Chapter 2 Multi-armed Bandits

这部分内容进入了强化学习最简单的一种特例——**多臂老虎机问题 (Multi-armed Bandits)**。

如果在上一节我们是在讨论“怎么走迷宫（序列决策）”，那么这一节我们把问题简化为“怎么选餐馆（单步决策）”。在这个简化设定下，我们不需要考虑状态转移（Environment is fixed），只需要专注于**评估动作的价值**。

多臂老虎机 (Multi-armed Bandits)

1. 直觉与生活例子 (Intuition)

核心场景：赌场里的老虎机

- **名字由来**：老式的老虎机有一个拉杆，被称为“单臂强盗（One-armed Bandit）”。如果你面前有一排这种机器（比如 k 台），这就是“多臂老虎机”。
- **游戏规则**：
  - 每一台机器吐钱的概率和金额都不一样（有的容易赢，有的只会吞钱）。
  - 但你不知道哪台是最好的。
  - 你有 N 次拉杆的机会。
- **你的目标**：在有限的次数里，赢走最多的钱。

核心难题：试一试还是接着玩？

- 假设你试了第1台，赢了10块；试了第2台，输了5块。
- **Exploitation (利用/贪婪)**：你就一直玩第1台。但万一第3台其实每次都能赢100块呢？你错过了赚大钱的机会。
- **Exploration (探索)**：你去试第3台。但如果第3台是个坑，你这就浪费了一次机会。
- 这就是我们在上一节提到的**探索-利用困境**的最简化数学模型。

---

2. 数学定义 (Mathematical Formalism)

我们把上面的赌场直觉翻译成数学语言。

2.1 设定

- **k-armed Bandit Problem**: 我们有 k 个选项（动作）。
- **Action (A_t)**: 在时刻 t，你选择的某个动作（拉动某根拉杆）。
- **Reward (R_t)**: 这是一个**随机变量**。哪怕拉同一根杆，每次给的钱也可能不一样，但它服从某个固定的概率分布。

2.2 两个关键的“价值”

这是理解本节最重要的一对概念：**“上帝视角”** vs **“凡人视角”**。

1. **真实价值 (True Value, $Q^*(a)$)** —— 上帝视角
   - 这是动作 a **真正**平均能给你多少回报。
   - **公式**: $Q^*(a) \triangleq \mathbb{E}[R_t|A_t = a]$
   - *解释*: $\mathbb{E}$ 代表期望值。如果你拉无限次杆，平均每次得到的钱就是这个数。在现实中，**这个值是未知的**。
2. **估计价值 (Estimated Value, $Q_t(a)$)** —— 凡人视角
   - 这是你**以为**动作 a 能给你多少回报。
   - 你的目标是让估计值 $Q_t(a)$ 尽可能接近真实值 $Q^*(a)$。

▼ **详细解释**

这个公式是强化学习（特别是多臂老虎机问题）中定义**动作价值 (Action Value)** 的基石。它定义了当我们谈论“某个动作好不好”时，数学上到底是在指什么。

$$Q^*(a) \triangleq \mathbb{E}[R_t|A_t = a]$$

我将其拆解为**直观翻译**、**符号详解**和**核心逻辑**三部分。

---

1. 直观翻译

用一句话翻译这个公式：

> “动作 a 的真实价值 (Q^*)，定义为：当我们选择了动作 a 之后，预期能得到的平均奖励。”

