Introduction to Artificial Intelligence

Welcome to an exciting journey into the world of Artificial Intelligence! This course will equip you with the foundational concepts and practical skills needed to understand and build intelligent systems.

By Nazir Hossain Sahin

The Essence of Artificial Intelligence

AI is all about creating machines that can think, learn, and act like humans. Imagine computers that can solve complex problems, understand your language, and even play games better than any human!

It's not just sci-fi anymore; AI is revolutionizing industries from healthcare to entertainment, driving automation, enhancing accuracy, and providing powerful data analysis for better decision-making.

Problem Solving as Search

At its core, many AI problems boil down to finding the right path. We define a problem with a starting point, a goal, and possible actions. Think of it like navigating a maze!

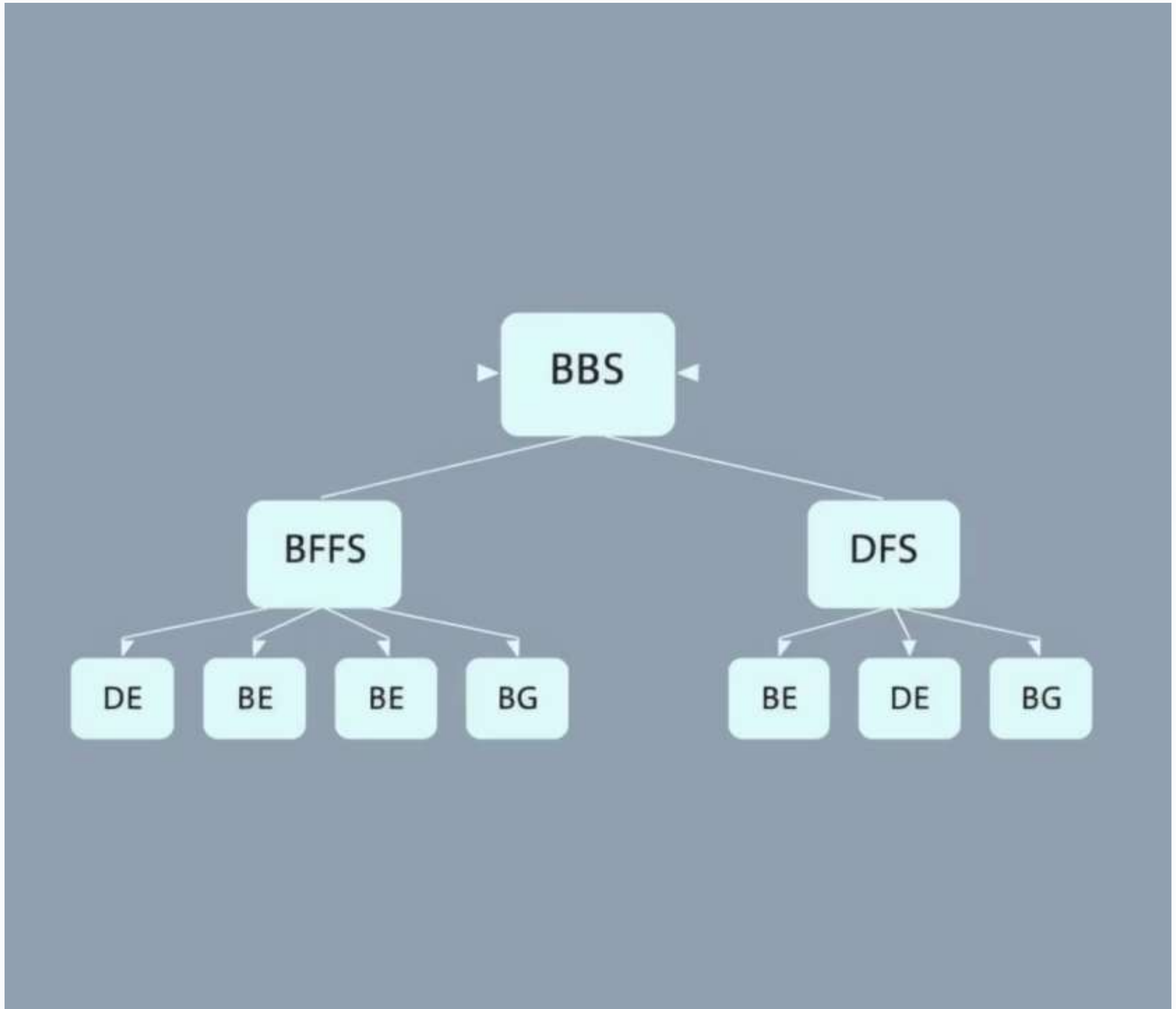
Uninformed Search

These are brute-force methods, like exploring every possible path without any special guidance. Think of Breadth-First Search (BFS) and Depth-First Search (DFS) as methodical explorers.

Informed Search

Here, we use "heuristics" – clever shortcuts or estimates – to guide our search. Algorithms like A* (A-Star) and Best-First Search use this domain-specific knowledge to find solutions more efficiently.

Lab Task 1: Visualizing Search Algorithms



In our first lab, you'll bring these concepts to life! You'll implement BFS and DFS, then visualize how they explore a given graph or tree. It's a great way to see how these algorithms behave in action.

