7 深度强化学习的挑战

本章介绍了现有深度强化学习研究和应用中的挑战，包括：（1）样本效率问题；（2）训练稳定性；（3）灾难性遗忘问题；（4）探索相关问题；（5）元学习和表示学习对于强化学习方法的跨任务泛化性能；（6）有其他智能体作为环境一部分的多智能体强化学习；（7）通过模拟到现实迁移来弥补模拟环境和现实世界间的差异；（8）对大规模强化学习使用分布式训练来缩短执行时间，等等。本章提出了以上挑战，并介绍了一些可能的解决方案和研究方向，来引出本书第二个板块的前沿主题，从第 8 章到第 12 章，给读者提供关于深度强化学习现有方法的缺陷、近来发展和未来方向的相对全面的理解。

7.1 样本效率

强化学习中一个**样本高效**（Sample-Efficient，或称**数据高效**，Data-Efficient）的算法意味着这个算法可以更好地利用收集到的样本，从而实现更快速的策略学习。使用同样数量的训练样本（比如按强化学习中的时间步来统计），相比于其他样本低效的方法，一个样本效率高的方法可以在学习曲线或最终结果上表现得更好。以 Pong 游戏为例，一个普通人可能通过几十次尝试就基本掌握游戏规则并取得较好的分数。然而，对于现有的强化学习算法（尤其是无模型的方法）而言，它可能需要成百上千个样本来逐渐学到一些有用的策略。这构成了强化学习中的一个关键问题：我们如何为智能体设计更有效的强化学习算法，从而用更少的样本更快地学习？

这个问题的重要性主要是由于实时或现实世界中的智能体与环境交互往往有较大的代价，甚至目前即使在模拟环境中的交互也需要一定的时间和能源上的消耗。多数现有强化学习算法在解决大规模或连续空间问题时有较低的学习效率，以至于一个典型的训练过程即使有着较快的模拟速度在当前计算能力下也需要难以忍受的等待时间。对于现实世界的交互过程情况可能更糟，一

些潜在的问题，比如时间消耗、设备损耗、强化学习探索过程中的安全性和失败情况下的风险等，都对实践中强化学习算法的学习效率提出了更高的要求。

提高数据使用效率，一方面需要包含有用信息的先验知识，另一方面需要能够从可获得数据中更高效提取信息的方式。从这两方面出发，现有文献中有许多方式解决学习效率的问题：

- 从**专家示范**（Expert Demonstrations）中学习。这个想法需要一个专家来提供有高奖励值的训练样本，实际上属于**模仿学习**（Imitation Learning）的范畴。它尝试不仅模仿专家的动作选择，而且学习一个能解决未见过情况的泛化策略。模仿学习和强化学习的结合实际上是一个很有前景的研究领域，在近几年来被广泛研究，并应用于如围棋游戏、机器人学习等，来缓解强化学习低学习效率的问题。

  从专家示范中学习的关键是从可获得的示范数据集中提取能生成好的动作的潜在规则，并将其用于更广泛的情况。

- **基于模型**（Model-Based）的强化学习而不是**无模型**（Model-Free）强化学习。如前面章节所介绍的，一个基于模型的强化学习方法一般指智能体不仅学会一个预测其动作的策略，而且学习一个环境的模型来辅助其动作规划，因此可以加速策略学习的速度。环境的模型基本包括两个子模型：一个是**状态转移模型**（State Transition Model），它可以给出智能体做出动作后的状态变化；一个是**奖励模型**（Reward Model），它决定了智能体能从环境中得到多少奖励作为其动作的反馈。

  学习准确的环境模型可以为更好地评估智能体的当前策略提供额外信息，而这可以使整个学习过程更高效。然而，基于模型的方法有它自己的缺点，比如，实践中，基于模型的方法经常会有**模型偏差**（Model Bias）的问题，即基于模型的方法经常固有地假设学习到的环境模型能准确地刻画真实环境，但是对于模型只能从少量样本中学习的情况，这往往不成立，即实际模型基本不准确。在真实环境中，当策略基于不准确或者有偏差的模型进行学习时可能会产生问题。

  举例来说，一种基于模型的高效强化学习算法叫 PILCO (Deisenroth et al., 2011)，它应用非参数化的概率模型高斯过程来近似环境的动力学模型。它利用了高斯过程简单直接的求解过程来有效地学习模型，而不是采用神经网络拟合。策略评估和改进是基于所学的概率模型。对于现实世界中一个推车双钟摆上翻（Cart-Double-Pendulum Swing Up）任务，PILCO 方法用仅 20 到 30 次尝试就能学会一个控制的有效策略，而其他方法像多层感知机可能最终需要至少几百次尝试的样本来学习一个动力学模型。然而，PILCO 方法也有它自己的问题，比如，由于学习策略参数是一个非凸优化问题，难以保证能搜索到最优控制方式，而且高斯过程的求解无法扩展到复杂模型的高维参数空间上。

  通过解决存在的缺陷来设计更加高效的学习算法。上述两种方法尝试通过利用额外信息来解决学习效率问题，如专家示范数据和环境建模信息。如果没有额外信息可以利用或环境的动态模型难以准确学到，那么我们就应该改进算法本身的学习效率而不利用额外信息。强化学习算法根据它们的更新方式一般分为两类：**在线策略**（On-Policy）和**离线策略**（Off-

Policy），如之前章节中所介绍的。在线策略方法对策略的评估有较小的偏差（Bias）但有较大的方差（Variance），而离线策略方法可以利用一个较大的随机采样批来实现较小的估计方差。

近年来，更加先进和有效算法被不断提出。多数算法是针对一些传统算法中的特定缺陷。比如，为了减小策略梯度的方差，Critic 网络被引入来估计 Actor-Critic 的动作–价值函数（Action-Value Function）；为了将强化学习任务从小规模扩展到大规模，DQN 采用了深度神经网络来改进基于表格（Tabular-based）的 Q-Learning 算法；为了解决 DQN 更新规则中使用最大化算子造成的过估计问题，Double DQN 算法使用了一个额外的 Q 网络；为了促进探索，基于参数噪声的 Noisy DQN 被提出，柔性 Actor-Critic（Soft Actor-Critic，缩写为 SAC）对策略的概率分布采用自适应熵；为了将 DQN 方法从只能解决离散任务扩展到连续任务，深度确定性策略梯度算法（Deep Deterministic Policy Gradient，缩写为 DDPG）被提出；为了稳定 DDPG 算法的学习过程，孪生延迟 DDPG（Twin Delayed DDPG，缩写为 TD3）提出用额外的网络和延迟更新的方式来优化策略；为了确保在线策略强化学习策略优化的安全更新，基于信赖域的算法像信赖域策略优化算法（Trust Region Policy Optimization，缩写为 TRPO）被提出；为了缩减 TRPO 二阶优化方法的计算时间，近端策略优化（Proximal Policy Optimization，缩写为 PPO）算法采用一阶近似；为了加速二阶自然梯度下降方法，使用 Kronecker 因子化信赖域的 Actor-Critic 算法（Actor Critic Using Kronecker-Factored Trust Region，缩写为 ACKTR）提出在二阶优化过程中使用 Kronecker 因子化（Kronecker-Factored）方法近似逆 Fisher 信息矩阵；最大化后验策略梯度（Maximum A Posteriori Policy Optimization，MPO）(Abdolmaleki et al., 2018) 算法和它的在线策略变体 V-MPO (Song et al., 2019) 用一种“强化学习作为推理”的观点实现策略优化。MPO 使用概率推理工具，像期望最大化算法（Expectation Maximization，EM）来优化最大熵强化学习目标。以上的算法只是整个强化学习算法领域发展的一小部分，我们希望读者到文献中查找更多改进算法学习效率和其他缺陷的强化学习算法。与此同时，所提出的强化学习算法结构变得越来越复杂，有更多灵活的参数可以被自适应地学习或人为选择，而这需要在强化学习研究中对其进行更加细致的考虑。有时额外的超参数可以显著改进学习表现，但有时它们使得学习过程更加敏感，而你需要对具体情况具体分析。

