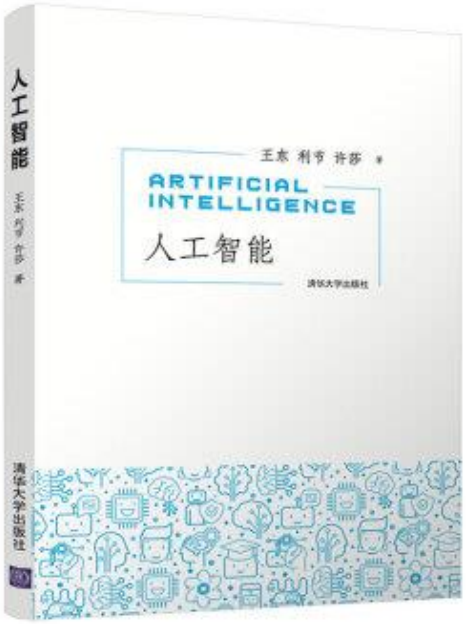






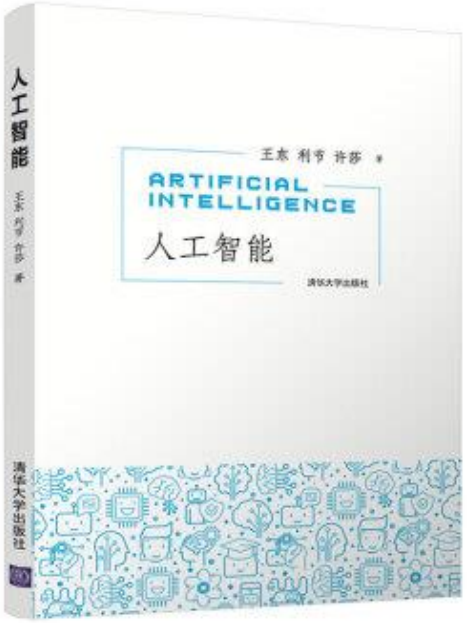
目 录

- 现代机器人发展史
- 基于设计的机器人
- 基于学习的机器人
- 深度强化学习方法



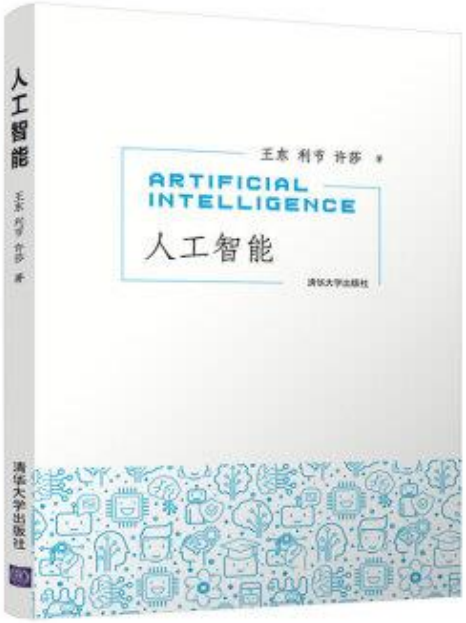
目 录

- 现代机器人发展史
- 基于设计的机器人
- 基于学习的机器人
- 深度强化学习方法



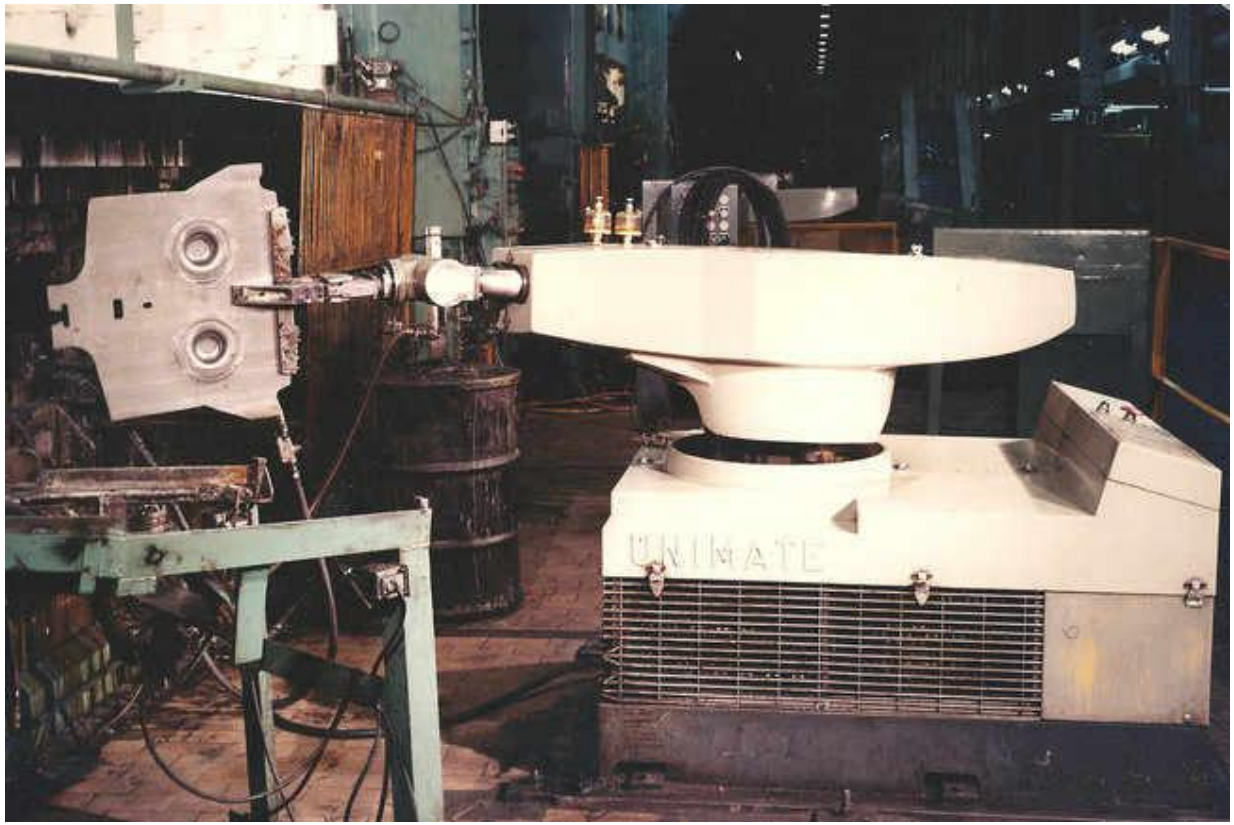
现代机器人发展史

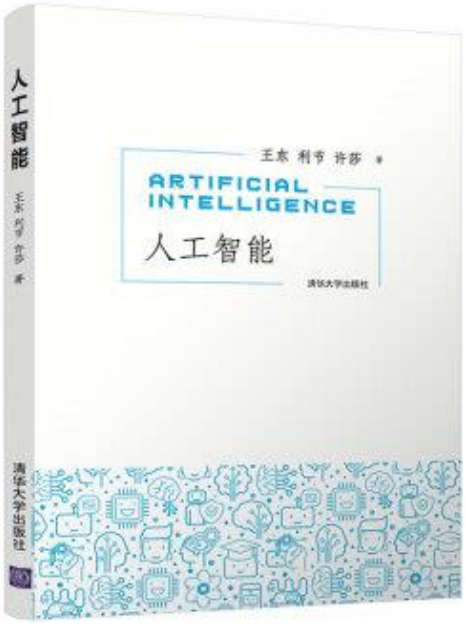
1927 年美国西屋公司（Westinghouse Electric）的工程师温兹利（Roy J. Wensley）制造了第一个机器人“电报箱”。电报箱具有无线电报功能，并可回答一些简单问题。



