Project “Deep RL Arm Manipulation”

Dmitry Gavrilenko

1 Reward Functions

If the robot touches the target object, it receives +1 reward and the episode ends. In the gripper base scenario, the episode ends and the reward is received only if the robot touches the object with its gripper base part.

If the robot touches the ground or the episode exceeds 100 steps, the robot receives -1 reward and the episode ends.

The robot continuously receives a small reward dependently on the distance change between the object and the gripper. The distance change is averaged with its previous value in order to filter out noise or decrease latency artifacts of the observation->control pipeline. The distance decrease results in a small positive reward, and vice-versa. The small reward plays the role of the heuristics in the search algorithm, enabling the robot quickly finds solution towards the goal.

The distance change reward is scaled by exp(-distGoal) to increase the importance of correct motions that occur near the target object.

Additionally, the robot receives -0.01 reward for every step to penalize stationary behavior.

Reward function is implemented as assignment operators to rewardHistory variable in `./gazebo/ArmPlugin.cpp` module.

2 Hyperparameters

The camera input resolution is 64x64 pixels. Therefore, INPUT_WIDTH and INPUT_HEIGHT are also set to 64 pixels.

In case of LOCKBASE = true, the robot has only two active joints. NUM_ACTIONS equals 4 to provide two actions per joint.

“RMSprop” optimizer with LEARNING_RATE 0.1f is borrowed from catch sample. It provides better accuracy than “Adam” with LEARNING_RATE 0.001f.

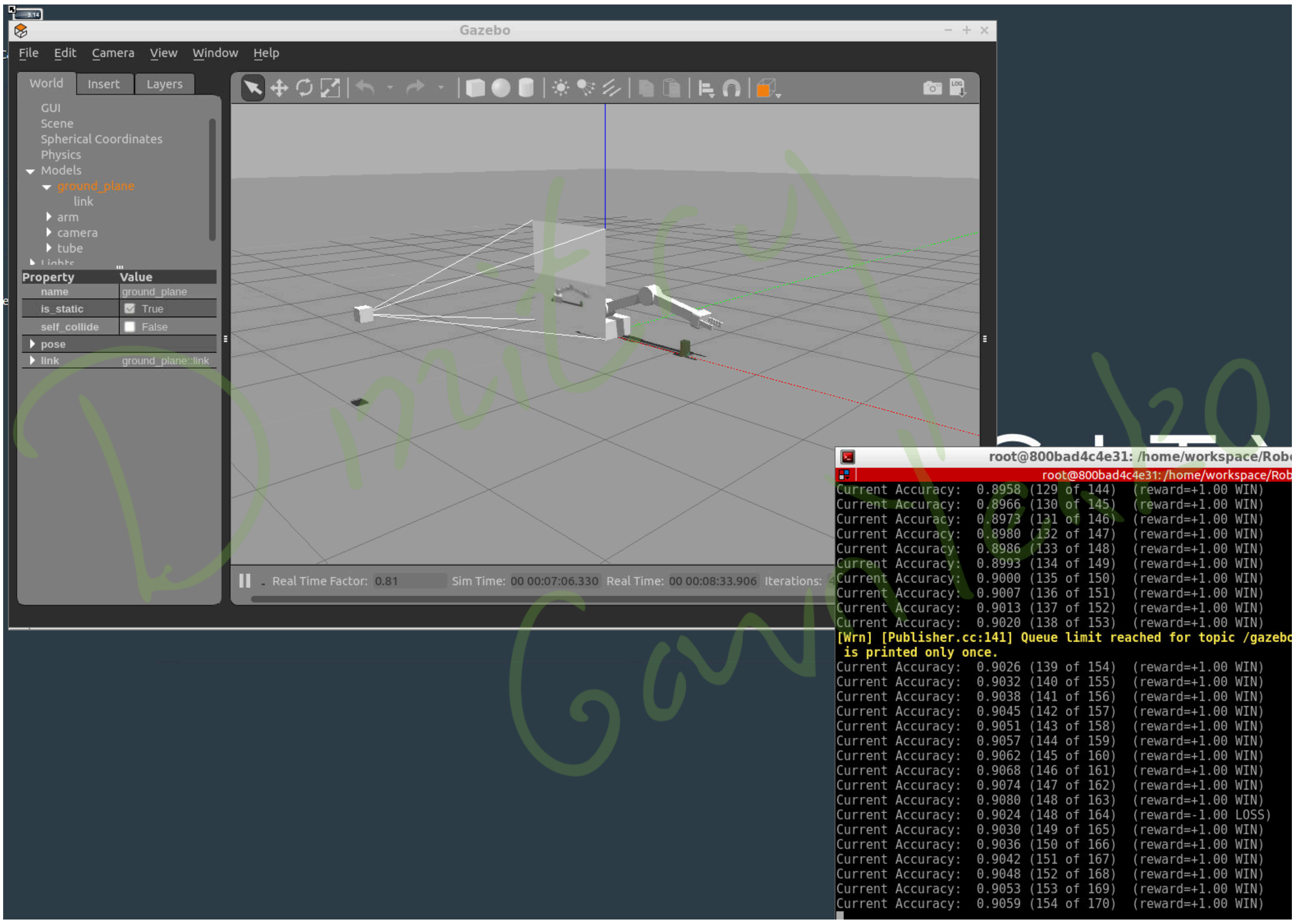
REPLAY_MEMORY with 1000 samples and BATCH_SIZE 32 were tuned up experimentally.