在上面例子中，我们假设数据样本包含丰富信息，而只是强化学习算法的学习效率较低。实践中，经常见到样本缺乏有用信息的情况，尤其是稀疏奖励的任务。比如，对于单个二值变量表示任务成功与否的情况来说，中间样本可能全部都是直接奖励（Immediate Reward）值为 0，从而没有任何区分度。这些样本中的信息自然就很稀疏。像这样的情况，在没有充分的奖励函数指引的情况下，有效探索空间的方式可能就很关键。像后见之明经验回放（Hindsight Experience Replay）(Andrychowicz et al., 2017)，分层学习结构 (Kulkarni et al., 2016)、内在奖励（Intrinsic Reward） (Sukhbaatar et al., 2018)、好奇心驱使的探索 (Pathak et al., 2017) 和其他有效的探索机制 (Houthooft et al., 2016) 都被用于一些工作中。强化学习中的学习效率由

于强化学习的固有性质被探索过程显著地影响，而有效的探索可以通过采集到更有信息的样本而提高从样本中学习的效率。由于探索是强化学习中的另一个巨大挑战，它将在后续小节之一中被单独讨论。

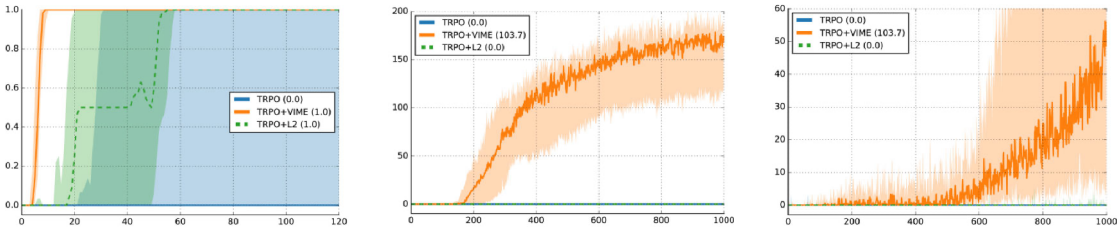
7.2　学习稳定性

深度强化学习可能非常不稳定或有随机性。这里的“不稳定”指，在多次训练中，每次学习表现在随时间变化的横向比较中的差异。随时间变化的不稳定，学习过程体现为有巨大的局部方差或在单次学习曲线上的非单调增长，比如有时学习表现甚至由于某些原因会下降。在多次训练中，不稳定的学习过程体现为在每一个阶段上的多次学习表现之间的巨大差异，而这将导致横向对比中的巨大方差。

深度神经网络的不稳定性和不可预测性在深度强化学习领域被进一步加剧，移动的目标分布、数据不满足独立同分布条件、对价值函数的不稳定的有偏差估计等因素导致了梯度估计器中的噪声，而进一步造成不稳定的学习表现。不同于监督学习在固定的数据集上学习（这里不考虑批限制的强化学习），强化学习经常是从高度相关的样本中学习的。比如，学习智能体大多采用策略探索得到的样本，要么是用在线策略学习的当前策略，要么是离线策略学习的先前策略（有时甚至是其他策略）。智能体和环境之间连续交互产生的样本可能是高度相关的，这打破了有效学习神经网络的独立性条件。由于价值函数是由当前策略选择的轨迹估计的，价值函数和估计它的策略之间也有依赖关系。由于策略随训练时间改变，参数化的价值函数的优化流形也随时间改变。考虑到为了便于在训练中探索，策略往往具有一定的随机性，价值函数于是更加难以追寻，而这也会导致用来学习的数据不满足独立同分布条件。不稳定的学习过程主要是由策略梯度或价值函数估计的变化造成的。然而，有偏差估计是强化学习中不稳定表现的另一根源，尤其是当偏差本身也不稳定的时候。举例来说，回想第 2 章，为了实现用 $Q^w(s,a)$ 对动作价值函数 $Q^\pi(s,a)$ 进行的无偏差估计，可兼容函数拟合条件（Compatible Function Approximation Condition）需要被满足。同时，有一些其他条件来确保价值函数的无偏差估计，以及一些进一步的要求条件来保证高级强化学习算法对策略改进有正确且准确的梯度计算。然而，实践中，这些要求或条件经常被放宽，而导致对价值函数的不稳定有偏差估计，或者策略梯度中较大的方差。多数情况下，人们讨论强化学习算法中估计的偏差和方差之间的权衡，而不稳定的偏差项本身也可能促成不稳定的学习表现。也有一些其他因素会导致不稳定的学习表现，比如探索策略中的随机性、环境中的随机性、数值计算的随机种子等。

论文 (Houthooft et al., 2016) 提出了以 Variational Information Maximizing Exploration（VIME）作为一种应用于一般强化学习算法中的探索方式。一些学习表现展示于他们所做的算法比较中，在三种不同的环境上使用 TRPO 或 TRPO+VIME 算法的学习结果基本上在学习曲线上都显示出了较大的方差，如图 7.1 所示。对于环境 MountainCar 来说，TRPO 算法的学习曲线能够覆盖整个奖励值范围 $[0,1]$，而且对 TRPO+VIME 方法在 HalfCheetah 环境也是类似的情况。我们需要注

意相比于其他一些强化学习算法，TRPO 在多数情况下已经是一个相对稳定的算法，它使用对梯度下降的二阶优化和信赖域限制。其他算法像 DDPG 可能在训练过程中表现得更加不稳定，有噪声的探索甚至可能在训练了较长一段时间后显著降低学习表现 (Fujimoto et al., 2018)。



(a) MountainCar (b) HalfCheetah (c) CartPoleSwingup

图 7.1 VIME 实验中的学习曲线。图片改编自文献 (Houthooft et al., 2016)（见彩插）

强化学习过程中的随机性会给准确评估算法表现带来困难，而这也显示出使用不同随机种子获得平均结果的重要性。

先前关于强化学习的调研 (Henderson et al., 2018) 中给出了一些关于深度强化学习实验中不稳定性和敏感性相关的结论：

- 策略网络结构可以对 TRPO 和 DDPG 算法的结果有显著影响。
- 对于策略网络或价值网络的隐藏层，ReLU 或 Leaky ReLU 激活函数往往在多个环境和多个算法上有最好的表现。而这个效果的大小对不同算法或环境不一致。
- 奖励值缩放的效果对不同环境和不同缩放值不一致。
- 5 个随机种子（通常的报告设置）可能不足以论证显著的结果，因为如果你仔细挑选随机种子，不同的随机种子可能得到完全不重合的置信区间，即使采用完全相同的实现方式。
- 环境动态的稳定性可能严重影响强化学习算法的学习表现。比如，一个不稳定的环境可以迅速削弱 DDPG 算法的有效学习表现。

人们已经有很长一段时间在尝试解决强化学习中的稳定性问题。为了解决累计奖励函数在原始 REINFORCE 算法中的较大方差，价值函数拟合被引入来估计奖励值。进一步地，动作价值函数也被用于奖励函数近似，这降低了方差，即使它可能是有偏差的。像这样方法构成了深度强化学习算法的主流——结合 Q-Learning 和策略梯度（Policy Gradient）方法，如之前第 6 章中所介绍的。在原始 DQN (Mnih et al., 2013) 中，使用目标网络和延迟更新，以及经验回放池帮助缓解了不稳定学习的问题。通常一个深度函数拟合器需要多次梯度更新而不是单次更新来达到收敛，而目标网络给学习过程提供了一个稳定的目标，这有助于在训练数据上收敛。在某种程度上，它可以满足同分布条件，而强化学习在没有目标网络时会将其打破。经验回放池给 DQN 提供了一种离线策略的学习方式，而从回放池中随机采样到的训练数据更接近于独立同分布数据，这也有助于稳定学习过程。更多关于 DQN 的细节在第 4 章中有所介绍。此外，TD3 算法（在第 6 章中介

