

# Reinforcement Learning Assignment - 1

## Temporal Difference Learning in Tic-Tac-Toe

Comparative Analysis of Greedy and  $\epsilon$ -Greedy Strategies

**Submitted by:**

Deep Das, Himal Rana, Vikram Singh

Roll No: *U23AI052, U23AI053, U23AI034*

**Course:** Reinforcement Learning

**Institution:** *Sardar Vallabhbhai National Institute of Technology, Surat*

**Date:** January 26, 2026

# 1 Aim

To implement Temporal Difference (TD) learning for the Tic-Tac-Toe game and analyze the performance of Greedy and  $\epsilon$ -Greedy strategies through self-play and learning curves.

## 2 Objectives

- To design a Tic-Tac-Toe environment for reinforcement learning.
- To implement TD learning agents for both players.
- To compare Greedy and  $\epsilon$ -Greedy exploration strategies.
- To visualize learning progress using win-rate curves.

## 3 Theory

Reinforcement Learning (RL) is a learning paradigm where an agent learns optimal behavior by interacting with an environment and receiving rewards. Temporal Difference (TD) learning updates value estimates based on the difference between successive predictions.

In Tic-Tac-Toe, the game is modeled as a Markov Decision Process (MDP) where:

- States represent board configurations.
- Actions correspond to valid moves.
- Rewards are assigned based on win, loss, or draw outcomes.

The  $\epsilon$ -Greedy strategy balances exploration and exploitation by choosing a random action with probability  $\epsilon$  and the best-known action otherwise.

## 4 Environment Design

The Tic-Tac-Toe environment is implemented using a  $3 \times 3$  board:

- Player X is represented by +1.
- Player O is represented by -1.
- Empty cells are represented by 0.

The environment supports:

- State reset

- Valid move generation
- Winner detection
- Game termination checking

## 5 TD Learning Agent

Each agent maintains a value function  $V(s)$  stored as a dictionary. The update rule used is:

$$V(s) \leftarrow V(s) + \alpha[V(s') - V(s)]$$

where  $\alpha$  is the learning rate.

### 5.1 Reward Scheme

- Win: 1.0
- Loss: 0.0
- Draw: 0.5

## 6 Training Methodology

Agents are trained through self-play for multiple episodes. Four configurations are evaluated:

1. Greedy vs Greedy
2. Greedy vs  $\epsilon$ -Greedy
3.  $\epsilon$ -Greedy vs Greedy
4.  $\epsilon$ -Greedy vs  $\epsilon$ -Greedy

Each configuration is trained for 10,000 episodes, and cumulative win statistics are recorded.

## 7 Implementation

The system is implemented in Python using NumPy and Matplotlib.

## 7.1 Sample Code Snippet

```
1 class TDAgent:
2     def __init__(self, player, alpha=0.1, epsilon=0.1):
3         self.player = player
4         self.V = defaultdict(float)
5         self.alpha = alpha
6         self.epsilon = epsilon
```

## 8 Results

Learning curves were plotted to visualize agent performance over time. The  $\epsilon$ -Greedy vs  $\epsilon$ -Greedy configuration showed the most stable learning behavior and higher draw rates, indicating optimal play.

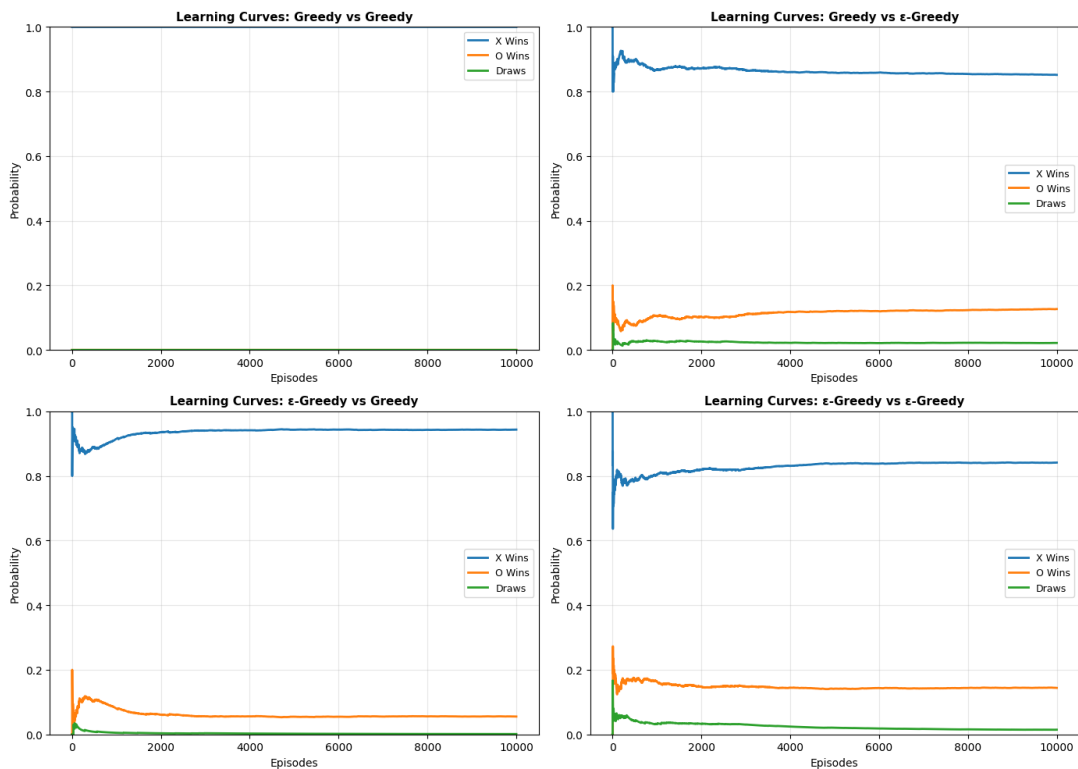


Figure 1: Learning Curves for Different Strategy Combinations

## 9 Game Demonstration

After training, agents were set to greedy mode and allowed to play a game. The resulting gameplay demonstrated rational move selection and frequent draw outcomes, consistent with optimal Tic-Tac-Toe strategies.

## 10 Conclusion

Temporal Difference learning successfully enables agents to learn optimal Tic-Tac-Toe strategies through self-play.  $\epsilon$ -Greedy exploration improves learning stability and prevents premature convergence, leading to better overall performance.

## 11 References

1. Sutton, R. S., & Barto, A. G., *Reinforcement Learning: An Introduction*.