# Multi-Agent Reinforcement Learning:

# A Report on Challenges and Approaches

**Introduction：**

Reinforcement Learning (RL) refers to both the learning problem and sub-field of machine learning.假说，所有的目标和假设都是为了最大化期望奖赏。

**Hypothesis 1 (The Reward Hypothesis).** *That all of what we mean by goals and purposes can be well thought of as the maximization of the expected value of the cumulative sum of a received scalar signal (called reward).*

EC is different from RL in that there is no explicit interaction between the environment and the agent. 而许多当今社会很复杂的问题都可以抽象成多智能体学习问题。

**RL Methods：**

**Basic:**

**1. MDP**

**Definition 1 (Markov Decision Process).** *A Markov decision process is a tuple* $(\mathcal{S}, \mathcal{A}, \mathcal{R}, p)$ *such that*

$$p(s', r|s, a) = Pr\{S_t = s', R_t = r \mid S_{t-1} = s, A_{t-1} = a\} \tag{2}$$

*where* $S_t \in \mathcal{S}$ *(state space),* $A_t \in A$ *(action space),* $R_t \in \mathcal{R}$ *(reward space) and* $p$ *defines the dynamics of the process.*

**Definition 2 (Discounted Returns).** *Discounted Return is defined as the total sum of rewards following a time step t until the end of the sequence of rewards discounted by a factor* $\gamma$ *at each time step*

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots \tag{3}$$
$$G_t = R_{t+1} + \gamma G_{t+1}$$

*where* $R_i \in \mathcal{R}\ \forall i$ *and* $\gamma \in [0, 1]$.

**2. Value function**

**Definition 3 (Policy).** *A policy is defined as the probability distribution of actions at a given states.*

$$\pi(A_t = a \mid S_t = s)\ \forall S_t \in \mathcal{S} \tag{4}$$

*where* $A_t \in \mathcal{A}(s)$ *is the state specific action space.*

**Definition 4** (**State Value Function**). *Value function of a state s under policy π is defined as the expected return when starting in state s and following a policy π to take actions*

$$V^{\pi}(s) = \mathbb{E}_{\pi}\left[G_t \mid S_t = s\right] \ \forall s \in \mathcal{S} \tag{5}$$

**Definition 5** (**Action Value Function**). *Value function of a state s and action a under policy π is defined as the expected return when starting in state s, taking action a and following a policy π to take actions further.*

$$Q^{\pi}(s,a) = \mathbb{E}_{\pi}\left[G_t \mid S_t = s, A_t = a\right] \ \forall s \in \mathcal{S}, a \in \mathcal{A}(s) \tag{6}$$

### 3. Bellman equation

**Definition 6.** *The Bellman Expectation Equation for $V^{\pi}(s)$ is given by*

$$\begin{aligned} V^{\pi}(s) &= \mathbb{E}_{\pi}\left[G_t \mid S_t = s\right] \\ &= \mathbb{E}_{\pi}\left[R_{t+1} + \gamma G_{t+1} \mid S_t = s\right] \\ &= \sum_{a} \pi(a|s) \left[ \sum_{s',r} p(s', r|s, a) \left[ r + \gamma \mathbb{E}\left[G_{t+1} | S_{t+1} = s'\right]\right]\right] \\ &= \sum_{a} \pi(a|s) \left[ \sum_{s',r} p(s', r|s, a) \left[ r + \gamma V^{\pi}(s')\right]\right] \\ &= \mathbb{E}_{\pi}\left[R_{t+1} + \gamma V^{\pi}(S_{t+1}) \mid S_t = s\right] \end{aligned} \tag{7}$$

The above expectation considers all possibilities of actions by the policy and the induced states by those actions defined by the environment dynamics. Similarly, we have