绍）在 DQN 的稳定技术上应用目标策略平滑正则化（Target Policy Smooth Regularization）方法，基于相似动作有相似值的平滑性假设，从而在动作目标价值的估计中加入噪声，以减小方差。同时，TD3 使用了一对 Critic 而不是像 DDPG 中的一个，而这进一步稳定了学习表现。另一方面，对于基于策略梯度的方法来说，TRPO 使用二阶优化通过更全面的信息提供更稳定的更新，以及使用对更新后策略的限制来保证其保守但稳定的进步。

然而，即使有了以上工作，不稳定性、随机性和对初值及超参数的敏感性都使得强化学习研究人员在不同任务上评估算法和复现结果有一定困难，而这仍旧是强化学习社区的一个巨大挑战。

7.3　灾难性遗忘

由于强化学习通常有动态的学习过程而非像监督学习一样在固定的数据集上学习，它可以被看作是追逐一个移动目标的过程，而数据集在整个过程不断被更新。比如，在第 2 章中我们介绍了在线策略价值函数 $V^{\pi}(s)$ 和动作价值函数 $Q^{\pi}(s,a)$，它们都是用当前策略 π 来估计的。但是策略在整个学习过程中都在更新，这会导致对价值函数的动态估计。尽管通过离线策略回放池可以用一个相对稳定的训练集来缓解这个问题，回放池中的样本仍旧随着智能体的探索过程而不断改变。因此，一个叫作**灾难性遗忘**（Catastrophic Interference 或 Catastrophic Forgetting）(Kirkpatrick et al., 2017) 的问题可能在学习过程中发生，尤其是当策略或价值函数是基于神经网络的深度学习方法时，这个问题描述了其在解决如上所述的增量学习过程中有较差能力的现象。新的数据经常使得已训练过的网络改变很多来拟合它，从而忘记网络在之前训练过程中所学到的内容，即使这些内容也是有用的。这是在强化学习方法中使用神经网络做拟合器的一种局限性。

相较于离线策略方式，自然的人类学习过程实际更接近于在线策略学习。人们每天都在实时地学习新事物而不是一直从记忆中学习。然而，在线策略强化学习方法仍旧在努力提高学习效率，并且企图防止灾难性遗忘的问题。基于信赖域的方法像 TRPO 和 PPO 对学习过程中更新策略的潜在范围做了限制，来保证稳定但相对缓慢的学习表现进步。对于在线策略学习，样本通常以相关联数据的形式被采集，这极大促使了灾难性遗忘的发生。因此，离线策略学习方法使用经验回放池来缓解这个问题，从而在某种程度上保留旧数据来学习。像优先经验回放（Prioritized Experience Replay）和后见之明经验回放（Hindsight Experience Replay）的技术作为更复杂和先进的方式被提出，按照回放池中数据的重要性或者其目标来使用数据。

灾难性遗忘也发生在学习过程分为几个阶段的情况中。比如，在模拟到现实的策略迁移过程中，策略通常需要在模拟环境中预训练而后利用现实世界数据微调。然而，实践中，两个过程可能使用不同的损失函数，而且损失函数可能不总是与整体强化学习目标一致。如在文献 (Jeong et al., 2019a) 中，图像观察量被嵌入潜在表示而作为策略的输入，这个嵌入网络（Embedding Network）在模拟到现实的适应过程中通过一个自监督损失函数来微调，而非使用原来在模拟训练过程中的强化学习损失。这种在多阶段训练过程损失函数上的不匹配也可能在实践中造成灾难性遗忘，这

意味着策略可能遗忘预训练中获得的技能。为了解决这个问题，固定部分网络层并用之前的损失函数继续更新网络可以在后训练（Post-Training）过程中尽可能保持预训练的网络。另一个相似的想法是残差策略学习（Residual Policy Learning），如 8.6 节中所提到的，它也固定了预训练网络的权重并在旁边添加了一个新的网络来学习修正项。

7.4 探索

探索是强化学习中另一个主要的挑战，它会显著影响学习效率。相比于探索和利用间的权衡（Exploration-Exploitation Trade-Off）这个强化学习中经典且为人所知问题，这里着重于探索本身的挑战。强化学习中探索的困难可能来自稀疏的奖励函数、较大的动作空间和不稳定的环境，以及现实世界中探索的安全性问题等。探索意味着通过交互来获取更多关于环境的信息，通常与利用相对。利用指通过开发已知信息来最大化奖励。强化学习的学习过程基于试错。除非那些最优的轨迹在之前被探索过，否则最优的策略无法被学到。举例来说，雅达利游戏像 OpenAI Gym 中的 Montezuma's Revenge、Pitfall 由于探索的困难，对于一般强化学习算法会很难解决，这几个游戏的场景如图 7.2 所示，其中通常包括一个复杂的迷宫，需要较复杂的一系列操作来解决。它们像一个解迷宫的问题但是有着更复杂的结构和层次。Montezuma's Revenge 是一个非常典型的稀疏奖励任务，这使得强化学习的探索非常难以进行。在一个游戏场景中，Montezuma's Revenge 的智能体必须完成几十个连续动作来通过一个房间，而这个游戏有 23 个不同的房间场景需要智能体指导它自己通过。相似的情况在 Pitfall 游戏中也有。这些游戏常用作评估强化学习方法在探索能力方面的基准。OpenAI 和 Deepmind (Aytar et al., 2018) 都声称他们用高效的深度强化学习方法解决了 Montezuma's Revenge 游戏。然而，这些结果可能不令人满意。在他们的解决方案中，专

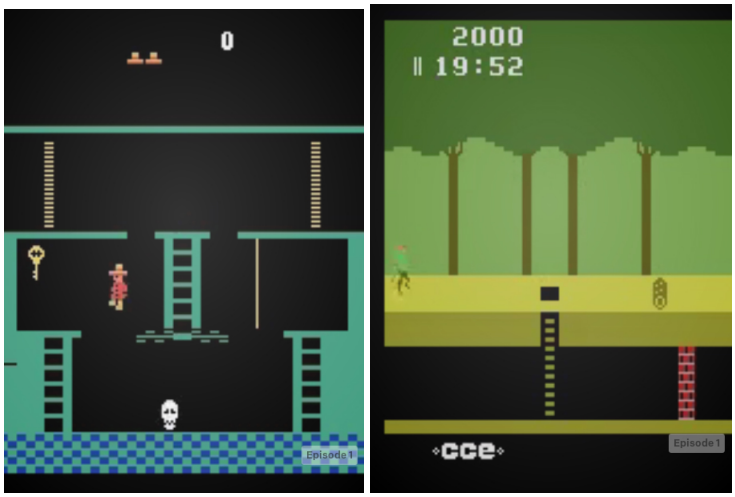


图 7.2 难以学习的雅达利游戏：Montezuma's Revenge（左）和 Pitfall（右）（见彩插）

家示范都被用于辅助探索。比如，在 Deepmind 的解决方案中，他们让智能体观察 YouTube 视频，而 OpenAI 使用人类示范来更好地初始化智能体位置。

这里稀疏奖励任务的瓶颈实际在于探索本身。稀疏奖励可能使价值网络和策略网络在一个不平滑且非凸的超曲面上优化，甚至在训练的某些阶段有不连续的情况。因此，一步优化后的策略可能无法帮助探索到更高奖励的区域。基于传统探索策略的智能体，比如随机动作或 ϵ-贪心（ϵ-Greedy）策略，会发现很难在探索过程中遇到高奖励值的轨迹。而即便它们采样到近最优（Near-Optimal）的轨迹，基于价值的或基于策略的优化方法可能也没有对这些样本充分重视，而导致失败情况或者缓慢的学习过程。上面描述的问题提出了当前深度强化学习方法的缺陷。

除稀疏奖励外，较大的动作空间和不稳定的环境也对强化学习智能体的探索造成困难。一个典型的例子是在文献 (Vinyals et al., 2019) 中解决的《星际争霸 II》（StarCraft）游戏。表 7.1[1] 中比较了雅达利游戏、围棋和《星际争霸》的信息类型、动作空间、游戏中的活动次数和玩家数量。大的动作空间和长的游戏控制序列使得在《星际争霸》中探索一个好的策略十分困难。此外，多玩家的设置使得对手在某种程度上成为游戏环境的一部分，这也增加了探索的难度。

