UNIT - I

# Introduction to Artificial Intelligence

## Introduction:

The foundation of Artificial Intelligence (AI) was laid with the development of Boolean theory and principles by a mathematician named Boole and others researchers. AI has been of interest to the researchers as they have always aimed to make machines more intelligent than humans and tried to simulate intelligent behaviour.

AI researchers generally use one of the two basic approaches, namely bottom-up and top-down, for creating intelligent machines. In bottom-up approach, the belief is to achieve artificial intelligence by building electronic replicas of the human brain which is a complex network of neurons. In the top-down approach, the attempt is to mimic the behaviour of brain with computer programs.

AI currently comprises of a huge variety of subfields ranging from general-purpose areas such as perception, logical reasoning, to specific tasks such as game playing, theorem proving, diagnosing diseases, etc. AI is engaged in two significantly different enterprises, namely a science of human intelligence and an engineering discipline concerned with building smarter physical systems. This field is truly a multi-disciplinary field and is based on the work done in different disciplines such as logic, cognition, linguistics, philosophy, psychology, anthropology, computing etc. AI programs tend to be large, and it would have not been possible to work unless there is great advancement in speed and memory of computers.

Brief History: There was no relation between Human Intelligence and machine till early 1950 even though philosphers in 400 B.C. Psychologists further strengthened the idea that humans and other living creatures can be considered to be information processing machines. Mathematicians provided tools to manipulate certain or uncertain logic statements of certainity as well as probabilistic statements.

John McCarthy organised a conference on machine intelligence in 1956 and since then the field was known as Artificial Intelligence. In 1957, the first version of a new program named as General Problem solver (GPS) was developed and tested. since, then many programs were developed and McCarthy announced his new development called LISP in 1958. LISP was adopted as the language of choice by most AI developers. Another program named STUDENT, was developed during late 1960 which could solve algebra story problem. Fuzzy set and uncertain conditions, neural networks are considered as possible ways of achieving Artificial Intelligence. Frame Theory and PROLOG language for storing structured knowledge to be used by AI programs. Expert system is a program that uses logical rules that are derived from the knowledge of experts to answer the question. In 1980, research organisations and corporate sector started developing AI systems.

By 1985, over hundred of companies offered machine vision systems in US. More recently, work in AI has started from agent point of view and in understanding the theoretical basics for intelligence has gone hand in hand with improvements in capabilities of real systems.

## Intelligent systems:

- AI is a combination of Computer science, psychology and philosophy. John McCarthy was one of the founders of AI field who stated that AI is the science and engineering of making intelligent machine, especially intelligent programs.
- AI can be defined as a study of making computers do thing intelligently It requires the study and creation of computer systems that exhibit some form of intelligence or the characteristics which we associate with intelligence in human behaviour such as learning, reasoning, inferencing, perceiving and comprehending information.
- Understanding of AI requires understanding of related terms such as intelligence, knowledge, reasoning, cognition, learning and number of other computation related terms
- AI shows best on complex problems for which general principles do not help much. Generally it is thought to be an advanced s/w engineering and sophisticated s/w techniques for hard problems that cannot be solved in any easy way.
- There are two versions of AI goals.
  1) The first view is that AI is about duplicating what the human brain does (congnitive science)
  2) The second view is that AI is about duplicating what the human brain should do i.e., Doing things logically or rationally.
- Turing test was designed to provide a satisfactory operational definition of Intelligence. Turing said that a system is called intelligent if it has ability to pass the turing test.

- A system is said to have passed turing test if a human questionaire is unable to determine from repeated questions of any kind, weather he or she is talking to another person or to a Machine/system.
- The very first so called intelligent system named ELIZA passed the turing test.

ELIZA:

It is a program that conversed with user in English. This program was able to converse about any subject because it stored information in databanks.

Another feature was Eliza was its ability to pic up speech patterns from users questions and provide responses using those patterns. In this mode Eliza mostly rephrased the users statements as questions and posted those to the users.

