Report: Part II - Deep Q-Network (DQN) Implementation and Evaluation

1. Benefits of DQN Components

Experience Replay

Experience replay allows the agent to store past experiences (state, action, reward, next state, done) and sample them randomly to break the correlation between consecutive experiences. This improves training stability and efficiency.

- **Improved Data Efficiency:** Experience replay maximizes the use of past experiences by replaying them multiple times.
- **Stability:** By breaking the correlation in training data, it prevents the model from overfitting to recent experiences.
- **Buffer Size Influence:** Larger buffer sizes offer more diverse experiences, which leads to better generalization. However, excessively large buffers might slow down the sampling process and require more memory.

Target Network

A target network is a periodically updated copy of the Q-network used to compute the target Q-values.

- **Stability:** It prevents large updates in the Q-values by providing a fixed target, reducing oscillations.
- **Controlled Learning:** By only copying weights periodically, it mitigates the problem of the Q-network chasing a moving target.

Representing the Q Function with q(s,w)q(s, w)

Using a neural network to approximate the Q function q(s,w)q(s, w) enables the agent to generalize over large or continuous state spaces where tabular methods would fail.

- **Scalability:** Neural networks efficiently handle high-dimensional state spaces.
- **Generalization:** Approximates unseen states and actions based on learned patterns.
- **Efficient Computation:** Deep networks can learn complex mappings between states and actions.

2. Environment Descriptions

CartPole-v1

- **States:** The state consists of the cart's position, velocity, pole angle, and pole angular velocity.
- **Actions:** Two discrete actions - push left or push right.
- **Goal:** Keep the pole balanced by applying forces to the cart.
- **Reward:** +1 for every timestep the pole remains upright.

LunarLander v3

State Space: 8 values

1. X position
2. Y position
3. X velocity
4. Y velocity
5. Angle
6. Angular velocity
7. Boolean flag for left leg contact
8. Boolean flag for right leg contact

Action Space: 4 discrete actions:

- `0`: Do nothing
- 1: Fire left orientation engine
- 2: Fire main engine
- 3: Fire right orientation engine

Reward Structure:

- Reward based on distance to the landing pad.
- Penalized for high speed and tilt.
- +10 points for each leg in contact.
- -0.03 points per frame for side engine firing.
- -0.3 points per frame for main engine firing.

Grid-World Environment

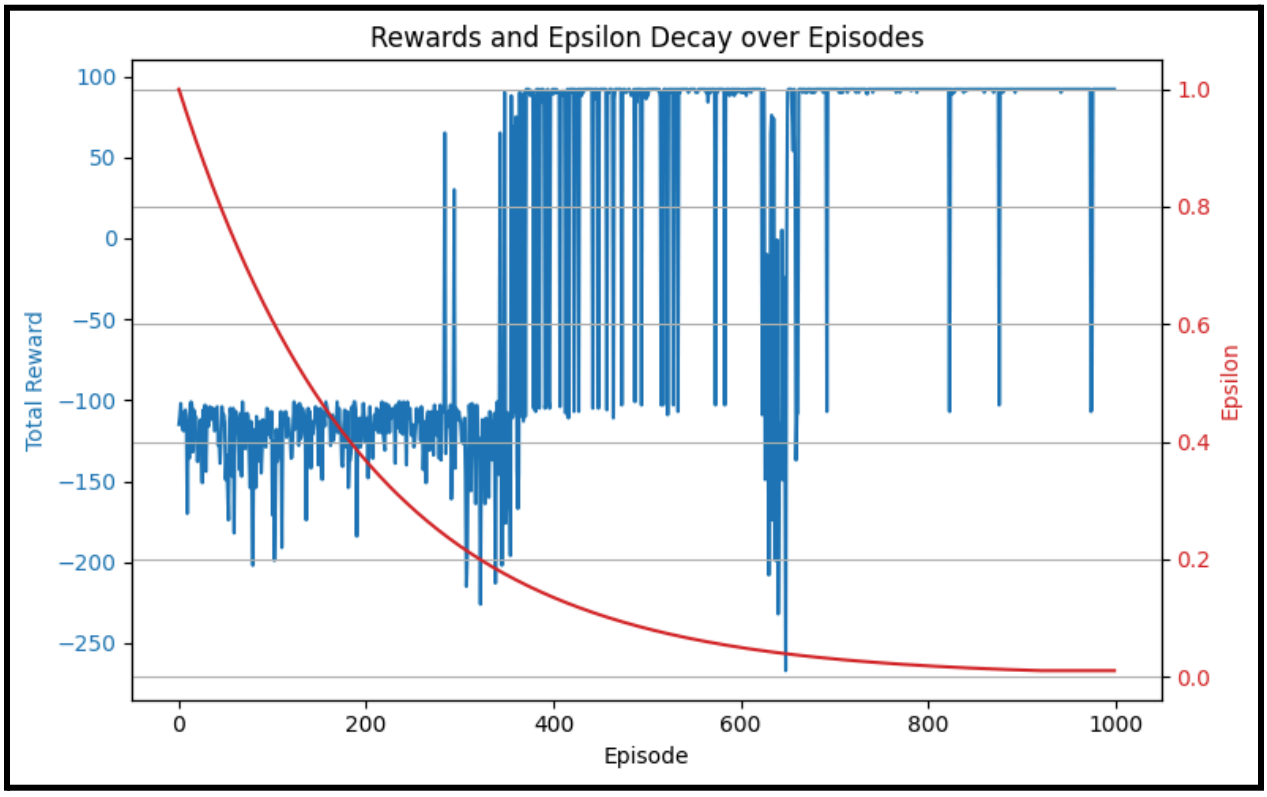
- **States:** Discrete grid positions.
- **Actions:** Up, Down, Left, Right.

