WSI Uczenie ze wzmocnieniem sprawozdanie

Małgorzata Grzanka

26.05.2024

Spis treści

1 Wstep

Celem ćwiczenia było zaimplementowanie algorytmu Q-learning. Nastepnie, algorytm należało przetestować, rozwiazujac za jego pomoca problem Taxi, dostepny w pakiecie gymnasium (https:// gymnasium.farama.org/environments/toy_text/taxi/).

2 Zasada działania Q-learningu

2.1 Ogólny koncept

Q-learning polega na uczeniu agenta najoptymalniejszych zachowań w jakimś środowisku.

Najważniejszym elementem w procesie uczenia jest Q-function. Q-function w jakimś stanie s_t po akcji a_t to suma nagród, które gracz osiagnie, robiac akcje a_t i dalej wykonujac ruchy zgodnie z polityka π. Zatem agent powinien w wiekszości przypadków wybierać akcje, dla której, przy danym stanie środkowiska, Q-function zwróci najwieksza wartość.

$$Q^{\pi}(s_t, a_t) = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 r_{t+3} + \dots$$

W Q-learningu, jako że wartości Q-function sa na poczatku inicjowane losowymi wartościami, możemy obliczać wartości Q-function za pomoca jej wartości dla kolejnego stanu (wybieramy najwieksza wartość Q-function dla kolejnego ruchu):

$$Q^{\pi}(s_t, a_t) = r_t + \gamma \cdot max_{a_{t+1}}(Q^{\pi}(s_{t+1}, a_i))$$

Dla mniej skomplikowanego środowiska, jakim jest Taxi problem, wszystkie wartości, które może przyjać Q-function, można zapisać w postaci tabeli o wymiarach $|S|$ x $|A|$, gdzie $|S|$ to liczba wszystkich mozliwych stanów w grze, a $|A|$ to liczba możliwych akcji, które można w tej grze podjać.

2.2 Uczenie sie

Proces uczenia polega na wyznaczeniu przez algorytm odpowiednich wartości Q-function poprzez wielokrotne ruchy agenta w środowisku i otrzymywanie od tego środowiska sprzeżenia zwrotnego w postaci nagrody. Algorytm daży do tego, aby różnica pomiedzy wartościa Q-function otrzymanej z tabeli była najbliższa możliwa wartości wyliczonej za pomoca wzoru na Q-function.

Stosuje sie gradient descend do minimalizacji tego błedu:

$$Q^{\pi}(s_t, a_t) = Q^{\pi}(s_t, a_t) + \alpha \cdot (r_t + \gamma \cdot max_{a_{t+1}}(Q^{\pi}(s_{t+1}, a_i)) - Q^{\pi}(s_t, a_t))$$

2.3 Wybór akcji

Wybór jakiejś acji przez agenta odbywa sie z pradopodobieństwami wyliczonymi na podstawie strategii boltzmannowskiej:

$$P(a|s) = \frac{e^{\frac{Q(s,a)}{T}}}{\sum_{a'} e^{\frac{Q(s,a')}{T}}}$$

Parametr T (temperatura) kontroluje balans miedzy eksploracja a eksploatacja w strategii Boltzmannowskiej. Wysoka temperatura sprzyja eksploracji (losowemu wybieraniu akcji), podczas gdy niska temperatura sprzyja eksploatacji (wybieraniu najlepszej znanej akcji).

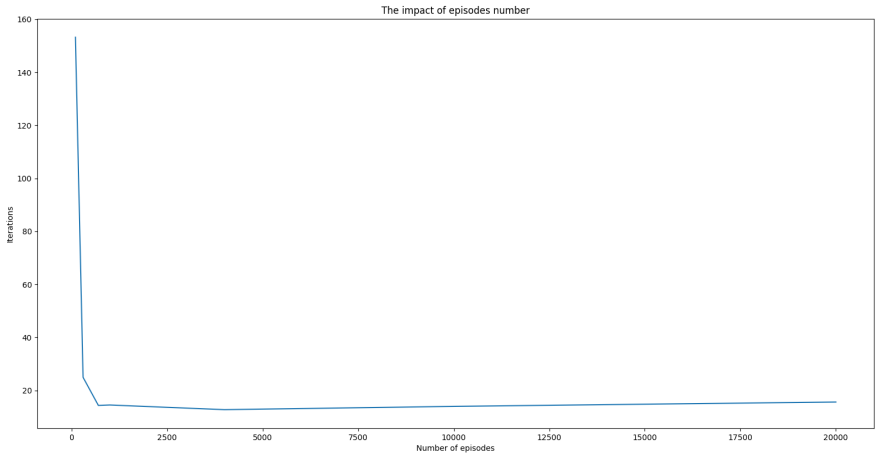
3 Hiperparametry

Mój model algorytmu Q-learningu przyjmuje nastepujace hiperparaetry:

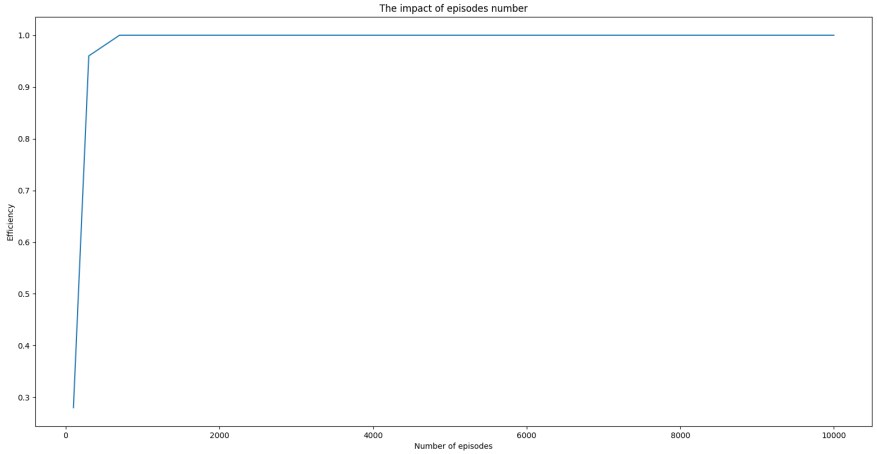
- dyskonto - jak bardzo przyszłe nagrody sa brane pod uwage przy podejmowaniu bieżacych decyzji.
- T - temperatura, kontroluje balans miedzy eksploracja a eksploatacja w strategii Boltzmannowskiej.
- epizody - ile symulacji treningowych algorytm powinien rozegrać, aby sie nauczyć.
- learning rate - współczynnik dla minimalizacji bledu Q-function (dla algorytmu gradient desced).

4 Wpływ liczby epizodów

4.1 Wyniki



Rysunek 1: Wpływ liczby epizodów na liczbe iteracji podczas symulacji



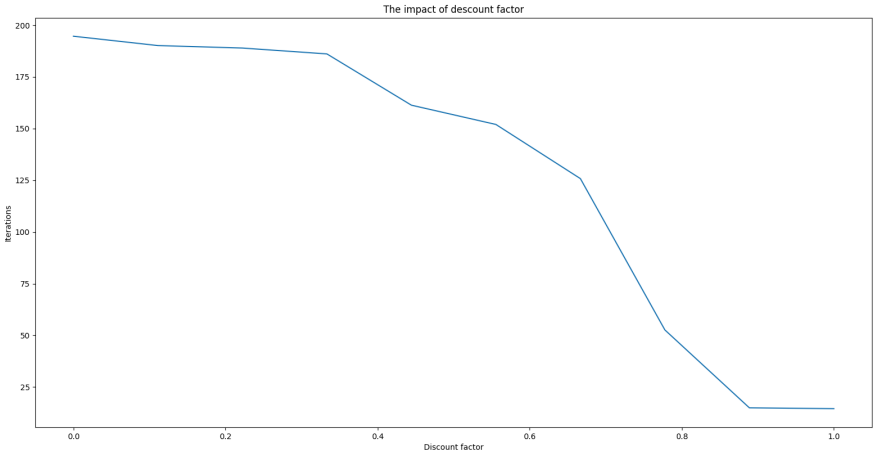
Rysunek 2: Wpływ liczby epizodów na skuteczność agenta podczas symulacji