**Definition 7.** *The Bellman Expectation Equation for $Q^{\pi}(s,a)$ is given by*

$$\begin{aligned} Q^{\pi}(s,a) &= \mathbb{E}_{\pi}\left[G_t \mid S_t = s, A_t = a\right] \\ &= \mathbb{E}_{\pi}\left[R_{t+1} + \gamma Q^{\pi}(S_{t+1}, A_{t+1}) \mid S_t = s, A_t = a\right] \end{aligned} \tag{8}$$

$$\mathbf{V}^{\pi} = \mathcal{R}^{\pi} + \gamma \mathcal{P}^{\pi} \mathbf{V}^{\pi} \tag{9}$$

$$\implies \mathbf{V}^{\pi} = (\mathbf{I} - \gamma \mathcal{P}^{\pi})^{-1} \mathcal{R}^{\pi} \tag{10}$$

where $\mathbf{V}^{\pi}$ represents the value vector for each node in a Markov decision process under a policy $\pi$ and $\mathcal{P}^{\pi}$ represents the state transition matrix. It should be noted that the runtime of this form is prohibitive in practice ($\mathcal{O}(n^3)$) and we will see practical solutions in §3. This also forms the solution for action value because of the following relation

$$V^{\pi}(s) = \sum_{a} \pi(a|s) Q^{\pi}(s,a) \tag{11}$$

**Definition 8.** *The Bellman Optimality Equation for $V^\star(s)$ is given by*

$$\begin{aligned} V^\star(s) &= \max_\pi V^\pi(s) \\ &= \max_a Q^\star(s,a) \\ &= \max_a \sum_{s',r} p(s',r|s,a)\left[r + \gamma V^\star(s')\right] \end{aligned} \tag{12}$$

**Definition 9.** *The Bellman Optimality Equation for $Q^\star(s,a)$ is given by*

$$\begin{aligned} Q^\star(s,a) &= \max_\pi Q^\pi(s,a) \\ &= \sum_{s',r} p(s',r|s,a)\left[r + \gamma \max_{a'} Q^\star(s',a')\right] \end{aligned} \tag{13}$$

**Methods:**

两种迭代方法解决动态规划问题的有价值迭代和策略迭代，但前提条件是智能体了解环境动态信息。

今天的大多数强化学习技巧可以被归为对于状态、动作对表格的估计和学习状态、动作的选择策略。

**1. Q-learning:**

目标是最小化期望 loss 以学习最好的状态动作对。

$$\mathcal{L}(\theta) = \mathbb{E}_\pi\left[(Q_\theta(s,a) - y)^2\right]$$

where $y = r + \gamma \max_{a'} Q_{\theta'}(s',a')$. $y$ represents the Q-Learning target value.

该方法通过大量采样 trajectory 进行估计，同时为了覆盖所有的策略可能进行ε探索或者玻尔兹曼探索

Boltzmann Exploration：基于 Q 值概率化，通过概率采样选择动作。

$$P(a|s) = \frac{exp(Q(s,a))}{\sum_a exp(Q(s,a))}$$

https://www.jianshu.com/p/dcfd927e7598

另一种用于稳定性能和克服灾难性遗忘问题的技术是使用经验回放缓冲区，从该缓冲区中随机采样转换。同时也能破坏从 policy 产生的 trajectory 的相关性，能更泛化地学习。

**2. PG methods:**

They explicitly learn a stochastic policy distribution $\pi\theta$ parameterized by $\theta$. If we denote the reward of a trajectory $\tau$ generated by policy $\pi\theta(\tau)$ as $r(\tau)$ :

$$J(\theta) = \mathbb{E}_{\pi_\theta}[r(\tau)] = \int \pi_\theta(\tau) r(\tau) d\tau \tag{15}$$

$$\begin{aligned} \nabla J(\theta) &= \int \nabla \pi_\theta(\tau) r(\tau) d\tau \\ &= \int \pi_\theta(\tau) \nabla \log \pi_\theta(\tau) r(\tau) d\tau \\ &= \mathbb{E}_{\pi_\theta}[\nabla \log \pi_\theta(\tau) r(\tau)] \end{aligned} \tag{16}$$

Now, the probability of generating the trajectory is

$$\pi_\theta(\tau) = \mathcal{P}(s_0) \prod_{t=1}^{T} \pi_\theta(a_t|s_t) p(s_{t+1}|s_t, a_t) \tag{17}$$

where $\mathcal{P}$ refers to the ergodic distribution of the MDP. In simple terms, it is the probability of the agent being in some initial state. Taking the log and the gradient reveals a surprisingly beautiful result

$$\log \pi_\theta(\tau) = \log \mathcal{P}(s_0) + \sum_{i=1}^{T} \log \pi_\theta(a_t|s_t) + \log p(s_{t+1}|s_t, a_t) \tag{18}$$

$$\nabla \log \pi_\theta(\tau) = \sum_{i=1}^{T} \nabla \log \pi_\theta(a_t|s_t)$$

$$\implies \nabla J(\theta) = \mathbb{E}_{\pi_\theta}\left[\nabla\left(\sum_{t=1}^{T} \log \pi_\theta(a_t|s_t)\right) r(\tau)\right]$$

we can now just run Monte-Carlo simulations and approximate the gradient to find the best parameters θ*.

