Michał Kępa, Michał Maras

Algorytmy ewolucyjne

Neuroewolucja w grach

Projekt zaliczeniowy

Cel

Celem projektu jest wykorzystanie algorytmów ewolucyjnych jako sposobu trenowania sieci neuronowych, grających w następujące gry:

- CartPole – OpenAI Gym benchmark
- Pong – OpenAI Atari Gym oraz własna implementacja
- 2048 – własna implementacja
- CarRacing – własna implementacja, OpenAI Gym (być może)

Do trenowania sieci neuronowych zostaną użyte algorytmy: ES(μ, λ), CMA-ES, NEAT oraz LM-MA-ES.

Sieć neuronowa jako agent w grze

Sieć neuronowa może być wykorzystana jako agent grający w grę. W każdej chwili (turze, klatce) agent ma za zadanie na podstawie obecnego stanu gry wykonać jeden z dozwolonych w grze ruchów. Dla reprezentacji stanu sieć (tzw. policy network) oblicza najkorzystnieszą politykę akcji – rozkład prawdopodobieństwa, na podstawie którego losowana jest akcja.

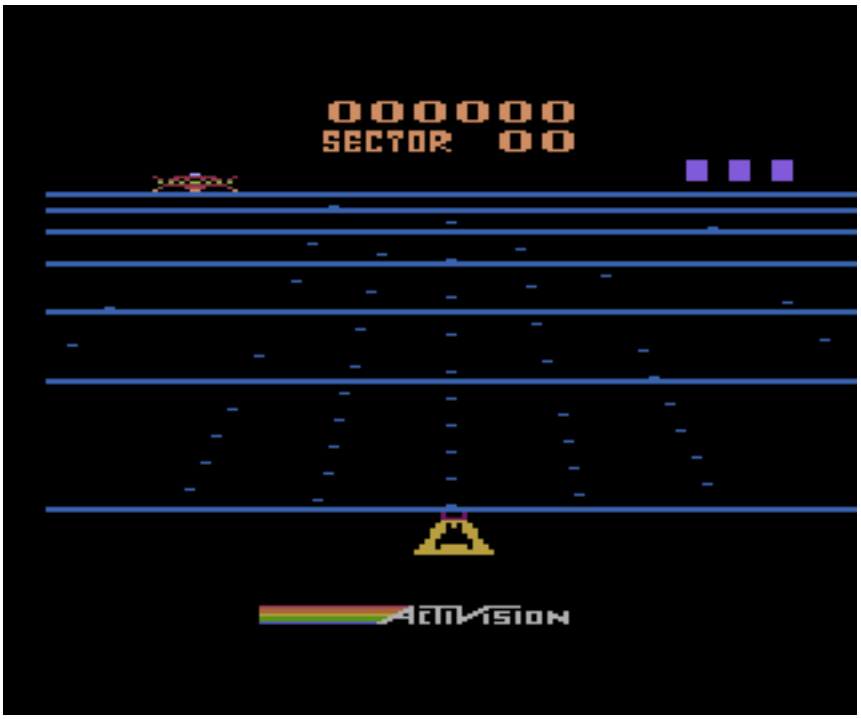
Ustalenie wag w sieci neuronowej jest problemem optymalizacyjnym – wagi mogą być reprezentowane przez wektor w przestrzeni $\mathbb{R}^n$. Jeżeli zdefiniujemy dla wektora wag funkcję przystosowania, możemy wykorzystać do tego problemu algorytmy ewolucyjne. Więcej informacji odnośnie reprezentacji stanu, funkcji przystosowania, topologii sieci oraz uzyskanych wyników dla każdej z gier znajduje się w odpowiednich sekcjach poniżej.

Pong

Beam Rider (RAM, OpenAI Gym)

Wstęp:
Podjęliśmy próbę wykorzystania algorytmu CMA-ES do optymalizacji wag sieci grającej w grę Beam Rider dostępną w OpenAI Gym. Do tego zadania postanowiliśmy użyć zawartości pamięci RAM maszyny Atari jako wejścia sieci neuronowej. Atari 2600, na którym uruchamiana jest gra, posiada 128 bajtów pamięci RAM. Z racji tego, że jest to

maszyna 8-bitowa, stan gry reprezentuje 128 liczb. Takie sformułowanie problemu wymaga znacznie większej sieci neuronowej, a co za tym idzie znacznie większej wymiarowości problemu optymalizacyjnego. Zdecyowaliśmy się na sieć z jedną ukrytą warstwą posiadającą 64 neurony oraz 4 neuronami wyjścia – wektor wag ma długość równą 8516.



Opis gry:
W grze gracz steruje statkiem kosmicznym i ma za zadanie wyeliminować wszystkich przeciwników. Dostępne akcje to poruszanie statkiem w lewo lub prawo, strzał oraz użycie torpedy. Torped można używać jedynie trzy razy w każdym poziomie. Gracz ma trzy życia, traci jedno za każdym razem gdy uderzy w pocisk przeciwnika. Aby przejść do następnego poziomu, gracz musi pokonać lub ominąć określoną liczbę przeciwników. Gracz otrzymuje punkty za zniszczenie przeciwnika.

Funkcje przystosowania i wyniki:
Niestety, algorytm CMA-ES wymaga obliczenia wartości własnych macierzy kowariancji, co wymaga $O(n^3)$ operacji. Przy wymiarowości problemu równej 8516 macierz kowariancji ma prawie 80 milionów liczb, co powoduje znaczne problemy. Mimo tego, udało nam się wyznaczyć 500 generacji.

Pierwsza wykorzystana przez nas funkcja przystosowania to liczba klatek, które agent przetrwał. Wynikała ona głównie z niezrozumienia mechanik gry (zdaje się, że gracz musi pokonać pewnych przeciwników, aby przejść dalej; może ich jednak omijać w nieskończoność), lecz spowodowała niespodziewane zachowanie agenta. Po niecałych 100 generacjach wszyscy agenci zaczęli otrzymywać stałą wartość równą 10000. Było to spowodowane tym, że aby rozpocząć grę należało wybrać dowolną akcję inną niż strzał. Gdy gracz nie rozpoczął gry przez 10000 klatek, gra sama się wyłączała. Z racji tego, że ten wynik był znacznie lepszy niż wszystkie wcześniej osiągnięte przez agenta, wagi zostały dopasowane tak, by szansa na wybranie strzału na początku gry była bliska 100%. Agent nigdy nie grał w grę i otrzymywał za to stosunkowo wysoki wynik.

Po zmianie funkcji przystosowania na sumę wyniku z gry i liczby klatek przemnożonej przez małą stałą, agent decydował się uruchamiać grę zawsze. Już po kilku generacjach szansa na ruch w jednym kierunku znacznie malała – agent używał jedynie trzech pozostałych akcji. Po około 200 generacjach agent przestał również poruszać się w drugim kierunku, pozostawając ciągle w tym samym miejscu. W dalszych generacjach, z drobnymi wyjątkami, nie było widać większych zmian.