The main characteristics of Eliza are:-

1) Simulation of Intelligence: ELIZA programs are not intelligent at all in real sense. They do not understand the meaning of utterance. Instead these programs simulate intelligent behaviour quite effectively by recognising keywords and phrases by using a table look up. One of few ways of responding question is chosen.

2) Quality of Response: It is limited by the sophistication of the ways in which they can process the input text at a syntactic level.

Eg: The number of templates available is the serious limitation.

3)

Coherence: The earlier versions of the system imposed no structure on the conversation. Each statement was entirely on the current input and no context information was used. Any sense of intelligence depends strongly on the coherence of the conversation as judged by the user.

4) Semantics: Such systems have no semantic representation of the context of either the users input or the reply. That is why we say that it doesn't have intelligence of understanding of what we are saying. But it looks that it imitates the humans conversation style, because of this it passed turing test.

Categorization of Intelligent Systems:-

In order to design intelligent systems it is important to categorize these systems. There are four possible categorizes of systems

1) Systems that think like human
2) Systems that act like human
3) Systems that think rationally
4) Systems that act rationally

1) System that think like human: It requires congnitive modelling approaches. Most of the time it is a black box where we are not clear about our thought process. One has to know the functioning of the brain and its mechanism for processing information. Neural network is a computing model for processing information similar to brain.

2) System that act like humans: These requires that overall behaviour of the system should be human like which would be achieved by observation. Turing test is an example.

3) System that think rationally: These relies on logic rather than human to measure correctness. For thinking rationally/logically. Logical formulae and theories are used for synthesizing outcomes. Eg: Rama is a human and all humans are mortal then one can conclude logically RAMA is mortal.

4) Systems that acts rationally: This is the final category of intelligent systems where by rational behaviour we mean doing the right thing. Even if the method is illogical the observed behaviour must be rational.

Components of AI program:-

AI techniques must be independent of the problem domain as far as possible. Any AI program should have knowledge base and navigational capability which contains control strategy and inference mechanism.

1) Knowledge Base: AI program should be learning in nature and update it's knowledge accordingly. knowledge base generally consists of facts and rules and has the following characteristics

a) It is voluminous in nature and requires proper structuring.

b) It may be incomplete and imprecise.

c) It may be dynamic and keep on changing.

2) Control strategy: It determines which rule should be applied To know this rule some heuristic or thumb rules based on problem domain may be used.

3) Inference Mechanism: It requires search through knowledge base and drives new knowledge using the existing knowledge with the help of inference rules.

## Foundations of AI:

AI is interdisciplinary in nature, foundations of AI are in various fields such as mathematics, neuroscience, control theory, linguistics.

Mathematics: AI systems use formal logical methods and boolean logic, analysis of limits to what can be computed, probability theory, uncertainity that forms the basics for most modern approaches to AI, fuzzy logic etc..

Neuro Science: This science of medicine helps in studying the functioning of brains. In early studies injured and abnormal people was used to understand what parts of brain worked. Now recent studies use accurate sensors to correlate brain activity to human thought. Researches are working to know as to how to have a mechanical brain, such systems will require parallel computation, remapping and interconnections to a large extent.

Control Theory: Machines can modify their behaviour in response the environment. Steam engine governer, thermostat, waterflo regulators are few examples of control theory.

In 1950 control theory could only describe linear system AI largely rose as a response to this short coming. This theory of stable feed back systems help in building syste

that transition from initial state to goal state with minimum energy.

Linguistics: Speech demonstrates so much of human intelligence. Analysis of human language reveals that thought taking place in ways not understood in our settings. children can create sentences they have never heard before. Languages and thoughts are believed to be tightly into wined.

## Tic-Tac-Toe Game Playing:

Here we see three approaches for solving tic-tac-toe game. It is a 2 player game with one player making 'x' and other making '0' at their turn in the spaces in a 3x3 grid. The player who succeeds in playing 3 respective marks in any horizontal, vertical or diagonal wins the game. Here we are considering one human player and the other be a computer program. The objective is to play this game using computer is to write a program which never loses.

The approaches we use to play this game we increase the complexity, use of generalization, clarity of their knowledge and extensibility of their approach.