表 7.1　对比不同的游戏

	雅达利游戏	围棋	《星际争霸》
信息类型	近完美	完美	不完美
动作空间	17	361	10^{26}
每场游戏的活动次数	100/s	100/s	1000/s
玩家数量	单个	两个	多个

为了解决探索的问题，研究人员调查了包括模仿学习、内在奖励（Intrinsic Reward）、分层学习等概念。通过模仿学习，智能体试图模仿来自人类或其他的专家示范来改进学习效率并减少探索到近最优样本的困难。内在奖励是基于这样的观念，即行为不仅是外在奖励的结果，而且也受到内在欲求的驱使，比如希望获得关于未知的更多有效信息。举例来说，婴儿可以通过好奇心驱使的探索很快地学习关于世界的知识。好奇心是一种内部驱动来改进智能体的学习，使其朝向更有探索性的策略改进。更多的内部驱动力需要在研究中探索。分层学习将复杂且难以探索的任务分解成小的子任务，这使其容易学习。举例来说，封建制网络（Feudal Network，FuN）作为封建制强化学习（Feudal Reinforcement Learning）中的一个关键方法使用了有管理者和工作者的层次性结构来解决 Montezuma's Revenge，实现更有效的探索和学习 (Vezhnevets et al., 2017)。

近年来，一些新方法被提出来解决探索问题，其中一个称为 Go-Explore，它不是一个深度强化学习的解决方案。Go-Explore 的主要想法是首先使用无神经网络的确定性训练来探索游戏世界，即不使用深度强化学习的方法，随后使用一个深度神经网络来模仿学习最好的轨迹，从而使得策略能够对环境的随机性鲁棒。为了解决大规模高度复杂游戏，比如《星际争霸 II》，DeepMind

[1]数据源：Oriol Vinyals, Deep Reinforcement Learning Workshop, NeurIPS 2019.

的研究人员 (Vinyals et al., 2019) 使用了基于族群的训练（Population-based Training，PBT）机制来有效探索全局最优策略，其中智能体集合成为联盟（League）。不同的智能体被初始化到策略分布中的不同集群（Clusters）上，来保证探索过程的多样性。基于族群的训练相比于单个智能体对策略空间有更充分的探索。

现实世界中的探索也与安全性问题相关。举例来说，当考虑一辆由智能体控制的自动驾驶车辆时，有车祸的失败情况也是智能体应该从中进行学习的。但是现实中一辆实际的车不可能被用来采集这些失败情况的样本，而使智能体以可接受的低损耗从中学习。现实的车辆甚至不能采用随机动作来探索，因为它可能导致灾难性的结果。相同的问题也存在于其他现实世界应用中，比如机器人操作、机器人手术等。为了解决这个问题，模拟到现实的转移（Sim-to-Real Transfer）的方法可以用于将强化学习部署到现实世界，它先在模拟中进行训练，再将策略转移到现实中。

7.5 元学习和表征学习

除改善一个具体任务上的学习效率外，研究人员也在寻求能够提高在不同任务上整体学习表现的方法，这与模型的**通用性**（Generality）和**多面性**（Versatility）相关。因此，我们会问，如何让智能体基于它所学习的旧任务来在新任务上更快地学习？而在这里可以介绍多个概念，包括元学习（Meta-Learning）、表征学习（Representation Learning）、迁移学习（Transfer Learning）等。

元学习的问题实际上可以追溯到 1980—1990 年 (Bengio et al., 1990)。近来深度学习和深度强化学习重新将这个问题带入我们的视野。许多令人兴奋的想法被提出，比如那些与模型无关的元学习（Model-Agnostic Meta-Learning）方法，以及一些更强大的跨任务学习方法在近年来都有快速发展。元学习的最初目的是让智能体解决不同问题或掌握不同技能。然而，我们无法忍受它对每个任务都从头学习，尤其是用深度学习来拟合的时候。**元学习**（Meta-Learning），也称**学会学习**，是让智能体根据以往经验在新任务上更快学习的方法，而非将每个任务作为一个单独的任务。通常一个普通的学习者学习一个具体任务的过程被看作是元学习中的内循环（Inner-Loop）学习过程，而元学习者（Meta-Learner）可以通过一个外循环（Outer-Loop）学习过程来更新内循环学习者。这两种学习过程可以同时优化或者以一种迭代的方式进行。三个元学习的主要类别为循环模型（Recurrent Model）、度量学习（Metric Learning）和学习优化器（Optimizer）。结合元学习和强化学习，可以得到**元强化学习**（Meta-Reinforcement Learning）方法。一种有效的元强化学习方法像与模型无关的元学习 (Finn et al., 2017) 可以通过小样本学习（Few-Shot Learning）或者几步更新来解决一个简单的新任务。

对于一个具体的任务领域，不同的任务之间可能有隐藏的关联性质。我们是否能让智能体从这个域内采样到的一些任务中学习这些潜在的规律，从而将所学到的内容泛化到其他任务上来更快地学习？这个学习潜在的关系或规律的过程与一个叫**表征学习**（Representation Learning）(Bengio et al., 2013) 的概念密切相关。表征学习起初在机器学习中提出，被定义为从原始数据中学习表示方式和提取有效信息或特征来便于分类器或预测器（比如强化学习中的策略）使用。表

征学习试图学习抽象且简洁的特征来表示原始材料，并且通过这种抽象，预测器或分类器不会降低它们的表现，而有更高的学习效率。学习隐藏的表示对于强化学习中提高学习效率十分有用，将这些规律迁移有利于在不同任务上的学习过程。表征学习通常可以用于学习强化学习环境中复杂状态的简单表示，这被称为**状态表征学习**（State Representation Learning，SRL）。这个表示包含在一个合适的抽象空间下的不变性和独特性特征，而这是从多样化的任务域中提炼出来的。举例来说，在一个拍摄物体运动的视频的一系列帧中，物体表面角上的关键点（或者物体表面上其他的特殊点）集合是对物体运动的一种恒定且鲁棒的表示，尽管帧中的像素点总是随着物体运动而改变。这些关键点有时在计算机视觉术语中称为描述器（Descriptors），它们存在一个描述器空间中。在这种表示方式下，这些关键点的位置在物体运动中将会改变，因此可以用来表示物体的运动。不同的物体有不同的关键点集合，因而也可以用来区分物体。强化学习中的表征学习对需要跨域的强化学习策略很重要，包括不同的任务域、模拟到现实的域迁移等。它是一个有希望且在探索中的方向，可以用于研究人类是如何利用知识进行规划的。

7.6　多智能体强化学习

在之前介绍的章节中，环境中只有一个智能体来寻找最优策略，这属于单智能体强化学习。除单智能体强化学习外，我们实际可以在同一个场景中设置多个智能体，来对多智能体策略进行同时探索，这个过程可以交替或者同时进行，称为多智能体强化学习（Multi-Agent Reinforcement Learning，MARL）。MARL 是一个有希望且值得探索的方向，提供了一种能够研究非常规强化学习情况的方式，包括群体智能、智能体环境的动态变化、智能体本身的创新等。

现代学习算法更多的是出色的受试者（Test-Takers），而非创新者。智能体的智能上限可能受到其所在环境的限制。因此，创新的产生成为人工智能（Artificial Intelligence，AI）中一个较热的话题。一种通向这个愿景的最有希望路径是通过多智能体的社会交互来学习。在多智能体学习中，智能体如何击败对手或与他人合作不是由环境的建造者决定的。举例来说，古老的围棋游戏的发明者从未定义什么策略能够击败对手，而对手通常也构成了动态环境的一部分。然而，在一代又一代人类玩家或人工智能体的自我演化过程中，大量先进的策略被发明出来，每个智能体作为其他人环境的一部分，而对自身的提高也构成他人的新挑战。

