Unit 1: Overview of Artificial Intelligence

Objective of this Unit

Familiarize students with the basic concepts of artificial Intelligence and its applications.

Outcome of this Unit

After completion of this unit, student will be able to understand strong AI and weak AI and identify problems applicable to AI

INDEX

1.0 Introduction

The term "**Artificial**" means synthetic (i.e., man-made) and "**Intelligence** is the cognitive ability of an individual to learn from experience, to reason well, to remember important information, and to cope with the demands of daily living"

1. How do you decide if someone (something?) is intelligent?
2. Are animals intelligent?
3. If animals are intelligent, how do you measure their intelligence?

Collective intelligence in animals, like how ants work together to solve complex problems, and how certain animals, like dolphins, show signs of self-awareness, memory, and problem-solving skills. Dolphins can recognize themselves in mirrors, use tools, and even coordinate their breathing in unique ways.

This leads to questions about machine intelligence. Humans have different opinions: some think machines can't truly think because they are just circuits, while others believe machines can be more "intelligent" due to their processing speed. This raises the question: Can a machine ever truly "think" or be "intelligent" like a living being?

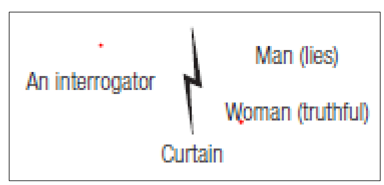
Dolphins can control their breathing voluntarily, which means they have more control over their brain function compared to humans. One half of a dolphin's brain can rest at a time, allowing them to sleep with one side of the brain active while the other rests.

Dolphins also pass the mirror test, which means they can recognize their own reflection in a mirror. This shows they have self-awareness, as they can identify that the reflection is actually of themselves.

"Dolphins use their snouts" means that dolphins use the front part of their face, specifically the long, beak-like part of their nose, to interact with objects or their environment. Dolphins have adapted to use their snouts for various purposes, such as foraging for food, digging in the sand, or even holding tools like sponges to protect their snouts while searching on the ocean floor.

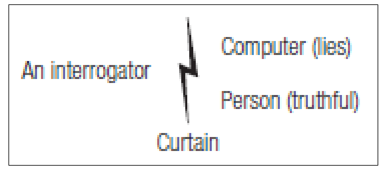
1.1 Turing Test Alan Turing proposed two versions of a game called the **imitation game** to study thinking and intelligence.

First Version: Distinguishing Gender



- A person, called the **interrogator**, sits in a room separated by a curtain. On the other side is either a man or a woman.
- The interrogator's job is to figure out the gender of the person behind the curtain by asking questions.
- The game assumes that the man may lie to trick the interrogator, while the woman will always answer honestly.
- To avoid giving hints through voice, communication happens through a computer or written responses, so only the answers matter. If the man is able to fool the interrogator into thinking he's a woman, he "wins" the game.

Second Version: Distinguishing a Human from a Computer



In the second version (the Turing Test we know today):

- The **interrogator** again asks questions to figure out who's behind the curtain: a human or a computer.
- Here, the **computer tries to act like a human** and might sometimes lie, while the human remains truthful.
- If the computer successfully convinces the interrogator that it's human, it is considered to "pass" the test, which means it is seen as "intelligent."

Turing's main idea was that intelligence could mean behaving like a human, even if it doesn't mean thinking exactly like a human. So, if a computer can consistently convince people that it's human, it could be thought of as having human-like intelligence. Turing believed this test would need to be repeated multiple times to ensure that the computer truly acts human-like over time.

To make the Turing Test effective, it's important to ask questions that help us understand if a computer can truly imitate human thinking. Some examples of questions and challenges in choosing them are:

1. **Avoiding Calculation Questions**: For example, asking a hard math question like "What is the square root of 1,000,017?" wouldn't work well. Computers can calculate these answers instantly. If a computer wanted to trick someone, it would pretend to take longer to answer or even make a mistake, because it knows people aren't usually fast at these calculations.
2. **Asking About the Weather**: You might think asking "What's the weather right now?" would stump a computer, assuming it can't look outside. But computers connected to the internet can easily check a weather website and respond with the current conditions. So, this question wouldn't reveal much about the computer's intelligence.
3. **Asking Emotional Questions**: Questions like "Are you afraid of dying?" or "What does it feel like to be in love?" can be difficult for a computer, since they involve human emotions. While emotions are part of human experience, they might not be the best way to judge intelligence itself.

Objections to Machine Intelligence

In his original paper, Turing anticipated several criticisms about machine intelligence:

- **Head-in-the-Sand Objection**: Some people believe that human thinking is unique and gives humans a special status. They think admitting that computers could think would threaten this uniqueness. Turing felt this concern shouldn't stop exploration of machine intelligence.
- **Theological Objection**: Some argued that only humans have souls, and thinking is tied to the soul. Creating thinking machines, they argued, would interfere with divine power. Turing countered that building machines might be seen as preparing them for a soul, as part of a higher plan.

- **Lady Lovelace's Objection**: Ada Lovelace, an early programmer, believed computers can only do what they are programmed for and can never truly "surprise" us. Turing argued that computers often behave in unexpected ways and that people who hold this view assume that humans can foresee all outcomes, which is not always true.

In summary, Turing believed that these objections didn't truly challenge the idea that machines could one day exhibit human-like intelligence.

Ned Block's criticism of the Turing Test argues that a computer could pass it by simply using stored responses rather than actually thinking or understanding.

Here's how his argument works:

- Every question and answer can be represented as a sequence of 0s and 1s inside a computer (using ASCII code).
- A Turing Test usually involves about 50 questions and answers, lasting around an hour. All possible questions and answers could be stored in a huge database.
- If a computer had access to this massive database, it could simply look up the correct answer for any question it's asked, without needing to think or understand.
- This means a computer might pass the Turing Test by relying on pre-programmed responses rather than true intelligence.

Block's main point is that, if passing the Turing Test only requires looking up answers, we can't really say the computer is "intelligent." Instead, it's just following a mechanical process.

John Searle's "Chinese Room" argument criticizes the Turing Test by questioning whether following rules to give the right responses means real understanding.

Here's his thought experiment:

1. Imagine someone who doesn't know Chinese is locked in a room and given a rulebook.
2. They receive questions in Chinese (which look like unfamiliar symbols) and use the rulebook to pick appropriate responses in Chinese characters.
3. From outside the room, the responses look correct, as if the person understands Chinese. But the person inside doesn't actually understand any Chinese—they're just following instructions.

Searle argues that this is similar to how a computer operates: it processes input and produces output based on programmed rules, but it doesn't understand what it's doing. So, he says, even if a computer can pass the Turing Test, that doesn't prove it's truly intelligent or understands anything.

Searle further extends this idea with a second example:

- Imagine a whole gym of people following specific instructions to pass around messages without understanding them. Just like the Chinese Room, this doesn't create real understanding.

In simple words, Searle believes that intelligence isn't just about correct responses; it requires understanding, which machines don't have.

The main argument from both Block and Searle against the Turing Test is that simply observing how something behaves doesn't tell us what's happening inside it. They believe that intelligence isn't just about responses; it requires understanding the inner workings of the mind or machine, not just its outward behavior.

But, there is an example that shows this is not always true: in the 19th century, physicist **Ernest Rutherford** used experiments to understand the structure of atoms. By bombarding gold foil with particles, he could deduce that atoms have a dense core (the nucleus) and are mostly empty space. This discovery was made through observing how particles behaved, without directly seeing inside the atom.

In conclusion, defining intelligence is challenging, and that's why Alan Turing developed the Turing Test. Turing believed that if something could pass the Turing Test, it would show it has the intelligence needed to handle complex challenges, similar to a human who is considered intelligent.

