Toward Imagination-assisted Deep Reinforcement Learning for Human-robot Interaction

Mohammad Thabet[1], Massimiliano Patacchiola[2], and Angelo Cangelosi[1]

Abstract— **Deep reinforcement learning has proven to be a great success in allowing agents to learn complex tasks. However, its application to actual robots can be prohibitively expensive. Furthermore, the unpredictability of human behavior in human-robot interaction (HRI) tasks can hinder convergence to a good policy. This paper proposes an architecture that allows agents to learn models of stochastic environments and use them to accelerate learning. The models can be used to generate imaginary rollouts that can supplement or even replace real interactions. We demonstrate our architecture on a simulated HRI task in which an agent has to respond to random human orders.**

I. INTRODUCTION

Deep reinforcement learning (RL) has been applied successfully to a variety of problems recently such as playing Atari games at super-human level [1], and for robot control [2]. However, Applying RL methods to real robots can be extremely costly, since acquiring thousands of episodes of interactions with the environment often requires a lot of time, and can lead to physical damage. Furthermore, in human-robot interaction (HRI) scenarios, human actions can be unpredictable, which can significantly impede convergence to a good policy.

One way of alleviating these problems is to have the agent learn a model of the environment, and use this model to generate synthetic interaction data that can be used in conjunction with real data to train the agent. If such a model is stochastic in nature, then the unpredictability in state changes can be taken into account, thus allowing more natural interaction with humans. Much like how people learn, an agent with a model of its environment can generate imaginary scenarios that can be used to help optimize its performance. This approach has garnered much attention in the field recently, and is sometimes refered to as endowing agents with *imagination* [3], [4], [5].

In this paper we propose an architecture that allows an agent to learn a stochastic model of the environment and use it to learn optimal policies in RL problems. The work most similar to our own is that by Ha and Schmidhuber [5], in which they build models of computer game environments and use them to train agents to play. By contrast, we apply similar techniques on actual HRI scenarios. We demonstrate the feasibility of our architecture on a simulated HRI task in which the agent has to respond to random orders from a human.

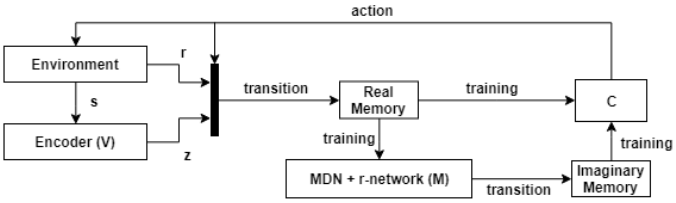

Fig. 1. Overview of the proposed architecture. M can be trained on real transitions and then used to generate imaginary transitions. C can then be trained on both real and imaginary transitions.

II. METHODS

Our proposed architecture consists of three parts: the vision module (V) that produces abstract representations of input images, the environment model (M) which generates imaginary rollouts, and the controller (C) that learns to map states into actions. We assume that the environment is Markovian and is fully represented at any given time by the input image. Figure 1 shows an overview of the architecture.

V comprises the encoder part of a variational auto-encoder (VAE) [6], and is responsible for mapping the high-dimensional input images into low-dimensional state representations. All further processing of the input images are made in this low-dimensional latent space, which is generally computationally less expensive. The reason for using a VAE instead of a vanilla auto-encoder is that the VAE maps input images into a continuous region in the latent space. This makes the environment model more robust and ensures that its output is always meaningful and can be mapped back into realistic images.

M is responsible for generating synthetic transitions, and predicts future states z_{t+1} and the reward r_t based on current states z_t and input actions a_t. it is implemented as a mixture density network (MDN) [7], and learns the conditional probability distribution of the next state $P(z_{t+1}|z_t, a_t)$. The advantage of using an MDN is that it is possible to learn a model of stochastic environments, in which an action taken in a given state can lead to multiple next states. This is especially useful for use in HRI tasks, in which the human response to actions taken by the robot cannot be expected with certainty. The MDN is complemented by a separate model called the r-model that learns the reward for each state-action pair. This model is implemented as a feed-froward neural network. To generate imaginary rollouts, M can be

[1] Mohammad Thabet and Angelo Cangelosi are with the School of Computer Science, University of Manchester, United Kingdom, mohammad.thabet@postgrad.manchester.ac.uk, angelo.cangelosi@manchester.ac.uk

[2] Massimiliano Patacchiola is with the School of Informatics, the University of Edinbrugh, United Kingdom, mpatacch@ed.ac.uk

seeded with an initial state from V, and then run in closed loop where its output is fed back into its input along with the selected action.

Lastly, C is responsible for selecting the appropriate action in a given state. It is implemented as a simple Q-network, and learns to estimate the action values for states.