---

2. 符号逐一拆解

1. $Q^*(a)$ —— **真实价值 (True Value)**
    - Q: 代表 **Quality**（质量）或 **Quantity**（量化值）。在RL中通用表示“价值”。
    - a: 代表某个具体的**动作 (Action)**。比如“拉动第3号拉杆”或“选择去吃海底捞”。
    - $*$ **(星号)**: 在数学里通常代表**“最优”**或**“真实/理想”**的状态。
        - 这里它强调这是**上帝视角**的数值。在现实中，我们通常不知道它，我们的目标就是去**猜测/估计**这个值。
2. $\triangleq$ —— **定义为 (Defined as)**
    - 这不是推导出来的“等于”，而是我们人为规定的**定义**。
3. $\mathbb{E}[...]$ —— **期望值 (Expectation)**
    - 这是概率论术语。你可以把它简单理解为**“无限次尝试后的理论平均值”**。
    - **为什么用期望不用具体的数？**
        - 因为老虎机是**随机**的。
        - 比如动作 a 是“抽盲盒”。你可能抽到隐藏款（奖励100），也可能抽到普通款（奖励10）。你无法说动作 a 的回报是100还是10。
        - 但如果你抽一亿次，平均下来每次大概值 35 元。这个 **35** 就是 $\mathbb{E}$。
4. R_t —— **奖励 (Reward at time t)**
    - 这是在时刻 t 获得的实际奖励数值。它是一个**随机变量**（每次可能不一样）。
5. $|$ —— **条件 (Given that)**
    - 读作“在……条件下”。它限定了范围。我们不是算所有时候的平均分，我们只算**发生了后面那件事**时候的平均分。
6. $A_t = a$ —— **动作 (Action at time t is a)**
    - A_t 是时刻 t 的动作（随机变量）。
    - a 是我们指定的那个特定动作（比如“左边的拉杆”）。

- 整句的意思是：**“假设我们在时刻 t 确实选了动作 a”**。

---

3. 举个“餐厅打分”的例子

假设你面前有一家餐厅（动作 a），你想知道它到底好不好吃（$Q^*(a)$）。

- **现状**：这家餐厅厨师发挥不稳定（随机性）。
  - 周一去吃，巨好吃，打分 10 分（$R_1 = 10$）。
  - 周二去吃，咸了，打分 4 分（$R_2 = 4$）。
  - 周三去吃，一般，打分 7 分（$R_3 = 7$）。
- **公式含义**：
  - $A_t = a$：指**“只要我是去这家餐厅吃饭”**（排除了去别的餐厅的情况）。
  - $\mathbb{E}[R_t | \dots]$：指这家餐厅**真实的平均水平**。
  - 如果你去了无数次，你会发现平均分稳定在 **7.0分**。
  - 那么：$Q^*(a) = 7.0$。

---

4. 关键点：为什么这个公式很重要？

这个公式揭示了强化学习的**根本矛盾**：

1. $Q^*(a)$ **是“神”知道的数**：它是客观存在的真理（这家餐厅的真实水平）。
2. $Q_t(a)$ **是“人”猜的数**：这是我们在前几节提到的 Q_t（估计值）。
   - 刚开始，你只去吃了一次（周二去吃的），得了4分。
   - 你心里的估计值 $Q_t(a) = 4$。
   - 但这不等于真实值 $Q^*(a) = 7$。

强化学习的全过程，就是通过不断的尝试（收集 R_t），让你的猜测 $Q_t(a)$ 无限逼近真实值 $Q^*(a)$ 的过程。

---

3. 怎么算？(Algorithms)

既然我们不知道真实价值 $Q^*(a)$，我们怎么做决策？这就需要**动作价值方法 (Action-value Methods)**。

3.1 样本平均法 (Sample-Average Method)

这是最符合直觉的估算方法：**看历史平均分**。

- **直觉**：如果第1台机器我玩了3次，分别赢了5、10、0块。那我就估计它的价值是 $(5+10+0)/3=5$。
- **公式**:

$$Q_t(a) = \frac{\text{sum of rewards when } a \text{ taken prior to } t}{\text{number of times } a \text{ was taken prior to } t} = \frac{\sum_{i=1}^{t-1} r_i \cdot \mathbb{1}_{a_i=a}}{\sum_{i=1}^{t-1} \mathbb{1}_{a_i=a}}.$$

