Проект по предметот: **Агентно-базирани системи**

Тема: **Споредба на перформансите на DQN и PPO во CartPole-v1 и PPO и DDPG во Pendulum-v1 од библиотеката stable-baselines3**

Студент: **Александар Стојанов 211067**

Линк до кодот: https://github.com/Aleksandar-Stojanov/Agentno-bazirani-Sistemi-Proekt

Содржина

1. Апстракт

Со овој проект ја истражуваме ефикасноста на алгоритмите за Reinforcement Learning од библиотеката Stable-Baselines3, конкретно DQN, PPO и DDPG, применети на Gym околините CartPole-v1 и Pendulum-v1. Преку хиперпараметарска оптимизација со Optuna, се истражуваат и користат оптималните конфигурации за секој алгоритам користен во околините. Резултатите вклучуваат анализа на наградите по епизоди, споредба помеѓу алгоритмите и дискусија за предизвиците и перформансите. Овој проект придонесува кон разбирање на примената на Reinforcement Learning во различни околини со различен простор на состојби и акции, discrete како CartPole-v1 и continuous како Pendulum-v1.

2. Вовед

2.1 Проблем

Reinforcement Learning е еден од најважните пристапи во областа на машинското учење, но изборот на соодветен алгоритам и неговите хиперпараметри е многу важен за успешна примена на одредена околина. Различната сложеност на задачите, како што се оние претставени од Gym околините CartPole и Pendulum, бара постепено истражување за да се постигне максимална ефикасност.

2.2 Цел

Целта на проектот е да се споредат перформансите на алгоритмите DQN, PPO и DDPG на две различни околини. Преку хиперпараметарска оптимизација со Optuna, се настојува да се постигнат најдобрите можни резултати за секој алгоритам. Дополнително, се истражува како различните околини влијаат врз изборот на хиперпараметрите и постигнатите резултати. Изборот на Gym околините CartPole-v1 и Pendulum-v1 е поради нивната едноставност и истовремено нивната способност да покажат различни аспекти на RL алгоритмите. Овие околини претставуваат динамички системи кои бараат стратегиско донесување одлуки, што ги прави идеални за тестирање на RL техники.

2.3 Место во однос на сродни истражувања

Други истражувања во оваа област главно се фокусираат на примена на RL алгоритми на различни околини и на развој на нови методологии за оптимизација на хиперпараметрите. Овој проект гради врз тие основи, применувајќи стабилни алгоритми од Stable-Baselines3 и интегрирајќи ја Optuna за систематска оптимизација.

3. Сродни истражувања

3.1 „CartPole using Stable Baselines“

Овој труд ја истражува околината CartPole користејќи го алгоритамот PPO од библиотеката stable-baselines3 што на некој начин е дел од моето истражување. По ова истражување се добиени одлични перформанси на моделот во околината.

3.2 „Stable-Baselines3: Reliable Reinforcement Learning Implementations“

Овој труд повеќе е фокусиран на сите алгоритми и перформанси на Stable-Baselines3 и подобрување на библиотеката со текот на времето со цел давање на подобри перформанси. Овде се разгледани голем дел од алгоритмите од библиотеката. Со моето истражување е слично на тој начин што се користи истата библиотека и дел од алгоритмите.

3.3 „A COMPARATIVE STUDY OF DEEP REINFORCEMENT LEARNING MODELS: DQN VS PPO VS A2C“

Во овој труд се споредуваат алгоритмите DQN, PPO и A2C од Stable-Baselines3 во разни околини во Atari со споредба на вредностите на наградата во одредени епизоди и нивните перформанси. Сличноста со мојот проект е во користење на истите алгоритми од истата библиотека но во различна околина.

4. Опис на агентот

Проектот се базира на четири различни алгоритми за Reinforcement Learning, вклучувајќи DQN и PPO за средината CartPole-v1 и PPO и DDPG за средината Pendulum-v1. За овие алгоритми се користи библиотеката Stable-Baselines3, која обезбедува имплементации на различни reinforcement learning (RL) алгоритми.

1. DQN (Deep Q-Network):

DQN (Deep Q-Network) од Stable-Baselines3 е алгоритам за RL кој користи невронска мрежа за да научи Q-функција што одредува оптимални акции за максимизирање на наградата. Главните компоненти се:

a) Q-функција: Предвидува награда за дадена акција во одредена состојба.
b) Replay Buffer: Складира искуства за тренирање.
c) Целна мрежа: Обезбедува стабилност со периодично ажурирање од главната мрежа.
d) Експлорација-експлоатација: Користи є-greedy стратегија за балансирање.
e) Белманова равенка: Тренира со минимизирање на разликата помеѓу предвидените и целните Q-вредности.

DQN работи за средини со дискретни акции (на пример, CartPole). Кодот за имплементација е лесен за поставување и користење.

