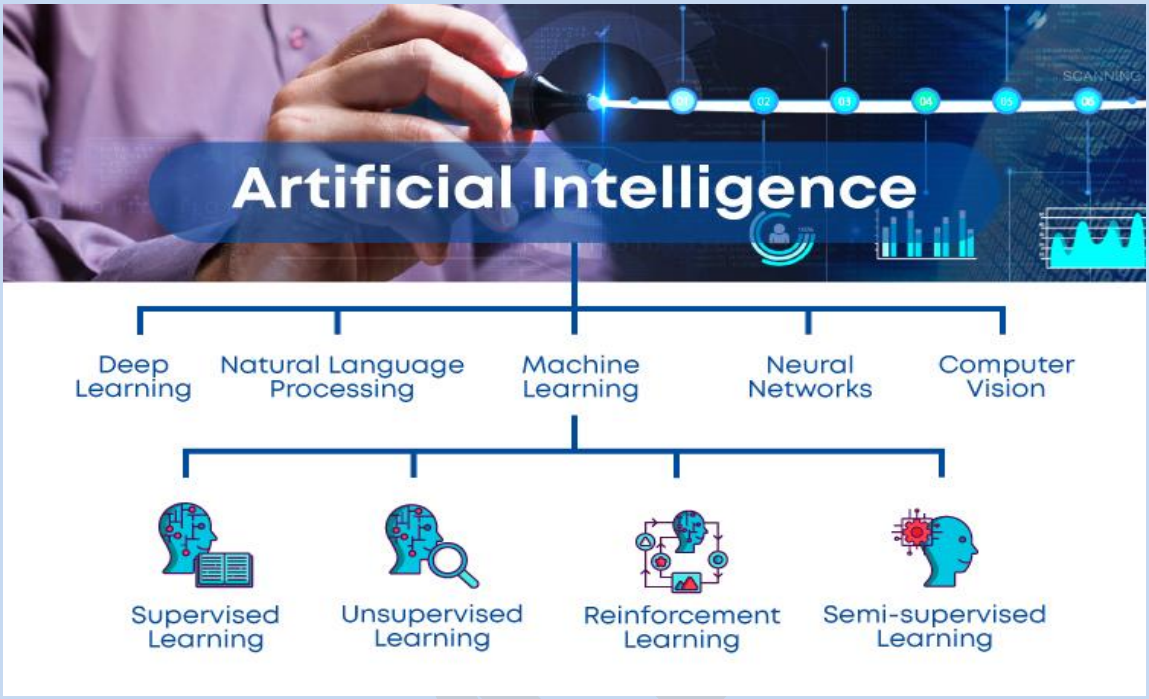


1. What is Artificial Intelligence?

Definition: Artificial Intelligence is the study of how to make computers do things, which, at the moment, people do better.
According to the father of Artificial Intelligence**, John McCarthy**, it is "The science and engineering of making intelligent machines, especially intelligent computer programs".
Artificial Intelligence is a way of making a computer, a **computer-controlled robot**, or a **software think intelligently**, in the similar manner the intelligent humans think

DATA : Unformatted information and Raw Facts – Numeric or Categorical

INFORMATION: It is the result of processing and organizing the data in response to a specific field

KNOWLEDGE : Understanding the problem statement and its domain (What to do?) E.g., Finance, Manufacturing, Sales, Forecasting etc.

INTELLIGENCE : The ability to acquire and apply the knowledge and skills to the machines

