

目 录	1
-----	---

目 录

1 模版备用	2
2 Q learning 算法	3
3 Sarsa 算法	4
4 Policy Gradient 算法	5
5 DQN 算法	6
6 SoftQ 算法	7
7 SAC-S 算法	8
8 SAC 算法	9

1 模版备用

算法
1: 测试

2 Q learning 算法

Q-learning 算法¹

- 1: 初始化 Q 表 $Q(s, a)$ 为任意值, 但其中 $Q(s_{terminal}, \cdot) = 0$, 即终止状态对应的 Q 值为 0
 - 2: **for** 回合数 = $1, M$ **do**
 - 3: 重置环境, 获得初始状态 s_1
 - 4: **for** 时步 = $1, t$ **do**
 - 5: 根据 $\varepsilon - greedy$ 策略采样动作 a_t
 - 6: 环境根据 a_t 反馈奖励 r_t 和下一个状态 s_{t+1}
 - 7: 更新策略:
 - 8: $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_t + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)]$
 - 9: 更新状态 $s_{t+1} \leftarrow s_t$
 - 10: **end for**
 - 11: **end for**
-

¹Reinforcement Learning: An Introduction

3 Sarsa 算法

Sarsa 算法¹

- 1: 初始化 Q 表 $Q(s, a)$ 为任意值, 但其中 $Q(s_{terminal}, \cdot) = 0$, 即终止状态对应的 Q 值为 0
 - 2: **for** 回合数 $= 1, M$ **do**
 - 3: 重置环境, 获得初始状态 s_1
 - 4: 根据 $\varepsilon - greedy$ 策略采样初始动作 a_1
 - 5: **for** 时步 $= 1, t$ **do**
 - 6: 环境根据 a_t 反馈奖励 r_t 和下一个状态 s_{t+1}
 - 7: 根据 $\varepsilon - greedy$ 策略 s_{t+1} 和采样动作 a_{t+1}
 - 8: **更新策略:**
 - 9: $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_t + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]$
 - 10: 更新状态 $s_{t+1} \leftarrow s_t$
 - 11: 更新动作 $a_{t+1} \leftarrow a_t$
 - 12: **end for**
 - 13: **end for**
-

¹Reinforcement Learning: An Introduction

4 Policy Gradient 算法

REINFORCE 算法: Monte-Carlo Policy Gradient¹

```
1: 初始化策略参数  $\theta \in \mathbb{R}^{d'}$  ( e.g., to  $\mathbf{0}$  )
2: for 回合数 = 1,  $M$  do
3:   根据策略  $\pi(\cdot | \cdot, \theta)$  采样一个 (或几个) 回合的 transition
4:   for 时步 = 1,  $t$  do
5:     计算回报  $G \leftarrow \sum_{k=t+1}^T \gamma^{k-t-1} R_k$ 
6:     更新策略  $\theta \leftarrow \theta + \alpha \gamma^t G \nabla \ln \pi(A_t | S_t, \theta)$ 
7:   end for
8: end for
```

¹Reinforcement Learning: An Introduction

5 DQN 算法

DQN 算法

- 1: 初始化策略网络参数 θ
 - 2: 复制参数到目标网络 $\hat{Q} \leftarrow Q$
 - 3: 初始化经验回放 D
 - 4: **for** 回合数 = 1, M **do**
 - 5: 重置环境, 获得初始状态 s_t
 - 6: **for** 时步 = 1, t **do**
 - 7: 根据 $\varepsilon - greedy$ 策略采样动作 a_t
 - 8: 环境根据 a_t 反馈奖励 r_t 和下一个状态 s_{t+1}
 - 9: 存储 transition 即 (s_t, a_t, r_t, s_{t+1}) 到经验回放 D 中
 - 10: 更新环境状态 $s_{t+1} \leftarrow s_t$
 - 11: **更新策略:**
 - 12: 从 D 中采样一个 batch 的 transition
 - 13: 计算实际的 Q 值, 即 $y_j = \begin{cases} r_j & \text{对于终止状态 } s_{j+1} \\ r_j + \gamma \max_{a'} Q(s_{j+1}, a'; \theta) & \text{对于非终止状态 } s_{j+1} \end{cases}$
 - 14: 对损失 $(y_j - Q(s_j, a_j; \theta))^2$ 关于参数 θ 做随机梯度下降
 - 15: **end for**
 - 16: 每 C 个回合复制参数 $\hat{Q} \leftarrow Q$ (此处也可像原论文中放到小循环中改成每 C 步, 但没有每 C 个回合稳定)
 - 17: **end for**
-

