Lecture 20

Actor-Critic Methods

In section 18.1, we discussed the likelihood policy gradient given as

$$\frac{\partial}{\partial\theta}\mathbb{E}\Bigg[\sum_{k=0}^{N} g(x_k,u_k)\Bigg] = \mathbb{E}\Bigg[\Bigg(\sum_{k=0}^{N} g(x_k,u_k)\Bigg)\Bigg(\frac{\partial}{\partial\theta}\sum_{k=0}^{N}\log p_\theta(u_k|x_k)\Bigg)\Bigg] \tag{20.1}$$

$$= \mathbb{E}\Bigg[\Bigg(\sum_{k=0}^{N} g(x_k,u_k)\Bigg)\Bigg(\sum_{k=0}^{N}\frac{\partial}{\partial\theta}\log p_\theta(u_k|x_k)\Bigg)\Bigg] \tag{20.2}$$

$$= \mathbb{E}\Bigg[\sum_{k=0}^{N}\Bigg(g(x_k,u_k)\sum_{n=0}^{k}\frac{\partial}{\partial\theta}\log p_\theta(u_n|x_n)\Bigg)\Bigg] \tag{20.3}$$

$$\tag{20.4}$$

since

$$\mathbb{E}\Bigg[g(x_k,u_k)\frac{\partial}{\partial\theta}\log p_\theta(u_n|x_n)\Bigg] = 0 \tag{20.5}$$

for all $n > k$. We can also write

$$\frac{\partial}{\partial\theta}\mathbb{E}\Bigg[\sum_{k=0}^{N} g(x_k,u_k)\Bigg] = \mathbb{E}\Bigg[\sum_{k=0}^{N}\Bigg(\frac{\partial}{\partial\theta}\log p_\theta(u_k|x_k)\sum_{n=k}^{N} g(x_n,u_n)\Bigg)\Bigg]. \tag{20.6}$$

We previously discussed how policy gradient approaches can perform poorly due to a large variance in the estimated gradient. One way to improve the variance is to use a "baseline". We showed this in the weight-perturbation case. Consider a baseline added to the policy gradient update

$$\frac{\partial}{\partial\theta}\mathbb{E}\Bigg[\sum_{k=0}^{N} g(x_k,u_k)\Bigg] = \mathbb{E}\Bigg[\sum_{k=0}^{N}\Bigg(\frac{\partial}{\partial\theta}\log p_\theta(u_k|x_k)\Bigg(\sum_{n=k}^{N} g(x_n,u_n) - b(x_k)\Bigg)\Bigg)\Bigg] \tag{20.7}$$

$$= \mathbb{E}\Bigg[\sum_{k=0}^{N}\Bigg(\frac{\partial}{\partial\theta}\log p_\theta(u_k|x_k)\sum_{n=k}^{N} g(x_n,u_n)\Bigg)\Bigg] - \mathbb{E}\Bigg[\sum_{k=0}^{N}\Bigg(\frac{\partial}{\partial\theta}\log p_\theta(u_k|x_k)b(x_k)\Bigg)\Bigg] \tag{20.8}$$

If we only focus on the baseline, we have:

$$
\begin{aligned}
&= \mathbb{E}\Big[\sum_{k=0}^{N}\Big(\frac{\partial}{\partial\theta}\log p_\theta(u_k|x_k)b(x_k)\Big)\Big] && (20.9)\\
&= \int\sum_{k=0}^{N}\Big(\frac{\partial}{\partial\theta}\log p_\theta(u_k|x_k)b(x_k)\Big)p_\theta(u_k|x_k)du_k && (20.10)\\
&= \int\sum_{k=0}^{N}\Big(\frac{\partial}{\partial\theta}p_\theta(u_k|x_k)b(x_k)\Big)du_k && (20.11)\\
&= \sum_{k=0}^{N}b(x_k)\frac{\partial}{\partial\theta}\int p_\theta(u_k|x_k)du_k && (20.12)\\
&= 0 && (20.13)
\end{aligned}
$$

We can observe that adding a baseline still permits a unbiased estimate of the gradients so long as the baseline does not depend on θ.

Now let's revisit the policy gradient update

$$
\frac{\partial}{\partial\theta}\mathbb{E}\Big[\sum_{k=0}^{N}g(x_k,u_k)\Big] = \mathbb{E}\Big[\sum_{k=0}^{N}\Big(\frac{\partial}{\partial\theta}\log p_\theta(u_k|x_k)\sum_{n=k}^{N}g(x_n,u_n)\Big)\Big]. \tag{20.14}
$$

We can observe that

$$
\sum_{n=k}^{N}g(x_n,u_n) = Q(x_k,u_k) \tag{20.15}
$$

and

$$
\frac{\partial}{\partial\theta}\mathbb{E}\Big[\sum_{k=0}^{N}g(x_k,u_k)\Big] = \mathbb{E}\Big[\sum_{k=0}^{N}\Big(\frac{\partial}{\partial\theta}\log p_\theta(u_k|x_k)Q(x_k,u_k)\Big)\Big]. \tag{20.16}
$$

