Actor Critic Methods

Easwar Subramanian

TCS Innovation Labs, Hyderabad

Email : easwar.subramanian@tcs.com / cs5500.2020@iith.ac.in

October 14, 2022

Overview

1. Review
2. Variance Reduction Techniques
3. Towards Actor-Critic Formulation

Review

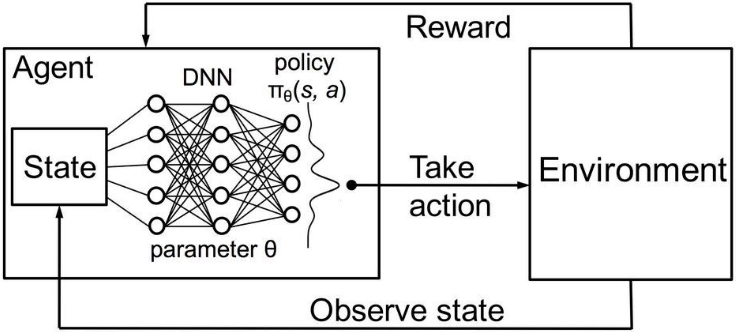
Policy Based Reinforcement Learning

- We will directly parametrize the policy

$$\pi_\theta(a|s) = P(a|s, \theta)$$

- We will consider model free control with parametrized policies
  - ★ With state-value functions Q, computing $\arg\max$ over actions gets tricky when action space is large or continuous
  - ★ Better convergence properties
  - ★ Can learn stochastic policies

Policy Using Function Approximators



- If action space is discrete
  - ★ Network could output a vector of probabilities (softmax)
- If action space is continuous
  - ★ Network could output the parameters of a distribution (For e.g.. mean and variance of a Gaussian)

Policy Optimization

A policy $\pi(\cdot)$ is parametrized by parameter θ and denoted by π_θ

Performance of a policy π_θ is given by

$$J(\theta) = V^{\pi_\theta}(s) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t r_{t+1} | s_0 = s\right]$$

Goal of RL is to find a policy

$$\pi_\theta^* = \arg\max_{\pi_\theta} V^{\pi_\theta}(s) = \arg\max_{\pi_\theta} \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t r_{t+1} | s_0 = s\right]$$

We will look for π_θ^* in class of stochastic policies by finding θ that maximizes $J(\theta)$

Policy Gradient

- Let $J(\theta)$ be the 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

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

- $\nabla_\theta J(\theta)$ is the policy gradient and
- α is the step size parameter

Policy Gradient Formulation

- Objective function $J(\theta)$ for policy gradient approach is written as,

$$J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t r_{t+1} | \pi_\theta\right] = \sum_{\tau \sim \pi_\theta} \left[P(\tau;\theta) G(\tau)\right]$$

where

$$G(\tau) = \sum_{t=0}^{\infty} \gamma^t r_{t+1}$$

with $\tau \sim \pi_\theta$ denoting the state-action sequence given by $s_0, a_0, s_1, a_1, \cdots, s_t, a_t, \cdots$

- Goal is to find θ^* such that

$$\theta^* = \arg\max_\theta J(\theta)$$

Policy Gradient Estimate

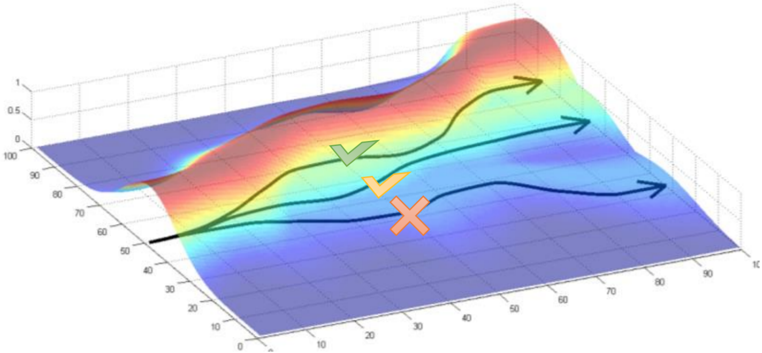
- Gradient derivation yields the following estimate

