The Option-Critic Architecture

Pierre-Luc Bacon and **Jean Harb** and **Doina Precup**
Reasoning and Learning Lab, School of Computer Science
McGill University
{pbacon, jharb, dprecup}@cs.mcgill.ca

https://alversafa.github.io/blog/2018/11/28/optncrtc.html

Options Framework

1. The options framework is a hierarchical reinforcement learning framework that was introduced in (R. Sutton, D. Precup and S. Singh, 1999) and (D. Precup, 2000).

2. Options in this framework stands for higher level policies over lower level policies.

Abstract

Temporal abstraction is key to scaling up learning and planning in reinforcement learning. While planning with temporally extended actions is well understood, creating such abstractions autonomously from data has remained challenging. We tackle this problem in the framework of options [Sutton, Precup & Singh, 1999; Precup, 2000]. We derive policy gradient theorems for options and propose a new *option-critic* architecture capable of learning both the internal policies and the termination conditions of options, in tandem with the policy over options, and without the need to provide any additional rewards or subgoals. Experimental results in both discrete and continuous environments showcase the flexibility and efficiency of the framework.

Main

Algorithms and Architecture

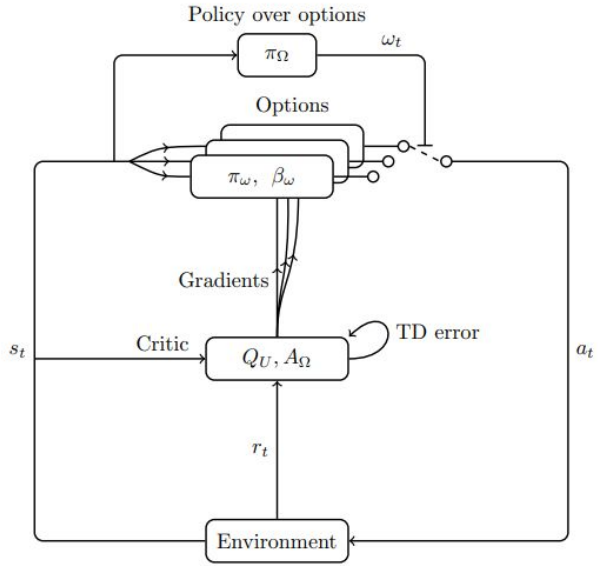


Figure 1: Diagram of the option-critic architecture. The option execution model is depicted by a *switch* $\perp$ over the *contacts* $-\circ$. A new option is selected according to π_Ω only when the current option terminates.

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
  $Q_U(s,\omega,a) \leftarrow Q_U(s,\omega,a) + \alpha\delta$ Critic update

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

  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*

Question: How to train policy over options (π_Ω) ??

Game Board:

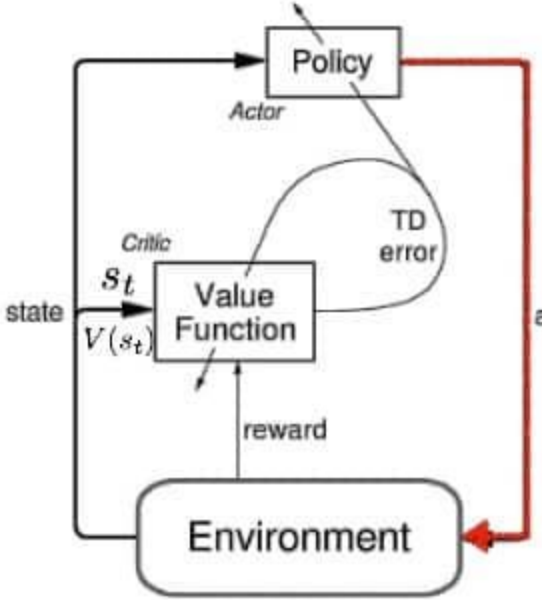
Current state (*s*): 0 0 0
0 1 0

Q Table: γ = 0.95

	0 0 0 1 0 0	0 0 0 0 1 0	0 0 0 0 0 1	1 0 0 0 0 0	0 1 0 0 0 0	0 0 1 0 0 0
↑	0.2	0.3	1.0	-0.22	-0.3	0.0
↓	-0.5	-0.4	-0.2	-0.04	-0.02	0.0
→	0.21	0.4	-0.3	0.5	1.0	0.0
←	-0.6	-0.1	-0.1	-0.31	-0.01	0.0

