Problem Characteristics in Artificial Intelligence

To choose an appropriate method for a particular problem first we need to categorize the problem based on the following characteristics.

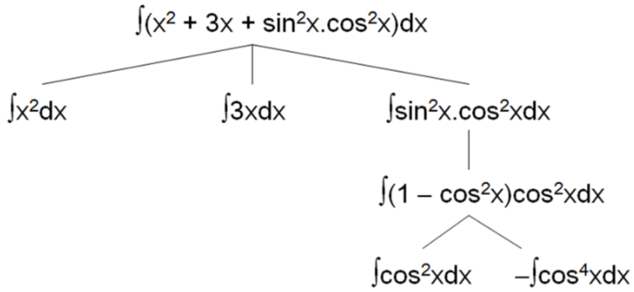
1. Is the problem decomposable into small sub-problems which are easy to solve?
2. Can solution steps be ignored or undone?
3. Is the universe of the problem is predictable?
4. Is a good solution to the problem is absolute or relative?
5. Is the solution to the problem a state or a path?
6. What is the role of knowledge in solving a problem using artificial intelligence?
7. Does the task of solving a problem require human interaction?

1. Is the problem decomposable into small sub-problems which are easy to solve?

Can the problem be broken down into smaller problems to be solved independently?

The decomposable problem can be solved easily.

Example: In this case, the problem is divided into smaller problems. The smaller problems are solved independently. Finally, the result is merged to get the final result.

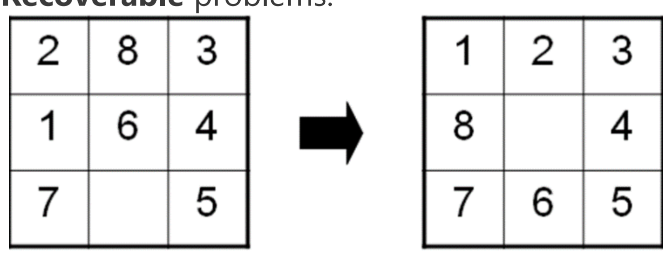


2. Can solution steps be ignored or undone?

In the Theorem Proving problem, a lemma that has been proved can be ignored for the next steps.

Such problems are called **Ignorable** problems.
In the 8-Puzzle, Moves can be undone and backtracked.

Such problems are called **Recoverable** problems.



In Playing Chess, moves can be retracted.

Such problems are called **Irrecoverable** problems.
Ignorable problems can be solved using a simple control structure that never backtracks. **Recoverable** problems can be solved using backtracking. **Irrecoverable** problems can be solved by recoverable style methods via planning.

3. Is the universe of the problem is predictable?

In Playing Bridge, We cannot know exactly where all the cards are or what the other players will do on their turns. Uncertain outcome!

For **certain-outcome problems**, planning can be used to generate a sequence of operators that is guaranteed to lead to a solution.
For **uncertain-outcome problems**, a sequence of generated operators can only have a good probability of leading to a solution. Plan revision is made as the plan is carried out and the necessary feedback is provided.

4. Is a good solution to the problem is absolute or relative?

The Travelling Salesman Problem, we have to try all paths to find the shortest one. Any path problem can be solved using heuristics that suggest good paths to explore.

For best-path problems, a much more exhaustive search will be performed.

5. Is the solution to the problem a state or a path

The Water Jug Problem, the path that leads to the goal must be reported.

A path-solution problem can be reformulated as a state-solution problem by describing a state as a partial path to a solution. The question is whether that is natural or not.

6. What is the role of knowledge in solving a problem using artificial intelligence?

Playing Chess
Consider again the problem of playing chess. Suppose you had unlimited computing power available. How much knowledge would be required by a perfect program? The answer to this question is very little—just the rules for determining legal moves and some simple control mechanism that implements an appropriate search procedure.
Additional knowledge about such things as good strategy and tactics could of course help considerably to constrain the search and speed up the execution of the program. Knowledge is important only to constrain the search for a solution.

