Stochastic Policy and Deterministic Policy

背景与定义

策略（Policy），是基于状态（State）决定动作（Action）的方法。强化学习是寻找可以最大化序列决策中的收益的策略的一系列算法。根据策略函数生成动作的数学性质，可以分为确定性策略（Deterministic Policy）和随机性策略（Stochastic Policy）两个类型。

确定性策略，是使用确定性函数 (Deterministic Function) 建模的策略。

> 即 $a = \pi(s)$ ， $\pi : S \to A$
>
> 上式中， π 为确定性策略的函数，其为状态空间 S 到动作空间 A 的映射。

随机性策略，是使用关于各个状态的动作概率分布函数来建模的策略。

> 即 $a \sim \pi(\cdot|s)$ ， $\pi : \mathcal{A} \times S \to [0, 1]$
>
> 假如把动作空间 A 的事件集合记为 $\mathcal{A}$
>
> 上式中， π 为随机性策略的函数，是动作空间的事件集合 $\mathcal{A}$ 与状态空间 S 的直和，到 $[0, 1]$ 的映射。

本文将介绍和对比强化学习中，随机性策略与确定性策略这两种策略建模的方法及其相关性质。

这两种策略在强化学习中，都得到了广泛的使用。使用随机性策略的 Actor-Critic 类算法，有例如 TRPO [1]、PPO [2] 、SAC [3]。而确定性策略的 Actor-Critic 类算法则有包括DPG [4]、DDPG [5]，等等。

策略梯度定理

随机性策略梯度与确定性策略的梯度的训练目标，是最大化该策略下的累计回报的期望，于是经过推导，可以获得两种类型的策略梯度定理的表达式：

随机性策略梯度定理 (Stochastic Policy Gradient Theorem)：

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

确定性策略梯度定理 (Deterministic Policy Gradient Theorem)：

$$\nabla_\theta \ J(\theta) = \mathbb{E}_{s\sim\rho^{\mu_\theta}}[\nabla_\theta \mu_\theta(s) \ \nabla_a \ Q(s,a)|a = \mu_\theta(s)]$$

> 这里需要额外注意的是，不同于随机性策略梯度定理，确定性策略梯度定理的严格成立，需要满足以下函数连续性条件：

- 该环境的马尔可夫决策过程的转移概率函数，$p(s'|s,a)$，及其关于动作的导数，$\nabla_a p(s'|s,a)$，都连续。
- 确定性策略的建模函数，$\mu_\theta(s)$，和该建模函数对模型参数的导数，$\nabla_\theta \mu_\theta(s)$，都连续。
- 该环境的奖励函数，$r(s,a)$，及其关于动作的导数，$\nabla_a r(s,a)$，都连续。
- 该环境的初始状态分布，$p_1(s)$，连续。

在使用策略梯度定理时，我们无法知晓真实的动作价值函数。因此需要使用最新的动作价值函数的模型，Q_ϕ，来近似真实的动作价值函数的数值，Q。动作价值函数模型的训练方法是使用某种贝尔曼算子获得动作价值函数的估计，Q_{target}，来拟合纠正当前价值模型的动作价值函数的数值，Q_ϕ。

$$\min \ \|Q_\phi - Q_{\text{target}}\|$$

原理与机制

对于随机性策略，算法流程包括下述步骤：执行采样，获得动作，并与环境交互产生回报奖励，并让策略的参数调整为累计奖励最大的方向。

从随机性策略梯度的表达式上看，其本质是一个累计回报关于随机策略函数的优化算法。根据 log-derivative trick，计算一个函数的期望的导数，等价于计算该函数与对应概率分布的Score Function的乘积的导数的期望。

而对于确定性策略，由于采用了确定性的函数来建模策略，在每个状态下，策略将决定一个确定性的动作，而不需要进行动作空间内的采样。因此，累计奖励的数值将不需要对该动作空间进行积分，这一点可以从确定性策略梯度的表达式上直观可见。这是两种策略梯度定理在原理和机制上的核心区别。

随机性策略梯度与确定性策略梯度的关联

随机性策略与确定性策略的梯度定理从公式上看，似乎相关性很小，不过草蛇灰线这里其实隐藏着一个精妙的关联。简单的来说，就是当随机性策略的方差为零时，确定性策略的梯度恰好为此时随机性策略的梯度。严格的论述如下：