The intra-option policies, termination functions and policy over options belong to the actor part of the system while the critic consists of Qu and AΩ. The option-critic architecture does not prescribe how to obtain π_Ω since a variety of existing approaches would apply

Actor-Critic



- Actor: decides which action to take
- Critic: tells the actor how good its action was and how it should adjust

(Figure from Sutton & Barto, 1998)

Actor-critic methods consist of two models, which may optionally share parameters:

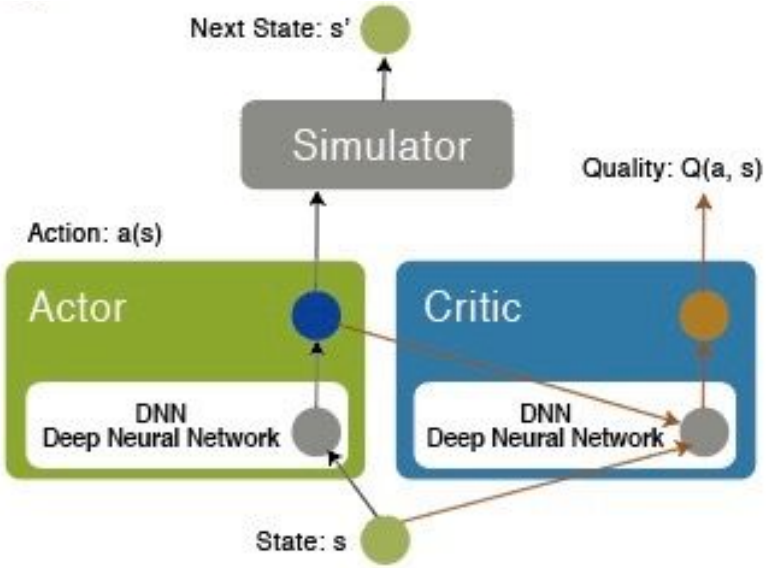
- **Critic** updates the value function parameters w and depending on the algorithm it could be action-value $Q_w(a|s)$ or state-value $V_w(s)$.
- **Actor** updates the policy parameters θ for $\pi_\theta(a|s)$, in the direction suggested by the critic.

Let's see how it works in a simple action-value actor-critic algorithm.

1. Initialize s, θ, w at random; sample $a \sim \pi_\theta(a|s)$.
2. For $t = 1 \dots T$:
    1. Sample reward $r_t \sim R(s, a)$ and next state $s' \sim P(s'|s, a)$;
    2. Then sample the next action $a' \sim \pi_\theta(a'|s')$;
    3. Update the policy parameters: $\theta \leftarrow \theta + \alpha_\theta Q_w(s, a) \nabla_\theta \ln \pi_\theta(a|s)$;
    4. Compute the correction (TD error) for action-value at time t:
    $\delta_t = r_t + \gamma Q_w(s', a') - Q_w(s, a)$
    and use it to update the parameters of action-value function:
    $w \leftarrow w + \alpha_w \delta_t \nabla_w Q_w(s, a)$
    5. Update $a \leftarrow a'$ and $s \leftarrow s'$.

Two learning rates, α_θ and α_w, are predefined for policy and value function parameter updates respectively.

DDPG(Deep Deterministic Policy Gradient)



Quick Facts

- DDPG is an off-policy algorithm.
- DDPG can only be used for environments with continuous action spaces.
- DDPG can be thought of as being deep Q-learning for continuous action spaces.

