Dawid Woźniak
Informatyka - Uczenie Maszynowe I Sztuczna Inteligencja

Uczenie ze Wzmocnieniem

N-krokowa SARSA

1. Opis ćwiczenia.

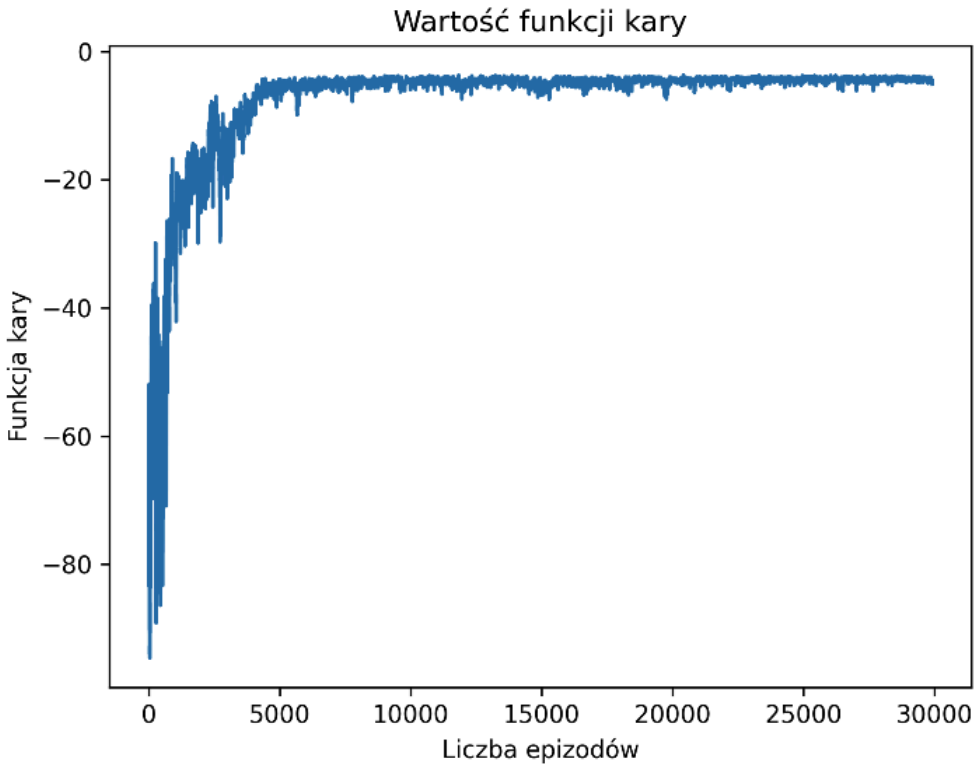
Celem laboratorium było zaimplementowanie i przetestowanie na prostym problemie n-krokowego sterowania SARSA poza polityką. Implementacja została przetestowana w symulacji przejazdu samochodu przez zakręt.

