ISE/ECE 7202: Reinforcement Learning

Ohio State University, Autumn 2021

Parinaz Naghizadeh

naghizadeh.1@osu.edu

Lecture 1: Introduction

August 24, 2021

Outline

- Introduction and general concepts
- About this course, topics to be covered
- Course logistics

What is reinforcement learning?

Informally speaking, reinforcement learning is learning how to behave optimally in an unknown environment through trial-and-error.

What is reinforcement learning?

Informally speaking, reinforcement learning is learning how to behave optimally in an unknown environment through trial-and-error.

A little more formally, it is a computational approach to the following problem of learning from interaction: a learner/agent observes its environment, takes an action, observes its consequences, and uses these "reinforcements" to plan future behavior so as to achieve some (long-term) goal.

What is exciting about reinforcement learning?

The game-playing AIs: AlphaGo (2015), AlphaGo Zero (2017), MuZero (2019), ...

- Beats human experts
- Learns from scratch: later versions do not rely on seeing any examples of other humans' plays, or even an accurate simulator
- Plays different, discovers new winning moves!
- Portable algorithm: in principle, can be applied to many other tasks

RL is not a recent discovery...

There are even electro-mechanical machines demonstrating trial-and-error learning

`https://techchannel.att.com/play-video.cfm/2010/3/16/In-Their-Own-Words-Claude-Shannon-Demonstrates-Machine-Learning`

RL is not a recent discovery...

There are even electro-mechanical machines demonstrating trial-and-error learning
https://techchannel.att.com/play-video.cfm/2010/3/16/In-Their-Own-Words-Claude-Shannon-Demonstrates-Machine-Learning

However...

- Recent interest largely due to improvement in computational power
- as well as some new algorithms and advances in the field

Relation of RL to other fields (I)

Optimal control and operations research, in particular dynamic programming

- In principle, the sequential decision problems we study in this course are solvable by DP.
- Many of the algorithms we see in this course are routed in DP ideas.
- Hence the equivalent names "approximate dynamic programming" and "neuro-dynamic programming".
- We will see that RL can be used as opposed to DP in problems where:
  1. the environment model is not fully known to the agent,
  2. (typically) the agent is not trying to learn the dynamics of the environment, and
  3. parameterized approximations can be used to address DP's "curse of dimensionality".

Relation of RL to other fields (I)

Optimal control and operations research, in particular dynamic programming

- In principle, the sequential decision problems we study in this course are solvable by DP.
- Many of the algorithms we see in this course are routed in DP ideas.
- Hence the equivalent names "approximate dynamic programming" and "neuro-dynamic programming".
- We will see that RL can be used as opposed to DP in problems where:
  1. the environment model is not fully known to the agent,
  2. (typically) the agent is not trying to learn the dynamics of the environment, and
  3. parameterized approximations can be used to address DP's "curse of dimensionality".

Relation of RL to other fields (I)

Optimal control and operations research, in particular dynamic programming

- In principle, the sequential decision problems we study in this course are solvable by DP.
- Many of the algorithms we see in this course are routed in DP ideas.
- Hence the equivalent names "approximate dynamic programming" and "neuro-dynamic programming".
- We will see that RL can be used as opposed to DP in problems where:
  1. the environment model is not fully known to the agent,
  2. (typically) the agent is not trying to learn the dynamics of the environment, and
  3. parameterized approximations can be used to address DP's "curse of dimensionality".

Relation of RL to other fields (I)

Optimal control and operations research, in particular dynamic programming

- In principle, the sequential decision problems we study in this course are solvable by DP.
- Many of the algorithms we see in this course are routed in DP ideas.
- Hence the equivalent names "approximate dynamic programming" and "neuro-dynamic programming".
- We will see that RL can be used as opposed to DP in problems where:
    1. the environment model is not fully known to the agent,
    2. (typically) the agent is not trying to learn the dynamics of the environment, and
    3. parameterized approximations can be used to address DP's "curse of dimensionality".

Relation of RL to other fields (II)

RL as an ML paradigm

- Supervised learning
- Unsupervised learning
- Reinforcement learning

Relation of RL to other fields (II)

RL as an ML paradigm

- Supervised learning (instructive vs evaluative feedback)
- Unsupervised learning
- Reinforcement learning

Relation of RL to other fields (II)

RL as an ML paradigm

- Supervised learning (instructive vs evaluative feedback)
- Unsupervised learning (uncovering structures does not necessarily inform reward maximization)
- Reinforcement learning

Relation of RL to other fields (II)

RL as an ML paradigm

- Supervised learning (instructive vs evaluative feedback)
- Unsupervised learning (uncovering structures does not necessarily inform reward maximization)
- Reinforcement learning

$\Rightarrow$ RL is unique in having evaluative feedback and a sequential nature.
As such it raises the challenge of exploration vs exploitation.

Main elements of an RL problem (I)

- **The agent** and its **actions**.
- **The environment** and its **state**.
- **The model**, or dynamics of the environment. Knowledge/use of this information can be used to classify RL methods into model-free vs model-based methods.
- **The policy**: describing how the agent behaves, or more formally, the (probability of) choice of an action given an observed state.

Main elements of an RL problem (I)

- **The agent** and its **actions**.
- **The environment** and its **state**.
- **The model**, or dynamics of the environment. Knowledge/use of this information can be used to classify RL methods into model-free vs model-based methods.
- **The policy**: describing how the agent behaves, or more formally, the (probability of) choice of an action given an observed state.

Main elements of an RL problem (II)

- **The reward**: the immediate feedback from the environment, which depends on the agent's action and the environment's state.
- **The value function**: the notion of long-term goodness of a state. It estimates how much reward an agent expects to accumulate if starting from a certain state. Takes future states and their rewards into consideration.