2. PPO (Proximal Policy Optimization):

PPO (Proximal Policy Optimization) е популарен алгоритам за засилувачко учење во Stable-Baselines3, познат по стабилноста и ефикасноста. Применет е на CartPole-v1 и Pendulum-v1, со прилагодени хиперпараметри како што се големината на околината и бројот на чекори. Главни карактеристики:

a) Политика: Користи невронска мрежа за моделирање на стратегија (policy) која одредува веројатност на акција за дадена состојба.
b) Clipping: Ограничува големи промени во политиката за да избегне нестабилност.
c) Advantage Function: Претставува разлика помеѓу набљудуваната награда и предвидената награда од вредносната функција, за подобра евалуација на акции.
d) On-Policy: Користи искуства само од тековната политика за тренирање.
e) Multi-environments: Ефикасно се тренира паралелно во повеќе средини.

PPO е погоден за континуирани и дискретни акции, како што се CartPole или Pendulum, поради неговата стабилност и едноставна имплементација.

3. DDPG (Deep Deterministic Policy Gradient):

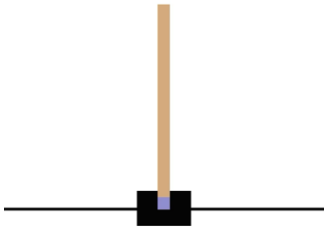
DDPG (Deep Deterministic Policy Gradient) е алгоритам за RL кој работи со континуирани простори на акција и користи off-policy метод. Главните карактеристики се:

a) Детерминистичка политика: Акциите се избираат директно од политиката без случајност.
b) Actor-Critic архитектура:
   Има две мрежи:
   - Actor: Одредува кои акции да се преземат.
   - Critic: Оценува колку се добри акциите на актерот.
c) Target мрежи: За стабилизирање на учењето, се користат target мрежи што се ажурираат бавно.
d) Replay buffer: DDPG учи од претходни искуства кои се чуваат во мемориски буфер, што ја зголемува ефикасноста на примерите.
e) Процес на обука: Critic се учи преку минимизирање на Белмановата грешката, а actor се учи преку максимизирање на проценките на критикот.

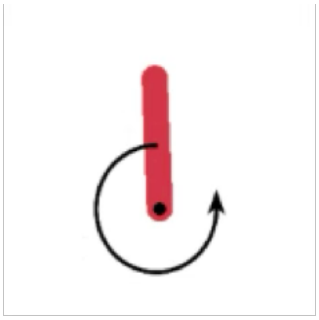
Предности: Работи со континуирани акции и има добра ефикасност на примерите.
Недостатоци: Чувствителен е на хиперпараметри и бара внимателно подесување.

Податочни Множества:

- За сите експерименти користени се стандардните RL средини од *gymnasium*, конкретно:
    - *CartPole-v1* - задача која има цел да го одржи столбот прав преку прилагодување на акциите на агентот.
    - *Pendulum-v1* - задача за континуирани акции, каде агентот треба да го задржи пентелумот во вертикална позиција.



слика 1. Околината CartPole-v1



слика 2. Околината Pendulum-v1

Користени Техники:

- **Optuna**: За автоматска оптимизација на хиперпараметри, се користи Optuna за да се најдат најдобри вредности за хиперпараметрите како што се стапката на учење, големината на баферот, и други параметри за PPO, DQN и DDPG.

Алатки и Ресурси:

- *Stable-Baselines3*: Користена е оваа библиотека за имплементација на алгоритмите DQN, PPO и DDPG.
- *Optuna*: Користена за хиперпараметрична оптимизација преку пробување на различни конфигурации.
- *Gymnasium*: За симулирање на средините на агенти.
- *Matplotlib*: За визуелизација на наградите на агентите во текот на тестирањето.

Параметри и Прилагодувања:

