Deep RL Arm Manipulation

Balaveera Thatikonda

Abstract—In this project attempted to get 90 percent to touch an object with the gripper of an arm using Deep RL. Created a DQN Agent with the provided frame work and tuned the hyper parameter to get the required accuracy..

Index Terms—Robotics, Deep RL Arm, Reinforcement Learning, Udacity.

1 Introduction

In order for the robot's arm to capture the object of interest, the arm needs to be trained by providing rewards. So Deep RL (Reinforcement Learning) approach is chosen. Started with the provided frame work in project workspace and then created the DQN agent, adjusted the awards while tuning the hyper parameters.

2 Background / Formulation

RL enables the industrial robots to train themselves to perform new tasks, saves money compared to direct programming techniques requiring engineering resources.

2.1 Deep RL

In this Deep RL approach the robot is trained itself to achieve a goal by providing the awards for the positive actions and penalties for the negative actions. The robot will start in an arbitrary position and that image is captured by sensor and analyzed with the goal and accordingly a reward is given if that action is going towards the goal, otherwise a penalty is given. By this way the robot will try to get more rewards to reach the goal and trains itself to do the following two tasks.

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

3 Simulations

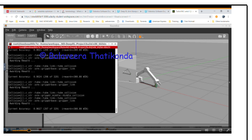
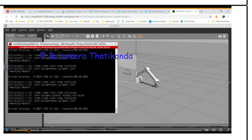
Following are the images of the robot completing the two tasks(refer to Table 1). Started with the provided environment - Deep RL project workspace. Subscribed to the camera and collision topics, created a DQN Agent and assigned the rewards for gripper and arm touching the object. Used the position based mechanism to control the arm joints movement.

4 Set up

4.0.1 Robot

Used the provided robot with an arm, a gripper and an object to touch. Subscribed to the camera(captures the images) and collision topics (captures the which part of arm had touched the object). Created a DQN Agent by passing the hyper parameters constructor with the possible number of actions defined as 2x the number of joints (DOF) as they can rotate both clockwise and counter clockwise. Assign a reward of 300 for touching the object and penalty of -300 if not, here followed the smoothed moving average of the delta of the distance to the goal method. For the movement of the arm used the position based control approach as it is simple and best for the simulations.

TABLE 1
Robot's Arm touching the Object

90 % Accuracy Gripper touching the object	
80 % accuracy arm touching the object	

```
Current Accuracy:  0.9021 (295 of 327)  (reward=+300.00 WIN)

Collision1().c_str :tube::
Collision2().c_str :arm::gripper
 Awarding Reward ...

Collision1().c_str :tube::tube_link::tube_collision
Collision2().c_str :arm::gripperbase::gripper_link
 Awarding Reward ...

Current Accuracy:  0.9024 (296 of 328)  (reward=+300.00 WIN)

Collision1().c_str :tube::tube_link::tube_collision
Collision2().c_str :arm::gripperbase::gripper_link
 Awarding Reward ...

Collision1().c_str :tube::tube_link::tube_collision
Collision2().c_str :arm::gripper_middle::middle_collision
Collision1().c_str :tube::tube_link::tube_collision
Collision2().c_str :arm::gripperbase::gripper_link
 Awarding Reward ...

Current Accuracy:  0.9027 (297 of 329)  (reward=+300.00 WIN)
```

TABLE 2
Hyper Parameters

Parameter Name	Value
INPUT$_WIDTH$	64
INPUT$_HEIGHT$	64
NUM_ACTIONS	DOF2
OPTIMIZER	RMSProp
LEARNING$_RATE$	01f
REPLAY$_MEMORY$	10000
BATCH$_SIZE$	32
USE$_LSTM$	FALSE
LSTM$_SIZE$	256

4.0.2 Packages/Programs used

Packages - Gazebo.
Topics received - camera, collisions
Functions Updated in ArmPlugin.cpp - ArmPlugin::Load(), ArmPlugin::createAgent(), ArmPlugin::onCollisionMsg(), ArmPlugin::updateAgent(), ArmPlugin::onUpdate()
Tuned Hyper Parameters : Input Width and Height, Optimizer, Learning Rate, Batch Size, use LSTM and LSTM size.

4.0.3 Execution

Launched the Robert Arm in the Gazebo

5 Data Acquisition

Allowed the robot to keep trying to touch the object and at around 300+ tries, reaches the goal of the touching the object with 80%

6 Results

The robot arm started touching the object after 40 tries. It acquired the required accuracy after 300 tries of more than 80% and 90% (refer to Table 1).

7 Discussion

After filling up the TODOs, started with given values for INPUT_WIDTH and HEIGHT like 512 and set the Optimizer as RMSProp, had an issue in going forward got memory errors, then reduced to 64 and then it is executing with no errors, but no wins. Tweaked the Rewards for Win (300) and loss(-300) values and used the smoothing average of the delta of the distance to the object for the interim rewards. Slowly increased the learning from 0.001 to 0.1. At this point arm was able to get Wins after 40 episodes but the accuracy is not improving, then increased the batch_size to 32 , after this saw the wins after 40 episodes and accuracy is also increasing, required accuracy around 250+ tries. Then tried by using the LSTM to true and its size to 256, but not able to get even the wins after 100+ tries. So finally arrived at above values as in Table 2 for hyper parameters and got required accuracy reaching the goal.

8 Conclusion / Future work

Achieved the required goal. This package can be deployed for closed environments where there are not much dynamism in the environment, but it can train itself to recognize new objects to pick and place in respective bins.

8.0.1 Future Work

In future, would like to deploy on a moving robot, train in different set of environments and improve it iteratively.
There are multiple location where this robot can be deployed but should be tested for those kinds of environments, like removing the utensils from dishwasher and arranging then on the table, pick and put the toys in respective bins in the toys room, can scan the returned books in a library and can arrange the items(books, toys, etc) cluttered in the home.

9 References

1) Project slack
2) Common Questions in Project