附件1：

|  |  |
| --- | --- |
| 课程名称（中英文） | 强化学习与控制Reinforcement Learning and Control |
| 课程先修条件  （中英文） | 掌握古典及现代控制理论、线性代数、概率论，具有最优化、最优控制的课程基础。最好了解一定的机器学习知识。  Proficient in classical and modern control theory, linear algebra, and probability theory, with a solid foundation in courses on optimization and optimal control. Familiarity with machine learning is preferred. |
| 课程大纲及  考核方式 | **Lesson 1: Overview of Reinforcement Learning (2 hours)**   1. Introduction to the concept and definition of artificial intelligence, including its development history, renowned scholars, and typical applications. 2. Overview of the development history of reinforcement learning, covering notable scholars, development patterns, and trends. 3. Introduction to typical applications of reinforcement learning, such as Tic-tac-toe, AlphaGo, robotics, and StarCraft. 4. Discussion of the main challenges faced by reinforcement learning.   **Lesson 2: Fundamentals of Reinforcement Learning (2 hours)**   1. Introduction to the RL framework and fundamental elements, including Markov Decision Processes (MDP), state and state space, action and action space, policy, reward function, and model, supplemented with examples. 2. Explanation of the theoretical foundations of RL, covering the state-value function (V) and action-value function (Q), deterministic and stochastic policies, the self-consistency condition, and Bellman's optimality principle.   **Lesson 3: Model-Free Learning - Monte Carlo Methods (2 hours)**   * Introduction to the basic principles of the Monte Carlo (MC) method, including MC estimation, the ε-greedy strategy, and applicable conditions. * Overview of on-policy Monte Carlo algorithms, covering the fundamental ideas and origins, ε-greedy strategy, algorithm framework, and key formulas. * Explanation of off-policy Monte Carlo algorithms, including the principle of importance sampling and comparing the advantages and disadvantages of off-policy and on-policy Monte Carlo methods.   **Lesson 4: Model-Free Learning - Temporal Difference Methods (2 hours)**   1. Introduction to the basic principles of Temporal Difference (TD) methods, including the differences between TD and Monte Carlo methods, comparing their advantages and disadvantages, key estimation formulas, and applicable conditions. 2. Overview of the Sarsa algorithm, including its framework, core formulas, and example problem-solving. 3. Explain the Q-learning algorithm, covering its framework and core formulas and comparing the differences, advantages, and disadvantages between it and Sarsa.   **Lesson 5: Model-Based Learning - Dynamic Programming (2 hours)**   1. Introduction to dynamic programming, including Bellman’s optimality principle and problem descriptions. 2. Explanation of the basic principles of dynamic programming, covering policy evaluation and policy improvement. 3. Overview the general dynamic programming framework and discuss the differences and connections between model-free RL and model-based RL.   **Lesson 6: Value Function Approximation (2 hours)**   1. Introduction to basic methods for value function approximation, including polynomial fitting, kernel function fitting, encoding methods, and neural networks. 2. Overview of unconstrained value function weight optimization methods, such as stochastic and batch gradient descent. 3. Explanation of constrained value function weight optimization methods, including the Lagrangian dual method, KKT conditions, and penalty function methods.   **Lesson 7: Policy Approximation Methods (2 hours)**   1. Introduction to policy gradient methods, a significant category of RL, discussing the differences, advantages, and disadvantages of policy gradient and value learning methods and the introduction of policy networks and algorithm applicability. 2. Overview of the development of policy gradient methods, including the principles of policy gradients, the Monte Carlo-based REINFORCE algorithm, and the introduction of baselines. 3. Detailed explanation of the Actor-Critic algorithm, including its framework, basic principles, and derivation of key formulas.   **Lesson 8: Deep Reinforcement Learning (2 hours)**   1. Introduction to deep learning, especially the basic principles and training process of deep neural networks. 2. Discussion of the challenges in deep reinforcement learning. 3. Explanation of DQN, including introducing Target Q networks and delayed update mechanisms, using experience replay, and its effectiveness. 4. Overview of other typical deep RL algorithms, including A3C, DDPG, and TRPO.   **Lesson 9: Approximate Dynamic Programming (2 hours)**   1. Explanation of the necessity of introducing models, analyzing the limitations and shortcomings of model-free RL, and the role of models. 2. Introduction to model-based RL algorithms for discrete-time systems, covering problem descriptions, derivation of key iterative formulas, frameworks, and a comparison with model-free RL. 3. Overview of model-based RL algorithms for continuous-time systems, including problem descriptions, introducing the HJB equation, and deriving key iterative formulas and frameworks.   **Lesson 10: State Constraints and Safety (2 hours)**   1. Introduction to Model Predictive Control (MPC), including problem descriptions, solution methods, and application cases. 2. Discuss the differences and connections between MPC and model-based RL, comparing objective functions, fundamental elements, and iterative formulas and exploring the potential integration of the two. 3. Explanation of the connections between model-based RL and optimal estimation.   **Lesson 11: Advanced Topics in Reinforcement Learning (Discussion, 2 hours)**   * First algorithm discussion session: group presentations (rotational presenters) and teacher-student discussions.  1. Choose any RL algorithm to present, focusing on classical papers (e.g., A3C, TRPO, Deep QN) or any RL-related topic, focusing on methodological summaries (e.g., RL convergence, on-policy/off-policy, MDP observability, MDP stationary distribution). 2. The class is divided into four groups, each with five members. One presenter is selected per group. Students present for 15 minutes, followed by a 10-minute discussion.   **Lesson 12: Applications of RL in Intelligent Vehicles, Games/Robotics, and Finance/Management (Discussion, 2 hours)**   * First RL application discussion session: group presentations (rotational presenters) and teacher-student discussions. Choose one of the following three application scenarios:  1. Intelligent vehicles, covering issues such as environmental perception, autonomous decision-making, or motion control (e.g., lane detection, vehicle recognition, highway lane-change decision-making, intersection collaborative driving, trajectory tracking control, lane-keeping control, adaptive cruise control). 2. Games/Robotics, covering topics like Tetris, Need for Speed, Dota, Go, Chess, Gomoku, StarCraft, bipedal robots, robotic vacuum cleaners, and space robotic arms. 3. Finance/Management, including stock price prediction, quantitative trading, portfolio management, option pricing, high-frequency trading, electronic currencies, peer-to-peer lending, blockchain, and shared services. 4. Focus on the basic concepts, task decomposition, initial solutions, and the advantages and disadvantages of applying RL to these problems. 5. The class is divided into four groups, each with five members. One presenter is selected per group. Students present for 15 minutes, followed by a 10-minute discussion.   **Lessons 13–15: Applications of RL (Midterm Discussions, 2 hours each)**   * Follow-up RL application discussions, focusing on methods, results, and simulation verification.  1. Introduce RL applications in specific scenarios, covering problem formulation, design of state/action spaces and reward functions, algorithm selection, learning acceleration techniques, and strategies for handling specific situations. 2. Discuss simulation validation results, recommending platforms like Matlab, C++, or Python for simulation.   **Lesson 16: Project Defense and Showcase (Defense, 2 hours)**   * Students present their RL projects in groups using PPT, with feedback from teachers and teaching assistants.  1. Projects should be based on the RL algorithm research or application examples discussed in previous sessions. 2. The class is divided into four groups, each with five members, with each member presenting (<5 minutes per person)—group presentation: 25 minutes; Q&A: 5 minutes.   Final reports must be submitted within one week of the course ends, adhering to the IEEE Journal format (double column, maximum 10 pages).  Assessment Methods: Course Assignments + Team Project + Class Performance |