Uczenie A.I. grać w Tetris

Jakub Zieliński

Cel projektu

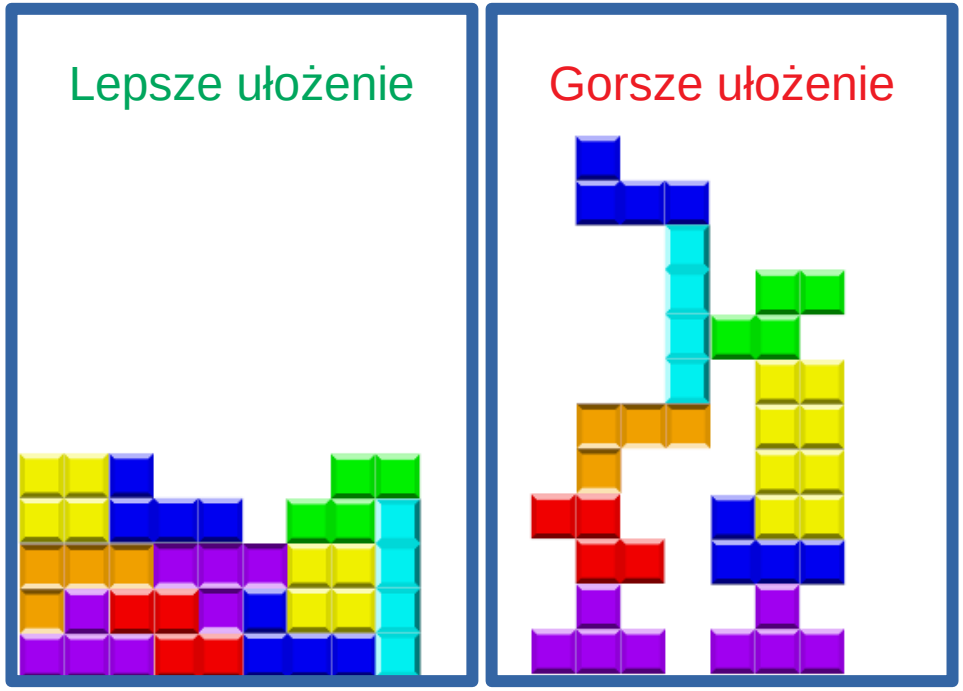
- Otrzymać sieć nauronową, która będzie w stanie wyczyścić przynajmniej jedną linię w Tetris
- Stworzyć gracza “idealnego”

Plan projektu

- Storzenie gry Tetris w Python’ie
- Wykorzystanie modułu NEAT-python do neuroewolucji sieci
  - Zdefiniowanie funkcji dopasowania
- Nauczenie sieci przez N generacji, aż otrzyma się kompetentnego gracza

Definiowanie funkcji dopasowania

- + otrzymany wynik gry
- + długość bloków obok siebie (ułożone linie)
- - liczba dziur na planszy
- - wysokość ustawionych tetrimino
- - Przedwczesne przegranie



Sieć

- 232 wejścia: ułożony grid 20x10 + kontrolowany tetrimino 4x4 + następny tetrimino 4x4)
- 40 wyjść: wszystkie możliwe kombinacje położenia (10) i rotacji tertimino (4)

Metoda nauki

- Losujemy N zestawów (gier) po P tetrimino dla każdej generacji
- Rozegranie gier przez O osobników i policzenie dopasowania po zakończeniu (uśrednione przez liczbę gier)
- Wykonujemy G generacji aby otrzymać najlepszego gracza

Wykonane nauki

- 10 zestawów po 20 tetrimino
- 200 generacji po 40 osobników
- Brak zmian w funkcji dopasowania

- 10 zestawów po 20 tetrimino
- 200 generacji po 80 osobników
- Brak zmian w funckji dopasowania

- 20 zestawów po 20 tetrimino
- 200 (+300) generacji po 40 osobników
- Brak zmian w funkcji dopasowania

- 10 zestawów po 20 tetrimino
- 300 generacji po 80 osobników
- Zwiększenie kary za dziury

Nauka 5

- Zmieniona sieć (214 wejść, inny sposób definicji inputu)
- Zmiana funkcji dopasowania (zamiast długości linii, sprawdzamy różnice poziomów)
- Zwiększenie liczby zestawów