6 SoftQ 算法

SoftQ 算法

- 1: 初始化参数 θ 和 ϕ
 - 2: 复制参数 $\bar{\theta} \leftarrow \theta, \bar{\phi} \leftarrow \phi$
 - 3: 初始化经验回放 D
 - 4: **for** 回合数 = 1, M **do**
 - 5: **for** 时步 = 1, t **do**
 - 6: 根据 $\mathbf{a}_t \leftarrow f^\phi(\xi; \mathbf{s}_t)$ 采样动作, 其中 $\xi \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
 - 7: 环境根据 a_t 反馈奖励 s_t 和下一个状态 s_{t+1}
 - 8: 存储 transition 即 (s_t, a_t, r_t, s_{t+1}) 到经验回放 D 中
 - 9: 更新环境状态 $s_{t+1} \leftarrow s_t$
 - 10: **更新 soft Q 函数参数:**
 - 11: 对于每个 $s_{t+1}^{(i)}$ 采样 $\{\mathbf{a}^{(i,j)}\}_{j=0}^M \sim q_{\mathbf{a}'}$
 - 12: 计算 empirical soft values $V_{\text{soft}}^\theta(\mathbf{s}_t)^1$
 - 13: 计算 empirical gradient $J_Q(\theta)^2$
 - 14: 根据 $J_Q(\theta)$ 使用 ADAM 更新参数 θ
 - 15: **更新策略:**
 - 16: 对于每个 $s_t^{(i)}$ 采样 $\{\xi^{(i,j)}\}_{j=0}^M \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
 - 17: 计算 $\mathbf{a}_t^{(i,j)} = f^\phi(\xi^{(i,j)}, \mathbf{s}_t^{(i)})$
 - 18: 使用经验估计计算 $\Delta f^\phi(\cdot; \mathbf{s}_t)^3$
 - 19: 计算经验估计 $\frac{\partial J_\pi(\phi; \mathbf{s}_t)}{\partial \phi} \propto \mathbb{E}_\xi \left[\Delta f^\phi(\xi; \mathbf{s}_t) \frac{\partial f^\phi(\xi; \mathbf{s}_t)}{\partial \phi} \right]$, 即 $\hat{\nabla}_\phi J_\pi$
 - 20: 根据 $\hat{\nabla}_\phi J_\pi$ 使用 ADAM 更新参数 ϕ
 - 21:
 - 22: **end for**
 - 23: 每 C 个回合复制参数 $\bar{\theta} \leftarrow \theta, \bar{\phi} \leftarrow \phi$
 - 24: **end for**
-

$$\begin{aligned}
 {}^1 V_{\text{soft}}^\theta(\mathbf{s}_t) &= \alpha \log \mathbb{E}_{q_{\mathbf{a}'}} \left[\frac{\exp(\frac{1}{\alpha} Q_{\text{soft}}^\theta(\mathbf{s}_t, \mathbf{a}'))}{q_{\mathbf{a}'}(\mathbf{a}')} \right] \\
 {}^2 J_Q(\theta) &= \mathbb{E}_{\mathbf{s}_t \sim q_{\mathbf{s}_t}, \mathbf{a}_t \sim q_{\mathbf{a}_t}} \left[\frac{1}{2} \left(\hat{Q}_{\text{soft}}^\theta(\mathbf{s}_t, \mathbf{a}_t) - Q_{\text{soft}}^\theta(\mathbf{s}_t, \mathbf{a}_t) \right)^2 \right] \\
 {}^3 \Delta f^\phi(\cdot; \mathbf{s}_t) &= \mathbb{E}_{\mathbf{a}_t \sim \pi^\phi} \left[\kappa \left(\mathbf{a}_t, f^\phi(\cdot; \mathbf{s}_t) \right) \nabla_{\mathbf{a}'} Q_{\text{soft}}^\theta(\mathbf{s}_t, \mathbf{a}') \Big|_{\mathbf{a}' = \mathbf{a}_t} \right. \\
 &\quad \left. + \alpha \nabla_{\mathbf{a}'} \kappa \left(\mathbf{a}', f^\phi(\cdot; \mathbf{s}_t) \right) \Big|_{\mathbf{a}' = \mathbf{a}_t} \right]
 \end{aligned}$$

