Journal of Artificial Intelligence and Capsule Networks (ISSN: 2582-2012)
www.irojournals.com/aicn/

Using Deep Reinforcement Learning For Robot Arm Control

Kiran G Krishnan

National Institute of Technology Calicut, Kozhikode, India

E-mail: kichukiran97@gmail.com

Abstract

Reinforcement learning is a well-proven and powerful algorithm for robotic arm manipulation. There are various applications of this in healthcare, such as instrument assisted surgery and other medical interventions where surgeons cannot find the target successfully. Reinforcement learning is an area of machine learning and artificial intelligence that studies how an agent should take actions in an environment so as to maximize its total expected reward over time. It does this by trying different ways through trial-and-error, hoping to be rewarded for the results it achieves. The focus of this paper is to use a deep reinforcement learning neural network to map the raw pixels from a camera to the robot arm control commands for object manipulation.

Keywords: Deep reinforcement learning, robot arm, gazebo, deep learning, artificial intelligence

1. Introduction

Reinforcement Learning is incredibly useful for AI because it provides a training mechanism in which the agent can learn from its experience. Furthermore, the current state of Reinforcement Learning is that it can be applied end-to-end to create sophisticated models. In this paper, the researchers use a deep reinforcement learning network for object manipulation. This system is ideal for manipulating objects in real-time. The system can be scaled up to work with multi-camera setups to create 3D representations of objects and environments. A reward function is required to feedback the neural network. A 3-DoF robotics arm is simulated in Gazebo. Gazebo is a 3D physics simulator developed by the Open Source Robotics Foundation. The simulator is primarily intended to be used as a tool for developing and testing software that drives robots, but it can also be used as an educational tool to learn

DOI: https://doi.org/10.36548/jaicn.2022.3.002
Received: 27.06.2022, received in revised form: 23.07.2022, accepted: 09.08.2022, published: 19.08.2022

about robotics [1]. Implementation of gazebo plugin for the arm, operating via a C++ API for the popular PyTorch library for deep learning frameworks [5]. The purpose of this plugin is to demonstrate and explore the feasibility of performing deep learning inference on an actual robot. It provides a new set of APIs for performing general-purpose inference, as well as utilities for executing and saving the results [10].

The introduction of end-to-end approach revealed a new horizon to resolve challenging robotics problems. To solve a 3-DoF robotics arm kinematic control for object manipulation problem using Deep Q-learning network [6].

2. Gazebo Setup

The gazebo is usually an outdoor booth setup with a series of stands in a circular or squared pathway. The Gazebo plugin is very useful when you need to dynamically generate structural geometry for any part of your experience. It gives users the option to choose the level of detail desired for the 3D primitive shapes. Gazebo plugin also contains a pre-made environment of an outdoor playground ready for import, complete with ground and sky texture. 3-DoF robotics arm is one of the common arm that is deployed in the industrial applications.

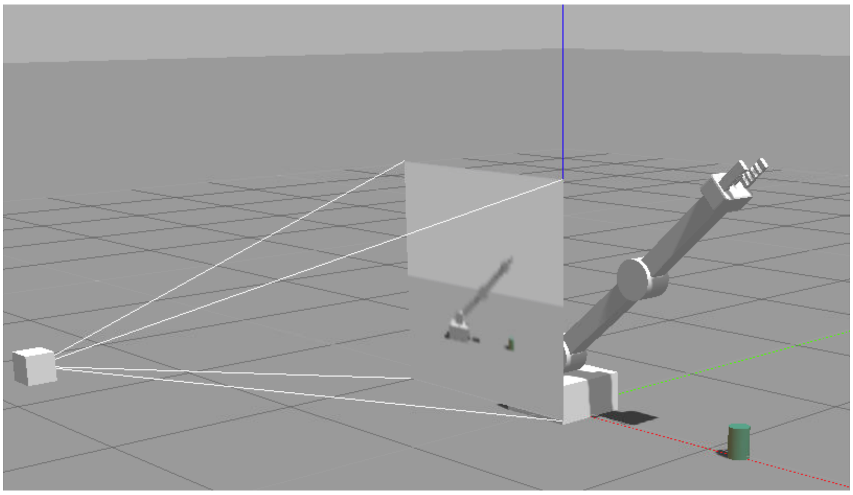

Figure 1. Gazebo 3-DOF robotic arm

3. Control Setup

In this project, there are two options available for the robot arm kinematic control. They are positional control and velocity control [3]. Positional control is much more simpler to use. Positional control is a good way of describing when an object or group of objects is at a certain geometric orientation in relation to the origin point. In other words, how far away the object or group of objects is from your set origin point on the world grid. A robot's positional control will allow it to move its servo motors, and be able to see exactly where all bodily extremities are, thus having full control over all bodily extremities.

