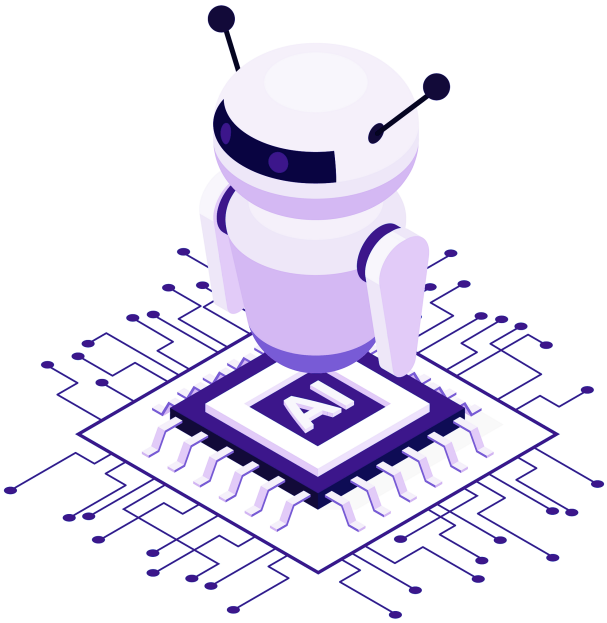

Deep Reinforcement Learning

Professor Mohammad Hossein Rohban

Homework 6:

Multi-Armed Bandits

Designed By:

Mohammad Mohammadi

mohammadm97i@gmail.com

Arshia Gharooni

arshiyagharoony@gmail.com

RIML

Deep RL Course

Spring 2025

Preface

Deep Reinforcement Learning (DeepRL) combines deep learning with reinforcement learning, enabling agents to make decisions based on high-dimensional inputs. A fundamental problem in reinforcement learning is the multi-armed bandit (MAB) problem, where an agent must choose among several options (arms) with uncertain rewards. The challenge lies in balancing the exploration of less-known arms with the exploitation of arms that have yielded high rewards.

This assignment focuses on implementing and comparing several bandit agents. It covers both classic MAB algorithms (such as Random, Explore-First, UCB, and Epsilon-Greedy agents) as well as contextual bandit methods using LinUCB. In addition to coding tasks marked with `TODO`, the notebook poses several conceptual questions. Your answers should discuss the theoretical and practical trade-offs inherent in these methods without revealing any solution hints.

Grading

The grading will be based on the following criteria, with a total of 105.25 points:

Task	**Points**
Task 1: Oracle Agent	3.5
Task 2: Random Agent	2
Task 3: Explore-First Agent	5.75
Task 4: UCB Agent	7
Task 5: Epsilon-Greedy Agent	2.25
Task 6: LinUCB Agent	27.5
Task 7: Final Comparison and Analysis	6
Task 8: Final Deep-Dive Questions	41.25
Clarity and Quality of Code	5
Clarity and Quality of Report	5
Bonus 1: Writing your report in Latex	10

Submission

The deadline for this homework is 1404/01/17 (April 6th 2025) at 11:59 PM.

Please submit your work by following the instructions below:

- Zip all the files together with the following naming format:

  `DRL_HW6_[StudentNumber]_[FullName].zip`

  - Replace `[FullName]` and `[StudentNumber]` with your full name and student number, respectively. Your `[FullName]` must be in CamelCase with no spaces.
- Submit the zip file through Quera in the appropriate section.
- We provided this LaTeX template for writing your homework solution. There is a 10-point bonus for writing your solution in LaTeX using this template and including your LaTeX source code in your submission, named `HW6_Solution.zip`.
- If you have any questions about this homework, please ask them in the Homework section of our Telegram Group.
- If you are using any references to write your answers, consulting anyone, or using AI, please mention them in the appropriate section. In general, you must adhere to all the rules mentioned here and here by registering for this course.

Keep up the great work and best of luck with your submission!

Contents

1 Notebook Structure Map

1.1 Setup and Environment

- Importing libraries, setting up reproducibility and plotting functions.
- Initial configuration for the experiments.