现代机器人发展史

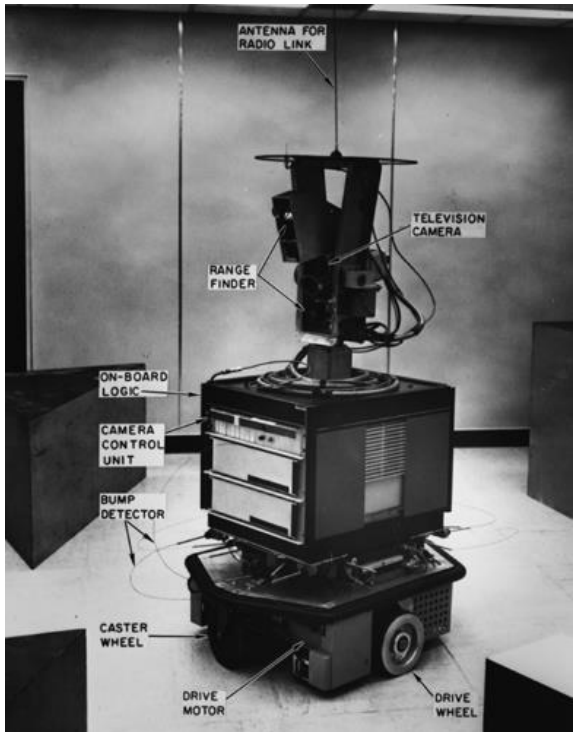
1958 年，英格伯格（Joseph Engelberger）和德沃尔成立了世界上第一个机器人公司，称为UNIMATION，1959 年，这家公司制造出了世界上第一台工业机器人。

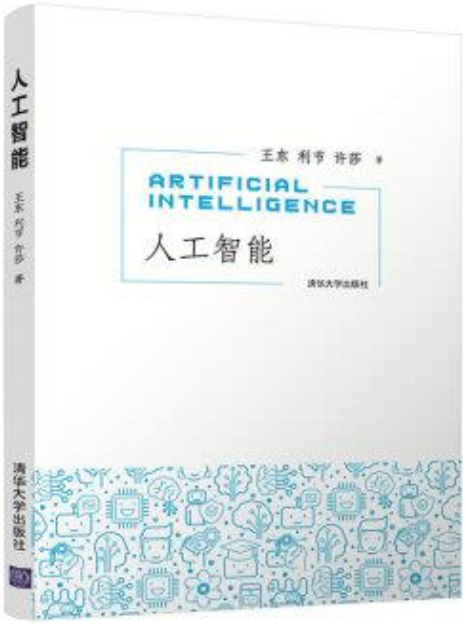




现代机器人发展史

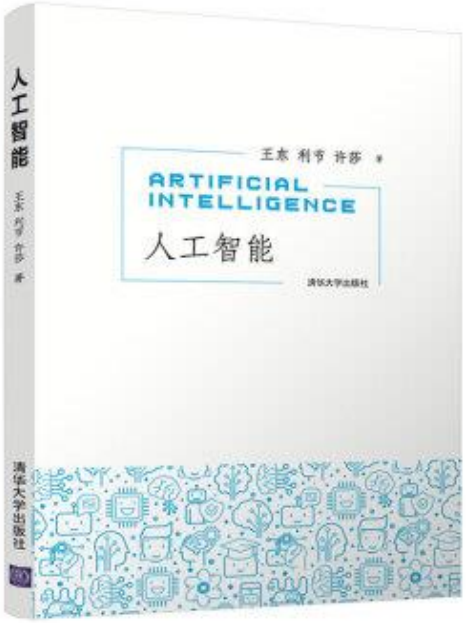
近年来，人工智能技术为机器人的发展带来了又一次飞跃。机器人集成了更多感知和学习的能力，可以在复杂场景中做到平衡行走，摔倒后站立起来；可以通过视觉与听觉对环境进行有效感知，并做出行为决策；可以识别人的身份、情绪；可以快速适应新的工作环境。





目 录

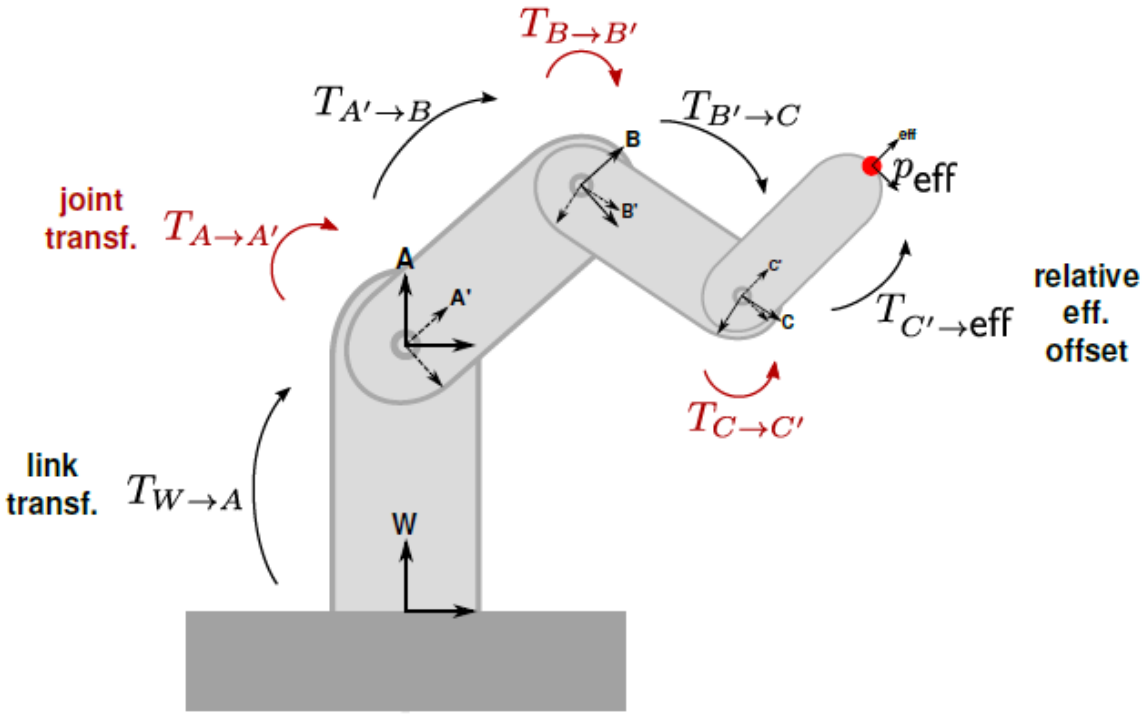
- 现代机器人发展史
- 基于设计的机器人
- 基于学习的机器人
- 深度强化学习方法

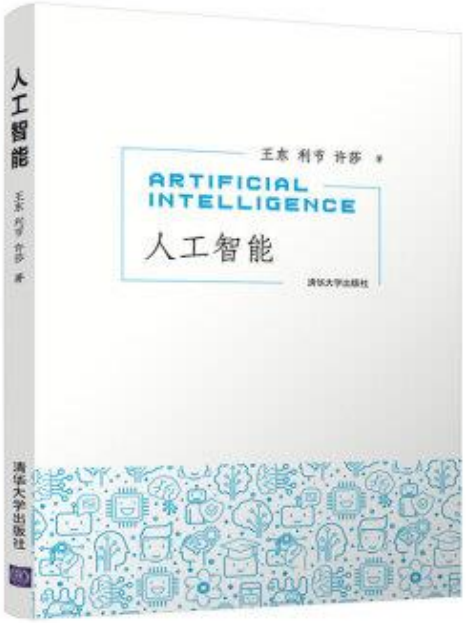


基于设计的机器人

1)操作机器人

操作机器人需要完成某种指定的动作，如抓取、焊接等。对机器人而言，实现动作的关键在于如何将机械臂的末端放置到预设位置。所谓末端，是指机械臂的工作点，一般是机械臂的末端。

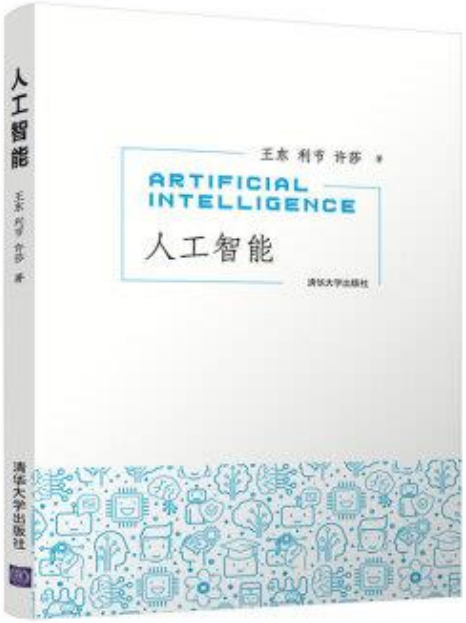




基于设计的机器人

1)操作机器人

为完成这一任务，我们需要解决如下三个问题：（1）如何设置各个连接杆的夹角；（2）如何对每个连接杆施力以实现上述目标夹角；（3）如何通过监控夹角的改变过程平稳地到达目标夹角。上述三个问题需要基于运动学（Kinematics）、动力学（Dynamics）和控制理论（Control theory）来解决。



