University of Waterloo

The Option-Critic Architecture

Author: Pierre-Luc Bacon, Jean Harb, Doina Precup
Speaker: Zebin KANG

June 26, 2018

Content

Background
Research Problem

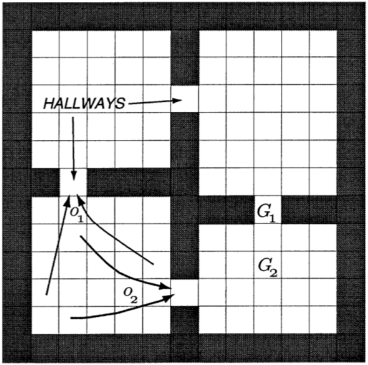

Figure 1: Finding subgoals in four-room domain and learning policies to achieve these subgoals

- $\mathcal{S}$: a set of states
- $\mathcal{A}$: a set of actions
- P: a transition function, mapping $\mathcal{S} \times \mathcal{A}$ to $\mathcal{S} \rightarrow [0, 1]$
- r: a reward function, mapping $\mathcal{S} \times \mathcal{A}$ to $\mathbb{R}$
- π: a policy, the probability distribution over actions conditioned on states, i.e. $\pi : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$
- $V_\pi(s) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r_{t+1} | s_0 = s]$: the **value function** of a policy π
- $Q_\pi(s, a) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r_{t+1} | s_0 = s, a_0 = a]$: the **action-value function** of a policy π
- $\rho(\theta, s_0) = \mathbb{E}_{\pi_\theta}[\sum_{t=0}^{\infty} \gamma^t r_{t+1} | s_0]$: the **discounted return** with respect a specific start state s_0

Policy Gradient Theorem [2]

Uses stochastic gradient descent to optimize a performance objective over a given family of parametrized stochastic policies π_θ:

$$\frac{\partial \rho(\theta, s_0)}{\partial \theta} = \sum_s \mu_{\pi_\theta}(s|s_0) \sum_a \frac{\partial \pi_\theta(a|s)}{\partial \theta} Q_{\pi_\theta}(s, a)$$

where $\mu_{\pi_\theta}(s|s_0) = \sum_{t=0}^{\infty} \gamma^t P(s_t = s|s_0, \pi)$ is a discounted weighting of state along the trajectories starting from s_0 and $Q_{\pi_\theta}(s, a) = \mathbb{E}\{\sum_{k=1}^{\infty} \gamma^{k-1} r_{t+k}|s_t = s, a_t = a, \pi\}$ is the action-value given a policy.

a Markovian option: $\omega = (\mathcal{I}_\omega, \pi_\omega, \beta_\omega)$

- Ω: the set of all histories and $\omega \in \Omega$
- $\mathcal{I}_\omega$: an initiation set and $\mathcal{I}_\omega \subset S$
- π_ω: an intra-option policy, mapping $S \times A$ to $[0, 1]$
- β_ω: a termination function, mapping S to $[0, 1]$
- $\pi_{\omega,\theta}$: an intra-option policy of ω parametrized by θ
- $\beta_{\omega,\vartheta}$: a termination function of ω parametrized by ϑ

Option-value Function can be defined as:

$$Q_\Omega(s,\omega) = \sum_a \pi_{\omega,\theta}(a|s) Q_U(s,\omega,a)$$

where Q_U is the option-action-value function

$$Q_U(s,\omega,a) = r(s,a) + \gamma \sum_{s'} P(s'|s,a) U(\omega,s')$$

The function U is the **option-value function upon arrival**:

$$U(\omega,s') = (1-\beta_{\omega_t,\vartheta}(s'))Q_\Omega(s',\omega) + \beta_{\omega_t,\vartheta}(s')V_\Omega(s')$$

Intra-Option Policy Gradient Theorem (Theorem 1)

Given a set of Markov options with stochastic intra-option policies differentiable in their parameters θ, the **gradient of the option-value function with respect to** θ and initial condition (s_0, ω_0):

$$\frac{\partial Q_\Omega(s_0, \omega_0)}{\partial \theta} = \sum_{s,\omega} \mu_\Omega(s, \omega | s_0, \omega_0) \sum_a \frac{\partial \pi_{\omega,\theta}(a|s)}{\partial \theta} Q_U(s, \omega, a)$$

where $\mu_\Omega(s, \omega | s_0, \omega_0)$ is a discounted weighting of state-option pairs along trajectories starting from (s_0, ω_0):

$$\mu_\Omega(s, \omega | s_0, \omega_0) = \sum_{t=0}^{\infty} \gamma^t P(s_t = s, \omega_t = \omega | s_0, \omega_0)$$

Termination Gradient Theorem (Theorem 2)

Given a set of Markov options with stochastic termination functions differentiable in their parameters ϑ, the **gradient of the option-value function upon arrival with respect to** ϑ and the initial condition (s_1, ω_0) is:

$$\frac{\partial U(\omega_0, s_1)}{\partial \vartheta} = -\sum_{s', \omega} \mu_\Omega(s', \omega | s_1, \omega_0) \frac{\partial \beta_{\omega,\vartheta}(s')}{\partial \vartheta} A_\Omega(s', \omega)$$

where $\mu_\Omega(s', \omega | s_1, \omega_0)$ is a discounted weighting of state-option pairs along trajectories from (s_1, ω_0):

$$\mu_\Omega(s', \omega | s_1, \omega_0) = \sum_{t=0}^{\infty} \gamma^t P(s_{t+1} = s', \omega_t = \omega | s_1, \omega_0)$$

and $A_\Omega(s', \omega) = Q_\Omega(s', \omega) - V_\Omega(s')$ is the advantage function [5].