Velocity control is important for high-speed situations where it can be near impossible to keep your hands on the two bang sticks if you're using just one hand. When using velocity control, one should have relatively low sensitivity so that inputs don't snowball into uncontrollable trajectories that take valuable time for corrections, killing the flow and resulting in a poor overall performance rating.

Table 1. Discrete Action Encoding

Index	Action
0	J0 -
1	J0 +
2	J1 -
3	J1 +
4	J2 +
5	J2 -

4. Reward Function

Deep Q-Network output is usually mapped to a particular action [4]. The rewards system is designed to train the robot arm to touch the object of interest in one attempt [9]. Each episode is limited to a certain number of attempts, penalty will be issued if maximum length of the episode is reached without achieving the objective [8] [7]. Interim reward or penalty will be issued while the robot arm is moving based on the distance from the object of interest. If the robot arm touch the object of interest, it will issue a win reward and episode is ended.

Table 2. Rewards Function

Event	Rewards
Ground Collision	-1.0
Object Collision	-1.0
End of Episode	-1.0

Success	1.0
Interim Rewards	5.0 x Î”d
Interim Distance Smoothing Factor	0.22

5. Hyper-parameters

The final hyper-parameters configuration is shown in table 3.

Table 3. Training Hyper-parameters

Parameter	Value	Description
Channel	3	Input Image Channels (RGB)
Batch size	32	Input data batch size
Replay memory	10000	Size of Memory buffer for experience replay
Optimizer	Adam	DQN Network Loss Optimizer
Learning rate	0.005	Network Learning Rate
LSTM	False	Flag to enable Recurrent Network (LSTM)
Delta	0.096	Joint angle increment per step in rad (5.5 deg)
EPS decay	400	Epsilon delay steps

5.1 Input Dimensions

The default input width and height parameters are 512x512. It is reduced to 128x128, as the default size was excessively large and did not provide a significant improvement. With this changes, it able to reduce the GPU memory used in training [2].

5.2 Optimizer

The Adam optimizer is widely used in vast number of different domains and the general robustness to un-tuned parameters.

5.3 Learning Rate

The initial guess for the learning rate is 0.01, a lower learning rate will prevent the network to reach plateau.

6. Results

6.1 Task 1

As for task 1, the robot arm need to the touch the object with at least a 90% accuracy for a minimum of 100 runs. After 130 iterations, it manage to achieve more than 90% accuracy. Explain the detail description of proposed method with diagram, algorithm, and flowchart.

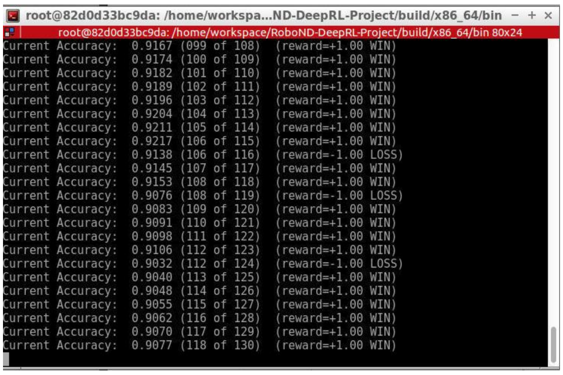

Figure 2. Snapshot log of task 1 during execution

6.2 Task 2

As for task 2, the robot arm's gripper base need to touch the object with at least a 80% accuracy for a minimum of 100 runs. After 500 iterations, it manage to achieve more than 80%.