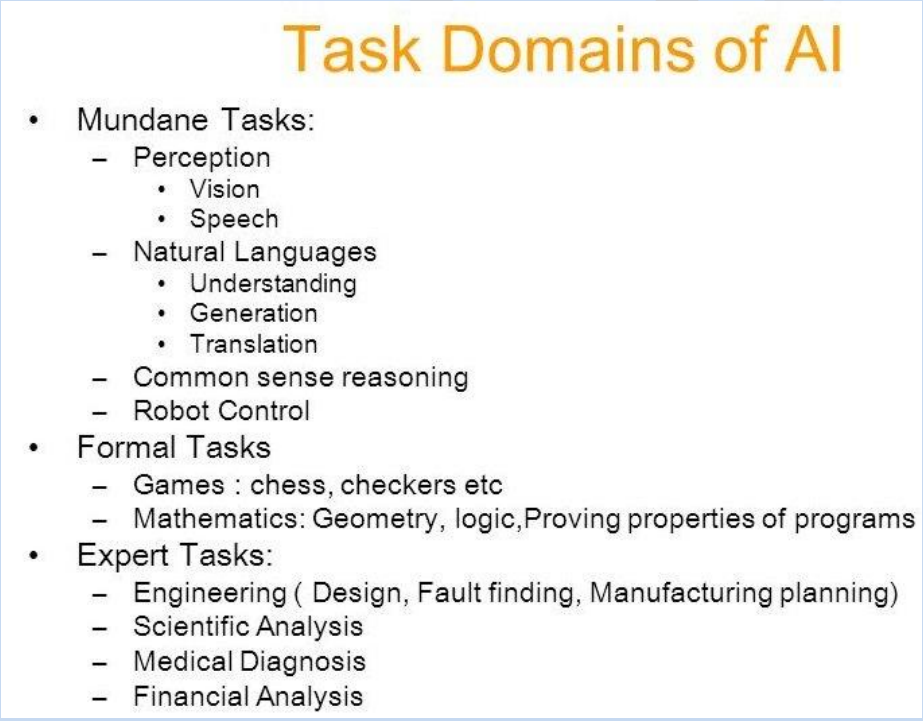
2. **Problems associated with Artificial Intelligence?**

Common Sense Reading: AI focuses on reasoning about physical objects and their relationships with each other. It also reasons the actions and other consequences

Game Playing and Theorem Proving: These share the properties that people who do well overall, and are intelligent

General Problem Solver (GPS): It is the problem of applying symbolic manipulators to the logical expressions

3. **Task/Application Domains of Artificial Intelligence?**



4. **What is Physical System Hypothesis?**

Artificial Intelligence practitioners define a physical system as

1. Symbols
2. Expressions
3. Symbol Structures
4. Systems

It contains the process so that it can act upon the symbols and Expressions
Computers provide the perfect medium for the experimentation since they can be programmed to simulate physical system symbol that we like

5. What are Artificial Intelligence Technique (Important)?

There are appropriate techniques to solve multi-domain related problems.

1. The knowledge captures Generalizations
2. It can be understood by the people
3. It can be easily modified to correct the errors and reflect changes in the world
4. It can be used in many situations even if it is not totally accurate or complete
5. It can narrow down the range of possibilities for error

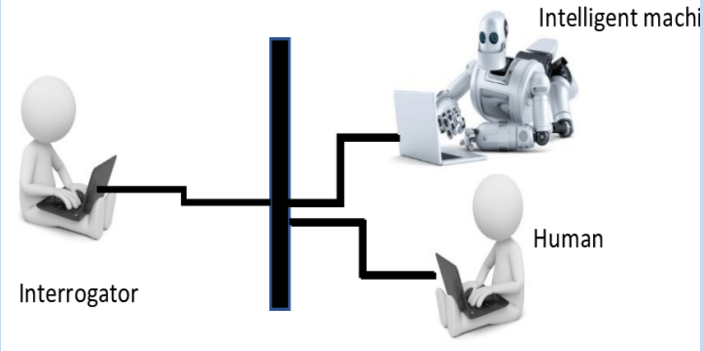
Important AI Techniques
Search: Provides a problem-solving approach where there is no direction or approach available

Use of Knowledge: Provides a way of Solving Complex Problems by exploring the structures of objects that are involved

Abstraction: Provides a way to show the necessary and relevant features by hiding the unimportant ones

6. Explain Turing Test?

In 1950, Alan Turing proposed a method for determining whether a machine can artificially think



One person A plays the role of interrogator. Questions are asked by him to both computer and other human, and simultaneous answers are received from both

The trick is to determine the difference between Human answer and Computer answer

IF THE INTERROGATOR FAILS TO DETERMINE THE DIFFERENCE, IT MEANS THE MACHINE HAS PASSED THE TEST

Less Desirable Properties of Knowledge

1. It is huge
2. It is difficult to characterize correctly
3. It is constantly varying
4. It differs from data to data by being organized in a way that it corresponds to its applications . It is Complicated

7. What are the Steps to Solve Problems in Artificial Intelligence

Steps to solve the problem of building an Artificial Intelligence system:

1. Define the problem accurately including detailed specification and determine the suitable solution
2. Scrutinize(double-check) the problem carefully as some features may have effect on the chosen method of solution(Algorithm)
3. Segregate(separate) the background knowledge needed in the solution of the problem
4. Choose the method having highest accuracy of solution

8. **Explain Tic-Tac-Toe problem & How it can be solved using AI Techniques (Important)?**

A. Tic-Tac-Toe problem

1	**2**	**3**
4	**5**	**6**
7	**8**	**9**

An element contains the value **0 for blank**, **1 for X** and **2 for O**. A **MOVETABLE** vector consists of **19,683** elements (3^9) and is needed where each element is a nine element vector.