**3. AC:**

一个有助于减少 PG 方差的想法是最大化一个 PG 目标，这个目标使用相对回报的差异，即优势函数。

$$\nabla J(\theta) = \mathbb{E}_{\pi_\theta}\left[\nabla\left(\sum_{t=1}^{T}\log\pi_\theta(a_t|s_t)\right)(G_t - b)\right] \tag{19}$$

where $b$ is the introduced baseline.

$$\begin{aligned}\mathbb{E}_{\pi_\theta}\left[\nabla\left(\sum_{t=1}^{T}\log\pi_\theta(a_t|s_t)\right)b\right] &= \int\sum_{t=1}^{T}\pi_\theta(a_t|s_t)\nabla\log\pi_\theta(a_t|s_t)bd\tau \\ &= \int\nabla\sum_{t=1}^{T}\pi_\theta(a_t|s_t)bd\tau \\ &= \int\nabla\pi_\theta(\tau)bd\tau \\ &= b\nabla\int\pi_\theta(\tau)d\tau \\ &= b\nabla 1 = 0\end{aligned} \tag{20}$$

以上计算式表示增加 baseline 可以保证无偏同时又减小了方差。现代解决这个问题的方法是使用另一个参数化价值函数作为 baseline，通常被称为 critic。

目标函数和 baseline 的差被叫做优势函数。

**4. Deterministic PG:**

该方法考虑一个确定性策略形式为 a = μθ(s). 因此，DPG 就是方差为 0 的随机策略方法受限的一种情况。

$$\nabla_\theta J(\mu_\theta) = \mathbb{E}\left[\nabla_\theta\mu_\theta(s)\nabla_a Q^\mu(s,a)\big|_{a=\mu_\theta(s)}\right] \tag{21}$$

This result is surprising as well because this also does not depend on the environment dynamics. The policy is decided by solving the following maximization problem.

$$\mu^{k+1}(s) = \underset{a}{argmax}\, Q^{\mu^k}(s,a) \tag{22}$$

However, this is computationally hard to do at each decision step. Instead, multiple gradients are averaged from multiple trajectories to reduce variance in the estimate. This work has since been extended by [Lillicrap et al., 2015] to be more stable by using the idea of Experience Replay Buffer and Target Network updates to keep the target network constant for few gradient steps.

**Deep RL：For Multi-Agent: MADRL**

**1. Joint Action Space**

$$\mathcal{T} : \mathcal{S} \times \mathcal{A}_1 \times \mathcal{A}_2 \times \ldots \times \mathcal{A}_n \rightarrow PD(\mathcal{S}) \quad (23)$$

$$R_i : \mathcal{S} \times \mathcal{A}_1 \times \mathcal{A}_2 \times \ldots \times \mathcal{A}_n \rightarrow \mathcal{R} \quad (24)$$

$$\pi_{\theta_i} : \mathcal{S} \times \mathcal{A}_1 \times \mathcal{A}_2 \times \ldots \times \mathcal{A}_n \rightarrow [0, 1] \quad (25)$$

where PD(S) represents the probability distribution over the resultant state space. all the routines above are now exponentially dependent on the action space.

**2. Game-Theoretic Effects (博弈论效应)**

在马尔科夫游戏中，没有一个可支配的确定型决策，因为对手的决策是不确定的。

举例一个随机策略游戏，剪刀石头布，reward 如下图所示，o 是对手的 action，a 是 agent 的 action：

| | | Agent | | |
|---|---|---|---|---|
| | | rock | paper | scissors |
| Opponent | rock | 0 | 1 | -1 |
| | paper | -1 | 0 | 1 |
| | scissors | 1 | -1 | 0 |

线性约束的问题预期回报政策π和总价值回报 V 如下：

$$\pi_{\text{paper}} - \pi_{\text{scissors}} \geq V \quad (27)$$

$$-\pi_{\text{rock}} + \pi_{\text{scissors}} \geq V \quad (28)$$

$$\pi_{\text{rock}} - \pi_{\text{paper}} \geq V \quad (29)$$

$$\pi_{\text{rock}} + \pi_{\text{paper}} + \pi_{\text{scissors}} = 1 \quad (30)$$

Linear Programming gives a solution for this as $\pi = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ and $V = 0$.

