UNIT 2

- Applications of AI
  - Game Playing
  - Deep Blue in Chess,
  - IBM Watson in Jeopardy,
  - Google'sDeep Mind in AlphaGo

List of Experiment(s):
1. Implementation of (Sudoku, Crossword Puzzle, or Wumpus World, etc)

Game Playing

- It is an important domain of artificial intelligence
- Game playing is a popular application of artificial intelligence that involves the development of computer programs to play games, such as chess, checkers, or Go
- Games don't require much knowledge; the only knowledge we need to provide is the rules, legal moves and the conditions of winning or losing the game
- Both players try to win the game. So, both of them try to make the best move possible at each turn
- Searching techniques like BFS(Breadth First Search) are not accurate for this as the branching factor is very high, so searching will take a lot of time
- So, we need another search procedures that improve –
- **Generate procedure** so that only good moves are generated
- **Test procedure** so that the best move can be explored first

Contd..

- The goal of game playing in artificial intelligence is **to develop algorithms that can learn how to play games and make decisions** that will lead to winning outcomes
- One of the earliest examples of successful game playing AI is the chess program Deep Blue, developed by IBM, which defeated the world champion Garry Kasparov in 1997.
- AI has been applied to a wide range of games, including two-player games, multiplayer games, and video games.

Contd..

- There are two main approaches to game playing in AI
  - Rule-based systems
  - Machine learning-based systems
- Rule-based systems use a set of fixed rules to play the game
- Machine learning-based systems use algorithms to learn from experience and make decisions based on that experience.
- In recent years, machine learning-based systems have become increasingly popular, as they are able to learn from experience and improve over time, making them well-suited for complex games such as Go.
- For example, AlphaGo, developed by DeepMind, was the first machine learning-based system to defeat a world champion in the game of Go.

Game Playing

- **Deep Blue in Chess**
  - First World Champion class chess computer
  - World Chess Champion Garry Kasparov resigned the last game of a six-game match against IBM's Deep Blue supercomputer on 11 May 1997

- **IBM Watson in Jeopardy**
  - Watson is a question-answering computer system capable of answering questions posed in natural language, developed in IBM's DeepQA project by a research team led by principal investigator David Ferrucci.
  - First computer to defeat TV game show Jeopardy! champions (Ken Jennings and Brad Rutter).

- **Google's Deep Mind in AlphaGo**
  - AlphaGo is a computer program that plays the board game Go.
  - It was developed by DeepMind Technologies which was later acquired by Google.
  - AlphaGo versus Lee Sedol, also known as the Google DeepMind Challenge Match, was a five-game Go match between 18-time world champion Lee Sedol and AlphaGo, a computer Go program developed by Google DeepMind, played in Seoul, South Korea between the 9th and 15th of March 2016.
  - It is able to do this by using a novel form of reinforcement learning, in which AlphaGo Zero becomes its own teacher.

Advantages of Game Playing in Artificial Intelligence:

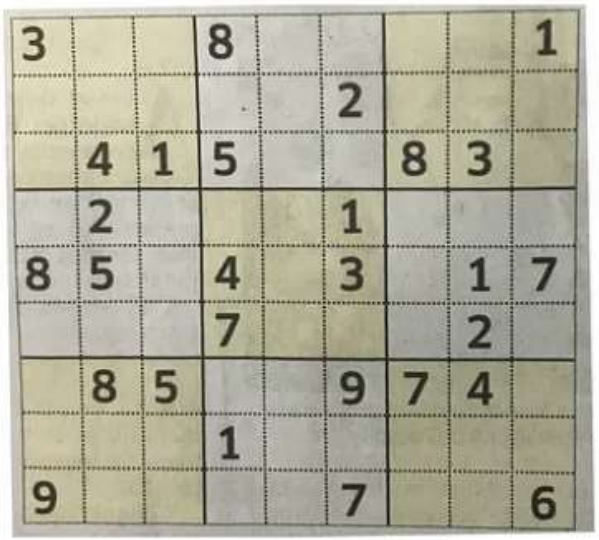
- Advancement of AI: Game playing has been a driving force behind the development of artificial intelligence and has led to the creation of new algorithms and techniques that can be applied to other areas of AI.
- Education and training: Game playing can be used to teach AI techniques and algorithms to students and professionals, as well as to provide training for military and emergency response personnel.
- Research: Game playing is an active area of research in AI and provides an opportunity to study and develop new techniques for decision-making and problem-solving.
- Real-world applications: The techniques and algorithms developed for game playing can be applied to real-world applications, such as robotics, autonomous systems, and decision support systems.

Disadvantages of Game Playing in Artificial Intelligence:

- Limited scope: The techniques and algorithms developed for game playing may not be well-suited for other types of applications and may need to be adapted or modified for different domains.
- Computational cost: Game playing can be computationally expensive, especially for complex games such as chess or Go, and may require powerful computers to achieve real-time performance.

Sudoku

- It is a logic-based, combinatorial number-placement puzzle.
- The objective is to fill a 9×9 grid with digits so that each column, each row, and each of the nine 3×3 subgrids that compose the grid contain all of the digits from 1 to 9.




```
normal_sudoku = [
                    [3,0,0,8,0,0,0,0,1],
                    [0,0,0,0,0,2,0,0,0],
                    [0,4,1,5,0,0,8,3,0],
                    [0,2,0,0,0,1,0,0,0],
                    [8,5,0,4,0,3,0,1,7],
                    [0,0,0,7,0,0,0,2,0],
                    [0,8,5,0,0,9,7,4,0],
                    [0,0,0,1,0,0,0,0,0],
                    [9,0,0,0,0,7,0,0,6]
                ]
```

Generic constraints for Sudoku

The rules for Sudoku need to be set as constraints to solve this problem. Given that Sudoku is a 9x9 grid, the rules of the game are mentioned here:

Constraint 1: Each cell should be filled with a single value between 1 and 9

Constraint 2: Each row should contain every number from 1 to 9 once

Constraint 3: Each column should contain every number from 1 to 9 once

Constraint 4: Each 3x3 grid, starting from top left, should contain every number from 1 to 9 once

Constraint 5: Each diagonal should contain every number from 1 to 9 once

Crossword Puzzle

- A **10 x 10 Crossword grid** is provided, along with a set of words (or names of places) which need to be filled into the grid. The cells in the grid are initially, either **+** signs or – signs. Cells marked with a **'+'** have to be **left** as they are. Cells marked with a **'-'** need to be **filled** up with an appropriate character. You are also given an array of words that need to be filled in Crossword grid.

Input :

```
+++++++++-
-+++++++++-
-------++-
-+++++++++-
-+++++++++-
-++++------
------+++-
-+++++++++-
+---------
++++++++++
```

Output :

```
+++++++++C
P++++++++H
HISTORY++E
Y++++++++M
S++++++++I
I++++MATHS
CIVICS+++T
S++++++++R
+GEOGRAPHY
++++++++++
```

- The approach behind this is to recursively check for each word in the vertical position and in the horizontal position. Then fill the word in the matrix that can be the best fit in the corresponding position of the grid, then update the crossword grid by filling the gap with that word.