Assignment 4 - Reinforcement Learning

Name: Abdullah Al Noman
UTA ID: 1002286368
Course & Section: 2258-CSE-6363-007

Introduction

This assignment explores the evolution of reinforcement learning from simple tabular methods to deep neural architectures. In Part 1, a Q-Learning agent learns the optimal policy in the discrete `FrozenLake-v1` environment. In Part 2, the same principle is extended to a continuous state space via Deep Q-Learning on Atari Breakout. The objective is to understand how agents learn through interaction and how hyperparameters affect learning outcomes.

1 Part 1 — Q-Learning and Policy Iteration on FrozenLake

Objective

- Implement tabular Q-Learning for `FrozenLake-v1`.
- Evaluate the impact of key hyperparameters: learning rate (α), discount factor (γ), and exploration rate (ϵ).
- Compare the learned policy with a Policy Iteration baseline.

Methodology

The environment was initialized using:

```
gym.make("FrozenLake-v1", is_slippery=True)
```

The Q-Learning update rule is defined as:

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)]$$

An ϵ-greedy policy with exponential decay controlled exploration. The following hyperparameters were tested:

- $\alpha \in \{0.1, 0.5, 0.9\}$
- $\gamma \in \{0.80, 0.95, 0.999\}$
- ϵ: $(1.0 \rightarrow 0.1, 0.999)$, $(1.0 \rightarrow 0.01, 0.9995)$, $(0.5 \rightarrow 0.1, 0.9999)$

Evaluation was performed over 3,000 greedy-policy rollouts. A Policy Iteration baseline using explicit transition probabilities (`env.P`) was also implemented.

Results

The Q-Learning agent achieved a mean success rate of ≈ 0.74 after 25,000 episodes, while the Policy Iteration baseline achieved ≈ 0.82, showing convergence to the optimal policy.

Effect of Hyperparameters:

- **Learning rate (α)**: Low α slowed convergence, while high α caused instability. $\alpha = 0.5$ yielded the best stability-speed balance.
- **Discount factor (γ)**: Higher γ improved long-term planning, with $\gamma = 0.95$–0.99 performing best.
- **Exploration rate (ϵ)**: Gradual decay ($1.0 \rightarrow 0.1, 0.999$) promoted steady learning. Faster decay led to premature exploitation.

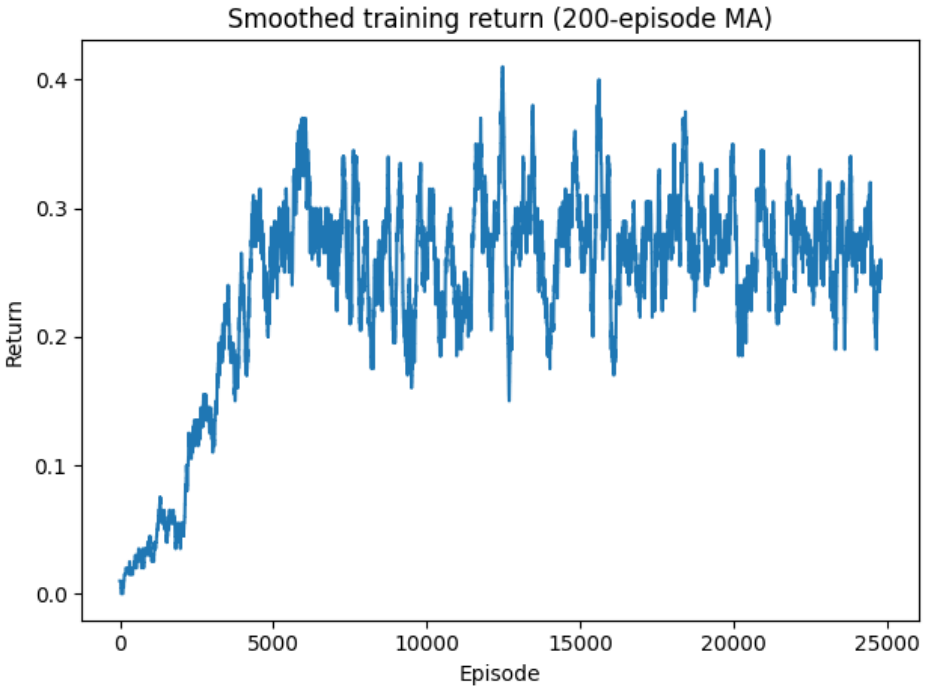

Figure 1: Smoothed training return (200-episode moving average) for Q-Learning on FrozenLake.

Figure 2: Greedy policy success rate over training episodes for Q-Learning on FrozenLake.

Discussion

Q-Learning approximated the optimal policy obtained from Policy Iteration. The results highlight that moderate α, high γ, and controlled ϵ decay yield effective learning. Despite environmental stochasticity, the agent successfully learned a stable navigation strategy.

2 Part 2 — Deep Q-Learning on Atari Breakout

Objective

Adapt the Q-Learning framework to Deep Q-Learning on a visual state space, using the Atari `Breakout` environment.

Methodology

A Dueling Double DQN architecture was implemented using PyTorch:

- **Convolutional Layers:** 84×84 grayscale preprocessing for spatial feature extraction.
- **Dueling Streams:** Separate value (V) and advantage (A) estimators combined as $Q(s,a) = V(s) + (A(s,a) - \bar{A})$.
- **Double DQN:** Mitigated Q-value overestimation via decoupled target selection and evaluation.
- **Experience Replay:** 1M capacity for uncorrelated sampling.

- **Target Network:** Updated every 30k steps.

Training used RMSProp (lr = 5e-5, $\gamma = 0.99$) for 300,000 steps.

Results

Example training logs:

```
Step 250,000 | eps=0.050 | recent avg return=4.05 | replay=200,000 | device=cuda
Step 300,000 | eps=0.050 | recent avg return=9.75 | replay=200,000 | device=cuda
Evaluation: mean return = 2.20 ± 1.33 (N=10 episodes)
```

The agent learned basic paddle control and ball interception, showing clear improvement in score trends. Longer training (1M+ steps) is expected to reach human-level performance.

Discussion

Compared to tabular Q-Learning, DQN scales effectively to high-dimensional visual spaces. Key improvements include:

- **Dueling Network:** Better value estimation.
- **Double Q-Learning:** Reduced overestimation bias.
- **Replay Buffer:** Stabilized learning via decorrelation.
- **Target Network:** Improved stability.

Comparative Analysis

Aspect	**FrozenLake (Q-Learning)**	**Atari Breakout (DQN)**
State Representation	Discrete (16 states)	High-dimensional pixel frames
Value Approximation	Tabular Q-table	CNN with Dueling Architecture
Exploration Strategy	ϵ-greedy	Decaying ϵ with replay buffer
Convergence Time	Few thousand episodes	Hundreds of thousands of steps
Performance Metric	Success rate (goal reached)	Average return (game score)

Table 1: Comparison between Tabular and Deep Q-Learning approaches.

Overall Conclusion

This study demonstrates the progression from tabular to deep reinforcement learning. Q-Learning on FrozenLake confirmed theoretical principles of exploration, value updates, and hyperparameter tuning, while Deep Q-Learning on Atari illustrated scalability using function approximation. Together, they show how RL agents can learn optimal behaviors in both simple and complex environments through iterative policy refinement.

3 References

1. Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction* (2nd ed.). MIT Press.

2. Mnih, V., et al. (2015). "Human-Level Control Through Deep Reinforcement Learning," *Nature*, 518(7540), 529–533.

3. Farama Foundation (2023). *Gymnasium Documentation.* Retrieved from https://gymnasium.farama.org