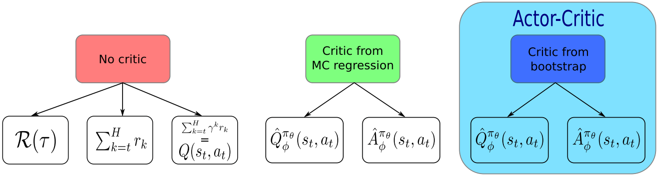
# From Policy Gradient to Actor-Critic methods

## PG with baseline versus Actor-Critic

Olivier Sigaud

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

## Being truly actor-critic



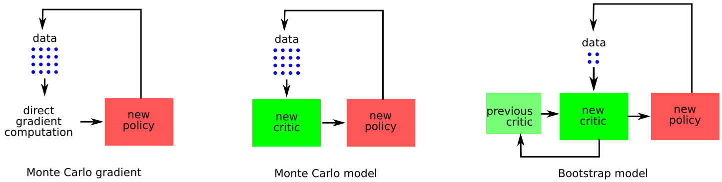

- Policy gradient methods with $V$, $Q$ or $A$ baselines contain a policy and a critic
- Are they actor-critic?
- The answer: only with bootstrap

## 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."

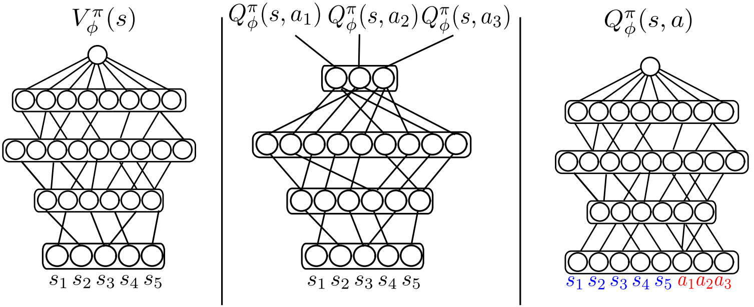
Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction (Second edition)*. MIT Press, 2018, p. 331

## Monte Carlo versus Bootstrap approaches



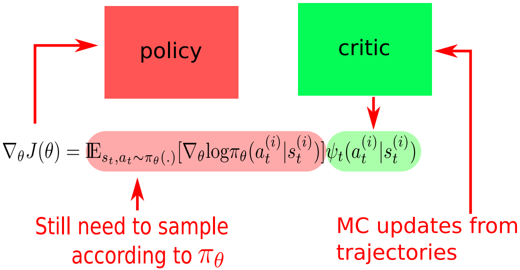

- Three options:
  - MC gradient: Compute the true $Q^{\pi_\theta}(s_t^{(i)}, a_t^{(i)})$ over each trajectory
  - MC model: Compute a model $\hat{Q}_\phi^{\pi_\theta}$ over a set of trajectories, using Monte Carlo + regression, throw it away after each policy gradient step
  - Bootstrap: Update a model $\hat{Q}_\phi^{\pi_\theta}$ over a set of trajectories, using TD methods, keep it over policy gradient steps

## Practical implementation of neural critics




- $\hat{V}^\pi_\phi$ is smaller, but not necessarily easier to estimate
- Given the implicit max in $\hat{V}^\pi_\phi(s)$, approximation may be less stable than $\hat{Q}^\pi_\phi(s)$ (?)
- Note: a critic network provides a value even in unseen states
- Sutton&Barto: with bootstrap, asymptotic convergence of the critics (when stable)

## Bootstrap properties (1)

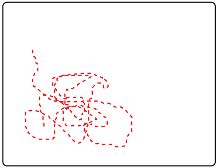



- With a model $\hat{Q}_\phi(s_t^{(i)}, a_t^{(i)})$, we can compute the gradient over a single state using:

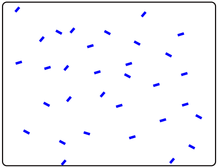
$$\nabla_\theta \log \pi_\theta(a_t^{(i)}|s_t^{(i)})\hat{Q}_\phi(s_t^{(i)}, a_t^{(i)})$$

- This is true even if $\hat{Q}_\phi^{\pi_\theta}$ is obtained from Monte Carlo

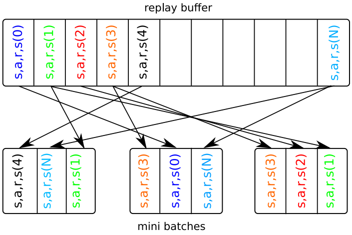
## Using a replay buffer



Non i.i.d. samples



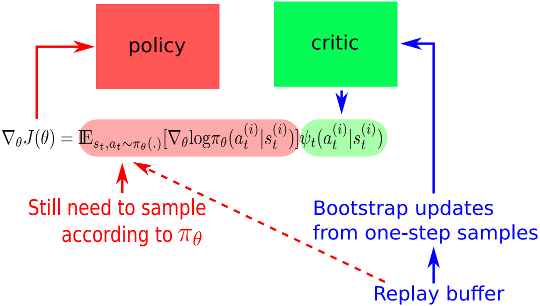
i.i.d. samples




- 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

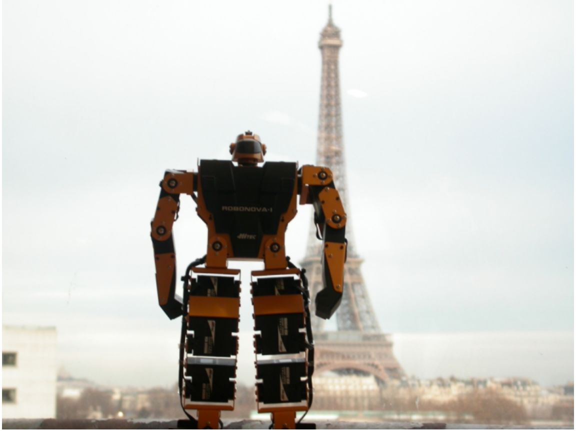
Lin, L.-J. (1992) Self-Improving Reactive Agents based on Reinforcement Learning, Planning and Teaching. *Machine Learning*, 8(3/4), 293–321

## Bootstrap properties (2)




- If $\hat{Q}_\phi^{\pi_\theta}$ is obtained from bootstrap, everything can be done with the current step
- Samples to update the critic do not need to be the same as to update the actor
- This defines the shift from policy gradient to actor-critic
- This is the crucial step to become off-policy
- However, using the replay buffer 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.