DAC: The Double Actor-Critic Architecture for Learning Options

NeurIPS 2019
Shangtong Zhang, Shimon Whiteson

Presenter: Ehsan Mehralian
March 17, 2020
Outline

• Problem statement
• Option Critic
• Double Actor Critic
Problem statement

• Temporal abstraction is a key component in RL:
  • Better exploration
  • Faster learning
  • Better generalization
  • Transfer learning

• MDP + Temporal Abstract actions = SMDP

• SMDP algorithms are data inefficient —> The option framework (Sutton et al., 1999)

• Rises two problems:
  • Learning options
  • Learning a master policy
Previous works

• Based on finding subgoals:
  • Difficult to scale up
  • Can be as expensive as the entire task

• Using value-based methods:
  • Can’t cope with large action spaces
  • Policy based methods have better convergence properties with function approximation
The Option Critic framework (PL Bacon et.al, 2017)

- Blurs the line between *discovering* options and *learning* options
- The first scalable end-to-end approach
  - No slow down within a single task
  - Faster convergence in transfer learning
Background

- MDP

\[ M \equiv \{S, A, R(s, a), P(s'|s, a), P_0(s), \gamma\} \]

- Goal:

\[ \pi^* = \underset{\theta}{\arg \max} \rho(\pi_\theta) = \underset{\theta}{\arg \max} \mathbb{E}_{\tau \sim p_\theta(\tau)} \left[ \sum_{t=1}^{\infty} \gamma^{t-1} r_t | s_0, \pi_\theta \right] \]

- Policy Gradient

\[ \frac{\partial \rho}{\partial \theta} = \sum_s d^\pi(s) \sum_a \frac{\partial \pi(s, a)}{\partial \theta} Q^\pi(s, a) \]

\[ d^\pi(s) = \sum_{t=0}^{\infty} \gamma^t Pr(s_t = s | s_0, \pi) \]
The Options Framework

- A Markovian Option $\omega \in \Omega$ is a triple: $(I_\omega, \pi_\omega, \beta_\omega)$
  
  - $I_\omega$ Initiation set $I_\omega \subset S$
  
  - $\pi_\omega$ Intra-option policy
  
  - $\beta_\omega$ Termination function $\beta_\omega : S \rightarrow [0, 1]$

- Let $\pi_{\omega,\theta}$ denote the intra-option policy of option $\omega$ parametrized by $\theta$ and $\beta_{\omega,\vartheta}$, the termination function of $\omega$ parameterized by $\vartheta$. 
The Option Critic framework

- Discounted return

\[ \rho(\Omega, \theta, \vartheta, s_0, w_0) = \mathbb{E}_{\Omega, \theta, \omega}\left[ \sum_{t=0}^{\infty} \gamma^t r_{t+1} | s_0, \omega_0 \right] \]
The Option Critic framework

• Discounted return

$$\rho(\Omega, \theta, \vartheta, s_0, w_0) = \mathbb{E}_{\Omega, \theta, \omega} \left[ \sum_{t=0}^{\infty} \gamma^t r_{t+1} | s_0, \omega_0 \right]$$

• Option-value function

$$Q_{\Omega}(s, w) = \sum_a \pi_{\omega, \theta}(a | s) Q_U(s, \omega, a)$$
The Option Critic framework

• Discounted return

\[ \rho(\Omega, \theta, \vartheta, s_0, w_0) = \mathbb{E}_{\Omega,\theta,\omega,\vartheta}[\sum_{t=0}^{\infty} \gamma^t r_{t+1} | s_0, \omega_0] \]

• Option-value function

\[ Q_{\Omega}(s, w) = \sum_{a} \pi_{\omega,\theta}(a|s)Q_{U}(s, \omega, a) \]

• Value of executing an action in the context of a state-option pair

\[ Q_{U}(s, \omega, a) = r(s, a) + \gamma \sum_{s'} P(s' | s, a) U(\omega, s') \]
The Option Critic framework

- Discounted return
  \[
  \rho(\Omega, \theta, \emptyset, s_0, w_0) = \mathbb{E}_{\Omega,\theta,\omega} \left[ \sum_{t=0}^{\infty} \gamma^t r_{t+1} | s_0, \omega_0 \right]
  \]

- Option-value function
  \[
  Q_\Omega(s, w) = \sum_a \pi_{\omega,\theta}(a|s)Q_U(s, \omega, a)
  \]

- Value of executing an action in the context of a state-option pair
  \[
  Q_U(s, \omega, a) = r(s, a) + \gamma \sum_{s'} P(s'|s, a)U(\omega, s')
  \]

- Option-value function upon arrival
  \[
  U(\omega, s') = (1 - \beta_{\omega,\emptyset}(s'))Q_\Omega(s', w) + \beta_{\omega,\emptyset}(s')V_\Omega(s')
  \]
The Option Critic framework (cont.)