The algorithm makes moves by pursuing the following:

1. View the vector as a ternary number. Convert it to a decimal number.
2. Use the decimal number as an index in MOVETABLE and access the vector.
3. Set BOARD to this vector indicating how the board looks after the move.

This approach is capable in time but it has several disadvantages. It takes more space and requires stunning effort to calculate the decimal numbers. This method is specific to this game and cannot be completed.

B. The second approach

The structure of the data is as before but we use **2 for a blank**, **3 for an X** and **5 for an O**.
A variable called **TURN** indicates 1 for the first move and 9 for the last.

The algorithm consists of three actions:
***MAKE2** which returns 5 if the centre square is blank; otherwise it returns any blank non-corner square, i.e. 2, 4, 6 or 8.
***POSSWIN (p)** returns 0 if player p cannot win on the next move
and otherwise returns the number of the square that gives a winning move.
It checks each line using products **3*3*2 = 18** gives a win for **X**, **5*5*2=50** gives a win

for **O**, and the winning move is the holder of the blank.
***GO (n)** makes a move to square n setting **BOARD[n] to 3 or 5**.

This algorithm is more involved and takes longer but it is more efficient in storage which compensates for its longer time. It depends on the programmer's skill.

C. The final approach

The structure of the data consists of **BOARD** which contains a nine element vector, a list of board positions that could result from the next move and a number representing an estimation of how the board position leads to an ultimate win for the player to move.
This algorithm looks ahead to make a decision on the next move by deciding which the most promising move or the most suitable move at any stage would be and selects the same.

Consider all possible moves and replies that the program can make.

Continue this process for as long as time permits until a winner emerges, and then choose the move that leads to the computer program winning, if possible in the shortest time.

Actually this is most difficult to program by a good limit but it is as far that the technique can be extended to in any game. This method makes relatively fewer loads on the programmer in terms of the game technique but the overall game strategy must be known to the adviser.

9. What is Question Answering?

Let us consider Question Answering systems that accept input in English and provide answers also in English.

For example, consider the following situation:
Text
Rani went shopping for a new Coat. She found a red one she really liked.
When she got home, she found that it went perfectly with her favourite dress.
Question
1. What did Rani go shopping for?
2. What did Rani find that she liked?
3. Did Rani buy anything?
Method 1
Data Structures

A set of templates that match common questions and produce patterns used to match against inputs.

Templates and patterns are used so that a template that matches a given question is associated with the corresponding pattern to find the answer in the input text. For example, the template who did x y generates x y z if a match occurs and z is the answer to the question. The given text and the question are both stored as strings.

Algorithm

Answering a question requires the following four steps to be followed:

Compare the template against the questions and store all successful matches to produce a set of text patterns. Pass these text patterns through a substitution process to change the person or voice and produce an expanded set of text patterns.

Apply each of these patterns to the text; collect all the answers and then print the answers.

Example

In question 1 we use the template WHAT DID X Y which generates Rani go shopping for z and after substitution we get Rani goes shopping for z and Rani went shopping for z giving z [equivalence] a new coat

In question 2 we need a very large number of templates and also a scheme to allow the insertion of ‘find’ before ‘that she liked’; the insertion of ‘really’ in the text; and the substitution of ‘she’ for ‘Rani’ gives the answer ‘a red one’.

Question 3 cannot be answered.

Comments

This is a very primitive approach basically not matching the criteria we set for intelligence and worse than that, used in the game. Surprisingly this type of technique was actually used in ELIZA which will be considered later in the course.

Method 2

Data Structures

Take, for example sentence:

‘She found a red one she really liked’.

Event2	**Event2**
instance: finding	instance: liking
tense: past	tense: past
agent: Rani	modifier: much
object: Thing1	object: Thing1

Thing1
instance: coat
colour: red

The question is stored in two forms: as input and in the above form.

Algorithm

*Convert the question to a structured form using English know how, then use a marker to indicate the substring (like 'who' or 'what') of the structure, that should be returned as an answer. If a slot and filler system is used a special marker can be placed in more than one slot.
*The answer appears by matching this structured form against the structured text.
*The structured form is matched against the text and the requested segments of the question are returned.