假设随机性策略使用了一种，可以通过修改参数来逼近 δ 分布的某种概率分布函数进行建模，记为 v_σ，并由参数 σ 来控制方差的大小和逼近的程度：

$$\lim_{\sigma \to 0} v_\sigma = \delta$$

$$\pi_{\mu_\theta,\sigma}(a|s) = v_\sigma(\mu_\theta(s), a)$$

该随机性策略，v_σ，还需要满足以下条件：

- 可以逼近 δ 分布，并可以具备 δ 分布对任意连续函数的卷积特性。
- 对于任意动作 a' 做中心点的概率分布函数 $v_\sigma(a', \cdot)$，它的支撑集合需要具有Lipschitz边界，该函数需要在边界处趋于无，且在支撑集合上可微分。

- 对于任意动作 a' 做中心点的概率分布函数 $v_\sigma(a',\cdot)$，对于任意动作 a ， $\nabla_{a'}\, v_\sigma(a',a)$ 都存在。
- v_σ 需要具有平移不变性，即 $v_\sigma(a',a)=v_\sigma(a'+\delta,a+\delta)$

则当 v_σ 逼近某个 δ 分布的时候，随机性策略的梯度恰好等于某一确定性策略的梯度，即：

$$\lim_{\sigma\to 0}\ \nabla_\theta\ J_{\pi_{\mu_\theta,\sigma}}(\theta)=\nabla_\theta\ J_{\mu_\theta}(\theta)$$

同样的，此处确定性策略梯度定理的存在，需要满足前述的函数连续性条件，以及一些额外的有界性条件：

- 该环境的马尔可夫决策过程的转移概率函数， $p(s'|s,a)$ ，及其关于动作的导数， $\nabla_a p(s'|s,a)$ ，都连续且有界。
- 确定性策略的建模函数， $\mu_\theta(s)$ ，和该建模函数对模型参数的导数， $\nabla_\theta\mu_\theta(s)$ ，都连续。
- 该环境的奖励函数， $r(s,a)$ ，及其关于动作的导数， $\nabla_a r(s,a)$ ，都连续且有界。
- 该环境的初始状态分布， $p_1(s)$ ，连续且有界。

详细证明可见于论文《Deterministic Policy Gradient Algorithms》[4]。

随机性策略与确定性策略的比较

在具体实践中，随机策略梯度和确定性策略梯度之间存在着一些至关重要的区别，有以下几点：

1、成立条件

从上述随机性策略与确定性策略的梯度定理的成立条件可以发现，确定性策略梯度要严格成立所需要的建模条件和环境条件会更为严苛。

2、高维表现

随机性策略的理想梯度需要对状态空间和动作空间进行求积分计算期望，而确定性策略的梯度仅对状态空间进行积分计算期望。因此，在使用蒙特卡洛法进行实际计算求解时，准确计算随机策略梯度的大小，可能需要更多样本。尤其是当动作空间是高维的情况下，确定性策略往往会表现得更好。

