

● تمرینات هوش مصنوعی

مصنوعی، به عنوان یک علم مبتنی بر رایانه، به دنبال توسعه عامل‌های هوشمند است که بتوانند در مواقع مختلف بهترین رفتار را از خود نشان دهند. استدلال منطقی یکی از مهارت‌های اساسی در هوش مصنوعی است. 1- این استدلال به معنی توانایی عامل‌های هوشمند در استنتاج‌های منطقی و صحیح برای حل مسائل و تصمیم‌گیری است. 2- برای این منظور، داده‌های دقیق و کاملی لازم است تا عامل‌ها بتوانند به درستی استدلال کنند.

استدلال منطقی در هوش مصنوعی به دو صورت می‌تواند انجام شود:

3- استدلال ریاضی: در این روش، از قوانین منطق ریاضی استفاده می‌شود. مثلاً اثبات قضیه‌ها یا تحلیل مسائل با استفاده از قوانین منطق ریاضی.

4- استدلال زبانی: در این حالت، از مدل‌های زبانی بزرگ استفاده می‌شود تا به هوش مصنوعی کمک کند تا از زبان طبیعی استنتاج‌های منطقی برای تصمیم‌گیری کند.

5- استدلال منطقی، ابزاری قدرتمند برای توسعه هوش مصنوعی است و توانایی آن در حل مسائل و تصمیم‌گیری را افزایش می‌دهد.

● تمرین : 8 وزیر

```
from typing import List, Tuple
from pprint import pprint
import random
```

```
def empty_slots(board: List[List[int]]) ->
List[Tuple[int, int]]:
    result = [(x//8, x%8) for x in range(64)]
    for i, row in enumerate(board):
        for j, slot in enumerate(row):
            if slot != 0:
                for k in range(8):
                    if (j, k) in result:
                        result.remove((j, k))
                    if (k, i) in result:
                        result.remove((k, i))
                for k in range(j, -1, -1):
                    if (k, i-abs(k-j)) in result:
                        result.remove((k, i-abs(k-j)))
                for k in range(j, 8):
                    if (k, i+abs(k-j)) in result:
                        result.remove((k, i+abs(k-j)))
                for k in range(i, -1, -1):
                    if (j+abs(k-i), k) in result:
```

```
        result.remove((j+abs(k-i), k))
    for k in range(i, 8):
        if (j-abs(k-i), k) in result:
            result.remove((j-abs(k-i), k))
```

```
    return result
```

```
if __name__ == "__main__":
    left_queens = 8
    board = None
```

```
while left_queens:
    left_queens = 8
    board = [[0] * 8 for _ in range(8)]
```

```
    while True:
        empty = empty_slots(board)
        if len(empty) == 0:
            break
        x, y = random.choice(empty)
        board[y][x] = 1
        left_queens -= 1
```

```
pprint(board)
```

■ خروجی :

```
[[0, 0, 0, 0, 1, 0, 0, 0],  
[0, 1, 0, 0, 0, 0, 0, 0],  
[0, 0, 0, 1, 0, 0, 0, 0],  
[0, 0, 0, 0, 0, 0, 1, 0],  
[0, 0, 1, 0, 0, 0, 0, 0],  
[0, 0, 0, 0, 0, 0, 0, 1],  
[0, 0, 0, 0, 0, 1, 0, 0],  
[1, 0, 0, 0, 0, 0, 0, 0]]
```

هشت وزیر یک معمای شطرنجی و ریاضیاتی است که بر اساس آن باید n وزیر شطرنج در یک صفحه $n \times n$ شطرنج به گونه ای قرار داده شوند که هیچ یک زیر ضرب دیگری نباشند. وزیر به صورت افقی، عمودی و اریب حرکت می کند، بنابراین هر وزیر را در طول، عرض و قطر متفاوتی قرار داد. اولین و مشهورترین شکل این مسئله معمای **هشت وزیر** است که برای حل آن باید ۸ وزیر را در یک صفحه معمولی (8×8) شطرنج قرار داد. این مسئله ۹۲ جواب دارد که ۱۲ جواب آن منحصر به فرد است، یعنی بقیه جواب ها از تقارن جواب های اصلی به دست می آید. مسئله n وزیر در صورتی جواب دارد که n مساوی ۱ یا بیشتر از ۳ باشد. یعنی مسئله دو وزیر و سه وزیر راه حلی ندارند 1.

برای حل این مسئله، می توان از الگوریتم های مختلفی مانند الگوریتم عقب گرد، الگوریتم مونت کارلو، روش مکاشفه ای،

الگوریتم ژنتیک و ... استفاده کرد1. این مسئله از جمله مسائل NP در هوش مصنوعی است که روش‌های جستجوی معمولی قادر به حل آن‌ها نخواهد بود1.