2. Wyniki.

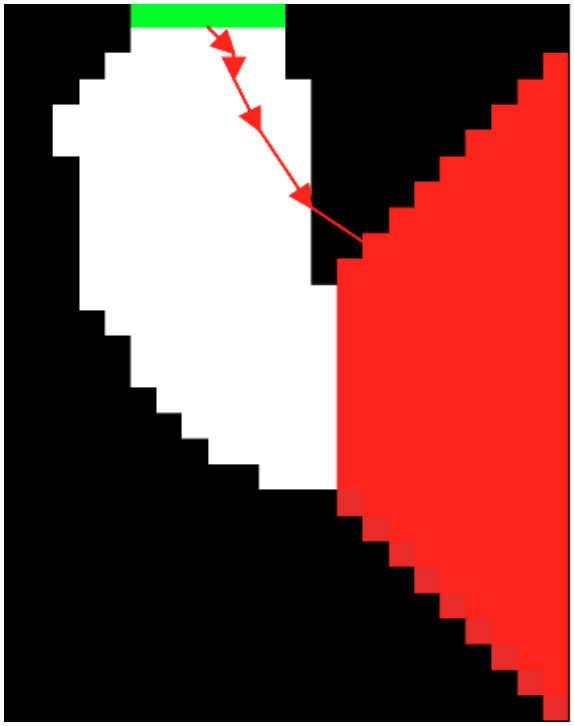
Zadano 3 typy zakrętów do pracy, b (łatwy), c (średni) oraz d (trudny). Początkowo, w ramach weryfikacji implementacji rozwiązanie zostało przetestowane na wariancie b i zwizualizowano przykładowe trasy dla optymalnej polityki zachłannej w tym rozwiązaniu. Następnie, dla przykładu c zostało przeprowadzone studium parametryczne w celu oszacowania optymalnych parametrów dla algorytmu. Na koniec, optymalne parametry wybrane w studium parametrycznym zostały użyte to przetestowania rozwiązania w wariancie d.

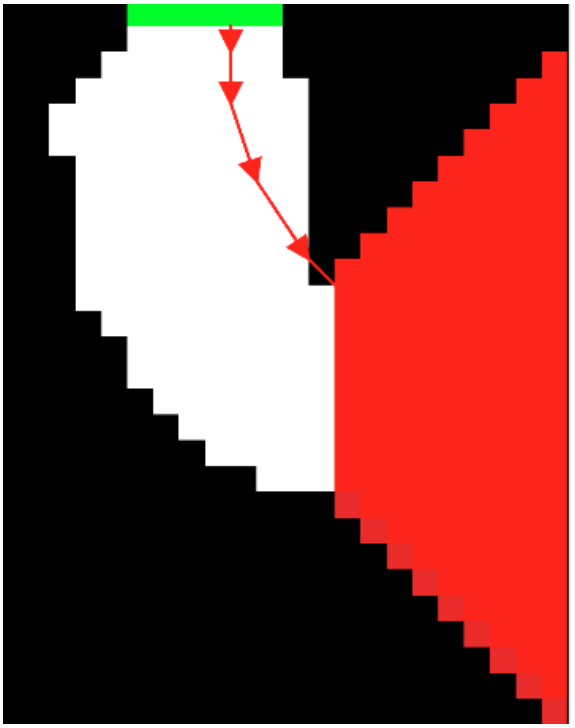
2.1 Weryfikacja prostego przejazdu testowego

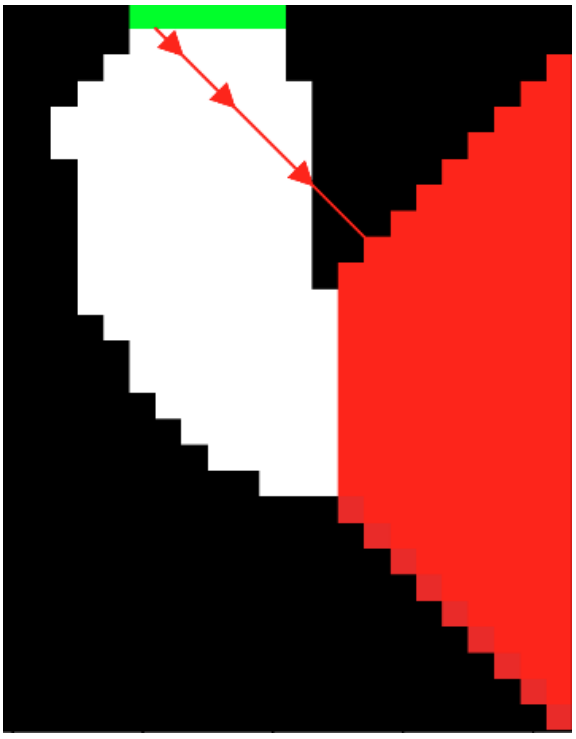
Przejazd dla wariantu b, został przetestowany dla 30 tysięcy epizodów. Przebieg funkcji kary wygląda następująco:



Oto kilka przejazdów dla zachłannej polityki optymalnej:



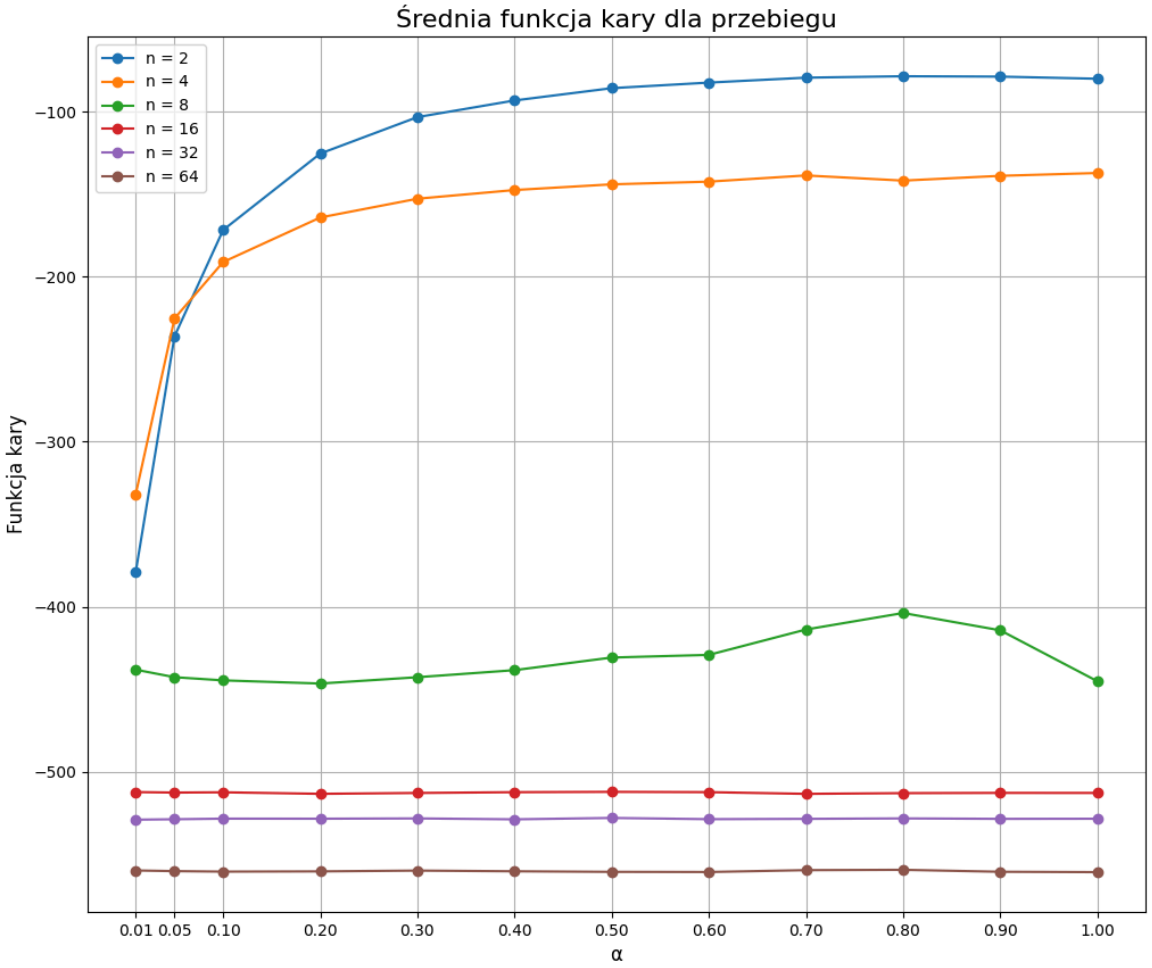




Jak widzimy, zgodnie z obserwacjami wartości funkcji kary w trakcie uczenia, algorytm z sukcesem nauczył się przejazdu przez zakręt b.