EPS_END is set 0.01 to prevent the robot unlearn good behavior. This decreases chances of random actions towards the end of the learning. The recommendation was borrowed from udacity-robotics.slack forum.

An experiment with LSTM and velocity controls showed good convergence rate but took a lot of training time (tens of thousands iterations). Therefore, to fit the training hardware resource constraints, the neural network does not use memory and controls the joint rotations directly. USE_LSTM is set false, and the training time takes hundreds of iterations.

3 Results

The robot reaches over 90% accuracy for the arm touching the object task in about 170 iterations, as shown in the pictures below:

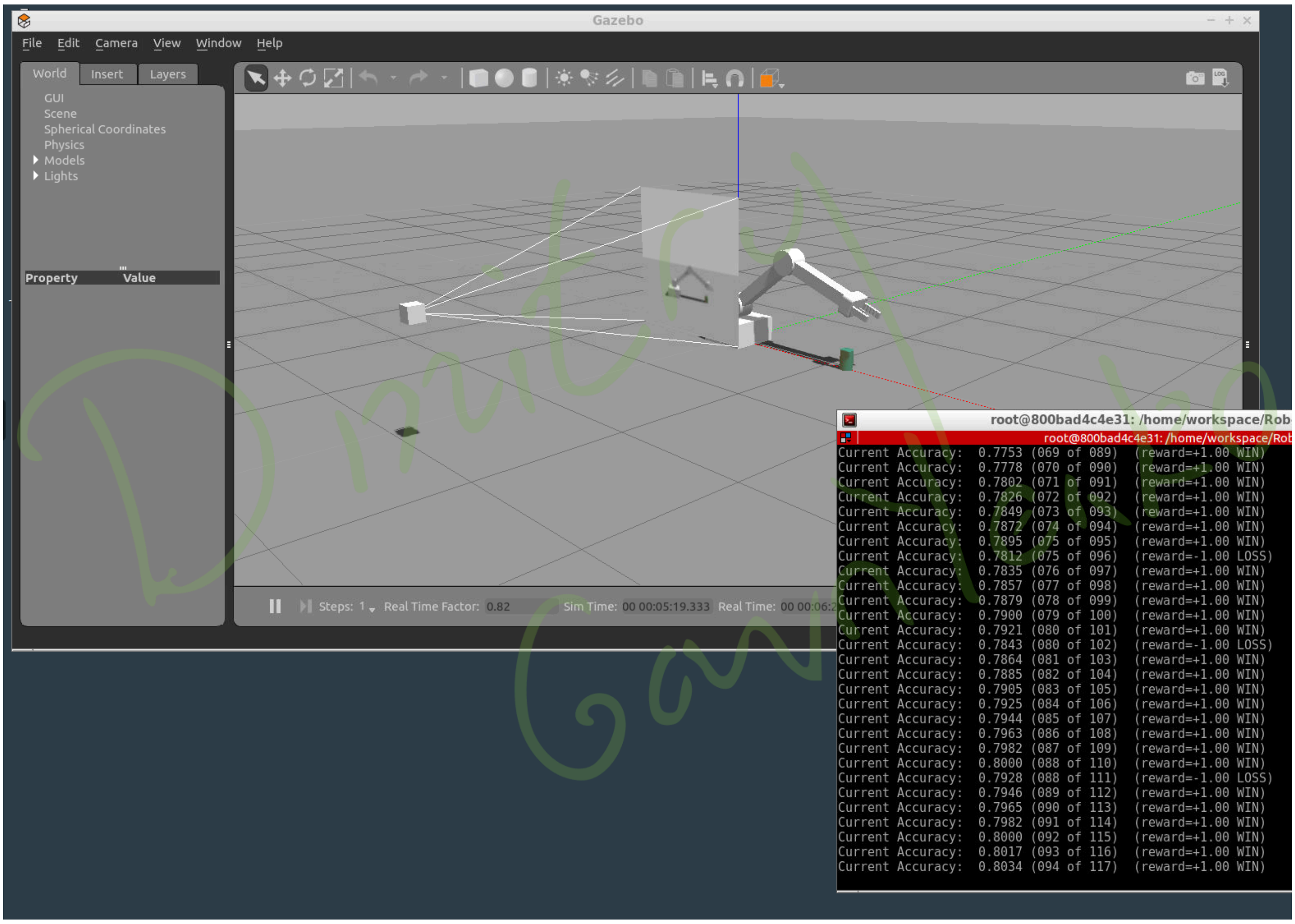


Property	Value
name	ground_plane
is_static	✓ True
self_collide	False
pose	
link	ground_plane::link

Real Time Factor: 0.81 Sim Time: 00 00:07:06.330 Real Time: 00 00:08:33.906 Iterations:

```
root@800bad4c4e31: /home/workspace/Rob
Current Accuracy:  0.8958 (129 of 144)  (reward=+1.00 WIN)
Current Accuracy:  0.8966 (130 of 145)  (reward=+1.00 WIN)
Current Accuracy:  0.8973 (131 of 146)  (reward=+1.00 WIN)
Current Accuracy:  0.8980 (132 of 147)  (reward=+1.00 WIN)
Current Accuracy:  0.8986 (133 of 148)  (reward=+1.00 WIN)
Current Accuracy:  0.8993 (134 of 149)  (reward=+1.00 WIN)