MARL 中结合传统的博弈论（Game Theory）和现代深度强化学习的方法近来在文献 (Lanctot et al., 2017; Nowé et al., 2012) 中有所探索，以及一些新的想法如自我博弈（Self-Play）(Berner et al., 2019; Heinrich et al., 2016; Shoham et al., 2003; Silver et al., 2018a)、优先虚拟自我博弈（Prioritized Fictitious Self-Play）(Vinyals et al., 2019)、基于族群的训练 (Population-Based Training，PBT）(Jaderberg et al., 2017; Vinyals et al., 2019) 和独立性强化学习（Independent Reinforcement Learning，InRL）(Lanctot et al., 2017; Tan, 1993)。MARL 不仅使得探索多智能体环境中的分布式智能成为可能，而且有助于在较大规模复杂环境中学习近最优或近平衡的智能体策略，比如，Deepmind 用于掌握游戏《星际争霸 II》的 AlphaStar，如图 7.3 所示。AlphaStar 的框架中用到了

PBT，通过使用一个联盟（League）的智能体，每一个智能体由图 7.3 中一个带索引值的色块来表示，这种训练方式被用来保证在策略空间的充分探索。在 PBT 中，策略优化的单位不再是每个智能体的单一策略，而是整个联盟的智能体。整体策略不仅关于一个具体策略，而更是整个联盟中智能体的整体表现。更多关于 MARL 的内容在第 11 章中有详细介绍。

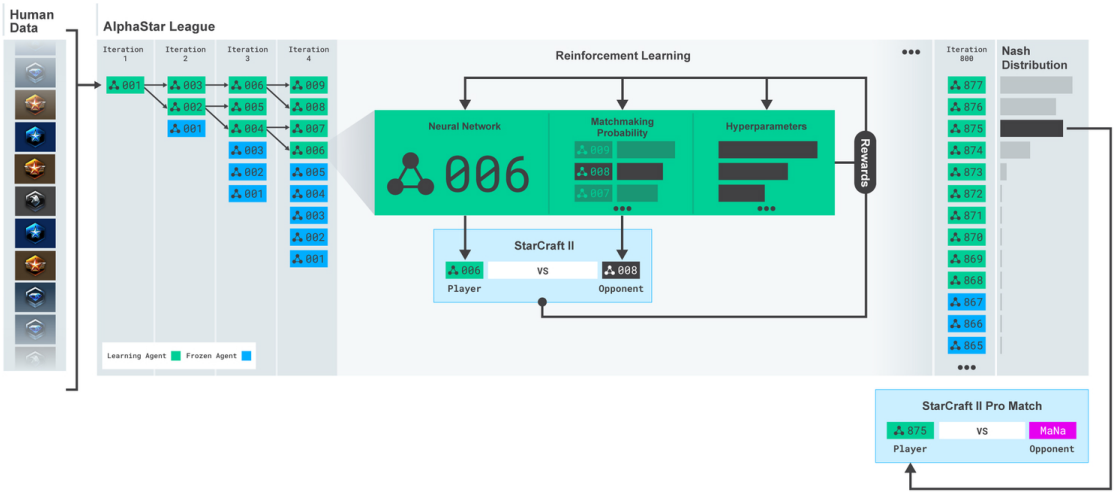


图 7.3 AlphaStar 的训练机制。每个小方块表示一个 AlphaStar 联盟中训练的智能体（见彩插）

7.7 模拟到现实

强化学习方法可以成功地解决大量模拟环境中的任务，甚至在一些具体领域可以超过最好的人类表现，比如围棋游戏。然而，应用强化学习方法到现实任务上的挑战仍旧未被解决。除了雅达利游戏、策略性计算机游戏、纸牌游戏，强化学习在现实世界中的潜在应用包括机器人控制、车辆自动驾驶、无人机自动控制等。这些涉及现实世界中硬件的任务通常对安全性和准确性有较高要求。对于这些情况，一个误操作可能导致灾难性后果。当策略是通过强化学习方法学到的时候，这个问题就更加值得考虑，因为即便不考虑现实世界的采样效率，学习智能体的探索过程也会有巨大影响。现代工业中的机器控制仍旧严重依赖传统控制方法，而非最先进的机器学习或强化学习解决方案。然而，用一个聪明的智能体来控制这些物理机械仍旧是一个很好的追求，而大量相关领域的研究人员正为之努力。

近年来，深度强化学习被逐渐应用到越来越多的控制问题中。但是由于强化学习算法较高的样本复杂度以及其他一些物理限制，许多在模拟中展示的能力尚未在现实世界中复现。我们主要通过机器人学习的例子来展示这些内容，而这是一个越发活跃的研究方向，吸引了来自学术界和工业界的关注。

指导性策略搜索（Guided Policy Search，GPS）(Levine et al., 2013) 是一种能够直接用真实机器

人在有限时间内训练的算法。通过所学线性动态模型进行轨迹优化，这个方法能够以较少的环境交互学会复杂的操作技巧。研究人员也探索了用多个机器人进行并行化训练的方法 (Levine et al., 2018)。文献 (Kalashnikov et al., 2018) 提出能同时在 7 个真实机器人上进行分布式训练的 QT-Opt 算法，但是需要持续 4 个月的 800 个小时的机器人数据采样时间作为代价。他们成功示范了直接在现实世界部署的机器人学习，但是其时间消耗和资源上的要求一般是无法接受的。更进一步来说，直接在物理系统上训练策略的成功例子尚且只在有限的领域得到验证。

模拟到现实迁移（Sim-to-Real Transfer）则是可以替代直接在现实中训练深度强化学习智能体的方法，由于模拟性能的提升和一些其他原因，模拟到现实迁移的方法比之前受到更多注意。相比于直接在现实世界中训练，模拟到现实迁移可以通过在模拟中快速学习来实现。近年来，许多模拟到现实的方法成功将强化学习智能体部署到现实中 (Akkaya et al., 2019; Andrychowicz et al., 2018)。然而，相比于直接在现实环境中部署训练过程，模拟到现实的方法也有它本身的缺陷，这主要由模拟和现实环境的差异造成，称为现实鸿沟（Reality Gap）。在实践中有大量因素会导致现实鸿沟，而这由具体系统而定。举例来说，系统动力学过程的差异将导致模拟和现实的动力学鸿沟，如图 7.4 所示是一个例子。不同的方法被提出来解决模拟到现实迁移的问题，后续还会介绍。

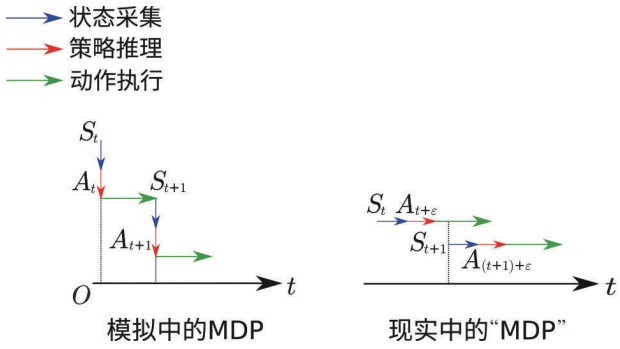


图 7.4　图片展示了模拟和现实中 MDP 的差异，它是由状态采集和策略推理过程产生的时间延迟造成的，这是造成现实鸿沟的可能因素之一（见彩插）

我们首先要理解现实鸿沟的概念。现实应用中的现实鸿沟可以在某种程度上用文献 (Jeong et al., 2019b) 中的图 7.5 来理解，该图展示了机器人上模拟轨迹和现实轨迹的差异，以及模拟和参考信号的差异。对于强化学习进行机器人控制任务来说，参考信号是发送给智能体的控制信号，从而在机械臂的关节角度上获得预期的行为。由于延迟、惯性和其他动力学上的不准确性，模拟和现实中的轨迹都会与参考信号有显著差异。此外，现实中的轨迹与模拟中的不同就是现实鸿沟。图中的系统识别（System Identification）是一种确认系统中动力学参数值的方法，可以用在策略或者模拟器中来缩减模拟动力学过程和现实的差异。泛化力模型（Generalized Force Model，GFM）是一个在论文 (Jeong et al., 2019b) 中新提出的方法，可以用额外的力来校正模拟器，从而生成与现实更接近的模拟轨迹。然而，即使使用了识别和校正的方法，现实鸿沟依然可能存在，

