SONI
Date 09/3/22
Page

Artificial Intelligence -

Artificial Intelligence suggest that machines can mimic humans in -

- Talking
- Thinking
- Learning
- Planning
- Understanding

→ Artificial Intelligence is also called Machine Intelligence and Computer Intelligence.

→ Artificial intelligence is composed of 02 words "Artificial" and "Intelligence" where Artificial defines "man made" and intelligence defines "thinking power", hence AI means "a man-made thinking power."

→ It is a branch of computer science by which we can create intelligent machines which can behave like a human, think like humans, and able to make decisions." and also work while trying to solve a problem.

→ AI is a way of making a computer, a computer-controlled robot, or software think intelligently, in the similar manner the intelligent humans think.

→ John McCarthy widely recognized as the "Father of AI".

→ In the Mid 1950s McCarthy coined the term "AI" and defined it as "the science of making intelligent machines".

* Intelligence – The ability of a system to calculate, reason, perceive relationships and analogies, learn from ~~exper~~ memory, solve problems, comprehend complex ideas, use natural language fluently, classify, generalize and adapt new situations.

* Artificial Intelligence examples –

- Self Driving Cars
- E-payment
- Google Maps
- Text AutoCorrect
- Automated Translation
- Social Media
- Face Detection
- Search Algorithm
- Robots
- Automated Investment
- NLP - Natural Language Processing
- Flying Drones
- Apple Siri
- Amazon Alexa

* Advantages of Artificial Intelligence

(i) High Accuracy with less errors - AI machines or systems are prone to less errors and high accuracy as it takes dec decisions as per pre-experience or information.

(ii). High-Speed - AI systems can be of very high speed and fast-decision making because of that AI systems can beat a chess champion in the chess game.

(iii). High reliability - AI machines are highly reliable and can perform the same action multiple times with high accuracy.

(iv). Useful for risky areas - AI machines can be helpful in situations such as defusing a bomb, exploring the ocean floor, where to employ a human can be risky.

(v). Digital Assistant - AI can be very useful to provide digital assistant to the users such as AI technology is current used by various

E-commerce websites to show the products as per customer requirement.

- Useful as a public utility -

AI can be very useful for public utilities such as a self driving car which can make our journey safer and hassle-free, facial recognition for security purpose, Natural language processing to communicate with the human in human-language, etc.

* Disadvantages of Artificial Intelligence

Even technology has some disadvantages and the same goes for Artificial Intelligence. Following are the disadvantages of AI.

(i) High Cost - The hardware and software requirements of AI is very costly as it requires lots of maintenance to meet current world requirements.

(ii). Can't think out of the box - Even we are making smarter machines with AI. but still they cannot work out of the box, as the robot will only do that work for which they are trained or programmed.

(iii). No feelings and emotions - AI machines can be outstanding performer but still does it does not have the feeling so it cannot make any kind of emotional attachement with human and may sometime be harmful for users if the proper care is not taken.

(iv). Increase dependency on machine - With the increment of technology, people are getting more dependent on devices and hence they are losing their mental capabilities.

(v). No original Creativity - As humans are so creative and image some new ideas but still AI machines cannot beat this power of human intelligence and cannot be creative and ~~imagination~~ imaginative.

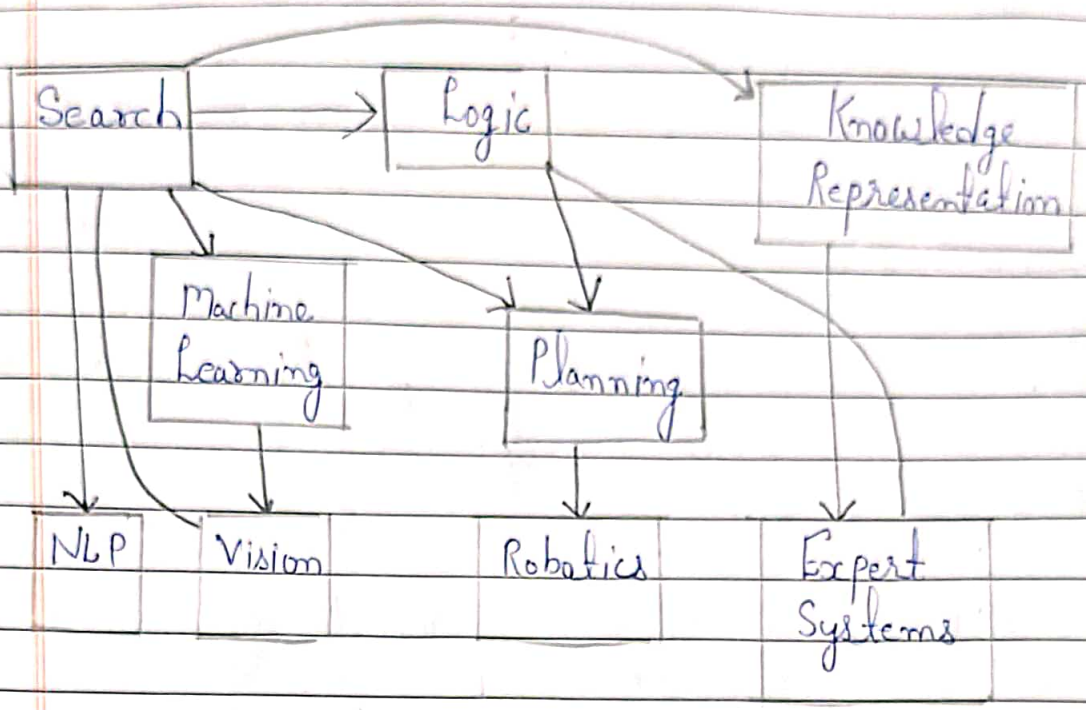
Q. What is difference between ai and natural intelligence ?

Features	AI	Human Intelligence
(i) Emergence	AI is an advancement made by human insights.	On the other hand human creatures are made with the intrinsic capacity to think, reason, review, etc.
(ii) Pace/Rate of AI and human	As compared to people, computers can handle more data at a speedier.	In terms of speed human cannot beat the speed of AI or machines.
(iii) Decision Making	AI is profoundly objective in choice making because it analyzes based on absolutely accumulated data.	Human's choices may be affected by subjective components which are not based on figures alone.
(iv) Perfection	AI frequently produces precise comes about because it ~~capabap~~ capacities based on a set of modified rules.	For human insights, there's more often than not a room for "human error" as certain subtle elements may be

SONI
Date
Page 7

Features	AI	Human Intelligence
		missed at one point or the other.
(iv) ~~Decision Making~~	~~AI is~~	
(v) Energy Consumption	Modern computer generally uses 2 watts energy.	Human brains uses about 25 watts.
(v) Modification of AI & Human	AI takes much more time to adjust to unused changes.	Human insights can be adaptable in reaction to the changes to its environment. This makes individuals able to memorize and ace different skills.
(vi) Versatility	AI can as it were perform less assignments at the same time as a framework can as it were learn duties one at a time.	The human judgement skills underpins multitasking as prove by ~~different~~ differing and concurrent roles.
(vii) Social Networking	AI has not aced the capacity to choose up on related social and enthusiastic cues.	On the other hand, as social creatures people are much way better at social interaction Since they can prepare
and	are delicate to other feelings	theoretical data, have self-awareness.

* Areas of AI and Some Dependencies -



* System that act like humans: Turing Test

→ "The art of creating machines that perform functions that require intelligence when performed by people."
(Kurzweil)

→ "The study of how to make computers do things at which, at the moment, people are better".
(Rich and Knight)

~~• You enter a~~

- Turing machine in AI

• You enter a room which has a computer terminal. You have a fixed period of time to type what you want into the terminal, and study the replies. At the other end of the line is either a human being or a computer system.

• If it is a computer system, and at the end of the period you cannot reliably determine whether it is a system or a human, then the system is deemed to be intelligent.

