Options

Temporal Abstraction in Reinforcement Learning

Laurens Weitkamp 01-15-2020

Problem Statement

Real life tasks come at multiple scales;

- Biking here from home involves planning over long term, and short-term muscle movements

Can we define an MDP like the scenario above that can be solved easily with RL algorithms?

Options

Generalize the idea of a sequence of actions as an *option*. *We pre-define the number of options*

each option ω is defined with the following

- A termination function β_{ω}(s) in [0, 1]
- A policy π_{ω}(a|s)
- (sometimes we additionally define an initiation set, in what state is this option available?)

We learn a *policy-over-options* π_{Ω}(ω|s)

- What option should we choose in this state?

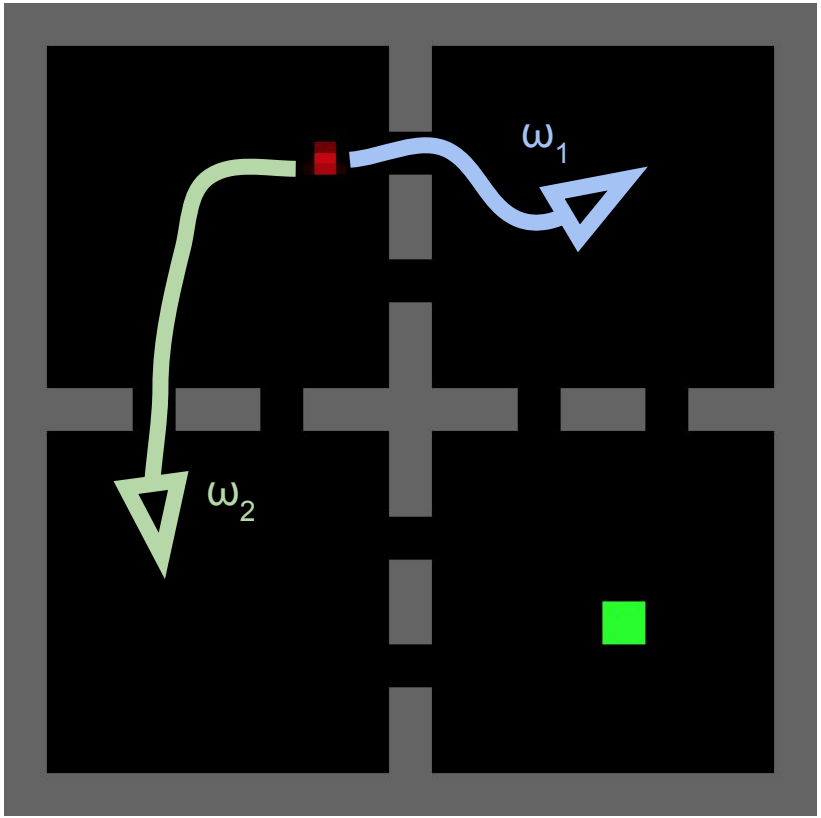
Typical Case Study

Learn to solve the gridworld in 1k episodes

Change the goal location

→ only the policy-over-options should require training to solve this new location

Options should be highly transferable!



How can we learn options?

Pre deep learning:

- Q-learning(!)
- Intra options

Post deep learning

- Option-critic and its many extensions

Learning options: one-step Q-learning[1]

- Sample $\omega \sim \pi_\Omega(\omega|s_t)$
- Follow ω for k steps, when $\beta=1$ with $\beta \sim \beta_\omega(s_{t+k})$

$Q(\omega, s_t) \leftarrow Q(\omega, s_t) + \alpha\,[r + \gamma^k \max_{\omega'} Q(\omega', s_{t+k+1}) - Q(\omega, s_t)]$ (r is cumulative and discounted)

- Sample $\omega \sim \pi_\Omega(\omega|s_{t+k+1})$

Can you see an issue here?

We only update a single option given a (possibly) large number of steps

[1]: Richard S. Sutton, Doina Precup, Satinder Singh

Learning options: intra-options[1]

The premise of intra-options learning is the following:

Act as if the option currently active has been activated precisely in this step

But we need to define a new function; *The value function upon arrival in a new state*

$$U(\omega, s_t) \leftarrow (1 - \beta_\omega(s)) Q(\omega, s_t) + \beta_\omega(s) \max_{\omega'} Q(\omega', s_t)$$

The value if we keep using the current option

The greedy value if we switch to a new option

$$Q(\omega, s_t) \leftarrow Q(\omega, s_t) + \alpha [r_{t+1} + \gamma U(\omega, s_{t+1}) - Q(\omega, s_t)]$$

[1]: Richard S. Sutton, Doina Precup, Satinder Singh

Deep Options RL: option-critic [1]

Utilize the idea of intra-option value learning!

