# WSI Ćwiczenie 6

Konrad Karpiuk

## 1. Opis zadania

Zaimplementować algorytm Q learning i użyć go do wyznaczenia polityki decyzyjnej dla problemu FrozenLake8x8.

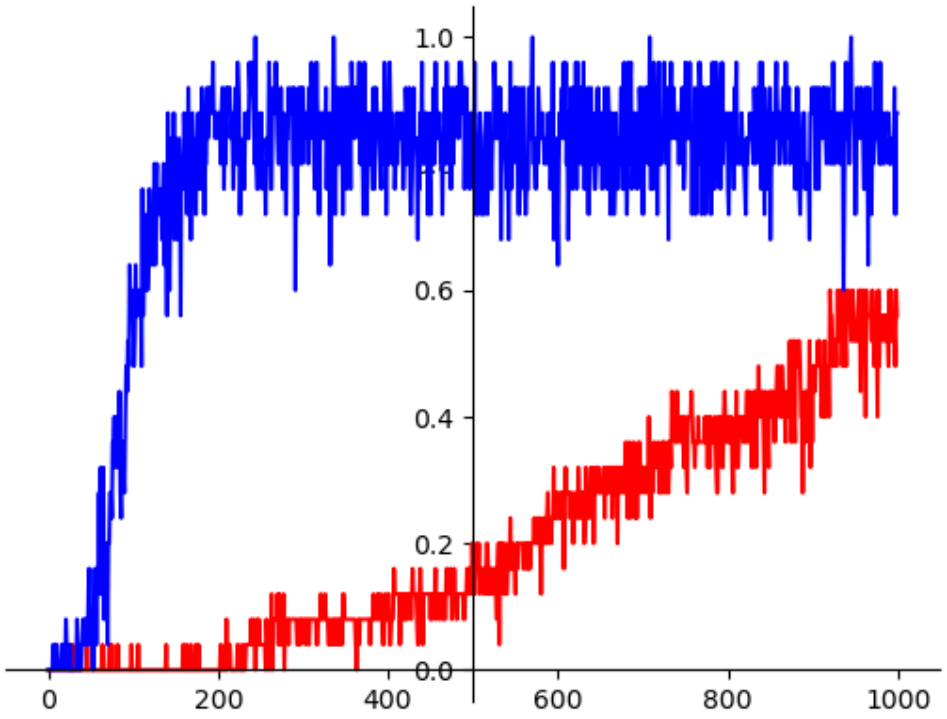
W każdym teście parametry algorytmu Q wynosiły: $\beta$ = 0,9, $\gamma$ = 0,8. Akcje wybierano metodą $\varepsilon$-zachłanną z wartością parametru $\varepsilon$ = 0,1.

## 2. Wersja bez poślizgu

Na początku rozpatrywany jest problem z wyłączonym poślizgiem, oznacza to, że agent zawsze wykona zleconą mu akcję. Z takimi założeniami od razu nasuwa się na myśl strategia polegająca na unikaniu dziur, gdyż powodują one zakończenie epizodu z brakiem nagrody oraz unikaniu wchodzenia w ścianę, gdyż jest to pusty ruch, który nie przybliża agenta do pola z nagrodą. Aby zastosować tę strategię przyjąłem następującą punktację:

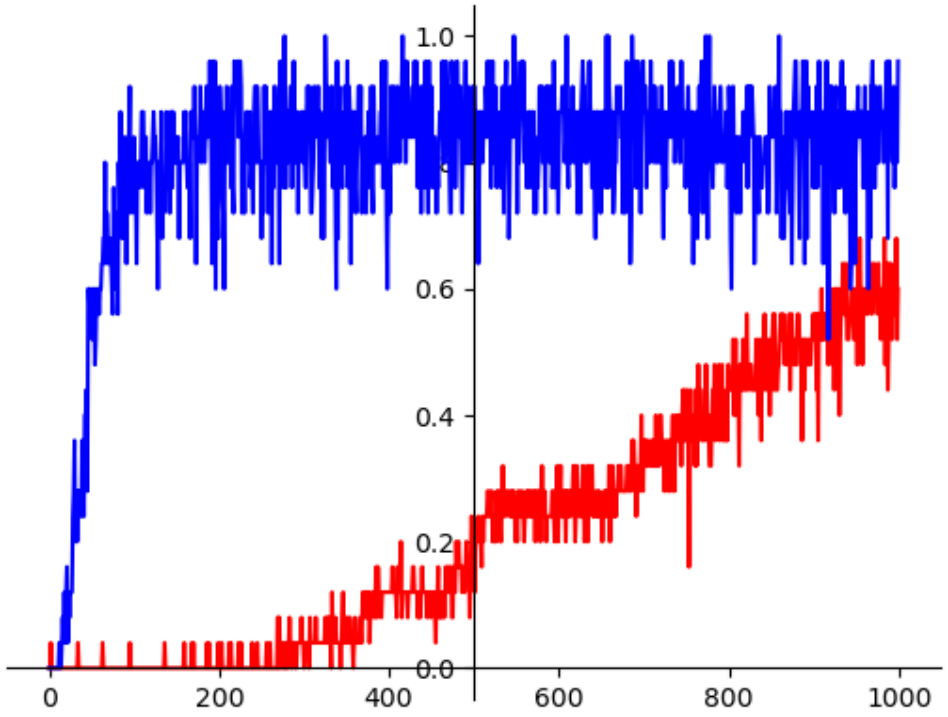
- Dotarcie do pola z prezentem: 1 pkt,
- Wpadnięcie do dziury: -1 pkt,
- Wejście w ścianę: -1 pkt.