基于设计的机器人

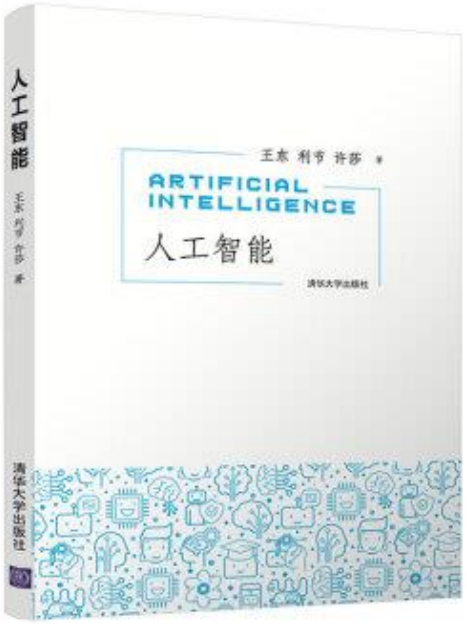
2)移动机器人

移动机器人是在地面上移动的机器人。移动机器人的任务有两个：一是实现有效的移动，二是对环境建立地图，并估计自身在地图中的位置。

基于设计的机器人

2)移动机器人

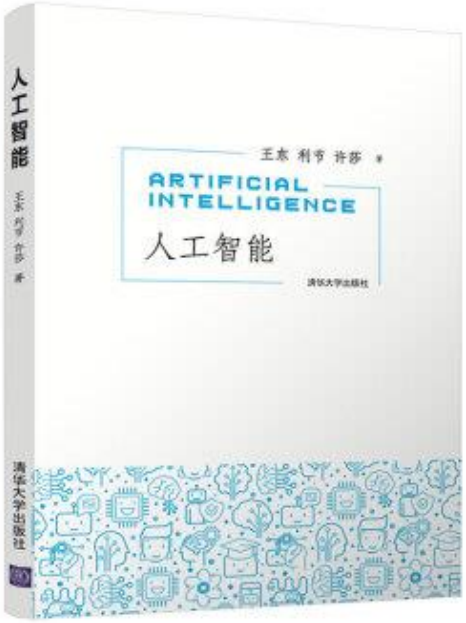
不同种类的机器人实现有效移动的方式不同。例如，自动驾驶汽车通过加油可以实现有效移动，但要考虑移动时的限制条件（如不能平行移动，移动时需考虑尾部占用的空间等）。而人形机器人要设计合理的四肢动作（如抬腿轨迹），并通过动力系统实现该动作。不论哪种移动方式，我们都假设机器人对自己的行为特性是了解的，只要给予合理的动力就可以实现有效移动。



基于设计的机器人

2)移动机器人

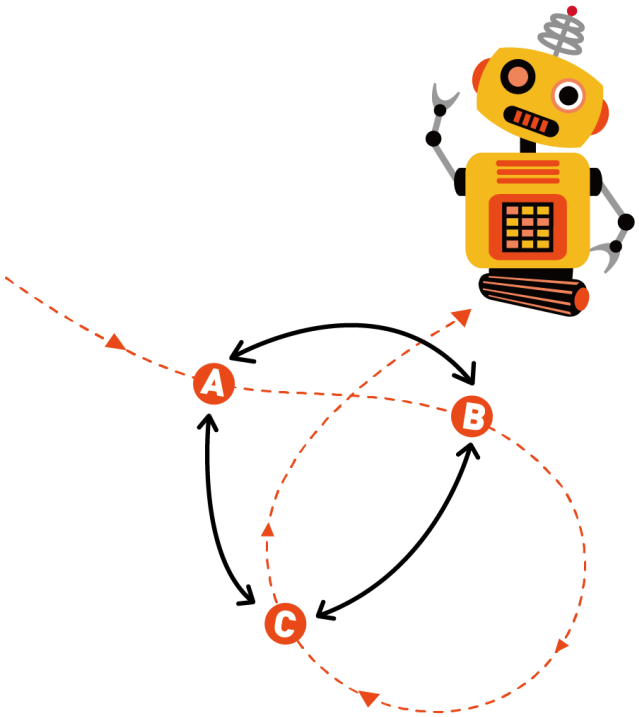
需要对环境建立地图，并估计机器人自身在地图中的位置。一种常见的方式是推着机器人先走一遍既定路线，让它记录下所见到的场景。这些场景的记录称为观测变量，可以是摄像头拍摄的照片，也可以是雷达的反馈信号等。经过若干次观测后，机器人即可建立一幅内部地图。

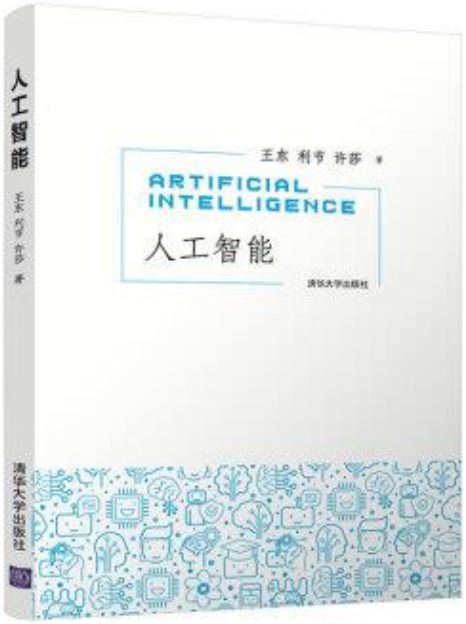


基于设计的机器人

2)移动机器人

当前很多机器人具有实时学习地图的功能，即同时进行位置估计和地图构建，这一方法称为同步定位与地图构建（Simultaneous localization and mapping，SLAM）算法。

