

RUTGERS UNIVERSITY
School of Engineering
Department of Electrical & Computer Engineering
ECE 472 – Robotics & Computer Vision– Fall 2022

Project 2 - Reinforcement Learning

Name (last, first) : Mehmet Ali Soner

netID : mas996

RUID: 196000499

Date: November 26, 2022

1 Problem 1

Cart Pole code assignment, code + explanation

I have structured the code into two main parts: a part before and after training. The part before training is mainly to test out components such as the video capturing of an episode, the overall environment and the functions built within. First we call the following terminal commands:

```
!apt-get update
!apt-get install -y xvfb x11-utils
%pip install pyvirtualdisplay==0.2.*
%pip install gym[classic_control]
```

I am calling apt-get update since I ran into issues with xvfb couple of times and this seem to resolve it. We install xvfb and pyvirtualdisplay for the video capturing. Then we import the gym package and create the environment for "CartPole":

```
# Prepare environment, we will first demonstrate a video withouth any training

import gym

if gym.__version__ < '0.26':
    env = gym.make('CartPole-v0', new_step_api=True, render_mode='single_rgb_array').
        unwrapped
else:
    env = gym.make('CartPole-v0', render_mode='rgb_array').unwrapped
```

After this, I will skip few things such as the video capturing and importing other various packages. I want to focus on the DQN's linear layer. In this model, we have to have two possible outputs in our final/linear layer since we will either push the cart to the right or to the left. In order to add this layer, we have to know the output size of each convolutional layer so that we have the right input size for the linear layer. That's why we call conv2d_size_out three times since we have three convolutional layers. We then calculate the input size and add it with:

```
linear_input_size = convw * convh * 32
self.head = nn.Linear(linear_input_size, outputs)
```

We then skip few functions that

step: how many times through environment. observation = [x cart, y cart, pole angle, pole ang. velocity]. reward = 1 for each step. done = when episode is done (maybe pole tipped over too much)

2 Problem 2

Explain DQN algorithm in paragraphs, include definitions of state, action, environment, reward.

Notes: Agents and environment: agent(s) interacts with environment, which can be a simulation, such as the cart pole example. Each step, the agent observes the environment (state of environment) and takes action and receives a reward based on that. Agents learn from repeated trials; these are called episodes. RL framework trains a policy for the agent to follow. Policy shows which actions to take one after the other in order to maximize reward.

In RL, want to train the agent to make better decision or act better with each episode. Policy == neural network.

3 Problem 3

Performance metrics and plots

4 Problem 4

Three other problems for RL; state, action, environment and reward for each