1.2 Strong and Weak AI

There are two main ways of thinking about artificial intelligence (AI):

1. **Weak AI (MIT approach)**:
   - This perspective focuses on any system that behaves intelligently, no matter how it achieves that behavior.
   - The important thing is that the system performs its tasks well. For example, if a robot can hear sounds and respond correctly, it is considered successful, regardless of whether it mimics human hearing.
   - Weak AI is mainly about solving problems and getting good results.
2. **Strong AI (Carnegie-Mellon approach)**:
   - This view emphasizes how intelligent behavior should be similar to how humans do things.
   - For example, a strong AI system that hears sounds would try to replicate the way human ears work, including all the parts like the eardrum and cochlea.
   - Strong AI supporters believe that if a computer can understand and replicate human processes, it could eventually have consciousness and intelligence, like in movies such as "I, Robot," "AI," and "Blade Runner."

Here's a table differentiating **Strong AI** and **Weak AI** with examples:

Aspect	Strong AI	Weak AI
Definition	Artificial intelligence that has consciousness, self-awareness, and the ability to think, learn, and perform tasks like a human.	AI designed to perform specific tasks without consciousness or general intelligence.
Capabilities	Generalized; capable of reasoning, solving unfamiliar problems, and understanding abstract concepts.	Specialized; performs predefined tasks efficiently but cannot generalize beyond them.
Learning	Learns and adapts like a human over time, even in unknown scenarios.	Learns within the constraints of its programming and training data.
Autonomy	Operates autonomously with decision-making abilities akin to human intelligence.	Dependent on programmed algorithms and specific instructions.
Examples	Hypothetical (e.g., a machine that understands and generates all forms of knowledge, like an AI equivalent of human intelligence).	Voice assistants (Siri, Alexa), spam filters, recommendation systems (Netflix, Amazon).
Applications	Not yet fully realized; potential applications include advanced robotics with human-like intelligence.	Widely used in automation, customer service chatbots, and data analysis.
Existence	Theoretical; no existing implementation.	Practical; extensively implemented in various industries.

Key Difference:

- **Strong AI** is a vision of achieving human-like cognition in machines.
- **Weak AI** is the practical implementation of AI that exists today for specific purposes.

1.3 Heuristics

Heuristics and algorithms are two different approaches used to solve problems.

- **Heuristic:** A heuristic is like a shortcut or a general rule that helps you make decisions or solve problems more easily. It doesn't guarantee that you'll always get the right answer, but it usually leads you in the right direction. For example, if you're looking for a restaurant, a heuristic might be to choose one that has a lot of people because that might mean it's good.
- **Algorithm:** An algorithm is a step-by-step procedure or a set of rules that you follow to get a specific result. It is very precise, and if you follow it correctly, you'll always get the same outcome. For example, sorting a list of numbers using a specific method (like bubblesort or quicksort) is done through an algorithm.

In simple, heuristics are helpful guidelines that can lead you to a solution but don’t ensure success, while algorithms provide a clear, reliable method to achieve a specific result.

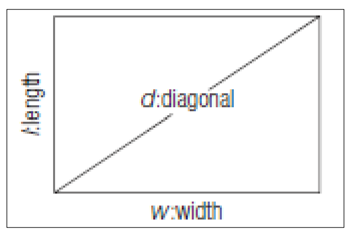
1.3.1 The Diagonal of a Rectangular Solid: Solving a Simpler, but Related Problem

The diagonal of a rectangular solid (a three-dimensional box) can be a challenging problem to solve. A helpful approach is to first look at a simpler problem that is related to it.

One of the ways to solve difficult problems is suggested by George Polya in his book *How to Solve It*. He advises that when you face a tough problem, try to tackle a simpler version of it first. This often helps you understand the bigger problem better.

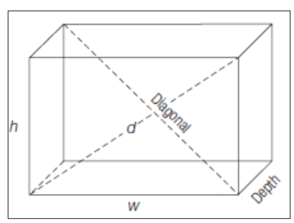
For example, to find the length of the diagonal of a rectangular solid, you can start by finding the diagonal of a rectangle (which is a flat, two-dimensional shape).

1. **Finding the diagonal of a rectangle**: To find the diagonal of a rectangle, you can use the Pythagorean theorem, which says that in a right triangle, the square of the length of the hypotenuse (the diagonal) is equal to the sum of the squares of the other two sides. If the rectangle has a width (w) and a length (l), the diagonal (d) can be found using this formula:



$$d = Sqrt\,\breve{(}h^2 + w^2).$$

2. **Extending to a rectangular solid**: Once you know how to find the diagonal of a rectangle, you can use a similar method to find the diagonal of a rectangular solid. The solid has three dimensions: width (w), length (l), and height (h). You can think of it as two rectangles stacked on top of each other. You can then apply the Pythagorean theorem again to find the diagonal of the solid.



By solving the simpler problem first, you gain insight that helps you with the original, more complex problem.

$$\text{Diagonal} = \text{Sqrt}\ (d^2 + \text{depth}^2) = Sqrt\ (h^2 + w^2 + \text{depth}^2)$$

1.3.2 **The Water Jug Problem: Working Backward**

The Water Jug Problem is about measuring a specific amount of water using two jugs of different sizes. For example, you want to measure exactly 12 quarts of water, and you have an 8-quart jug and an 18-quart jug.

Instead of trying random methods, you can use a strategy suggested by George Polya, which is to work backward from your goal. Here's how it works:

1. **Goal State:** You want to end up with 12 quarts of water in the 18-quart jug. You can get to this point by filling the 18-quart jug and having 2 quarts in the 8-quart jug.
2. **Working Backward:**
   - **Step (a):** You have 12 quarts in the 18-quart jug and 0 quarts in the 8-quart jug.
   - **Step (b):** To reach this state from the previous step, you need to pour 2 quarts from the 18-quart jug into the 8-quart jug.
   - **Step (c):** Before that, you had 2 quarts left in the 18-quart jug after filling and emptying the 8-quart jug twice.
   - **Step (d):** Initially, you filled the 18-quart jug from a tap or well.

By following these steps backward, you can see how to reach your goal:

1. Fill the 18-quart jug from the tap.
2. Pour from the 18-quart jug into the 8-quart jug until the 8-quart jug is full, leaving you with 2 quarts in the 18-quart jug.
3. Pour those 2 quarts into the 8-quart jug.
4. Fill the 18-quart jug again and pour from it into the 8-quart jug until the 8-quart jug is full again. This will leave you with exactly 12 quarts in the 18-quart jug.

This backward approach is useful for solving problems. In the early days of AI, researchers used a similar method in projects like the General Problem Solver (GPS). They studied how people solved problems and used those strategies to create problem-solving programs.

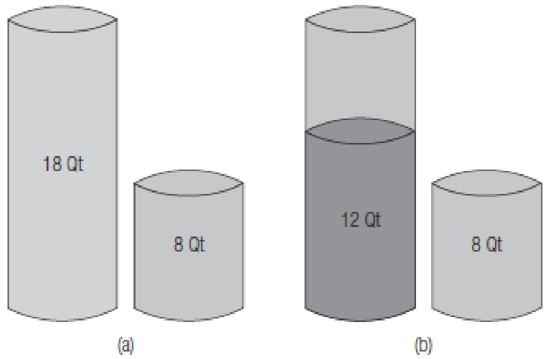


Figure 1.11
Water Jug Problem. (a) Initial state and (b) Final state.

