Sprawozdanie z projektu
Podstawy Gymnasium — Q-Learning i TD3

Michał Burda, Kamil Poniewierski

13 maja 2025

1 Wstep

Celem projektu było zapoznanie sie z biblioteka `Gymnasium` oraz implementacja algorytmów uczenia ze wzmocnieniem (Reinforcement Learning) w środowiskach dyskretnym i ciagłym.

Zrealizowano dwa eksperymenty:

- **Zadanie za 6 punktów:** środowisko `CliffWalking-v0` i algorytm Q-learning.
- **Zadanie za 8 punktów:** środowisko `MountainCarContinuous-v0` i algorytm TD3.

2 Zadanie 1: CliffWalking z Q-learningiem

2.1 Opis środowiska

Środowisko `CliffWalking-v0` to klasyczny przykład siatki, po której agent ma przejść z punktu startowego do celu, unikajac "klifów", które daja duża kare.

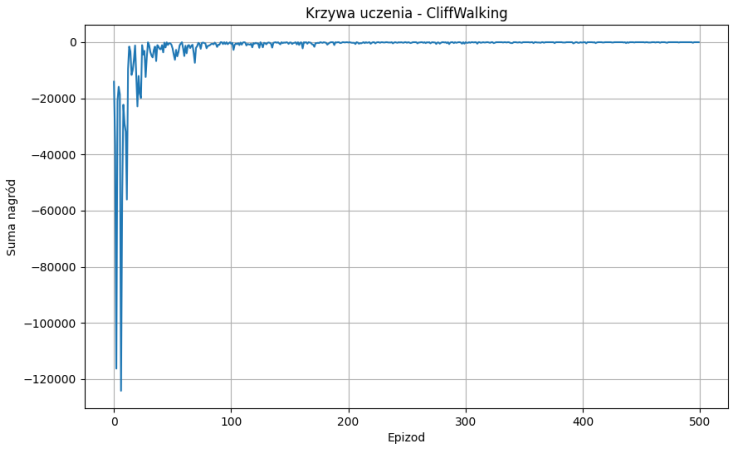
2.2 Parametry i implementacja

- Współczynnik uczenia $\alpha = 0.1$
- Współczynnik dyskontowy $\gamma = 0.9$
- Poczatkowe epsilon: 1.0, minimalne: 0.01, współczynnik zmniejszania: 0.995

Użyto algorytmu **Q-learning**, którego aktualizacja oparta była o równanie Bellmana:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left(r + \gamma \max_{a'} Q(s',a') - Q(s,a) \right)$$

2.3 Wyniki i obserwacje



Rysunek 1: Krzywa uczenia Q-learning w środowisku CliffWalking

Agent poczatkowo czesto wpadał w "klif", ale z czasem zaczał uczyć sie bezpiecznej drogi. Współczynnik eksploracji malał, co skutkowało stabilniejszymi wynikami.

3 Zadanie 2: MountainCarContinuous z TD3 i PPO

3.1 Opis środowiska

Środowisko `MountainCarContinuous-v0` to klasyczny problem z ciagła przestrzenia stanów i akcji. Agent steruje samochodem, który musi wjechać na góre po lewej stronie, rozpedzajac sie poprzez ruchy tam i z powrotem. Zbyt słaby silnik nie pozwala na bezpośredni podjazd, dlatego agent musi nauczyć sie odpowiedniego rozpedu.

3.2 Zastosowane algorytmy

W eksperymencie porównano dwa podejścia:

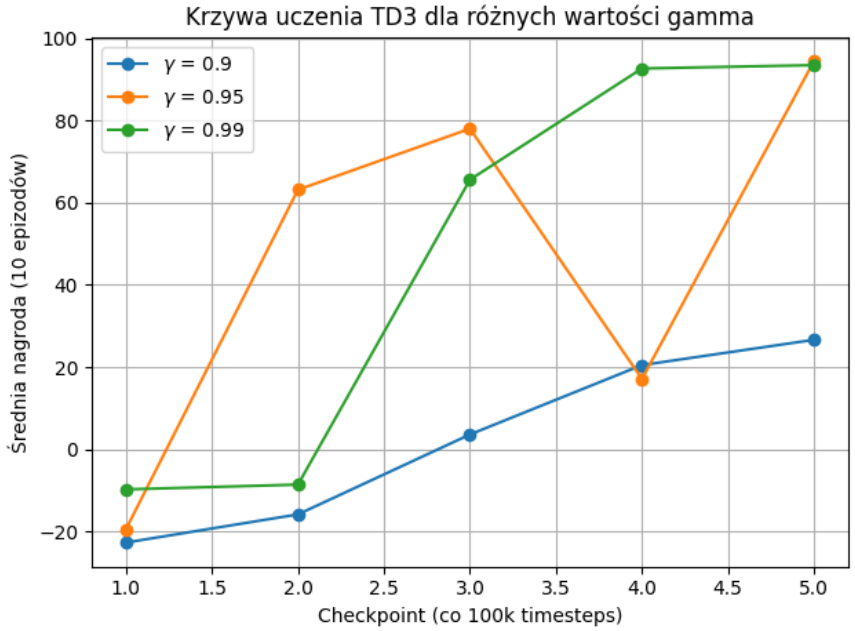
- **TD3 (Twin Delayed Deep Deterministic Policy Gradient)** — deterministyczny algorytm off-policy dobrze radzacy sobie w środowiskach z ciagła przestrzenia akcji.
- **PPO (Proximal Policy Optimization)** — popularny algorytm on-policy oparty na stochastycznej polityce.

3.3 Parametry eksperymentu

Dla TD3 przetestowano trzy wartości współczynnika dyskontowego: $\gamma = 0.9$, $\gamma = 0.95$, $\gamma = 0.99$. Pozostałe parametry:

- Liczba timesteps: 500 000 (5 etapów po 100 000)
- Architektura sieci: dwie warstwy po 256 neuronów
- Learning rate: $3 \cdot 10^{-4}$
- Szum eksploracyjny: Ornstein-Uhlenbeck

3.4 Wyniki TD3

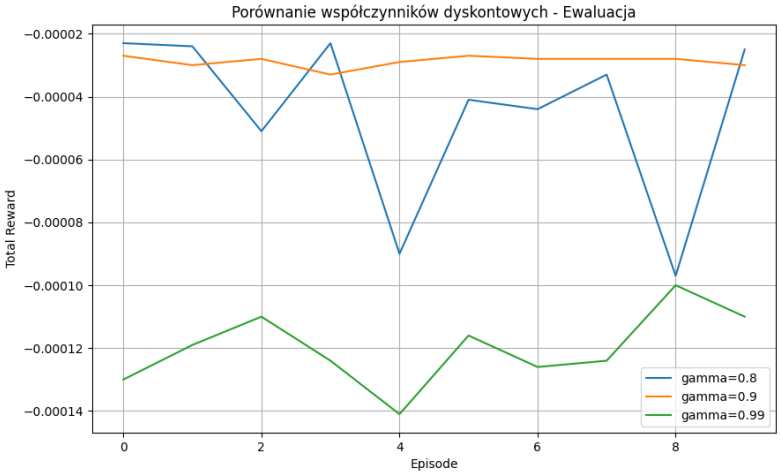


Rysunek 2: Krzywa uczenia TD3 dla różnych wartości gamma

Wyniki wskazuja, że:

- Agent osiagnał cel (dotarcie na szczyt) dla wszystkich testowanych wartości γ — 0.9, 0.95 oraz 0.99.
- Najwyższe i najbardziej stabilne nagrody uzyskano przy $\gamma = 0.95$ i $\gamma = 0.99$.
- Wartość $\gamma = 0.9$ również pozwoliła na nauke skutecznej strategii, choć z niższa jakościa nagród końcowych.

3.5 Wyniki PPO



Rysunek 3: Krzywe uczenia PPO — brak osiagniecia celu dla żadnej wartości γ

W przypadku PPO, mimo prób z różnymi konfiguracjami i współczynnikami γ, agent nie był w stanie osiagnać celu w środowisku. Uzyskiwane nagrody były znacznie niższe, a polityka nie doprowadzała pojazdu do szczytu.

3.6 Wnioski

Z eksperymentów wynika, że:

- Algorytm TD3 skutecznie nauczył agenta osiagać cel dla wszystkich testowanych wartości γ.
- PPO okazał sie nieskuteczny w tym środowisku, prawdopodobnie ze wzgledu na stochastyczny charakter polityki i trudności w eksploracji w środowiskach z ciagłymi akcjami.
- Wysoka wartość γ (0.95–0.99) wspierała lepsze wyniki końcowe.