Learning Options
Architecture and Algorithm

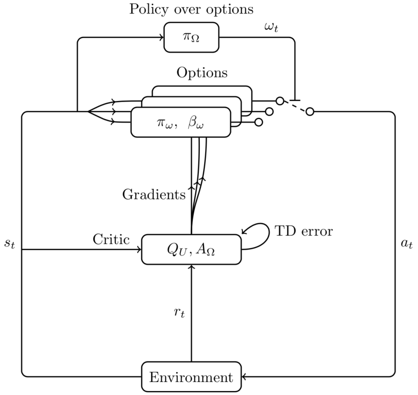

Figure 2: Diagram of the option-critic architecture

Algorithm 1: Option-critic with tabular intra-option Q-learning

---

$s \leftarrow s_0$
Choose ω according to an ϵ-soft policy over options $\pi_\Omega(s)$
repeat
  Choose a according to $\pi_{\omega,\theta}(a \mid s)$
  Take action a in s, observe s', r

  1. Options evaluation:
  $\delta \leftarrow r - Q_U(s, \omega, a)$
  if s' *is non-terminal* **then**
    $\delta \leftarrow \delta + \gamma(1 - \beta_{\omega,\vartheta}(s'))Q_\Omega(s', \omega) + \gamma\beta_{\omega,\vartheta}(s') \max_{\bar{\omega}} Q_\Omega(s', \bar{\omega})$
  end
  $Q_U(s, \omega, a) \leftarrow Q_U(s, \omega, a) + \alpha\delta$

  2. Options improvement:
  $\theta \leftarrow \theta + \alpha_\theta \frac{\partial \log \pi_{\omega,\theta}(a \mid s)}{\partial\theta} Q_U(s, \omega, a)$
  $\vartheta \leftarrow \vartheta - \alpha_\vartheta \frac{\partial \beta_{\omega,\vartheta}(s')}{\partial\vartheta} (Q_\Omega(s', \omega) - V_\Omega(s'))$

  if $\beta_{\omega,\vartheta}$ *terminates in* s' **then**
  choose new ω according to ϵ-soft$(\pi_\Omega(s'))$
  $s \leftarrow s'$
until s' *is terminal*

Experiments
Four-rooms Domains

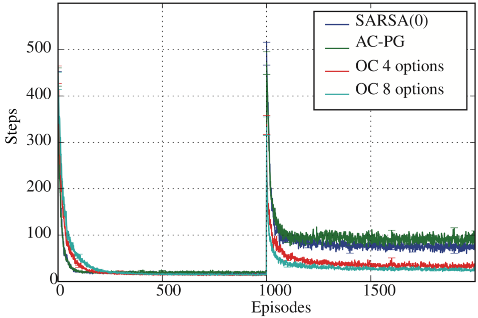

Figure 3: After a 1000 episodes, the goal location in the four-rooms domain is moved randomly. Option-critic ("OC") recovers faster than the primitive actor-critic ("AC-PG") and SARSA(0). Each line is averaged over 350 runs.

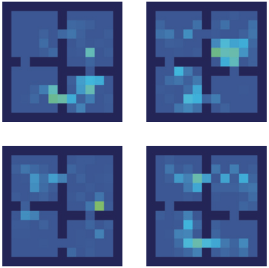

Figure 4: Termination probabilities for the option-critic agent learning with 4 options. The darkest color represents the walls in the environment while lighter colors encode higher termination probabilities.

Experiments
Pinball Domains

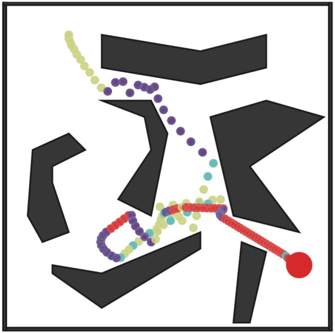

Figure 5: Pinball: Sample trajectory of the solution found after 250 episodes of training using 4 options All options (color-coded) are used by the policy over options in successful trajectories. The initial state is in the top left corner and the goal is in the bottom right one (red circle).

