Assignment 3: Imitation Learning

Varun Shinde – 2024JRB2029

April 23, 2025

1 Algorithm Approach

We employ the DAgger (Dataset Aggregation) algorithm for imitation learning. DAgger is an iterative approach designed to address the limitations of behavior cloning. Instead of solely relying on expert trajectories (which the learner may not visit), the agent generates its own trajectories and queries the expert for corresponding actions. These expert-labeled states are then used to improve the learner's policy.

2 Environment Specifications

2.1 Observation Spaces

1. **Hopper-v4:** $(-\infty, \infty)$, shape: (11,), dtype: float64
   - **qpos (5 elements):** Position values of the robot's body parts.
   - **qvel (6 elements):** Velocity values (time derivatives of positions).
2. **Ant-v4:** $(-\infty, \infty)$, shape: (105,), dtype: float64
   - **qpos (13 elements):** Position values of the robot's body parts.
   - **qvel (14 elements):** Velocities of these body parts.
   - **cfrc_ext (78 elements):** External forces acting on each body part. Represented as a 13×6 array (force and torque components in 3D).

2.2 Action Spaces

1. **Hopper-v4:** $(-1.0, 1.0)$, shape: (3,), dtype: float64
   - A 3-dimensional vector representing torque applied to each joint.
2. **Ant-v4:** $(-\infty, \infty)$, shape: (8,), dtype: float64
   - An 8-dimensional vector representing motor commands to the ant's joints.

2.3 Evaluation:

We are making use of reward as a evaluation metric, while saving the best model. This is done since reward will be a appropriate metric to judge whether the current state of the learner policy is performing better than rest of iterations.

3 Pseudocode:

Algorithm 1 DAgger Training Loop

while not converged **do**
  Generate rollout using current learner policy π_θ to collect state-action pairs
  Add generated trajectories to the replay buffer $\mathcal{D}$
  Sample n trajectories from the replay buffer
  for each sampled state s_t **do**
    Query expert policy π^* to obtain expert action $a_t^* = \pi^*(s_t)$
    Predict learner action $a_t = \pi_\theta(s_t)$
    Compute MSE loss: $\mathcal{L} = \frac{1}{n}\sum_{t=1}^{n} \|a_t - a_t^*\|^2$
  end for
  Update learner policy π_θ using gradient descent on $\mathcal{L}$
  Evaluate current policy π_θ by generating evaluation trajectories
  Compute average reward R_{avg}
  if $R_{\text{avg}} > R_{\text{best}}$ **then**
    Save model checkpoint
    $R_{\text{best}} \leftarrow R_{\text{avg}}$
  end if
end while

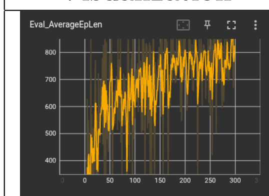
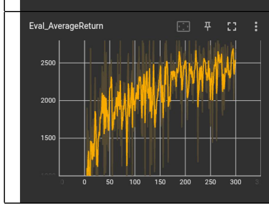
<table>
<tr><th>Visualization</th><th>Inference</th></tr>
<tr><td></td><td>This graph shows the average episode length. The increasing trend indicates that the agent is learning to sustain longer episodes, reflecting improved stability and performance.</td></tr>
<tr><td></td><td>This plot represents the average return (cumulative reward). The upward trend suggests that the learner is successfully mimicking the expert and achieving higher rewards, validating the effectiveness of the imitation learning approach.</td></tr>
</table>

Table 1: Evaluation Metrics During DAgger Training on Hopper-v4 Environment

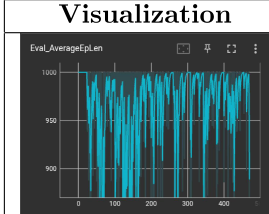
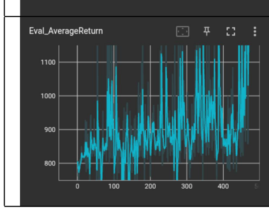
<table>
<tr><th>Visualization</th><th>Inference</th></tr>
<tr><td></td><td>This graph shows the average episode length. The increasing trend indicates that the agent is learning to sustain longer episodes, reflecting improved stability and performance.</td></tr>
<tr><td></td><td>This plot represents the average return (cumulative reward). The upward trend suggests that the learner is successfully mimicking the expert and achieving higher rewards, validating the effectiveness of the imitation learning approach.</td></tr>
</table>

Table 2: Evaluation Metrics During DAgger Training on Ant-v4 Environment