Chapter 1

• Give an overview of the whole RL problem...
  ■ Before we break it up into parts to study individually

• Introduce the cast of characters
  ■ Experience (reward)
  ■ Policies
  ■ Value functions
  ■ Models of the environment

• Tic-Tac-Toe example

adapted from R. S. Sutton and A. G. Barto: Reinforcement Learning: An Introduction
What is Reinforcement Learning?

- An approach to Artificial Intelligence
- Learning from interaction
- Goal-oriented learning
- Learning about, from, and while interacting with an external environment
- Learning what to do—how to map situations to actions—so as to maximize a numerical reward signal
Complete Agent

- Temporally situated
- Continual learning and planning
- Object is to affect the environment
- Environment is stochastic and uncertain

adapted from R. S. Sutton and A. G. Barto: Reinforcement Learning: An Introduction
Connections to other Fields

Artificial Intelligence

Psychology

Control Theory and Operations Research

Reinforcement Learning (RL)

Neuroscience

Artificial Neural Networks
Classical Conditioning

Pavlov’s finding (1890-1900s, nobel prize 1904):

Initially, sight of food leads to dog salivating. If sound of bell consistently accompanies or precedes presentation of food, then after a while the sound of the bell leads to salivating.
**Terminology**

- **food** → **salivating**
- **unconditioned stimulus, US (reward)** → **unconditioned response, UR**

Sound of bell consistently precedes food. Afterwards, bell leads to salivating:

- **bell** → **salivating**
- **conditioned stimulus, CS (expectation of reward)** → **conditioned response, CR**
Instrumental Conditioning

- **Classical Conditioning**: only concerned with prediction of reward; doesn’t consider agent’s actions. Reward usually depends on what you’ve done!

Edward L. Thorndike's "Law of effect" (1911):

- Responses to a situation that are followed by satisfaction are strengthened.
- Responses that are followed by discomfort are weakened.
Thorndike “puzzle box”
“Of several responses made to the same situation, those which are accompanied or closely followed by satisfaction to the animal will, other things being equal, be more firmly connected with the situation, so that, when it recurs, they will be more likely to recur; those which are accompanied or closely followed by discomfort to the animal will, other things being equal, have their connections with that situation weakened, so that, when it recurs, they will be less likely to occur. The greater the satisfaction or discomfort, the greater the strengthening or weakening of the bond”

(Thorndike, 1911, p. 244)
Skinner Box

Diagram showing a Skinner Box with labeled parts:
- Pellet dispenser
- Dispenser tube
- Speaker
- Signal lights
- Lever
- Electric grid
- Food cup
- To shock generator
Clicker Training

- used by animal trainers to teach behaviors

- developed by Marian Kruse and Keller Breland (two of Skinner’s first students)

- used on over 150 species and robots [Kaplan et al., 2002]

- basic idea: use "click" as a secondary reinforcer.

- trainer "marks" desired behavior with click at precisely the right time
Key Features of RL

• Learner is not told which actions to take
• Trial-and-Error search
• Possibility of delayed reward
  ▪ Sacrifice short-term gains for greater long-term gains
• The need to *explore* and *exploit*
• Considers the whole problem of a goal-directed agent interacting with an uncertain environment
Examples of Reinforcement Learning

- **Robocup Soccer Teams**  Stone & Veloso, Riedmiller et al.
  - World’s best player of simulated soccer, 1999; Runner-up 2000

- **Inventory Management**  Van Roy, Bertsekas, Lee & Tsitsiklis
  - 10-15% improvement over industry standard methods

- **Dynamic Channel Assignment**  Singh & Bertsekas, Nie & Haykin
  - World’s best assigner of radio channels to mobile telephone calls

- **Elevator Control**  Crites & Barto
  - (Probably) world’s best down-peak elevator controller

- **Many Robots**
  - navigation, bi-pedal walking, grasping, switching between skills...