2.2 Studium parametryczne

Dla wariantu c zostało przeprowadzone studium parametryczne dla następujących parametrów:
$\alpha \in \{0.01, 0.05, 0.1, 0{,}2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$
$n \in \{2,\ 4,\ 8,\ 16,\ 32,\ 64\}$
Wartość n ograniczyłem do 64, gdyż obliczenia zajmowały dużo czasu i nie spodziewałem się poprawy dla większych wartości. Co będzie widoczne na wykresie, można było ograniczyć tą wartość nawet bardziej. Parametr n mówi nam ile potencjalnych kroków w przód staramy się rozważyć, do zaktualizowania naszej funkcji wartościującej stan. Obliczenia przeprowadziłem dla 10 tysięcy epizodów

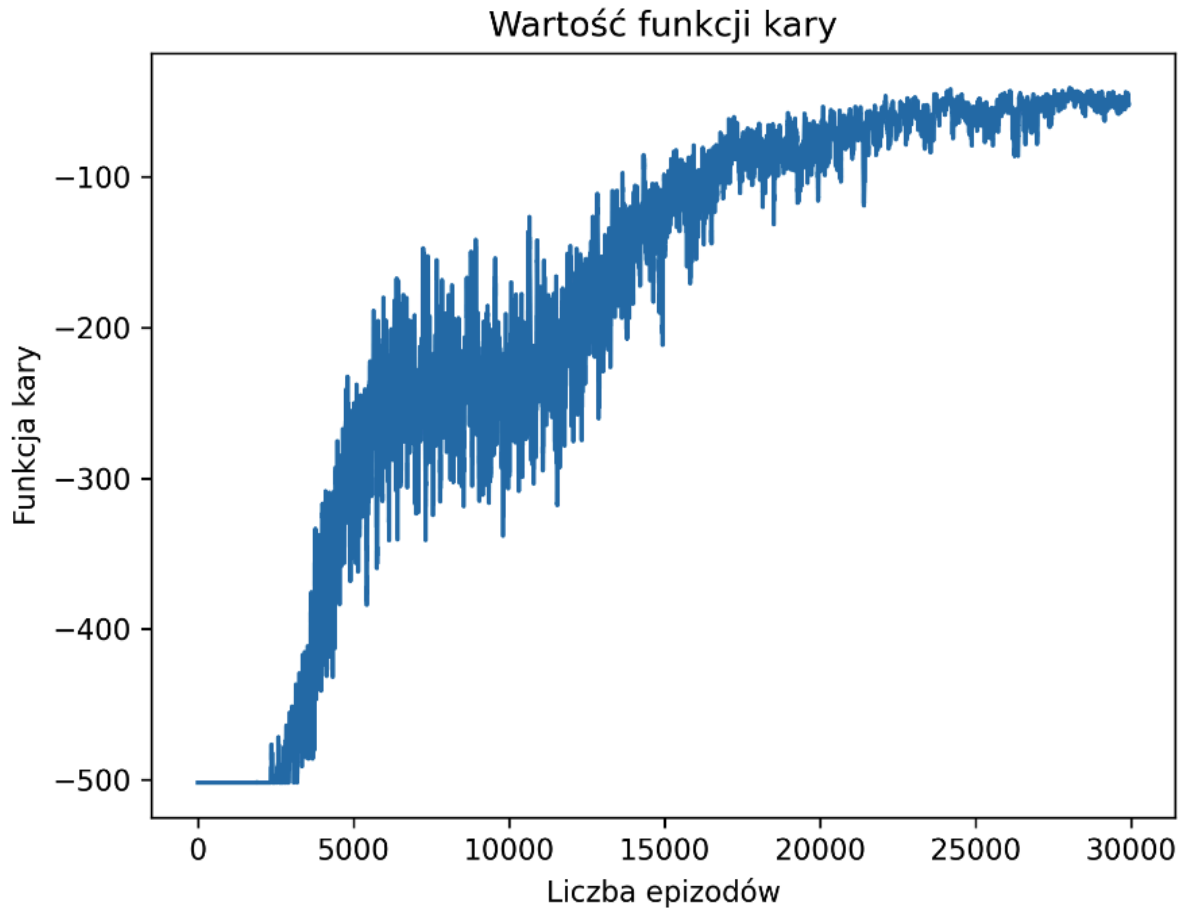


