EECE 5614: Reinforcement Learning

Shalaka Padalkar

NU ID: 002831387

Project 4

1. Standard Deep Q Network

Q. How did you implement the training process? Include the code for this part.

A screenshot of a computer code

AI-generated content may be incorrect.

The environment and the DQN agent are defined in the class MazeEnv and DQNAgent. The train() function implements the training loop for the DQN agent over multiple episodes. In each episode, the agent interacts with the environment using an epsilon-greedy policy to balance exploration and exploitation. Transitions are stored in a replay buffer, and the agent updates its Q-network at regular intervals using minibatch samples. The epsilon value decays over episodes to encourage more exploitation over time. Episode rewards and losses are tracked and smoothed to visualize learning progress.

Q. Did you make any changes to the network structure? If so, specify what changes you made.

The network structure provided by TA was a 4-layer MLP using ReLU activations. I tried extending it to include LayerNorm after each hidden layer. It was observed that this layer improved the agent’s learning stability and convergence by normalizing intermediate activations.

Q. How did you implement ε?



Epsilon is implemented as suggested in the Project guidelines PDF where k is the episode number. In the initial episodes, lower k results in close to 1.0 value of epsilon and more exploration. As the episodes progress, the epsilon decays exponentially. The minimum value is capped at 0.1 to ensure that the agent does not become too greedy and biased.

Q. Report all the parameters and hyperparameters.

|  |  |
| --- | --- |
| Size of replay memory D | 5000 |
| Maximum length of episode Tepi | 50 |
| Number of steps to update Q-network NQU | 1 |
| The discount factor γ | 0.99 |
| Size of minibatch Nbatch | 128 |
| Learning rate α | 1e-4 |
| Soft update hyperparameter η | 5e-2 |
| Number of episodes Nepi | 2000 |

Q. Plot Average Reward Vs Average Loss during training with respect to episode. Explain the trend seen in these plots.

A graph of a graph

AI-generated content may be incorrect.

A graph of a loss

AI-generated content may be incorrect.

The average reward graph shows a steep rise initially, indicating rapid learning. It plateaus near the episode number 100, suggesting the agent consistently reaches the goal state and receives high rewards after sufficient training.

The average loss graph spikes early due to unstable Q-value estimates but gradually decreases and stabilizes, indicating that the model has learned a reliable Q-function and is no longer making large prediction errors.

Q. Show the final obtained policy on the maze. Explain whether the obtained policy in intuitively optimal or not.

A screenshot of a puzzle

AI-generated content may be incorrect.

The agent consistently selects actions that lead directly to the goal state, avoiding unnecessary loops. It effectively navigates around the walls and avoids the penalty zones like the red cell adjacent to the starting state.

Q. Show the final state values (i.e. maximum of the Q values at each state) on the maze. Is the range of numbers what you expected? Please explain.

A screenshot of a diagram

AI-generated content may be incorrect.

Yes, the range of number is as expected. It follows an order that the state values go from high to low as we move away from the goal state cell.

Q. Show a path on the maze starting from the start state and following the obtained policy. Hopefully it reaches to the goal state!

A screenshot of a computer game

AI-generated content may be incorrect.

Reaches the goal state.

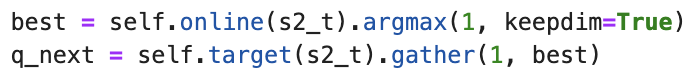
1. Double Deep Q-Network

Q. How did you change standard DQN code to implement the training process? Include the code for this part.

In standard DQN, the target Q-value is computed using the max Q-value from the target network.



Whereas in Double DQN, the action is chosen using the online network, but the value is estimated using the target network.



