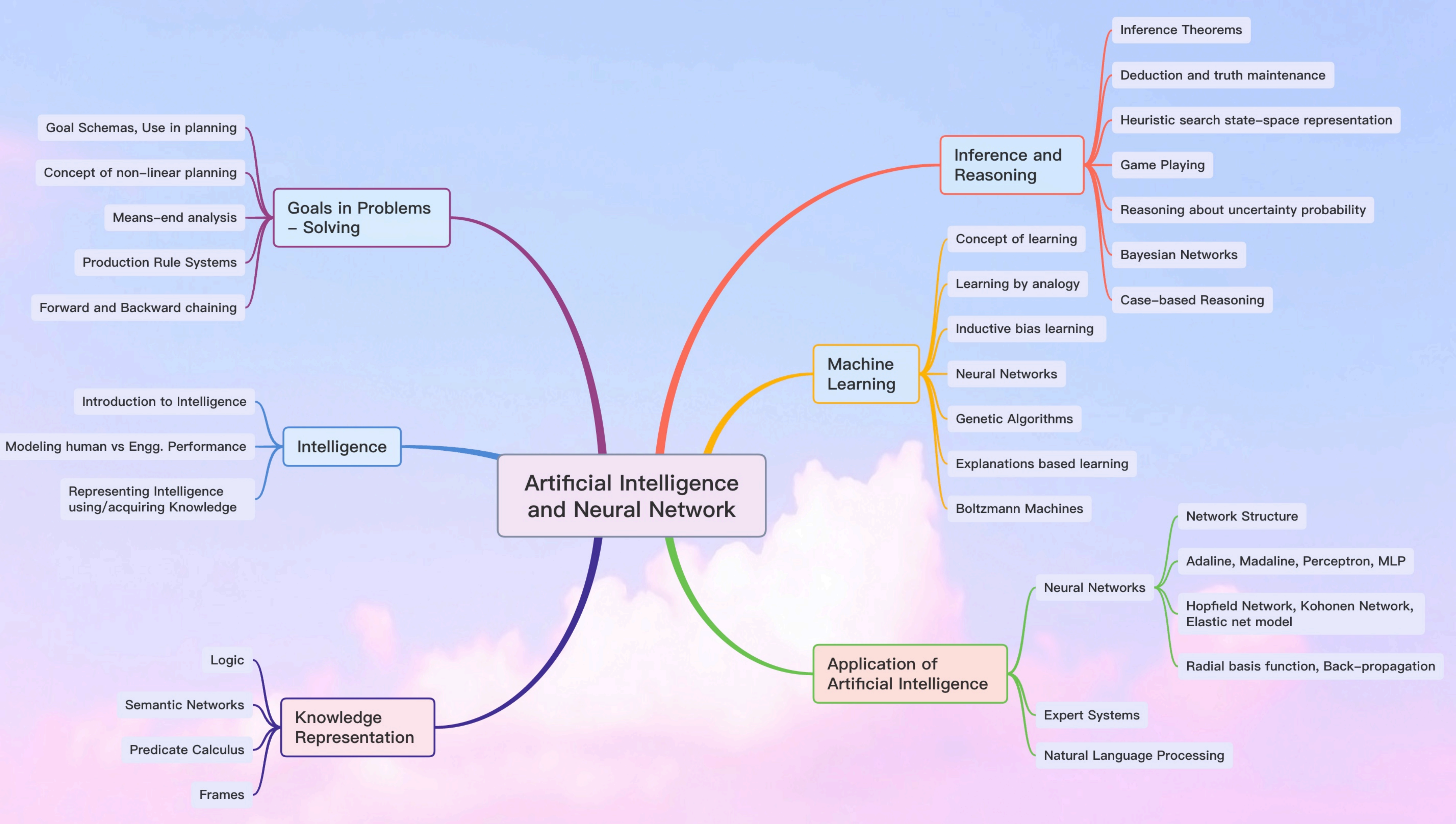
UNIT 1

Goals in Problem Solving


Artificial Intelligence and Neural Network
Goals in Problems – Solving
Goal Schemas, Use in planning
Concept of non-linear planning
Means-end analysis
Production Rule Systems
Forward and Backward chaining
Intelligence
Introduction to Intelligence
Modeling human vs Engg. Performance
Representing Intelligence using/acquiring Knowledge
Knowledge Representation
Logic
Semantic Networks
Predicate Calculus
Frames
Inference and Reasoning
Inference Theorems
Deduction and truth maintenance
Heuristic search state-space representation
Game Playing
Reasoning about uncertainty probability
Bayesian Networks
Case-based Reasoning
Machine Learning
Concept of learning
Learning by analogy
Inductive bias learning
Neural Networks
Genetic Algorithms
Explanations based learning
Boltzmann Machines
Application of Artificial Intelligence
Neural Networks
Network Structure
Adaline, Madaline, Perceptron, MLP
Hopfield Network, Kohonen Network, Elastic net model
Radial basis function, Back-propagation
Expert Systems
Natural Language Processing

Contents

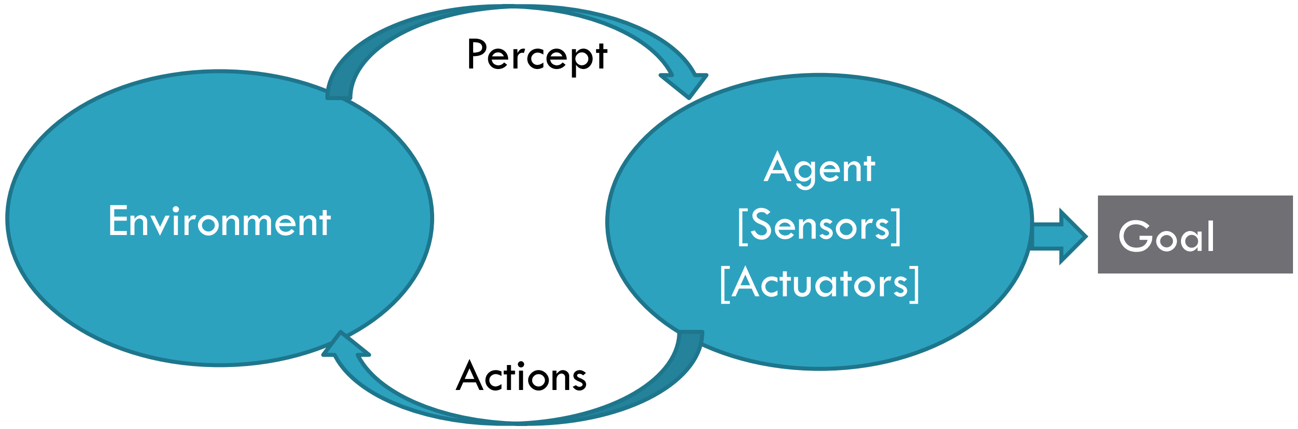
Goal Schemas Use in Planning

Goal: The state of affairs that a plan is intended to achieve and that (when achieved) terminates behavior intended to achieve it.

OR

The place designated at the end.

Goal Schemas Use in Planning



Given the goal, the agent wants to find the sequence of actions that will let him/her achieve the goal.

→Given current situation what will be the next step?

Goal Schemas Use in Planning

To build a system to solve a particular problem, we need to do four things:

