The Deep Reinforcement Learning Algorithm for Controlling Robotic Arms

Arda Kaya

***Abstract*— In this paper Deep Reinforcement Learning algorithm will be discussed and one of the open-source project Jetson Reinforcement will be demonstrated in Gazebo simulation environment. The main goal of this paper sharing process of using Deep Reinforcement Learning Algorithm to solve these two robotic problems:**

- **Have any part of the robot arm touch the object of interest, with at least a 90 percent accuracy**
- **Have only the gripper base of the robot arm touch the object, with at least a 80 percent accuracy.**

I. INTRODUCTION

The main goal of this paper sharing process of using Deep Reinforcement Learning Algorithm to solve these two robotic problems:

- Have any part of the robot arm touch the object of interest, with at least a 90 percent accuracy
- Have only the gripper base of the robot arm touch the object, with at least a 80 percent accuracy.

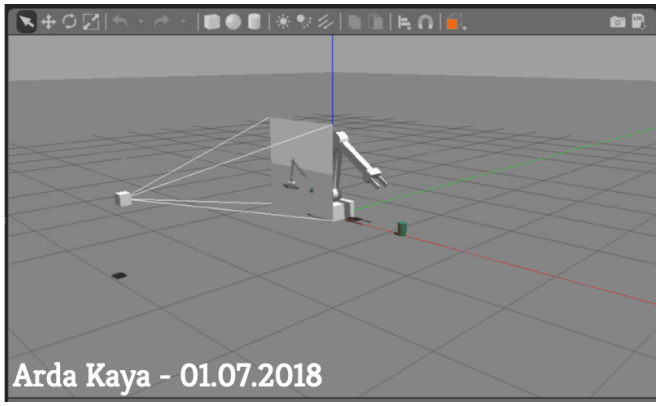

Fig. 1. Gazebo View or the scene

II. BACKGROUND

The main advantage of the Reinforcement Learning algorithm is learning by interactions with its environment. It is not taught by a trainer what to do and it learns what actions to take to get the highest reward in the situation by trial and error, even when the reward is not obvious and immediate. It learns how to solve problems rather than being taught what solutions look like. The template projects is based on Dustin Franklins open source project Jetson Reinforcement.

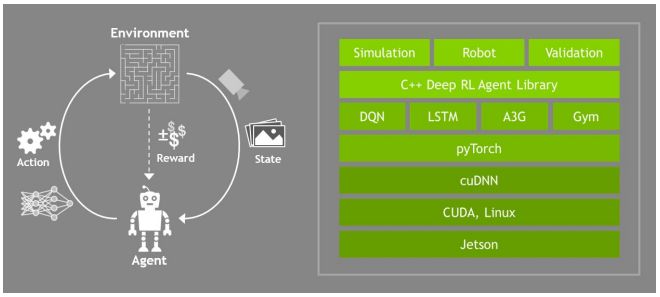

Fig. 2. Jetson Reinforcement Project Diagram

III. REWARD SYSTEM

In the Reinforcement Learning Algorithm there is a set of actions, a set of observations and reward. The goal of this algorithm is finding the most optimized series of actions . To do that algorithm maps history of actions and rewards. Therefore, result of actions decided by a reward.

A. Distance to Goal Rewards

To encourage the robotic arm for getting closer to cylinder a reward system was built. Every frame the distance between the arm gripper and goal measured and averaged. Then, this value multiplied with theREWARD-WIN and REWARD-MULTIPLIER values. Lastly, to stabilize the result Thanh function was used.

B. The First Problem

The first robotic problem needs a reward system that encourages the robotic arm to touch the cylinder by its any part. You can find related values in ArmPlugin.ccp in project link provided earlier.

- REWARD-WIN = 0.13
- REWARD-LOSS = -0.13

If the robotic arm can:

- Touch the cylinder gets 10 * REWARD-WIN
- Not touch the cylinder in limited time gets REWARD-LOSS
- Touch the ground 10 * REWARD-LOSS

C. The Second Problem

The first robotic problem needs a reward system that encourages the robotic arm to touch the cylinder by only its grip part.

- REWARD-WIN = 20
- REWARD-LOSS = -20
- REWARD-MULTIPLIER = 10

If the robotic arm can:

- Touch the cylinder by its grip it gets REWARD-WIN
- Touch the ground REWARD-LOSS
- Touch the cylinder by other parts it gets REWARD-LOSS

IV. Hyperparameters

Same hyper parameters values were used to solve the both problems.

- INPUT-WIDTH 64
- INPUT-HEIGHT 64
- OPTIMIZER ”RMSprop”
- LEARNING-RATE 0.1f
- REPLAY-MEMORY 20000
- BATCH-SIZE 32
- USE-LSTM false

V. Results

A. The First Problem

You can watch the video of the result from this link.

- Total Trial 511
- Accuracy 90

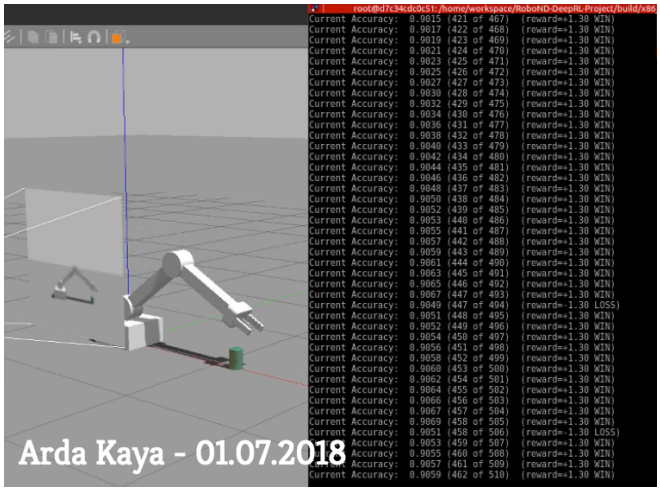

Fig. 3. Result-1

B. The Second Problem

You can watch the video of the result from this link.

- Total Trial 939
- Accuracy 80

VI. Discussion

90 percent accuracy for the first problem and 80 percent accuracy for the second problem has achieved with the Hyperparameters and the Reward System that shared previously.

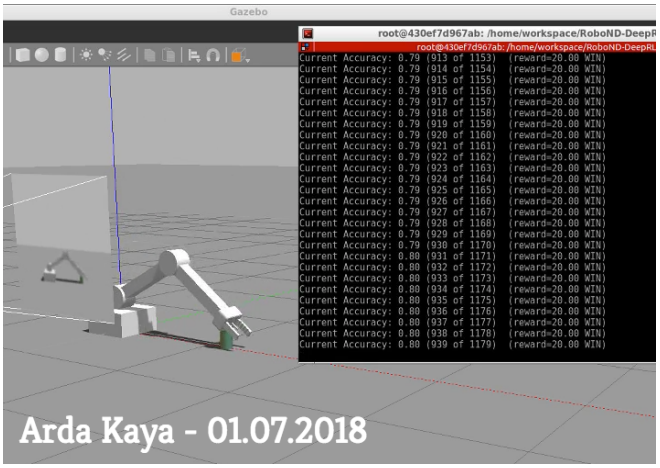

Fig. 4. Result-2

VII. Future Work

- LSTM was closed because it was causing a lot of loss. There will be more trials to adapt it into this project.
- In this paper only one camera has used for deciding next actuation of the robotic arm therefore scene setup is designed for one axis motion. There will be experiment on integrating multiple cameras and modifying scene setup for three axis motions. As an example the target cylinder would appear not just in same place, instead it would appear left or right ramdomly.