عنوان: حل تمرین هوش مصنوعی

نام ونام خانوادگی: علی محمد ابراهیمی

شماره دانشجوبی: 40016340893063

نام استاد: سرکار خانم عصایی معمم

تمرین شماره 1 : PEAS ربات فوتبالیست را بنویسید:

برخورد با مسأل غیر قطعی:

رفتار با مسائل غیر قطعی در دنیای واقعی:

رفتار با مسائل غیر قطعی در د نیای واقعی اغلب نیازمند انعطاف و تعامل با محیط است. این شامل استفاده از استراتژیهایی مانند ارزیابی ریسک، پذیرش عدم قطعیت، و استفاده از منابع و تجربههای قبلی برای اتخاذ تصمیمات است. همچنین، توانایی پذ یرش اینکه ممکن است اطلاعات مورد نیاز برای .تصمیم گیری کامل نباشد و بهبود پذیری در برابر تغییرات محیطی نیز از اهمیت بالایی برخوردار است

به طور کلی، مواجهه با مسائل غیر قطعی به معنای داشتن اطلاعات ناقص یا محدود در مورد وضعیت یا پیشبینیهای آینده است. در د نیای واقعی ، بسیار ی از تصمیمات بر اساس این نوع اطلاعات گرفته میشوند. برای مثال، در حوزه کسب و کار، تصمیمات استراتژیک باید بر اساس پیشبینیهایی از بازار، رقبا، و شرایط اقتصادی گرفته شود که همگی دارای عدم قطعیت هستند. برای مدیریت ا ین عدم قطعیت، افراد با ید از ابزارهایی مانند مدلسازی ریسک ، ارز یا بی تأ ثیرات مختلف، و استفاده از دادهها و تجربیات قبلی استفاده کنند تا تصمیمات بهتری بگیرند. این همچنینن یازمند انعطاف پذ یر ی و توانایی سازگاری با تغییرات و شرا یط جدید است

مدلسازی ر یسک: استفاده از مدلهای ر یاضی و آماری برای تخمین و مدیریت ریسکهای مختلفی که باعث عدم قطعیت میشوند، مانند تغییرات در بازار یا تغییرات قوانین و مقررات

پذیرش عدم قطعیت : قبول این واقعیت که همیشه اطلاعات کاملی در دسترس نیست واینکه تصمیمات باید.بر اساس اطلاعات ناقص گرفته شوند

استفاده از تجربه های گذشته: بررسی تجربیات گذشته و استفاده از آنها برای ارزیابی و تصمیمگیری در مورد مواجهه با مسائل جدید

ارتقاء انعطاف پذ یری: توسعه مهارتها و روشهایی برای سازگاری با تغییرات و تطورات ناگها نی در.محیط مشارکت و همکاری: جمع آوری دیدگاهها و نظرات مختلف اعضای گروه و مشارکت آنها در فرآیند تصمیمگیری، که میتواند به تحلیل بهتر مسائل و رسیدن به تصمیمات کارآمدتر کمک کنداستفاده از سنار یوها: تدوین سناریوهای مختلف برای آینده و بررسی تأثیرات مختلف هر سناری و بر تصمیمات حال و آینده

پیشبینی معمول: بر اساس تجربه و دادههای موجود، پیشب ینیهایی در مورد آینده انجام داده و بر اساس آنها تصمیم گیری کرد

|  |  |  |
| --- | --- | --- |
|  | def is\_safe(board, row, col): # Check row on the left side  for i in range(col): if board[row][i] == 1:  return False  # Check upper diagonal on left side for i, j in zip(range(row, -1, -1), range(col, -1, -1)): if board[i][j] == 1: return False  # Check lower diagonal on left side for i, j in zip(range(row, len(board)), range(col, -1, -1)): if board[i][j] == 1:  return False return True def solve\_queens(board, col): if col >= len(board):  return True for i in range(len(board)): if is\_safe(board, i, col):  board[i][col] = 1 if solve\_queens(board, col + 1):  return True board[i][col] = 0 return False  def print\_board(board): for row in board:  print(row)  board = [[0]\*8 for \_ in range(8)] if solve\_queens(board, 0):  print\_board(board) else:  print("No solution exists.") | 8وزیر: |

|  |  |
| --- | --- |
| این کد با استفاده از روش بازگشتی به حل مسئله میپردازد و در صورت وجود راه حل، صفحه شطرنجی را که محل قرارگیری هر وزیر را نشان میدهد، چاپ میکند  تابعis\_safe بررسی میکند که آیا میتوان یک وزیر را در مکان مشخصی از صفحه شطرنجی قرار داد یا نه. برای این منظور، تابع ابتدا از تمام وزیرهای سمت چپ و سپس از قطرهای بالا و پایین به سمت چپ برای بررسی تهدیدهای ممکن استفاده میکند.  تابعsolve\_queens از یک حلقه برای گردش در ستونهای صفحه شطرنجی استفاده میکند و برای هر ستون، تلاش میکند تا یک موقعیت امن برای قرار دادن وزیرها پیدا کند. اگر راه حل پیدا شود، تابع solve\_queens بازمیگردد و درغیر ا ین صورت، بازگشت به مرحله قبلی رخ میدهد.  تابعprint\_board صفحه شطرنجی را به صورت ماتریس چاپ میکند، که در آن مقاد یر 1 نشان دهنده قرار گرفتن وز یر در آن موقعیت و مقاد یر 0 نشان دهنده خانه های خالی است.  در انتها، ابتدا یک صفحه شطرنجی خالی ا یجاد میشود و سپس تابعsolve\_queens صدا زده میشود تا راه حل را بیابد. اگر راه حل پیدا شود، صفحه شطرنجی با راه حل چاپ میشود، درغیر این صورت  .چاپ میشود "No solution exists." پیام   |  | | --- | | [0, 0, 0, 0, 0, 1, 0, ]0  [0, 0, 0, 0, 0, 0, 0, ]0  [0, 0, 0, 0, 0, 0, 1, ]0  [0, 0, 0, 0, 0, 0, 0, ]0  [0, 0, 0, 0, 0, 0, 0, ]1  [0, 0, 0, 0, 0, 0, 0, ]0  [0, 0, 0, 0, 0, 0, 0, ]0  [0, 0, 0, 0, 0, 0, 0, ]0 | |

)(PVH)˄˜H)→P