Q-learning 算法理论总结

1. Q-learning 的目标

Q-learning 本质是为了得到最优动作价值函数 Q^*，因为最优动作价值函数就是在状态 s 执行动作 a 后，未来一直遵循最优策略，所能得到的最大期望回报，其中就隐含最优策略，即每次选择每个状态动作价值最大的那个。

贝尔曼最优方程：

$$Q^*(s,a) = \mathbb{E}[R_{t+1} + \gamma \max_{a' \in \mathcal{A}} Q^*(S_{t+1}, a')]$$

2. Model-free 的处理方法

Q-learning 作为 model-free 算法，无法直接使用求和形式 $Q^*(s,a) = r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s'|s,a) \max_{a'} Q^*(s',a')$，因为它不知道环境模型——既不知道状态转移概率 $p(s'|s,a)$，也不知道奖励函数 $r(s,a)$，因此无法计算这个求和。但 Q-learning 的目标仍然是找到满足期望形式 $Q^*(s,a) = \mathbb{E}[R_{t+1} + \gamma \max_{a' \in \mathcal{A}} Q^*(S_{t+1}, a')]$ 的最优 Q 函数，它通过与环境交互获得采样 (s, a, R_{t+1}, s_{t+1})，用单个样本 $R_{t+1} + \gamma \max_{a' \in \mathcal{A}} Q(s_{t+1}, a')$ 近似期望进行增量更新，通过大量采样和迭代，利用大数定律使 Q 值逐渐收敛到满足贝尔曼最优方程的 Q^*。

3. Off-policy 双策略架构

Q-learning 属于 off-policy 算法，即自始至终使用固定的双策略架构：

3.1 Behavior Policy（行为策略）

- **定义**：ε-greedy 策略

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

  其中 $\varepsilon \in (0,1)$ 是探索率，$|\mathcal{A}|$ 是动作空间大小

- **策略解释**：

- 以概率 $(1-\varepsilon)$ 选择当前最优动作（利用）
- 以概率 ε 随机选择任意动作（探索）
- 确保所有动作都有被选择的可能性

- **作用**：具有探索性，平衡探索与利用，与环境交互，采集经验样本

3.2 Target Policy（目标策略）

- **定义**：Greedy 策略（贪婪策略）

$$\pi^*(a|s) = \begin{cases} 1, & a = \text{arg max}_a Q(s,a) \\ 0, & a \neq \text{arg max}_a Q(s,a) \end{cases}$$

- **策略解释**：
  - 确定性策略，总是选择 Q 值最大的动作
  - 不进行探索，纯粹利用当前知识
  - 体现了对最优动作的 “贪婪” 选择

- **作用**：在 Q-learning 更新公式中，用于计算 $\text{max}_{a' \in \mathcal{A}} Q(s_{t+1}, a')$
  - 即在下一状态 s_{t+1} 时，选择使 Q 值最大的动作
  - 这个最大值用于构建 TD target：$R_{t+1} + \gamma \text{ max}_{a' \in \mathcal{A}} Q(s_{t+1}, a')$
  - 体现了 “假设未来按最优策略行动” 的思想

3.3 策略演化机制

策略的定义方式从算法开始到结束保持恒定：

- Behavior policy 始终是 ε-greedy
- Target policy 始终是 greedy

虽然策略定义固定，但 Q 表的持续更新导致策略行为的演化：

- **初始阶段**：Q 值随机或为零，策略行为接近随机
- **学习过程**：Q 值逐步逼近 Q^*，策略行为逐渐优化
- **收敛阶段**：Q 值接近最优，策略行为趋向最优

