Tabular Q-Learning vs Deep Q-Learning (DQN) on Snake Game

1. Implementation Summary

In both approaches, the Snake game was integrated with reinforcement learning agents that used the same state representation — consisting of handcrafted features such as the snake's current direction, relative food position, and immediate dangers. The action space was limited to four directions: up, down, left, and right.

The **tabular Q-learning** method used a dictionary (Q-table) to store Q-values for each discrete state-action pair. It followed the standard Q-learning update rule and used epsilon-greedy action selection with linear decay.

The **DQN agent**, on the other hand, replaced the Q-table with a neural network that approximated the Q-values. It employed experience replay to break temporal correlations and used a target network to stabilize learning. The neural network was trained using the MSE loss between predicted and target Q-values.

2. Performance Comparison

Tabular Q-learning started off learning quickly in the early episodes but struggled to improve beyond a certain point. By episode 500, it achieved an average score of around 2.0. The learning curve showed noisy fluctuations, and while there was steady progress at times, it often plateaued and regressed.

In contrast, DQN demonstrated a consistent and much steeper learning curve. It achieved an average score above 12 by episode 500, with more stability and higher peak performance. It clearly learned to survive longer, collect more food, and avoid collisions more effectively.

3. Observations

- Tabular Q-learning is limited by its reliance on exact state matches. Since the Snake game can generate a large number of unique game states, the Q-table cannot generalize across similar but unseen situations. This results in slower learning and poor scalability.

- DQN excels in this setting due to its ability to generalize across states using function approximation. The neural network is capable of learning patterns in the state space and leveraging them to make better decisions even in unseen scenarios.

- The use of replay memory and target networks in DQN provided additional stability, helping the agent learn more efficiently and avoid instability often seen in naive reinforcement learning setups.

4. Conclusion

DQN significantly outperforms tabular Q-learning in the Snake game environment. It learns faster, scores higher, and generalizes better to complex situations. While tabular methods can work for simple environments, deep reinforcement learning methods like DQN are clearly more effective for games with large and dynamic state spaces like Snake.