Paper Critique

Shuvrajeet Das, DA24D402

Course: DA7400, Fall 2024, IITM
Paper: [UNVEILING OPTIONS WITH NEURAL DECOMPOSITION]
Date: [04-08-2024]

Make sure your critique Address the following points:
1. The problem the paper is trying to address
2. Key contributions of the paper
3. Proposed algorithm/framework
4. How the proposed algorithm addressed the described problem
Note: Be concise with your explanations. Unnecessary verbosity will be penalized. Please don't exceed 2 pages.

---

1 The problem the paper is trying to address

The paper addresses the problem of limited generalization in reinforcement learning (RL) agents. Specifically, it tackles the challenge that agents often learn policies for specific tasks without the ability to generalize or transfer this knowledge to related tasks. This limitation hinders the efficiency and adaptability of RL agents when faced with new but similar tasks.

2 Key contributions of the paper

The contributions are:

- **Neural Decomposition for Option Extraction:** The paper introduces a novel method for decomposing neural networks encoding policies into sub-policies, which are then used to create temporally extended actions known as options. This approach allows for the extraction of options even when the original network was not explicitly designed to learn them.
- **Generalization Across Tasks:** The proposed method enhances the ability of RL agents to transfer knowledge across tasks. By decomposing policies into sub-policies, the method identifies options that can generalize to new tasks, improving the efficiency of learning in different but related environments.
- **Levin Loss-Based Option Selection:** The paper introduces a selection mechanism based on minimizing the Levin loss for a uniform policy over the extracted options. This helps in choosing the most effective options from a potentially large set, thereby guiding the agent's exploration more effectively in new tasks.

3 Proposed algorithm/framework

The algorithm consists of the following steps:

1. **Decompose** the neural network $\pi_\theta(s)$ into a neural tree.
2. **Extract** sub-policies from the neural tree.
3. **Wrap** sub-policies in while-loops to create options.
4. **Select** useful options using Levin loss.

4 How the Proposed Algorithm Addresses the Problem

4.1 Decomposition of Neural Networks into Sub-Policies

Given a neural network $\pi_\theta(s)$ for policy π, decompose it into a "neural tree" where each node represents a linear function $f_i(s) = W_i s + b_i$. The neural tree can be viewed as an oblique decision tree with linear decision boundaries.

4.2 Creation of Temporally Extended Actions (Options)

Extracted sub-policies π_i from the neural tree are executed within while-loops, effectively producing options. Each option β_i is defined as:

$$\beta_i(s) = \begin{cases} \pi_i(s), & \text{for } t < T_i \\ \text{noop}, & \text{otherwise} \end{cases}$$

where T_i is the number of iterations for which the option β_i is active.

4.3 Selection of Useful Options Using Levin Loss

To manage the exponential growth of options with respect to network size, use Levin loss $\mathcal{L}_{\text{Levin}}$:

$$\mathcal{L}_{\text{Levin}} = \mathbb{E}_{\pi'} [\text{steps to reach high-reward states}]$$

where π' denotes a subset of selected options that minimizes the expected steps to reach high-reward states in a new task.