- *解释*: 分子是你选动作 a 获得的总奖励，分母是你选动作 a 的次数。
- **大数定律**: 当你试的次数趋向于无穷大，这个平均值就会等于真实价值 $Q^*(a)$。

指示函数，判断是否选了动作 a

$$\underline{\mathbb{1}_{a_i=a}}$$

3.2 怎么选动作？(Action Selection)

算出了分值 $Q_t(a)$，下一步就是选哪个。两种策略：

A. 贪婪策略 (Greedy)

- **规则**：只选分最高的那个。如果平分，随便挑一个。
- **公式**: $A_t = \underset{a \in \mathcal{A}}{\operatorname{argmax}} Q_t(a)$
- **缺点**：这是纯粹的**利用 (Exploitation)**。如果你运气不好，第一次玩最好的机器时输了（虽然它平均很高，但偶尔会输），你就再也不会选它了，从而陷入**次优解 (Sub-optimal)**。

B. ϵ-贪婪策略 (ϵ-Greedy)

- **规则**：抛硬币决定。
  - **绝大多数时候** (概率 $1-\epsilon$)：贪婪，选分最高的。

    - **偶尔** (概率 ϵ)：**闭眼瞎选**，在所有动作里随机挑一个。
- **优点**：ϵ (Epsilon) 强制你去**探索 (Explore)** 那些还没被看好的动作。随着时间推移，你能发现所有动作的真实好坏，避免错过“沧海遗珠”。

▼ **解释**

1. 公式的符号意思

公式：

$$A_t = \underset{a \in \mathcal{A}}{\operatorname{argmax}} Q_t(a)$$

这个公式在数学上读作：“在时间 t，选择那个能让 $Q_t(a)$ 最大的动作 a”。

- A_t：**Action at time** t。表示你在第 t 次尝试时，最终决定采取的那个**动作**（比如选哪台老虎机）。
- argmax：**Argument of the Maximum**。这是一个数学算子，它的意思是：我不关心最大的数值是多少，我关心的是**哪个变量（a）导致了这个最大值**。
    - *例子*：如果有两个动作，Q(吃饭)=10，Q(睡觉)=100。
    - $\max(Q)$ = 100（最大值是100）。
    - $\operatorname{argmax}(Q)$ = 睡觉（是“睡觉”这个动作产生了最大值）。
- $a \in \mathcal{A}$：表示动作 a 属于整个动作集合 $\mathcal{A}$。意思是把所有可能的动作都拿出来比一比。
- $Q_t(a)$：**Quality / Value**。表示在第 t 时刻，你心目中对动作 a 的**价值估计**（通常就是你提到的平均收益）。

