# 第9章
# 深度强化学习与规划

周炜星 谢文杰

华东理工大学金融学系

2023年秋

# 纲要

# 纲要

# 规划

- 规划是指智能体并不与真实环境进行交互，而是基于智能体构建的环境模型来产生模拟数据，基于模拟数据完成对值函数和策略函数的更新和优化。
- 在智能体规划过程中，需要建模环境模型，能够进行虚拟的行为交互。在围棋博弈中，对弈者不需要真正的落子也能在大脑中模拟落子后对方的行动以及自己可采取的动作。
- 一般来说，对弈者能在大脑中模拟对弈的步数越多，其围棋对弈水平就越高。
- 为什么对弈双方能够做到呢？因为围棋对弈时人类大脑中有围棋规则，而围棋规则就是围棋对弈的环境模型，人类能够通过理解和应用围棋规则，完成假想的对弈行为。
- 对弈者通过在大脑中模拟对弈结果，获得真实环境中最优策略。

# 纲要

# 深度强化学习模型分类

- 深度强化学习模型一般可以分成两大类，一类是基于值函数（Value-based）的深度强化学习模型，如DQN等；
- 另一类是基于策略函数（Policy-based）的深度强化学习模型，如TRPO、PPO等。
- 基于Aactor-Critic框架的深度强化学习模型将两者结合起来，融合了策略梯度优化算法和值函数优化算法的优点，使得模型训练更加高效，产生了A2C、A3C、DDPG和TD3等算法。

# 深度强化学习模型分类

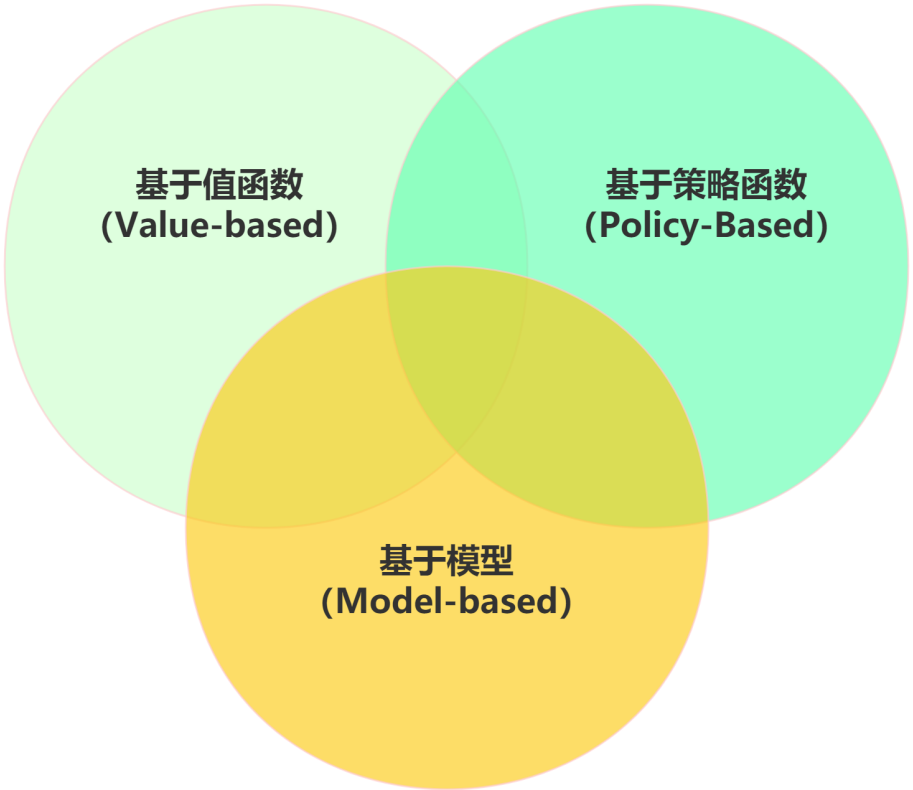



图 1: 基于值函数、基于策略和基于模型的深度强化学习的关系示意图

# 深度强化学习中的学习模块

- 深度强化学习的关键是学习模块。
- 深度强化学习模型的基础是马尔科夫决策过程（MDP）。
- MDP可以表示成一个五元组 $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$，其中$\mathcal{S}$ 表示状态集合，$\mathcal{A}$ 表示动作集合，$P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$表示状态转移函数，$P(s_t, a_t, s_{t+1})$是状态转移概率，$R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathcal{R}$是奖励函数。
- 在现实世界中，复杂决策问题的环境状态$\mathcal{S}$非常复杂，直接影响了智能体决策性能。
- 深度强化学习模型在围棋、视频游戏、机器人等领域中取得了非凡的效果，主要得益于智能体学习的环境模型与真实环境非常相似。

