Assignment 3 – Deep Q-Learning on Pong

Name: Preetpal Singh
StudentID: 8804336

1. Introduction

This report presents the implementation, experimentation, and evaluation of a Deep Q-Network (DQN)

applied to the Atari game PongDeterministic-v4. The goal is to understand how hyperparameters—specifically

batch size and target network update frequency—affect learning stability, convergence, and performance.

2. Network Architecture

The network is based on the classic DeepMind 2015 DQN architecture, adapted for Pong.

The architecture processes 4 stacked frames (grayscale, cropped, downsampled, normalized)

to capture temporal movement.

• Input: State shape (4 × 80 × 80)

• Conv1: 4→32, kernel 8×8, stride 4

• Conv2: 32→64, kernel 4×4, stride 2

• Conv3: 64→64, kernel 3×3, stride 1

• Flatten: 2688 units

• FC1: 2688→512

• FC2: 512→6 (legal Atari Pong actions)

• Activations: ReLU throughout

• Loss: Mean Squared Error (MSE)

• Optimizer: Adam (learning rate 1e-4)

• Discount factor: $\gamma = 0.95$

• Replay buffer: 50,000 transitions

• Target network update: Varied by experiment

3. Experimental Setup

Three main experiments were conducted:

A. Baseline

– Batch size = 8

– Target update = 10 episodes

B. Batch Size Variation

– Batch size = 16

– Target update = 10 episodes

C. Target Update Frequency Variation

– Batch size = 8

– Target update = 3 episodes

Each experiment was run for multiple episodes to analyze effects on reward progression and moving-average stability.

4. Metrics Collected

The following metrics were collected:

1. Episode Reward

– Raw reward for each episode.

– Pong reward ranges between -21 and +21.

2. Moving Average Reward (window = 5 episodes)

– Smooths noise and highlights convergence trends.

3. Qualitative Stability Indicators

– Oscillations, divergence, plateau regions, and improvement rate.

These metrics allow comparison of learning speed and policy stability.

5. Results & Observations – Baseline (Batch=8, Target Update=10)

The baseline configuration represents standard DQN training conditions.

Observations:

• Rewards began near -21, typical of random play.

• Gradual improvement was observed after several episodes.

• The moving average curve was noisy due to low batch size.

• Target network updates every 10 episodes provided stable Q-value targets.

• Convergence was slow but consistent.

Conclusion:

This setup provides moderate stability with slower learning.

6. Results & Observations – Batch Size = 16

Increasing the batch size reduces gradient noise.

Observations:

• Reward curve smoothed significantly.

• Gradient updates became more stable.

• The agent began improving earlier than in the baseline condition.

• Training speed (per update) is slower due to larger batch processing,

but convergence rate (per episode) improved.

Conclusion:

Batch size 16 improves performance stability and early learning speed.

7. Results & Observations – Target Update = 3 Episodes

Reducing target network update interval makes Q-targets follow the main network too closely.

Observations:

• Learning curve showed oscillations and instability.

• Q-values tended to overshoot due to rapid target updates.

• Some short-term bursts of improvement were observed but not maintained.

Conclusion:

Updating the target network too frequently harms stability.

8. Comparative Summary

Experiment	Batch Size	Target Update	Stability	Speed	Final Performance
Baseline	8	10	Medium	Medium	Moderate
Batch=16	16	10	High	Fast	Highest
Target=3	8	3	Low	Unstable	Low

Key Insight:

• Larger batch size improves stability and performance.

• Smaller target update frequency increases instability.

• Best configuration: Batch size 16, Target update = 10.

9. Final Conclusion

This assignment demonstrated the impact of key hyperparameters on DQN performance.

Through experimentation, it was shown that larger batch sizes improve stability,

while overly frequent target network updates harm convergence.

The best performing configuration was:

• Batch size = 16

• Target update = 10

This setup achieved the highest moving average reward and exhibited strong training stability.