7. Does the task of solving a problem require human interaction?

Sometimes it is useful to program computers to solve problems in ways that the majority of people would not be able to understand. This is fine if the level of the interaction between the computer and its human users is problem-in solution-out.

But increasingly we are building programs that require intermediate interaction with people, both to provide additional input to the program and to provide additional reassurance to the user.

The **solitary problem**, in which there is no intermediate communication and no demand for an explanation of the reasoning process.

Control Strategy and Requirements in AI

Given a problem definition, it is very difficult to decide which rule to choose from a set of rules which can solve the problem efficiently. This question usually arises when we have more than one rule (and sometimes fewer than one rule) that will match with the left side of the current state. Even without a great deal of thought, it is clear that such decisions will have a crucial impact on how quickly. and even whether, a problem is finally solved.

There are mainly two requirements to choose a control strategy :-

1. The **first requirement** of a good control strategy is that it causes **motion**.

Example: Consider the water jug problem. Suppose we implemented the simple control strategy of starting each time at the top of the list of rules and choosing the first applicable one. If we did that, we would never solve the problem. We would continue indefinitely filling the 4-gallon jug with water Control strategies that do not cause motion will never lead to a solution.

2. The **second requirement** of a good control strategy is that it be systematic.

Example: Consider the water jug problem. On each cycle, choose at random from among the applicable rules. This strategy is better than the first. It causes motion. It will lead to a solution eventually. But we are likely to arrive at the same state several times during the process and to use many more steps than are necessary. Because the control strategy is not systematic, we may explore a particular useless sequence of operators several times before we finally find a solution.

Artificial Intelligence and its Task Domains

According to the father of Artificial Intelligence, John McCarthy, it is *"The science and engineering of making intelligent machines.*

Artificial Intelligence tasks are divided into three groups, Mundane Tasks, Formal Tasks and Expert Tasks.

Mundane Tasks:

- Perception
  - Vision
  - Speech
- Natural Languages
  - Understanding
  - Generation
  - Translation
- Common sense reasoning
- Robot Control

Formal Tasks

- Games: chess, checkers, go, etc.
- Mathematics: Geometry, logic, Proving properties of programs, Integral calculus

Expert Tasks:

- Engineering (Design, Fault finding, Manufacturing planning)
- Scientific Analysis
- Medical Diagnosis
- Financial Analysis

Water Jug Problem in Artificial Intelligence

Water Jug Problem Definition,

"You are given two jugs, a 4-liter one and a 3-liter one. Neither has any measuring markers on it. There is a pump that can be used to fill the jugs with water. How can you get exactly 2 liters of water into a 4-liter jug."

Representation of water Jug Problem in terms of state-space search,

State: (x, y)

Where, x represents the quantity of water in a 4-liter jug and
y represents the quantity of water in a 3-liter jug.

That is, x = 0, 1, 2, 3, or 4; y = 0, 1, 2, 3

Start state: (0, 0).

Goal state: (2, n) for any n.

Here need to start from the current state and end up in a goal state.

Production Rules for Water Jug Problem in Artificial Intelligence

1	(x, y) is X<4 ->(4, Y)	Fill the 4-liter jug
2	(x, y) if Y<3 -> (x, 3)	Fill the 3-liter jug
3	(x, y) if x>0 -> (x-d, d)	Pour some water out of the 4-liter jug.
4	(x, y) if Y>0 -> (d, y-d)	Pour some water out of the 3-liter jug.
5	(x, y) if x>0 -> (0, y)	Empty the 4-liter jug on the ground

6	(x, y) if y>0 -> (x,0)	Empty the 3-liter jug on the ground
7	(x, y) if X+Y >= 4 and y>0 -> (4, y-(4-x))	Pour water from the 3-liter jug into the 4-liter jug until the 4-liter jug is full
8	(x, y) if X+Y>=3 and x>0 -> (x-(3-y), 3))	Pour water from the 4-liter jug into the 3-liter jug until the 3-liter jug is full.
9	(x, y) if X+Y <=4 and y>0 -> (x+y, 0)	Pour all the water from the 3-liter jug into the 4-liter jug.
10	(x, y) if X+Y<=3 and x>0 -> (0, x+y)	Pour all the water from the 4-liter jug into the 3-liter jug.
11	(0, 2) -> (2, 0)	Pour the 2-liter water from the 3-liter jug into the 4-liter jug.
12	(2, Y) -> (0, y)	Empty the 2-liter in the 4-liter jug on the ground.

