SIT320 Task 10 – Report

1. Introduction

This task applies Dynamic Programming (DP) and Reinforcement Learning (RL) to Tic-Tac-Toe. The aim was to model the game as a Markov Decision Process (MDP), solve it with Value Iteration, train a Q-Learning agent, and compare results.

2. Task 1 – DP with Value Iteration

- States: All valid boards with X to play.
- Actions: Empty cells.
- Transitions: After X moves, O plays randomly.
- Rewards: +1 (X win), −1 (O win), 0 (draw/ongoing).
- Discount: $\gamma=1$.

Method: Value Iteration updated values until convergence, deriving an optimal policy for X.
Result: Policy recommends strong openings (corner/center). Simulated games confirmed X avoids losses vs random O.

3. Task 2 – Q-Learning

- Setup: X learns via tabular Q-Learning; O is random.
- Hyper-parameters: $\alpha=0.5$, $\gamma=1$, ε decays 1.0→0.05 over 20k episodes.
- Update Rule:
  $Q(s,a) \leftarrow Q(s,a) + \alpha(r + \gamma \max_{a'} Q(s',a') - Q(s,a))$

Result: Early win rates were low (due to exploration). After training, X achieved ~90–95% win rate vs random O.

4. DP vs RL Comparison

- DP: Exact, optimal strategy, needs full model.
- RL: Learns from interaction, initially weak, converges close to DP performance.
Observation: RL matches DP after enough training, showing learning without environment knowledge.

5. Reflection

- ε-decay: Helped exploration early, then stable play later.
- $\alpha=0.5$: Balanced speed and stability. Smaller α = slower learning; larger α = unstable.
- $\gamma=1$: Correct for episodic games.
Overall, RL required tuning but achieved near-DP performance.

6. Conclusion

DP with Value Iteration provided the optimal policy, while Q-Learning learned to approximate it through self-play. Both methods produced strong strategies, highlighting the difference between planning with full knowledge and learning from experience.