QT-Opt: Scalable Deep Reinforcement Learning for Vision-Based Robotic Manipulation

arXiv:1806.10293, Kalashnikov et al, 2018.

Sumamrized by Hyecheol (Jerry) Jang

Department of Computer Sciences
University of Wisconsin–Madison

RL Paper Study, Jun. 29. 2020

Table of Contents

Motivation: Why Robotics + Reinforcement Learning

- Usually, Robots are good at **repetitive tasks** (e.g. Assembly Line)

Motivation: Why Robotics + Reinforcement Learning

- Usually, Robots are good at **repetitive tasks** (e.g. Assembly Line)
- Want to make Robots that **identifies surroundings** and **behave accordingly**, but it is difficult

Motivation: Why Robotics + Reinforcement Learning

- Usually, Robots are good at **repetitive tasks** (e.g. Assembly Line)
- Want to make Robots that **identifies surroundings** and **behave accordingly**, but it is difficult
  - **Deep Learning**
    Provide ability to handling real-world scenarios
  - **Reinforcement Learning**
    Provide ability to make decision in long-term, using previous experiences in complex and robust scenarios

Motivation: Why Robotics + Reinforcement Learning

- Usually, Robots are good at **repetitive tasks** (e.g. Assembly Line)
- Want to make Robots that **identifies surroundings** and **behave accordingly**, but it is difficult
  - **Deep Learning**
    Provide ability to handling real-world scenarios
  - **Reinforcement Learning**
    Provide ability to make decision in long-term, using previous experiences in complex and robust scenarios
- Combining two techniques
  - Able to learn policy continuously from their experience
  - No need for manual engineering, use data they collects

Motivation: Difficulites of Using RL in Robotics

- Varience in **visual and physical property of objects**

Motivation: Difficulites of Using RL in Robotics

- Varience in **visual and physical property of objects**
  - Hardness of object (Soft or Hard)
  - Surface Characteristics (Slippery, Sticky, . . .)
  - Color Variation
  - Shape Variation
  - . . .

Motivation: Difficulites of Using RL in Robotics

- Varience in **visual and physical property of objects**
  - Hardness of object (Soft or Hard)
  - Surface Characteristics (Slippery, Sticky, ...)
  - Color Variation
  - Shape Variation
  - ...
- **Noise** of sensors

Motivation: Difficulites of Using RL in Robotics

- Varience in **visual and physical property of objects**
  - Hardness of object (Soft or Hard)
  - Surface Characteristics (Slippery, Sticky, . . .)
  - Color Variation
  - Shape Variation
  - . . .
- **Noise** of sensors

$\Rightarrow$ Still hard to handle though we have sufficiently large training set

$\Rightarrow$ Collecting those training set is expensive (real experiments)

Motivation: Previous Works

- Focused on learning narrow, individual tasks
  - hitting a ball
  - opening door
  - throwing objects
  - ...

Motivation: Previous Works

- Focused on learning narrow, individual tasks
  - hitting a ball
  - opening door
  - throwing objects
  - ...

$\Rightarrow$ Use **Grasping** to achieve *generalization*

Motivation: Previous Works

- Focused on learning narrow, individual tasks
  - hitting a ball
  - opening door
  - throwing objects
  - ...

  ⇒ Use **Grasping** to achieve *generalization*

- Approached the grasping task as predicting a *grasp pose*
  1. Observe the scene (*Normally, using a depth camera*)
  2. Choose best location to grasp
  3. Reach the location (open-loop setting)

Motivation: Previous Works

- Focused on learning narrow, individual tasks
  - hitting a ball
  - opening door
  - throwing objects
  - ...

$\Rightarrow$ Use **Grasping** to achieve *generalization*

- Approached the grasping task as predicting a *grasp pose*
  1. Observe the scene (*Normally, using a depth camera*)
  2. Choose best location to grasp
  3. Reach the location (open-loop setting)

  - Different with how humans and animals behave
  - Grasp is a **dynamical process** that sence and control at each stage

Motivation: Previous Works

- Focused on learning narrow, individual tasks
  - hitting a ball
  - opening door
  - throwing objects
  - ...