The solution to Water Jug Problem in Artificial Intelligence

(0, 0) – Start State

(4, 0) – Rule 1, Fill the 4-liter jug

(1, 3) – Rule 8, Pour water from the 4-liter jug into the 3-liter jug until the 3-liter jug is full.

(1, 0) – Rule 6, Empty the 3-liter jug on the ground

(0, 1) – Rule 10, Pour all the water from the 4-liter jug into the 3-liter jug.

(4, 1) – Rule 1, Fill the 4-liter jug

(2, 3) – Rule 8, Pour water from the 4-liter jug into the 3-liter jug until the 3-liter jug is full.

(2, 0) – Rule 6, Empty the 3-liter jug on the ground

Goal State reached

Means-Ends Analysis Artificial Intelligence

In Artificial Intelligence, we have studied many search strategies which traverse either in forward or backward direction, but a mixture of these two is usually appropriate to solve a complex and large problem.

Such a mixed search algorithm, allow us to solve the major part of a problem first and then go back and solve the small problems which arise while combining the major parts of the problem. Such a technique is called **Means-Ends Analysis (MEA)**.

The MEA analysis process centered on finding the difference between the current state and goal state and applying the operators to reduce this difference.

To solve a given problem, we need to apply the MEA Algorithm recursively.
Following are the major steps that describe the working principle of the MEA search technique to solve the problem.

1. First, find the difference between Initial (start) State and Goal State.
2. From the available set of operators, select an operator which can be applied to the current state to reduce the difference between the current state and goal state.
3. Apply the selected operator.

In the MEA process, first, we find the differences between the current state and the goal state.

Once we find the differences, then we apply an operator to reduce the differences.

But sometimes it is possible that an operator cannot be applied to the current state.

Hence, we divide the current state into sub-problems, and then we apply an operator on sub-problems, such type of analysis is called **Operator Subgoaling**.

Let’s assume the Current state as CURRENT and Goal State as GOAL, then the following are the **steps of the MEA algorithm**.

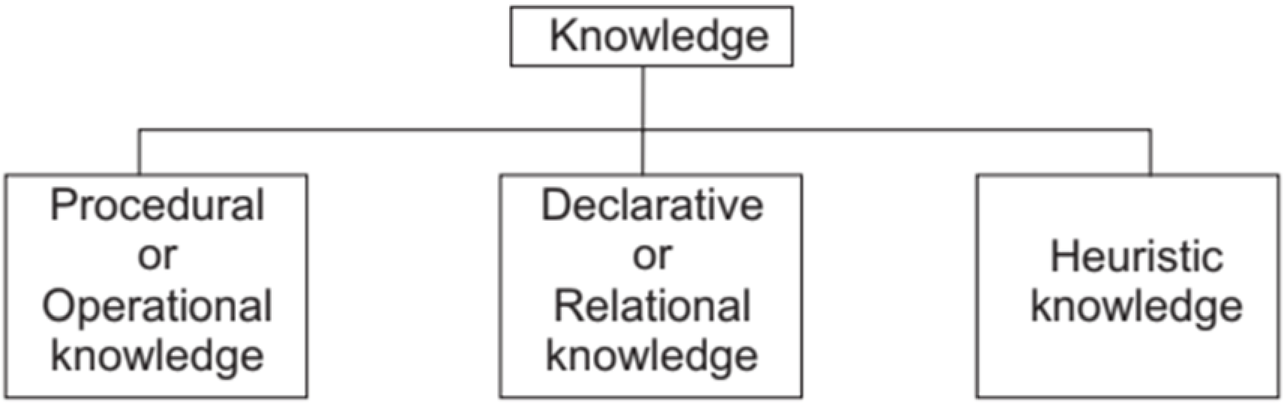
Step 1: Compare CURRENT to GOAL, if there are no differences between both then return Success and Exit.
Step 2: Else, select the most significant difference between CURRENT and GOAL and reduce it by doing the following steps until success or failure occurs.
a) Select a new operator **O** which is applicable for the current difference, if there is no such operator, then signal failure.
b) Attempt to apply operator **O** to CURRENT. Make a description of two states.
i) O-Start, a state in which O’s preconditions are satisfied.
ii) O-Result, the state that would result if O were applied In O-start.
c) If **(First-Part <—— MEA (CURRENT, O-START)** And **LAST-Part <—— MEA (O-Result, GOAL)**, are successful, then signal Success and return the result of combining FIRST-PART, O, and LAST-PART.