- **TD-Gammon and Jellyfish**  Tesauro, Dahl
  - World’s best backgammon player

- **Many more recent ones...**
Supervised Learning

Training Info = desired (target) outputs

Error = (target output − actual output)
Reinforcement Learning

Training Info = evaluations ("rewards" / "penalties")

Inputs → RL System → Outputs ("actions")

Objective: get as much reward as possible
Elements of RL

- **Policy**: what to do
- **Reward**: what is good
- **Value**: what is good because it *predicts* reward
- **Model**: what follows what

adapted from R. S. Sutton and A. G. Barto: Reinforcement Learning: An Introduction
A Somewhat Less Misleading View...

adapted from R. S. Sutton and A. G. Barto: Reinforcement Learning: An Introduction
An Extended Example: Tic-Tac-Toe

Assume an imperfect opponent:
— he/she sometimes makes mistakes

adapted from R. S. Sutton and A. G. Barto: Reinforcement Learning: An Introduction
An RL Approach to Tic-Tac-Toe

1. Make a table with one entry per state:

<table>
<thead>
<tr>
<th>State</th>
<th>( V(s) ) – estimated probability of winning</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \boxed{\ + ! ! ! ! } )</td>
<td>.5</td>
</tr>
<tr>
<td>( \boxed{! ! ! ! - ! ! ! ! } )</td>
<td>.5</td>
</tr>
<tr>
<td>( \ldots )</td>
<td>\ldots</td>
</tr>
<tr>
<td>( \boxed{! ! ! ! X ! ! ! ! } )</td>
<td>1</td>
</tr>
<tr>
<td>( \ldots )</td>
<td>\ldots</td>
</tr>
<tr>
<td>( \boxed{! ! ! ! O ! ! ! ! } )</td>
<td>0</td>
</tr>
<tr>
<td>( \ldots )</td>
<td>\ldots</td>
</tr>
<tr>
<td>( \boxed{! ! ! ! O ! ! ! ! } )</td>
<td>0</td>
</tr>
</tbody>
</table>

2. Now play lots of games. To pick our moves, look ahead one step:

- Just pick the next state with the highest estimated prob. of winning — the largest \( V(s) \); a greedy move.
- But 10% of the time pick a move at random; an exploratory move.

adapted from R. S. Sutton and A. G. Barto: Reinforcement Learning: An Introduction
RL Learning Rule for Tic-Tac-Toe

Starting Position

On „greedy“ moves: We increment each $V(s)$ toward $V(s')$ – a **backup**:

$$V(s) \leftarrow V(s) + \alpha [V(s') - V(s)]$$

„Temporal Difference Learning“

a small positive fraction, e.g., $\alpha = .1$

the **step-size parameter**

$s$ – the state before our greedy move

$s'$ – the state after our greedy move

adapted from R. S. Sutton and A. G. Barto: Reinforcement Learning: An Introduction
How can we improve this Tic-Tac-Toe player?

• Take advantage of symmetries
  ■ representation/generalization
  ■ How might this backfire?
• Do we need “random” moves? Why?
  ■ Do we always need a full 10%?
• Can we learn from “random” moves?
• Can we learn offline?
  ■ Pre-training from self play?
  ■ Using learned models of opponent?
• ...
### e.g. Generalization

#### Table

<table>
<thead>
<tr>
<th>State</th>
<th>( V )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( s_1 )</td>
<td></td>
</tr>
<tr>
<td>( s_2 )</td>
<td></td>
</tr>
<tr>
<td>( s_3 )</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>( s_N )</td>
<td></td>
</tr>
</tbody>
</table>

#### Generalizing Function Approximator

<table>
<thead>
<tr>
<th>State</th>
<th>( V )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Train here

---

adapted from R. S. Sutton and A. G. Barto: Reinforcement Learning: An Introduction
### e.g. Generalization

<table>
<thead>
<tr>
<th>State</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_1$</td>
<td></td>
</tr>
<tr>
<td>$s_2$</td>
<td></td>
</tr>
<tr>
<td>$s_3$</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>$s_N$</td>
<td></td>
</tr>
</tbody>
</table>

Train here

---

adapted from R. S. Sutton and A. G. Barto: Reinforcement Learning: An Introduction
How is Tic-Tac-Toe Too Easy?

- Finite, small number of states
- One-step look-ahead is always possible
- State completely observable
- ...
The (old version of the) Book

- Part I: The Problem
  - Introduction
  - Evaluative Feedback
  - The Reinforcement Learning Problem
- Part II: Elementary Solution Methods
  - Dynamic Programming
  - Monte Carlo Methods
  - Temporal Difference Learning
- Part III: A Unified View
  - Eligibility Traces
  - Generalization and Function Approximation
  - Planning and Learning
  - Dimensions of Reinforcement Learning
  - Case Studies