Uczenie ze wzmocnieniem w implementacji bota do gry StarCraft 2

SI w grach komputerowych

Michał Dams, Krzysztof Szymaniak, Maja Grabowska

Politechnika Wrocławska
Wydział Informatyki i Telekomunikacji

1 Wstęp

Starcraft 2 to popularna strategia czasu rzeczywistego wyprodukowana przez firmę Blizzard Entertainment. Gra toczy się w futurystycznym świecie, w którym gracze kontrolują jedną z trzech ras: ludzi, protosów lub zergów. Gracze muszą zbierać surowce, budować swoje bazy, szkolić jednostki i prowadzić taktyczne bitwy przeciwko innym graczom lub w trybie fabularnym przeciwko komputerowym przeciwnikom. Starcraft 2 jest znany ze złożonej mechaniki gry.

1.1 Zastosowanie SI w Starcraft 2

Sztuczna inteligencja (AI) znajduje zastosowanie w botach w grze Starcraft 2. Boty to programy komputerowe, które mogą grać w grę w imieniu gracza lub działać jako przeciwnicy komputerowi. AI pozwala na stworzenie botów, które potrafią samodzielnie podejmować decyzje i działać w sposób zbliżony do człowieka, dzięki czemu gra z nimi jest bardziej realistyczna i wymagająca.

AI w botach Starcraft 2 może być wykorzystana na wiele sposobów, od prostych taktyk do bardziej złożonych strategii, które uwzględniają różne czynniki, takie jak ilość surowców, jednostek wrogów i teren. AI może również pomóc botom w podejmowaniu decyzji dotyczących budowy baz, szkolenia jednostek i przemieszczania ich po mapie.

Ponadto, AI może być wykorzystana do uczenia botów poprzez tzw. uczenie maszynowe, co oznacza, że bot może nauczyć się grać w grę poprzez analizę dużych ilości danych i doświadczeń. Dzięki temu boty mogą stać się jeszcze bardziej wyrafinowane i zagrażające dla graczy ludzkich, co może przyczynić się do dalszego rozwoju i udoskonalania gier strategicznych.

1.2 AlphaStar

AlphaStar to sztuczna inteligencja stworzona przez firmę DeepMind, która została zaprojektowana do grania w grę Starcraft 2. AlphaStar wykorzystuje zaawansowane technologie uczenia maszynowego i głęboką sieć neuronową, aby nauczyć się strategii i taktyk potrzebnych do zwycięstwa w grze.

AlphaStar został stworzony jako projekt badawczy, którego celem było wykorzystanie uczenia maszynowego do osiągnięcia wyższego poziomu rozgrywki w grze Starcraft 2. Aby osiągnąć ten cel, naukowcy z DeepMind stworzyli wiele wersji AlphaStar, które uczyły się grać w grę poprzez symulacje i naukę na podstawie dużej ilości danych.

W 2019 roku AlphaStar stał się pierwszą sztuczną inteligencją, która pokonała profesjonalnych graczy w grze Starcraft 2 na poziomie mistrzowskim. AlphaStar wykorzystał wiele zaawansowanych strategii, które zostały nauczone przez uczenie maszynowe, takie jak "micro", czyli precyzyjne sterowanie jednostkami podczas walki, a także umiejętność przewidywania ruchów przeciwnika.

Sukces AlphaStar w grze Starcraft 2 ma potencjalne zastosowanie w innych dziedzinach, takich jak rozwój systemów autonomicznych i sztucznej inteligencji w przemyśle czy usprawnienie systemów wykorzystywanych w medycynie.

