What is alpha-beta pruning? What are the two commitments of logic? Define them.

Alpha-beta pruning is a modified version of the minimax algorithm. It is an optimization technique for the minimax algorithm.

The two-parameter can be defined as:

1. **Alpha:** The best choice (highest-value) we have found so far at any point along the path of Maximizer. The initial value of alpha is **-∞**.
2. **Beta:** The best choice (lowest-value) we have found so far at any point along the path of Minimizer. The initial value of beta is **+∞**.

The two commitments are:

1. **Law of Excluded Middle:** This principle states that for any proposition (statement), either it is true or its negation is true, and there is no third possibility. In simpler terms, it means something is either true or false, and there's no in-between or uncertainty.

2. **Law of Bivalence:** This principle builds upon the first and states that any proposition can only have one truth value at a time. It cannot be both true and false simultaneously. This principle ensures consistency and clarity in reasoning and argumentation, eliminating contradictions and ensuring clear distinctions between true and false.

Explain genetic algorithm as a local search.

Genetic Algorithms are adaptive heuristic search algorithms which simulate the process of natural selection. They are considered global optimization algorithms due to their ability to explore a large search space efficiently. However, they exhibit following characteristics of local search:

1. **Population-Based Search:** Genetic algorithms maintain a population of potential solutions rather than a single solution.

2. **Exploitation and Exploration:** Genetic algorithms balance exploitation and exploration.

3. **Crossover and Mutation:** Crossover and mutation operators in genetic algorithms introduce small changes to individuals, exploring local variations around the current solutions.

4. **Fitness Evaluation:** The fitness function guides the selection of individuals for reproduction.

5. **Selection Pressure:** Selection pressure influences the probability of selecting individuals with higher fitness.

Explain any two local search algorithms in detail.

1. **Hill Climbing Algorithm:**

- Start with an initial solution.
- Evaluate the objective function to determine the quality of the solution.
- Generate a set of candidate solutions by making small modifications to the current solution.
- Evaluate the objective function for each candidate solution.
- Select the candidate solution that improves the objective function the most.
- Repeat steps 3-5 until no further improvement can be made.

Advantage: Simple and efficient

Disadvantage: Gets stuck in local optima

2. **Local Beam Search Algorithm:**

- Start with k randomly generated solutions.
- Evaluate the objective function for each solution.
- Generate a set of candidate solutions by making small modifications to the current states.
- Evaluate the objective function for each candidate solution.
- Select the k best candidate solutions to become the new current states.
- Repeat steps 3-5 until a stopping criterion is met.

Advantage: Increased chance to discover global optimum

Disadvantage: Gets stuck in local optima in case of large search space

Explain Memory bounded heuristic search in detail.

Memory bounded heuristic search is used to overcome the memory requirements of simple heuristic search.

IDA* Algorithm:

Step 1: Set the root node as the current node, and find the f-score.
Step 2: Set the cost limit as a **threshold** for a node
Step 3: Expand the current node to its children and find f-scores.
Step 4: If for any node the **f-score > threshold**, prune that node and store it in the visited node list.
Step 5: If the Goal node is found then return the path from the start node to Goal node.
Step 6: If the Goal node is not found, then **repeat from step 2** by changing the threshold with the minimum pruned value from the visited node list.

RBFS Algorithm:

Step 1: If curr_state = goal state Return curr_state

Step 2: Expand children

Step 3: If empty return failure

Step 4: Do

for each c (child of the curr_node)

f[c] = maximum(g(c) + h(c), f[curr_node])

best cost = lowest of f-cost, i.e., f[c]

if (best cost >prev f-cost found) return failure

consider the alternative f-cost

go to step 1 with (best-cost node, min (f-cost, alternative f-cost))

Explain simulated annealing search in detail.

Simulated Annealing Algorithm:

- Start with an initial solution.
- Set the initial temperature to a high value.
- Repeat the following steps until the stopping criterion is met:
  - Generate a new solution by making a small modification to the current solution.
  - Evaluate the objective function of the new solution.
  - If the new solution improves the objective function, accept it as the new current solution.
  - If the new solution does not improve the objective function, accept it with a probability that depends on the difference between the objective function values of the current and new solutions and the current temperature.
  - Decrease the temperature according to a cooling schedule.
- Return the current solution as the final solution.

