Robotics Project : Deep RL Arm Manipulation

Shotaro Oyama

Abstract—The goal of this project is to create a DQN agent and define reward functions to teach a robotic arm to carry out two primary objectives: First, have any part of the robot arm touch the object of interest, with at least a 90% accuracy. Second, have only the gripper base of the robot arm touch the object, with at least a 80% accuracy.

Index Terms—Deep RL Arm Manipulation, Reinforcement Learning, DQN, Gazebo, Robotics Nanodegree, Udacity

1 Background

DEEP Reinforcement Learning for Robotics is a paradigm shift. The basic idea is to let a robot figure out, through trial and error, the best way to achieve the goal. The learning framework is characterized by an agent learning to interact with its environment. At each time step, the agent receives the environment's state, and the agent must choose an appropriate action in response. One time step later, the agent receives a reward and a new state. All agents have the goal to maximize expected cumulative reward, or the expected sum of rewards attained over all time steps. In this report, with training DQN agent by the information from camera and sensors on the robotic arm, the arm become to be manipulated as the way how it is expected.

2 Rewards

Rewards consists o of the calculation from the result on each time step, and the incident when terminating the trial.

2.1 Terminal step

The reward values provided on each incident are as Table 1.

TABLE 1
Reward on Incident

Incident	Reward Value
Collision on Arm (Target)	+100.0
Collision on Arm (Non-Target)	-10.0
Ground Contact	-100.0
Episode Timeout	-100.0

The trigger on Collision on Arm is changed from every part of the arm (in Task 1) to just the gripper of the arm (in Task 2). It made very difficulf for the arm to meet the objective.

2.2 Each time step

The reward needs to be defined so that the it will encourage the agent to accomplish its task. So the reward is defined based on the distance between gripper and the object. From the result of several trial, the rewards include the Smoothed Moving Average, or SMMA, and the distance itself. By using SMMA, it shapes the reward function and gives the agent gradual feedback to let it know that its doing better as it gets closer to the object of interest, that helps eliminate fluctuations and allows for agent to follow prevailing trend to maximize its rewards, allowing the agent to learn faster. The reason why distance itself was also used as rewards, is that it could make the learning faster. Also, the distance was adjusted a little bit in order for the Arm to get more precise target position. The Reward function is defined as follows:

```
distDelta=lastGoalDistance−distGoal
alpha=0.2f
avgGoalDelta=
  (avgGoalDelta∗alpha)+(distDelta∗(1−alpha))
rewardDist= −pow((1+distGoal),3)/20
rewardHistory=avgGoalDelta∗4+rewardDist
```

The balance of 2 type of rewards, avgGoalDelta and rewardDist was adjusted by using pow function and coefficients, which was tweaked several times especially in Task 2.

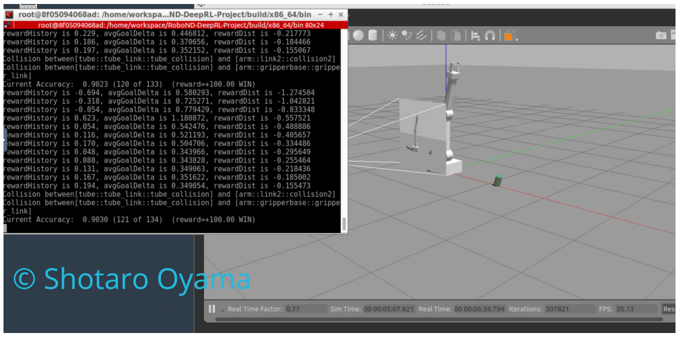
3 Hyperparameters

After tweaking, the same hyperparameters was used in both tasks. The highlight is as follows.

TABLE 2
HyperParameter

Hyperparameter	Value
INPUT WIDTH	64
INPUT HEIGHT	64
OPTIMIZER	RMSprop
LEARNING RATE	0.01f
REPLAY MEMORY	20000
BATCH SIZE	64
USE LSTM	true
LSTM SIZE	256
VELOCITY CONTROL	Off