* → The Turing Test approach

→ a human questioner cannot tell if

* there is a computer or a human answering his question, via teletype (remote communication)

• The computer must behave intelligently

→ Intelligent behaviour

• to achieve human-level performance in all cognitive tasks

→ These cognitive tasks include-

• Natural Language Processing (NLP)

→ NLP is required to communicate with Interrogator in general human language. like english.

• Knowledge representation - To store and retrieve information during the test.

• Automated reasoning - To use the previously stored information for answering the questions.

- Machine learning — To adapt new changes and can detect generalized patterns.
- Vision (For total Turing test) — To recognize the interrogator actions and other objects during a test.
- Motor Control (For total Turing test) — To act upon objects if requested.

* **Philosphy of AI** – While exploiting the powers of the computer systems, the curiosity of human, lead him to wonder.

"Can a machine think and behave like humans do?"

Thus, the development of AI started with the intention of creating similar intelligence in machines that we find and regard high in humans

* **Goals of AI**

To Create Expert Systems: The systems which exhibit intelligent behaviours behaviors, learn, demonstrate, explain and advice its users.

To implement Human Intelligence in Machines:

Creating systems that Understand, think, learn and behave like humans.

* AI Terminologies

(1) **Agent** = Agents are systems or software programs capable of autonomous, purposeful and reasoning directed towards one or more goals.

→ They are also called assistants, brokers, bots, droids, intelligent agents and software agents.

(2) **Autonomous Robot** = Robot free from external control or influence and able independently.

(3) **Backward Chaining** = Strategy of working backward for Reason / Cause of a problem.

(4) **Blackboard** = It is the memory inside computer, which is used for communication between the cooperative expert systems.

(5) **Environment** = It is a part of real or computational world inhabited by the agent.

(6) Heuristics - It is the knowledge based on Trial-and-error, evaluations and experimentation.

(7). Knowledge Engineering - Acquiring knowledge from human experts and other resources.

(8). Percepts - It is the format in which the agent obtains information about the environment.

(9). Pruning - Overriding unnecessary and irrelevant considerations ~~and~~ in AI.

(10) Rule - It is a format of representing knowledge base in Expert System. It is in the form of IF-Then-Else.

(11) Shell - A shell is a software that helps in designing interference engine, knowledge base and user interface of an expert system.

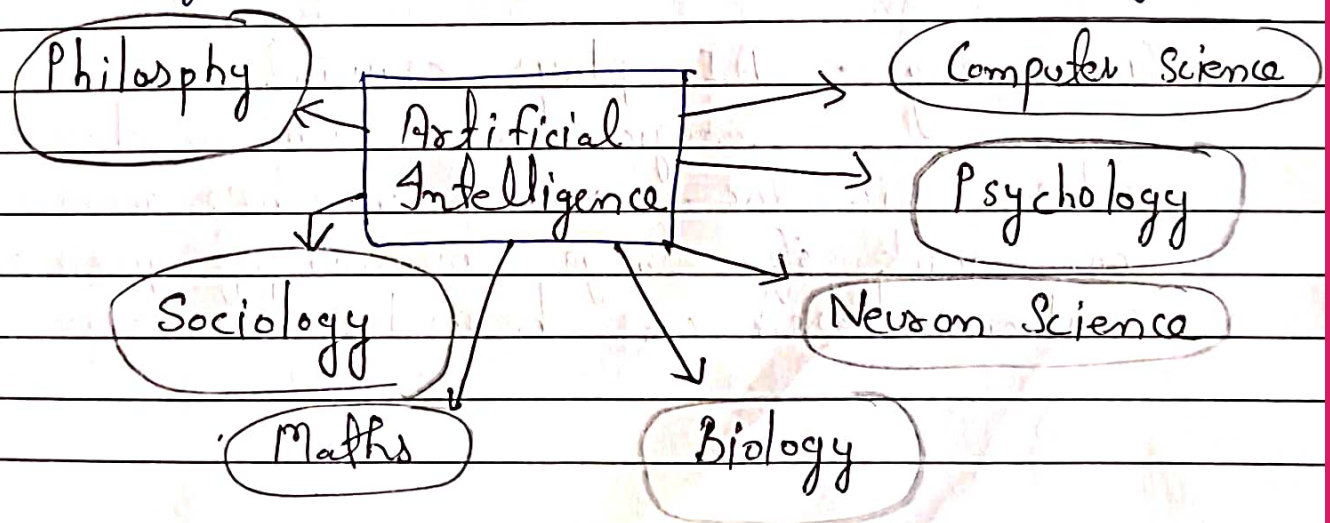
(12). Task - It is the goal the agent is tries to accomplish.

(13). Turing Test - A test developed by "Allan Turing" to test the ~~intelligent~~ intelligence of a machine as compared to human intelligence.

8. What contributes to AI?

Artificial intelligence is a science and techology based on disciplines such as Computer Science, Biology, Psychology, Linguistics, Mathematics and Engineering.

→ A major thrust of AI is in the development of computer functions associated with human intelligence, such as reasoning, learning and problem solving.



* **AI Technique** – AI Technique is a manner to organize and use the knowledge efficiently in such a way that :-

→ It should be perceivable by the people who provide it.

→ It should be easily modifiable to correct errors.

→ It should be useful in many situations though it is incomplete or inaccurate.

AI techniques elevate the speed of execution of the complex program it is equipped with.

* **Applications of AI**

(i). **Gaming** – AI plays crucial role in stategic games such as chess, poker, tic-tac-toe etc., where machine can think of large number of possible positions based on heuristic knowledge.

(ii) Natural Language Processing (NLP) – It is possible to interact with the computer that understands natural language spoken by humans.

(iii) Expert Systems – There are some applications which integrate machine, software and special information to impart reasoning and advising. They provide explanation and advice to the users.

(iv) Vision Systems – These systems understand, interpret and comprehend visual input on the computer.

For eg –

(a). A spying aeroplane takes photographs which are used to figure out ~~spt~~ spatial information or map of the areas.

(b) Doctors use clinical expert system to diagnose the patient.

(c). Police use computer software that can recognize the face of criminal with the stored portrait made by forensic artist.

(v). Speech Recognition - Some intelligent systems are capable of hearing and comprehending the language in terms of sentences and their meanings while a human talks to it. It can handle different accents slang words, noise in the background, change in human's noise due to cold etc.

(vi). Handwriting Recognition - The handwriting recognition software reads the text written on paper by a pen or on screen by a stylus. It can recognize the shapes of the letters and convert it into editable text.

(vii) Intelligent Robots - Robots are able to perform the tasks given by a human. They have sensors to detect physical data from the real world such as light, heat, movement, temperature, sound etc.

→ They have efficient processors, multiple sensors and huge memory, to exhibit intelligence.
→ In addition, they are capable of learning from their mistakes and they can adapt to the new environment.

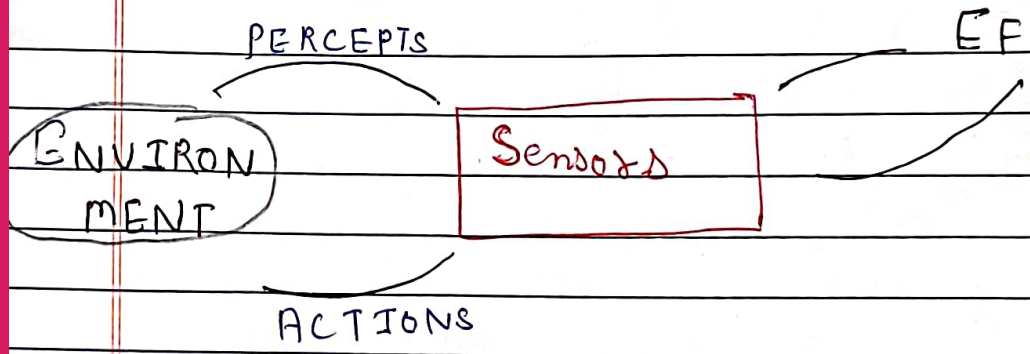
* Intelligence - The ability of a system to calculate, reason, perceive relationships and analogies, learn from experience, store and retrieve information from memory, solve problems, comprehend comprehend complex ideas, use natural language fluently, classify, generalize and adapt new situations.