# 深度强化学习中的规划模块

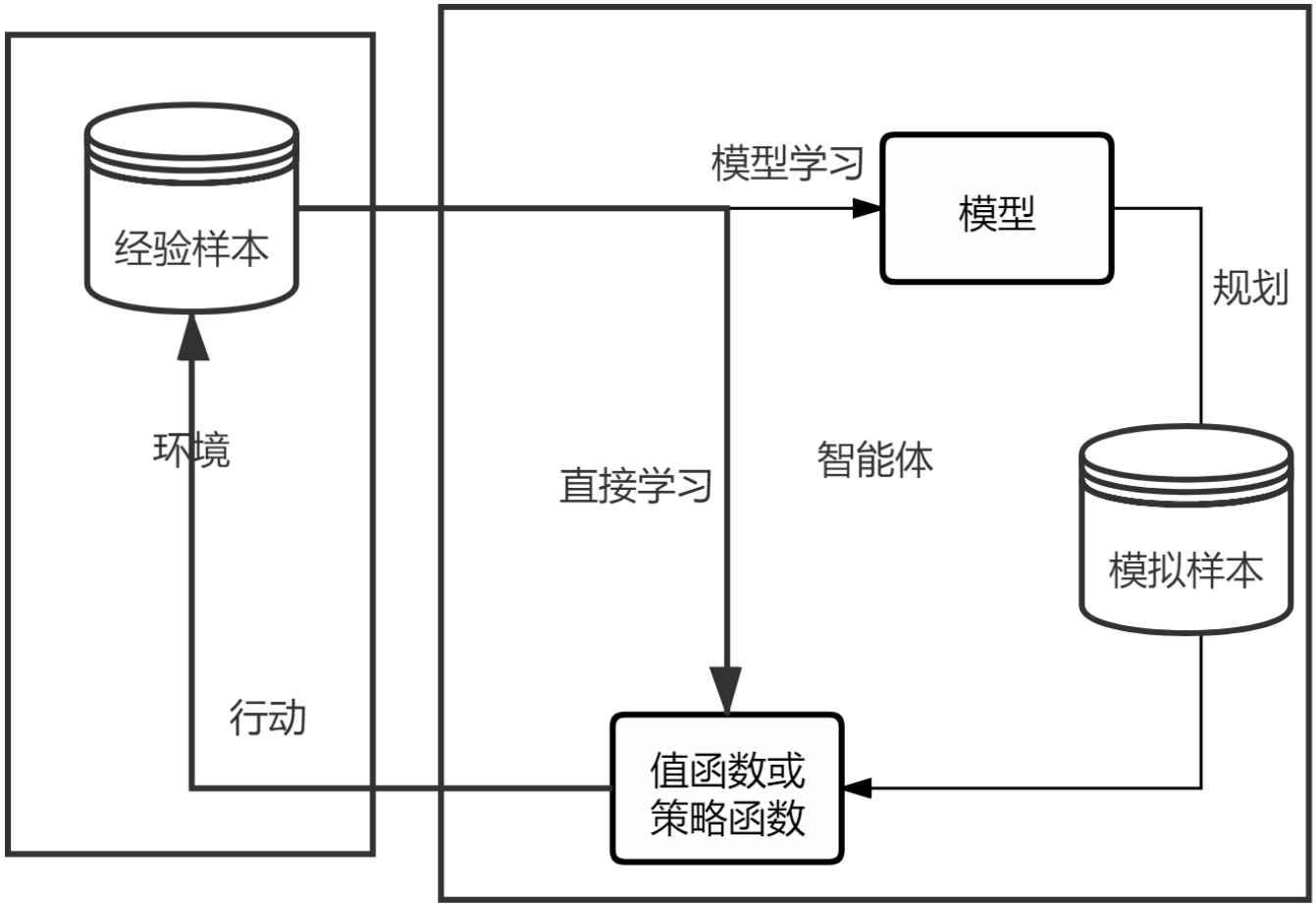



图 2: 基于模型的深度强化学习中智能体规划示意图

# 纲要

# Dyna框架简介

一般来说，状态转移函数$P$和奖励函数$R$作为环境模型的一部分。在基于模型的强化学习方法中，我们可以参数化状态转移函数$P$和奖励函数$R$：

$$\begin{aligned} s_{t+1} &= P_w(s_t, a_t) \\ r_{t+1} &= R_w(s_t, a_t). \end{aligned} \tag{1}$$