剪刀石头布的最优策略为随机策略。

彻底理解多智能体设置下的学习问题仍然是一个开放性问题。**在多智能体的环境下，评估学习不要对于收敛到纳什均衡过分看重？？？**

然而, Evolutionary Game Theory (进化博弈论)是首选框架。

**3. Credit Assignment (信用分配) and Lazy Agent (懒惰智能体) Problem**

信用分配关心的是系统的成功该如何归因于系统的各部分的贡献。

由部分可观测而引起的另一个现象叫做懒惰智能体。当一个智能体学会了一个有用的策略，另一个智能体就会减少探索，因此学习信息被第一个智能体占用并利用了。部分可观测使得一个 agent 过分学习，一个 agent 欠学习。

**4. Non-Markov Nature of Environments 非马尔可夫环境性质**

非马尔可夫性质的环境意味着需要利用以前时间的信息，处理这个问题的方法有使用循

环网络，允许使用过去的状态历史。

这个问题的现代处理方法采用了相同的方法，即使用门控神经网络或卷积神经网络来构建状态序列的隐藏表示。

Decentralized Actor, Centralized Critic AC with Centralized, 集中训练，分开执行。

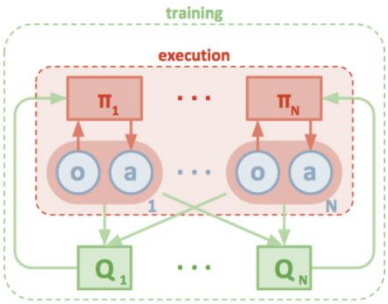


Figure 4: Overview of Multi-Agent Decentralized Actor, Centralized Critic

要使分开执行成为可能，只需要个体 agent 可以观测到自己的 observation。

有一个集中的 critic 给每个 agent 提供完全全局 state 的 observation 观测。这有利于解决内部智能体通讯的受限的问题。

它包括 MDP 的拓展，通常被称为 the Decentralized Partially Observable Markov Decision Processes (**Dec-POMDPs**).

这种方法允许使用策略执行中不可用的信息，具有弱收敛性。

最近使用了一种聪明的方法，通过使用政策梯度的反事实基线来估计优势函数。

为了克服这种复杂性，我们往往有必要采取去中心化策略（decentralized policies），即每个智能体仅根据局部的动作观察历史而选择它们自己的动作。此外，即使在联合动作空间不是特别大的情况，学习期间的局部可观察性和通信约束也可能需要使用去中心化策略。

在合作的环境下，联合动作通常只能产生全局奖励，因此每个智能体都很难判断它自己对团队的贡献。然而，这些奖励在合作的情况下通常是不可获取的，且往往不能鼓励单个智能体为更大的集体利益而做出牺牲。这种现象通常会在充满挑战的任务中大幅度地阻碍所智能体的学习，即使相对较少的智能体也会产生这种显现。

首先，COMA 使用一个中心化的 critic（只用于学习期间），同时执行期间只需要 actor。由于学习是中心化的，因此我们可以使用中心化的 critic 限定联合动作和所有可用的状态信息，同时每个智能体的策略只限定在其自己的动作观察历史上。

第二，COMA 使用反事实基线（counter-factual baseline）表示不使用事实采用的动作作为基线。**这一想法的灵感来自差异奖励**（difference rewards），**其中每个智能体从目标奖励中学习，目标奖励将比较全局奖励与当那个智能体的动作被默认动作替代时接收的奖励。**尽管差异奖励是执行多智能体信度分配（credit assignment）的有效方法，但是它们需要访问模拟器或者评估的奖励函数，并且总体上不清楚如何选择默认动作。COMA 通过使用中心化的 critic 计算一个特定智能体的优势函数而解决了这一问题，该函数对比了**当前联合动作的评估回报**与**边际化单一智能体动作的反事实基线**，同时保持其他智能体的动作不变。这类似于计算一个贵族效用（aristocrat utility），但避免了策略和原来效用函数之间的递归性相互依赖问题，这主要是因为反事实基线（counter-factual baseline）对策略梯度的期望贡献为零。因此，COMA 并不依赖额外的模拟、近似或关于适当默认行动的假设，它会依赖于集中式的 critic 为每一个智能体计算一个分离的基线，以推理特定智能体动作改变的反事实。

