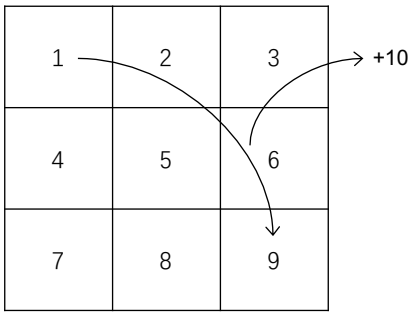
SI252 Reinforcement Learning 2020/04/08

Homework 6

Professor: Ziyu Shao *Due: 2020/06/06 11:59am*

1. **3x3 Grid World**:



The agent can be in one of the nine cells at any starting time. It can then move in one of four directions:{E,S,W,N}. If the agent hits a wall, it remains in its current cell and gets a reward -1. When the agent moves to cell 1, it then immediately moves to cell 9 and gets a reward of 10. The discount factor $\gamma = 0.9$.

   (a) Under the uniform policy (equal probabilities for each possible actions), compute the value of each state(cell).

   (b) For subproblem (a), show numerical results obtained by value-iteration algorithm and TD algorithm. Discuss the pros and cons of each algorithm.

   (c) Find the optimal value of each state and corresponding optimal policy.

   (d) For subproblem (c), show numerical results obtained by policy-iteration algorithm and Q-learning algorithm. Discuss the pros and cons of each algorithm.

2. **Examples in the book "Reinforcement Learning: An Introduction" (second edition) by Richard S. Sutton and Andrew G. Barto**
For the following examples, solve corresponding exercises and reproduce corresponding figures. **You can are required to select 3 examples. The remaining 15 examples are bonus problems**. Try to finish such examples as many as possible.

   (a) Example 4.1 with Exercises 4.1- 4.3 and Figure 4.1

   (b) Example 4.2 with Exercises 4.4-4.7 and Figure 4.2

   (c) Example 4.3 with Exercises 4.8-4.10 and Figure 4.3

(d) Example 5.1 with Exercises 5.1-5.2 and Figure 5.1

(e) Example 5.3 with Figure 5.2

(f) ~~Example 5.4 with Exercise 5.7 and Figure 5.3~~

(g) Example 5.5 with Exercise 5.8 and Figure 5.4

(h) Example 6.1 with Exercise 6.2 and Figure 6.1

(i) Example 6.2 with Exercises 6.3-6.6 and two graphs

(j) Example 6.3 with Figure 6.2

(k) Example 6.5 with Exercises 6.9-6.10 and one graph

(l) Example 6.6 with Exercises 6.11-6.12 , two graphs and Figure 6.3

(m) Example 6.7 with Exercise 6.13 and Figure 6.5

(n) Example 8.1 with Exercise 8.1 and Figures 8.2 & 8.3

(o) Examples 8.2, 8.3 with Exercises 8.2-8.5 and Figures 8.4 & 8.5

(p) Example 9.1 with Figure 9.1

(q) Example 9.2 with Figure 9.2

(r) Example 13.1 with Exercise 13.1 and Figures 13.1 & 13.2

3. **Python Implementation of REINFORCEjs**. Written by Java language, REINFORCEjs is a Reinforcement Learning library that implements several common RL algorithms supported with fun web demos. The web address is: here. The Java source code is maintained in GitHub.

   (a) Reproduce the "GridWorld: Dynamic Programming Demo" by Python.

   (b) Reproduce the "GridWorld: Temporal Difference Learning Demo" by Python.

   (c) (**Bonus Problem**) Reproduce the "PuckWorld: DQN Demo" by Python.

   (d) (**Bonus Problem**) Reproduce the "WaterWorld: DQN Demo" by Python.

4. **Bonus Problem: OpenAI Spinning Up in Deep RL**. Welcome to Spinning Up in Deep RL! This is an educational resource produced by OpenAI that makes it easier to learn about deep reinforcement learning (deep RL). Please study the documents and install the environment. Either PyTorch or TensorFlow are allowed. In your report, please provide detailed figures and analysis to show many aspects of performances of various DRL algorithms.

   (a) Finish the problem set 1: "Basics of Implementation" . It includes three exercises: Gaussian Log-Likelihood, Policy for PPO, and Computation Graph for TD3.

   (b) Finish the problem set 2: "Algorithm Failure Modes" . It includes two exercises: Value Function Fitting in TRPO, and Silent Bug in DDPG.

5. **Bonus Problem: Tianshou Platform for Deep RL**. Tianshou is a reinforcement learning platform based on pure PyTorch. Tianshou provides a fast-speed framework and pythonic API for building the deep reinforcement learning agent. Please confirm at least two scenarios of the following table:

RL Platform	Tianshou	Baselines	Ray/RLlib	PyTorch DRL	rlpyt
GitHub Stars	stars 801	stars 9.6k	stars 11k	stars 2.3k	stars 1.4k
Algo - Task	PyTorch	TensorFlow	TF/PyTorch	PyTorch	PyTorch
PG - CartPole	9.03±4.18s	None	15.77±6.28s	None	?
DQN - CartPole	10.61±5.51s	1046.34±291.27s	40.16±12.79s	175.55±53.81s	?
A2C - CartPole	11.72±3.85s	*(~1612s)	46.15±6.64s	Runtime Error	?
PPO - CartPole	35.25±16.47s	*(~1179s)	62.21±13.31s (APPO)	29.16±15.46s	?
DDPG - Pendulum	46.95±24.31s	*(>1h)	377.99±13.79s	652.83±471.28s	172.18±62.48s
TD3 - Pendulum	48.39±7.22s	None	620.83±248.43s	619.33±324.97s	210.31±76.30s
SAC - Pendulum	38.92±2.09s	None	92.68±4.48s	808.21±405.70s	295.92±140.85s

**: Could not reach the target reward threshold in 1e6 steps in any of 10 runs. The total runtime is in the brackets.*

?: We have tried but it is nontrivial for running non-Atari game on rlpyt. See here.