2 Implementacja bota wykorzystującego uczenie ze wzmocnieniem

2.1 Założenia

Z uwagi na złożoność gry projekt został mocno urposzczony. Możliwości decyzyjne modelu zostały ograniczone do 6 następujących akcji:

1. ekspansja: budowa nowej bazy lub produkcja większej ilości robotników
2. eksploracja: wysłanie pracownika na zwiad do losowego miejsca na mapie przewidzianego na ekspancje. Ma na celu znalezienie kolejnych baz przeciwnika.
3. budowa: budowa budynków niezbędnych do trenowania jednostek
4. trening: trening większej ilości jednostek. Dla uproszczenia model ma do dyspozycji tylko 2 jednostki naziemne.
5. atak: wysłanie jednostek bojowych do ataku na jedno z wymienionych: jednostki przeciwnika w pobliżu, budynki przeciwnika w pobliżu, widoczne jednostki przeciwnika w dowolnym miejscu na mapie, widoczne budynki przeciwnika w dowolnym miejscu na mapie lub pozycję startową przeciwnika
6. grupowanie: zgrupowanie bezczynnych jednostek wojskowych w pobliżu najbardziej wysuniętej bazy.

Dodatkowym ułatwieniem jest w pełni oskryptowany początek każdej gry. Pierwsze 2:30 min każdej rozgrywki przebiegają w ten sam sposób, który obejmuje budowę większej liczby robotników oraz najbardziej podstawowych budynków. Ma to na celu zapewnienie dobrego startu do każdej gry.

2.2 Mechanizm nagrody

Opracowany mechanizm nagrody jest bardzo prosty. Model może zdobyć nagrodę na 2 sposoby:

– Każdy atak przeprowadzony przez jednostki miltarne gwarantuje modelowi małą ilość nagrody.
– Zwycięstwo w rozgrywce zapewnia bardzo dużą ilość nagrody, porażka zaś ich utratę.

2.3 Dobór algorytmu

Do nauki modelu został wybrany algytm Proximal Policy Optimization (PPO), z uwagi na jego popularność, stabilność procesu uczenia oraz dobre wyniki.

PPO jest jednym z najbardziej popularnych algorytmów stosowanych w grach wideo, w tym w grach takich jak Starcraft 2, a także w innych dziedzinach, takich jak robotyka, systemy autonomiczne czy medycyna.

Algorytm PPO wykorzystuje ideę prostego gradientu z prostym procesem aktualizacji, co pozwala na skuteczne uczenie modeli policy gradient bez ryzyka destabilizacji lub oscylacji. W algorytmie PPO wykorzystuje się tzw. proximal clipping, czyli obcięcie gradientów funkcji straty, co pozwala na kontrolowanie wielkości zmian w modelu w trakcie aktualizacji.

Do nauki modelu została użyta biblioteka StableBaselines3, która zawiera w sobie między innymi gotową implementacje algorytmu PPO.

2.4 Trening modelu

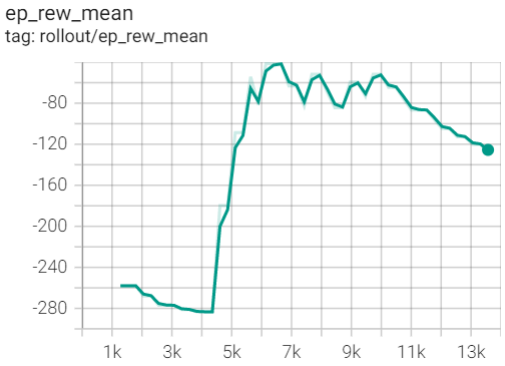
Trening modelu odbywał się na podstawie rozgrywek 1 vs 1. Trenowany model w każdej grze mierzył się z AI zaimplementowanym w samej grze przez jej twórców. Wybór między dostępnymi rasami dla każdej rozgrywki był taki sam: model grał Protossami, a przeciwnik Zergami. Poziom trudności przeciwnika ustawiony był na "Trudny".

Podczas uczenia model rozegrał 69 gier. Agent wykonywał pewne działania w środowisku (np. wytrenowanie nowej jednostki) i obserwował, jak zmienia się stan środowiska. Jedna taka wymiana akcja-obserwacja jest określana jako timestep. Trening zawarł się w prawie 14000 takich wymian. Stan gry przekazywany jest modelowi przez urposzczony obraz minimapy zawierającej infrmację o wszystkich budynkach i jednostakch obydwu graczy, jak również o ilości minerałów i gazu w pobliżu baz gracza.

