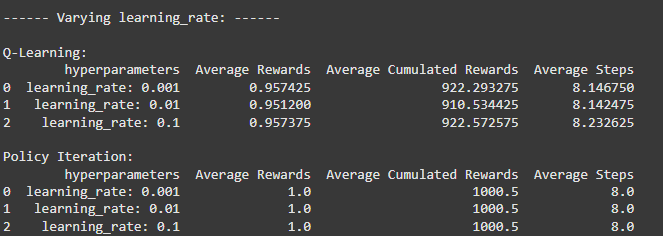
Part 1: **Q-Learning and Policy Iteration on the Frozen Lake Environment**

Outputs from Part1.ipynb

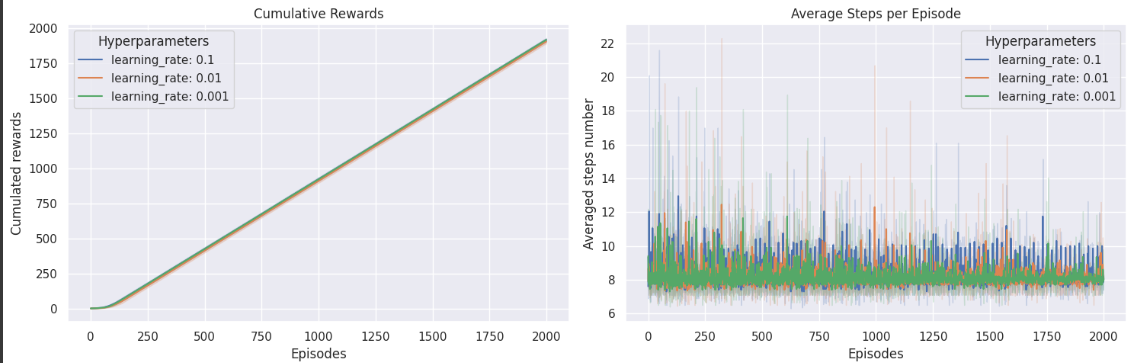
Time taken ~ **30 minutes**

1. **Varying learning rate (α)**

****

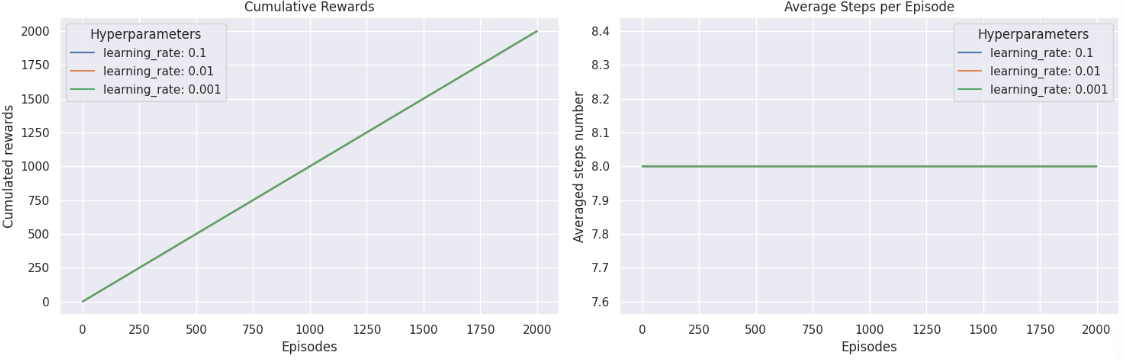
**Impact of learning rate (α) in learning process**

1. **Q – Learning** (varying **α** butt fixed γ = 0.8, ε= 0.01)



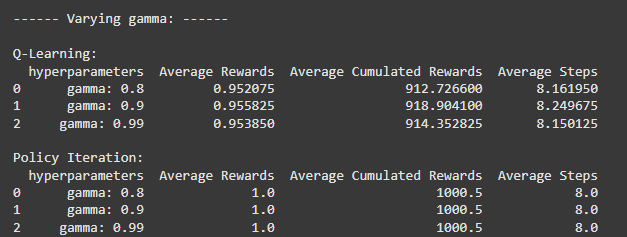
* All three learning rates achieve similar cumulative rewards
* Higher learning rate shows slightly faster early learning, allows faster convergence early on but can be noisy.

