|  |  |  |  |
| --- | --- | --- | --- |
| A red and white logo  Description automatically generated | Politechnika Bydgoska im. J. J. Śniadeckich  Wydział Telekomunikacji, Informatyki i Elektrotechniki | | A blue logo with a white background  Description automatically generated |
| **Przedmiot** | Analiza Regresji i Szeregów Czasowych | | |
| **Prowadzący** | dr inż. Damian Ledziński | | |
| **Temat** | Gra Atari i Stable-Baselines | | |
| **Student** | Zuzanna Tarazewicz | **Nr Indeksu** | 116954 |
| **Ocena** |  | **Data Oddania** |  |

Spis Treści

[Cel 3](#_Toc172563502)

[Wykonanie zadania 3](#_Toc172563503)

[Wybór gry 3](#_Toc172563504)

[Wybór modelu 3](#_Toc172563505)

[Wykonanie 3](#_Toc172563506)

[Zaimportowanie odpowiednich bibliotek 3](#_Toc172563507)

[Zdefiniowanie środowiska 4](#_Toc172563508)

[Trening modelu 4](#_Toc172563509)

[Testowanie modelu 5](#_Toc172563510)

[Wynik 6](#_Toc172563511)

[Wnioski 6](#_Toc172563512)

# Cel

Celem laboratorium było wyuczenie algorytmu ze wzmocnieniem (RL, Reinforcement Learning) umiejącego grać w wybraną grę Atari.

Uczenie ze wzmocnieniem jest gałęzią uczenia maszynowego. Zajmuje się tym, w jaki sposób agenci mogą podejmować decyzje w środowiskach, aby zmaksymalizować nagrody. Jednym z zastosowań jest trenowanie agentów do gier komputerowych, gdzie agent musi się nauczyć optymalnych strategii na podstawie doświadczeń ze środowiska.

# Wykonanie zadania

## Wybór gry

W tym przypadku padło na grę Zaxxon z biblioteki OpenAI Gymnasium (znane w starszej wersji jako Gym). Biblioteka Gymnasium zawiera dużo środowisk do tworzenia agentów uczenia ze wzmocnieniem, w tym posiada sporą kolekcję gier Atari.

## Wybór modelu

Po zrobieniu wstępnego rozeznania w temacie uczenia ze wzmocnieniem wybór padł na agenta Deep Q Learning (DQN).

## Wykonanie

### Zaimportowanie odpowiednich bibliotek

import gymnasium as gym

from gymnasium.wrappers import AtariPreprocessing, FrameStack

from stable\_baselines3 import DQN

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

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

from stable\_baselines3.common.monitor import Monitor

Uczenie następuje za pomocą biblioteki gymnasium i stable\_baselines3. Początkowo wybór padł na starszą wersję biblioteki gymnasium – gym, jednakże ze względu na problemy z utworzeniem środowiska i brak dalszego wsparcia – wybór padł na nowszą wersje biblioteki. Za pośrednictwem tych dwóch bibliotek możliwe było wybór środowiska, wybór trybu renderowania, wyuczenie modelu i zapisanie video modelu podczas gry.

### Zdefiniowanie środowiska

def makeEnv*(envId: str, renderMode = None)*:

env = gym.make(envId, render\_mode=renderMode)

env = AtariPreprocessing(env, frame\_skip=1, scale\_obs=True)

env = FrameStack(env, 4)

return env

envId = 'ALE/Zaxxon-v5'

env = DummyVecEnv([lambda: makeEnv(envId)])

Środowisko zostaje zdefiniowane za pośrednictwem funkcji makeEnv, która przyjmuje od jednego do dwóch argumentów (drugi argument jest predefiniowany, dlatego można go pominąć) i zwraca utworzone środowisko, które zostaje zapisane pod zmienną env.

### Trening modelu

model = DQN('MlpPolicy', env, verbose=1, buffer\_size=10000, learning\_starts=1000,

batch\_size=32, gamma=0.99, target\_update\_interval=1000, train\_freq=4,

gradient\_steps=1, exploration\_fraction=0.1, exploration\_final\_eps=0.01)

model.learn(total\_timesteps=50000)

model.save('dqn\_zaxxon.pkl')

Trening modelu był najdłuższym elementem tworzenia programu. Szkolony agent DQN został wyuczony w 50000 krokach. Po wyuczeniu model został zapisany jako dqn\_zaxxon.pkl. Wynik uczenia był następujący:

----------------------------------

| rollout/ | |

| exploration\_rate | 0.01 |

| time/ | |

| episodes | 56 |

| fps | 351 |

| time\_elapsed | 141 |

| total\_timesteps | 49603 |

| train/ | |

| learning\_rate | 0.0001 |

| loss | 0.000128 |

| n\_updates | 12150 |

----------------------------------

Co jest dość zadawalającym wynikiem, ze względu na to, że strata jest bardzo bliska zeru. Oznacza to, że algorytm ma małą stratę – więc uczenie przebiega pomyślnie.

### Testowanie modelu

videoFolder = 'videos/'

def makeRecEnv*(envId: str, renderMode = 'rgb\_array')*:

env = gym.make(envId, render\_mode=renderMode)

env = AtariPreprocessing(env, frame\_skip=1)

env = FrameStack(env, 4)

env = Monitor(env)

return env

recording\_env = DummyVecEnv([lambda: makeRecEnv(envId)])

recording\_env = VecVideoRecorder(recording\_env, videoFolder, record\_video\_trigger=lambda x: x == 0, video\_length=10000, name\_prefix="dqn\_zaxxon")

modelPath = 'dqn\_zaxxon.pkl'

model = DQN.load(modelPath)

obs = recording\_env.reset()

for \_ in range(10000):

action, \_states = model.predict(obs)

obs, rewards, dones, infos = recording\_env.step(action)

if dones:

obs = recording\_env.reset()

recording\_env.close()

Podczas testowania modelu wykorzystany jest zapisany wcześniej model i utworzenie nowego, testowego środowiska, w którym powstanie nagranie działania programu. Podczas testowania zostały nagrane pliki mp4, w różnych miejscach testu, co prezentuje to jak naprawdę działa. Z tym etapem nastąpiły problemy z brakami pewnych programów. Trzeba było uczyć program na nowo po zainstalowaniu programu ffmpeg i biblioteki moviepy. Wymaganie biblioteki moviepy było szczególnie interesujące, ponieważ nie jest ona w żaden sposób importowana do programu.

## Wynik

A video game screen with a blue and brown box

Description automatically generated with medium confidence

Pomimo pomyślnych testów, algorytm nie radzi sobie najlepiej, jednakże sukcesem jest to, że udało się przeprowadzić uczenie od początku do końca – włącznie z nagraniem prezentacji działania programu.

# Wnioski

* Pomimo użycia mniejszej ilości kodu do napisania algorytmu uczenia ze wzmocnieniem, niż przy sieciach neuronowych, to napisanie tego było bardziej skomplikowane.
* Pomimo interesującego systemu nagród i kar, ciężej kontrolować to, czy taki algorytm na pewno uczy się dobrze.
* Biblioteka gymnasium od OpenAI zapewnia dużą ilość środowisk, które pomagają zagłębić się w tematykę uczenia ze wzmocnieniem.