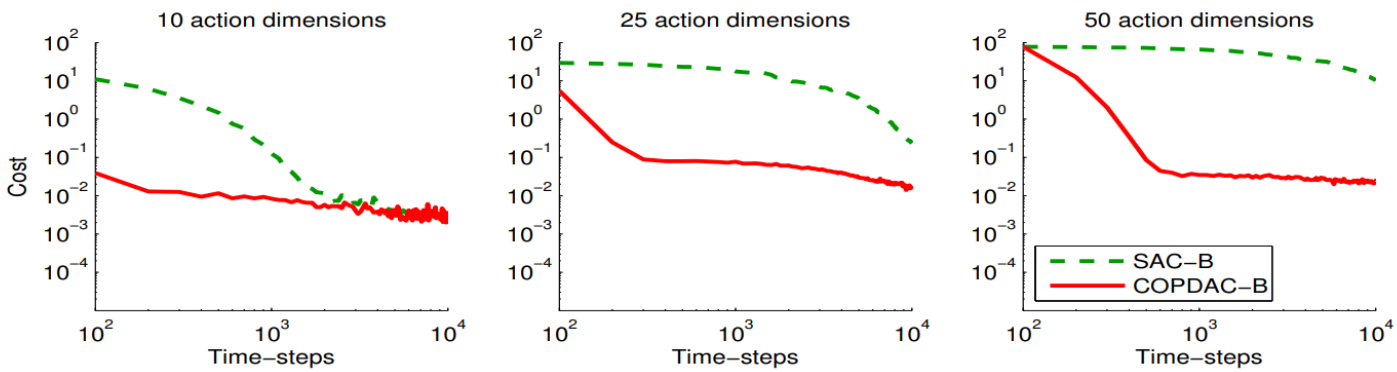


图1：随机性策略算法 (SAC) 与确定性策略算法 (COPDAC) 在不同动作维度的 continuous bandit 任务中的表现对比

3、探索程度

随机性策略在原理上会更有益于探索完整的状态和行动空间。确定性策略没有这种优势，所以需要额外一些措施来保证探索的多样性和完整性，这需要引入off-policy 数据，比如使用来自多个其它策略的数据 ，或是给确定性策略的动作生成过程引入噪声等的方式来丰富数据来源。

4、收敛现象

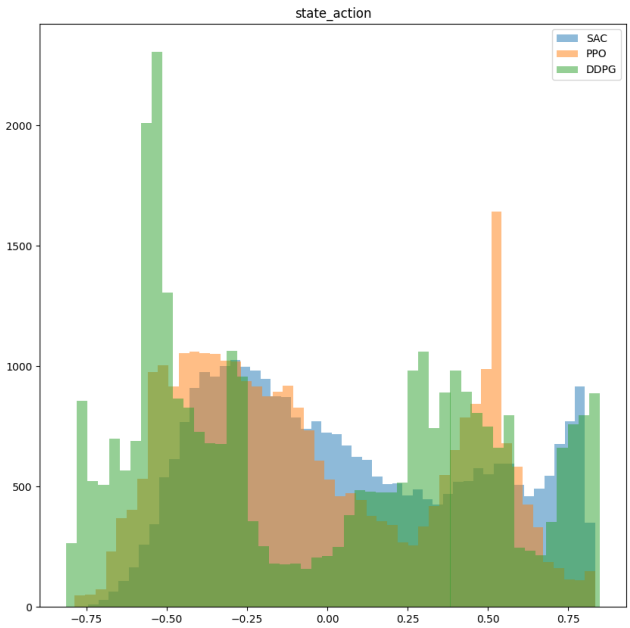
使用随机性策略梯度算法，随着训练中的算法模型逐渐寻优找到一个好的策略，随机性策略会逐渐变得更具确定性。

但这会使得随机性策略梯度， $\nabla_\theta \pi_\theta(a|s)$ ，在均值附近变化得更为剧烈，使得该随机性策略梯度的数值更难估计和计算。比如，假如使用高斯分布， $\mathcal{N}(\mu, \sigma^2)$ ，来建模一个随机性策略，其策略梯度的方差将大致与 $1/\sigma^2$ 成正比例关系[6]，随着随机性策略会逐渐变得更确定， σ^2 变小，策略梯度的方差将迅速变大。

