Opis projektu zaliczeniowego z przedmiotu WdSI
Mateusz Romel, nr indeksu 155047

State of Art

1. Pierwszym z rozwiązań jest Q-learning Algorithm. Jest to metoda uczenia ze wzmocnieniem, która polega na budowaniu tabeli Q, gdzie każda komórka określa wartość stanu i akcji. Algorytm iteracyjnie aktualizuje wartości Q, aby z czasem uczyć się optymalnej polityki decyzyjnej.

Zalety rozwiązania:
-Skuteczny w środowiskach o nieznanej dynamice, ponieważ nie wymaga modelu środowiska.

Wady rozwiązania:
- Może być nieefektywny w stanach o dużej przestrzeni, ponieważ wymaga tabeli Q dla wszystkich możliwych stanów i akcji.
- Złożoność obliczeniowa rośnie wraz ze wzrostem liczby stanów i akcji.

2. Kolejnym rozwiązaniem jest AlphaZero Algorithm. Jest to algorytm opracowany przez DeepMind, który łączy Monte Carlo Tree Search (MCTS) z sieciami neuronowymi. AlphaZero uczy się grać na poziomie mistrzowskim w wielu grach planszowych poprzez samouczenie się i generowanie swoich danych treningowych.

Zalety rozwiązania:
-Niezależność od specyfiki gry; AlphaZero może nauczyć się dowolnej gry przez samodzielne granie.
-Wysoka wydajność i skuteczność w grach o złożonej przestrzeni stanów, takich jak szachy czy Go.

Wady rozwiązania:
-Wymaga ogromnej mocy obliczeniowej
-Proces treningowy może być czasochłonny

3.Ostatnim trzecim rozwiązaniem jest MuZero Algorithm. Jest to zaawansowana wersja AlphaZero, opracowana również przez DeepMind. MuZero dodatkowo uczy się modelu środowiska, co pozwala mu efektywnie podejmować decyzje bez znajomości zasad gry.

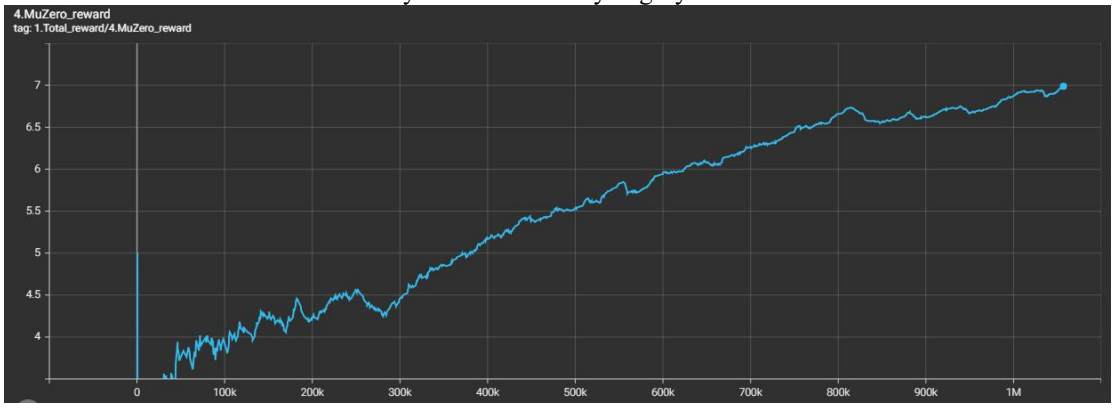
Zalety rozwiązania:
-Zdolność do nauki bez znajomości zasad gry, co czyni go bardziej uniwersalnym.
-Wykorzystanie modelu środowiska do przewidywania przyszłych stanów, co zwiększa skuteczność strategii.

Wady rozwiązania:
-Jeszcze większe wymagania obliczeniowe i zasobowe w porównaniu do AlphaZero.

Opis wybranej koncepcji

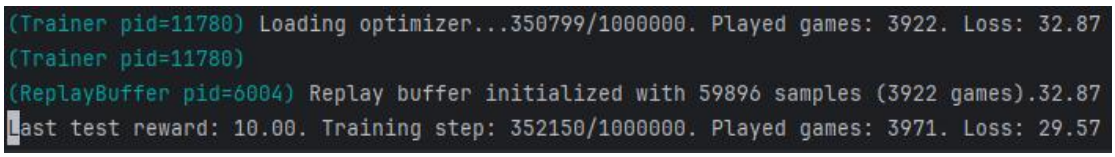
Wybrana została koncepcja wykorzystująca algorytm MuZero, ponieważ sama gra posiada około 4,5 tryliona możliwych pozycji. Aby algorytm działał poprawnie należało mieć gotowy skrypt connect4.py, który implementuje zasady działające w grze.
Jako wyjście algorytmu możliwe jest wykorzystanie tensorboard oraz obserwowanie nagrody dla MuZero. Ta w trakcie nauki rośnie. Gdy osiągnie 10 model powinien mieć 100 procent skuteczności. Możliwe jest również rozegranie gry z wcześniej nauczonym modelem.
MuZero uczy się modelu środowiska gry oraz strategii gry jednocześnie. Metoda składa się z kilku kluczowych kroków. Po pierwsze MuZero uczy się modelu, który przewiduje przyszłe stany gry na podstawie obecnych stanów i akcji. Po drugie Algorytm korzysta z Monte Carlo Tree Search (MCTS) do eksploracji możliwych stanów gry i wyboru najlepszych akcji. Na koniec MuZero optymalizuje swoją politykę i funkcję wartości poprzez iteracyjne aktualizacje na podstawie wyników symulacji gier.
Aby zrealizować działanie tego algorytmu niezbędna jest karta graficzna o dużej mocy obliczeniowej. Niestety takiej możliwości zabrakło, ale została wykorzystana słabsza karta graficzna.
Procedura testowania wyglądała następująco. Po pobraniu projektu dostępnego na https://github.com/werner-duvaud/muzero-general bardzo duża część czasu została poświęcona odpowiedniej instalacji bibliotek. Zdecydowano się na Python 3.8 Ponieważ najlepiej współpracował z opisanymi wersjami bibliotek. Po wykonaniu instrukcji instalacji bibliotek według README.md kontynuowano uruchomienie algorytmu. Wybrano naukę gry connect4. Trening przebiegał dosyć długo. Bołącznie 8 dni. Algorytm z małą przerwą był w stanie osiągnąć następujące wyniki:

Rys.1 Pierwsze testy algorytmu



Rys. 2 kontynuacja nauki algorytmu.

Jak można zauważyć na powyższych wykresach algorytm coraz częściej zdobywa nagrodę 10 za wygraną gdzie za przegraną otrzymuje 0. Wykresy zostały wygładzone i znajduje się na nich chwilowa wartość średnia nagrody. Mimo kilku dni treningu algorytm z małą mocą obliczeniową był w stanie tylko osiągnąć pułap 7 punktów. Informacje, które daje algorytm w lini komend można zobaczyć na rysunku 3.



Rys.3 Dalsza nauka algorytmu

Z przetrenowanym modelem można również zagrać używając opcji „Play against Muzero”.

Źródła:

1. 26646701.pdf
2. 1911.08265v2.pdf
3. werner-duvaud/muzero-general: MuZero