Project: Deep RL Arm Manipulation

Oleksii Zhurbytskyi

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

Index Terms—Robot, IEEEtran, Udacity, LaTeX, Deep RL.

1 Reward Functions

1.1 Objective 1: Any part of the robot arm touch the object of interest

1) If any part of robot touch the object the reward is REWARD_WIN * 10
2) If any part of robot touch the ground the penalty is REWARD_LOSS * 10
3) Additional reward/penalty if the robot do not touch the ground depends of the smoothed moving average of the delta of the distance to the goal

1.2 Objective 2: Only the gripper base of the robot arm touch the object

1) If the gripper base of robot touch the object the reward is REWARD_WIN * 20
2) If any other part of robot touch the object the penalty is REWARD_LOSS * 5
3) If any part of robot touch the ground the penalty is REWARD_LOSS * 10
4) Additional reward/penalty if the robot do not touch the ground depends of the smoothed moving average of the delta of the distance to the goal

The joint movements were controlled by position.

2 Hyperparameters

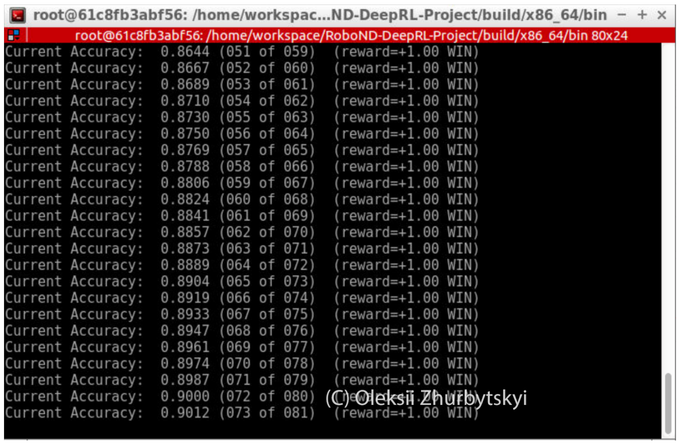
The main factors in choosing hyperparameters were the size of the input image, the amount of RAM and learning results. Two optimizers were tested, and it turned out that the Adam shows better results than the RMSProp for this task. LEARNING_RATE and REPLAY_MEMORY hyperparameters have been selected by trial and error starting from 0.1 for LEARNING_RATE and 1000 for REPLAY_MEMORY. If the goal could not be achieved, the LEARNING_RATE was decreased and the REPLAY_MEMORY was increased.

- INPUT_WIDTH is 64 and INPUT_HEIGHT is 64 - were chosen according to the size of the image
- OPTIMIZER is Adam
- LEARNING_RATE is 0.1 for Objective 1 and 0.01 for Objective 2
- REPLAY_MEMORY is 1000 for Objective 1 and 10000 for Objective 2
- BATCH_SIZE is 256 as we have a large amount for RAM
- LSTM_SIZE is 256

Thus, the only differences in hyperparameters for two objectives are LEARNING_RATE and REPLAY_MEMORY.

3 Results

For both objectives specified accuracy has been achieved in the provided environment. It turned out that the sooner the robot reaches the goal for the first time, the faster the training goes. The provided screenshots show that the goals were achieved within 100 runs.