---

2. 对贪婪（Greedy）逻辑的确认

> 你的提问：是选Q最大的嘛？也就是说对每一个动作都试几次，得到每次的R，然后算出平均的R也就是Q，然后选最大的Q？

是的，核心逻辑就是“只选目前分数（Q）最高的”。

但是有一点细微的差别：**纯粹的贪婪策略并没有强制规定“先对每个动作试几次”**。

- 它非常短视。如果一开始大家的Q都是0，它可能随机选一个；

- 一旦选了一个，并且运气好赢了一点钱（Q变正了），它可能就会**一直选这个**，而完全不去尝试那些还没试过的动作（因为还没试过，Q可能是0，比不过正数）。

Q的计算方式通常确实如你所说：

$Q_t(a) = \frac{\text{该动作历史得到的总奖励}}{\text{该动作被选择的次数}}$

也就是**平均收益（Sample Average）**。

---

3. 为什么“第一次输了”会导致错过？（你的例子分析）

> 你的提问：例子说的是第一次输了，导致暂时的Q小，但是从分布来说，可能理想值是最大的，所以不选错过了是吗？

完全正确！ 这就是贪婪策略最大的缺陷——**“一叶障目”**。

我们可以用一个具体的数值例子来推演你描述的过程：

场景设定：

- **老虎机 A（好机器）**：平均能赢 10 块钱，但波动大（有时候会输）。
- **老虎机 B（烂机器）**：平均只能赢 3 块钱，但很稳定。

过程推演：

1. **初始状态**：你对 A 和 B 的估计都是 0。即 $Q(A)=0, Q(B)=0$。
2. **第一步（尝试 A）**：你决定先试一下 A。
   - **运气不好**：虽然 A 平均好，但这次你倒霉，输了 2 块钱（$R=-2$）。
   - **更新估计**：此时你心目中，$Q(A) = -2$。
3. **第二步（做决策）**：你是贪婪的，你要选 Q 最大的。
   - $Q(A) = -2$
   - $Q(B) = 0$ （还没玩过，保持初始值）
   - **决策**：因为 $0 > -2$，所以你选择了 **B**。
4. **第三步（尝试 B）**：你玩了 B，赢了 3 块钱（$R = 3$）。
   - **更新估计**：$Q(B) = 3$。
5. **未来的死循环**：

- 现在 $Q(A) = -2$。
- 现在 $Q(B) = 3$。
- 因为 $3 > -2$，下一把你还是会选 B... 再下一把还是 B...
- B 的分数会稳定在 3 左右，始终大于 A 的 -2。

结局：

虽然 A 其实是平均能赢 10 块的“神机”，但因为你**第一次尝试时的坏运气（Bad Luck）加上你的贪婪（不再给它翻身的机会）**，你永远错过了它，陷在了 B 这个“次优解（Sub-optimal）”里。

这就是 PPT 里说的 **“陷入次优解”** 的含义。

这一节的核心是**“即使没有复杂的状态转移，光是平衡探索和利用就已经很难了”**。

- **Sample-average** 解决了“怎么评价过去”的问题。
- **ϵ-greedy** 解决了“怎么面对未来”的问题（在稳妥拿钱和寻找潜力股之间平衡）。

增量式更新与非平稳问题

1. 直觉与生活例子 (Intuition)

1.1 增量式更新 (Incremental Implementation)

场景：计算班级平均身高

想象你是班长，要计算全班的平均身高。

$$\tilde{Q}_n \triangleq \frac{\tilde{R}_1 + \cdots + \tilde{R}_{n-1}}{n-1},$$

- **笨办法（Sample Average）：**
  - 第1个人进来，记录 `170`。平均=170。
  - 第2个人进来，记录 `170, 180`。你把两个数加起来除以2。平均=175。

    - ...
    - 第100个人进来，你要把**前99个人的身高重新加一遍**，再加上第100个人的，然后除以100。
    - **问题**：这太浪费脑子（内存）和时间（计算量）了！你需要记下所有人的历史数据。
- **聪明办法（Incremental Update）**：
    - 你只记一个数：**当前的平均值**（比如175）。
    - 第100个人（身高180）进来。
    - 你心里算：他比平均值高5厘米（`Error = 180 - 175 = 5`）。
    - 这5厘米的"多余部分"平摊到100个人头上，每个人分 `5/100 = 0.05`。
    - 新的平均值 = 旧平均值 + 0.05 = 175.05。
    - **优势**：你不需要记得前99个人的具体身高，只需要记得现在的平均值和人数。

1.2 非平稳问题 (Non-stationary Problem)

场景：评价一家餐厅

- **平稳环境 (Stationary)**：餐厅厨师没变，水平一直稳定。你用过去10年的平均分来评价它是准确的。
- **非平稳环境 (Non-stationary)**：餐厅最近换了大厨，或者甚至换了老板。
    - 如果你还在算"过去10年的平均分"，那么5年前的难吃经历会拖累现在的评分。
    - **解决**：你应该更看重**最近**的体验，哪怕5年前很难吃，只要最近3次好吃，评分就该很高。这就是"遗忘旧历史，重视新数据"。