III. Experiment

To demonstrate the viability of our proposed architecture, we designed a simulated HRI experiment in which the agent learns to pick and place objects as instructed by its human partner. In the experiment, the human starts by pointing at any one of three objects placed on a table, which the agent picks up. The human can then either point at another object at random, at which point the agent has to place the object it currently holds back on the table and pick the new one, or they can request a handover.

The task is formulated as an RL problem in which the agent can choose from 4 discrete actions at any given time: pick/place objects 1, 2, or 3, and perform a handover. The agent gets a reward of +1 for correctly picking up an object, 0 for putting an object back, +5 for correctly handing over, or -5 for choosing an incorrect action. An episode terminates if either a handover is correctly performed, or the agent chooses an incorrect action. The images used in the simulation were taken using the iCub robot.

To train the system, first we trained the VAE on a set of images that includes examples of all possible states the agent might encounter (i.e. different combinations of object places and gestures). The images were scaled down to a manageable 64×64 resolution and compressed into 4 dimensions in the VAE. Afterwards, we trained the controller using the standard Q-learning technique for 1000 episodes, where the states were given by the 4-dimensional output of the encoder part of the VAE. Concurrently, the environment model was trained on transition data collected during these episodes. The MDN model had 48 Gaussian components and was trained for 4 epochs for each episode. The controller converged to an optimal policy after around 500 episodes of training.

To test the environment model, we trained another controller entirely on imaginary data generated by the environment model. It was trained for 1000 episodes and with the same architecture and parameters as the original. During 10 test runs, each with 100 episodes of interaction with the real environment, the controller successfully completed 78% of the episodes on average, compared to the 100% success rate of the original controller. This drop in performance is expected since the environment model is imperfect. Figure 2 shows visualizations of the states imagined by the environment model during one imaginary rollout. The images were created by mapping the output of the model to images using the decoder part of the VAE. It is important here to note that, except for the first image that seeds the model, none of these images are real; they are entirely imagined by the model. They represent what the model thinks is going to happen next given a certain state-action pair. Furthermore, each imaginary rollout results in a different scenario, which reflects the stochastic nature of the environment.

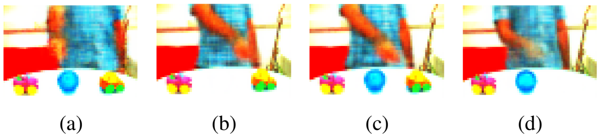
(a) (b) (c) (d)

Fig. 2. A sample imaginary rollout produced by the environment model. The images are visualizations of states imagined by the model. (a), (b), and (c) represent the human asking the agent to pick up object, while (d) represent a request for a handover.

IV. Conclusion

In this paper we presented an architecture that allows an agent to learn a model of stochastic environments in an RL setting. This allows the agent to significantly reduce the amount of interactions it needs to make with the actual environment. This is especially useful for tasks involving real robots in which collecting real data can be expensive. Furthermore, the ability to model stochastic environments makes this approach well-suited for HRI tasks where the actions of humans can be unpredictable. We demonstrated the viability of our architecture in a simulated HRI task, showing how an environment model can be learned and used to generate imaginary rollouts.

In future work, we will explore ways to combine both imaginary and real data to accelerate learning from scratch. Furthermore, the approach will be applied on more complex tasks using real robots. The architecture will be extended to model non-Markovian environments by using recurrent neural networks in the model. We will also investigate using imaginary rollouts for predicting future outcomes and how this information can be used as a sort of lookahead to further accelerate learning.

ACKNOWLEDGMENT

This project has received funding from the European Unions Horizon 2020 framework programme for research and innovation under the Marie Sklodowska-Curie Grant Agreement No.642667 (SECURE)

References

[1] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, “Playing atari with deep reinforcement learning,” *arXiv preprint arXiv:1312.5602*, 2013.

[2] S. Levine, C. Finn, T. Darrell, and P. Abbeel, “End-to-end training of deep visuomotor policies,” *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1334–1373, 2016.

[3] T. Weber, S. Racanière, D. P. Reichert, L. Buesing, A. Guez, D. J. Rezende, A. P. Badia, O. Vinyals, N. Heess, Y. Li, *et al.*, “Imagination-augmented agents for deep reinforcement learning,” *arXiv preprint arXiv:1707.06203*, 2017.

[4] G. Kalweit and J. Boedecker, “Uncertainty-driven imagination for continuous deep reinforcement learning,” in *Conference on Robot Learning*, 2017, pp. 195–206.

[5] D. Ha and J. Schmidhuber, “World models,” *arXiv preprint arXiv:1803.10122*, 2018.

[6] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *arXiv preprint arXiv:1312.6114*, 2013.

[7] C. M. Bishop, “Mixture density networks,” Citeseer, Tech. Rep., 1994.