- **Define the problem:** It includes precise specifications of what the initial situation(s) will be as well as what final situations constitute acceptable solutions to the problem
- **Analyse the problem:** Few very important aspects those having immense impact on the appropriateness of various possible techniques for problem solving is to be critically examined
- **Isolate and Represent the task knowledge:** Knowledge necessary to solve the problem must be identified, isolated and represented
- **Choose and apply the best technique:** Among the alternatives identify the best technique to solve the problem and apply it

Planning

- To find a sequence of actions that achieves a given goal

OR

- Given a set of operator instances (defining possible primitive actions by an agent), an initial state description, and a goal state description or predicate, the planning agent computes a plan.
- **Problem Solving Agents + Knowledge-based Agents = Planning Agents**
- **Plan:** A sequence of operator instances, such that "executing" them in the initial state will change the world to a state satisfying the goal state description. Goals are usually specified as a conjunction of goals to be achieved

Planning

Linear Planning: Works on one goal until completely solved before moving on to the next goal.

- Planning algorithm maintains goal stack.
- Implications:
  - No interleaving of goal achievement
  - Efficient search if goals do not interact (much)
- Advantages:
  - Reduced search space, since goals are solved one at a time
  - Advantageous if goals are (mainly) independent
  - Linear planning is sound
- Disadvantages:
  - Linear planning may produce suboptimal solutions (based on the number of operators in the plan)
  - Linear planning is incomplete.

