Predictive Control Using Learned State Space Models via Rolling Horizon Evolution

Alvaro Ovalle and Simon M. Lucas

Queen Mary University of London

icabs Guangzhou 2021

PRL Workshop

Motivation

- A large part of the interest in model-based reinforcement learning derives from the potential utility to acquire a forward model capable of strategic long-term decision making.
- If an agent succeeds in learning a useful predictive model, it still requires a mechanism to harness it to generate and select among competing simulated plans.
- Following the theme of the workshop, we explore the combination of current deep learning and variational inference techniques to learn a model of the world with online planning evolutionary algorithms.

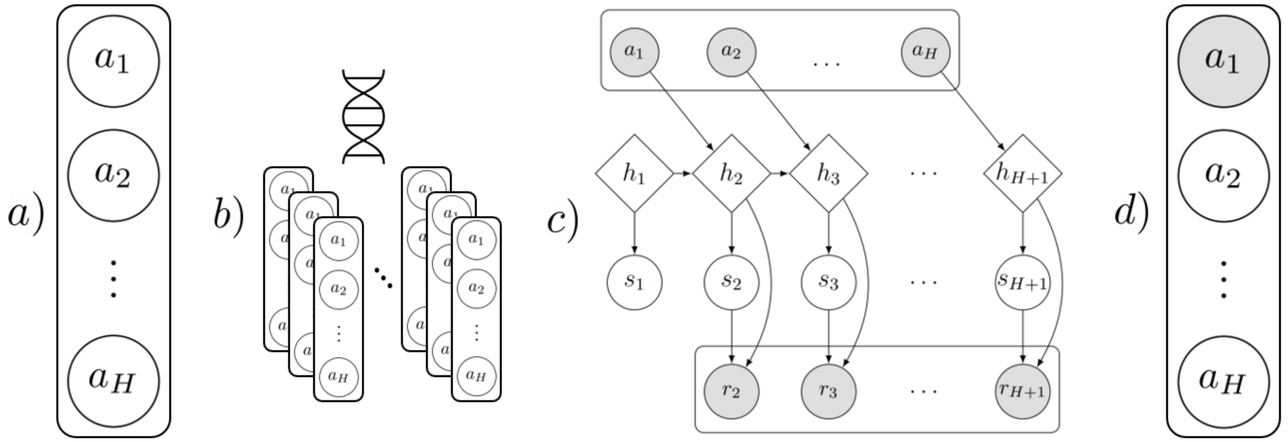
Learning a Model of the World

- The model is learned from raw pixel images with a recurrent state space model (RSSM) (Hafner et al. 2018).
- The RSSM consists of a deterministic and stochastic component.
- The deterministic component allows for the storage, access and transport of information into the future, while the stochastic component permits it to deal with uncertainty and to consider multiple futures.

Rolling Horizon Evolution

- Rolling Horizon Evolution (RHE) (Perez-Liébana et al. 2013) is a family of general real-time planning algorithms with close connections to Model Predictive Control (MPC).
- The idea behind RHE is the application of evolutionary algorithmic techniques to action sequences. It consists in the random generation of N action sequences of length H which then are manipulated by genetic operators (e.g. mutation and crossover). Each of the candidate sequences is executed inside a forward model up to the planning horizon H.
- Although RHE was originally devised and has often been used in conjunction with perfect simulators, it can be adapted to a broader type of situations due to its generality, as long as it is possible to instantiate a forward model to simulate and evaluate the rollouts.

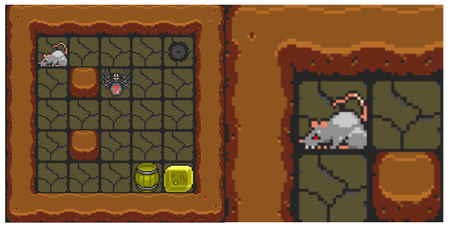
SSM and RHE Integration



Throughout training the model being approximated by the NN architecture interacts with RHE in the following manner:

- A) The planner starts by sampling an action sequence of size H.
- B) Then it applies genetic operators to the action sequence. In this case we opt for a simple mutation strategy to generate N different action sequences.
- C) Using the approximate model, the agent simulates trajectories in latent space by executing each of the candidate action sequences. The rewards for every step of the plan are predicted by passing both the deterministic (h) and stochastic (s) states through a neural network. The reward sequences corresponding to each of the plans are gathered and evaluated.
- D) RHE selects the best action sequence out of the candidates and executes the first action in the actual environment.

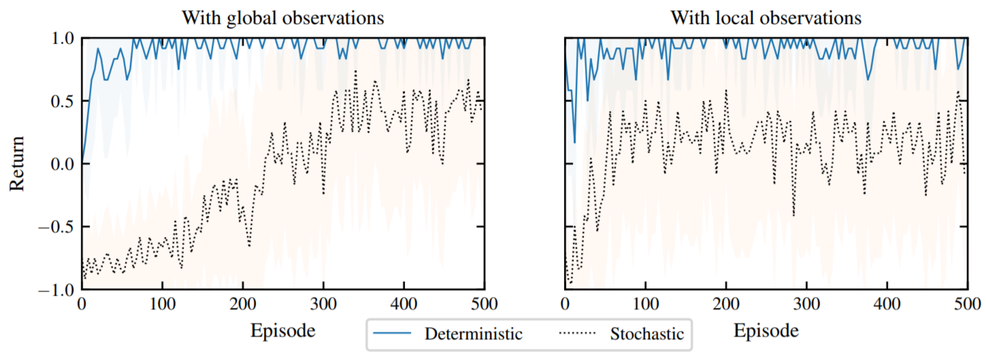
Case Study



We verify the capacity of the agent to learn a forward model while simultaneously being able to search, select and execute adequate plans in a top-down grid-based visual task.

We conduct four variations of the experiment testing the agent's ability to deal with stochastic elements or when its observations are constrained to a local region of the environment.

Case Study



Performance in the task during 500 episodes. The plots show mean and standard deviation over three seeds. The left shows the performance of the agent when it observes the whole grid, while on the right the agent only observes its local neighborhood. Solid and dashed lines represent the performance of the agent in deterministic and stochastic environments respectively.