### Approach I:-

Let us represent 3x3 board as nine elements vector. Each element in a vector can contain any of the following three digits:

- 0 - representing blank position
- 1 - indicating x player move
- 2 - indicating 0 player move

It is assumed that this program makes use of a move table, that consist of vector of $3^9$ (19683) elements.

| Index | Current Board Position | New Board Position |
|---|---|---|
| 0 | 000000000 | 000010000 |
| 1 | 000000001 | 002000001 |
| 2 | 000000002 | 000100002 |
| 3 | 000000010 | 002000010 |
| | ⋮ | |

The entries of the table are carefully designed manually in advance keeping in mind that the computer should never lose. Each entry is indexed by decimal representation of current board position digits. Initially the board is empty and is represented by nine zeros. The best new board position (000010000) is obtained by putting 1 (representing move by x player) in the fifth cell. All possible board positions are stored in Current Board Position column along with its corresponding next best possible board position in New Board Position column. Once the table is designed, the computer program has to simply do the table lookup.

Algorithm:

step 1: View the vector (board) as a ternary number.

step 2: Get an index by converting this vector to its corresponding decimal number.

step 3: Get the vector from New Board Position stored at the index. The vector thus selected represents the way the board will look after the move that should be made.

step 4: So set board position equal to that vector.

Disadvantages: This version of the program is very efficient in terms of time but has several disadvantages.

a) It requires lot of memory space to store the move table.

b) To specify entries in move table manually, lot of work is required.

c) Creating move table is highly error prone as data to be entered is voluminous.

d) This approach cannot be extended to 3D tic-tac-toe. In this case, $3^{27}$ board position are to be stored.

e) This program is not intelligent at all as it does not meet any of AI requirements.

Approach II :-

The board B[1..9] is represented by a nine-element vector. There are in all 9 moves represented by a integer 1 (first move) to 9 (last move), we will use the following three sub procedures.

- Go(n) - Using this function computer can make a move in square n.
- Make_2 - This function helps the computer to make valid 2 moves.
- PossWin(P) - If player P can win in the next move then it returns the index (from 1 to 9) of the square that constitutes a winning move, otherwise it returns 0.

The strategy applied by human for this game is that if human is winning in the next move then he/she plays in the desired square, else if human is not winning in the next move then one checks if the opponent is winning. If so then block that square, otherwise try making valid two in any row, column, or diagonal.

Let us represent computer by C and human by H. The player who is playing first will be called X player. Since computer can be first or second player, the following table consists of rules to be applied by the computer for all 9 moves.

| (C plays X, H plays O) | (H plays X, C plays O) |
|---|---|
| 1 move : Go(5) / Go(1) | 2 move : If B[5] is blank, then Go(5) else Go(1) |
| 3 move : If B[9] is blank, then Go(9) else {make 2} Go(3) | 4 move : {By now human (playing X) has played 2 moves} : If PossWin(X) then {block X} Go(PossWin(X)) else {make 2} Go(Make_2) |
| 5 move : {By now both have played 2 moves} : If PossWin(X) then {X wins} Go(PossWin(X)) else if PossWin(O) {block O} then Go(PossWin(O)) else if B[7] is blank then Go(7) else Go(3) | 6 move : {By now computer has played 2 moves} : If PossWin(O) then {O wins} Go(PossWin(O)) else if PossWin(X) {block X} then Go(PossWin(X)) else Go(Make_2) |
| 7 & 9 moves : {By now human (playing O) has played 3 chances} : If PossWin(X) then {X wins} Go(PossWin(X)) else {block O} if PossWin(O) then Go(PossWin(O)) else Go(Anywhere) | 8 move : {By now computer has played 3 chances} : If PossWin(O) then {O wins} Go(PossWin(O)) else {block O} if PossWin(X) then Go(PossWin(X)) else Go(Anywhere) |

Advantage:

This version of the program is memory efficient and easier to understand as complete strategy has been determined in advance but has several disadvantages. It applies heuristics, so can be treated as one step towards AI approach.

Disadvantages:

a) Not as efficient as first one with respect to time. Several conditions are checked before each moue.

b) still cannot generalize to 3-D.

