Sprawozdanie - Laboratorium 3

Inteligencja Obliczeniowa w Analizie Danych Cyfrowych

Autorzy:
Jakub Kubicki
Eryk Mikołajek
Marcin Zub

1. Cel ćwiczenia

- Napisanie programu rozwiązującego problem Inverted pendulum
- Wykorzystanie Gymnasium podczas implementacji
- Użycie sieci neuronowych do rozwiązania problemu

2. Opis problemu

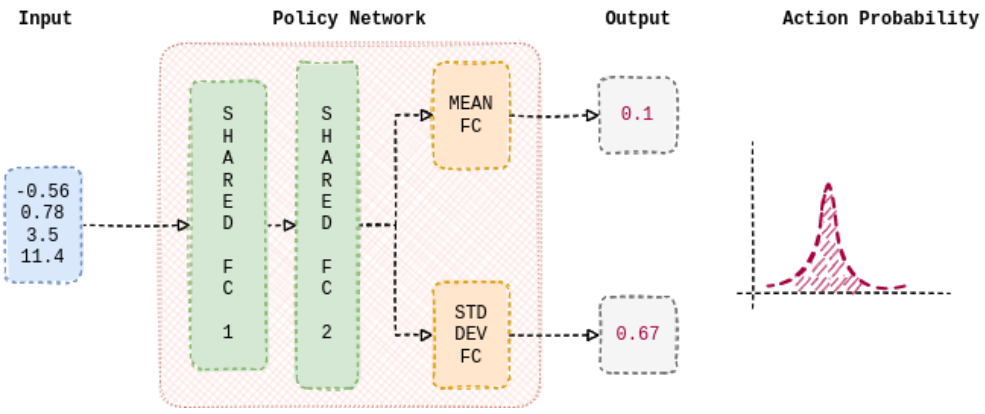
Inverted pendulum to problem polegający na utrzymaniu pionowej pozycji wahadła przymocowanego do wózka poruszającego się poziomo, mimo naturalnej tendencji do upadku. Wyzwanie polega na precyzyjnym sterowaniu siłą wózka, by przeciwdziałać grawitacji i zachować równowagę, używając równań ruchu i metod kontroli.

3. Realizacja rozwiązania

Projekt został zrealizowany w języku Python. Wykorzystaliśmy narzędzie Gymnasium oraz zaawansowaną sieć neuronową.

Sieć neuronowa

Parameterized Policy Network to sieć, która przyjmuje parametry wejściowe w postaci wektora, aby następnie za ich pomocą wygenerować decyzję. W odniesieniu do naszego problemu, sieć ta będzie uczyć się odpowiedniej strategii ruchów, które pozwolą na pionowe utrzymanie wahadła. Nasz model ma na celu naukę polityki (policy) sterowania, która minimalizuje energię zużytą do utrzymania wahadła w pożądanym położeniu.



Algorytm
Użyty przez nas algorytm nosi nazwę REINFORCE. Jest to skrót od REward Increment = Non Negative Factor × Offset Reinforcement × Characteristic Eligibility. To jeden z najbardziej popularnych algorytmów stosowanych podczas uczenia ze wzmocnieniem. Algorytm działa na zasadzie interakcji ze środowiskiem. Podczas każdej akcji, przyjmuje on obserwacje ze środowiska, aby następnie zwrócić rozkład prawdopodobieństwa dla danej akcji. Następnie obliczana jest nagroda, która może przyjąć postać sumy wszystkich nagród z danego epizodu. Na jej bazie aktualizowane są wag sieci.