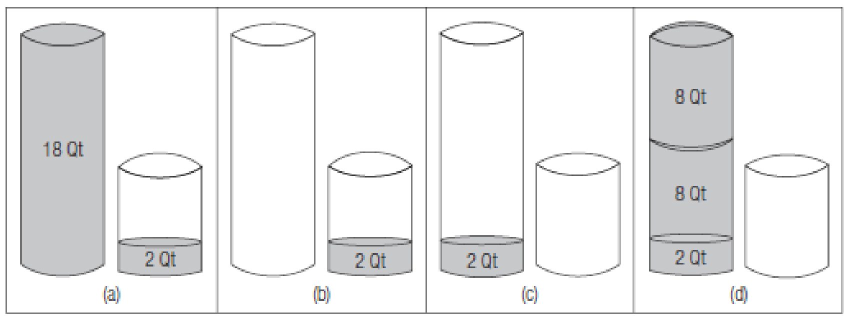


Fig: Starting with the goal state and working backward.

1.4 Key Characteristics of AI Problems

1. **Large in Size**: AI problems often involve massive amounts of data or scenarios.
2. **Complexity**: They are difficult to solve using simple algorithms.
3. **Human Expertise**: They require knowledge that humans have, such as rules or reasoning, making AI suitable for such tasks.

Examples of AI Applications:

- **Medical Diagnosis**: AI, especially expert systems, excels here because it deals with complex processes requiring knowledge from many experts. For instance, MYCIN is a system that uses over 400 rules to diagnose bacterial infections and provides probabilities for its suggestions.
- **Chess**: Chess programs use AI to make decisions in a game with billions of possible outcomes. While current chess programs can beat most humans, they don’t fully replicate human intelligence because they can't explain their moves like a true expert.

Examples Not Typically AI-Based:

- **Shopping at a Supermarket**: Scanning barcodes is straightforward and doesn’t need AI. However, an advanced system could suggest items to buy based on your habits, acting like a "smart advisor."
- **ATMs**: Traditional ATMs handle simple transactions. But if an ATM could analyze spending habits and provide financial advice, it would qualify as an AI system.

What Makes a Problem Suitable for AI?

- It involves complex decision-making.
- There is a large amount of specialized knowledge.
- The task is beyond what humans or simple programs can handle efficiently.

Why AI Matters:

AI systems can manage knowledge in ways humans cannot, like combining thousands of rules to make decisions or providing insights even experts may not foresee. For example, in chess or medical diagnosis, AI systems handle complexity and scale far beyond human limits.

1.5 Applications and Methods

To build an intelligent system, it must be able to understand and interact with the real world. This requires two key elements:

1. **A Formal Framework**: The system needs a structured way, like logic, to represent and understand the world around it.
2. **Handling Uncertainty**: The system must deal with situations where outcomes are not straightforward. For example, a fever could be caused by different factors like an infection or inflammation. The system uses rules and reasoning to figure out the most likely cause.

Key Areas of AI Research:

- **Expert and Automated Reasoning Systems**: These systems use knowledge and logical rules to solve problems, like diagnosing illnesses or identifying car accidents.
- **Game Playing**: AI systems for games like chess rely on search algorithms to predict the outcomes of moves and make strategic decisions.
- **Learning**: Just like humans need practice to improve, intelligent systems need to learn. Researchers have used models based on animal brains (neural networks) and human evolution (evolutionary computation) to teach AI systems how to learn and adapt.

Creating Complex AI:

- Some researchers believe intelligent systems can "grow" from simple starting rules, much like patterns in **Cellular Automata** (CA). These are systems where simple rules can lead to very complex behavior, offering hope for creating human-level AI someday.

AI Application Areas:

1. **Search Algorithms and Puzzles**: Solving problems by searching for the best solution.
2. **Games**: Playing strategy games like chess using advanced decision-making.
3. **Reasoning Systems**: Drawing logical conclusions from knowledge.
4. **Expert Systems**: Specialized systems that mimic human experts in fields like medicine.
5. **Cellular Automata**: Exploring how simple rules lead to complex patterns.
6. **Neural and Evolutionary Computation**: Mimicking how animals and humans learn and evolve.
7. **Knowledge Representation**: Structuring knowledge so AI can use it effectively.
8. **Reasoning with Uncertainty**: Making decisions even when information is incomplete.

Note

- A **heuristic algorithm** is a problem-solving approach designed to produce a solution that is good enough for solving a particular problem, especially when finding an exact or optimal solution is impractical due to complexity or time constraints.
- **State Space Graph (Universe of Discourse)**

  A state space graph **is a representation of all possible states of a system or problem and the transitions (or actions) between those states. It is often used in** artificial intelligence**,** graph theory**, and** problem-solving **domains to model and explore potential solutions.**

- **Search space** refers to the complete set of all possible solutions or configurations that can be explored to solve a given problem. In computational or optimization contexts, it represents the "universe" in which an algorithm searches for the optimal or desired solution.
- A **space tree** is a hierarchical representation of the search space, structured in the form of a tree where:
  - **Root Node**: Represents the initial state.
  - **Branches**: Represent possible actions or decisions.
  - **Leaf Nodes**: Represent terminal states, which may or may not be solutions.

1.5.1 Search Algorithms and Puzzles:

The 15-puzzle, 8-puzzle, and 3-puzzle are simple puzzles that help explain how search algorithms and problem-solving work. Here's how these puzzles and related ideas can be understood:

1. **What is the 15-puzzle?**
   - The 15-puzzle has 15 numbered tiles (1 to 15) placed in a 4x4 grid, leaving one square blank. You slide tiles into the blank space to rearrange them.
   - Smaller versions of this puzzle, like the 8-puzzle (3x3 grid) and 3-puzzle (2x2 grid), are easier to handle.
2. **How does it work?**
   - The blank square can move in one of four directions: up (↑), down (↓), left (←), or right (→), depending on the surrounding tiles.
   - The goal is to start with a random arrangement (start state) and rearrange the tiles to match a specific target (goal state), often with tiles in numeric order.

Figure 1.13
Setting up a 15-puzzle.

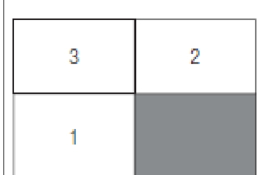


Figure 1.14
Using a 3-puzzle.

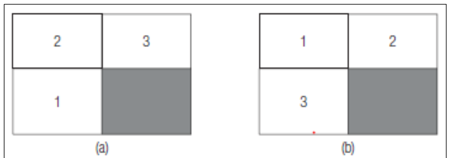


Figure 1.15
States for solving the 3-puzzle. (a) Initial state and (b) Goal state.

   - For example, in the 3-puzzle, only two directions are possible at any time because of the smaller grid.
3. **State-Space Graph and Space Tree:**
   - The **state-space graph** represents all possible arrangements of the puzzle tiles (nodes) and the legal moves (connections).
   - A smaller version of this, the **space tree**, starts from the initial arrangement and branches out to possible moves until reaching the goal.
4. **Blind Search Methods:**
   - **Blind search** explores the puzzle without knowing which path leads to the solution. Common methods are:
     - **Depth-First Search (DFS):** Goes as deep as possible down one path before backtracking.
     - **Breadth-First Search (BFS):** Explores all possible moves at one level before moving to the next.
5. **The Challenge of Combinatorial Explosion:**
   - For large puzzles like the 15-puzzle, the number of possible states grows extremely fast (over 16 trillion for the 15-puzzle). This makes solving them impractical using blind search methods alone.
6. **Heuristics and Intelligent Search:**
   - To overcome the challenge, **heuristics** are used. These are strategies or rules of thumb to estimate how close a state is to the goal.
   - **Forward-looking algorithms** like hill climbing, beam search, and best-first search focus on moving toward the goal based on these estimates.
   - **Backward-looking algorithms** like branch-and-bound or the A* algorithm work by estimating both the distance traveled and the remaining distance to the goal.

1.5.2 Two Player games:

Two-player games like **Nim**, **tic-tac-toe**, and **chess** are different from puzzles because you're not just trying to reach a goal; you also have to stop your opponent from reaching theirs. These games are called **adversarial games**, where both players compete against each other. This type of game has been important in Artificial Intelligence (AI) research for many years because it involves strategy, decision-making, and predicting your opponent's moves.

