



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)
دانشکده ریاضی و علوم کامپیوتر

گزارش درس کارشناسی
هوش مصنوعی

گزارش پروژه ۴ درس هوش مصنوعی

هستی برقراریان

استاد درس
مهدی قطعی

مدرس کارگاه
بهنام یوسفی مهر

خرداد ۱۴۰۱

چکیده

در این پروژه، سعی داریم که بازی سودوکو را با استفاده از ارضای محدودیت و روش های فراابتکاری پیاده سازی کنیم.

فصل اول

مقدمه

مقدمه

در این پروژه، قصد داریم بازی سودوکو را با استفاده از روش‌های هوش مصنوعی پیاده‌سازی کنیم. سودوکو یک بازی منطقی است که شامل یک جدول 9×9 سلولی است که باید آن را به طور کامل با اعداد از ۱ تا ۹ پر کنید. هر سطر، ستون و بلوک 3×3 جدول باید شامل تمام اعداد از ۱ تا ۹ باشد.

برای حل این بازی، از روش‌های ارضای محدودیت و روش‌های فراابتکاری پیشرفته استفاده می‌کنیم. از جمله روش‌های استفاده شده می‌توان به الگوریتم **Backtracking** و **Constraint Propagation** اشاره کرد. الگوریتم **Backtracking** یک الگوریتم بازگشتی است که با امتحان کردن اعداد مختلف در هر خانه و حل مسئله در صورت تطابق می‌رود و در صورت ناتطابق بودن به مرحله قبل برمی‌گردد. **Propagation Constrain** نیز با استفاده از قوانین و محدودیت‌های بازی، محدوده اعداد ممکن در هر خانه را کاهش می‌دهد.

در فصل دوم توضیحاتی درباره‌ی کدهای به کار رفته ارائه می‌شود و فصل سوم را به جمع‌بندی و نتیجه‌گیری می‌پردازیم.

فصل دوم

پیاده‌سازی

پیاده‌سازی

کدی که در نوت بوک قرار دارد، این کد است:

```
def solve_sudoku(board):
    if not find_empty_cell(board):
        return True

    row, col = find_empty_cell(board)
    available_numbers = get_available_numbers(board, row, col)

    for num in available_numbers:
        board[row][col] = num

        if solve_sudoku(board):
            return True

        board[row][col] = 0

    return False

def find_empty_cell(board):
    for i in range(9):
        for j in range(9):
            if board[i][j] == 0:
                return i, j
    return None

def get_available_numbers(board, row, col):
    available_numbers = set(range(1, 10))

    # Remove numbers in the same row
    for j in range(9):
        if board[row][j] in available_numbers:
            available_numbers.remove(board[row][j])

    # Remove numbers in the same column
    for i in range(9):
        if board[i][col] in available_numbers:
            available_numbers.remove(board[i][col])

    # Remove numbers in the same 3x3 box
    start_row = 3 * (row // 3)
    start_col = 3 * (col // 3)
    for i in range(3):
        for j in range(3):
            if board[start_row + i][start_col + j] in available_numbers:
                available_numbers.remove(board[start_row + i][start_col
+ j])

    return available_numbers
```

این کد حل کردن بازی Sudoku را با استفاده از روش Backtracking پیاده‌سازی می‌کند. الگوریتم Backtracking یک الگوریتم بازگشتی است که با تلاش برای قرار دادن اعداد مختلف در هر خانه و پیدا کردن تطابق، به مراحل بعدی می‌رود. در صورت عدم تطابق، به مراحل قبلی بازگشت می‌کند. تابع solve_sudoku ابتدا خانه‌های خالی را با استفاده از تابع find_empty_cell پیدا می‌کند. سپس با استفاده از تابع get_available_numbers اعداد مجاز برای هر خانه را دریافت می‌کند. سپس با استفاده از الگوریتم Backtracking، برای هر عدد مجاز در خانه، آن را در خانه قرار می‌دهد و به طور بازگشتی این فرایند را تکرار می‌کند تا بازی Sudoku را حل کند. پس از اجرای تابع solve_sudoku، اگر راه حلی وجود داشته باشد، جدول حل شده را چاپ می‌کند. در غیر این صورت، پیغام "No solution exists." را چاپ می‌کند.



فصل سوم

جمع‌بندی و نتیجه‌گیری

جمع‌بندی و نتیجه‌گیری

طی این پروژه، ما با استفاده از روش Backtracking، بازی Sudoku را پیاده‌سازی کردیم. این الگوریتم با استفاده از مکانیزم بازگشتی، تلاش می‌کند با پر کردن خانه‌های خالی جدول، یک ترکیب صحیح از اعداد ۱ تا ۹ را در هر سطر، ستون و بلوک 3×3 ایجاد کند.

با توجه به پیاده‌سازی اولیه، می‌توان این کد را بهبود بخشید و از روش‌ها و الگوریتم‌های پیشرفته‌تری برای حل Sudoku استفاده کرد. مثلاً می‌توان الگوریتم‌هایی مانند الگوریتم‌های ژنتیک یا الگوریتم‌های تطبیقی به کد اضافه کرد تا بهبود و بهینه‌سازی بیشتری در حل Sudoku ایجاد شود.

منابع و مراجع

[1] <https://www.geeksforgeeks.org/sudoku-backtracking-7/>

[2] <https://iq.opengenus.org/backtracking-sudoku/>

[3] Chat GPT

Abstract

Abstract

In this project, we are trying to implement the Sudoku game using constraint satisfaction and meta-heuristic methods.



**Amirkabir University of Technology
(Tehran Polytechnic)**

Department of Mathematics and Computer science

AI project

Sudoku

**By
Hasti Bargharariyan**

**Supervisor
Dr. Mahdi Ghatee**

**Advisor
Behnam Yousefimehr**

June 2023