* Agents – An agent is anything that can perceive its environment through sensors and acts upon that environment through efforts effectors.

→ A human agent has sensory organs such as eyes, ears, nose, tongue and skin parallel to the sensors and other organs such as hands, legs, mouth for effectors.

→ A robotic agent replaces cameras and infrared range finders for the sensors and various motors and actuators for effectors.

→ A software agent has encoded bit strings as its programs and actions.



* Agents Terminology

(i) Performance Measure of Agent - It is the criteria, which determines how successful an agent is.

(ii) Behaviour of Agent - It is the action that agent performs after any given sequence of percepts.

(iii) Percepts - It is agent's perceptual inputs as at a given instance.

(iv) Percept Sequence - It is the history of all that an agent has perceived till date.

(v) Agent Function - It is a map from the per precept sequence to an action.

* Rationality - Rationality is concerned with expected actions and results depending upon what the agent has perceived. Performing actions with the aim of obtaining useful information is an important part of rationality.

* Ideal Rational Agent - An ideal rational agent is the one, which is capable of doing expected actions to maximize its performance measure, on the basis of -

→ Its percept sequence.
→ Its built in knowledge base.

*→ Rationality of an agent depends on the following 04 factors -

→ The performance measures, which determine the degree of success.

→ Agent's Percept Sequence till now.

→ The agent's prior knowledge about the environment.

→ The actions that the agent can carry out.

*→ A rational agent always performs right action where the right action means the action that causes the agent to be most successful in the given percept sequence.

The problem the agent solve is characterized, by Performance Measure, Environment, Actuators and Sensors (PEAS).

Q1. What is Agent ? What are different types of Agents ?

→ An agent can be anything that perceive its environment through actuators.
An Agent runs in the cycle of perceiving, thinking and acting.
An Agent can be :-

- Human-Agent - A human agent has eyes, ears and other organs which work for sensors and hand, legs, vocal tract work for actuators.

- Robotic Agent - A robotic agent can have cameras, infrared range finder, NLP for sensors and various motors for actuators.

- Software Agent - Software agent can have keystrokes file contents as sensory input and act on those inputs and display output on the screen.

Q2) What is an Intelligent agent?

An intelligent agent (IA) is an entity that makes a decision, that enables artificial intelligence to be put into action. It can also be described as a software entity that conducts operations in the place of users or programs after sensing the environment.

→ It uses actuators to initiate action in that environment.

Q3) Discuss the ~~structure of intelligent agent~~ environment & behaviour of IA with diagram.

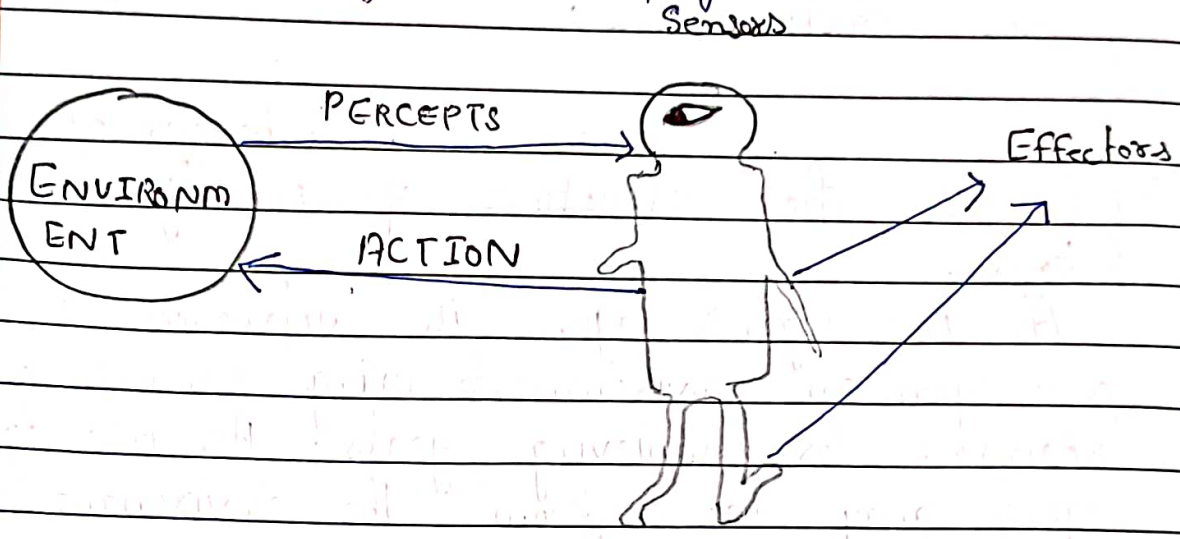
An intelligent agent is autonomous entity which act upon an environment using sensors and actuators for achieving goals. An intelligent agent may learn from the environment to achieve their goals.

Eg - A thermostat is an example of an intelligent agent.

- Sensor — Sensor is a device which detects the change in the environment and sends the information to other electronic devices. An agent observes its environment through sensors.

- Actuators - Actuators are the component of machines that converts energy into motion. The actuators are only responsible for moving and controlling a system.
An actuators can be an electric motor, gears, rails etc.

- Effectors - Effectors are the devices which affect the environment.
Effectors can be legs, wheels, arms, fingers, wings and display screen.



Following are the main 04 rules for an AI agent:-

Rule 1 - An AI agent must have the ability to perceive the envisonment.

Rule 2 - The observation must be used to make decisions.

Rule 3 - Decision should result in an action.

Rule 4 - The action taken by an AI agent must be a rational action.

Q(D). Discuss the structure of Intelligent agent.

The intelligent Agent structure is the combination of Agent Function, Architecture and Agent Program.

→ Agent = Architecture + Agent Program

The 03 entities are described below-

(1). Architecture- Architecture is the machinery on which the agent executes its action. It is essentially a device with embedded actuators and sensors.

eg - Autonomous cars which have various motion and GPS sensors attached to them and actuators based on the inputs, aids in actual driving.

(2). Agent Function - Agent Function helps in mapping all the information it has gathered from the environment into action.

(3). Agent Program - The Agent Program performs the execution of the Agent Function. The execution happens on top of Agent Architecture and produces the desired function.

Q5) What is environment of intelligent agent? Discuss the properties of environment of intelligent agent.

An environment in artificial intelligence is the surrounding of the agent. The agent takes input from the ~~enviroa~~ environment through sensors and delivers the output to the environment through actuators.

The environment has multifold properties-

(i) Discrete / Continuous - If there are a limited number of distinct, clearly defined, states of the environment, the environment is discrete (for eg - chess); otherwise it is continuous (for eg. driving).

(ii) Observable / Partially Observable - If it is possible to determine the complete state of the environment at each time point from the percepts it is observable; otherwise it is only partially observable.

(iii) Static / Dynamic - If the environment does not change while an agent is acting, then it is static; otherwise it is dynamic.

(iv) Single agent / Multiple agent - The environment may contain other agents which may be of the same or different kind as that of the agent.

(iv) Accessible / Inaccessible – If the agent's sensory apparatus can have access to the complete state of the environment, then the environment is accessible to that agent.

(v) Deterministic / Non-deterministic – If the next state of the environment is completely determined by the current state and the actions of the agent, then the environment is determ. deterministic; otherwise it is non-deterministic.

(vi) Episodic / Non-episodic – In an episodic environment, each episode consists of the agent perceiving and then acting. The quality of its action depends just on the episode itself. Subsequent episodes do not depend on the actions in the previous episodes. Episodic environments are much simpler because the agent does not need to think ahead.

(6).

* Problem in AI - To call a problem AI complete reflects an attitude that it would not be solved by a simple specific algorithm.

AI complete problems are hypothesised to include computer vision, natural language understanding and dealing with unexpected circumstances while solving any real-world problem.