Advantage: Guaranteed convergence

Disadvantage: Computationally expensive

What are the components of a first order logic? What is the difference between the two quantifiers in the logics?

First-order-logic is an extension of propositional logic, which deals with simple statements and logical operators.

Parts of First-order-logic:

1. Syntax: Syntax represents the rules to write an expression

2. Semantics: Semantics represents techniques to evaluate an expression

Components of First Order Logic:

Element	Example	Meaning
Constant	1, 2, A, John, Mumbai, cat, ...	Values that cannot be changed
Variables	x, y, z, a, b, ...	Can take up any value and can also change
Predicates	Brother, Father, >, ...	Defines a relationship between its input terms
Function	sqrt, LeftLegOf, ...	Computes a defined relation of input term
Connectives	∧, ∨, ¬, ⇒, ⇔	Used to form complex sentences using atomic sentences
Equality	==	Relational operator that checks equality
Quantifier	∀, ∃	Imposes a quantity on the respective variable

Quantifiers are the symbols that permit to determine the scope of the variable in the logical expression. There are two types of quantifier:

- **Universal quantifier (∀):** Represents statements that are true for all elements in the domain of discourse. For example, "∀x (Human(x) → Mortal(x))" means "every human is mortal".
- **Existential quantifier (∃):** Represents statements that are true for at least one element in the domain of discourse. For example, "∃x (Dragon(x) ∧ Fire(x))" means "there exists a dragon that breathes fire".

Feature	Universal quantifier	Existential quantifier
Meaning	For all	There is at least one
Symbol	∀	∃
Connective	→	∧
Example	"∀x (Human(x) →Mortal(x))" means "every human is mortal"	"∃x (Dragon(x) ∧ Fire(x))" means "there exists at least one dragon that breathes fire"
Negation	¬(∀x P(x)) is equivalent to ∃x ¬P(x)	¬(∃x P(x)) is equivalent to ∀x ¬P(x)

What is synchronic and diachronic? What are causal rules?

Synchronic approach refers to the analysis of AI model at a specific point of time. Synchronic approaches are efficient for static tasks.

Diachronic approach involves studying the changes in AI model over time. Diachronic approaches are crucial for dynamic tasks or understanding changes over time.

Causal rules describe the cause-and-effect relationships between variables within a system. These rules express how changes in one variable causes changes in another variable.

Applications of causal rules:

1. **Machine Learning:** They help understand the cause-and-effect relationships between features and outcomes.

2. **Predictive Modelling:** They help understand the cause-and-effect relationships between features and outcomes.

What are diagnostic rules? What is a model based reasoning systems?

Diagnostic Rules are explicit, if-then statements used to diagnose problems based on observed symptoms. It consists of two parts:

- If part: Describes the symptoms.
- Then part: Concludes the cause of the problem based on the "if" conditions.

Example: "If the website is unresponsive, then it could be a server problem."

Model-Based Reasoning Systems diagnose problems by reasoning about the internal structure and behaviour of a system based on a formal model.

Components:

- Model: Represents the system's components, relationships, and expected behaviour.
- Observation process: Simulates how observations arise from the system's internal state.
- Reasoning mechanism: Analyses observed data and compares it with the model's predicted behaviour to pinpoint discrepancies and diagnose faults.

Example: Anomaly detection systems for cyber security.

What are the types of learning?

1. **Supervised Machine Learning:** The main aim of the supervised learning technique is to map the input variable(x) with the output variable(y). Applications: Risk Assessment, Fraud Detection, Spam filtering, etc.
2. **Unsupervised Machine Learning:** The main aim of the unsupervised learning algorithm is to group or categories the unsorted dataset according to the similarities, patterns, and differences. Applications: Network Analysis, Recommendation Systems, etc.

