موضوع: تمرین های هوش مصنوعی

استاد محترم: سرکار خانم دکتر عصایی معمم

دانشجو: کیمیاهاشم پور

شماره دانشجویی: 40016341054144

سال: نیم سال دوم – 1402

---

## تمرین اول : peas ربات فوتبالیست

### معیارکارایی:

برد بازی ، گل زدن بیشتر از تیم حریف ، بیرون نرفتن از چهارچوب بازی ، خطا نکردن رو ی بازیکن حریف ، سرعت جابجایی قابل قبول برای پشت سر گذاشتن بازیکن تیم حریف ، توانایی شناسایی و تشخیص عوامل محیطی و فیزیکی بازی ، استفاده نکردن دست بجز دروازه بان

### محیط:

زمین چمن، فوتسال ، ساحل

### عملگرها:

شوت ، چیپ ، سانتر کردن

### سنسور:

سنسور تشخیص توپ ، سرعت توپ و جهت آن ، سنسور تشخیص فاصله ، سنسور آنالیز بازیکنان حریف ، سنسور خطوط سنسور تشخیص بازیکن حریف یا بازیکن خودی ، دروازه ها ، سنسور اب و هوا

## تمرین دوم : با مسائل غیر قطعی چگونه رفتار میکنیم؟

راه حل مسائل غیر قطعی در هوش مصنوعی مرتبط با مدیریت و تصمیم گیری در شرایطی که دارای عدم قطعیت هستند می باشد. برای حل اینگونه مسائل، میتوان از رویکردها و تکنیک های زیر استفاده کرد:

### 1) احتماالت و آمار:

استفاده از مفاهیم احتماالت و آمار برای مدل ساز ی و پیش بینی وقوع رویدادها در شرایط عدم قطعیت.

### 2) مدل سازی بیزی :

استفاده از مدل های بیزی برای نمایش علاقه مندی ها و توزیع های احتمالی در مسائل غیر قطعی.

### 3) تئور ی تصمیم گیر ی :

اعمال تکنیک های تصمیم گیری چون مدل های مارکوف تصمیم گیری و فرآیندهای تصمیم گیری مارکوف برای تعیین تصمیم های بهینه در شرایطی که دارای عدم قطعیت هستند.

### 4) اطلاعات فازی :

استفاده از اطلاعات فازی برای مدل سازی عدم قطعیت و عدم دقت در داده ها و تصمیم گیری ها.

### 5) تکنیک های ترکیبی :

ترکیب اطلاعات احتمالی و داده های مشاهده شده با دانش پیشین و تجربی به منظور بهبود تصمیم گیری در شرایط عدم قطعیت

### 6) الگوریتم های بهینه سازی :

استفاده از الگوریتم های بهینه سازی برای یافتن راه حل های بهینه در مسائل غیرقطعی

### 7) تکنیک های تحلیل حساسیت :

تجزیه و تحلیل حساسیت برای درک تأثیر پارامترها و عوامل مختلف بر نتایج تصمیم گیری در شرایط عدم قطعیت

8) شبکه های عصبی :

استفاده از شبکه های عصبی برای مدل سازی و پیش بینی در شرایط عدم قطعیت

ترکیبی از این رویکردها و تکنیکها بسته به مسئله مورد نظر و میزان عدم قطعیت می تواند به راه حل های موثری در مسائل غیر قطعی در هوش مصنوعی منجر شود

## تمرین سوم : کد ۸ وزیر رو پیدا کنید و درمورد آن بررسی کنید

چک کردن آیا می توان وزیری را در سلول قرار داد یا خیر:

```
def is_safe(board, row, col, n):
```

چک کردن رد ی فافقی ( سمت چپ ):

```
for i in range(col):
    if board[row][i] == 1:
        return False
```

چک کردن قطر بالا به چپ:

```
for i, j in zip(range(row, -1, -1), range(col, -1, -1)):
    if board[i][j] == 1:
        return False
```

چک کردن قطر پایین به چپ:

```
for i, j in zip(range(row, n, 1), range(col, -1, -1)):
    if board[i][j] == 1:
        return False
```

حالت پایه : اگر تمام وزیرها قرار گرفته باشند:

```
def solve_n_queens_util(board, col, n):
    if col >= n:
        return True
```

برای هر سلول در ستون فعلی:

```
for i in range(n):
```

چک کردن آیا می توان وزیر را در این سلول قرار داد:

```
if is_safe(board, i, col, n):
```

قرار دادن وزیر در این سلول:

```
board[i][col] = 1
```

ادامه به جستجوی ستون بعدی:

```
if solve_n_queens_util(board, col + 1, n):
    return True
```

اگر قرار گرفتن وزیر در این سلول به حل مسئله منجر نشود آن را از صفحه

حذف میکنیم:

```
board[i][col] = 0
```

اگر هیچ یک از سلول ها منجر به حل مسئله نشود:

```
def solve_n_queens(n):
    return False
```

