目录

1 模版备用

算法①

```
测试
```

---

① 脚注

2 Q learning 算法

Q-learning 算法①

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

---

①Reinforcement Learning: An Introduction

3 Sarsa 算法

Sarsa 算法①

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

---

① Reinforcement Learning: An Introduction

4 DQN 算法

DQN 算法①

```
初始化策略网络参数 $\theta$
复制参数到目标网络 $\hat{Q} \leftarrow Q$
初始化经验回放 $D$
for 回合数 $= 1, M$ do
    重置环境，获得初始状态 $s_t$
    for 时步 $= 1, t$ do
        根据 $\varepsilon - greedy$ 策略采样动作 $a_t$
        环境根据 $a_t$ 反馈奖励 $r_t$ 和下一个状态 $s_{t+1}$
        存储 transition 即 $(s_t, a_t, r_t, s_{t+1})$ 到经验回放 $D$ 中
        更新环境状态 $s_{t+1} \leftarrow s_t$
        更新策略:
        从 $D$ 中采样一个 batch 的 transition
        计算实际的 $Q$ 值，即 $y_j$②
        对损失 $L(\theta) = (y_i - Q(s_i, a_i; \theta))^2$ 关于参数 $\theta$ 做随机梯度下降③
    end for
    每 $C$ 个回合复制参数 $\hat{Q} \leftarrow Q$④]
end for
```

---

①Playing Atari with Deep Reinforcement Learning

②$$y_i = \begin{cases} r_i & \text{对于终止状态} s_{i+1} \\ r_i + \gamma \max_{a'} Q(s_{i+1}, a'; \theta) & \text{对于非终止状态} s_{i+1} \end{cases}$$

③$\theta_i \leftarrow \theta_i - \lambda \nabla_{\theta_i} L_i(\theta_i)$

④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

5 PER-DQN 算法

PER-DQN 算法①

```
初始化策略网络参数 $\theta$
复制参数到目标网络 $\hat{Q} \leftarrow Q$
初始化经验回放 $D$
for 回合数 $= 1, M$ do
    重置环境，获得初始状态 $s_t$
    for 时步 $= 1, t$ do
        根据 $\varepsilon - greedy$ 策略采样动作 $a_t$
        环境根据 $a_t$ 反馈奖励 $r_t$ 和下一个状态 $s_{t+1}$
        存储 transition 即 $(s_t, a_t, r_t, s_{t+1})$ 到经验回放 $D$，并根据 TD-error 损失确定其优先级 $p_t$
        更新环境状态 $s_{t+1} \leftarrow s_t$
        更新策略：
        按照经验回放中的优先级别，每个样本采样概率为 $P(j) = p_j^\alpha / \sum_i p_i^\alpha$，从 $D$ 中采样一个大小为 batch 的 transition
        计算各个样本重要性采样权重 $w_j = (N \cdot P(j))^{-\beta} / \max_i w_i$
        计算 TD-error $\delta_j$ ; 并根据 TD-error 更新优先级 $p_j$
        计算实际的 $Q$ 值，即 $y_j$②
        根据重要性采样权重调整损失 $L(\theta) = (y_j - Q(s_j, a_j; \theta) \cdot w_j)^2$，并将其关于参数 $\theta$ 做随机梯度下降③
    end for
    每 $C$ 个回合复制参数 $\hat{Q} \leftarrow Q$④]
end for
```

---

①Playing Atari with Deep Reinforcement Learning

②$$y_i = \begin{cases} r_i & \text{对于终止状态} s_{i+1} \\ r_i + \gamma \max_{a'} Q(s_{i+1}, a'; \theta) & \text{对于非终止状态} s_{i+1} \end{cases}$$

③$\theta_i \leftarrow \theta_i - \lambda \nabla_{\theta_i} L_i(\theta_i)$

④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

6 Policy Gradient 算法

REINFORCE 算法：Monte-Carlo Policy Gradient①

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

---

①Reinforcement Learning: An Introduction

7 Advantage Actor Critic 算法

Q Actor Critic 算法

```
初始化 Actor 参数 $\theta$ 和 Critic 参数 $w$
for 回合数 $= 1, M$ do
    根据策略 $\pi_\theta(a|s)$ 采样一个 (或几个) 回合的 transition
    更新 Critic 参数①
    for 时步 $= t+1, 1$ do
        计算 Advantage，即 $\delta_t = r_t + \gamma Q_w(s_{t+1}, a_{t+1}) - Q_w(s_t, a_t)$
        $w \leftarrow w + \alpha_w \delta_t \nabla_w Q_w(s_t, a_t)$
        $a_t \leftarrow a_{t+1}$,$s_t \leftarrow s_{t+1}$
    end for
    更新 Actor 参数 $\theta \leftarrow \theta + \alpha_\theta Q_w(s, a) \nabla_\theta \log \pi_\theta(a \mid s)$
end for
```

