

# Raport projektu - Connect Four

Mateusz Tkaczyk  
Jakub Rudnik  
Piotr Kurosad  
Filip Łojek

8 marca 2024

## 1 Wprowadzenie do projektu

### 1.1 Technologie

python3.11

### 1.2 Narzędzia

- tensorflow,
- numpy (ML),
- pygame (GUI).

### 1.3 Opis

Gra w klasyczną wersję *Connect Four* z desktopowym GUI. Będziemy grali przeciwko algorytmowi opartemu o rozwiązanie ML. Do wyboru będą dwa modele:

- klasyfikacyjny, w którym odpowiedzią modelu będzie ruch (dokładnie jego indeks od 0 do 6),
- regresyjny, w którym odpowiedzią modelu będzie ewaluacja danej pozycji.

Danymi wejściowymi będzie plansza gry oraz informacja o tym, kto wykonuje następny ruch. Możliwe, że do modeli zostaną dostarczone dodatkowe informacje w zależności od potrzeb. Dane treningowe/ewaluacyjne zostaną wygenerowane przez jeden z dostępnych silników do gry w *Connect Four*.

## 1.4 Zbieranie danych

Dane zostały zebrane za pomocą optymalnego bota do connect4. Ma on przeliczoną całą grę i mamy pewność, że dla każdej pozycji zwróci on nam dokładną ewaluację każdego ruchu. Ewaluuje on pozycje w następujący sposób (zakładając optymalny przebieg rozgrywki po każdym ruchu):

- Jeśli po danym ruchu jesteśmy w stanie wygrać grę, to wynik tej ewaluacji jest dodatni.
- Jeśli po danym ruchu jesteśmy w stanie tylko przegrać, wynik ewaluacji jest ujemny.
- Jeśli dany ruch prowadzi do remisu, wynik ewaluacji to 0.
- Im wyższy wynik dodatni, tym szybciej po danym ruchu możemy wygrać, im bardziej ujemny wynik tym szybciej przegrywamy.

Dane zbieramy w następującym formacie:

opis\_pozycji,ewaluacja<sub>1</sub>,ewaluacja<sub>2</sub>,...,ewaluacja<sub>7</sub>

gdzie opis\_pozycji jest wypisanymi jeden po drugim liczbami, oznaczającymi kolejne ruchy (indeksy kolumn, do których wrzucane były żetony) prowadzące do obecnej pozycji. Ewaluacja<sub>i</sub> oznacza natomiast ewaluację ruchu do *i*-tej kolumny przy obecnej pozycji. Zdecydowaliśmy się dodatkowo na zebranie danych następującymi sposobami:

- Patrząc tylko na możliwe optymalne ruchy, budujemy drzewo optymalnych strategii.
- Patrząc na wszystkie możliwe rozgrywki do danej głębokości, wygenerowaliśmy wszystkie rozgrywki do 8 ruchów od początku gry.
- Generując zupełnie losowe rozgrywki.
- Generując losowe rozgrywki, ale im lepszy ruch, tym większe prawdopodobieństwo mamy na jego wybranie.

Przy każdym z tych sposobów, każda wygenerowana pozycja ma przypisaną ewaluację każdego możliwego ruchu. Dzięki takiemu generowaniu danych, mamy nadzieję, że bot będzie w stanie poradzić sobie z graczem grającym dobre ruchy, ale jednocześnie powinien umieć wykorzystać błędy swojego przeciwnika.