Preliminaries and Notation

A Markov Decision Process consists of a set of states $\mathcal{S}$, a set of actions $\mathcal{A}$, a transition function $\mathrm{P} : \mathcal{S} \times \mathcal{A} \to (\mathcal{S} \to [0, 1])$ and a reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. For convenience, we develop our ideas assuming discrete state and action sets. However, our results extend to continuous spaces using usual measure-theoretic assumptions (some of our empirical results are in continuous tasks). A (Markovian stationary) *policy* is a probability distribution over actions conditioned on states, $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$. In discounted problems, the value function of a policy π is defined as the expected return: $V_\pi(s) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r_{t+1} \mid s_0 = s\right]$ and its action-value function as $Q_\pi(s, a) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r_{t+1} \mid s_0 = s, a_0 = a\right]$, where $\gamma \in [0, 1)$ is the *discount factor*. A policy π is *greedy* with respect to a given action-value function Q if $\pi(s, a) > 0$ iff $a = \mathrm{argmax}_{a'} Q(s, a')$. In a discrete MDP, there is at least one optimal policy which is greedy with re-

The options framework (Sutton, Precup, and Singh 1999; Precup 2000) formalizes the idea of temporally extended actions. A Markovian option $\omega \in \Omega$ is a triple $(\mathcal{I}_\omega, \pi_\omega, \beta_\omega)$ in which $\mathcal{I}_\omega \subseteq \mathcal{S}$ is an initiation set, π_ω is an *intra-option* policy, and $\beta_\omega : \mathcal{S} \rightarrow [0, 1]$ is a termination function. We also assume that $\forall s \in \mathcal{S}, \forall \omega \in \Omega : s \in \mathcal{I}_\omega$ (i.e., all options are available everywhere), an assumption made in the majority of option discovery algorithms. We will discuss how to dispense with this assumption in the final section. (Sutton, Precup, and Singh 1999; Precup 2000) show that an MDP endowed with a set of options becomes a Semi-Markov Decision Process (Puterman 1994, chapter 11), which has a corresponding optimal value function over options $V_\Omega(s)$ and option-value function $Q_\Omega(s, \omega)$. Learning and planning algorithms for MDPs have their counterparts in this setting. However, the existence of the underlying MDP offers the possibility of learning about many different options in parallel : this is the idea of *intra-option learning*, which we leverage in our work.

Learning Options

We adopt a continual perspective on the problem of learning options. At any time, we would like to distill all of the available experience into every component of our system: value function and policy over options, intra-option policies and termination functions. To achieve this goal, we focus on learning option policies and termination functions, assuming they are represented using differentiable parameterized function approximators.

We consider the *call-and-return* option execution model, in which an agent picks option ω according to its policy over options π_Ω , then follows the intra-option policy π_ω until termination (as dictated by β_ω), at which point this procedure is repeated. Let $\pi_{\omega,\theta}$ denote the intra-option policy of option ω parametrized by θ and $\beta_{\omega,\vartheta}$, the termination function of

Algorithms and Architecture

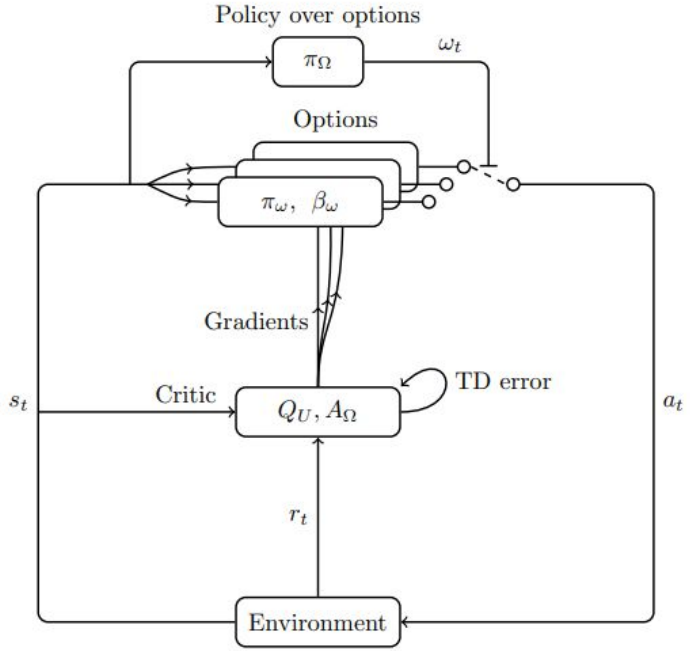


Figure 1: Diagram of the option-critic architecture. The option execution model is depicted by a *switch* $\perp$ over the *contacts* $-\circ$. A new option is selected according to π_Ω only when the current option terminates.