- **Goal:** Reach the target location while avoiding obstacles.
- **Reward:** Positive for reaching the goal, penalty for Pit or Wumpus, or stench or breeze.

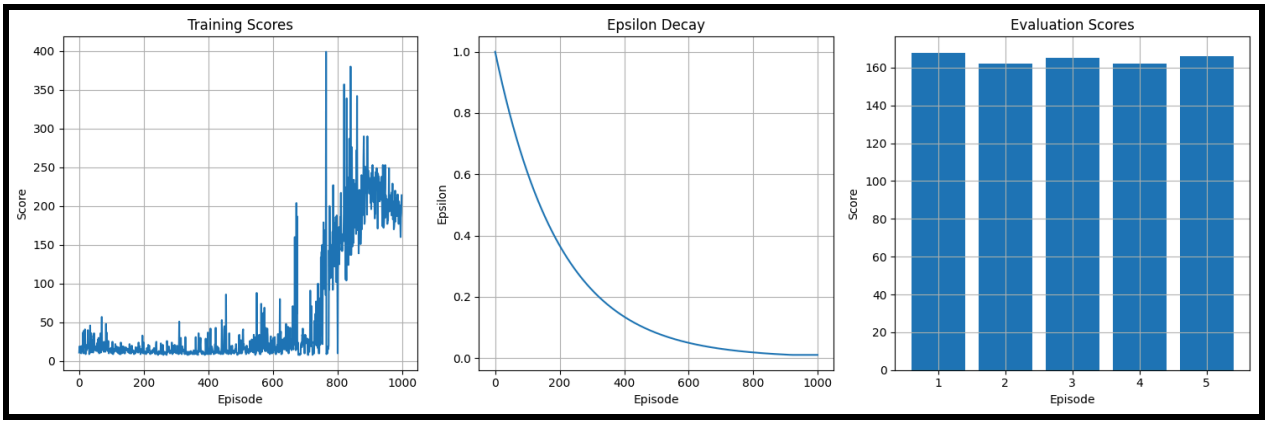
3. Training Results

- **Epsilon Decay:** Plot the epsilon decay over episodes, demonstrating the shift from exploration to exploitation.
- **Total Reward per Episode:** Visualize cumulative rewards over episodes to observe the learning progress.