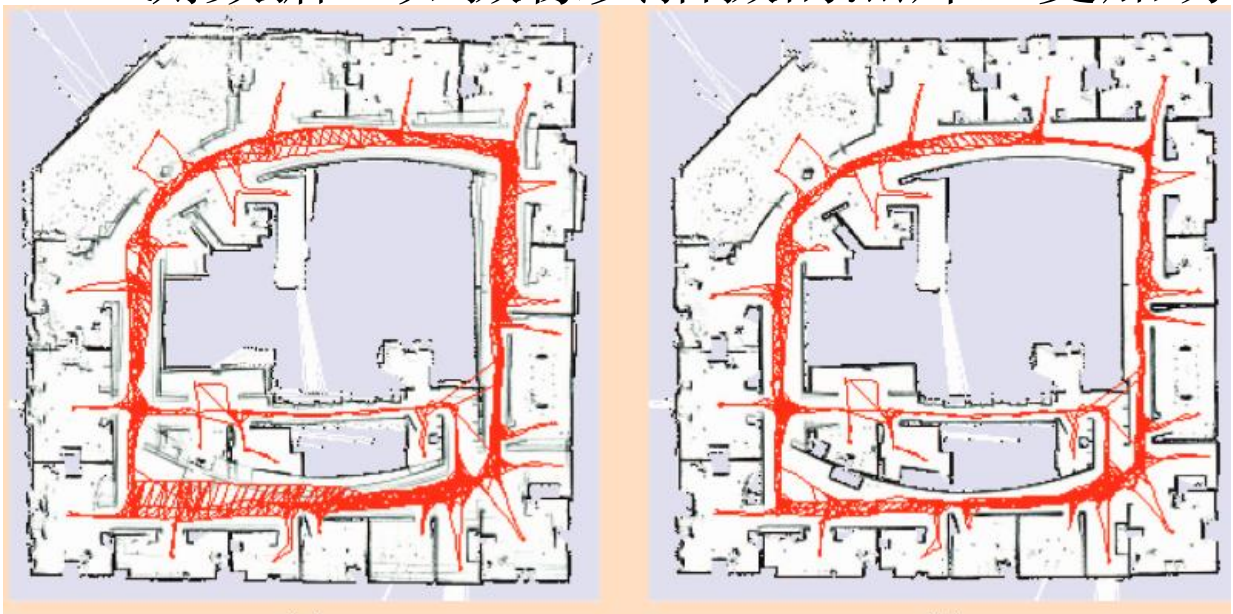


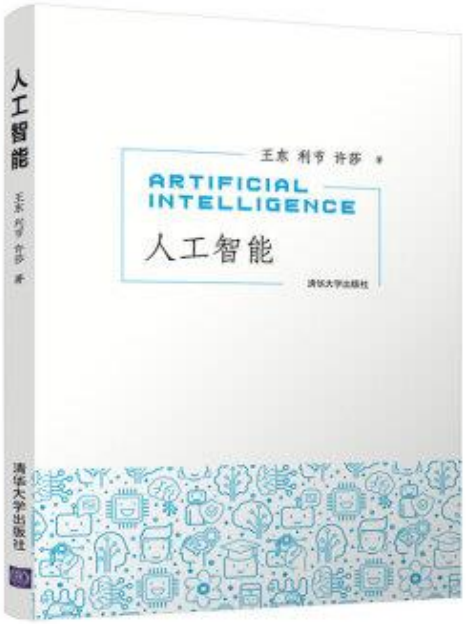


基于设计的机器人

2)移动机器人

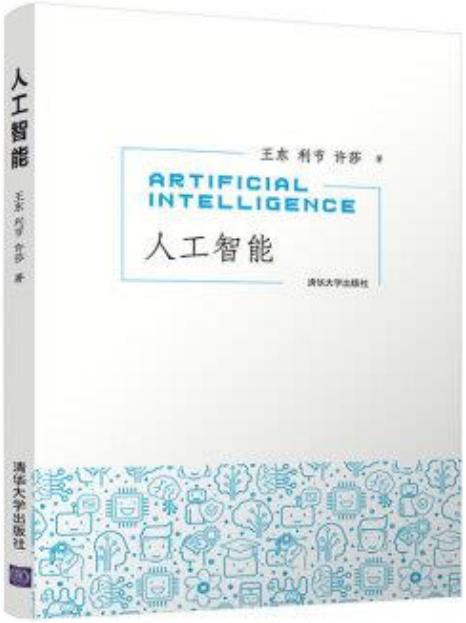
图中每个红点是机器人所在的位置（注意，这些位置是估计值），点之间的连线是不同位置之间的关联性。左侧是粗略估计的地图，右侧是对该地图修正后的结果。在修正过程中，通过算法对每个点的位置进行调整，使之与观测数据（如摄像头拍摄的照片）更加吻合。





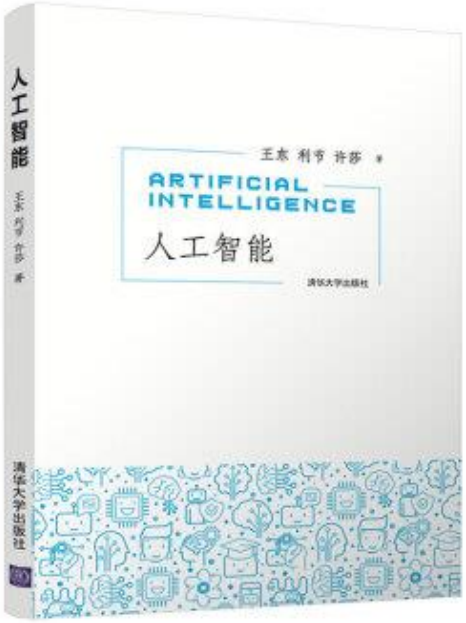
目 录

- 现代机器人发展史
- 基于设计的机器人
- 基于学习的机器人
- 深度强化学习方法



基于学习的机器人

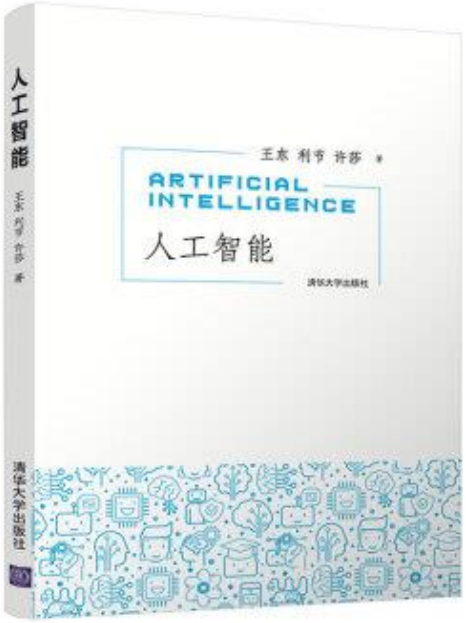
基于学习的机器人。与人为设计的机器人不同，这种机器人的行为方式很大程度上来源于外界经验。这些经验有可能是人传授的，也有可能是机器人自己尝试出来的。不论哪种方式，都不需要对机器人的每一个动作进行设计，只需要告诉它要完成什么目标，剩下的由机器人去自主学习。这种学习方法称为**强化学习**。



基于学习的机器人

1)简单的例子：模仿学习

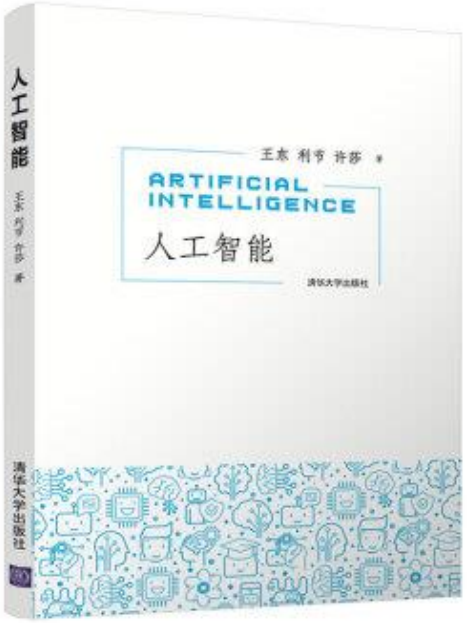
示教机器人可以认为是最简单的学习。所谓示教，意思是由人拉着机器人的末端把任务执行一遍，机器人记住这一过程中每个连接点的角度、角速度等，即可重现这一过程。然而，这种示教并非自主学习，机器人还是需要计算完成示范过程所需要的力矩。



基于学习的机器人

1)简单的例子：模仿学习

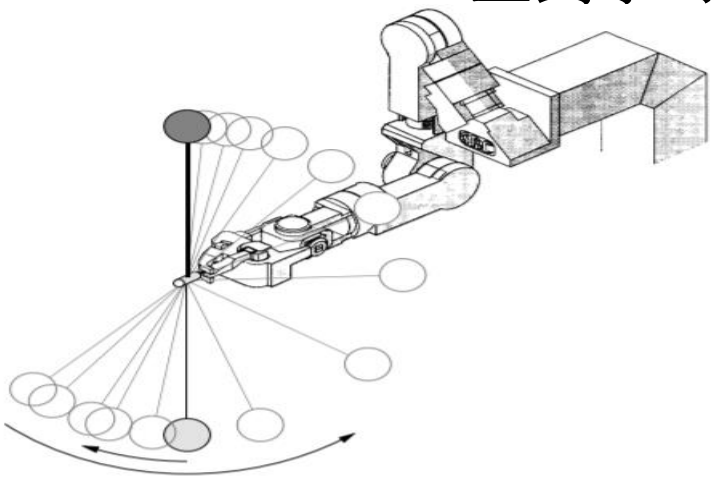
如果机器人可以通过主动学习示教过程来完成目标任务，这种机器人即是模仿学习机器人。在这一学习过程中，机器人会通过不断尝试施于每个连接上的力矩以获得和示教过程一致的运动轨迹，最终获得完成示教过程的操作技巧。注意，模仿学习中的力矩大小是通过学习得到的，而示教机器人的力矩大小是通过计算得到的。

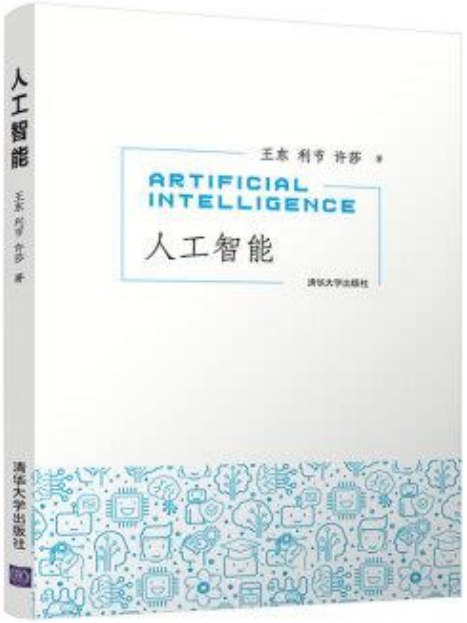


基于学习的机器人

1)简单的例子：模仿学习

模仿学习不仅可以学习运动轨迹，还可以学习行为策略。例如，机器要学习如何将小球绕起来。人完成这一过程的动作是很复杂的，而且人的反馈机制和机器的反馈机制有所不同，模仿人手指的动作比较困难。但是，机器可以模仿小球成功绕起来后的轨迹形状，并以此为目标进行尝试，直到学习到将小球绕起来的技巧。