```
def bfs(graph, start):    visited = []    queue = [start]
while queue:        node = queue.pop(0)        if node not
in visited:            visited.append(node)
neighbors = graph[node]            for neighbor in
neighbors:                queue.append(neighbor)    return
visited
```

A*: The Optimal Pathfinding Algorithm

1 The A* Formula: f(n) = g(n) + h(n)

A* is a superstar in pathfinding, combining the best of both worlds: it's efficient AND guaranteed to find the shortest path under the right conditions.

2 g(n): Cost from Start

This is the actual cost incurred to reach node 'n' from our starting point. Think of it as the distance already traveled.

3 h(n): Estimated Cost to Goal

This is our "heuristic" – an educated guess about the cost from node 'n' to the final goal. The better the guess, the faster A* works!

4 Admissibility: No Overestimation

For A* to always find the absolute shortest path, its heuristic must never overestimate the true cost to the goal. It's like having a reliable navigation system!

AI in Games: Thinking Strategically

Ever wondered how computers beat us at Chess? It's all about strategic thinking! We'll explore how AI agents navigate the competitive world of games, especially two-player, zero-sum games like Tic-Tac-Toe or Chess.

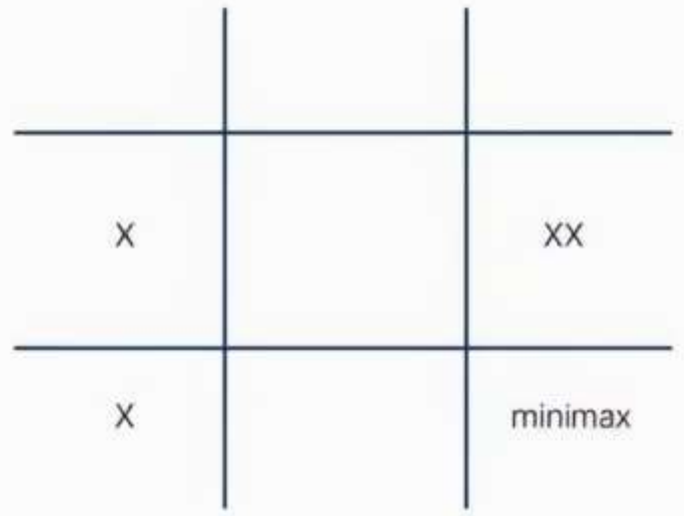
Minimax Algorithm

This core algorithm assumes your opponent is just as smart as you are and will always make the move that minimizes your score. It's about finding the best possible move assuming the worst-case scenario.