Stosując tę strategię otrzymano następujące wyniki:



Kolorem czerwonym oznaczono wyniki z prób wykorzystujących strategię domyślną (1 pkt za dojście do prezentu i brak kar), a niebieskim kolorem oznaczono moją strategię. Jak można zauważyć strategia wykorzystująca kary okazała się dużo lepszym rozwiązaniem w porównaniu do domyślnej strategii i już od około dwusetnego epizodu pozwalała na osiąganie pola z prezentem prawie za każdym razem. Widoczne wahania wokół wartości 0,9 na wykresie wynikają z wielkości parametru $\varepsilon$ (w każdym ruchu agent ma 10% szans na wykonanie losowego ruchu, w tym wpadnięcie w dziurę).

W trakcie obserwowania symulacji zauważyłem, że głównym problemem powodującym osiąganie dobrych wyników dopiero w okolicy epizodu nr 200 jest błądzenie agenta po dużych obszarach w lewym oraz w prawym górnym rogu mapy. Aby zachęcić go do szybszego opuszczania tych początkowych terenów zdecydowałem w drugiej strategii dodać dodatkową karę: jeśli agent wykona ruch na pole, z którego dopiero co przyszedł (czyli zawróci), to otrzyma karę -0,1 pkt. Kara ta powoduje, że po początkowym błądzeniu w obszarach pozbawionych dziur, pola w tych obszarach posiadają tablice wypełnione wartościami ujemnymi, więc gdy tylko nadarzy się okazja na opuszczenie obszaru, agent zdecyduje się na to. Podejście to premiuje przemieszczanie się agenta w głąb mapy, co zwiększa szansę na znalezienie prezentu. Zastosowanie drugiej strategii przyniosło następujące wyniki:

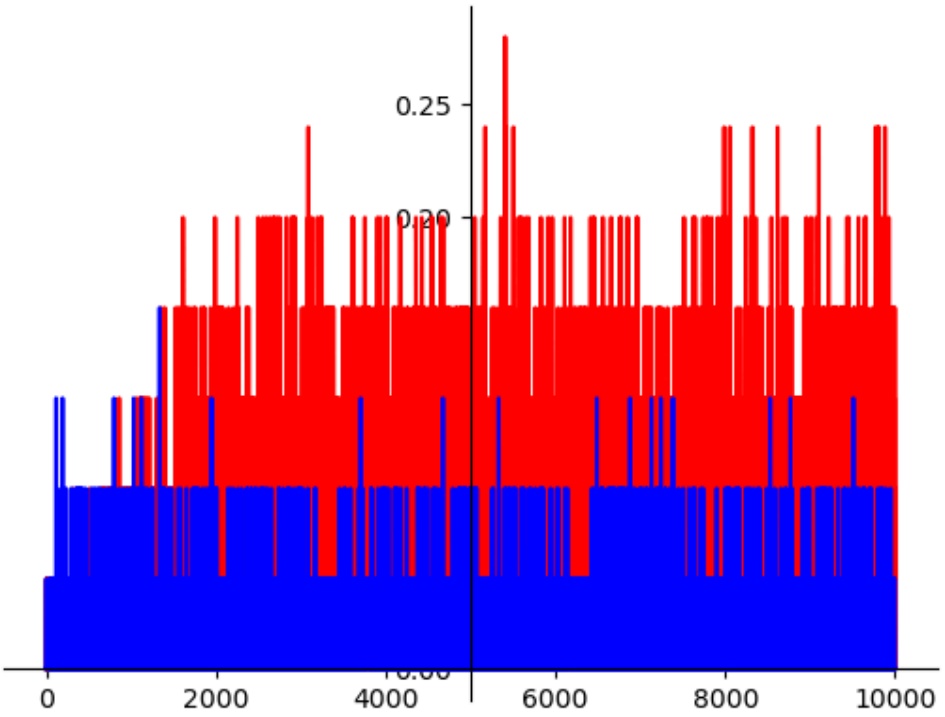



Kolorem niebieskim oznaczono wyniki uzyskane dzięki strategii z karą za zawracanie, a kolorem czerwonym przedstawione są wyniki uzyskane strategią domyślną. Zgodnie z oczekiwaniami otrzymano jeszcze wcześniejsze osiągnięcie poziomu ok. 90% trafności do celu. Niebieski wykres już po kilku epizodach podnosi się ponad