intra-option learning (Sutton, Precup, and Singh 1999, section 8). Specifically, the definition of the option-value function can be written as:

$$Q_\Omega(s,\omega) = \sum_a \pi_{\omega,\theta}(a \mid s)\, Q_U(s,\omega,a) \ , \qquad (1)$$

where $Q_U : \mathcal{S} \times \Omega \times \mathcal{A} \to \mathbb{R}$ is the value of executing an action in the context of a state-option pair:

$$Q_U(s,\omega,a) = r(s,a) + \gamma \sum_{s'} \mathrm{P}(s' \mid s,a)\, U(\omega,s') \ . \qquad (2)$$

$$U(\omega,s') = (1-\beta_{\omega,\vartheta}(s'))Q_\Omega(s',\omega) + \beta_{\omega,\vartheta}(s')V_\Omega(s') \qquad (3)$$

The state-value function can again be decomposed into immediate reward plus discounted value of successor state,

$$v_\pi(s) = \mathbb{E}_\pi \left[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s \right]$$

The action-value function can similarly be decomposed,

$$q_\pi(s, a) = \mathbb{E}_\pi \left[R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) \mid S_t = s, A_t = a \right]$$

has been initiated or is executing at time t in state s_t, then the probability of transitioning to (s_{t+1}, ω_{t+1}) in one step is:

$$\mathrm{P}\left(s_{t+1}, \omega_{t+1} \mid s_t, \omega_t\right) = \sum_a \pi_{\omega_t,\theta}\left(a \mid s_t\right) \mathrm{P}(s_{t+1} \mid s_t, a)($$

$$(1 - \beta_{\omega_t,\vartheta}(s_{t+1}))\mathbf{1}_{\omega_t=\omega_{t+1}} + \beta_{\omega_t,\vartheta}(s_{t+1})\pi_\Omega(\omega_{t+1} \mid s_{t+1})) \tag{4}$$

Clearly, the process given by (4) is homogeneous. Under mild conditions, and with options available everywhere, it is in fact ergodic, and a unique stationary distribution over state-option pairs exists.

intra-option learning (Sutton, Precup, and Singh 1999, section 8). Specifically, the definition of the option-value function can be written as:

$$Q_\Omega(s,\omega) = \sum_a \pi_{\omega,\theta}(a \mid s)\, Q_U(s,\omega,a) \ , \qquad (1)$$

where $Q_U : \mathcal{S} \times \Omega \times \mathcal{A} \to \mathbb{R}$ is the value of executing an action in the context of a state-option pair:

$$Q_U(s,\omega,a) = r(s,a) + \gamma \sum_{s'} \mathrm{P}(s' \mid s,a)\, U(\omega,s') \ . \quad (2)$$

option policies, assuming that they are stochastic and differentiable. From (1 , 2), it follows that:

$$\frac{\partial Q_\Omega(s,\omega)}{\partial\theta} = \left(\sum_a \frac{\partial \pi_{\omega,\theta}(a \mid s)}{\partial\theta} Q_U(s,\omega,a)\right) + \sum_a \pi_{\omega,\theta}(a \mid s) \sum_{s'} \gamma\, \mathrm{P}(s' \mid s,a)\, \frac{\partial U(\omega,s')}{\partial\theta}.$$

We can further expand the right hand side using (3) and (4), which yields the following theorem:

Theorem 1 (Intra-Option Policy Gradient Theorem)**.** *Given a set of Markov options with stochastic intra-option policies differentiable in their parameters θ, the gradient of the expected discounted return with respect to θ and initial condition (s_0,ω_0) is:*

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

where $\mu_\Omega(s,\omega \mid s_0,\omega_0)$ is a discounted weighting of state-option pairs along trajectories starting from (s_0,ω_0): $\mu_\Omega(s,\omega \mid s_0,\omega_0) = \sum_{t=0}^{\infty} \gamma^t \mathrm{P}(s_t = s, \omega_t = \omega \mid s_0,\omega_0)$.