此模型中的即时奖励只和当前状态和动作有关。在一些复杂环境模型中，即时奖励会与下一个状态相关，模型可以表示成：

$$\begin{aligned} s_{t+1} &= P_w(s_t, a_t) \\ r_{t+1} &= R_w(s_t, a_t, s_{t+1}). \end{aligned} \tag{2}$$

# Dyna框架简介

该公式建模了环境模型的映射关系，可以用深度神经网络模型表示复杂函数。环境模型的学习过程就是函数逼近过程，只要确定函数参数即可。在环境模型学习过程中，智能体可以采用监督学习方法拟合模型参数。模型拟合的样本数据同样是智能体与环境交互所获得轨迹数据，采样经验回放机制中的经验池存储经验轨迹样本数据，课表示成四元组形式：

$$(s_i, a_i, r_i, s_i'), \quad i = 1, 2, 3, \cdots, N, \tag{3}$$

其中，$N$为经验池的经验轨迹样本数量。环境模型的表示可以采用多种模型，包括查表式模型、线性期望模型、线性高斯模型、高斯决策模型和深度神经网络模型等。在实际应用中，我们一般采用深度神经网络模型。

# Dyna框架的模型学习

状态转移函数$P_{w_1}(s_t, a_t)$和奖励函数$R_{w_2}(s_t, a_t)$用深度神经网络表示，且参数分别为$w_1$和$w_2$。我们采用梯度下降算法进行优化，可以得到状态转移函数$P_{w_1}(s_t, a_t)$的估计参数

$$\hat{w}_1 = \arg\min_{w_1} \frac{1}{N} \sum_{t=0}^{N} \left(P_{w_1}(s_t, a_t) - s_{t+1}\right)^2. \tag{4}$$

奖励函数$R_{w_2}(s_t, a_t)$的估计参数同样可以表示为：

$$\hat{w}_2 = \arg\min_{w_2} \frac{1}{N} \sum_{t=0}^{N} \left(R_{w_2}(s_t, a_t) - r_{t+1}\right)^2. \tag{5}$$

# Dyna框架的模型学习

在深度强化学习中，环境模型的学习方法很多，上述公式只是简单的例子。在实际应用中，我们可以有很多更优选择，如将状态转移过程建模成随机模型，模型输出下一个状态的概率，通过采样来确定模型下一个状态：

$$\begin{aligned} s_{t+1} &\sim P_w(s_{t+1}|s_t, a_t) \\ r_{t+1} &\sim R_w(r_{t+1}|s_t, a_t). \end{aligned} \tag{6}$$

在基于模型的深度强化学习中，环境模型的学习过程可以转化成两个概率分布函数的逼近问题，采用KL散度作为目标函数进行梯度下降优化。监督学习的思想是无监督学习和强化学习的基础，被应用于深度强化学习不同方法，作为模型学习的基本思想，在复杂机器学习算法（如深度强化学习算法）的理解和改进方面具有重要意义。

# Dyna框架的模型学习

智能体完成模型学习后，运用学习到的环境模型$\hat{R}$和$\hat{P}$可以进行规划。智能体的规划过程是基于马尔科夫决策过程的采样过程。此时的马尔科夫决策过程五元组为$M=(S,A,\hat{R},\hat{P},\gamma)$，其中$\hat{R}$为智能体学习到的奖励函数，$\hat{P}$是智能体学习到的状态转移函数，$S$是模型状态空间，$A$是智能体动作空间。智能体在学习到的环境模型中进行采样，获得模拟数据，采用四元组形式进行保存：