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[\nabla_\theta \log P(\tau; \theta) G(\tau) \right]$$

- Sample based estimate is given by

$$\nabla_\theta J(\theta) \approx \frac{1}{K} \sum_{i=1}^{K} \nabla_\theta \log P(\tau^{(i)}; \theta) G(\tau^{(i)})$$

Policy Gradient : Intuition



- Increase the probability of paths with positive $G(\tau)$
- Decrease the probability of paths with negative $G(\tau)$
- Formalize the notion of 'trial and error'

Slide Credit: Pieter Abeel : Deep RL Boot Camp

Policy Gradient : Model Free Formulation

Model free formulation of the policy gradient is given by

$$\nabla_\theta J(\theta) \approx \frac{1}{K} \sum_{i=1}^{K} \left[\sum_{t=0}^{\infty} \nabla_\theta \log \pi(a_t^{(i)} | s_t^{(i)}) \right] \left[\sum_{t=0}^{\infty} \gamma^t r_{t+1}^{(i)} \right]$$

REINFORCE : Monte-Carlo based Policy Gradient

Algorithm REINFORCE : MC based Policy Gradient

1: Initialize policy network π with parameters θ_1 and learning rate α
2: **for** $n = 1$ to N **do**
3: Sample K trajectories from π_{θ_n}
4: Calculate gradient estimate

$$\nabla_{\theta_n} J(\theta_n) \approx \frac{1}{K} \sum_{i=1}^{K} \left[\sum_{t=0}^{\infty} \nabla_{\theta_n} \log \pi(a_t^{(i)} | s_t^{(i)}) \right] \left[\sum_{t=0}^{\infty} \gamma^t r_{t+1}^{(i)} \right]$$

5: Perform gradient update

$$\theta_{n+1} = \theta_n + \alpha \nabla_{\theta_n} J(\theta_n)$$

6: **end for**

Issues with Gradient Estimate

- The gradient estimate, thus calculated, is unbiased but has high variance (reason : we are sampling stochastic paths)
- Hence the gradient descent is slow to converge
- Some variance reduction techniques are required in practice

Variance Reduction Techniques

Discount Factor and Variance Reduction

Gradient estimate is given by,

$$\nabla_\theta J(\theta) \approx \frac{1}{K}\sum_{i=1}^{K}\left[\sum_{t=0}^{\infty}\nabla_\theta \log \pi(a_t^{(i)}|s_t^{(i)})\right]\left[\sum_{t=0}^{\infty}\gamma^t r_{t+1}^{(i)}\right]$$

One can rewrite the above equation as

$$\nabla_\theta J(\theta) \approx \frac{1}{K}\sum_{i=1}^{K}\left[\sum_{t=0}^{\infty}\nabla_\theta \log \pi(a_t^{(i)}|s_t^{(i)}) \cdot \left[\sum_{k=0}^{\infty}\gamma^k r_{k+1}^{(i)}\right]\right]$$

- For infinite horizon MDPs having $\gamma < 1$ not only helps in proving convergence of algorithms but also helps reduce variance of the policy gradient estimate
- Ignoring reward terms 'far' into the future gives us a reasonable approximation to policy gradient but with lower variance

Aside : Score Function

Score function in policy gradient is the term

$$\nabla_\theta \log \pi(a_t|s_t)$$

Expectation of the score function is zero

$$\begin{aligned}
\mathbb{E}_{a_t|s_t}\left[\nabla_\theta \log \pi(a_t|s_t)\right] &= \int_{a_t} \pi(a_t|s_t)\nabla_\theta \log \pi(a_t|s_t)\, da_t \\
&= \int_{a_t} \nabla_\theta \pi(a_t|s_t)\, da_t \\
&= \nabla_\theta \int_{a_t} \pi(a_t|s_t)\, da_t \\
&= \nabla_\theta 1 = 0
\end{aligned}$$

Principle of Causality

Causality : Policy at time t' cannot affect reward at time t when $t < t'$.