Types of Knowledge Representation AI

Knowledge can be defined as the body of facts and principles accumulated by humankind or the act, fact, or state of knowing.

For example, in biological organisms, knowledge is stored as complex structures of interconnected neurons. The structures correspond to symbolic representations of the knowledge possessed by the organism, the facts, rules and so on.

Knowledge is of three types as shown below:



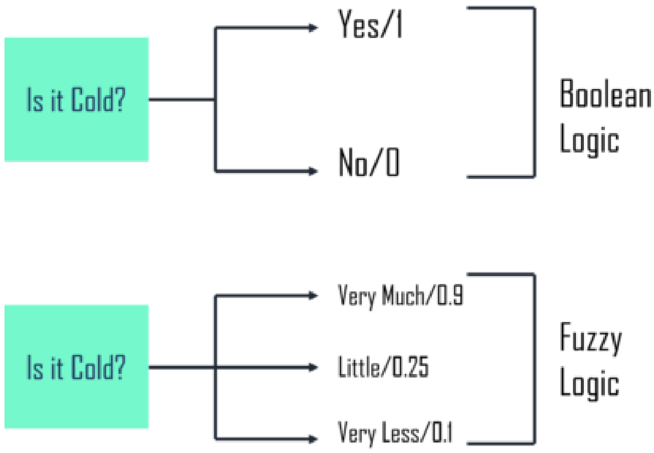
1. Procedural or Operational Knowledge : It is defined as compiled knowledge related to the performance of some task. For example, steps to solve a quadratic equation are expressed as procedural knowledge.

2. Declarative or Relational Knowledge : It is passive knowledge expressed as statements of facts about the world. For example, personnel data in a database. Such data are explicit pieces of independent knowledge.

3. Heuristic Knowledge : Heuristics means using some rules of thumb or tricks or strategies to simplify the solution to problems. We acquire this after much experience.

Fuzzy Logic

1. Fuzzy refers to something which is unclear or vague. Fuzzy Logic in AI provides valuable flexibility for reasoning.
2. Fuzzy Logic (FL) is a method of reasoning that resembles human reasoning. This approach is similar to how humans perform decision making. And it involves all intermediate possibilities between YES and NO.



3. The Fuzzy logic was invented by **Lotfi Zadeh** who observed that unlike computers, humans have a different range of possibilities between YES and NO, such as:

Certainly Yes
Possibly Yes
Cannot Say
Possibly No
Certainly No

4. The Fuzzy logic works on the levels of possibilities of input to achieve a definite output.
5. The implementations of Fuzzy logic are:

    - It can be implemented in systems with different sizes and capabilities such as **micro-controllers, large networked** or **workstation-based systems.**
    - Also, it can be implemented in **hardware, software** or a combination of **both**.

