强化学习原理

DDPG 论文笔记

姓名：石若川　学号：2111381　专业：智能科学与技术

1 Deterministic Policy Gradient(DPG)

David Silver 等人与 2014 年提出了 DPG 算法，将连续动作空间中的确定性策略梯度算法用于强化学习。一般的策略梯度算法基本思想中利用参数化的概率 $\pi_\theta(a|s) = \mathbb{P}[a|s;\theta]$ 来表示随机策略。而 DPG 中，作者使用确定性的策略 $a = \mu_\theta(s)$。随机策略在实际使用中需要更大的样本，需要更多的计算资源，而确定性策略能够避免这一问题。然而确定性的策略只采样一个动作，难以保证探索。为了保证算法进行探索，作者引入了 off-policy 的算法，使用随机策略选作为行为策略，使用确定性策略作为目标策略，算法使用 actor-critic 架构。

1.1 随机策略

对于带折扣的 MDP 问题，作者对折扣状态分布进行如下定义：

$$\int_{\mathcal{S}} \sum_{t=1}^{\infty} \gamma^{t-1} p_1(s) p(s \to s', t, \pi) \mathrm{d}s$$

因此可以将目标函数写为：

$$\begin{aligned} J(\pi_\theta) &= \int_{\mathcal{S}} \rho^\pi(s) \int_{\mathcal{A}} \pi_\theta(s,a) r(s,a) \mathrm{d}a \mathrm{d}s \\ &= \mathbb{E}_{s\sim\rho^\pi, a\sim\pi_\theta}[r(s,a)] \end{aligned} \tag{1}$$

随机策略的梯度可以写为

$$\begin{aligned} \nabla_\theta J(\pi_\theta) &= \int_{\mathcal{S}} \rho^\pi(s) \int_{\mathcal{A}} \nabla_\theta \pi_\theta(a|s) Q^\pi(s,a) \mathrm{d}a \mathrm{d}s \\ &= \mathbb{E}_{s\sim\rho^\pi, a\sim\pi_\theta}[\nabla_\theta \log \pi_\theta(a|s) Q^\pi(s,a)] \end{aligned} \tag{2}$$

Actor-Critic 广泛应用的一种基于策略梯度架构。式2中的 $Q^\pi(s,a)$ 为未知的动作值函数，Critic 中使用 $Q^w(s,a)$ 去近似 $Q^\pi(s,a)$。当满足以下条件时，$Q^w(s,a)$ 为无偏估计：

- $Q^w(s,a) = \nabla_\theta \log \pi_\theta(a \mid s)^T w$
- w 使得 $\varepsilon^2(w) = E_{s\sim\rho^\pi, a\sim\pi_\theta}\left[(Q^{\mathrm{w}}(s,a) - Q^\pi(s,a))^2\right]$ 最小

所以，式2可以写为：

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{s\sim\rho^\pi, a\sim\pi_\theta}[\nabla_\theta \log \pi_\theta(a|s) Q^w(s,a)] \tag{3}$$

对于 off-policy 的 Actor-Critic 框架而言，由行为策略 $\beta(a|s) \neq \pi_\theta(a|s)$ 采样动作，目标函数写为：

$$\begin{aligned} J_\beta(\pi_\theta) &= \int_\mathcal{S} \rho^\beta(s) V^\pi(s) \mathrm{d}s \\ &= \int_\mathcal{S} \int_\mathcal{A} \rho^\beta(s) \pi_\theta(a|s) Q^\pi(s,a) \mathrm{d}a\mathrm{d}s \end{aligned}$$

目标函数的梯度写为：

$$\nabla_\theta J_\beta(\pi_\theta) \approx \int_\mathcal{S} \int_\mathcal{A} \rho^\beta(s) \nabla_\theta \pi_\theta(a|s) Q^\pi(s,a) \mathrm{d}a\mathrm{d}s \tag{4}$$

$$= \mathbb{E}_{s\sim\rho^\beta, a\sim\beta}\left[\frac{\pi_\theta(a|s)}{\beta_\theta(a|s)} \nabla_\theta \log \pi_\theta(a|s) Q^\pi(s,a)\right] \tag{5}$$

