Policy Gradient策略梯度

七月在线 褚则伟
zeweichu@gmail.com
2017年10月

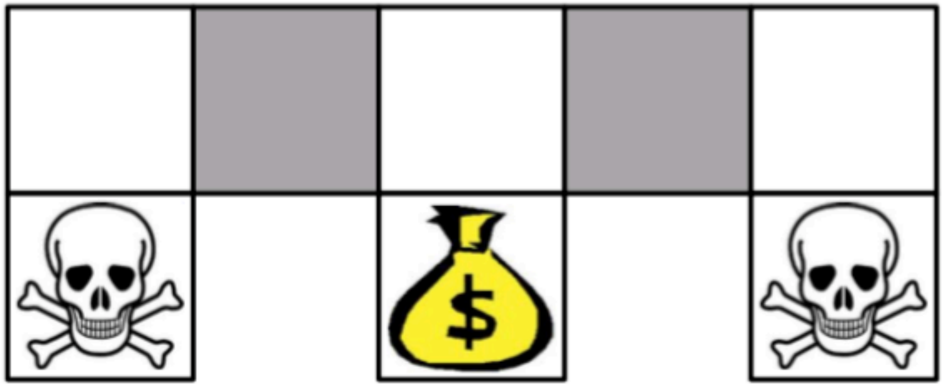
目录

- Policy Gradient
- Actor Critic
- Continuous Mountain Car代码实战

增强学习的一些分类

- Value based
  - 值函数
  - Q值函数
- Policy Based
  - 不需要值函数
  - 直接优化Policy
- Actor Critic
  - 学习值函数
  - 学习Policy

Deterministic policy的问题



- The agent cannot differentiate the grey states
- Consider features of the following form (for all N, E, S, W)

$$\phi(s, a) = \mathbf{1}(\text{wall to N}, a = \text{move E})$$

- Compare value-based RL, using an approximate value function

$$Q_\theta(s, a) = f(\phi(s, a), \theta)$$

- To policy-based RL, using a parametrised policy

$$\pi_\theta(s, a) = g(\phi(s, a), \theta)$$

Policy Network

- 不需要优化Q值函数，直接优化策略函数 π

$$a = \pi(a|s, \mathbf{u}) \text{ or } a = \pi(s, \mathbf{u})$$

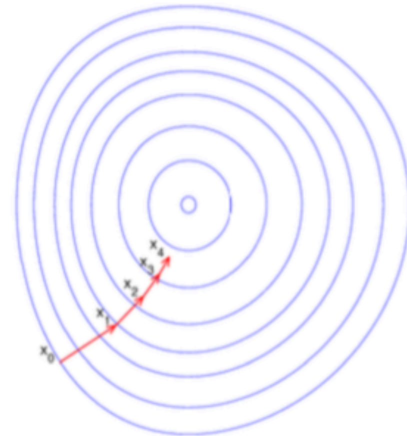
- 优化discounted reward

$$L(\mathbf{u}) = \mathbb{E}\left[r_1 + \gamma r_2 + \gamma^2 r_3 + \dots \mid \pi(\cdot, \mathbf{u})\right]$$

- 直接用SGD做优化

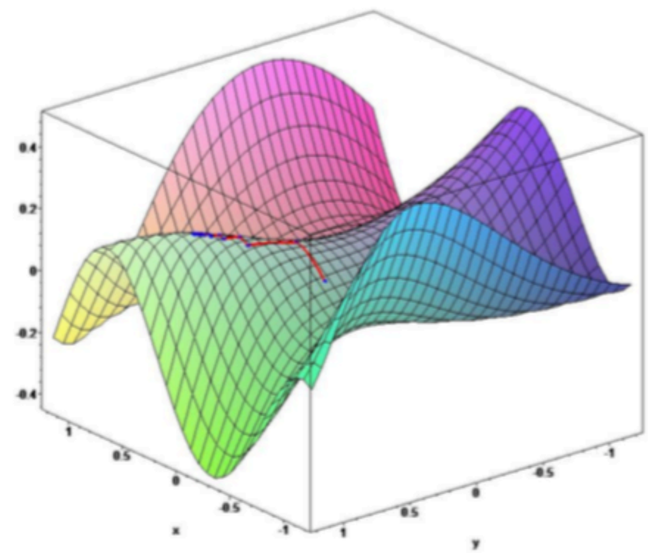
Gradient Ascent

- Let $J(\theta)$ be any policy objective function
- Policy gradient algorithms search for a *local* maximum in $J(\theta)$ by ascending the gradient of the policy, w.r.t. parameters θ



$$\Delta\theta = \alpha \nabla_\theta J(\theta)$$

- Where $\nabla_\theta J(\theta)$ is the policy gradient



$$\nabla_\theta J(\theta) = \begin{pmatrix} \frac{\partial J(\theta)}{\partial \theta_1} \\ \vdots \\ \frac{\partial J(\theta)}{\partial \theta_n} \end{pmatrix}$$