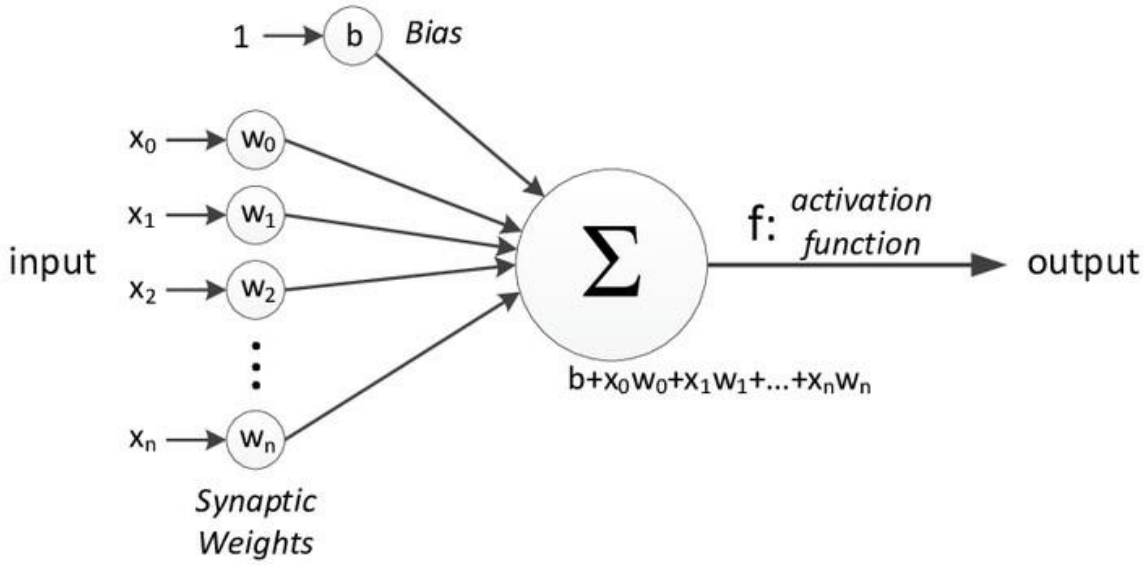
3. **Semi-Supervised Machine Learning:** The main aim of semi-supervised learning is to effectively use all the available data. It overcomes the drawbacks of supervised learning and unsupervised learning
4. **Reinforcement Learning:** Reinforcement learning works on a feedback-based process, in which an AI agent automatically explore its surrounding by hit & trial, learning from experiences, and improving its performance. Applications: Video games, Robotics, etc.

What is an ensemble learning? Give a simple mathematical model for a neuron.

Ensemble learning is a powerful machine learning technique that combines multiple models to improve overall performance. The idea is that different models can capture different aspects of the data, and combining their predictions leads to a more robust and accurate outcome.

Advantage: Improved accuracy
Disadvantage: Increased chances of redundancy

Mathematical model for neuron



What are the two choices for activation function? What are categories of neural network structures?

The activation function decides whether a neuron should be activated or not by calculating the weighted sum and further adding bias to it. Its purpose is to introduce non-linearity into the output of a neuron.

Choices for activation function:

1. **Linear Activation Function:** Maintains the linear relationship between input and output, making it unsuitable for complex tasks where non-linearity is crucial.

2. **Non-Linear Activation Function:** Introduces non-linearity, allowing the network to learn complex relationships between input and output. e.g. Sigmoid function, Tanh function, ReLU function, Softmax function, etc.

Categories of Neural Network:

1. **Perceptron:** The simplest form of neural network, with only one neuron. Classifies data into two categories

2. **Feedforward Neural Network:** Information flows unidirectionally from input to output through hidden layers.

3. **Multilayer Perceptron:** A specific type of feedforward neural network with multiple hidden layers.

4. **Convolutional Neural Network:** Specialized for processing image data.

5. **Radial Basis Functional Neural Network:** Uses radial basis functions instead of activation functions in hidden neurons.

6. **Recurrent Neural Network:** Processes sequential data with internal loops.

7. **Long Short-Term Memory (LSTM):** Enables RNNs to learn long-term dependencies in data.

8. **Sequence to Sequence Models:** Used for translating sequences of data, into another sequence.

9. **Modular Neural Network:** Composed of multiple smaller neural networks combined to perform a specific task.