---

2. 数学推导：从笨办法到聪明公式

我们来一步步推导图片中那个极其优美的**增量更新公式**。

$$
\begin{aligned}
\tilde{Q}_{n+1} &= \alpha\tilde{R}_n + (1-\alpha)\tilde{Q}_n \\
&= \alpha\tilde{R}_n + (1-\alpha)(\alpha\tilde{R}_{n-1} + (1-\alpha)\tilde{Q}_{n-1}) \\
&= \alpha\tilde{R}_n + (1-\alpha)\alpha\tilde{R}_{n-1} + (1-\alpha)^2\tilde{Q}_{n-1} \\
&= \sum_{i=1}^{n} \alpha(1-\alpha)^{n-i}\tilde{R}_i + (1-\alpha)^n\tilde{Q}_1.
\end{aligned}
$$

2.1 符号设定

- Q_n: 动作被选了 $n-1$ 次后的估计价值（即旧的平均值）。
- R_n: 第 n 次获得的奖励。
- Q_{n+1}: 把 R_n 算进去后的新平均值。

2.2 推导步骤 (Derivation)

根据定义，新的平均值是所有 n 个奖励的总和除以 n：

$Q_{n+1} = \frac{1}{n}\sum_{i=1}^{n} R_i$

我们可以把第 n 个奖励 R_n 单独提出来：

$= \frac{1}{n}\left(R_n + \sum_{i=1}^{n-1} R_i\right)$

注意，旧的平均值 Q_n 等于前 $n-1$ 个奖励的和除以 $n-1$，所以 $\sum_{i=1}^{n-1} R_i = (n-1)Q_n$。代入上式：

$= \frac{1}{n}\left(R_n + (n-1)Q_n\right)$

展开括号：

$= \frac{1}{n}\left(R_n + nQ_n - Q_n\right)$

把 nQ_n 除以 n 提出来：

$= Q_n + \frac{1}{n}\left(R_n - Q_n\right)$

最终公式 (The Golden Rule of RL Update):

$Q_{n+1} = Q_n + \frac{1}{n}(R_n - Q_n)$

这个公式极其重要，它揭示了强化学习更新的一般形式：

$NewEstimate \leftarrow OldEstimate + StepSize \times (Target - OldEstimate)$

- **Target (R_n)**: 我们刚得到的真实奖励。
- **OldEstimate (Q_n)**: 我们原本的预测。
- **Target - OldEstimate**: **预测误差 (Error)**。我看走眼了多少？
- **StepSize ($1/n$)**: 步长。我们根据误差修正预测的幅度。

---

3. 进阶：处理变化的世界 (Non-stationary)

如果环境变化（非平稳），我们不能用 $1/n$ 作为步长。

3.1 为什么 $1/n$ 不行？

随着 n 变大（比如玩了10000次），步长 $1/n$ 会变得无限小。

这意味着，**到了后期，无论新的奖励 R_n 多么离谱，智能体几乎都不再更新它的估计值了**。它"固执"地守着旧观念。

3.2 解决方案：固定步长 α

我们用一个常数 $\alpha \in (0, 1]$ 来代替 $1/n$。

$$Q_{n+1} = Q_n + \alpha(R_n - Q_n)$$

3.3 指数加权平均 (Exponential Recency-Weighted Average)

这会导致什么结果？让我们展开递归公式看看：

$$Q_{n+1} = \alpha R_n + (1-\alpha)Q_n$$
$$= \alpha R_n + (1-\alpha)[\alpha R_{n-1} + (1-\alpha)Q_{n-1}]$$
$$= \alpha R_n + \alpha(1-\alpha)R_{n-1} + \alpha(1-\alpha)^2 R_{n-2} + \dots$$

结论：

- R_n (最近的奖励) 权重是 α。
- R_{n-1} (上一次的奖励) 权重是 $\alpha(1-\alpha)$。
- R_i (很久以前的奖励) 权重是 $\alpha(1-\alpha)^{n-i}$。