For example:

- **Nim:** Players take turns removing stones from piles. The player forced to take the last stone loses the game.
- **Tic-tac-toe and chess:** You must think ahead to block your opponent while working toward your own victory.

A more complex example is the **Iterated Prisoner's Dilemma**:

- Two criminals are caught and put in separate cells.
- They are told they'll get a lighter punishment if they betray (tell on) the other person.
- If one betrays and the other stays loyal, the loyal one gets a longer sentence.
- If both betray, they both get punished.
- If both stay loyal, they get less punishment.
- The best choice depends on whether they trust each other and what their future intentions are (like committing more crimes together or not).

1.5.3 Automated Reasoning

An **automated reasoning system** is software that uses given facts to figure out new facts. This process is called **deduction**. Let's look at an example puzzle:

- There are two people, Michael and Louis.
- They each have one job: a **post office clerk** or a **French professor**.
- Michael speaks only English, and Louis has a PhD in French.
- Who does which job?

To solve this puzzle with a reasoning system, you first need to represent the information in a way the system understands, using statements called **clauses**. For example:

- "Michael or Louis works as a clerk" can be written as:
  `Works_As(Clerk, Michael) | Works_As(Clerk, Louis)`
  (Here, `|` means "or.")

However, just writing the problem in this format isn't enough for the system to solve it. The system also needs **common sense** or **world knowledge**. For example:

- Common sense tells us that a French professor must know French.
- From the given information, Michael only knows English, so he cannot be the French professor.

Using this reasoning, we conclude:

- Michael is the post office clerk.
- Louis is the French professor.

For small problems like this, humans can solve them easily. But for complex problems, an automated reasoning system can be very helpful.

1.5.4 Production Rules and Expert Systems

Production rules are a way to represent knowledge in Artificial Intelligence (AI). Each rule has the format:

- **IF (Condition), THEN Action**
  Example:
  - IF you have a headache, THEN take two aspirins.
- **IF (Condition), THEN Fact**
  Example:
  - IF A > B and B > C, THEN A > C.

These rules are commonly used in **expert systems**, which are AI programs that have specialized knowledge about a specific area. For example, in car troubleshooting:

1. IF the car won't start, THEN check the headlights.
2. IF the headlights work, THEN check the gas gauge.
3. IF the gas tank is empty, THEN add gasoline.
4. IF the headlights don't work, THEN check the battery.

With enough production rules, even someone without much technical knowledge can use an expert system to solve problems, like diagnosing and fixing a car issue.

History:
Expert systems started in the 1970s (examples: MYCIN, DENDRAL, PROSPECTOR) and became very popular by the late 1980s.

1.5.5 Cellular Automata (CA)

A **Cellular Automaton (CA)** is like a grid made up of cells, which can exist in one of a few possible states. For example, a cell could be:

- **State 1:** Alive (shaded/black)
- **State 0:** Dead (unshaded/white)

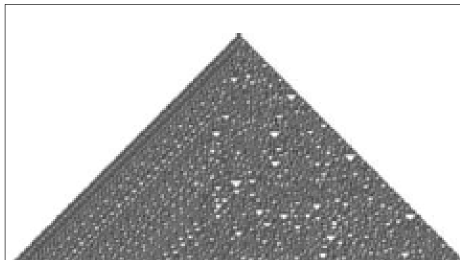


Figure 1.17
Part of a one-dimensional CA.

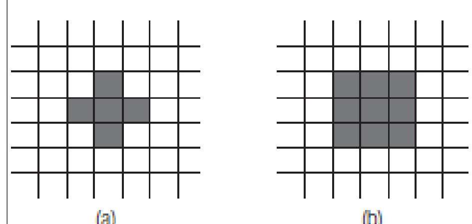


Figure 1.18
Neighborhoods in a two-dimensional rectangular cellular automaton. (a) The Von Neumann neighborhood and (b) the Moore neighborhood.

Key Properties:

1. **Shape of the Grid (Physical Topology):**
   - The grid can have different shapes, like rectangular or hexagonal.
2. **Update Rule:**
   - Each cell's future state is determined by its current state and the states of the nearby cells (its **neighborhood**).
   - These updates happen at fixed time intervals, and all cells are updated at the same time.

How it Works:

- Each cell has a neighborhood of nearby cells, which typically includes the 8 surrounding cells (top, bottom, left, right, and diagonals).
- A simple set of rules decides whether a cell stays alive, dies, or becomes alive in the next step.

Example:

In a rectangular grid, if a cell is surrounded by too many alive cells, it might "die" in the next update due to overcrowding. Similarly, if it’s surrounded by just the right number of alive cells, it might stay alive or become alive.

Why It's Interesting:

Even though the rules are simple, **Cellular Automata** can create very complex and fascinating patterns, like those seen in **Game of Life**, a famous example of CA.

1.5.6 Neural Computation

To create **Artificial Intelligence (AI)**, researchers often take inspiration from the human brain, which is the best example of intelligence. One approach is using **Neural Networks**, which try to mimic how the brain’s nervous system works in parallel and distributed ways.

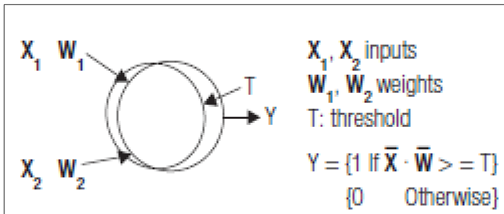


Figure 1.19
Threshold Logic Unit.

The Basics of Neural Networks:

1. **Artificial Neuron (Threshold Logic Unit - TLU):**
   - An artificial neuron takes inputs (e.g., **X1** and **X2**) and multiplies them by weights (e.g., **W1** and **W2**).
   - The neuron adds up these weighted inputs and checks if the total reaches a certain threshold.
   - If the total is greater than or equal to the threshold, the output is **1** (ON); otherwise, it’s **0** (OFF).
2. **Example – AND Function:**
   - A TLU can perform a Boolean **AND** operation.
   - It only outputs **1** when both inputs are **1**. Otherwise, the output is **0**.
3. **Learning Weights (Perceptron Learning Rule):**
   - Neural networks learn by adjusting the weights.
   - If the neuron gives the wrong output, the system changes the weights slightly.
   - This process repeats until the outputs are correct for all inputs.
4. **Discriminant (Decision Boundary):**
   - The TLU separates inputs into two groups: those that produce **0** and those that produce **1**.
   - This separation is represented by a straight line, called a **discriminant**, in a graph.

Larger Neural Networks:

- Simple neurons are combined to form larger neural networks with hundreds or thousands of such units.
- These networks can do complex tasks, like:
  - Recognizing handwriting.
  - Predicting stock prices based on past data.

Inspiration from the Brain:

Just like the brain has billions of connected neurons, neural networks in AI are made of many interconnected artificial neurons working together to solve problems.

1.5.7 Genetic Algorithms

AI researchers use different ideas to solve problems, and one approach is inspired by **Darwin's theory of evolution**. This approach mimics how living things adapt to their environments over time. Instead of taking millions of years, this process is simulated much faster on a computer.

What is Evolution in AI?

1. **Natural Selection in Nature:**
   In the real world, plants and animals evolve and adapt using processes like:
   - **Natural selection**: The strongest survive.
   - **Reproduction**: Passing on traits to offspring.
   - **Mutation**: Small random changes in genes.
   - **Recombination**: Mixing genes from parents.
2. **Evolution in AI:**
   AI uses similar processes to improve problem-solving. This approach is called **Evolutionary Computation**.

What are Genetic Algorithms (GA)?

