به نام خدا

پروژه دوم: تاکسی خودران

درس: مبانی هوش مصنوعی

اعضا:

علی پورقیصری

طاها داوری

محمدامین مولوی زاده

الگوریتم مورد استفاده در این کد Q-learning است.

زیرا Q-Learning: یک الگوریتم بدون مدل (Model-free) و بدون سیاست (Off-policy) است.

```
import gym
import numpy as np
import pickle, os


env = gym.make("Taxi-v3")
```

این کد کتابخانه gym را برای ایجاد محیط و کتابخانه numpy را برای انجام عملیات آرایه وارد می کند. همچنین کتابخانه os را برای اهداف مدیریت فایل است. سپس، یک نمونه از محیط Taxi-v3 با استفاده از متد gym.make() ایجاد می کند.

```
number_of_actions = env.action_space.n
number_of_states = env.observation_space.n


Q = np.zeros([number_of_states, number_of_actions])
reward = None
```

فضای مشاهده به تعداد حالت های ممکنی که عامل می تواند در آن باشد اشاره دارد، در حالی که فضای عمل به تعداد اقداماتی که عامل می تواند در هر حالت انجام دهد اشاره دارد. این کد یک جدول Q با اندازه [تعداد حالات، تعداد عملکرد] را مقداردهی می کند، که در آن هر ورودی نشان دهنده پاداش مورد انتظار برای انجام یک اقدام خاص در یک وضعیت خاص است. پاداش متغیر به None مقداردهی اولیه می شود.

```
max_iter_number = 1000
G = 0 #goal state
alpha = 0.618

for episode in range(1,max_iter_number+1):
    isDone = False
    G, reward = 0,0
    state = env.reset()

    while isDone != True:
        action = np.argmax(Q[state])
        nextState, reward, isDone, info = env.step(action)
        Q[state, action] += alpha * (reward + np.max(Q[nextState]) - Q[state,
action])
        G += reward
        state = nextState
```

max_iter_number حداکثر تعداد قسمت ها را برای آموزش نماینده تعیین می کند. کد قبل از هر قسمت متغیرهای isDone، G و reward را مقداردهی اولیه می کند. در این حلقه، عامل اقدامی را بر اساس وضعیت فعلی انتخاب می کند و با استفاده از الگوریتم Q-learning جدول Q را به روز می کند.

- max_iter_number حداکثر تعداد اپیزودهایی را تعیین می کند که نماینده برای آنها آموزش می دهد.
- G پاداش تجمعی است که نماینده در طول یک قسمت دریافت می کند و در شروع هر قسمت به ۰ بازنشانی می شود.
- alpha نرخ یادگیری است که تعیین می‌کند اطلاعات جدید چقدر بر به‌روزرسانی مقادیر Q تأثیر می‌گذارد.
- isDone نشان می دهد آیا با رسیدن به حالت هدف یا با تجاوز از حداکثر تعداد مراحل مجاز، قسمت به پایان رسیده است یا خیر.
- reward، پاداش فوری دریافت شده توسط عامل برای انجام یک اقدام در یک وضعیت خاص است.
- state وضعیت فعلی محیط است که با فراخوانی () env.reset مقداردهی اولیه می شود.
- G با اضافه کردن پاداش فوری به روز می شود.
- state به حالت بعدی به روز می شود.

```
state = env.reset()
isDone = None

while isDone != True:
    action = np.argmax(Q[state])
    state, reward, isDone, info = env.step(action)
    env.render()
```

در این قسمت عامل آموزش دیده را تست می‌کنیم و مراحل را نیز نمایش می‌دهیم.

```
with open("TaxiProblem_Qtable.pkl", 'wb') as f:
    pickle.dump(Q, f)
```

این قطعه کد، Q-table را که یک فرهنگ لغت حاوی مقادیر عملکرد وضعیت است، در فایلی با نام TaxiProblem_Qtable.pkl با استفاده از تابع dump کتابخانه‌ی pickle ذخیره می کند. آرگومان 'wb' مشخص می کند که فایل باید در حالت باینری برای نوشتن باز شود. این اجازه می دهد تا محتویات جدول Q در قالب دودویی برای استفاده بعدی ذخیره شوند.

```
with open("TaxiProblem_Qtable.pkl", 'rb') as f:
    QtestFile = pickle.load(f)

print(QtestFile)
```

با استفاده از دستور بالا می‌توان فایل ذخیره را شد فراخوانی، استفاده و چاپ کرد.