Planning

Non Linear Planning :

- Use goal set instead of goal stack. Include in the search space all possible sub goal orderings.
- Handles goal interactions by interleaving.
- Advantages:
  - Non-linear planning is sound.
  - Non-linear planning is complete.
  - Non-linear planning may be optimal with respect to plan length (depending on search strategy employed)
- Disadvantages:
  - Larger search space, since all possible goal orderings may have to be considered.
- Somewhat more complex algorithm; more bookkeeping.

Means-End Analysis

- Reducing differences between current state and goal state
  - Stop when difference is 0 (no difference)

- Subgoals
  - Intermediate goals – not your final goal-state
  - Means-end analysis is also considered a way to break up a problem into pieces (subgoals)

Means-End Analysis

- Allows both backward and forward searching.
- This means we could solve major parts of a problem first and then return to smaller problems when assembling the final solution.
- GPS was the first AI program to exploit means-ends analysis.
- STRIPS (A robot Planner) is an advanced problem solver that incorporates means-ends analysis and other techniques.
- Very loosely the means-ends analysis algorithm is:
  - Until the goal is reached or no more procedures are available:
    - Describe the current state, the goal state and the differences between the two.
    - Use the difference the describe a procedure that will hopefully get nearer to goal.
    - Use the procedure and update current state.
  - If goal is reached then **success** otherwise **fail**.

Analogy

- Borrowing a solution already used to solve a similar problem
- Example problem
  - Patient has a tumor in location that makes it inoperable
  - One possibility is to use a high-powered beam to destroy the tumor from the outside
  - Problem: beam will also damage surrounding healthy tissue

Similar problem

- Evil king lives in a castle with his army
- Good king wants to destroy the evil king
- Good king amasses a huge army to defeat the evil king
- Problem: only narrow roads bordered by natural (immovable) obstacles lead to evil king's castle; no single road can hold the entire good king's army

Solution to castle problem

- Good king divides the army into smaller divisions
  - Each division goes down a separate road at the same time
  - All divisions meet at the castle simultaneously to overtake the evil king and his army

Solution to tumor problem

- Divide beam into weaker beams
- Send all weak beams into body simultaneously but from different angles
- Combined strength of beams at tumor site will destroy tumor

Production Rule System

- A production system consists of a set of rules, each consisting of a left side (a pattern), that determines the applicability of the rule and a right side that describes the operation to be performed id the rule is applied.
- Example: [A, clean] → move right

  Pattern Action
- Have one or more knowledge base (database) that contain whatever information is appropriate for the particular task

Production Rule System

- A **Control Strategy** that specifies the order in which the rules will be selected and a way of resolving the conflicts that arise when several rules matched at once
  - The first requirement of a good control strategy is that it cause motion

    No motion action: filling the jug each time
  - The second requirement of a good control strategy is that it should be systematic

    No systematic action: random selection action
  - The systematic control strategy is the good search technique
- Have a rule applier

Defining Problems as a State Space Search: An Example – A Water Jug Problem

Problem

- You are given two jugs, a 4 litre one and a 3 litre one. None of them have any measuring markers on it. There is a pump that can be used to fill the jugs with water. How can you get exactly a 2 litres of water in to the 4 litre jug?

Defining Problems as a State Space Search: An Example – A Water Jug Problem

State Space

- Set of ordered pair of integers (x, y), such that x= 0, 1, 2, 3, or 4 and y= 0, 1, 2, or 3, where x represents the quantity of water in 4 litre jug and y represents the quantity of water in 3 litre jug
- The start state is (0, 0)
- The final state is (2, n)

Defining Problems as a State Space Search: An Example – A Water Jug Problem → Rules

1. (x, y) if x<4 → (4, y)
   Fill the 4 litre jug
2. (x, y) if y<3 → (x, 3)
   Fill the 3 litre jug
3. (x, y) if y>0 → (x, 0)
   Empty the 3 litre jug
4. (x, y) if x>0 → (0, y)
   Empty the 4 litre jug
5. (x, y) if x>o → (x-d, y)
   Put some water out of 4 l jug
6. (x, y) if y>0 → (x, y-d)
   Put some water out of 3 l jug
7. (x, y) if x+y>=4 & y>0 →(4, y-(4-x))
   Put water from 3l jug into 4l jug to fill it
8. (x, y) if x+y>=3 & x>0 →(x-(3-y), 3)
   Put water from 4l jug into 3l jug to fill it
9. (x, y) if x+y<3 & x>0 →(0, x+y)
   Put all the water from 4l jug in to 3l jug
10. (x, y) if x+y<4 & y>0 →(x+y, 0)
    Put all the water from 3l jug in to 4l jug