Jak możemy odczytać z wykresu, dla konfiguracji z $n >= 16$, algorytm w ogóle nie uczy się przejazdu przez zakręt. Podobnie, chodź z trochę lepszymi wartościami jest dla $n = 8$. Dzieje się tak, gdyż algorytm stara się wtedy zbytnio aktualizować funkcję wartościującą stan swoim przewidywaniem, nie mając jeszcze dostatecznych informacji. Być może, dla

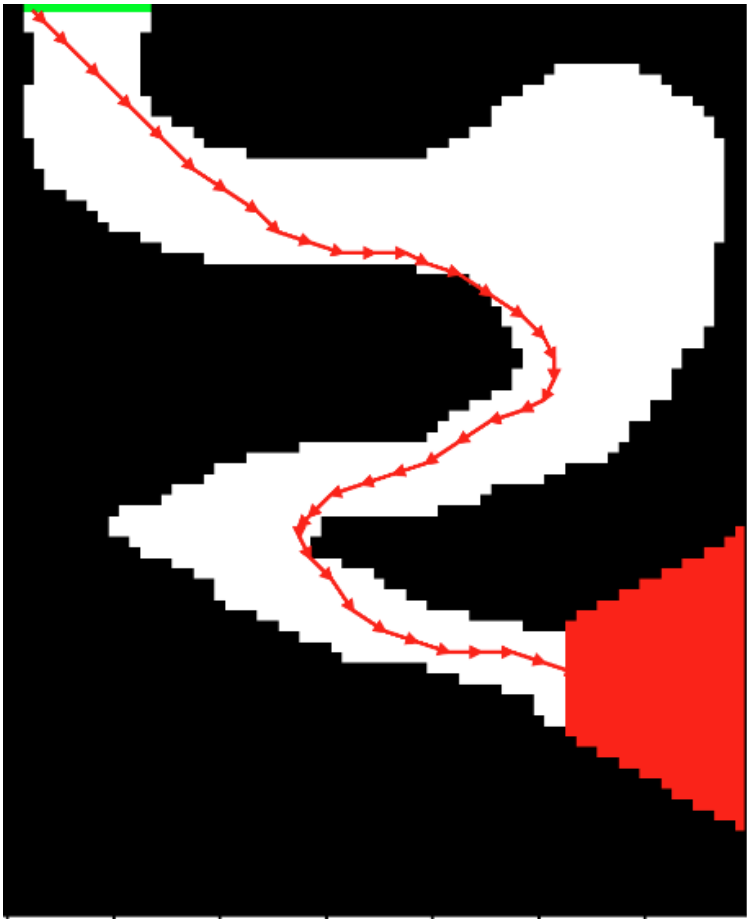
większej ilości epizodów, algorytm zacząłby się uczyć. Najlepszy wynik otrzymałem dla $n = 2$ oraz $\alpha = 0.8$ gdzie średnia wartość funkcji kary wyniosła -78.
Ciekawą opcją dla mnie wydałoby się stosowanie różnej konfiguracji parametru n np. zaczęcie dla małego n, a wraz z momentem gdy algorytm zaczyna się uczyć, przełączyć się na większą wartość lecz nie przetestowałem takiej konfiguracji.