1. **Policy iteration** (varying **α** butt fixed γ = 0.8, ε= 0.01)

****

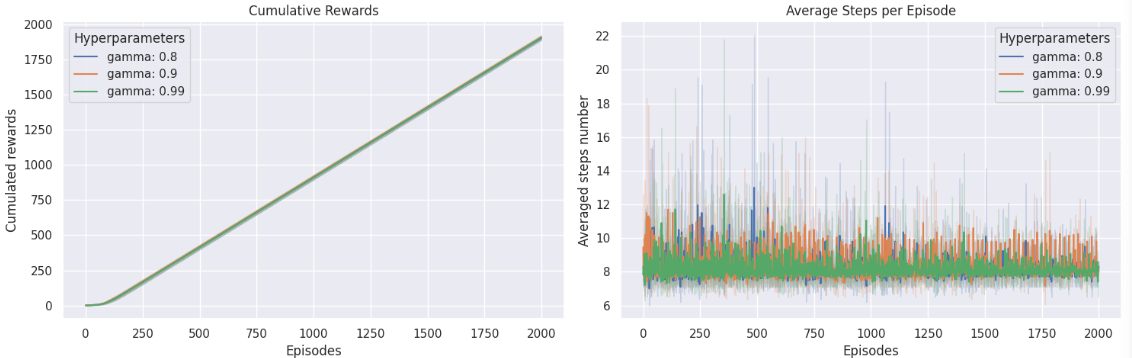
* Policy iteration is not affected by learning rate.

1. **Varying discount factor (γ)**

****

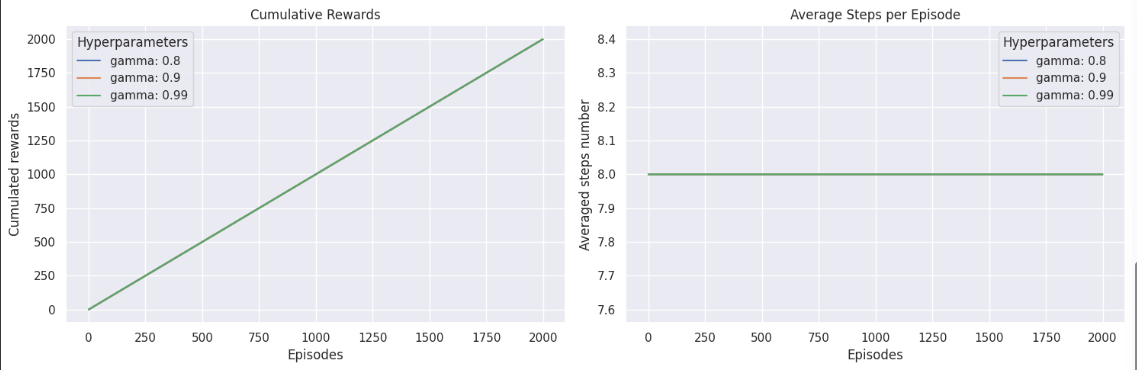
**Impact of discount rate (**γ**) in learning process**

1. **Q – Learning** (varying **γ** but fixed α = 0.1 ε= 0.01)



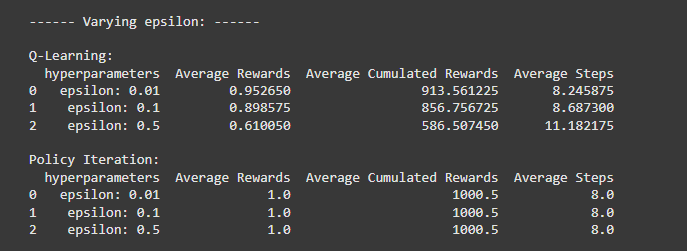
* Small difference across γ = [0.8, 0.9, 0.99]
* γ = 0.9 yield a higher cumulative reward
* Higher γ slightly improves cumulative reward over time (as it favors long-term rewards).
* Noisy step counts early on; variance narrows with episodes.

