Report of HW3 (Deep Reinforcement Learning)

Alireza Molla Ali Hosseini

(Dated: July 2022)

INTRODUCTION

Tasks

- Change the Hyperparameters or reward function to reach a better performance (for lab 7)
- train a deep RL agent on a different Gym enviroment

The main tasks of this homework was written above. Therefor my first step is to use Lab 7 environment and try to enhance the performance of the RL-agent. My second task is to create another RL-agent to solve the Acrobat Gym environment.

The goal of the agent is to interact with the emulator (environment) by selecting actions in a way that maximizes future rewards. In order to do so, we have to optimize the function Q*(s,a) (action-reward function) which s is some (previous) sequences and a is an action.

The basic idea behind many reinforcement learning algorithms is to estimate the action-value function, by using the Bellman equation as an iterative update. But this is impractical. Instead, it is common to use a function approximator to estimate the action-value function. In the reinforcement learning community this is typically a linear function approximator, but sometimes a non-linear function approximator is used instead, such as a neural network (DQN).

In this homework the loss function is Huber-loss function (Smooth-L1-Loss)

METHODS

Replay Memory

It is a function used in the training steps to push the tuple of state, action, next state and reward into the memory. The memory itself has a specified capacity. This function can also sample from the previous memories which will be useful in the training steps as well.

DQN

It is the deep neural network that will be used as an agent in the architecture. The network used for lab 7 and the Acrobot are similar with two hidden layers (first one with 64 neurons and second one with 4096) and ReLU function as the activation function (Parameters of the network is listed in Table I. The input is the state space and the output will be action space (in lab 7 state space has 4 dimension and action space has 2 dimension. The Acrobot, however, has 6 state space dimension and 3 action space dimension). Table II

Epsilon Greedy

Epsilon Greedy is one of the action policies to choose an action. In this policy, firstly it find the best action which is the output of the deep network then with probability of equal or greater than the epsilon will choose it. Otherwise it can choose a random action (among the action space).

Softmax

Softmax is another policy to choose an action. In this policy there is a variable which is like a temperature to control the level of randomness of the selection (at high temperature the choice will be random but at lower temperature the choice will be more deterministic). At zero temperature it will always choose the best action (output of the network) but in other temperatures it chooses the action from a softmax distribution. The temperature itself also come from exploration profile (Figure 1) which is a decaying function from a variable number to zero with specified number of steps.

Training

In training first state will be the first environment step and then use softmax policy for action. Then go to the next state by using the action, keep track of the reward and pushing the previous state, action, next state and reward to the memory. In this stage we can also implement a strategy for reward to enhance the performance of the agent. We repeat this process until we reach a minimum number for training, then we use the update step function which will use both policy network and target network to increase performance of the agent and actually solve the problem.

Update step

This function is representing the main concept of an RL-agent and its functionality. It samples uniformly from previous observations (from replay memory) and set the next output equal to the sample reward if it is the last state otherwise set the it to sample reward plus the best action (from DQN) with a discounted factor (gamma). Then it tries to minimize the loss function (between the next output and the output of DQN).

RESULTS

The firs part of task which was related to the Cartpole environment was solved by a RL-agent after about 500 steps (after 500 steps the agent was able to choose the action optimally to solve the environment and get maximum reward). The performance of the agent is shown in Figure 2

The second part of the homework was related to the Acrobot environment which the RL-agent was able to solve the problem and got maximum reward after about 500 steps. The results are plotted in Figure 3

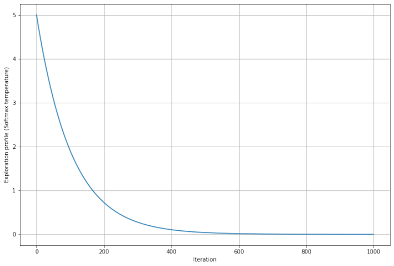
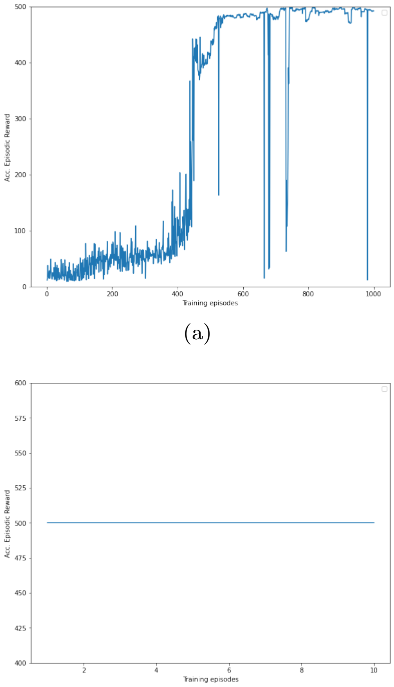


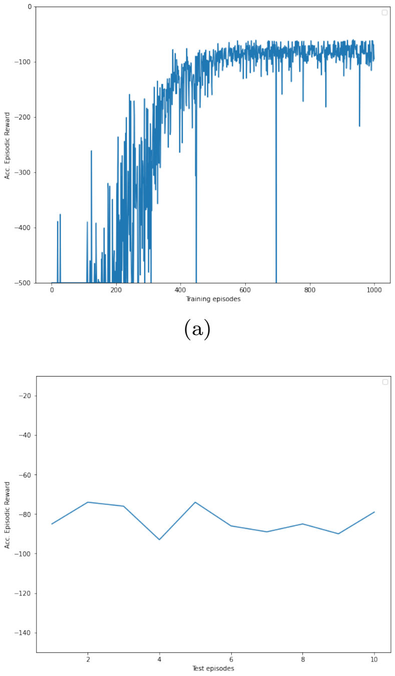
FIG. 1: Exploration Profile for the softmax policy



(a)

(b)

FIG. 2: (a) Performance of the agent for the Cartpole environment (b) Sample Rewards of the agent after training



(a)

(b)

FIG. 3: (a) Performance of the agent for the Acrobot environment (b) Sample Rewards of the agent after training

Parameter	**Value**
Batch size	256
target network update steps	10
replay memory capacity	10000
minimum samples for training	1000
Linear layers	3
Activation function	**ReLU**
Optimizer	**Adam**
Loss Function	**Smooth-L1-Loss**
learning rate	1e-3
Gamma	0.99

TABLE I: DQN Parameters of Cartpole and Acrobot

Environment	**Space**	**Dimension**
CartPole-v1	**STATE SPACE**	4
	ACTION SPACE	2
Acrobot-v1	**STATE SPACE**	6
	ACTION SPACE	3

TABLE II: Gym Environment Parameters