Explain the following terms:

1) **Problem Space:** The problem space refers to the entire set of possible states that an algorithm can encounter while searching for a solution.

2) **Problem Instance:** A problem instance is a specific occurrence of a problem within the problem space.

3) **Problem Space graph:** A problem space graph is a graphical representation of the states in the problem space and the transitions between them.

4) **Depth of a problem:** The depth of a problem is the measure of the length of the shortest path from the initial state to the goal state.

5) **Space Complexity:** Space complexity is a measure of the amount of memory required by an algorithm to solve a problem.

6) **Time Complexity:** Time complexity is a measure of the amount of time or number of basic operations required by an algorithm to solve a problem.

7) **Admissibility:** It's a property of heuristic function. An admissible heuristic never overestimates the cost to reach the goal, ensuring that the algorithm remains effective and optimal.

8) **Branching Factor:** The branching factor is the average number of branches of a node in a search tree.

Compare the performance of Breadth First, Depth First, Bidirectional, Uniform Cost and Interactive Deepening methods for search in terms of Time, Space, Optimality, and Completeness.

Algorithm	Time Complexity	Space Complexity	Optimality	Completeness
BFS	$O(b^d)$	$O(b^d)$	Optimal for unit costs	Complete for finite spaces
DFS	$O(b^m)$	O(bm)	Not guaranteed to be optimal	Complete for finite spaces
Bidirectional	$O(b^{d/2})$	$O(b^{d/2})$	Optimal if costs are the same	Complete if BFS is complete
UCS	$O(b^{1 + c/x})$	$O(b^{1 + c/x})$	Optimal for arbitrary costs	Complete for non-negative costs
IDDFS	$O(b^d)$	O(bd)	Optimal for unit costs	Complete for finite spaces

where:

b: Branching Factor

d: Depth of the Shallowest Goal Node

m: Maximum Depth of the Search Tree

c: Cost of the Optimal Solution

x: Minimum Cost Between Any Two States

What are the disadvantages associated with Brute-Force Search Strategies?

1. It can be very **time consuming**
2. It can be very **resource consuming**, especially if the search space is large
3. It can sometimes **find sub-optimal solutions**, since it does not use any heuristics
4. It can be **vulnerable to local minima**
5. It can be **inefficient** for complex problems

What are the various methods for graph traversal? Discuss in brief.

1. **Breadth-First Search (BFS):** Explores all nodes at a given level before moving to the next level.
2. **Depth-First Search (DFS):** Explores one path as deep as possible before backtracking.
3. Bidirectional Search: Searches simultaneously from the start and goal nodes, reducing total search time.
4. **Uniform Cost Search:** Prioritizes exploring nodes with lower costs, guaranteeing the optimal path regardless of distance.
5. **Iterative Deepening Search (IDS):** Repeatedly applies DFS with increasing depth limits until the goal is found.
6. **A* Search***:* Combines greedy best-first search with an informed heuristic estimate of remaining cost.

Explain the functioning of probabilistic reasoning system.

Probabilistic reasoning is a way of knowledge representation where we apply the concept of probability to indicate the uncertainty in knowledge. In probabilistic reasoning, there are two ways to solve problems with uncertain knowledge:

- **Bayes' rule**

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

P(A|B) is known as **posterior**, which we need to calculate, and it will be read as Probability of hypothesis A when we have occurred an evidence B.

P(B|A) is called the **likelihood**, in which we consider that hypothesis is true, then we calculate the probability of evidence.

P(A) is called the **prior probability**, probability of hypothesis before considering the evidence

P(B) is called **marginal probability**, pure probability of an evidence.

$$P(A_i|B) = \frac{P(A_i)*P(B|A_i)}{\sum_{i=1}^{k} P(A_i)*P(B|A_i)}$$

Where A_1, A_2, A_3, ..., A_n is a set of mutually exclusive and exhaustive events.

- **Bayesian Statistics**

Bayesian statistics is a branch of statistics that uses probabilistic reasoning to analyse and interpret data. It provides a framework for making statistical inferences and estimating probabilities based on data and prior knowledge.

How chart-parsing algorithm resolve the problems of ambiguity? Give example.