1.2 Multi-Armed Bandit (MAB) Overview

- Description of the bandit problem in both stochastic and deterministic formulations.
- Discussion of reward probabilities.

1.3 Oracle Agent

- The Oracle uses privileged information to determine the maximum expected reward.
- **TODO:** Compute the oracle reward (2 points).
- Questions:
  - What insight does the oracle reward provide? (0.75 points)
  - Why is the oracle considered "cheating"? (0.75 points)

1.4 Random Agent (RndAg)

- This agent selects actions uniformly at random.
- **TODO:** Choose a random action (1 point).
- Questions:
  - Why is its reward lower and highly variable? (0.25 points)
  - How could the agent be improved without learning? (0.75 points)

1.5 Explore-First Agent (ExpFstAg)

- The agent explores randomly for a fixed number of steps and then exploits the best arm.
- **TODOs:**
  - Update Q-value (3 points)
  - Choose action based on exploration versus exploitation (1 point)
- Questions:
  - Why might a short exploration phase lead to fluctuations? (0.75 points)
  - What are the trade-offs of using a fixed exploration phase? (1 point)

1.6 UCB Agent (UCB_Ag)

- Uses an exploration bonus to balance learning.
- **TODOs:**
  - Update Q-value (3 points)
  - Compute the exploration bonus (4 points)

1.7 Epsilon-Greedy Agent (EpsGdAg)

- Selects the best-known action with probability $1 - \varepsilon$ and a random action with probability ε.
- **TODO:** Choose a random action based on ε (1 point)
- Questions:
  - Why does a high ε result in lower immediate rewards? (0.5 points)
  - What benefits might decaying ε over time offer? (0.75 points)

1.8 LinUCB Agent (Contextual Bandits)

- Leverages contextual features using a linear model.
- **TODOs:**
  - Compute UCB for an arm given context (7 points)
  - Update parameters A and b for an arm (7 points)
  - Compute UCB estimates for all arms (7 points)
  - Choose the arm with the highest UCB (4 points)
- Questions:
  - How does LinUCB leverage context to outperform classical methods? (1.25 points)
  - What role does the α parameter play in the exploration bonus? (1.25 points)

1.9 Final Comparison and Analysis

- **Comparison:** UCB vs. Explore-First agents.
  - Under what conditions might an explore-first strategy outperform UCB? (1.25 points)
  - How do design choices affect short-term vs. long-term performance? (1.25 points)
- Impact of extending the exploration phase (e.g., 20 vs. 5 steps). (1.5 points total)
- Discussion on why ExpFstAg might sometimes outperform UCB in practice. (2 points)

1.10 Final Deep-Dive Questions

- **Finite-Horizon Regret and Asymptotic Guarantees** (4 points)

  Many algorithms (e.g., UCB) are analyzed using asymptotic (long-term) regret bounds. In a finite-horizon scenario (say, 500–1000 steps), explain intuitively why an algorithm that is asymptotically optimal may still yield poor performance. What trade-offs arise between aggressive early exploration and cautious long-term learning? Deep Dive: Discuss how the exploration bonus, tuned for asymptotic behavior, might delay exploitation in finite time, leading to high early regret despite eventual convergence.

- **Hyperparameter Sensitivity and Exploration–Exploitation Balance** (4.5 points)

  Consider the impact of hyperparameters such as ϵ in ϵ-greedy, the exploration constant in UCB, and the α parameter in LinUCB. Explain intuitively how slight mismatches in these parameters can lead to either under-exploration (missing the best arm) or over-exploration (wasting pulls on suboptimal arms). How would you design a self-adaptive mechanism to balance this trade-off in practice? Deep Dive: Provide insight into the "fragility" of these parameters in finite runs and how a meta-algorithm might monitor performance indicators (e.g., variance in rewards) to adjust its exploration dynamically.