从而影响策略从模拟到现实中迁移。

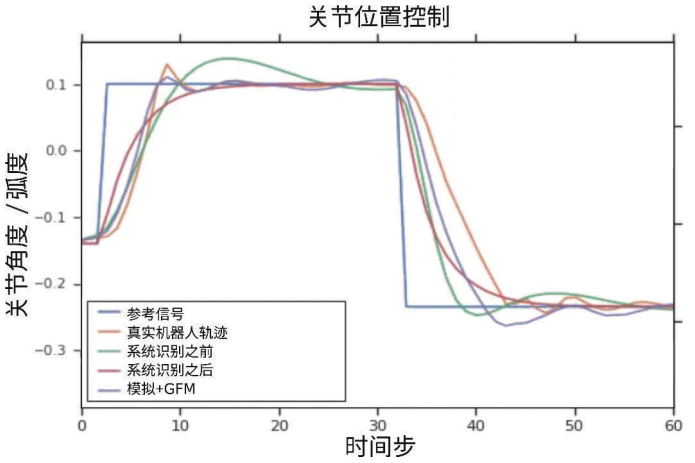


图 7.5 在一个简单的关节角度控制过程中，机器人控制的参考信号、模拟和现实中的差异。图片改编自文献 (Jeong et al., 2019b)（见彩插）

除了由于不同动力学过程导致的每一个时间步上模拟现实轨迹的差异，现实鸿沟也有其他来源。比如，在连续的现实世界控制系统中，有系统响应时间延迟或系统观察量构建过程耗时，而这些在有离散时间步的理想模拟情况下可能都不存在。如图 7.4 所示，在模拟环境或传统强化学习设置下，状态采集和策略推理过程都认为是始终没有时间损耗的，而在现实情况下，这两个过程都可能需要相当的时间，这使得智能体总是根据先前动作执行时的先前状态产生的滞后观察量来进行动作选择。

上面的问题也会使得模拟和现实的轨迹展现出不同的模式，如图 7.6 所示。考虑一个物体操作任务，即使我们假定有很快的神经网络前向过程（Forward Process）而忽略策略推理的时间消耗，现实世界中物体位置也可能需要一个摄像机来捕捉并用一些定位技术来追踪，而这需要相当的时间来处理。这个观察量构建的过程会引入时间延迟，从而即使在完全相同的控制信号下，现实轨迹和模拟轨迹的对比图上也会展示出时间间隙。这类**延迟观察量**使得现实世界中的强化学习智能体只能够接受先前观察量 O_{t-1} 来对当前步做出动作选择 A_t，而非直接根据当前状态 S_t。因此实践中的策略根据时间延迟 δ 通常会有形式 $\pi(A_t|O_{t-\delta})$，而这不同于模拟中根据实时观察量训练的策略，从而会产生较差的现实表现。一种解决这个问题的方式是修改模拟器，使其有相同的时间延迟，从而训练智能体去学习。然而，这会导致其他的问题，比如如何精确地表示和测量模拟和现实中的时间延迟，如何保证基于延迟观察量学习的智能体的表现等。近来，文献 (Ramstedt et al., 2019) 提出了实时强化学习方法，文献 (Xiao et al., 2020) 提出了“边运动边思考（Thinking While Moving）”的方法，在连续时间 MDP 设置下减轻了强化学习对于实时环境中延迟观察量和并发动作选择（Concurrent Action Choices）的问题，使得在现实世界中的控制轨迹更加平滑。

如上所述，从强化学习角度来看模拟到现实迁移的主要问题在于：在模拟中训练得到的策略

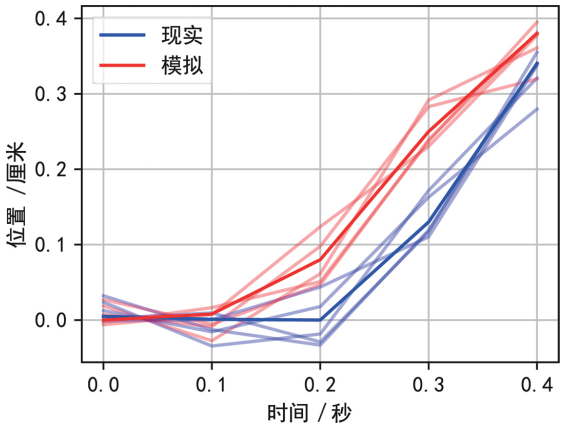


图 7.6　图片展示了物体观察状态（位置）在同一控制信号下的时间延迟。由于现实中额外的观察量构建过程，现实世界轨迹（下方）相比于模拟轨迹（上方）有一定延迟。不同的线体现了多次测试结果，加粗的线为均值（见彩插）

由于现实鸿沟不能在现实世界中始终正常使用，这个现实鸿沟即模拟和现实的差异。由于这个模型的差异，模拟环境中的成功策略无法很好地迁移到相应的现实中。总体来说，解决模拟到现实迁移的方法可以分为至少两个大类：零样本（Zero-Shot）方法和自适应学习方法。将控制策略从模拟迁移到现实的问题可以被看作是**域自适应**（Domain Adaption）的一个例子，即将一个在源域（Source Domain）中训练的模型迁移到新的目标域（Target Domain）。这些方法背后的一个关键假设是不同的域有公共的特征，从而在一个域中的表征方式和行为会对其他域有用。域自适应要求新的域中的数据适应预训练的策略。在新的域中获取数据的复杂性或困难程度，比如在现实世界中收集样本，这种自适应学习方法因而需要有较高的效率。像元学习 (Arndt et al., 2019; Nagabandi et al., 2018)、残差策略学习（Residual Policy Learning）(Johannink et al., 2019; Silver et al., 2018b) 和渐进网络（Progressive Networks）(Rusu et al., 2016a,b) 等方法被用于这些情形。**零样本**（Zero-Shot）迁移是一个与域自适应互补一类技术，它尤其适用于在模拟中学习。这意味着在迁移过程中没有任何基于现实世界数据的进一步学习过程。**域随机化**（Domain Randomization）是零样本迁移中典型的一类方法。通过域随机化，源和目标域的差异被建模为源域中的随机性。通过域随机化可以学到更普适的策略，而非过拟合到具体模拟器设置的特征策略。根据具体的应用，随机化可以被施加到不同的特征上。举例来说，对于机器人操作任务，摩擦力和质量的大小、力矩和速度的误差在实际机器人上都会影响到控制的精度。因此，在模拟器中这些参数可以被随机化，从而用强化学习训练一个更鲁棒的策略 (Peng et al., 2018)，这个过程称为动力学随机化（Dynamics Randomization）。在视觉域下的随机化可以用于直接将基于视觉的策略从模拟迁移到现实，而不需要任何现实的图像来训练 (Sadeghi et al., 2016; Tobin et al., 2017)。可能的视觉随机化的特征变量包括纹理、光照条件和物体位置等。

现实鸿沟通常是依赖于具体任务的，它可能由动力学参数或者动力学过程的定义不同造成。

除了动力学随机化 (Peng et al., 2018) 或视觉特征（观察量）随机化，还有一些其他方法来跨越现实鸿沟。利用系统识别（System Identification）来学习一个对动力学敏感（Dynamics-Aware）的策略 (Yu et al., 2017; Zhou et al., 2019) 是一个有希望的方向，它试图学习一个以系统特征为条件的策略，这些系统特征包括动力学参数或者轨迹的编码。也有一些方法来最小化模拟与现实的差异，比如之前介绍的 GFM 方法用于进行力校正，等等。模拟到现实通过模拟到模拟（Sim-to-Real via Sim-to-Sim）(James et al., 2019) 是另一个跨过现实鸿沟的方法，它使用随机到标准自适应网络（Randomized-to-Canonical Adaptation Networks, RCANs）来将随机的或现实世界图像转化成它们同等的非随机的标准型，而与模拟环境中的类似。渐进网络 (Rusu et al., 2016a) 也可以用于模拟到现实迁移 (Rusu et al., 2016b)，这是一个普适的框架，重复利用任何低级视觉特征到高级策略中，从而迁移到新的任务上，它以一种组合式但是简单的方法来构建复杂技能。