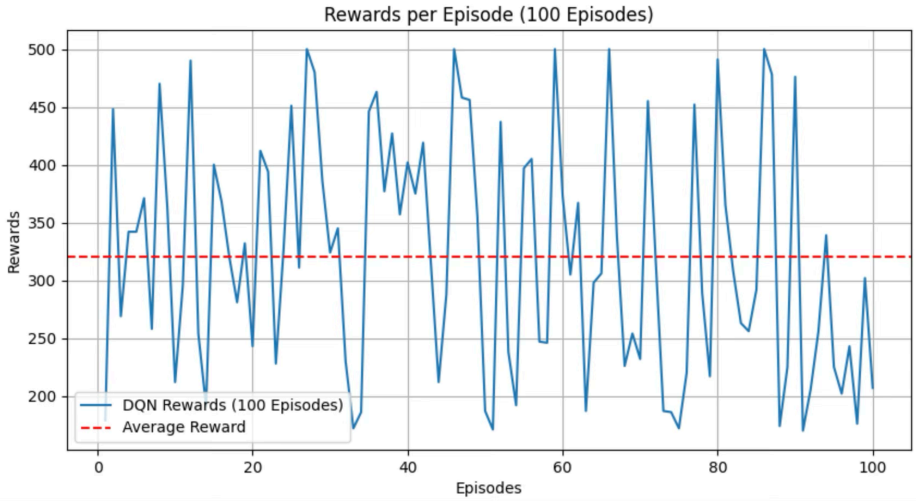
- **learning_rate**:
  - Одредување на ратата на учење, односно експериментирањето на моделот (преземањето на ризици)
- **batch_size**:
  - Специфицирање на бројот на примероци кои се користат при тренирањето.
- **buffer_size**:
  - Се однесува на големината на баферот каде се зачувани претходните искуства на моделот како состојби награди акции и слично.
- **gamma (Discount Factor)**:
  - Контролирање на важноста на идните добивки за разлика од добивките во моментот.
- **net_arch (Network Architecture)**:
  - Дефинирање на структурата на невронските мрежи.
- **exploration_fraction кај DQN**:
  - Помага при балансирање на испробувањето на нови акции во зависност со користење на веќе познатите.
- **n_steps кај PPO**:
  - Број на чекори потребни пред да се ажурира моделот.
- **ent_coef (Entropy Coefficient) кај PPO**:
  - Поттикнува ризик и истражување во однос на избирање веќе направени акции.
- **tau кај DDPG**:
  - Помага за стабилизирање на тренирањето.

5. Резултати

Сите агенти за двете околини се евалуирани на 100 епизоди, а се тренирани на n_trials=40 со total_timesteps = 300000.

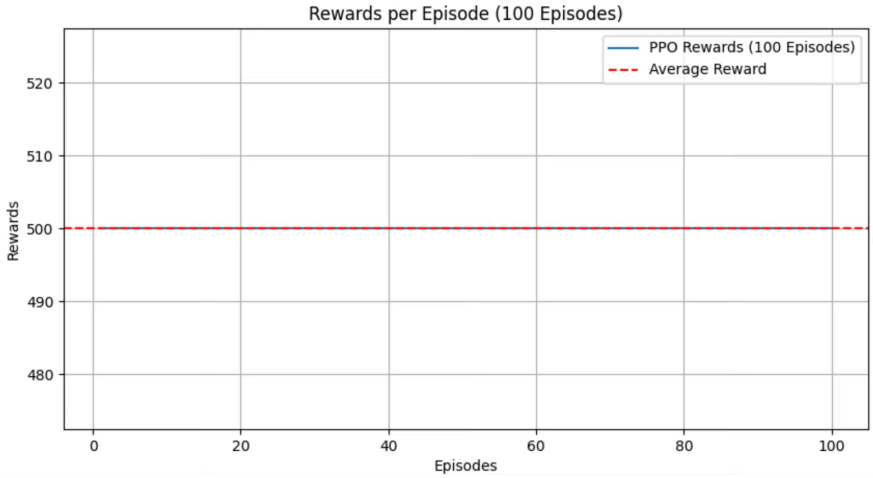
5.1 CartPole-v1

Резултатите за околината CartPole-v1 од алгоритмите DQN и PPO се различни. DQN по тренирањето со оптималните параметри, дава како резултат просечни резултати 320.82 со доста дивергенции во резултатите. Максималната добивка со овој агент е 500.



Слика 3. Евалуација на DQN за 100 епизоди на околината CartPole-v1

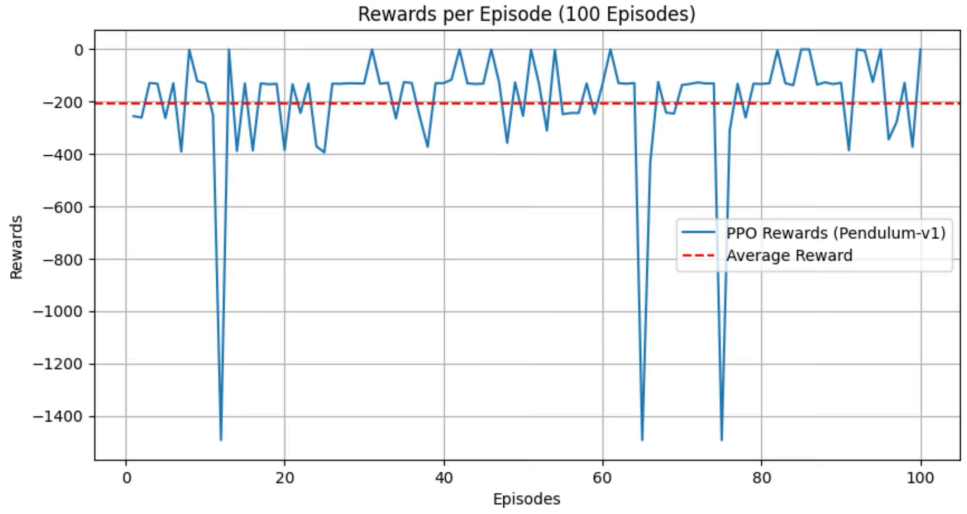
За разлика од резултатите со DQN, резултатите од PPO се одлични, односно во секоја од епизодите, моделот постигнува максимална добивка (500), што значи просечната и максималната добивка се поклопуваат.