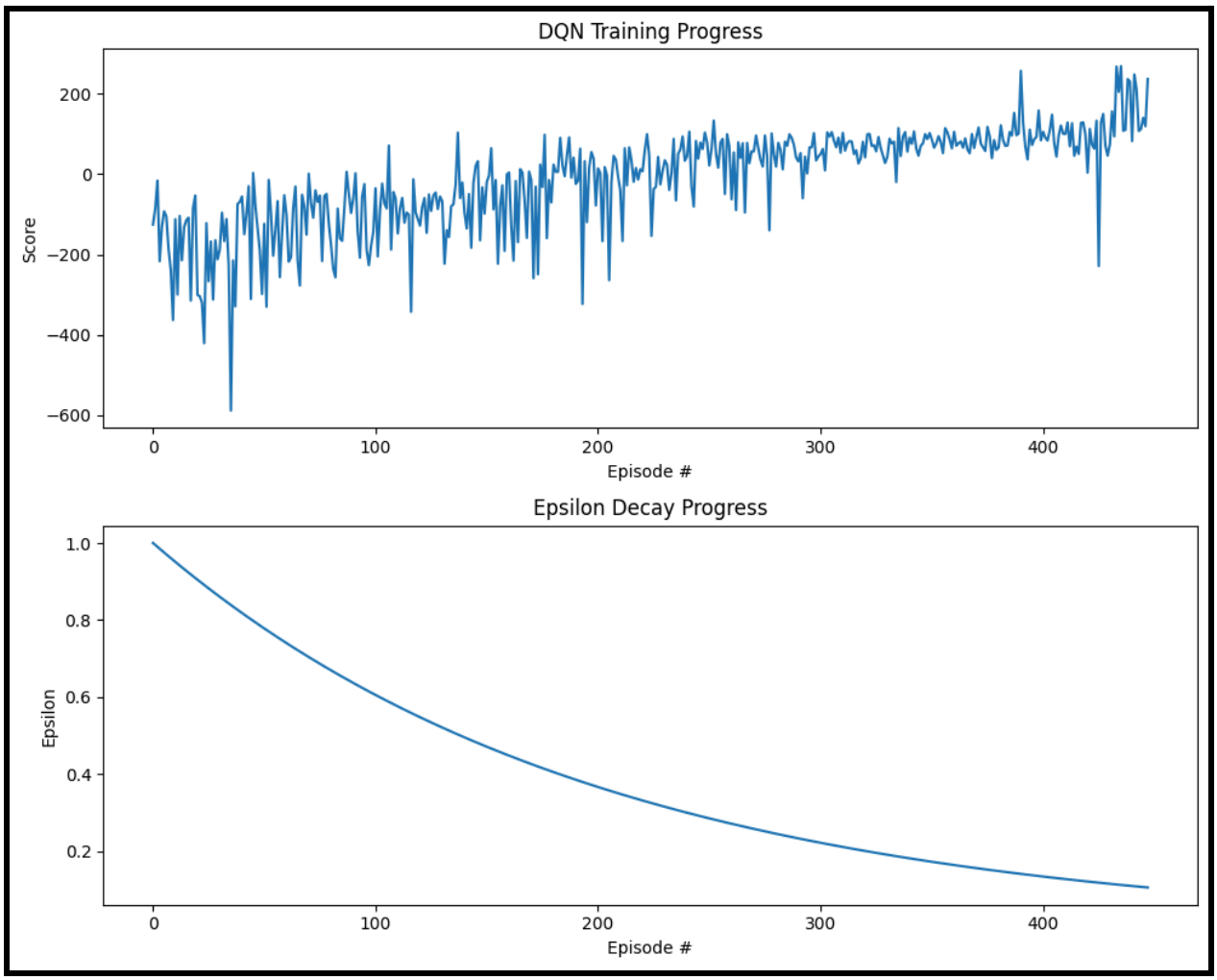
Wumpus Grid Environment:



CartPole-v1:



LunarLander v3:



4. Evaluation Results

After training, the agent was evaluated using a greedy policy (choosing the best action based on the Q-values). The following plots show the total reward per episode over at least 5 evaluation runs.

