

گزارش پروژه سوم

پروژه هزارتو

درس: مبانی و کاربردهای هوش مصنوعی

استاد راهنما: دکتر حسین کارشناس نجف آبادی

اعضای گروه:

علی‌اکبر احراری- 4003613001

مهرآذین مرزوق- 4003613055

پاییز 1402

فهرست

[گزارش کار الگوریتم 3](#_Toc154444543)

[ساخت محیط 3](#_Toc154444544)

[ساخت q-table و مقداردهی اولیه پارامترها 3](#_Toc154444545)

[عملیات اصلی 4](#_Toc154444546)

[رسم نمودار 6](#_Toc154444547)

[نمونه‌ای از خروجی 7](#_Toc154444548)

[کتابخانه‌های استفاده شده 8](#_Toc154444549)

[منابع 9](#_Toc154444550)

# گزارش کار الگوریتم

# ساخت محیط

ابتدا با استفاده از کتابخانه Gym به ساخت محیط کلی پروژه می‌پردازیم. در کد بعدی برنامه به وسیله تابع env.reset()، محیط برنامه را به صورت مجدد تنظیم کرده و مقدار بازگشته که حالت اولیه می‌باشد را درون observation ذخیره می‌کنیم.

env = gym.make("maze-random-10x10-plus-v0")  
observation = env.reset()

# ساخت q-table و مقداردهی اولیه پارامترها

جدول Q که متشکل از یک جدول 10 در 10 می‌باشد را مقداردهی اولیه می‌کنیم. همچنین تعداد عملیات ممکن(Actions) را داخل متغیر num\_actions ذخیره می‌کنیم.

در ادامه پارامترهای مورد نیاز در برنامه را مقداردهی اولیه می‌کنیم.

# Q-learning parameters  
alpha = 0.1 # Learning rate  
gamma = 0.9 # Discount factor  
epsilon = 0.99 # Epsilon-greedy parameter  
min\_epsilon = 0.1 # Minimum epsilon value  
epsilon\_decay = 0.99 # Epsilon decay rate  
  
NUM\_EPISODES = 1000 # Maximum episodes  
total\_finds = 0 # Total number of times the agent finishes the maze  
episodes = list() # Episodes agent finished the maze

# عملیات اصلی

با ورود به حلقه اصلی برنامه، ابتدا محیط را راه‌اندازی مجدد می‌کنیم. مقادیری را نیز برای دنبال کردن مجموع پاداش، قدم‌ها و همچنین بررسی به پایان رسیدن قسمت‌ها درنظر می‌گیریم. در ادامه وارد حلقه دوم برنامه شده و با env.render() محیط برنامه را به صورت بصری مجسم می‌کنیم. با استفاده از Q-values به‌دست آمده از جدول Qها، به صورت حریصانه به یافتن بهترین حرکت بعدی می‌پردازیم.

for episode in range(NUM\_EPISODES):  
  
 state = env.reset()  
 state = int(state[0] \* 10 + state[1])  
  
 total\_reward = 0  
 steps = 0  
 done = False  
  
 while not done and steps <= 100:  
 env.render()  
  
 action = np.argmax(Q[state]) # Greedy action based on Q-values

سپس با استفاده از تابع step و قراردادن action به‌دست آمده به‌عنوان ورودی این تابع، حرکت مورد نظر را عملی می‌کنیم. سپس حالت بعدی برنامه را پیدا می‌کنیم. حال با استفاده از آن‌ها، جدول Q را با استفاده از فرمول Q-learningبه‌روزرسانی می‌کنیم. در ادامه با استفاده از یک شرط، بررسی می‌کنیم آیا عامل به‌صورت موفقیت‌آمیز به هدف رسیده یا خیر. در صورت رسیدن موفقیت‌آمیز، آن قسمت را به لیست قسمت‌های موفق می‌افزاییم. پس از بررسی شرط، یک قدم به جلو می‌رویم.

# Perform the chosen action  
next\_state, reward, done, \_ = env.step(action)  
next\_state = int(next\_state[0] \* 10 + next\_state[1])  
  
# Update Q-value for the state-action pair  
Q[state][action] += alpha \* (reward + gamma \* np.max(Q[next\_state]) - Q[state][action])  
  
total\_reward += reward  
state = next\_state  
  
if done:  
 total\_finds += 1  
 episodes.append(episode)  
 break  
  
steps += 1

Epsilon پارامتر اکتشاف- بهره‌برداری را با تجزیه آن پس از هر قسمت بر اساس میزان decay-rate (Epsilon\_decay) به‌روزرسانی می‌کند و با پیشرفت آموزش، تغییر به سمت exploitationرا تضمین می‌کند.

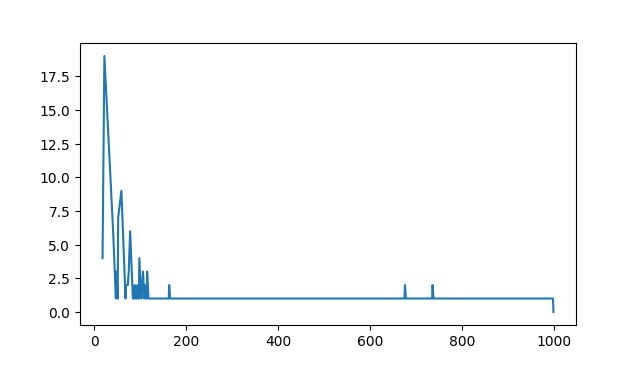
# Epsilon decay after each episode  
epsilon = max(min\_epsilon, epsilon \* epsilon\_decay)

# رسم نمودار

در بخش نهایی پروژه، به رسم یک نمودار می‌پردازیم که در آن به نمایش تعداد قدم‌های منجربه برد در هر قسمت می‌پردازیم. Yها نشان‌دهنده تعداد قدم‌های برداشته شده و Xها نشان‌دهنده هر قسمت می‌باشند. در پایان برنامه را می‌بندیم.

y = list()  
for i in range(len(episodes) - 1):  
 y.append(episodes[i + 1] - episodes[i])  
y.append(0)  
x = episodes  
plot.plot(x, y)  
plot.show()  
# Close the environment  
env.close()

# نمونه‌ای از خروجی



Winning rate: 0.916

# کتابخانه‌های استفاده شده

Numpy: انجام محاسبات عددی و کار با آرایه‌ها

Matplotlib: رسم نمودار

Gym و Gym\_maze: ساخت محیط مناسب

# منابع

[An introduction to Q-Learning: Reinforcement Learning (floydhub.com)](https://blog.floydhub.com/an-introduction-to-q-learning-reinforcement-learning/)

[An Introduction to Q-Learning: A Tutorial For Beginners | DataCamp](https://www.datacamp.com/tutorial/introduction-q-learning-beginner-tutorial)

[Bing Chat with GPT-4](https://www.bing.com/search?form=NTPCHB&q=Bing+AI&showconv=1)

Artificial Intelligence: A Modern Approach, Textbook by Peter Norvig and Stuart J. Russell