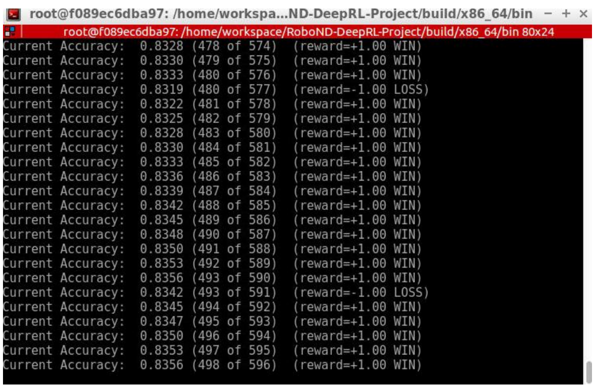

Figure 3. Snapshot log of task 2 during execution

7. Conclusion/Future work

The DQN agent is capable of achieve compelling performance on two of the tasks as demonstrated above. The movement generated by the DQN network is not smooth. If the object of interest is not located in the image, the decision of the DQN network at that instant will become non-deterministic and may result in unstable oscillation.

The following methods can be used to improve the results as a part of future work

- Implement Double Q-Learning
- Scaling rewards
- Resizing input dimensions
- Would a different network configuration work better maybe?
- Yes, using a LSTM architecture. It will require a lot of training set.

Different types of control that haven't tried yet but think could provide improvements?

- Velocity control will make the robot arm movement more unstable
- Positional control is much simpler to use and more stable

References

[1] Ai, Qingsong, Mengyuan Zhao, Kun Chen, Xuefei Zhao, Li Ma, and Quan Liu. "Flexible coding scheme for robotic arm control driven by motor imagery decoding." *Journal of Neural Engineering* (2022).

[2] Azizkhani, Milad, Isuru S. Godage, and Yue Chen. "Dynamic control of soft robotic arm: A simulation study." *IEEE Robotics and Automation Letters* 7, no. 2 (2022): 3584-3591.

[3] Chen, Jong-Chen. "Using Artificial Neuro-Molecular System in Robotic Arm Motion Control—Taking Simulation of Rehabilitation as an Example." *Sensors* 22, no. 7 (2022): 2584.

[4] Deng, Lei, Shuaishuai Sun, Matthew Daniel Christie, Wenxing Li, Donghong Ning, Haiping Du, Shiwu Zhang, and Weihua Li. "Innovative variable stiffness and variable damping magnetorheological actuation system for robotic arm positioning." *Journal of Intelligent Material Systems and Structures* (2022): 1045389X221099453.

[5] Franceschetti, Andrea, Elisa Tosello, Nicola Castaman, and Stefano Ghidoni. "Robotic arm control and task training through deep reinforcement learning." In *International Conference on Intelligent Autonomous Systems*, pp. 532-550. Springer, Cham, 2022.

[6] Li, Nana, Yong Wang, Kun Huang, Pengfei Shen, Shuangfei Li, and Lin Zhou. "A DoS Attacks Detection Aglorithm Based on Snort-BASE for Robotic Arm Control Systems." *Journal of Computer and Communications* 10, no. 4 (2022): 1-13.

[7] Linares-Barranco, Alejandro, Enrique Pinero-Fuentes, Salvador Canas-Moreno, Antonio Rios-Navarro, Chenxi Wu, Jingyue Zhao, D. Zendrikov, and G. Indiveri. "Towards hardware Implementation of WTA for CPG-based control of a Spiking Robotic Arm." *arXiv preprint arXiv:2202.07064* (2022).

[8] Tang, Zhiqiang, Peiyi Wang, Wenci Xin, and Cecilia Laschi. "Learning-Based Approach for a Soft Assistive Robotic Arm to Achieve Simultaneous Position and Force Control." *IEEE Robotics and Automation Letters* 7, no. 3 (2022): 8315-8322.

[9] Xu, Baoguo, Wenlong Li, Deping Liu, Kun Zhang, Minmin Miao, Guozheng Xu, and Aiguo Song. "Continuous Hybrid BCI Control for Robotic Arm Using Noninvasive Electroencephalogram, Computer Vision, and Eye Tracking." *Mathematics* 10, no. 4 (2022): 618.

[10] Yuvaraj, S., Abhishek Badholia, P. William, K. Vengatesan, and Rahul Bibave. "Speech Recognition Based Robotic Arm Writing." In *Proceedings of International Conference on Communication and Artificial Intelligence*, pp. 23-33. Springer, Singapore, 2022.