Defining Problems as a State Space Search: An Example – A Water Jug Problem → Solution

Water Quantity in 4 l Jug	Water Quantity in 3 l Jug	Rule Applied
0	0	2
0	3	10
3	0	2
3	3	7
4	2	4
0	2	10
2	0	

- A farmer has to cross a river with his Fox, Goose, and Grain. Each trip, his boat can only carry himself and one of his possessions. How can he cross the rivers if an unguarded fox eats the goose and an unguarded goose the grain.
- Start state [Left(Fa, Go, Fr, Fo) | Right( )]
- Goal state [Left( ) | Right(Fa, Go, Fr, Fo)]

Forward and Backward Chaining

Inference Scheme:

Inference engines in ES are responsible for deciding how the knowledge data in the KB should be used. They are responsible for the control and execution of the reasoning strategies used by ES. Backward chaining and forward chaining are strategies used to specify how rules contained in a knowledge base rule system are to be executed.

Forward and Backward Chaining

IF the weather is rainy

AND the distance is >= 100 kilometers

THEN transportation is by car (**1**)

IF transportation is by car

THEN passenger insurance will be considered (**2**)

IF passenger insurance is considered

THEN transportation insurance cost = Rs 10,000 (**3**)

Forward and Backward chaining

Backward Chaining: suppose we want to establish the fact that “transportation insurance cost = 10000” assuming we know only that “the weather is rainy” and “the distance is 150 km”. Backward chaining works backward from the conclusion:

Is this fact known? → No
Can it be obtained from the rule? → Yes, from rule 3
Which fact needs to be known? → “Passenger insurance is considered”
Is this fact known? → No
Can it be obtained from the rule? → yes, from rule 2
Which fact need to know? → “Transportation by car”
Is this fact known? → No
Can it be obtained from the rule? → yes, from rule 1
Which fact need to be known? → “The weather is rainy” and
“Distance >=100 km”
Are these facts known? → Yes, “The weather is rainy” and
“distance is 150 km”
Therefore, it is true that “transportation is by car”.
Therefore, it is true that “passengers insurance is considered”.
Therefore, it is true that transportation insurance cost = 10000”.

We started with the fact we wanted to prove and tried to establish all the facts needed to reach the goal. This reasoning method is called backward chaining. Backward chaining is applied when a goal or hypothesis is chosen as the starting point for problem solving. Backward chaining is also known as goal-directed, top-down or consequence driven.

Forward and Backward chaining

Forward Chaining: this approach goes forward from starting point, via the conclusion generated at each step.

Suppose we want to prove that "transportation insurance cost = 10000" assuming we know only that "the weather is rainy" and "the distance is 150 km"

Forward Chaining: this approach goes forward from starting point, via the conclusion generated at each step. Suppose we want to prove that "transportation insurance cost = 10000" assuming we know only that "the weather is rainy" and "the distance is 150 km"

Is the fact known? → No
Which fact do we know? → "The weather is rainy." And
"The distance >= 100 km."
Which fact follow from it? → "Transportation is by car." Rule (1)
Is this what we want to prove? → No
What fact follow from it? → "Passenger insurance will be considered."
Rule (2)
Is this what we want to prove? → No
What fact follow from it? → "Transportation insurance cost = 10000."
Rule (3)
Is this what we want to prove? → Yes.

Mycin-Style Probabilities and its Applications

- An expert system for treating blood infections
- MYCIN would attempt to diagnose patients based on reported symptoms and medical test results
- Could ask some more information and lab test results for diagnosis
- It would recommend a course of treatment, if requested MYCIN would explain the reasoning that lead to its diagnosis and recommendations
- Uses about 500 production rules
- MYCIN operated at roughly the same level of competence as human specialists in blood infections
- Uses backward chaining for reasoning

References

- Elaine Rich and Kevin Knight, Artificial Intelligence, 2e
- Russel & Norvig, 3e