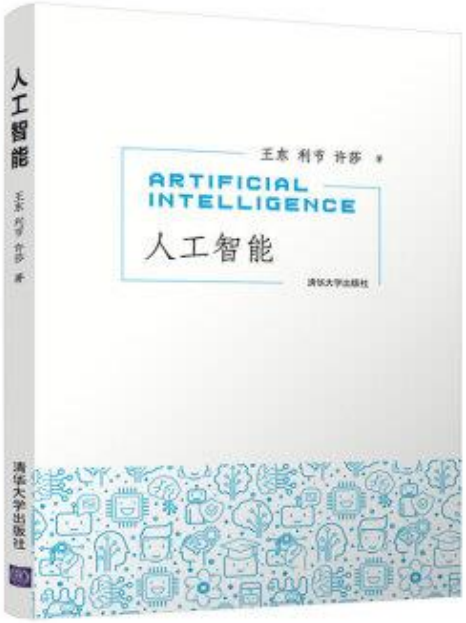




基于学习的机器人

2)强化学习

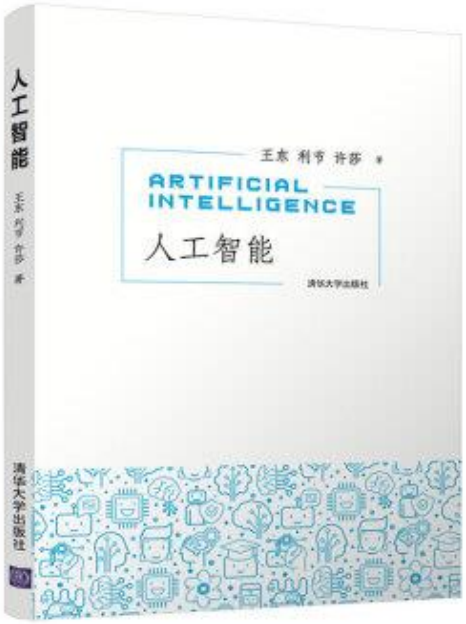
在模仿学习中，需要给机器人一个或几个例子，让它仿照操作。在一般的强化学习中，没有这些例子可以模仿，机器人需要主动和环境打交道，从中得到反馈，基于此不断修正自己的行为方式，直至得到最优回报。



基于学习的机器人

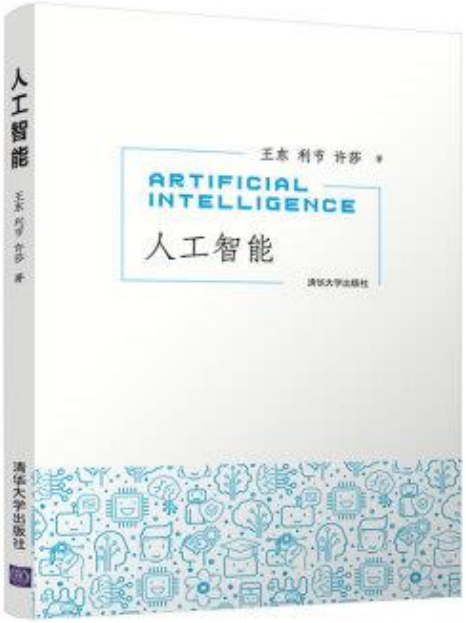
2)强化学习

以我们小时候学习走路的经验为例：刚开始的时候孩子并不会有任何目的性的动作，只会随机活动四肢，家长也不会刻意帮他抬腿、迈步，也不会解释行走的好处，但会在他偶尔站立、扶着墙挪动的时候给予鼓掌、拥抱等鼓励，让孩子倾向于继续尝试这些动作。同时，当孩子站错了姿势摔倒时家长并不会马上批评他，但他会感到疼痛，下次就会避免做出类似的错误动作。经过多次尝试后，孩子就会渐渐学习到站立、迈步的技巧，最终一点点学会走路。



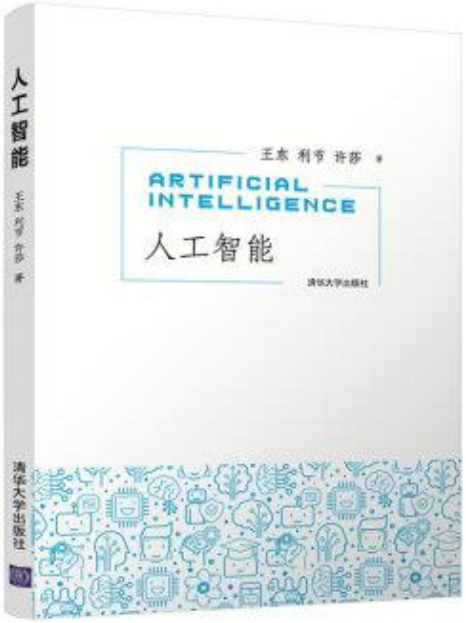
目 录

- 现代机器人发展史
- 基于设计的机器人
- 基于学习的机器人
- 深度强化学习方法



深度强化学习方法

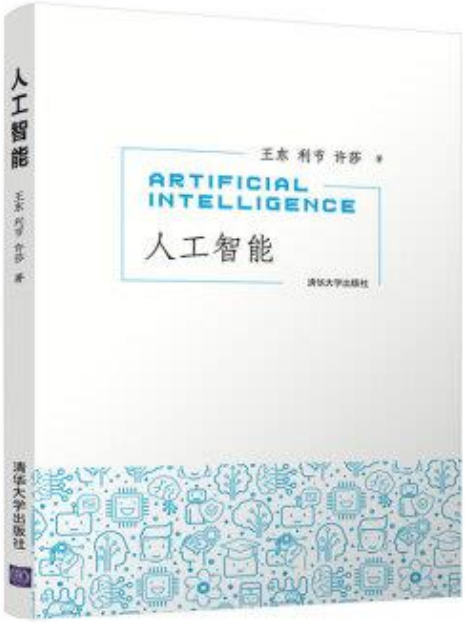
将深度学习和强化学习结合起来的方法称为深度强化学习。深度强化学习是当前机器学习乃至整个人工智能领域研究的热点之一，取得了一系列让人振奋的成果。下面我们将讨论几个深度强化学习的例子。



深度强化学习方法

1)Atari 游戏

游戏一直是强化学习擅长的领域，从最初Samuel 的西洋棋到Tesauro 的TD-Gammon 。然而，在2016 年以前，可能没有人会想到机器玩起游戏来竟如此强大，不仅可以在简单游戏中战胜人类业余选手，还可以在极为复杂的任务中战胜人类顶尖高手。