Alpha-Beta Pruning

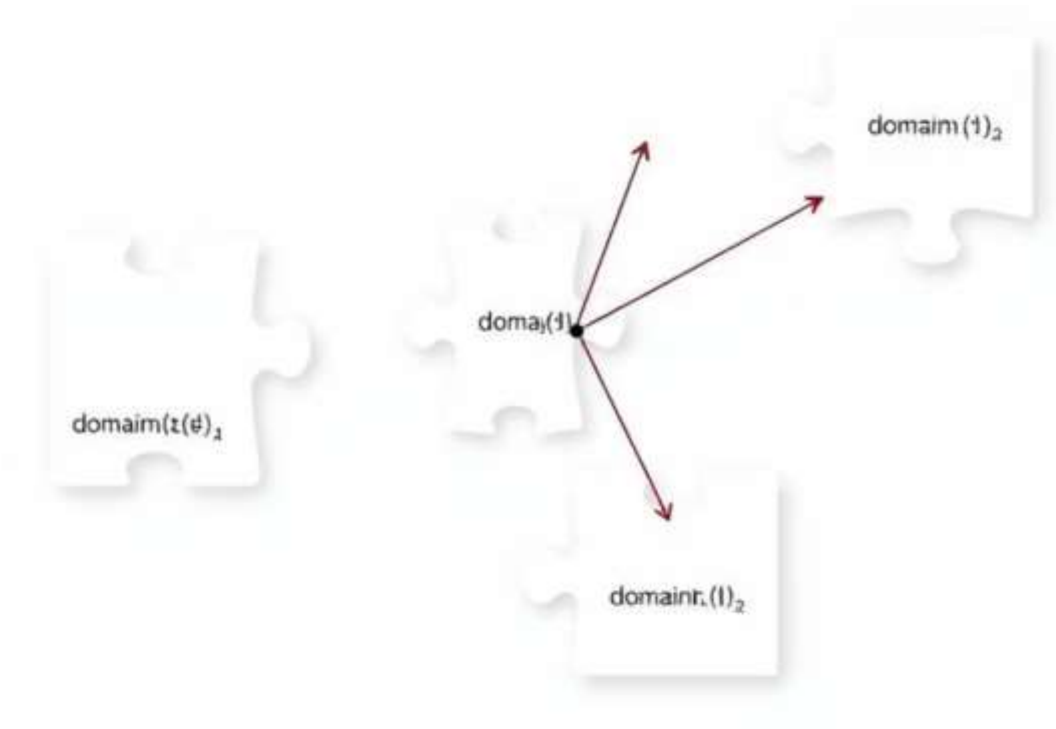
A genius optimization that dramatically speeds up Minimax. It allows the AI to "prune" or skip exploring branches of the game tree that it already knows won't lead to a better outcome.

Lab Task 2: Tic-Tac-Toe with Minimax



Get ready to build your own unbeatable opponent! In this lab, you'll implement the Minimax algorithm to create an AI player for Tic-Tac-Toe. You'll observe how the AI always makes the optimal move, leading to either a win or a draw.

This hands-on experience will solidify your understanding of adversarial search and how AI can achieve strategic mastery in games.