6. Applications of Fuzzy Logic

    The Fuzzy logic is used in various fields such as automotive systems, domestic goods, environment control, etc. Some of the common applications are:

    - It is used in the **aerospace field** for **altitude control** of spacecraft and satellite.
    - This controls the **speed and traffic** in the **automotive systems.**

Certainty Factor

1. The **Certainty Factor (CF)** is a numeric value which tells us about how likely an event or a statement is supposed to be true.
2. It is somewhat similar to what we define in probability, but the difference in it is that an agent after finding the probability of any event to occur cannot decide what to do.
3. Based on the probability and other knowledge that the agent has, this **certainty factor** is decided through which the agent can decide whether to declare the statement true or false.
4. The value of the **Certainty factor** lies between **-1.0 to +1.0**, where the negative 1.0 value suggests that the statement can never be true in any situation, and the positive 1.0 value defines that the statement can never be false.
5. The value 0 suggests that the agent has no information about the event or the situation.
6. A minimum **Certainty factor** is decided for every case through which the agent decides whether the statement is true or false. This minimum **Certainty factor** is also known as the threshold value.
7. For example, if the minimum **certainty factor** (threshold value) is 0.4, then if the value of CF is less than this value, then the agent claims that particular statement false.

Monotonic Reasoning:

1. In monotonic reasoning, once the conclusion is taken, then it will remain the same even if we add some other information to existing information in our knowledge base.
2. In monotonic reasoning, adding knowledge does not decrease the set of prepositions that can be derived.
3. To solve monotonic problems, we can derive the valid conclusion from the available facts only, and it will not be affected by new facts.
4. Monotonic reasoning is not useful for the real-time systems, as in real time, facts get changed, so we cannot use monotonic reasoning.
5. Monotonic reasoning is used in conventional reasoning systems, and a logic-based system is monotonic.
6. Any theorem proving is an example of monotonic reasoning.
7. **Example:**

   Earth revolves around the Sun.

   It is a true fact, and it cannot be changed even if we add another sentence in knowledge base like, "The moon revolves around the earth" Or "Earth is not round," etc.

Scripts

1. A script is a structured representation describing a stereotyped sequence of events in a particular context.
2. Scripts are used in natural language understanding systems to organize a knowledge base in terms of the situations that the system should understand.
3. Scripts use a frame-like structure to represent the commonly occurring experience like going to the movies, eating in a restaurant or shopping in a supermarket.
4. Thus, a script is a structure that prescribes a set of circumstances that could be expected to follow on from one another.
5. The components of a script include:
   - Entry condition: These are basic condition which must be fulfilled before events in the script can occur.
   - Results: Condition that will be true after events in script occurred.
   - Props: Slots representing objects involved in events
   - Roles: These are the actions that the individual participants perform.
   - Track: Variations on the script. Different tracks may share components of the same scripts.
   - Scenes: The sequence of events that occur.

Frames

1. A frame is a record like structure which consists of a collection of attributes and its values to describe an entity in the world.
2. Frames are the AI data structure which divides knowledge into substructures by representing stereotypes situations.
3. It consists of a collection of slots and slot values. These slots may be of any type and sizes. Slots have names and values which are called facets.
4. A frame is also known as **slot-filter knowledge representation** in artificial intelligence.
5. The frame is a type of technology which is widely used in various applications including Natural language processing and machine visions.
6. Example: Let's suppose we are taking an entity, Peter. Peter is an engineer as a profession, and his age is 25, he lives in city London, and the country is England. So following is the frame representation for this:

Slots	**Filter**
Name	Peter
Profession	Doctor
Age	25
Marital status	Single
Weight	78

Conceptual Dependency

1. The Conceptual Dependency is used to represent knowledge of Artificial Intelligence.
2. It should be powerful enough to represent these concepts in the sentence of natural language.
3. It states that different sentence which has the same meaning should have some unique representation.
4. There are 5 types of states in Conceptual Dependency:
    - Entities
    - Actions
    - Conceptual cases
    - Conceptual dependencies
    - Conceptual tense
5. Main Goals of Conceptual Dependency:
    - It captures the implicit concept of a sentence and makes it explicit.
    - It helps in drawing inferences from sentences.
    - For any two or more sentences that are identical in meaning. It should be only one representation of meaning.
    - It provides a means of representation which are language independent.
    - It develops language conversion packages.