Consider the statement "The large can can hold water."

Ambiguity Resolution: The algorithm explores two interpretations based on different parse trees:

1. "The large can (container) can hold water."

   - The first "can" is interpreted as a noun, representing a container.

2. "The large can (ability) can hold water."

   - The first "can" is interpreted as a verb, representing the ability to hold water.

The chart-parsing algorithm efficiently handles ambiguity by exploring multiple possible parse trees. The chart structure stores partial structures, and backtracking allows the algorithm to consider different interpretations of the sentence.

How the grammar is augmented when it is expressed using features? Explain.

An **augmented grammar** is any grammar whose productions are augmented with **conditions** expressed using features. Features may be associated with any nonterminal symbol in a derivation. A feature associated with a nonterminal symbol is shown following that nonterminal separated from it by a ".", e.g. A.COUNT is the COUNT feature associated with the nonterminal A. When the value is included with the feature, the feature and its value are bracketed and the dot omitted, as in A[COUNT 1]. If no ambiguity arises, the feature name may be omitted, as in A[1]. The values of the features may be **assigned** by using the assignment operator ":="

e.g. A.COUNT := B.COUNT - 1

Elaborate on the semantic augmentations for an English fragment including tense, quantification and pragmatic interpretation

Semantic augmentation involves enriching an English fragment's representation with additional information related to meaning.

1. Tense: Tense is a grammatical feature that indicates the time at which an action or event occurs. Example:

Original fragment: "She walks to the store."

Augmented with tense: "Present: She walks to the store."

2. Quantification: Quantification involves specifying the quantity of entities involved in an action or event. This can include quantifiers like "some," "all," "none," etc., and numerical expressions. Example:

Original fragment: "Cats sleep."

Augmented with quantification: "Some cats sleep."

3. Pragmatic Interpretation: Pragmatic interpretation involves considering contextual factors, speaker intentions, and implied meaning beyond the literal semantics of the words used. Example:

Original fragment: "John has a car."

Augmented with pragmatic interpretation: "John owns a car”

Explain Learning decision trees

Decision tree learning is a supervised learning approach used in statistics, data mining and machine learning. In this approach, a classification decision tree is used as a predictive model to draw conclusions about a set of observations.

How should an agent go about building a decision tree? Can rules be skips or reads be omitted? Why negation is considered as failure?

Building a decision tree:
Step-1: Begin the tree with the root node, says S, which contains the complete dataset.
Step-2: Find the best attribute in the dataset using Attribute Selection Measure (ASM).
Step-3: Divide the S into subsets that contains possible values for the best attributes.
Step-4: Generate the decision tree node, which contains the best attribute.
Step-5: Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes.

Skipping rules may lead to incomplete information and inaccurate decisions if the skipped rule was relevant. Skipping reads may miss crucial data needed for accurate tree construction.

Negation can be interpreted as failure in decision trees if the agent encounters a negative rule but cannot find a matching condition to activate it. This might indicate missing information or an incomplete knowledge base.

If a decision tree can be incrementally built from the top down by recursively selecting a feature to split, then what are the basis for selecting these features?

1. **Gini Impurity** is a score that evaluates how accurate a split is among the classified groups.

$$\text{Gini Impurity} = 1 - \sum p_i^2$$

Here,

- p_i is the proportion of elements in the set that belongs to the i^{th} category.

2. **Information gain** represents the difference in entropy before and after a split on a given attribute.

$$\text{Information Gain}(S,a) = \text{Entropy}(S) - \sum \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

where:

- *a* represents a specific attribute or class label
- *Entropy(S)* is the entropy of dataset, S
- |Sv|/ |S| represents the proportion of the values in S_v to the number of values in dataset, S
- *Entropy(S_v)* is the entropy of dataset, S_v

Write short note on Entropy

Entropy is a metric to measure the impurity in a given attribute. It specifies randomness in data.

$$\text{Entropy}(S) = -\sum p(c)\log_2 p(c)$$

where:

- S represents the data set that entropy is calculated
- c represents the classes in set, S
- p(c) represents the proportion of data points that belong to class c to the number of total data points in set, S