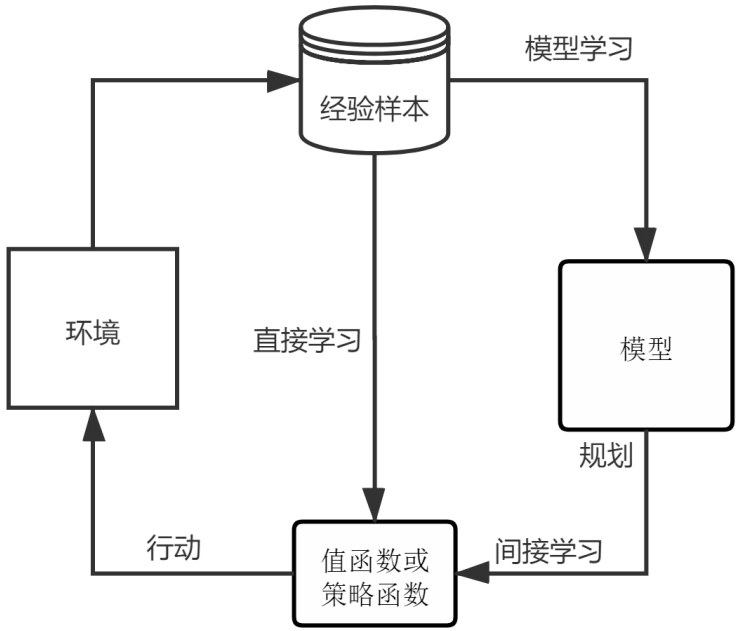
$$(\hat{s}_i,\hat{a}_i,\hat{r}_i,\hat{s}_i'),\quad i=1,2,3,...,n. \tag{7}$$

公式中$n$为模拟轨迹的样本数量。智能体基于规划得到的模拟轨迹样本数据和智能体与真实环境交互获得的经验轨迹数据共同训练智能体策略函数。

# 纲要

# Dyna-Q算法简介

Dyna是一个算法框架，融合了模型学习过程和策略函数学习过程。Dyna-Q算法融合了Q-learning算法和Dyna学习过程，智能体通过环境模型进行规划，生成模拟轨迹数据共同更新和改进策略函数。



# Dyna-Q算法简介

Dyna-Q算法融合了Q-learning算法，其智能体的主要任务是，智能体与环境交互获得经验轨迹样本数据训练状态-动作值函数。图3中的直接学习过程表明，智能体基于与环境交互获得的经验轨迹样本数据改进动作值函数，Dyna-Q算法采用$\epsilon$-贪心策略生成轨迹样本数据：

$$\pi(s,a)=\begin{cases}1-\epsilon+\frac{\epsilon}{|\mathcal{A}|}, & a=\arg\max_a Q(s,a)\\ \frac{\epsilon}{|\mathcal{A}|}, & a\neq\arg\max_a Q(s,a)\end{cases}. \tag{8}$$

经验轨迹数据表示为$\langle s,a,r,s'\rangle$，其中$a$是基于$\epsilon$-贪心策略产生的动作，$r$是环境返回的即时奖励，$s'$是环境返回的下一个状态。

# Dyna-Q算法简介

Dyna-Q算法采用经验轨迹数据$\langle s, a, r, s' \rangle$更新状态-动作值函数：

$$Q(s,a) \leftarrow Q(s,a) + \alpha(r + \gamma \max_{a'} Q(s',a') - Q(s,a)). \tag{9}$$

Dyna-Q算法也采用经验轨迹数据$\langle s, a, r, s' \rangle$更新环境模型：

$$\begin{aligned} P_{w_1}(s,a) &\xleftarrow{\text{update}} \langle s, a, s' \rangle \\ R_{w_2}(s,a) &\xleftarrow{\text{update}} \langle s, a, r \rangle \end{aligned} \tag{10}$$

# Dyna-Q算法简介

间接学习过程包含了模型学习和模型规划，智能体通过模型规划产生模拟样本：

$$\begin{aligned} \hat{s}' &\sim P_w(s|s,a) \\ \hat{r} &\sim R_w(r|s,a) \end{aligned} \tag{11}$$