  因为 $(1-\alpha) < 1$，所以时间越久远，权重呈**指数级衰减**。这就像人类的记忆，越久远的事情印象越模糊，对当前决策的影响越小。

总结与下一步

这一节解决了算法“怎么算才快”和“怎么适应变化”的问题。

到目前为止，我们讨论的都是**Bandit（老虎机）问题，它的特点是没有状态 (Context)** —— 也就是不管你拉哪个杆，机器的状态不会变。

- *接下来的高阶内容（Contextual Bandits 或 Full RL）**会引入：“如果我拉了杆，机器变红了，下次中奖概率变了怎么办？” 这将带我们进入 **马尔可夫决策过程 (MDP)** 的世界。

这部分内容介绍了一种完全不同的解决思路：**梯度老虎机算法 (Gradient Bandit Algorithms)**。

在此之前的方法（如 ϵ-Greedy）都是基于**“价值”（**Value, 即 $Q(a)$）来决策的——我们估计每个动作能得多少分，然后选分高的。**
而这一节的方法是基于“偏好”**（Preference, 即 $H(a)$）——我不关心具体得多少分，我只关心我有多想选它。

梯度老虎机算法 (Gradient Bandit Algorithms)

1. 直觉与核心思想 (Intuition)

1.1 从“估值”到“偏好”

- **以前的做法 (Value-based)**：
  - 你试图算出：“吃那家餐厅平均能得 **4.5分**，吃这家能得 **3.0分**”。
  - 决策：因为 $4.5 > 3.0$，所以我选那家。
- **现在的做法 (Gradient-based)**：
  - 你维护一个**偏好值** $H(a)$。这不是具体的分数，而是一个相对的“热度”或“权重”。
  - 如果某次吃饭体验特别好（高于平时水平），你就增加这家店的“热度”；如果体验不好，就降低它的“热度”。
  - **概率选择**：热度越高的店，被选中的**概率**越大。

1.2 比较基准 (Baseline)

- **核心逻辑**：好与坏是**相对**的。
- 如果你总是得100分，突然得了个90分，这虽然是高分，但相对于你的“平均水平”是**差**的，应该降低偏好。
- 如果你总是得10分，突然得了个20分，这是**好**的，应该增加偏好。
- 这个“平均水平”就是**Baseline ($\bar{r}_t$)**。

---

2. 数学定义 (Mathematical Formulation)

2.1 动作偏好与Softmax分布

- $H_t(a)$: 在时刻 t 对动作 a 的**偏好 (Preference)**。
  - *注意*：这个数值没有具体的物理意义（不是奖励的大小），只有相对大小有意义。
- $\pi_t(a)$: 选择动作 a 的**概率**。我们使用 **Softmax 函数** 将偏好转化为概率：

  $\pi_t(a) \triangleq \frac{e^{H_t(a)}}{\sum_{b \in \mathcal{A}} e^{H_t(b)}}$
  - 所有概率之和为 1。
  - $H_t(a)$ 越大，$\pi_t(a)$ 越接近 1。

2.2 更新公式 (The Algorithm)

在每一步 t，选中了动作 A_t，获得了奖励 R_t，无论是选中还是未选中的动作，都要更新偏好：

$$H_{t+1}(a) = H_t(a) + \alpha(R_t - \bar{R}_t)(1_{a=A_t} - \pi_t(a))$$

- α: 学习率 (Step-size)。
- $R_t - \bar{R}_t$: **优势 (Advantage)**。当前的奖励比平均基准好多少？
- $1_{a=A_t}$: 指示函数。如果是当前被选中的动作，值为1；否则为0。
- $\pi_t(a)$: 当前该动作的被选概率。