Appendix

Augmented Process

If ω_t has been initiated or is executing at time t, then the discounted probability of transitioning to (s_{t+1}, ω_{t+1}) is:

$$\mathrm{P}_{\gamma}^{(1)}\left(s_{t+1}, \omega_{t+1} \middle| s_t, \omega_t\right) = \sum_a \pi_{\omega_t}\left(a \middle| s_t\right) \gamma \, \mathrm{P}(s_{t+1} \middle| s_t, a) \Big((1 - \beta_{\omega_t}(s_{t+1})) \mathbf{1}_{\omega_t = \omega_{t+1}} + \beta_{\omega_t}(s_{t+1}) \pi_{\Omega}\left(\omega_{t+1} \mid s_{t+1}\right) \Big) \ .$$

When conditioning the process from (s_t, ω_{t-1}), the discounted probability of transitioning to s_{t+1}, ω_t is:

$$\mathrm{P}_{\gamma}^{(1)}\left(s_{t+1}, \omega_t \mid s_t, \omega_{t-1}\right) = \Big((1 - \beta_{\omega_{t-1}}(s_t)) \mathbf{1}_{\omega_t = \omega_{t-1}} + \beta_{\omega_{t-1}}(s_t) \pi_{\Omega}\left(\omega_t \mid s_t\right) \Big) \sum_a \pi_{\omega_t}\left(a \mid s_t\right) \gamma \, \mathrm{P}\left(s_{t+1} \mid s_t, a\right) \ .$$

More generally, the k-steps discounted probabilities can be expressed recursively as follows:

$$\mathrm{P}_{\gamma}^{(k)}\left(s_{t+k}, \omega_{t+k} \mid s_t, \omega_t\right) = \sum_{s_{t+1}} \sum_{\omega_{t+1}} \Big(\mathrm{P}_{\gamma}^{(1)}\left(s_{t+1}, \omega_{t+1} \mid s_t, \omega_t\right) \mathrm{P}_{\gamma}^{(k-1)}\left(s_{t+k}, \omega_{t+k} \mid s_{t+1}, \omega_{t+1}\right) \Big) ,$$

$$\mathrm{P}_{\gamma}^{(k)}\left(s_{t+k}, \omega_{t+k-1} \mid s_t, \omega_{t-1}\right) = \sum_{s_{t+1}} \sum_{\omega_t} \Big(\mathrm{P}_{\gamma}^{(1)}\left(s_{t+1}, \omega_t \mid s_t, \omega_{t-1}\right) \mathrm{P}_{\gamma}^{(k-1)}\left(s_{t+k}, \omega_{t+k-1} \mid s_{t+1}, \omega_t\right) \Big) .$$

Proof of the Intra-Option Policy Gradient Theorem

Taking the gradient of the option-value function:

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

$$= \sum_a \left(\frac{\partial \pi_{\omega,\theta}(a|s)}{\partial\theta} Q_U(s,\omega,a) + \pi_{\omega,\theta}(a|s) \frac{\partial Q_U(s,\omega,a)}{\partial\theta} \right)$$

$$= \sum_a \left(\frac{\partial \pi_{\omega,\theta}(a \mid s)}{\partial\theta} Q_U(s,\omega,a) + \pi_{\omega,\theta}(a \mid s) \sum_{s'} \gamma \mathrm{P}(s' \mid s,a) \frac{\partial U(\omega,s')}{\partial\theta} \right), \quad (6)$$

$$\frac{\partial U(\omega,s')}{\partial\theta} =$$

$$(1-\beta_{\omega,\vartheta}(s'))\frac{\partial Q_\Omega(s',\omega)}{\partial\theta} + \beta_{\omega,\vartheta}(s')\frac{\partial V_\Omega(s')}{\partial\theta}$$

$$= (1-\beta_{\omega,\vartheta}(s'))\frac{\partial Q_\Omega(s',\omega)}{\partial\theta} + \beta_{\omega,\vartheta}(s') \sum_{\omega'} \pi_\Omega(\omega' \mid s') \frac{\partial Q_\Omega(s',\omega')}{\partial\theta}$$

$$= \sum_{\omega'} \left((1-\beta_{\omega,\vartheta}(s'))\mathbf{1}_{\omega'=\omega} + \beta_{\omega,\vartheta}(s')\pi_\Omega(\omega' \mid s') \right) \frac{\partial Q_\Omega(s',\omega')}{\partial\theta} . \quad (7)$$

where (7) follows from the assumption that θ only appears in the intra-option policies. Substituting (7) into (6) yields a recursion which, using the previous remarks about augmented process can be transformed into:

$$\frac{\partial Q_\Omega(s,\omega)}{\partial\theta} = \sum_a \frac{\partial \pi_{\omega,\theta}(a \mid s)}{\partial\theta} Q_U(s,\omega,a) +$$

$$\sum_a \pi_{\omega,\theta}(a \mid s) \sum_{s'} \gamma \mathrm{P}(s' \mid s,a) \sum_{\omega'} \left(\beta_{\omega,\vartheta}(s')\pi_\Omega(\omega' \mid s') + (1-\beta_{\omega,\vartheta}(s'))\mathbf{1}_{\omega'=\omega} \right) \frac{\partial Q_\Omega(s',\omega')}{\partial\theta}$$

$$= \sum_a \frac{\partial \pi_{\omega,\theta}(a \mid s)}{\partial\theta} Q_U(s,\omega,a) + \sum_{s'}\sum_{\omega'} \mathrm{P}_\gamma^{(1)}(s',\omega' \mid s,\omega) \frac{\partial Q_\Omega(s',\omega')}{\partial\theta}$$

$$= \sum_{k=0}^{\infty}\sum_{s',\omega'} \mathrm{P}_\gamma^{(k)}(s',\omega'|s,\omega) \sum_a \frac{\partial \pi_{\omega',\theta}(a|s')}{\partial\theta} Q_U(s',\omega',a).$$

The gradient of the expected discounted return with respect to θ is then:

$$\frac{\partial Q_\Omega(s_0,\omega_0)}{\partial\theta} =$$

$$\sum_{s,\omega}\sum_{k=0}^{\infty} \mathrm{P}_\gamma^{(k)}(s,\omega \mid s_0,\omega_0) \sum_a \frac{\partial \pi_{\omega,\theta}(a \mid s)}{\partial\theta} Q_U(s,\omega,a)$$

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

Proof of the Termination Gradient Theorem

The expected sum of discounted rewards starting from (s_1, ω_0) is given by:

$$U(\omega_0, s_1) = \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^{t-1} r_t \,\middle|\, s_1, \omega_0\right] .$$

We start by expanding U as follows:

$$\begin{aligned} U(\omega, s') &= (1 - \beta_{\omega,\vartheta}(s')) Q_\Omega(s', \omega) + \beta_{\omega,\vartheta}(s') V_\Omega(s') \\ &= (1 - \beta_{\omega,\vartheta}(s')) \sum_a \pi_{\omega,\theta}(a \mid s') \Big(\\ &\qquad r(s', a) + \sum_{s''} \gamma \, \mathrm{P}(s'' \mid s', a) \, U(\omega, s'') \Big) \\ &+ \beta_{\omega,\vartheta}(s') \sum_{\omega'} \pi_\Omega(\omega' \mid s') \sum_a \pi_{\omega',\theta}(a \mid s') \Big(\\ &\qquad r(s', a) + \sum_{s''} \gamma \, \mathrm{P}(s'' \mid s', a) \, U(\omega', s'') \Big) . \end{aligned}$$

The gradient of U is then:

$$\frac{\partial U(\omega, s')}{\partial \vartheta} = \frac{\partial \beta_{\omega,\vartheta}(s')}{\partial \vartheta} \underbrace{(V_\Omega(s') - Q_\Omega(s', \omega))}_{-A_\Omega(s', \omega)} +$$

$$(1 - \beta_{\omega,\vartheta}(s')) \sum_a \pi_{\omega,\theta}(a|s') \sum_{s''} \gamma \, \mathrm{P}(s''|s', a) \frac{\partial U(\omega, s'')}{\partial \vartheta} .$$

Using the structure of the augmented process:

$$\begin{aligned} \frac{\partial U(\omega, s')}{\partial \vartheta} &= -\frac{\partial \beta_{\omega,\vartheta}(s')}{\partial \vartheta} A_\Omega(s', \omega) + \\ &\qquad \sum_{\omega'} \sum_{s''} \mathrm{P}_\gamma^{(1)}(s'', \omega' \mid s', \omega) \frac{\partial U(\omega', s'')}{\partial \vartheta} \\ &= -\sum_{\omega', s''} \sum_{k=0}^{\infty} \mathrm{P}_\gamma^{(k)}(s'', \omega' \mid s', \omega) \frac{\partial \beta_{\omega',\vartheta}(s'')}{\partial \vartheta} A_\Omega(s'', \omega') . \end{aligned}$$