Q-learning 算法通过间接机制实现策略优化：

- 不直接调整策略参数
- 仅通过更新 Q 值改变策略行为
- 简单的值迭代实现复杂的策略改进

4. 增量更新机制

采样 $(s_t, a_t, R_{t+1}, s_{t+1})$ 后，对该状态-动作对采用增量更新的方式：$Q_{\text{new}}(s_t, a_t) = Q_{\text{old}}(s_t, a_t) + \alpha(\text{target} - Q_{\text{old}}(s_t, a_t))$，其中 α 为学习率（标准 Q-learning 中 $0 < \alpha \leq 1$）

学习率 α 的影响分析

对于被更新的状态-动作对 (s, a)：

1. $\alpha = 1$：完全替换，直接跳到目标：$Q_{\text{new}}(s,a) = (\mathcal{T} Q_{\text{old}})(s,a)$
2. $\alpha = 0$：完全不动：$Q_{\text{new}}(s,a) = Q_{\text{old}}(s,a)$
3. $0 < \alpha < 1$：部分朝 $(\mathcal{T} Q_{\text{old}})(s,a)$ 移动，保守更新
4. $\alpha > 1$：过度更新，会跨越目标值，可能导致震荡，算法不稳定
5. $\alpha < 0$：反向更新，朝着与目标相反的方向更新，算法发散，完全无法学习

具体的更新公式为：

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[R_{t+1} + \gamma \max_{a' \in \mathcal{A}} Q(s_{t+1}, a') - Q(s_t, a_t)]$$

5. 收敛性证明

本节证明在固定学习率 α（$0 < \alpha \leq 1$）下的期望收敛性。

收敛性的前提条件

Q-learning 收敛到最优 Q^* 需要满足以下条件：

1. 有限的状态和动作空间
2. 所有状态-动作对被充分访问
3. 折扣因子 $0 \leq \gamma < 1$

接下来证明 Q-learning 更新规则在期望意义下的收敛性：通过迭代应用基于采样的更新公式，动作价值函数 Q 将收敛到最优动作价值函数 Q^*，即对所有 $(s,a) \in \mathcal{S} \times \mathcal{A}$，有 $Q(s,a) \to Q^*(s,a)$。

5.1 证明需要用到的数学工具

为了证明 Q-learning 的收敛性，我们需要以下数学概念：

1. 贝尔曼最优算子 $\mathcal{T}$

- **定义**：算子（函数空间上的映射），将 Q 函数映射为新的 Q 函数

$$(\mathcal{T} Q)(s,a) := \mathbb{E}[R_{t+1} + \gamma \max_{a' \in \mathcal{A}} Q(S_{t+1}, a')]$$

  其中 $\mathcal{T} : (\mathcal{S} \times \mathcal{A} \to \mathbb{R}) \to (\mathcal{S} \times \mathcal{A} \to \mathbb{R})$，输入 Q 函数，输出新 Q 函数
- **作用**：描述 Q 值的理想更新方向，如果能计算完整期望，Q 值应朝此方向更新

2. 不动点性质

- **定义**：Q^* 是贝尔曼最优算子 $\mathcal{T}$ 的不动点

$$\mathcal{T} Q^* = Q^*$$

- **作用**：刻画了最优动作价值函数 Q^* 的特征，是我们的收敛目标

3. 压缩映射性质

- **定义**：算子 $\mathcal{T}$ 满足 γ-压缩性

$$\|\mathcal{T}Q_1 - \mathcal{T}Q_2\|_\infty \le \gamma\|Q_1 - Q_2\|_\infty$$

其中 $0 < \gamma < 1$ 是折扣因子

- **作用**：保证迭代收敛，每次应用算子会缩小 Q 函数间的距离

4. 无穷范数 $\|\cdot\|_\infty$

- **定义**：Q 函数的最大值范数

$$\|Q\|_\infty = \max_{s,a} |Q(s,a)|$$

- **作用**：度量 Q 函数间的最大偏差，确保所有 (s,a) 对都收敛（最坏情况保证）

5. Banach 不动点定理

- **定义**：完备度量空间中的压缩映射存在唯一不动点
- **作用**：从理论上保证 Q^* 的存在性和唯一性，确保 Q-learning 有唯一收敛目标

5.2 期望意义下更新方向分析

Q-learning 更新公式：

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha[R_{t+1} + \gamma \max_{a' \in \mathcal{A}} Q(s_{t+1},a') - Q(s_t,a_t)]$$