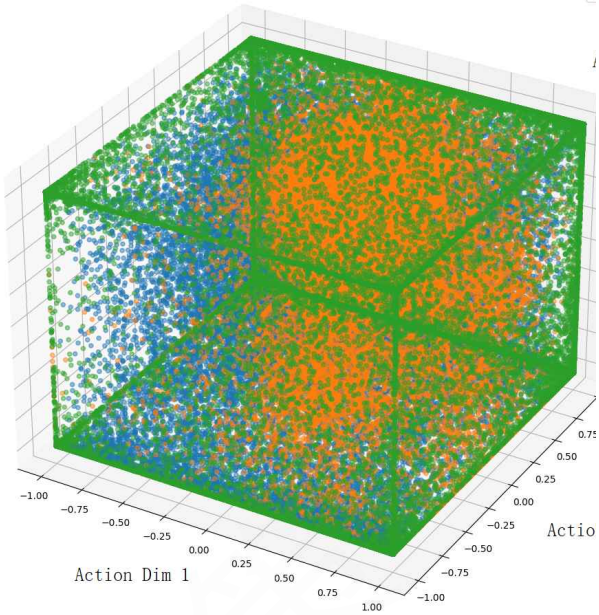
确定性策略会在这一方面表现更为稳定一些。

5、适用场景

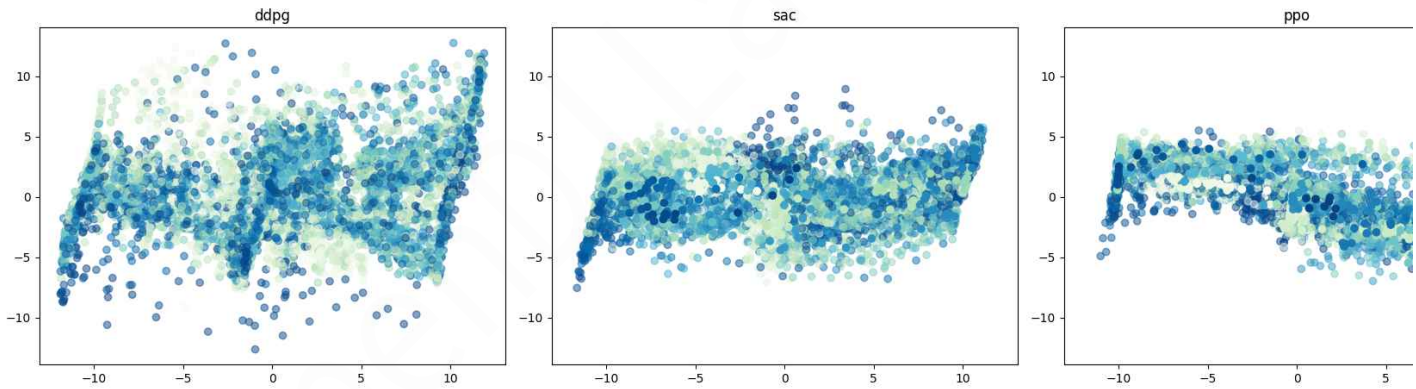
另外，在很多应用场景中，使用确定性策略来部署模型是有意义的。这会得到更稳定和确定性的输出，也更容易定位故障和复现问题。比如让一个机器人的机械臂按照指定输出来执行一个标准动作。而对于随机性策略，如果也需要在部署中，去输出一个确定性的输出，往往我们会使用随机性策略的动作分布的特定数值点，比如均值点，作为确定性输出的值。



相同 episode return 的RL专家智能体，降维后的“状态-动作”分布直方图



相同 episode return 的RL专家智能体，三维连续动点图



相同 episode return 的RL专家智能体，状态空间通过PCA算法降维后的分布图
（图中，颜色从浅至深，代表时刻变化，从起始至终止）

图2：环境：MuJoCo Hopper 算法：DDPG/SAC/PPO

相反，一般来说，仅使用确定性策略来生成数据，将无法确保动作空间和状态空间被充分的探索，并由此可能导致获得一个次优的解决方案。当然，如果环境中有足够噪声以确保充分探索，那么此时即使使用确定性策略，训练也可以生效并获得最优结果。

参考文献

[1] Schulman J, Levine S, Abbeel P, et al. Trust region policy optimization[C]//International conference on machine learning. PMLR, 2015: 1889-1897.

[2] Schulman J, Wolski F, Dhariwal P, et al. Proximal policy optimization algorithms[J]. arXiv preprint arXiv:1707.06347, 2017.

[3] Haarnoja T, Zhou A, Abbeel P, et al. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor[C]//International conference on machine learning. PMLR, 2018: 1861-1870.

[4] Silver D, Lever G, Heess N, et al. Deterministic policy gradient algorithms[C]//International conference on machine learning. Pmlr, 2014: 387-395.

[5] Lillicrap T P, Hunt J J, Pritzel A, et al. Continuous control with deep reinforcement learning[J]. arXiv preprint arXiv:1509.02971, 2015.

[6] Zhao T, Hachiya H, Niu G, et al. Analysis and improvement of policy gradient estimation[J]. Advances in Neural Information Processing Systems, 2011, 24.