Furthermore, we can observe that if we choose our baseline to be the cost-to-go, we have

$$
\begin{aligned}
\frac{\partial}{\partial\theta}\mathbb{E}\Big[\sum_{k=0}^{N}g(x_k,u_k)\Big] &= \mathbb{E}\Big[\sum_{k=0}^{N}\Big(\frac{\partial}{\partial\theta}\log p_\theta(u_k|x_k)(Q(x_k,u_k)-J(x_k))\Big)\Big] && (20.17)\\
&= \mathbb{E}\Big[\sum_{k=0}^{N}\Big(\frac{\partial}{\partial\theta}\log p_\theta(u_k|x_k)A(x_k,u_k)\Big)\Big]. && (20.18)
\end{aligned}
$$

where $A(x_k,u_k) = Q(x_k,u_k) - J(x_k)$ is known as the advantage.

Note, that we can also write the policy gradient update for the infinite-horizon case as

$$
\frac{\partial}{\partial\theta}\mathbb{E}[J(x)] = \mathbb{E}\Big[\Big(\frac{\partial}{\partial\theta}\log p_\theta(u|x)A(x,u)\Big)\Big]. \tag{20.19}
$$

20.1. Trust Region Policy Optimization

Trust Region Policy Optimization (TRPO) is an actor critic approach that attempts to limit the size of the policy update, similar to guided policy search.

The TRPO update is given as

$$
\begin{aligned}
\theta_{k+1} &= \arg\max \mathcal{L}(\theta_k,\theta) && (20.20)\\
&\quad s.t. \quad D_{KL}(\theta||\theta_k) \le \delta && (20.21)
\end{aligned}
$$

where $\mathcal{L}(\theta_k, \theta)$ is a surrogate advantage function

$$\mathcal{L}(\theta_k, \theta) = \mathbb{E}_{s,a\sim\pi_{\theta_k}}\left[\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)} A^{\pi_{\theta_k}}(s,a)\right] \tag{20.22}$$

and

$$D_{KL}(\theta||\theta_k) = \mathbb{E}_{s\sim\pi_{\theta_k}}\left[D_{KL}(\pi_\theta(a|s)||\pi_{\theta_k}(a|s))\right] \tag{20.23}$$

Instead of using an analytical solution, TRPO uses an approximate solution by taking the Taylor series expansion

$$\mathcal{L} \approx g^T(\theta - \theta_k) \tag{20.24}$$

$$D_{KL}(\theta||\theta_k) \approx \frac{1}{2}(\theta - \theta_k)^T H(\theta - \theta_k) \tag{20.25}$$

g just happens to be the policy gradient update. The approximate problem can be solved

$$\theta_{k+1} = \theta_k + \alpha^j \sqrt{\frac{2\delta}{g^T H^{-1} g}} H^{-1} g \tag{20.26}$$

where α^j is used for a backtracking line-search. The conjugate gradient algorithm is used to solve for H.

The value function is found by doing regression on the costs.

20.2. Proximal Policy Optimization

Proximal Policy Optimization (PPO) tries to achieve the same thing as TRPO.

PPO updates policies

$$\theta_{k+1} = \arg\max_\theta \mathbb{E}_{s,a\sim\pi_{\theta_k}}\left[L(s, a, \theta_k, \theta)\right] \tag{20.27}$$

where

$$L(s, a, \theta_k, \theta) = \min\left(\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)} A^{\pi_{\theta_k}}(s,a), g(\epsilon, A^{\pi_{\theta_k}(s,a)})\right) \tag{20.28}$$

$$\tag{20.29}$$

and

$$g(\epsilon, A) = \begin{cases} (1+\epsilon)A & A \geq 0 \\ (1-\epsilon)A & A < 0 \end{cases} \tag{20.30}$$

20.3. Deep Deterministic Policy Gradient

Deep Deterministic Policy Gradient (DDPG) is very similar to the Deep Q-learning example we explored in class, except we leverage policy gradient updates.

- $y(r, s', d) = r + \gamma(1-d)Q_{\phi_t}(s', \mu_\theta(s'))$
- $\nabla_\phi \frac{1}{|B|}\sum(Q_\phi(s,a) - y(r, s', d))^2$
- $\nabla_\theta \frac{1}{|B|}\sum Q_\phi(s, \mu_\theta(s))$
- update target networks

Bibliography

[1] Dimitri Bertsekas. *Reinforcement Learning and Optimal Control*. Athena Scientific, 2019.

[2] Russ Tedrake. *Underactuated Robotics*. 2023.

[3] John Schulman. Trust region policy optimization. *arXiv preprint arXiv:1502.05477*, 2015.

[4] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[5] TP Lillicrap. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

© Joseph Moore, 2025