第三，COMA 使用一个 critic 表征以高效地计算反事实基线。在单次前向传播中，以所有其它智能体的动作为条件，它会为给定智能体的所有不同动作计算 Q-values。因为单个集中式的 critic 可用于全部的智能体，所有智能体的所有 Q-values 都能在一个批量的单次前向传播中计算。

http://www.sohu.com/a/217441065_465975

$$A^a(s, \mathbf{u}) = Q(s, \mathbf{u}) - \sum_{u'^a} \pi^a(u'^a|\tau^a) Q(s, (\mathbf{u}^{-a}, u'^a))$$

where the advantage estimate is computed for each agent and the baseline marginalizes out the actions (u) of an agent a. This allows the centralized critic to reason about the counter-factual in which only a's actions change.

训练使用 AC 的方法。

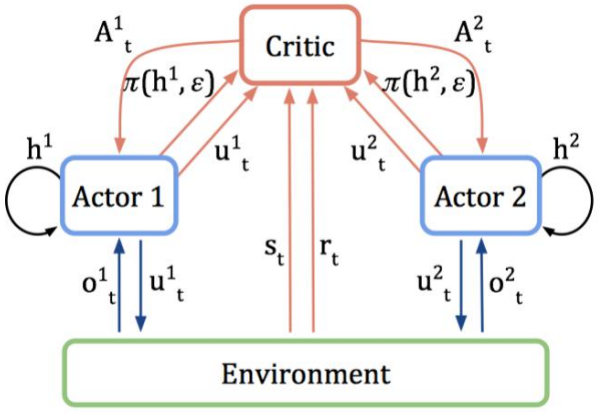


Figure 5: Information Flow in COMA [Foerster et al., 2017]

h 表示门控神经网络中存在的 actor 的隐藏状态，以说明环境的非马尔科夫性质。

另一种方法使用 Q-learning 相同的范式只是提出一个新的目标函数以监督 loss，主要想法就叫 QMIX。使用联合值函数 the joint value function 的线性分解，线性分解的每个部分作为一个 agent 的 value function，同时保持局部最大和全局最大的值函数 value function 的单调性。

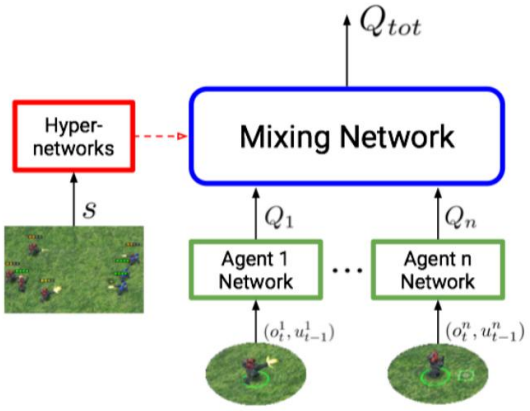


Figure 6: Information Flow in QMIX [Rashid et al., 2018]

Hyper Network 是一个辅助网络。

Mixing Network 通过获取辅助超网络生成的权值的绝对值来加强单调性。

训练使用 DQN 方法

$$\frac{\partial Q_{tot}}{\partial Q_a} \geq 0 \text{ (for all agent's Q-Networks)}$$

$$\mathcal{L}(\theta) = \sum_i [y_i^{tot} - Q_{tot}(\tau, \mathbf{u}, s; \theta)]$$

$$y_i^{tot} = r + \gamma \max_{\mathbf{u}'} Q_{tot}(\tau', \mathbf{u}', s'; \theta^-)$$

**Future Work**

A large set of problems still stay open in both the **theoretical** and **applied aspects** of Reinforcement Learning Systems for Multi-Agent Systems.

还是建议以 Dec-POMDPs 作为理论基础。