Implementacja
Nasza implementacja składa się z dwóch klas:

- *Policy Network*
- *REINFORCE*

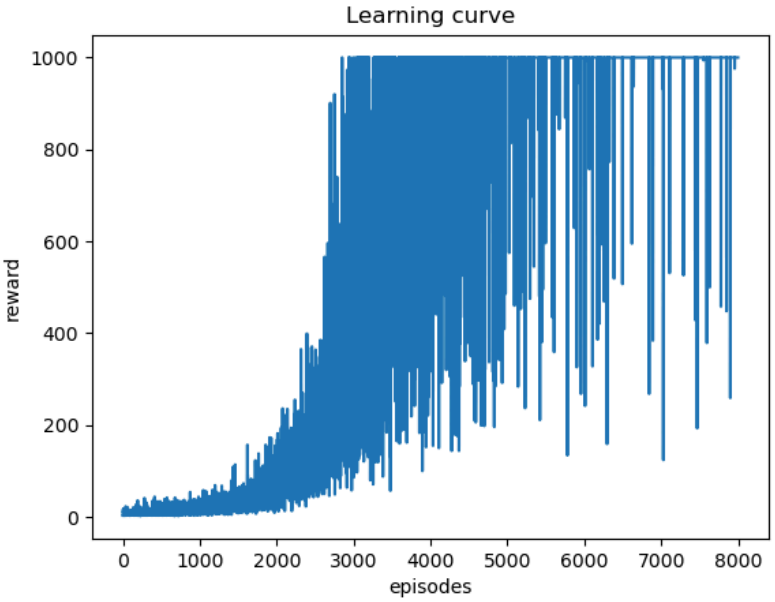
Klasa Policy Network definiuje sieć neuronową służącą do oszacowania średniej i odchylenia standardowego rozkładu normalnego, z którego próbkowany jest działanie. Sieć ta składa się z trzech warstw liniowych połączonych z funkcjami aktywacji Tanh(), gdzie pierwsze dwie warstwy tworzą wspólne cechy, a ostatnia warstwa odpowiada za obliczenie średniej i odchylenia standardowego.

Klasa REINFORCE implementuje algorytm REINFORCE wraz z metodami do próbkowania akcji z wykorzystaniem sieci polityki oraz aktualizacji wag sieci w oparciu o zgromadzone nagrody.

4. Statystyki

Discount factor: zgodnie z założeniami w początkowej fazie przyjęliśmy wartość 0.9, jednakże po zamianie na wartość 0.99 otrzymaliśmy lepsze wyniki, dlatego też ostatecznie przyjęliśmy tą wartość.

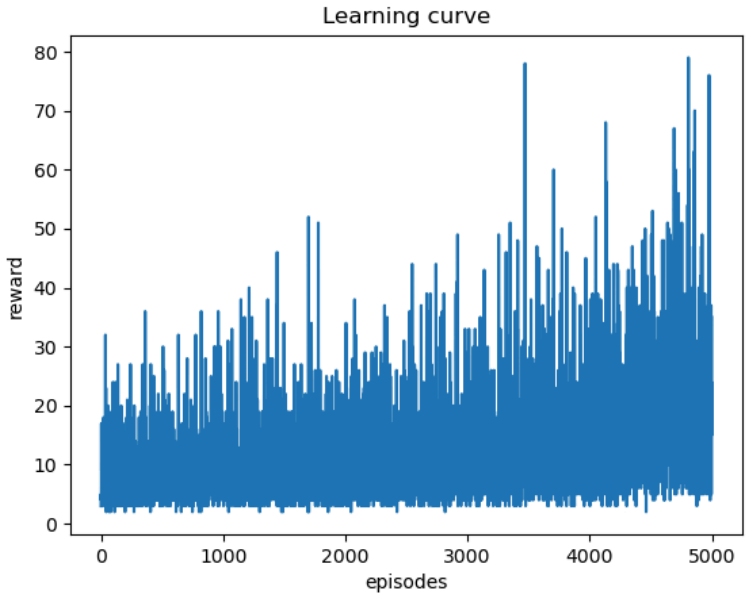
Learning curve wykres:
Dla współczynnika discount factor - 0.99



