Updated Plan of Action

Summer of Science 2025

Aadeshveer Singh
24B0926
24b0926@iitb.ac.in

June 16, 2024

This updated Plan of Action reflects accomplishments from the first phase of the Summer of Science program and outlines the intended learning trajectory for the remaining duration.

Phase 1 Accomplishments (Weeks 1-4 of RL SoS)

Week 1: Foundational Logic, Automata, and Temporal Logic

Focus: Establishing theoretical groundwork for MDPs.

Key Accomplishments:

- **Propositional Logic (Huth & Ryan Ch. 1.1-1.6):**
  - Mastered syntax, semantics, connectives, truth tables, normal forms.
  - Understood satisfiability, tautologies, contradictions.
  - Explored logical consequence (entailment) and formal proof systems (Natural Deduction), including conceptual understanding of soundness and completeness.
  - Completed H&R Exercises 1.1, 1.2, 1.3, 1.4.
- **Predicate Logic (First-Order Logic - Huth & Ryan Ch. 2.1-2.3):**
  - Studied syntax (terms, predicates, quantifiers $\forall$, $\exists$).
  - Understood semantics (interpretations, domains, variable assignments).
  - Explored validity, satisfiability, and entailment in FOL.
- **Finite Automata (Baier & Katoen Ch. 4.1; Huth & Ryan):**
  - Understood formal definitions and operation of DFAs and NFAs.
  - Grasped language acceptance, equivalence of DFAs/NFAs (subset construction).
  - Studied Regular Expressions and their equivalence to FAs (Kleene's Theorem conceptually).
- **Linear Temporal Logic (LTL) Basics (Huth & Ryan Ch. 3.1-3.2):**
  - Learned syntax and intuitive meaning of key temporal operators (F, G, X, U).
  - Practiced expressing simple system properties using LTL.
- **Practice:** Successfully completed Practice Sheet 1 covering these foundational topics.

Week 2: Stochastic Processes - Markov Chains

Focus: Understanding systems with probabilistic transitions.

Key Accomplishments:

- **Markov Chain Fundamentals (Baier & Katoen Ch. 10 targeted lookups; Sutton & Barto Ch. 3 context; Lectures):**
  - Understood formal definition of Discrete-Time Markov Chains (DTMCs), state space, Transition Probability Matrix (TPM), and the Markov Property.
  - Studied N-step transition probabilities and Chapman-Kolmogorov equations.
  - Explored classification of states (accessibility, communication, recurrence, transience, periodicity, absorbing states).
  - Grasped the concept of stationary distributions and conditions for their existence.
- **Practice:** Working through Practice Sheet 2, applying MC concepts to various problems (e.g., Knight's tour, Gambler's ruin elements, Mazes).

Week 3: MDPs, k-Armed Bandits, and Dynamic Programming Introduction

Focus: Formalizing decision-making under uncertainty and initial RL algorithms.

Key Accomplishments:

- **Markov Decision Processes (Sutton & Barto Ch. 3):**
  - Mastered formal definition (S, A, P, R, γ), policies (π), state-value functions (v_π), and action-value functions (q_π).
  - Derived and understood Bellman Expectation Equations for v_π and q_π.
  - Understood optimal value functions (v_*, q_*) and Bellman Optimality Equations.
- **Multi-Armed Bandits (Sutton & Barto Ch. 2):**
  - Implemented and experimentally compared various bandit algorithms:
    - ϵ-greedy (stationary and non-stationary settings).
    - Optimistic Initial Values.
    - Upper Confidence Bound (UCB).
    - Gradient Bandit algorithms (with and without baseline).
  - Gained practical insights into the exploration-exploitation trade-off.
- **Dynamic Programming Introduction (Sutton & Barto Ch. 4, up to 4.4):**
  - Studied Policy Evaluation, Policy Improvement, Policy Iteration (PI), and Value Iteration (VI) algorithms.
- **Implementations:**
  - Developed a custom GridWorld environment using Pygame.
  - Implemented core components of Policy Iteration (Policy Evaluation, Policy Improvement) for the GridWorld.
  - Began implementation and debugging of Policy Iteration for Jack's Car Rental problem, including advanced NumPy vectorization for expectation calculations.
  - Implemented Value Iteration for the Gambler's Problem, reproducing classic results.

