Deep Q-Learning is a reinforcement learning algorithm that combines Q-Learning with Deep Learning to solve complex decision-making problems. It allows an agent to learn how to act optimally in environments with large state spaces by approximating a function, known as the *Q-function*, which evaluates the quality of an action taken in a given state.

Q-function

The Q-function, $Q(s, a)$, represents the expected cumulative reward an agent will receive after taking action a in state s, and then following the optimal policy. The cumulative reward is computed as:

$$Q(s, a) = r + \gamma \max_{a'} Q(s', a'),$$

Where:

- r is the immediate reward received after taking action a in state s.
- s' is the next state reached.
- a' is the next action.
- $\gamma \in [0, 1]$ is the discount factor, which balances immediate and future rewards.

Key Techniques

- **Replay Buffer**: A memory that stores past experiences (s, a, r, s'). Randomly sampling experiences from the buffer during training reduces correlations between consecutive samples, improving learning stability.
- **Exploration-Exploitation Balance**: The agent uses an ϵ-greedy policy to choose actions, where it explores randomly with probability ϵ and exploits the best-known action otherwise.

High-Level Workflow

1. Observe the current state s.
2. Choose an action a using an ϵ-greedy policy.
3. Execute the action, observe the reward r and next state s'.
4. Store the experience (s, a, r, s') in the replay buffer.
5. Sample a mini-batch of experiences from the buffer to train the Q-network.