在Dyna-Q算法的规划过程中，智能体从访问过的状态中选择一个状态$s$，并从状态$s$采用过的动作中随机选择一个动作$a$，利用模型规划产生下一个状态$\hat{s}'$和即时奖励$\hat{r}$，得到的模拟轨迹样本数据$\langle s, a, \hat{r}, \hat{s}' \rangle$。智能体基于模拟轨迹样本数据更新状态-动作值函数：

$$Q(s,a) \leftarrow Q(s,a) + \alpha(\hat{r} + \gamma \max_{a'} Q(\hat{s}', a') - Q(s,a)). \tag{12}$$

# Dyna-Q算法伪代码

**Algorithm 26:** 基于模型的强化学习算法 Dyna-Q 伪代码

**Input:** 状态空间 $\mathcal{S}$，动作空间 $\mathcal{A}$，折扣系数 $\gamma$
初始化的状态-动作值函数 $Q(s,a)=0$
初始化环境模型 $P_{w_1}(s_t,a_t)$ 和 $R_{w_2}(s_t,a_t)$
**Output:** 最优的状态-动作值函数 $Q(s,a)$ 和最优策略 $\pi^*$

**for** $k=0,1,2,\cdots$ **do**
  %% 每次循环针对一条轨迹
  初始化状态 $s$
  **for** $t=0,1,2,\cdots,T$ **do**
    采用 $\epsilon$-贪心策略：

$$\pi(s,a)=\begin{cases}1-\epsilon+\frac{\epsilon}{|\mathcal{A}|}, & a=\arg\max_a Q(s,a)\\ \frac{\epsilon}{|\mathcal{A}|}, & a\neq\arg\max_a Q(s,a)\end{cases} \tag{9.13}$$

    产生一步轨迹 $\langle s,a,r\rangle$，其中，$a$ 是基于 $\epsilon$-贪心策略产生的动作，$r$ 是环境返回的即时奖励。
    状态-值函数更新：

$$Q(s,a)\leftarrow Q(s,a)+\alpha(r+\gamma\max_{a'}Q(s',a')-Q(s,a)) \tag{9.14}$$

    模型更新：

$$\begin{aligned}P_{w_1}(s,a)&\xleftarrow{\text{update}}\langle s,a,s'\rangle\\ R_{w_2}(s,a)&\xleftarrow{\text{update}}\langle s,a,r\rangle\end{aligned} \tag{9.15}$$

    智能体进入下一个状态 $s=s'$
    **if** 模型 $P_{w_1}(s,a)$ 和 $R_{w_2}(s,a)$ 已学习到，可进行规划 **then**
      **for** $i=1:n$ **do**
        从访问过的状态中选择一个状态 $s$，
        从状态 $s$ 采用过的动作中随机选择一个动作 $a$，
        利用模型规划产生下一个状态 $s'$ 和即时奖励 $\hat{r}$：

$$\begin{aligned}\hat{s}'&\sim P_{w_1}(s|s,a)\\ \hat{r}&\sim R_{w_2}(r|s,a)\end{aligned} \tag{9.16}$$

        值函数更新：

$$Q(s,a)\leftarrow Q(s,a)+\alpha(\hat{r}+\gamma\max_{a'}Q(\hat{s}',a')-Q(s,a)) \tag{9.17}$$