Expert system

1. An expert system is a computer program that is designed to solve complex problems and to provide decision-making ability like a human expert.
2. It performs this by extracting knowledge from its knowledge base using the reasoning and inference rules according to the user queries.

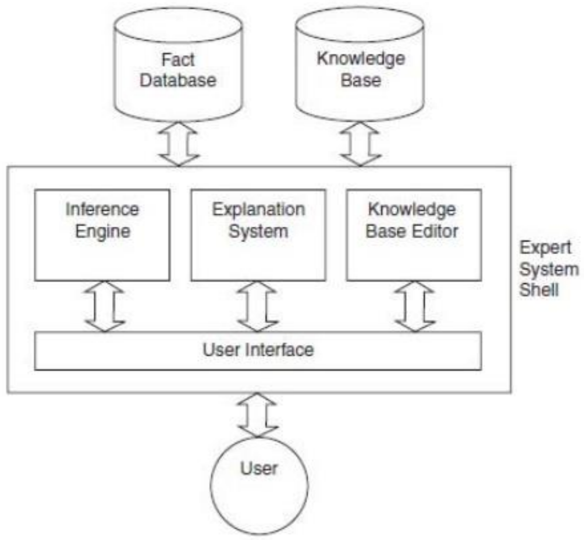


Fig Expert System Architecture

3. The **knowledge base** contains the specific domain knowledge that is used by an expert to derive conclusions from facts.
4. In the case of a rule-based expert system, this domain knowledge is expressed in the form of a series of rules.
5. The **explanation system** provides information to the user about how the inference engine arrived at its conclusions.
6. If the system has used faulty reasoning to arrive at its conclusions, then the user may be able to see this by examining the data given by the explanation system.
7. The **fact database** contains the case-specific data that are to be used in a particular case to derive a conclusion.
8. The user of the expert system interfaces with it through a **user interface**, which provides access to the inference engine, the explanation system, and the knowledge-base editor.
9. The **inference engine** is the part of the system that uses the rules and facts to derive conclusions. The inference engine will use forward chaining, backward chaining, or a combination of the two to make inferences from the data that are available to it.
10. The **knowledge-base editor** allows the user to edit the information that is contained in the knowledge base. The knowledge-base editor is not usually made available to the end user of the system but is used by the knowledge engineer or the expert to provide and update the knowledge that is contained within the system.

Levels of Natural Language Processing

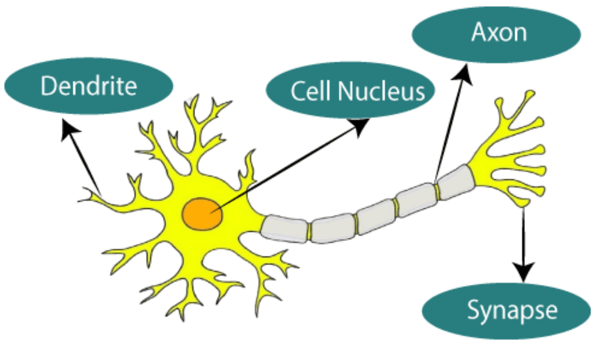
NLP stands for **Natural Language Processing,** is the technology that is used by machines to understand, analyze, manipulate, and interpret human's languages.

Levels of NLP :

1. Morphological Analysis-Individual words are analyzed into their components, and nonword tokens, such as punctuation, are separated from the words.
2. Syntactic Analysis-Linear sequences of words are transformed into structures that show how the words relate to each other. Some word sequences may be rejected if they violate the language's rules for how words may be combined. For example, an English syntactic analyzer would reject the sentence "Boy the go the to store."
3. Semantic Analysis-The structures created by the syntactic analyzer are assigned meanings. In other words, a mapping is made between the syntactic structures and objects in the task domain. Structures for which no such mapping is possible may be rejected. For example, in most universes, the sentence "Colorless green ideas sleep furiously" [Chomsky, 1957] would be rejected as semantically anomalous.
4. Discourse Integration-The meaning of an individual sentence may depend on the sentences that precede it and may influence the meanings of the sentences that follow it. For example, the word "it" in the sentence, "John wanted it," depends on the prior discourse context, while the word "John" may influence the meaning of later sentences (such as, "He always had.")
5. Pragmatic Analysis-The structure representing what was said is reinterpreted to determine what was actually meant. For example, the sentence "Do you know what time it is?" should be interpreted as a request to be told the time.