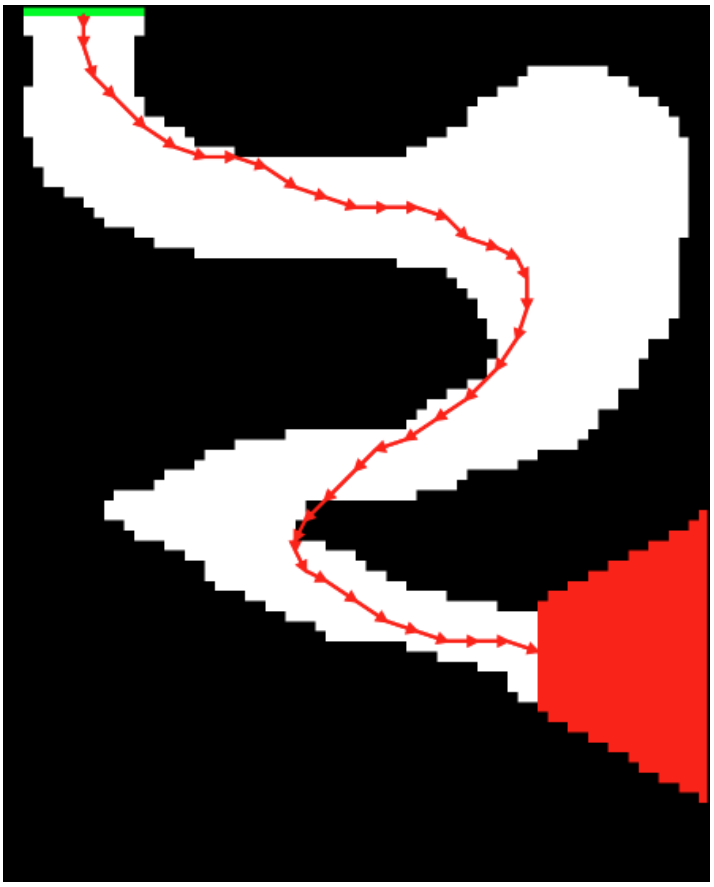
2.3 Test dla optymalnych parametrów

Dla otrzymanych optymalnych parametrów $n = 2$ oraz $\alpha = 0.8$ przetestowałem najtrudniejszy wariant d dla 30 tysięcy epizodów. Oto przebieg funkcji kary:



Jak widzimy na wykresie funkcja kary zaczyna się stabilizować około 20 tysięcy epizodów i wtedy nasz algorytm już powinien bezbłędnie przejeżdżać zakręt dla polityki optymalnej, dla uczącej polityki epsilon-zachłannej mogą być odchylenia ze względu na losowość. Warty obserwacji jest jeszcze początek przebiegu - do około 3 tysięcy epizodów mamy stałą karę równą około 500. Oznacza to, że dopiero od około tego przebiegu nasz algorytm zaczyna przejeżdżać przez cały zakręt. Podobnie jak w podpunkcie 2.1 zwizualizowałem również kilka przykładowych tras dla optymalnej polityki zachłannej.





Jak można zaobserwować algorytm nauczył się przejazdu przez najtrudniejszy przykład. Można zauważyć, że algorytm nauczył się jechać jak najbliżej wewnętrznej krawędzi zakrętu.

3. Wnioski

Udało się poprawnie zaimplementować algorytm n-krokowego sterowania SARSA poza polityką. Wariant poza polityką wydaje się ciekawym podejściem. Ważnym i podchwytliwym aspektem jest obliczanie parametru ρ, który uwzględnia to, że za pomocą polityki epsilon-zachłannej chcemy nauczyć naszą optymalną politykę zachłanną. Stworzenie studium parametrycznego, kolejny raz pozwoliło mi się przekonać jak ważny w algorytmie jest wybór parametrów - dobierając je w zły sposób można otrzymać nieużyteczne wyniki.