- When we take an action at timestep t, it can only affect the rewards from timesteps t and onwards.

Recall that,

$$\nabla_\theta J(\theta) \approx \frac{1}{K} \sum_{i=1}^{K} \left[\sum_{t=0}^{\infty} \nabla_\theta \log \pi(a_t^{(i)} | s_t^{(i)}) \cdot \left[\sum_{k=0}^{\infty} \gamma^k r_{k+1}^{(i)} \right] \right]$$

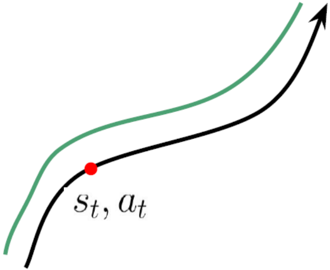
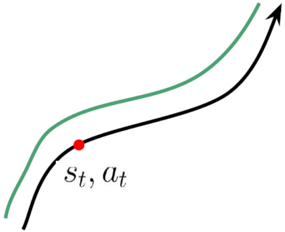


Figure Source: Jie-Han-Chen:SlideShare

Principle of Causality



$$G(\tau) = \sum_{t=0}^{\infty} \gamma^t r_{t+1}$$

Let $\tau_{a:b}$ denote the states and actions visited from time a to b and

$$G_{a:b}(\tau) = \sum_{t=a}^{b} \gamma^t r_{t+1}$$

Therefore for any time t, we have,

$$G(\tau) = G_{0:t-1}(\tau) + G_{t:\infty}(\tau)$$

Figure Source: Jie-Han-Chen:SlideShare

Temporal Structure

$$
\begin{aligned}
\nabla_\theta J(\theta) &= \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{\infty}\nabla_\theta \log \pi(a_t|s_t)\cdot\left[\sum_{k=0}^{\infty}\gamma^k r_{k+1}\right]\right] \\
&= \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{\infty}\nabla_\theta \log \pi(a_t|s_t) G(\tau)\right] \\
&= \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{\infty}\nabla_\theta \log \pi(a_t|s_t) G_{0:t-1}(\tau) + \sum_{t=0}^{\infty}\nabla_\theta \log \pi(a_t|s_t) G_{t:\infty}(\tau)\right] \\
&= \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{\infty}\nabla_\theta \log \pi(a_t|s_t) G_{0:t-1}(\tau)\right] \\
&\quad + \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{\infty}\nabla_\theta \log \pi(a_t|s_t) G_{t:\infty}(\tau)\right]
\end{aligned}
$$

Temporal Structure

Consider evaluating the expectation of the first term

$$\mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{\infty}\nabla_\theta \log \pi(a_t|s_t) G_{0:t-1}(\tau)\right] = \left[\sum_{t=0}^{\infty} G_{0:t-1}(\tau)\, \mathbb{E}_{\pi_\theta} \nabla_\theta \log \pi(a_t|s_t)\right]$$

$$= \sum_{t=0}^{\infty} G_{0:t-1} \cdot 0 = 0$$

Therefore, the policy gradient estimate with temporal structure is given by

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{\infty}\nabla_\theta \log \pi(a_t|s_t) G_{t:\infty}(\tau)\right]$$

Temporal Structure

The sample estimate of the gradient expression is given by

$$\nabla_\theta J(\theta) \approx \frac{1}{K}\sum_{i=1}^{K}\left[\sum_{t=0}^{\infty}\nabla_\theta \log \pi(a_t^{(i)}|s_t^{(i)}) \cdot \left[\sum_{t'=t}^{\infty}\gamma^{t'-t} r_{t'+1}^{(i)}\right]\right]$$