Week 4: Dynamic Programming Deep Dive, Implementations, and Midterm Reporting

Focus: Consolidating DP understanding, completing implementations, and report preparation.

Key Accomplishments:

- **Dynamic Programming Mastery (Sutton & Barto Ch. 4 complete):**
  - Solidified understanding of Policy Iteration and Value Iteration, including their convergence properties and differences.
  - Studied asynchronous DP and generalized policy iteration concepts.
- **Completed Implementations for DP Case Studies:**
  - Finalized and tested Policy Iteration for the custom GridWorld.
  - Successfully implemented and converged Policy Iteration for Jack's Car Rental, demonstrating results.

  - Verified Value Iteration implementation for the Gambler's Problem across different parameters.

- **Midterm Report:** Compiled theoretical learnings and implementation results into the midterm report.

- **Problem Sheet 2 (Markov Chains):** Aiming for full completion.

Phase 2 Planned Work (Weeks 5-8 of RL SoS)

Week 5: Formulating RL Problems & Advanced MDP Concepts

Focus: Bridging theory to practical RL problem setup and exploring richer MDP models.

Topics:

- **Reward Engineering and Shaping:**
  - Principles of effective reward design; sparse vs. dense rewards.
  - Potential-based reward shaping (Ng, Harada, Russell, 1999) - theory, benefits (policy invariance), and pitfalls.
- **RL Problem Formulation Details:**
  - Episodic vs. Continuing tasks; Horizon considerations.
  - Role and impact of the Discounting factor (γ) in depth.
- **Practical Application with Gym MDPs:**
  - Explore and analyze standard OpenAI Gymnasium environments (e.g., CartPole, MountainCar, FrozenLake).
  - Implement and test basic interaction loops with these environments.
- **Advanced MDP Models (Introductions and Core Concepts):**
  - Hidden Markov Models (HMMs): Definition, key problems (filtering, smoothing, decoding), contrast with MDPs.
  - Partially Observable MDPs (POMDPs): Formal definition, belief states, challenges, overview of solution approaches.
  - Inverse Reinforcement Learning (IRL): Concept of learning rewards from expert demonstrations; overview of key ideas (e.g., MaxEnt IRL).

Week 6: Model-Free Reinforcement Learning - Prediction and Control

Focus: Learning optimal behavior without a full model of the environment.

Topics:

- **Monte Carlo (MC) Methods (Sutton & Barto Ch. 5):**
  - MC Prediction (First-visit, Every-visit) for estimating v_π and q_π.
  - MC Control (On-policy: Exploring Starts, ϵ-greedy; Off-policy: Importance Sampling - ordinary and weighted).
  - Implementation of MC control for a simple Gym environment (e.g., Blackjack or a GridWorld without known transitions).
- **Temporal Difference (TD) Learning (Sutton & Barto Ch. 6):**
  - TD(0) Prediction: Algorithm and advantages over MC.
  - SARSA (On-policy TD Control): Algorithm, update rule, convergence properties.
  - Q-Learning (Off-policy TD Control): Algorithm, update rule, convergence proof sketch, distinction from SARSA.

- Expected SARSA.
- Implementation of Q-learning and SARSA for Gym environments (e.g., FrozenLake, CliffWalking).

- **Exploration vs. Exploitation Revisited:** In-depth analysis of ϵ-greedy, optimistic initialization, UCB (if not fully covered in bandits), and softmax exploration in the context of MC/TD control.

- **N-step Bootstrapping (Sutton & Barto Ch. 7 - if time permits):**
  - N-step TD prediction, N-step SARSA. Unifying MC and TD.

Week 7: Function Approximation and Deep Reinforcement Learning

Focus: Scaling RL algorithms to large state/action spaces using approximation, and advanced policy optimization.