• \((s, \omega)\) pairs lead to an augmented state space

• One step probability transition

\[
P(s_{t+1}, \omega_{t+1} \mid s_t, \omega_t) = \sum_a \pi_{\omega_t, \theta}(a \mid s_t) P(s_{t+1} \mid s_t, a_t) (1 - \beta_{\omega, \theta}(s_{t+1})) 1_{w_t = w_{t+1}} + \beta_{\omega, \theta}(s_{t+1}) \pi_\omega(\omega_{t+1} \mid s_{t+1})
\]
The option Critic framework (cont.)

- **Theorem 1 (Intra-Option Policy Gradient Theorem)**

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

- **Theorem 2 (Termination Gradient Theorem)**

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

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

• Works great!

• But:

  • Cannot directly leverage recent advances in gradient-based policy optimization from MDPs

• Solution:

  • DAC (The Double Actor-Critic Architecture)
The Double Actor-Critic Architecture for Learning Options (DAC)

- Reformulate the option framework as two parallel augmented MDPs
  - A policy based method
  - All policy optimization algorithms can be used off the shelf (advantage over Option Critic)
- Apply an actor-critic algorithm on each augmented MDP (double actor critic)
- Show that one critic is enough.
Two Augmented MDPs

• The high-MDP: $M^H$
  
  • The agent makes high-level decisions (i.e., option selection) in $M^H$ according to $\pi, \{\beta_\omega\}$ and thus optimizes $\pi, \{\beta_\omega\}$

• The low-MDP: $M^L$

  • The agent makes low-level decisions (i.e., action selection) in $M^L$ according to $\{\pi_\omega\}$ and thus optimizes $\{\pi_\omega\}$
High MDP

- Define a dummy option $\#$ and $\Omega^+ = \Omega \cup \{\#\}$

- Interpret a state-option pair as a new state and an option as a new action.

\[
M^\mathcal{H} \equiv \{S^\mathcal{H}, A^\mathcal{H}, p^\mathcal{H}, p_0^\mathcal{H}, r^\mathcal{H}, \gamma\}, \quad S^\mathcal{H} \equiv \mathcal{O}^+ \times S, \quad A^\mathcal{H} \equiv \mathcal{O},
\]

\[
p^\mathcal{H}(S_{t+1}^\mathcal{H}|S_t^\mathcal{H}, A_t^\mathcal{H}) \doteq p^\mathcal{H}((O_t, S_{t+1})|(O_{t-1}, S_t), A_t^\mathcal{H)) \doteq \mathbb{I}_{A_t^\mathcal{H}=O_t} p(S_{t+1}|S_t, O_t),
\]

\[
p_0^\mathcal{H}(S_0^\mathcal{H}) \doteq p_0^\mathcal{H}((O_{-1}, S_0)) \doteq p_0(S_0) \mathbb{I}_{O_{-1}=\#},
\]

\[
r^\mathcal{H}(S_t^\mathcal{H}, A_t^\mathcal{H}) \doteq r^\mathcal{H}((O_{t-1}, S_t), O_t) \doteq r(S_t, O_t)
\]

- Define Markov Policy

\[
\pi^\mathcal{H}(A_t^\mathcal{H}|S_t^\mathcal{H}) \doteq \pi^\mathcal{H}(O_t|(O_{t-1}, S_t)) \doteq p(O_t|S_t, O_{t-1}) \mathbb{I}_{O_{t-1} \neq \#} + \pi(S_t, O_t) \mathbb{I}_{O_{t-1}=\#}
\]
Low MDP

- Interpret state-option pair as state and leave actions unchanged.

\[ M^L = \{ S^L, A^L, p^L, p_0^L, r^L, \gamma \}, \quad S^L = S \times O, \quad A^L = A, \]

\[ p^L(S^L_{t+1}|S^L_t, A^L_t) = p^L((S_{t+1}, O_{t+1})|(S_t, O_t), A_t) = p(S_{t+1}|S_t, A_t)p(O_{t+1}|S_{t+1}, O_t), \]

\[ p_0^L(S^L_0) = p^L((S_0, O_0)) = p_0(S_0)\pi(S_0, O_0), \]

\[ r^L(S^L_t, A^L_t) = r^L((S_t, O_t), A_t) = r(S_t, A_t) \]

- Define Markov policy:

\[ \pi^L(A^L_t|S^L_t) = \pi^L(A_t|(S_t, O_t)) = \pi_{O_t}(A_t|S_t) \]
How to sample from these MDPs?

• Consider trajectories with non zero probability:

\[ \Omega = \{ \tau | p(\tau | \pi, O, M) > 0 \} \]

\[ \Omega^H = \{ \tau^H | p(\tau^H | \pi^H, M^H) > 0 \} \]

\[ \Omega^L = \{ \tau^L | p(\tau^L | \pi^L, M^L) > 0 \} \]

• Where \( \tau = \{ S_0, O_0, S_1, O_1, \ldots, S_T \} \)

• Define functions: \( f^H : \Omega \rightarrow \Omega^H \) \( f^L : \Omega \rightarrow \Omega^L \)
How to sample from these MDPs?