- and α is a step-size parameter

Policy Objective

- 给定一个策略 $\pi_\theta(s,a)$ 和参数 θ 如何找到最佳的 θ
- 如何确定一个策略 π_θ 的好坏?
- 直接用SGD做优化

One step MDP

- 如果我们只考虑一步MDP
  - 初始状态s ~ d(s)
  - 我们考虑一步就结束 $r = \mathcal{R}_{s,a}$

$$
\begin{aligned}
J(\theta) &= \mathbb{E}_{\pi_\theta}[r] \\
&= \sum_{s \in \mathcal{S}} d(s) \sum_{a \in \mathcal{A}} \pi_\theta(s,a) \mathcal{R}_{s,a}
\end{aligned}
$$

$$
\begin{aligned}
\nabla_\theta J(\theta) &= \sum_{s \in \mathcal{S}} d(s) \sum_{a \in \mathcal{A}} \pi_\theta(s,a) \nabla_\theta \log \pi_\theta(s,a) \mathcal{R}_{s,a} \\
&= \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s,a) r]
\end{aligned}
$$

一个小技巧

- 如何得到参数对一个policy的gradient

$$
\begin{aligned}
\nabla_\theta \pi_\theta(s,a) &= \pi_\theta(s,a)\frac{\nabla_\theta \pi_\theta(s,a)}{\pi_\theta(s,a)} \\
&= \pi_\theta(s,a)\nabla_\theta \log \pi_\theta(s,a)
\end{aligned}
$$

Multi-step MDP

- 如果我们考虑长线的回报，那么就要考虑优化Q(s, a)乘上概率

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(s,a)\ Q^{\pi_\theta}(s,a)\right]$$

考虑整个trajectory

- 如果我们考虑一整个trajectory

We let τ denote a state-action sequence $s_0, u_0, \ldots, s_H, u_H$. We overload notation: $R(\tau) = \sum_{t=0}^{H} R(s_t, u_t)$.

$$U(\theta) = \mathrm{E}[\sum_{t=0}^{H} R(s_t, u_t); \pi_\theta] = \sum_{\tau} P(\tau; \theta) R(\tau)$$

In our new notation, our goal is to find θ:

$$\max_{\theta} U(\theta) = \max_{\theta} \sum_{\tau} P(\tau; \theta) R(\tau)$$

考虑整个trajectory

- 如果我们考虑一整个trajectory

$$\nabla U(\theta) \approx \hat{g} = \frac{1}{m} \sum_{i=1}^{m} \nabla_\theta \log P(\tau^{(i)}; \theta) R(\tau^{(i)})$$

展开整个trajectory

$$\nabla_\theta \log P(\tau^{(i)};\theta) = \nabla_\theta \log \left[\prod_{t=0}^{H} \underbrace{P(s_{t+1}^{(i)}|s_t^{(i)}, u_t^{(i)})}_{\text{dynamics model}} \cdot \underbrace{\pi_\theta(u_t^{(i)}|s_t^{(i)})}_{\text{policy}} \right]$$

$$= \nabla_\theta \left[\sum_{t=0}^{H} \log P(s_{t+1}^{(i)}|s_t^{(i)}, u_t^{(i)}) + \sum_{t=0}^{H} \log \pi_\theta(u_t^{(i)}|s_t^{(i)}) \right]$$

$$= \nabla_\theta \sum_{t=0}^{H} \log \pi_\theta(u_t^{(i)}|s_t^{(i)})$$

- 与P无关

REINFORCE

function REINFORCE

Initialise θ arbitrarily

for each episode $\{s_1, a_1, r_2, ..., s_{T-1}, a_{T-1}, r_T\} \sim \pi_\theta$ **do**

for $t = 1$ to $T - 1$ **do**

$\theta \leftarrow \theta + \alpha \nabla_\theta \log \pi_\theta(s_t, a_t) v_t$

end for

end for

return θ

end function

Policy Gradient小结

- Policy $\pi(\theta)$ 是一个神经网络
- 用初始状态的回报作为优化的目标

$$V_{\pi(\theta)} = \mathbb{E}_{\pi(\theta)}[r_0 + \gamma r_1 + \gamma^2 r_2 + \cdots]$$