1.2 确定性策略

考虑确定性策略 $\mu_\theta : \mathcal{S} \to \mathcal{A}$，目标函数为：

$$\begin{aligned} J(\mu_\theta) &= \int_\mathcal{S} \rho^\mu(s) r(s, \mu_\theta(s)) \mathrm{d}s \\ &= \mathbb{E}_{s\sim\rho^\mu}\left[r(s, \mu_\theta(s))\right] \end{aligned} \tag{6}$$

梯度为：

$$\begin{aligned} \nabla_\theta J(\mu_\theta) &= \int_\mathcal{S} \rho^\mu(s) \nabla_\theta \mu_\theta(s) \nabla_a Q^\mu(s,a)|_{a=\mu_\theta(s)} \mathrm{d}s \\ &= \mathbb{E}_{s\sim\rho^\mu}\left[\nabla_\theta \mu_\theta(s) \nabla_a Q^\mu(s,a)|_{a=\mu_\theta(s)}\right] \end{aligned} \tag{7}$$

对于确定性策略的 Actor-Critic 架构，利用 $\mu_\theta(s)$ 替换随机策略中的 $\pi(s,a)$，目标函数写为：

$$\begin{aligned} J_\beta(\mu_\theta) &= \int_\mathcal{S} \rho^\beta(s) V^\mu(s) \mathrm{d}s \\ &= \int_\mathcal{S} \rho^\beta(s) Q^\mu(s, \mu_\theta(s)) \mathrm{d}s \end{aligned} \tag{8}$$

梯度为：

$$\begin{aligned} \nabla_\theta J_\beta(\mu_\theta) &\approx \int_\mathcal{S} \rho^\beta(s) \nabla_\theta \mu_\theta(a|s) Q^\mu(s,a) \mathrm{d}s \\ &= \mathbb{E}_{s\sim\rho^\beta}\left[\nabla_\theta \mu_\theta(s) \nabla_a Q^\mu(s,a)|_{a=\mu_\theta(s)}\right] \end{aligned} \tag{9}$$

策略的更新过程如下，其中 Critic 使用 Q-learning 的方法进行更新。

$$\delta_t = r_t + \gamma Q^w(s_{t+1}, \mu_\theta(s_{t+1})) - Q^w(s_t, a_t) \tag{10}$$

$$w_{t+1} = w_t + \alpha_w \delta_t \nabla_w Q^w(s_t, a_t) \tag{11}$$

$$\theta_{t+1} = \theta_t + \alpha_\theta \nabla_\theta \mu_\theta(s_t) \nabla_a Q^w(s_t, a_t)|_{a=\mu_\theta(s)} \tag{12}$$

一般的 off-policy 型 Actor-Critic 算法均需要进行重要性采样，但是由于确定性策略去掉了对于动作的积分，所以在 Actor 中不需要进行重要性采样。

1.3 相容函数逼近

作者提出了在不影响确定性策略梯度下，使得 $\nabla_a Q^\mu(s,a)$ 可以被 $\nabla_a Q^w(s,a)$ 替代的相容条件。当满足以下条件时，对于确定性策略 $\mu_\theta(s)$，$\nabla_\theta J_\beta(\theta) = \mathbb{E}\left[\nabla_\theta \mu_\theta(s) \nabla_a Q^w(s,a)|_{a=\mu_\theta(s)}\right]$

1. $\nabla_a Q^w(s,a)|_{a=\mu_\theta(s)} = \nabla_\theta \mu_\theta(s)^\top w$
2. w 使得均方误差 $MSE(\theta,w) = \mathbb{E}\left[\epsilon(s;\theta,w)^T \epsilon(s;\theta,w)\right]$ 最小，其中 $\epsilon(s;\theta,w) = \nabla_a Q^w(s,a)|_{a=\mu_\theta(s)} - \nabla_a Q^\mu(s,a)|_{a=\mu_\theta(s)}$

