强化学习基础篇（三十三）Dyna算法

1、使用模型进行规划

基于模型的强化学习算法的主要成分可以分为学习（Learning）和规划（Planning）两个部分。学习是指从真实的经验轨迹数据集中学习环境模型$\mathcal{M}_\eta = \langle \mathcal{P}_\eta, \mathcal{R}_\eta \rangle$，即学习环境的马尔科夫决策过程$\text{MDP}\langle \mathcal{S}, \mathcal{A}, \mathcal{P}_\eta, \mathcal{R}_\eta \rangle$。

规划指基于环境模型，求解基于该模型的最有价值函数或最优策略，即求解该模型的马尔科夫决策过程。

具体而言，规划首先基于环境模型$\mathcal{M}_\eta = \langle \mathcal{P}_\eta, \mathcal{R}_\eta \rangle$生成大量的模拟经验轨迹数据，随后使用Model-free的方法（例如Policy Gradient, Value Iteration）从生成的模拟经验轨迹数据中学习价值函数或策略函数。

Sample-Based Planning

基于采样的规划（Sample-Based Planning）是实现规划的简单方法，其基于模型生成采样数据：

$$S_{t+1} \sim \mathcal{P}_\eta \left(S_{t+1} \mid S_t, A_t \right)$$
$$R_{t+1} = \mathcal{R}_\eta \left(R_{t+1} \mid S_t, A_t \right)$$

然后使用蒙特卡洛控制，Sarsa或Q-Learning算法进行学习。

Q-planning算法

为了直观理解规划过程，可以看看Q-planning算法的具体流程， Q-planning算法以基于表格的Q-learning算法为基础，并从环境中进行随机采样，又被称为基于表格的随机采样Q-planning算法，算法流程如下：

- （1）随机选择状态s和动作a，其中， $s \in S, a \in A(s)$；
- （2）将状态状态s和动作a输入环境模型$\mathcal{M}_\eta = \langle \mathcal{P}_\eta, \mathcal{R}_\eta \rangle$，环境模型$\mathcal{M}$返回奖励$r$以及下一个状态$s'$；
- （3）将模型经验(s, a, r, s')作为Q-learning算法的输入：