- The above policy gradient estimate with temporal structure is also an unbiased estimate of the true policy gradient but has **lower variance** since it has '*thrown out*' a few terms

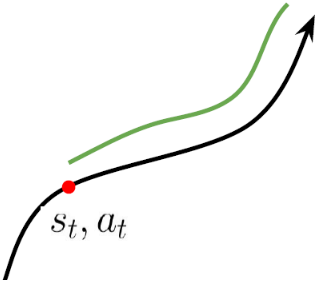
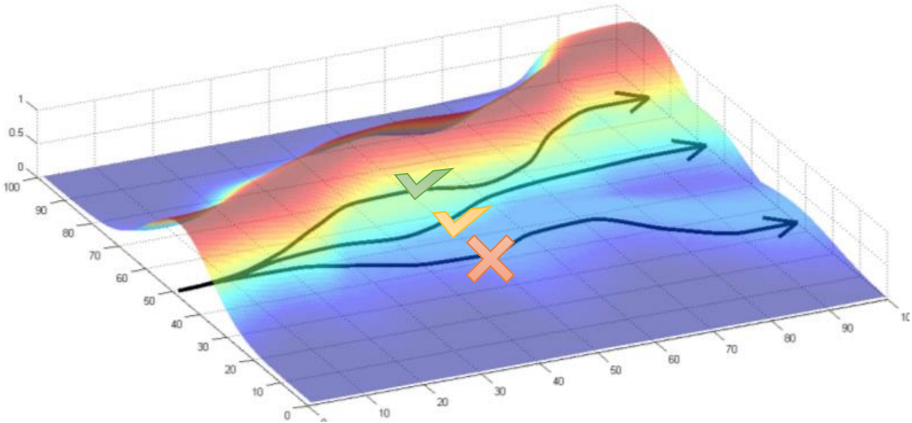


Figure Source: Jie-Han-Chen:SlideShare

Need for a Baseline



What if all paths have positive reward sum ?

Baseline

Can we subtract a baseline without biasing the gradient ?

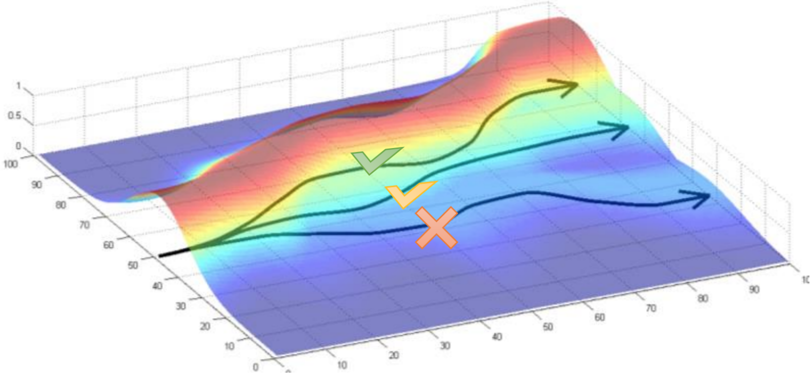
Let $b(s_t)$ be a baseline that is conditioned on s_t. Then,

$$\mathbb{E}_{a_t|s_t}\left[b(s_t)\nabla_\theta \log \pi(a_t|s_t)\right] = b(s_t)\mathbb{E}_{a_t|s_t}\left[\nabla_\theta \log \pi(a_t|s_t)\right] = 0$$

Therefore,

$$\begin{aligned}
\nabla_\theta J(\theta) &= \mathbb{E}_{\tau\sim\pi_\theta}\left[\nabla_\theta \log \pi(a_t|s_t)\cdot G_{t:\infty}(\tau)\right] \\
&= \mathbb{E}_{\tau\sim\pi_\theta}\left[\nabla_\theta \log \pi(a_t|s_t)\cdot G_{t:\infty}(\tau)\right] - \mathbb{E}_{\tau\sim\pi_\theta}\left[b(s_t)\nabla_\theta \log \pi(a_t|s_t)\right] \\
&= \mathbb{E}_{\tau\sim\pi_\theta}\left[\nabla_\theta \log \pi(a_t|s_t)\cdot \left[G_{t:\infty}(\tau) - b(s_t)\right]\right]
\end{aligned}$$

