Artificial Intelligence

Code	Credit Hours
CS- 370	3-1

Course Description

What do web search, speech recognition, face recognition, machine translation, autonomous driving, and automatic scheduling have in common? These are all complex real-world problems, and the goal of artificial intelligence (AI) is to tackle these with rigorous mathematical tools. In this course, you will learn the foundational principles that drive these applications. Specific topics include problem formulation, search, game playing, Markov decision processes, logic, and application of AI in robotics.

Textbook:

1. Stuart Russell and Peter Norvig. Artificial Intelligence: A Modern Approach. 4th edition. Prentice Hall. 2022

Reference Book:

1. Tom Mitchell, Machine Learning. McGraw-Hill.
2. Richard Sutton and Andrew Barto, Reinforcement Learning: An introduction. MIT Press, 1998 3. http://web.stanford.edu/class/cs221/
3. http://cs229.stanford.edu
4. http://www.robots.ox.ac.uk/~az/lectures/ml/
5. https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-034-artificial-intelligence-fall-2010/demonstrations/
6. http://web.stanford.edu/class/stats202/content/viewhw.html?hw48. http://www.cs.cmu.edu/~tom/mlbook-chapter-slides.html

Prerequisites

CS110 Fundamentals of Programming

ASSESSMENT SYSTEM FOR THEORY

Quizzes	10%
Assignments	10%
MSE	30%
ESE	40%
Project	10%

ASSESSMENT SYSTEM FOR LAB

Lab work	80%
Lab Project	20%

Teaching Plan

Week	Topics	Learning Outcomes
Week 1	Introduction to AI Ethical Aspects of AI and Biases	• Define artificial intelligence and differentiate it from related fields such as machine learning and deep learning. • Explain basic AI techniques and methodologies, including supervised learning, unsupervised learning, and reinforcement learning. • Describe various applications of AI across different industries, such as healthcare, finance, and automotive. • Evaluate the potential benefits and limitations of AI in real-world scenarios. • Gain hands-on experience with basic AI tools and platforms (e.g., TensorFlow, PyTorch, scikit-learn). • Execute simple AI experiments to understand practical implementations. • Identify and describe key ethical issues related to AI development and deployment, including fairness, accountability, transparency, and privacy. • Define AI bias and analyze its causes and consequences.
Week 2	Intelligent Agents & Concept of Rationality Types of Agents	• Define intelligent agents and describe their key characteristics. • Explain the concept of rationality in the context of AI and how it applies to agent behavior. • Identify and describe various types of agents (e.g., simple reflex agents, model-based agents, goal-based agents, utility-based agents). • Evaluate the strengths and limitations of each type of agent in different scenarios. • Develop criteria for rationality and apply them to design intelligent agents. • Implement basic rational agents in simulated environments.
Week 3	Problem Formulation, Problem-Solving Agents, Search Algorithms Uninformed Searching, BFS, DFS, UCS	• Understand the importance of problem formulation in AI. • Define state space, initial state, goal state, and operators for various problem-solving scenarios. • Describe the architecture and functioning of problem-solving agents. • Develop and test problem-solving agents using basic search algorithms.

		• Implement and compare different search algorithms, including uninformed (e.g., BFS, DFS, UCS) and informed search algorithms (e.g., Greedy BFS, A*).
Week 4	Uninformed Searching: IDS, Hill Climbing, Beam Search	• Define and explain Iterative Deepening Search (IDS), Hill Climbing, and Beam Search. • Describe the scenarios where each of these techniques is applicable. • Develop and test implementations of IDS, Hill Climbing, and Beam Search. • Analyze the strengths and weaknesses of each technique in solving complex problems. • Compare the effectiveness, efficiency, and limitations of IDS, Hill Climbing, and Beam Search. • Apply these techniques to various problem-solving scenarios to assess their practicality.
Week 5	Informed Searching Algorithms, Greedy BFS, A* Search, Weighted A* Memory Bounded Search	• Define informed search algorithms and explain how they use heuristics to improve search efficiency. • Describe the principles and operations of Greedy Best-First Search (BFS), A* Search, and Weighted A*. • Develop implementations of Greedy BFS, A* Search, and Weighted A* in a programming language. • Test these algorithms on different problem sets to evaluate their performance. • Compare the performance, optimality, and completeness of Greedy BFS, A*, and Weighted A*. • Define memory-bounded search techniques and their importance in handling large search spaces. • Describe key memory-bounded algorithms such as Iterative Deepening A* (IDA*) and Recursive Best-First Search (RBFS).
Week 6	Introduction to Games, Optimal decision in Games, Minimax Alpha-beta Pruning	• Define game theory concepts and their relevance to AI. • Explain optimal decision-making strategies in games. • Describe the Minimax algorithm and its application in two-player games. • Develop and test a Minimax implementation for a given game. • Evaluate the performance and effectiveness of Minimax in various game scenarios. • Discuss the limitations and challenges of using Minimax for complex games. • Define Alpha-Beta Pruning and its role in optimizing the Minimax algorithm. • Explain how Alpha-Beta Pruning reduces the search space in game trees.