We finally obtain:

$$\begin{aligned} &\frac{\partial U(\omega_0, s_1)}{\partial \vartheta} = \\ &-\sum_{\omega, s'} \sum_{k=0}^{\infty} \mathrm{P}_\gamma^{(k)}(s', \omega \mid s_1, \omega_0) \frac{\partial \beta_{\omega,\vartheta}(s')}{\partial \vartheta} A_\Omega(s', \omega) \\ &= -\sum_{\omega, s'} \mu_\Omega(s', \omega | s_1, \omega_0) \frac{\partial \beta_{\omega,\vartheta}(s')}{\partial \vartheta} A_\Omega(s', \omega) . \end{aligned}$$

Algorithms and Architecture

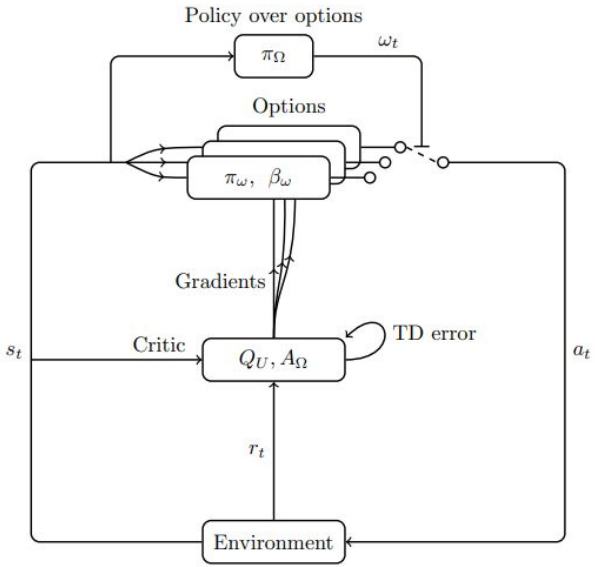


Figure 1: Diagram of the option-critic architecture. The option execution model is depicted by a *switch* $\perp$ over the *contacts* $-\circ$. A new option is selected according to π_Ω only when the current option terminates.

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

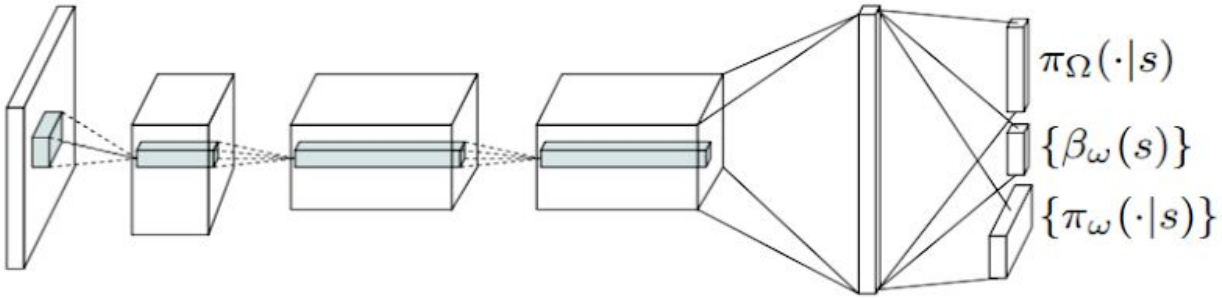
$s \leftarrow s_0$
Choose ω according to an ϵ-soft policy over options $\pi_\Omega(s)$
repeat
  Choose a according to $\pi_{\omega,\theta}(a \mid s)$
  Take action a in s, observe s', r