$\Rightarrow$ Use **Grasping** to achieve *generalization*

- Approached the grasping task as predicting a *grasp pose*
  1. Observe the scene (*Normally, using a depth camera*)
  2. Choose best location to grasp
  3. Reach the location (open-loop setting)

  - Different with how humans and animals behave
  - Grasp is a **dynamical process** that sence and control at each stage

$\Rightarrow$ **Where this researches start!!**

Table of Contents

Goal

Use Reinforcement Learning with Deep Neural Network
to **perform pre-grasp manipulation**,
response to dynamic disturbances,
and **learn grasping in a generic framework**
that makes minimal assumptions about the task

Goal: Constraint/Condition + Literature Review

- **Closed-loop condition** (With feedback, *Morrison, et al.*)
  - For the other papers work on closed-loop grasping, they deals with servoing problems.
  - This paper focuses on making generalized RL algorithm
  - In practice, it makes Kalashnikov et al.'s method (this method) to autonomously acquire complicated grasping strategy

Goal: Constraint/Condition + Literature Review

- **Closed-loop condition** (With feedback, *Morrison, et al.*)
  - For the other papers work on closed-loop grasping, they deals with servoing problems.
  - This paper focuses on making generalized RL algorithm
  - In practice, it makes Kalashnikov et al.'s method (this method) to autonomously acquire complicated grasping strategy
- **Self-supervised** learning task
  - Compare to prevoius work(by Zeng et al.), Kalashnikov et al. utilize more general action space
  - Actions consist of end-effector **Cartesian motion** and **gripper opening/closing**

Goal: Constraint/Condition + Literature Review

- **Closed-loop condition** (With feedback, *Morrison, et al.*)
  - For the other papers work on closed-loop grasping, they deals with servoing problems.
  - This paper focuses on making generalized RL algorithm
  - In practice, it makes Kalashnikov et al.'s method (this method) to autonomously acquire complicated grasping strategy
- **Self-supervised** learning task
  - Compare to prevoius work(by Zeng et al.), Kalashnikov et al. utilize more general action space
  - Actions consist of end-effector **Cartesian motion** and **gripper opening/closing**
- Observation comes from **a single RGB camera** over the sholder
  - Many current grasping system utilizes depth sensing
  - Using wrist-mounted cameras

Table of Contents

Overview of Model Architecture: MDP

- General Formulation of Robotic Manipulation:
  Based on **Markov Decision Process (MDP)**
  - partially observed formulation (POMDP) is more general.
  - However, assuming current observation contains all necessary information for this task, it is sufficient to use MDP.

Overview of Model Architecture: MDP

- General Formulation of Robotic Manipulation:
  Based on **Markov Decision Process (MDP)**
  - partially observed formulation (POMDP) is more general.
  - However, assuming current observation contains all necessary information for this task, it is sufficient to use MDP.
- MDP have a **general and powerful formalism** for decision making problems.
  However, it is **hard to train**

Overview of Model Architecture: MDP

- General Formulation of Robotic Manipulation:
  Based on **Markov Decision Process (MDP)**
  - partially observed formulation (POMDP) is more general.
  - However, assuming current observation contains all necessary information for this task, it is sufficient to use MDP.
- MDP have a **general and powerful formalism** for decision making problems.
  However, it is **hard to train**
- For each step of MDP:
  1. Observes Image from robot's camera (see Fig. 1)
  2. choose a gripper command, Reward:
     - failed grasp: reward of 0
     - successful grasp: reward of 1
       Defined *success* when the robot holds the object above a certain height