4.2 Wnioski

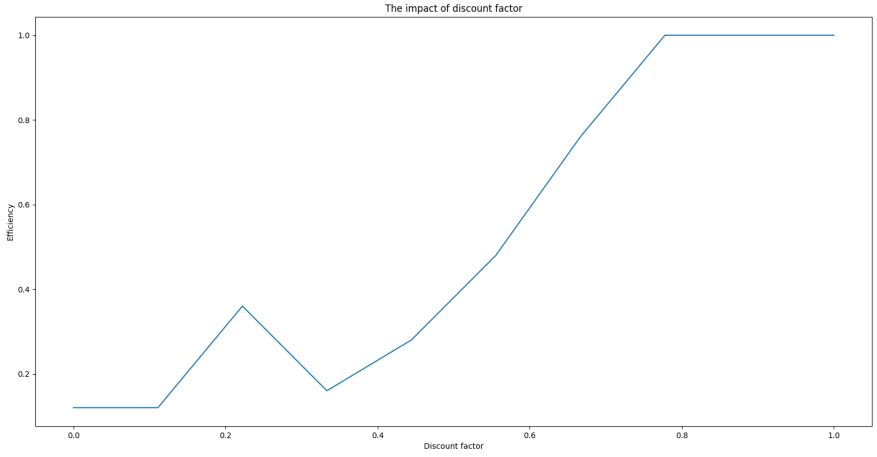
Wraz z wrostem epizodów, wartości przyjmowane przez Q-function sa bardziej dopasowane dla danego środowiska (wiecej prób podczas treningu agenta). Dlatego, dla małej ilości epizodów, wytrenowanemu agentowi ukończenie symulacji zajmuje bardzo dużo czasu (ciagle wykonuje niepoprawne ruchy), a jego skuteczność jest bardzo słaba. Liczba iteracji potrzebna do zakończenia symulacji przez wytrenowanego agenta spada gwałtownie już przy około 400 próbach, ale swoje absolutne minimum (około 12 iteracji) osiaga dla liczby epizodów równej około 4000. Jeśli chodzi o skuteczność, jest ona równa 100% dla liczby epizodów wiekszej niż 1000.

5 Wplyw wartości dyskontu

5.1 Wyniki



Rysunek 3: Wpływ wartości dyskontu na liczbe iteracji podczas symulacji



Rysunek 4: Wpływ wartości dyskontu na skuteczność agenta podczas symulacji

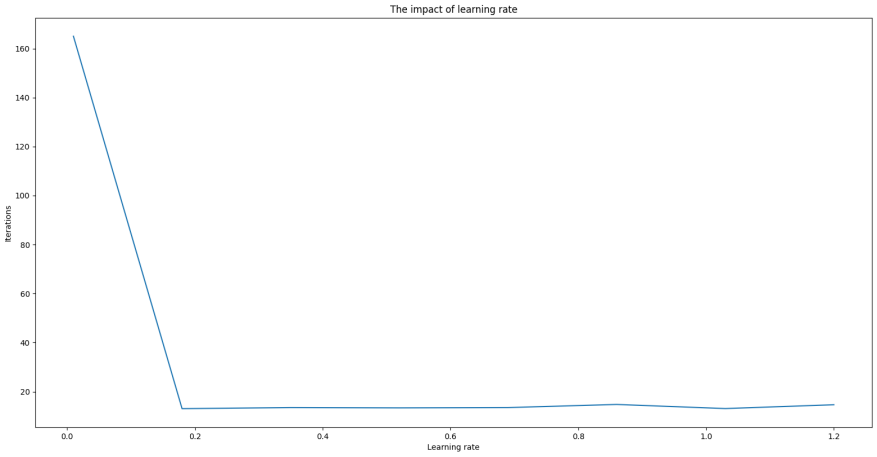
5.2 Wnioski

Współczynnik dyskontowy określa, jak bardzo przyszłe nagrody sa uwzgledniane w porównaniu do bieżacych nagród i przyjmuje wartości od 0 do 1.

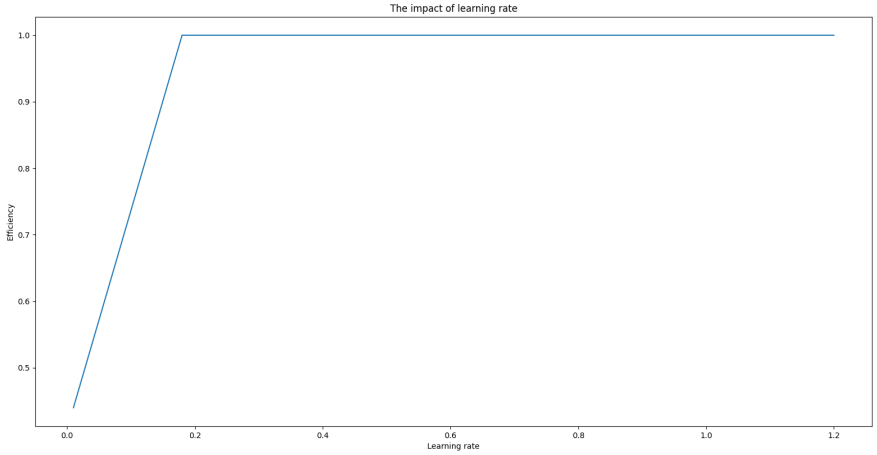
Dla rozważanego problemu, lepsza skuteczność i szybsze dotarcia do końca symulacji osiagane sa dla wyższych wartościu dyskontu. Małe wartości dyskontu sprzyjaja dużym, natychmiastowy nagrodom, a wieksze prowadza do bardziej optymalnych strategii w dłuższym horyzoncie czasowym. Jednak może to również sprawić, że agent bedzie bardziej wrażliwy na zmienność i stochastyczność środowiska, co może utrudniać szybkie zdobycie nagród. Dla tej symulacji, lepsze okazały sie te wieksze wartości współczynnika dyskontu.