7 SAC-S 算法

SAC-S 算法¹

```

1: 初始化参数  $\psi, \bar{\psi}, \theta, \phi$ 
2: for 回合数 = 1,  $M$  do
3:   for 时步 = 1,  $t$  do
4:     根据  $\mathbf{a}_t \sim \pi_\phi(\mathbf{a}_t | \mathbf{s}_t)$  采样动作  $a_t$ 
5:     环境反馈奖励和下一个状态,  $\mathbf{s}_{t+1} \sim p(\mathbf{s}_{t+1} | \mathbf{s}_t, \mathbf{a}_t)$ 
6:     存储 transition 到经验回放中,  $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\mathbf{s}_t, \mathbf{a}_t, r(\mathbf{s}_t, \mathbf{a}_t), \mathbf{s}_{t+1})\}$ 
7:     更新环境状态  $\mathbf{s}_{t+1} \leftarrow \mathbf{s}_t$ 
8:     更新策略:
9:      $\psi \leftarrow \psi - \lambda_V \hat{\nabla}_\psi J_V(\psi)$ 
10:     $\theta_i \leftarrow \theta_i - \lambda_Q \hat{\nabla}_{\theta_i} J_Q(\theta_i)$  for  $i \in \{1, 2\}$ 
11:     $\phi \leftarrow \phi - \lambda_\pi \hat{\nabla}_\phi J_\pi(\phi)$ 
12:     $\bar{\psi} \leftarrow \tau\psi + (1 - \tau)\bar{\psi}$ 
13:   end for
14: end for

```

¹Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor

8 SAC 算法

SAC 算法¹

- 1: 初始化网络参数 θ_1, θ_2 以及 ϕ
 - 2: 复制参数到目标网络 $\bar{\theta}_1 \leftarrow \theta_1, \bar{\theta}_2 \leftarrow \theta_2$,
 - 3: 初始化经验回放 D
 - 4: **for** 回合数 = 1, M **do**
 - 5: 重置环境, 获得初始状态 s_t
 - 6: **for** 时步 = 1, t **do**
 - 7: 根据 $\mathbf{a}_t \sim \pi_\phi(\mathbf{a}_t | \mathbf{s}_t)$ 采样动作 a_t
 - 8: 环境反馈奖励和下一个状态, $\mathbf{s}_{t+1} \sim p(\mathbf{s}_{t+1} | \mathbf{s}_t, \mathbf{a}_t)$
 - 9: 存储 transition 到经验回放中, $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\mathbf{s}_t, \mathbf{a}_t, r(\mathbf{s}_t, \mathbf{a}_t), \mathbf{s}_{t+1})\}$
 - 10: 更新环境状态 $s_{t+1} \leftarrow s_t$
 - 11: 更新策略:
 - 12: 更新 Q 函数, $\theta_i \leftarrow \theta_i - \lambda_Q \hat{\nabla}_{\theta_i} J_Q(\theta_i)$ for $i \in \{1, 2\}$ ²³
 - 13: 更新策略权重, $\phi \leftarrow \phi - \lambda_\pi \hat{\nabla}_\phi J_\pi(\phi)$ ⁴
 - 14: 调整 temperature, $\alpha \leftarrow \alpha - \lambda \hat{\nabla}_\alpha J(\alpha)$ ⁵
 - 15: 更新目标网络权重, $\bar{\theta}_i \leftarrow \tau \theta_i + (1 - \tau) \bar{\theta}_i$ for $i \in \{1, 2\}$
 - 16: **end for**
 - 17: **end for**
-

²Soft Actor-Critic Algorithms and Applications

² $J_Q(\theta) = \mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_t) \sim \mathcal{D}} \left[\frac{1}{2} (Q_\theta(\mathbf{s}_t, \mathbf{a}_t) - (r(\mathbf{s}_t, \mathbf{a}_t) + \gamma \mathbb{E}_{\mathbf{s}_{t+1} \sim p} [V_{\bar{\theta}}(\mathbf{s}_{t+1})]))^2 \right]$

³ $\hat{\nabla}_\theta J_Q(\theta) = \nabla_\theta Q_\theta(\mathbf{a}_t, \mathbf{s}_t) (Q_\theta(\mathbf{s}_t, \mathbf{a}_t) - (r(\mathbf{s}_t, \mathbf{a}_t) + \gamma (Q_{\bar{\theta}}(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}) - \alpha \log(\pi_\phi(\mathbf{a}_{t+1} | \mathbf{s}_{t+1}))))$

⁴ $\hat{\nabla}_\phi J_\pi(\phi) = \nabla_\phi \alpha \log(\pi_\phi(\mathbf{a}_t | \mathbf{s}_t)) + (\nabla_{\mathbf{a}_t} \alpha \log(\pi_\phi(\mathbf{a}_t | \mathbf{s}_t)) - \nabla_{\mathbf{a}_t} Q(\mathbf{s}_t, \mathbf{a}_t)) \nabla_\phi f_\phi(\epsilon_t; \mathbf{s}_t), \mathbf{a}_t = f_\phi(\epsilon_t; \mathbf{s}_t)$

⁵ $J(\alpha) = \mathbb{E}_{\mathbf{a}_t \sim \pi_t} [-\alpha \log \pi_t(\mathbf{a}_t | \mathbf{s}_t) - \alpha \bar{\mathcal{H}}]$