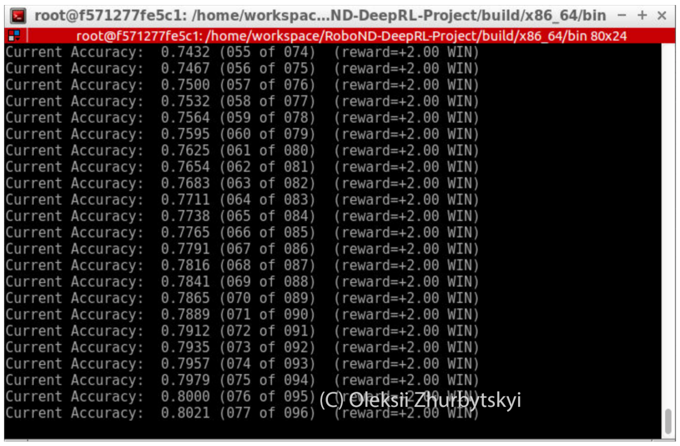

root@61c8fb3abf56: /home/workspac...ND-DeepRL-Project/build/x86_64/bin
root@61c8fb3abf56: /home/workspace/RoboND-DeepRL-Project/build/x86_64/bin 80x24

```
Current Accuracy:  0.8644 (051 of 059)  (reward=+1.00 WIN)
Current Accuracy:  0.8667 (052 of 060)  (reward=+1.00 WIN)
Current Accuracy:  0.8689 (053 of 061)  (reward=+1.00 WIN)
Current Accuracy:  0.8710 (054 of 062)  (reward=+1.00 WIN)
Current Accuracy:  0.8730 (055 of 063)  (reward=+1.00 WIN)
Current Accuracy:  0.8750 (056 of 064)  (reward=+1.00 WIN)
Current Accuracy:  0.8769 (057 of 065)  (reward=+1.00 WIN)
Current Accuracy:  0.8788 (058 of 066)  (reward=+1.00 WIN)
Current Accuracy:  0.8806 (059 of 067)  (reward=+1.00 WIN)
Current Accuracy:  0.8824 (060 of 068)  (reward=+1.00 WIN)
Current Accuracy:  0.8841 (061 of 069)  (reward=+1.00 WIN)
Current Accuracy:  0.8857 (062 of 070)  (reward=+1.00 WIN)
Current Accuracy:  0.8873 (063 of 071)  (reward=+1.00 WIN)
Current Accuracy:  0.8889 (064 of 072)  (reward=+1.00 WIN)
Current Accuracy:  0.8904 (065 of 073)  (reward=+1.00 WIN)
Current Accuracy:  0.8919 (066 of 074)  (reward=+1.00 WIN)
Current Accuracy:  0.8933 (067 of 075)  (reward=+1.00 WIN)
Current Accuracy:  0.8947 (068 of 076)  (reward=+1.00 WIN)
Current Accuracy:  0.8961 (069 of 077)  (reward=+1.00 WIN)
Current Accuracy:  0.8974 (070 of 078)  (reward=+1.00 WIN)
Current Accuracy:  0.8987 (071 of 079)  (reward=+1.00 WIN)
Current Accuracy:  0.9000 (072 of 080)
Current Accuracy:  0.9012 (073 of 081)  (reward=+1.00 WIN)
```

Fig. 1. Result for Objective 1 - any part of the robot arm touch the object of interest, with at least a 90% accuracy.

4 Future work

As a result of the experiments, it became obvious that training begins to go well only after the robot first touches the target. Therefore, further improvements should be concentrated precisely on this task. Most likely this can be achieved by optimizing the reward functions.


root@f571277fe5c1: /home/workspac...ND-DeepRL-Project/build/x86_64/bin
root@f571277fe5c1: /home/workspace/RoboND-DeepRL-Project/build/x86_64/bin 80x24

```
Current Accuracy:  0.7432 (055 of 074)  (reward=+2.00 WIN)
Current Accuracy:  0.7467 (056 of 075)  (reward=+2.00 WIN)
Current Accuracy:  0.7500 (057 of 076)  (reward=+2.00 WIN)
Current Accuracy:  0.7532 (058 of 077)  (reward=+2.00 WIN)
Current Accuracy:  0.7564 (059 of 078)  (reward=+2.00 WIN)
Current Accuracy:  0.7595 (060 of 079)  (reward=+2.00 WIN)
Current Accuracy:  0.7625 (061 of 080)  (reward=+2.00 WIN)
Current Accuracy:  0.7654 (062 of 081)  (reward=+2.00 WIN)
Current Accuracy:  0.7683 (063 of 082)  (reward=+2.00 WIN)
Current Accuracy:  0.7711 (064 of 083)  (reward=+2.00 WIN)
Current Accuracy:  0.7738 (065 of 084)  (reward=+2.00 WIN)
Current Accuracy:  0.7765 (066 of 085)  (reward=+2.00 WIN)
Current Accuracy:  0.7791 (067 of 086)  (reward=+2.00 WIN)
Current Accuracy:  0.7816 (068 of 087)  (reward=+2.00 WIN)
Current Accuracy:  0.7841 (069 of 088)  (reward=+2.00 WIN)
Current Accuracy:  0.7865 (070 of 089)  (reward=+2.00 WIN)
Current Accuracy:  0.7889 (071 of 090)  (reward=+2.00 WIN)
Current Accuracy:  0.7912 (072 of 091)  (reward=+2.00 WIN)
Current Accuracy:  0.7935 (073 of 092)  (reward=+2.00 WIN)
Current Accuracy:  0.7957 (074 of 093)  (reward=+2.00 WIN)
Current Accuracy:  0.7979 (075 of 094)  (reward=+2.00 WIN)
Current Accuracy:  0.8000 (076 of 095)  (reward=+2.00 WIN)
Current Accuracy:  0.8021 (077 of 096)  (reward=+2.00 WIN)
```

Fig. 2. Result for Objective 2 - only the gripper base of the robot arm touch the object, with at least a 80% accuracy.