1. **Policy iteration** (varying **γ** but fixed α = 0.1 ε= 0.01)

****

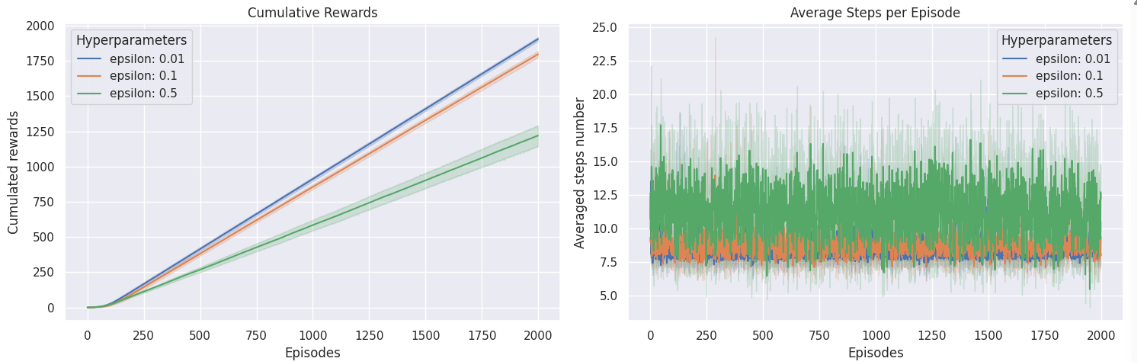
* γ does not significantly impact the final policy
* Policy iteration computes the optimal policy w/0 γ affecting performance

1. Varying exploration rate (ε)



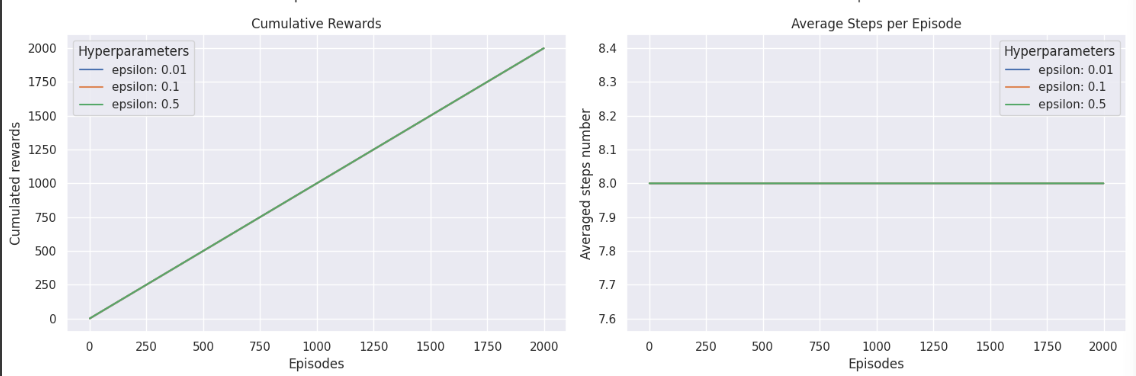
**Impact of exploration rate (ε) in learning process**

1. **Q – Learning** (varying **ε** but fixed α = 0.1, γ = 0.8)



* **ε** = 0.01 converges fast with high reward **, ε** = 0.1 learns slower, and  **ε** = 0.5 learns the slowest and shows **highest variance** in step count. So, some degree of exploration is beneficial, but higher exploration rate leads to slow learning.
* Low exploration rate may miss better policies but lead to fast learning whereas high exploration rate leads to better exploration but delays convergence.

1. **Policy iteration** (varying **ε** but fixed α = 0.1, γ = 0.8)

****

* Achieves an average reward of 1.0
* Learning in policy iteration is not affected by exploration rate

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

I could not get the Atari game environment to render on human mode and had to do rgb\_array.

The runs folder is attached. The logs are present in the pong\_light.to visualize using Tensorflow.

Model did not train for long as Colab crashed repeatedly.

The game play is saved as a gif. I was only able to score 1 point.

A screenshot of a graph

AI-generated content may be incorrect.

A graph on a black background

AI-generated content may be incorrect.