Dexterous Manipulation by Multi-Task RL and Domain Randomization: Phase 2 Report

Abstract

Learning to control a robot by directly applying model-free Reinforcement Learning (RL) is prone to fail due to extreme sample inefficiency. We propose to address this issue by employing several techniques to improve sample complexity. In simulation we employ reward shaping, multi-task learning, and apprenticeship learning. To transfer the learned policy to the real robot we use domain randomization techniques to improve the robustness of the learned policy. In subsequent phases we plan to use learned domain randomization to target performance on the real system rather than robustness.

1 Learning in Simulation

We propose to use a model-free end-to-end learning based approach to acquire a policy to control the robot in simulation. Our method builds on top of DDPG [Lillicrap et al., 2015] – an off-policy RL algorithm for continuous control. To improve sample efficiency of our learning algorithm we augment canonical DDPG with the following components (1) double clipped Q-learning [Fujimoto et al., 2018], (2) n-step returns, and (3) layer normalization [Ba et al., 2016] after the first layer of the Actor and Critic networks.

Learning desired manipulation behaviors from extremely sparse rewards poses a complex exploration challenge, which we tackle with (1) reward shaping [Popov et al., 2017], (2) multi-task learning [Riedmiller et al., 2018], and (3) apprenticeship learning [Matiisen et al., 2020]. We design a set of auxiliary rewards (tasks) of increasing difficulty, where each reward function is sparse (e.g. it is zero almost everywhere, except for a small region of state space). To enable reward gradient we apply sigmoid-like smoothing on the boundaries of the target region of the state space. We consider the following rewards:

- $r_{\text{reach}} = 1$ if any robot fingertip is very close to the cube, 0 otherwise.
- $r_{\text{grasp}} = 1$ if the cube above the ground, 0 otherwise.
- $r_{\text{move}} = 1$ if the cube is very close to the target location, 0 otherwise.

- $r_{\text{orient}} = 1$ if the cube's orientation is close to the target, 0 otherwise.

2 Domain randomization

To facilitate successful transfer of a policy that we learn entirely in simulation to the real system we have primarily used domain randomization. The goal of domain randomization is to ensure that the learned policy is robust to the differences between the simulator and the real world by covering a broad range of possible worlds in simulation. Following the work of Tobin et al. [2017], Peng et al. [2018], and Andrychowicz et al. [2020] we use a simple domain randomization strategy where we hand-select the parameters to randomize and train in simulators with randomized dynamics. More formally, if we denote the parameters of the simulator by μ and distribution over trajectories induced by μ and a policy π by P_μ^π, we then define a distribution ρ_μ over those parameters and attempt to optimize the expected return in the simulator:

$$\tilde{J}(\pi) = \mathbb{E}_{\mu \sim \rho_\mu} \left[\mathbb{E}_{\tau \sim P_\mu^\pi} \left[\sum_{t=0}^{T-1} r(s_t, a_t) \right] \right]. \tag{1}$$

During training, we run our training algorithm described in our phase 1 proposal which incorporates reward shaping, multi-task learning, and apprenticeship learning, but at each iteration we sample a vector μ of simulator parameters.

Randomizations. We randomize the following parameters of the simulator, always using a uniform distribution over the parameter range unless otherwise indicated:

- Camera rate: 9 to 11 frames per second, to randomize the object position latency.
- Time step: 0.003 to 0.005 seconds, to randomize the time discretization in the simulator.
- Cube position: at each frame we add gaussian noise to the cube position with standard deviation 0.001 to simulate the noisy pose estimation.
- Cube mass: 0.91 to 0.97
- Gravity: -9.83 to -9.79
- Restitution: 0.77 to 0.83
- Lateral friction: 0.09 to 0.11
- Max velocity: 9.9 to 10.1

We found the camera rate and time step to be especially important to randomize.

3 Design decisions

When moving from the simulator to the real robot, there were also several lower level design decisions that we found to be important and that we will enumerate in this section.

Specifically, we employ action repeat of 100 to compensate for lower 10Hz frequency of the object detection system. With this in mind, we limit each training episode by 10k true environment steps (e.g. the policy will only perform 100 actions during an episode). We also randomize initial position of the robot's fingers in simulation to better match the real robot.

4 Next steps: learning the randomization

While we did not have enough time during this phase of the competition, in the next phase we plan to implement more sophisticated learned randomization and expect this to improve our performance. Our current approach simply randomizes the parameters in hopes of covering the dynamics of the true system. By learning a distribution over simulator parameters based on data from the real system we hope to tailor our randomization more specifically to the prescribed tasks. Following the work of Chebotar et al. [2019], Akkaya et al. [2019], and Peng et al. [2020], we plan to parameterize a distribution over the simulation parameters. We will then automatically tune the distribution over the environments as is done in those papers to increase the likelihood of sampling simulator parameters that are similar to the real system. We expect that this additional tuning can improve performance on the real system.

References

Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016.

Yevgen Chebotar, Ankur Handa, Viktor Makoviychuk, Miles Macklin, Jan Issac, Nathan Ratliff, and Dieter Fox. Closing the sim-to-real loop: Adapting simulation randomization with real world experience. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8973–8979. IEEE, 2019.

Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, pages 1582–1591, 2018.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Tambet Matiisen, Avital Oliver, Taco Cohen, and John Schulman. Teacher–student curriculum learning. *IEEE Transactions on Neural Networks and Learning Systems*, 31(9):3732–3740, Sep 2020. ISSN 2162-2388. doi: 10.1109/tnnls.2019.2934906. URL `http://dx.doi.org/10.1109/TNNLS.2019.2934906`.

Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 1–8. IEEE, 2018.

Xue Bin Peng, Erwin Coumans, Tingnan Zhang, Tsang-Wei Lee, Jie Tan, and Sergey Levine. Learning agile robotic locomotion skills by imitating animals. *arXiv preprint arXiv:2004.00784*, 2020.

Ivaylo Popov, Nicolas Heess, Timothy Lillicrap, Roland Hafner, Gabriel Barth-Maron, Matej Vecerik, Thomas Lampe, Yuval Tassa, Tom Erez, and Martin Riedmiller. Data-efficient deep reinforcement learning for dexterous manipulation, 2017.

Martin A. Riedmiller, Roland Hafner, Thomas Lampe, Michael Neunert, Jonas Degrave, Tom Van de Wiele, Volodymyr Mnih, Nicolas Heess, and Jost Tobias Springenberg. Learning by playing - solving sparse reward tasks from scratch. *CoRR*, abs/1802.10567, 2018. URL `http://arxiv.org/abs/1802.10567`.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 23–30. IEEE, 2017.