当今的计算框架利用离散的基于二值运算的计算过程，因此在某种程度上，我们应当始终承认模拟和现实世界的差异。这是因为后者在时间和空间上是连续的（至少在经典物理系统中）。只要学习算法不足够高效而能够直接像人脑一样应用于现实世界（或者即便可以实现），在模拟环境中得到一些预训练模型也总是有用的。如果模型在一定程度上有对现实环境的泛化能力就会更好，而这是模拟到现实迁移算法的意义。换句话说，模拟到现实迁移算法提供了始终考虑到在现实鸿沟下的学习模型方法论，而无关于模拟器本身有多精确。

7.8 大规模强化学习

如前面小节中所讨论的，强化学习在现实世界的应用目前遭遇到的如延迟观察量、域变换等问题，通常属于现实鸿沟的问题范畴。然而，也有其他一些因素阻止了强化学习的应用，或在模拟情况下，或在现实世界中。最有挑战性的问题之一是强化学习的**可扩展性**（Scalability），尽管深度强化学习利用了深度神经网络的通用表达能力，而这提出了大规模强化学习的挑战。

我们可以首先看一些例子。在像掌握大规模实时计算机游戏的应用中，如《星际争霸 II》（*StarCraft*）和《刀塔 2》（*Dota*），DeepMind 和 OpenAI 的团队分别提出了 AlphaStar (Vinyals et al., 2019) 和 OpenAI Five (Berner et al., 2019) 方法。在 AlphaStar 中，深度强化学习和监督学习（比如，模仿学习中的行为克隆）都被用于一个基于族群的训练（Population-Based Training，PBT）框架中，以及用到高级网络结构如 Scatter Connections、Transformer 和 Pointer 网络，这使得深度强化学习在整个策略中实际上只占一小部分。在 AlphaStar 中最终解决任务的关键步骤是如何高效地从存在的示范数据中学习和使用预训练的策略，作为强化学习智能体的初始状态，以及如何有效地结合来自联盟中不同智能体的不同次优策略。在 OpenAI Five 中，一个自我博弈（Self-Play）的框架被用于训练，而非 PBT 框架，但它也使用了从人类示范中模仿学习的方法。上述事实说明，在多数情况下，当前的深度强化学习算法本身对于完美地从端到端去解决一个大规模任务可能仍旧是不足够有效且高效的。一些其他技术如模仿学习等通常需要被用来解决这些大规模问题。

此外，并行训练框架也常于解决大规模问题。举例来说，在解决现实中机器人学习的算法 QT-

Opt (Kalashnikov et al., 2018) 中，为了实现并行的机器人采样，它应用了一个包含在线和离线数据的经验回放缓存，以及分布式训练工作者来高效地从缓存数据中学习。一个分布式或并行的采样和训练框架对于解决这类大规模问题很关键，尤其是对高维的状态和动作空间。文献 (Espeholt et al., 2018) 提出了重要性加权的行动者-学习者结构（Importance Weighted Actor-Learner Architecture，IMPALA），而文献 (Espeholt et al., 2019) 提出了可扩展高效深度强化学习（Scalable, Efficient Deep-RL，SEED）来实现大规模分布式强化学习。另外，强化学习的分布式框架通常与不同计算设备（比如 CPU 和 GPU）间的平衡有关，如第 18 章中所讨论的。在强化学习算法方面，异步优势 Actor-Critic（Asynchronous Advantage Actor-Critic，A3C）(Mnih et al., 2016)、分布式近端策略优化（Distributed Proximal Policy Optimizaion，DPPO）(Heess et al., 2017)、循环缓存分布式 DQN（Recurrent Peplay Distributed DQN，R2D2）(Kapturowski et al., 2019) 等算法在近年来被提出，来更好地支持强化学习中的并行采样和训练。更多关于强化学习中并行计算的内容在第 12 章中有所介绍。

7.9　其他挑战

除了上面提到的（深度）强化学习中的挑战，也有一些其他挑战，比如深度强化学习的可解释性 (Madumal et al., 2019)、强化学习应用的安全性问题 (Berkenkamp et al., 2017; Garcıa et al., 2015)、相关理论中复杂度证明 (Koenig et al., 1993; Lattimore et al., 2013) 中的困难、强化学习算法的效率 (Jin et al., 2018) 和收敛性质 (Papavassiliou et al., 1999)，以及理解清楚强化学习方法在整个人工智能中的作用和角色等。这些内容超出本书范畴，有兴趣的读者可以自行探索这些领域的前沿。

在本章最后，我们引用 Richard Sutton[2]的一些话，**“我们从这些痛苦的教训中应当学到的一点是通用型（General Purpose）模型的力量，即那些能够随着计算能力提升而不断扩展的方法，它们甚至到极其巨大的计算量时也能工作。有两个看起来能够以这种方式任意扩展的方法是搜索和学习。”**这些话基于这样的观察，即在计算机象棋或计算机围棋，以及像语音识别和计算机视觉等领域上的以往成功，一般的统计性方法（如神经网络）胜过了基于人类知识的方法。因此，智能系统中的嵌入式知识可能只能在较短时间内满足研究人员，而在长期阻碍了通用人工智能的整体发展过程。**“第二个从痛苦的教训中学到的东西是大脑中实际的内容是极其复杂的，且这种复杂性是不可更改的；我们应当停止寻找简单的方式来考虑大脑中的内容，比如用简答的方式考虑空间、物体、多个智能体或对称性。所有的这些都是任意的、本质上复杂的外在环境的部分。它们不是我们应当嵌入的东西，因为它们的复杂度是无穷的；相反，我们应当只构建元方法来找到和采集这种任意的复杂度。”**这句话阐释了提出元方法来自然地处理世界的复杂度的重要性，而非使用人为构建的、有具体用途的、相对简单的认知结构和决策机制。

[2]Richard S. Sutton. “The Bitter Lesson.”March 13, 2019.

参考文献

ABDOLMALEKI A, SPRINGENBERG J T, TASSA Y, et al., 2018. Maximum a posteriori policy optimisation[J]. arXiv preprint arXiv:1806.06920.

AKKAYA I, ANDRYCHOWICZ M, CHOCIEJ M, et al., 2019. Solving rubik's cube with a robot hand[J]. arXiv preprint arXiv:1910.07113.

ANDRYCHOWICZ M, WOLSKI F, RAY A, et al., 2017. Hindsight experience replay[C]//Advances in Neural Information Processing Systems. 5048-5058.

ANDRYCHOWICZ M, BAKER B, CHOCIEJ M, et al., 2018. Learning dexterous in-hand manipulation[J]. arXiv preprint arXiv:1808.00177.

ARNDT K, HAZARA M, GHADIRZADEH A, et al., 2019. Meta reinforcement learning for sim-to-real domain adaptation[J]. arXiv preprint arXiv:1909.12906.

AYTAR Y, PFAFF T, BUDDEN D, et al., 2018. Playing hard exploration games by watching youtube[C]// Advances in Neural Information Processing Systems. 2930-2941.

BENGIO Y, BENGIO S, CLOUTIER J, 1990. Learning a synaptic learning rule[M]. Université de Montréal, Département d'informatique et de recherche opérationnelle.

BENGIO Y, COURVILLE A, VINCENT P, 2013. Representation learning: A review and new perspectives[J]. IEEE transactions on pattern analysis and machine intelligence, 35(8): 1798-1828.

BERKENKAMP F, TURCHETTA M, SCHOELLIG A, et al., 2017. Safe model-based reinforcement learning with stability guarantees[C]//Advances in Neural Information Processing Systems. 908-918.

BERNER C, BROCKMAN G, CHAN B, et al., 2019. Dota 2 with large scale deep reinforcement learning[J]. arXiv preprint arXiv:1912.06680.

DEISENROTH M, RASMUSSEN C E, 2011. Pilco: A model-based and data-efficient approach to policy search[C]//Proceedings of the 28th International Conference on Machine Learning (ICML-11). 465-472.

ESPEHOLT L, SOYER H, MUNOS R, et al., 2018. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures[J]. arXiv preprint arXiv:1802.01561.