$$Q(s,a) \leftarrow Q(s,a) + [r + \gamma max_a Q(s',a) - Q(s,a)]$$

- （4）重复步骤（1）~步骤（3），直到获得理想的动作值函数或达到终止条件。

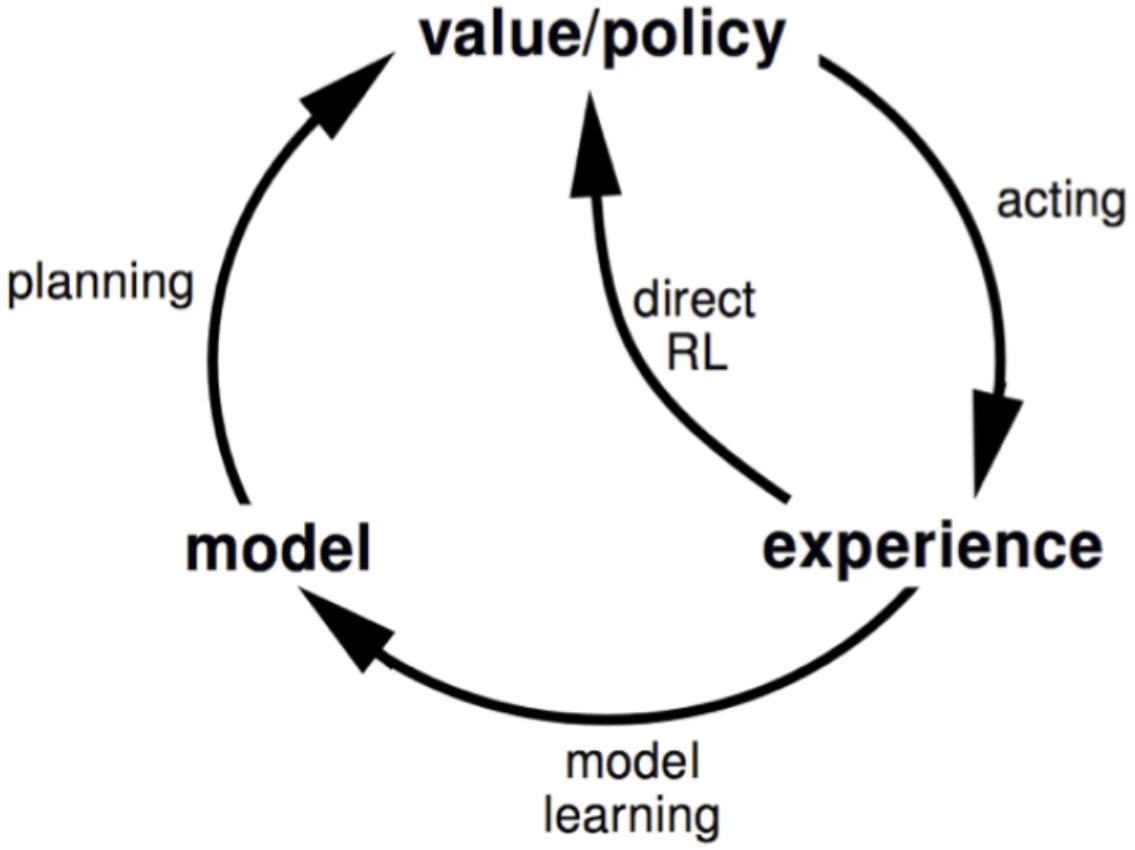
2、Dyna算法

之前说过学习（Learning）和规划（Planning），接下来介绍框架整合的Dyna算法，即同时包含学习过程和规划过程。在整合框架的更新价值函数的过程中，不仅使用环境模型生成的模拟经验数据：

$$S' \sim \mathcal{P}_\eta \left(S' \mid S, A \right)$$
$$R = \mathcal{R}_\eta (R \mid S, A)$$

同时也会使用与环境交互过程中获得的真实经验数据（$S' \sim \mathcal{P}^a_{s,s'}$，$R = \mathcal{R}^a_s$）。

其架构如下图：



框架中主要涉及的元素包括了：经验（Experience），模型（Model）以及价值/策略（value/policy）。其中经验主要由两方面用途：一方面用于环境模型的学习，并随后基于环境模型改进价值函数或者策略函数，该过程称为间接的强化学习；另一方面通过强化学习算法直接进行价值函数或者策略函数的更新，该过程称为直接强化学习（Direct RL）。

Dyna算法通过联合使用真实经验数据和模拟经验数据，能够在学习的过程中同时规划价值函数和策略函数，使得智能体在实际任务中获得更优的策略。

3、Dyna-Q算法

为了更好理解Dyan算法架构，这里给出基于表格的Dyna-Q算法的具体流程:

Initialize $Q(s,a)$ and $Model(s,a)$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}(s)$
Do forever:
(a) $S \leftarrow$ current (nonterminal) state
(b) $A \leftarrow \varepsilon$-greedy(S,Q)
(c) Execute action A; observe resultant reward, R, and state, S'
(d) $Q(S,A) \leftarrow Q(S,A) + \alpha\big[R + \gamma \max_a Q(S',a) - Q(S,A)\big]$
(e) $Model(S,A) \leftarrow R, S'$ (assuming deterministic environment)
(f) Repeat n times:
$S \leftarrow$ random previously observed state
$A \leftarrow$ random action previously taken in S
$R, S' \leftarrow Model(S,A)$
$Q(S,A) \leftarrow Q(S,A) + \alpha\big[R + \gamma \max_a Q(S',a) - Q(S,A)\big]$

步骤（a）到步骤（e）基于真实的经验数据，步骤（f）则基于模拟经验数据。需要注意的是，步骤（f）主要用于改进智能体的策略，改进的程度由重复次数n决定，n越大，智能体在每次迭代中策略提升得越为明显。

智能体首先从历史状态空间中随机采样一个状态S, 随后根据该状态S使用过的动作中随机采样一个动作A，并基于状态S与动作Q，利用环境模型获得新的状态S'和奖励R。最后，根据Q-learning算法更新动作值函数$Q(s,a)$。在下一轮迭代中，基于步骤（f）（即基于模拟经验数据）更新的动作值函数$Q(s,a)$可作为真实动作值函数计算的依据和指导，能够让智能体在实际环境中更快，更好地完成任务。