目录

1 模版备用

算法

1: 测试

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

[1]Reinforcement 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 Policy Gradient 算法

REINFORCE 算法：Monte-Carlo Policy Gradient[1]

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 $= 1, t$ **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

[1] Reinforcement Learning: An Introduction

5 DQN 算法

DQN 算法[1]

初始化策略网络参数 θ
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 θ 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

[1] Playing 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 SoftQ 算法

SoftQ 算法

初始化参数 θ 和 ϕ
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 θ
    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 ϕ

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

7 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

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

8 SAC 算法

SAC 算法[1]

初始化网络参数 θ_1, θ_2 以及 ϕ
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 $\boldsymbol{a}_t \sim \pi_\phi\left(\boldsymbol{a}_t \mid \mathbf{s}_t\right)$ 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：
    更新 Q 函数，$\theta_i \leftarrow \theta_i - \lambda_Q \hat{\nabla}_{\theta_i} J_Q\left(\theta_i\right)$ for $i \in \{1,2\}$[2][3]
    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

[2]Soft Actor-Critic Algorithms and Applications

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

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

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

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