Topics:

- **Value Function Approximation (Sutton & Barto Ch. 9-11):**
  - Need for approximation (curse of dimensionality).
  - Linear function approximation: features, gradient descent methods (Gradient MC, Semi-gradient TD(0), Semi-gradient SARSA). Understanding the deadly triad.
- **Deep Q-Networks (DQN):**
  - Using Neural Networks as function approximators for Q-values.
  - Key techniques: Experience Replay, Target Networks.
  - Introduction to DQN variants (e.g., Double DQN, Dueling DQN - conceptual overview).
  - **Main Project Implementation:** Continue/Intensify implementing DQN for Flappy Bird.
- **Policy Gradient Methods (Sutton & Barto Ch. 13):**
  - Policy approximation $\pi(a|s, \theta)$.
  - Policy Gradient Theorem (understanding its derivation and implications).
  - REINFORCE algorithm (Monte Carlo Policy Gradient), with and without baseline.
  - Conceptual overview of Actor-Critic methods (e.g., A2C/A3C).
- **Advanced Policy Optimization - Proximal Policy Optimization (PPO):**
  - Understanding the motivation for PPO (stability and sample efficiency improvements over simpler policy gradients).
  - Core concepts: Clipped surrogate objective, trust region methods (conceptual link).
  - Overview of PPO algorithm structure.
  - (Stretch Goal/If time allows after DQN focus) Initial exploration of PPO implementation or application.
- **Model-Based RL (Overview - Sutton & Barto Ch. 8):**
  - Learning a model of the environment.
  - Dyna-Q: Integrating planning, acting, and learning.

- Comparison: Model-based vs. Model-free RL.

- **RL Applications Deep Dive (Conceptual):**
  - AlphaGo/AlphaZero: MCTS, neural network architecture, self-play.
  - Robotics applications: Challenges and successes.

Week 8: Project Completion, Advanced Topics, and Final Reporting

Focus: Finalizing Flappy Bird project, exploring advanced topics, and report preparation.

Topics & Activities:

- **Flappy Bird Project with DQN:**
  - Intensive work: implementation, debugging, hyperparameter tuning, experimentation. * Visualization of agent learning and performance.
- **Eligibility Traces (Sutton & Barto Ch. 12 - if time permits):**
  - TD(λ), SARSA(λ), Watkins's Q(λ). Unifying MC and TD learning across different time scales.
- Consolidation and Review: Review all major topics covered.
- Final Report Submission: Comprehensive document detailing theoretical understanding, project design, implementation, experimental results, challenges, and learnings throughout the SoS.
- Preparation for final presentation/viva if applicable.

References and Learning Resources (To be Maintained/Updated)

This Plan of Action will primarily draw upon the following resources, supplemented by additional papers and online materials as needed.

Primary Textbooks:

1. Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction (2nd ed.)*. MIT Press.
2. Huth, M., & Ryan, M. (2004). *Logic in Computer Science: Modelling and Reasoning about Systems (2nd ed.)*. Cambridge University Press.
3. Baier, C., & Katoen, J.-P. (2008). *Principles of Model Checking*. MIT Press. (For targeted reference on formalisms).

Key Online Lecture Series & Slides:

1. David Silver (DeepMind/UCL) - RL Lecture Series: https://youtube.com/playlist?list=PLqYmG7hTraZDVH599EItlEWsUOsJbAodm
2. Balaraman Ravindran (NPTEL IIT Madras) - RL Lecture Series: https://youtube.com/playlist?list=PLwRJQ4m4UJjNymuBM9RdmB3Z9N5-0IlY0
3. Pieter Abbeel (UC Berkeley) - Deep Reinforcement Learning / CS188 AI lectures.
4. Dave Parker (University of Birmingham) - Probabilistic Model Checking Lectures (including MDPs): https://www.prismmodelchecker.org/lectures/pmc/

Survey Papers / Additional Materials:

1. Various Authors (2019). State-of-the-Art Reinforcement Learning Algorithms. *International Journal of Engineering Research & Technology (IJERT)*, *8*(12). Available: https://www.ijert.org/research/state-of-the-art-reinforcement-learning-algorithms-IJERTV8IS12033 pdf

Software & Libraries:

1. Python 3.x
2. Gymnasium (OpenAI Gym fork)
3. NumPy, Matplotlib, Seaborn
4. PyTorch or TensorFlow/Keras
5. Pygame (for custom environments/Flappy Bird)

Additional Support:

1. Practice problem sheets provided by SoS organizers.
2. Discussions with mentor and peers.