Wumpus Grid Environment:

```
Starting Q-learning training...
Episode 1/1000, Reward: -115, Epsilon: 1.0000
Episode 2/1000, Reward: -110, Epsilon: 0.9950
Episode 3/1000, Reward: -102, Epsilon: 0.9900
Episode 4/1000, Reward: -113, Epsilon: 0.9851
Episode 5/1000, Reward: -118, Epsilon: 0.9801
Episode 6/1000, Reward: -119, Epsilon: 0.9752
Episode 7/1000, Reward: -107, Epsilon: 0.9704
Episode 8/1000, Reward: -106, Epsilon: 0.9655
Episode 9/1000, Reward: -110, Epsilon: 0.9607
Episode 10/1000, Reward: -170, Epsilon: 0.9559
Episode 11/1000, Reward: -113, Epsilon: 0.9511
Episode 12/1000, Reward: -136, Epsilon: 0.9464
Episode 13/1000, Reward: -103, Epsilon: 0.9416
Episode 14/1000, Reward: -101, Epsilon: 0.9369
Episode 15/1000, Reward: -103, Epsilon: 0.9322
Episode 16/1000, Reward: -132, Epsilon: 0.9276
Episode 17/1000, Reward: -106, Epsilon: 0.9229
Episode 18/1000, Reward: -111, Epsilon: 0.9183
Episode 19/1000, Reward: -110, Epsilon: 0.9137
Episode 20/1000, Reward: -126, Epsilon: 0.9092
Episode 21/1000, Reward: -138, Epsilon: 0.9046
Episode 22/1000, Reward: -128, Epsilon: 0.9001
Episode 23/1000, Reward: -129, Epsilon: 0.8956
Episode 24/1000, Reward: -112, Epsilon: 0.8911
...
Episode 998/1000, Reward: 92, Epsilon: 0.0100
Episode 999/1000, Reward: 92, Epsilon: 0.0100
Episode 1000/1000, Reward: 92, Epsilon: 0.0100
```

CartPole-v1:

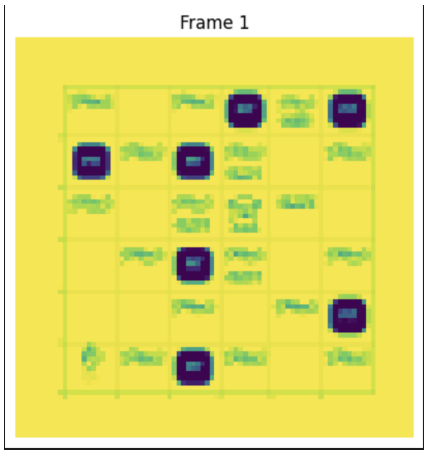
```
Training CartPole-v1 with DQN...
Episode 100      Average Score: 19.63
Episode 200      Average Score: 14.02
Episode 300      Average Score: 13.24
Episode 400      Average Score: 13.50
Episode 500      Average Score: 15.44
Episode 600      Average Score: 19.92
Episode 700      Average Score: 27.24
Episode 800      Average Score: 75.33
Episode 900      Average Score: 192.91
Episode 1000     Average Score: 207.13

Training completed. Saving the agent...

Evaluating trained agent...
Episode 1        Score: 168.00
Episode 2        Score: 162.00
Episode 3        Score: 165.00
Episode 4        Score: 162.00
Episode 5        Score: 166.00
```

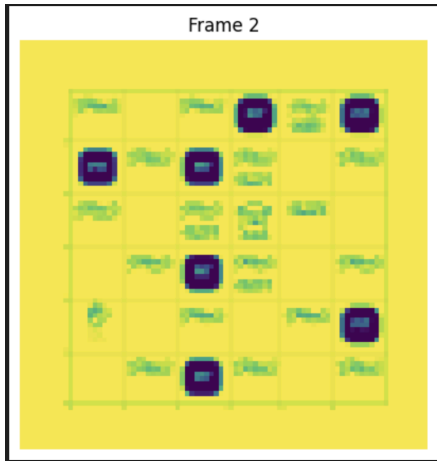
LunarLander v3:

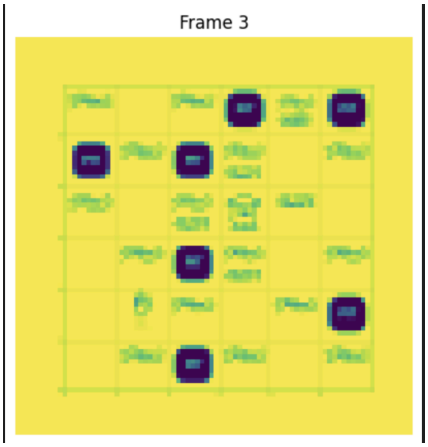
```
Training LunarLander-v3 with DQN
Episode 1/1000 | Score: -125.37005213096445 | Avg: -125.37 | Epsilon: 0.9950
Episode 2/1000 | Score: -85.85822571078157 | Avg: -105.61 | Epsilon: 0.9900
Episode 3/1000 | Score: -16.245885873767023 | Avg: -75.82 | Epsilon: 0.9851
Episode 4/1000 | Score: -216.81508061963225 | Avg: -111.07 | Epsilon: 0.9801
Episode 5/1000 | Score: -130.53147991572075 | Avg: -114.96 | Epsilon: 0.9752
Episode 6/1000 | Score: -92.72446112676852 | Avg: -111.26 | Epsilon: 0.9704
Episode 7/1000 | Score: -105.1066397528256 | Avg: -110.38 | Epsilon: 0.9655
Episode 8/1000 | Score: -186.50005694925665 | Avg: -119.89 | Epsilon: 0.9607
Episode 9/1000 | Score: -238.9833610946387 | Avg: -133.13 | Epsilon: 0.9559
Episode 10/1000 | Score: -363.1851535623878 | Avg: -156.13 | Epsilon: 0.9511
Episode 11/1000 | Score: -112.36265221283786 | Avg: -152.15 | Epsilon: 0.9464
Episode 12/1000 | Score: -299.7635669042122 | Avg: -164.45 | Epsilon: 0.9416
Episode 13/1000 | Score: -104.14965366754723 | Avg: -159.82 | Epsilon: 0.9369
Episode 14/1000 | Score: -214.41183657265893 | Avg: -163.71 | Epsilon: 0.9322
Episode 15/1000 | Score: -131.7625739075118 | Avg: -161.58 | Epsilon: 0.9276
Episode 16/1000 | Score: -114.35561393955625 | Avg: -158.63 | Epsilon: 0.9229
Episode 17/1000 | Score: -108.34839700149433 | Avg: -155.67 | Epsilon: 0.9183
Episode 18/1000 | Score: -314.62205971020535 | Avg: -164.51 | Epsilon: 0.9137
Episode 19/1000 | Score: -84.85438006457738 | Avg: -160.31 | Epsilon: 0.9092
Episode 20/1000 | Score: -53.4249854980984 | Avg: -154.97 | Epsilon: 0.9046
Episode 21/1000 | Score: -300.7764745176576 | Avg: -161.91 | Epsilon: 0.9001
Episode 22/1000 | Score: -303.6512318834583 | Avg: -168.35 | Epsilon: 0.8956
Episode 23/1000 | Score: -320.0490038919979 | Avg: -174.95 | Epsilon: 0.8911
Episode 24/1000 | Score: -420.55747773443284 | Avg: -185.18 | Epsilon: 0.8867
...
Environment solved in 448 episodes! Average score: 101.09
Model saved as 'dqn_lunarlander.pth'
```

