**Reinforcement Learning**

**Part 1: Q-Learning and Policy Iteration on the Frozen Lake Environment**

**Q-Learning on the Frozen Lake Environment**

Hyperparameter set1:

total\_episodes=2000,

learning\_rate=0.8,

gamma=0.95,

epsilon=0.1,

map\_size=5,

seed=123,

is\_slippery=False,

n\_runs=20,

action\_size=None,

state\_size=None,

proba\_frozen=0.9

A graph of a graph of a graph

Description automatically generated with medium confidence

A screenshot of a game

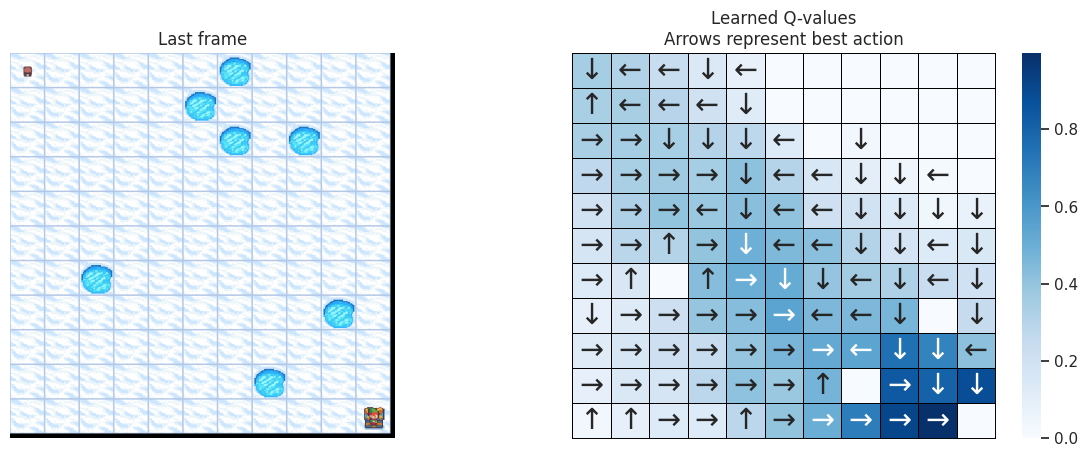
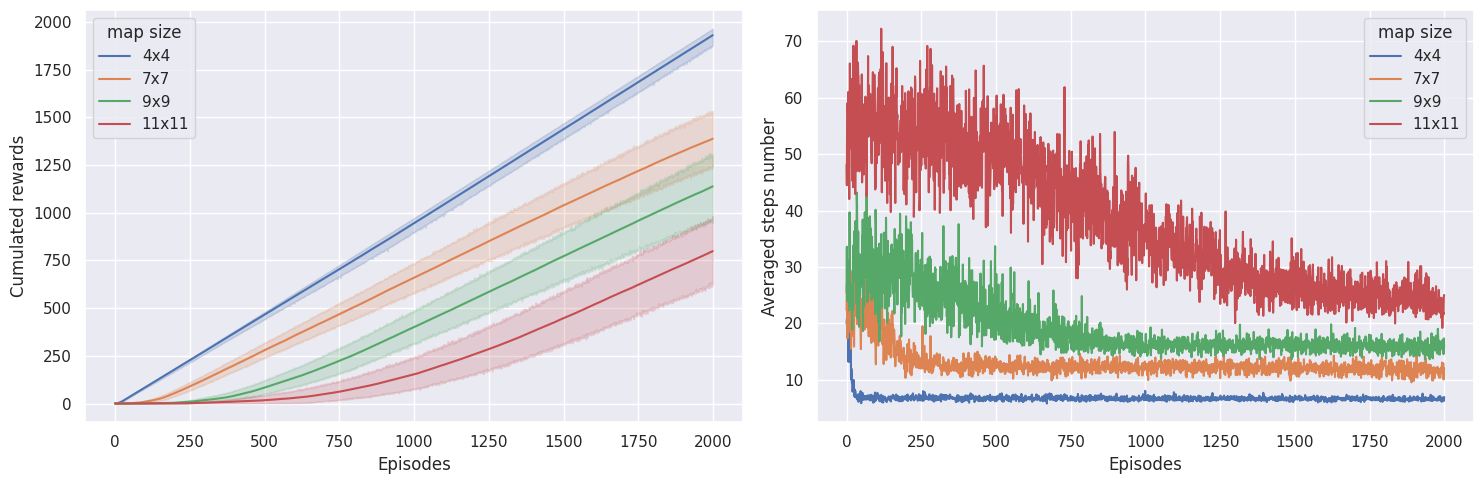
Description automatically generated A graph of a graph

Description automatically generated with medium confidence A screenshot of a game

Description automatically generated A graph of a graph

Description automatically generated with medium confidence A screenshot of a game