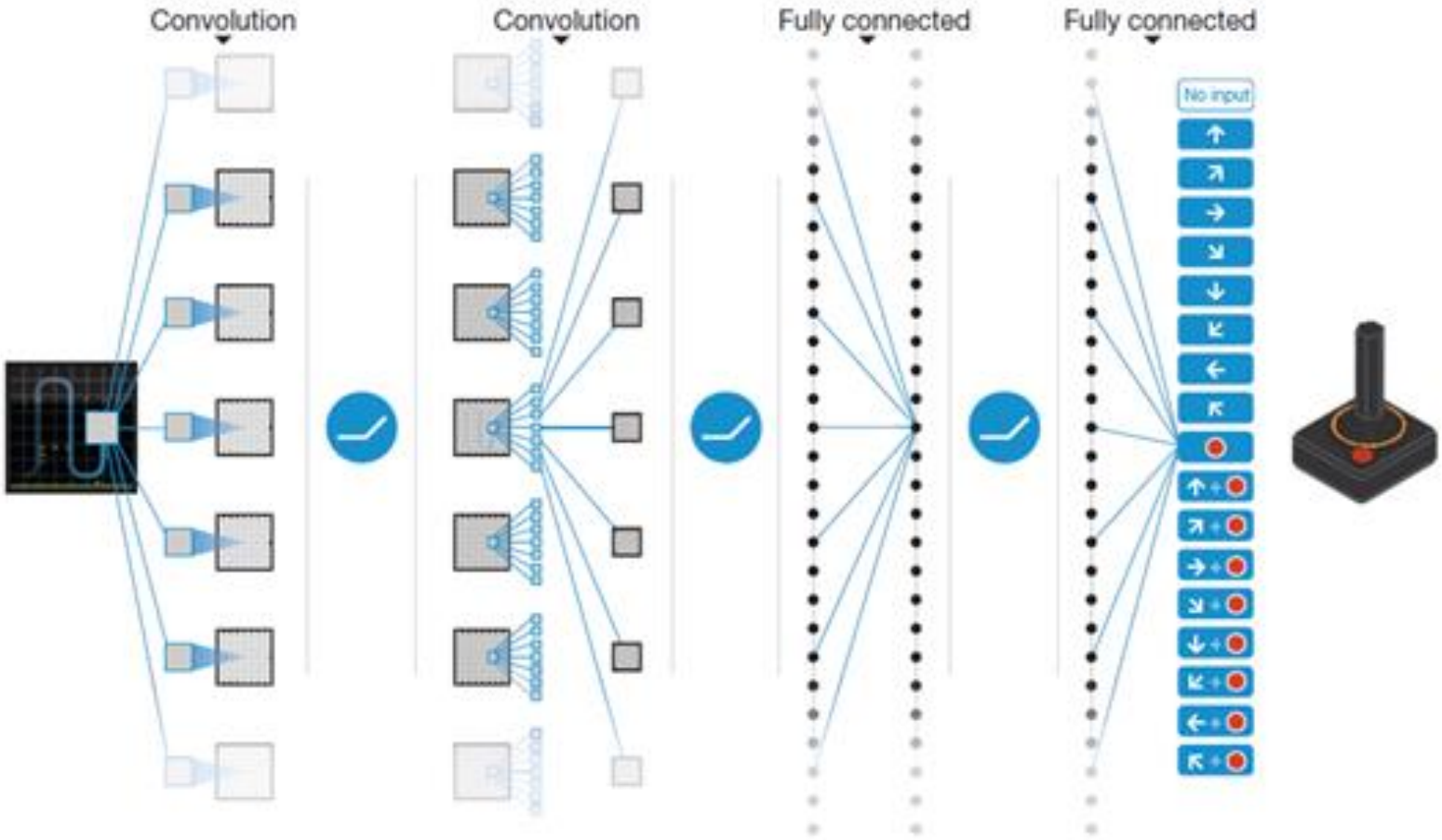
深度强化学习方法

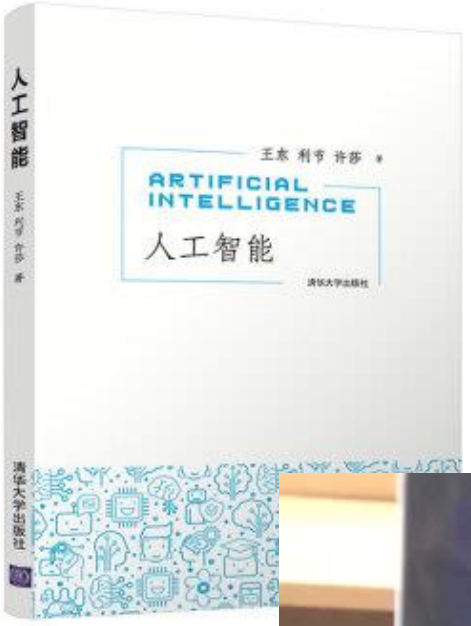
1)Atari 游戏

突破从Deep Mind 公司利用深度Q-learning 网络（DQN）教会机器玩Atari 游戏开始。Atari 平台包括49 个游戏，学习方法很简单：把游戏画面传给计算机，让它通过观察这些画面来控制游戏杆，像人一样操作游戏。基于学习信号的复杂性和交互性，这是一个典型的强化学习任务，其中，观察值为所看到的游戏画面，动作为对游戏杆的操纵，反馈为屏幕上给出的奖励分数，总收益为游戏结束后得到的总分值。在这一任务中，唯一的困难是输入的观察值太过原始（游戏画面），将这一观察值直接作为状态输入很难被机器理解。

深度强化学习方法

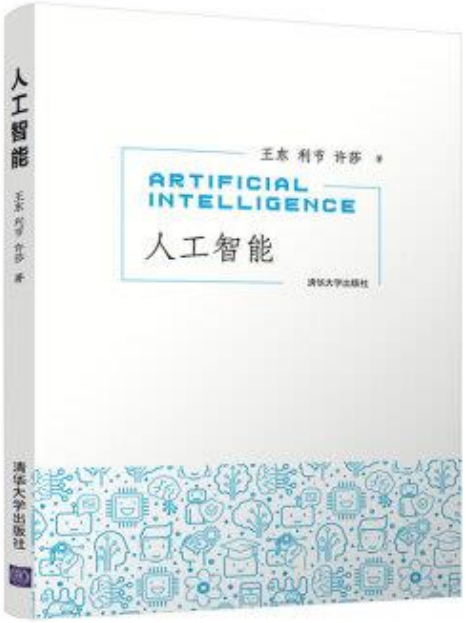
1)Atari 游戏





深度强化学习方法

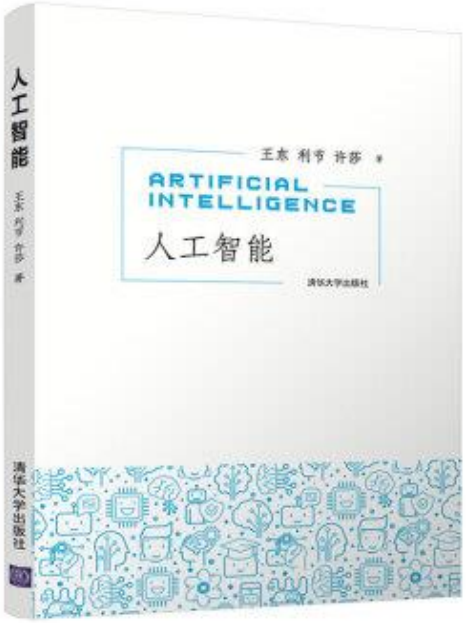
1)Atari 游戏



深度强化学习方法

2)AlphaGo Zero

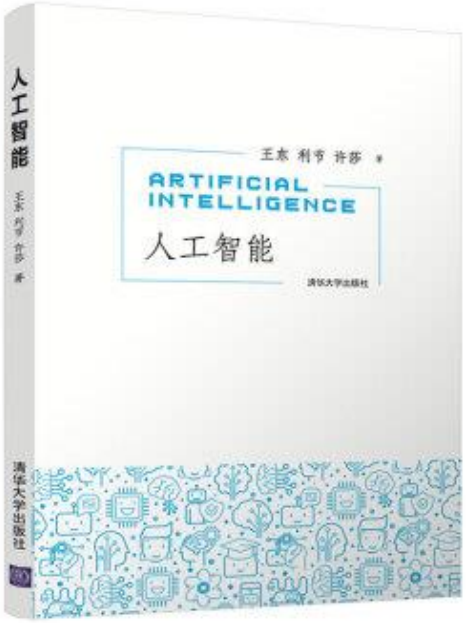
AlphaGo 是深度强化学习的另一份杰作。在AlphaGo 之前，已经有数个团队在开发围棋程序，如Zen 和Crazy Stone。这些程序的基本思路和击败卡斯巴罗夫的国际象棋程序深蓝类似，大量采用启发式搜索算法。由于围棋的复杂性，这些先期程序的棋力仅与业余高手相当，无法击败高段职业棋手。



深度强化学习方法

2)AlphaGo Zero

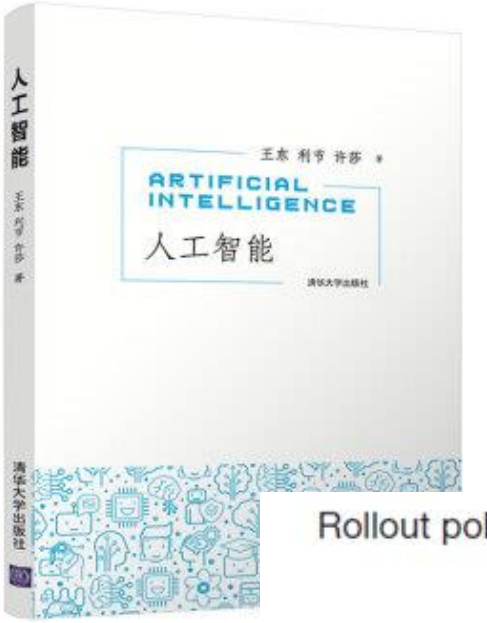
AlphaGo 是DeepMind 开发的围棋程序，其基本思路是采用深度强化学习方法，将整个棋盘作为输入，通过若干层CNN 网络提取棋局状态，并基于强化学习方法进行训练。为了取得更好的效果，AlphaGo 还加入了大量监督学习以模仿人类的走棋方法。