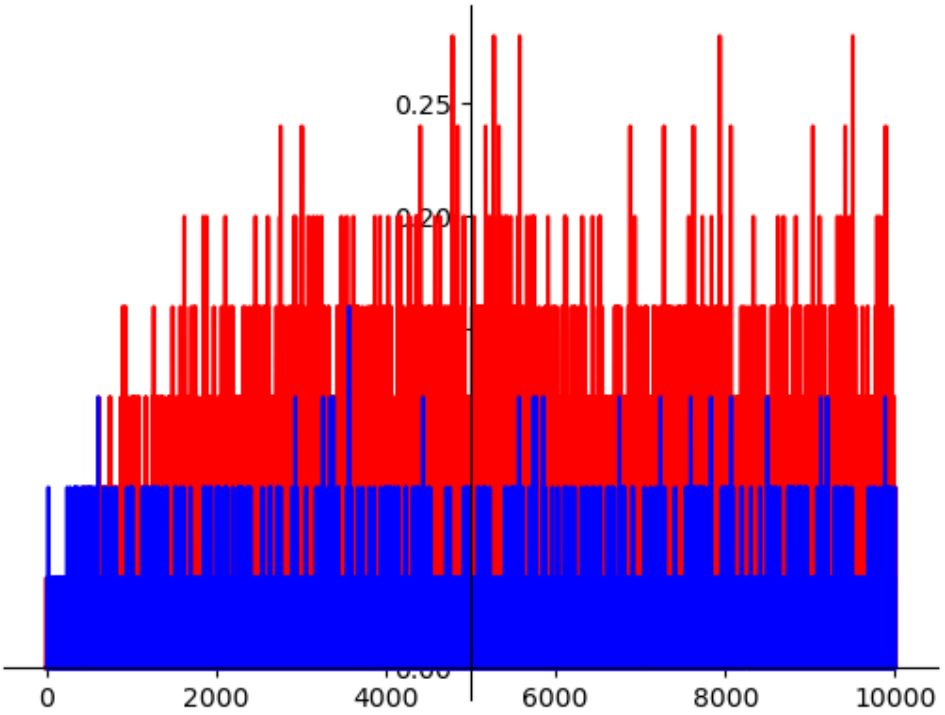
poziom 0,0 i bardzo szybko rośnie. Druga strategia dla przyjętych danych okazała się być najlepsza.

## 3. Wersja z poślizgiem

Drugą częścią ćwiczenia jest wykorzystanie poprzednich strategii w środowisku zawierającym poślizg - agent ma szansę ⅓ , że pójdzie we wskazanym kierunku oraz po ⅓ szans, że pójdzie skręci o 90 stopni. Ze względu na większą losowość zwiększono liczbę epizodów do 10000. Uzyskano następujące wyniki:




Tym razem strategia domyślna (kolor czerwony) okazała się być bardziej efektywna od strategii dającej karę za wpadnięcie w dziurę i wejście w ścianę(kolor niebieski). Spójrzmy jeszcze na wyniki uzyskane przez wykorzystanie drugiej strategii:




Wykres ten wygląda bardzo podobnie do poprzedniego. Według mnie wyniki te są bardzo sensowne. Moim zdaniem najlepsza taktyka, jaką można obrać w przypadku poślizgu, to dotarcie do prawej krawędzi planszy, a potem cały czas wchodzenie w ścianę, gdyż albo straci się turę w przypadku powodzenia, albo pójdzie w górę lub w dół, co jest bezpiecznym ruchem. Po odpowiednio dużej liczbie iteracji metoda ta doprowadzi w końcu do prezentu. Jedyna możliwość porażki to wyczerpanie limitu czasu (200 iteracji) przed osiągnięciem prezentu. Natomiast w przypadku stosowania obu z zaproponowanych przeze mnie taktyk po dojściu do prawej krawędzi algorytm wybierze albo ruch w lewo, albo ruch w prawo, gdyż daje on ⅔ szans na brak kary. A więc program będzie często wykonywał ruch, który umożliwia wpadnięcie do dziury i zresetowanie gry. Z tego powodu nie dziwi mnie, że algorytm domyślny radzi sobie lepiej, gdyż jest on w stanie zastosować bezpieczną strategię wchodzenia w ścianę. Przypadek ten udowadnia, że nie istnieje jedna optymalna strategia do każdego problemu, a każde zadanie wymaga indywidualnego podejścia zależnego od przyjętych zasad.