    **if** $s$ 为终止状态 **then**
      开始下一条轨迹采样
  %% 计算最优策略
  **for** $s\in S$ **do**
    $\pi^*(s)=\arg\max_a Q(s,a)$

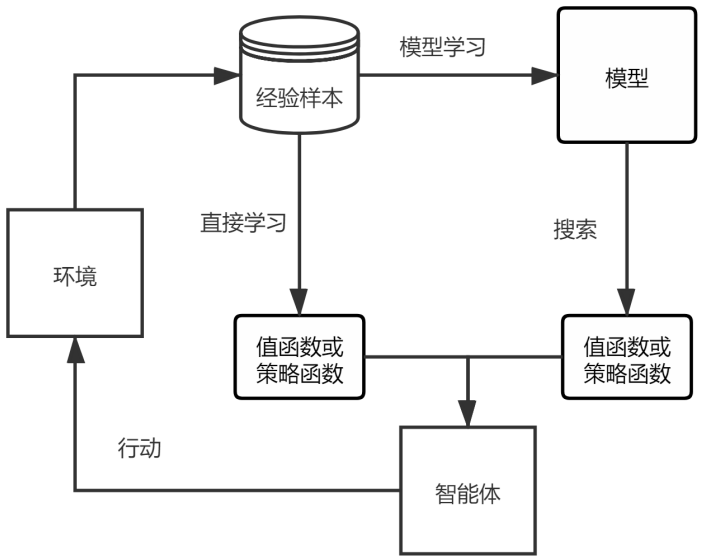
# 纲要

# Dyna-2框架简介

- Dyna框架从经验轨迹样本中学习环境模型，智能体基于模型进行规划，模拟生成轨迹样本数据。
- Dyna框架融合经验轨迹样本和模拟轨迹样本对智能体策略函数进行更新，模型性能得到了一定的提升。
- 2008年，David Silver 和其导师Sutton教授提出了Dyna-2算法框架。
- Dyna-2算法框架引入一个非常重要的概念：搜索。
- Dyna框架在智能体学习的基础上引入了规划。规划是指智能体并不实际与环境进行交互，而是与学习的环境交互生成模拟的轨迹数据，并基于模型产生的模拟数据完成对值函数和策略函数的更新和优化。

# Dyna-2框架简介

图4展示了融合学习和搜索的Dyna-2框架图。Dyna-2框架用搜索模块替换了Dyna框架中的规划模块。Dyna-2模型同时学习了两个值函数或者策略函数，一个值函数基于智能体与环境交互的经验轨迹样本学习所得，另一个是智能体利用学习到的环境模型进行搜索所得。我们介绍Dyna-2中搜索模块如何构建值函数或者策略函数。



# Dyna-2框架简介

- 在搜索过程中，智能体选定一个初始状态$s$，从初始状态$s$开始进行多次采样，可以构成一棵树，节点为状态，边为智能体在模拟动作后状态转移过程。
- 搜索算法需要找出初始状态$s$下的最优动作$a$。
- 一般可采用蒙特卡罗模拟，智能体从初始状态$s$出发，通过简单的随机策略多次模拟生成多条轨迹，分析多条轨迹估计不同状态-动作的值函数$Q(s,a)$。
- 蒙特卡罗模拟需要完整地模拟一条轨迹，然后计算初始状态$s$下不同动作的值函数$Q(s,a)$。
- 智能体通过$\max_{a'} Q(s,a')$选择价值最大的动作进行决策，然后智能体转移到下一个状态进行搜索。

# 纲要

# 编程实践模块介绍

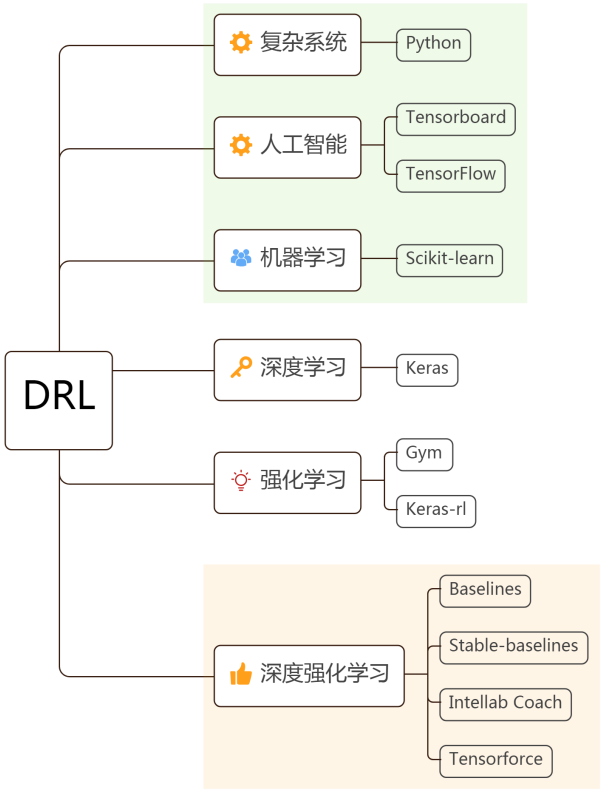



图 5: 深度强化学习工具

# 课后习题

1. 简述学习和规划的区别？
2. 模型无关和基于模型的强化学习方法主要区别是什么？
3. 举例一个搜索算法及其应用。
4. 简述Dyna框架。
5. 简述Dyna-Q算法。
6. 简述Dyna-2框架。
7. 对比不同深度强化学习开源代码库。