ESPEHOLT L, MARINIER R, STANCZYK P, et al., 2019. Seed rl: Scalable and efficient deep-rl with accelerated central inference[J]. arXiv preprint arXiv:1910.06591.

FINN C, ABBEEL P, LEVINE S, 2017. Model-agnostic meta-learning for fast adaptation of deep networks[C]//Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org: 1126-1135.

FUJIMOTO S, VAN HOOF H, MEGER D, 2018. Addressing function approximation error in actor-critic methods[J]. arXiv preprint arXiv:1802.09477.

GARCIA J, FERNÁNDEZ F, 2015. A comprehensive survey on safe reinforcement learning[J]. Journal of Machine Learning Research, 16(1): 1437-1480.

HEESS N, SRIRAM S, LEMMON J, et al., 2017. Emergence of locomotion behaviours in rich environments[J]. arXiv:1707.02286.

HEINRICH J, SILVER D, 2016. Deep reinforcement learning from self-play in imperfect-information games[J]. arXiv:1603.01121.

HENDERSON P, ISLAM R, BACHMAN P, et al., 2018. Deep reinforcement learning that matters[C]// Thirty-Second AAAI Conference on Artificial Intelligence.

HOUTHOOFT R, CHEN X, DUAN Y, et al., 2016. Vime: Variational information maximizing exploration[Z].

JADERBERG M, DALIBARD V, OSINDERO S, et al., 2017. Population based training of neural networks[J]. arXiv preprint arXiv:1711.09846.

JAMES S, WOHLHART P, KALAKRISHNAN M, et al., 2019. Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 12627-12637.

JEONG R, AYTAR Y, KHOSID D, et al., 2019a. Self-supervised sim-to-real adaptation for visual robotic manipulation[J]. arXiv preprint arXiv:1910.09470.

JEONG R, KAY J, ROMANO F, et al., 2019b. Modelling generalized forces with reinforcement learning for sim-to-real transfer[J]. arXiv preprint arXiv:1910.09471.

JIN C, ALLEN-ZHU Z, BUBECK S, et al., 2018. Is q-learning provably efficient?[C]//Advances in Neural Information Processing Systems. 4863-4873.

JOHANNINK T, BAHL S, NAIR A, et al., 2019. Residual reinforcement learning for robot control[C]// 2019 International Conference on Robotics and Automation (ICRA). IEEE: 6023-6029.

KALASHNIKOV D, IRPAN A, PASTOR P, et al., 2018. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation[J]. arXiv preprint arXiv:1806.10293.

KAPTUROWSKI S, OSTROVSKI G, DABNEY W, et al., 2019. Recurrent experience replay in distributed reinforcement learning[C]//International Conference on Learning Representations.

KIRKPATRICK J, PASCANU R, RABINOWITZ N, et al., 2017. Overcoming catastrophic forgetting in neural networks[J]. Proceedings of the national academy of sciences, 114(13): 3521-3526.

KOENIG S, SIMMONS R G, 1993. Complexity analysis of real-time reinforcement learning[C]// Proceedings of the AAAI Conference on Artificial Intelligence. 99-107.

KULKARNI T D, NARASIMHAN K, SAEEDI A, et al., 2016. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation[C]//Advances in Neural Information Processing Systems. 3675-3683.

LANCTOT M, ZAMBALDI V, GRUSLYS A, et al., 2017. A unified game-theoretic approach to multiagent reinforcement learning[C]//Advances in Neural Information Processing Systems. 4190-4203.

LATTIMORE T, HUTTER M, SUNEHAG P, et al., 2013. The sample-complexity of general reinforcement learning[C]//Proceedings of the 30th International Conference on Machine Learning. Journal of Machine Learning Research.

LEVINE S, KOLTUN V, 2013. Guided policy search[C]//International Conference on Machine Learning. 1-9.

LEVINE S, PASTOR P, KRIZHEVSKY A, et al., 2018. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection[J]. The International Journal of Robotics Research, 37(4-5): 421-436.

MADUMAL P, MILLER T, SONENBERG L, et al., 2019. Explainable reinforcement learning through a causal lens[J]. arXiv preprint arXiv:1905.10958.

MNIH V, KAVUKCUOGLU K, SILVER D, et al., 2013. Playing atari with deep reinforcement learning[J]. arXiv preprint arXiv:1312.5602.

MNIH V, BADIA A P, MIRZA M, et al., 2016. Asynchronous methods for deep reinforcement learning[C]//International Conference on Machine Learning (ICML). 1928-1937.

NAGABANDI A, CLAVERA I, LIU S, et al., 2018. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning[J]. arXiv preprint arXiv:1803.11347.

NOWÉ A, VRANCX P, DE HAUWERE Y M, 2012. Game theory and multi-agent reinforcement learning[M]//Reinforcement Learning. Springer: 441-470.

PAPAVASSILIOU V A, RUSSELL S, 1999. Convergence of reinforcement learning with general function approximators[C]//International Joint Conference on Artificial Intelligence: volume 99. 748-755.

PATHAK D, AGRAWAL P, EFROS A A, et al., 2017. Curiosity-driven exploration by self-supervised prediction[C]//Proceedings of the International Conference on Machine Learning (ICML).

PENG X B, ANDRYCHOWICZ M, ZAREMBA W, et al., 2018. Sim-to-real transfer of robotic control with dynamics randomization[C]//2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE: 1-8.

RAMSTEDT S, PAL C, 2019. Real-time reinforcement learning[C]//Advances in Neural Information Processing Systems. 3067-3076.

RUSU A A, RABINOWITZ N C, DESJARDINS G, et al., 2016a. Progressive neural networks[J]. arXiv preprint arXiv:1606.04671.

RUSU A A, VECERIK M, ROTHÖRL T, et al., 2016b. Sim-to-real robot learning from pixels with progressive nets[J]. arXiv preprint arXiv:1610.04286.

SADEGHI F, LEVINE S, 2016. Cad2rl: Real single-image flight without a single real image[J]. arXiv preprint arXiv:1611.04201.

SHOHAM Y, POWERS R, GRENAGER T, 2003. Multi-agent reinforcement learning: a critical survey[J]. Web manuscript.

SILVER D, HUBERT T, SCHRITTWIESER J, et al., 2018a. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play[J]. Science, 362(6419): 1140-1144.

SILVER T, ALLEN K, TENENBAUM J, et al., 2018b. Residual policy learning[J]. arXiv preprint arXiv:1812.06298.

SONG H F, ABDOLMALEKI A, SPRINGENBERG J T, et al., 2019. V-mpo: On-policy maximum a posteriori policy optimization for discrete and continuous control[J]. arXiv preprint arXiv:1909.12238.

SUKHBAATAR S, LIN Z, KOSTRIKOV I, et al., 2018. Intrinsic motivation and automatic curricula via asymmetric self-play[C]//International Conference on Learning Representations.

TAN M, 1993. Multi-agent reinforcement learning: Independent vs. cooperative agents[C]//Proceedings of the International Conference on Machine Learning (ICML).

TOBIN J, FONG R, RAY A, et al., 2017. Domain randomization for transferring deep neural networks from simulation to the real world[C]//ROS.

VEZHNEVETS A S, OSINDERO S, SCHAUL T, et al., 2017. Feudal networks for hierarchical reinforcement learning[C]//Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org: 3540-3549.

VINYALS O, BABUSCHKIN I, CZARNECKI W M, et al., 2019. Grandmaster level in starcraft ii using multi-agent reinforcement learning[J]. Nature, 575(7782): 350-354.

XIAO T, JANG E, KALASHNIKOV D, et al., 2020. Thinking while moving: Deep reinforcement learning with concurrent control[J]. arXiv preprint arXiv:2004.06089.

YU W, TAN J, LIU C K, et al., 2017. Preparing for the unknown: Learning a universal policy with online system identification[J]. arXiv preprint arXiv:1702.02453.

ZHOU W, PINTO L, GUPTA A, 2019. Environment probing interaction policies[J]. arXiv preprint arXiv:1907.11740.