COL778 - Principles Of Autonomous Systems (Assignment -III)

2024JRB2022 - Sai Satvik S, 2024JRB2028 - Parthasaradhi Reddy N

April 23, 2025

Hyper parameters

- Beta - 0.99 and exponential decay
- Max episode length - 5000
- Iterations(max): Ant-150, Hopper-100
- LR=0.0001, Adam Optimizer, MSE Loss
- Ant - v4: Hidden Layers - 2 , Size - 64
- Hopper - v4: Hidden layers - 3 , Size - 32

Implementation

- **Mixture Policy (`forward`):**
  - Receives a batch of observations.
  - Draws a uniform random number in $[0, 1)$.
  - If the draw is below β, returns the expert's action for each state.
  - Otherwise, returns the learner's (MLP) action.
  - Ensures expert outputs are converted to tensors if needed.
- **Pure Learner for Evaluation (`get_action`):**
  - With gradients disabled, returns only the learner's action given an observation.
- **Supervised Update (`update`):**
  - Zeroes gradients on the optimizer.
  - Performs a forward pass of the learner's network on a batch of states.
  - Computes the mean-squared error to the expert-labeled actions.
  - Backpropagates and takes an optimizer step.
  - Returns the scalar loss value.
- **Training Iteration (`train_iteration`):**
  1. *Batch Supervised Training:*
     - Sample all stored transitions.
     - Split into mini-batches and call `update` on each.
  2. *Decay Mixing Coefficient:* Update β according to a schedule (e.g. exponentially decreasing in iteration count).
  3. *Data Aggregation via Rollouts:*
     - For several trajectories:
       * Reset environment and step until termination.

        - ∗ At each state, call `forward()` to sample either expert or learner.
        - ∗ Always query the expert for the “true” action label.
        - ∗ Store $(s, a_{\text{expert}}, s', r, \text{done})$ in a local list.
    - – Add all new expert-labeled transitions to the replay buffer.
4. *Evaluation & Model Checkpointing:*
    - – Use `get_action()` to collect evaluation rollouts.
    - – Compute average return; if it improves upon the best so far, save the learner’s parameters.
5. **Returns the loss, collected rollouts, and the cumulative environment steps.**

Summary

- Using exponential decay on beta value gave better performance when compared to fixed beta value as agent exponentially stops relying more on the expert.Also as the beta value is high initially replay buffer is populated by expert actions and agent learns better.
- As we train for more iterations, the ant agent is learning to stabilize itself, and move forward. While initially it is moving faster, and jumping high, and rolling out.
- Higher episode length yields longer travel distance by the agents
- We all tried different approach by mixing the actions both from the expert and model, it is like exploration, and it giving good results. Its like adding noise the actions.