- Genetic Algorithms are a method in AI inspired by evolution. They work like this:
  1. **Problem as a String**:
     The problem is represented as a series of binary numbers (0s and 1s).
     For example, in a puzzle, moving up could be **00**, moving down **01**, right **10**, and left **11**.
  2. **Random Start**:
     The algorithm starts with a random group of solutions (strings of 0s and 1s).
  3. **Fitness Function**:
     - Each solution is checked to see how good it is using a **fitness function**.
     - Solutions that are closer to the goal (like solving a puzzle) get higher scores.
  4. **Genetic Operators**:
     - **Selection**: Keep the best solutions.

    - **Reproduction**: Combine good solutions to create new ones.
    - **Mutation**: Make small random changes to improve solutions.

Example: Solving a Puzzle

- Imagine a puzzle where you move a blank space.
- Each move is encoded as a binary string:
  - **00**: Move up
  - **01**: Move down
  - **10**: Move right
  - **11**: Move left
- Two random solutions could be:
  - **11100010**: A sequence of moves.
  - **00110110**: Another sequence of moves.
- The **fitness function** checks which solution gets closer to solving the puzzle.
  - If **00110110** brings the puzzle closer to the solution than **11100010**, it gets a higher fitness score.

Why Use Genetic Algorithms?

Genetic Algorithms are powerful because they can adapt and improve solutions over time. They are helpful for solving complex problems where traditional methods might struggle.

1.5.8 Knowledge Representation

When working with Artificial Intelligence (AI), one of the first challenges is figuring out how to represent knowledge so that AI systems can use it to solve problems. The way this knowledge is stored and displayed is critical because it can make solving problems easier or harder.

Key Points:

1. **Importance of Representation**:
   To act intelligently, AI systems need to store and process knowledge. How this knowledge is represented affects how quickly and accurately the system can solve problems.
   - **Good Representation**: Makes solving problems faster and easier to understand.
   - **Bad Representation**: Can slow down problem-solving and make solutions confusing.
2. **Example: Missionaries and Cannibals Problem**:
   - The task is to move three missionaries and three cannibals across a river without breaking rules (e.g., cannibals can't outnumber missionaries).
   - Different ways to represent the problem:
     - **Text-Based**: Write the state as "W:3M3CB" (all on the west bank) and the goal as "E:3M3CB" (all on the east bank).
     - **Visual-Based**: Use stick figures or pictures to show the boat and people.
   - A good representation helps you see the solution path more clearly.

3. **Logic-Based Representations**:
    - Early AI programs, like **Logic Theorist** and **GPS**, used logical rules to solve problems.
    - Example: Winograd's **Blocks World** (1972) involved a robot arm arranging blocks, combining problem-solving, language understanding, and visual analysis.
4. **Production Rules and Expert Systems**:
    - AI systems often use "if-then" rules to represent knowledge clearly and solve problems efficiently.
    - Thousands of expert systems use this approach.
5. **Graphical Representations**:
    - Graphical methods like **state-space representations** (all possible states of a system) help visualize and solve problems.
    - **Semantic Networks**: Use diagrams to represent relationships between concepts. These are like the early versions of object-oriented programming.
6. **Frames and Scripts**:
    - **Frames**: Organize knowledge hierarchically, like a table with categories (e.g., a university has buildings, faculty, and students). Frames also allow inheritance (e.g., a "dog" inherits traits from "animal").
    - **Scripts**: Extensions of frames that show sequences of events, like the steps involved in eating at a restaurant.
7. **Minsky's Society of Mind**:
    - Marvin Minsky suggested that intelligent behavior comes from combining many small, simple agents. These agents work together to form a "society" that appears intelligent.

Why Representation Matters:

The right representation helps AI systems solve problems faster, understand language better, and simulate intelligent behavior. Different methods (logic, graphs, frames, etc.) are chosen based on what fits the problem best.

1.5.9 Uncertainty Reasoning

Traditional mathematics deals with clear-cut answers: something is either true or false. For example, a set is either a part of another set, or it is not. But in real life—and in Artificial Intelligence (AI)—things are not always this certain. There's often some *uncertainty* or a "gray area" where something is partly true or partly false.

Key Points:

1. **Life is Uncertain**:
    - Example: You might catch a cold from someone coughing on the bus—or you might not. There's no definite yes or no.

2. **Fuzzy Sets**:
    - In real life, some things don't fit neatly into one category or another.
    - Example: People satisfied with their jobs:
        - Some people love their jobs but feel underpaid.
        - Satisfaction varies for different reasons and degrees.
    - Fuzzy sets describe this uncertainty. A person might belong to the "satisfied" set *to some degree*.
        - **1.0** = Fully satisfied with their job.
        - **0.0** = Not satisfied at all.
        - Anywhere in between (e.g., 0.5) = Partly satisfied.
3. **Applications of Fuzzy Logic**:
   Fuzzy logic works well in situations where things aren't black-and-white. Examples:
    - **Cameras**: Adjust shutter speed based on sunlight—not exact but within a range.
    - **Washing Machines**: Choose wash cycles based on how dirty the clothes are.
    - **Thermostats**: Keep room temperatures comfortable within a range, rather than an exact value.
    - **Cars**: Adjust brake pressure based on road and weather conditions.

Why It Matters:

Fuzzy logic is useful in AI because it handles uncertainty and approximate reasoning—similar to how humans make decisions in real life. It helps machines work more flexibly and adaptively in unpredictable situations.

1.6 Early History of AI

The idea of building intelligent machines has fascinated humans for centuries. The journey to develop Artificial Intelligence (AI) involves contributions from philosophers, mathematicians, and logicians throughout history. Here's a breakdown:

1. **Ancient Efforts**:
   - Ancient Egyptians tried to imitate intelligence with hidden priests inside statues giving advice. These were clever tricks, not real intelligence.
2. **Aristotle's Contributions (350 BC)**:
   - Aristotle, an ancient Greek philosopher, laid the foundation for scientific thinking and logic.
   - He focused on the ability to reason, which he believed set humans apart from other living beings.
   - His idea of separating "form" (the method) from "matter" (the actual content) inspired *data abstraction* in computer science—how methods are separated from the objects they manipulate.
3. **George Boole (1800s)**:
   - Developed Boolean algebra, a system to express logical relationships. This became fundamental in computer logic and AI.
4. **Raymond Llull (1200s)**:
   - A Spanish scholar who tried to create a logical system to prove Christian beliefs.
   - His ideas used diagrams and simple devices to represent logical reasoning.
   - His work influenced later thinkers.
5. **Leibniz (1600s)**:
   - Expanded Llull's ideas, proposing a "universal algebra" or "logical calculus" to solve any logical argument.
   - He believed reasoning could be reduced to combining and substituting symbols, similar to modern algorithms.
6. **Kurt Gödel (1931)**:
   - Proved that no mathematical system could be entirely complete. There will always be statements that can't be proven true or false within the system, showing the limits of logic.
7. **René Descartes (1600s)**:
   - Famous for the statement, "I think, therefore I am."
   - He explored the relationship between the mind and physical reality, which influenced ideas about intelligence and reasoning.
8. **Modern Perspective on Mind and Body**:
   - Today, many believe that the mind and body are not entirely separate but interconnected, influencing how we approach AI.

Why It Matters:

The history of AI shows how human reasoning, logic, and philosophy have shaped our understanding of intelligence. Each thinker contributed ideas—like reasoning, abstraction, and the limits of logic—that now form the foundation of modern AI.