Method 3

Data Structures

World model contains knowledge about objects, actions and situations that are described in the input text. This structure is used to create integrated text from input text.

Algorithm

Convert the question to a structured form using both the knowledge contained in Method 2 and the World model, generating even more possible structures, since even more knowledge is being used.
Sometimes filters are introduced to prune the possible answers.
To answer a question, the scheme followed is: Convert the question to a structured form as before but use the world model to resolve any ambiguities that may occur. The structured form is matched against the text and the requested segments of the question are returned.

10. LEVEL OF THE AI MODEL?

1. To test psychological theories of human performance. Ex. PARRY [Colby, 1975] – a program to simulate the conversational behavior of a paranoid person.
2. To enable computers to understand human reasoning – for example, programs that answer questions based upon newspaper articles indicating human behavior.
3. To enable people to understand computer reasoning. Some people are reluctant to accept computer results unless they understand the mechanisms involved in arriving at the

results.

4. To exploit the knowledge gained by people who are best at gathering information. This persuaded the earlier workers to simulate human behavior in the SB part of AISB simulated behavior. Examples of this type of approach led to GPS (General Problem Solver)

Lab Program

1. **Implement the non-parametric Locally Weighted Regression algorithm in order to fit data points. Select appropriate data set for your experiment and draw graphs.**

What is Regression?

Regression analysis is a statistical method that helps us to analyse and understand the relationship between two or more variables of interest. The process that is adapted to perform regression analysis helps to understand which factors are important, which factors can be ignored and how they are influencing each other.

Regression is used for continuous values

● Dependent Variable: This is the variable that we are trying to understand or forecast.

● Independent Variable: These are factors that influence the analysis or target variable and provide us with information regarding the relationship of the variables with the target variable

What is Locally weighted Linear Regression?

● Locally weighted linear regression is a non-parametric algorithm

● The model does not learn a fixed set of parameters as is done in ordinary linear regression.

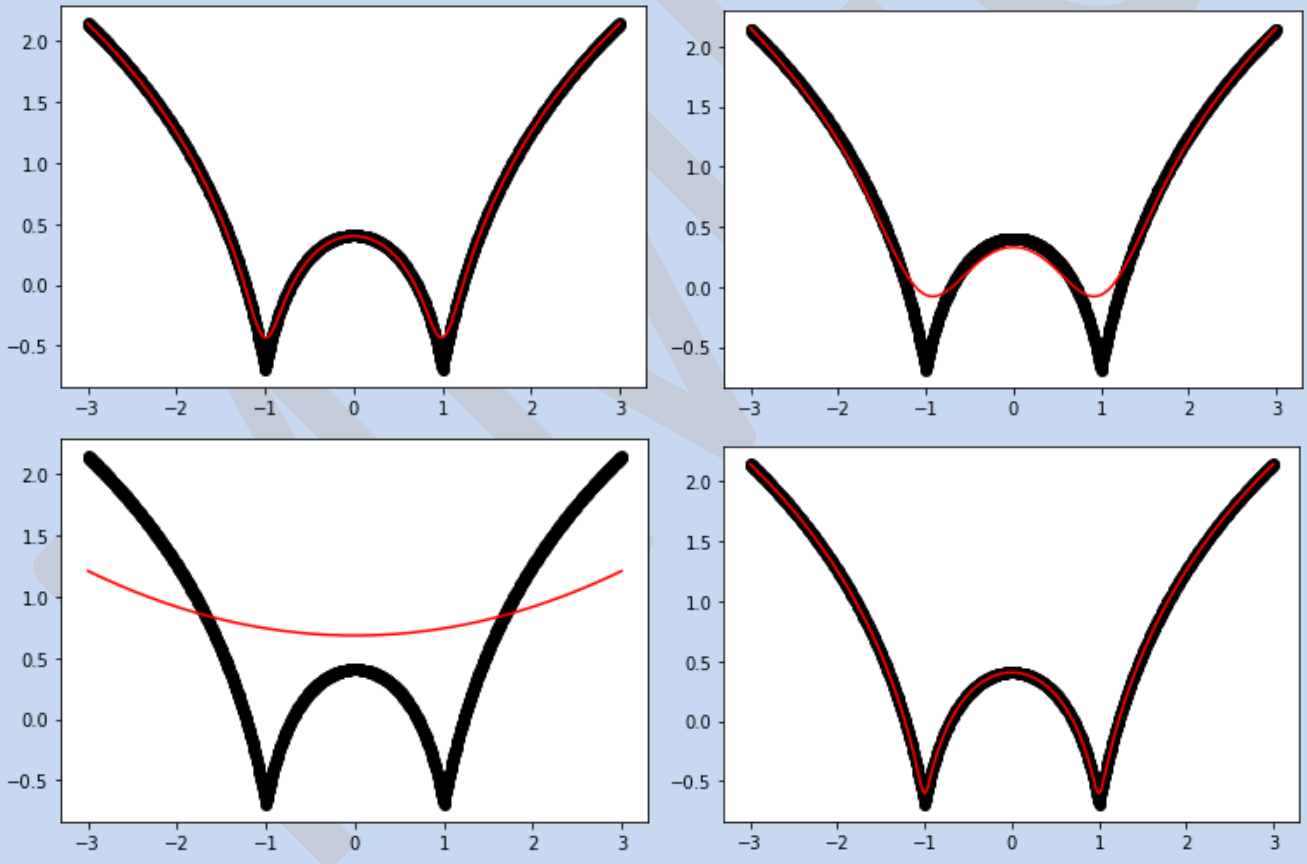
● Rather parameters theta are computed individually for each query point x.