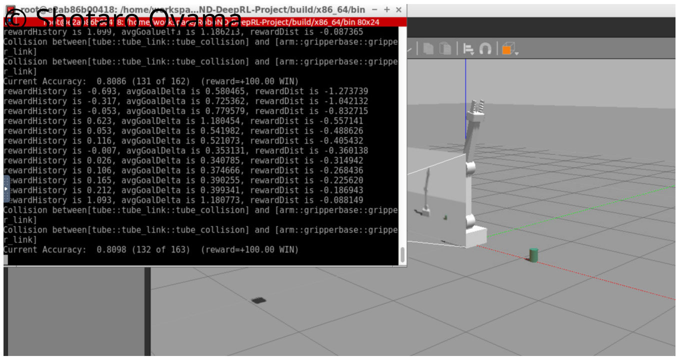
- INPUT WIDTH & HEIGHT was decreased from 512 to 64, to reduce complexity of neural network
- ”RMSprop” OPTIMIZER was selected in spite of ADAM, because of better results



```
root@8f05094068ad: /home/workspace/RoboND-DeepRL-Project/build/x86_64/bin 80x24
rewardHistory is 0.229, avgGoalDelta is 0.446812, rewardDist is -0.217773
rewardHistory is 0.186, avgGoalDelta is 0.370656, rewardDist is -0.184466
rewardHistory is 0.197, avgGoalDelta is 0.352152, rewardDist is -0.155067
Collision between[tube::tube_link::tube_collision] and [arm::link2::collision2]
Collision between[tube::tube_link::tube_collision] and [arm::gripperbase::gripper_link]
Current Accuracy:  0.9023 (120 of 133)  (reward=+100.00 WIN)
rewardHistory is -0.694, avgGoalDelta is 0.580293, rewardDist is -1.274504
rewardHistory is -0.318, avgGoalDelta is 0.725271, rewardDist is -1.042821
rewardHistory is -0.054, avgGoalDelta is 0.779429, rewardDist is -0.833348
rewardHistory is 0.623, avgGoalDelta is 1.180872, rewardDist is -0.557521
rewardHistory is 0.054, avgGoalDelta is 0.542476, rewardDist is -0.488886
rewardHistory is 0.116, avgGoalDelta is 0.521193, rewardDist is -0.405657
rewardHistory is 0.170, avgGoalDelta is 0.504706, rewardDist is -0.334486
rewardHistory is 0.048, avgGoalDelta is 0.343966, rewardDist is -0.295649
rewardHistory is 0.088, avgGoalDelta is 0.343828, rewardDist is -0.255464
rewardHistory is 0.131, avgGoalDelta is 0.349063, rewardDist is -0.218436
rewardHistory is 0.167, avgGoalDelta is 0.351622, rewardDist is -0.185002
rewardHistory is 0.194, avgGoalDelta is 0.349054, rewardDist is -0.155473
Collision between[tube::tube_link::tube_collision] and [arm::link2::collision2]
Collision between[tube::tube_link::tube_collision] and [arm::gripperbase::gripper_link]
Current Accuracy:  0.9030 (121 of 134)  (reward=+100.00 WIN)
```

root@8f05094068ad: /home/workspa...ND-DeepRL-Project/build/x86_64/bin
Real Time Factor: 0.77
Sim Time: 00 00:05:07.821
Real Time: 00 00:06:36.794
Iterations: 307821
FPS: 35.13

Fig. 1. Task1 result



```
root@e2ab86b00418: /home/workspace/RoboND-DeepRL-Project/build/x86_64/bin 80x24
rewardHistory is 1.099, avgGoalDelta is 1.186213, rewardDist is -0.087365
Collision between[tube::tube_link::tube_collision] and [arm::gripperbase::gripper_link]
Collision between[tube::tube_link::tube_collision] and [arm::gripperbase::gripper_link]
Current Accuracy:  0.8086 (131 of 162)  (reward=+100.00 WIN)
rewardHistory is -0.693, avgGoalDelta is 0.580465, rewardDist is -1.273739
rewardHistory is -0.317, avgGoalDelta is 0.725362, rewardDist is -1.042132
rewardHistory is -0.053, avgGoalDelta is 0.779579, rewardDist is -0.832715
rewardHistory is 0.623, avgGoalDelta is 1.180454, rewardDist is -0.557141
rewardHistory is 0.053, avgGoalDelta is 0.541982, rewardDist is -0.488626
rewardHistory is 0.116, avgGoalDelta is 0.521073, rewardDist is -0.405432
rewardHistory is -0.007, avgGoalDelta is 0.353131, rewardDist is -0.360138
rewardHistory is 0.026, avgGoalDelta is 0.340785, rewardDist is -0.314942
rewardHistory is 0.106, avgGoalDelta is 0.374666, rewardDist is -0.268436
rewardHistory is 0.165, avgGoalDelta is 0.390255, rewardDist is -0.225620
rewardHistory is 0.212, avgGoalDelta is 0.399341, rewardDist is -0.186943
rewardHistory is 1.093, avgGoalDelta is 1.180773, rewardDist is -0.088149
Collision between[tube::tube_link::tube_collision] and [arm::gripperbase::gripper_link]
Collision between[tube::tube_link::tube_collision] and [arm::gripperbase::gripper_link]
Current Accuracy:  0.8098 (132 of 163)  (reward=+100.00 WIN)
```

root@e2ab86b00418: /home/workspa...ND-DeepRL-Project/build/x86_64/bin

Fig. 2. Task2 result

- LEARNING RATE was set to 0.01 after tweaking several values especially for achieving the objective of task 2
- LSTM MEMORY, BATCH SIZE, and LSTM SIZE were set as much as possible, taking computer resource into account. It seemed make the Arm working more softly.
- VELOCITY CONTROL was turned off because it added more complexity to the arm, resulted poorer result.

4 Results

Both tasks met the required accuracy. (Table. 2, Fig. 1 and Fig. 2) At Task 1, the arm was able to achieve the objective, almost from the begining. However, at Task 2, the arm had failed many times in the first 10 episodes. Task 2 required more elaborated behavior of the arm, so it seemed taking time to learn that complicated movement. Especially, the angle of the arm when it was close to the object was so important that it could avoid hitting the object by its LINK TOUCH or GRIPPER. This movement seemed to be learned well in larger LSTM size.

TABLE 3
Result Summary

Task	Accuracy
Task1	90.3% (121 of 134)
Task2	80% (132 of 163)

5 Future Work

VELOCITY CONTROL may have possibility to improve the result of this project. At this time it was turned off because of its complexity, but if it is handled correctly, it may be able to realize more complicated robotic arm behavior. Also, assuming in actual use case, the arm should be required to grasp the object. Training will be done y-axis in addition to x-axis and z-axis which were target in this report.