The journey to creating intelligent machines is marked by many pioneers and inventions. Here's an easy-to-follow summary:

1. **Stanhope Demonstrator (1775)**:
   - Charles Stanhope built the **first logic machine**, which used colored glass slides to test simple logical arguments.
   - It was basic but paved the way for machines to perform reasoning.
2. **Charles Babbage (1800s)**:
   - Known as the "father of computers," Babbage designed the **Difference Engine**, a machine to perform calculations.
   - He later designed the **Analytical Engine**, which could theoretically play games and reason like humans, but it was never built.
   - **Lady Ada Lovelace**, his collaborator, is considered the **first programmer** because she wrote instructions for this machine.
3. **George Boole (1800s)**:
   - Created **Boolean algebra**, a logical system that became the foundation for computer science.
   - His work helps computers process logic using 1s and 0s.
4. **Claude Shannon (1900s)**:
   - Called the "father of information science," Shannon applied Boolean logic to circuits.
   - His work laid the foundation for modern computers and explored how machines could learn and play games like chess.
5. **Early Game-Playing Machines**:
   - **Nimotron (1938)**: A machine that could play the game of Nim by using an algorithm to make the best moves.
   - **The Turk (1790)**: A fake "thinking" chess machine. It had a hidden human player inside, tricking people into believing it was intelligent.
6. **Torres y Quevedo**:
   - A Spanish inventor who created a rule-based system to play the chess endgame (King and Rook vs. King). This was an early example of **expert systems**.
7. **Konrad Zuse (1940s)**:
   - Built the **first digital computer** powered by electricity, called the **Z3**, which could perform complex calculations.
   - His work linked engineering with Boolean logic, essential for AI and computers.

Key Takeaways: These milestones show how human curiosity and logic led to the intelligent machines we use today.

- **Logic and Reasoning**: Early machines aimed to mimic human reasoning through rules and logic.
- **Game-Playing Machines**: These paved the way for understanding how machines could "think" strategically.
- **Pioneers Like Babbage and Zuse**: They laid the groundwork for modern computers and AI.
- **Boolean Logic**: Became the language of computers, enabling them to process decisions.

1.7 Recent History of AI to the Present

Since World War II, with the invention of computers, computer science and programming have greatly improved by taking on the challenge of teaching computers to play and excel at difficult board games. Games like chess, checkers, Go, and Othello have especially benefited from the use of AI techniques, helping computers learn strategies and make better decisions.

1.7.1 Games

The development of artificial intelligence (AI) has been heavily influenced by games, which have served as a testing ground for creating intelligent machines.

- In 1959, **Arthur Samuel** developed a program to play checkers. The program used a set of 50 rules (heuristics) and improved itself by playing against its own versions, adopting strategies from the winning version. Although it played well, it never fully mastered the game.
- Chess has also been a focus of AI development because it is widely seen as a game that requires intelligence. The first serious chess program was created in 1959 by **Newell, Simon, and Shaw**, based on a framework called the Shannon-Turing Paradigm. Over the years:
  - In the 1970s, chess programs reached an *Expert level* (top 1% of tournament players).
  - By 1983, **Ken Thompson's Belle** became the first program to achieve the *Master level*.
  - Programs like **Hitech** and **Deep Thought** continued improving, with Deep Thought regularly defeating Grandmasters.
  - IBM's **Deep Blue**, a successor to Deep Thought, famously defeated World Chess Champion **Garry Kasparov** in 1997, a milestone in AI history. Since then, top chess programs have played at a level nearly indistinguishable from the best human players.
- In 1989, **Jonathan Schaeffer** started working on checkers with his program **Chinook**. Chinook competed against **Marion Tinsley**, the longtime Checkers World Champion, in the 1990s. While Tinsley won some matches, Chinook showed great progress and led efforts to "solve" the game of checkers by analyzing every possible position.
- Other games like backgammon, poker, bridge, Othello, and Go have also contributed to advancements in AI. Each game provides unique challenges that push the boundaries of AI techniques.

1.7.2 Expert Systems

Expert systems are a key success story in artificial intelligence. They are computer programs designed to solve specific problems by mimicking the decision-making abilities of human experts. Here's an explanation of their history and significance:

- **What makes expert systems special?**
  Expert systems separate the knowledge base (the information they use) from the

inference engine (the logic they use to make decisions). They can combine knowledge from multiple experts and handle reasoning and uncertainty effectively.

- **Early examples:**
  - **DENDRAL** (1960s, Stanford University): It analyzed chemical compounds using mass spectrographs and was designed for Martian soil studies. It showed how expert knowledge could be encoded into a computer.
  - **MYCIN** (1970s, Stanford University): It helped diagnose blood infections and trained medical residents with over 400 rules. It became a model for building future knowledge-based systems.
  - **PROSPECTOR** (1970s): It was used for mineral exploration and introduced inference networks.
- **Other important systems:**
  - **XCON**: Used 10,000 rules to configure electrical circuits for VAX computers.
  - **GUIDON**: A tutoring system based on MYCIN.
  - **HEARSAY I and II**: Early systems for speech understanding.
  - **AM (Artificial Mathematician)**: Worked on mathematical problem-solving.
  - The **Dempster-Shafer Theory** and **fuzzy logic** introduced new ways to handle uncertainty.
- **Modern impact (since the 1980s):**
  Thousands of expert systems have been developed for tasks like diagnosis, planning, monitoring, and control. They are now often embedded in other software, such as medical devices and cars (e.g., traction control systems).
- **Tools for building expert systems:**
  Tools like Emycin, OPS, EXSYS, and CLIPS make it easier to create expert systems. These systems also use specialized languages to represent knowledge.
- **Everyday applications:**
  Today, expert systems work behind the scenes to improve daily life, such as enhancing online shopping experiences (e.g., managing shopping carts).

1.7.3 Neural Computing

The early development of artificial neural networks (ANNs) and their applications can be explained simply:

- **Early Research:**
  In the 1940s, **McCulloch and Pitts** studied how animal nervous systems work. They created one of the first models of artificial neural networks. However, their model had a major limitation—it couldn't learn.
- **Perceptron and Learning:**
  In the 1950s, **Frank Rosenblatt** developed the *Perceptron Learning Rule*, a method to help simple neural networks adjust their connections (called "weights") to solve problems. These networks, called *single-layer networks*, connect all neurons directly to inputs.
  However, in the 1960s, **Minsky and Papert** showed that single-layer networks couldn't solve certain problems, like the XOR function. Their criticism caused funding for neural network research to drop significantly.

- **Revival in the 1980s:**
  Research picked up again in the 1980s:
    - **Hopfield networks:** These networks used a mathematical "energy function" to find solutions to complex problems.
    - **Backpropagation (mid-1980s):** A breakthrough learning method for *multilayered networks* (networks with multiple layers of neurons). Backpropagation allowed these networks to learn more complex tasks.
- **Applications of Neural Networks:**
    - Predicting stock market trends, like the Dow Jones average.
    - Optical character recognition (OCR), such as reading printed text.
    - **ALVINN project:** At Carnegie Mellon University, a backpropagation network was used to help steer a self-driving vehicle, *Navlab*. This system could warn drowsy or impaired drivers when their car left the lane.
- **Future Vision:**
  Researchers hope that neural networks will one day allow fully autonomous vehicles, enabling us to focus on other tasks like reading or talking while the car drives itself.

1.7.4 Evolutionary Computation

- **Genetic Algorithms (GAs):**
  GAs are a type of **evolutionary computation**, inspired by natural selection and genetics. They solve complex problems, often called optimization problems, by using **probability** (randomness) and **parallelism** (testing many solutions at once). This method, developed by **John Holland**, is good at finding the best solutions in situations with many possibilities.
- **What is Evolutionary Computation?**
  While GAs focus on solving optimization problems, **evolutionary computation** is broader. It involves using ideas from evolution (like mutation and selection) to create intelligent systems.
- **Rodney Brooks' Approach to AI:**
    - Brooks, a former MIT AI Lab director, had a different view on how to achieve human-level intelligence, which is often called the "holy grail" of AI.
    - He **rejected traditional AI methods** that rely on symbols and rules to represent knowledge. Instead, he believed that intelligence comes from how an agent (like a robot) interacts with its environment.
    - **Subsumption Architecture:**
        - Brooks proposed building intelligent systems in **layers**, where higher layers rely on the basic ones. For example, in a robot that avoids obstacles:
            - A basic layer handles movement.
            - A higher layer uses this movement to avoid obstacles.
    - His philosophy was to let the **world itself serve as a guide**, rather than using complex internal representations.
- **Famous Work:**
  Brooks is best known for creating insect-like robots. These robots act autonomously,

interact with their environment, and work together as a group, showcasing how intelligence can emerge from simple behaviors and interactions.

