Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»

Кафедра «Системы обработки информации и управления»



**Лабораторная работа №7 по дисциплине**

**«Методы машинного обучения»**

**По теме «Алгоритмы Actor-Critic»**

**ИСПОЛНИТЕЛЬ:**

Позняк А.А.

Группа ИУ5-22M

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

"\_\_"\_\_\_\_\_\_\_\_\_\_\_2023 г.

Москва 2023

Оглавление

[Задание 3](#_Toc137164600)

[Текст программы 4](#_Toc137164601)

[Экранные формы 7](#_Toc137164602)

Задание

* Реализуйте любой алгоритм семейства Actor-Critic для произвольной среды.

Текст программы

%%capture

!apt install python-opengl

!apt install ffmpeg

!apt install xvfb

!pip3 install pyvirtualdisplay

# Virtual display

from pyvirtualdisplay import Display

virtual\_display = Display(visible=0, size=(1400, 900))

virtual\_display.start()

# Install the specific setuptools and wheel version required to install the dependencies

!pip install setuptools==65.5.0 wheel==0.38.4

!pip install -r https://raw.githubusercontent.com/huggingface/deep-rl-class/main/notebooks/unit6/requirements-unit6.txt

import pybullet\_envs

import panda\_gym

import gym

import os

from huggingface\_sb3 import load\_from\_hub, package\_to\_hub

from stable\_baselines3 import A2C

from stable\_baselines3.common.evaluation import evaluate\_policy

from stable\_baselines3.common.vec\_env import DummyVecEnv, VecNormalize

from stable\_baselines3.common.env\_util import make\_vec\_env

from huggingface\_hub import notebook\_login

env\_id = "AntBulletEnv-v0"

# Create the env

env = gym.make(env\_id)

# Get the state space and action space

s\_size = env.observation\_space.shape[0]

a\_size = env.action\_space

print("\_\_\_\_\_OBSERVATION SPACE\_\_\_\_\_ \n")

print("The State Space is: ", s\_size)

print("Sample observation", env.observation\_space.sample()) # Get a random observation

print("\n \_\_\_\_\_ACTION SPACE\_\_\_\_\_ \n")

print("The Action Space is: ", a\_size)

print("Action Space Sample", env.action\_space.sample()) # Take a random action

env = make\_vec\_env(env\_id, n\_envs=4)

env = VecNormalize(env, norm\_obs=True, norm\_reward=True, clip\_obs=10.)

model = A2C(policy = "MlpPolicy",

env = env,

gae\_lambda = 0.9,

gamma = 0.99,

learning\_rate = 0.00096,

max\_grad\_norm = 0.5,

n\_steps = 8,

vf\_coef = 0.4,

ent\_coef = 0.0,

policy\_kwargs=dict(

log\_std\_init=-2, ortho\_init=False),

normalize\_advantage=False,

use\_rms\_prop= True,

use\_sde= True,

verbose=1)

model.learn(2\_000\_000)

# Save the model and VecNormalize statistics when saving the agent

model.save("a2c-AntBulletEnv-v0")

env.save("vec\_normalize.pkl")

Экранные формы