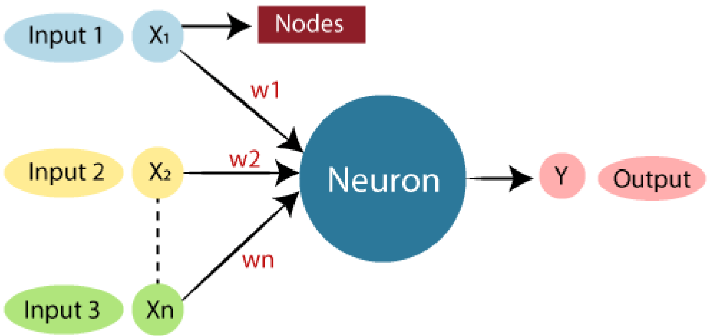
Artificial Neural Network

The term "**Artificial Neural Network**" is derived from Biological neural networks that develop the structure of a human brain. Similar to the human brain that has neurons interconnected to one another, artificial neural networks also have neurons that are interconnected to one another in various layers of the networks. These neurons are known as nodes.



The given figure illustrates the typical diagram of Biological Neural Network.

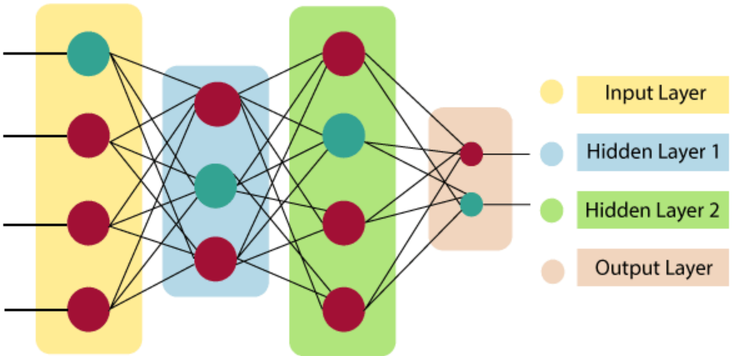
The typical Artificial Neural Network looks something like the given figure.



Dendrites from Biological Neural Network represent inputs in Artificial Neural Networks, cell nucleus represents Nodes, synapse represents Weights, and Axon represents Output.

The artificial neural network is designed by programming computers to behave simply like interconnected brain cells.

Artificial Neural Network primarily consists of three layers:



Input Layer: As the name suggests, it accepts inputs in several different formats provided by the programmer.

Hidden Layer: The hidden layer presents in-between input and output layers. It performs all the calculations to find hidden features and patterns.

Output Layer: The input goes through a series of transformations using the hidden layer, which finally results in output that is conveyed using this layer.

Applications of ANN

1. social media

Artificial Neural Networks are used heavily in Social Media. For example, let's take the **'People you may know'** feature on Facebook that suggests you people that you might know in real life so that you can send them friend requests. Well, this magical effect is achieved by using Artificial Neural Networks that analyze your profile, your interests, your current friends, and also their friends and various other factors to calculate the people you might potentially know.

2. Marketing and Sales

When you log onto E-commerce sites like Amazon and Flipkart, they will recommend your products to buy based on your previous browsing history. Similarly, suppose you love Pasta, then Zomato, Swiggy, etc. will show you restaurant recommendations based on your tastes and previous order history. This uses Artificial Neural Networks to identify the customer likes, dislikes, previous shopping history, etc. and then tailor the marketing campaigns accordingly.

3. Healthcare

Artificial Neural Networks are used in Oncology to train algorithms that can identify cancerous tissue at the microscopic level at the same accuracy as trained physicians. Various rare diseases may manifest in physical characteristics and can be identified in their premature stages by using **Facial Analysis** on the patient photos.