深度强化学习方法

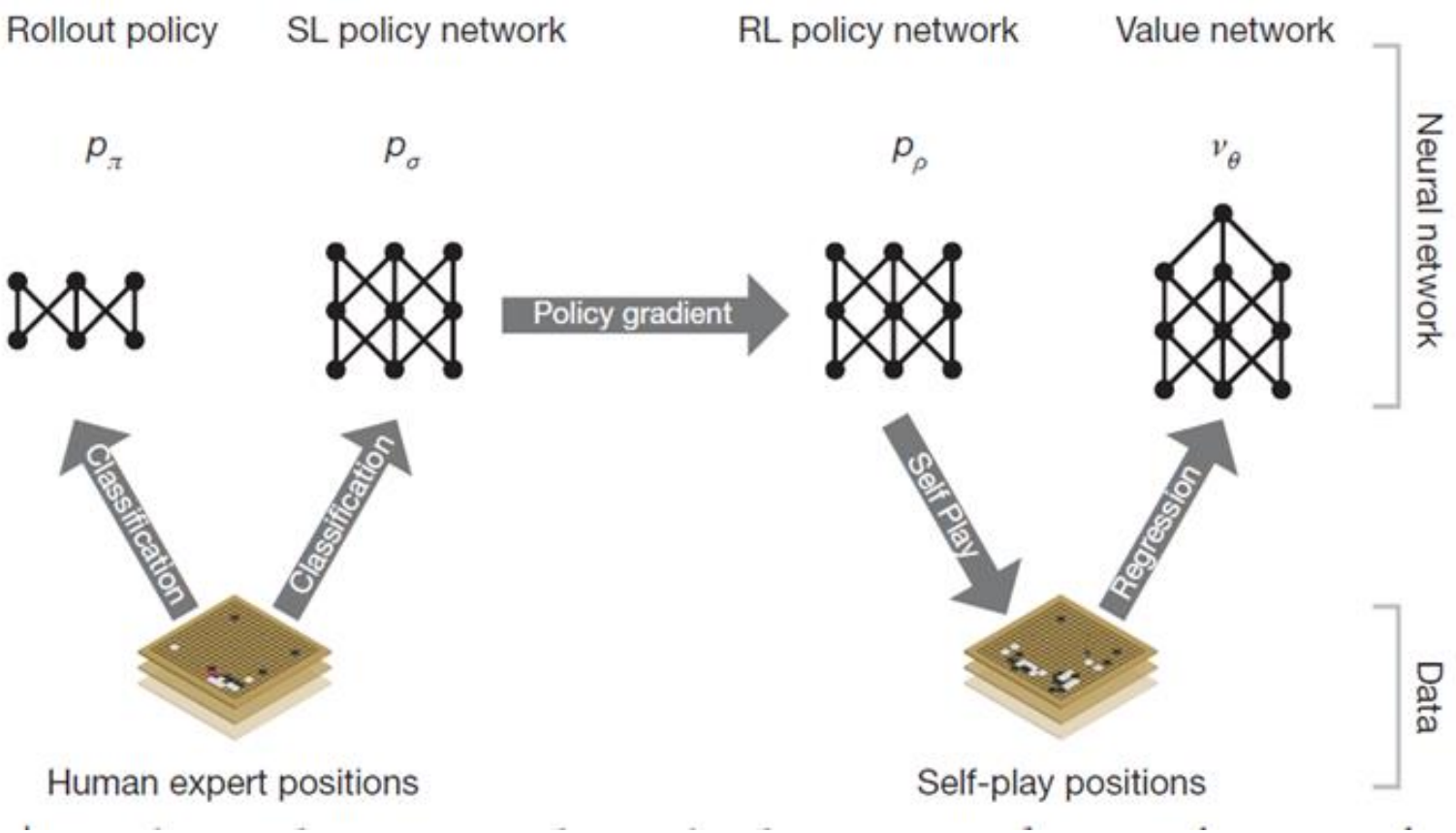
2)AlphaGo Zero

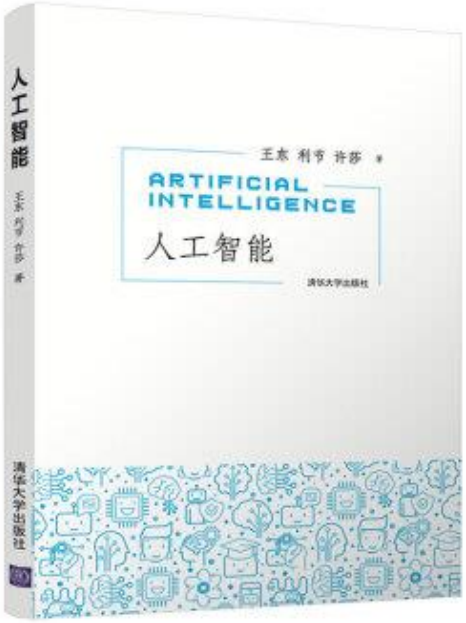
2017 年10 月19 日，DeepMind 团队在自然杂志发表论文，不再学习人类的棋谱，而是完全依赖深度强化学习，通过自我对奕学习机器自己的走棋方式。该系统称为AlphaGo Zero。因为没有人类的棋局信息，AlphaGo Zero 学到的是完全属于机器的围棋，只管胜败，不计手段的围棋。DeepMind 的论文表明，使用64 个GPU 和19 个CPU，AlphaGo Zero 用三天时间完成了自我对弈490 万局。几天之内它就发展出击败人类顶尖棋手的技能，而早期的AlphaGo 要达到同等水平需要数月的训练。



深度强化学习方法

2)AlphaGo Zero

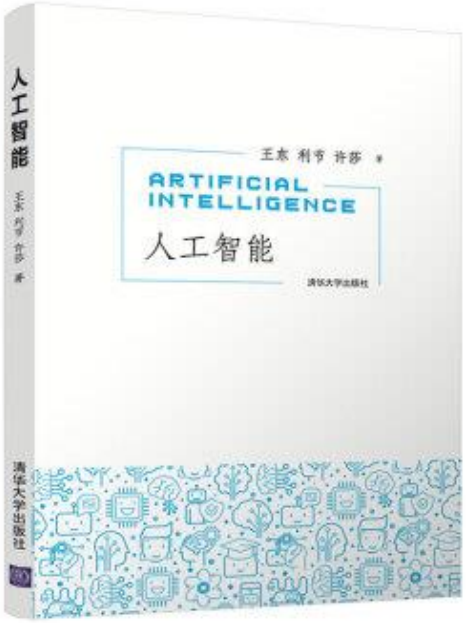




深度强化学习方法

3)实体机器人

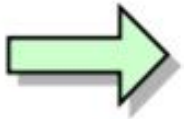
2008 年日本Oita 大学的一个工作，目的是让一只叫作AIBO 的机器狗去亲吻一只白色的机器狗。作者采用和Atari 游戏网络类似的结构。该网络输入为摄像头拍到的照片，输出为直走、左转、右转三个命令，网络结构是全连接网络。在训练时，如果接触到白狗，会给系统一个很大的奖励，如果白狗从视野消失，会给以相应的惩罚。通过多次训练，AIBO 即可学习到如何接近白狗的行为方式。



深度强化学习方法

3)实体机器人

Start



Goal!