1.7.5 Natural Language Processing

To create intelligent systems, it makes sense for them to understand human language. This idea was important to early researchers, and two famous programs, **Eliza** and **SHRDLU**, demonstrated different approaches to this goal.

Eliza

- Developed by **Joseph Weizenbaum** at MIT, with help from psychiatrist **Kenneth Colby**.
- Eliza acted like a therapist, using simple tricks to mimic a conversation.
    - Example: If you said, *"I feel tired,"* Eliza might reply, *"You say you feel tired. Tell me more."*
    - It didn't actually understand what you said—it just matched patterns in your words.
- **Public Reaction:**
    - People enjoyed talking to Eliza, even though they knew it was just a program. This surprised Weizenbaum, who found it unsettling.
    - Colby later created a similar program called **DOCTOR**.
- **Significance:**
    - Eliza didn't contribute much to natural language processing (NLP), but it showed how people can form emotional connections with machines pretending to understand them.

SHRDLU

- Created by **Terry Winograd** at MIT, SHRDLU was a more advanced language program.
- It could understand and respond to English commands by reasoning about a small, controlled world: a tabletop with blocks of different shapes and colors.
    - For example, if you told it to move a red block under a green one, SHRDLU would know it had to clear the green block first.
    - It didn't just respond randomly—it understood the tasks and carried them out logically.

HEARSAY and HWIM

- **HEARSAY:** Focused on speech recognition and used a "blackboard architecture." This allowed different parts of the system (like phonetics and grammar) to work together and rule out unlikely combinations of words.
- **HWIM (Hear What I Mean):** Tried to understand spoken language about travel budgets, using a vocabulary of 1,000 words. However, it was too ambitious and didn't perform as well as HEARSAY.

The early language programs mentioned earlier, like Eliza and SHRDLU, had some understanding of specific, limited worlds. However, they lacked **common sense knowledge**—the basic understanding of how the real world works.

The Problem with Microworlds

- These programs could only handle specific scenarios, known as **microworlds**.
  - Example: A program might know how to handle a conversation about ordering food at a restaurant.
  - But it wouldn't know basic things like whether the waiter is alive or whether they usually wear clothes.
- This lack of general knowledge was a big challenge for natural language processing (NLP) and artificial intelligence (AI).

The Solution: Common Sense Knowledge

- **Douglas Lenat** started building a massive database of common sense knowledge to help solve this problem.
- This project has been ongoing for over 25 years and aims to give machines a more general understanding of the world.

A New Approach: Statistical NLP

- In recent years, NLP shifted from relying on "world knowledge" to using **statistics** to analyze language.
- This approach uses large datasets, like the **Penn Treebank**, which has over a million words of English text. These texts were manually analyzed to show how sentences are structured.
- **Statistical Parsing:**
  - Rules for sentence structures (like grammar) are given probabilities based on how often they occur in real-life text.
  - For example, a rule might say that a certain word order is 80% likely in English.
  - This method was shown to work well, even for complex sentences like those in **The New York Times**, which are hard for many humans to understand.

This shift to using statistics and probabilities has made NLP systems more powerful and better at handling real-world language. It's no longer just about specific tasks—it's about understanding language in a broader, more flexible way.

1.7.6 Bioinformatics

Bioinformatics is a new field where computer science techniques, like algorithms and data analysis, are used to study biology, especially at the molecular level. It focuses on managing and analyzing large amounts of biological data.

Key Areas in Bioinformatics

1. **Structural Genomics**
   - The goal is to figure out the structure of every protein we observe.
   - Techniques like **automated discovery** and **data mining** help researchers identify these protein structures.
2. **Case-Based Reasoning**
   - This approach uses examples of known protein structures to predict the structure of new proteins.
   - Researchers Jurisica and Glasgow explored this in their studies.
3. **Microarray Data Analysis**
   - One of the fastest-growing areas in bioinformatics.
   - Microarray data involves studying how genes are expressed in different conditions, helping scientists understand diseases or cell behavior.

The Challenges

- Microbiologists are overwhelmed by the **huge variety and quantity of data** available in molecular biology.
- They have to understand molecular sequences and structures from massive databases, which is very complex.

How AI Helps

- **Knowledge Representation:** AI helps organize biological knowledge in a way that computers can understand.
- **Machine Learning:** AI can find patterns in the data, making it easier for scientists to make discoveries.

In simple words, bioinformatics uses computer science and AI to help biologists handle and make sense of vast amounts of biological data.

1.8 AI in the new Millennium

Why AI is Special in Academia

- **Exploring the Future:** AI is unique because it helps us imagine and shape what life might be like in the future.
- **Quick Adoption:** Even though AI is relatively new, its methods have quickly become part of regular computer technology.

Examples of AI in Everyday Technology

- **ALVINN:** This was an early AI system that used a type of brain-like network (neural network) to drive a car around a university campus.
- **Financial Systems:** Many AI tools now help make decisions about buying and selling stocks using different AI methods like neural networks and expert systems.
- **Web Agents:** These AI programs search the internet to find news articles that users might find interesting.

Future Impact of AI and Technology

- **Longer Lives:** People are living longer, and this trend will continue thanks to improvements in medicine, drugs, nutrition, and health knowledge.
- **Advanced Prosthetics:** New devices will help people with disabilities live more freely.
- **Smart Systems:** Tiny, hidden AI systems might enhance our brain functions, making it hard to tell where a person ends and the machine begins.

Challenges Ahead

- **Cost and Access:** Initially, these advanced technologies will be expensive and not available to everyone.
- **Political and Social Issues:** There will be important decisions about who gets to use these technologies and how society adapts to longer lifespans.
- **Ethical Questions:** Questions like “What makes us human?” and “How will society handle people living over 100 years?” will arise.

Different AI Approaches and Their Future Roles

- **Various Methods:** AI uses many techniques like logic, searching, organizing knowledge, genetic algorithms, and neural networks.
- **Expert Systems and More:** These systems could help us in many ways, from understanding language and seeing the world like humans do, to building smart robots.
- **Uncertain Leaders:** It’s hard to predict which AI method will become the most important, but many will work together to improve our lives.

Benefits and Dangers of AI

- **Amazing Possibilities:** AI can bring great benefits, making life easier and more efficient.
- **Risks:** There are dangers, such as accidents from unexpected interactions between technology and the environment or AI being used by bad people (e.g., terrorists using armed robots).
- **Acceptance of Risks:** People are likely to accept these risks because of the huge benefits AI offers, whether they realize it or not.

The Future of Robots

- **Current Roles:** Robots already help build machines and do simple tasks like vacuuming or shopping.
- **Advanced Help:** In the future, robots could assist in more complex areas like search and rescue missions or providing remote medical care.
- **Emotional Robots:** Robots might even show emotions and form bonds with humans, making them seem more human-like.
- **Blurring Lines:** As robots become more advanced, it might become hard to distinguish between our online (digital) lives and our real-world interactions.
- **Human vs. Machine Intelligence:** There's a possibility that robots could become smarter than humans, raising questions about what it means to be human.

Preparing for the Future

- **Thinking Ahead:** By considering these questions now, we can better prepare for the changes and challenges that AI and advanced technologies will bring.