Solving Puzzles with Constraints

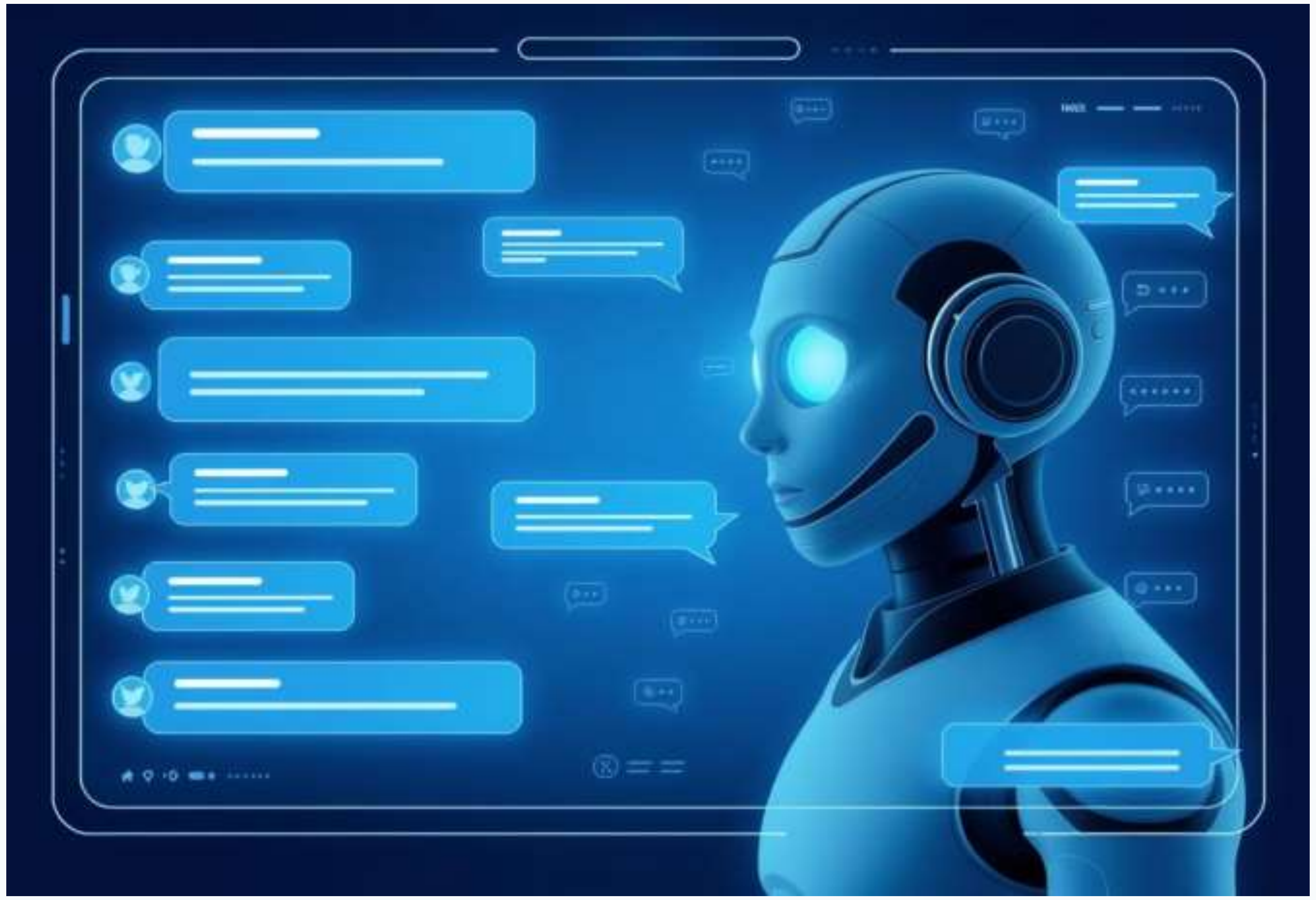
Constraint Satisfaction Problems (CSPs) are a fascinating area where AI tackles puzzles defined by variables, their possible values (domains), and specific rules (constraints).

Classic Example: Cryptarithmetic Puzzles

Think of "SEND + MORE = MONEY." Each letter is a unique digit, and the sum must hold true. CSP techniques help us crack these codes!

We'll explore techniques like Arc Consistency to prune impossible values, Backtracking to systematically explore solutions, and Constraint Propagation to automatically update related constraints.

AI in Action: NLP & Robotics



Natural Language Processing (NLP)

This field lets computers understand and generate human language. From chatbots that answer your questions to translation tools that bridge languages, NLP is everywhere!

Robotics

Giving machines eyes and hands! Robotics combines AI with engineering to create machines that perceive their environment and act in the physical world, from autonomous cars to robotic surgeons.

Tools of the Trade

Programming Language

Our course leverages **Python**, the industry-standard language for AI and machine learning, recognized for its simplicity and vast ecosystem.

Essential Libraries

- **NumPy** & **Pandas**: Foundational for efficient numerical operations and data analysis.
- **Scikit-learn**: Provides a comprehensive suite of classical machine learning algorithms.
- **TensorFlow** / **PyTorch**: Leading frameworks for building and deploying complex deep learning models.