每次更新使用单个采样 $(s_t, a_t, R_{t+1}, s_{t+1})$，更新后的 Q 值 $Q_{\text{new}}(s_t,a_t)$ 是随机变量，因为虽然更新的是同一个 Q 表项 $Q(s_0,a_0)$，但由于下一状态 s_{t+1} 的随机性，每次采样更新可能得到不同的新值。这正是 $Q_{\text{new}}(s_t,a_t)$ 具有随机性的原因。

具体数值例子：假设 $Q_{\text{old}}(s_0,a_0) = 5.0$，$\alpha = 0.1$，$\gamma = 0.9$

- 如果转移到 s_1（Q 表中 s_1 的最大 Q 值是 8）：$Q_{\text{new}}(s_0,a_0) = 5.0 + 0.1[2 + 0.9 \times 8 - 5.0] = 5.42$
- 如果转移到 s_2（Q 表中 s_2 的最大 Q 值是 3）：$Q_{\text{new}}(s_0,a_0) = 5.0 + 0.1[2 + 0.9 \times 3 - 5.0] = 4.97$
- 如果转移到 s_3（Q 表中 s_3 的最大 Q 值是 10）：$Q_{\text{new}}(s_0,a_0) = 5.0 + 0.1[2 + 0.9 \times 10 - 5.0] = 5.60$

但通过大量采样更新，这些随机更新的累积效应会使 Q 值逼近其期望值，最终收敛到最优 Q^*。

我们分析单次采样更新在期望意义下的方向：

$$\begin{aligned}\mathbb{E}[Q_{\text{new}}(s,a)] &= Q_{\text{old}}(s,a) + \alpha[\mathbb{E}[R_{t+1} + \gamma \max_{a' \in \mathcal{A}} Q(S_{t+1}, a')] - Q_{\text{old}}(s,a)] \\ &= Q_{\text{old}}(s,a) + \alpha[(\mathcal{T}Q_{\text{old}})(s,a) - Q_{\text{old}}(s,a)] \\ &= (1-\alpha)Q_{\text{old}}(s,a) + \alpha(\mathcal{T}Q_{\text{old}})(s,a)\end{aligned}$$

解释：对于被更新的状态-动作对 (s,a)，单次采样更新后，在期望意义下，$Q_{\text{new}}(s,a)$ 这个数值会从 $Q_{\text{old}}(s,a)$ 朝 $(\mathcal{T}Q_{\text{old}})(s,a)$ 方向移动了 α 倍的距离。而 Q^* 是贝尔曼最优算子 $\mathcal{T}$ 的不动点，这说明更新方向是正确的。

5.3 误差收缩证明：证明收敛速率

基于 5.2 的点态分析，我们现在将结果推广到整个函数空间。对于任意状态-动作对 (s,a)，上述期望等式都成立，因此可以写成函数形式（这里 Q_{new}、Q_{old}、Q^* 都表示定义在 $\mathcal{S} \times \mathcal{A}$ 上的函数）：