X. Problem Characteristics

(i) Is the problem decomposable?
(ii) Can solution steps be ignored or undone?
(iii) Is the universe predictable?
(iv) Is a good solution absolute or relative?
(v) Is the solution a state or a path?
(vi) What is the role of knowledge?
(vii) Does the task require interaction with a person?

* Local Search Algorithms –

They start from a prospective solution and they move to a neighboring solution. They can return a valid solution even if it is interrupted at any time before they end.

• Hill-Climbing Search

It is an iterative algorithm that starts with an arbitrary solution to a problem and attempts to find a better solution by changing a single element of the solution incrementally.

→ If the change produces a better solution, an incremental change is taken as a new solution. This process is repeated until there are no further improvements.

Function Hill-Climbing (problem), return a state, that is a local maximum.

Disadvantage – This algorithm is neither complete, nor optimal.

Local Beam Search -

In this algorithm, it holds k numbers of states at any given time. At the start, these states are generated randomly.

The successors of these k states are computed with the help of objective function. If any of these successors is the maximum value of the objective function, then the algorithm stops.

Otherwise the Cinitial k states and k number of successors of ~~these~~ states = 2k) states are placed in pool. The pool is then sorted numerically. The highest k states are selected as new initial states. This process continues until a maximum value is reached.

* Simulated Annealing =

Annealing is the process of heating and cooling a metal to change its internal structure for modifying its physical properties. When the metal cools, its new structure is seized, and the metal retains its newly obtained properties. In simulated annealing process, the temperature is kept variable.

→ We initially set the temperature high and then allow it to 'cool' slowly as the algorithm proceeds. When the temperature is high, the algorithm is allowed to accept worse solutions with high frequency.

* Travelling Salesman Problem

In this algorithm, the objective is to find a low-cost tour that starts from a city, visits all cities en-route exactly once and end at the same starting city.

* Uniformed search-

→ Also called blind, exhaustive or brute force search, uses no information about the problem to guide the search and therefore may not be very efficient.

* Informed Search -

→ Also called heuristic or intelligent search, uses information about the problem to guide the search, usually guesses the distance to a goal state and therefore efficient, but the search may not be always possible.

* BRUTE - FORCE SEARCH STRATEGIES

There are most simple, as they do not need any domain-specific knowledge.

→ They work fine with small number of possible states.

Requirements -

→ State description
→ A set of valid operator
→ Initial state
→ Goal state description

(1). Breadth - First Search

It starts from the root node, explores the neighboring nodes first and moves towards the next level neighbors.

It generates one tree at a time until the solution is found. It can be implemented using FIFO queue data structure. This method provides shortest path to the solution.

If ~~branching factor~~

Algorithms

1) Create a variable called NODE-LIST and set it to initial state.

2) Until a goal state is found or NODE-LIST is empty do

(a). Remove the first element from NODE-LIST and call it E. If NODE-LIST was empty, quit

(b). For each way that each rule can match the state described in E do:

(i). Apply the rule to generate a new state.

(ii). If the new state is goal state, quit and return this state.

(iii). Otherwise, add the new state to the end of NODE-LIS

BREADTH FIRST SEARCH is

→ One of the simplest search strategies.
→ Complete. if there is a solution, BFS is guaranted to Find it.
→ If there are multiple solutions, then a minimal solution will be fou
→ The algorithm is optimal, if all operators have the same cost. Otherwise, BFS finds a solution with the shortest path length.

- Time complexity : $O(b^d)$
- Space complexity : $O(b^d)$
- Optimality : Yes

b - branching factor
d - Depth of the shallowest goal node

Maximum length of any path (m) in search space

If branching factor (average no. of child nodes for a given node) = b and depth d, then number of nodes at level d = b^d.

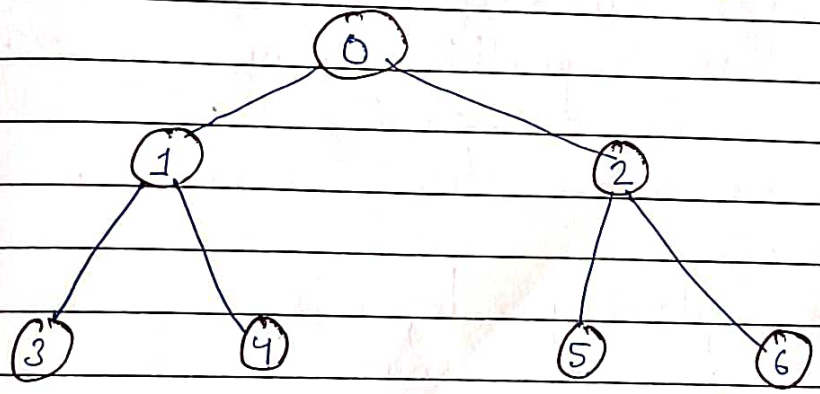
The total no. of nodes created in worst case is $b + b_2 + b_3 + \ldots + b^d$.

Disadvantage - Since each level of nodes is saved for creating next one, it consumes a lot of memory space. Space requirement to store nodes is exponential.

Advantage

Its complexity depends on the number of nodes. It can check duplicate nodes.

→ Find the path of minimal length to the goal.



* ## Depth - First Search

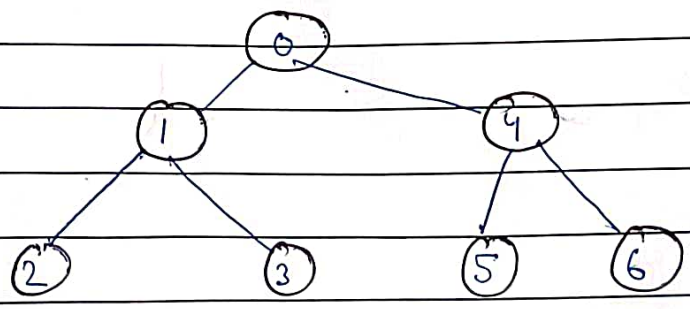
It is implemented in recursion with LIFO stack data structure. It creates the same yet set of nodes as Breadth - First method, only in the different order.

As the nodes on the single path are stored in each iteration from root to leaf node, the space requirement to store nodes in linear. With branching factor b and depth as m, the storage space is bm.

- **Disadvantage** — This algorithm may not terminate and go on infinitely on one path. The solution to this issue ~~to~~ is to choose a cut off depth. If the ideal cut-off is d, and if chosen cut-off is lesser than d, then this algorithm may fail.

If chosen cut-off is more than d, then execution time increases.

Its complexity depends on the number of paths. It cannot check duplicate nodes.



~~15~~ 10

Algorithm

(1). Create a variable called NODE-LIST and set it to initial state.

(2). Until a goal state is found or NODE-LIST is empty do

(a). Remove the first element from NODE-LIST and call it E. If NODE-LIST was empty, quit.

(b). For each way that each rule can match the state described in E do:

(i). Apply the rule to generate a new state.

(ii) If the new state is a goal state, quit and return this state.

(iii). Otherwise, add the new state in front of NODE-LIST.

DFS search is —

① The algorithm takes exponential time.

② If N is the maximum depth of a node in the search space, in the worst ~~case~~ case the algorithm will take time $O(b^d)$

③. The space taken is linear in the depth of the search tree $O(bN)$.

NOTE- The time taken by the algorithm is related to the maximum depth to the search tree. If the search tree has infinite depth, the algorithm may not terminate. This can happen if the search space is infinite. It can also happen if the search space contains cycles. The latter case can be handled by checking for cycles in the algorithm. Thus, Depth First Search is not complete.

Heuristic Search Technique –

To solve large problems with large number of possible states, problem-specific knowledge needs to be added to increase the efficiency of search algorithms.

They calculate the cost of optimal path between two states.

→ A heuristic function for slidingtiles games is computed by counting number of moves that each tile makes from its goal and adding these number of moves for all tiles.

- **Pure Heuristic Search** It expands nodes in the order of their heuristic values. It creates two lists, a closed list for the already expanded nodes and an open list for the creates but unexpanded nodes.