2 Deep Deterministic Policy Gradient(DDPG)

DPG 所提出的相容函数条件过于苛刻，在实际使用中很难满足。DDPG 提出了一种结合 DQN 和 DPG 的算法。

DPG 使用确定性策略选择动作，Critic 利用 Q-learning 形式的贝尔曼方程更新行为值函数 $Q(s,a)$，Actor 的利用以下公式进行更新：

$$\begin{aligned}
\nabla_{\theta^\mu} J &\approx \mathbb{E}_{s_t \sim \rho^\beta}\left[\nabla_{\theta^\mu} Q(s,a|\theta^Q)|_{s=s_t, a=\mu(s_t|\theta^\mu)}\right] \\
&= \mathbb{E}_{s_t \sim \rho^\beta}\left[\nabla_a Q(s,a|\theta^Q)|_{s=s_t, a=\mu(s_t)} \nabla_{\theta_\mu} \mu(s|\theta^\mu)|_{s=s_t}\right]
\end{aligned}$$

DQN 利用神经网络拟合行为值函数，解决具有连续状态空间和动作空间的强化学习问题。在 DQN 出现之前，利用神经网络拟合行为值函数进行训练时会出现不稳定的情况。这是因为训练神经网络时，需要满足样本是独立同分布的假设。然而，采样的样本轨迹会依赖前一时刻的状态-动作，因此并不满足独立同分布。DQN 使用经验回放机制来训练神经网络。它将代理与环境的交互经验存储在一个经验池中，然后随机抽样这些经验进行训练。这种随机抽样有助于打破数据之间的相关性，并使得训练更加稳定。另外，DQN 引入了独立的目标网络，它与在线网络具有相同的架构，但是参数更新频率较低。这种延迟更新的策略有助于减小 TD 学习中的偏差，提高算法的稳定性。

DDPG 结合了 DQN 和 DPG 的思想，将 DPG 中的行为值函数利用神经网络表示。DDPG 同样使用经验回放，将采样的 (s_t, a_t, r_t, s_{t+1}) 存入回放缓冲区中，每个时间步 Actor 和 Critic 从经验缓冲区中均匀采样一个 minibatch 进行更新。不同于 DQN 通过直接复制权重更新网络参数，DDPG 利用滤波的方法更新网络参数 θ：

$$\theta' \leftarrow \tau\theta + (1-\tau)\theta' \quad \tau \ll 1$$

这是一种软更新的方式，可以极大地提高学习的稳定性。

由于观测空间中不同物理量的单位不同，所以神经网络难以找到合适的超参数进行有效学习。因此，DDPG 引入了批归一化的方法，使得样本具有相同的均值和方差。借助批归一化，网络可以在具有不同单位的任务重学习，无需手动调整单位在一定范围之内

最后，为促进探索，DDPG 对 Actor 策略添加了 Ornstein-Uhlenbeck 噪音过程 $\mathcal{N}$，构造出一个探索策略 μ'：

$$\mu'(s_t) = \mu(s_t|\theta_t^\mu) + \mathcal{N}$$

DDPG 的伪代码如下所示：

Algorithm 1 DDPG algorithm

Randomly initialize critic network $Q(s,a|\theta^Q)$ and actor $\mu(s|\theta^\mu)$ with weights θ^Q and θ^μ.
Initialize target network Q' and μ' with weights $\theta^{Q'} \leftarrow \theta^Q$, $\theta^{\mu'} \leftarrow \theta^\mu$
Initialize replay buffer R
for episode = 1, M **do**
  Initialize a random process $\mathcal{N}$ for action exploration
  Receive initial observation state s_1
  for t = 1, T **do**
    Select action $a_t = \mu(s_t|\theta^\mu) + \mathcal{N}_t$ according to the current policy and exploration noise
    Execute action a_t and observe reward r_t and observe new state s_{t+1}
    Store transition (s_t, a_t, r_t, s_{t+1}) in R
    Sample a random minibatch of N transitions (s_i, a_i, r_i, s_{i+1}) from R
    Set $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$
    Update critic by minimizing the loss: $L = \frac{1}{N}\sum_i (y_i - Q(s_i, a_i|\theta^Q))^2$
    Update the actor policy using the sampled policy gradient:

$$\nabla_{\theta^\mu} J \approx \frac{1}{N}\sum_i \nabla_a Q(s,a|\theta^Q)|_{s=s_i,a=\mu(s_i)} \nabla_{\theta^\mu}\mu(s|\theta^\mu)|_{s_i}$$