Approach iii :-

In this approach, we choose board position to be a magic square of order 3; blocks numbered by magic number. The magic squa of order n consists of $n^2$ distinct numbers (from 1 to $n^2$), such that the numbers in all rows, all columns, and both diagonal sum to the same constant. It is generated using the followi method if n is odd. There might be various other methods fo generating magic square. The one we have used is defined belo

It begins by placing 1 in middle of top row, then incremente placing subsequent numbers in the square diagonally up and rig If a filled square is encountered, move vertically down one square instead, then continue as before. The counting is wrapped around, so that falling off the top returns on the bottom and falling off the right returns on the left. One can easily show that the sum must be $n[(n^2+1)/2]$ for each row, column, and diagonal.

| | | |
|---|---|---|
| 8 | 1 | 6 |
| 3 | 5 | 7 |
| 4 | 9 | 2 |

In this approach, we maintain a list of the blocks played by each player. For the sake of convenience, each block is identified by its number. The following strategy for possible win for a player is used. Obviously in our case, we are considering a computer to be a player for which the strategy is suggested as follows.

1. Each pair of blocks a player owns is considered.
2. Difference D between 15 and the sum of the two blocks is compu
   If D<0 or D>9, then these two blocks are not collinear an so can be ignored; otherwise if the block representing difference is blank then player can move in that block.
3. This strategy will produce a possible win for a player.

Let us see the execution of the program from the start and the strategy used by a computer. Assume that human is the first player (H) and computer is the second player (C).

Turn 1: Suppose H plays in the eighth block.

Turn 2: C plays in fifth block (fixed move)

Turn 3: H plays in first block.

Turn 4: C checks if H can win or not

Compute sum of blocks played by H

$s = 8+1 = 9$

Compute $D = 15-9 = 6$

The sixth block is a winning block for H, there are on either list so C blocks it with 6th block. The 6th block is recorded in the lis

Turn 5: H plays in fourth block.

Turn 6: C checks if C can win as follows:

Compute sum of blocks played by C

$S = 5 + 6 = 11$

Compute $D = 15 - 11 = 4$; Discard this block as it already exists in x list

Now C checks whether H can win

Compute sum of pair of square from list of H which have not been used earlier

$S = 8 + 4 = 12$

Compute $D = 15 - 12 = 3$

Block 3 is free, so C plays in this block. The third block is recorded in the list of computer.

Turn 7: If H plays in second or ninth block, then computer wins.

Let us assume that H plays in second block.

Turn 8: C checks if it can win as follows:

Compute sum of blocks played by C which has not been used earlier.

$S = 5 + 3 = 8$;

Compute $D = 15 - 8 = 7$

Block 7 is free, so C plays in 7th blocks & wins the game.

If H plays in seventh block at its Turn 7, then there is a draw.

status of both lists after 7th move

| 8 | 1 | 4 | 2 | | |
|---|---|---|---|---|---|

player X (Human)

| 5 | 6 | 3 | | | |
|---|---|---|---|---|---|

player O (computer)

## Advantages:

a) This approach to be extended to handle 3D Tic-Tac-Toe

b) It could also be extended to handle games more complicated than Tic-Tac-Toe.

## Drawbacks:

This program will require much more time than other two. As it must search every representing all possible new sequences before making each move.

## Three Dimensional Tic-Tac Toe:-

This game is similar to the traditional game of Tic-Tac-Toe but is played on cube of order 3. Cube is created by stacking three grids each square of order 3. To win a player must place 3 of their symbols on three square that line up vertically, horizontally or diagonally on a single grid, or spread evenly overall 3 grids. This game uses magic cube.

| | | |
|---|---|---|
| 8 | 24 | 10 |
| 12 | 7 | 23 |
| 22 | 11 | 9 |

Grid 1

| | | |
|---|---|---|
| 15 | 1 | 26 |
| 25 | 14 | 3 |
| 2 | 27 | 13 |

Grid 2

| | | |
|---|---|---|
| 19 | 17 | 6 |
| 5 | 21 | 16 |
| 18 | 4 | 20 |

Grid 3

The magic cube is 3 dimensional equivalent of magic squar i.e., numbered from 1 to 27. The magic cube has a magic constant which is equal to $\frac{n(n^3+1)}{2}$ where 'n' is order of cub For order 3 cube the sum of numbers on each row, each column on 6 outer surfaces of a cube, each rows, each columns and tw diagonals of middle grid and the four main space diagonals is equa to a single number that is 42 in this case.