A* Search –

It is best-known form of Best First search. It avoids expanding paths that are already expensive, but expands most promising paths first.

$f(n) = g(n) + h(n)$

where,

- $g(n)$ the cost (so far) to reach the node.
- $h(n)$ estimated cost to get from the node to the goal.
- $f(n)$ estimated total cost of path through n to goal. It is implemented using priority queue by increasing $f(n)$.

→ A* is much more efficient that uninformed searc

* Greedy Best First Search

It expands the node that is estimated to be ~~closed~~ closest to goal. It expands nodes based on $f(n) = h(n)$.

It is implemented using priority queue.

- Disadvantage - It can get stuck in loops. It is not optimal.

~~option~~

* A* Algorithm

The A* search algorithm (pronounced as "A-star") is a tree search algorithm that finds a path from a given initial node to a given goal node (or one passing a given goal test). It employs a "heuristic estimate" which ranks each node by an estimate of the best route that goes through that node. It visits the nodes in order of this heuristic estimate.

Similar to greedy best-first search but is more accurate because A* takes into account the nodes that have already been traversed.

From A*, we note that $f = g + h$

Algorithm

(1). Initialize : Set OPEN = (s) ; CLOSED = ()
$g(s) = 0$, $f(s) = h(s)$

(2). Fail : If OPEN = (), Terminate and Fail.

(3). Select : Select the minimum cost state, n from OPEN, save in CLOSED.

(4). Terminate : If $n \in G$, Terminate with success and return $f(n)$.

(5). Expand : for each successor, m, of n

(a). If $m \in$ [OPEN $\cup$ CLOSED]

Set $g(m) = g(n) + c(n, m)$
Set $f(m) = g(m) + h(m)$

Insert m in OPEN

(b). If $m \in$ [OPEN $\cup$ CLOSED]

Set $g(m) = \min\{g(m), g(n) + c(n, m)\}$

Set $f(m) = g(m) + h(m)$
If $f(m)$ has decreased and $m \in$ CLOSED

Move m to OPEN

* Best First Search :-

→ A combination of depth first and breadth first searches.

→ Depth first search is good because a solution can be found without computing all nodes and breadth first is good because it does not get trapped in dead ends.

→ The best first search allows us to switch between ~~path~~ paths thus gaining the benifits of both approaches.

OPEN is a priority queue of nodes that have been evaluated by the heuristic function but which have not yet been expanded into successors. The most promising nodes are at the front.

CLOSED are nodes that have already been generated and these nodes must be stored because a graph is being used in preference to a tree.

Algorithm

(1). Start with OPEN holding the initial state.

(2). Until a goal is found or there are no nodes left on open do.

- Pick the best node on OPEN.
- Generate its successors.
- For each successor DO.

  - If it has not been generated before, evaluated it, add it to OPEN and record its parent.

  - If it has been generated before change the parent if this new path is better and in that case update the cost of getting to any successor nodes.

(3). If a goal is found or no more nodes left in OPEN, quit, else return to 2.

47

* Alpha-beta pruning algorithm -

- Pruning - eliminates a branch of the search tree from consideration without exhaustive examination of each node.

- α-β pruning - The basic idea is to prune positions of the search tree that cannot improve the utility value of the max or min node, by just considering the value of nodes seen so far.

→ Alpha-beta pruning is used on top of minimum search to detect paths that do not need to be explored. The ~~ins~~ intuition is:

- The ~~Max~~ MAX player, is always trying to maximize the score. Call this α.

- The MIN player is always trying to minimize the score. Call this β.

- Alpha cut off - Given a Max node n, cutoff the search below n (i.e., don't generate or examine any more of n's children) if alpha(n) >= beta(n)

(alpha increases and passes beta from below)

- Beta cutoff — Given a Min node n, cut off the search below n (i.e., don't generate or examine any more of n's children) if beta(n) <= alpha(n)
- Carry alpha and beta values down during search ~~Bumut~~ Pruning occurs whenever alpha >= beta

Algorithm :—

function ALPHA-BETA-SEARCH (state) returns an action

inputs : state, current state in game

v → MAX VALUE (state, $-\infty$, $+\infty$)

return the action in SUCCESSORS (state) with value v

* PROGRAMMING IN LOGIC (PROLOG)

→ Prolog is a logic programming language. It has important role in AI. Unlike many other programming languages, Prolog is intended primarily as a declarative programming language.

→ In prolog, logic is expressed as relations (called as Facts and Rules).

→ Core heart of prolog lies at the logic being applied.

→ Formulation or Computation is carried out by running a query over these relations.

Installation in Linux:

Open a terminal (Ctrl + Alt + T) and type:

```
sudo apt-get install swi-prolog
```

* Key Features=

(1) Unification – The basic idea is, can the given terms be made to represent the same structure.

(2) Backtracking – When a task fails, prolog traces backwards and tries to satisfy previous task.

(3) Recursion – Recursion is the basis for any search in program.

* Advantages–

(1) Easy to build database. Doesn't need a lot of programming effort.

(2) Pattern matching is easy. Search is recursion based.

(3) It has built in list handling. Makes it easier to play with any algorithm involving lists.

Disadvantages -

① LISP (another logic programming language) dominates over prolog with respect to I/O features.

② Sometimes input and output is not easy.

* Applications - Prolog is highly used in artificial intelligence (AI). Prolog is also used for pattern matching over natural language parse trees.

* Constraint Satisfaction Problems

→ It is the process of finding a solution through a set of constraints that impose conditions that the variables must satisfy.

→ A CSP is defined by a set of variable $x_1; x_2; \ldots; x_n$ and a set of constraints $C_1; C_2; \ldots; C_m$.

→ Each variable x_i has a nonempty domain D_i of possible values.

→ An assignment that does not violate any constraints is called a consistent or legal assignment.

CSP can be given an incremental ~~for~~ formulation as a standard search problem as follows-

- Initial state - The empty assignment
- Successor function - A value can be assigned to any unassigned variable, provided that it does not conflict with previously assigned variable.
- Goal Test - The current assignment is complete.
- Path cost - A constant cost for every step.

* Game Playing — Adversarial search, or a game tree search, is a technique for analyzing an adversarial game in order to try to determine who can win the game and what moves the players should make in order to win.

→ Adversarial search is one of the oldest topics in AI.

→ It is developed by Shannon in 1950 and independently by Turing in 1951.

- 2-Person Games :-

- Players - We call them Max & min.
- Initial state - Includes board position and whose turn it is.
- Operators - Those correspond to legal moves
- Terminal Test -

Utility Function -

* MiniMax Algorithm :-

(1) Generate the whole game tree.

(2) Apply the utility function to leaf nodes to get their values.

(3). Use the utility of nodes at level n to derive the utility of nodes at level n-1.

(4) Continue backing up values towards the root (one layer at a time).

(5) Eventually the backed up values reach the top of the tree, at which point Max chooses the move that yields the highest value. This is called the minimax decision because it maximises the utility for Max on the assumption that Min will play perfectly to minimise it.

Algorithm MiniMax (Depth-First Version)
To determine the minimax value $V(J)$ do the following-

(1). If J is terminal, return $V(J) = e(J)$; otherwise.
(2). Generate J's successors $J_1, J_2, \ldots, J_b$.
(3). Evaluate $V(J_1), V(J_2), \ldots, V(J_b)$ from left to right.
(4). If J is a Max node, return $V(J) = \max[V(J_1) \ldots V(J_b)]$
(5) If J is a MIN node, return $V(J) = \min[V(J_1) \ldots, V(J_b)]$

* Limitations =

→ Not always feasible to traverse entire tree.

→ Time limitations.

KNOWLEDGE REPRESENTATION

* Knowledge — The data is collection of facts. The information is organised as data and facts about the task domain.

→ Data, information, and past experience combined together are termed as knowledge.

* Knowledge representation — It is a method to organise and formalise the knowledge in the knowledge base. It is in the form of IF-THEN-ELSE rules.