---

①这里结合 TD error 的特性按照从 $t+1$ 到 1 计算法 Advantage 更方便

8 PPO-Clip 算法

PPO-Clip 算法①②

```
初始化策略网络 (Actor) 参数 $\theta$ 和价值网络 (Critic) 参数 $\phi$
初始化 Clip 参数 $\epsilon$
初始化 epoch 数 $K$
初始化经验回放 $D$
for 回合数 $= 1, 2, \cdots, M$ do
    根据策略 $\pi_\theta$ 采样 $C$ 个时步数据，收集轨迹 $\tau = \{s_0, a_0, r_1, ..., s_t, a_t, r_{t+1}, \cdots\}$ 到经验回放 $D$ 中
    for epoch 数 $k = 1, 2, \cdots, K$ do
        计算折扣奖励 $\hat{R}_t$
        计算优势函数，即 $A^{\pi_{\theta_k}} = V_{\phi_k} - \hat{R}_t$
        结合重要性采样计算 Actor 损失，如下：
        $L^{CLIP}(\theta) = \frac{1}{|D|T} \sum_{\tau \in D} \sum_{t=0}^{T} min(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)} A^{\pi_{\theta_k}}(s_t, a_t), g(\epsilon, A^{\pi_{\theta_k}}(s_t, a_t)))$③

        梯度下降更新 Actor 参数：$\theta_{k+1} \leftarrow \theta_k + \alpha_\theta L^{CLIP}(\theta)$
        更新 Critic 参数:
        $\phi_{k+1} \leftarrow \phi_k + \alpha_\phi \frac{1}{|D|T} \sum_{\tau \in D} \sum_{t=0}^{T} (V_{\phi_k}(s_t) - \hat{R}_t)^2$
    end for
    清空经验回放 $D$
end for
```

---

① Proximal Policy Optimization Algorithms

② https://spinningup.openai.com/en/latest/algorithms/ppo.html

③ $L^{CLIP}(\theta) = \hat{E}_t[min(r_t(\theta)\hat{A}_t, clip(r_t(\theta), 1 - \epsilon, 1 + \epsilon)\hat{A}_t)]$

9 PPO-KL 散度算法

PPO-KL 散度算法①②

```
初始化策略网络 (Actor) 参数 $\theta$ 和价值网络 (Critic) 参数 $\phi$
初始化 KL 散度参数 $\lambda$
初始化回合数量 $M$
初始化 epoch 数量 $K$
初始化经验回放 $D$
for 回合数 $= 1, 2, \cdots, M$ do
    根据策略 $\pi_\theta$ 采样 $C$ 个时步数据，收集轨迹 $\tau = \{s_0, a_0, r_1, ..., s_t, a_t, r_{t+1}, \cdots\}$ 到经验回放 $D$ 中
    for epoch 数 $k = 1, 2, \cdots, K$ do
        计算折扣奖励 $\hat{R}_t$
        计算优势函数，即 $A^{\pi_{\theta_k}} = V_{\phi_k} - \hat{R}_t$
        结合重要性采样计算 Actor 损失，如下：
        $J_{PPO}(\theta) = \frac{1}{|D|T} \sum_{\tau \in D} \sum_{t=1}^{T} \frac{\pi_\theta(a_t|s_t)}{\pi_{old}(a_t|s_t)} \hat{A}_t - \lambda KL[\pi_{old}|\pi_\theta]$
        梯度下降更新 Actor 参数：$\theta_{k+1} \leftarrow \theta_k + \alpha_\theta J^{PPO}(\theta)$
        更新 Critic 参数:
        $\phi_{k+1} \leftarrow \phi_k + \alpha_\phi \frac{1}{|D|T} \sum_{\tau \in D} \sum_{t=0}^{T} (V_{\phi_k}(s_t) - \hat{R}_t)^2$
        if $KL[\pi_{old}|\pi_\theta] > \beta_{high} KL_{target}$ then
            $\lambda \leftarrow \alpha\lambda$
        else if $KL[\pi_{old}|\pi_\theta] < \beta_{low} KL_{target}$ then
            $\lambda \leftarrow \frac{\lambda}{\alpha}$
        end if
    end for
    清空经验回放 $D$
end for
```