Current Accuracy:  0.9000 (135 of 150)  (reward=+1.00 WIN)
Current Accuracy:  0.9007 (136 of 151)  (reward=+1.00 WIN)
Current Accuracy:  0.9013 (137 of 152)  (reward=+1.00 WIN)
Current Accuracy:  0.9020 (138 of 153)  (reward=+1.00 WIN)
[Wrn] [Publisher.cc:141] Queue limit reached for topic /gazebo
 is printed only once.
Current Accuracy:  0.9026 (139 of 154)  (reward=+1.00 WIN)
Current Accuracy:  0.9032 (140 of 155)  (reward=+1.00 WIN)
Current Accuracy:  0.9038 (141 of 156)  (reward=+1.00 WIN)
Current Accuracy:  0.9045 (142 of 157)  (reward=+1.00 WIN)
Current Accuracy:  0.9051 (143 of 158)  (reward=+1.00 WIN)
Current Accuracy:  0.9057 (144 of 159)  (reward=+1.00 WIN)
Current Accuracy:  0.9062 (145 of 160)  (reward=+1.00 WIN)
Current Accuracy:  0.9068 (146 of 161)  (reward=+1.00 WIN)
Current Accuracy:  0.9074 (147 of 162)  (reward=+1.00 WIN)
Current Accuracy:  0.9080 (148 of 163)  (reward=+1.00 WIN)
Current Accuracy:  0.9024 (148 of 164)  (reward=-1.00 LOSS)
Current Accuracy:  0.9030 (149 of 165)  (reward=+1.00 WIN)
Current Accuracy:  0.9036 (150 of 166)  (reward=+1.00 WIN)
Current Accuracy:  0.9042 (151 of 167)  (reward=+1.00 WIN)
Current Accuracy:  0.9048 (152 of 168)  (reward=+1.00 WIN)
Current Accuracy:  0.9053 (153 of 169)  (reward=+1.00 WIN)
Current Accuracy:  0.9059 (154 of 170)  (reward=+1.00 WIN)
```