• Knowledge Based Agent — A knowledge based agent needs a KB and an inference mechanism. It operates by storing sentences in its knowledge base, inferring new sentences with the interference mechanism and using them to deduce which action to take....

KNOWLEDGE REPRESENTATION

* Knowledge — The data is collection of facts. The information is organised as data and facts about the task domain.

→ Data, information, and past experience combined together are termed as knowledge.

* Knowledge representation — It is a method to organise and formalise the knowledge in the knowledge base. It is in the form of IF-THEN-ELSE rules.

• Knowledge Based Agent — A knowledge based agent needs a KB and an inference mechanism. It operates by storing sentences in its knowledge base, inferring new sentences with the interference mechanism and using them to deduce which action to take....

* Mean End Analysis

It is a problem solving technique that identifies the current state, defines the end goal and determine the action plan to reach the end state in a modular way.

→ End goals are split into sub-goals and sub-sub goals and then action plans and drawn to achieve sub-goals first and then move towards achieving the main goal progressively.

→ Most of the problem-solving strategies will have either forward action or backward action.

Uses of Mean End Analysis (MEA) —

(1) MEA is a creative problem-solving techniques used in AI application for a quite long time.
from the search space of possible solutions available in the system. AI selects the best possible solution by applying the right search strategy a algorithm.

- In the general Management area, MEA facilitates organisation planning to attain the goals.

→ MEA helps in implementing Business transformation projects by identifying as it is a state, defining to state and listening the new business processes to be developed.

→ In personal life also one can follow MEA methodology to solve problems or attain a specific goal.

(60)

First Order Predicate Logic (FOPL) -

→ The first order predicate logic (FOPL) is backbone of AI, as well a method of formal representation of Natural language (NL) text.

→ The prolog language for AI programming has its fandations in FOPL.

→ First order logic, like all other systems of formal logic is a method of formalizing natural languages into a computable format.

→ This, in turn, allow us to treat problems impressed through linguistic sentences in a formal manner.

From the formalization of natural language we then derive the capacity to formulate and prove theorems which in turn makes it possible to conduct inferential reasoning in disciplines such as mathematics, computer science and philosphy.

→ Propositional logic doesn't allow the conduct of reasoning over variables and functions having general and mutable content, which means that its capacity for abstraction is limited.

→ This means also that the first logical computing systems couldn't solve problems whose solutions lies in vector spaces of which the propositional space is a subspace.

→ This problem was overcome thanks to the development of a formal logical system, first-order logic which includes variables and thus allows for abstraction.

Automatic Reasoning and Inference—

→ Today, first order reasoning is a fundamental component of symbolic reasoning for machine learning systems.

→ Modern expert systems all use first or higher order logic, which allows the conduct of abstract reasoning and inference in an automated manner.

→ There are also specialized programming languages for first order logic.

→ The most famous of these in Prolog, where the acronym stands for "propositional logic" in a rather confusing manner but whose syntax corresponds to a major extent to that of first order logic.

→ Prolog is characterized by the rapidity with which knowledge bases can be built and requires little training by human analysts in order for them to be able to encode their knowledge.

* Difference b/w Propositional and first order Logic —

→ The first difference between the two relates to the fact that first order logic includes propositional logic, but the opposite isn't true.

This means that all problems expressed in propositional logic can be treated under first order logic but all problems in first order logic cannot be treated in propositional logic.

→ The second difference relates to the nature of the elementary unit which constitutes the formulas of the two formal systems.

Propositional logic uses propal propositions and logical operators in order to constitute its well-formed formulas.

First order logic, in addition to these, also uses variables, quantifiers and relationships.

→ The third difference concerns the capacity for the abstraction of formulas expressed by the two systems.

In propositional logic, the system doesn't allow to handle problems that involve mutating or undetermined parts.

This means that the validity of a solution to a formula found in propositional logic is ~~not~~ restricted to that formula.

Characteristics	Propositional Logic	First order Logic
(1) Includes others	No	Yes
(2) Uses variables	No	Yes
(3) Unit of analysis	Propositions and operators	Predicates, objects, relationships, quantifiers.

eg :- x is an integer.

subject : x
Predicate : is an integer

int(x) :- x is an integer.

* Universe of Discourse - The universe of discourse (UOD) is the set of all things we wish to talk about ; that is, the set of all objects that we can sensibly assign to a variable in a propositional function.

* Syntax of First order logic

→ First order logic has a syntax that's comprised of symbols belonging to one of the two classes:
Logical and non-logical symbol.

→ Logical symbols are those that correspond to logical operators or ~~connectivites~~ connectives.

eg of such symbols include the logical operators $\vee$, $\wedge$, $\rightarrow$ which serve functions identical to their corresponding operators in propositional logic.

→ Logical symbols of this type are always and only interrupted in the sense of the logical operation that they represent and their meaning is never conditioned by the domain of discussion in which we use first order logic.

→ Non-logical symbols comprise predicates and relationships but also contents and functions. The meaning associated with non-logical symbols is domain-specific and their conversion to sentence in a natural language requires conversion rules and interpretation.

* Predicates -

Predicates are the fundamental components of formulas and indicate relations between objects.
These relationships can be only of that are allows in a discursive domain.

* Quantifiers -

→ Quantifiers express properties of entire collections of objects, instead of enumerating the objects by name.

First order logic contains two standard quantifiers-

1. Universal Quantifiers
2. ~~Exit~~ Existential Quantifiers

→ We can more formally define quantifiers through the usage of two symbols :-
$\forall$ and $\exists$

* Inference Engine – Use of efficient procedures and rules by the inference engine is essential in deducing a correct, flawless solution.

In case of knowledge-based ES, the Inference Engine acquires and manipulates the knowledge from the knowledge base to arrive at a particular solution.

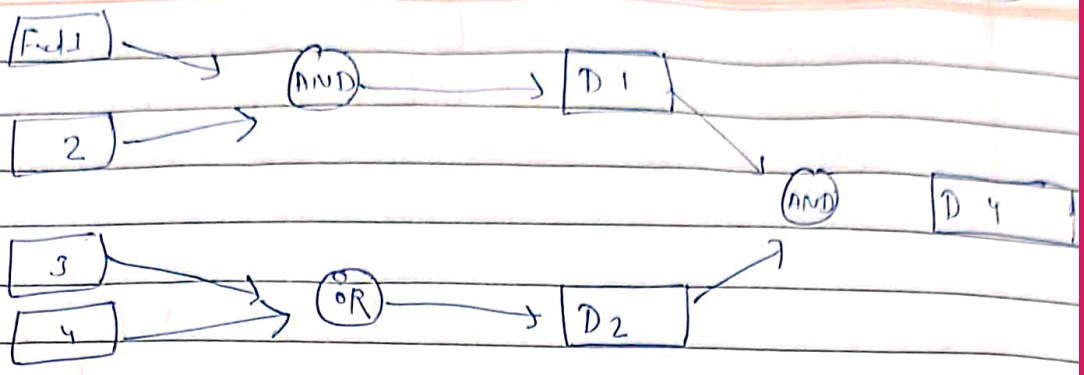
To recommend a solution, the Inference Engine uses the following strategies –

- Forward Chaining
- Backward Chaining

* Forward Chaining

→ It is strategy of an expert system to answer the question, "What can happen next?"

Here, the Inference Engine follows the chain of conditions and derivations and finally deduces the outcomes. It considers all the facts and rules, and sorts them before concluding to a solution.



* Backward Chaining

→ With this strategy, an expert systems finds out the answer to the question, "Why this happened?"

→ On the basis of what has already happened the Inference Engine tries to find out which conditions could have happened in the past for this result. This strategy is followed for finding out cause or reason.

* Expert Systems — Expert systems (ES) are one of the prominent research domain of AI. It is introduced by the researchers at Stanford University, Computer Science Department.

→ The expert systems are the computer applications developed to solve complex problems in a particular domain, at the level of extra-ordinary human intelligence and expertise.