1. Extend the idea of learning options through policy gradients
    a. Introduces two new theorems; intra-option policy gradient and termination gradient
2. Learn $Q_{\Omega}(\omega, s)$ and $Q_{U}(\omega, s, a)$
3. The critic consists of Q_{U} and $A_{\Omega}(\omega, s) = Q_{\Omega}(\omega, s) - V_{\Omega}(s)$

~ Capable of beating atari games

[1]: Pierre-Luc Bacon, Jean Harb, Doina Precup

Deep Options RL

1. When waiting is not an option: Learning options with a deliberation cost
2. Inferring options: we treat options as latent variables
3. Discovering options for exploration by minimizing cover time

And much more, but not highly cited papers

What’s wrong with options? [1]

- Termination is hard to learn
  - Forget about it?
  - Penalty for short options?
- Learning options often fails or ‘collapses’
  - The policy-over-options **chooses a new option each state**
  - The policy-over-options **chooses only one option in each state**
- Are options always interpretable?

But those are only technical issues

What’s wrong with options? [2]

Is the idea of an option-based model realistic?

- Pre-defining the number of options is problematic
- Wouldn’t we want to learn infinitely many options?
- Why wouldn’t we add another level of hierarchy?
    - A policy over policies over options over options

Hierarchical/Temporal Alternatives

Feudal Reinforcement Learning[1] (deep learning version works much different)

- Create multiple resolutions of the environment
- Learn a set of policies at each level

Meta-Learning Shared Hierarchies[2]

- Similar to options, but no termination function; walk for N steps and activate new sub-policy (no option-value function, everything is PPO..)

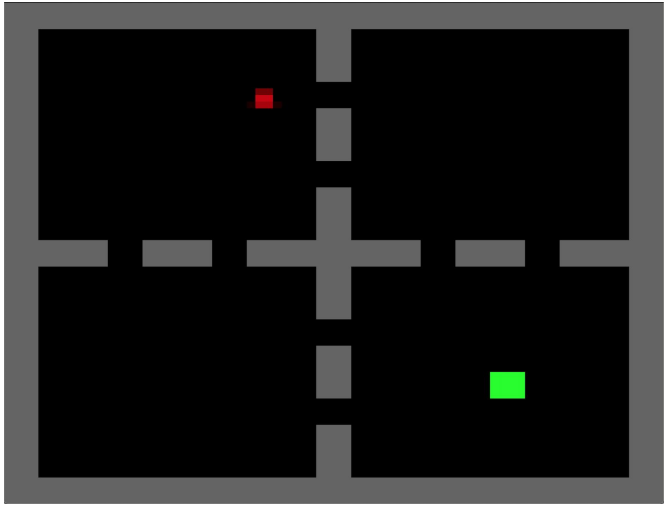
Data-efficient hierarchical reinforcement learning[3]

[1]: Peter Dayan, Geoffrey Hinton
[2]: Kevin Frans, Jonathan Ho, Xi Chen, Pieter Abbeel, John Schulman
[3]: O Nachum, SS Gu, H Lee, S Levine

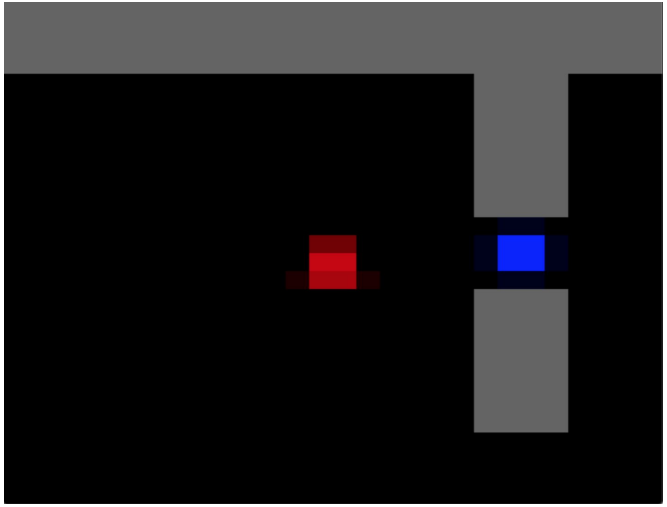
What Am I Doing?

Assume you have some GPS that might not account for obstacles (road blocking)

Can this scale up to multiple environments, i.e. can we have highly transferable and understandable options using this environmental split?



Global Information



Local Information