The robot reaches over 80% accuracy for the gripper base touching the object task in about 117 iterations, as shown in the picture below.


Gazebo
File Edit Camera View Window Help
World Insert Layers
GUI
Scene
Spherical Coordinates
Physics
Models
Lights
Property Value
Steps: 1 Real Time Factor: 0.82 Sim Time: 00 00:05:19.333 Real Time: 00 00:06:2

```
root@800bad4c4e31: /home/workspace/Rob
Current Accuracy:  0.7753 (069 of 089)  (reward=+1.00 WIN)
Current Accuracy:  0.7778 (070 of 090)  (reward=+1.00 WIN)
Current Accuracy:  0.7802 (071 of 091)  (reward=+1.00 WIN)
Current Accuracy:  0.7826 (072 of 092)  (reward=+1.00 WIN)
Current Accuracy:  0.7849 (073 of 093)  (reward=+1.00 WIN)
Current Accuracy:  0.7872 (074 of 094)  (reward=+1.00 WIN)
Current Accuracy:  0.7895 (075 of 095)  (reward=+1.00 WIN)
Current Accuracy:  0.7812 (075 of 096)  (reward=-1.00 LOSS)
Current Accuracy:  0.7835 (076 of 097)  (reward=+1.00 WIN)
Current Accuracy:  0.7857 (077 of 098)  (reward=+1.00 WIN)
Current Accuracy:  0.7879 (078 of 099)  (reward=+1.00 WIN)
Current Accuracy:  0.7900 (079 of 100)  (reward=+1.00 WIN)
Current Accuracy:  0.7921 (080 of 101)  (reward=+1.00 WIN)
Current Accuracy:  0.7843 (080 of 102)  (reward=-1.00 LOSS)
Current Accuracy:  0.7864 (081 of 103)  (reward=+1.00 WIN)
Current Accuracy:  0.7885 (082 of 104)  (reward=+1.00 WIN)
Current Accuracy:  0.7905 (083 of 105)  (reward=+1.00 WIN)
Current Accuracy:  0.7925 (084 of 106)  (reward=+1.00 WIN)
Current Accuracy:  0.7944 (085 of 107)  (reward=+1.00 WIN)
Current Accuracy:  0.7963 (086 of 108)  (reward=+1.00 WIN)
Current Accuracy:  0.7982 (087 of 109)  (reward=+1.00 WIN)
Current Accuracy:  0.8000 (088 of 110)  (reward=+1.00 WIN)
Current Accuracy:  0.7928 (088 of 111)  (reward=-1.00 LOSS)
Current Accuracy:  0.7946 (089 of 112)  (reward=+1.00 WIN)
Current Accuracy:  0.7965 (090 of 113)  (reward=+1.00 WIN)
Current Accuracy:  0.7982 (091 of 114)  (reward=+1.00 WIN)
Current Accuracy:  0.8000 (092 of 115)  (reward=+1.00 WIN)
Current Accuracy:  0.8017 (093 of 116)  (reward=+1.00 WIN)
Current Accuracy:  0.8034 (094 of 117)  (reward=+1.00 WIN)
```

4 Future Work

The following ideas may further improve the quality of the DRL:

- Pre-train the neural network with labeled data: a recommended action per a given frame, prepared by a human operator

- Try Temporal Difference learning

- Parallelize the training task against multiple simulated or real robots, working in parallel on different computers, and filling in replay memory buffer