Characteristics —

- High performance
- Understandable
- Reliable
- Highly responsive

Capabilities — • Advising
• ~~Instructing~~ • Instructing and assisting human in decision making
- Demonstrating
- Deriving a solution
- Diagnosing
- Explaining
- Interpreting input
- Predicting results
- Justifying the conclusion
- Suggesting alternative options to a problem

- Incapable of -

→ Substituting human decision makers

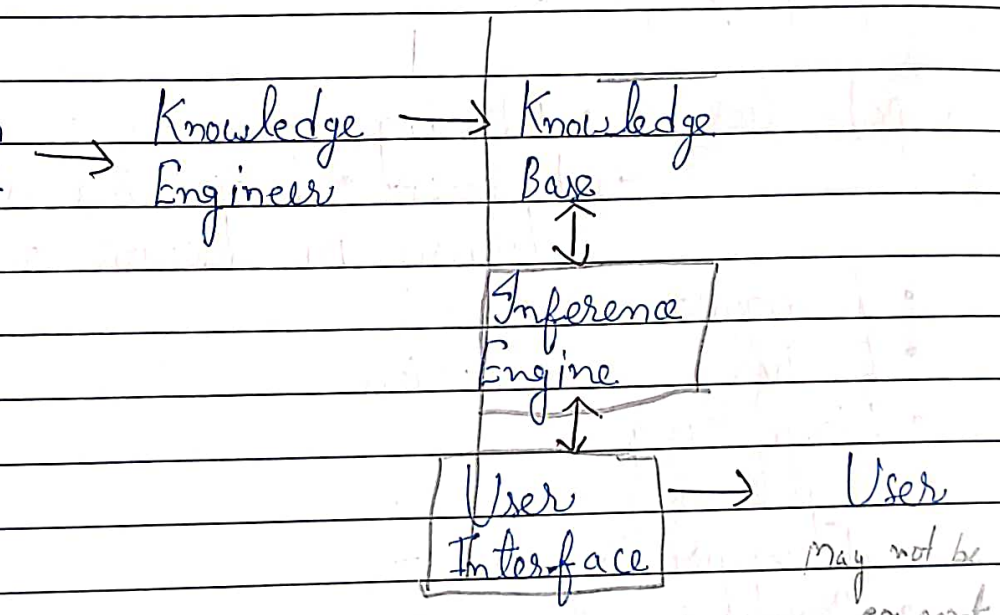
→ Possessing human capabilities

→ Producing accurate output for inadequate knowledge base

→ Refining their own knowledge

The components of ES includes -

- Knowledge Base
- Inference Engine
- User Inference



* Applications of Expert System

The following table shows where ES can be applied-

Application	Description
Design Domain	Camera lens design, automobile design.
Medical Domain	Diagnosis Systems to deduce cause of disease from observed data, conduction medical operations on human.
Process Control System	Controlling a physical process based on monitoring.
Knowledge Domain	Finding out faults in vehicles, computers.
Finance/ Commerce	Detection of possible fraud, suspicious transactions, stock market trading, Airline scheduling, cargo scheduling.

* Benifits of Expert Systems-

- Availability — They are easily available due to mass production of software.
- Less Production Cost — Production cost is reasonable. This makes them affordable.
- Speed — They offer great speed. They reduce the amount of work an individual puts in.
- Less Error Rate — Error rate is low as compared to human errors.
- Reducing Risk — They can work in the environment dangerous to humans.
- Steady response — They work steadily without getting emotional, tensed or fatigued.

Fuzzy Logic Systems-

Fuzzy logic system (FLS) produce acceptable but definite output in response to incomplete, ambiguous distorted or inaccurate (fuzzy) input.

* Fuzzy Logic — Fuzzy logic (FL) is a method of reasoning that resembles human reasoning. The approach of FL imitates the way of decision making in humans that involves all intermediate possibilities between digital values Yes and No.

→ The conventional logic block that a computer can understand takes precise input and produces a definite output as TRUE or FALSE, which is equivalent to human's YES or NO.

→ The inventor of fuzzy logic, Lotfi Zadeh observed that unlike computers, the human decision making includes a range of possibilities between YES and NO, such as-

CERTAINLY	YES
POSIBLY	~~NO~~ YES
CANNOT	SAY
POSIBLY	NO
~~CERTAINLY~~	NO

Po'

The fuzzy logic works on the levels of possibilities of input to achieve the definite output.

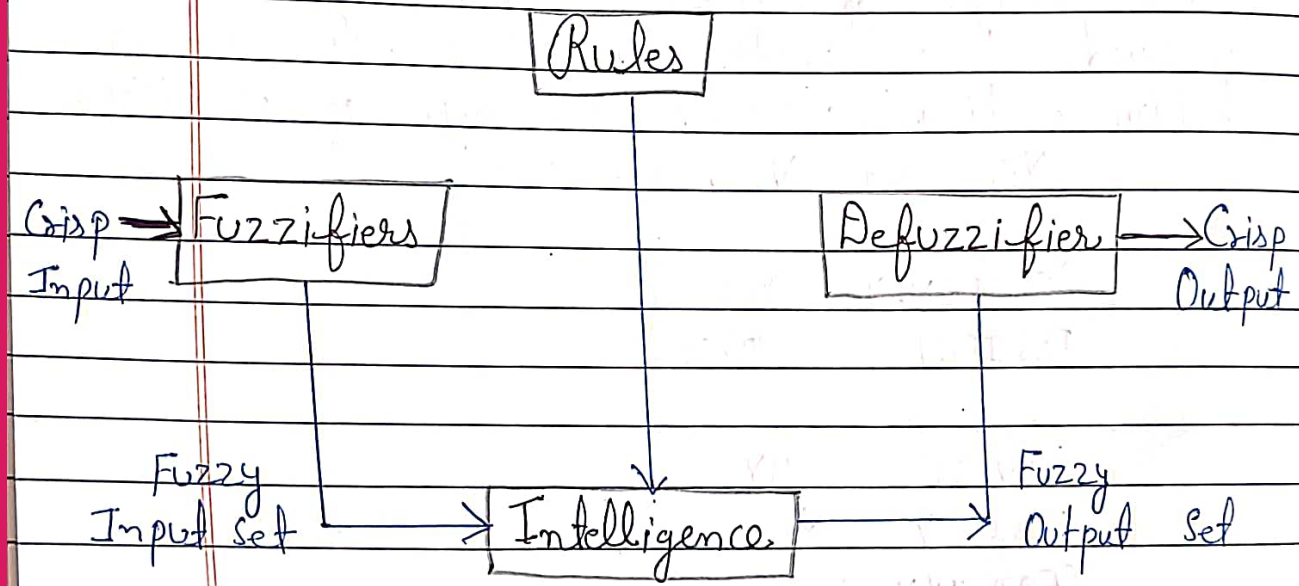
→ Fuzzy logic is useful for commercial and practical purposes.

→ It can control machines and consumer products.

→ It may not give accurate reasoning, but acceptable reasoning.

→ Fuzzy logic helps to deal with the uncertainly in engineering.

Fuzzy Logic Systems Architecture –



It has 04 main parts as shown -

(1). Fuzzification Module - It transforms the system inputs, which are crisp numbers, into fuzzy sets. It splits the input signal into 05 steps such as

LP	x	is	Large Positive
MP	x	is	Medium Positive
S	x	is	Small
MN	x	is	Medium Negative
LN	x	is	Large Negative

(2) Knowledge Base - It stores IF-THEN rules provided by experts.

(3) Inference Engine - It simulates the human reasoning process by making fuzzy inference on the inputs and IF-THEN rules.

(4). Defuzzification Module - It transforms the fuzzy set obtained by the inference engine into a crisp value.

Application Area -

(1). Automotive Systems → Automatic Gearboxes
→ Four-Wheel Steering
→ Vehicle environment control

(2). Consumer Electronics → Hi-Fi Systems
→ Photocopies
→ Still and Video Cameras
→ Television

(3). Domestic Goods → Microwave Ovens
→ Refrigerators
→ Toasters
→ Vacuum Cleaners
→ Washing Machines.

