Arm Manipulator with Deep RL
Project Write-up

Jorge Crespo Cedeño

No Institute Given

1 Introduction

This write-up describes the training of a arm manipulator using Deep Reinforcement Learning to touch a tube on the ground. In the first challenge, any part of the arm can touch the tube at least 80% of the times with a minimum of 100 runs. In the second one, only the gripper base can touch the tube 90% of the times with a minimum of 100 runs. In neither of both challenges, the gripper base cannot touch the ground.

2 Reward Functions

The DQN agent chooses an action. This action indicates the joint to be actuated upon and whether to increase or decrease the position of that joint. Alternatively, the joint can be controlled by increasing or decreasing the velocity, but position control is used for this project. The position of the selected joint is increased or decreased by 0.15.

The reward, for the first challenge, is given as follows:

- -0.4 when the gripper base touches the ground
- -0.4 when the number of frames captured by the camera exceed 100.
- +4.0 when any part of the arm touches the tube
- -a value which depends on the speed of the arm towards the tube. This is calculated as the smoothed moving average of the difference between the successive goal distances.

The reward, for the second challenge, is given as follows:

- -0.4 when the gripper base touches the ground
- -0.4 when the number of frames captured by the camera exceed 100.
- +4.0 when the gripper base touches the tube
- -0.4 when any part of the arm but the gripper base touches the ground.
- -a value which depends on the speed of the arm towards the tube. This is calculated as the smoothed moving average of the difference between the successive goal distances.

The magnitude of the negative reward is small compared to the positive one, because very small negative rewards, -4.0 for instance, make the agent hesitant to move since it tries to avoid a big penalty when not touching the tube. The positive reward is big compared to the magnitude of the negative reward in order to give a big incentive to the agent to learn from a successful arm movement.

3 Hyperparameters

The hyperparameters are the same for both objectives.

Table 1. Hyperparameters.

Hyperparameter	Value	Reason
INPUT_WIDTH	64	The camera frames are 64 pixels wide
INPUT_HEIGHT	64	The camera frames are 64 pixels high
OPTIMIZER	RMSprop	Root mean square prop
LEARNING_RATE	0.025	A small value to avoid overshooting the minimum
REPLAY_MEMORY	1000	The amount of past experiences to store
BATCH_SIZE	16	Number of transitions randomly sampled from replay memory
USE_LSTM	true	Use Long Short Term Memory
LSTM_SIZE	128	Size of LSTM cell

4 Results

The required accuracy for the first challenge was obtained after approximately 10000 runs.

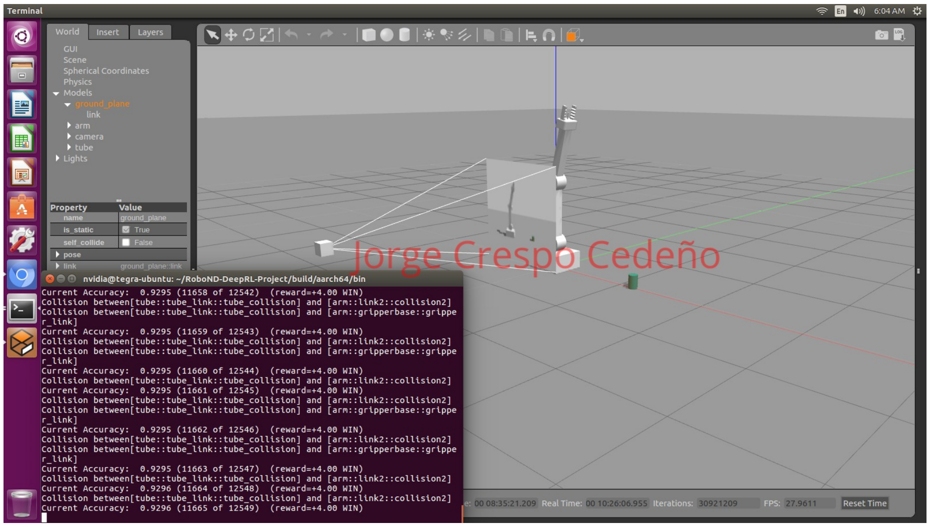
An interesting case is the second challenge, which has almost the same hyperparameters as the first one, with the restriction of getting a positive reward only when the gripper touches the tube, and with addition of one negative reward when an arm part different than the gripper touches the tube. In this second case, the number of runs required to obtained at leat 80% accuracy is 2000 approximately. Althought the learning was stopped soon after reaching 81%, probably a 90% accuracy could have been obtained with less than 10000 runs. A plausible explanation of this case is the addition of a negative reward (when a part different than the gripper was touching the tube). This extra information might be the reason for the DQN agent to learn quicker a more precise function.

4.1 First Challenge

Fig. 1 shows the arm touching the tube with the required accuracy.

4.2 Second Challenge

Fig. 2 shows only the gripper base of the arm touching the tube with the required accuracy.