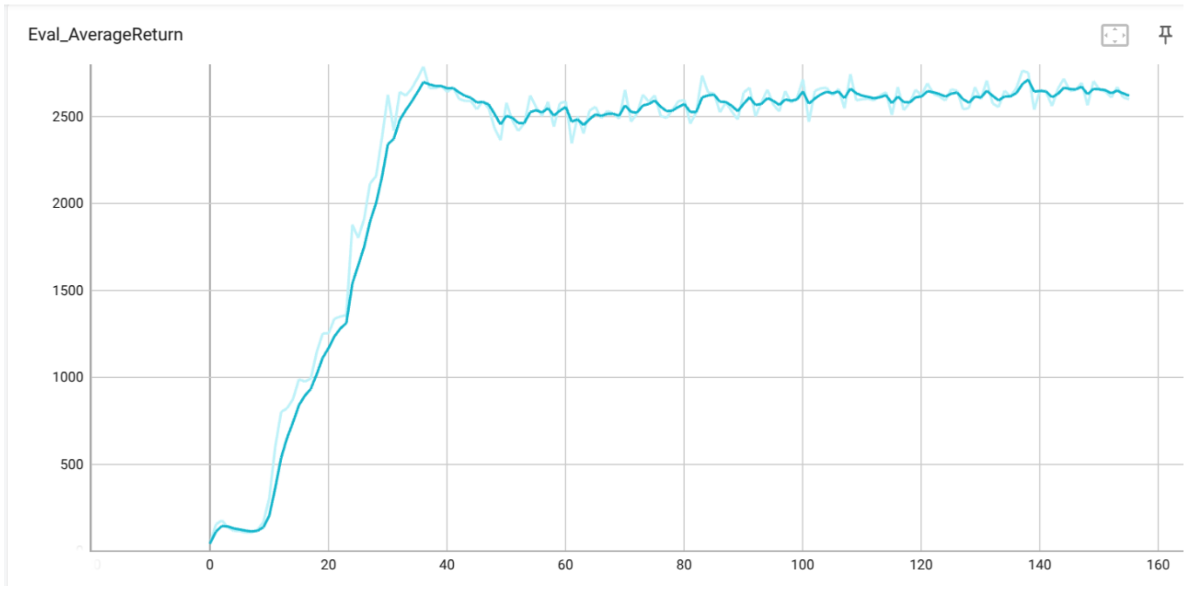

Figure 1: Ant agent training average return

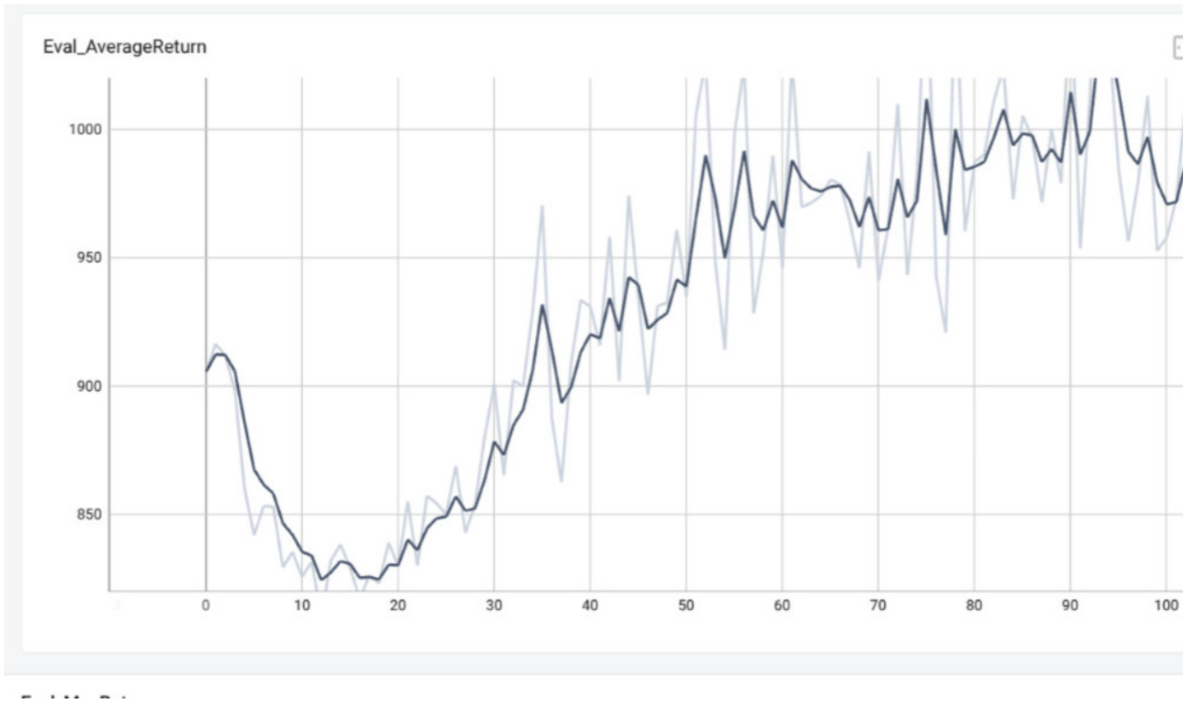

Figure 2: Hopper agent training average return