In simple words, AI is rapidly becoming a fundamental part of our technology and daily lives. While it offers incredible benefits and possibilities for the future, it also brings significant ethical, social, and safety challenges that we need to address as the technology continues to advance.

Short Answer Questions

1. **Introduction**
    - Define artificial intelligence (AI).
    - Mention two primary goals of AI.
    - List two characteristics of intelligent systems.
2. **The Turing Test**
    - What is the Turing Test?
    - Who proposed the Turing Test and in which year?
    - What is the primary goal of the Turing Test?
    - What are intelligent systems?
3. **Strong AI versus Weak AI**
    - Define Strong AI and Weak AI.
    - Give one example of a Strong AI application and one of a Weak AI application.
    - What is the key difference between Strong AI and Weak AI?
4. **Heuristics**
    - What is a heuristic?
    - Why are heuristics important in AI problem-solving?
    - Provide one example of a heuristic used in real-world applications.
5. **Identifying Problems Suitable for AI**
    - List two criteria for problems that are suitable for AI.
    - Name one real-world problem where AI has shown significant success.
6. **Applications and Methods**
    - Mention two domains where AI is widely applied.
    - What are the two primary methods used in AI?
7. **Early History of AI**
    - Name one significant milestone in the early history of AI.
    - What was the role of the Dartmouth Conference in AI history?
8. **Recent History of AI to the Present**
    - Name one significant development in AI from the 21st century.
    - What is deep learning, and why is it significant in recent AI advancements?
9. **AI in the New Millennium**
    - What is the role of AI in the Fourth Industrial Revolution?
    - List one major breakthrough in AI after the year 2000.

Long Answer Questions

1. **Introduction**
    - Explain the scope and importance of AI in modern technology.
    - Discuss the key components of an intelligent system.
2. **The Turing Test**
    - Describe the Turing Test in detail. How does it evaluate a machine's intelligence?
    - Critically analyze the limitations of the Turing Test.
    - What is the criteria for Turing test? How the Turing test works?
3. **Strong AI versus Weak AI**
    - Compare and contrast Strong AI and Weak AI with suitable examples.
    - Discuss the ethical implications of achieving Strong AI.
4. **Heuristics**
    - Explain the concept of heuristics and their role in AI decision-making processes.
    - Discuss the strengths and weaknesses of heuristic-based methods in AI.
5. **Identifying Problems Suitable for AI**
    - Elaborate on the characteristics of problems that are suitable for AI solutions.
    - Provide examples of at least three real-world problems and explain how AI addresses them.
6. **Applications and Methods**
    - Discuss five significant applications of AI in different industries.
    - Explain the difference between rule-based methods and learning-based methods in AI.
7. **Early History of AI**
    - Trace the development of AI from its inception to the 1970s.
    - Highlight the contributions of early AI pioneers like Alan Turing and John McCarthy.
8. **Recent History of AI to the Present**
    - Discuss the major advancements in AI from the 1980s to the present.
    - How has the integration of big data and cloud computing influenced AI development?
9. **AI in the New Millennium**
    - Analyze the impact of AI technologies like machine learning, natural language processing, and robotics in the 21st century.
    - Discuss the societal challenges and benefits of AI in the new millennium.

Questions for Discussion (given in text book)

1. How would you define Artificial Intelligence?
2. Distinguish between strong and weak AI.
3. ALICE is the software that has won the Loebner Prize several times in the recent past. Go
4. online to find a version of this software. What can you tell us about ALICE?
5. What was Alan Turing's significant contribution to Artificial Intelligence?
6. What did John McCarthy contribute to Artificial Intelligence?
7. Why would an ATM and its programming not be a good example of AI programming?
8. Why is medical diagnosis a very typical and suitable domain for AI research?
9. Why have two-person games been a very suitable domain of study for AI?
10. Explain the role of computer chess with regard to AI.
11. What is an expert system?
12. Name three forms of knowledge representation.

Exercises (given in text book)

1. A variation of the Turing test is the so-called Inverted Turing test; in this test, a computer must determine whether it is dealing with a person or another computer. Can you envision any practical applications for this version of the Turing test? (Hint: In recent years, have you tried purchasing tickets for a popular sports or entertainment event online?)
2. A second variation of the Turing test is the Personal Turing test. Imagine you are trying to determine if you are communicating with your friend or with a computer pretending to be your friend. If a computer passes this test, what legal or ethical questions do you envision will arise?
3. Many people consider the use of language as a necessary attribute of intelligence. Koko is a gorilla trained by Dr. Francine Patterson of Stanford University in the use of American Sign Language. Koko was able to form word combinations for words unknown to her; for example, she represented the word ring by the words bracelet and finger, which she already knew. Does this gorilla's possession of knowledge modify your thinking on the subject of animal intelligence? If so, then in what ways? Could you envision an intelligence test for Koko?
4. Consider the following tests for a city to be considered a great city: - It should be possible to obtain a steak dinner at 3:00 a.m. - A classical music concert should be scheduled somewhere within the city bounds each evening. - A major sporting event should be scheduled each evening. Further, suppose a small town somewhere in America determines that they want to pass this test. To do so, they open a 24-hour steak joint and purchase a symphony orchestra and major sports franchise. Do you feel that this small town passes our litmus test for being a great city? Relate this discussion to the criteria for passing the original Turing test and the possession of intelligence (Dennett, 2004).
5. Suppose you want to design a threshold logic unit to emulate a two-input OR function. Can you determine a threshold and the weights to accomplish this task?
6. Suggest a strategy for the Iterated Prisoner's Dilemma, wherein the game is repeated n times for some unknown value n. How might you measure its success in the long run?

7. A genetic algorithm is to be employed to solve the instance of the 3-puzzle provided in the text. Suggest a string representation for a potential solution. What fitness function would you suggest?
8. Suggest a heuristic that would help to hail a taxi on a visit to New York City (or any other major city) during rush hour when taxis are scarce.
9. What heuristic do lions employ as they pursue their prey?
10. Suggest possible rules for an expert system designed to help select a suitable dog for your household.
11. Before Copernicus, the earth was considered to be the center of the heavens. After Copernicus, the earth was merely one of many planets circling the sun. Before Darwin, humans were considered to be a species apart (and above?) the rest of the living organisms on this planet. After Darwin, we were just another species of animals that had evolved from one-celled organisms. Suppose that in fifty years we have achieved human-level AI, and further suppose that successors to the robots Cog, Paro, and Kismet actually experience emotions rather than just pretending to do so. At such a point in our history, what claims will humans cling to as forming the core of their "specialness"? Are such claims essential or even desirable?
12. Suppose that at some time in the future, NASA plans an unmanned mission to Europa, a moon of the planet Jupiter. Suppose that at the time of launch our understanding of this moon's surface is scant. Suggest advantages to sending an "army" of Rodney Brooks-insect-type robots rather than one or two more substantial machines.
13. Should Eliza be considered a relational artifact? Defend your answer.
14. Listen to the song "Are We Human or Are We Dancer?" by Killers. What do you believe the lyrics of this song mean, and how do they relate to our course of study? You might wish to consult the lively online discussion (the song can be accessed on YouTube).
15. How would you define AI problems to be different from other types of problems? Name five problem-solving techniques typically used in AI?
16. Develop a new Turing Test for AI that would be applicable today.
17. Research the Lovelace 2 Robot (http://www.bbc.com/news/technology-30144069). Do you feel that the criteria for this new Turing Test for Robots is acceptable? How would you compare it to your answer in Question 2?

REFERENCES

Text Book

1. **Stephen Lucci** and Danny Kobec, Artificial Intelligence in the 21st Century – A living Introduction, Mercury Learning and Information, 2nd Edition 2016 (eBook available)