**Комп’ютерний практикум №3**

**Марківські процеси прийняття рішень. Навчання з підкріпленням.**

**ПІБ:** Шляхтун Денис Михайлович.

**Група:** ІП-14.

**Мета роботи:** ознайомитись з методами пошуку в умовах невідомості та навчання з підкріпленням в моделях на основі станів; дослідити їх використання для інтелектуального агента в типовому середовищі.

**Завдання:** обрати середовище моделювання та задачу, що містить агента, який може бути навчений методом «з підкріпленнями». В обраному середовищі вирішити задачу знаходження найкращої стратегії поведінки, реалізувавши один з методів. Виконати дослідження реалізованого методу.

**Номер варіанту:** 22.

**Завдання для варіанту:** лабіринт, вплив коефіцієнта знецінення, Q-learning.

**Середовище:** двомірне середовище розміром 10х10, містить «стіни», що відповідають областям, в які не можна потрапити і які утворюють лабіринт, і стан з «винагородами», при досягненні якого завершується епізод, виконання кожної дії зменшує загальну винагороду на постійну величину – вартість руху (-1). У середовищі можливі рухи вгору, вниз, вліво, вправо. Можливо здійснити рух у сторону стіни чи межі карти, але переміщення здійснено не буде. Єдина винагорода (99) знаходиться всередині лабіринту і є кінцевою ціллю агента. Середовище не є стохастичним, адже вибір напряму руху та нагорода не є випадковими величинами. Початковою точкою є верхній лівий кут, кінцевою є край червоної лінії у нижній частині рисунку.

Зображення, що містить символ, Графіка, логотип, дизайн

Автоматично згенерований опис

**Метод вирішення задачі:** Q-learning – метод навчання з підкріпленням, що використовується для навчання агента оптимальним стратегіям дій. Q-learning має онлайн стратегію, адже агент взаємодіє з середовищем у реальному часі, отримуючи нагороди та оновлюючи свою стратегію під час виконання, також агент не має повне знання про середовище заздалегідь. Для виконання алгоритму потрібно досліджувати середовище. Перед виконанням алгоритму агенту треба надати усі стани та можливі дії, які він може виконувати. При проходженні агенту по середовищу потрібно надавати нагороду за кожну дію агента.

**Реалізація методу:**

class QLearningAgent():

def \_\_init\_\_(self, actions, learning\_rate, discount\_factor, exploration\_prob):

self.height = 10

self.width = 10

self.actions = actions

self.learning\_rate = learning\_rate

self.discount\_factor = discount\_factor

self.exploration\_prob = exploration\_prob

self.Q\_table = np.zeros((self.height, self.width, len(self.actions)))

def choose\_action(self, current\_state):

if np.random.rand() < self.exploration\_prob:

action = self.actions[np.random.randint(0, len(self.actions))]

else:

action = self.actions[np.argmax(self.Q\_table[current\_state])]

return action

def add\_reward(self, current\_state, action, reward, next\_state):

action\_ = self.actions.index(action)

self.Q\_table[current\_state][action\_] += self.learning\_rate \* \

(reward + self.discount\_factor \*

np.max(self.Q\_table[next\_state]) - self.Q\_table[current\_state][action\_])

**Результати застосування розробленого методу:**

learning\_rate = 0.7, discount\_factor = 0.7, exploration\_prob = 0.2

Кожна клітинка містить значення Q-функції для рухів вгору (лівий верхній кут кожної клітинки), вниз (правий верхній), вліво (нижній лівий), вправо (нижній правий).

Зображення, що містить знімок екрана, візерунок

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, Графік, схема

Автоматично згенерований опис

Фінальна нагорода – 59

Максимальна нагорода – 65

**Оцінка результатів:** для досягнення цілі потрібно зробити щонайменше 35 кроків, тобто ідеальне значення нагороди за прохід -1\*34+99=65, чого зміг досягти алгоритм, тому можна стверджувати, що алгоритм досягнув оптимальної стратегії і загалом його застосування є доцільним. Відповідно до графіку, нагорода стабілізувалася орієнтовно після 40 ітерацій алгоритму. Оскільки при виконанні алгоритму було поставлено обмеження у 10000 кроків, а нагорода ніколи не доходила до -10000, то алгоритм завжди знаходив цільовий стан.

**Задача дослідження впливу параметра алгоритму:** вплив коефіцієнта знецінення.

Коефіцієнт знецінення 1

Зображення, що містить знімок екрана, візерунок

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, Графік, ряд

Автоматично згенерований опис

Фінальна нагорода – 58

Максимальна нагорода – 63

Коефіцієнт знецінення 0.95

Зображення, що містить знімок екрана, візерунок

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, Графік, ряд

Автоматично згенерований опис

Фінальна нагорода – 59

Максимальна нагорода – 63

Коефіцієнт знецінення 0.5

Зображення, що містить знімок екрана, візерунок

Автоматично згенерований опис

Зображення, що містить текст, Графік, знімок екрана, схема

Автоматично згенерований опис

Фінальна нагорода – 60

Максимальна нагорода – 65

Коефіцієнт знецінення 0.05

Зображення, що містить знімок екрана, візерунок

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, монітор, ряд

Автоматично згенерований опис

Фінальна нагорода – -10000

Максимальна нагорода – -415

Коефіцієнт знецінення 0

Зображення, що містить знімок екрана, візерунок

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, монітор, Прямокутник

Автоматично згенерований опис

Фінальна нагорода – -10000

Максимальна нагорода – -1961

При значенні коефіцієнта знецінення від 0.5 до 1 немає значної різниці графіків нагород, хоча найкраща максимальна нагорода виявилася саме для коефіцієнта знецінення 0.5. З коефіцієнтами 0.05 та 0 алгоритм з часом переставав знаходити рішення задачі, тому взагалі не підходять для вирішення задачі. Варто відмітити поведінку значення Q-функції, яка для вищих значень коефіцієнта знецінення «прокладає» дорогу додатними значеннями і у сторону зменшення коефіцієнта значення функції зменшується і повний шлях прослідкувати важче.