Evaluating Hierarchical Deep Learning Methods For Vision Based Robot Manipulation Of Dynamic Objects In Unstructured Environments

Department for Interdisciplinary Research
College of Engineering Trivandrum

November 18, 2019

Guided by,
Linu Shine
Electronics and Comm. Engg.

Presented by,
Sreejith Krishnan R
Robotics & Automation

Motivation

Why robot manipulation?

- In order to assist in general tasks, autonomous robots should be able to interact with dynamic objects in unstructured environments
- Designing machines that can grasp and manipulate objects with anything approaching human levels of dexterity is first on the to-do list for robotics [1]

Hierarchical reinforcement learning allows to,

- Decompose complex task into hierarchy of sub-tasks
- Reuse sub-tasks in similar domain
- DNA for artificial intelligent agents

" Stop learning tasks, start learning skills " (Satinder Singh)

Literature survey I

Title	Learning ambidextrous robot grasping policies [2] - Science Robotics - 2019
Methodology	Learns policies on synthetic training datasets generated using analytic models and domain randomization over a diverse range of objects, cameras, and parameters of physics for robust transfer from simulation to reality
Merits	• Current state of the art in object picking • 95% success rate in real robot • 300 mean pick per hour
Demerits	• Only for object picking

Literature survey II

Title	Comparing Task Simplifications to Learn Closed-Loop Object Picking Using Deep Reinforcement Learning [3] - IEEE Robotics and Automation Letters - 2019
Methodology	Uses autoencoder to reduce dimensionality of camera data which is given to 3 layer CNN to get a low dimensional encoding. This encoding is used by a 2 layer feed-forward neural network to predict the optimum action. Uses RL to train the mentioned networks
Merits	• No hand labeled data required
Demerits	• Low success rate (78%) for manipulation of objects in clutter by real robot • Non modular. Difficult to reuse model for similar task

Literature survey III

Title	Regularized Hierarchical Policies for Compositional Transfer in Robotics [4] - DeepMind - 2019
Methodology	Use hierarchical modular policies for continuous control.
Merits	• Best sample efficiency on both simulated and real robot • Uses MPO optimization algorithm which reduces the number of hyperparameters
Demerits	• High level tasks are not automatically decomposed to sub tasks • Low level policy is shared across all low level tasks making interpretability complicated • Transferring specific skills from sub-tasks policy in a predictable manner is difficult • Experiment results are obtained using model whose inputs include pose of objects in workspace

Research Gap

- Low success rate on real world
- High training time due to low data efficiency
- Difficult to apply skills learned from one task to similar tasks

Objectives

Phase 1

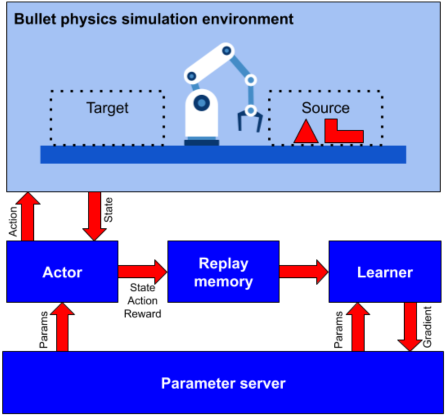
- Evaluate HRL for table clearing task on simulated environment
- Apply learned policies from simulated to real environment. Explore strategies to close the gap between success rate in simulation and real world

Phase 2

- Explore methods to interpret skills learned by HRL
- Evaluate reusability of skills learned from table clearing task to similar tasks. Explore strategies for optimum reusability
- Explore strategies to improve mean picks per hour

Methodology I

- Options framework, MAXQ and hDQN HRL methods are considered for evaluation
- Evaluation will be based on success rate, mean picks per hour and training time
- Simulation environment will be created on Bullet Physics Simulator
- URDF model of ABB IRB 120 robot with 2 finger parallel gripper, table and objects should be provided to simulator



- 3D model of ABB IRB 120 robot is available. URDF file can be created using this models

Methodology II

- 3D CAD model of gripper of ABB IRB robot in robotics lab can be developed and URDF model can be created
- URDF database of 1000 random generated objects for robot grasping is available [5]
- ShapeNetSem [6] contains 12,000 3D models spread across 270 object categories annotated with real-world dimensions and weight. URDF files can be automatically generated from 3D models
- Camera mounted on end effector of robot will give visual input to models. Learning phase will find the optimum action to take for the given visual input. Learned actions will be according to the value function tuned to optimize required parameters like manipulation time. Action will be of format $\begin{bmatrix} dx & dy & dz & d\alpha & d\beta & d\gamma & open \end{bmatrix}$.

Methodology III

- GORILA architecture [7] will be used to reduce training time. Simulator, actor, learner and parameter server will be separate processes interconnected by ZeroMQ [8] which can be distributed across multiple compute nodes.
- Preemptible VMs from Google Cloud [9] ($0.006655 / vCPU hour and $0.000892 / GB hour) can be used as compute nodes. NVIDIA Tesla K80 with 12 GB RAM costs $0.135 / hour can be used when GPU is more effective for computation (eg:- hDQN).
- Policies learned from simulator will be evaluated on ABB IRB 120 robot. Depth camera mounted on endeffector will provide visual feedback. Camera will be directly connected to computer. Program running in the computer will connect with robot via socket created in RAPID [10] program running on robot controller. Robot and computer should be on same network. RAPID program will read

actions send from computer, execute it and send feedback to computer.

References I

Richard Hodson.
How robots are grasping the art of gripping.
https://www.nature.com/articles/d41586-018-05093-1.
Accessed: 2019-09-24.

Jeffrey Mahler, Matthew Matl, Vishal Satish, Michael Danielczuk, Bill DeRose, Stephen McKinley, and Ken Goldberg.
Learning ambidextrous robot grasping policies.
Science Robotics, 4(26):eaau4984, 2019.

M. Breyer, F. Furrer, T. Novkovic, R. Siegwart, and J. Nieto.
Comparing task simplifications to learn closed-loop object picking using deep reinforcement learning.
IEEE Robotics and Automation Letters, 4(2):1549–1556, April 2019.

References II

Markus Wulfmeier, Abbas Abdolmaleki, Roland Hafner, Jost Tobias Springenberg, Michael Neunert, Tim Hertweck, Thomas Lampe, Noah Siegel, Nicolas Heess, and Martin A. Riedmiller.
Regularized hierarchical policies for compositional transfer in robotics.
CoRR, abs/1906.11228, 2019.

Google.
Procedurally generated random objects.
`https://sites.google.com/site/brainrobotdata/home/models`.
Accessed: 2019-09-24.

References III

Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu.
ShapeNet: An Information-Rich 3D Model Repository.
Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.

Arun Nair, Praveen Srinivasan, Sam Blackwell, Cagdas Alcicek, Rory Fearon, Alessandro De Maria, Vedavyas Panneershelvam, Mustafa Suleyman, Charles Beattie, Stig Petersen, Shane Legg, Volodymyr Mnih, Koray Kavukcuoglu, and David Silver.
Massively parallel methods for deep reinforcement learning.
CoRR, abs/1507.04296, 2015.

References IV

ZeroMQ.
Zeromq.
https://zeromq.org/.
Accessed: 2019-09-24.

Google.
Google cloud platform pricing.
https://cloud.google.com/pricing/list.
Accessed: 2019-09-25.

ABB.
RAPID reference manual.
https://library.e.abb.com/public/688894b98123f87bc1257cc50044e809/Technical%20reference%20manual_RAPID_3HAC16581-1_revJ_en.pdf.
Accessed: 2019-09-24.