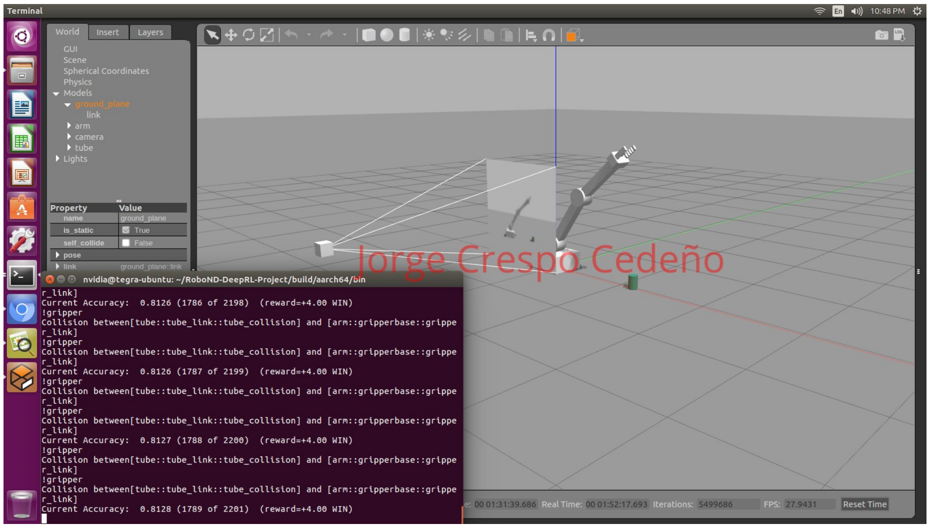
Property	Value
name	ground_plane
is_static	☑ True
self_collide	□ False
pose	
link	ground_plane::link

nvidia@tegra-ubuntu: ~/RoboND-DeepRL-Project/build/aarch64/bin

```
Current Accuracy:  0.9295 (11658 of 12542)  (reward=+4.00 WIN)
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:  0.9295 (11659 of 12543)  (reward=+4.00 WIN)
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:  0.9295 (11660 of 12544)  (reward=+4.00 WIN)
Collision between[tube::tube_link::tube_collision] and [arm::link2::collision2]
Current Accuracy:  0.9295 (11661 of 12545)  (reward=+4.00 WIN)
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:  0.9295 (11662 of 12546)  (reward=+4.00 WIN)
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:  0.9295 (11663 of 12547)  (reward=+4.00 WIN)
Collision between[tube::tube_link::tube_collision] and [arm::link2::collision2]
Current Accuracy:  0.9296 (11664 of 12548)  (reward=+4.00 WIN)
Collision between[tube::tube_link::tube_collision] and [arm::link2::collision2]
Current Accuracy:  0.9296 (11665 of 12549)  (reward=+4.00 WIN)
```

e: 00 08:35:21.209 Real Time: 00 10:26:06.955 Iterations: 30921209 FPS: 27.9611

Fig. 1. Any part of the arm manipulator touching the tube.



Property	Value
name	ground_plane
is_static	☑ True
self_collide	□ False
pose	
link	ground_plane::link

nvidia@tegra-ubuntu: ~/RoboND-DeepRL-Project/build/aarch64/bin

```
r_link]
Current Accuracy:  0.8126 (1786 of 2198)  (reward=+4.00 WIN)
!gripper
Collision between[tube::tube_link::tube_collision] and [arm::gripperbase::gripper_link]
!gripper
Collision between[tube::tube_link::tube_collision] and [arm::gripperbase::gripper_link]
Current Accuracy:  0.8126 (1787 of 2199)  (reward=+4.00 WIN)
!gripper
Collision between[tube::tube_link::tube_collision] and [arm::gripperbase::gripper_link]
!gripper
Collision between[tube::tube_link::tube_collision] and [arm::gripperbase::gripper_link]
Current Accuracy:  0.8127 (1788 of 2200)  (reward=+4.00 WIN)
!gripper
Collision between[tube::tube_link::tube_collision] and [arm::gripperbase::gripper_link]
!gripper
Collision between[tube::tube_link::tube_collision] and [arm::gripperbase::gripper_link]
Current Accuracy:  0.8128 (1789 of 2201)  (reward=+4.00 WIN)
```

e: 00 01:31:39.686 Real Time: 00 01:52:17.693 Iterations: 5499686 FPS: 27.9431

Fig. 2. Only the gripper of the arm manipulator touching the tube.

5 Future Work

In the case of the first challenge, although the reached accuracy was stable, i.e., the agent continue winning after 90% of success rate, the training took several hours. This time can be reduce by experimenting with bigger learning rates, since a small learning rate makes the agent learn slowly. In the case of the second challenge, the 80% required accuracy was reached much faster, in less than two hours approximately, with the same learning rate. In this case, bigger batch sizes can be tried to allow the agent break the correlation of successive experiences. The idea is that the agent learns several ways, i.e., arm configurations, to touch the gripper.