		• Compare the efficiency of Minimax with and without Alpha-Beta Pruning.
Week 7	Cut-off Search, Evaluation functions Forward Pruning,	• Define cut-off search and its purpose in game AI. • Explain the role of evaluation functions in estimating the value of game states. • Develop cut-off search algorithms incorporating evaluation functions. • Test and refine evaluation functions to improve the performance of game-playing agents. • Analyze the trade-offs between search depth and accuracy in cut-off search. • Optimize evaluation functions to enhance decision-making in complex games.
Week 8	Stochastic Games, Expectiminimax	• Define stochastic games and explain how they differ from deterministic games. • Describe the Expectiminimax algorithm and its application in stochastic game scenarios. • Develop an implementation of the Expectiminimax algorithm. • Test the algorithm on various stochastic games to evaluate its performance. • Compare the effectiveness of Expectiminimax with other game-playing algorithms. • Assess the challenges and benefits of applying Expectiminimax in real-world stochastic games.
Week 9		
Week 10	Constraint Satisfaction Problem Backtracking, Arc Consistency, Inference in CSPs	• Understand the definition and properties of constraint satisfaction problems, including variables, domains, and constraints. • Learn to apply backtracking algorithms and techniques for enhancing efficiency such as arc consistency and other forms of inference. • Analyze the applicability and limitations of CSPs in solving real-world problems.
Week 11	Bayesian Networks	• Grasp the fundamentals of Bayesian networks, including their structure, and how they represent joint conditional probabilities. • Develop skills to perform probabilistic inference using Bayesian networks and understand the algorithms like variable elimination and belief propagation. • Explore the application of Bayesian networks in decision-making and prediction tasks.
Week 12	Propositional logic, syntax, semantics First order predicate logic	• Understand the syntax of propositional logic, including the formulation of well-formed formulas.

		• Comprehend the semantics of propositional logic, including truth tables and logical equivalences. • Learn methods for proving propositions using logical deductions and understand the use of logic in computer algorithms. • Understand the syntax and semantics of first-order predicate logic, including quantifiers and logical connectives. • Develop the ability to convert real-world scenarios into logical expressions using first-order logic. • Learn about the methods for reasoning and inference in first-order predicate logic, including unification and resolution.
Week 13	Genetic Algorithms	• Understand the principles of genetic algorithms, including selection, crossover, mutation, and fitness evaluation. • Learn to design and implement genetic algorithms for optimization problems. • Analyze the performance and convergence properties of genetic algorithms in various applications.
Week 14 - 15	Markov Decision Processes Policy Evaluation Value Iteration	• Grasp the fundamental components of Markov Decision Processes (MDPs), including states, actions, transition probabilities, and rewards. • Learn to formulate problems as MDPs and understand the role of policies in decision making. • Study solution methods such as dynamic programming for solving MDPs. • Understand the concept of policy evaluation in the context of reinforcement learning and its purpose in assessing the value of a given policy. • Learn computational techniques for policy evaluation, including iterative methods. • Analyze the convergence and efficiency of different policy evaluation methods. • Understand the value iteration algorithm for solving Markov Decision Processes and its convergence properties. • Learn to implement the value iteration algorithm and understand its computational aspects. • Explore the application of value iteration in practical decision-making scenarios.
Week 16	Reinforcement Learning	• Understand the fundamental concepts of reinforcement learning, including the exploration-exploitation tradeoff and reward systems. • Learn various model-based and model-free reinforcement learning algorithms and their applications.

		• Explore advanced topics in reinforcement learning such as deep reinforcement learning and policy gradient methods.
Week 17	Project Presentations	
Week 18	**ESE**	

Practical:

Lab No.	Description
Lab 1	Introduction to Python
Lab 2	Intelligent Agents
Lab 3	Uninformed Searching
Lab 4	Uniformed / Informed Searching
Lab 5	Informed Searching
Lab 6	Games: Alpha-Beta Pruning
Lab 7	Games
Lab 8	Stochastic Games
Lab 9	CSPs
Lab 10	Bayesian Networks
Lab 11	Logical Reasoning
Lab 12	Genetic Algorithms
Lab 13	Markov Decision Processes
Lab 14	Lab Exam