- Lemma 1: $f^H : \Omega \rightarrow \Omega^H$ is a bijection and
  
  $$p(\tau|\pi, O, M) = p(\tau^H|\pi^H, M^H), r(\tau) = r(\tau^H)$$

- Lemma 2: $f^L : \Omega \rightarrow \Omega^L$ is a bijection and
  
  $$p(\tau|\pi, O, M) = p(\tau^L|\pi^L, M^L), r(\tau) = r(\tau^L)$$

- So, Sampling from $\{\pi, O, M\}$ is equivalent to sampling from $\{\pi^H, M^H\}$ and $\{\pi^L, M^L\}$
How to optimize $\pi^H$ and $\pi^L$ in these MDPs?

- Proposition:

$$J = \int r(\tau)p(\tau|\pi, O, M)d\tau = \int r(\tau^H)p(\tau^H|\pi^H, M^H)d\tau^H = \int r(\tau^L)p(\tau^L|\pi^L, M^L)d\tau^L$$

- Optimizing $\pi, O$ in $M$ is equivalent to optimizing $\pi^H$ in $M^H$ and optimizing $\pi^L$ in $M^L$
How to optimize $\pi^H$ and $\pi^L$ in these MDPs?

- Observation 1: $M^H$ depends on $\{\pi_o\}$ while $\pi^H$ depends on $\pi$ and $\{\beta_o\}$
  - When we keep the intra-option policies $\{\pi_o\}$ fixed and optimize $\pi^H$, we are implicitly optimizing $\pi$ and $\{\beta_o\}$

- Observation 2: $M^L$ depends on $\pi$ and $\{\beta_o\}$ while $\pi^L$ depends on $\{\pi_o\}$
  - When we keep the master policy $\pi$ and the termination conditions $\{\beta_o\}$ fixed and optimize $\pi^L$, we are implicitly optimizing $\{\pi_o\}$

---

**Algorithm 1: Pseudocode of DAC**

**Input:**
Parameterized $\pi$, $\{\pi_o, \beta_o\}_{o \in O}$
Policy optimization algorithms $A_1, A_2$

Get an initial state $S_0$
$t \leftarrow 0$

**while** True **do**
  Sample $O_t$ from $\pi^H(\cdot | (O_{t-1}, S_t))$
  Sample $A_t$ from $\pi^c(\cdot | (S_t, O_t))$
  Execute $A_t$, get $R_{t+1}, S_{t+1}$
  // The two optimizations can be done in any order or alternatively
  Optimize $\pi^H$ with $(S^H_t, A^H_t, R_{t+1}, S^H_{t+1})$ and $A_1$
  Optimize $\pi^c$ with $(S^c_t, A^c_t, R_{t+1}, S^c_{t+1})$ and $A_2$

  $t \leftarrow t + 1$

**end**
Do we need two critics?

• **Proposition:** When state-value functions are used as critics, one critic can be expressed in terms of the other, and hence only one critic is necessary

\[
v_{\pi^H}((o, s')) = \sum_{o'} \pi^H(o'|(o, s')) v_{\pi^L}((s', o')),
\]

where

\[
v_{\pi^H}((o, s')) \doteq \mathbb{E}_{\pi^H, M^H} \left[ \sum_{i=1}^{\infty} \gamma^{i-1} R_{t+i}^H \mid S_t^H = (o, s') \right],
\]

\[
v_{\pi^L}((s', o')) \doteq \mathbb{E}_{\pi^L, M^L} \left[ \sum_{i=1}^{\infty} \gamma^{i-1} R_{t+i}^L \mid S_t^L = (s', o') \right],
\]
Experiments setup

- Want to see
  - DAC vs. existing gradient based option learnings
  - DAC vs. hierarchy-free methods
- DAC can use any policy optimization (AC, SAC, NAC, PPO, ...).
- Here focus on DAC + PPO.
Results

• Single task

Figure 1: Online performance on a single task

• DAC + A2C similar to OC

• DAC + PPO similar to PPO
Results

- Transfer Learning

CartPole = (balance, balance_sparse)
Reacher = (easy, hard)
Cheetah = (run, backward)
Fish = (upright, downleft)
Walker1 = (squat, stand)
Walker2 = (walk, backward)

Figure 2: Online performance for transfer learning
Recap

• **Problem:** An end to end, policy based method to learn options and policy over them

• **Solution:** Option Critic

• **Limitation:** Can’t use other policy optimization algorithms off-the-shelf

• **Solution:** Reformulate the SMDP of the option framework as two augmented MDPs (Double Actor Critic)
Thank you