Need for a Baseline



A good choice for baseline :

$$b = \mathbb{E}(G(\tau)) \approx \frac{1}{K}\sum_{i=1}^{K} G(\tau^{(i)})$$

Slide Credit: Pieter Abeel : Deep RL Boot Camp

Popular choices of Baseline

- Constant Baseline

$$b = \mathbb{E}(G(\tau)) \approx \frac{1}{K}\sum_{i=1}^{K} G(\tau^{(i)})$$

- Time Dependent Baseline

$$b_t = \frac{1}{K}\sum_{i=1}^{K} G_{t:\infty}(\tau^{(i)})$$

- Optimal Baseline

$$b = \frac{\mathbb{E}_\tau(\nabla_\theta \log \pi(a_t|s_t)^2 G_{t:\infty}(\tau))}{\mathbb{E}_\tau(\nabla_\theta \log \pi(a_t|s_t)^2)}$$

- State dependent expected return

$$b(s) = \mathbb{E}_{\pi_\theta}[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots |s_t = s] = V^\pi(s)$$

State dependent expected return

$$b(s) = \mathbb{E}_{\pi_\theta}[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots | s_t = s] = V^\pi(s)$$

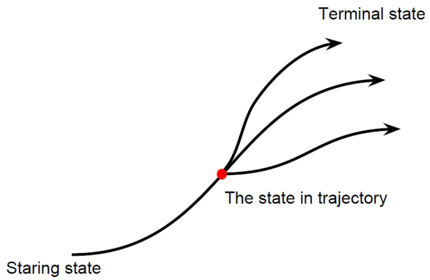


Figure Source:
Jie-Han-Chen:SlideShare

Vanilla Policy Gradient Algorithm

Algorithm Vanilla Policy Gradient Algorithm

1: Initialize policy network π with parameters θ_1 learning rate α and baseline b
2: **for** $n = 1$ to N **do**
3: Sample K trajectories by executing the policy π_{θ_n}
4: At each time step of each trajectory compute $G_t = \sum_{t'=t}^{\infty} \gamma^{t'-t} r_{t+1}$ and advantage estimate $A_t = G_t - b(s_t)$
5: Calculate gradient estimate

$$\nabla_{\theta_n} J(\theta_n) \approx \frac{1}{K} \sum_{i=1}^{K} \left[\sum_{t=0}^{\infty} \nabla_{\theta_n} \log \pi(a_t^{(i)} | s_t^{(i)}) A_t \right]$$

6: Perform gradient update

$$\theta_{n+1} = \theta_n + \alpha \nabla_{\theta_n} J(\theta_n)$$

7: **end for**

Improvements to Vanilla Policy Gradient

- The REINFORCE and Vanilla policy gradient as described above is on-policy
  - ★ There is an off-policy way to do policy gradient algorithms
- We do learning by Monte-Carlo roll-outs
  - ★ Will be addressed by Actor-Critic method

Towards Actor-Critic Formulation

Temporal Structure and Actor-Critic Algorithms

The policy gradient estimate with temporal structure (takes causality into account) is given by

$$\nabla_\theta J(\theta) \quad = \quad \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{\infty} \nabla_\theta \log \pi(a_t | s_t) G_{t:\infty}(\tau) \right]$$

where

$$G_{a:b}(\tau) = \sum_{t=a}^{b} \gamma^t r_{t+1}$$

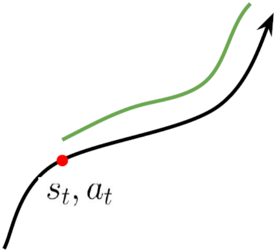
Sample estimate is given by

$$\nabla_\theta J(\theta) \approx \frac{1}{K} \sum_{i=1}^{K} \left[\sum_{t=0}^{\infty} \nabla_\theta \log \pi(a_t^{(i)} | s_t^{(i)}) \cdot \left[\sum_{t'=t}^{\infty} \gamma^{t'-t} r_{t'+1}^{(i)} \right] \right]$$

