# Reinforcement Learning

Artificial Intelligence

**Bahja Ali**

2 october 2025

# Introduction

The objective of this project is to explore reinforcement learning (RL) through two methods : **Value Iteration** : a model-based algorithm relying on a complete model of the environment, **Q-Learning** : a model-free algorithm that learns directly from interactions.

The environment is a maze where the agent must reach an exit while maximizing its rewards. This project allows us to compare the performance of the two approaches, analyze the effect of exploration, as well as the sensitivity to key parameters (**discount factor** $\gamma$, **learning rate** $\alpha$, and r**andomness probability** $p$ **of the environment**).

# 1 Methodology

## 1.1 Environment

The maze is represented as a $7 \times 7$ grid, with :

- Blocked states (walls) where the agent cannot move.
- Two terminal states (the exits) with a positive reward (10 for the upper exit, 100 for the lower-left exit).
- A negative movement cost of $-1$ to encourage the agent to find a short path.

## 1.2 Implemented Algorithms

**Value Iteration** : Iterates over state values until convergence, using the Bellman equation.

**Q-Learning** : Updates a $Q(s, a)$ table from observed transitions, with two exploration strategies :

- $\varepsilon$-greedy : random choice with probability $\varepsilon$.
- Softmax (Max-Boltzmann) : probabilistic choice based on a temperature parameter $\tau$.

## 1.3 Training Parameters

- Number of episodes : 1000 (20 repetitions on average).
- $\gamma = 0.9$ (except sensitivity analysis).
- $\alpha = 0.1$.
- Exploration policies : fixed $\varepsilon$, decaying $\varepsilon$, fixed $\tau$, decaying $\tau$.

## 2 Résultats et discussion

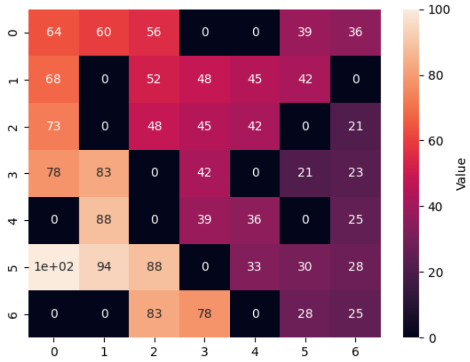


FIGURE 1 – Heatmap of state values obtained by Value Iteration

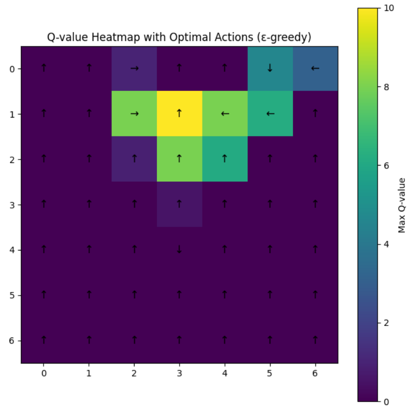


FIGURE 2 – $\varepsilon$-greedy

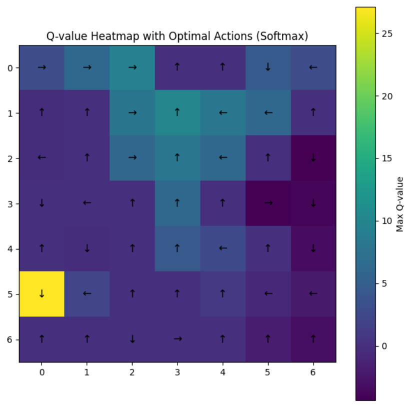
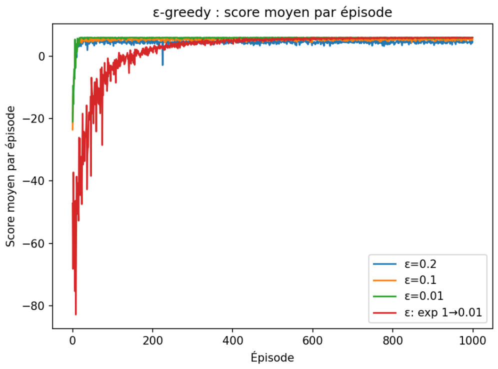


FIGURE 3 – Softmax

When there is a probability of non-determinism $p > 0$, the same action can lead to different states. *Value Iteration* explicitly encodes this randomness in the **Bellman equation**. *Q-learning* encodes it implicitly through sampling : the $Q(s, a)$ values become expectations after many updates.