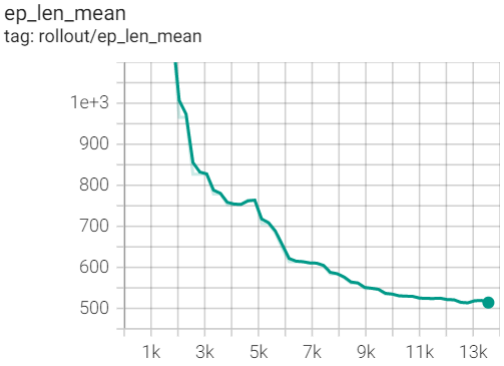
2.5 Test wytrenowanego modelu

W ramach testu wytrenowanego modelu przeprowadzono 10 potyczek przeciwko rasie Terran. Poziom przeciwnika w każdym wypadku ustawiony był na "Trudny". Współczynnik zwycięstw ukształtował sie następująco:

Możliwości modelu z założenia zostały ograniczone do produkcji 2 jednostek bojowych w celu uproszczenia treningu modelu. Porażki podczas testów zazwyczaj wynikały z tego, że przeciwnik masowo szkolił jednostki latające lub niewidzialne, z którymi model nie był w stanie poradzić sobie przy pomocy swoich podstawowych jednostek.



Rysunek 1. Średnia wysokość nagrody (oś Y) od ilości timestep-ów (oś X)

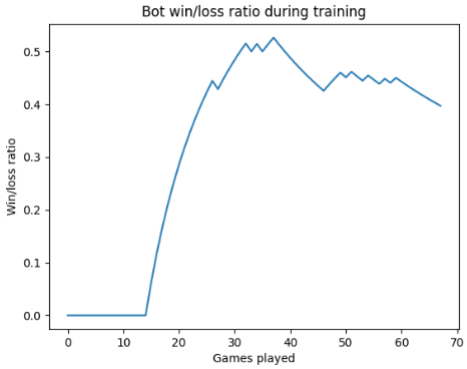


Rysunek 2. Średnia długość rozgrywki [s] (oś Y) od ilości timestep-ów (oś X)

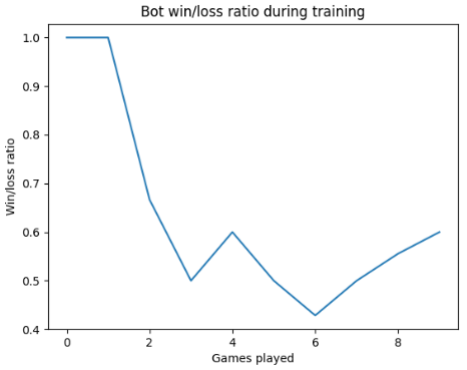
2.6 Wnioski

Model na samym początku trningu nie był w stanie wygrać żadnej gry, jednak po jego przeprowadzeniu bilans zwycięstw do porażek przeciwko "Trudnemu" SI wynosił około 60%. Wykres przedstawiający ilość zdobywanej nagrody podczas uczenia pokazuje stabilny wzrost na początku oraz uplasowanie się w mniej więcej stałej wartości w dalszej części treningu.

Oba fakty dowodzą poprawności implementacji środowiska gry oraz prawidłowego procesu uczenia modelu. W celu poprawy osiągów modelu należałoby rozwinąć zakres podejmowanych przez niego decyzji np. szkolenie większej ilości typów jednostek, co pozwoliłoby lepiej dostosować się do sytuacji w czasie rozgrywki. W takim wypadku konieczny byłby znacznie dłuższy trening modelu.



Rysunek 3. Bilans zwycięstw i przegranych na przestrzeni kolejnych rozgrywek



Rysunek 4. Bilans zwycięstw i przegranych podczas testów (Zerg, poziom trudny

3 Podsumowanie

Dokument ten opisuje implementację uczenia ze wzmocnieniem w grze StarCraft 2. Poprzez ograniczenie problemu do przestrzeni 6 unikalnych czynności oraz opracowanie odpowiedniego mechanizmu nagrody wytrenowy model jest w stanie skutecznie rywalizować z SI na poziomie 'Hard' zaimplementowanym w samej grze przez jej twórców, osiągając przy tym współczynnik zwycięstw około 60%.