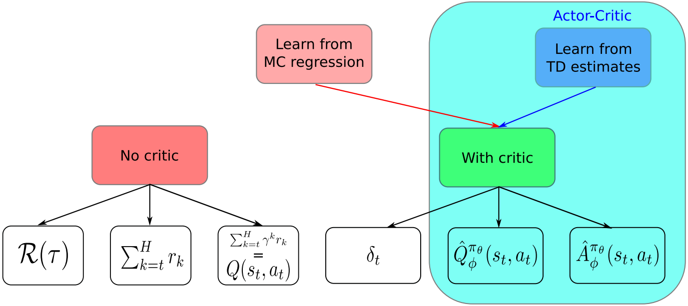
From Policy Gradient to Actor-Critic methods

PG with baseline versus Actor-Critic

Olivier Sigaud

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

Being truly actor-critic



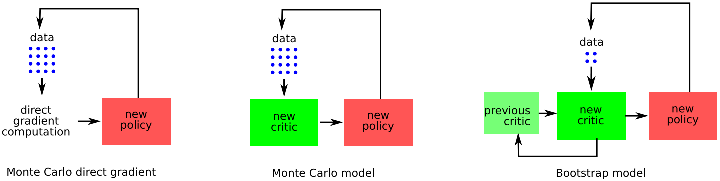
- PG methods with V, Q or A baselines contain a policy and a critic
- Are they actor-critic?
- Only if the critic is learned from bootstrap!

Being Actor-Critic

- "Although the REINFORCE-with-baseline method learns both a policy and a state-value function, we do not consider it to be an actor–critic method because its state-value function is used only as a baseline, not as a critic."
- "That is, it is not used for bootstrapping (updating the value estimate for a state from the estimated values of subsequent states), but only as a baseline for the state whose estimate is being updated."
- "This is a useful distinction, for only through bootstrapping do we introduce bias and an asymptotic dependence on the quality of the function approximation."

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction (Second edition)*. MIT Press, 2018, p. 331

Monte Carlo versus Bootstrap approaches



- Three options:
  - MC direct gradient: Compute the true $Q^{\pi_{\boldsymbol{\theta}}}$ over each trajectory
  - MC model: Compute a model $\hat{Q}^{\pi_{\boldsymbol{\theta}}}_{\boldsymbol{\phi}}$ over rollouts using MC regression, throw it away after each policy gradient step
  - Bootstrap: Update a model $\hat{Q}^{\pi_{\boldsymbol{\theta}}}_{\boldsymbol{\phi}}$ over samples using TD methods, keep it over policy gradient steps
  - Sutton&Barto: Only the latter ensures "asymptotic convergence" (when stable)

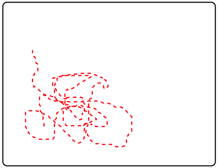
Single step updates

- With a model $\psi_t(s_t^{(i)}, a_t^{(i)})$, we can compute $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$ over a single state using:

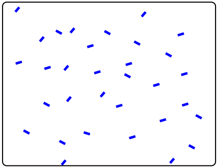
$$\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a_t^{(i)}|s_t^{(i)}) \psi_t(s_t^{(i)}, a_t^{(i)})$$

- With $\psi_t = \hat{Q}_{\boldsymbol{\phi}}^{\pi_{\boldsymbol{\theta}}}(s_t^{(i)}, a_t^{(i)})$ or $\psi_t = \hat{A}_{\boldsymbol{\phi}}^{\pi_{\boldsymbol{\theta}}}(s_t^{(i)}, a_t^{(i)})$
- This is true whatever the way to obtain $\hat{Q}_{\boldsymbol{\phi}}^{\pi_{\boldsymbol{\theta}}}$ or $\hat{A}_{\boldsymbol{\phi}}^{\pi_{\boldsymbol{\theta}}}$
- Crucially, samples used to update $\hat{Q}_{\boldsymbol{\phi}}^{\pi_{\boldsymbol{\theta}}}$ or $\hat{A}_{\boldsymbol{\phi}}^{\pi_{\boldsymbol{\theta}}}$ do not need to be the same as samples used to compute $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$

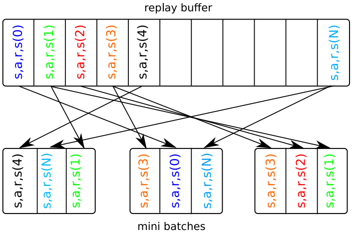
Using a replay buffer



Non i.i.d. samples



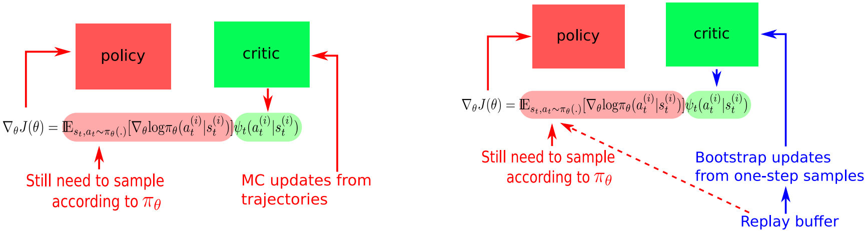
i.i.d. samples



- Agent samples are not independent and identically distributed (i.i.d.)
- Shuffling a replay buffer (RB) makes them more i.i.d.
- It improves a lot the sample efficiency
- Recent data in the RB come from policies close to the current one

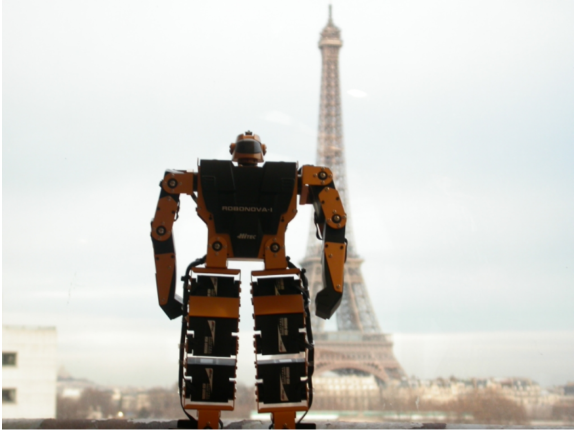
Lin, L.-J. (1992) Self-Improving Reactive Agents based on Reinforcement Learning, Planning and Teaching. *Machine Learning*, 8(3/4), 293–321

Bootstrap properties



- If $\hat{Q}^{\pi_{\boldsymbol{\theta}}}_{\boldsymbol{\phi}}$ is obtained from bootstrap, everything can be done from a single sample
- Samples to compute $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$ still need to come from $\pi_{\boldsymbol{\theta}}$
- Samples to update the critic do not need this anymore
- This defines the shift from policy gradient to actor-critic
- This is the crucial step to become off-policy
- However, using bootstrap comes with a bias
- Next lesson: bias-variance trade-off

Any question?



Send mail to: Olivier.Sigaud@upmc.fr

Long-Jin Lin.
Self-Improving Reactive Agents based on Reinforcement Learning, Planning and Teaching.
Machine Learning, 8(3/4):293–321, 1992.

Richard S. Sutton and Andrew G. Barto.
Reinforcement Learning: An Introduction (Second edition).
MIT Press, 2018.