Slide 1: Introduction

Deep Reinforcement Learning (DRL): A subset of reinforcement learning (RL) that combines RL with deep learning techniques to enable agents to make decisions from high-dimensional inputs.

Deep Q-Learning (DQL): A specific type of DRL algorithm that leverages Q-learning principles and neural networks for function approximation.

Slide 3: Deep Reinforcement Learning

Function Approximation: The use of models (e.g., neural networks) to approximate complex mathematical functions.

State Space: The set of all possible states an agent can encounter in its environment.

Slide 4: High-Dimensional Sensory Inputs

Convolutional Networks: Neural networks designed for processing grid-like data such as images.

Multilayer Perceptrons: A class of fully connected feedforward neural networks.

Slide 5: RL vs. Supervised Learning

Hand-Labeled Data: Data manually annotated for supervised learning tasks.

Reward Signal: The feedback provided to an RL agent based on its actions.

Slide 6: Differences in Data

Stationary Data Distribution: Data that does not change over time, typical in supervised learning.

Dependent Data Sequence: Data points that are sequentially dependent, common in RL.

Slide 7: DRL Algorithm Types

Policy-Based Methods: Algorithms that optimize the policy directly.

Value-Based Methods: Algorithms that focus on optimizing a value function.

Actor-Critic: Combines policy and value-based approaches.

Slide 8: Model-Based and Model-Free Methods

Model-Based RL: Relies on a model of the environment for planning and decision-making.

Model-Free RL: Does not require a model of the environment; learns directly from interaction.

Slide 9-10: Deep Q-Learning (DQL)

Q-Value: Represents the expected cumulative reward for taking a certain action in a given state.

Forward Pass: A process in neural networks where input data is propagated forward to compute the output.

Slide 11-12: Experience Replay

Experience Replay: A mechanism to store and reuse past experiences to break correlation in data and stabilize learning.

Replay Buffer: A storage for past experience tuples (s, a, r, s′, a′ ).

Slide 13: Challenges in DQL

History Representation: Fixed-length representation of past states for stable learning.

Target Instability: Issues caused by rapidly changing target values during training.

Slide 14-16: Value Function Approximation

Reward Signal: A numerical value representing the outcome of an action.

Finite Markov Decision Process (MDP): A framework where outcomes depend only on the current state and action.

Slide 17-19: Optimization and Loss

Objective Function: A mathematical function used to guide learning by minimizing loss.

Stochastic Gradient Descent (SGD): An optimization algorithm that updates weights iteratively based on a subset of data.

Slide 20-22: Target Network

Prediction Network: Computes predicted Q-values.

Target Network: Used to compute stable target values for training.

Slide 23: Input Preprocessing

Dimensionality Reduction: Simplifying input data, such as converting RGB images to grayscale or resizing.

Slide 24-27: Training and Atari Games

Epsilon-Greedy Strategy: Balances exploration and exploitation by selecting random actions with probability epsilon.

Frame Skipping: A technique to speed up training by skipping frames while retaining the previous action.

Slide 28-32: Implementation Examples

Q-Network: A neural network that estimates Q-values for state-action pairs.

Control Fragments: Precomputed sequences of actions used to control complex systems.

Slide 33: Advantages and Shortcomings of DQL

Generalization: The ability of a model to perform well on unseen states.

Continuous State/Action Spaces: Scenarios where states or actions are not discrete, making DQL less effective.