Overview of Model Architecture: MDP

- General Formulation of Robotic Manipulation:
  Based on **Markov Decision Process (MDP)**
  - partially observed formulation (POMDP) is more general.
  - However, assuming current observation contains all necessary information for this task, it is sufficient to use MDP.
- MDP have a **general and powerful formalism** for decision making problems.
  However, it is **hard to train**
- For each step of MDP:
  1. Observes Image from robot's camera (see Fig. 1)
  2. choose a gripper command, Reward:
     - failed grasp: reward of 0
     - successful grasp: reward of 1
       Defined *success* when the robot holds the object above a certain height

Overview of Model Architecture: MDP

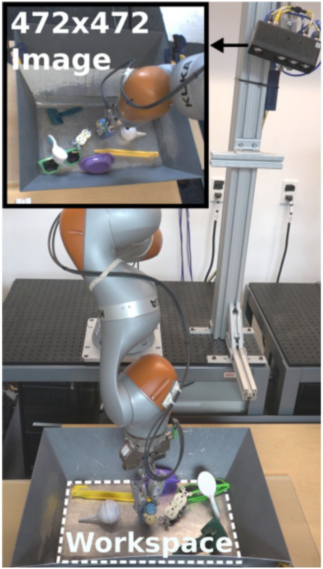

Figure 1: Configuration of robot cell, with a sample observation image on top-right box

Overview of Model Architecture: Algorithm Selection

- Usually, Generalization needs diverse data
  - However, recollecting experience on numerous objects after every policy update is impractical
  - Reason for **not using on-policy algorithm**

Overview of Model Architecture: Algorithm Selection

- Usually, Generalization needs diverse data
  - However, recollecting experience on numerous objects after every policy update is impractical
  - Reason for **not using on-policy algorithm**
- Using **scalable off-policy algorithm** based on Q-learning
  - actor-critic algorithm are popular for handling continuous actions
  - However, Kalashnikov et al. found **scalable and more stable ways** to train only Q-function

Overview of Model Architecture: Algorithm Selection

- Usually, Generalization needs diverse data
  - However, recollecting experience on numerous objects after every policy update is impractical
  - Reason for **not using on-policy algorithm**
- Using **scalable off-policy algorithm** based on Q-learning
  - actor-critic algorithm are popular for handling continuous actions
  - However, Kalashnikov et al. found **scalable and more stable ways** to train only Q-function
- Large Dataset and Network (See Fig. 2)
  - Kalashnikov et al. devised **distributed** training system (with 7 robots)
  - **Asynchronously update** target values, collect **on-policy data**, reloads **off-policy data** from previous experiences, and train network on both data stream.

Overview of Model Architecture: Algorithm Selection

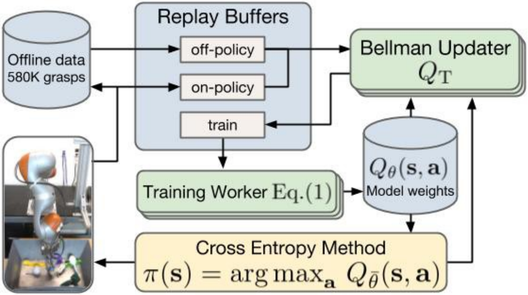

Figure 2: Distributed Reinforcement Learning infrastructure for QT-Opt.

Table of Contents

References

- Kalashnikov, Dmitry, et al. QT-Opt: Scalable Deep Reinforcement Learning for Vision-Based Robotic Manipulation. 28 Nov. 2018, arxiv.org/abs/1806.10293.
- Irpan, Alex, and Peter Pastor. Scalable Deep Reinforcement Learning for Robotic Manipulation. 28 June 2018, ai.googleblog.com/2018/06/scalable-deep-reinforcement-learning.html.
- Morrison, Douglas, et al. "Closing the Loop for Robotic Grasping: A Real-Time, Generative Grasp Synthesis Approach." Robotics: Science and Systems XIV, 2018, doi:10.15607/rss.2018.xiv.021.