Main elements of an RL problem (II)

- **The reward**: the immediate feedback from the environment, which depends on the agent's action and the environment's state.
- **The value function**: the notion of long-term goodness of a state. It estimates how much reward an agent expects to accumulate if starting from a certain state. Takes future states and their rewards into consideration.

Main elements of an RL problem (II)

- **The reward**: the immediate feedback from the environment, which depends on the agent's action and the environment's state.
- **The value function**: the notion of long-term goodness of a state. It estimates how much reward an agent expects to accumulate if starting from a certain state. Takes future states and their rewards into consideration.

Perhaps the most important component of almost all RL algorithms is a method for efficiently estimating the value functions.

About this course: topics to be covered

Background and formalization

- Simple examples and terminology (x1)
- Multi-armed bandits (x2)
- Markov decision processes (MDPs) (x2-3)

About this course: topics to be covered

Background and formalization

- Simple examples and terminology (x1)
- Multi-armed bandits (x2)
- Markov decision processes (MDPs) (x2-3)

Tabular methods

- Dynamic programming (x3-4)
- Monte Carlo methods (x2)
- Temporal difference methods, e.g., Q-learning (x2)
- Summary and other methods (x1)

About this course: topics to be covered

Background and formalization

- Simple examples and terminology (x1)
- Multi-armed bandits (x2)
- Markov decision processes (MDPs) (x2-3)

Tabular methods

- Dynamic programming (x3-4)
- Monte Carlo methods (x2)
- Temporal difference methods, e.g., Q-learning (x2)
- Summary and other methods (x1)

Approximate solution methods

- RL with function approximation (x3)
- Policy gradient methods (x2)
- Actor-critic methods (x1)
- Summary and other methods (x1)

About this course: topics to be covered

Background and formalization

- Simple examples and terminology (x1)
- Multi-armed bandits (x2)
- Markov decision processes (MDPs) (x2-3)

Tabular methods

- Dynamic programming (x3-4)
- Monte Carlo methods (x2)
- Temporal difference methods, e.g., Q-learning (x2)
- Summary and other methods (x1)

Approximate solution methods

- RL with function approximation (x3)
- Policy gradient methods (x2)
- Actor-critic methods (x1)
- Summary and other methods (x1)

Advanced topics (x4)

- Multi-agent RL
- Off-policy evaluation and learning
- Reward design and shaping
- Imitation learning, inverse RL
- etc

Why learn all these RL methods?

"There are no methods that are guaranteed to work for all or even most problems, but there are enough methods to try on a given challenging problem with a reasonable chance of success at the end."

D. Bertsekas, "Reinforcement learning and optimal control".

Some challenges and limitations

- Defining and representing the state is challenging. We abstract from it and focus on decision making only.
- Reward function design and shaping: choosing the size of rewards, addressing reward sparsity.
- Design of other elements of the RL system in simulations: the environment, choice of action space, etc.

Some challenges and limitations

- Defining and representing the state is challenging. We abstract from it and focus on decision making only.
- Reward function design and shaping: choosing the size of rewards, addressing reward sparsity.
- Design of other elements of the RL system in simulations: the environment, choice of action space, etc.
- Words (or videos) of caution :)
  - RL for improving agent design
    `https://storage.googleapis.com/quickdraw-models/sketchRNN/designrl/bipedhard_compare_vs_augment.mp4`
    `https://storage.googleapis.com/quickdraw-models/sketchRNN/designrl/augmentbipedhard.lognormal.blooper.mp4`
  - Design of a robotic arm to grasp and move blocks
    `https://medium.com/@BonsaiAI/deep-reinforcement-learning-models-tips-tricks-for-writing-rew`
  - The Cobra Effect.

Course logistics

- **Office hours**: Tuesdays 11:30am-12:30pm, Thursdays 4-5pm.
- **Homework (50% of grade)**: 5 homeworks, roughly biweekly. Typically includes reading a paper as well as a programming problem. There is also a bonus homework 0 due next week.
- **Project (50% of grade)**: Groups of 2 highly encouraged. Includes choosing a problem in your field of research/choice, motivation, formulation, some literature review, and (if applicable) solving it using one of the algorithm's from class or otherwise a proposed solution approach. Graded based on interim steps, final in-class presentation, and a final report.

Course logistics

- **Office hours**: Tuesdays 11:30am-12:30pm, Thursdays 4-5pm.
- **Homework (50% of grade)**: 5 homeworks, roughly biweekly. Typically includes reading a paper as well as a programming problem. There is also a bonus homework 0 due next week.
- **Project (50% of grade)**: Groups of 2 highly encouraged. Includes choosing a problem in your field of research/choice, motivation, formulation, some literature review, and (if applicable) solving it using one of the algorithm's from class or otherwise a proposed solution approach. Graded based on interim steps, final in-class presentation, and a final report.

Course logistics

- **Office hours**: Tuesdays 11:30am-12:30pm, Thursdays 4-5pm.
- **Homework (50% of grade)**: 5 homeworks, roughly biweekly. Typically includes reading a paper as well as a programming problem. There is also a bonus homework 0 due next week.
- **Project (50% of grade)**: Groups of 2 highly encouraged. Includes choosing a problem in your field of research/choice, motivation, formulation, some literature review, and (if applicable) solving it using one of the algorithm's from class or otherwise a proposed solution approach. Graded based on interim steps, final in-class presentation, and a final report.

Next lecture

- Key sequential decision-making terminology and mathematical notation
- A (simple) example of an RL problem
- Introduction to multi-armed bandits
- **Homework:** (Bonus) homework 0 will be posted tonight, and will be due in one week