$$\begin{aligned}\text{步骤 1（更新公式的函数形式）：} \quad & Q_{\text{new}} = (1-\alpha)Q_{\text{old}} + \alpha(\mathcal{T}Q_{\text{old}}) \\ \text{步骤 2（两边减去}Q^*\text{）：} \quad & Q_{\text{new}} - Q^* = (1-\alpha)Q_{\text{old}} + \alpha(\mathcal{T}Q_{\text{old}}) - Q^* \\ \text{步骤 3（}Q^*\text{拆分代入）：} \quad & Q_{\text{new}} - Q^* = (1-\alpha)Q_{\text{old}} + \alpha(\mathcal{T}Q_{\text{old}}) - (1-\alpha)Q^* - \alpha Q \\ \text{步骤 4（重新组合）：} \quad & Q_{\text{new}} - Q^* = (1-\alpha)[Q_{\text{old}} - Q^*] + \alpha[(\mathcal{T}Q_{\text{old}}) - Q^*] \\ \text{步骤 5（定义}e = Q - Q^*\text{）：} \quad & e_{\text{new}} = (1-\alpha)e_{\text{old}} + \alpha(\mathcal{T}Q_{\text{old}} - Q^*) \\ \text{步骤 6（利用}\mathcal{T}Q^* = Q^*\text{）：} \quad & e_{\text{new}} = (1-\alpha)e_{\text{old}} + \alpha(\mathcal{T}Q_{\text{old}} - \mathcal{T}Q^*)\end{aligned}$$

步骤 7：应用压缩性质

$$\|\mathcal{T}Q_{\text{old}} - \mathcal{T}Q^*\|_\infty \le \gamma\|Q_{\text{old}} - Q^*\|_\infty = \gamma\|e_{\text{old}}\|_\infty$$

所以有：

$$\begin{aligned}\|e_{\text{new}}\|_\infty &= \|(1-\alpha)e_{\text{old}} + \alpha(\mathcal{T}Q_{\text{old}} - \mathcal{T}Q^*)\|_\infty \\ &\le \|(1-\alpha)e_{\text{old}}\|_\infty + \|\alpha(\mathcal{T}Q_{\text{old}} - \mathcal{T}Q^*)\|_\infty \quad \text{（三角不等式）} \\ &= (1-\alpha)\|e_{\text{old}}\|_\infty + \alpha\|\mathcal{T}Q_{\text{old}} - \mathcal{T}Q^*\|_\infty \quad \text{（标量提取）} \\ &\le (1-\alpha)\|e_{\text{old}}\|_\infty + \alpha\gamma\|e_{\text{old}}\|_\infty \quad \text{（压缩性质）} \\ &= [1 - \alpha(1-\gamma)]\|e_{\text{old}}\|_\infty\end{aligned}$$

5.4 收敛性结论

- **收缩因子（Contraction Factor）**：$\rho = 1 - \alpha(1-\gamma)$

- 由于 $0 < \alpha \leq 1$ 且 $0 < \gamma < 1$，故 $\rho < 1$
- 第 t 步迭代后：$\|e_t\|_\infty \leq \rho^t \|e_0\|_\infty$，这保证了误差的几何衰减
  - 其中 $\|e_t\|_\infty = \|Q_t - Q^*\|_\infty = \max_{s,a} |Q_t(s,a) - Q^*(s,a)|$
  - 表示第 t 步时 Q 函数与最优 Q^* 的最大偏差

6. 算法实施

证明了收敛性，接下来不断将采样数据代入，在线更新，即每采样一步就立即更新一次对应的 $Q(s,a)$ 值，通过大量采样覆盖各个状态的各个动作，迭代直到满足预设的训练终止条件。

常见的训练终止条件

1. **固定 Episode 数**：达到预设的训练轮数
2. **收敛判定**：连续 N 个 episode 的平均奖励变化小于阈值
3. **Q 值稳定**：在一个完整 episode 或固定步数后，比较整个 Q 函数的变化：$\|Q_{\text{new}} - Q_{\text{old}}\|_\infty <$ threshold（其中 Q_{new}、Q_{old} 表示整个 Q 函数），即所有状态-动作对的最大变化幅度小于阈值

7. 算法特点总结

- **优势**：简单易实现、保证收敛到最优、无需环境模型
- **局限**：表格型限制于有限状态空间、样本效率较低
- **关键参数**：学习率 α 和探索率 ε 需要仔细调整
- **实践提示**：固定小学习率通常比衰减学习率更稳定