    Update the target networks:

$$\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'}$$
$$\theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'}$$

  end for
end for

3 实验结果

作者在多种模拟物理环境下进行实验测试如图3.1。测试结果表明，目标网络和批归一化的设置是必要的，没有目标网络的传统 DPG 在许多环境中表现较差。

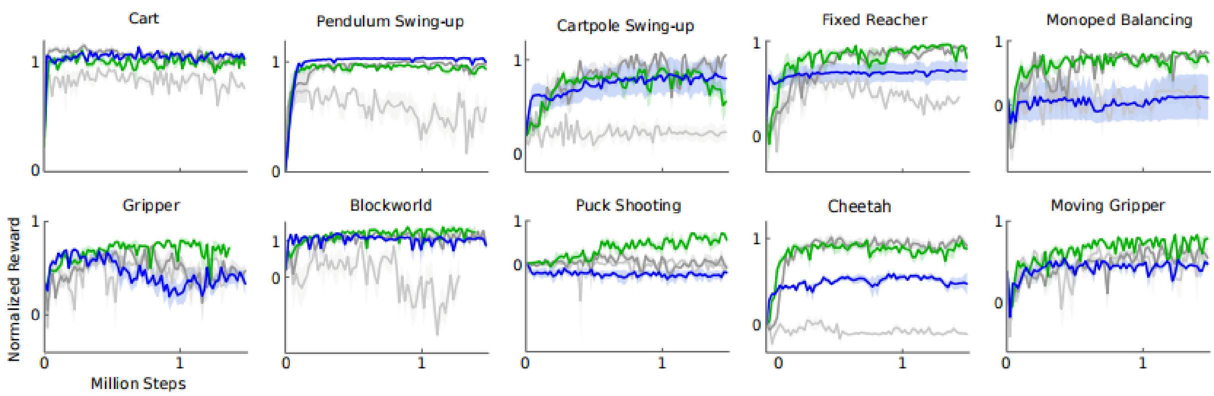

Figure 2: Performance curves for a selection of domains using variants of DPG: original DPG algorithm (minibatch NFQCA) with batch normalization (light grey), with target network (dark grey), with target networks and batch normalization (green), with target networks from pixel-only inputs (blue). Target networks are crucial.

图 3.1: 实验结果

图3.2说明在简单任务中，DDPG 能准确估算收益，没有系统性偏差。对于难度较大的任务，Q 行为值函数的估计值更差，但 DDPG 仍能学习到良好的策略。

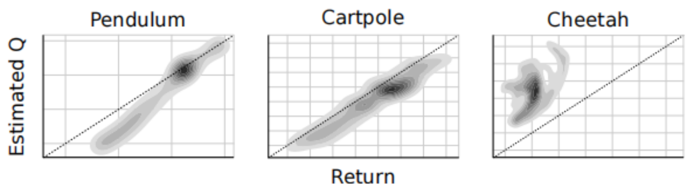

Figure 3: Density plot showing estimated Q values versus observed returns sampled from test episodes on 5 replicas. In simple domains such as pendulum and cartpole the Q values are quite accurate. In more complex tasks, the Q estimates are less accurate, but can still be used to learn competent policies. Dotted line indicates unity, units are arbitrary.

图 3.2: 行为值函数估计

4 总结

DDPG 结合了 DQN 和 DPG，提出了一种算法，能够在具有连续动作空间的多个领域中稳定解决具有挑战性的问题。在不需要对环境进行任何修改的情况下，学习过程是稳定的。

实验中发现在 Atari 问题中 DDPG 比 DQN 找到解决方案时所需的步骤要少得多。这表明，若有更多的模拟时间，DDPG 可能可以解决更困难的问题。