## Development of AI languages:

AI languages have traditionally been those which stress on knowledge representation schemes; pattern matching, flexible search, and programs as data. The typical examples of such langu are LISP, Pop-2, ML, and Prolog. LISP is a functional languag based on lambda calculus and Prolog is a logic language based on first-order predicate logic. Both languages are declara languages where one is concerned about 'what to compute', and not 'how to compute'. Pop-2 is a stack-based languag e providing greater flexibility and has some similarity to LISP.

Pop 11 is an extension of pop-2 is embedded in an AI program environment called poplog permits mixed use of logic and functional languages like PROLOG, LISP and ML. Other hybrid programming languages exists which focus on the domain level as opposed to implementation level. For example, the KLONE of languages support modelling at the knowledge level.

Environment such as KEE, ART and CLIPS provides a variety of knowledge representation schemes. AI programming can exploit any language from BASIC through c to smalltalk.

## Current Trends in AI:

Conventional computing (hard computing) is based on the concept of precise modelling and analyzing to yield accurate results. Hard computing techniques work well for simple problems, but is bound by the NP-Complete set which include problems, often occurring in biology, medicine, humanities, management sciences, and similar fields. Such problems remain intractable to conventional mathematical and analytical methods as these problems are so large that it would take lifetime of the Universe to solve them, even at super computing speeds. ~~Soft~~ computing which is a complementary field to hard computing and is a foundation component for emerging field of conceptual intelligence.

Soft computing techniques resembles biological process more closely than traditional techniques. These are largely based on formal logical systems.

Soft computing represents a significant paradigm shift in the aim of computing. A shift which reflects the fact the human mind unlike the present day computers posses a remarkable ability to store and process information which is permissively impresized, uncertain and unstructured or uncategorised. It started in the early 1990's refers to a collection of computational techniques in computer science, machine learning and some engineering disciplines which

study model and analyse very complex phenomenon, those for which more conventional measures have tho not yield low cost, analytic and complex solution. Components of soft computing include neural networks, fuzzy systems, evoultionary algorithms, swarm intelligence, etc.

Neural networks have been developed based on functioning of the human brain. Attempts to model the biological neuron have led to the development of the field called artificial neural network. These systems have been developed in order to facilitate predicting features in advance based on the previous details available.

Evolutionary techniques mostly involve meta-heuristic optimization algorithms such as evolutionary algorithms (comprising genetic algorithms, evolutionary programming, etc.) and swarm intelligence (comprising ant colony optimization and praticle swarm optimization).

Genetic algorithms based on Darwin theory of evolution were developed mainly by emulating the nature and behaviour of biological chromosome. Genetic algorithms are favoured for search problems which require the identification of a global optimal solution without getting stuck in local minima or maxima.

Swarm intelligence is a type of artificial intelligence based on the collective behaviour of decentralized, self-organized systems. Social insects like ants, bees, wasps, and termites perform their simple tasks independent of other members of the colony. However, they are able to solve complex problems emerging in their daily lives by mutual cooperation. This

emergent behaviour of self-organization by a group of social insects is known as swarm intelligence.

Ant colony algorithm was developed to emulate the behaviou of real ants and found suitable applications in multi-agent system to solve difficult combinatorial optimization problems Behaviour of ants has applications in the discrete optimizatio problems. An ant algorithm is one in which a set of artificial ants (agents) cooperate to find the solution of a problem by exchanging information via pheromone deposited on graph edges.

Advantages of swarm intelligence systems are adaptability (self-organizing), robustness (ability to find a new solution if the current solution becomes invalid), reliability (agents can be added or removed without disturbing behaviour of the total system because of the distributed nature), simplicity and has no central control.