ایجاد صفحه شطرنج خالی:

```
board = [[0 for _ in range(n)] for _ in range(n)]
```

حل مسئله با فراخوانی اولیه از ستون اول:

```
if not solve_n_queens_util(board, 0, n):
    print (هیچ راه حلی وجود ندارد)
    return False
```

نمایش جواب:

```
for i in range(n):
    for j in range(n):
        print(board[i][j], end=" ")
    print()
return True
```

برای حل مسئله 8 وزیر  n = 8 تابع را فراخوانی میکنیم با:

```
solve_n_queens(8)
```

تمرین چهارم : به نظر شما پیچیدگی زمانی دستو پاگیر تره یا پیچیدگی حافظه؟

بستگی به نوع مسئله و الگوریتمی که برای حل آن استفاده میشود دارد که پیچیدگی حافظه یا زمانی کدامیک بیشتر است. در برخی موارد مسائلی وجود دارند که پیچیدگی حافظه آنها بسیار بالاست و نیاز به استفاده از منابع حافظه بالا دارند. برای مثال الگوریتم هایی که برای پردازش

تصویر و صدا استفاده می شوند به دلیل بزرگی حجم داده های ورودی نیاز به استفاده از حافظه بالا دارند

در مقابل در بسیاری از مسائل پیچیدگی زمانی بیشتر از پیچیدگی حافظه است. به عنوان مثال الگوریتم هایی که برای مرتب سازی اعداد استفاده میشوند نیاز به حافظه کمتری دارند ولی زمان بیشتری برای اجرای آن ها لازم است.

بنابراین برای انتخاب بهترین الگوریتم برای حل یک مسئله باید به دو پیچیدگی حافظه و زمانی توجه کرد و الگوریتمی را انتخاب کرد که برای آن مسئله پیچیدگی کمتری داشته باشد

تمرین پنجم :

## تحلیل WUMPUS

Wumpus یک بازی کامپیوتری است که در آن بازیکنان باید در یک شبکه از اتاق ها و تونل ها حرکت کنند و به دنبال هیولای و امپوس بگردند. هدف اصلی این بازی یافتن گنجینه و خروج از محل بازی است. اما بازیکنان باید همچنین از مخاطرات مختلفی مانند چاه ها موشهای سمی و هیولای و امپوس بپرهیزند.

بازی Wumpus یک بازی استراتژیک است که نیاز به تصمیم گیری هوشمندانه برای حرکت در محیط دارد. بازیکنان باید از شاخصهای مختلف محیط استفاده کنند تا مکانهای مختلف را کشف کرده و خطرات را پیش بینی کنند. همچنین آنها باید به طور منظم از تجهیزات مختلف مانند فانوس شمع و نقشه استفاده کنند تا بهترین راه حل برای پیدا کردن گنجینه و خروج از محل بازی را پیدا کنند.

با توجه به محیط پرخطر و موانع مختلف بازی Wumpus چالش های زیادی را برای بازیکنان ارائه می دهد. این بازی نه تنها نیاز به استراتژی و تصمیم گیری هوشمندانه دارد بلکه همچنین مهارتهای حل مسئله و تجزیه

و تحلیل را نیز تقویت می کند.

مشابه دنياي مکش دنياي Wumpus شبکه ای از مربع است که توسط دیوارهایی احاطه شده اند. که هر مربع می تواند شامل عاملها و اشیاء باشد. وظیفه عامل یافتن طلا و بازگشتن به نقطه شروع و بالا رفتن از غار است. براي مشخص نمودن وظیفه عامل ادراکات عملیات و اهداف آن را باید مشخص کنیم. در دنیای Wumpus اینها به صورت زیر هستند:

از مربعی که شامل Wumpus است و مربع هاي مجاور (نه) (قطري) عامل بوي بدي را دریافت د ت می کند.

در مربعهايي که مستقيماً مجاور با چاله ها هستند عامل نسیمی را دریافت می کند.

در مربعی که طلا وجود دارد. عامل یک درخششی را درک می کند.

زمانی که یک عامل به داخل دیواره قدم بر می دارد ضربه ای را دریافت می کند.

زمانی که Wumpus کشته میشود. فریادی سر می دهد که هر جایی از غار شنیده می شود.

ادراکات به عامل به صورت لیستی از پنج سیمبول داده می شود.

مانند دنياي مكش، عملايتي براي جلو رفتن چرخيدن 90 به سمت چپ چرخيدن 90 به سمت راست وجود دارد.

عامل نابود خواهد شد زمانی که وارد یک مربع شامل سیاده چاله و یا کي Wumpus زنده می شود.

هدف عامل یافتن طلا و برگرداندن آن به خانه شروع با سرعت تمام است. بدون آنکه کشته شود.