Paper Critique

Shuvrajeet Das, DA24D402

Course: DA7400, Fall 2024, IITM
Paper: [LEARNING MULTI-LEVEL HINDSIGHT]
Date: [30-08-2024]

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 core issue is that learning multiple levels of policies in parallel is inherently unstable because changes in a policy at one level can affect the transition and reward functions at higher levels. This makes it difficult to learn a hierarchy of policies jointly, especially in continuous state and action spaces.

The problem is exacerbated by non-stationary state transition functions in nested, multi-level hierarchies. When all policies within the hierarchy are trained simultaneously, the transition functions at higher levels continue to change as long as the policies below them are being updated.

2 Key contributions of the paper

The paper introduces a framework HAC or the Hierarchical Actor Critic Framework with the following properties:

- **Hierarchical Learning:** HAC allows agents to simultaneously and independently learn multiple levels of policies, addressing instability in nested hierarchies.
- **Efficient Learning:** By treating lower-level policies as optimal, HAC trains each level independently, speeding up learning in complex tasks with continuous states and actions.
- **Hindsight Techniques:** HAC introduces hindsight action and goal transitions, enabling the agent to learn from achieved states and goals during training, which is useful in sparse reward environments.
- **Subgoal Testing:** HAC features a mechanism to evaluate the achievability of subgoals, preventing the pursuit of unrealistic goals and fostering practical goal-setting.

3 Proposed algorithm/framework

Algorithm 1 Hierarchical Actor-Critic (HAC)

Require: Key agent parameters: number of levels in hierarchy k, maximum subgoal horizon H, and subgoal testing frequency λ.

Ensure: k trained actor and critic functions $\pi_0, \ldots, \pi_{k-1}, Q_0, \ldots, Q_{k-1}$

for M episodes **do**
    $s \leftarrow S_{\text{init}}, g \leftarrow G_{k-1}$
    `train_level`$(k-1, s, g)$
    Update all actor and critic networks
end for
function TRAIN-LEVEL(i :: level, s :: state, g :: goal)
    $s_i \leftarrow s, g_i \leftarrow g$
    for H attempts or until g_n, $i \leq n < k$ achieved **do**
        $a_i \leftarrow \pi_i(s_i, g_i) + \text{noise}$
        if $i > 0$ **then**
            Determine whether to test subgoal a_i
            $s'_i \leftarrow$ `train_level`$(i-1, s_i, a_i)$
        else
            Execute primitive action a_0 and observe next state s'_0
        end if
        `Replay Buffer`$_i \leftarrow [s = s_i, a = a_i, r = \{-1, 0\}, s' = s'_i, g = g_i, \gamma = \{\gamma, 0\}]$
        if $i > 0$ and a_i missed then **then**
            if a_i was tested then **then**
                `Replay Buffer`$_i \leftarrow [s = s_i, a = a_i, r = \text{Penalty}, s' = s'_i, g = g_i, \gamma = 0]$
            end if
            $a_i \leftarrow s'_i$
        end if
        `Replay Buffer`$_i \leftarrow [s = s_i, a = a_i, r = TBD, s' = s'_i, g = g_i, \gamma = TBD]$
        $s_i \leftarrow s'_i$
    end for
    `Replay Buffer`$_i \leftarrow$ `Perform HER using HER_Storage`$_i$ `transitions`
    return s'_i
end function

4 How the proposed algorithm addressed the problem

1. **Instability Due to Non-Stationary Transitions**
   - **Problem:** Higher-level policies become unstable as lower-level policies change during learning.
   - **Solution:** HAC uses hindsight action transitions, treating the actual achieved state as if it were intended. This stabilizes higher-level policy learning by simulating an optimal lower-level policy.
2. **Difficulty in Parallel Learning**
   - **Problem:** Changes at one level can destabilize others, complicating parallel policy learning.
   - **Solution:** HAC treats lower-level policies as optimal through hindsight transitions, allowing independent training of each level and stabilizing the overall process.