Experiments

Pinball Domains

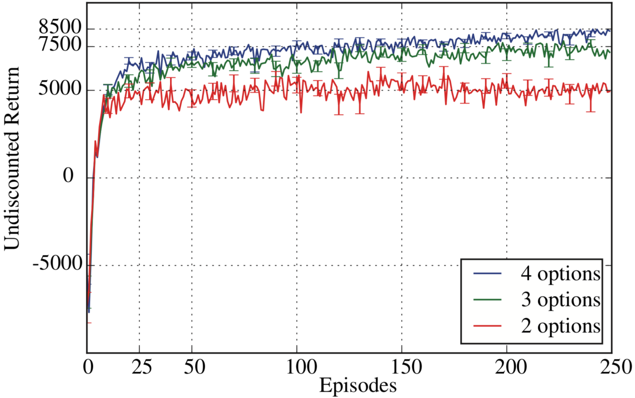

Figure 6: Learning curves in the Pinball domain.

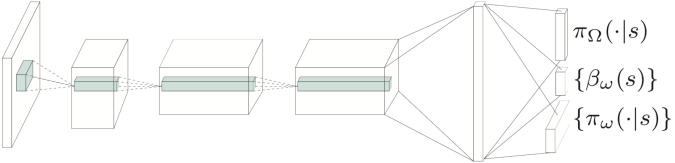

Figure 7: Extend deep neural network architecture [8]. A concatenation of the last 4 images is fed through the convolutional layers, producing a dense representation shared across intra-option policies, termination functions and policy over options.

Experiments

Arcade Learning Environment

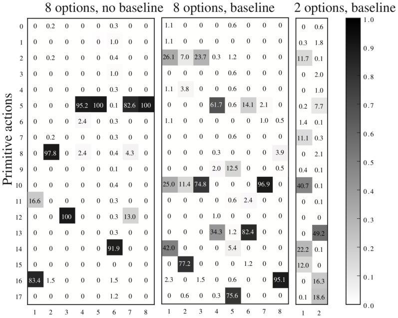

Figure 8: Seaquest: Using a baseline in the gradient estimators improves the distribution over actions in the intra-option policies, making them less deterministic. Each column represents one of the options learned in Seaquest. The vertical axis spans the 18 primitive actions of ALE. The empirical action frequencies are coded by intensity.

Experiments
Arcade Learning Environment

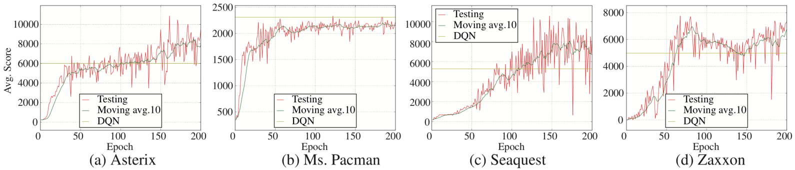

Figure 9: Learning curves in the Arcade Learning Environment. The same set of parameters was used across all four games: 8 options, 0.01 termination regularization, 0.01 entropy regularization, and a baseline for the intra-option policy gradients.

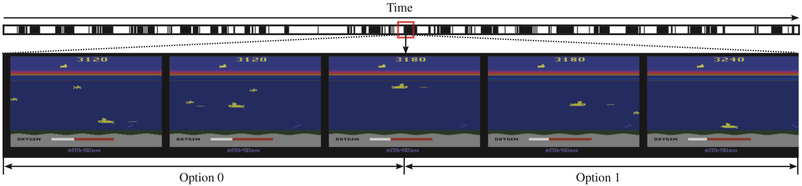

Figure 10: Up/down specialization in the solution found by option-critic when learning with 2 options in Seaquest. The top bar shows a trajectory in the game, with “white” representing a segment during which option 1 was active and “black” for option 2.

Conclusion

- Proves "Intra-Option Policy Gradient Theorem" and "Termination Gradient Theorem"
- Raises the option-critic architecture and algorithm
- Verifies the option-critic architecture with experiments in various domains

References

[1] Bacon, P. L., Harb, J., & Precup, D. (2017, February). The Option-Critic Architecture. In AAAI (pp. 1726-1734).

[2] Sutton, R. S., McAllester, D. A., Singh, S. P., & Mansour, Y. (2000). Policy gradient methods for reinforcement learning with function approximation. In Advances in neural information processing systems (pp. 1057-1063).

[3] Sutton, R. S., Precup, D., & Singh, S. (1999). Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. Artificial intelligence, 112(1-2), 181-211.

[4] Sutton, R. S. (1984). Temporal credit assignment in reinforcement learning.

[5] Baird III, L. C. (1993). Advantage updating (No. WL-TR-93-1146). WRIGHT LAB WRIGHT-PATTERSON AFB OH.

[6] Mann, T., Mankowitz, D., & Mannor, S. (2014, January). Time-regularized interrupting options (TRIO). In International Conference on Machine Learning (pp. 1350-1358).

[7] Konda, V. R., & Tsitsiklis, J. N. (2000). Actor-critic algorithms. In Advances in neural information processing systems (pp. 1008-1014).

[8] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing atari with deep reinforcement learning. arXiv