- This gradient estimate is the starting point of actor-critic algorithms

Temporal Structure and Actor-Critic Algorithms

$$\nabla_\theta J(\theta) \approx \frac{1}{K} \sum_{i=1}^{K} \left[\sum_{t=0}^{\infty} \nabla_\theta \log \pi(a_t^{(i)} | s_t^{(i)}) \cdot \left[\sum_{t'=t}^{\infty} \gamma^{t'-t} r_{t'+1}^{(i)} \right] \right]$$

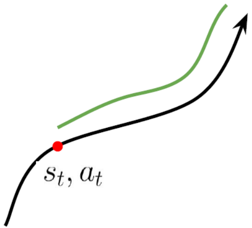


The green curve represents the entity in the inner summation

Figure Source: Jie-Han-Chen:SlideShare

The Critic

$$\nabla_\theta J(\theta) \approx \frac{1}{K} \sum_{i=1}^{K} \left[\sum_{t=0}^{\infty} \nabla_\theta \log \pi(a_t^{(i)} | s_t^{(i)}) \cdot \left[\sum_{t'=t}^{\infty} \gamma^{t'-t} r_{t'+1}^{(i)} \right] \right]$$

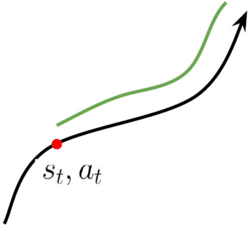


The inner summation is an estimate of $Q^{\pi_\theta}(s_t, a_t)$!!

Figure Source: Jie-Han-Chen:SlideShare

The Critic

$$\nabla_\theta J(\theta) \approx \frac{1}{K}\sum_{i=1}^{K}\left[\sum_{t=0}^{\infty}\nabla_\theta \log \pi(a_t^{(i)}|s_t^{(i)}) \cdot \left[\sum_{t'=t}^{\infty}\gamma^{t'-t} r_{t'+1}^{(i)}\right]\right]$$



The inner summation is an estimate of $Q(s_t, a_t)$ and it gives an estimate of how 'good' the action a_t was in state s_t (and hence the name '**critic**')

Figure Source: UCB : Sergey Levine

Policy Gradient Theorem

$$
\begin{aligned}
\nabla_\theta J(\theta) &= \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{\infty}\{\nabla_\theta \log \pi_\theta(a_t|s_t) G_{t:\infty}(\tau)\}\right] \\
&= \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{\infty}\mathbb{E}_{\tau\sim\pi_\theta}\left(\{\nabla_\theta \log \pi_\theta(a_t|s_t) G_{t:\infty}(\tau)\}\Big|\, s_t, a_t\right)\right] \\
&= \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{\infty}\left\{\nabla_\theta \log \pi_\theta(a_t|s_t)\, \underbrace{\mathbb{E}_{\tau\sim\pi_\theta}\left(G_{t:\infty}(\tau)\Big|\, s_t, a_t\right)}_{??}\right\}\right]
\end{aligned}
$$

Policy Gradient Theorem

For suitable objective function $J(\theta)$, we have,

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty}\left\{\nabla_\theta \log \underbrace{\pi_\theta(a_t|s_t)}_{\text{Actor}}\underbrace{Q^{\pi_\theta}(s_t, a_t)}_{\text{Critic}}\right\}\right]$$

- Replacaes the single path reward by $Q^{\pi_\theta}(s_t, a_t)$
- Policy gradient theorem applies to start state objective, average reward objective and average value objective

(More on this later !!)