Reinforcement Learning for Vision-based Robot Grasping

Project 15
Outline

• Introduction
• Method
• Experiments
• Result & Analysis
• QA
Introduction

RL in Vision-based Grasping

• **Challenges in vision-based robot grasping.**
  - Complex Visual Environments
  - Object Recognition and Localization
  - Depth Perception
  - Handling Unknown or Unseen Objects

• **The advantage of RL over traditional methods**
  - Adaptability to New Scenarios
  - Learning from Interaction
  - Handling High-Dimensional Data
  - Continuous Improvement
  - Decision-Making in Complex Environments
## RL Approach vs. Traditional Methods in Robotic Grasping

<table>
<thead>
<tr>
<th>Learning Methodology</th>
<th>Adaptability and Flexibility</th>
<th>Handling Complex Data</th>
<th>Efficiency and Performance Over Time</th>
<th>Scalability and Generalization</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Traditional Methods</strong></td>
<td>• Often rely on deterministic algorithms.</td>
<td>• Limited adaptability to changes in the environment or task.</td>
<td>• Limited in processing complex, high-dimensional data like real-time visual inputs</td>
<td>• Scaling to different tasks can require significant reprogramming.</td>
</tr>
<tr>
<td></td>
<td>• Require extensive programming and predefined rules.</td>
<td></td>
<td>• May require simplification of the environment for effective operation</td>
<td>• Limited in generalizing learned behavior to different tasks.</td>
</tr>
<tr>
<td></td>
<td>• Limited adaptability to changes in the environment or task.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>RL Approach</strong></td>
<td>• Employs a stochastic, trial-and-error learning process.</td>
<td>• Highly adaptable to new and changing environments</td>
<td>• Performance improves with experience and data accumulation.</td>
<td>• Better at scaling and transferring knowledge to different tasks.</td>
</tr>
<tr>
<td></td>
<td>• Learns from interactions with the environment.</td>
<td>• Capable of generalizing learning to handle unseen situations</td>
<td>• Can generalize strategies across various scenarios.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Continuously adapts and optimizes strategies based on feedback.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Traditional Methods
- Perform well in structured, unchanging environments.
- Struggle with variability and unpredicted scenarios.
- Limited adaptability to changes in the environment or task.

### RL Approach
- Highly adaptable to new and changing environments.
- Capable of generalizing learning to handle unseen situations.

### Handling Complex Data
- Particularly effective in dealing with high-dimensional sensory data.
- Leverages advanced neural networks for deep learning from complex inputs.

### Efficiency and Performance Over Time
- Consistent performance in known conditions.
- Performance does not improve over time unless reprogrammed.

### Scalability and Generalization
- Better at scaling and transferring knowledge to different tasks.
- Can generalize strategies across various scenarios.
Objectives of RL-based Grasping

• **Enhancing Manipulation Skills**: Enhancing Robotic Grasping with Reinforcement Learning. Enhanced complexity, adaptability, and success in grasping tasks.

• **Learning and Adaptation**: Enable robots to learn grasping techniques incrementally and apply them to unfamiliar objects and situations.

• **Overcoming Programming Limitations**: Shift from hardcoded instructions to a model where robots learn autonomously from their environment.
Method

We focus on exploring two of the prominent reinforcement learning algorithms: the Deep Q-Network (DQN) and the Branch Dueling Q-Network (BDQ)

- **DQN**: Learn optimal policies directly from high-dimensional sensory inputs
- **BDQ**:
  - Splits neural network into two: one estimating the state value, the other estimating the advantage of each action.
  - Allows more nuanced decision-making
  - Helps the robot to distinguish between the value of the state and the value of the actions available
  - Crucial in grasping tasks: efficiency of actions can vary significantly based on the context.
Method

We focus on exploring two of the prominent reinforcement learning algorithms: the Deep Q-Network (DQN) and the Branch Dueling Q-Network (BDQ).

- **State**: View of the environment from the eyes of the actor.
- **Reward**: \( r = (\text{grasp detected}) \cdot (r_g + c \cdot \Delta h) - r_{tp} \)
- **Action**: the agent acts on the environment at time \( t \), and the environment returns the reward and state of the action at time \( t + 1 \).

Terminal Reward: \( r_t = 10000 \)
Grasping Reward: \( r_g = 100 \)
\( \Delta h \) Scale Coefficient: \( c = 1000 \)
Time Penalty: \( r_{tp} = 200 \)
Method: Deep Q-Network (DQN) in Robotic Grasping

- **Basics of DQN**: A reinforcement learning algorithm that enables robots to make decisions based on environmental feedback.
- **Application in Grasping**: DQN helps robots analyze and learn from each interaction, progressively improving their ability to grasp and manipulate objects.

\[
L(\theta) = E_{(s,a,r,s') \sim U(D)} \left[ \left( r + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta) \right)^2 \right]
\]
Method: Branch Dueling Q-Network (BDQ)

- **Problem:** Both tabular Q-learning and DQN have been shown to suffer from the overestimation of the action values
- **BDQ:** a branching variant of the Dueling Double Deep Q-Network (Dueling DDQN)
  - **Shared Decision Module:** computes a latent representation, used for evaluation of the state value and the factorized (state-dependent) action advantages
  - **Aggregation Layer:** Combine the state value and factorized advantages to output the Q-value

\[
L = \mathbb{E}_{(s,a,r,s') \sim D} \left[ \frac{1}{N} \sum_d (y_d - Q_d(s,a_d))^2 \right]
\]

Experiments

- Experimental setup
  - Dataset: pybullet-data
  - Robot Model: WSG50
  - Simulator: PyBullet
  - Framework: Gym
  - RL Library: stable-baselines
  - Evaluation Metrics: success-rate
Experiments
Results & Analysis

- **Goal:** how the neural network size affects the success rate
- **Method:** trained each model with varying action dimension padding 8 and 33
- **Conclusion:** model with smaller pads has a higher success rate on both training and testing set.
- **Analysis:**
  - **Exploration Challenges:** If the action space is larger, the agent may face more difficulty in effective exploration during the learning process.
  - **Curse of Dimensionality:** As the action space grows, the complexity of the problem may increase, making the learning process more challenging.

<table>
<thead>
<tr>
<th>Models</th>
<th>Training Set (%)</th>
<th>Testing Set (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DQN_8pads</td>
<td>75%</td>
<td>65%</td>
</tr>
<tr>
<td>DQN_33pads</td>
<td>58%</td>
<td>49%</td>
</tr>
<tr>
<td>BDQ_8pads</td>
<td>91%</td>
<td>90%</td>
</tr>
<tr>
<td>BDQ_33pads</td>
<td>31%</td>
<td>26%</td>
</tr>
</tbody>
</table>
Results & Analysis

DQN vs BDQ:
- Compare the best models of DQN and BDQ

Conclusion:
- BDQ: higher success rate
- DQN: converge faster

Further improvement:
- Transfer to a different scene to test the robustness.
- Parameter tuning:
  - Increasing the number of training episodes, adjusting learning rates

<table>
<thead>
<tr>
<th>Models</th>
<th>Training Set (%)</th>
<th>Testing Set (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DQN_8pads</td>
<td>75%</td>
<td>65%</td>
</tr>
<tr>
<td>BDQ_8pads</td>
<td>91%</td>
<td>90%</td>
</tr>
</tbody>
</table>
Thank you!

Q&A