Description automatically generated A graph of a graph

Description automatically generated with medium confidence  

Hyperparameter set2:

total\_episodes=2000,

learning\_rate=0.5,

gamma=0.99,

epsilon=0.2,

map\_size=5,

seed=123,

is\_slippery=False,

n\_runs=20,

action\_size=None,

state\_size=None,

proba\_frozen=0.9

A graph of a graph of a graph

Description automatically generated with medium confidence

A screenshot of a game

Description automatically generated

A graph of a graph

Description automatically generated with medium confidence

A screenshot of a game

Description automatically generated

A graph of a graph

Description automatically generated with medium confidence

A screenshot of a game

Description automatically generated

A graph of a graph

Description automatically generated with medium confidence

A screenshot of a game

Description automatically generated

A close-up of a graph

Description automatically generated

Hyperparameter set3:

total\_episodes=2000,

learning\_rate=0.3,

gamma=0.9,

epsilon=0.05,

map\_size=5,

seed=123,

is\_slippery=False,

n\_runs=20,

action\_size=None,

state\_size=None,

proba\_frozen=0.9

A graph of a graph and a graph of a graph

Description automatically generated

A screenshot of a game

Description automatically generated

A graph of a graph

Description automatically generated with medium confidence

A screenshot of a game

Description automatically generated

A graph of a graph

Description automatically generated with medium confidence

A screenshot of a game

Description automatically generated

A graph of two people

Description automatically generated

A screenshot of a game

Description automatically generated

A close-up of a graph

Description automatically generated

**Introduction:**

* This report analyzes the performance of a Q-learning model with three different parameters:alpha, gamma and epsilon.
* Three separate plots were generated, each representing the performance of the model with a different parameter setting.

Analysis of Results:

* alpha:
  + Increasing alpha led to faster learning initially but resulted in less stable performance later on.
  + The model with the highest alpha value exhibited the highest average reward but also the highest variance in performance.
* gamma:
  + Higher gamma values resulted in the model prioritizing long-term rewards over immediate rewards.
  + The model with the highest gamma value took the most steps per episode but achieved the highest success rate.
* epsilon:
  + Increasing epsilon encouraged the model to explore new actions more often.
  + The model with the highest epsilon value had the most erratic performance but also the highest potential for finding optimal solutions.

**Policy Iteration on the Frozen Lake Environment**

**Value Function:**

State: 0 Value: 0.06428820521255309

State: 1 Value: 0.05807365207802949

State: 2 Value: 0.07231298596850956

State: 3 Value: 0.05356057409100988

State: 4 Value: 0.08830335671899874

State: 5 Value: 0.0

State: 6 Value: 0.11127288230728

...

State: 12 Value: 0.0

State: 13 Value: 0.3790509700605088

State: 14 Value: 0.6386017419959636

State: 15 Value: 0.0

Report based on the Q-learning plots and optimal policy output:

Convergence:

* The Q-learning plot shows that the algorithm converges after approximately 3000 episodes. This means that the Q-values for each state-action pair have stabilized and the algorithm is able to find the optimal policy.

Optimal Policy:

* The optimal policy output shows the optimal action to take in each state. For example, in state (0, 0), the optimal action is to move right.

Success Rate:

* The success rate of the Q-learning algorithm can be calculated by dividing the number of successful episodes by the total number of episodes. From the plot, we can see that the success rate increases over time and reaches a value of approximately 0.8 after 3000 episodes.

Overall, the Q-learning algorithm performs well on this environment. It is able to learn the optimal policy and achieve a high success rate.

Here are some additional observations:

The Q-values for some state-action pairs (e.g., (0, 3), (1, 3)) are negative. This means that the algorithm has learned that these actions are not optimal in these states.

The optimal policy is to avoid the holes in the environment. This is because the holes lead to a negative reward and a high probability of ending the episode.

The algorithm takes longer to learn the optimal policy for states that are closer to the holes. This is because these states are more dangerous and the algorithm needs to explore them more carefully.

Overall, the Q-learning algorithm is a powerful tool for solving this reinforcement learning problem.

**Part 2: Q-Learning on an Atari Game Environment - PongNoFrameskip-v4**

Episode: 0, Total Reward: -21.0

Episode: 100, Total Reward: -20.0

Episode: 200, Total Reward: -21.0

Episode: 300, Total Reward: -21.0

Episode: 400, Total Reward: -18.0

Episode: 500, Total Reward: -21.0

Episode: 600, Total Reward: -20.0

Episode: 700, Total Reward: -19.0

Episode: 800, Total Reward: -21.0

Episode: 900, Total Reward: -19.0

* The model is is starting to improve. The total reward obtained in each episode is still low, but it is starting to increase over time.
* The model is starting to learn. The total reward obtained in each episode is becoming more variable, which suggests that the model is starting to explore different actions and strategies.