直观解释这个公式：

1. **对于被选中的动作 ($a = A_t$)**：
    - 更新量 $\propto (R_t - \bar{R}_t)(1 - \pi_t(A_t))$。
    - 如果 $R_t > \bar{R}_t$ (表现好)，偏好 $H(A_t)$ **增加**。
    - 如果 $R_t < \bar{R}_t$ (表现差)，偏好 $H(A_t)$ **减小**。
2. **对于没被选中的动作 ($a \neq A_t$)**：
    - 更新量 $\propto (R_t - \bar{R}_t)(0 - \pi_t(a))$。
    - 既然概率和为1，如果被选中动作的偏好增加了，其他动作的偏好就必须**相对减小**（反向更新），以维持平衡。

---

3. 关键推导：为什么要这么更新？(Derivation)

这部分展示了该算法其实是在做**随机梯度上升 (Stochastic Gradient Ascent)**，目的是最大化**总预期奖励**。这是一个非常漂亮的理论证明。

3.1 目标函数

我们的目标是最大化预期奖励 $\mathbb{E}[R_t]$：

$$\text{Maximize } \mathbb{E}[R_t] = \sum_{x \in \mathcal{A}} \pi_t(x) Q^*(x)$$

其中 $Q^*(x)$ 是动作 x 的真实价值。

3.2 梯度上升

我们要调整参数 $H_t(a)$ 让目标函数变大。根据梯度上升法：

$$H_{t+1}(a) = H_t(a) + \alpha \frac{\partial \mathbb{E}[R_t]}{\partial H_t(a)}$$

3.3 推导步骤 (Step-by-Step)

Step 1: 展开梯度

$$\frac{\partial \mathbb{E}[R_t]}{\partial H_t(a)} = \sum_{x \in \mathcal{A}} Q^*(x) \frac{\partial \pi_t(x)}{\partial H_t(a)}$$

Step 2: 引入基准 (Baseline Trick)

这是一个数学技巧。因为 $\sum \pi_t(x) = 1$，所以 $\sum \frac{\partial \pi_t(x)}{\partial H} = 0$。我们可以任意减去一个常数 B_t 而不改变结果：

$= \sum_{x \in \mathcal{A}} (Q^*(x) - B_t) \frac{\partial \pi_t(x)}{\partial H_t(a)}$

Step 3: Log-Likelihood 技巧

我们把导数项乘以 $\frac{\pi_t(x)}{\pi_t(x)}$：

$= \sum_{x \in \mathcal{A}} \pi_t(x) (Q^*(x) - B_t) \frac{1}{\pi_t(x)} \frac{\partial \pi_t(x)}{\partial H_t(a)}$

这变成了**期望**的形式 $\mathbb{E}_\pi[\dots]$。

Step 4: Softmax 的导数性质

对于 Softmax 函数，其导数有特殊性质：

$\frac{\partial \pi_t(x)}{\partial H_t(a)} = \pi_t(x)(1_{x=a} - \pi_t(a))$

代入上式，得到梯度的解析解：

$\frac{\partial \mathbb{E}[R_t]}{\partial H_t(a)} = \sum_{x \in \mathcal{A}} \pi_t(x)(Q^*(x) - B_t)(1_{x=a} - \pi_t(a))$

$= \mathbb{E}\left[(Q^*(A_t) - B_t)(1_{A_t=a} - \pi_t(a))\right]$

Step 5: 随机采样 (Stochastic Approximation)

我们无法计算期望，所以我们用单次采样 R_t 来近似 $Q^*(A_t)$，并设基准 $B_t = \bar{R}_t$。

最终得到更新公式：

$H_{t+1}(a) = H_t(a) + \alpha(R_t - \bar{R}_t)(1_{a=A_t} - \pi_t(a))$

总结

到这里，你已经掌握了 Bandit 问题中最主流的两类方法：

1. **Value-based**: 估算 Q 值，比如 ϵ-Greedy。
2. **Gradient-based**: 优化偏好 H 值，比如 Softmax Gradient。

这为你后续学习更复杂的策略梯度方法（Policy Gradient, 如 PPO, TRPO）打下了坚实的数学基础，因为它们的思想源头就在这里。