Introduction to Reinforcement Learning

Reinforcement Learning

- The science of decision making
- Equivalents in other fields
  - Engineering: Optimal Control
  - Neuroscience: Dopamine / Reward System (TD is very similar)
  - Psychology: Classical Conditioning (animal behaviour)
  - Math: Operations Research
  - Economics: Game Theory / Utility Theory
- Unifying Framework for sequential decision making problems

Reinforcement vs Supervised

- No supervisor
- There is reward signal, but no indication of ‘best solution’
  - Feedback doesn’t rely on knowledge of correct action
  - Learning from a critic instead of a teacher
    - Movie critics can tell when a movie is good or bad, and give a rating, but they don’t know what the ‘perfect’ movie is
- Delayed rewards / feedback

Sequential Learning and Decision Making

- Dynamic system / environment
- Continuous time
- Agent gets to influence environment and its perception (data it receives)
- Goal: Select actions to maximize total future reward
- Actions may have long term consequences
- Reward may be delayed
- It may be better to sacrifice immediate reward to gain more long-term reward
- Agent and Environment
  - Observation (received from environment)
  - Reward (received from environment)
  - Action (influences the environment)
  - At each time step t the agent
    - Executes action A_t
    - Receives observation O_t
    - Receives scalr reward R_t
  - The environment
    - Receives action A_t
    - Emits Observation O_t
    - Emits scalar reward R_t

History and State

- The history (H_t) is the complete sequence of observations, actions and rewards up to time t
- History determines what the environment will do next, but can get too long for the agent to use fully
- The state is the information used to determine what happens next

Environment State S^e_t

- The environment's private representation
- I.e. whatever data the environment uses to pick the next observation / reward
- Not usually visible to the agent
- May contain irrelevant information

Information State (Markov State)
Contains all useful information from the history

A state State S_t is Markov iff

- $\rho[S_{t+1} \mid S_t] = \rho[S_{t+1} \mid S_1, ..., S_t]$
- Current Markov State would give the same decision as if all the previous Markov States were given

Agent State S^a_t

- The agent's internal representation
- Agent gets to choose how to create its own Agent State
- Whatever information the agent uses to pick the next action
- I.e. the information used by RL algorithms
- Can be any function of history $S^a_t = f(H_t)$

Environment state is Markov

- Markov state for stunt helicopter would be
    - Velocity/Position/Fuel/Wind
- All previous info can be discarded

Policy, Value, and Model

Policy

- Maps from state to action
- The 'brain' of the agent
- Can be probabilistic or deterministic

Value function

- Prediction of expected future reward
    - Value function (usually) decreases after a reward is obtained
    - *Future* reward decreases
- Automatically accounts for risk

Model

- Predicts what the environment will do next
- Transitions Model
    - ρ predicts the next state (i.e. environment is dynamic)
- Rewards Model
    - R predicts the next (immediate) reward

Value Discounting

- Value short term more than long term
    - Gradual, each term is multiplied by
    - $\gamma^n R_{t+n}$
        - γ is discount factor, n is time steps in the future

- Indirectly determines distance of look ahead

Categorizing RL Agents

Value Based
- No Policy (implicit)
  - Value function provides all need info for decisions
- Uses Value Function

Policy Based
- Policy
- No value function

Model Free
- Policy and / or Value Function
- No Model
  - Doesn't try to explicitly understand how environment works

Model Based
- Policy and / or Value Function
- Model
  - Can be used for look-ahead

Actor Critic
- Policy
- Value Function

Problems within Reinforcement Learning

Learning and Planning: Sequential decision making

Reinforcement Learning
- The environment is initially unknown
- The agent interacts with the environment
- The agent improves its policy

Planning
- A model of the environment is known
- The agent performs computations with its model (without any external interaction)

Exploration and Exploitation

Exploitation
- Exploits known information to maximize reward (Go to your favourite restaurant)

Exploration
- Finds more information about the environment (Try a new restaurant)

A balance of both is needed
- Exploration only → Never 'cash in'
- Exploitation only → Never 'improve'

RL is like trial-and-error learning
- Agent should discover a good policy - from its experiences of the environment

Prediction and Control

Prediction
Evaluate the future reward given a policy

Control
Optimize the future reward → Find the best policy

Prediction problem must be solved before the control problem can be solved

Credit-Assignment - How do you distribute credit for success among the many decisions that may have been involved in producing it?

Applications of RL

- Managing investment portfolio
- Backgammon
- Controlling a power station
- Robotic movement
- Robotic stunt maneuvers (helicopter)
- Multi-use agents (agent that can play any Atari game)

Course Outline

Part 1: Elementary Reinforcement Learning

- Introduction to RL
- Markov Decision Processes
- Planning by Dynamic Programming
- Model-Free Prediction
- Model-Free Control

Part 2: Reinforcement Learning in Practice

- Value Function Approximation
- Policy Gradient Methods
- Integrating Learning and Planning
- Exploration and Exploitation
- Case Study - RL in Game