This decouples action selection from evaluation, reducing overestimation bias in Q-values

Q. Report all the parameters and hyperparameters.

|  |  |
| --- | --- |
| Size of replay memory D | 5000 |
| Maximum length of episode Tepi | 50 |
| Number of steps to update Q-network NQU | 1 |
| The discount factor γ | 0.99 |
| Size of minibatch Nbatch | 64 |
| Learning rate α | 1e-4 |
| Soft update hyperparameter η | 5e-2 |
| Number of episodes Nepi | 1200 |

Q. Plot Average Reward Vs Average Loss during training with respect to episode. Explain the trend seen in these plots.

A graph showing a line

AI-generated content may be incorrect.

A graph showing a loss

AI-generated content may be incorrect.

The average reward starts off low and negative, indicating that the agent initially performs poorly. As the episodes progress, the reward increases rapidly and then stabilizes near episode number 90–95, suggesting the agent has learned a near-optimal policy to reach the goal consistently.

Q. Show the final obtained policy on the maze. Explain whether the obtained policy in intuitively optimal or not.

A screenshot of a puzzle

AI-generated content may be incorrect.

The policy obtained is near-optimal. In a couple of states, usually near the walls, the agent is not able to learn the best action.

Q. Show the final state values (i.e. maximum of the Q values at each state) on the maze. Is the range of numbers what you expected? Please explain.

A screenshot of a graph

AI-generated content may be incorrect.

Yes, the state values range from high to low as we move farther from the goal state.

Q. Show a path on the maze starting from the start state and following the obtained policy. Hopefully it reaches to the goal state!

A screenshot of a computer game

AI-generated content may be incorrect.

It reaches the goal state.

Q. Compare results.

Double DQN provides more stable and reliable learning than Standard DQN by reducing overestimation of Q-values—especially important in environments with stochastic transitions, like our maze problem with p=0.025. Even with fewer episodes (1200 for double DQN and 2000 for standard DQN), it achieves a comparable policy.

1. Dueling Deep Q-Network

Q. How did you change code to implement the training process? Include the code for this part.

The Dueling DQN class is defined with two separate streams inside the network: A Value stream that estimates the value of a state and an Advantage stream that estimates the relative importance of each action. In the forward function of Dueling DQN, the final Q-values are computed using the formula provided in the PDF:

A black text on a white background

AI-generated content may be incorrect.

Q. Report all the parameters and hyperparameters.

|  |  |
| --- | --- |
| Size of replay memory D | 5000 |
| Maximum length of episode Tepi | 50 |
| Number of steps to update Q-network NQU | 1 |
| The discount factor γ | 0.99 |
| Size of minibatch Nbatch | 64 |
| Learning rate α | 1e-3 |
| Soft update hyperparameter η | 1e-2 |
| Number of episodes Nepi | 500 |

Q. Plot Average Reward Vs Average Loss during training with respect to episode. Explain the trend seen in these plots.

A graph with blue lines

AI-generated content may be incorrect.

A graph of a loss

AI-generated content may be incorrect.

The average reward starts off negative (indicating poor policies) but quickly improves after approximately 50 episodes. Around 100th episode, the agent starts to consistently achieve higher rewards, converging, showing that it has learned a near-optimal policy.

Q. Show the final obtained policy on the maze. Explain whether the obtained policy in intuitively optimal or not.

A diagram of different colored squares

AI-generated content may be incorrect.

The obtained policy navigates well around the walls and effectively avoids the penalty zones like the yellow and red cells, achieving maximum rewards by the time it reaches the goal state. So the obtained policy is a near-optimal policy.

Q. Show the final state values (i.e. maximum of the Q values at each state) on the maze. Is the range of numbers what you expected? Please explain.

A screenshot of a graph

AI-generated content may be incorrect.

Yes, the state values are as expected.

Q. Show a path on the maze starting from the start state and following the obtained policy. Hopefully it reaches to the goal state!

A screenshot of a computer game

AI-generated content may be incorrect.

Q. Compare results.

The Dueling DQN learns to estimate state values more effectively. It shows slower initial learning, but steadily rises after approximately 75-80 episodes.