Слика 4. Евалуација на PPO за 100 епизоди на околината CartPole-v1

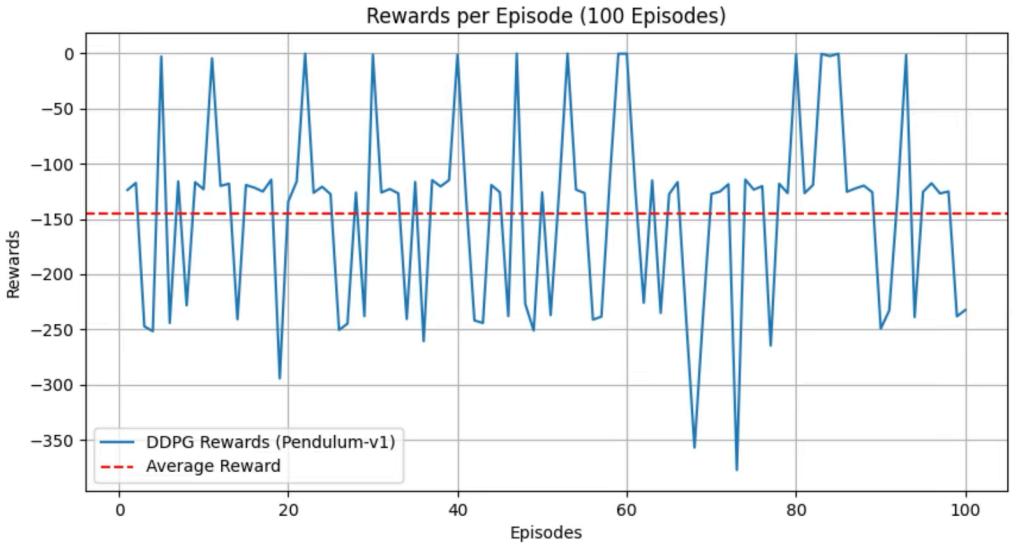
5.2 Pendulum-v1

Резултатите за околината Pendulum-v1 од алгоритмите PPO и DDPG не се разликуваат премногу. Овие резултати исто така не се оптимални. Со резултатите од PPO се добива просечен резултат со -205.83 вредност за добивка. Максимален резултат добива со вредност -1.33, но исто така има и поголема дивергенција на резултатите каде се добиваат и резултати до -1400.



Слика 5. Евалуација на PPO за 100 епизоди на околината Pendulum-v1

Резултатите од DDPG се за малку подобри каде како просечен резултат се добива -145.21, а за максимален резултат вредност од -0.16. Исто така нема многу дивергенција во резултатите како во PPO, така што овде најнискиот резултат кој се добива е околу -360.



Слика 6. Евалуација на DDPG за 100 епизоди на околината Pendulum-v1

Алгоритам	Околина	Макс. можна награда	Макс. награда	Просечна награда (100 епизоди)
DQN	CartPole-v1	500	500	320.82
PPO	CartPole-v1	500	500	500
PPO	Pendulum-v1	0	-1.33	-205.33
DDPG	Pendulum-v1	0	-0.16	-145.21

6. Заклучок

Овој проект покажа дека изборот на вистинскиот RL алгоритам може да доведе до подобри перформанси на моделот, а RL алгоритамот значително зависи од типот на околината во која работиме. DQN и PPO се успешни во дискретни задачи како што е CartPole-v1, но сепак PPO покажа далу подобри перформанси во околината. PPO ја покажува својата робусност и во континуирани околини како Pendulum-v1. Но сепак DDPG дава постабилни и подобри резултати во истата околина за споредба од PPO. DDPG, иако моќен, бара поголем напор за прилагодување. Оптимизацијата со Optuna беше клучна за подобрување на перформансите на сите алгоритми. Во иднина, соработка помеѓу различни RL техники може да доведе до понатамошни подобрувања и применливост.

Референци:

https://github.com/SwamiKannan/CartPole-using-Stable-Baselines/tree/main
https://www.jmlr.org/papers/volume22/20-1364/20-1364.pdf
https://arxiv.org/pdf/2407.14151
https://stable-baselines3.readthedocs.io/en/master/
https://optuna.readthedocs.io/en/stable/