SKRYPT DO LABORATORIUM

SIECI SAMOUCZĄCE SIĘ

LABORATORIUM 2:

Zastosowanie programowania dynamicznego do wyznaczania optymalnej strategii metodą iteracji wartości

Jerzy Dembski

Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Rozwoju Regionalnego
Program Operacyjny Polska Cyfrowa na lata 2014-2020,
Oś Priorytetowa nr 3 "Cyfrowe kompetencje społeczeństwa" Działanie nr 3.2 "Innowacyjne rozwiązania na rzecz aktywizacji cyfrowej"
Tytuł projektu: „Akademia Innowacyjnych Zastosowań Technologii Cyfrowych (AI Tech)”

1. Opis ćwiczenia

Wymagania wstępne:

Zapoznanie się z treścią wykładu.

Cele ćwiczenia:

Zapoznanie się z iteracyjną wersją metody programowania dynamicznego.

Spodziewane efekty kształcenia - umiejętności i kompetencje:

Umiejętność posługiwania się metodą programowania dynamicznego do szukania strategii optymalnej. Rozumienie procesów decyzyjnych Markowa i równań równowagi Bellmana.

Metody dydaktyczne:

Dostarczenie skryptów do zadania ,,Żeglarz" - agenta poruszającego się wśród wodnych przeszkód w Matlabie i Pythonie zawierających wiele przydatnych funkcji do realizacji zadania oraz skryptu xyz.m zawierającego realizację algorytmu iteracji strategii w programowaniu dynamicznym na prostym przykładzie procesu o 3 stanach.

Zasady oceniania/warunek zaliczenia ćwiczenia

Za zadania zrobione w całości na zajęciach przysługuje ocena 5, przy czym zadanie uznaje się za zrobione, gdy średnia suma nagród dla testu (po uruchomieniu skryptu lub funkcji sailor_test) wynosi co najmniej 7,5 dla map map_easy oraz map_middle oraz jest dodatnia dla mapy map_big. Ocena za zadanie dokończone po zajęciach zależy od stopnia jego kompletności i stopnia realizacji podczas zajęć. Za zadanie dokończone w całości po zajęciach przysługuje ocena 3 (pod warunkiem obecności na zajęciach i wiarygodnej prezentacji zadania).

Wykaz literatury podstawowej do ćwiczenia:

Richard Sutton, Andrew G. Barto, Reinforcement Learning: An Introduction, MIT Press, Cambridge, MA, 2018.
Jerzy Dembski, Materiały do przedmiotu Sieci samouczące się

2. Przebieg ćwiczenia

Po krótkim wprowadzeniu do ćwiczenia przez prowadzącego, studenci przystępują do realizacji zadania w wybranym środowisku (Matlab lub Python) z możliwością ciągłych konsultacji osiąganych rezultatów z prowadzącym. Pod koniec zajęć prowadzący ocenia końcowe rezultaty pracy. Zadania wykonane w całości na zajęciach, jak i dokończone poza zajęciami wraz ze sprawozdaniami są umieszczane na stronie kursu na eNauczanie.

3. Wprowadzenie do ćwiczenia

Zadanie: Uzupełnić skrypt sailor_train.m tak, aby pozwalał znaleźć optymalną strategię żeglarza startującego z losowego pola pierwszej kolumny, polegającą na maksymalizacji sumy nagród. Problem jest niedeterministyczny - żeglarz po wyborze kierunku (akcji) tylko z pewnym prawdopodobieństwem trafia do wybranego pola. Z powodu wiatru, prądów morskich i innych przyczyn może popłynąć w bok, a nawet do tyłu w stosunku do wyznaczonego celu. Model środowiska w postaci prawdopodobieństw przejść i nagród znajduje się w pliku environment.m oraz w plikach map (wartości nagród). Do uczenia należy wykorzystać iteracyjną wersję metodę programowania dynamicznego z iteracją wartości lub iteracją strategii. Na zaliczenie ćwiczenia należy wykonać jedną z powyższych metod, ale warto obie sprawdzić i porównać.

Zawartość skryptów w Matlabie:

sailor_train.m - skrypt służący do uczenia (wymagający uzupełnienia),
sailor_test.m - skrypt służący do testowania wyuczonej strategii,
sailor_vizual.m - skrypt służący do wizualizacji ruchu żeglarza,
environment.m - funkcja zwracająca stan oraz nagrodę na podstawie stanu w poprzednim kroku oraz akcji wykonanej w tym stanie,
draw.m - funkcja rysująca sztuczne morze.

Q.mat - plik zawierający tablicę użyteczności par { (stan,akcja)}, który po załadowaniu (load Q) tworzy trójwymiarową tablicę w pamięci roboczej. Pierwsze dwa wymiary odpowiadają współrzędnym planszy, trzeci wymiar określa akcję (1-ruch w prawo,2 - do góry, 3 - w lewo, 4 - do dołu)

W przypadku języka Python nazwy są analogiczne, ale odpowiadają funkcjom w skrypcie sailor_funct.py.

Dodatkowy skrypt xyz.m zawiera algorytm z iteracją strategii dla prostego problemu 3-stanowego. Użyteczności stanów obliczane są poprzez rozwiązanie układu równań liniowych (odwrócenie macierzy współczynników M). Zmodyfikowany skrypt xyz_osobliwy.m pokazuje problem MDP w wersji osobliwej - gdy wyznacznik macierzy M jest zerowy a wartości nagród w trakcie procesu decyzyjnego rosną do nieskończoności.

4. Forma i zawartość sprawozdania

Krótki opis zadania, wybranych metod jego rozwiązania wraz z uzasadnieniem wyboru oraz uzyskanych wyników.

Dodatki