From MCTS to AlphaZero

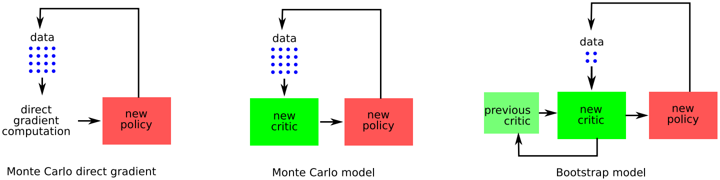
Olivier Sigaud

Sorbonne Université
http://www.isir.upmc.fr/personnel/sigaud

Background

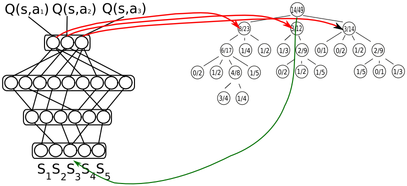
- MCTS plays well Go or chess, but is quite inefficient
- Areas for improvement:
  - Avoid forgetting $Q(s, a)$ after each step
  - Avoid using Monte Carlo simulations again each time to evaluate $Q(s, a)$
  - Instead of running random simulations, play good moves with higher probabilities
- Adding a critic network solves these issues

Actor-Critic vs Monte Carlo



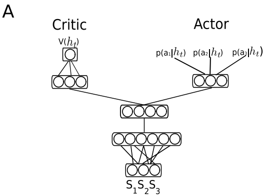
- Monte Carlo direct gradient: Estimate $Q(s,a)$ over rollouts
- Monte Carlo model: learn a model $\hat{Q}(s,a)$ over rollouts using MC regression, throw it away after each update
- Bootstrap: Update a model $\hat{Q}(s,a)$ over samples using TD methods, keep it over policy gradient steps
- The bootstrap approach is much more sample efficient
- It introduces bias and reduces variance

MCTS + Critic



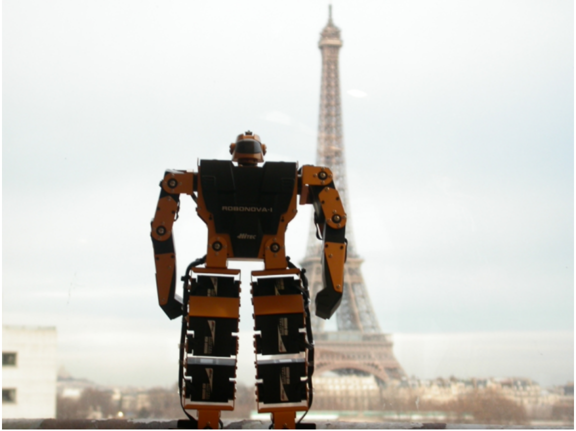
- Learns a critic $\hat{Q}(s,a)$ for all states over all rollouts
- Using a DQN-like architecture
- Still builds a plan with an MPC-like approach, not using $\max_a \hat{Q}(s,a)$ as policy
- The MCTS search process helps balancing samples, favors exploration
- In AlphaZero:
  - Instead of playing random rollouts, can play rollouts driven by $\hat{Q}(s,a)$
  - The critic $\hat{Q}(s,a)$ can be pre-trained with expert moves (AlphaGo vs AlphaZero)

AlphaZero: from DQN-like to actor-critic



- Learning a policy and a $\hat{V}(s)$ function is more efficient than using a $\hat{Q}(s, a)$ function

Any question?



Send mail to: `Olivier.Sigaud@upmc.fr`