We observe that as $p$ increases, the values decrease (the cost $-1$ accumulates and terminal rewards are reached less often).



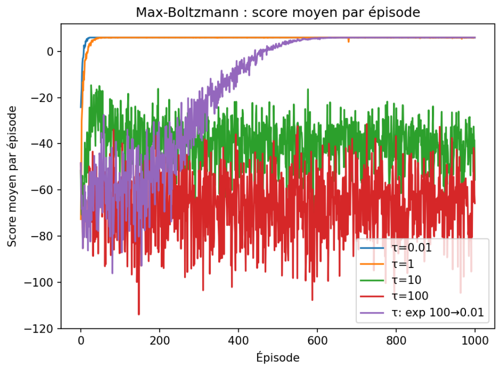


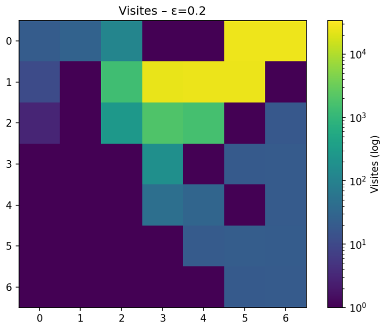
FIGURE 4 – Average score per episode for different values of $\varepsilon$

FIGURE 5 – Average score per episode for different values of $\tau$

When $p$ increases, the average returns decrease and convergence slows down.

*Value Iteration* remains more robust because it directly optimizes the expected transitions.

Q-learning is noisier : it requires more episodes to smooth out randomness, and its performance degrades more quickly in highly stochastic environments.



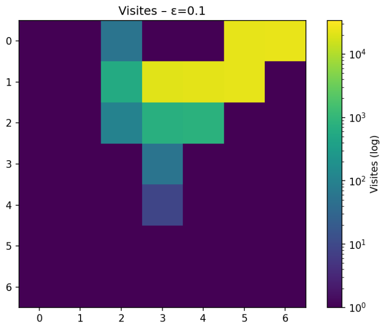
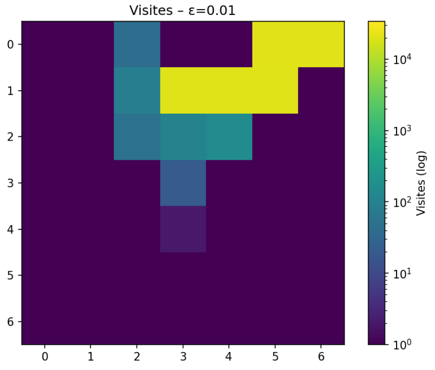


FIGURE 6 – Heatmap of visits for $\varepsilon = 0.2$

FIGURE 7 – Heatmap of visits for $\varepsilon = 0.1$



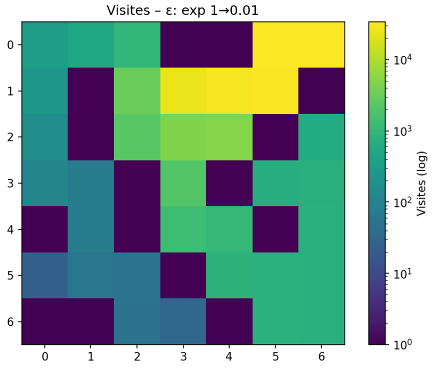


FIGURE 8 – Heatmap of visits for $\varepsilon = 0.01$

FIGURE 9 – Heatmap of visits for $\varepsilon$ decreasing from 1 to 0.01

The $\varepsilon$-greedy strategy explores uniformly : a too-large $\varepsilon$ slows down learning, while a decaying $\varepsilon$ gives a good compromise.