深度强化学习方法

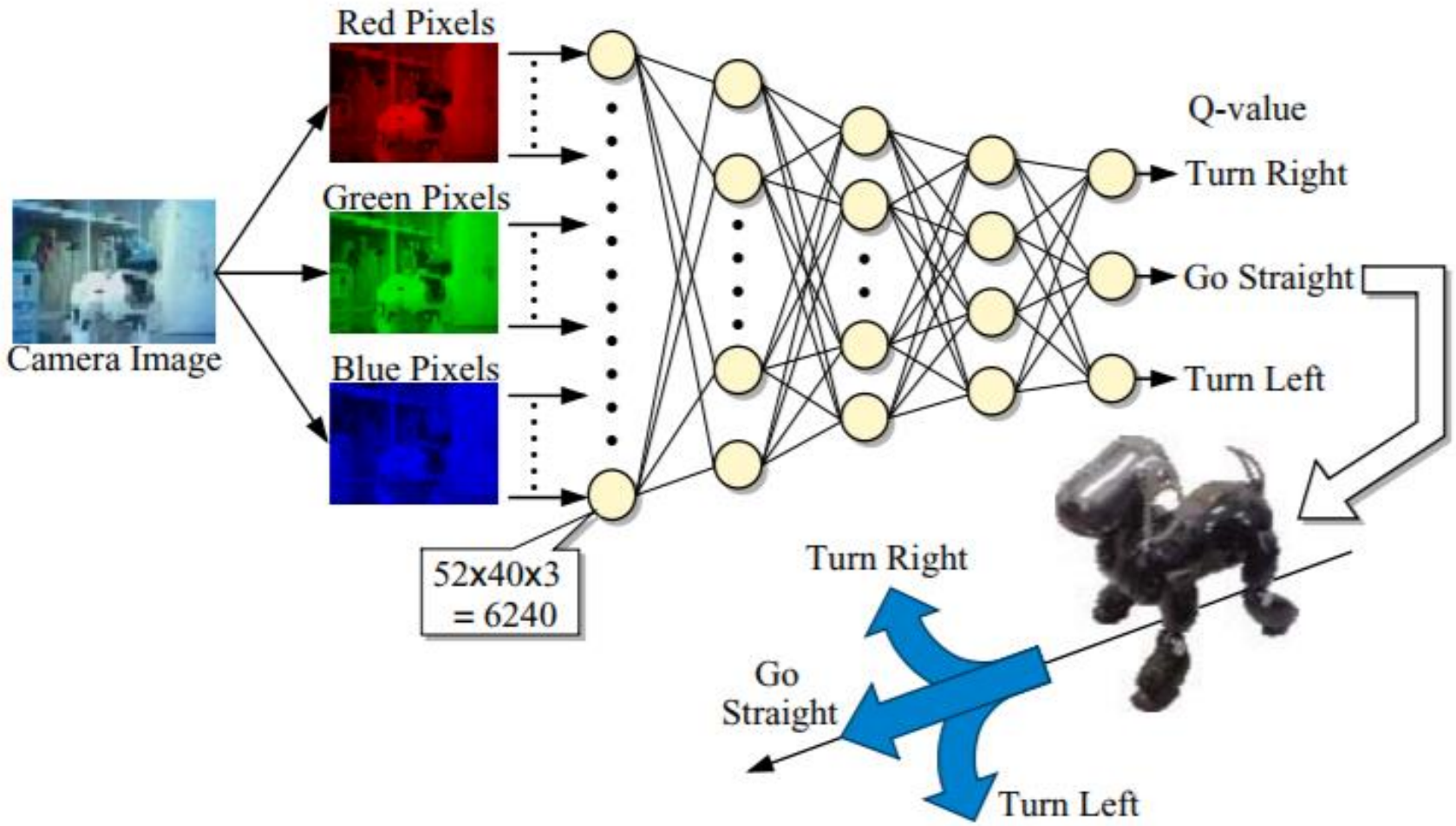
3)实体机器人

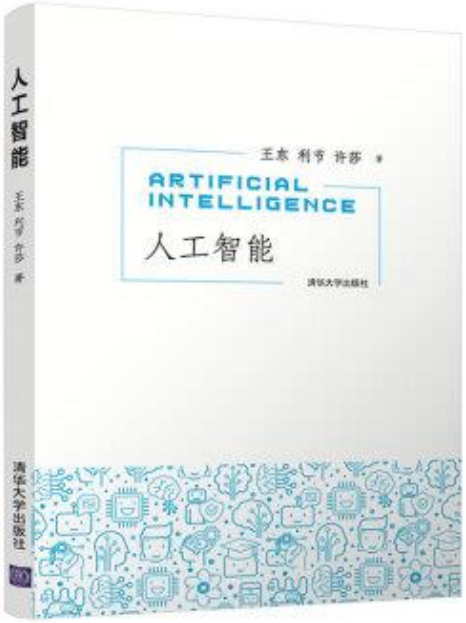
Kissing AIBO Task (2008)

The world's first end-to-end Deep Reinforcement Learning (DQN) applied to a Real Robot

深度强化学习方法

3)实体机器人



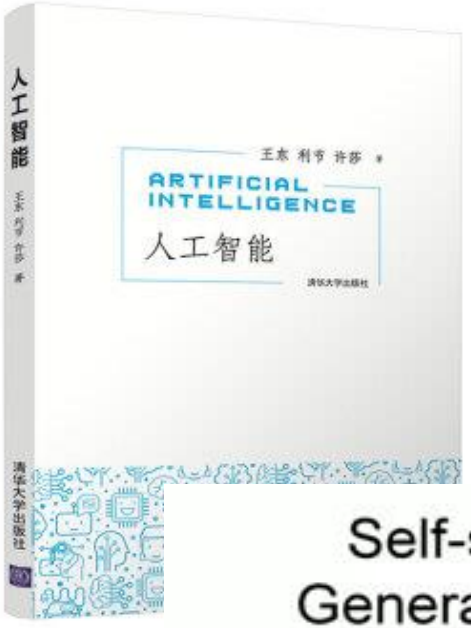


深度强化学习方法

3)实体机器人

美国加洲大学伯克利分校2017 年让一辆小车自动学习到在室内复杂环境下的驾驶技术。这个小车只有一个摄像头，训练的目标是在行驶过程中不要发生碰撞。

网络训练时，小车根据当前策略自主驾驶，如果没有发生碰撞，给予一个正向奖励，如果发生了碰撞，则给予一个负面惩罚。经过四个小时的自主学习后，该小车学会了安全驾驶的技巧。



深度强化学习方法

3)实体机器人

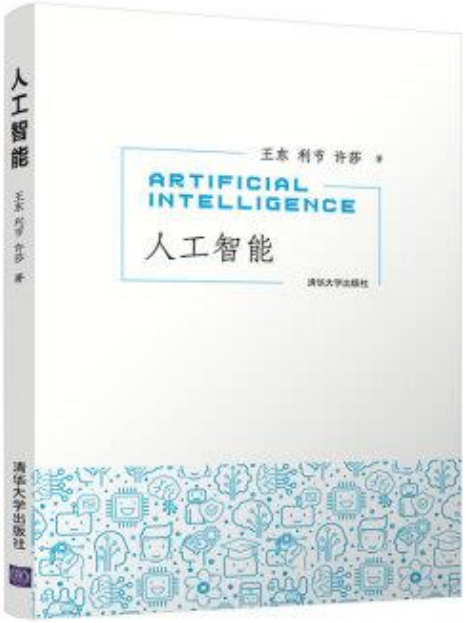
美国加洲大学伯克利分校2017 年让一辆小车

Self-supervised Deep Reinforcement Learning with Generalized Computation Graphs for Robot Navigation

Gregory Kahn, Adam Villaflor, Bosen Ding, Pieter Abbeel, Sergey Levine

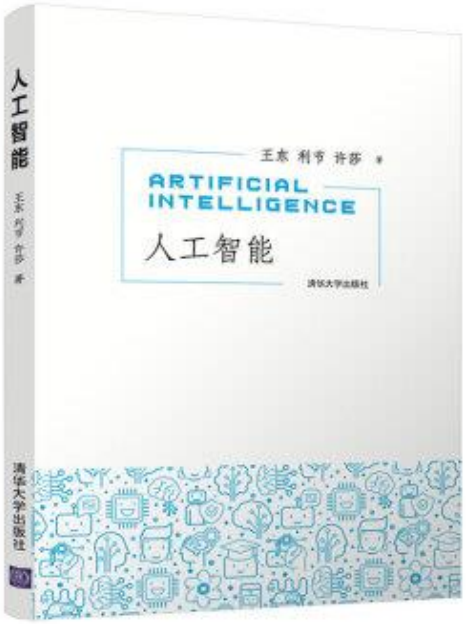
BAIR

BERKELEY ARTIFICIAL INTELLIGENCE RESEARCH



总结

讨论了基于人为设计的机器人和基于学习的机器人。这两种机器人各有优缺点：基于人为设计的机器人安全可靠，但要处理复杂环境中的复杂任务则比较困难；基于学习的机器人可以适应复杂场景，但需要做大量尝试。对于虚拟机器人，怎么尝试都无所谓，但对实体机器人，试错带来的风险往往不可容忍。因此，当前基于学习的机器人大多用于虚拟任务。如何利用模拟数据对实体机器人进行大规模训练，以减少实际尝试带来的风险是非常重要的研究方向。



The end !