Robot Learning 2023

02069144

Coursework 2

Description Of Your Implementation

High-level summary

Our implementation is a closed loop planning method using cross entropy to determine an optional action sequence. Since we are trying to learn the best representation of the model, parameters such as planning horizon, number of resamples and the percentage of samples (k) are hyperparameters that were tuned for training and testing.
During training, we used a planning horizon of 30, number of refits 2 and k of 0.3. These parameters ensured that the paths found where highly explorative, incentivising the robot to explore a wide range of states while still trying to reach the goal. Each episode had a maximum length of 200 (since this is the maximum time we have in testing) to reach the goal. If the robot reached the goal before that time or reached the maximum episode of length, we iteratively changed the goal from the real goal state and the initial state, this increased model exploration and prevented wasting time to reset the robot. Resetting was only done when the robot got stuck, impedance one, the position of the robot stayed the same after taking an action. During testing, we used a planning horizon of 20, number of refits 3 and k of 0.2. These parameters made the path found more exploitative, increasing the number of refits, and decreasing the percentage of samples we pick, thus getting better quality plans. We only replan if we have no more planned actions (we use all the actions in the planning horizon) or are within 0.15 of the goal and use the "Funnelling technique".
In both training and testing we used the same reward function; number of samples used in cross entropy (400) and used a "Funnelling technique" to ensure the robot reached the goal consistently when in a close radius of the goal. We used a dynamic reward function, when the robot is within 0.2 the goal the reward is dense and when the robot distance is greater than 0.2 it is sparse. The "Funnelling technique" was used when the robot within 0.15 of the goal. Within this distance, the robot uses cross entropy planning with a planning horizon of 10, but with 100 number of samples used and k of 0.1, however we recalculate after every step.

Interesting findings and ideas

The implementations which I found particularly interesting which improved the learned model was the method which we implemented that prevented the need to reset the model. As explained above, during training, the goal of the robot dynamically changed between the initial position to goal state. This saved valuable resetting time, but also improved the learned model around the goal state, improving the time to reach the goal during testing.
Another technique which improved the robot's performance was the use of the "Funnelling technique", with the parameters describe the time taken to calculate an action was less 0.3 seconds. This method worked much better than explicitly calculating the best direction to the goal without considering the model dynamics, in many cases the robot did not reach the goal when directly executing the direct path to the goal, a small amount of planning saved large amounts of time in practice.
There were other strategies we explored but resulted in worse performance. Using an annealing strategy for the planning horizon during testing, we began the planning horizon at 40 and annealed it by a factor of 0.8. However, we found that using a smaller planning horizon produced better and more consistent results. Another, idea we explored was to use the best path we found during testing to help guide the robot during training, by using the mean angle of the path as the initial mean rather than 0, in the first pass of the cross-entropy method. However, this method introduced too much bias and was not able to generalize well when testing it to many different environments. We also tried to incorporate the model's uncertainty in the reward function during testing to incentivise the robot to move towards uncertain areas, however this did not produce better results, since we are already trying to be explorative with the methods describe above.