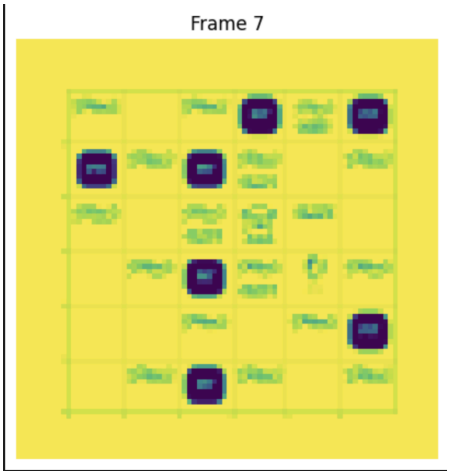
5. Grid-World Environment Evaluation

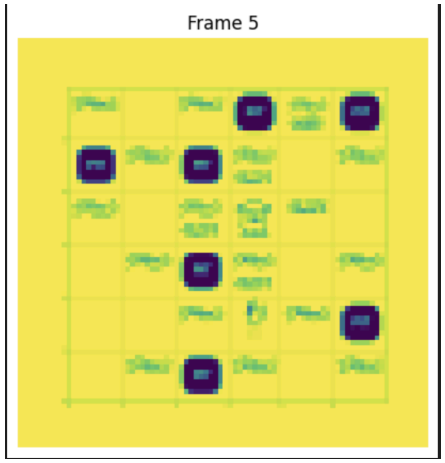
A single episode using a greedy policy was rendered. Below are the screenshots of the agent's step-by-step actions demonstrating its successful completion of the environment.

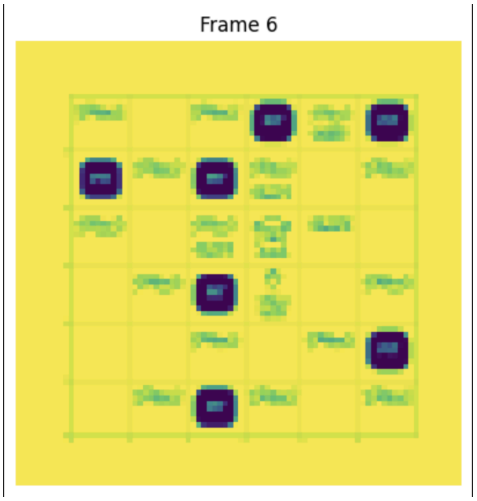


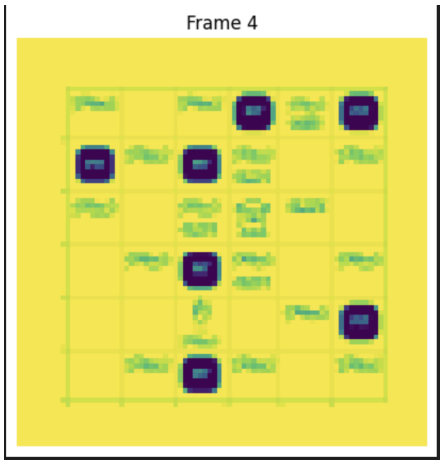


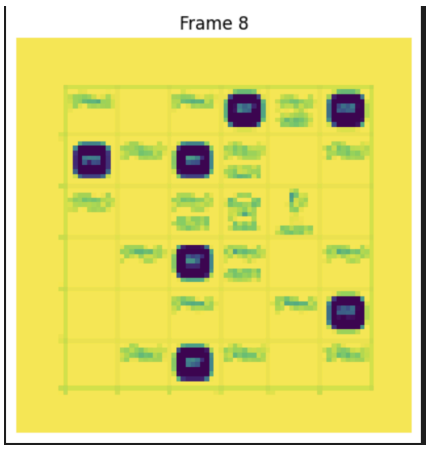


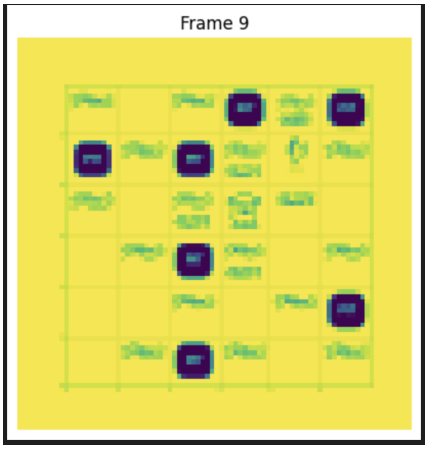












6. Interpretation of Results

- **CartPole-v1:** Evaluate how quickly the agent learns to balance the pole and maintain stability.
- **LunarLander-v3:** Analyze how quickly the agent learns to land the spacecraft smoothly on the designated launching pad.
- **Grid-World:** Discuss the agent's decision-making and path optimization.