- **Context Incorporation and Overfitting in LinUCB** (4 points)

  LinUCB uses context features to estimate arm rewards, assuming a linear relation. Intuitively, why might this linear assumption hurt performance when the true relationship is complex or when the context is high-dimensional and noisy? Under what conditions can adding context lead to worse performance than classical (context-free) UCB? Deep Dive: Discuss the risk of overfitting to noisy or irrelevant features, the curse of dimensionality, and possible mitigation strategies (e.g., dimensionality reduction or regularization).

- **Adaptive Strategy Selection** (4.25 points)

  Imagine designing a hybrid bandit agent that can switch between an explore-first strategy and UCB based on observed performance. What signals (e.g., variance of reward estimates, stabilization of Q-values, or sudden drops in reward) might indicate that a switch is warranted? Provide an intuitive justification for how and why such a meta-strategy might outperform either strategy alone in a finite-time setting. Deep Dive: Explain the challenges in detecting when exploration is "enough" and how early exploitation might capture transient improvements even if the long-term guarantee favors UCB.

- **Non-Stationarity and Forgetting Mechanisms** (4 points)

  In non-stationary environments where reward probabilities drift or change abruptly, standard bandit algorithms struggle because they assume stationarity. Intuitively, explain how and why a "forgetting" or discounting mechanism might improve performance. What challenges arise in choosing the right decay rate, and how might it interact with the exploration bonus? Deep Dive: Describe the delicate balance between retaining useful historical information and quickly adapting to new trends, and the potential for "chasing noise" if the decay is too aggressive.

- **Exploration Bonus Calibration in UCB** (3.75 points)

  The UCB algorithm adds a bonus term that decreases with the number of times an arm is pulled. Intuitively, why might a "conservative" (i.e., high) bonus slow down learning—even if it guarantees asymptotic optimality? Under what circumstances might a less conservative bonus be beneficial, and what risks does it carry? Deep Dive: Analyze how a high bonus may force the algorithm to continue sampling even when an arm's estimated reward is clearly suboptimal, thereby delaying convergence. Conversely, discuss the risk of prematurely discarding an arm if the bonus is too low.

- **Exploration Phase Duration in Explore-First Strategies** (4 points)

  In the Explore-First agent (ExpFstAg), how does the choice of a fixed exploration period (e.g., 5 vs. 20 steps) affect the regret and performance variability? Provide a scenario in which a short exploration phase might yield unexpectedly high regret, and another scenario where a longer phase might delay exploitation unnecessarily. Deep Dive: Discuss how the "optimal" exploration duration can depend heavily on the underlying reward distribution's variance and the gap between the best and other arms, and why a one-size-fits-all approach may not work in practice.

- **Bayesian vs. Frequentist Approaches in MAB** (4 points)

  Compare the intuition behind Bayesian approaches (such as Thompson Sampling) to frequentist methods (like UCB) in handling uncertainty. Under what conditions might the Bayesian approach yield superior practical performance, and how do the underlying assumptions about prior knowledge

influence the exploration–exploitation balance? Deep Dive: Explore the benefits of incorporating prior beliefs and the risk of bias if the prior is mis-specified, as well as how Bayesian updating naturally adjusts the exploration bonus as more data is collected.

- **Impact of Skewed Reward Distributions** (3.75 points)

  In environments where one arm is significantly better (skewed probabilities), explain intuitively why agents like UCB or ExpFstAg might still struggle to consistently identify and exploit that arm. What role does variance play in these algorithms, and how might the skew exacerbate errors in reward estimation? Deep Dive: Discuss how the variability of rare but high rewards can mislead the agent's estimates and cause prolonged exploration of suboptimal arms.

- **Designing for High-Dimensional, Sparse Contexts** (5 points)

  In contextual bandits where the context is high-dimensional but only a few features are informative, what are the intuitive challenges that arise in using a linear model like LinUCB? How might techniques such as feature selection, regularization, or non-linear function approximation help, and what are the trade-offs involved? Deep Dive: Provide insights into the risks of overfitting versus underfitting, the increased variance in estimates from high-dimensional spaces, and the potential computational costs versus performance gains when moving from a simple linear model to a more complex one.