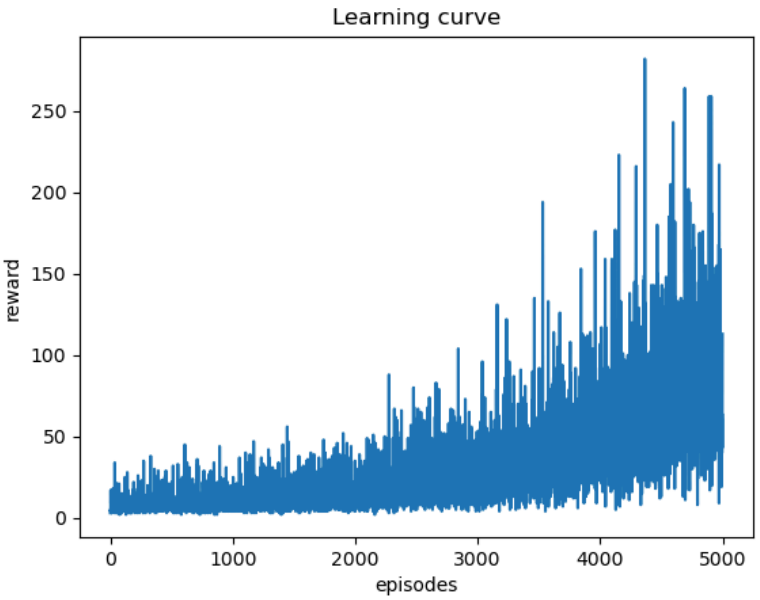
Zauważyć można odcięcie przy nagrodzie równej 1000 - jest to charakterystyka konkretnego środowiska, które nie pozwala na lepszą nagrodę.

Różne wartości współczynnika discount factor:

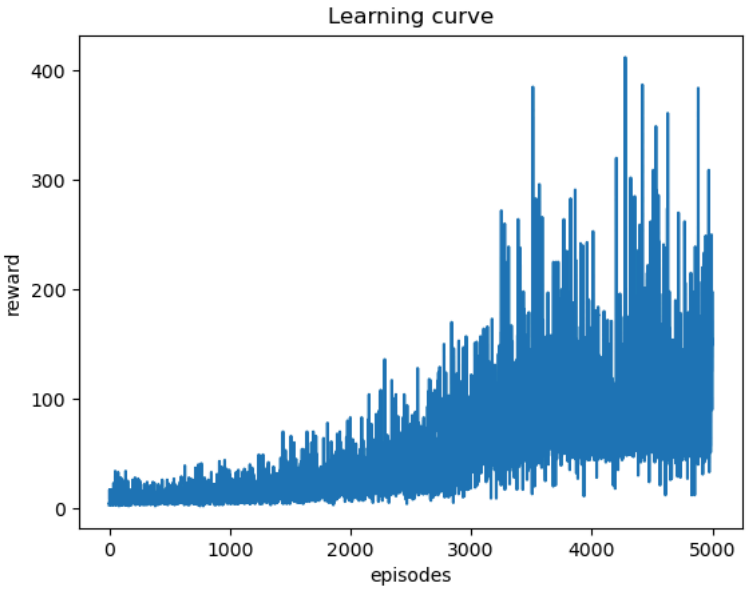
Discount factor: 0.1
Średnia wartość *reward* pod koniec uczenia: 19
Learning curve:



Discount factor: 0.5
Średnia wartość *reward* pod koniec uczenia: 80
Learning curve:



Discount factor: 0.7:
Średnia wartość *reward* pod koniec uczenia: 117
Learning curve:



5. Wnioski

Przetestowano zachowanie modeli dla różnych wartości współczynnika discount factor. Wpływa on na sposób, w jaki agent uwzględnia przyszłe nagrody w procesie podejmowania decyzji. Wraz ze wzrostem wartości w.w współczynnika, średnia wartość dla *reward* pod koniec uczenia maleje. Współczynnik ma też istotny wpływ na stabilność samego uczenia modelu.