

From Policy Gradient to Actor-Critic methods

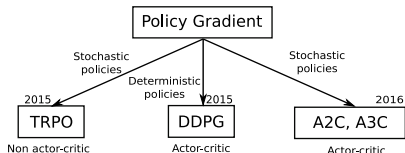
Advantage Actor Critic

Olivier Sigaud

Sorbonne Université
<http://people.isir.upmc.fr/sigaud>



Advantage Actor Critic (A2C)



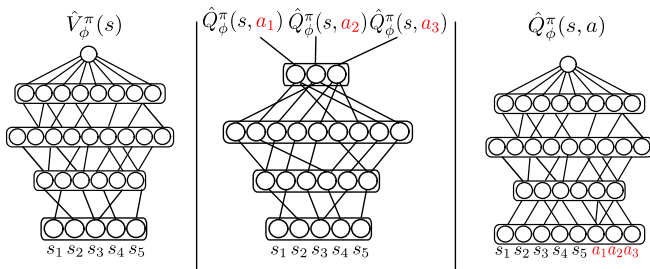
- ▶ A crucial move from Policy Gradient methods to Actor-Critic methods
- ▶ The earliest actor-critic algorithm of the deep RL era using stochastic policies
- ▶ It directly derives from the basic Policy Gradient method
- ▶ The critic is learned using bootstrap, which makes it an actor-critic algorithm
- ▶ The A2C paper focuses more on A3C, an asynchronous version where several agents generate data without using a replay buffer
- ▶ A2C can be seen as a simplified version of A3C with a single agent



Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. (2016) Asynchronous methods for deep reinforcement learning. *arXiv preprint arXiv:1602.01783*

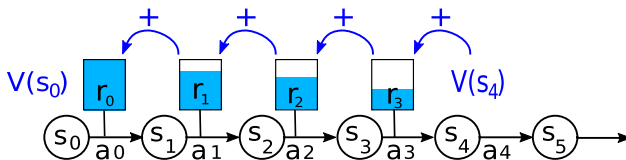
Main distinguishing features

- ▶ To perform policy gradient, you need the advantage function
- ▶ Computes the advantage function as value function minus the return of the current N-step trajectory
- ▶ Adds entropy regularization to favor exploration in the gradient calculation step
- ▶ Uses N-step updates
- ▶ Does not use a replay buffer
- ▶ Note that A2C is Actor-Critic, but on-policy, so **one cannot equate Actor-Critic and off-policy**
- ▶ But adding a replay buffer and making it more off-policy would be straightforward

Choice of a V critic

- ▶ Main point: By contrast with $Q(s, a)$, $V(s)$ can be estimated in the same way irrespective of using discrete or continuous actions
- ▶ \hat{V}_ϕ^π is smaller, but not necessarily easier to estimate (implicit max over actions)
- ▶ Temporal difference error: $\delta = [r(\mathbf{s}_t) + \gamma V_\phi^i(\mathbf{s}_{t+1}) - V_\phi^i(\mathbf{s}_t)]$
- ▶ Standard update rule: $V_\phi^{i+1}(\mathbf{s}_t) \leftarrow V_\phi^i(\mathbf{s}_t) + \alpha \delta$

Advantage function calculation



- To perform policy gradient updates, one needs to compute $\hat{A}_\phi(s_t, a_t)$
- By definition, $A(s_t, a_t) = Q(s_t, a_t) - V(s_t)$
- A2C computes the advantage with $\hat{A}_\phi(s_t, a_t) = R_t(s_t) - V_\phi(s_t)$
- $R_t(s_t) = \sum_{i=0}^{N-1} \gamma^i r_{t+i} + \gamma^N V_\phi(s_{t+N})$ is the return of the current N-step trajectory from state s_t
- $R_t(s_t)$ can be seen as an approximate of $Q(s_t, a_t)$ computed along one trajectory

Policy Gradient updates

- ▶ The standard Policy Gradient update is:

$$\nabla_{\theta} J(\theta) = \mathbb{E}_{\mathbf{s}_t, \mathbf{a}_t \sim \pi_{\theta}(\cdot)} [\nabla_{\theta} \log \pi_{\theta}(\mathbf{a}_t^{(i)} | \mathbf{s}_t^{(i)})] \hat{A}_{\phi}(\mathbf{s}_t, \mathbf{a}_t)$$

- ▶ But to favor exploration, A2C adds an entropy term to the gradient calculation
- ▶ Thus the policy update rule is:

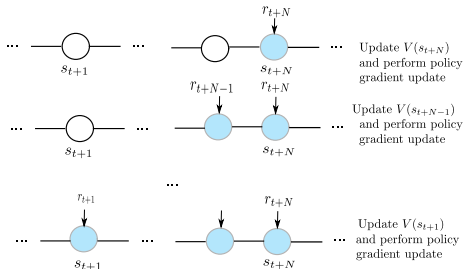
$$\nabla_{\theta} J(\theta) = \mathbb{E}_{\mathbf{s}_t, \mathbf{a}_t \sim \pi_{\theta}(\cdot)} [\nabla_{\theta} [\log \pi_{\theta}(\mathbf{a}_t^{(i)} | \mathbf{s}_t^{(i)}) (R_t - V_{\phi}(\mathbf{s}_t)) - \beta \mathcal{H}(\pi_{\theta}(\mathbf{s}_t))]]$$

- ▶ where $\mathcal{H}(\pi_{\theta}(\mathbf{s}_t))$ is the entropy of policy π_{θ} at state \mathbf{s}_t .
- ▶ Note that A2C adds entropy in the update of the actor, but outside the critic, whereas SAC adds it in the critic target, which has a deeper impact.



Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A. Abbeel, P. et al. (2018) Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*

N-step updates



- ▶ The agent performs N steps in the environment (or less if the episode stops earlier in the episodic case) before each update
- ▶ At each update, the agent has collected up to N states and rewards
- ▶ It can update the value of the last state using the last reward, the value of the second last step with two rewards
- ▶ And so on up to the first state of the current collection
- ▶ It updates both the critic and the policy at each update

Any question?



Send mail to: Olivier.Sigaud@upmc.fr



Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, P., et al. (2018).

Soft actor-critic algorithms and applications.

arXiv preprint arXiv:1812.05905.



Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., Silver, D., and Kavukcuoglu, K. (2016).

Asynchronous methods for deep reinforcement learning.

In Balcan, M. and Weinberger, K. Q., editors, *Proceedings of the 33rd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1928–1937. JMLR.org.