CODE

```
import numpy as np
import matplotlib.pyplot as plt

def local_regression(x0, X, Y, tau):
    x0 = [1, x0]
    X = [[1, i] for i in X]
    X = np.asarray(X)
    xw = (X.T) * np.exp(np.sum((X - x0) ** 2, axis=1) / (-2 * tau))
    beta = np.linalg.pinv(xw @ X) @ xw @ Y @ x0
    return beta

def draw(tau):
    prediction = [local_regression(x0, X, Y, tau) for x0 in domain]
    plt.plot(X, Y, 'o', color='black')
    plt.plot(domain, prediction, color='red')
```

```
    plt.show()

X = np.linspace(-3, 3, num=1000)
domain = X
Y = np.log(np.abs(X ** 2 - 1) + .5)

draw(10)
draw(0.1)
draw(0.01)
draw(0.001)
```

Output



Here's a step-by-step explanation of the code:

1. Import necessary libraries: **numpy** for numerical operations and **matplotlib.pyplot** for data visualization.
2. Define the **local_regression** function:
    - This function takes four parameters:
        - **x0**: The point at which we want to estimate the regression.
        - **X**: The input features (independent variables).
        - **Y**: The corresponding target values (dependent variable).
        - **tau**: The bandwidth parameter that controls the width of the neighborhood for local regression.
    - The function first prepares the input data for regression by appending a constant 1 to each feature (for the intercept term) and exponentiating part of the weighted distance calculation.
    - It calculates a weighted linear regression using the specified bandwidth (**tau**) and returns the coefficients of the linear model.
3. Define the **draw** function:
    - This function takes one parameter, **tau**, and is responsible for drawing the local regression line based on the given **tau**.
    - It calls the **local_regression** function for each data point in the **domain** (X-axis values) to predict the corresponding Y-values.
    - Then, it plots the original data points (**X** and **Y**) as black dots and the predicted values as a red line.
4. Generate the dataset:
    - Generate **X** values (input features) using **numpy.linspace** to create 1000 evenly spaced values from -3 to 3.
    - Define the **domain** as the same as **X**.
    - Generate **Y** values (target values) based on a mathematical function: **np.log(np.abs(X ** 2 - 1) + .5)**. This function creates a curve.
5. Call the **draw** function four times with different **tau** values:
    - **draw(10)**: Draws the local regression line with a bandwidth of 10.
    - **draw(0.1)**: Draws the local regression line with a bandwidth of 0.1.
    - **draw(0.01)**: Draws the local regression line with a bandwidth of 0.01.
    - **draw(0.001)**: Draws the local regression line with a bandwidth of 0.001.

I. ASSIGNMENT QUESTIONS (25-Sept-2023)

1. **What is Artificial Intelligence?**
   a. **Data, Information, Knowledge, Intelligence?**
   b. **Explain Turning Test?**
2. **Explain Task Domain/Application in AI?**
   a. **What are Artificial Intelligence Technique?**
3. **Explain Tic-Tac-Toe problem & How it can be solved using AI Techniques?**
4. **What is Question Answering & explain all three methods?**
5. **<u>Lab Program</u>: Implement the non-parametric Locally Weighted Regression algorithm in order to fit data points. Select appropriate data set for your experiment and draw graphs.**

MNEC