Welcome to the study of reinforcement learning! This set of lecture notes accompanies the undergraduate course CS/STAT 184 and is intended to be a friendly yet rigorous introduction to this exciting and active subfield of machine learning. Here are some questions you might have before embarking on this journey:

What is reinforcement learning (RL)? Broadly speaking, RL is a subfield of machine learning that studies how an agent can learn to make sequential decisions in an environment.

Why study RL? RL provides a powerful framework for attacking a wide variety of problems, including robotic control, video games and board games, resource management, language modelling, and more. It also provides an interdisciplinary paradigm for studying animal and human behavior. Many of the most stunning results in machine learning, ranging from AlphaGo to ChatGPT, are built on top of RL.

Is this book for me? This book assumes familiarity with multivariable calculus, linear algebra, and probability. For Harvard undergraduates, this would be fulfilled by Math 21a, Math 21b, and Stat 110. Stat 111 is strongly recommended but not required. Here is a non-comprehensive list of topics of which this book will assume knowledge:

- **Linear Algebra:** Vectors, matrices, matrix multiplication, matrix inversion, eigenvalues and eigenvectors, and the Gram-Schmidt process.
- **Multivariable Calculus:** Partial derivatives, gradient, directional derivative, and the chain rule.
- **Probability:** Random variables, probability distributions, expectation, variance, covariance, conditional probability, Bayes' rule, and the law of total probability.

How does reinforcement learning differ from other machine learning paradigms? Here is a list of comparisons:

- **Supervised learning.** Supervised learning concerns itself with learning a mapping from inputs to outputs (e.g. image classification). Typically the data takes the form of input-output pairs that are assumed to be sampled independently from some generating distribution. In RL, however, the data is generated by the agent interacting with the environment, meaning the observations depend on the agent's behaviour and are not independent from each other. This requires a more general set of tools.

  Conversely, supervised learning is a well-studied field that provides many useful tools for RL. For example, it may be useful to use supervised learning to predict how valuable a given state is, or to predict the probability of transitioning to a given state.

0.1 Overview

Chapter 1 introduces **Markov Decision Processes**, the dominant mathematical framework for studying RL. We'll discuss **dynamic programming** algorithms for solving MDPs, including **policy evaluation**, **policy iteration**, and **value iteration**.

Chapter 2 then discusses **multi-armed bandits**, a simpler problem that is often used as a warm-up to RL.

0.2 Notation

We will use the following notation throughout the book. This notation is inspired by Sutton and Barto and AJKS .

otation.

We will use *lowercase letters* to index over *uppercase letters*.

0.2.1 Multi-Armed Bandits

$[N]$	The set $\{0, 1, \dots, N-1\}$.
K	The number of arms
T	The number of time steps (i.e. algorithm iterations).

0.2.2 MDPs

$\mathcal{S}$	The state space.
$\mathcal{A}$	The action space.
$s \in \mathcal{S}$	A state.
$a \in \mathcal{A}$	An action.
$r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$	The reward function.
$P : \mathcal{S} times \mathcal{A} \to \triangle(\mathcal{S})$	The transition probabilities.
$\gamma \in (0, 1)$	The discount factor.
$\pi : \mathcal{S} \to \mathcal{A}$	A policy.
$V^\pi(s) \in \mathbb{R}$	The value function of policy π.
$Q^\pi(s, a) \in \mathbb{R}$	The action-value function of policy π.
$\pi^\star, V^\star, Q^\star$	The optimal policy, value function, and action-value function.

0.2.3 Vectors

Note that a vector $v \in \mathbb{R}^d$ can be thought of as a function from $v : [d] \to \mathbb{R} : i \mapsto v_i$. Similarly, we will often identify a real-valued function on states, such as the transition matrix for a given state-action pair, with the corresponding vector. That is, we will treat $P(\cdot \mid s, a) \in \mathbb{R}^{|\mathcal{S}|}$.

0.3 Challenges of reinforcement learning

Exploration-exploitation tradeoff. Should the agent try a new action or stick with the action that it knows is good?

Prediction. The agent might want to predict the value of a state or state-action pair.

Policy computation (control). In a complex environment, even if the dynamics are known, it can still be challenging to compute the best policy.

0.4 Resources

Inspired by the Stat 110 textbook and Stat 111 lecture notes.

This book seeks to provide an intuitive understanding before technical treatment.

Refer to AJK Sutton and Barto, online sources.