- Gradient ascent

$$\theta_{t+1} = \theta_t + \alpha \nabla J(\theta_t)$$

- 利用policy gradient拟合gradient

$$\nabla J(\theta) = \mathbb{E}_{\pi}[\gamma^t R_t \nabla_\theta \log \pi(a|s_t, \theta)]$$

Actor-critic

- Actor: 策略(policy)网络, 选择下一个动作
- Critic: 评估Q(s,a)的近似值，相当于策略评估

- 优化discounted reward

- 直接用SGD做优化

Actor-critic

function QAC
    Initialise s, θ
    Sample $a \sim \pi_\theta$
    for each step **do**
        Sample reward $r = \mathcal{R}_s^a$; sample transition $s' \sim \mathcal{P}_{s,\cdot}^a$
        Sample action $a' \sim \pi_\theta(s', a')$
        $\delta = r + \gamma Q_w(s', a') - Q_w(s, a)$
        $\theta = \theta + \alpha \nabla_\theta \log \pi_\theta(s, a) Q_w(s, a)$
        $w \leftarrow w + \beta \delta \phi(s, a)$
        $a \leftarrow a', s \leftarrow s'$
    end for
end function

Compatible Function Approximation

- 如果Value Function与policy compatible

$$\nabla_w Q_w(s,a) = \nabla_\theta \log \pi_\theta(s,a)$$

- value function最小化MSE

$$\varepsilon = \mathbb{E}_{\pi_\theta}\left[\left(Q^{\pi_\theta}(s,a) - Q_w(s,a)\right)^2\right]$$

- 那我们就可以用它来做policy gradient

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(s,a)\ Q_w(s,a)\right]$$

Actor-critic

- 证明

$$
\begin{aligned}
\nabla_w \varepsilon &= 0 \\
\mathbb{E}_{\pi_\theta}\left[(Q^\theta(s,a) - Q_w(s,a))\nabla_w Q_w(s,a)\right] &= 0 \\
\mathbb{E}_{\pi_\theta}\left[(Q^\theta(s,a) - Q_w(s,a))\nabla_\theta \log \pi_\theta(s,a)\right] &= 0 \\
\mathbb{E}_{\pi_\theta}\left[Q^\theta(s,a)\nabla_\theta \log \pi_\theta(s,a)\right] &= \mathbb{E}_{\pi_\theta}\left[Q_w(s,a)\nabla_\theta \log \pi_\theta(s,a)\right]
\end{aligned}
$$

$$
\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(s,a)\, Q_w(s,a)\right]
$$

Actor-critic

- Actor: 策略(policy)网络, 选择下一个动作
- Critic: 评估Q(s,a)的近似值，相当于策略评估

$$
\begin{aligned}
\mathbb{E}[\nabla_\theta \log \pi_\theta(s,a) B(s)] &= \sum_{s \in \mathcal{S}} d^{\pi_\theta} \sum_{a \in \mathcal{A}} \pi_\theta(s,a) \frac{\nabla_\theta \pi_\theta(s,a)}{\pi_\theta(s,a)} B(s) \\
&= \sum_{s \in \mathcal{S}} d^{\pi_\theta} B(s) \nabla_\theta \sum_{a \in \mathcal{A}} \pi_\theta(s,a) \\
&= 0
\end{aligned}
$$

$$A^{\pi_\theta}(s,a) = Q^{\pi_\theta}(s,a) - V^{\pi_\theta}(s)$$

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(s,a)\ A^{\pi_\theta}(s,a)\right]$$

Actor-critic小结

- 用神经网络（或其他）来拟合advantage function
- 用神经网络（或其他）来拟合策略网络
- 同步更新actor和critic

AlphaGo Zero

- Silver et. al., Mastering the game of Go without human knowledge

Project: Continuous Mountain Car

Thank you!

Q&A