6 Wpływ wartości learning rate

6.1 Wyniki



Rysunek 5: Wpływ wartości learning rate na liczbe iteracji podczas symulacji



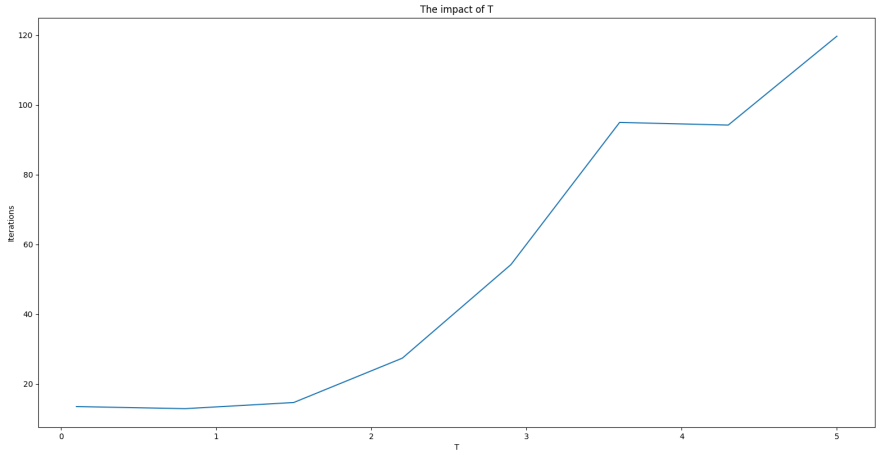
Rysunek 6: Wpływ wartości learning rate na skuteczność agenta podczas symulacji

6.2 Wnioski

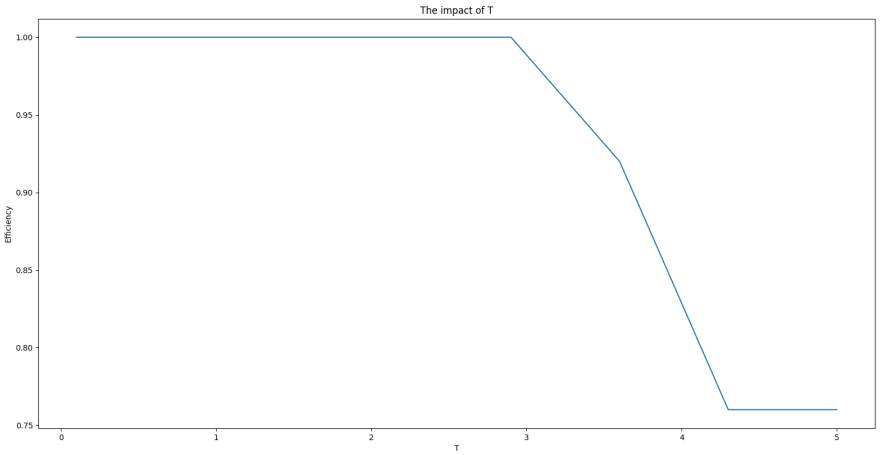
Im wieksze learning rate, tym szybciej agent uczy sie właściwych ruchów na symulacji. Dla bardzo małych learning rate, liczba epizodów równa 3000 nie była wystarczajaca dla właściwego nauczenia agenta - liczba iteracji była bardzo duża, a skuteczność bardzo mała. Jednakże, dla bardzo duzych learning rate, wartości Q-function staja sie za duże dla poprawnego policzenia prawdopodobieństwa strategia Boltzmana (overflow w podnoszeniu e to potegi). Idealna wartościa dla tego problemu sa wartości z przedziału 0.4 - 1.

7 Wpływ wartości T

7.1 Wyniki



Rysunek 7: Wpływ wartości T na liczbe iteracji podczas symulacji



Rysunek 8: Wpływ wartości T na skuteczność agenta podczas symulacji

7.2 Wnioski

Im wieksze T, tym wyliczane prawdopodobieństwa mniej zależa od wartości Q-function dla możliwych akcji. Oznacza to, że prawdopodobieństwa wykonania

każdej akcji w danym stanie sa do siebie zbliżone. Oznacza to, że agent jest nastawiony bardziej na eksploracje nowych rozwiazań. Z kolei dla małych wartości T, agent skupia sie bardziej na eksploatacji znlezionych wcześniej wartości Q-function. W przypadku problemu Taxi, bardziej efektywne sa mniejsze wartości T (z przedziału 0 - 1).