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

  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*

Deep neural network architecture



Input : A concatenation of the last 4 images

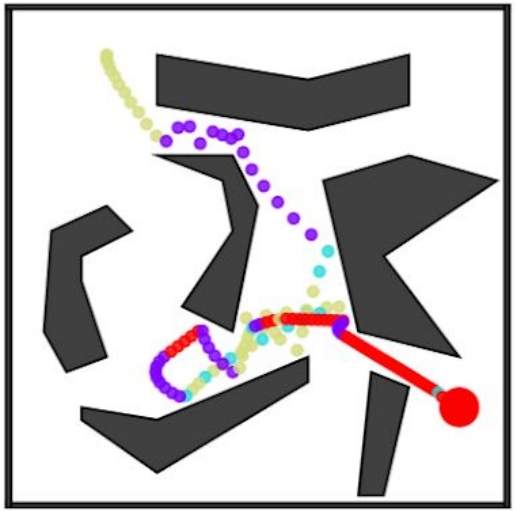
3 convolutional layers

1 Dense layer

Sigmoid layer for termination probability - One output
Dense Softmax - Intra policy

Experiments

Pinball Domain



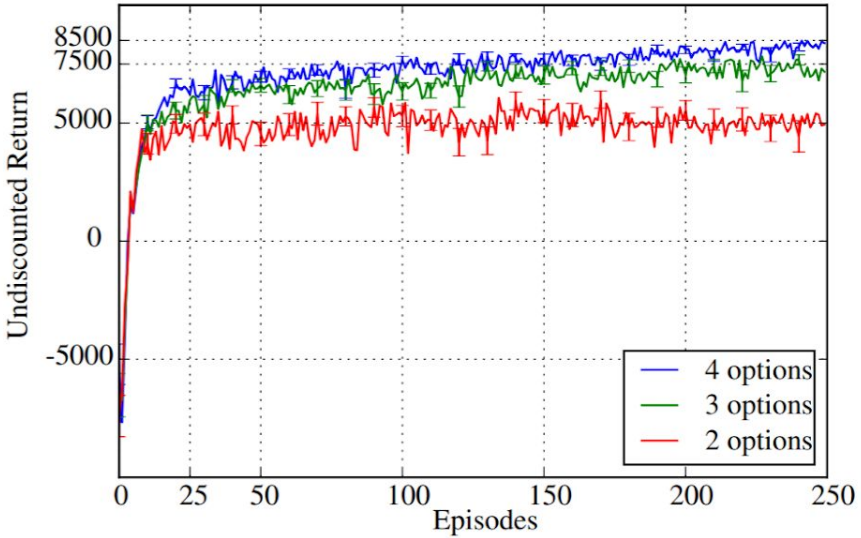


Figure 5: Learning curves in the Pinball domain.

Experiments

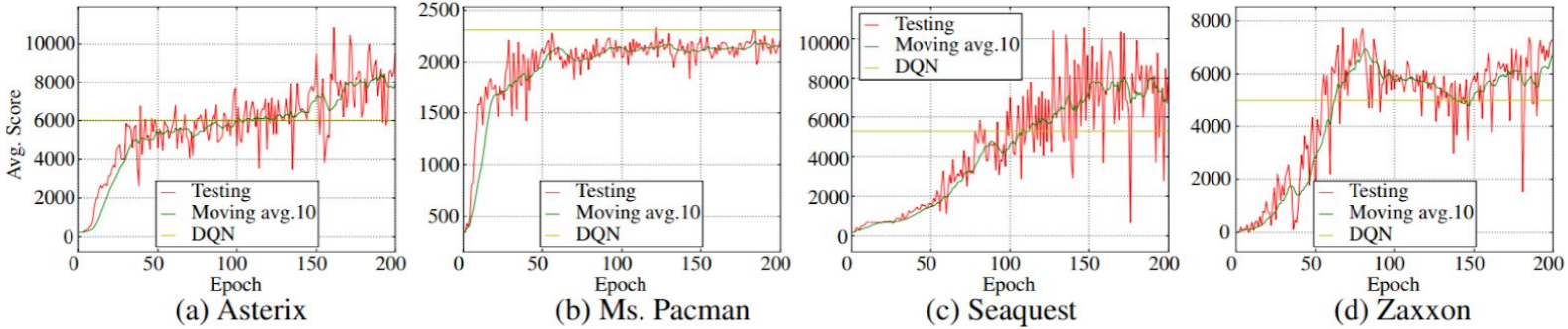


Figure 8: Learning curves in the Arcade Learning Environment. The same set of parameters was used across all four games: 8 options, 0.01 termination regularization, 0.01 entropy regularization, and a baseline for the intra-option policy gradients.

Limitation

Requires specifying the number of options