---

①Proximal Policy Optimization Algorithms

②Emergence of Locomotion Behaviours in Rich Environments

10 DDPG 算法

DDPG 算法①

```
初始化 critic 网络 $Q\left(s, a \mid \theta^{Q}\right)$ 和 actor 网络 $\mu(s|\theta^{\mu})$ 的参数 $\theta^{Q}$ 和 $\theta^{\mu}$
初始化对应的目标网络参数，即 $\theta^{Q'} \leftarrow \theta^{Q}, \theta^{\mu'} \leftarrow \theta^{\mu}$
初始化经验回放 $R$
for 回合数 $= 1, M$ do
    选择动作 $a_t = \mu\left(s_t \mid \theta^{\mu}\right) + \mathcal{N}_t$，$\mathcal{N}_t$ 为探索噪声
    环境根据 $a_t$ 反馈奖励 $s_t$ 和下一个状态 $s_{t+1}$
    存储 transition$(s_t, a_t, r_t, s_{t+1})$ 到经验回放 $R$ 中
    更新环境状态 $s_{t+1} \leftarrow s_t$
    更新策略:
    从 $R$ 中取出一个随机批量的 $(s_i, a_i, r_i, s_{i+1})$
    求得 $y_i = r_i + \gamma Q'\left(s_{i+1}, \mu'\left(s_{i+1} \mid \theta^{\mu'}\right) \mid \theta^{Q'}\right)$
    更新 critic 参数，其损失为：$L = \frac{1}{N}\sum_i \left(y_i - Q\left(s_i, a_i \mid \theta^{Q}\right)\right)^2$
    更新 actor 参数：$\nabla_{\theta^{\mu}} J \approx \frac{1}{N}\sum_i \nabla_a Q\left(s, a \mid \theta^{Q}\right)\big|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^{\mu}} \mu\left(s \mid \theta^{\mu}\right)\Big|_{s_i}$
    软更新目标网络：$\theta^{Q'} \leftarrow \tau\theta^{Q} + (1-\tau)\theta^{Q'}$，$\theta^{\mu'} \leftarrow \tau\theta^{\mu} + (1-\tau)\theta^{\mu'}$
end for
```

---

①Continuous control with deep reinforcement learning

11 SoftQ 算法

SoftQ 算法

```
初始化参数 $\theta$ 和 $\phi$
复制参数 $\bar{\theta} \leftarrow \theta, \bar{\phi} \leftarrow \phi$
初始化经验回放 $D$
for 回合数 $= 1, M$ do
    for 时步 $= 1, t$ do
        根据 $\mathbf{a}_t \leftarrow f^\phi(\xi; \mathbf{s}_t)$ 采样动作，其中 $\xi \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
        环境根据 $a_t$ 反馈奖励 $s_t$ 和下一个状态 $s_{t+1}$
        存储 transition 即 $(s_t, a_t, r_t, s_{t+1})$ 到经验回放 $D$ 中
        更新环境状态 $s_{t+1} \leftarrow s_t$
        更新 soft Q 函数参数:
        对于每个 $s_{t+1}^{(i)}$ 采样 $\left\{\mathbf{a}^{(i,j)}\right\}_{j=0}^{M} \sim q_{\mathbf{a}'}$
        计算 empirical soft values $V_{\text{soft}}^{\theta}(\mathbf{s}_t)$①
        计算 empirical gradient $J_Q(\theta)$②
        根据 $J_Q(\theta)$ 使用 ADAM 更新参数 $\theta$
        更新策略:
        对于每个 $s_t^{(i)}$ 采样 $\left\{\xi^{(i,j)}\right\}_{j=0}^{M} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
        计算 $\mathbf{a}_t^{(i,j)} = f^\phi\left(\xi^{(i,j)}, \mathbf{s}_t^{(i)}\right)$
        使用经验估计计算 $\Delta f^\phi(\cdot; \mathbf{s}_t)$③
        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 $\hat{\nabla}_\phi J_\pi$
        根据 $\hat{\nabla}_\phi J_\pi$ 使用 ADAM 更新参数 $\phi$

    end for
    每 $C$ 个回合复制参数 $\bar{\theta} \leftarrow \theta, \bar{\phi} \leftarrow \phi$
end for
```

---

① $V_{\text{soft}}^{\theta}(\mathbf{s}_t) = \alpha \log \mathbb{E}_{q_{\mathbf{a}'}}\left[\frac{\exp\left(\frac{1}{\alpha} Q_{\text{soft}}^{\theta}(\mathbf{s}_t, \mathbf{a}')\right)}{q_{\mathbf{a}'}(\mathbf{a}')}\right]$

② $J_Q(\theta) = \mathbb{E}_{\mathbf{s}_t \sim q_{\mathbf{s}_t}, \mathbf{a}_t \sim q_{\mathbf{a}_t}}\left[\frac{1}{2}\left(\hat{Q}_{\text{soft}}^{\bar{\theta}}(\mathbf{s}_t, \mathbf{a}_t) - Q_{\text{soft}}^{\theta}(\mathbf{s}_t, \mathbf{a}_t)\right)^2\right]$

③
$$\begin{aligned}\Delta f^\phi(\cdot; \mathbf{s}_t) = & \mathbb{E}_{\mathbf{a}_t \sim \pi^\phi}\left[\kappa\left(\mathbf{a}_t, f^\phi(\cdot; \mathbf{s}_t)\right) \nabla_{\mathbf{a}'} Q_{\text{soft}}^{\theta}(\mathbf{s}_t, \mathbf{a}')\Big|_{\mathbf{a}'=\mathbf{a}_t}\right. \\ & \left. + \alpha \nabla_{\mathbf{a}'} \kappa\left(\mathbf{a}', f^\phi(\cdot; \mathbf{s}_t)\right)\Big|_{\mathbf{a}'=\mathbf{a}_t}\right]\end{aligned}$$

12 SAC-S 算法

SAC-S 算法①

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

---

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

13 SAC 算法

SAC 算法①

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

---

②Soft Actor-Critic Algorithms and Applications

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

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

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

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