4. Personal Assistants

I am sure you all have heard of Siri, Alexa, Cortana, etc. and also heard them based on the phones you have!!! These are personal assistants and an example of speech recognition that uses **Natural Language Processing** to interact with the users and formulate a response accordingly. Natural Language Processing uses artificial neural networks that are made to handle many tasks of these personal assistants such as managing the language syntax, semantics, correct speech, the conversation that is going on, etc.

Genetic algorithm

1. A genetic algorithm is used to solve complicated problems with a greater number of variables & possible outcomes/solutions.
2. The combinations of different solutions are passed through the Darwinian based algorithm to find the best solutions. The poorer solutions are then replaced with the offspring of good solutions.
3. Genetic algorithms help in optimizing the solutions to any particular problem.
4. The whole process of genetic algorithms is a computer program simulation in which the attributes of the problem & solution are treated as the attributes of the Darwinian theory. The basic processes which are involved in genetic algorithms are as follows:

    - A population of solutions is built to any particular problem. The elements of the population compete with each other to find out the fittest one.
    - The elements of the population that are fit are only allowed to create offspring (better solutions).
    - The genes from the fittest parents (solutions) create a better offspring. Thus, future solutions will be better and sustainable.
5. The working of a **genetic algorithm in AI** is as follows:

    - The components of the population, i.e., elements, are termed as genes in **genetic algorithms in AI**. These genes form an individual in the population (also termed as a chromosome).
    - A search space is created in which all the individuals are accumulated. All the individuals are coded within a finite length in the search space.
    - Each individual in the search space (population) is given a fitness score, which tells its ability to compete with other individuals.
    - All the individuals with their respective fitness scores are sought & maintained by the genetic algorithm & the individuals with high fitness scores are given a chance to reproduce.
    - The new offspring are having better ‘partial solutions’ as compared to their parents. Genetic algorithms also keep the space of the search space dynamic for accumulating the new solutions (offspring).
    - This process is repeated until the offspring’s do not have any new attributes/features than their parents (convergence). The population converges at the end, and only the fittest solutions remain along with their offspring (better solutions). The fitness score of new individuals in the population (offspring) are also calculated.

Genetic operators

1. A genetic operator is an operator used in genetic algorithms to guide the algorithm towards a solution to a given problem. There are three main types of operators (mutation, crossover and selection), which must work in conjunction with one another in order for the algorithm to be successful.
2. Genetic operators are used to create and maintain genetic diversity (mutation operator), combine existing solutions (also known as chromosomes) into new solutions (crossover) and select between solutions (selection).
3. **Selection**:
   - Selection operators give preference to better solutions (chromosomes), allowing them to pass on their 'genes' to the next generation of the algorithm.
   - The best solutions are determined using some form of objective function (also known as a 'fitness function' in genetic algorithms), before being passed to the crossover operator.
   - The selection operator may also simply pass the best solutions from the current generation directly to the next generation without being mutated; this is known as elitism or elitist selection.
4. **Crossover**:
   - Crossover is the process of taking more than one parent solutions (chromosomes) and producing a child solution from them.
   - By recombining portions of good solutions, the genetic algorithm is more likely to create a better solution.
   - As with selection, there are a number of different methods for combining the parent solutions, including the edge recombination operator (ERO) and the 'cut and splice crossover' and 'uniform crossover' methods.
   - Crossover methods may be particularly suited to certain problems; the ERO is generally considered a good option for solving the travelling salesman problem.
5. **Mutation**:
   - The mutation operator encourages genetic diversity amongst solutions and attempts to prevent the genetic algorithm converging to a local minimum by stopping the solutions becoming too close to one another.
   - In mutating the current pool of solutions, a given solution may change entirely from the previous solution.
   - By mutating the solutions, a genetic algorithm can reach an improved solution solely through the mutation operator.
   - As with the crossover operator, the mutation method is usually chosen to match the representation of the solution within the chromosome.