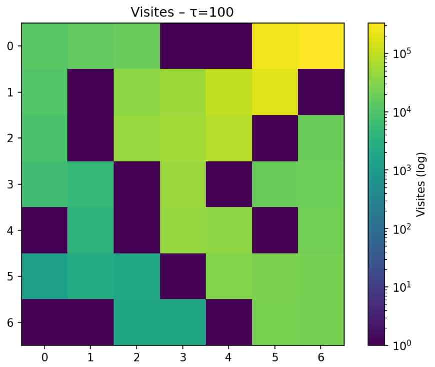


FIGURE 10 – Heatmap of visits for $\tau = 100$

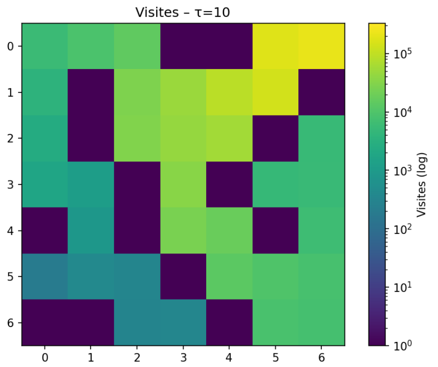


FIGURE 11 – Heatmap of visits for $\tau = 10$

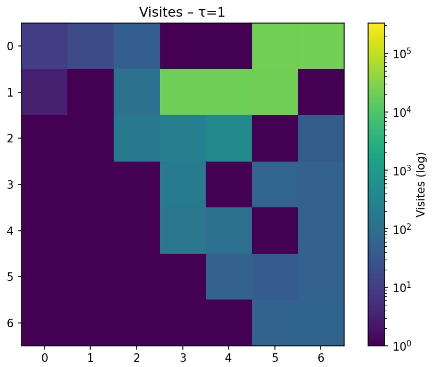


FIGURE 12 – Heatmap of visits for $\tau = 1$

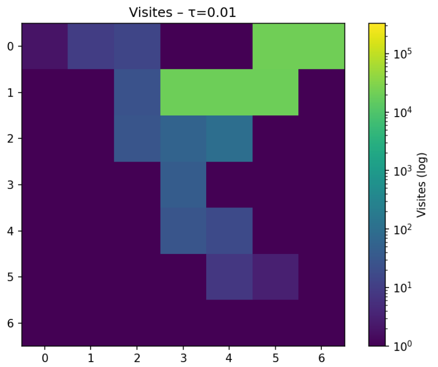


FIGURE 13 – Heatmap of visits for $\tau = 0.01$

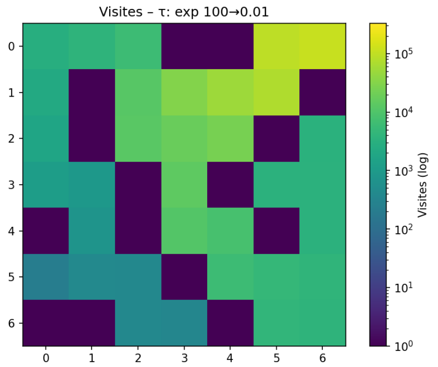


FIGURE 14 – Heatmap of visits for $\tau$ decreasing from 100 à 0.01

The **Softmax (Max-Boltzmann)** strategy explores proportionally to the $Q$-values, which allows for a better balance between exploration and exploitation if $\tau$ is well chosen. A $\tau$ that is too large ($\approx 100$) leads to almost random exploration and low scores, while a $\tau$ that is too small reduces the policy to greedy behavior.

When the agent converges, it follows the shortest path because negative rewards accumulate along longer trajectories, while the **propagation of values** or the update of **$Q$-values** naturally favors actions leading most quickly to the exit.

## 3 Conclusion

In this project, we implemented and compared two reinforcement learning approaches : **Value Iteration**, a deterministic model-based method, and **Q-Learning**, a model-free method. Our results show that **Value Iteration** quickly converges to an optimal policy when the environment is deterministic, while **Q-Learning** remains more flexible and succeeds in adapting even in the presence of randomness in transitions.

The study of exploration strategies (*$\varepsilon$-greedy* and *Max-Boltzmann*) highlighted their essential role in balancing exploration and exploitation : a decaying $\varepsilon$ rate and a well-chosen $\tau$ allow sufficient exploration at the beginning of training, followed by efficient exploitation. We also showed that increasing the probability of non-determinism $p$ slows down learning and degrades performance, but the learnt values integrate this randomness by adjusting reward expectations.

In conclusion, this work illustrates the importance of value update mechanisms and exploration strategies to achieve optimal behavior, even in uncertain environments.