(4). Environment Control → Air Conditioners / Dryers / Heaters
→ Humidifiers

* Advantages =

→ Mathematical concepts within fuzzy reasoning are very simple.
→ You can modify a FLS by just adding or deleting rules due to flexibility of fuzzy logic.
→ FLS are easy to construct and understand.

→ Fuzzy logic systems can take imprecise, distorted, noisy input information.

→ Fuzzy logic is a solution to complex problems in all fields in life, including medicine, as it resembles human reasoning and decision making.

* Disadvantages of FLSs

→ There is no systematic approach to fuzzy system designing.

→ They are understandable only ~~when~~ when simple.

→ They are suitable for the problems which do not need high accuracy.

Bayesian Networks (BN)

These are the graphical structures used to represent the probabilistic relationship among a set of random variables.

Bayesian networks are also called Belief Networks or Bayes Nets. BNs reason about uncertain domain.

→ In these networks, each node represents a random variable with specific proposition.

eg - in a medical diagnosis domain, the node Cancer represents the proposition that a ~~patent~~ patient has cancer.

The edges connecting the nodes represent probabilistic dependencies among those random variables. If out of two nodes, one is affecting the other then they must be directly connected in the directions of the effect. The strength of the relationship between variables is quantified by the probability associated with each node.

There is an only constraints on the arcs in a BN that you cannot return to a node simply by following directed arcs. Hence the BNs are called Directed Acyclic Graphs (DAGs)

BN are capable of handling multivalued variables simultaneously. The BN variables are composed of two dimensions-

- Range of propositions.
- Probabilities assigned to each of the propositio

→ The structure of BN is ideal for combining prior knowledge and observer data. BN can be used to learn the casual relationships and understand various problem domains and to predict future events, ~~even~~ even in case of missing data.

ARTIFICIAL NEURAL NETWORKS (ANNs) -

→ The inventor of the first neurocomputer, Dr. Robert Hecht - Nielsen, defines a neural network as -

"... a computing system made up of a number of simple, highly interconnected processing elements which process information by their dynamic state response to external inputs."

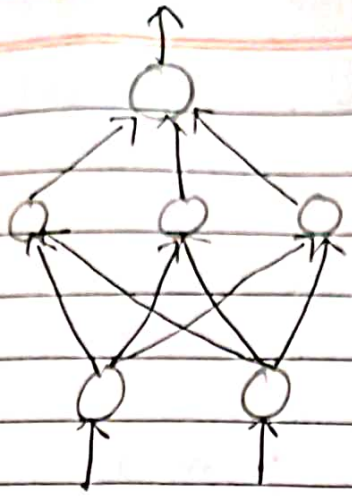
There are two ANN topologies -

- FeedForward & Feedback

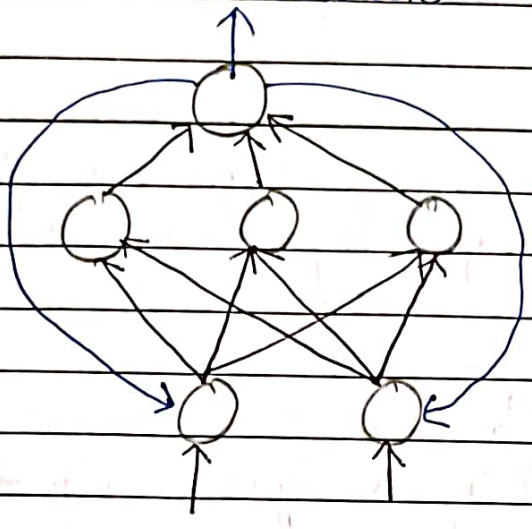
(1). FeedForward ANN The information flow is unidirectional. A unit sends information to other unit from which it does not receive any information. There are no feedback loops.

→ They are used in pattern generation / recognition / classification. They have fixed inputs and outputs.

(2)



(1) FeedBack ANN – They are used in content addressable memories.



Working – In the topology diagram shown, each arrow represents a connection between two neurons and indicates the pathway for the flow of information. Each connection has a weight, an integer number that controls the signal b/w the two neurons.

If the network generates a "good or desired" output, there is no need to adjust the weights. However, if the network generates a "poor or undesired" output or an error, then the system alters the weights in order to improve subsequent results.

* Applications of Neural Networks

(1) Aerospace – Autopilot aircrafts, aircraft fault detection.

(2) Automotive – Automobile guidance systems.

(3) Military – Weapon orientation and steering, target tracking, object discrimination, facial recognition, signal/image identification.

(4) Electronics – Code sequence prediction, IC chip layout, chip failure analysis, machine vision, voice synthesis.

(5) Medical – Cancer cell analysis, EEG and ECG analysis, ~~prothet~~ prosthetic design, transplant time optimizer.

(6) Speech – Speech recognition, speech classification, text to speech conversion.

(7) Telecommunications – Image and data compression, automated information services, real time spoken language translation.

(8) Transportation – Truck Brake system diagnosis, vehicle scheduling, routing systems.

(9) Software – Pattern Recognition in facial recognition, optical character recognition etc.

(10) Control – ANNs are often used to make steering decisions of physical vehicles.

(11) Time series Prediction – ANNs are used to make predictions on stocks and natural calamities.

Backpropagation –

Backpropagation is a widely used algorithm for training feedforward neural networks. It computes the gradient of the loss function with respect to the network weights and is very efficient, rather than naively directly computing the gradient with respect to each individual weight. This efficiency makes it possible to use gradient methods to train multi-layer networks and update weights to minimize loss; variants such as gradient descent or stochastic gradient are often used.

The backpropagation algorithm works by computing the gradient of the loss function with respect to each weight via the layer by layer, and iterating backward from the last layer to avoid redundant computation of intermediate terms in the chain rule.

- Working of Backpropagation -

Neural networks use supervised learning to generate output vectors from input vectors that the network operates on. It compares generated output to the desired output and generates an error report if the result does not match the generated output vector. Then it adjusts the weights according to the bug report to get your desired output.

* Backpropagation Algorithm -

Step 1 :- Inputs x, arrive through the pre-connected path.

Step 2 :- The input is modeled using true weight W. Weights are usually chosen randomly.

Step 3 - Calculate the output of each neuron from the input layer, to the hidden layer, to the output layer.

Step 4 - Calculate the error in the output

Backpropagation Error = Actual Output - Desired Output

Step 5 - From the output layer, go back to the hidden layer to adjust the weights to reduce the error.

Step 6 - Repeat the process until the desired output is achieved.

Types -

- Static backpropagation - It is a network designed to map static inputs for static outputs.
- Recurrent backpropagation - Recursive backpropagation is another network used for fixed point learning.

Advantages -

→ It is simple, fast and easy to program.
→ It is flexible and efficient.
→ No need for users to learn any special functions.

- Disadvantages -

→ Performance is highly dependent on input data.

→ Spending too much time training.

→ The matrix-based approach is preferred over a mini-batch.

* **Scripts** – A script is a structure that prescribes a set of circumference which could be expected to follow on from one another.

→ It is similar to a thought sequence or a chain of situations which could be anticipated.

→ It could be considered to consist of a number of slots or frames but with more specialized roles.

Scripts are benificial because –

- Events tend to occur in known runs or patterns.
- Causal relationships between events exist.
- Entry conditions exist which allow an event to take place.
- Prerequesting exist upon events taking place.

eg – when a student progresses through a degree scheme or when a purchaser buys a house.

* Advantages —

→ Ability to predict events.

→ A single coherent interpretation may be build up from a collection of observation.

* Disadvantages —

→ Less general than frames.

→ May not be suitable to represent all kinds of knowledge.

* The components of a script include :-

- Entry Conditions — These must be satisfied before events in the script can occur.

Results — Conditions that will be true after events in script occur.

Props — Slots representing objects involved in events.

~~Track~~ Roles — Persons involved in the events.

~~Scenes~~ Tracks — Variations on the script. Different tracks may share components of the same script.

Scenes - The sequence of events that occur. Events are represented in conceptual dependency form.